Practical Statistics for Geoscientists

Online Edition

2022

Original version by David Heslop

Updated by David Percy

Contents

Years of geological research and exploration using traditional methods have discovered a lot of relatively obvious theoretical principals and economic deposits; we have to use more sophisticated and sensitive techniques to uncover what remains!

Swan and Sandilands

1 Introduction to the course

In this introduction we'll take a brief look at the aims of the course and discuss practical issues such as software installation, recommended reading and course assessment.

1.1 Aims of the course

Our aims are pretty simple and of course they need to be limited because it is impossible to teach you all the statistics methods you will need in such a short course. During the course we'll address 3 main issues.

1. Helping you to think in a statistical manner.
2. Providing an introduction to statistical concepts.
3. Giving a basic introduction to a small number of statistical techniques.

The overarching aim of the course is more broad. Hopefully you'll see how useful statistics are and after the course you'll have the confidence to use statistics independently and apply the methods most appropriate for your research problems.

1.2 Recommended reading

If you simply go to the library and find a book called something like "An Introduction to Statistics" you'll probably discover that it's hundreds pages of impenetrable equations that don't seem to give you an introduction unless you're an expert already. Fortunately, some very good books have been written that

consider statistics in a geological context. In this way the situations and examples should be more familiar and you'll be able to see how statistics can be applied to the problems that you are working on. Of course there are also some very good statistics tutorials available for free on the internet, but it can take a long time to find them. I've included a recommended reading list at the end of the course notes that will hopefully guide you in the right direction. Some of the recommended textbooks can also be downloaded for free thanks to the ANU's library subscription and I've marked these in the reading list.

1.3 Course concept

The course is designed to be interactive, in terms of both discussing problems and computer exercises that will show you how certain statistical techniques can be applied to data. Throughout the text I've provided code snippets that show how a certain task can be performed and these can act as a template later on when you want to work on your own problems. Don't worry if you don't have any programming experience, we'll be starting right from the beginning and I've added lots of text describing what each line of the code does. Also, think about what you are doing, you won't learn anything by blindly copying the examples. Instead, try to see how each step in the analysis works and how it is related to the code in the examples.

One key point is not to work ahead of the material that is being discussed. We'll need to consider important points and if you're to understand them it means being involved with the discussion rather than typing away like crazy trying to complete all the examples. There are no prizes for finishing the examples first, so there's no point in trying to rush through them.

1.4 R, an environment for statistical computing and graphics

Throughout the course we'll be using the R programming language intensively. It's no problem if you haven't used R before, we'll start with very simple problems and gradually address more complicated tasks as we progress through the course. R provides a wide variety of statistical and graphical techniques and I selected it for this course for a number of reasons.

- It is an interpreted language, this means we can work step by step through a series of commands without the need for compiling complete sets of code.
- R is based around so-called *inbuilt* functions that provide quick access to thousands of different data processing methods.

- R is the official programming language of the statistics community, so for almost any statistical task you have, there will be R code available to perform it.
- R contains versatile graphics libraries allowing you to view your data in a variety of ways (visualization is an important part of statistics).
- R is FREE! It is also available for a variety of platforms (Windows, Mac, Linux).

1.4.1Installing R

If you don't have R installed then you just need to go to the home page:

http://www.r-project.org/ and download the appropriate version for your computer. You can then install R just like you would install any other piece of software. At certain points during the course we'll need to use special R *packages* to perform certain tasks. These packages can also be downloaded and installed using a single command in R (you'll need an active internet connection). Once you have R installed and you start the software you'll see a screen that looks something like the one in Figure 1.1

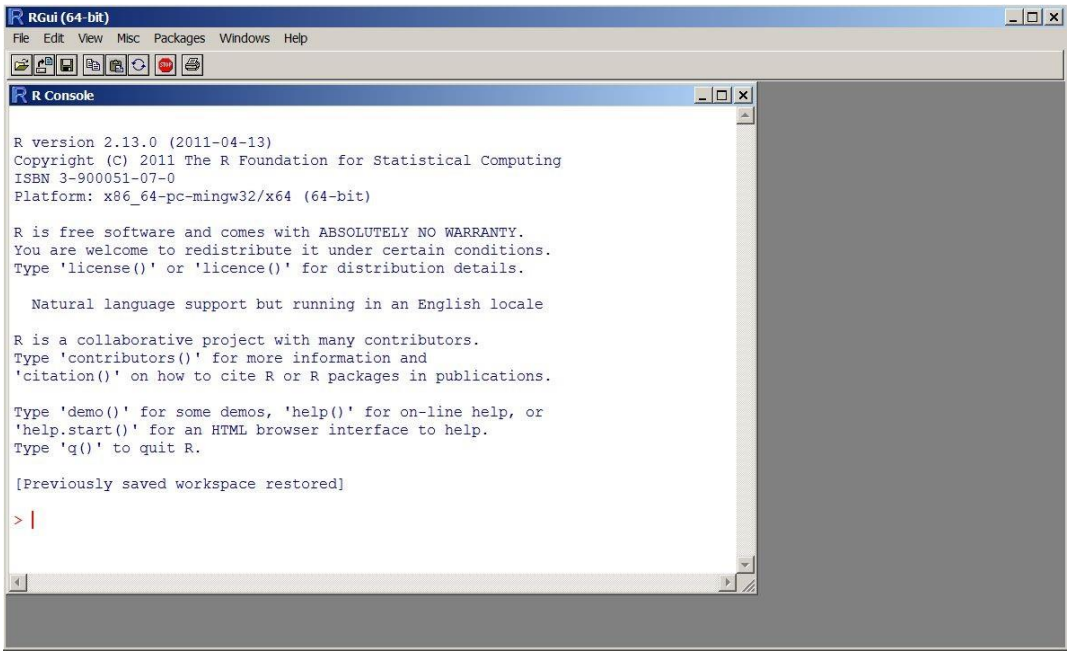


Figure 1.1: *A screenshot of the basic R installation. You can interact with R by typing commands at the > prompt in the console.*

1.4.2 RStudio

As you'll see, the R interface is pretty simple, which limits how efficiently you can interact with it. Rstudio provides a more complete way of interacting with R and allows scripts to be written and then executed, existing variables to be displayed on screen, etc (Figure 1.2). Download Rstudio from:

http://rstudio.org/download/

and getting it working on your system it will certainly enhance the useability of your R installation. It makes working with data much easier, as well.

Sometimes I'll also include the answers that R writes to the screen within the gray boxes. In such cases you can see that this is output rather than a command because the prompt symbol is missing. For example, repeating the calculation from above.

Example code: 2

```
> 1 + 1
[1] 2
```

Finally, for most of the commands I have also included *comments*. A comment is a piece of text that can be included in the command, but which will be ignored by R. This might seem a bit pointless, but comments provide a way in which to annotate commands and provide an explanation of what they are doing. The comment symbol in R is #, which means any text after the # will be ignored by R.

Example code: 3

```
> 1 + 1 # does 1 + 1 really equal 2, let's check
[1] 2
```

Therefore when you're typing in the commands from the examples the comments provide an explanation of what the command does, but you certainly don't need to type the comments in yourself.

1.4.4 Setting up R for the course

Finally we'll be using a number of example data sets during the course and we need to make sure that R knows where to find them. You should download the zip file that contains the data files (stored in the Rdata format) and extract the files to an appropriate folder on you system. Once you've started R you can simply click File

then Change dir … and then navigate to the folder in which you extracted the data files. Once this is set up, R will know where to look for the data files you want to load. If you would like to work with RStudio you need to perform a similar process. Go to Tools, Set Working Directory then Choose Directory and select the folder where you extracted the data files to.

The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data.

John Tukey

2 An introduction to statistical thinking

In this chapter we're going to look at a wide variety of topics that provide a foundation for the rest of the course. Hopefully I can justify to you that statistics are an essential tool for anyone studying the natural sciences and then we'll consider how we start to think in a statistical way about different situations and types of data.

2.1 Why do we need statistics

It is not uncommon to hear scientists say that they "hate" or "don't need" statistics. In such conversations it normally only takes a few minutes before a quote from the 19th Century British Prime Minister Benjamin Disraeli is uttered:

There are three kinds of lies: lies, damned lies, and statistics.

There is no doubt that statistics are used to lie to us on a regular basis, just think of the statistics quoted in the TV commercials of companies trying to sell us things. As scientists, however, as long as we use statistics properly, they form the most powerful tool we have to separate fact from fiction.

Some scientists will say that they can simply look at plots of their data and see all the information they need to make inferences and draw conclusions about the system they are studying. Our brains seem to be very good at spotting patterns in the real world and often we can convince ourselves that we can see a certain pattern in a given data set (normally the pattern that supports our preconceived ideas). In this brief introduction we'll look at a few historical examples to demonstrate how statistics are an essential tool to scientists and the mistakes that can be made when a proper statistical analysis is not undertaken.

2.1.1 Hunting for Ceres

At the start of the year 1801, the Italian astronomer Giuseppe Piazzi discovered a new planet positioned between Mars and Jupiter (we now know that the new eighth planet is actually a small asteroid, Figure 2.1). The plant was christened "Ceres" and astronomers started to track its position.

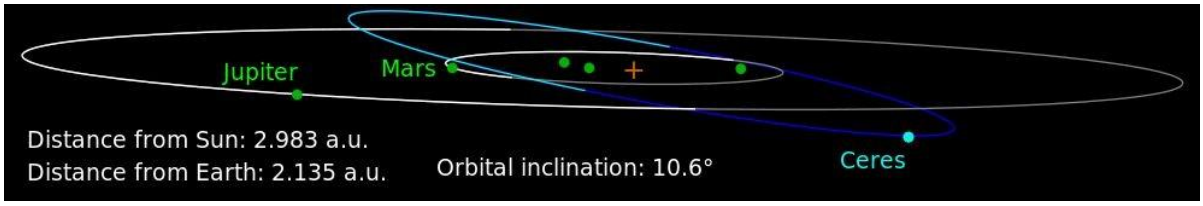


Figure 2.1: *The orbit of Ceres.*

After just 42 days Ceres disappeared behind the Sun when only 19 imprecise observations of its path had been made. Based on this scant amount of data Giuseppe Piazzi made predications of when and where Ceres would reappear out of the Sun's glare. However, Ceres didn't reappear where expected and the new planet was "lost". The 23 year old German, Carl Friedrich Gauss, heard of this problem and using statistical methods he had developed when he was just 18, extracted sufficient information from the existing observations to make a prediction of the position of Ceres based on Kepler's second law of planetary motion (Figure 2.2).

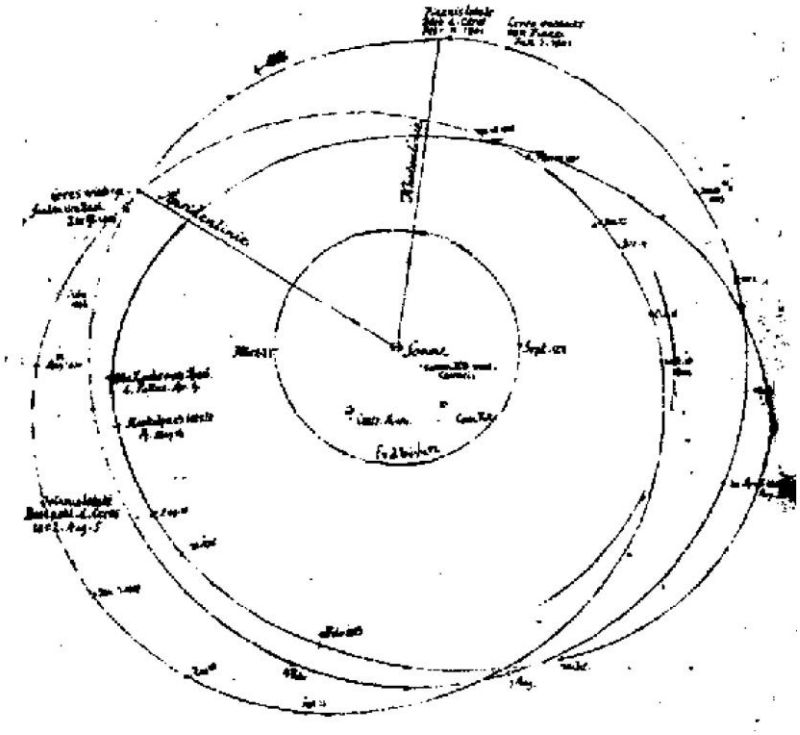


Figure 2.2: *A (very poor) reproduction of the sketch in Gauss' notebook that shows his calculated orbit for Ceres. The Sun is positioned in the center of the sketch and the orbit of Ceres is connected to the Sun by two straight-lines.*

The prediction Gauss made was close enough that Ceres was found again and it made him a scientific celebrity. Gauss' key insight was the observations of Ceres' motion would be *normally* (bell-shaped) distributed such that observations towards the middle of the distribution should be considered as more reliable than those towards the extremes. Maybe this idea seems obvious to us now, but at the time this statistical insight allowed Gauss to identify patterns in the Ceres data that everyone else had missed.

The normal distribution, also known as the *Gaussian* distribution, appears commonly in nature (hence the name) and we'll be looking at it in more detail later.

2.1.2 Tasting tea in Cambridge

At an afternoon tea party at Cambridge University in the late 1930's one of the guests announced that "Tea tastes different if you put the milk in first". The professors around the table picked up on the lady's comment and started to discuss how you could design an experiment to test this claim. Clearly they would need to prepare some tea without her looking and then test if she could correctly identify on taste alone which cups had been made with the milk in first.

When we think about this, designing the experiment is actually more difficult than it sounds. For example if we gave the lady just one cup of tea she has a 50% chance of guessing correctly even if she can't really tell the difference. Alternatively, if we prepared two cups of tea, one with the milk in first and one with the milk in last, the lady still has a 50% chance of getting the right answer by luck as long as she knows in advance that the two cups are different. Let's take another example, imagine we make up 10 different cups and the lady tastes them in a random order. If she only gets 9 out of 10 correct is that a sufficiently high number to conclude that she really can taste the difference? What if she had only got 7 out of 10, is that good enough to conclude that she can taste the difference?

The key advantage of a statistical approach is that we take what appears to be a subjective decision, e.g., is 9 out of 10 enough, and with a properly designed experiment we can reach an objective conclusion. Therefore, what any given investigator believes to be a sufficient number of correct identifications becomes unimportant and instead the statistics guide the criteria so that an objective decision can be made. In the experiment designed by the Cambridge professors the lady tasted 50 randomly order cups of tea and identified correctly wether the milk had been added first or last in all of them (a pretty convincing result).

2.1.3 Fisher at the Agricultural College

One of the people at the Cambridge tea party was Ronald Aylmer Fisher who made a number of very important contributions to a variety of topics in statistics. Early in his career Fisher had been contacted by Sir John Russell who was head of the Rothamsted Agricultural Experimental Station near London. For over 90 years the station had been running experiments on different types of fertilizers to see if they could improve crop yields. Unfortunately, the experiments had not been performed in a consistent manner, with each researcher designing their experiments without talking to the other researchers and developing their own methods for taking variables such as annual rainfall into account. This meant that although the Station had a wealth of data it was extremely difficult to extract a consistent picture from it. Fisher published a series of landmark works based on the Rothamsted data (given the catchy title "Studies in Crop Variation"), but it took one of the greatest statisticians who ever lived to work through the experiments and place them in a consistent statistical framework.

If you are not one of the greatest statisticians of all time then it's essential that you think about the statistical framework that you will need to analyze your results and design your experiments around that framework. If you simply perform a series of experiments and then decide you need to "do some statistics" you'll probably struggle just like the researchers at Rothamsted. To quote Fisher:

To consult the statistician after an experiment is finished is often merely to ask him to conduct a post-mortem examination. He can perhaps say what the experiment died of.

2.1.4 Do you need statistics?

As you might have guessed from the three examples above, I'm going to suggest that you do need statistics. I can't think of any field in the geosciences where, at the most basic level, information is stored in some form other than numbers. Therefore we have to be comfortable dealing with large (sometimes massive) numerical data sets.

As we saw in the example of Fisher at the Agricultural College, if we don't think in a statistical way we can easily end up wasting our time. Imagine that you suddenly discovered that your work had been in vain because rather than performing a proper statistical analysis you just formed conclusions on the basis of your own subjective analysis of the data. Our brains seem to be very good at spotting patterns, sometimes this can be useful, but often in the case of data analysis we can convince ourselves that certain patterns exist that really don't. There's not much we can do about this, it just seems to be the way our brains are wired, but to make advances in science and try to limit the mistakes we make, we should take an objective (i.e., statistical) approach rather than relying upon subjective intuition. To try to convince you how bad our intuition can be, I'm going to give two examples.

2.1.4.1 Professors examine data

David Warton, a statistician from the University of New South Wales ran an experiment to assess how well structures in data could be identified simply by visual inspection of plots rather than using detailed statistical methods. Plotted data sets were presented to 6 statistics lecturers and in less than 50% of the cases could they correctly identify certain structures that were known to exist in the data. This provides a good example of why we cannot simply rely on our intuition to interpret experimental data. We can run a similar experiment now. In Figure 2.3 you can see two collections of points, which ones are positioned randomly and which are not?

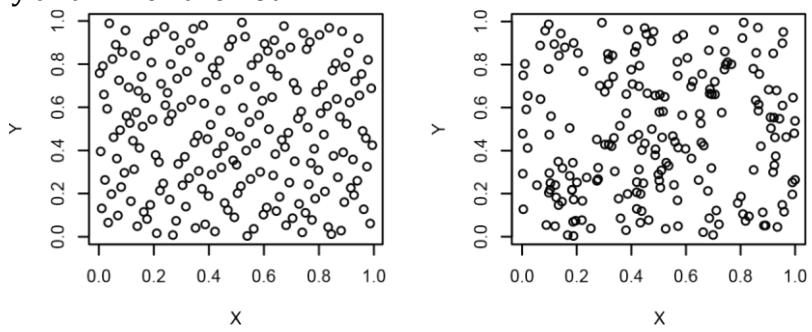


Figure 2.3: *Which set of data points are distributed randomly, the ones in the left panel or the ones on the right?*

2.1.4.2 The Monty Hall problem

The following is more of a problem in probability theory than statistics, but it does demonstrate just how far off our intuition can be when it comes to thinking about how even apparently simple systems work. The *Monty Hall Problem* was brought to fame when discussed in *Parade* magazine by the world's "most

intelligent" person Marilyn vos Savant (she has a quoted IQ of 228). The statement of the problem is as follows:

Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say A, and the host, who knows what's behind the doors, opens another door, say C, which has a goat. He then says to you, "Do you want to pick door B?" Is it to your advantage to switch your choice?

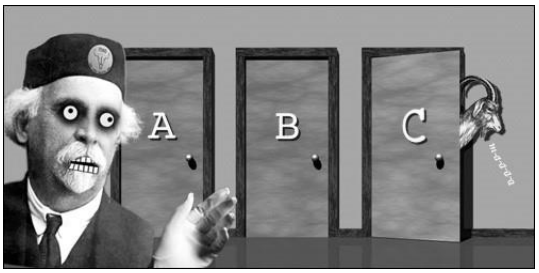


Figure 2.4: *In search of a new car, the player picks a door, say A. The game show host then opens one of the other doors, say C, to reveal a goat and offers to let the player pick door B instead of door A.*

So here's the big question, if you want to maximize your probability of winning a car rather than a goat, should you stick with the door you selected first, switch to the alterative door, or doesn't it make any difference which door you select?

2.2 Samples, populations and assumptions

One of the key aims of statistics is to make inferences about a population based upon the information contained in a sample of the population. To form these inferences it is essential to make assumptions about how both the sample and the population behave. In the next sections we'll look at a few examples to see how the relationship between samples, populations and assumptions works.

2.2.1 The Canberra taxi game with a twist

In Canberra each taxi is given a licence plate number "TX *some number*". This has led to a game whereby locals try to spot all of the taxis in numerical order. So you start by trying to spot taxi "TX 1", then some months later you may see "TX 2", then "TX 3", etc. Of course you have to spot them in sequence, if you see "TX 2" before you have seen "TX 1" then it doesn't count. We're going to play another game based on the numbers of taxi license plates that will (hopefully) provide some insights into how to think in a statistical manner.

Imagine I arrive in Canberra, go to a street and write down the numbers of the first 5 taxis that I spot. For example (once I've sorted them into numerical order):

73,179,280,405,440.

Now, on the basis of these numbers I want to address the question:

What is the total number of taxis (N) in the city?

Here we are using a *sample* (our 5 taxi numbers) to draw inferences concerning a *population* (all the taxis in the city). Clearly the problem cannot be solved with the provided information, however, we can make an *estimate* of *N* using the information we have, some necessary *assumptions*, and some simple statistics. The key point is that unless we are very lucky, our estimate of *N* will be incorrect. Don't think of statistics in terms of right or wrong answers, think of statistics in terms of better or worse answers. So what we are interested in is a good *estimate* of the total number of taxis.

If we just take the taxi numbers at face value we can see the problems with forming a good estimate. Given the numbers above, one person may say:

The highest taxi number is 440, therefore I can say that there must be at least 440 taxis in Canberra.

A reply to this estimate could be:

What if some taxi numbers in the sequence between 1 and 440 are missing (i.e., the taxis are not numbered in a continuous sequence), then there could be less than 440.

Or alternatively:

What if some taxis share the same number, then there could be more than 440.

An even more conservative estimate would be:

I've seen 5 taxis on the street therefore there must be at least 5 taxis in Canberra.

To which we could ask:

What if the driver of a taxi simply keeps driving around the block and each time gets out of their taxi and swaps the licence plate with another they have stored in the car? That way you could see 5 different licence plate numbers, but it would always be the same taxi.

Okay, this last argument might be unreasonable, but it does show that we can come up with arguments that will only allow us to estimate that the total number of taxis in Canberra is 1. It goes without saying that 1 is a very bad estimate of the population size. Therefore we can see that simply spotting 5 taxis doesn't help too much in estimating the size of the population unless we make some basic assumptions about how the taxi numbers behave. Such assumptions could be:

- The numbering system starts with Taxi 1.
- The numbering system runs in sequence with no missing numbers.
- No two taxis have the same number.
- Taxi numbers are distributed uniformly through the city.

These simple assumptions will form the basis of our assessment. It is important to realize that if our assumptions are incorrect or we've missed a key assumption then our estimate may be poor.

There are two simple ways we can estimate the number of taxis, the *median* estimate and the *extreme* estimate. In the median estimate we find the middle number in the sequence, which is 280, and calculate that the difference between the median value of the sample and the assumed first value of the population (taxi number 1) is 279. Therefore we can estimate that the difference between the median value and the largest taxi number in the population is 280+279 = 559 (Figure 2.5). Notice that this approach employs all of the assumptions listed above.

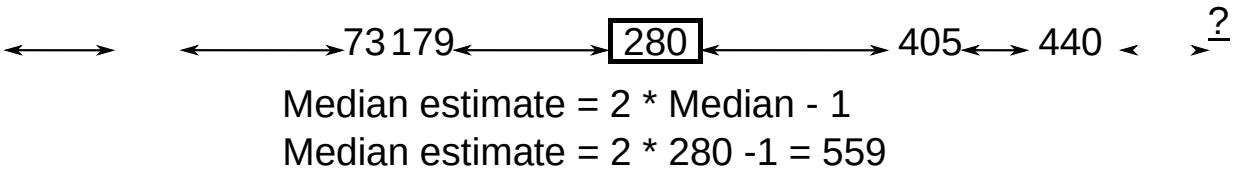
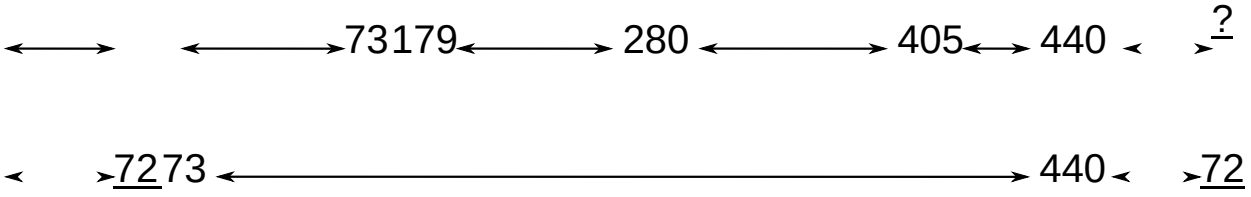


Figure 2.5: *Schematic diagram showing how the median estimate is formed for the taxi number problem.*

The extreme estimate looks at the edges of the data rather than the center. The lowest number in our sample is 73 therefore there is a difference of 72 between the first number in our sample and the assumed lowest number in the population (taxi number 1). We then look at the highest number in our sample and add 72 to make an estimate of the population size, so 440+72 = 512
(Figure 2.6).



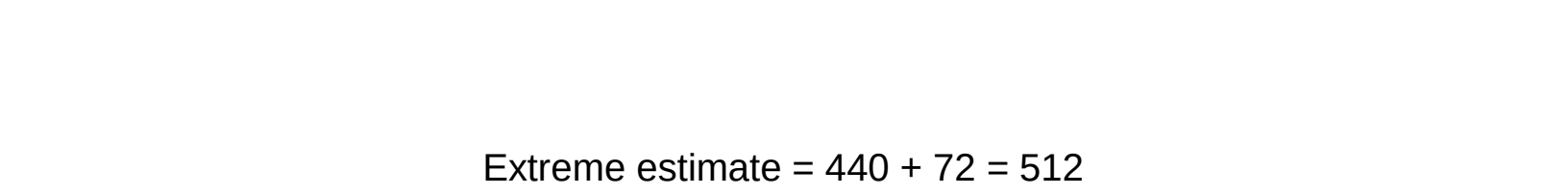


Figure 2.6: *Schematic diagram showing how the extreme estimate is formed for the taxi number problem.*

You can see that our two estimates are different and of course if we repeated the experiment we would collect 5 different numbers in our sample and obtain different estimates for the population. The key point, however, is that we are using a sample to draw inference concerning the population. To do this we make estimates that rely on assumptions. If we have bad assumptions then our estimate (unless we are very lucky) will also be bad. There may be more that one method with which to make an estimate and these methods cannot be expected to yield the same result (although you would hope that they are consistent).

It may seem that the taxi number problem is a trivial example and we could simply telephone the taxi company and ask them how many taxis they have. This was, however, an important problem in World War II when the Allies could make estimates of how many missiles the German forces had at their disposal. In order to keep track of their weapons the Germans painted sequential serial numbers on the outside of their missiles. I'm guessing that in this situation, Eisenhower couldn't telephone Hitler and ask him how many missiles he had, so instead soldiers were ordered to record the serial numbers of any German missiles that they captured. These numbers were returned for analysis and predictions of the total number of German missiles could be made. Out of interest, the median estimate outperforms the extreme estimate and it didn't take long before the Germans stopped painting sequential numbers on the sides of their missiles.

2.3 Representative and non-representative samples

In our Canberra taxi number problem it was quite simple to collect a sample of taxi licence plate numbers. The primary requirement of any sample is that it is *representative* of the population. For example we could collect taxi numbers on a street in the north of Canberra. If, however, taxis with numbers lower than 500 worked in the north of the city and taxis with numbers greater than 500 worked in the south, then clearly our sample would not be representative of the population of the taxis in the city. It would though be representative of the taxi numbers in the north of the city.

2.3.1 A historical example of a non-representative sample

In the run-up to the 1936 U.S. election the Literary Digest (a popular magazine) decided to undertake a poll to predict who would be the new president. The magazine selected 10 million people to include in the survey by picking from 3 lists of names; their own subscribers, registered car owners and telephone users. In total 2.4 million people responded to the poll, which is a large sample given that the total population at the time was only around 130 million. The results of the poll predicted a clear win for the Republican candidate, Alf Landon. On election day, however, the Democratic candidate Franklin D. Roosevelt won with a landslide victory. So why was the result of the Literary Digest poll so wrong?

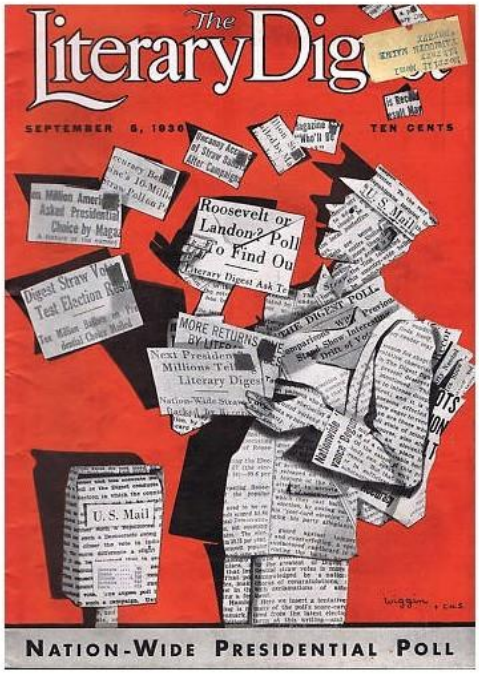


Figure 2.7: *The front cover of the issue of the Literary Digest in which they announced the result of the election poll.*

The simple answer is that the sample of people surveyed by the Literary Digest was not representative of the population as whole. The election took place during the Great Depression and lists of people who had magazine subscriptions and owned cars and telephones were biased towards the middle classes with higher than average incomes. In general, the middle classes favored the Republican Party, hence the result of the pole suggested a win for Alf Landon. This is a clear example of a nonrepresentative sample leading to a poor statistical estimate. Contrastingly, George Gallup performed a similar poll for the same election, which involved a much smaller sample size, but selected the voters to specifically obtain a demographically representative sample. On the basis of his poll, Gallup predicated the outcome of the election correctly.

2.3.2 Geological sampling

In the geosciences, collecting a representative sample can be challenging because our access to rocks, etc., is limited. The design of a sampling scheme should always be given careful thought and we can examine the relationship between the population and available samples in geological terms. We will only look at this briefly because sampling design is something that can change quite dramatically according to the specifics of a field and the questions being asked. One important point to stress is that throughout this course, when we use the word *sample* it implies a statistical sample rather than a geological sample. You'll notice that when we discuss geological samples I will use words like *specimen* to try to avoid any confusion.

The **Hypothetical Population** corresponds to the complete geological entity (Figure 2.8). In some cases the hypothetical population only exists in theory because parts may have been lost to erosion, etc. The **Available Population** represents the existing parts of the geological entity. Finally the **Accessible Population** is the material which can be collected to form a sample and therefore is used to represent the entity. Given that one of the main aims of statistics is to use statistical samples in order to draw conclusions concerning populations it is essential to consider if the accessible population is representative of the hypothetical and available populations.

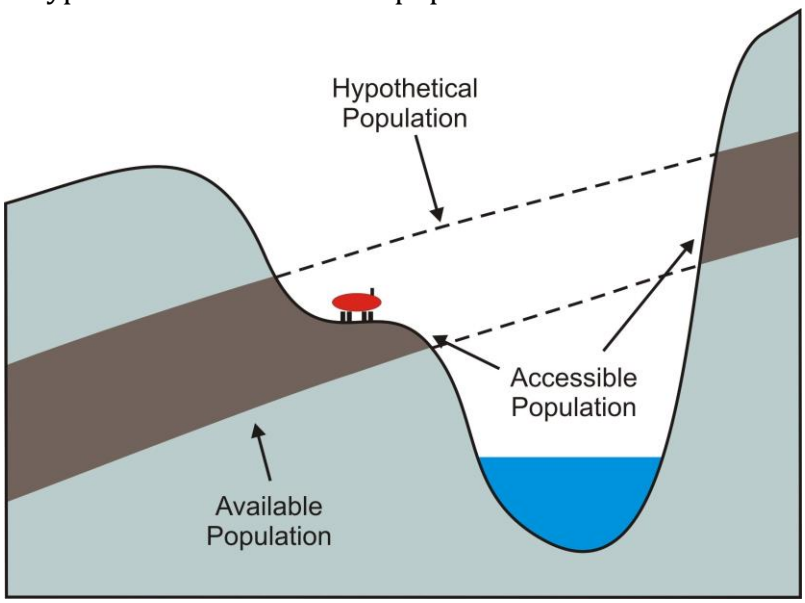


Figure 2.8: *A cross-section showing the different populations that can be considered in a geological scenario.*

2.3.3 A note on notation

Given that we will be using information in a sample to draw inferences concerning a population it is important to think about using a consistent notation. The system we'll employ is that where a parameter derived for a sample is represented with a letter, the corresponding parameter for the population will be given the equivalent Greek symbol. For example the standard deviation of a sample will be defined as s, so the standard deviation of the population will be σ. In this way is it easy to distinguish if we are referring to a sample or a population. There are a few exceptions in this system, the most common one is that for a sample, X, the mean of the sample is $\bar{X}$, but the mean of the population is μ. Hopefully such cases won't cause too many problems and the explanation given in the text should clear up any confusion.

2.4 Types of data

Geoscience data can come in a variety of different forms. As an example you might work with micropalaeontological data which could be expressed as:

- Presence or absence of a given taxon.
- Percentage data giving a relative abundance of the taxa.

- Concentrations of each taxon.

Certain statistical approaches can only be applied to specific forms of data. Therefore the type of statistical technique we must use will depend on the type of data that we have available. It is important to consider what type of data you have and what limitations that places on you. In this section we'll look quickly at some of the different types of data and the problems that may be associated with them.

2.4.1 Nominal or categorical data

This is data that cannot be ordered, for example, a list of fossil species from a limestone bed or the minerals identified in a thin section. Such information can be converted into a binary form of "presence versus absence". For example, the assemblage of benthic foraminifer taxa in a sediment could be represented by their presence or absence.

Taxon	Presence	Binary representation
Asterigerina	present	1
Archaias	present	1
Laevepeneroplis	absent	0
Textularia	present	1
Rosalina	absent	0
Miliolinella	present	1
Quinqueloculina	present	1
Triloculina	present	1

There are special statistical methods for nominal data, but we won't be dealing within them in this course.

2.4.2 Discrete data

This data can only be represented by integers, for example, the frequency of occurrence per space or time interval (Figure 2.9).

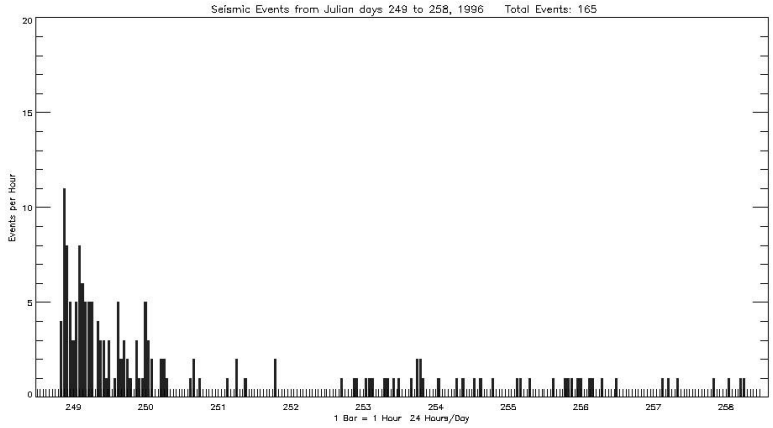


Figure 2.9: *The number of earthquakes per hour is a discrete data set (clearly you cannot have half an earthquake).*

2.4.3 Ordinal data

With ordinal data the values are used to denote a position within a sequence. This allows a qualitative, but not quantitative, rank order. A classical example of ordinal data is Mohs scale of mineral hardness.

Hardness	Mineral	Absolute Hardness
1	Talc	1
2	Gypsum	2
3	Calcite	9
4	Fluorite	21
5	Apatite	48
6	Orthoclase Feldspar	72
7	Quartz	100
8	Topaz	200
9	Corundum	400
10	Diamond	1500

We can see that corundum (9) is twice as hard as topaz (8), but diamond (10) is almost four times as hard as corundum. Therefore the hardness rankings in Mohs scale only denote a sequence but do not tell us about the absolute hardness.

2.4.4 Directional data

Directional data are normally expressed as angles, for example the flow direction of a lava. This means we need to consider data in terms of circles (in the 2D case) or spheres (3D case). We can see that even with something as simple as taking the mean of two directions we cannot apply statistics without careful thought (Figure 2.10).

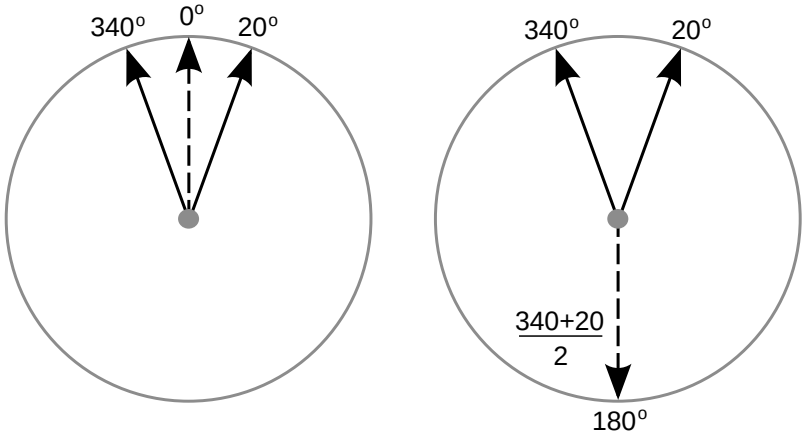


Figure 2.10: *An example of the problems associated with directional data. If we calculate the average of two angles using simple arithmetic, the value may not be the one we expect!*

2.4.5 Closed data

Closed data are normally in the form of percentages, parts per million, etc. The important point is that the values will always total to a constant, for example, 100%. A common example of closed data is sediment grain sizes that are split into sand, silt and clay fractions, which are then expressed as percentage abundance and plotted in a ternary diagram (Figure 2.11). Clearly the sand, silt and clay fractions for any given sample must add up to 100%. Closed data are surprisingly common in the geosciences and we'll be paying special attention to them in Chapter 9.

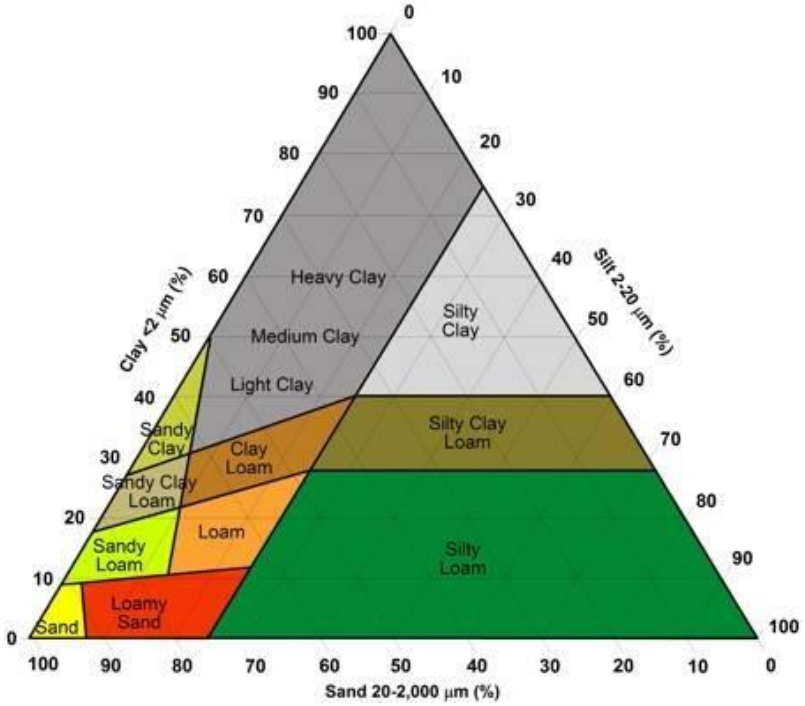


Figure 2.11: *A ternary diagram showing the classification of sediments depending on their composition of sand, silt and clay. Notice that all positions in the diagram correspond to compositions that add to 100%.*

2.4.6 Interval scale data

These are data relating to a scale, where the zero point does not represent the fundamental termination of the scale (Figure 2.12).

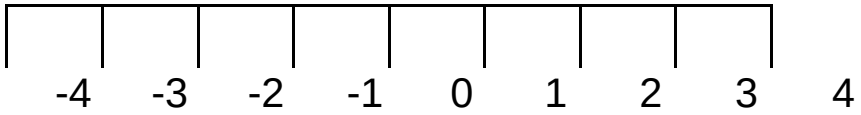


Figure 2.12: *An example of an interval scale. Notice that the values are equally spaced but negative values are also possible because zero does not represent the end of the scale.*

The classic example of interval scale data is the Celsius temperature scale. Ask yourself the question:

It was 0°C today, but tomorrow it will be twice as warm. What will the temperature be tomorrow?

This demonstrates that calculations such as ratios are meaningless for interval scale data.

2.4.7 Ratio scale data

Ratio scale data is the best form of data for statistical analysis. The data is continuous, the zero point is fundamentally meaningful, and as the name suggests, ratios are meaningful. An example of ratio scale data is the Kelvin temperature scale. Ask yourself the question:

It was 273 K today, but tomorrow it will be twice as warm. What will the temperature be tomorrow?

Length is also a form of ratio scale data, so if a fossil is 1.5 meters long, how long would a fossil be that is half the size?

Jetzt stehen die Chancen 50:50 oder sogar 60:60.
(Now our chances are 50:50, if not even 60:60)

Reiner Calmund (German football coach)

3 Statistics and probabilities

Statistics and probability are very closely related. A key part of statistics is the examination and use of so-called *probability distributions* which provide a tool with which to make interferences concerning a population based on the information contained in a sample. In this section we'll look at probability distributions in a general way and see what information they can and can't give us.

3.1 Discrete probability distributions

We'll start with a somewhat formal definition of discrete probability distributions and then look at some practical examples:

A discrete random variable takes on various values of x with probabilities specified by its probability distribution p(*x*)*.*

Consider what happens when we roll a fair 6 sided die, what is the probability that we will throw a given number? Because the die is fair we know that there is an equal probability that it will land on any of its 6 sides, so the probability is simply 1/6 = 0.167. We can represent this information with an appropriate discrete probability distribution (Figure 3.1).

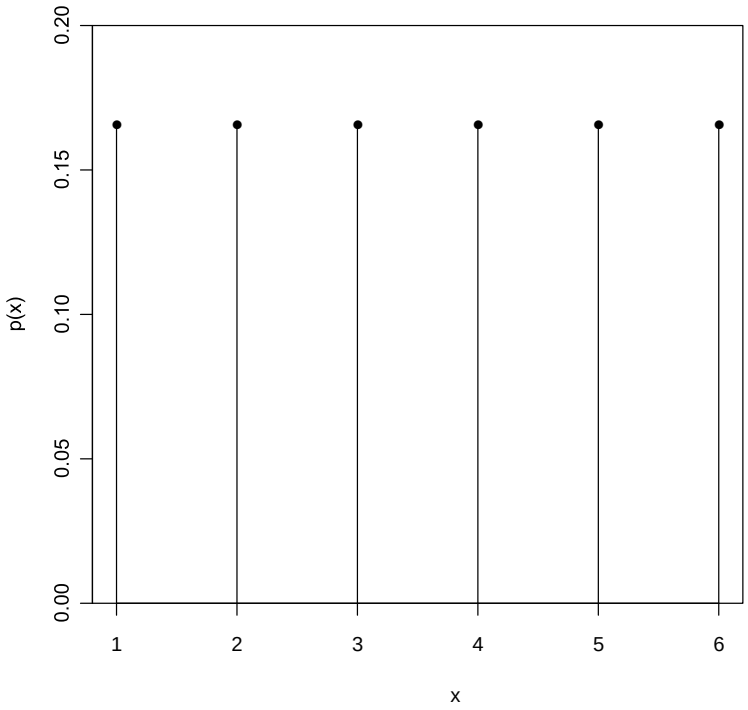


Figure 3.1: *The discrete probability distribution describing the probability, p, of obtaining a given value, x, when a fair die is thrown once.*

The probability distribution looks exactly like we would expect, with the chance of throwing each number having the same probability of 1/6. The use of the word "discrete" tells us that only certain results are allowed, for example, we cannot consider the probability of throwing a value of 2.5 because it is clearly impossible given the system we are studying. The discrete nature of the system is demonstrated in the distribution, with all values except the allowable results of 1, 2, 3, 4, 5 and 6 have a probability of zero.

Rolling a single die once is a simple case, what about if we roll two dice and add up their values? There are 11 possible totals between 2 (rolling two ones) and 12 (rolling two sixes) and the probability distribution is shown in Figure 3.2.

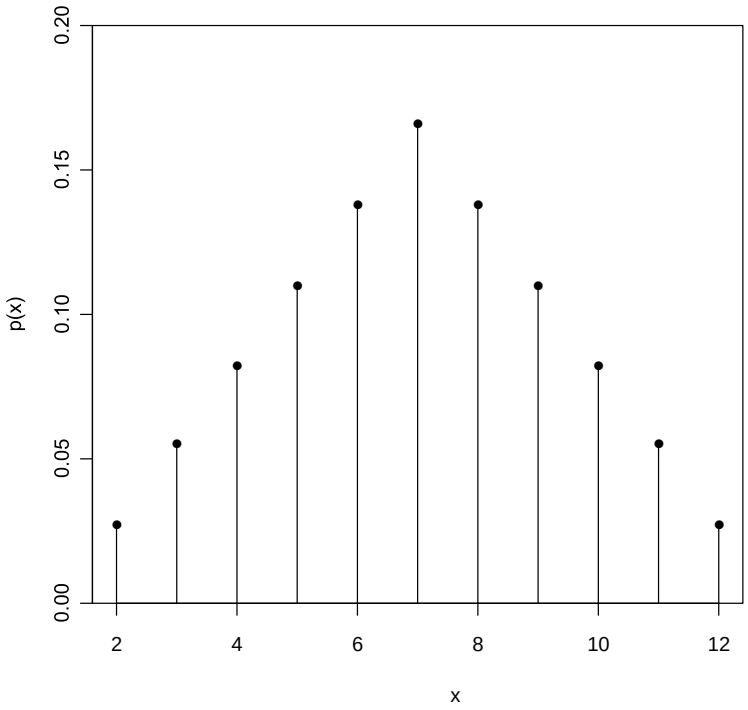


Figure 3.2: *The discrete probability distribution describing the probability, p, of obtaining a given total, x, when 2 fair dice are thrown.*

Let's look at the kind of information the probability distribution for rolling two dice and summing their values can provide us with. For example, we are most likely to throw a total of 7, which has a probability of 0.167, whilst the chance of throwing a total of 2 is under 0.03 (in other words less than 3%). We can also combine probabilities, for example, the probability of throwing a total of 4 or less is just the probability of throwing a total of 4 plus the probability of throwing a total of 3 plus the probability of throwing a total of 2 (0.083+0.056+0.028 = 0.167). The chance of throwing a total of 5 or a total of 7 is the probability of throwing a total of 5 plus the probability of throwing a total of 7 (0.111 + 0.167 = 0.278). Of course if we summed the probabilities of all the different possible outcomes they would equal 1 because we know that for any given trial one of the allowable outcomes has to occur.

3.2 Continuous probability distributions

As we have just seen, discrete probability distributions are appropriate when the outcome of a certain trial, for example rolling a die, can only take certain values. In the case of continuous probability distributions the result of the trial can take any value. The classic example of a continuous probability distribution is the bell-shaped normal (sometimes called Gaussian) distribution. Normal distributions are characterized by a mean (μ), which defines the position of their center, and a standard deviation (σ) that controls their width. We find that in the real world many quantities are normally distributed (hence the name), as an example we'll look at intelligence quotient (IQ). The way the results of modern IQ tests are structured is to yield a normal distribution with a mean of 100 and a standard deviation of 15 (Figure 3.3).

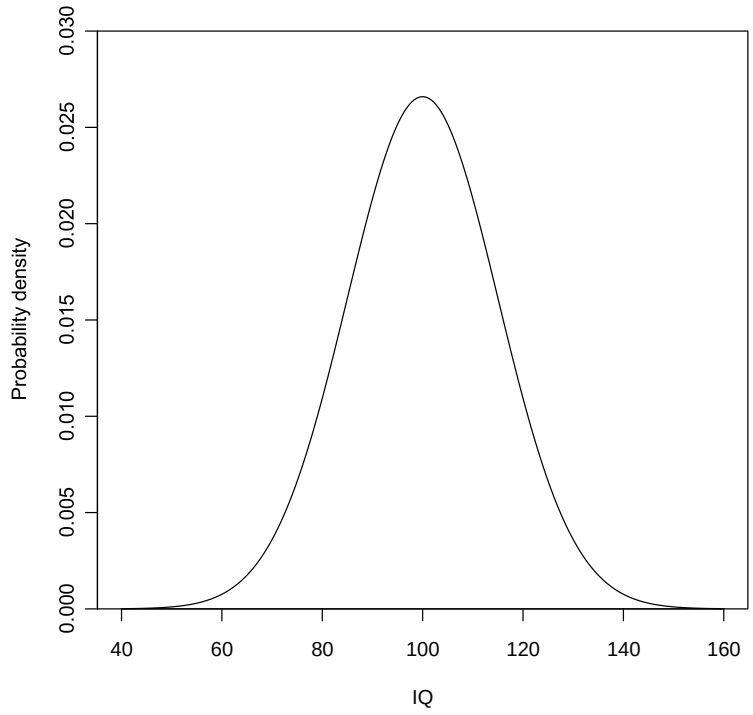


Figure 3.3: *The normal continuous probability distribution describing IQ scores that have a mean of 100 and a standard deviation of 15.*

The first thing to notice is that the distribution is symmetrical about its center, which is positioned on the mean value of 100. The width of the distribution is controlled by the standard deviation, if we used a larger standard deviation the distribution would be wider and lower and if we had used a smaller standard deviation it would be narrower and higher (Figure 3.4).

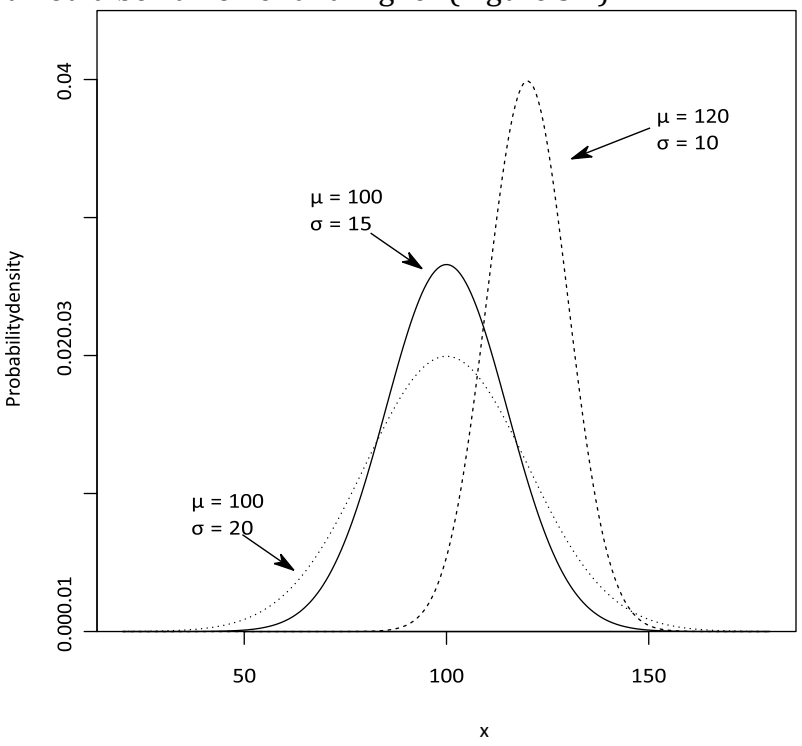


Figure 3.4: *Examples of normal distributions with different means (μ) and standard deviations (σ).*

There are some important differences in how we must interpret discrete and continuous distributions. For example, it is not as simple to answer the question "what it the probability of a candidate scoring 100 on a IQ test" as it may seem. If we used the same approach as we did for discrete distributions we would simply read off the *y*-axis value at an IQ of 100 and quote that as a probability. However, if we assume that the IQ score can take any value (i.e., there are an infinite number of possible test scores), then the probability of obtaining a given score exactly is zero. We can however make statements concerning probabilities if we consider ranges of values, for example, what is the probability that a randomly selected candidate will score between 80 and 110 points. By definition the integral of a continuous distribution is 1 and if we simply integrate the distribution between 80 and 110 we will obtain the probability of a score in that interval (Figure 3.5). This is the reason why continuous probability distributions are expressed in terms of *probability densities* (see the *y*-axis of Figure 3.3) rather than straight probabilities as in the discrete case. If we do this for the interval [80,110] we find the probability is ~0.66.

