Process Improvement Using Data

Release 10d109

Kevin Dunn

01 February 2023

CONTENTS

7 Applications of Process Improvement using Data   403

Index   407

This book is a guide on how to improve processes using the large quantities of data that are routinely collected from process systems. It is in a state of a *semi-permanent draft*.

We cover *data visualization* (page 1) first, in Chapter 1, since most data analysis studies start by plotting the data. This is an extremely brief introduction to this topic, only illustrating the most basic plots required for this book. Please consult the references in this chapter for more exciting plots that provide insight to your data.

This is followed by Chapter 2 on *univariate data analysis* (page 29), which is a comprehensive treatment of univariate techniques to *quantify variability* and then to *compare variability*. We look at various univariate distributions and consider tests of significance from a confidence-interval viewpoint. This is arguably a more useful and intuitive way, instead of using hypothesis tests.

The next chapter, Chapter 3, is on *monitoring charts* (page 109) to track variability: a straightforward application of univariate data analysis and data visualization from the previous two chapters.

Chapter 4 introduces the area of multivariate data. The first natural application is *least squares modelling* (page 151), where we learn how variation in one variable is related to another variable. This chapter briefly covers multiple linear regression and outliers. We don't cover nonlinear regression models but hope to add that in future updates to the book.

Chapter 5 covers *designed experiments* (page 231), where we intentionally introduce variation into our system to learn more about it. We learn how to use the models from the experiments to optimize our process (e.g. for improved profitability).

The final chapter, Chapter 6, is on *latent variable modelling* (page 315) where we learn how to deal with multiple variables and extract information from them. This section is divided in several chapters (PCA, PLS, and applications). It is still a work in progress and will be improved in the future.

Because this is a predominantly electronic book, we resort to many hyperlinks in the text. We recommend a good PDF reader that allows forward and back navigation of links. However, we have ensured that a printed copy can be navigated just as easily, especially if you use the table of contents and index for cross referencing.

Updates: This book is continually updated; there isn't a fixed edition. You should view it as a wiki. You might currently have an incomplete or older draft of the document. The latest version is always available at https://learnche.org/pid.

Stuart Young (2011), Mudassir Rashid (2011), Yasser Ghobara (2012), Pedro Castillo (2012), Miles Montgomery (2012), Cameron DiPietro (2012), Andrew Haines (2012), Krishna Patel (2012), Xin Yuan (2013), Sean Johnstone (2013), Jervis Pereira (2013), and Ghassan Marjaba (2014), Kyla Sask (2015, and 2016). *Their contributions are greatly appreciated.*

The textbook was used in an online course from July to August 2014, Experimentation for Improvement[1]. Comments and feedback from that course have greatly improved this book. *Thanks to all the Courserians*. That Coursera course was relaunched, and is still active. All videos created for that, as well as videos created for the Ontario Online Initiative have been embedded in the textbook. Look for the YouTube videos on the web version of this book, or if reading it from a PDF, watch for the icon shown.

In particular, I'd like to thank Devon Mordell, from McMaster University, for her informal help on editing parts of the book. As well as countless others who have via email or web forms provided feedback. Any errors, poor spelling and grammar are entirely my own fault – any feedback to improve them will be appreciated[2].

Thanks also to instructors at other universities who have used these notes and slides in their courses and provided helpful feedback.

---

[1] https://www.coursera.org/learn/experimentation
[2] https://docs.google.com/forms/d/e/1FAIpQLScENs2JsKnS1HL4OAlu__blZZlcJdc7P0yDuSvyM3X7mLqsoQ/viewform
[3] https://creativecommons.org/licenses/by-sa/4.0/
[4] kgdunn@gmail.com

1.1 Data visualization in context

This is the first chapter in the book. Why? Many of you have heard the phrase "plot your data," but seldom are we shown what appropriate plots look like.

In this section we consider quantitative plots – plots that show numbers. We cover various plots that will help you gain more insight from your data. We end with a list of tips for effective data visualization.

Usage examples

You can use the material in this chapter when you must learn more about your system from the data. For example, you may get these questions:

- *Co-worker*: Here are the yields (final output value) from a given system for the last 3 years (1256 data points). Can you help me:
    - effectively communicate what the time trends are in the data?
    - summarize the yield values?

- *Manager*: How can we effectively summarize the (a) number and (b) types of defects on our 17 products for the last 12 months?

- *Yourself*: We produce products in a batchwise manner. For each batch we have 25 different sensors that we record a value for at a rate of 5 readings per minute, over a total interval of 300 minutes. How can we visualize these $25 \times 5 \times 300 = 37500$ data points?

What we will cover
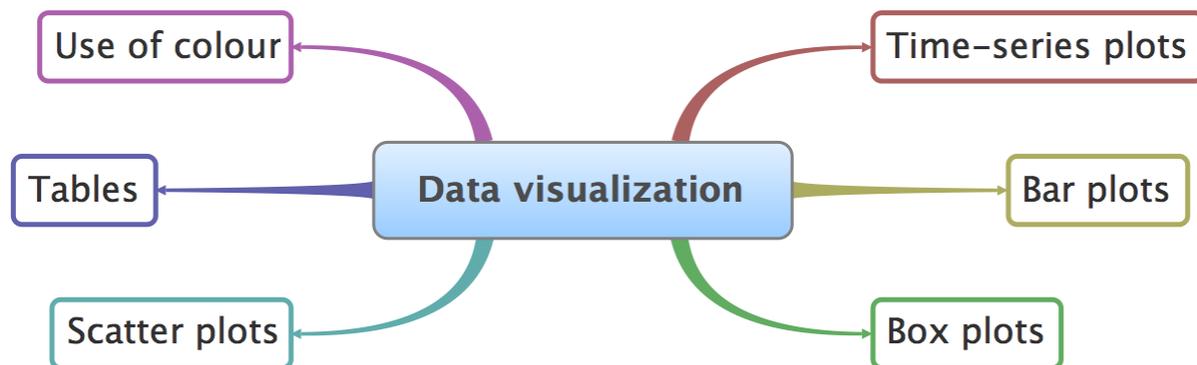
1.2 References and readings

1. Edward Tufte, *Envisioning Information*, Graphics Press, 1990. (10th printing in 2005)

2. Edward Tufte, *The Visual Display of Quantitative Information*, Graphics Press, 2001.

3. Edward Tufte, *Visual Explanations: Images and Quantities, Evidence and Narrative*, 2nd edition, Graphics Press, 1997.

4. Stephen Few, *Show Me the Numbers* and *Now You See It: Simple Visualization Techniques for Quantitative Analysis*; both from Analytics Press.

5. William Cleveland, *Visualizing Data*, 1st edition, Hobart Press, 1993.

6. William Cleveland, *The Elements of Graphing Data*, 2nd edition, Hobart Press, 1994.

7. Su, It's Easy to Produce Chartjunk Using Microsoft Excel 2007 but Hard to Make Good Graphs[5], *Computational Statistics and Data Analysis*, **52** (10), 4594-4601, 2008.

1.3 Time-series plots

Video for this section.

We start off by considering a plot most often seen in engineering applications: the time-series plot. The time-series plot is a univariate plot: it shows only one variable. It is a 2-dimensional plot in which one axis, the time-axis, shows graduations at an appropriate scale (seconds, minutes, weeks, quarters, years), while the other axis shows the numeric values. Usually, the time-axis is displayed horizontally, but this is not a requirement: some interesting analysis can be done with time running vertically.

Many statistical packages call this a line plot, as it can be used generally to display any sort of sequence, whether it is along time or some other ordering. The time-series plot is an excellent way to visualize long sequences of data. It tells a visual story along the sequence axis, and the human brain is incredible at absorbing this high density of data, locating patterns in the data such as sinusoids, spikes, and outliers, and separating any noise from signal.

Here are some tips for effective time-series plots:

• The software should have horizontal and vertical zooming ability. Once zoomed in, there must be tools to scroll up, down, left and right.

---

[5] https://dx.doi.org/10.1016/j.csda.2008.03.007

- Always label the *x*-axis appropriately with (time) units that make sense.
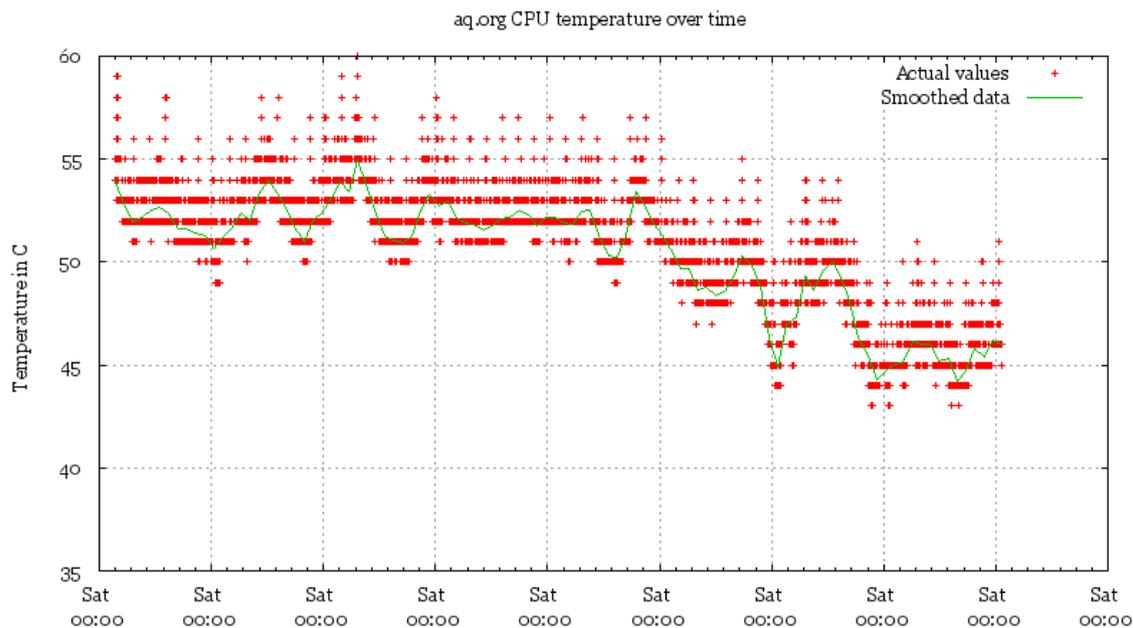
This plot, found on the Internet, shows a computer's CPU temperature with time. There are several problems with the plot, but the key issue here is the *x*-axis. This plot is probably the result of poor default settings in the software, but as you will start to realize, bad defaults are very common in most software packages. They waste your time when you have to repeatedly modify the charts, especially if you are just starting out with exploring the data. Good software will sensibly label the time-based axis for you.

- When plotting more than one trajectory (a vector of values) against time, it is helpful if the lines do not cross or jumble too much. This allows you to clearly see the relationship with other variables. The use of a second *y*-axis on the right-hand side is helpful when plotting two trajectories, but when plotting three or more trajectories that are in the same numeric range, it is better to use several parallel axes.
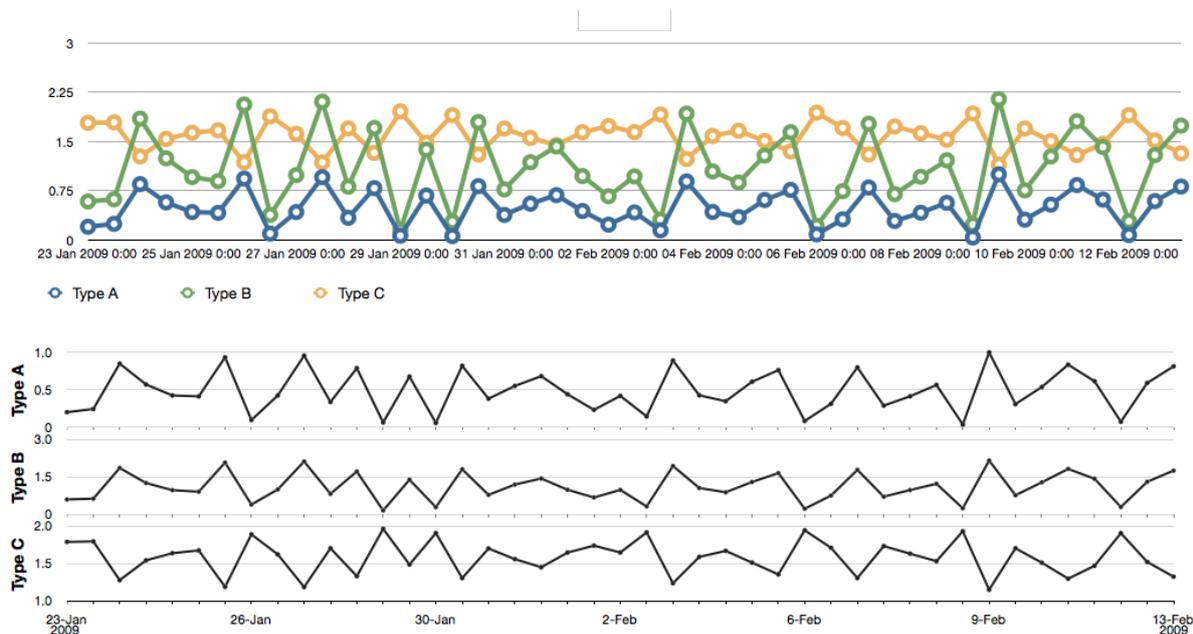
- Using the same data as in the previous tip, a much improved visualization technique is to use sparklines to represent the sequence of data.
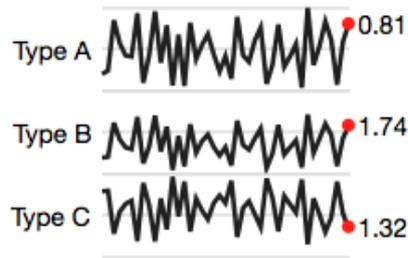
Sparklines are small graphics that carry a high density of information. The human eye is easily capable of absorbing about 100 dots or points per linear centimeter and around 10000 points per square centimeter. These sparklines convey the same amount of information as the previous plots and are easy to consume on hand-held devices such as cellphones and tablet computing devices that are common in chemical plants and other engineering facilities. Read more about them from this hyperlink[6].

- When plotting money values over time (e.g. sales of your product over the past 10 years), adjust for inflation effects by dividing by the consumer price index or an appropriate factor. Distortions due to the time value of money can be very misleading, as this example of retail sales shows[7]. For Canadians, here is a Canadian inflation calculator[8] from the Bank of Canada that can help you. For most countries you can almost certainly find something similar from the country's national bank or a government office.

- If you ever ask yourself, "Why are we being shown so little?" then you must request more data before and after the time period or current sequence shown. A typical example is stock-price data (see *example figure of Apple's stock* (page 4)). There are numerous graphical "lies" in magazines and reports where the plot shows a drastic change in trend, but in the context of prior data, that trend is a small aberration. Again, this brings into play the brain's remarkable power to discern signal from noise, but to do this, our brains require context. Ask for the extra context, or look for it, if not provided.

---

[6] https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0001OR
[7] http://people.duke.edu/~rnau/411infla.htm
[8] https://www.bankofcanada.ca/rates/related/inflation-calculator

1. Got to buy some of this stock!
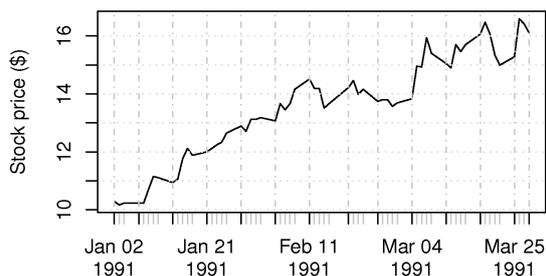
2. But, here is some more context

3. And, even further context
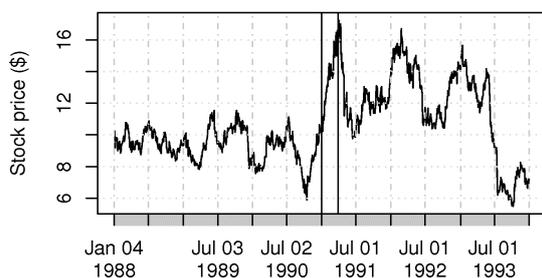
4. To finish: all available data
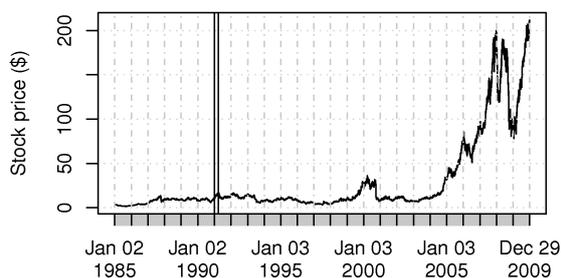
1.4 Bar plots

The bar plot is another univariate plot on a two-dimensional axis. The two axes are not called *x*- or *y*-axes. Instead, one axis is called the *category axis* showing the category name, while the other, the *value axis*, shows the numeric value of that category, given by the length of the bar.
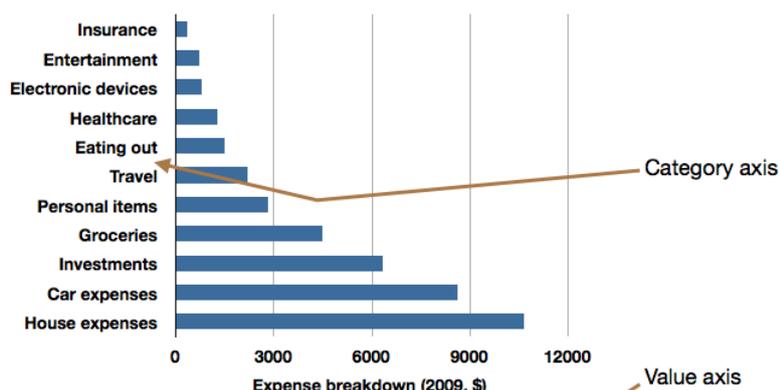
Here is some advice for bar plots:

- Use a bar plot when there are many categories and interpretation of the plot does not differ if the category axis is reshuffled. (It might be easier to interpret the plot with a particular ordering; however, the interpretation won't be different with a different ordering of the categories.)

- A time-series plot is more appropriate than a bar plot when there is a time-based ordering to the categories, because usually you want to imply some sort of trend with time-ordered data. Therefore

do not use a bar plot for time trends, rather use a time-series plot.
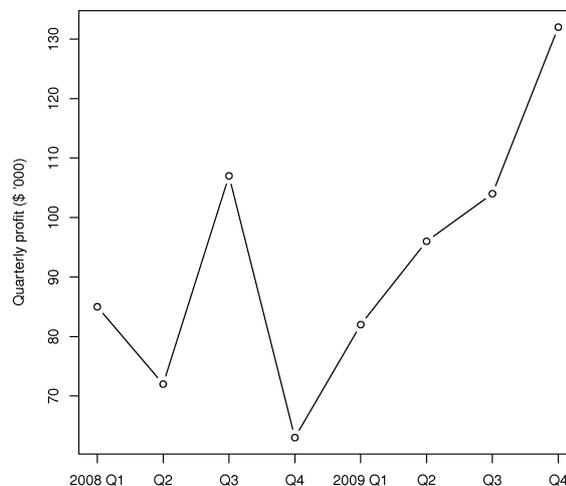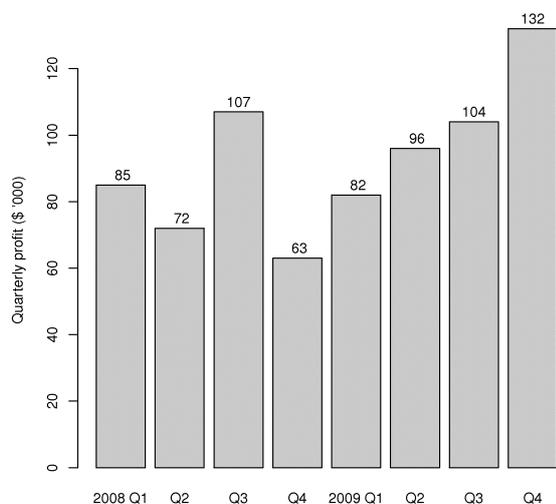
Use this R code to draw the figures:

```
                                    quarterly-profit-barplots.R
labels = c("2008 Q1", "Q2", "Q3", "Q4",
           "2009 Q1", "Q2", "Q3", "Q4")
profit = c(45, 32, 67, 23, 42, 56, 64, 92)+40

# Draw a bar-plot
bp = barplot(profit,
             names.arg=labels,
             axisnames=TRUE,
             ylab="Quarterly profit ($ '000)",
             border = TRUE)
text(bp, profit+3,
     labels=format(profit),
     xpd = TRUE,
     col = "black")

# Now rather use a line plot.
# Graph profit, but turn off axes
# and annotations
plot(profit, type="b", axes=TRUE,
     ann=FALSE, xaxt="n")

# Show the x-axis using our labels
axis(1, at=1:8, lab=labels)

# Plot title
title(ylab="Quarterly profit ($ '000)")
```

or this Python code:

```
                                 quarterly-profit-barplots.py
import pandas as pd
import matplotlib.pyplot as plt

labels = ["2008 Q1", "Q2", "Q3", "Q4", "2009 Q1", "Q2", "Q3", "Q4"]
profit = (
    pd.DataFrame(
        data=[45, 32, 67, 23, 42, 56, 64, 92],
        index=labels,
        columns=["Quarterly profit ($ '000)"]
```

(continues on next page)

```
    )
      + 40
)

# # Draw a bar-plot
ax = profit.plot.bar(color='lightgrey')
ax.set_ylabel("Quarterly profit ($ '000)")
plt.show()

# Now rather use a line plot.
ax = profit.plot.line(marker="o")
ax.set_ylabel("Quarterly profit ($ '000)")
plt.show()
```

- Bar plots can be wasteful as each data point is repeated several times:

  1. Left edge (line) of each bar

  2. Right edge (line) of each bar

  3. The height of the colour in the bar

  4. The number's position (up and down along the y-axis)

  5. The top edge of each bar, just below the number

  6. The number itself

  To this end, Tufte defines the data ink ratio as:

  $$\text{Data-ink ratio} = \frac{\text{total ink for data}}{\text{total ink for graphics}}$$
  $$= 1 - \text{proportion of ink that can be erased without loss of data information}$$

  The heuristic is to maximize this ratio as far as possible by using the ink (pixels) for only the data.

- Rather use a table than a bar plot for a handful of data points.

	Profit ($ '000)
Ontario	562
Manitoba	423
Quebec	231
Nova Scotia	181
Alberta	90
British Columbia	82
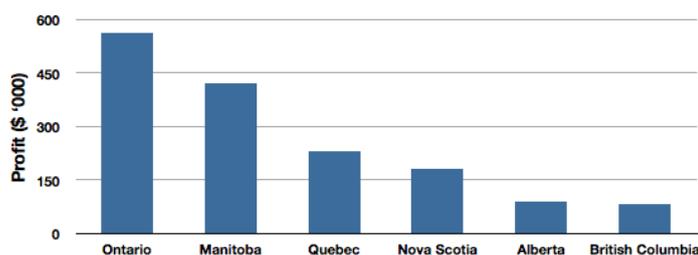
- Don't use cross-hatching, textures or unusual shading in the plots. This creates distracting visual vibrations.

Death Rates in Virginia
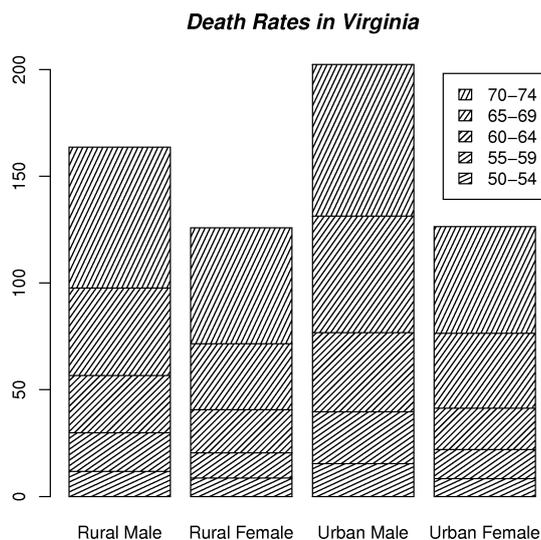
- Use horizontal bars if

  - there is some ordering to the categories (it is often easier to read the category labels from top-to-bottom), or

  - if the labels do not fit side-by-side: don't make the reader have to rotate the page to interpret the plot; rotate the plot for the reader.

- You can place the labels inside the bars.

- You should start the noncategory axis at zero: the bar's area shows the magnitude. Starting bars at a nonzero value distorts the meaning.

1.5 Box plots

Video for
this
section.

Box plots are an efficient summary of one variable (univariate chart), but can also be used effectively to compare variables that are in the same units of measurement.

The box plot shows the so-called *five-number summary* of a univariate data series:

1. Minimum sample value

2. 25th percentile[9] (1st quartile[10])

3. 50th percentile (median)

4. 75th percentile (3rd quartile)

5. Maximum sample value

The 25th percentile is the value below which 25% of the observations in the sample are found. The distance from the 3rd to the 1st quartile is also known as the interquartile range (IQR) and represents the data's spread, similar to the standard deviation.

---

[9] https://en.wikipedia.org/wiki/Percentile
[10] https://en.wikipedia.org/wiki/Quartile

The following data are thickness measurements of 2-by-6 boards (2-by-6 refers for the thickness and depth of a wooden board), taken at six locations around the edge. Here is a sample of the measurements and a summary of the first 100 boards (code in R and Python respectively):

board-thickness-boxplot.R
```R
all.boards = read.csv("http://openmv.net/file/six-point-board-thickness.csv")
boards = all.boards[1:100, 2:7]

# Look at the start and end of the data
# Examine and summarize your data before
# doing anything else
head(boards)
tail(boards)

summary(boards)

boxplot(boards)
```

board-thickness-boxplot.py
```python
import pandas as pd
import matplotlib.pyplot as plt

all_boards = pd.read_csv("http://openmv.net/file/six-point-board-thickness.csv")
boards = all_boards.iloc[0:100, 1:7]

# Look at the start and end of the data
# Examine and summarize your data before
# doing anything else
boards.head()
boards.tail()
boards.describe()
ax = boards.plot.box(fontsize=16)
plt.show()
```

The following box plot is a graphical summary of these numbers.
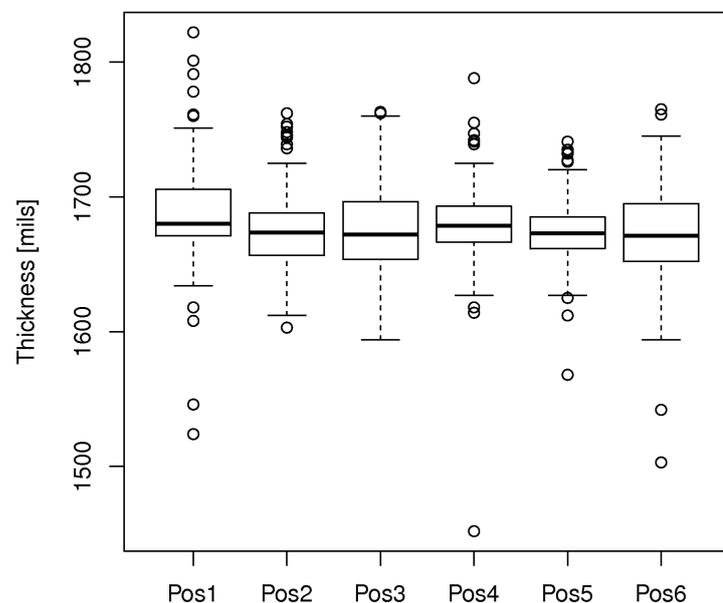
A box plot is great for comparisons. In this figure we see how the thickness at position 1 is greater than at the other positions. It is also the position with high variability, indicating that something about the saw blade at that position is not what it should be. The median is also not balanced between the two quantiles for this box plot, when compared to the others.

Some variations for the box plot are possible:

- Show outliers as dots, where an outlier is most commonly defined as any point 1.5 IQR distance units away from the box. The box's upper bound is at the 25th percentile, and the boxes lower bound is at the 75th percentile.

- The whiskers on the plots are drawn *at most* 1.5 IQR distance units away from the box, however, if the whisker is to be drawn beyond the bound of the data vector, then it is redrawn at the edge of the data instead (i.e. it is clamped, to avoid it exceeding).

- Use the mean instead of the median [*not too common*].

- Use the 2% and 98% percentiles rather than the upper and lower hinge values.

Example

In a final exam for a particular course at McMaster University there was an open-ended question. These data values are the grades[11] achieved for the answer to that question, broken down by whether the student used a systematic method, or not. No grades were given for using a systematic method; grades were awarded only for answering the question.

A systematic method is any method that assists the student with problem solving. For example, a strategry could be to: define the problem, identify knowns/unknowns and assumptions, explore alternatives, plan a strategy, implement the strategy and then check the solution.

Draw two box plots next to each other that compare the grades of students who did, or did not use a problem solving strategy. Comment on any features you notice in the comparison.

Answer

Several points are apparent in the box plot:
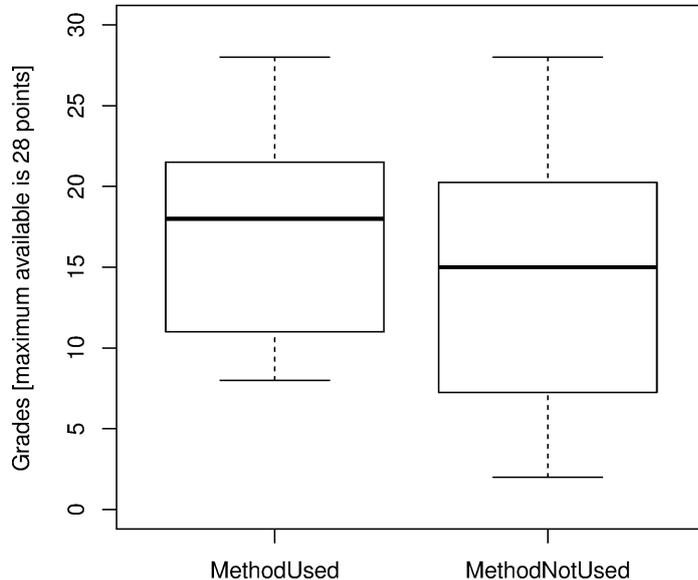
- students in either category achieved the highest grade possible

- the spread (interquartile distance) when using the problem solving method is smaller

- both box plots show a skew to the lower left tail (compare the median to the first and third quartiles)

- we will use a *confidence interval* (page 70) in a later chapter to judge whether this difference is statistically significant or not.

---

[11] http://openmv.net/info/systematic-method

More readings

You can read more about box plots in the paper by Hadley Wickham and Lisa Stryjewsk[12]. It summarizes variations of this plot, such as the violin plot, and two-dimensional versions of it. It is a power summary plot that has been around since 1970.

1.6 Relational graphs: scatter plots

Video for this section.

This is a plot many people are comfortable using. It helps you understand the relationship between two variables - a bivariate plot - as opposed to the previous charts that are univariate. A scatter plot is a collection of points shown inside a box formed by two axes at 90 degrees to each other. The marker's position is located at the intersection of the values shown on the horizontal (x) axis and vertical (y) axis.

The unspoken intention of a scatter plot is usually to ask the reader to draw a causal relationship between the two variables. However, not all scatter plots actually show causal phenomena, as the figure below tries to convince you:
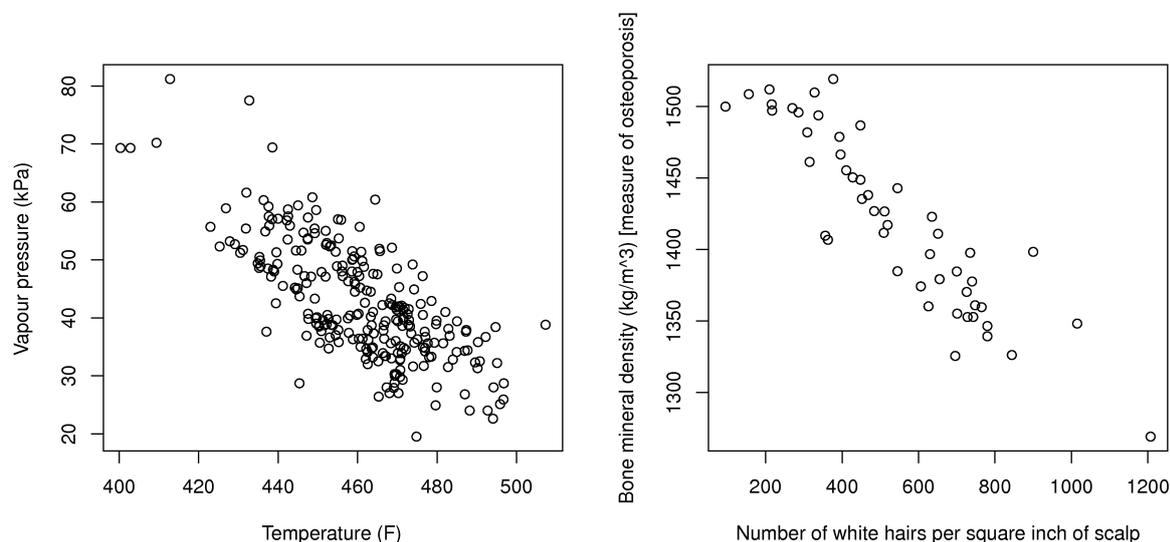
This source code generates similar, but not identical, figures to those shows here in the text.

```
                                    scatter-plot-example.R
# Plot of temperature vs vapour pressure
data_file = "http://openmv.net/file/distillation-tower.csv"
distillation = read.csv(data_file)

plot(distillation$Temp9,
     distillation$VapourPressure,
     xlab="Temperature (F)",
     ylab="Vapour pressure (kPa)")

# Plot of white hairs vs BMD
#    Osteoporosis (fake) data: number of white
#    hairs per square inch vs bone mineral
#    density (measurement of osteoporosis)
#    vs kg/m^3 (1500 kg/m3 is typical)
N = 50
white.hairs = round(rnorm(N,
                    mean=500,
```

(continues on next page)

---

[12] https://vita.had.co.nz/papers/boxplots.pdf

```
                    sd=150))
bone.mineral.density = -0.25 * white.hairs + 1550 + rnorm(N, mean=0, sd=25)

plot(white.hairs, bone.mineral.density,
     xlab = "Number of white hairs per square inch of scalp",
     ylab = "Bone mineral density (kg/m^3) [measure of osteoporosis]")
```

The equivalent code in Python:

```
                        scatter-plot-example.py
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Plot of temperature vs vapour pressure
data_file = "http://openmv.net/file/distillation-tower.csv"
distillation = pd.read_csv(data_file)
ax = distillation.plot.scatter(x="Temp9",
                               y="VapourPressure",
                               marker="o", s=20)
ax.set_xlabel("Temperature (F)")
ax.set_ylabel("Vapour pressure (kPa)")
plt.show()

# Plot of white hairs vs BMD
#    Osteoporosis (fake) data: number of white
#    hairs per square inch vs bone mineral
#    density (measurement of osteoporosis)
#    vs kg/m^3 (1500 kg/m3 is typical)
N = 50
white_hairs = np.random.normal(loc=500, scale=150, size=N)
bone_mineral_density = -0.25 * white_hairs + 1550 + np.random.normal(loc=0, scale=25, size=N)
fig2, ax2 = plt.subplots(nrows=1, ncols=1)
ax2.plot(white_hairs, bone_mineral_density, "o", ms=10)
ax2.set_xlabel("Number of white hairs per square inch of scalp")
ax2.set_ylabel("Bone mineral density (kg/m$^3$) [measure of osteoporosis]")
plt.show()
```

Strive for graphical excellence by doing the following:

- Make each axis as tight as possible.

- Avoid heavy grid lines.

- Use the least amount of ink.

- Do not distort the axes.

There is an unfounded fear that others won't understand your 2D scatter plot. Tufte (*Visual Display of Quantitative Information*, p 83) shows that there are no scatter plots in a sample (1974 to 1980) of U.S., German and British dailies, despite studies showing that 12-year-olds can interpret such plots: Japanese newspapers frequently use them.

You will see this in industrial settings as well. The next time you go into an industrial control room (or look carefull at some screens in online videos), try finding any scatter plots. The audience is not to blame: it is the producers of these charts who assume the audience is incapable of interpreting them.

Note: Assume that if you can understand the plot, so will your audience.

Further improvements can be made to your scatter plots. For example, extend the frames only as far as

your data:
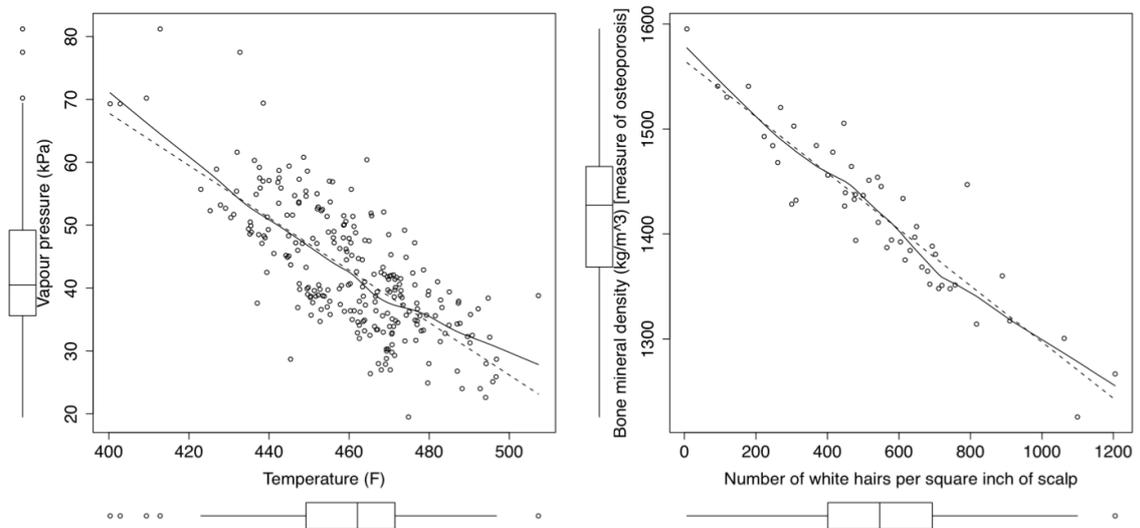
You can add box plots and histograms to the side of the axes to aide interpretation:
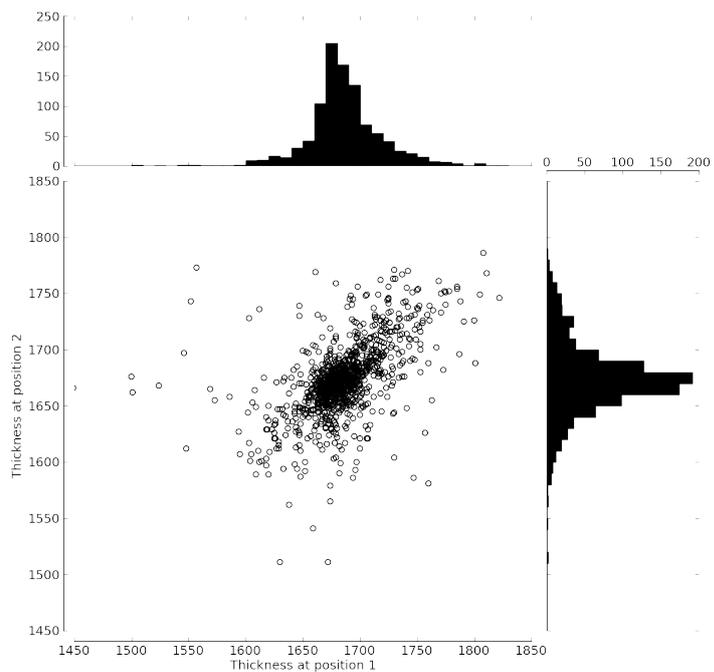
Add a third variable to the plot by adjusting the marker size, and add a fourth variable with the use of colour:
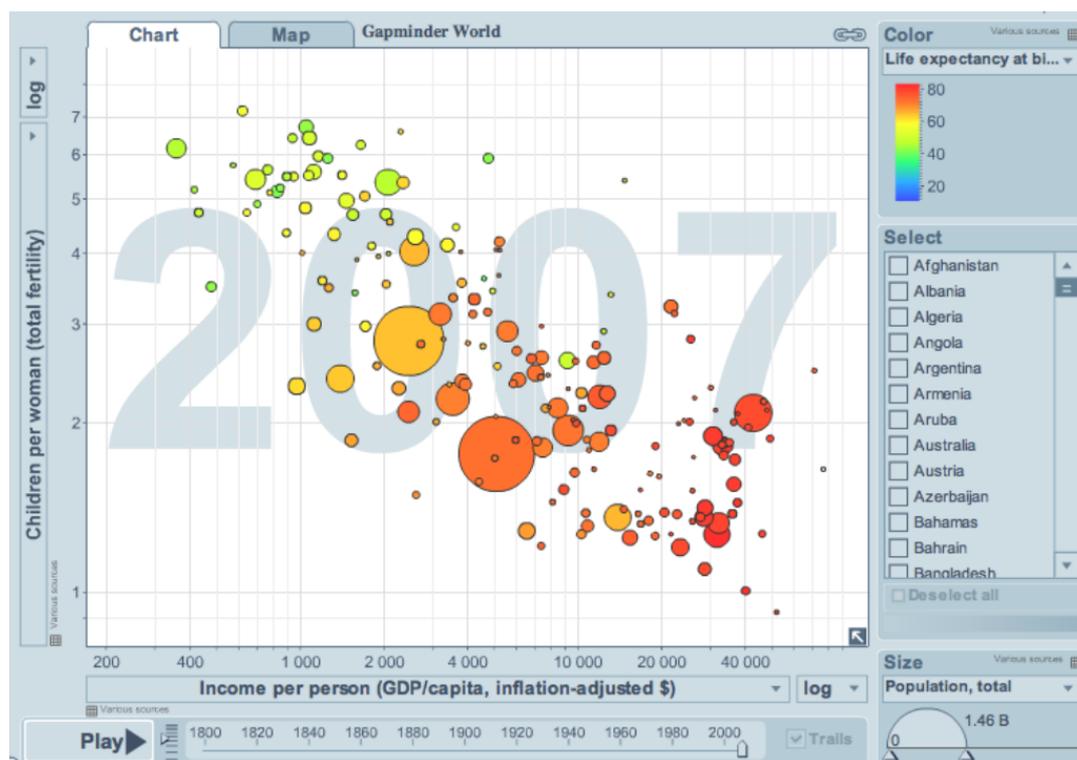
This example, from https://gapminder.org[13] , shows data until 2007 for:

1. income per person (*x*-axis);

2. against fertility (*y*-axis);

3. the size of each data point is proportional to the country's population;

4. the marker colour shows life expectancy at birth (years).

5. The GapMinder website allows you to "play" the graph over time, effectively adding a fifth dimension to the 2D plot.

So 5 dimensions in a 2D surface. A 6th dimension cab be added if using technology such as VR glasses, to create a 3rd dimension, to display another variable from the data set.

Use the hyperlink above to see how richer countries move towards lower fertility and higher income over time.

1.7 Tables as a form of data visualization

A data table, or a spreadsheet, is an efficient format for comparative data analysis on categorical objects. Usually, the items being compared are placed in a column, while the categorical objects are in the rows. The quantitative value is then placed at the intersection of the row and column, called the *cell*. The following examples demonstrate data tables.

This table compares monthly payments for buying or leasing various cars (categories). The first two columns are being compared; the other columns contain additional, secondary information.

---

[13] https://yint.org/gapminder-example

	Bank loan monthly payments	Monthly lease payment	Minimum downpayment for lease	Total interest paid over 48 months	Monthly insurance payment
Ford Fusion	552	395	0	2,529	180
Honda Civic	538	424	0	2,466	236
Mazda 3	506	478	1,000	2,318	251
Toyota Yaris	435	490	1,000	1,992	198
VW Golf	596	550	2,500	2,730	244

The next table compares defect types (number of defects) for different product grades (categories).

	Total defects	A	B	C	D	E
A4636	131	37	21	28		45
A2524	86	20	24	21	1	20
A3713	75	17	13	18		27
A4452	73	5	33	17		18
A4088	72	14	16	12	2	28
A2103	68	14	13	14	1	26
A2156	68	16	13	19	2	18
A3681	66	12	16	9	1	28
A1366	50	11	15	12		12
A2610	39	5	7	12		15
Total	**728**	**151**	**171**	**162**	**7**	**237**

This particular table raises more questions:

- Which defects cost us the most money?

- Which defects occur most frequently? The table does not contain any information about production rate. For example, if there were 1850 lots of grade A4636 (first row) produced, then defect A occurs at a rate of $37/1850 = 1/50$. And if 250 lots of grade A2610 (last row) were produced, then, again, defect A occurs at a rate of $1/50$. Redrawing the table on a production-rate basis would be useful if we are making changes to the process and want to target the most problematic defect.

- If we are comparing a type of defect over different grades, then we are now comparing down the table, instead of across the table. In this case, the fraction of defects for each grade would be a more useful quantity to display.

- If we are comparing defects within a grade, then we are comparing across the table. Here again, the fraction of each defect type, weighted according to the cost of that defect, would be more appropriate.

Three common pitfalls to avoid:

1. *Avoid using pie charts when tables will do.*

   Pie charts are tempting when we want to graphically break down a quantity into components. I have used them erroneously myself (here is an example on a website that I helped with: http://www.macc.mcmaster.ca/gradstudies.php). We won't go into details here, but I strongly suggest you read the convincing evidence of Stephen Few in: "Save the pies for dessert"[14]. The key problem is that the human eye cannot adequately decode angles; however, we have no problem with linear data.

2. *Avoid arbitrary ordering along the first column; usually, alphabetically or in time order is better.*

---

[14] https://www.perceptualedge.com/articles/08-21-07.pdf

Listing the car types alphabetically is trivial: instead, list them by some other third criterion of interest, perhaps minimum down payment required, typical lease duration, or total amount of interest paid on the loan. That way you get some extra context to the table for free.

3. *Avoid using excessive grid lines.*

Tabular data should avoid vertical grid lines, except when the columns are so close that mistakes will be made. The human eye will use the visual white space between the numbers to create its own columns.

	Total defects	A	B	C	D	E
A4636	131	37	21	28		45
A2524	86	20	24	21	1	20
A3713	75	17	13	18		27
A4452	73	5	33	17		18
A4088	72	14	16	12	2	28
A2103	68	14	13	14	1	26
A2156	68	16	13	19	2	18
A3681	66	12	16	9	1	28
A1366	50	11	15	12		12
A2610	39	5	7	12		15
Total	**728**	**151**	**171**	**162**	**7**	**237**

	Total defects	A	B	C	D	E
A4636	131	37	21	28		45
A2524	86	20	24	21	1	20
A3713	75	17	13	18		27
A4452	73	5	33	17		18
A4088	72	14	16	12	2	28
A2103	68	14	13	14	1	26
A2156	68	16	13	19	2	18
A3681	66	12	16	9	1	28
A1366	50	11	15	12		12
A2610	39	5	7	12		15
Total	**728**	**151**	**171**	**162**	**7**	**237**

To wrap up this section is a demonstration of tabular data in a different format, based on an idea of Tufte in *The Visual Display of Quantitative Information*, p. 158. Here we compare the corrosion resistance and roughness of a steel surface for two different types of coatings, A and B.

A layout that you expect to see in a standard engineering report:

Product	Corrosion resistance		Surface roughness	
	Coating A	Coating B	Coating A	Coating B
K135	0.30	0.22	30	42
K136	0.45	0.39	86	31
P271	0.22	0.24	24	73
P275	0.40	0.44	74	52
S561	0.56	0.36	70	75
S567	0.76	0.51	63	70

And the layout advocated by Tufte:
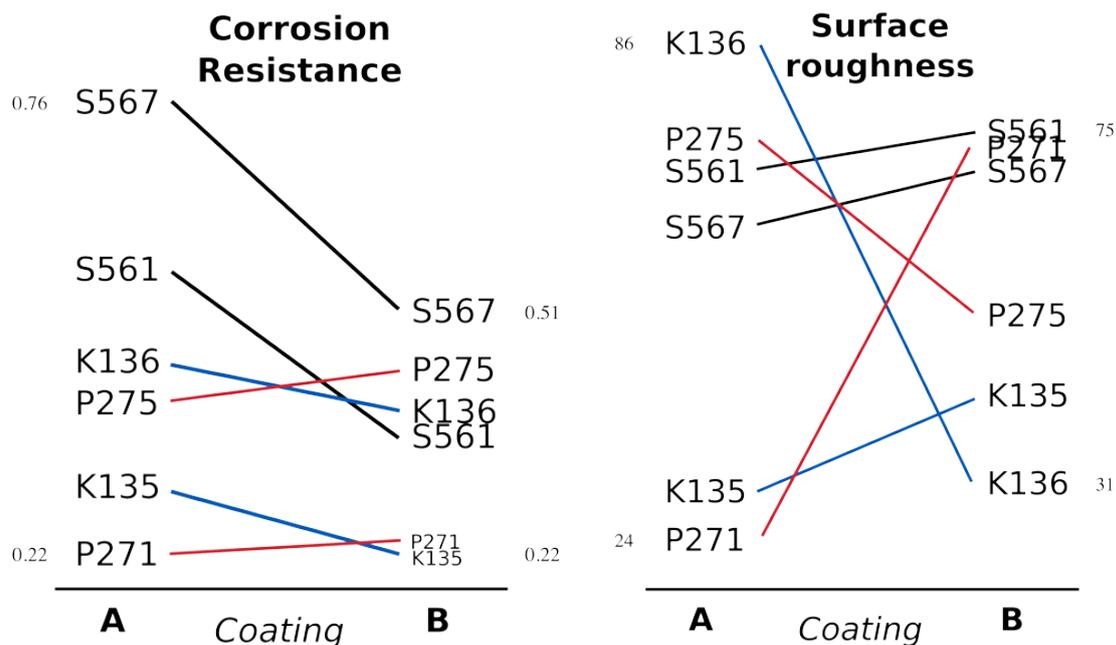
Note how the slopes carry the information about the effect of changing the coating type. The rearranged row ordering shows these changes as well. This idea is effective for two treatments but could be extended to three or four treatments by adding extra "columns". Only the extremes are numbered, but every point could be numbered if the values are also required by the readers.

1.8 Topics of aesthetics and style

We won't cover these topics, but *Tufte's books* (page 2) contain remarkable examples that discuss effective use of colour for good contrast, varying line widths, and graph layout (e.g. use more horizontal than vertical - an aspect ratio of about 1.4 to 2.0; and flow the graphics into the location in the text where discussed).

1.8.1 Data frames (axes)

Frames are the basic containers that surround the data and give context to our numbers. Here are some tips:

1. Use round numbers.

2. Generally, tighten the axes as much as possible, except . . .

3. When showing comparison plots, all axes must have the same minima and maxima.

1.8.2 Colour

Colour is very effective in all graphical charts. However, you must bear in mind that your readers might be colour-blind, or the document might be read from a grayscale printout, or viewed on an electronic device where colours are shown differently than you might intend.

Note also that a standard colour progression does *not* exist. We often see dark blues and purples representing low numbers and reds the higher numbers, with greens, yellows and orange in-between. There are several such colour schemes[15] - there isn't a universal standard. The only safest colour progression is the grayscale axis, ranging from black to white at each extreme: this satisfies both colour-blind readers and users of your grayscale printed output.

See the *section on scatter plots* (page 13) for an example of the effective use of colour.

1.9 General summary: revealing complex data graphically

There is no generic advice that applies in every instance. These tips are useful, though, in most cases:

- If the question you want answered is causality, then show causality (the most effective way is with bivariate scatter plots). If trying to answer a question with alternatives, show comparisons (with tiles of plots or a simple table).

- Words and graphics belong together. Add labels to plots for outliers, and explain interesting points. Add equations and even small summary tables on top of your plots. Remember that a graph should be like a paragraph of text, not necessarily just a graphical display of numbers that you discuss later on.

- Avoid obscure coding on the graph. Don't label points as "A", "B", "C", ... and then put a legend: "A: grade TK133", "B: grade RT231", "C: grade TK134". Just put the labels directly on the plot.

- Do not assume your audience is ignorant and won't understand a complex plot. Conversely, don't try to enliven a plot with decorations and unnecessary graphics (flip through a copy of almost any weekly news magazine for examples of this sort of embellishment). As Tufte mentions more than once in his books, "*If the statistics are boring, then you've got the wrong numbers.*". The graph should stand on its own.

- When the graphics involve money and time, make sure you adjust the money for inflation.

- Maximize the data-ink ratio = (ink for data) / (total ink for graphics). Maximizing this ratio, within reason, means you should (a) eliminate nondata ink and (b) erase redundant data-ink.

- Maximize data density. Humans can interpret data displays[16] of around 100 data points per centimeter (250 data points per linear inch) and around 10000 per square centimeter (60000 data points per square inch).

---

[15] https://en.wikipedia.org/wiki/Color_scheme
[16] https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0001OR

1.10 Exercises

Question 1

The data shown here are the number of visits to a university website for a particular statistics course. There are 90 students in the class.
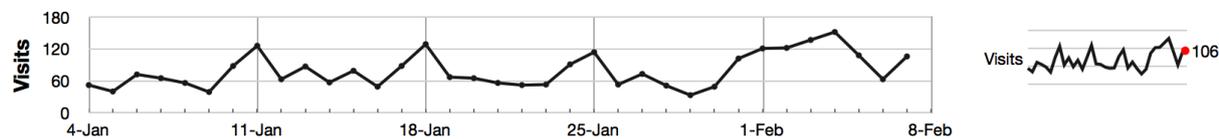
1. What are the names (type) of the 2 plots shown?

2. List any 2 interesting features in these data.

Solution

1. The plots are a time-series plot and a sparkline. The sparkline shows exactly the same data, just a more compact form (without the labelling on the axes).

2. Features shown in the data are:

   - A noticeable weekly cycle; probably assignments are due the next day!

   - A sustained, high level of traffic in the first week February - maybe a midterm test.

   - Some days have more than 90 visits, indicating that students visit the site more than once per day, or due to external visitors to the site.

Question 2

What are the names of the axes on a bar plot?

Solution

The category axis and value axis.

Question 3

Which types of features can can the human eye easily pick out of a time series plot?

Solution

Features such as sinusoids, spikes, gaps (missing values), upward and downward trends are quickly picked out by the human eye, even in a poorly drawn plot.

Question 4

Why is the principle of minimizing "data ink" so important in an effective visualization? Give an scientific or engineering example of why this important.

Solution

It reduces the time or work to interpret that plot, by eliminating elements that are non-essential to the plot's interpretation. Situations which are time or safety critical are examples, for example in an operator control room, or medical facility (operating room).

Question 5

Describe what the main difference(s) between a bar chart and a histogram are.

Solution

The solution is taken directly from:
https://www.forbes.com/sites/naomirobbins/2012/01/04/a-histogram-is-not-a-bar-chart/

- Histograms are used to show distributions of variables while bar charts are used to compare variables.

- Histograms plot quantitative data with ranges of the data grouped into bins or intervals while bar charts plot categorical data.

- Bars can be reordered in bar charts but not in histograms.

- There are no spaces between the bars of a histogram since there are no gaps between the bins. An exception would occur if there were no values in a given bin but in that case the value is zero rather than a space. On the other hand, there are spaces between the variables of a bar chart.

- The bars of bar charts typically have the same width. The widths of the bars in a histogram need not be the same as long as the total area is one hundred percent if percents are used or the total count if counts are used. Therefore, values in bar charts are given by the length of the bar while values in histograms are given by areas.

Question 6

Write out a list of any features that can turn a plot into a poor visualization. Think carefully about plots you encountered in textbooks and scientific publications, or the lab reports you might have recently created for a university or college course.

Question 7

This question is an extension to visualizing more than 3 variables. Investigate on your own the term "*scatterplot matrix*", and draw one for the Food texture data set[17]. See the `car` library in R to create an effective scatterplot matrix with the `scatterplotMatrix` function. List some bullet-points that interpret the plot.

Solution
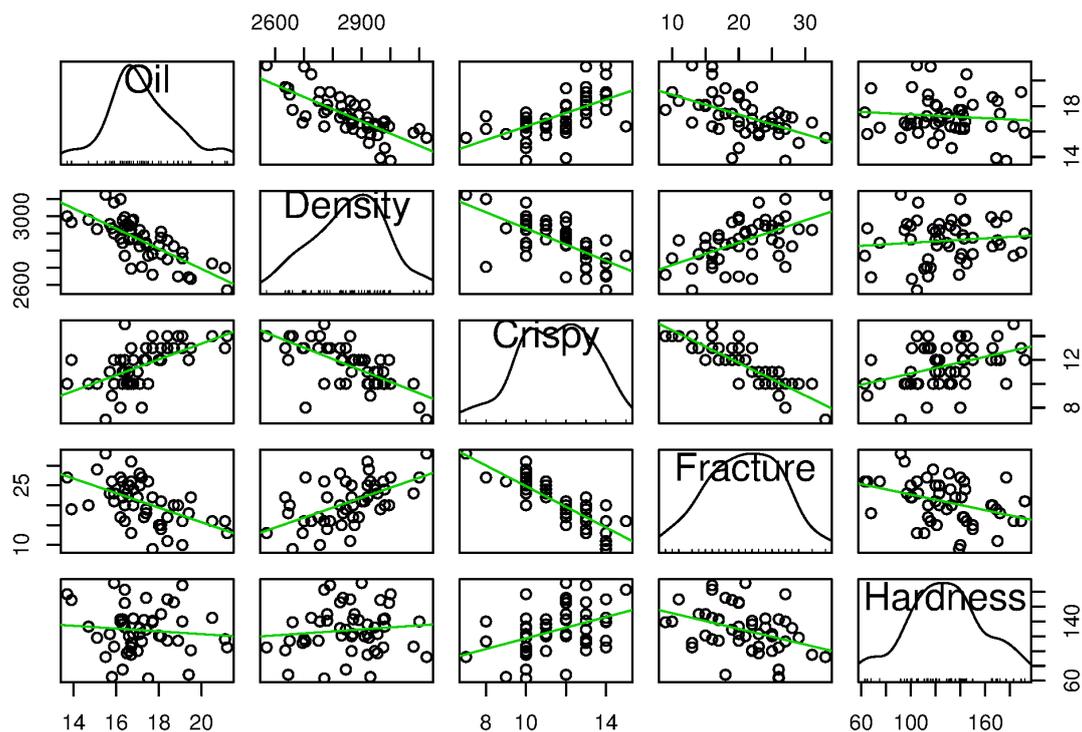

```
───────────────────────────── R code ─────────────────────────────
library(car)
data_file = 'http://openmv.net/file/food-texture.csv'
food <- read.csv(data_file)

# Hide the smoother and bounds
scatterplotMatrix(food[,2:6])
```

From this plot we see histograms of the 5 univariate distributions on the diagonal plots; the off-diagonal plots are the bivariate correlations between each combination of variable. The trend line (solid light green) shows the linear regression between the two variables. The lower diagonal part of the plot is a 90 degree rotation of the upper diagonal part. Some software packages will just draw either the upper or lower part.

From these plots we quickly gain an insight into the data:

- Most of the 5 variables have a normal-like distribution, except for `Crispy`, but notice the small notches on the middle histogram: they are equally spaced, indicating the variable is not continuous;

---

[17] http://openmv.net/info/food-texture

it is quantized[18]. The `Fracture` variable also displays this quantization.

- There is a strong negative correlation with oiliness and density: oilier pastries are less dense (to be expected).

- There is a positive correlation with oiliness and crispiness: oilier pastries are more crisp (to be expected).

- There is no relationship between the oiliness and hardness of the pastry.

- There is a negative correlation between density and crispiness (based on the prior relationship with `Oil`): less dense pastries (e.g. more air in them) and crispier.

- There is a positive correlation between `Density` and `Fracture`. As described in the dataset file, `Fracture` is the angle by which the pastry can be bent, before it breaks; more dense pastries have a higher fracture angle.

- Similarly, a very strong negative correlation between `Crispy` and `Fracture`, indicating the expected effect that very crispy pastries have a low fracture angle.

- The pastry's hardness seems to be uncorrelated to all the other 4 variables.

Question 8

Using the Website traffic data set[19]

1. Create a chart that shows the *variability* in website traffic for each day of the week.

2. Use the same data set to describe any time-based trends that are apparent.

Solution

1. A suitable chart for displaying variability on a per-day basis is the boxplot, one box for each day of the week. This allows you to see *between-day* variation when comparing the boxes side by side, and get an impression of the *variability within* each variable, by examining how the box's horizontal lines are spread out (25th, 50th and 75th percentiles).

2. A box plot is an effective way to summarize and compare the data for each day of the week.

```R
                                  R code
web = read.csv('http://openmv.net/file/website-traffic.csv')

# Re-order the factors in this order
day.names = c("Saturday", "Sunday", "Monday", "Tuesday", "Wednesday","Thursday", "Friday" )
days = factor(web$DayOfWeek, level=day.names)
boxplot(web$Visits ~ days)
```

---

[18] https://en.wikipedia.org/wiki/Quantization_(signal_processing)

[19] http://openmv.net/info/website-traffic
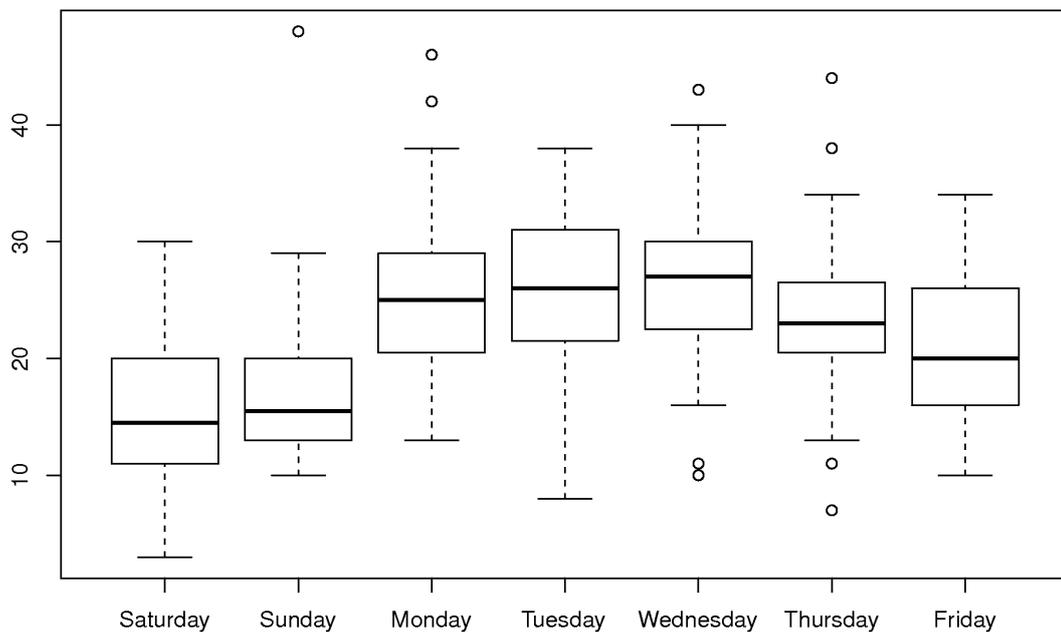
The box plot shows:

- Much less website traffic on Saturdays and Sundays, especially Sunday which has less spread than Saturday.

- Visits increase during the weekday, peaking on Wednesday and then dropping down by Friday.

- All week days seem to have about the same level of spread, except Friday, which is more variable.

- This is a website of academic interest, so these trends are expected.

3. A time-series plot of the data shows increased visits in September and October, and declining visits in November and December. This coincides with the phases of the academic term. A plot of the total number of visits within each month will show this effect clearly. The lowest number of visits were recorded in late June and July.

web.visits
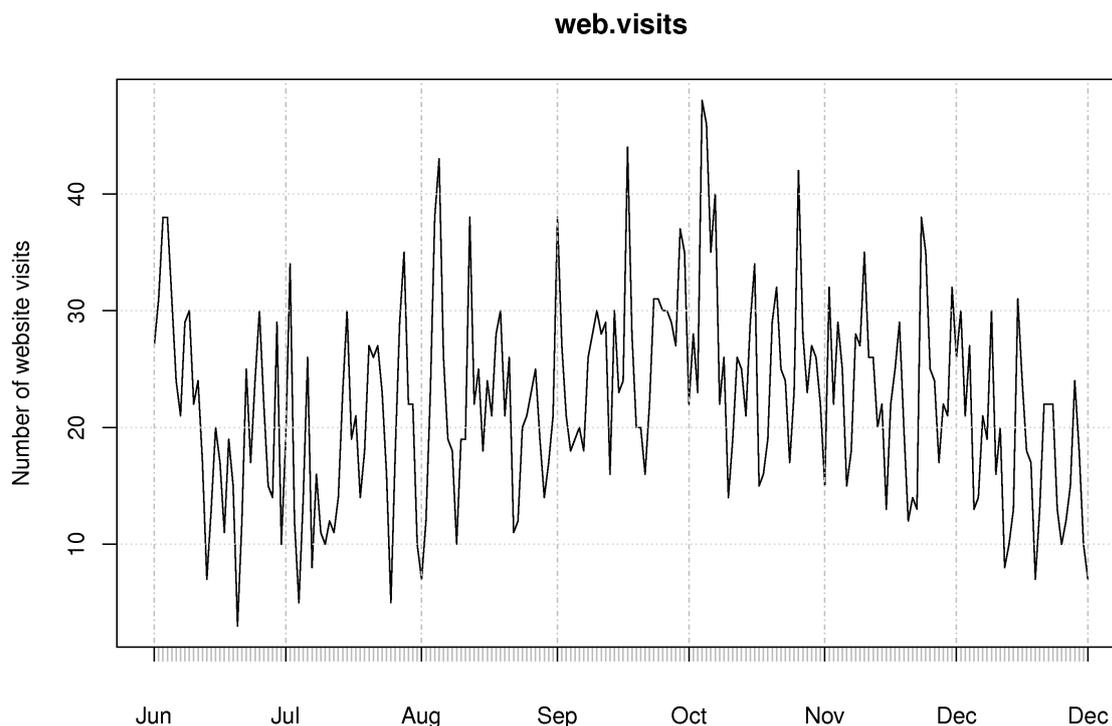
The best way to draw the time-series plot is to use proper time-based labelling on the x-axis, but we won't cover that topic here. If you are interested, read up about the `xts` package (see the R tutorial[20]) and it's plot command. See how it is used in the code below:

```
                                    R.code
web = read.csv('http://openmv.net/file/website-traffic.csv')

layout(matrix(c(1,2), 1, 2))
plot(web$Visits, type="o")

# A better plot using the xts library
library(xts)
date.order = as.Date(web$MonthDay, format=" %B %d")
web.visits = xts(web$Visits, order.by=date.order)
plot(web.visits, major.format="%b")
```

Question 9

Load the room temperature[21] dataset into R, Python or MATLAB, or whichever software tool you prefer to plot with.

1. Plot the 4 trajectories, `FrontLeft, FrontRight, BackLeft` and `BackRight` on the same plot.

2. Comment on any features you observe in your plot.

3. Be specific and describe how sparklines of these same data would improve the message the data is showing.

---

[20] https://learnche.org/4C3/Software_tutorial

[21] http://openmv.net/info/room-temperature

            Chapter 1. Visualizing Process Data

Solution

1. You could use the following code to plot the data:

```
                                       R code
data_file = 'http://openmv.net/file/room-temperature.csv'
roomtemp <- read.csv(data_file)
summary(roomtemp)
ylim = c(290, 300)

plot(roomtemp$FrontLeft,
        type='l',
        col="blue",
        ylim=c(290, 300),
        xlab="Sequence order",
        ylab="Room temperature [K]")
lines(roomtemp$FrontRight,
        type='b',
        pch='o',
        col="blue")
lines(roomtemp$BackLeft,
        type='l',
        col="black")
lines(roomtemp$BackRight,
        type='b',
        pch='o',
        col="black")

legend(20, 300,
        legend=c("Front left",
                 "Front right",
                 "Back left",
                 "Back right"),
        col=c("blue", "blue",
              "black", "black"),
        lwd=2,
        pch=c(NA, "o", NA, "o"))
```
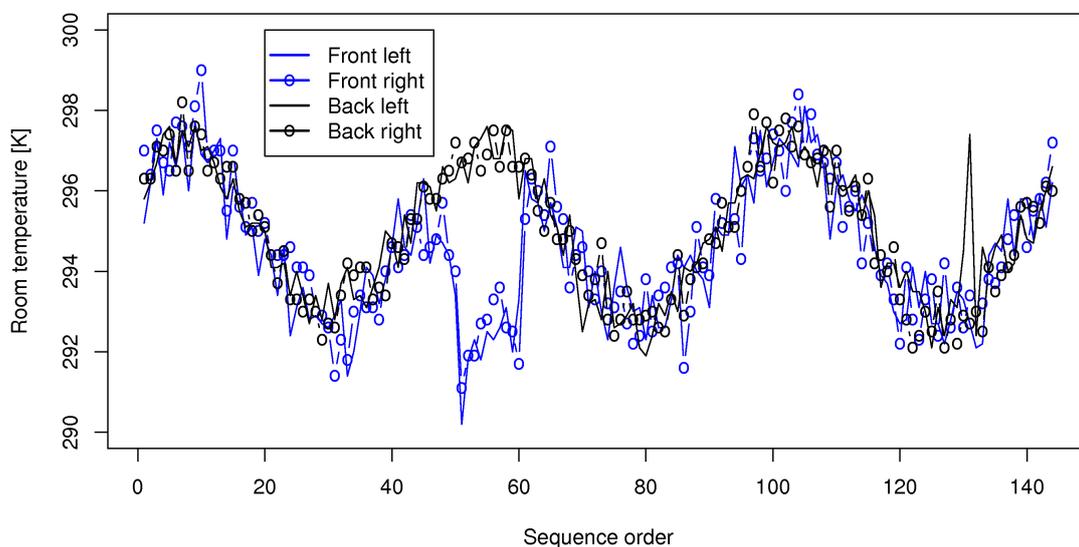

A sequence plot of the data is good enough, though a time-based plot is better.

2. • Oscillations, with a period of roughly 48 to 50 samples (corresponds to 24 hours) shows a daily cycle in the temperature.

   • All 4 temperatures are correlated (move together).

   • There is a break in the correlation around samples 50 to 60 on the front temperatures (maybe a door or window was left open?). Notice that the oscillatory trend still continues within the offset

region - just shifted lower.

- A spike up in the room's back left temperature, around sample 135.

3. The above plot was requested to be on one axis, which leads to some clutter in the presentation. Sparklines show each trajectory on their own axis, so it is less cluttered, but the same features would still be observed when the 4 tiny plots are stacked one on top of each other.
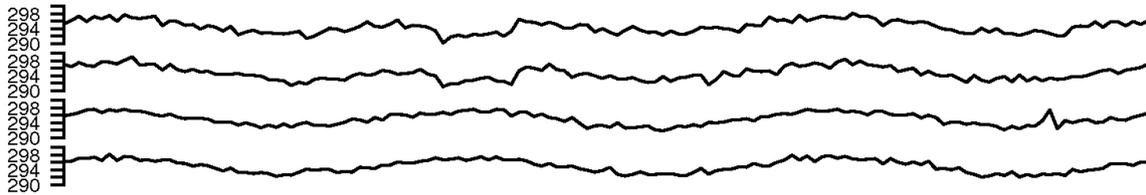
If you looked around for how to generate sparklines in R you may have come across this website[22]. Notice in the top left corner that the `sparklines` function comes from the `YaleToolkit`, which is an add-on package to R. We show how to install packages in the tutorial[23]. Once installed, you can try out that `sparklines` function:

- First load the library: `library(YaleToolkit)`

- Then see the help for the function: `help(sparklines)` to see how to generate your sparklines

Question 10

Load the six point board thickness[24] dataset, available from datasets website.

1. Plot a boxplot of the first 100 rows of data to match the figure *in these notes* (page 9)

2. Explain why the thick center line in the box plot is not symmetrical with the outer edges of the box.

Solution

1. The following code will load the data, and plot a boxplot for the first 100 rows:

```R
data_file = 'http://openmv.net/file/six-point-board-thickness.csv'
boards <- read.csv(data_file)
summary(boards)

plot(boards[1:100,2], type='l')
plot(boards[1:100,5], type='l')
first100 <- boards[1:100, 2:7]

# Ignore the first date/time column: using only Pos1, Pos2, ... Pos6 columns
boxplot(first100, ylab="Thickness [mils]")
```

---

[22] https://cran.r-project.org/web/packages/YaleToolkit/
[23] https://learnche.org/4C3/Software_tutorial/Extending_R_with_packages
[24] http://openmv.net/info/six-point-board-thickness
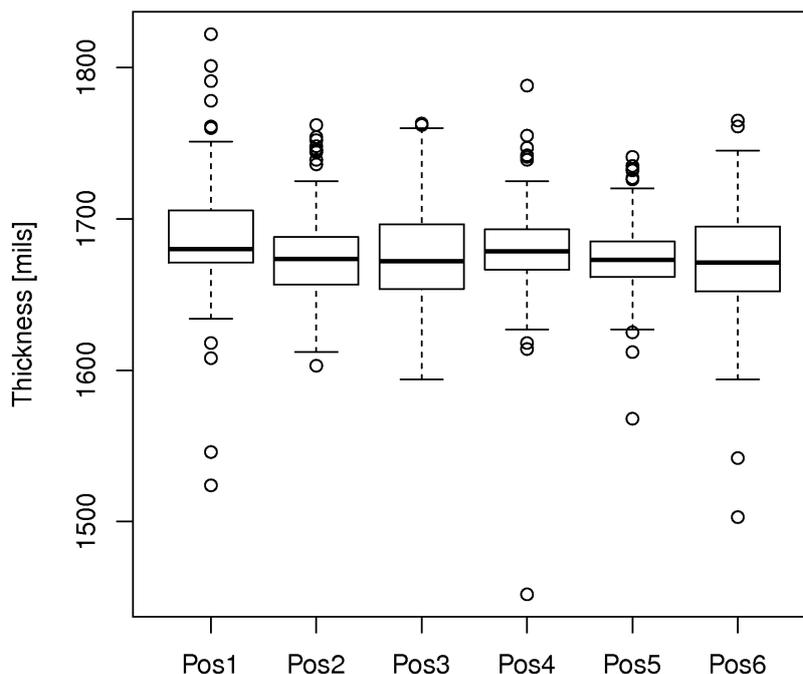
2. The thick center line on each boxplot is the median (50th percentile) of that variable. The top and bottom edges of the box are the 25th and 75th percentile, respectively. If the data are from a symmetric distribution, such as the t or normal distribution, then the median should be approximately centered with respect to those 2 percentiles. The fact that it is not, especially for position 1, indicates the data are *skewed* either to the left (median is closer to upper edge) or the the right (median closer to the lower edge).

Question 11

Read the short, clearly written article by Stephen Few on the pitfalls of pie charts: Save the pies for dessert, https://www.perceptualedge.com/articles/08-21-07.pdf[25]. The article presents an easy-to-read argument against pie charts that will hopefully convince you.

Here's a great example that proves his point[26] from the Canada Revenue Agency.

Question 12

Enrichment:

- Watch this 20 minute video[27] that shows how a 2-dimensional plot comes alive to show 5 dimensions of data. What are the 5 dimensions?

- A condensed version from this, 4 minute YouTube video[28] shows Hans Rosling giving a new perspective on the same data. This Economist article[29] has some interesting background on Dr. Rosling, as does this page, giving a selection of his work[30].

Video for this section.

---

[25] https://www.perceptualedge.com/articles/08-21-07.pdf

[26] https://www.canada.ca/en/revenue-agency/corporate/about-canada-revenue-agency-cra/individual-income-tax-return-statistics.html

[27] https://www.ted.com/talks/hans_rosling_the_best_stats_you_ve_ever_seen

[28] https://www.youtube.com/watch?v=jbkSRLYSojo

[29] https://www.economist.com/technology-quarterly/2010/12/11/making-data-dance

[30] https://www.economist.com/babbage/2010/12/09/hans-roslings-greatest-hits

2.1 Univariate data analysis in context

This section gives a starting idea to the general area of data analysis. We cover concepts from univariate data analysis shown in the pictorial outline below. This section is only a *review of these concepts* for one single variable. If you have more than one variable, you can repeat the analysis for each one. Later, in the *multivariate chapter* (page 315), we learn how to extract information from multiple variables at the same time.

Some introductory statistics textbooks, for more detailed background, are recommend further down.

2.1.1 Usage examples

The material in this section is used whenever you want to learn more about a single variable in your data set. For example:

- *Co-worker*: Here are the final output values, on a scale from 0 to 100%, from a batch system for the last 3 years (1256 data points).

  - What sort of distribution do the data have?

  - Yesterday our output value was less than 50%, what are the chances of that happening under typical conditions?

- *Yourself*: We have historical failure rate data for certain equipment in our factories. What is the probability that 3 of the same type of equipment will fail this year?

- *Manager*: We have 2 duplicate reactors. Does reactor 1 have better final product purity, on average, than reactor 2?

- *Colleague*: What does the 95% confidence interval for the density of our powder ingredient really mean?

2.1.2 What we will cover
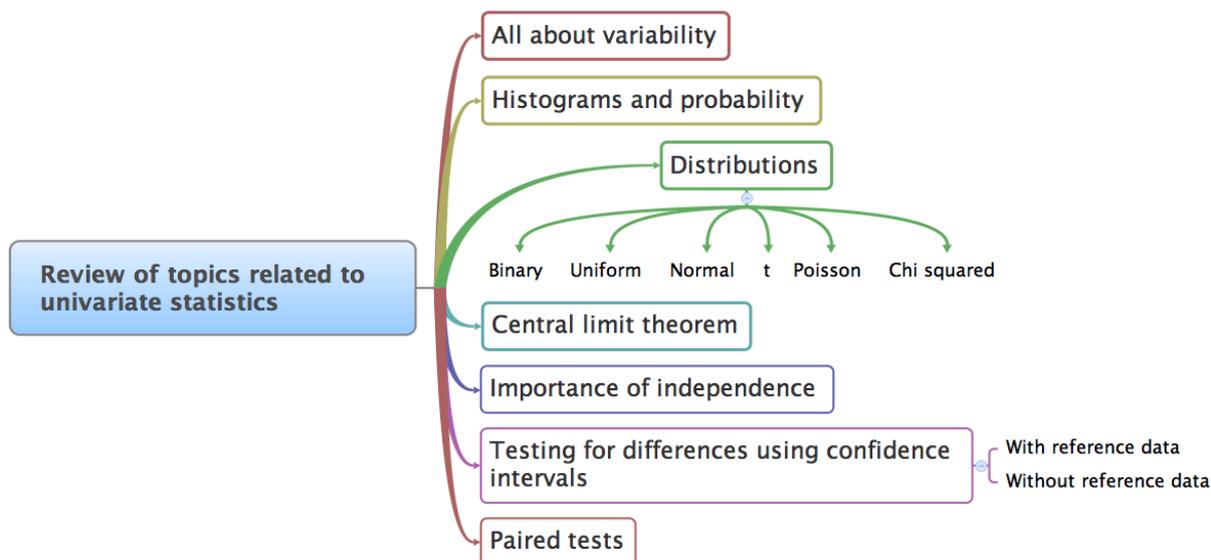
2.2 References and readings

Any standard statistics text book will cover the topics from this part of the book in much greater depth than these notes. Some that you might refer to:

1. **Recommended**: Box, Hunter and Hunter, *Statistics for Experimenters*, Chapter 2.

2. Hodges and Lehmann, *Basic Concepts of Probability and Statistics*.

3. Hogg and Ledolter, *Engineering Statistics*.

4. Montgomery and Runger, *Applied Statistics and Probability for Engineers*.

2.3 What is variability?

Life is pretty boring without variability, and this book, and almost all the field of statistics would be unnecessary if things did not naturally vary.
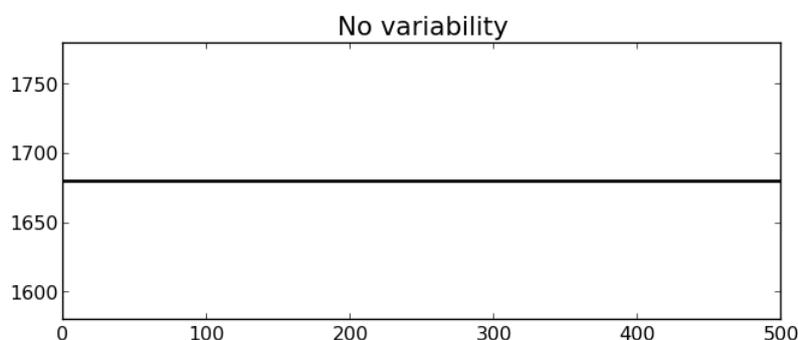
Fortunately, we have plenty of variability in the recorded data from our processes and systems:

- Raw material properties are not constant.

- Unknown sources, often called "*error*" (note that the word error in statistics does not have the usual negative connotation from English). These errors are all sources of variation which our imperfect knowledge of the process cannot account for.
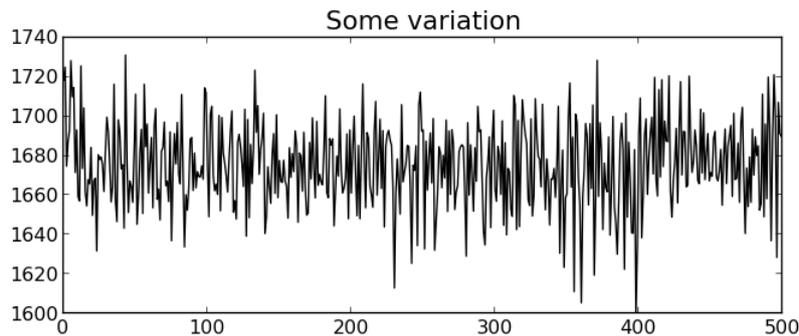
- Measurement and sampling variability: sensor drift, spikes, noise, recalibration shifts, errors in our sample analysis and laboratory equipment.
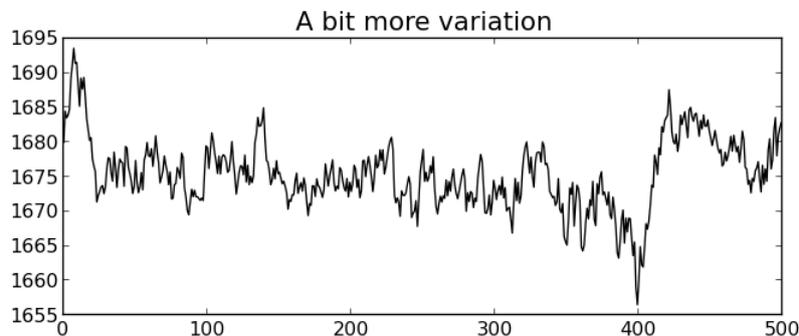
- Production disturbances:

  – external conditions change, such as ambient temperature, or humidity, and

  – pieces of plant equipment break down, wear out and are replaced.
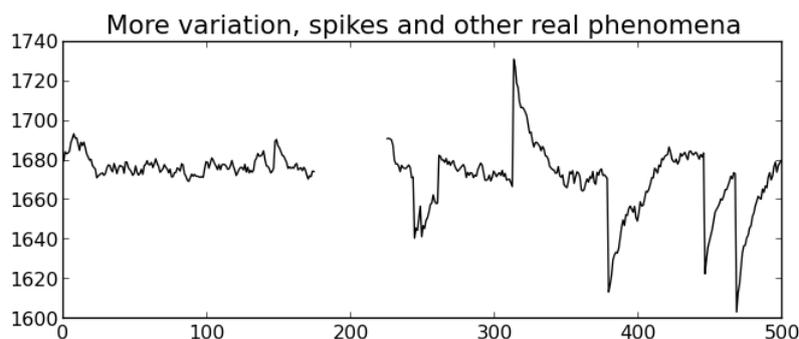
- Feedback control systems introduce variability in your process, in order to reduce variability in another part of the process. Think of what a *feedback control system* (page 33) does. See page 222 or page 879 of the freely available textbook by Dr. Thomas Marlin[31] for visual illustrations.

- Operating staff: introduce variability into a process in feedback manner (i.e. they react to process upsets) or in a feed-forward manner, for example, to preemptively act on the process to counteract a known disturbance. By doing so they introduce variability into a process.

---

[31] http://pc-textbook.mcmaster.ca

All this variability, although a good opportunity to keep many of use employed, comes at a price as described next.

2.3.1 The high cost of variability in your final product

Assertion

> Customers expect both uniformity and low cost when they buy your product. Variability defeats both objectives.

Three broad outcomes are possible when you sell a highly variable product:

1. The customer may be totally unable to use your product for the intended purpose. Imagine a food ingredient such as fresh milk, or a polymer with viscosity that is too high, or a motor oil with unsuitable properties that causes engine failure.

2. Your product leads to poor performance. The user must compensate for the poor properties through additional cost: more energy will be required to work with a polymer whose melting point is higher than expected, longer reaction times will be required if the catalyst purity is not at specification.

3. Your brand is diminished: your products, even though acceptable will be considered with suspicion in the future.

   An extreme example was the food poisoning and deaths that occurred due to the listeriosis outbreak at Maple Leaf Foods, Canada in 2008. The bacterial count in food products is always non-zero, however the established tolerance limits were exceeded during this outbreak.

   Another example was the inadvertent acceleration that occurred in some Toyota car models in 2010. It is still uncertain whether this was manufacturer error or driver error.

In addition to the risk of decreasing your market share (see the above 3 points), variability in your product also has these costs:

1. Inspection costs: to mitigate the above risks you must inspect your product before you ship it to your customers. It is prohibitively expensive and inefficient to test every product (known as "*inspecting quality into your product*"). A production line with low variability on the other hand, requires less inspection of every product.

   The pharmaceutical industry is well known to be inefficient in this respect, with terms such as "100% inspection" and even "200% inspection". Furthermore, some types of inspection are destructive, and therefore 100% inspection is not feasible.

2. Off-specification products: must be reworked, disposed of, or sold at a loss or much lower profit. These costs are ultimately passed onto your customers, costing you money.

Note: the above discussion assumes that you are able to quantify product quality with one or more univariate quality metrics and that these metrics are independent of each other. Quality is almost always a multivariate attribute of the product. We will *discuss the use of multivariate methods* (page 315) to judge product quality later.

2.3.2 The high cost of variability in your raw materials

Turning the above discussion around, with you on the receiving end of a highly variable raw material:

- If you do not implement any sort of process control system, then any variability in these raw materials that you receive and process is manifest as variability in your final product. This usually shows up in proportion: higher variability in the inputs results in higher variability in the product quality.
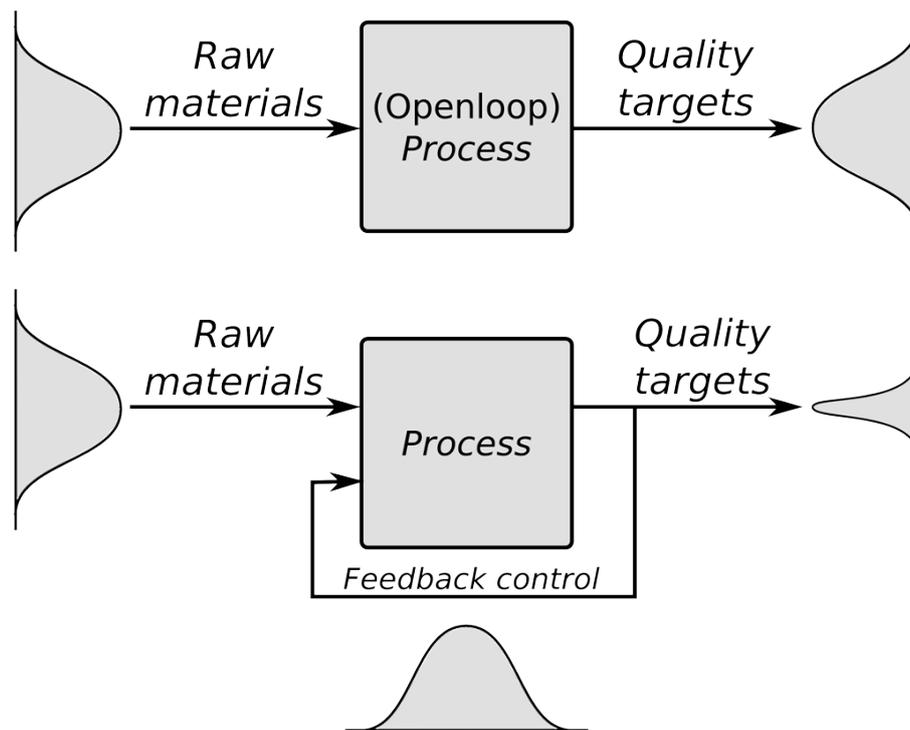
- Even if you do take feedback or feed-forward corrective control: you have to incur additional cost, since you have to process materials that are not to specification: this will require energy and/or time, reducing your profit due to the supplier's raw material variability.

  Note: Feedback control around a given set point can be seen as *introducing* additional variation into a process to counteract other sources of variation (called *disturbances* in the process control lingo). This is done with the hope of reducing the output variability.

2.3.3 Dealing with variability

So, how do we make progress despite this variability? This whole book, and all of statistical data analysis, is about variability:

- in the *data visualization section* (page 1) we gave some hints how to plot graphics that **show the variability** in our process clearly

- in this chapter we learn how to **quantify variability** and then **compare variability**

- later we consider how to *construct monitoring charts* (page 109) to **track variability**

- in the section on *least squares modelling* (page 151) we learn how **variation in one variable might affect another variable**

- with *designed experiments* (page 231) we intentionally **introduce variation** into our process to learn more about the process (e.g. so that we can optimize our process for improved profitability); and

- and in the *latent variable modelling* (page 315) section we learn how to deal with **multiple variables**, simultaneously extracting information from the data to understand how variability affects the process.

2.4 Histograms and probability distributions

The *previous section* (page 30) has hopefully convinced you that variation in a process is inevitable. This section aims to show how we can visualize and quantify any variability in a recorded vector of data.

A histogram is a summary of the variation in a measured variable. It shows the *number* of samples that occur in a *category*: this is called a **frequency distribution**. For example: number of children born, categorized against their birth gender: male or female.

Children born in Hamilton, April 2009, by gender
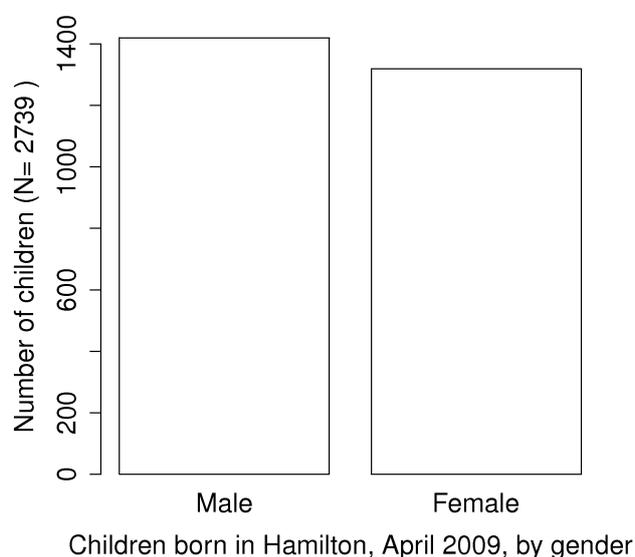
The raw data in the above example was a vector that consisted of 2739 text entries, with 1420 of them as `Male` and 1319 of them as `Female`. In this case `Female` and `Male` represent the two categories.

Histograms make sense for categorical variables, but a histogram can also be derived from a continuous variable. Here is an example showing the mass of cartons of 1 kg of flour. The continuous variable, mass, is divided into equal-size bins that cover the range of the available data. Notice how the packaging system has to overfill each carton so that the vast majority of packages weigh over 1 kg (what is the average package mass?). If the variability in the packaging system could be reduced - the spread of the data made narrower - then the histogram can be shifted to the left, thereby reducing overfill.

Histogram of data

---
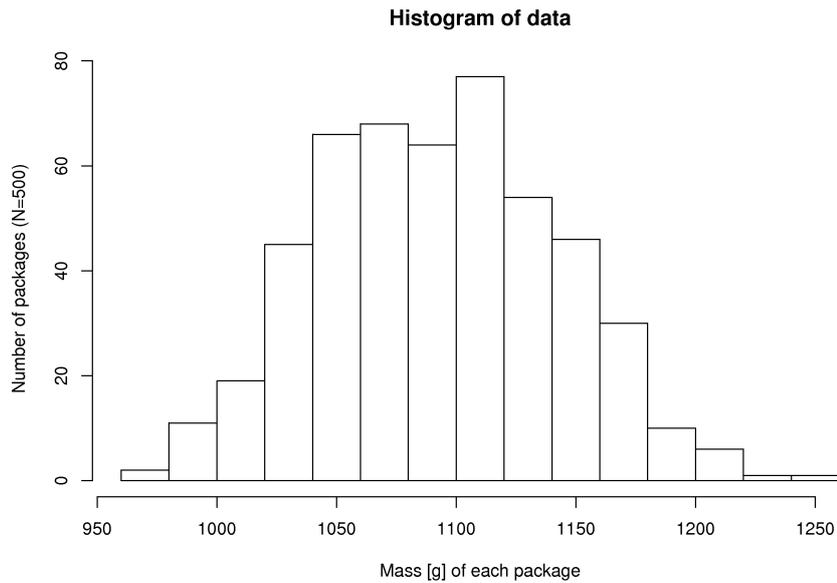
histogram-normal-distribution.R
```
# Create 500 normally distributed points
# with a mean of 1100 and standard deviation
# of 50 units.
data = rnorm(500, mean=1100, sd=50)
hist(data,
     xlab="Mass [g] of each package",
     ylab="Number of packages (N=500)")
```

---

histogram-normal-distribution.py
```python
# Create 500 normally distributed points
# with a mean of 1100 and standard deviation
# of 50 units.
import numpy as np
import matplotlib.pyplot as plt

N = 500
values = np.random.normal(loc=1100,
                          scale=50,
                          size=N)

plt.hist(values, color="white", bins=8)
plt.xlabel("Mass [g] of each package")
plt.ylabel("Number of packages (N={})".format(N))
plt.show()
```

Try creating a fictitious histogram for each of the following situations:

- The grades for a class for a really easy test.

- The numbers thrown from a 6-sided die.

- The annual income for people in your country.

- Analytical measurements taken in a laboratory, by the same person or computerized process.

In preparing the above histograms, what have you implicitly inferred about time-scales? These histograms show the long-term distribution (probabilities) of the system being considered. This is why *concepts of chance and random phenomena* can be use to described systems and processes. Probabilities can be used to describe our long-term expectations. Let us contrast some long-term and short-term expectations next:

2.4. Histograms and probability distributions   **35**

- The long-term sex ratio at birth 1.06:1 (boy:girl) is expected in Canada; but a newly pregnant mother would not know the sex.

- The long-term data from a process shows an 85% output yield from our batch reactor; but tomorrow it could be 59% and the day after that 86%.

- We know that a fair die has a 16.67% chance of showing a 4 when thrown, but we cannot predict the value of the next throw.

Even if we have complete mechanistic knowledge of our process, the concepts from probability and statistics are useful to summarize and communicate information about past behaviour, and the expected future behaviour.

Steps to creating a frequency distribution, illustrated with 4 examples, labelled A, B, C, and D.

1. Decide what you are measuring:

   A. acceptable or unacceptable metal appearance: yes/no

   B. number of defects on a metal sheet: none, low, medium, high

   C. yield from the batch reactor: somewhat continuous - quantized due to rounding to the closest integer

   D. daily ambient temperature, in Kelvin: continuous values

2. Decide on a resolution for the measurement axis:

   A. acceptable/unacceptable (1/0) code for the metal's appearance

   B. use a scale from 1 to 4 that grades the metal's appearance

   C. batch yield is measured in 1% increments, reported either as 78, 79, 80, 81%, *etc*.

   D. temperature is measured to a 0.05 K precision, but we can report the values in bins of 5K

3. Report the number of observations in the sample or population that fall within each bin (resolution step):

   A. number of metal pieces with appearance level "acceptable" and "unacceptable" are added up

   B. number of pieces with defect level 1, 2, 3, 4 are counted

   C. number of batches with yield inside each bin level are calculated

   D. number of temperature values inside each bin level are computed

4. Plot the number of observations in category as a bar plot. If you plot the number of observations divided by the total number of observations, N, then you are plotting the **relative frequency**.

A relative frequency, also called density, is sometimes preferred:

- we do not need to report the total number of observations, N

- it can be compared to other distributions

- if N is large enough, then the relative frequency histogram starts to resemble the population's distribution

- the area under the histogram is equal to 1, and related to probability
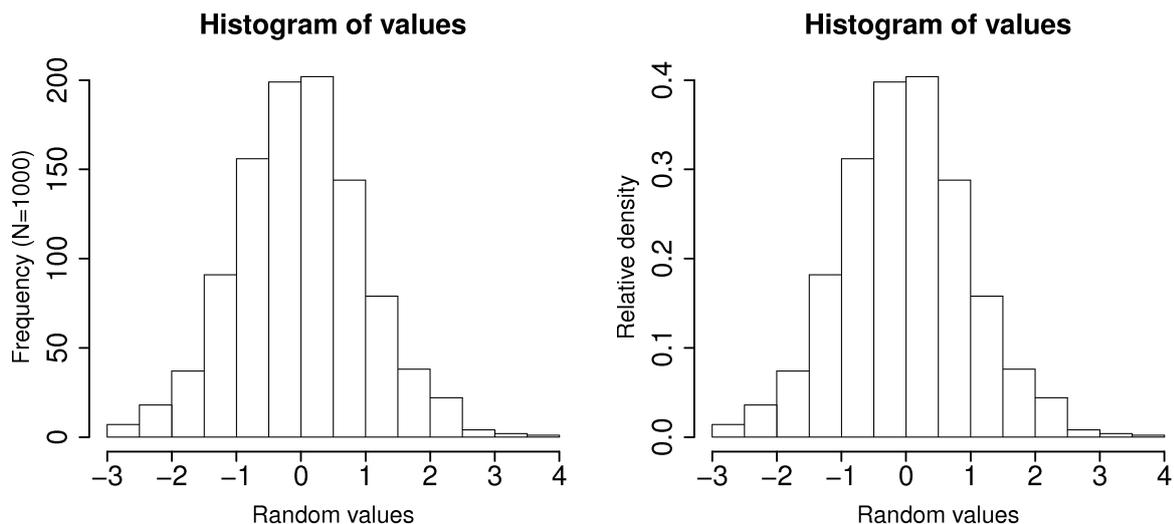
```
                                        ── histogram-area.R ──────────
# 1000 normally distributed values
N = 1000
values = rnorm(N)
hist(values, freq=TRUE,  xlab="Random values",
        cex.lab=1.5, cex.main=1.8, lwd=2,
        cex.sub=1.8, cex.axis=1.8,
        ylab=paste0("Frequency (N=",N,")"))
hist(values, freq=FALSE, xlab="Random values",
        cex.lab=1.5, cex.main=1.8, lwd=2,
        cex.sub=1.8, cex.axis=1.8,
        ylab="Relative density")

# Compare the two plots: only the y-axis
# changes but the general shape remains.
```

```
                                        ── histogram-area.py ─────────
# Create 1000 normally distributed points
# with mean of 0 and standard deviation of 1.
import numpy as np
import matplotlib.pyplot as plt

N = 1000
values = np.random.normal(loc=0,
                          scale=1,
                          size=N)

plt.subplot(1, 2, 1)
plt.hist(values, color="white")
plt.ylabel("Frequency (N={})".format(N))

plt.subplot(1, 2, 2)
plt.hist(values,
        color="white",
        # For older matplotlib versions
        normed=True,
        # Rather, use 'density' instead
        #density=True
        )
plt.ylabel("Relative density")

plt.tight_layout()
plt.show()
```

2.5 Some terminology

We review a couple of concepts that you should have seen in a prior statistical course or elsewhere. If unfamiliar, please type the word or concept in a search engine for more background.

Population

> A large collection of observations that *might* occur; a set of *potential* measurements. Some texts consider an infinite collection of observations, but a large number of observations is good enough.

Sample

> A collection of observations that have *actually* occurred; a set of *existing* measurements that we have recorded in some way, usually electronically.

Batch viscosity for the past 5 years
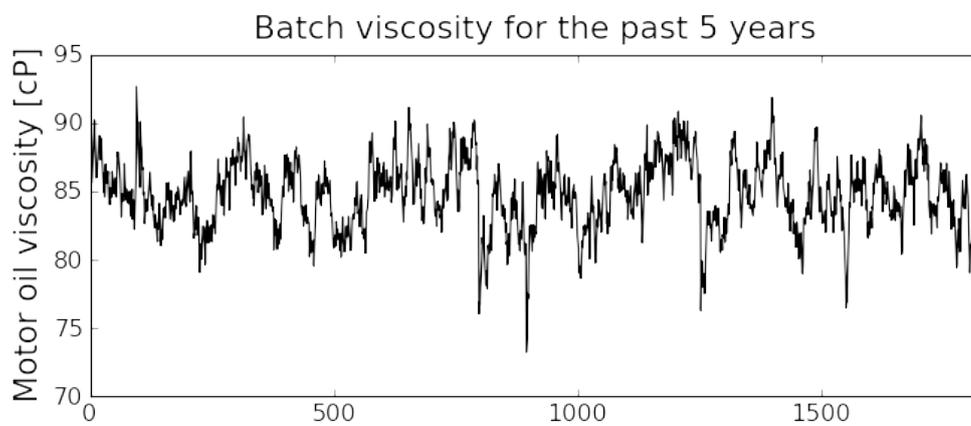
In engineering applications where we have plenty of data, we can characterize the population from all available data. The figure here shows the viscosity of a motor oil, from all batches produced in the last 5 years (about 1 batch per day). These 1825 data points, though technically a *sample* are an excellent surrogate for the *population* viscosity because they come from such a long duration. Once we have characterized these samples, future viscosity values will likely follow that same distribution, provided the process continues to operate in a similar manner.

Distribution

> Distributions are used to summarize, in a compact way, a much larger collection of a much larger collection of data points. Histograms, just discussed above, are one way of visualizing a distribution. We can also express distributions by a few numerical parameters. See below.

Probability

> The area under a plot of relative frequency distribution is equal to 1. Probability is then the fraction of the area under the frequency distribution curve (also called density curve).

> Superimpose a vertical line on your fictitious histograms you drew earlier to indicate:

- the probability of a test grades less than 80%;

- the probability that the number thrown from a 6-sided die is less than or equal to 2;

- the probability of someone's income exceeding $60000;

- the probability of the measurement exceeding a certain critical value.

Parameter

A parameter is a value that describes the population's **distribution** in some way. For example, the population mean.

Statistic

A statistic is an estimate of a population parameter.

Mean (location)

The mean, or average, is a measure of location of the distribution. For each measurement, x_i, in your sample

$$\text{population mean:} \quad \mathcal{E}\{x\} = \mu = \frac{1}{N}\sum x$$

$$\text{sample mean:} \quad \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

where N represents the size of the entire population, and n is the number of samples measured from the population.

```
─────────── create-normally-distributed-values.R ───────────
# A vector of 50 normally distributed
# random numbers
N = 50
x = rnorm(N)
mean(x)

# Run the code several times, to check
# that the mean is approximately 0

# Check what the 'x' variable contains.
```

```
─────────── create-normally-distributed-values.py ───────────
# A vector of 50 normally distributed
# random numbers. If you have Python 3.8
# or higher, consider using the
# 'statistics' package instead.

import numpy as np

N = 50
x = np.random.normal(size=N)
print(np.mean(x))
# Run the code several times, to check
# that the mean is approximately 0

# Check what the 'x' variable contains.
```

This is only one of several statistics that describes your data: if you told your customer that the average density of your liquid product was 1.421 g/L, and nothing further, the customer might assume all lots of the same product have a density of 1.421 g/L. But we know from *our earlier discussion* (page 30) that there will be variation. We need information, in addition to the mean, to quantify the distribution of values: *the spread*.

Variance (spread)

A measure of spread, or variance, is also essential to quantify your distribution.

$$\text{Population variance :} \quad \mathcal{V}\{x\} = \mathcal{E}\left\{(x-\mu)^2\right\} = \sigma^2 = \frac{1}{N}\sum(x-\mu)^2$$

$$\text{Sample variance :} \quad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$$

Dividing by $n-1$ makes the variance statistic, s^2, an unbiased estimator of the population variance, σ^2. However, in many data sets our value for n is large, so using a divisor of n, which

you might come across in computer software or other texts, rather than $n - 1$ as shown here, leads to little difference.

```
─────────── create-normally-distributed-values-with-variance-parameter.R ───────────
# A vector of 50 normally distributed
# random numbers with a standard
# deviation of 5
N = 50
spread = 5
x = rnorm(N, sd=spread)

paste0('Standard deviation = ',
       round(sd(x), 3))
paste0('The variance is    = ',
       round(var(x), 3))
paste0('Square root of variance = ',
       round(sqrt(var(x)), 3))


# Run the code several times.
```

```
────────── create-normally-distributed-values-with-variance-parameter.py ──────────
# A vector of 50 normally distributed
# random numbers with a standard
# deviation of 5.
# If you have Python 3.8
# or higher, consider using the
# 'statistics' package instead.

import numpy as np

N = 50
spread = 5
x = np.random.normal(loc=0, scale=spread, size=N)
print("Standard deviation     = " +\
      str(np.std(x)))
print("The variance is        = " +\
      str(np.var(x)))
print("Square root of variance = " +\
      str(np.sqrt(np.var(x))))


# Run the code several times.
```

The square root of variance, called the standard deviation is a more useful measure of spread: it is easier to visualize on a histogram and has the advantage of being in the same units of measurement as the variable itself.

Degrees of freedom

The denominator in the sample variance calculation, $n - 1$, is called the degrees of freedom. We have one fewer than n degrees of freedom, because there is a constraint that the sum of the deviations around $\bar{x}$ must add up to zero. This constraint is from the definition of the mean. However, if we knew what the sample mean was without having to estimate it, then we could subtract each x_i from that value, and our degrees of freedom would be n.

Outliers

Video for this section.

Outliers are hard to define precisely, but an acceptable definition is that an outlier is a point that is unusual, given the context of the surrounding data. Another definition which is less useful, but nevertheless points out the problem of concretely defining what an outlier is, is this: "*An outlier - I know it when I see it!*"

The following 2 sequences of numbers show the number **4024** that appears in the first sequence, has become an outlier in the second sequence. It is an outlier based on the surrounding context.

- 4024, 5152, 2314, 6360, 4915, 9552, 2415, 6402, 6261

- 4, 61, 12, 64, 4024, 52, -8, 67, 104, 24

Median (robust measure of location)

The median is an alternative measure of location. It is a sample statistic, not a population statistic, and is computed by sorting the data and taking the middle value (or average of the middle 2 values, for even n). It is also called a robust statistic, because it is insensitive (robust) to outliers in the data.

Note: The median is the most robust estimator of the sample location: it has a breakdown of 50%, which means that just under 50% of the data need to be replaced with unusual values before the median breaks down as a suitable estimate. The mean on the other hand has a breakdown value of $1/n$, as only one of the data points needs to be unusual to cause the mean to be a poor estimate. To compute the median in R, use the `median(x)` function on a vector x.

Governments will report the median income, rather than the mean, to avoid influencing the value with the few very high earners and the many low earners. The median income per person is a more fair measure of location in this case.

Median absolute deviation, MAD (robust measure of spread)

A robust measure of spread is the MAD, the median absolute deviation. The name is descriptive of how the MAD is computed:

$$\text{mad}\{x_i\} = c \cdot \text{median}\{\|x_i - \text{median}\{x_i\}\|\} \qquad \text{where} \qquad c = 1.4826$$

The constant c makes the MAD consistent with the standard deviation when the observations x_i are normally distributed. The MAD has a breakdown point of 50%, because like the median, we can replace just under half the data with outliers before the MAD estimate becomes unbounded. To compute the MAD in R, use the `mad(x)` function on a vector x.

```R
# A vector of 500 normally distributed
# random numbers

x <- rnorm(500)

paste0('Without any outliers:')
paste0('Standard deviation = ', sd(x))
paste0('The MAD is         = ', mad(x))
print('These two should agree mostly')

# Run it several times to verify that the
# two are similar, when they are not
# outliers

# Now add a huge outlier:
x[2] <- 9876
paste0('But now add an outlier...')
paste0('*Standard deviation = ', sd(x))
paste0('*The MAD is         = ', mad(x))
paste0('See how MAD is not affected.')
```

Enrichment reading: read pages *1 to 8* of "Tutorial to Robust Statistics[32]", PJ Rousseeuw, *Journal of Chemometrics*, **5**, 1-20, 1991.

---

[32] https://dx.doi.org/10.1002/cem.1180050103

2.6 Binary (Bernoulli) distribution

Systems that have binary outcomes (pass/fail; yes/no) must obey the probability principle that: $p(\text{pass}) + p(\text{fail}) = 1$. That is, the sum of the probabilities of the two possible outcomes must add up to exactly one. A Bernoulli distribution only has a single parameter, p_1, the probability of observing event 1. The probability of the second event is the difference with 1: that is $p_2 = 1 - p_1$.

An example: a histogram for a system that produces 70% acceptable product, $p(\text{pass}) = 0.7$, could look like:
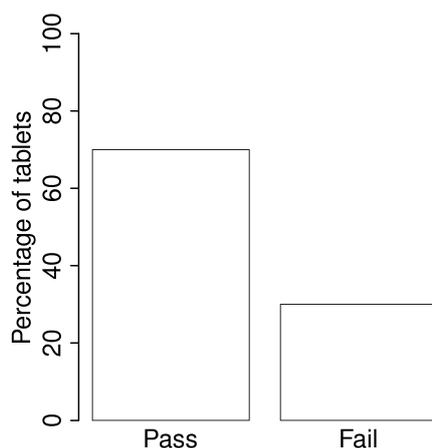
If each observation is independent of the other, then:

- For the above system where $p(\text{pass}) = 0.7$, what is probability of seeing the following sequential outcomes: **pass**, **pass**, **pass** (3 times in a row)?

  $(0.7)(0.7)(0.7) = 0.343$, about one third

- What is the probability of seeing the sequence: **pass**, **fail**, **pass**, **fail**, **pass**, **fail**?

  $(0.7)(0.3)(0.7)(0.3)(0.7)(0.3) = 0.0093$, less than 1%

Another example: you work in a company that produces tablets. The machine creates acceptable, unbroken tablets 97% of the time, so $p_{\text{acceptable}} = 0.97$, so $p_{\text{defective}} = 0.03$.

- In a future batch of 850,000 tablets, how many tablets are expected to be defective? (Most companies will call this quantity "the cost of waste".)

  $850000 \times (1 - 0.97) = 25500$ tablets per batch will be defective

- You take a random sample of n tablets from a large population of N tablets. What is the chance that **all** n tablets are acceptable if p is the Bernoulli population parameter of finding acceptable tablets:

Sample size	$p = 95\%$	$p = 97\%$
$n = 10$		
$n = 50$		
$n = 100$		

- Are you surprised by the large reduction in the number of defective tablets for only a small increase in p? It is for this reason that a well-performing process producing accetable product does not need to have inspection of every product produced.

2.7 Uniform distribution

A uniform distribution arises when an observation's value is equally as likely to occur as all the other options of the recorded values. The classic example are dice: each face of a die is equally as likely to show up as any of the other faces. This forms a discrete, uniform distribution.

The histogram for an event with 4 possible outcomes that are uniformly distributed is shown below. Notice that the *sample* histogram will not necessarily have equal bar heights for all categories (bins), especially for small sample sizes.
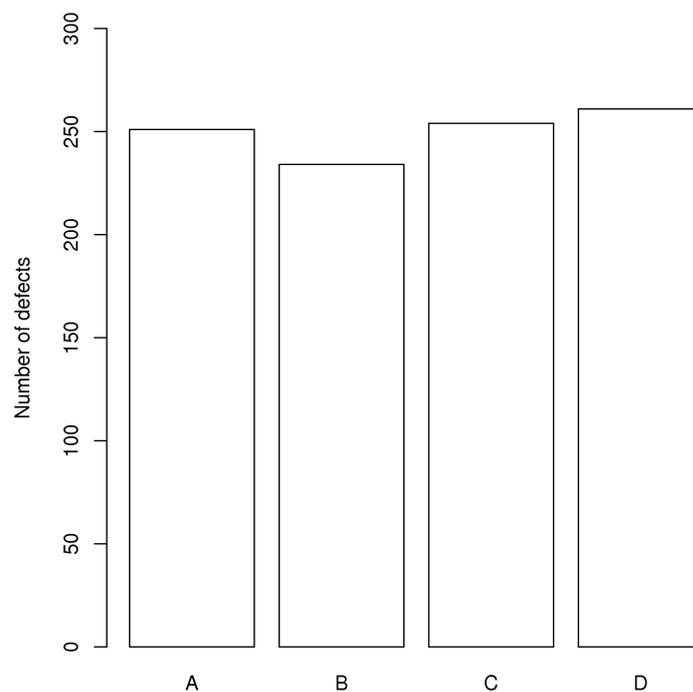
You can simulate uniformly distributed random numbers in most software packages. As an example, to generate 50 uniformly distributed random *integers* between 2 and 10, inclusive, in various languages:

```
─── uniform-distribution-example.R ──────────────
as.integer(runif(50, 2, 11))

# run the code several times to verify
# the numbers are between 2 and 10
```

```
─── uniform-distribution-example.py ──────────────
import numpy as np
(np.random.rand(50) * (10 - 2) + 2).round()

# run the code several times to verify
# the numbers are between 2 and 10
```

A continuous, uniform distribution arises when there is equal probability of every measurement occurring within a given lower- and upper-bound. This sort of phenomena is not often found in practice. Usually, continuous measurements follow some other distribution, of which we will discuss the normal and t-distribution next.

2.8 Normal distribution

Before introducing the normal distribution, we first look at two important concepts: the Central limit theorem, and the concept of independence. Both concepts are used in important derivations, based on the normal distribution.

2.8.1 Central limit theorem

The Central limit theorem plays an important role in the theory of probability and in the derivation of the normal distribution. We don't prove this theorem here, but we only use the result that:

> The average of a sequence of values *from any distribution* will approach the normal distribution, provided the original distribution has finite variance.

The condition of finite variance is true for almost all systems of practical interest.

Any distribution with finite variance → (Take independent samples) $x_1, x_2, x_3, x_4, ..., x_n$ → (Calculate the average of these n samples) $\bar{x}$
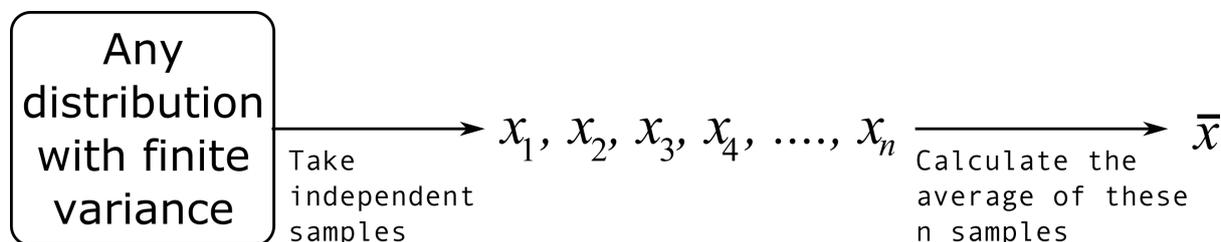
The critical requirement for the central limit theorem to be true is that the samples used to compute that average are independent of each together. The average produced from such samples will be more nearly normal though. Note: we **do not** require the original data to be normally distributed. This is a common misconception though.

Imagine a case where we are throwing dice. The distributions, shown below, are obtained when we throw a die M times and we plot the distribution of the *average* of these M throws.
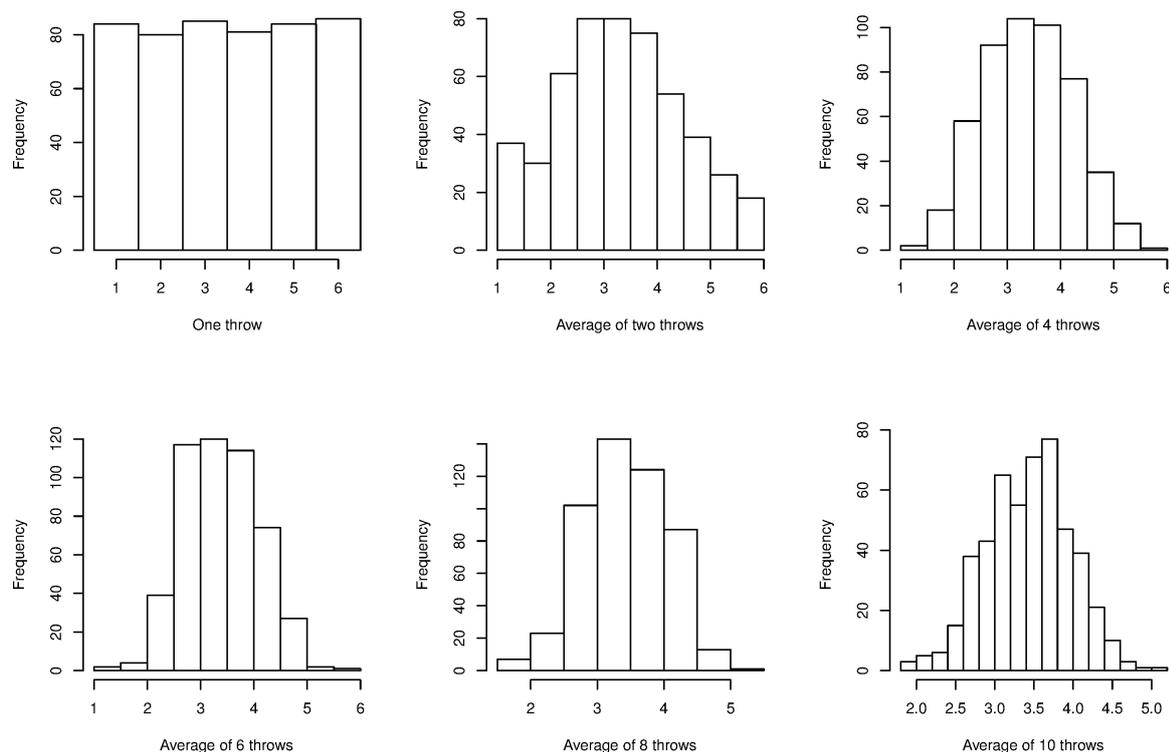
As one sees from the above figures, the distribution from these averages quickly takes the shape of the so-called *normal distribution*. As M increases, the y-axis starts to form a peak. Try it yourself:

```
—————————————————————————— simulate-CLT.R ——————————————————————————
N = 500

# Layout the plots in 2 rows and 3 columns
m <- t(matrix(seq(1,6), 3, 2))
layout(m)

# Throw the dice several times
s1 <- as.integer(runif(N, 1, 7))
s2 <- as.integer(runif(N, 1, 7))
s3 <- as.integer(runif(N, 1, 7))
s4 <- as.integer(runif(N, 1, 7))
s5 <- as.integer(runif(N, 1, 7))
s6 <- as.integer(runif(N, 1, 7))
s7 <- as.integer(runif(N, 1, 7))
s8 <- as.integer(runif(N, 1, 7))
s9 <- as.integer(runif(N, 1, 7))
s10 <- as.integer(runif(N, 1, 7))


hist(s1, main="", xlab="One throw", breaks=seq(0,6)+0.5)
bins = 8
hist((s1+s2)/2, breaks=bins,
     main="", xlab="Average of two throws")
hist((s1+s2+s3+s4)/4, breaks=bins,
     main="", xlab="Average of 4 throws")
hist((s1+s2+s3+s4+s5+s6)/6, breaks=bins,
     main="", xlab="Average of 6 throws")
bins=12
hist((s1+s2+s3+s4+s5+s6+s7+s8)/8, breaks=bins,
     main="", xlab="Average of 8 throws")
hist((s1+s2+s3+s4+s5+s6+s7+s8+s9+s10)/10, breaks=bins,

     main="", xlab="Average of 10 throws")
```

What is the engineering significance of this averaging process (which is really just a weighted sum)? Many of the quantities we measure are bulk properties, such as viscosity, density, or particle size. We can conceptually imagine that the bulk property measured is the combination of the same property, measured on smaller and smaller components. Even if the value measured on the smaller component is not normally distributed, the bulk property will be as if it came from a normal distribution.

2.8.2 Independence

The assumption of independence is widely used in statistical work and is a condition for using the central limit theorem.

Note: The assumption of independence means that the samples we have in front of us are *randomly taken* from a population. If two samples are independent, there is no possible relationship between them.

We frequently violate this assumption of independence in engineering applications. Think about these examples for a while:

- A questionnaire is given to a group of people. What happens if they discuss the questionnaire in sub-groups prior to handing it in?

  We are not going to receive n independent answers, rather we will receive as many independent opinions as there are sub-groups.

- The rainfall amount, recorded every day, over the last 30 days.

  These data are not independent: if it rains today, it can likely rain tomorrow as the weather usually stays around for some days. These data are not useful as a representative sample of typical rainfall, however they are useful for complaining about the weather. Think about the case if we had considered rainfall in hourly intervals, rather than daily intervals.

- The snowfall, recorded on 3 January for every year since 1976: independent or not?

  These sampled data will be independent.

- The impurity values in the last 100 batches of product produced is shown below. Which of the 3 time sequences has independent values?

  In chemical processes there is often a transfer from batch-to-batch: we usually use the same lot of raw materials for successive batches, the batch reactor may not have been cleaned properly between each run, and so on. It is very likely that two successive batches (k and $k + 1$) are somewhat related, and less likely that batch k and $k + 2$ are related. In the figure below, can you tell which sequence of values are independent?
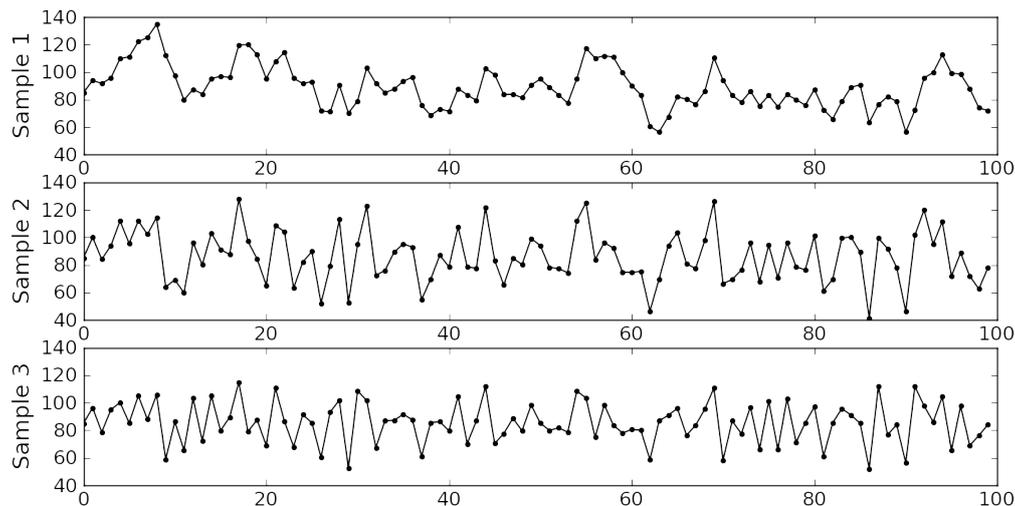
Sequence 2 (sequence 1 is positively correlated, while sequence 3 is negatively correlated).

- We need a highly reliable pressure release system. Manufacturer A sells a system that fails 1 in every 100 occasions, and manufacturer B sells a system that fails 3 times in every 1000 occasions. Given this information, answer the following:

  - The probability that system A fails: $p(A_{\text{fails}}) = 1/100$

  - The probability that system B fails:$p(B_{\text{fails}}) = 3/1000$

  - The probability that both system A and fail at the same time:
    $p(\text{both A and B fail}) = \frac{1}{100} \cdot \frac{3}{1000} = 3 \times 10^{-5}$, but only if system A and B are totally independent.

  - For the previous question, what does it mean for system A to be totally independent of system B?

    It means the 2 systems must be installed in parallel, so that there is no interaction between them at all.

  - How would the probability of both A and B failing simultaneously change if A and B were not independent?

    The probability of both failing simultaneously will increase.

2.8.3 Formal definition for the normal distribution

Video for this section.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Normal distribution when μ=0 and σ=1
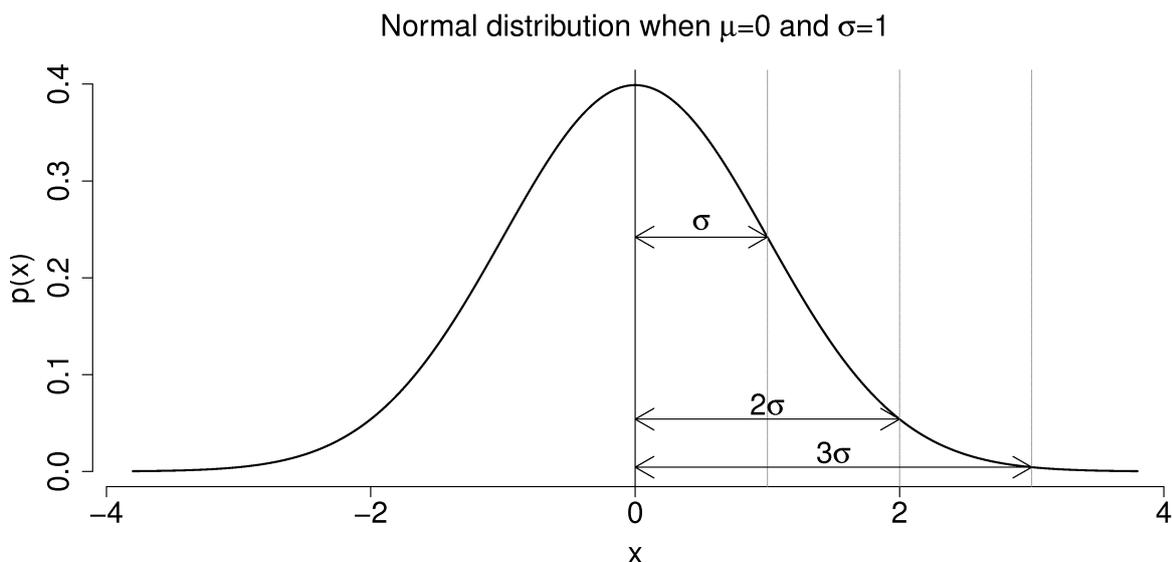
- x is the variable of interest

- $p(x)$ is the probability of obtaining that value of x

- μ is the population average for the distribution (first parameter)

- σ is the population standard deviation for the distribution, and is always a positive quantity (second parameter)

Some questions:

1. What is the maximum value of $p(x)$ and where does it occur, using the formula above?

2. What happens to the shape of $p(x)$ as σ gets larger ?

3. What happens to the shape of $p(x)$ as $\sigma \to 0$?

4. Fill out this table:

x	σ	μ	$p(x)$
0	1	0	
1	1	0	
-1	1	0	

To calculate the point on the curve $p(x)$ we use the `dnorm(...)` function in R. It requires you specify the two parameters:

```R
# x=0, mu=0, and sigma=1
# This is the maximum of the curve
dnorm(x = 0, mean = 0, sd = 1)   # 0.3989423

# x=1, mu=0, and sigma=1
dnorm(x = 1, mean = 0, sd = 1)   # 0.2419707

# x=-1, mu=0, and sigma=1
# It is symmetrical
dnorm(x = -1, mean = 0, sd = 1) # 0.2419707

# x=+3, mu=0, and sigma=1
```

(continues on next page)

```
# This is at a point very far from center
dnorm(x = +3, mean = 0, sd = 1) # 0.00443185
```

Some useful points:

- The total area from $x = -\infty$ to $x = +\infty$ is 1.0; we cannot calculate the integral of $p(x)$ analytically.

- σ is the distance from the mean, μ, to the point of inflection

- The normal distribution only requires two parameters to describe it: μ and σ

- The area from $x = -\sigma$ to $x = \sigma$ is about 70% (68.3% exactly) of the distribution. So we have a probability of about 15% of seeing an x value greater than $x = \sigma$, and also 15% of $x < -\sigma$

- The tail area outside $\pm 2\sigma$ is about 5% (2.275 outside each tail)

It is more useful to calculate the area under $p(x)$ from $x = -\infty$ to a particular point x. This is called the cumulative distribution, and is discussed more fully in *the next section* (page 50).

```
                              ———— R code ————
# gives area from -Inf to -1,
# for mu=0, sigma=1
pnorm(-1, mean = 0, sd = 1)    # 0.1586553

# Gives area from -Inf to +1,
# for mu=0, sigma=1
pnorm(1, mean = 0, sd = 1)     # 0.8413447

# Spread is wider, but the
# fractional area is the same
pnorm(3, mean = 0, sd = 3)     # 0.8413447
```

You might still find yourself having to refer to tables of cumulative area under the normal distribution, instead of using the `pnorm()` function (for example in a test or exam). If you look at the appendix of most statistical texts you will find these tables, and there is one *at the end of this chapter* (page 79). Since these tables cannot be produced for all combinations of mean and standard deviation parameters, they use what is called *standard form*.

Video for this section.

$$z_i = \frac{x_i - \text{mean}}{\text{standard deviation}}$$

The values of the mean and standard deviation are either the population parameters, if known, or using the best estimate of the mean and standard deviation from the sampled data.

For example, if our values of x_i come from a normal distribution with mean of 34.2 and variance of 55. Then we could write $x \sim \mathcal{N}(34.2, 55)$, which is short-hand notation of saying the same thing. The equivalent z-values for these x_i values would be: $z_i = \dfrac{x_i - 34.2}{\sqrt{55}}$.

This transformation to standard form **does not change the distribution** of the original x, it only changes the parameters of the distribution. You can easily prove to yourself that z is normally distributed as $z \sim \mathcal{N}(0.0, 1.0)$. So statistical tables only report the area under the distribution of a z value with mean of zero, and unit variance.

This is a common statistical technique, to standardize a variable, which we will see several times. Standardization takes our variable from $x \sim \mathcal{N}(\text{some mean}, \text{some variance})$ and converts it to $z \sim \mathcal{N}(0.0, 1.0)$. It is just as easy to go backwards, from a given z-value and return back to our original x-value.

The units of z are dimensionless, no matter what the original units of x were. Standardization also allows us to straightforwardly compare 2 variables that may have different means and spreads. For example if our company has two reactors at different locations, producing the same product. We can standardize a variable of interest, e.g. viscosity, from both reactors and then proceed to use the standardized variables to compare performance.

Consult a statistical table found in most statistical textbooks for the normal distribution, such as the one found at the *end of this chapter* (page 79). Make sure you can firstly understand how to read the table. Secondly, duplicate a few entries in the table using R. Complete these small exercises by estimating what the rough answer should be. Use the tables first, then use R to get a more accurate estimate.

1. Assume x, the measurement of biological activity for a drug, is normally distributed with mean of 26.2 and standard deviation of 9.2. What is the probability of obtaining an activity reading less than or equal to 30.0?

   ────────────────────────────── R code ──────────────────────────────
   ```
   # We know that the probability should be 50%
   # if the activity is equal to the mean.

   x <- 26.2
   mu <- 26.2
   sigma <- ____
   pnorm(x, mean=mu, sd=sigma)

   # Now modify this above to answer the question.
   ```

2. Assume x is the yield for a batch process, with mean of 85 g/L and **variance** of 16 $g^2.L^{-2}$. What proportion of batch yield values lie between 75 and 95 g/L?

   ────────────────────────────── R code ──────────────────────────────
   ```
   mu <- 85 # g/L
   sigma <- sqrt(16) # g/L
   x.left <- ___
   area.left.tail <- pnorm(x.left,
                           mean=mu,
                           sd=sigma)

   x.right <- ___
   area.right.tail <- pnorm(x.right,
                            mean=mu,
                            sd=sigma)

   # Now subtract the two areas to get
   # the answer. Why?
   ```

2.8.4 Checking for normality: using a q-q plot

Often we are not sure if a sample of data can be assumed to be normally distributed. This section shows you how to test whether the data are normally distributed, or not.

Before we look at this method, we need to introduce the concept of the inverse cumulative distribution function (inverse CDF). Recall the **cumulative distribution** is the area underneath the distribution function, $p(z)$, which goes from $-\infty$ to z. For example, the area from $-\infty$ to $z = -1$ is about 15%, as we showed earlier, and we can use the `pnorm()` function in R to verify that.

Now the **inverse cumulative distribution** is used when we know the area, but want to get back to the value along the z-axis. For example, below which value of z does 95% of the area lie for a standardized normal distribution? Answer: $z = 1.64$. In R we use the `qnorm(0.95, mean=0, sd=1)` to calculate

this value. The q stands for quantile[33], because we give it the quantile and it returns the z-value: e.g. `qnorm(0.5)` gives 0.0.
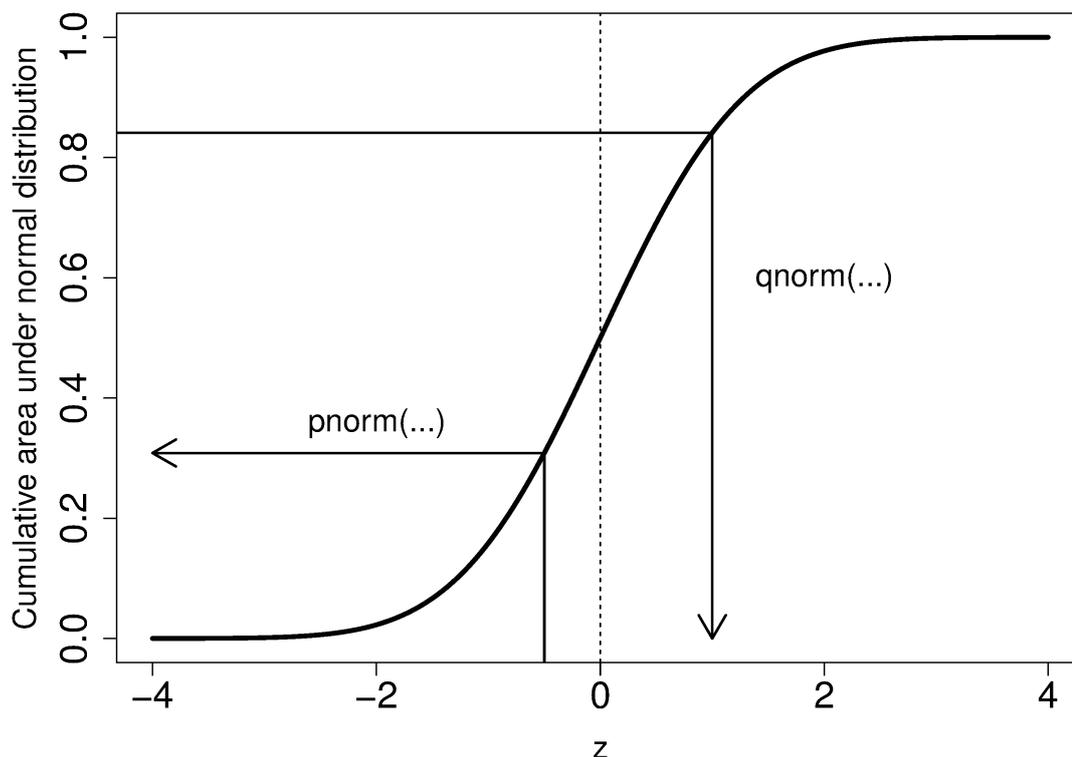
On to checking for normality. We start by first constructing some quantities that we would expect for truly normally distributed data. Secondly, we construct the same quantities for the actual data. A plot of these 2 quantities against each other will reveal if the data are normal, or not.

1. Imagine we have N observations which are normally distributed. Sort the data from smallest to largest. The first data point should be the $(1/N \times 100)$ quantile, the next data point is the $(2/N \times 100)$ quantile, the middle, sorted data point is the 50th quantile, $(1/2 \times 100)$, and the last, sorted data point is the $(N/N \times 100)$ quantile.

   The middle, sorted data point from this truly normal distribution must have a z-value on the standardized scale of 0.0 (we can verify that by using `qnorm(0.5)`). By definition, 50% of the data should lie below this mid point. The first data point will be at `qnorm(1/N)`, the second at `qnorm(2/N)`, the middle data point at `qnorm(0.5)`, and so on. In general, the ith sorted point should be at `qnorm((i-0.5)/N)`, for values of $i = 1, 2, \ldots, N$. We subtract off 0.5 by convention to account for the fact that `qnorm(1.0) = Inf`. So we construct this vector of theoretically expected quantities from the inverse cumulative distribution function.

```
N = 10
index = seq(1, N)
P = (index - 0.5) / N
P
[1] 0.05  0.15  0.25  0.35  0.45  0.55  0.65  0.75  0.85  0.95
theoretical.quantity = qnorm(P)
[1] -1.64 -1.04 -0.674 -0.385 -0.126  0.125  0.385  0.6744 1.036  1.64
```

2. We also construct the actual quantiles for the sampled data. First, standardize the sampled data by subtracting off its mean and dividing by its standard deviation. Here is an example of 10 batch yields (see actual values below). The mean yield is 80.0 and the standard deviation is 8.35. The

---

[33] https://en.wikipedia.org/wiki/Quantile

standardized yields are found by subtracting off the mean and dividing by the standard deviation. Then the standardized values are sorted. Compare them to the theoretical quantities.

```
yields <- c(86.2, 85.7, 71.9, 95.3, 77.1, 71.4, 68.9, 78.9, 86.9, 78.4)
mean.yield <- mean(yields)            # 80.0
sd.yield <- sd(yields)                # 8.35

yields.z = (yields - mean.yield)/sd.yield
[1] 0.734  0.674 -0.978  1.82 -0.35 -1.04 -1.34 -0.140  0.818 -0.200

yields.z.sorted = sort(yields.z)
[1] -1.34 -1.04 -0.978 -0.355 -0.200 -0.140  0.674  0.734  0.818  1.82

theoretical.quantity  # numbers are rounded in the printed output
[1] -1.64 -1.04 -0.674 -0.385 -0.126  0.125  0.385  0.6744 1.036  1.64
```

3. The final step is to plot this data in a suitable way. If the sampled quantities match the theoretical quantities, then a scatter plot of these numbers should form a 45 degree line.

```
plot(theoretical.quantity, yields.z.sorted, type="p")
```
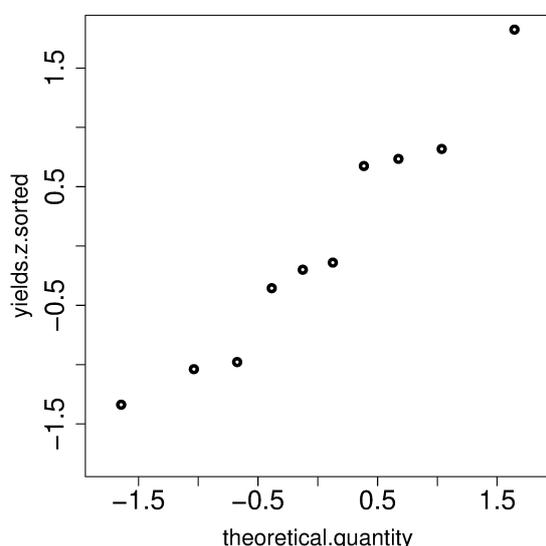

A built-in function exists in R that runs the above calculations and shows a scatter plot. The 45 degree line is added using the `qqline(...)` function. However, a better function that adds a confidence limit envelope is included in the `car` library (see the *Package Installer* menu in R for adding libraries from the internet).

```
qqnorm(yields)
qqline(yields)

# or, using the ``car`` library
library(car)
qqPlot(yields)
```
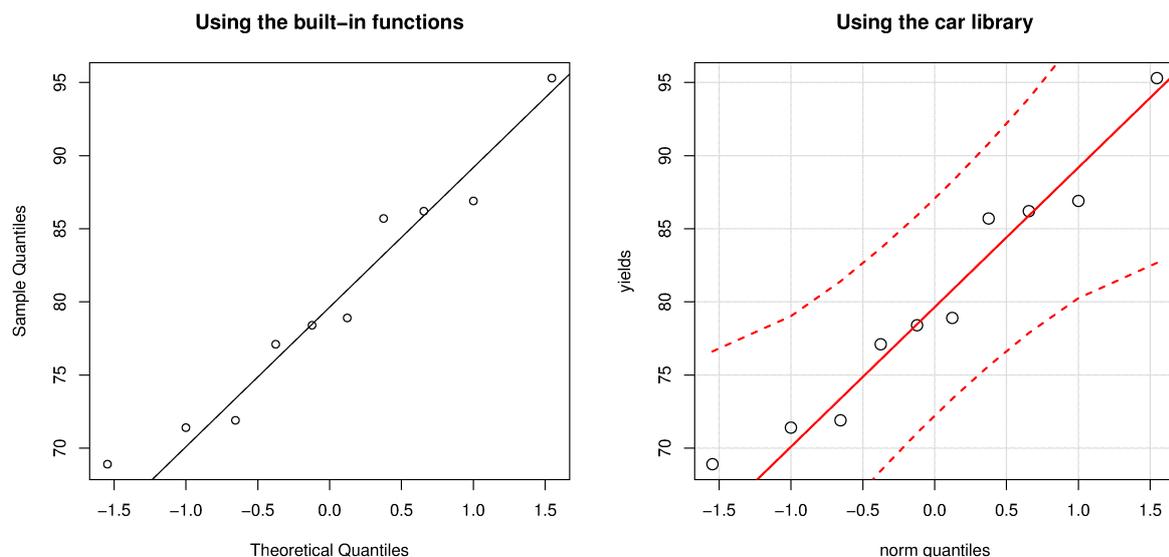
All the above code together in one script for you to test out:

```R code
N = 10
index <- seq(1, N)
P <- (index - 0.5) / N
theoretical.quantity <- qnorm(P)

yields <- c(86.2, 85.7, 71.9, 95.3, 77.1,
            71.4, 68.9, 78.9, 86.9, 78.4)
mean.yield <- mean(yields)          # 80.0
sd.yield <- sd(yields)              # 8.35

yields.z <- (yields - mean.yield)/sd.yield
yields.z.sorted <- sort(yields.z)

plot(theoretical.quantity,
     yields.z.sorted,
     type="p")

qqnorm(yields)
qqline(yields)

# or, using the ``car`` library
library(car)

qqPlot(yields)
```

The R plot rescales the y-axis (sample quantiles) back to the original units to make interpretation easier. We expect some departure from the 45 degree line due to the fact that these are only a sample of data. However, large deviations indicates the data are not normally distributed. An error region, or confidence envelope, may be superimposed around the 45 degree line.

