

Introduction to Statistics

with R

**Version 0.1**

# Licence

This manual is © 2013-14, Anne Segonds-Pichon.

This manual is distributed under the creative commons Attribution-Non-Commercial-Share Alike 2.0 licence. This means that you are free:

* to copy, distribute, display, and perform the work
* to make derivative works

Under the following conditions:

* Attribution. You must give the original author credit.
* Non-Commercial. You may not use this work for commercial purposes.
* Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a licence identical to this one.

Please note that:

* For any reuse or distribution, you must make clear to others the licence terms of this work.
* Any of these conditions can be waived if you get permission from the copyright holder.
* Nothing in this license impairs or restricts the author's moral rights.

Full details of this licence can be found at

<http://creativecommons.org/licenses/by-nc-sa/2.0/uk/legalcode>

Table of contents

[Introduction to Statistics 1](#_Toc372614934)

[with R 1](#_Toc372614935)

[Introduction 4](#_Toc372614936)

[Chapter 1: Power analysis 5](#_Toc372614937)

[Chapter 2: Drawing graphs with R 9](#_Toc372614938)

[Histogram 10](#_Toc372614939)

[Boxplots 13](#_Toc372614941)

[Bar charts 17](#_Toc372614942)

[Scatterplots 18](#_Toc372614943)

[Line graphs 19](#_Toc372614945)

[Exporting graphs from RStudio 20](#_Toc372614946)

[Power Analysis with qualitative data 21](#_Toc372614948)

[A bit of theory: the Chi2 test 25](#_Toc372614949)

[A bit of theory: the null hypothesis and the error types. 27](#_Toc372614950)

[Chapter 4: Quantitative data 28](#_Toc372614951)

[3-1 A bit of theory: descriptive stats 28](#_Toc372614952)

[The mean 28](#_Toc372614953)

[The median: 28](#_Toc372614954)

[The variance 29](#_Toc372614955)

[The Standard Deviation (SD) 29](#_Toc372614956)

[Standard Deviation vs. Standard Error 30](#_Toc372614957)

[Confidence interval 30](#_Toc372614958)

[3-2 A bit of theory: Assumptions of parametric data 32](#_Toc372614959)

[How can you check that your data are parametric/normal? 33](#_Toc372614960)

[Power analysis with a t-test 33](#_Toc372614962)

[3-3 A bit of theory: the t-test 40](#_Toc372614963)

[Independent t-test 41](#_Toc372614964)

[Paired t-test 42](#_Toc372614965)

[3-4 Comparison of more than 2 means: Analysis of variance 46](#_Toc372614967)

[A bit of theory 46](#_Toc372614968)

[Power analysis with an ANOVA 49](#_Toc372614970)

[3-5 Correlation 54](#_Toc372614971)

[A bit of theory: Correlation coefficient 55](#_Toc372614973)

[Outliers and influential cases 57](#_Toc372614974)

# Introduction

R is a popular language and environment that allows powerful and fast manipulation of data, offering many statistical and graphical options.

Graphical representation of data is pivotal when one wants to present scientific results, in particular in publications. R allows you to build top quality graphs (much better than Excel for example).

In this manual however, we are going to focus on the statistical possibilities of R. Whatever package you use, you need some basic statistical knowledge if only to design your experiments correctly, so there is no way out of it!

And don’t forget: you use stats to present your data in a comprehensible way and to make your point; this is just a tool, so don’t hate it, use it!

“To consult a 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." R.A.Fisher, 1938.

# Chapter 1: Power analysis

First, the definition of power: it is the probability of detecting a specified effect at a specified significance level. Now ‘specified effect’ refers to the effect size which can be the result of an experimental manipulation or the strength of a relationship between 2 variables. And this effect size is ‘specified’ because prior to the power analysis you should have an idea of the size of the effect you expect to see. The ‘probability of detecting’ bit refers to the ability of a test to detect an effect of a specified size. The recommended power is 0.8 which means you have an 80% chance of detecting an effect if one genuinely exists.

The main output of a power analysis is the estimation of a sufficient sample size. This is of pivotal importance of course. If you sample is too big, it is a waste of resources; if it is too small, you may miss the effect (p>0.05) which would also means a waste of resources. On a more practical point of view, when you write a grant, you need to justify your sample size which you do through a power analysis. Finally, it is all about the ethics of research really which is encapsulated in the Home office’s 3 R: Replacement, Refinement and Reduction. The latter in particular relates directly to power calculation as it refers to ‘methods which minimise animal use and enable researcher to obtain comparable levels of information from fewer animals’ (NC3Rs website).

When should you run your power analysis? It depends of what you expect from it: the most common output being the sample size, you should run it before doing the actual experiment (*a priori* analysis). The correct sequence from hypothesis to results should be:

**Hypothesis**

**Experimental design**

**Choice of a Statistical test**

**Power analysis**

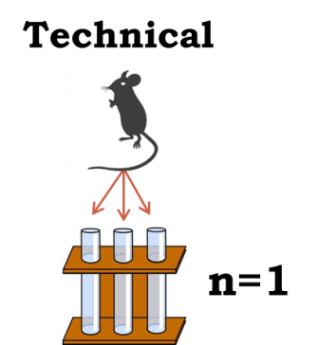
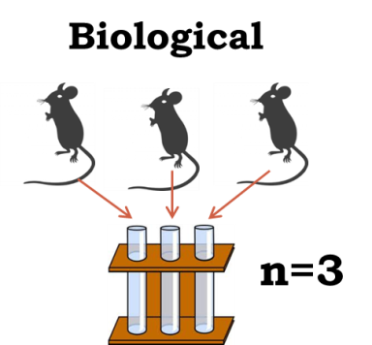
**Sample size**

**Experiment(s)**

**(Stat) analysis of the results**

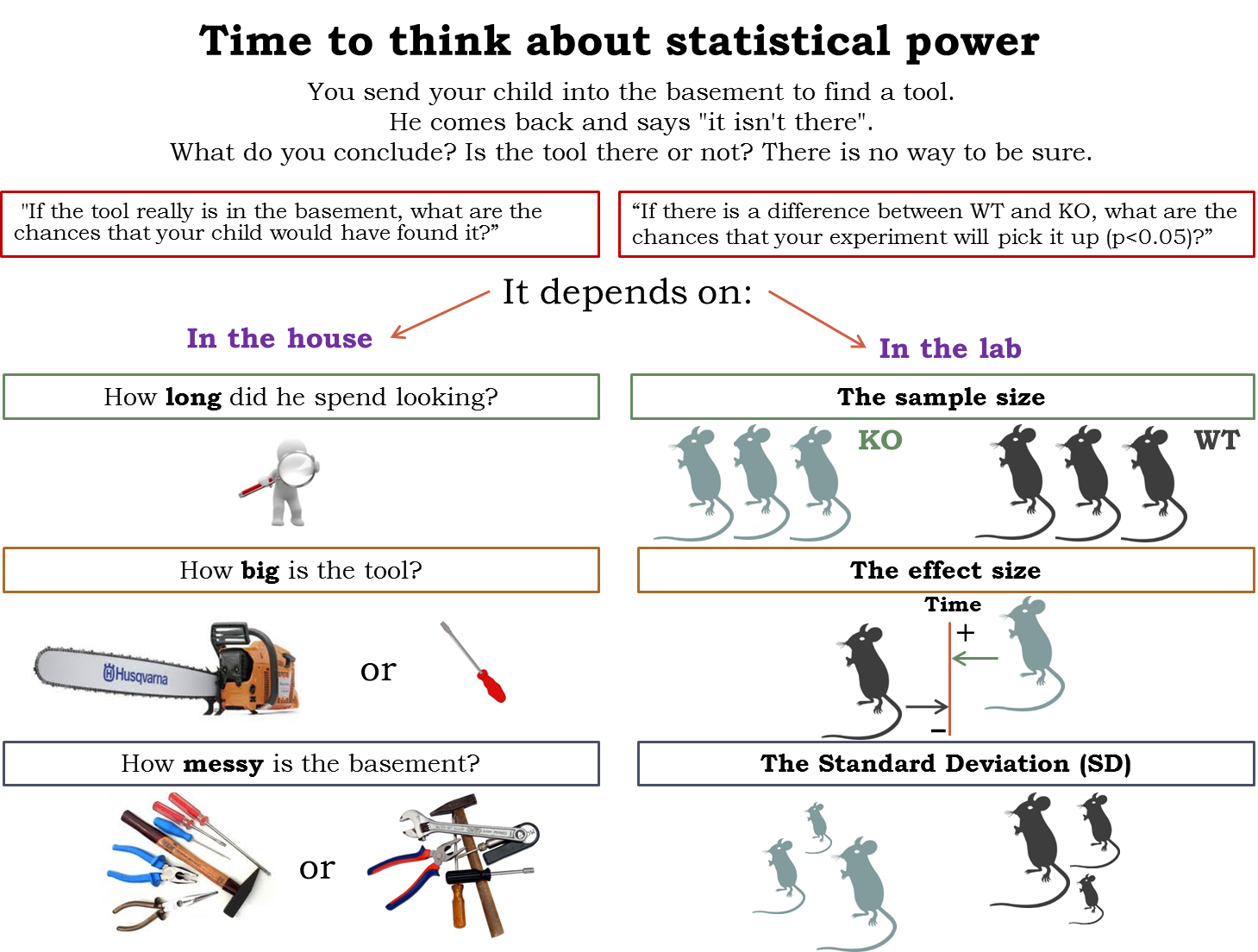
The power analysis depends on the relationship between 6 variables: the effect size of biological interest, the standard deviation, the significance level, the desired power, the sample size and the alternative hypothesis. The significance level is about the p-value, it is generally agreed to be 5% and we will come back to it later. The desired power, as mentioned earlier is usually 80%. The alternative hypothesis is about choosing between one and 2-sided tests, it is a technical thing and we will come back to it later as well. So we are left with the 3 variables on which we have pretty much no control or about which we cannot decide arbitrarily: the effect size, the sample size and the standard deviation. To help you understand what they are and how much they are connected, I will use an example.

Let’s make it simple, say you are studying a gene which is expressed in the brain and you are using a mouse model. Your hypothesis is that knocking out that gene will affect the mouse’s behaviour. The next step is to design your experiment. You are going to create a KO mouse in which gene A is inoperative and you are going to compare WT and KO mice’s behaviour through a set of tasks. Let’s say that the output of one of these tasks is a quantitative variable, the time taken by the mouse to achieve one task for example. Now you need to translate the hypothesis for this particular task into statistical question. The hypothesis is that knocking out gene A will affect KO mice behaviour which can be quantified by a change in the time taken to achieve the task. Statistically you need to know: what type of data you are going to collect (time), which test you are going to use to compare the 2 genotypes and how many mice you will need. When thinking about sample size, it is very important to consider the difference between technical and biological replicates. Technical replicates involve taking several samples from one tube and analysing it across multiple conditions. Biological replicates are different samples measured across multiple conditions.

Now you remember that a power analysis is about the relationship between 6 variables, 3 of which are directly linked to the experiment: the sample size, the effect size and the standard deviation.

The diagram below explains about these 3 variables.



First, the **sample size**, the name itself is self-explanatory. The aim of a power analysis is usually to find the appropriate sample size as in the one which will allow you to detect a specified effect. This effect, also called **effect size** of biological interest, can only be determined scientifically not statistically. It is either a difference that would be meaningful biologically, like an increase/decrease of 10% or 20% for a particular variable, or what you expect to get based on preliminary data. The larger the effect size, the smaller the experiment will need to be to detect it.

The **Standard Deviation** (SD) is basically the noise in your data, the variability you observe between your values. This you get ideally from a pilot study or from previous experiments or even the literature.

Now going back to the effect size, there are actually 2 different ones: the absolute one which is basically the difference between the mean value of, say, your control group and the one of you treatment group, and the relative one, also referred to as Cohen’s d. This one is the more useful and more widely used one as it accounts for the variability in your data.

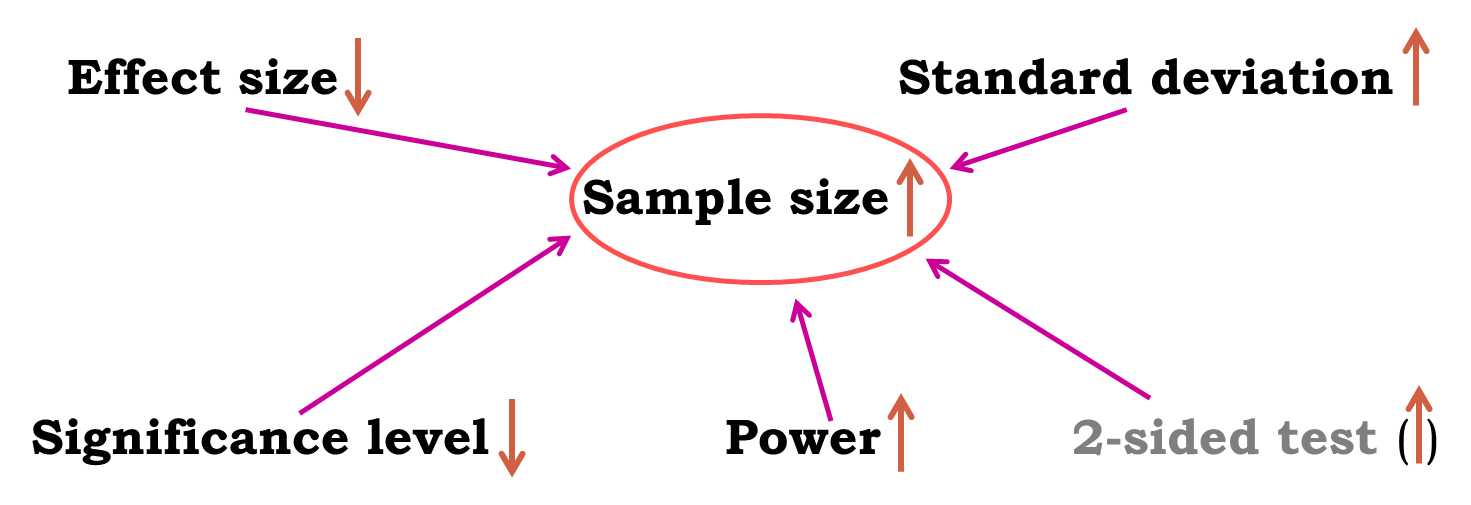
Cohen’s d = Mean 1 – Mean 2

Pooled SD

The **significance level** refers to the famous p-value which, for a test to be ‘significant’, should be below 0.05 (the 5% threshold). Going back to our experiment about finding out more on gene A, you would defined the p-value as the probability that a difference as big as the one observed between the WT and the KO could be found even if the knock-out of gene A does not affect the mouse’s behaviour. Basically it means that if you find a significant difference (p<0.05) between your 2 groups of mice, so corresponding to the effect size of biological interest, there is less than 5% chance that you would have been able to observe it if the knocking out of gene A did not affect the behaviour of the mouse.

The last variable is the so-called **alternative hypothesis**: one or two-sided test. This refers to the distribution of your variable: are you going to look at one side or at the 2 sides of it. In other words, and again going back to our experiment, do you want to answer the question: does it take longer to KO mice to achieve the task or do you simply want to know if there is a difference at all. Most of the time, in bench science, you go for the second question, even though you might think of one direction more than the other. You don’t have enough evidence to ‘choose’ to look only at one side of the distribution. It is pretty much only in clinical trials that people go for one-sided tests. They have already tested a particular drug for example in many systems/species so they have plenty of evidence about the effect of the drug. Finally, it is 2 times easier to reach significance with a one-side test than a 2-side one so a reviewer will always be suspicious if you go for the first one and if he asked for justification, you’d better have one!

The basic idea behind the power analysis is that if you fix any five of the variables, a mathematical relationship can be used to estimate the sixth. So going back one more time to our example, running the power analysis, your question can be: What **sample size** do I need to have an 80% probability (**power**) to detect an average 5 minutes difference (**effect size** and **standard** **deviation**) at a 5% **significance level** using a **2-sided test**? The variables are all linked and will vary as shown in the diagram below.



Now here is the good news, there are packages that can do the power analysis for you … providing of course you have some prior knowledge of the key parameters. Mostly you need to have some idea of the difference you are expecting to see or that would make sense, together with some information on the standard deviation. We will use G\*Power as we go through the statistical tests.

About power analysis, the important message is: after you have designed your **experiment**, run a **power analysis** to estimate the appropriate **sample size** that will allow you to do **good and ethical science.**

# Chapter 2: Drawing graphs with R

To draw graphs, we are going to use a package called *ggplot2* which is more powerful and more versatile than plot() which comes with the basic version of R. You will also need to download *Hmisc.*

The first time we want to use a package, we need to install it. To do so, you type:

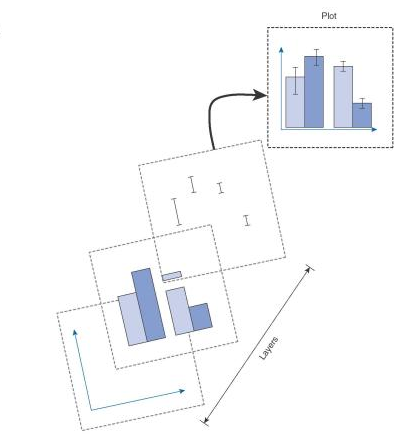
> install.packages(“ggplot2”)

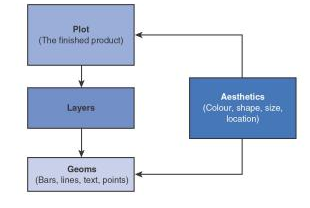
We then need to activate by executing the command:

> Library(ggplot2)

Each time we start an R session, we need to activate the package(s) we will need but we don’t need to install them again. To do so we can either use the command above or click on Packages in the graph panel. The list of downloaded packages will appear and all we have to do is select the one you want.

You can also install packages by clicking on Packages.

In *ggplot2*, a graph is made up of a series of layers.

There are a variety of geom() functions that determine what kind of geometric object is printed on a layer. Among the list of the most common ones are: geom\_bar(), geom\_histogram(), geom\_boxplot() and geom\_line(). Now these geoms have aesthetic properties (aes for short) that determine what values you are plotting, what they look like and where they are plotted ...