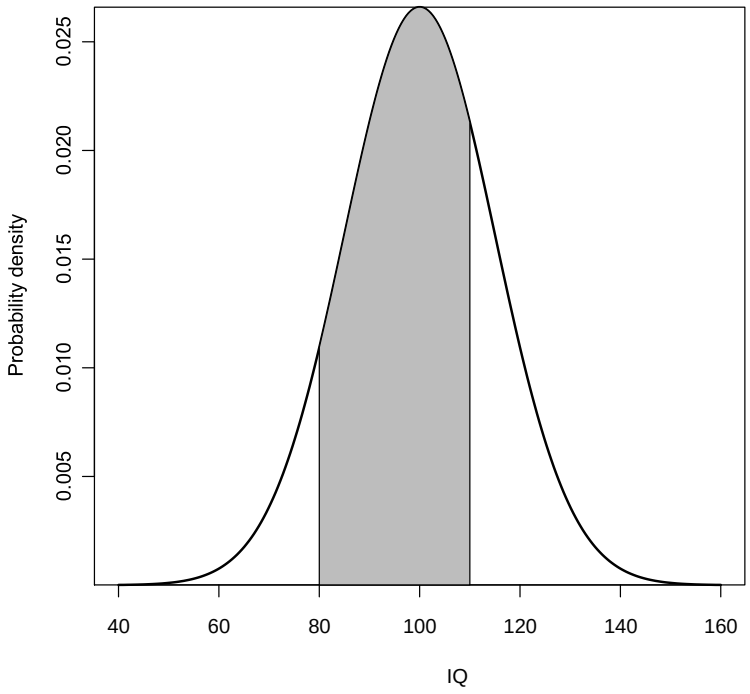


Figure 3.5: *The probability of a random candidate obtaining an IQ score of between 80 and 110 can be found by integration of the corresponding interval of the normal distribution (shaded region).*

We can also determine the probability of scoring more or less than a given value. Marilyn vos Savant (developer of the Monty Hall problem) has a quoted IQ of 228. This leads to an obvious question; what proportion of people will have an IQ of 228 or higher. We use the same procedure as above and simply integrate a normal distribution with a mean of 100 and a standard deviation of 15 in the interval [228,∞], Figure 3.6.

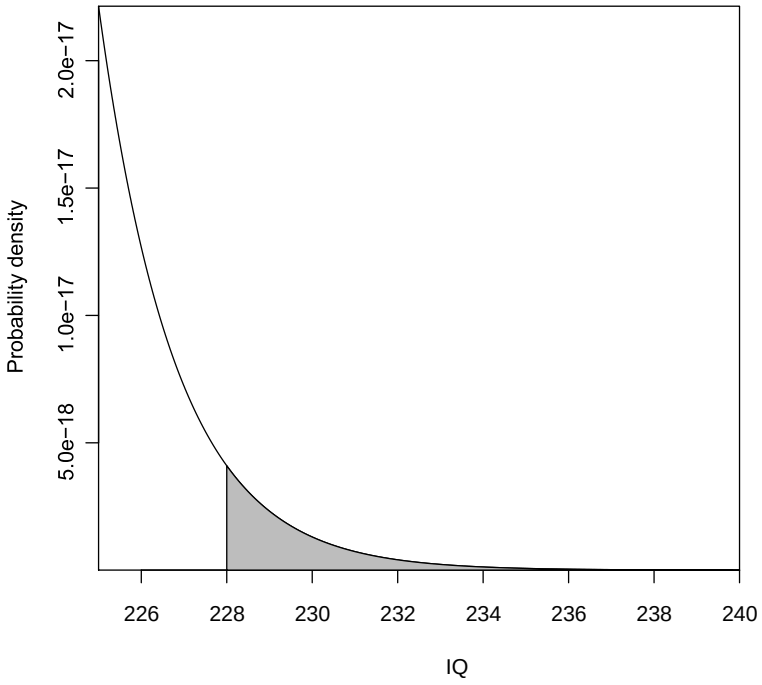


Figure 3.6: *The probability of a random candidate obtaining an IQ score of 228 or higher can be found by integration of the corresponding interval of the normal distribution (n.b. the shaded region extends to +∞).*

We can see that the probability densities associated with this part of the distribution are very low and the number of people expected to have an IQ of 228 or higher corresponds to less than 1 person in every 1000 trillion. How can we interpret this result, does it mean that Marilyn vos Savant is the cleverest person who will ever live or maybe we can take it as evidence that she lied about her IQ? Well the answer is probably simpler. The normal distribution provides a model of how the IQ scores are expected to be distributed, but it is certainly not a perfect model. We can expect it to perform well in the region where we find the majority of the cases (the center), but as we head out to the extremes of the distribution, called the *tails*, it will perform poorly. Therefore we must take the probabilities associated with Marilyn vos Savant's IQ with a pinch of salt.

To demonstrate this point concerning the tails of distribution, ask yourself what is the probability of someone having an IQ of 0 or less? Clearly it's not possible to have a negative IQ, but if we take a normal distribution with a mean of 100 and a SD of 15 and integrate the interval $[-\infty,0]$ we find the probability according to the model is 1.3×10^{-11}, which admittedly is very low, but is clearly not zero (Figure 3.7).

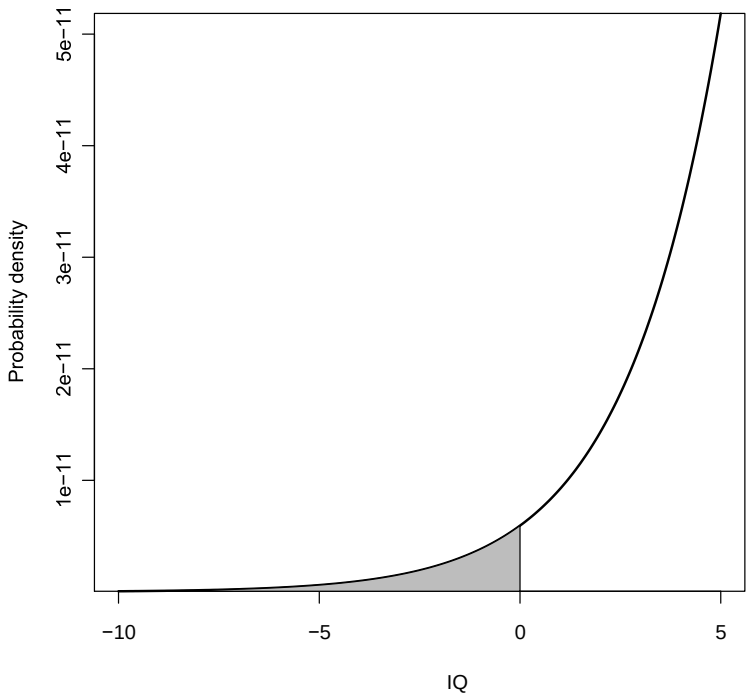


Figure 3.7: *The probability of a random candidate obtaining an IQ score of 0 or lower can be found by integration of the corresponding interval of the normal distribution (n.b. the shaded region extends to -∞).*

3.2.0.1 Different forms of probability distribution

In the previous section we focused on the normal distribution because it is simple to understand and many statistical methods assume that data are normally distributed. It is important to point out, however, that a vast array of different distributions exist that represent a wide variety of systems and processes (we'll meet some of them later on). To give an example of a different type of distribution the grain sizes of the particles in a sediment typically follow a log-normal distribution. In a log-normal distribution the values, x, are not normally distributed, but instead the values $\log(x)$ are normally distributed (Figure 3.8).

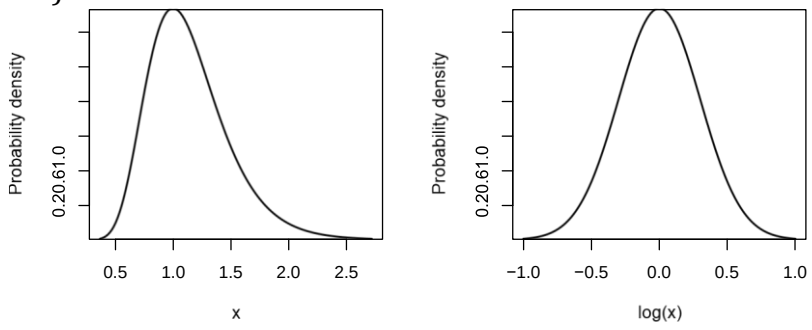


Figure 3.8: *An example of a log-normal distribution (left). The log-normal distribution becomes a normal distribution when expressed in terms of logarithms (right)*

3.2.1 An example: the confidence interval on a mean

As mentioned above, probability distributions play a key role in allowing us to make inferences concerning a population on the basis of the information contained in a sample. At this stage in the course you'll just need to accept the steps we're about to take without worrying why they work, the key point is to demonstrate how we can use probability distributions to make inferences.

Returning to our example of IQ scores, imagine that I don't know what the average IQ of the population is (remember it's defined as a population mean $\mu = 100$ with a population standard deviation of $\sigma = 15$), so I'm going to estimate it using a statistical sample. I choose 10 people at random and obtain their IQ scores to form my sample, X. The scores in the sample are as follows: 107.9,106.8,88.1,100.8,94.4,99.0,84.4,110.9,110.0,85.7

Based on this sample I wish to estimate the mean of the population, μ. Of course it's easy to find the mean of the sample, $\bar{X}$, simply by adding the values in X and dividing by the number of values, n (10 in this case).

$$\bar{X} = \frac{\sum X}{n} \tag{3.1}$$

For my sample $\bar{X} = 98.8$, which is close to, but not exactly the same as, the true value of 100. Of course it's not surprising that the mean of the sample is not exactly the same as the mean of the population, it is after all just a sample. The key step is to use the information in the sample to draw inferences concerning the mean of the population. Specifically we want to define a confidence interval, so that we can make the statement:

$$\mu = \bar{X} \pm \text{sampling error} \tag{3.2}$$

In this way we won't be able to make a definite statement about the precise value of the population mean, but instead we'll be able to say with a specific probability that μ lies in a certain interval (based on the sampling error).

To find the confidence interval let's imagine that we repeated the above experiment an infinite number of times, collecting a new sample of scores from 10 different people and calculating a new value of $\bar{X}$ each time. If the values in X come from a normal distribution (which we know they do) the collection of infinity $\bar{X}$ values would be normally distributed with a mean of μ and a standard deviation of $\sigma/\sqrt{n}$, where n is still the size of each sample. This is a so-called *sampling distribution* and it is shown in Figure 3.9.

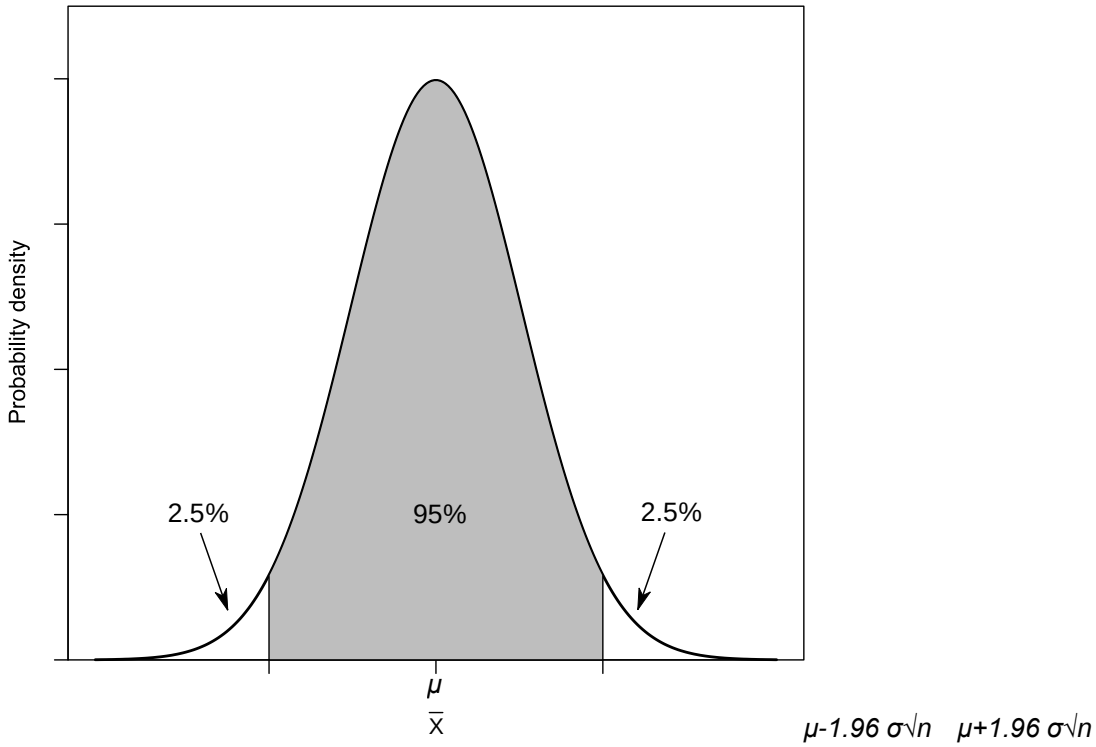


Figure 3.9: *Distribution of sample means ($\bar{X}$). The central 95% of the distribution provides the basis for estimating a confidence interval for the population mean.*

Of course I can't take an infinite number of samples, but the sampling distribution provides me with a model with which to estimate the population mean within a confidence interval.

Examination of the sampling distribution shows that 95% of the samples should yield values of $\bar{X}$ that lie within the interval $\mu \pm 1.96\sigma/\sqrt{n}$. So there is a 95% chance that our original sample with $\bar{X}$ = 98.8 falls into this interval. This can be written more formally as:

$$\Pr(\mu - 1.96\sigma/\sqrt{n} < \bar{X} < \mu + 1.96\sigma/\sqrt{n}) = 95\% \quad (3.3)$$

With a bit of rearrangement the inequality can be written to solve for μ:

$$\Pr(\bar{X} - 1.96\sigma/\sqrt{n} < \mu < \bar{X} + 1.96\sigma/\sqrt{n}) = 95\% \quad (3.4)$$

therefore we can say with a 95% probability that μ lies in the interval $\bar{X} \pm 1.96\sigma/\sqrt{n}$, but we still have a problem, we don't know σ. The standard deviation, s, of our sample, X, will provide an estimate of the population standard deviation, σ. However, in the same way that $\bar{X}$ was not exactly equal to μ, we cannot expect s to be exactly equal to σ. This unreliability in s as an estimate of σ means that we must widen the interval on μ slightly in order to retain 95% confidence. So rather than using the value of 1.96 corresponding to the central 95% of the sampling distribution in Figure 3.9, we have to look at the central 95% of a distribution called Student's t-distribution (Figure 3.10). We can now rewrite our inequality to include s rather than σ:

$$\bar{X} \pm 1.96\sigma/\sqrt{n}$$

$$\Pr(\bar{X} - 2.26s/\sqrt{n} < \mu < \bar{X} + 2.26s/\sqrt{n}) = 95\% \quad (3.5)$$

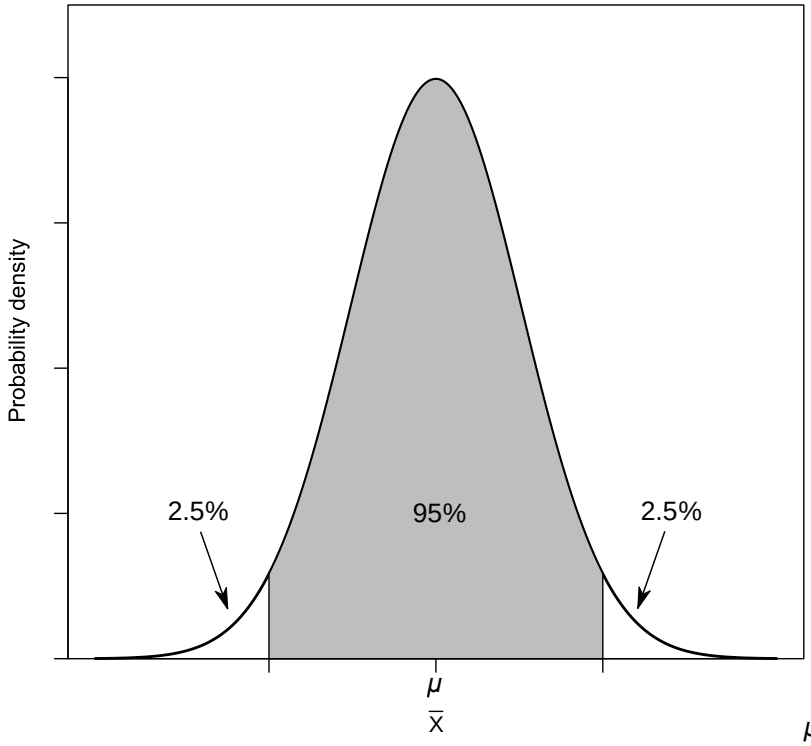


Figure 3.10: *Student's t-distribution allows the uncertainty in s as an estimator of σ to be included in the determined confidence interval on the population mean, μ.*

We now have all the information we need to estimate the 95% confidence interval on μ:

$\bar{X} = 98.$ 8 the mean of the sample

$s = 10.$ 20 the standard deviation of the sample

n = 10 the size of the sample

$98.8 - 2.26\frac{10.2}{\sqrt{10}} = 91.$ 5 lower bound of the 95% confidence interval 1

$98.8 + 2.26\frac{10.2}{\sqrt{10}} = 106.$ upper bound of the 95% confidence interval

So we can say with 95% confidence that μ lies in the interval [91.5,106.1]. In this way we used the information in a sample to make inferences about a population using probability distributions as a model to link the sample and the population. We'll be using probability distributions in a similar manner later on.

By a small sample, we may judge of the whole piece.

Miguel de Cervantes from Don Quixote

4 Hypothesis testing

We'll start by considering the very simple example of flipping a coin which can either land *heads* or *tails* side up. If your coin is "fair", in other words it is equally likely to land on heads or tails when flipped, what is the probability that you will flip a tail? The answer is pretty obvious, if we have two possibilities and they carry an equal probability then:

$$p = \frac{1}{2} \tag{4.1}$$

So let's try this for real, flip a coin 10 times and from your sample of results calculate the probability that you will flip a tail (in other words count the number of times you flipped a tail and divide it by the total number of flips, which is 10). We know that a fair coin should yield p = 0.5, so if you didn't flip 5 tails is it safe to assume that your coin is not fair? Of course it's possible that your coin gave a result close to 0.5, maybe 0.4 or 0.6, so what do you consider to the acceptable range of p values from your experiment in which the coin can still be considered to be fair?

The primary problem with this question is that the answers we give are subjective. One person may say the you must achieve p = 0.5 exactly, whilst a more relaxed person may consider anything in the interval [0.3,0.7] to be okay. As scientists we want to avoid these kind of subjective choices because they mean that two different people can make two different judgements based on the same data sets. Statistics help us to interpret the data in an objective manner and thus remove the effects of our personal beliefs (which as we saw in the Monty Hall problem can be very much in error).

This problem is similar to the experimental design to test if the lady drinking tea at an garden party in Cambridge can really tell the difference if you put the milk in the cup first. How many cups of tea should she drink and what proportion should she get correct before you can conclude that she can really can tell the difference in taste. Let's go back to our coin flipping experiment. Imagine we flip a coin 100 times and after each flip calculate the current probability of a tails by dividing the number of tails obtained at that point by the current number of flips. I did this and the results are shown in Figure 4.1.

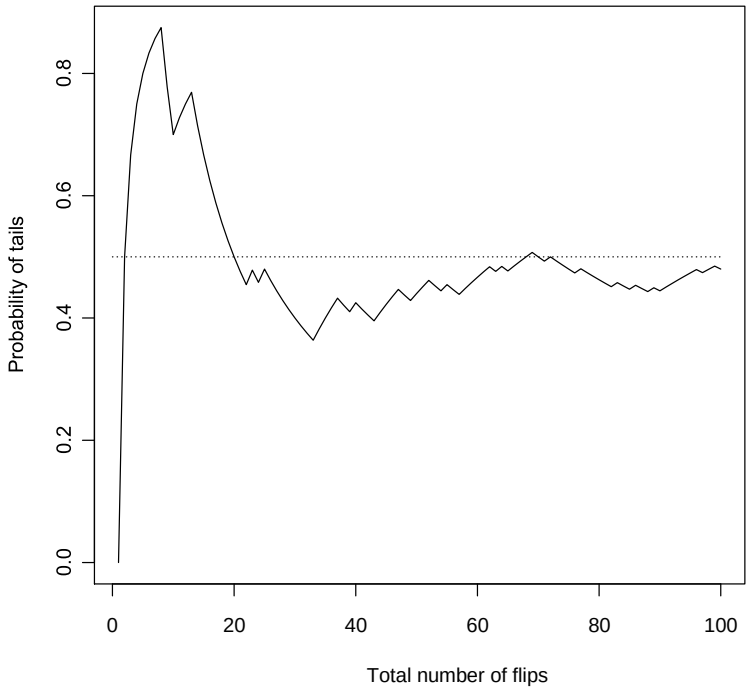


Figure 4.1: *The results of flipping a coin 100 times. After each flip the probability of flipping a tail is calculated based on the data so far. The dashed horizontal line shows p = 0.5, which is the probability expected for a fair coin.*

We can see from the results of our coin flips that the experimental probability gets close to the expected value of $p = 0.5$, but even after 100 flips we're not exactly at 0.5, so can the coin be judged to be fair?

Clearly we can't say that a coin is only fair if it gives $p = 0.5$ exactly. This would mean that every time we repeat the 100 coin flips we would always need to get 50 heads and 50 tails. Instead, we have to decide what is the acceptable range of p values in which the coin can still be considered fair and what that range depends on (for example, the total number of flips included in the experiment). To make these kind of decisions we will employ *hypothesis testing*.

4.1 Hypotheses and hypothesis testing

Lets start with a somewhat formal definition of a hypothesis, which is;

> *A tentative assumption made in order to draw out and test its logical or empirical consequences.*

To test a hypothesis we need to state two hypotheses:

- **Null Hypothesis** (H_0): the proposition.
- **Alternative Hypothesis** (H_1): if H_0 is doubtful, what does that imply.

Lets apply these definitions to the coin flipping experiment we examined above. If we want to perform a hypothesis test to judge if our coin is fair we need to state the null and alternative hypotheses:

- **Null Hypothesis** (H_0): the coin is fair.
- **Alternative Hypothesis** (H_1): the coin is not fair.

A hypothesis test allows us to evaluate the possibility of H_0 given the available experimental data. If H_0 does not appear to be very likely on the basis of the data, then we must reject H_0 and instead accept H_1. For example if we flipped our coin 100 times and obtained 100 tails we would feel pretty safe in rejecting the null hypothesis; *the coin is fair*, instead accepting the alternative hypothesis; *the coin is not fair*.

How we could go about testing the null hypothesis for our coin flipping experiment? We want to test if our coin is fair, so lets consider how a perfectly fair coin would behave. We actually studied a similar problem in Section 3.1 when we were studying discrete probability distributions. If we repeated the coin flipping experiment with a total of 10 flips a number of times we would obtain a distribution of results that would describe the probability of any given result. For example, if my coin is fair, what proportion of the experiments would yield 1 tail and 9 heads? Using the *binomial distribution* we can find what the distribution of results would look like for our coin tossing experiment if we repeated it an infinite number of times (i.e., a perfect representation of the system). The binomial distribution representing the 10 flip experiment for a fair coin is shown in Figure 4.2.

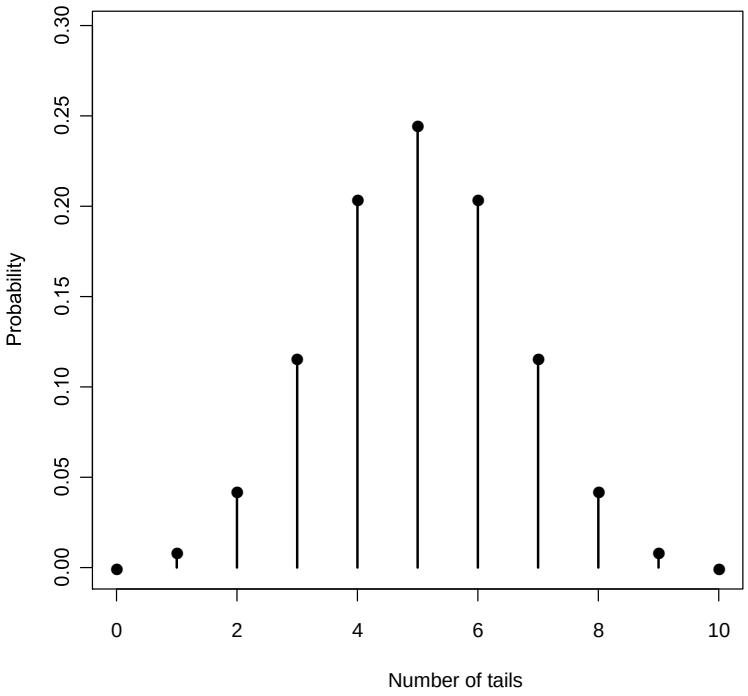


Figure 4.2: *The distribution of results for a fair coin flipped a total of 10 times. As expected most trials give a result around the 5 tails region, thus this region of the distribution is associated with high probabilities. We should also expect, however, to occasionally see extreme results with low probabilities, for example 10 tails out of 10 (which carries a probability of approximately 0.001).*

Now we'll use the binomial distribution to look at the probabilities for an experiment including 100 flips and how many tails we can expect to get in any given trial (Figure 4.3).

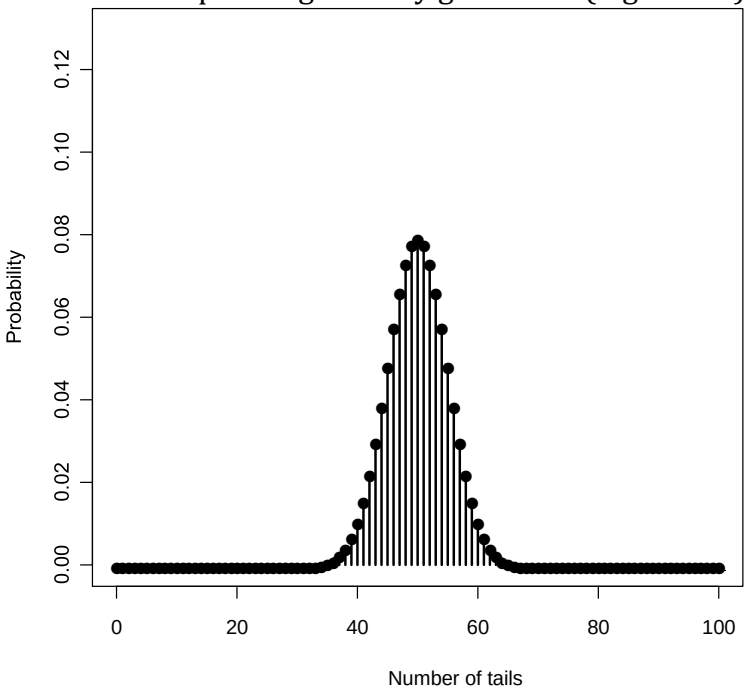


Figure 4.3: *The distribution of results for a fair coin flipped a total of 100 times. As expected most trials yield a result around the 50 tails region, thus this region of the distribution is associated with high probabilities.*

As before, extreme results have low probabilities, for example, we should only expect to observe a trial that produces 100 tails out of 100 in about 1 in every 10^{30} trials. That means if you had started flipping coins at the birth of the Universe and completed an 100 flip trial every 15 minutes you probably still wouldn't have got 100 tails out of 100 yet. This might seem like a stupid statement to make, but what it shows us is that for a truly fair coin, results at the extremes (*i.e.,* the extremes of the distribution) are very unlikely and results towards the center of the distribution are much more likely. We can use this information to test the hypothesis that a given coin is fair. If the result of our 100 flips experiment falls into a low probability region of the distribution we know that the chance of getting such a result for a truly fair coin is low, which suggests that our coin may not in fact be fair.

Let's look at our binomial distribution for 100 flips of a fair coin again, with a specific focus on the extremes (the tails) of the distribution. If we add up the probabilities of the results (as we did in Section 3.1) we find there is only a 5% chance that an experiment will result in 59 or more tails and a 5% chance that my experiment will result in 41 or less tails (Figure 4.4). This tells us that for an experiment consisting of 100 flips of a fair coin we would expect to get between 42 and 58 tails in 90% of the cases. If our coin is unfair, however, we should get an unexpectedly low or high number of tails, that doesn't fit with the probabilities expected from the binomial distribution.

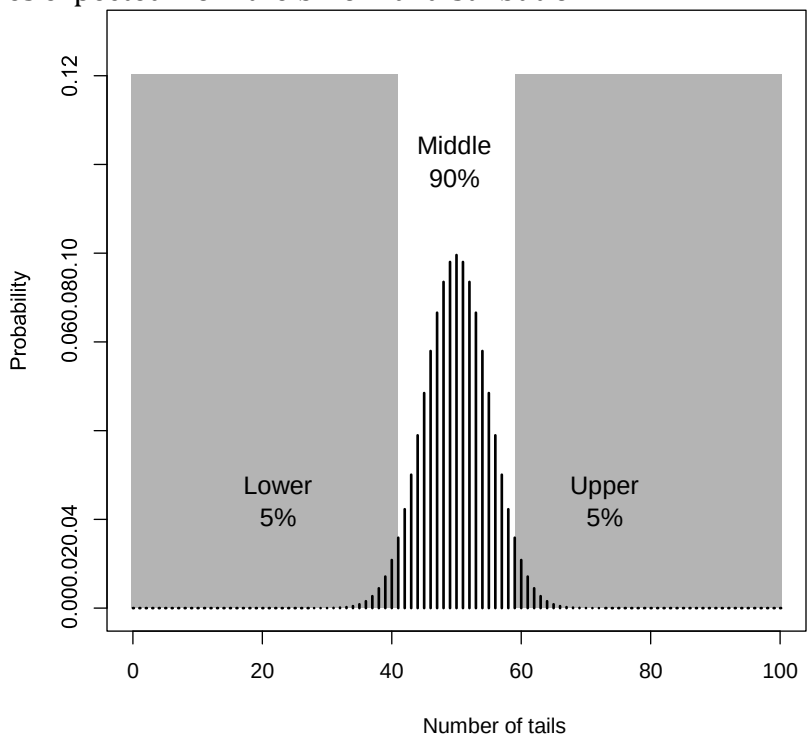


Figure 4.4: *The distribution of results for a fair coin flipped a total of 100 times. We can assess how likely an extreme result is by adding up the probabilities in the tails of the distribution. In the case of a fair coin flipped 100 times, 5% of the experiments should yield* 641 *tails and 5% of the experiments should yield >59 tails. The remaining 90% of the experiments should give between 42 and 58 tails.*

Remember that our null hypothesis (H_0) is that the coin is fair, so at the start of the experiment we are assuming the coin to be fair. If you now did 100 flips of the coin and only got 41 tails or less, you could say that there is only a 5% chance that you would get 41 tails or less in a given experiment, which would make you think that the coin may not be fair. The opposite case of having too many tails also holds. If you got 59 tails or more you could say that there is only a 5% chance that you would get 59 tails or more in a given experiment, which again would make you think that the coin may not be fair. To place this argument into a more robust framework we need to introduce the concept of *significance levels*.

4.1.1 Significance levels

You can think of significance levels like a "trial by jury" court system, where a person is innocent until proven guilty. In our specific case we consider a null hypothesis to be true until we have sufficient evidence to reject it. Returning to our legal analogy, what is the probability of an innocent person being found guilty of a crime they didn't commit (*i.e.*, the available evidence leads to an incorrect guilty verdict)? If 1 person in 100 is wrongly found guilty, the significance level of the system would be 0.01 (in other words a probability of 1%). Alternatively, maybe 1 person in 1000 is wrongly found guilty, the significance level of this system is 0.001 (a probability of 0.1%).

Returning to our coins example, we found that for a experiment consisting of 100 flips of a fair coin we would expect to get between 42 and 58 tails in 90% of the cases. Therefore if we perform an experiment and get either 641 or >59 tails we could reject the null hypothesis (the coin is fair) and accept the alternative hypothesis (the coin is not fair) at a significance level (α) of 0.1. Here the $\alpha = 0.1$ is telling us that there is a 10% chance that given the available data we have incorrectly accepted the alternative hypothesis when the null hypothesis was in fact true.

Maybe we really want to make sure that we don't incorrectly reject the null hypothesis, so we will work with a significance level of $\alpha = 0.01$, which means that our experiment has to fall further into the tails of the binomial distribution before we will reject the null hypothesis (Figure 4.5). For 100 coin flips, if the number of tails fell in the interval [37,62] we would accept the null hypothesis with a significance level of 0.01. If, however, the number of tails was 636 or >63 we would see that the probability of such a result for a fair coin is low (≤1%) and therefore reject the null hypothesis and adopt the alternative hypothesis with a significance level of 0.01.

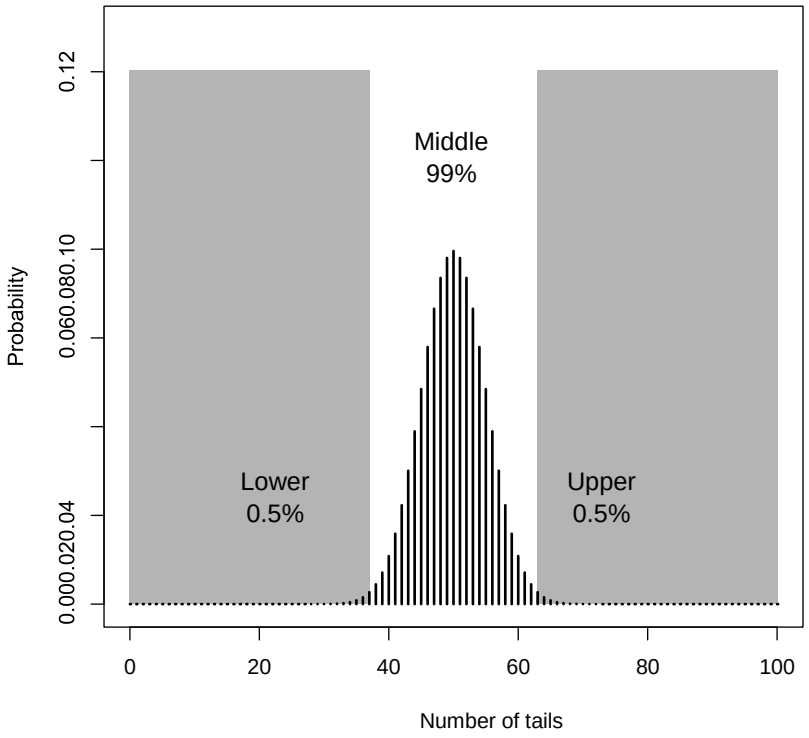


Figure 4.5: *The distribution of results for a fair coin flipped a total of 100 times. We can assess how likely an extreme result is by adding up the probabilities in the extremes of the distribution. In the case of a fair coin flipped 100 times, 0.5% of the experiments should yield* 6*36 tails and 0.5% of the experiments should yield >63 tails. The remaining 99% of the experiments should give between 37 and 62 tails per experiment.*

4.1.1.1 Important points concerning significance levels

You can see through the use of significance levels that we can never be 100% certain of anything in statistics. For example you may flip a fair coin 100 times and get 100 tails, the probability of you doing so is incredibly low, but it is possible. With this in mind we can never say that a coin is truly fair (or unfair) because an extreme result may just be a result of random chance. Therefore our decisions concerning the acceptance/rejection of hypothesis must be made in conjunction with a significance level that says what the probability is that we incorrectly rejected the null hypothesis. With this in mind we need to consider two important points. First the value of the significance level must be selected before the test is performed. This is to avoid abuse of tests, where it could be decided in advance what result is wanted, e.g., the coin is not fair, and then a significance level is chosen to ensure the test gives the desired result. Second, significance levels only tell us about the probability that we have incorrectly rejected the null hypothesis. Significance levels don't give any information about alternative possibilities, for example, incorrectly accepting the null hypothesis.

4.2 An example: Meltwater particles

We have taken two field trips, one to Greenland and one to Antarctica. On each trip we collected 1 litre of water at a number of locations and measured the concentration of microparticles. A total of 16 locations were visited in Antarctica yielding the concentrations (in *ppm*):

3.7,2.0,1.3,3.9,0.2,1.4,4.2,4.9,0.6,1.4,4.4,3.2,1.7,2.1,4.2,3.5

and 18 locations were visited in Greenland yielding the concentrations (again in *ppm*):

3.7,7.8,1.9,2.0,1.1,1.3,1.9,3.7,3.4,1.6,2.4,1.3,2.6,3.7,2.2,1.8,1.2,0.8

To help understand the transport of meltwater particles we want to test if the *variance* in meltwater particle concentrations is the same in Antarctica and Greenland (the variance is just the standard deviation squared). The variance of the population is denoted by; σ^2, however because we are working with a sample we have to make an *estimate of the population variance* by calculating the *sample variance*, s^2:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \qquad (4.2)$$

where *n* is the size of the given data set (i.e., *n*=18 for Greenland and *n*=16 for Antarctica). Using equation 4.2 we find that:

$s^2_{Antarctica} = 2.2263$ $s^2_{Greenland} = 2.6471$

and now we must use this information to draw inferences concerning $\sigma^2_{Antarctica}$ and $\sigma^2_{Greenland}$. To test if the meltwater variances at Antarctica and Greenland are the same, first we must state our hypotheses:

- H_0: The population variances are the same ($\sigma^2_{Antarctica} = \sigma^2_{Greenland}$)
- H_1: The population variances are not the same ($\sigma^2_{Antarctica} \neq \sigma^2_{Greenland}$)

To compare the two variances we will perform a statistical test know as the *F*-test (named in honour of Sir Ronald A. Fisher). The first step of the *F*-test is to compare the variance values by taking their ratio. Of course if $s^2_{Antarctica}$ and $s^2_{Greenland}$ are identical their ratio will give a value of 1. The ratio of the microparticle data is:

$$\frac{s^2_{Greenland}}{s^2_{Antarctica}} = \frac{2.6471}{2.2263} = 1.1890$$

But we still have a problem. Can we consider 1.1890 close enough to our ideal value of 1 that we can accept H_0 and conclude that $\sigma^2_{Antarctica} = \sigma^2_{Greenland}$ or is 1.1890 sufficiently different to our ideal value of 1 that we must reject H_0 and accept H_1, such that $\sigma^2_{Antarctica} \neq \sigma^2_{Greenland}$.

This is the same form of problem as we had in our coin flipping experiment. We therefore need to be able to generate a probability distribution that represents the possible values of variance ratios and we need to select a significance level against which the null hypothesis can be tested.

Earlier we discussed the need to make assumptions in order to draw statistical inference and here we will make the assumption that the data from both Antarctica and Greenland come from normal distributions. Therefore we can take two normal distributions with the same variance and sample 16 random numbers from the first (to represent the Antarctica sampling) and 18 random numbers from

the second (to represent the Greenland sampling), find their respective estimated variances and then take the ratio. Because the distributions have the same variance we know that their their values of σ^2 are the same, however, because we are dealing with samples their values of s^2 will be slightly different each time we draw a set of random numbers and therefore the ratios will form a distribution. The so-called *F*-distribution gives the distribution of ratios for an infinite number of samples. We can control the sample sizes the *F*-distribution represents by adjusting its *degrees of freedom*.

We calculated our ratio with $s^2_{Greenland}$ as the numerator and $s^2_{Antarctica}$ as the denominator. Because the sample sizes for Greenland and Antarctica are 18 and 16, respectively, our *F*-test will employ an *F*-distribution with {18-1,16-1} degrees of freedom. We can then compare the ratio obtained from our Greenland and Antarctica samples to the distribution of ratios expected from two normal distributions with the same variances. We'll perform the *F*-test at the α =0.05 level, which means we need to check if our ratio for the Greenland and Antarctica samples is more extreme than the 5% most extreme values of an *F*-distribution with {18-1,16-1} degrees of freedom. Because our variance ratio could possibly take values less than 1 (if the numerator is less than the denominator) or values greater than 1 (if the numerator is greater than the denominator) our 5% of extreme values must consider the lowest 2.5% and the highest 2.5%, as shown in Figure 4.6.

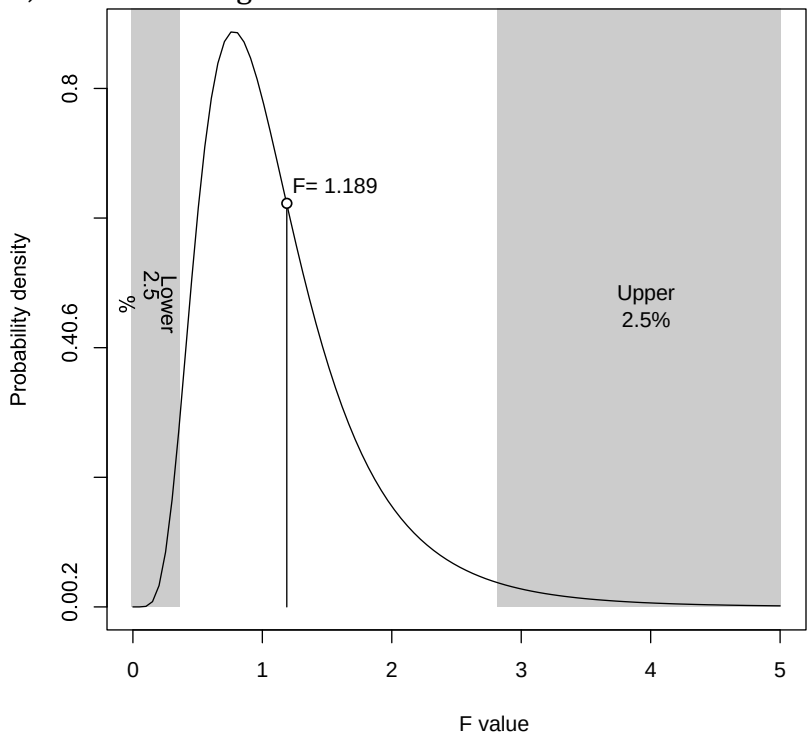


Figure 4.6: *An F-distribution with {18-1,16-1} degrees of freedom. The extreme 5% of the Fvalues are shown by the shaded regions. The F-value of the Greenland to Antarctica variance ratio is shown as an open symbol.*

We can see that our variance ratio for the Greenland and Antarctica samples does not fall into the extremes, so at the α = 0.05 significance level we accept the null hypothesis that $\sigma_{Antarctica}^2 = \sigma_{Greenland2}$.

Voila, we have now performed an *F*-test and shown the population variances of the meltwater particle concentrations at Greenland and Antarctica are the same at the 0.05 significance level. You could now take this information and build it into your understanding of how meltwater particle systems work. This is an important point, the *F*-test has given us some statistical information, but the job of understanding what that information means in a geological sense is your responsibility.

4.2.1 Review of the *F*-test

Let's quickly review how we performed the *F*-test because it forms a good template of how most statistical tests are performed.

- Formulate the null (H_0) and alternative (H_1) hypotheses.
- Choose the significance level (α) at which the test will be performed.
- Calculate the test statistic (ratio of the variances in the case of the *F*-test). • Compare the test statistic to a critical value or values (obtained from the extremes of the *F*-distribution in the case of the *F*-test).
- Accept or reject H_0.

Finally, it is important to consider if we made any assumptions during the statistical test. For the *F*-test we assumed that the Antarctica and Greenland samples both came from a normal distribution. If this is not a valid assumption then the results of the *F*-test will not be valid. Such assumptions make the *F*-test a so-called *parametric* test, which just means that we assume that the data come from a specific form of distribution.

4.3 Now it's your turn: Mesozoic belemnites

Here's your chance to try an *F*-test for yourself. We'll work through all the steps that you need in R and hopefully it will make you more comfortable with how R works. Belemnites have been found in two Mesozoic horizons (**A** and **B**). The lengths of the fossil remains have been measured and recorded (Figure 4.7).

Figure 4.7: *Belemnite fossils (photo courtesy of the Natural History Museum, London).*

We will now perform an *F*-test to determine if the variances of the lengths of the samples from horizons **A** and **B** are the same at the 0.05 significance level. Our first step is to state the null and alternative hypotheses:

H_0: The variances are the same ($\sigma_A^2 = \sigma_B^2$) H_1: The

variances are different (σ_A^2 6= σ_B^2) Then we must

calculate the *F* statistic:

$$F = \frac{s_A^2}{s_B^2}$$

To perform the calculations we must first load the data into R. The data is stored in the file Belemnites.Rdata and it includes two variables A and B.

Example code: 4

```
> rm(list=ls()) # clean out the memory
> load('Belemnites.Rdata') # load the data file
> ls() # show the variables in the memory

[1] 'A' 'B'
```

We now have the two variables in the memory. If we want to the look at the values they contain we just give the variable name and hit the enter key. For example to look at the values in the variable B:

Example code: 5

```
> B
 [1] 5.13 6.59 6.02 3.42 4.92 4.32 3.98 3.77 5.29 4.57 5.06 4.63 4.54 5.37 5.73 [16] 7.11 3.64 3.98 6.04 4.61
```

Note that the values in the square brackets tell you what position in B the beginning of the displayed row corresponds to. For example [16] indicates that the displayed row starts with the 16^{th} value of B, which is 7.11. Using R we can perform a wide variety of mathematical procedures, which makes it very useful when were need to calculate various statistical values.

To calculate the variances of A and B, we will use the function var, give the commands:

Example code: 6

```
> A_var=var(A) # calculate the variance of A and store it in A_var
> B_var=var(B) # calculate the variance of B and store it in B_var
```

We've created two new variables A _var and B var, which contain the variances of A and B, respectively. In turn we can use these values to calculate the *F*-statistic.

Example code: 7

```
> F=A_var/B_var
> F
[1] 0.3375292
```

So the ratio of the sample variances is 0.338 and now we need to compare this value to a *F*distribution to see if it lies in the most extreme 5%. This is what we call a *two-sided* test, so as before we need to consider the lowest 2.5% and highest 2.5% of the *F*-distribution. Therefore we're accounting for the possibility that the ratio is significantly less than 1 or significantly greater than 1 (just like in our example with the microparticle concentrations). To find the values of *F* for the extremes we first need to find the number of degrees of freedom for the distribution. We can do this using the length function, which tells us how many entries there are in a given variable. Once we know the degrees of freedom we can find the value at which the *F*-distribution reaches a given probability using the function qf. The qf function has 3 inputs; the probability we are interested in, the number of degrees of freedom of the numerator and the number of degrees of freedom of the denominator.

Example code: 8

```
> nA=length(A) # number of entries in A
> nB=length(B) # number of entries in B
> F_crit1=qf(0.025,nA-1,nB-1) # find the lower critical value
> F_crit2=qf(1-0.025,nA-1,nB-1) # find the upper critical value
```

We've now calculated all the values that we need and we can display them together to finish off the test.

Example code: 9

```
> F #value of the test statistic
[1] 0.3375292
> F_crit1 #lower critical value of the F-distribution
[1] 0.3797761
> F_crit2 #upper critical value of the F-distribution [1] 2.566993
```

We can see that our *F*-value from the belemnite data is less than the lower critical value of the *F*-distribution (Figure 4.8). This means it is an extreme result and at the $\alpha = 0.05$ significance level we must reject the null hypothesis and accept the alternative hypothesis that the population variances of the lengths of A and B are different.

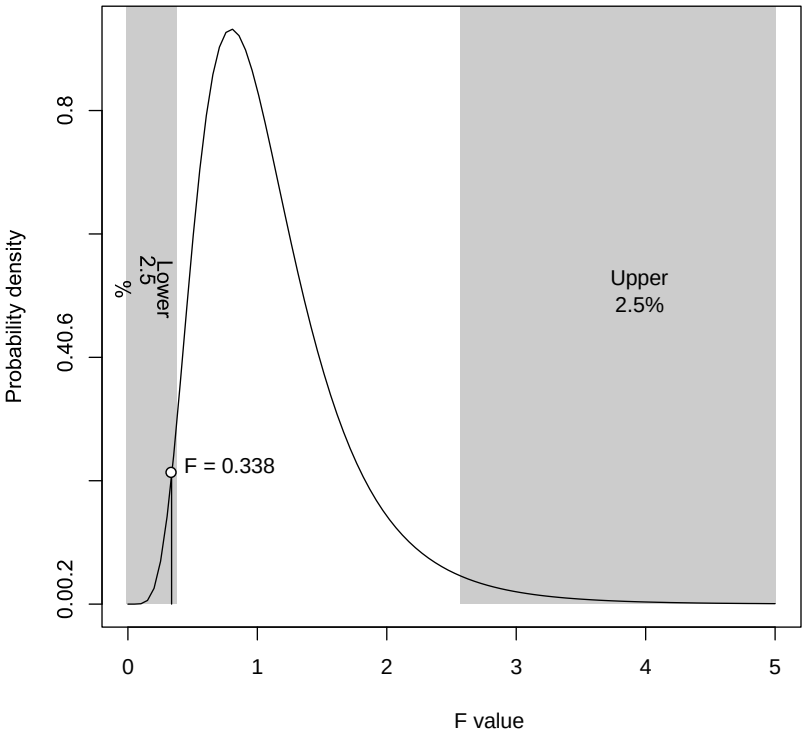


Figure 4.8: *An F-distribution with {17,19} degrees of freedom. The extreme 5% of the F-values are shown by the shaded regions. The F-value of the belemnite variance ratio is shown as an open symbol.*

4.4 Other hypothesis tests

There is a vast number of different hypothesis tests with which to investigate many aspects of different data sets. Nearly all hypothesis tests work in the same way, following the sequence:

- Formulate the null (H_0) and alternative (H_1) hypotheses.
- Choose the significance level (α) at which the test will be performed.
- Calculate the test statistic(s).
- Compare the test statistic to a critical value or values (obtained from a suitable distribution).
- Accept or reject H_0.

So if there is some property you need to test in order to make inferences about the geological system you are studying you can usually find a test that will do it for you. We'll look at one final example to again demonstrate the general nature of hypothesis testing.

4.4.1 Meltwater particles revisited

We're going to use the same meltwater particle data set as we analyzed earlier, but this time rather than using an *F*-test to find if the variances of the Antarctica and Greenland samples are the same we'll study the population means instead. Specifically, we'll employ Student's *t*-test to determine if the means of the

Antarctica and Greenland samples are the same or not. The *t*-test does have assumptions and it's important that we pay attention to them:

- The populations have the same variance.
- The samples come from normal distributions.