The q-q plot, quantile-quantile plot, shows the quantiles of 2 distributions against each other. In fact, we can use the horizontal axis for any distribution, it need not be the theoretical normal distribution. We might be interested if our data follow an F-distribution then we could use the quantiles for that theoretical distribution on the horizontal axis.

We can use the q-q plot to compare any 2 *samples of data*, even if they have different values of N, by calculating the quantiles for each sample at different step quantiles (e.g. 1, 2, 3, 4, 5, 10, 15, ... 95, 96, 97, 98, 99), then plot the q-q plot for the two samples. You can calculate quantiles for any sample of data using the `quantile` function in R. The simple example below shows how to compare the q-q

plot for 1000 normal distribution samples against 2000 F-distribution samples.

```
                                    ——— qqplot-comparison.R ———
# 1000 normal values
rand.norm <- rnorm(1000)

# 2000 values from F-distribution
rand.f <- rf(2000, df1=200, df=150)

# looks sort of normally distributed
hist(rand.f, freq=FALSE, ylim=c(0, 2.6),
     main="Are these data from a normal distribution?",
     ylab="Frequency")

# Add the density line on top
lines(density(rand.f))

# But your eye is being fooled ...
# See the heavy tail
library(car)

qqPlot(rand.f, distribution="norm")
```
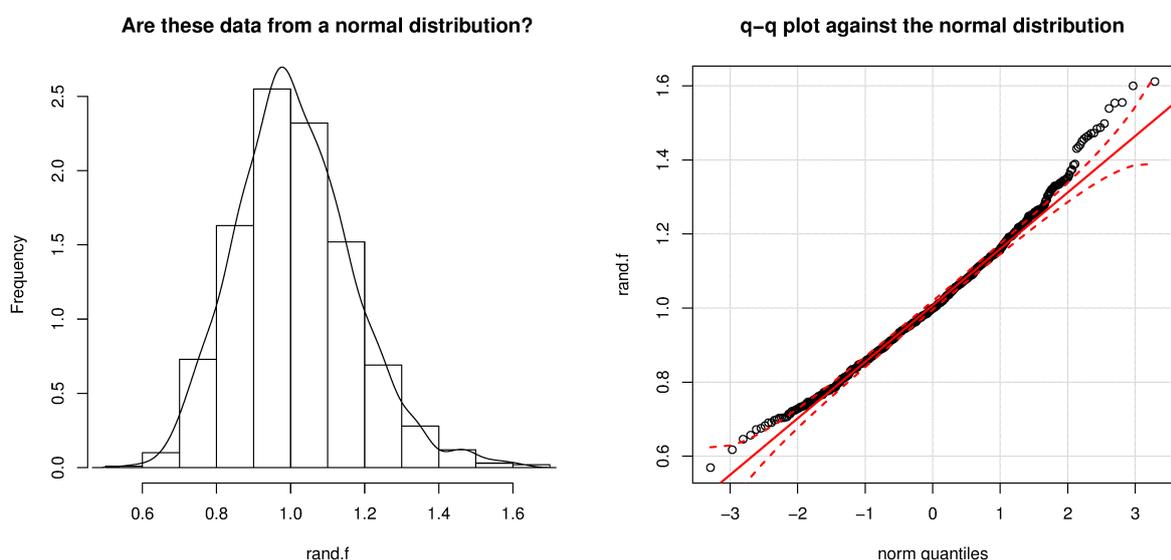

Even though the histogram of the F-distribution samples looks normal to the eye (left), the q-q plot (right) quickly confirms it is definitely not normal, particularly, that the right-tail is too heavy.

2.8.5 Introduction to confidence intervals from the normal distribution

We introduce the concept of confidence intervals here as a straightforward application of the normal distribution, Central limit theorem, and standardization.

Suppose we have a quantity of interest from a process, such as the daily profit. We have many measurements of this profit, and we can easily calculate the **average** profit. But we know that if we take a different data set of profit values and calculate the average, we will get a similar, but different average. Since we will never know the true population average, the question we want to answer is:

> What is the range within which the true (population) average value lies? E.g. give a range for the true, but unknown, daily profit.

This range is called a confidence interval, and we study them *in more depth later on* (page 63). We will use an example to show how to calculate this range.

Let's take n values of this daily profit value, let's say $n = 5$.

1. An estimate of the population mean is given by $\bar{x} = \dfrac{1}{n}\sum_{i}^{i=n} x_i$ (we *saw this before* (page 39))

2. The estimated population variance is $s^2 = \dfrac{1}{n-1}\sum_{i}^{i=n} (x_i - \bar{x})^2$ (we also *saw this before* (page 39))

3. This is new: the estimated mean, $\bar{x}$, is a value that is also normally distributed with mean of μ and variance of σ^2/n, with only one requirement: this result holds only if each of the x_i values are independent of each other.

    Mathematically we write: $\bar{x} \sim \mathcal{N}\left(\mu, \sigma^2/n\right)$.

    This important result helps answer our question above. It says that repeated estimates of the mean will be an accurate, unbiased estimate of the population mean, and interestingly, the variance of that estimate is decreased by using a greater number of samples, n, to estimate that mean. This makes intuitive sense: the more **independent** samples of data we have, the *better* our estimate ("better" in this case implies lower error, i.e. lower variance).

    We can illustrate this result as shown below:

Raw data distribution, and sample mean distribution
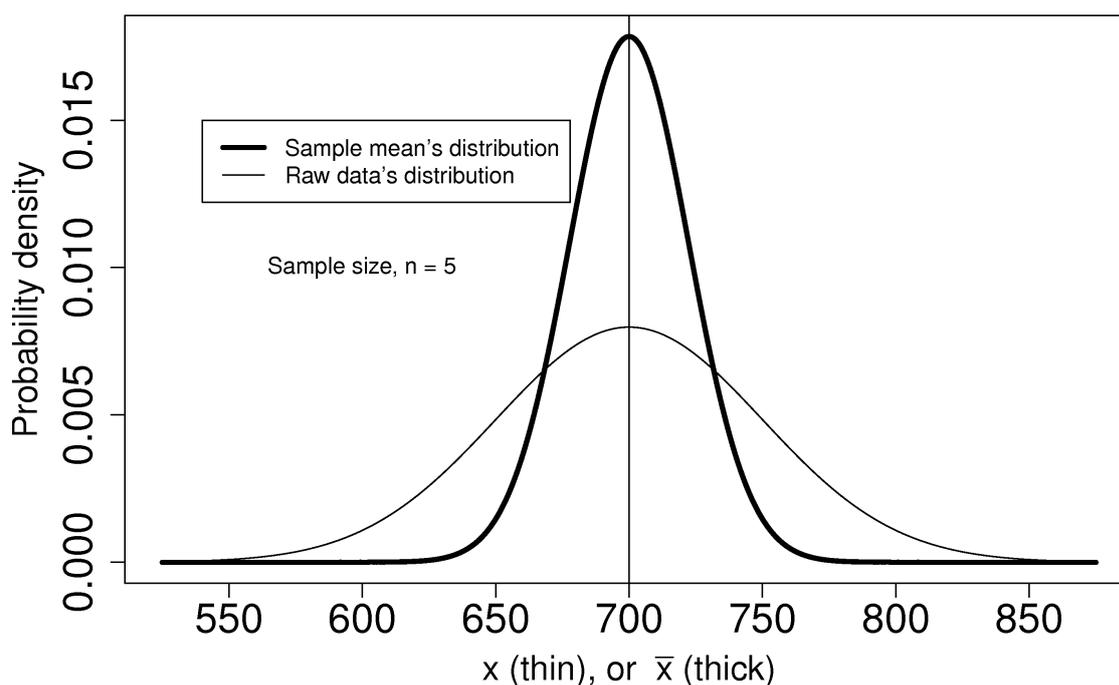
The true population (but unknown to us) profit value is $700.

- The 5 samples come from the distribution given by the thinner line: $x \sim \mathcal{N}\left(\mu, \sigma^2\right)$

- The $\bar{x}$ average comes from the distribution given by the thicker line: $\bar{x} \sim \mathcal{N}\left(\mu, \sigma^2/n\right)$.

4. Creating z values for each x_i raw sample point:

$$z_i = \frac{x_i - \mu}{\sigma}$$

5. The z-value for $\overline{x}$ would be:

$$z = \frac{\overline{x} - \mu}{\sigma/\sqrt{n}}$$

which subtracts off the unknown population mean from our estimate of the mean, and divides through by the standard deviation for $\overline{x}$. We can illustrate this as:

Raw data distribution, and sample mean distribution
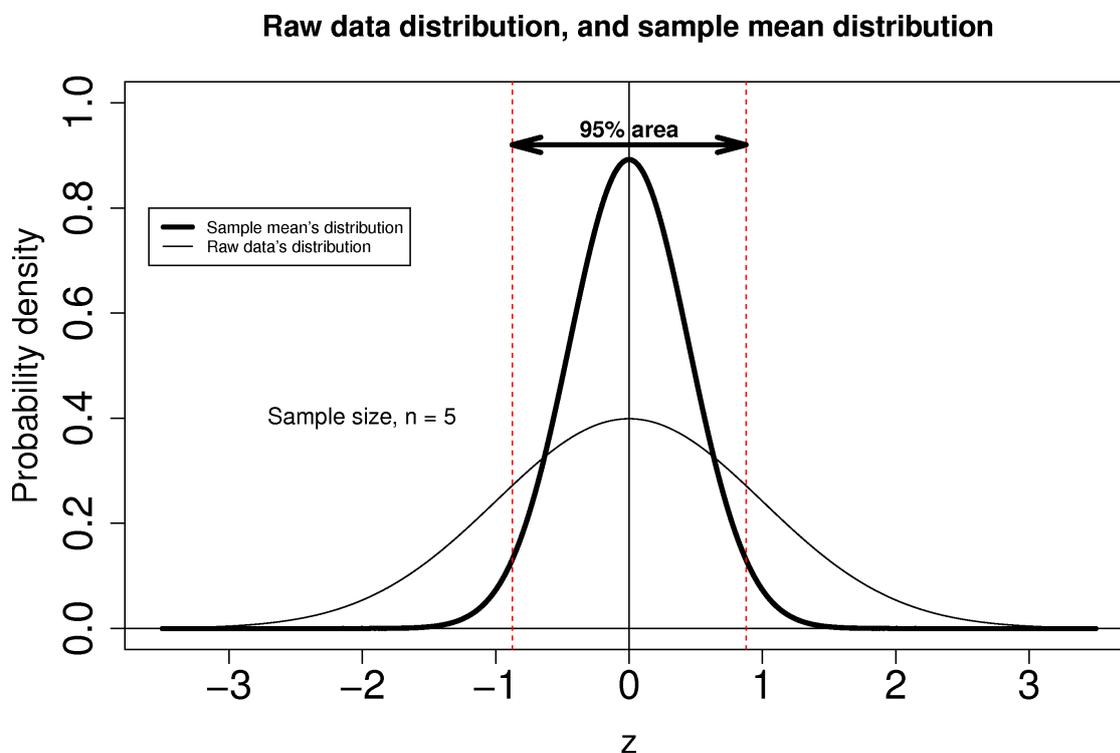
6. Using the known normal distribution for $\overline{x} \sim \mathcal{N}\left(\mu, \sigma^2/n\right)$, we can find the vertical, dashed red lines shown in the previous figure, that contain 95% of the area under the distribution for $\overline{x}$.

7. These vertical lines are symmetrical about 0, and we will call them $-c_n$ and $+c_n$, where the subscript n refers to the fact that they are from the normal distribution (it doesn't refer to the n samples). From the preceding section on q-q plots we know how to calculate the c_n value from R: using `qnorm(1 - 0.05/2)`, so that there is 2.5% area in each tail.

8. Finally, we construct an interval for the true population mean, μ, using the standard form:

$$
\begin{array}{ccccc}
-c_n & \leq & z & \leq & +c_n \\
-c_n & \leq & \dfrac{\overline{x} - \mu}{\sigma/\sqrt{n}} & \leq & +c_n \\
\overline{x} - c_n \dfrac{\sigma}{\sqrt{n}} & \leq & \mu & \leq & \overline{x} + c_n \dfrac{\sigma}{\sqrt{n}} \\
\text{LB} & \leq & \mu & \leq & \text{UB}
\end{array}
\tag{2.1}
$$

Notice that the lower and upper bound are a function of the known sample mean, $\overline{x}$, the values for c_n which we chose, the known sample size, n, and the unknown population standard deviation, σ.

So to estimate our bounds we must know the value of this population standard deviation. This is not very likely, (I can't think of any practical cases where we know the population standard deviation, but not the population mean, which is the quantity we are constructing this range for), however there is a hypothetical example in *the next section* (page 59) to illustrate the calculations.

The t-distribution is required to remove this impractical requirement of knowing the population standard deviation.

2.9 The t-distribution

Suppose we have a quantity of interest from a process, such as the daily profit. In the preceding section we started to answer the useful and important question:

> What is the range within which the true average value lies? E.g. the range for the true, but unknown, daily profit.

But we got stuck, because the lower and upper bounds we calculated for the true average, μ were a function of the unknown population standard deviation, σ. Repeating *the prior equation for confidence interval* (page 65) where we know the variance:

$$
\begin{array}{ccccc}
-c_n & \leq & \dfrac{\overline{x} - \mu}{\sigma/\sqrt{n}} & \leq & +c_n \\[2mm]
\overline{x} - c_n \dfrac{\sigma}{\sqrt{n}} & \leq & \mu & \leq & \overline{x} + c_n \dfrac{\sigma}{\sqrt{n}} \\[2mm]
\text{LB} & \leq & \mu & \leq & \text{UB}
\end{array}
$$

which we derived by using the fact that $\dfrac{\overline{x} - \mu}{\sigma/\sqrt{n}}$ is normally distributed.

An obvious way out of our dilemma is to replace σ by the sample standard deviation, s, which is exactly what we will do, however, the quantity $\frac{\overline{x} - \mu}{s/\sqrt{n}}$ is not normally distributed, but is t-distributed. Before we look at the details, it is helpful to see how similar in appearance the t and normal distribution are: the t-distribution peaks slightly lower than the normal distribution, but it has broader tails. The total area under both curves illustrated here is 1.0.
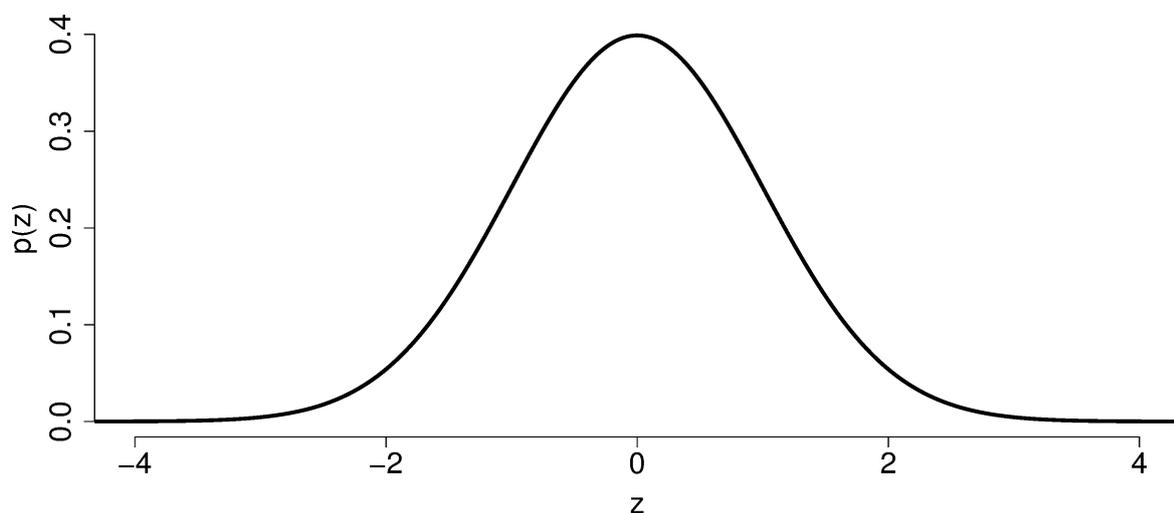
There is one other requirement we have to ensure in order to use the t-distribution: the values that we sample, x_i must come from a normal distribution (carefully note that in the previous section we didn't have this restriction!). Fortunately it is easy to check this requirement: just use the *q-q plot method described earlier* (page 50). Another requirement, which we had before, was that we must be sure these measurements, x_i, are independent.

```
┌─────────────────┐
│     Normal      │  Take
│  distribution   │  independent ────────▶  x₁, x₂, x₃, x₄, ...., xₙ ─── Calculate ──▶  x̄
│   N(μ, σ²)      │  samples                                            the mean
└─────────────────┘
```
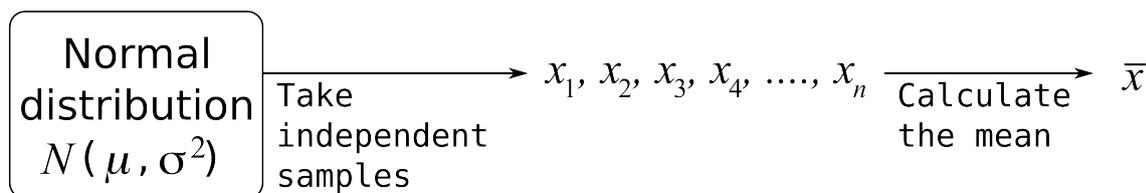
So given our n samples, which are independent, and from a normal distribution, we can now say:

$$\frac{\overline{x} - \mu}{s/\sqrt{n}} \sim t_{n-1} \tag{2.2}$$

Compare this to the previous case where our n samples are independent, and we happen to know, by some unusual way, what the population standard deviation is, σ:

$$\frac{\overline{x} - \mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0, 1)$$

So the more practical and useful case where $z = \frac{\overline{x}-\mu}{s/\sqrt{n}} \sim t_{n-1}$ can now be used to construct an interval for μ. We say that z follows the t-distribution with $n - 1$ degrees of freedom, where the degrees of freedom refer to those from the calculating the *estimated* standard deviation, s.

Note that the new variable z only requires we know the population mean (μ), not the population standard deviation; rather we use our estimate of the standard deviation $s/\sqrt{n}$ in place of the population standard deviation.

We will come back to (2.2) in a minute; let's first look at how we can calculate values from the t-distribution in computer software.

2.9.1 Calculating the t-distribution

- In R we use the function `dt(x=..., df=...)` to give us the values of the probability density values, $p(x)$, of the t-distribution (compare this to the `dnorm(x, mean=..., sd=...)` function for the normal distribution).

```
──────────────────────────────── R code ────────────────────────────────
x = 0.0

# Recall, for the normal distribution:
dnorm(x, mean=0, sd=1)      # 0.3989423

# For the t-distribution we don't have
# a sigma, but we do need to say how
# many degrees of freedom we have:

dof <- 8
dt(x, df=dof)               # 0.386699

# Shows that the t-distribution has a
# lower peak than the normal distribution.
# Try it again, but with fewer and
# greater degrees of freedom (`dof`).
```

- The cumulative area from $-\infty$ to x under the probability density curve gives us the probability that values less than or equal to x could be observed. It is calculated in R using `pt(q=..., df=...)`. For example, `pt(1.0, df=8)` is 0.8267. Compare this to the R function for the standard normal distribution: `pnorm(1.0, mean=0, sd=1)` which returns 0.8413.

```R
――――――――――――――――――――――― R code ―――――――――――――――――――――――
q = 1.0

# Recall, for the normal distribution:
pnorm(q, mean=0, sd=1) # 0.8413447

# For the t-distribution we need to
# specify the degrees of freedom:

dof <- 8
pt(q, df=dof)           # 0.8267032

# Shows that the t-distribution is
# similar, but the areas are slightly
# different.
```

- And similarly to the `qnorm` function which returns the ordinate for a given area under the normal distribution, the function `qt(0.8267, df=8)` returns 0.9999857, close enough to 1.0, which is the inverse of the previous example.

```R
――――――――――――――――――――――― R code ―――――――――――――――――――――――
p = 0.5

# Recall, for the normal distribution:
qnorm(p, mean=0, sd=1)   # 0.0

# For the t-distribution:

dof <- 8
qt(p, df=dof)            # 0.0

# Both distributions have their 50%
# quantile at p=0. But try it for
# other values of probability, p.
```

2.9.2 Using the t-distribution to calculate our confidence interval

Returning back to (2.2) we stated that

$$\frac{\overline{x} - \mu}{s/\sqrt{n}} \sim t_{n-1}$$

We can plot the t-distribution for a given value of $n - 1$, the degrees of freedom. Then we can locate vertical lines on the x-axis at $-c_t$ and $+c_t$ so that the area between the verticals covers say 95% of the total distribution's area. The subscript t refers to the fact that these are critical values from the t-distribution.

Then we write:

$$
\begin{array}{ccccc}
-c_t & \leq & z & \leq & +c_t \\
-c_t & \leq & \dfrac{\overline{x} - \mu}{s/\sqrt{n}} & \leq & +c_t \\
\overline{x} - c_t \dfrac{s}{\sqrt{n}} & \leq & \mu & \leq & \overline{x} + c_t \dfrac{s}{\sqrt{n}} \\
\text{LB} & \leq & \mu & \leq & \text{UB}
\end{array}
\tag{2.3}
$$

Now all the terms in the lower and upper bound are known, or easily calculated.

So we finish this section off with an example. We produce large cubes of polymer product on our process. We would like to estimate the cube's average viscosity, but measuring the viscosity is a destructive laboratory test. So using 9 independent samples taken from this polymer cube, we get the 9 lab values of viscosity: `23, 19, 17, 18, 24, 26, 21, 14, 18`.

If we repeat this process with a different set of 9 samples we will get a different average viscosity. So we recognize the average of a sample of data, is itself just a single estimate of the population's average. What is more helpful is to have **a range**, given by a lower and upper bound, that we can say the true population mean lies within.

1. The average of these nine values is $\bar{x} = 20$ units.

2. Using the Central limit theorem, what is the distribution from which $\bar{x}$ comes?

   $$\bar{x} \sim \mathcal{N}\left(\mu, \sigma^2/n\right)$$

   This also requires the assumption that the samples are independent estimates of the population viscosity. We **don't** have to assume the x_i are normally distributed.

3. What is the distribution of the sample average? What are the parameters of that distribution?

   The sample average is normally distributed as $\mathcal{N}\left(\mu, \sigma^2/n\right)$

4. Assume, for some hypothetical reason, that we know the population viscosity standard deviation is $\sigma = 3.5$ units. Calculate a lower and upper bound for μ:

   The interval is calculated using from an *earlier equation when discussing the normal distribution* (page 56):

   $$\text{LB} = \bar{x} - c_n \frac{\sigma}{\sqrt{n}}$$
   $$= 20 - 1.95996 \cdot \frac{3.5}{\sqrt{9}}$$
   $$= 20 - 2.286 = \mathbf{17.7}$$
   $$\text{UB} = 20 + 2.286 = \mathbf{22.3}$$

5. We can confirm these 9 samples are normally distributed by using a q-q plot (not shown, but you can use the code below to generate the plot). This is an important requirement to use the t-distribution, next.

6. Calculate an estimate of the standard deviation.

   $$s = 3.81$$

7. Now construct the z-value for the sample average and from what distribution does this z come from?

   It comes the t-distribution with $n - 1 = 8$ degrees of freedom, and is given by $z = \dfrac{\bar{x} - \mu}{s/\sqrt{n}}$

8. Construct an interval, symbolically, that will contain the population mean of the viscosity. Also calculate the lower and upper bounds of the interval assuming the internal to span 95% of the area of this distribution.

   The interval is calculated using (2.3):

   $$\text{LB} = \bar{x} - c_t \frac{s}{\sqrt{n}}$$
   $$= 20 - 2.306004 \cdot \frac{3.81}{\sqrt{9}}$$
   $$= 20 - 2.929 = 17.1$$
   $$\text{UB} = 20 + 2.929 = 22.9$$

   using from R that `qt(0.025, df=8)` and `qt(0.975, df=8)`, which gives `2.306004`

```r
# ─────────────────────── R code ───────────────────────
# Step 0: the raw data
viscosity <- c(23, 19, 17, 18,
               24, 26, 21, 14, 18)
n <- length(viscosity)

# Step 1:
x.avg <- mean(viscosity)

# Step 5: Verify the data are normal
library(car)
qqPlot(viscosity)

# Step 6:
x.sd <- sd(viscosity)

# Step 7: t-distribution
dof <- n - 1

# Step 8:
conf.level <- 0.95

# Can be calculated at either
# the lower tail
c.t <- qt(p = (1-conf.level)/2,
          df = dof)

# or the upper tail
c.t <- qt(p = 1-(1-conf.level)/2,
          df = dof)

LB <- x.avg - c.t * x.sd / sqrt(n)
UB <- x.avg + c.t * x.sd / sqrt(n)
paste0('The ', round(conf.level*100, 0),
       '% confidence interval is: ')
paste0('[', round(LB, 1), '; ', round(UB, 1), ']')
```

Comparing the answers for parts 4 and 8 we see the interval, for the same level of 95% certainty, is wider when we have to estimate the standard deviation. This makes sense: the standard deviation is an estimate (meaning there is error in that estimate) of the true standard deviation. That uncertainty must propagate, leading to a wider interval within which we expect to locate the true population viscosity, μ.

We will interpret confidence intervals in more detail a *little later on* (page 63).

2.10 Poisson distribution

The Poisson distribution is useful to characterize rare events (number of cell divisions in a small time unit), system failures and breakdowns, or number of flaws on a product (contaminations per cubic millimetre). These are events that have a very small probability of occurring within a given time interval or unit area (e.g. pump failure probability per minute = 0.000002), but there are many opportunities for the event to possibly occur (e.g. the pump runs continuously). A key assumption is that the events must be independent. If one pump breaks down, then the other pumps must not be affected; if one flaw is produced per unit area of the product, then other flaws that appear on the product must be independent of the first flaw.

Let n = number of opportunities for the event to occur. If this is a time-based system, then it would be the number of minutes the pump is running. If it were an area/volume based system, then it might be the number of square inches or cubic millimetres of the product. Let p = probability of the event occurring: e.g. $p = 0.000002$ chance per minute of failure, or $p = 0.002$ of a flaw being produced per

square inch. The rate at which the event occurs is then given by $\eta = np$ and is a count of events per unit time or per unit area. A value for p can be found using long-term, historical data.

There are two important properties:

1. The mean of the distribution for the rate happens to be the rate at which unusual events occur =
   $\eta = np$

2. The variance of the distribution is also η. This property is particularly interesting - state in your own words what this implies.

Formally, the Poisson distribution can be written as $\dfrac{e^{-\eta}\eta^x}{x!}$, with a plot as shown for $\eta = 4$. Please note the lines are only guides, the probability is only defined at the integer values marked with a circle.
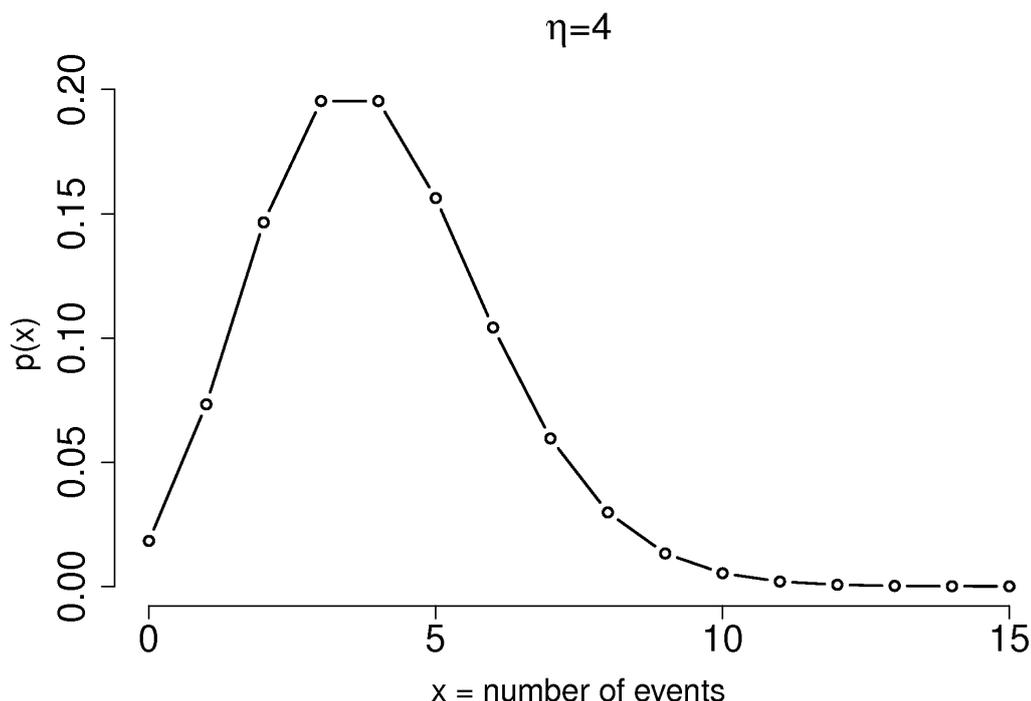
$p(x)$ expresses the probability that there will be x occurrences (must be an integer) of this rare event in the same interval of time or unit area as η was measured.

Example: Equipment in a chemical plant can and will fail. Since it is a rare event, let's use the Poisson distribution to model the failure rates. Historical records on a plant show that a particular supplier's pumps are, on average, prone to failure in a month with probability $p = 0.01$ (1 in 100 chance of failure each month). There are 50 such pumps in use throughout the plant. *What is the probability that* either 0, 1, 3, 6, 10, or 15 *pumps will fail this year?* (Create a table)

$$\eta = 12\,\frac{\text{months}}{\text{year}} \times 50\,\text{pumps} \times 0.01\,\frac{\text{failure}}{\text{month}} = 6\,\frac{\text{pump failures}}{\text{year}}$$

x	$p(x)$
0	0.25% chance
1	1.5%
3	8.9
6	16%
10	4.1%
15	0.1%

```
──────────────────────────── R code ────────────────────────────
x <- c(0, 1, 3, 6, 10, 15)

# Note: R calls the Poisson parameter 'lambda'
dpois(x, lambda=6)

# Output:
# 0.0025 0.0149 0.0892 0.161 0.0413 0.001
```

2.11 Confidence intervals

So far we have calculated point estimates of parameters, called statistics. In the last section in the t-distribution we already calculated a confidence interval. In this section we formalize the idea, starting with an example.

Example: a new customer is evaluating your product, they would like a confidence interval for the impurity level in your sulphuric acid. You can tell them: "*the range from 429ppm to 673ppm contains the true impurity level with 95% confidence*". This is a compact representation of the impurity level. You could have told your potential customer that

- the sample mean from the last year of data is 551 ppm

- the sample standard deviation from the last year of data is 102 ppm

- the last year of data are normally distributed

But a confidence interval conveys a similar concept, in a useful manner. It gives an estimate of the location and spread and uncertainty associated with that parameter (e.g. impurity level in this case).

Let's return to the previous viscosity example, where we had the 9 viscosity measurements `23, 19, 17, 18, 24, 26, 21, 14, 18`. The sample average was $\overline{x} = 20.0$ and the standard deviation was $s = 3.81$. The z-value is: $z = \dfrac{\overline{x} - \mu}{s/\sqrt{n}}$. And we showed this was distributed according to the t-distribution with 8 degrees of freedom.

Calculating a confidence interval requires we find a range within which that z-value occurs. Most often we are interested in symmetrical confidence intervals, so the procedure is:

$$
\begin{array}{ccccc}
-c_t & \leq & z & \leq & +c_t \\
-c_t & \leq & \dfrac{\overline{x} - \mu}{s/\sqrt{n}} & \leq & +c_t \\
\overline{x} - c_t \dfrac{s}{\sqrt{n}} & \leq & \mu & \leq & \overline{x} + c_t \dfrac{s}{\sqrt{n}} \\
\text{LB} & \leq & \mu & \leq & \text{UB}
\end{array}
\tag{2.4}
$$

The critical values of c_t are `qt(1 - 0.05/2, df=8) = 2.306004` when we used the 95% confidence interval (2.5% in each tail). We calculated that LB = 20.0 - 2.92 = 17.1 and that UB = 20.0 +

$2.92 = 22.9$.

2.11.1 Interpreting the confidence interval

- The expression in (2.4) should not be interpreted to mean that the viscosity is 20 units and lies inside the LB (lower-bound) to UB (upper-bound) range of 17.1 to 22.9 with a 95% probability. In fact, the sample mean lies exactly at the mid-point of the range with 100% certainty - that is how the range was calculated.

- What the expression in (2.4) **does imply** is that μ lies in this interval. The confidence interval is a range of possible values for μ, not for $\bar{x}$. Confidence intervals are for parameters, not for statistics.

- Notice that the upper and lower bounds are a function of the data sample used to calculate $\bar{x}$ and the number of points, n. If we take a different sample of data, we will get different upper and lower bounds.

- What does the level of confidence mean?

  It is the probability that the true population viscosity, μ is in the given range. At 95% confidence, it means that 5% of the time the interval *will not contain* the true mean. So if we collected 20 sets of n samples, 19 times out of 20 the confidence interval range **will contain** the true mean, but one of those 20 confidence intervals is expected not to contain the true mean.

- What happens if the level of confidence changes? Calculate the viscosity confidence intervals for 90%, 95%, 99%.

Confidence	LB	UB
90%	17.6	22.4
95%	17.1	22.9
99%	15.7	24.2

  As the confidence level is *increased*, our interval widens, indicating that we have a more reliable region, but it is less precise. With a wider interval we have greater confidence that the true parameter will be inside that region.

  Try it out:

  ```
                                          ─ R code ─
  # Try varying this value:
  conf.level <- 0.90

  viscosity <- c(23, 19, 17, 18,
                 24, 26, 21, 14, 18)
  n <- length(viscosity)
  x.avg <- mean(viscosity)
  x.sd <- sd(viscosity)
  dof <- n - 1
  c.t <- qt(p = 1-(1-conf.level)/2,
            df = dof)
  LB <- x.avg - c.t * x.sd / sqrt(n)
  UB <- x.avg + c.t * x.sd / sqrt(n)
  paste0('The ', round(conf.level*100, 0),
         '% confidence interval is: ')
  paste0('[', round(LB, 1), '; ', round(UB, 1), ']')
  ```

- What happens if the level of confidence is 100%?

The confidence interval is then infinite. We are 100% certain this infinite range contains the population mean, however this is not a useful interval. Test it out in the code above; also try creating an interval with 99.9% confidence, and then 99.99% confidence.

- What happens if we increase the value of n?

  As intuitively expected, as the value of n increases, the confidence interval decreases in width.

- Returning to the case above, where at the 95% level we found the confidence interval was $[17.1; 22.9]$ for the bale's viscosity. What if we were to analyze the bale thoroughly, and found the population viscosity to be 23.2. What is the probability of that occurring?

  Less than 5% of the time.

2.11.2 Confidence interval for the mean from a normal distribution

The aim here is to formalize the calculations for the confidence interval of $\overline{x}$, given a sample of n

a) independent points, taken from

b) the normal distribution.

Be sure to check those two assumptions before going ahead.

There are 2 cases: one where you know the population standard deviation (unlikely), and one where you do not (the usual case). It is safer to use the confidence interval for the case when you do not know the standard deviation, as it is a more conservative (i.e. wider) interval.

The detailed derivation for the two cases was covered in earlier sections.

Case A. Variance is known

When the variance is known, the confidence interval is given by (2.5) below, derived from this z-deviate: $z = \dfrac{\overline{x} - \mu}{\sigma/\sqrt{n}}$ back in the *section on the normal distribution* (page 56).

$$
\begin{array}{ccccc}
-c_n & \leq & z & \leq & +c_n \\
-c_n & \leq & \dfrac{\overline{x} - \mu}{\sigma/\sqrt{n}} & \leq & +c_n \\
\overline{x} - c_n \dfrac{\sigma}{\sqrt{n}} & \leq & \mu & \leq & \overline{x} + c_n \dfrac{\sigma}{\sqrt{n}} \\
\text{LB} & \leq & \mu & \leq & \text{UB}
\end{array}
\tag{2.5}
$$

The values of c_n are `qnorm(1 - 0.05/2) = 1.96` when we happen to use the 95% confidence interval (2.5% in each tail).

Case B. Variance is unknown

In the more realistic case when the variance is unknown we use the equation *derived in the section on the t-distribution* (page 59), and repeated here below. This is derived from the z-deviate: $z = \dfrac{\overline{x} - \mu}{s/\sqrt{n}}$:

$$
\begin{array}{ccccc}
-c_t & \leq & z & \leq & +c_t \\[4pt]
-c_t & \leq & \dfrac{\overline{x} - \mu}{s/\sqrt{n}} & \leq & +c_t \\[4pt]
\overline{x} - c_t \dfrac{s}{\sqrt{n}} & \leq & \mu & \leq & \overline{x} + c_t \dfrac{s}{\sqrt{n}} \\[4pt]
\text{LB} & \leq & \mu & \leq & \text{UB}
\end{array}
\tag{2.6}
$$

The values of c_t are `qt(1 - 0.05/2, df=...)` when we use the 95% confidence interval (2.5% in each tail). This z-deviate is distributed according to the t-distribution, since we have additional uncertainty when using the standard deviation estimate, s, instead of the population standard deviation, σ.

Comparison

If we have the fortunate case where our estimated variance, s^2, is equal to the population variance, σ^2, then we can compare the 2 intervals in equations (2.5) and (2.6). The only difference would be the value of the c_n from the normal distribution and c_t from the t-distribution. For typical values used as confidence levels, 90% to 99.9%, values of $c_t > c_n$ for any degrees of freedom.

This implies the confidence limits are wider for the case when the standard deviation is unknown, leading to more conservative results, reflecting our uncertainty of the standard deviation parameter, σ.

2.12 Testing for differences and similarity

Video for this section.

These sort of questions often arise in data analysis:

- We want to change to a cheaper material, B. Does it work as well as A?

- We want to introduce a new catalyst B. Does it improve our product properties over the current catalyst A?

Either we want to confirm things are statistically the same, or confirm they have changed. Notice that in both the above cases we are testing the population mean (location). Has the mean shifted or is it the same? There are also tests for changes in variance (spread), which we will cover. We will work with an example throughout this section.

Example: A process operator needs to verify that a new form of feedback control on the batch reactor leads to improved yields. Yields under the current control system, A, are compared with yields under the new system, B. The last ten runs with system A are compared to the next 10 sequential runs with system B. The data are shown in the table, and shown in graphical form as well. (Note that the box plot uses the median, while the plots on the right show the mean.)

Batch yield (%) for two trials | A | B
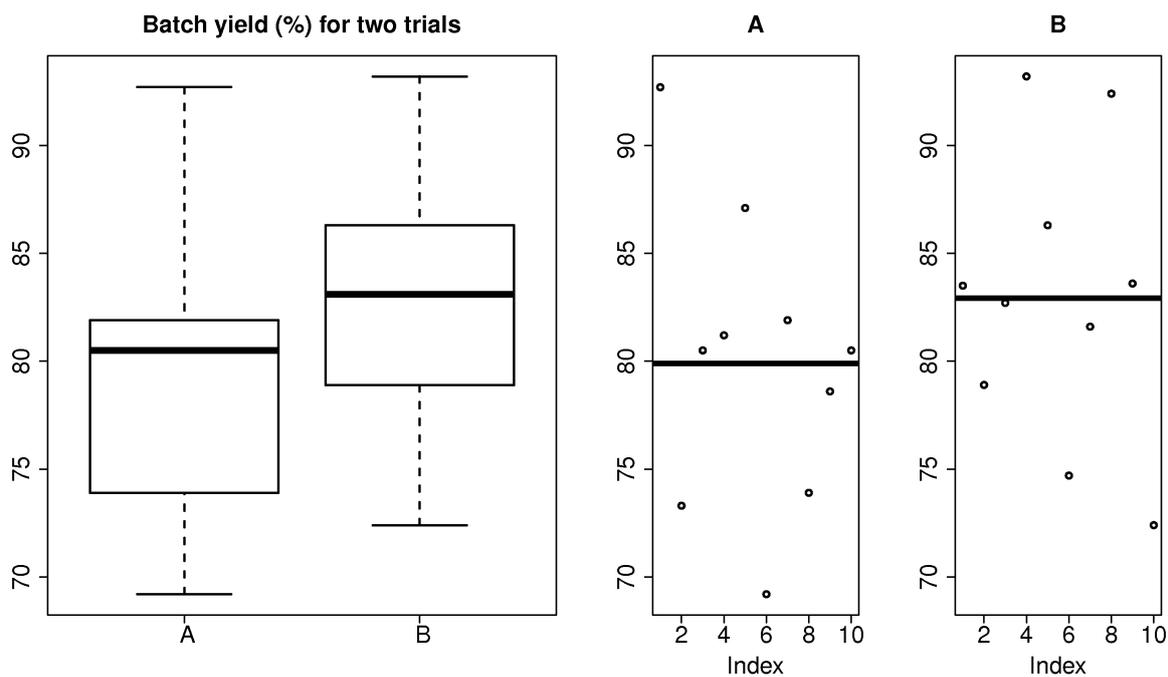
Experiment number	Feedback system	Yield	Experiment number	Feedback system	Yield
1	A	92.7	11	B	83.5
2	A	73.3	12	B	78.9
3	A	80.5	13	B	82.7
4	A	81.2	14	B	93.2
5	A	87.1	15	B	86.3
6	A	69.2	16	B	74.7
7	A	81.9	17	B	81.6
8	A	73.9	18	B	92.4
9	A	78.6	19	B	83.6
10	A	80.5	20	B	72.4
Mean		79.89	Mean		82.93
Standard deviation		6.81	Standard deviation		6.70

```
                                         R code
# Generate the boxplot
A <- c(92.7, 73.3, 80.5, 81.2, 87.1,
       69.2, 81.9, 73.9, 78.6, 80.5)
B <- c(83.5, 78.9, 82.7, 93.2, 86.3,
       74.7, 81.6, 92.4, 83.6, 72.4)

data.A <- data.frame(observe=A, method='A')
data.B <- data.frame(observe=B, method='B')
data <- rbind(data.A, data.B)

limits <- range(data$observe)
boxplot(data$obs ~ data$method, lwd=2,
        main="Batch yield (%) for two trials")
```

We address the question of whether or not there was a *significant difference* between system A and B. A significant difference means that when system B is compared to a suitable reference, that we can be

sure that the long run implementation of B will lead, in general, to a different yield (%). We want to be sure that any change in the 10 runs under system B were not *only due to chance*, because system B will cost us $100,000 to install, and $20,000 in annual software license fees.

> *Note*: those with a traditional statistical background will recognize this section as one-sided hypothesis tests. We will only consider tests for a significant increase or decrease, i.e. one-sided tests, in this section. We use confidence intervals, rather than hypothesis tests; the results are exactly the same. Arguably the confidence interval approach is more interpretable, since we get a bound, rather that just a clear-cut yes/no answer.

There are two main ways to test for a significant increase or significant decrease.

2.12.1 Comparison to a long-term reference set

Continuing the above example we can compare the past 10 runs from system B with the 10 runs from system A. The average difference between these runs is $\overline{x}_B - \overline{x}_A = 82.93 - 79.89 = 3.04$ units of improved yield. Now, if we have a long-term reference data set available, we can compare if any 10 historical, sequential runs from system A, followed by another 10 historical, sequential runs under system A had a difference that was this great. If not, then we know that system B leads to a definite improvement, not likely to be caused by chance alone.

Here's the procedure:

1. Imagine that we have 300 historical data points from this system, tabulated in time order: yield from batch 1, 2, 3 ... (the data are available on the website[34]).

2. Calculate the average yields from batches 1 to 10. Then calculate the average yield from batches 11 to 20. Notice that this is exactly like the experiment we performed when we acquired data for system B: two groups of 10 batches, with the groups formed from sequential batches.

3. Now subtract these two averages: (group average 11 to 20) minus (group average 1 to 10).

4. Repeat steps 2 and 3, but use batches 2 to 11 and 12 to 21. Repeat until all historical batch data are used up, i.e. batches 281 to 290 and 291 to 300. The plot below can be drawn, one point for each of these difference values.

Difference between means of 2 adjacent groups (10 batches per group)
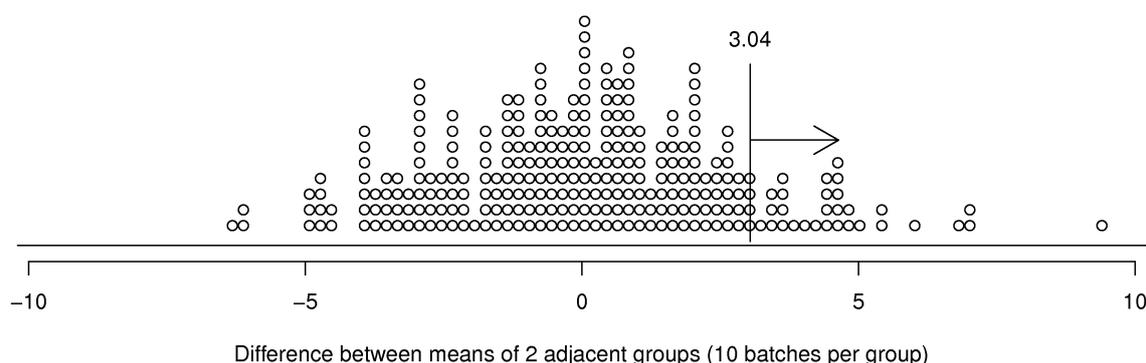
The vertical line at 3.04 is the difference value recorded between system B and system A. From this we can see that historically, there were 31 out of 281 batches, about 11% of historical data, that had a difference value of 3.04 or greater. So there is a 11% probability that system B was better than system A purely by chance, and not due to any technical superiority. Given this information, we can now judge, if the improved control system will be economically viable and judge, based on internal company criteria, if this is a suitable investment, also considering the 11% risk that our investment will fail.

---

[34] http://openmv.net/info/batch-yields

Notice that no assumption of independence or any form of distributions was required for this work! The only assumption made is that the historical data are relevant. We might know this if, for example, no substantial modification was made to the batch system for the duration over which the 300 samples were acquired. If however, a different batch recipe were used for sample 200 onwards, then we may have to discard those first 200 samples: it is not fair to judge control system B to the first 200 samples under system A, when a different operating procedure was in use.

So to summarize: we can use a historical data set if it is relevant. And there are no assumptions of independence or shape of the distribution, e.g. a normal distribution.

In fact, for this example, the data were not independent, they were autocorrelated. There was a relationship from one batch to the next: $x[k] = \phi x[k-1] + a[k]$, with $\phi = -0.3$, and $a[k] \sim \mathcal{N}\left(\mu = 0, \sigma^2 = 6.7^2\right)$. As an aside you can simulate your own set of autocorrelated data using this R code:

R code

```
N <- 300
phi <- -0.3
spread <- 6.7
location <- 79.9

# create a vector of zeros
A.hist <- numeric(N)
for (k in 2:N)
{
  A.hist[k] <- phi*(A.hist[k-1]) +
          rnorm(1, mean = 0, sd = spread)
}
A.hist <- A.hist + location

# Note: your plot will look different to
# the text, because it will be from a
# different set of random numbers
title = paste0("Autocorrelation between ",
       "successive values of batch yield")
plot(A.hist[1:N-1], A.hist[2:N],
     xlab = "x[k]",
     ylab = "x[k+1]",
     main = title,
     lwd = 3,
     xlim = c(60,100),
     ylim = c(60,100))

lines(lowess(A.hist[1:N-1], A.hist[2:N]))

# Hint: run the code several times, with
# different values of variable `phi`.
```

We can visualize this autocorrelation by plotting the values of $x[k]$ against $x[k+1]$:

Autocorrelation between successive values of batch yield
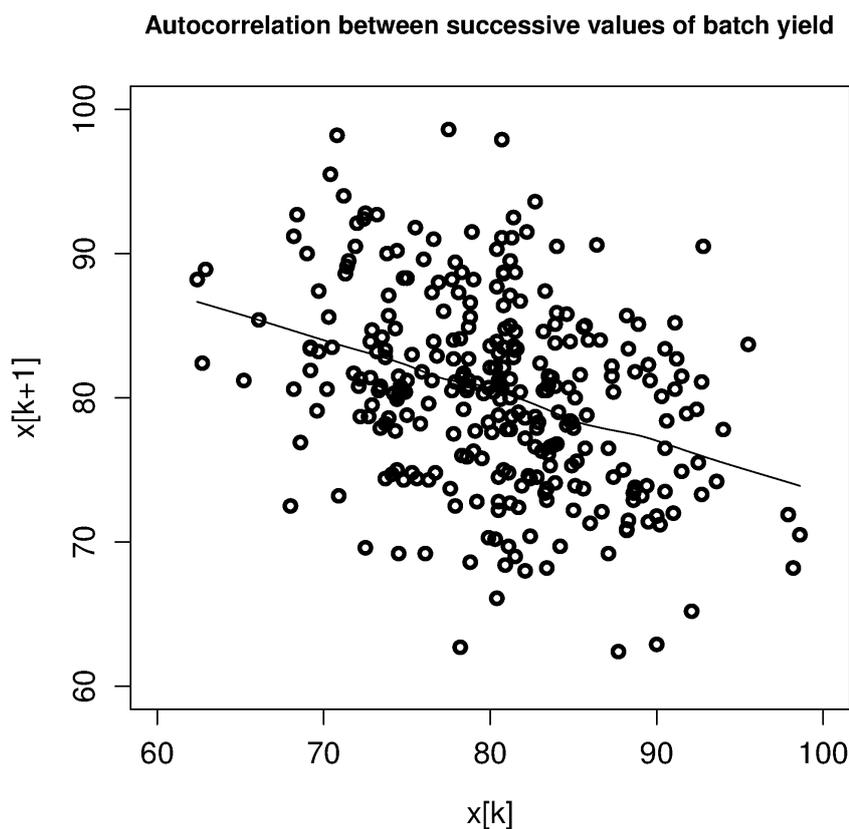
We can immediately see the data are **not independent**, because the slope is non-zero.

2.12.2 Comparison when a reference set is not available

A reference data set may not always be available; we may only have the data from the 20 experimental runs (10 from system A and 10 from B) and nothing else. We can proceed to compare the data, but we will require a strong assumption of random sampling (independence), which is often not valid in engineering data sets. Fortunately, engineering data sets are usually large - we are good at collecting data - so the methodology in the preceding section on using a reference set, is greatly preferred, when possible.

How could the assumption of independence (random sampling) be made more realistically? How is the lack of independence detrimental? We show below that the assumption of independence is made twice: the samples within group A and B must be independent; furthermore, the samples between the groups should be independent. But first we have to understand why the assumption of independence is required, by understanding the usual approach for estimating if differences are significant or not.

The usual approach for assessing if the difference between $\bar{x}_B - \bar{x}_A$ is significant follows this approach:

1. Assume the data for sample A and sample B have been independently sampled from their respective populations.

2. Assume the data for sample A and sample B have the same population variance, $\sigma_A = \sigma_B = \sigma$ (there is a test for this, see the next section).

3. Let the sample A have population mean μ_A and sample B have population mean μ_B.

4. From the central limit theorem (this is where the assumption of independence of the samples within each group comes), we know that:

$$\mathcal{V}\{\overline{x}_A\} = \frac{\sigma_A^2}{n_A} \qquad\qquad \mathcal{V}\{\overline{x}_B\} = \frac{\sigma_B^2}{n_B}$$

5. Assuming independence again, but this time between groups, this implies the average of each sample group is independent, i.e. $\overline{x}_A$ and $\overline{x}_B$ are independent of each other. This allows us to write:

$$\mathcal{V}\{\overline{x}_B - \overline{x}_A\} = \frac{\sigma^2}{n_A} + \frac{\sigma^2}{n_B} = \sigma^2\left(\frac{1}{n_A} + \frac{1}{n_B}\right) \tag{2.7}$$

6. Using the central limit theorem, even if the samples in A and the samples in B are non-normal, the sample averages $\overline{x}_A$ and $\overline{x}_B$ will be more normal as the sample size becomes progressively larger. So the difference between these means will also be more normal: $\overline{x}_B - \overline{x}_A$. Now express this difference in the form of a z-deviate (standard form):

$$z = \frac{(\overline{x}_B - \overline{x}_A) - (\mu_B - \mu_A)}{\sqrt{\sigma^2\left(\frac{1}{n_A} + \frac{1}{n_B}\right)}} \tag{2.8}$$

We could ask, what is the probability of seeing a z value from equation (2.8) of that magnitude? Recall that this z-value is the equivalent of $\overline{x}_B - \overline{x}_A$, expressed in deviation form, and we are interested if this difference is due to chance. So we should ask, what is the probability of getting a value of z **greater** than this, or **smaller** that this, depending on the case?

The only question remains is what is a suitable value for σ? As we have seen before, when we have a large enough reference set, then we can use the value of σ from the historical data, called an *external estimate*. Or we can use an *internal estimate* of spread; both approaches are discussed below.

Now we know the approach required, using the above 6 steps, to determine if there was a significant difference. And we know the assumptions that are required: normally distributed and independent samples. But how can we be sure our data are independent? This is the most critical aspect, so let's look at a few cases and discuss, then we will return to our example and calculate the z-values with both an *external* and *internal* estimate of spread.

Discuss whether these experiments would lead to independent data or not, and how we might improve the situation.

a) We are testing a new coating to repel moisture. The coating is applied to packaging sheets that are already hydrophobic, however this coating enhances the moisture barrier property of the sheet. In the lab, we take a large packaging sheet and divide it into 16 blocks. We coat the sheet as shown in the figure and then use the $n_A = 8$ and $n_B = 8$ values of hydrophobicity to judge if coating B is better than coating A.

Some problems with this approach:

- The packaging sheet to which the new coating is applied may not be uniform. The sheet is already hydrophobic, but the hydrophobicity is probably not evenly spread over the sheet, nor are any of the other physical properties of the sheet. When we measure the moisture repelling property with the different coatings applied, we will not have an accurate measure of whether coating A or B worked better. We must randomly assign blocks A and B on the packaging sheet.

- Even so, this may still be inadequate, because what if the packaging sheet selected has overly high or low hydrophobicity (i.e. it is not representative of regular packaging sheets). What should be done is that random packaging sheets should be selected, and they should be selected across different lots from the sheet supplier (sheets within one lot are likely to be more similar than between lots). Then on each sheet we apply coatings A and B, in a random order on each sheet.

- It is tempting to apply coating A and B to one half of the various sheets and measure the *difference* between the moisture repelling values from each half. It is tempting because this approach would cancel out any base variation between difference sheets, as long as that variation is present across the entire sheet. Then we can go on to assess if this difference is significant.

  There is nothing wrong with this methodology, however, there is a different, specific test for paired data, covered in a *later section* (page 76). If you use the above test, you violate the assumption in step 5, which requires that $\overline{x}_A$ and $\overline{x}_B$ be independent. Values within group A and B are independent, but not their sample averages, because you cannot calculate $\overline{x}_A$ and $\overline{x}_B$ independently.

b) We are testing an alternative, cheaper raw material in our process, but want to be sure our product's final properties are unaffected. Our raw material dispensing system will need to be modified to dispense material B. This requires the production line to be shut down for 15 hours while the new dispenser, lent from the supplier, is installed. The new supplier has given us 8 representative batches of their new material to test, and each test will take 3 hours. We are inclined to run these 8 batches over the weekend: set up the dispenser on Friday night (15 hours), run the tests from Saturday noon to Sunday noon, then return the line back to normal for Monday's shift. How might we violate the assumptions required by the data analysis steps above when we compare 8 batches of material A (collected on Thursday and Friday) to the 8 batches from material B (from the weekend)? What might we do to avoid these problems?

  - The 8 tests are run sequentially, so **any changes** in conditions between these 8 runs and the 8 runs from material A will be confounded (confused) in the results. List some actual scenarios how

---

confounding between the weekday and weekend experiments occur:

- For example, the staff running the equipment on the weekend are likely not the same staff that run the equipment on weekdays.

- The change in the dispenser may have inadvertently modified other parts of the process, and in fact the dispenser itself might be related to product quality.

- The samples from the tests will be collected and only analyzed in the lab on Monday, whereas the samples from material A are usually analyzed on the same day: that waiting period may degrade the sample.

This confounding with all these other, potential factors means that we will not be able to determine whether material B caused a true difference, or whether it was due to the other conditions.

- It is certainly expensive and impractical to randomize the runs in this case. Randomization would mean we randomly run the 16 tests, with the A and B chosen in random order, e.g. `A B A B A A B B A A B B B A B A`. This particular randomization sequence would require changing the dispenser 9 times.

- One suboptimal sequence of running the system is `A A A A B B B B A A A A B B B B`. This requires changing the dispenser 4 times (one extra change to get the system back to material A). We run each (`A A A A B B B B`) sequence on two different weekends, changing the operating staff between the two groups of 8 runs, making sure the sample analysis follows the usual protocols: so we reduce the chance of confounding the results.

Randomization might be expensive and time-consuming in some studies, but it is the insurance we require to avoid being misled. These two examples demonstrate this principle: **block what you can and randomize what you cannot**. We will review these concepts again in the *design and analysis of experiments section* (page 231). If the change being tested is expected to improve the process, then we must follow these precautions to avoid a process upgrade/modification that does not lead to the expected improvement; or the the converse - a missed opportunity of implementing a change for the better.

External and internal estimates of spread

So to recap the progress so far, we are aiming to test if there is a *significant, long-term difference* between two systems: A and B. We showed the most reliable way to test this difference is to compare it with a body of historical data, with the comparison made in the same way as when the data from system A and B were acquired; this requires no additional assumptions, and even allows one to run experiments for system B in a **non-independent** way.

But, because we do not always have a large and relevant body of data available, we can calculate the difference between A and B and test if this difference could have occurred by chance alone. For that we use equation (2.8), but we need an estimate of the spread, σ.

External estimate of spread

The question we turn to now is what value to use for σ in equation (2.8). We got to that equation by assuming we have no historical, external data. But what if we did have some external data? We could at least estimate σ from that. For example, the 300 historical batch yields has $\sigma = 6.61$:

Check the probability of obtaining the z-value in (2.8) by using the hypothesis that the value $\mu_B - \mu_A = 0$. In other words we are making a statement, or a test of significance. Then we calculate

this z-value and its associated *cumulative probability*:

$$z = \frac{(\overline{x}_B - \overline{x}_A) - (\mu_B - \mu_A)}{\sqrt{\sigma^2 \left(\dfrac{1}{n_A} + \dfrac{1}{n_B} \right)}}$$

$$z = \frac{(82.93 - 79.89) - (\mu_B - \mu_A)}{\sqrt{6.61^2 \left(\dfrac{1}{10} + \dfrac{1}{10} \right)}}$$

$$z = \frac{3.04 - 0}{2.956} = \mathbf{1.03}$$

The probability of seeing a z-value from $-\infty$ up to 1.03 is 84.8% (use the `pnorm(1.03)` function in R). But we are interested in the probability of obtaining a z-value **larger** than this. Why? Because $z = 0$ represents no improvement, and a value of $z < 0$ would mean that system B is worse than system A. So what are the chances of obtaining $z = 1.03$? It is (100-84.8)% = 15.2%, which means that system B's performance could have been obtained by pure luck in 15.2% of cases.

```
──────────────────────────────── R code ────────────────────────────────
A <- c(92.7, 73.3, 80.5, 81.2, 87.1,
       69.2, 81.9, 73.9, 78.6, 80.5)
B <- c(83.5, 78.9, 82.7, 93.2, 86.3,
       74.7, 81.6, 92.4, 83.6, 72.4)

xA.avg <- mean(A)
xB.avg <- mean(B)
n.A <- length(A)
n.B <- length(B)
sigma.external <- 6.61 # given

den <- sigma.external**2 * (1/n.A + 1/n.B)
z <- (xB.avg - xA.avg) / sqrt(den)

# Probability of this z?
# We have normalized to zero mean
# and to unit standard deviation:
p <- pnorm(z, mean=0, sd=1)  # 0.8481164
paste0('Probability by chance: ',
        round((1-p)*100, 1), '%')
```

We interpret this number of "15.2%" in the summary section, but let's finally look at what happens if we have no historical data - then we generate an *internal* estimate of σ from the 20 experimental runs alone.

Internal estimate of spread

The sample variance from each system was $s_A^2 = 6.81^2$ and $s_B^2 = 6.70^2$, and in this case it happened that $n_A = n_B = 10$, although the sample sizes do not necessarily have to be the same.

If the variances are comparable (there is a *test for that below* (page 78)), then we can calculate a *pooled variance*, s_P^2, which is a weighted sum of the sampled variances:

$$s_P^2 = \frac{(n_A - 1)s_A^2 + (n_B - 1)s_B^2}{n_A - 1 + n_B - 1}$$

$$s_P^2 = \frac{9 \times 6.81^2 + 9 \times 6.70^2}{18}$$

$$s_P^2 = 45.63$$

Now using this value of s_P instead of σ in (2.8):

$$
\begin{aligned}
z &= \frac{(\overline{x}_B - \overline{x}_A) - (\mu_B - \mu_A)}{\sqrt{\sigma^2 \left(\dfrac{1}{n_A} + \dfrac{1}{n_B}\right)}} \\
&= \frac{(82.93 - 79.89) - (\mu_B - \mu_A)}{\sqrt{s_P^2 \left(\dfrac{1}{10} + \dfrac{1}{10}\right)}} \\
&= \frac{3.04 - 0}{\sqrt{45.63 \times 2/10}} \\
z &= \mathbf{1.01}
\end{aligned}
$$

The probability of obtaining a z-value greater than this can be calculated as 16.4% using the t-distribution with 18 degrees of freedom (use `1-pt(1.01, df=18)` in R). We use a t-distribution because an estimate of the variance is used, s_p^2, not a population variance, σ^2.

```R
A <- c(92.7, 73.3, 80.5, 81.2, 87.1,
       69.2, 81.9, 73.9, 78.6, 80.5)
B <- c(83.5, 78.9, 82.7, 93.2, 86.3,
       74.7, 81.6, 92.4, 83.6, 72.4)

xA.avg <- mean(A)
xB.avg <- mean(B)
n.A <- length(A)
n.B <- length(B)
# degrees of freedom
dof <- n.A - 1 + n.B - 1
var.pooled <- ((n.A - 1) * var(A) +
          (n.B - 1) * var(B)) / dof

den <- var.pooled * (1/n.A + 1/n.B)
z <- (xB.avg - xA.avg) / sqrt(den)

# Probability of this z?
# Compare it against the t-distribution:
p <- pt(z, df = dof)  # 0.8361346

paste0('Probability by chance: ',
       round((1-p)*100, 1), '%')
```

As an aside: we used a normal distribution for the external σ and a t-distribution for the internal s. Both cases had a similar value for z (compare $z = 1.01$ to $z = 1.03$). Note however that the probabilities are higher in the t-distribution's tails, which means that even though we have similar z-values, the probability is greater: 16.4% against 15.2%. While this difference is not much from a practical point of view, it illustrates the difference between the t-distribution and the normal distribution.

The results from this section were achieved by only using the 20 experimental runs, no external data. However, it made some strong assumptions:

- The variances of the two samples are comparable, and can *therefore be pooled* (page 78) to provide an estimate of σ.

- The usual assumption of independence within each sample is made (which we know not to be true for many practical engineering cases).

- The assumption of independence between the samples is also made (this is more likely to be true in this example, because the first runs to acquire data for A are not likely to affect the runs for system B).

2.12. Testing for differences and similarity 75

- Each sample, A and B, is assumed to be normally distributed.

Summary and comparison of methods

Let's compare the 3 estimates. Recall our aim is to convince ourself/someone that system B will have better long-term performance than the current system A.

If we play devil's advocate, our *null hypothesis* is that system B has no effect. Then it is up to us to prove, convincingly, that the change from A to B has a systematic, permanent effect. That is what the calculated probabilities represent :, the probability of us being wrong.

1. Using only reference data: 11% (about 1 in 10)

2. Using the 20 experimental runs, but an external estimate of σ: 15.2% (about 1 in 7)

3. Using the 20 experimental runs only, no external data: 16.4% (about 1 in 6)

The reference data method shows that the trial with 10 experiments using system B could have actually been taken from the historical data with a chance of 11%. A risk adverse company may want this number to be around 5%, or as low as 1% (1 in 100), which essentially guarantees the new system will have better performance.

When constructing the reference set, we have to be sure the reference data are appropriate. Were the reference data acquired under conditions that were similar to the time in which data from system B were acquired? In this example, they were, but in practice, careful inspection of plant records must be made to verify this.

The other two methods mainly use the experimental data, and provide essentially the same answer *in this case study*, though that is not always the case. The main point here is that our experimental data are usually not independent. However, by careful planning, and expense, we can meet the requirement of independence by randomizing the order in which we acquire the data. Randomization is the insurance (cost) we pay so that we do not have to rely on a large body of prior reference data. But in some cases it is not possible to randomize, so blocking is required. More on blocking in the *design of experiments section* (page 275).

2.13 Paired tests

Video for this section.

A paired test is a test that is run twice on the same object or batch of materials. You might see the nomenclature of "two treatments" being used in the literature. For example:

- A drug trial could be run in two parts: each person randomly receives a placebo or the drug, then 3 weeks later they receive the opposite, for another 3 weeks. Tests are run at 3 weeks and 6 weeks and the difference in the test result is recorded.

- We are testing two different additives, A and B, where the additive is applied to a base mixture of raw materials. Several raw material lots are received from various suppliers, supposedly uniform. Split each lot into 2 parts, and run additive A and B on each half. Measure the outcome variable, e.g. conversion, viscosity, or whatever the case might be, and record the difference.

- We are testing a new coating to repel moisture. The coating is applied to randomly selected sheets in a pattern [A | B] or [B | A] (the pattern choice is made randomly). We measure the repellent property value and record the difference.

In each case we have a table of n samples recording the **difference values**. The question now is whether the difference is significant, or is it essentially zero?

The advantage of the paired test is that any systematic error in our measurement system, what ever it might be, is removed as long as that error is consistent. Say for example we are measuring blood pressure, and the automated blood pressure device has a bias of -5 mmHg. This systematic error will cancel out when we subtract the 2 test readings. In the example of the raw materials and additives: any variation in the raw materials and its (unintended) effect on the outcome variable of interest will be cancelled.

The disadvantage of the paired test is that we lose degrees of freedom. Let's see how:

1. Calculate the n differences: $w_1 = x_{B,1} - x_{A,1}$; $w_2 = x_{B,2} - x_{A,2}$, ... to create the sample of values $\mathbf{w} = [w_1, w_2, \ldots, w_n]$

2. Assume these values, w_i, are independent, because they are taken on independent objects (people, base packages, sheets of paper, *etc*)

3. Calculate the mean, $\overline{w}$ and the standard deviation, s_w, of these n difference values.

4. What do we need to assume about the population from which w comes? Nothing. We are not interested in the w values, we are interested in $\overline{w}$. OK, so what distribution would values of $\overline{w}$ come from? By the central limit theorem, the $\overline{w}$ values should be normally distributed as $\overline{w} \sim \mathcal{N}\left(\mu_w, \sigma_w^2/n\right)$, where $\mu_w = \mu_{A-B}$.

5. Now calculate the z-value, but use the sample standard deviation, instead of the population standard deviation.

$$z = \frac{\overline{w} - \mu_w}{s_w/\sqrt{n}}$$

6. Because we have used the sample standard deviation, s_w, we have to use to the t-distribution with $n - 1$ degrees of freedom, to calculate the critical values.

7. We can calculate a confidence interval, below, and if this interval includes zero, then the change from treatment A to treatment B had no effect.

$$\overline{w} - c_t \frac{s_w}{\sqrt{n}} < \mu_w < \overline{w} + c_t \frac{s_w}{\sqrt{n}}$$

The value of c_t is taken from the t-distribution with $n - 1$ degrees of freedom at the level of confidence required: use the `qt(...)` function in R to obtain the values of c_t.

The loss of degrees of freedom can be seen when we use exactly the same data and treat the problem as one where we have n_A and n_B samples in groups A and B and want to test for a difference between μ_A and μ_B. You are encouraged to try this out. There are more degrees of freedom, $n_A + n_B - 2$ in fact when we use the t-distribution with the *pooled variance shown here* (page 74). Compare this to the case just described above where there are only n degrees of freedom.

2.14 Other types of confidence intervals

There are several other confidence intervals that you might come across in your career. We merely mention them here and don't cover their derivation. What is important is that you understand *how* to interpret a confidence interval. Hopefully the previous discussion achieved that.

2.14.1 Confidence interval for the variance

This confidence interval finds a region in which the normal distribution's variance parameter, σ, lies. The range is obviously positive, since variance is a positive quantity. For reference, this range is:

$$\left[\frac{(n-1)S^2}{\chi^2_{n-1,\alpha/2}} \quad \text{to} \quad \frac{(n-1)S^2}{\chi^2_{n-1,1-\alpha/2}} \right]$$

- n is the number of samples
- S^2 is the sample variance
- $\chi^2_{n-1,\alpha/2}$ are values from the χ^2 distribution with $n-1$ and $\alpha/2$ degrees of freedom
- $1 - \alpha$: is the level of confidence, usually 95%, so $\alpha = 0.05$ in that case.

2.14.2 Confidence interval for the ratio of two variances

One way to test whether we can pool (combine) two variances, taken from two different normal distributions, is to construct the ratio: $\frac{s_1^2}{s_2^2}$. We can construct a confidence interval, and if this interval contains the value of 1.0, then we have no evidence to presume they are different (i.e. we can assume the two population variances are similar).

$$F_{\alpha/2,\nu_1,\nu_2} \frac{s_2^2}{s_1^2} \quad < \quad \frac{\sigma_2^2}{\sigma_1^2} \quad < \quad F_{1-\alpha/2,\nu_1,\nu_2} \frac{s_2^2}{s_1^2}$$

where we use $F_{\alpha/2,\nu_1,\nu_2}$ to mean the point along the cumulative F-distribution which has area of $\alpha/2$ using ν_1 degrees of freedom for estimating s_1 and ν_2 degrees of freedom for estimating s_2. For example, in R, the value of $F_{0.05/2,10,20}$ can be found from `qf(0.025, 10, 20)` as 0.2925. The point along the cumulative F-distribution which has area of $1 - \alpha/2$ is denoted as $F_{1-\alpha/2,\nu_1,\nu_2}$, and α is the level of confidence, usually $\alpha = 0.05$ to denote a 95% confidence level.

2.14.3 Confidence interval for proportions: the binomial proportion confidence interval

Sometimes we measure the proportion of successes (passes). For example, if we take a sample of n independent items from our production line, and with an inspection system we can judge pass or failure. The proportion of passes is what is important, and we wish to construct a confidence region for the population *proportion*. This allows one to say the population proportion of passes lies between the given range. As in *the proportion of packaged pizzas with 20 or more pepperoni slices is between 86 and 92%.*

Incidentally, it is this confidence interval that is used in polls to judge the proportion of people that prefer a political party. One can run this confidence interval backwards and ask: how many independent people do I need to poll to achieve a population proportion that lies within a range of $\pm 2\%$, 19 times out of 20? The answer actually is function of the poll result! But the worst case scenario is a split-poll, and that requires 2400 respondents.

2.15 Statistical tables for the normal- and t-distribution

Normal distribution
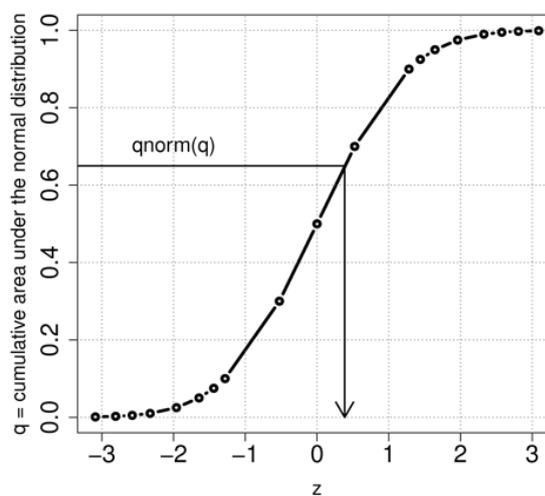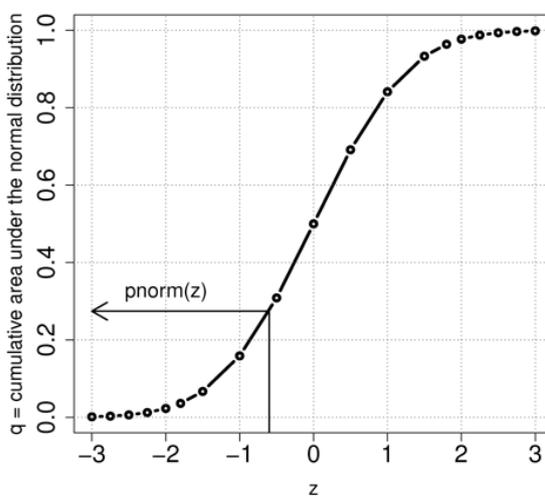
z	q = cumulative area under the normal distribution
-3.00	0.001350
-2.75	0.002980
-2.50	0.006210
-2.25	0.01222
-2.00	0.02275
-1.80	0.03593
-1.50	0.06681
-1.00	0.1587
-0.50	0.3085
0.00	**0.5**
0.50	0.6915
1.00	0.8413
1.50	0.9332
1.80	0.9641
2.00	0.9773
2.25	0.9878
2.50	0.9938
2.75	0.9970
3.00	0.9987

q = cumulative area under the normal distribution	z
0.001	-3.090
0.0025	-2.807
0.005	-2.576
0.01	-2.326
0.025	-1.960
0.05	-1.645
0.075	-1.440
0.1	-1.282
0.3	-0.5244
0.5	**0.0**
0.7	0.5244
0.9	1.282
0.925	1.440
0.95	1.645
0.975	1.960
0.99	2.326
0.995	2.576
0.9975	2.807
0.999	3.090

t distribution
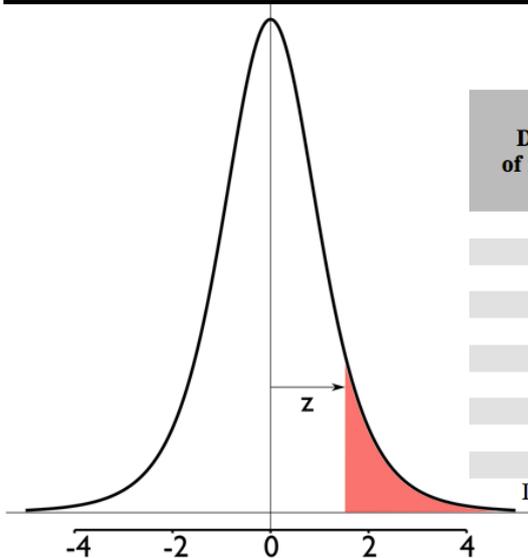
Degrees of freedom	z-value when area under the tail is						
	0.4	0.25	0.1	0.05	0.025	0.01	0.005
1	0.325	1.000	3.08	6.31	12.7	31.8	63.7
2	0.289	0.816	1.89	2.92	4.30	6.97	9.92
3	0.277	0.765	1.64	2.35	3.18	4.54	5.84
4	0.271	0.741	1.53	2.13	2.78	3.75	4.60
5	0.267	0.727	1.48	2.02	2.57	3.37	4.03
10	0.260	0.700	1.37	1.81	2.23	2.76	3.17
15	0.258	0.691	1.34	1.75	2.13	2.60	2.95
20	0.257	0.687	1.33	1.72	2.09	2.53	2.85
30	0.256	0.683	1.31	1.70	2.04	2.46	2.75
60	0.254	0.679	1.30	1.67	2.00	2.39	2.66
Infinite	0.253	0.674	1.28	1.64	1.96	2.33	2.58

If interested, here is the code used to generate these figures

```
                                        ─── R code ───
# The source code used to generate the
# *normal distribution* section:
q <- c(seq(-3.0, -2.0, 0.25),
       c(-1.8, -1.5, -1.0, -0.5, 0, 0.5,
         1.0, 1.5, 1.8),
       seq(2.0, 3.0, 0.25))
cumulative.quantile = pnorm(q)


p <- c(0.001, 0.0025, 0.005, 0.010, 0.025,
       0.05, 0.075, 0.10, 0.3, 0.5, 0.7,
       0.9, 0.925, 0.950, 0.975, 0.99,
       0.995, 0.9975, 0.999)
cumulative.probability = qnorm(p)

layout(matrix(c(1,2), 1, 2))
par(mar = c(4.2, 4.2, 0.2, 1))
plot(q, cumulative.quantile,
     type = "b",
         main = "",
         xlab = "z",
         ylab = "q = cumulative area under the normal distribution",
         cex.lab = 1.4,
         cex.main = 1.8,
         lwd = 4,
         cex.sub = 1.8,
         cex.axis = 1.8,
         ylim = c(0, 1))
grid(col="gray30")
a1 = -0.6
arrows(a1, y = -0.2, x1 = a1,
       y1 = pnorm(a1),
       code = 0, lwd = 2)
arrows(a1, y = pnorm(a1), x1 = -3,
       y1 = pnorm(a1), code = 2, lwd = 2)
text(-2, pnorm(a1) + 0.05, "pnorm(z)",
     cex = 1.5)

plot(cumulative.probability, p,
     type = "b",
         main = "",
         xlab = "z",
         ylab = "q = cumulative area under the normal distribution",
         cex.lab = 1.4,
         cex.main = 1.8,
         lwd = 4,
         cex.sub = 1.8,
         cex.axis = 1.8,
         ylim = c(0, 1))
grid(col = "gray30")
a1 = qnorm(0.65)
arrows(a1, y = 0, x1 = a1,
       y1 = pnorm(a1), code = 1, lwd = 2)
arrows(a1, y=pnorm(a1), x1 = -5,
       y1 = pnorm(a1), code = 0, lwd = 2)
text(-2, pnorm(a1)+0.05, "qnorm(q)",
    cex = 1.5)


# The source code used to generate the t-distribution section:
dof <- c(1, 2, 3, 4, 5, 10, 15, 20,
         30, 60, Inf)
tail.area.oneside <- c(0.4, 0.25, 0.1,
             0.05, 0.025, 0.01, 0.005)

n.dof <- length(dof)
n.tails <- length(tail.area.oneside)

values <- matrix(0, nrow=n.dof, ncol=n.tails)
```

(continues on next page)

```
k = 0
for (entry in tail.area.oneside){
    k = k + 1
    values[ , k] <- abs(qt(entry, dof))
}
round(values,3)

par(mar=c(4.2, 4.2, 0.2, 1))
z <- seq(-5, 5, 0.01)
probabilty <- dt(z, df=5)
plot(z, probabilty,
    type = "l",
    main = "",
    xlab = "z",
    ylab = "Probabilities from the t-distribution",
    cex.lab = 1.4,
    cex.main = 1.8,
    lwd = 4,
    cex.sub = 1.8,
    cex.axis = 1.8)
abline(h = 0)
z = 1.5
abline(v = z)
abline(v = 0)
```

2.16 Exercises

Question 1

Recall that $\mu = \mathcal{E}(x) = \frac{1}{N} \sum x$ and $\mathcal{V}\{x\} = \mathcal{E}\left\{(x-\mu)^2\right\} = \sigma^2 = \frac{1}{N} \sum (x-\mu)^2$.

1. What is the expected value thrown of a fair 6-sided die? (Note: plural of die is dice)

2. What is the expected variance of a fair 6-sided die?

Short answer: 3.5; 2.92

Question 2

Characterizing a distribution: Compute the mean, median, standard deviation and MAD for salt content for the various soy sauces given in this report[35] (page 41) as described in the the article from the Globe and Mail[36] on 24 September 2009. Plot a box plot of the data and report the interquartile range (IQR). Comment on the 3 measures of spread you have calculated: standard deviation, MAD, and interquartile range.

The raw data are given below in units of milligrams of salt per 15 mL serving:

```
[460, 520, 580, 700, 760, 770, 890, 910, 920, 940, 960, 1060, 1100]
```

Short answer: IQR = 240 mg salt/15 mL serving

---

[35] https://beta.images.theglobeandmail.com/archive/00245/Read_the_report_245543a.pdf

[36] https://www.theglobeandmail.com/incoming/salt-variation-between-brands-raises-call-for-cuts/article4287171/

Question 3

Give a reason why Statistics Canada reports the median income when reporting income by geographic area. Where would you expect the mean to lie, relative to the median? Use this table[37] to look up the income for Hamilton. How does it compare to Toronto? And all of Canada?

Solution

We described how easily the *mean is influenced by unusual data points* (page 41). Take any group of people anywhere in the world, and there will always be a few who earn lots of money (not everyone can be the CEO, especially of a bank!). Also, since no one earns negative income, the distribution piles up at the left, with fewer people on the right. This implies that the mean will lie above the median, since 50% of the histogram area must lie below the median, by definition. A previous student pointed out that low income earners are less likely to file tax returns, so they are underrepresented in the data.

Even though the median is a more fair way of reporting income, and robust to unusual earners (many low income earners, very few super-rich), I would prefer if Statistics Canada released a histogram - that would tell a lot more - even just the MAD, or IQR would be informative. It was surprising that Hamilton showed higher median earnings per family than Toronto. I infer from this that there are more low income earners in Toronto and Canada than in Hamilton, but without the histograms it is hard to be sure. Also, I wasn't able to find exactly what StatsCan means by a family - did they include single people as a "family"? Maybe there are more, wealthy singles in Toronto, but they are aren't included in the numbers. The median income *per person* would be a useful statistic to help judge that.

Question 4

Use the data set on raw materials[38].

- How many variables in the data set?

- How many observations?

- The data are properties of a powder. Plot each variable, one at a time, and locate any outliers. R-users will benefit from the R tutorial[39] (see the use of the `identify` function).

Solution

See the code below that generates the plots. Outliers were identified by visual inspection of these plots. Recall an outlier is an unusual/interesting point, and a function of the surrounding data. You can use a box plot to locate *preliminary* outliers, but recognize that you are leaving the computer to determine what is unusual. Automated outlier detection systems work moderately well, but there is no substitute (yet!) for visual inspection of the data.

The same few samples appear to be outliers in most of the variables.

```
rm <- read.csv('http://openmv.net/file/raw-material-properties.csv')

ncol(rm)    # 7 columns
```

---

[37] https://www150.statcan.gc.ca/cgi-bin/tableviewer.pl?page=l01/cst01/famil107a-eng.htm
[38] http://openmv.net/info/raw-material-properties
[39] https://learnche.org/4C3/Software_tutorial

```
nrow(rm)      # 36 rows

# Plot the data as you normally would
plot(rm$size1, ylab="Particle size: level 1")

# Now use the identify(...) command, with the same data as you plotted. Use the
# "labels" option to let R use the "Sample" column to label points where you click
identify(rm$size1, labels=rm$Sample)

# After issuing the "identify(...)" command, click on any interesting points in the
# plot.  Right-click anywhere to stop selecting points.


# Repeat with the other columns
plot(rm$size2, ylab="Particle size: level 2")
identify(rm$size2, labels=rm$Sample)
plot(rm$size3, ylab="Particle size: level 3")
identify(rm$size3, labels=rm$Sample)
plot(rm$density1, ylab="Particle density: level 1")
identify(rm$density1, labels=rm$Sample)
plot(rm$density2, ylab="Particle density: level 2")
identify(rm$density2, labels=rm$Sample)
plot(rm$density3, ylab="Particle density: level 3")
identify(rm$density3, labels=rm$Sample)
```
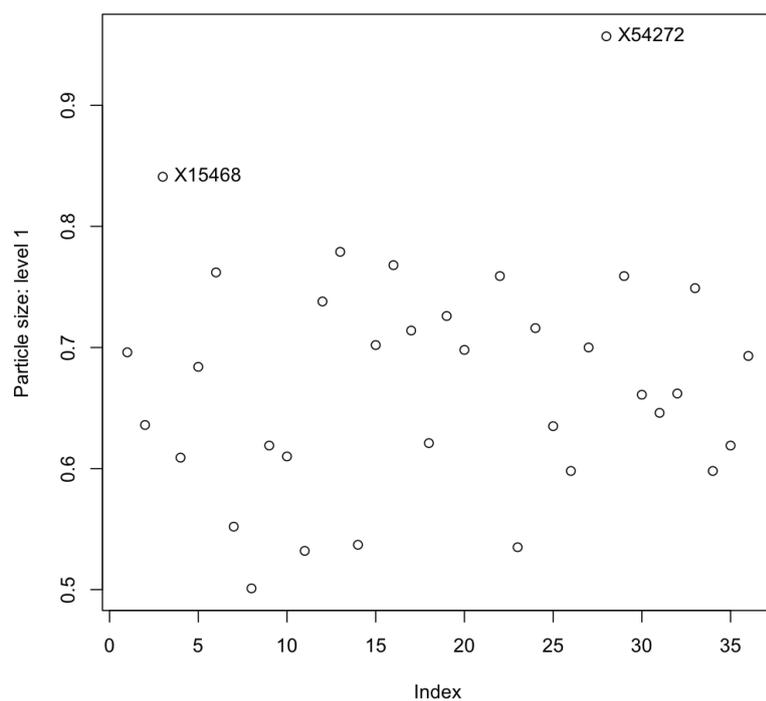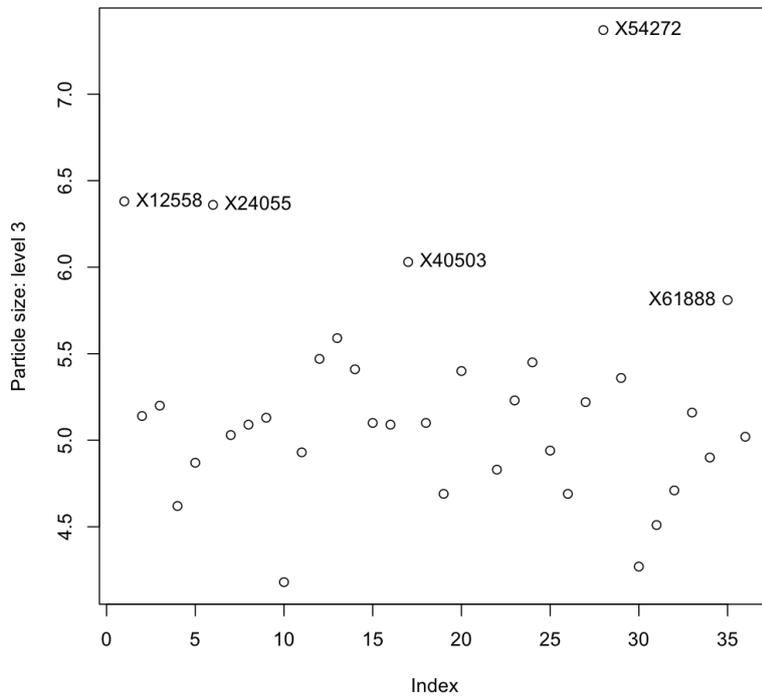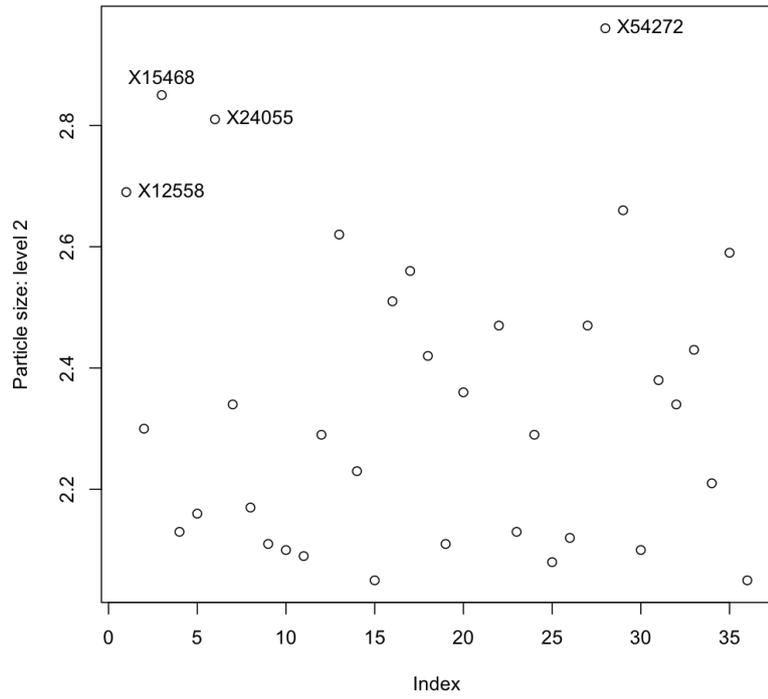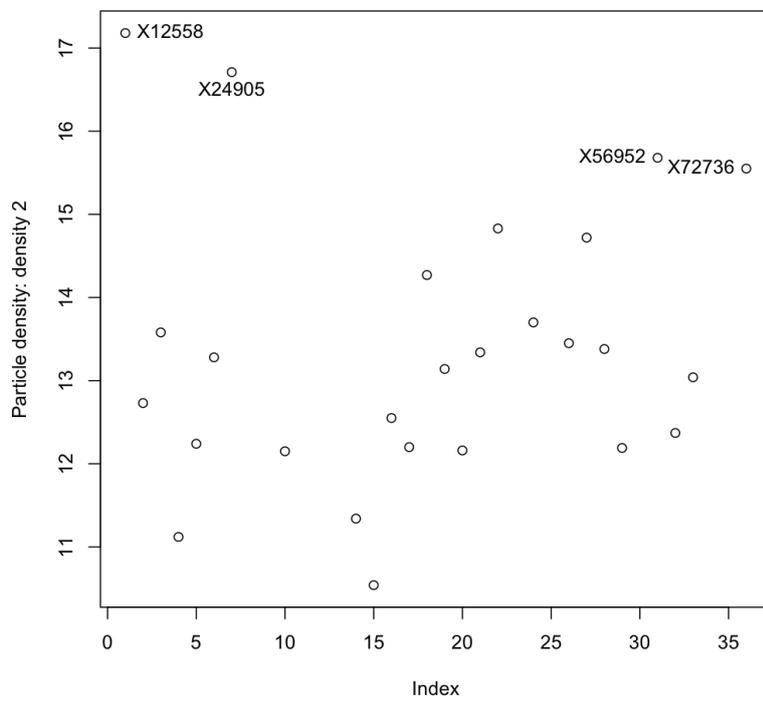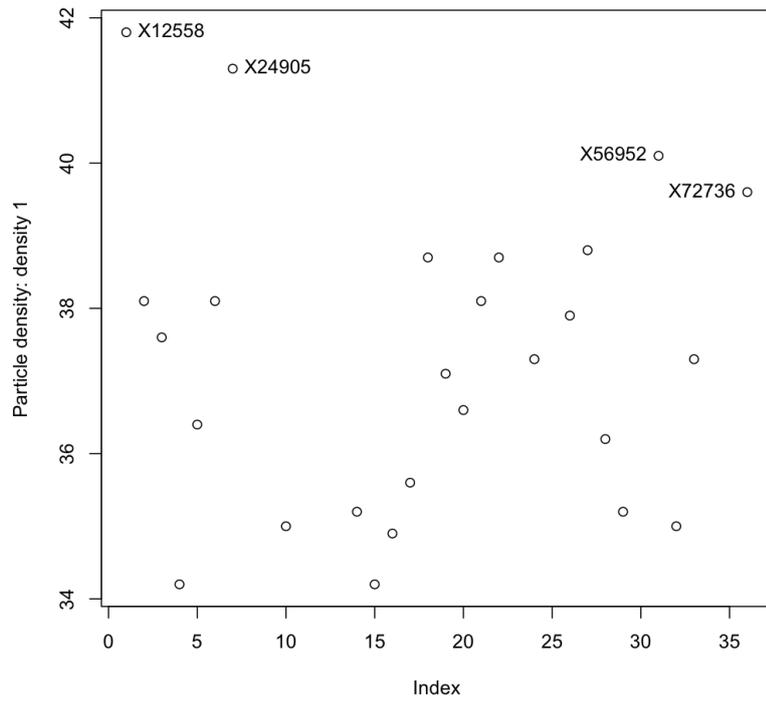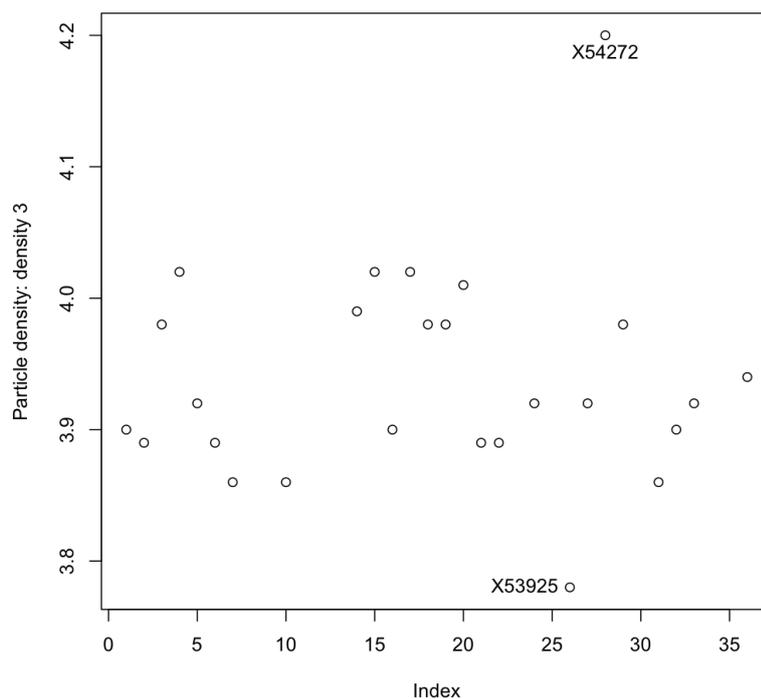
Question 5

Write a few notes on the purpose of feedback control, and its effect on variability of process quality.

Question 6

Use the section on Historical data[40] from Environment Canada's website and use the `Customized Search` option to obtain data for the `HAMILTON A` station from 2000 to 2009. Use the settings as `Year=2000`, and `Data interval=Monthly` and request the data for 2000, then click `Next year` to go to 2001 and so on.

- For each year from 2000 to 2009, get the total snowfall and the average of the `Mean temp` over the whole year (the sums and averages are reported at the bottom of the table).

- Plot these 2 variables against time

- Now retrieve the long-term averages for these data from a different section of their website[41] (use the same location, `HAMILTON A`, and check that the data range is 1971 to 2000). Superimpose the long-term average as a horizontal line on your previous plot.

- **Note**: the purpose of this exercise is more for you to become comfortable with web-based data retrieval, which is common in most companies.

- **Note**: please use any other city for this question if you prefer.

---

[40] https://climate.weather.gc.ca/index_e.html
[41] https://climate.weather.gc.ca/climate_normals/index_e.html

Question 7

Does the number of visits in the website traffic[42] data set follow a normal distribution? If so, what are the parameters for the distribution? What is the likelihood that you will have between 10 and 30 visits to the website?

Short answer: These data are normally distributed according to the q-q plot.

Question 8

The ammonia concentration in your wastewater treatment plant is measured every 6 hours. The data for one year are available from the dataset website[43].

1. Use a visualization plot to hypothesize from which distribution the data might come. Which distribution do you think is most likely? Once you've decided on a distribution, use a qq-plot to test your decision.

2. Estimate location and spread statistics assuming the data are from a normal distribution. You can investigate using the `fitdistr` function in R, in the MASS package.

3. What if you were told the measured values are not independent. How does it affect your answer?

4. What is the probability of having an ammonia concentration greater than 40 mg/L when:

   - you may use only the data (do not use *any* estimated statistics)

   - you use the estimated statistics for the distribution?

   Note: Answer this entire question using computer software to calculate values from the normal distribution. But also make sure you can answer the last part of the question by hand, (when given the mean and variance), and using a table of normal distributions.

Question 9

We take a large bale of polymer composite from our production line and using good sampling techniques, we take 9 samples from the bale and measure the viscosity in the lab for each sample. These samples are independent estimates of the population (bale) viscosity. We will believe these samples follow a normal distribution (we could confirm this in practice by running tests and verifying that samples from any bale are normally distributed). Here are 9 sampled values: `23, 19, 17, 18, 24, 26, 21, 14, 18`.

- The sample average

- An estimate of the standard deviation

- What is the distribution of the sample average, $\overline{x}$? What are the parameters of that distribution?

    Additional information: I use a group of samples and calculate the mean, $\overline{x}$, then I take another group of samples and calculate another $\overline{x}$, and so on. Those values of $\overline{x}$ are not going to be the same, but they should be similar. In other words, the $\overline{x}$ also has a distribution. So this question asks what that distribution is, and what its parameters are.

---

[42] http://openmv.net/info/website-traffic
[43] http://openmv.net/info/ammonia

- Construct an interval, symbolically, that will contain, with 95% certainty (probability), the population mean of the viscosity.

  Additional information: To answer this part, you should move everything to z-coordinates first. Then you need to find the points $-c$ and $+c$ in the following diagram that mark the boundary for a 95% of the total area under the distribution. This region is an interval that will contain, with 95% certainty, the population mean of the viscosity, μ. Write your answer in form: LB $< \mu <$ UB.
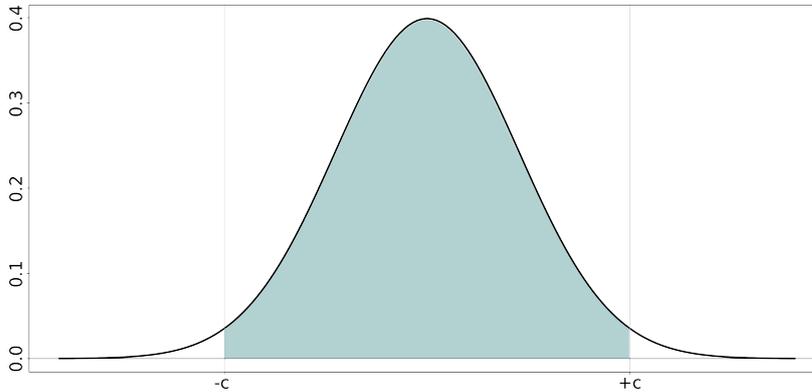
- Now assume that for some hypothetical reason we know the standard deviation of the bale's viscosity is $\sigma = 3.5$ units, calculate the population mean's interval numerically.

  Additional information: In this part you are just finding the values of LB and UB

Short answer: Average = 20, standard deviation = 3.81

Question 10

You are responsible for the quality of maple syrup produced at your plant. Historical data show that the standard deviation of the syrup viscosity is 40 cP. How many lab samples of syrup must you measure so that an estimate of the syrup's long-term average viscosity is inside a **range** of 60 cP, 95% of the time? This question is like the previous one: except this time you are given the range of the interval UB $-$ LB, and you need to find n.

Short answer: 7 samples

Question 11

Your manager is asking for the average viscosity of a product that you produce in a batch process. Recorded below are the 12 most recent values, taken from consecutive batches. State any assumptions, and clearly show the calculations which are required to estimate a 95% confidence interval for the mean. Interpret that confidence interval for your manager, who is not sure what a confidence interval is.

Raw data:	[13.7, 14.9, 15.7, 16.1, 14.7, 15.2, 13.9, 13.9, 15.0, 13.0, 16.7, 13.2]
Mean:	14.67
Standard deviation:	1.16

Ensure you can also complete the question by hand, using statistical tables.

Question 12

A new wastewater treatment plant is being commissioned and part of the commissioning report requires a statement of the confidence interval of the biochemical oxygen demand (BOD)[44]. How many samples must you send to the lab to be sure the true BOD is within a range of 2 mg/L, centered about the sample average? If there isn't enough information given here, specify your own numbers and assumptions and work with them to answer the question.

Question 13

One of the questions we posed at the start of this chapter was: Here are the yields from a batch bioreactor system[45] for the last 3 years (300 data points; we run a new batch about every 3 to 4 days).

1. What sort of distribution do the yield data have?

2. A recorded yield value was less than 60%, what are the chances of that occurring? Express your answer as: *there's a 1 in n chance* of it occurring.

3. Which assumptions do you have to make for the second part of this question?

Question 14

One aspect of your job responsibility is to reduce energy consumption on the plant floor. You ask the electrical supplier for the energy requirements (W.h) for running a particular light fixture for 24 hours. They won't give you the raw data, only their histogram when they tested randomly selected bulbs (see the data and code below).

```
> bin.centers <- c(4025, 4075, 4125, 4175, 4225, 4275, 4325, 4375)
> bin.counts <- c(4, 19, 14,  5,  4,  1,  2,  1)
> barplot(bin.counts, names.arg=bin.centers, ylab="Number of bulbs (N=50)",
    xlab="Energy required over 24 hours (W.h)", col="White", ylim=c(0,20))
```

[44] https://en.wikipedia.org/wiki/Biochemical_oxygen_demand
[45] http://openmv.net/info/batch-yields
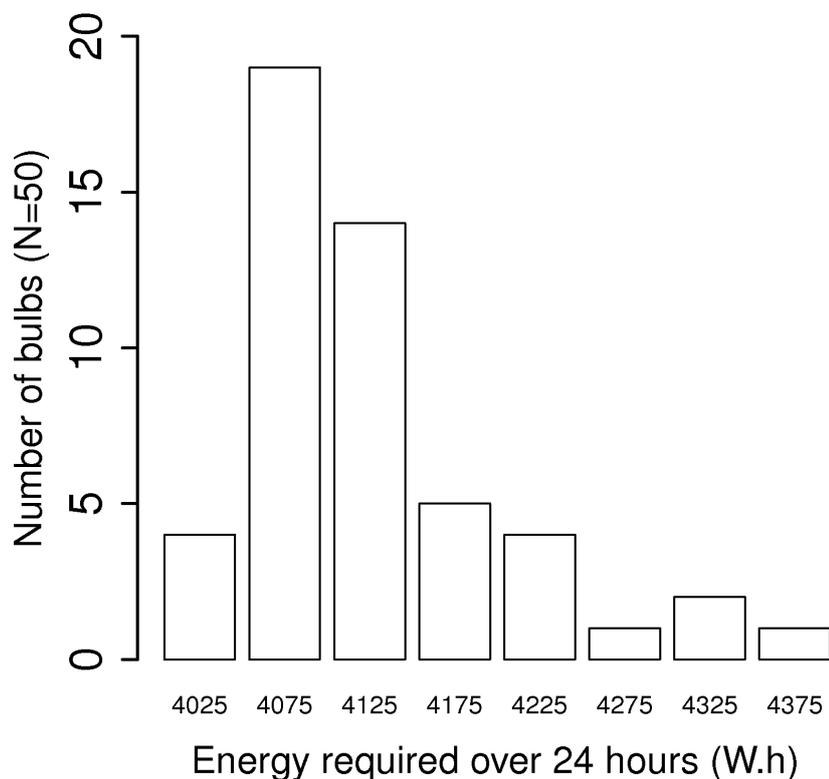
- Calculate an estimate of the mean and standard deviation, even though you don't have the original data.

- What is a confidence interval for the mean at 95% probability, stating and testing any assumptions you need to make.

Short answer: mean = 4127, standard deviation = 78.9

Question 15

The confidence interval for the population mean takes one of two forms below, depending on whether we know the variance or not. At the 90% confidence level, for a sample size of 13, compare and comment on the upper and lower bounds for the two cases. Assume that $s = \sigma = 3.72$.

$$-c_n \quad \leq \quad \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \quad \leq \quad c_n$$

$$-c_t \quad \leq \quad \frac{\bar{x} - \mu}{s/\sqrt{n}} \quad \leq \quad c_t$$

Question 16

A major aim of many engineers is/will be to reduce the carbon footprint of their company's high-profile products. Next week your boss wants you to evaluate a new raw material that requires 2.6 $\frac{\text{kg } CO_2}{\text{kg product}}$ less than the current material, but the final product's brittleness must be the same as achieved with the current raw material. This is a large reduction in CO_2, given your current production capacity of 51,700 kg of product per year. Manpower and physical constraints prevent you from running a randomized test; you don't have a suitable database of historical data either.

One idea you come up with is to use to your advantage the fact that your production line has three parallel reactors, TK104, TK105, and TK107. They were installed at the same time, they have the same geometry, the same instrumentation, *etc*; you have pretty much thought about every factor that might vary between them, and are confident the 3 reactors are identical. Typical production schedules split the raw material between the 3 reactors. Data on the website[46] contain the brittleness values from the three reactors for the past few runs on the current raw material.

1. Which two reactors would you pick to run your comparative trial on next week?

2. Repeat your calculations assuming pairing.

Short answer: You can do an ordinary test of differences, or a paired test. Also note that there are missing data which reduce the degrees of freedom.

Question 17

Use the website traffic data[47] from the dataset website:

*   Write down, symbolically, the z-value for the difference in average visits on a Friday and Saturday.

*   Estimate a suitable value for the variance and justify your choice.

*   What is the probability of obtaining a z-value of this magnitude or smaller? Would you say the difference is significant?

*   Pick any other 2 days that you would find interesting to compare and repeat your analysis.

Solution

---

[46] http://openmv.net/info/brittleness-index
[47] http://openmv.net/info/website-traffic

- Let our variable of interest be the difference between the average of the 2 groups: $\overline{x}_{\text{Fri}} - \overline{x}_{\text{Sat}}$. This variable will be distributed normally (why? - see the notes) according to $\overline{x}_{\text{Fri}} - \overline{x}_{\text{Sat}} \sim \mathcal{N}\left(\mu_{\text{Fri}} - \mu_{\text{Sat}}, \sigma_{\text{diff}}^2\right)$. So the z-value for this variable is: $z = \dfrac{(\overline{x}_{\text{Fri}} - \overline{x}_{\text{Sat}}) - (\mu_{\text{Fri}} - \mu_{\text{Sat}})}{\sigma_{\text{diff}}}$

- The variance of the difference, $\sigma_{\text{diff}}^2 = \sigma^2 \left(\dfrac{1}{n_{\text{Fri}}} + \dfrac{1}{n_{\text{Sat}}}\right)$, where σ^2 is the variance of the number of visits to the website on Friday and Saturday. Since we don't know that value, we can estimate it from pooling the 2 variances of each group. We should calculate first that these variances are comparable (they are; but you *should confirm this yourself* (page 78)).

$$\sigma^2 \approx s_P^2 = \frac{(n_{\text{Fri}} - 1)s_{\text{Fri}}^2 + (n_{\text{Sat}} - 1)s_{\text{Sat}}^2}{n_{\text{Fri}} - 1 + n_{\text{Sat}} - 1}$$
$$= \frac{29 \times 45.56 + 29 \times 48.62}{58}$$
$$= 47.09$$

- The z-value calculated from this pooled variance is:

$$z = \frac{20.77 - 15.27}{47.09\left(\dfrac{1}{30} + \dfrac{1}{30}\right)} = 3.1$$

But since we used an estimated variance, we cannot say that z comes from the normal distribution anymore. It now follows the t-distribution with 58 degrees of freedom (which is still comparable to the normal distribution - see question 7 below). The corresponding probability that $z < 3.1$ is 99.85%, using the t-distribution with 58 degrees of freedom. This difference is significant; there is a very small probability that this difference is due to chance alone.

- The code was modified to generate the matrix of z-value results in the comments below. The largest difference is between Sunday and Wednesday, and the smallest difference is between Monday and Tuesday.

```
website <- read.csv('http://openmv.net/file/website-traffic.csv')
attach(website)

visits.Mon <- Visits[DayOfWeek=="Monday"]
visits.Tue <- Visits[DayOfWeek=="Tuesday"]
visits.Wed <- Visits[DayOfWeek=="Wednesday"]
visits.Thu <- Visits[DayOfWeek=="Thursday"]
visits.Fri <- Visits[DayOfWeek=="Friday"]
visits.Sat <- Visits[DayOfWeek=="Saturday"]
visits.Sun <- Visits[DayOfWeek=="Sunday"]

# Look at a boxplot of the data from Friday and Saturday
bitmap('website-boxplot.png', type="png256", width=7, height=7,
    res=250, pointsize=14)
par(mar=c(4.2, 4.2, 0.2, 0.2))  # (bottom, left, top, right)
boxplot(visits.Fri, visits.Sat, names=c("Friday", "Saturday"), ylab="Number of visits",
        cex.lab=1.5, cex.main=1.8, cex.sub=1.8, cex.axis=1.8)
dev.off()

# Use the "group_difference" function from question 4
group_difference(visits.Sat, visits.Fri)
# z = 3.104152
# t.critical = 0.9985255 (1-0.001474538)

# All differences: z-values
# ---------------------------
#        Mon        Tue       Wed       Thu       Fri       Sat       Sun
# Mon  0.0000000         NA        NA        NA        NA        NA   NA
# Tue -0.2333225   0.000000        NA        NA        NA        NA   NA
```

(continues on next page)

```
# Wed -0.7431203 -0.496627 0.000000       NA       NA       NA   NA
# Thu  0.8535025  1.070370 1.593312 0.000000       NA       NA   NA
# Fri  2.4971347  2.683246 3.249602 1.619699 0.000000       NA   NA
# Sat  5.4320361  5.552498 6.151868 4.578921 3.104152  0.000000   NA
# Sun  3.9917201  4.141035 4.695493 3.166001 1.691208 -1.258885    0
```

Question 18

You plan to run a series of 22 experiments to measure the economic advantage, if any, of switching to a corn-based raw material, rather than using your current sugar-based material. You can only run one experiment per day, and there is a high cost to change between raw material dispensing systems. Describe two important precautions you would implement when running these experiments, so you can be certain your results will be accurate.

Question 19

There are two analytical techniques for measuring biochemical oxygen demand (BOD)[48]. You wish to evaluate the two testing procedures, so that you can select the test which has lower cost, and fastest turn-around time, but without a compromise in accuracy. The table contains the results of the each test, performed on a sample that was split in half.

1. Is there a *statistical* difference in accuracy between the two methods?

2. Review the raw data and answer whether there is a practical difference in accuracy.

Dilution method	Manometric method
11	25
26	3
18	27
16	30
20	33
12	16
8	28
26	27
12	12
17	32
14	16

---

[48] https://en.wikipedia.org/wiki/Biochemical_oxygen_demand

Question 20

Plot the cumulative probability function for the normal distribution and the t-distribution on the same plot.

- Use 6 degrees of freedom for t-distribution.

- Repeat the plot for a larger number of degrees of freedom.

- At which point is the t-distribution indistinguishable from the normal distribution?

- What is the practical implication of this result?

Solution

```
z <- seq(-5, 5, 0.1)
norm <- pnorm(z)

bitmap('normal-t-comparison.png', type="png256", width=12, height=7,
    res=300, pointsize=14)
par(mar=c(4.2, 4.2, 2.2, 0.2))

layout(matrix(c(1,2), 1, 2))
plot(z, norm, type="p", pch=".", cex=5, main="Normal and t-distribution (df=6)",
    ylab="Cumulative probability")
lines(z, pt(z, df=6), type="l", lwd=2)
legend(0.5, y=0.35, legend=c("Normal distribution", "t-distribution (df=8)"),
    pch=c(".", "-"), pt.cex=c(5, 2))

plot(z, norm, type="p", pch=".", cex=5, main="Normal and t-distribution (df=35)",
    ylab="Cumulative probability")
lines(z, pt(z, df=35), type="l", lwd=2)
legend(0.5, y=0.35, legend=c("Normal distribution", "t-distribution (df=35)"),
    pch=c(".", "-"), pt.cex=c(5, 2))

dev.off()
```
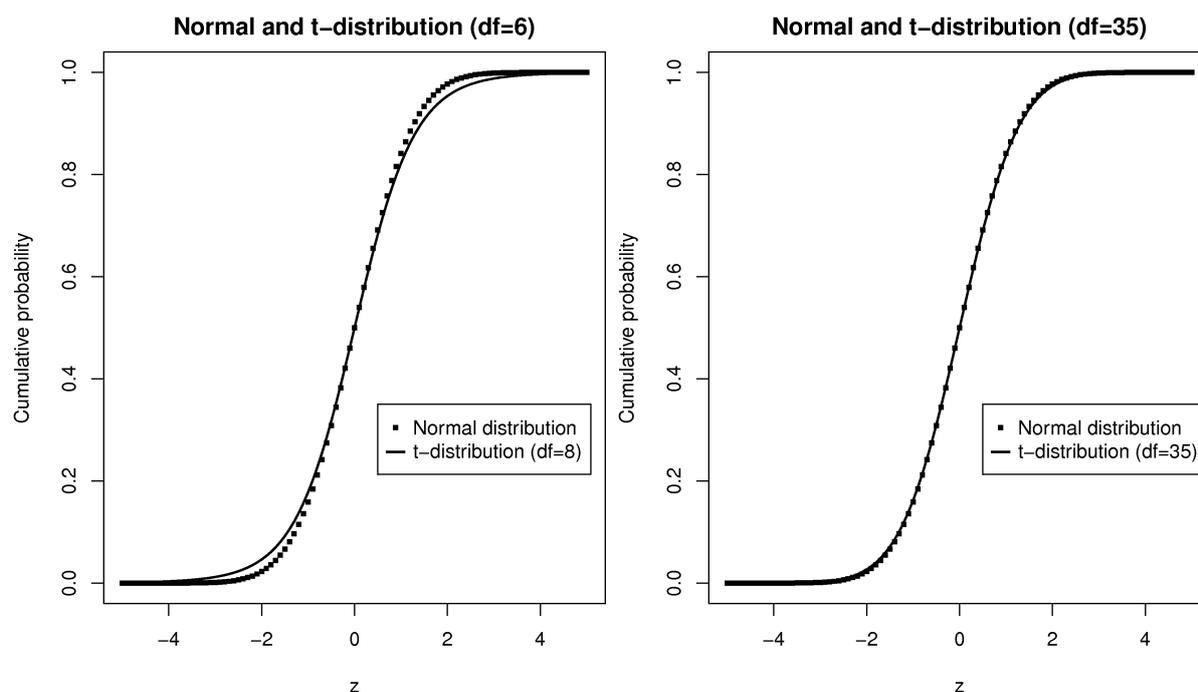

The above source code and figure output shows that the t-distribution starts being indistinguishable

from the normal distribution after about 35 to 40 degrees of freedom. This means that when we deal with large sample sizes (over 40 or 50 samples), then we can use critical values from the normal distribution rather than the t-distribution. Furthermore, it indicates that our estimate of the variance is a pretty good estimate of the population variance for largish sample sizes.

Question 21

Explain why tests of differences are insensitive to unit changes. If this were not the case, then one could show a significant difference for a weight-loss supplement when measuring waist size in millimetres, yet show no significant difference when measuring in inches!

Question 22

A food production facility fills bags with potato chips. The advertised bag weight is 35.0 grams. But, the current bagging system is set to fill bags with a mean weight of 37.4 grams, and this done so that only 1% of bags have a weight of 35.0 grams or less.

- Back-calculate the standard deviation of the bag weights, assuming a normal distribution.

- Out of 1000 customers, how many are lucky enough to get 40.0 grams or more of potato chips in their bags?

Short answer: standard deviation = 1.03 grams

Question 23

A food production facility fills bags with potato chips with an advertised bag weight of 50.0 grams.

1. The government's *Weights and Measures Act* requires that at most 1.5% of customers may receive a bag containing less than the advertised weight. At what setting should you put the target fill weight to meet this requirement exactly? The check-weigher on the bagging system shows the long-term standard deviation for weight is about 2.8 grams.

2. Out of 100 customers, how many are lucky enough to get 55.0 grams or more of potato chips in their bags?

Question 24

The following confidence interval is reported by our company for the amount of sulphur dioxide measured in parts per billion (ppb) that we send into the atmosphere.

$$123.6\,\text{ppb} \le \mu \le 240.2\,\text{ppb}$$

Only $n = 21$ raw data points (one data point measured per day) were used to calculate that 90% confidence interval. A z-value would have been calculated as an intermediate step to get the final confidence interval, where $z = \dfrac{\overline{x} - \mu}{s/\sqrt{n}}$.

1. What assumptions were made about those 21 raw data points to compute the above confidence interval?

2. Which lower and upper critical values would have been used for z? That is, which critical values are used before unpacking the final confidence interval as shown above.

3. What is the standard deviation, s, of the raw data?

4. Today's sulphur dioxide reading is 460 ppb and your manager wants to know what's going on; you can quickly calculate the probability of seeing a value of 460 ppb, or greater, to help judge the severity of the pollution. How many days in a 365 calendar-day year are expected to show a sulphur dioxide value of 460 ppb or higher?

5. Explain clearly why a wide confidence interval is not desirable, from an environmental perspective.

 Solution

1. The 21 data points are independent and come from *any distribution* of finite variance.

2. From the t-distribution at 20 degrees of freedom, with 5% in each tail: $c_t = 1.72 = $ `qt(0.95, df=20)`. The t-distribution is used because the standard deviation is estimated, rather than being a population deviation.

3. The standard deviation may be calculated from:

$$UB - LB = 240.2 - 123.6 = 2 \times c_t \frac{s}{\sqrt{n}} = (2)(1.72)\frac{s}{\sqrt{n}}$$

$$s = \frac{(116)(\sqrt{n})}{(2)(1.72)}$$

$$s = 154.5 \, \text{ppb}$$

Note the very large standard deviation relative to the confidence interval range. This is the reason why so many data points were taken (21), to calculate the average, because the raw data comes from a distribution with such a large variation.

An important note here is the large estimated value for the standard deviation and realized it was so wide, that it would imply the distribution produced values with negative sulphur dioxide concentration (which is physically impossible). However, note that when dealing with large samples (21 in this case), the distinction between the normal and the t-distribution is minimal. Further, the raw data are not necessarily assumed to be from the normal distribution, they could be from any distribution, including one that is heavy-tailed, such as the F-distribution[49] (see the yellow and green lines in particular).

4. The probability calculation requires a mean value. Our best guess for the mean is the midpoint of the confidence interval, which is always symmetric about the estimated process mean, $\bar{x} = \frac{240.2 - 123.6}{2} + 123.6 = 181.9$. Note that this is not the value for μ, since μ is unknown.

$$z = \frac{460 - 181.9}{154.5} = 1.80$$

Probability is `1 - pt(1.8, df=20)` $= 1 - 0.9565176 = 0.0434824$, or about $0.0434824 \times 365 = 15.9$, or about 16 days in the year (some variation is expected, if you have used a statistical table)

5. A wide confidence interval implies that our sulphur dioxide emissions are extremely variable (the confidence interval bounds are a strong function of the process standard deviation). Some days we

---

[49] https://en.wikipedia.org/wiki/File:F_distributionPDF.png

are putting more pollution up into the air and balancing it out with lower pollution on other days. Those days with high pollution are more environmentally detrimental.

Question 25

A common unit operation in the pharmaceutical area is to uniformly blend powders for tablets. One such unit is illustrated below (figure taken from Wikipedia[50]). In this question we consider blending an excipient (an inactive magnesium stearate base), a binder, and the active ingredient. The mixing process is tracked using a wireless near infrared (NIR) probe embedded in a V-blender. The mixer is stopped when the NIR spectra become stable. A new supplier of magnesium stearate is being considered that will save $ 294,000 per year.
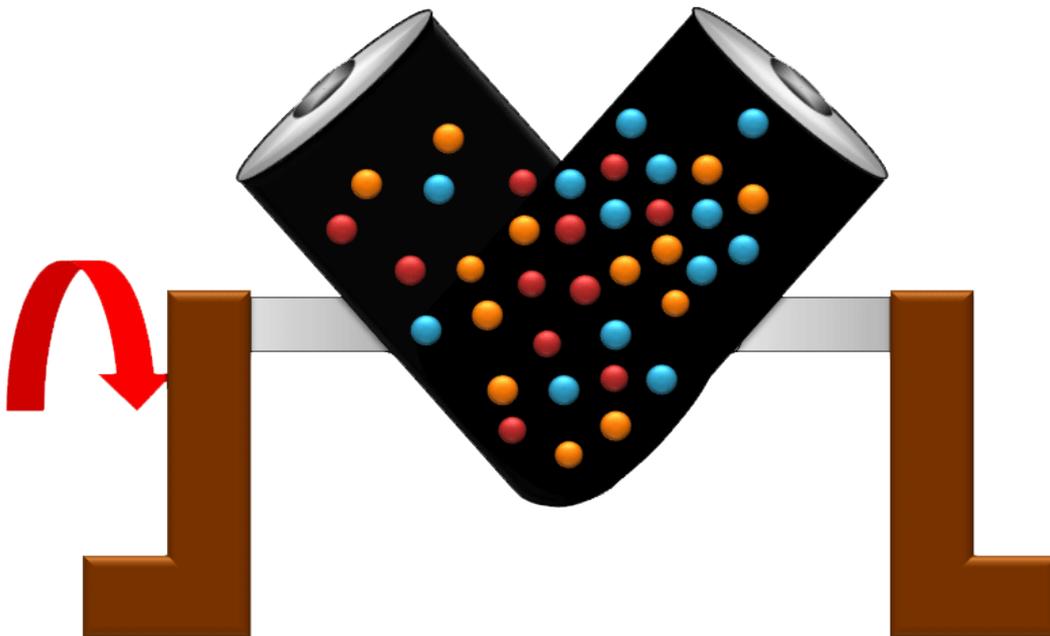
The 15 most recent runs with the current magnesium stearate supplier had an average mixing time of 2715 seconds, and a standard deviation of 390 seconds. So far you have run 6 batches from the new supplier, and the average mixing time of these runs is 3115 seconds with a standard deviation of 452 seconds. Your manager is not happy with these results so far - this extra mixing time will actually cost you more money via lost production.

The manager wants to revert back to the original supplier, but is leaving the decision up to you; what would be your advice? Show all calculations and describe any additional assumptions, if required.

Short answer: This problem is open-ended: pay attention to having a significant difference vs a practical difference.

---

[50] https://en.wikipedia.org/wiki/Industrial_mixer

Question 26

List an advantage of using a paired test over an unpaired test. Give an example, not from the notes, that illustrates your answer.

Question 27

An *unpaired* test to distinguish between group A and group B was performed with 18 runs: 9 samples for group A and 9 samples for group B. The pooled variance was 86 units.

Also, a *paired* test on group A and group B was performed with 9 runs. After calculating the paired differences, the variance of these differences was found to be 79 units.

Discuss, in the context of this example, an advantage of paired tests over unpaired tests. Assume 95% confidence intervals, and that the true result was one of "no significant difference between method A and method B". Give numeric values from this example to substantiate your answer.

Question 28

You are convinced that a different impeller (mixing blade) shape for your tank will lead to faster, i.e. shorter, mixing times. The choices are either an axial blade or a radial blade, as shown in this figure from Wikipedia[51].
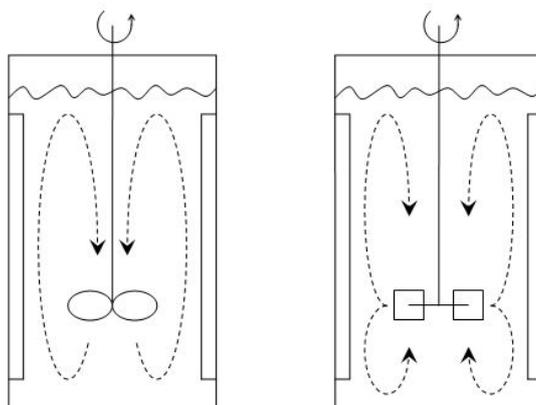
Before obtaining approval to run some experiments, your team wants you to explain how you will interpret the experimental data. Your reply is that you will calculate the average mixing time from each blade type and then calculate a confidence interval for the difference. A team member asks you what the following 95% confidence intervals would mean:

1. -453 seconds $\leq \mu_{\text{Axial}} - \mu_{\text{Radial}} \leq 390$ seconds

2. -21 seconds $\leq \mu_{\text{Axial}} - \mu_{\text{Radial}} \leq 187$ seconds

For both cases (a) explain what the confidence interval means in the context of this experiment, and (b) whether the recommendation would be to use radial or axial impellers to get the shortest mixing time.

---

[51] https://en.wikipedia.org/wiki/Impeller

3. Now assume the result from your experimental test was
-21 seconds $\leq \mu_{\text{Axial}} - \mu_{\text{Radial}} \leq 187$ seconds; how can you make the confidence interval narrower?

Question 29

The paper by PJ Rousseeuw, "Tutorial to Robust Statistics[52]", *Journal of Chemometrics*, **5**, 1-20, 1991 discusses the breakdown point of a statistic.

1. Describe what the breakdown point is, and give two examples: one with a low breakdown point, and one with a high breakdown point. Use a vector of numbers to help illustrate your answer.

2. What is an advantage of using robust methods over their "classical" counterparts?

Solution

1. PJ Rousseeuw defines the breakdown point on page 3 of his paper as "... the smallest fraction of the observations that have to be replaced to make the estimator unbounded. In this definition one can choose which observations are replaced, as well as the magnitude of the outliers, in the least favourable way".

   A statistic with a low breakdown point is the mean, of the n values used to calculate the mean, only 1 needs to be replaced to make the estimator unbounded; i.e. its breakdown point is $1/n$. The median though has a breakdown point of 50%, as one would have to replace 50% of the n data points in the vector before the estimator becomes unbounded.

   Use this vector of data as an example: $[2, 6, 1, 9151616, -4, 2]$. The mean is 1525270, while the median is 2.

2. • Robust methods are insensitive to outliers, which is useful when we need a measure of location or spread that is calculated in an automated way. It is increasingly prevalent to skip out the "human" step that might have detected the outlier, but our data sets are getting so large that we can't possibly visualize or look for outliers manually anymore.

   • As described in the above paper by Rousseeuw, robust methods also emphasize outliers. Their "lack of sensitivity to outliers" can also be considered an advantage.

Question 30

1. Why are robust statistics, such as the median or MAD, important in the analysis of modern data sets? Explain, using an example, if necessary.

2. What is meant by the break-down point of a robust statistic? Give an example to explain your answer.

Solution

1. Data sets you will have to deal with in the workplace are getting larger and larger (lengthwise), and processing them by trimming outliers (see Question 5 later) manually is almost impossible. Robust statistics are a way to summarize such data sets without point-by-point investigation.

---

[52] https://dx.doi.org/10.1002/cem.1180050103

This is especially true for automatic systems that you will build that need to (a) acquire and (b) process the data to then (c) produce meaningful output. These systems have to be capable of dealing with outliers and missing values.

2. The breakdown point is the number of contaminating data points required before a statistic (estimator) becomes unbounded, i.e. useless. For example, the mean requires only 1 contaminating value, while the median requires 50% + 1 data points before it becomes useless.

   Consider the sequence $[2, 6, 1, 91511, -4, 2]$. The mean is 15253, while the median is 2, which is a far more useful estimate of the central tendency in the data.

Question 31

Recall that $\mu = \mathcal{E}(x) = \frac{1}{N}\sum x$ and $\mathcal{V}\{x\} = \mathcal{E}\left\{(x - \mu)^2\right\} = \sigma^2 = \frac{1}{N}\sum(x - \mu)^2$.

1. What is the expected value thrown of a fair, 12-sided dice?

2. What is the expected variance of a fair, 12-sided dice?

3. Simulate 10,000 throws in a software package (R, MATLAB, or Python) from this dice and see if your answers match those above. Record the average value from the 10,000 throws, call that average $\bar{x}$.

4. Repeat the simulation 10 times, calculating the average value of all the dice throws. Calculate the mean and standard deviation of the 10 $\bar{x}$ values and *comment* whether the results match the theoretically expected values.

Solution

The objective of this question is to recall basic probability rules.

1. Each value on the dice is equally probable, so the expected value thrown will be:

$$\mathcal{E}(X) = \sum_{i=1}^{12} x_i P(x_i) = P(x)\sum_{i=1}^{12} x_i = \frac{1}{12}(1 + 2 + \cdots + 12) = \mathbf{6.5}$$

   This value is the population mean, μ.

2. Continuing the notation from the above question we can derive the expected variance as,

$$\mathcal{V}(X) = \frac{1}{N}\sum_i^{12}(x_i - \mu)^2 = \frac{1}{12}\cdot\left[(1 - 6.5)^2 + (2 - 6.5)^2 + \ldots + (12 - 6.5)^2\right] \approx \mathbf{11.9167}$$

3. Simulating 10,000 throws corresponds to 10,000 independent and mutually exclusive random events, each with an outcome between 1 and 12. The sample mean and variance from my sample was calculated using this code in R:

$$\bar{x} = 6.5219$$
$$s^2 = 12.03732$$

```
# Set the random seed to a known point, to allow
# us to duplicate pseudorandom results
set.seed(13)

x.data <- as.integer(runif(10000, 1, 13))

# Verify that it is roughly uniformly distributed
# across 12 bins
hist(x.data, breaks=seq(0,12))

x.mean <- mean(x.data)
x.var <- var(x.data)
c(x.mean, x.var)
```

4. Repeating the above simulation 10 times (i.e. 10 independent experiments) produces 10 different estimates of μ and σ^2. Note, your answer should be slightly different, and different each time you run the simulation.

```
N <- 10
n <- 10000
x.mean <- numeric(N)
x.var <- numeric(N)
for (i in 1:N) {
  x.data <- as.integer(runif(n, 1, 13))
  x.mean[i] <- mean(x.data)
  x.var[i] <- var(x.data)
}

x.mean
# [1] 6.5527 6.4148 6.4759 6.4967 6.4465
# [6] 6.5062 6.5171 6.4671 6.5715 6.5485

x.var
# [1] 11.86561 11.84353 12.00102 11.89658 11.82552
# [6] 11.83147 11.95224 11.88555 11.81589 11.73869

# You should run the code several times and verify whether
# the following values are around their expected, theoretical
# levels.  Some runs should be above, and other runs below
# the theoretical values.
# This is the same as increasing "N" in the first line.

# Is it around 6.5?
mean(x.mean)

# Is it around 11.9167?
mean(x.var)

# Is it around \sigma^2 / n = 11.9167/10000 = 0.00119167 ?
var(x.mean)
```

Note that each $\overline{x} \sim \mathcal{N}\left(\mu, \sigma^2/n\right)$, where $n = 10000$. We know what σ^2 is in this case: it is our theoretical value of **11.92**, calculated earlier, and for $n = 10000$ samples, our theoretical expectation is that $\overline{x} \sim \mathcal{N}\left(6.5, 0.00119167\right)$.

Calculating the average of those 10 means, let's call that $\overline{\overline{x}}$, shows a value close to 6.5, the theoretical mean.

Calculating the variance of those 10 means shows a number around 0.00119167, as expected.

Question 32

Removed. Was a duplicate of a prior question (number 13).

Question 33

1. At the 95% confidence level, for a sample size of 7, compare and comment on the upper and lower bounds of the confidence interval that you would calculate if:

   a) you know the population standard deviation

   b) you have to estimate it for the sample.

   Assume that the calculated standard deviation from the sample, s matches the population $\sigma = 4.19$.

2. As a follow up, overlay the probability distribution curves for the normal and t-distribution that you would use for a sample of data of size $n = 7$.

3. Repeat part of this question, using larger sample sizes. At which point does the difference between the t- and normal distributions become *practically* indistinguishable?

4. What is the implication of this?

Question 34

Engineering data often violate the assumption of independence. In this question you will create (simulate) sequences of autocorrelated data, i.e. data that lack independence, and investigate how lack of independence affects our results.

The simplest form of autocorrelation is what is called lag-1 autocorrelation, when the series of values, x_k is correlated with itself only 1 step back in time, x_{k-1}:

$$x_k = \phi x_{k-1} + a_k$$

The a_k value is a random error and for this question let $a_k \sim \mathcal{N}\left(\mu = 0, \sigma^2 = 25.0\right)$.

Create 3 sequences of autocorrelated data with:

    A: $\phi = +0.7$ (positively correlated)

    B: $\phi = 0.0$ (uncorrelated data)

    C: $\phi = -0.6$ (negatively correlated)

For case A, B and C perform the following analysis. Repeat the following 1000 times (let $i = 1, 2, \ldots, 1000$):

- Create a vector of 100 autocorrelated x values using the above formula, using the current level of ϕ

- Calculate the mean of these 100 values, call it $\bar{x}_i$ and store the result

At this point you have 1000 $\bar{x}_i$ values for case A, another 1000 $\bar{x}_i$ values for case B, and similarly for case C. Now answer these questions:

1. Assuming independence, which is obviously not correct for 2 of the 3 cases, nevertheless, from which population should $\bar{x}$ be from, and what are the 2 parameters of that population?

2. Now, using your 1000 simulated means, estimate those two population parameters.

3. Compare your estimates to the theoretical values.

Comment on the results, and the implication of this regarding tests of significance (i.e. statistical tests to see if a significant change occurred or not).

Solution

We expect that case B should match the theoretical case the closest, since data from case B are truly independent, since the autocorrelation parameter is zero. We expect case A and C datasets, which violate that assumption of independence, to be biased one way or another. This question aims to see **how** they are biased.

```
nsim <- 1000           # Number of simulations
x.mean <- numeric(nsim) # An empty vector to store the results

set.seed(37)           # so that you can reproduce these results
for (i in 1:nsim)
{
    N <- 100           # number of points in autocorrelated sequence
    phi <- +0.7        # ** change this line for case A, B and C **
    spread <- 5.0      # standard deviation of random variables
    x <- numeric(N)
    x[1] = rnorm(1, mean=0, sd=spread)
    for (k in 2:N){
       x[k] <- phi*x[k-1] + rnorm(1, mean=0, sd=spread)
    }
    x.mean[i] <- mean(x)
}
theoretical <- sqrt(spread^2/N)

# Show some output to the user
c(theoretical, mean(x.mean), sd(x.mean))
```

You should be able to reproduce the results I have below, because the above code uses the `set.seed(...)` function, which forces R to generate random numbers in the same order on my computer as yours (as long as we all use the same version of R).

- Case A: `0.50000000, 0.00428291, 1.65963302`

- Case B: `0.50000000, 0.001565456, 0.509676562`

- Case C: `0.50000000, 0.0004381761, 0.3217627596`

The first output is the same for all 3 cases: this is the theoretical standard deviation of the distribution from which the $\overline{x}_i$ values come: $\overline{x}_i \sim \mathcal{N}\left(\mu, \sigma^2/N\right)$, where $N = 100$, the number of points in the autocorrelated sequence. This result comes from the central limit theorem, which tells us that $\overline{x}_i$ should be normally distributed, with the same mean as our individual x-values, but have smaller variance. That variance is σ^2/N, where σ is the variance of the distribution from which we took the raw x values. That theoretical variance value is $25/100$, or theoretical standard deviation of $\sqrt{25/100} = \mathbf{0.5}$.

But, the central limit theorem only has one *crucial* assumption: that those raw x values are independent. We intentionally violated this assumption for case A and C.

We use the 1000 simulated values of $\overline{x}_i$ and calculate the average of the 1000 $\overline{x}_i$ values and the standard deviation of the 1000 $\overline{x}_i$ values. Those are the second and third values reported above.

We see in all cases that the mean of the 1000 values nearly matches 0.0. If you run the simulations again, with a different seed, you will see it above zero, and sometimes below zero for all 3 cases. So we can conclude that lack of independence *does not* affect the estimated mean.

The major disagreement is in the variance though. Case B matches the theoretical variance; data that are positively correlated have an inflated standard deviation, 1.66; data that are negatively correlated have a deflated standard deviation, 0.32 when $\phi = -0.6$.

This is problematic for the following reason. When doing a test of significance, we construct a confidence interval:

$$
\begin{array}{ccccc}
-c_t & \leq & \dfrac{\overline{x} - \mu}{s/\sqrt{n}} & \leq & +c_t \\[2ex]
\overline{x} - c_t \dfrac{s}{\sqrt{n}} & \leq & \mu & \leq & \overline{x} + c_t \dfrac{s}{\sqrt{n}} \\[2ex]
\text{LB} & \leq & \mu & \leq & \text{UB}
\end{array}
$$

We use an estimated standard deviation, s, whether that is found from pooling the variances or found separately (it doesn't really matter), but the main problem is that s is not accurate when the data are not independent:

- For positive correlations (quite common in industrial data): our confidence interval will be too wide, likely spanning zero, indicating no statistical difference, when in fact there might be one.

- For negative correlations (less common, but still seen in practice): our confidence interval will be too narrow, more likely to indicate there is a difference.

The main purpose of this question is for you to see how use to understand what happens when a key assumption is violated. There are cases when an assumption is violated, but it doesn't affect the result too much.

In this particular example there is a known theoretical relationship between ϕ and the inflated/deflated variance that can be derived (with some difficulty). But in most situations the affect of violating assumptions is too difficult to derive mathematically, so we use computer power to do the work for us: but then we still have to spend time thinking and interpreting the results.

Question 35

Sulphur dioxide is a byproduct from ore smelting, coal-fired power stations, and other sources.

These 11 samples of sulphur dioxide, SO_2, measured in parts per billion [ppb], were taken from our plant. Environmental regulations require us to report the 90% confidence interval for the mean SO_2 value.

$$180, \ 340, \ 220, \ 410, \ 101, \ 89, \ 210, \ 99, \ 128, \ 113, \ 111$$

1. What is the confidence interval that must be reported, given that the sample average of these 11 points is 181.9 ppb and the sample standard deviation is 106.8 ppb?

2. Why might Environment Canada require you to report the confidence interval instead of the mean?

Question 36

A concrete slump test is used to test for the fluidity, or workability, of concrete. It's a crude, but quick test often used to measure the effect of polymer additives that are mixed with the concrete to improve workability.

The concrete mixture is prepared with a polymer additive. The mixture is placed in a mold and filled to the top. The mold is inverted and removed. The height of the mold minus the height of the remaining concrete pile is called the "slump", as shown in this figure from Wikipedia[53].
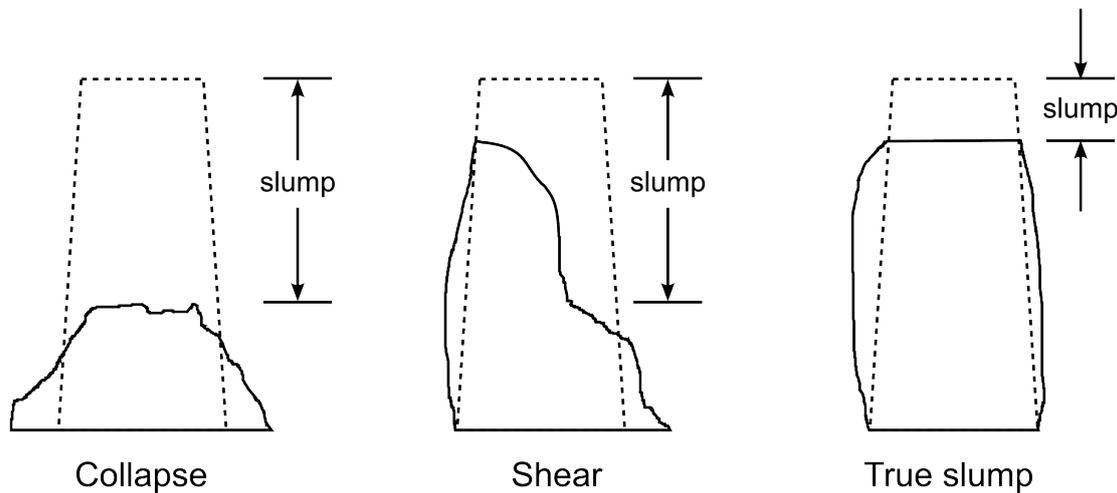
Collapse   Shear   True slump

Your company provides the polymer additive, and you are developing an improved polymer formulation, call it B, that hopefully provides the same slump values as your existing polymer, call it A. Formulation B costs less money than A, but you don't want to upset, or lose, customers by varying the slump value too much.

1. You have a single day to run your tests (experiments). Preparation, mixing times, measurement and clean up take 1 hour, only allowing you to run 10 experiments. Describe all precautions, and why you take these precautions, when planning and executing your experiment. Be very specific in your answer (use bullet points).

2. The following slump values were recorded over the course of the day:

Additive	Slump value [cm]
A	5.2
A	3.3
B	5.8
A	4.6
B	6.3
A	5.8
A	4.1
B	6.0
B	5.5
B	4.5

What is your conclusion on the performance of the new polymer formulation (system B)? Your

---

[53] https://en.wikipedia.org/wiki/File:Types_of_concrete_slump.jpg

conclusion must either be "send the polymer engineers back to the lab" or "let's start making formulation B for our customers". Explain your choice clearly.

To help you, $\bar{x}_A = 4.6$ and $s_A = 0.97$. For system B: $\bar{x}_B = 5.62$ and $s_B = 0.69$.

Note: In your answer you must be clear on which assumptions you are using and, where necessary, why you need to make those assumptions.

3. Describe the circumstances under which you would rather use a paired test for differences between polymer A and B.

4. What are the advantage(s) of the paired test over the unpaired test?

5. Clearly explain which assumptions are used for paired tests, and why they are likely to be true in this case?

6. The slump tests were actually performed in a paired manner, where pairing was performed based on the cement supplier. Five different cement suppliers were used:

Supplier	Slump value [cm] from A	Slump value [cm] from B
1	5.2	5.8
2	3.3	4.5
3	4.6	6.0
4	5.8	5.5
5	4.1	6.2

Use these data, and provide, if necessary, an updated recommendation to your manager.

