Unit 1: From Data to Conclusions

Mhairi McNeill and Robert Reside

17 June 2016

# What is data?

What isn't data? Any piece of information about the world is potentially data. The prices on a menu is data; the gender of your 5 closest friends is data; a book is data; a picture of your cat is data; the goals scored by your favourite team is data; a conversation is data.

Now, clearly, some of these sources of data are more useful than others. Data becomes useful when we can use it make conclusions about the world. To be useful data needs to fulfil two criteria. First, we need to be able to analyse it. For that reason we will be focusing numeric data (menu prices, number of goals) and category data (gender of friends). These are the types of data we can analyse using statistics.

The second criteria for usefulness is the quality of data. I have left quality intentionally vague here because there is a lot ways data can be low quality, ways which I am sure you have an intuitive understanding of. I want to know if a neighbourhood has good restaurants. Compare the two following sources of data: firstly, I know didn't like the last restaurant I went to there. Secondly I have a dataset of 20,000 detailed online restaurant reviews from the last 5 years written by 1,000s of people. Which data source do you think is better quality for answering my question? The second data has far more information, about far more restaurants and from more people than just me.

Don't confuse quantity of data for quality though. Think of the following headline: 'Tory economic policies deeply unpopular says survey of 10,000 Labour voters'. How useful is that survey for understanding general public opinion? Wouldn't a survey of 500 people chosen completely at random be better?

Be careful with data quality, and match the strength of your conclusions to the strength of your data. We very rarely have data that is perfect for the question we want to answer. For example, the online restaurant review data, while probably better than my single opinion, only comes from people who have actually taken the time to review the restaurant. This probably represents the strongest opinions of all restaurant goers - you'd be much more likely to leave a review if you really loved the restaurant or really hated it. Also, people who regularly leave online reviews are probably younger on average, and probably eat out more often than average. Don't give in if you don't have perfect data (if the 20,000 reviews are all negative then the neighbourhood probably does have remarkably terrible restaurant). But for good answers to your questions quality data is far more important than any of the techniques we are about to discuss below. Always keep quality in mind.

## Types of Data

Earlier I said that we’d be looking at numeric data and categorical data. Numeric data is anything that can be measured by a number. That number can be a whole number. For example: how old is this child? Or it can be a decimal number. For example: what is the height of this child?

Categorical data describes the category something is in. Each child is in a year at school, and has a gender. The year a child is in is a special kind of categorical data known as ordinal data. This is because the categories have a natural ordering: year 1 is less than year 2 which is less than year 3 etc. While gender has no natural ordering; this is sometimes called nominal data.

The type of data matters because it effects what type of analysis will work best on the data. Some data has an ambiguous type. You could argue that the year a child is in is also numeric data (1, 2, 3 etc.). That's fine: it just means more than one type of analysis will work on that data.

You'll notice that we have lots of bits of information we can measure about each child. We'll be working with this type of data a lot. In this example where we have surveyed school children each child makes up an *observation*; each measurement (age, height, year, gender) is an *variable*. Thinking about observations and variables is a very useful way of looking at data, which we will see more of later.

### Exercises

1. What data do you imagine capturing in your future? This could be for work or for a hobby. Give three examples of numeric data and three examples of categorical data.
2. Of the categorical data which was ordinal and which was nominal?
3. What might be the potential quality issues with:
4. Doing medical trials to test the efficacy of your drug.
5. Evaluating your personality by asking all your friends.
6. A internet survey to find out current attitudes to gay marriage. How would the quality issues change if you paid people to fill out the survey?
7. Bonus Question: In the 2015 UK General Election the polls were predicting a vote much closer than the final result. What do you think might be potential quality issues with doing election polls? Can you find out why the poll-makers think they got the result wrong?

### Answers

1. Some examples:

* Customer service
  + Numeric: number of customers, satisfaction scores, number of complaints.
  + Categorical: complaint types (nominal), dispute resolved - yes or no (ordinal potentially - yes > no), gender of customers (nominal)
* A runner
  + Numeric: heartrate, miles run, distance run
  + Categorical: surface type (nominal), month ran in (ordinal), gear worn (nominal).
* Social media
  + Numeric: Monthly visits, bounce rate, number of engagements
  + Categorical: browser type (nominal), keywords used (nominal), type of engagement (possibly ordinal - is a share > a like?)

1. This is a huge question. There's so many answers, here's some examples of good answers.
   * Placebo effect. We tend to find any intervention effective (need to have a control group treated with nothing).
   * Are the patents in your trial the population as will be actually using your drug.
   * Can you find very rare adverse effects with your sample size?
   * How are you measuring effectiveness. If it's subjective will the people measuring this be biased and want the drug to succeed. (This is normally dealt with by doing double blind trials - where neither the patients of the doctors know which drugs are real and which are placebo.)
   * Biased sample. Your friends probably like you or they wouldn't be your friend.
   * They won't want to tell you how they actually feel because it would hurt your feelings.
   * Internet surveys tend to be quite biased in the type of people who respond. Only a certain type of person willingly fills in surveys.
   * Often with this type of survey you'll get a very keen group of people who care about the results coming out one way or the other and then try to encourage everyone they know who thinks like them to answer the survey.
   * Paying money will probably get a wider range of people to take part (still probably not a representative sample of people). However, people become more likely to just put in random answers to get their money, without thinking about the questions.
2. Poll makers spend a lot of time trying to get a representative sample of the population to answer their questions. But it is difficult. In the last election they think that they under-sampled people who were at work during the time the phone polls and door to door polls were done.

# Working with R

To analyse numeric and categorical data we are going to use a programming language called R. You might have used a point-and-click interface for analysing data before like Excel, SPSS or Minitab. Using R is somewhat similar to these programmes but also somewhat different. The biggest difference is that you'll be typing commands rather than clicking on boxes and menus. At first this is more difficult but will almost certainly become quicker and easier in the long run.

There's a bit of debate about what software you should use to analyse data. Every way of doing it has advantages and disadvantages. We'll be using R in this course because:

1. When you type everything you have a written record of everything you've done to a piece of data. This makes it easier to check your logic and the preserve the original data.
2. You will often need to do the same piece of analysis over and over on different datasets. Having the code to do the analysis written out makes it easy to repeat.
3. R is free and open source.
4. R has a lot of excellent libraries (downloadable extensions to R) which make generating plots, doing specialised techniques, making reports and even producing interactive dashboards easy.

If you've learnt to programme before, that's great. I'm sure you will find learning R straightforward. There are a few idiosyncratic features of R that can trip up experienced programmers. You might find this guide useful: <http://www.johndcook.com/blog/r_language_for_programmers/>

If you've never done programming before that is also great. R is a good language for a beginner and once you've learnt R it will be much easier to pick up other languages, which opens up a huge range of things you can do with computers!

### Installing R

We are going to install R the programming language. R is programme that actually does the analysis - you interact with this programme by typing R commands. For example, if you type 2 + 2, R will do the maths and return the result to you.

Then we are going to install an IDE (interactive development environment) called RStudio. RStudio is a very popular way of interacting with R. While you don't need it to write R code RStudio makes it easy to do things like: 1. Write longer pieces of code and get R to run it piece by piece. 2. Keep track of what data R knows about. 3. Organise all your R code files. 4. See plots and tables you've created in R. 5. Make special types of R files e.g. reports, dashboards etc. 6. Access R's in-built help 7. Manage which R packages (extensions to R) you have installed.

R is free and open source. It is written by volunteers and all the all packages you'll use were also written by volunteers.

RStudio is also free and open source, but is made by a profit making company. They make their money by selling a professional version of RStudio that runs on a sever and has support.

[Videos here]

### Getting started.

Once you have RStudio installed and opened you should see four panels. We'll ignore the two right-hand-side panels for just now.

The bottom-left panel is the R interpreter. We can type R commands into the interpreter and R will do some calculations and return an answer. Let's start with one of the most basic commands possible. Let's get R to add 2 and 2. Type 2 + 2 into the interpreter and press enter.

You should see something like this:

2 + 2

## [1] 4

After you hit enter R understood the command and found the answer, returning it almost instantly to you. (The little [1] in front of the answer just means that there's only one element to the answer).

Now, type the same thing in the top left panel. When you press enter here nothing will happen. The top left panel is basically just a very simple text editor like Notepad on windows or TextEdit on a Mac. You can absolutely write R code in a separate text editor and many people do. A big advantage of writing code inside RStudio is that it's very easy to transfer code from the editor to the R interpreter. Just move your cursor to the line with 2 + 2 and press ctrl and enter at the same time. Pressing cmd+enter also works on a Mac. The code you have written will now appear in the interpreter along with the answer.

If you have a longer piece of code that goes across multiple lines you will need to highlight the lines and then press ctrl+enter. Try this just now after typing this in the top left editor.

1 + 2 +  
4

## [1] 7

You can save the code you have written in the text editor and come back to it at any time. Just go to **File** and then **Save As…**. When going through this unit keep everything you’ve written in the text editor saved.