You can see that the assumption of normal distributions is the same as for the *F*-test, but now we have the added assumption that the Antarctica and Greenland samples have the same variance. Fortunately, in our earlier analysis we established using an *F*-test that the Antarctica and Greenland samples have the same variance so we know that our data meet this assumption. If we want to test if the two means are the same, first we must state our hypotheses:

H_0: The means are the same ($\mu_{Antarctica} = \mu_{Greenland}$) H_1: The means are different ($\mu_{Antarctica} \neq \mu_{Greenland}$)

and we'll work with a significance level of $\alpha = 0.05$. The *t*-statistic is a little more complicated to calculate than the *F*-statistic, but it is still just an equation that we plug known values into.
Specifically:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \qquad S = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \tag{4.3}$$

where n_1,n_2 are the number of values from Antarctica (16) and Greenland (18), $\bar{X}_1$,$\bar{X}_2$, are the mean sample values from Antarctica and Greenland and s_1^2, s_2^2 are the sample variances from Antarctica and Greenland. We can calculate the test-statistic in R, using the variables A (Antarctica) and G (Greenland) stored in the file microparticles.Rdata. As you type in commands into R, check that you can see how they marry up with the terms in equation 4.3. Also notice that we can reuse variables names such as top and bottom once we are finished with them (R simply overwrites the existing values).

Example code: 10

```
> rm(list=ls()) #clear the memory
> load('microparticles.Rdata') #load the data file containing A and G
> n1=length(A) #define the number of values in A
> n2=length(G) #define the number of values in G

> top=(n1-1)*var(A)+(n2-1)*var(G) # calculate the numerator of S
> bottom=n1+n2-2 # calculate the denominator of S
> S=sqrt(top/bottom) # find the pooled sample variance
> top=mean(A)-mean(G) #calculate the numerator of the test statistic > bottom=S*sqrt(1/n1+1/n2)
#calculate the denominator of the test statistic
> t=top/bottom #calculate the test statistic
```

```
> t #display the final t value
[1] 0.375768
```

We'll compare our test statistic to critical values drawn from a *t*-distribution. The formulation of the *t*-distribution follows a similar concept to that of the *F*-distribution. We take two normal distributions with the same variances and means, draw a random sample from each of them and then calculate the corresponding value of *t*. If we did this infinitely many times we would produce a *t*-distribution. To find our critical values we need to work with a *t*-distribution that has $n_1 - 1 + n_2 - 1$ degrees of freedom and because this is a two-sided test (i.e., the means can be different in two ways, either $\mu_{Antarctica} > \mu_{Greenland}$ or $\mu_{Antarctica} < \mu_{Greenland}$) we need to look at the lower and upper 2.5% extremes. Once we know the degrees of freedom we can find the value at which the *t*-distribution reaches a given probability using the function qt, which has two inputs; the probability we are interested in and the total number of degrees of freedom.

Example code: 11

```
> alpha=0.05 # set the significance level
> t_crit1=qt(1-alpha/2,n1-1+n2-1) # calculate the upper critical value
> t_crit2=qt(alpha/2,n1-1+n2-1) # calculate the lower critical value
```

We can now look at all the values together:

Example code: 12

```
> t #display the final t value [1] 0.375768

> t_crit1 #display the upper critical t value [1] 2.036933

> t_crit2 #display the lower critical t value
[1] -2.036933
```

The first thing to notice is that the *t*-distribution is symmetrical about 0 (Figure 4.9) therefore the upper and lower critical values have the same magnitude but different signs. We can see that the *t*-value for the microparticle data lies between the critical values rather than in the extremes of the distribution. Therefore we can accept the null hypothesis and state that $\mu_{Antarctica} = \mu_{Greenland}$ at the 0.05 significance level.

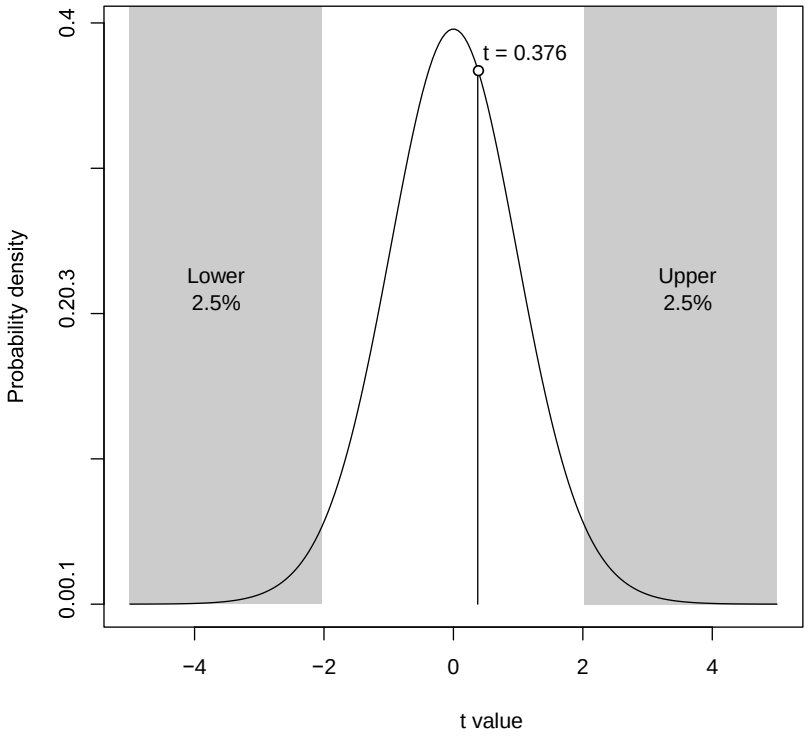


Figure 4.9: *A t-distribution with 32 degrees of freedom. The extreme 5% of the t-values are shown by the shaded regions. The t-value of the microparticle data is shown as an open symbol.*

The invalid assumption that correlation implies cause is probably among the two or three most serious and common errors of human reasoning.

Stephen Jay Gould

5 Correlation and regression

The focus of this chapter is correlation and regression, which are two closely related techniques. You may find that the terms are used somewhat interchangeably in the literature, so a good stating place is to provide clear definitions of what the two techniques actually are.

Linear correlation: to what *degree* are variables related linearly. **Linear regression**: *how* are variables related linearly.

I'm sure that you are familiar with correlation and regression from packages like EXCEL that include them as part of their *trendlines* options. We're going to look at both correlation and regression in more detail and examine what important information they can and cannot provide you with.

5.1 Correlation

As mentioned above, correlation tells us the *degree* to which variables (2 or more) are related linearly. Correlation usually expresses the degree of the relationship with a single value. By far the most common value used in correlation analysis is the Pearson product-moment correlation coefficient (PPMCC) and it will be our focus in this section. You've all probably used a variant of the PPMCC before in the "R-squared" value EXCEL allows you to add to trendlines in scatter plots. In which case, you should be familiar with at least a qualitative interpretation of the PPMCC which is simply the value "R" that is used to calculate "R-squared". We will use a lowercase *r* rather than *R* so that it won't cause any confusion with the language R that we are using for the examples. In EXCEL, *r* is only a few mouse clicks away, but to understand what information it can give us we need to look at how it is calculated (sorry, but it's unavoidable) and how we can draw inferences from it.

5.1.1 Sample correlation

If we have a data set consisting of two variables, let's say; X and Y , first we look at how far each value is away from its corresponding mean. We'll call these differences *deviations* and represent them with lowercase letters:

$$x = X - \bar{X}, \tag{5.1}$$

$$y = Y - \bar{Y}. \tag{5.2}$$

Once we've found the deviations, r is given by the formula:

$$r = \frac{\sum xy}{\sqrt{\sum x^2}\sqrt{\sum y^2}}. \tag{5.3}$$

Remember that the $\sum$ simply means sum all the values together. We'll apply the correlation analysis to some real data to gain an appreciation of how equation 5.3 works and what it is telling us.

The data set stored in gravels.Rdata contains two variables from a study of how the size of gravel on a river bed varies as we move downstream. The variable size contains the size of the gravels in φ units, where larger values indicate small sizes (for example a piece of gravel with φ = -6 is larger than a piece of gravel with φ = -5). Don't worry if you're not familiar with the φ system, it's not so important, but we can justify its use straight way. In natural systems we tend to find that grain sizes are log-normally distributed (we discussed this in Section 3.2.0.1), but one of the assumptions of the correlation analysis is that the data are (at least approximately) normally distributed. The φ grain size scale is defined as:

$$\varphi = -\log_2 D/D_0, \tag{5.4}$$

where D is the diameter of the particle and D_0 is a reference diameter equal to 1 mm. So if the data are log-normally distributed, the φ scale transforms them into a normal distribution and the assumptions of the correlation analysis are met. The second variable in the data set is the downstream distance, dist,

which has units of kilometers. In this case a value of, for example, *dist* = 10 corresponds to a piece of gravel being collected 10 km downstream with respect to the starting location of the experiment.

The first thing we'll do is plot the data to take a look at it. We can do this in R using the commands.

Example code: 13

```
> rm(list=ls()) #clear the R memory
> load('gravels.Rdata') #load the data file

> #plot the data points with black symbols and label the axes
> plot(dist,size,col='black',xlab='Distance [km]',ylab='Gravel size [phi]')
```

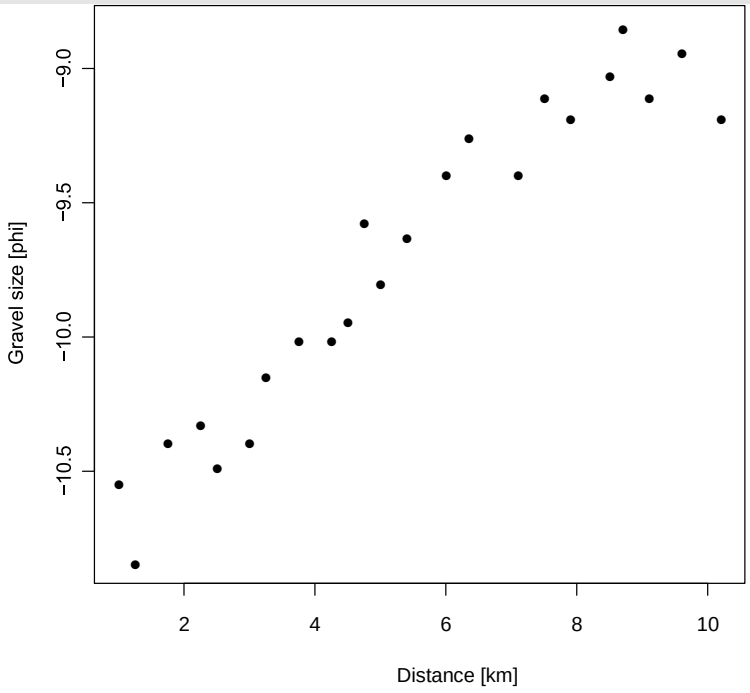


Figure 5.1: *Plot of how gravel size varies downstream in a river.*

Just by looking at the plot we can see that there may be a linear relationship between gravel size and distance, but we'll need to calculate the PPMCC to quantify the degree of the relationship. As we saw in equations 5.1 and 5.2, the first step to calculating *r* is to find the deviations of the variables. The structure of the deviations is the key to understanding equation 5.3, so we'll now plot them in a new figure and consider them in a general way in the next section.

Example code: 14

```
> x=dist-mean(dist) #calculate the deviations in the distance
> y=size-mean(size) #calculate the deviations in the gravel size
> plot(x,y,col='black') #plot the deviations
```

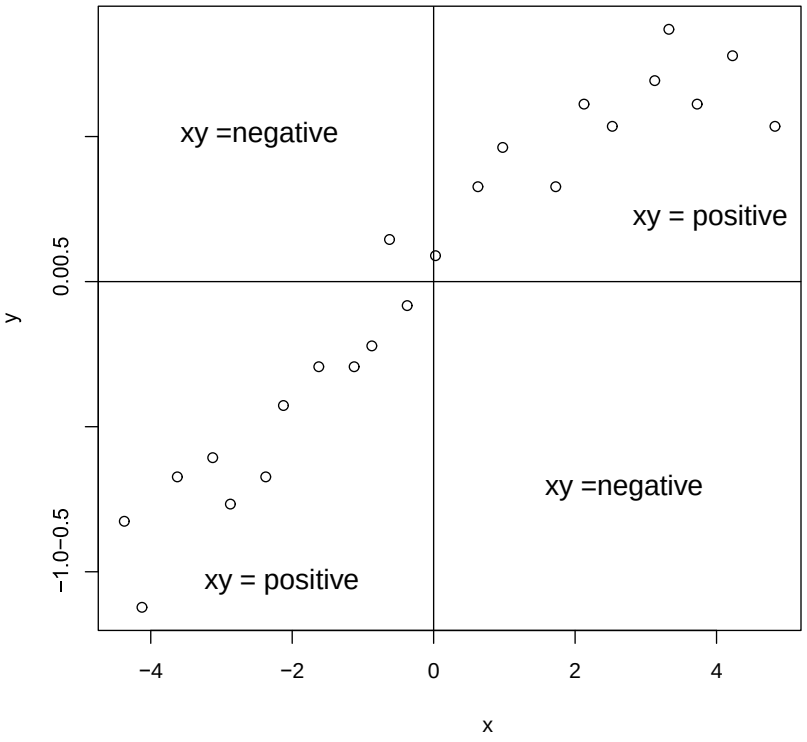


Figure 5.2: *Deviations of the gravel data set. Notice how the sign of xy will depend on the quadrant of the plot where the deviations lie.*

Looking back at equation 5.3 we can see that the top of the equation involves multiplying the two sets of deviations together. This multiplication gives us a measure of how well the two variables are moving together. If the deviations of a given observation have the same sign (i.e., the point representing the deviations lies in the first or third quadrant) the product *xy* will be positive. Alternatively, if the deviations of a given observation have different signs (i.e., the point representing the deviations lies in the second or fourth quadrant) the product *xy* will be negative. If values in *X* and *Y* exhibit a positive relationship, i.e., as *X* increases *Y* also increases, then most of our deviations will define points that lie in the first or third quadrants and when we form the sum $\sum xy$, we'll get a positive number. In the opposite case of a negative relationship, i.e., as one variable increases the other decreases, most of the deviations will define points in the second or fourth quadrant. In such cases $\sum xy$ will be negative. The last case to consider is when no relationship exists between *X* and *Y* . In this situation the points defined by *x* and *y* will be spread amongst all four quadrants and the signs of *xy* will cancel to give a value of $\sum xy$ close to zero.

We can see that the sign of $\sum xy$ tells us about the sense of the correlation, i.e., is it positive or negative, but there are problems with the magnitude of the value. Clearly, each value of *xy* will depend on the units of the data. In the example of our river gravels, if the down stream distance was in millimeters rather than kilometers each value of *xy* would be 10^6 times larger because $10^6\ mm = 1\ km$. To compensate for the problem we take the magnitudes of the *x* and *y* values into account in the denominator of equation 5.3 and this makes the PPMCC "scale invariant". We've calculated and plotted the deviations of the gravel data already and now it is a simple task to calculate the PPMCC using R.

Example code: 15

```
> top=sum(x*y) #calculate top of the equation
> bottom=sqrt(sum(x^2))*sqrt(sum(y^2)) #calculate bottom of the equation
> r=top/bottom #calculate the PPMCC
> r #print the value to the screen
```

Of course, calculation of the PPMCC is a common task in statistics and R can do the whole process with a call to the function cor. Calculate the PPMCC for the gravel data again using cor:

Example code: 16

```
> r=cor(dist,size) #use the built-in function to find PPMCC
```

You should obtain a PPMCC value of ~0.96 for the gravel data.

The value of r can range between -1 (a perfect negative correlation) and 1 (a perfect positive correlation). It's important to now repeat our definition from above that correlation measures the *degree* to which variables are related **linearly**. Therefore a value of r close to zero does not mean that there is no relationship between x and y, but that there is no linear relationship between them. This is demonstrated in Figure 5.3, which shows a variety of different relationships and their corresponding PPMCC. Some of the cases show a clear relationship between x and y, but because it is not a straight-line relationship the corresponding value of r is close to 0.

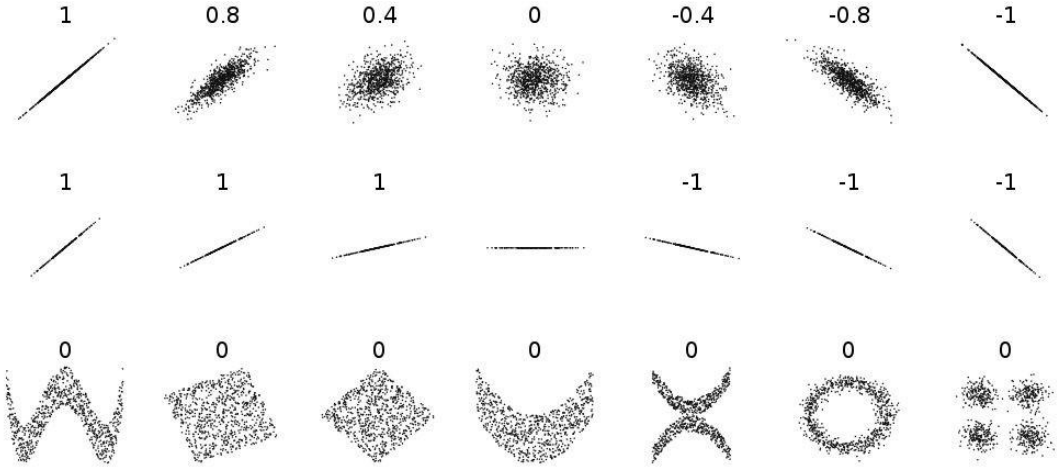


Figure 5.3: *Examples of the PPMCC value for different data distributions. Notice that some cases yield r = 0 even though there is a clear relationship between x and y. This is because the PPMCC only measures the extent of the linear relationship.*

5.1.2 The coefficient of determination, r^2

The coefficient of determination, r^2, is simply the square of the value given by equation 5.3 (yes it is that easy). We're not going to dwell on r^2 except to look what advantages and disadvantages it has over using r. First the slight disadvantage, because r^2 must always be a positive number, we lose information about

the sign of the relationship (is it positive or negative). The advantage of r^2 is that it tells us what proportion of the total variation in Y is accounted for in the linear relationship with X. Taking our gravel data as an example, we found $r = 0.96$ and thus $r^2 = 0.92$. Therefore 92% of the variation in the gravel sizes is accounted for by their position downstream, whilst $100(1 - r^2)\%$ of their variation is not accounted for by their position downstream.

5.1.3 Population correlation

In the previous section we calculated the PPMCC for a sample of gravels from a river. This tells us about the linear correlation in our sample and if we returned to the river and collected a new set of gravels we would expect to get a slightly different value of r to the one we got above. Clearly the aim of such an analysis is to make inferences about the population (all the gravels in the river) on the basis of the sample (the gravels we collected). Here the maths and statistical reasoning becomes a little involved so we're not going to worry about why the method works (look at the recommended reading list if you want more details), but instead focus on the application of the method.

To begin, the value of r we calculated for the sample provides an estimate of the population correlation coefficient, ρ. However, because this is an estimate we need to consider what extra information we can obtain for ρ. First we'll perform a hypothesis test to test the significance of ρ and then we'll calculate the confidence interval on ρ.

5.1.4 Test for significance of a correlation coefficient

The aim of this test is to find if ρ is significantly different from 0. This is particularly useful because if ρ is significantly different from 0 it implies there is a significant correlation between X and Y . The hypotheses for the test are:

- Null hypothesis (H_0): $\rho = 0$
- Alternative hypothesis (H_1): $\rho \neq 0$

To calculate the test statistic we use the value of r calculated above and n, the number of data points in the sample. The test statistic, t, is given by:

$$t = r\sqrt{\frac{n-2}{1-r^2}} \tag{5.5}$$

The test statistic can then be compared to a critical value drawn from a Student's t-distribution with $n-2$ degrees of freedom. If the calculated value of t exceeds the critical t then we can reject the null hypothesis and conclude that a significant correlation exists between X and Y at the chosen significance level, α. Lets try out this test on our river gravels using R.

Example code: 17

```
> n=length(dist) #number of observations in the analysis
> t=r*sqrt((n-2)/(1-r^2)) #test statistic t value
> alpha=0.05 #set the significance level for the test
```

```
> tcrit=qt(1-alpha/2,n-2) #find the 2-sided critical value
```

We find that the test statistic (15.14) is indeed larger than the critical value (2.08) so we reject the null hypothesis and accept the alternative hypothesis. Therefore a significant correlation exists between gravel size and downstream distance at the α = 0.05 significance level.

5.1.5 Confidence interval for the population correlation

If we can show that ρ is significantly different from 0 at some desired significance level then we can also calculate the confidence interval on ρ. The confidence interval will tell us the range in which the true value of ρ should lie with a given probability. The first step is to take our calculated value of r and apply Fisher's z transform:

$$z_r = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right), \tag{5.6}$$

The standard error of z_r is approximately:

$$SE = \frac{1}{\sqrt{n-3}}, \tag{5.7}$$

where n is the sample size. Because z_r is normally distributed we can find confidence intervals for z_r from a normal distribution just like in Section 3.2.1. For example, the 95% confidence interval on z_r would be $[z_r - (1.96 * SE), z_r + (1.96 * SE)]$. We now have the confidence interval for z_r and to find what this corresponds to in terms of correlation coefficients we need to apply the inverse z transform (i.e., convert from z_r values back to r values):

$$r = \frac{e^{2z}-1}{e^{2z}+1} \tag{5.8}$$

We'll now calculate the confidence interval on ρ for our river gravels using R the step-by-step way. See below for a shortcut.

Example code: 18

```
> zr=0.5*log((1+r)/(1-r)) #perform the Fisher z transform
> SE=1/sqrt(n-3) #estimate the standard error
> C=0.95 #set the confidence level as 95%
> nC=-qnorm((1-C)/2) # value from a normal distribution
> z_low=zr-(nC*SE) #lower value of the zr confidence interval
> z_up=zr+(nC*SE) #upper value of the zr confidence interval
> r_low=(exp(2*z_low)-1)/(exp(2*z_low)+1) # inverse z transform
> r_up=(exp(2*z_up)-1)/(exp(2*z_up)+1) # inverse z transform
> r_low #show the lower limit of the interval on screen
> r_up #show the upper limit of the interval on screen
```

Given the 95% confidence interval of ρ is [0.90,0.98] you can see that it is not particularly meaningful when people quote *r* values to a large number of decimal places. As with most of these common statistical tasks, R can perform the calculation automatically using the function cor.test as shown below.

Example code: 19

```
> cor.test(dist,size)
```

5.1.6 The influence of outliers on correlation

An outlier is a data point that deviates markedly from other members of the sample in which it occurs. An outlier could be from an extreme end of a distribution, for example Marilyn vos Savant's IQ of 228, or simply the product of a large measurement error. The important point is that outliers can have a dramatic effect on the calculated PPMCC because they allow a large amount of the data variability to be described by a single point (because the product of their deviations is very large in relative terms). Shown in Figure 5.4 is a small data set with a correlation that is effectively zero (r^2 = 0.03). When an outlier is included in the same data set the correlation becomes highly significant (r^2 = 0.81) because most of the variability in the data is controlled by the single outlier.

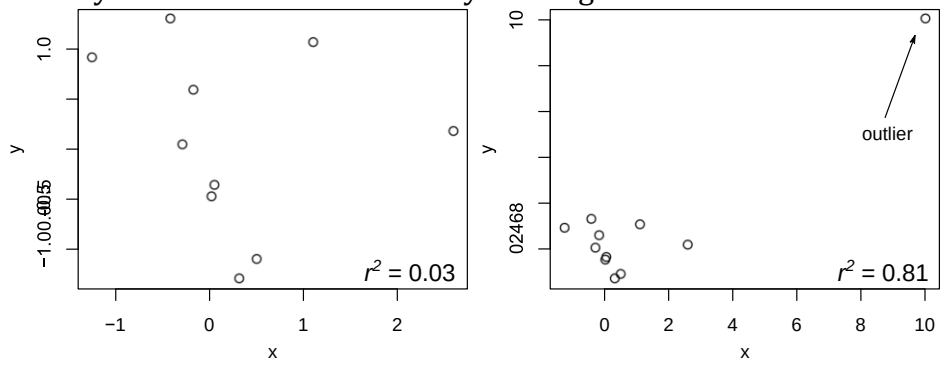


Figure 5.4: *The correlation in a data set can be strongly influenced by the presence of outliers. The panel on the left shows a data set which exhibits no correlation. However, when an outlier is added to the data the correlation increases dramatically.*

It is therefore important that you check your data in advance for outliers and consider removing them from the analysis.

5.1.7 Spurious correlations

It's important to consider what a significant correlation between two variables really tells us. For example, observations through the 19th Century show that a strong correlation existed in the USA between the number of priests and the number of alcoholics. How should we interpret such a relationship? Does it tell us that most priests are alcoholics, or possibly attending church drives people to alcoholism? We can think of a number of possible scenarios that link priests and alcoholics, but the truth is much simpler. As the population of the USA increased in size through the 19th Century, so of course the number of both priests and alcoholics also increased. In this way population size is a confounding variable that renders any kind of cause and effect relationship between priests and

alcoholics spurious. Therefore, it is accurate to say that the correlation between the number of priests and alcoholics is statistically significant, but any naive inference concerning cause and effect is clearly meaningless.

5.2 Regression

In the previous section we assessed the *degree* to which two variables are related linearly using correlation. The aim of regression is to quantify *how* the variables are related linearly. To put this in simpler terms, how do we find the straight-line that will give the best description of the sample data. Again this is a task that is performed commonly in EXCEL, but we need to look at the process in more detail. Firstly, you should all be familiar with the equation for a straight-line:

$$Y = a + bX, \tag{5.9}$$

where a is the intercept (the value of Y when X = 0) and b is the gradient. Let's again consider our river gravels. With distance as the X value and grain size as the Y value we fit a straight line to the data to obtain a and b for our studied sample. Of course what really interests us is making inferences from the sample to estimate the relationship for the population. Thus whilst we can calculate a and b, what we really want is confidence intervals for α and β. It's also important to note that there are assumptions associated with linear regression, we'll state them now and then look at them in more detail later:

- Requires an independent (X) and dependent (Y) variable.
- The errors associated to X should be orders of magnitude less than those on Y.
- Both X and Y should be normally distributed.

5.2.1 Calculating a and b for a sample

Equation 5.9 describes a straight-line, but it is very unlikely that our data will fall perfectly onto a straight-line (from our plots above we know the river gravel data certainly don't). Therefore if we are going to properly relate X and Y we need to include an collection of errors, E, that tell us what the difference is between the data and the line:

$$Y = a + bX + E, \tag{5.10}$$

This idea is shown graphically in Figure 5.5.

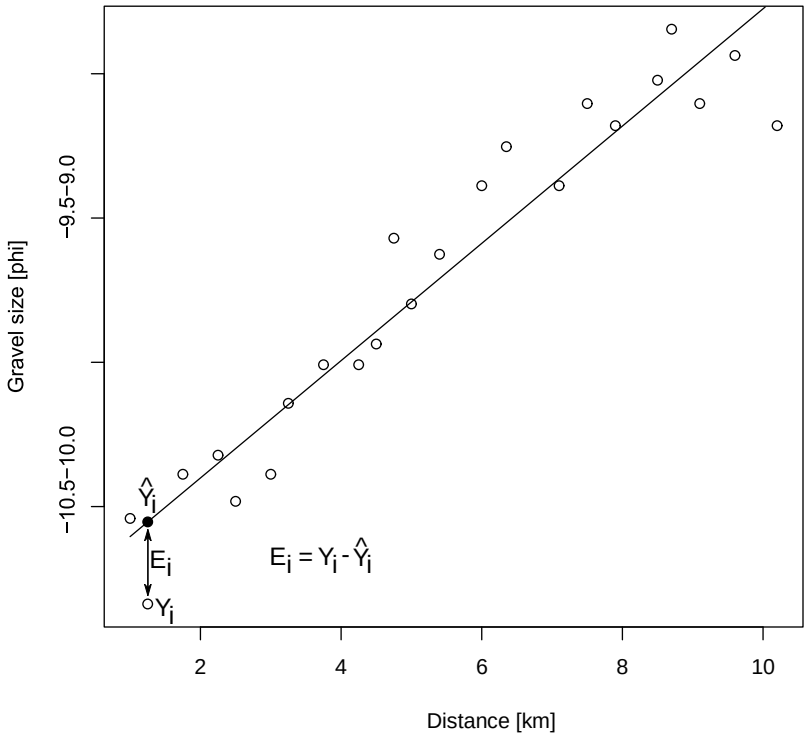


Figure 5.5: *The error E_i for point i based on the difference between the data point Y_i and its corresponding point on the line $\hat{Y}_i$ (more on this later).*

A line that fits the data closely will produce small errors, whilst a poorly fitting line will produce large errors. Therefore if we can get the collection of errors in *E* as small as possible we will know that we've found the best fitting line. We can do this by finding the line that produces the smallest possible value of $\sum E^2$. Another property of *E* that we can use to help us is that for the best fitting line $\sum E = 0$. It makes intuitive sense that for the best fitting line all the errors will cancel each other out. To make the errors in *E* as small as possible (i.e., minimize $\sum E^2$) and ensure that $\sum E = 0$ we use the approach of least-squares (which is the technique Gauss developed when he was 18 years old and subsequently used to make his predictions about the orbit of Ceres). To calculate *b*, we'll again use deviations in *X* and *Y* (equations 5.1 and 5.2):

$$b = \frac{\sum xy}{\sum x^2}. \tag{5.11}$$

We also know that the best fit line must pass through the point ($\bar{X}, \bar{Y}$), i.e., the mean of the data, so once *b* is found we can calculate *a*:

$$a = \bar{Y} - b\bar{X}. \tag{5.12}$$

We now know the equation for the best-fit line, which provides us with a linear model relating *X* and *Y*. We can therefore use the line to make predictions about *Y* given a value or values of *X*. If we were interested in the value of *Y* at a value of *X* denoted by X_0, the prediction of *Y* is given by:

$$\hat{Y}_0 = a + bX_0, \tag{5.13}$$

note the $\hat{Y}$ notation that denotes we are making a prediction of Y. Let's try this out in R, again using the downstream gravel data and the deviations we calculated earlier. We'll make predictions for the original X values in order to draw the regression line in the plot (Figure 5.6).

Example code: 20

```
> b = sum(x*y)/sum(x^2) #use the deviations to find the slope
> a = mean(size)-b*mean(dist) #use the original data to find the intercept
> Yhat = a+b*dist #predicted gravel size for the distance values
> plot(dist,size,col='black',xlab='Distance [km]',ylab='Gravel size [phi]')
> lines(dist,Yhat,col='black') #plot the regression line
```

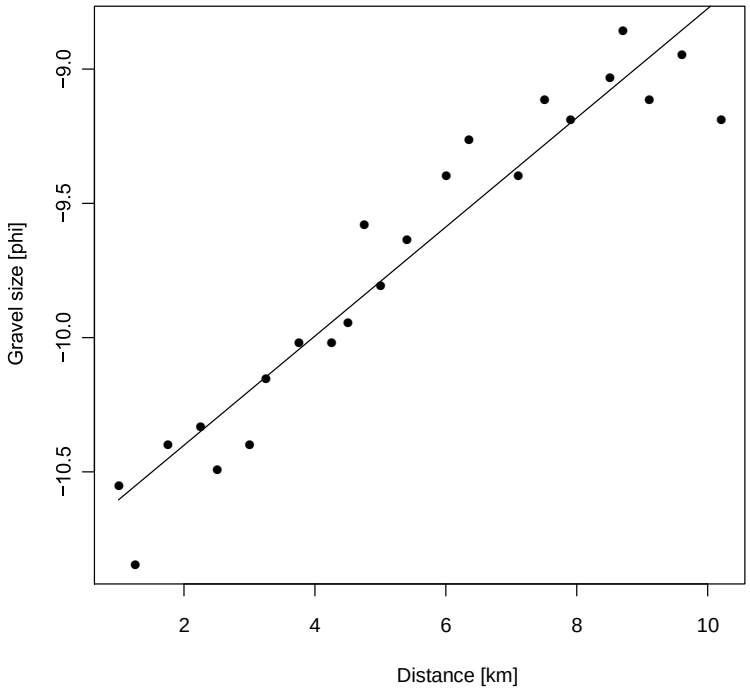


Figure 5.6: *Regression line showing the linear relationship relating gravel size to distance downstream.*

5.2.2 The influence of outliers on regression

As we saw earlier, outliers can have a dramatic effect on the correlation of two variables and the same problem exists in regression. An outlier can be so far away from the real data trend that it pulls the regression line towards it, yielding a spurious relationship (Figure 5.7).

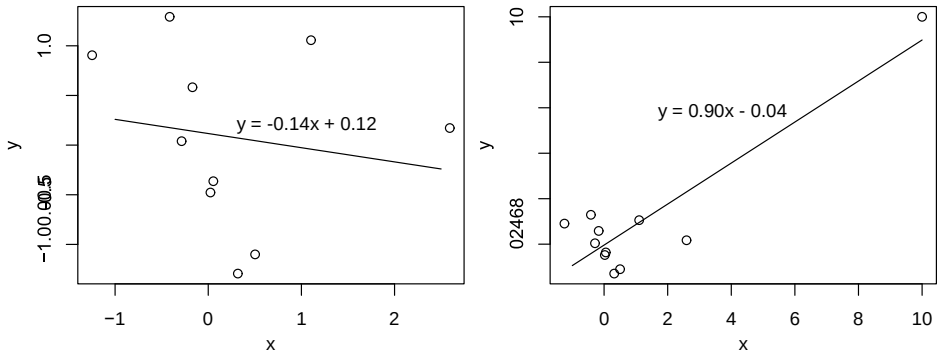


Figure 5.7: *Regression can be strongly influenced by the presence of outliers. The panel on the left shows the regression for a data set with no outliers. However, when an outlier is added to the data (right) the regression line is pulled towards it, changing the regression dramatically.*

Therefore, as with correlations, it is essential that you check your data set in advance for outliers and consider removing them from the analysis.

5.2.3 Confidence interval for the slope

Now that we've found b for our sample it's important to draw inferences concerning the population, specifically the slope β. The first step is to examine the *residuals*, which are simply the difference between the values of Y measured for the sample and the corresponding predicted values $\hat{Y}$ obtained from the regression equation. What we need to know first is the estimated variance of the residuals, s^2, given by:

$$s^2 = \frac{1}{n-2}\sum(Y - \hat{Y})^2 \tag{5.14}$$

From the estimated variance it is then simple to calculate a confidence interval for the slope β again using a Student's t-distribution with $n - 2$ degrees of freedom. For example, if we wanted to find the 95% confidence interval, we would use

$$\beta = b \pm t_{0.025}\frac{s}{\sqrt{\sum x^2}} \tag{5.15}$$

The 0.025 value is used because the confidence interval is a two-sided statistic and (100−95)/100/2 = 0.025. As with the other confidence intervals we studied earlier, the interval here has a 95% chance of containing the true value of β. Let's carry on with our gravels example and calculate the 95% confidence interval for the slope in R.

Example code: 21

```
> Yhat = a+b*dist #predicted gravel size for the distance values
> s2 = 1/(n-2)*sum((size-Yhat)^2) #estimated variance of residuals > s = sqrt(s2) #estimated standard deviation of the residuals
> t=-qt(0.025,n-2) #obtain the t distribution value with dof = n-2
> beta_low=b-t*s/sqrt(sum(x^2)) #lower value of 95% CI
> beta_up=b+t*s/sqrt(sum(x^2)) #upper value of 95% CI
```

```
> b #show the value of the slope on screen
> beta_low #show the lower value of the slope confidence interval
> beta_up #show the upper value of the slope confidence interval
```

So we can say that on the basis of our sample gradient, b = 0.203 $\varphi\ km^{-1}$, there is a 95% probability that the true population gradient lies in the interval [0.175, 0.231] $\varphi\ km^{-1}$.

5.2.4 Confidence interval for the intercept

Not surprisingly, the calculation of a confidence interval for the population intercept, α, is similar to the approach taken for the slope in the previous section. Therefore we won't dwell on it too much, the key pieces of information we need are the intercept estimated from the sample (a obtained from equation 5.12), the estimated variance of the residuals (s^2 obtained by equation 5.14), the mean of X ($\bar{X}$), the deviations of X (x obtained from equation 5.1) and the number of observations included in the sample (n). The 95% confidence interval for α is then given by:

$$\alpha = a \pm t_{0.025} s \sqrt{\frac{1}{n} + \frac{\bar{X}^2}{\sum x^2}} \tag{5.16}$$

Let's go back to our gravels example and see how we would calculate the confidence interval for α in R.

Example code: 22

```
> Yhat = a+b*dist #predicted gravel size for the distance values
> s2 = 1/(n-2)*sum((size-Yhat)^2) #estimated variance of residuals
> s = sqrt(s2) #estimated standard deviation of the residuals
> t=-qt(0.025,n-2) #obtain the t distribution value with dof=n-2
> alpha_low=a-t*s*sqrt(1/n+mean(dist^2)/sum(x^2)) #lower value of 95% CI
> alpha_up=a+t*s*sqrt(1/n+mean(dist^2)/sum(x^2)) #upper value of 95% CI
> a #show the value of the slope on screen
> alpha_low #show the lower value of the intercept confidence interval
> alpha_up #show the upper value of the intercept confidence interval
```

So we can say that on the basis of our sample intercept, a = -10.81 φ, there is a 95% probability that the true population intercept lies in the interval [-10.99, -10.62] φ.

5.2.5 Making predictions from a regression model

With equation 5.13 we saw how easy it is make predictions of Y given some value of X once a and b have been found. You won't be surprised to hear that things are not quite that simple. We have to think about what we are predicting and of course include some form of confidence interval in our predications given that our regression is based on a sample that is scattered around the regression line. Here our gravels data set provides an illustrative example of the different predications we can make.

5.2.5.1 Predicting a mean

In the sections above we found that a = -10.807 and b = 0.203, so for example, at distance 5 km downstream we can make a predication of the the gravel size:

$$a + bX_0 = \hat{Y}_0,$$
$$-10.807 + 0.203 * 5 = -9.792$$

Okay, so we've predicted a value of -9.8 φ but what does this value correspond to? It is a prediction of the **mean** of Y at X_0. So in the case of our example data set the estimated **mean** gravel size at a downstream distance of 5 km is -9.8 φ. The focus of the previous sections was the calculation of confidence intervals on the slope and intercept and in a similar manner we need to include a confidence interval on the estimated mean because of the uncertainty associated with working with a sample rather than the whole population and the misfit between the data and the regression model. Again we'll use the estimated variance of the residuals, s^2 (equation 5.14). The 95% confidence interval for the mean of Y_0 at the position X_0 is given by:

$$\mu_0 = (a + bX_0) \pm t_{0.025} s \sqrt{\frac{1}{n} + \frac{(X_0 - \bar{X})^2}{\sum x^2}} \tag{5.17}$$

where t again represents Student's t-distribution with n–2 degrees of freedom. Let's perform this calculation in R for the gravel data at a position X_0 = 5 km (Figure 5.8).

Example code: 23

```
> Yhat = a+b*dist #predicted gravel size for the distance values
> s2 = 1/(n-2)*sum((size-Yhat)^2) #estimated variance of residuals
> s = sqrt(s2) #estimated standard deviation of the residuals
> t=-qt(0.025,n-2) #obtain the t distribution value with dof=2
> X0=5 # make predictions for a distance of 5 km downstream
> C=t*s*sqrt(1/n+(X0-mean(dist))^2/sum(x^2)) #half-width of the CI
> mu0=a+b*X0 #prediction of mean at X0
> mu0_low=mu0-C #lower value of the 95% CI
> mu0_up=mu0+C #upper value of the 95% CI
> mu0 #show the value of the mean at XO
> mu0_low #show the lower value of the mean confidence interval
> mu0_up #show the upper value of the mean confidence interval
> plot(dist,size,col='black',xlab='Distance [km]',ylab='Gravel size [phi]')
> lines(dist,Yhat,col='black') #plot the regression line
> lines(c(X0,X0),c(mu0_low,mu0_up)) #plot the CI for the predicted mean
```

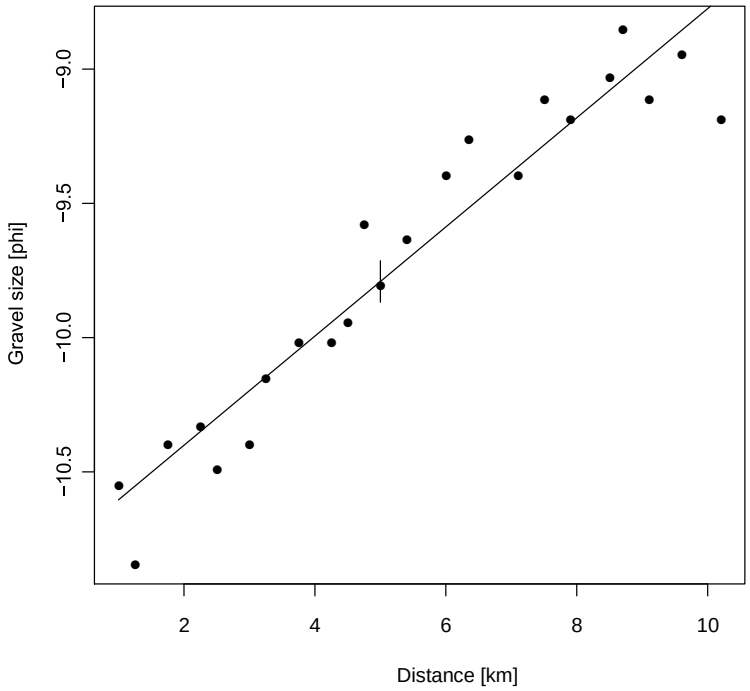


Figure 5.8: *Regression line showing the linear relationship relating gravel size to distance downstream. The vertical line shows the 95% confidence interval for the mean gravel size at a distance of 5 km downstream (calculated using equation 5.17).*

In the above example we considered a single value of X_0, however, if we consider a collection of values spanning the interval between the minimum and maximum of X we can draw a band around the regression line that shows how the confidence interval on the mean changes as a function of distance downstream (Figure 5.9). Let's do this in R.

Example code: 24

```
> X0=seq(min(dist),max(dist),0.1) #sequence of distances at a 0.1 km spacing
> C=t*s*sqrt(1/n+(X0-mean(dist))^2/sum(x^2)) #half-width of the CI
> mu0=a+b*X0 #prediction of means at various X0
> mu0_low=mu0-C #lower values of the 95% CI
> mu0_up=mu0+C #upper values of the 95% CI
> plot(dist,size,col='black',xlab='Distance [km]',ylab='Gravel size [phi]')
> lines(dist,Yhat,col='black') #plot the regression line
> lines(X0,mu0_low, lty=2) #plot the lower CI for the predicted means (dashed)
> lines(X0,mu0_up, lty=2) #plot the upper CI for the predicted means (dashed)
```

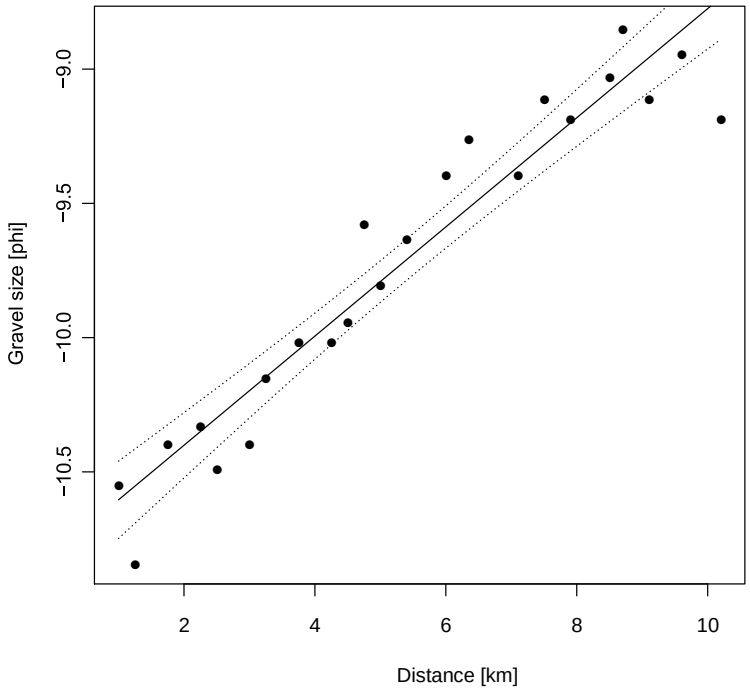


Figure 5.9: *Regression line showing the linear relationship relating gravel size to distance downstream. The dashed lines shows the band of 95% confidence intervals for the mean gravel size downstream, calculated using equation 5.17 and a collection of X_0 values.*

Let's think about the band defined by the confidence intervals in more detail. First, we can see that the band gets wider towards the edges of the data (i.e., low values of X and high values of X). This is because as we move away from the center of the data our predictions of the mean contain larger uncertainties. This level of uncertainty is controlled by the term $(X_0 - \bar{X})^2$ in equation 5.17, which we can see will yield larger values (and thus wider confidence intervals) the more removed X_0 is from X. The second point to note is what would happen to the width of the confidence intervals as we increased the size of the sample, n. If we imagine that we had an infinitely large sample, *i.e.*, the whole population, we can see that equation 5.17 would become:

$$\mu_0 = (a + bX_0) \pm t_{0.025} s \sqrt{0} \qquad (5.18)$$

so the uncertainty disappears and we can make a perfect prediction of μ_0. Although the lab work associated with measuring the size of infinity pieces of gravel may be a challenge this demonstrates that as n gets larger the size of the confidence interval will decrease. This shouldn't be too surprising to you because clearly the larger the sample size the better it will approximate the natural system. Of course we can also use equation 5.17 in an alternative manner if we need to make predictions of the mean grain size with a certain level of uncertainty. Beginning with my original sample I can estimate the value of n that would be required to make a prediction within a confidence interval of a certain size and then increase the size of my sample appropriately.

5.2.5.2 Predicting a single future observation

Imagine after I've collected my river gravels and performed a regression analysis, I decide to return to the river to collect one more piece of gravel at a distance X_0 downstream. Clearly at any given point along the river the gravels will have a distribution of sizes and this needs to be included in our analysis. The best prediction of the size of this new piece of gravel is still the mean given by the regression model, but now we have to include an uncertainty relating to the distribution of gravel sizes at a given location. The equation for the predication interval for a single future observation is:

$$\mu_0 = (a + bX_0) \pm t_{0.025} s \sqrt{\frac{1}{n} + \frac{(X_0 - \bar{X})^2}{\sum x^2} + 1}. \quad (5.19)$$

Again we'll perform the calculation in R for the gravel data at a collection of positions, X_0, and add the prediction intervals to the existing plot as dotted lines (Figure 5.10).

Example code: 25

```
> X0=seq(min(dist),max(dist),0.1) #sequence of distances at a 0.1 km spacing
> P=t*s*sqrt(1/n+(X0-mean(dist))^2/sum(x^2)+1) #half-width of the PI
> mu0=a+b*X0 #prediction of mean at X0
> mu0_low=mu0-P #lower value of the 95% PI
> mu0_up=mu0+P #upper value of the 95% PI
> lines(X0,mu0_low, lty=3) #plot the lower Prediction Interval (dotted)
> lines(X0,mu0_up, lty=3) #plot the upper Prediction Interval (dotted)
```

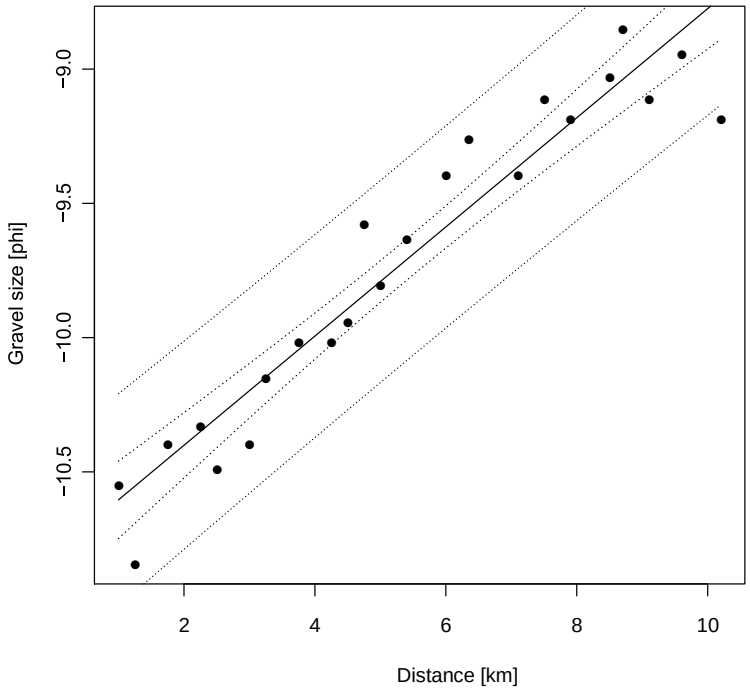


Figure 5.10: *Regression line showing the linear relationship relating gravel size to distance downstream. The dashed lines show the band of 95% confidence intervals for the mean gravel size downstream, calculated using equation 5.17 and a collection of X_0 values. The dotted lines show the 95% prediction intervals for the gravel size of a single future observation, calculated using equation 5.19.*

You'll see that equation 5.19 looks very similar to equation 5.17, but notice the +1 inside the square-root. This term has an important effect as we increase n. Again, for an infinitely large sample, equation 5.19 would become:

$$\mu_0 = (a + bX_0) \pm t_{0.025} s \sqrt{1,}$$

so we can see the prediction interval will never be reduced to 0, no matter how many samples are included in the analysis. At first this may seem a bit odd because by considering an infinitely large sample you would think that we had removed all uncertainty, but this is not the case. As mentioned above, we know that at a given distance along the river not all the gravel particles will have the same size. Therefore, we will be selecting a single piece from a distribution of sizes and thus there is a level of uncertainty associated with the prediction. A further example of this is given in Figure 5.11 where n is increased for an artificial data set. You can see that as n increases the confidence interval on the mean gets smaller, but the prediction interval stays approximately the same width. This is because the data naturally exhibit a scatter around the mean that is not influenced by n and the prediction interval has to take this scatter into account.

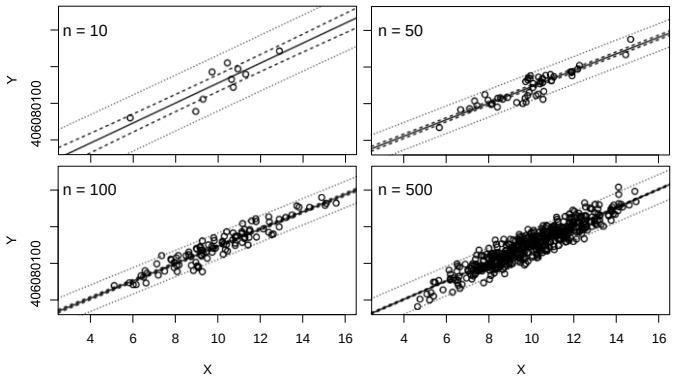


Figure 5.11: *Notice how with increasing sample size (denoted by n) the width of the confidence interval of the prediction of the mean (dashed lines) decreases and the level of uncertainty is reduced. In contrast the prediction intervals on a future observation (dotted lines) remain approximately constant because they have to incorporate the uncertainty created by the scatter of the data. This scatter is a natural property of the data and is therefore independent of sample size.*

5.2.5.3 Make sure you choose the correct interval

There is often confusion as to which kind of interval (confidence interval on the mean or prediction interval) to use in an analysis. You'll commonly find that people use confidence intervals on the mean when in fact they should be using the larger predication intervals. This inappropriate use of the confidence interval on the mean gives the impression that the data has less uncertainly associated with it and can lead to spurious inferences. Think carefully about what questions you are asking of your regression model and make sure that you use the correct form of interval.