The general form of a command will look like that:

> MyGraph<-ggplot(MyData, aes(variable for x axis, variable for y axis))

That will work if we are plotting the raw data. If we plot means or error bars, then the form of the command will be different as we will see later.

If you need help with *ggplot2*, you can look into **R Graphics Cookbook** which .pdf version is available online at:

deca.cuc.edu.cn/Community/media/p/23508/download.aspx

Also good: <http://www.cookbook-r.com>

With the command above we have created a graph object but with no graphical elements. We need to add layers and to do so we just use the symbol ‘+’.

Now, as we just mentioned, some of these graphs require more than data. For example a box plot requires median, min and max values, same thing for error bars. The last thing you want to do is having to calculate those as you just want R to draw your graphs from raw data. Luckily, *ggplt2* has built-in functions called ‘stats’ which do exactly that.

The easiest way to learn and understand how it works is to try some examples. We will learn to draw histograms, boxplots, bars, scatterplots and line charts.

### Histogram

We will need the packages *reshape* and *reshape2*

We are going to plot data from the file: **DownloadFestival.dat.** In this file, is measured the level of hygiene of 810 concert-goes over 3 days (!).

First we need to load the data into a dataframe.

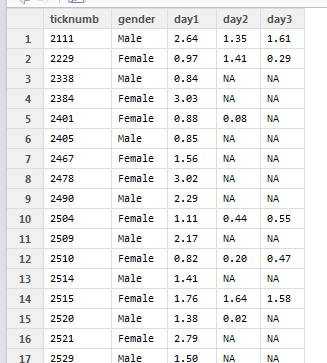
>setwd("wherever/is/the/file”)

>FestivalData<-read.table("DownloadFestival.dat", sep="\t", header=T)

The line below would also work:

> FestivalData <- read.delim(“DownloadFestival.dat”)

A good thing to do once you have loaded your data is to have a look at the file. You can do it by clicking on the name of the file in the workspace panel or by typing the command:

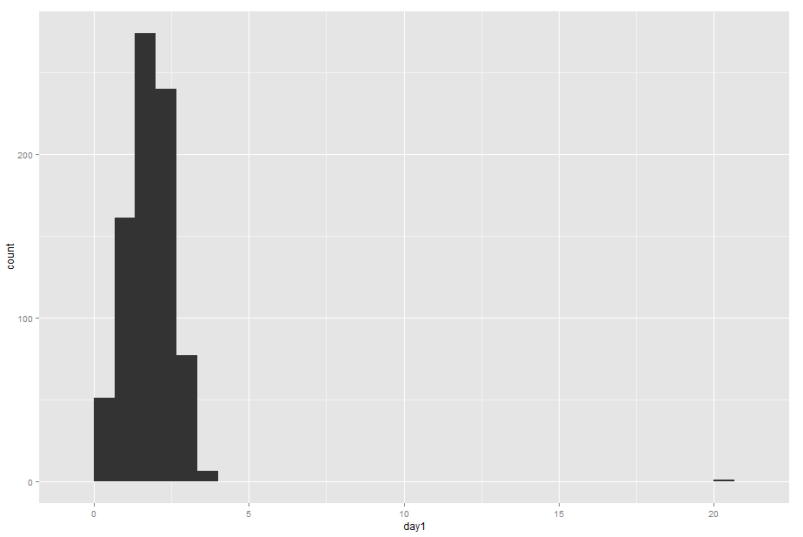


> View(FestivalData)

Now let’s say we want to look at the distribution of the values in day1.

To do so you go:

> Day1Histogram <- ggplot(FestivalData, aes(day1))

So this command tells R we want to plot the values in day1 (aes(day1)) from the file FestivalData.

With this we have only the object but not the layers so the next step is:

> Day1Histogram+geom\_histogram()

Now R knows we want to plot the data as a histogram and draws this graph. It is a histogram all right but not a very pretty one so we may want to make it prettier. There are several things we can play with: the width of the bins for instance.

>Day1Histogram+geom\_histogram(binwidth=0.4)

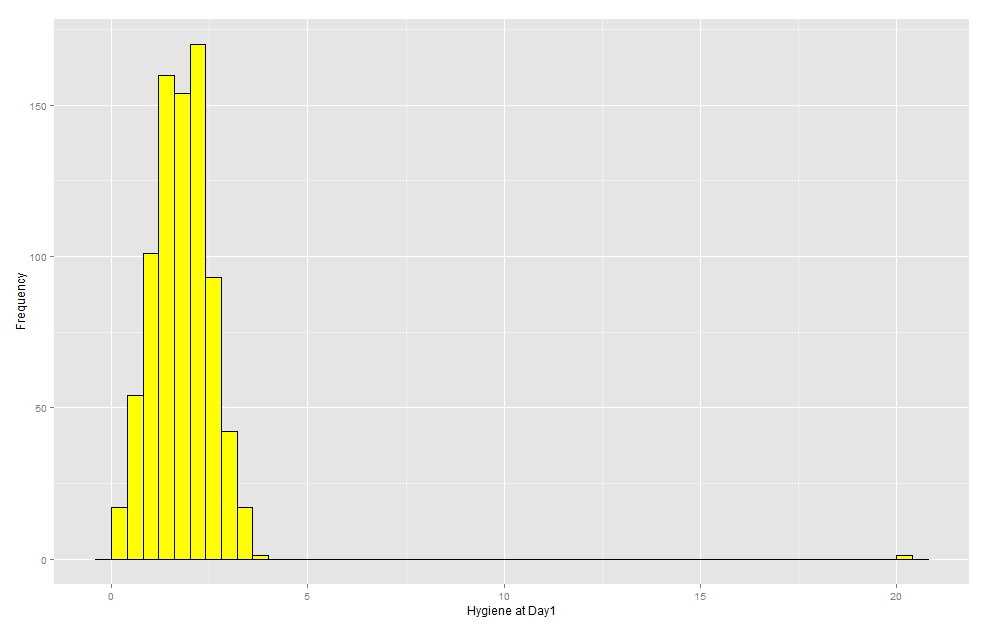
We can also add labels which is another layer, hence:

>Day1Histogram+geom\_histogram(binwidth=0.4)+labs(x="Hygiene at Day1", y="Frequency")

Finally we may want to play with the colours, in which case we can go:

>Day1Histogram+geom\_histogram(binwidth=0.4, color="black", fill="yellow")+labs(x="Hygiene at Day1", y="Frequency")

You can find a list of colours with that link: <http://sape.inf.usi.ch/quick-reference/ggplot2/colour> or at the end of the manual.



Now we may want to look at the 3 days at the same time, for that we will have to restructure the file to be able to use the facet() function. To do that we will need to download a package called ‘*reshape2*’ (and also ***‘****reshape’*!).

Once you have done that, and before plotting the graph, we will use the melt() function which basically will restructure your file or, as the help (?melt) puts it ‘Melt an object into a form suitable for easy casting’. We need to tell R we want to stack the day’s variable but we want to keep the ‘ticknumb’ and the ‘gender’ as identifier.

>FestivalData.stack<-melt(FestivalData, id = c("ticknumb","gender"))

Or:

>FestivalData.stack<-melt(FestivalData, id = 1:2)

Let’s rename the column which contains the days:

>colnames(FestivalData.stack)[3]<-"day"

Now the data are ready and we can apply the facet() function. It comes in 2 flavours: facet\_grid() or facet\_wrap(). We will use the first one.

>Histogram.3days<-ggplot(FestivalData.stack,aes(value))

+geom\_histogram(binwidth=0.4, color="black", fill="yellow")

+labs(x="Hygiene at Day1", y="Frequency")

+facet\_grid(gender~day)

>Histogram.3days

Basically you recognise the line we typed earlier, we have simply added the facet() function and asked for rows as gender and columns as days.

### M:\Work\R Course\Exercises\Histogram 3 days 2 genders.jpeg

Now one thing you may have noticed is that the x=axis is much wider than it needs to be. This is explained by the presence of a potential outlier. An outlier is a value which is very different from the rest of the data. We will go more in depth about it later, the important thing here is to be able to visualise data and spot anomalies.

Incidentally, if you don’t like the grey background or find that the font is too small, you can change the general theme of *ggplot2*:

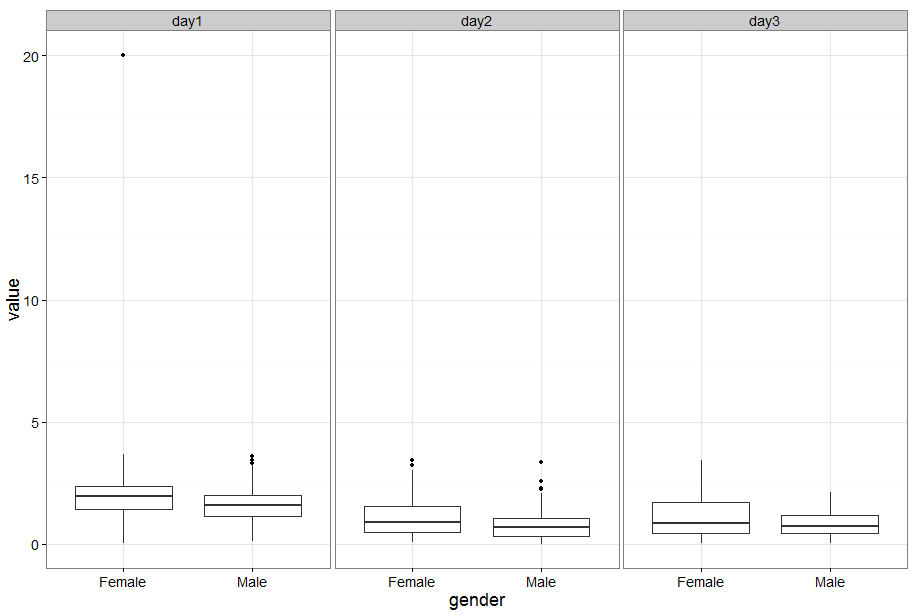
>theme\_set(theme\_bw(base\_size=18))

It changes the default setting in *ggplot2* so it will last for the entire session.

### 

### Boxplots

Another way to explore data graphically is boxplots. We will apply the boxplot() function to the ‘melted’ version of our festival data:



>Boxplot.festival <-ggplot(FestivalData.stack, aes(gender,value))

+geom\_boxplot()

+facet\_grid(~day)

So with the command above, we have asked *ggplot2* to draw a boxplot geom\_boxplot()from data contained in the festivalData.stack file from which we only wanted x=gender and y=value: aes(gender,value). Now since the variable gender is already used within each day, we only need to specify day to the facet() function.

The black dots are ‘official’ outliers. There are different ways to deal with those, one of which is to eliminate them once it is established they are aberrant values.

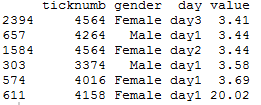
In R, there are different ways to eliminate an outlier and we can start by visualizing it by sorting out the data so that the very high values are at the bottom. But first, we should eliminate the NA values as they tend to mix things up:

> FestivalData.stack.clean<-FestivalData.stack[!is.na(FestivalData.stack$value),]

Now we can sort out the data:

>FestivalData.stack.clean <- FestivalData.stack.clean[order(FestivalData.stack.clean$value),]

We can then check the values by looking at the bottom values for instance:



>tail(FestivalData.stack.clean)

And we can eliminate the value(s):

> FestivalData.stack.clean2<-FestivalData.stack.clean[FestivalData.stack.clean$value<20,]

To check:

>tail(FestivalData.stack.clean2)

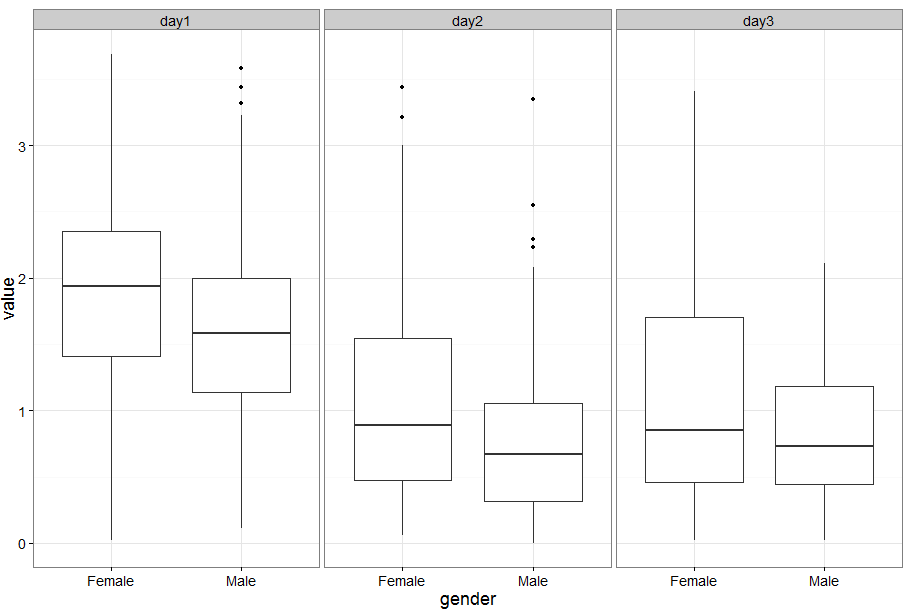
Or if we know it is a typo, we can correct it:

> FestivalData.stack.clean [1197,4]<-2.02

We can then check how our graphs look like now.

> Boxplot<-ggplot(FestivalData.stack.clean2, aes(gender,value))

+geom\_boxplot()+facet\_grid(~day)

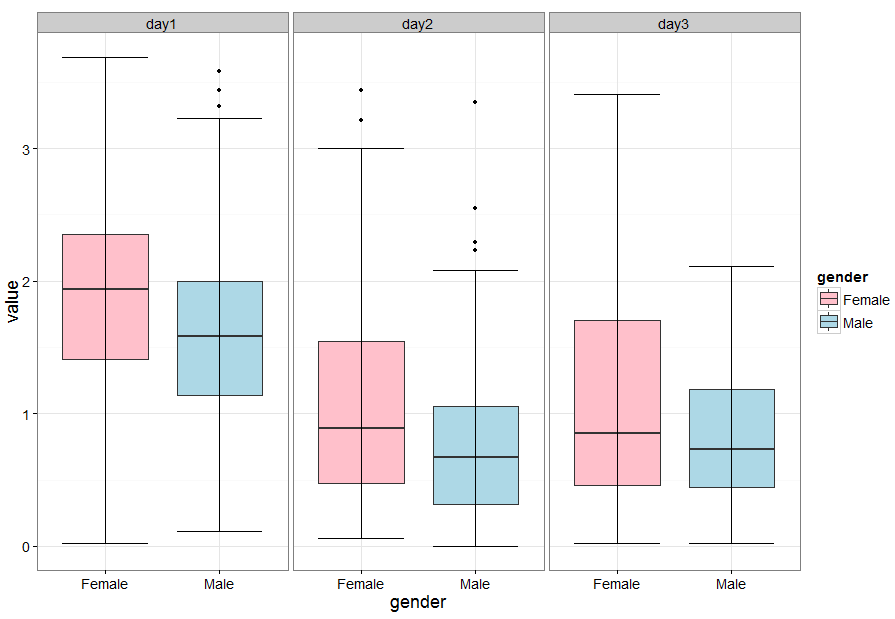


If we want to make the boxplots look prettier like play with the whiskers for instance, we can do:

> Boxplot<-ggplot(FestivalData.stack.clean2, aes(gender,value, fill=gender))

+geom\_boxplot()+stat\_boxplot(geom="errorbar")

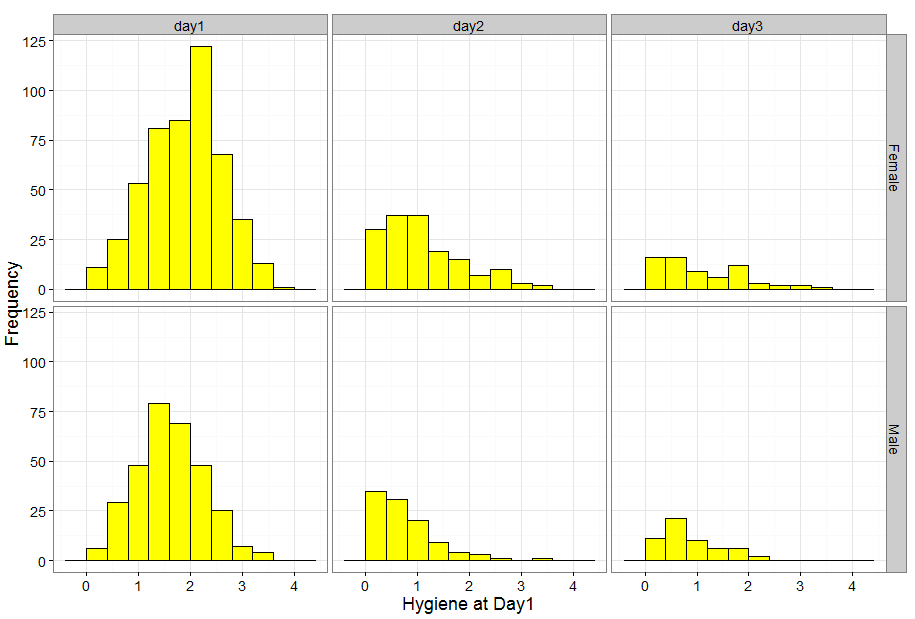
+scale\_fill\_manual(values=c("pink","lightblue"))+facet\_grid(~day)



If you want to change the order of the levels on the x-axis:

>Boxplot.festival+scale\_x\_discrete(limits=c("Male","Female"))

And the histograms would look like that without the outlier:



### Bar charts

Now that you have explored your data, you might want to plot in a more ‘classical’ way, like a bar chart.