### Vectors

A very important thing to understand about R is that it works with *vectors*. Vectors are a list of items. Often this will be a list of numbers, but we will also work with vectors of strings (in programming languages anything made up of letters is called a string) and vectors made up of True and False values. In R we call a vector of strings a character vector. A vector made up of only True and False is a logical vector.

There are other types of vector in R, but for just now we will only be working with numeric, character and logical vectors. To make a vector in R we surround the elements with c( and ). For example

c(1, 4, 6, 3)

## [1] 1 4 6 3

Below is an example of the three vectors types we will learn about.

Numeric | c(2.2, 0.1, 5) Character | c('apple', 'pear', 'e') Logical | c(TRUE, FALSE, FALSE)

### Exercise

1. Create a numeric vector containing this data: 10, 11, 11, 14. (Type in the text editor and use ctrl+enter to send to the interpreter)
2. Create a character vector containing this data: 'small', 'small', 'small', 'large'.

### Assignment

Creating a vector isn't very useful if we can't store that vector somewhere. In a programming language if we want to store some data, or the results of a calculation we use assignment. Assignment is basically giving a name to a piece of data. If we want to use that data we can just find it by using its name, rather than typing out the data again.

In R we can assign using an arrow. Here’s an example of assigning the name x to a vector:

x <- c(1, 4, 6, 3)

Now we've assigned a vector to x we can see what x is at any point by just typing x into the interpreter.

x

## [1] 1 4 6 3

Here we have been giving names to vectors but we can give a name to any type of R object. Here's how we would assign a name to a number:

days <- 5

We can also give names to more complicated objects which will will see later.

Names in R can only contain letters, numbers, '.' and '\_'. They can't start with a number or an underline. Try to make names descriptive, so that when you go back to a piece of analysis you should be able to understand what everything is. Since you can't have spaces in names you can use the underscore character as a space:

days\_until\_launch <- 5

### Exercise

1. Give the numeric vector we created earlier the name height.
2. Give the character vector the name height\_category.

### The difference between 'apple' and apple

A very common point of confusion for new programmers is the difference between a string like 'apple' and a name like apple.

First first 'apple' with quotes is simply a piece of data. If we type this into the interpreter R sees a new string we have created.

'apple'

## [1] "apple"

The second is a name given to an object. Since we haven’t called anything apple yet R doesn't know anything about apple.

apple

## Error in eval(expr, envir, enclos): object 'apple' not found

Another way of putting it is that 'apple' is something that gets given a name:

fruit <- 'apple'

And apple is a name that you can give to data:

apple <- TRUE

Now you've understood the basics of R, we can begin analysing some data. Starting with some very basic, small data we'll discover ways the data can visualised and summarised which will help us understand it better and answer questions we have about the data.

# Visualising Data

## Bar plots

Here's an example of a piece of data. You asked 15 people how much they liked spinach on a scale from -3 to 3. Where 3 means they love spinach, 0 means they feel completely neutral on spinach and -3 means they hate spinach. Here are the results:

0, 1, -3, 0, -2, 0, -3, 1, 0, -3, -3, 3, 3, 0, 2

Create a vector in R containing the data above. Assign the name spinach\_rating to it.

Now, if I want to understand what the general opinion is of spinach I can simply read though the list of scores given and get a general impression. I can see quite a few 0s and -3s. It's possible to understand data just by reading it when we have a small dataset like this. However, imagine we had 100 responses or 1000 responses. We need to do something to understand what our data is actually saying.

The first thing to do is the make a summary table of the data. How many people gave each of the available scores. R makes this very easy:

table(spinach\_rating)

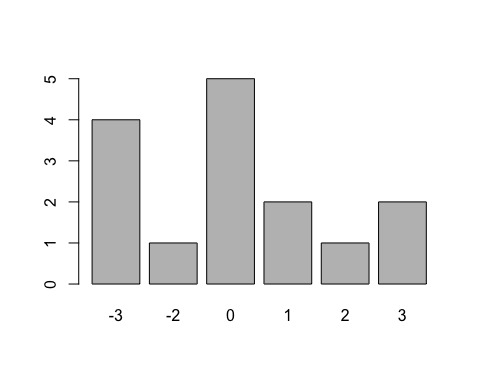
## spinach\_rating  
## -3 -2 0 1 2 3   
## 4 1 5 2 1 2

We can see that the most popular score to give was a zero, with a minus three the second most popular. We can do even better than that though. A plot will make the results even more clear. Assign the table you have just created to a variable called spinach\_table.

spinach\_table <- table(spinach\_rating)

And then use the barplot to create a plot.

barplot(spinach\_table)



It's very easy to go from a raw list of numbers to a picture that gives us a very clear overview of the numbers. We can now make some conclusions about the the data: of our 15 respondents most people were neutral on spinach or hated it, although 4 out of the 15 gave spinach an overall positive rating.

## Functions

In the code above we used two functions from R: table and barplot. A function is anything that takes an input and returns an output.

The table function took our vector of data as an input and transformed it into an table.

The barplot function took in a table and transformed it into plot.

We'll be using a lot of functions, and later we'll even be writing our own functions. They will all follow the same basic pattern of taking in an input and returning an output. We also always write them like this, with the input going in brackets after the name of the function:

some\_function(some\_input)

The output is normally a new R object, like the table we created above and assigned to spinach\_table. But we can also have other types of output - like the plot we saw.

## Histograms

You also asked the same 15 people how tall they were. Here are the results, all in centimetres:

133, 110, 224, 134, 135, 136, 125, 137, 104, 132, 114, 130, 129, 237, 131

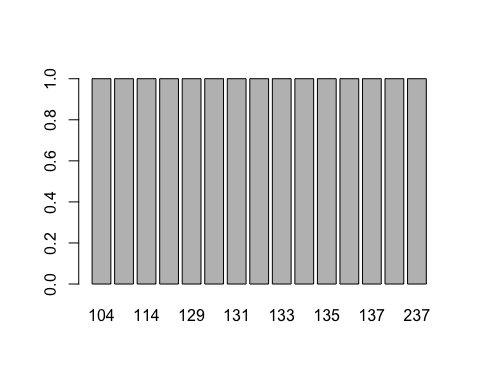
Now, just looking at those raw numbers it's a little tricky to say much about how tall the people we surveyed are.

Let's try the technique we learnt above and translate those numbers into a table and a plot:

height <- c(133, 110, 224, 134, 135, 136, 125, 137, 104, 132, 114, 130,   
129, 237, 131)  
height\_table <- table(height)  
  
height\_table

## height  
## 104 110 114 125 129 130 131 132 133 134 135 136 137 224 237   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

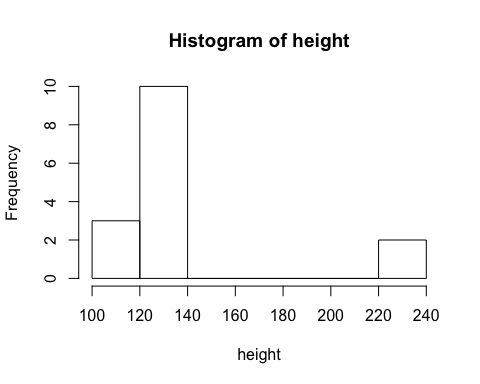
barplot(height\_table)



That was not that helpful. Since every person in our sample had a different height each count in our table as just one. We need a different way of summarising this data.

Let's start with a plot for this data. We're going to use a histogram, a type of visualisation that looks very similar to a barplot except that each bar counts a range of values, instead of just one single value. Here's how we do this in R:

hist(height)

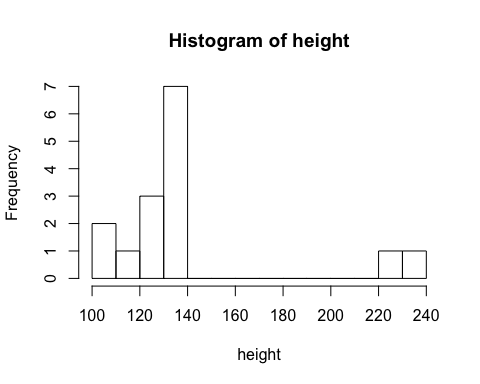


Now we can see that most of the heights were between 100cm and 140cm, with a couple of very tall people who were over 220cm. Perhaps most of the people you surveyed were children. Note that the bars in the histogram are directly next to each other - while the bars in a barplot have spaces between them. That is how you can tell at a glance which type of plot you are looking at.

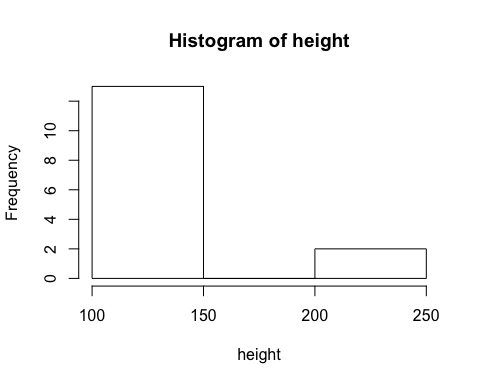
With histograms, the size of the groups can change the shape of plot significantly. If we have more groups we can get a more accurate picture of the data, but too many groups then the plot becomes meaningless - like the barplot of heights we created earlier.