5.2.6 Choosing the independent (X) and dependent (Y) variables

In the case of our river gravels it is clear that the independent variable is the distance downstream and the dependent variable is gravel size. In many natural data sets, however, the choice of which variable is independent and which is dependent is less obvious, but it can have important implications for the outcome of the regression analysis. We'll investigate this problem using a classic data set collected by Fisher. The data consist of measurements of irises, specifically the length and width of the sepals of 50 individuals.

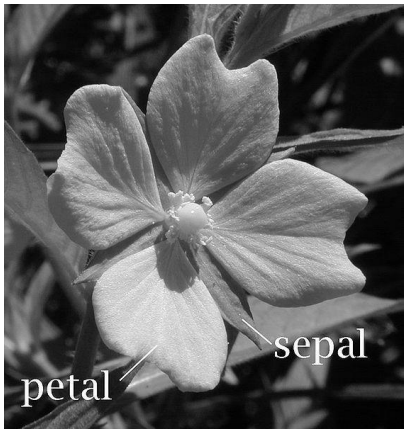


Figure 5.12: *The sepals of a flower lie between the petals.*

With this type of data it's very difficult to say which is the independent variable and which is the dependant variable (if we can even claim that such a relationship exists at all). We'll start by loading the data from the file iris regression.Rdata and plot len (sepal length) as the independent variable and wid (sepal width) as the dependant variable (Figure 5.13). In the examples above we calculated the various correlation and regression parameters from scratch, but now we have a chance to use some of the functions built into R to make things a bit easier.

Example code: 26

```
> rm(list=ls()) #clear the memory
> dev.off() #close the figure windows
> load('iris_regression.Rdata') #load the data file
> plot(len,wid,xlab='Length [cm]',ylab='Width [cm]') #plot the variables
```

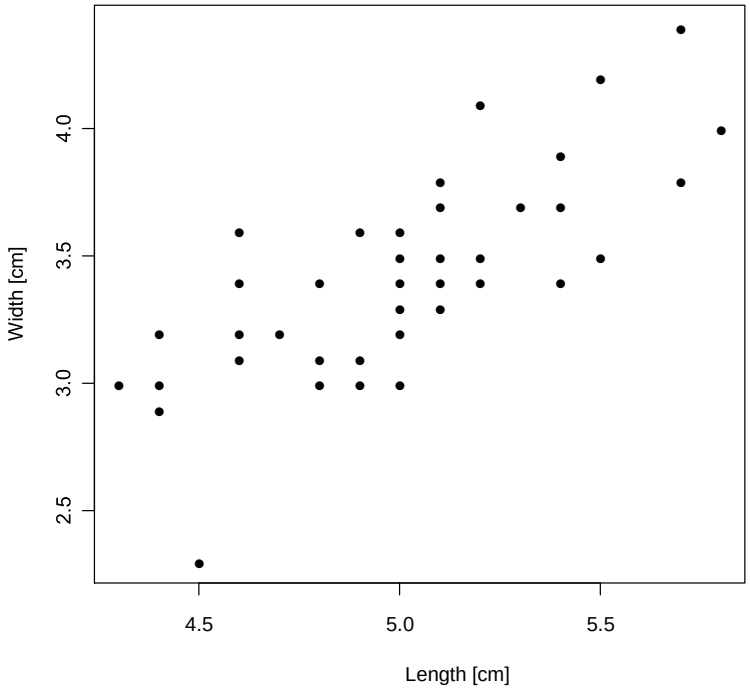


Figure 5.13: *The sepal data set from 50 of the irises measured by Fisher.*

The first thing we need to test is if there is a significant correlation between sepal length and width. We can do this using the hypothesis test outlined in Section 5.1.4.

Example code: 27

```
> r=cor(len,wid) #calculate the sample correlation coefficient
> r #show the correlation coefficient on screen
```

```
> n=length(wid) #the number of observations in the data set
> t=r*sqrt((n-2)/(1-r^2)) #statistic for significance of correlation
> alpha=0.05 #set the significance level for the test > tcrit=qt(1-alpha/2,n-2) #critical value for significance of correlation
> t #show the test statistic on screen
> tcrit # show the critical value on screen
```

The correlation coefficient is $r = 0.74$ and we've found that the corresponding test statistic (7.7) is larger than the critical value (2.0), therefore the correlation between sepal length and width is significant at the $\alpha = 0.05$ level. We can now find the regression relationship between sepal length and width. To do this we'll use the function lm, which in R is the standard approach to building linear models. We'll then add the regression line to the plot (Figure 5.14).

Example code: 28

```
> model=lm(wid~len) #form a linear model to predict wid from len
> model$coefficients #display the model coefficients
> X0=sort(len) #sort length values to make predictions from regression
> Yhat=model$coefficients[2]*X0+model$coefficients[1]
> lines(X0,Yhat) #add regression line to the plot
```

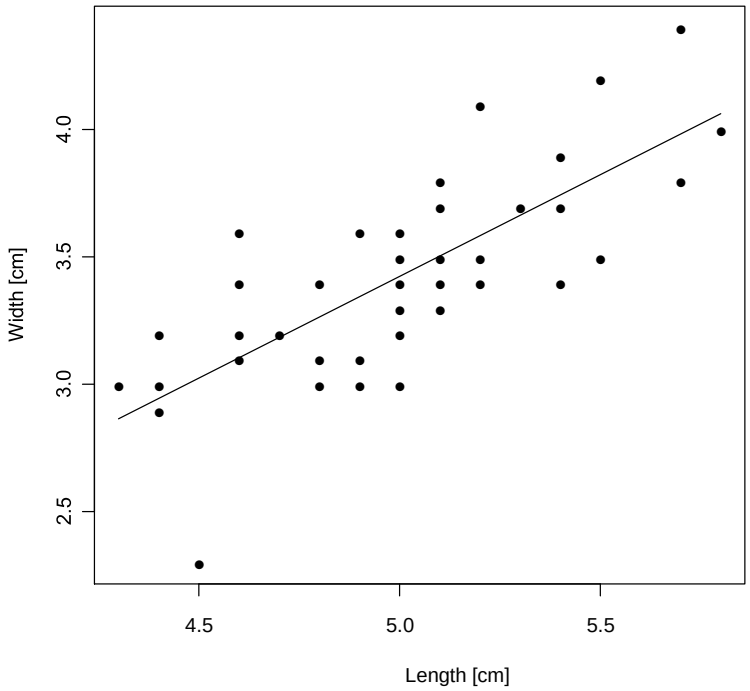


Figure 5.14: *The sepal data set from 50 of the irises measured by Fisher. The line shows the regression relationship between sepal length (assumed to be the independent variable) and width (assumed to be the dependant variable).*

We can ask R to return a full summary of the regression model stored in model using the function summary.

Example code: 29

```
> summary(model) Call:
lm(formula = wid ~ len)

Residuals:
      Min       1Q   Median       3Q      Max
-0.72394 -0.18273 -0.00306 0.15738 0.51709

Coefficients:
                  Estimate Std. Error t value Pr(>|t|) (Intercept) -
0.5694 0.5217 -1.091  0.281 len         0.7985 0.1040 7.681 6.71e-
10 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1                    1
Residual standard error: 0.2565 on 48 degrees of freedom
Multiple R-squared: 0.5514,                    Adjusted R-squared: 0.542
F-statistic: 58.99 on 1 and 48 DF, p-value: 6.71e-10
```

We can now make predictions of sepal width based on the values of sepal length with the regression equation:

$$width = 0.7985 * length - 0.5694 \quad (5.20)$$

What will happen if we try the regression the other way around. Rather than predicting the width from the length, we want to predict the length from the width. We'll start by switching the variables so that wid is the independent variable and len is the dependent variable and then we'll calculate the correlation coefficient.

Example code: 30

```
> r=cor(wid,len) #calculate the sample correlation coefficient
> r #show the correlation coefficient on screen
```

We find that the *r* value (0.74) for wid versus len is identical to the *r* value for len versus wid. Therefore switching the assignment of the independent and dependant variables does not alter the correlation and we know the correlation remains significant. Now we'll find and plot the regression equation to predict length from width, again using the lm function (Figure 5.15).

Example code: 31

```
> plot(wid,len,xlab='Width [cm]',ylab='Length [cm]') #plot the variables
> model=lm(len~wid) #form a linear model to predict len from wid
> model$coefficients #display the model coefficients
> X0=sort(wid) #sort width values to make predictions from regression
> Yhat=model$coefficients[2]*X0+model$coefficients[1]
> lines(X0,Yhat)
> summary(model)

Call: lm(formula = len ~ wid)

Residuals:
        Min          1Q      Median          3Q         Max
-0.52476 -0.16286 0.02166 0.13833 0.44428

Coefficients:
                    Estimate Std. Error t value Pr(>|t|) (Intercept)
          2.6390 0.3100 8.513 3.74e-11 *** wid         0.6905 0.0899
          7.681 6.71e-10 ***
---
           Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1               1

Residual standard error: 0.2385 on 48 degrees of freedom
    Multiple R-squared: 0.5514,                Adjusted R-squared: 0.542
```

```
F-statistic: 58.99 on 1 and 48 DF, p-value: 6.71e-10
```

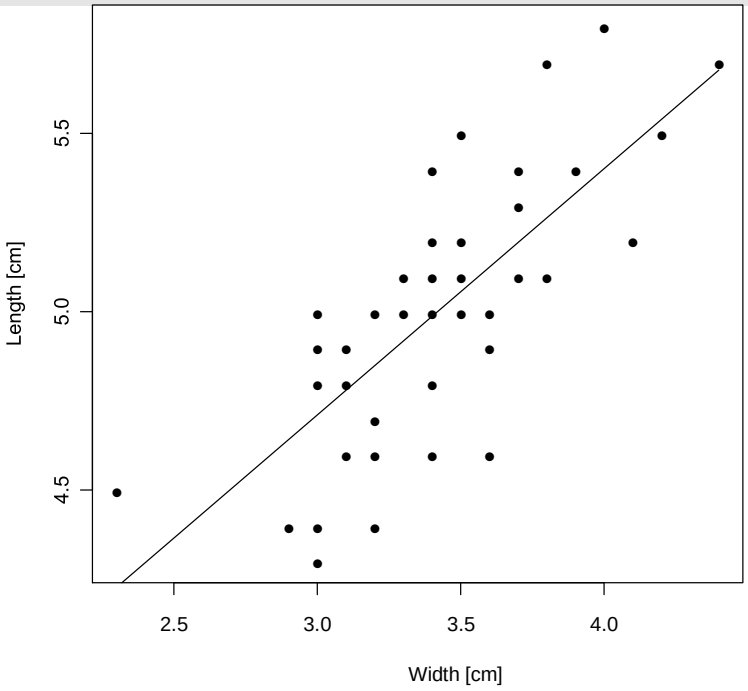


Figure 5.15: *The sepal data set from 50 of the irises measured by Fisher. The line shows the regression relationship between sepal width (assumed to be the independent variable) and length (assumed to be the dependent variable).*

We now have a regression equation to predict sepal length from sepal width:

$$length = 0.6905 * width - 2.6390 \quad (5.21)$$

Now we have to ask ourselves an important question; are the regression relationships given in equations 5.20 and 5.21 equivalent? We can test this by rewriting equation 5.21 to make a prediction of sepal width:

$$length = 0.6905 * width - 2. \quad 6390 \quad (5.22)$$

$$\frac{length - 2.6390}{0.6905} = width \quad (5.23)$$

Now that both regression equations are written in terms of sepal length to make predictions about sepal width we can compare the regression lines on the same plot (Figure 5.16).

Example code: 32

```
> rm(list=ls()) #clear the memory
> dev.off() #close the figure windows
> load('iris_regression.Rdata') #load the data file
> plot(len,wid,xlab='Length [cm]',ylab='Width [cm]') #plot the variables
> Xlen=c(min(len),max(len)) #min and max values of the length
> lines(Xlen,0.7985*Xlen-0.5694,lty=2) #plot first line as dashed
> lines(Xlen,(Xlen-2.6390)/0.6905,lty=3) #plot second line as dotted
```

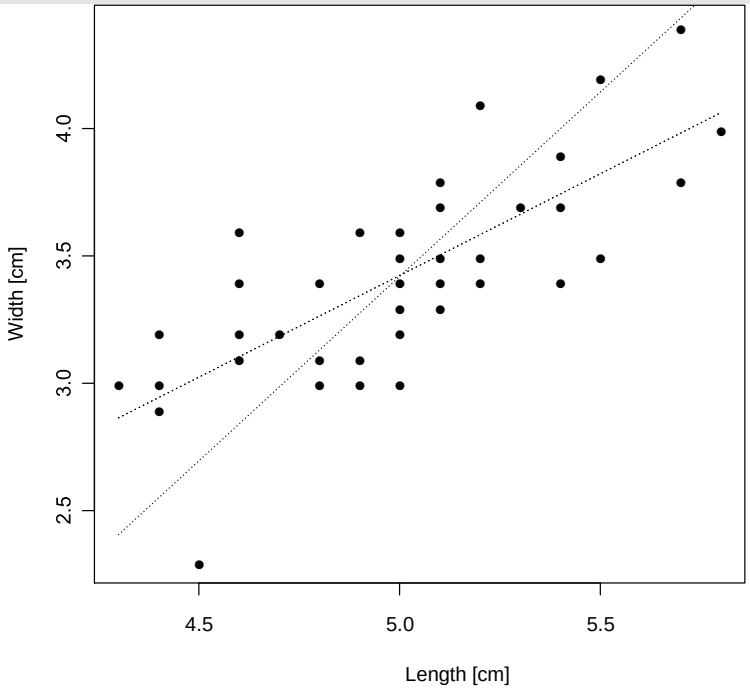


Figure 5.16: *Comparison of the two regression equations obtained with sepal length as the independent variable (dashed) and sepal with as the independent variable (dotted). The two lines are different so we can see that the regression equations are not equivalent.*

We find that the two regression equations are not equivalent and the analysis is sensitive to the selection of the independent and dependant variables. We can see why this effect occurs if we look back at the way the regression line is calculated. In Figure 5.5 we saw that the best-fit regression line is found by minimizing the sum of the squared residuals, which represent the errors in *Y* but the possibility of errors in *X* is not considered. This also fits with the assumption we stated earlier that the errors associated to *X* should be orders of magnitude less than those on *Y* (in other words the errors in *X* are unimportant compared to those in *Y*). So the regression is calculated by minimizing the residuals in *Y* and does not consider the *X* direction. Therefore when we switch the variables on the *X* and *Y* axes we will obtain a different regression equation (Figure 5.17).

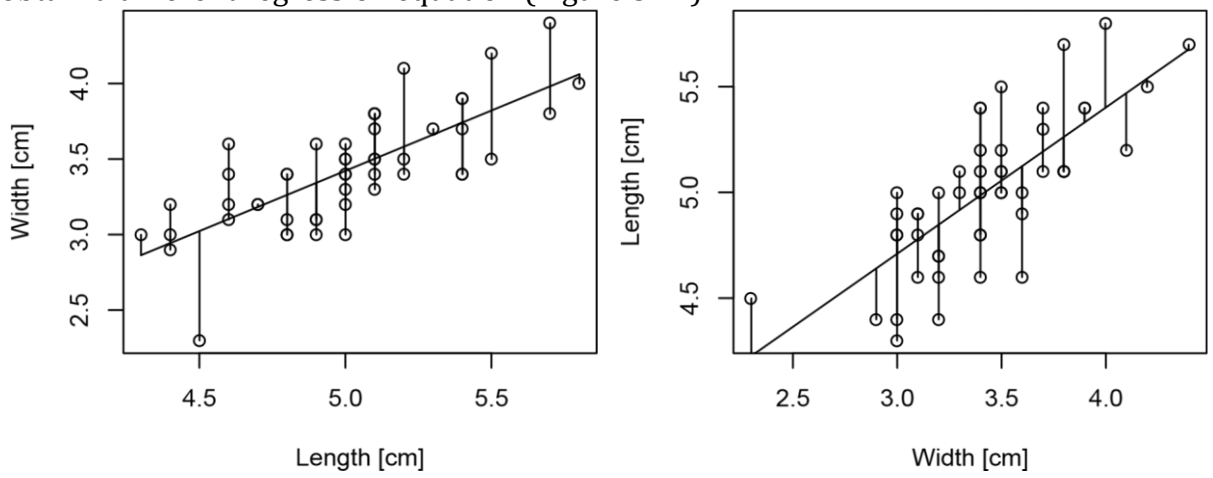


Figure 5.17: *Depending on how the independent and dependant variables are selected, the errors which are minimized are different and therefore different regression equations are obtained.*

5.2.6.1 The Reduced Major Axis line

One solution to the problem of the regression equation depending on an arbitrary assignment of the independent and dependent variables is to use an alternative technique that considers errors in both the *X* and *Y* directions. Rather than minimizing with respect to *Y* the reduced major axis (RMA) considers variation in both *X* and *Y* by minimizing the total areas of the triangles that connect each data point to the regression line (Figure 5.18).

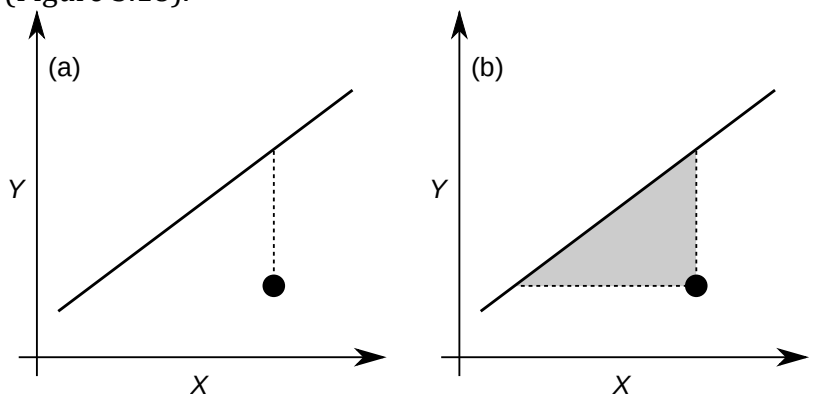


Figure 5.18: *Comparison of the (a) least-squares and (b) RMA approaches. Least-squares is based on the differences (dashed line) between the model and data in the Y direction. RMA is based on the areas of the triangles (shaded) defined by the differences between the model and data in both the X and Y directions.*

As before the RMA is expressed using a straight line; $Y = a + bX$, where

$$b = s_Y / s_X, \qquad (5.24)\ a = \bar{Y} - b\bar{X}, \qquad (5.25)$$

where s_Y represents the standard deviation of the Y variable and s_X represents the standard deviation of the X variable. Returning to Fisher's irises, we can perform the RMA analysis in R and include the resulting line in our plot (Figure 5.19).

Example code: 33

```
> b=sd(wid)/sd(len) #find the slope of the RMA line
> b #show the slope value on screen
> a=mean(wid)-b*mean(len) #find the intercept of the RMA line > a #show the intercept value on screen
> Xlen=c(min(len),max(len)) #min and max values of the length
> lines(Xlen,b*Xlen+a,lty=1) #plot the RMA line as solid
```

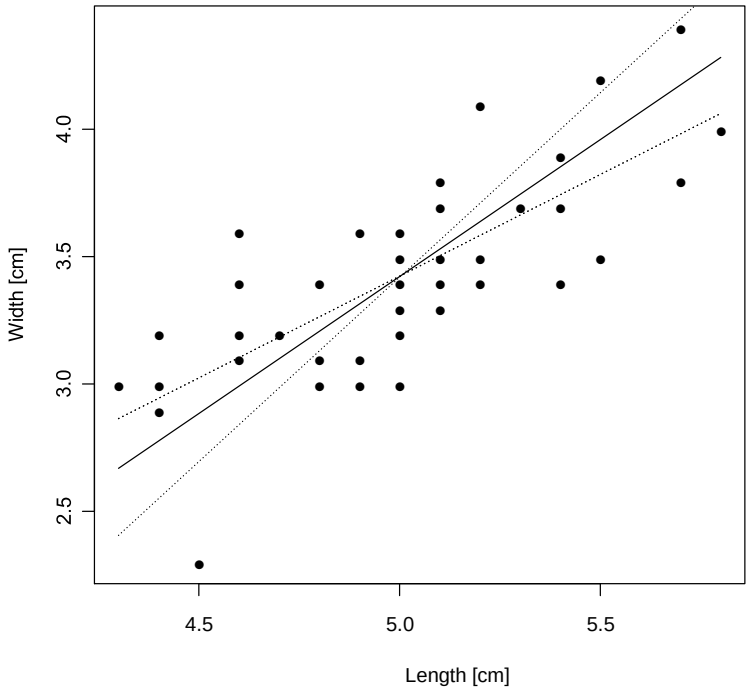


Figure 5.19: *Comparison of the two regression equations obtained with sepal length as the independent variable (dashed) and sepal width as the dependent variable (dotted). The two lines are different so we can see that the regression equations are not equivalent. The RMA line is shown as a solid line.*

The RMA line lies between the two other regression estimates and has the equation:

$$width = 1.0754 * length - 1.9554 \quad (5.26)$$

But what happens when we switch *X* and *Y* again?

Example code: 34

```
> b=sd(len)/sd(wid) #find the slope of the RMA line
> b #show the slope value on screen
> a=mean(len)-b*mean(wid) #find the intercept of the RMA line
> a #show the intercept value on screen
```

So when we swap the axes we obtain the RMA line:

$$length = 0.9299 * width - 1.8183 \quad (5.27)$$

If we rearrange equation 5.27 to make predictions of sepal width on the basis of sepal length we get:

$$length = 0.9299 * width - 1. \quad 8183 \quad (5.28)$$

$$width = \frac{length - 1.8183}{0.9299} \quad (5.29)$$

$$width = 1.0754 * length - 1. \quad 9554 \quad (5.30)$$

Compare this result to equation 5.26 and we can see the RMA line is not influenced by our selection of *X* and *Y* . One drawback of the RMA approach, however, is the calculation of confidence and prediction intervals is not so straight forward. A paper by Warton (see recommended reading) gives a good discussion of the different approaches to regression and how the different techniques can be applied.

6 Multiple linear regression

In the previous chapter we looked in detail at correlation and regression when there was a linear relationship between two variables. For obvious reasons, the case of two variables is called a *bivariate* problem and now we'll extend the idea to *multivariate* problems where we consider more variables.

6.1 Moving to higher dimensions

In mathematics, dimensions are the parameters required to describe the position and relevant characteristics of any object within a conceptual space. In this way the dimensions of a space are the total number of different parameters used for all possible objects considered in the model. Let's try to make this clearer with some examples.

Imagine I have a data set composed of a single measured parameter and that for the sake of simplicity I know that my measurements will always lie between 0 and 1. Because I only have one parameter in my data I only need a 1D representation, which is a straight line. So I could represent the data by plotting their positions along a line (Figure 6.1) that starts at the coordinate (0) and finishes at (1).

Now imagine that I measure an additional parameter (again for simplicity assumed to be between 0 and 1) and combine it with my original parameters to make a 2D data set. If I'm now going to represent my 2D data in a plot (Figure 6.1) I need place the points within a square with the corners positioned at the coordinates (0,0), (1,0), (0,1) and (1,1).

We can extend this basic principal further. Each time I add a new parameter, and thus a new dimension, to my data set I simply add an extra coordinate to the space that I require to represent the data (Figure 6.1). So a 3D data set is represented in a cube with eight corners; (0,0,0), (1,0,0) through to (1,1,1). But, what would happen if I have a 4D data set? In such a case we'll require a 4D space that has 16 corners ranging from (0,0,0,0) to (1,1,1,1). This 4D shape is known as a *hypercube* and although we can't see what a 4D cube looks like without exploding our 3D brains, we can easily represent it in a 4D conceptual mathematical space.

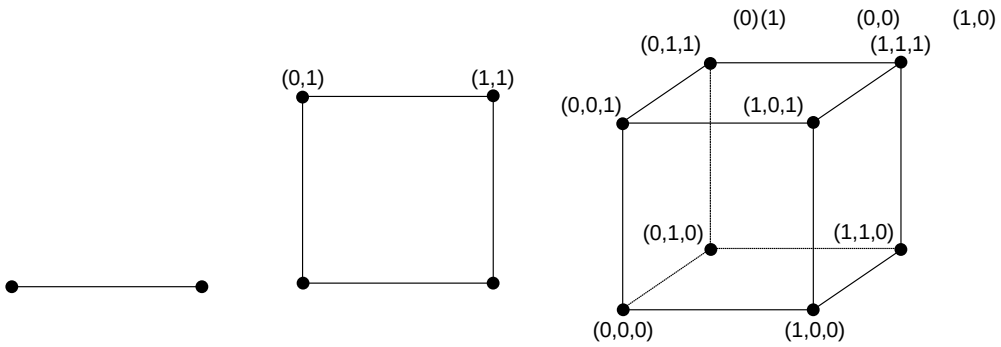


Figure 6.1: *How different numbers of dimensions can be represented using increasingly complex coordinate systems.*

We can continue this process of adding as many dimensions as we want (and some problems require a lot of dimensions) and we never hit a limit. Instead, we just add an extra coordinate each time we need one. Just to give you some idea of how extreme these things can get, mathematicians have discovered a symmetrical shape called "The Monster" that looks a bit like a snowflake and exists in 196883 dimensional space!

Although we can't draw or build a 4D hypercube, we can attempt to represent it using projections. For example the cube drawn in Figure 6.1 is a projection of a 3D body on to a 2D surface (a piece of paper). Similarly we can project a 4D hypercube into 2D. Such a projection is shown in Figure 6.2 and you should be able to tell immediately that it would be difficult to interpret data plotted in such a projected space. If that's the case for 4D, just imagine what would happen for more complicated data sets with higher numbers of dimensions.

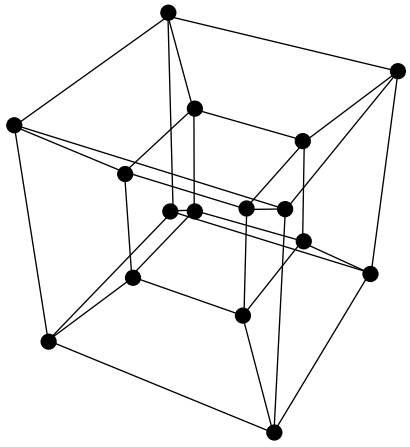


Figure 6.2: *A 2D projection of a 4D hypercube (a tesseract).*

If you would like to see what a 4D hypercube looks like projected into 3D you should visit the Arche de la D´efense in Paris. Alternatively, if you would like to see a 2D representation of a 3D projection of a 4D hypercube then you can look at Figure 6.3.

Figure 6.3: *La Grande Arche de la D´efense in Paris*

Before we finish with this section it's important to note that many statistical techniques rely on calculating the distance between points (we'll look at this in more detail later). Of course in 2D we can calculate the straight-line distance between points using Pythagoras' theorem, whereby we draw a straight-line along the *X* dimension and a straight-line along the *Y* dimension to form a right angled triangle and then the distance is given by the length of the hypotenuse. This procedure can be extended to 3 or more dimensions, so for any given number of dimensions we can calculate the distance between two points using Pythagoras' theorem (Figure 6.4).

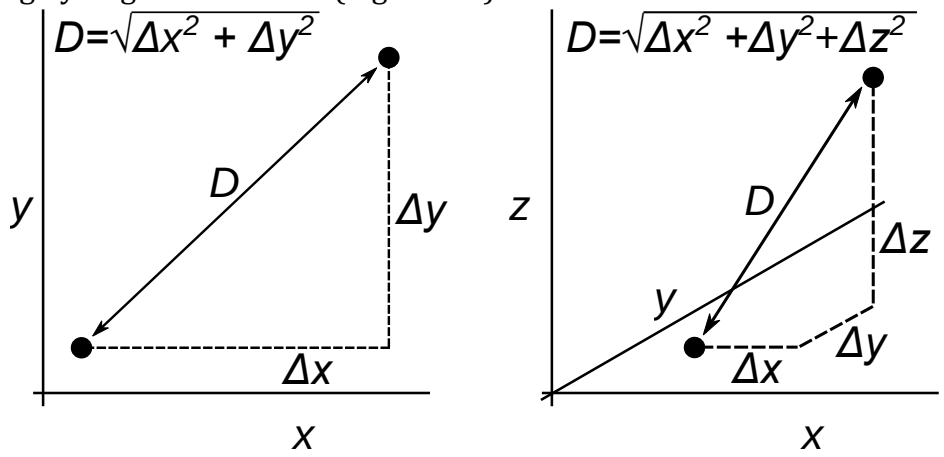


Figure 6.4: *Pythagoras' theorem works in 2 (left), 3 (right) or as many dimensions as you want.*

6.2 The basics of multiple linear regression

Multiple linear regression (MLR) is simply a linear regression with more than one independent variable. For example, what happens when *Y* is controlled by >1 independent variables? In the 3 dimensional case there are two independent variables, which we'll call X_1 and X_2, so the regression equation to relate them to *Y* then becomes:

$$\hat{Y} = b_1X_1 + b_2X_2 + a \tag{6.1}$$

Such an equation defines a plane, where *a* is the intercept on the *Y* -axis (the value when both X_1 and X_2 are zero) and b_1 and b_2 are the slopes in the X_1 and X_2 directions, respectively (Figure 6.5). Let's consider

the next case, what happens when Y is controlled by 3 independent variables? Simple, we just extend the regression equation to include the new independent variable, X_3, which gives the 4-dimensional case

$$\hat{Y} = b_1X_1 + b_2X_2 + b_3X_3 + a \qquad (6.2)$$

In this case the regression equation defines a hyperplane that, just like a hypercube, we cannot visualize easily. However, the method for fitting (hyper-)planes to data with more than one regressor variable is the same as bivariate (X,Y) regression, namely; least-squares. We'll examine this problem in more detail in R. First we'll generate some artificial data with a known structure (so we know the result we should get in advance) and then we'll build a regression model using the lm function.

The first step is to create 2 independent variables, X_1 and X_2. To do this we'll generate 2 sets of random numbers in R, each consisting of 500 values. Then we'll calculate the dependent variable, Y, based on an arbitrary relationship: $Y = 0.5X_1 + 1.25X_2 + 0.9$.

Example code: 35

```
> rm(list=ls()) # clear the memory
> dev.off() # close all the existing graphics
> X1=runif(500) # 500 random numbers for X1
> X2=runif(500) # 500 random numbers for X2
> Y = (0.5*X1) + (1.25*X2) + 0.9 # produce values of Y
```

We can now plot the data. To do this we'll have to download the scatterplot3d package and build it.

Example code: 36

```
> install.packages('scatterplot3d') #install the package for 3D plots
> library(scatterplot3d) #initialize the package
> scatterplot3d(X1,X2,Y) #plot X1 and X2 vs. Y
```

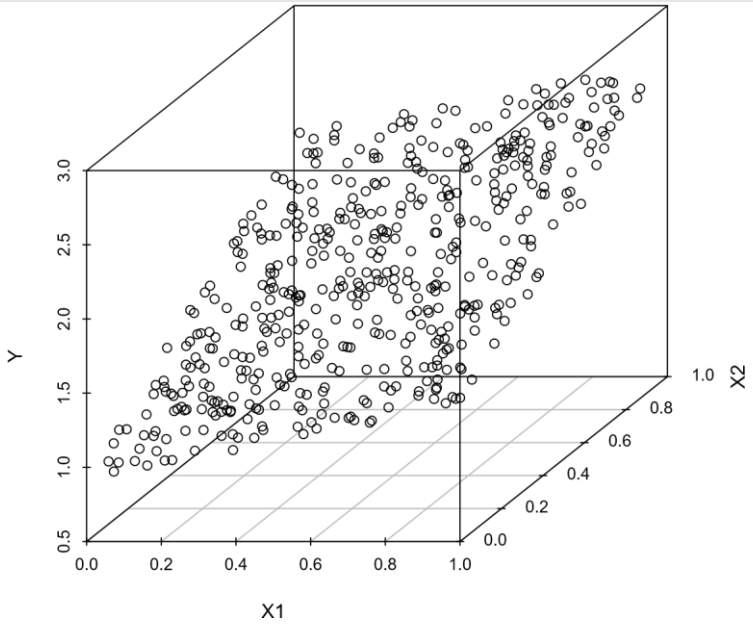


Figure 6.5: *Scatter plot of the dependent variable Y formed as a function of the independent variables X_1 and X_2.*

Next, we'll build a MLR model using the lm function and then look at the coefficients of the fitted equation.

Example code: 37

```
> fit=lm(Y~X1+X2) #form the MLR model with Y dependant on X1 and X2
> coefficients(fit) #coefficients of the best fit model

(Intercept)          X1          X2
       0.90        0.50        1.25
```

You should see that the values of the coefficients match those we used to build the Y values. Although this is no great surprise, it does demonstrate how we can use the lm function for MLR.

6.2.1 Identifying significant regressors and making predictions

When working with real data sets we encounter a problem in that some of the independent parameters may control the dependant parameter, but others may not. To understand the MLR model properly we must find which variables are important and which are unimportant. Therefore, we need statistical parameters that provide information on the quality of a MLR and how important each variable is in the analysis. Fortunately, the lm function provides us with this information, which we can request with the summary command.

We'll form a regression model similar to the one above where we construct a data set and then investigate the information returned by lm. We'll start by generating 4 independent parameters, with each one composed of 500 random numbers. To make things more realistic we'll also add some random numbers into the system which will act like measurement noise.

Example code: 38

```
> rm(list=ls()) # clear the memory
> dev.off() # close all the existing graphics
> X1=runif(500) # 500 random numbers for X1
> X2=runif(500) # 500 random numbers for X2
> X3=runif(500) # 500 random numbers for X3
> X4=runif(500) # 500 random numbers for X4
> E=rnorm(500)*0.1 # 500 normally distributed random numbers
```

We'll now build the dependent variable, Y, using the relationship $Y = 0.5X_1+1.25X_2+1.05X_4+0.9$ and then add on the errors. Notice that we only use 3 out of the 4 independent variables. Since X_3 is not included, there should be no significant relationship between X_3 and Y. Finally, we'll add the simulated random errors to Y.

Example code: 39

```
> Y=(0.5*X1)+(1.25*X2)+(1.05*X4)+0.9 # form Y using only X1, X2 and X4
> Y=Y+E #add the simulated errors into Y
> fit=lm(Y~X1+X2+X3+X4) # form the MLR model
> coefficients(fit) #coefficients of the best fit model
```

You should be able to see that the coefficients for X_1, X_2 and X_4 are close to those that we used in the construction of Y, but don't match perfectly. The reason for the slight mismatch in the coefficients is the errors we added into Y, which means that our estimate, b, of the true model parameters, β, is not perfect. The coefficient for X_3 is small, which implies that the influence X_3 has on Y is small. We can obtain the Y values predicted by the returned linear model using the function fitted and the information stored in the variable fit (Figure 6.6).

Example code: 40

```
> Yhat=fitted(fit)
> plot(Y,Yhat,xlab='Y',ylab='Predicted Y')
```

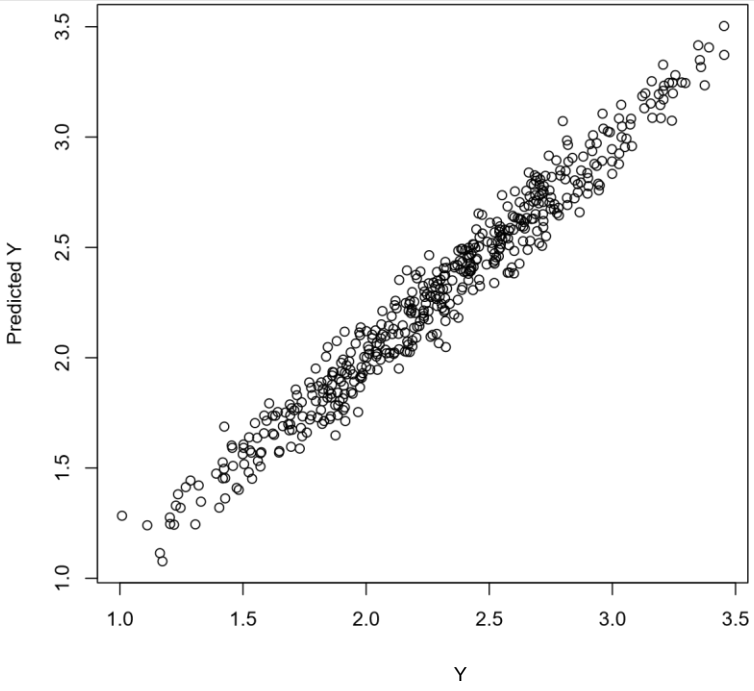


Figure 6.6: *Scatter plot of the dependent variable Y against the predictions, Y^, made by the MLR model.*

Clearly we'll have to extend the statistical analysis further if we are to obtain information about β rather than simply b. Fortunately, the lm function can give us more information about which coefficients are important in the regression model. This information can be obtained using the summary command.

Example code: 41

```
> summary(fit)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4)

Residuals:
      Min       1Q   Median       3Q      Max
-0.302141 -0.073881 0.005527 0.071022 0.287857

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.91170 0.01585 57.536 <2e-16 *** X1 0.49786 0.01526
32.628 <2e-16 ***
X2 1.25338 0.01579 79.374 <2e-16 *** X3 0.01457 0.01562 0.933
0.351
X4           1.01231      0.01501 67.455 <2e-16 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1                  1

Residual standard error: 0.09745 on 495 degrees of freedom Multiple R-
squared: 0.9598,    Adjusted R-squared: 0.9595
F-statistic: 2955 on 4 and 495 DF, p-value: < 2.2e-16
```

You can see the estimates (Estimate) and standard errors (Std. Error) of the regression coefficients, b, listed in the table. We can use these values to estimate confidence intervals on the true regression coefficients, β. Again we'll need to use Student's t distribution, this time with $n - k$ degrees of freedom, where n is the number of data points and k is the number of regressors (including the intercept). Then the 95% confidence interval of a given β coefficient can be estimated as:

$$\beta = b \pm t_{0.025}SE \qquad (6.3)$$

where SE is the quoted standard error. For example, finding the 95% confidence interval associated with the X_4 coefficient could be obtained in R by:

Example code: 42

```
> b=fit$coefficients[5] # get the coefficient for X4
> SE=summary(fit)$coefficients[5,2] # standard error for the coefficient
> n=length(Y) # number of data points in the model
> k=length(fit$coefficients) # number of regressors in the model
> tval=-qt(0.025,n-k) # value from t distribution with n-k dof
> beta_low=b-tval*SE # calculate the lower value of the confidence interval
> beta_up=b+tval*SE # calculate the upper value of the confidence interval
```

When you look at the values in beta_low and beta_up you should (hopefully) find that they define an interval which contains the true value of the X_4 coefficient (1.05) which we used when we created the original data. Fortunatly, there is a R function which will calculate the confidence intervals for us. We just need to call confint and define the confidence level we want. As an example we can calculate the 95% confidence value for the model in fit:

Example code: 43

```
> confint(fit,level=0.95) # find the 95% confidence intervals for fit

                2.5 %     97.5 %
(Intercept) 0.88056983 0.94283674
X1          0.46787723 0.52783661
X2 1.22235535 1.28440573 X3 -
0.01612402 0.04526898 X4 0.98282237
1.04179386
```

Remember that we constructed the data using the relationship $Y = 0.5X_1 + 1.25X_2 + 0X_3 + 1.05X_4 + 0.9$ and we can see that the estimated confidence intervals for β span the true values of β (note, your coefficients and confidence intervals might be slightly different to the ones above because you will have used different random numbers in the construction of E, which will change the final model fit very slightly).

We can also test which of the variables make a significant contribution to the model, by finding a t-value given by the ratio of a coefficient to its standard error. We'll do this for the X_3 coefficient (which we know makes no contribution to the dependent variable). We can then compare the t-value to a Student's t-distribution with $n - k$ degrees of freedom to find the significance of the regressor in the model, which is termed a p-value.

Example code: 44

```
> b=fit$coefficients[4] # get the coefficient for X3
> SE=summary(fit)$coefficients[4,2] # standard error for the coefficient
> n=length(Y) # number of data points in the model
> k=length(fit$coefficients) # number of regressors in the model
> pval=2*pt(b/SE,n-k,lower.tail=FALSE) #find the p-value
```

The p-values for each regression coefficient are actually reported in the final column of the table we looked at earlier using the summary function. So the final column of the table Pr(>|t|) tells us which variables make a significant contribution to the regression model. Variables with low numbers in the Pr(>|t|) column are highly significant, whilst those with high numbers are non-significant. For example, if we want to find which variables make a significant contribution to the regression model at the 0.05 level, we would look for variables in the model with values of Pr(>|t|) less than or equal to 0.05. Clearly the lm results from the above example tell us that X_3 is non-significant at the 0.05 level, which is not surprising considering that it was not used in the construction of the Y data.

6.2.1.1 An example: Plate tectonics

In their 1975 paper *"On the relative importance of driving forces of plate motion"*, Forsyth and Uyeda examined the processes that possibly control the speed at which tectonic plates move. They considered 4 different plate properties that could influence plate speed:

- Total plate area.
- Area of continental part.
- Effective length of ridge.
- Effective length of trench.

In this case *effective* length is defined as "the length of the boundary which is capable of exerting a net driving or resisting force. For example, two mid-ocean ridges on opposite sides of a plate exert no net force on the plate because their effects cancel out". Forsyth and Uyeda collected a data set for 12 plates (Figure 6.7).

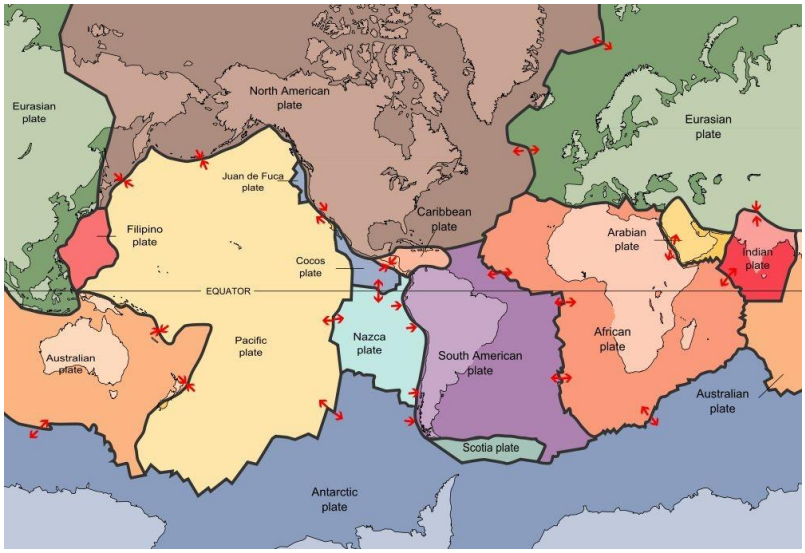


Figure 6.7: *The tectonic plates.*

Plate name	Total area (10^6 km^2)	Continental area (10^6 km^2)	Effective ridge length (10^2 km)	Effective trench length (10^2 km)	Speed (cm/yr)
North American	60	36	86	10	1.1
South American	41	20	71	3	1.3
Pacific	108	0	119	113	8.0
Antarctic	59	15	17	0	1.7
Indian	60	15	108	83	6.1
African	79	31	58	9	2.1

Eurasian	69	51	35	0	0.7
Nazca	15	0	54	52	7.6
Cocos	2.9	0	29	25	8.6
Caribbean	3.8	0	0	0	2.4
Philippine	5.4	0	0	30	6.4
Arabian	4.9	4.4	27	0	4.2

So the big question is which independent variables could control plate motion and in what way? We can perform MLR with the four parameters to see how well they can predict the plate speed. The data are held in a data file called plates.Rdata and the first step is to load them into R.

Example code: 45

```
> rm(list=ls()) # clear the memory
> dev.off() # close all the existing graphics
> load('plates.Rdata') # load the data file
```

There are 5 parameters in the data set; total area is the total area (10^6 km^2), cont_area is the continental area (10^6 km^2), ridge len is the effective ridge length (10^2 km), trench len is the effective trench length (10^2 km) and speed is the plate speed (cm/yr). We can now build the MLR model to find an equation which predicts plate speed based on the four independent parameters.

Example code: 46

```
> fit=lm(speed~total_area+cont_area+ridge_len+trench_len) #perform the MLR >
summary(fit) #view the MLR results

Call: lm(formula = speed ~ total_area + cont_area + ridge_len + trench_len)

Residuals:
      Min       1Q   Median       3Q      Max
-2.00396 -0.41947 -0.05667 0.23691 2.57130

Coefficients:
                  Estimate Std. Error t value Pr(>|t|) (Intercept) 4.54710
      0.73409       6.194 0.000448 *** total_area -0.03767
      0.02211 -1.704 0.132193 cont_area  -0.01529       0.04743 -
0.322 0.756580 ridge_len      -0.01442       0.02067 -0.698 0.507746
trench_len      0.08037       0.02511 3.200 0.015063 *
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1                      1

Residual standard error: 1.401 on 7 degrees of freedom
Multiple R-squared: 0.8597,                  Adjusted R-squared: 0.7796
F-statistic: 10.73 on 4 and 7 DF, p-value: 0.004144
```

From the Pr(>|t|) column it appears that only the intercept and effective trench length play a significant role in the regression. This is confirmed by the confidence intervals on the coefficients, whereby the intervals for the intercept and effective trench length are the only ones not to span 0.

Example code: 47

```
> confint(fit,level=0.95)

                    2.5 %       97.5 %
(Intercept)    2.81125995    6.28293689
total_area    -0.08994543    0.01460916
cont_area     -0.12745309    0.09687012
ridge_len     -0.06329900    0.03444902
trench_len 0.02098227 0.13975114
```

Finally we can plot the measured plate speeds against the speeds predicted by the regression model (Figure 6.8).

Example code: 48

```
> Yhat=fitted(fit) # get the predicted values
> plot(speed,Yhat,xlab='Speed (cm/yr)',ylab='Predicted Speed (cm/yr)')
```

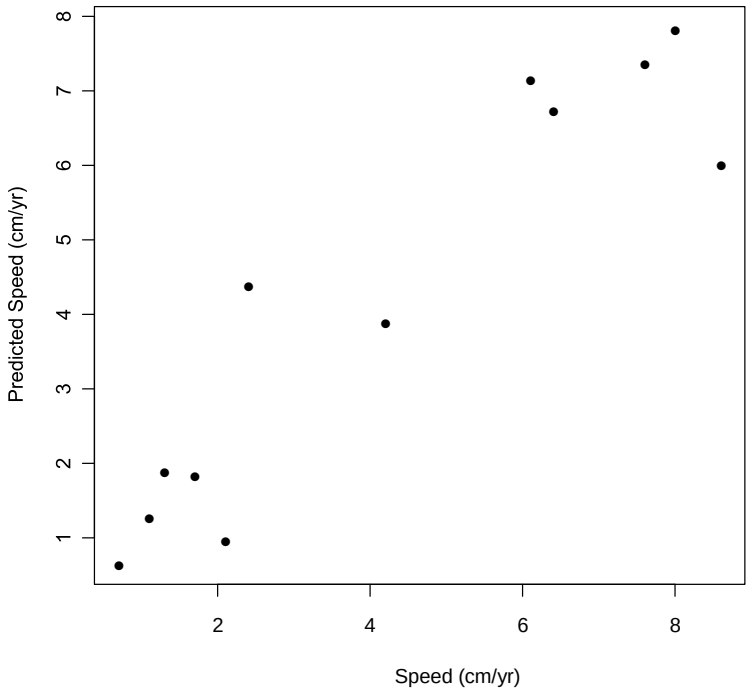


Figure 6.8: *Scatter plot of the measured plate speed against the predictions of plate speed made by the MLR model.*

6.2.2 Multicollinearity

MLR assumes that the regressors are independent of each other (*i.e.,* they exhibit no correlation). If, however, a correlation exists between the regressors then they are not independent and suffer from *multicollinearity*. If there was a perfect correlation between the regressors X_1 and X_2, then an infinite number of MLR solutions would exist that can all explain the data equally well (however, perfect correlation is very rare in geological situations). In practice, if multicollinearity exists the contribution of any regressor to the MLR depends on the other regressors that are already included in the model. Therefore it is important to check for any correlation between regressors before beginning MLR and think about the variables you are using.

We will use R to form a data set with variables which suffer from multicollinearity and then perform a simple MLR. First, we'll set up the independent variables (X_1 and X_2) each consisting of 500 random numbers. We'll make a third regressor, X_3, as a function of X_1 and X_2, thus introducing multicollinearity. Then we'll form the dependent variable, Y , according to the relationship; $Y = 1.3X_1 + 0.8X_2 + 1.1X_3 + 0.5$.

Example code: 49

```
> rm(list=ls()) # clear the memory
> dev.off() # close all the existing graphics
> X1=runif(500) # 500 random numbers for X1
> X2=runif(500) # 500 random numbers for X2
> X3=(0.5*X1)+(0.75*X2)+1.3 # make X3 a function of X1 and X2
> Y=(1.3*X1)+(0.8*X2)+(1.1*X3)+0.5 # calculate the dependent variable
> fit=lm(Y~X1+X2+X3) # form the MLR model
> summary(fit)

Call:
lm(formula = Y ~ X1 + X2 + X3)

Residuals:
          Min            1Q        Median            3Q           Max
-1.801e-14 -1.360e-16 4.220e-17 2.078e-16 7.019e-15

Coefficients: (1 not defined because of singularities)
                        Estimate Std. Error            t value Pr(>|t|)
(Intercept)  1.930e+00  1.259e-16  1.532e+16 <2e-16  ***  X1  1.850e+00
1.603e-16 1.154e+16 <2e-16 ***
X2                    1.625e+00 1.614e-16 1.007e+16          <2e-16 ***
X3                          NA              NA             NA           NA
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1                          1

Residual standard error: 1.047e-15 on 497 degrees of freedom
Multiple R-squared:              1,             Adjusted R-squared:           1
F-statistic: 1.119e+32 on 2 and 497 DF, p-value: < 2.2e-16
```

We've obviously managed to confuse R, the NA values mean *Not Available* because the multicollinearity means a stable solution can't be found. We can also see that the coefficients returned by the model are not close to the ones we used to build the model, this means that we could fundamentally misunderstand the system we're working with. Something surprising happens when we look at the predictions of the regression model (Figure 6.9).