Question 37

You are planning a series of experiments to test alternative conditions in a store and see which conditions lead to higher sales.

Which practical steps would you take to ensure independence in the experimental data, when investigating:

1. adjustable halogen lighting: **A** = soft and dim lighting and **B** = brighter lighting

2. alternative shelving: **A** = solid white metal shelves and **B** = commercial stainless steel racking

Solution

By Cameron DiPietro and Andrew Haines (2012 class)

Randomization is expensive and inconvenient; however, the high cost is to ensure that the results attained in each study are not affected by unmeasured disturbances. We also have to take care to control measured disturbances as far as possible.

1. To ensure independence when investigating adjustable halogen lighting: A = soft and dim lighting and B = brighter lighting, the following experiments and conditions may be run:

   - All light fixtures are changed correctly during the swap from A to B and the same scenario from B to A

- Keep prices of all products the same during days with A lighting and days with B lighting

- Do not inform customers of A to B swap or B to A swap in lighting

- Ensure product quality

- Use the same amount of voltage throughout the store for each lighting arrangement

- Keep the store stocked the same for everyday during experiment

- Use random days for each light fixture

- Maintain the same advertisements for the store during the study

- Do not inform employees of lighting swaps to ensure identical employee to customer relationships

- Compensate for any holiday or unexpected short days of store hours

- Have employees work randomized shifts to ensure no patterns in employees moods during light fixture swaps

- Employees have the same mindset to customers (if a retail business) during both A and B lighting arrangements

- Assume all data from A and B light fixtures have identical population variance

If lighting A and B are installed simultaneously, then it might be possible to even run different tests during the day, randomly allocated.

2. To ensure independence when investigating alternative shelving: A = solid white metal shelves and B = commercial stainless steel racking, the following experiments and conditions may be run:

- Shelving size remains the same and in the same location

- Identical product placement on both shelves A and B, if possible

- Being able to control everything other than the variable being studied of shelves

- Distances between shelves identical

- Ensure employees have the same mindset during each customer visit

- Identical number of items per shelf

- Same shelf distances from checkout

- Clean each shelf in the same manner for both A and B

- Keep prices and sales the same throughout the study period

Clearly the shelf study cannot be easily implemented, since the logistics of unstocking, removing shelf A, replacing with shelf B and restocking them is extremely costly.

One thing to consider in such cases is to run the experiments in two separate stores that are as similar as possible in all other respects (e.g. built in the area with similar profiles of customers, similar store layout, etc.).

Question 38

This question gives you exposure to analyzing a larger data set than seen in the preceding questions.

Your manager has asked you to describe the flow rate characteristics of the overhead stream leaving the top of the distillation column[54] at your plant. You are able to download one month of data, available from this website[55], from 1 March to 31 March, taken at one minute intervals to answer this question.

---

[54] https://en.wikipedia.org/wiki/Fractionating_column

[55] http://openmv.net/info/distillate-flow

3.1 Process monitoring in context

In the first section we learned about *visualizing data* (page 1), then we moved on to reviewing *univariate statistics* (page 29). This section now combines both topics, showing how to create a system that monitors a single, univariate, value from any process. These monitoring systems are easily implemented online, and generate great value for companies that use them in day-to-day production. This is one of their greatest advantages: almost no training is required to interpret the visualization and secondly the human eye can quickly pick up any patters or trends in the plots; both expected and unexpected patterns.

Monitoring charts are a graphical tool, enabling anyone to rapidly detect a problem by visual analysis. The next logical step after detection of a problem is to diagnose it, but we will cover diagnosis in the section on *latent variable models* (page 315).

This section is the last section where we deal with univariate data; after this section we start to use and deal with 2 or more variables.

3.1.1 Usage examples

Video for this section.

The material in this section is used whenever you need to rapidly detect problems. It has tangible application in many areas - in fact, you have likely encountered these monitoring charts in areas such as a hospital (monitoring a patient's heart beat), stock market charts (for intraday trading), or in a processing/manufacturing facility (control room computer screens).

- *Co-worker*: We need a system to ensure an important dimension on our product is stable and consistent over the entire shift.

- *Yourself*: We know that as the position of a manufacturing robot moves out of alignment that our product starts becoming inconsistent; more variable. How can we quickly detect this slow drift in alignment and predict when to stop the process and perform preventative maintenance?

- *Manager*: the hourly average profit, and process throughput is important to the head-office; can we create a system for them to track that?

- *Potential customer*: what is your process capability - we are looking for a new supplier that can provide a low-variability raw material for us with C_{pk} of at least 1.6, preferably higher.

Note: process monitoring is mostly *reactive* and not *proactive*. So it is suited to *incremental* process improvement, which is typical of most improvements. However, using the monitoring charts to make proactive changes to avoid a bigger problem later in time is certainly possible by adding additional rules and calculations to the plots. For example, rules to forecast a few steps ahead, with prediction intervals, can be easily added.

We point out in the *next section* (page 111) that process monitoring is not a feedback control system. So that section should be read in the context of thinking reactively and proactively (in a feed forward anticipatory manner).

3.1.2 What we will cover

We will consider 3 main charts after introducing some basic concepts: Shewhart charts, CUSUM charts and (exponentially weighted moving average) charts. The EWMA chart has an adjustable parameter that captures the behaviour of a Shewhart chart at one extreme and a CUSUM chart at the other extreme, or a combination of both is possible by settings this parameter on a sliding scale.

3.1.3 Concepts

Concepts and acronyms that you must be familiar with by the end of this section:

- Shewhart chart, CUSUM chart and EWMA chart

- Phase 1 and phase 2 when building a monitoring system

- False alarms

- Type 1 and type 2 errors

- LCL and UCL

- Target

- C_p and C_{pk}

- Outliers

- Real-time implementation of monitoring systems

3.2 References and readings

1. **Recommended**: Box, Hunter and Hunter, *Statistics for Experimenters*, Chapter 14 (2nd edition)

2. **Recommended**: Montgomery and Runger, *Applied Statistics and Probability for Engineers*.

3. Hunter, J.S. "The Exponentially Weighted Moving Average[56]", *Journal of Quality Technology*, **18** (4) p 203 - 210, 1986.

4. MacGregor, J.F. "Using On-Line Process Data to Improve Quality: Challenges for Statisticians[57]", *International Statistical Review*, **65**, p 309-323, 1997.

---

[56] https://asq.org/quality-resources/articles/the-exponentially-weighted-moving-average?id=27d7a4ac83cf47a18df2d09729369f41
[57] https://dx.doi.org/10.1111/j.1751-5823.1997.tb00311.x

3.3 What is process monitoring about?

Most industries have now realized that product quality is not an option. There was historical thinking that quality is the equivalent of "gold-plating" your product, but that has mostly fallen away. Product quality is not always a cost-benefit trade-off: it is beneficial to you in the long-term to improve your product quality, and for your customers as well.

As we spoke about in the *univariate review section* (page 29), good quality products (low variability) actually boost your profits by lowering costs in the long term. You have lower costs when you *do not* have to scrap off-specification product, or have to rework bad product. You have increased long-term sales with more loyal customers and improved brand reputation as a reliable and consistent supplier.

An example that most people in North America can relate to is the rise in Asian car manufacturers' market share, at the expense American manufacturers' market share. The market has the perception that Asian cars are more reliable than American cars and resale rates certainly reflect that. The perception has started to change since 2010, as North American manufacturers have become more quality conscious. That is an illustration of how lack of variability in your product can benefit you.

In order to achieve this high level of final product quality, our systems should be producing low variability product at every step of the manufacturing process. Rather than wait till the end of the process to *discover* poor quality product, we should be monitoring, in real-time, the purchased raw materials and also the intermediate steps in our process. When we discover unusual variability the lofty aim is to make (permanent) process adjustments to avoid that variability from ever occurring again.

Notice here that process monitoring is not intended to be automatic feedback control. It has the same principles of quantifying unusual operation (errors) and reacting to them in some way, but the intention with *process monitoring* is:

- that any process adjustments are **infrequent** [not frequently on a set cycle, as feedback control does],

- these adjustments are made **manually** [not automatically with actuators],

- and take place due to **special causes** [not due to regularly occurring process disturbances].

As seen by the items in square brackets above, automatic feedback control is applied continuously by computer systems and makes short-term, temporary changes to the system to keep it at the desired target (called the setpoint) in the face of process disturbances. Process monitoring is very different therefore to feedback control.

Note that process monitoring is often called statistical process control (SPC). This can lead to unnecessary confusion with process control, i.e. the design and implementation of feedback control, feed-forward control and other automated control systems. We will not use the term SPC, rather we will use the term *process monitoring*.

3.3.1 Monitoring charts

We use monitoring charts, also called control charts, to display and detect this unusual variability. A monitoring chart is a display of one value (variable), against time, or in sequence order. These time-based plots also show some additional information: usually a target value, and one or more limits lines are superimposed on the plot. The plots are most useful when displayed in real-time, or close to real-time. There are various technical ways to express what a monitoring chart does exactly, but a general definition is that a monitoring chart helps you detect outliers and other unusual time-based behaviour.

The key points are that a monitoring chart:

- is most often a time-series plot, or some sort of sequence plot,

- a target value (center line) may be shown,

- one or more limit lines are shown, such as the UCL (upper control limit) or LCL (lower control limit),

- they are displayed and updated in real-time, or as close to real-time as possible, so that the chart appears to move from right to left.

Here is an example that shows these properties.

Tank temperature, TC241 [degC]
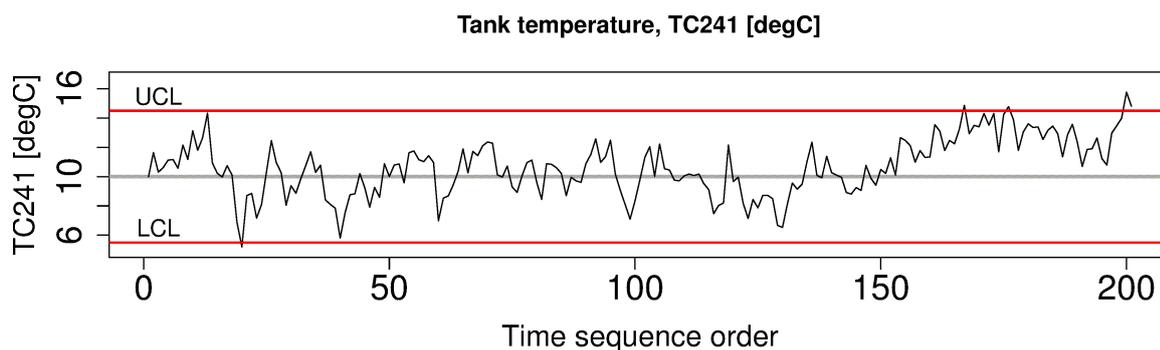
3.3.2 General approach

Monitoring charts are developed in 2 phases. You will see the terminology of:

- **Phase 1**: building and testing the chart from historical data that you have collected. This phase is performed off-line, it is very iterative, and you will spend most of your time here. The primary purpose of this phase is to

  - find portions of the data that are from stable operation

  - use these stable portions to calculate suitable control chart limits

  - ensure that your chart works as expected based on historical data

- **Phase 2**: We use the monitoring chart on new, fresh data from the process. This phase is implemented with computer hardware and software for real-time display of the charts. This phase is skipped if the phase 1 testing is not successful (e.g. too many false alarms). We discuss reasons for failure in the section on *judging the chart's performance* (page 118).

3.3.3 What should we monitor?

Any variable can be monitored. However, the purpose of process monitoring is so that you can **react early** to bad, or unusual operation. This implies we should monitor variables as soon as they become available, preferably in real-time. They are more suitable than variables that take a long time to acquire (e.g. laboratory measurements). We should not have to wait to the end of the production line to find our process was out of statistical control.

Data/measurements available at the start of your process, suc as raw material data from your supplier should also be monitored as soon as it is available, e.g. when received by your company, or even earlier - before the supplier ships it to you.

Intermediate variables measured from sensors at all points along the production process are (a) available much more frequently and without delay, (b) are more precise, (c) are usually more meaningful to the operating staff than final quality variables from the lab, and (d) contain the "fingerprint" of the fault, helping the engineers with diagnosis of what the problem is and point to which part(s) of the process need adjustment (see *MacGregor, 1997*).

Note that we do not have to monitor variables that are measured only from on-line sensors. The variable could be a calculation made from the on-line measurements.

For example, an energy balance could be calculated from various thermocouples on the process and the degree of mismatch in the energy balance could be critical to quality. For example, the mismatch could indicate an unexpected source of heat into or out of the process - so monitor that mismatch, rather than the raw temperature data. Similarly, a mass balance can be monitored in real-time, such as a total mass balance, or a carbon (or other elemental) balance. This is common in the mining industry and bio-processing industries.

Discuss one of these unit operations with your colleague. Which variables would you monitor?

- Waste water treatment process
- Tablet/pharmaceutical manufacturing
- Oil and gas (e.g. a distillation column)
- Food-processing or bio-engineering (e.g. fermentation) unit
- Mineral processing plant (e.g. a flotation cell)
- Plastics processing (e.g. a twin-screw extruder)

3.3.4 In-control vs out-of-control

Every book on quality control gives a slightly different viewpoint, or uses different terminology for these terms.

In this book we will take "in-control" to mean that the behaviour of the process is stable over time. Note though, that in-control *does not* mean the variable of interest meets the specifications required by the customer, or set by the plant personnel. All that "in control" means is that there are no **special causes** in the data, i.e. the process is stable. A special cause, or an assignable cause is an event that occurs to move the process, or destabilize it. Process monitoring charts aim to detect such events. The opposite of "special cause" operation is common cause operation, or stable process operation.

Note: Our objective: quickly detect abnormal variation, and fix it by finding the root cause. In this section we look at the "detection" problem. Diagnosis and process adjustment are two separate steps that follow detection.


3.4 Shewhart charts

A Shewhart chart, named after Walter Shewhart from Bell Telephone and Western Electric, monitors that a process variable remains on target and within given upper and lower limits. It is a monitoring chart for *location*. It answers the question whether the variable's location is stable over time. It does not track anything else about the measurement, such as its standard deviation. Looking ahead: *we show later* (page 119) that a pure Shewhart chart needs extra rules to help monitor the location of a variable effectively.

The defining characteristics of a Shewhart chart are: a target, upper and lower control limits (UCL and LCL). These action limits are defined so that no action is required as long as the variable plotted remains within the limits. In other words a special cause is not likely present if the points remain within the UCL and LCL.

3.4.1 Derivation using theoretical parameters

Define the variable of interest as x, and assume that we have samples of x available in sequence order. No assumption is made regarding the distribution of x. The average of n of these x-values is defined as $\overline{x}$, which from the *Central limit theorem* (page 44) we know will be more normally distributed with unknown population mean μ and unknown population variance σ^2/n, where μ and σ refer to the distribution that samples of x came from. The figure here shows the case for $n = 5$.
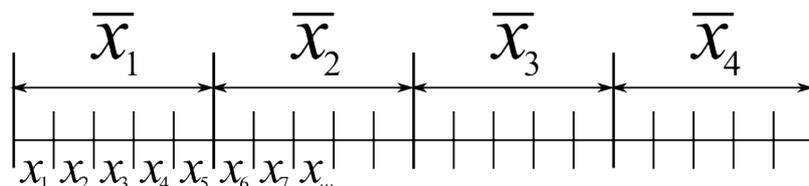
So by taking subgroups of size n values, we now have for each subgroup a newly calculated variable, $\overline{x}$ and we will define a shorthand symbol for its standard deviation: $\sigma_{\overline{X}} = \sigma/\sqrt{n}$. Writing a z-value for $\overline{x}$, and its associated confidence interval for μ is now easy after studying *the section on confidence intervals* (page 63):

$$z = \frac{\overline{x} - \mu}{\sigma_{\overline{X}}}$$

Assuming we know $\sigma_{\overline{X}}$, which we usually do not in practice, we can invoke the normal distribution and calculate the probability of finding a value of z between $c_n = -3$ to $c_n = +3$:

$$-c_n \quad \leq \quad \frac{\overline{x} - \mu}{\sigma_{\overline{X}}} \quad \leq \quad +c_n$$

$$\overline{x} - c_n\sigma_{\overline{X}} \quad \leq \quad \mu \quad \leq \quad \overline{x} + c_n\sigma_{\overline{X}} \tag{3.1}$$

$$\text{LCL} \quad \leq \quad \mu \quad \leq \quad \text{UCL}$$

The reason for $c_n = \pm 3$ is that the total area between that lower and upper bound spans 99.73% of the area (in R: `pnorm(+3) - pnorm(-3)` gives 0.9973). So it is highly unlikely, a chance of 1 in 370, that a data point, $\overline{x}$, calculated from a subgroup of n raw x-values, will lie outside these bounds.

The following illustration should help connect the concepts: the raw data's distribution happens to have a mean of 6 and standard deviation of 2, while it is clear the distribution of the subgroups of 5 samples (thicker line) is much narrower.

Shewhart chart: using theoretical (usually unknown) parameters
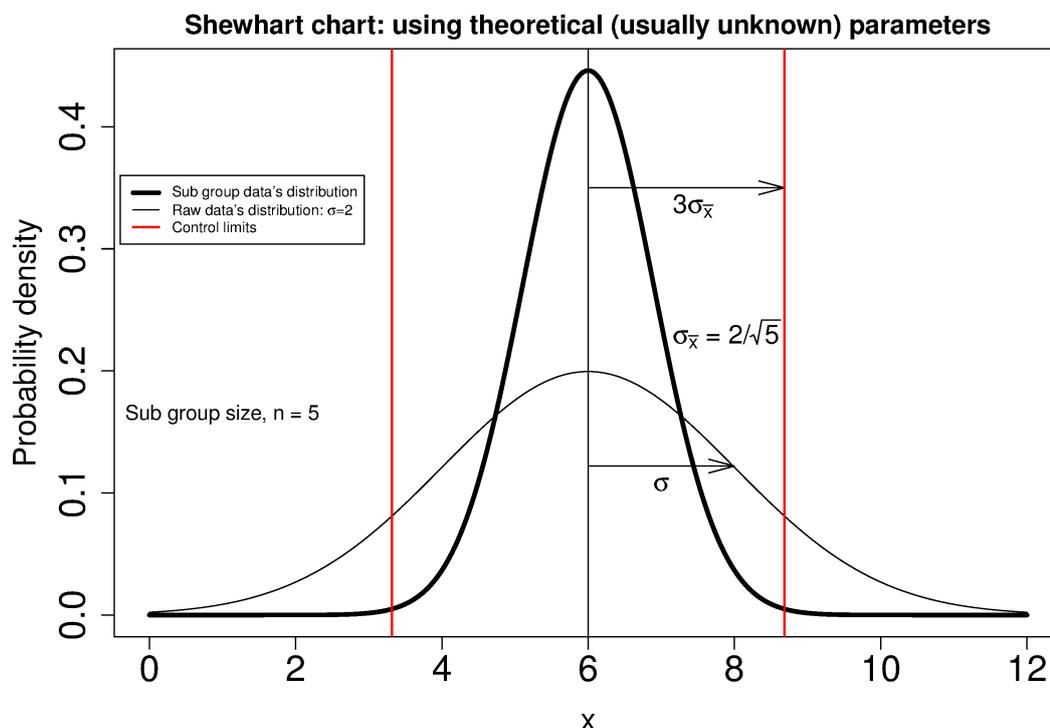
3.4.2 Using estimated parameters instead

The derivation in equation (3.1) requires knowing the population variance, σ, and assuming that our target for x is μ. The latter assumption is reasonable, but we will estimate a value for σ instead, using the data.

Let's take a look at phase 1, the step where we are building the monitoring chart's limits from historical data. Create a new variable $\overline{\overline{x}} = \dfrac{1}{K}\sum_{k=1}^{K} \overline{x}_k$, where K is the number of $\overline{x}$ samples we have available to build the monitoring chart, called the phase 1 data. Note that $\overline{\overline{x}}$ is sometimes called the *grand mean*. Alternatively, just set $\overline{\overline{x}}$ to the desired target value for x or use a long portion of stable data to estimate a suitable target

The next hurdle is σ. Define s_k to be the standard deviation of the n values in the kth subgroup. We do not show it here, but for a subgroup of n samples, an unbiased estimator of σ is given by $\dfrac{\overline{S}}{a_n}$, where $\overline{S} = \dfrac{1}{K}\sum_{k=1}^{K} s_k$ is simply the average standard deviation calculated from K subgroups. Values for a_n are looked up from a table, or using the formula below, and depend on the number of samples we use within each subgroup.

n	2	3	4	5	6	7	8	10	15
a_n	0.7979	0.8862	0.9213	0.9400	0.9515	0.9594	0.9650	0.9727	0.9823

More generally, using the $\Gamma(...)$ function, for example `gamma(...)` in R or MATLAB, or `math.gamma(...)` in Python, you can reproduce the above a_n values.

$$a_n = \frac{\sqrt{2}\,\Gamma(n/2)}{\sqrt{n-1}\,\Gamma(n/2-0.5)}$$

Notice how the a_n values tend to 1.0 the larger the subgroup size, indicating we need less of a correction to make the standard deviation less biased. Once we have this unbiased estimator for the standard deviation from these K subgroups, we can write down suitable lower and upper control limits for the Shewhart chart:

$$\text{LCL} = \bar{\bar{x}} - 3 \cdot \frac{\overline{S}}{a_n\sqrt{n}} \qquad \text{UCL} = \bar{\bar{x}} + 3 \cdot \frac{\overline{S}}{a_n\sqrt{n}} \tag{3.2}$$

It is highly unlikely that all the data chosen to calculate the phase 1 limits actually lie within these calculated LCL and UCLs. Those portions of data not from stable operation, which are outside the limits, should not have been used to calculate these limits. Those unstable data bias the limits to be wider than required.

Exclude these outlier data points and recompute the LCL and UCLs. Usually this process is repeated 2 to 3 times. It is wise to investigate the data being excluded to ensure they truly are from unstable operation. If they are from stable operation, then they should not be excluded. These data may be *violating the assumption of independence* (page 120). One may consider using wider limits, or use an *EWMA control chart* (page 123).

Example

Bales of rubber are being produced, with every 10th bale automatically removed from the line for testing. Measurements of colour intensity are made on 5 sides of that bale, using calibrated digital cameras under controlled lighting conditions. The rubber compound is used for medical devices, so it needs to have the correct colour, as measured on a scale from 0 to 255. The average of the 5 colour measurements is to be plotted on a Shewhart chart. So we have a new data point appearing on the monitoring chart after every 10th bale.

In the above example the raw data are the bale's colour. There are $n = 5$ values in each subgroup. Collect say $K = 20$ samples of good production bales considered to be from stable operation. No special process events occurred while these bales were manufactured.

The data below represent the average of the $n = 5$ samples from each bale, there are $K = 20$ of these subgroups.

$$\bar{x} = [245, 239, 239, 241, 241, 241, 238, 238, 236, 248, 233, 236, 246, 253, 227, 231, 237, 228, 239, 240]$$

The overall average is $\bar{\bar{x}} = 238.8$ and $\overline{S} = 9.28$. The raw data are available on this website[58] and you can verify the values of $\bar{\bar{x}}$ and $\overline{S}$ were correctly calculated.

- Calculate the lower and upper control limits for this Shewhart chart.

- Were there any points in the phase 1 data (training phase) that exceeded these limits?

---

[58] http://openmv.net/info/rubber-colour

- LCL $= \overline{\overline{x}} - 3 \cdot \dfrac{\overline{S}}{a_n \sqrt{n}} = 238.8 - 3 \cdot \dfrac{9.28}{(0.94)(\sqrt{5})} = 225.6$

- UCL $= \overline{\overline{x}} + 3 \cdot \dfrac{\overline{S}}{a_n \sqrt{n}} = 238.8 + 3 \cdot \dfrac{9.28}{(0.94)(\sqrt{5})} = 252.0$

- The group with $\overline{x} = 253$ exceeds the calculated upper control limit.

- That $\overline{x}$ point should be excluded and the limits recomputed. You can show the new $\overline{\overline{x}} = 238.0$ and $\overline{S} = 9.68$ and the new LCL $= 224$ and UCL $= 252$.

In source code:

```R
# Given information (but calculate yourself
# from http://openmv.net/info/rubber-colour)
xbar = c(245, 239, 239, 241, 241, 241, 238,
         238, 236, 248, 233, 236, 246, 253,
         227, 231, 237, 228, 239, 240)

# Number of measurements per subgroup
N.sub = 5

# Average of the 20 standard deviations
# of the 20 subgroups
S = 9.28

# xdb = x double bar = overall mean =
#       mean of the means
xdb = mean(xbar)

num.an = sqrt(2) * gamma(N.sub/2)
den.an = sqrt(N.sub-1) * gamma((N.sub-1)/2)
an = num.an / den.an

LCL = xdb - (3 * S/(an * sqrt(N.sub)))
UCL = xdb + (3 * S/(an * sqrt(N.sub)))
paste0('Control limits: [', round(LCL, 2),
       '; ', round(UCL,2), ']')

paste0('Number > UCL: ', sum(xbar > UCL))
paste0('Number < LCL: ', sum(xbar < LCL))

# Exclude the one subgroup above the UCL.
# Do this by setting it to 'NA' (missing)
xbar[xbar > UCL] = NA

# Calculate the mean, removing missing
# values (ignore it).
xdb = mean(xbar, na.rm=TRUE)

# 'S' will change also. If you download the
# raw data (link above), you can prove
# that the new 'S' will be:
S = 9.68

# The 'an' and 'N.sub' will not change.

LCL = xdb - (3 * S/(an * sqrt(N.sub)))
UCL = xdb + (3 * S/(an * sqrt(N.sub)))
paste0('Control limits: [', round(LCL, 0),
       '; ', round(UCL,0), ']')
```

3.4.3 Judging the chart's performance

There are 2 ways to judge performance of a monitoring chart. In particular here we discuss the Shewhart chart:

1. Error probability.

We define two types of errors, Type I and Type II, which are a function of the lower and upper control limits (LCL and UCL).

You make a **type I error** when your sample is typical of normal operation, yet, it falls outside the UCL or LCL limits. We showed in the theoretical derivation that the area covered by the upper and lower control limits is 99.73%. The probability of making a type I error, usually denoted as α is then $100 - 99.73 = 0.27\%$.

Synonyms for a **type I error**: false alarm, false positive (used mainly for testing of diseases), producer's risk (used for acceptance sampling, because here as the producer you will be rejecting an acceptable sample), false rejection rate, or alpha.

You make a **type II error** when your sample really is abnormal, but falls within the the UCL and LCL limits and is therefore not detected. This error rate is denoted by β, and it is a function of the degree of abnormality, which we derive next.

Synonyms for a **type II error**: false negative (used mainly for testing of diseases), consumer's risk (used for acceptance sampling, because your consumer will be receiving available product which is defective), false acceptance rate, or beta.

To quantify the probability β, recall that a Shewhart chart is for monitoring location, so we make an assumption that the new, abnormal sample comes from a distribution which has shifted its location from μ to $\mu + \Delta\sigma$ (e.g. Δ can be positive or negative). Now, what is the probability this new sample, which come from the shifted distribution, will fall within the existing LCL and UCL? This figure shows the probability is $\beta = (1 - \text{the shaded area})$.

$$\alpha = Pr\left(\overline{x} \text{ is in control, but lies outside the limits}\right) = \text{type I error rate}$$

$$\beta = Pr\left(\overline{x} \text{ is not in control, but lies inside the limits}\right) = \text{type II error rate}$$
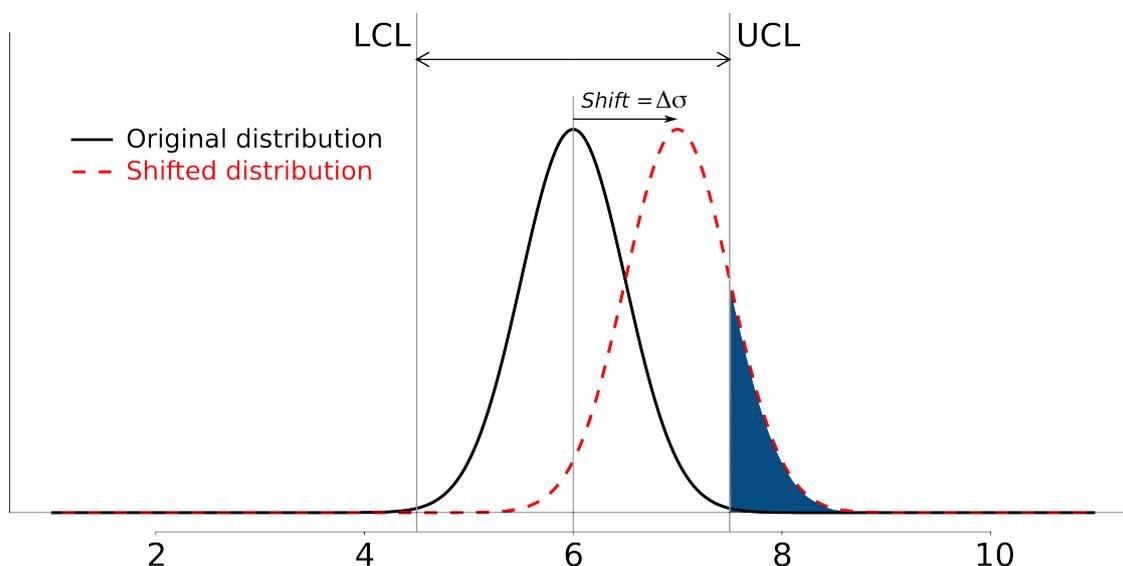
The table highlights that β is a function of the amount by which the process shifts = Δ, where $\Delta = 1$ implies the process has shifted up by 1σ. The table was calculated for $n = 4$ and used critical limits of $\pm 3\sigma_{\overline{X}}$. You can calculate your own values of β using this line of R code: `beta <- pnorm(3 - delta*sqrt(n)) - pnorm(-3 - delta*sqrt(n))`

Δ	0.25	0.50	0.75	1.00	1.50	2.00
β when $n = 4$	0.9936	0.9772	0.9332	0.8413	0.5000	0.1587

```
                                    R code
delta <- 1
n <- 4
beta <- pnorm(+3 - delta*sqrt(n)) -
        pnorm(-3 - delta*sqrt(n))

paste0('When delta=', delta, ' and n=', n,
       ' then beta = ', round(beta, 4))
```

The key point you should note from the table is that a Shewhart chart is *not good* (it is slow) at detecting a change in the location (level) of a variable. This is surprising given the intention of the plot is to monitor the variable's location. Even a moderate shift of 0.75σ units ($\Delta = 0.75$) will only be detected around 6.7% of the time ($100 - 93.3\%$) when $n = 4$. We will discuss *CUSUM charts* (page 121) and the Western Electric rules, next, as a way to overcome this issue.

It is straightforward to see how the type I, α, error rate can be adjusted - simply move the LCL and UCL up and down, as required, to achieve your desired error rates. There is nothing wrong in arbitrarily shifting these limits - *more on this later* (page 120) in the section on adjusting limits.

However what happens to the type II error rate as the LCL and UCL bounds are shifted away from the target? Imagine the case where you want to have $\alpha \to 0$. As you make the UCL higher and higher, the value for α drops, but the value for β will also increase, since the control limits have become wider! **You cannot simultaneously have low type I and type II error**, or as said more colloquially, "there is no free lunch".

2. Using the average run length (ARL)

The average run length (ARL) is defined as the average number of sequential samples we expect before seeing an out-of-bounds, or out-of-control signal. This is given by the inverse of α, as ARL = $\frac{1}{\alpha}$. Recall for the theoretical distribution we had $\alpha = 0.0027$, so the ARL = 370. Thus we expect a run of 370 samples before we get an out-of-control signal.

3.4.4 Extensions to the basic Shewhart chart to help monitor stability of the location

The Western Electric rules: we saw above how sluggish the Shewhart chart is in detecting a small shift in the process mean, from μ to $\mu + \Delta\sigma$. The **Western Electric rules** are an attempt to more rapidly detect a process shift, by raising an alarm when these *improbable* events occur:

1. Two out of 3 points lie beyond 2σ on the same side of the centre line

2. Four out of 5 points lie beyond 1σ on the same side of the centre line

3. Eight successive points lie on the same side of the center line

However, an alternative chart, the CUSUM chart is more effective at detecting a shift in the mean. Notice also that the theoretical ARL, $1/\alpha$, is reduced by using these rules in addition to the LCL and UCL bounds.

Adding robustness: the phase I derivation of a monitoring chart is iterative. If you find a point that violates the LCL and UCL limits, then the approach is to remove that point, and recompute the LCL and UCL values. That is because the LCL and UCL limits would have been biased up or down by these unusual points $\overline{x}_k$ points.

> This iterative approach can be tiresome with data that has spikes, missing values, outliers, and other problems typical of data pulled from a process database (historian). Robust monitoring charts are procedures to calculate the limits so the LCL and UCL are resistant to the effect of outliers. For example, a robust procedure might use the medians and MAD instead of the mean and standard deviation. An examination of various robust procedures, especially that of the interquartile range, is given in the paper by D. M. Rocke, Robust Control Charts[59], *Technometrics*, **31** (2), p 173 - 184, 1989.

> *Note*: do not use robust methods to calculate the values plotted on the charts during phase 2, only use robust methods to calculate the chart limits in phase 1!

Warning limits: it is common to see warning limits on a monitoring chart at $\pm 2\sigma$, while the $\pm 3\sigma$ limits are called the action limits. Real-time computer systems usually use a colour scheme to distinguish between the warning state and the action state. For example, the chart background changes from green, to orange to red as the deviations from target become more severe.

Adjusting the limits: The $\pm 3\sigma$ limits are not set in stone. Depending on the degree to which the source data obey the assumptions, and the frequency with which spikes and outliers contaminate your data, you may need to adjust your limits, usually wider, to avoid frequent false alarms. Nothing makes a monitoring chart more useless to operators than frequent false alarms ("crying wolf[60]"). However, *recall that there is no free lunch* (page 118): you cannot simultaneously have low type I and type II error.

Changing the subgroup size: It is perhaps a counterintuitive result that increasing the subgroup size, n, leads to a more sensitive detection system for shifts in the mean, because the control limits are pulled in tighter. However, the larger n also means that it will take longer to see the detection signal as the subgroup mean is averaged over more raw data points. So there is a trade-off between subgroup size and the run length (time to detection of a signal).

3.4.5 Mistakes to avoid

1. Imagine you are monitoring an aspect of the final product's quality, e.g. viscosity, and you have a product specification that requires that viscosity to be within, say 40 to 60 cP. It is a mistake to place those **specification limits** on the monitoring chart as a guide when to take action. It is also a mistake to use the required specification limits instead of the LCL and UCL. The monitoring chart is to detect abnormal variation in the process and gives a signal on when to take action, not to inspect for quality specifications. You can certainly have another chart for that, but the process monitoring chart's limits are intended to monitor process stability, and these Shewhart stability limits are calculated differently. Ideally the specification limits lie beyond the LCL and UCL action limits.

2. Shewhart chart limits were calculated with the assumption of **independent subgroups** (e.g. subgroup i has no effect on subgroup $i + 1$). For a process with mild autocorrelation, the act of creating subgroups, with n samples in each group, removes most, if not all, of the relationship

---

[59] https://dx.doi.org/10.2307/1268815
[60] https://en.wikipedia.org/wiki/The_Boy_Who_Cried_Wolf

between subgroups. However processes with heavy autocorrelation (slow moving processes sampled at a high rate, for example), will have LCL and UCL calculated from equation (3.2) that will raise false alarms too frequently. In these cases you can widen the limits, or remove the autocorrelation from the signal. More on this in the later section on *exponentially weighted moving average (EWMA) charts* (page 123).

3. Using Shewhart charts on two or more **highly correlated quality variables**, usually on your final product measurement, can increase your type II (consumer's risk) dramatically. We will come back to this very important topic in the section on *latent variable models* (page 397), where we will counterintuitively prove that even having individual charts each within their respective limits can result where it is outside the joint limits.

3.5 CUSUM charts

We *showed earlier* (page 119) that the Shewhart chart is not too sensitive to detecting shifts in the mean. Depending on the subgroup size, n, we showed that it can take several consecutive samples before a warning or action limit is triggered. The cumulative sum chart, or CUSUM chart, allows more rapid detection of these shifts away from a target value, T.

The following equation shows how this chart works.

$$
\begin{aligned}
S_0 &= (x_0 - T) \\
S_1 &= (x_0 - T) + (x_1 - T) = S_0 + (x_1 - T) \\
S_2 &= (x_0 - T) + (x_1 - T) + (x_2 - T) = S_1 + (x_2 - T)
\end{aligned}
\tag{3.3}
$$

In general $S_t = S_{t-1} + (x_t - T)$

Case 1, raw data: no shift in mean

Case 1: S(t)

Time steps

Case 2, a shift of 1.2 units (0.4 sigma) at t=150

Case 2: S(t)

Time steps
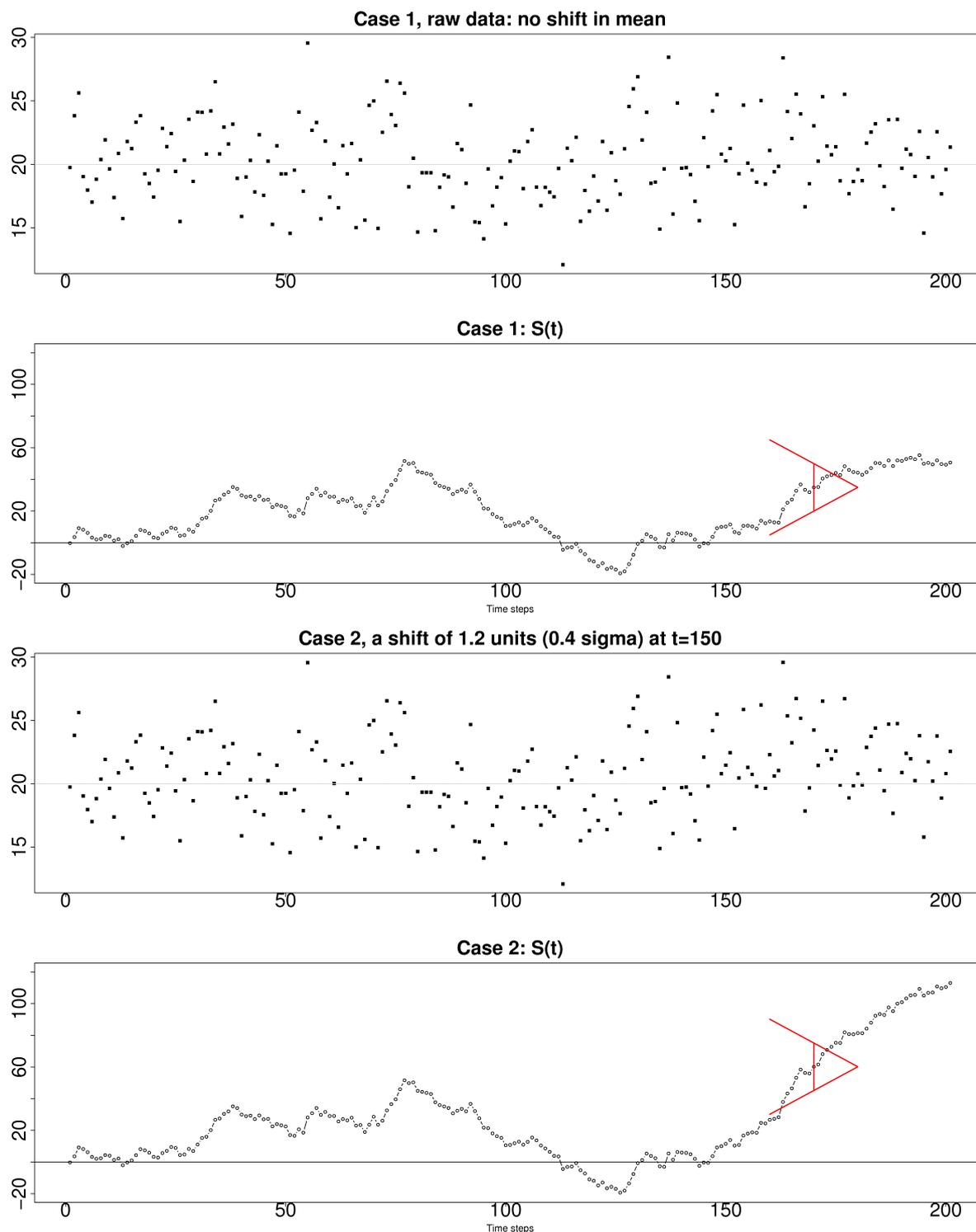
Values of S_t are the values plotted on the y-axis of a CUSUM chart. Imagine during a period of good, stable, in-control process operation around the target T, then these S_t numbers are just random errors, with mean of zero. The long-term sum of S_t is also zero, as the positive and negative errors keep cancelling out.

So imagine a CUSUM chart where at some time point the process mean shifts up by Δ units, causing future values of x_t to be $x_t + \Delta$ instead. Now the summation in the last equation of (3.3) has an extra Δ term added at each step to S_t. Every point will build up an accumulation of Δ, which shows up as a

positive or negative slope in the CUSUM chart.

The CUSUM chart is extremely sensitive to small changes. The example chart is shown here for a process where the mean is $\mu = 20$, and $\sigma = 3$. A small shift of $0.4 \times 3 = 1.2$ units (i.e from 20 to 21.2) occurs at $t = 150$. This shift is almost imperceptible in the raw data (see the 3rd row in the figure). However, the CUSUM chart rapidly picks up the shift by showing a consistent rising slope.

This figure also shows how the CUSUM chart is used with the 2 masks. Notice that there are no lower and upper bounds for S_t. A process that is on target will show a "wandering" value of S, moving up and down. In fact, as the second row in the figure shows, a surprising amount of movement up and down occurs even when the process is in control.

What is of interest however is a persistent change in slope in the CUSUM chart. The angle of the superimposed V-mask is the control limit: the narrower the mouth of the mask, the more sensitive the CUSUM chart is to deviations from the target. Both the type I and II error are set by the angle of the V and the leading distance (the distance from the short vertical line to the apex of the V).

The process is considered in control as long as all points are within the arms of the V shape. The mask in the second row of the plot shows "in control" behaviour, while the mask in the fourth row detects the process mean has shifted, and an alarm should be raised.

Once the process has been investigated the CUSUM value, S_t is often reset to zero; though other resetting strategies exist. A tabular version of the CUSUM chart also exists which tends to be the version used in software systems.

The purpose of this section is not to provide formulas for the V-mask or tabular CUSUM charts, only to explain the CUSUM concept to put the next section on EWMA control charts in perspective.

3.6 EWMA charts

The two previous charts highlight 2 extremes of monitoring charts. On the one hand, a Shewhart chart assumes each subgroup sample is independent (unrelated) to the next - implying there is no "memory" in the chart. On the other hand, a CUSUM chart has an infinite memory, all the way back to the time the chart was started or reset at $t = 0$ (see the *equation in the prior section* (page 121)).

As an introduction to the exponentially weighted moving average (EWMA) chart, consider first the simple moving average (MA) chart. This chart is used just like a Shewhart chart, except the samples that make up each subgroup are calculated using a moving window of width n. The case of $n = 5$ is shown below.
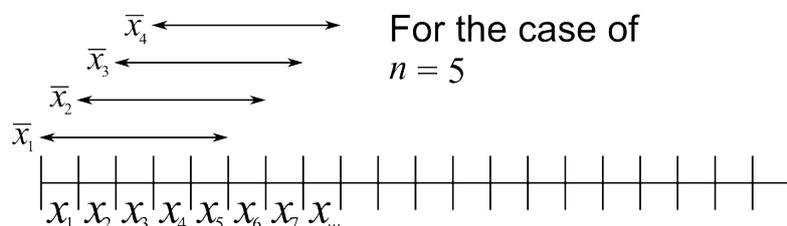
The MA chart plots values of $\overline{x}_t$, calculated from groups of size n, using equal weight for each of the n most recent raw data.

$$\overline{x}_t = \frac{1}{n}x_{t-1} + \frac{1}{n}x_{t-2} + \ldots + \frac{1}{n}x_{t-n}$$

The EWMA chart is similar to the MA chart, but uses different weights; heavier weights for more recent observations, tailing off exponentially to very small weights further back in history. Let's take a

look at a derivation.

Define the process target as T and define x_t as a new data measurement arriving now. We then try to *create an estimate of that incoming value, giving some weight, λ, to the actual measured value, and the rest of the weight, $1 - \lambda$, to the prior estimate*.

Let us write the estimate of x_t as $\hat{x}_t$, with the $\wedge$ mark above the x_t to indicate that it is a prediction of the actual measured x_t value. The prior estimate is therefore written as $\hat{x}_{t-1}$.

So putting into equation form that "an estimate of that incoming value, is given by some weight, λ and the rest of the weight, $1 - \lambda$, to the prior estimate":

$$
\begin{aligned}
\hat{x}_t &= \lambda x_t + (1 - \lambda)\,\hat{x}_{t-1} \\
\hat{x}_t &= \hat{x}_{t-1} + \lambda\,(x_t - \hat{x}_{t-1}) \\
\hat{x}_{t+1} &= \hat{x}_t + \lambda\,(x_{t+1} - \hat{x}_t) \\
\hat{x}_{t+1} &= \lambda x_{t+1} + (1 - \lambda)\,\hat{x}_t
\end{aligned}
\tag{3.4}
$$

To start the EWMA sequence we define the value for $\hat{x}_0 = T$ and $\hat{x}_1 = \lambda x_1 + T(1 - \lambda)$. A worked example is given further on in this section.

The last line in the equation group above shows that a 1-step-ahead prediction for x at time $t + 1$ is a weighted sum of two components: the current measured value, x_t, and secondly the predicted value, $\hat{x}_t$, with the weights summing up to 1. This gives a way to experimentally find a suitable λ value from historical data: adjust it up and down until the differences between $\hat{x}_{t+1}$ and the actual measured values of x_{t+1} are small.

The next plot shows visually what happens as the weight of λ is changed. In this example a shift of $\Delta = 1\sigma = 3$ units occurs abruptly at $t = 150$. This is of course not known in practice, but the purpose here is to illustrate the effects of choosing λ. Prior to that change the process mean is $\mu = 20$ and the raw data has $\sigma = 3$.

The first chart is the raw data and also a Shewhart chart with subgroup size of 1; the control limits are at ± 3 time the standard deviation, so at 11.0 and 19.0 units. This control chart barely picks up the shift, as was explained in a *prior section* (page 119).

The second, third and fourth charts are EWMA charts with different values of λ; the line is the value on the left-hand side of equation (3.4), in other words it is $\hat{x}_{t+1}$, the EWMA value at time t. We see that as λ decreases, the charts are smoother, since the averaging effect is greater: more and more weight is given to the history, $\hat{x}_t$, and less weight to the current data point, x_t. See equation (3.4) to understand that interpretation. Also note carefully how the control limits become narrower as the λ decreases, as is explained shortly below.

To see why $\hat{x}_t$ represents historical data, you can recursively substitute and show that:

$$
\hat{x}_{t+1} = \sum_{i=0}^{i=t} w_i x_i = w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots
$$

where the weights are: $\quad w_i = \lambda(1 - \lambda)^{t-i}$

which emphasizes that the prediction is a just a weighted sum of the raw measurements, with weights declining in time.

The final chart of the sequence of 5 charts is a CUSUM chart, which is *the ideal chart* (page 121) for picking up such an abrupt shift in the level.

Raw data (and also a Shewhart monitoring chart; limits at 3 sigma)

EWMA with λ = 0.8

EWMA with λ = 0.4

EWMA with λ = 0.1

CUSUM

Time steps
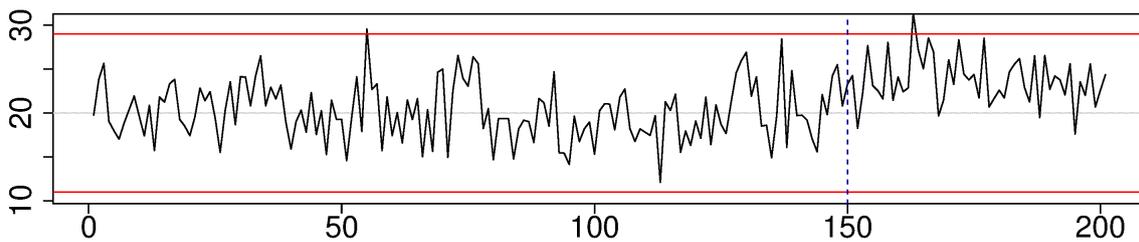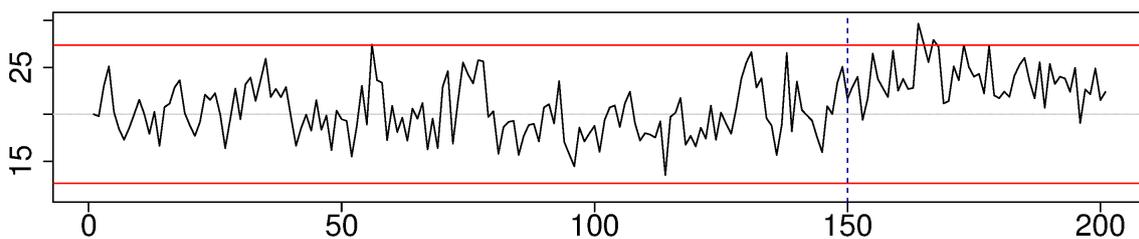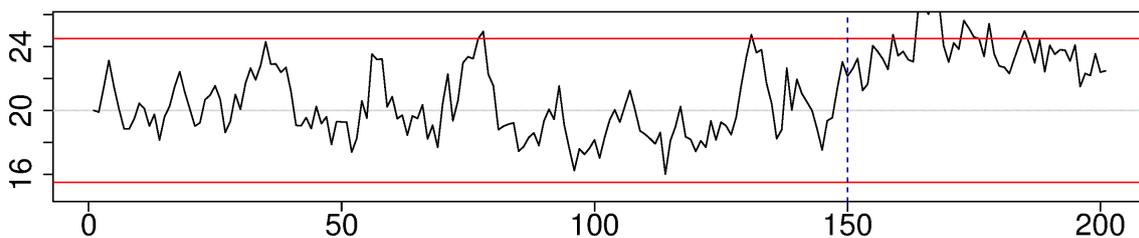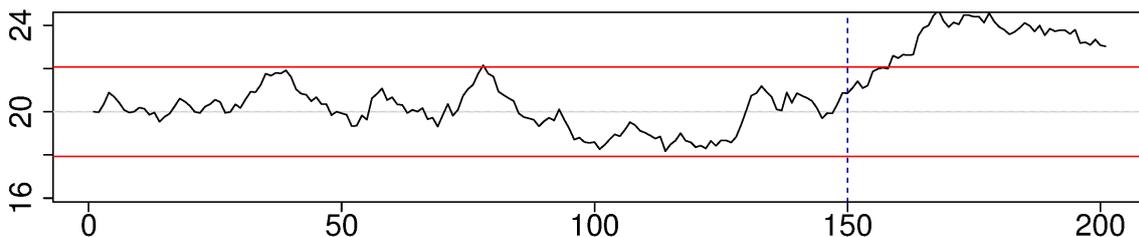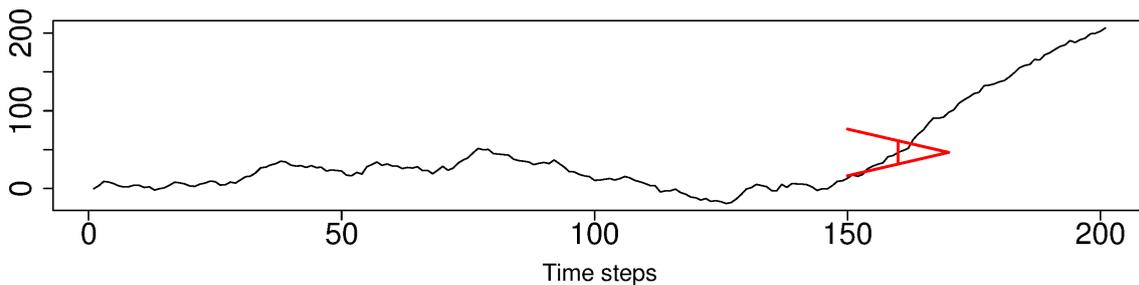
In the next figure, we show a comparison of the weights used in different monitoring charts studied so far.

From the above discussion and the weights shown for the 4 different charts, it should be clear now how an EWMA chart is a tradeoff between a Shewhart chart and a CUSUM chart. As $\lambda \to 1$, the EWMA chart behaves more as a Shewhart chart, giving only weight to the most recent observation. While as $\lambda \to 0$ the EWMA chart starts to have an infinite memory (like a CUSUM chart). There are 12 data points used in the example, so the CUSUM 'weight' is one twelfth or ≈ 0.0833.

Shewhart weights

CUSUM weights; each weight is 0.0833

MA weights when N=5

EWMA weights when λ = 0.6
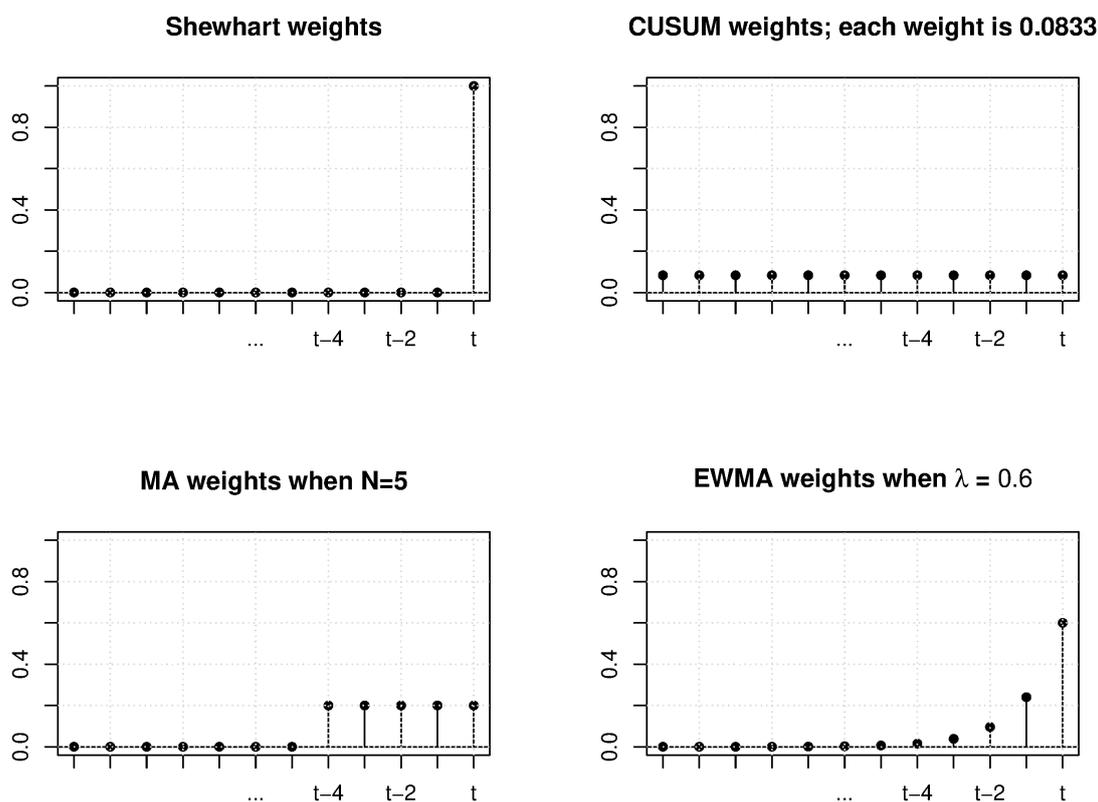
The upper and lower control limits for the EWMA plot are plotted in the same way as the Shewhart limits, but calculated differently:

$$\text{LCL} = \bar{\bar{x}} - K \cdot \sigma_{\text{Shewhart}} \sqrt{\frac{\lambda}{2 - \lambda}} \qquad \text{UCL} = \bar{\bar{x}} + K \cdot \sigma_{\text{Shewhart}} \sqrt{\frac{\lambda}{2 - \lambda}} \qquad (3.5)$$

where σ_{Shewhart} represents the standard deviation as calculated for the Shewhart chart. K is usually a value of 3, similar to the 3 standard deviations used in a Shewhart chart, but can of course be set to any level that balances the type I (false alarms) and type II errors (not detecting a deviation which is present already).

An interesting implementation can be to show both the Shewhart and EWMA plot on the same chart, with both sets of limits. The EWMA value plotted is actually the one-step ahead prediction of the next x-value, which can be informative for slow-moving processes.

The R code here shows one way of calculating the EWMA values for a vector of data. Once you have pasted this function into R, use it as `ewma(x, lambda=..., target=...)`.

```R code
ewma <- function(x, lambda, target=x[1]){
    N <- length(x)
    y <- numeric(N)
    y[1] = target
    for (k in 2:N){
        error = x[k - 1] - y[k - 1]
        y[k] = y[k - 1] + lambda*error
    }
return(y)
}

# Try using this function now:
x <- c(200, 210, 190, 190, 190, 190)
ewma(x, lambda = 0.3, target = 200)
```

Here is a worked example, starting with the assumption the process is at the target value of $T = 200$ units, and $\lambda = 0.3$. We intentionally show what happens if the new value stays fixed at 190: you see the value plotted gets only a weight of 0.3, while the 0.7 weight is for the prior historical value. Slowly the value plotted catches up, but there is always a lag. The value plotted on the chart is from the last equation in the set of (3.4).

Sample number	Raw data x_t	Value plotted on chart: $\hat{x}_t$
0	NA	200
1	200	$0.3 \times 200 + 0.7 \times 200 = 200$
2	210	$0.3 \times 210 + 0.7 \times 200 = 203$
3	190	$0.3 \times 190 + 0.7 \times 203 = 199.1$
4	190	$0.3 \times 190 + 0.7 \times 199.1 = 196.4$
5	190	$0.3 \times 190 + 0.7 \times 196.4 = 194.5$
6	190	$0.3 \times 190 + 0.7 \times 194.5 = 193.1$

3.7 Other types of monitoring charts

You may encounter other charts in practice:

- The *S chart* is for monitoring the subgroup's standard deviation. Take the group of n samples and show their standard deviation on a Shewhart-type chart. The limits for the chart are calculated using similar correction factors as were used in the derivation for the $\overline{x}$ Shewhart chart. This chart has a LCL ≥ 0.

- The *R chart* was a precursor for the *S chart*, where the *R* stands for range, the subgroup's maximum minus minimum. It was used when charting was done manually, as standard deviations were tedious to calculate by hand.

- The *np chart* and *p chart* are used when monitoring the proportion of defective items using a pass/fail criterion. In the former case the sample size taken is constant, while in the latter the proportion of defective items is monitored. These charts are derived using the binomial distribution.

- The *exponentially weight moving variance* (EWMV) chart is an excellent chart for monitoring for an increase in product variability. Like the λ from an EWMA chart, the EWMV also has a sliding parameter that can balance current information and historical information to trade-off sensitivity. More information is available in the paper by J.F. MacGregor, and T.J. Harris, "The Exponentially

Weighted Moving Variance[61]", *Journal of Quality Technology*, **25**, p 106-118, 1993.

3.8 Process capability

Note: This section is not about a particular monitoring chart, but is relevant to the topic of process monitoring.

Video for this section.

3.8.1 Centered processes

Purchasers of your product may request a process capability ratio (PCR) for each of the quality attributes of your product. For example, your plastic product is characterized by its Mooney viscosity and melting point. A PCR value can be calculated for either property, using the definition below:

$$\text{PCR} = \frac{\text{Upper specification limit} - \text{Lower specification limit}}{6\sigma} = \frac{\text{USL} - \text{LSL}}{6\sigma} \tag{3.6}$$

Since the population standard deviation, σ, is not known, an estimate of it is used. Note that the lower specification limit (LSL) and upper specification limit (USL) are **not the same** as the lower control limit (LCL) and upper control limit (UCL) as were calculated for the Shewhart chart. The LSL and USL are the tolerance limits required by your customers, or set from your internal specifications.

Interpretation of the PCR:

- assumes the property of interest follows a normal distribution

- assumes the process is centered (i.e. your long term mean is halfway between the upper and lower specification limits)

- assumes the PCR value was calculated when the process was stable

The PCR is often called the process width. Let's see why by taking a look at a process with PCR=0.5 and then PCR=2.0. In the first case $\text{USL} - \text{LSL} = 3\sigma$. Since the interpretation of PCR assumes a centered process, we can draw a diagram as shown below:
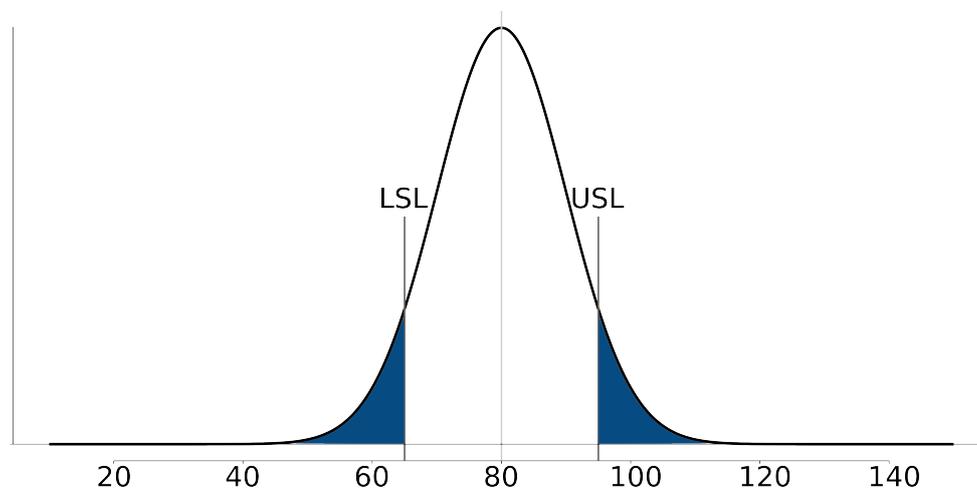
The diagram is from a process with mean of 80 and where LSL=65 and USL=95. These specification are fixed, set by our production guidelines. If the process variation is $\sigma = 10$, then this implies that

---

[61] https://learnche.org/literature/item/178/the-exponentially-weighted-moving-variance

PCR=0.5. Assuming further that the our production is centered at the mean of 80, we can calculate how much defective product is produced in the shaded region of the plot. Assuming a normal distribution:

- z for LSL = $(65 - 80)/10 = -1.5$

- z for USL = $(95 - 80)/10 = 1.5$

- Shaded area probability = `pnorm(-1.5) + (1-pnorm(1.5))` = 13.4% of production is out of the specification limits.

Contrast this to the case where PCR = 2.0 for the same system. To achieve that level of process capability, using the *same upper and lower specifications* we have to reduce the standard deviation by a factor of 4, down to $\sigma = 2.5$. The figure below illustrates that almost no off-specification product is produced for a centered process at PCR = 2.0. There is a width of 12σ units from the LSL to the USL, giving the process location (mean) ample room to drift left or right without creating additional off-specification product.
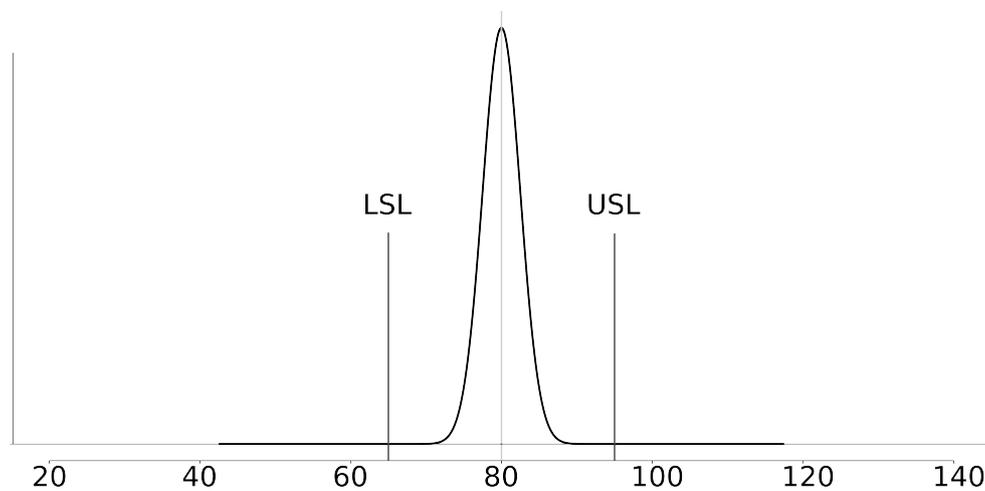
Note: You will probably come across the terminology C_p, especially when dealing with 6 sigma programs. This is the same as PCR for a centered process.

3.8.2 Uncentered processes

Processes are not very often centered between their upper and lower specification limits. So a measure of process capability for an uncentered processes is defined:

$$\text{PCR}_k = C_{pk} = \min \left(\frac{\text{Upper specification limit} - \bar{\bar{x}}}{3\sigma}; \frac{\bar{\bar{x}} - \text{Lower specification limit}}{3\sigma} \right) \qquad (3.7)$$

The $\bar{\bar{x}}$ term would be the process target from a Shewhart chart, or simply the actual average operating point. Notice that C_{pk} is a one-sided ratio, only the side closest to the specification is reported. So even an excellent process with C_p = 2.0 that is running off-center will have a lower C_{pk}.

It is the C_{pk} value that is requested by your customer. Values of 1.3 are usually a minimum requirement, while 1.67 and higher are requested for health and safety-critical applications. A value of $C_{pk} \geq 2.0$ is termed a six-sigma process, because the distance from the current operating point, $\bar{\bar{x}}$, to the closest specification is at least 6σ units.

You can calculate that a shift of 1.5σ from process center will introduce only 3.4 defects per million. This shift would reduce your C_{pk} from 2.0 to 1.5.

Note: It must be emphasized that C_{pk} and C_p numbers are only useful for a process which is stable. Furthermore the assumption of normally distributed samples is also required to interpret the C_{pk} value.

3.9 The industrial practice of process monitoring

This preceding section of the book is only intended to give an overview of the concepts of process monitoring. As you move into an industrial environment you will find there are many such systems already in place. Higher levels of management track statistics from a different point of view, often summarizing data from an entire plant, geographic region, or country. The techniques learned in this book, while focusing mainly on unit operations, are equally applicable though to data from a plant, region, or country.

You may come across systems called dashboards, which are often part of enterprise resource planning (ERP) systems. These dashboards are supposed to monitor the pulse of a company and are tracked like any other monitoring chart discussed above. Another area is called business intelligence (BI) systems. These typically track sales and other financial information.

Yet another acronym is the KPI, key performance indicator, which is a summary variable, such as profit per hour, or energy cost per unit of production. These are often monitored and acted on by site managers on a daily or weekly basis. Sites in a global company with the lowest KPIs receive the greatest scrutiny.

But at the unit operation and plant level, you will likely find the hardest part of getting a monitoring chart implemented is the part where you need access to the data. Getting data out of most database systems is not easy, though it has improved quite a bit in the last few years.

It is critical that your monitoring chart display the quantity as close to real-time as possible. It is almost as if the monetary value of the information in a monitoring chart decays exponentially from the time an event occurs. It is hard to diagnose and correct a problem detected yesterday, and harder still if the problem occurred last week or month.

You will also realize that good operator training to interpret and act on the monitoring chart is time-consuming; operators are often cycled between different units or plants, so frequent re-training is required. Concepts from the *data visualization* (page 1) section are helpful to minimize training effort - make sure the online plots contain the right level of information, without clutter, so they can be acted on accurately.

Another side effect of large quantities of data are that you will have to work with IT groups to manipulate large chunks of data on dedicated networks, separate from the rest of the plant. The last thing you want to be responsible for is clogging the company network with your traffic. Most industries now have a "production" network running in parallel to the "corporate" network. The production network carries real-time data, images from cameras and so forth, while the company network carries office email and web traffic.

3.9.1 Approach to implement a monitoring chart in an industrial setting

Here is some general guidance; feel free to adjust the steps as required for your unique situation.

1. Identify the variable(s) to monitor. Make sure each variable shows different, uncorrelated phenomena to avoid redundancy. If unsure which variables to select, use a *multivariate monitoring system* (page 397).

2. Retrieve historical data from your computer systems, or lab data, or paper records.

3. Import the data and just plot it. Do you see any time trends, outliers, spikes, missing data gaps? Investigate these (to learn more about your process), but then remove them to create the phase 1 data set.

4. Locate any regions of data which are from generally stable operation. Remove spikes and outliers that will bias your control limits calculations. In other words, find regions of common-cause operation.

5. Split your phase 1 data into say a 60% and 40% split. Set aside 40% of the cleaned portion to use as phase 1 testing data later on. Keep the data from outliers, spikes and unstable process operation aside as another testing data set (to ensure that these problems are actually detectable).

6. Using the cleaned 60% portion, estimate limits that you would expect to contain this stable region of operation just by looking at the plots.

7. On the 60% portion, calculate preliminary control limits (UCL, LCL), using the formulas shown in this section. They should agree with limits in the previous step.

8. How does your chart work? Test your chart on the 40% cleaned portion. These testing data should not raise many alarms. Any alarms raised will be type I errors, so you can quantify your type I error rate from the fraction of false alarms raised.

9. Test your chart on the unusual data you found earlier. You can quantify the type II error by counting the fraction of this bad data that went undetected by your chart.

10. Adjust the limits and monitoring chart parameters (e.g. λ) if necessary, to achieve the required type I and type II balance that is acceptable to your operation staff. You may even have to resort to using a different chart, or monitoring based on a different variable.

11. Test the chart on your desktop computer for a couple of days. When you detect an unusual event, go and check with the process operators and verify the event. Would they have reacted to it, had they known about it? Or, would this have been a false alarm? You may need to refine your limits, or the value you are plotting again.

12. Remember that this form of charting is not an expert system - it will not diagnose problems: you have to use your engineering knowledge by looking at patterns in the chart, and use knowledge of other process events.

13. Demonstrate the system to your colleagues and manager. But show them economic estimates of the value of early detection. They are usually not interested in the plots alone, so convert the statistics into monetary values. For example, dollars saved if we had detected that problem in real-time, rather than waiting till later.

14. Installation and operator training will take time. This assumes that you have real-time data acquisition systems and real-time processing systems in place - most companies do. You will have to work with your company's IT staff to get this implemented.

15. Listen to your operators for what they want to see. Use principles of *good data visualization* (page 1) to reduce unnecessary information. Make your plots interactive - if you click on an unusual point it should "drill-down" and give you more details and historical context.

16. Future monitoring charts are easier to get going, once the first system is in place.

3.10 Industrial case study

ArcelorMittal's steel mill in Hamilton, Ontario, (formerly called Dofasco) has used multivariate process monitoring tools in many areas of their plant for decades now. One of their most successful applications is that applied to their casting operation. In this section we only focus on the application; the sort of multivariate calculations used by this system are discussed *later on* (page 315).

The computer screenshot shows the monitoring system, called Caster SOS (Stable Operation Supervisor), which is followed by the operators. There are several charts on the screen: two charts, called "Stability Index 1" and "Stability Index 2", are one-sided monitoring charts. Notice the warning limits and the action limits. In the middle is a two-sided chart. A wealth of information is presented on the screen - their design was heavily influenced and iterated on several times, working with the *operators*. The screen shot is used with permission of Dr. John MacGregor.
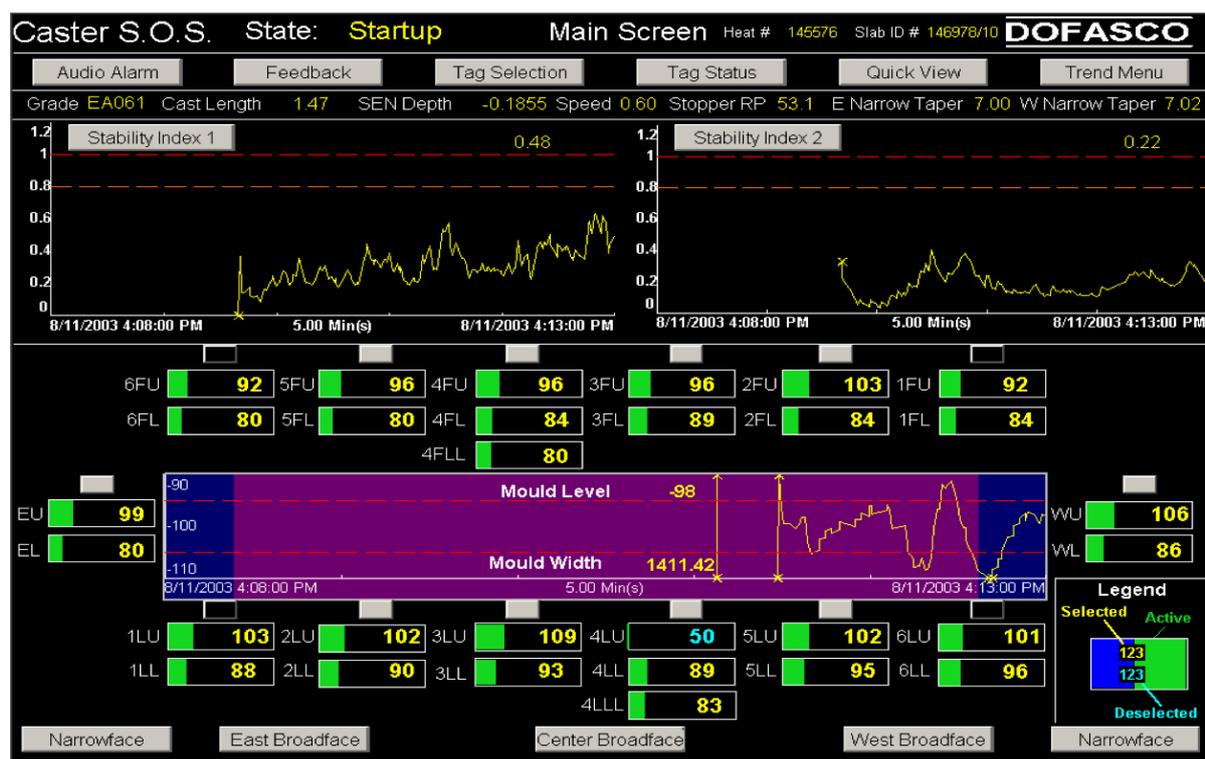
The economics of monitoring charts cannot be overstated. The ArcelorMittal example above was introduced around 1997. The calculations required by this system are complex - however the computer systems performs them in near real-time, allowing the operators to take corrective action within a few seconds. The data show a significant reduction in breakouts since 1997 (*used with permission of Dr. John MacGregor*).
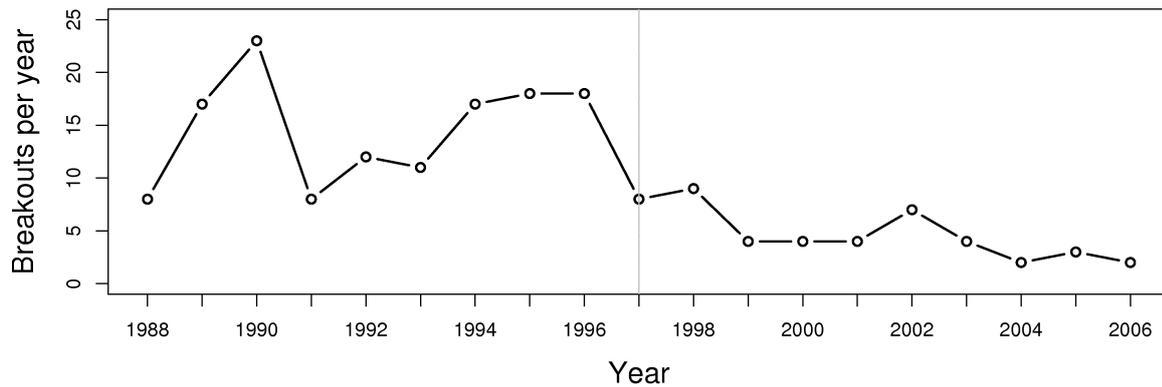
The economic savings and increased productivity is in the millions of dollars per year, as each breakout costs around $200,000 to $500,000 due to process shutdowns and/or equipment damage.

3.11 Summary

Montgomery and Runger list 5 reasons why monitoring charts are widely used. After this section of the book you should understand the following about the charts and process monitoring:

1. These tools are proven to improve productivity (i.e. to reduce scrap and rework, as described above), and to increase process throughput.

2. They detect defective production, consistent with the concept of "doing it right the first time", a mantra that you will increasingly hear in the manufacturing workplace.

3. A monitoring chart with good limits will prevent over-control of the process. Operators are trained not to make process adjustments unless there is a clear warning or alarm from the chart.

4. The patterns generated by the plots often help determine what went wrong, providing some diagnostic value to the operators. We will see a more formal tool for process diagnosis though in the *latent variable section* (page 315).

5. Monitoring charts are required to judge if a process is stable over time. A stable process allows us to calculate our process capability, which is an important metric for your customers.

3.12 Exercises

Question 1

Is it fair to say that a monitoring chart is like an online version of a *confidence interval* (page 63)? Explain your answer.

Question 2

Use the batch yields data[62] and construct a monitoring chart using the 300 yield values. Use a subgroup of size 5. Report your target value, lower control limit and upper control limit, showing the calculations you made. I recommend that you write your code so that you can reuse it for other questions.

Solution

Please see the code below. The Shewhart chart's parameters are as below, with plots generated from the R code.

- Target = 80.4

- Lower control limit at 3 standard deviations = 71.1

- Upper control limit at 3 standard deviations = 89.6
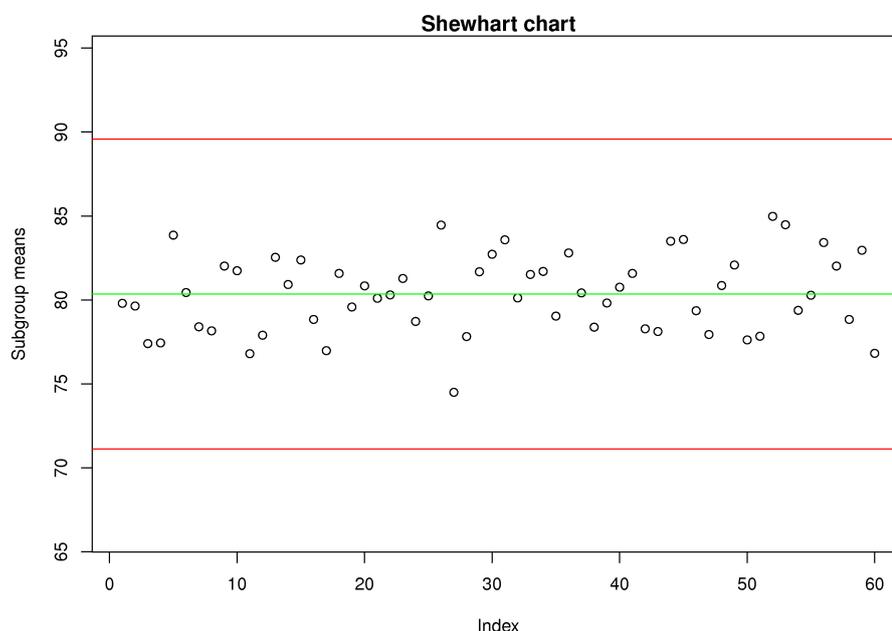
Try it yourself:

```
                                                    R code
data_file <- 'http://openmv.net/file/batch-yields.csv'
batch <- read.csv(data_file)

# make sure we have the expected data
summary(batch)
attach(batch)

# To get a feel for the data;
# looks pretty good; no unusual outliers
plot(Yield)

N = length(Yield)
N.sub = 5          # subgroup size
subgroup <- matrix(Yield, N.sub, N/N.sub)
N.groups <- ncol(subgroup)
dim(subgroup)      # 5 by 60 matrix
```

(continues on next page)

---

[62] http://openmv.net/info/batch-yields

```
subgroup.sd <- apply(subgroup, 2, sd)
subgroup.xbar <- apply(subgroup, 2, mean)

# Take a look at what these numbers mean
plot(subgroup.xbar,
     type="b",
     ylab="Subgroup average")
plot(subgroup.sd,
     type="b",
     ylab="Subgroup spread")

# Report your target value, lower control
# limit and upper control limit, showing
# the calculations you made.
target <- mean(subgroup.xbar)
Sbar <- mean(subgroup.sd)

# a_n value is from the table when
# subgroup size = 5
an <- 0.94
an.num <- sqrt(2)*gamma(N.sub/2)
an.den <- sqrt(N.sub-1)*gamma(N.sub/2-0.5)
an <- an.num/an.den
sigma.estimate <- Sbar / an
LCL <- target - 3 * sigma.estimate/sqrt(N.sub)
UCL <- target + 3 * sigma.estimate/sqrt(N.sub)
c(LCL, target, UCL)
plot(subgroup.xbar,
     ylim=c(LCL-5, UCL+5),
     ylab="Subgroup means",
     main="Shewhart chart")
abline(h=target, col="green")
abline(h=UCL, col="red")
abline(h=LCL, col="red")
```

Question 3

The boards data[63] on the website are from a line which cuts spruce, pine and fir (SPF) to produce general quality lumber that you could purchase at Rona, Home Depot, etc. The price that a saw mill receives for its lumber is strongly dependent on how accurate the cut is made. Use the data for the 2 by 6 boards (each row is one board) and develop a monitoring system using these steps.

a) Plot all the data.

b) Now assume that boards 1 to 500 are the phase 1 data; identify any boards in this subset that appear to be unusual (where the board thickness is not consistent with most of the other operation)

c) Remove those unusual boards from the phase 1 data. Calculate the Shewhart monitoring limits and show the phase 1 data with these limits. Note: choose a subgroup size of 7 boards.

d) Test the Shewhart chart on boards 501 to 2000, the phase 2 data. Show the plot and calculate the type I error rate (α) from the phase 2 data; assuming, of course, that all the phase 2 data are from in-control operation.

e) Calculate the ARL and look at the chart to see if the number looks about right. Use the time information in the raw data and your ARL value to calculate how many minutes between a false alarm. Will the operators be happy with this?

---

[63] http://openmv.net/info/six-point-board-thickness

f) Describe how you might calculate the consumer's risk (β).

g) How would you monitor if the saws are slowly going out of alignment?

Question 4

Your process with Cpk of 2.0 experiences a drift of 1.5σ away from the current process operating point towards the closest specification limit. What is the new Cpk value; how many defects per million items did you have before the drift? And after the drift?

Solution

The new Cpk value is 1.5. The number of defects per million items at Cpk = 2.0 is 0.00098 (essentially no defects), while at Cpk = 1.5 it is 3.4 defects per million items. You only have to consider one-side of the distribution, since Cpk is by definition for an uncentered process, and deals with the side closest to the specification limits.

```
                                    ─── R code ───
Cpk <- 1.5
n.sigma.distance <- 3 * Cpk
dpm <- pnorm(-n.sigma.distance,
             mean=0,
             sd=1) * 1E6
paste0('Defects per million = ', round(dpm,3))
```

Question 5

Which type of monitoring chart would be appropriate to detect unusual spikes (outliers) in your production process?

Solution

A Shewhart chart has no memory, and is suited to detecting unusual spikes in your production. CUSUM and EWMA charts have memory, and while they would pick up this spike, they would also create a long duration of false alarms after that. So those charts are much less appropriate.

Question 6

A tank uses small air bubbles to keep solid particles in suspension. If too much air is blown into the tank, then excessive foaming and loss of valuable solid product occurs; if too little air is blown into the tank the particles sink and drop out of suspension.
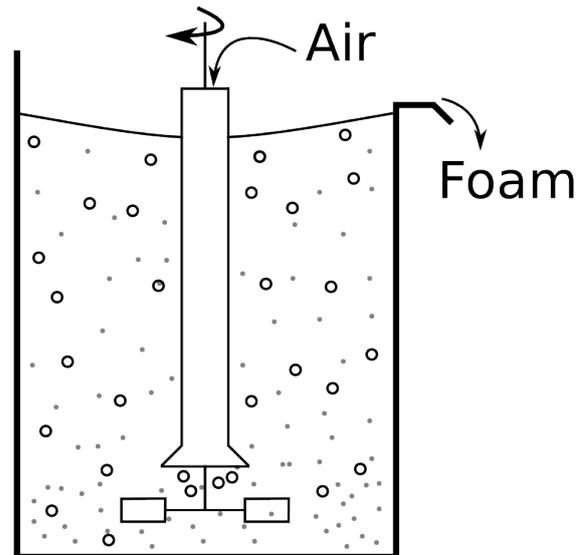
1. Which monitoring chart would you use to ensure the airflow is always near target?

2. Use the aeration rate dataset[64] from the website and plot the raw data (total litres of air added in a 1 minute period). Are you able to detect any problems?

3. Construct the chart you described in part 1, and show it's performance on all the data. Make any necessary assumptions to construct the chart.

4. At what point in time are you able to detect the problem, using this chart?

5. Construct a Shewhart chart, choosing appropriate data for phase 1, and calculate the Shewhart limits. Then use the entire dataset as if it were phase 2 data.

   - Show this phase 2 Shewhart chart.

   - Compare the Shewhart chart's performance to the chart in part 3 of this question.

Solution

Solution based on work by Ryan and Stuart (2011 class)

1. A CUSUM chart would be a suitable chart to monitor that the airflow is near target. While a Shewhart chart is also intended to monitor the location of a variable, it has a much larger run length for detecting small shifts. An EWMA chart with small λ (long memory) would approximate a CUSUM chart, and so would also be suitable

2. The aeration rate dataset is depicted below:

---

[64] http://openmv.net/info/aeration-rate
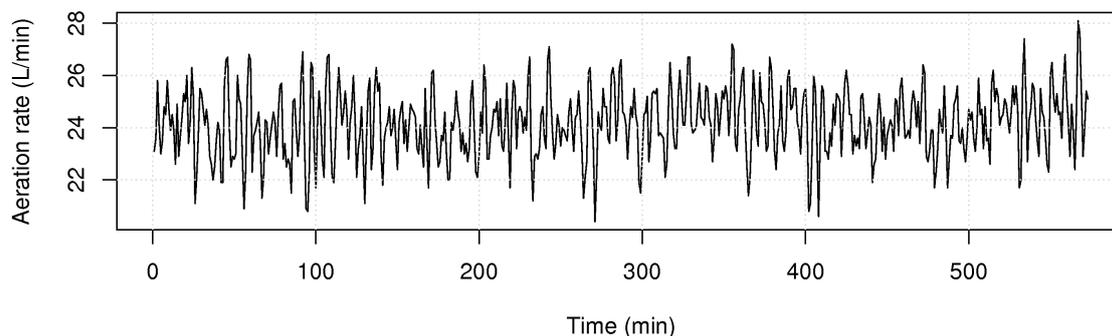
It is very difficult to assess problems from the raw data plot. There might be a slight upward shift around 300 and 500 minutes.

3. Assumptions for the CUSUM chart:

   • We will plot the CUSUM chart on raw data, though you could use subgroups if you wanted to.

   • The target value can be the mean (24.17) of all the data, or more robustly, use the median (24.1), especially if we expect problems with the raw data (true of almost every real data set).

4. The CUSUM chart, using the median as target value showed a problem starting to occur around $t = 300$. So we recalculated the median, using only data from 0 to $t = 200$, to avoid biasing the target value. Using this median instead, 23.95, we get the following CUSUM chart:
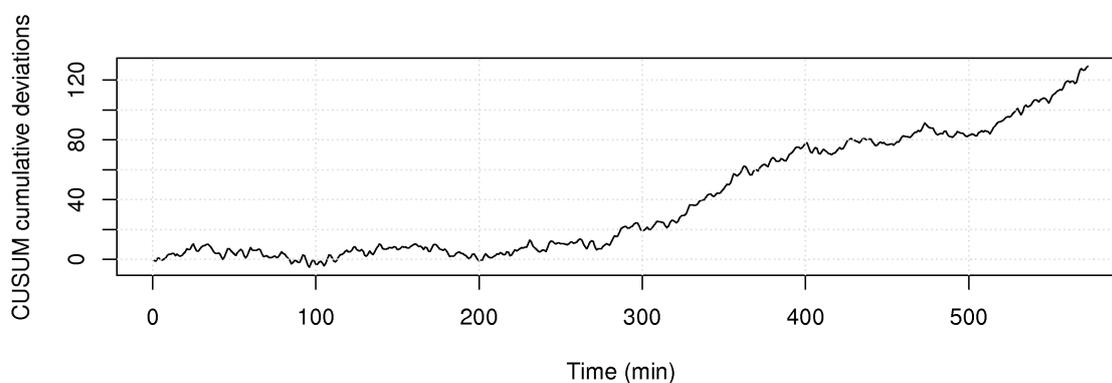
5. The revised CUSUM chart suggests that the error occurs around 275 min, as evidenced by the steep positive slope thereafter. It should be noted that the CUSUM chart begins to bear a positive slope around 200 min, but this initial increase in the cumulative error would likely not be diagnosable (i.e. using a V-mask).

```
# Code by Ryan and Stuart (2011 class)

CUSUM <- function(x, target){
    N <- length(x)
    S <- numeric(N)
    S[1] = x[1] - target
    for (t in 2:N){
        S[t] = S[t-1] + (x[t] - target)
    }
```

(continues on next page)

Chapter 3. Process Monitoring

```
    return(S)
}

# Import data and remove missing values (NA)
aeration.data <- read.csv('http://openmv.net/file/aeration-rate.csv')
aeration <- na.omit(aeration.data$Aeration)

# Plot raw data
bitmap('aeration-rate-raw-data.png', type="png256",
       width=10, height=4, res=300, pointsize=14)
plot(aeration, type="l", xlab="Time (min)", ylab="Aeration rate (L/min)")
grid()
dev.off()

# Plot CUSUM Chart
target <- median(aeration[1:200])
bitmap('aeration-CUSUM.png', type="png256",
       width=10, height=4, res=300, pointsize=14)
plot(CUSUM(aeration, target), type="l", xlab="Time (min)",
     ylab="CUSUM cumulative deviations")
grid()
dev.off()

# Plot the Shewhart chart: see code from the other question to
# calculate the control limits
LCL <- 22.1
UCL <- 25.8
N <- 5
subgroups <- matrix(aeration, N, length(aeration)/N)
x.mean <- numeric(length(aeration)/N)
x.sd <- numeric(length(aeration)/N)

# Calculate mean and sd of subgroups (see R-tutorial)
x.mean <- apply(subgroups, 2, mean)
x.sd <- apply(subgroups, 2, sd)
ylim <- range(x.mean) + c(-5, +5)
xdb <- target   # use the same CUSUM target !

bitmap('aeration-Shewhart-chart.png',
       type="png256", width=10, height=4, res=300, pointsize=14)
par(mar=c(4.2, 4.2, 0.5, 0.5))
par(cex.lab=1.3, cex.main=1.5, cex.sub=1.5, cex.axis=1.5)
plot(seq(1,length(x.mean)*N, N), x.mean, type="b", pch=".", cex=5, main="",
     ylab="Phase II subgroups", xlab="Time order", ylim=ylim)
abline(h=UCL, col="red")
abline(h=LCL, col="red")
abline(h=xdb, col="green")
lines(c(275, 275), ylim, col="blue")
text(280, 29, "CUSUM detected problem at t=275",adj = c(0,0))
dev.off()
```

6. Using the iterative Shewhart code from the previous question, we used

   - Phase I was taken far enough away from the suspected error: 0 - 200 min

   - Subgroup size of $n = 5$

   - $\bar{\bar{x}} = 23.9$

   - $\bar{S} = 1.28$

   - $a_n = 0.940$

   - LCL $= 23.9 - 3 \cdot \frac{1.28}{0.940\sqrt{5}} = 22.1$

   - UCL $= 23.9 + 3 \cdot \frac{1.28}{0.940\sqrt{5}} = 25.8$

The Shewhart chart applied to the entire dataset is shown below. In contrast to the CUSUM chart, the Shewhart chart is unable to detect the problem in the aeration rate. Unlike the CUSUM chart, which has infinite memory, the Shewhart chart has no memory and cannot adequately assess the location of the monitored variable in relation to its specified target. Instead, the Shewhart chart merely monitors aeration rate with respect to the control limits for the process. Since the aeration rate does not exceed the control limits for the process (i.e. process remains in control), the Shewhart chart does not detect any abnormalities.
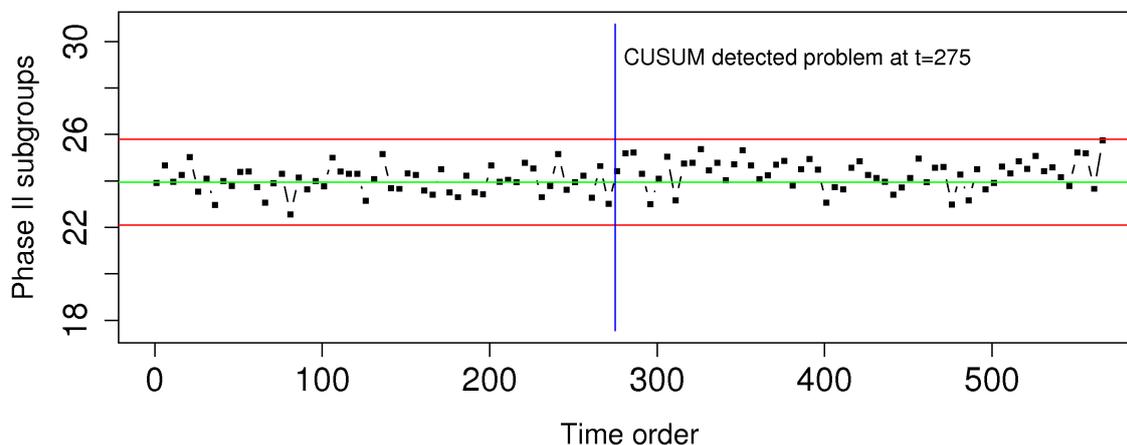
If you used the Western Electric rules, in addition to the Shewhart chart limits, you would have picked up a consecutive sequence of 8 points on one side of the target around $t = 350$.

Question 7

Do you think a Shewhart chart would be suitable for monitoring the closing price of a stock on the stock market? Please explain your answer if you agree, or describe an alternative if you disagree.

Solution

No, a Shewhart chart is not suitable for monitoring stock prices. Stock prices are volatile variables (not stable), so there is no sense in monitoring their location. Hopefully the stock is moving up, which it should on average, but the point is that stock prices are not stable. Nor are stock prices independent day-to-day.

So what aspect of a stock price is stable? The difference between the opening and closing price of a stock is remarkably stationary. Monitoring the day-to-day change in a stock price would work. Since you aren't expected to know this fact, any reasonable answer that attempts to monitor a *stable* substitute for the price will be accepted. E.g. another alternative is to remove the linear up or down trend from a stock price and monitor the residuals.

There are many alternatives; if this sort of thing interests you, you might find the area called technical analysis[65] worth investigating. An EWMA chart is widely used in this sort of analysis.

---

[65] https://en.wikipedia.org/wiki/Technical_analysis

Question 8

Describe how a monitoring chart could be used to prevent over-control of a batch-to-batch process. (A batch-to-batch process is one where a batch of materials is processed, followed by another batch, and so on).

Solution

Over-control of any process takes place when too much corrective action is applied. Using the language of feedback control, your gain is the right sign, but the magnitude is too large. Batch processes are often subject to this phenomenon: e.g. the operator reduces the set-point temperature for the next batch, because the current batch produced product with a viscosity that was too high. But then the next batch has a viscosity that is too low, so the operator increases the temperature set-point for the following batch. This constant switching is known as over-control (the operator is the feedback controller and his/her gain is too high, i.e. they are over-reacting).

A monitoring chart such as a Shewhart chart would help the operator: if the previous batch was within the limits, then s/he should not take any corrective action. Only take action when the viscosity value is outside the limits. An EWMA chart would additionally provide a one-step ahead prediction, which is an advantage.

Question 9

You need to construct a Shewhart chart. You go to your company's database and extract data from 10 periods of time lasting 6 hours each. Each time period is taken approximately 1 month apart so that you get a representative data set that covers roughly 1 year of process operation. You choose these time periods so that you are confident each one was from in control operation. Putting these 10 periods of data together, you get one long vector that now represents your phase 1 data.

- There are 8900 samples of data in this phase 1 data vector.

- You form subgroups: there are 4 samples per subgroup and 2225 subgroups.

- You calculate the mean within each subgroup (i.e. 2225 means). The mean of those 2225 means is 714.

- The standard deviation within each subgroup is calculated; the mean of those 2225 standard deviations is 98.

1. Give an unbiased estimate of the process standard deviation?

2. Calculate lower and upper control limits for operation at ± 3 of these standard deviations from target. These are called the action limits.

3. Operators like warning limits on their charts, so they don't have to wait until an action limit alarm occurs. Discussions with the operators indicate that lines at 590 and 820 might be good warning limits. What percentage of in control operation will lie inside the proposed warning limit region?

Short answer: Unbiased estimate of the process standard deviation = 106.4; UCL = 874; LCL = 554.

Question 10

If an exponentially weighted moving average (EWMA) chart can be made to approximate either a CUSUM or a Shewhart chart by adjusting the value of λ, what is an advantage of the EWMA chart over the other two? Describe a specific situation where you can benefit from this.

Question 11

The most recent estimate of the process capability ratio for a key quality variable was 1.30, and the average quality value was 64.0. Your process operates closer to the lower specification limit of 56.0. The upper specification limit is 93.0.

What are the two parameters of the system you could adjust, and by how much, to achieve a capability ratio of 1.67, required by recent safety regulations. Assume you can adjust these parameters independently.

Question 12

A bagging system fills bags with a target weight of 37.4 grams and the lower specification limit is 35.0 grams. Assume the bagging system fills the bags with a standard deviation of 0.8 grams:

1. What is the current Cpk of the process?

2. To what target weight would you have to set the bagging system to obtain Cpk=1.3?

3. How can you adjust the Cpk to 1.3 without adjusting the target weight (i.e. keep the target weight at 37.4 grams)?

Short answer: Current Cpk = 1.0

Question 13

Plastic sheets are manufactured on your blown film line. The Cp value is 1.7. You sell the plastic sheets to your customers with specification of 2 mm $\pm$ 0.4 mm.

1. List three important assumptions you must make to interpret the Cp value.

2. What is the theoretical process standard deviation, σ?

3. What would be the Shewhart chart limits for this system using subgroups of size $n = 4$?

4. Illustrate your answer from part 2 and 3 of this question on a diagram of the normal distribution.

Question 14

The following charts show the weight of feed entering your reactor. The variation in product quality leaving the reactor was unacceptably high during this period of time.

Raw data: weight of feed entering

Time−ordered samples

Shewhart chart: weight of feed entering

Time order
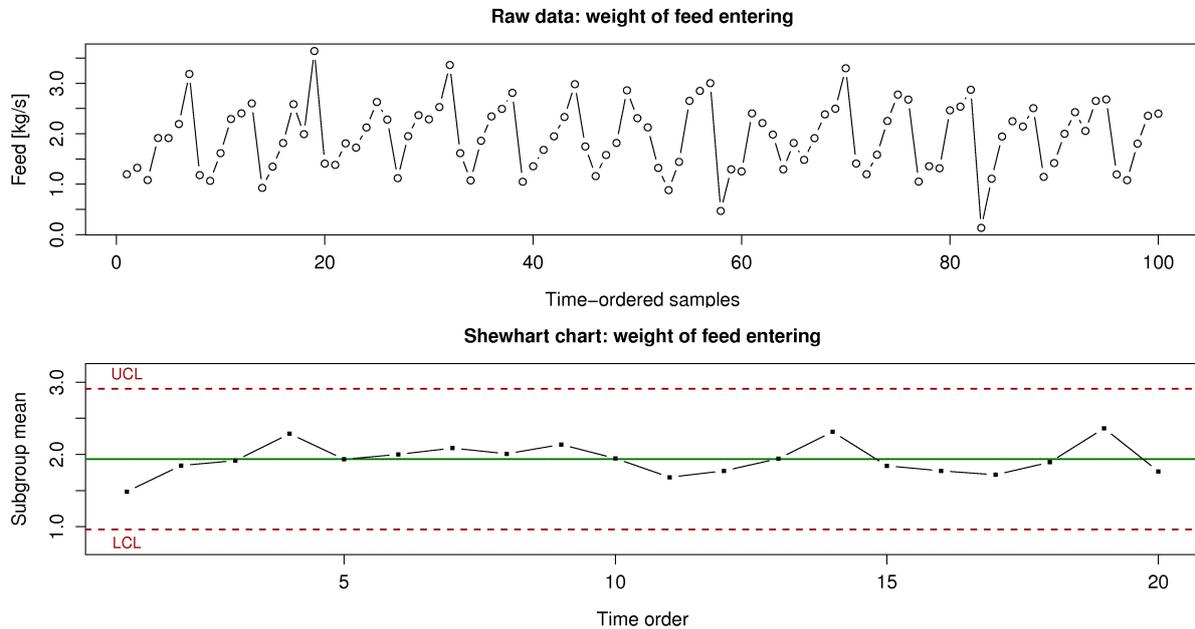
1. What can your group of process engineers learn about the problem, using the time-series plot (100 consecutive measurements, taken 1 minute apart).

2. Why is this variability not seen in the Shewhart chart?

3. Using concepts described elsewhere in this book, why might this sort of input to the reactor have an effect on the quality of the product leaving the reactor?

Question 15

You will come across these terms in the workplace. Investigate one of these topics, using the Wikipedia link below to kick-start your research. Write a paragraph that (a) describes what your topic is and (b) how it can be used when you start working in a company after you graduate, or how you can use it now if you are currently working.

- Lean manufacturing[66]

- Six sigma[67] and the DMAIC cycle. See the list of companies[68] that use six sigma tools.

- Kaizen[69] (a component of The Toyota Way[70])

- Genchi Genbutsu[71] (also a component of The Toyota Way[72])

---

[66] https://en.wikipedia.org/wiki/Lean_manufacturing
[67] https://en.wikipedia.org/wiki/Six_Sigma
[68] https://en.wikipedia.org/wiki/List_of_Six_Sigma_companies
[69] https://en.wikipedia.org/wiki/Kaizen
[70] https://en.wikipedia.org/wiki/The_Toyota_Way
[71] https://en.wikipedia.org/wiki/Genchi_Genbutsu
[72] https://en.wikipedia.org/wiki/The_Toyota_Way

In early 2010 Toyota experienced some of its worst press coverage on this very topic. Here is an article[73] in case you missed it.

Question 16

The Kappa number is a widely used measurement in the pulp and paper industry. It can be measured on-line, and indicates the severity of chemical treatment that must be applied to a wood pulp to obtain a given level of whiteness (i.e. the pulp's bleachability). Data on the website[74] contain the Kappa values from a pulp mill. Use the first 2000 data points to construct a Shewhart monitoring chart for the Kappa number. You may use any subgroup size you like. Then use the remaining data as your phase 2 (testing) data. Does the chart perform as expected?

Short answer: The intention of this question is for you to experience the process of iteratively calculating limits from phase 1 data and applying them to phase 2 data.

Question 17

In this section we showed how one can monitor any variable in a process. Modern instrumentation though capture a wider variety of data. It is common to measure point values, e.g. temperature, pressure, concentration and other hard-to-measure values. But it is increasingly common to measure spectral data. These spectral data are a vector of numbers instead of a single number.

Below is an example from a pharmaceutical process: a complete spectrum can be acquired many times per minute, and it gives a complete chemical fingerprint or signature of the system. There are 460 spectra in figure below; they could have come, for example, from a process where they are measured 5 seconds apart. It is common to find fibre optic probes embedded into pipelines and reactors to monitor the progress of a reaction or mixing.

Write a few bullet points how you might monitor a process where a spectrum (a vector) is your data source, and not a "traditional" single point measurement, like a temperature value.

[73] https://www.reuters.com/article/us-toyota-us-manufacturers-analysis/toyota-stumbles-but-its-kaizen-cult-endures-idUSTRE6161RV20100207
[74] http://openmv.net/info/kappa-number
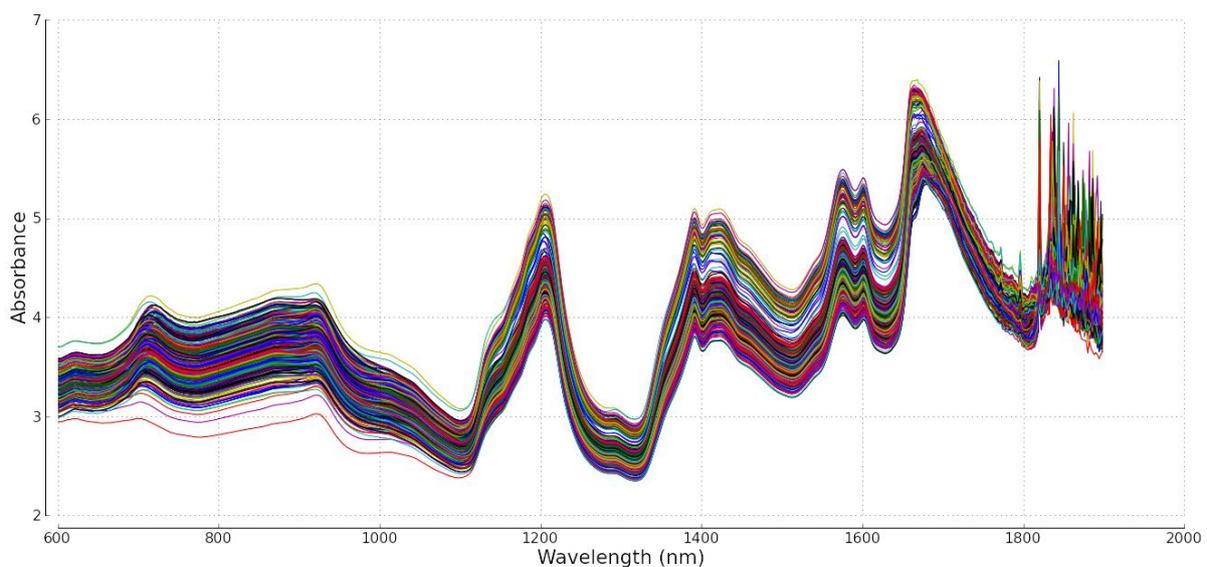
Question 18

The carbon dioxide measurement is available from a gas-fired furnace[75]. These data are from phase 1 operation.

1. Calculate the Shewhart chart upper and lower control limits that you would use during phase 2 with a subgroup size of $n = 6$.

2. Is this a useful monitoring chart? What is going in this data?

3. How can you fix the problem?

Solution

Solution based on work by Ryan and Stuart (2011 class)

First a plot of the raw data will be useful:
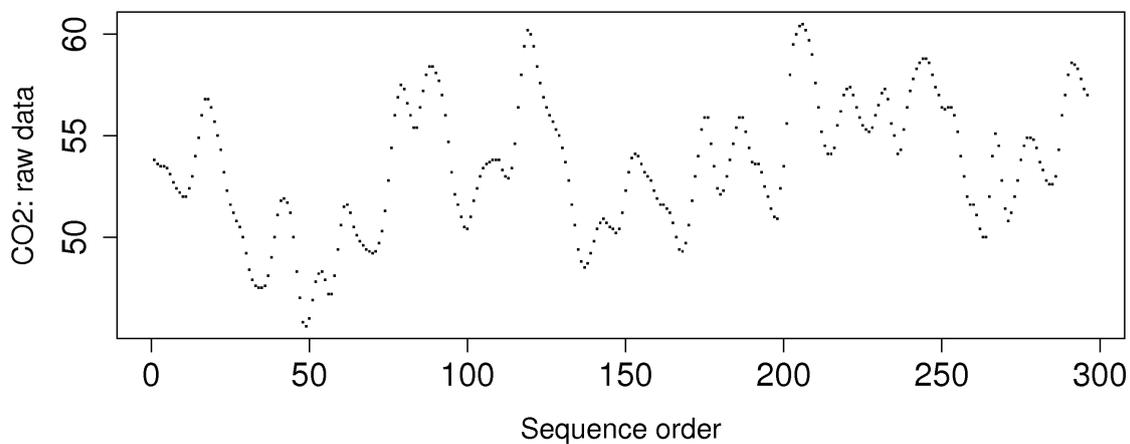
1. Assuming that the CO_2 data set is from phase 1 operation, the control limits were calculated as follows:

    - Assume subgroups are independent

    - $\bar{\bar{x}} = \frac{1}{K} \sum\limits_{k=1}^{K} \bar{x}_k = 53.5$

    - $\bar{S} = \frac{1}{K} \sum\limits_{k=1}^{K} s_k = 1.10$

    - $a_n = 0.952$

    - LCL $= 53.5 - 3 \cdot \frac{1.10}{0.952\sqrt{6}} = 52.08$

    - UCL $= 53.5 + 3 \cdot \frac{1.10}{0.952\sqrt{6}} = 54.92$

2. The Shewhart chart using a subgroup of size 6 is not a useful monitoring chart. There are too many false alarms, which will cause the operators to just ignore the chart. The problem is that the first assumption of independence is not correct and has a detrimental effect, as shown in *a previous question* (page 102).

---

[75] http://openmv.net/info/gas-furnace
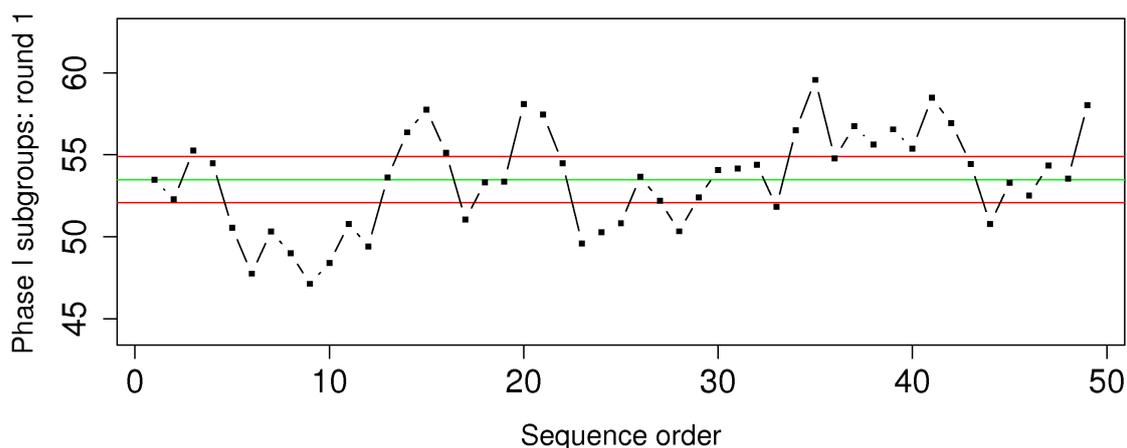
3. One approach to fixing the problem is to subsample the data, i.e. only use every k^{th} data point as the raw data, e.g. $k = 10$, and then form subgroups from that sampled data.

   Another is to use a larger subgroup size. Use the autocorrelation function[76], and the corresponding `acf(...)` function in R to verify the degree of relationship. Using this function we can see the raw data are unrelated after the 17th lag, so we could use subgroups of that size. However, even then we see the Shewhart chart showing frequent violation, though fewer than before.

   Yet another alternative is to use an EWMA chart, which takes the autocorrelation into account. However, the EWMA chart limits are found from the assumption that the subgroup means (or raw data, if subgroup size is 1), are independent.

   So we are finally left with the conclusion that perhaps there data really are not from in control operation, or, if they are, we must manually adjust the limits to be wider.

```
                                             R code
file <- 'http://openmv.net/file/gas-furnace.csv'
data <- read.csv(file)
CO2 <- data$CO2
N.raw <- length(CO2)
N.sub <- 6

# Change ``N.sub`` to 10, 15, 20, etc
# At N.sub <- 17 we see the
# autocorrelation disappear

# Plot all the data
par(mar=c(4.2, 4.2, 0.5, 0.5))
par(cex.lab=1.3, cex.main=1.5,
    cex.sub=1.5, cex.axis=1.5)
plot(CO2, type="p", pch=".", cex=2,
     main="", ylab="CO2: raw data",
     xlab="Sequence order")

# Create the subgroups on ALL the raw data.
# Form a matrix with `N.subgroup` rows by
# placing the vector of data down each row,
# then going across to form the columns.
# Calculate the mean and standard deviation
# within each subgroup (columns of the matrix)

subgroups <- matrix(CO2, N.sub, N.raw/N.sub)
subgroups.S <- apply(subgroups, 2, sd)
subgroups.xbar <- apply(subgroups, 2, mean)
ylim <- range(subgroups.xbar) + c(-3, +3)
```

(continues on next page)

---

[76] https://en.wikipedia.org/wiki/Autocorrelation

Chapter 3. Process Monitoring

```
# Keep adjusting N.sub until you don't see
# any autocorrelation between subgroups
acf(subgroups.xbar)

# Create a function to calculate
# Shewhart chart limits
shewhart_limits <- function(xbar, S,
                            sub.n, N.stdev=3){
  # Give the xbar and S vector containing
  # the subgroup means and standard
  # deviations.  Also give the subgroup
  # size used. Returns the lower and upper
  # control limits for the Shewhart chart
  # (UCL and LCL) which are N.stdev away
  # from the target.

  # xdb = x.double.bar = mean of means
  xdb <- mean(xbar)
  s.bar <- mean(S)
  num.an <- sqrt(2)*gamma(sub.n/2)
  den.an <- sqrt(sub.n-1)*gamma((sub.n-1)/2)
  an <- num.an / den.an
  LCL <- xdb - 3*s.bar/(an*sqrt(sub.n))
  UCL <- xdb + 3*s.bar/(an*sqrt(sub.n))
  return(list(LCL, xdb, UCL))
}

limits <- shewhart_limits(subgroups.xbar,
                          subgroups.S, N.sub)
LCL <- limits[1]
xdb <- limits[2]
UCL <- limits[3]
c(LCL, xdb, UCL)

# Any points outside these limits?
par(mar=c(4.2, 4.2, 0.5, 0.5))
par(cex.lab=1.3, cex.main=1.5,
    cex.sub=1.5, cex.axis=1.5)
plot(subgroups.xbar, type="b", pch=".",
     cex=5, main="", ylim=ylim,
     ylab="Phase I subgroups: round 1",
     xlab="Sequence order")
abline(h=UCL, col="red")
abline(h=LCL, col="red")
abline(h=xdb, col="green")
lines(subgroups.xbar, type="b", pch=".",
      cex=5)
```

Question 19

The percentage yield from a batch reactor, and the purity of the feedstock are available as the Batch yield and purity[77] data set. Assume these data are from phase 1 operation and calculate the Shewhart chart upper and lower control limits that you would use during phase 2. Use a subgroup size of $n = 3$.

1. What is phase 1?

2. What is phase 2?

3. Show your calculations for the upper and lower control limits for the Shewhart chart on the *yield value*.

---

[77] http://openmv.net/info/batch-yield-and-purity

4. Show a plot of the Shewhart chart on these phase 1 data.

Solution

Solution based on work by Ryan McBride, Stuart Young, and Mudassir Rashid (2011 class)

1. Phase 1 is the period from which historical data is taken that is known to be "in control". From this data, upper and lower control limits can be established for the monitored variable that contain a specified percent of all in control data.

2. Phase 2 is the period during which new, unseen data is collected by process monitoring in real-time. This data can be compared with the limits calculated from the "in control" data.

3. Assuming the dataset was derived from phase 1 operation, the batch yield data was grouped into subgroups of size 3. However, since the total number of data points (N=241) is not a multiple of three, the data set was truncated to the closest multiple of 3, i.e. $N_{new} = 240$, by removing the last data point. Subsequently, the mean and standard deviation were calculated for each of the 80 subgroups. From this data, the lower and upper control limits were calculated as follows:

$$\overline{\overline{x}} = \frac{1}{80} \sum_{k=1}^{80} \overline{x}_k = \mathbf{75.3}$$

$$\overline{S} = \frac{1}{80} \sum_{k=1}^{80} s_k = \mathbf{5.32}$$

$$\text{LCL} = \overline{\overline{x}} - 3 \cdot \frac{\overline{S}}{a_n \sqrt{n}} = \mathbf{64.9}$$

$$\text{UCL} = \overline{\overline{x}} + 3 \cdot \frac{\overline{S}}{a_n \sqrt{n}} = \mathbf{85.7}$$

$$\text{using } a_n = 0.886 \quad \text{ for a subgroup size of 3}$$

$$\text{and } \overline{\overline{x}} = 75.3$$

Noticing that the mean for subgroup 42, $\overline{x}_{42} = 63.3$, falls below this LCL, the control limits were recalculated excluding this subgroup from phase 1 data (see R-code). Following this adjustment, the new control limits were calculated to be:

- LCL = 65.0

- UCL = 85.8

4. Shewhart charts for both rounds of the yield data (before and after removing the outlier):
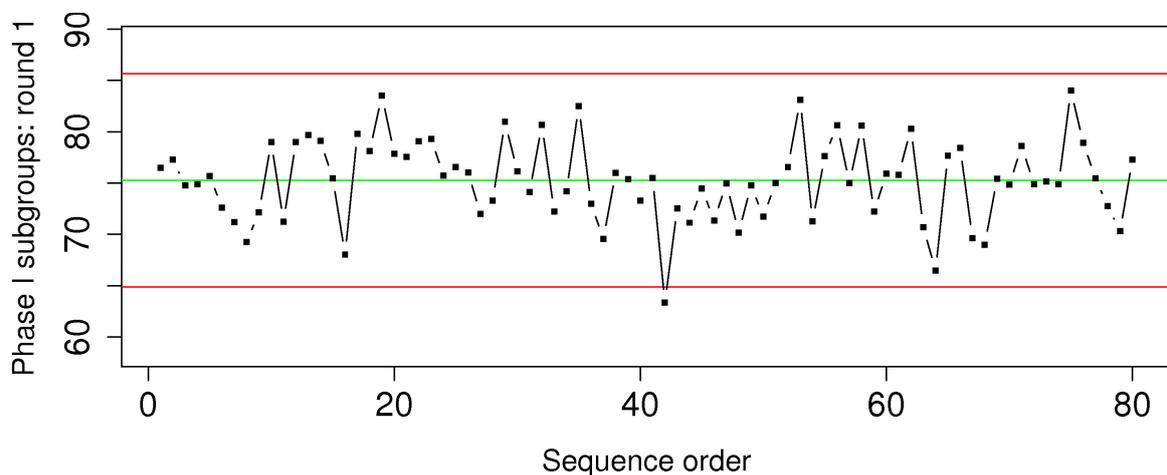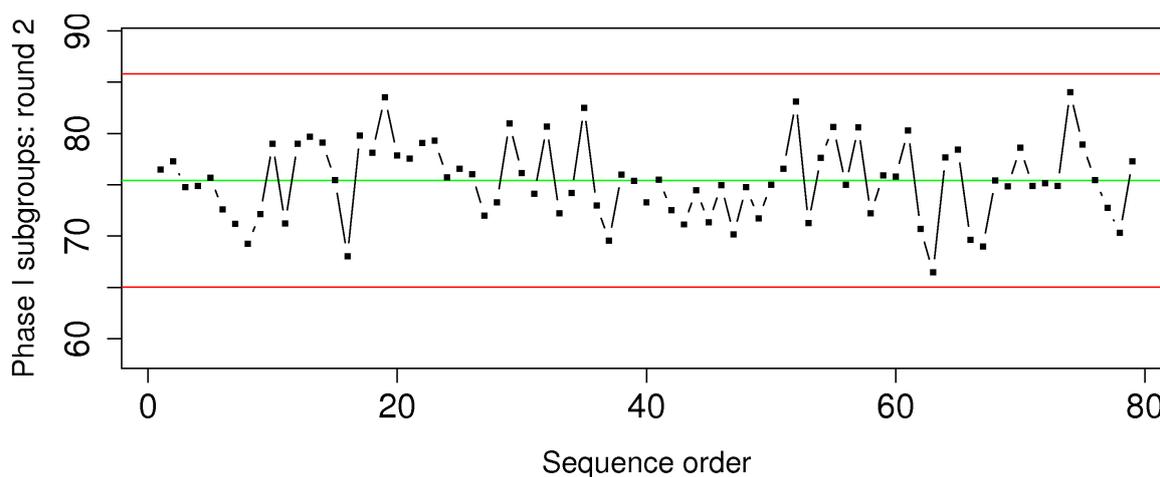
```
# Thanks to Mudassir for his source code to
# recursively calculate the limits. Some
# updates were made.

file <- 'http://openmv.net/file/batch-yield-and-purity.csv'
data <- read.csv(file)
y <- data$yield
variable <- "Yield"
N <- 3

# No further changes required. The code
# below will work for any new data set
subgroups <- matrix(y, N, length(y)/N)
x.mean <- numeric(length(y)/N)
x.sd <- numeric(length(y)/N)

# Calculate mean and sd of subgroups
# (see R-tutorial)
x.mean <- apply(subgroups, 2, mean)
x.sd <- apply(subgroups, 2, sd)
ylim <- range(x.mean) + c(-5, +5)
k <- 1
doloop <- TRUE

# Prevent infinite loops
while (doloop & k < 5){

  num.an <- sqrt(2)*gamma(N/2)
  den.an <- sqrt(N-1)*gamma((N-1)/2)
  an <- num.an / den.an

  S <- mean(x.sd)
  xdb <- mean(x.mean) # x-double bar
  LCL <- xdb - (3*S/(an*sqrt(N)))
  UCL <- xdb + (3*S/(an*sqrt(N)))
  print(c(LCL, UCL))

  # Create a figure on every loop
  par(mar=c(4.2, 4.2, 0.5, 0.5))
  par(cex.lab=1.3, cex.main=1.5,
      cex.sub=1.5, cex.axis=1.5)
  plot(x.mean, type="b", pch=".",
       cex=5, main="",
  ylab=paste("Phase I subgroups: round", k),
       xlab="Sequence order", ylim=ylim)
  abline(h=UCL, col="red")
  abline(h=LCL, col="red")
  abline(h=xdb, col="green")
  lines(x.mean, type="b", pch=".", cex=5)
```

(continues on next page)

```
if (!(any(x.mean < LCL) | any(x.mean > UCL))){
  # Finally!  No more points to exclude
  doloop <- FALSE
}
k <- k + 1

# Retain in x.sd and x.mean only those
# entries that are within the control
# limits
x.sd <- x.sd[x.mean>=LCL]
x.mean <- x.mean[x.mean>=LCL]
x.sd <- x.sd[x.mean<=UCL]
x.mean <- x.mean[x.mean<=UCL]
} # end: while doloop
```

Question 20

You will hear about 6-sigma processes frequently in your career. What does it mean exactly that a process is "6-sigma capable"? Draw a diagram to help illustrate your answer.

4.1 Least squares modelling in context

This section begins a new part: we start considering more than one variable at a time. However, you will see the tools of confidence intervals and visualization from the previous sections coming into play so that we can interpret our least squares models both analytically and visually.

The following sections, on design and analysis of experiments and latent variable models, will build on the least squares model we learn about here.

4.1.1 Usage examples

[Video for this section.](#)

The material in this section is used whenever you need to interpret and quantify the relationship between two or more variables.

- *Colleague*: How is the yield from our lactic acid batch fermentation related to the purity of the sucrose substrate?

  You: The yield can be predicted from sucrose purity with an error of plus/minus 8%

  Colleague: And how about the relationship between yield and glucose purity?

  You: Over the range of our historical data, there is no discernible relationship.

- *Engineer 1*: The theoretical equation for the melt index is non-linearly related to the viscosity

  Engineer 2: The linear model does not show any evidence of that, but the model's prediction ability does improve slightly when we use a non-linear transformation in the least squares model.

- *HR manager*: We use a least squares regression model to graduate personnel through our pay grades. The model is a function of education level and number of years of experience. What do the model coefficients mean?

4.1.2 What you will be able to do after this section
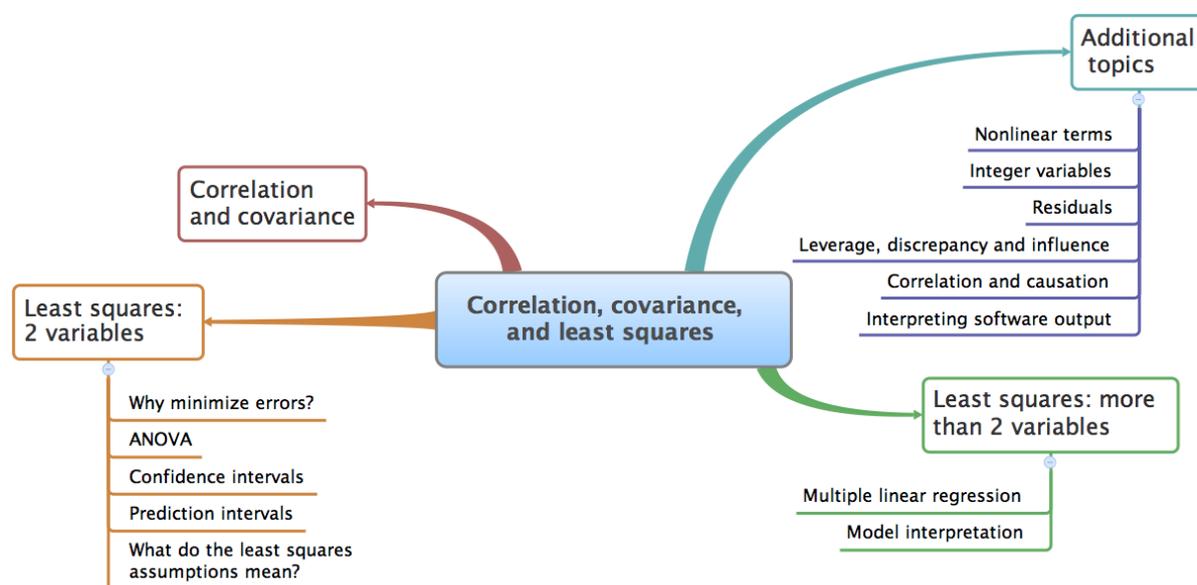
4.2 References and readings

This section is only a simple review of the least squares model. More details may be found in these references.

- **Recommended**: John Fox, *Applied Regression Analysis and Generalized Linear Models*, Sage.

- **Recommended**: N.R. Draper and H. Smith, *Applied Regression Analysis*, Wiley.

- Box, Hunter and Hunter, *Statistics for Experimenters*, selected portions of Chapter 10 (2nd edition), Wiley.

- Hogg and Ledolter, *Applied Statistics for Engineers and Physical Scientists*, Prentice Hall.

- Montgomery and Runger, *Applied Statistics and Probability for Engineers*, Wiley.

4.3 Covariance

Video for this section.

You probably have an intuitive sense for what it means when two things are correlated. We will get to correlation next, but we start by first looking at covariance. Let's take a look at an example to formalize this, and to see how we can learn from data.

Consider the measurements from a gas cylinder; temperature (K) and pressure (kPa). We know the ideal gas law applies under moderate conditions: $pV = nRT$.

- Fixed volume, $V = 20 \times 10^{-3} \text{m}^3 = 20$ L

- Moles of gas, $n = 14.1$ mols of chlorine gas, molar mass = 70.9 g/mol, so this is 1 kg of gas

- Gas constant, $R = 8.314$ J/(mol.K)

Given these numbers, we can simplify the ideal gas law to: $p = \beta_1 T$, where $\beta_1 = \dfrac{nR}{V} > 0$. These data are collected from sampling the system:

	T = Cylinder temperature (K)	*p* = Cylinder pressure (kPa)	*h* = Room humidity (%)
	273	1600	42
	285	1670	48
	297	1730	45
	309	1830	49
	321	1880	41
	333	1920	46
	345	2000	48
	357	2100	48
	369	2170	45
	381	2200	49
Mean	327	1910	46.1
Variance	1320	43267	8.1

The formal definition for covariance between any two variables is: [terminology used here was defined *in a previous section* (page 39)]

$$\text{Cov}\{x, y\} = \mathcal{E}\{(x - \overline{x})(y - \overline{y})\} \qquad \text{where} \qquad \mathcal{E}\{z\} = \overline{z} \tag{4.1}$$

Use this to calculate the covariance between temperature and pressure by breaking the problem into steps:

- First calculate deviation variables. They are called this because they are now the deviations from the mean: $T - \overline{T}$ and $p - \overline{p}$. Subtracting off the mean from each vector just centers their frame of reference to zero.

- Next multiply the two vectors, element-by-element, to calculate a new vector $(T - \overline{T})(p - \overline{p})$.

```R
                                    ─── R code ───
temp <- c(273, 285, 297, 309, 321, 333,
          345, 357, 369, 381)
pres <- c(1600, 1670, 1730, 1830, 1880,
          1920, 2000, 2100, 2170, 2200)
humidity <- c(42, 48, 45, 49, 41, 46,
              48, 48, 45, 49)

temp.centered <- temp - mean(temp)
pres.centered <- pres - mean(pres)
product <- temp.centered * pres.centered

# R does element-by-element  multiplication in the above line
print(product)
# [1] 16740 10080  5400  1440   180
#        60  1620  5700 10920 15660

# Average of 'product':
mean(product)    # 6780

# Calculated covariance is 7533.33
paste0('Covariance of temperature and ',
```

(continues on next page)

```
        'pressure is = ',
        round(cov(temp, pres), 2))

# The covariance of a variable with
# itself is just the variance:
paste0('Covariance with itself is = ',
       round(cov(temp, temp), 2))
paste0('while the variance = ',
       round(var(temp), 2))
```

- The expected value of this product can be estimated by using the average, or any other suitable measure of location. In this case `mean(product)` in R gives 6780. This is the covariance value.

- More specifically, we should provide the units as well: the covariance between temperature and pressure is 6780 [K.kPa] in this example. Similarly the covariance between temperature and humidity is 202 [K.%].

In your own time calculate a rough numeric value and give the units of covariance for these cases:

x	y
x = age of married partner 1	y = age of married partner 2
x = gas pressure	y = gas volume at a fixed temperature
x = mid term mark for this course	y = final exam mark
x = hours worked per week	y = weekly take home pay
x = cigarettes smoked per month	y = age at death
x = temperature on top tray of distillation column	y = top product purity

Also describe what an outlier observation would mean in these cases.

One last point is that the covariance of a variable with itself is the variance: $\text{Cov}\{x, x\} = \mathcal{V}(x) = \mathcal{E}\{(x - \overline{x})(x - \overline{x})\}$, a definition *we saw earlier* (page 39).

Using the `cov(temp, pres)` function in R gives `7533.333`, while we calculated 6780. The difference comes from $6780 \times \dfrac{N}{N-1} = 7533.33$, indicating that R divides by $N-1$ rather than N. This is because the variance function in R for a vector x is internally called as `cov(x, x)`. Since R returns the unbiased variance, it divides through by $N - 1$. This inconsistency does not really matter for large values of N, but emphasizes that one should always read the documentation for the software being used.

Note that deviation variables are not affected by a *shift* in the raw data of x or y. For example, measuring temperature in Celsius or Kelvin has no effect on the covariance number; but measuring it in Celsius vs Fahrenheit does change the covariance value.

4.4 Correlation

The variance and covariance values are units dependent. For example, you get a very different covariance when calculating it using grams vs kilograms. The correlation on the other hand removes the effect of scaling and arbitrary unit changes. It is defined as:

$$\text{Correlation} = r(x, y) = \frac{\mathcal{E}\left\{(x - \bar{x})(y - \bar{y})\right\}}{\sqrt{\mathcal{V}\{x\}\,\mathcal{V}\{y\}}} = \frac{\text{Cov}\{x, y\}}{\sqrt{\mathcal{V}\{x\}\,\mathcal{V}\{y\}}} \tag{4.2}$$

It takes the covariance value and divides through by the units of x and of y to obtain a dimensionless result. The values of $r(x, y)$ range from -1 to $+1$. Also note that $r(x, y) = r(y, x)$.

So returning back to our example of the gas cylinder, the correlation between temperature and pressure, and temperature and humidity can be calculated now as:

```R
# R code
temp <- c(273, 285, 297, 309, 321, 333, 345,
          357, 369, 381)
pres <- c(1600, 1670, 1730, 1830, 1880, 1920,
          2000, 2100, 2170, 2200)
humidity <- c(42, 48, 45, 49, 41, 46, 48,
              48, 45, 49)

# Correlation between temperature
# and pressure is high: 0.9968355
cor(temp, pres)

# Correlation between temperature
# and humidity is low: 0.3803919
cor(temp, humidity)

# What is correlation of humidity
# and pressure?
cor(___, ___)
```

Note that correlation is the same whether we measure temperature in Celsius or Kelvin. Study the plots here to get a feeling for the correlation value and its interpretation:
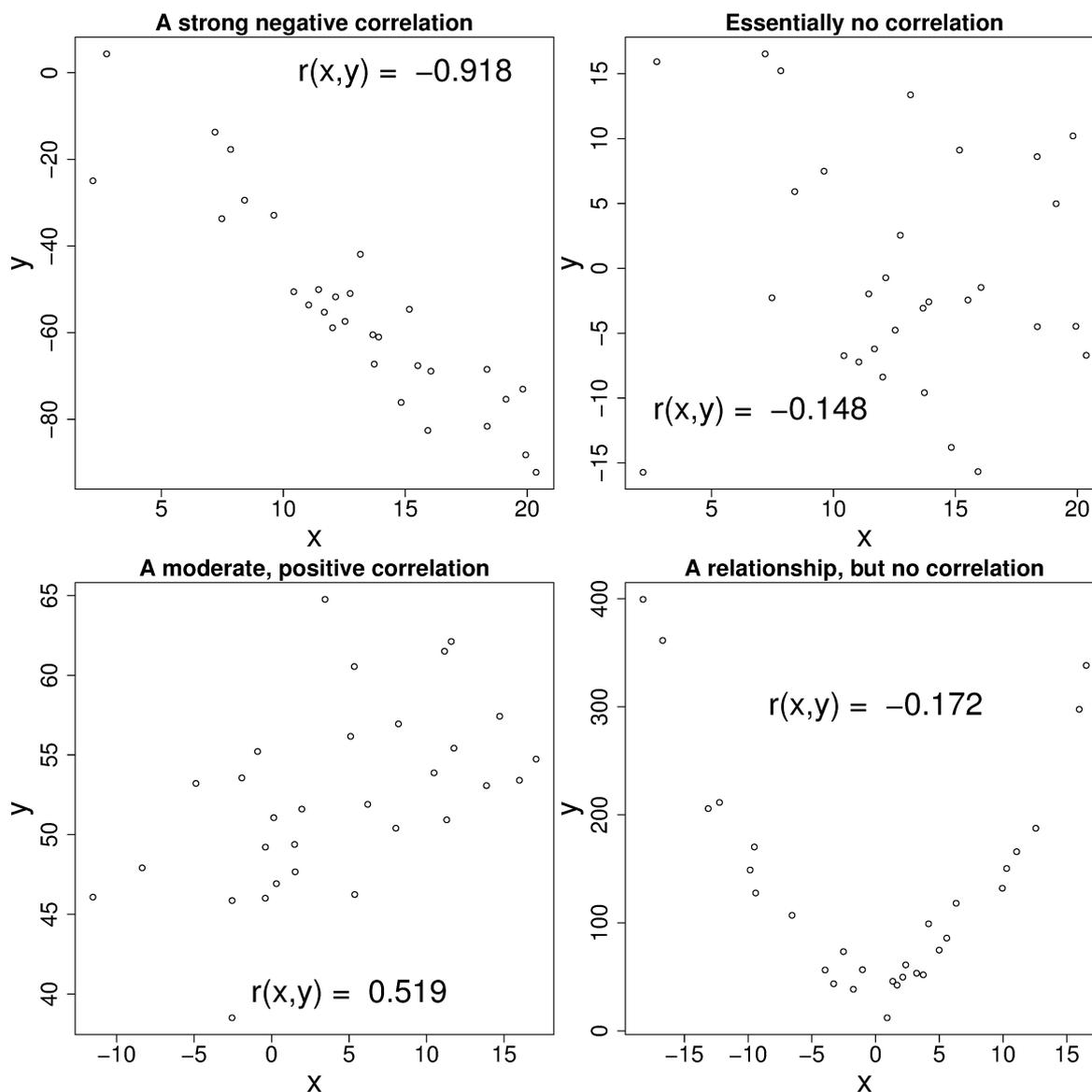
4.5 Some definitions

Be sure that you can derive (and interpret!) these relationships, which are derived from the definition of the covariance and correlation:

- $\mathcal{E}\{x\} = \overline{x}$

- $\mathcal{E}\{x + y\} = \mathcal{E}\{x\} + \mathcal{E}\{y\} = \overline{x} + \overline{y}$

- $\mathcal{V}\{x\} = \mathcal{E}\{(x - \overline{x})^2\}$

- $\mathcal{V}\{cx\} = c^2 \mathcal{V}\{x\}$

- $\mathrm{Cov}\{x, y\} = \mathcal{E}\{(x - \overline{x})(y - \overline{y})\}$ which we take as the definition for covariance

- $\mathcal{V}\{x + x\} = 2\mathcal{V}\{x\} + 2\mathrm{Cov}\{x, x\} = 4\mathcal{V}\{x\}$

- $\mathrm{Cov}\{x, y\} = \mathcal{E}\{xy\} - \mathcal{E}\{x\}\mathcal{E}\{y\}$

- $\mathrm{Cov}\{x, c\} = 0$

- $\text{Cov}\{x + a, y + b\} = \text{Cov}\{x, y\}$

- $\text{Cov}\{ax, by\} = ab \cdot \text{Cov}\{x, y\}$

- $\mathcal{V}\{x + y\} \neq \mathcal{V}\{x\} + \mathcal{V}\{y\}$, which is counter to what might be expected.

- Rather:

$$\begin{aligned}
\mathcal{V}\{x + y\} &= \mathcal{E}\{(x + y - \overline{x} - \overline{y})^2\} \\
&= \mathcal{E}\{((x - \overline{x}) + (y - \overline{y}))^2\} \\
&= \mathcal{E}\{(x - \overline{x})^2 + 2(x - \overline{x})(y - \overline{y}) + (y - \overline{y})^2\} \\
&= \mathcal{E}\{(x - \overline{x})^2\} + 2\mathcal{E}\{(x - \overline{x})(y - \overline{y})\} + \mathcal{E}\{(y - \overline{y})^2\} \\
&= \mathcal{V}\{x\} + 2\text{Cov}\{x, y\} + \mathcal{V}\{y\} \\
\mathcal{V}\{x + y\} &= \mathcal{V}\{x\} + \mathcal{V}\{y\}, \qquad \text{only if } x \text{ and } y \text{ are independent}
\end{aligned} \tag{4.3}$$

4.6 Least squares models with a single x-variable

The general linear least squares model is a very useful tool (in the right circumstances), and it is the workhorse for a number of algorithms in data analysis.

This part covers the relationship between two variables only: x and y. In a *later part on general least squares* (page 185) we will consider more than two variables and use matrix notation. But we start off slowly here, looking first at the details for relating two variables.

We will follow these steps:

1. Model definition (this subsection)

2. Building the model

3. Interpretation of the model parameters and model outputs (coefficients, R^2, and standard error S_E)

4. Consider the effect of unusual and influential data

5. Assessment of model residuals

The least squares model postulates that there is a linear relationship between measurements in vector x and vector y of the form:

$$\begin{aligned}
\mathcal{E}\{\mathrm{y}\} &= \beta_0 + \beta_1 \mathrm{x} \\
\mathrm{y} &= \beta_0 + \beta_1 \mathrm{x} + \epsilon
\end{aligned} \tag{4.4}$$

The β_0, β_1 and ϵ terms are *population* parameters, which are unknown (see the *section on univariate statistics* (page 38)). The ϵ term represents any unmodelled components of the linear model, measurement error, and is simply called *the error* term. Notice that the error is not due to x, but is the error in fitting y; we will return to this point in the section on *least squares assumptions* (page 166). Also, if there is no relationship between x and y then $\beta_1 = 0$.

We develop **the least squares method** to estimate these parameters; these estimates are defined as $b_0 = \hat{\beta}_0$, $b_1 = \hat{\beta}_1$ and $e = \hat{\epsilon}$. Using this new nomenclature we can write, for a given observation i:

$$\begin{aligned}
y_i &= b_0 + b_1 x_i + e_i \\
\hat{y}_i &= b_0 + b_1 x_i
\end{aligned} \tag{4.5}$$

$$\hat{y} = b_0 + b_1 x$$
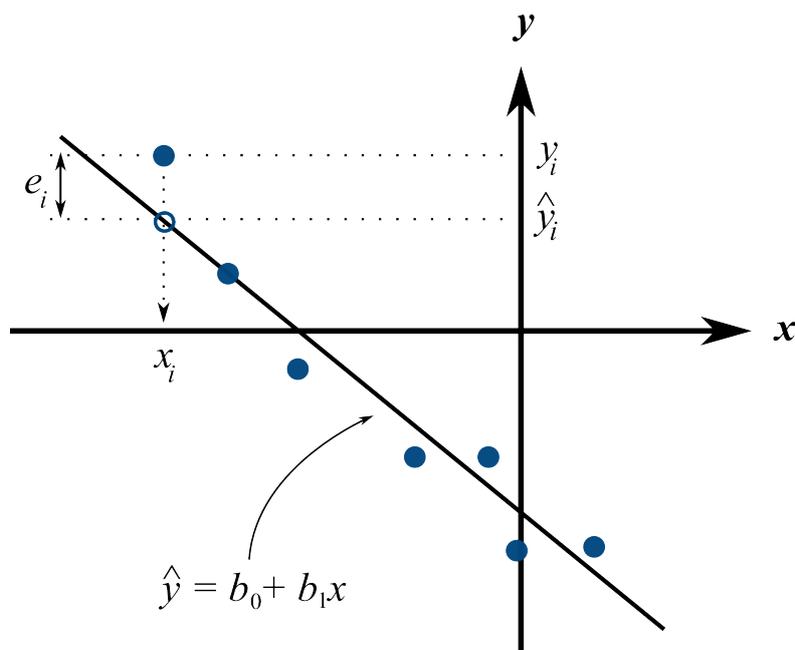
Presuming we have calculated estimates b_0 and b_1 we can use the model with a new x-observation, x_i, and predict its corresponding $\hat{y}_i$. The error value, e_i, is generally non-zero indicating out prediction estimate of $\hat{y}_i$ is not exact. All this new nomenclature is illustrated in the figure.