We can set the number of groups in our histogram by changing the breaks argument in the function. Compare the two plots below to the one before.

hist(height, breaks = 10)



hist(height, breaks = 3)



This is why it is important to try a range of plot sizes; try to use histograms which describe the data well, and don't hide important features.

What if you want to summarise the height data into a table? You can use the cut function. This function turns numeric data into a set of groups, which we can then use table to summarise.

height\_groups <- cut(height, 3)  
table(height\_groups)

## height\_groups  
## (104,148] (148,193] (193,237]   
## 13 0 2

# Describing data

Let's go back to our spinach opinion data. You will often find that you want just one number which can describe the dataset overall; we call that number an average. This lets us know what the typical respondent thinks of spinach. The most common type of average is a mean, which is very easy to calculate in R:

mean(spinach\_rating)

## [1] -0.2666667

So, on average our survey respondents gave spinach a rating of -0.3 i.e. they were very slightly negative on spinach overall. Note, that I rounded the answer given by R. It's rarely helpful to know the exact value given by R; it’s just hard for your audience to read. Also, giving an answer to 10 decimal places implies that you can measure spinach opinion to 10 decimal places, which of course you can't.

### The maths

The maths behind how R does this is very simple. To calculate a mean you just need to add up all the values in your dataset and divide by the number of values you just added up. Written down mathematically:

Here is all the values in your dataset and , means take the sum of them all. is a symbol we use for the mean of values and is just the size of the dataset.

We can see that this gives results that make sense. What do you think the typical value in these three numbers would be: 3, 4, 5? I think 4 would be a reasonable answer, it's in the middle. Let's check what the mean is:

### Exercise

What do you think the average should be for these values? Calculate the mean by hand and compare:

1. 2, 2, 2
2. 4, 6
3. 2, 2, 5

## Median

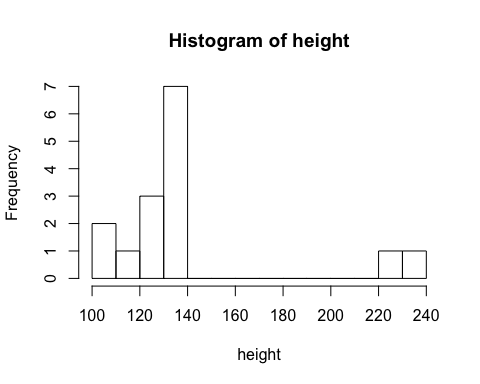
Let's go back to the heights dataset. What's the mean height of our respondents?

mean(height)

## [1] 140.7333

On average our respondents were about 140cm. Fair enough, right? Let's have a look at our histogram again

hist(height, breaks = 10)



To me that doesn't seem quite right, because most of the respondents were less 140cm tall. The mean is getting pulled upwards by those two very tall people.

This type of data is known as skewed data. Data is skewed when a small about of it is much higher or lower than the general trend. When you have skewed data it is often better to use a different measure of average known as the median.

median(height)

## [1] 132

I would argue that 132cm is a fairer measure of of the typical height for these respondents. When you have skewed data the mean and the median will often be very different. It's often a question of judgement which is the better measure. This is why it is important to see plots of your data; they will allow you to understand the true shape of the data and summarise it fairly.

### Exercise

A company has three regular employees and a CEO. The CEO gets paid £120,000 per year. The three other employees all make £24,000 per year.

1. What is the mean salary of all workers at this company? What is the median salary?
2. Which average salary would it be fair to put on an advert for a new employee?

### The maths

How does R calculate a median? This is also very simple. The median is just the middle value when we write out all the values in order. So, to find the median for 2, 4, 6, 2, 5, first write the values out in order:

2, 2, 4, 5, 6

And find the middle value.

2, 2, **4**, 5, 6

So here the median would be 4.

If the middle values is between two numbers, just take the mean of the numbers on either side. So, for this example:

2, 2, **4**, **5**, 5, 6

The median is ${4 + 5 \over 2} = 4.5$

### Exercise

Calculate the median, by hand, for the three sets of numbers below:

1. 2, 2, 2
2. 4, 6
3. 2, 2, 5

Which sets of numbers have the same mean and median?

## Standard deviation

I'm trying to decide what film to watch tonight. I've got two options and they've both been rated 6 times.

First film: 2 stars, 3 stars, 3 stars, 3 stars, 3 stars, 4 stars

Second film: 1 star, 1 star, 1 star, 5 stars, 5 stars, 5 stars

The mean rating for both films is 3 stars. But, there's clearly something different about these two films. The first film has 6 people agreeing that it is okay. The second film is very divisive. Half the reviewers hated it and half the reviewers loved it.

There's more to a dataset than just the mean; we need some type of statistic that captures the amount of agreement in a dataset. We often call the amount of agreement/disagreement the variability of a dataset. And like the average there's a few options for how to measure this. Your first option is the standard deviation. Like all statistics we have seen so far, R makes this very easy to calculate.

film\_1 <- c(2, 3, 3, 3, 3, 4)  
film\_2 <- c(1, 1, 1, 5, 5, 5)  
  
sd(film\_1)

## [1] 0.6324555

sd(film\_2)

## [1] 2.19089

But, like the mean, the standard deviation doesn't work so well on skewed datasets. We have the equivalent of the median for measuring variability and it is known as the interquartile range. Below is how to use it in R:

IQR(film\_1)

## [1] 0

IQR(film\_2)

## [1] 4

### The maths

The standard deviation is basically the average difference from the mean, across all values in your dataset. Since we don't care if the difference from the mean is positive or negative, we square all the differences from the mean. To counteract every value being squared we then take the square root of the final answer.

We don't need to take the final square root. If we leave it out, we have another measure of variability called the variance. This is also very easy to calculate in R.

var(film\_1)

## [1] 0.4

You will often see the standard deviation and variance formulas calculated with on the bottom of the fraction. Indeed, that is the formula R is actually using to calculate the standard deviation and the variance. We will discuss why this is later in the section on samples and populations.

The interquartile range is calculated in a similar way to the median. We put all our values in order:

1, 1, 1, 5, 5, 5

First we find the median:

1, 1, **1**, **5**, 5, 5

Here it's three. Now, for each half of the data on each side of the median, we find another median:

1, **1**, 1, 5, **5**, 5

The difference between those two medians (1 and 5) is the interquartile range (4).

### Exercise

A. Look at the following 4 ways of describing a dataset. What are the advantages and disadvantages of each?

1. Raw data.
2. A frequency table.
3. Descriptive summary.
4. A plot.
5. Going back to our spinach opinion and height data, we also asked for opinions on chocolate. We used the same scale where -3 hate, 0 is neutral and 3 is love. Here are the raw data:

3, 3, 0, 3, -3, 3, 0, 2, 2, 2, 3, 3, 2, 3, 1

Create a frequency table, a descriptive summary and a plot for this data. Write a couple of sentences about your conclusions on chocolate opinion.

### Answers

A. 1. No information lost. Not very easy to understand.

1. Again, no information lost. Can still be difficult to understand - particularly if very few of the values are the same.
2. Very compact. Answers questions we have about the data. Important information about the shape and spread of the data is lost.
3. Easy to understand overview of the data. Can often show us the shape of the data well. But information may be lost and it can be difficult to get raw numbers out.

# Dataframes in R

Throughout this unit we have been using the spinach survey data very frequently. It might make sense to keep all this data together in one R object. After all, the spinach rating, and the heights and the chocolate ratings all come from the same people. So it probably makes sense to have this data linked together somehow.

In R we can link several vectors together by using a *dataframe*. This is very useful when we have data in the form of multiple variables of certain observations. In our spinach data each observation is a survey respondent, and each variable is some fact about them.

To link data together we use the function data.frame. This takes in multiple vectors of data and converts them into a special data frame object:

height <- c(133, 110, 224, 134, 135, 136, 125, 137, 104, 132, 114, 130, 129, 237, 131)  
spinach\_rating <- c(0, 1, -3, 0, -2, 0, -3, 1, 0, -3, -3, 3, 3, 0, 2)  
chocolate\_rating <- c(3, 3, 0, 3, -3, 3, 0, 2, 2, 2, 3, 3, 2, 3, 1)  
  
data.frame(height, spinach\_rating, chocolate\_rating)

## height spinach\_rating chocolate\_rating  
## 1 133 0 3  
## 2 110 1 3  
## 3 224 -3 0  
## 4 134 0 3  
## 5 135 -2 -3  
## 6 136 0 3  
## 7 125 -3 0  
## 8 137 1 2  
## 9 104 0 2  
## 10 132 -3 2  
## 11 114 -3 3  
## 12 130 3 3  
## 13 129 3 2  
## 14 237 0 3  
## 15 131 2 1

Let's give this object a name, so that we can easily access it later. It is a good idea to give dataframes descriptive names like rating\_and\_height\_data or spinach\_data. However, if you are only going to be working with one dataframe it's fine to call it df (short for data-frame) because it saves typing.

df <- data.frame(height, spinach\_rating, chocolate\_rating)

All the vectors being linked together must be the same length. Dataframes are primary used for linking observation/variable data together. In this situation it doesn't make sense to have a different number of observations for different variables.