Example code: 50

```
> Yhat=fitted(fit) # get the predicted values
> plot(Y,Yhat,xlab='Y',ylab='Predicted Y values') # plot Y and Yhat
```

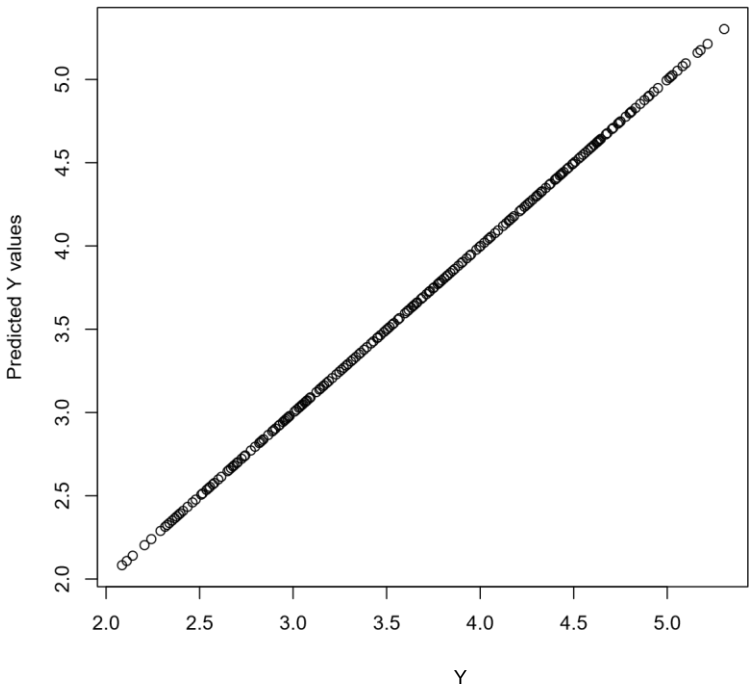


Figure 6.9: *Scatter plot of the Y values against the predictions of Y.*

We can see that although the predicted coefficients are incorrect, the actual predictions of *Y* are perfect. Therefore, the derived relationship between the variables and *Y* is incorrect because of the multicollinearity, but the prediction of *Y* is accurate. This allows us to make predictions but not to quantify the underlying relationship.

Multicollinearity can be difficult to detect in real data sets where the relationship between the regressor variables may be poorly understood. There are however cases such as compositional data where we would expect multicollinearity to occur (Chapter 9). As with bivariate regression it is also important to consider the following points.

1. Are you sure the relationship is linear?
2. Are there any outliers which could strongly influence the regression (influential values can be found using Cook's distance)?
3. Are the data normally distributed?

6.2.3 The taste test

Now it's your turn, 77 breakfast cereals were rated in terms of their taste. Use MLR to find the relationship between taste rating and:

- Calorie content.
- Sodium content.

- Carbohydrate content.
- Potassium content.

The R file cereals.Rdata contains the following variables; rating, calories, Na, carbo and K. Use MLR to find the significant regressors and determine the regression relationship. How does predicted taste compare to the taste rating? Use the code from the previous examples as a template, the solution to the problem is on the next page.

We will assume that any effects of multicollinearity are negligible. Start by performing the regression:

Example code: 51

```
> rm(list=ls()) # clear the memory
> dev.off() # close all the existing graphics
> load('cereals.Rdata') #load the data file
> fit=lm(rating~calories+carbo+K+Na) #form the MLR model
> summary(fit) # model summary

Call:
lm(formula = rating ~ calories + carbo + K + Na)

Residuals:
      Min       1Q   Median       3Q      Max
-13.693  -4.820  -0.966   3.685  24.756

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 65.47364   8.52176   7.683 5.98e-11 ***
calories    -0.54754   0.04566 -11.992  < 2e-16 ***
carbo        1.28172   0.22697   5.647 3.04e-07 ***
K            0.09131   0.01298   7.033 9.59e-10 ***
Na           0.03986   0.03512   1.135     0.26
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1  1

Residual standard error: 7.505 on 72 degrees of freedom
Multiple R-squared: 0.7295,     Adjusted R-squared: 0.7145
F-statistic: 48.56 on 4 and 72 DF,  p-value: < 2.2e-16

> hat=fitted(fit) # find the predicted values
> plot(rating,hat,xlab='Rating',ylab='Predicted Rating') # plot Y and Yhat
```

The results show that the intercept term and the calories, carbo and K variables are significant in the regression model, whilst sodium is not significant.

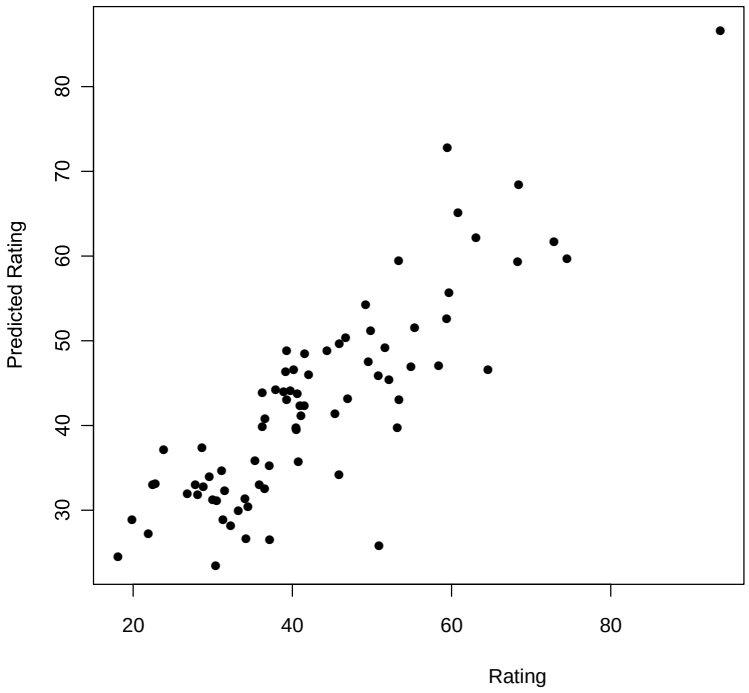


Figure 6.10: *Scatter plot of the taste rating values against the predictions of taste rating.*

The average human has one breast and one testicle.

Des McHale

7 Cluster analysis

In the previous discussion of regression techniques we assumed that variables were related by some form of continuous relationship that could be described by a line or a similar function. In some situations, however, we may expect data to fall into a collection of discrete groups rather than describing a continuous path. An example is given on the left of Figure 7.1, which shows the regression line for the river gravels we studied earlier. Not surprisingly, we found that the change in gravel size as we moved downstream could be described by a continuous line. On the right of Figure 7.1 you can see a very different situation where the data points appear in distinct groups. Would it make sense to fit a straight-line through this data? Clearly not, we would be using a continuous function to try and describe a data set that appears not to be continuous. Instead we can gain insights into the data by trying to find and characterize the groups of data within the sample. To do this we will employ *cluster analysis*.

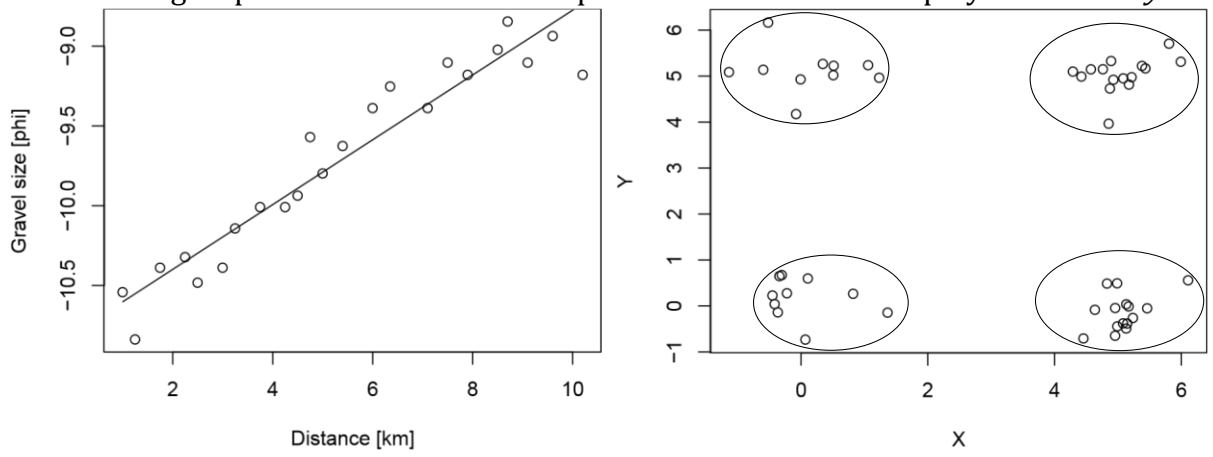


Figure 7.1: *An example of the kinds of data that should be analyzed using regression analysis (left) and cluster analysis (right).*

7.1 The principals behind cluster analysis

Cluster analysis, also called segmentation analysis or taxonomy analysis, is a way to create groups of objects, or clusters, in such a way that the properties of objects in the same cluster are very similar and the properties of objects in different clusters are quite distinct. Cluster analysis is a multivariate technique so it can work in large numbers of dimensions, where each dimension represents one property of the data set. There are a variety of different clustering methods, we'll only be looking at a

small number of them, but they all work on a similar principal, which is to measure the similarity between data points.

7.1.1 Distance as a measure of similarity

We naturally think of distance as a good measure of similarity. For example, look at Figure 7.2, which shows two variables (*X* and *Y*) for 3 data points (*A*, *B* and *C*). If you were asked which point is the most similar to point *A* you would probably say point *B* simply because it is closest. Alternatively, if you were asked which point is the most dissimilar to point *A* you would probably say *C* because it is the point which is furthest away. Therefore, a natural measure of the similarity of points in a parameter space is the distance which separates them.

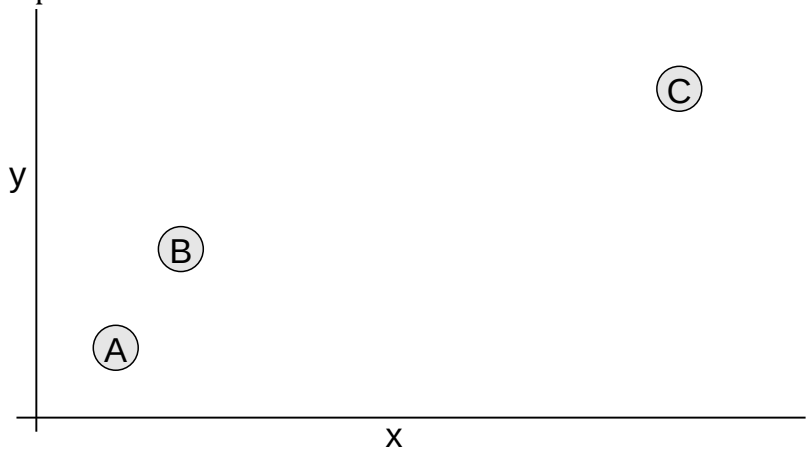


Figure 7.2: *How can we judge which point, B or C, is the most similar to A?*

There are a number of different ways in which we can measure distance. The one we are most familiar with is the Euclidean distance, which is simply the length of a straight-line connecting two points. As we saw in Section 6.1 the Euclidean distance can be calculated for any given number of dimensions using Pythagoras' theorem. There are, however, different ways to measure distance. A simple example is the *Manhattan Street Block* distance, which rather than measuring the shortest straight-line distance between two points, instead sums the absolute differences of their coordinates (Figure 7.3). This is also known as *Taxicab geometry* because if you imagine the grid layout of streets in New York, a taxi driver cannot go in a straight-line between two locations, but instead can just move along north-south avenues or east-west streets.

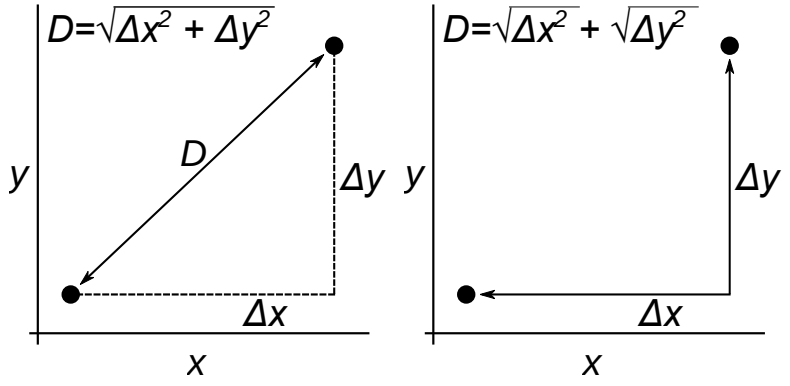


Figure 7.3: *A comparison of Euclidean (left) and Manhattan Street Block (right) distances.*

There is even a measure of distance known as the *Canberra distance*! This measures the distance of data points from a central point of interest, analogous to how the suburbs are set up in Canberra. Not surprisingly the Canberra distance was developed by two workers at C.S.I.R.O. in the ACT.

7.1.2 Data normalization

In cases where the numbers of one variable are much larger than the other variables they can bias the analysis because they dominate the distance measurement. For example, in an analysis of humans we could have:

- Height
- Foot length
- Finger width
- Umbilical radius

In this case it is clear that the absolute variations in height will be much larger than the other variables, therefore they will dominate the analysis. To avoid this effect the data are *standardized* before the analysis, this means setting the mean of each variable to 0 and the standard deviation to 1. This is also known as taking the *zscore* and for a collection of values, *X*, we can standardize the data using the equation:

$$z = \frac{X - \bar{X}}{s}, \tag{7.1}$$

where $\bar{X}$ and *s* are the mean and standard deviation of the values in X, respectively. This is a simple operation to perform in R and as an example we'll standardize a sample of 100 random numbers.

Example code: 52

```
> x=runif(100) # 100 random numbers to act as data
> mean(x) # Find the mean (should be ~0.5)
> sd(x) # Find the S.D. (should be ~0.29)
> z=x-mean(x) # x minus the mean of x
> mean(z) # Find the mean of the adjusted values (0.0)
> z=z/sd(x) # divide by the S.D. of x
> sd(z) # Find the S.D. of the adjusted values (1.0)
```

Not surprisingly there is a function in R that will perform the calculation automatically. We can call the scale function, which will standardize the data (in the case of matrices it will standardize each column separately).

Example code: 53

```
> x = runif(100) # 100 random numbers to act as data
> z = scale(x,center=TRUE,scale=TRUE) # standardize the data in x
```

```
> mean(z) # Find the mean of the adjusted values (0.0)
> sd(z) # Find the S.D. of the adjusted values (1.0)
```

One last point is that some cluster analysis techniques work best with normally distributed data. This is not a strict assumption, but log-normal variables, for example, should be log transformed (Figure 7.4).

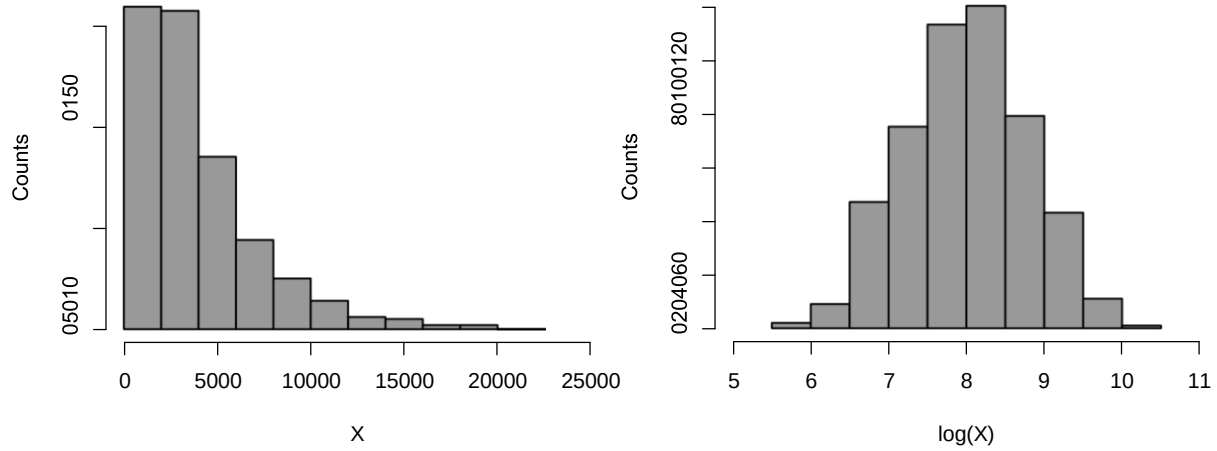


Figure 7.4: *A log-normal distribution (left) can be transformed into a normal distribution (right) simply by taking the logarithm of the values.*

7.2 Hierarchical clustering

One of the simplest forms of clustering is *hierarchical clustering*. As the name suggests we cluster points together to form a hierarchy with similar points being clustered early in the process and dissimilar points being clustered later. It is easiest to demonstrate the procedure graphically and we'll do it for 6 points (*A* through *F*) in two dimensional space. The positions of these points are shown in Figure 7.5.

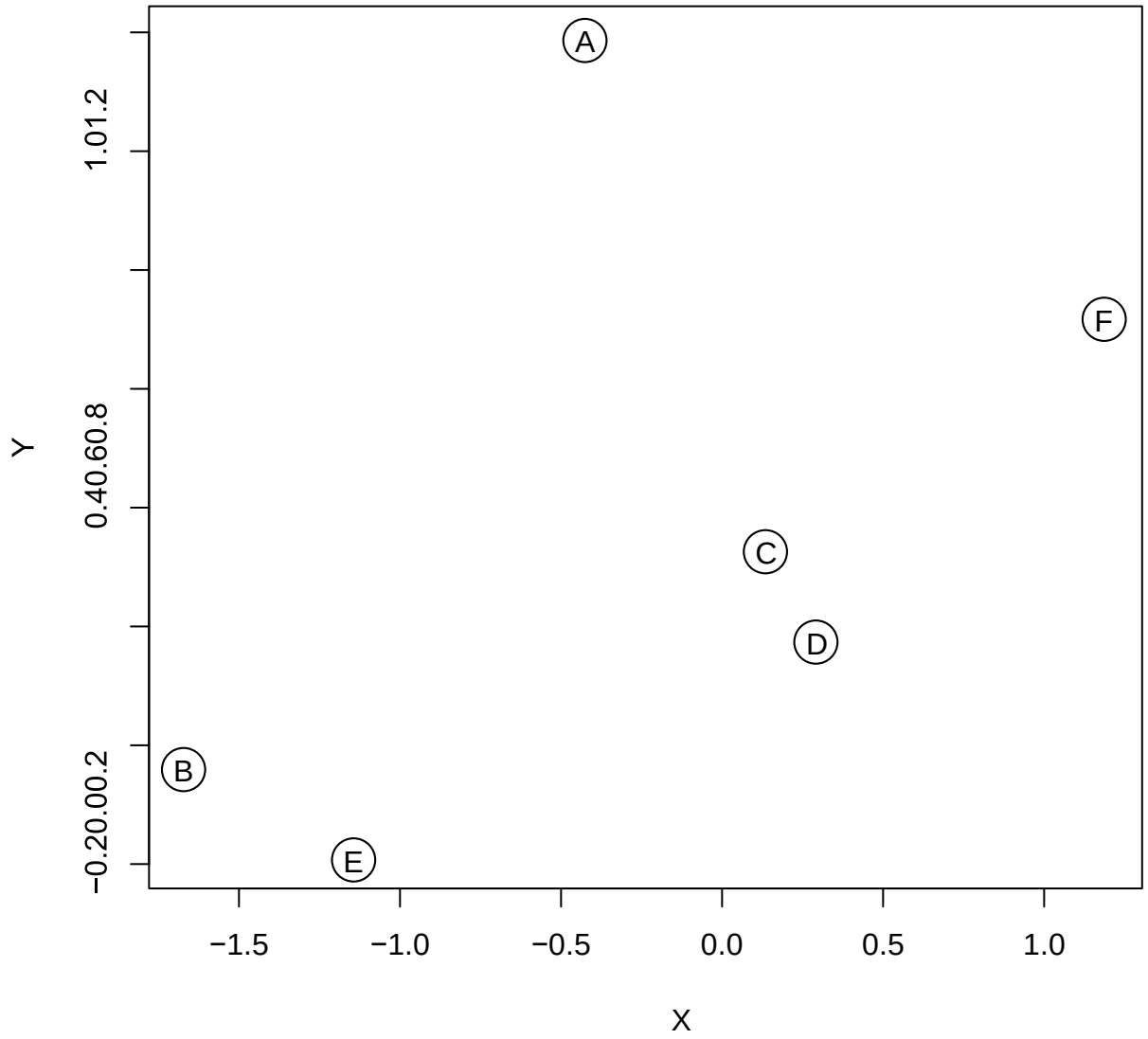


Figure 7.5: *In our hierarchical clustering example we start with 6 data points based on 2 measured parameters, X and Y. The individual points are labeled so that we can see how the clustering procedure advances.*

The first step of the hierarchical clustering procedure is to find the 2 points that are most similar to each other. Because we know that distance is a good measure of similarity, we can simply say that the two points closest to each other are the most similar. Examination of the data shows that the closest two points are *C* and *D* so we connect them with a line and create a new data point (shown by a black dot) at the center of the line (Figure 7.6). Simultaneously we'll track the development of the clusters in a so-called *dendrogram*, in which we draw a link between points *C* and *D* at a height corresponding to the distance separating them.

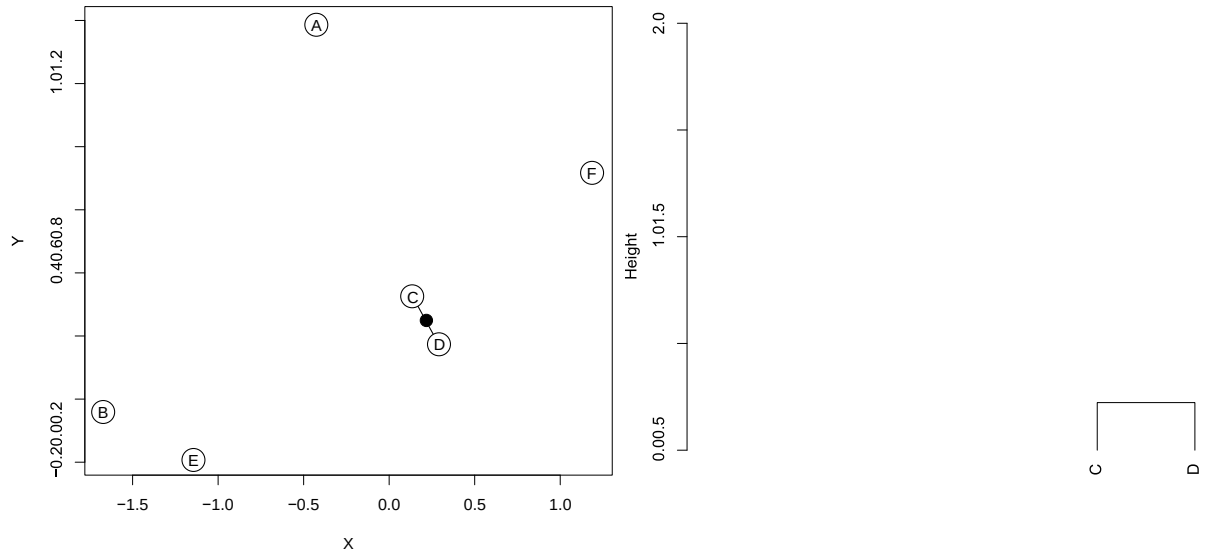


Figure 7.6: *The closest two points, C and D, are deemed to be the most similar and are connected. A new point is then created halfway along the link between the points. This connection can be represented in the dendrogram on the right, with the link between the points drawn at a height corresponding to the distance separating them.*

Because we have now connected *C* and *D* together, they are no longer considered separately in the analysis, but instead are represented by the single point midway along their link. Now we need to find the next most similar pair of points, which is *B* and *E*. Again we link the points and create a new point midway between them (Figure 7.7). We can also add this link to the dendrogram. Notice in the dendrogram that the link between *B* and *E* is higher than the link of *C* and *D* because the points are separated by a greater distance.

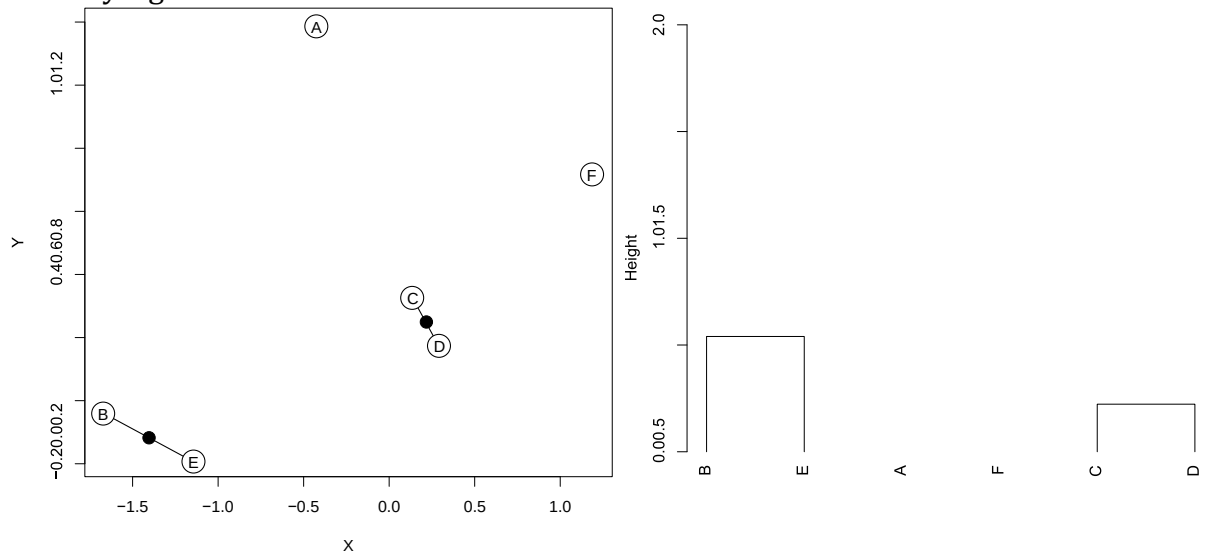


Figure 7.7: *Points B and E are connected and an additional link is added to the dendrogram.*

The next shortest connection is between *F* and the midpoint in the link between *C* and *D*. Again we connect these points and insert a new point halfway along the connection (Figure 7.8). In the dendrogram a connection is made between *F* and the center of the *C* and *D* connection.

You should now be able to see how the dendrogram records the sequence of connections that are being formed in the data set.

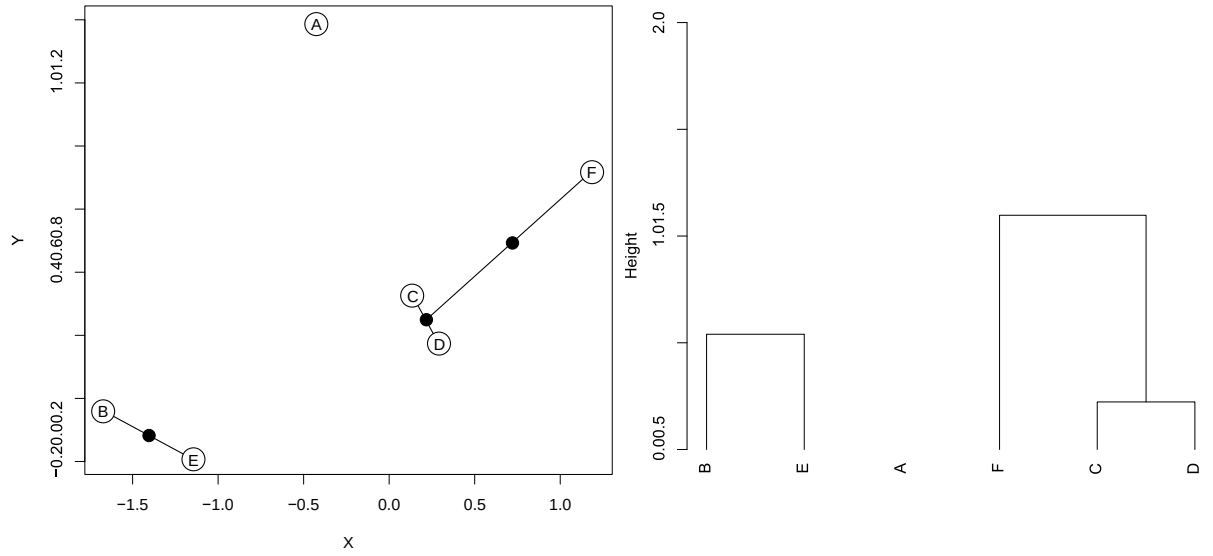


Figure 7.8: *The next connection links F with the point between C and D. This is represented in the dendrogram as a link connecting into the existing C to D link.*

The next shortest connection is point *A* to the middle of the connection between *F* and *C* to *D*. This link is also added to the dendrogram (Figure 7.9).

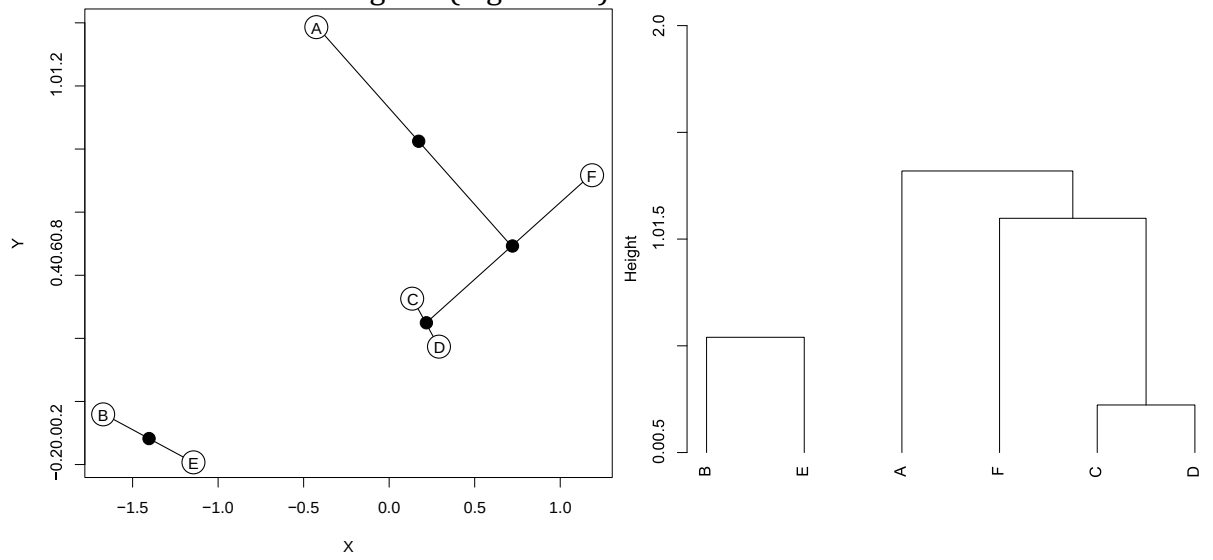


Figure 7.9: *Point A is now included in the hierarchy and a link is formed in the dendrogram that connects it to the existing cluster of points containing C, D and F.*

Finally, the center point of the *B* to *E* link is connected to the center point that was inserted when the previous link was made. This connects *B* and *E* to the other data points and the final link is placed in the dendrogram (Figure 7.10).

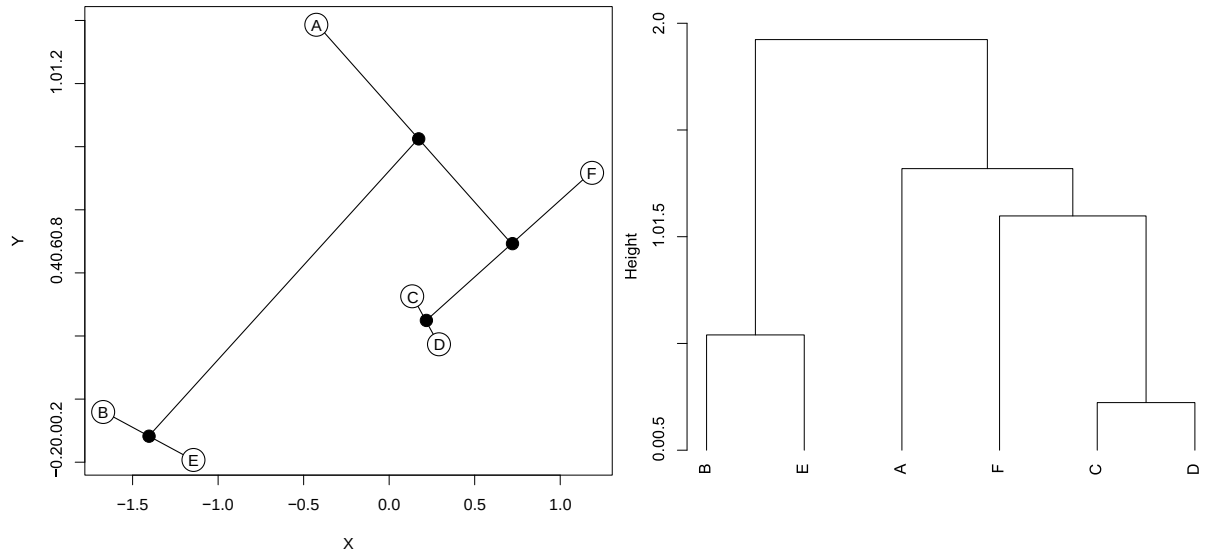


Figure 7.10: *The final connection completes the hierarchical clustering routine and the resulting link in the dendrogram shows how all 6 points are connected to each other, the length of the connections and the sequence in which the connections were made.*

To show how hierarchical clustering can be performed in R we'll use the same set of data points and calculate a dendrogram directly. As a first step we'll enter the data values by hand and plot them.

Example code: 54

```
> rm(list=ls()) # clear the memory
> dev.off() # close all existing graphics windows
> X=c(-0.432, -1.665, 0.125, 0.287, -1.146, 1.190) # input X values > Y=c(1.189, -0.037, 0.327, 0.174, -0.186, 0.725) # input Y values
> plot(X,Y,xlab='X',ylab='Y') #plot the data points
```

You should now have a collection of data points like the ones shown in Figure 7.5. When performing cluster analysis in R it is assumed that your data is in a matrix with each column representing a different variable (thus the rows are individual cases). We can combine the X and Y variables together using the cbind function and then assign each row a name (the letters A through to F) using the rownames function.

Example code: 55

```
> XY=cbind(X,Y) #combine the variables into a matrix (6 rows, 2 columns)
> rownames(XY)=c('A','B','C','D','E','F') #case names to identify points
```

Now that we've got the data into the correct format we can perform the hierarchical clustering. First we must calculate a distance matrix that contains all the Euclidean distances between the points (remember that distances are fundamental to cluster analysis). Using these distances the function hclust will calculate the clustering solution, which can then be plotted as a dendrogram.

Example code: 56

```
> D=dist(XY); # calculate distance matrix
> hc = hclust(D, 'average') # calculate the clustering solution
> plot(hc,hang=-1) # plot the final dendrogram
```

You should find that your dendrogram is the same as the one in Figure 7.10. In the example above it was easy for us to see how the points should be connected together and how the dendrogram evolved because it considered a two-dimensional problem. For data with a high number of dimensions the structure of the hierarchical clustering can be illustrated in a dendrogram and a data set can be interpreted in terms of which points cluster together.

7.2.1 An example: Fisher's irises

We looked at sepal measurements from Fisher's iris data set when we studied regression. Now we'll include length and width measurements for the petals as well as the sepals. This means that we have a 4 dimensional data set. Fisher knew that his data set contained two different species of iris, so we'll see if we can detect the different species by performing hierarchical clustering on the sepal and petal measurements. The data are stored in the file iris_cluster.Rdata that contains one variable called irises, which consists of 4 columns (1 for each measurement type) and 30 rows (1 for each measured flower).

Example code: 57

```
> rm(list=ls()) # clear the memory
> dev.off() # close all existing graphics windows >
load('iris_cluster.Rdata') # load the data file
> D=dist(irises) # calculate the Euclidean distance matrix
> hc=hclust(D, 'average') #calculate the clustering solution
> plot(hc,hang=-1) #plot the final dendrogram
```

The resulting dendrogram should look like the one in Figure 7.11. The structure of the dendrogram clearly shows two clusters of data that correspond to the two different species. The final link in the dendrogram connects the two clusters together and the height of the link shows that large differences exist between the two species on the basis of their petal and sepal measurements.

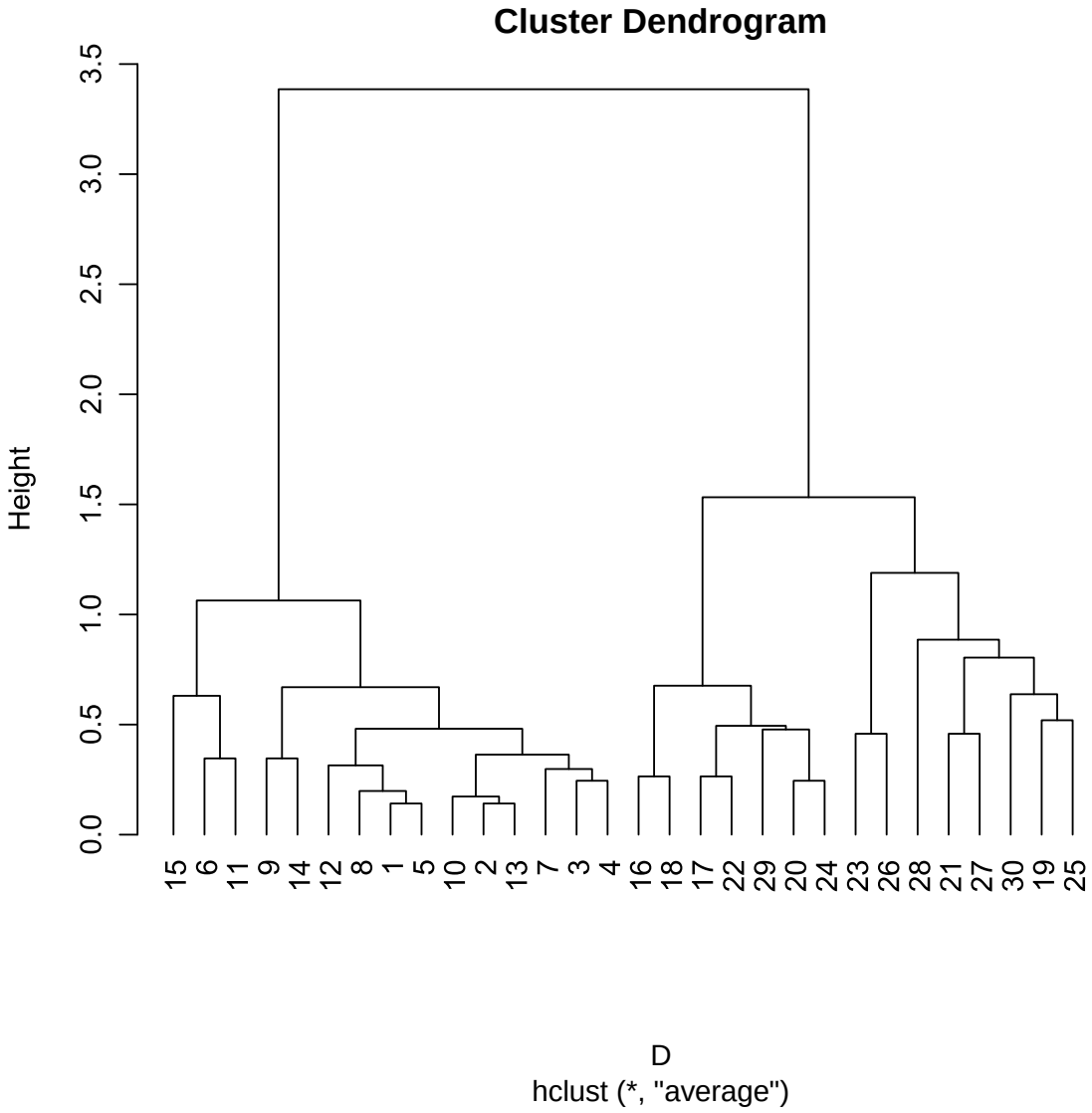


Figure 7.11: *Final dendrogram for 30 cases taken from Fisher's iris data set. Notice how the dendrogram reveals clearly the presence of two different species in the sample, which are only connected by the final link. The numbers along the bottom correspond to the case number in the data set.*

7.2.2 Disadvantages of dendrograms

Although dendrograms are simple to interpret they can become complicated when data sets with a large number of cases are considered. Fisher's original data set contained 100 cases and we'll run the cluster analysis again considering all the cases. This is the same as the previous exercise, but the data is stored in the file iris _cluster2.Rdata.

Example code: 58

```
> rm(list=ls()) # clear the memory
> dev.off() # close all existing graphics windows
> load('iris_cluster2.Rdata') # load the data file
```

```
> D=dist(irises) # calculate the Euclidean distance matrix > hc=hclust(D, 'average') # calculate the clustering solution
> plot(hc,hang=-1) # plot the final dendrogram
```

The final dendrogram is shown in Figure 7.12 and you can see that so many cases are now included that it has become difficult to interpret (if for no other reason than we can no longer see the case numbers along the bottom). When we looked at smaller data sets the resulting dendrograms could be easily interpreted. For larger data sets the overall structures are still obvious, for example two different species of iris, but the fine details require close examination.

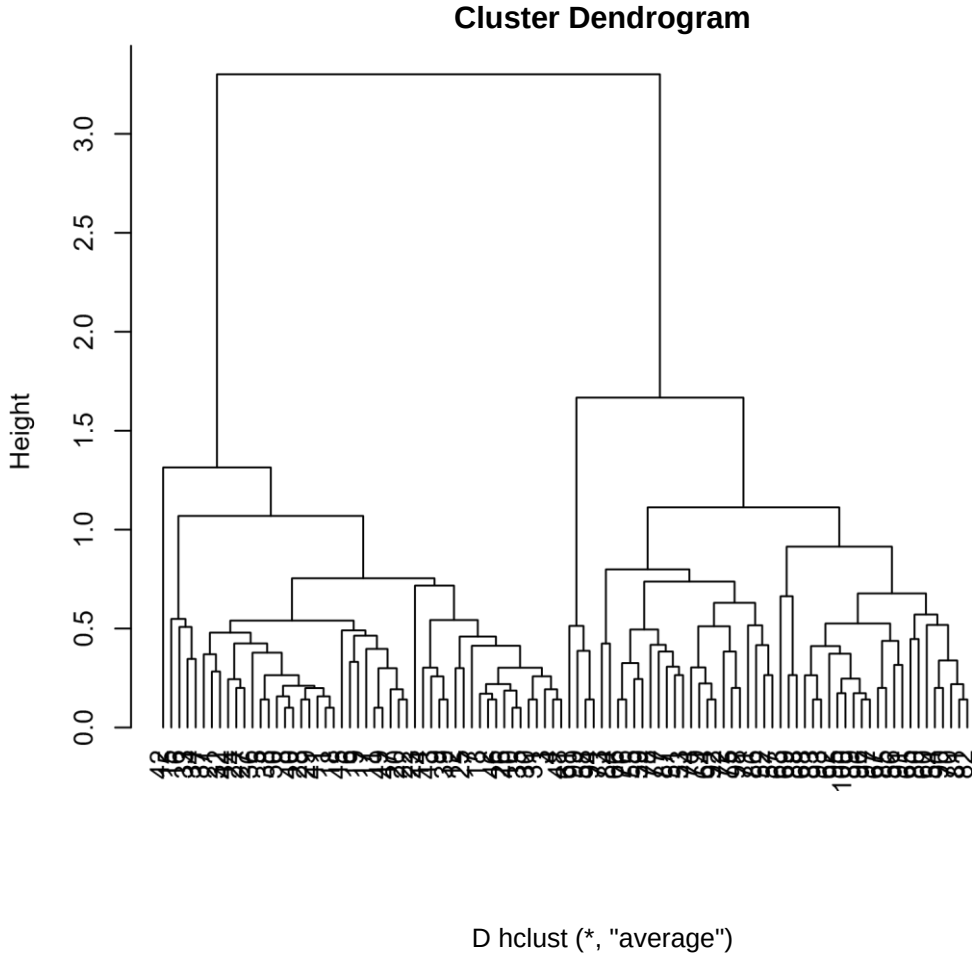


Figure 7.12: *Final dendrogram for the 100 cases that compose Fisher's complete iris data set. Notice how the fine details of the dendrogram become difficult to interpret when a large number of cases are included.*

7.3 Deterministic k-means clustering

K-means clustering is another way of finding groups within large multivariate data sets. It aims to find an arrangement of cluster centers and associate every case in the data set to one of these clusters. The best cluster solution minimizes the total distance between the data points and their assigned cluster center. The procedure for calculating a k-means clustering follows 4 basic steps:

1. Choose the number of clusters for the analysis.
2. Assume starting positions for the cluster centers.
3. Attribute every sample to its nearest cluster center.
4. Recalculate the positions of the cluster centers until a minimum total distance is obtained across all the cases.

To demonstrate the ideas behind k-means clustering we'll look at a graphical example that considers a simple two-dimensional data set. The data set clearly contains two different groups of points, those with high *X* and high *Y* values and those with low *X* and low *Y* values (Figure 7.13). In k-means clustering we aim to find cluster centers that mark the centers of the different groups. Each case is then assigned to the nearest cluster center and it is assumed that they have similar characteristics. Of course, the characteristics of a cluster are given by the location of the center in the parameter space. Finally, because each case can only belong to one cluster this is a so-called *hard* clustering technique.

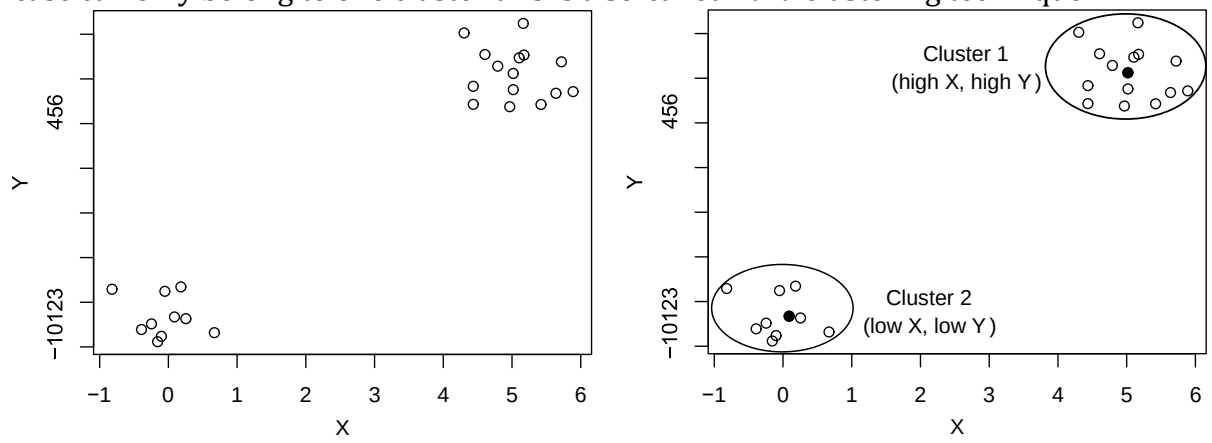


Figure 7.13: *A simple example of k-means clustering. The cases on the left show that there are two clear groups in the data. K-means involves defining the centers of these groups (marked by filled circles) and associating the individual cases to them to form clusters. The positions of the cluster centers in the parameter space define the characteristics of that cluster.*

7.3.1 Visualizing cluster solutions

Because the cluster centers have locations with the same number of dimensions as the original data it can be difficult to visualize the results of a cluster solution (we'll return to this problem in more detail in Chapter 8). One option to visualize the distribution of cases and the locations of the cluster centers is to simply choose two variables from the input data set and plot the cluster solution using those variables (although the calculation of the solution still involves all of the variables).

Therefore to obtain a clear plot of the clusters in two-dimensions we must choose 2 variables which show a good separation between the cluster centers. To look at this problem we'll return to Fisher's 100 irises and perform a k-means clustering of the 4 petal and sepal measurements, splitting the data into 2 clusters (Figure 7.14).

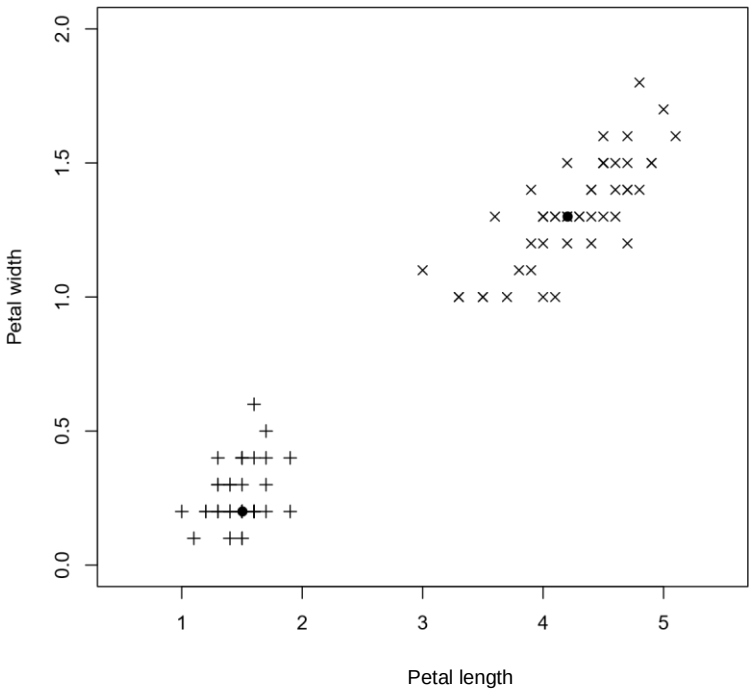


Figure 7.14: *K-means clustering of Fisher's iris data set. When plotted using the petal length and petal width variables, a 2 cluster solution can be visualized with a clear separation between the cluster centers (filled circles). Each case is assigned to its nearest cluster center. Cluster 1 (plus signs) is characterized by a center with small petal widths and small petal lengths. Cluster 2 (multiplication signs) is characterized by a center with large petal widths and large petal lengths.*

If we repeat the exercise but now calculate a more complex 3 cluster solution we can see that it becomes harder to visualize the results with just 2 parameters (Figure 7.15). There is a strong overlap between two of the clusters, which may indicate that there are too many clusters in the model (*i.e.*, the data really only consists of 2 groups) or that we haven't chosen the best parameters to display the results.

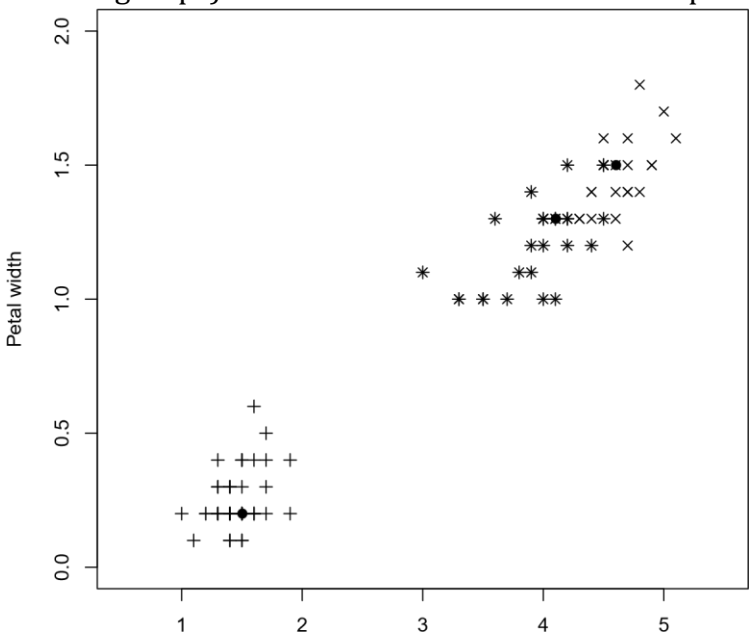


Figure 7.15: *K-means clustering of Fisher's iris data set. When plotted using the petal length and petal width variables a 3 cluster solution shows a clear overlap between two of the clusters (different clusters are shown with stars, plus and multiplication signs). There is a poor separation between two of the clusters in the top right of the plot.*

7.3.2 What input data should you use?

The properties of a cluster center are determined from its position within the data space. Therefore if we are to understand the results of the cluster analysis we must have a clear understanding of each of the input variables. This means that you shouldn't include variables in the analysis that you don't understand because you won't be able to interpret them in the final solution.