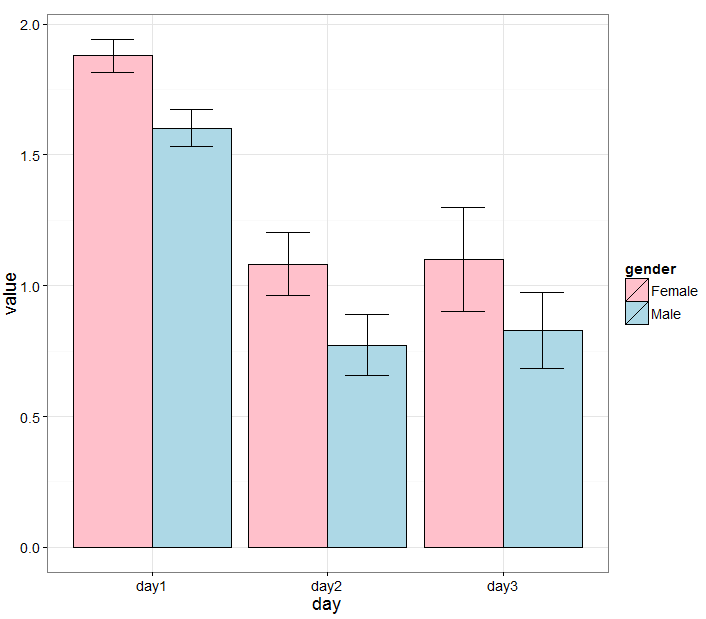
>BarFestival<-ggplot(FestivalData.stack.clean2, aes(day,value,fill=gender))

+stat\_summary(fun.y=mean, geom="bar", position="dodge", color="black") +stat\_summary(fun.data=mean\_cl\_normal, geom="errorbar", position=position\_dodge(width=0.9), width=0.5)

You recognise the first part of the command ggplot() in which you state which file you use and which variables you are using from that file. Now, a bar chart does not plot the raw data like the histogram and the box plot do. R needs to calculate the mean and the errorbars hence the stat\_summary functions. With the first one, you state which stat you want (mean), how you want to plot it (geom="bar") and the fact that you want the bars next to one another (position="dodge").With the second, you do the same for the error bars except for the dodge bit. If we want to specify the exact position or the width with dodge, we need to write it like that position=position\_dodge().

And if we want to change the default colours, we can go:

+scale\_fill\_manual(values=c("pink", "lightblue"))



### Scatterplots

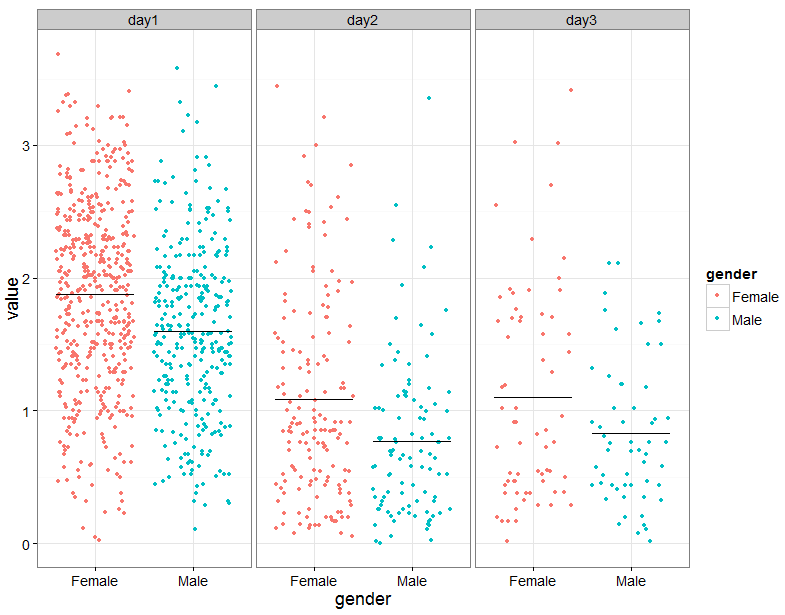
For scatterplots, we start with the same command as before. We then tell R we want to plot points and we add the jitter option so that they appear randomly scattered and not in one column. We finally add the facet\_grid function to see the 3 days separately.

>Scatterplot<-ggplot(FestivalData.stack.clean2, aes(gender, value, colour=gender))+geom\_point(position="jitter")+facet\_grid(~day)

If we want to add a line for the mean, we then add:

+geom\_errorbar(stat= "hline", yintercept="mean", width=0.8, , colour="black", aes(ymax=..y.., ymin=..y..))

### With the command above, you are basically telling R that you want to plot error bars (geom\_errorbar) but as a horizontal line (hline) which y-coordinate will be “mean” and the error bars themselves will also be the mean y.



### 

### Line graphs

Finally we are going to draw a line graph.

Once again we are going to tell R what type of graph we want to draw from which file and from which variable from that file. We will then add layers to plot the means as points stat\_summary(fun.y=mean,geom="point"), then we will tell R to join those points stat\_summary(fun.y=mean,geom="line")and finally we will add some error bars stat\_summary(fun.data=mean\_cl\_normal, geom="errorbar", width=0.2).

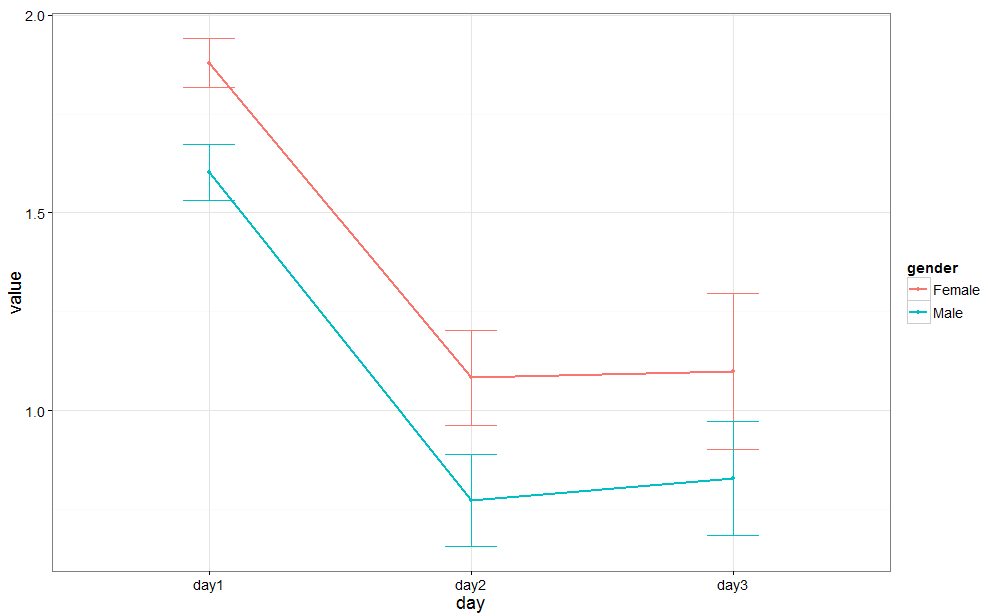
>Linegraph<-ggplot(FestivalData.stack.clean2,aes(day,value,colour=gender, group=gender))

+stat\_summary(fun.y=mean,geom="point")

+stat\_summary(fun.y=mean,geom="line")

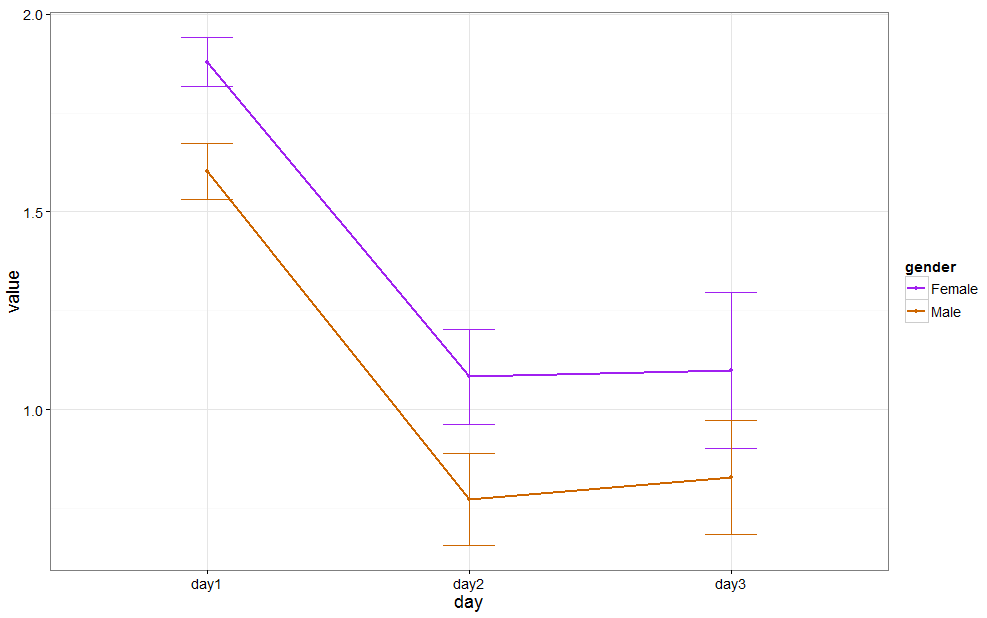
+stat\_summary(fun.data=mean\_cl\_normal, geom="errorbar", width=0.2)

>Linegraph



And if we don’t like the colour, we can change it by adding the layer:

+scale\_colour\_manual(values=c("purple","darkorange4"))



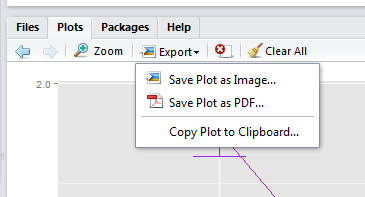
To save a graph, for instance the one we just drew, we go:

>Linegraph.saved<-ggsave(Linegraph, file="Line.png")

### 

### Exporting graphs from RStudio

Graphs can be exported from RStudio using the item Export from the menu bar in the graphics window. The graph can be saved as a pdf or as a range of image formats. The size of the graph can be altered.



# Chapter 3: Qualitative data

Package needed: *gmodels* and *ggplot2*

Let’s talk about the important stuff: your data. The first thing you need to do good stats is to know your data inside out. They are generally organised into variables, which can be divided into 2 categories: *qualitative* and *quantitative.*

Qualitative data are non-numerical data and the values taken can be names (also *nominal* data, e.g. variable sex: male or female). The values can also be numbers but not numerical (e.g. an experiment number is a numerical label but not a unit of measurement). A qualitative variable with intrinsic order in their categories is *ordinal*. Finally, there is the particular case of qualitative variable with only 2 categories, it is then said to be *binary* or *dichotomous* (e.g. alive/dead or male/female).

We are going to use an example to go through the analysis and the plotting of categorical data.

*Example* (File: cats.dat)

A researcher is interested in whether cats could be trained to line dance. He tries to train them to dance by giving them either food or affection as a reward (**training**) for dance-like behaviour. At the end of the week a note is made of which animal could line dance and which could not (**dance**). All the variables are dummy variables (categorical).

The pivotal (!) question is: Is there an effect of training on cats’ ability to learn to line dance? You have already designed your experiment and chosen your statistical test: it will be a Fisher’s exact test (or a Chi-square)

### Power Analysis with qualitative data

The next step is to run a power analysis. In an ideal world, you would have run a pilot study to get some idea of the type of effect size you are expecting to see. Let’s start with this ideal situation. Let’s say, in your pilot study, you found that 30% of the cats did line dance after received affection and 70% did so after received food.

Using G\*Power (see below), you should follow a 4 steps approach.

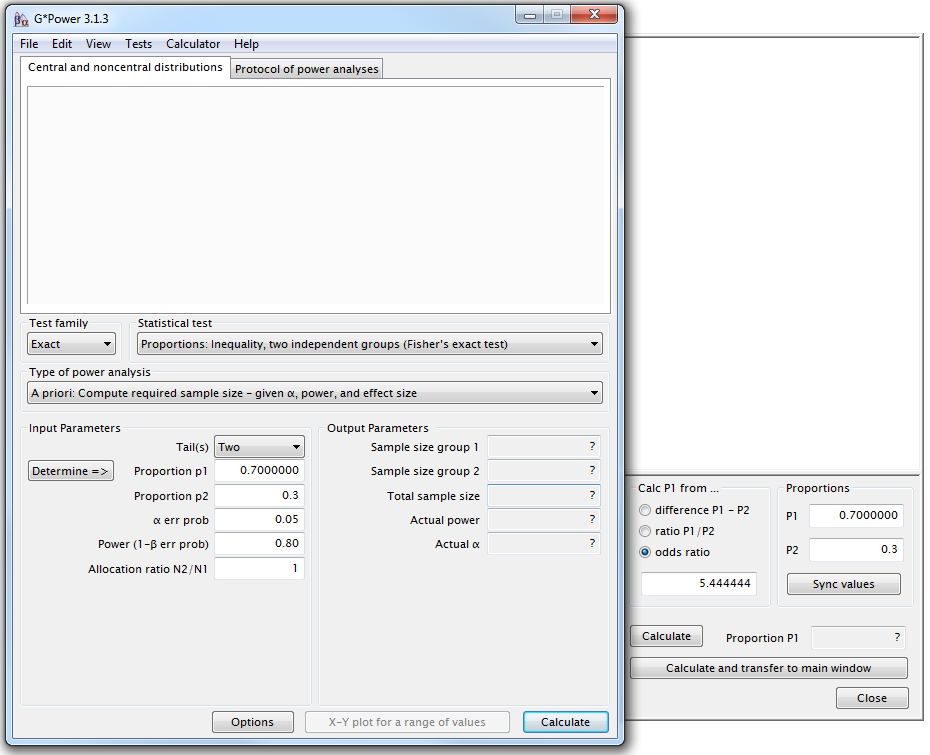
-Step 1: the Test family. You are going for the Fisher’s exact test, you should go for ‘Exact’.

-Step 2: the Statistical Test: you are looking at proportions and you want to compare 2 independent group.

-Step 3: the Type of Power Analysis: you know your significant threshold (α=0.05), the power you are aiming for (80%), you have the results from the pilot study so you can calculate the effect size: you go for an ‘A Priori’ analysis.

-Step 4: the tricky one, you need to Input Parameters. Well, it is the tricky one when you have no idea of the effect size but in this case you are OK. Plus if you enter the results for the pilot study, G\*Power calculates the effect size for you.

So if you do all that, G\*Power will tell you that you need 2 samples of 29 cats to reach a power of 80%. In other words: if you want to be at least 80% sure to spot a treatment effect, if indeed there is one, you will need about 60 cats altogether .

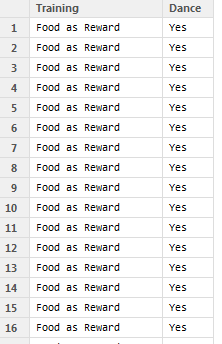


Now back to R, there are 2 ways categorical data can be entered: as raw data or as weighted cases. We will look at both.

The raw scores

The data are in the file “cats.dat”. So to open the file, we will type the command:

>setwd("M:/Work/R Course/Exercises/") to tell R where the file is, then:

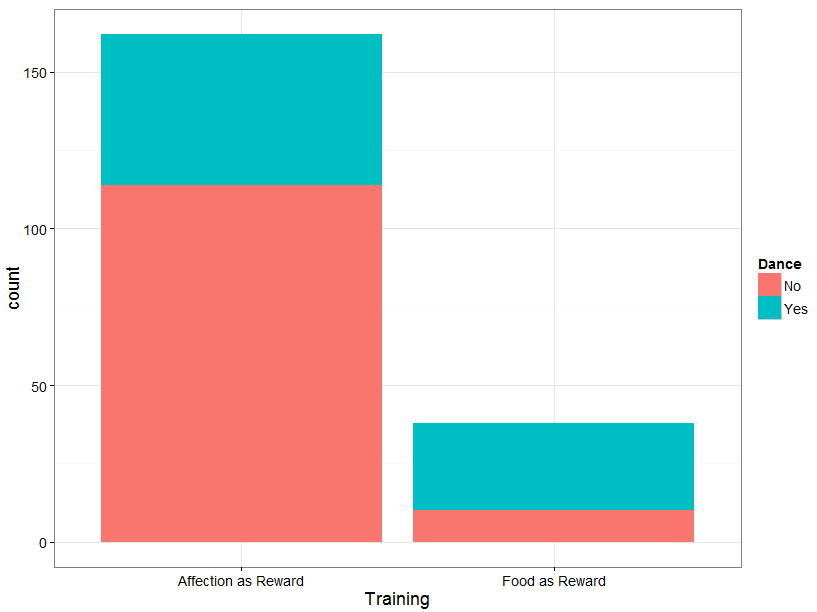
>cat.data<-read.table("cats.dat", sep="\t", header=T) to open the file and finally if we want to see how the data look like:

>View(cat.data)

To plot the data, remember we need to tell *ggplot2* from which data.frame we want to use the data (cat.data), then we need to tell which variable we want to use so remember these are called aesthetics in *ggplot2* so we type (aes(x=Training, fill=Dance)). That’s our first layer, we just told *ggplot2*, we want to draw a graph. Now we need to say which type of graph so that’s geom\_bar(). If we just type that we will get the graph below.