4.6.1 Minimizing errors as an objective

Our immediate aim however is to calculate the b_0 and b_1 estimates from the n pairs of data collected: (x_i, y_i).

Here are some valid approaches, usually called objective functions for making the e_i values small. One could use:

1. $\sum_{i=1}^{n} (e_i)^2$, which leads to the least squares model

2. $\sum_{i=1}^{n} (e_i)^4$

3. sum of perpendicular distances to the line $y = b_0 + b_1 x$

4. $\sum_{i=1}^{n} \|e_i\|$ is known as the least absolute deviations model, or the l-1 norm problem

5. *least median of squared error* model, which a robust form of least squares that is far less sensitive to outliers.

The traditional least squares model, the first objective function listed, has the lowest possible variance for b_0 and b_1 when certain additional *assumptions are met* (page 166). The low variance of these parameter estimates is very desirable, for both model interpretation and using the model. The other objective functions are good alternatives and may useful in many situations, particular the last alternative.

Other reasons for so much focus on the least squares alternative is because it is computationally tractable by hand and very fast on computers, and it is easy to prove various mathematical properties. The other forms take much longer to calculate, almost always have to be done on a computer, may have multiple solutions, the solutions can change dramatically given small deviations in the data (unstable, high variance solutions), and the mathematical proofs are difficult. Also the interpretation

of the least squares objective function is suitable in many situations: it penalizes deviations quadratically; i.e. large deviations much more than the smaller deviations.

You can read more about least squares alternatives in the book by Birkes and Dodge: "Alternative Methods of Regression".

4.6.2 Solving the least squares problem and interpreting the model

Having settled on the least squares objective function, let's construct the problem as an optimization problem and understand its characteristics.

The least squares problem can be posed as an unconstrained optimization problem:

$$
\begin{aligned}
\min_{b_0, b_1} f(b_0, b_1) &= \sum_{i=1}^{n} (e_i)^2 \\
&= \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2
\end{aligned}
\tag{4.6}
$$

Returning to our example of the gas cylinder. In this case we know that $\beta_0 = 0$ from theoretical principles. So we can solve the above problem by trial and error for b_1. We expect $b_1 \approx \beta_1 = \dfrac{nR}{V} = \dfrac{(14.1 \text{ mol})(8.314 \text{ J/(mol.K)})}{20 \times 10^{-3} \text{m}^3} = 5.861 \text{ kPa/K}$. So construct equally spaced points of $5.0 \le b_1 \le 6.5$, set $b_0 = 0$. Then calculate the objective function using the (x_i, y_i) data points recorded earlier using (4.6).
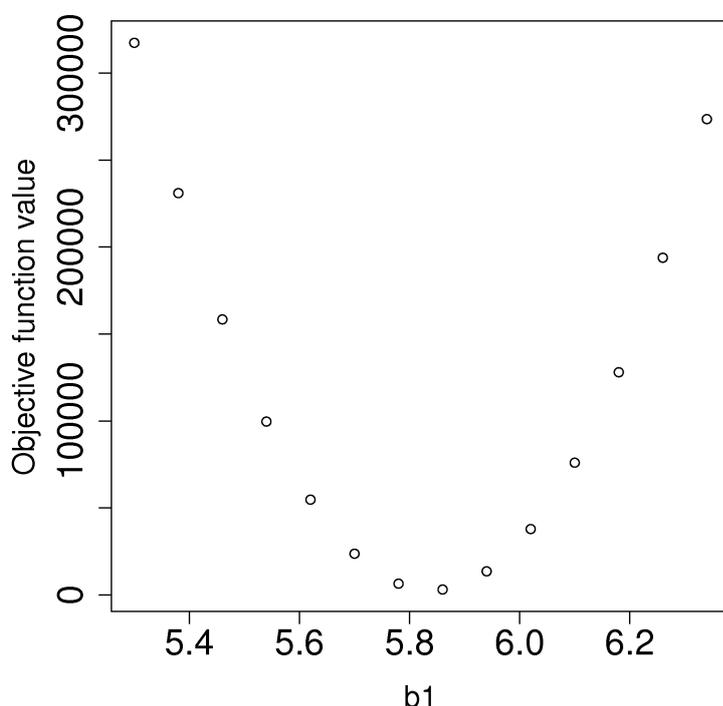
We find our best estimate for b_1 roughly at 5.88, the minimum of our grid search, which is very close to the theoretically expected value of 5.86 kPa/K.

For the case where we have both b_0 and b_1 varying we can construct a grid and tabulate the objective function values at all points on the grid. The least squares objective function will always be shaped like a bowl for these cases, and a unique minimum always be found, because the objective function is convex.
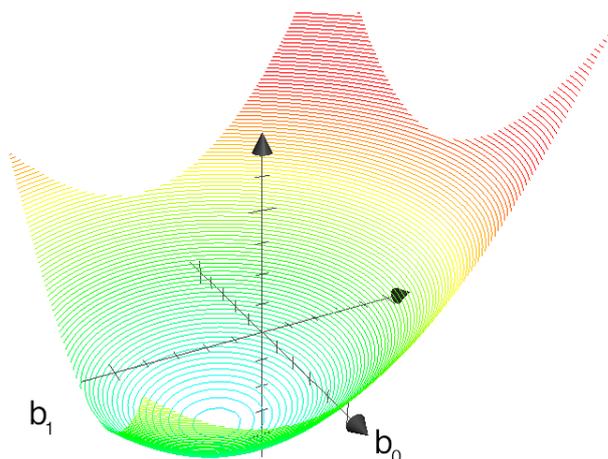
The above figure shows the general nature of the least-squares objective function where the two horizontal axes are for b_0 and b_1, while the vertical axis represents the least squares objective function $f(b_0, b_1)$.

The illustration highlights the quadratic nature of the objective function. To find the minimum analytically we start with equation (4.6) and take partial derivatives with respect to b_0 and b_1, and set those equations to zero. This is a required condition at any optimal point (see a reference on optimization theory), and leads to 2 equations in 2 unknowns.

$$
\begin{aligned}
\frac{\partial f(b_0, b_1)}{\partial b_0} &= -2 \sum_i^n (y_i - b_0 - b_1 x_i) = 0 \\
\frac{\partial f(b_0, b_1)}{\partial b_1} &= -2 \sum_i^n (x_i)(y_i - b_0 - b_1 x_i) = 0
\end{aligned}
\tag{4.7}
$$

Now divide the first line through by n (the number of data pairs we are using to estimate the parameters) and solve that equation for b_0. Then substitute that into the second line to solve for b_1. From this we obtain the parameters that provide the least squares optimum for $f(b_0, b_1)$:

$$
\begin{aligned}
b_0 &= \overline{y} - b_1 \overline{x} \\
b_1 &= \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}
\end{aligned}
\tag{4.8}
$$

Verify for yourself that:

1. The first part of equation (4.7) shows $\sum_i e_i = 0$, also implying the average error is zero.

2. The first part of equation (4.8) shows that the straight line equation passes through the mean of the data $(\overline{x}, \overline{y})$ without error.

3. From second part of equation (4.7) prove to yourself that $\sum_i (x_i e_i) = 0$, just another way of saying the dot product of the x-data and the error, $x^T e$, is zero.

4. Also prove and *interpret* that $\sum_i (\hat{y}_i e_i) = 0$, the dot product of the predictions and the errors is zero.

5. Notice that the parameter estimate for b_0 depends on the value of b_1: we say the estimates are correlated - you cannot estimate them independently.

6. You can also compute the second derivative of the objective function to confirm that the optimum is indeed a minimum.

Remarks:

1. What units does parameter estimate b_1 have?

   - The units of y divided by the units of x.

2. Recall the *temperature and pressure example* (page 152): let $\hat{p}_i = b_0 + b_1 T_i$:

   1. What is the interpretation of coefficient b_1?

      - A one Kelvin increase in temperature is associated, on average, with an increase of b_1 kPa in pressure.

   2. What is the interpretation of coefficient b_0?

      - It is the expected pressure when temperature is zero. Note: often the data used to build the model are not close to zero, so this interpretation may have no meaning.

3. What does it mean that $\sum_i (x_i e_i) = x^T e = 0$ (i.e. the dot product is zero):

   - The residuals are uncorrelated with the input variables, x. There is no information in the residuals that is in x.

4. What does it mean that $\sum_i (\hat{y}_i e_i) = \hat{y}^T e = 0$

   - The fitted values are uncorrelated with the residuals.

5. How could the denominator term for b_1 equal zero? And what would that mean?

   - This shows that as long as there is variation in the x-data that we will obtain a solution. We get no solution to the least squares objective if there is no variation in the data.

4.6.3 Example

We will refer back to the following example several times. Calculate the least squares estimates for the model $y = b_0 + b_1 x$ from the given data. Also calculate the predicted value of $\hat{y}_i$ when $x_i = 5.5$

x	10.0	8.0	13.0	9.0	11.0	14.0	6.0	4.0	12.0	7.0	5.0
y	8.04	6.95	7.58	8.81	8.33	9.96	7.24	4.26	10.84	4.82	5.68
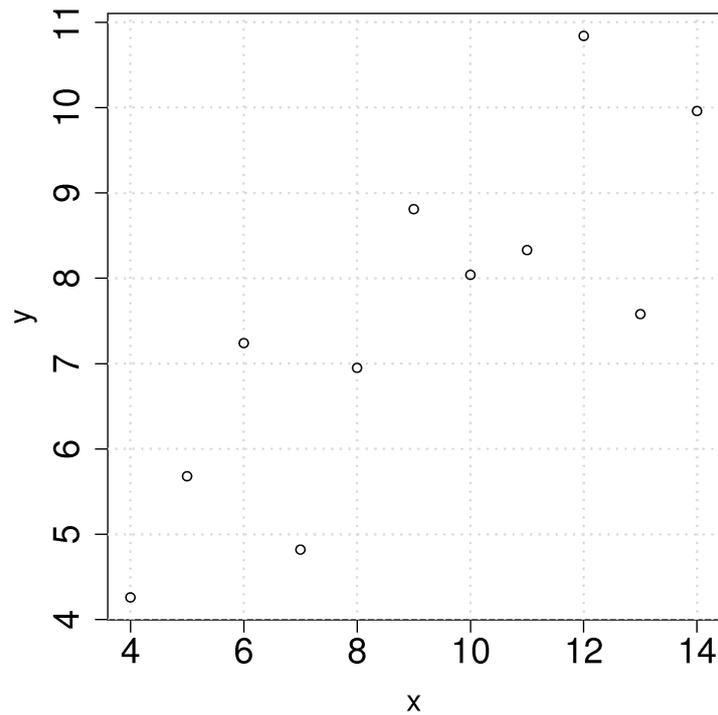
To calculate the least squares model in R:

```R
                                                        R code
x <- c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y <- c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96,
       7.24, 4.26, 10.84, 4.82, 5.68)

# "Calculate for me the linear model,
# where y is described by x"
mod.ls <- lm(y ~ x)

# Call:
# lm(formula = y ~ x)
#
# Coefficients:
# (Intercept)           x
#            3.0001      0.5001

# You can get more information with
summary(mod.ls)

print('The model coefficients are: ')
coefficients(mod.ls)

b0 <- coefficients(mod.ls)[1]
b1 <- coefficients(mod.ls)[2]
x.new <- 5.5
y_predicted <- b0 + b1 * x.new
paste0('Given a new x value of ', x.new,
       ' the predicted y = ',
       round(y_predicted, 3))
```

- $b_0 = 3.0$

- $b_1 = 0.5$

- When $x_i = 5$, then $\hat{y}_i = 3.0 + 0.5 \times 5.5 = 5.75$

4.7 Least squares model analysis

Once we have fitted the b_0 and b_1 terms using the data and the equations from *the prior section* (page 160), it is of interest to know how well the model performed. That is what this section is about. In particular:

1. Analysis of variance: breaking down the data's variability into components

2. Confidence intervals for the model coefficients, b_0 and b_1

3. Prediction error estimates for the y-variable

4. We will also take a look at the interpretation of the software output.

In order to perform the second part we need to make a few assumptions about the data, and if the data follow those assumptions, then we can derive confidence intervals for the model parameters.

4.7.1 The variance breakdown

Video for
this
section.

Recall that *variability* (page 30) is what makes our data interesting. Without variance (i.e. just flat lines) we would have nothing to do. The analysis of variance is just a tool to show how much variability in the y-variable is explained by:

1. Doing nothing (no model: this implies $\hat{y} = \overline{y}$)

2. The model ($\hat{y}_i = b_0 + b_1 x_i$)

3. How much variance is left over in the errors, e_i

These 3 components must add up to the total variance we started with. By definition, the variance is computed about a mean, so the variance of no model (i.e. the "doing nothing" case) is zero. So the total variance in vector y is just the sum of the other two variances: the model's variance, and the error variance. We show this next.

Using the accompanying figure, we see that geometrically, at any fixed value of x_i, that any y value above or below the least squares line, call it y_i and shown with a circle, must obey the distance relationship:

Distance relationship:	$(y_i - \overline{y})$	$=$ $(\hat{y}_i - \overline{y}) + (y_i - \hat{y}_i)$
Squaring both sides:	$(y_i - \overline{y})^2$	$=$ $(\hat{y}_i - \overline{y})^2 + 2(\hat{y}_i - \overline{y})(y_i - \hat{y}_i) + (y_i - \hat{y}_i)^2$
Sum and simplify:	$\sum (y_i - \overline{y})^2$	$=$ $\sum (\hat{y}_i - \overline{y})^2 + \sum (y_i - \hat{y}_i)^2$
	Total sum of squares (TSS)	$=$ Regression SS (RegSS) + Residual SS (RSS)
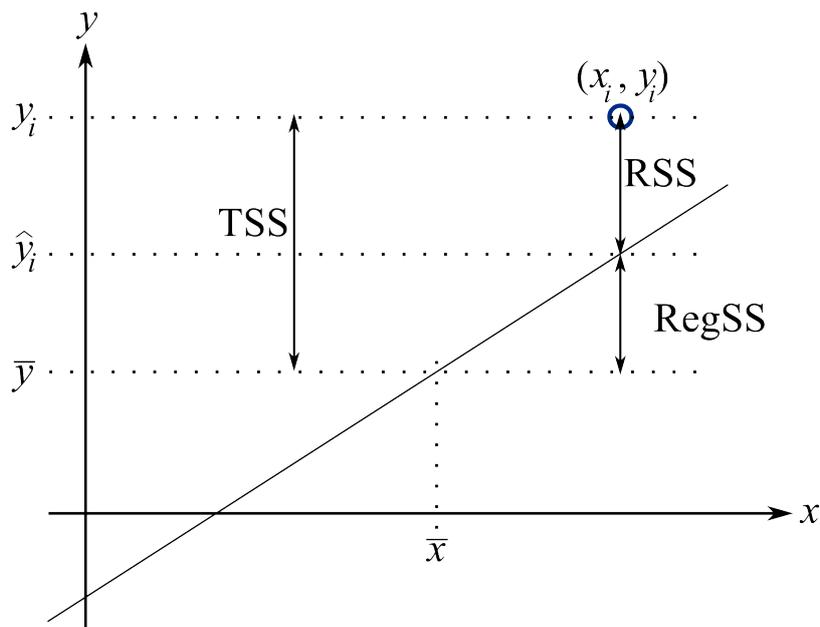
The total sum of squares (TSS) is the total variance in the vector of y-data. This broken down into two components: the sum of squares due to regression, $\sum (\hat{y}_i - \overline{y})^2$, called RegSS, and the sum of squares of the residuals (RSS), $\sum e_i^2 = e^T e$.

It is convenient to write these sums of squares (variances) in table form, called an Analysis of Variance (ANOVA) table:

Type of variance	Distance	Degrees of freedom	SSQ	Mean square
Regression	$\hat{y}_i - \overline{y}$	k ($k = 2$ in the examples so far)	RegSS	RegSS/k
Error	$y_i - \hat{y}_i$	$n - k$	RSS	RSS/$(n - k)$
Total	$y_i - \overline{y}$	n	TSS	TSS/n

Interpreting the standard error

The term $S_E^2 = \text{RSS}/(n - k)$ is one way of quantifying the model's performance. The value $S_E = \sqrt{\text{RSS}/(n - k)} = \sqrt{(e^T e)/(n - k)}$ is called the standard error. It is really just the standard deviation of the error term, accounting correctly for the degrees of freedom.

Example: Assume we have a model for predicting batch yield in kilograms from x = raw material purity, what does a standard error of 3.4 kg imply?

Answer: Recall if the assumption of normally distributed errors is correct, then this value of 3.4 kg indicates that about two thirds of the yield predictions will lie within ± 3.4 kg, and that 95% of the yield predictions will lie within $\pm 2 \times 3.4$ kg. We will quantify the prediction interval more precisely, but the standard error is a good approximation for the error of y.

Exercise

For two extreme cases:

1. $y_i = e_i$, i.e. where $b_0 = 0$ and $b_1 = 0$. In other words, our y_i measurements are just random noise.

2. $y_i = b_0 + b_1 x_i + e_i$, for any values of b_0 and b_1, that model fits the data perfectly, with no residuals.

Do the following in the space below:

- draw a generic plot

- create an ANOVA table with fake values

- write down the value of the ratio $\dfrac{\text{RegSS}}{\text{TSS}}$

- interpret what this ratio means: $F_0 = \dfrac{\text{mean square of regression}}{\text{mean square of residuals}}$

From this exercise we learn that:

- The null model ($y_i = e_i$) has ratio $\dfrac{\text{RegSS}}{\text{TSS}} = 0$.

- Models where the fit is perfect have a ratio $\dfrac{\text{RegSS}}{\text{TSS}} = 1$. This number is called R^2, and we will see why it is called that next.

Derivation of R^2

As introduced by example in the previous part, $R^2 = \dfrac{\text{RegSS}}{\text{TSS}} = \dfrac{\sum_i \left(\hat{y}_i - \overline{y} \right)^2}{\sum_i \left(y_i - \overline{y} \right)^2}$: simply the ratio between the variance we can explain with the model (RegSS) and the total variance we started off with (TSS). We can also write that $R^2 = 1 - \dfrac{\text{RSS}}{\text{TSS}}$, based on the fact that TSS = RegSS + RSS.

From the above ratios it is straightforward to see that if $R^2 = 0$, it requires that $\hat{y}_i = \overline{y}$: we are predicting just a flat line, the mean of the y data. On the other extreme, an $R^2 = 1$ implies that $\hat{y}_i = y_i$, we have perfect predictions for every data point.

The nomenclature R^2 comes from the fact that it is the square of the correlation between x and y. Recall from the *correlation section* (page 155) that

$$r(x, y) = \frac{\mathcal{E}\left\{ (x - \overline{x})(y - \overline{y}) \right\}}{\sqrt{\mathcal{V}\left\{ x \right\} \mathcal{V}\left\{ y \right\}}} = \frac{\text{Cov}\left\{ x, y \right\}}{\sqrt{\mathcal{V}\left\{ x \right\} \mathcal{V}\left\{ y \right\}}}$$

and can range in value from -1 to $+1$. The R^2 ranges from 0 to $+1$, and is the square of $r(x, y)$. R^2 is just a way to tell how far we are between predicting a flat line (no variation) and the extreme of being able to predict the model building data, y_i, exactly.

The R^2 value is likely well known to anyone that has encountered least squares before. This number must be interpreted with caution. It is most widely **abused** as a way to measure *"how good is my model"*.

These two common examples illustrate the abuse. You likely have said or heard something like this before:

1. "the R^2 value is really high, 90%, so this is a good model".

2. "Wow, that's a really low R^2, this model can't be right - it's no good".

How **good**, or how suitable a model is *for a particular purpose* is almost never related to the R^2 value. The goodness of a model is better assessed by:

- your engineering judgment: does the *interpretation* of model parameters make sense?

- use testing data to verify the model's predictive performance,

- using cross-validation tools (we will see this topic later on) to see how well the model performs on new, unseen and unused testing data.

We will see later on that R^2 can be arbitrarily increased by adding terms to the linear model, as we will see in the section on *multiple linear regression (MLR)* (page 185). So sometimes you will see the adjusted R^2 used to account for the k terms used in the model:

$$R^2_{\text{adj}} = 1 - \frac{\text{RSS}/(n-k)}{\text{TSS}/(n-1)}$$

where $k = 2$ for the case of estimating a model $y_i = b_0 + b_1 x_i$, as there are 2 parameters.

4.7.2 Confidence intervals for the model coefficients b_0 and b_1

Note: A good reference for this section is the book by Fox (Chapter 6), and the book by Draper and Smith.

Up to this point we have made no assumptions about the data. In fact we can calculate the model estimates, b_0 and b_1 as well as predictions from the model without any assumptions on the data. It is only when we need additional information such as confidence intervals for the coefficients and prediction error estimates that we must make assumptions.

Recall the b_1 coefficient represents the average effect on y when changing the x-variable by 1 unit. Let's say you are estimating a reaction rate (kinetics) from a linear least squares model, a standard step in reactor design, you would want a measure of confidence of your coefficient. For example, if you calculate the reaction rate as $k = b_1 = 0.81 \, \text{s}^{-1}$ you would benefit from knowing whether the 95% confidence interval was $k = 0.81 \pm 0.26 \, \text{s}^{-1}$ or $k = 0.81 \pm 0.68 \, \text{s}^{-1}$. In the latter case it is doubtful whether the reaction rate is of practical significance. Point estimates of the least squares model parameters are satisfactory, but the confidence interval information is richer to interpret.

We first take a look at some assumptions in least squares modelling, then return to deriving the confidence interval.

Assumptions required for analysis of the least squares model

Video for this section.

Recall that the population (true) model is $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ and b_0 and b_1 are our estimates of the model's coefficients, and e be the estimate of the true error ϵ. Note we are assuming imperfect knowledge of the y_i by lumping all errors into e_i. For example, measurement error, structural error (we are not sure the process follows a linear structure), inherent randomness, and so on.

Furthermore, our derivation for the confidence intervals of b_0 and b_1 requires that we assume:

1. Linearity of the model, and that the values of x are fixed (have no error). This implies that the error captured by ϵ is the error of y, since the $\beta_0 + \beta_1 x$ terms are fixed.

   • In an engineering situation this would mean that your x variable has much less uncertainty than the y variable; and is often true in many situations.

2. The variance of y is the same (constant) at all values of x, known as the constant error variance assumption.

   • The variability of y can be non-constant in several practical cases (e.g. our measurement accuracy deteriorates at extreme high and low levels of x).
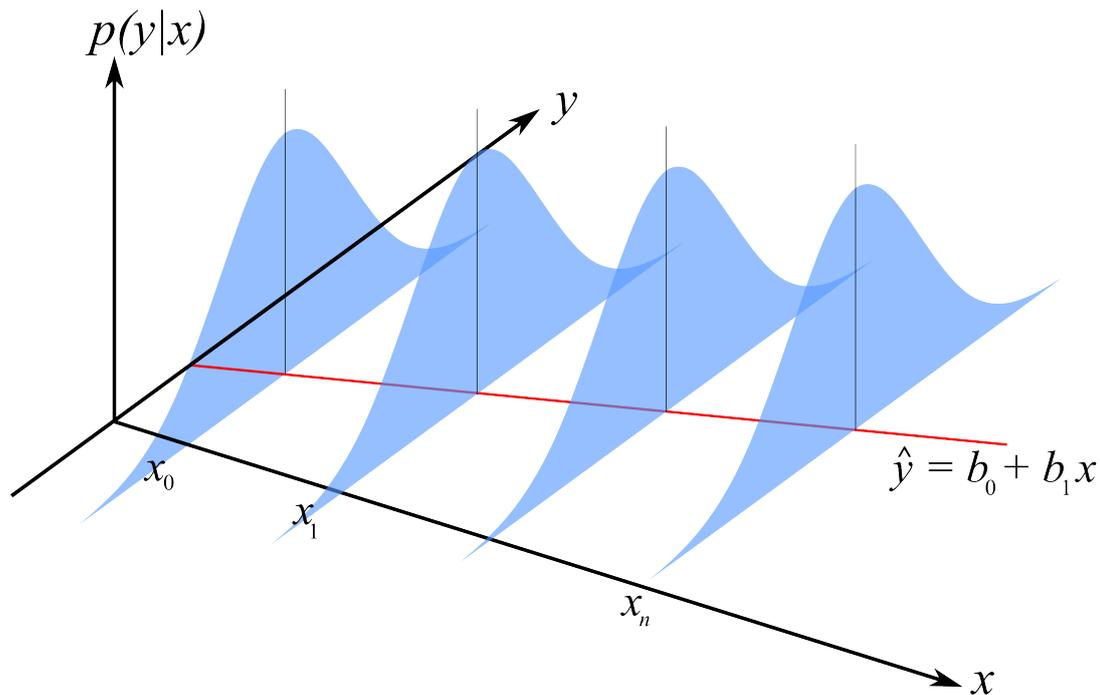
Illustration of the constant error variance assumption and the normally distributed error assumption.

3. The errors are normally distributed: $e_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$. This also implies that $y_i \sim \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma_\epsilon^2)$ from the first linearity assumption.

4. Each error is independent of the other. This assumption is often violated in cases where the observations are taken in time order on slow moving processes (e.g. if you have a positive error now, your next sample is also likely to have a positive error). We will have more to say about this later when we check for independence with an *autocorrelation test* (page 180).

5. In addition to the fact that the x values are fixed, we also assume they are independent of the error. If the x value is fixed (i.e. measured without error), then it is already independent of the error.

   • When the x values are not fixed, there are cases where the error gets larger as x gets smaller/larger.

6. All y_i values are independent of each other. This again is violated in cases where the data are collected in time order and the y_i values are autocorrelated.

Note: Derivation of the model's coefficients do not require these assumptions, only the derivation of the coefficient's confidence intervals require this.

Also, if we want to interpret the model's S_E as the estimated standard deviation of the residuals, then it helps if the residuals are normally distributed.

Confidence intervals for β_0 and β_1

Recall from our discussions on *confidence intervals* (page 63) that we need to know the mean and variance of the population from which b_0 and b_1 come. Specifically for the least squares case:

$$b_0 \sim \mathcal{N}(\beta_0, \mathcal{V}\{\beta_0\}) \qquad \text{and} \quad b_1 \sim \mathcal{N}(\beta_1, \mathcal{V}\{\beta_1\})$$

Once we know those parameters, we can create a z-value for b_0 and b_1, and then calculate the confidence interval for β_0 and β_1. So our quest now is to calculate $\mathcal{V}\{\beta_0\}$ and $\mathcal{V}\{\beta_1\}$, and we will use the 6 assumptions we made in the previous part.

Start from the equations that define b_0 and b_1 *in the prior section* (page 160) where we showed that:

$$b_0 = \bar{y} - b_1\bar{x}$$

$$b_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

$$b_1 = \sum m_i y_i \qquad \text{where} \qquad m_i = \frac{x_i - \bar{x}}{\sum_j (x_j - \bar{x})^2}$$

That last form of expressing b_1 shows that every data point contributes a small amount to the coefficient b_1. But notice how it is broken into 2 pieces: each term in the sum has a component due to m_i and one due to y_i. The m_i term is a function of the x-data only, and since we assume the x's are measured without error, that term has no error. The y_i component is the only part that has error.

So we can write:

$$b_1 = m_1 y_1 + m_2 y_2 + \ldots + m_N y_N$$

$$\mathcal{E}\{b_1\} = \mathcal{E}\{m_1 y_1\} + \mathcal{E}\{m_2 y_2\} + \ldots + \mathcal{E}\{m_N y_N\}$$

$$\mathcal{V}\{b_1\} = m_1^2 \mathcal{V}\{y_1\} + m_2^2 \mathcal{V}\{y_2\} + \ldots + m_N^2 \mathcal{V}\{y_N\}$$

$$\mathcal{V}\{b_1\} = \sum_i \left(\frac{x_i - \bar{x}}{\sum_j (x_j - \bar{x})^2}\right)^2 \mathcal{V}\{y_i\}$$

$$\mathcal{V}\{b_1\} = \frac{\mathcal{V}\{y_i\}}{\sum_j (x_j - \bar{x})^2}$$

where j is an index for all data points used to build the least squares model.

Questions:

1. So now apart from the numerator term, how could you decrease the error in your model's b_1 coefficient?

   - Use samples that are far from the mean of the x-data.

   - Use more samples.

2. What do we use for the numerator term $\mathcal{V}\{y_i\}$?

   - This term represents the variance of the y_i values at a given point x_i. If (a) there is no evidence of lack-of-fit, and (b) if y has the same error at all levels of x, then we can write that $\mathcal{V}\{y_i\} = \mathcal{V}\{e_i\} = \frac{\sum e_i^2}{n-k}$, where n is the number of data points used, and k is the number of coefficients estimated (2 in this case). The $n - k$ quantity is the degrees of freedom.

Now for the variance of $b_0 = \overline{y} - b_1\overline{x}$. The only terms with error are b_1, and $\overline{y}$. So we can derive that:

$$\mathcal{V}\{b_0\} = \left(\frac{1}{N} + \frac{\overline{x}^2}{\sum_j (x_j - \overline{x})^2}\right) \mathcal{V}\{y_i\}$$

Summary of important equations

$$\mathcal{V}\{\beta_0\} \approx \mathcal{V}\{b_0\} = \left(\frac{1}{N} + \frac{\overline{x}^2}{\sum_j (x_j - \overline{x})^2}\right) \mathcal{V}\{y_i\}$$

$$\mathcal{V}\{\beta_1\} \approx \mathcal{V}\{b_1\} = \frac{\mathcal{V}\{y_i\}}{\sum_j (x_j - \overline{x})^2}$$

where $\quad \mathcal{V}\{y_i\} = \mathcal{V}\{e_i\} = \dfrac{\sum e_i^2}{n - k}$, if there is no lack-of-fit and the y's are independent of each other.

For convenience we will define some short-hand notation, which is common in least squares:

$$S_E^2 = \mathcal{V}\{e_i\} = \mathcal{V}\{y_i\} = \frac{\sum e_i^2}{n - k} \qquad \text{or } S_E = \sqrt{\frac{\sum e_i^2}{n - k}}$$

$$S_E^2(b_0) = \mathcal{V}\{b_0\} = \left(\frac{1}{N} + \frac{\overline{x}^2}{\sum_j (x_j - \overline{x})^2}\right) S_E^2$$

$$S_E^2(b_1) = \mathcal{V}\{b_1\} = \frac{S_E^2}{\sum_j (x_j - \overline{x})^2}$$

You will see that S_E is an estimate of the standard deviation of the error (residuals), while $S_E(b_0)$ and $S_E(b_1)$ are the standard deviations of estimates for b_0 and b_1 respectively.

Now it is straight forward to construct **confidence intervals for the least squares model parameters**. You will also realize that we have to use the t-distribution, because we are using an estimate of the variance.

$$
\begin{array}{ccccc}
-c_t & \leq & \dfrac{b_0 - \beta_0}{S_E(b_0)} & \leq & +c_t \\
b_0 - c_t S_E(b_0) & \leq & \beta_0 & \leq & b_0 + c_t S_E(b_0)
\end{array}
\qquad
\begin{array}{ccccc}
-c_t & \leq & \dfrac{b_1 - \beta_1}{S_E(b_1)} & \leq & +c_t \\
b_1 - c_t S_E(b_1) & \leq & \beta_1 & \leq & b_1 + c_t S_E(b_1)
\end{array}
$$

(4.9)

Example

Returning *back to our ongoing example* (page 161), we can calculate the confidence interval for β_0 and β_1. We calculated earlier already that $b_0 = 3.0$ and $b_1 = 0.5$. Using these values we can calculate the standard error:

```
                                                  R code
x <- c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y <- c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96,
       7.24, 4.26, 10.84, 4.82, 5.68)

# "Calculate for me the linear model,
# where y is described by x"
mod.ls <- lm(y ~ x)

# We can find what the "b0" and "b1"
# values are in several different ways:
summary(mod.ls)
```

(continues on next page)

```
# or using
print('The model coefficients are: ')
coefficients(mod.ls)

# Model predictions:
print('The predicted values are: ')
predict(mod.ls)
# 8.001   7.000   9.501   7.501   8.501
# 10.001  6.00   5.000   9.001   6.500   5.501

# Prediction error = observed - predicted
error <- y - predict(mod.ls)
N <- length(x)

# The SE = standard error = 1.236603
std.error <- sqrt(sum(error^2) / (N-2))
paste0('Standard error SE = ',
        round(std.error, 3))
```

Use that S_E value to calculate the confidence intervals for β_0 and β_1, and use that $c_t = 2.26$ at the 95% confidence level. You can calculate this value in R using `qt(0.975, df=(N-2))`. There are $n - 2$ degrees of freedom, the number of degrees of freedom used to calculate S_E.

First calculate the S_E value and the standard errors for the b_0 and b_1. Substitute these into the equation for the confidence interval and calculate:

$$S_E = 1.237$$

$$S_E^2(b_1) = \frac{S_E^2}{\sum_j (x_j - \overline{x})^2} = \frac{1.237^2}{110} = 0.0139$$

$$S_E^2(b_0) = \left(\frac{1}{N} + \frac{\overline{x}^2}{\sum_j (x_j - \overline{x})^2} \right) S_E^2 = \left(\frac{1}{11} + \frac{9^2}{110} \right) 1.237^2 = 1.266$$

The 95% confidence interval for β_0:

$$
\begin{array}{ccccc}
-c_t & \leq & \dfrac{b_0 - \beta_0}{S_E(b_0)} & \leq & +c_t \\
3.0 - 2.26 \times \sqrt{1.266} & \leq & \beta_0 & \leq & 3.0 + 2.26 \times \sqrt{1.266} \\
0.457 & \leq & \beta_0 & \leq & 5.54
\end{array}
$$

The confidence interval for β_1:

$$
\begin{array}{ccccc}
-c_t & \leq & \dfrac{b_1 - \beta_1}{S_E(b_1)} & \leq & +c_t \\
0.5 - 2.26 \times \sqrt{0.0139} & \leq & \beta_1 & \leq & 0.5 + 2.26 \times \sqrt{0.0139} \\
0.233 & \leq & \beta_1 & \leq & 0.767
\end{array}
$$

The plot shows the effect of varying the slope parameter, b_1, from its lower bound to its upper bound. Notice that the slope always passes through the mean of the data $(\overline{x}, \overline{y})$.
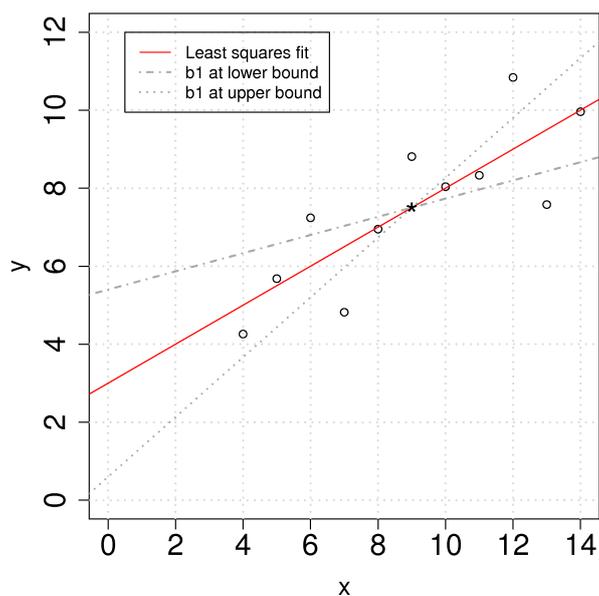
In many cases the confidence interval for the intercept is not of any value because the data for x is so far away from zero, or the true value of the intercept is not of concern for us.

R code

```r
x <- c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y <- c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96,
       7.24, 4.26, 10.84, 4.82, 5.68)

# "Calculate for me the linear model,
# where y is described by x"
mod.ls <- lm(y ~ x)

# You can (and should at the beginning)
# calculate the confidence intervals as shown
# above. But there is a short-cut, to save
# time, and is less error prone:
confint(mod.ls)

#                 2.5 %    97.5 %
# (Intercept) 0.4557369 5.5444449
# x           0.2333701 0.7668117

# If you want the confidence interval at any
# other level, for example, at the 90% level:
confint(mod.ls, level=0.90)

#                 5 %       95 %
# (Intercept) 0.9383030 5.061879
# x           0.2839568 0.716225

# Compare this to the calculated value by hand
# above. It is exactly the same!
```

4.7.3 Prediction error estimates for the y-variable

Apart from understanding the error in the model's coefficient, we also would like an estimate of the error when predicting $\hat{y}_i$ from the model, $y_i = b_0 + b_1 x_i + e_i$ for a new value of x_i. This is known as the prediction interval, or prediction error interval.

A naive first attempt

We might expect the error is related to the average size of the residuals. After all, *our assumptions we made earlier* (page 166) showed the standard error of the residuals was the standard error of the y:

$$S_E^2 = \mathcal{V}\{e_i\} = \mathcal{V}\{y_i\} = \frac{\sum e_i^2}{n - k}.$$
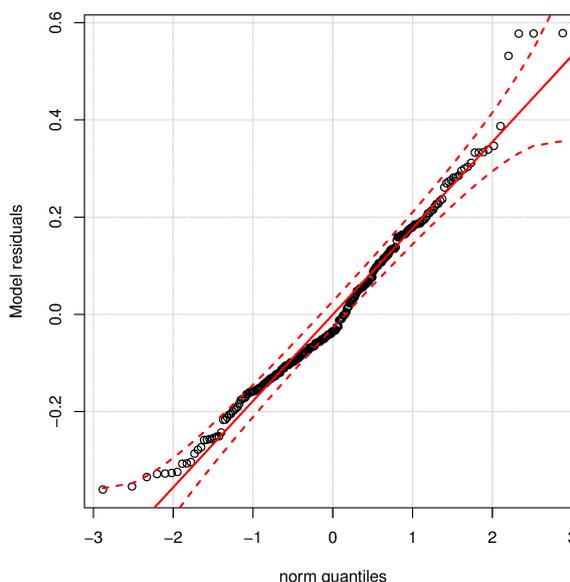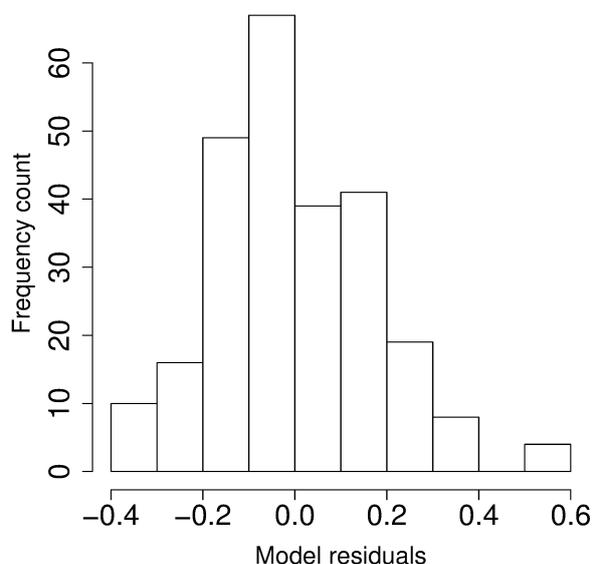
A typical histogram of the residuals looks as shown here: it is always centered around zero, and appears to be normally distributed. So we could expect to write our prediction error as $\hat{y}_{\text{new}} = (b_0 + b_1 x_{\text{new}}) \pm c \cdot S_E$, where c is the number of standard deviations around the average residual, for example we could have set $c = 2$, approximating the 95% confidence limit.

But there is something wrong with that error estimate. It says that our prediction error is constant at any value of x_i, even at values far outside the range where we built the model. This is a naive estimate of the prediction error. We have forgotten that coefficients b_0 and b_1 have error, and that error must be propagated into $\hat{y}_{\text{new}}$.

This estimate is however a reasonable guess for the prediction interval when you only know the model's S_E and don't have access to a calculator or computer to calculate the proper prediction interval, shown next.

A better attempt to construct prediction intervals for the least squares model

Note: A good reference for this section is Draper and Smith, *Applied Regression Analysis*, page 79.

The derivation for the prediction interval is similar to that for b_1. We require an estimate for the variance of the predicted y at at given value of x. Let's fix our x value at x_* and since $b_0 = \overline{y} - b_1\overline{x}$, we can write the prediction at this fixed x value as $\hat{y}_* = \overline{y} - b_1(x_* - \overline{x})$.

$$\mathcal{V}\{y_*\} = \mathcal{V}\{\overline{y}\} + \mathcal{V}\{b_1(x_* - \overline{x})\} + 2\mathrm{Cov}\{\overline{y}, b_1(x_* - \overline{x})\}$$

$$\mathcal{V}\{y_*\} = \frac{S_E^2}{n} + (x_* - \overline{x})^2 S_E^2(b_1) + 0$$

You may read the reference texts for the interesting derivation of this variance. However, this is only the variance of the average predicted value of y. In other words, it is the variance we expect if we repeatedly brought in observations at x_*. The prediction error of an individual observation, x_i, and its corresponding prediction, $\hat{y}_i$, is inflated slightly further:

$$\mathcal{V}\{\hat{y}_i\} = S_E^2 \left(1 + \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_j (x_j - \overline{x})^2}\right), \text{where } j \text{ is the index for all points used to build the least squares}$$
model.

We may construct a prediction interval in the standard manner, assuming that $\hat{y}_i \sim \mathcal{N}\left(\overline{\hat{y}_i}, \mathcal{V}\{\hat{y}_i\}\right)$. We will use an estimate of this variance since we do not know the population variance. This requires we use the t-distribution with $n - k$ degrees of freedom, at a given degree of confidence, e.g. 95%.

$$-c_t \quad < \quad \frac{\hat{y}_i - \overline{\hat{y}_i}}{\sqrt{V\{\hat{y}_i\}}} \quad < \quad +c_t$$
$$\hat{y}_i - c_t\sqrt{V\{\hat{y}_i\}} \quad < \quad \overline{\hat{y}_i} \quad < \quad \hat{y}_i + c_t\sqrt{V\{\hat{y}_i\}}$$

This is a prediction interval for a new prediction, $\hat{y}_i$ given a new x value, x_i. For example, if $\hat{y}_i = 20$ at a given value of x_i, and if $c_t\sqrt{V\{\hat{y}_i\}} = 5$, then you will usually see written in reports and documents that, the prediction was 20 ± 5. A more correct way of expressing this concept is to say the true prediction at the value of x_i lies within a bound from 15 to 25, with 95% confidence.

Implications of the prediction error of a new y

Let's understand the interpretation of $\mathcal{V}\{\hat{y}_i\} = S_E^2 \left(1 + \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_j (x_j - \overline{x})^2}\right)$ as the variance of the predicted $\hat{y}_i$ at the given value of x_i. Using the previous example where we calculated the least squares line, now:

1. Now let's say our x_{new} happens to be $\overline{x}$, the center point of our data. Write down the upper and lower value of the prediction bounds for the corresponding $\hat{y}$, given that $c_t = 2.26$ at the 95% confidence level.

   - The LB $= \hat{y}_i - c_t\sqrt{V\{\hat{y}_i\}} = 7.5 - 2.26 \times \sqrt{(1.237)^2\left(1 + \frac{1}{11} + \frac{(x_i - \overline{x})^2}{\sum_j (x_j - \overline{x})^2}\right)} = $
     $7.5 - 2.26 \times 1.29 = 7.50 - 2.917 = 4.58$

   - The UB $= \hat{y}_i + c_t\sqrt{V\{\hat{y}_i\}} = 7.5 + 2.26 \times \sqrt{(1.237)^2\left(1 + \frac{1}{11} + \frac{(x_i - \overline{x})^2}{\sum_j (x_j - \overline{x})^2}\right)} = $
     $7.5 + 2.26 \times 1.29 = 7.50 + 2.917 = 10.4$

2. Now move left and right, away from $\bar{x}$, and mark the confidence intervals. What general shape do they have?

- The confidence intervals have a quadratic shape due to the square term under the square root. The smallest prediction error will always occur at the center of the model, and expands progressively wider as one moves away from the model center. This is illustrated in the figure and makes intuitive sense as well.
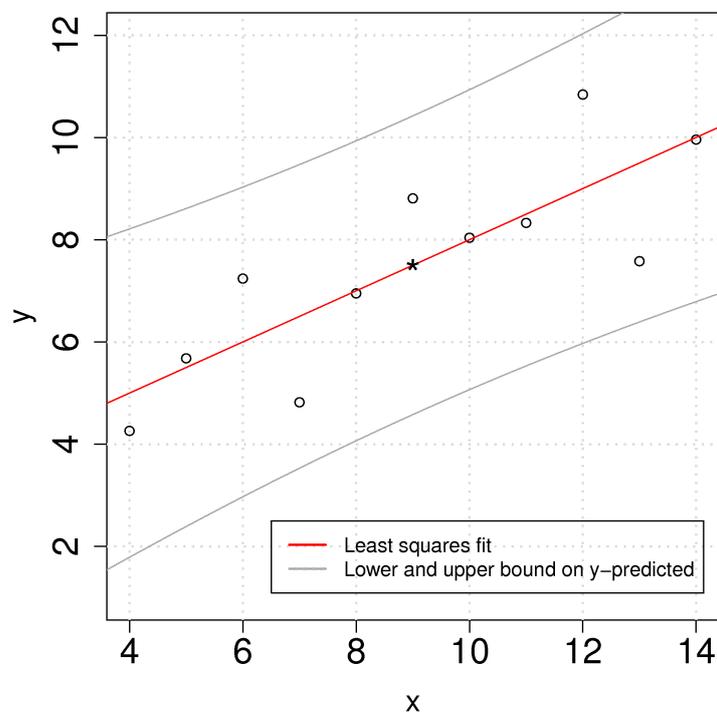
4.7.4 Interpretation of software output

To complete this section we show how to interpret the output from computer software packages. Most packages have very standardized output, and you should make sure that whatever package you use, that you can interpret the estimates of the parameters, their confidence intervals and get a feeling for the model's performance.

The following output is obtained in R for the *example* (page 161) we have been using in this section. The Python version follows below.

```
                                              R code
x <- c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y <- c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96,
       7.24, 4.26, 10.84, 4.82, 5.68)

# "Calculate for me the linear model,
# where y is described by x"
mod.ls <- lm(y ~ x)

summary(mod.ls)
```

and produces this output:

```
Call:
lm(formula = y ~ x)
```

(continues on next page)

```
Residuals:
    Min      1Q   Median      3Q      Max
-1.92127 -0.45577 -0.04136  0.70941  1.83882

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
x             0.5001     0.1179   4.241  0.00217 **
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 1.237 on 9 degrees of freedom
Multiple R-squared: 0.6665,     Adjusted R-squared: 0.6295
F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.002170
```

Make sure you can calculate the following values using the equations developed so far, based on the above software output:

- The intercept term b_0 = 3.0001.

- The slope term b_1 = 0.5001.

- The standard error of the model, S_E = 1.237, using $n - k = 11 - 2 = 9$ degrees of freedom.

- Using the standard error, calculate the standard error for the intercept = $S_E(b_0) = 1.1247$.

- Using the standard error, calculate the standard error for the slope = $S_E(b_1) = 0.1179$.

- The z-value for the b_0 term is 2.667 (R calls this the t value in the printout, but in our notes we have called this $z = \dfrac{b_0 - \beta_0}{S_E(b_0)}$; the value that we compare to the t-statistic and used to create the confidence interval).

- The z-value for the b_1 term is 4.241 (see the above comment again).

- The two probability values, Pr(>|t|), for b_0 and b_1 should be familiar to you; they are the probability with which we expect to find a value of z greater than the calculated z-value (called t value in the output above). The smaller the number, the more confident we can be the confidence interval contains the parameter estimate.

- You can construct the confidence interval for b_0 or b_1 by using their reported standard errors and multiplying by the corresponding t-value. For example, if you want 99% confidence limits, then look up the 99% values for the t-distribution using $n - k$ degrees of freedom, in this case it would be qt((1-0.99)/2, df=9), which is ± 3.25. So the 99% confidence limits for the slope coefficient would be $[0.5 - 3.25 \times 0.1179; 0.5 + 3.25 \times 0.1179] = [0.12; 0.88]$.

- The $R^2 = 0.6665$ value.

- Be able to calculate the residuals: $e_i = y_i - \hat{y}_i = y_i - b_0 - b_1 x_i$. We expect the median of the residuals to be around 0, and the rest of the summary of the residuals gives a feeling for how far the residuals range about zero.

Using Python, you can run the following code:

```Python
import numpy as np
import statsmodels.api as sm

X = np.array([10, 8, 13, 9, 11, 14,
              6, 4, 12, 7, 5])
y = np.array([8.04, 6.95, 7.58, 8.81,
```

```
          8.33, 9.96, 7.24, 4.26,
          10.84, 4.82, 5.68])

# We do want to estimate a 'b0' term
X = sm.add_constant(X)
model = sm.OLS(y, X)
results = model.fit()
print(results.summary())
print('Standard error = {}'.format(\
    np.sqrt(results.scale)))
```

which produces the following output:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.667
Model:                            OLS   Adj. R-squared:                  0.629
Method:                 Least Squares   F-statistic:                     17.99
Date:                Tue, 01 Jan 2019   Prob (F-statistic):            0.00217
Time:                        00:00:00   Log-Likelihood:                -16.841
No. Observations:                  11   AIC:                             37.68
Df Residuals:                       9   BIC:                             38.48
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.0001      1.125      2.667      0.026       0.456       5.544
x1             0.5001      0.118      4.241      0.002       0.233       0.767
==============================================================================
Omnibus:                        0.082   Durbin-Watson:                   3.212
Prob(Omnibus):                  0.960   Jarque-Bera (JB):                0.289
Skew:                          -0.122   Prob(JB):                        0.865
Kurtosis:                       2.244   Cond. No.                         29.1
==============================================================================

Standard error = 1.2366033227263207
```

As for the R code, we can see at a glance:

- The intercept term $b_0 = 3.0001$.

- The slope term $b_1 = 0.5001$.

- The standard error of the model, $S_E = 1.237$, using $n - k = 11 - 2 = 9$ degrees of freedom. The summary output table does not show the standard error, but you can get it from `np.sqrt(results.scale)`, where `results` is the Python object from fitting the linear model.

- Using the standard error, calculate the standard error for the intercept = $S_E(b_0) = 1.1247$, which is reported directly in the table.

- Using the standard error, calculate the standard error for the slope = $S_E(b_1) = 0.1179$, which is reported directly in the table.

- The z-value for the b_0 term is 2.667 (Python calls this the `t`-value in the printout, but in our notes we have called this $z = \dfrac{b_0 - \beta_0}{S_E(b_0)}$; the value that we compare to the t-statistic and used to create the confidence interval).

- The z-value for the b_1 term is 4.241 (see the above comment again).

- The two probability values, P>|t|, for b_0 and b_1 should be familiar to you; they are the probability with which we expect to find a value of z greater than the calculated z-value (called t value in the output above). The smaller the number, the more confident we can be the confidence interval contains the parameter estimate.

- You can construct the confidence interval for b_0 or b_1 by using their reported standard errors and multiplying by the corresponding t-value. For example, if you want 99% confidence limits, then look up the 99% values for the t-distribution using $n - k$ degrees of freedom, in this case it would be from scipy.stats import t; t.ppf(1-(1-0.99)/2, df=9), which is ±3.25. So the 99% confidence limits for the slope coefficient would be
  $[0.5 - 3.25 \times 0.1179; 0.5 + 3.25 \times 0.1179] = [0.117; 0.883]$. However, the table output gives you the 95% confidence interval. Under the column 0.025 and 0.975 (leaving 2.5% in the lower and upper tail respectively). For the slope coefficient, for example, this interval is [0.233; 0.767]. If you desire, for example, the 99% confidence interval, you can adjust the code:
  print(results.summary(alpha=1-0.99))

- The $R^2 = 0.6665$ value.

- Be able to calculate the residuals: $e_i = y_i - \hat{y}_i = y_i - b_0 - b_1 x_i$.

4.8 Investigating an existing linear model

4.8.1 Summary so far

We have introduced the linear model, $y = \beta_0 + \beta_1 x + \varepsilon$ and shown how to estimate the 2 model parameters, $b_0 = \hat{\beta}_0$ and $b_1 = \hat{\beta}_1$. This can be done on any data set without any additional assumptions. But, in order to calculate confidence intervals so we can better understand our model's performance, we must make several assumptions of the data. In the next sections we will learn how to interpret various plots that indicate when these assumptions are violated.

Along the way, while investigating these assumptions, we will introduce some new topics:

- Transformations of the raw data to better meet our assumptions

- Leverage, outliers, influence and discrepancy of the observations

- Inclusion of additional terms in the linear model (multiple linear regression, MLR)

- The use of training and testing data

It is a common theme in any modelling work that the most informative plots are those of the residuals - the unmodelled component of our data. We expect to see no structure in the residuals, and since the human eye is excellent at spotting patterns in plots, it is no surprise that various types of residual plots are used to diagnose problems with our model.

4.8.2 The assumption of normally distributed errors

We look for normally distributed errors because if they are non-normal, then the standard error, S_E and the other variances that depend on S_E, such as $\mathcal{V}(b_1)$, could be inflated, and their interpretation could be in doubt. This might, for example, lead us to infer that a slope coefficient is not important when it actually is.

This is one of the easiest assumptions to verify: use a *q-q plot* (page 50) to assess the distribution of the residuals. Do *not* plot the residuals in sequence or some other order to verify normality - it is extremely difficult to see that. A q-q plot highlights very clearly when tails from the residuals are too heavy. A histogram may also be used, but for real data sets, the choice of bin width can dramatically distort the interpretation - rather use a q-q plot. Some code for R:

```
model = lm(...)
library(car)
qqPlot(model)           # uses studentized residuals
qqPlot(resid(model))    # uses raw residuals
```

If the residuals appear non-normal, then attempt the following:

- Remove the outlying observation(s) in the tails, but only after careful investigation whether that outlier really was unusual

- Use a suitable transformation of the y-variable

- Add *additional terms to the least squares model* (page 185)

The simple example shown here builds a model that predicts the price of a used vehicle using only the mileage as an explanatory variable.
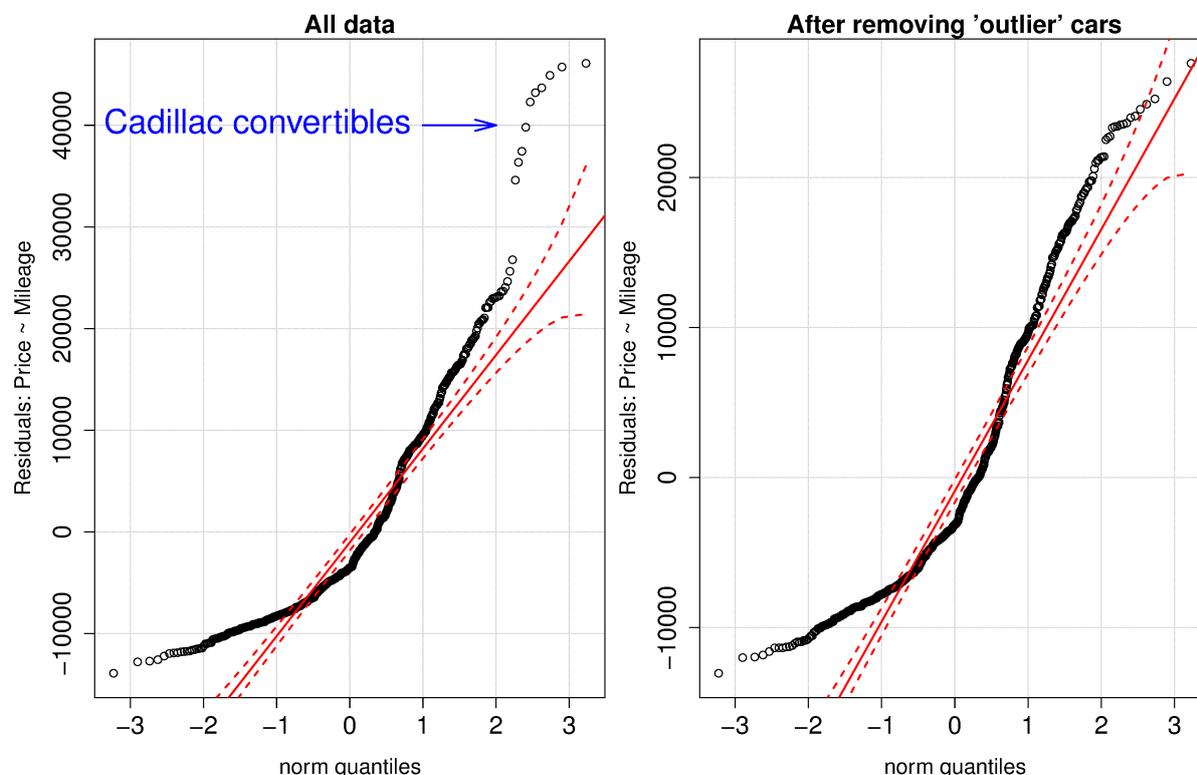
The group of outliers were due to 10 observations of a certain class of vehicle (Cadillac convertibles) that distorted the model. We removed these observations, which now limits our model to be useful

only for other vehicle types, but we gain a smaller standard error and a tighter confidence interval. These residuals are still very non-normal though.

$$\text{Before}: \quad b_1 = -0.173 \qquad -0.255 \leq \beta_1 \leq -0.0898 \qquad S_E = \$9789$$
$$\text{After}: \quad b_1 = -0.155 \qquad -0.230 \leq \beta_1 \leq -0.0807 \qquad S_E = \$8655$$

The slope coefficient (*interpretation*: each extra mile on the odometer reduces the sale price on average by 15 to 17 cents) has a tighter confidence interval after removing those unusual observations.

Removing the Cadillac cars from our model indicates that there is more than just mileage that affect their resale value. In fact, the lack of normality, and structure in the residuals leads us to ask which other explanatory variables can be included in the model.

In the next fictitious example the y-variable is non-linearly related to the x-variable. This non-linearity in the y shows up as non-normality in the residuals if only a linear model is used. The residuals become more linearly distributed when using a square root transformation of the y before building the linear model.
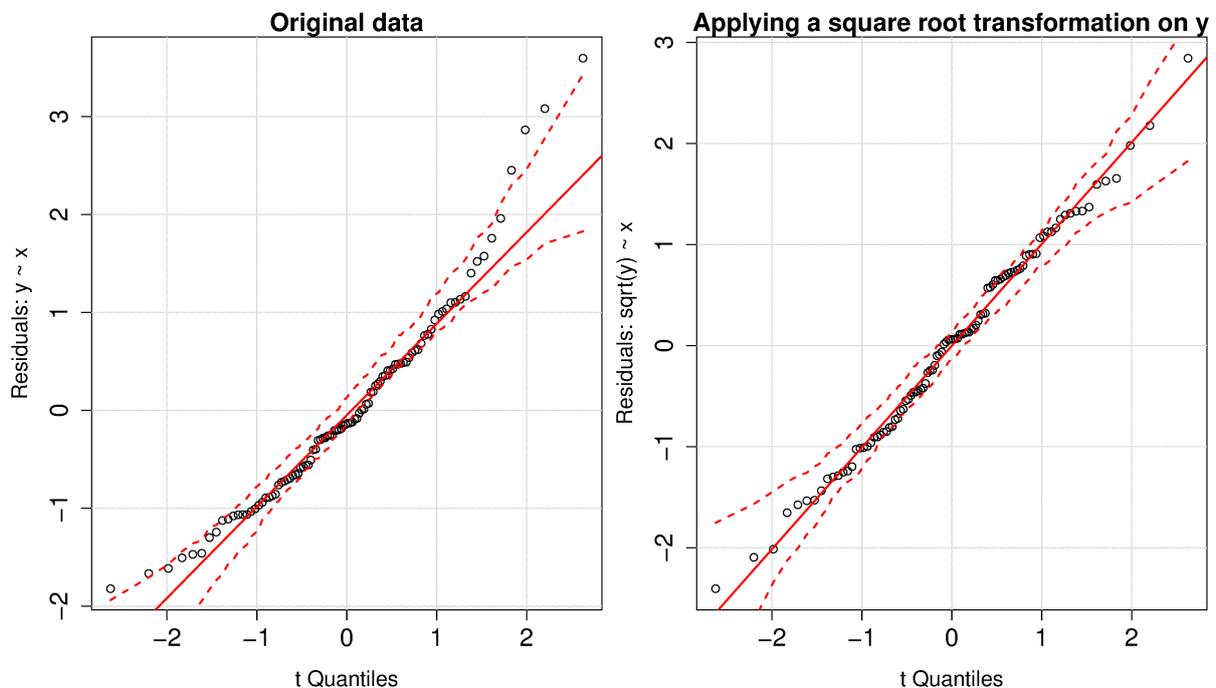
More discussion about transformations of the data is given in the section on *model linearity* (page 182).

4.8.3 Non-constant error variance

It is common in many situations that the variability in y increases or decreases as y is increased (e.g. certain properties are more consistently measured at low levels than at high levels). Similarly, variability in y increases or decreases as x is increased (e.g. as temperature, x, increases the variability of a particular y increases).

Violating the assumption of non-constant error variance increases the standard error, S_E, undermining the estimates of the confidence intervals, and other analyses that depend on the standard error. Fortunately, it is only problematic if the non-constant variance is extreme, so we can tolerate minor violations of this assumption.

To detect this problem you should plot:

- the predicted values of y (on the x-axis) against the residuals (y-axis)

- the x values against the residuals (y-axis)

This problem reveals itself by showing a fan shape across the plot; an example is shown in the figure.
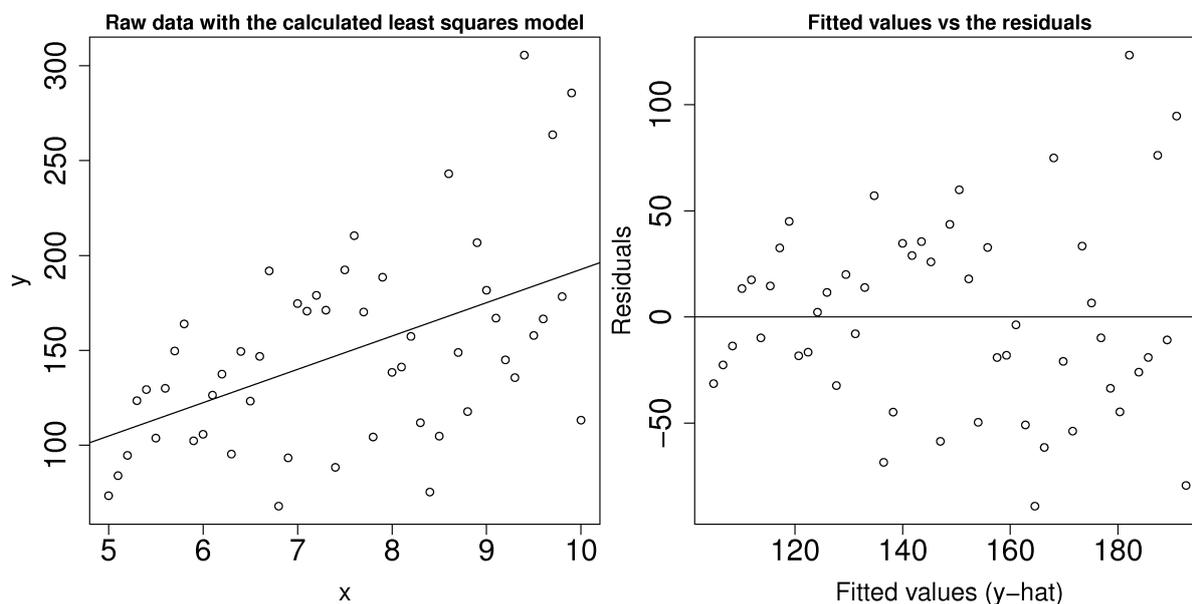
To counteract this problem one can use weighted least squares, with smaller weights on the high-variance observations, i.e. apply a weight inversely proportional to the variance. Weighted least squares minimizes: $f(\mathrm{b}) = \sum_i^n (w_i e_i)^2$, with different weights, w_i for each error term. More on this topic can be found in the book by Draper and Smith (p 224 to 229, 3rd edition).

4.8.4 Lack of independence in the data

The assumption of independence in the data requires that values in the y variable are independent. Given that we have assumed the x variable to be fixed, this implies that the errors, e_i are independent. The reason for independence is required for the central limit theorem, which was used to derive the various standard errors.

Data are not independent when they are correlated with each other. This is common on slow moving processes: for example, measurements taken from a large reactor are unlikely to change much from one minute to the next.

Treating this problem properly comes under the topic of time-series analysis, for which a number of excellent textbooks exist, in particular the one by Box and Jenkins. But we will show how to detect autocorrelation, and provide a make-shift solution to avoid it.

If you suspect that there may be lack of independence, use plots of the residuals in time order. Look for patterns such as slow drifts, or rapid criss-crossing of the zero axis.

Time order of the y-variable | Raw data with the calculated least squares model | Residuals in time-order
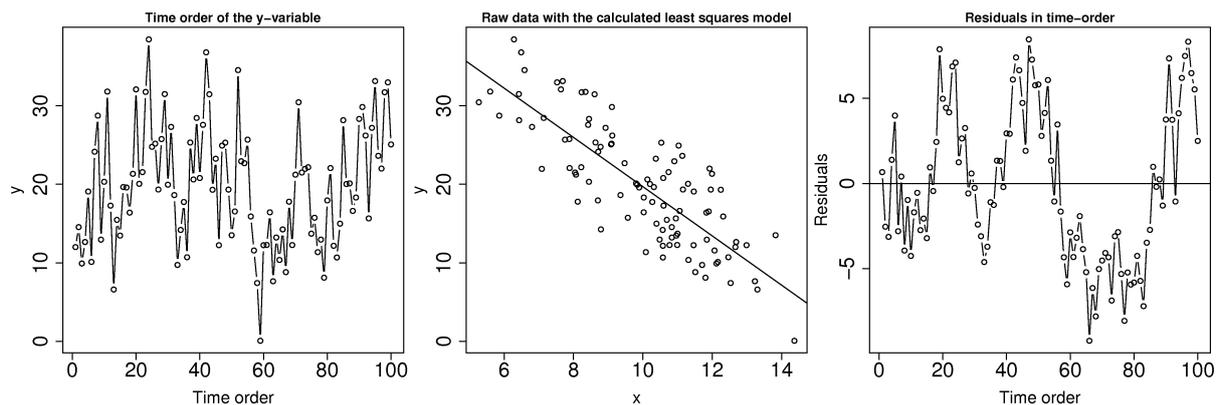
One way around the autocorrelation is to subsample - use only every k^{th} sample, where k is a certain number of gaps between the points. How do we know how many gaps to leave? Use the autocorrelation function[78] to determine how many samples. You can use the `acf(...)` function in R, which will show how many significant lags there are between observations. Calculating the autocorrelation accurately requires a large data set, which is a requirement anyway if you need to subsample your data to obtain independence.

Here are some examples of the autocorrelation plot: in the first case you would have to leave at least 16 samples between each sub-sample, while the second and third cases require a gap of 1 sample, i.e. use only every second data point.

Long autocorrelation: $\phi = 0.95$ | Medium autocorrelation: $\phi = 0.3$ | Negative autocorrelation: $\phi = -0.6$
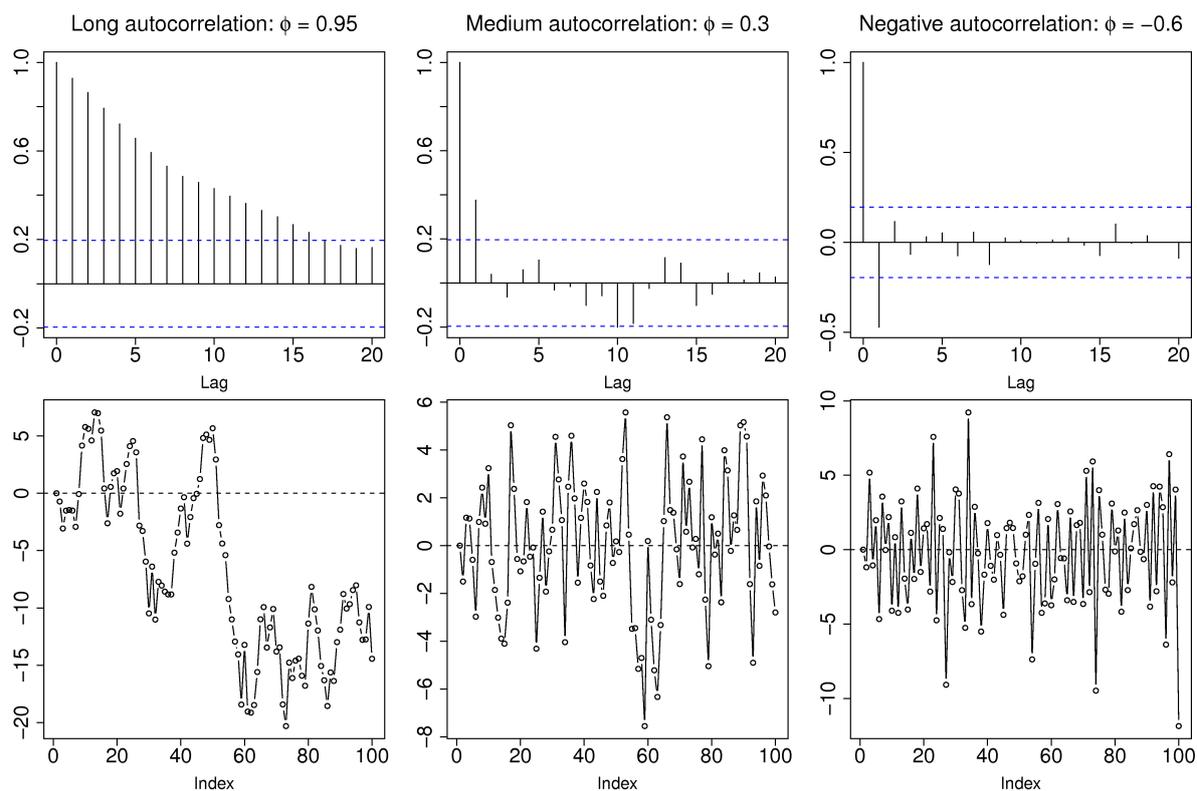
Another test for autocorrelation is the Durbin-Watson test. For more on this test see the book by Draper and Smith (Chapter 7, 3rd edition); in R you can use the `durbinWatsonTest(model)` function in `library(car)`. Try generating autocorrelation of varying strength (positive, e.g. `phi_long = 0.80` and negative, e.g. `phi_long = -0.75`) in the code below. Inspect the plots which are generated as a result, especially the time order plot: get a feeling for what a strong and weak

---

[78] https://en.wikipedia.org/wiki/Autocorrelation

positive/negative correlation looks like in the time order.

```R
# Adjust this autocorrelation parameter:                  ─ R code ─
phi_long = 0.80


N = 1005
data <- numeric(N)
for (k in 2:N){
  data[k] = rnorm(1, sd=4) +
                     phi_long * data[k-1]
}
x <- data + 50
summary(x)

# Plot autocorrelation in the first 100 points
plot(data[1:100], type='b',
    main='Raw data', xlab = 'Time order')

plot.new()
lims = c(30,70)
plot(x[1:1000], x[2:1001], asp=1,
    xlim=lims, ylim=lims)
model <- lm(x[2:1001] ~ x[1:1000])
abline(model, col="darkgreen", lwd=2)
text(30, 30, paste("Correlation = r = ",
                   round(cor(x[2:1001],
                          x[1:1000]), 2)),
    col="darkgreen", cex=1.5, adj = c(0, NA))
```

4.8.5 Linearity of the model (incorrect model specification)

Recall that the linear model is just a tool to either learn more about our data, or to make predictions. Many cases of practical interest are from systems where the general theory is either unknown, or too complex, or known to be non-linear.

Certain cases of non-linearity can be dealt with by simple transformations of the raw data: use a **non-linear transformation** of the raw data and then build a *linear model* as usual. An alternative method which fits the non-linear function, using concepts of optimization, by minimizing the sum of squares is covered in a section on non-linear regression. Again the book by Draper and Smith (Chapter 24, 3rd edition), may be consulted if this topic is of further interest to you. Let's take a look at a few examples.

We saw earlier a case where a square-root transformation of the y variable made the residuals more normally distributed. There is in fact a sequence of transformations that can be tried to modify the distribution of a single variable: $x_{\text{transformed}} \leftarrow x_{\text{original}}^{p}$.

- When p goes from 1 and higher, say 1.5, 1.75, 2.0, *etc*, it compresses small values of x and inflates larger values.

- When p goes down from 1, 0.5 ($\sqrt{x}$), 0.25, -0.5, -1.0 ($1/x$), -1.5, -2.0, *etc*, it compresses large values of x and inflates smaller values.

- The case of $\log(x)$ approximates $p = 0$ in terms of the severity of the transformation.

In other instances we may know from first-principles theory, or some other means, what the expected non-linear relationship is between an x and y variable.

- In a distillation column the temperature, T is inversely proportional to the logarithm of the vapour pressure, P. So fit a linear model, $y = b_0 + b_1 x$ where $x \leftarrow 1/T$ and where $y \leftarrow P$. The slope

coefficient will have a different interpretation and a different set of units as compared to the case when predicting vapour pressure directly from temperature.

- If $y = p \times q^x$, then we can take logs and estimate this equivalent linear model:
  $\log(y) = \log(p) + x\log(q)$, which is of the form $y = b_0 + b_1 x$. So the slope coefficient will be an estimate of $\log(q)$.

- If $y = \dfrac{1}{p + qx}$, then invert both sides and estimate the model $y = b_0 + b_1 x$ where $b_0 \leftarrow p$, $b_1 \leftarrow q$ and $y \leftarrow 1/y$.

- There are plenty of other examples, some classic cases being the non-linear models that arise during reactor design and biological growth rate models. With some ingenuity (taking logs, inverting the equation), these can often be simplified into linear models.

- Some cases cannot be linearized and are best estimated by non-linear least squares methods. However, a make-shift approach which works quite well for simple cases is to perform a grid search. For example imagine the equation to fit is $y = \beta_1 \left(1 - e^{-\beta_2 x}\right)$, and you are given some data pairs (x_i, y_i). Then for example, create a set of trial values $\beta_1 = [10, 20, 30, 40, 50]$ and $\beta_2 = [0.0, 0.2, 0.4, 0.8]$. Build up a grid for each combination of β_1 and β_2 and calculate the sum of squares objective function for each point in the grid. By trial-and-error you can converge to an approximate value of β_1 and β_2 that best fit the data. You can then calculate S_E, but not the confidence intervals for β_1 and β_2.

Before launching into various transformations or non-linear least squares models, bear in mind that the linear model may be useful over the region of interest. In the figure we might only be concerned with using the model over the region shown, even though the system under observation is known to behave non-linearly over a wider region of operation.
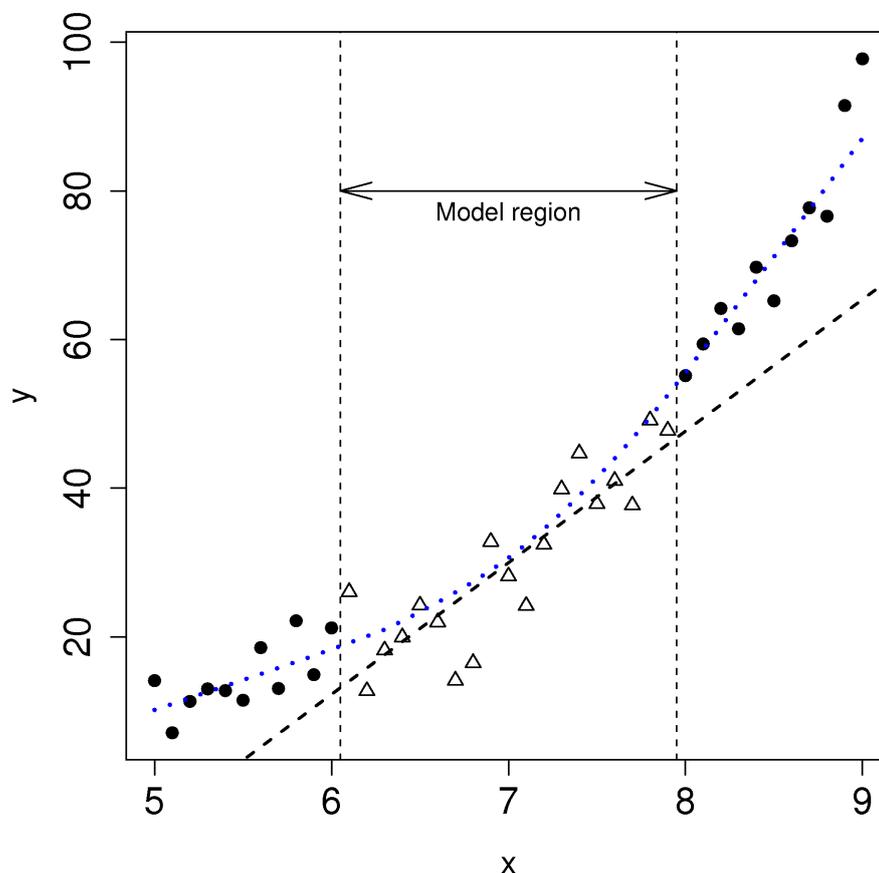
How can we detect when the linear model is not sufficient anymore? While a q-q plot might hint at problems, better plots are the same two plots for detecting *non-constant error variance* (page 179):

- the predicted values of y (on the x-axis) against the residuals (y-axis)

- the x values against the residuals (y-axis)

Here we show both plots for the example just prior (where we used a linear model for a smaller sub-region). The last two plots look the same, because the predicted $\hat{y}$ values, $\hat{y} = b_0 + b_1 x_1$; in other words, just a linear transformation of the x values.
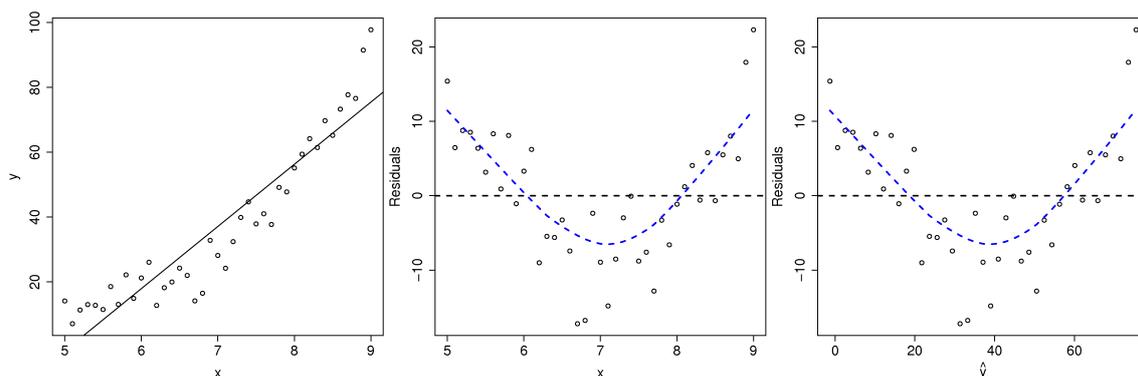
Transformations are considered successful once the residuals appear to have no more structure in them. Also bear in mind that structure in the residuals might indicate the model is missing an additional explanatory variable (see the section on *multiple linear regression* (page 185)).

Another type of plot to diagnose non-linearity present in the linear model is called a *component-plus-residual plot* or a *partial-residual plot*. This is an advanced topic not covered here, but well covered in the *Fox reference* (page 152).

4.9 Summary of steps to build and investigate a linear model

1. Plot the data to assess model structure and degree of correlation between the x and y variable.

```
plot(x, y)              # plot the raw data
lines(lowess(x,y))      # superimpose non-parametric smoother to see correlation
```

2. Fit the model and examine the printed output.

```
model <- lm(y ~ x)    # fit the model: "y as described by variable x"
summary(model)

confint(model)
```

- Investigate the model's standard error, how does it compare to the range of the y variable?

- Calculate confidence intervals for the model parameters and interpret them.

3. Visualize the model's predictions in the context of the model building data.

```
plot(x, y)
lines(lowess(x,y))          # show the smoother
abline(model, col="red")    # and show the least squares model
```

4. Plot a normal probability plot, or a q-q plot, of the residuals. Are they normally distributed? If not, investigate if a transformation of the y variable might improve them. But also see the additional plots on checking for non-linearity and consider adding extra explanatory variables.

```
library(car)
qqPlot(resid(model))
```

5. Plot the residuals against the x-values. We expect to see no particular structure. If you see trends in the data, it indicates that a transformation of the x variable might be appropriate, or that there are unmodelled phenomena in the y variable - we might need an additional x variable.

```
plot(x, resid(model))
abline(h=0, col="red")
```

6. Plot the residuals in time (sequence) order. We expect to see no particular trends in the data. If there are patterns in the plot, assess whether autocorrelation is present in the y variable (use the `acf(y)` function in R). If so, you might have to sub-sample the data, or resort to proper time-series analysis tools to fit your model.

```
plot(resid(model))
abline(h=0, col="red")
lines(lowess(resid(model), f=0.2))    # use a shorter smoothing span
```

7. Plot the residuals against the fitted-values. By definition of the least-squares model, the covariance between the residuals and the fitted values is zero. You can verify that $e^T \hat{y} = \sum_i^n e_i \hat{y}_i = 0$. A fan-shape to the residuals indicates the residual variance is not constant over the range of data: you will have to use weighted least squares to counteract that. It is better to use *studentized residuals* (page 194), rather than the actual residuals, since the actual residuals can show non-constant variance even though the errors have constant variance.

```
plot(predict(model), rstudent(model))
lines(lowess(predict(model), rstudent(model)))
abline(h=0, col="red")
```

8. Plot the predictions of y against the actual values of y. We expect the data to fall around a 45 degree line.

```
plot(y, predict(model))
lines(lowess(y, predict(model), f=0.5))     # a smoother
abline(a=0, b=1, col="red")                 # a 45 degree line
```

4.10 More than one variable: multiple linear regression (MLR)

Video for
this
section.

We now move to including more than one explanatory x variable in the linear model. We will:

1. introduce some matrix notation for this section

2. show how the optimization problem is solved to estimate the model parameters

3. how to interpret the model coefficients

4. extend our tools from the previous section to analyze the MLR model

5. use integer (yes/no *or* on/off) variables in our model.

First some motivating examples:

- A relationship exists between x_1 = reactant concentration and x_2 = temperature with respect to y = reaction rate. We already have a linear model between $y = b_0 + b_1 x_1$, but we want to improve our understanding of the system by learning about the temperature effect, x_2.

- We want to predict melt index in our reactor from the reactor temperature, but we know that the feed flow and pressure are also good explanatory variables for melt index. How do these additional variables improve the predictions?

- We know that the quality of our plastic product is a function of the mixing time, and also the mixing tank in which the raw materials are blended. How do we incorporate the concept of a mixing tank indicator in our model?

4.10.1 Multiple linear regression: notation

To help the discussion below it is useful to omit the least squares model's intercept term. We do this by first centering the data.

$$y_i = b_0 + b_1 x_i$$
$$\overline{y} = b_0 + b_1 \overline{x}$$
$$y_i - \overline{y} = 0 + b_1(x_i - \overline{x}) \qquad \text{by subtracting the previous lines from each other}$$

This indicates that if we fit a model where the x and y vectors are first mean-centered, i.e. let $x = x_{\text{original}} - \text{mean}\left(x_{\text{original}}\right)$ and $y = y_{\text{original}} - \text{mean}\left(y_{\text{original}}\right)$, then we still estimate the same slope for b_1, but the intercept term is zero. All we gain from this is simplification of the subsequent analysis. Of course, if you need to know what b_0 was, you can use the fact that $b_0 = \overline{y} - b_1 \overline{x}$. Nothing else changes: the $R^2, S_E, S_E(b_1)$ and all other model interpretations remain the same. You can easily prove this for yourself.

So in the rest of the this section we will omit the model's intercept term, since it can always be recovered afterwards.

The general linear model is given by:

$$y_i = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon_i$$

$$y_i = [x_1, x_2, \ldots, x_k] \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + \epsilon_i$$

$$y_i = \underbrace{x^T}_{(1 \times k)} \underbrace{\beta}_{(k \times 1)} + \epsilon_i$$

And writing the last equation n times over for each observation in the data:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,k} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \ldots & x_{n,k} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$$

where:

- **y**: $n \times 1$

- **X**: $n \times k$

- **b**: $n \times 1$

- **e**: $n \times 1$

4.10.2 Estimating the model parameters via optimization

As with the simple least squares model, $y = b_0 + b_1 x$, we aim to minimize the sum of squares of the errors in vector **e**. This least squares objective function can be written compactly as:

$$
\begin{aligned}
f(\mathbf{b}) &= \mathbf{e}^T \mathbf{e} \\
&= (\mathbf{y} - \mathbf{X}\mathbf{b})^T (\mathbf{y} - \mathbf{X}\mathbf{b}) \\
&= \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{X}\mathbf{b} + \mathbf{b}\mathbf{X}^T \mathbf{X}\mathbf{b}
\end{aligned}
$$

Taking partial derivatives with respect to the entries in **b** and setting the result equal to a vector of zeros, you can prove to yourself that $\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T\mathbf{y}$. You might find the Matrix Cookbook[79] useful in solving these equations and optimization problems.

Three important relationships are now noted:

1. $\mathcal{E}\{\mathbf{b}\} = \beta$

2. $\mathcal{V}\{\mathbf{b}\} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} S_E^2$

3. An estimate of the standard error is given by: $\sigma_e \approx S_E = \sqrt{\dfrac{\mathbf{e}^T \mathbf{e}}{n - k}}$, where k is the number of parameters estimated in the model and n is the number of observations.

These relationships imply that our estimates of the model parameters are unbiased (the first line), and that the variability of our parameters is related to the $\mathbf{X}^T\mathbf{X}$ matrix and the model's standard error, S_E.

Going back to the single variable case we showed in the section where we derived *confidence intervals* (page 168) for b_0 and b_1 that:

$$
\mathcal{V}\{b_1\} = \frac{S_E^2}{\sum_j \left(x_j - \overline{\mathrm{x}}\right)^2}
$$

Notice that our matrix definition, $\mathcal{V}\{\mathbf{b}\} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} S_E^2$, gives exactly the same result, remembering the x variables have already been centered in the matrix form. Also recall that the variability of these estimated parameters can be reduced by (a) taking more samples, thereby increasing the denominator size, and (b) by including observations further away from the center of the model.

---

[79] https://www.google.ca/search?q=The+Matrix+Cookbook/

Example

Let $x_1 = [1, 3, 4, 7, 9, 9]$, and $x_2 = [9, 9, 6, 3, 1, 2]$, and $y = [3, 5, 6, 8, 7, 10]$. By inspection, the x_1 and x_2 variables are negatively correlated, and the x_1 and y variables are positively correlated (also positive covariance). Refer to the definition of covariance in *an equation from the prior section* (page 153).

After mean centering the data we have that $x_1 = [-4.5, -2.5, -1.5, 1.5, 3.5, 3.5]$, and $x_2 = [4, 4, 1, -2, -4, -3]$ and $y = [-3.5, -1.5, -0.5, 1.5, 0.5, 3.5]$. So in matrix form:

$$\mathbf{X} = \begin{bmatrix} -4.5 & 4 \\ -2.5 & 4 \\ -1.5 & 1 \\ 1.5 & -2 \\ 3.5 & -4 \\ 3.5 & -3 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} -3.5 \\ -1.5 \\ -0.5 \\ 1.5 \\ 0.5 \\ 3.5 \end{bmatrix}$$

The $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{y}$ matrices can then be calculated as:

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 55.5 & -57.0 \\ -57.0 & 62 \end{bmatrix} \qquad \mathbf{X}^T\mathbf{y} = \begin{bmatrix} 36.5 \\ -36.0 \end{bmatrix}$$

Notice what these matrices imply (remembering that the vectors in the matrices have been centered). The $\mathbf{X}^T\mathbf{X}$ matrix is a scaled version of the covariance matrix of $\mathbf{X}$. The diagonal terms show how strongly the variable is correlated with itself, which is the variance, and always a positive number. The off-diagonal terms are symmetrical, and represent the strength of the relationship between, in this case, x_1 and x_2. The off-diagonal terms for two uncorrelated variables would be a number close to, or equal to zero.

The inverse of the $\mathbf{X}^T\mathbf{X}$ matrix is particularly important - it is related to the standard error for the model parameters - as in: $\mathcal{V}\{\mathbf{b}\} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} S_E^2$.

$$\left(\mathbf{X}^T\mathbf{X}\right)^{-1} = \begin{bmatrix} 0.323 & 0.297 \\ 0.297 & 0.289 \end{bmatrix}$$

The non-zero off-diagonal elements indicate that the variance of the b_1 coefficient is related to the variance of the b_2 coefficient as well. This result is true for most regression models, indicating we can't accurately interpret each regression coefficient's confidence interval on its own.

For the two variable case, $y = b_1 x_1 + b_2 x_2$, the general relationship is that:

$$\mathcal{V}(b_1) = \frac{1}{1 - r_{12}^2} \times \frac{S_E^2}{\sum x_1^2}$$

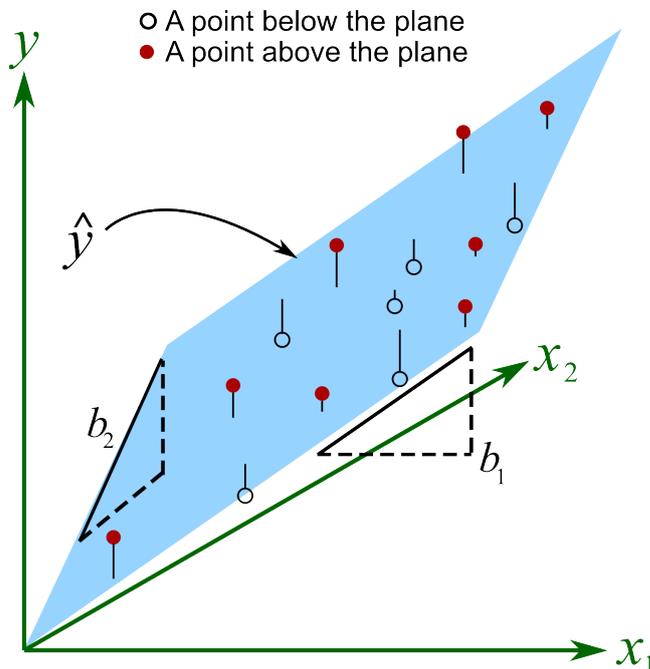$$\mathcal{V}(b_2) = \frac{1}{1 - r_{12}^2} \times \frac{S_E^2}{\sum x_2^2}$$

where r_{12}^2 represents the correlation between variable x_1 and x_2. What happens as the correlation between the two variables increases?

4.10.3 Interpretation of the model coefficients

Let's take a look at the case where $y = b_1 x_1 + b_2 x_2$. We can plot this on a 3D plot, with axes of x_1, x_2 and y:

The points are used to fit the plane by minimizing the sum of square distances shown by vertical lines from each point to the plane. The interpretation of the slope coefficients for b_1 and b_2 is **not the same** as for the case with just a single x variable.

When we have multiple x variables, then the value of coefficient b_1 is the average change we would expect in **y** for a one unit change in x_1 provided we hold x_2 fixed. It is the last part that is new: we must assume that other x variables are fixed.

For example, let $y = b_T T + b_S S = -0.52T + 3.2S$, where T is reactor temperature in Kelvin, and S is substrate concentration in g/L, and y is yield in μg, for a bioreactor reactor system. The $b_T = -0.52\mu$g/K coefficient is the decrease in yield for every 1 Kelvin increase in temperature, holding the substrate concentration fixed.

This is a good point to introduce some terminology you might come across. Imagine you have a model where y is the used vehicle price and x_1 is the mileage on the odometer (we expect that b_1 will be negative) and x_2 is the number of doors on the car. You might hear the phrase: "the effect of the number of doors, controlling for mileage, is not significant". The part "controlling for . . . " indicates that the controlled variable has been added to regression model, and its effect is accounted for. In other words, for two vehicles with the same mileage, the coefficient b_2 indicates whether the second hand price increases or decreases as the number of doors on the car changes (e.g. a 2-door vs a 4-door car).

In the prior example, we could say: the effect of substrate concentration on yield, controlling for temperature, is to increase the yield by 3.2 μg for every increase in 1 g/L of substrate concentration.

4.10.4 Integer (dummy, indicator) variables in the model

Now that we have introduced multiple linear regression to expand our models, we also consider these sort of cases:

- We want to predict yield, but want to indicate whether a radial or axial impeller was used in the reactor and learn whether it has any effect on yield.

- Is there an important difference when we add the catalyst first and then the reactants, or the reactants followed by the catalyst?

- Use an indicator variable to show if the raw material came from the supplier in Spain, India, or Vietnam and interpret the effect of supplier on yield.
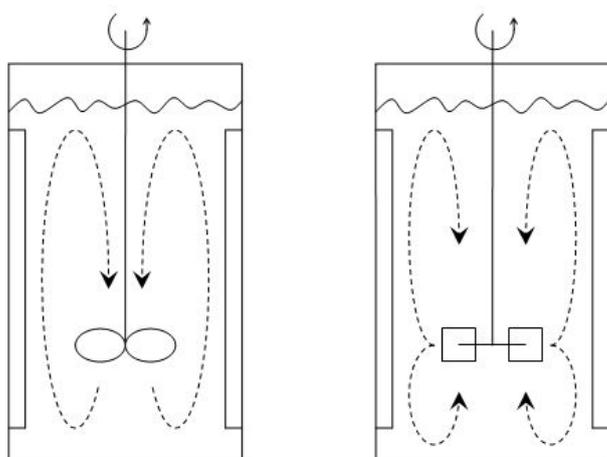
Axial and radial blades; figure from Wikipedia[80]

We will start with the simplest case, using the example of the radial or axial impeller. We wish to understand the effect on yield, $y[\mu g]$, as a function of the impeller type, and impeller speed, x.

$$y = \beta_0 + \beta_1 x + \gamma d + \varepsilon$$
$$y = b_0 + b_1 x + g d_i + e_i$$

where $d_i = 0$ if an axial impeller was used, or $d_i = 1$ if a radial impeller was used. All other least squares assumptions hold, particularly that the variance of y_i is unrelated to the value of d_i. For the initial discussion let's assume that $\beta_1 = 0$, then geometrically, what is happening here is:

---

[80] https://en.wikipedia.org/wiki/Impeller
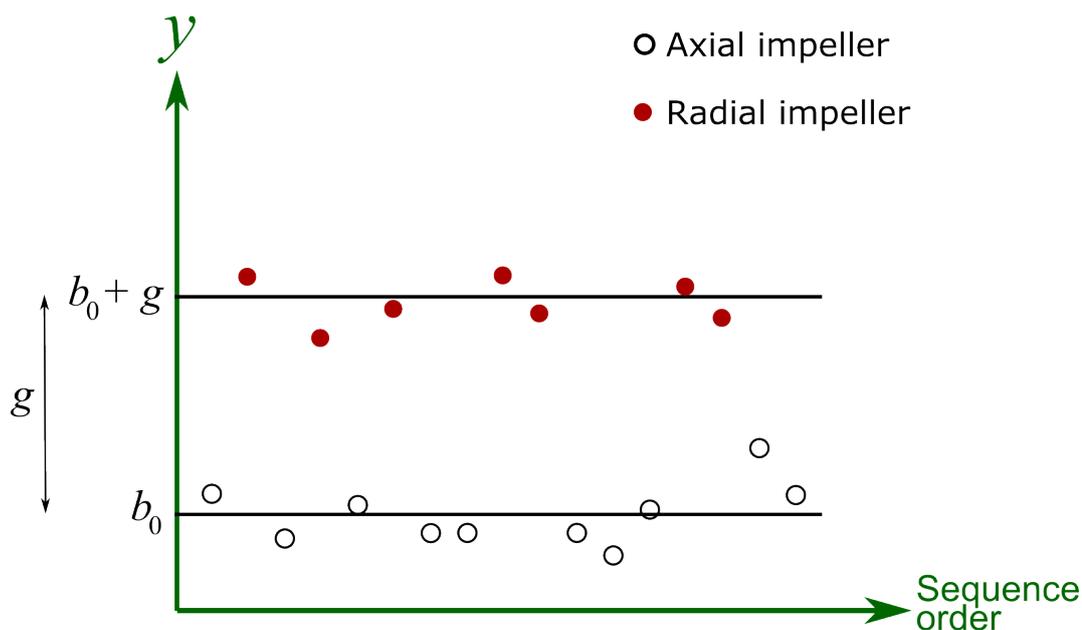
The γ parameter, estimated by g, is the difference in intercept when using a different impeller type. Note that the lines are parallel.

$$\text{Axial impellers:} \qquad y = b_0 + 0$$
$$\text{Radial impellers:} \qquad y = b_0 + g$$

Now if $\beta_1 \neq 0$, then the horizontal lines in the above figure are tilted, but still parallel to each other. Nothing else is new here, other than the representation of the variable used for d_i. The interpretation of its coefficient, g, is the same as with any other least squares coefficient. In this particular example, had $g = -56\mu$g, it would indicate that the average decrease in yield is 56 μg when using a radial impeller.

The rest of the analysis tools for least squares models can be used quite powerfully. For example, a 95% confidence interval for the impeller variable might have been:

$$-32\mu\text{g} \leq \gamma \leq 21\mu\text{g}$$

which would indicate the impeller type has no significant effect on the yield amount, the y-variable.

Integer variables are also called dummy variables or indicator variables. Really what is happening here is the same concept as for multiple linear regression, the equation of a plane is being estimated. We only use the equation of the plane at integer values of d, but mathematically the underlying plane is actually continuous.
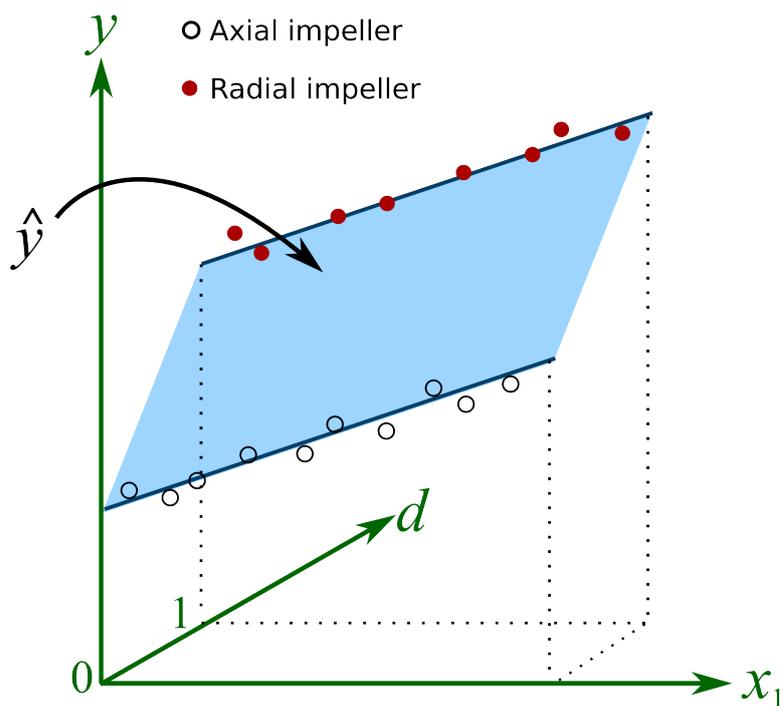
We have to introduce additional terms into the model if we have integer variables with more than 2 levels. In general, if there are p-levels, then we must include $p - 1$ terms. For example, if we wish to test the effect of y = yield achieved from the raw material supplier in Spain, India, or Vietnam, we could code:

- Spain: $d_{i1} = 0$ and $d_{i2} = 0$

- India: $d_{i1} = 1$ and $d_{i2} = 0$

- Vietnam: $d_{i1} = 0$ and $d_{i2} = 1$.

and solve for the least squares model: $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + \gamma_1 d_1 + \gamma_2 d_2 + \varepsilon$, where γ_1 is the effect of the Indian supplier, holding all other terms constant (i.e. it is the incremental effect of India relative to Spain); γ_2 is the incremental effect of the Vietnamese supplier relative to the base case of the Spanish supplier. Because of this somewhat confusing interpretation of the coefficients, sometimes people will assume they can sacrifice an extra degree of freedom, but introduce p new terms for the p levels of the integer variable, instead of $p - 1$ terms.

- Spain: $d_{i1} = 1$ and $d_{i2} = 0$ and $d_{i3} = 0$

- India: $d_{i1} = 0$ and $d_{i2} = 1$ and $d_{i3} = 0$

- Vietnam: $d_{i1} = 0$ and $d_{i2} = 0$ and $d_{i3} = 1$

and $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + \gamma_1 d_1 + \gamma_2 d_2 + \gamma_3 d_3 + \varepsilon$, where the coefficients γ_1, γ_2 and γ_3 are assumed to be more easily interpreted. However, calculating this model will fail, because there is a built-in perfect linear combination. The $\mathbf{X}^T \mathbf{X}$ matrix is not invertible.

4.11 Outliers: discrepancy, leverage, and influence of the observations

Unusual observations will influence the model parameters and also influence the analysis from the model (standard errors and confidence intervals). In this section we will examine how these outliers influence the model.

Outliers are in many cases the most interesting data in a data table. They indicate whether there was a problem with the data recording system, they indicate sometimes when the system is operating really well, though more likely, they occur when the system is operating under poor conditions. Nevertheless, outliers should be carefully studied for (a) why they occurred and (b) whether they should be retained in the model.

4.11.1 Background
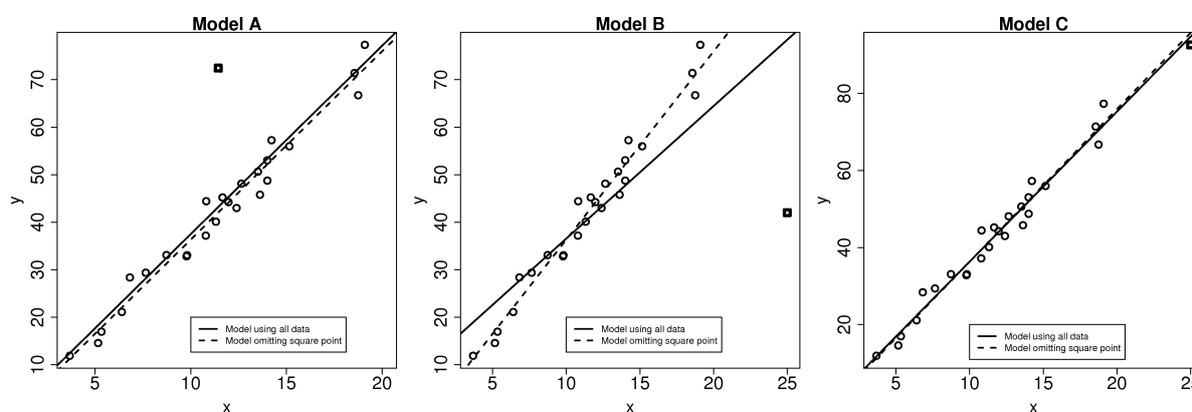
A discrepancy is a data point that is unusual *in the context of the least squares model*, as shown in the first figure here. On its own, from the perspective of either x or y alone, the square point is not unusual. But it is unusual in the context of the least squares model. When that square point is removed, the updated least squares line (dashed line) is obtained. This square point clearly has little influence on the model, even though it is discrepant.

The discrepant square point in model B has much more influence on the model. Given that the objective function aims to minimize the sum of squares of the deviations, it is not surprising that the slope is pulled towards this discrepant point. Removing that point gives a different dashed-line estimate of the slope and intercept.

In model C the square point is not discrepant in the context of the model. But it does have high leverage on the model: a small change in this point has the potential to be influential on the model.

Can we quantify how much *influence* these *discrepancies* have on the model; and what is *leverage*? The following general formula is helpful in the rest of this discussion:

$$\text{Leverage} \times \text{Discrepancy} = \text{Influence on the model}$$

4.11.2 Leverage

Leverage measures how much each observation contributes to the model's prediction of $\hat{y}_i$. It is also called the hat value, h_i, and simply measures how far away the data point is from the center of the model, but it takes the model's correlation into account:

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2} \qquad \text{and} \qquad \bar{h} = \frac{k}{n} \qquad \text{and} \qquad \frac{1}{n} \le h_i \le 1.0$$

The average hat value can be calculated theoretically. While it is common to plot lines at 2 and 3 times the average hat value, always plot your data and judge for yourself what a large leverage means. Also notice that smallest hat value is always positive and greater or equal to $1/n$, while the largest hat value possible is 1.0. Continuing the example of models A, B and C: the hat values for models B and C are the same, and are shown here. The last point has very high leverage.
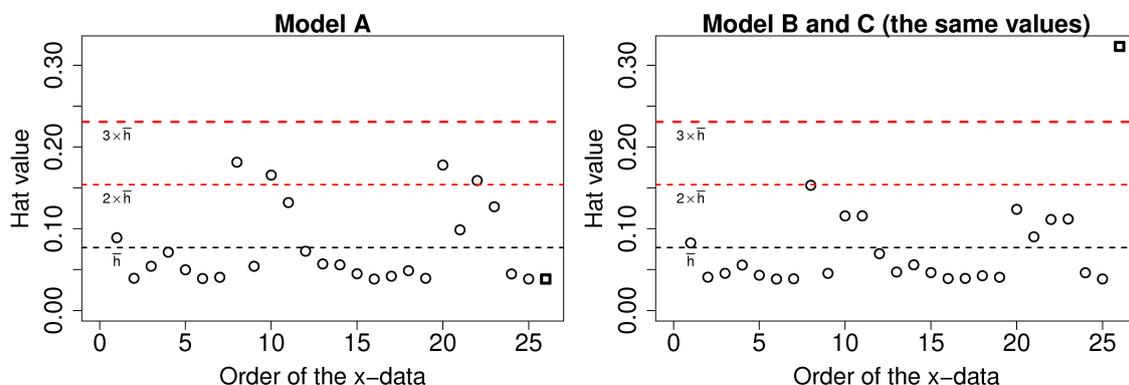
4.11.3 Discrepancy

Discrepancy can be measured by the residual distance. However the residual is not a complete measure of discrepancy. We can imagine cases where the point has such high leverage that it drags the entire model towards it, leaving it only with a small residual. One way then to isolate these points is to divide the residual by $1 - \text{leverage} = 1 - h_i$. So we introduce a new way to quantify the residuals here, called *studentized residuals*:

$$e_i^* = \frac{e_i}{S_{E(-i)}\sqrt{1 - h_i}}$$

Where e_i is the residual for the ith point, as usual, but $S_{E(-i)}$ is the standard error of the model when deleting the ith point and refitting the model. This studentized residual accounts for the fact that high leverage observations pull the model towards themselves. In practice the model is not recalculated by omitting each point one at a time, rather there are shortcut formula that implement this efficiently. Use the `rstudent(lm(y~x))` function in R to compute the studentized residuals from a given model.
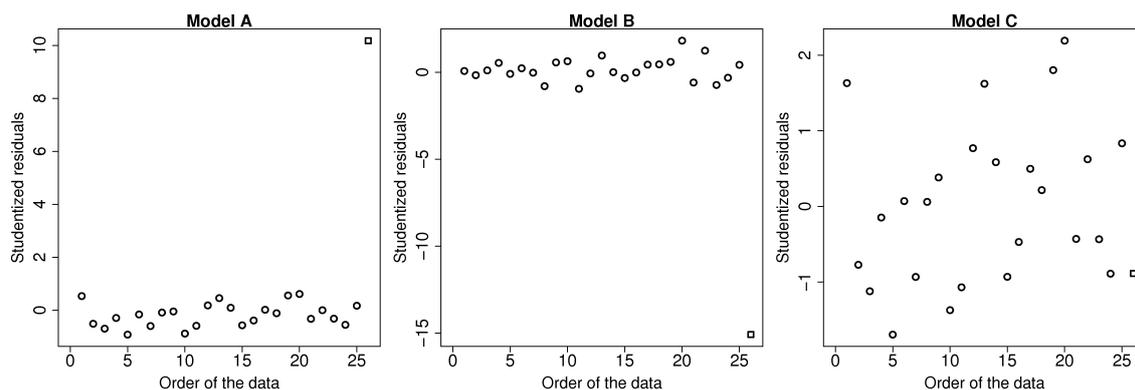
This figure illustrates how the square point in model A and B is highly discrepant, while in model C it does not have a high discrepancy.

4.11.4 Influence

The influence of each data point can be quantified by seeing how much the model changes when we omit that data point. The influence of a point is a combination its leverage and its discrepancy. In model A, the square point had large discrepancy but low leverage, so its influence on the model parameters (slope and intercept) was small. For model C, the square point had high leverage, but low discrepancy, so again the change in the slope and intercept of the model was small. However model B had both large discrepancy and high leverage, so its influence is large.

One measure is called *Cook's statistic*, usually called D_i, and often referred to just as *Cook's D*. Conceptually, it can be viewed as the change in the model coefficients when omitting an observation, however it is much more convenient to calculate it as follows:

$$D_i = \frac{e_i^2}{k \times \frac{1}{n} \sum e_i^2} \times \frac{h_i}{1 - h_i}$$

where $\frac{1}{n} \sum e_i^2$ is called the mean square error of the model (the average square error). It is easy to see here now why influence is the product of discrepancy and leverage.

The values of D_i are conveniently calculated in R using the `cooks.distance(model)` function. The results for the 3 models are shown. Interestingly for model C there is a point with even higher influence than the square point. Can you locate that point in the least squares plot?
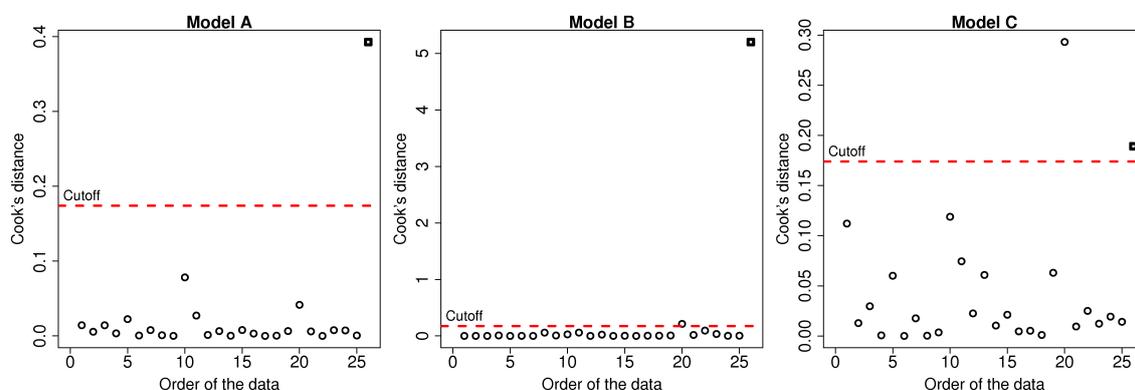
4.12 Enrichment topics

These topics are not covered in depth in this book, but might be of interest to you. I provide a small introduction to each topic, showing what their purpose is, together with some examples.
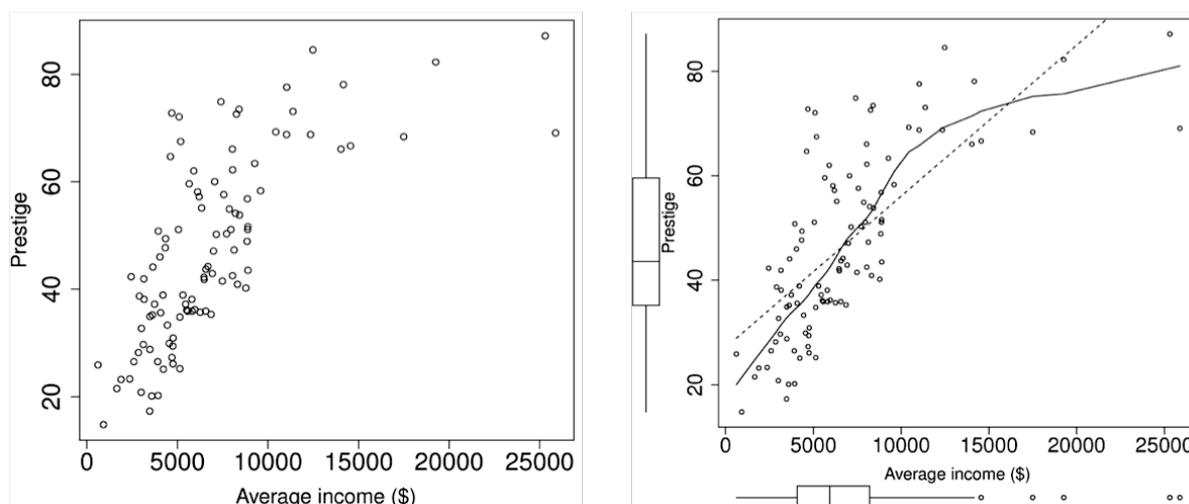
4.12.1 Nonparametric models

Nonparametric modelling is a general model where the relationship between x and y is of the form: $y = f(x) + \varepsilon$, but the function $f(x)$, i.e. the model, is left unspecified. The model is usually a smooth function.

Consider the example of plotting Prestige (the Pineo-Porter prestige[81] score) against Income, from the 1971 Canadian census. A snippet of the data is given by:

```
                      education income women prestige census type
ECONOMISTS                14.44   8049 57.31     62.2   2311 prof
VOCATIONAL.COUNSELLORS    15.22   9593 34.89     58.3   2391 prof
PHYSICIANS                15.96  25308 10.56     87.2   3111 prof
NURSING.AIDES              9.45   3485 76.14     34.9   3135   bc
POSTAL.CLERKS             10.07   3739 52.27     37.2   4173   wc
TRAVEL.CLERKS             11.43   6259 39.17     35.7   4193   wc
BABYSITTERS                9.46    611 96.53     25.9   6147 <NA>
BAKERS                     7.54   4199 33.30     38.9   8213   bc
MASONS                     6.60   5959  0.52     36.2   8782   bc
HOUSE.PAINTERS             7.81   4549  2.46     29.9   8785   bc
```

The plot on the left is the raw data, while on the right is the raw data with the nonparametric model (line) superimposed. The smoothed line is the nonparametric function, $f(x)$, referred to above, and $x =$ Income (\$), and $y =$ Prestige.

For bivariate cases, the nonparametric model is often called a *scatterplot smoother*. There are several methods to calculate the model; one way is by locally weighted scatterplot smoother (LOESS), described as follows. Inside a fixed subregion along the x-axis (called the window):

- collect the x- and y-values inside this window

- calculate a fitted y-value, but use a weighted least squares procedure, with weights that peak at the center of the window and declines towards the edges,

---

[81] https://en.wikipedia.org/wiki/John_Porter_(sociologist)

- record that average y-value against the window's center (x-value)

- slide the window along the x axis and repeat

The *model* is the collection of these x- and y-values. This is why it is called nonparameteric: there are no parameters to quantify the model. For example: if the relationship between the two variables is linear, then a linear smooth is achieved. It is hard to express the relationship between x and y in written form, so usually these models are shown visually. The nonparametric model is not immune to outliers, but it is resistant to them.

More details can be found in W.S. Cleveland, Robust Locally Weighted Regression and Smoothing Scatterplots[82], *Journal of the American Statistical Association*, **74** (368), p. 829-836, 1979.

4.12.2 Robust least squares models

Outliers are often the most interesting observations and are usually the points from which we learn the most about the system. A manual step where we review the outliers and their influence should always done for any important model. For example, inspection of the residual plots as described in the preceding sections.

However, the ability to build a linear model that is not heavily influenced by outliers might be of interest in certain cases.

- The model is built automatically and is not reviewed by a human (e.g. as an intermediate step in a data-mining procedure). This is increasingly common in systems that build on top of the least squares model to improve their performance in some way.

- The human reviewer is not skilled to know which plots to inspect for influential and discrepant observations, or may not know how to interpret these plots.

Some criticism of robust methods are that there are too many different robust methods and that these routines are much more computationally expensive than ordinary least squares. The first point is true, as this as a rapidly evolving field, however the latter objection is not of too much concern these days. Robust methods are now available in most decent software packages, and are stabilizing towards a few reliable robust estimators.

If you would like to read up some more, a nice introduction targeted at engineering readers is given in PJ Rousseeuw's "Tutorial to Robust Statistics[83]", *Journal of Chemometrics*, **5**, 1-20, 1991.

In R the various efforts of international researchers is being consolidated. The `robustbase` package provides basic functionality that is now well established in the field; use that package if you want to assemble various robust tools yourself. On the other hand, a more comprehensive package called `robust` is also available which provides robust tools that you should use if you are not too concerned with the details of implementation.

For example:

```
> data <- read.csv('http://openmv.net/file/distillation-tower.csv')

# Using ordinary least squares
# ----------------------------
> summary(lm(data$VapourPressure ~ data$TempC2))
```

(continues on next page)

---

[82] https://www.jstor.org/stable/2286407
[83] https://dx.doi.org/10.1002/cem.1180050103

```
Call:
lm(formula = data$VapourPressure ~ data$TempC2)

Residuals:
    Min      1Q   Median      3Q     Max
-5.59621 -2.37597  0.06674  2.00212 14.18660

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 195.96141    4.87669   40.18   <2e-16 ***
data$TempC2  -0.33133    0.01013  -32.69   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.989 on 251 degrees of freedom
Multiple R-squared: 0.8098,     Adjusted R-squared: 0.8091
F-statistic:  1069 on 1 and 251 DF,  p-value: < 2.2e-16

# Use robust least squares (with automatic selection of robust method)
# -----------------------------------------------------------------
> library(robust)
> summary(lmRob(data$VapourPressure ~ data$TempC2))

Call: lmRob(formula = data$VapourPressure ~ data$TempC2)

Residuals:
      Min         1Q    Median        3Q       Max
-5.2631296 -1.9805384 0.1677174 2.1565730 15.8846460

Coefficients:
             Value       Std. Error  t value      Pr(>|t|)
(Intercept) 179.48579886  4.92870640  36.41641120  0.00000000
data$TempC2  -0.29776778  0.01021412 -29.15256677  0.00000000

Residual standard error: 2.791 on 251 degrees of freedom
Multiple R-Squared: 0.636099

Test for Bias:
            statistic    p-value
M-estimate   7.962583 0.018661525
LS-estimate 12.336592 0.002094802
```

In this example the two models perform similarly in terms on their S_E, b_0 and b_1 values, as well as confidence intervals for them.

4.12.3 Logistic modelling (regression)

There are many practical cases in engineering modelling where our y-variable is a discrete entity. The most common case is pass or failure, naturally coded as $y = 0$ for failure, and $y = 1$ is coded as success. Some examples:

- Predict whether our product specifications are achieved ($y = 0$ or 1) given the batch reaction's temperature as x_1, the reaction duration x_2 and the reactor vessel, where $x_3 = 0$ for reactor A and $x_3 = 1$ for reactor B.

- Predict the likelihood of making a sale in your store ($y = 0$ or 1), given the customer's age x_1, whether they are a new or existing customers, x_2 is either 0 or 1, and the day of the week as x_3.

- Predict if the final product will be y = acceptable, medium, or unsellable based on the raw material's properties x_1, x_2, x_3 and the ambient temperature x_4.

We could naively assume that we just code our y variable as 0 or 1 (pass/fail) and build our least squares model as usual, using the x variables. While a seemingly plausible approach, the problems are that:

- The predictions when using the model are not dichotomous (0 or 1), which is not too much of a problem if we interpret our prediction more as a probability. That is, our prediction is the probability of success or failure, according to how we coded it originally. However the predictions often lie outside the range $[0, 1]$. We can attempt to compensate for this by clamping the output to zero or one, but this non-linearity causes instability in the estimation algorithms.

- The errors are not normally distributed.

- The variance of the errors are not constant and the assumption of linearity breaks down.
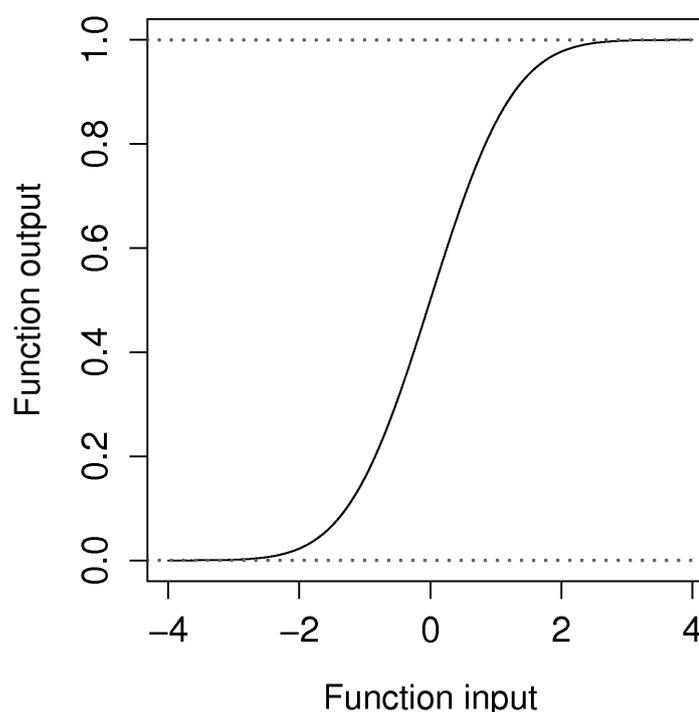
A logistic model however accounts for the nature of the y-variable by creating a function, called a logistic function, which is bounded between 0 and 1. In fact you are already familiar with such a function: the cumulative probability of the normal distribution does exactly this.

Once the y data are appropriately transformed, then the model can be calculated. In R one uses the `glm(y ~ x1 + x2, family=binomial)` function to build a model where `y` must be a factor variable: type `help(factor)` to learn more. The model output is interpreted as any other.

4.12.4 Testing of least-squares models

Before launching into this concept, first step back and understand why we are building least squares models. One objective is to learn more about our systems: (a) what is the effect of one variable on another, or (b) is the effect significant (examine the confidence interval). Another objective is purely predictive: build a model so that we can use it to make predictions. For this last case we must test our model's capability for accurate predictions.

The gold standard is always to have a testing data set available to quantify how good (adequate) your least squares model is. It is important that (a) the test set has no influence on the calculation of the

model parameters, and (b) is representative of how the model will be used in the future. We will illustrate this with 2 examples: you need to build a predictive model for product viscosity from 3 variables on your process. You have data available, once per day, for 2006 and 2007 (730 observations).

- Use observation 1, 3, 5, 7, . . . 729 to build the least squares model; then use observation 2, 4, 6, 8, . . . 730 to test the model.

- Use observations 1 to 365 (data from 2006) to build the model, and then use observations 366 to 730 (data from 2007) to test the model.

In both cases, the testing data has no influence on the model parameters. However the first case is not representative of how the model will be used in the future. The results from the first case are likely to give over-optimistic results, while the second case represents the intended use of the model more closely, and will have more honest results. Find out sooner, rather than later, that the model's long-term performance is not what you expect. It may be that you have to keep rebuilding the model every 3 months, updating the model with the most recent data, in order to maintain it's predictive performance.

How do we quantify this predictive performance? A common way is to calculate the root mean square of the prediction error (RMSEP), this is very similar to the *standard error* (page 164) that we saw earlier for regression models. Assuming the errors are centered at zero and follow a normal distribution, the RMSEP can be interpreted as the standard deviation of the prediction residuals. It is important the RMSEP be calculated only from new, unseen testing data. By contrast, you might see the term RMSEE (root mean square error of estimation), which is the RMSEP, but calculated from the training (model-building) data. The RMSEE $\approx S_E$ = standard error; the small difference being due to the denominator used (n versus $n - k$).

$$\text{RMSEP} = \sqrt{\frac{1}{n} \sum_i^n \left(y_{\text{new},i} - \hat{y}_{\text{new},i}\right)^2}$$

The units of RMSEP and RMSEE are the same as the units of the y-variable.

In the *latent variable modelling* (page 315) section of the book we will introduce the concept of cross-validation to test a model. Cross-validation uses the model training data to simulate the testing process. So it is not as desirable as having a fresh testing data set, but it works well in many cases. Cross-validation can be equally well applied to least squares models. We will revisit this topic later.

4.12.5 Bootstrapping

Bootstrapping is an extremely useful tool when theoretical techniques to estimate confidence intervals and uncertainty are not available to us.

Let's give an example where bootstrapping is strictly not required, but is definitely useful. When fitting a least squares model of the form $y = \beta_0 + \beta_1 x$ we are interested in the confidence interval of the slope coefficient, β_1. Recall this coefficient indicates by how much the y-variable changes on average when changing the x variable by one unit. The slope coefficient might represent a rate constant, or be related to the magnitude of the feedback control loop gain. Whatever the case, it is important we understand the degree of uncertainty associated with it, so we can make an appropriate judgement.

In the preceding section on least squares model analysis we *derived this confidence interval* (page 169) for β_1, repeated here:

$$
\begin{array}{ccccc}
-c_t & \leq & \dfrac{b_1 - \beta_1}{S_E(b_1)} & \leq & +c_t \\
b_1 - c_t S_E(b_1) & \leq & \beta_1 & \leq & b_1 + c_t S_E(b_1)
\end{array}
$$

Visualize this confidence in the context of the following example where x is the dose of radiation administered (rads), and y is the survival percentage. The plot shows the data and the least square slope coefficient (notice the y variable is a transformed variable, `log(survival)`).

The thick line represents the slope coefficient (-0.0059) using all the data. Clearly the unusual point number 13 has some influence on that coefficient. Eliminating it and refitting the model makes the slope coefficient more steep (-0.0078), which could change our interpretation of the model. This raises the question though: what happens to the slope coefficient when we eliminate other points in the training data? How sensitive are our model parameters *to the data themselves*?
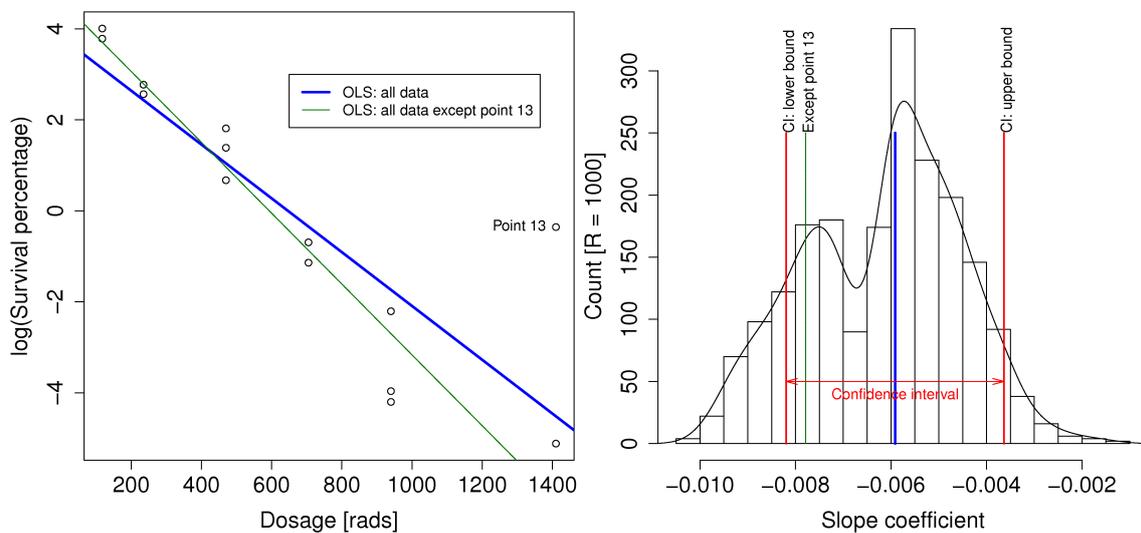
Bootstrapping gives us an indication of that sensitivity, as shown in the other plot. The original data set had 14 observations. What bootstrapping does is to randomly select 14 rows from the original data, allowing for duplicate selection. These selected rows are used to build a least squares model, and the slope coefficient is recorded. Then another 14 random rows are selected and this process is repeated `R` times (in this case `R=1000`). On some of these occasions the outlier points will be included, and other times they will be excluded.

A histogram of the 1000 computed slope coefficients is shown here. This histogram gives us an additional indication of the uncertainty of the slope coefficient. It shows many possible slope coefficients that could have been obtained. One in particular has been marked, the slope when point 13 was omitted.

For completeness the confidence interval at the 95% level for β_1 is calculated here, and also superimposed on the histogram.

$$
\begin{array}{ccccc}
-c_t & \leq & \dfrac{b_1 - \beta_1}{S_E(b_1)} & \leq & +c_t \\
-0.005915 - 2.1788 \times 0.001047 & \leq & \beta_1 & \leq & -0.005915 + 2.1788 \times 0.001047 \\
-0.0082 & \leq & \beta_1 & \leq & -0.0036
\end{array}
$$

This confidence interval, together with the bootstrapped values of b_1 give us additional insight when when making our interpretation of b_1.

By now you should also be wondering whether you can bootstrap the confidence interval bounds! That's left as exercise for interested readers. The above example was inspired from an example in ASA Statistics Computing and Graphics[84], **13** (1), 2002.

4.13 Exercises

Question 1

Use the distillation column data set[85] and choose any two variables, one for x and one as y. Then fit the following models by least squares in any software package you prefer:

- $y_i = b_0 + b_1 x_i$

- $y_i = b_0 + b_1(x_i - \bar{x})$ (what does the b_0 coefficient represent in this case?)

- $(y_i - \bar{y}) = b_0 + b_1(x_i - \bar{x})$

Prove to yourself that centering the x and y variables gives the same model for the 3 cases in terms of the b_1 slope coefficient, standard errors and other model outputs.

Solution

Once you have created an x and y variable in R, compare the output from these 3 models:

```
# Model 1
summary(lm(y ~ x))

# Model 2
x.mc <- x - mean(x)
summary(lm(y ~ x.mc))

# Model 3
y.mc <- y - mean(y)
summary(lm(y.mc ~ x.mc))
```

Question 2

For a x_{new} value and the linear model $y = b_0 + b_1 x$ the prediction interval for $\hat{y}_{\text{new}}$ is:

$$\hat{y}_i \pm c_t \sqrt{V\{\hat{y}_i\}}$$

where c_t is the critical t-value, for example at the 95% confidence level.

Use the distillation column data set[86] and with y as `VapourPressure` (units are kPa) and x as `TempC2` (units of degrees Farenheit) fit a linear model. Calculate the prediction interval for vapour pressure at these 3 temperatures: 430, 480, 520 °F.

Solution

[84] http://stat-computing.org/newsletter/
[85] http://openmv.net/info/distillation-tower
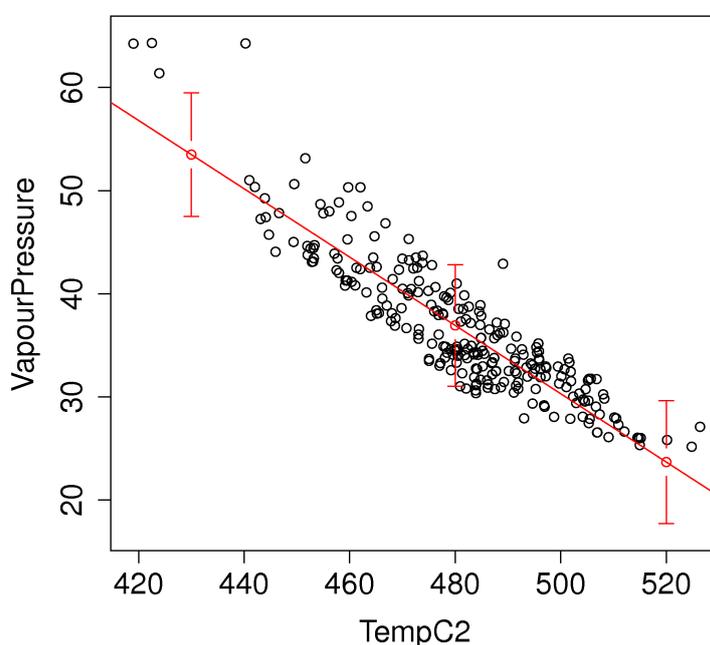[86] http://openmv.net/info/distillation-tower

The prediction interval is dependent on the value of $x_{\text{new, i}}$ used to make the prediction. For this model, $S_E = 2.989$ kPa, $n = 253$, $\sum_j (x_j - \overline{x})^2 = 86999.6$, and $\overline{x} = 480.82$.

$$\mathcal{V}(\hat{y}_{\text{new,i}}) = S_E^2 \left(1 + \frac{1}{n} + \frac{(x_{\text{new}} - \overline{x})^2}{\sum_j (x_j - \overline{x})^2} \right)$$

Calculating this term manually, or using the `predict(model, newdata=..., int="p")` function in R gives the 95% prediction interval:

- $x_{\text{new}} = 430$ °F: $\hat{y}_{\text{new}} = 53.49 \pm 11.97$, or [47.50, 59.47]

- $x_{\text{new}} = 480$ °F: $\hat{y}_{\text{new}} = 36.92 \pm 11.80$, or [31.02, 42.82]

- $x_{\text{new}} = 520$ °F: $\hat{y}_{\text{new}} = 23.67 \pm 11.90$, or [17.72, 29.62]

Visualizing the prediction intervals


```
dist <- read.csv('http://openmv.net/file/distillation-tower.csv')
attach(dist)
model <- lm(VapourPressure ~ TempC2)
summary(model)

# From the above output
SE = sqrt(sum(resid(model)^2)/model$df.residual)
n = length(TempC2)
k = model$rank
x.new = data.frame(TempC2 = c(430, 480, 520))
x.bar = mean(TempC2)
x.variance = sum((TempC2-x.bar)^2)
var.y.hat = SE^2 * (1 + 1/n + (x.new-x.bar)^2/x.variance)
c.t = -qt(0.025, df=n-k)
y.hat = predict(model, newdata=x.new, int="p")
PI.LB = y.hat[,1] - c.t*sqrt(var.y.hat)
PI.UB = y.hat[,1] + c.t*sqrt(var.y.hat)

# Results from y.hat agree with PI.LB and PI.UB
y.hat
#        fit      lwr      upr
# 1 53.48817 47.50256 59.47379
# 2 36.92152 31.02247 42.82057
```

(continues on next page)

```
# 3 23.66819 17.71756 29.61883
y.hat[,3] - y.hat[,2]
plot(TempC2, VapourPressure, ylim = c(17, 65), main="Visualizing the prediction intervals")
abline(model, col="red")
library(gplots)
plotCI(x=c(430, 480, 520), y=y.hat[,1], li=y.hat[,2], ui=y.hat[,3], add=TRUE, col="red")
```

Question 3

Refit the distillation model from the previous question with a transformed temperature variable. Use $1/T$ instead of the actual temperature.

- Does the model fit improve?

- Are the residuals more normally distributed with the untransformed or transformed temperature variable?

- How do you interpret the slope coefficient for the transformed temperature variable?

- Use the model to compute the predicted vapour pressure at a temperature of 480 °F, and also calculate the corresponding prediction interval at that new temperature.

Solution

- Using the `model.inv <- lm(VapourPressure ~ I(1/TempC2))` instruction, one obtains the model summary below. The model fit has improved slightly: the standard error is 2.88 kPa, reduced from 2.99 kPa.

```
Call:
lm(formula = VapourPressure ~ I(1/TempC2))

Residuals:
     Min       1Q   Median       3Q      Max
-5.35815 -2.27855 -0.08518  1.95057 13.38436

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -120.760      4.604  -26.23   <2e-16 ***
I(1/TempC2)  75571.306   2208.631   34.22   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.88 on 251 degrees of freedom
Multiple R-squared: 0.8235,     Adjusted R-squared: 0.8228
F-statistic:  1171 on 1 and 251 DF,  p-value: < 2.2e-16
```

- The residuals have roughly the same distribution as before, maybe a little more normal on the left tail, but hardly noticeable.

Model with temperature / Model with inverted temperature
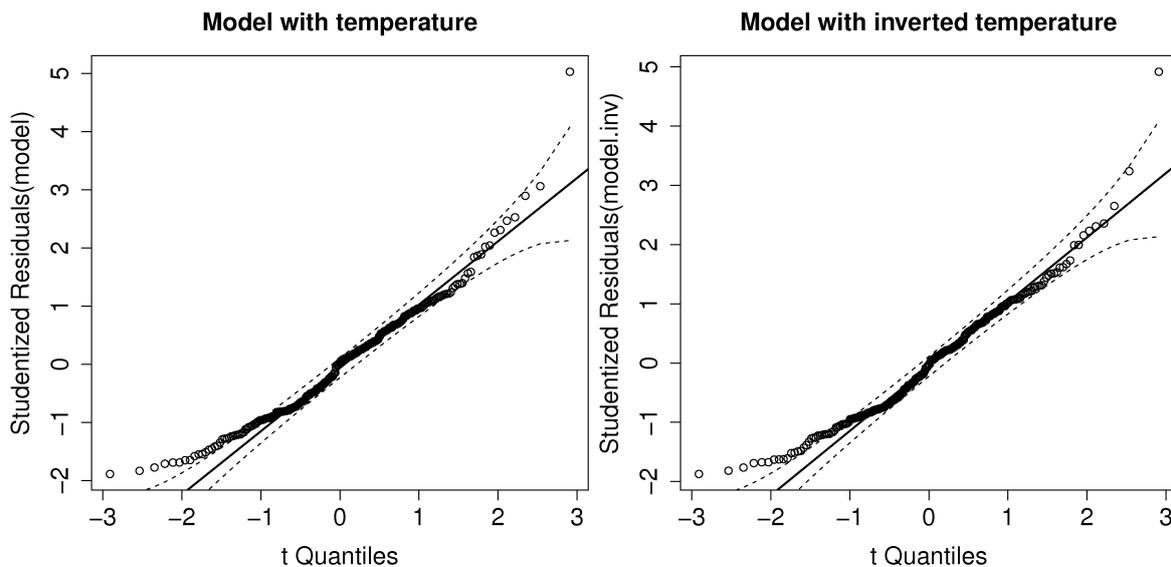
- The slope coefficient of 75571 has units of `kPa.°F`, indicating that each one unit *decrease* in temperature results in an *increase* in vapour pressure. Since division is not additive, the change in vapour pressure when decreasing 10 d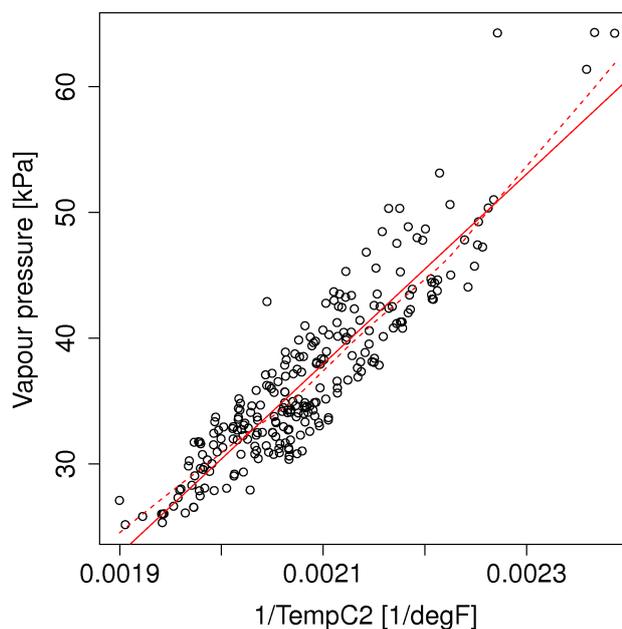egrees from 430 °F is a different decrease to that when temperature is 530 °F. The interpretation of transformed variables in linear models is often a lot harder. The easiest interpretation is to show a plot of 1/T against vapour pressure.

- The predicted vapour pressure at 480 °F is 36.68 kPa ±11.37, or within the range [31.0 to 42.4] with 95% confidence, very similar to the prediction interval from question 2.