It is also important not to over-represent any one process or property of the data set. If I have 9 input variables which represent sediment transport mechanisms and only 1 which represents source area then my analysis will be biased towards a separation based only on transport. You need to think carefully about your input data and what it represents, don't include parameters in the analysis without a clear reason to do so. It is very tempting to include all of your measured data into an analysis, but ultimately this can make the solution very difficult to interpret, so instead you should try to form a well designed input.

7.3.3 How many clusters should be included in a model

Selecting how many clusters to include in your data analysis can be a challenge. If you include too few clusters the true relationships in the data may remain hidden, whilst if you choose too many the solution will be overly complex and you won't be able to interpret it. Generally, you should try and keep the number of clusters low and make sure there is a physically realistic interpretation that explains what process each cluster represents.

7.3.3.1 Silhouette plots

A number of statistical tools exist that you can use to decide how many clusters to include in your analysis. We'll look at one specific approach called *silhouettes*. The silhouette value for each data point is a measure of how similar that point is to the other points in its own cluster compared to the points in other clusters. Silhouette values range from -1 to +1, where:

- +1 : points are very distant from their neighboring clusters.
- 0 : points are not distinctly in one cluster or another. • -1: points are probably

  assigned to the wrong cluster.

Therefore a good cluster solution will be one which produces silhouette values close to +1. The values for all the cases in a given solution can be displayed in a *silhouette plot*. Let's look at an example of some two-dimensional data that clearly contains 4 clusters (Figure 7.16).

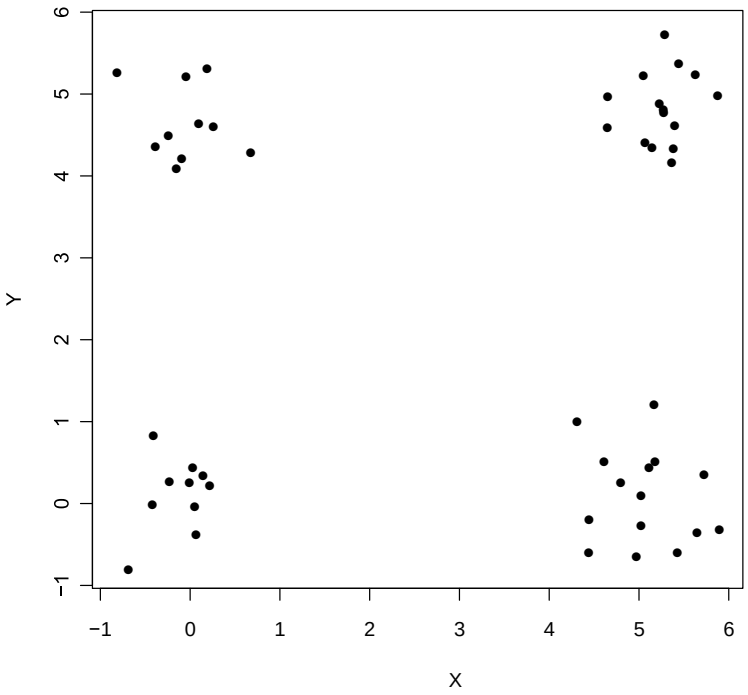


Figure 7.16: *Example two-dimensional data set that clearly contains 4 clusters.*

First we'll calculate a 2 cluster solution, which is clearly inadequate to explain the data given that we know it contains 4 clusters. The data is split into two clusters and the silhouette values are generally around 0.5 (Figure 7.17).

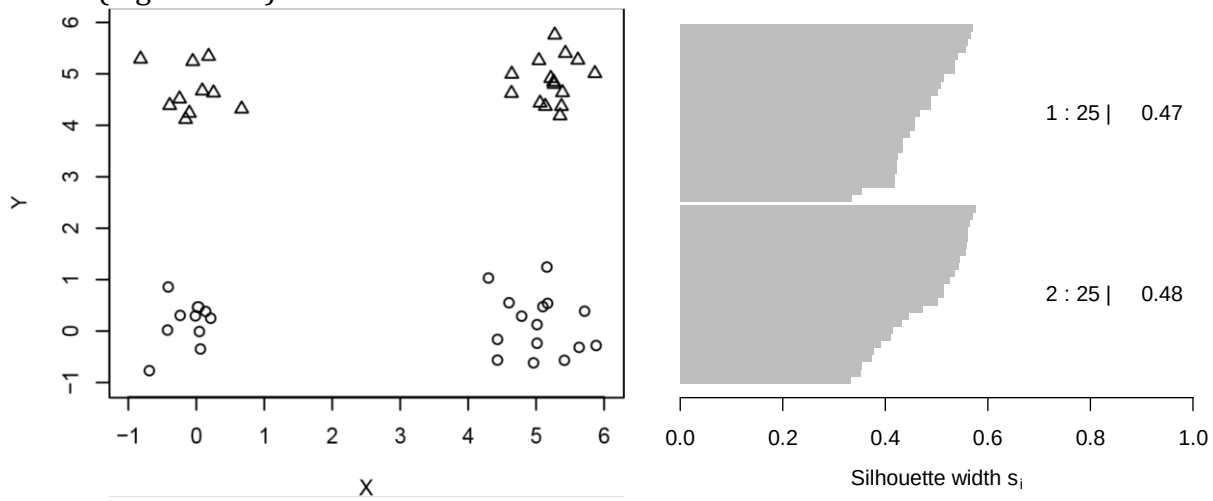


Figure 7.17: *The assignments of the individual cases to the final clusters are indicated by the different symbols (left). The silhouette plot on the right gives a histogram of the case silhouette values for each cluster (gray bars). The numbers to the right of the plot show how many cases are in each cluster and what their average silhouette value is. For example there are 25 cases in cluster 1 and their average silhouette value is 0.47.*

Now we can try a more complex 3 cluster solution and again examine the silhouette plot (Figure 7.18). We can see that the silhouette values for clusters 2 (crosses) and 3 (triangles) are close to 1, which

suggests that points have been assigned to them correctly. Cluster 1 (circles), however, is still spanning two sets of data points, so its silhouette values are reduced.

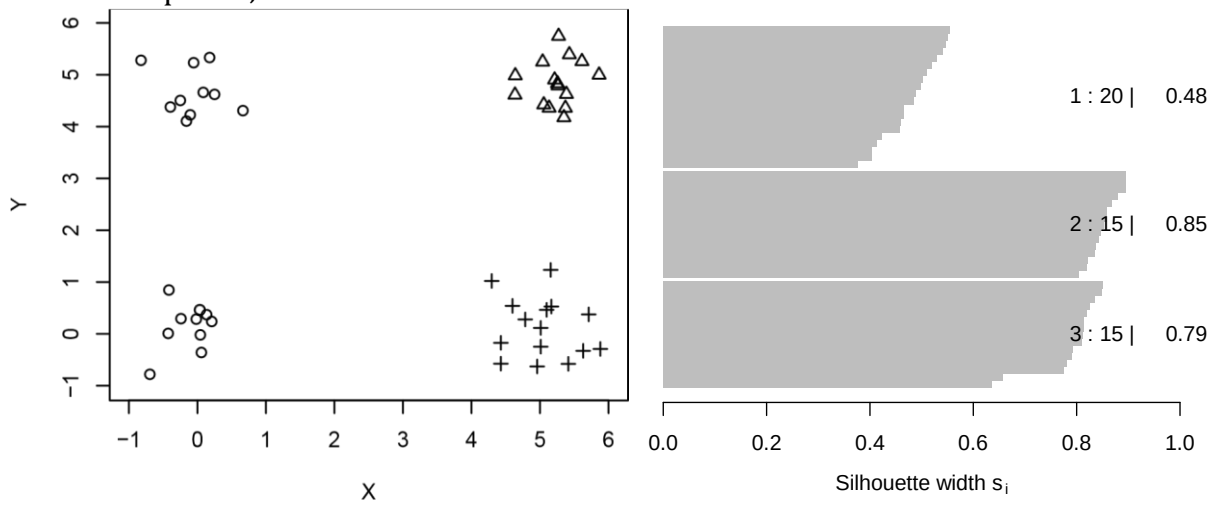


Figure 7.18: *The assignments of the individual cases to the final clusters are indicated by the different symbols (left). The silhouette plot on the right gives a histogram of the case silhouette values for each cluster (gray bars). The numbers to the right of the plot show how many cases are in each cluster and what their average silhouette value is. In this case the values for cluster 1 are still low because it is spanning two of the clusters in the data.*

A 4 cluster solution clearly (and not surprisingly) does a good job of partitioning the data into its correct groups (Figure 7.19). None of the cases are assigned wrongly so all the clusters have high silhouette values.

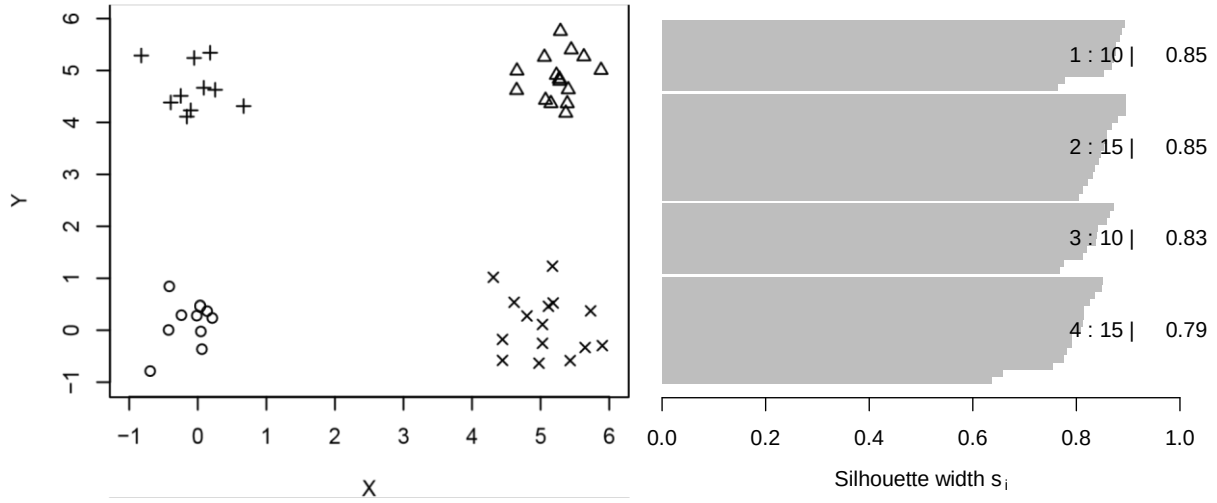


Figure 7.19: *The assignments of the individual cases to the final clusters are indicated by the different symbols (left). The silhouette plot on the right gives a histogram of the case silhouette values for each cluster (gray bars). The numbers to the right of the plot show how many cases are in each cluster and what their average silhouette value is. In this case the values for all the clusters are high because the data have been grouped correctly.*

So far we have only considered the cases where not enough, or just enough, clusters were included. But what happens when we calculate a solution that contains too many clusters? We'll work with the same data set, but this time calculate a 5 cluster solution (Figure 7.20). We can see that in order to accommodate the extra cluster, one of the clusters has been split into two parts (bottom right). This means that some of the points in clusters 4 and 5 have low silhouette values because they are not distinctly in one cluster or another or may have been assigned to the wrong cluster.

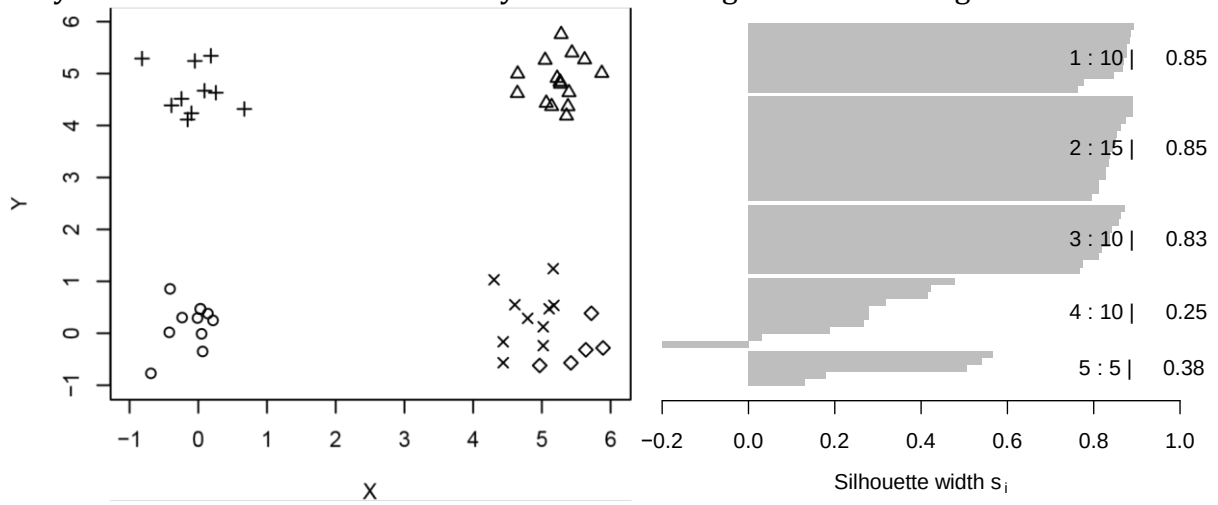


Figure 7.20: *The assignments of the individual cases to the final clusters are indicated by the different symbols (left). The silhouette plot on the right gives a histogram of the case silhouette values for each cluster (gray bars). The numbers to the right of the plot show how many cases are in each cluster and what their average silhouette value is. In this case one group in the data has been split into two parts to accommodate an additional cluster. This leads to an ambiguity in the assignment of the cases that is reflected in the silhouette plot.*

A 6 cluster solution is clearly too complex, with 2 of the data groups (top left and bottom right) being split in order to accommodate the additional clusters (Figure 7.21). The low values in the silhouette plot show the high uncertainties associated with the case assignments when the model is overly complex.

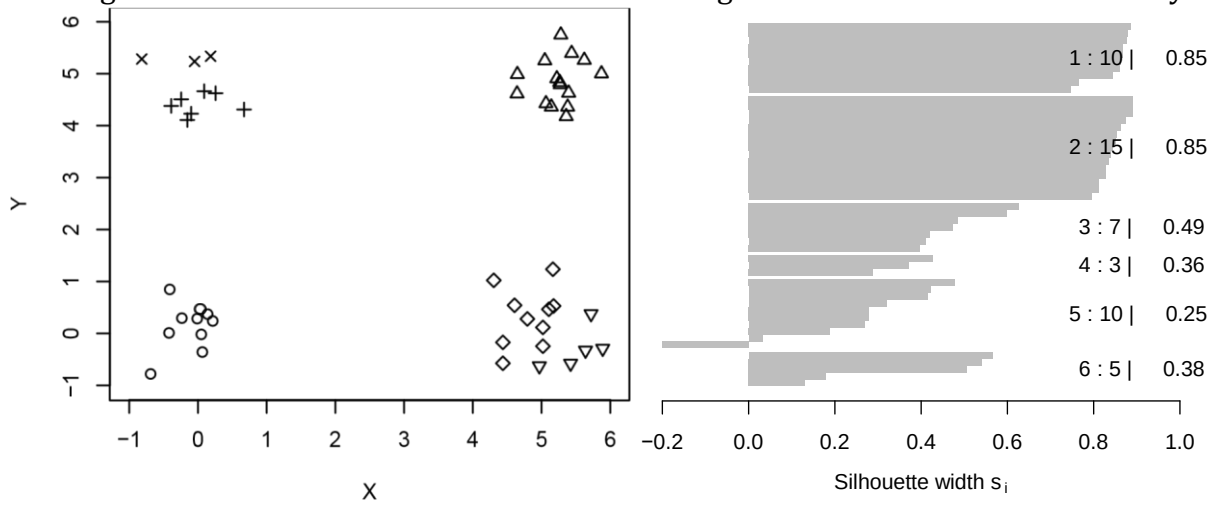


Figure 7.21: *The assignments of the individual cases to the final clusters are indicated by the different symbols (left). The silhouette plot on the right gives a histogram of the case silhouette values for each cluster*

(gray bars). The numbers to the right of the plot show how many cases are in each cluster and what their average silhouette value is. In this case two groups in the data have been split into parts to accommodate the additional clusters. This leads to an ambiguity in the assignment of the cases that is reflected in the silhouette plot.

As a general approach to model selection we can calculate the average silhouette value across all cases for a given number of clusters. If we then compare the different values for different models we can choose the number of clusters which returns the highest average silhouette value (Figure 7.22).

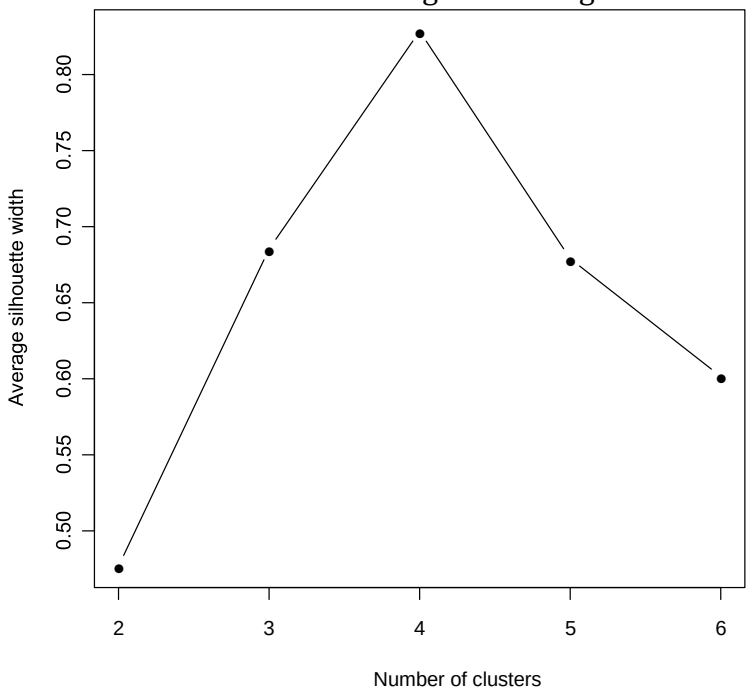


Figure 7.22: *For our example data set the average silhouette value reaches a maximum when 4 clusters are included in the analysis. This suggests we should select a 4 cluster model to represent the data. Of course we set the data up to have 4 clusters so this result is not surprising, but it demonstrates the approach.*

7.4 Now it's your turn: Portuguese rocks

We'll look at an example data set composed of the oxide concentrations of 134 Portuguese rocks. The collection of examined rocks is known to include granites, diorites, marbles, slates, limestones and breccias. The measured oxides include (all expressed as %); SiO_2, Al_2O_3, Fe_2O_3, MnO, CaO, MgO, Na_2O, K_2O and TiO_2. We'll perform a k-means clustering of the data set and see what kind of groups appear (and importantly if they make sense geologically). First we'll load the data, which is stored in the file rocks cluster.Rdata and we'll standardize the columns of the data matrix (called input) using the scale function.

Example code: 59

```
> rm(list=ls()) # clear the memory
```

```
> dev.off() # close all the figure windows
> load('rocks_cluster.Rdata') #load data set
> z=scale(input,center=TRUE,scale=TRUE) # zscore the data
```

If you haven't used the cluster package in R before, you will need to install it and make it ready for use. This is simple:

Example code: 60

```
> install.packages('cluster') #download the cluster package
> library(cluster) # prepare the package for use
```

As an example we'll calculate a 3 cluster solution and plot the silhouette plot (which also gives the mean silhouette value). There is also a specially written function called classify rocks which will print to the screen how many of each rock type are included in each of the clusters.

Example code: 61

```
> fit=pam(z,3) # k-means solution with 3 clusters
> plot(fit) # plot the results (click to advance through the plots)
> source('classify_rocks.R') # display which rocks are in the clusters
--------------
Cluster 1 contains:
Number of granites = 30
Number of diorites = 5
Number of marbles = 0
Number of slates = 0
Number of limestones = 0
Number of breccias = 0
--------------
Cluster 2 contains:
Number of granites = 2
Number of diorites = 5
Number of marbles = 0
Number of slates = 7
Number of limestones = 0
Number of breccias = 0
--------------
Cluster 3 contains:
Number of granites = 0
Number of diorites = 0
Number of marbles = 51
Number of slates = 0
```

```
Number of limestones = 28
Number of breccias = 6
```

Repeat the process with different numbers of clusters and write down the mean silhouette value for each one. As with the example above you can use these values to select which model seems to represent the data the best. You can obtain the mean silhouette value for a given cluster model, for example fit, with the command:

Example code: 62

```
> fit$silinfo$avg.width
```

The results of the analysis are shown on the next page.

The results of the silhouette analysis are shown in Figure 7.23 and suggest that a two cluster model should be selected.

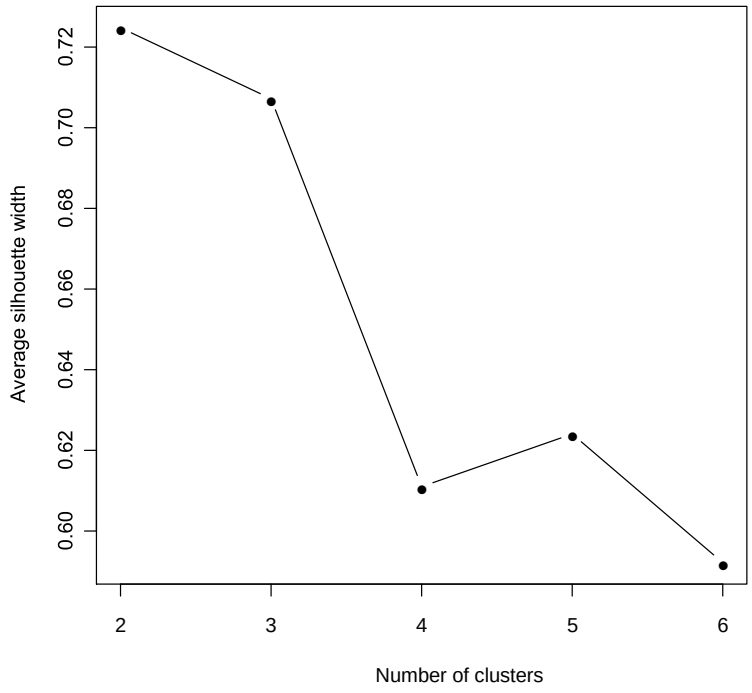


Figure 7.23: *Plot of the average silhouette value as a function of the number of clusters for the Portuguese rocks data set. The silhouette values suggest we should select a 2 cluster model to represent the data.*

If we perform the analysis for two clusters we can look at how the rocks are assigned to the different cluster centers and the properties of the cluster centers. We'll start by producing the two cluster model and looking at the cluster assignments.

Example code: 63

```
> fit=pam(z,2) # k-means solution with 2 clusters
> plot(fit) # plot the results (click to advance through the plots)
```

```
> source('classify_rocks.R') # display which rocks are in the clusters
--------------
Cluster 1 contains:
Number of granites = 32
Number of diorites = 10
Number of marbles = 0
Number of slates = 7
Number of limestones = 0
Number of breccias = 0
--------------
Cluster 2 contains:
Number of granites = 0
Number of diorites = 0
Number of marbles = 51
Number of slates = 0
Number of limestones = 28
Number of breccias = 6
```

The assignment of the rocks into the two clusters appears to make geological sense with Si-rich rocks in Cluster 1 and Ca-rich rocks in Cluster 2. The cluster centers are stored in the fit variable and we will need to examine them to understand the properties of the cluster centers. First we can simply display the locations of the cluster centers on the screen.

Example code: 64

```
> fit$medoids

                SiO2          Al2O3          Fe2O3          MnO           CaO           MgO
[1,] 1.4478011 1.0980917 0.6567171 0.9195634 -1.3148995 -0.2510658
[2,] -0.7766486 -0.7490279 -0.5847489 -0.5722223 0.7888373 -0.2183949
                Na2O          K2O          TiO2
[1,]  1.2156960  1.3678200  0.4784906  [2,]  -
0.7085785 -0.6963739 -0.4365755
```

In their current form the coordinates of the centers still correspond to the standardized parameters, so they are difficult to interpret. To transform the variables back to the original measurement space we must perform the inverse of the standardization procedure. This is done by multiplication with the original column standard deviations of input and then adding on the column means of input. As an example we'll perform this operation on the second cluster center, which is stored in the second row of fit$medoids.

Example code: 65

```
> CC=fit$medoids[2,] # extract the cluster center data
> CC=CC*apply(input,2,sd) # multiply by the column SD of input
```

```
> CC=CC+colMeans(input) # add the column means of input
> CC #display the cluster center on screen
```

For the center of the second cluster we obtain oxide percentages of:

SiO_2[%]	Al_2O_3[%]	Fe_2O_3[%]	MnO[%]	CaO[%]	MgO[%]	Na_2O[%]	K_2O[%]	TiO_2[%]
0.5	0.4	0.1	0	54.4	0.6	0.07	0.13	0

Given that cluster 2 contains limestones and marbles the composition above is not surprising. Repeat the calculation to find the composition of the first cluster center and check that it matches with what you would expect from the rocks included in the cluster.

Simplicity, simplicity, simplicity! I say, let your affairs be as two or three, and not a hundred or a thousand. Simplify, simplify.

Henry David Thoreau

8 Dimension reduction techniques

In the same way that we can draw a projection of a three-dimensional (3D) cube on a twodimensional (2D) surface, for multidimensional data sets we can make plots that project a large number of dimensions into 2D. The limitation of this approach, however, is that the plots can become difficult to interpret (we discussed this in Section 6.1).

In Section 6.2 we plotted a 3D data set in `R` by projecting it into 2D. If you look back at Figure 6.5 you'll see that the projection makes it difficult to interpret the data and you can imagine that as the number of data dimensions increases, this problem gets worse. We're going to look at a number of different techniques that are aimed at reducing the number of dimensions of a data set in order that the data structure can be interpreted more easily. Generally these approaches are called *dimension reduction* techniques and come in a variety of different forms. We'll focus mainly on a technique called *principal component analysis* and then briefly look at some alternative methods.

8.1 Principal component analysis

Look at the two-dimensional data set in Figure 8.1, how many dimensions do we need to describe its variability?

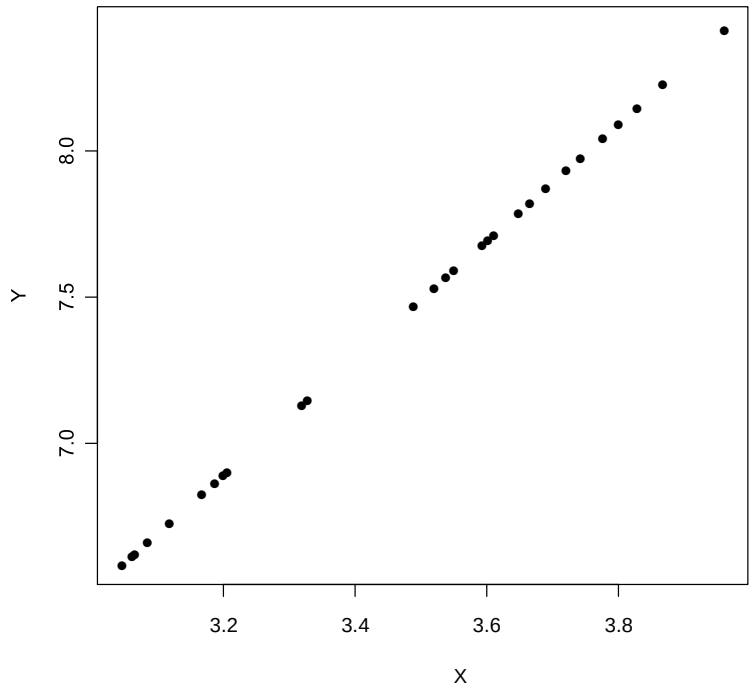


Figure 8.1: *An example 2D data set. How many dimensions do we need to describe the variability of the data fully?*

Because the data points all fall exactly on a straight-line we can describe their variability fully with a single dimension (a straight-line passing through the points). This means that we are able to take the 2D data and by exploiting its structure, specifically the perfect correlation between X and Y, reduce the number of dimensions needed to represent it to 1. This is shown in Figure 8.2 where the data points are plotted on a 1 dimensional line.

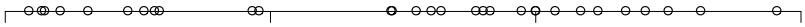


Figure 8.2: *Because the data points in Figure 8.1 fall on a line, we can rotate our coordinate system so that all of the points lie along a single dimension. Therefore their full variability can be explained in 1D.*

Now consider the two-dimensional data set in Figure 8.3, how many dimensions do we need to describe its variability fully?

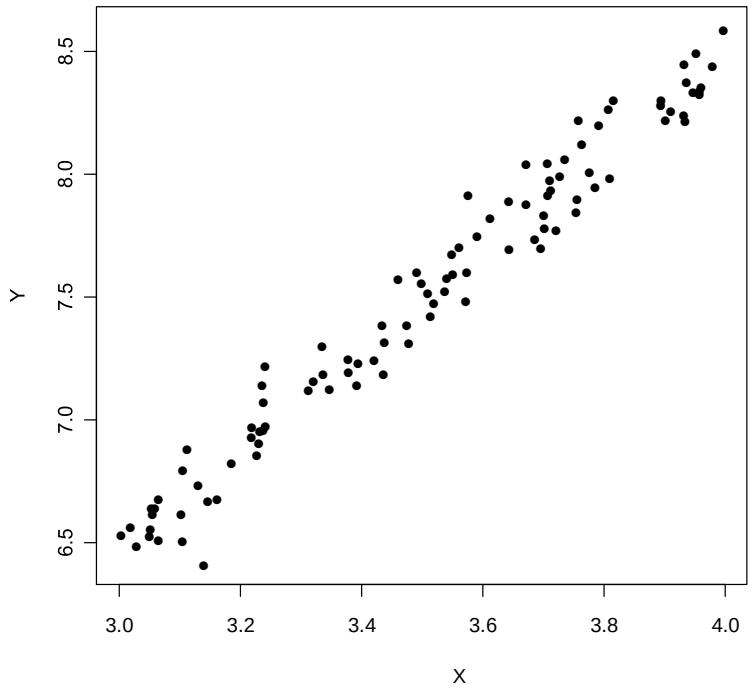


Figure 8.3: *Another example of a 2D data set, where the points do not all fall perfectly onto a straight-line. How many dimensions do we need to describe the variability of the data fully?*

Because the points do not lie perfectly on a straight-line we will still need two dimensions to describe the variability fully. However, we could make a compromise and say that the deviations from a straight-line are very minor so we could still represent the vast majority of the data variability using a single dimension passing along the main path of the data. By doing this we'll lose some information about the data (specifically their deviations from the line) but maybe that is an acceptable loss given that we can reduce the number of dimensions required to represent most of the data variability.

Often in data sets with many variables, a number of the variables will show the same variation because they are controlled by the same process. Therefore in multivariate data sets we often have data *redundancy*. We can take advantage of this redundancy by replacing groups of correlated variables with new uncorrelated variables (the so-called *principal components*). Principal component analysis (PCA) generates a new set of variables based on a linear combination of the original parameters. All the principal components are orthogonal (at right-angles) to each other so there is no redundant information because no correlation exists between them. There are as many principal components as original variables, however, because of data redundancy it is common for the first few principal components to account for a large proportion of the total variance of the original data.

Principal components are formed by rotating the data axes and shifting the origin to the point corresponding to the multivariate mean of the data. Let's look at an example similar to the one above. We'll take a 2D data set and see how much of the data variability is explained by the principal components (Figure 8.4).

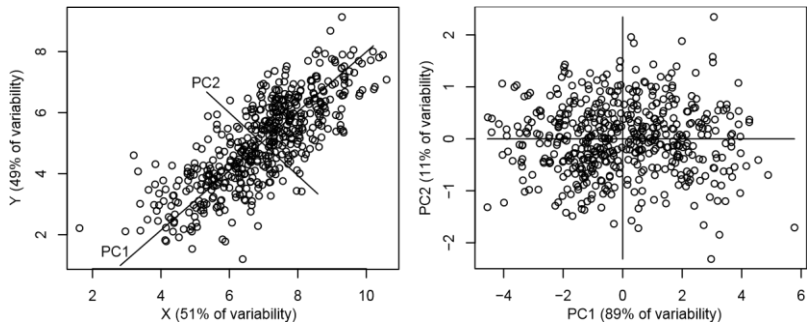


Figure 8.4: *In the original data set (left) each dimension explains approximately 50% of the total variability. Principal components (marked as PC1 and PC2) can be fitted to the data. The data can then be plotted with the principal components defining a new coordinate system (right) where 89% of the total variability can be explained using a single dimension.*

Now we'll consider a 3D case. In Figure 8.5 the three-dimensional data can be represented very well using only the first 2 principal components. This simplifies the data into a two-dimensional system and allows it to be plotted more easily. The first 2 principal components account for 99% of the total data variability, whilst the third component accounts for the remaining 1%.

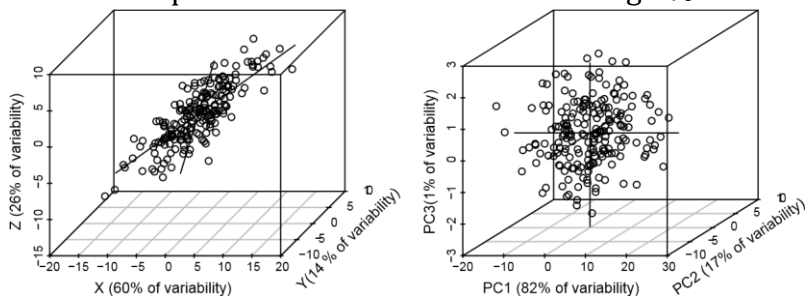


Figure 8.5: *The original data set is shown on the left and the variability explained by each dimension is shown in the axis labels. The principal components can be fitted to the data (black lines). The data can then be plotted with the principal components defining a new coordinate system (right).*

Given that the third principal component describes such a small amount of the total data variability we can consider dropping it from the plot. Therefore if we only plot the first 2 principal components, as in Figure 8.6, we lose 1% of the data variability from the representation, but maybe this is worth it when we can represent the data in a 2D rather than 3D plot.

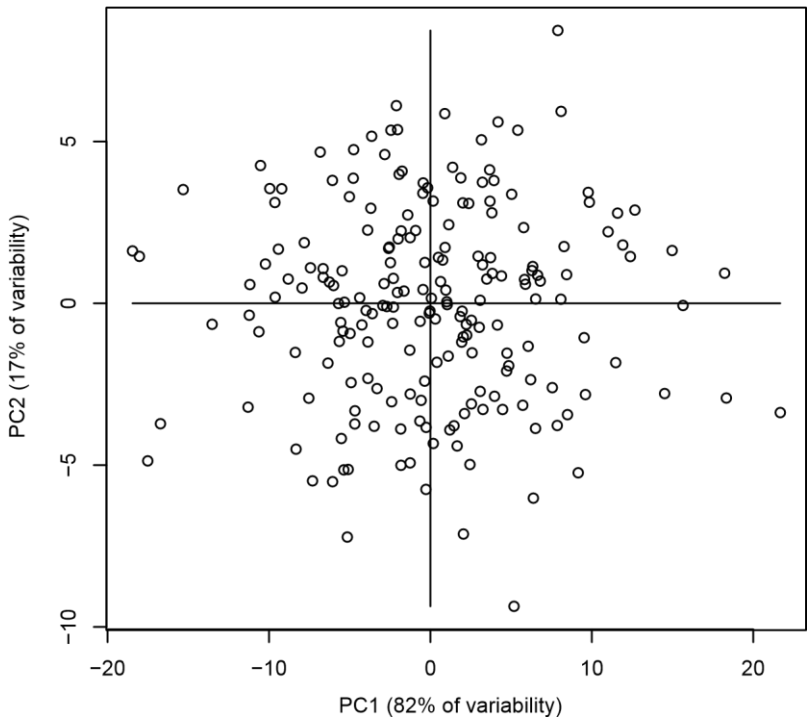


Figure 8.6: *By only plotting the first 2 principal components we lose a small amount of information (1% in this case), but the data can be plotted in two dimensions and becomes easier to interpret.*

We're not going to dwell on how the principal components are calculated, but instead we'll focus on their application. To give you a hint, the principal components are obtained from socalled eigenvector analysis, which involves the calculation of eigenvalues and eigenvectors. The eigenvectors describe the directions of the principal components through the data, whilst the eigenvalues describe the length of the components (the longer the component, the more of the data variability it describes). Information on how a data set is related to it's principal components are given by the *scores* and *loadings*. The scores give the coordinates of the data points in the principal component space and the loadings show how the orientation of the principal components is related to the original coordinate system. If these concepts aren't too clear to you at the moment, some examples should help.

8.1.1 Data normalization

In Section 7.1.2 we discussed why data normalization is important if the variables to be analyzed have different scales. Normalization is also important in PCA, otherwise it is possible that a small number of parameters could dominate the analysis. As before, we'll be using the scale function to standardize the data before we perform the PCA.

8.1.2 Building blocks

Before we look at geological applications, we'll consider an artificial example that will hopefully give you an appreciation of how we can use PCA. We'll form an artificial data set that contains data redundancy, *i.e.*, a number of the parameters carry correlated information, then we'll apply PCA. Our data set is based on 25 random blocks, like those shown in Figure 8.7. Each block is defined by the lengths of its long axis

(X_1), intermediate axis (X_2) and the short axis (X_3). Of course as the names suggest, the lengths must obey the relationship: $X_1 > X_2 > X_3$. We'll now calculate some more properties of the blocks based on the length of the axes.

- X_1 = Longest axis length.
- X_2 = Intermediate axis length.
- X_3 = Shortest axis length.
- X_4 = Longest diagonal length ($\sqrt{X_1^2 + X_2^2 + X_3^2}$).
- X_5 = The ratio $(X_1 + X_2)/X_3$.
- X_6 = Ratio of the block's surface area to its volume.

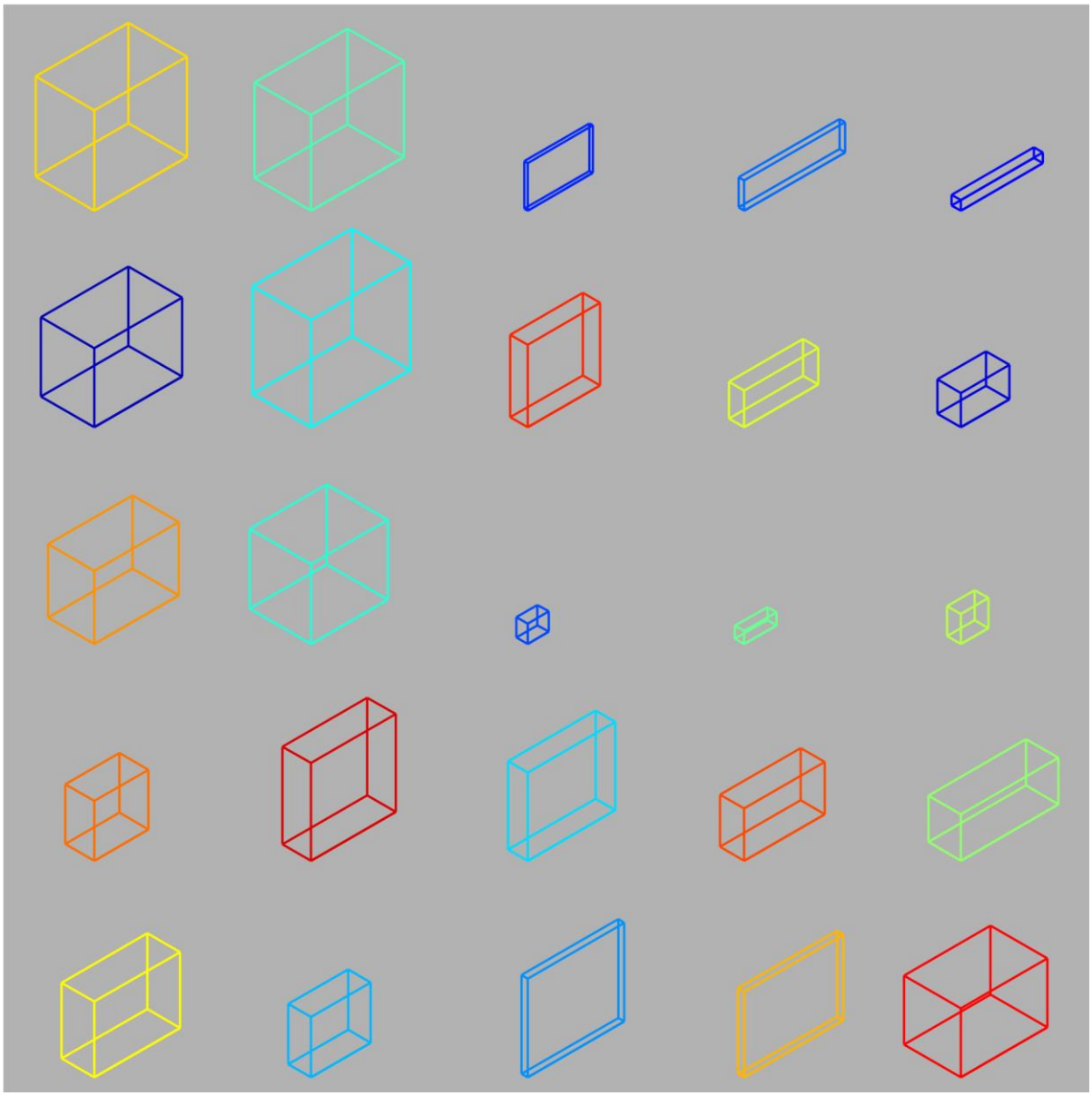


Figure 8.7: *25 blocks with randomly generated dimensions.*

The first thing we can do to investigate the structure of the blocks data set is to look at how the different parameters (X_1 through X_6) are correlated to each other. Given that we know how the data set was constructed we should have a general feeling of what correlations should exist. We can calculate a so-called *correlation matrix*, which consists of the correlation of each parameter to all the other parameters and plot it in an form which is easy to interpret (Figure 8.8).

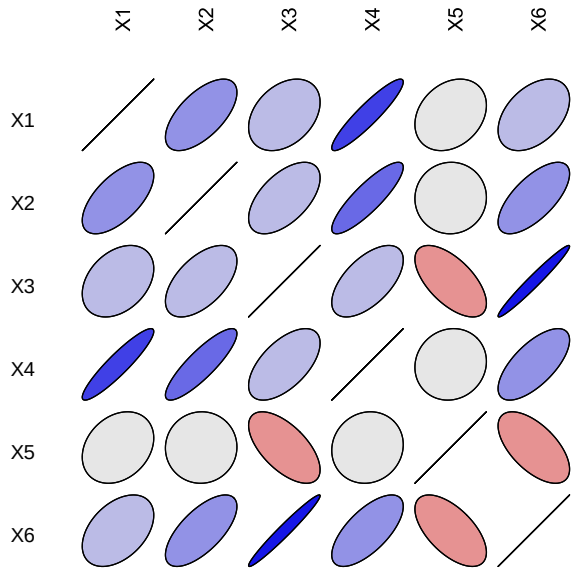


Figure 8.8: *Graphical representation of the blocks correlation matrix. The colour and orientation of the ellipses indicates either a positive (blue) or negative (red) correlation. Additionally, a darker shade of the colour indicates a stronger correlation. The strength of the correlation is also indicated by the form of the ellipse, ranging from circles (no correlation) to straight-lines (perfect correlation).*

The correlation matrix shows the kind of relationships we would expect. For example a strong positive correlation exists between X_1 and X_4, which is not surprising because as the longest side of the block increases the length of the diagonal should also increase. We also see negative relationships, for example between X_3 and X_5, again this is not surprising because as the length of the shortest side increases, X_5 will decrease because the denominator in the ratio becomes larger. Because some of the parameters in the blocks data set are correlated to each other it means that we have data redundancy and PCA should provide us with a good representation of the data variability. We can test this supposition be calculating the so-called *scree plot*, which shows how much of the data variability each principal component represents (Figure 8.9).

pc

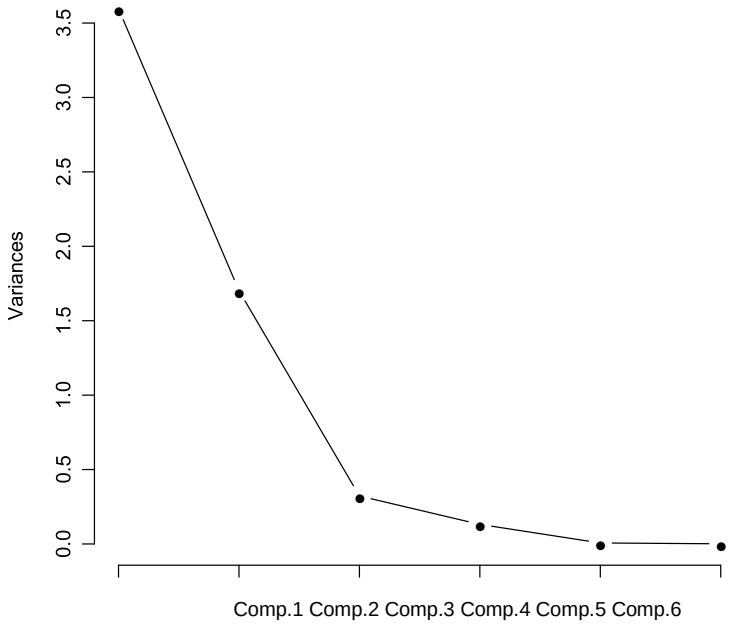


Figure 8.9: *A scree plot of the PCA of the blocks data set. The scree plot shows how much variance each principal component explains.*

Remember, the aim of PCA is to provide a good representation of the data variability in a small number of dimensions. Therefore it is important to check the scree plot and see what proportion of the variance is explained by the first 2 or 3 principal components. In the case of the blocks data, the first two principal components explain >90% of the data variability. Therefore we know that we are not losing too much information by only considering the first 2 principal components. We can now plot the blocks according to their coordinates with respect to the principal components. This information is given by the *scores* and we'll simply plot each block according to its scores with respect to the first and second principal components (Figure 8.10).

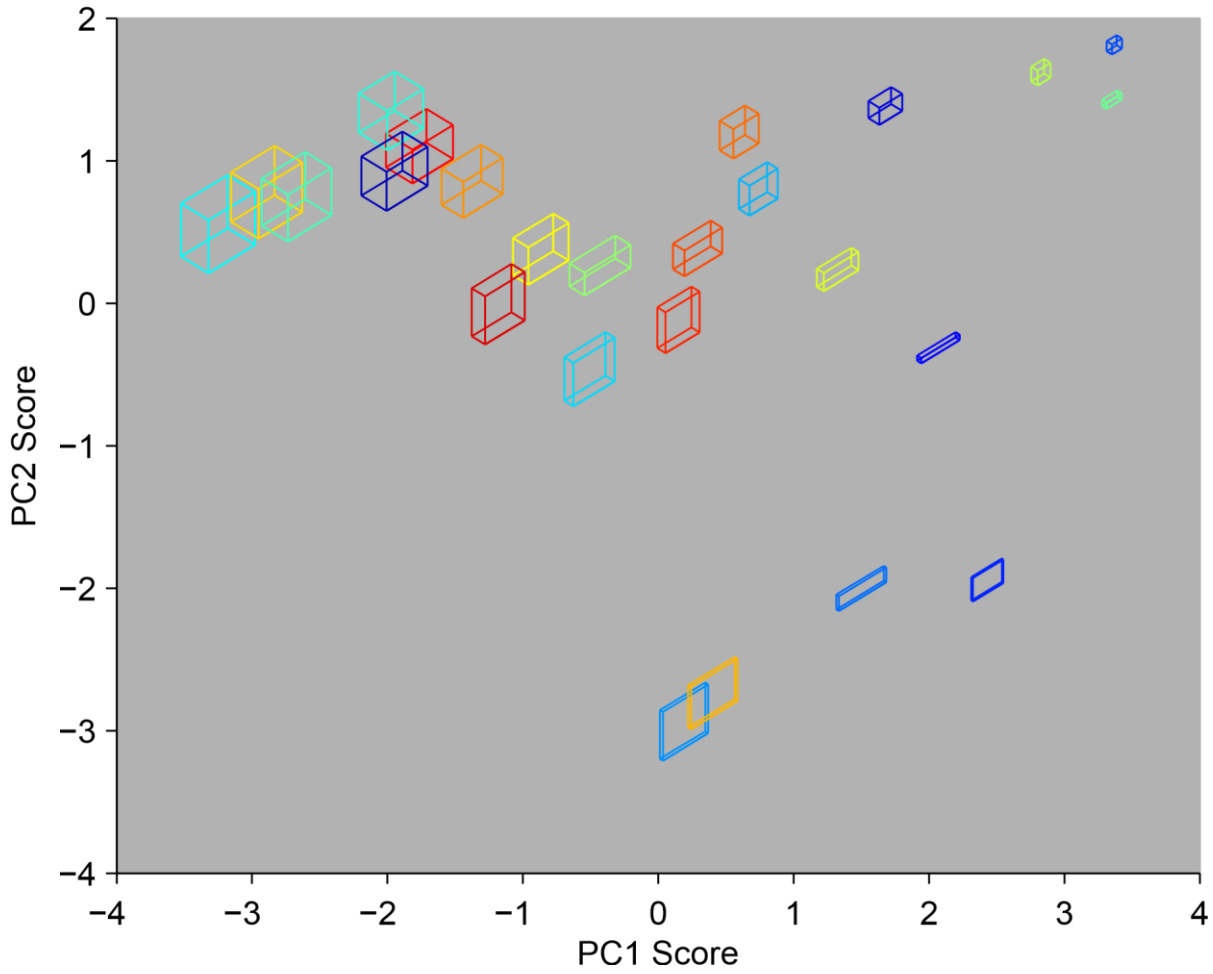


Figure 8.10: *PCA representation for the 25 blocks based on the scores of the first 2 principal components.*

We can now see that the positions of the blocks in the principal component space follow general trends according to their shape characteristics. We can get this information in more detail by studying the *loadings*, which show how the directions of the principal components are related to the original parameters.

Parameter	Component 1	Component 2
X_1	-0.37	0.44
X_2	-0.44	0.20
X_3	-0.44	-0.34
X_4	-0.47	0.30
X_5	0.15	0.69
X_6	-0.47	-0.29

Looking at the first principal component we can see that all the parameters, except X_5, change in the same way (*i.e.*, they have the same sign). So as the scores on the first principal component increase, the parameters X_1, X_2, X_3, X_4 and X_6 will decrease (because of the minus signs on the loadings) and X_5 will increase. These loadings make sense when we look back at Figure 8.10 and see that the blocks become larger as the scores on the first principal component become more negative. Therefore the first principal component is orientated in such a way as to represent the overall size of the blocks. The second principal

component is a little more complicated. The parameters X_3 and X_6 have opposite signs to the other parameters. Again this is a little clearer if we look back at the scores in Figure 8.10. We can see that blocks with high scores on the second principal component tend to be more equidimensional, but those with large negative scores are more tabular. This makes sense given that in the loadings, X_3 has a different sign to X_1 and X_2. Therefore it appears the second principal component is representing the shape of the blocks rather than their size.