>bar.chart<-ggplot(cat.data, aes(x=Training, fill=Dance))+geom\_bar()

>bar.chart

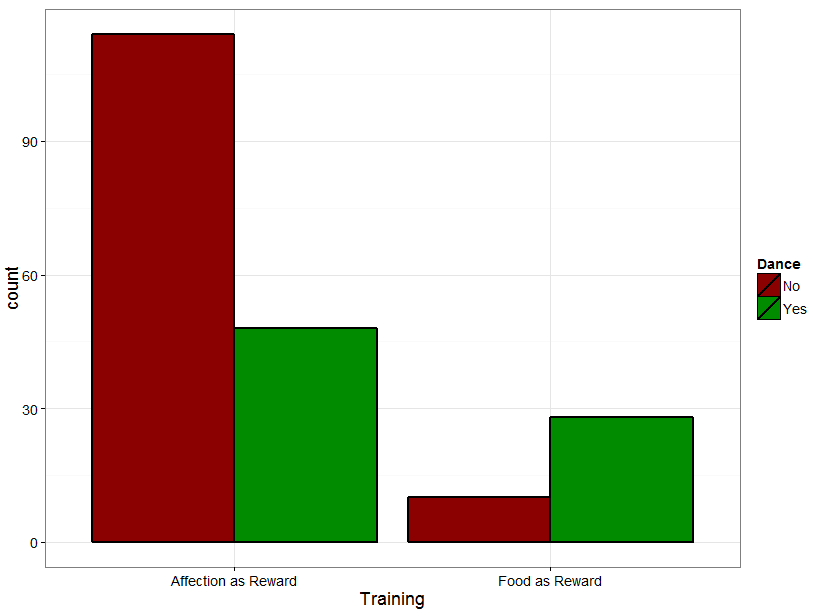


If we want to personalize it, we could do for example:

>bar.chart<-ggplot(cat.data,aes(x=Training,fill=Dance)) +geom\_bar(colour="black",size=1,position="dodge")

+scale\_fill\_manual(values=c("red4","green4"))

>bar.chart



Now the problem here is that we are plotting the counts and not the frequencies which can produce ugly graphs if the size of the groups is imbalanced like here. To plot the frequencies we first need to create a contingency table from your data.

>contingency.table<-xtabs(~Dance+Training,cat.data)

>contingency.table



We then ask R to transform the values into frequencies.

>contingency.table100<-prop.table(contingency.table,2)

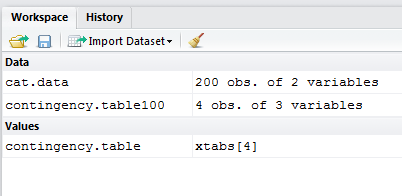
>contingency.table100\*100

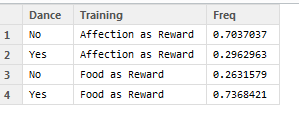
That will give we the column percentages; for the row we would type: (contingency.table,1)



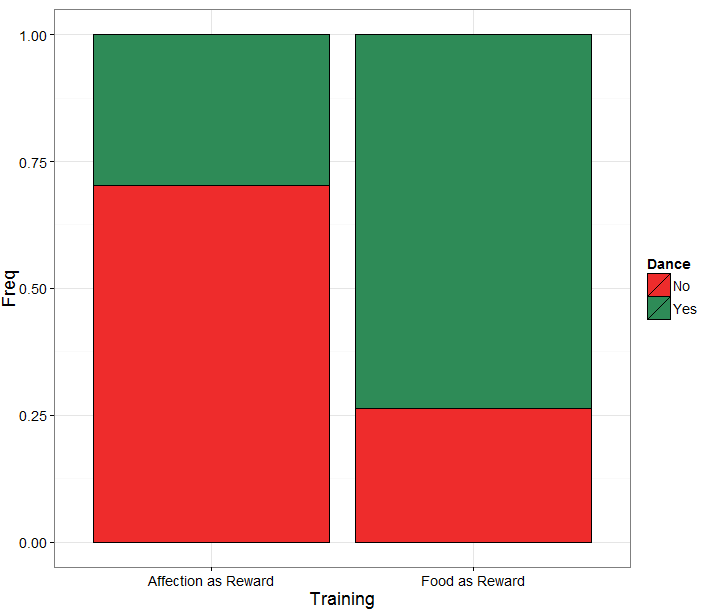
Finally, before plotting the graph it’s best to transform the table into a data frame as *ggplot2* likes data frames.

>contingency.table100<-as.data.frame(contingency.table100)



You will see a new item in the workspace under data.

And now you can draw your graph:



>bar.contingency100<-ggplot(contingency.table100,aes(Training,Freq, fill=Dance))

+stat\_summary(fun.y=identity,geom="bar", position="stack", color="black")

+scale\_fill\_manual(values=c("firebrick2","seagreen4"))

>bar.contingency100

As mentioned before, to analyse such data you need to use a Fisher’s exact test but we could also use a 2 test.

Both tests will give we the same-ish p-value for big samples but for small samples the difference can be a bit more important and the p-value given by Fisher’s exact test is more accurate. Having said that, the calculation of the Fisher’s exact test is quite complex whereas the one for 2 is quite easy so only the calculation of the latter is going to be presented here. Also, the Fisher’s test is often only available for 2x2 tables, so in a way the 2 is more general.

For both tests, the idea is the same: how different are the observed data from what we would have expected to see by chance i.e. if there were no association between the 2 variables. Or, looking at the table we may also ask: knowing that 76 of the 200 cats did dance and that 162 of them received affection, what is the probability that those 76 dancers would be so unevenly distributed between the 2 types of reward?

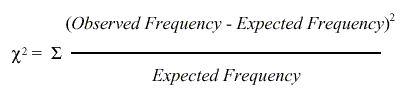
## A bit of theory: the Chi2 test

It could be either:

* a one-way 2 test, which is basically a test that compares the observed frequency of a variable in a single group with what would be the expected by chance.
* a two-way 2 test, the most widely used, in which the observed frequencies for two or more groups are compared with expected frequencies by chance. In other words, in this case, the 2 tells you whether or not there is an association between 2 categorical variables.

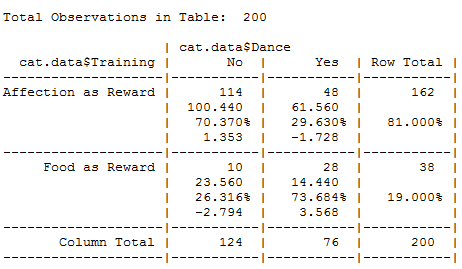
An important thing to know about the 2, and for the Fisher’s exact test for that matter, is that it does not tell we anything about causality; it is simply measuring the strength of the association between 2 variables and it is your knowledge of the biological system we are studying which will help we to interpret the result. Hence, we generally have an idea of which variable is acting the other.

The Chi2 value is calculated using the formula below:



The observed frequencies are the one we measured, the values that are in our table. Now, the expected ones are calculated this way:

Expected frequency = (row total)\*(column total)/grand total

****

So, for example: the expected frequency of cats that would line dance after having received food as reward is: (76\*38)/200=14.44

We can also think in terms of probabilities:

* probability of line dancing: 38/200
* probability of receiving food: 76/200

If the 2 events are independent, the probability of the 2 occurring at the same time (the expected frequency) will be: 38/200\*76/200 = 0.072 and 7.2% of 200 is 14.4.

Intuitively, one can see that we are kind of averaging things here, we try to find out the values we should have got by chance. If you work out the values for all the cells, you get:

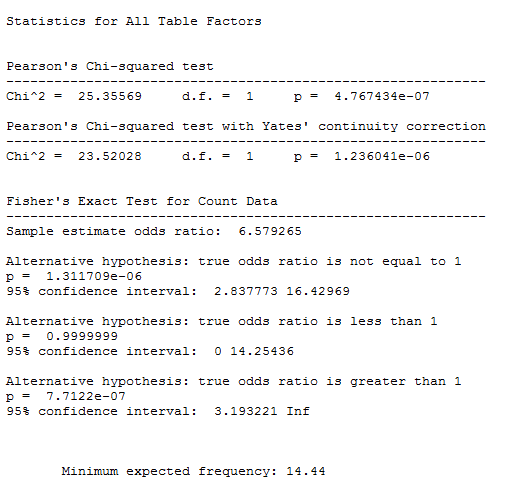
To run a Chi2 analysis, we are going to use the CrossTable() function in the *gmodels* package.

Now CrossTable() is definitely a function for which you would want to change the default settings.

To do that, type:

>?CrossTable and see what the settings are.

>CrossTable(cat.data$Training,cat.data$Dance,expected=T,chisq=T,fisher=T, sresid=T, prop.c=F, prop.r=T,prop.t=F, prop.chisq=F,format=c("SPSS"))



So for the cat, the 2 value is:

(114-100.4)2/100.4 + (48-61.6)2/61.6 + (10-23.6)2 /23.6 + (28-14.4)2/14.4 = 25.35

There is only one assumption that you have to be careful about when you run a 2:with 2x2 contingency tables you should not have cells with an expected count below 5 as if it is the case it is likely that the test is not accurate (for larger tables, all expected counts should be greater than 1 and no more than 20% of expected counts should be less than 5).

If you have a high proportion of cells with a small value in it, then you should use a Fisher’s exact test. So when you have more than 2 categories and a small sample you are in trouble. You have 2 solutions to solve the problem: either you collect more data or you group the categories to boost the proportions.

If you remember the 2’s formula, the calculation gives you estimation of the difference between your data and what you would have obtained if there was no association between your variables. Clearly, the bigger the value of the 2, the bigger the difference between observed and expected frequencies and the more likely the difference is to be significant.

As you can see here the p-values vary slightly between the 2 tests (p=1.31e-06 vs.p=4.776-07) though the conclusion remains the same: cats only care about food. Though the samples are not very big here, the assumption for the 2 is met so you can choose either test.

## A bit of theory: the null hypothesis and the error types.

The null hypothesis (H0) corresponds to the absence of effect (e.g.: the animals rewarded by food are as likely to line dance as the ones rewarded by affection) and the aim of a statistical test is to accept or to reject H0. As mentioned earlier, traditionally, a test or a difference are said to be “significant” if the probability of type I error is: α =< 0.05 (max α=1). It means that the level of uncertainty of a test usually accepted is 5%. It also means that there is a probability of 5% that you may be wrong when you say that your 2 means are different, for instance, or you can say that when you see an effect you want to be at least 95% sure that something is significantly happening.

|  |  |  |
| --- | --- | --- |
| Statistical decision | True state of H0 | |
| H0 True (no effect) | H0 False (effect) |
| Reject H0 | Type I error (False Positive) | Correct (True Positive) |
| Do not reject H0 | Correct (True Negative) | Type II error (False Negative) |

Tip: if your p-value is between 5% and 10% (0.05 and 0.10), I would not reject it too fast if I were you. It is often worth putting this result into perspective and asks yourself a few questions like:

* what the literature says about what am I looking at?
* what if I had a bigger sample?
* have I run other tests on similar data and were they significant or not?

The interpretation of a border line result can be difficult so it is important to look at the whole picture.

The specificity and the sensitivity of a test are closely related to Type I and Type II errors.

**Specificity** = Number of True Negatives / (Number of False Positives + Number of True Negatives). A test with a high specificity has a low [type I error](http://en.wikipedia.org/wiki/Type_I_and_type_II_errors) rate.

**Sensitivity** = Number of True Positives / (Number of False Negatives + Number of True Positives). A test with a high sensitivity has a low [type II error](http://en.wikipedia.org/wiki/Type_I_and_type_II_errors) rate.

# Chapter 4: Quantitative data

Packages needed: *car*, *pastecs*, *psych, fBasics, multcomp, reshape, reshape2* and of course *ggplot2* to draw the graphs.

When it comes to quantitative data, more tests are available but assumptions must be met before applying them. In fact, there are 2 types of stats tests: parametric and non-parametric ones. Parametric tests have 4 assumptions that must be met for the tests to be accurate. Non-parametric tests are based on ranks and they make few or no assumptions about populations parameters like normality (e.g. Mann-Whitney test).

## 3-1 A bit of theory: descriptive stats

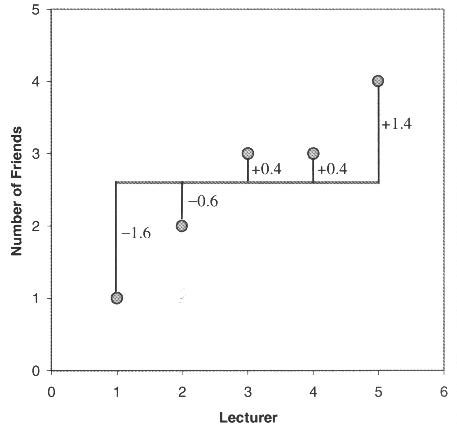
The mean (or average) µ = average of all values in a column

It can be considered as a model because it summaries the data.

- Example: number of friends of each members of a group of 5 lecturers: 1, 2, 3, 3 and 4

Mean: (1+2+3+3+4)/5 = 2.6 friends per lecturer: clearly an hypothetical value!

Now if the values were: 1, 1, 2, 4 and 5 the mean would also be 2.6 but clearly it would not give an accurate picture of the data. So, how can you know that it is an accurate model? You look at the difference between the real data and your model. To do so, you calculate the difference between the real data and the model created and you make the sum so that you get the total error (or sum of differences).



∑(xi - µ) = (-1.6) + (-0.6) + (0.4) + (0.4) + (1.4) = 0 And you get no errors !

Of course: positive and negative differences cancel each other out. So to avoid the problem of the direction of the error, you can square the differences and instead of sum of errors, you get the Sum of Squared errors (SS).

- In our example: SS = (-1.6)2 + (-0.6)2 + (0.4)2 + (0.4)2 + (1.4)2 = 5.20

The median: The median is the value exactly in the middle of an ordered set of numbers.

Example 1: 18 27 34 52 54 59 61 68 78 82 85 87 91 93 100, Median = 68

Example 2: 18 27 27 34 52 52 59 61 68 68 85 85 85 90, Median = 60

### The variance

This SS gives a good measure of the accuracy of the model but it is dependent upon the amount of data: the more data, the higher the SS. The solution is to divide the SS by the number of observations (N). As we are interested in measuring the error in the sample to estimate the one in the population, we divide the SS by N-1 instead of N and we get the *variance* (S2) = SS/N-1

- In our example: Variance (S2) = 5.20 / 4 = 1.3

Why N-1 instead N?

If we take a sample of 4 scores in a population they are free to vary but if we use this sample to calculate the variance, we have to use the mean of the sample as an estimate of the mean of the population. To do that we have to hold one parameter constant.

- Example: mean of the sample is 10

We assume that the mean of the population from which the sample has been collected is also 10. If we want to calculate the variance, we must keep this value constant which means that the 4 scores cannot vary freely:

- If the values are 9, 8, 11 and 12 (mean = 10) and if we change 3 of these values to 7, 15 and 8 then the final value must be 10 to keep the mean constant.

- If we hold 1 parameter constant, we have to use N-1 instead of N.

- It is the idea behind the *degree of freedom*: one less than the sample size.

### The Standard Deviation (SD)

The problem with the variance is that it is measured in squared units which is not very nice to manipulate. So for more convenience, the square root of the variance is taken to obtain a measure in the same unit as the original measure: the *standard deviation.*

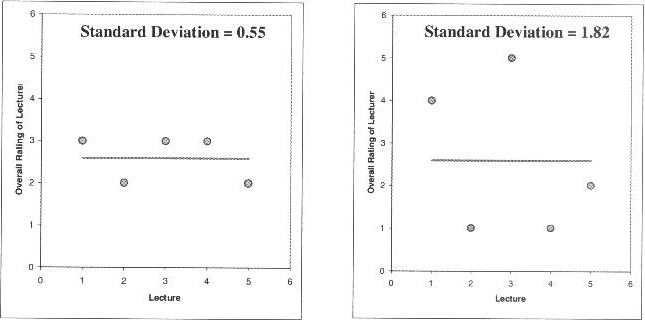
- S.D. = √(SS/N-1) = √(S2), in our example: S.D. = √(1.3) = 1.14

- So you would present your mean as follows: µ = 2.6 +/- 1.14 friends

The standard deviation is a measure of how well the mean represents the data or how much your data are scattered around the mean.

- small S.D.: data close to the mean: mean is a good fit of the data (graph on the left)

- large S.D.: data distant from the mean: mean is not an accurate representation (graph on the right)

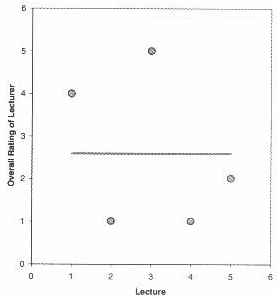
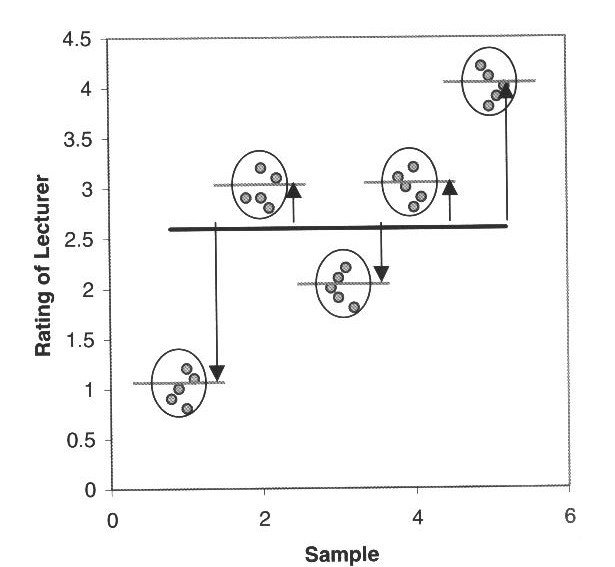


### Standard Deviation vs. Standard Error

Many scientists are confused about the difference between the standard deviation (S.D.) and the *standard error of the mean* (S.E.M. = S.D. / √N).

- The S.D. (graph on the left) quantifies the scatter of the data and increasing the size of the sample does not decrease the scatter (above a certain threshold).

- The S.E.M. (graph on the right) quantifies how accurately you know the true population mean, it’s a measure of how much you expect sample means to vary. So the S.E.M. gets smaller as your samples get larger: the mean of a large sample is likely to be closer to the true mean than is the mean of a small sample.



A big S.E.M. means that there is a lot of variability between the means of different samples and that your sample might not be representative of the population.

A small S.E.M. means that most samples means are similar to the population mean and so your sample is likely to be an accurate representation of the population.

**Which one to choose?**

- If the scatter is caused by biological variability, it is important to show the variation. So it is more appropriate to report the S.D. rather than the S.E.M. Even better, you can show in a graph all data points, or perhaps report the largest and smallest value.

- If you are using an in vitro system with theoretically no biological variability, the scatter can only result from experimental imprecision (no biological meaning). It is more sensible then to report the S.E.M. since the S.D. is less useful here. The S.E.M. gives your readers a sense of how well you have determined the mean.