```
# Model with inverted temperature
model.inv <- lm(VapourPressure ~ I(1/TempC2))
summary(model.inv)

plot(1/TempC2, VapourPressure, xlab="1/TempC2 [1/degF]", ylab="Vapour pressure [kPa]")
abline(model.inv, col="red")
lines(lowess(1/TempC2, VapourPressure), lty=2, col="red")

x.new = data.frame(TempC2 = c(430, 480, 520))
```

```
y.hat = predict(model.inv, newdata=x.new, int="p")
y.hat
#        fit       lwr       upr
# 1 54.98678 49.20604 60.76751
# 2 36.67978 30.99621 42.36334
# 3 24.56899 18.84305 30.29493

layout(matrix(c(1,2), 1, 2))
library(car)
qqPlot(model, main="Model with temperature", col=c(1, 1))
qqPlot(model.inv, main="Model with inverted temperature",col=c(1, 1))
```

Question 4

Again, for the distillation model, use the data from 2000 and 2001 to build the model (the first column in the data set contains the dates). Then use the remaining data to test the model. Use x = TempC2 and y = VapourPressure in your model.

- Calculate the RMSEP for the testing data. How does it compare to the standard error from the model?

- Now use the influencePlot(...) function from the car library, to highlight the influential observations in the model building data (2000 and 2001). Show your plot with observation labels (observation numbers are OK). See part 5 of the R tutorial[87] for some help.

- Explain how the points you selected are influential on the model?

- Remove these influential points, and refit the model on the training data. How has the model's slope and standard error changed?

- Recalculate the RMSEP for the testing data; how has it changed?

Short answer: RMSEP = 4.18 kPa; standard error = 2.68 kPa.

Question 5

The Kappa number data set[88] was used in an *earlier question* (page 144) to construct a Shewhart chart. The *"Mistakes to avoid"* (page 120) section (Process Monitoring), warns that the subgroups for a Shewhart chart must be independent to satisfy the assumptions used to derived the Shewhart limits. If the subgroups are not independent, then it will increase the type I (false alarm) rate.

This is no different to the independence required for least squares models. Use the autocorrelation tool to determine a subgroup size for the Kappa variable that will satisfy the Shewhart chart assumptions. Show your autocorrelation plot and interpret it as well.

---

[87] https://learnche.org/4C3/Software_tutorial
[88] http://openmv.net/info/kappa-number

Question 6

You presume the yield from your lab-scale bioreactor, y, is a function of reactor temperature, batch duration, impeller speed and reactor type (one with with baffles and one without). You have collected these data from various experiments.

Temp = T [℃]	Duration = d [minutes]	Speed = s [RPM]	Baffles = b [Yes/No]	Yield = y [g]
82	260	4300	No	51
90	260	3700	Yes	30
88	260	4200	Yes	40
86	260	3300	Yes	28
80	260	4300	No	49
78	260	4300	Yes	49
82	260	3900	Yes	44
83	260	4300	No	59
64	260	4300	No	60
73	260	4400	No	59
60	260	4400	No	57
60	260	4400	No	62
101	260	4400	No	42
92	260	4900	Yes	38

- Use software to fit a linear model that predicts the yield from these variables (the data set is available from the website[89]). See the R tutorial[90] for building linear models with integer variables in R.

- Interpret the meaning of each effect in the model. If you are using R, then the `confint(...)` function will be helpful as well. Show plots of each x variable in the model against yield. Use a box plot for the baffles indicator variable.

- Now calculate the $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{y}$ matrices; include a column in the $\mathbf{X}$ matrix for the intercept. Since you haven't mean centered the data to create these matrices, it would be misleading to try interpret them.

- Calculate the least squares model estimates from these two matrices. See the R tutorial[91] for doing matrix operations in R, but you might prefer to use MATLAB for this step. Either way, you should get the same answer here as in the first part of this question.

[89] http://openmv.net/info/bioreactor-yields
[90] https://learnche.org/4C3/Software_tutorial
[91] https://learnche.org/4C3/Software_tutorial

Question 7

In the section on comparing differences between two groups we used, without proof, the fact that:

$$\mathcal{V}\{\overline{x}_B - \overline{x}_A\} = \mathcal{V}\{\overline{x}_B\} + \mathcal{V}\{\overline{x}_A\}$$

Prove this statement, and clearly explain all steps in your proof.

Question 8

The production of low density polyethylene is carried out in long, thin pipes at high temperature and pressure (1.5 kilometres long, 50mm in diameter, 500 K, 2500 atmospheres). One quality measurement of the LDPE is its melt index. Laboratory measurements of the melt index can take between 2 to 4 hours. Being able to predict this melt index, in real time, allows for faster adjustment to process upsets, reducing the product's variability. There are many variables that are predictive of the melt index, but in this example we only use a temperature measurement that is measured along the reactor's length.

These are the data of temperature (K) and melt index (units of melt index are "grams per 10 minutes").

Temperature = T [Kelvin]	441	453	461	470	478	481	483	485	499	500	506	516
Melt index = m [g per 10 mins]	9.3	6.6	6.6	7.0	6.1	3.5	2.2	3.6	2.9	3.6	4.2	3.5

The following calculations have already been performed:

- Number of samples, $n = 12$

- Average temperature = $\overline{T} = 481$ K

- Average melt index, $\overline{m} = 4.925$ g per 10 minutes.

- The summed product, $\sum_i \left(T_i - \overline{T}\right)\left(m_i - \overline{m}\right) = -422.1$

- The sum of squares, $\sum_i \left(T_i - \overline{T}\right)^2 = 5469.0$

1. Use this information to build a predictive linear model for melt index from the reactor temperature.

2. What is the model's standard error and how do you interpret it in the context of this model? You might find the following software software output helpful, but it is not required to answer the question.

```
Call:
lm(formula = Melt.Index ~ Temperature)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5771 -0.7372  0.1300  1.2035  1.2811

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) --------    8.60936   4.885 0.000637
Temperature --------    0.01788  -4.317 0.001519

Residual standard error: 1.322 on 10 degrees of freedom
Multiple R-squared: 0.6508,     Adjusted R-squared: 0.6159

F-statistic: 18.64 on 1 and 10 DF,  p-value: 0.001519
```

3. Quote a confidence interval for the slope coefficient in the model and describe what it means. Again, you may use the above software output to help answer your question.

Short answer: m = 42.0 - 0.0772 T

Question 9

For a distillation column, it is well known that the column temperature directly influences the purity of the product, and this is used in fact for feedback control, to achieve the desired product purity. Use the distillation data set[92] , and build a least squares model that predicts `VapourPressure` from the temperature measurement, `TempC2`. Report the following values:

1. the slope coefficient, and describe what it means in terms of your objective to control the process with a feedback loop

2. the interquartile range and median of the model's residuals

3. the model's standard error

4. a confidence interval for the slope coefficient, and its interpretation.

You may use any computer package to build the model and read these values off the computer output.

Question 10

Use the bioreactor data[93], which shows the percentage yield from the reactor when running various experiments where temperature was varied, impeller speed and the presence/absence of baffles were adjusted.

1. Build a linear model that uses the reactor temperature to predict the yield. Interpret the slope and intercept term.

2. Build a linear model that uses the impeller speed to predict yield. Interpret the slope and intercept term.

3. Build a linear model that uses the presence (represent it as 1) or absence (represent it as 0) of baffles to predict yield. Interpret the slope and intercept term.

   Note: if you use R it will automatically convert the `baffles` variable to 1's and 0's for you. If you wanted to make the conversion yourself, to verify what R does behind the scenes, try this:

   ```
   # Read in the data frame
   bio <- read.csv('http://openmv.net/file/bioreactor-yields.csv')

   # Force the baffles variables to 0's and 1's
   bio$baffles <- as.numeric(bio$baffles) - 1
   ```

4. Which variable(s) would you change to boost the batch yield, at the lowest cost of implementation?

5. Use the `plot(bio)` function in R, where `bio` is the data frame you loaded using the `read.csv(...)` function. R notices that `bio` is not a single variable, but a group of variables, i.e. a

---

[92] http://openmv.net/info/distillation-tower

[93] http://openmv.net/info/bioreactor-yields

data frame, so it plots what is called a *scatterplot matrix* instead. Describe how the scatterplot matrix agrees with your interpretation of the slopes in parts 1, 2 and 3 of this question.

Solution

The R code (below) was used to answer all questions.

1. • The model is: $\hat{y} = 102.5 - 0.69T$, where T is tank temperature.

   • Intercept = 102.5 % points is the yield when operating at $0\,°$C. Obviously not a useful interpretation, because data have not been collected in a range that spans, or is even close to $0\,°$C. It is likely that this bioreactor system won't yield any product under such cold conditions. Further, a yield greater than 100% is not realizable.

   • Slope = -0.69 $\frac{[\%]}{[°C]}$, indicating the yield decreases, on average, by about 0.7 units for every degree increase in tank temperature.

2. • The model is: $\hat{y} = -20.3 + 0.016S$, where S is impeller speed.

   • Intercept = -20.3 % points is the yield when operating no agitation. Again, obviously not a useful interpretation, because the data have not been collected under these conditions, and yield can't be a negative quantity.

   • Slope = 0.016 $\frac{[\%]}{[\mathrm{RPM}]}$, indicating the yield increases, on average, by about 1.6 percentage points per 100 RPM increase.

3. • The model is: $\hat{y} = 54.9 - 16.7B$, where B is 1 if baffles are present and $B = 0$ with no baffles.

   • Intercept = 54.9 % points yield is the yield when operating with no baffles (it is in fact the average yield of all the rows that have "No" as their baffle value).

   • Slope = -16.7 %, indicating the presence of baffles decreases the yield, on average, by about 16.7 percentage points.

4. This is an open-ended, and case specific. Some factors you would include are:

   • Remove the baffles, but take into account the cost of doing so. Perhaps it takes a long time (expense) to remove them, especially if the reactor is used to produce other products that do require the baffles.

   • Operate at lower temperatures. The energy costs of cooling the reactor would factor into this.

   • Operate at higher speeds and take that cost into account. Notice however there is one observation at 4900 RPM that seems unusual: was that due to the presence of baffles, or due to temperature in that run? We'll look into this issue with multiple linear regression later on.


   Note: Please note that our calculations above are not the true effect of each of the variables (temperature, speed and baffles) on yield. Our calculations assume that there is no interaction between temperature, speed and baffles, and that each effect operates independent of the others. That's not necessarily true. See the section on *interpreting MLR coefficients* (page 189) to learn how to "control for the effects" of other variables.

   ---

5. The scatterplot matrix, shown below, agrees with our interpretation. This is an information rich visualization that gives us a feel for the multivariate relationships and really summarizes all the variables well (especially the last row of plots).

- The yield-temperature relationship is negative, as expected.

- The yield-speed relationship is positive, as expected.

- The yield-baffles relationship is negative, as expected.

- We can't tell anything about the yield-duration relationship, as it doesn't vary in the data we have (there could/should be a relationship, but we can't tell).
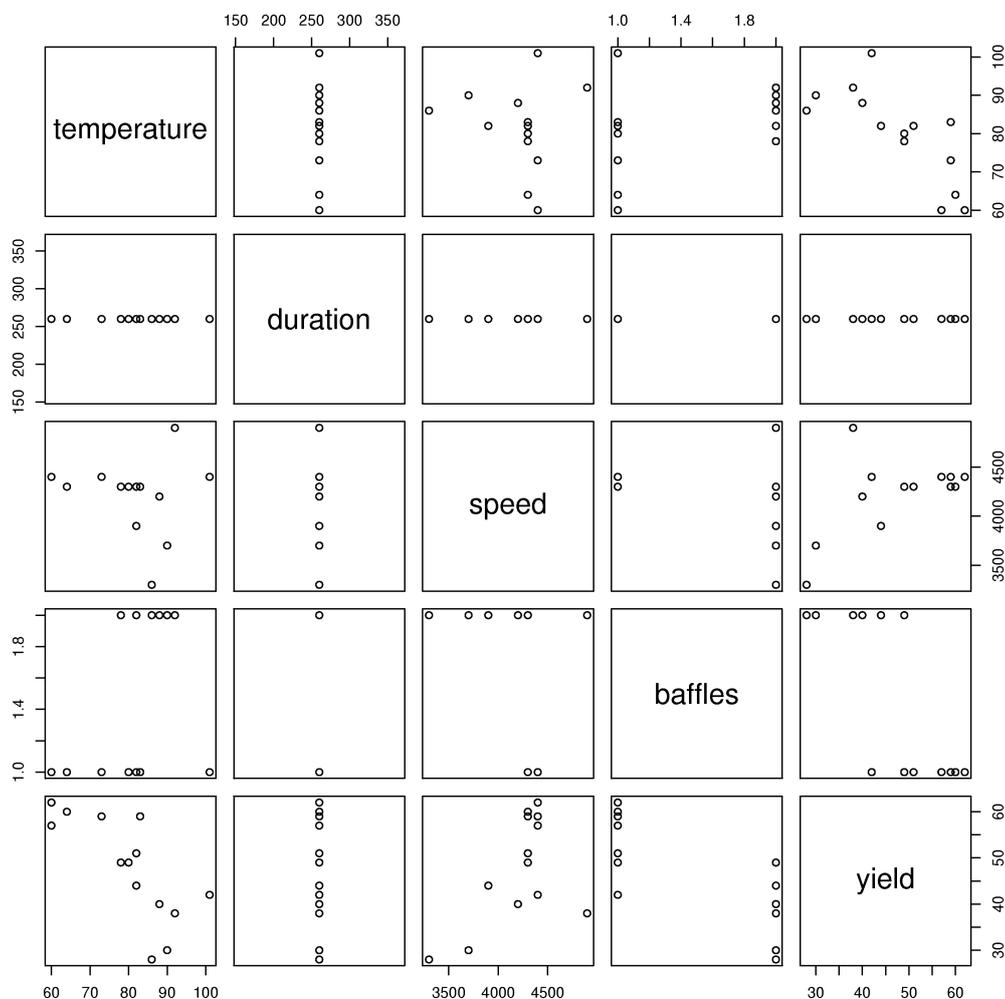

```
bio <- read.csv('http://openmv.net/file/bioreactor-yields.csv')
summary(bio)

# Temperature-Yield model
model.temp <- lm(bio$yield ~ bio$temperature)
summary(model.temp)

# Impeller speed-Yield model
model.speed <- lm(bio$yield ~ bio$speed)
summary(model.speed)

# Baffles-Yield model
model.baffles <- lm(bio$yield ~ bio$baffles)
summary(model.baffles)

# Scatterplot matrix
bitmap('bioreactor-scatterplot-matrix.png', type="png256",
```

(continues on next page)

```
        width=10, height=10, res=300)
plot(bio)
dev.off()
```

Question 11

Use the gas furnace data[94] from the website to answer these questions. The data represent the gas flow rate (centered) from a process and the corresponding CO_2 measurement.

1. Make a scatter plot of the data to visualize the relationship between the variables. How would you characterize the relationship?

2. Calculate the variance for both variables, the covariance between the two variables, and the correlation between them, $r(x, y)$. Interpret the correlation value; i.e. do you consider this a strong correlation?

3. Now calculate a least squares model relating the gas flow rate as the x variable to the CO_2 measurement as the y-variable. Report the intercept and slope from this model.

4. Report the R^2 from the regression model. Compare the squared value of $r(x, y)$ to R^2. What do you notice? Now reinterpret what the correlation value means (i.e. compare this interpretation to your answer in part 2).

5. **Advanced**: Switch x and y around and rebuild your least squares model. Compare the new R^2 to the previous model's R^2. Is this result surprising? How do interpret this?

Question 12

A new type of thermocouple[95] is being investigated by your company's process control group. These devices produce an *almost* linear voltage (millivolt) response at different temperatures. In practice though it is used the other way around: use the millivolt reading to predict the temperature. The process of fitting this linear model is called *calibration*.

1. Use the following data to calibrate a linear model:

Temperature [K]	273	293	313	333	353	373	393	413	433	453
Reading [mV]	0.01	0.12	0.24	0.38	0.51	0.67	0.84	1.01	1.15	1.31

   Show the linear model and provide the predicted temperature when reading 1.00 mV.

2. Are you satisfied with this model, based on the coefficient of determination (R^2) value?

3. What is the model's standard error? Now, are you satisfied with the model's prediction ability, given that temperatures can usually be recorded to an accuracy of ± 0.5 K with most inexpensive thermocouples.

4. What is your (revised) conclusion now about the usefulness of the R^2 value?

---

[94] http://openmv.net/info/gas-furnace
[95] https://en.wikipedia.org/wiki/Thermocouple

Note: This example explains why we don't use the terminology of *independent* and *dependent* variables in this book. Here the temperature truly is the independent variable, because it causes the voltage difference that we measure. But the voltage reading is the independent variable in the least squares model. The word *independent* is being used in two different senses (its English meaning *vs* its mathematical meaning), and this can be misleading.

Question 13

1. Use the linear model you derived in the *gas furnace question* (page 212), where you used the gas flow rate to predict the CO_2 measurement, and construct the analysis of variance table (ANOVA) for the dataset. Use your ANOVA table to reproduce the residual standard error, S_E value, that you get from the R software output.

   Go through the R tutorial[96] to learn how to efficiently obtain the residuals and predicted values from a linear model object.

2. Also for the above linear model, verify whether the residuals are normally distributed.

3. Use the linear model you derived in *the thermocouple question* (page 212), where you used the voltage measurement to predict the temperature, and construct the analysis of variance table (ANOVA) for that dataset. Use your ANOVA table to reproduce the residual standard error, S_E value, that you get from the R software output.

Solution

1. The ANOVA table values were calculated in the code solutions for question 2:

Type of variance	Distance	Degrees of freedom	SSQ	Mean square
Regression	$\hat{y}_i - \overline{y}$	$k - 2$	709.9	354.9
Error	$y_i - \hat{y}_i$	$n - k$	2314.9	7.87
Total	$y_i - \overline{y}$	n	3024.8	10.2

   The residual standard error, or just standard error, $S_E = \sqrt{\frac{2314.9}{296-2}} = 2.8\,\%CO_2$, which agrees with the value from R.

2. These residuals were normally distributed, as verified in the q-q plot:

---

[96] https://learnche.org/4C3/Software_tutorial
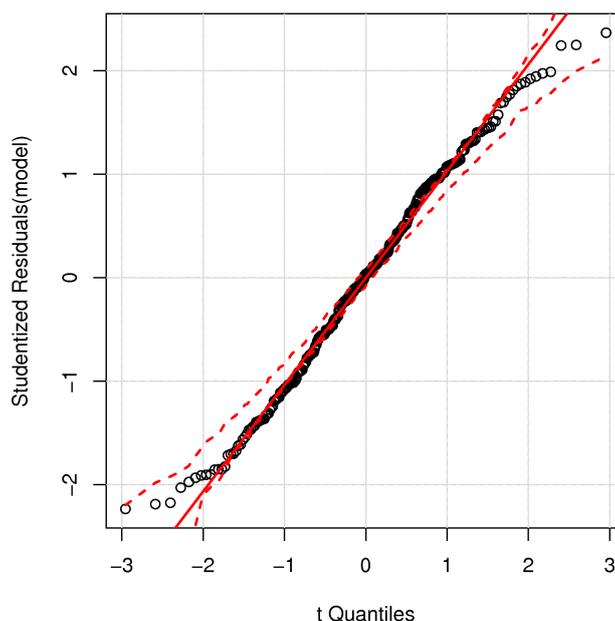
As mentioned in the `help(qqPlot)` output, the dashed red line is the confidence envelope at the 95% level. The single point just outside the confidence envelope is not going to have any practical effect on our assumption of normality. We expect 1 point in 20 to lie outside the limits.

Read ahead, if required, on the meaning of *studentized residuals* (page 194), which are used on the y-axis.

3. For the thermocouple data set:

Type of variance	Distance	Degrees of freedom	SSQ	Mean square
Regression	$\hat{y}_i - \overline{y}$	$k - 2$	32877	16438
Error	$y_i - \hat{y}_i$	$n - k$	122.7	15.3
Total	$y_i - \overline{y}$	n	33000	3300

The residual standard error, or just standard error, $S_E = \sqrt{\frac{122.7}{10-2}} = 3.9$ K, which agrees with the value from R.

Question 14

Use the mature cheddar cheese data set[97] for this question.

1. Choose any x-variable, either `Acetic` acid concentration (already log-transformed), `H2S` concentration (already log-transformed), or `Lactic` acid concentration (in original units) and use this to predict the `Taste` variable in the data set. The `Taste` is a subjective measurement, presumably measured by a panel of tasters.

   Prove that you get the same linear model coefficients, R^2, S_E and confidence intervals whether or not you first mean center the x and y variables.

2. What is the level of correlation between each of the x-variables. Also show a scatterplot matrix to learn what this level of correlation looks like visually.

---

[97] http://openmv.net/info/cheddar-cheese

- Report your correlations as a 3×3 matrix, where there should be 1.0's on the diagonal, and values between -1 and $+1$ on the off-diagonals.

3. Build a linear regression that uses all three x-variables to predict y.

   - Report the slope coefficient and confidence interval for each x-variable

   - Report the model's standard error. Has it decreased from the model in part 1?

   - Report the model's R^2 value. Has it decreased?

Solution

1. We used the acetic acid variable as x and derived the following two models to predict taste, y:

   - No mean centering of x and y: $y = -61.5 + 15.65x$

   - With mean centering of x and y: $y = 0 + 15.65x$

   These results were found from *both* models:

   - Residual standard error, S_E = 13.8 on 28 degrees of freedom

   - Multiple R-squared, R^2 = 0.30

   - Confidence interval for the slope, b_a was: $6.4 \leq b_A \leq 24.9$.

   Please see the R code at the end of this question.

   If you had used x = H2S, then $S_E = 10.8$ and if used x = Lactic, then $S_E = 11.8$.

2. The visual level of correlation is shown in the first 3×3 plots below, while the relationship of each x to y is shown in the last row and column:
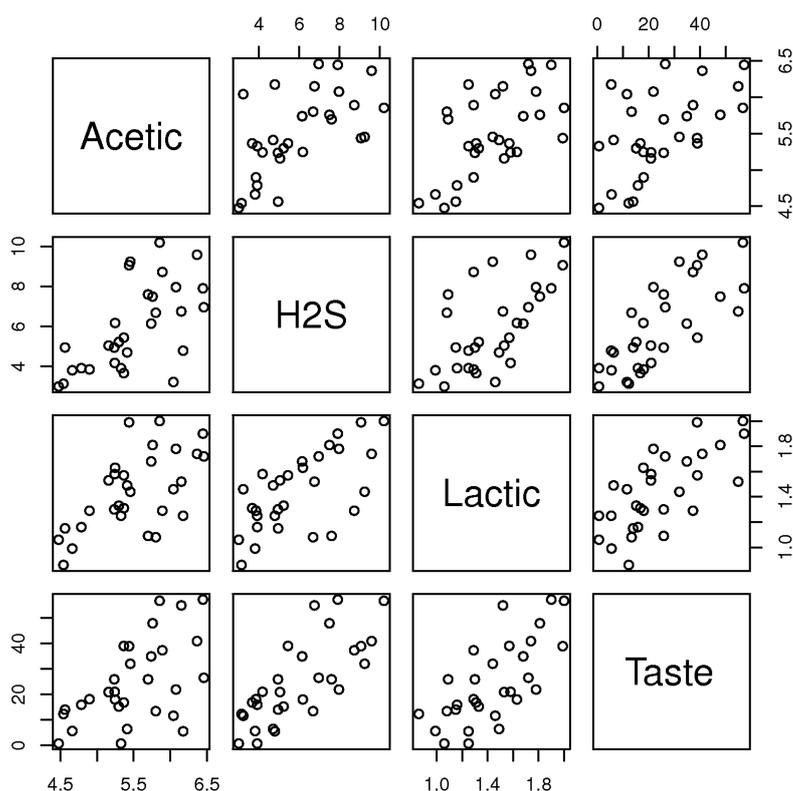
The numeric values for the correlation between the x-variables are:

$$\begin{bmatrix} 1.0 & 0.618 & 0.604 \\ 0.618 & 1.0 & 0.644 \\ 0.604 & 0.644 & 1.0 \end{bmatrix}$$

There is about a 60% correlation between each of the x-variables in this model, and in each case the correlation is positive.

3. A combined linear regression model is $y = -28.9 + 0.31x_A + 3.92x_S + 19.7x_L$ where x_A is the log of the acetic acid concentration, x_S is the log of the hydrogen sulphide concentration and x_L is the lactic acid concentration in the cheese. The confidence intervals for each coefficient are:

- $-8.9 \le b_A \le 9.4$

- $1.4 \le b_S \le 6.5$

- $1.9 \le b_A \le 37$

The R^2 value is 0.65 in the MLR, compared to the value of 0.30 in the single variable regression. The R^2 value will always decrease when adding a new variable to the model, even if that variable has little value to the regression model (yet another caution related to R^2).

The MLR standard error is 10.13 on 26 degrees of freedom, a decrease of about 3 units from the individual regression in part 1; a small decrease given the y-variable's range of about 50 units.

Since each x-variable is about 60% correlated with the others, we can loosely interpret this by inferring that *either* `lactic`, *or* `acetic` *or* `H2S` could have been used in a single-variable regression. In fact, if you compare S_E values for the single-variable regressions, (13.8, 10.8 and 11.8), to the combined regression S_E of 10.13, there isn't much of a reduction in the MLR's standard error.

This interpretation can be quite profitable: it means that we get by with one only one x-variable to make a reasonable prediction of taste in the future, however, the other two measurements must be consistent. In other words we can pick lactic acid as our predictor of taste (it might be the cheapest of the 3 to measure). But a new cheese with high lactic acid, must also have high levels of H2S and acetic acid for this prediction to work. If those two, now unmeasured variables, had low levels, then the predicted taste may not be an accurate reflection of the true cheese's taste! We say "the correlation structure has been broken" for that new observation.

Other, advanced explanations:

Highly correlated x-variables are problematic in least squares, because the confidence intervals and slope coefficients are not independent anymore. This leads to the problem we see above: the acetic acid's effect is shown to be insignificant in the MLR, yet it was significant in the single-variable regression! Which model do we believe?

This resolution to this problem is simple: look at the raw data and see how correlated each of the x-variables are with each other. One of the shortcomings of least squares is that we must invert $\mathbf{X}'\mathbf{X}$. For highly correlated variables this matrix is unstable in that small changes in the data lead to large changes in the inversion. What we need is a method that handles correlation.

One quick, simple, but suboptimal way to deal with high correlation is to create a new variable, $x_{\text{avg}} = 0.33x_A + 0.33x_S + 0.33x_L$ that blends the 3 separate pieces of information into an average. Averages are always less noisy than the separate variables the make up the average. Then use this average in a single-variable regression. See the code below for an example.

```
cheese <- read.csv('http://openmv.net/file/cheddar-cheese.csv')
summary(cheese)

# Proving that mean-centering has no effect on model parameters
x <- cheese$Acetic
y <- cheese$Taste
summary(lm(y ~ x))
confint(lm(y ~ x))

x.mc <- x - mean(x)
y.mc <- y - mean(y)
summary(lm(y.mc ~  x.mc))
confint(lm(y.mc ~  x.mc ))

# Correlation amount in the X's.  Also plot it
cor(cheese[,2:5])
bitmap('cheese-data-correlation.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
plot(cheese[,2:5])
dev.off()

# Linear regression that uses all three X's
model <- lm(cheese$Taste ~ cheese$Acetic + cheese$H2S + cheese$Lactic)
summary(model)
confint(model)

# Use an "average" x
x.avg <- 1/3*cheese$Acetic + 1/3*cheese$H2S + 1/3*cheese$Lactic
model.avg <- lm(cheese$Taste ~ x.avg)
summary(model.avg)
confint(model.avg)
```

Question 15

In this question we will revisit the bioreactor yield[98] data set and fit a linear model with all x-variables to predict the yield. (This data was also used *in a previous question* (page 209).)

1. Provide the interpretation for each coefficient in the model, and also comment on each one's confidence interval when interpreting it.

2. Compare the 3 slope coefficient values you just calculated, to those from the previous question:

   - $\hat{y} = 102.5 - 0.69T$, where T is tank temperature

   - $\hat{y} = -20.3 + 0.016S$, where S is impeller speed

   - $\hat{y} = 54.9 - 16.7B$, where B is 1 if baffles are present and $B = 0$ with no baffles

   Explain why your coefficients do not match.

3. Are the residuals from the multiple linear regression model normally distributed?

4. In this part we are investigating the variance-covariance matrices used to calculate the linear model.

   1. First center the x-variables and the y-variable that you used in the model.

      Note: feel free to use MATLAB, or any other tool to answer this question. If you are using R, then you will benefit from this page in the R tutorial[99]. Also, read the help for the `model.matrix(...)` function to get the **X**-matrix. Then read the help for the `sweep(...)` function, or more simply use the `scale(...)` function to do the mean-centering.

   2. Show your calculated $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{y}$ variance-covariance matrices from the centered data.

   3. Explain why the interpretation of covariances in $\mathbf{X}^T\mathbf{y}$ match the results from the full MLR model you calculated in part 1 of this question.

   4. Calculate $\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ and show that it agrees with the estimates that R calculated (even though R fits an intercept term, while your **b** does not).

5. What would be the predicted yield for an experiment run without baffles, at 4000 rpm impeller speed, run at a reactor temperature of 90 °C?

Question 16

In this question we will use the LDPE data[100] which is data from a high-fidelity simulation of a low-density polyethylene reactor. LDPE reactors are very long, thin tubes. In this particular case the tube is divided in 2 zones, since the feed enters at the start of the tube, and some point further down the tube (start of the second zone). There is a temperature profile along the tube, with a certain maximum temperature somewhere along the length. The maximum temperature in zone 1, `Tmax1` is reached some fraction `z1` along the length; similarly in zone 2 with the `Tmax2` and `z2` variables.

We will build a linear model to predict the `SCB` variable, the short chain branching (per 1000 carbon atoms) which is an important quality variable for this product. Note that the last 4 rows of data are known to be from abnormal process operation, when the process started to experience a problem. However, we will pretend we didn't know that when building the model, so keep them in for now.

---

[98] http://openmv.net/info/bioreactor-yields
[99] https://learnche.org/4C3/Software_tutorial/Vectors_and_matrices
[100] http://openmv.net/info/ldpe

1. Use only the following subset of x-variables: `Tmax1`, `Tmax2`, `z1` and `z2` and the y variable = `SCB`. Show the relationship between these 5 variables in a scatter plot matrix.

   Use this code to get you started (make sure you understand what it is doing):

   ```
   LDPE <- read.csv('http://openmv.net/file/ldpe.csv')
   subdata <- data.frame(cbind(LDPE$Tmax1, LDPE$Tmax2, LDPE$z1, LDPE$z2, LDPE$SCB))
   colnames(subdata) <- c("Tmax1", "Tmax2", "z1", "z2", "SCB")
   ```

   Using bullet points, describe the nature of relationships between the 5 variables, and particularly the relationship to the y-variable.

2. Let's start with a linear model between `z2` and `SCB`. We will call this the `z2` model. Let's examine its residuals:

   1. Are the residuals normally distributed?

   2. What is the standard error of this model?

   3. Are there any time-based trends in the residuals (the rows in the data are already in time-order)?

   4. Use any other relevant plots of the predicted values, the residuals, the x-variable, as described in class, and diagnose the problem with this linear model.

   5. What can be done to fix the problem? (You don't need to implement the fix yet).

3. Show a plot of the hat-values (leverage) from the `z2` model.

   1. Add suitable horizontal cut-off lines to your hat-value plot.

   2. Identify on your plot the observations that have large leverage on the model

   3. Remove the high-leverage outliers and refit the model. Call this the `z2.updated` model

   4. Show the updated hat-values and verify whether the problem has mostly gone away

   Note: see the R tutorial on how to rebuild a model by removing points

4. Use the `influenceIndexPlot(...)` function in the `car` library on both the `z2` model and the `z2.updated` model. Interpret what each plot is showing for the two models. You may ignore the *Bonferroni p-values* subplot.

Solution

1. A scatter plot matrix of the 5 variables is
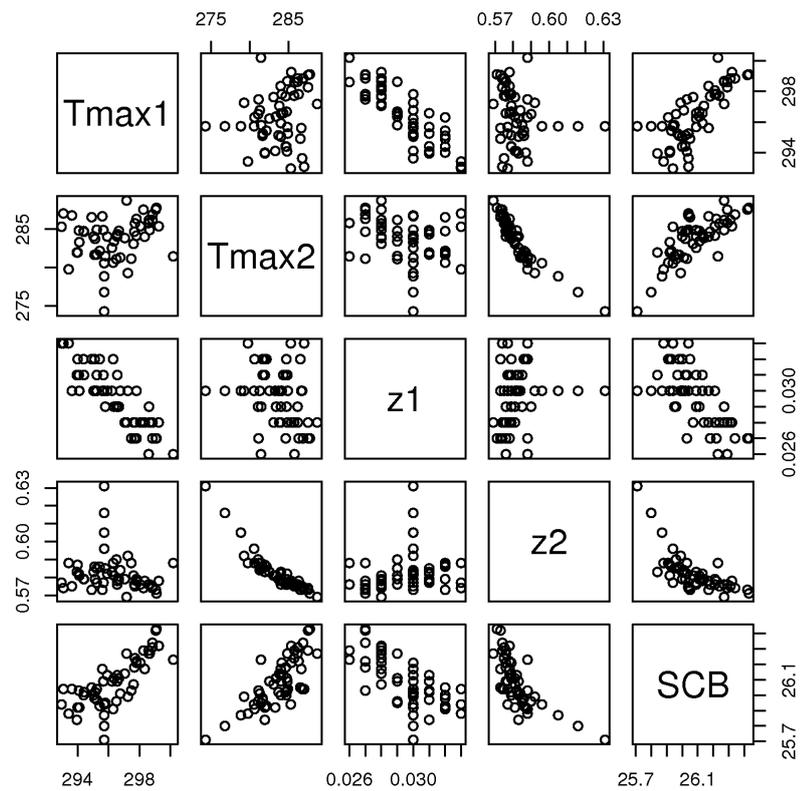
- `Tmax1` and `z1` show a strongish negative correlation

- `Tmax1` and `SCB` show a strong positive correlation

- `Tmax2` and `z2` have a really strong negative correlation, and the 4 outliers are very clearly revealed in almost any plot with `z2`

- `z1` and `SCB` have a negative correlation

- `Tmax2` and `SCB` have a negative correlation

- Very little relationship appears between `Tmax1` and `Tmax2`, which is expected, given how/where these 2 data variables are recorded.

- Similarly for `Tmax2` and `z2`.

2. A linear model between `z2` and `SCB`: $\widehat{\text{SCB}} = 32.23 - 10.6z_2$

First start with a plot of the raw data with this regression line superimposed:
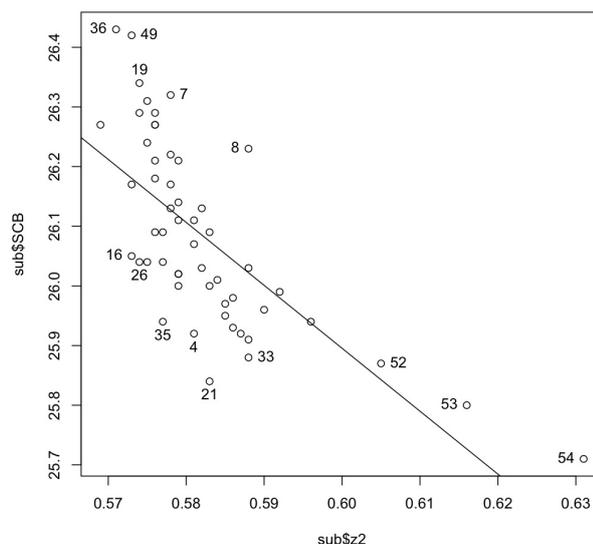
which helps when we look at the q-q plot of the Studentized residuals to see the positive and the negative residuals:
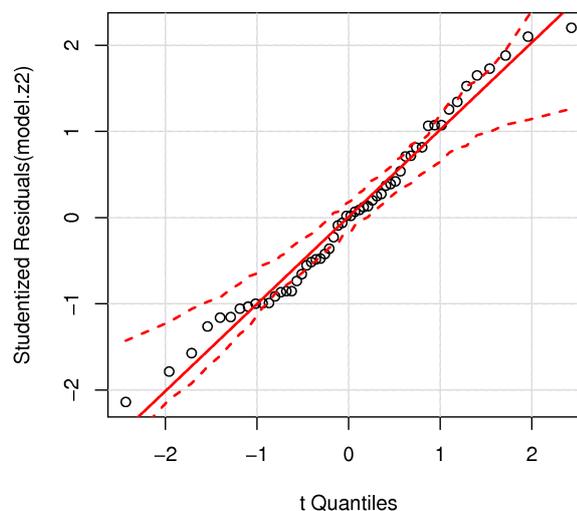
1. We notice there is no strong evidence of non-normality, however, we can see a trend in the tails on both sides (there are large positive residuals and large negative residuals). The identified points in the two plots help understand which points affect the residual tails.

2. This model's standard error is $S_E = 0.114$, which should be compared to the range of the y-axis, 0.70 units, to get an idea whether this is large or small, so about 15% of the range. Given that a conservative estimate of the prediction interval is $\pm 2 S_E$, or a total range of $4 S_E$, this is quite large.

3. The residuals in time-order
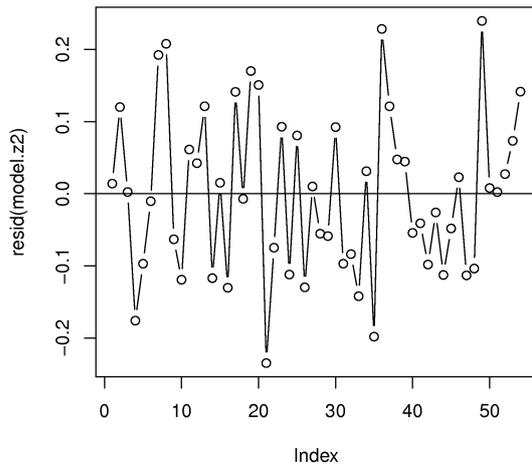
Show no consistent structure, however we do see the short upward trend in the last 4 points. The autocorrelation function (not shown here), shows there is no autocorrelation, i.e. the residuals appear independent.

4. Three plots that do show a problem with the linear model:

   - *Predictions vs residuals*: definite structure in the residuals. We expect to see no structure, but a definite trend, formed by the 4 points is noticeable, as well as a negative correlation at high predicted SCB.
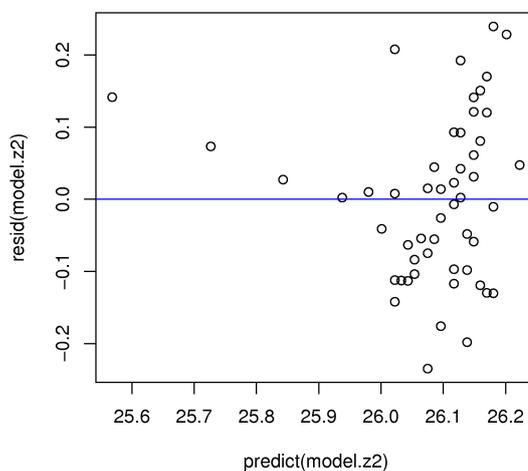
   - x-variable vs residuals: definite structure in the residuals, which is similar to the above plot.

   - Predicted vs measured y: we expect to see a strong trend about a 45° line (shown in blue). The strong departure from this line indicates there is a problem with the model
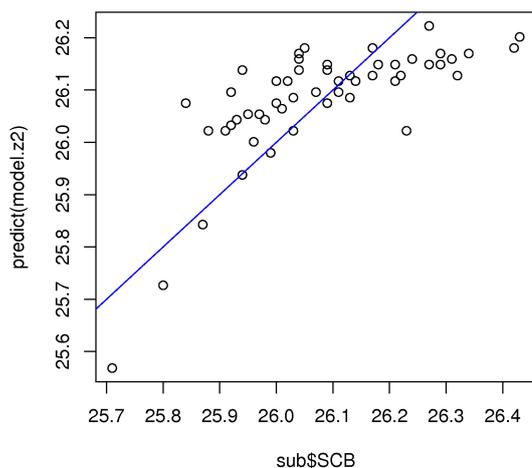
5. We can consider removing the 4 points that strongly bias the observed *vs* predicted plot above.

3. A plot of the hat-values (leverage) from the regression of `SCB` on `z2` is:
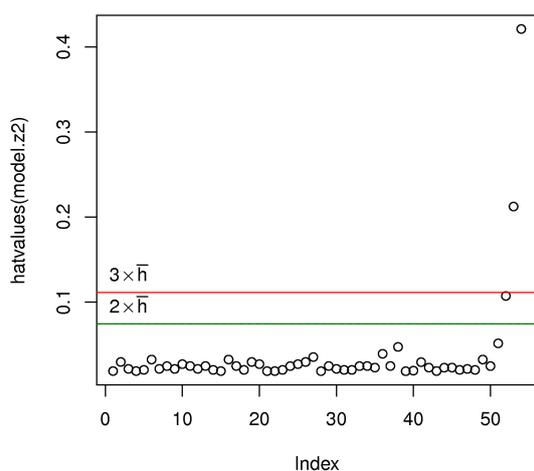
with 2 and 3 times the average hat value shown for reference. Points 52, 53 and 54 have leverage that is excessive, confirming what we saw in the previous part of this question.

Once these points are removed, the model was rebuilt, and this time showed point 51 as an high-leverage outlier. This point was removed and the model rebuilt.

The hat values from this updated model are:
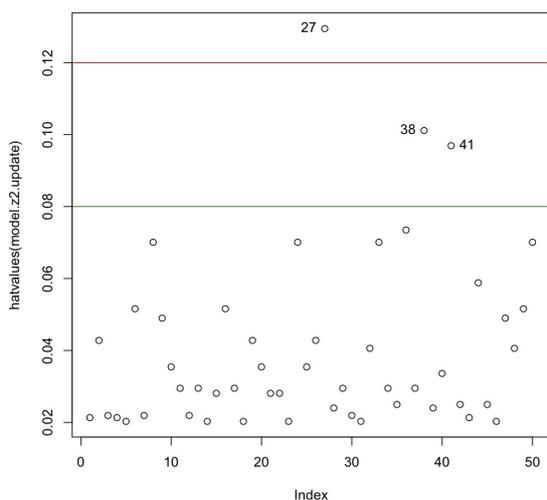
which is reasonable to stop at, since the problem has mostly gone away. If you keep omitting points, you will likely deplete all the data. At some point, especially when there is no obvious structure in the residuals, it is time to stop interrogating (i.e. investigating) and removing outliers.

The updated model has a slightly improved standard error $S_E = 0.11$ and the least squares model fit (see the R code) appears much more reasonable in the data.

4. The influence index plots for the model with all 54 points is shown first, followed by the influence index plot of the model with only the first 50 points.
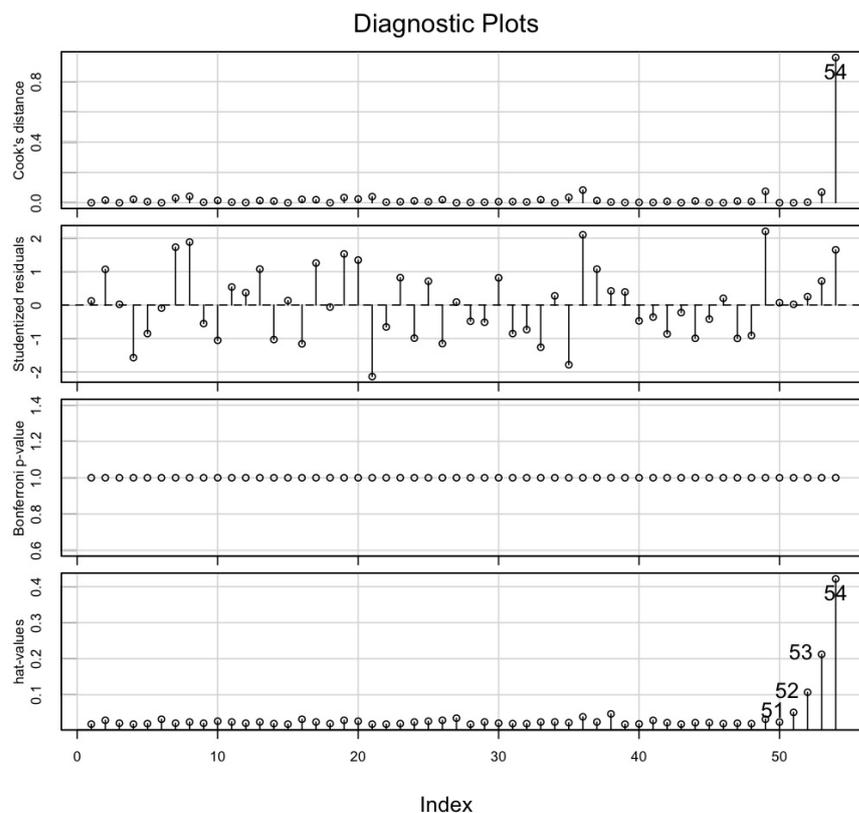
The increasing leverage, as the abnormal process operation develops is clearly apparent. This leverage is not "bad" (i.e. influential) initially, because it is "in-line" with the regression slope. But by observation 54, there is significant deviation that observation 54 has high residuals distance, and therefore a combined high influence on the model (high Cook's D).

Diagnostic Plots
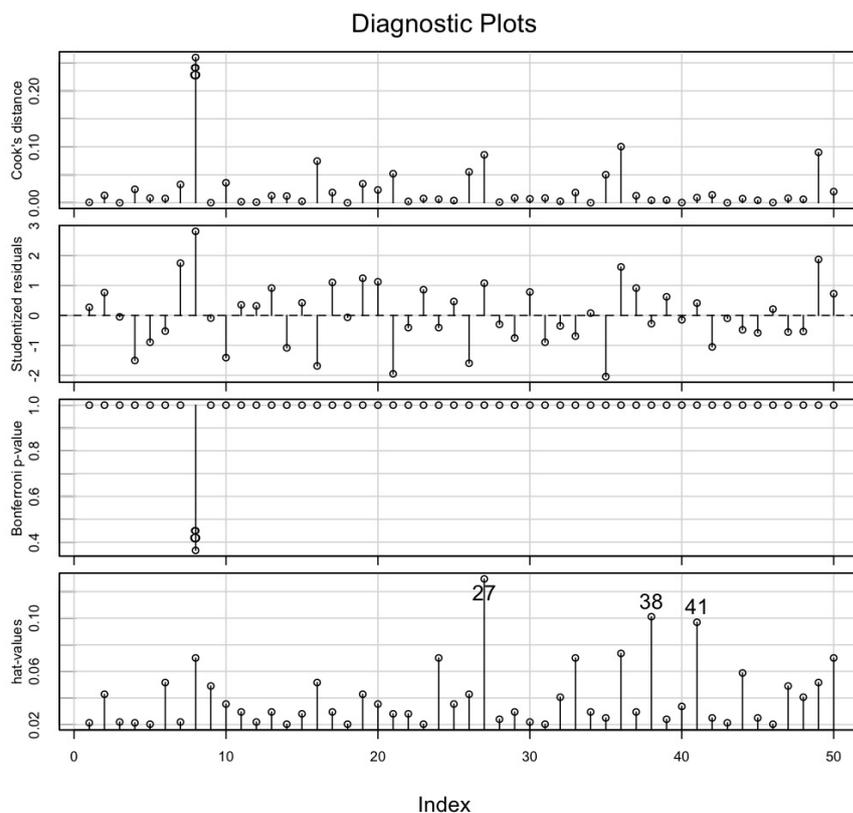
The updated model shows shows only point 8 as an influential observation, due to its moderate leverage and large residual. However, this point does not warrant removal, since it is just above the cut-off value of $4/(n - k) = 4/(50 - 2) = 0.083$ for Cook's distance.

The other large hat values don't have large Studentized residuals, so they are not influential on the model.

Notice how the residuals in the updated model are all a little smaller than in the initial model.

All the code for this question is given here:

```
LDPE <- read.csv('http://openmv.net/file/LDPE.csv')
summary(LDPE)
N <- nrow(LDPE)

sub <- data.frame(cbind(LDPE$Tmax1, LDPE$Tmax2, LDPE$z1, LDPE$z2, LDPE$SCB))
colnames(sub) <- c("Tmax1", "Tmax2", "z1", "z2", "SCB")

bitmap('ldpe-scatterplot-matrix.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
plot(sub)
dev.off()

model.z2 <- lm(sub$SCB ~ sub$z2)
summary(model.z2)
```

(continues on next page)

```
# Plot raw data
bitmap('ldpe-z2-SCB-raw-data.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
plot(sub$z2, sub$SCB)
abline(model.z2)
identify(sub$z2, sub$SCB)
dev.off()

# Residuals normal? Yes, but have heavy tails
bitmap('ldpe-z2-SCB-resids-qqplot.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
library(car)
qqPlot(model.z2, id.method="identify")
dev.off()

# Residual plots in time order: no problems detected
# Also plotted the acf(...): no problems there either
bitmap('ldpe-z2-SCB-raw-resids-in-order.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
plot(resid(model.z2), type='b')
abline(h=0)
dev.off()

acf(resid(model.z2))

# Predictions vs residuals: definite structure in the residuals!
bitmap('ldpe-z2-SCB-predictions-vs-residuals.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
plot(predict(model.z2), resid(model.z2))
abline(h=0, col="blue")
dev.off()

# x-data vs residuals: definite structure in the residuals!
bitmap('ldpe-z2-SCB-residual-structure.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
plot(sub$Tmax2, resid(model.z2))
abline(h=0, col="blue")
identify(sub$z2, resid(model.z2))
dev.off()

# Predictions-vs-y
bitmap('ldpe-z2-SCB-predictions-vs-actual.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
plot(sub$SCB, predict(model.z2))
abline(a=0, b=1, col="blue")
identify(sub$SCB, predict(model.z2))
dev.off()

# Plot hatvalues
bitmap('ldpe-z2-SCB-hat-values.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
plot(hatvalues(model.z2))
avg.hat <- 2/N
abline(h=2*avg.hat, col="darkgreen")
abline(h=3*avg.hat, col="red")
text(3, y=2*avg.hat, expression(2 %*% bar(h)), pos=3)
text(3, y=3*avg.hat, expression(3 %*% bar(h)), pos=3)
identify(hatvalues(model.z2))
dev.off()

# Remove observations (observation 51 was actually detected after
# the first iteration of removing 52, 53, and 54: high-leverage points)
build <- seq(1,N)
remove <- -c(51, 52, 53, 54)
model.z2.update <- lm(model.z2, subset=build[remove])

# Plot updated hatvalues
```

```
plot(hatvalues(model.z2.update))
N <- length(model.z2.update$residuals)
avg.hat <- 2/N
abline(h=2*avg.hat, col="darkgreen")
abline(h=3*avg.hat, col="red")
identify(hatvalues(model.z2.update))
# Observation 27 still has high leverage: but only 1 point

# Problem in the residuals gone? Yes
plot(predict(model.z2.update), resid(model.z2.update))
abline(h=0, col="blue")

# Does the least squares line fit the data better?
plot(sub$z2, sub$SCB)
abline(model.z2.update)

# Finally, show an influence plot
influencePlot(model.z2, id.method="identify")
influencePlot(model.z2.update, id.method="identify")

# Or the influence index plots
influenceIndexPlot(model.z2, id.method="identify")
influenceIndexPlot(model.z2.update, id.method="identify")


#-------- Use all variables in an MLR (not required for question)

model.all <- lm(sub$SCB ~ sub$z1 + sub$z2 + sub$Tmax1 + sub$Tmax2)
summary(model.all)
confint(model.all)
```

Question 17

A concrete slump test is used to test for the fluidity, or workability, of concrete. It's a crude, but quick test often used to measure the effect of polymer additives that are mixed with the concrete to improve workability.

The concrete mixture is prepared with a polymer additive. The mixture is placed in a mold and filled to the top. The mold is inverted and removed. The height of the mold minus the height of the remaining concrete pile is called the "slump".
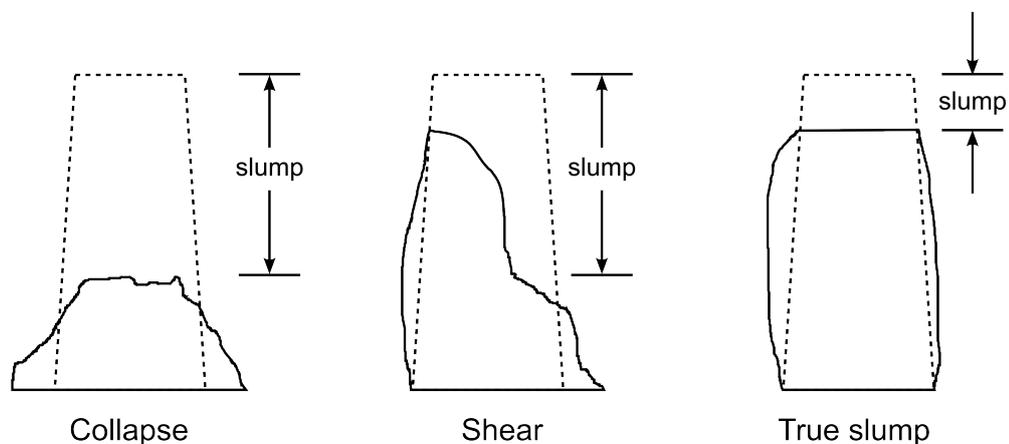

Figure from Wikipedia[101]

---

[101] https://en.wikipedia.org/wiki/File:Types_of_concrete_slump.jpg

Your company provides the polymer additive, and you are developing an improved polymer formulation, call it B, that hopefully provides the same slump values as your existing polymer, call it A. Formulation B costs less money than A, but you don't want to upset, or lose, customers by varying the slump value too much.

The following slump values were recorded over the course of the day:

Additive	Slump value [cm]
A	5.2
A	3.3
B	5.8
A	4.6
B	6.3
A	5.8
A	4.1
B	6.0
B	5.5
B	4.5

You can derive the 95% confidence interval for the true, but unknown, difference between the effect of the two additives:

$$
\begin{array}{ccccc}
-c_t & \leq & z & \leq & +c_t \\
(\bar{x}_B - \bar{x}_A) - c_t\sqrt{s_P^2\left(\frac{1}{n_B} + \frac{1}{n_A}\right)} & \leq & \mu_B - \mu_A & \leq & (\bar{x}_B - \bar{x}_A) + c_t\sqrt{s_P^2\left(\frac{1}{n_B} + \frac{1}{n_A}\right)} \\
1.02 - 2.3\sqrt{0.709\left(\frac{1}{5} + \frac{1}{5}\right)} & \leq & \mu_B - \mu_A & \leq & 1.02 + 2.3\sqrt{0.709\left(\frac{1}{5} + \frac{1}{5}\right)} \\
-0.21 & \leq & \mu_B - \mu_A & \leq & 2.2
\end{array}
$$

Fit a least squares model to the data using an integer variable, $x_A = 0$ for additive A, and $x_A = 1$ for additive B. The model should include an intercept term also: $y = b_0 + b_A x_A$. *Hint*: use R to build the model, and search the R tutorial with the term *categorical variable* or *integer variable* for assistance.

Show that the 95% confidence interval for b_A gives exactly the same lower and upper bounds, as derived above with the traditional approach for tests of differences.

Solution

This short piece of R code shows the expected result when regressing the slump value onto the binary factor variable:

```
additive <- as.factor(c("A", "A", "B", "A", "B", "A", "A", "B", "B", "B"))
slump <- c(5.2, 3.3, 5.8, 4.6, 6.3, 5.8, 4.1, 6.0, 5.5, 4.5)
confint(lm(slump ~ additive))

               2.5 %    97.5 %
(Intercept)  3.7334823 5.466518
additive    -0.2054411 2.245441
```

Note that this approach works only if your coding has a one unit difference between the two levels. For example, you can code $A = 17$ and $B = 18$ and still get the same result. Usually though $A = 0$ and

$B = 1$ or the $A = 1$ and $B = 2$ coding is the most natural, but all 3 of these codings would give the same confidence interval (the intercept changes though).

Question 18

Some data were collected from tests where the compressive strength, x, used to form concrete was measured, as well as the intrinsic permeability of the product, y. There were 16 data points collected. The mean x-value was $\overline{x} = 3.1$ and the variance of the x-values was 1.52. The average y-value was 40.9. The estimated covariance between x and y was -5.5.

The least squares estimate of the slope and intercept was: $y = 52.1 - 3.6x$.

1. What is the expected permeability when the compressive strength is at 5.8 units?

2. Calculate the 95% confidence interval for the slope if the standard error from the model was 4.5 units. Is the slope coefficient statistically significant?

3. Provide a rough estimate of the 95% prediction interval when the compressive strength is at 5.8 units (same level as for part 1). What assumptions did you make to provide this estimate?

4. Now provide a more accurate, calculated 95% prediction confidence interval for the previous part.

Question 19

A simple linear model relating reactor temperature to polymer viscosity is desirable, because measuring viscosity online, in real time is far too costly, and inaccurate. Temperature, on the other hand, is quick and inexpensive. This is the concept of *soft sensors*, also known as *inferential sensors*.

Data were collected from a rented online viscosity unit and a least squares model build:

$$\hat{v} = 1977 - 3.75T$$

where the viscosity, v, is measured in Pa.s (Pascal seconds) and the temperature is in Kelvin. A reasonably linear trend was observed over the 86 data points collected. Temperature values were taken over the range of normal operation: 430 to 480 K and the raw temperature data had a sample standard deviation of 8.2 K.

The output from a certain commercial software package was:

```
                    Analysis of Variance
---------------------------------------------------------
                              Sum of          Mean
Source                DF      Squares         Square
Model                  2      9532.7          4766.35
Error                 84      9963.7           118.6
Total                 86     19496.4
Root MSE         XXXXX
R-Square         XXXXX
---------------------------------------------------------
```

1. Which is the causal direction: does a change in viscosity cause a change in temperature, or does a change in temperature cause a change in viscosity?

2. Calculate the `Root MSE`, what we have called standard error, S_E in this course.

3. What is the R^2 value that would have been reported in the above output?

4. What is the interpretation of the slope coefficient, -3.75, and what are its units?

5. What is the viscosity prediction at 430K? And at 480K?

6. In the future you plan to use this model to adjust temperature, in order to meet a certain viscosity target. To do that you must be sure the change in temperature will lead to the desired change in viscosity.

   What is the 95% confidence interval for the slope coefficient, *and interpret* this confidence interval in the context of how you plan to use this model.

7. The standard error features prominently in all derivations related to least squares. Provide an interpretation of it and be specific in any assumption(s) you require to make this interpretation.

Solution

1. The causal direction is that a change in temperature causes a change in viscosity.

2. The `Root MSE` $= S_E = \sqrt{\dfrac{\sum e_i^2}{n-k}} = \sqrt{\dfrac{9963.7}{84}} = \mathbf{10.9}$ Pa.s.

3. $R^2 = \dfrac{\text{RegSS}}{\text{TSS}} = \dfrac{9532.7}{19496.4} = \mathbf{0.49}$

4. The slope coefficient is $-3.75 \frac{\text{Pa.s}}{K}$ and implies that the viscosity is expected to decrease by 3.75 Pa.s for every one degree increase in temperature.

5. The viscosity prediction at 430K is $1977 - 3.75 \times 430 = \mathbf{364.5}$ Pa.s and is **177** Pa.s at 480 K.

6. The confidence interval is

$$b_1 \pm c_t S_E(b_1)$$
$$-3.75 \pm 1.98 \frac{S_E^2}{\sum_j (x_j - \overline{x})^2}$$
$$-3.75 \pm 1.98 \frac{10.9}{697}$$
$$-3.75 \pm 0.031$$

   where $\dfrac{(x_j - \overline{x})^2}{n-1} = 8.2$, so one can solve for $(x_j - \overline{x})^2$ (though any reasonable value/attempt to get this value should be acceptable) and $c_t = 1.98$, using $n - k$ degrees of freedom at 95% confidence.

   Interpretation: this interval is extremely narrow, i.e. our slope estimate is precise. We can be sure that any change made to the temperature in our system will have the desired effect on viscosity in the feedback control system.

7. The standard error, $S_E = 10.9$ Pa.s is interpreted as the amount of spread in the residuals. In addition, if we assume the residuals to be normally distributed (easily confirmed with a q-q plot) and independent. If that is true, then S_E is the one-sigma standard deviation for the residuals and we can say 95% of the residuals are expected within a range of $\pm 2 S_E$.

DESIGN AND ANALYSIS OF EXPERIMENTS

Note: Coursera students

If you are using this chapter with the Coursera MOOC[102] (massive open online course), then we wish to welcome you and want to let you know that this book is generally part of a larger set of notes. The cross-references in this chapter will point you to other parts, where background knowledge is provided.

This chapter was written for engineers originally, but you will see the examples are very general and can be applied to any other systems.

You can safely skip over the section on *Experiments with a single variable at two levels* (page 237); that section is not covered in the MOOC. You can also initially skip the section on *Why learning about systems is important* (page 235), but make sure you come back and read it.

5.1 Design and analysis of experiments in context

This chapter will take a totally different approach to learning about and understanding systems in general, not only (chemical) engineering systems. The systems we could apply this to could be as straightforward as growing plants or perfecting your favourite recipe at home. Or they may be as complex as the entire production line in a large factory producing multiple products and shipping them to customers.

In order to learn about a system, we have to disturb it and change it. This is to ensure cause and effect. If we do not intentionally change the system, we are only guessing, or using our intuition. To disturb the system, we change several factors. When we make these changes, we say that we have "run an experiment".

In this chapter we learn the best way to intentionally disturb the system to learn more about it. We will use some of the tools of *least squares modelling* (page 151), *visualization* (page 1) and *univariate statistics* (page 29) that were described in earlier chapters. Where necessary, we will refer back to those earlier sections.

---

[102] https://yint.org/experiments

5.2 Terminology

The area of designed experiments uses specific terminology.

Every experiment has these two components:

1. An *outcome*: the result or the *response* from an experiment.

2. One or more factors: a *factor* is the thing you can change to influence the outcome. Factors are also called *variables*.

An important aspect about the outcome is that it is always measurable–in some way. In other words, after you finish the experiment, you must have some measurement.

Let's use an example of growing plants. The outcome of growing a plant might be the height of the plant, or the average width of the leaves, or the number of flowers on the plant. These are numeric measurements, also called quantitative measurements. Qualitative measurements are also possible. For example, perhaps the outcome is the colour of the flower: light red, red, or dark red. A qualitative outcome might also be a description of what happened, for example, *pass* or *fail*.

An experiment can have an *objective*, which combines an outcome and the need to adjust that outcome. For example, you may want to maximize the height of the plant. Most often you want to maximize or minimize the outcome as your objective. Sometimes, though, you want the outcome to be *the same* even though you are changing factors. For example, you might want to change a recipe for your favourite pastry to be gluten-free but keep the taste the same as the original recipe. Your outcome is taste, and your objective is "the same".

Every experiment always has an outcome. Every experiment does not have to have an objective, but usually we have an objective in our mind.

Another term we will use is factors. In the plant example, you could have changed three factors:

1. The amount of water that you give the plant each day

2. The amount of fertilizer that you give the plant each week

3. The type of soil you use, A or B

All experiments must have at least one factor that is changed. We distinguish between two types of factors: *numeric factors* and *categorical factors*.

Numeric factors are quantified by measuring, such as giving 15 mL of water or 30 mL of water to the plant each day. An important point about numerical variables is that there is some order to them. 15 mL of water is less than 30 mL or water. Another name for this type of factor is a quantitative factor.

Categorical factors usually take on a limited number of values. For example, soil type A or soil type B could be used to grow the plants. Categorical variables have no implicit ordering. You could have switched the names of soil A and soil B around. Categorical variables and qualitative variables can be used as synonyms.

Most categorical variables can be converted to continuous variables, with some careful thought. For example: no water vs some water (categorical) can be converted to 0 mL and 40 mL (now it is numeric). In the case of soil A vs soil B it might be that soil A contains a higher level of nutrients in total than soil B, so a numeric version of this factor could be measured as nutrient load.

If you were working in the area of marketing, you might try three different colours of background in your advertising poster. Those 3 colours are categorical variables in the context of the experiment.

Most experiments will have both numeric and categorical factors.

When we perform an experiment, we call it a run. If we perform eight experiments, we can say "there are eight runs" in the set of experiments.

5.3 Usage examples

After you complete this chapter, you will be able to answer questions such as those presented in these scenarios:

- *Colleague*: We have this list of eight plausible factors that affect the polymer melt index (the outcome). How do we narrow down the list to a more manageable size and rank their effect on melt index?

- *You*: Our initial screening experiments reduced the list down to three factors of interest. Now, how do we perform the rest of the experiments?

- *Manager*: Two years ago someone collected these experimental data for the effectiveness of a new chemical to treat water. What interesting results do you see in this data, and where should we operate the system to achieve water quality that meets the required standards?

- *Colleague*: The current production settings for our food product gives us good shelf life, but the energy used is high. How can we determine other settings (factors) that give long shelf life but reduce the energy consumed?

- *Colleague*: We would like to run experiments by varying temperature and pressure, but operating at both high temperature and pressure is unsafe. How do we plan such an experiment?

Here's a visual representation of the topics we will cover in this chapter.
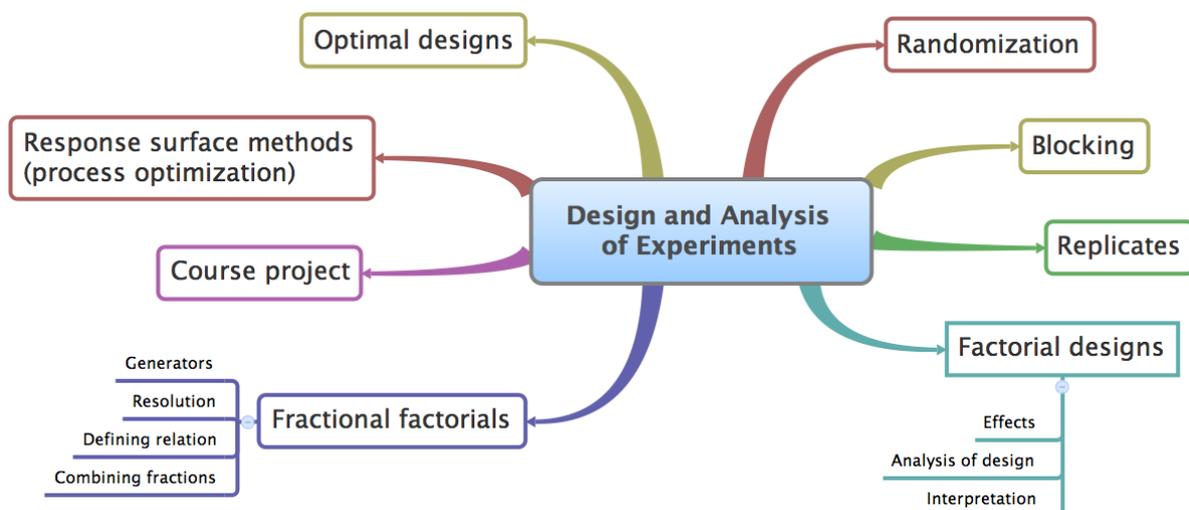
5.4 References and readings

- **Strongly recommended**: Box, Hunter and Hunter, *Statistics for Experimenters*, 2nd edition. Chapters 5 and 6 with topics from Chapters 11, 12, 13 and 15 are the most heavily used in this chapter.

- Søren Bisgaard: Must a Process Be in Statistical Control Before Conducting Designed Experiments?[103], with discussion (part 1[104], part 2[105], part 3[106], part 4[107], part 5[108] and a rejoinder[109]).

- George Box and J. Stuart Hunter, "The 2^{k-p} Fractional Factorial Designs - Part I[110]", *Technometrics*, **3**, 311-351, 1961.

- George Box and J. Stuart Hunter, "The 2^{k-p} Fractional Factorial Designs - Part II[111]", *Technometrics*, **3**, 449-458, 1961.

- George Box, "Evolutionary Operation: A Method for Increasing Industrial Productivity[112]", *Journal of the Royal Statistical Society* (Applied Statistics), **6**, 81-101, 1957.

- William G. Hunter and J. R. Kittrell, "Evolutionary Operation: A Review[113]", *Technometrics*, **8**, 389-397, 1966.

- Heather Tye, "Application of Statistical Design of Experiments Methods in Drug Discovery[114]", *Drug Discovery Today*, **9**, 485-491, 2004.

- R.A. Fisher, Statistical Methods, Experimental Design and Scientific Inference[115], Oxford Science Publications, 2003.

- Raymond H. Myers, Douglas C. Montgomery and Christine M. Anderson-Cook, Response Surface Methodology: Process and Product Optimization Using Designed Experiments[116], Wiley, 2009.

- William Hill and William Hunter, "A Review of Response Surface Methodology: A Literature Survey[117]", *Technometrics*, **8**, 571-590, 1966.

- Owen L. Davies, The Design and Analysis of Industrial Experiments[118], Chapter 11, revised 2nd edition, Hafner, 1967.

---

[103] https://dx.doi.org/10.1080/08982110701826721
[104] https://dx.doi.org/10.1080/08982110701866198
[105] https://dx.doi.org/10.1080/08982110801894892
[106] https://dx.doi.org/10.1080/08982110801890148
[107] https://dx.doi.org/10.1080/08982110801924509
[108] https://dx.doi.org/10.1080/08982110801894900
[109] https://dx.doi.org/10.1080/08982110801973118
[110] https://www.jstor.org/stable/1266725
[111] https://www.jstor.org/stable/1266553
[112] https://www.jstor.org/stable/2985505
[113] https://www.jstor.org/stable/1266686
[114] https://dx.doi.org/10.1016/S1359-6446(04)03086-7
[115] https://www.amazon.com/Statistical-Methods-Experimental-Scientific-Inference/dp/0198522290
[116] https://www.amazon.com/Response-Surface-Methodology-Optimization-Experiments/dp/0470174463
[117] https://www.jstor.org/stable/1266632
[118] https://www.amazon.com/The-design-analysis-industrial-experiments/dp/B0007J7BME

5.5 Why learning about systems is important

One of the important reasons why we must experiment is that it brings us increased knowledge and a better understanding of our system. That could lead to profit, or it could help us manufacture products more efficiently. Once we learn what really happens in our system, we can fix problems and optimize the system, because we have an improved understanding of cause and effect.

As described *in the first reference, the book by Box, Hunter and Hunter* (page 234), learning from and improving a system is an iterative process. It usually follows this cycle:

- Make a conjecture (hypothesis), which we believe is true.

- If it is true, we expect certain consequences.

- Experiment and collect data. Are the consequences that we expected visible in the data?

- If so, it may lead to the next hypothesis. If not, we formulate an alternative hypothesis. Or perhaps it is not so clear cut: we see the consequence, but not to the extent expected. Perhaps modifications are required in the experimental conditions.

And so we go about learning. One of the most frequent reasons we experiment is to fix a problem with our process. This is called troubleshooting. We can list several causes for the problem, change the factors, isolate the problem, and thereby learn more about our system while fixing the problem.

5.5.1 An engineering example

Let's look at an example. We expect that compounds A and B should combine in the presence of a third chemical, C, to form a product D. An initial experiment shows very little of product D is produced. Our goal is to maximize the amount of D. Several factors are considered: temperature, reaction duration and pressure. Using a set of structured experiments, we can get an initial idea of which factors actually impact the amount of D produced. Perhaps these experiments show that only temperature and reaction duration are important and that pressure has little effect. Then we go ahead and adjust only those two factors, and we keep pressure low (to save money because we can now use a less costly, low-pressure reactor). We repeat several more systematic *response surface* (page 278) experiments to maximize our production goal.

The iterations continue until we find the most economically profitable operating point. At each iteration we learn more about our system and how to improve it. The key point is this: you must disturb your system, and then observe it. This is the principle of causality, or *cause and effect*.

It is only by *intentional manipulation* of our systems that we learn from them. Collecting happenstance data, (everyday) operating data, does not always help, because it is confounded by other events that occur at the same time. Everyday, happenstance data is limited by feedback control systems.

5.5.2 Feedback control

Feedback control systems keep the region of operation to a small zone. Better yields or improved operation might exist beyond the bounds created by our automatic control systems. Due to safety concerns, and efficient manufacturing practices, we introduce automated feedback control systems to prevent deviating too far from a desired region of operation. As a result, data collected from such systems has low information quality.

An example would be making eggs for breakfast. If you make eggs the same way each morning (a bit of butter, medium heat for 5 minutes, flip and cook it for 1 minute, then eat), you will never experience anything different. The egg you make this morning is going to taste very similar to one last year, because of your good control system. That's happenstance data.

You must intentionally change the system to perturb it, and then observe it.

5.5.3 Another engineering example

Here's a great example from the book by Box, Hunter and Hunter. Consider the negative-slope *relationship between pressure and yield* (page 236): as pressure increases, the yield drops. A line could be drawn through the points from the happenstance measurements, taken from the process at different times in the past. That line could be from a *least squares model* (page 151). It is true that the observed pressure and yield are correlated, as that is exactly what a least squares model is intended for: to quantify correlation.

The true mechanism in this system is that pressure is increased to remove the frothing that occurs in the reactor. Higher frothing occurs when there is an impurity in the raw material, so operators increase reactor pressure when they see frothing (i.e. high impurity). However, it is the high impurity that actually causes the lower yield, not the pressure itself. These relationships between yield, pressure and impurity levels are illustrated below, based an adaption from the book by Box, Hunter and Hunter, Chapter 14 (1st edition) or Chapter 10 (2nd edition).
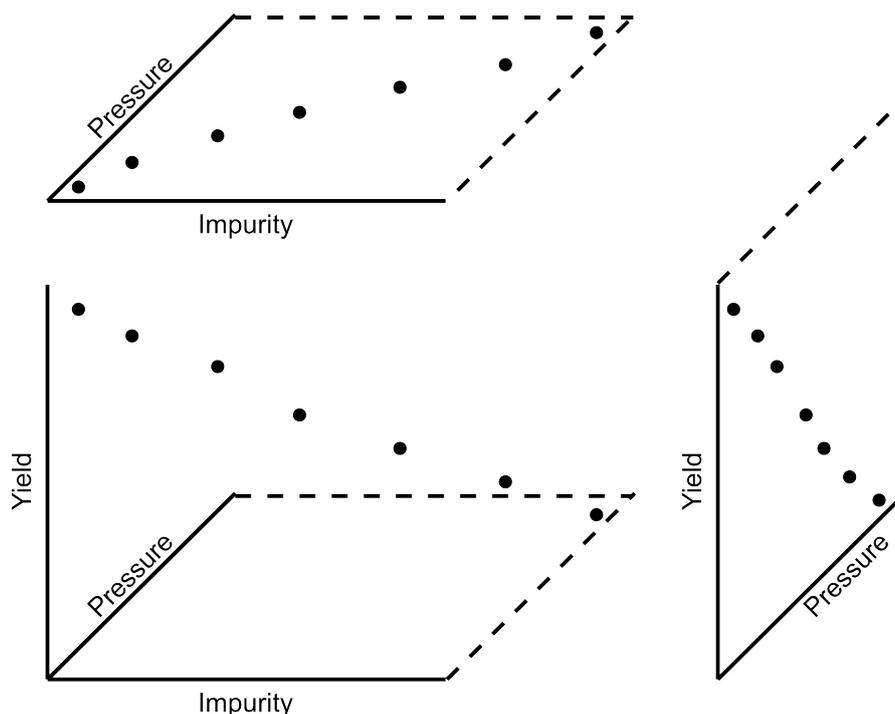
Pressure is correlated with the yield, but there is no cause-and-effect relationship between them. The happenstance relationship only appears in the data because of the operating policy, causing them to be correlated, but it is not cause and effect. That is why happenstance data cannot be relied on to imply cause and effect. An experiment in which the pressure is changed from low to high, performed on the same batch of raw materials (i.e. at constant impurity level), will quickly reveal that there is no causal relationship between pressure and yield.

Another problem with using happenstance data is that they are not taken in random order. Time-based effects, such as changes in the seasonal or daily temperatures, will affect a process. We are all well aware of slow changes: fridges and stoves degrade over time, cars need periodic maintenance. Even our human bodies follow this rule. If we do not randomize the order of experiments, we risk inferring a causal relationship when none actually exists.

Designed experiments are the only way we can be sure that these correlated events are causal. You often hear people repeat the (incomplete) phrase that "correlation does not imply causality". That is only half-true: the other half of the phrase is "correlation is a necessary, but not sufficient, condition for causality".

In summary, do not rely on anecdotal "evidence" from colleagues. Always question the system, and always try to perturb the system intentionally. In practice you won't always be allowed to move the system too drastically, so at the end of this chapter we will discuss *response surface methods* (page 278) and *evolutionary operation* (page 286), which can be implemented on-line in production processes.

Experiments are the most efficient way to extract information about a system, that is, the most information in the fewest number of changes. So it is always worthwhile to experiment.

5.6 Experiments with a single variable at two levels

This is the simplest type of experiment. It involves an outcome variable, y, and one input variable, x. The x-variable could be a continuous numeric one, such as temperature, or discrete on, such as yes/no, on/off, A/B. This type of experiment could be used to answer questions such as the following:

- Has the reaction yield increased when using catalyst A or B?

- Does the concrete's strength improve when adding a particular binder or not?

- Does the plastic's stretchability improve when extruded at various temperatures (a low or high temperature)?

We can perform several runs (experiments) at level A, and some runs at level B. These runs are randomized (i.e. do not perform all the A runs, and then the B runs). We strive to hold all other disturbance variables constant so we pick up only the A-to-B effect. Disturbances are any variables that might affect y but, for whatever reason, we don't wish to quantify. If we cannot control the disturbance, then at least we can use *pairing* (page 76) and *blocking* (page 275). Pairing is when there is one factor in our experiment; blocking is when we have more than one factor.

5.6.1 Recap of group-to-group differences

We have already seen in the *univariate statistics section* (page 70) how to analyze this sort of data. We first calculate a pooled variance, then a z-value, and finally a confidence interval based on this z. Please refer back to that section to review the important assumptions we have to make to arrive at this equation:

$$s_P^2 = \frac{(n_A - 1)s_A^2 + (n_B - 1)s_B^2}{n_A - 1 + n_B - 1}$$

$$z = \frac{(\overline{x}_B - \overline{x}_A) - (\mu_B - \mu_A)}{\sqrt{s_P^2 \left(\frac{1}{n_A} + \frac{1}{n_B}\right)}}$$

$$-c_t \quad \leq \quad z \quad \leq \quad c_t$$

$$(\overline{x}_B - \overline{x}_A) - c_t \times \sqrt{s_P^2 \left(\frac{1}{n_A} + \frac{1}{n_B}\right)} \quad \leq \quad \mu_B - \mu_A \quad \leq \quad (\overline{x}_B - \overline{x}_A) + c_t \times \sqrt{s_P^2 \left(\frac{1}{n_A} + \frac{1}{n_B}\right)}$$

We consider the effect of changing from condition A to condition B to be a *statistically* significant effect when this confidence interval does not span zero. However, the width of this interval and how symmetrically it spans zero can cause us to come to a different, *practical* conclusion. In other words, we override the narrow statistical conclusion based on the richer information we can infer from the width of the confidence interval and the variance of the process.

5.6.2 Using linear least squares models

There's another interesting way that you can analyze data from an A versus B set of tests and get the identical result to the methods we showed in the section where *we made group-to-group comparisons* (page 70). In this method, instead of using a t-test, we use a least squares model of the form:

$$y_i = b_0 + g d_i$$

where y_i is the response variable d_i is an indicator variable. For example, $d_i = 0$ when using condition A and $d_i = 1$ for condition B. Build this linear model, and then examine the *confidence interval* for the coefficient g. The following R function uses the y-values from experiments under condition A and the values under condition B to calculate the least squares model:

```
lm_difference <- function(groupA, groupB)
{
    # Build a linear model with groupA = 0, and groupB = 1

    y.A <- groupA[!is.na(groupA)]
    y.B <- groupB[!is.na(groupB)]
    x.A <- numeric(length(y.A))
    x.B <- numeric(length(y.B)) + 1
    y <- c(y.A, y.B)
    x <- c(x.A, x.B)
    x <- factor(x, levels=c("0", "1"), labels=c("A", "B"))

    model <- lm(y ~ x)
    return(list(summary(model), confint(model)))
}

brittle <- read.csv('http://openmv.net/file/brittleness-index.csv')

# We developed the "group_difference" function in the Univariate section
group_difference(brittle$TK104, brittle$TK107)
lm_difference(brittle$TK104, brittle$TK107)
```

Use this function in the same way you did in *the carbon dioxide exercise in the univariate section* (page 91). For example, you will find when comparing TK104 and TK107 that $z = 1.4056$ and the confidence interval is $-21.4 \leq \mu_{107} - \mu_{104} \leq 119$. Similarly, when coding $d_i = 0$ for reactor TK104 and $d_i = 1$ for reactor TK107, we get the least squares confidence interval for parameter g: $-21.4 \leq g \leq 119$. This is a little surprising, because the first method creates a pooled variance and calculates a z-value and then a confidence interval. The least squares method builds a linear model, and then calculates the confidence interval using the model's standard error.

Both methods give identical results, but by very different routes.

5.6.3 The importance of randomization

We *emphasized in a previous section* (page 70) that experiments must be performed in random order to avoid any unmeasured, and uncontrolled, disturbances from impacting the system.

The concept of randomization was elegantly described in an example by Fisher in Chapter 2 of his book, *The Design of Experiments* (page 234). A lady claims that she can taste the difference in a cup of tea when the milk is added after the tea or when the tea is added after the milk. By setting up N cups of tea that contain either the milk first (M) or the tea first (T), the lady is asked to taste these N cups and make her assessment. Fisher shows that if the experiments are performed in random order, the actual set of decisions made by the lady are just one of many possible outcomes. He calculates all possibilities (we show how below), and then he calculates the probability of the lady's actual set of decisions being due to chance alone. If the lady has test score values better than by random chance, then there is a reasonable claim the lady is reliable.

Let's take a look at a more engineering-oriented example. We *previously considered* (page 91) the brittleness of a material made in either TK104 or TK107. The same raw materials were charged to each reactor. So, in effect, we are testing the difference due to using reactor TK104 or reactor TK107. Let's call them case A (TK104) and case B (TK107) so the notation is more general. We collected 20 brittleness values from TK104 and 23 values from TK107. We will only use the first 8 values from TK104 and the first 9 values from TK107 (you will see why soon):

Case A	254	440	501	368	697	476	188	525	
Case B	338	470	558	426	733	539	240	628	517

Fisher's insight was to create one long vector of these outcomes (length of vector = $n_A + n_B$) and randomly assign "A" to n_A of the values and "B" to n_B of the values. One can show that there are $\dfrac{(n_A + n_B)!}{n_A! n_B!}$ possible combinations. For example, if $n_A = 8$ and $n_B = 9$, then the number of unique ways to split these 17 experiments into two groups of 8 (A) and 9 (B) is 24,310 ways. For example, one way is BABB ABBA ABAB BAAB, and you would therefore assign the experimental values accordingly (B = 254, A = 440, B = 501, B = 368, A = 697, etc.).

Only one of the 24,310 sequences will correspond to the actual data printed in the above table. Although all the other realizations are possible, they are fictitious. We do this because the null hypothesis is that there is no difference between A and B. Values in the table could have come from either system.

So for each of the 24,310 realizations, we calculate the difference of the averages between A and B, $\bar{y}_A - \bar{y}_B$, and plot a histogram of these differences. This is shown below, together with a vertical line indicating the actual realization in the table. There are 4956 permutations that had a greater difference than the one actually realized; that is, 79.6% of the other combinations had a smaller value.

Had we used a formal test of differences where we pooled the variances, we would have found a z-value of 0.8435, and the probability of obtaining that value, using the t-distribution with $n_A + n_B - 2$ degrees of freedom, would be 79.3%. See how close they agree?
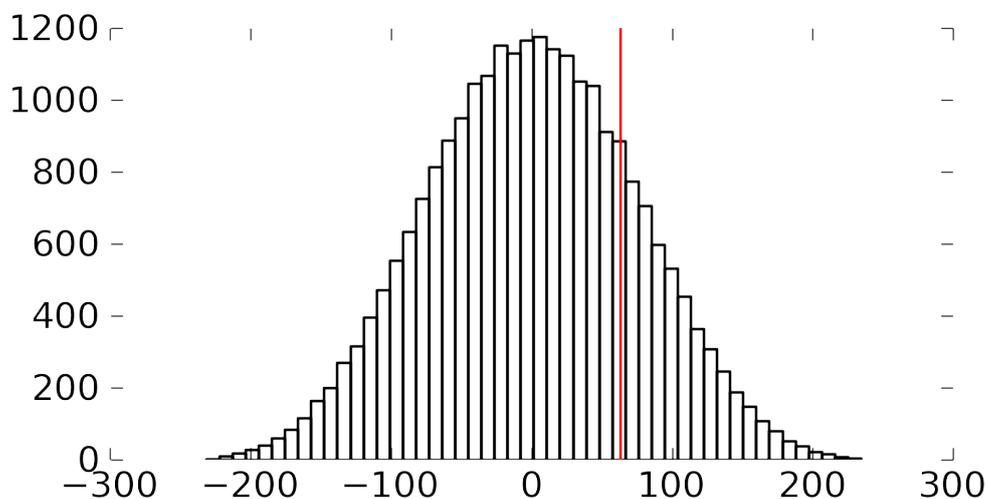
The figure shows the differences in the averages of A and B for the 24,310 realizations. The vertical line represents the difference in the average for the one particular set of numbers we measured in the experiment.

Recall that independence is required to calculate the z-value for the average difference and compare it against the t-distribution. By randomizing our experiments, we are able to guarantee that the results we obtain from using t-distributions are appropriate. Without randomization, these z-values and confidence intervals may be misleading.

The reason we prefer using the t-distribution approach over randomization is that formulating all random combinations and then calculating all the average differences as shown here is intractable. Even on my relatively snappy computer it would take 3.4 years to calculate all possible combinations for the complete dataset: 20 values from group A and 23 values from group B. (It took 122 seconds to calculate a million of them, so the full set of 960,566,918,220 combinations would take more than 3 years.)

5.7 Changing one single variable at a time (COST)

How do we go about running our experiments when there is more than one variable present that affects our outcome, y? In this section we describe **how not to do it**.

You will certainly have seen the recommendation that we must **c**hange **o**ne **s**ingle variable at a **t**ime (COST):

- Something goes wrong with a recipe: for example, the pancakes are not as fluffy as normal, or the muffins don't rise as much as they should. You are convinced it is the new brand of all-purpose flour you recently bought. You change only the flour the next time you make pancakes to check your hypothesis.

- University labs are notorious for asking you to change one variable at a time. The reason is that these labs intend for you to learn what the effect of a single variable is on some other variable (e.g.

change temperature in a distillation column to improve product purity). The labs teach you that this is good scientific procedure, which is fine if your goal is to only initially learn about a system, especially a new system that has never been explored.

However, when you want to *optimize and improve* a process, then a different style of experiments is required, where multiple factors are changed simultaneously.

We have known since the mid-1930s (almost 85 years), due to the work by *R. A. Fisher* (page 234), that changing **o**ne **f**actor **a**t a **t**ime (OFAT) is not an efficient way for experimentation. Note that OFAT is an alternative name for COST, and an example of OFAT is illustrated in the figure.

Consider a bioreactor where we are producing a particular enzyme. The yield, our outcome variable, is known to be affected by these six variables: dissolved oxygen level, agitation rate, reaction duration, feed substrate concentration, substrate type and reactor temperature. For illustration purposes, let's assume that temperature and feed substrate concentration are chosen, as they have the greatest effect on yield. The goal would be to maximize the yield.

The base operating point is 346 K with a feed substrate concentration of 1.5 g/L, marked with a circle in the figure below. At these conditions, we report a yield from the reactor of approximately 63%.
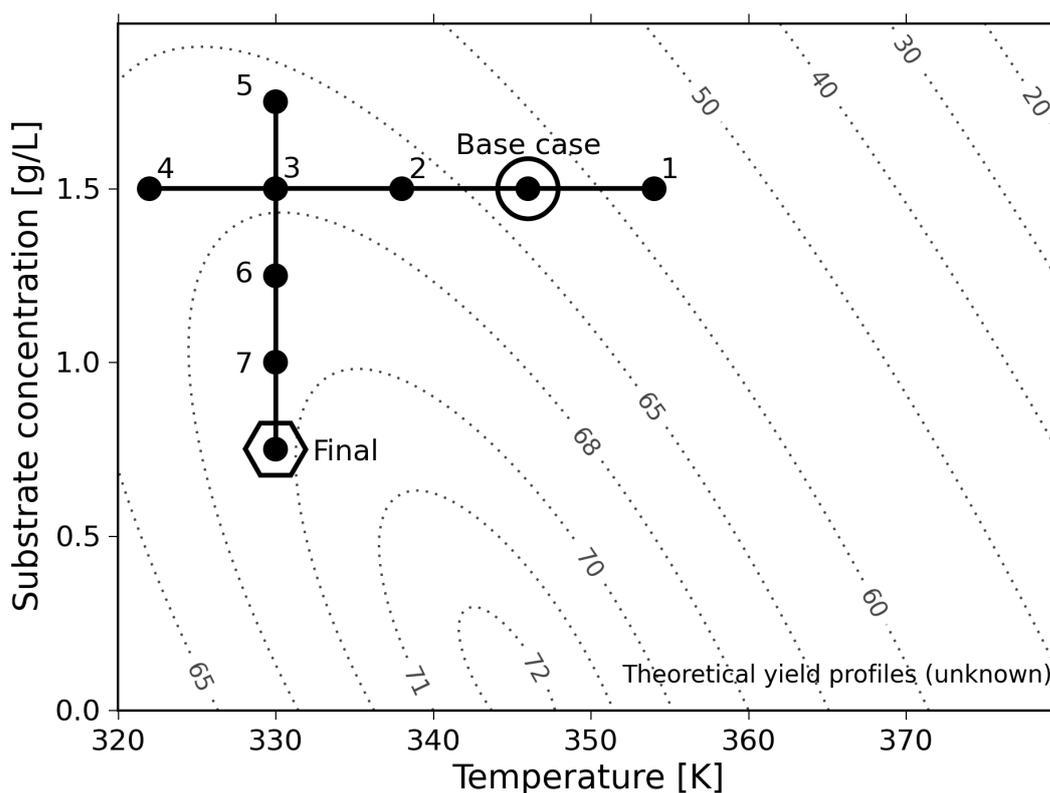
At this point, we start to investigate the effect of temperature. We decide to move up by 10 degrees to 356 K, marked as point 1. After running the experiment, we record a lower yield value than our starting point. So we go in the other direction and try temperatures at 338 K, 330 K and 322 K. We are happy that the yields are increasing, but experiment 4 shows a slightly lower yield. So we figure that we've reached a plateau in terms of the temperature variable. Our manager is pretty satisfied because we've boosted yield from 63% to around 67%. These four runs have cost us around $10,000 in equipment time and manpower costs so far.

5.7. Changing one single variable at a time (COST) 241

We now get approval to run four more experiments, and we decide to change the substrate feed concentration. But we're going to do this at the best temperature found so far, 330 K, at run 3. Our intuition tells us that higher feed concentrations should boost yield, so we try 1.75 g/L. Surprisingly, that experiment lowers the yield. There's likely something we don't understand about the reaction mechanism. Anyhow, we try the other direction, down to 1.25 g/L, and we see a yield increase. We decide to keep going, down to 1.0 g/L, and finally to 0.75 g/L. We see very little change between these last two runs, and we believe we have reached another plateau. Also, our budget of eight experimental runs is exhausted.

Our final operating point chosen is marked on the plot with a hexagon, at 330 K and 0.75 g/L. We're proud of ourselves because we have boosted our yield from 63% to 67%, and then from 67% to 69.5%. We have also learned something interesting about our process: the temperature appears to be negatively correlated with yield, and the substrate concentration is negatively correlated with yield. An unexpected observation!

The problem with this approach is that it leaves undiscovered values behind. Changing one single variable at a time leads you into thinking you've reached the optimum, when all you've done in fact is trap yourself at a suboptimal solution.

Furthermore, notice that we would have got a completely different outcome had we decided to first change substrate concentration, S, and then temperature, T. We would have likely landed closer to the optimum. This is very unsatisfactory: we cannot use methods to optimize our processes that depend on the order of experiments!

We have not yet even considered the effect of the other four variables of dissolved oxygen level, agitation rate, reaction duration and substrate type. We have suboptimally optimized the system in two dimensions, but there are in fact six dimensions. Although the OFAT (or COST) approach can get you close to the optimum in two variables, you have little to no hope of using this approach successfully with multiple factors.

Designed experiments, on the other hand, provide an efficient mechanism to learn about a system, often in fewer runs than the COST approach, and avoid misleading conclusions that might be drawn from the COST approach. Designed experiments are always run in random order – as we will presently see – and we will get the same result, no matter the order.

5.8 Full factorial designs

Video for
this
section.

In this section we learn how, and why, we should change more than one variable at a time. We will use factorial designs because

- We can visually interpret these designs, and see where to run future experiments;

- These designs require relatively few experiments; and

- They are often building blocks for more complex designs.

Most often we have two or more factors that affect our response variable, y. In this section we consider the case when these factors are at two levels. Some examples would be to operate at low or high pH, select long operating times or short operating times, use catalyst A or B and use mixing system A or B. The general guidance is to choose the low and high values at the edges of normal operation. It is **not** wise to use the minimum and maximum values that each factor could possibly have; they will likely be too extreme. We will see an example of this in the section on *saturated designs* (page 271).

5.8.1 Using two levels for two or more factors

Let's take a look at the mechanics of factorial designs by using our previous example where the conversion, y, is affected by two factors: temperature, T, and substrate concentration, S.

The range over which they will be varied is given in the table. This range was identified by the process operators as being sufficient to actually show a difference in the conversion, but not so large as to move the system to a totally different operating regime (that's because we will fit a linear model to the data).

Factor	Low level, $-$	High level, $+$
Temperature, T	338 K	354 K
Substrate level, S	1.25 g/L	1.75 g/L

1. Write down the factors that will be varied: T and S.

2. Write down the coded runs in standard order, also called Yates order, which alternates the sign of the first variable the fastest and the last variable the slowest. By convention we start all runs at their low levels and finish off with all factors at their high levels. There will be 2^k runs, where k is the number of variables in the design and the 2 refers to the number of levels for each factor. In this case, $2^2 = 4$ experiments (runs). **We perform the actual experiments in random order**, but always write the table in this standard order.

Experiment	T [K]	S [g/L]
1	$-$	$-$
2	$+$	$-$
3	$-$	$+$
4	$+$	$+$

3. Add an additional column to the table for the response variable. The response variable is a quantitative value, y, which in this case is the conversion measured as a percentage.

Experiment	Order	T [K]	S [g/L]	y [%]
1	3	$-$	$-$	69
2	2	$+$	$-$	60
3	4	$-$	$+$	64
4	1	$+$	$+$	53

Experiments were performed in random order; in this case, we happened to run experiment 4 first and experiment 3 last.

4. For simple systems you can visualize the design and results *as shown in the following figure* (page 243). This is known as a cube plot.
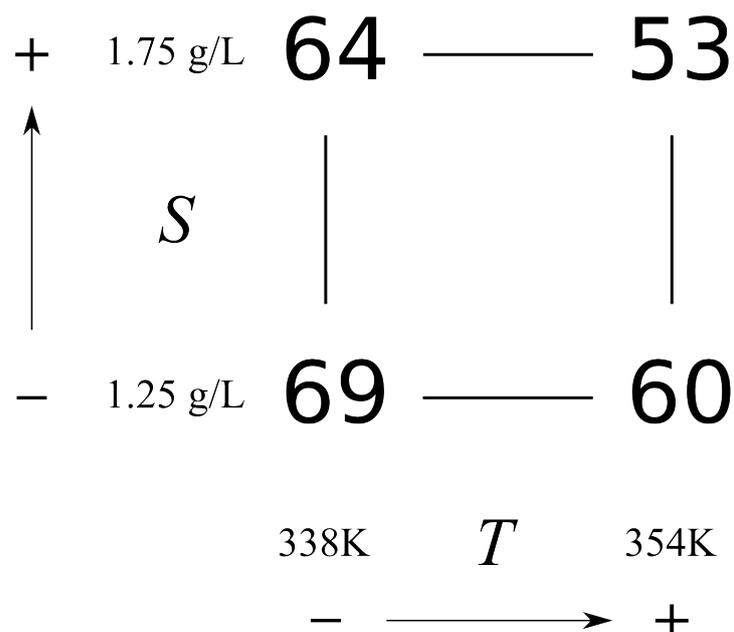
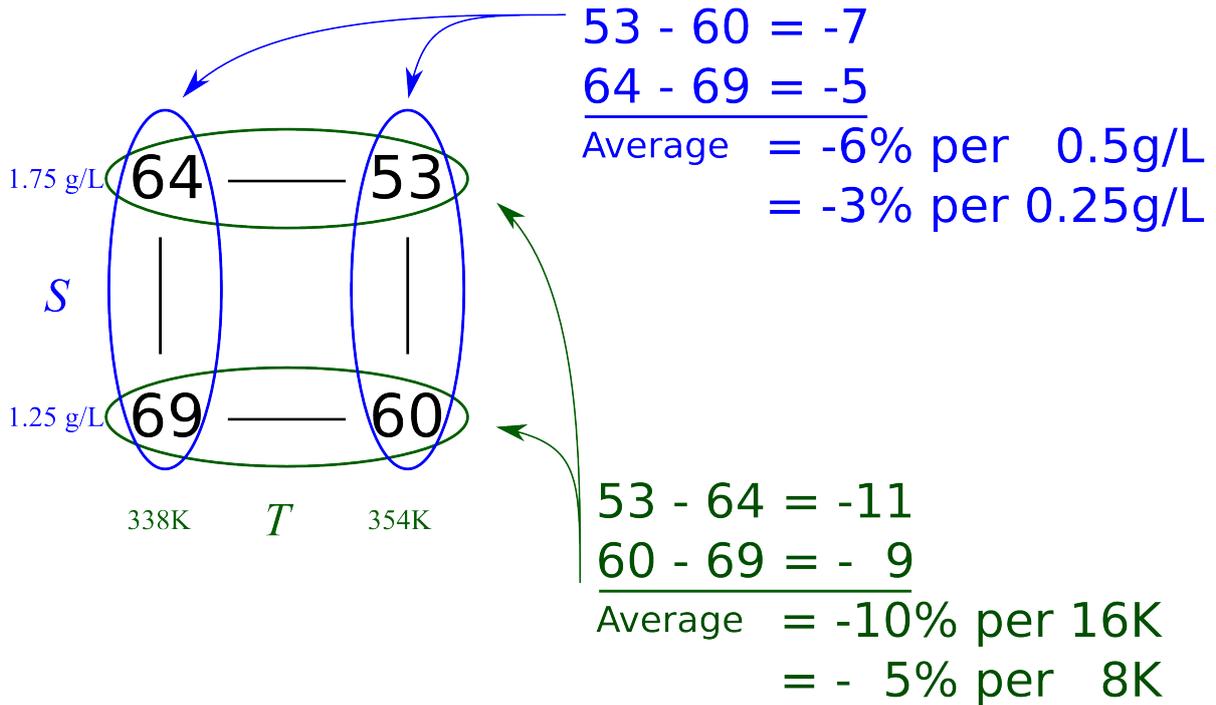### 5.8.2 Analysis of a factorial design: main effects

The first step is to calculate the main effect of each variable. The effects are considered, by convention, to be the difference from the high level to the low level. So the interpretation of a main effect is by how much the outcome, y, is adjusted when changing the variable.

Consider the two runs where S is at the $-$ level for both experiments 1 and 2. The only change between these two runs is the **temperature**, so the temperature effect is $\Delta T_{S-} = 60 - 69 = -9\%$ per $(354 - 338)$ K, that is, a -9% change in the conversion outcome per $+16$ K change in the temperature.

Runs 3 and 4 both have S at the $+$ level. Again, the only change is in the **temperature**: $\Delta T_{S+} = 53 - 64 = -11\%$ per $+16$ K. So we now have two temperature effects, and the average of them is a -10% change in conversion per $+16$ K change in temperature.

We can perform a similar calculation for the main effect of substrate concentration, S, by comparing experiments 1 and 3: $\Delta S_{T-} = 64 - 69 = -5\%$ per 0.5 g/L, while experiments 2 and 4 give $\Delta S_{T+} = 53 - 60 = -7\%$ per 0.5 g/L. So the average main effect for S is a -6% change in conversion for every 0.5 g/L change in substrate concentration. You should use the following *graphical method* (page 244) when calculating main effects from a cube plot by hand.

$$53 - 60 = -7$$
$$64 - 69 = -5$$
$$\text{Average} = -6\% \text{ per } 0.5\text{g/L}$$
$$= -3\% \text{ per } 0.25\text{g/L}$$

$$53 - 64 = -11$$
$$60 - 69 = -9$$
$$\text{Average} = -10\% \text{ per } 16\text{K}$$
$$= -5\% \text{ per } 8\text{K}$$

This visual summary is a very effective method of seeing how the system responds to the two variables. We can see the gradients in the system and the likely region where we can perform the next experiments to improve the bioreactor's conversion.

The following surface plot illustrates the true, but unknown, surface from which our measurements are taken. Notice the slight curvature on the edges of each face. The main effects estimated above are a linear approximation of the conversion over the region spanned by the factorial.
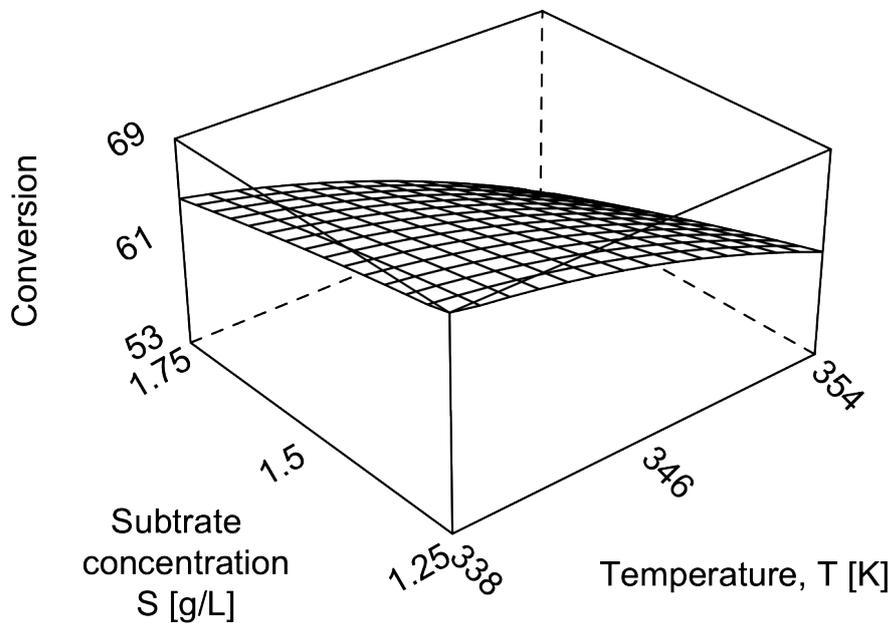
An interaction plot is an *alternative way to visualize these main effects* (page 245). Use this method when you don't have computer software to draw the surfaces. [We saw this earlier in the *visualization section* (page 1)]. We will discuss interaction plots more in the next section. Here is an illustration of one such plot for a system with little interaction.
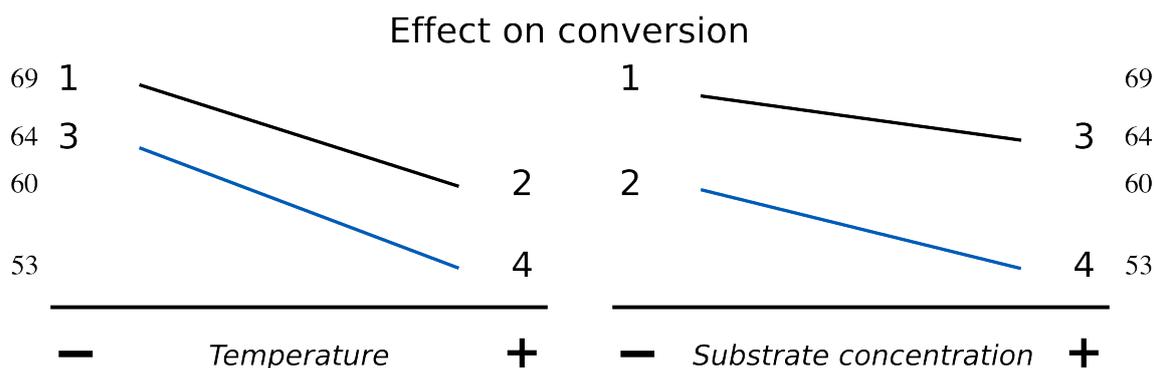
Effect on conversion

5.8.3 Analysis of a factorial design: interaction effects

We expect in many real systems that the main effect of temperature, T, for example, is different at other levels of substrate concentration, S. It is quite plausible for a bioreactor system that the main temperature effect on conversion is much greater if the substrate concentration, S, is also high, while at low values of S, the temperature effect is smaller.

We call this result an *interaction*, when the effect of one factor is different at different levels of the other factors. Let's give a practical, everyday example: assume your hands are covered with dirt or oil. We know that if you wash your hands with cold water, it will take longer to clean them than washing with hot water. So let factor **A** be the temperature of the water; factor **A** has a significant effect on the time taken to clean your hands.

Consider the case when washing your hands with cold water. If you use soap with cold water, it will take less time to clean your hands than if you did not use soap. It is clear that factor **B**, the categorical factor of using no soap vs some soap, will reduce the time to clean your hands.

Now consider the case when washing your hands with hot water. The time taken to clean your hands with hot water when you use soap is greatly reduced, far faster than any other combination. We say there is an interaction between using soap and the temperature of the water. This is an example of an interaction that works to help us reach the objective faster.

The effect of warm water enhances the effect of soap. Conversely, the effect is soap is enhanced by using warm water. So symmetry means that if soap interacts with water temperature, then we also know that water temperature interacts with soap.

In summary, interaction means the effect of one factor depends on the level of the other factor. In this example, that implies the effect of soap is different, depending on if we use cold water or hot water. Interactions are also symmetrical. The soap's effect is enhanced by warm water, and the warm water's effect is enhanced by soap.

Let's use a *different system here to illustrate* (page 246) interaction effects, but still using T and S as the variables being changed and keeping the response variable, y, as the conversion, shown by the contour lines.
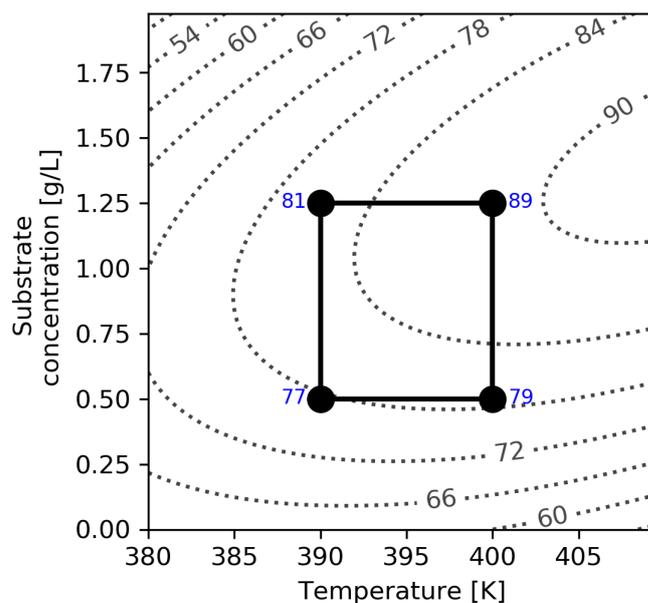
Experiment	T [K]	S [g/L]	y [%]
1	− (390 K)	− (0.5 g/L)	77
2	+ (400 K)	− (0.5 g/L)	79
3	− (390 K)	+ (1.25 g/L)	81
4	+ (400 K)	+ (1.25 g/L)	89

The main effect of temperature for this system is

- $\Delta T_{S-} = 79 - 77 = 2\%$ per 10 K

- $\Delta T_{S+} = 89 - 81 = 8\%$ per 10 K

which means that the average temperature main effect is 5% per 10 K.

Notice how different the main effect is at the low and high levels of S. So the average of the two is an incomplete description of the system. There is some other aspect to the system that we have not captured.

Similarly, the main effect of substrate concentration is

- $\Delta S_{T-} = 81 - 77 = 4\%$ per 0.75 g/L

- $\Delta S_{T-} = 89 - 79 = 10\%$ per 0.75 g/L

which gives the average substrate concentration main effect as 7% per 0.75 g/L.

The data may also be visualized using an *interaction plot* (page 247) here, showing a higher degree of interaction.
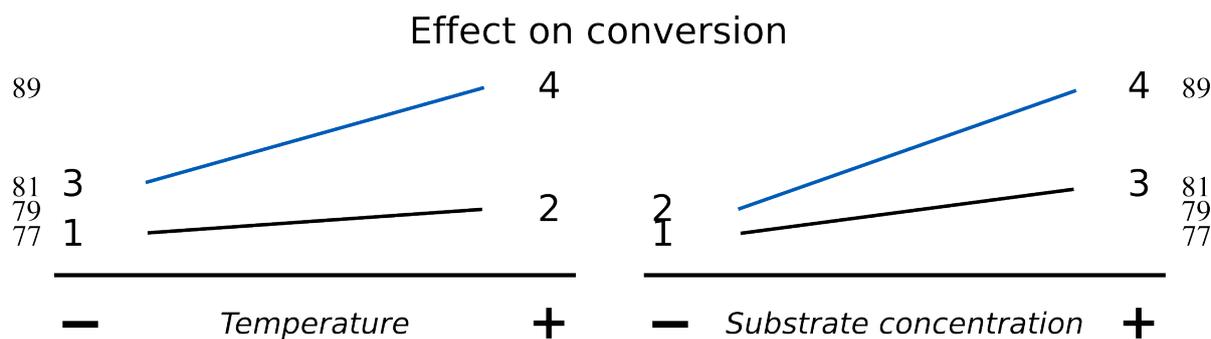
The lack of parallel lines is a clear indication of interaction. The temperature effect is stronger at high levels of S, and the effect of S on conversion is also greater at high levels of temperature. What is missing is an interaction term, given by the product of temperature and substrate. We represent this as $T \times S$ and call it the temperature-substrate interaction term.

This interaction term should be zero for systems with no interaction, which implies the lines are parallel in the interaction plot. Such systems will have roughly the same effect of T at both low and high values of S (and in between). So then, a good way to quantify interaction is by how different the main effect terms are at the high and low levels of the other factor in the interaction. The interaction *must* also be symmetrical: if T interacts with S, then S interacts with T by the same amount.

We can quantify the interaction of our current example in this way. For the T interaction with S:

- Change in conversion due to T at high S: $89 - 81 = +8$

- Change in conversion due to T at low S: $79 - 77 = +2$

- The half difference: $[+8 - (+2)]/2 = \mathbf{3}$

For the S interaction with T,

- Change in conversion due to S at high T: $89 - 79 = +10$

- Change in conversion due to S at low T: $81 - 77 = +4$

- The half difference: $[+10 - (+4)]/2 = \mathbf{3}$

A large, positive interaction term indicates that temperature and substrate concentration will increase conversion by a greater amount when both T and S are high. Similarly, these two terms will rapidly reduce conversion when they both are low.

We will get an improved appreciation for interpreting main effects and the interaction effect when we consider the analysis in the form of a linear, least squares model.

5.8.4 Analysis by least squares modelling

Let's review the *original system (the one with little interaction)* (page 243) and analyze the experimental data using a least squares model. We represent the original data here, with the baseline conditions:

Experiment	T [K]	S [g/L]	y [%]
Baseline	**346 K**	**1.50**	
1	$-$ (338 K)	$-$ (1.25 g/L)	69
2	$+$ (354 K)	$-$ (1.25 g/L)	60
3	$-$ (338 K)	$+$ (1.75 g/L)	64
4	$+$ (354 K)	$+$ (1.75 g/L)	53

It is standard practice to represent the data from designed experiments in a centered and scaled form: $\dfrac{\text{variable} - \text{center point}}{\text{range}/2}$. This gives the following values:

- $T_- = \dfrac{338 - 346}{(354 - 338)/2} = \dfrac{-8}{8} = -1$

- $S_- = \dfrac{1.25 - 1.50}{(1.75 - 1.25)/2} = \dfrac{-0.25}{0.25} = -1$

Similarly, $T_+ = +1$ and $S_+ = +1$, while the center points (baseline experiment) would be $T_0 = 0$ and $S_0 = 0$.

We will propose a least squares model that describes this system:

$$\begin{aligned}
\text{Population model}: & \quad y = \beta_0 + \beta_T x_T + \beta_S x_S + \beta_{TS} x_T x_S + \varepsilon \\
\text{Sample model}: & \quad y = b_0 + b_T x_T + b_S x_S + b_{TS} x_T x_S + e
\end{aligned}$$

We have four parameters to estimate and four data points. This means when we fit the model to the data, we will have no residual error, because there are no degrees of freedom left. If we had replicate experiments, we would have degrees of freedom to estimate the error, but more on that later. Writing the above equation for each observation,

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & T_- & S_- & T_-S_- \\ 1 & T_+ & S_- & T_+S_- \\ 1 & T_- & S_+ & T_-S_+ \\ 1 & T_+ & S_+ & T_+S_+ \end{bmatrix} \begin{bmatrix} b_0 \\ b_T \\ b_S \\ b_{TS} \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}$$

$$\begin{bmatrix} 69 \\ 60 \\ 64 \\ 53 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 \\ 1 & -1 & +1 & -1 \\ 1 & +1 & +1 & +1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_T \\ b_S \\ b_{TS} \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}$$

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$$

Where the last line is a more compact representation. Notice then that the *matrices from linear regression*

(page 185) are:

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 246 \\ -20 \\ -12 \\ -2 \end{bmatrix}$$

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 1/4 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} 246 \\ -20 \\ -12 \\ -2 \end{bmatrix} = \begin{bmatrix} 61.5 \\ -5 \\ -3 \\ -0.5 \end{bmatrix}$$
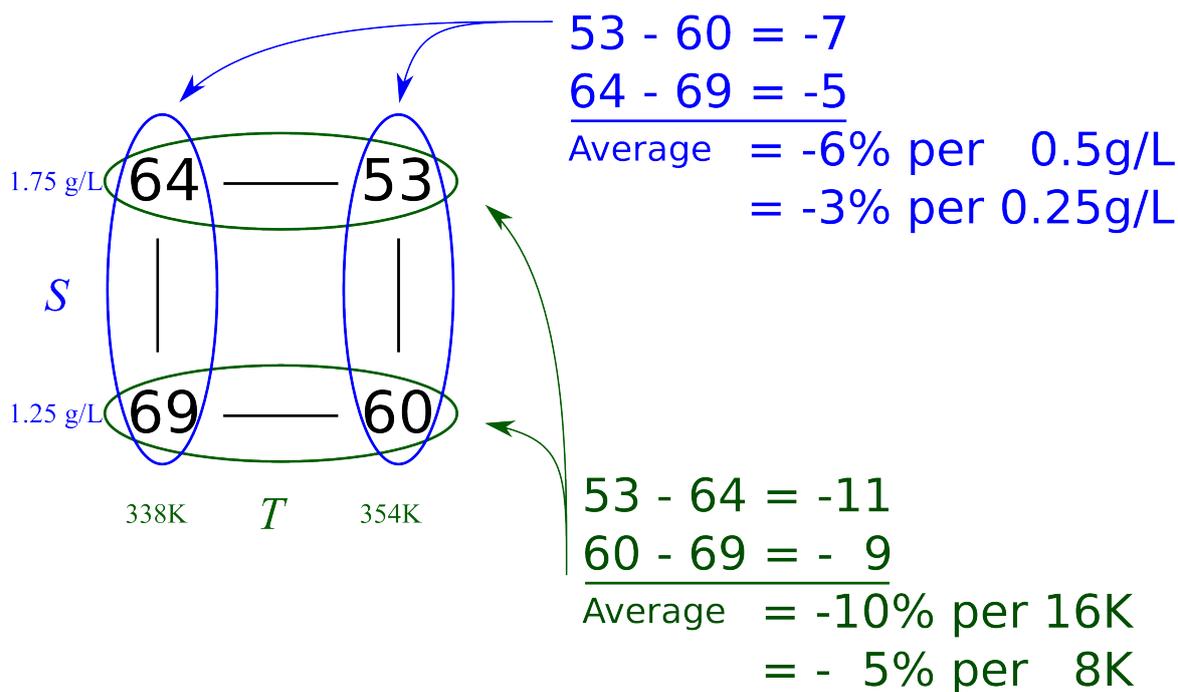
Video for this section.

Some things to note are (1) the orthogonality of $\mathbf{X}^T\mathbf{X}$ and (2) the interpretation of these coefficients.

1. Note how the $\mathbf{X}^T\mathbf{X}$ matrix has only zeros on the off-diagonals: it indicates that matrix $\mathbf{X}$ is orthogonal and confirms, algebraically, what we knew intuitively. The change we made in temperature, T, was independent of the changes we made in substrate concentration, S. This means that we can separately calculate *and interpret* the slope coefficients in the model.

2. What is the interpretation of, for example, $b_T = -5$? Recall, from the section on *linear regression interpretation* (page 189), that it is the effect of increasing the temperature by **1 unit**. In this case, the x_T variable has been normalized, but this slope coefficient represents the effect of changing x_T from 0 to 1, which in the original units of the variables is a change from 346 to 354 K, that is, an 8 K increase in temperature. It equally well represents the effect of changing x_T from -1 to 0: a change from 338 K to 346 K decreases conversion by 5%.

   Similarly, the slope coefficient for $b_S = -3$ represents the expected decrease in conversion when S is increased from 1.50 g/L to 1.75 g/L.

   Now contrast these numbers with those in the *graphical analysis done previously* (page 244) and repeated below. They are the same, as long as we are careful to interpret them as the change over **half the range**.

$$53 - 60 = -7$$
$$64 - 69 = -5$$
$$\overline{\text{Average}} = -6\% \text{ per } 0.5\text{g/L}$$
$$= -3\% \text{ per } 0.25\text{g/L}$$

$$53 - 64 = -11$$
$$60 - 69 = -9$$
$$\overline{\text{Average}} = -10\% \text{ per } 16\text{K}$$
$$= -5\% \text{ per } 8\text{K}$$

The 61.5 term in the least squares model is the expected conversion at the baseline conditions. Notice from the least squares equations how it is just the average of the four experimental values, even though we did not actually perform an experiment at the center.

Let's return to the *system with high interaction* (page 246) where the four outcome values in standard order were 77, 79, 81 and 89. Looking back, the baseline operation was $T = 395$ K and $S = \frac{1.25 - 0.5}{2} = 0.875$ g/L; you should prove to yourself that the least squares model is

$$y = 81.5 + 2.5x_T + 3.5x_S + 1.5x_T x_S$$

The interaction term can now be readily interpreted: it is the additional increase in conversion seen when both temperature and substrate concentration are at their high level. If T is at the high level and S is at the low level, then the least squares model shows that conversion is expected at $81.5 + 2.5 - 3.5 - 1.5 = 79$. The interaction term has *decreased* conversion by 1.5 units.
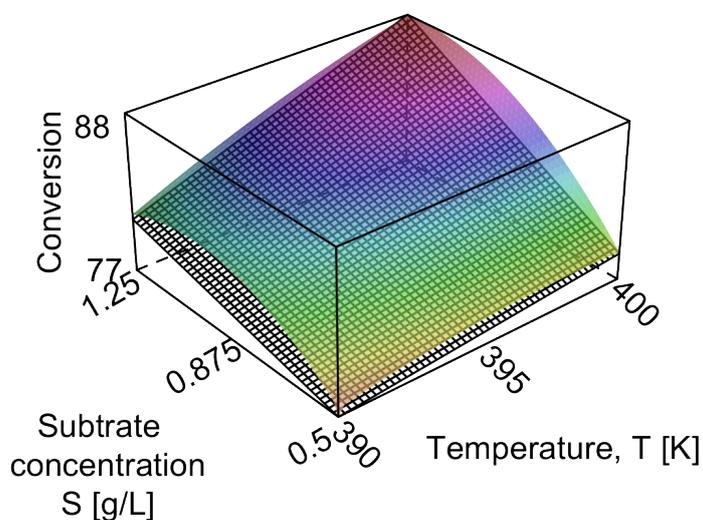
Finally, out of interest, the nonlinear surface that was used to generate the experimental data for the interacting system is coloured in the illustration. In practice we never know what this surface looks like, but we estimate it with the least squares plane, which appears below the nonlinear surface as black and white grids. The corners of the box are outer levels at which we ran the factorial experiments.

The corner points are exact with the nonlinear surface, because we have used the four values to estimate four model parameters. There are no degrees of freedom left, and the model's residuals are therefore zero. Obviously, the linear model will be less accurate away from the corner points when the true system is nonlinear, but it is a useful model over the region in which we will use it later in the *section on response surface methods* (page 278).

5.8.5 Example: design and analysis of a three-factor experiment

This example should be done by yourself. It is based on Question 19 in the exercises for Chapter 5 in Box, Hunter and Hunter (2nd edition).

The data are from a plastics molding factory that must treat its waste before discharge. The y-variable represents the average amount of pollutant discharged (lb per day), while the three factors that were varied were

- C = the chemical compound added (choose either chemical P or chemical Q)

- T = the treatment temperature (72 °F or 100 °F)

- S = the stirring speed (200 rpm or 400 rpm)

- y = the amount of pollutant discharged (lb per day)

Experiment	Order	C	T [°F]	S [rpm]	y [lb]
1	5	Choice P	72	200	5
2	6	Choice Q	72	200	30
3	1	Choice P	100	200	6
4	4	Choice Q	100	200	33
5	2	Choice P	72	400	4
6	7	Choice Q	72	400	3
7	3	Choice P	100	400	5
8	8	Choice Q	100	400	4

1. Draw a geometric figure that illustrates the data from this experiment.

2. Calculate the main effect for each factor by hand.

    For the **C effect**, there are four estimates of C:

    $$\frac{(+25) + (+27) + (-1) + (-1)}{4} = \frac{50}{4} = \mathbf{12.5}$$

    For the **T effect**, there are four estimates of T:

    $$\frac{(+1) + (+3) + (+1) + (+1)}{4} = \frac{6}{4} = \mathbf{1.5}$$

    For the **S effect**, there are four estimates of S:

    $$\frac{(-27) + (-1) + (-29) + (-1)}{4} = \frac{-58}{4} = \mathbf{-14.5}$$

3. Calculate the 3 two-factor interactions (2fi) by hand, recalling that interactions are defined as the half difference going from high to low.

For the **CT interaction**, there are two estimates of CT. Recall that interactions are calculated as the half difference going from high to low. Consider the change in C when

- T_{high} (at S high) = $4 - 5 = -1$
- T_{low} (at S high) = $3 - 4 = -1$

This gives a first estimate of $[(-1) - (-1)]/2 = 0$. Similarly,

- T_{high} (at S low) = $33 - 6 = +27$
- T_{low} (at S low) = $30 - 5 = +25$

gives a second estimate of $[(+27) - (+25)]/2 = +1$.

The average **CT** interaction is therefore $(0 + 1)/2 = \mathbf{0.5}$. You can interchange C and T and still get the same result.

For the **CS interaction**, there are two estimates of CS. Consider the change in C when

- S_{high} (at T high) = $4 - 5 = -1$
- S_{low} (at T high) = $33 - 6 = +27$

This gives a first estimate of $[(-1) - (+27)]/2 = -14$. Similarly,

- S_{high} (at T low) = $3 - 4 = -1$
- S_{low} (at T low) = $30 - 5 = +25$

gives a second estimate of $[(-1) - (+25)]/2 = -13$.

The average **CS** interaction is therefore $(-13 - 14)/2 = \mathbf{-13.5}$. You can interchange C and S and still get the same result.

For the **ST interaction**, there are two estimates of ST: $(-1 + 0)/2 = \mathbf{-0.5}$. Calculate in the same way as above.

4. Calculate the single three-factor interaction (3fi).

There is only a single estimate of CTS. The CT effect at high S is 0, and the CT effect at low S is $+1$. The CTS interaction is then $[(0) - (+1)]/2 = \mathbf{-0.5}$.

You can also calculate this by considering the CS effect at the two levels of T, or by considering the ST effect at the two levels of C. All three approaches give the same result.

5. Compute the main effects and interactions using matrix algebra and a least squares model.

Video for this section.

$$
\begin{bmatrix} 5 \\ 30 \\ 6 \\ 33 \\ 4 \\ 3 \\ 5 \\ 4 \end{bmatrix} = \begin{bmatrix} +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & +1 & -1 & +1 & -1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & -1 & +1 & -1 & +1 & -1 & -1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & -1 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_C \\ b_T \\ b_S \\ b_{CT} \\ b_{CS} \\ b_{TS} \\ b_{CTS} \end{bmatrix}
$$

$$\mathbf{y} = \mathbf{Xb}$$

6. Use computer software to build the following model and verify that:

$$y = 11.25 + 6.25x_C + 0.75x_T - 7.25x_S + 0.25x_Cx_T - 6.75x_Cx_S - 0.25x_Tx_S - 0.25x_Cx_Tx_S$$

Learning notes:

- The chemical compound could be coded either as (chemical P = −1, chemical Q = +1) or (chemical P = +1, chemical Q = −1). The interpretation of the x_C coefficient is the same, regardless of the coding.

- Just the tabulation of the raw data gives us some interpretation of the results. Why? Since the variables are manipulated independently, we can just look at the relationship of each factor to y, without considering the others. It is expected that the chemical compound and speed have a strong effect on y, but we can also see the **chemical** × **speed** interaction. You can see this last interpretation by writing out the full **X** design matrix and comparing the bold column, associated with the b_{CS} term, with the y column.

A note about magnitude of effects

In this text we quantify the effect as the change in response over *half the range* of the factor. For example, if the center point is 400 K, the lower level is 375 K and the upper level is 425 K, then an effect of `"−5"` represents a reduction in y of 5 units for every increase of 25 K in x.

We use this representation because it corresponds with the results calculated from least-squares software. Putting the matrix of −1 and +1 entries into the software as **X**, along with the corresponding vector of responses, y, you can calculate these effects as $\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{Xy}$.

Other textbooks, specifically Box, Hunter and Hunter, will report effects that are double ours. This is because they consider the effect to be the change from the lower level to the upper level (double the distance). The advantage of their representation is that binary factors (catalyst A or B; agitator on or off) can be readily interpreted, whereas in our notation, the effect is a little harder to describe (simply double it!).

The advantage of our methodology, though, is that the results calculated by hand would be the same as those from any computer software with respect to the magnitude of the coefficients and the standard errors, particularly in the case of duplicate runs and experiments with center points.

Remember: our effects are half those reported in Box, Hunter and Hunter, and in some other textbooks; our standard error would also be half of theirs. The conclusions drawn will always be the same, as long as one is consistent.

5.8.6 Assessing significance of main effects and interactions

When there are no replicate points, then the number of factors to estimate from a full factorial is 2^k from the 2^k observations. There are no degrees of freedom left to calculate the standard error or the confidence intervals for the main effects and interaction terms.

The standard error can be estimated if complete replicates are available. However, a complete replicate is onerous, because a complete replicate implies the entire experiment is repeated: system setup, running the experiment and measuring the result. Taking two samples from one actual experiment and measuring y twice is not a true replicate. That is only an estimate of the measurement error and analytical error.

Furthermore, there are better ways to spend our experimental budget than running complete replicate experiments – see the section on *screening designs* (page 271) later on. Only later in the overall experimental procedure should we run replicate experiments as a verification step and to assess the statistical significance of effects.

There are two main ways we can determine if a main effect or interaction is significant: by using a Pareto plot or the standard error.

Pareto plot

Note: This is a makeshift approach that is only applicable if all the factors are centered and scaled.

A full factorial with 2^k experiments has 2^k parameters to estimate. Once these parameters have been calculated, for example, by using a *least squares model* (page 248), then plot as shown the absolute value of the model coefficients in sorted order, from largest magnitude to smallest, ignoring the intercept term. Significant coefficients are established by visual judgement – establishing a visual cutoff by contrasting the small coefficients to the larger ones.
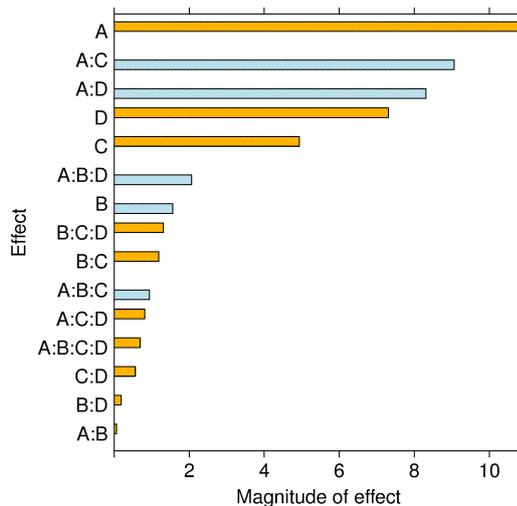
The example shown in the bar graph was from a full factorial experiment where the results for y in standard order were $y = [45, 71, 48, 65, 68, 60, 80, 65, 43, 100, 45, 104, 75, 86, 70, 96]$.

We would interpret that factors **A**, **C** and **D**, as well as the interactions of **AC** and **AD**, have a significant and causal effect on the response variable, y. The main effect of **B** on the response y is small, at least over the range that **B** was used in the experiment. Factor **B** can be omitted from future experimentation in this region, though it might be necessary to include it again if the system is operated at a very different point.

The reason why we can compare the coefficients this way, which is not normally the case with least squares models, is that we have both centered and scaled the factor variables. If the centering is at typical baseline operation, and the range spanned by each factor is that expected over the typical operating range, then we can fairly compare each coefficient in the bar plot. Each bar represents the influence of that term on y for a one-unit change in the factor, that is, a change over half its operating range.

Obviously, if the factors are not scaled appropriately, then this method will be error prone. However,

---

the approximate guidance is accurate, especially when you do not have a computer or if additional information required by the other methods (discussed below) is not available. It is also the only way to estimate the effects for *highly fractionated and saturated designs* (page 271).

Standard error: from replicate runs or from an external dataset

---

Note: It is often better to spend your experimental budget screening for additional factors rather than replicating experiments.

If there are more experiments than parameters to be estimated, then we have extra degrees of freedom. Having degrees of freedom implies we can calculate the standard error, S_E. Once S_E has been found, we can also calculate the standard error for each model coefficient, and then confidence intervals can be constructed for each main effect and interaction. And because the model matrix is orthogonal, the confidence interval for each effect is independent of the other. This is because the general confidence interval is $\mathcal{V}(\mathbf{b}) = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} S_E^2$, and the off-diagonal elements in $\mathbf{X}^T\mathbf{X}$ are zero.

For an experiment with n runs, and where we have coded our $\mathbf{X}$ matrix to contain -1 and $+1$ elements, and when the $\mathbf{X}$ matrix is orthogonal, the standard error for coefficient b_i is

$S_E(b_i) = \sqrt{\mathcal{V}(b_i)} = \sqrt{\dfrac{S_E^2}{\sum x_i^2}}$. Some examples:

- A 2^3 factorial where every combination has been repeated will have $n = 16$ runs, so the standard error for each coefficient will be the same, at $S_E(b_i) = \sqrt{\dfrac{S_E^2}{16}} = \dfrac{S_E}{4}$.

- A 2^3 factorial with three additional runs at the center point would have the following least squares representation:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$$

$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_{c,1} \\ y_{c,2} \\ y_{c,3} \end{bmatrix} =
\begin{bmatrix}
1 & A_- & B_- & C_- & A_-B_- & A_-C_- & B_-C_- & A_-B_-C_- \\
1 & A_+ & B_- & C_- & A_+B_- & A_+C_- & B_-C_- & A_+B_-C_- \\
1 & A_- & B_+ & C_- & A_-B_+ & A_-C_- & B_+C_- & A_-B_+C_- \\
1 & A_+ & B_+ & C_- & A_+B_+ & A_+C_- & B_+C_- & A_+B_+C_- \\
1 & A_- & B_- & C_+ & A_-B_- & A_-C_+ & B_-C_+ & A_-B_-C_+ \\
1 & A_+ & B_- & C_+ & A_+B_- & A_+C_+ & B_-C_+ & A_+B_-C_+ \\
1 & A_- & B_+ & C_+ & A_-B_+ & A_-C_+ & B_+C_+ & A_-B_+C_+ \\
1 & A_+ & B_+ & C_+ & A_+B_+ & A_+C_+ & B_+C_+ & A_+B_+C_+ \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} b_0 \\ b_A \\ b_B \\ b_C \\ b_{AB} \\ b_{AC} \\ b_{BC} \\ b_{ABC} \end{bmatrix} +
\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_{c,1} \\ e_{c,2} \\ e_{c,3} \end{bmatrix}
$$

And substituting in the values, using vector shortcut notation for **y** and **e**:

$$
\mathbf{y} = \begin{bmatrix} 1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 \\ 1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ 1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ 1 & +1 & +1 & -1 & +1 & -1 & -1 & -1 \\ 1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ 1 & +1 & -1 & +1 & -1 & +1 & -1 & -1 \\ 1 & -1 & +1 & +1 & -1 & -1 & +1 & -1 \\ 1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_A \\ b_B \\ b_C \\ b_{AB} \\ b_{AC} \\ b_{BC} \\ b_{ABC} \end{bmatrix} + \mathbf{e}
$$

Note that the center point runs do not change the orthogonality of **X** (verify this by writing out and computing the $\mathbf{X}^T\mathbf{X}$ matrix and observing that all off-diagonal entries are zeros). However, as we expect after having studied the section on *least squares modelling* (page 151), additional runs decrease the variance of the model parameters, $\mathcal{V}(\mathbf{b})$. In this case, there are $n = 2^3 + 3 = 11$ runs, so the standard error is decreased to $S_E^2 = \dfrac{\mathbf{e}^T\mathbf{e}}{11 - 8}$. However, the center points do not further reduce the variance of the parameters in $\sqrt{\dfrac{S_E^2}{\sum x_i^2}}$, because the denominator is still 2^k (**except for the intercept term**, whose variance is reduced by the center points).

Once we obtain the standard error for our system and calculate the variance of the parameters, we can multiply it by the critical t-value at the desired confidence level in order to calculate the confidence limit. However, it is customary to just report the standard error next to the coefficients, so that users can apply their own level of confidence. For example,

$$
\text{Temperature effect}, b_T = 11.5 \pm 0.707
$$
$$
\text{Catalyst effect}, b_K = 1.1 \pm 0.707
$$

Even though the confidence interval of the temperature effect would be $11.5 - c_t \times 0.707 \le \beta_T \le 11.5 + c_t \times 0.707$, it is clear that at the 95% significance level, the above representation shows the temperature effect is significant, while the catalyst effect is not ($c_t \approx 2$).

### Refitting the model after removing nonsignificant effects

After having established which effects are significant, we can exclude the nonsignificant effects and increase the degrees of freedom. (We do not have to recalculate the model parameters – why?) The residuals will be nonzero now, so we can then estimate the standard error and apply all the tools from least squares modelling to assess the residuals. Plots of the residuals in experimental order, against fitted values, q-q plots and all the other assessment tools from earlier are used, as usual.

Continuing the above example, where a 2^4 factorial was run, the response values in standard order were $y = [71, 61, 90, 82, 68, 61, 87, 80, 61, 50, 89, 83, 59, 51, 85, 78]$. The significant effects were from **A**, **B**, **D** and **BD**. Now, omitting the nonsignificant effects, there are only five parameters to estimate, including the intercept, so the standard error is $S_E^2 = \dfrac{39}{16 - 5} = 3.54$, with 11 degrees of freedom. The $S_E(b_i)$ value for all coefficients, except the intercept, is $\sqrt{\dfrac{S_E^2}{16}} = 0.471$, and the critical t-value at the

95% level is `qt(0.975, df=11)` = 2.2. So the confidence intervals can be calculated to confirm that these are indeed significant effects.

There is some circular reasoning here: postulate that one or more effects are zero and increase the degrees of freedom by removing those parameters in order to confirm the remaining effects are significant. Some general advice is to first exclude effects that are definitely small, and then retain medium-size effects in the model until you can confirm they are not significant.

Variance of estimates from the COST approach versus the factorial approach
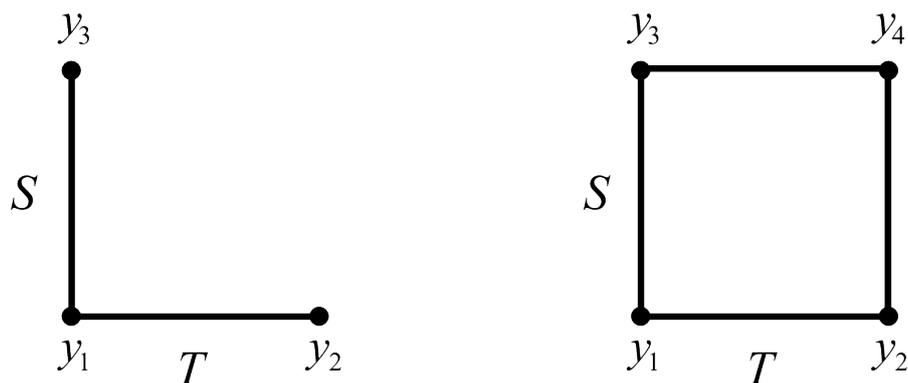
Finally, we end this section on factorials by illustrating their efficiency. Contrast the two cases: COST and the full factorial approach. For this analysis we define the main effect simply as the difference between the high and low values (normally we divide through by 2, but the results still hold). Define the variance of the measured y value as σ_y^2.

COST approach	Fractional factorial approach
The main effect of T is $b_T = y_2 - y_1$.	The main effect is $b_T = 0.5(y_2 - y_1) + 0.5(y_4 - y_3)$.
The variance is $\mathcal{V}(b_T) = \sigma_y^2 + \sigma_y^2$.	The variance is $\mathcal{V}(b_T) = 0.25(\sigma_y^2 + \sigma_y^2) + 0.25(\sigma_y^2 + \sigma_y^2)$.
So $\mathcal{V}(b_T) = 2\sigma_y^2$.	And $\mathcal{V}(b_T) = \sigma_y^2$.

Not only does the factorial experiment estimate the effects with much greater precision (lower variance), but the COST approach cannot estimate the effect of interactions, which is incredibly important, especially as systems approach optima that are on ridges (see the contour plots earlier in this section for an example).

Factorial designs make each experimental observation work twice.

5.8.7 Summary so far

- The factorial experimental design is intentionally constructed so that each factor is independent of the others. There are 2^k experiments for k factors.

  - This implies the $\mathbf{X}^T\mathbf{X}$ matrix is easily constructed (a diagonal matrix, with a value of 2^k for each diagonal entry).

  - These coefficients have the lowest variability possible: $(\mathbf{X}^T\mathbf{X})^{-1}S_E^2$.

  - We have uncorrelated estimates of the slope coefficients in the model. That is, we can be sure the value of the coefficient is unrelated to the other values.

- However, we still need to take the usual care in *interpreting* the coefficients. The usual precaution, using the example below, is that the temperature coefficient b_T is the effect of a one-degree change, holding all other variables constant. That's not possible if b_{TS}, the interaction between T and S, is significant: we cannot hold the TS constant while changing b_T.

$$y = b_0 + b_T x_T + b_S x_S + b_{TS} x_T x_S + e$$

  We cannot interpret the main effects separately from the interaction effects when we have significant interaction terms in the model. Also, if you conclude the interaction term is significant, then you must also include all main factors that make up that interaction term in the model.

  For another example, with an interpretation, please see Box, Hunter and Hunter (2nd edition), page 185.

- Factorial designs use the collected data much more efficiently than one-at-a-time experimentation. As shown in *the preceding section* (page 258), the estimated variance is halved when using a factorial design compared to a COST approach.

- A small or zero effect from an x variable to the y response variable implies the y is insensitive to that x. This is desirable in some situations. It means we can adjust that x without affecting y, sometimes stated as "the y is robust to changes in x".

5.8.8 Example: analysis of systems with 4 factors

In the prior sections you have seen how to analyze experiments with 2 factors and 3 factors. The logic to analyze systems with 4 or more factors proceeds in exactly the same way. The video here shows how to go about this.

5.9 Fractional factorial designs

When there are many factors that we have identified as being potentially important, then the 2^k runs required for a full factorial can quickly become large and too costly to implement.

For example, you are responsible for a cell-culture bioreactor at a pharmaceutical company and there is a drive to minimize the production of an inhibiting by-product. Variables thought to be related with this by-product are: two types of **T** = temperature profile (T_-: a slow ramp over time or T_+ a fast initial ramp then constant temperature), the **D** = dissolved oxygen at a low and high level, the **A** = agitation rate at a slow and faster speed, **P** = pH at a low and high level, and two blends of **S** = substrate type. These five factors, at two levels, require $2^5 = 32$ runs. It would take almost a year to

collect the data at all the combinations of **T**, **D**, **A**, **P** and **S** if each experiment requires 10 days (typical in this industry), and if parallel reactors are not available.

Furthermore, we are probably interested in only the 5 main effects and 10 two-factor interactions (2fi). The remaining 10 three-factor interactions, 5 four-factor interactions, a single five-factor interaction are likely not too important either, at least initially. A full factorial would estimate 32 effects, even if we likely only interested in at most 16 of them (5 main effects + 10 of the 2fi's + 1 intercept).

Running a half fraction, or quarter fraction, of the full set will allow us to estimate the main effects and two-factor interactions (2fi) in many cases, at the expense of *confounding* the higher interactions. [We will explain exactly what is meant by that term *confounding* later on, but for now you can interpret it is as 'confused with'].

For many real systems it is the main effects that are mostly of interest, and at most two-factor interactions. Very very seldom do three-factor interactions occur, nor are they often of practical significance. As such, we are willing to allow some confounding (confusing) with these factors.

So let's move into this section where we show how to construct and analyze these fractional factorials, which are tremendously useful when screening many variables - especially for a first-pass at experimenting on a new system. They are used when you, as the experimenter, suspect that you have more factors on your list than are actually needed. Which ones can you eliminate?

5.9.1 Half fractions

A half fraction has $\frac{1}{2}2^k = 2^{k-1}$ runs. But which half of the runs do we omit? Let's use an example of a 2^3 full factorial which has 8 experiments. The half-fraction would have 4 runs. Since 4 runs can be represented by a 2^2 factorial, we start by writing down the usual 2^2 factorial for any 2 factors (we will use **A** and **B** in this example, but you can use any 2 factors). Now create the 3rd factor as the product of the first two, **C** = **AB**.

Experiment	A	B	C = AB
1	−	−	+
2	+	−	−
3	−	+	−
4	+	+	+

So this is our half-factorial designed experiment in 3 factors, but it only requires 4 experiments as shown by the open points in the figure. The experiments given by the solid points are not run.
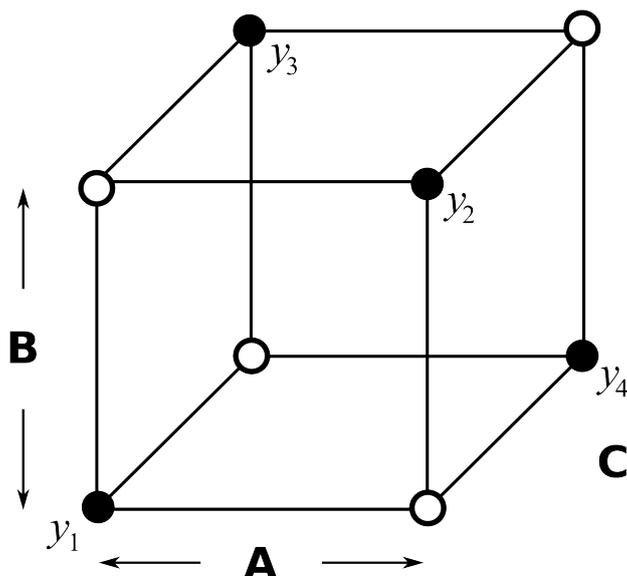
What have we lost by running only half of the full factorial? Let's write out the full design and matrix of all interactions, then construct the **X** matrix for the least squares model.

Experiment	A	B	C	AB	AC	BC	ABC	Intercept
1	−	−	+	+	−	−	+	+
2	+	−	−	−	−	+	+	+
3	−	+	−	−	+	−	+	+
4	+	+	+	+	+	+	+	+

Before even constructing the **X**-matrix, you can see that **A=BC**, and that **B=AC** and **C=AB** (this last association was intentional), and intercept **I=ABC**.

The least squares model would be:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$$

$$y_i = b_0 + b_A x_A + b_B x_B + b_C x_C + b_{AB} x_{AB} + b_{AC} x_{AC} + b_{BC} x_{BC} + b_{ABC} x_{ABC} + e_i$$

$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} =
\begin{bmatrix}
1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\
1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\
1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\
1 & +1 & +1 & +1 & +1 & +1 & +1 & +1
\end{bmatrix}
\begin{bmatrix} b_0 \\ b_A \\ b_B \\ b_C \\ b_{AB} \\ b_{AC} \\ b_{BC} \\ b_{ABC} \end{bmatrix} +
\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}
$$

The **X** matrix is not *orthogonal* (page 250) anymore, because one or more columns are exactly identical to another column, also known as collinearity. Notice that 4 of the columns are the same as the other 4: we have perfect collinearity between 4 pairs of columns. Also note this system is underdetermined as there are more unknowns than equations.

For these reasons the least squares model cannot be solved by inverting the $\mathbf{X}^T \mathbf{X}$ matrix. Prove it to yourself by using this code:

```
─────────────────────────────── R code ───────────────────────────────
int <- c(1, 1, 1, 1)
A <- c(-1, +1, -1, +1)
B <- c(-1, -1, +1, +1)
C = A * B
X = cbind(int, A, B, C, AB=A*B,
          AC=A*C, BC=B*C, ABC=A*B*C)

XtX <- t(X) %*% X
print("The X'X matrix is = ")
print(XtX)

print('Calculate the inverse (it will fail!)')
solve(XtX)

# We cannot, since the determinant is 0:
det(XtX)
```

To resolve this problem we can reformulate the model to obtain independent columns, grouping together the columns which are identical. There are now 4 equations and 4 unknowns:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 \\ 1 & -1 & +1 & -1 \\ 1 & +1 & +1 & +1 \end{bmatrix} \begin{bmatrix} b_0 + b_{ABC} \\ b_A + b_{BC} \\ b_B + b_{AC} \\ b_C + b_{AB} \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}$$

Writing it this way clearly shows how the main effects and two-factor interactions are *confounded*.

- $b_0 + b_{ABC} = \widehat{\beta}_0 \rightarrow \mathbf{I + ABC}$

- $b_A + b_{BC} = \widehat{\beta}_A \rightarrow \mathbf{A + BC}$: this implies β_A estimates the **A** main effect and the **BC** interaction

- $b_B + b_{AC} = \widehat{\beta}_B \rightarrow \mathbf{B + AC}$

- $b_C + b_{AB} = \widehat{\beta}_C \rightarrow \mathbf{C + AB}$

It means we cannot separate, for example, the effect of the **BC** interaction from the main effect of **A**: the least-squares coefficient is a sum of both these effects. Similarly for the other pairs. This is why we say the factor **A** is confounded with the two-factor interaction **BC**. Factor **B** is confounded with **AC**, and factor **C** is confounded with **AB**. Also the intercept is not a pure estimate of the intercept; it is confounded with the 3-factor interaction **ABC**.

This is what we have lost by running a half-fraction: the benefit of doing fewer experiments is paid by the price of confounding within the factors we estimate.

We introduce the terminology that **A** is an alias for **BC**, similarly that **B** is an alias for **AC**, *etc*, because we cannot separate these aliased effects.

5.9.2 Generators and defining relationships

Calculating which main effects and two-factor interactions will be confounded with each other, called the confounding pattern, can be tedious for larger values of k. Here we introduce an easy way to calculate the confounding pattern.

Recall for the half-fraction of a 2^k factorial that the first $k-1$ main factors are written down, then the final k^{th} factor is *generated* from the product of the previous $k-1$ factors. Consider the case of a 2^4 half fraction with factors **A**, **B**, **C** and **D**. The half-fraction has $\frac{1}{2}2^4 = 2^3 = 8$ experiments, so we write this 2^3 factorial in factors **A**, **B**, and **C**, then set:

$$\textbf{D = ABC}$$

This is called the *generating relation* for the design. Here are some rules when working with this notation:

- A factor multiplied by itself is the identity, or intercept column: **A** × **A** = **I**, **B** × **B** = **I**, etc. Think about that: if you look at the previous designs we have written out, this makes sense. Any column multiplied by itself is equal to a column of ones.

- A factor multiplied by a column of ones is equal to itself. For example: **D** × **I** = **D**

- The intercept **I** is simply a column of ones, which is what the intercept column is. And for emphasis: **I** × **I** = **I**.

- You can substitute in the *generating relation* of **D = ABC**, and like with an algebraic equation, we can multiply both sides by **D** to get **D** × **D** = **ABC** × **D**, which simplifies to **I** = **ABCD**. Another way to get this same result it to substitute the generating relationship in twice: **ABC** × **D** = **ABC** × **ABC** = **AABBCC** = **I I I** = **I** = **ABCD**.

This last part, **I = ABCD**, is called the *defining relation* for this design. Notice that we started with the *generating relation* and simplified it by multiplying the terms in that relationship with each other. Since there were two terms, **ABC** and **D**, we multiplied them, and ended up with **I = ABCD**.

This is our defining relationship for this design:

$$\textbf{I = ABCD}$$

We will discuss this topic again later with more examples. The main point though is that the effects which are aliased (confounded) with each other can be found quickly by multiplying the effect we are interested in by the defining relationship. For example, if we wanted to know what the main effect **A** would be confounded with in this 2^{4-1} half fraction we should multiply **A** by the defining relationship as in

$$\textbf{A = A} \times \textbf{I = A} \times \textbf{ABCD = BCD}$$

indicating that **A** is aliased with the 3-factor interaction **BCD**. What is the aliasing for these effects:

- What is main effect **B** aliased with? (*Answer*: **ACD**)

- What is the 2fi **AC** aliased with? (*Answer*: **BD**)

Another example:

Returning back to the 2^{3-1} half fraction in the [*previous section*](#) (page 260), use the generating relation to verify the aliasing of main-effects and two-factor interactions derived earlier by hand.

- First calculate the defining relationship. It is **I** = …..

- Aliasing for **A**? (*Answer*: **BC**)

- Aliasing for **B**? (*Answer*: **AC**)

- Aliasing for **C**? (*Answer*: **AB**: recall this is how we generated that half fraction)

- Aliasing for the intercept term, **I**? (*Answer*: **ABC**)

Yet another example:

Which aliasing (confounding) would occur if you decided for a 2^{4-1} design to generate the half-fraction by using the 2-factor interaction term **AC** rather than the 3-factor interaction term **ABC**.

- First write out your generating relationship: **D** = **AC**

- Now calculate the defining relationship: **I** = ...

- Aliasing for **A**? (*Answer*: **CD**)

- Aliasing for **B**? (*Answer*: **ABCD**)

- Aliasing for **C**? (*Answer*: **AD**)

Why is this a poorer choice than using **D = ABC** to generate the half-fraction? *Answer*: the main effects of **A** and **C** which could be important, are aliased with 2fi. Had we generated the design with the usual 3fi term, **ABC**, the main effects would only be aliased with three-factor interactions (3fi).

5.9.3 Generating the complementary half-fraction

Returning to our example in the *previous section* (page 260) of a half-fraction from a full 2^3 factorial, and imagine the half-fraction of 4 runs was completed. Imagine that all 3 factors showed significant effect on the outcome. Further, imagine that one of the factors actually gave a direction opposite to what was expected. This is really interesting, and unexpected new knowledge.

The original generator was **C = AB** and the defining relation was **I = ABC**; so factor **C** was aliased with the 2fi of **AB**. If it was factor **C** that had an opposite sign, it could be due to **C**, or due to **AB**. So you wish to complete the full-factorial and run the other half fraction to find out. This will help clarify that interesting factor, because it will remove the aliasing when you then analyze all 8 data points together.

The defining relation for the complementary half-fraction is **I = -ABC**, or multiply both sides by **C** to equivalently obtain **IC = C = -AB**. This shows the complementary half fraction is in fact generated by **C = -AB**, while the original half-fraction was generated by **C = AB**. This is a general rule that applies to half-fractions.

Let's return to the table in the *previous section* (page 260) and generate the other 4 runs from that **C = -AB** defining relationship:

Experiment	A	B	C = − AB
5	−	−	−
6	+	−	+
7	−	+	+
8	+	+	−
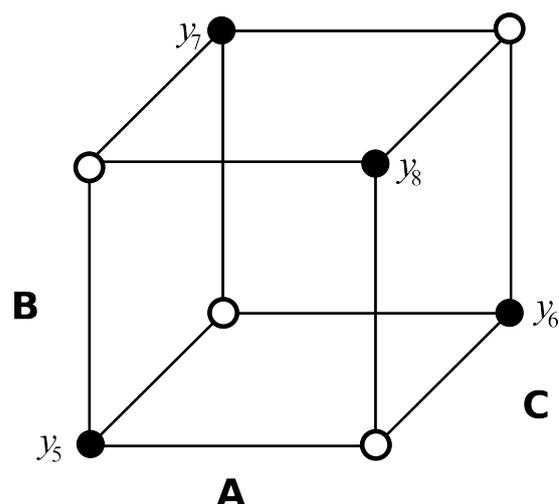
After running these additional 4 experiments shown (in random order of course) we have a complete set of 8 runs. Analyzing the data together we can calculate the main effects and two-factor interactions without aliasing because we are back to the usual full factorial of 2^3 runs. Confirm it for yourself visually in the plot alongside.

So we see that we can always complete our half-fraction by creating a complementary fraction. This complimentary fraction is found by flipping the sign on the generating factor. For example, changing the sign from **C = AB** to **-C = AB**. In the illustration this is equivalent to running the 4 experiments at the closed circles.

5.9.4 Generators: to determine confounding due to blocking

Generators are also great for determining the blocking pattern. Recall the case described earlier where we only had enough material to run two sets of 4 experiments to complete our 2^3 full factorial. An unintended disturbance could have been introduced by running the first half-fraction on different materials to the second half-fraction. We to confound the two blocks of experiments with the 3-factor interaction, **ABC**. So if there is an effect due to the blocks (i.e. the raw materials) or if there truly was a 3-factor interaction, it will show up as a significant coefficient for b_{ABC}.

So *in general* if you run a full 2^k factorial in two blocks you should create a 2^{k-1} half fraction to run as the first block, and then run the other block on the complementary half-fraction. You should always confound your block effect on the highest possible interaction term. Then block 1 runs will have that highest interaction factor with all positive signs, and block 2 will have all negative signs for that interaction factor.

Here are the block generators you can use when splitting a 2^k factorial in 2 blocks:

k	Design	Block 1 defining relation	Block 2 defining relation
3	2^{3-1}	I=ABC	I=-ABC
4	2^{4-1}	I=ABCD	I=-ABCD
5	2^{5-1}	I=ABCDE	I=-ABCDE

5.9.5 Highly fractionated designs: beyond half-fractions

Running a half-fraction of a 2^k factorial is not the only way to reduce the number of runs. In general, we can run a 2^{k-p} fractional factorial. A system with 2^{k-1} is called a *half fraction*, while a 2^{k-2} design is a quarter fraction, and so on.

The purpose of a fractionated design is to reduce the number of experiments when your budget - or time - does not allow you to complete a full factorial. Also, the full factorial is often not required, especially when k is greater than about 4, since the higher-order interaction terms are almost always insignificant. If we have a budget - or time - for only 8 experiments, then our options are to run a:

- 2^3 full factorial on **3 factors**

- 2^{4-1} half fraction, investigating **4 factors**

- 2^{5-2} quarter fraction looking at the effects of **5 factors**

- 2^{6-3} fractional factorial with **6 factors**, or a

- 2^{7-4} fractional factorial with **7 factors**.

At the early stages of our work we might prefer to screen many factors, $k = 6$ or 7, accepting a very complex confounding pattern, because we are uncertain which factors actually affect our response. Later, as we are optimizing our process, particularly as we approach an optimum, then the 2 factor and perhaps 3-factor interactions are more dominant. So investigating and calculating these effects more accurately and more precisely becomes important and we have to use full factorials. But by then we have hopefully identified much fewer factors k than what we started off with.

So this section is concerned with the trade-offs as we go from a full factorial with 2^k runs to a highly fractionated factorial, 2^{k-p}.

Example 1

You have identified 7 factors that affect your response.

- What is the smallest number of runs that can be performed to screen for the main effects?

  A 2^{7-p} fractional factorial would have 64 ($p = 1$), 32 ($p = 2$), 16 ($p = 3$), or 8 ($p = 4$) runs. The last case is the smallest number that can be used to estimate the intercept and 7 main effects (8 data points, 8 unknowns).

- What would be the generators and defining relation for this $2^{7-4} = 8$ run experiment and what is the aliasing structure?

  1. Assign **A**, **B**, ... **G** as the 7 factor names. Since there are 8 runs, start by writing out the first 3 factors, **A**, **B**, and **C**, in the usual full factorial layout for these $2^3 = 8$ runs.

  2. Next we assign the remaining factors to the highest-possible interaction terms from these 3 factors. Since we've already assigned **A**, **B** and **C** we only have to assign the other 4 factors, **D**, **E**, **F**, and **G**. Pick the 4 interaction terms that we are least interested in. In this particular example, we have to use all (saturate) the interaction terms.

     - **D = AB**

     - **E = AC**

     - **F = BC**

     - **G = ABC**

Record the experimental results for y in the last column.

Experiment	A	B	C	D=AB	E=AC	F=BC	G=ABC	y
1	−	−	−	+	+	+	−	77.1
2	+	−	−	−	−	+	+	68.9
3	−	+	−	−	+	−	+	75.5
4	+	+	−	+	−	−	−	72.5
5	−	−	+	+	−	−	+	67.9
6	+	−	+	−	+	−	−	68.5
7	−	+	+	−	−	+	−	71.5
8	+	+	+	+	+	+	+	63.7

1. So the 4 generators we used to create, or generate, the experimental design are **I = ABD**, **I = ACE**, **I = BCF** and **I = ABCG**. The generator terms such as **ABD** and **ACE** are called *"words"*.

2. The *defining relationship* is a sequence of words which are all equal to **I**. The defining relation is found from the product of all possible generator combinations, and then simplified to be written as **I = ...**.

   The rule is that a 2^{k-p} factorial design is produced by p generators and has a defining relationship of 2^p words. So in this example there are p generators and $2^p = 2^4 = 16$ words in our defining relation. They are:

   – Intercept: **I** [1]

   – Each generator combined with **I**: **I = ABD = ACE = BCF = ABCG** [2,3,4,5]

   – Two combinations of generators: **I = BDCE = ACDF = CDG = ABEF = BEG = AFG** [6 to 11]

   – Three combinations of generators: **I = DEF = ADEG = CEFG = BDFG** [12 to 15]

   – Four combinations of generators: **I = ABCDEFG** [16]

   The 16 words in the defining relationship are written as: **I = ABD = ACE = BCF = ABCG = BCDE = ACDF = CDG = ABEF = BEG = AFG = DEF = ADEG = CEFG = BDFG = ABCDEFG**. The shortest length word, not counting the intercept, has 3 letters. We will refer to this later as the *design's resolution* (page 269).

3. The aliasing or confounding pattern for any desired effect can be calculated by multiplying the defining relationship by that effect. Let's take **A** as an example, below, and multiply it by the 16 words in the defining relation:

   AI = BD = CE = ABCF = BCG = ABDCE = CDF = ACDG = BEF = ABEG = FG = ADEF = DEG = ACEFG = ABDFG = BCDEFG. So our factor **A** estimate is not just factor **A**, it is also a combined estimate of:

   $$\widehat{\beta}_A \rightarrow \text{A} + \textbf{BD} + \textbf{CE} + \text{ABCF} + \text{BCG} + \text{ABCDE} + \text{CDF} + \text{ACDG} + \text{BEF} + \text{ABEG} + \textbf{FG} + \text{ADEF} + \text{DEG} + \text{ACEFG} + \text{ABDFG} + \text{BCDEFG}.$$

   $$\widehat{\beta}_A \approx \text{A} + \textbf{BD} + \textbf{CE} + \textbf{FG} + \dots$$

   So by performing 8 runs instead of the full 2^7, we confound the main effects with a large number of 2-factor and higher interaction terms. In particular, the main effect of **A** is

confounded here with the **BD**, **CE** and **FG** two-factor interactions. Any 3 and higher-order interaction confounding is usually not of interest.

Listed below are all the aliases for the main effects, reporting only the two-factor interactions. The bold words indicate the confounding that was intentionally created when we set up the design.

- $\widehat{\beta}_0$ = ABCDEFG

- $\widehat{\beta}_\mathbf{A} \rightarrow$ A + BD + CE + FG

- $\widehat{\beta}_\mathbf{B} \rightarrow$ B + AD + CF + EG

- $\widehat{\beta}_\mathbf{C} \rightarrow$ C + AE + BF + DG

- $\widehat{\beta}_\mathbf{D} \rightarrow$ **D + AB** + CG + EF

- $\widehat{\beta}_\mathbf{E} \rightarrow$ **E + AC** + BG + DF

- $\widehat{\beta}_\mathbf{F} \rightarrow$ **F + BC** + AG + DE

- $\widehat{\beta}_\mathbf{G} \rightarrow$ G + CD + BE + AF

4. If this confounding pattern is not suitable, for example, if you expect interaction **BG** to be important but also main effect **E**, then choose a different set of generators before running the experiment. Or more simply, reassign your variables (temperature, pressure, pH, agitation, *etc*) to different letters of **A**, **B**, *etc* to obtain another, hopefully more desirable, confounding relationship.

Example 2

From a cause-and-effect analysis, flowcharts, brainstorming session, expert opinions, operator opinions, and information from suppliers you might determine that there are 8 factors that could impact an important response variable. Rather than running $2^8 = 256$ experiments, you can run $2^{8-4} = 16$ experiments. (Note: you cannot have fewer experiments: $2^{8-5} = 8$ runs are not sufficient to estimate the intercept and 8 main effects).

So the 2^{8-4} factorial will have 2^4 runs. Assign the first 4 factors to **A**, **B**, **C** and **D** in the usual full factorial manner to create these 16 runs, then assign the remaining 4 factors to the three-factor interactions:

- **E** = **ABC**, or **I** = **ABCE**

- **F** = **ABD**, or **I** = **ABDF**

- **G** = **BCD**, or **I** = **BCDG**

- **H** = **ACD**, or **I** = **ACDH**

So by multiplying all combinations of the words we obtain the complete defining relationship. We expect $p = 4$ generators and $2^p = 16$ words in the defining relationship.

- Two at a time: **I = ABCE $\times$ ABDF = CDEF, I = ABCE $\times$ BCDG = ADEG**, *etc*

- Three at a time: **I = ABCE $\times$ ABDF $\times$ BCDG = BEFG**, *etc*

- Four at a time: **I = ABCE $\times$ ABDF $\times$ BCDG $\times$ ACDH = ABCDEFGH**.

The defining relationship is **I = ABCE = ABDF = BCDG = ACDH = CDEF = ADEG = ... = ABCDEFGH**, and the shortest word, not counting the intercept, has 4 characters. We will refer to this later as the *design's resolution* (page 269).

Next we can calculate all aliases of the main effects. So for **A = IA = BCE = BDF = ABCDG = CDH = ACDEF = DEG = . . . = BCDEFGH**, indicating that A will be confounded with **BCE + BDF + ABCDG + . . .**. In this example none of the main effects have been aliased with two-factor interactions. The aliasing is only with 3-factor and higher interaction terms.

Summary

1. It is tedious and error prone to calculate the aliasing structure by hand, so computer software is useful in this case. For example, for the 2^{7-4} system can be created in R by first loading the `BHH2` package, then using the command `ffDesMatrix(k=7, gen=list(c(4,1,2), c(5,1,3), c(6,2,3), c(7,1,2,3)))`. See the R tutorial[119] for more details on how to install packages like BHH2.

2. The choice of generators is not unique and other choices may lead to a different, more preferable confounding pattern. But it is often easier to use the letters **A, B, C**, *etc*, then just reassign the factors to the letters to achieve the "least-worst" confounding for your situation.

3. In general, a 2^{k-p} factorial design is produced by p generators and has a defining relationship of 2^p words.

There is a quick way to calculate if main effects will be confounded with 2fi or 3fi without having to go through the process shown in this section. This is described next when we look at *design resolution* (page 269).

5.9.6 Design resolution

The resolution of a design is given by the length of the shortest word in the defining relation. We normally write the resolution as a subscript to the factorial design using Roman numerals. Some examples:

1. The 2^{7-4} *example 1* in the previous section had the shortest word of 3 characters, so this would be called a 2^{7-4}_{III} design. Main effects were confounded with 2-factor interactions in that example.

2. The 2^{8-5} *example 2* had as the shortest word length of 4 characters, so this would be a 2^{8-5}_{IV} design. Main effects were confounded with 3-factor interactions.

3. Finally, imagine the case of 2^{5-1}, in other words a half-fraction design. The first four factors are written in the standard factorial way, but the fifth factor is generated from **E = ABCD**. So its defining relation is **I = ABCDE**, where the shortest word, not counting the intercept, is 5 characters - it is a 2^{5-1}_{V} design. You can verify for yourself that main effects will be confounded with 4-factor interactions in this design.

You can consider the resolution of a design to be an indication of how clearly the effects can be separated in a design. The higher the resolution, the lower the degree of confounding: this is desirable. Within reason, always aim for a higher resolution design given your experimental budget, but also accept a lower resolution, at least initially, in order to test for more factors. We will discuss this further in the section on *screening designs* (page 271).

You can interpret the resolution index as follows: let main effects = 1, two-factor interactions = 2, three-factor interactions = 3, *etc*. Then subtract this number from the resolution index to show how that effect is aliased. Consider a resolution IV design, since $4 - 1 = 3$, it indicates that main effects are

---

[119] https://learnche.org/4C3/Software_tutorial

aliased with 3fi, but not with two-factor interactions; and $4 - 2 = 2$ indicates that 2fi are aliased with each other. Here is a summary:

Resolution III designs: are good for screening

- Are excellent for initial screening: to separate out the important factors from the many potential factors. To identify important variables that can be studied in the next experiments.

- Main effects are not confounded with each other.

- Main effects are aliased with two-factor interactions $(3 - 1 = 2)$.

- Two-factor interactions are aliased with main effects $(3 - 2 = 1)$.

Resolution IV designs: are good for characterizing

- Most useful for characterizing (learning about and understanding) a system, both the main effects and the interactions, since:

- Interactions are very interesting in many systems, since they go beyond just the main effects and can improve your process even more (or sometimes take away from it too!).

- Main effects are not confounded with each other.

- Main effects are not aliased with two-factor interactions either $(4 - 1 = 3)$.

- Main effects are aliased with three-factor interactions though $(4 - 1 = 3)$.

- Two-factor interactions are still aliased with each other though $(4 - 2 = 2)$.

Resolution V designs: are good for optimizing

- For optimizing a process, learning about complex effects, and developing high-accuracy predictive models, since:

- Main effects are not confounded with each other.

- Main effects are not aliased with two-factor interactions.

- Two-factor interactions are not aliased with each other either.

- But two-factor interactions are aliased with three-factor interactions $(5 - 2 = 3)$.

The above guidance about using resolution IV and V designs for characterization and optimization is fairly general - there are many cases where a satisfactory optimization can be performed with a resolution IV experiment.

In this text currently, for resolution III, IV and V designs we look at factorial designs. However, there are a number of other design types which can also be used. If you are interested, please research Plackett-Burman designs, Box-Behnken designs, central composite designs, and *definitive screening designs* (page 291).

You can use the following table to visualize the trade-off between design resolution, the number of factors (k), the number of runs required, and the aliasing pattern. The table is adapted from the text by Box, Hunter and Hunter (2nd edition, p 272), and (1st edition, p 410).

Number of factors, *k*

Number of runs	3	4	5	6	7	8	9
4	2^{3-1}_{III} ±C=AB						
8	2^3 *full*	2^{4-1}_{IV} ±D=ABC	2^{5-2}_{III} ±D=AB ±E=AC	2^{6-3}_{III} ±D=AB ±E=AC ±F=BC	2^{7-4}_{III} ±D=AB ±E=AC ±F=BC ±G=ABC		
16	2^3 *twice*	2^4 *full*	2^{5-1}_{V} ±E=ABCD	2^{6-2}_{IV} ±E=ABC ±F=ABD	2^{7-3}_{IV} ±E=ABC ±F=ABD ±G=ACD	2^{8-4}_{IV} ±E=ABC ±F=ABD ±G=ACD ±H=BCD	2^{9-5}_{III}
32	2^3 *4 times*	2^4 *twice*	2^5 *full*	2^{6-1}_{VI} ±F=ABCDE	2^{7-2}_{IV} ±F=ABC ±G=ABDE	2^{8-3}_{IV} ±F=ABC ±G=ABD ±H=ACDE	2^{9-4}_{IV}
64	2^3 *8 times*	2^4 *4 times*	2^5 *twice*	2^6 *full*	2^{7-1}_{VII} ±G=ABCDEF	2^{8-2}_{V} ±G=ABCD ±H=ABEF	2^{9-3}_{IV}

increasing cost (downward, on left)

increasing information about additional factors (→) *lower resolution, greater aliasing*

5.9.7 Saturated designs for screening

A saturated design can be likened to a well trained doctor asking you a few, but very specific, questions to identify a disease or problem. On the other hand, if you sit there just tell the doctor all your symptoms, you may or may not get an accurate diagnosis. Designed experiments, like visiting this doctor, shortens the time required to identify the major effects in a system, and to do so as accurately as possible, within limited budget.

Saturated designs are most suited for screening, and should always be run when you are investigating a new system with many factors. These designs are usually of resolution III and allow you to determine the main effects with a low number of experiments.

For example, a 2^{7-4}_{III} factorial, introduced in the section on *highly fractionated designs* (page 266), will screen 7 factors in 8 experiments. Once you have run the 8 experiments you can quickly tell which subset of the 7 factors are actually important, and spend the rest of your budget on clearly understanding these effects and their interactions. Bear in mind that there is a risk of confounding, as previously described in that section.

Let's see how by continuing the previous example, repeated again below with the corresponding values of y. Recall it was a *set of eight experiments in seven factors* (page 266):

Experiment	A	B	C	D=AB	E=AC	F=BC	G=ABC	y
1	−	−	−	+	+	+	−	77.1
2	+	−	−	−	−	+	+	68.9
3	−	+	−	−	+	−	+	75.5
4	+	+	−	+	−	−	−	72.5
5	−	−	+	+	−	−	+	67.9
6	+	−	+	−	+	−	−	68.5
7	−	+	+	−	−	+	−	71.5
8	+	+	+	+	+	+	+	63.7

Use a least squares model to estimate the coefficients in the model:

$$\mathbf{y} = \mathbf{Xb}$$

$$\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

where $\mathbf{b} = [b_0, b_A, b_B, b_C, b_D, b_E, b_F, b_G]$. The matrix $\mathbf{X}$ is essentiall a copy of the above table, but with an added column of 1's for the intercept term. Notice that the $\mathbf{X}^T\mathbf{X}$ matrix will be diagonal. Make sure you can calculate $\mathbf{X}^T\mathbf{X}$ by hand, at least once. It is also straightforward to calculate the solution vector (by hand!), which you can confirm to be $\mathbf{b} = [70.7, -2.3, 0.1, -2.8, -0.4, 0.5, -0.4, -1.7]$.

How do you assess which main effects are important? There are eight data points and eight parameters, so there are no degrees of freedom and the residuals are all zero. In this case you have to use a *Pareto plot* (page 255), which requires that your variables have been suitably scaled in order to judge importance of the main effects relative to each other. The Pareto plot would be given as shown below, and as usual, it does not show the intercept term.

```
                                      R.code
# Create vectors for each factor in the experiment
A = B = C = c(-1, +1)
design = expand.grid(A=A, B=B, C=C)
A = design$A
B = design$B
C = design$C
D = A*B
E = A*C
F = B*C
G = A*B*C
y = c(77.1, 68.9, 75.5, 72.5, 67.9, 68.5, 71.5, 63.7)

demo = lm(y ~ A + B + C + D + E + F + G)
summary(demo)

# OK, now we are ready to generate the Pareto plot.
# Let's use a library to do that for us.

# library(pid) <-- best to use this!
# It is better to uncomment and use the line above.

# But this embedded R script on this website does not have the
# "pid" library available. So we will load the required function
# from an external server instead:
source('https://yint.org/paretoPlot.R')

# And now we can generate the plot:
paretoPlot(demo)

# Try getting the results manually:
X_matrix = model.matrix(demo)
XtX <- t(X_matrix) %*% X_matrix
```

<div align="right">(continues on next page)</div>

```
print('The XtX matrix is:')
print(XtX)

Xty <- t(X_matrix) %*% y
b = solve(XtX) %*% Xty
print('The solution vector is:')
print(b)
```
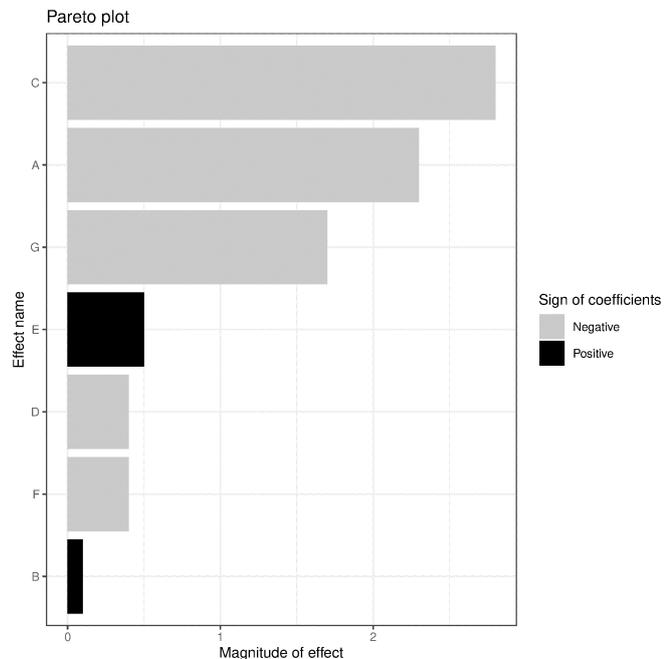

Significant effects would be **A**, **C** and **G**. The next largest effect, **E**, though fairly small, could be due to the main effect **E** or due to the **AC** interaction, because recall the confounding pattern, up to the 2 factor-interactions, for main effect was $\widehat{\beta}_{\mathbf{E}} \rightarrow \mathbf{E} + \mathbf{AC} + \mathbf{BG} + \mathbf{DF}$.

The factor **B** is definitely not important to the response variable in this system and can be excluded in future experiments, as could **F** and **D** likely. Future experiments should focus on the **A**, **C** and **G** factors and their interactions. We show how to use these existing 8 experiments in the above table, but add a few new ones in the next section on design foldover and by understanding projectivity.

A side note on screening designs is a mention of Plackett and Burman designs. These designs can sometimes be of greater use than a highly fractionated design. A fractional factorial must have 2^{k-p} runs, for integers k and p: i.e. either $4, 8, 16, 32, 64, 128, \ldots$ runs. Plackett-Burman designs are screening designs that can be run in any multiple of 4 greater or equal to 12:, i.e. $12, 16, 20, 24, \ldots$ runs. The Box, Hunter, and Hunter book has more information in Chapter 7, but another interesting paper on these topic is by Box and Bisgaard: "What can you find out from 12 experimental runs?", which shows how to screen for 11 factors in 12 experiments.

Video for this section.

An important mention to readers interested in other, arguable better screening strategies, is to consider *definitive screening designs* (page 291).

5.9.8 Design foldover

Experiments are not a one-shot operation. They are almost always sequential, as we learn more and more about our system. Once the first screening experiments are complete there will always be additional questions. In this section we consider two common questions that arise after an initial set of fractional factorials have been run.

Dealias a single main effect (switch sign of one factor)

In the previous example we had a 2_{III}^{7-4} system with generators **D=AB**, **E=AC**, **F=BC**, and **G=ABC**. Effect **C** was the largest effect. But we cannot be sure it was large due to factor **C** alone: it might have been one of the interactions it is aliased with. The aliasing pattern for effect **C** was: $\widehat{\beta}_C \rightarrow$ **C** + **AE** + **BF** + **DG**. For example, we might have reason to believe the **AE** interaction is large. We would like to do some additional experiments so that **C** becomes unconfounded with any two-factor interactions.

The way we can do this is to run another 8 experiments, but this time just change the sign of **C** to **-C**; in other words, re-run the original 8 experiments where the *only* thing that is changed is to flip the signs on column **C**; the columns which are generated from column **C** should remain as they were. This implies that the generators have become **D=AB**, **E=-AC**, **F=-BC**, and **G=-ABC**. We must emphasize, do not re-create these generated columns from new signs in column **C**. What we have now is another 2_{III}^{7-4} design. You can calculate the aliasing pattern for the recent 8 experiments is $\widehat{\beta}_C \rightarrow$ **C** - **AE** - **BF** - **DG**.

Now consider putting all 16 runs together and analyzing the joint results. There are now 16 parameters that can be estimated. Using computer software you can see that factor C will have no confounding with any two-factor interactions. Also, any two-factor interactions involving **C** are removed from the other main effects. For example, factor **A** was originally confounded with **CE** with the first 8 experiments; but that will be removed when analyzing all 16 together.

So our **general conclusion** is: switching the sign of one factor will de-alias that factor's main effect, and all its associated two-factor interactions when analyzing the two fractional factorials together. In the above example, we will have an unconfounded estimate of **C** and 2-factor interactions involving **C** will be unconfounded with main effects: i.e. **AC**, **BC**, **CD**, **CE**, **CF** and **CG**.

Increase design resolution (switching all signs)

One can improve the aliasing structure of a design when switching all the signs of all factors from the first fraction. Switching the signs means that we take the complete design matrix of factor settings, and simply flip the signs to create the second fraction. A factor setting that was run at the low level in the first fraction is then run at a high level in the second fraction. This includes generated factors: imagine that **D = AB**, and we had **A** $= -1$, **B** $= +1$ for a particular run in the first fraction. In the first fraction we would have **D** $= (-1)(+1) = -1$, while in the second fraction it would simply be **D** $= +1$, with **A** $= +1$ and **B** $= -1$ respectively, since we simply flip signs. **We do not regenerate the generated factors.**

In the 2_{III}^{7-4} example, we ran 8 experiments. If we now run another 8 experiments with all the signs switched, then these 8+8 experiments would be equivalent to a 2_{IV}^{7-3} design. This resolution IV design means that all the main effects can now be estimated without confounding from any two-factor interactions. However, the two-factor interactions are still confounded with themselves.

This is a good strategy in general: to run the first fraction of runs to assess the main effects. It serves as a good checkpoint, as well providing intermediate results to colleagues, it can be used to get approval/budget to run the next set of experiments. If we then perform another set of runs we know that we are doing them in a way that captures the most additional information, and with the least confounding. Remember experimentation is not done in a single go; it is sequential. We perform experiments, analyze the results, and design further experiments to reach our goal.

5.9.9 Projectivity

A final observation for this section is how fractional factorials will collapse down to a full factorial under certain conditions.

Consider the diagram here, where a half fraction in factors **A**, **B** and **C** was run (4 experiments) at the closed points.

● Half-fraction runs
○ Experiments not run
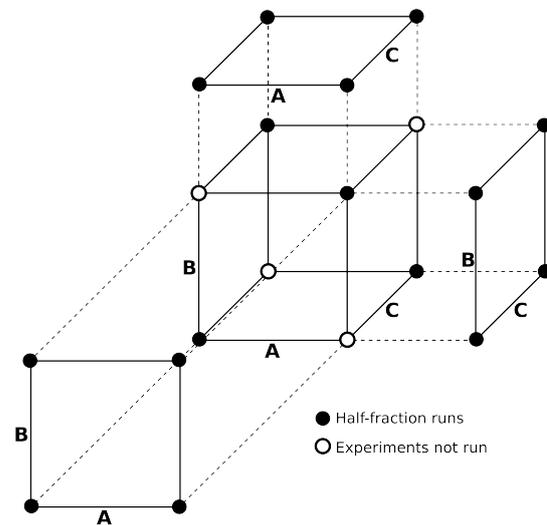
On analyzing the data, the experimenter discovers that factor **C** does not actually have an impact on response y. This means the **C** dimension could have been removed from the experiment, and is illustrated by projecting the **A** and **B** factors forward, removing factor **C**. Notice that this is now a full factorial in factors **A** and **B**. The same situation would hold if either factor **B** or factor **A** were found to be unimportant. Furthermore if two factors are found to be unimportant, then this corresponds to 2 replicate experiments in 1 factor.

This projectivity of factorials holds in general for a larger number of factors. The above example, actually a 2_{III}^{3-1} experiment, was a case of projectivity = 2. In general, projectivity = P = resolution -1. So if you have a resolution IV fractional factorial, then it has projectivity = $P = 4 - 1$, implying that it contains a full factorial in 3 factors. So a 2_{IV}^{6-2} (16 runs) system with 6 factors, contains an embedded full factorial using a combination of any 3 factors; if any 3 factors were found unimportant, then a replicated full factorial exists in the remaining 3 factors.

5.10 Blocking and confounding for disturbances

5.10.1 Characterization of disturbances

External disturbances will always have an effect on our response variable, y. Operators, ambient conditions, physical equipment, lab analyses, and time-dependent effects (catalyst deactivation, fouling), will impact the response. This is why it is crucial to *randomize* (page 239) the order of experiments: so that these **unknown, unmeasurable, and uncontrollable** disturbances cannot systematically affect the response.

However, certain disturbances are **known, or controllable, or measurable**. For these cases we perform pairing and blocking. We have already discussed pairing in the univariate section: pairing is when two experiments are run on the same subject and we analyze the differences in the two response

values, rather than the actual response values. If the effect of the disturbance has the same magnitude on both experiments, then that disturbance will cancel out when calculating the difference. The magnitude of the disturbance is expected to be different between paired experiments, but is expected to be the same within the two values of the pair.

Blocking is slightly different: blocking is a special way of running the experiment so that the disturbance actually does affect the response, but we construct the experiment so that this effect is not misleading.

Finally, a disturbance can be characterized as a **controlled disturbance**, in which case it isn't a disturbance anymore, as it is held constant for all experiments, and its effect cancels out. But it might be important to investigate the controlled disturbance, especially if the system is operated later on when this disturbance is at a different level.

5.10.2 Blocking and confounding

Video for this section.

It is common for known, or controllable or measurable factors to have an effect on the response. However these disturbance factors might not be of interest to us during the experiment. Cases are:

- *Known and measurable, not controlled*: Reactor vessel A is known to achieve a slightly better response, on average, than reactor B. However both reactors must be used to complete the experiments in the short time available.

- *Known, but not measurable nor controlled*: There is not enough material to perform all 2^3 runs, there is only enough for 4 runs. The impurity in either the first batch A for 4 experiments, or the second batch B for the other 4 runs will be different, and might either increase or decrease the response variable (we don't know the effect it will have).

- *Known, measurable and controlled*: Reactor vessel A and B have a known, measurable effect on the output, y. To control for this effect we perform all experiments in either reactor A or B, to prevent the reactor effect from confounding (confusing) our results.
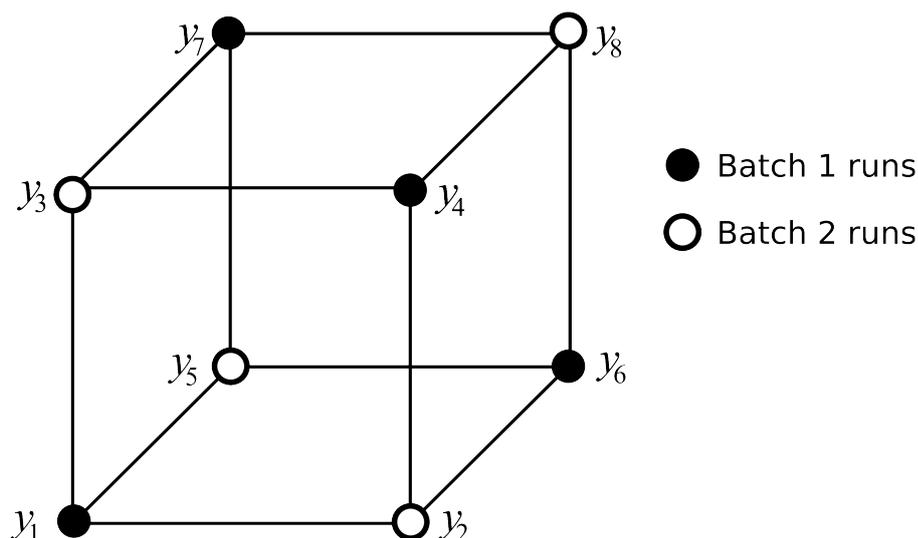
In this section then we will deal with disturbances that are known, but their effect may or may not be measurable. We will also assume that we cannot control that disturbance, but we would like to minimize its effect.

For example, if we don't have enough material for all 2^3 runs, but only enough for 4 runs, the question is how to arrange the 2 sets of 4 runs so that the known, by unmeasurable disturbance from the

impurity has the least effect on our results and interpretation of the 3 factors.

Our approach is to intentionally *confound* the effect of the disturbance with an effect that is expected to be the least significant. The $A \times B \times C$ interaction term is almost always going to be small for many systems, so we will split the runs that the first 4 are run at the low level of ABC and the other four at the high level, as illustrated.

Each group of 4 runs is called a *block* and the process of creating these 2 blocks is called blocking. The experiments within each block must be run randomly.

Experiment	A	B	C	AB	AC	BC	ABC	Response, y
1	−	−	−	+	+	+	− (batch 1)	$\widetilde{y}_1$
2	+	−	−	−	−	+	+ (batch 2)	$\mathring{y}_2$
3	−	+	−	−	+	−	+ (batch 2)	$\mathring{y}_3$
4	+	+	−	+	−	−	− (batch 1)	$\widetilde{y}_4$
5	−	−	+	+	−	−	+ (batch 2)	$\mathring{y}_5$
6	+	−	+	−	+	−	− (batch 1)	$\widetilde{y}_6$
7	−	+	+	−	−	+	− (batch 1)	$\widetilde{y}_7$
8	+	+	+	+	+	+	+ (batch 2)	$\mathring{y}_8$

If the raw material has a significant effect on the response variable, then we will not be able to tell whether it was due to the $A \times B \times C$ interaction, or due to the raw material, since
$$\hat{\beta}_{ABC} \to \underbrace{\text{ABC interaction}}_{\text{expected to be small}} + \text{raw material effect}.$$

But the small loss due to this confusion of effects, is the gain that we can still estimate the main effects and two-factor interactions without bias, provided the effect of the disturbance is constant. Let's see how we get this result by denoting $\widetilde{y}_i$ as a y response from the first batch of materials and let $\mathring{y}_i$ denote a response from the second batch.

Using the least squares equations you can show for yourself that (some are intentionally left blank for you to complete):

$$\hat{\beta}_A = -\widetilde{y}_1 + \mathring{y}_2 - \mathring{y}_3 + \widetilde{y}_4 - \mathring{y}_5 + \widetilde{y}_6 - \widetilde{y}_7 + \mathring{y}_8$$
$$\hat{\beta}_B =$$
$$\hat{\beta}_C =$$
$$\hat{\beta}_{AB} = +\widetilde{y}_1 - \mathring{y}_2 - \mathring{y}_3 + \widetilde{y}_4 + \mathring{y}_5 - \widetilde{y}_6 - \widetilde{y}_7 + \mathring{y}_8$$
$$\hat{\beta}_{AC} =$$
$$\hat{\beta}_{BC} =$$
$$\hat{\beta}_{ABC} =$$

Imagine now the y response was increased by g units for the batch 1 experiments, and increased by h units for batch 2 experiments. You can prove to yourself that these biases will cancel out for all main effects and all two-factor interactions. The three factor interaction of $\hat{\beta}_{ABC}$ will however be heavily confounded.

Another way to view this problem is that the first batch of materials and the second batch of materials can be represented by a new variable, called D with value of $D_- =$ batch 1 and $D_+ =$ batch 2. We will show next that we must consider this new factor to be generated from the other three: **D = ABC**.

We will also address the case when there are more than two blocks in the next section on the *use of generators* (page 262). For example, what should we do if we have to run a 2^3 factorial but with only enough material for 2 experiments at a time?

5.11 Response surface methods

The purpose of response surface methods (RSM) is to optimize a process or system. RSM is a way to explore the effect of operating conditions (the factors) on the response variable, y. As we map out the unknown response surface of y, we move our process as close as possible towards the optimum, taking into account any constraints.

Initially, when we are far away from the optimum, we will use factorial experiments. As we approach the optimum then these factorials are replaced with better designs that more closely approximate conditions at the optimum.

Notice how it is a *sequential* approach. RSM then is a tool the describes how we should run these sequential sets of experiments. At the start of this *section on designed experiments* (page 240) we showed how sequential experimentation (COST) leads to sub-optimal solutions. Why are we advocating sequential experimentation now? The difference is that here we use sequential experiments by changing *multiple factors simultaneously*, and not changing only one factor at a time.

RSM concept for a single variable: COST approach

We will however first consider just the effect of a single factor, x_1 as it relates to our response, y. This is to illustrate the general response surface process.
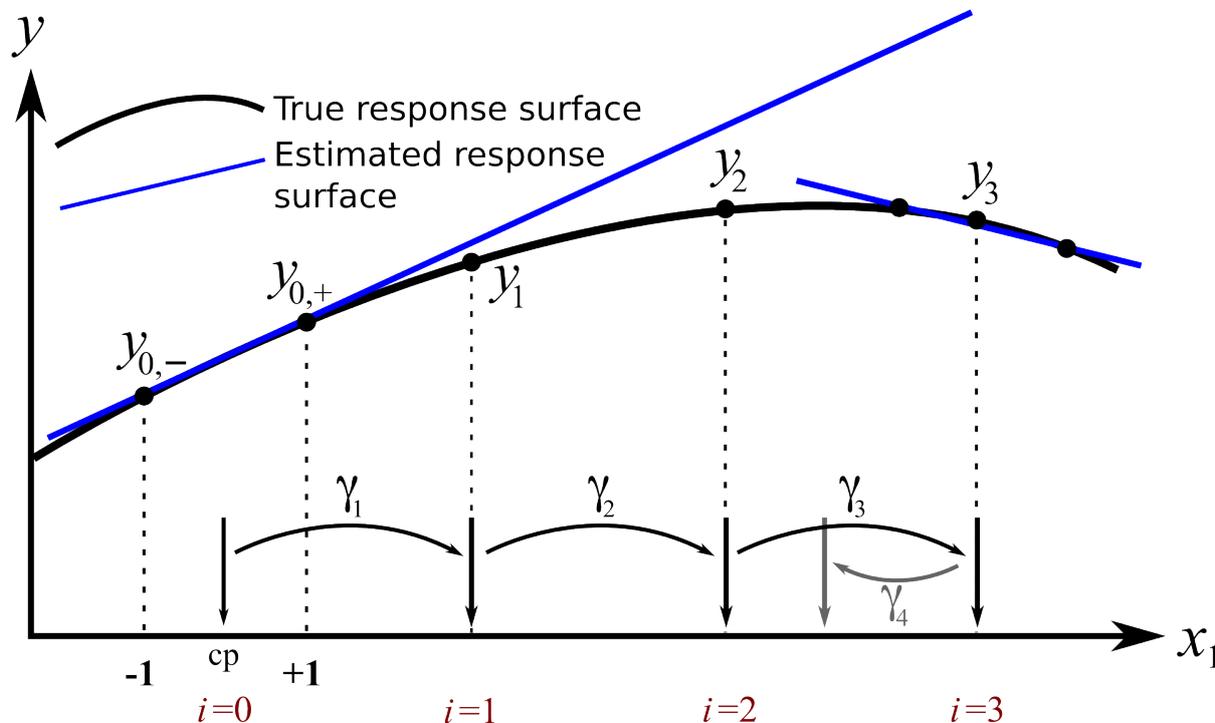
We start at the point marked $i = 0$ as our initial baseline (cp=center point). We run a 2-level experiment, above and below this baseline at -1 and $+1$, in coded units of x_1, and obtain the corresponding response values of $y_{0,-}$ and $y_{0,+}$. From this we can estimate a best-fit straight line and move in the direction that increases y. The sloping tangential line, also called the *path of steepest ascent*.

Make a move of step-size = γ_1 units along x_1 and measure the response, recorded as y_1. The response variable increased, so we keep going in this direction.

Make another step-size, this time of γ_2 units in the direction that increases y. We measure the response, y_2, and are still increasing. Encouraged by this, we take another step of size γ_3. The step-sizes, γ_i should be of a size that is big enough to cause a change in the response in a reasonable number of experiments, but not so big as to miss an optimum.

Our next value of y_3 is about the same size as y_2, indicating that we have plateaued. At this point we can take some exploratory steps and refit the tangential line (which now has a slope in the opposite direction). Or we can just use the accumulated points $y = [y_{0-}, \ y_{0+}, \ y_1, \ y_2, \ y_3]$ and their corresponding x-values to fit a non-linear curve. Either way, we can then estimate a different step-size γ_4 that will bring us closer to the optimum.

This univariate example is in fact what experimenters do when using the *COST approach* (page 240) described earlier. We have:

- exploratory steps of different sizes towards an optimum

- refit the model once we plateau

- repeat

This approach works well if there really is only a single factor that affects the response. But with most systems there are multiple factors that affect the response. We show next how the exact same idea is used, only we change multiple variables at a time to find the optimum on the response surface.

5.11.1 Response surface optimization via a 2-variable system example

This example considers a new system here where two factors, temperature **T**, and substrate concentration **S** are known to affect the yield from a bioreactor. But in this example we are not just interested in yield, but actually the total profit from the system. This profit takes into account energy costs, raw materials costs and other relevant factors. The illustrations in this section show the contours of profit in light grey, but in practice these are obviously unknown.

We currently operate at this baseline condition:

- **T** = 325 K

- **S** = 0.75 g/L

- Profit = \$407 per day

We start by creating a full factorial around this baseline by choosing $\Delta_T = 10$ K, and $\Delta_S = 0.5$ g/L based on our knowledge that these are sufficiently large changes to show an actual difference in the response value, but not too large so as to move to a totally different form of operation in the bioreactor.

The results from the full factorial are in the table here:

Experiment	T (actual)	S (actual)	T (coded)	S (coded)	Profit
Baseline	325 K	0.75 g/L	0	0	407
1	320 K	0.50 g/L	−	−	193
2	330 K	0.50 g/L	+	−	310
3	320 K	1.0 g/L	−	+	468
4	330 K	1.0 g/L	+	+	571

Clearly the promising direction to maximize profit is to operate at higher temperatures and higher substrate concentrations. But how much much higher and in what ratio should we increase T and S? These answers are found by building a linear model of the system from the factorial data:

$$\hat{y} = b_0 + b_T x_T + b_S x_S + b_{TS} x_T x_S$$

$$\hat{y} = 389.8 + 55x_T + 134x_S - 3.50x_T x_S$$

where $x_T = \dfrac{x_{T,\text{actual}} - \text{center}_T}{\Delta_T/2} = \dfrac{x_{T,\text{actual}} - 325}{5}$ and similarly, $x_S = \dfrac{x_{S,\text{actual}} - 0.75}{0.25}$.

The model shows that we can expect an increase of \$55/day of profit for a unit increase in x_T (coded units). In real-world units that would require increasing temperature by $\Delta x_{T,\text{actual}} = (1) \times \Delta_T/2 = 5\text{K}$ to achieve that goal. That scaling factor comes from the coding we used:

$$x_T = \frac{x_{T,\text{actual}} - \text{center}_T}{\Delta_T/2}$$

$$\Delta x_T = \frac{\Delta x_{T,\text{actual}}}{\Delta_T/2}$$

Similarly, we can increase S by $\Delta x_S = 1$ unit $= 1 \times \Delta_S/2 = 0.5/2 = 0.25$ g/L real-world units, to achieve a \$134 per day profit increase.

The interaction term is small, indicating the response surface is mostly linear in this region. The illustration shows the model's contours (straight, green lines). Notice that the model contours are a good approximation to the actual contours (dotted, light grey), which are unknown in practice.

Contours of profit per day ($)
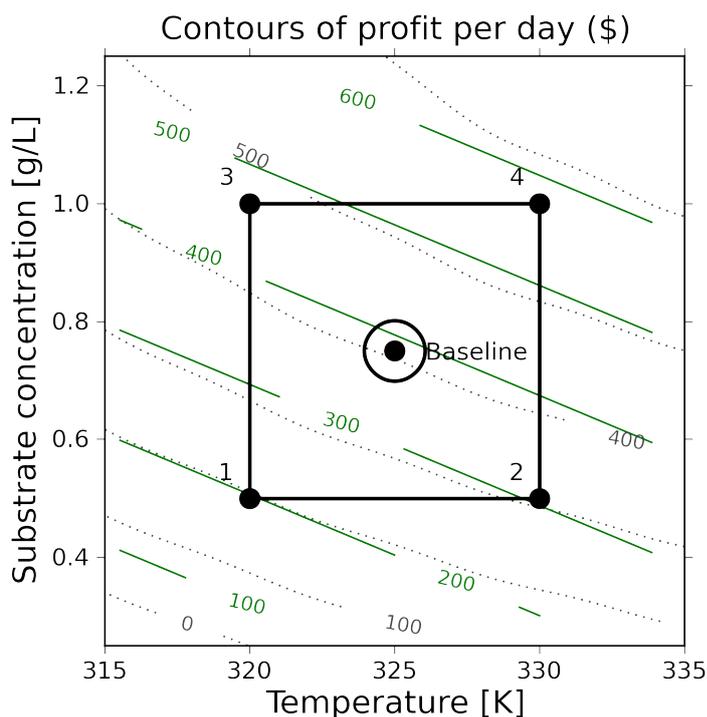
To improve our profit in the optimal way we move along our estimated model's surface, in the direction of steepest ascent. This direction is found by taking partial derivatives of the model function, ignoring the interaction term, since it is so small.

$$\frac{\partial \hat{y}}{\partial x_T} = b_T = 55 \qquad\qquad \frac{\partial \hat{y}}{\partial x_S} = b_S = 134$$

This says for every $b_T = 55$ coded units that we move by in x_T we should also move x_S by $b_S = 134$ coded units. Mathematically:

$$\frac{\Delta x_S}{\Delta x_T} = \frac{134}{55}$$

The simplest way to do this is just to pick a move size in one of the variables, then change the move size of the other one.

So we will choose to increase $\Delta x_T = 1$ coded unit, which means:

$$\Delta x_T = 1$$

$$\Delta x_{T,\text{actual}} = 5\,\text{K}$$

$$\Delta x_S = \frac{b_S}{b_T}\Delta x_T = \frac{134}{55}\Delta x_T$$

but we know that $\quad \Delta x_S = \dfrac{x_{S,\text{actual}}}{\Delta_S/2}$

$$\Delta x_{S,\text{actual}} = \frac{134}{55} \times 1 \times \Delta_S/2 \text{ by equating previous 2 lines}$$

$$\Delta x_{S,\text{actual}} = \frac{134}{55} \times 1 \times 0.5/2 = \mathbf{0.61}\,\text{g/L}$$

- $T_5 = T_{\text{baseline}} + \Delta x_{T,\text{actual}} = 325 + 5 = 330\,\text{K}$

- $S_5 = S_{\text{baseline}} + \Delta x_{S,\text{actual}} = 0.75 + 0.6 = 1.36\,\text{g/L}$

So when we run the next experiment at these conditions. The daily profit is $y_5 = \$\,669$, improving quite substantially from the baseline case.

We decide to make another move, in the same direction of steepest ascent, i.e. along the vector that points in the $\frac{134}{55}$ direction. We move the temperature up 5K, although we could have used a larger or smaller step size if we wanted:

- $T_6 = T_5 + \Delta x_{T,\text{actual}} = 330 + 5 = 335\,\text{K}$

- $S_6 = S_5 + \Delta x_{S,\text{actual}} = 1.36 + 0.61 = 1.97\,\text{g/L}$

Again, we determine profit at $y_6 = \$\,688$. It is still increasing, but not by nearly as much. Perhaps we are starting to level off. However, we still decide to move temperature up by another 5 K and increase the substrate concentration in the required ratio:

- $T_7 = T_6 + \Delta x_{T,\text{actual}} = 335 + 5 = 340\,\text{K}$

- $S_7 = S_6 + \Delta x_{S,\text{actual}} = 1.97 + 0.61 = 2.58\,\text{g/L}$

The profit at this point is $y_7 = \$\,463$. We have gone too far as profit has dropped off. So we return back to our *last best point*, because the surface has obviously changed, and we should refit our model with a new factorial in this neighbourhood:

Experiment	T (actual)	S (actual)	T	S	Profit
6	335 K	1.97 g/L	0	0	688
8	331 K	1.77 g/L	−	−	694
9	339 K	1.77 g/L	+	−	725
10	331 K	2.17 g/L	−	+	620
11	339 K	2.17 g/L	+	+	642

This time we have deciding to slightly smaller ranges in the factorial $\text{range}_T = 8 = (339 - 331)\,\text{K}$ and $\text{range}_S = 0.4 = (2.17 - 1.77)\,\text{g/L}$ so that we can move more slowly along the surface.

Contours of profit per day ($)
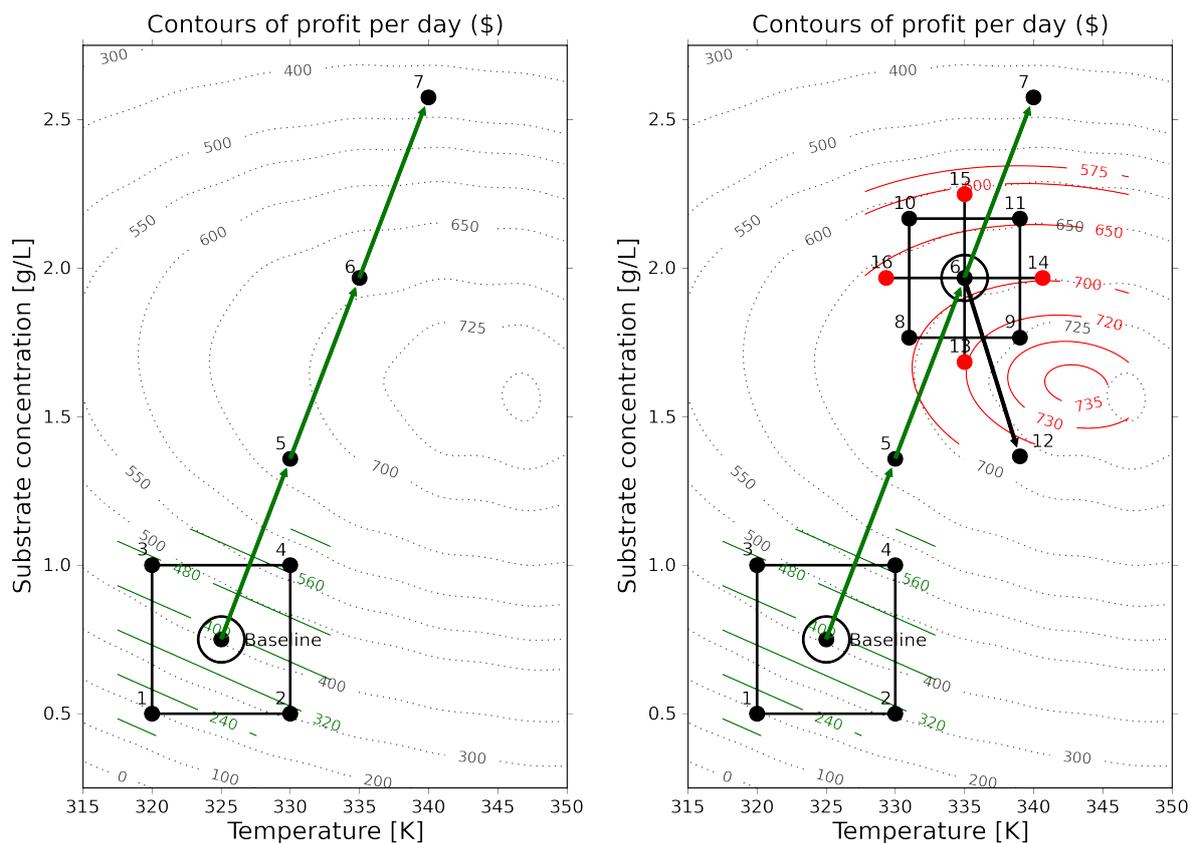
A least squares model from the 4 factorial points (experiments 8, 9, 10, 11, run in random order), seems to show that the promising direction now is to increase temperature but decrease the substrate concentration.

$$\hat{y} = b_0 + b_T x_T + b_S x_S + b_{TS} x_T x_S$$

$$\hat{y} = 673.8 + 13.25 x_T - 39.25 x_S - 2.25 x_T x_S$$

As before we take a step in the direction of steepest ascent of b_T units along the x_T direction and b_S units along the x_S direction. Again we choose $\Delta x_T = 1$ unit, though we must emphasize that we could used a smaller or larger amount, if desired.

$$\frac{\Delta x_S}{\Delta x_T} = \frac{-39}{13}$$

$$\Delta x_S = \frac{-39}{13} \times 1$$

$$\Delta x_{S,\text{actual}} = \frac{-39}{13} \times 1 \times 0.4/2 = -0.6 \,\text{g/L}$$

$$\Delta x_{T,\text{actual}} = 4 \,\text{K}$$

- $T_{12} = T_6 + \Delta x_{T,\text{actual}}$ = 335 + 4 = 339K

- $S_{12} = S_6 + \Delta x_{S,\text{actual}}$ = 1.97 - 0.6 = 1.37 g/L

We determine that at run 12 the profit is y_{12} = $ 716. But our previous factorial had a profit value of $725 on one of the corners. Now it could be that we have a noisy system; after all, the difference between $716 and $725 is not too much, but there is a relatively large difference in profit between the other points in the factorial.

One must realize that as one approaches an optimum we will find:

- The response variable will start to plateau, since, recall that the first derivative is zero at an optimum, implying the surface flattens out, and all points, in all directions away from the optimum are worse off.

- The response variable remains roughly constant for two consecutive jumps, because one has jumped over the optimum.

- The response variable decreases, sometimes very rapidly, because we have overshot the optimum.

- The presence of curvature can also be inferred when interaction terms are similar or larger in magnitude than the main effect terms.

An optimum therefore exhibits curvature, so a model that only has linear terms in it will not be suitable to use to find the direction of steepest ascent along the *true response surface*. We must add terms that account for this curvature.

Checking for curvature

The factorial's center point can be predicted from $(x_T, x_S) = (0, 0)$, and is just the intercept term. In the last factorial, the predicted center point was $\hat{y}_{\text{cp}}$ = \$670; yet the actual center point from run 6 showed a profit of \$ 688. This is a difference of \$18, which is substantial when compared to the main effects' coefficients, particularly of temperature.

So when the measured center point value is quite different from the predicted center point in the linear model, then that is a good indication there is curvature in the response surface. The way to accommodate for that is to add quadratic terms to the estimate model.

Adding higher-order terms using central composite designs

We will not go into too much detail about central composite designs, other than to show what they look like for the case of 2 and 3 variables. These designs take an existing orthogonal factorial and augment it with axial points. This is great, because we can start off with an ordinary factorial and always come back later to add the terms to account for nonlinearity.

The axial points are placed $4^{0.25} = 1.4$ coded units away from the center for a 2 factor system, and $8^{0.25} = 1.7$ units away for a $k = 3$ factor system. Rules for higher numbers of factors, and the reasoning behind the 1.4 and 1.7 unit step size can be found, for example in the textbook by Box, Hunter and Hunter.

2-factor system   3-factor system
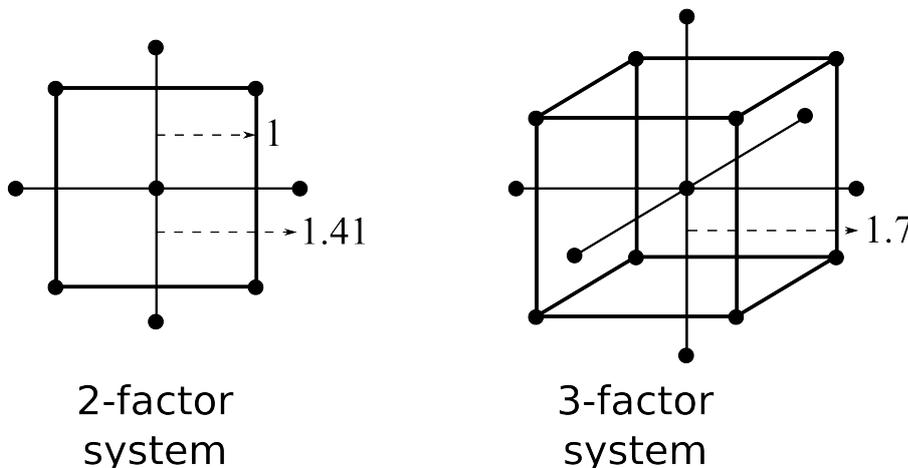
So a central composite design layout was added to the factorial in the above example and the experiments run, randomly, at the 4 axial points.

The four response values were $y_{13} = 720$, $y_{14} = 699$, $y_{15} = 610$, and $y_{16} = 663$. This allows us to estimate a model with quadratic terms in it: $y = b_0 + b_T x_T + b_S x_S + b_{TS} x_T x_S + b_{TT} x_T^2 + b_{SS} x_S^2$. The parameters in this model are found in the usual way, using a least-squares model:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

$$
\begin{bmatrix}
y_8 \\
y_9 \\
y_{10} \\
y_{11} \\
y_6 \\
y_{13} \\
y_{14} \\
y_{15} \\
y_{16}
\end{bmatrix}
=
\begin{bmatrix}
1 & -1 & -1 & +1 & +1 & +1 \\
1 & +1 & -1 & -1 & +1 & +1 \\
1 & -1 & +1 & -1 & +1 & +1 \\
1 & +1 & +1 & +1 & +1 & +1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -1.41 & 0 & 0 & 2 \\
1 & 1.41 & 0 & 0 & 2 & 0 \\
1 & 0 & 1.41 & 0 & 0 & 2 \\
1 & -1.41 & 0 & 0 & 2 & 0
\end{bmatrix}
\begin{bmatrix}
b_0 \\
b_T \\
b_S \\
b_{TS} \\
b_{TT} \\
b_{SS}
\end{bmatrix}
+ \mathbf{e}
$$

$$y = 688 + 13 x_T - 39 x_S - 2.4 x_T x_S - 4.2 x_T^2 - 12.2 x_S^2$$

Notice how the linear terms estimated previously are the same! The quadratic effects are clearly significant when compared to the other effects, which was what prevented us from successfully using a linear model to project out to point 12 previously.

The final step in the response surface methodology is to plot this model's contour plot and predict where to run the next few experiments. As the solid contour lines in the illustration show, we should run our next experiments roughly at $T = 343$K and $S = 1.60$ g/L where the expected profit is around $736. We get those two values by eye-balling the solid contour lines, drawn from the above non-linear model. You could find this point analytically as well.

This is not exactly where the true process optimum is, but it is pretty close to it (the temperature of $T = 343$K is just a little lower that where the true optimum is.

This example has demonstrated how powerful response surface methods are. A minimal number of experiments has quickly converged onto the true, unknown process optimum. We achieved this by building successive least squares models that approximate the underlying surface. Those least squares models are built using the tools of fractional and full factorials and basic optimization theory, to climb the hill of steepest ascent.

5.11.2 The general approach for response surface modelling

1. Start at your baseline conditions and identify the main factors based on physics of the process, operator input, expert opinion input, and intuition. Also be aware of any constraints, especially for safe process operation. Perform factorial experiments (full or fractional factorials), completely randomized. Use the results from the experiment to estimate a linear model of the system:

$$\hat{y} = b_0 + b_A x_A + b_B x_B + b_C x_C \ldots + b_{AB} x_A x_B + b_{AC} x_A x_C + \ldots$$

2. The main effects are usually significantly larger than the two-factor interactions, so these higher interaction terms can be safely ignored. Any main effects that are not significant may be dropped for future iterations.

3. Use the model to estimate the path of steepest ascent (or descent if minimizing y):

$$\frac{\partial \hat{y}}{\partial x_1} = b_1 \qquad \frac{\partial \hat{y}}{\partial x_2} = b_2 \qquad \ldots$$

The path of steepest ascent is climbed. Move any one of the main effects, e.g. b_A by a certain amount, Δx_A. Then move the other effects: $\Delta x_i = \frac{b_i}{b_A}\Delta x_A$. For example, Δx_C is moved by $\frac{b_C}{b_A}\Delta x_A$.

If any of the Δx_i values are too large to safely implement, then take a smaller proportional step in all factors. Recall that these are coded units, so unscale them to obtain the move amount in real-world units.

4. One can make several sequential steps until the response starts to level off, or if you become certain you have entered a different operating mode of the process.

5. At this point you repeat the factorial experiment from step 1, making the last best response value your new baseline. This is also a good point to reintroduce factors that you may have omitted earlier. Also, if you have a binary factor; investigate the effect of alternating its sign at this point. These additional factorial experiments should also include center points.

6. Repeat steps 1 through 5 until the linear model estimate starts to show evidence of curvature, or that the interaction terms start to dominate the main effects. This indicates that you are reaching an optimum.

   • Curvature can be assessed by comparing the predicted center point, i.e. the model's intercept = b_0, against the actual center point response(s). A large difference in the prediction, when compared to the model's effects, indicates the response surface is curved.

7. If there is curvature, add axial points to expand the factorial into a central composite design. Now estimate a quadratic model of the form:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_{12} x_1 x_2 + \ldots + b_{11} x_1^2 + b_{22} x_2^2 + \ldots$$

8. Draw contour plots of this estimated response surface (all data analysis software packages have contour plotting functions) and determine where to place your sequential experiments. You can also find the model's optimum analytically by taking derivatives of the model function.

What is the response variable when optimizing more than one outcome?

Response surface methods consider optimization of a single outcome, or response variable, called y. In many instances we are interested in just a single response, but more often we are interested in a multi-objective response, i.e. there are trade-offs. For example we can achieve a higher production rate, but it is at the expense of more energy.

One way to balance all competing objectives is to rephrase the y variable in terms of total costs, or better still, net profit. This makes calculating the y value more complex, as we have to know the various costs and their relative weightings to calculate the profit. Now you have a single y to work with.

Another way is to superimpose the response surfaces of two or more y-variables. This is tremendously helpful when discussing and evaluating alternate operating points, because plant managers and operators can then visually see the trade-offs.

Summary

1. In the previous sections we used factorials and fractional factorials for screening the important factors. When we move to process optimization, we are assuming that we have already identified the important variables. In fact, we might find that variables that were previously important, appear unimportant as we approach the optimum. Conversely, variables that might have been dropped out earlier, become important at the optimum.

2. Response surface methods generally work best when the variables we adjust are numerically continuous. Categorical variables (yes/no, catalyst A or B) are handled by fixing them at one or the other value, and then performing the optimization conditional on those selected values. It is always worth investigating the alternative values once the optimum has been reached.

3. Many software packages provide tools that help with an RSM study. If you would like to use R in your work, we highly recommend the `rsm` package by Russel Lenth, available in R. You can read more about the package in this article[120] as well as a case-study[121].

5.12 Evolutionary operation

Evolutionary operation (EVOP) is a tool to help maintain a full-scale process at its optimum. Since the process is not constant, the optimum will gradually move away from its current operating point. Chemical processes drift due to things such as heat-exchanger fouling, build-up inside reactors and tubing, catalyst deactivation, and other slowly varying disturbances in the system.

EVOP is an iterative hunt for the process optimum by making small perturbations to the system. Similar to response surface methods, once every iteration is completed, the process is moved towards the optimum. The model used to determine the move direction and levels of next operation are from full or fractional factorials, or designs that estimate curvature, like the central composite design.

Because every experimental run is a run that is expected to produce saleable product (we don't want off-specification product), the range over which each factor is varied must be small. Replicate runs are also made to separate the signal from noise, because the optimum region is usually flat.

Some examples of the success of EVOP and a review paper are in these readings:

- George Box: Evolutionary Operation: A Method for Increasing Industrial Productivity[122]", *Journal of the Royal Statistical Society* (Applied Statistics), **6**, 81 - 101, 1957.

- William G. Hunter and J. R. Kittrell, "Evolutionary Operation: A Review[123]", *Technometrics*, **8**, 389-397, 1966.

Current day examples of EVOP do not appear in the scientific literature much, because this methodology is now so well established.

---

[120] https://cran.r-project.org/web/packages/rsm/vignettes/rsm.pdf
[121] https://cran.r-project.org/web/packages/rsm/vignettes/rs-illus.pdf
[122] https://www.jstor.org/stable/2985505
[123] https://www.jstor.org/stable/1266686

5.13 General approach for experimentation

We complete this section with some guidance for experimentation in general. The main point is that experiments are never run in one go. You will always have more questions after the first round. Box, Hunter and Hunter provide two pieces of guidance on this:

1. The best time to run an experiment is after the experiment. You will discover things from the previous experiment that you wish you had considered the first time around.

2. For the above reason, do not spend more than 20% to 25% of your time and budget on your first group of experiments. Keep some time aside to add more experiments and learn more about the system.

The **first phase** is usually *screening*. Screening designs are used when developing new products and tremendous uncertainty exists; or sometimes when a system is operating so poorly that one receives the go-ahead to manipulate the operating conditions wide enough to potentially upset the process, but learn from it.

- The ranges for each factor may also be uncertain; this is a perfect opportunity to identify suitable ranges for each factor.

- You also learn how to run the experiment on this system, as the operating protocol isn't always certain ahead of time. It is advisable to choose your first experiment to be the center point, since the first experiment will often "fail" for a variety of reasons (you discover that you need more equipment midway, you realize the protocol isn't sufficient, *etc*). Since the center point is not required to analyze the data, it worth using that first run to learn about your system. If successful though, that center point run can be included in the least squares model.

- Include as many factors into as few runs as possible. Use a saturated, resolution III design, or a Plackett and Burman design.

- Main effects will be extremely confounded, but this is a preliminary investigation to isolate the important effects.

The **second phase** is to add *sequential experiments* to the previous experiments.

- Use the concept of foldover: switching the sign of the factor of interest to learn more about a single factor, or switch all signs to increase the design's resolution.

- If you had a prior screening experiment, use the concept of projectivity in the important factors to limit the number of additional experiments required.

- Move to quarter and half-fractions of higher resolution, to better understand the main effects and 2-factor interactions.

The **third phase** is to start *optimizing* by exploring the response surface using the important variables discovered in the prior phases.

- Use full or half-fraction factorials to estimate the direction of steepest ascent or descent.

- Once you reach the optimum, then second order effects and curvature must be added to assess the direction of the optimum.

The **fourth phase** is to *maintain the process optimum* using the concepts of evolutionary operation (EVOP).

- An EVOP program should always be in place on a process, because raw materials change, fouling and catalyst deactivation take place, and other slow moving disturbances have an effect. You should be always hunting for the process optimum.

5.14 Extended topics related to designed experiments

This section is just an overview of some interesting topics, together with references to guide you to more information.

5.14.1 Experiments with mistakes, missing values, or belatedly discovered constraints

Many real experiments do not go smoothly. Once the experimenter has established their -1 and $+1$ levels for each variable, they back that out to real units. For example, if temperature was scaled as $T = \dfrac{T_{\text{actual}} - 450\text{K}}{25\text{K}}$, then $T = -1$ corresponds to 425K and $T = +1$ corresponds to 475K.

But if the operator mistakenly sets the temperature to $T_{\text{actual}} = 465K$, then it doesn't quite reach the $+1$ level required. This is not a wasted experiment. Simply code this as $T = \dfrac{465 - 450}{25} = 0.6$, and enter that value in the least squares model for matrix $\mathbf{X}$. Then proceed to calculate the model parameters using the standard least squares equations. Note that the columns in the X-matrix will not be orthogonal anymore, so $\mathbf{X}^T\mathbf{X}$ will not be a diagonal matrix, but it will be almost diagonal.

Similarly, it might be discovered that temperature cannot be set to 475K when the other factor, for example concentration, is also at its high level. This might be due to physical or safety constraints. On the other hand, $T = 475K$ can be used when concentration is at its low level. This case is the same as described above: set the temperature to the closest possible value for that experiment, and then analyze the data using a least squares model. The case when the constraint is known ahead of time is *dealt with later on* (page 289), but in this case, the constraint was discovered just as the run was to be performed.

Also see the section on *optimal designs* (page 290) for how one can add one or more additional experiments to fix an existing bad set of experiments.

The other case that happens occasionally is that samples are lost, or the final response value is missing for some reason. Not everything is lost: recall the main effects for a full 2^k factorial are estimated k times at *each combination of the factors* (page 258).

If one or more experiments have missing y values, you can still estimate these main effects, and sometimes the interaction parameters by hand. Furthermore, analyzing the data in a least squares model will be an undetermined system: more unknowns than equations. You could choose to drop out higher-order interaction terms to reduce the equations to a square system: as many unknowns as equations. Then proceed to analyze the results from the least squares model as usual. There are actually slightly more sophisticated ways of dealing with this problem, as described by Norman Draper in "Missing Values in Response Surface Designs[124]", *Technometrics*, **3**, 389-398, 1961.

The above discussion illustrates clearly our preference for using the least squares model: whether the experimental design was executed accurately or not: the least squares model always works, whereas the *short cut tools* (page 243) developed for perfectly executed experiments will fail.

---

[124] https://www.jstor.org/stable/1266729

5.14.2 Handling of constraints

Most engineering systems have limits of performance, either by design or from a safety standpoint. It is also common that optimum production levels are found close to these constraints. The factorials we use in our experiments must, by necessity, span a wide range of operation so that we see systematic change in our response variables, and not merely measure noise. These large ranges that we choose for the factors often hit up again constraints.

A simple bioreactor example for 2 factors is shown: at high temperatures and high substrate concentrations we risk activating a different, undesirable side-reaction. The shaded region represents the constraint where we may not operate. We could for example replace the (T_+, C_+) experiment with two others, and then analyze these 5 runs using least squares.
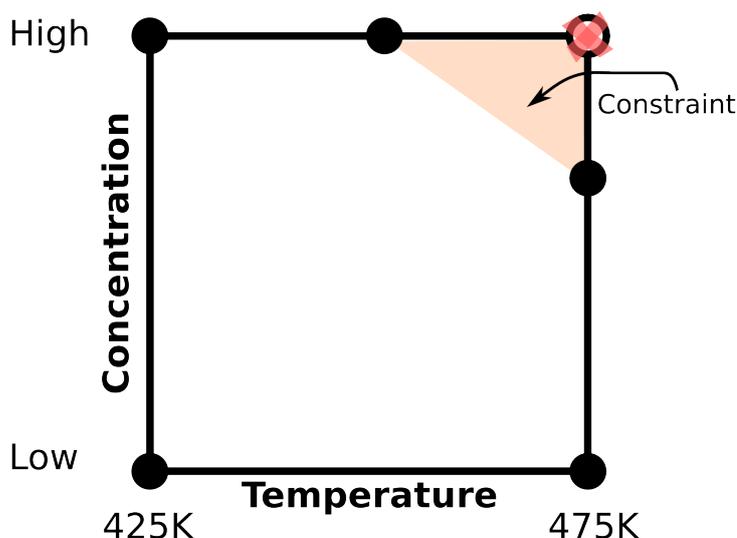
Unfortunately, these 5 runs do not form an orthogonal (independent) $\mathbf{X}$ matrix anymore. We have lost orthogonality. We have also reduced the space (or volume when we have 3 or more factors) spanned by the factorial design.

It is easy to find experiments that obey the constraints for 2-factor cases: run them on the corner points. But for 3 or more factors the constraints form planes that cut through a cube. We then use *optimal designs* (page 290) to determine where to place our experiments. A D-optimal design works well for constraint-handling because it finds the experimental points that would minimize the loss of orthogonality (i.e. they try to achieve the most orthogonal design possible). A compact way of stating this is to maximize the determinant of $\mathbf{X}^T\mathbf{X}$, which is why it is called D-optimal (it maximizes the determinant).

These designs are generated by a computer, using iterative algorithms. See the D-optimal reference in the *section on optimal designs* (page 290) for more information.

5.14.3 Optimal designs

If you delve into the modern literature on experimental methods you will rapidly come across the concept of an *optimal* design. This begs the question, what is sub-optimal about the factorial designs we have focussed on so far?

A full factorial design spans the maximal space possible for the k factors. From least squares modelling we know that large deviations from the model center reduces the variance of the parameter estimates. Furthermore, a factorial ensures the factors are moved independently, allowing us to estimate their effects independently as well. These are all "optimal" aspects of a factorial.

So again, what is sub-optimal about a factorial design? A factorial design is an excellent design in most cases. But if there are constraints that must be obeyed, or if the experimenter has an established list of possible experimental points to run, but must choose a subset from the list, then an "optimal" design is useful.

All an optimal design does is select the experimental points by optimizing some criterion, subject to constraints. Some examples:

- The design region is a cube with a diagonal slice cut-off on two corner due to constraints. What is the design that spans the maximum volume of the remaining cube?

- The experimenter wishes to estimate a non-standard model, e.g.
  $y = b_0 + b_A x_A + b_{AB} x_{AB} + b_B x_B + b_{AB} \exp^{-\frac{d x_A + e}{f x_B + g}}$ for fixed values of d, e, f and g.

- For a central composite design, or even a factorial design with constraints, find a smaller number of experiments than required for the full design, e.g. say 14 experiments (a number that is not a power of 2).

- The user might want to investigate more than 2 levels in each factor.

- The experimenter has already run n experiments, but wants to add one or more additional experiments to improve the parameter estimates, i.e. decrease the variance of the parameters. In the case of a D-optimal design, this would find which additional experiment(s) would most increase the determinant of the $\mathbf{X}^T\mathbf{X}$ matrix.

The general approach with optimal designs is

1. The user specifies the model (i.e. the parameters).

2. The computer finds all possible combinations of factor levels that satisfy the constraints, including center-points. These are now called the *candidate points* or candidate set, represented as a long list of all possible experiments. The user can add extra experiments they would like to run to this list.

3. The user specifies a small number of experiments they would actually like to run.

4. The computer algorithm finds this required number of runs by picking entries from the list so that those few runs optimize the chosen criterion.

The most common optimality criteria are:

- A-optimal designs minimize the average variance of the parameters, i.e. minimizes trace $\left\{ (\mathbf{X}^T\mathbf{X})^{-1} \right\}$

- D-optimal designs minimize the general variance of the parameters, i.e. maximize det $\left(\mathbf{X}^T\mathbf{X} \right)$

- G-optimal designs minimize the maximum variance of the predictions

- V-optimal designs minimize the average variance of the predictions

It must be pointed out that a full factorial design, 2^k is already A-, D- G- and V-optimal. Also notice that for optimal designs the user must specify the model required. This is actually no different to factorial and central composite designs, where the model is implicit in the design.

The algorithms used to find the subset of experiments to run are called candidate exchange algorithms. They are usually just a brute force evaluation of the objective function by trying all possible combinations. They bring a new trial combination into the set, calculate the objective value for the criterion, and then iterate until the final candidate set provides the best objective function value.

Readings

- St. John and Draper: "D-Optimality for Regression Designs: A Review[125]", *Technometrics*, **17**, 15-, 1975.

5.14.4 Definitive screening designs

The final type of design to be aware of is a class of designs called the definitive screening design, and below is a link that you can read up some more information.

These designs are a type of *optimal design* (page 290). Optimal designs can be very flexible. For example, if you had a limited budget you can create an optimal design for a given number of factors you are investigating to maximize one of these optimality criteria to fit your budget. A computer algorithm is used to find the settings for each one of the budgeted number of runs, so that the optimization criterion is maximized. In other words the computer is designing the experiments for you, so they have some very distinct advantages.

The readings below give more details, and a practical implementation of these designs using the R software package.

Readings

- John Lawson "DefScreen: Definitive Screening Designs, in package "daewr": Design and Analysis of Experiments with R[126]".

- Bradley Jones: "Class of Three-Level Designs for Definitive Screening in the Presence of Second-Order Effects[127]", Journal of Quality Technology, 2011.

5.14.5 Mixture designs

The area of mixture designs is incredibly important for optimizing recipes, particularly in the area of fine chemicals, pharmaceuticals, food manufacturing, and polymer processing. Like factorial designs, there are screening and optimization designs for mixtures also.

A mixture design is required when the factors being varied add up to 100% to form a mixture. Then these factors cannot be adjusted in an independent, factorial-like manner, since their proportion in the recipe must add to 100%: $\sum_i x_i = 1$. These designs result in triangular patterns (called simplexes). The experimental points at the 3 vertices are for pure components x_A, x_B, or x_C. Points along the sides represent a 2-component mixture, and points in the interior represent a 3-component blend.

---

[125] https://www.jstor.org/stable/1267995
[126] https://rdrr.io/cran/daewr/man/DefScreen.html
[127] https://yint.org/dsdesign

How a mixture design is just a constraint in an ordinary factorial
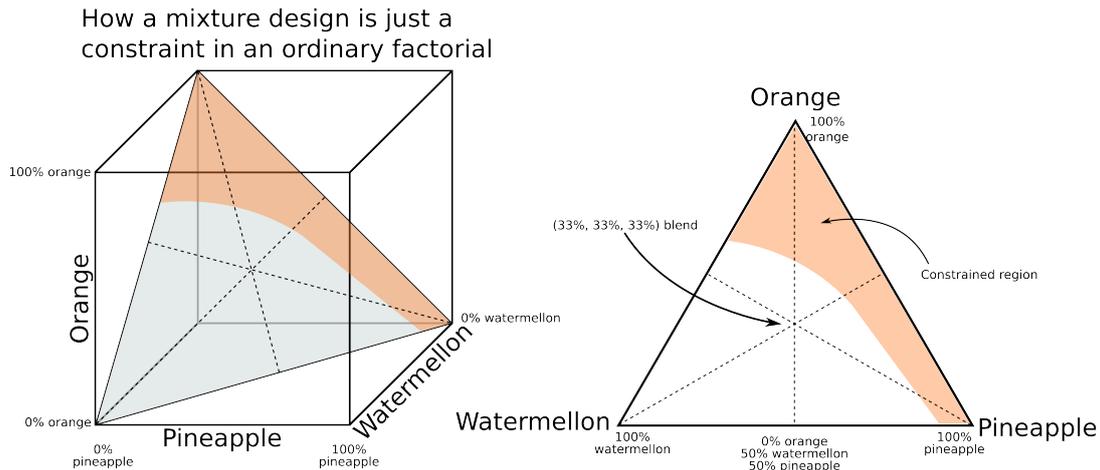
In the above figure on the right, the shaded region represents a constraint that cannot be operated in. A D-optimal algorithm must then be used to select experiments in the remaining region. The example is for finding the lowest cost mixture for a fruit punch, while still meeting certain taste requirements (e.g. watermelon juice is cheap, but has little taste). The constraint represents a region where the acidity is too high.

5.15 Exercises

Question 1

These readings are to illustrate the profound effect that designed experiments have had in some areas.

- Application of Statistical Design of Experiments Methods in Drug Discovery[128] and using DOE for high-throughput screening to locate new drug compounds[129].

- High traffic websites offer a unique opportunity to perform testing and optimization. This is because each visitor to the site is independent of the others (randomized), and these tests can be run in parallel. Read more in this brief writeup[130] on how Google uses testing tools to optimize YouTube, one of their web properties. Unfortunately they use the term "multivariate" incorrectly - a better term is "multi-variable"; nevertheless, the number of factors and combinations to be tested is large. It's well known that fractional factorial methods are used to analyze these data.

- See three chemical engineering examples of factorial designs in Box, Hunter, and Hunter: Chapter 11 (1st edition), or page 173 to 183 in the second edition.

---

[128] https://dx.doi.org/10.1016/S1359-6446(04)03086-7

[129] https://dx.doi.org/10.1016/1359-6446(96)10025-8

[130] https://youtube.googleblog.com/2009/08/look-inside-1024-recipe-multivariate.html

Question 2

Your family runs a small business selling low dollar value products over the web. They want to improve sales. There is a known effect from the day of the week, so to avoid that effect they run the following designed experiment every Tuesday for the past eight weeks. The first factor of interest is whether to provide free shipping over $30 or over $50. The second factor is whether or not the purchaser must first create a profile (user name, password, address, etc) before completing the transaction. The purchaser can still complete their transaction without creating a profile.

These are the data collected:

Date	Free shipping over . . .	Profile required before transaction	Total sales made
05 January 2010	$30	Yes	$ 3275
12 January 2010	$50	No	$ 3594
19 January 2010	$50	No	$ 3626
26 January 2010	$30	No	$ 3438
02 February 2010	$50	Yes	$ 2439
09 February 2010	$30	No	$ 3562
16 February 2010	$30	Yes	$ 2965
23 February 2010	$50	Yes	$ 2571

1. Calculate the average response from replicate experiments to calculate the 4 corner points.

2. Calculate and interpret the main effects in the design.

3. Show the interaction plot for the 2 factors.

4. We will show in the next class how to calculate confidence intervals for each effect, but would you say there is an interaction effect here? How would you interpret the interaction (whether there is one or not)?

5. What is the recommendation to increase sales?

6. Calculate the main effects and interactions by hand using a least squares model. You may confirm your result using software, but your answer should not just be the computer software output.

Solution

1. This is a 2^2 factorial system with a replicate at each point. We might not have *covered replicates* (page 254) in class by the time you had to do this assignment. So you should average the replicate points and then calculate the main effects and other terms for this system. You will get the same result if you analyze it as two separate factorials and then average the results - it's just more work that way though.

2. The experiment results in standard form with 4 corner points:

A	B	Average sales
−	−	$\frac{1}{2}(3438 + 3562) = \$3,500$
+	−	$\frac{1}{2}(3594 + 3626) = \$3,610$
−	+	$\frac{1}{2}(3275 + 2965) = \$3,120$
+	+	$\frac{1}{2}(2439 + 2571) = \$2,505$

where **A** = free shipping over \$30 (low level) and \$50 (high level), and let **B** $= -1$ if no profile is required, or $+1$ if a profile is required before completing the transaction.

- The main effect for free shipping (**A**) is $= \frac{1}{2}(3610 - 3500 + 2505 - 3120) = \dfrac{-505}{2} = -252.50$

  This indicates that sales decrease by \$252.50, on average, when going from free shipping over \$30 to \$50. One might expect, within reason, that higher sales are made when the free shipping value is higher (people add purchases so they reach the free shipping limit). That is shown by the very small effect of \$50 when no profile is required. However when a profile is required, we see the opposite: a drop in sales!

- The main effect of creating a profile **B** $= \frac{1}{2}(3120 - 3500 + 2505 - 3610) = \dfrac{-1485}{2} = -742.50$

  Indicating that sales drop by \$742.50 on average when requiring a profile before completing the transaction vs not requiring a profile. The drop in sales is less when offering free shipping over \$30 than when free shipping is for \$50 or more in purchases.

Not required for this question, but one of the best ways to visualize a small factorial, or a subset of a larger factorial, is with a cube plot:

3. The interaction plot which visually shows the main effects described above is:

Effect on sales
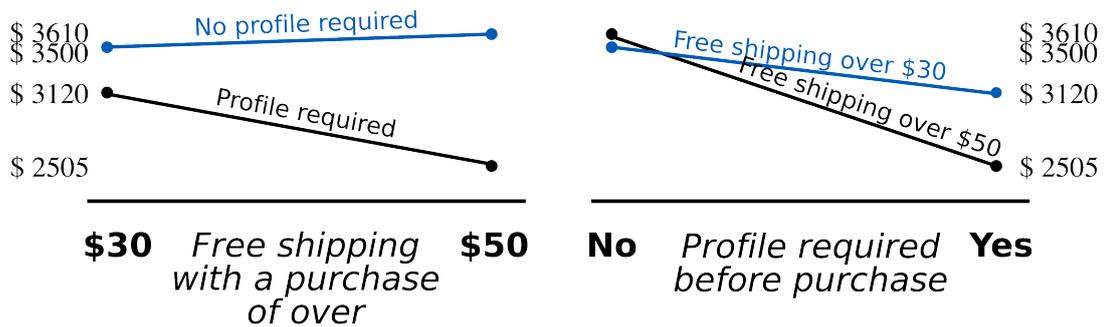
4. The interaction term can be calculated in two ways, both giving the same answer. Only one way is shown here:

   - **A** at high **B**: -$615.00

   - **A** at low **B**: $ 110.00

   - **AB** interaction = $\frac{1}{2}(-615 - 110) = \frac{-725}{2} = -362.50$

   This interaction term is larger than one of the main effects, so I would judge this to be important. Also, it is roughly 10% of the $y_i =$ daily sales values, so it is definitely important.

   In part 1 we showed the main effect of requiring a profile is to decrease sales. The strong negative interaction term here indicates that sales are even further reduced when free shipping is over $50, rather than $30. Maybe it's because customers "give up" making their purchase when free shipping is at a higher amount *and* they need to create a profile - perhaps they figure this isn't worth it. If they get free shipping over $30, the penalty of creating a profile is not as great anymore. This last effect might be considered counterintuitive - but I'm not an expert on why people buy stuff.

   In general, an interaction term indicates that the usual main effects are increased or decreased more or less than they would have been when acting on their own.

5. Sales can be definitely increased by not requiring the user to create a profile before completing the transaction (creating a profile is a strong deterrent to increasing sales, whether free shipping over $30 or $50 is offered). The effect of free shipping when not requiring a profile is small. The raw data for the case when no profile was required (below), show slightly higher sales when free shipping over $50 is required. Further experimentation to assess if this is significant or not would be required.

Date	Free shipping over …	Profile required before transaction	Total sales made that day
12 January 2010	$50	No	$ 3594
19 January 2010	$50	No	$ 3626
26 January 2010	$30	No	$ 3438
09 February 2010	$30	No	$ 3562

6. A least squares model can be calculated from the average of each replicate. Then there are 4 observations and 4 unknowns. Using the design matrix, in standard order, we can set up the

following least squares model:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 \\ 1 & -1 & +1 & -1 \\ 1 & +1 & +1 & +1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_{\mathbf{A}} \\ b_{\mathbf{A}} \\ b_{\mathbf{AB}} \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}$$

$$\begin{bmatrix} 3500 \\ 3610 \\ 3120 \\ 2505 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 \\ 1 & -1 & +1 & -1 \\ 1 & +1 & +1 & +1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_{\mathbf{A}} \\ b_{\mathbf{A}} \\ b_{\mathbf{AB}} \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}$$

And solving the regression coefficients (note the orthogonality in the $\mathbf{X}^T\mathbf{X}$ matrix):

$$\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

$$\mathbf{b} = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}^{-1} \begin{bmatrix} +3500 + 3610 + 3120 + 2505 \\ -3500 + 3610 - 3120 + 2505 \\ -3500 - 3610 + 3120 + 2505 \\ +3500 - 3610 - 3120 + 2505 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 12735 \\ -505 \\ -1485 \\ -725 \end{bmatrix}$$

$$\begin{bmatrix} b_0 \\ b_{\mathbf{A}} \\ b_{\mathbf{A}} \\ b_{\mathbf{AB}} \end{bmatrix} = \begin{bmatrix} 3184 \\ -126 \\ -371 \\ -181 \end{bmatrix}$$

The final model is $y = 3184 - 126x_{\mathrm{A}} - 371x_{\mathrm{B}} - 181x_{\mathrm{AB}}$.

Compare the values in the $\mathbf{X}^T\mathbf{y}$ vector to the calculations for the main effects and interactions to see the similarity. The least squares model parameters are half the size of the main effects and interactions reported above, because of how the parameters are interpreted in the least squares model.

Particularly the effect of requiring a profile, x_B, is to reduce sales by $2\times371 = 742$.

Question 3

More readings:

1. It is worth reading this paper by Bisgaard to see how the same tools shown in these notes were used to solve a real industrial problem: designed experiments, autocorrelation plots, data visualization, and quality control charts. Also he describes how the very real pressure from managers, time-constraints and interactions with team-members impacted the work.

   "The Quality Detective: A Case Study[131]" (and discussion), *Philosophical Transactions of the Royal Society A*, **327**, 499-511, 1989.

---

[131] https://dx.doi.org/10.1098/rsta.1989.0006

2. George Box, The R. A. Fisher Memorial Lecture, 1988, "Quality Improvement - An Expanding Domain for the Application of Scientific Method[132]", *Philosophical Transactions of the Royal Society - A*, **327**: pages 617-630, 1989.

Question 4

Note: This is a tutorial-type question: all the sub-questions build on each other. All questions deal with a hypothetical bioreactor system, and we are investigating four factors:

- **A** = feed rate: slow or medium

- **B** = initial inoculant size (300g or 700g)

- **C** = feed substrate concentration (40 g/L or 60 g/L)

- **D** = dissolved oxygen set-point (4mg/L or 6 mg/L)

The 16 experiments from a full factorial, 2^4, were randomly run, and the yields from the bioreactor, y, are reported here in standard order: y = [60, 59, 63, 61, 69, 61, 94, 93, 56, 63, 70, 65, 44, 45, 78, 77].

1. Calculate the 15 main effects and interactions and the intercept, using computer software.

2. Use a Pareto-plot to identify the significant effects. What would be your advice to your colleagues to improve the yield?

3. Refit the model using only the significant terms identified in the second question.

   - Explain why you don't actually have to recalculate the least squares model parameters.

   - Compute the standard error and confirm that the effects are indeed significant at the 95% level.

4. Write down the exact settings for **A**, **B**, **C**, and **D** you would provide to the graduate student running a half-fraction in 8 runs for this system.

5. Before the half-fraction experiments are even run you can calculate which variables will be confounded (aliased) with each other. Report the confounding pattern for these main effects and for these two-factor interactions. Your answer should be in this format:

   - Generator =

   - Defining relationship =

   - Confounding pattern:

     - $\widehat{\beta}_{\mathbf{A}} \rightarrow$

     - $\widehat{\beta}_{\mathbf{B}} \rightarrow$

     - $\widehat{\beta}_{\mathbf{C}} \rightarrow$

     - $\widehat{\beta}_{\mathbf{D}} \rightarrow$

     - $\widehat{\beta}_{\mathbf{AB}} \rightarrow$

     - $\widehat{\beta}_{\mathbf{AC}} \rightarrow$

     - $\widehat{\beta}_{\mathbf{AD}} \rightarrow$

---

[132] https://dx.doi.org/10.1098/rsta.1989.0017

- $\widehat{\beta}_{\mathbf{BC}} \rightarrow$

- $\widehat{\beta}_{\mathbf{BD}} \rightarrow$

- $\widehat{\beta}_{\mathbf{CD}} \rightarrow$

6. Now use the 8 yield values corresponding to your half fraction, and calculate as many parameters (intercept, main effects, interactions) as you can.

- Report their numeric values.

- Compare your parameters from this half-fraction (8 runs) to those from the full factorial (16 runs). Was much lost by running the half fraction?

- What was the resolution of the half-fraction?

- What is the projectivity of this half-fraction? And what does this mean in light of the fact that factor **A** was shown to be unimportant?

- Factor **C** was found to be an important variable from the half-fraction; it had a significant coefficient in the linear model, but it was aliased with **ABD**. Obviously in this problem, the foldover set of experiments to run would be the *other half-fraction*. But we showed a way to de-alias a main effect. Use that method to show that the other 8 experiments to de-alias factor **C** would just be the other 8 experiment not included in your first half-fraction.

Question 5

Your group is developing a new product, but have been struggling to get the product's stability, measured in days, to the level required. You are aiming for a stability value of 50 days or more. Four factors have been considered:

- **A** = monomer concentration: 30% or 50%

- **B** = acid concentration: low or high

- **C** = catalyst level: 2% or 3%

- **D** = temperature: 393K or 423K

These eight experiments have been run so far:

Experiment	Order	A	B	C	D	Stability
1	5	−	−	−	−	40
2	6	+	−	−	+	27
3	1	−	+	−	+	35
4	4	+	+	−	−	21
5	2	−	−	+	+	39
6	7	+	−	+	−	27
7	3	−	+	+	−	27
8	8	+	+	+	+	20

Where would you run the next experiment to try get the stability above 50 or greater?

Question 6

The following diagram shows data from a central composite design. The factors were run at their standard levels, and there were 4 runs at the center point.

1. Calculate the parameters for a suitable quadratic model in these factors. Show your matrices for $\mathbf{X}$ and $\mathbf{y}$.

2. Draw a response surface plot of $\mathbf{A}$ *vs* $\mathbf{B}$ over a suitably wide range beyond the experimental region.

3. Where would you move $\mathbf{A}$ and $\mathbf{B}$ if your objective is to increase the response value?

   1. Report your answer in coded units.

   2. Report your answer in real-world units, if the full factorial portion of the experiments were ran at:

      - $\mathbf{A}$ = *stirrer speed*, 200rpm and 340 rpm

      - $\mathbf{B}$ = *stirring time*, 30 minutes and 40 minutes
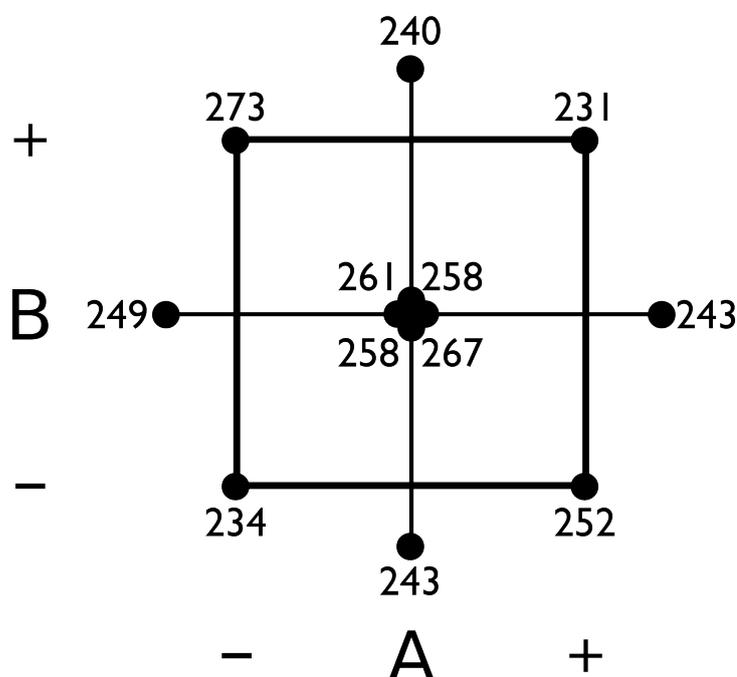
You might feel more comfortable setting up the problem in MATLAB. You can use the contour plot[133] functions in MATLAB to visualize the results.

If you are using R, you can use the `rbind(...)` or `cbind(...)` functions to build up your $\mathbf{X}$ matrix row-by-row or column-by-column. The equivalent of meshgrid in R is the `expand.grid(...)` function. See the R code on the course website[134] that shows how to generate surface plots in R.

---

[133] https://www.mathworks.com/help/matlab/ref/contour.html
[134] https://learnche.org/4C3/Design_and_analysis_of_experiments_(2014)

Question 7

A full 2^3 factorial was run as shown:

Experiment	A	B	C
1	30%	232	Larry
2	50%	232	Larry
3	30%	412	Larry
4	50%	412	Larry
5	30%	232	Terry
6	50%	232	Terry
7	30%	412	Terry
8	50%	412	Terry

- What would be the D-optimal objective function value for the usual full 2^3 factorial model?

- If instead experiment 2 was run at (A,B,C) = (45%, 200, Larry), and experiment 3 run at (A, B, C) = (35%, 400, Larry); what would be the D-optimal objective function value?

- What is the ratio between the two objective function values?

Solution

- The D-optimal objective function is to maximize the determinant of the design matrix, i.e. $\det\left(\mathbf{X}^T\mathbf{X}\right)$.

  Since this is a full factorial in 3 factors, with all runs perfectly at the -1 and $+1$ levels, then the determinant is the product of the diagonal entries and is $8^8 = 16777216$. In MATLAB, this would be `det(eye(8) * 8)`.

- Assuming the columns in $\mathbf{X}$ are in the order of [intercept, **A**, **B**, **C**, **AB**, **AC**, **BC**, **ABC**], then row 2 in matrix $\mathbf{X}$ would be $[1, 0.5, -1.35, -1, -0.675, -0.5, 1.35, 0.675]$ and row 3 would be $[1, -0.5, 0.867, -1, -0.4333, 0.5, -0.867, 0.4333]$

  The determinant with these two rows replaced in now 6.402×10^6.

- The ratio is $\frac{6.402 \times 10^6}{16777216} = 0.38$, a fairly large reduction in the objective.

Question 8

In your start-up company you are investigating treatment options for reducing the contamination level of soil that has been soaked with hydrocarbon products. You have two different heaps of contaminated soil from two different sites. You expect your treatment method to work on any soil type though.

Your limited line of credit allows only 9 experiments, even though you have identified at least 6 factors which you expect to have an effect on the treatment.

1. Write out the set of experiments that you believe will allow you to learn the most relevant information, given your limited budget. Explain your thinking, and present your answer with 7 columns: 6 columns showing the settings for the 6 factors and one column for the heap from which the test sample should be taken. There should be 9 rows in your table.

2. What is the projectivity and resolution of your design?

Solution

1. When given a constraint on the number of experiments, we would like to examine the highest number of factors, but with the lowest tradeoff in the associated resolution.

   There are 6 factors to examine. As stated, we would like our treatment method to work on *any* contaminated soil sample, however we have testing soil only from 2 sites. This is a blocking variable, since we might expect differences due to the site where the soil came from, but we want it to have the least possible effect on our results.

   An alternative way to view this problem is to assume that soil is an extra factor in the experiment, but when choosing the generators, we will associate it with the highest level of confounding possible. This latter interpretation makes it easier to use the table in the notes.

   Using the *table in the notes* (page 270), and looking down the column with 7 factors, we are constrained to the cell with 8 experiments, since the next cell down has 16 experiments, which is too many. So a 2_{III}^{7-4} design would be most appropriate.

   We would write out our usual 2^3 full factorial, then assign **D=AB**, **E=AC**, **F=BC** and **G=ABC**. We will let that last factor be the heap of soil factor, as it has the highest level of confounding.

   We can run a 9th experiment. In this case, I would put all variables at the center point (if they are continuous), and use a 50/50 blend of the two soil samples. Also, I would run this experiment first, to iron out any experimental protocol issues that I will didn't think of; rather discover them on this first run, which can be discarded in the analysis later on.

   Alternatively, if I'm confident with my experimental procedure, I can choose to do experiment 9 last, if at all, as a replicate of any interesting previous experiment that gives an unexpected (good or bad) result.

   A table for the experiments would be:

Experiment	A	B	C	D=AB	E=AC	F=BC	G=ABC
1	−	−	−	+	+	+	Heap 1
2	+	−	−	−	−	+	Heap 2
3	−	+	−	−	+	−	Heap 2
4	+	+	−	+	−	−	Heap 1
5	−	−	+	+	−	−	Heap 2
6	+	−	+	−	+	−	Heap 1
7	−	+	+	−	−	+	Heap 1
8	+	+	+	+	+	+	Heap 2
9	0	0	0	0	0	0	50/50

2. The design has resolution = $R = 3$, from the table in the notes. The projectivity is $R - 1 = 2$.

Question 9

A factorial experiment was run to investigate the settings that minimize the production of an unwanted side product. The two factors being investigated are called **A** and **B** for simplicity, but are:

- **A** = reaction temperature: low level was 420 K, and high level was 440 K

- **B** = amount of surfactant: low level was 10 kg, high level was 12 kg

A full factorial experiment was run, randomly, on the same batch of raw materials, in the same reactor. The system was run on two different days though, and the operator on day 2 was a different person. The recorded amount, in grams, of the side product was:

Experiment	Run order	Day	A	B	Side product formed
1	2	1	420 K	10 kg	89 g
2	4	2	440 K	10 kg	268 g
3	5	2	420 K	12 kg	179 g
4	3	1	440 K	12 kg	448 g
5	1	1	430 K	11 kg	196 g
6	6	2	430 K	11 kg	215 g

1. What might have been the reason(s) for including experiments 5 and 6?

2. Was the blocking for a potential day-to-day effect implemented correctly in the design? Please show your calculations.

3. Write out a model that will predict the amount of side product formed. The model should use coded values of **A** and **B**. Also write out the **X** matrix and **y** vector that can be used to estimate the model coefficients using the equation $b = (X'X)^{-1} X'y$.

4. Solve for the coefficients of your linear model, either by using $b = (X'X)^{-1} X'y$ directly, or by some other method.

5. Assuming the blocking for the day-to-day effect was implemented correctly, does your model show whether this was an important effect on the response or not? Explain your answer.

6. You have permission to run two further experiments to find an operating point that reduces the unwanted side product. Where would you place your next two runs, and show how you select these values. Please give your answer in the original units of **A** and **B**.

7. As you move along the response surface, performing new experiments to approach the optimum, how do you know when you are reaching an optimum? How does your experimental strategy change? Please give specific details, and any model equations that might help illustrate your answer.

Solution

1. Experiments 5 and 6 from the standard order might have been included as baseline experiments, since they appear at the center point for factors **A** and **B**.

   These two runs give 2 degrees of freedom as well, which helps with estimating confidence intervals on the least squares parameters.

Also, since one of them was performed first, it could have been used to establish the experimental workflow. In other words, the experiment was used to see how to run the experiment the first time. If things go horribly wrong, then this data point can just be discarded. If we had started with a corner of the factorial, we would have had to repeat that experiment if it failed, or if it succeeded, had a duplicate experiment at the one corner but not the others.

Finally, it could also have been used to assess the effect of the operators, since runs 5 and 6 are identical, though in this case runs 5 and 6 are on different days, so it could be the day-to-day being measured here.

2. Yes. If we consider the day effect to be a new factor, **C**, then we could runs 1 to 4 as a half fraction in 3 factors. The least disruptive generator would be **C = AB**. Using this we can see that runs 1 and 4 should be run on one day, and runs 2 and 3 on the next day: this is what was done. The center points can be run on either day, and in this case one was run on each day.

Using this generator confounds the interaction effect, **AB** with the day-to-day (and operator-to-operator) effect. We can never clear up that confounding with this set of experiments.

3. The model would have the form:

$$y = b_0 + b_A x_A + b_B x_B + b_{AB} x_{AB} + e$$

The matrices and vectors to solves this least squares model are:

$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} =
\begin{bmatrix}
1 & -1 & -1 & +1 \\
1 & +1 & -1 & -1 \\
1 & -1 & +1 & -1 \\
1 & +1 & +1 & +1 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} b_0 \\ b_A \\ b_B \\ b_{AB} \end{bmatrix} +
\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{bmatrix}
$$

$$
\begin{bmatrix} 89 \\ 268 \\ 179 \\ 448 \\ 196 \\ 215 \end{bmatrix} =
\begin{bmatrix}
1 & -1 & -1 & +1 \\
1 & +1 & -1 & -1 \\
1 & -1 & +1 & -1 \\
1 & +1 & +1 & +1 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} b_0 \\ b_A \\ b_B \\ b_{AB} \end{bmatrix} +
\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{bmatrix}
$$

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$$

4. Using the above matrices we can calculate $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'y}$, even by hand!

$$
\mathbf{X'X} =
\begin{bmatrix}
6 & 0 & 0 & 0 \\
0 & 4 & 0 & 0 \\
0 & 0 & 4 & 0 \\
0 & 0 & 0 & 4
\end{bmatrix}
$$

$$
\mathbf{X'y} =
\begin{bmatrix}
89 + 268 + 179 + 448 + 196 + 215 \\
-89 + 268 - 179 + 448 \\
-89 - 268 + 179 + 448 \\
+89 - 268 - 179 + 448
\end{bmatrix} =
\begin{bmatrix} 1395 \\ 448 \\ 270 \\ 90 \end{bmatrix}
$$

$$
\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'y} =
\begin{bmatrix} 232.5 \\ 112 \\ 67.5 \\ 22.5 \end{bmatrix}
$$

5. The above least squares solution shows the two main effects are large: 112 and 67.5 for a one unit change (coded units). Relative to these two, the interaction term of $b_{AB} = 22.5$ is small. This implies the day-to-day effect (which is confounded with the operator effect) is small.

6. A new run in **A** and **B** would be at *lower* values of **A** and **B**, since we want to reduce the side product. We will make a move from the baseline point by reducing factor **A** by 1 unit, and then ratio that with the necessary change in **B** to go down the direction of steepest descent:

$$\Delta x_A = -1$$

$$\Delta x_{A,\text{actual}} = -10\,\text{K}$$

$$\Delta x_B = \frac{b_B}{b_A}\Delta x_A = \frac{67.5}{112}\Delta x_A$$

but we know that
$$\Delta x_B = \frac{x_{B,\text{actual}}}{\Delta_B/2}$$

$$\Delta x_{B,\text{actual}} = \frac{b_B}{b_A}\Delta x_A \times \Delta_B/2 \text{ by equating previous 2 lines}$$

$$\Delta x_{B,\text{actual}} = \frac{67.5}{112} \times (-1) \times 2\text{kg}/2$$

$$\Delta x_{B,\text{actual}} = -\mathbf{0.60}\text{ kg}$$

Note that $\Delta_B \neq \Delta x_B$. The former is the range for factor **B**, the latter is the amount by which we change factor **B** from the baseline. So the new settings for the next experiment would be at:

- **A** = 430 - 10 = 420 K

- **B** = 11 − 0.60 = 10.4 kg

7. An optimum is present in a factorial experiment if you notice that:

- interaction terms start to become large,

- the center point in the factorial has higher/lower values than any of the corner points (remember that with an optimum you are the peak or the valley)

- curvature terms, i.e. quadratic terms, in the model are larger than the main effect.

The experimental strategy changes by included axial points into the factorial design, allowing one to calculate the quadratic terms in the model, such as a $b_{AA}x_A^2$ term for the quadratic effect of factor **A**.

Question 10

Adapted from Box, Hunter and Hunter

A liquid polymer formulation is being made that is applied as a polish to wood surfaces. The group responsible for the product have identified 3 elements to the formulation that have an effect of the liquid polish's final quality attributes (FQAs: this acronym is becoming a standard in most companies these days).

- **A**: amount of reactive monomer in the recipe (10% at the low level and 30% at the high level)

- **B**: the amount of chain length regulator (1% at the low level and 4% at the high level)

- **C**: the type of chain length regulator (regulator P at the − level or regulator Q at the + level)

In class we have focused on the case where our y-variable is continuous, but it could also be *descriptive*. In this question we also see what happens when we have more than one y-variable.

- y_1 = Milky appearance: either *Yes* or *No*

- y_2 = Viscous: either *Yes* or *No*

- y_3 = Yellow colour: either *No* or *Slightly*

The following table captures the 8 experiments in standard order, although the experiments were run in a randomized order.

Experiment	A	B	C	y_1	y_2	y_3
1	−	−	P	Yes	Yes	No
2	+	−	P	No	Yes	No
3	−	+	P	Yes	No	No
4	+	+	P	No	No	No
5	−	−	Q	Yes	Yes	No
6	+	−	Q	No	Yes	Slightly
7	−	+	Q	Yes	No	No
8	+	+	Q	No	No	Slightly

1. What is the cause of a milky appearance?

2. What causes a more viscous product?

3. What is the cause of a slight yellow appearance?

4. Which conditions would you use to create a product was *not* milky, was of low viscosity, and had no yellowness?

5. Which conditions would you use to create a product was *not* milky, was of low viscosity, and had some yellowness?

Solution

Tables are often frowned on by people, but the reality is they are sometimes one of the best forms of visualizing data. In this example we see:

1. The milky appearance is caused by low levels of **A** = amount of reactive monomer (10% in this recipe), since milky appearance is correlated with that column.

2. A more viscous product is caused by low levels of **B** = amount of chain length regulator (1% in this recipe), since the change in signs in **B** match the viscous column.

3. The yellow appearance is due to an interaction: in this case only when using chain length regulator Q *and* when using high levels of reactive monomer in the recipe (30%) do we see the yellow appearance.

4. Such a product can be obtained by using

   - **A** = high amount of reactive monomer in the recipe (30%)

   - **B** = high amounts of chain length regulator (4%)

   - **C** = use chain length regulator P

   These correspond to conditions in experiment 4.

5. Such a product can be obtained by using

   - **A** = high amount of reactive monomer in the recipe (30%)

- **B** = high amounts of chain length regulator (4%)

- **C** = use chain length regulator Q

These correspond to conditions in experiment 8.

In all these questions we can conclusively state there is cause and effect, since we see repeated changes in the factors (holding the other variables and disturbances constant) and the corresponding effects in the 3 y-variables.

Question 11

Using a 2^3 factorial design in 3 variables (**A** = temperature, **B** = pH and **C** = agitation rate), the conversion, y, from a chemical reaction was recorded.

Experiment	A	B	C	y
1	−	−	−	72
2	+	−	−	73
3	−	+	−	66
4	+	+	−	87
5	−	−	+	70
6	+	−	+	73
7	−	+	+	67
8	+	+	+	87

- **A** = $\dfrac{\text{temperature} - 150°\text{C}}{10°\text{C}}$

- **B** = $\dfrac{\text{pH} - 7.5}{0.5}$

- **C** = $\dfrac{\text{agitation rate} - 50\text{rpm}}{5\text{rpm}}$

1. Show a cube plot for the recorded data.

2. Estimate the main effects and interactions by hand.

3. Interpret any results from part 2.

4. Show that a least squares model for the full factorial agrees with the effects and interactions calculated by hand.

5. Approximately, at what conditions (given in real-world units), would you run the next experiment to improve conversion. Give your settings in coded units, then unscale and uncenter them to get real-world units.

Question 12

1. Why do we block groups of experiments?

2. Write a 2^3 factorial design in two blocks of 4 runs, so that no main effect or 2 factor interaction is confounded with block differences.

Solution

1. When performing experiments in groups, for example, half the experiments are run on day one and the others on day 2, we must block the experiments we choose to run on each day, to avoid inadvertently introducing a new effect, a day-to-day effect in the model. In other words, we must choose in a careful way the half group of runs we place on day 1 and day 2.

   Blocking applies in many other cases: sometimes we have to use two batches of raw materials to do an experiment, because there is not enough for the full factorial. We must block to prevent the effect of raw materials to affect our y-variable.

   Or to run the experiments efficiently in a short time, we choose to do them in parallel in two different reactors. Here we must block against the reactor effect.

2. For a 2^3 system we have factors **A**, **B** and **C**. To avoid the main effect being confounded with any 2 factor interactions we must assign the blocks to the highest interaction, i.e. the **ABC** interaction.

   Writing out the design in standard order:

Experiment	A	B	C	ABC
1	−	−	−	−
2	+	−	−	+
3	−	+	−	+
4	+	+	−	−
5	−	−	+	+
6	+	−	+	−
7	−	+	+	−
8	+	+	+	+

   This table indicates we should do all experiments in column **ABC** with a − in one block, and the experiments with a + should be done in the second block. The main effects will not be confounded with any 2-factor interactions in this case.

   Another way you can interpret blocking is as follows. Consider the block to be a new factor in your experiment, call it factor **D**, where **D** at the low level corresponds to experiments in the first block, and **D** at the high level would be experiments in the second block.

   But we can only run 8 experiments, so we now use the table in the course notes (derived from page 272 in Box, Hunter and Hunter, 2nd edition), and see the layout that will cause least disruption is to assign **D = ABC**. This gives the same experimental layout above.

Question 13

Factors related to the shrinkage of plastic film, produced in an injection molding device, are being investigated. The following factors have been identified by the engineer responsible:

- **A** = mold temperature

- **B** = moisture content

- **C** = holding pressure

- **D** = cavity thickness

- **E** = booster pressure

- **F** = cycle time

- **G** = gate size

Experiment	A	B	C	D	E	F	G	y
1	−	−	−	+	+	+	−	14.0
2	+	−	−	−	−	+	+	16.8
3	−	+	−	−	+	−	+	15.0
4	+	+	−	+	−	−	−	15.4
5	−	−	+	+	−	−	+	27.6
6	+	−	+	−	+	−	−	24.0
7	−	+	+	−	−	+	−	27.4
8	+	+	+	+	+	+	+	22.6

You can obtain a copy of this data set if you install the BsMD package in R. Then use the following commands:

```
library(BsMD)
data(BM93.e3.data)

# Use only a subset of the original experiments
X <- BM93.e3.data[1:8, 2:10]
```

1. How many experiments would have been required for a full factorial experiment?

2. What type of fractional factorial is this (i.e. is it a half fraction, quarter fraction . . .)?

3. Identify all the generators used to create this design. A table, such as on page 272 in Box, Hunter and Hunter, 2nd edition will help.

4. Write out the complete defining relationship.

5. What is the resolution of this design?

6. Use a least squares approach to calculate a model that fits these 8 experiments.

7. What effects would you judge to be significant in this system? The engineer will accept your advice and disregard the other factors, and spend the rest of the experimental budget only on the factors deemed significant.

8. What are these effects aliased with (use your defining relationship to find this).

9. Why is in necessary to know the confounding pattern for a fractional factorial design.

Solution

1. There are 7 factors in this experiment, so a full factorial would require $2^7 = 128$ experiments.

2. This is a one-sixteenth fraction, $8/128 = 1/16$.

3. Since the are 7 factors in 8 runs, the *DOE tradeoff table* (page 270) indicates the possible generators are **D = AB**, **E = AC**, **F = BC** and **G = ABC**. However, that doesn't mean the experiments were generated with exactly those factors. For example, these experiments could have interchanged the **A** and **B** columns, in which case factors **E** and **F** would be different.

   However, when checking the columns in our table against these generators we see that the experiments were derived from exactly these same generators. It is customary to record the generators in the form **I = . . .** , so our generators are:

   - **I = ABD**

   - **I = ACE**

   - **I = BCF**

   - **I = ABCG**.

4. The defining relationship is the product of all possible generator combinations. Since there are 4 generators, there are 2^4 words in the defining relationship. A similar example in the course notes shows that the defining relationship is:

   I = ABD = ACE = BCF = ABCG = BCDE = ACDF = CDG = ABEF = BEG = AFG = DEF = ADEG = CEFG = BDFG = ABCDEFG

5. It is a resolution III design, by noting the shortest word in the defining relationship is of length 3 (and verified in the table above).

6. The least squares model would be found by setting $- = -1$ and $+ = +1$ in the table above as the **X** matrix, and adding an additional column of 1's to account for the intercept. This gives a total of 8 columns in the matrix. The $\mathbf{X}^T\mathbf{X}$ will be diagonal, with 8's on the diagonal. The **y** vector is just the table of results above.

   From this we calculate $\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ (MATLAB and R code is given at the end).

   $$y = 20.35 - 0.65x_A - 0.25x_B + 5.05x_C - 0.45x_D - 1.45x_E - 0.15x_F + 0.15x_G$$

7. From this we judge effect **C**, **E** and to a smaller extent, effect **A**, to be significant.

8. However, these main effects are aliased with:

   - **C** (multiply **C** by every word in the defining relationship)

     – **CABD = ABCD**

     – **CACE = AE**

     – **CBCF = BF**

     – **CABCG = ABG**

     – **CBCDE = BDE**

     – **CACDF = ADF**

5.15. Exercises **309**

- - **CCDG = DG**

  - **CABEF = ABCEF**

  - **CBEG = CBEG**

  - **CAFG = ACFG**

  - **CDEF = CDEF**

  - **CADEG = ACDEG**

  - **CCEFG = EFG**

  - **CBDFG = BCDFG**

  - **CABCDEFG = ABDEFG**

- **E** (reporting only the 2 factor interactions)

  - **AC**

  - **BG**

  - **DF**

- **A** (reporting only the 2 factor interactions)

  - **BD**

  - **CE**

  - **FG**

9. It is necessary to know the confounding pattern because it helps to interpret the coefficients. For example, we see that factor **C** is aliased with the **AE** interaction, and we also see that factors **A** and **E** are important. We cannot be sure though if that large coefficient for **C** is due purely to **C**, or if it is also due to the **AE** interaction.

   The only way we can uncouple that coefficient is by performing additional, *foldover* experiments.

The R code for this question are given below, and also code to draw the Pareto plot to determine the most important coefficients.