Hopefully, this example demonstrates the concepts behind PCA and how we can go about the interpretation of the results. Of course in a manufactured example like this one it is easier to understand the results because all the relationships between the different parameters are defined in advance. In real world examples it is necessary to study the principal component scores and loadings in detail in order to form a clear understanding of a PCA structure.

8.1.3 Oreodont skull measurements

Oreodonts were large pig-like mammals that were widespread in North America during the Oligocene and Miocene (Figure 8.11).

Figure 8.11: *Agriochoerus antiquus.*

The skulls of 82 individuals were examined and the measurements of four different parts of the skull were taken:

- Braincase width.
- Cheek tooth length.

- Bulla length.
- Bulla depth.

We'll now perform a PCA on the oreodont skull measurements. As a first step we'll load the data file (Orodont.Rdata) and plot all the variables in the data matrix, X, against each other in a series of 2D plots (Figure 8.12).

Example code: 66

```
> rm(list=ls())
> load('Orodont.Rdata')

> install.packages('R.utils') #download the utils package
> library('R.utils') #prepare package to use subplots function

> subplots(n=6, nrow=2) # prepare 6 plots in 2 rows
> plot(X[,1],X[,2],xlab='Braincase width',ylab='Cheek tooth length')
> plot(X[,1],X[,3],xlab='Braincase width',ylab='Bulla length')
> plot(X[,1],X[,4],xlab='Braincase width',ylab='Bulla depth')
> plot(X[,2],X[,3],xlab='Cheek tooth length',ylab='Bulla length')
> plot(X[,2],X[,4],xlab='Cheek tooth length',ylab='Bulla depth')
> plot(X[,3],X[,4],xlab='Bulla length',ylab='Bulla depth')
```

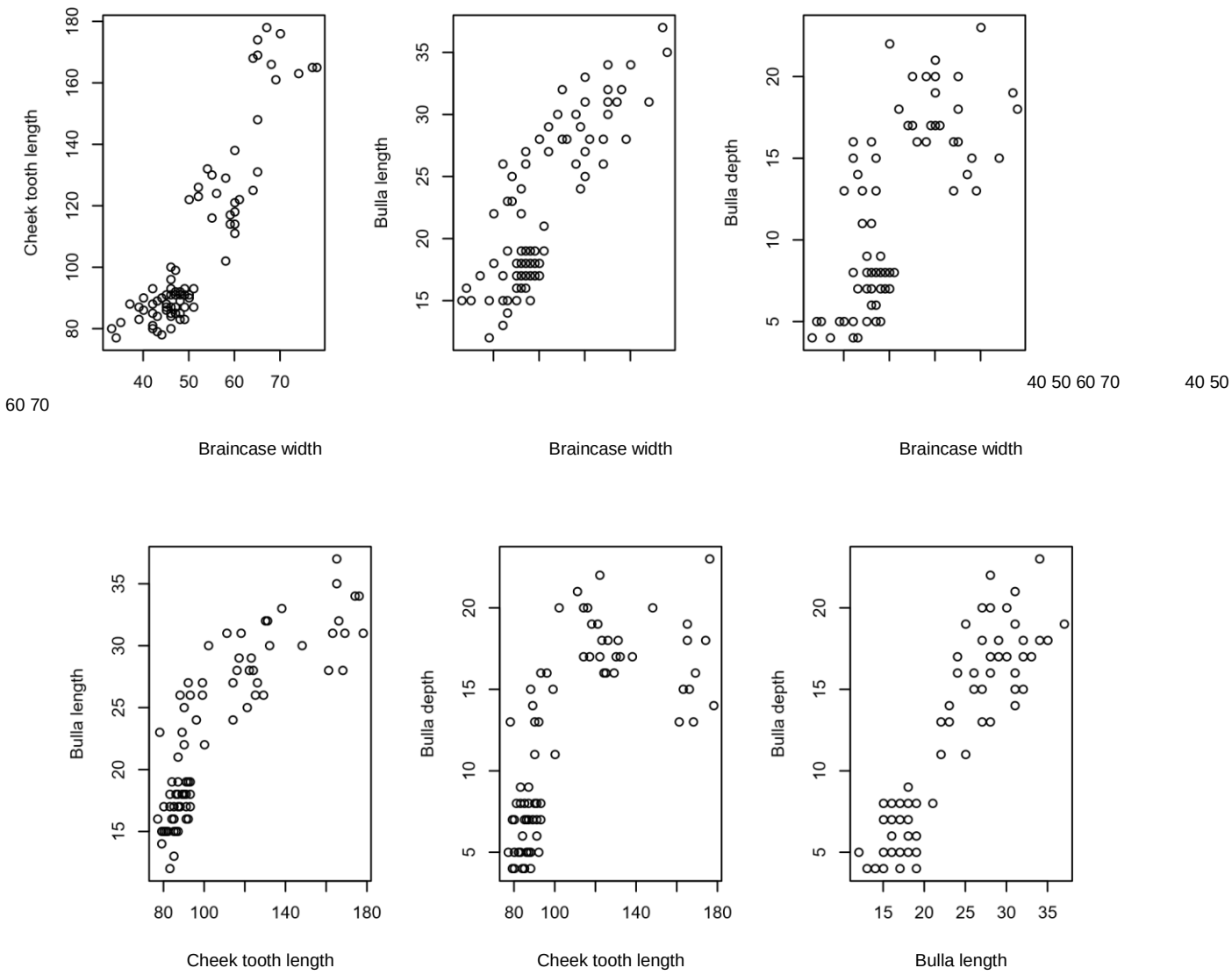


Figure 8.12: *The various oreodont skull measurements displayed as a collection as bivariate plots.*

Just as with the blocks example we can calculate the correlation matrix and plot it (Figure 8.13). We see strong positive relationships between the 4 different measurement types, this is not too surprising because we would expect that as one component of a skull gets larger, all the other components become larger as well. The high correlation suggests that a data redundancy may exist that can be exploited by PCA.

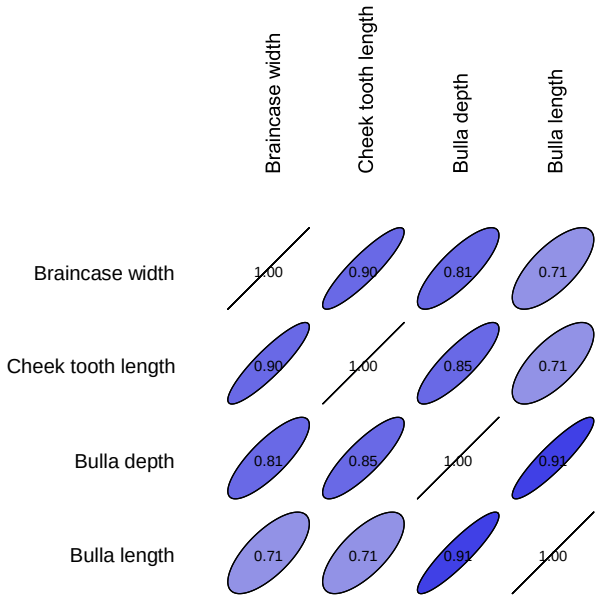


Figure 8.13: *Correlation matrix for the four sets of measurements made on the collection of oreodont skulls.*

The first step of the PCA is to standardize the data using the scale function and then process the data with the princomp function. To illustrate the results we'll then plot the first 2 principal components (Figure 8.14).

Example code: 67

```
> dev.off()
> Xz=scale(X,center=TRUE,scale=TRUE) #standardize the data
> pc=princomp(Xz) # calculate the PCA solution
> plot(pc$scores[,1],pc$scores[,2],xlab='PC1',ylab='PC2') #plot the PCA scores
```

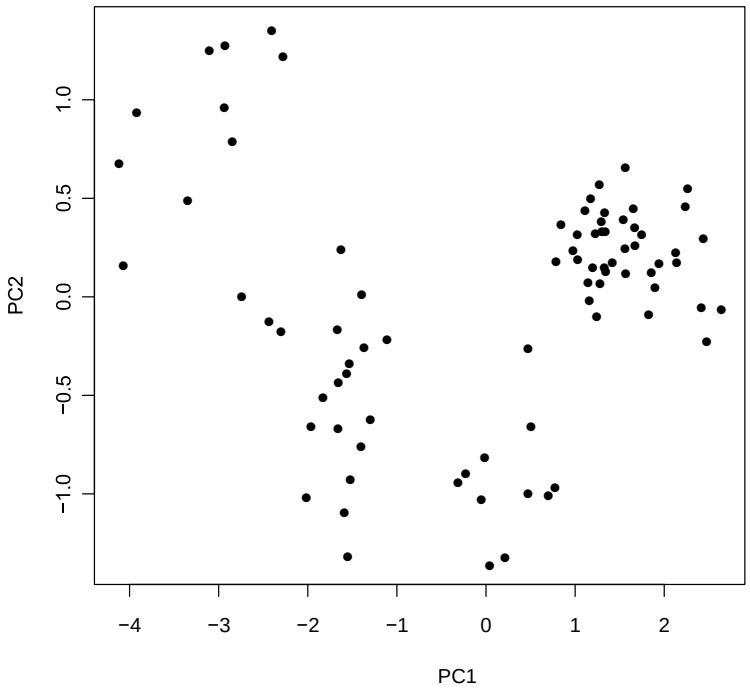


Figure 8.14: *Score of the first two principal components for the 82 oreodont skulls.*

We can see that the collection of skulls appears to fall into two different groups that may correspond to different species. To understand these groups in more detail we must first look at the scree plot to see how much of the data variability is explained by the first 2 principal components and then the principal component loadings. First the scree plot (Figure 8.15).

Example code: 68

```
> sum(pc$sd[1:2]^2)/sum(pc$sd^2) #proportion of the variance explained
> plot(pc,type='lines') # create the scree plot
```

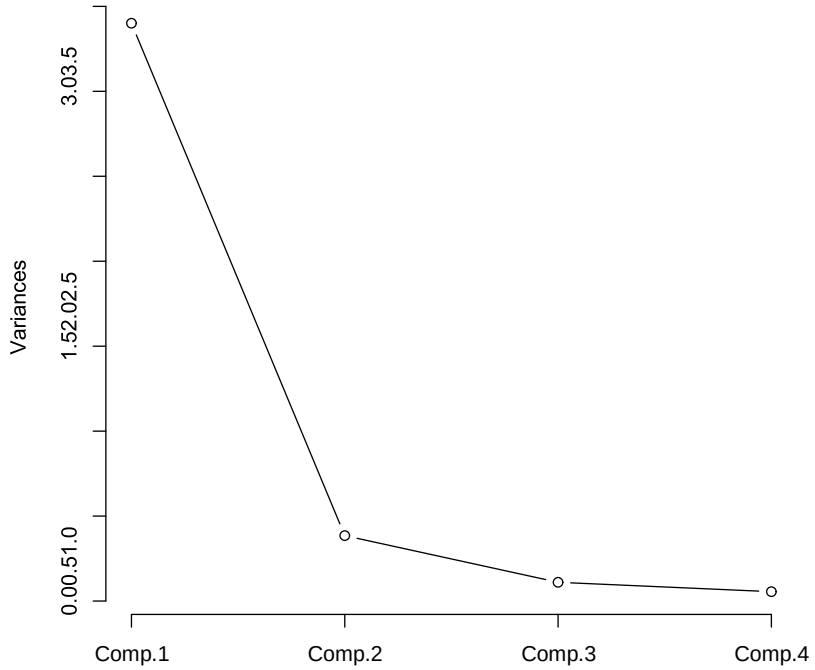


Figure 8.15: *Scree plot from the PCA of the oreodont skulls. The first 2 principal components explain ~96% of the data variability.*

The scree plot shows that the first two principal components explain about 96% of the data variability, which suggests the scores in Figure 8.14 provide a reliable representation of the data. To understand what the first two principal components represent we must look at the loadings.

Example code: 69

```
> pc$loadings[,1:2]
                          Comp.1      Comp.2 Braincase width     -0.4971044 0.4878506 Cheek
tooth length -0.5012622 0.4681118
Bulla depth                -0.5185804 -0.2901168
Bulla length               -0.4823876 -0.6772780
```

All 4 variables have a similar influence on PC1, this suggests PC1 must represent changes in skull size. For PC2, the size of the Bulla controls the component in a different way to the size of the teeth and braincase. This suggests PC2 is related to the shape of the skull.

8.1.4 Now it's your turn: Fisher's irises

The petal and sepal lengths and widths for 100 irises are stored in the variable `X` in the data file iris cluster2.Rdata. You should now be able to perform PCA on the iris data by looking at the oreodont example. To check that you're on the right track, a plot of the scores for the first two principal components is given in Figure 8.16.

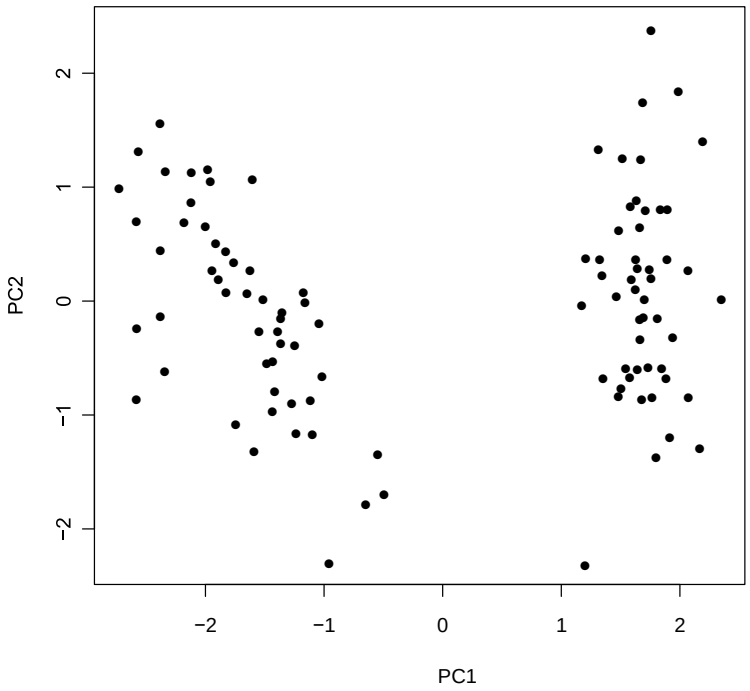


Figure 8.16: *Score of the first two principal components for Fisher's 100 irises.*

The scores show two groupings, which supports the results of our earlier cluster analysis and again indicates that two different species are contained in the sample. The first two principal components explain over 96% of the variability and therefore provide a good representation of the data set.

8.1.5 A typical application: Marine micropaleontology

There are a large number of different species of benthic foraminifera (~4000) and they respond to environmental variables such as:

- Water depth
- Temperature
- Nutrient content
- oxygen content
- and many others

A typical benthic foraminifera data set will contain the number of individuals of different taxa (maybe ~400) from different depths in a sediment core or many different surface sediment locations. In this form we may have a 400 dimensional space and we would need 79800 charts if we wanted to make bivariate scatter plots comparing each of the taxa's abundances. How can we use PCA to make the analysis of the assemblage an easier task?

Often in data sets with many variables, a number of the variables will show the same variation because they are controlled by the same process. Many of the taxa will respond to the same environmental conditions in a similar way, *i.e.*, we have data redundancy. Therefore, we can perform a PCA and use the loadings to find which taxa vary with which principal component. We know which environmental parameters control the specific taxa abundance and therefore we can make an environmental interpretation about what the individual principal components represent. There can be some complicating issues when working with abundance data such as foraminifera percentages because they are closed data, but we'll talk about this in more detail in Chapter 9.

8.2 Nonlinear mapping

A *nonlinear map* (NLM) attempts to find a way in which a high dimensional data set can be represented in 2 or 3 dimensions, whilst maintaining the underlying structure of the data. It does this by trying to find a way to position the data points in a lower number of dimensions so that the distances between all the points are the same as in the original higher dimensional data set.

One way to think of this approach is if we took a collection of data points in high dimensional space and connected each point to all the other points with springs. These springs should be just the right length so each one is not stretched or compressed and there is no stress in the system. Now imagine we take this collection of points connected by springs and without disconnecting anything, we try to rearrange the points in 2D in such a way that the system still has no stress. To do this successfully we will need to ensure that each of the springs stays the same length and is not stressed or compressed in the final configuration of points. If we can achieve this then we know that the distance of any given point to all the other points must be the same in the 2D arrangement as it was in the higher dimensional space and therefore we will have preserved the underlying structure of the data.

To make the explanation above a little clearer, we'll consider some simple examples. Imagine I generate a collection of points distributed randomly within a 3D sphere like the ones in Figure 8.17. If I try to map this collection of points in a sphere from 3D down to 2D it is reasonable to assume that I get a circle full of points.

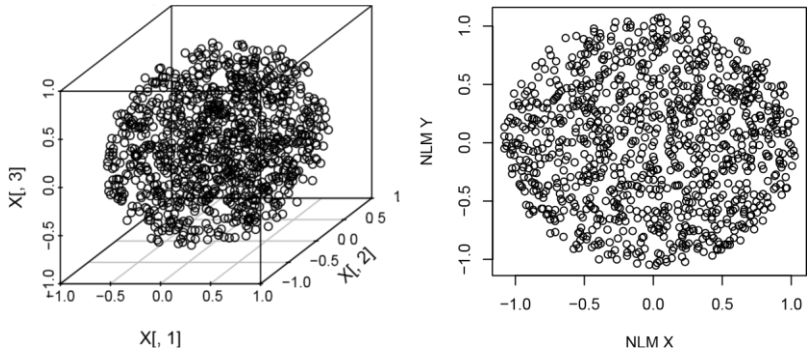


Figure 8.17: *A data set consisting of 1000 random points within a 3D sphere were generated (left). A NLM of the 3D data to map it into 2D produces points within a circle.*

Alternatively, we can start with random points in a 5D hypersphere (we can't visualize this but it is simple enough to represent mathematically). If we map from 5D to 3D we'll get a sphere full of points and if we map to 2D we'll get a circle full of points (Figure 8.18).

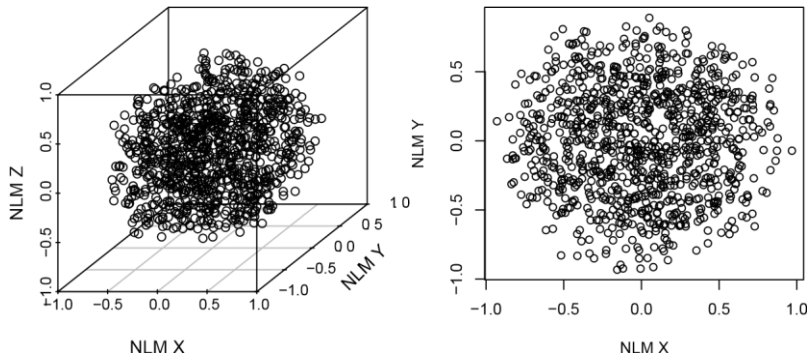


Figure 8.18: *A data set consisting of 1000 random points within a 5D sphere. A NLM of the data into 3D produces a sphere of points (left) and mapping into 2D produces a circle of points (right).*

8.2.1 How does it work

There are a number of different NLM methods (generally termed *multidimensional scaling*), we will focus on the method of Sammon (1969) because of its simplicity. If our starting data set has *m* dimensions and we want to map it into *p* dimensions (therefore $p < m$), we need to define two different sets of Euclidean distances:

δ_{ij} = the distance between points *i* and *j* in *m*-dimensional space

d_{ij} = the distance between points *i* and *j* in *p*-dimensional space

For any mapping into *p* dimensional space we can calculate the *stress* [0,1], which is given by:

$$E = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{(\delta_{ij}-d_{ij})^2}{\delta_{ij}} \qquad (8.1)$$

A value of $E = 0$ means we get a perfect representation of the *m* dimensional data structure in *p* dimensions, and as *E* increases towards 1 the representation becomes poorer. The computer does the hard work, testing lots of different mappings to find which one gives the lowest *E* value (although the mapping most probably won't be perfect).

8.2.2 NLM example: Fisher's irises

We've already studied Fisher's irises using both cluster analysis and PCA. We reached the conclusion that the sample may contain two different species and we can now check that we find a similar structure in the data using NLM. As before, the lengths and widths of the petals and sepals for 100 irises are stored in the variable X in the data file NLM iris.Rdata. We'll load the data and standardize each of the columns. To calculate the NLM we'll use the R function sammon, which is contained in the package install.packages. The required input for sammon is a distance matrix corresponding to δ in equation 8.1. We'll calculate the distance matrix using the dist function directly in the call to sammon (Figure 8.19).

Example code: 70

```
> rm(list=ls()) # clear the memory
```

```
> dev.off() # close all the graphics windows
> install.packages('MASS') #download the package including sammon
> library(MASS) #prepare the package for use
> load('NLM_iris.Rdata') # load the iris data for 100 plants
> Z=scale(X,center=TRUE,scale=TRUE) # standardize the columns
> Y = sammon(dist(X),k=2) # NLM of distance matrix of X
> plot(Y$points[,1],Y$points[,2],xlab='NLM X',ylab='NLM Y') # plot NLM points
```

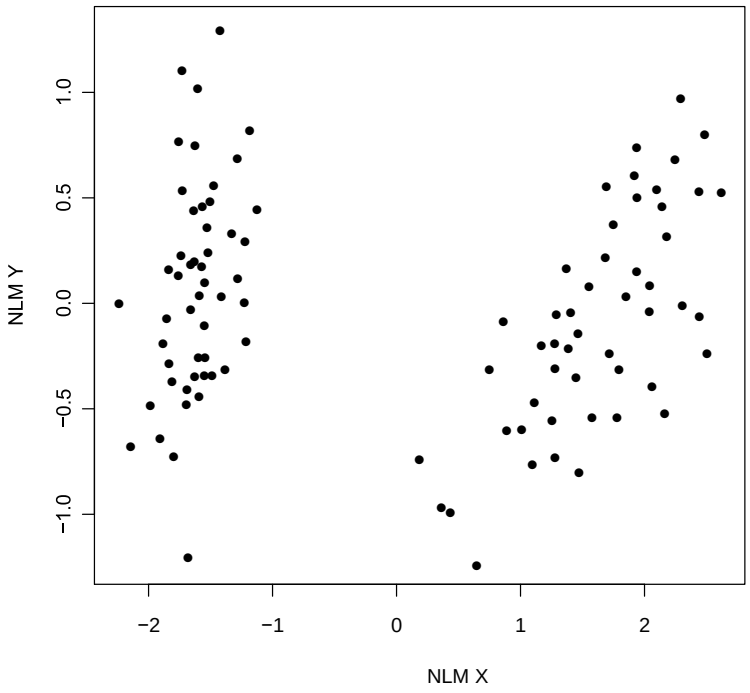


Figure 8.19: *NLM of Fisher's iris data, mapping the data from 4 dimensions down to 2 dimensions.*

So, on the basis of the flower measurements the NLM also reveals the presence of two distinct groups of irises, which may be two different species. An important point to note is the names and units of the NLM axes. As we saw in equation 8.1 the NLM is based on the distances between points and the axes will combine the different variables together in such a way that we cannot give meaningful names or units to the axes. Therefore we just assign the axes arbitrary names, like the ones in Figure 8.19, which show that we are dealing with an NLM.

8.2.3 NLM example: Portuguese rocks

In Section 7.4 we looked at an example data set composed of oxide concentrations for 134 Portuguese rocks. The collection of examined rocks was known to include granites, diorites, marbles, slates, limestones and breccias. The measured oxides included (all expressed as %); SiO_2, Al_2O_3, Fe_2O_3, MnO,

CaO, MgO, Na_2O, K_2O and TiO_2. We'll extend the data set and include an additional 9 physical-mechanical measurements, which are as follows:

- RMCS: Compression breaking load (kg/cm^2).
- RCSG: Compression breaking load after freezing tests (kg/cm^2).
- RMFX: Bending strength (kg/cm^2).
- MVAP: Volumetric weight (kg/m^3).
- AANP: Water absorbtion (%).
- PAOA: Apparent porosity (%).
- CDLT: Thermal linear expansion coefficient ($10^{-6}/^oC$).
- RDES: Abrasion test (mm).
- RCHQ: Impact test: minimum fall height (cm).

We now have an 18 dimensional data set and if we wanted to create a collection of scatter plots showing all the different combinations of variables we would need 144 plots. Instead we can try to reveal the underlying structure of the data by using NLM to map it into 2D (Figure 8.20). This is just like the example above, which requires standardization of the columns in the data matrix and the calculation of the interpoint distances.

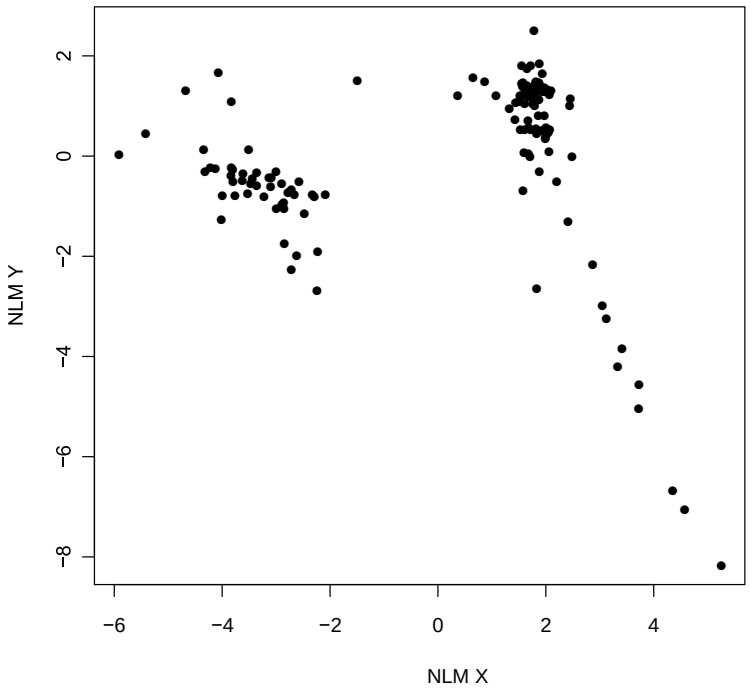


Figure 8.20: *NLM of the Portuguese rocks data set containing both oxide and physical-mechanical measurements. The data is mapped from 18 to 2 dimensions.*

8.2.3.1 Cluster analysis and NLM

When we studied k-means clustering we analyzed the Portuguese rocks data set and concluded that there were two different groups of data. When working with multivariate data sets it is a challenge to represent the results of a cluster analysis because the cluster centers exist in the same high dimensional space as the original data. One solution to the problem is to try to represent the original data in a lower number of dimensions, for example by using PCA or NLM, and then building the results of the cluster analysis into that solution. As an example, Figure 8.21 takes the 2D representation of the Portuguese rocks data set obtained by NLM, but codes the points according to the cluster assignments that were found in Section 7.4. In this way we can combine the techniques of cluster analysis and dimension reduction to give more detailed insights into the data.

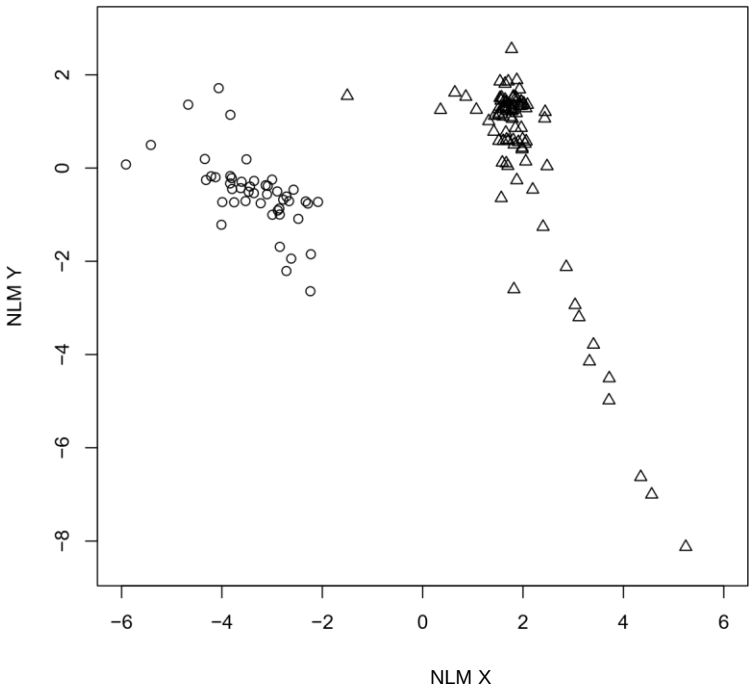


Figure 8.21: *A 2D NLM representation of the 18 dimensional rocks data set. The results of a 2 cluster k-means solution are combined with the NLM representation by plotting each point with a symbol according to the cluster center to which it is assigned (i.e., all the points marked by circles belong to one cluster and all the points marked by triangles belong to another cluster.)*

We can repeat the same process for more complex cluster solutions (i.e., ones involving more cluster centers). The NLM doesn't change, but the points are assigned symbols according to which of the clusters they are assigned to. A 3 cluster solution is shown in Figure 8.22 and a 4 cluster solution is shown in Figure 8.23.

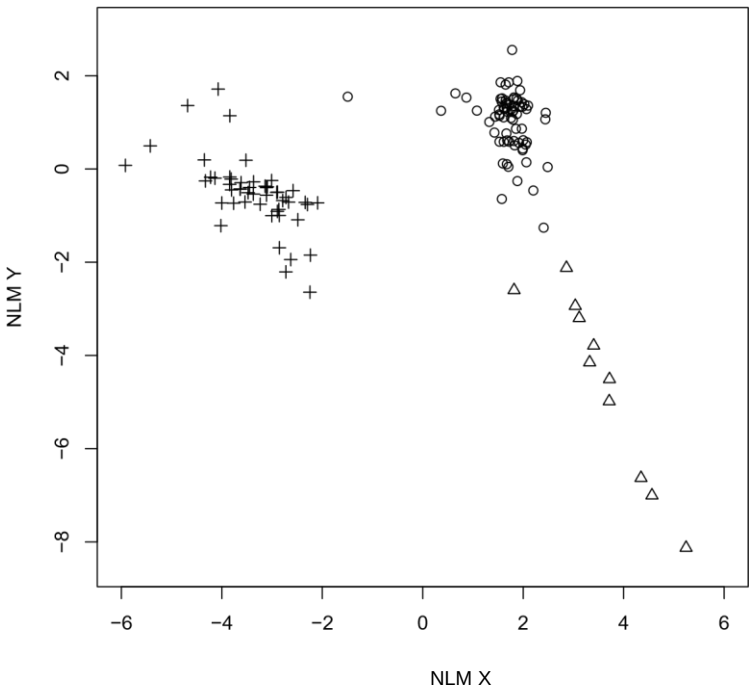


Figure 8.22: *A 2D NLM representation of the 18 dimensional rocks data set. The results of a 3 cluster k-means solution are combined with the NLM representation by plotting each point with a symbol according to the cluster center to which it is assigned.*

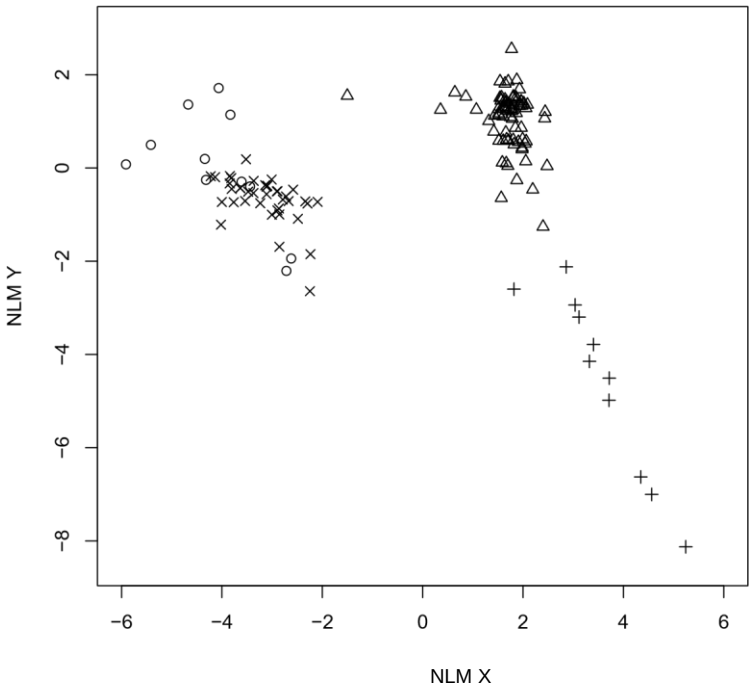


Figure 8.23: *A 2D NLM representation of the 18 dimensional rocks data set. The results of a 4 cluster k-means solution are combined with the NLM representation by plotting each point with a symbol according*

to the cluster center to which it is assigned. Two of the clusters (marked by o and x symbols) overlap strongly, which suggests that 4 clusters are too many to give an appropriate representation of the data.

8.2.4 Nonlinear dimensionality reduction by locally linear embedding

Locally linear embedding (LLE) is a recently developed technique that, as suggested by the name, focuses on the local structure of the data rather than the overall global structure. Sammon's approach to NLM employs the distances between all of the points in a data set, therefore it adopts a global view of the data structure. LLE also attempts to find a low dimension representation that preserves the distances between points, but for any given data point it only considers the distances to its nearest neighbors. Therefore local structure should be preserved and, hopefully, if the local structure is well preserved then the global structure will also be preserved. An example of this approach is given by the 3 dimensional *Swiss roll* function (Figure 8.24).

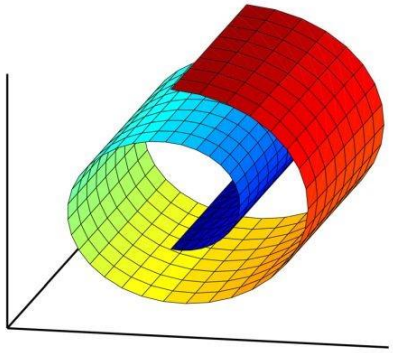


Figure 8.24: *The 3D swiss roll function. Notice how the function evolves with a looping form.*

We can see that local structure in the Swiss roll function is very important. If we only consider inter-point distances then we get a false impression of the structure of the function. For example, the straight-line distance between the start (red) and end (blue) of the function is less than the distance between the start and the point where the function has completed half a loop (yellow). Therefore using only interpoint distance we would have to conclude that the blue region of the function is more closely related to the start of the function than the yellow region. However, since we can see how the function evolves, it is clear that the yellow part of the function is in fact more closely related to the start of the function. A global technique like NLM is not designed to preserve the kind of evolving structure that we see in the Swiss roll function, but LLE is. The idea behind LLE is to find networks of neighboring points in high dimensional space, like those in Figure 8.25, and in an approach quite similar to Sammon mapping find a representation that preserves the inter-point distances in 2D. This is repeated for each data point and thus we obtain a low-dimensional mapping that focusses on local rather than global structure.

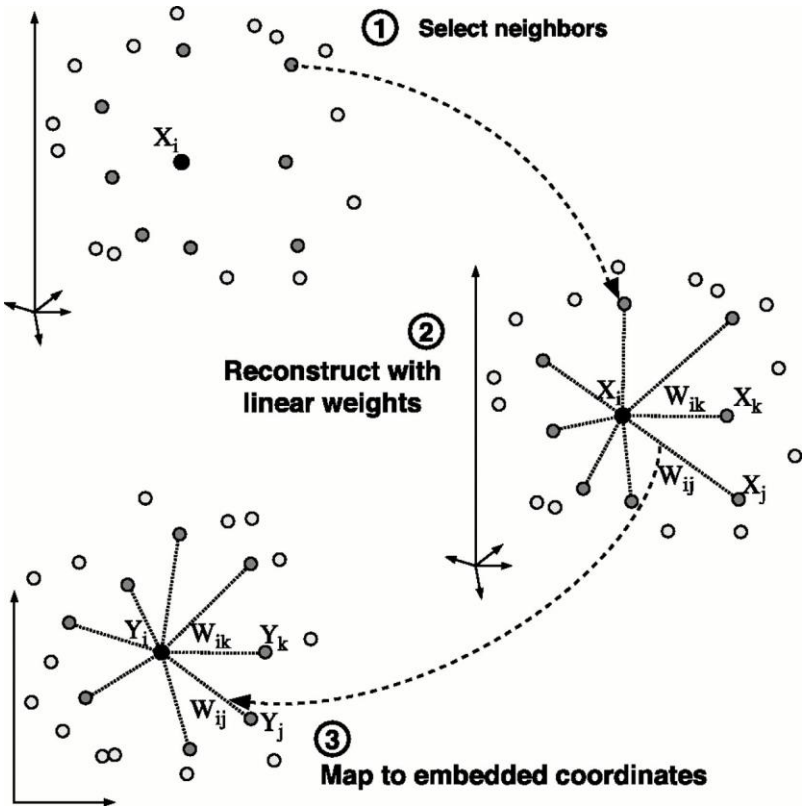


Figure 8.25: *Schematic showing how LLE works. Distances are preserved within local neighbourhoods rather than globally, therefore the low-dimensional mapping focusses on local structure. Image taken from http://cs.nyu.edu/~roweis/lle/algorithm.html*

So, what happens when we apply LLE and Sammon mapping to the Swiss roll function (Figure 8.26). Because LLE is based on local behavior, the Swiss roll is successfully unrolled and the local structure is clearly preserved. The Sammon map, however, performs less well and simply compresses the function along its long axis, failing to uncover the true underlying structure.

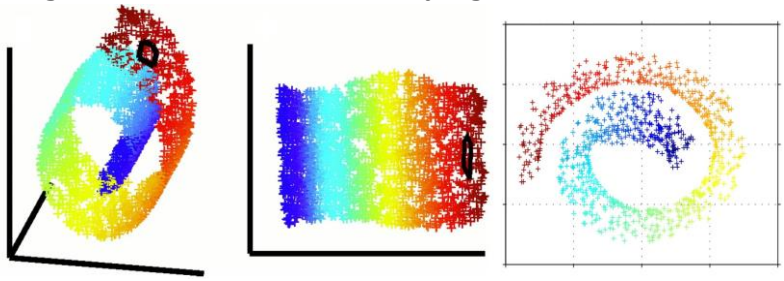


Figure 8.26: *Examples of the Swiss roll function in 3D (right) mapped into 2D using the LLE (middle) and Sammon approaches (right). Notice the fundamental differences in the final mappings that result from considering local or global structure.*

8.2.5 Nonlinear principal components analysis

Dimension reduction is an active field of research. As all kinds of databases become larger and larger we need to find ways in which to analyze their content quickly and represent the information they contain

in a concise way. This has led to the development of techniques such as nonlinear principal components, where the components are no longer straight-lines. This means that data sets that may contain non-linear structures can be represented efficiently. An example of nonlinear principal component analysis is given in Figure 8.27.

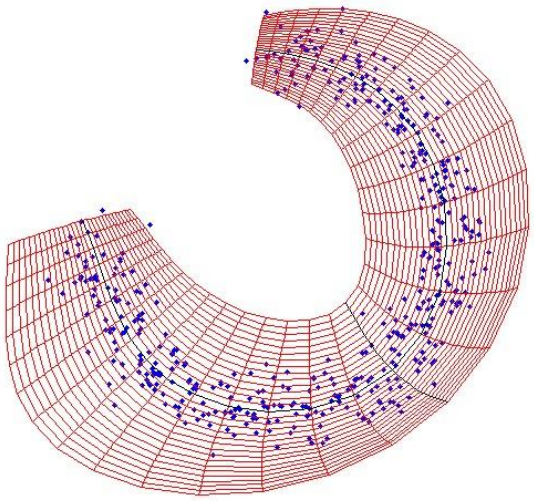


Figure 8.27: *An example of nonlinear principal component analysis where the individual components can be curved in order to represent nonlinear structures in a data set (taken from http://www.nlpca.de/).*

The study of 'non-Euclidean' geometry brings nothing to students but fatigue, vanity, arrogance, and imbecility.

Matthew Ryan

9 Analysis of compositional data

In this chapter we're going to look at compositional analysis, which is important in many fields of the geosciences. Compositional data (also know as *closed data*), such as the percentage of different minerals in a rock or the abundances of different microfossils in a sediment, are subjected regularly to statistical analysis but they have a number of issues that require special attention. We'll start by looking at the problems associated with compositional data and then see how we can go about solving them.

9.1 Absolute and relative information

Imagine that in my kitchen I keep a jar full of jelly beans with three different colours; red, green and blue. My flatmates like to eat my jelly beans without telling me, but sometimes they also buy new jelly beans and refill the jar. My flatmates are also picky about which colours of jelly beans they will eat, Richard only eats red, Greg only eats green and Becky only eats blue. To monitor how my jelly beans are being eaten and replaced, I decide to stop eating them myself and instead keep a record of how the content of my jar changes over a week. To start I put 50 red, 40 green and 25 blue jellybeans in the empty jar and then count them again after 1 week. The results look like this:

Time	Number of red	Number of green	Number of blue
Start	50	40	25
Finish	55	5	20

So, the results are clear. There are more red jelly beans in the jar than at the start of the experiment, therefore Richard must have bought a new bag of red beans. Greg owes me 35 green jelly beans and Becky owes just 5 blue.

What would have happened in the same experiment if I had expressed all the results as percentages rather than the absolute number of jelly beans. The same results would look like this:

Time	Percentage of red	Percentage of green	Percentage of blue
Start	43	35	22
Finish	69	6	25

How can we interpret these results? The percentage of red and blue have both increased, does that mean that both Richard and Becky both bought replacement beans? Well we know from the absolute numbers of beans that only Richard bought new beans, but this information has been lost when we express the results as percentages. What we can see is that the percentage data only carries relative information, which is how abundant one colour of bean is compared to the other colours of beans. Because we have no absolute information we can't say how the number of individual colours of beans are changing. This is a key concept, *compositional data only carry relative information*.

9.2 Properties of compositional data

In the previous section we saw that compositional data only carry relative information. In statistical analysis they also require special consideration because they are constrained in two important ways.

- All the components must be represented by non-negative numbers (i.e., numbers that are >0).
- The contributions to each composition must total 100% (or 1, or similar).

This second requirement is called the *sum-to-one* constraint. An example of closed data are sediment grain sizes that are split into sand, silt and clay size fractions. For any given sample, the sum of the contributions from each of the size fractions must add to 100%.

Sample	Sand [%]	Silt [%]	Clay [%]	**Total** [%]
1	77.5	19.5	3.0	**100.0**

2	71.9	24.9	3.2	**100.0**
3	50.7	36.1	13.2	**100.0**
4	52.3	41.0	6.7	**100.0**
5	70.0	26.5	3.5	**100.0**
6	66.5	32.2	1.3	**100.0**
7	43.1	55.3	1.6	**100.0**
8	53.4	36.8	9.8	**100.0**
9	15.5	54.4	30.1	**100.0**
10	31.7	41.5	26.8	**100.0**

Because of the sum-to-one constraint, all the information of a D component composition can be given by $D - 1$ components. For example consider the grain size compositions for three more samples, where missing values are shown with question marks.

Sample	Sand [%]	Silt [%]	Clay [%]	**Total** [%]
1	21.3	57.5	?	**100.0**
2	?	13.2	10.7	**100.0**
3	63.2	?	7.8	**100.0**

Even though each case is represented with only two values, we can immediately work out what the percentage of the remaining grain size fraction will be because the total for each case must be 100%. Therefore all the information on the 3 grain size components can be represented with just 2 components.

This might seem like a trivial issue, but its effect can be dramatic. As an example think of the case of a marine micropaleontologist, who has counted the numbers of two different foraminifer taxa, *A* and *B*, through a sediment core. They now want to test how *A* and *B* correspond to each other because it is possible that the two taxa abundances are controlled by the same environmental conditions. There are two ways to perform this analysis, by comparing the absolute numbers of individuals or by comparing the percentage abundances of the two species (Figure 9.1)

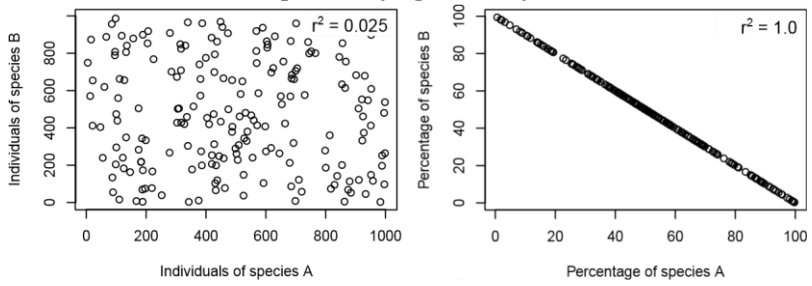


Figure 9.1: *The comparison of the foraminifer taxa A and B expressed in absolute counts (left) and as percentages (right). The r^2 values give the coefficient of determination for each comparison.*

We can see that the results of the two forms of analysis are dramatically different. When we employ counts of individuals there is clearly no significant correlation between *A* and *B*, but when we use percentages we obtain a perfect negative correlation. The reason behind this result is clear, if we have a given percentage of species *A*, then the corresponding percentage of species *B* must be (100-*A*)%. Therefore, when we plot a collection of results we'll obtain a perfect negative relationship that is purely a product of the manner in which we have decided to express the data and has no physical meaning.

You might have two responses to this problem. First, you could decide never to use percentage data, which is a nice idea but almost impossible in practise. Some forms of data can only be expressed in a relative sum-to-one form, for example mineral composition that may be given in %, *ppt*, *ppm*, etc. Although it is possible to express data such as foraminifer abundance in absolute terms, in practise it is very difficult, therefore most assemblage information is normally given in percentages. Your second response to the problem could be to say that it only deals with 2 parameters whereas your own data set contains many more, for example, 50 different foraminifer taxa. Unfortunately this doesn't solve the problem, the sum-to-one constraint will still induce false correlation when different components of a composition are studied, no matter how many parts it is made up from. To my knowledge (at the time of writing), there is no statistical technique available that can quantify correctly the correlations between the different parts of a composition (this is worrying when you think how often you see such correlations employed in the scientific literature).

We have now seen some of the problems caused by the sum-to-one constraint, but in some cases we can use it to our advantage. You'll be familiar with representing grain size data in a triangular ternary plot, where each edge of the triangle represents the abundance of one of the grain size fractions. Our ability to represent such 3D data sets in a 2D figure (a triangle) without any loss of information is a direct result of the sum-to-one constraint (Figure 9.2).

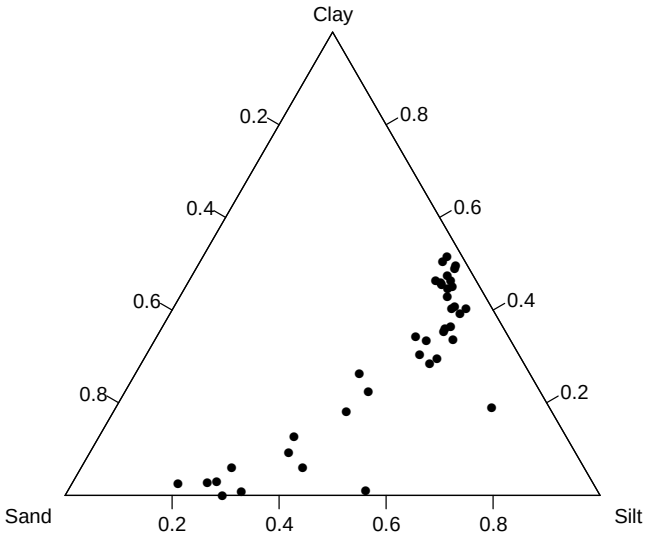


Figure 9.2: *Example of grain size data characterized by the relative proportions of sand, silt and clay, plotted in a 2D ternary plot.*

Of course for a given case each of the abundances of the grain size fractions must also be >0, otherwise the case will plot outside of the triangle. The non-negativity and sum-to-one constraints should make intuitive sense to us. Imagine that someone told you that "*my sediment contains -20% sand*" or "*the three*

grain size fractions in my sediment add up to 105%" you would know that there is something wrong with their data.

9.2.1 The simplex as a sample space

In the case of the grain size data we looked at above we can see that as a result of the sum-toone and non-negativity constraints the data points are restricted to lying within a triangle. Let's think about a simpler case where our data is constructed from just two components, which we'll call *A* and *B*. If we represent these two parameters in a Euclidean sample space, we find that points are only allowed to lie on a diagonal line between the coordinates (1,0) and (0,1). If a data point sits anywhere except this line then it would need to either contain negative components (therefore breaking the non-negativity constraint) or $A+B$ 6= 1 (therefore breaking the sum-to-one constraint). So the compositional data are forced to lie on a 1D line that exists in a 2D space (Figure 9.3).

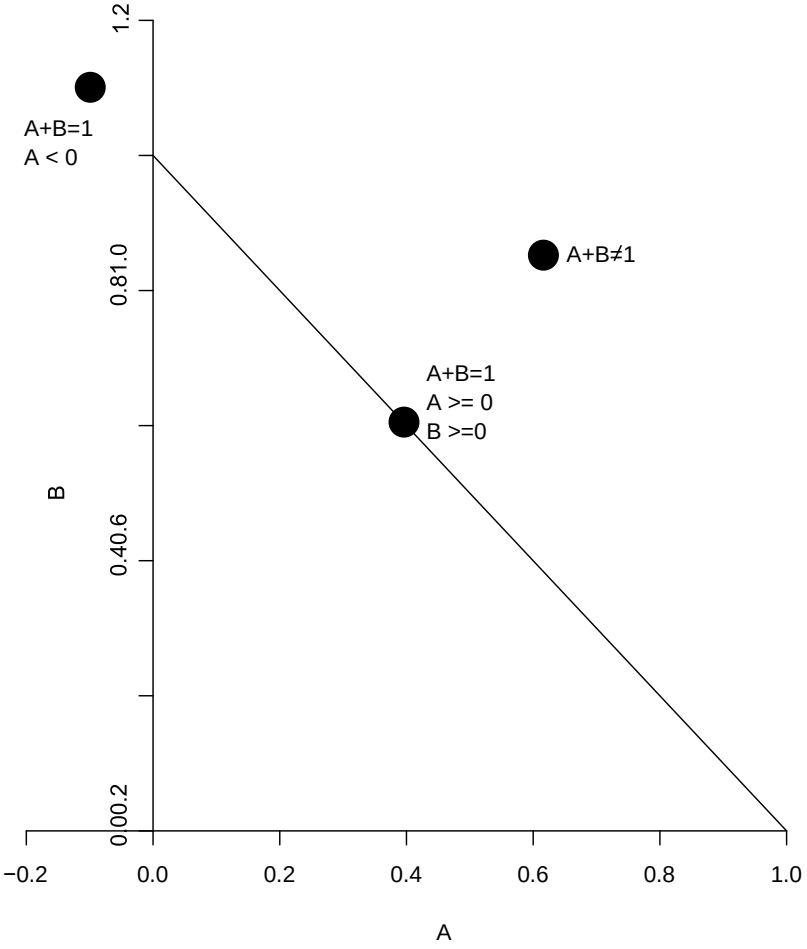


Figure 9.3: *Example of the constraints that act on 2 part compositions. Such compositions must lie on the diagonal line between (1,0) and (0,1), which means they meet both the non-negativity and sum-to-one constraints. Points that do not fall on the line must violate one or both of the constraints.*

Similarly, if we consider compositions with three components, all of the cases must lie on a 2D triangle within a 3D space. Because of the sum-to-one constraint the corners, or vertices, of the triangle must be positioned at (1,0,0), (0,1,0) and (0,0,1), Figure 9.4.