## Working with dataframes

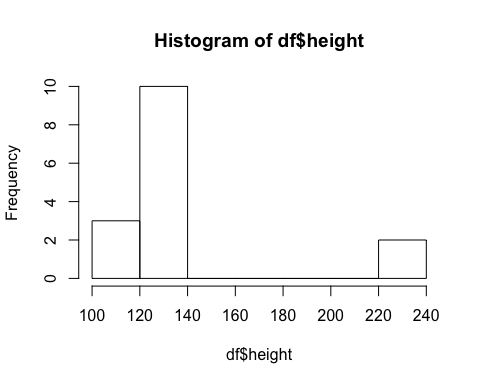
If we want to access the individual vectors in a dataframe we need to tell R where that vector is (it's inside a dataframe). We do this using the $ operator.

df$height

## [1] 133 110 224 134 135 136 125 137 104 132 114 130 129 237 131

df$heigth means the vector named height inside the dataframe df. Every time you use the height vector you will need to use df$height. For example:

hist(df$height)



Keeping objects in dataframes and accessing them using $ does seem awkward at first. However, it is often useful in the long run for three reasons:

1. You will often be working with multiple datasets at the same time it can get confusing working out which dataset each variable is associated with.
2. Later we be learning about some R libraries (dplyr and ggplot2) which are designed to work well with dataframes. You really need to have your data in dataframes to use these libraries effectively and in turn they make working with dataframes easier.
3. Most data gets read into R in the form of a dataframe.

If we want to add an extra vector to our dataframe we can just assign to a name in that dataframe directly.

df$age <- c("Adult", "Child", "Adult", "Adult", "Child", "Adult", "Child",   
 "Adult", "Child", "Child", "Child", "Adult", "Adult", "Adult",   
 "Adult")

Deleting a vector from a data frame can just be done by setting the vector to null.

df$age <- NULL

A very useful function in R is str. It gives you a string representation of what is in an object. Using it on a dataframe will give you a compact description of all the vectors in that dataframe.

str(df)

## 'data.frame': 15 obs. of 3 variables:  
## $ height : num 133 110 224 134 135 136 125 137 104 132 ...  
## $ spinach\_rating : num 0 1 -3 0 -2 0 -3 1 0 -3 ...  
## $ chocolate\_rating: num 3 3 0 3 -3 3 0 2 2 2 ...

The 'Environment' panel on the top right in RStudio will also give you the descriptions from str if you click on the triangle next to a dataframe. You can also view the entire dataframe by using View (or by clicking on the name of a dataframe in the Environment panel). This is useful if you are familiar with working in Excel, and like to be able to see the data you are working with.

View(df)

Unlike Excel you cannot edit a file from the view mode - although this is a good thing because you can't accidently mess up for data!

## Reading external data

Almost all data you work with in R will be read in from an external file (typing out all your data into vectors get difficult very quickly!). We will now read in an extended version of our spinach data. We will be using this data throughout the rest of the unit.

The file is in the form of a CSV (comma separated value file). This is the most common form of text data, and is a very good choice saving data. If you open up the data in a text editor you will be able to see that each variable is separated by a comma and each new observation goes on a new row. This form of data is easy for lots of different programs to use so makes it a very good choice when you need to store data.

To read in a CSV file just use the R function read.csv. You will need to find where your CSV file is stored on your computer and copy the file path (don't worry if you are struggling to find the file path, we will cover an easier way soon).

df <- read.csv('C://path//to//data//spinich\_data.csv')

Having to type out the full path to your data gets tedious very quickly. It also makes it more difficult to share your work, because other people will have a different path to their data, so will need to manually edit all file paths.

A good solution is to use projects inside RStudio. For each project RStudio creates a little file that tells RStudio that all the code and data we need is contained in one folder. Creating one is easy. Make sure you have the code and the csv file in the same folder in your computer. I'd recommend making a new folder with just those two files. Then in RStudio go to 'File', then 'New Project...'. Chose the make a project in an existing directory and chose the location of the directory with your code and spinach data file.

Now when you want to read in the data RStudio knows that the data is in the same place as your code and this will work:

df <- read.csv('spinach\_data.csv')

Also, the 'Files' window in the bottom right in RStudio will now show you all the files you have in your project's folder.

# Data subsets

## Numeric subsets

Often when analysing data we want to find and use certain elements of a vector. Let's have a look at our spinach ratings again:

df$spinach\_rating

## [1] 0 1 -3 0 -2 0 -3 1 0 -3 -3 3 3 0 2

If we want to take only the first element of this, we can use square brackets after the vector to tell R that we only want to first element.

df$spinach\_rating[1]

## [1] 0

If we want the first three elements instead we can put a vector in the square brackets.

df$spinach\_rating[c(1, 2, 3)]

## [1] 0 1 -3

The position of an element is often called the elements *index*. Above we found the spinach ratings with index 1, 2 and 3.

### Exercise

1. Extract elements 3 and 4 from the spinach rating vector.
2. Extract the element with index 10.

## Logical subsets

At the start I introduced you to three types of vectors: numeric, character and logical. We've seen lots of numeric data, a few pieces of character data but we've not looked at logical data very much. While we often have data which is naturally true or false, most commonly in R we create logical data to help us do other things.

It's very easy to create a logical vector from another type of vector. We just use R code to write down a true or false answer question. For example, if we want to create a new vector which tells us if someone is under 150cm tall we just write down:

df$height < 150

## [1] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [12] TRUE TRUE FALSE TRUE

R returns to us a vector of true and false values. You might remember the meaning of the < symbol. It is just a short hand way of saying 'less-than'. For every element of df$height R checks if that element is less than 150cm and returns TRUE or FALSE correspondingly.

We can use this to create a new vector in our data frame called is\_tall, where being tall is defined as being over 150cm. Here we are using the reverse of 'less-than', >, which means 'greater-than'.

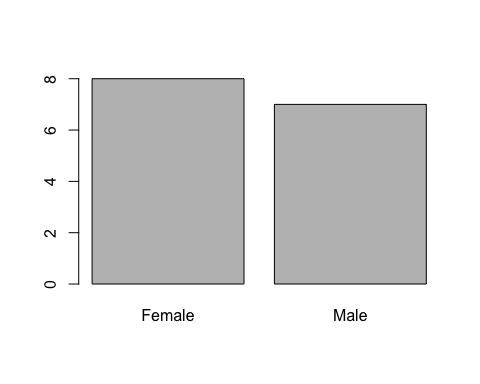
df$is\_tall <- df$height > 150

### Exercise

1. Add a new logical vector to df which tells you if someone likes spinach or not (Say anyone giving a rating greater than zero likes spinach). Use str to verify that df has changed.
2. Create a bar chart to see the proportions of people who like spinach.

In the new data you've just read in have a look at the gender variable. Let's start by making a barplot.

gender\_table <- table(df$gender)  
barplot(gender\_table)



We have two genders in this dataset: male and female. Male survey participants form a subgroup of the data, as do female survey participants. Very commonly, particularly when we have large data, we want to analyse different subsets of our data differently. We can use the double equal sign to check with a character vector is equal to a value.

df$gender == 'Male'

## [1] TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE FALSE FALSE TRUE  
## [12] FALSE TRUE FALSE FALSE

df$gender == 'Female'

## [1] FALSE FALSE TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE FALSE  
## [12] TRUE FALSE TRUE TRUE

Now, we are probably interested in whether our spinach sentiment is different in males and in females. We can use logical subsetting to work this out. Just like we did with numeric subsetting, if we put a logical vector inside square brackets we will get a subset. Only this time, rather than getting a subset for certain indexes in the data we get a subset which only contains the true.

numeric <- c(1, 2, 3, 4)  
logical <- c(TRUE, FALSE, TRUE, FALSE)  
  
numeric[logical]

## [1] 1 3

This makes it easy to find all the spinach ratings that come from men:

df$spinach\_rating[df$gender == 'Male']

## [1] 0 1 0 -3 1 -3 3

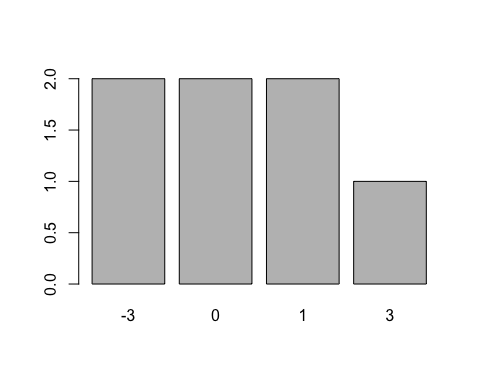
And come from women:

df$spinach\_rating[df$gender == 'Female']

## [1] -3 -2 0 0 -3 3 0 2

Once we have that it's easy to do apply anything we've learnt to just a subset of the data including plotting and finding summary statistics. Rather than using all of the spinach ratings (df$spinach\_rating), we just use a subset of the spinach ratings (df$spinach\_rating[df$gender == 'Male']). Here's a summary of male spinach ratings:

men\_spinach\_table <- table(df$spinach\_rating[df$gender == 'Male'])  
barplot(men\_spinach\_table)



mean(df$spinach\_rating[df$gender == 'Male'])

## [1] -0.1428571

median(df$spinach\_rating[df$gender == 'Male'])

## [1] 0

sd(df$spinach\_rating[df$gender == 'Male'])

## [1] 2.193063

IQR(df$spinach\_rating[df$gender == 'Male'])

## [1] 2.5

# Exercise

1. Find all the bar charts and summary statistics above for females.
2. We can also subset the data by age. Make a table and barplot of the different values of age in our data.
3. Find the mean and standard deviation spinach rating for each age group.
4. Do you think there's a difference between adults in children in the spinach rating?