Choosing between SD and SEM also depends on what you want to show. If you just want to present your data on a descriptive purpose then you go for the SD. If you want the reader to be able to infer an idea of significance then you should go for the SEM or the Confidence Interval (see below). We will go a bit more in details later.

### Confidence interval

The confidence interval quantifies the uncertainty in measurement. The mean you calculate from your sample of data points depends on which values you happened to sample. Therefore, the mean you calculate is unlikely to equal the true population mean. The size of the likely discrepancy depends on the variability of the values and the sample size. If you combine those together, you can calculate a 95% confidence interval (95% CI), which is a range of values. If the population is normal (or nearly so), you can be 95% sure that this interval contains the true population mean.

95% of observations in a normal distribution lie within +/- 1.96\*SE

One other way to look at error bars:

|  |  |  |
| --- | --- | --- |
| **Error bars** | **Type** | **Description** |
| Standard deviation (SD) | Descriptive | Typical or average difference between the data points and their mean. |
| Standard error (SEM) | Inferential | A measure of how variable the mean will be, if you repeat the whole study many times. |
| Confidence interval (CI), usually 95% CI | Inferential | A range of values you can be 95% confident contains the true mean. |

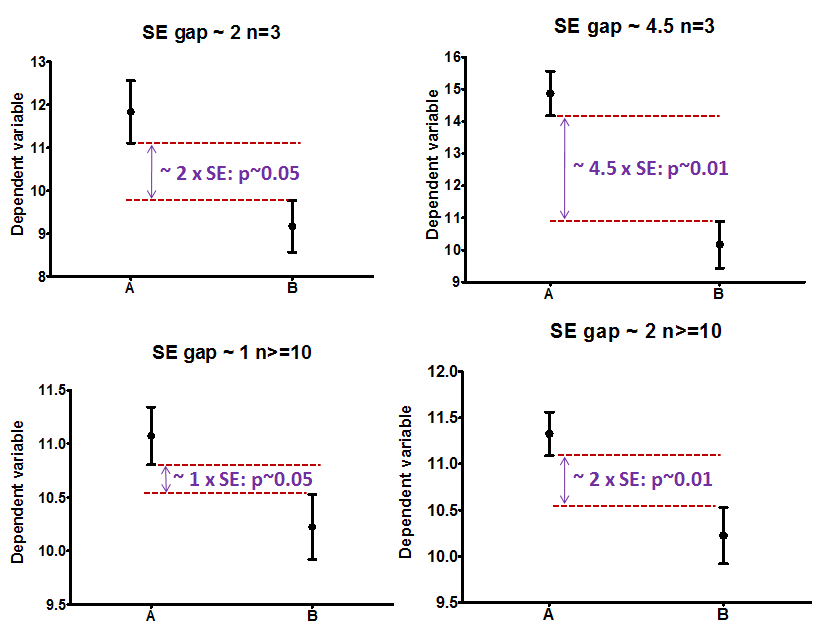
From Geoff Cumming *et al*.

If you want to compare experimental results, it could be more appropriate to show inferential error bars such as SE or CI rather than SD.

However if n is very small (for example n=3), rather than showing error bars and statistics, it is better to simply plot the individual data points.

*In the same way, you can estimate statistical significance using the overlap rule for 95% CI bars.*

*In the same way, you can estimate statistical significance using the overlap rule for 95% CI bars.*

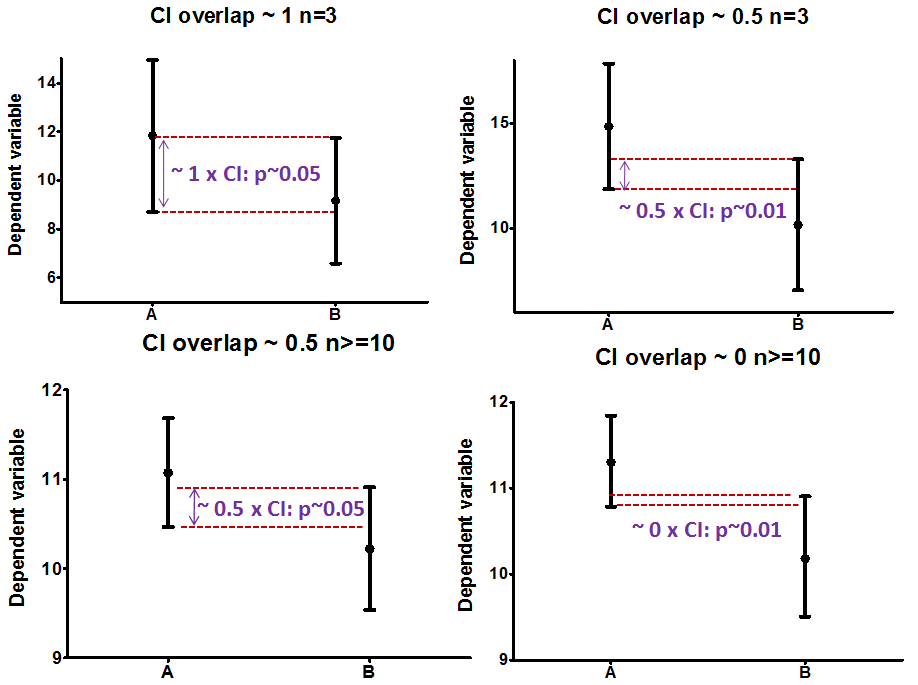


*You can estimate statistical significance using the overlap rule for SE bars.*

*In the same way, you can estimate statistical significance using the overlap rule for 95% CI bars.*

*In the same way, you can estimate statistical significance using the overlap rule for 95% CI bars.*

*In the same way, you can estimate statistical significance using the overlap rule for 95% CI bars.*



*In the same way, you can estimate statistical significance using the overlap rule for 95% CI bars.*

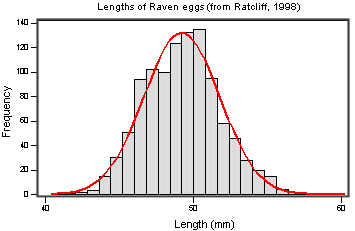
## 3-2 A bit of theory: Assumptions of parametric data

When you are dealing with quantitative data, the first thing you should look at is how they are distributed, how they look like. The distribution of your data will tell you if there is something wrong in the way you collected them or enter them and it will also tell you what kind of test you can apply to make them say something.

T-test, analysis of variance and correlation tests belong to the family of parametric tests and to be able to use them your data must comply with 4 assumptions.

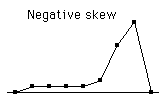
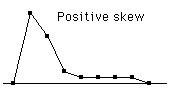
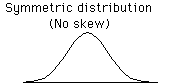
1) The data have to be normally distributed (normal shape, bell shape, Gaussian shape).

Example of normally distributed data:



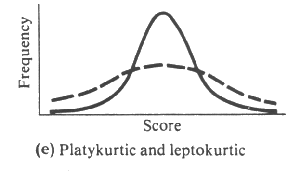
There are 2 main types of departure from normality:

- Skewness: lack of symmetry of a distribution



- Kurtosis: measure of the degree of peakedness in the distribution

The two distributions below have the same variance approximately the same skew, but differ markedly in kurtosis.

2) Homogeneity in variance: The variance should not change systematically throughout the data.

3) Interval data: The distance between points of the scale should be equal at all parts along the scale

4) Independence: Data from different subjects are independent so that values corresponding to one subject do not influence the values corresponding to another subject. Basically, it means it means one measure per subject. There are specific designs for repeated measures experiments.

### How can you check that your data are parametric/normal?

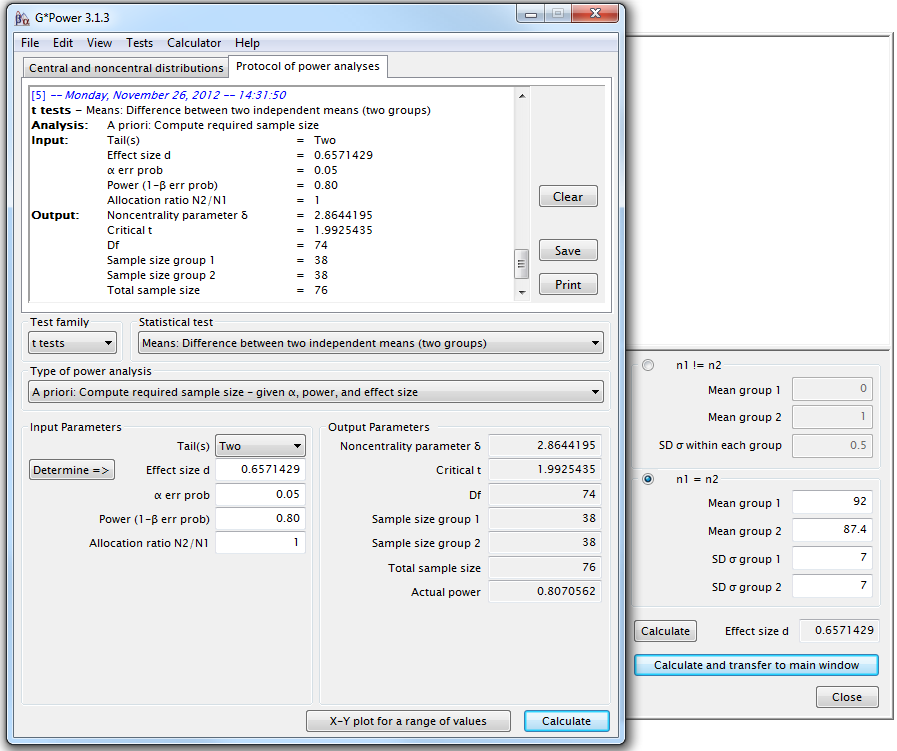
Let’s try it through an example.

*Example* (File: coyote.csv) csv as in ‘comma-separated values’

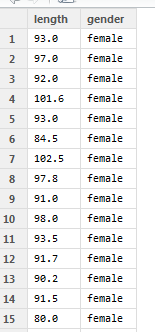
You want to know if male coyotes are bigger than female coyotes. Of course, before doing anything else, you design your experiment and you are told that to compare 2 samples you need to apply a t-test (we will explain this test later). So basically you are going to catch coyotes and hopefully you will manage to catch males and females. When your samples are big enough, you compare them. Now, the tricky bit here is the ‘big enough’ bit.

### Power analysis with a t-test

Let’s say, you don’t have data from a pilot study but you have found some information in the literature. In a study run in similar conditions as in the one you intend to run, male coyotes were found to measure: 92cm+/- 7cm (SD). You expect a 5% difference between genders with a similar variability in the female sample.



We need a sample size of n=76 (2\*38).



Once the data are collected, we check for normality.

Though normality tests are good, the best way to get a real good idea of what is going on is to plot our data. When it comes to normality, there are 3 ways to plot our data: the box plot, the scatter plot and the histogram. We are going to do them all with R.

To get the data in R we go:

> setwd("M:/Work/R Course/Exercises/")

> coyote<-read.csv("coyote.csv",header=TRUE)

> View(coyote) or head(coyote)

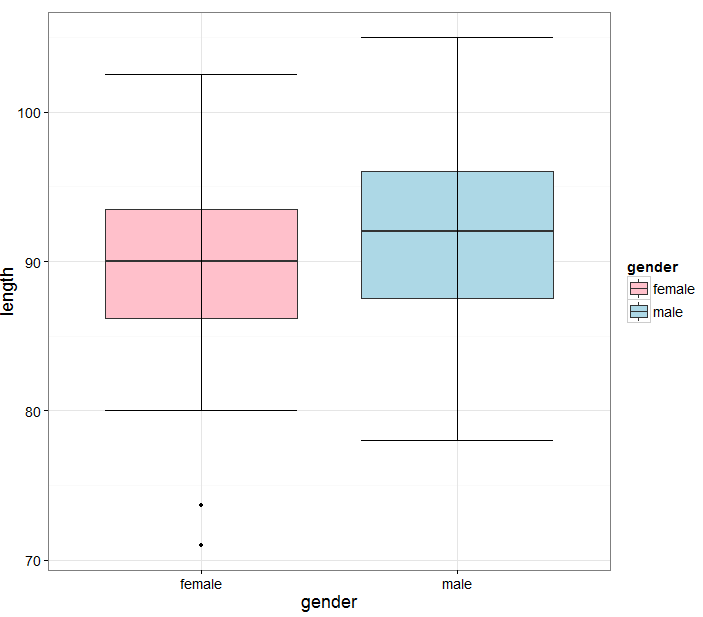
Now let’s start with the boxplot. With the command below we are telling R that we want to plot a boxplot with *ggplot2*, then we add a layer with stat\_boxplot() so that the whiskers look prettier and finally we choose the colour scale\_fill\_manual().

>boxplot.coyote<-ggplot(coyote,aes(gender,length,fill=gender)) +geom\_boxplot()

+stat\_boxplot(geom="errorbar",width=0.1)

+scale\_fill\_manual(values=c("pink", "lightblue"))

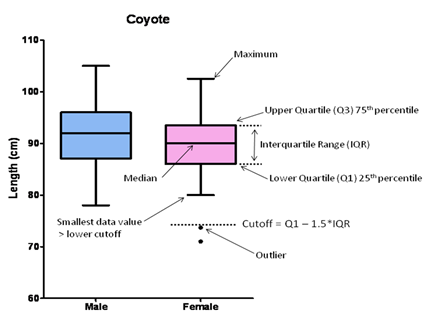
>boxplot.coyote



The anatomy of a boxplot is explained in the graph below.

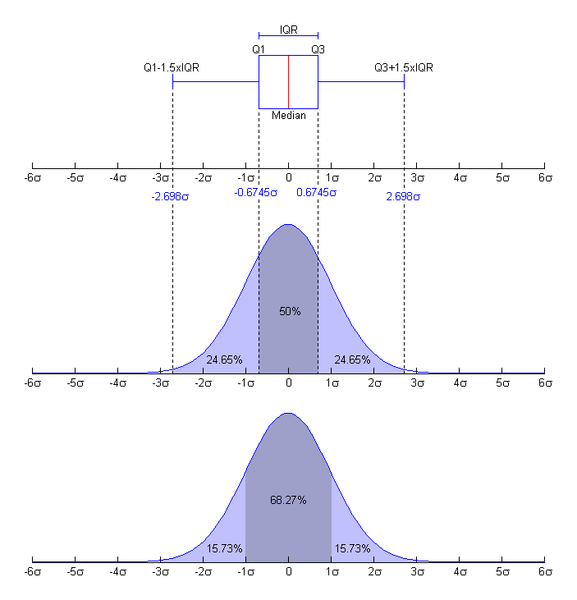
It is very important that you know how a box plot is built. It is rather simple and it will allow you to get a pretty good idea about the distribution of your data at a glance.

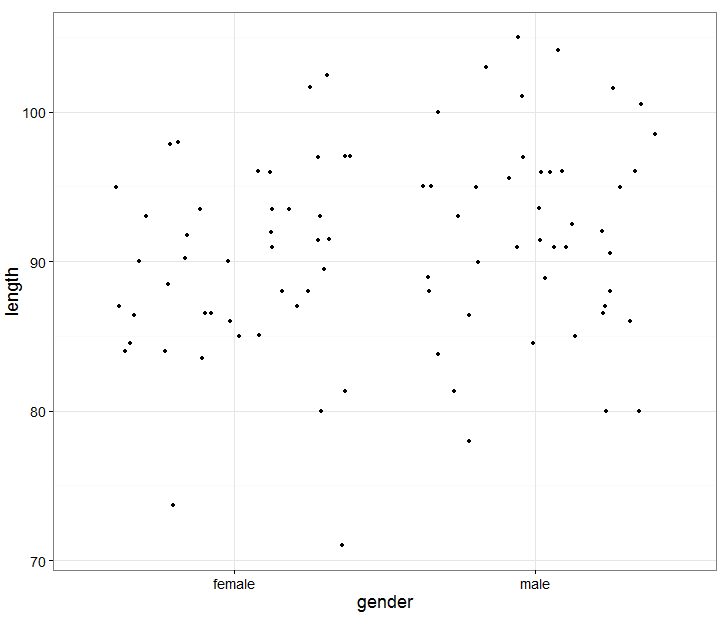
If the distribution is normal-ish then the box plot should be symmetrical-ish and if both (like in our case) are of the same size-ish, then we can be confident that the variances are about the same.



Below you can see the relationship between box plot and histogram.

Regarding the outliers, there is no really right or wrong attitude. If there is a technical issue or an experimental problem, you should remove it of course but if there is nothing obvious, it is up to you. I would always recommend keeping outliers if you can; you can run the analysis with and without it for instance and see what effect it has on the p-value. If the outcome is still consistent with your hypothesis, then you should keep it. If not, then it is between you and your conscience!

[](http://upload.wikimedia.org/wikipedia/commons/8/89/Boxplot_vs_PDF.png)



One other way to explore the data is the scatter plot.

>scatterplot.coyote<-ggplot(coyote,aes

(gender,length))

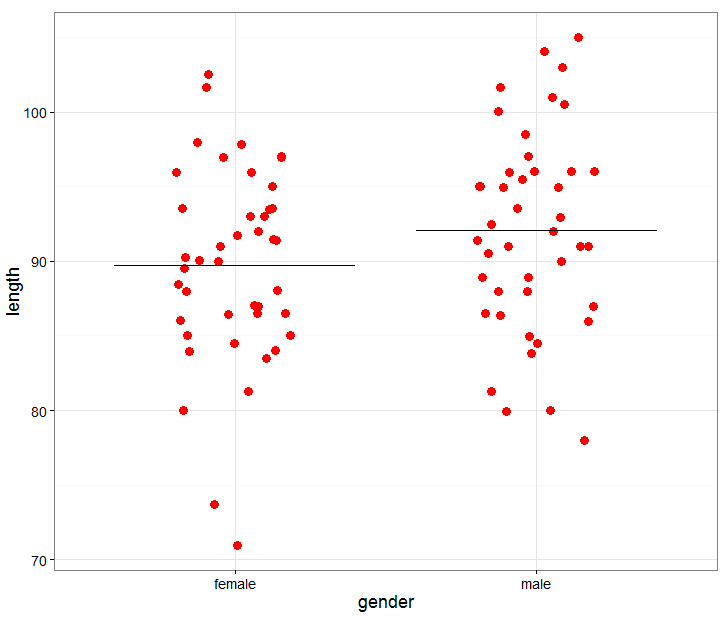
+geom\_point(position="jitter")

>scatterplot.coyote

Now you may want to improve this graph. For instance you may want to narrow the scatter or add a line for the mean or change the colour of the points. To do so you need to specify the width of the ‘jitter’, add a layer for the mean and specify the colour of the points:

>scatterplot.coyote<-ggplot(coyote,aes(gender,length)) +geom\_point(colour="red",position=position\_jitter(width=0.2),size=4)

+geom\_errorbar(stat= "hline", yintercept="mean", width=0.8, aes(ymax=..y.., ymin=..y..))