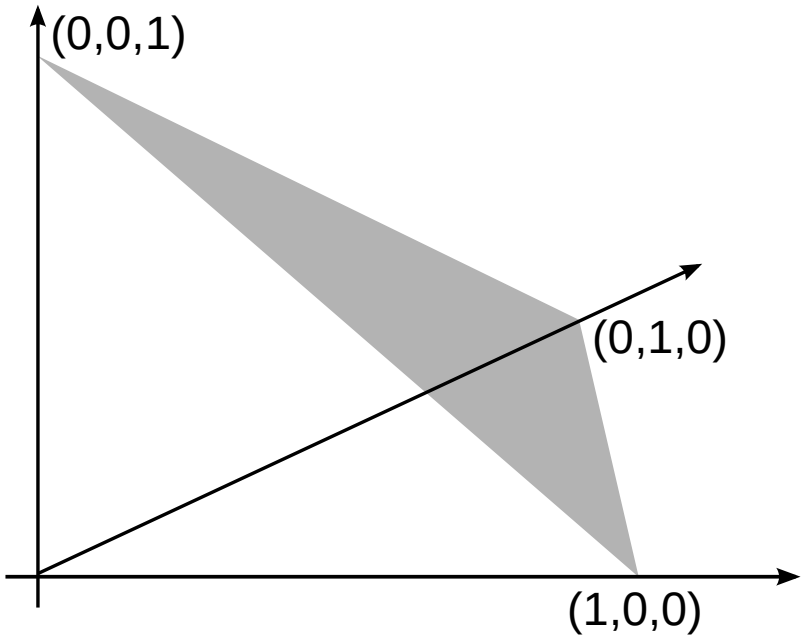


Figure 9.4: *Example of the constraints that act on 3 part compositions. Such compositions must lie on a triangle (shaded) between the points (1,0,0), (0,1,0) and (0,0,1).*

We can extend this idea to higher numbers of dimensions, but they become more difficult to represent in a diagram. For example a composition with four components must lie within a 3D tetrahedron with the corners (1,0,0,0), (0,1,0,0), (0,0,1,0) and (0,0,0,1).

The general rule that emerges is that a D component composition must reside within a D-1 dimensional simplex. So a 1D simplex is a line, a 2D simplex is a triangle, a 3D simplex is a tetrahedron, a 4D simplex is a pentachoron, etc. This means that compositions are not allowed to exist within any point in Euclidean space, but instead they reside in a so-called simplicial sample space. At this point you might think "so what", but simplicial sample spaces have a number of difficulties associated within them.

9.3 Important aspects of compositional analysis

We have seen in the earlier sections that compositional data have special properties that make them different to ratio-scale data. For example, most statistical techniques assume that the data can potentially range between $-\infty$ and ∞, whereas compositional data will lie in the interval [0,1] or in the case of percentages [0,100], etc. Additionally, most statistical methods assume that the data are held in a Euclidean space, where the dimensions are at 90^o to each other. If we think about the simple case of the ternary diagram, we can see that the dimensions are at 60^o to each other, so we clearly have a problem.

The problems of working with statistics in a simplex are more fundamental than you might think. This is because Euclidean geometry, which is fundamental to statistical analysis, does not work within a simplex. We'll look at some of these issues now.

9.3.0.1 Distances between points in a simplex

When studying cluster analysis we saw how the similarity between two data points can be measured in terms of the distance separating them. This is simple for points in a Euclidean space, where we can just

measure the distance between points along each dimension and then use Pythagoras' theorem to find the length of the line separating the points. When we try something similar in a ternary plot we hit a problem. The dimensions are at 60^o to each other, so when we draw lines along the dimensions we don't form a right-angled triangle and we can't apply Pythagoras' theorem (Figure 9.5). Maybe at this stage we shouldn't be too worried because we can simply use a bit of extra geometry that doesn't rely on Pythagoras' theorem to find the distance between the two points. However, we do have a problem because the algorithms that are available for calculating various statistics assume that you can use Pythagoras' theorem, so you're going to have a lot of work if you plan to rewrite them all. In fact the problem is more fundamental because what we think is a straight-line in our Euclidean minds has a different form in simplicial space.

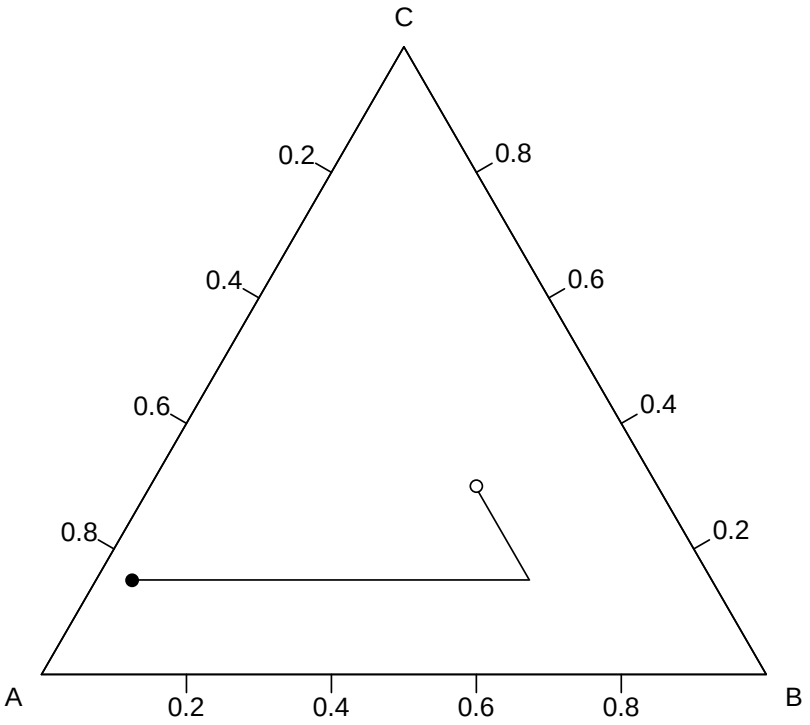


Figure 9.5: *Because the angles between the dimensions in a simplex are not 90^o we can't use Pythagoras' theorem to find the distance between the two points.*

9.3.0.2 Straight lines in a simplex

In the previous section we found that we couldn't employ Pythagoras' theorem in a simplex because the dimensions are not at 90^o to each other. Now we're going to make things even worse! We are used to thinking in terms of Euclidean geometry, where a straight-line is "straight". In the case of the simplex however, the dimensions are not at right angles and straight-lines actually appear as curves (Figure 9.6).

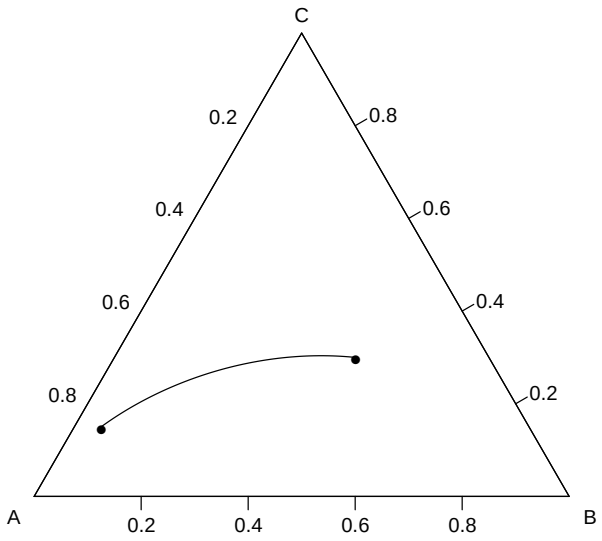


Figure 9.6: *The shortest path between two points in a simplex appears as a curve rather than a straight-line because the dimensions are not at right-angles to each other.*

We won't dwell on this point because geometry within a simplex is a whole field within itself (if you're interested you should see the recommended reading for more details). The fact that the shortest path connecting two points in a simplex appears as a curve should re-emphasize the point that we really won't be able to use Pythagoras' theorem and Euclidean distances.

9.3.1 Statistics without Euclidean distances

Okay, we can't use Euclidean distances with compositional data, but how much does that really limit us? We saw that Euclidean distances are employed in regression analysis (in the form of residuals) and clustering (to assess similarity), but they are also important in very basic statistics. For example, we all know that you find the mean of a sample of numbers by summing them and dividing by the size of the sample. So we know how to calculate a mean, but what does it really represent? Well it give a proper definition:

The arithmetic mean minimizes the sum of squared Euclidean distances to the individual values.

So Euclidean distances are even involved in a simple mean, therefore we can't even calculate something as basic as an average composition using standard statistical methods.

9.4 Solving the problems

The problems associated with the statistical analysis of compositional data were first identified by Karl Pearson in 1896. An effort was then undertaken to redesign a number of different statistical methods so they could be applied to compositional data. However, to go through every existing statistical approach and find ways to adjust it for compositional data was just too large a task and the problems associated with compositional data became largely ignored. Instead standard statistical techniques would be

applied to compositional data ignoring the fact that the results could be entirely spurious. Occasional papers were published warning of the problems with compositional data analysis, but they were largely ignored because no solutions to the issues could be given.

The framework for the statistical analysis of composition data was set out by John Aitchison in a series of papers in the 1980s and 1990s. Aitchison's solution to the problem was not to try to redesign all statistical approaches to make them suitable for compositional data, but instead he developed a method to transform composition data so that it could be analyzed using standard statistical approaches. Aitchison defined two key criteria that the analysis of compositional data must meet, which we'll look at now.

9.4.1 Scale invariance

As we saw above, compositional data only carry relative information and any statistical analysis should respect this issue. As the name suggests, scale invariance simply implies that the results of a statistical analysis should not depend on the units that the composition is expressed in. For example if we choose to use %, *ppt* or *ppm* the inferences we draw from the statistical analysis should always be the same.

9.4.2 Subcompositional coherence

We'll start with Aitchison's definition of subcompositional coherence:

Subcompositional coherence demands that two scientists, one using full compositions and the other using subcompositions of these full compositions, should make the same inference about relations within common parts.

To illustrate this problem we'll consider a sample composed of 25 specimens of hongite. One investigator defines the composition of the hongites by percentage occurrence of the five minerals; albite, blandite, cornite, daubite and endite. Another investigator decides that they will only examine the three minerals; blandite, daubite and endite. To demonstrate the differences in the way the data is represented by the two investigators, consider the example of one specimen.

Investigator	Albite[%]	Blandite[%]	Cornite[%]	Daubite[%]	Endite[%]	Total[%]
1	48.8	31.7	3.8	6.4	9.3	100
2	X	66.9	X	13.5	19.6	100

Both investigators want to find out if there is a relationship between the relative abundances of daubite and endite, so they perform a correlation analysis using all 25 specimens. The results of this analysis are shown in Figure 9.7.

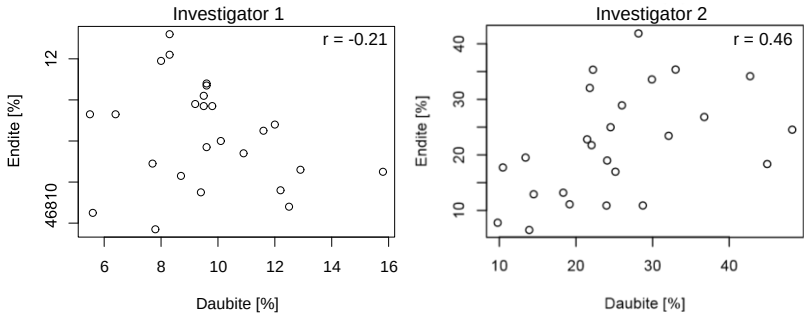


Figure 9.7: *The correlation of daubite and endite for the two investigators*

Notice that the correlations obtained by the two investigators don't even have the same sign, this means they would reach very different conclusions about how daubite and endite are related in their specimens. This result demonstrates that correlations do not exhibit subcompositional coherence (they are said to be *subcompositional incoherent*) and the results of the statistical analysis depend on which minerals were included in the original mineral quantification. Clearly, this is not a good situation to be in.

9.5 Log-ratio analysis

Aitchison developed the *log-ratio transform*, which allows compositional data to be transformed in such a way that they can be examined using standard statistical methods. The development of the log-ratio transform takes into consideration both scale invariance and subcompositional coherence. One form of transform developed by Aitchison is the *additive log-ratio* (*alr*). To perform the *alr* we must choose one variable as a denominator and divide all the other variables by that denominator and then take the logarithm of the resulting ratios:

$$y_{ij} = \log(x_{ij}/x_{iD}) \tag{9.1}$$

Here, x_{ij} and x_{iD} are the j^{th} and D^{th} constituents of the i^{th} specimen. So, in a specimen with 3 components (*A*,*B*,*C*) the *alr* would result in:

$$\left[\log\left(\frac{A}{C}\right), \log\left(\frac{B}{C}\right)\right]$$

As an example, the *alr* of the composition [80, 15, 5]%:

$$alr\,[80, 15, 5] = \left[\log\left(\frac{80}{5}\right), \log\left(\frac{15}{5}\right)\right] = [2.77, 1.10]$$

Let's think about the *alr* in more detail. First it is based on taking ratios of the parameters, which fits with us only having relative information because a ratio shows how one component is related to another. Ratios will give us scale invariance because a coefficient relating to the use of a given unit will disappear. For example, let's consider 2 mineral components, *A* and *B*, with percentage abundances of 80% and 20%, respectively. We could also write the abundances in terms of proportions (0.8 and 0.2), *ppt* (800 and 200), *ppm* (800,000 and 200,000), etc. When we take the ratio the units fall away and all the systems give the same result:

$$\frac{A}{B} = \frac{80}{20} = \frac{0.8}{0.2} = \frac{800}{200} = \frac{800,000}{200,000} = 4$$

Ratios also help to provide subcompositional coherence because the ratios within a subcomposition are equal to the corresponding ratios within the full composition. Returning to the composition of our first hongite specimen:

Investigator	Albite[%]	Blandite[%]	Cornite[%]	Daubite[%]	Endite[%]	Total[%]
1	48.8	31.7	3.8	6.4	9.3	100
2	X	66.9	X	13.5	19.6	100

If the two investigators take the ratios of their daubite and endite percentages they will get the same result.

$$\frac{6.4}{9.3} = \frac{13.5}{19.6} = 0.96$$

Thus, the ratios do not depend on which components a given investigator decided to quantify. Taking the logarithm has a number of advantages, but one of the most obvious is that it will transform any ratio to lie in the interval $[-\infty,\infty]$ which fits with the requirement we discussed in Section 9.3.

So Aitchison's approach is pretty simple and follows 3 basic steps.

1. Transform the compositional data using the log-ratio approach.
2. Analyze the transformed data using standard statistical techniques.
3. Perform the inverse transform to return the result to compositional space.

We can see that the third step requires an inverse transform to convert log-ratio values back into compositions that reside within a unit simplex. As an example, consider log-ratios formed from 3 components, *A*, *B* and *C*, where *C* is the denominator variable. Then to perform the inverse-*alr* (alr^{-1}) and recover the values of *A*, *B*, and *C*:

$$A = \frac{\exp(\log(A/C))}{\exp(\log(A/C)) + \exp(\log(B/C)) + 1}$$

$$B = \frac{\exp(\log(B/C))}{\exp(\log(A/C)) + \exp(\log(B/C)) + 1}$$

$$C = \frac{1}{\exp(\log(A/C)) + \exp(\log(B/C)) + 1}$$

To demonstrate the *alr* we looked at the example:

$$alr\,[80, 15, 5] = \left[\log\left(\frac{80}{5}\right), \log\left(\frac{15}{5}\right)\right] = [2.77, 1.10]$$

The corresponding alr^{-1} would therefore be:

$$\frac{\exp(2.77)}{\exp(2.77) + \exp(1.10) + 1} = 0.80$$

$$\frac{\exp(1.10)}{\exp(2.77) + \exp(1.10) + 1} = 0.15$$

$$\frac{1}{\exp(2.77) + \exp(1.10) + 1} = 0.05$$

We'll now look at a number of different examples to see how Aitchison's method can be applied and how it gives results that are consistent with the behaviour of compositional data.

9.5.1 Finding an average composition

A sample of 23 basalts from the Isle of Skye in Scotland has been characterized according to specimen compositions of Na_2O+K_2O (which we'll call *A*), Fe_2O_3 (which we'll call *F*) and MgO (which we'll call *M*). Because we have 3 component compositions we can plot them in a ternary diagram (Figure 9.8).

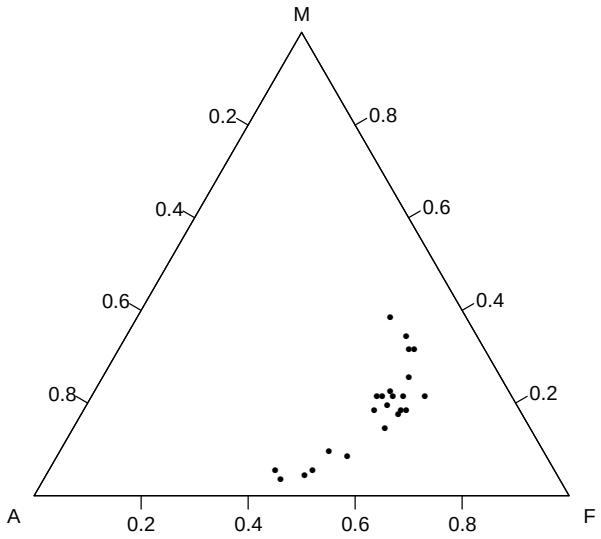


Figure 9.8: *A ternary plot showing the composition of 23 basalt specimens from Scotland.*

It would be tempting to simply calculate the mean of the *A*, *F* and *M* components directly, but as we've seen in the previous sections this would yield a spurious result. Instead we must take the log-ratio of the data, calculate the means of the log-ratios and then transform those means back into *A*, *F* and *M* values (Figure 9.9). Fortunately, R has a package available called compositions, which will help us with this procedure. Once we define a data set as a collection of compositions using the acomp function, R knows that it should always use log-ratio analysis when calculating statistics. The data are stored in the file AFM.Rdata, which contains the data matrix AFM.

Example code: 71

```
> rm(list=ls()) #clear the memory
> dev.off() #close all graphic windows
> install.packages('compositions') >
library(compositions)
> load('AFM.Rdata') #load the data
> AFMc=acomp(AFM) #create a data set which contains compositions
> plot(AFMc,labels=c('A','F','M'),pch=20) # plot the specimens
> ternaryAxis(side=1:3) #add ticks to the axes
> plot(mean(AFMc),pch=15,add=TRUE) #plot the mean as a filled square
```

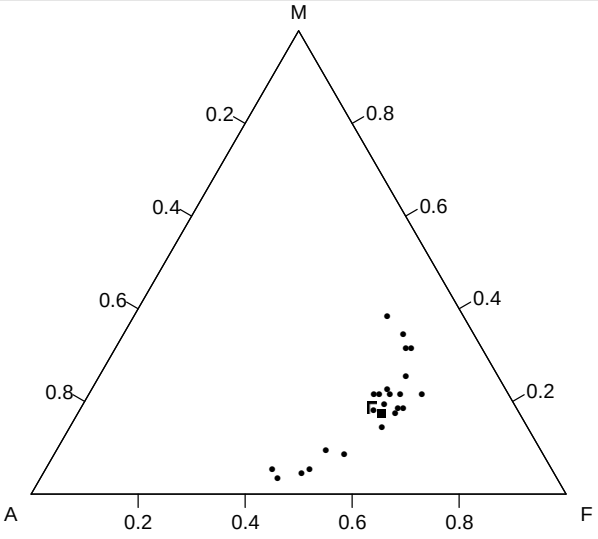


Figure 9.9: *A ternary plot showing the composition of 23 basalt specimens from Scotland and the resultant mean composition (filled square). For comparison the position of the average of the A, F and M components without taking the compositional nature of the data into consideration is shown (open square).*

We can see that the mean composition sits where we would expect, in the center of the curved distribution of data points. For comparison, I also calculated the mean using the incorrect method of just taking the average of each component without using the log-ratio transform. This composition is marked by a open square and it is positioned towards the edge of the data distribution, which is obviously not what we would expect for a mean.

9.5.2 Principal components of compositional data

We studied principal components in detail in Chapter 8. Now we'll try to find the principal components of the *A*, *F* and *M* values of our 23 Scottish basalts. Before we start the compositional analysis let's take a look at what happens if we calculate the principal components without accounting for the fact that we are working with compositions (Figure 9.10).

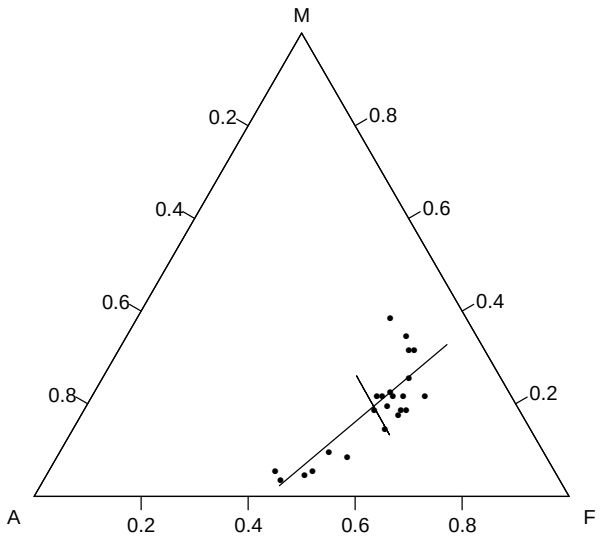


Figure 9.10: *The first two principal components of the basalt data set calculated without applying the log-ratio transform*

We can see from this analysis that as expected the two principal components are at right-angles to each other. The data distribution, however, seems quite curved and the principal components don't do a very good job of describing this curvature. There is an additional problem, if we lengthen the principal components, they will eventually extend outside the limits of the ternary plot, which means that they are not physically realistic for a simplicial sample space.

Calculating the principal components using the log-ratio approach is simple in R, we can add them to a plot with a simple command in the straight function (Figure 9.11). For completeness we'll start from the beginning by loading the data, etc, but if you still have all the basalt data loaded from the previous example then you can go straight into the PCA.

Example code: 72

```
> rm(list=ls()) #clear the memory
> dev.off() #close all graphic windows
> install.packages('compositions') >
library(compositions)
> load('AFM.Rdata') #load the data
> AFMc=acomp(AFM) #create a data set which contains compositions
> plot(AFMc,labels=c('A','F','M'),pch=20)# plot the specimens
> ternaryAxis(side=1:3) # add ticks to the plot
> pc=princomp(AFMc) # perform PCA on the compositions
> straight(mean(AFMc),pc$Loadings,col='black') # plot the components
```

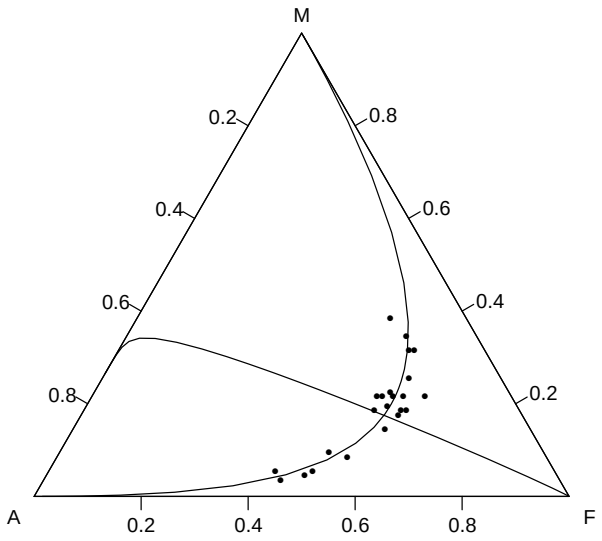


Figure 9.11: *The first two principal components of the basalt data set calculated using the log-ratio transform*

The first thing we notice is that the principal components appear to be curves, however when we think back to Section 9.3.0.2 this is not surprising because we know that straight-lines appear to be curved inside a simplex. This curvature also means that unlike the non-*alr* example above, the principal components will never extend beyond the limits of the ternary plot. The curve in the first principal component allows it to describe the curvature in the data very clearly and we can emphasize this point by looking at the variability explained by each of the components and comparing it to the non-*alr* analysis (Figure 9.12).

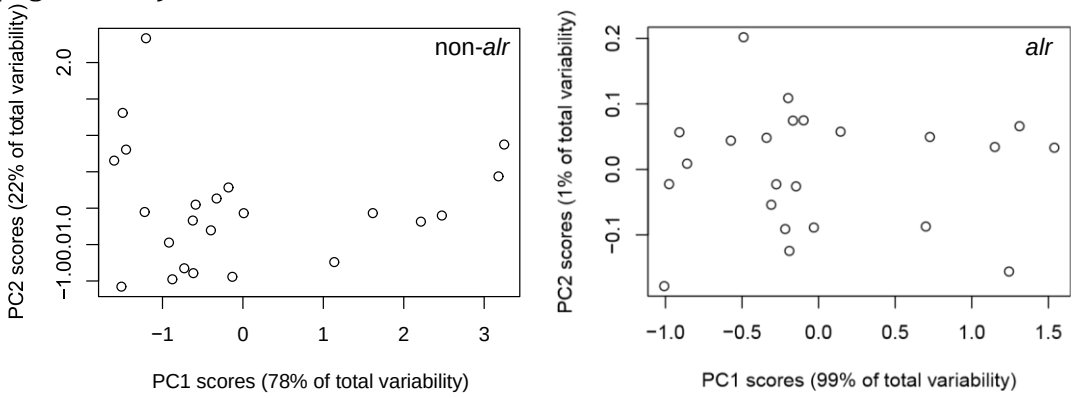


Figure 9.12: *Scores for the first two principal components of the Scottish basalts using the traditional non-alr approach (left) and the alr method (right).*

We can see the scores from the non-*alr* approach describe a slightly curved path because the principal components cannot represent the data properly. In constrast, by applying the *alr* we obtain principal components that provide a very good representation of the data with the first component explaining over 99% of the data variability.

9.5.3 Confidence regions for compositional data

Confidence regions allow us to assess the difference between samples in a quantitative way. Imagine that I return to Skye and collect 1 new basalt specimen. I then quantify the composition of the specimen in terms of its *A*, *F* and *M* content. Finally I want to assess if my new basalt specimen falls into the same compositional distribution as my earlier sample of 23 specimens. There are a number of different methods that can be used to draw confidence intervals around multivariate data sets. These confidence regions then provide significance levels with which we can assess if a point belongs to a distribution or not. Unfortunately such statistics are not designed for compositional data sets, but the log-ratio method will come to our rescue. We can calculate a confidence region for the log-ratio transformed data and then perform the inverse transform to find how the confidence regions are represented in the original compositional space.

Let's look at an example which employs our Scottish basalt data set. To start, we'll reload the data and plot a ternary diagram with a new specimen included (Figure 9.13). The composition of the new specimen is; A=25%, F=45% and M=30%.

Example code: 73

```
> rm(list=ls()) # clear the memory
> dev.off() # close all of the graphics windows
> load('AFM.Rdata') # load the AFM data set
> AFMc=acomp(AFM)            # define the data as compositions
> plot(AFMc,labels=c('A','F','M'),pch=20) # plot the original data
> ternaryAxis(side=1:3) # add ticks to the ternary plot
> Xnew=c(25,45,30) # define a new sample
> Xnew=acomp(Xnew) # define the new sample as a composition
> plot(Xnew,add=TRUE,pch=0) # add the new sample to the plot
```

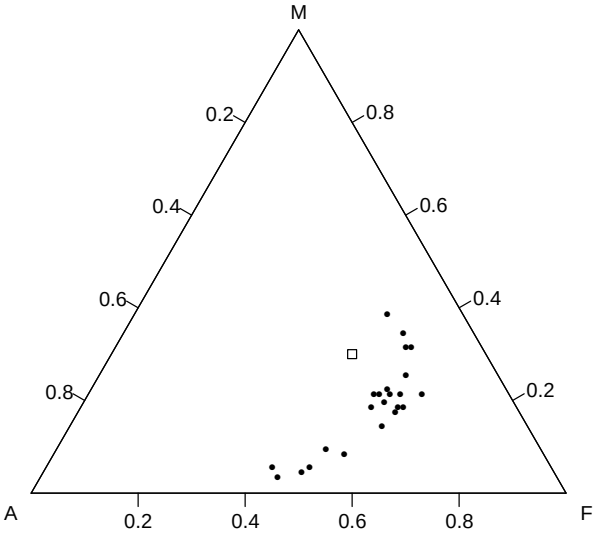


Figure 9.13: *A ternary plot showing the composition of 23 basalt specimens from Scotland (dots). We then collect a new specimen and determine its composition (square). Does the new specimen belong to the same population distribution as our original sample of 23 basalts?*

For this example we'll calculate the log-ratios directly (rather than letting R do it in the background) and plot them in order that we can see the structure of the data after the log-ratio transform (Figure 9.14). To calculate the *alr* we'll use *M* as the denominator in the ratios. This is, however, an arbitrary decision and the results should not be sensitive to which parameter we use as the denominator.

Example code: 74

```
> x1=log(AFM[,1]/AFM[,3]) # first log-ratio using M as the denominator
> x2=log(AFM[,2]/AFM[,3]) # second log-ratio using M as the denominator
> plot(x1,x2,xlab='log(A/M)',ylab='log(F/M)',pch=20) #plot the log ratios
> points(log(25/30),log(45/30),pch=0) # plot the log ratios of the new sample
```

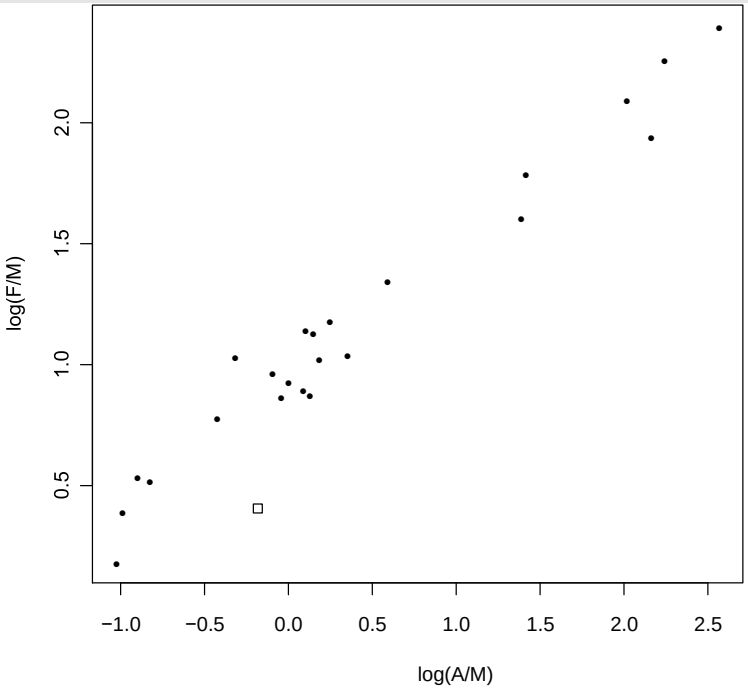


Figure 9.14: *A plot of the 23 basalt specimens from Scotland (dots) and the new specimen (square) in log-ratio space.*

Working with the log-ratios we now need to define a confidence ellipse around the 2D distribution of data points using the R function ellipse. To do this we need to use some statistical quantities such as covariance matrices (calculated with the cov function) that we haven't considered in the earlier chapters. At this stage we are just going to perform the necessary calculations and not worry too much about how the process works. You'll see that the position of the center of the confidence ellipse is simply the mean of each of the log-ratios, which can be found using the function colMeans (which returns the mean of each column in a matrix). We'll then add the confidence ellipses to the plot (Figure 9.15).

Example code: 75

```
> X=cbind(x1,x2) # form a matrix with the 2 log-ratios (1 per column)
> install.packages('ellipse') #download the ellipse package
> library('ellipse') # prepare the package for use
> SD=cov(X) # covariance matrix of X
> e1=ellipse(SD, centre=colMeans(X), level=0.9) # 90% confidence ellipse
> e2=ellipse(SD, centre=colMeans(X), level=0.95) # 95% confidence ellipse
> e3=ellipse(SD, centre=colMeans(X), level=0.99) # 99% confidence ellipse
> lines(e1,type='l', lty=1) #plot the 90% ellipse as a solid line
> lines(e2,type='l', lty=3) #plot the 95% ellipse as a dotted line
> lines(e3,type='l', lty=5) #plot the 99% ellipse as a dashed line
```

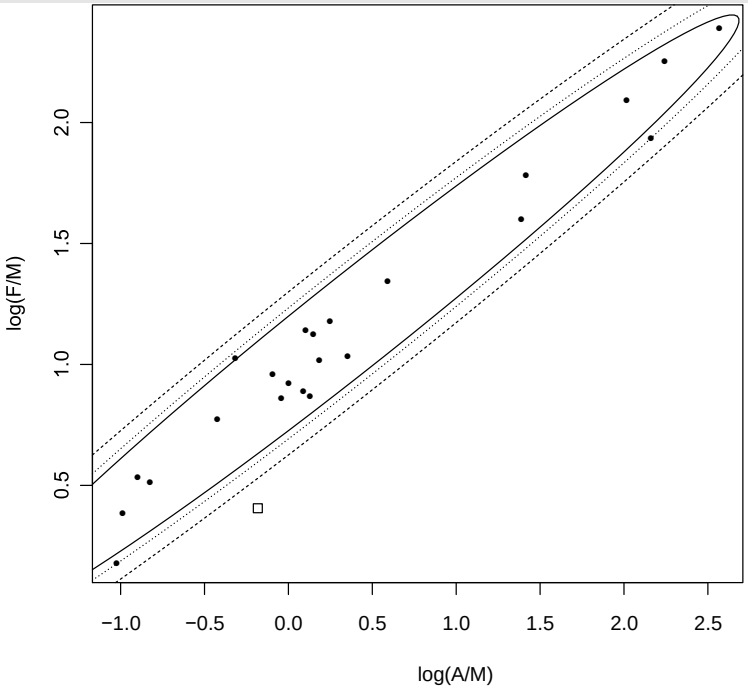


Figure 9.15: *90% (solid line), 95% (dotted line) and 99% (dashed) confidence regions of the true population distribution represented by the original 23 basalt specimens.*

We can see that the new sample is located outside of even the 99% confidence ellipse for the original specimens, so it would appear that it originates from a different population. Of course when plotted simply as log-ratios it is difficult to imagine what the confidence ellipses will look like in the original ternary plot. This is no problem, we can use the inverse *alr* to transform the ellipse information stored in e1, e2 and e3 back into the original *A*, *F* and *M* compositional space. We can then plot all the information together in a ternary plot (Figure 9.16). This processes involves a number of commands in R, so we'll only look at the case of the 99% ellipse stored in e3. If you want to perform the same process for the 90% or 95% ellipses you would need to use the e1 and e2 variables, respectively.

Example code: 76

```
> plot(AFMc,labels=c('A','F','M'),pch=20) # plot the original data
> ternaryAxis(side=1:3) # add ticks to the ternary plot
> plot(Xnew,add=TRUE,pch=0) # add the new sample to the plot
> z1=exp(e3[,1])/(exp(e3[,1])+exp(e3[,2])+1) # inverse-alr to find A
> z2=exp(e3[,2])/(exp(e3[,1])+exp(e3[,2])+1) # inverse-alr to find F
> z3=1/(exp(e3[,1])+exp(e3[,2])+1) # inverse-alr to find M
> lines(acomp(cbind(z1,z2,z3)),type='l',lty=1) # plot 99% confidence ellipse
```

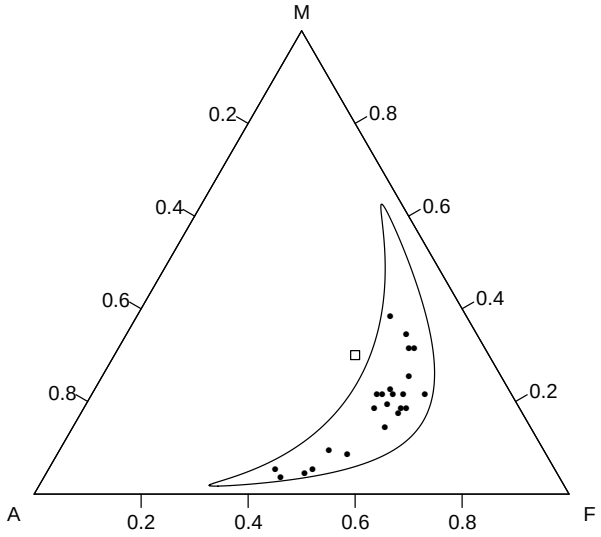


Figure 9.16: *The 99% confidence ellipse (solid) of the basalt specimens (dots) shows that it is highly unlikely that the new specimen (square) comes from the same underlying population.*

9.5.4 Regression and compositional data

In Chapter 5 we looked at regression in some detail, and as you may have guessed, standard regression cannot be applied to compositional data directly. As an example we're going to study sediments collected at 39 different depths in an Arctic lake. The sediments have been classified in terms of their sand, silt and clay contents and we can plot them in a ternary plot. The data are stored in the file ArcticLake.Rdata, which contains a data matrix called gs that has 3 columns corresponding to the *sand*, *silt* and *clay* percentages. Also included in the data file is a variable called depth, which records the water depths at which the sediments were collected. To give a visual impression of any relationship between depth and grain size composition, we will show the data in ternary plot and colour code the points according to their depth (Figure 9.17).

Example code: 77

```
> rm(list=ls()) # clear the memory
```

```
> dev.off() # close all of the graphics windows
> load('ArcticLake.Rdata') # load the grain size data set
> install.packages('plotrix') # install package for color scales
> library(plotrix) # prepare plotrix for use
> cmap=color.scale(depth,c(1,0,0),c(0,0,1)) # color scale based on depth
> plot(acomp(gs),col=cmap,pch=16,labels=c('Sand','Silt','Clay')) # ternary plot
```

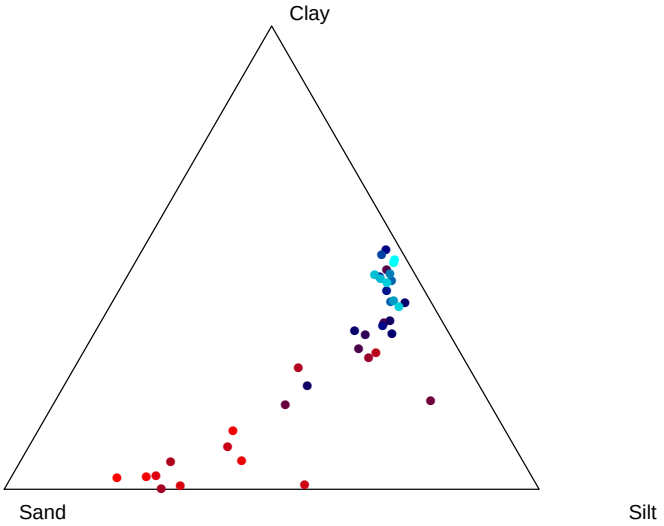


Figure 9.17: *Ternary diagram for the sand, silt and clay compositions of 39 sediments from an Arctic lake. Points are colour coded according to the depth at which they were collected, with red and blue indicating shallow and deep water, respectively.*

The pattern in the ternary plot shows a clear relationship between the grain size composition of the sediments and the depth at which they were collected. At shallower depths the sediments contain more sand and as depth increases the sediments become more fine grained. This is exactly what we would expect from gravity settling with the largest particles being deposited in shallower waters and only the finer particles making it towards the center of the lake where the water is deeper. But can we find the regression relationship between grain size and depth?

We could start our analysis by ignoring the fact we are working with compositions and just blindly apply a regression analysis to the sand, silt and clay percentages as a function of depth. We could then plot the results of the regression analysis in the ternary plot. You can probably see by now that this isn't going to work because we can't just ignore the problems associated with compositional data, but if we did, the regression plot would look like Figure 9.18 (there is no point in showing you how to calculate this regression relationship because it is wrong).

clay

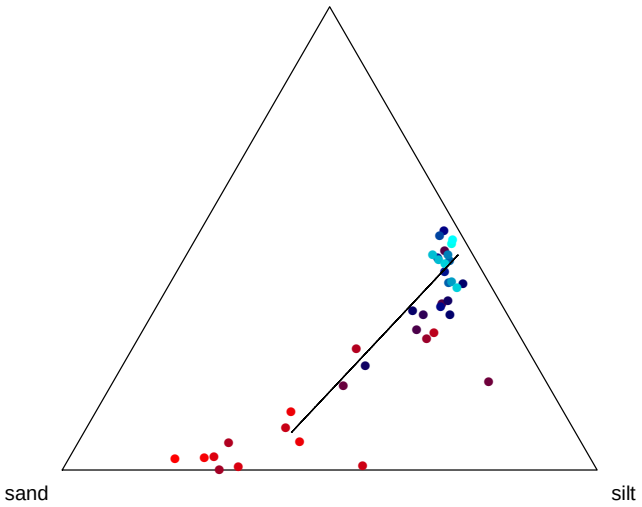


Figure 9.18: *A regression of sediment grain size and collection depth that ignores the fact that the grain size data are compositions.*

We can see that the regression line in Figure 9.18 doesn't do a very good job of explaining the data. The data shows a curved trend, whilst the line is straight and we can see that if we extended the line it would eventually pass outside the boundaries of the diagram. By now, it should be clear what we need to do to solve these problems. We'll apply the *alr* transform to the grainsize data, calculate the regression in log-ratio space and then transform the results back to the original compositional data space. We'll perform the regression using the lm function just as we did in Chapter 5. We'll first look at the regression in log-ratio space (Figure 9.19) and then plot everything in the ternary plot.

Example code: 78

```
> sand=gs[,1] # get the sand data from the matrix
> silt=gs[,2] # get the silt data from the matrix
> clay=gs[,3] # get the clay data from the matrix
> depth_hat=seq(1,200,len=500) # collection of depths for predictions

> lr1=log(sand/clay) # form the first log-ratio
> b_lr1=lm(lr1~log(depth)) #regress the log-ratio and depth

# make predictions for the first log-ratio
> lr1_hat=log(depth_hat)*b_lr1$coefficients[2]+b_lr1$coefficients[1]
> plot(log(depth),lr1,col='red',xlab='log(Depth [m])',ylab='log-ratio')
> lines(log(depth_hat),lr1_hat,col='red') # regression for the 1st log-ratio

> lr2=log(silt/clay) # form the second log-ratio
> b_lr2=lm(lr2~log(depth)) # regress the log-ratio and depth

# make predictions for the second log-ratio
> lr2_hat=log(depth_hat)*b_lr2$coefficients[2]+b_lr2$coefficients[1]
> points(log(depth),lr2,col='green') # plot the second log-ratios
> lines(log(depth_hat),lr2_hat,col='green') # regression for the 2nd log-ratio
```

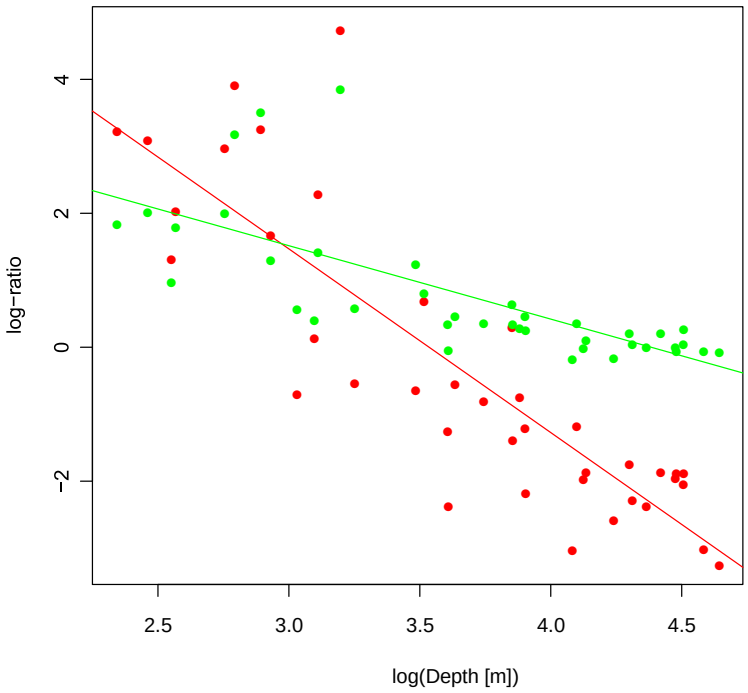


Figure 9.19: *Regression for the two log-ratios against the log of depth. Red shows the regression for log(sand/clay) and green shows log(silt/clay).*

Now we have the regression models that are represented by a collection of points calculated as a function of depth along each of the regression lines in Figure 9.19. The points are stored in the variables lr1 ̲hat and lr2 hat and we simply need to perform the inverse *alr* transform to return the points to the original sand, silt and clay compositional data space. Then we'll display the transformed points in a ternary plot to illustrate the regression relationship (Figure 9.20).

Example code: 79

```
> sand_hat=exp(lr1_hat)/(exp(lr1_hat)+exp(lr2_hat)+1) # find predicted sand
> silt_hat=exp(lr2_hat)/(exp(lr1_hat)+exp(lr2_hat)+1) # find predicted silt
> clay_hat=1/(exp(lr1_hat)+exp(lr2_hat)+1) # find predicted clay
> hat=cbind(sand_hat,silt_hat,clay_hat) #combine predictions into a matrix
> hat_comp=acomp(hat) #define the predictions matrix as compositions
> cmap=color.scale(depth,c(1,0,0),c(0,0,1)) # color scale based on depth
> plot(acomp(gs),col=cmap,pch=16,labels=c('Sand','Silt','Clay')) # plot > lines(hat_comp) #
add the points corresponding to the regression line
```

clay

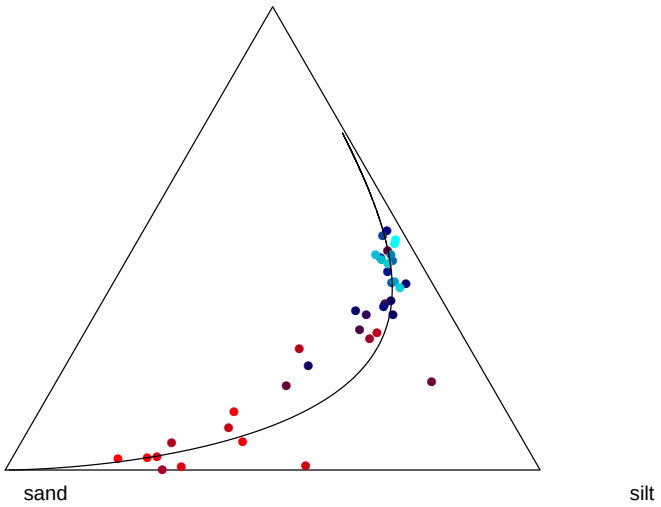


Figure 9.20: *Log-ratio based regression line for the Arctic lake sediment data.*

The calculated regression line captures the curvature of the data and we can see that even if the line was extended it would still provide a physically meaningful model. For example, as the depth becomes shallower the regression line moves towards sediments consisting of only sand and at greater depths the line moves towards sediments composed of only clay.

9.6 Outstanding issues in compositional analysis

In this introduction to the statistical analysis of compositional data we have only considered quite simple cases. Of course things are never as simple as they seem and not all statistical techniques can be applied to compositional data even if we do employ the *alr* transform. Determining the extent of the correlation that exists between parts of a compositional data set is still an open question. As we saw in Section 9.4.2, subcompositional incoherence means that traditional correlations between two parts of a composition are meaningless and unfortunately the use of the *alr* transform does not solve this problem.

A family of different log-ratio transforms now exists and some statistical techniques require you to use a specific transform, whilst others are insensitive to the type of transform you employ. This means you will always need to do some reading to check that you are using the correct approach for the questions you are trying to address. A list of good background texts is provided in the recommended reading.

9.6.1 Dealing with zeros

Finally, one major issue when dealing with compositional data in the geosciences is the presence of zeros in a data set. For example we might have counted foraminifer abundance in a collection of sediments and a certain species may be absent in one of the assemblages. It would therefore be represented in the foraminifer abundances as 0%. However, when we apply the *alr* we then need to take the logarithm of 0, which is not defined.

Various strategies have been suggested to deal with the presence of zeros in compositional data, most of which focus on replacing them with very small values. This is still a problem for geoscience data sets that may contain large numbers of zeros and as yet a satisfactory way to deal with such data has yet to be developed.

Maybe there is light at the end of the tunnel, in 2011 Michael Greenarce from the Universitat Pompeu Fabra developed a method for dealing with zeros by relaxing the subcompositional coherence requirement. He found that you could deal with zeros as long as you were willing to let your analysis be very slightly subcompositionally incoherent. We will have to wait and see how successful this approach is.

9.6.2 Final thoughts

The important take home message from this chapter demonstrates a fundamental point in the statistical analysis of data. It is very easy to do your analysis incorrectly if you are not familiar with the techniques you are using. If you feed compositional data into a piece of statistics software it will quite happily provide you with spurious results based on the assumption of a Euclidean sample space. For example, we saw that you can't calculate a correlation for two parts of a compositional data set, but there is nothing to stop you putting the numbers into Excel and producing an r^2, which is in fact meaningless. Given the amount of time you will have spent collecting your data it is worth investing some time to make sure that you are doing the statistics properly. There is no point in spending months or years collecting data and then wasting all that effort by performing a statistical analysis incorrectly and drawing spurious conclusions.

10 Recommended reading

10.1 Light reading

Best, J. (2004). *More Damned Lies and Statistics, How Numbers Confuse Public Issues*. University of California Press.

Salsburg, D. (2001). *The Lady Tasting Tea: How Statistics Revolutionized Science in the Twentieth Century*. Holt.

10.2 General statistics for geoscientists

Borradaile, G.J. (2003). *Statistics of Earth Science Data*. Springer.

Davis, J.C. (2002). *Statistics and Data Analysis in Geology*. Wiley.

Swan, A.R.H. & M. Sandilands (1995). *Introduction to Geological Data Analysis*. Blackwell Science.

10.3 Statistics with computers

Marques de Sa´, J.P. (2007). *Applied Statistics Using SPSS, STATISTICA, MATLAB and R*. Springer.

Middleton, G.V. (2000). *Data Analysis in the Earth Sciences using MATLAB*. Prentice Hall.

Trauth, M.J. (2010). *MATLAB Recipes for Earth Sciences*. Springer.

10.4 More advanced texts

Aitchison, J. (2003). *The Statistical Analysis of Compositional Data*. The Blackburn Press.

Buccianti, A., G. Mateu-Figueras & V. Pawlowsky-Glahn (2006). *Compositional Data Analysis in the Geosciences: From Theory to Practice*. Geological Society of London.

Pawlowsky-Glahn, V. & A. Buccianti (2011). *Compositional Data Analysis: Theory and Applications*. Wiley.

Warton D.I., I.J Wright, D.S. Falster & M. Westoby (2006). *Bivariate line-fitting methods for allometry*. Biological Reviews 81(2), 259-291.