# Statistical Inference

## Samples and Populations

Going back to our spinach data. When I calculated the average I said "on average our survey respondents gave spinach a rating of -0.3". You might notice that was only a statement about the 15 people interviewed. Most of the time when we do a survey it's not because we are very interested in the opinions of the exact people in the survey, but because we hope that this sample of people will tell us something about the population as a whole. When we survey 15 people about spinach what we really want to know is what does everyone in the world think about spinach overall (or at least what does everyone in your country thinks about spinach). Going from facts about a sample to facts about a population is called statistical inference. Almost every time you gather data you'll need to do some form of statistical inference. We only very rarely have data about every single thing we are interested in; we need to find a way to make a guess about the whole from looking at the part.

How can we go from saying something about a sample to saying something about a population? Well, first we can estimate that average spinach sentiment is -0.3 across the population - the same as across the sample. That seems to be the most we can say, based on the information we have. However, the statistical inference we are going to cover will gives us ways of going beyond that: it gives us a measure of uncertainty. This uncertainty measure how close we might be to the real, population value. This lets us understand what values we might see if we repeated the experiment the next day. It also lets us formulate hypothesis about the population and test if they are true.

### Exercise

1. If we had interviewed 150 people, would we be more or less certain about our estimate for spinach sentiment, than with the 15 people we currently have?
2. What if everyone in the survey had answered 0 or -1? In the actual survey we had a relatively large standard deviation of 2.1. How would our certainty about the real answer change if the standard deviation was 0.2?

### Answers

1. The more people you survey the more certain you be that the estimate you get is close to the real answer.
2. The smaller the standard deviation the more certain you can be out your answer. If every value in the sample is close together then we would expect that doing a second sample would also give values close together, and give a similar mean. If we have a very wide range of values then finding another 10 values might change the mean significantly, therefore we must be less certain about that mean.

## The formula for standard deviation

Most of the time we have only a sample, but occasionally you know what you have the whole population: you might have data on every one of your customers, or you might have seen every member of a rare species. When you have the full population you don't need to do any statistical inference; you can calculate the exact answer.

Most statistics you might want to calculate is done the same way in the population and in a sample. The mean is always just the sum of all values divided by the number of values. However, there is one very important statistic which has a slightly different definition in a sample: the standard deviation. Earlier when I presented the standard deviation formula I gave:

However, this is actually only the formula for the population standard deviation. When you have a sample you need to use this formula, the only difference is in the denominator, rather than :

This is because in a sample you don't know the exact answer for the mean. As a result most values are probably very slightly further away from the true mean than the sample mean. Luckily, we can mathematically prove that changing the denominator to will correct for this underestimation. This link provides more mathematical detail about the deveration of the standard devation formula (<https://www.ma.utexas.edu/users/mks/M358KInstr/SampleSDPf.pdf>).

For large values of the difference between the two formulas is pretty negligible, and having to work with actual populations is very rare. However, I hope this explains the somewhat confusing in the formula.

# Hypothesises Tests and P-Values

The most common type of statistical inference carried out is hypothesis tests. This is particularly useful for doing a scientific study, when you have a hypothesis about a natural phenomenon. You would then gather data and use the data to test the hypothesis. A similar pattern is often followed in businesses - someone in the company has an hypothesis about their customers, and then gathers data on them to see if they were right. It's also very common to have a dataset collected by someone else and, though exploring that data, generate hypothesis about the population.

To do a hypothesis test you must start with the *null hypothesis*. This is simply the hypothesis that nothing has happened. We will cover a few varieties of hypothesis tests, which each have their own null hypothesis. You then gather the data, and do a hypothesis test you if you have enough data to say that null hypothesis isn't true. The nature of hypothesis testing means that you can never say with confidence that a given hypothesis is true. You can only say that the null hypothesis is unlikely.

When we do a hypothesis test, we assume the null hypothesis is true, and then we mathematically simulate the types of answers we would get if that was true. At this point we can compare the answer we actually get, with the types of answers we would expect under the null hypothesis. If the answer we had is quite likely to happen under the null hypothesis then we cannot reject the null hypothesis; it is perfectly reasonable that the null hypothesis is true, perhaps nothing is happening here. However, if we find that our result is very unlikely under the null hypothesis then we have evidence that the null hypothesis might not be true and there might be some real effect in the population. At this point, the opposite of the null-hypothesis (often called the alternative hypothesis) looks more likely.

To be more precise, we decide to reject or accept the null hypothesis using a special statistic called a *p-value*. All hypothesis tests produce a p-value, which we can then use make decisions about our null hypothesis. The p-value is:

**The probability of seeing a result as extreme (or more extreme) than the result we have, given the null hypothesis is true**

So, if we have a high p-value then we can say that it's reasonable that this result happened under the null hypothesis, so the null hypothesis may well be true. I would recommend learning the definition above - it's important to have a precice understanding of what a p-value actually means.

If we have a low p-value, then it would be surprising to see our results when the null hypothesis is true. This means that we have good reason to believe the null hypothesis isn't true.

A very common misunderstanding is that the p-value is the probability of the null hypothesis being true. But this isn't right; it's something subtly different. We cannot use hypothesis testing to get probabilities of individual hypothesis - all we can talk about is the probability of seeing a result, under different hypotheses.

By convention we normally declare that whenever the p-value is less than 0.05 then that's a low enough probability to reject the null hypothesis. That is, if data like this appears less than 5% of the time under the null hypothesis then the null hypothesis isn't true. However, this cut-off point is rather arbitrary; there's basically no difference the strength of evidence when the p-value is 0.049 and 0.051. And remember when p is 0.05 one in twenty times we would see something like our data if the null hypothesis is true. A cut-off p-value of 0.01 is also often used, and some branches of physics use cut -ff values as low as 0.0000003.

When the p-value is below a prespecified value and we reject the null hypothesis based on this, we often say that our result has reached 'statistical significance' or that there is 'a statistically significant difference'.

### Examples - Two Sample T-test

### Spinach rating in men and women

One of the other pieces of data we collected on our survey was the gender of our participants. Earlier we found that the average spinach rating from females was 0.3, and the average ratings from men was -0.7. Do you women think actually like spinach more than men? Ignore everything anything else you might think you know from your experiences. Based on only on this data do you think there is a difference between males and females? On average we do see a slightly higher rating for females, but we only have 15 respondents. Is this enough data to be confident about anything in the real word?

Whether women like spinach more than men is exactly the type of hypothesis we can test using hypothesis testing. The test we use *two sample t-test*. For a two sample t-test the null hypothesis is always that there is no difference between the two groups we are looking at, in this case we can write the null hypothesis more precisely:

There is no difference between male's average spinach ratings and female's average spinach ratings.

We use a function in R called t.test to calculate the results of this test for us. The ~ symbol means that we are writing a formula in R. The input to t.test can be roughly read as 'spinach rating depends on gender'. Asking if there is a difference between men and women's spinach ratings is effectively the same as asking if spinach rating depends on gender.

t.test(df$spinach\_rating ~ df$gender)

##   
## Welch Two Sample t-test  
##   
## data: df$spinach\_rating by df$gender  
## t = -0.20423, df = 12.747, p-value = 0.8414  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -2.692705 2.228419  
## sample estimates:  
## mean in group Female mean in group Male   
## -0.3750000 -0.1428571

We get quite a bit of output from our test. The important bit to focus on just enough is where it says p-value = 0.3877. Going back to our definition of a p-value from earlier and using our null hypothesis and p-value what can we say?

The probability of seeing a result as extreme (or more extreme) than the result we have, if there was no difference between men and women is 0.39.

i.e. when there truly is no difference between men and women's spinach preference would expect to see some difference just due to random chance. We'd see a difference about as large as we've seen about 40% of the time. Can we reject the null hypothesis based on this result? No way! This data is entirely consistent with the null hypothesis. We have good reason to think that overall there's no difference between men and women even though this data is showing a small difference.