If you want 2 different colours for the genders:

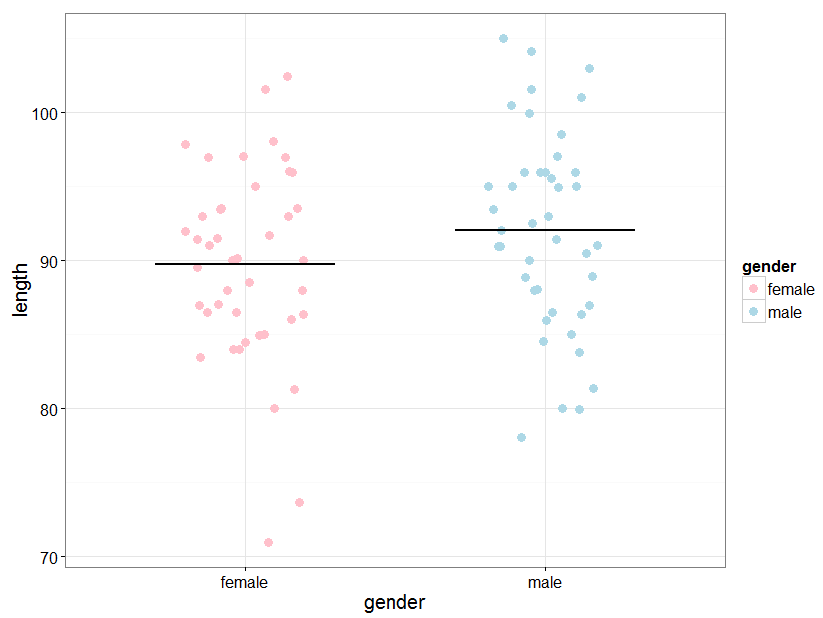
scatterplot.coyote<-ggplot(coyote, aes(gender, length, colour=gender))

+geom\_point(position=position\_jitter(width=0.2),size=4)

+geom\_errorbar(stat= "hline", colour="black",yintercept="mean", width=0.6, size=1,aes(ymax=..y.., ymin=..y..))

+scale\_colour\_manual(values=c("pink", "lightblue"))

scatterplot.coyote



Finally, the histograms.

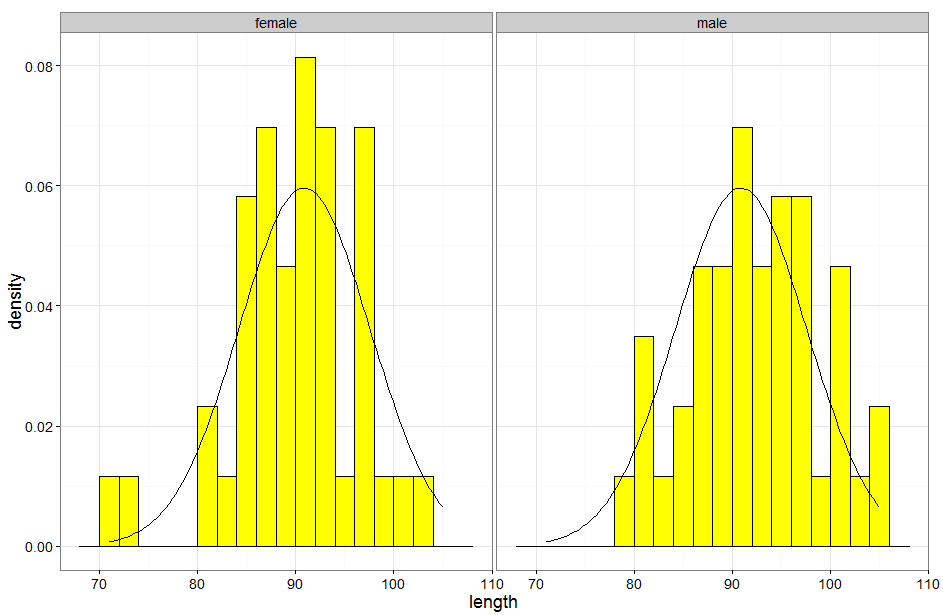
> histogram.coyote<-ggplot(coyote, aes(length))

>+geom\_histogram(aes(y=..density..),colour="black", fill="yellow",binwidth=2)

>+facet\_grid(.~gender)

And if you want to add the normality trend, you can add an extra layer:

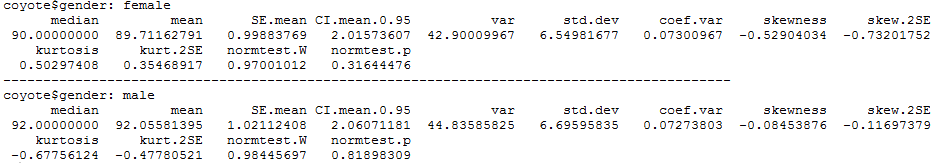
>+stat\_function(fun=dnorm,args=list(mean=mean(coyote$length,na.rm=TRUE),sd=sd(coyote$length,na.rm=TRUE))



So from the graphs we have plotted, we can tell that the first and second assumptions are likely to be met: the data seem normal enough (symmetry of the graphs) and the variability seems comparable between groups (spread fo the data). My preference goes to the box plot as it tells you in one go anything you need to know: where are you with the first 2 assumptions and it shows you outliers.

Still we may come across cases where it is not that obvious so can ask R to test for the normality (Shapiro-Wilk test or D’agostino and Pearson tests) and homogeneity of variance (Levene test). Here we are going to use 2 new functions: stat.desc() which gives a statistical summary of the data and test for normality (Shapiro-Wilk test ) and by() which allows to do it for males and females separately in one go.

> by(coyote$length,coyote$gender, stat.desc, basic=F, norm=T)

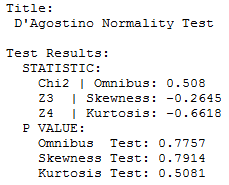
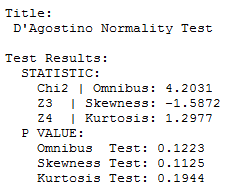


There is no significant departure from normality for females (p=0.316) or males (p=0.819).

Now the stat.desc() function calculates the Shapiro-Wilk test which is OK when all values are unique which is not the case with our dataset. The D’agostino and Pearson test does not have this problem and is also quite intuitive to understand. However it can only be run on samples n=<20. Basically, it first computes the [skewness (for asymmetry) and kurtosis](mk:@MSITStore:C:\Program%20Files\GraphPad\Prism%205\prism5.chm::/stat_skewness_and_kurtosis.htm) (for shape) and calculates how far each of these values differs from the values expected with a Gaussian distribution. It then computes a single p-value from the sum of these discrepancies.

You can apply the D’agostino and Pearson test with the following command:

>by(coyote$length,coyote$gender, dagoTest)



So we are good: females: p=0.1223 and males: p=0.7757.

That was the first assumption; now we can check the second assumption (homogeneity of variances) using the Levene test .

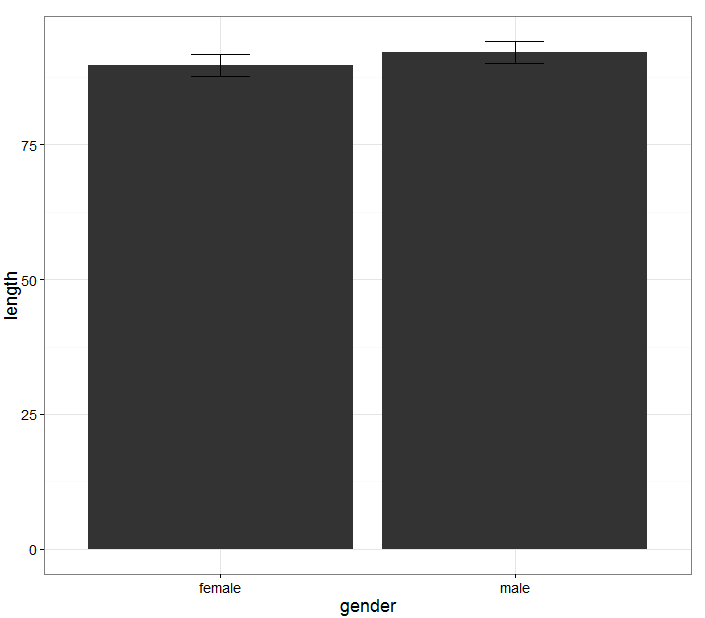
Second assumption:

>leveneTest(coyote$length, coyote$gender,center=mean)

****

So good again but not surprising news: the variances of the 2 genders do not differ significantly (p=0.698).

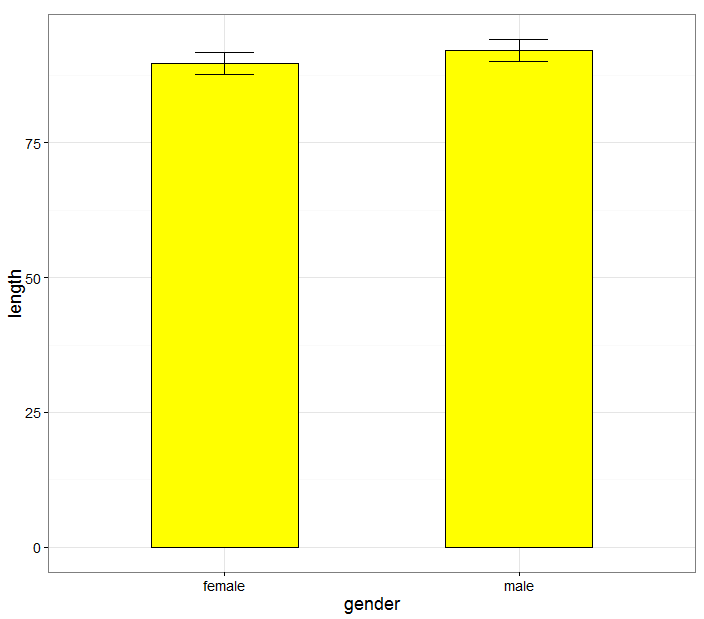
Don't be too quick to switch to nonparametric tests. While they do not assume Gaussian distributions, these tests do assume that the shape of the data distribution is the same in each group. So if your groups have very different standard deviations and so are not appropriate for a parametric test, they should not be analysed its non-parametric equivalent either. However parametric tests like ANOVA and t-tests are rather robust, especially when the samples are not too small so you can get away with small departure from normality and small differences in variances. Often the best approach is to transform the data and logarithms or reciprocals does the trick, restoring equal variance.

Finally we may want to represent the data as a classical bar chart. To achieve that, we can type the lines below:

>barchart.coyote<-ggplot(coyote, aes(gender, length)) +stat\_summary(fun.y=mean, geom="bar", aes(width=0.5))+stat\_summary(fun.data=mean\_cl\_normal, geom="errorbar", width=0.2)

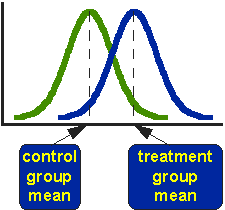
This is not the prettiest graph ever. We can improve that by adding a few things:

>barchart.coyote<-ggplot(coyote, aes(gender, length)) +stat\_summary(fun.y=mean, geom="bar", fill=”yellow”, colour=”black”, aes(width=0.5)) +stat\_summary(fun.data=mean\_cl\_normal, geom="errorbar", width=0.2)



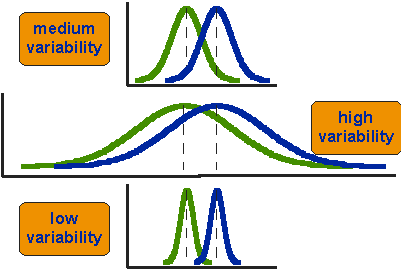
## 3-3 A bit of theory: the t-test

The t-test assesses whether the means of two groups are *statistically* different from each other. This analysis is appropriate whenever you want to compare the means of two groups.



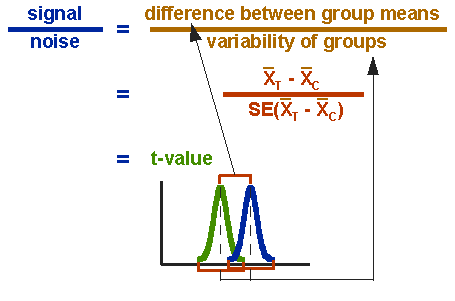
The figure above shows the distributions for the treated (blue) and control (green) groups in a study. Actually, the figure shows the idealized distribution. The figure indicates where the control and treatment group means are located. The question the t-test addresses is whether the means are statistically different.

What does it mean to say that the averages for two groups are statistically different? Consider the three situations shown in the figure below. The first thing to notice about the three situations is that the difference between the means is the same in all three. But, you should also notice that the three situations don't look the same -- they tell very different stories. The top example shows a case with moderate variability of scores within each group. The second situation shows the high variability case. The third shows the case with low variability. Clearly, we would conclude that the two groups appear most different or distinct in the bottom or low-variability case. Why? Because there is relatively little overlap between the two bell-shaped curves. In the high variability case, the group difference appears least striking because the two bell-shaped distributions overlap so much.



This leads us to a very important conclusion: when we are looking at the differences between scores for two groups, we have to judge the difference between their means relative to the spread or variability of their scores. The t-test does just that.

The formula for the t-test is a ratio. The top part of the ratio is just the difference between the two means or averages. The bottom part is a measure of the variability or dispersion of the scores. Figure 3 shows the formula for the t-test and how the numerator and denominator are related to the distributions.



The t-value will be positive if the first mean is larger than the second and negative if it is smaller.

The choice between the 2 is very intuitive. If you measure a variable in 2 **different populations**, you choose the independent t-test as the 2 populations are independent from each other. If you measure a variable 2 times in the **same population**, you go for the paired t-test.

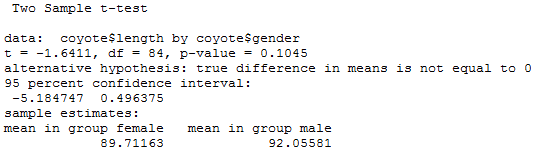
So say, you want to compare the weights of 2 breeds of sheep. To do so, you take a sample of each breed (the 2 samples have to be comparable) and you weigh each animal. You then run an Independent-samples t-test on your data to find out if the difference is significant.

You may also want to test the effect of a diet on the level of a particular molecule in sheep’s blood: to do so you choose one sample of sheep and you take a blood sample at day 1 and another one say at day 30. This time you apply a Paired-Samples t-test as you are interested in each individual difference between day 1 and day 30.

### Independent t-test

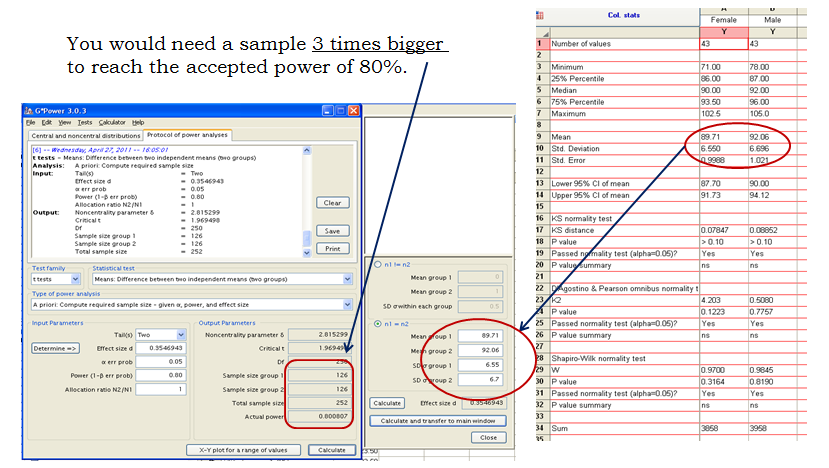
>ttest.coyote<-t.test(coyote$length~coyote$gender, var.equal=T)

>ttest.coyote



So males coyotes are bigger than females but not significantly so (p=0.1045).

You might have noticed that despite having collected the ‘recommended’ sample size, we did not reach significance. This is because the difference observed in the collected sample is smaller than expected. If we now consider the data as the one of a pilot study and run the power analysis again, we would need a sample 3 times bigger to reach a power of 80%. Now is the time to wonder whether a 2.3cm (<3%) is biologically relevant.



### Paired t-test

The choice between independent and paired t-tests is very intuitive. If you measure a variable in 2 **different populations**, you choose the independent t-test as the 2 populations are independent from each other. If you measure a variable 2 times in the **same population**, you go for the paired t-test.

So say, you want to compare the weights of 2 breeds of sheep. To do so, you take a sample of each breed (the 2 samples have to be comparable) and you weigh each animal. You then run an Independent-samples t-test on your data to find out if the difference is significant.

You may also want to test the effect of a diet on the level of a particular molecule in sheep’s blood: to do so you choose one sample of sheep and you take a blood sample at day 1 and another one say at day 30. This time you apply a Paired-Samples t-test as you are interested in each individual difference between day 1 and day 30.

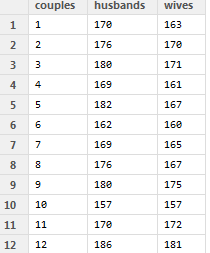
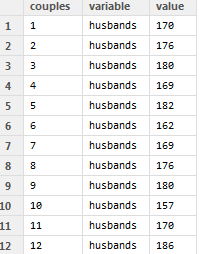
Now let’s try a Paired t-test. As we mentioned before, the idea behind the paired t-test is to look at a difference between 2 paired individuals or 2 measures for a same individual. For the test to be significant, the difference must be different from 0.