```
A <- B <- C <- c(-1, 1)
d <- expand.grid(A=A, B=B, C=C)
y <- c(14.0, 16.8, 15.0, 15.4, 27.6, 24.0, 27.4, 22.6)
A <- d$A
B <- d$B
C <- d$C
D <- A*B
E <- A*C
F <- B*C
G <- A*B*C
model <- lm(y ~ A + B + C + D + E + F + G)
summary(model)

coeff <- coef(model)[2:length(coef(model))]

# Pareto plot of the absolute coefficients
library(lattice)
bitmap('fractional-factorial-question.png', type="png256",
       width=6, height=6, res=300, pointsize=14)
barchart(sort(abs(coeff)), xlab="Magnitude of effect",
```

(continues on next page)

```
        ylab = "Effect", col=0)
dev.off()
```
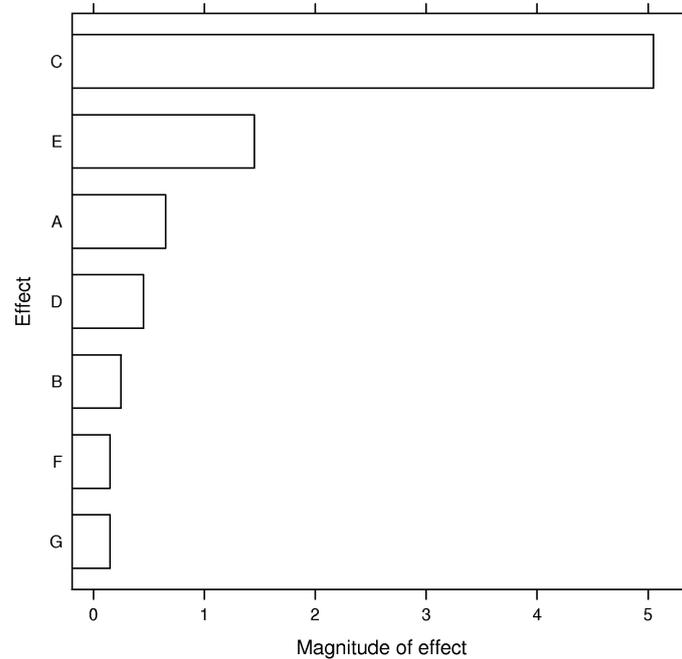

Question 14

One of the experiment projects investigated by a previous student of this course was understanding effects related to the preparation of uncooked, breaded chicken strips.

The student investigated these 3 factors in a full factorial design [*]:

- **D** = duration: low level at 15 minutes; and high level = 22 minutes.

- **R** = position of oven rack: low level = use middle rack; high level = use low oven rack (this coding, *though unusual*, was used because the lower rack applies more heat to the food).

- **P** = preheated oven or not: low level = short preheat (30 seconds); high level = complete preheating.

[*] The student actually investigated 4 factors, but found the effect of oven temperature to be negligible!

The response variable was y = taste, the average of several tasters, with higher values being more desirable.

Experiment	D	R	P	Taste
1	–	–	–	3
2	+	–	–	9
3	–	+	–	3
4	+	+	–	7
5	–	–	+	3
6	+	–	+	10
7	–	+	+	4
8	+	+	+	7

A full factorial model, using the usual coding, was calculated from these 8 experiments:

$$y = 5.75 + 2.5x_D - 0.5x_R + 0.25x_P - 0.75x_Dx_R - 0.0x_Dx_P - 0.0x_Rx_P - 0.25x_Dx_Rx_P$$

1. What is the physical interpretation of the $+2.5x_D$ term in the model?

2. From the above table, at what real-world conditions should you run the system to get the highest taste level?

3. Does your previous answer match the above model equation? Explain, in particular, how the non-zero *two factor* interaction term affects taste, and whether the interaction term reinforces the taste response variable, or counteracts it, when the settings you identified in part 2 are used.

4. If you decided to investigate this system, but only had time to run 4 experiments, write out the fractional factorial table that would use factors **D** and **R** as your main effects and confound factor **P** on the **DR** interaction.

   Now add to your table the response column for taste, extracting the relevant experiments from the above table.

   Next, write out the model equation and estimate the 4 model parameters from your reduced set of experiments. Compare and comment on your model coefficients, relative to the full model equation from all 8 experiments.

Question 15

Your company is developing a microgel-hydrogel composite, used for controlled drug delivery with a magnetic field. A previous employee did the experimental work but she has since left the company. You have been asked to analyze the existing experimental data.

- Response variable: y = sodium fluorescein (SF) released [mg], per gram of gel

- The data collected, in the original units:

Experiment	Order	**M** = microgel weight [%]	**H** = hydrogel weight [%]	y
1	4	4	10	119
2	1	8	10	93
3	6	4	16	154
4	3	8	16	89
5	2	6	13	85
6	5	6	13	88
7	9	3.2	13	125
8	7	8.8	13	111
9	10	6	17.2	136
10	8	6	8.8	98

1. What was likely the reason the experimenter added experiments 5 and 6?

2. Why might the experimenter have added experiments 7, 8, 9 and 10 after the first six? Provide a rough sketch of the design, and all necessary calculations to justify your answer.

3. What is the name of the type of experimental design chosen by the employee for *all 10 experiments in the table*?

4. Using these data, you wish to estimate a nonlinear approximation of the response surface using a model with quadratic terms. Write out the equation of such a model that can be calculated from these 10 experiments (*also read the next question*).

5. Write out

   - the $\mathbf{X}$ matrix,

   - the corresponding symbolic entries in $\mathbf{b}$

   - and the $\mathbf{y}$ vector

   that you would use to solve the equation $\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ to obtain the parameter estimates of the model you proposed in the previous part. You must use data from all 10 experiments.

6. How many degrees of freedom will be available to estimate the standard error and confidence intervals?

Question 16

Biological drugs are rapidly growing in importance in the treatment of certain diseases, such as cancers and arthritis, since they are designed to target very specific sites in the human body. This can result in treating diseases with minimal side effects. Such drugs differ from traditional drugs in the way they are manufactured – they are produced during the complex reactions that take place in live cell culture. The cells are grown in lab-scale bioreactors, harvested, purified and packaged.

These processes are plagued by low yields which makes these treatments very costly. Your group has run an initial set of experiments to learn more about the system and find better operating conditions to boost the yield. The following factors were chosen in the usual factorial manner:

- **G** = glucose substrate choice: a binary factor, either **Gm** at the low level code or **Gp** at the high level.

- **A** = agitation level: low level = 10 rpm and high level = 20 rpm, but can only be set at integer values.

- **T** = growth temperature: 30°C at the low level, or 36°C at the high level, and can only be set at integer values in the future, with a maximum value of 40°C.

- **C** = starting culture concentration: low level = 1100 and high level = 1400, and can only be adjusted in multiples of 50 units and within a range of 1000 to 2000 units.

A fractional factorial in 8 runs at the above settings, created by aliasing **C = GAT**, gave the following model in coded units:

$$y = 24 + 3x_G - 1.0x_A + 4.0x_T - 0.2x_G x_A - 0.79x_G x_T - 0.25x_A x_T + 3.5x_G x_A x_T$$

The aim is to find the next experiment that will improve the yield, measured in milligrams, the most.

1. What settings might have been used for the *baseline conditions* for this factorial experiment?

2. What is the resolution of this design?

3. Using the method of steepest ascent, state all reasonable assumptions you need to find the experimental conditions for **all 4 factors** for the next experiment. Give these 4 conditions in both the real-world units, as well as in the usual coded units of the experiment. Note however that your manager has seen that temperature has a strong effect on yield, and he has requested the next experiment be run at 40°C.

4. Report the expected yield at these proposed experimental conditions.

Solution

1. Baseline conditions are at **G = Gm** or **Gp** (either would work), **A** at 15 rpm, **T** at 30°C, and **C** at 1250 concentration units.

2. It is a four factor experiment, with 8 runs; from the table, for the given aliasing, it is a resolution IV design.

3. We assume that we can ignore all 2fi and 3fi - i.e. that they are small. Specifically, this implies that the 3.5 coefficient is for **C** and not for the product of $x_G x_A x_T$

   - Fix temperature at 40°C, implying that $T^{(next)} = 40°C$ and $x_T^{(next)} = \frac{40-33}{3} = 2.33$.

   - Factor **G** must be run at the highest level possible, i.e. **G = Gp**

   - Factor **A** must be run at a lower level, specifically $\Delta A = -0.25 \times 2.33 = -0.583$, or a deviation of -2.9 rpm from the baseline. Since we have to use integer values, that implies $A^{(next)} = 12$ rpm and $x_A^{(next)} = \frac{12-15}{5} = -0.6$.

   - Factor **C** must be run at a higher level, specifically $\Delta C = 3.5/4 \times 2.33 = 2.04$, or a deviation of +306 in actual units from the baseline. Since we have to round to the closest 50, that implies $C^{(next)} = 1550$ rpm and $x_C^{(next)} = \frac{1550-1250}{150} = +2$.

4. The predicted yield can be found by substituting the coded values into the model equation, choosing to either use or ignore the small interactions:

   With the interactions:

   $$\begin{aligned} y &= 24 + 3(+1) - 1.0(-0.6) + 4.0(2.33) - 0.2(+1)(-0.6) - 0.79(+1)(2.33) - 0.25(-0.6)(2.33) + 3.5(+2) \\ y &= \mathbf{42.3} \end{aligned}$$

   Without interactions:

   $$y = 24 + 3(+1) - 1.0(-0.6) + 4.0(2.33) + 3.5(+2) = 43.9$$

6.1 In context

This section considers the important area of latent variable modelling. These models have been shown, about 20 to 30 years ago, to be very powerful tools in dealing with the very data that (chemical) engineers face frequently. Our main goal of this section is to show how one can extract value from these data. But we first introduce the concept of a latent variable, and specifically the principal component analysis (PCA) model: the cornerstone of all latent variable models. Then we consider different ways to use our databases for interesting applications such as troubleshooting, soft-sensors, process monitoring, and new product development.

6.1.1 What we will cover
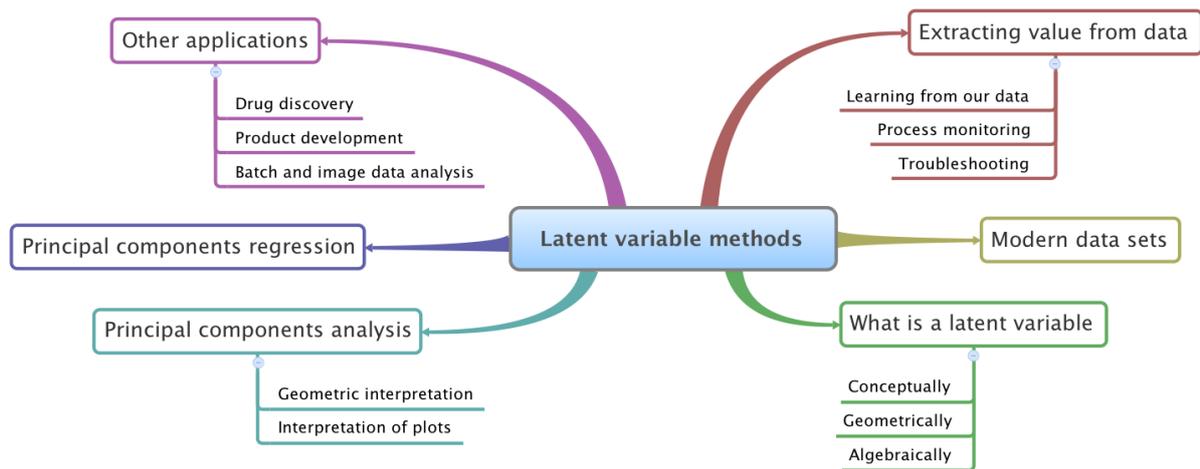
6.2 References and readings

These readings cover a variety of topics in the area of latent variable methods:

- **General**: A collection of important latent variable publications are collected at https://learnche.org/literature

- **General**: John MacGregor, Honglu Yu, Salvador García-Muñoz, Jesus Flores-Cerrillo, "Data-Based Latent Variable Methods for Process Analysis, Monitoring and Control[135]". *Computers and Chemical Engineering*, **29**, 1217-1223, 2005.

- **General**: Ericsson, Johansson, Kettaneth-Wold, Trygg, Wikström, Wold: "Multivariate and Megavariate Data Analysis".

- **About PCA**: Svante Wold, Kim Esbensen, Paul Geladi: "Principal Component Analysis[136]", *Chemometrics and Intelligent Laboratory Systems*, **2**, 37-52, 1987.

- **PLS**: Svante Wold, Michael Sjöström, Lennart Eriksson: "PLS-regression: A Basic Tool of Chemometrics[137]", *Chemometrics and Intelligent Laboratory Systems*, **58**, 109-130, 2001.

- **PLS**: S. Wold, S. Hellberg, T. Lundstedt, M. Sjöström and H. Wold, "PLS Modeling With Latent Variables in Two or More Dimensions", Frankfurt PLS meeting, 1987 (*available on request, by email to kgdunn@gmail.com*)

- **PLS**: Paul Geladi and Bruce Kowalski, "Partial Least-Squares Regression: A Tutorial[138]", *Analytica Chimica Acta*, **185**, 1-17, 1986.

- **PLS**: Paul Garthwaite, "An Interpretation of Partial Least Squares[139]", Journal of the American Statistical Association, **89**, 122-127, 1994.

- **Process monitoring**: John MacGregor and Theodora Kourti "Statistical Process Control of Multivariate Processes[140]", *Control Engineering Practice*, **3**, p 403-414, 1995.

- **Process monitoring**: J.V. Kresta, T.E. Marlin, and J.F. MacGregor "Multivariate Statistical Monitoring of Process Operating Performance[141]", *Canadian Journal of Chemical Engineering*, **69**, 35-47, 1991.

- **Contribution plots**: P Miller, RE Swanson, CE Heckler, "Contribution Plots: a Missing Link in Multivariate Quality Control[142]", *Applied Mathematics and Computer Science*, *8* (4), 775-792, 1998. (*hard to obtain, but available on request, by email to kgdunn@gmail.com*)

- **Soft sensors**: J.V. Kresta, T.E. Marlin, and J.F. MacGregor, "Development of Inferential Process Models Using PLS[143]". *Computers and Chemical Engineering*, **18**, 597-611, 1994.

- **Industrial applications**: Ivan Miletic, Shannon Quinn, Michael Dudzic, Vit Vaculik and Marc Champagne, "An Industrial Perspective on Implementing On-Line Applications of Multivariate Statistics[144]", *Journal of Process Control*, **14**, p. 821-836, 2004.

---

[135] https://dx.doi.org/10.1016/j.compchemeng.2005.02.007
[136] https://dx.doi.org/10.1016/0169-7439(87)80084-9
[137] https://dx.doi.org/10.1016/S0169-7439(01)00155-1
[138] https://dx.doi.org/10.1016/0003-2670(86)80028-9
[139] https://www.jstor.org/stable/2291207
[140] https://dx.doi.org/10.1016/0967-0661(95)00014-L
[141] https://dx.doi.org/10.1002/cjce.5450690105
[142] https://learnche.org/literature/item/78/contribution-plots-a-missing-link-in-multivariate-quality-control
[143] https://dx.doi.org/10.1016/0098-1354(93)E0006-U
[144] https://dx.doi.org/10.1016/j.jprocont.2004.02.001

- **Batch modelling and monitoring**: S. Wold, N. Kettaneh-Wold, J.F. MacGregor, K.G. Dunn, "Batch Process Modeling and MSPC[145]". *Comprehensive Chemometrics*, **2**, 163-197, 2009. (*available from the author on request, by email to kgdunn@gmail.com*)

- **Image analysis**: M. Bharati, and J.F. MacGregor "Multivariate Image Analysis for Real Time Process Monitoring and Control[146]", *Industrial and Engineering Chemistry Research*, **37**, 4715-4724, 1998

6.3 Extracting value from data

There are five main areas where engineers use large quantities of data.

1. **Improved process understanding**

   This is an implicit goal in any data analysis: either we confirm what we know about the process, or we see something unusual show up and learn from it. Plots that show, in one go, how a complex set of variables interact and relate to each other are required for this step.

2. **Troubleshooting process problems**

   Troubleshooting occurs after a problem has occurred. There are many potential sources that could have caused the problem. Screening tools are required that will help isolate the variables most related to the problem. These variables, combined with our engineering knowledge, are then used to troubleshoot why the problem occurred.

3. **Improving, optimizing and controlling processes**

   We have already introduced the concept of *designed experiments and response surface methods* (page 231). These are excellent tools to intentionally manipulate your process so that you can find a more optimal operating point, or even develop a new product. We will show how latent variable tools can be used on a large historical data set to improve process operation, and to move to a new operating point. There are also tools for applying process control in the latent variable space.

4. **Predictive modelling** (inferential sensors)

   The section on *least squares modelling* (page 151) provided you with a tool for making predictions. We will show some powerful examples of how a "difficult-to-measure" variable can be predicted in real-time, using other easy-to-obtain process data. Least squares modelling is a good tool, but it lacks some of the advantages that latent variable methods provide, such as the ability to handle highly collinear data, and data with missing values.

5. **Process monitoring**

   Once a process is running, we require monitoring tools to ensure that it maintains and stays at optimal performance. We have already considered *process monitoring charts* (page 109) for univariate process monitoring. In this section we extend that concept to monitoring multiple variables.

---

[145] https://dx.doi.org/10.1016/B978-044452701-1.00108-3
[146] https://dx.doi.org/10.1021/ie980334l

6.3.1 The types of data engineers deal with now

When industrial manufacturing and chemical engineering started to develop around the 1920's to 1950's, data collected from a process were, at most, just a handful of columns. These data were collected manually and often at considerable expense.

The "classical" tools required to visualize and understand these datasets are *scatter plots* (page 11), *time-series plots* (page 2), *Shewhart charts* (page 114) and *EWMA charts* (page 123) for process monitoring, and *multiple linear regression* (page 185) (MLR) least-squares models; all the tools which we have already learned about so far.

We will represent any data set as a matrix, called $\mathbf{X}$, where each row in $\mathbf{X}$ contains values taken from an *object* of some sort. These rows, or *observations* could be a collection of measurements at a particular point in time, various properties on a sample of final product, or a sample of raw material from a supplier. The columns in $\mathbf{X}$ are the values recorded for each observation. We call these the *variables* and there are K of them.
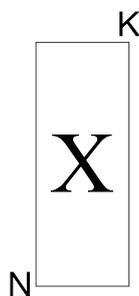
These data sets from the 1950's frequently had many more rows than columns, because it was expensive and time-consuming to measure additional columns. The choice of which columns to measure was carefully thought out, so that they didn't unnecessarily duplicate the same measurement. As a result:

- the columns of X were often independent, with little or no overlapping information

- the variables were measured in a controlled environment, with a low amount of error

These data sets meet all the assumptions required to use the so-called "classical" tools, especially least squares modelling. Data sets that engineers currently deal with though can be of any configuration with both large and small N and large and small K, but more likely we have many columns for each observation.

Small N and small K

These cases are mostly for when we have expensive measurements, and they are hard to obtain frequently. Classical methods to visualize and analyze these data always work well: scatterplots, linear regression, *etc*.

Small N and large K

This case is common for laboratory instrumentation, particularly spectroscopic devices. In recent years we are routinely collecting large quantities of data. A typical example is with near-infrared probes embedded at-line. These probes record a spectral response at around 1000 to 2000 different wavelengths. The data are represented in $\mathbf{X}$ using one wavelength per column and each sample appears in a row. The illustration here shows data from $N = 460$ samples, with data recorded every 2 nm ($K = 650$).
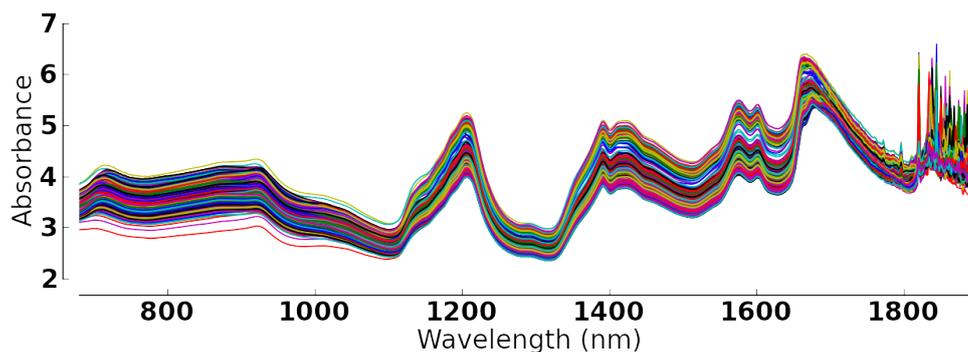
Obviously not all the columns in this matrix are important; some regions are more useful than others, and columns immediately adjacent to each other are extremely similar (non-independent).

An ordinary least squares regression model, where we would like to predict some y-variable from these spectral data, cannot be calculated when $K > N$, since we are then estimating more unknowns than we have observations for. A common strategy used to deal with non-independence is to select only a few columns (wavelengths in the spectral example) so that $K < N$. The choice of columns is subjective, so a better approach is required, such as *projection to latent structures* (page 375).

Large N and small K

A current-day chemical refinery easily records about 2 observations (rows) per second on around 2000 to 5000 variables (called tags); generating in the region of 50 to 100 Mb of data per second.

For example, a modest size distillation column would have about 35 temperature measurements, 5 to 10 flow rates, 10 or so pressure measurements, and then about 5 more measurements derived from these recorded values.
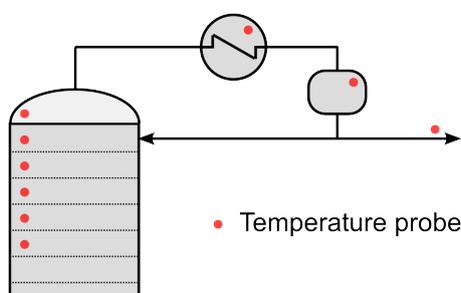
An industrial distillation example[147] is given on the data set website with $K = 27$, from a small column in Canada.

N approximately equal to K

The case of squarish matrices mostly occurs by chance: we just happen to have roughly the same number of variables as observations.

X and Y matrices

This situation arises when we would like to predict one or more variables from another group of variables. We have already seen this data structure in the *least squares section* (page 185) where $M = 1$, but more generally we would like to predict several y-values from the same data in **X**.

---

[147] http://openmv.net/info/distillation-tower
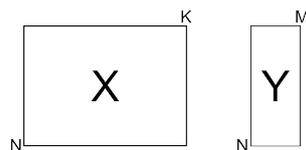
The "classical" solution to this problem is to build and maintain M different least squares models. We will see in the section on *projection to latent structures* (page 375) that we can build a single regression model. The sections on *principal component regression* (page 371) also investigates the above data structure, but for single y-variables.

3D data sets and higher dimensions

These data tables are becoming very common, especially since 2000 onwards. A typical example is for image data from digital cameras. In this illustration a single image is taken at a point in time. The camera records the response at 6 different wavelengths, and the $x - y$ spatial directions (top-to-bottom and left-to-right). These values are recorded in a 3D data cube.
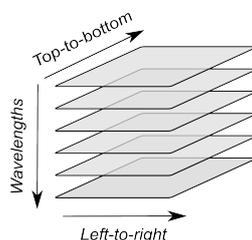
A fourth dimension can be added to this data if we start recording images over time. Such systems generate between 1 and 5 Mb of data per second. As with the spectral data set mentioned earlier, these camera systems generate large quantities of redundant data, because neighbouring pixels, both in time and spatially, are so similar. It is a case of high noise and little real information.

Batch data sets

Batch systems are common with high-value products: pharmaceuticals, fine-chemicals, and polymers. The **Z** matrix below contains data that describes how the batch is prepared and also contains data that is constant over the duration of the whole batch. The **X** matrix contains the recorded values for each variable over the duration of the batch. For example, temperature ramp-up and ramp-down, flow rates of coolant, agitator speeds and so on. The final product properties, recorded at the end of the batch, are collected in matrix **Y**.
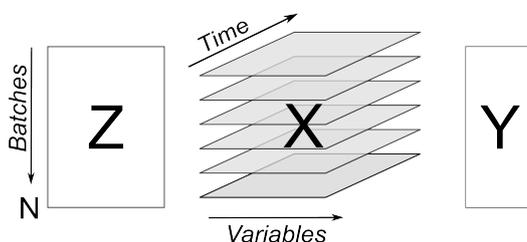
An example of batch trajectory data, in matrix **X**, where there are 4 variables, recorded at 80 times points, on about 20 batches is shown here:
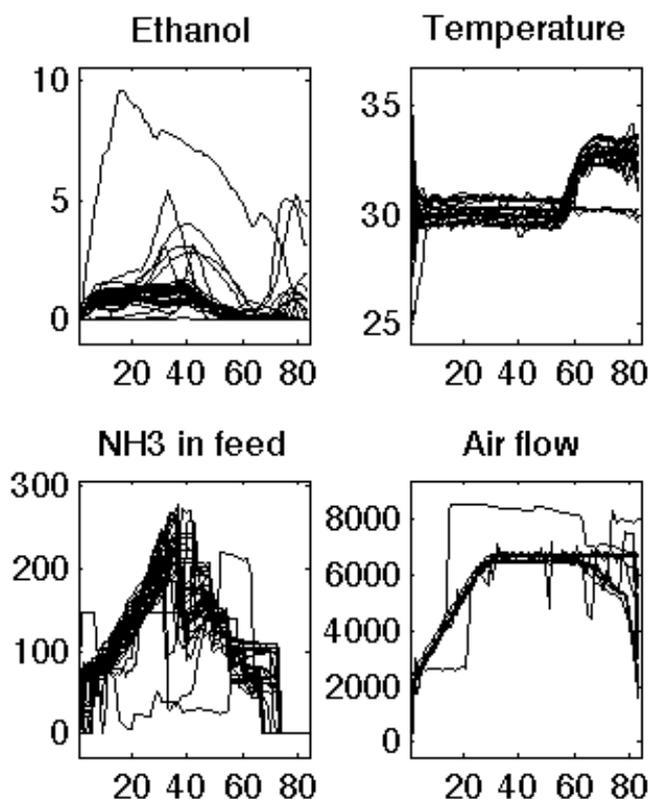
Data fusion

This is a recent buzz-word that simply means we collect and use data from multiple sources. Imagine the batch system above: we already have data in $\mathbf{Z}$ recorded by manual entry, data in $\mathbf{X}$ recorded by sensors on the process, and then $\mathbf{Y}$, typically from lab measurements. We might even have a near infrared probe in the reactor that provides a complete spectrum (a vector) at each point in time. The process of combining these data sets together is called data fusion. Each data set is often referred to as a block. We prefer to use the term multiblock data analysis when dealing with combined data sets.

6.3.2 Issues faced with engineering data

Size of the data

The most outstanding feature of the above data sets is their large size, both in terms of the number of rows and columns. This is primarily because data acquisition and data storage has become cheap.

The number of rows isn't too big of a deal: we can sub-sample the data, use parallel processors on our computers or distributed computing (a.k.a. cloud computing) to deal with this. The bigger problem is the number of columns in the data arrays. A data set with K columns can be visualized using $K(K-1)/2$ *pairs of scatterplots* (page 326); this is manageable for $K < 8$, but the quadratic number of combinations prevents us from using scatterplot matrices to visualize this data, especially when $K > 10$.

The need here is for a tool that deals with large K.

Lack of independence

The lack of independence is a big factor in modern data sets - it is problematic for example with MLR where the $\mathbf{X'X}$ becomes singular as the data become more dependent. Sometimes we can make our data more independent by selecting a reduced number of columns, but this requires good knowledge of the system being investigated, is time-consuming, and we risk omitting important variables.

Low signal to noise ratio

Engineering systems are usually kept as stable as possible: the ideal being a flat line. Data from such systems have very little signal and high noise. Even though we might record 50 Mb per second from various sensors, computer systems can, and actually do, "throw away" much of the data. This is not advisable from a multivariate data analysis perspective, but the reasoning behind it is hard to fault: much of the data we collect is not very informative. A lot of it is just from constant operation, noise, slow drift or error.

Finding the interesting signals in these routine data (also known as happenstance data), is a challenge.

Non-causal data

This happenstance data is also non-causal. The opposite case is when one runs a designed experiment; this intentionally adds variability into a process, allowing us to conclude cause-and-effect relationships, if we properly block and randomize.

But happenstance data just allows us to draw inference based on correlation effects. Since correlation is a prerequisite for causality, we can often learn a good deal from the correlation patterns in the data. Then we use our engineering knowledge to validate any correlations, and we can go on to truly verify causality with a randomized designed experiment, if it is an important effect to verify.

Errors in the data

Tools, such as least squares analysis, assume the recorded data has no error. But most engineering systems have error in their measurements, some of it quite large, since much of the data is collected by automated systems under non-ideal conditions.

So we require tools that relax the assumption that measurements have no error.

Missing data

Missing data are very common in engineering applications. Sensors go off-line, are damaged, or it is simply not possible to record all the variables (attributes) on each observation. Classical approaches are to throw away rows or columns with incomplete information, which might be acceptable when we have large quantities of data, but could lead to omitting important information in many cases.

In conclusion, we require methods that:

- are able to rapidly extract the relevant information from a large quantity of data

- deal with missing data

- deal with 3-D and higher dimensional data sets

- be able to combine data on the same object, that is stored in different data tables

- handle collinearity in the data (low signal to noise ratio)

- assume measurement error in all the recorded data.

Latent variable methods are a suitable tool that meet these requirements.

6.4 What is a latent variable?

We will take a look at what a latent variable is conceptually, geometrically, and mathematically.

6.4.1 Your health

Your overall health is a latent variable. But there isn't a single measurement of "*health*" that can be measured - it is a rather abstract concept. Instead we measure physical properties from our bodies, such as blood pressure, cholesterol level, weight, various distances (waist, hips, chest), blood sugar, temperature, and a variety of other measurements. These separate measurements can be used by a trained person to judge your health, based on their experience of seeing these values from a variety of healthy and unhealthy patients.

In this example, your *health* is a latent, or hidden variable. If we had a sensor for health, we could measure and use that variable, but since we don't, we use other measurements which all contribute in some way to assessing health.

6.4.2 Room temperature

Conceptually

Imagine the room you are in has 4 temperature probes that sample and record the local temperature every 30 minutes. Here is an example of what the four measurements might look like over 3 days.
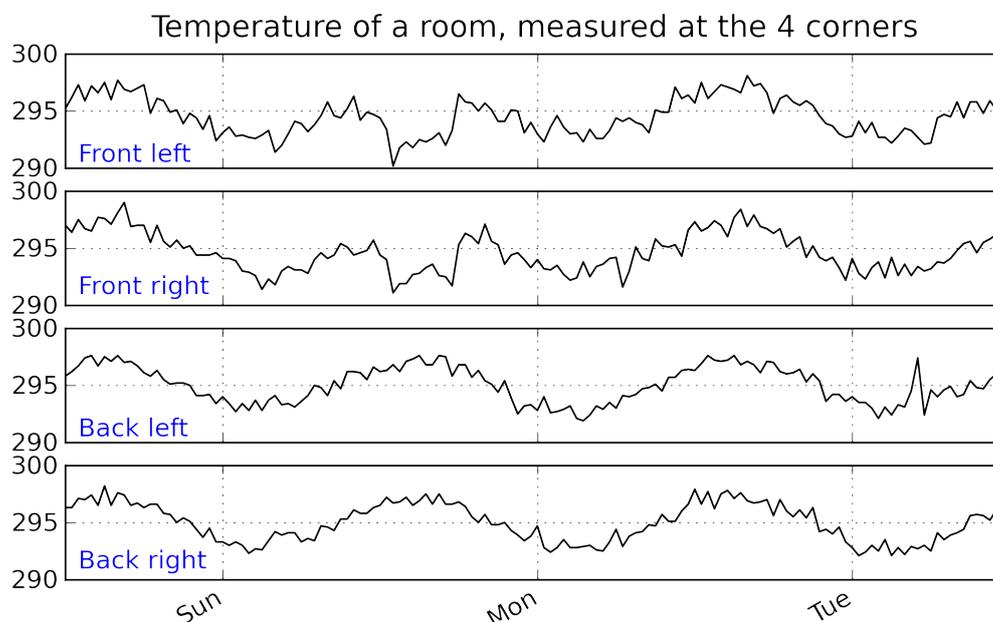
In table form, the first few measurements are:

Date	x_1	x_2	x_3	x_4
Friday 11:00	295.2	297.0	295.8	296.3
Friday 11:30	296.2	296.4	296.2	296.3
Friday 12:00	297.3	297.5	296.7	297.1
Friday 12:30	295.9	296.7	297.4	297.0
Friday 13:00	297.2	296.5	297.6	297.4
Friday 13:30	296.6	297.7	296.7	296.5

The general up and down fluctuations are due to the daily change in the room's temperature. The single, physical phenomenon being recorded in these four measurements is just the variation in room temperature.

If we added two more thermometers in the middle of the room, we would expect these new measurements to show the same pattern as the other four. In that regard we can add as many thermometers as we like to the room, but we won't be recording some new, independent piece of information with each thermometer. There is only one true variable that drives all the temperature readings up and down: it is a latent variable.

Notice that we don't necessarily have to know what *causes* the latent variable to move up and down (it could be the amount of sunlight on the building; it could be the air-conditioner's settings). All we know is that these temperature measurements just reflect the underlying phenomenon that drives the up-and-down movements in temperature; they are *correlated* with the latent variable.

Notice also the sharp spike recorded at the back-left corner of the room could be due to an error in the temperature sensor. And the front part of the room showed a dip, maybe because the door was left open for an extended period; but not long enough to affect the other temperature readings. These two events go against the general trend of the data, so we expect these periods of time to *stand out* in some way, so that we can detect them.

Mathematically

If we wanted to summarize the events taking place in the room we might just use the average of the recorded temperatures. Let's call this new, average variable t_1, which summarizes the other four original temperature measurements x_1, x_2, x_3 and x_4.

$$t_1 = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} \begin{bmatrix} p_{1,1} \\ p_{2,1} \\ p_{3,1} \\ p_{4,1} \end{bmatrix} = x_1 p_{1,1} + x_2 p_{2,1} + x_3 p_{3,1} + x_4 p_{4,1}$$

and suitable values for each of the weights are $p_{1,1} = p_{2,1} = p_{3,1} = p_{4,1} = 1/4$.

Mathematically the correct way to say this is that t_1 is a *linear combination* of the raw measurements $(x_1, x_2, x_3$ and $x_4)$ given by the weights $(p_{1,1}, p_{2,1}, p_{3,1}, p_{4,1})$.

Geometrically

We can visualize the data from this system in several ways, but we will simply show a 3-D representation of the first 3 temperatures: x_1, x_2, x_3.
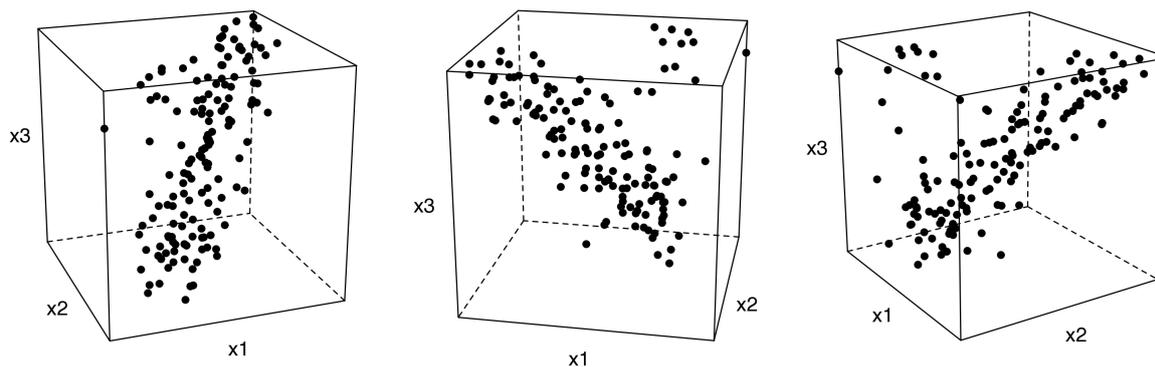
The 3 plots show the same set of data, just from different points of view. Each observation is a single dot, the location of which is determined by the recorded values of temperature, x_1, x_2 and x_3. We will use this representation in the next section again.

Note how correlated the data appear: forming a diagonal line across the cube's interior, with a few outliers (described above) that don't obey this trend.

The main points from this section are:

• Latent variables capture, in some way, an underlying phenomenon in the system being investigated.

• After calculating the latent variables in a system, we can use these fewer number of variables, instead of the K columns of raw data. This is because the actual measurements are *correlated* with the latent variable.

The examples given so far showed what a single latent variables is. In practice we usually obtain several latent variables for a data array. At this stage you likely have more questions, such as "*how many latent variables are there in a matrix*" and "*how are the values in* **P** *chosen*", and "*how do we know these latent variables are a good summary of the original data*"?

We address these issues more formally in the next section on *principal component analysis* (page 325).

6.5 Principal Component Analysis (PCA)

Principal component analysis, PCA, builds a model for a matrix of data.

A model is always an approximation of the system from where the data came. The objectives for which we use that model *can be varied* (page 317).

In this section we will start by visualizing the data as well as consider a simplified, geometric view of what a PCA model looks like. A mathematical analysis of PCA is also required to get a deeper understanding of PCA, so we go into some detail on that point, however it can be skipped on first reading.

The first part of this section emphasizes the general interpretation of a PCA model, since this is a required step that any modeller will have to perform. We leave to the *second half of this section* (page 351) the important details of how to preprocess the raw data, how to actually calculate the PCA model, and how to validate and test it. This "reverse" order may be unsatisfying for some, but it is helpful to see how to use the model first, before going into details on its calculation.

6.5.1 Visualizing multivariate data

The data, collected in a matrix $\mathbf{X}$, contains rows that represent an *object* of some sort. We usually call each row an *observation*. The observations in $\mathbf{X}$ could be a collection of measurements from a chemical process at a particular point in time, various properties of a final product, or properties from a sample of raw material. The columns in $\mathbf{X}$ are the values recorded for each observation. We call these the *variables*.

Which variables should you use, and how many observations do you require? We address this issue later. For now though we consider that you have your data organized in this manner:
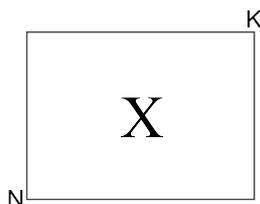
Consider the case of 2 variables, $K = 2$ (left) and $K = 3$ variables (right) for the room thermometers example *from earlier* (page 323):
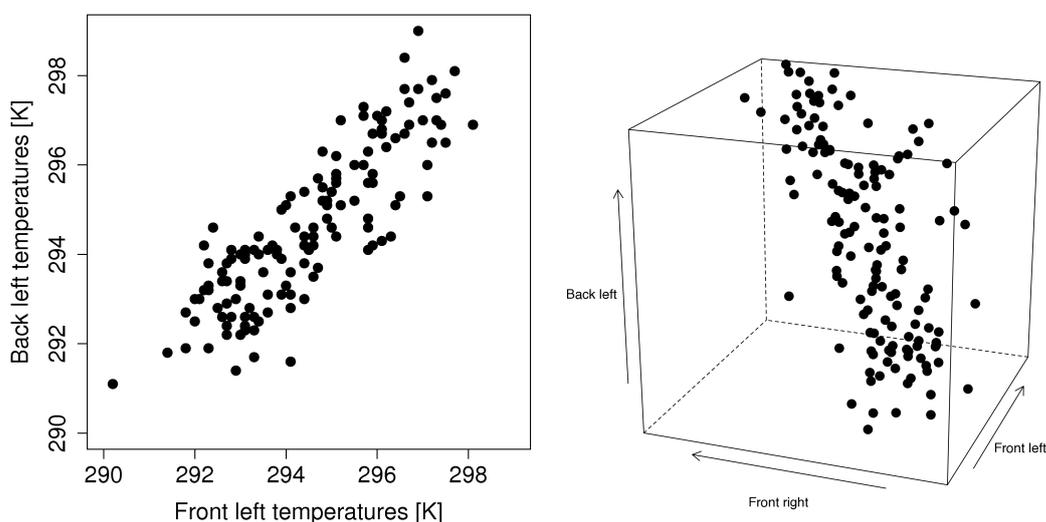
Each point in the plot represents one *object*, also called an *observation*. There are about 150 observations in each plot here. We sometimes call these plots *data swarms*, but they are really just ordinary scatterplots that we saw in the *visualization section* (page 1). Notice how the variables are correlated with each other, there is a definite trend. If we want to explain this trend, we could draw a line through the cloud swarm that *best explains* the data. This line now represents our best summary and estimate of what the data points are describing. If we wanted to describe that relationship to our colleagues we could just give them the equation of the best-fit line.

Another effective way to visualize small multivariate data sets is to use a scatterplot matrix. Below is an example for $K = 5$ measurements on $N = 50$ observations. Scatterplot matrices require $K(K-1)/2$ plots and can be enhanced with univariate histograms (on the diagonal plots), and linear regressions and loess smoothers on the off-diagonals to indicate the level of correlation between any two variables.
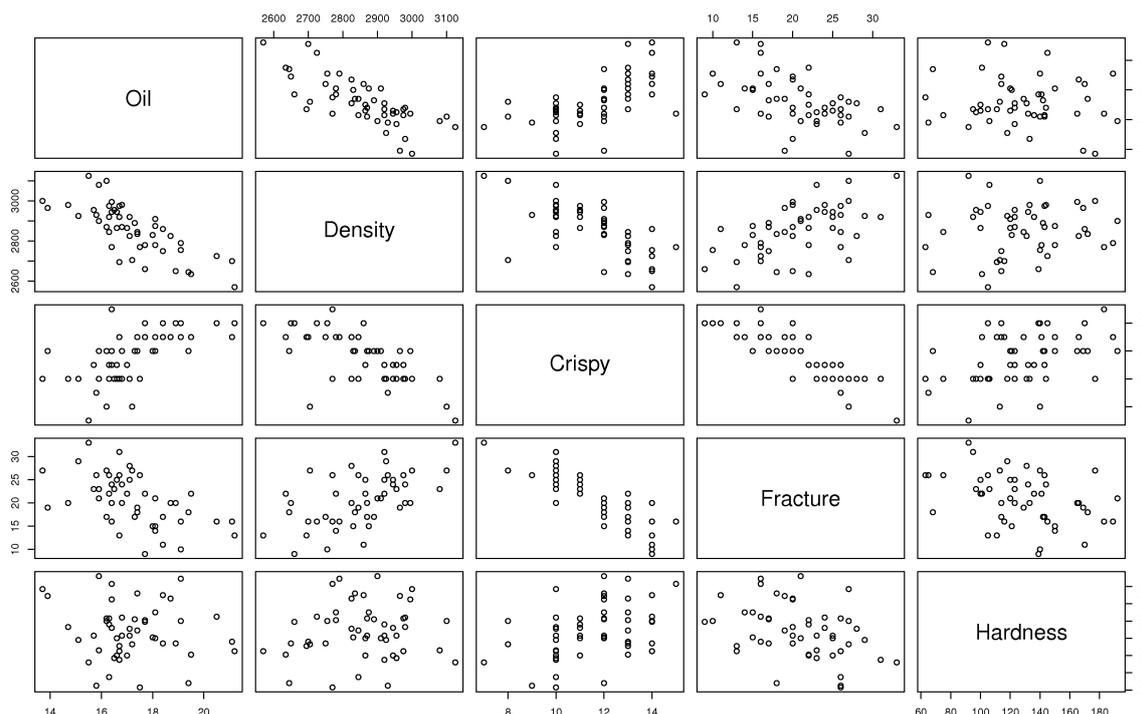
6.5.2 Geometric explanation of PCA

We refer to a K-dimensional space when referring to the data in $\mathbf{X}$. We will start by looking at the geometric interpretation of PCA when $\mathbf{X}$ has 3 columns, in other words a 3-dimensional space, using measurements: $[x_1, x_2, x_3]$.
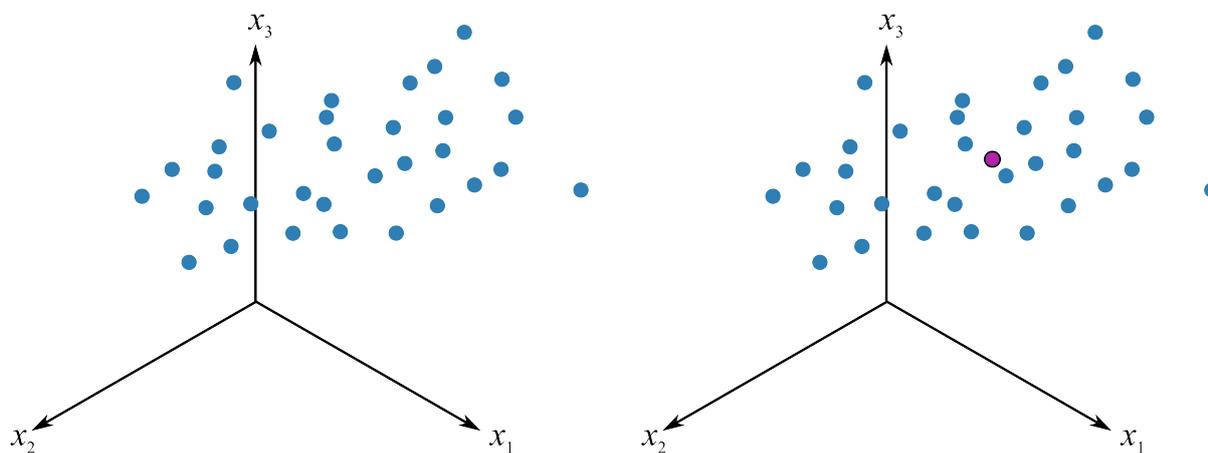
The raw data in the cloud swarm show how the 3 variables move together. The first step in PCA is to move the data to the center of the coordinate system. This is called mean-centering and removes the arbitrary bias from measurements that we don't wish to model. We also scale the data, usually to unit-variance. This removes the fact that the variables are in different units of measurement. Additional discussion on centering and scaling is *in the section on data preprocessing* (page 351).

After centering and scaling we have moved our raw data to the center of the coordinate system and each variable has equal scaling.

The best-fit line is drawn through the swarm of points. The more correlated the original data, the

better this line will explain the actual values of the observed measurements. This best-fit line will *best explain* all the observations with minimum residual error. Another, but equivalent, way of expressing this is that the line goes in the direction of *maximum variance of the projections onto the line*. Let's take a look at what that phrase means.
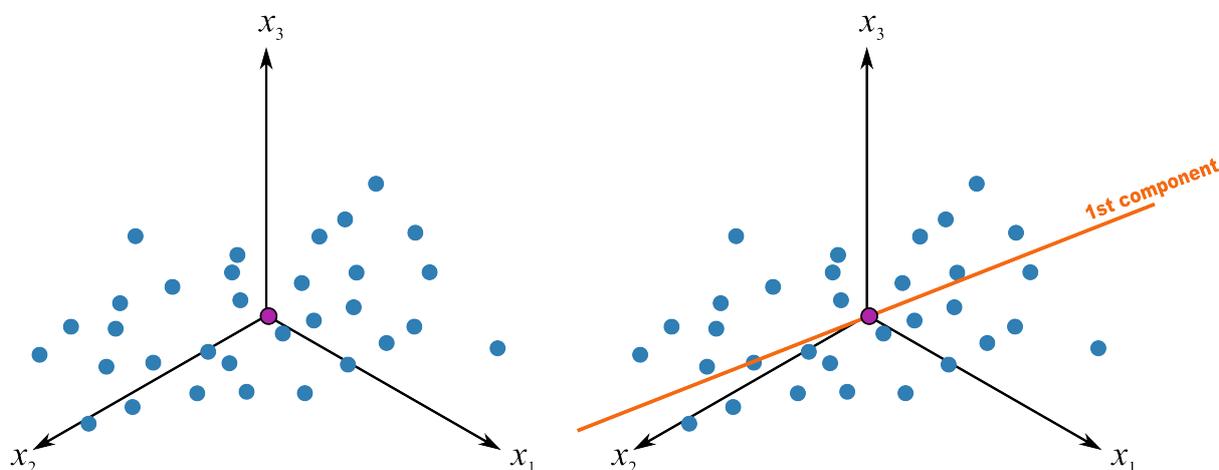
When the direction of the best-fit line is found we can mark the location of each observation along the line. We find the 90 degree projection of each observation onto the line (see the next illustration). The distance from the origin to this projected point along the line is called the *score*. Each observation gets its own score value. When we say the best-fit line is in the direction of maximum variance, what we are saying is that the variance of these scores will be maximal. (There is one score for each observation, so there are N score values; the variance of these N values is at a maximum). Notice that some score values will be positive and others negative.

After we have added this best-fit line to the data, we have calculated the first principal component, also called the first latent variable. Each principal component consists of two parts:

- The direction vector that defines the best-fit line. This is a K-dimensional vector that tells us which direction that best-fit line points, in the K-dimensional coordinate system. We call this direction vector $\mathbf{p}_1$, it is a $K \times 1$ vector. This vector starts at the origin and moves along the best-fit line. Since vectors have both magnitude and direction, we chose to rescale this vector so that it has magnitude of exactly 1, making it a unit-vector.

- The collection of N score values along this line. We call this our score vector, $\mathbf{t}_1$, and it is an $N \times 1$ vector.

- The subscript of "1" emphasizes that this is the first latent variable.
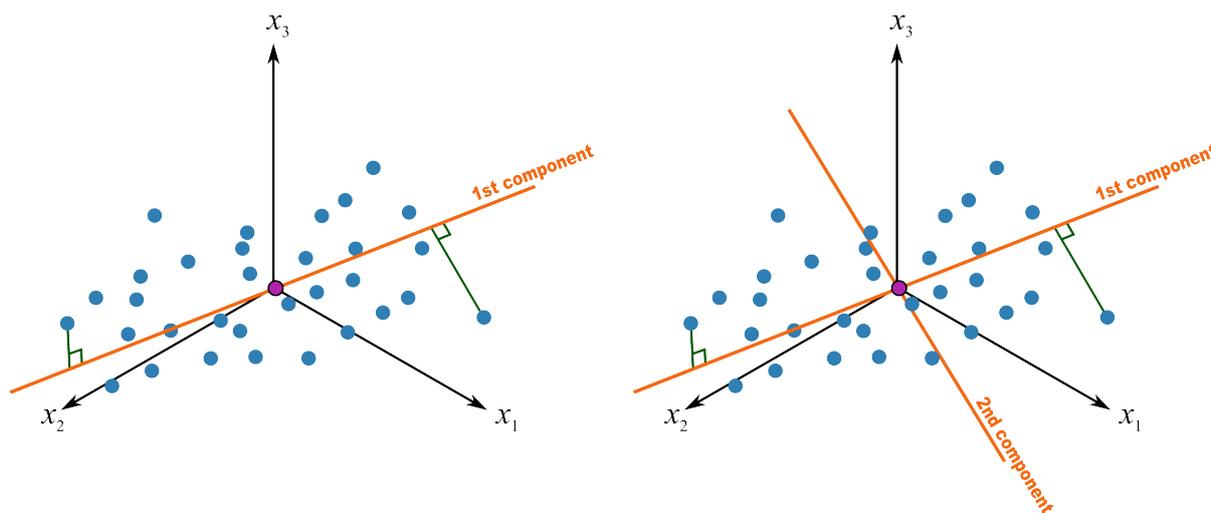
This first principal component is fixed and we now add a second component to the system. We find the second component so that it is perpendicular to the first component's direction. Notice that this vector also starts at the origin, and can point in any direction as long as it remains perpendicular to the first component. We keep rotating the second component's direction vector around until we find a direction that gives the greatest variance in the score values when projected on this new direction vector.
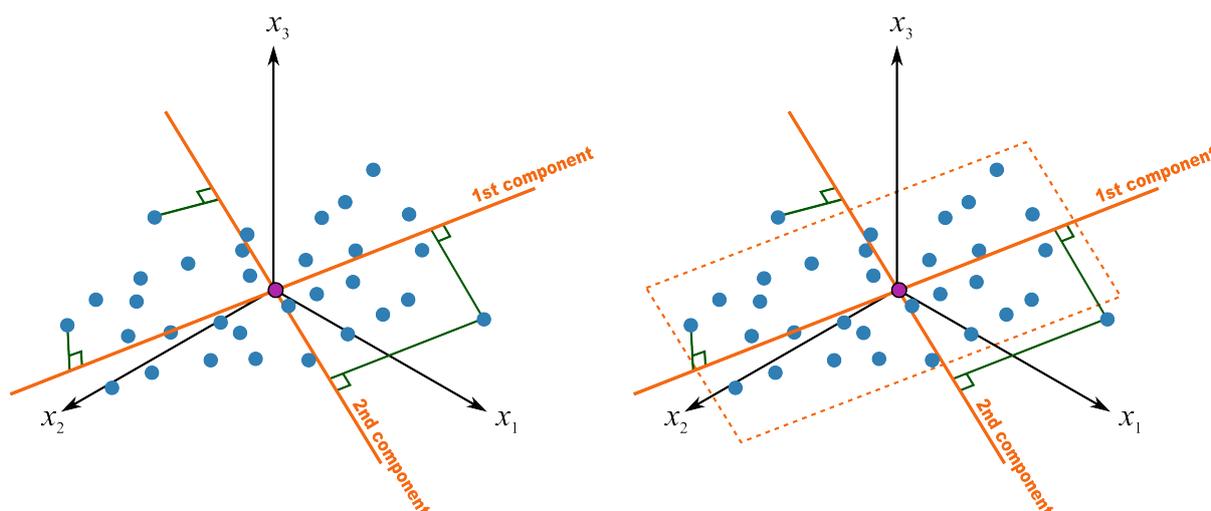
What that means is that once we have settled on a direction for the second component, we calculate the scores values by perpendicularly projecting each observation towards this second direction vector. The score values for the second component are the locations along this line. As before, there will be some positive and some negative score values. This completes our second component:

- This second direction vector, called $\mathbf{p}_2$, is also a $K \times 1$ vector. It is a unit vector that points in the direction of next-greatest variation.

- The scores (distances), collected in the vector called $\mathbf{t}_2$, are found by taking a perpendicular projection from each observation onto the $\mathbf{p}_2$ vector.

Notice that the $\mathbf{p}_1$ and $\mathbf{p}_2$ vectors jointly define a plane. This plane is the *latent variable model* with two components. With one component the latent variable model is just a line, with two components, the model is a plane, and with 3 or more components, the model is defined by a hyperplane. We will use the letter a to identify the number of components. The PCA model is said to have A components, or A latent variables, where $a = 1, 2, 3, \ldots A$.

This hyperplane is really just the best approximation we can make of the original data. The perpendicular distance from each point onto the plane is called the *residual distance* or *residual error*. So what a principal component model does is break down our raw data into two parts:

1. a latent variable model (given by vectors $\mathbf{p}$ and $\mathbf{t}$), and

2. a residual error.

A principal component model is one type of latent variable model. A PCA model is computed in such a way that the latent variables are oriented in the *direction that gives greatest variance* of the scores. There are other latent variable models, but they are computed with different objectives.

6.5.3 Mathematical derivation for PCA

Geometrically, when finding the *best-fit line* for the swarm of points, our objective was to minimize the error, i.e. the residual distances from each point to the best-fit line is the smallest possible. This is also mathematically equivalent to maximizing the variance of the scores, $\mathbf{t}_a$.

We briefly review here what that means. Let $\mathbf{x}_i'$ be a row from our data, so $\mathbf{x}_i'$ is a $1 \times K$ vector. We defined the score value for this observation as the distance from the origin, along the direction vector, $\mathbf{p}_1$, to the point where we find the perpendicular projection onto $\mathbf{p}_1$. This is illustrated below, where the score value for observation $\mathbf{x}_i$ has a value of $t_{i,1}$.
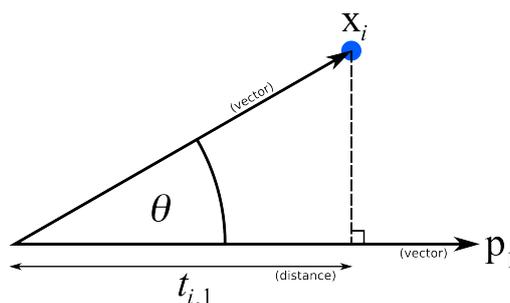
Recall from geometry that the cosine of an angle in a right-angled triangle is the ratio of the adjacent side to the hypotenuse. But the cosine of an angle is also used in linear algebra to define the dot-product. Mathematically:

$$\cos \theta = \frac{\text{adjacent length}}{\text{hypotenuse}} = \frac{t_{i,1}}{\|\mathbf{x}_i\|} \qquad \text{and also} \qquad \cos \theta = \frac{\mathbf{x}_i'\mathbf{p}_1}{\|\mathbf{x}_i\|\|\mathbf{p}_1\|}$$

$$\frac{t_{i,1}}{\|\mathbf{x}_i\|} = \frac{\mathbf{x}_i'\mathbf{p}_1}{\|\mathbf{x}_i\|\|\mathbf{p}_1\|}$$

$$t_{i,1} = \mathbf{x}_i'\mathbf{p}_1$$

$$(1 \times 1) = (1 \times K)(K \times 1)$$

where $\|\cdot\|$ indicates the length of the enclosed vector, and the length of the direction vector, $\mathbf{p}_1$ is 1.0, by definition.

Note that $t_{i,1} = \mathbf{x}_i'\mathbf{p}_1$ represents a *linear combination* (page 324)

$$t_{i,1} = x_{i,1}p_{1,1} + x_{i,2}p_{2,1} + \ldots + x_{i,k}p_{k,1} + \ldots + x_{i,K}p_{K,1}$$

So $t_{i,1}$ is the score value for the ith observation along the first component, and is a linear combination of the ith row of data, $\mathbf{x}_i$ and the direction vector $\mathbf{p}_1$. Notice that there are K terms in the linear combination: each of the K variables *contributes* to the overall score.

We can calculate the second score value for the ith observation in a similar way:

$$t_{i,2} = x_{i,1}p_{1,2} + x_{i,2}p_{2,2} + \ldots + x_{i,k}p_{k,2} + \ldots + x_{i,K}p_{K,2}$$

And so on, for the third and subsequent components. We can compactly write in matrix form for the ith observation that:

$$\mathbf{t}'_i = \mathbf{x}'_i \mathbf{P}$$
$$(1 \times A) = (1 \times K)(K \times A)$$

which calculates all A score values for that observation in one go. This is exactly what we *derived earlier* (page 324) in the example with the 4 thermometers in the room.

Finally, for an entire matrix of data, $\mathbf{X}$, we can calculate all scores, for all observations:

$$\mathbf{T} = \mathbf{XP}$$
$$(N \times A) = (N \times K)(K \times A)$$

(6.1)

6.5.4 More about the direction vectors (loadings)

The direction vectors $\mathbf{p}_1$, $\mathbf{p}_2$ and so on, are each $K \times 1$ unit vectors. These are vectors in the original coordinate space (the K-dimensional real-world) where the observations are recorded.

But these direction vectors are also our link to the latent-variable coordinate system. These direction vectors create a (hyper)plane that is embedded inside the K-dimensional space of the K original variables. You will see the terminology of *loadings* - this is just another name for these direction vectors:

$$\text{Loadings, a } K \times A \text{ matrix:} \qquad \mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \ldots & \mathbf{p}_A \end{bmatrix}$$

Once this hyperplane is mapped out, then we start to consider how each of the observations lie on this hyperplane. We start to be more and more interested in this reduced dimensional plane, because it is an A-dimensional plane, where A is often much smaller than K. Returning back to the case of the thermometers in a room: we had 4 thermometers ($K = 4$), but only one latent variable, $A = 1$. Rather than concern ourself with the original 4 measurements, we only focus on the single column of score values, since this single variables is the best summary possible of the 4 original variables.

How do we get the score value(s)? We use the *equation from the prior section* (page 331) (repeated here). It is the multiplication of the pre-processed data by the loadings vectors:

$$\mathbf{T} = \mathbf{XP}$$
$$(N \times A) = (N \times K)(K \times A)$$

and it shows how the loadings are our link from the K-dimensional, real-world, coordinate system to the A-dimensional, latent variable-world, coordinates.

Let's return to the *example of the 4 temperatures* (page 323). We derived there that a plausible summary of the 4 temperatures could be found from:

$$t_1 = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} \begin{bmatrix} p_{1,1} \\ p_{2,1} \\ p_{3,1} \\ p_{4,1} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix} = \mathbf{x}_i \mathbf{p}_1$$

So the loading vector for this example points in the direction $\mathbf{p}'_1 = [0.25, 0.25, 0.25, 0.25]$. This isn't a unit vector though; but we can make it one:

- Current magnitude of vector = $\sqrt{0.25^2 + 0.25^2 + 0.25^2 + 0.25^2} = 0.50$

- Divide the vector by current magnitude: $\mathbf{p}_1 = \dfrac{1}{0.5} \cdot [0.25, 0.25, 0.25, 0.25]$

- New, unit vector = $\mathbf{p}_1 = [0.5, 0.5, 0.5, 0.5]$

- Check new magnitude = $\sqrt{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2} = 1.0$

What would be the entries in the $\mathbf{p}_1$ loading vector if we had 6 thermometers? (*Ans* = 0.41; in general, for K thermometers, $1/\sqrt{K}$).

This is very useful, because now instead of dealing with K thermometers we can reduce the columns of data down to just a single, average temperature. This isn't a particularly interesting case though; you would have likely done this anyway as an engineer facing this problem. But the next *food texture example* (page 332) will illustrate a more realistic case.

6.5.5 PCA example: Food texture analysis

Let's take a look at an example to consolidate and extend the ideas introduced so far. This data set is from a food manufacturer[148] making a pastry product. Each sample (row) in the data set is taken from a batch of product where 5 quality attributes are measured:

1. Percentage oil in the pastry

2. The product's density (the higher the number, the more dense the product)

3. A crispiness measurement, on a scale from 7 to 15, with 15 being more crispy.

4. The product's fracturability: the angle, in degrees, through which the pasty can be slowly bent before it fractures.

5. Hardness: a sharp point is used to measure the amount of force required before breakage occurs.

A scatter plot matrix of these $K = 5$ measurements is shown for the $N = 50$ observations.

---

[148] http://openmv.net/info/food-texture
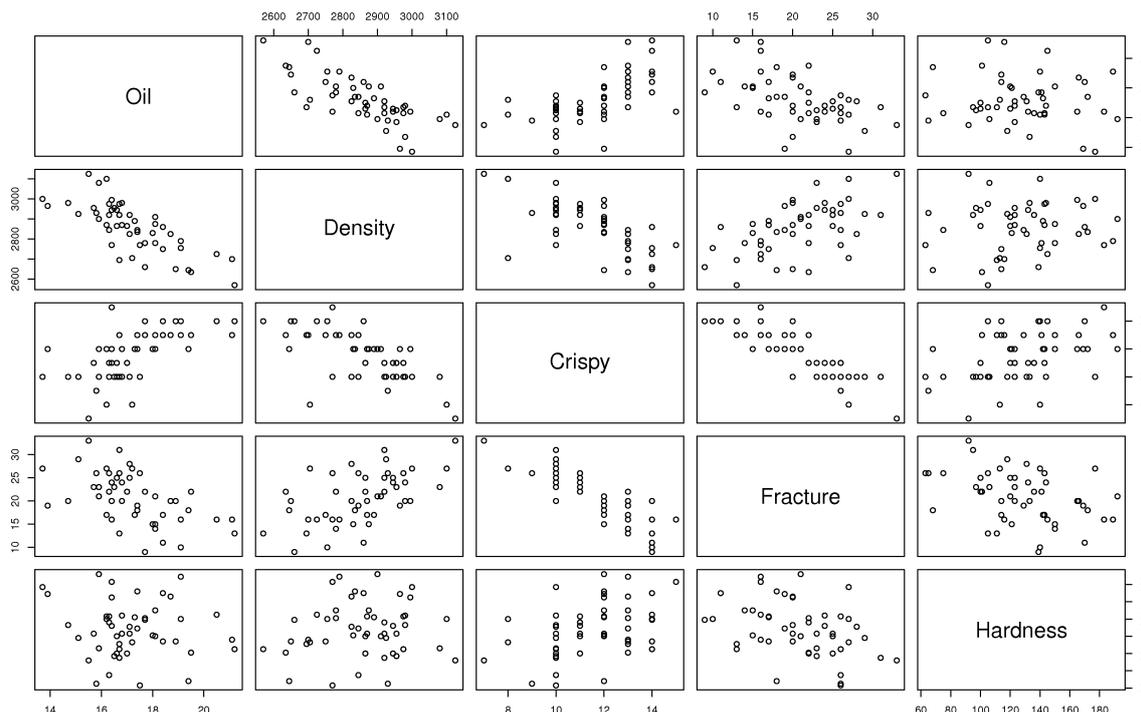
We can get by with this visualization of the data because K is small in this case. This is also a good starting example, because you can refer back to these scatterplots to confirm your findings.

Preprocessing the data

The first step with PCA is to center and scale the data. The box plots show how the raw data are located at different levels and have arbitrary units.
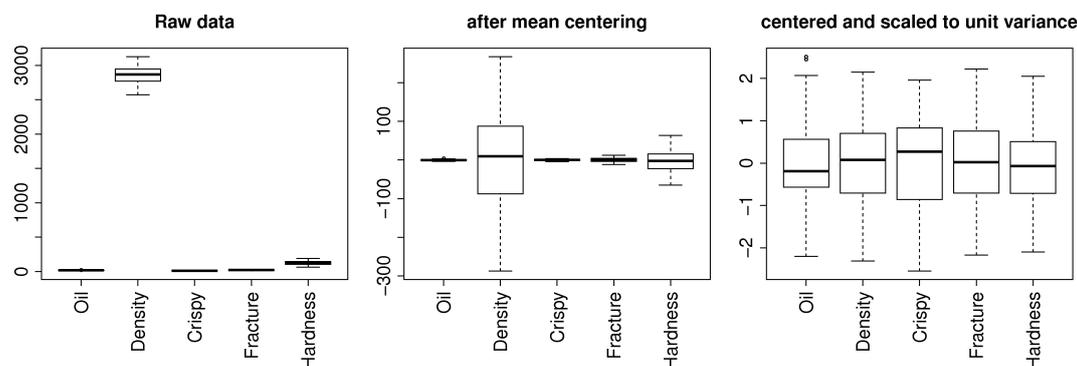
Centering removes any bias terms from the data by subtracting the mean value from each column in the matrix $\mathbf{X}$. For the kth column:

$$\mathbf{x}_{k,\text{center}} = \mathbf{x}_{k,\text{raw}} - \text{mean}\left(\mathbf{x}_{k,\text{raw}}\right)$$

Scaling removes the fact that the raw data could be in diverse units:

$$\mathbf{x}_k = \frac{\mathbf{x}_{k,\text{center}}}{\text{standard deviation}\left(\mathbf{x}_{k,\text{center}}\right)}$$

Then each column $\mathbf{x}_k$ is collected back to form matrix $\mathbf{X}$. This preprocessing is so common it is called autoscaling: center each column to zero mean and then scale it to have unit variance. After this preprocessing each column will have a mean of 0.0 and a variance of 1.0. (Note the box plots don't

quite show this final result, because they use the median instead of the mean, and show the interquartile range instead of the standard deviation).

Centering and scaling does not alter the overall interpretation of the data: if two variables were strongly correlated before preprocessing they will still be strongly correlated after preprocessing.

For reference, the mean and standard deviation of each variable is recorded below. In the last 3 columns we show the raw data for observation 33, the raw data after centering, and the raw data after centering and scaling:

Variable	Mean	Standard deviation	Raw data	After centering	After autoscaling
Oil	17.2	1.59	15.5	-1.702	-1.069
Density	2857.6	124.5	3125	267.4	+2.148
Crispy	11.52	1.78	7	-4.52	-2.546
Fracture	20.86	5.47	33	12.14	+2.221
Hardness	128.18	31.13	92	-36.18	-1.162

Loadings: $\mathbf{p}_1$

We will discuss how to determine the number of components to use *in a future section* (page 360), and *how to compute them* (page 353), but for now we accept there are two important components, $\mathbf{p}_1$ and $\mathbf{p}_2$. They are:

$$\mathbf{p}_1 = \begin{bmatrix} +0.46 \\ -0.47 \\ +0.53 \\ -0.50 \\ +0.15 \end{bmatrix} \quad \text{and} \quad \mathbf{p}_2 = \begin{bmatrix} -0.37 \\ +0.36 \\ +0.20 \\ -0.22 \\ +0.80 \end{bmatrix}$$

Where we might visualize that first component by a bar plot:
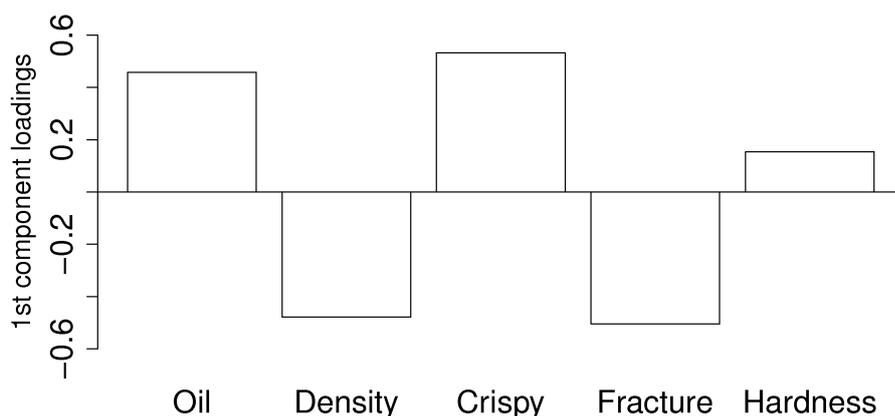
This plot shows the first component. All variables, except for hardness have large values in $\mathbf{p}_1$. If we write out the equation for t_1 for an observation i:

$$t_{i,1} = 0.46\ x_\text{oil} - 0.47\ x_\text{density} + 0.53\ x_\text{crispy} - 0.50\ x_\text{fracture} + 0.15\ x_\text{hardness} \tag{6.2}$$

Once we have centered and scaled the data, remember that a negative x-value is a value below the average, and that a positive x-value lies above the average.

For a pastry product to have a high t_1 value would require it to have some combination of above-average oil level, low density, and/or be more crispy and/or only have a small angle by which it

can be bent before it fractures, i.e. low fracturability. So pastry observations with high t_1 values sound like they are brittle, flaky and light. Conversely, a product with low t_1 value would have the opposite sort of conditions: it would be a heavier, more chewy pastry (higher fracture angle) and less crispy.

Scores: $\mathbf{t}_1$

Let's examine the score values calculated. As shown in equation (6.2), the score value is a linear combination of the data, $\mathbf{x}$, given by the weights in the loadings matrix, $\mathbf{P}$. For the first component, $\mathbf{t}_1 = \mathbf{X}\mathbf{p}_1$. The plot here shows the values in vector $\mathbf{t}_1$ (an $N \times 1$ vector) as a sequence plot
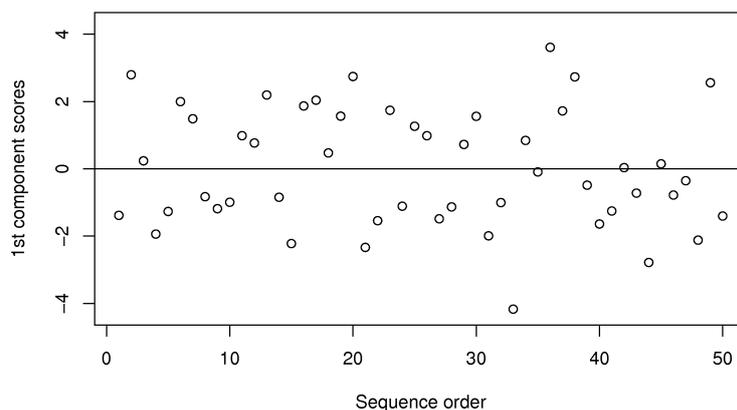
The samples appear to be evenly spread, some high and some low on the t_1 scale. Sample 33 has a t_1 value of -4.2, indicating it was much denser than the other pastries, and had a high fracture angle (it could be bent more than others). In fact, if we refer to the raw data[149] we can confirm these findings: $\mathbf{x}_{i=33} = [15.5, \ 3125, \ 7, \ 33, \ 92]$. Also refer back to the scatterplot matrix and mark the point which has density of 3125, and fracture angle of 33. This pastry also has a low oil percentage (15.5%) and low crispy value (7).

We can also investigate sample 36, with a t_1 value of 3.6. The raw data again confirm that this pastry follows the trends of other, high t_1 value pastries. It has a high oil level, low density, high crispiness, and a low fracture angle: $x_{36} = [21.2, \ 2570, \ 14, \ 13, \ 105]$. Locate again on the scatterplot matrices sample 36 where oil level is 21.2 and the crispiness is 14. Also mark the point where density = 2570 and the fracture value = 13 for this sample.

We note here that this component explains 61% of the original variability in the data. It's hard to say whether this is high or low, because we are unsure of the degree of error in the raw data, but the point is that a single variable summarizes about 60% of the variability from all 5 columns of raw data.

Loadings: $\mathbf{p}_2$

The second loading vector is shown as a bar plot:

---

[149] http://openmv.net/info/food-texture
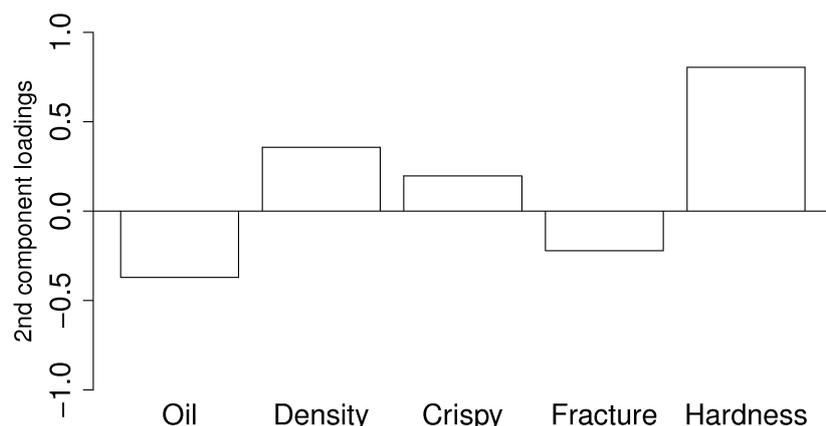
This direction is aligned mainly with the hardness variable: all other variables have a small coefficient in $\mathbf{p}_2$. A high t_2 value is straightforward to interpret: it would imply the pastry has a high value on the hardness scale. Also, this component explains an additional 26% of the variability in the dataset.

Because this component is orthogonal to the first component, we can be sure that this hardness variation is independent of the first component. One valuable way to interpret and use this information is that you can adjust the variables in $\mathbf{p}_2$, i.e. the process conditions that affect the pastry's hardness, without affecting the other pastry properties, i.e the variables described in $\mathbf{p}_1$.

6.5.6 Interpreting score plots

Before summarizing some points about how to interpret a score plot, let's quickly repeat what a score value is. There is one score value for each observation (row) in the data set, so there are are N score values for the first component, another N for the second component, and so on.

The score value for an observation, for say the first component, is the distance from the origin, along the direction (loading vector) of the first component, up to the point where that observation projects onto the direction vector. We repeat *an earlier figure here* (page 327), which shows the projected values for 2 of the observations.
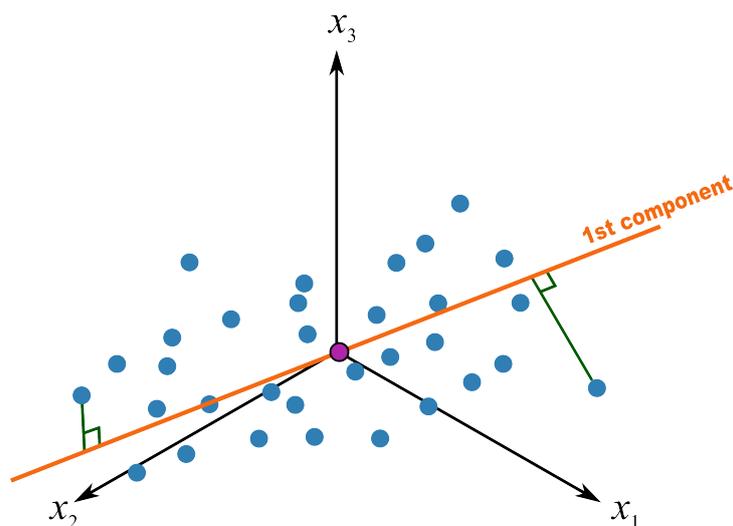
We used *geometric concepts in another section* (page 330) that showed we can write: $\mathbf{T} = \mathbf{XP}$ to get all the scores value in one go. In this section we are plotting values from the columns of $\mathbf{T}$. In particular,

for a single observation, for the ath component:

$$t_{i,a} = x_{i,1} \, p_{1,a} + x_{i,2} \, p_{2,a} + \ldots + x_{i,k} \, p_{k,a} + \ldots + x_{i,K} \, p_{K,a}$$

The first score vector, $\mathbf{t}_1$, explains the greatest variation in the data; it is considered the most important score from that point of view, at least when we look at a data set for the first time. (After that we may find other scores that are more interesting). Then we look at the second score, which explains the next greatest amount of variation in the data, then the third score, and so on. Most often we will plot:

- time-series plots of the scores, or sequence order plots, depending on how the rows of $\mathbf{X}$ are ordered

- scatter plots of one score against another score

An important point with PCA is that because the matrix $\mathbf{P}$ is orthonormal (see the *later section on PCA properties* (page 362)), any relationships that were present in $\mathbf{X}$ are still present in $\mathbf{T}$. We can see this quite easily using the previous equation. Imagine two observations taken from a process at different points in time. It would be quite hard to identify those similar points by looking at the K columns of raw data, especially when the two rows are not close to each other. But with PCA, these two similar rows are multiplied by the same coefficients in $\mathbf{P}$ and will therefore give similar values of t. So score plots allow us to rapidly locate similar observations.

When investigating score plots we look for *clustering*, *outliers*, time-based *patterns*. We can also colour-code our plots to be more informative. Let's take a look at each of these.

Clustering

We usually start by looking at the $(\mathbf{t}_1, \mathbf{t}_2)$ scatterplot of the scores, the two directions of greatest variation in the data. As just previously explained, observations in the rows of $\mathbf{X}$ that are similar will fall close to each other, i.e. they cluster together, in these score plots. Here is an example of a score plot, calculated from data from a fluidized catalytic cracking (FCC) process [Taken from the Masters thesis of Carol Slama (McMaster University, p 78, 1991)].

Fig. 5.2a
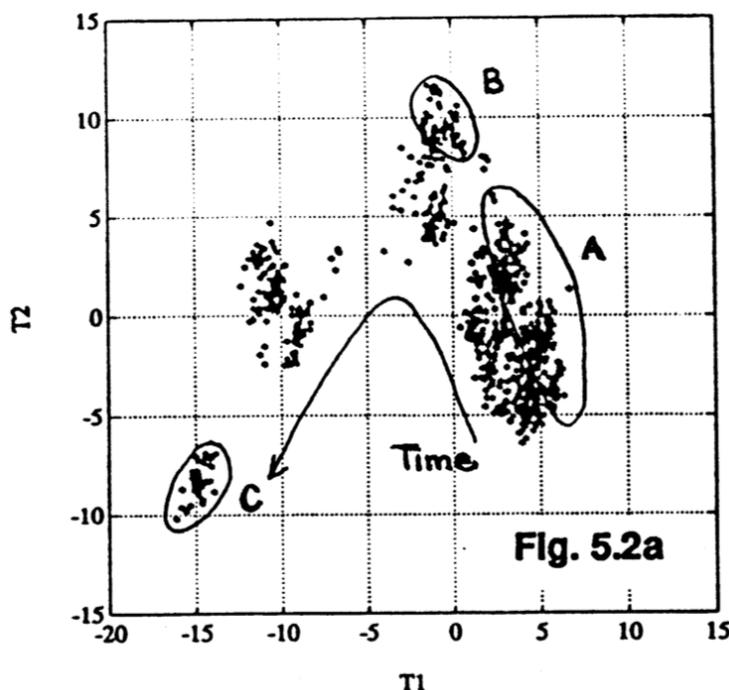
It shows how the process was operating in region A, then moved to region B and finally region C. This provides a 2-dimensional window into the movements from the $K = 147$ original variables.

Outliers

Outliers are readily detected in a score plot, and using the equation below we can see why. Recall that the data in $\mathbf{X}$ have been centered and scaled, so the x-value for a variable that is operating at the mean level will be roughtly zero. An observation that is at the mean value for all K variables will have a score vector of $\mathbf{t}_i = [0, 0, \dots, 0]$. An observation where many of the variables have values far from their average level is called a multivariate outlier. It will have one or more score values that are far from zero, and will show up on the outer edges of the score scatterplots.

Sometimes all it takes is for one variable, $x_{i,k}$ to be far away from its average to cause $t_{i,a}$ to be large:

$$t_{i,a} = x_{i,1} \ p_{1,a} + x_{i,2} \ p_{2,a} + \dots + x_{i,k} \ p_{k,a} + \dots + x_{i,K} \ p_{K,a}$$

But usually it is a combination of more than one x-variable. There are K terms in this equation, each of which *contribute* to the score value. A bar plot of each of these K terms, $x_{i,k} \ p_{k,a}$, is called a contribution plot. It shows which variable(s) most contribute to the large score value.

As an example from the *food texture data* (page 332) from earlier, we saw that observation 33 had a large negative $\mathbf{t}_1$ value. From *that prior equation* (page 334):

$$t_{33,1} = 0.46 \ x_{\text{oil}} - 0.47 \ x_{\text{density}} + 0.53 \ x_{\text{crispy}} - 0.50 \ x_{\text{fracture}} + 0.15 \ x_{\text{hardness}}$$

$$t_{33,1} = 0.46 \times -1.069 - 0.47 \times +2.148 + 0.53 \times -2.546 - 0.50 \times 2.221 + 0.15 \times -1.162$$

$$t_{33,1} = -4.2$$

The $K = 5$ terms that contribute to this value are illustrated as a bar plot, where the sum of the bar heights add up to -4.2:
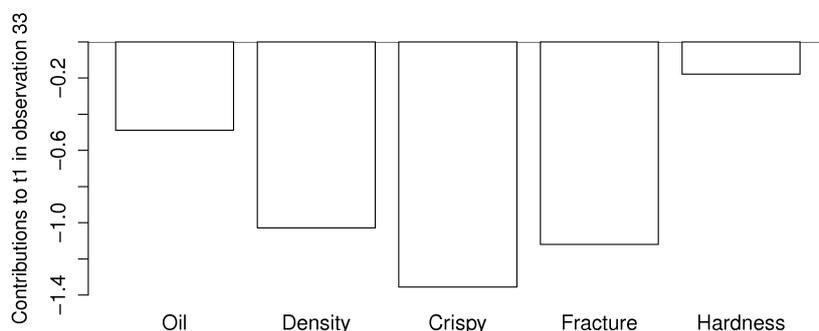
This gives a more accurate indication of exactly how the low t_i value was achieved. Previously we had said that pastry 33 was denser than the other pastries, and had a higher fracture angle; now we can see the relative contributions from each variable more clearly.

In the figure from the FCC process (in the *preceding subsection on clustering* (page 337)), the cluster marked C was far from the origin, relative to the other observations. This indicates problematic process behaviour around that time. Normal process operation is expected to be in the center of the score plot. These outlying observations can be investigated as to why they are unusual by constructing contribution bar plots for a few of the points in cluster C.

Time-based or sequence-based trends

Any strong and consistent time-based or sequence-order trends in the raw data will be reflected in the scores also. Visual observation of each score vector may show interesting phenomena such as oscillations, spikes or other patterns of interest. As just described, contribution plots can be used to see which of the original variables in $\mathbf{X}$ are most related with these phenomena.

Colour-coding

Plotting any two score variables on a scatter plot provides good insight into the relationship between those independent variables. Additional information can be provided by *colour-coding the points on the plot* (page 13) by some other, 3rd variable of interest. For example, a binary colour scheme could denote success of failure of each observation.

A continuous 3rd variable can be implied using a varying colour scheme, going from reds to oranges to yellows to greens and then blue, together with an accompanying legend. For example profitability of operation at that point, or some other process variable. A 4th dimension could be inferred by plotting smaller or larger points. We saw an example of these *high-density visualizations* (page 13) earlier.

Summary

- Points close the average appear at the origin of the score plot.

- Scores further out are either outliers or naturally extreme observations.

- We can infer, *in general*, why a point is at the outer edge of a score plot by cross-referencing with the loadings. This is because the scores are a linear combination of the data in $\mathbf{X}$ as given by the coefficients in $\mathbf{P}$.

- We can *determine exactly why* a point is at the outer edge of a score plot by constructing a contribution plot to see which of the original variables in $\mathbf{X}$ are most related with a particular score. This provides a more precise indication of exactly why a score is at its given position.

- Original observations in $\mathbf{X}$ that are similar to each other will be similar in the score plot, while observations much further apart are dissimilar. This comes from the way the scores are computed: they are found so that span the greatest variance possible. But it is much easier to detect this similarity in an A-dimensional space than the original K-dimensional space.

6.5.7 Interpreting loading plots

Recall that the loadings plot is a plot of the direction vectors that define the model. Returning back to a previous illustration:
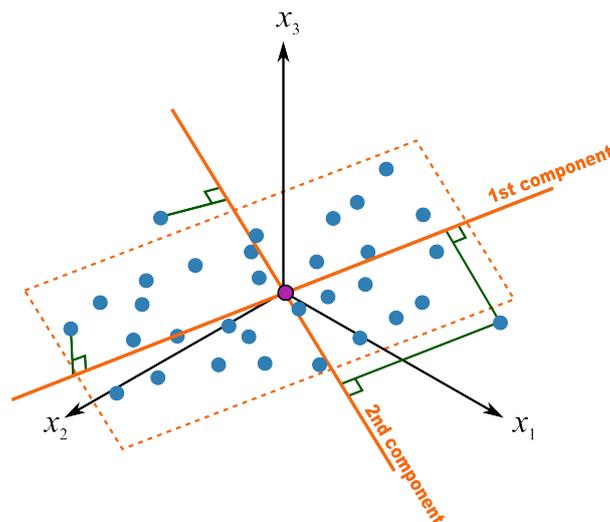
In this system the first component, $\mathbf{p}_1$, is oriented primarily in the x_2 direction, with smaller amounts in the other directions. A loadings plot would show a large coefficient (negative or positive) for the x_2 variable and smaller coefficients for the others. Imagine this were the only component in the model, i.e. it is a one-component model. We would then correctly conclude the other variables measured have little importance or relevance in understanding the total variability in the system. Say these 3 variables

represented the quality of our product, and we had been getting complaints about the variability of it. This model indicates we should focus on whatever aspect causes in variance in x_2, rather than other variables.

Let's consider another visual example where two variables, x_1 and x_2, are the predominant directions in which the observations vary; the x_3 variable is only "noise". Further, let the relationship between x_1 and x_2 have a negative correlation.
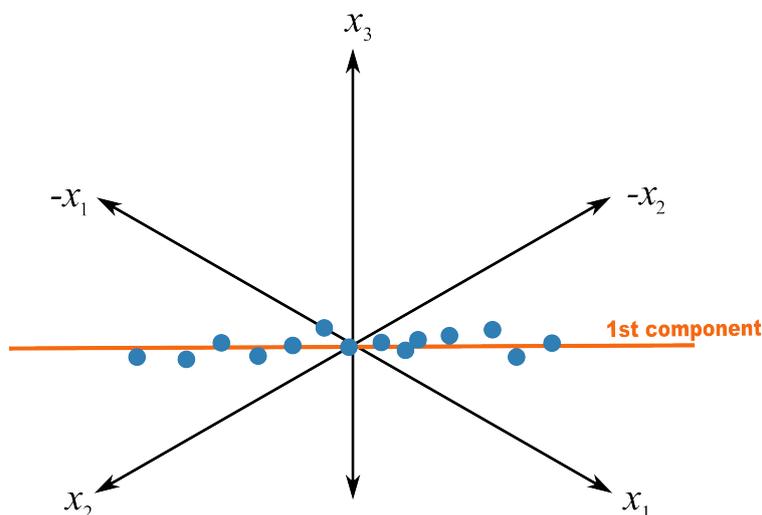
A model of such a system would have a loading vector with roughly equal weight in the $+x_1$ direction as it has in the $-x_2$ direction. The direction could be represented as $p_1 = [+1, -1, 0]$, or rescaled as a unit vector: $p_1 = [+0.707, -0.707, 0]$. An equivalent representation, with exactly the same interpretation, could be $p_1 = [-0.707, +0.707, 0]$.

This illustrates two points:

- Variables which have little contribution to a direction have almost zero weight in that loading.

- Strongly correlated variables, will have approximately the same weight value when they are positively correlated. In a loadings plot of p_i vs p_j they will appear near each other, while negatively correlated variables will appear diagonally opposite each other.

- Signs of the loading variables are useful to compare within a direction vector; but these vectors can be rotated by 180° and still have the same interpretation.

This is why they are called loadings: they show how the original variables load, (contribute), to creating the component.

Another issue to consider is the case when one has many highly correlated variables. Consider the *room temperature example* (page 323) where the four temperatures are highly correlated with each other. The first component from the PCA model is shown here:
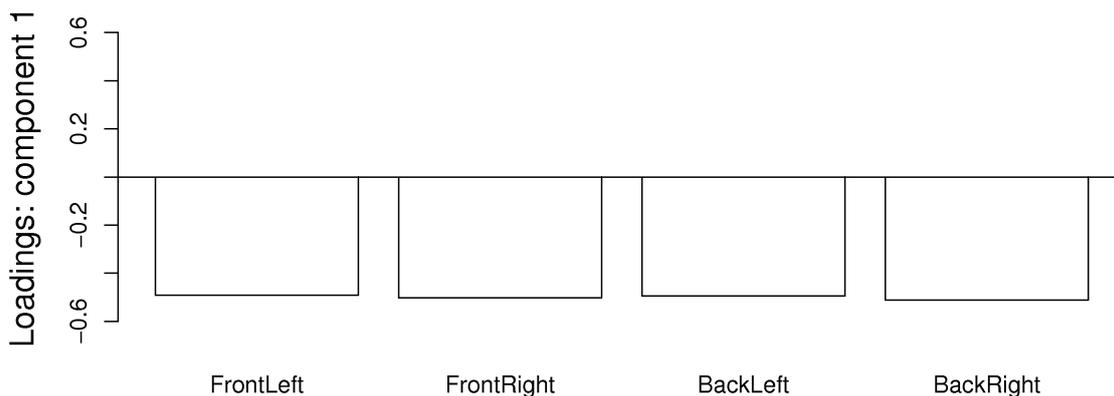
Notice how the model spreads the weights out evenly over all the correlated variables. Each variable is individually important. The model could well have assigned a weight of 1.0 to one of the variables and 0.0 to the others. This is a common feature in latent variable models: variables which have roughly equal influence on defining a direction are correlated with each other and will have roughly equal numeric weights.

Finally, one way to locate unimportant variables in the model is by finding which variables have small weights in all components. These variables can generally be removed, as they show no correlation to any of the components or with other variables.

6.5.8 Interpreting loadings and scores together

It is helpful to visualize any two score vectors, e.g. $\mathbf{t}_1$ *vs* $\mathbf{t}_2$, in a scatterplot: the N points in the scatterplot are the projection of the raw data onto the model plane described by the two loadings vectors, $\mathbf{p}_1$ and $\mathbf{p}_2$.

Any two loadings can also be shown in a scatterplot and interpreted by recalling that each loading direction is orthogonal and independent of the other direction.
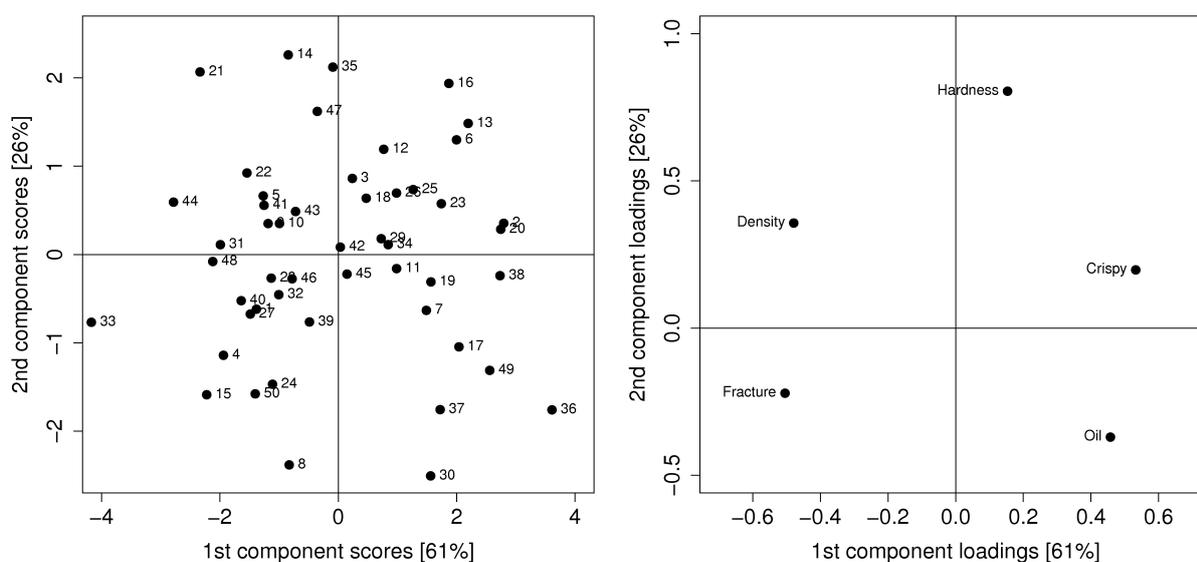
Side-by-side, these 2 plots very helpfully characterize all the observations in the data set. Recall observation 33 had a large, negative t_1 value. It had an above average fracture angle, an above average

density, a below average crispiness value of 7, and below average oil level of 15.5.

It is no coincidence that we can mentally superimpose these two plots and come to exactly the same conclusions, using only the plots. This result comes from the fact that the scores (left) are just a linear combination of the raw data, with weighting given by the loadings (right).

Use these two plots to characterize what values the 5 measurements would have been for these observations:

- sample 8:

- sample 20:

- sample 35:

- sample 42:

6.5.9 Predicted values for each observation

An interesting aspect of a PCA model is that it provides an estimate of each observation in the data set. Recall the latent variable model was oriented to create the best-fit plane to the data. This plane was oriented to minimize the errors, which implies the best estimate of each observation is its *perpendicular projection* onto the model plane.

Referring to the illustration and assume we have a PCA model with a single component, the best estimate of observation $\mathbf{x}_i$ is the point along the direction vector, $\mathbf{p}_1$, where the original observation is projected. Recall that the distance along that direction vector was $t_{i,1}$, but the actual point along $\mathbf{p}_1$ is a vector, and it is our best estimate of the original observation. We will call that estimate $\widehat{\mathbf{x}}_{i,1}$, indicating that it is an estimate of $\mathbf{x}_i$ along the first component.
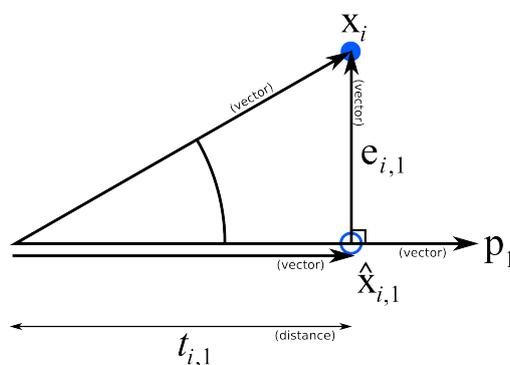
Since $\widehat{\mathbf{x}}_{i,1}$ is a vector, we can write it as the product of a magnitude value and a direction vector. The magnitude of $\widehat{\mathbf{x}}_i$ is t_i in the direction of $\mathbf{p}_1$, which is a unit vector, then mathematically we can write:

$$\widehat{\mathbf{x}}'_{i,1} = t_{i,1}\ \mathbf{p}'_1$$
$$(1 \times K) = (1 \times 1)(1 \times K)$$

This is the best prediction of the original observation using one component. If we added a second component to our model, then our estimate improves:

$$\widehat{\mathbf{x}}'_{i,2} = t_{i,1}\ \mathbf{p}'_1 + t_{i,2}\ \mathbf{p}'_2$$
$$(1 \times K) = (1 \times K) + (1 \times K)$$

With multiple components, we write:

$$\widehat{\mathbf{x}}'_{i,A} = \begin{bmatrix} t_{i,1} & t_{i,2}, & \ldots, & t_{i,A} \end{bmatrix} \mathbf{P}'$$

$$\widehat{\mathbf{x}}'_{i,A} = \mathbf{t}'_i \, \mathbf{P}'$$

$$(1 \times K) = (1 \times A)(A \times K)$$

This is interesting: $\widehat{\mathbf{x}}_{i,A}$ is a prediction of every variable in the ith observation. We only require the score values for that i^{th} observation in order to get this prediction. We multiply the scores $\mathbf{t}_i$ by the direction vectors in matrix $\mathbf{P}$ to get the prediction.

The preceding equation can be written in a way that handles the entire matrix $\mathbf{X}$:

$$\widehat{\mathbf{X}} = \mathbf{TP}'$$

$$(N \times K) = (N \times A)(A \times K) \tag{6.3}$$

Once we have the predicted value for an observation, we are also interested in the residual vector between the actual and predicted observation:

$$\mathbf{e}'_{i,A} = \mathbf{x}'_i - \widehat{\mathbf{x}}'_{i,A}$$

$$(1 \times K) = (1 \times K) - (1 \times K)$$

The residual *length* or *distance* is the sum of squares of this residual, then we take the square root to form a distance. Technically the *squared prediction error* (SPE) is just the sum of squares for each observation, but often we refer to the square root of this quantity as the SPE as well. Some software packages will scale the root of the SPE by some value; you will see this referred to as the DModX, distance to the model plane for $\mathbf{X}$.

$$\text{SPE}_i = \sqrt{\mathbf{e}'_{i,A}\mathbf{e}_{i,A}}$$

$$(1 \times 1) = (1 \times K)(K \times 1)$$

where $\mathbf{e}_{i,A}$ is the residual vector of the ith observation using A components.

6.5.10 Interpreting the residuals

We consider three types of residuals: residuals within each row of $\mathbf{X}$, called squared prediction errors (SPE); residuals for each column of $\mathbf{X}$, called R_k^2 for each column, and finally residuals for the entire matrix $\mathbf{X}$, usually just called R^2 for the model.

Residuals for each observation: the square prediction error

We have already introduced the *squared prediction error geometrically* (page 342). We showed in that section that the residual distance from the actual observation to the model plane is given by:

$$\mathbf{e}'_{i,A} = \mathbf{x}'_i - \widehat{\mathbf{x}}'_{i,A}$$

$$\mathbf{e}'_{i,A} = \mathbf{x}'_i - \mathbf{t}'_i\mathbf{P}'$$

Turning this last equation around we have:

$$\mathbf{x}'_i = \mathbf{t}'_i\mathbf{P}' + \mathbf{e}'_{i,A}$$

$$(1 \times K) = (1 \times A)(A \times K) + (1 \times K)$$

Or in general, for the whole data set

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E} = \widehat{\mathbf{X}} + \mathbf{E}$$

$$(N \times K) = (N \times A)(A \times K) + (N \times K)$$

This shows that each observation (row in $\mathbf{X}$) can be split and interpreted in two portions: a vector on-the-plane, $\mathbf{t}_i'\mathbf{P}'$, and a vector perpendicular to the plane, $\mathbf{e}_{i,A}'$. This residual portion, a vector, can be reduced to a single number, a distance value called SPE, as *previously described* (page 342).
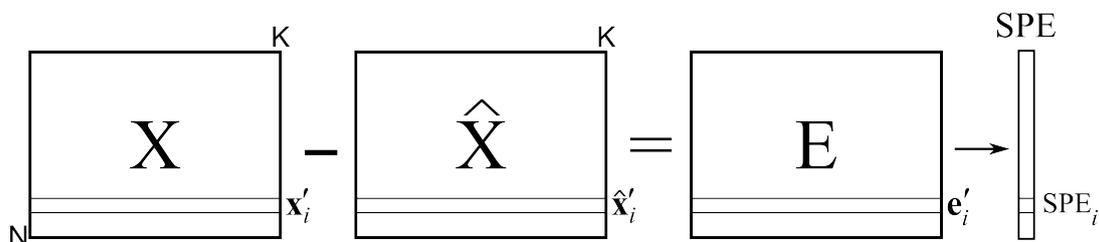
An observation in $\mathbf{X}$ that has $\text{SPE}_i = 0$ is exactly on the plane and follows the model structure exactly; this is the smallest SPE value possible. For a given data set we have a distribution of SPE values. We can calculate a confidence limit below which we expect to find a certain fraction of the data, e.g. a 95% confidence limit. We won't go into how this limit is derived, suffice to say that most software packages will compute it and show it.

The most convenient way to visualize these SPE values is as a sequence plot, or a line plot, where the y-axis has a lower limit of 0.0, and the 95% and/or 99% SPE limit is also shown. Remember that we would expect 5 out of 100 points to naturally fall above the 95% limit.

If we find an observation that has a large squared prediction error, i.e. the observation is far off the model plane, then we say this observation is *inconsistent with the model*. For example, if you have data from a chemical process, taken over several days, your first 300 observations show SPE values below the limit. Then on the 4th day you notice a persistent trend upwards in SPE values: this indicates that those observations are inconsistent with the model, indicating a problem with the process, as reflected in the data captured during that time.

We would like to know why, specifically which variable(s) in $\mathbf{X}$, are most related with this deviation off the model plane. As we did in the section on *interpreting scores* (page 336), we can generate a contribution plot.

$$\mathbf{e}_{i,A}' = \mathbf{x}_i' - \widehat{\mathbf{x}}_{i,A}'$$

Dropping the A subscript for convenience we can write the $1 \times K$ vector as:

$$\mathbf{e}_i' = \mathbf{x}_i' - \widehat{\mathbf{x}}_i'$$

$$(1 \times K) = \begin{bmatrix} (x_{i,1} - \hat{x}_{i,1}) & (x_{i,2} - \hat{x}_{i,2}) & \dots & (x_{i,k} - \hat{x}_{i,k}) & \dots & (x_{i,K} - \hat{x}_{i,K}) \end{bmatrix}$$

The SPE is just the sum of the squares of these K terms, so a residual contribution plot, most conveniently shown as a bar chart of these K terms, indicates which of the original K variable(s) are most associated with the deviation off the model plane. We say that the *correlation structure among these variables has been broken*. This is because PCA provides a model of the correlation structure in the data table. When an observation has a large residual, then that observation is said to break the correlation structure, and is inconsistent with the model.

Looking back at the *room-temperature example* (page 323): if we fit a model with one component, then the residual distance, shown with the 95% limit, appears as follows:
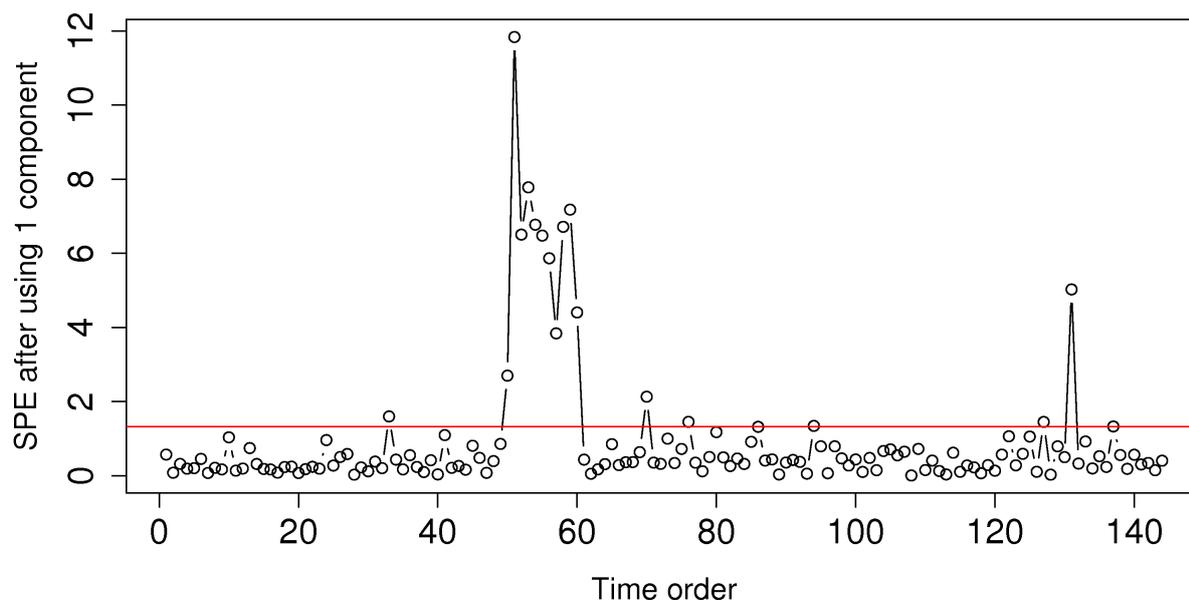
Using the raw data for this example[150], shown below, can you explain why we see those unusual points in the SPE plot around time 50 to 60?

Temperature of a room, measured at the 4 corners
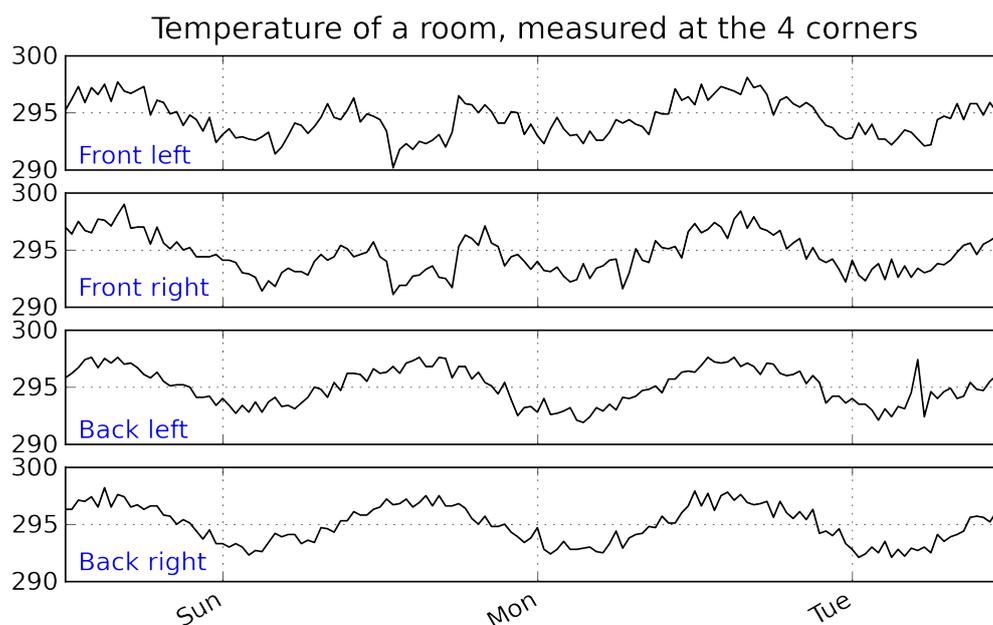
Finally, the SPE value is a complete summary of the residual vector. As such, it is sometimes used to colour-code score plots, as we mentioned back in the section on *score plots* (page 336). Another interesting way people sometimes display SPE is to plot a 3D data cloud, with $\mathbf{t}_1$ and $\mathbf{t}_2$, and use the SPE values on the third axis. This gives a fairly complete picture of the major dimensions in the model: the explained variation on-the-plane, given by $\mathbf{t}_1$ and $\mathbf{t}_2$, and the residual distance off-the-plane, summarized by SPE.

---

[150] http://openmv.net/info/room-temperature

Residuals for each column

Using the residual matrix $\mathbf{E} = \mathbf{X} - \mathbf{TP}' = \mathbf{X} - \widehat{\mathbf{X}}$, we can calculate the residuals for each column in the original matrix. This is summarized by the R^2 value for each column in $\mathbf{X}$ and gives an indication of how well the PCA model describes the data from that column.
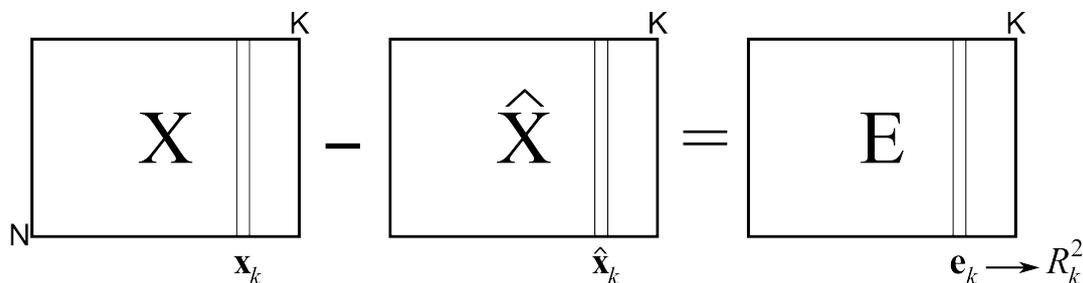
In the section on *least squares modelling* (page 151), the R^2 number was shown to be the ratio between the variance remaining in the residuals over the total variances we started off with, subtracted from 1.0. Using the notation in the previous illustration:

$$R_k^2 = 1 - \frac{\text{Var}(\mathbf{x}_k - \widehat{\mathbf{x}}_k)}{\text{Var}(\mathbf{x}_k)} = 1 - \frac{\text{Var}(\mathbf{e}_k)}{\text{Var}(\mathbf{x}_k)}$$

The R_k^2 value for each variable will increase with every component that is added to the model. The minimum value is 0.0 when there are no components (since $\widehat{\mathbf{x}}_k = \mathbf{0}$), and the maximum value is 1.0, when the maximum number of components have been added (and $\widehat{\mathbf{x}}_k = \mathbf{x}_k$, or $\mathbf{e}_k = \mathbf{0}$). This latter extreme is usually not reached, because such a model would be fitting the noise inherent in $\mathbf{x}_k$ as well.

The R^2 values for each column can be visualized as a bar plot for dissimilar variables (chemical process data), or as a line plot if there are many similar variables that have a logical left-to-right relationship, such as the case with *spectral variables* (page 347) (wavelengths).

Residuals for the whole matrix X

Finally, we can calculate an R^2 value for the entire matrix $\mathbf{X}$. This is the ratio between the variance of $\mathbf{X}$ we can explain with the model over the ratio of variance initially present in $\mathbf{X}$.

$$R^2 = 1 - \frac{\text{Var}(\mathbf{X} - \widehat{\mathbf{X}})}{\text{Var}(\mathbf{X})} = 1 - \frac{\text{Var}(\mathbf{E})}{\text{Var}(\mathbf{X})}$$

The variance of a general matrix, $\mathbf{G}$, is taken as the sum of squares of every element in $\mathbf{G}$. The example in the next section illustrates how to interpret these residuals. The smallest value of R^2 value is $R_{a=0}^2 = 0.0$ when there are no components. After the first component is added we can calculate $R_{a=1}^2$. Then after fitting a second component we get $R_{a=2}^2$. Since each component is extracting new information from $\mathbf{X}$, we know that $R_{a=0}^2 < R_{a=1}^2 < R_{a=2}^2 < \ldots < R_{a=A}^2 = 1.0$.

6.5.11 PCA example: analysis of spectral data

A data set, available on the dataset website[151], contains data on 460 tablets, measured at 650 different wavelengths.
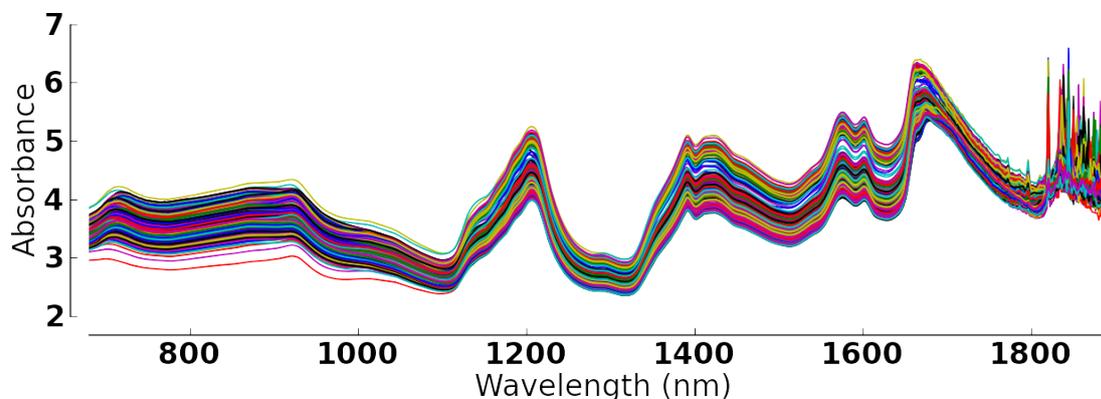
This R code will calculate principal components for this data:

```R
# Read large data file
file <- 'http://openmv.net/file/tablet-spectra.csv'
spectra <- read.csv(file, header = FALSE, row.names = 1)

# Only extract 4 components, but
# center and scale the data before
# calculation the components
model.pca <- prcomp(spectra,
                    center = TRUE,
                    scale =TRUE,
                    rank. = 4)

summary(model.pca)
```

which gives this output:

```
Importance of first k=4 (out of 460) components:
                         PC1     PC2     PC3     PC4
Standard deviation    21.8835 10.9748 3.60075 3.27081
Proportion of Variance  0.7368  0.1853 0.01995 0.01646
Cumulative Proportion   0.7368  0.9221 0.94200 0.95846
```

The R_a^2 (`Cumulative Proportion`) values shows the first component explains 73.7% of the variability in $\mathbf{X}$, the second explains an additional 18.5% for a cumulative total of 92.2%, and the third component explains an additional 1.99%. These three components together explain 94.2% of all the variation in $\mathbf{X}$. This means we have reduced $\mathbf{X}$ from a 460×650 matrix to a 460×3 matrix of scores, $\mathbf{T}$, and a 650×3 matrix of loadings, $\mathbf{P}$. This is a large reduction in data size, with a minimal loss of information.

Let's visually show what the R^2 values are for each column. Shown below are these values for the first 3 components. The first component (green, thin line) explains certain regions of the spectra very well, particularly the region around 1100nm. Wavelengths beyond 1800 nm are not well explained at all. The second component is primarily responsible for explaining additional variability in the 700 to 1100nm region. The third component only seems to explain the additional variability from 1700 to 1800nm. Fitting a fourth component is only going to start fitting the noisy regions of the spectrum on the very right. For these data we could use 2 components for most applications, or perhaps 3 if the region between 1700 and 1800nm was also important.

---

[151] http://openmv.net/info/tablet-spectra
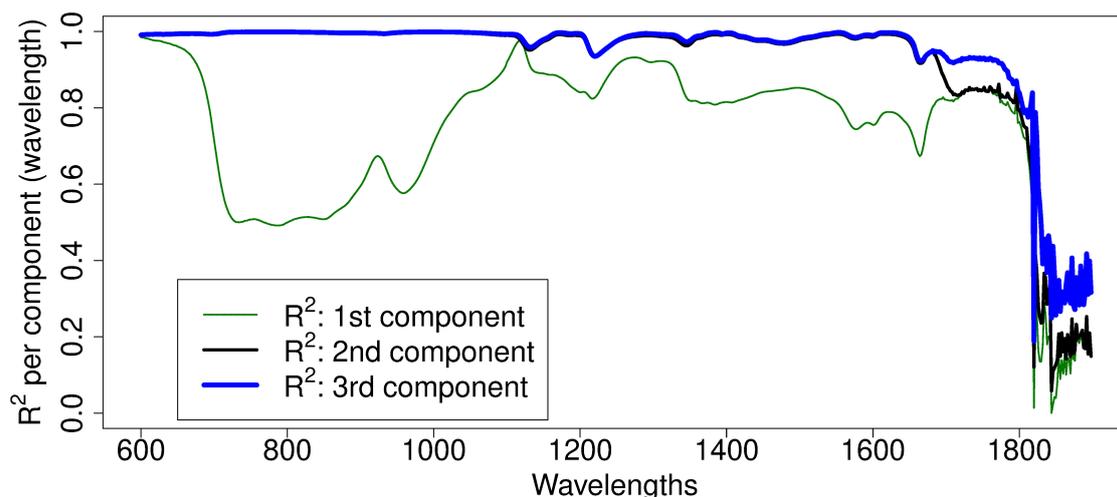
Finally, we can show the SPE plot for each observation. SPE values for each tablet become smaller and smaller as each successive component is added. Since each new component explains additional variance, the size of SPE must decrease. There don't appear to be any major outliers off the model's plane after the first component.
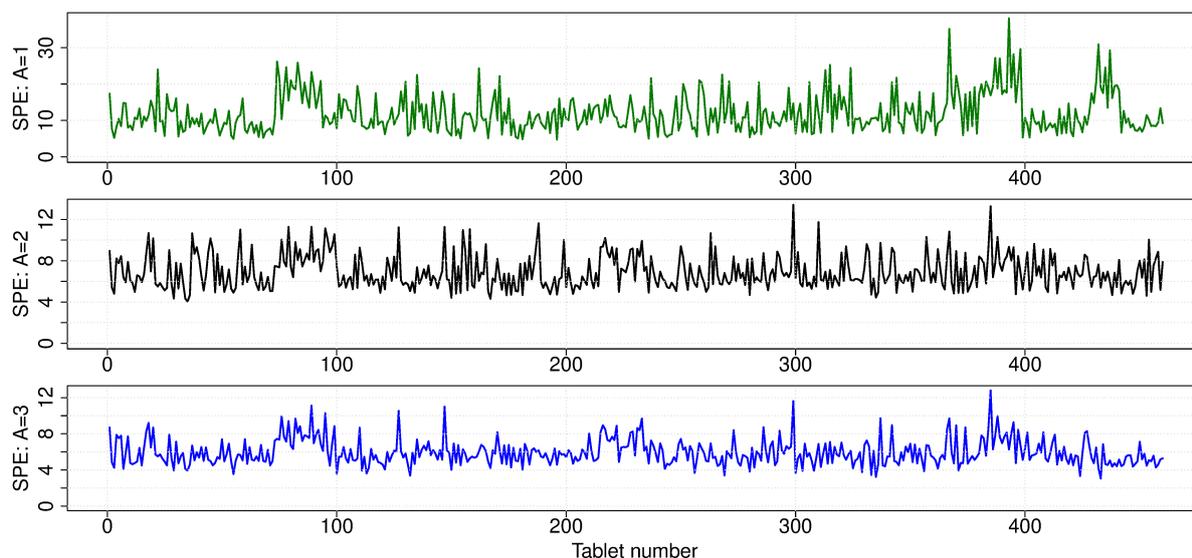
The code for the above plots is:

```
                                              R code
file <- 'http://openmv.net/file/tablet-spectra.csv'
spectra <- read.csv(file, header = FALSE, row.names = 1)



# Only extract 4 components, but
# center and scale the data before
# calculation the components
model.pca <- prcomp(spectra,
                    center = TRUE,
                    scale =TRUE,
                    rank. = 4)
spectra.P <- model.pca$rotation
spectra.T <- model.pca$x

# Baseline: mean and standard deviation per column
spectra.mean <- apply(spectra, 2, mean, na.rm=TRUE)
spectra.sd <- apply(spectra, 2, sd, na.rm=TRUE)
```

```r
# Remove the calculated mean from each column (margin=2)
# by using the subtract function (FUN argument)
spectra.mc <- sweep(spectra, 2, spectra.mean, FUN='-')

# Scale each column, dividing by the standard deviation
spectra.mcuv <- sweep(spectra.mc, 2, spectra.sd, FUN='/')

# Baseline variance
spectra.X2 <- spectra.mcuv * spectra.mcuv

# A = 1
#------
a = 1
spectra.Xhat.a <- spectra.T[,seq(1,a)] %*% t(spectra.P[,seq(1,a)])
spectra.E <- spectra.mcuv - spectra.Xhat.a
spectra.E2 <- spectra.E * spectra.E
spectra.Xhat.a.2 <- spectra.Xhat.a * spectra.Xhat.a

SPE.1 <- sqrt(apply(spectra.E2, 1, sum))
R2.k.a <- apply(spectra.Xhat.a.2, 2, sum) / apply(spectra.X2, 2, sum)

wavelengths <- seq(600, 1898, 2)
plot(wavelengths, R2.k.a, col='darkgreen',
     type='l', lwd=a*2, ylim=c(0,1),
     ylab=expression("R"^2*" per component (wavelength)"),
     xlab="Wavelengths")

# A = 2
#------
a = 2
spectra.Xhat.a <- spectra.T[,seq(1,a)] %*% t(spectra.P[,seq(1,a)])
spectra.E <- spectra.mcuv - spectra.Xhat.a

# mean for each row
spectra.E.mean <- apply(spectra.E, 1, mean, na.rm=TRUE)
spectra.E2 <- spectra.E * spectra.E
spectra.Xhat.a.2 <- spectra.Xhat.a * spectra.Xhat.a

SPE.2 <- sqrt(apply(spectra.E2, 1, sum))
R2.k.a <- apply(spectra.Xhat.a.2, 2, sum) / apply(spectra.X2, 2, sum)

lines(wavelengths, R2.k.a, col='black', type='l', lwd=a*2)


# A = 3
#------
a = 3
spectra.Xhat.a <- spectra.T[,seq(1,a)] %*% t(spectra.P[,seq(1,a)])
spectra.E <- spectra.mcuv - spectra.Xhat.a
spectra.E2 <- spectra.E * spectra.E
spectra.Xhat.a.2 <- spectra.Xhat.a * spectra.Xhat.a

SPE.3 <- sqrt(apply(spectra.E2, 1, sum))
R2.k.a <- apply(spectra.Xhat.a.2, 2, sum) / apply(spectra.X2, 2, sum)

lines(wavelengths, R2.k.a, col='blue', type='l', lwd=a*2)

legend(x=650, y=0.35,
       legend=c(expression("R"^2*": 1st component"),
                expression("R"^2*": 2nd component"),
                expression("R"^2*": 3rd component")),
       col=c("darkgreen", "black", "blue"),
       lty=c(1, 1, 1), lwd=c(2,4,6), cex=1.0)