### Spinach rating in adults and children

Okay, now let's see if our data supports a difference between spinach rating in adults and children. From the earlier exercise you'll have calculated that the average rating amongst adults is 0.7 and the average rating amounts children is -1.7. That's a fairly big difference but is it enough of a difference to be confident about a difference in the population?

Let's write out the null hypothesis for this test

There is no difference between adults and children in their rating of spinach.

And now let's use t.test to find the p-value.

t.test(df$spinach\_rating ~ df$age)

##   
## Welch Two Sample t-test  
##   
## data: df$spinach\_rating by df$age  
## t = 2.4595, df = 11.384, p-value = 0.03104  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.253861 4.412806  
## sample estimates:  
## mean in group Adult mean in group Child   
## 0.6666667 -1.6666667

Using the definition of p-value from earlier, what can we say as the result of this test?

The probability of seeing a result as extreme (or more extreme) than the result we have, if there was no difference between adults and children, would be 0.03.

This p-value is much lower, and using our cut-off value of 0.05 does seem to suggest that the null hypothesis is not true. That would lead to us accepting the idea that there probably is an overall difference between preference for spinach in adults and children. Another way of saying this is that there is a statistically significant difference in spinach rating in adults and children.

### An example - One Sample T-test

We've just seen an example doing a two-sample t-test. Here we are going to look at a one-sample t-test. This test is used for testing whether the average of a group might be equal to some pre-selected value. Very often we test is the average could be equal to zero; often we have good reason to believe the average will be zero and we want to check if the data can support that hypothesis.

Looking at the average spinach sentiment of -0.3 you wonder whether it is likely that overall people are perfectly neutral on spinach. We can formulate this question as a hypothesis test where the null hypothesis is:

Average spinach sentiment is zero

The other option alternative hypothesis would be that the average population spinach sentiment is something other than zero.

We use the same t.test function. We include the option mu = 0 because we are testing if the mean might be zero

t.test(df$spinach\_rating, mu = 0)

##   
## One Sample t-test  
##   
## data: df$spinach\_rating  
## t = -0.48712, df = 14, p-value = 0.6337  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -1.4407930 0.9074597  
## sample estimates:  
## mean of x   
## -0.2666667

From the output we can see that the p-value is 0.64. This means that, when we have a population of people who, on average, have a spinach sentiment of zero we would see data that looks like ours 63% of the time. That gives us no reason to reject the null hypothesis; it seems reasonable that the population average might be zero.

### Exercise

Carry out tests to investigate the following questions. Do the test in R and write a few sentences to summarise the results.

1. The average rating for chocolate is zero.
2. Average chocolate rating is different between men and women.
3. Average chocolate rating is different between adults and children.
4. Height is different for adults and children.

# Confidence Intervals

Creating hypothesis and testing them is a very useful way of analysing data. However, not all types of data analysis follow this pattern. Often we want some measure of population uncertainty, without having an explicit hypothesis about what we believe is happening. This is where confidence intervals come in useful; they give us a method of describing the amount of uncertainty there is in our estimates.

A confidence interval gives a lower and upper bound on our estimate. You will often see the confidence interval quoted as as a number followed by the upper and lower bound in brackets. For example: 'we found that the average rating given to spinach was -0.3 (-1.4, 0.9)'. This means that the confidence interval around our estimate for average spinich sentiment had a lower value of -1.4 and an upper value of 0.9.

However, be careful with confidence intervals. There precise meaning is a little tricky. Again, I would recommend memorising the following definition:

For a 95% confidence interval if we take multiple samples, and make a confidence interval for each sample, then 95% of the confidence intervals would contain the true sample.

Another way of putting this is what if we look a sample 100 times the true value would lie within 95 out of the 100 confidence intervals.

We can have other levels of confidence; 99% and 90% confidence internvals are also commonly used. For different values of confidence the definition is basically the same, but the frequency in which the true value lies in the confidence interval changes. For a 90% confidence interval the true value will like within 90% of repeted sample confidence intervals, for 99% confidence the true value will lie within 99% of repeted sample confidence intervals.

### Excercise

1. Thinking about the definition of a confidence interval, will 90% intervals be wider or narrower than 95% intervals? What about 99% intervals?

### Answer

1. 90% is the narrowest, then 95%, 99% is the widest. Each time take a sample and caluclate a confidence interval we have a chance of missing the true value in that confidence interval. The wider the interval the less likley we are to miss the true value. With a 90% interval we can miss the true value 10% of the time, while with a 99% interval we have to made the interval wide enough to only miss 1% of the time.

The output of our one sample t-test included the 95% confidence interval for the mean.

t.test(df$spinach\_rating)

##   
## One Sample t-test  
##   
## data: df$spinach\_rating  
## t = -0.48712, df = 14, p-value = 0.6337  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -1.4407930 0.9074597  
## sample estimates:  
## mean of x   
## -0.2666667

So, from the output above we can see that the 95% confidence interval for average spinach rating goes from -1.4 to 0.9. Or, to put that another way spinach rating is -0.3 (-1.4, 0.9). If we repeated our sample of 15 people 100 times, that confidence interval would contain the true mean 95 out of the 100 times. We can calculate other intervals using the conf.level parameter:

t.test(df$spinach\_rating, conf.level = 0.99)

##   
## One Sample t-test  
##   
## data: df$spinach\_rating  
## t = -0.48712, df = 14, p-value = 0.6337  
## alternative hypothesis: true mean is not equal to 0  
## 99 percent confidence interval:  
## -1.896288 1.362954  
## sample estimates:  
## mean of x   
## -0.2666667

In a two-sample t-test the confidence interval in the output tells gives us a confidence interval for the difference between our two groups.

t.test(df$spinach\_rating ~ df$age)

##   
## Welch Two Sample t-test  
##   
## data: df$spinach\_rating by df$age  
## t = 2.4595, df = 11.384, p-value = 0.03104  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.253861 4.412806  
## sample estimates:  
## mean in group Adult mean in group Child   
## 0.6666667 -1.6666667

Putting the above output into words, we can say on adults gave spinach an average rating of 0.7, and children gave spinach an average rating of -1.7. We see an average 2.3 point difference between adults and childen with a 95% confidence interval for that difference of (0.3, 4.4). We have evidence from an hypoethsis test to reject the hypoethesis that adults and children give the same average rating (p-value -0.03).

Confidence intervals and p-values rely on the same mathematics (which we will go into detail on soon), so they have a certain symmetry. Have a look at the confidence interval for the difference between genders:

t.test(df$spinach\_rating ~ df$gender)

##   
## Welch Two Sample t-test  
##   
## data: df$spinach\_rating by df$gender  
## t = -0.20423, df = 12.747, p-value = 0.8414  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -2.692705 2.228419  
## sample estimates:  
## mean in group Female mean in group Male   
## -0.3750000 -0.1428571

We rejcted the null hypoethesis that the difference between adults and children was zero; we did not reject the null hypoethesis that the difference between males and females was zero. Notice, that our our confidence interval for the difference between adults and children did not contain zero, while our confidence interval for the difference between males and females did.

Confidence intervals and hypoethesis tests always match. If we use a hypothesis test to reject a value then that value will not lie in the confidence interval. However, if we cannot reject a given value then that value will lie in the confidence interval.

Different confidence intervals corripsond to different degrees of certainty in our t-tests. The p-value for a differnt between adults and children was 0.03; if we chose our cut-off for statistical significance to be 0.01 for the age test we would not have rejected the null hypothesis. Look at the 99% confidence interval for the difference between adults and children:

t.test(df$spinach\_rating ~ df$age, conf.level = 0.99)

##   
## Welch Two Sample t-test  
##   
## data: df$spinach\_rating by df$age  
## t = 2.4595, df = 11.384, p-value = 0.03104  
## alternative hypothesis: true difference in means is not equal to 0  
## 99 percent confidence interval:  
## -0.5932309 5.2598975  
## sample estimates:  
## mean in group Adult mean in group Child   
## 0.6666667 -1.6666667

Now, this wider confidence interval does contain zero. That's because significance at the 0.01 level (the 1% level) corriponds to 99% confidence intervals. Since we cannot reject 0 as a value at the 1% level, the 99% confidence interval will contain zero. In general for a signicicance cut-off of % will match with a confidence interval at the (100 - )% level.

It's tempting to say that a 95% confidence interval has a 95% chance of containing the true mean. However,

### Excersise

1. Calulcate the 90%, 95% and 99% confidence intervals for the average chocolate rating. Confirm our intuition that the 90% interval will be narrower than the 95% interval and the 99% interval.
2. Write a short paragraph on the results of the two sample t-test for a difference between spinach rating for different genders.

# The mathematics behind p-values and confidence intervals

But where does that p-value actually come from? It can seem a little magical our ability to make statements about the population from just a sample. However, the maths behind hypothesis testing and confidence intervals is relatively simple; I hope you will be able to see the logic in it.