*Example* (file:husband and wife.csv)

>couples<-read.csv("husband and wife.csv", header=T)

*ggplot2* likes to think vertically so we need to restructure the data frame. For that we are going to use the melt() function.

>couples.stack<-melt(couples, id = "couples")

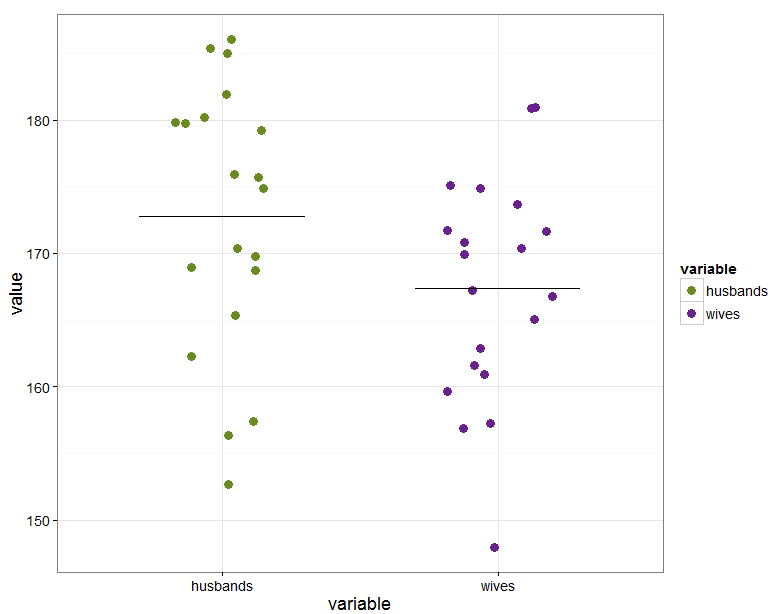


>scatter.couples.stack<-ggplot(couples.stack,aes(variable,value, colour=variable))

+geom\_point(position=position\_jitter(width=0.2),size=4) +geom\_errorbar(stat="hline",yintercept="mean",width=0.6,colour="black", aes(ymax=..y..,ymin=..y..))

+scale\_colour\_manual(values=c("olivedrab4", "darkorchid4"))

>scatter.couples.stack



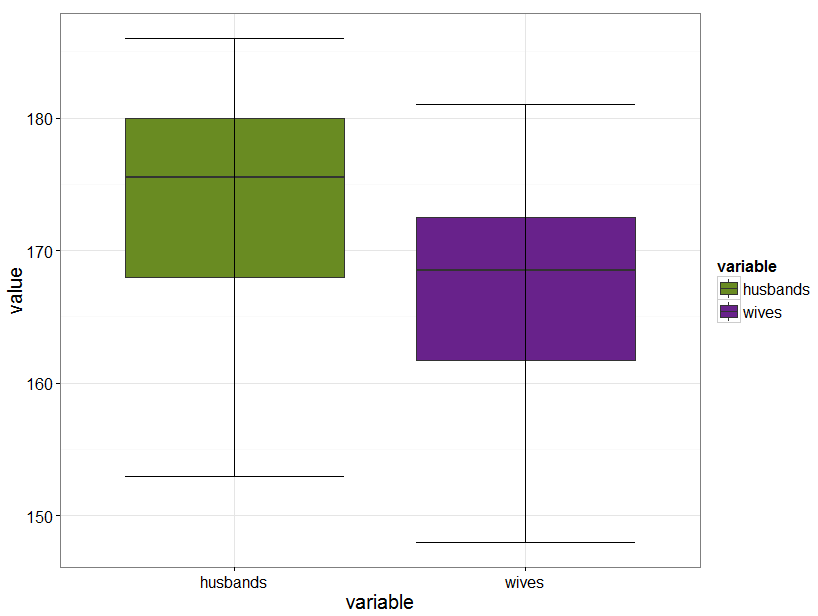
From the graph above, we can conclude that if husbands are taller than wives, this difference does not seem significant. Before running the paired t-test to get a p-value we are going to check that the assumptions for parametric stats are met. The box plots below seem to indicate that there is no significant departure from normality and this is confirmed by the D’Agostino-Pearson test test.

>boxplot.couples<-gplot(couples.stack,aes(variable,value,fill=variable))

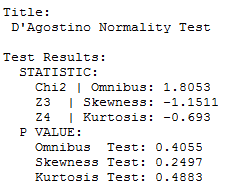
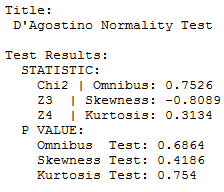
+geom\_boxplot()+stat\_boxplot(geom="errorbar")

+scale\_fill\_manual(values=c("olivedrab4", "darkorchid4"))

>boxplot.couples

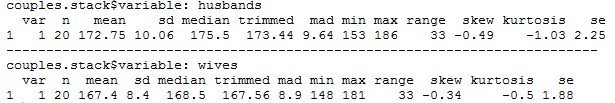


>by(couples.stack$value,couples.stack$variable, dagoTest)

As we are not using the stat.desc() function we don’t get any descriptive of the data. We can still get them with the describe() function.

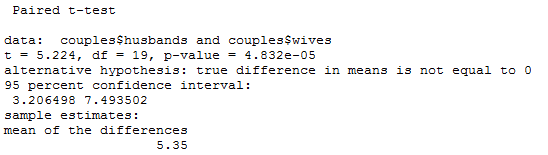
>by(couples.stack$value,couples.stack$variable, describe)



Let’s run the t-test:

>couples<-read.csv("husband and wife.csv", header=T)

>t.test(couples$husbands, couples$wives, paired=T)



You will have noticed that we did not run a test for the equality of variances in the paired t-test; this is because it is actually looking at only one sample: the difference between the husbands and the wives.

The paired t-test turns out to be highly significant (see Table above). So, how come the graph and the test tell us different things?

The problem is that we don’t really want to compare the mean size of the wives to the mean size of the husband, we want to look at the difference pair-wise, in other words we want to know if, on average, a given wife is taller or smaller than her husband. So we are interested in the mean difference between husband and wife.

We need to plot the difference. To do so, we need to create a new column called ‘difference’ in the original couples file, restructure this new file like we did before and plot the differences only.

We create a new column [,4] in the original data frame which will contain the difference:

>couples[,4]<- couples$husbands-couples$wives

>colnames(couples)[4]<-"difference"

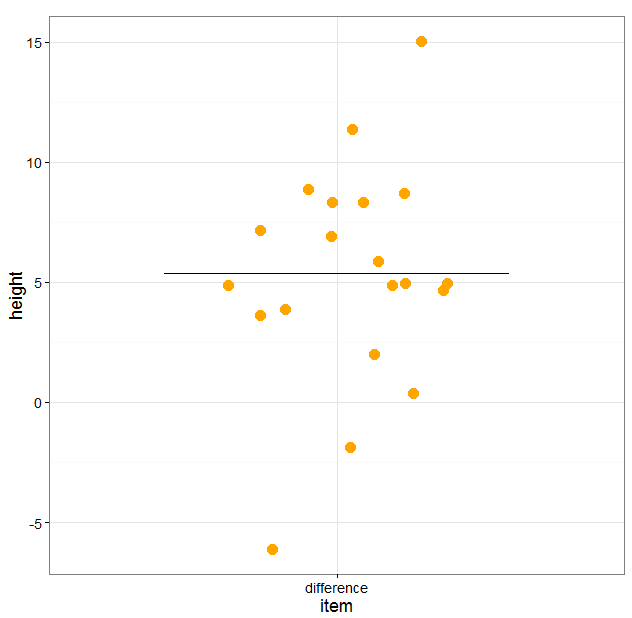
or

>couples$difference<-couples$wives-couples$husbands

We restructure the data and rename the columns to get a prettier file:

>couples.diff.stack<-melt(couples, id="couples")

>colnames(couples.diff.stack)[2:3]<-c("item","height")



And now we can plot the differences as a scatterplot:

>scatter.difference<-ggplot(couples.diff.stack,[couples.diff.stack$item=="difference",]

,aes(item,height))

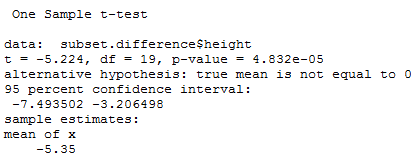
+geom\_point(position=position\_jitter(width=0.2),size=5,colour="orange")

+geom\_errorbar(stat="hline",yintercept="mean",width=0.6,aes(ymax=..y.., ymin=..y..))

>scatter.difference

To run a one-sample t-test.

>t.test(couples.diff.stack$height[couples.diff.stack$item=="difference"],mu=0)



## 3-4 Comparison of more than 2 means: Analysis of variance

## A bit of theory

When we want to compare more than 2 means (e.g. more than 2 groups), we cannot run several t-test because it increases the **familywise error rate** which is the error rate across tests conducted on the same experimental data.

Example: if we want to compare 3 groups (1, 2 and 3) and we carry out 3 t-tests (groups 1-2, 1-3 and 2-3), each with an arbitrary 5% level of significance, the probability of not making the type I error is 95% (= 1 - 0.05). The 3 tests being independent, we can multiply the probabilities, so the overall probability of no type I errors is: 0.95 \* 0.95 \* 0.95 = 0.857. Which means that the probability of making at least one type I error (to say that there is a difference whereas there is not) is 1 - 0.857 = 0.143 or 14.3%. So the probability has increased from 5% to 14.3%. If we compare 5 groups instead of 3, the family wise error rate is 40% (= 1 - (0.95)n)

To overcome the problem of multiple comparisons, we need to run an **Analysis of variance (ANOVA)**, which is an extension of the 2 groups comparison of a t-test but with a slightly different logic. If we want to compare 5 means, for example, we can compare each mean with another, which gives you 10 possible 2-group comparisons, which is quite complicated! So, the logic of the t-test cannot be directly transferred to the analysis of variance. Instead the ANOVA compares variances: if the variance amongst the 5 means is greater than the random error variance (due to individual variability for instance), then the means must be more spread out than we would have explained by chance.

The statistic for ANOVA is the F ratio:

|  |  |
| --- | --- |
| F = | variance among sample means |
| variance within samples (=random. Individual variability) |

also:

|  |  |
| --- | --- |
| F = | variation explained by the model (systematic) |
| variation explained by unsystematic factors |

If the variance amongst sample mean is greater than the error variance, then F>1. In an ANOVA, we test whether F is significantly higher than 1 or not.

Imagine we have a dataset of 78 data points, we make the hypothesis that these points in fact belong to 5 different groups (this is our hypothetical model). So we arrange your data into 5 groups and we run an ANOVA.

Below, is a typical example of analyse of variance table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source of variation** | **Sum of Squares** | **df** | **Mean Square** | **F** | **p-value** |
| Between Groups | 2.665 | 4 | 0.6663 | 8.423 | <0.0001 |
| Within Groups | 5.775 | 73 | 0.0791 |  |  |
| Total | 8.44 | 77 |  |  |  |

Let’s go through the figures in the table. First the bottom row of the table:

Total sum of squares = ∑(xi – Grand mean)2

In our case, Total SS = 8.44. If we were to plot your data to represent the total SS, we would produce the graph below. So the total SS is the squared sum of all the differences between each data point and the grand mean. This is a quantification of the overall variability in our data. The next step is to partition this variability: how much variability between groups (explained by the model) and how much variability within groups (random/individual/remaining variability)?



According to our hypothesis our data can be split into 5 groups because, for instance, the data come from 5 cell types, like in the graph below.

So we work out the mean for each cell type and we work out the squared differences between each of the means and the grand mean (∑ ni (Meani - Grand mean)2 ). In our example (second row of the table): Between groups SS = 2.665 and, since we have 5 groups, there are 5 – 1 = 4 df, the mean SS = 2.665/4 = 0.6663.

If you remember the formula of the variance (= SS / N-1, with df=N-1), you can see that this value quantifies the variability between the groups’ means: it is the between group variance.



***Within*** *group variability*

***Between*** *group variability*

There is one row left in the table, the within groups variability. It is the variability within each of the five groups, so it corresponds to the difference between each data point and its respective group mean:

Within groups sum of squares = ∑ (xi - Meani)2 which in our case is equal to 5.775.

This value can also be obtained by doing 8.44-2.665 = 5.775, which is logical since it is the amount of variability left from the total variability after the variability explained by your model has been removed.

In our example, the 5 groups sizes are 12, 12, 17, 17 and 17 so df = 5 x (n – 1) = 73

So the mean within groups: SS = 5.775/73 = 0.0791. This quantifies the remaining variability, the one not explained by the model, the individual variability between each value and the mean of the group to which it belongs according to your hypothesis. From this value can be obtained what is often referred to as the Pooled SD (=SQRT(MS(Residual or Within Group)). When obtained in a pilot study, this value is used in the power analysis.

At this point, we can see that the amount of variability explained by our model (0.6663) is far higher than the remaining one (0.0791).

So, we can work out the F-ratio: F = 0.6663 / 0.0791 = 8.423

The level of significance of the test is calculated by taking into account the F ratio and the number of df (degree of freedom) for the numerator and the denominator. In our example, p<0.0001, so the test is highly significant and we are more than 99% confident when we say that there is a difference between the groups’ means. This is an overall difference and even if we have an indication from the graph, we cannot tell which mean is significantly different from which.

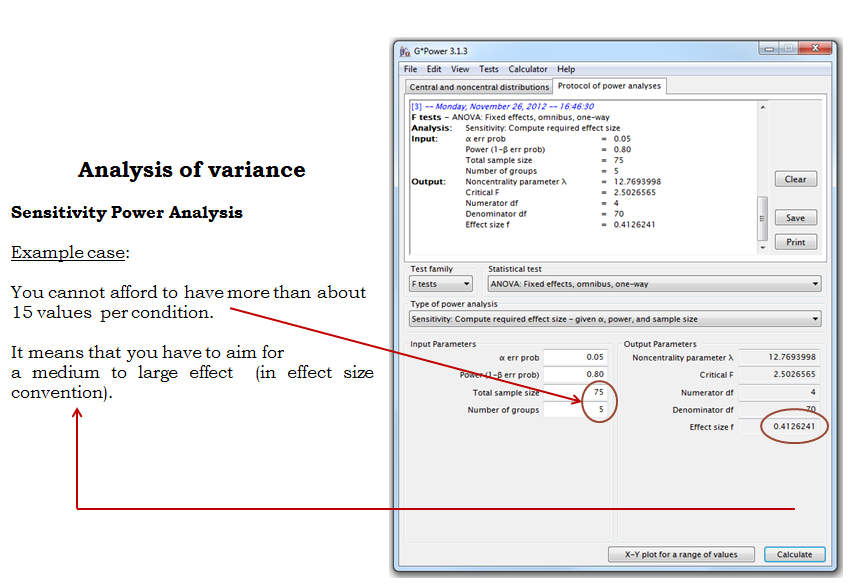
This is because the ANOVA is an “omnibus” test: it tells we that there is (or not) an overall difference between our means but not exactly which means are significantly different from which other ones. This is why we apply post-hoc tests. Post hoc tests could be compared to t-tests but with a more stringent approach, a lower significance threshold to correct for familywise error rate. We will go through post-hoc tests more in detail later.

*Example* (File: protein expression.csv)

Let’s do it in more details. We want to find out if there is a significant difference in terms of protein expression between 5 cell types.

As usual, we start by designing our experiment, we decide to go for an analysis of variance and then, we get to the point where we have to decide on the sample size. In this case, we are limited in terms of the sample size we can afford: only about n=15 per condition.

### Power analysis with an ANOVA



What we already know about these cell lines makes we quite confident that we can reach such an effect size so we decide to go ahead.

First we need to see whether the data meet the assumptions for a parametric approach. Well it does not look good: 2 out of 5 groups (C and D) show a significant departure from normality (See Table below). As for the homogeneity of variance, even before testing it, a look at the scatter plots and box plots (see Graphs below) tells us that there is no way the second assumption is met. The data from groups C and D are quite skewed and a look at the raw data shows more than a 10-fold jump between values of the same group (e.g. in group A, value line 4 is 0.17 and value line 10 is 2.09).

First we import the dataset:

>setwd("M:/Work/R Course/Exercises/")

>protein<-read.csv("protein.expression.csv",header=T)

Then for ease of graphical representation we restructure it:

>protein.stack<-melt(protein)

>colnames(protein.stack)<-c("line","expression")

and we get read of the missing values:

>protein.stack.clean <- protein.stack[!is.na(protein.stack$expression),]

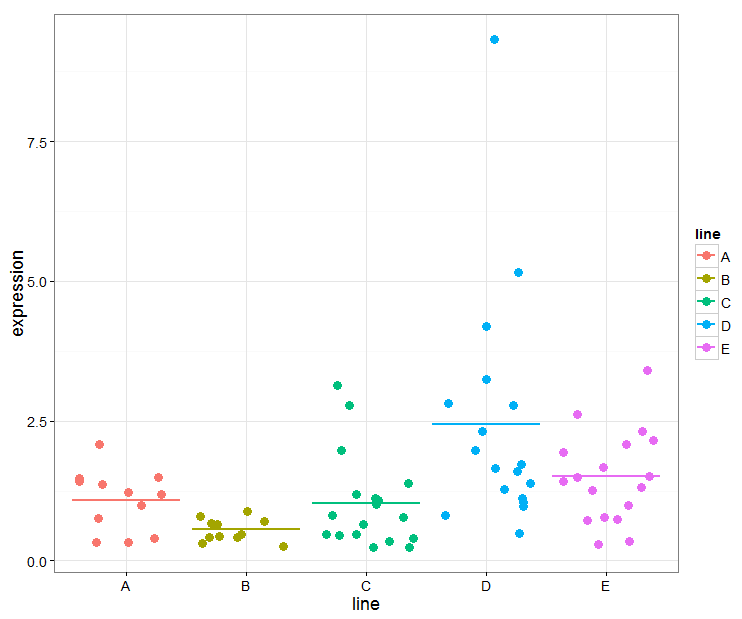
Now we can plot the data, either as scatterplot:

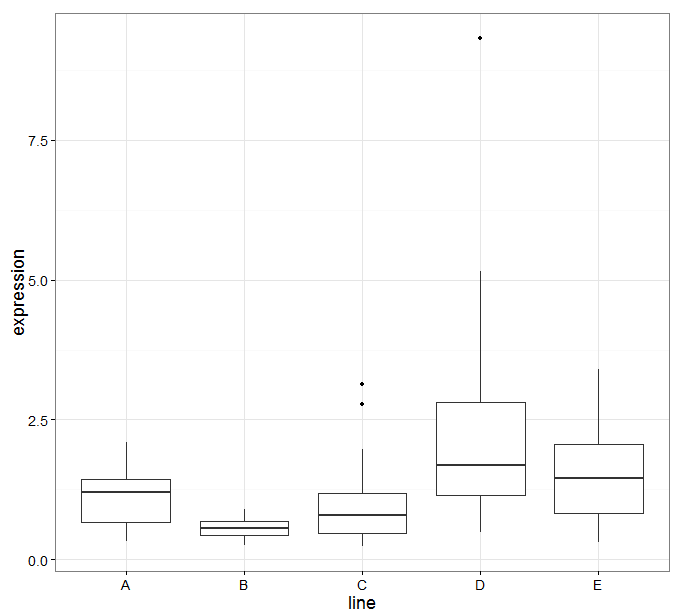
>scatter.plot<-ggplot(protein.stack.clean,aes(line,expression,colour=line))

+geom\_point(position="jitter",size=4)

+geom\_errorbar(stat="hline",size=1,yintercept="mean", aes(ymin=..y..,ymax=..y..))