# SPE plot
N <- dim(spectra)[1]
```

```
layout(matrix(c(1,2,3), 3, 1))
plot(seq(1, N), SPE.1, col='darkgreen',
     type='l', lwd=2,  ylab="SPE: A=1",
     ylim=c(0, max(SPE.1)))
plot(seq(1, N), SPE.2, col='black',
     type='l', lwd=2,  ylab="SPE: A=2",
     ylim=c(0, max(SPE.2)))
plot(seq(1, N), SPE.3, col='blue',
     type='l', lwd=2,  ylab="SPE: A=3",
     xlab="Tablet number", ylim=c(0, max(SPE.3)))
```

6.5.12 Hotelling's T^2

The final quantity from a PCA model that we need to consider is called Hotelling's T^2 value. Some PCA models will have many components, A, so an initial screening of these components using score scatterplots will require reviewing $A(A-1)/2$ scatterplots. The T^2 value for the ith observation is defined as:

$$T^2 = \sum_{a=1}^{a=A} \left(\frac{t_{i,a}}{s_a} \right)^2$$

where the s_a^2 values are constants, and are the variances of each component. The easiest interpretation is that T^2 is a scalar number that summarizes all the score values. Some other properties regarding T^2:

- It is a positive number, greater than or equal to zero.

- It is the distance from the center of the (hyper)plane to the projection of the observation onto the (hyper)plane.

- An observation that projects onto the model's center (usually the observation where every value is at the mean), has $T^2 = 0$.

- The T^2 statistic is distributed according to the F-distribution and is calculated by the multivariate software package being used. For example, we can calculate the 95% confidence limit for T^2, below which we expect, under normal conditions, to locate 95% of the observations.
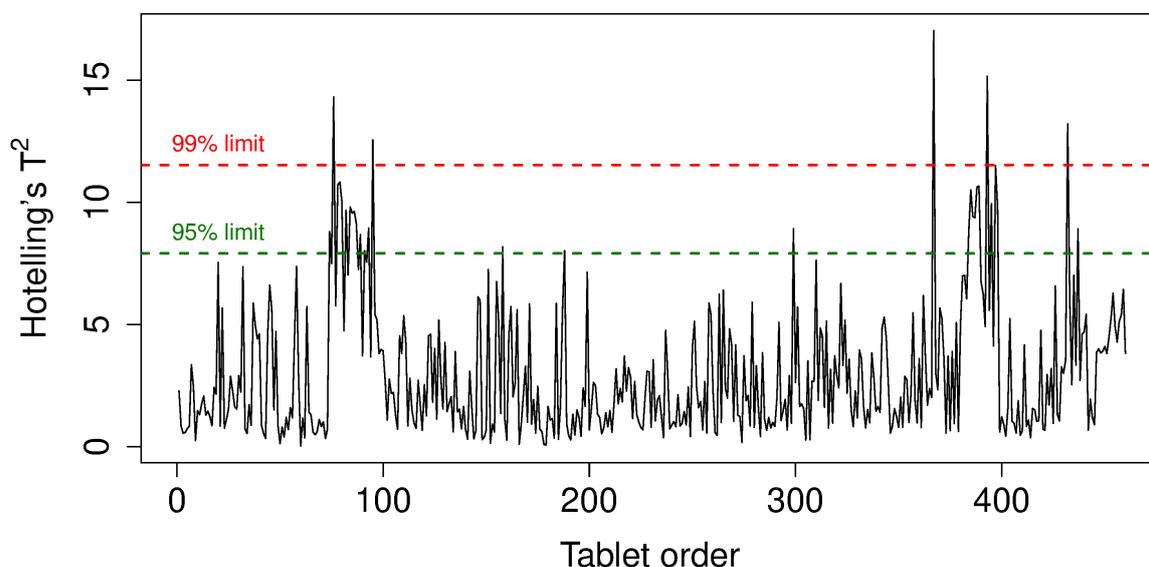
- It is useful to consider the case when $A = 2$, and fix the T^2 value at its 95% limit, for example, call that $T^2_{A=2,\alpha=0.95}$. Using the definition for T^2:

$$T^2_{A=2,\alpha=0.95} = \frac{t_1^2}{s_1^2} + \frac{t_2^2}{s_2^2}$$

  On a scatterplot of t_1 vs t_2 for all observations, this would be the equation of an ellipse, centered at the origin. You will often see this ellipse shown on t_i vs t_j scatterplots of the scores. Points inside this elliptical region are within the 95% confidence limit for T^2.

- The same principle holds for $A > 2$, except the ellipse is called a hyper-ellipse (think of a rugby-ball shaped object for $A = 3$). The general interpretation is that if a point is within this ellipse, then it is also below the T^2 limit, if T^2 were to be plotted on a line.

Score plot for tablet spectra
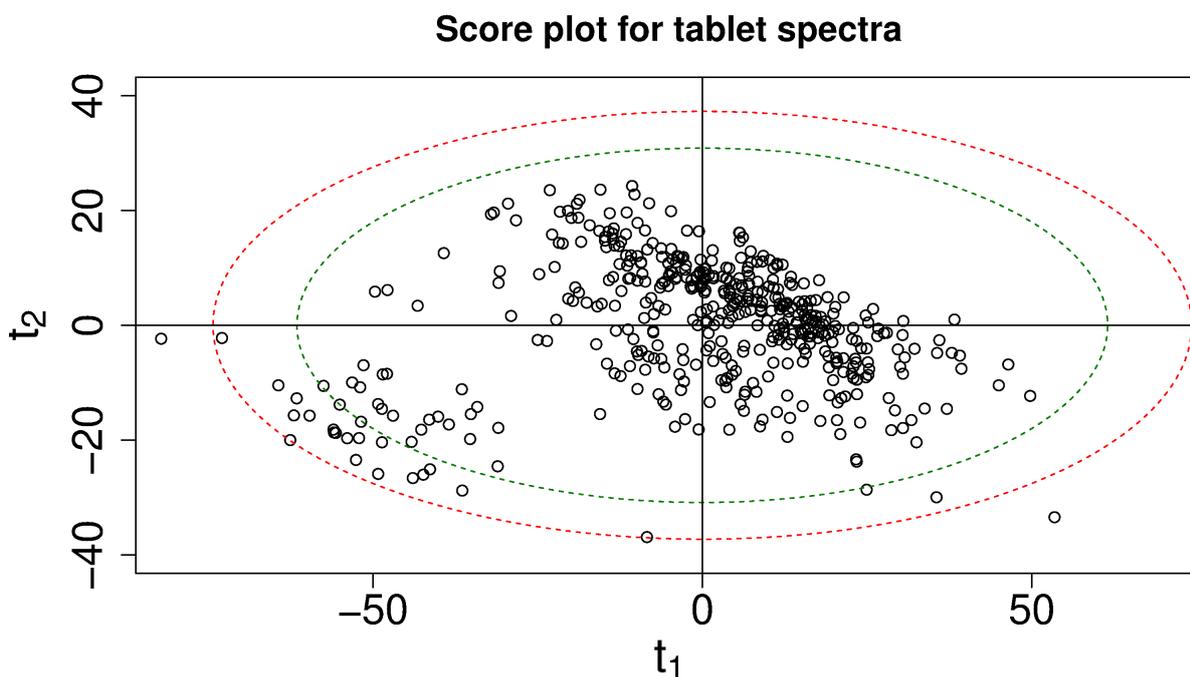
6.5.13 Preprocessing the data before building a model

The previous sections of this chapter considered the interpretation of a PCA latent variable model. From this section onwards we return to filling important gaps in our knowledge. There are 3 major steps to building any latent variable models:

1. Preprocessing the data

2. Building the latent variable model in the *algorithms section* (page 353)

3. *Testing the model* (page 358), including testing for the number of components to use.

We discuss the first step in this section, and the next two steps after that.

There are a number of possibilities for data preprocessing. We mainly discuss centering and scaling in this section, but outline a few other tools first. These steps are usually univariate, i.e. they are applied separately to each column in the raw data matrix $\mathbf{X}_{\text{raw}}$. We call the matrix of preprocessed data $\mathbf{X}$, this is the matrix that is then presented to the algorithm to build the latent variable model. Latent variable algorithms seldom work on the raw data.

Transformations

The columns in $\mathbf{X}_{\text{raw}}$ can be transformed: log, square-root and various powers (-1, -0.5, 0.5, 2) are popular options. These are used to reduce the effect of extreme measurements (e.g. log transforms), or because the transformed variable is known to be more correlated with the other variables. An example of this is in a distillation column: the inverse temperature is known to more correlated to the vapour pressure, which we know from first-principles modelling. Using the untransformed variable will lead to an adequate model, but the transformed variable, e.g. using the inverse temperature, can lead to a better model.

The tools we considered earlier on visualization and univariate distributions (histograms) can help assess which variables require transformation. But one's knowledge of the system is the most useful guide for knowing which transformations to apply. Note: latent variable models do not require each column in $\mathbf{X}_{\text{raw}}$ to be normally distributed: any type of quantitative variable may be used.

Expanding the X-matrix

Additional columns can and should be added to the **X**-matrix. This is frequently done in engineering systems where we can augment $\mathbf{X}_{\text{raw}}$ with columns containing heat, mass, and energy balances. It might be useful to add certain dimensionless numbers or other quantities that can be derived from the raw data.

Another step that is applied, usually to experimental data, is to add square and cross terms. For example, if 3 of the columns in $\mathbf{X}_{\text{raw}}$ were from a factorial designed experiment with center points, then augment $\mathbf{X}_{\text{raw}}$ with columns containing interaction terms: $x_1 x_2, \ x_1 x_3, \ x_2 x_3$. If face points or axial points (such as from a central composite design) were used, then also add the square terms to estimate the quadratic effects: $x_1^2, \ x_2^2, \ x_3^2$. When studying experimental data with latent variable methods (PCA or PLS), also add columns related to measured disturbance variables, often called covariates, and blocking variables - you won't know if they are important if they are not included.

The *general rule* is: add as many columns into $\mathbf{X}_{\text{raw}}$ as possible for the initial analysis. You can always prune out the columns later on if they are shown to be uninformative.

Dealing with outliers

Users often go through a phase of pruning outliers prior to building a latent variable model. There are often *uninteresting* outliers, for example when a temperature sensor goes off-line and provides a default reading of 0.0 instead of its usual values in the range of 300 to 400K. The automated tools used to do this are known by names such as trimming and winsorizing. These tools remove the upper and lower α percent of the column's tails on the histogram. But care should be taken with these automated approaches, since the most interesting observations are often in the outliers.

The course of action when removing outliers is to always mark their values as missing just for that variable in $\mathbf{X}_{\text{raw}}$, rather than removing the entire row in $\mathbf{X}_{\text{raw}}$. We do this because we can use the algorithms to calculate the latent variable model when missing data are present within a row.

Centering

Centering moves the coordinate system to a new reference point, usually the origin of the coordinate system in K variables (i.e. in K-dimensional space). Mean centering is effective and commonly used: after mean centering the mean of every column in $\mathbf{X}_{\text{raw}}$ will be exactly 0.0. An example of mean centering was given in the *food texture example* (page 332).

As we learned in the section on *univariate data analysis* (page 29), the mean has a low resistance to outliers: any large outlier will distort the value of the mean. So users often resort to trimming their data and then mean centering. In this regard, centering each column around its median is a better choice. We recommend median centering as it avoids the trimming step, and simultaneously highlights any outliers.

In the paper by Bro and Smilde on centering and scaling[152] they show how centering is far more influential on the model than scaling. Centering can be seen as adding a new principal component to the model, while scaling has much less of an effect.

Scaling

Scaling is an important step in latent variable modelling. Scaling can be seen as a way of assigning weights, or relative importance, to each column in $\mathbf{X}_{\text{raw}}$. If we don't know much about our data, then it is common to assign an equal weight to each column. We can do this by simply dividing each column by its standard deviation. After this scaling each column will have variance (and standard deviation) of exactly 1.0. This allows each column an equal opportunity of contributing to the model.

This sort of scaling is called unit-variance scaling. When combined with mean centering you will see the terminology that the data have been autoscaled.

Imagine a variable that is mostly constant, just noise. It will have a small standard deviation. When dividing by the standard deviation we artificially inflate its variance to the level of the other, truly-varying variables. These noisy, uninformative variables can be removed from $\mathbf{X}_{\text{raw}}$, or they can simply be multiplied by a smaller weight, so that their variance after preprocessing is less than 1.0. Such variables will also have small loading coefficients in all components, so they will be discovered during model investigation, if not sooner.

One could use the median absolute deviation (MAD) instead of the standard deviation to scale the columns, but it most cases, any approximate scaling vector will work adequately (see the Bro and Smilde paper referenced earlier).

6.5.14 Algorithms to calculate (build) PCA models

The different algorithms used to build a PCA model provide a different insight into the model's structure and how to interpret it. These algorithms are a reflection of how PCA has been used in different disciplines: PCA is called by different names in each area.

Eigenvalue decomposition

Note: The purpose of this section is not the theoretical details, but rather the interesting interpretation of the PCA model that we obtain from an eigenvalue decomposition.

Recall that the latent variable directions (the loading vectors) were oriented so that the variance of the scores in that direction were maximal. We can cast this as an optimization problem. For the first

---

[152] https://dx.doi.org/10.1002/cem.773

component:

$$\max \quad \phi = \mathbf{t}_1' \mathbf{t}_1 = \mathbf{p}_1' \mathbf{X}' \mathbf{X} \mathbf{p}_1$$
$$\text{s.t.} \quad \mathbf{p}_1' \mathbf{p}_1 = 1$$

This is equivalent to $\max \ \phi = \mathbf{p}_1' \mathbf{X}' \mathbf{X} \mathbf{p}_1 - \lambda \left(\mathbf{p}_1' \mathbf{p}_1 - 1 \right)$, because we can move the constraint into the objective function with a Lagrange multiplier, λ_1.

The maximum value must occur when the partial derivatives with respect to $\mathbf{p}_1$, our search variable, are zero:

$$\frac{\partial \phi}{\partial \mathbf{p}_1} = \mathbf{0} = \mathbf{p}_1' \mathbf{X}' \mathbf{X} \mathbf{p}_1 - \lambda_1 \left(\mathbf{p}_1' \mathbf{p}_1 - 1 \right)$$
$$\mathbf{0} = 2 \mathbf{X}' \mathbf{X} \mathbf{p}_1 - 2 \lambda_1 \mathbf{p}_1$$
$$\mathbf{0} = (\mathbf{X}' \mathbf{X} - \lambda_1 I_{K \times K}) \mathbf{p}_1$$
$$\mathbf{X}' \mathbf{X} \mathbf{p}_1 = \lambda_1 \mathbf{p}_1$$

which is just the eigenvalue equation, indicating that $\mathbf{p}_1$ is the eigenvector of $\mathbf{X}' \mathbf{X}$ and λ_1 is the eigenvalue. One can show that $\lambda_1 = \mathbf{t}_1' \mathbf{t}_1$, which is proportional to the variance of the first component.

In a similar manner we can calculate the second eigenvalue, but this time we add the additional constraint that $\mathbf{p}_1 \perp \mathbf{p}_2$. Writing out this objective function and taking partial derivatives leads to showing that $\mathbf{X}' \mathbf{X} \mathbf{p}_2 = \lambda_2 \mathbf{p}_2$.

From this we learn that:

- The loadings are the eigenvectors of $\mathbf{X}' \mathbf{X}$.

- Sorting the eigenvalues in order from largest to smallest gives the order of the corresponding eigenvectors, the loadings.

- We know from the theory of eigenvalues that if there are distinct eigenvalues, then their eigenvectors are linearly independent (orthogonal).

- We also know the eigenvalues of $\mathbf{X}' \mathbf{X}$ must be real values and positive; this matches with the interpretation that the eigenvalues are proportional to the variance of each score vector.

- Also, the sum of the eigenvalues must add up to sum of the diagonal entries of $\mathbf{X}' \mathbf{X}$, which represents of the total variance of the $\mathbf{X}$ matrix, if all eigenvectors are extracted. So plotting the eigenvalues is equivalent to showing the proportion of variance explained in $\mathbf{X}$ by each component. This is not necessarily a good way to judge the number of components to use, but it is a rough guide: use a Pareto plot of the eigenvalues (though in the context of eigenvalue problems, this plot is called a scree plot).

Pareto plot (scree plot) of R^2 per component
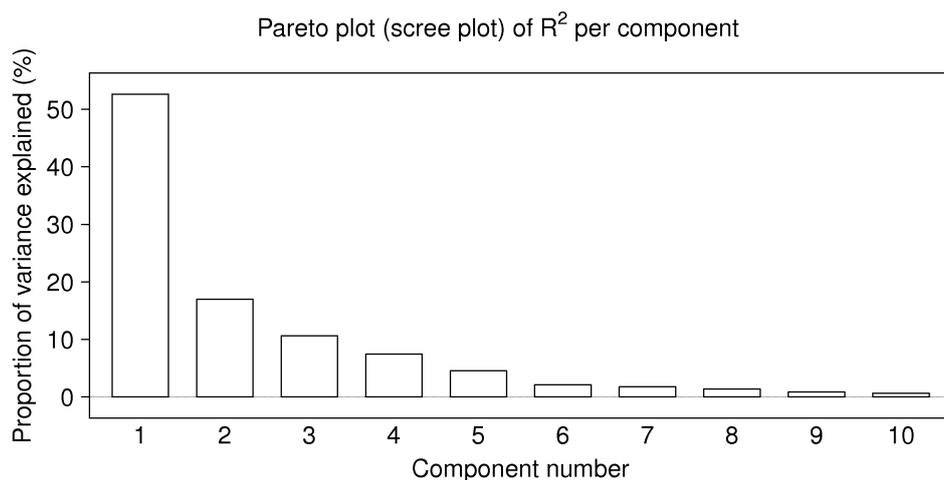
The general approach to using the eigenvalue decomposition would be:

1. Preprocess the raw data, particularly centering and scaling, to create a matrix $\mathbf{X}$.

2. Calculate the correlation matrix $\mathbf{X}'\mathbf{X}$.

3. Calculate the eigenvectors and eigenvalues of this square matrix and sort the results from largest to smallest eigenvalue.

4. A rough guide is to retain only the first A eigenvectors (loadings), using a Scree plot of the eigenvalues as a guide. Alternative methods to determine the number of components are described in the section on cross-validation and randomization.

However, we should note that calculating the latent variable model using an eigenvalue algorithm is usually not recommended, since it calculates all eigenvectors (loadings), even though only the first few will be used. The maximum number of components possible is $A_{\max} = \min(N, K)$. Also, the default eigenvalue algorithms in software packages cannot handle missing data.

Singular value decomposition

The singular value decomposition (SVD), in general, decomposes a given matrix $\mathbf{X}$ into three other matrices:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}'$$

Matrices $\mathbf{U}$ and $\mathbf{V}$ are orthonormal (each column has unit length and each column is orthogonal to the others), while $\mathbf{\Sigma}$ is a diagonal matrix. The relationship to principal component analysis is that:

$$\mathbf{X} = \mathbf{T}\mathbf{P}'$$

where matrix $\mathbf{P}$ is also orthonormal. So taking the SVD on our preprocessed matrix $\mathbf{X}$ allows us to get the PCA model by setting $\mathbf{P} = \mathbf{V}$, and $\mathbf{T} = \mathbf{U}\mathbf{\Sigma}$. The diagonal terms in $\mathbf{\Sigma}$ are related to the variances of each principal component and can be plotted as a scree plot, as was done for the *eigenvalue decomposition* (page 353).

Like the eigenvalue method, the SVD method calculates all principal components possible, $A = \min(N, K)$, and also cannot handle missing data by default.

Non-linear iterative partial least-squares (NIPALS)

The non-linear iterative partial least squares (NIPALS) algorithm is a sequential method of computing the principal components. The calculation may be terminated early, when the user deems that enough components have been computed. Most computer packages tend to use the NIPALS algorithm as it has two main advantages: it handles missing data and calculates the components sequentially.

The purpose of considering this algorithm here is three-fold: it gives additional insight into what the loadings and scores mean; it shows how each component is independent of (orthogonal to) the other components, and it shows how the algorithm can handle missing data.

The algorithm extracts each component sequentially, starting with the first component, direction of greatest variance, and then the second component, and so on.

We will show the algorithm here for the ath component, where $a = 1$ for the first component. The matrix $\mathbf{X}$ that we deal with below is the *preprocessed* (page 351), usually centered and scaled matrix, not the raw data.

1. The NIPALS algorithm starts by arbitrarily creating an initial column for $\mathbf{t}_a$. You can use a column of random numbers, or some people use a column from the $\mathbf{X}$ matrix; anything can be used as long as it is not a column of zeros.

2. Then we take every column in $\mathbf{X}$, call it $\mathbf{X}_k$, and regress it onto this initial column $\mathbf{t}_a$; store the regression coefficient as the entry in $p_{k,a}$. What this means, and it is illustrated below, is that we perform an ordinary least squares regression ($\mathbf{y} = \beta\mathbf{x}$), except our x-variable is this column of $\mathbf{t}_a$ values, and our y-variable is the particular column from $\mathbf{X}$.
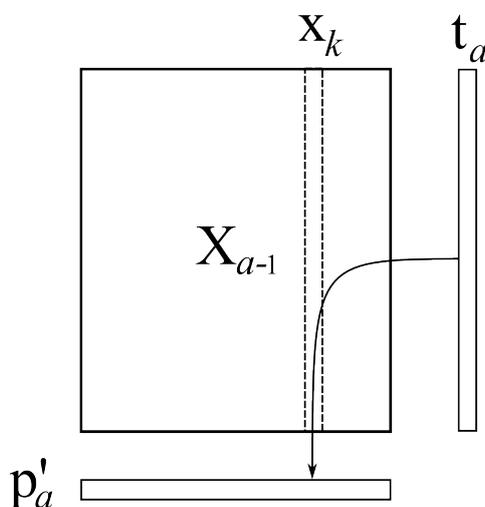
For ordinary least squares, you will remember that the solution for this regression problem is $\widehat{\beta} = \dfrac{\mathbf{x}'\mathbf{y}}{\mathbf{x}'\mathbf{x}}$. Using our notation, this means:

$$p_{k,a} = \frac{\mathbf{t}'_a \mathbf{X}_k}{\mathbf{t}'_a \mathbf{t}_a}$$

This is repeated for each column in $\mathbf{X}$ until we fill the entire vector $\mathbf{p}_k$. This is shown in the illustration where each column from $\mathbf{X}$ is regressed, one at a time, on $\mathbf{t}_a$, to calculate the loading entry, $p_{k,a}$ In practice we don't do this one column at time; we can regress all columns in $\mathbf{X}$ in go: $\mathbf{p}'_a = \dfrac{1}{\mathbf{t}'_a \mathbf{t}_a} \cdot \mathbf{t}'_a \mathbf{X}_a$, where $\mathbf{t}_a$ is an $N \times 1$ column vector, and $\mathbf{X}_a$ is an $N \times K$ matrix, explained more clearly in step 6.

3. The loading vector $\mathbf{p}'_a$ won't have unit length (magnitude) yet. So we simply rescale it to have magnitude of 1.0:

$$\mathbf{p}'_a = \frac{1}{\sqrt{\mathbf{p}'_a \mathbf{p}_a}} \cdot \mathbf{p}'_a$$

4. The next step is to regress every row in $\mathbf{X}$ onto this normalized loadings vector. As illustrated below, in our linear regression the rows in $\mathbf{X}$ are our y-variable each time, while the loadings vector is our x-variable. The regression coefficient becomes the score value for that i^{th} row:
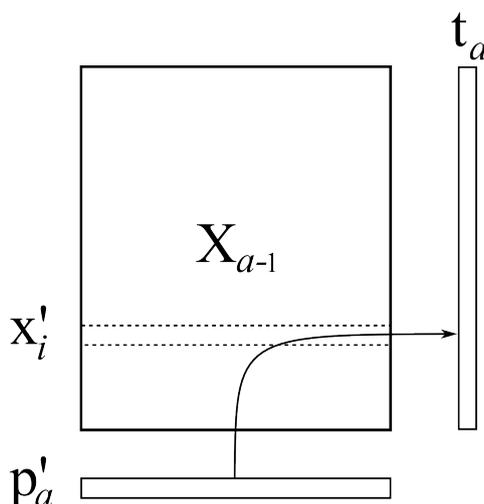
$$t_{i,a} = \frac{\mathbf{x}'_i \mathbf{p}_a}{\mathbf{p}'_a \mathbf{p}_a}$$

where $\mathbf{x}'_i$ is an $K \times 1$ column vector. We can combine these N separate least-squares models and calculate them in one go to get the entire vector, $\mathbf{t}_a = \dfrac{1}{\mathbf{p}'_a \mathbf{p}_a} \cdot \mathbf{X} \mathbf{p}_a$, where $\mathbf{p}_a$ is a $K \times 1$ column vector.

5. We keep iterating steps 2, 3 and 4 until the change in vector $\mathbf{t}_a$ from one iteration to the next is small (usually around 1×10^{-6} to 1×10^{-9}). Most data sets require no more than 200 iterations before achieving convergence.

6. On convergence, the score vector and the loading vectors, $\mathbf{t}_a$ and $\mathbf{p}_a$ are stored as the ath column in matrix $\mathbf{T}$ and $\mathbf{P}$ respectively. We then deflate the $\mathbf{X}$ matrix. This crucial step removes the variability captured in this component ($\mathbf{t}_a$ and $\mathbf{p}_a$) from $\mathbf{X}$:

$$\mathbf{E}_a = \mathbf{X}_a - \mathbf{t}_a \mathbf{p}'_a$$
$$\mathbf{X}_{a+1} = \mathbf{E}_a$$

For the first component, $\mathbf{X}_a$ is just the preprocessed raw data. So we can see that the second component is actually calculated on the residuals $\mathbf{E}_1$, obtained after extracting the first component.

This is called *deflation,* and nicely shows why each component is orthogonal to the others. Each subsequent component is only seeing variation remaining after removing all the others; there is no possibility that two components can explain the same type of variability.

After deflation we go back to step 1 and repeat the entire process for the next component. Just before accepting the new component we can use a test, such as a randomization test, or *cross-validation* (page 360), to decide whether to keep that component or not.

The final reason for outlining the NIPALS algorithm is to show one way in which missing data can be handled. All that step 2 and step 4 are doing is a series of regressions. Let's take step 2 to illustrate, but the same idea holds for step 4. In step 2, we were regressing columns from $\mathbf{X}$ onto the score $\mathbf{t}_a$. We can visualize this for a hypothetical system below

There are 3 missing observations (open circles), but despite this, the regression's slope can still be adequately determined. The slope is unlikely to change by very much if we did have the missing values. In practice though we have no idea where these open circles would fall, but the principle is the same: we calculate the slope coefficient just ignoring any missing entries.
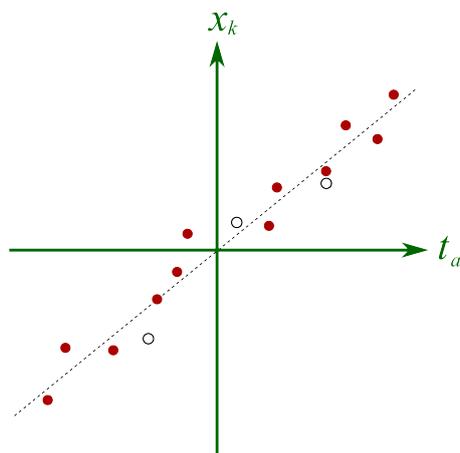
In summary:

- The NIPALS algorithm computes one component at a time. The first component computed is equivalent to the $\mathbf{t}_1$ and $\mathbf{p}_1$ vectors that would have been found from an eigenvalue or singular value decomposition.

- The algorithm can handle missing data in $\mathbf{X}$.

- The algorithm always converges, but the convergence can sometimes be slow.

- It is also known as the Power algorithm to calculate eigenvectors and eigenvalues.

- It works well for very large data sets.

- It is used by most software packages, especially those that handle missing data.

- Of interest: it is well known that Google used this algorithm for the early versions of their search engine, called PageRank[153].

6.5.15 Testing the PCA model

As mentioned previously there are 3 major steps to building a PCA model for engineering applications. We have already considered the first two steps in the preceding sections.

1. Preprocessing the data

2. Building the latent variable model

3. Testing the model, including testing for the number of components to use

The last step of testing, interpreting and using the model is where one will spend the most time. Preparing the data can be time-consuming the first time, but generally the first two steps are less

---

[153] http://ilpubs.stanford.edu:8090/422/

time-consuming. In this section we investigate how to determine the number of components that should be used in the model and how to use an existing latent variable model. The issue of interpreting a model has been addressed in the section on *interpreting scores* (page 336) and *interpreting loadings* (page 339).

Using an existing PCA model

In this section we outline the process required to use an existing PCA model. What this means is that you have already calculated the model and validated its usefulness. Now you would like to use the model on a new observation, which we call $\mathbf{x}'_{\text{new, raw}}$. The method described below can be efficiently applied to many new rows of observations by converting the row vector notation to matrix notation.

1. Preprocess your vector of new data in the same way as you did when you built the model. For example, if you took the log transform of a certain variable, then you must do so for the corresponding entry in $\mathbf{x}'_{\text{new, raw}}$. Also apply mean centering and scaling, using the mean centering and scaling information you calculated when you originally built the model.

2. Call this preprocessed vector $\mathbf{x}_{\text{new}}$ now; it has size $K \times 1$, so $\mathbf{x}'_{\text{new}}$ is a $1 \times K$ row vector.

3. Calculate the location, on the model (hyper)plane, where the new observation would project. In other words, we are calculating the scores:

$$\mathbf{t}'_{\text{new}} = \mathbf{x}'_{\text{new}} \mathbf{P}$$

    where $\mathbf{P}$ is the $K \times A$ matrix of loadings calculated when building the model, and $\mathbf{t}'_{\text{new}}$ is a $1 \times A$ vector of scores for the new observation.

4. Calculate the residual distance off the model plane. To do this, we require the vector called $\widehat{\mathbf{x}}'_{\text{new}}$, the point on the plane, a $1 \times K$ vector:

$$\widehat{\mathbf{x}}'_{\text{new}} = \mathbf{t}'_{\text{new}} \mathbf{P}'$$

5. The residual vector is the difference between the actual observation and its projection onto the plane. The K individual entries inside this residual vector are also the called the *contributions* to the error.

$$\mathbf{e}'_{\text{new}} = \mathbf{x}'_{\text{new}} - \widehat{\mathbf{x}}'_{\text{new}}$$

6. And the residual distance is the sum of squares of the entries in the residual vector, followed by taking a square root.

$$\text{SPE}_{\text{new}} = \sqrt{\mathbf{e}'_{\text{new}} \mathbf{e}_{\text{new}}}$$

    This is called the squared prediction error, SPE, even though it is more accurately a distance.

7. Another quantity of interest is Hotelling's T^2 value for the new observation:

$$T^2_{\text{new}} = \sum_{a=1}^{a=A} \left(\frac{t_{\text{new},a}}{s_a} \right)^2$$

    where the s_a values are the standard deviations for each component's scores, calculated when the model was built.

The above outline is for the case when there is no missing data in a new observation. When there are missing data present in $\mathbf{x}'_{\text{new}}$, then we require a method to estimate the score vector, $\mathbf{t}_{\text{new}}$ in step 3. Methods for doing this are outlined and compared in the paper by Nelson, Taylor and MacGregor[154]

---

[154] https://dx.doi.org/10.1016/S0169-7439(96)00007-X

and the paper by Arteaga and Ferrer[155]. After that, the remaining steps are the same, except of course that missing values do not contribute to the residual vector and the SPE.

6.5.16 Determining the number of components to use in the model with cross-validation

Cross-validation is a general tool that helps to avoid over-fitting - it can be applied to any model, not just latent variable models.

As we add successive components to a model we are increasing the size of the model, A, and we are explaining the model-building data, $\mathbf{X}$, better and better. (The equivalent in least squares models would be to add additional $\mathbf{X}$-variable terms to the model.) The model's R^2 value will increase with every component. As the following equation shows, the variance of the $\widehat{\mathbf{X}}$ matrix increases with every component, while the residual variance in matrix $\mathbf{E}$ must decrease.

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E}$$
$$\mathbf{X} = \widehat{\mathbf{X}} + \mathbf{E}$$
$$\mathcal{V}(\mathbf{X}) = \mathcal{V}(\widehat{\mathbf{X}}) + \mathcal{V}(\mathbf{E})$$

This holds for any model where the $\widehat{\mathbf{X}}$ and $\mathbf{E}$ matrices are completely orthogonal to each other: $\widehat{\mathbf{X}}'\mathbf{E} = \mathbf{0}$ (a matrix of zeros), such as in PCA, PLS and least squares models.

There comes a point for any real data set where the number of components, A = the number of columns in $\mathbf{T}$ and $\mathbf{P}$, extracts all systematic variance from $\mathbf{X}$, leaving unstructured residual variance in $\mathbf{E}$. Fitting any further components will start to fit this noise, and unstructured variance, in $\mathbf{E}$.

Cross-validation for multivariate data sets was described by Svante Wold in his paper on Cross-validatory estimation of the number of components in factor and principal components models[156], in *Technometrics*, **20**, 397-405, 1978.

The general idea is to divide the matrix $\mathbf{X}$ into G groups of rows. These rows should be selected randomly, but are often selected in order: row 1 goes in group 1, row 2 goes in group 2, and so on. We can collect the rows belonging to the first group into a new matrix called $\mathbf{X}_{(1)}$, and leave behind all the other rows from all other groups, which we will call group $\mathbf{X}_{(-1)}$. So in general, for the gth group, we can split matrix $\mathbf{X}$ into $\mathbf{X}_{(g)}$ and $\mathbf{X}_{(-g)}$.

Wold's cross-validation procedure asks to build the PCA model on the data in $\mathbf{X}_{(-1)}$ using A components. Then use data in $\mathbf{X}_{(1)}$ as new, testing data. In other words, we preprocess the $\mathbf{X}_{(1)}$ rows, calculate their score values, $\mathbf{T}_{(1)} = \mathbf{X}_{(1)}\mathbf{P}$, calculate their predicted values, $\widehat{\mathbf{X}}_{(1)} = \mathbf{T}_{(1)}\mathbf{P}'$, and their residuals, $\mathbf{E}_{(1)} = \mathbf{X}_{(1)} - \widehat{\mathbf{X}}_{(1)}$. We repeat this process, building the model on $\mathbf{X}_{(-2)}$ and testing it with $\mathbf{X}_{(2)}$, to eventually obtain $\mathbf{E}_{(2)}$.

After repeating this on G groups, we gather up $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_G$ and assemble a type of residual matrix, $\mathbf{E}_{A,\text{CV}}$, where the A represents the number of components used in each of the G PCA models. The CV subscript indicates that this is not the usual error matrix, $\mathbf{E}$. From this we can calculate a type of R^2 value. We don't call this R^2, but it follows the same definition for an R^2 value. We will call it Q_A^2 instead, where A is the number of components used to fit the G models.

$$Q_A^2 = 1 - \frac{\text{Var}(\mathbf{E}_{A,\text{CV}})}{\text{Var}(\mathbf{X})}$$

---

[155] https://dx.doi.org/10.1002/cem.750
[156] https://www.jstor.org/stable/1267639

We also calculate the usual PCA model on all the rows of $\mathbf{X}$ using A components, then calculate the usual residual matrix, $\mathbf{E}_A$. This model's R^2 value is:

$$R_A^2 = 1 - \frac{\text{Var}(\mathbf{E}_A)}{\text{Var}(\mathbf{X})}$$

The Q_A^2 behaves exactly as R^2, but with two important differences. Like R^2, it is a number less than 1.0 that indicates how well the testing data, in this case testing data that was generated by the cross-validation procedure, are explained by the model. The first difference is that Q_A^2 is always less than the R^2 value. The other difference is that Q_A^2 will not keep increasing with each successive component, it will, after a certain number of components, start to decrease. This decrease in Q_A^2 indicates the new component just added is not systematic: it is unable to explain the cross-validated testing data. We often see plots such as this one:
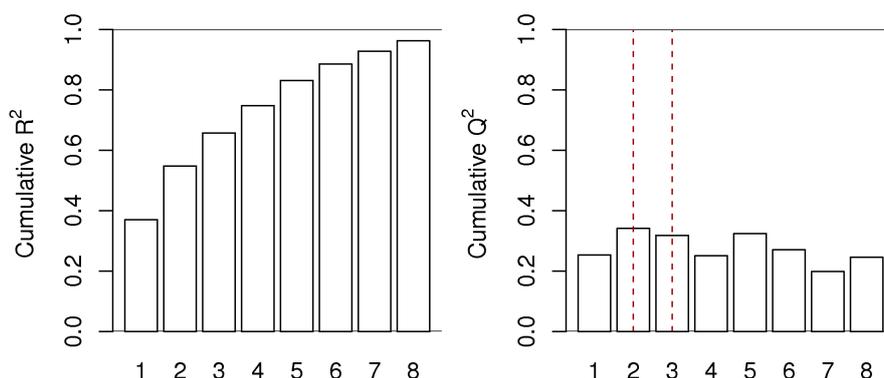
This is for a real data set, so the actual cut off for the number of components could be either $A = 2$ or $A = 3$, depending on what the 3rd component shows to the user and how interested they are in that component. Likely the 4th component, while boosting the R^2 value from 66% to 75%, is not really fitting any systematic variation. The Q^2 value drops from 32% to 25% when going from component 3 to 4. The fifth component shows Q^2 increasing again. Whether this is fitting actual variability in the data or noise is for the modeller to determine, by investigating that 5th component. These plots show that for this data set we would use between 2 and 5 components, but not more.

Cross-validation, as this example shows is never a precise answer to the number of components that should be retained when trying to learn more about a dataset. Many studies try to find the "true" or "best" number of components. This is a fruitless exercise; each data set means something different to the modeller and the objective for which the model was intended to assist.

The number of components to use should be judged by the relevance of each component. Use cross-validation as guide, and always look at a few extra components and step back a few components; then make a judgement that is relevant to your intended use of the model.

However, cross-validation's objective is useful for predictive models, such as PLS, so we avoid over-fitting components. Models where we intend to learn from, or optimize, or monitor a process may well benefit from fewer or more components than suggested by cross-validation.

6.5.17 Some properties of PCA models

We summarize various properties of the PCA model, most have been described in the previous sections. Some are only of theoretical interest, but others are more practical.

- The model is defined by the direction vectors, or loadings vectors, called $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_A$; each are a $K \times 1$ vector, and can be collected into a single matrix, $\mathbf{P}$, a $K \times A$ loadings matrix.

- These vectors form a line for one component, a plane for 2 components, and a hyperplane for 3 or more components. This line, plane or hyperplane define the latent variable model.

- An equivalent interpretation of the model plane is that these direction vectors are oriented in such a way that the scores have maximal variance for that component. No other directions of the loading vector (i.e. no other hyperplane) will give a greater variance.

- This plane is calculated with respect to a given data set, $\mathbf{X}$, an $N \times K$ matrix, so that the direction vectors best-fit the data. We can say then that with one component, the best estimate of the original matrix $\mathbf{X}$ is:

$$\widehat{\mathbf{X}}_1 = \mathbf{t}_1 \mathbf{p}_1 \qquad \text{or equivalently:} \qquad \mathbf{X}_1 = \mathbf{t}_1 \mathbf{p}_1 + \mathbf{E}_1$$

where $\mathbf{E}_1$ is the residual matrix after fitting one component. The estimate for $\mathbf{X}$ will have smaller residuals if we fit a second component:

$$\widehat{\mathbf{X}}_2 = \mathbf{t}_1 \mathbf{p}_1 + \mathbf{t}_2 \mathbf{p}_2 \qquad \text{or equivalently:} \qquad \mathbf{X}_2 = \mathbf{t}_1 \mathbf{p}_1 + \mathbf{t}_2 \mathbf{p}_2 + \mathbf{E}_2$$

In general we can illustrate this:
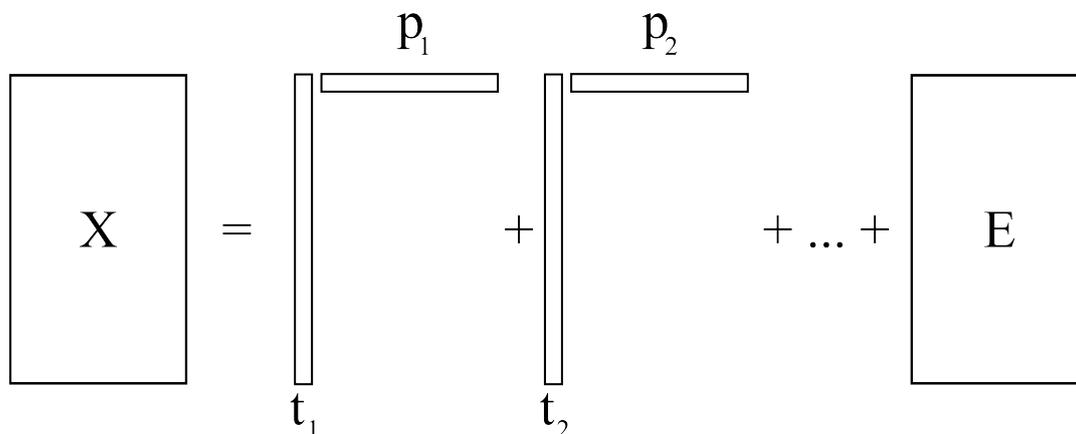
- The loadings vectors are of unit length: $\|\mathbf{p}_a\| = \sqrt{\mathbf{p}_a' \mathbf{p}_a} = 1.0$

- The loading vectors are independent or orthogonal to one another: $\mathbf{p}_i' \mathbf{p}_j = 0.0$ for $i \neq j$; in other words $\mathbf{p}_i \perp \mathbf{p}_j$.

- Orthonormal matrices have the property that $\mathbf{P}'\mathbf{P} = \mathbf{I}_A$, an identity matrix of size $A \times A$.

- These last 3 properties imply that $\mathbf{P}$ is an orthonormal matrix. From matrix algebra and geometry you will recall that this means $\mathbf{P}$ is a rigid rotation matrix. We are rotating our real-world data in $\mathbf{X}$ to a new set of values, scores, using the rotation matrix $\mathbf{P}$. But a rigid rotation implies that distances and angles between observations are preserved. Practically, this means that by looking at our data in the score space, points which are close together in the original K variables will be close to each other in the scores, $\mathbf{T}$, now reduced to A variables.

- The variance of the $\mathbf{t}_1$ vector must be greater than the variance of the $\mathbf{t}_2$ vector. This is because we intentionally find the components in this manner. In our notation: $s_1 > s_2 > \ldots > s_A$, where s_a is the standard deviation of the ath score.

- The maximum number of components that can be extracted is the smaller of N or K; but usually we will extract only $A \ll K$ number of components. If we do extract all components, $A^* = \min(N, K)$, then our loadings matrix, $\mathbf{P}$, merely rotates our original coordinate system to a new system without error.

- The eigenvalue decomposition of $\mathbf{X}'\mathbf{X}$ gives the loadings, $\mathbf{P}$, as the eigenvectors, and the eigenvalue for each eigenvector is the variance of the score vector.

- The singular value decomposition of $\mathbf{X}$ is given by $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}'$, so $\mathbf{V}' = \mathbf{P}'$ and $\mathbf{U}\mathbf{\Sigma} = \mathbf{T}$, showing the equivalence between PCA and this method.

- If there are no missing values in $\mathbf{X}$, then the mean of each score vector is 0.0, which allows us to calculate the variance of each score simply from $\mathbf{t}'_a \mathbf{t}_a$.

- Notice that some score values are positive and others negative. Each loading direction, $\mathbf{p}_a$, must point in the direction that best explains the data; but this direction is not unique, since $-\mathbf{p}_a$ also meets this criterion. If we did select $-\mathbf{p}_a$ as the direction, then the scores would just be $-\mathbf{t}_a$ instead. This does not matter too much, because $(-\mathbf{t}_a)(-\mathbf{p}'_a) = \mathbf{t}_a\mathbf{p}'_a$, which is used to calculate the predicted $\mathbf{X}$ and the residuals. But this phenomena can lead to a confusing situation for newcomers when different computer packages give different-looking loading plots and score plots for the same data set.

6.5.18 Latent variable contribution plots

We have *previously seen* (page 330) how contribution plots are constructed for a score value, for the SPE and for T^2. We breakdown the value, such as SPE, into its individual terms, one from each variable. Then we plot these K contribution values as a bar chart.

There are K contribution terms for a score: $t_{i,a} = \mathbf{x}_i\mathbf{p}_a$:

$$\begin{bmatrix} x_{i,1}\ p_{1,a} & x_{i,2}\ p_{2,a} & \cdots & x_{i,k}\ p_{k,a} & \cdots & x_{i,K}\ p_{K,a} \end{bmatrix}$$

The contribution to T^2 is similar to a score contribution, except we calculate the weighted summation over all the scores, $t_{i,a}$, where the weights are the variances of the ath score.

For SPE $= \sqrt{\mathbf{e}'_i\mathbf{e}_i}$, where $\mathbf{e}'_i = \mathbf{x}'_i - \hat{\mathbf{x}}'_i$, the bars in the contribution plots are:

$$\begin{bmatrix} (x_{i,1} - \hat{x}_{i,1}) & (x_{i,2} - \hat{x}_{i,2}) & \cdots & (x_{i,k} - \hat{x}_{i,k}) & \cdots & (x_{i,K} - \hat{x}_{i,K}) \end{bmatrix}$$

The SPE contributions are usually shown as the square of the values in brackets, accounting for the sign, as in $e_{i,k} = (x_{i,k} - \hat{x}_{i,k})$, and then plot each bar: $\text{sign}(e_{i,k}) \times e_{i,k}^2$. The squared values are more realistic indicators of the contributions, while the sign information might be informative in some cases.

The other point to mention here is that contributions are calculated *from* one point *to* another point. Most often, the *from* point is the model center or the model plane. So for SPE, the contributions are *from* the model plane *to* the ith observation off the model plane. The score contributions are *from* the model center *to* the observation's projection on the (hyper)plane.

But sometimes we would like to know, as in the figure below, what are the contribution from one point to another. And these start and end points need not be an actual point; for a group of points we can

use a suitable average of the points in the cluster. So there are point-to-point, point-to-group, group-to-point, and group-to-group contributions in the scores.

From the model center to a point   Four seperate contribution plots to learn   From one group to another group
                                      why the sequence of deviations occurred
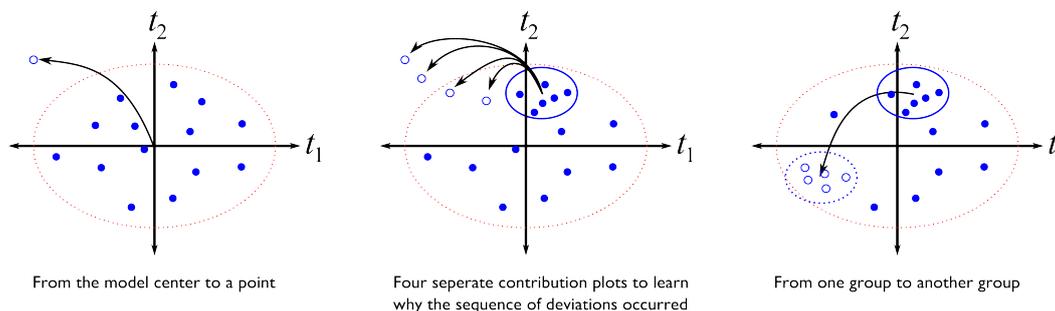
The calculation procedure is actually the same in all cases: for a group of points, collapse it down to the center point in the group, then calculate the point-to-point contribution. If the starting point is not specified, then the contribution will be from the model center, i.e. $(t_i, t_j) = (0, 0)$ to the point.

6.5.19 Using indicator variables in a latent variable model

Indicator variables, also called dummy variables, are most often binary variables that indicate the presence or absence of a certain effect. For example, a variable that shows if reactor A or reactor B was used. Its value is either a 0 or a 1 in the data matrix $\mathbf{X}$. It's valid to include these sort of variables in a principal component analysis model where they are used and interpreted as any other continuous variable.

Sometimes these variables are imported into the computer software, but *not used in the model*. They are only used in the display of results, where the indicator variable is shown in a different colour or with a different marker shape. We will see *an example of this for process troubleshooting* (page 393), to help isolate causes for poor yield from a process:
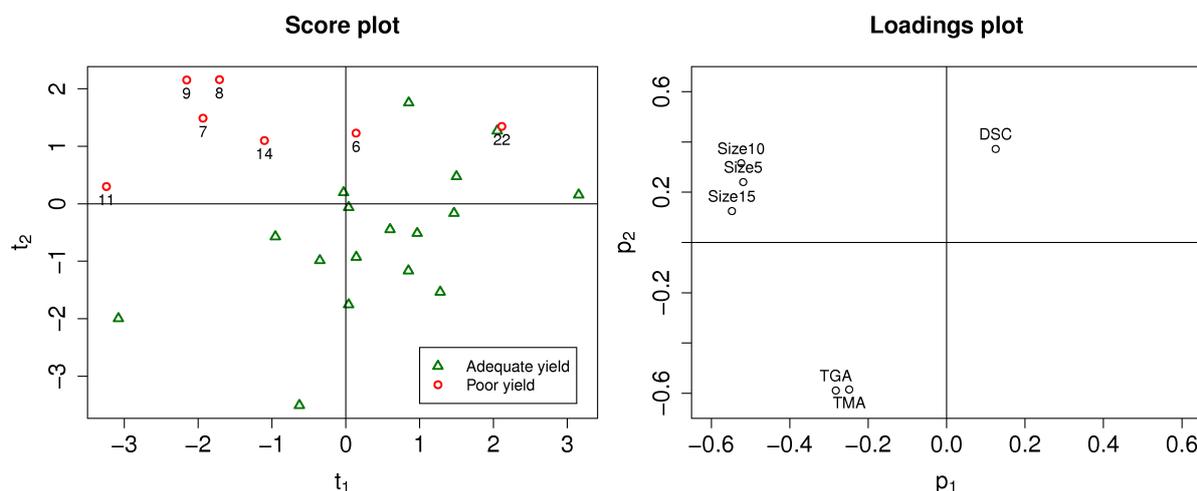
If the variable is included in the model then it is centered and scaled (preprocessed) like any other variable. Care must be taken to make sure this variable is reasonably balanced. There is no guide as to how balanced it needs to be, but there should be a good number of observations of both zeros and ones. The extreme case is where there are N observations, and only 1 of them is a zero or a one, and the other $N - 1$ observations are the rest. You are not likely to learn much from this variable in any case; furthermore, the scaling for this variable will be poor (the variance will be small, so dividing by this small variance will inflate that variable's variance).

Interpreting these sort of variables in a loading plot is also no different; strong correlations with this variable are interpreted in the usual way.

6.5.20 Visualization latent variable models with linking and brushing

Linking is when the same data point(s), are highlighted in two or more plots. This is used to highlight outliers or interesting points in a multivariate data set. The points could be highlighted in terms of colour and/or shape.

Brushing is the same as linking, except it is done in real-time as the user moves a mouse over a plot. This concept was described by Becker and Cleveland in their original article called Brushing Scatterplots[157], *Technometrics*, **29**, 127-142, 1987.
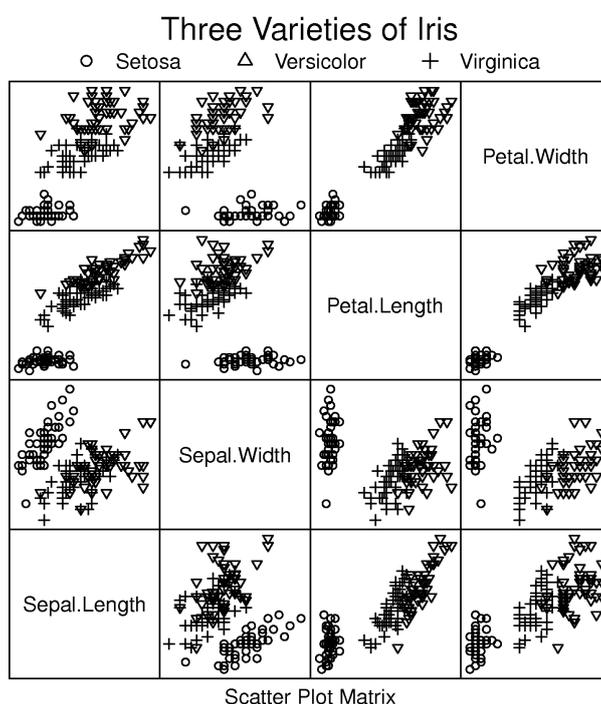
In this illustration we are considering the well-known iris data set, a multivariate data set consisting of the 4 length measurements taken on 3 species of iris. There are 150 observations (50 for each species). Linking is used to mark each iris species with a different marker shape (a different colour could also have been used). Brushing cannot be illustrated, but as shown in the paper by Becker and Cleveland, it would amount to dynamically changing the marker shape or colour of points in one plot, while the user selects those same observations in another plot.

This concept is very powerful to learn from, and to interrogate a latent variable model. For example, when we see interesting observations in the score plot, we can brush through the scores, while having a time series plot of the raw data open alongside. This would highlight what that score feature means in the context of the raw data.

---

[157] https://www.jstor.org/stable/1269768

6.5.21 PCA Exercises

Each exercise introduces a new topic or highlights some interesting aspect of PCA.

Room temperature data

- $N = 144$

- $K = 4 + 1$ column containing the date and time at which the 4 temperatures were recorded

- Web address: http://openmv.net/info/room-temperature

- Description: Temperature measurements from 4 corners of a room
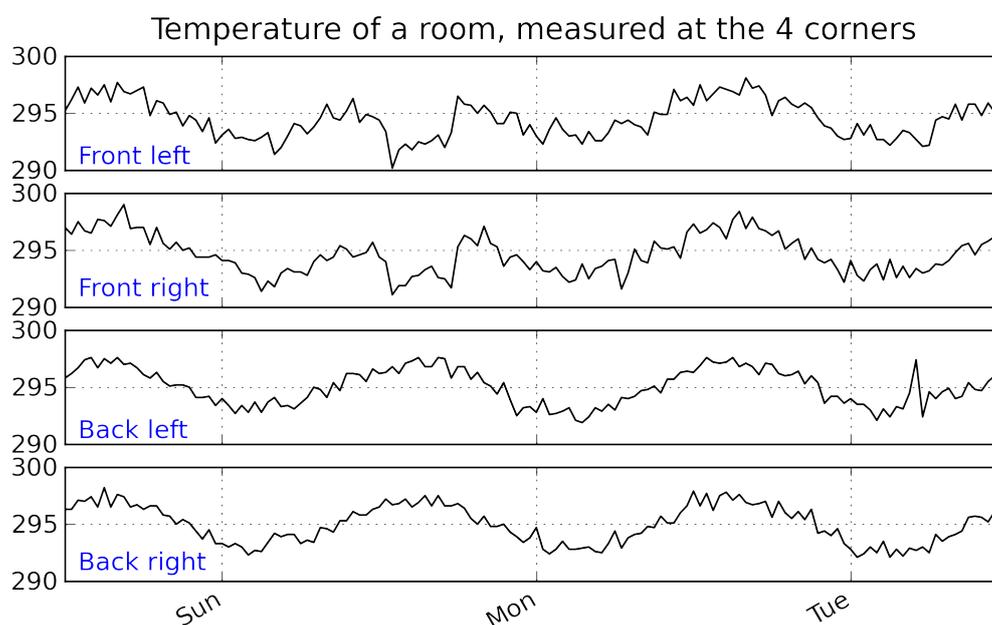
Objectives

Before even fitting the model:

1. How many latent variables do you expect to use in this model? Why?.

2. What do you expect the first loading vector to look like?

Now build a PCA model using any software package.

1. How much variation was explained by the first and second latent variables? Is this result surprising, *given the earlier description* (page 323) of the dataset?

2. Plot a time series plot (also called a line plot) of t_1. Did this match your expectations? Why/why not?

3. Plot a bar plot of the loadings for the second component. Given this bar plot, what are the characteristics of an observation with a large, positive value of t_2; and a large, negative t_2 value?

4. Now plot the time series plot for t_2. Again, does this plot match your expectations?

Now use the *concept of brushing* (page 365) to interrogate and learn from the model.

1. Plot a score plot of t_1 against t_2.

2. Also plot the time series plot of the raw data.

3. Select a cluster of interest in the score plot and see the brushed values in the raw data. Are these the values you expected to be highlighted?

4. Next plot the Hotelling's T^2 line plot, as *described earlier* (page 350). Does the 95% limit in the Hotelling's T^2 line plot correspond to the 95% limit in the score plot?

5. Also plot the SPE line plot. Brush the outlier in the SPE plot and find its location in the score plot.

6. Why does this point have a large SPE value?

7. Describe how a 3-D scatter plot would look with t_1 and t_2 as the (x, y) axes, and SPE as the z-axis.
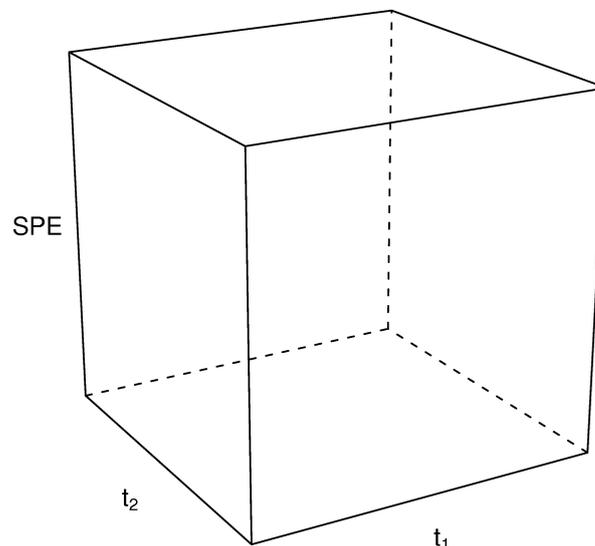
What have we learned?

- Interpreted that a latent variable is often a true driving force in the system under investigation.

- How to interpret a loadings vector and its corresponding score vector.

- Brushing multivariate and raw data plots to confirm our understanding of the model.

- Learned about Hotelling's T^2, whether we plot it as a line plot, or as an ellipse on a scatter plot.

- We have confirmed how the scores are on the model plane, and the SPE is the distance from the model plane to the actual observation.

Food texture data set

- $N = 50$

- $K = 5 + 1$ column containing the labels for each batch

- Web address: http://openmv.net/info/food-texture

- Description: Data from a *food manufacturer making a pastry product* (page 332). Each row contains the 5 quality attributes of a batch of product.

1. Fit a PCA model.

2. Report the R^2 values for the overall model and the R^2 values for each variable, on a per-component basis for components 1, 2, and 3. Comment on what each latent variable is explaining and by how much.

3. Plot the loadings plot as a bar plot for p_1. Does this match the values *given earlier* (page 332)? Interpret what kind of pastry would have a large positive t_1 value?

4. What feature(s) of the raw data does the second component explain? Plot sequence-ordered plots of the raw data to confirm your answer.

5. Look for any observations that are unusual. Are there any unusual scores? SPE values? Plot contribution plots for the unusual observations and interpret them.

Food consumption data set

This data set has become a classic data set when learning about multivariate data analysis. It consists of

- $N = 16$ countries in the European area

- $K = 20$ food items

- Missing data: yes

- Web address: http://openmv.net/info/food-consumption

- Description: The data table lists for each country the relative consumption of certain food items, such as tea, jam, coffee, yoghurt, and others.

	Real coffee	Instant coffee	Tea	Sweetener	Biscuits	Powder soup	Tin soup	Potato	Frozen fish	Frozen veggies	Apples	Oranges	Tinned fruit	Jam	Garlic	Butter	Margarine	Olive oil	Yoghurt	Crisp bread
Germany	90	49	88	19	57	51	19	21	27	21	81	75	44	71	22	91	85	74	30	26
Italy	82	10	60	2	55	41	3	2	4	2	67	71	9	46	80	66	24	94	5	18
France	88	42	63	4	76	53	11	23	11	5	87	84	40	45	88	94	47	36	57	3
Holland	96	62	98	32	62	67	43	7	14	14	83	89	61	81	15	31	97	13	53	15
Belgium	94	38	48	11	74	37	23	9	13	12	76	76	42	57	29	84	80	83	20	5
Luxembourg	97	61	86	28	79	73	12	7	26	23	85	94	83	20	91	94	94	84	31	24
England	27	86	99	22	91	55	76	17	20	24	76	68	89	91	11	95	94	57	11	28
Portugal	72	26	77	2	22	34	1	5	20	3	22	51	8	16	89	65	78	92	6	9
Austria	55	31	61	15	29	33	1	5	15	11	49	42	14	41	51	51	72	28	13	11
Switzerland	73	72	85	25	31	69	10	17	19	15	79	70	46	61	64	82	48	61	48	30
Sweden	97	13	93	31		43	43	39	54	45	56	78	53	75	9	68	32	48	2	93
Denmark	96	17	92	35	66	32	17	11	51	42	81	72	50	64	11	92	91	30	11	34
Norway	92	17	83	13	62	51	4	17	30	15	61	72	34	51	11	63	94	28	2	62
Finland	98	12	84	20	64	27	10	8	18	12	50	57	22	37	15	96	94	17		64
Spain	70	40	40		62	43	2	14	23	7	59	77	30	38	86	44	51	91	16	13
Ireland	30	52	99	11	80	75	18	2	5	3	57	52	46	89	5	97	25	31	3	9

1. Fit a PCA model to the data using 2 components.

2. Plot a loadings plot of p_1 against p_2. Which are the important variables in the first component? And the second component?

3. Since each column represents food consumption, how would you interpret a country with a high (positive or negative) t_1 value? Find countries that meet this criterion. Verify that this country does indeed have this interpretation (*hint*: use a contribution plot and examine the raw data in the table).

4. Now plot SPE after 2 components (don't plot the default SPE, make sure it is the SPE only after two components). Use a contribution plot to interpret any interesting outliers.

5. Now add a third component and plot SPE after 3 components. What has happened to the observations you identified in the previous question? Investigate the loadings plot for the third component now (as a bar plot) and see which variables are heavily loaded in the 3rd component.

6. Also plot the R^2 values for each variable, after two components, and after 3 components. Which variables are modelled by the 3rd component? Does this match with your interpretation of the loadings bar plot in the previous question?

7. Now plot a score plot of the 3rd component against the 1st component. Generate a contribution plot in the score from the interesting observation(s) you selected in part 4. Does this match up with your interpretation of what the 3rd component is modelling?

What we learned:

- Further practice of our skills in interpreting score plots and loading plots.

- How to relate contribution plots to the loadings and the R^2 values for a particular component.

Silicon wafer thickness

- $N = 184$

- $K = 9$

- Web address: http://openmv.net/info/silicon-wafer-thickness

- Description: These are nine thickness measurements recorded from various batches of silicon wafers. One wafer is removed from each batch and the thickness of the wafer is measured at the nine locations, as shown in the illustration.
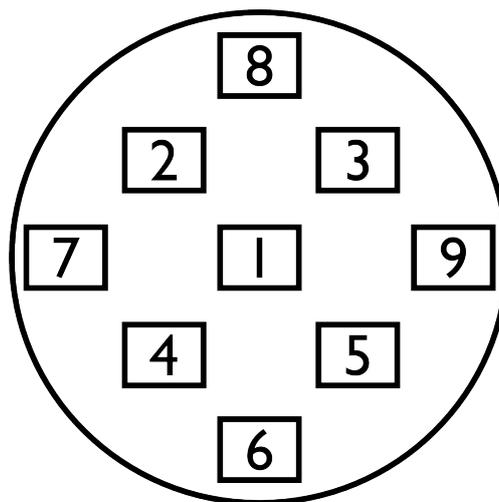
1. Build a PCA model on all the data.

2. Plot the scores for the first two components. What do you notice? Investigate the outliers, and the raw data for each of these unusual observations. What do you conclude about those observations?

3. Exclude the unusual observations and refit the model.

4. Now plot the scores plot again; do things look better? Record the R^2 and Q^2 values (from cross-validation) for the first three components. Are the R^2 and Q^2 values close to each other; what does this mean?

5. Plot a loadings plot for the first component. What is your interpretation of p_1? Given the R^2 and Q^2 values for this first component (previous question), what is your interpretation about the variability in this process?

6. And the interpretation of p_2? From a quality control perspective, if you could remove the variability due to p_2, how much of the variability would you be removing from the process?

7. Also plot the corresponding time series plot for t_1. What do you notice in the sequence of score values?

8. Repeat the above question for the second component.

9. Finally, plot both the t_1 and t_2 series overlaid on the same plot, in time-order, to see the smaller variance that t_2 explains.

What we learned:

• Identifying outliers; removing them and refitting the model.

• Variability in a process can very often be interpreted. The R^2 and Q^2 values for each component show which part of the variability in the system is due the particular phenomenon modelled by that component.

Process troubleshooting

Recent trends show that the yield of your company's flagship product is declining. You are uncertain if the supplier of a key raw material is to blame, or if it is due to a change in your process conditions. You begin by investigating the raw material supplier.

The data available has:

• $N = 24$

• $K = 6 + 1$ designation of process outcome

• Web address: http://openmv.net/info/raw-material-characterization

• Description: 3 of the 6 measurements are size values for the plastic pellets, while the other 3 are the outputs from thermogravimetric analysis (TGA), differential scanning calorimetry (DSC) and thermomechanical analysis (TMA), measured in a laboratory. These 6 measurements are thought to adequately characterize the raw material. Also provided is a designation `Adequate` or `Poor` that reflects the process engineer's opinion of the yield from that lot of materials.

Import the data, and set the `Outcome` variable as a secondary identifier for each observation, as shown in the illustration below. The observation's primary identifier is its batch number.

Lot number	Outcome	Size5	Size10	Size15	TGA	DSC	TMA
B370	Adequate	13.8	9.2	41.2	787.3	18.0	65.0
B880	Adequate	11.2	5.8	27.6	772.2	17.7	68.8
B452	Adequate	9.9	5.8	28.3	602.3	18.3	50.7
B287	Adequate	10.4	4.0	24.7	677.9	17.7	56.5
B576	Adequate	12.3	9.3	22.0	593.5	19.5	52.0
B914	Poor	13.7	7.8	27.0	597.9	18.1	49.8
B404	Poor	15.5	10.7	34.3	668.5	19.6	55.7
B694	Poor	15.4	10.7	35.9	602.8	19.2	53.6
B875	Poor	14.9	11.3	41.0	614.6	18.5	50.0
B475	Adequate	13.7	8.5	28.0	700.4	18.0	57.0
B517	Poor	16.1	11.6	39.2	682.8	17.5	56.4
B296	Adequate	12.8	5.4	23.7	739.4	18.2	59.8

1. Build a latent variable model for all observations and use auto-fit to determine the number of components. If your software does not have and auto-fit features (cross-validation), then use a Pareto plot of the eigenvalues to decide on the number of components.

2. Interpret component 1, 2 and 3 separately (using the loadings bar plot).

3. Now plot the score plot for components 1 and 2, and colour code the score plot with the `Outcome` variable. Interpret why observations with `Poor` outcome are at their locations in the score plot (use a contribution plot).

4. What would be your recommendations to your manager to get more of your batches classified as `Adequate` rather than `Poor`?

5. Now build a model only on the observations marked as `Adequate` in the Outcome variable.

6. Re-interpret the loadings plot for p_1 and p_2. Is there a substantial difference between this new loadings plot and the previous one?

What we learned:

- How to use an indicator variable in the model to learn more from our score plot.

- How to build a data set, and bring in new observations as testing data.

6.6 Principal Component Regression (PCR)

Principal component regression (PCR) is an alternative to multiple linear regression (MLR) and has many advantages over MLR.

In *multiple linear regression* (page 185) we have two matrices (blocks): $\mathbf{X}$, an $N \times K$ matrix whose columns we relate to the single vector, $\mathbf{y}$, an $N \times 1$ vector, using a model of the form: $\mathbf{y} = \mathbf{X}\mathbf{b}$. The solution vector $\mathbf{b}$ is found by solving $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. The variance of the estimated solution is given by $\mathcal{V}(\mathbf{b}) = (\mathbf{X}'\mathbf{X})^{-1} S_E^2$.

In the section on *factorial experiments* (page 243) we intentionally set our process to generate a matrix $\mathbf{X}$ that has independent columns. This means that each column is orthogonal to the others, you cannot express one column in terms of the other, and it results in a diagonal $\mathbf{X}'\mathbf{X}$ matrix.

On most data sets though the columns in $\mathbf{X}$ are correlated. Correlated columns are not too serious if they are mildly correlated. But the illustration here shows the problem with strongly correlated variables, in this example x_1 and x_2 are strongly, positively correlated. Both variables are used to create a predictive model for y. The model plane, $\hat{y} = b_0 + b_1 x_1 + b_2 x_2$ is found so that it minimizes the residual error. There is a unique minimum for the sum of squares of the residual error (i.e. the objective function). But very small changes in the raw x-data lead to almost no change in the objective function, but will show large fluctuations in the solution for $\mathbf{b}$ as the plane rotates around the axis of correlation. This can be visualized in this illustration.
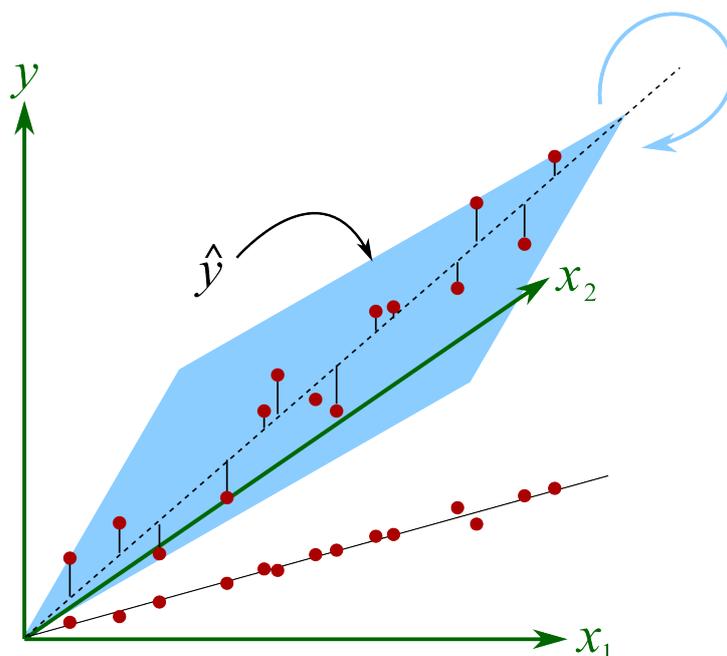
The plane will rotate around the axial, dashed line if we make small changes in the raw data. At each new rotation we will get very different values of b_1 and b_2, even changing in sign(!), but the objective function's minimum value does not change very much. This phenomena shows up in the least squares solution as wide confidence intervals for the coefficients, since the off-diagonal elements in $\mathbf{X}'\mathbf{X}$ will be large. This has important consequences when you are trying to learn about your process from this model: you have to use caution. A model with low or uncorrelated variables is well-supported by the data, and cannot be arbitrarily rotated.

The common "solution" to this problem of collinearity is to revert to variable selection. In the above example the modeller would select either x_1 or x_2. In general, the modeller must select a subset of uncorrelated columns from the K columns in $\mathbf{X}$ rather than using the full matrix. When K is large, then this becomes a large computational burden. Further, it is not clear what the trade-offs are, and how many columns should be in the subset. When is a correlation too large to be problematic?

We face another problem with MLR: the assumption that the variables in $\mathbf{X}$ are measured without error, which we know to be untrue in many practical engineering situations and is exactly what leads to the instability of the rotating plane. Furthermore, MLR cannot handle missing data. To summarize, the shortcomings of multiple linear regression are that:

- it cannot handle strongly correlated columns in $\mathbf{X}$

- it assumes $\mathbf{X}$ is noise-free, which it almost never is in practice

- cannot handle missing values in $\mathbf{X}$

- MLR requires that $N > K$, which can be impractical in many circumstances, which leads to

- variable selection to meet the $N > K$ requirement, and to gain independence between columns of $\mathbf{X}$, but that selection process is non-obvious, and may lead to suboptimal predictions.

The main idea with principal component regression is to replace the K columns in $\mathbf{X}$ with their uncorrelated A score vectors from PCA.

Multiple linear regression   Principal component regression
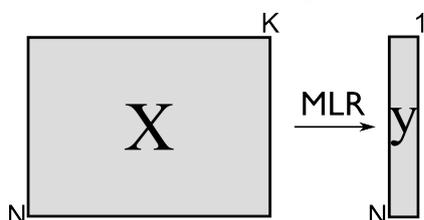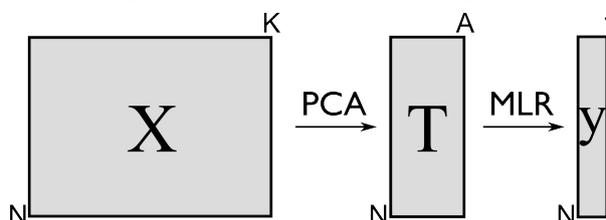
In other words, we replace the $N \times K$ matrix of raw data with a smaller $N \times A$ matrix of data that summarizes the original $\mathbf{X}$ matrix. Then we relate these A scores to the y variable. Mathematically it is a two-step process:

1. $\mathbf{T} = \mathbf{XP}$ from the PCA model

2. $\widehat{\mathbf{y}} = \mathbf{Tb}$ and can be solved as $\mathbf{b} = (\mathbf{T'T})^{-1}\mathbf{T'y}$

This has a number of advantages:

1. The columns in $\mathbf{T}$, the scores from PCA, are orthogonal to each other, obtaining independence for the least-squares step.

2. These $\mathbf{T}$ scores can be calculated even if there are missing data in $\mathbf{X}$.

3. We have reduced the assumption of errors in $\mathbf{X}$, since $\widehat{\mathbf{X}} = \mathbf{TP'} + \mathbf{E}$. We have replaced it with the assumption that there is no error in $\mathbf{T}$, a more realistic assumption, since PCA separates the noise from the systematic variation in $\mathbf{X}$. The $\mathbf{T}$'s are expected to have much less noise than the $\mathbf{X}$'s.

4. The relationship of each score column in $\mathbf{T}$ to vector y can be interpreted independently of each other.

5. Using MLR requires that $N > K$, but with PCR this changes to $N > A$; an assumption that is more easily met for short and wide $\mathbf{X}$ matrices with many correlated columns.

6. There is much less need to resort to selecting variables from $\mathbf{X}$; the general approach is to use the entire $\mathbf{X}$ matrix to fit the PCA model. We actually use the correlated columns in $\mathbf{X}$ to stabilize the PCA solution, much in the same way that extra data improves the estimate of a mean (recall the central limit theorem).

7. But by far one of the greatest advantages of PCR though is the free consistency check that one gets on the raw data, which you don't have for MLR. Always check the SPE and Hotelling's T^2 value for a new observation during the first step. If SPE is close to the model plane, and T^2 is within the range of the previous T^2 values, then the prediction from the second step should be reasonable.

Illustrated as follows we see the misleading strategy that is regularly seen with MLR. The modeller has build a least squares model relating x_1 and x_2 to y, over the given ranges of x. The closed circles represent the actual data, while the open circles are the projections of the x_1 and x_2 values on the $x_1 - x_2$ plane. The predictive model works adequately.

Actual data and the linear model built from it

New point

x_2

$+3$

-2

y

-5

$+6$

x_1
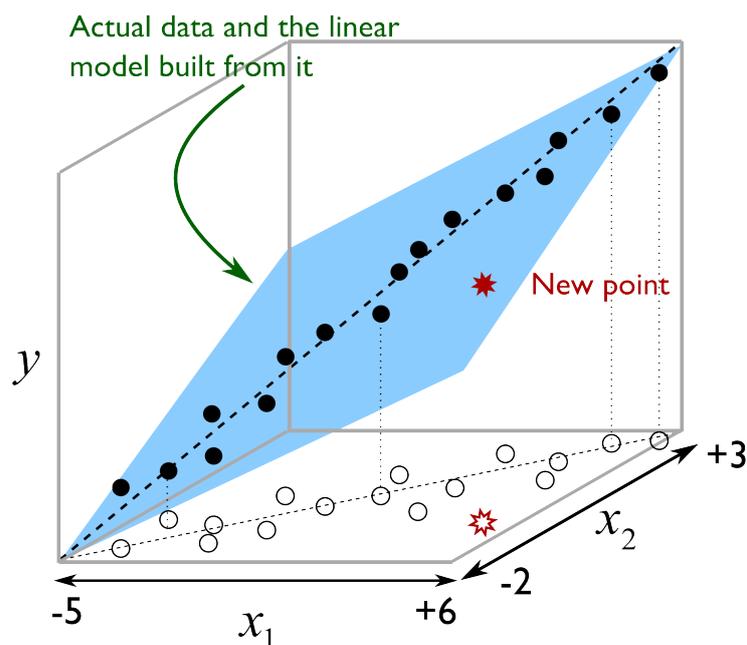
But the misleading strategy often used by engineers is to say that the model is valid as long as $-5 \leq x_1 \leq +6$ and $-2 \leq x_2 \leq +1$. If the engineer wants to use the model at the points marked with *, the results will be uncertain, even though those marked points obey the given constraints. The problem is that the engineer has not taken the correlation between the variables into account. With PCR we would immediately detect this: the points marked as * would have large SPE values from the PCA step, indicating they are not consistent with the model.

Here then is the procedure for **building** a principal component regression model.

1. Collect the $\mathbf{X}$ and y data required for the model.

2. Build a PCA model on the data in $\mathbf{X}$, fitting A components. We usually set A by cross-validation, but often components beyond this will be useful. Iterate back to this point after the initial model to assess if A should be changed.

3. Examine the SPE and T^2 plots from the PCA model to ensure the model is not biased by unusual outliers.

4. Use the columns in $\mathbf{T}$ from PCA as your data source for the usual multiple linear regression model (i.e. they are now the $\mathbf{X}$-variables in an MLR model).

5. Solve for the MLR model parameters, $\mathbf{b} = (\mathbf{T}'\mathbf{T})^{-1} \mathbf{T}'\mathbf{y}$, an $A \times 1$ vector, with each coefficient entry in $\mathbf{b}$ corresponding to each score.

Using the principal component regression model for a new observation:

1. Obtain your vector of new data, $\mathbf{x}'_{\text{new, raw}}$, a $1 \times K$ vector.

2. Preprocess this vector in the same way that was done when building the PCA model (usually just mean centering and scaling) to obtain $\mathbf{x}'_{\text{new}}$

3. Calculate the scores for this new observation: $\mathbf{t}'_{\text{new}} = \mathbf{x}'_{\text{new}}\mathbf{P}$.

4. Find the predicted value of this observation: $\widehat{\mathbf{x}}'_{\text{new}} = \mathbf{t}'_{\text{new}}\mathbf{P}'$.

5. Calculate the residual vector: $\mathbf{e}'_{\text{new}} = \mathbf{x}'_{\text{new}} - \widehat{\mathbf{x}}'_{\text{new}}$.

6. Then compute the residual distance from the model plane: $\text{SPE}_{\text{new}} = \sqrt{\mathbf{e}'_{\text{new}}\mathbf{e}_{\text{new}}}$

7. And the Hotelling's T^2 value for the new observation: $T_{\text{new}}^2 = \sum_{a=1}^{a=A} \left(\dfrac{t_{\text{new},a}}{s_a} \right)^2$.

8. Before calculating the prediction from the PCR model, first check if the SPE_{new} and T_{new}^2 values are below their 95% or 99% limits. If the new observation is below these limits, then go on to calculate the prediction: $\widehat{y}_{\text{new}} = \mathbf{t}_{\text{new}}'\mathbf{b}$, where $\mathbf{b}$ was from the

9. If either of the SPE or T^2 limits were exceeded, then one should investigate the contributions to SPE, T^2 or the individuals scores to see why the new observation is unusual.

   Predictions of $\widehat{y}_{\text{new}}$ when a point is above either limit, especially the SPE limit, are not to be trusted.

Multiple linear regression, though relatively simpler to implement, has no such consistency check on the new observation's x-values. It simply calculates a direct prediction for $\widehat{y}_{\text{new}}$, no matter what the values are in $\mathbf{x}_{\text{new}}$.

One of the main applications in engineering for PCR is in the use of software sensors, also called *inferential sensors* (page 396). The method of PLS has some distinct advantages over PCR, so we prefer to use that method instead, as described next.

6.7 Introduction to Projection to Latent Structures (PLS)

Projection to Latent Structures (PLS) is the first step we will take to extending latent variable methods to using more than one block of data. In the PLS method we divide our variables (columns) into two blocks: called $\mathbf{X}$ and $\mathbf{Y}$.

Learning how to choose which variables go in each block will become apparent later, but for now you may use the rule of thumb that says $\mathbf{X}$ takes the variables which are always available when using the model, while $\mathbf{Y}$ takes the variables that are *not always available*. Both $\mathbf{X}$ and $\mathbf{Y}$ must be available when building the model, but later, when using the model, only $\mathbf{X}$ is required. As you can guess, one of the major uses of PLS is for predicting variables in $\mathbf{Y}$ using variables in $\mathbf{X}$, but this is not its only purpose as a model. It is a very good model for process understanding and troubleshooting.

PLS can be used for process monitoring and for optimizing the performance of a process. It is also widely used for new product development, or for improving existing products. In all these cases the $\mathbf{Y}$ block most often contains the outcome, or quality properties.

However, PLS is most commonly used for prediction. And this is also a good way to introduce PLS. In (chemical) engineering processes we use it to develop software sensors (also known as inferential sensors) that predict time-consuming lab measurement in real-time, using the on-line data from our processes. In laboratories we use spectral data (e.g. NIR spectra) to predict the composition of a liquid; this is known as the calibration problem; once calibrated with samples of known composition we can predict the composition of future samples.

But why use the PLS method at all?

6.7.1 Advantages of the projection to latent structures (PLS) method

So for predictive uses, a PLS model is very similar to *principal component regression* (page 371) (PCR) models. And PCR models were a big improvement over using multiple linear regression (MLR). In brief, *PCR was shown to have these advantages* (page 373):

- It handles the correlation among variables in $\mathbf{X}$ by building a PCA model first, then using those orthogonal scores, $\mathbf{T}$, instead of $\mathbf{X}$ in an ordinary multiple linear regression. This prevents us from having to resort to variable selection.

- It extracts these scores $\mathbf{T}$ even if there are missing values in $\mathbf{X}$.

- We reduce, but don't remove, the severity of the assumption in MLR that the predictor's, $\mathbf{T}$ in this case, are noise-free. This is because the PCA scores are less noisy than the raw data $\mathbf{X}$.

- With MLR we require that $N > K$ (number of observations is greater than the number of variables), but with PCR this is reduced to $N > A$, and since $A \ll K$ this requirement is often true, especially for spectral data sets.

- We get the great benefit of a consistency check on the raw data, using SPE and T^2 from PCA, before moving to the second prediction step.

An important point is that PCR is a two-step process:

Multiple linear regression   Principal component regression
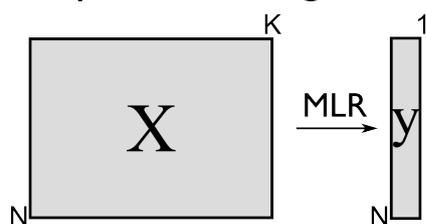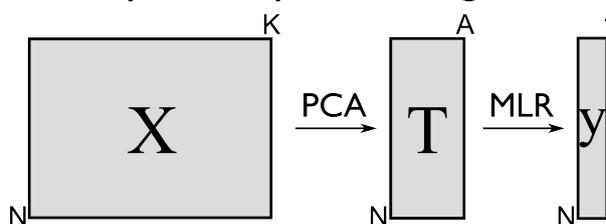
In other words, we replace the $N \times K$ matrix of raw data with a smaller $N \times A$ matrix of data that summarizes the original $\mathbf{X}$ matrix. Then we relate these scores to the y variable. Mathematically it is a two-step process:

1. $\mathbf{T} = \mathbf{XP}$

2. $\widehat{\mathbf{y}} = \mathbf{Tb}$ and can be solved as $\mathbf{b} = \left(\mathbf{T}'\mathbf{T}\right)^{-1}\mathbf{T}'\mathbf{y}$

The PLS model goes a bit further and introduces some additional advantages over PCR:

- A single PLS model can be built for multiple, correlated $\mathbf{Y}$ variables. This eliminates having to build M PCR models, one for each column in $\mathbf{Y}$.

- The PLS model directly assumes that there is error in $\mathbf{X}$ and $\mathbf{Y}$. We will return to this important point of an $\mathbf{X}$-space model later on.

- PLS is more efficient than PCR in two ways: with PCR, one or more of the score columns in $\mathbf{T}$ may only have a small correlation with $\mathbf{Y}$, so these scores are needlessly calculated. Or as is more common, we have to extract many PCA components, going beyond the level of what would normally be calculated (essentially over fitting the PCA model), in order to capture sufficient predictive columns in $\mathbf{T}$. This augments the size of the PCR model, and makes interpretation harder, which is already strained by the two-step modelling required for PCR.

Similar to PCA, the basis for PCR, we have that PLS also extracts sequential components, but it does so using the data in both $\mathbf{X}$ *and* $\mathbf{Y}$. So it can be seen to be very similar to PCR, but that it calculates the

model in one go. From the last point just mentioned, it is not surprising that PLS often requires fewer components than PCR to achieve the same level of prediction. In fact when compared to several regression methods, MLR, ridge regression and PCR, a PLS model is often the most "compact" model.

We will get into the details shortly, but as a starting approximation, you can visualize PLS as a method that extracts a single set of scores, $\mathbf{T}$, from both $\mathbf{X}$ and $\mathbf{Y}$ simultaneously.
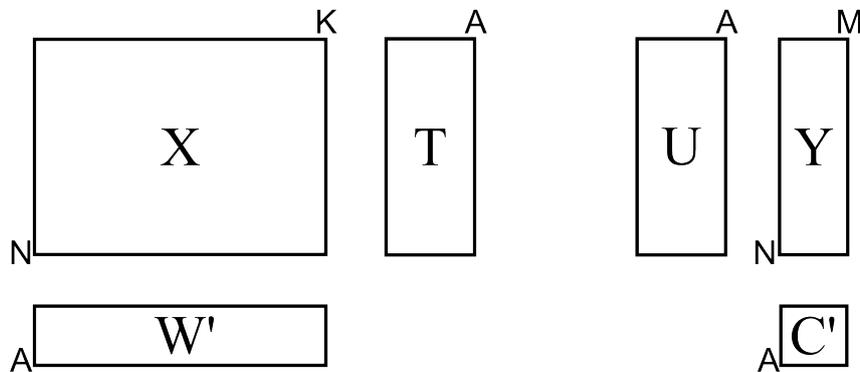
From an engineering point of view this is quite a satisfying interpretation. After all, the variables we chose to be in $\mathbf{X}$ and in $\mathbf{Y}$ come from the same system. That system is driven (moved around) by the *same underlying latent variables*.

6.7.2 A conceptual explanation of PLS

Now that you are comfortable with the concept of a latent variable using PCA and PCR, you can interpret PLS as a latent variable model, but one that has a different objective function. In PCA the objective function was to calculate each latent variable so that it best explains the available variance in $\mathbf{X}_a$, where the subscript A refers to the matrix $\mathbf{X}$ *before* extracting the ath component.

In PLS, we also find these latent variables, but we find them so they best explain $\mathbf{X}_a$ and best explain $\mathbf{Y}_a$, and so that these latent variables have the strongest possible relationship between $\mathbf{X}_a$ and $\mathbf{Y}_a$.

In other words, there are three simultaneous objectives with PLS:

1. The best explanation of the $\mathbf{X}$-space.
2. The best explanation of the $\mathbf{Y}$-space.
3. The greatest relationship between the $\mathbf{X}$- and $\mathbf{Y}$-space.

6.7.3 A mathematical/statistical interpretation of PLS

We will get back to the *mathematical details later on* (page 382), but we will consider our conceptual explanation above in terms of mathematical symbols.

In PCA, the objective was to best explain $\mathbf{X}_a$. To do this we calculated scores, $\mathbf{T}$, and loadings $\mathbf{P}$, so that each component, $\mathbf{t}_a$, had the greatest variance, while keeping the loading direction, $\mathbf{p}_a$, constrained to a unit vector.

$$\max : \mathbf{t}'_a \mathbf{t}_a \qquad \text{subject to} \quad \mathbf{p}'_a \mathbf{p}_a = 1.0$$

The above was shown to be a concise mathematical way to state that these scores and loadings best explain $\mathbf{X}$; no other loading direction will have greater variance of $\mathbf{t}'_a$. (The scores have mean of zero, so their variance is proportional to $\mathbf{t}'_a \mathbf{t}_a$).

For PCA, for the ath component, we can calculate the scores as follows (we are projecting the values in $\mathbf{X}_a$ onto the loading direction $\mathbf{p}_a$):

$$\mathbf{t}_a = \mathbf{X}_a \mathbf{p}_a$$

Now let's look at PLS. Earlier we said that PLS extracts a single set of scores, $\mathbf{T}$, from $\mathbf{X}$ and $\mathbf{Y}$ simultaneously. That wasn't quite true, but it is still an accurate statement! PLS actually extracts two sets of scores, one set for $\mathbf{X}$ and another set for $\mathbf{Y}$. We write these scores for each space as:

$$\mathbf{t}_a = \mathbf{X}_a \mathbf{w}_a \qquad \text{for the } \mathbf{X}\text{-space}$$
$$\mathbf{u}_a = \mathbf{Y}_a \mathbf{c}_a \qquad \text{for the } \mathbf{Y}\text{-space}$$

The objective of PLS is to extract these scores so that they have *maximal covariance.* Let's take a look at this. *Covariance was shown* (page 152) to be:

$$\text{Cov}\,(\mathbf{t}_a, \mathbf{u}_a) = \mathcal{E}\left\{(\mathbf{t}_a - \bar{\mathbf{t}}_a)(\mathbf{u}_a - \bar{\mathbf{u}}_a)\right\}$$

Using the fact that these scores have mean of zero, the covariance is proportional (with a constant scaling factor of N) to $\mathbf{t}'_a \mathbf{u}_a$. So in summary, each component in PLS is maximizing that covariance, or the dot product: $\mathbf{t}'_a \mathbf{u}_a$.

Now covariance is a hard number to interpret; about all we can say with a covariance number is that the larger it is, the greater the relationship, or *correlation*, between two vectors. So it is actually more informative to rewrite covariance in terms of *correlations* (page 155) and variances:

$$\text{Cov}\,(\mathbf{t}_a, \mathbf{u}_a) = \text{Correlation}\,(\mathbf{t}_a, \mathbf{u}_a) \times \sqrt{\text{Var}\,(\mathbf{t}_a)} \times \sqrt{\text{Var}\,(\mathbf{u}_a)}$$
$$\text{Cov}\,(\mathbf{t}_a, \mathbf{u}_a) = \text{Correlation}\,(\mathbf{t}_a, \mathbf{u}_a) \times \sqrt{\mathbf{t}'_a \mathbf{t}_a} \times \sqrt{\mathbf{u}'_a \mathbf{u}_a}$$

As this shows then, maximizing the covariance between $\mathbf{t}'_a$ and $\mathbf{u}_a$ is actually maximizing the 3 simultaneous objectives mentioned earlier:

1. The best explanation of the $\mathbf{X}$-space: given by $\mathbf{t}'_a \mathbf{t}_a$

2. The best explanation of the $\mathbf{Y}$-space. given by $\mathbf{u}'_a \mathbf{u}_a$

3. The greatest relationship between the $\mathbf{X}$- and $\mathbf{Y}$-space: given by correlation $(\mathbf{t}_a, \mathbf{u}_a)$

These scores, $\mathbf{t}'_a$ and $\mathbf{u}_a$, are found subject to the constraints that $\mathbf{w}'_\mathbf{a} \mathbf{w}_\mathbf{a} = 1.0$ and $\mathbf{c}'_\mathbf{a} \mathbf{c}_\mathbf{a} = 1.0$. This is similar to PCA, where the loadings $\mathbf{p}_a$ were constrained to unit length. In PLS we constrain the loadings for $\mathbf{X}$, called $\mathbf{w}_a$, and the loadings for $\mathbf{Y}$, called $\mathbf{c}_a$, to unit length.

The above is a description of one variant of PLS, known as SIMPLS[158] (simple PLS).

---

[158] https://dx.doi.org/10.1016/0169-7439(93)85002-X

6.7.4 A geometric interpretation of PLS

As we did with PCA (page 327), let's take a geometric look at the PLS model space. In the illustration below we happen to have $K = 3$ variables in $\mathbf{X}$, and $M = 3$ variables in $\mathbf{Y}$. (In general $K \neq M$, but $K = M = 3$ make explanation in the figures easier.) Once the data are centered and scaled we have just shifted our coordinate system to the origin. Notice that there is one dot in $\mathbf{X}$ for each dot in $\mathbf{Y}$. Each dot represents a row from the corresponding $\mathbf{X}$ and $\mathbf{Y}$ matrix.
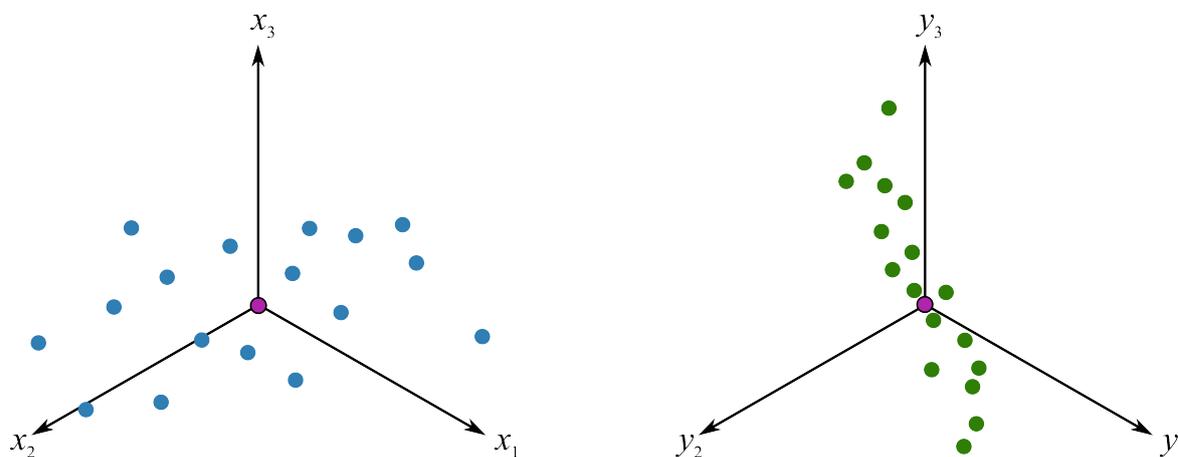
We assume here that you understand how the scores are the perpendicular projection of each data point onto each direction vector (if not, please review the *relevant section* (page 327) in the PCA notes). In PLS though, the direction vectors, $\mathbf{w}_1$ and $\mathbf{c}_1$, are found and each observation is projected onto the direction. The point at which each observation lands is called the $\mathbf{X}$-space score, t_i, or the $\mathbf{Y}$-space score, u_i. These scores are found so that the covariance between the t-values and u-values is maximized.
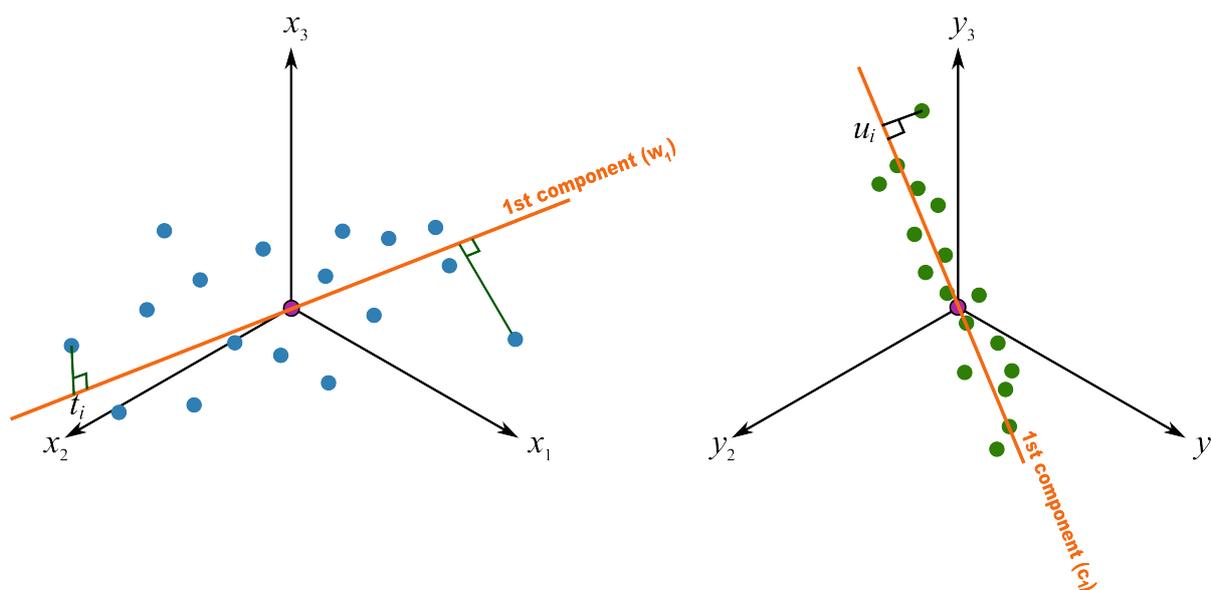
As *explained above* (page 377), this means that the latent variable directions are oriented so that they best explain $\mathbf{X}$, and best explain $\mathbf{Y}$, and have the greatest possible relationship between $\mathbf{X}$ and $\mathbf{Y}$.

The second component is then found so that it is orthogonal to the first component in the $\mathbf{X}$ space (the

second component is not necessarily orthogonal in the **Y**-space, though it often is close to orthogonal).
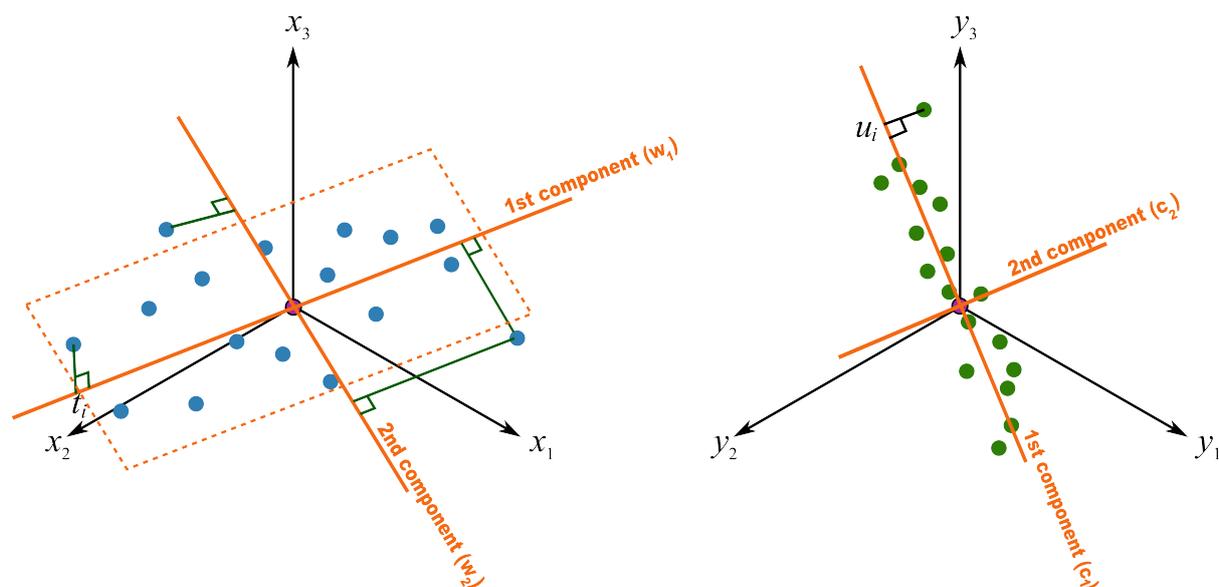
6.7.5 Interpreting the scores in PLS

Like in PCA, our scores in PLS are a summary of the data from *both* blocks. The reason for saying that, even though there are two sets of scores, **T** and **U**, for each of **X** and **Y** respectively, is that they have maximal covariance. We can interpret one set of them. In this regard, the **T** scores are more readily interpretable, since they are always available. The **U** scores are not available until **Y** is known. We have the **U** scores during model-building, but when we use the model on new data (e.g. when making predictions using PLS), then we only have the **T** scores.

The scores for PLS are interpreted in exactly the *same way as for PCA* (page 336). Particularly, we look for clusters, outliers and interesting patterns in the line plots of the scores.

The only difference that must be remembered is that these scores have a different orientation to the PCA scores. As illustrated below, the PCA scores are found so that they only explain the variance in **X**; the PLS scores are calculated so that they also explain **Y** and have a maximum relationship between **X** and **Y**. Most time these directions will be close together, but not identical.
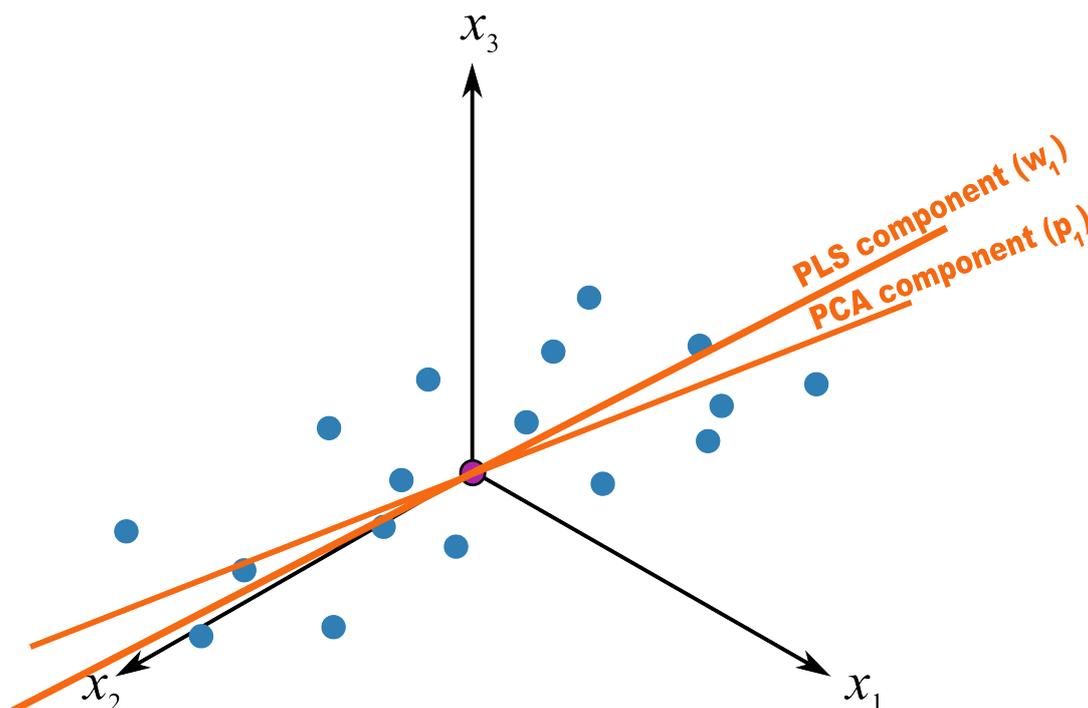
6.7.6 Interpreting the loadings in PLS

Like with the loadings from PCA (page 339), $\mathbf{p}_a$, we interpret the loadings $\mathbf{w}_a$ from PLS in the same way. Highly correlated variables have similar weights in the loading vectors and appear close together in the loading plots of all dimensions.

We tend to refer to the PLS loadings, $\mathbf{w}_a$, as weights; this is for reasons that will be explained soon.

There are two important differences though when plotting the weights. The first is that we superimpose the loadings plots for the $\mathbf{X}$ and $\mathbf{Y}$ space simultaneously. This is very powerful, because we not only see the relationship between the $\mathbf{X}$ variables (from the $\mathbf{w}$ vectors), we also see the relationship between the $\mathbf{Y}$ variables (from the $\mathbf{c}$ vectors), and even more usefully, the relationship between all these variables.

This agrees again with our (engineering) intuition that the $\mathbf{X}$ and $\mathbf{Y}$ variables are from the same system; they have been, somewhat arbitrarily, put into different blocks. The variables in $\mathbf{Y}$ could just have easily been in $\mathbf{X}$, but they are usually not available due to time delays, expense of measuring them frequently, *etc*. So it makes sense to consider the $\mathbf{w}_a$ and $\mathbf{c}_a$ weights simultaneously.

The second important difference is that we don't actually look at the $\mathbf{w}$ vectors directly, we consider rather what is called the $\mathbf{r}$ vector, though much of the literature refers to it as the $\mathbf{w}*$ vector (w-star). The reason for the change of notation from existing literature is that $\mathbf{w}*$ is confusingly similar to the multiplication operator (e.g. $\mathbf{w} * \mathbf{c}$: is frequently confused by newcomers, whereas $\mathbf{r} : \mathbf{c}$ would be cleaner). The $\mathbf{w}*$ notation gets especially messy when adding other superscript and subscript elements to it. Further, some of the newer literature on PLS, particularly SIMPLS, uses the $\mathbf{r}$ notation.

The $\mathbf{r}$ vectors show the effect of each of the original variables, in undeflated form, rather that using the $\mathbf{w}$ vectors which are the deflated vectors. This is explained next.

6.7.7 How the PLS model is calculated

This section assumes that you are comfortable with the *NIPALS algorithm for calculating a PCA model* (page 356) from $\mathbf{X}$. The NIPALS algorithm proceeds in exactly the same way for PLS, except we iterate through both blocks of $\mathbf{X}$ and $\mathbf{Y}$.
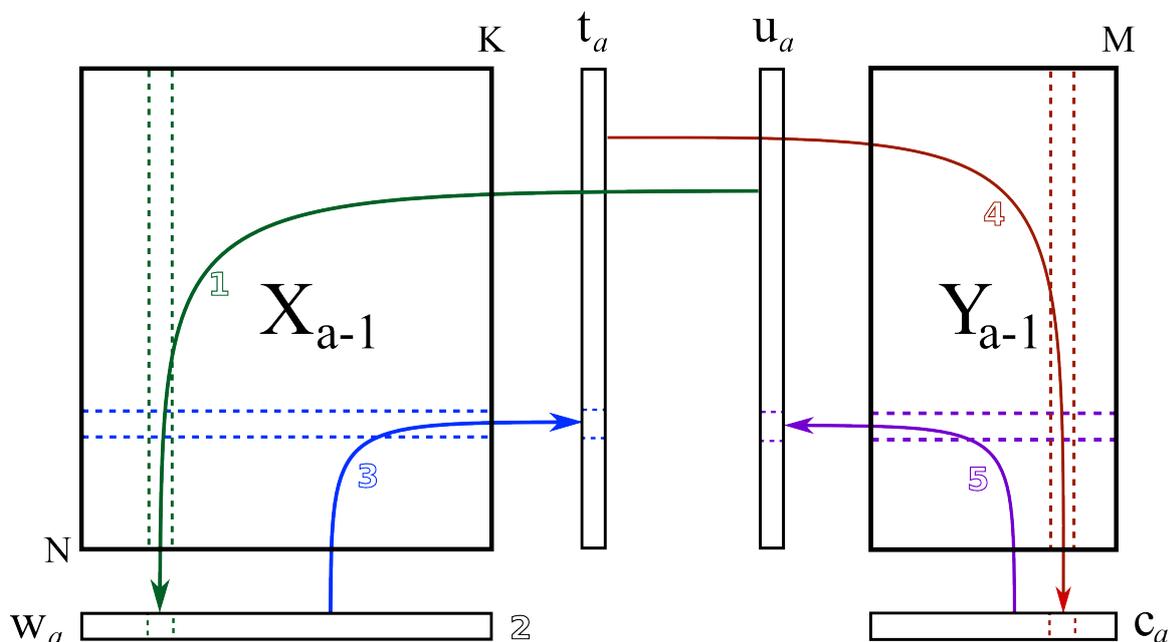
The algorithm starts by selecting a column from $\mathbf{Y}_a$ as our initial estimate for $\mathbf{u}_a$. The $\mathbf{X}_a$ and $\mathbf{Y}_a$ matrices are just the preprocessed version of the raw data when $a = 1$.

Arrow 1

Perform K regressions, regressing each column from $\mathbf{X}_a$ onto the vector $\mathbf{u}_a$. The slope coefficients from the regressions are stored as the entries in $\mathbf{w}_a$. Columns in $\mathbf{X}_a$ which are strongly correlated with $\mathbf{u}_a$ will have large weights in $\mathbf{w}_a$, while unrelated columns will have small, close to zero, weights. We can perform these regression in one go:

$$\mathbf{w}_a = \frac{1}{\mathbf{u}_a' \mathbf{u}_a} \cdot \mathbf{X}_a' \mathbf{u}_a$$

Step 2

Normalize the weight vector to unit length: $\mathbf{w}_a = \dfrac{\mathbf{w}_a}{\sqrt{\mathbf{w}_a' \mathbf{w}_a}}$.

Arrow 3

Regress every row in $\mathbf{X}_a$ onto the weight vector. The slope coefficients are stored as entries in $\mathbf{t}_a$. This means that rows in $\mathbf{X}_a$ that have a similar pattern to that described by the weight vector will have large values in $\mathbf{t}_a$. Observations that are totally different to $\mathbf{w}_a$ will have near-zero score values. These N regressions can be performed in one go:

$$\mathbf{t}_a = \frac{1}{\mathbf{w}_a' \mathbf{w}_a} \cdot \mathbf{X}_a \mathbf{w}_a$$

Arrow 4

Next, regress every column in $\mathbf{Y}_a$ onto this score vector, $\mathbf{t}_a$. The slope coefficients are stored in $\mathbf{c}_a$. We can calculate all M slope coefficients:

$$\mathbf{c}_a = \frac{1}{\mathbf{t}_a' \mathbf{t}_a} \cdot \mathbf{Y}_a' \mathbf{t}_a$$

Arrow 5

Finally, regress each of the N rows in $\mathbf{Y}_a$ onto this weight vector, $\mathbf{c}_a$. Observations in $\mathbf{Y}_a$ that are strongly related to $\mathbf{c}_a$ will have large positive or negative slope coefficients in vector $\mathbf{u}_a$:

$$\mathbf{u}_a = \frac{1}{\mathbf{c}_a' \mathbf{c}_a} \cdot \mathbf{Y}_a \mathbf{c}_a$$

This is one round of the NIPALS algorithm. We iterate through these 4 arrow steps until the $\mathbf{u}_a$ vector does not change much. On convergence, we store these 4 vectors: $\mathbf{w}_a, \mathbf{t}_a, \mathbf{c}_a$, and $\mathbf{u}_a$, which jointly define the ath component.

Then we deflate. Deflation removes variability already explained from $\mathbf{X}_a$ and $\mathbf{Y}_a$. Deflation proceeds as follows:

Step 1: Calculate a loadings vector for the X space

We calculate the loadings for the $\mathbf{X}$ space, called $\mathbf{p}_a$, using the $\mathbf{X}$-space scores:
$\mathbf{p}_a = \frac{1}{\mathbf{t}_a' \mathbf{t}_a} \cdot \mathbf{X}_a' \mathbf{t}_a$. This loading vector contains the regression slope of every column in $\mathbf{X}_a$ onto the scores, $\mathbf{t}_a$. In this regression the x-variable is the score vector, and the y variable is the column from $\mathbf{X}_a$. If we want to use this regression model in the usual least squares way, we would need a score vector (our x-variable) and predict the column from $\mathbf{X}_a$ as our y-variable.

If this is your first time reading through the notes, you should probably skip ahead to the next step in deflation. Come back to this section after reading about how to use a PLS model on new data, then it will make more sense.

Because it is a regression, it means that if we have a vector of scores, $\mathbf{t}_a$, in the future, we can predict each column in $\mathbf{X}_a$ using the corresponding slope coefficient in $\mathbf{p}_a$. So for the kth column, our prediction of column $\mathbf{X}_k$ is the product of the slope coefficient, $p_{k,a}$, and the score vector, $\mathbf{t}_a$. Or, we can simply predict the entire matrix in one operation: $\widehat{\mathbf{X}} = \mathbf{t}_a \mathbf{p}_a'$.

Notice that the loading vector $\mathbf{p}_a$ was calculated *after* convergence of the 4-arrow steps. In other words, these regression coefficients in $\mathbf{p}_a$ are not really part of the PLS model, they are merely calculated to later predict the values in the $\mathbf{X}$-space. But why can't we use the $\mathbf{w}_a$ vectors to predict the $\mathbf{X}_a$ matrix? Because after all, in arrow step 1 we were regressing columns of $\mathbf{X}_a$ onto $\mathbf{u}_a$ in order to calculate regression coefficients $\mathbf{w}_a$. That would imply that a good prediction of $\mathbf{X}_a$ would be $\widehat{\mathbf{X}}_a = \mathbf{u}_a \mathbf{w}_a'$.

That would require us to know the scores $\mathbf{u}_a$. How can we calculate these? We get them from $\mathbf{u}_a = \frac{1}{\mathbf{c}_a' \mathbf{c}_a} \cdot \mathbf{Y}_a \mathbf{c}_a$. And there's the problem: the values in $\mathbf{Y}_a$ are not available when the PLS model is being used in the future, on new data. In the future we will only have the new values of $\mathbf{X}$. This is why we would rather predict $\mathbf{X}_a$ using the $\mathbf{t}_a$ scores, since those $\mathbf{t}$-scores are available in the future when we apply the model to new data.

This whole discussion might also leave you asking why we even bother to have predictions of the $\mathbf{X}$. We do this primarily to ensure orthogonality among the $\mathbf{t}$-scores, by removing everything from $\mathbf{X}_a$ that those scores explain (see the next deflation step).

These predictions of $\widehat{\mathbf{X}}$ are also used to calculate the squared prediction error, a very important consistency check when using the PLS model on new data.

Step 2: Remove the predicted variability from X and Y

Using the loadings, $\mathbf{p}_a$ just calculated above, we remove from $\mathbf{X}_a$ the best prediction of $\mathbf{X}_a$, in other words, remove everything we can explain about it.

$$\widehat{\mathbf{X}}_a = \mathbf{t}_a \mathbf{p}_a'$$
$$\mathbf{E}_a = \mathbf{X}_a - \widehat{\mathbf{X}}_a = \mathbf{X}_a - \mathbf{t}_a \mathbf{p}_a'$$
$$\mathbf{X}_{a+1} = \mathbf{E}_a$$

For the first component, the $\mathbf{X}_{a=1}$ matrix contains the preprocessed raw $\mathbf{X}$-data. By convention, $\mathbf{E}_{a=0}$ is the residual matrix *before* fitting the first component and is just the same matrix as $\mathbf{X}_{a=1}$, i.e. the data used to fit the first component.

We also remove any variance explained from $\mathbf{Y}_a$:

$$\widehat{\mathbf{Y}}_a = \mathbf{t}_a \mathbf{c}_a'$$
$$\mathbf{F}_a = \mathbf{Y}_a - \widehat{\mathbf{Y}}_a = \mathbf{Y}_a - \mathbf{t}_a \mathbf{c}_a'$$
$$\mathbf{Y}_{a+1} = \mathbf{F}_a$$

For the first component, the $\mathbf{Y}_{a=1}$ matrix contains the preprocessed raw data. By convention, $\mathbf{F}_{a=0}$ is the residual matrix *before* fitting the first component and is just the same matrix as $\mathbf{Y}_{a=1}$.

Notice how in both deflation steps we only use the scores, $\mathbf{t}_a$, to deflate. The scores, $\mathbf{u}_a$, are not used for the reason described above: when applying the PLS model to new data in the future, we won't have the actual y-values, which means we also don't know the $\mathbf{u}_a$ values.

The algorithm repeats all over again using the deflated matrices for the subsequent iterations.

What is the difference between $\mathbf{W}$ and $\mathbf{R}$?

After reading about the *NIPALS algorithm for PLS* (page 382) you should be aware that we deflate the $\mathbf{X}$ matrix after every component is extracted. This means that $\mathbf{w}_1$ are the weights that best predict the $\mathbf{t}_1$ score values, our summary of the data in $\mathbf{X}_{a=1}$ (the preprocessed raw data). Mathematically we can write the following:

$$\mathbf{t}_1 = \mathbf{X}_{a=1}\mathbf{w}_1 = \mathbf{X}_1 \mathbf{w}_1$$

The problem comes once we deflate. The $\mathbf{w}_2$ vector is calculated from the deflated matrix $\mathbf{X}_{a=2}$, so interpreting these scores is a quite a bit harder.

$$\mathbf{t}_2 = \mathbf{X}_2 \mathbf{w}_2 = \left(\mathbf{X}_1 - \mathbf{t}_1 \mathbf{p}_1'\right)\mathbf{w}_2$$
$$= \left(\mathbf{X}_1 - \mathbf{X}_1 \mathbf{w}_1 \mathbf{p}_1\right)\mathbf{w}_2$$

The $\mathbf{w}_2$ is not really giving us insight into the relationships between the score, $\mathbf{t}_2$, and the data, $\mathbf{X}$, but rather between the score and the *deflated* data, $\mathbf{X}_2$.

Ideally we would like a set of vectors we can interpret directly; something like:

$$\mathbf{t}_a = \mathbf{X}\mathbf{r}_a$$

One can show, using repeated substitution, that a matrix $\mathbf{R}$, whose columns contain $\mathbf{r}_a$, can be found from: $\mathbf{R} = \mathbf{W}\left(\mathbf{P}'\mathbf{W}\right)^{-1}$. The first column, $\mathbf{r}_1 = \mathbf{w}_1$.

So our preference is to interpret the $\mathbf{R}$ weights (often called $\mathbf{W}^*$ in some literature), rather than the $\mathbf{W}$ weights when investigating the relationships in a PLS model.

6.7.8 Variability explained with each component

We can calculate R^2 values, since PLS explains both the $\mathbf{X}$-space and the $\mathbf{Y}$-space. We use the $\mathbf{E}_a$ matrix to calculate the cumulative variance explained for the $\mathbf{X}$-space.

$$R^2_{\mathbf{X},a,\text{cum}} = 1 - \frac{\text{Var}(\mathbf{E}_a)}{\text{Var}(\mathbf{X}_{a=1})}$$

Before the first component is extracted we have $R^2_{\mathbf{X},a=0} = 0.0$, since $\mathbf{E}_{a=0} = \mathbf{X}_{a=1}$. After the second component, the residuals, $\mathbf{E}_{a=1}$, will have decreased, so $R^2_{\mathbf{X},a}$ would have increased.

We can construct similar R^2 values for the $\mathbf{Y}$-space using the $\mathbf{Y}_a$ and $\mathbf{F}_a$ matrices. Furthermore, we construct in an analogous manner the R^2 values for each column of $\mathbf{X}_a$ and $\mathbf{Y}_a$, exactly as *we did for PCA* (page 346).

These R^2 values help us understand which components best explain different sources of variation. Bar plots of the R^2 values for each column in $\mathbf{X}$ and $\mathbf{Y}$, after a certain number of A components are one of the best ways to visualize this information.

6.7.9 Coefficient plots in PLS

After building an initial PLS model one of the most informative plots to investigate are plots of the $\mathbf{r} : \mathbf{c}$ vectors: using either bar plots or scatter plots. (The notation $\mathbf{r} : \mathbf{c}$ implies we superimpose a plot of $\mathbf{r}$ on a plot of $\mathbf{c}$.) These plots show the relationship between variables in $\mathbf{X}$, between variables in $\mathbf{Y}$, as well as the latent variable relationship between these two spaces. The number of latent variables, A, is much smaller number than the original variables, $K + M$, effectively compressing the data into a small number of informative plots.

There are models where the number of components is of moderate size, around $A = 4$ to 8, in which case there are several combinations of $\mathbf{r} : \mathbf{c}$ plots to view. If we truly want to understand how all the $\mathbf{X}$ and $\mathbf{Y}$ variables are related, then we must spend time investigating all these plots. However, the coefficient plot can be a useful compromise if one wants to learn, in a single plot, how the $\mathbf{X}$ variables are related to the $\mathbf{Y}$ variables using *all A components*.

> **Caution using the coefficients**
>
> It is not recommended that PLS be implemented in practice as described here. In other words, do not try make PLS like multiple linear regression and go directly from the $\mathbf{X}$'s to the $\mathbf{Y}$'s using $\widehat{\mathbf{y}}'_{\text{new}} = \mathbf{x}'_{\text{new}}\beta$.
>
> Instead, one of the major benefits of a PLS model is that we first calculate the scores, then verify T^2 and SPE are below their critical limits, e.g. 95% limits. If so, then we go ahead and calculate the predictions of $\mathbf{Y}$. Direct calculation of $\mathbf{Y}$ bypasses this helpful information. Furthermore, using the β coefficients directly means that we cannot handle missing data.
>
> *Only use the coefficients to learn about your system*. Do not use them for prediction.

The coefficient plot is derived as follows. First preprocess the new observation, $\mathbf{x}_{\text{new,raw}}$, to obtain $\mathbf{x}_{\text{new}}$.

- Project the new observation onto the model to get scores: $\mathbf{t}'_{\text{new}} = \mathbf{x}'_{\text{new}}\mathbf{R}$.

- Calculate the predicted $\widehat{\mathbf{y}}'_{\text{new}} = \mathbf{t}'_{\text{new}}\mathbf{C}'$ using these scores.

- Now combine these steps:

$$\begin{aligned}
\widehat{\mathbf{y}}'_{\text{new}} &= \mathbf{t}'_{\text{new}}\mathbf{C}' \\
\widehat{\mathbf{y}}'_{\text{new}} &= \mathbf{x}'_{\text{new}}\mathbf{R}\mathbf{C}' \\
\widehat{\mathbf{y}}'_{\text{new}} &= \mathbf{x}'_{\text{new}}\beta
\end{aligned}$$

where the matrix β is a $K \times M$ matrix: each column in β contains the regression coefficients for all K of the $\mathbf{X}$ variables, showing how they are related to each of the M $\mathbf{Y}$-variables.

From this derivation we see these regression coefficients are a function of *all* the latent variables in the model, since $\mathbf{R} = \mathbf{W}\left(\mathbf{P}'\mathbf{W}\right)^{-1}$ as shown in *an earlier section of these notes* (page 384).

In the example below there were $A = 6$ components, and $K = 14$ and $M = 5$. Investigating all 6 of the $\mathbf{r} : \mathbf{c}$ vectors is informative, but the coefficient plot provides an efficient way to understand how the $\mathbf{X}$ variables are related to this particular $\mathbf{Y}$ variable across all the components in the model.
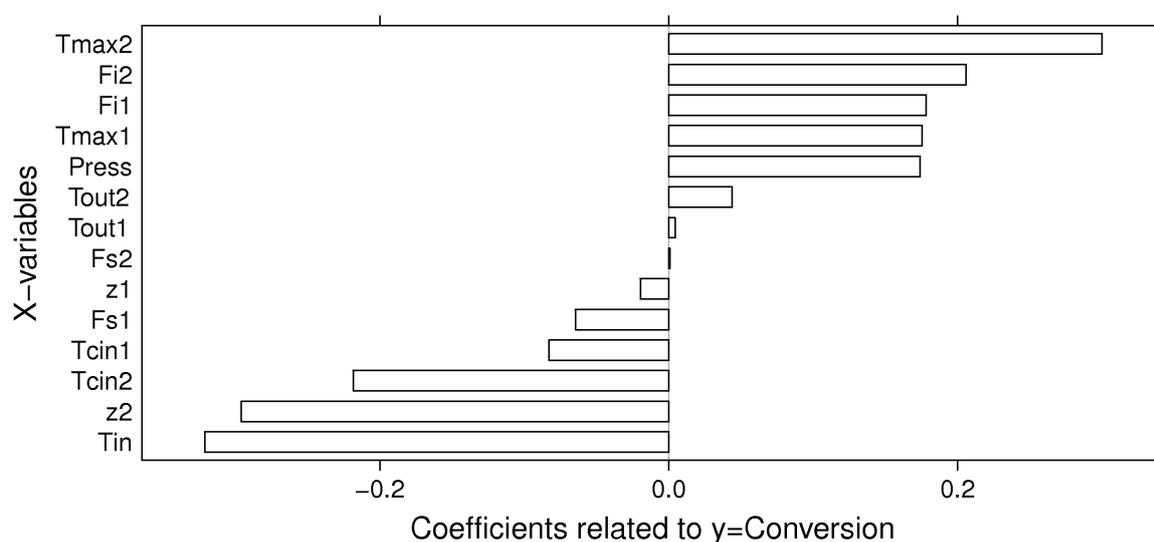
In this example the `Tin`, `z2`, `Tcin2` and `Tmax2`, `Fi2`, `Fi1`, `Tmax1`, and `Press` variables are all related to conversion, the y variable. This does not imply a cause and effect relationships, rather it just shows they are strongly correlated.

6.7.10 Analysis of designed experiments using PLS models

Data from a designed experiment, particularly factorial experiments, will have independent columns in $\mathbf{X}$. These data tables are adequately analyzed using *multiple linear regression* (page 185) (MLR) least squares models.

These factorial and fractional factorial data are also well suited to analysis with PLS. Since factorial models support interaction terms, these additional interactions should be added to the $\mathbf{X}$ matrix. For example, a full factorial design with variables $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ will also support the $\mathbf{AB}$, $\mathbf{AC}$, $\mathbf{BC}$ and $\mathbf{ABC}$ interactions. These four columns should be added to the $\mathbf{X}$ matrix so that the loadings for these variables are also estimated. If a *central composite design* (page 283), or some other design that supports quadratic terms has been performed, then these columns should also be added to $\mathbf{X}$, e.g.: A^2, B^2 and C^2.

The PLS loadings plots from analyzing these DOE data are interpreted in the usual manner; and the coefficient plot is informative if $A > 2$.

There are some other advantages of using and interpreting a PLS model built from DOE data, rather than using the MLR approach:

- If *additional data* (not the main factors) are captured during the experiments, particularly measurable disturbances, then these additional columns can, and should, be included in $\mathbf{X}$. These extra data are called covariates in other software packages. These additional columns will remove some of the orthogonality in $\mathbf{X}$, but this is why a PLS model would be more suitable.

- If multiple $\mathbf{Y}$ measurements were recored as the response, and particularly if these $\mathbf{Y}$ variables are correlated, then a PLS model would be better suited than building K separate MLR models. A good example is where the response variable from the experiment is a complete spectrum of measurements, such as from a NIR probe.

One other point to note when analyzing DOE data with PLS is that the Q^2 values from cross-validation are often very small. This makes intuitive sense: if the factorial levels are suitably spaced, then each experiment is at a point in the process that provides new information. It is unlikely that cross-validation, when leaving out one or more experiments, is able to accurately predict each corner in the factorial.

Lastly, models built from DOE data allow a much stronger interpretation of the loading vectors, $\mathbf{R} : \mathbf{C}$. This time we can infer cause-and-effect behaviour; normally in PLS models the best we can say is that the variables in $\mathbf{X}$ and $\mathbf{Y}$ are correlated. Experimental studies that are run in a factorial manner will break happenstance correlation structures; so if any correlation that is present, then this truly is causal in nature.

6.7.11 PLS Exercises

The taste of cheddar cheese

- $N = 30$

- $K = 3$

- $M = 1$

- Link to cheese data[159]

- Description: This very simple case study considers the taste of mature cheddar cheese. There are 3 measurements taken on each cheese: lactic acid, acetic acid and H_2S.

1. Import the data into R: `cheese <- read.csv('http://openmv.net/file/cheddar-cheese.csv')`

2. Use the `car` library and plot a scatter plot matrix of the raw data:

   - `library(car)`

   - `scatterplotMatrix(cheese[,2:5])`

---

[159] http://openmv.net/info/cheddar-cheese
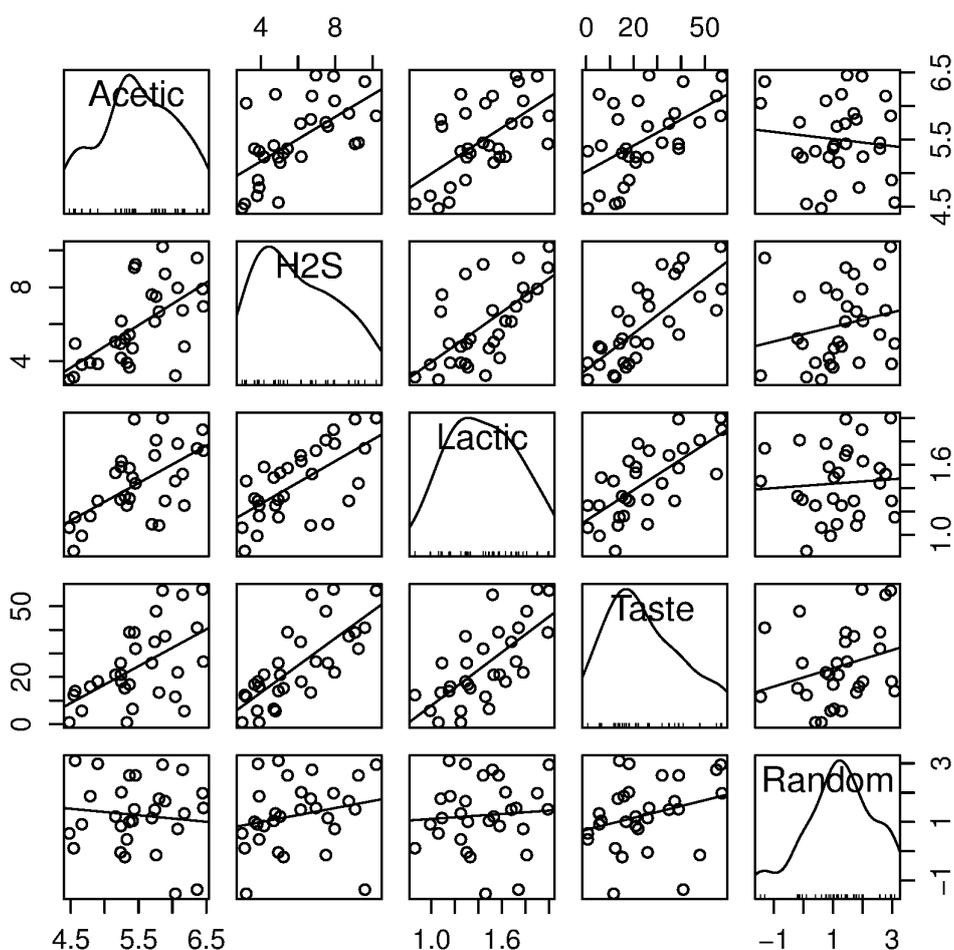
```
                                          cheese-plots.R
filename <- 'http://openmv.net/file/cheddar-cheese.csv'
cheese <- read.csv(filename)
summary(cheese)

library(car)
scatterplotMatrix(cheese[, 2:5],
                  col=c(1,1,1),
                  smooth=FALSE)
```

3. Using this figure, how many components do you expect to have in a PCA model on the 3 **X** variables: `Acetic`, `H2S` and `Lactic`?

4. What would the loadings look like?

5. Build a PCA model now to verify your answers.

```
                                          cheese-plots.R
filename <- 'http://openmv.net/file/cheddar-cheese.csv'
cheese <- read.csv(filename)
summary(cheese)

model.pca <- prcomp(cheese[, 2:5],
                    scale=TRUE)
summary(model.pca)
```

```
loadings.P <- model.pca$rotation
scores.T <- model.pca$x
```

6. Before building the PLS model, how many components would you expect? And what would the weights look like ($\mathbf{r}_1$, and $\mathbf{c}_1$)?

7. Build a PLS model and plot the $\mathbf{r} : \mathbf{c}_1$ bar plot. Interpret it.

8. Now plot the SPE plot; these are the SPE values for the projections onto the $\mathbf{X}$-space. Any outliers apparent?

9. In R, build a least squares model that regresses the `Taste` variable on to the other 3 $\mathbf{X}$ variables.

   - `model.lm <- lm(Taste ~ Acetic + H2S + Lactic, data=cheese)`

   - Report each coefficient $\pm 2 S_E(b_i)$. Which coefficients does R find significant in MLR? (You can use the `confint(model.lm)` function too.)

$$\beta_{\text{Acetic}} = \qquad \pm$$
$$\beta_{\text{H2S}} = \qquad \pm$$
$$\beta_{\text{Lactic}} = \qquad \pm$$

   - Report the standard error and the R_y^2 value for this model.

   - Compare this to the PLS model's R_y^2 value.

```R
                                          R code
cheese <- read.csv('http://openmv.net/file/cheddar-cheese.csv')
summary(cheese)

# Least squares model:
model.lm <- lm(Taste ~ Acetic + H2S +
               Lactic, data=cheese)
resid = residuals(model.lm)
resid.ssq = sum(resid**2)
standard.error = sqrt( resid.ssq /
                 model.lm$df.residual )
ssq.total = sum((cheese$Taste -
                mean(cheese$Taste)) ** 2)
R2.value = 1 - resid.ssq / ssq.total
paste0('Least squares SE = ',
       round(standard.error, 2))
paste0('Least squares R^2 = ',
       round(R2.value*100, 2), '%')
```

10. Now build a PCR model in R using only 1 component, then using 2 components. Again calculate the standard error and R_y^2 values.

```R
                                          R code
cheese <- read.csv('http://openmv.net/file/cheddar-cheese.csv')
summary(cheese)

# PCA model with only 2 components
model.pca <- prcomp(cheese[,2:4],
                    scale = TRUE,
                    rank. = 2)

scores.T <- model.pca$x

# PCR model using only PC 1
pcr.1 <- lm(cheese$Taste ~ scores.T[,1])
```

```
# PCR model using PC 1 and PC 2
pcr.2 <- lm(cheese$Taste ~ scores.T[,1:2])

SE.1 <- sqrt( sum( residuals(pcr.1)^2 ) /
                    pcr.1$df.residual )
SE.2 <- sqrt( sum( residuals(pcr.2)^2 ) /
                pcr.2$df.residual )

paste0('SE for PCR with 1 component: ',
        round(SE.1, 2))
paste0('SE for PCR with 2 components: ',
        round(SE.2, 2))
```

11. Plot the observed y values against the predicted y values for the PLS model.

12. PLS models do not have a standard error, since the degrees of freedom are not as easily defined. But you can calculate the RMSEE (root mean square error of estimation) = $\sqrt{\dfrac{\mathbf{e}'\mathbf{e}}{N}}$. Compare the RMSEE values for all the models just built.

Obviously the best way to test the model is to retain a certain amount of testing data (e.g. 10 observations), then calculate the root mean square error of prediction (RMSEP) on those testing data.

Comparing the loadings from a PCA model to a PLS model

PLS explains both the $\mathbf{X}$ and $\mathbf{Y}$ spaces, as well as building a predictive model between the two spaces. In this question we explore two models: a PCA model and a PLS model on the same data set.

The data are from the *plastic pellets troubleshooting example* (page 370).

- $N = 24$

- $K = 6 + 1$ designation of process outcome

- Link to raw materials data[160]

- Description: 3 of the 6 measurements are size values for the plastic pellets, while the other 3 are the outputs from thermogravimetric analysis (TGA), differential scanning calorimetry (DSC) and thermomechanical analysis (TMA), measured in a laboratory. These 6 measurements are thought to adequately characterize the raw material. Also provided is a designation `Adequate` or `Poor` that reflects the process engineer's opinion of the yield from that lot of materials.

1. Build a PCA model on all seven variables, including the 0-1 process outcome variable in the $\mathbf{X}$ space. Previously we omitted that variable from the model, this time include it.

2. How do the loadings look for the first, second and third components?

3. Now build a PLS model, where the $\mathbf{Y}$-variable is the 0-1 process outcome variable. In the previous PCA model the loadings were oriented in the directions of greatest variance. For the PLS model the loadings must be oriented so that they *also* explain the $\mathbf{Y}$ variable and the relationship between $\mathbf{X}$ and $\mathbf{Y}$. Interpret the PLS loadings in light of this fact.

4. How many components were required by cross-validation for the PLS model?

5. Explain why the PLS loadings are different to the PCA loadings.

---

[160] http://openmv.net/info/raw-material-characterization

Predicting final quality from on-line process data: LDPE system

- $N = 54$

- $K = 14$

- $K = 5$

- Link to dataset website[161] and description of the data.

1. Build a PCA model on the 14 **X**-variables and the first 50 observations.

2. Build a PCA model on the 5 **Y**-variables: `Conv`, `Mn`, `Mw`, `LCB`, and `SCB`. Use only the first 50 observations

3. Build a PLS model relating the **X** variables to the **Y** variables (using $N = 50$). How many components are required for each of these 3 models?

4. Compare the loadings plot from PCA on the **Y** space to the weights plot (c_1 vs c_2) from the PLS model.

5. What is the R_X^2 (not for **Y**) for the first few components?

6. Now let's look at the interpretation between the **X** and **Y** space. Which plot would you use?

   - Which variable(s) in **X** are strongly related to the conversion of the product (`Conv`)? In other words, as an engineer, which of the 14 **X** variables would you consider adjusting to improve conversion.

   - Would these adjustments affect any other quality variables? How would they affect the other quality variables?

   - How would you adjust the quality variable called `Mw` (the weight average molecular weight)?

6.8 Applications of Latent Variable Models

6.8.1 Improved process understanding

Interpreting the loadings plot (page 339) from a model is well worth the time spent. At the very least, one will confirm what you already know about the process, but sometimes there are unexpected insights that are revealed. Guide your interpretation of the loadings plot with contributions in the scores, and cross-referencing with the raw data, to verify your interpretation.

There are A loadings and score plots. In many cases this is far fewer than the K number of original variables. Furthermore, these A variables have a much higher signal and lower noise than the original data. They can also be calculated if there are missing data present in the original variables.

In the example shown here the company was interested in how their product performed against that of their competitor. Six variables called A to F were measured on all the product samples, (codes are used, because the actual variables measured are proprietary). The loadings for these 6 variables are labelled below, while the remaining points are the scores. The scores have been scaled and superimposed on the loadings plot, to create what is called a biplot. The square, green points were the competitor's product, while the smaller purple squares were their own product.

---

[161] http://openmv.net/info/LDPE
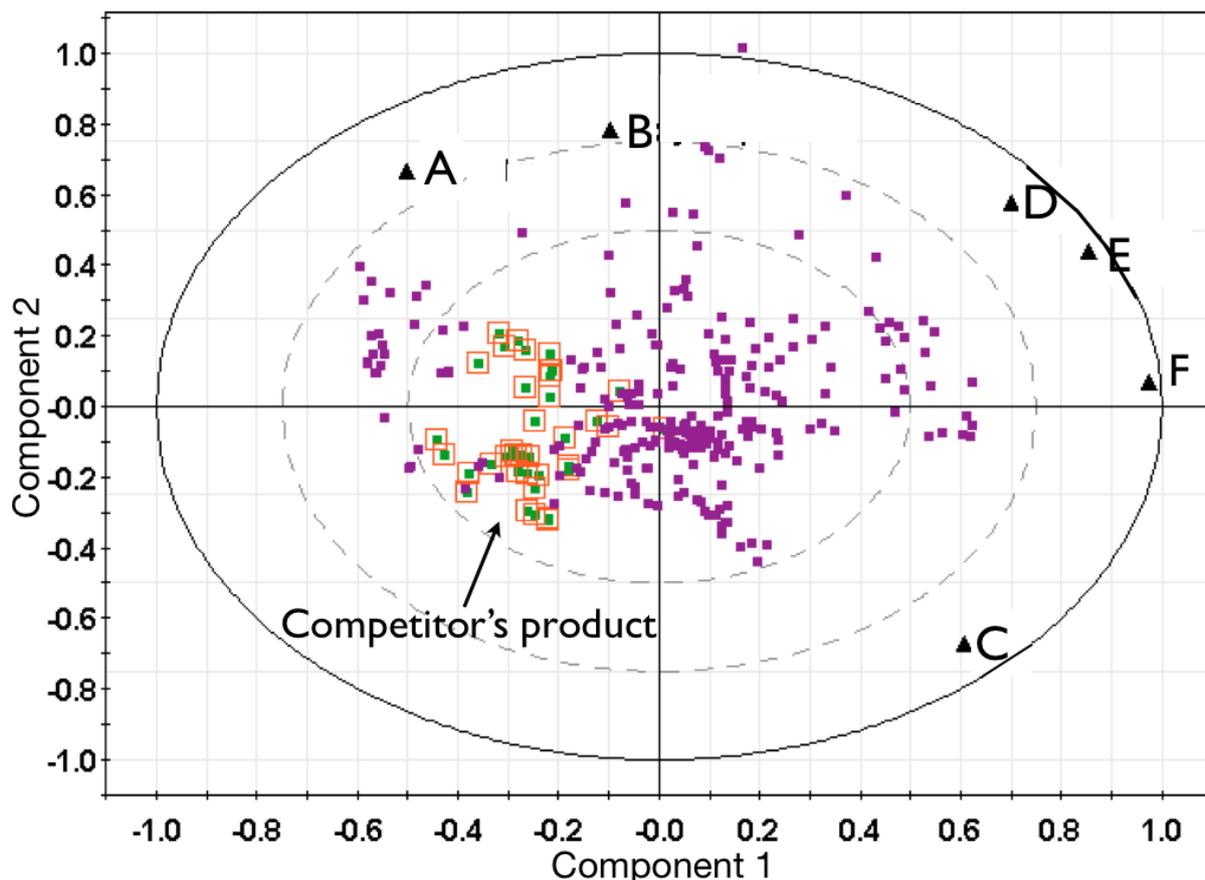
From this single figure the company learned that:

- The quality characteristics of this material is not six-dimensional; it is two-dimensional. This means that based on the data used to create this model, there is no apparent way to independently manipulate the 6 quality variables. Products produced in the past land up on a 2-dimensional latent variable plane, rather than a 6-dimensional space.

- Variables D, E and F in particular are very highly correlated, while variable C is also somewhat correlated with them. Variable A and C are negatively correlated, as are variables B and C. Variables A and B are positively correlated with each other.

- This company' competitor was able to manufacture the product with much lower variability than they were: there is greater spread in their own product, while the competitor's product is tightly clustered.

- The competitors product is characterized as having lower values of C, D, E, and F, while the values of A and B are around the average.

- The company had produced product similar to their competitor's only very infrequently, but since their product is spread around the competitor's, it indicates that they could manufacture product of similar characteristics to their competitor. They could go query the score values close those of those of the competitors and using their company records, locate the machine and other process settings in use at that time.

  However, it might not just be *how* they operate the process, but also which raw materials and their consistency, and the control of outside disturbances on the process. These all factor into the final product's variability.

It it is not shown here, but the competitor's product points are close to the model plane (low SPE values), so this comparison is valid. This analysis was tremendously insightful, and easier to complete on this single plot, rather than using plots of the original variables.

6.8.2 Troubleshooting process problems

We already saw a troubleshooting example in the section on *interpreting scores* (page 336). In general, troubleshooting with latent variable methods uses this approach:

1. Collect data from all relevant parts of the process: do not exclude variables that you think might be unimportant; often the problems are due to unexpected sources. Include information on operators, weather, equipment age (e.g. days since pump replacement), raw material properties being processed at that time, raw material supplier (indicator variable). Because the PCA model disregards unimportant or noisy variables, these can later be pruned out, but they should be kept in for the initial analysis. (Note: this does not mean the uninformative variables are not important - they might only be uninformative during the period of data under observation).

2. Structure the data so that the majority of the data is from normal, common-cause operation. The reason is that the PCA model plane should be oriented in the directions of normal operation. The rest of the $\mathbf{X}$ matrix should be from when the problem occurs and develops.
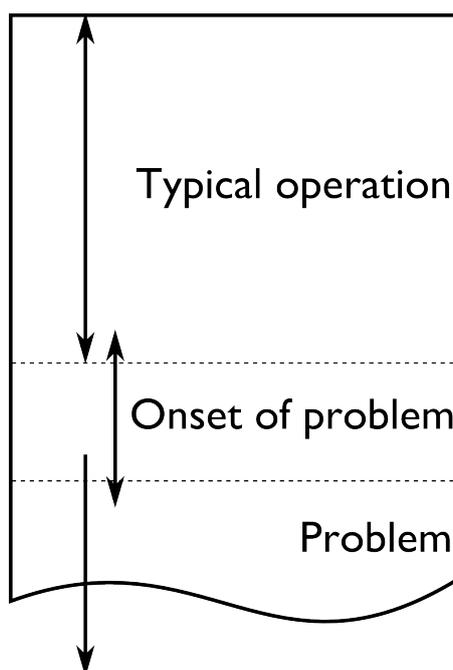
3. Given the wealth of data present on many processes these days, it is helpful to prune the $\mathbf{X}$ matrix so that it is only several hundred rows in length. Simply subsample, or using averages of time; e.g. hourly averages. Later we can come back and look at a higher resolution. Even as few as 50 rows can often work well.

4. Build the PCA model. You should observe the abnormal operation appearing as outliers in the score plots and SPE plots. If not, use colours or different markers to highlight the regions of poor operation in the scores: they might be clustered in a region of the score plot, but not appear as obvious outliers.

5. Interrogate and think about the model. Use the loadings plots to understand the general trends between the variables. Use contribution plots to learn why clusters of observations are different from others. Use contribution plots to isolate the variables related to large SPE values.

6. It should be clear that this is all iterative work; the engineer has to be using her/his brain to formulate hypotheses, and then verify them in the data. The latent variable models help to reduce the size of the problem down, but they do not remove the requirement to think about the data and interpret the results.

Here is an example where the yield of a company's product was declining. They suspected that their raw material was changing in some way, since no major changes had occurred on their process. They measured 6 characteristic values on each lot (batch) of raw materials: 3 of them were a size measurement on the plastic pellets, while the other 3 were the outputs from thermogravimetric analysis (TGA), differential scanning calorimetry (DSC) and thermomechanical analysis (TMA), measured in a laboratory. Also provided was an indication of the yield: "Adequate" or "Poor". There were 24 samples in total, 17 batches of adequate yield and the rest the had poor yield.

The score plot (left) and loadings plot (right) help isolate potential reasons for the reduced yield. Batches with reduced yield have high, positive t_2 values and low, negative t_1 values. What factors lead to batches having score values with this combination of t_1 and t_2? It would take batches with a combination of low values of TGA and TMA, and/or above average size5, size10 and size15 levels, and/or high DSC values to get these sort of score values. These would be the *generally expected* trends, based on an interpretation of the scores and loadings.
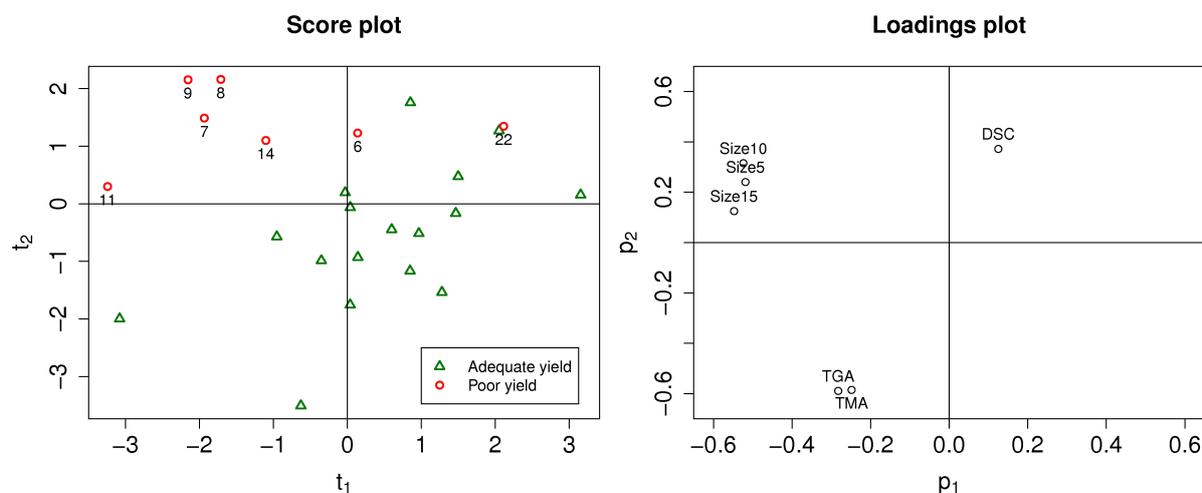
We can investigate *specific* batches and look at the contribution of each variable to the score values. Let's look at the contributions for batch 8 for both the t_1 and t_2 scores.

$$
\begin{array}{lccccccccccc}
t_{8,a=1} & = & x_{s5}\,p_{s5,1} & + & x_{s10}\,p_{s10,1} & + & x_{s15}\,p_{s15,1} & + & x_{TGA}\,p_{TGA,1} & + & x_{DSC}\,p_{DSC,1} & + & x_{TMA}\,p_{TMA,1} \\
t_{8,a=1} & = & -0.85 & - & 0.74 & - & 0.62 & + & 0.27 & + & 0.12 & + & 0.10 \\
t_{8,a=2} & = & x_{s5}\,p_{s5,2} & + & x_{s10}\,p_{s10,2} & + & x_{s15}\,p_{s15,2} & + & x_{TGA}\,p_{TGA,2} & + & x_{DSC}\,p_{DSC,2} & + & x_{TMA}\,p_{TMA,2} \\
t_{8,a=2} & = & 0.39 & + & 0.44 & + & 0.14 & + & 0.57 & + & 0.37 & + & 0.24
\end{array}
$$

Batch 8 is at its location in the score plot due to the low values of the 3 size variables (they have strong negative contributions to t_1, and strong positive contributions to t_2); and also because of its very large DSC value (the 0.57 contribution in t_2).

Batch 22 on the other hand had very low values of TGA and TMA, even though its size values were

below average. Let's take a look at the t_2 value for batch 22 to see where we get this interpretation:

$$
\begin{aligned}
t_{22,a=2} &= x_{\text{s5}}\, p_{\text{s5},2} &+& x_{\text{s10}}\, p_{\text{s10},2} &+& x_{\text{s15}}\, p_{\text{s15},2} &+& x_{\text{TGA}}\, p_{\text{TGA},2} &+& x_{\text{DSC}}\, p_{\text{DSC},2} &+& x_{\text{TMA}}\, p_{\text{TMA},2} \\
t_{22,a=2} &= -0.29 &-& 0.17 &-& 0.08 &+& 0.84 &-& 0.05 &+& 1.10
\end{aligned}
$$

This illustrates that the actual contribution values are a more precise diagnostic tool that just interpreting the loadings.

6.8.3 Optimizing: new operating point and/or new product development

This application area is rapidly growing in importance. Fortunately it is fairly straightforward to get an impression of how powerful this tool is. Let's return back to the *food texture example considered previously* (page 332), where data from a biscuit/pastry product was considered. These 5 measurements were used:

1. Percentage oil in the pastry

2. The product's density (the higher the number, the more dense the product)

3. A crispiness measurement, on a scale from 7 to 15, with 15 being more crispy.

4. The product's fracturability: the angle, in degrees, through which the pasty can be slowly bent before it fractures.

5. Hardness: a sharp point is used to measure the amount of force required before breakage occurs.

The scores and loadings plot are repeated here again:
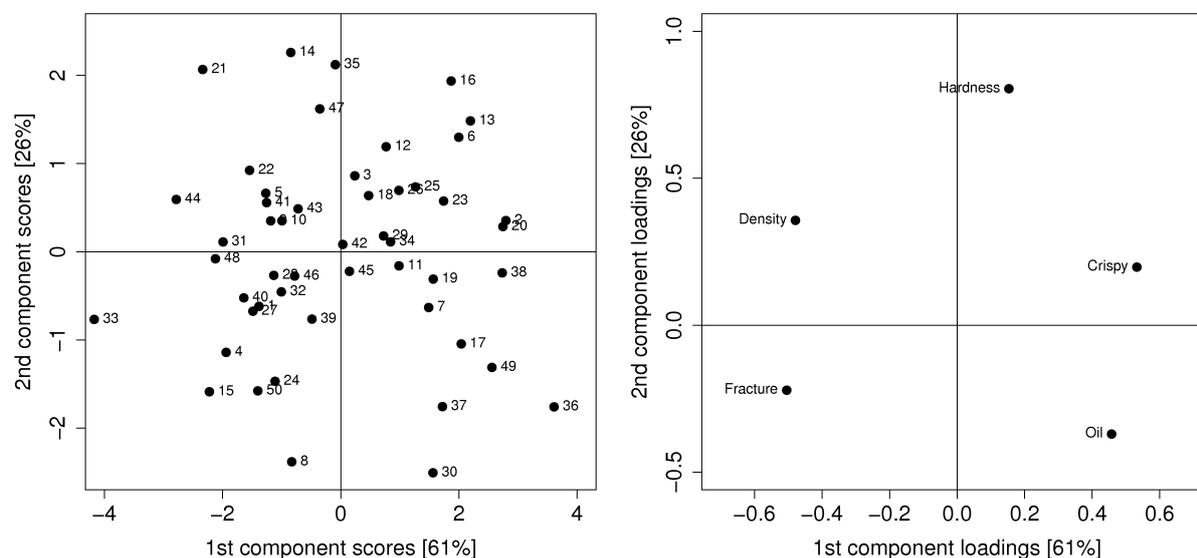
Process optimization follows the principle that certain regions of operation are more desirable than others. For example, if all the pastry batches produced on the score plot are of acceptable quality, there might be regions in the plot which are more economically profitable than others.

For example, pastries produced in the lower right quadrant of the score plot (high values of t_1 and low values of t_2), require more oil, but might require a lower cooking time, due to the decreased product density. Economically, the additional oil cost is offset by the lower energy costs. All other things being equal, we can optimize the process by moving production conditions so that we consistently produce

pastries in this region of the score plot. We could cross-reference the machine settings for the days when batches 17, 49, 36, 37 and 30 were produced and ensure we always operate at those conditions.

New product development follows a similar line of thought, but uses more of a "what-if" scenario. If market research or customer requests show that a pastry product with lower oil, but still with high crispiness is required, we can initially guess from the loadings plot that this is not possible: oil percentage and crispiness are positively correlated, not negatively correlated.

But if our manager asks, can we readily produce a pastry with the 5 variables set at [Oil=14%, Density=2600, Crispy=14, Fracture can be any value, Hardness=100]. We can treat this as a new observation, and following the steps described in the earlier *section on using a PCA model* (page 359), we will find that $\mathbf{e} = [2.50, 1.57, -1.10, -0.18, 0.67]$, and the SPE value is 10.4. This is well above the 95% limit of SPE, indicating that such a pastry is not consistent with how we have run our process in the past. So there isn't a quick solution.

Fortunately, there are systematic tools to move on from this step, but they are beyond the level of this introductory material. They involve the inversion and optimization of latent variable models. This paper is a good starting point if you are interested in more information: Christiane Jaeckle and John MacGregor, "Product design through multivariate statistical analysis of process data[162]". *AIChE Journal*, **44**, 1105-1118, 1998.

The general principle in model inversion problems is to manipulate the any degrees of freedom in the process (variables that can be manipulated in a process control sense) to obtain a product as close as possible to the required specification, but with low SPE in the model. A PLS model built with these manipulated variables, and other process measurements in $\mathbf{X}$, and collecting the required product specifications in $\mathbf{Y}$ can be used for these model inversion problems.

6.8.4 Predictive modelling (inferential sensors)

This section will be expanded soon, but we give an outline here of what inferential sensors are, and how they are built. These sensors also go by the names of software sensors or just soft sensors.

The intention of an inferential sensor is to infer a hard-to-measure property, usually a lab measurement or an expensive measurement, using a combination of process data and software-implemented algorithms.

Consider a distillation column where various automatic measurements are used to predict the vapour pressure. The actual vapour pressure is a lab measurement, usually taken 3 or 4 times per week, and takes several hours to complete. The soft sensor can predict the lab value from the real-time process measurements with sufficient accuracy. This is a common soft sensor on distillation columns. The lab values are used to build (train) the software sensor and to update in periodically.

Other interesting examples use camera images to predict hard-to-measure values. In the paper by Honglu Yu, John MacGregor, Gabe Haarsma and Wilfred Bourg[163] (*Ind. Eng. Chem. Res.*, **42**, 3036–3044, 2003), the authors describe how machine vision is used to predict, in real-time, the seasoning of various snack-food products. This sensors uses the colour information of the snacks to infer the amount of seasoning dispensed onto them. The dispenser is controlled via a feedback loop to ensure the seasoning is at target.

Once validated, a soft sensor can also reduce costs of a process by allowing for rapid feedback control of the inferred property, so that less off-specification product is produced. They also often have the

---

[162] https://dx.doi.org/10.1002/aic.690440509
[163] https://dx.doi.org/10.1021/ie020941f

side-effect that reduced lab sampling is required; this saves on manpower costs.

Soft sensors using latent variables will almost always be PLS models. Once the model has been built, it can be applied in real-time. The T^2 and SPE value for each new observation is checked for consistency with the model before a prediction is made. Contribution plots are used to diagnose unusual observations.

It is an indication that the predictive models need to be updated if the SPE and/or T^2 values are consistently above the limits. This is a real advantage over using an MLR-based model, which has no such consistency checks.

6.8.5 Process monitoring using latent variable methods

Any variable can be monitored using control charts, as we saw in the earlier section on *process monitoring* (page 109). The main purpose of these charts is to rapidly distinguish between two types of operation: in-control and out-of-control. We also aim to have a minimum number of false alarms (type I error: we raise an alarm when one isn't necessary) and the lowest number of false negatives possible (type II error, when an alarm should be raised, but we don't pick up the problem with the chart). We used Shewhart charts, CUSUM and EWMA charts to achieve these goals.

Consider the case of two variables, called x_1 and x_2, shown on the right, on the two horizontal axes. These could be time-oriented data, or just measurements from various sequential batches of material. The main point is that each variable's 3σ Shewhart control limits indicate that all observations are within control. It may not be apparent, but these two variables are negatively correlated with each other: as x_1 increases, the x_2 value decreases.
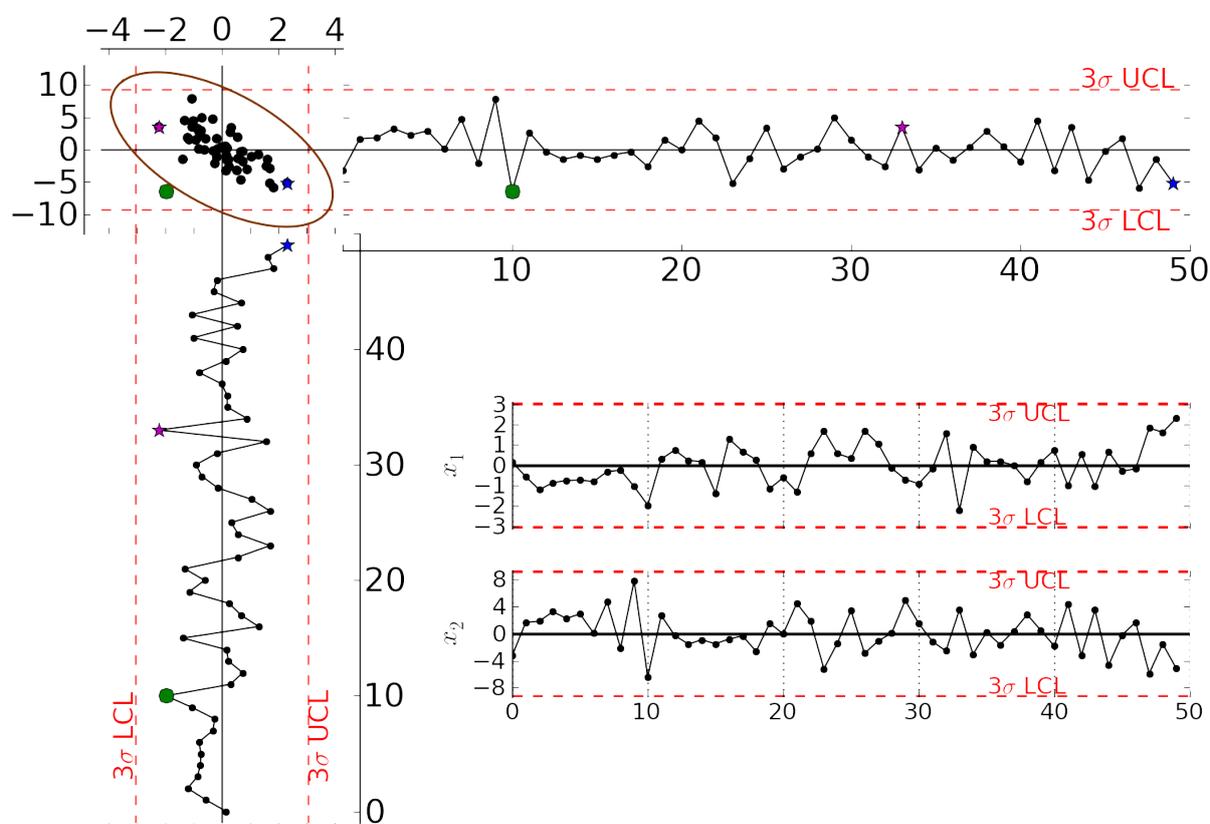
Rearranging the axes at 90 degrees to each other, and plotting the joint scatter plot of the two variables

in the upper left corner reveals the negative correlation, if you didn't notice it initially. Ignore the ellipse for now. It is clear that sample 10 (green closed dot, if these notes are printed in colour) is very different from the other samples. It is not an outlier from the perspective of x_1, nor of x_2, but jointly it is an outlier. This particular batch of materials would result in very different process operation and final product quality to the other samples. Yet a producer using separate control charts for x_1 and x_2 would not pick up this problem.

While using univariate control charts is *necessary* to pick up problems, univariate charts are not *sufficient* to pick up all quality problems if the variables are correlated. The key point here is that **quality is a multivariate attribute**. All our measurements on a system must be jointly within in the limits of common operation. Using only univariate control charts will raise the type II error: an alarm should be raised, but we don't pick up the problem with the charts.

Let's take a look at how process monitoring can be improved when dealing with *many attributes* (many variables). We note here that the same charts are used: Shewhart, CUSUM and EWMA charts, the only difference is that we replace the variables in the charts with variables from a *latent variable model*. We monitor instead the:

- scores from the model, $t_1, t_2, \ldots, t_A$

- Hotelling's $T^2 = \sum_{a=1}^{a=A} \left(\dfrac{t_a}{s_a} \right)^2$

- SPE value

The last two values are particularly appealing: they measure the on-the-plane and off-the-plane variation respectively, compressing K measurements into 2 very compact summaries of the process.

There are a few other good reasons to use latent variables models:

- The scores are orthogonal, totally uncorrelated to each other. The scores are also unrelated to the SPE: this means that we are not going to inflate our type II error rate, which happens when using correlated variables.

- There are far fewer scores than original variables on the process, yet the scores capture all the essential variation in the original data, leading to fewer monitoring charts on the operators' screens.

- We can calculate the scores, T^2 and SPE values even if there are missing data present; conversely, univariate charts have gaps when sensors go off-line.

- Rather than waiting for laboratory final quality checks, we can use the automated measurements from our process. There are many more of these measurements, so they will be correlated – we have to use latent variable tools. The process data are usually measured with greater accuracy than the lab values, and they are measured at higher frequency (often once per second). Furthermore, if a problem is detected in the lab values, then we would have to come back to these process data anyway to uncover the reason for the problem.

- But by far, one of the most valuable attributes of the process data is the fact that they are measured in real-time. The residence time in complex processes can be in the order of hours to days, going from start to end. Having to wait till much later in time to detect problems, based on lab measurements can lead to monetary losses as off-spec product must be discarded or reworked. Conversely, having the large quantity of data available in real-time means we can detect faults as they occur (making it much easier to decode what went wrong). But we need to use a tool that handles these highly correlated measurements.

A paper that outlines the reasons for multivariate monitoring is by John MacGregor, "Using on-line

process data to improve quality: Challenges for statisticians[164]", *International Statistical Review*, **65**, p 309-323, 1997.

We will look at the steps for phase I (building the monitoring charts) and phase II (using the monitoring charts).

Phase I: building the control chart

The procedure for building a multivariate monitoring chart, i.e. the phase I steps:

- Collect the relevant process data for the system being monitored. The preference is to collect the measurements of all attributes that characterize the system being monitored. Some of these are direct measurements, others might have to be calculated first.

- Assemble these measurements into a matrix **X**.

- As we did with univariate control charts, remove observations (rows) from **X** that are from out-of control operation, then build a latent variable model (either PCA or PLS). The objective is to build a model using only data that is from in-control operation.

- In all real cases the practitioner seldom knows which observations are from in-control operation, so this is an iterative step.

  - Prune out observations which have high T^2 and SPE (after verifying they are outliers).

  - Prune out variables in **X** that have low R^2.

- The observations that are pruned out are excellent testing data that can be set aside and used later to verify the detection limits for the scores, T^2 and SPE.

- The control limits depend on the type of variable:

  - Each score has variance of s_a^2, so this can be used to derive the Shewhart or EWMA control limits. Recall that Shewhart limits are typically placed at $\pm 3\sigma/\sqrt{n}$, for subgroups of size n.

  - Hotelling's T^2 and SPE have limits provided by the software (we do not derive here how these limits are calculated, though its not difficult).

  However, do not feel that these control limits are fixed. Adjust them up or down, using your testing data to find the desirable levels of type I and type II error.

- Keep in reserve some "known good" data to test what the type I error level is; also keep some "known out-of-control" data to assess the type II error level.

Phase II: using the control chart

The phase II steps, when we now wish to apply this quality chart on-line, are similar to the phase II steps for *univariate control charts* (page 112). Calculate the scores, SPE and Hotelling's T^2 for the new observation, $\mathbf{x}'_{\text{new}}$, as described in the *section on using an existing PCA model* (page 359). Then plot these new quantities, rather than the original variables. The only other difference is how to deal with an alarm.

The usual phase II approach when an alarm is raised is to investigate the variable that raised the alarm, and use your engineering knowledge of the process to understand why it was raised. When

---

[164] https://dx.doi.org/10.1111/j.1751-5823.1997.tb00311.x

using scores, SPE and T^2, we actually have a bit more information, but the approach is generally the same: use your engineering knowledge, in conjunction with the relevant contribution plot.

- A score variable, e.g. t_a raised the alarm. We *derived earlier* (page 336) that the contribution to each score was $t_{\text{new},a} = x_{\text{new},1} \; p_{1,a} + x_{\text{new},2} \; p_{2,a} + \ldots + x_{\text{new},k} \; p_{k,a} + \ldots + x_{\text{new},K} \; p_{K,a}$. It indicates which of the original K variables contributed most to the very high or very low score value.

- SPE alarm. The contribution to SPE for a new observation was derived in an *earlier section* (page 343) as well; it is conveniently shown using a barplot of the K elements in the vector below. These are the variables most associated with the broken correlation structure.

$$
\begin{aligned}
\mathbf{e}'_{\text{new}} &= \mathbf{x}'_{\text{new}} - \hat{\mathbf{x}}'_{\text{new}} = \mathbf{x}'_{\text{new}} - \mathbf{t}'_{\text{new}}\mathbf{P}' \\
&= \Big[(x_{\text{new},1} - \hat{x}_{\text{new},1}) \quad (x_{\text{new},2} - \hat{x}_{\text{new},2}) \quad \ldots \quad (x_{\text{new},k} - \hat{x}_{\text{new},k}) \quad \ldots \quad (x_{\text{new},K} - \hat{x}_{\text{new},K}) \Big]
\end{aligned}
$$

- T^2 alarm: an alarm in T^2 implies one or more scores are large. In many cases it is sufficient to go investigate the score(s) that caused the value of T^2_{new} to be large. Though as long as the SPE value is below its alarm level, many practitioners will argue that a high T^2 value really isn't an alarm at all; it indicates that the observation is multivariately in-control (on the plane), but beyond the boundaries of what has been observed when the model was built. My advice is to consider this point tentative: investigate it further (it might well be an interesting operating point that still produces good product).

6.8.6 Dealing with higher dimensional data structures

This section just gives a impression how 3-D and higher dimensional data sets are dealt with. Tools such as PCA and PLS work on two-dimensional matrices. When we receive a 3-dimensional array, such as an image, or a batch data set, then we must unfold that array into a (2D) matrix if we want to use PCA and PLS in the usual manner.

The following illustration shows how we deal with an image, such as the one taken from a colour camera. Imagine we have I rows and J columns of pixels, on 3 layers (red, green and blue wavelengths). Each entry in this array is an intensity value, a number between 0 and 255. For example, a pure red pixel is has the following 3 intensity values in layer 1, 2 and 3: (255, 0, 0), because layer 1 contains the intensity of the red wavelengths. A pure blue pixel would be (0, 0, 255), while a pure green pixel would be (0, 255, 0) and a pure white pixel is (255, 255, 255). In other words, each pixel is represented as a triplet of 3 intensity values.

Image layers

Unfold, going across the columns
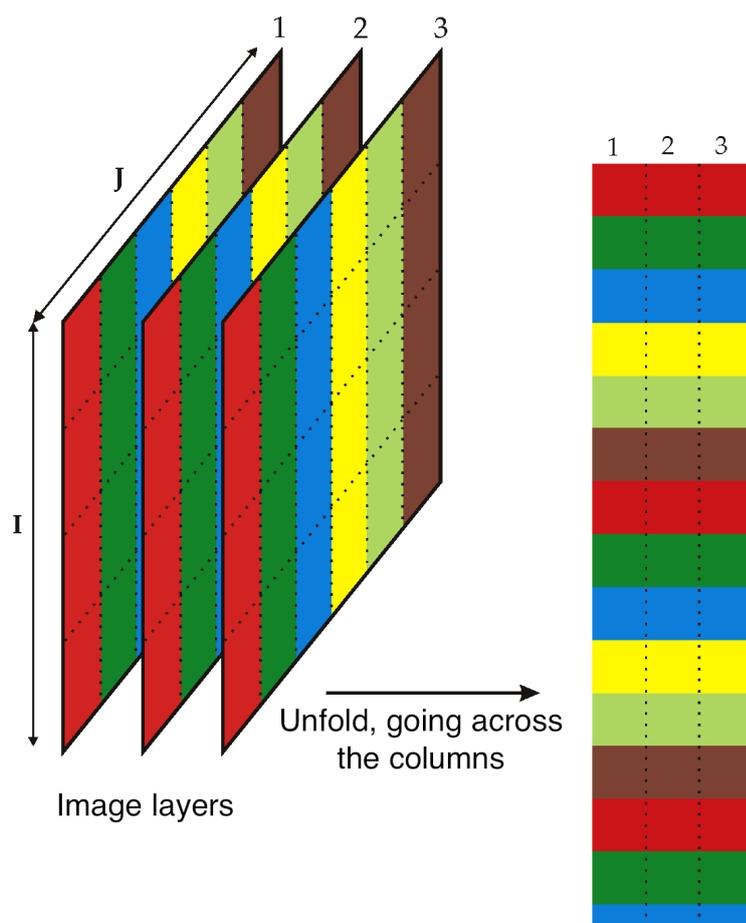
In the unfolded matrix we have IJ rows and 3 columns. In other words, each pixel in the image is represented in its own row. A digital image with 768 rows and 1024 columns, would therefore be unfolded into a matrix with 786,432 rows and 3 columns. If we perform PCA on this matrix we can calculate score values and SPE values: one per pixel. Those scores can be refolded back into the original shape of the image. It is useful to visualize those scores and SPE values in this way.

The RGB image from the data file

Intensity image of principal component 1

Residual intensity image after 1 PC extracted
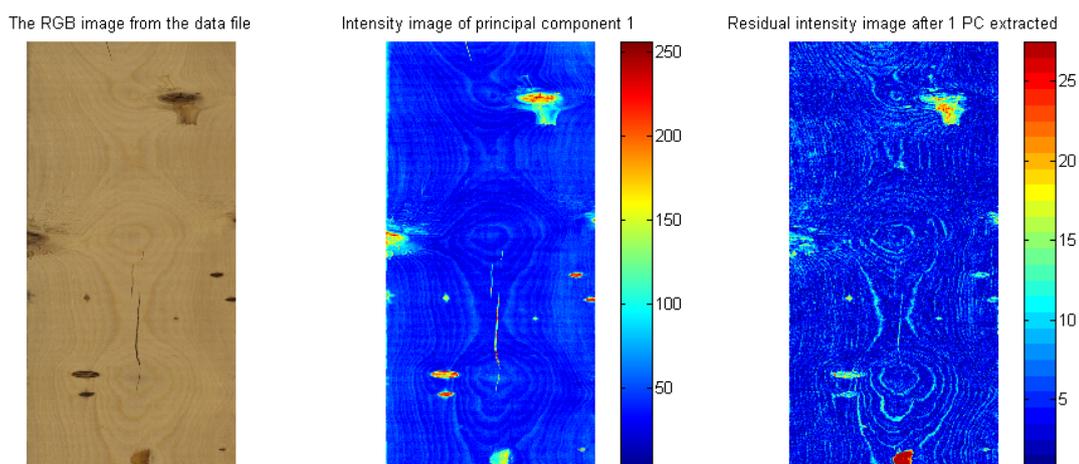
You can learn more about using PCA on image data in the manual that accompanies the interactive software that is freely available from http://macc.mcmaster.ca/maccmia.php.

APPLICATIONS OF PROCESS IMPROVEMENT USING DATA

Over the years since 2010 when this online book has been available, there has been the request of case studies and examples showing how the tools can be implemented. The focus of this chapter is not just the direct implementation of the equations and tools, but a discussion about the thought process, the details and issues that came up during the cases.

7.1 Product development and product improvement

This section covers product development, but it is called product improvement. The reason is that new products are seldom developed, but products are regularly improved as the following usage examples show:

7.1.1 Usage examples

- *Colleague*: Our most high profile customer wants us to develop a product with similar, but different specifications to the prior products we have made. Is it feasible to say '*yes*' to them?

- *You*: we have an existing product, but a customer just wants to change one of the 7 specifications: they want a slightly higher *viscosity*. Keep the ingredients and ratios the same, but which process setting(s) do we change?

- *Manager*: Keep the specifications the same, but adjust the process to use less energy and reduce emmissions, even if we use slightly more expensive materials, with different ratios. Is this possible?

- *Engineer*: A constraint has changed (e.g. a new government regulation, we have to use a different piece of equipment): how can we still get the same final product by adjusting the process conditions, or materials used in the process?

- *Financial controller*: We can buy raw ingredients from these 4 different suppliers. Which ones do we pick to most cost-effictively make the product, but still achieve the specifications?

- *Engineer 2*: Our current top line product is made with 6 different ingredients. Can we reduce this number down by adjusting the ratios or the choices of ingredients?

As these examples show: "product development" actually happens far more frequently than simply the case of a customer coming to ask for *different, entirely new, specifications* to those you currently have in your portfolio or product catalogue. The opposite case of changing these 3 things, in order to keep the *same specifications* is far more common:

- which ingredients (raw materials) do you use?

- which ingredient ratios, specified by mass fraction, do you use?

- which conditions do you implement to get the final product?

Both the case of creating an entirely new product, or improving an existing product can be considered with the methods described here.

The end goal is "faster development of personalized products and customer-centric development", using the information and databases we have accumulated over the many years of experience with the process.

7.2 Important concepts

7.2.1 What are the "Degrees of freedom"?

As just mentioned, there are 3 groups of things you can change:

1. **Select your ingredients**. This is a discrete choice: either you use an ingredient or raw materials, or you do not. It is a yes/no selection. You might have a whole catalogue, or database, of materials that you can select from. In many of the cases described in the "Usage examples" above this degree of freedom is actually fixed. In other words, you cannot change the ingredient choice and you must keep using what you already use. This is often due to regulations, or the fact that introducing new ingredients will be too expensive to test and validate and might lead to unexpected side-reactions or interactions.

2. **Adjust the ratios of the ingredients**. This is a sliding parameter: for example you can go from 45% weight fraction of material A, to 41% weight fraction, but remember by using less material A, the weight fractions of other materials change. The total weight fractions always add up to 1.0, so there is a constraint in the system, and adjusting one material will force the other material ratio to also be adjusted.

3. **Use different process conditions**. This group is where often you have the most degrees of freedom. You can adjust process settings used to make the product quite easily, such as temperature, pH, duration of certain steps, and order in which you add ingredients and complete the manufacturing steps. Because of the diversity of the options here, you might need to spend quite some time thinking about the process, and seeing what freedom you practically and economically have. Like the prior group, the ratios, this group of degrees of freedom also has some correlations in the historical data. For example, you might not be able to independently increase temperature in the process, without adjusting flow rate.

7.2.2 The "desired outcome"

This is a specification of what you want to achieve. Your end goal. It is often given as a vector of one or more specifications. For example: you might need to achieve a given viscosity, melting point and product density. These 3 numbers jointly defind the expectations.

Some entries in the desired outcome vector might simply be given as constraints. For example, "an *elongation* value of 15 or lower is acceptable, or a *shelf-life* of 30 days or greater is acceptable. This is more of a yes/no constraint: it is either met, or it is not. It creates a discontinuity in our system when we specify it as an equation later on. Discontinuities are often undesirable from a mathematical modelling and optmization perspective. However these can be dealt with by converting them to a smoothed version, such as by using a sigmoid function or a Gompertz function[165].

Finally, sometimes the desired outcome is a very large vector, such as time series showing the change of the product, such as elongation in a controlled experiments, or a pH over time. It can also be a spectrum, such as an NIR spectrum. The number of entries in this long vector are highly correlated. So the first step in such a situation is to use a *principal component model* (page 325) and understand the true lower dimensional space that the output space has. Then these, far smaller number of components, are used as a specification. Therefore the methods of product design are applicable in this case too.

7.2.3 The "rank"

Details to come in 2021, with illustrations and interactive case studies.

7.3 Process monitoring

Section will be added: showing an example with a Shewhart vs EWMA montoring chart.

7.4 Design of experiments

Some examples of DoE case studies and their creation, thought process, and eventual design.

---

[165] https://en.wikipedia.org/wiki/Gompertz_function