One of the most important results in the mathematical foundation to statistics is the *central limit theorem*. Without this theorem, basically none of the statistics we cover would exist. The central limit theorem tells gives us rules about what happens when we repeatedly take a sample. In practice, you would never repeatedly sample a population (why do 10 samples of 20 people when you could just do one big sample of 200 people?). However, having a mathematical theorem about what happens if you were to repeatedly take a sample is very useful, because we can know what would happen if we had repeated our sample without having to actually do it.

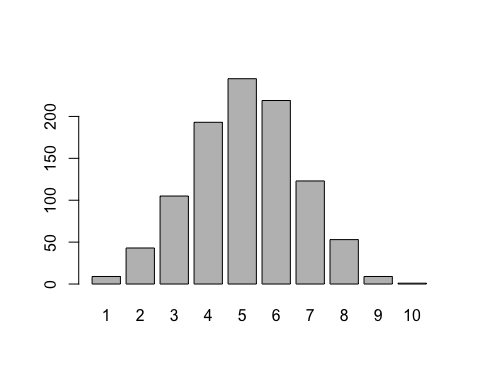
The central limit theorem says that if you take a large enough sample and find the mean, that mean will probability be close to the actual mean (obvious right). But it also gives us some idea of how close we are likely to be. We can know exactly what our chance of being very far away is, and what our chance of being very close is.

Think about flipping a coin 10 times. This is a great example because we know exactly what the average proportion of heads should be (a half). But every time we flip the coin we get a different estimate for the proportion of heads. Out of 10 coin flips we know that the average number of heads should be 5. However, each time we do the 10 coin tosses we get a different answer. I flipped a coin 10 times and counted the number of heads and I just got 4. Let me do that another 8 times and see what I get:

4 out of 10 3 out of 10 7 out of 10 6 out of 10 5 out of 10 5 out of 10 4 out of 10 5 out of 10

Now, we do sometimes get exactly the right answer, although most of the time we don’t (when doing statistics in the real word you will probably never get the exactly right answer!). But we rarely get very, very wrong answers.

Now I've done my 10 coin flip experiment 1000 times\*. Here's a bar chart showing how often I got each result

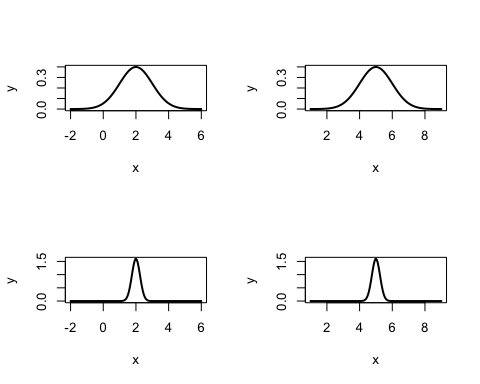


* I didn't actually flip 10 coins 1000 times - I got R to simulate what would happen if I did.

### Normal distributions

The shape that plot very closely follows what is known as the *normal distribution*. A distribution is just a mathematical way of writing down the probability of things happening. The distribution of the coin toss would describe 5 being quite likely and the probability of results decreasing the further they are from 5. All normal distributions follow that same pattern: one result being most likely, other results becoming less likely the further away they are from that result. Normal distributions vary in the result that is most likely, and the how likely father away results are. The plot below shows 4 different normal distributions:

par(mfrow = c(2, 2))  
  
x <- seq(-2,6,length = 200)  
y <- dnorm(x, mean = 2, sd = 1)  
plot(x,y, type = "l", lwd = 2)  
  
  
x <- seq(1,9,length = 200)  
y <- dnorm(x, mean = 5, sd = 1)  
plot(x,y, type = "l", lwd = 2)  
  
  
x <- seq(-2,6,length = 200)  
y <- dnorm(x, mean = 2, sd = 0.25)  
plot(x,y, type = "l", lwd = 2)  
  
  
x <- seq(1,9,length = 200)  
y <- dnorm(x, mean = 5, sd = 0.25)  
plot(x,y, type = "l", lwd = 2)



par(mfrow = c(1, 1))

The normal distributions on the left both have 2 has their most likely value; the distributions of the right have 5 as their most likely value. The difference between the top row and the bottom row is that the most we are much less likely to see values far away from the middle in the bottom row.

This idea of the most likely value and the spread of values may seem familiar from when we discussed averages and measures of spread. Indeed, the most likely value in a normal distribution is known as the mean, and we can measure the expected variation from the mean using the standard deviation.

The good thing about normal distributions is that we can make exact calculations using them. As long as you know the mean and the standard deviation of a normal distribution, we can use a formula to answer questions like: what is the probability of seeing a result greater than 4? What is the probability of seeing a result less than -2? While normal distributions are just a mathematical model for how random events happen, they are very useful because many random events happen in a way that is approximately normally distributed. For example, most people tend to have an average height, with heights far away from the average height becoming less and less common. Another important example is manufacturing processes. A widget machine will produce widgets approximately 2cm long, and the errors in this length will be distributed normally. If we are willing to assume that something we are measuring is distributed normally, we only need to calculate the mean and the standard deviation; then we have answers to all sorts of trickier questions. For example, how many people over 6 foot 5 are likely to get on our aeroplane? How likely is it we will produce a widget under 1.9cm long?

### More central limit theorem

Arguably, the most important application of normal distributions is the central limit theorem. As we saw for our 10 coins experiment when we repeatedly take a sample the average will follow the normal distribution. The most likely value of the mean is the actual, population mean. Furthermore, the central limit theorem says that the standard deviation of the disruption will be:

Where is the standard deviation of the population and is the sample size. The distribution of means from doing samples is known as the *sampling distribution*.

This makes sense since the larger the population size, the smaller the standard deviation, and the closer we expect to be to the true mean. Furthermore, if we have a small population standard deviation it is easier to find averages close to the true mean. Small population standard deviation, means small sampling standard deviation, means we expect to be close to the mean on average.

### Hypothesis Testing

Now finally, once you understand the central limit theorem, how hypothesis testing works is quite straightforward. If our null hypothesis is that the true mean is zero, we can generate the sampling distribution we would expect under that hypothesis. All we do now is look at the actual mean we got. Means that are close to zero are more likely under the null hypothesis and means that are far from zero are less likely. We just find the probability of getting a mean as far away from zero as the mean we actually got, and that is our p-value. The two sample t-test works in a very similar way; we find the sampling distribution if the difference between the two groups was zero, and then find the probability of seeing our result if there really was no difference.

The last piece of the puzzle is why is the test we use in R called the t-test? Well, the formula for standard deviation I showed you earlier replied on the population standard deviation. If we don't know the population mean and are trying to estimate it, we almost certainty don't have the population standard deviation and have to estimate it using the sample standard deviation. This means we are actually less certain that we are close to our true mean, so we have to use a distribution very like the normal distribution which captures this extra uncertainty; that distribution is known as the t-distribution and hence the name t-test.

### Confidence intervals

For calculating a confidence interval, the logic is slightly different. You can probably tell that from the 'multiple samples' definition of a confidence interval we get get there quite directly from the central limit theorm.

As a reminder, the central limit theorm says that a sample estimate of the mean will be close to the true value of the mean. We can describe the probablity of each possible estimated mean using a normal distrubtion centred around the true mean, with a standard devation that depends on the sample size and the true standard devation.

When calculating a confidence interval we basically just reverse the central limit theorm. We can use a distrubtion centred around the estimated mean, and with a standard devation that depends on the sample size and the estimated standard devation. Again, we need to use the t-distrubtion, because we are only esimated the standard devation. Just

As an important aside you may at some point here the words 'Bayesian' and 'frequentist'. The statistics covered in this unit are all frequentist statistics. There exists an alternative way of doing many of the tests we will see under a different mathematical framework, and this is called Bayesian statistics. There is some debate about which way of doing statistics is better. Although, currently, frequentist statistics are far more commonly used and understood so that is all we will be covering.

# Correlation

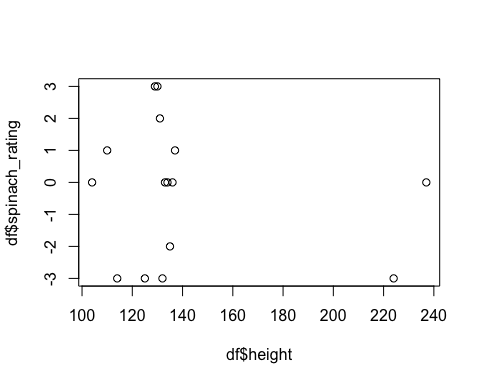
Most of what we have covered so far only looks at one variable at a time; plotting that variable, describing it, measuring the uncertainty in that distribution. However, almost always when we are analysing data we are interested in multiple variables. As you can imagine as soon as you start looking at more than one variable the complexity very quickly increases. You can do all the standard one variable analysis that we have covered so far on each variable. But you'll also be interested in the dependence between variables. One way of doing that is using tests like the two-sample t-test. When we analysed if spinach rating depends on age and gender we were each time looking at a pair of variables. But how can we analyse two continuous variables? How can we understand if spinach rating and height are connected, or spinach rating and chocolate rating?

When starting any type of analysis it is a good idea to look at a plot. Just like we looked at plots before we started calculating summary statistics and doing hypothesis test we will also look at plots for two variables. Plots can give a fuller description of the data than any simple number can.

### Scatterplots

When you have two continuous variables and you are interested in the connection between them the best thing to start with us a scatterplot. In this plot you simply plot the value of the first variables along the x-axis and the value of the second variable along the y-axis. This gives you a two dimensional summary of your data points. Again, this is very easy to do in R:

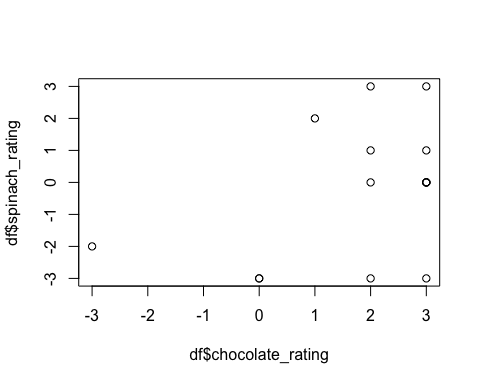
plot(df$height, df$spinach\_rating)



In the plot above we can see the skewed distribution of height, with most heights below 140cm, and a few above 220cm. We can also see that the average spinach rating seems quite unrelated to height. The two very tall people gave quite low ratings, but the sentiment was pretty mixed in the group of less tall people.

Now let's look at another two variables from our data - spinach rating against chocolate rating.

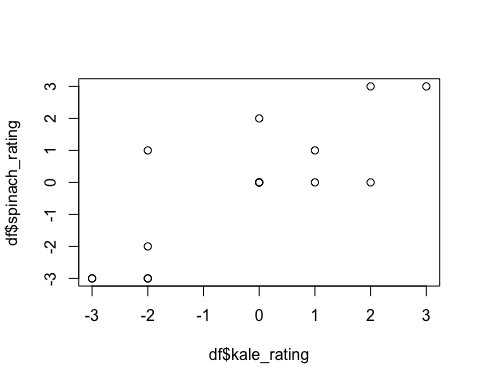
plot(df$chocolate\_rating, df$spinach\_rating)



Here we have quite a different pattern. There's a group of people who like both spinach and chocolate. But not liking spinach does not seem to be related to not liking chocolate.

Now let's look at chocolate rating against kale rating.

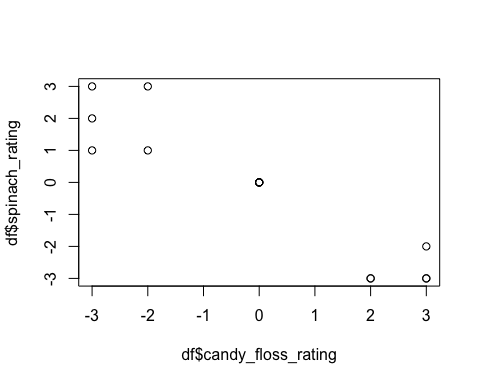
plot(df$kale\_rating, df$spinach\_rating)



Here we see a much stronger link; people who like kale also like spinach and people who dislike kale also dislike spinach.

Finally let's look at spinach rating vs. candy floss rating.

plot(df$candy\_floss\_rating, df$spinach\_rating)



Here we see another strong link, but a very different one. People who like candy floss are much less likely to like spinach and people who like spinach tend almost always give low ratings to candy floss.

We can define this idea of 'relatedness' using mathematics. We can produce a statistic called the 'correlation coefficient' which measures how strongly related two continuous variables are. This is very useful for formalising those intuitions we have that kale rating and spinach rating have much more in common than height and spinach rating.

Correlation goes on a scale from -1 to 1. When two variables are very tightly related: when one is high the other one is always high, when one is low the other one is always low then they will have a correlation coefficient close to one. When two variables are tightly oppositely related; when one is high the other one is always low, then their correlation coefficient will be close to -1. Cases like spinach and height will have correlations close to zero; there is no real pattern linking the two variables. Cases were variables are somewhat positively related like spinach and chocolate preference have correlation coefficents between 0 and 1.

Correlation can be found in R using the cor function.

cor(df$height, df$spinach\_rating)

## [1] -0.1834243

The table below gives the correlation coefficient for the four relationships we saw above:

rel <- c('Height vs. Spinach Rating',  
 'Chocolate Rating Vs. Spinach Rating',  
 'Kale Rating Vs. Spinach Rating',  
 'Candy Floss Rating Vs. Spinach Rating')  
  
  
cor <- c(cor(df$height, df$spinach\_rating),  
 cor(df$chocolate\_rating, df$spinach\_rating),  
 cor(df$kale\_rating, df$spinach\_rating),  
 cor(df$candy\_floss\_rating, df$spinach\_rating))  
  
knitr::kable(data.frame(Relationship = rel, Correlation = cor) )

|  |  |
| --- | --- |
| Relationship | Correlation |
| Height vs. Spinach Rating | -0.1834243 |
| Chocolate Rating Vs. Spinach Rating | 0.4006086 |
| Kale Rating Vs. Spinach Rating | 0.8210445 |
| Candy Floss Rating Vs. Spinach Rating | -0.9341192 |

### The mathematics

To calculate the correlation coefficient between two variables we must first calculate the covariance.

The covariance is similar to the variance of one variable, except rather than describing the spread of one variable, it describes the combined spread of two variables. The correlation coefficient is simple the covariance of the two variables divided by their individual standard devotions:

Dividing by the standard deviations is what insures that all correlation coefficients are normalised to be between -1 and 1.

Calculating covariance is similar to calculating the standard deviation, or the variance. For each data point in x, and the corresponding data point in y we multiply the difference from their means together, and divide by the number of data points.

If the two data points are both above their mean then we get a positive contribution to the covariance (positive times a positive is a positive). If the two data points are both below their mean then we also get a positive contribution to the covariance (negative times a negative is a positive). If one of the data points is above the mean, and the other below then we get a negative contribution of the covariance (negative times a positive is a negative).

If the pairs of data points are mainly above their respective means at the same time and below their respective means at the same time, the covariance will be overall positive. If the one of the pair in consistently above the mean, while the other is consistently below the mean then the covariance will be overall negative. If we get a mix of both cases (because there's no real trend between the two variables) then we will get a covariance close to zero.

# A hypothesis test for correlation

Very commonly we want to test if the true population correlation in a dataset is zero. We may suspect that two variables are not linked in any way (another way of saying this is that they are *independent*). The best way to prove this is with a hypothesis test. Again, very easy to do in R:

cor.test(df$height, df$spinach\_rating)

##   
## Pearson's product-moment correlation  
##   
## data: df$height and df$spinach\_rating  
## t = -0.67276, df = 13, p-value = 0.5129  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.6359338 0.3629410  
## sample estimates:  
## cor   
## -0.1834243

The null hypothesis for this test is that the population correlation is zero. For the relationship between height and spinach we got a p-value of 0.51. This means that the probability of seeing a correlation of at least -0.183 when we have 15 data points and the true correlation is zero is 0.51. This means that our result is entirely reasonable under the null hypothesis, and it would be fair to summarise that the population correlation could be zero.

Let's have a look at kale rating vs. spinach rating

cor.test(df$kale, df$spinach\_rating)

##   
## Pearson's product-moment correlation  
##   
## data: df$kale and df$spinach\_rating  
## t = 5.1857, df = 13, p-value = 0.0001754  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5329248 0.9385585  
## sample estimates:  
## cor   
## 0.8210445

Here we get a much smaller p-value (0.0002). So, it seems unlikely that this data would happen just due to random chance when there truly was no correlation between kale and spinach. We can reject the null hypothesis of no correlation between spinach and kale preference; there almost certainty is a correlation between your preference for those two vegetables.

Again, note that in both the outputs above we also get a confidence interval. This is the confidence interval for the true correlation coefficent. It's always a good idea to report this interval every time you report a correlation coefficent.

### Exercise

1. Do a correlation test for the relationship between spinach and candy floss, and the relationship between spinach and chocolate.
2. Write a few sentences explaining the result of each test.

# Statistics Summary

## One variable - continuous

* Bar chart
* Histogram
* Mean and standard deviation
* Median and interquartile range
* One sample hypothesis tests
* Confidence intervals

## One variable - discrete

* Bar charts
* Table

## Two variables - one continuous, one discrete

* Multiple bar charts/histograms
* Summaries for each subset
* Two sample hypothesis test
* Confidence intervals

# Two variables - both continuous

* Scatterplot
* Correlation
* Correlation test

# Two variables - both discrete

* Multiple bar charts
* Two way table
* Association Chi-Squared

### TO DO

* proportion t-tests and hypothesis tests
* paired t-tests
* Introduction to big data
* Comparing several groups One way ANOVA two pages
* Association Chi square one page
* Interrogating a data set three pages