>scatter.plot

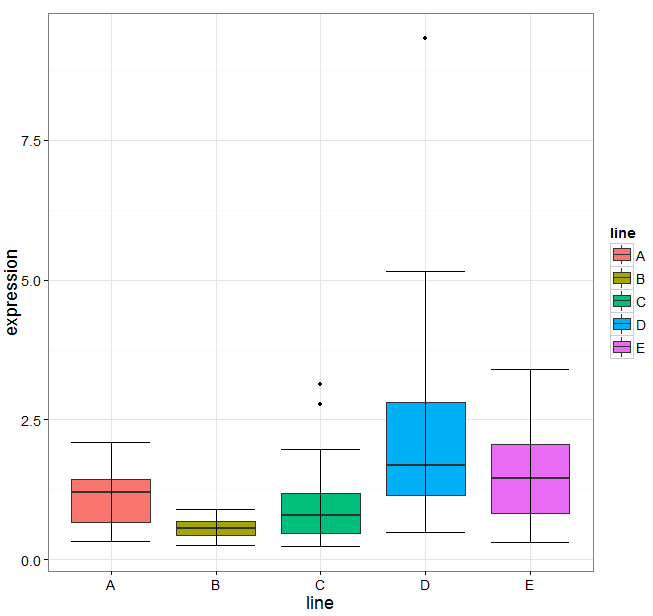




Or boxplot:

Simplest version:

>boxplot.protein<-ggplot(protein.stack.clean, aes(line,expression))+geom\_boxplot()

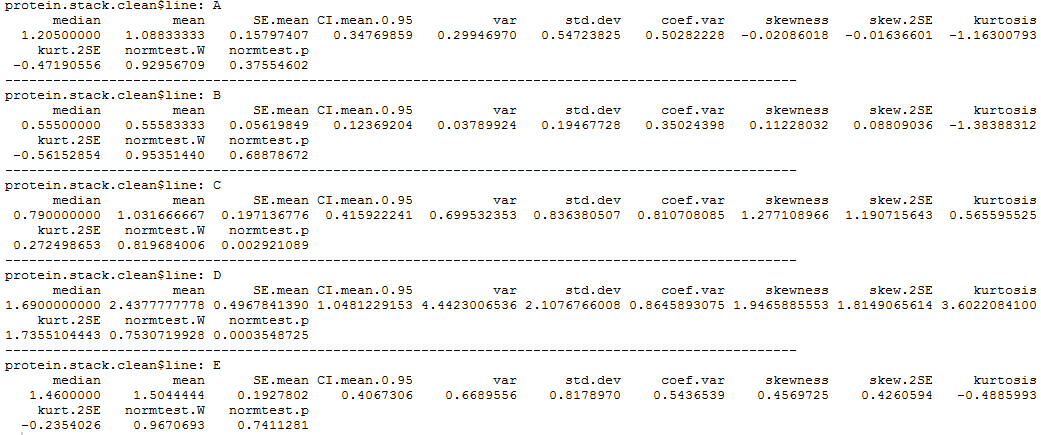


Prettier version:

>boxplot.protein<-ggplot(protein.stack.clean, aes(line,expression, fill=line))+geom\_boxplot()+stat\_boxplot(geom="errorbar")

First we need to see whether the data meet the assumptions for a parametric approach. Well it does not look good: 2 out of 5 groups (C and D) show a significant departure from normality (See Table below, we cannot use the D’Agostino test as R requires n>=20). As for the homogeneity of variance, even before testing it, a look at the scatter plots and box plots (see Graphs above) tells us that there is no way the second assumption is met. The data from groups C and D are quite skewed and a look at the raw data shows more than a 10-fold jump between values of the same group (e.g. in group A, value line 4 is 0.17 and value line 10 is 2.09).

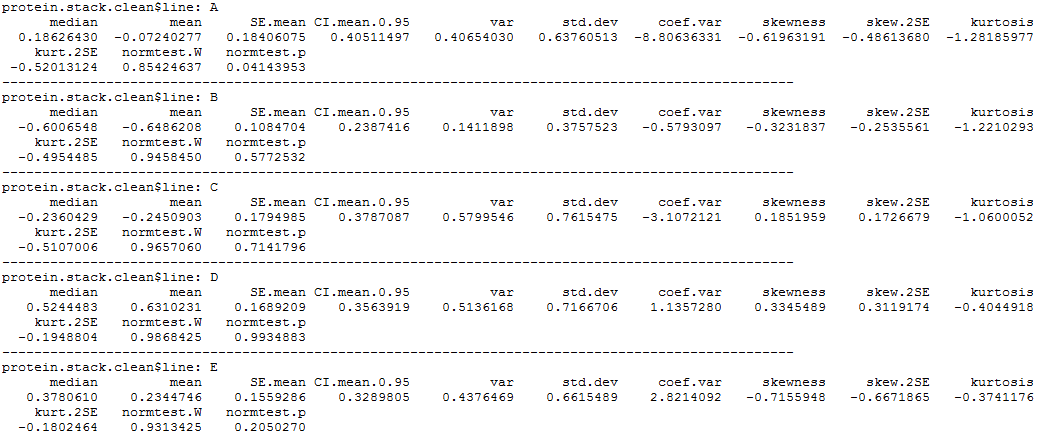
>by(protein.stack.clean$expression,protein.stack.clean$line,stat.desc, basic=F, norm=T)



A good idea would be to log-transform the data so that the spread is more balanced and to check again on the assumptions. The variability seems to be scale related: the higher the mean, the bigger the variability. This is a typical case for log-transformation.

To do so, you go to:

>protein.stack.clean$log10.expression<-log10(protein.stack.clean$expression)

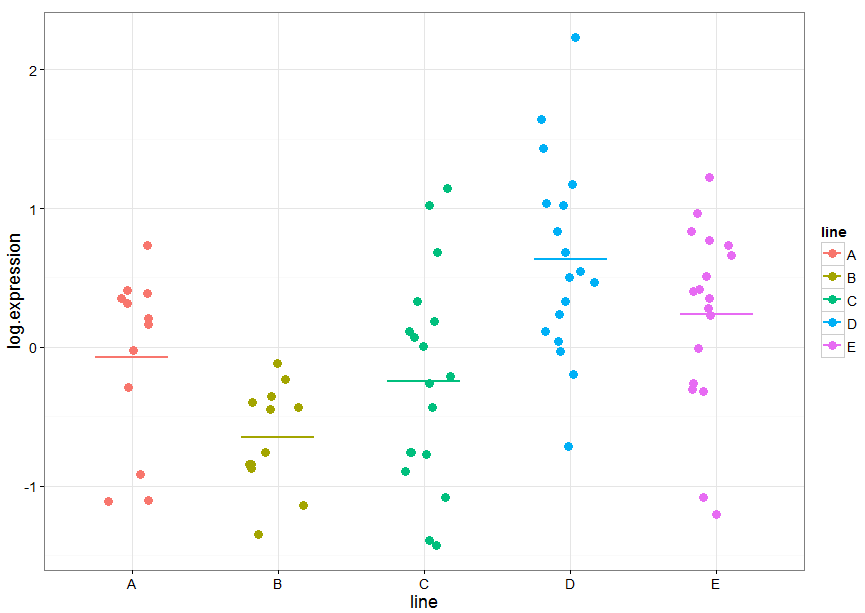


OK, the situation is getting better: the first assumption is met-ish and from what we see when we plot the transformed data (Box-plots and scatter plots below) the homogeneity of variance has improved a great deal.

>scatter.log.protein<ggplot(protein.stack.clean,aes(line,log.expression,colour=line))+geom\_point(position=position\_jitter(width=0.2),size=4)

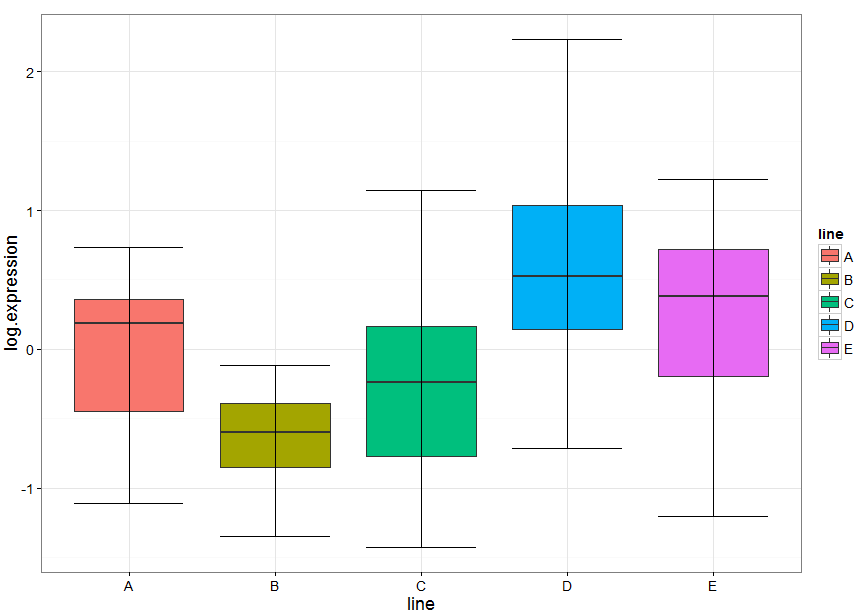
+geom\_errorbar(stat="hline",size=1,width=0.5,yintercept="mean", aes(ymin=..y..,ymax=..y..))

>scatter.log.protein



>boxplot.log.protein<-ggplot(protein.stack.clean, aes(line,log.expression, fill=line))+geom\_boxplot()+stat\_boxplot(geom="errorbar")

>boxplot.log.protein



One last thing before we run the ANOVA: we need to check for the second assumption: the homogeneity of variance. To do so, we do what we did before running the t-test: we run a Levene test:

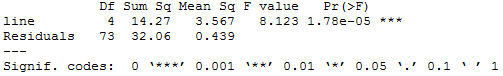
>leveneTest(protein.stack.clean$log.expression,protein.stack.clean$line,center=mean)



Now that we have sorted out the data, we can run the ANOVA: to do so, you go:

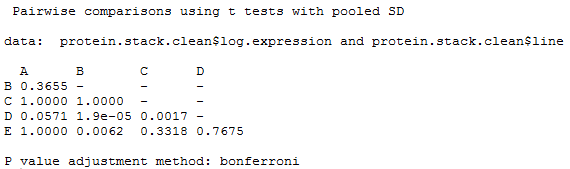
>anova.log.protein<-aov(log.expression~line,data=protein.stack.clean)

>summary(anova.log.protein)



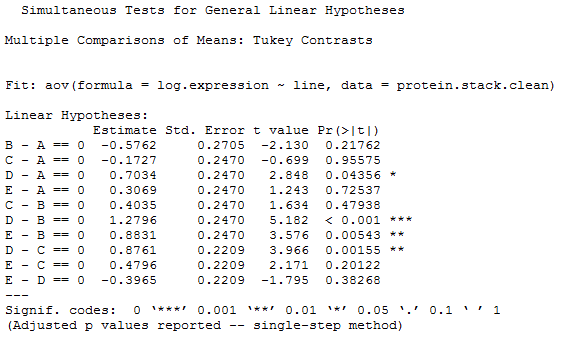
The overall p-value is significant (p=1.78e-05) so the next thing to do is to choose a post-hoc test. There are 2 widely used: the Bonferroni test which is quite conservative so we should only choose it when we are comparing no more than 5 groups and the Tukey which is more liberal. First let’s try the Bonferroni test. It is built into R:

>pairwise.t.test(protein.stack.clean$log.expression,protein.stack.clean$line, p.adj = "bonf")



Tukey can be implemented using the glht() function that is part of the *multcomp* package wuich needs to be installed and loaded.

>posthoc.tukey<-glht(anova.log.protein,linfct = mcp(line = "Tukey"))



Again, from the table above we can find out which pairwise comparison reaches significance and which does not.

## 3-5 Correlation

If we want to find out about the relationship between 2 variables, we can run a correlation.

*Example* (File: Exam Anxiety.dat)

>setwd("M:/Work/R Course/Exercises/")

>exam.anxiety<-read.table("Exam Anxiety.dat", sep="\t",header=T)

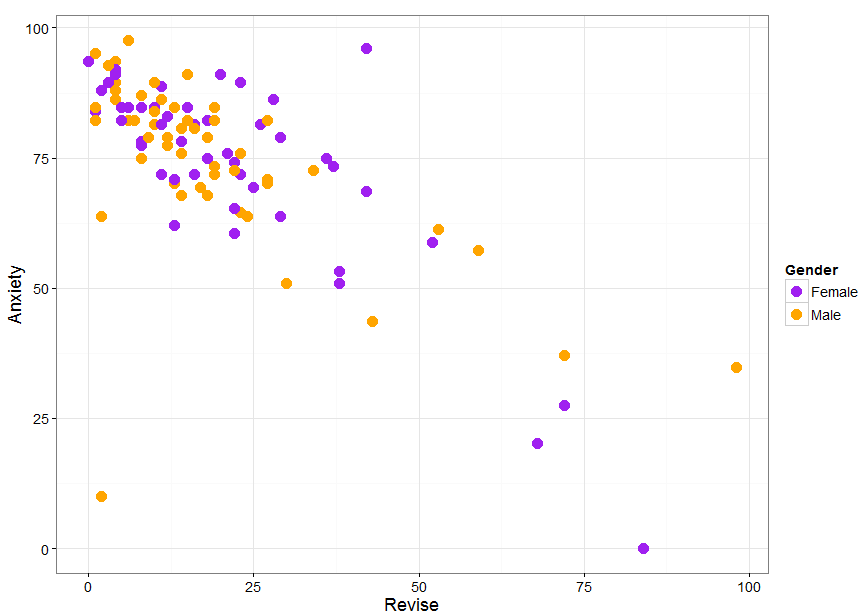
The first thing we are going to do is to plot the data. We will start with revising time vs. anxiety levels.

>graph.revise.anxiety<-gplot(exam.anxiety,

aes(Revise,Anxiety,colour=Gender))

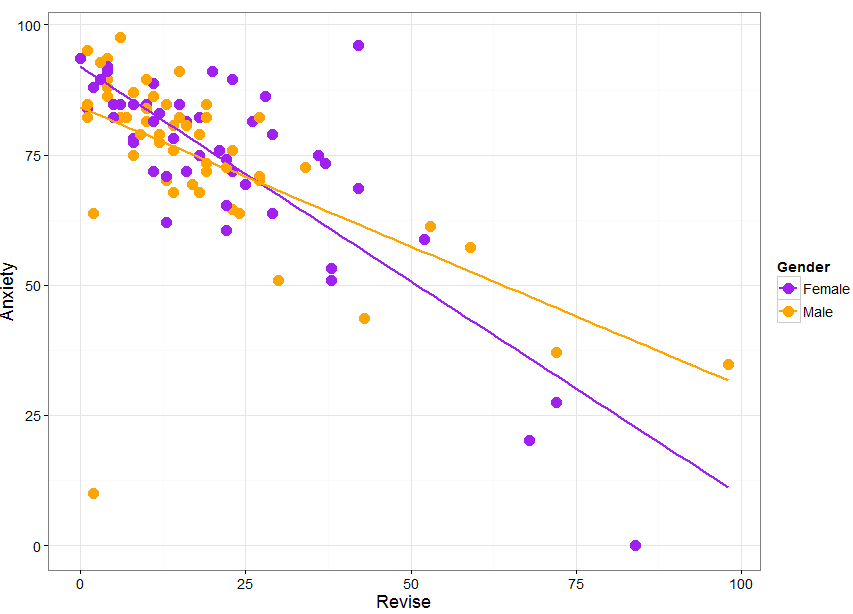
+geom\_point(size=5)+scale\_colour\_manual(values=c("purple","orange"))

>graph.revise.anxiety



By looking at the graph, one can think that something is happening here. To get a better idea we can add lines of best fit to the graph:

> graph.revise.anxiety+geom\_smooth(method=lm,se=F, fullrange=T,size=1)



Now, if you want to know if the relationship between your 2 variables is significant, you need to run a correlation test.

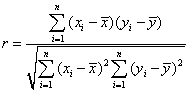
### A bit of theory: Correlation coefficient

A correlation is a measure of a linear relationship (can be expressed as straight-line graphs) between variables. The simplest way to find out whether 2 variables are associated is to look at whether they covary. To do so, you combine the variance of one variable with the variance of the other.

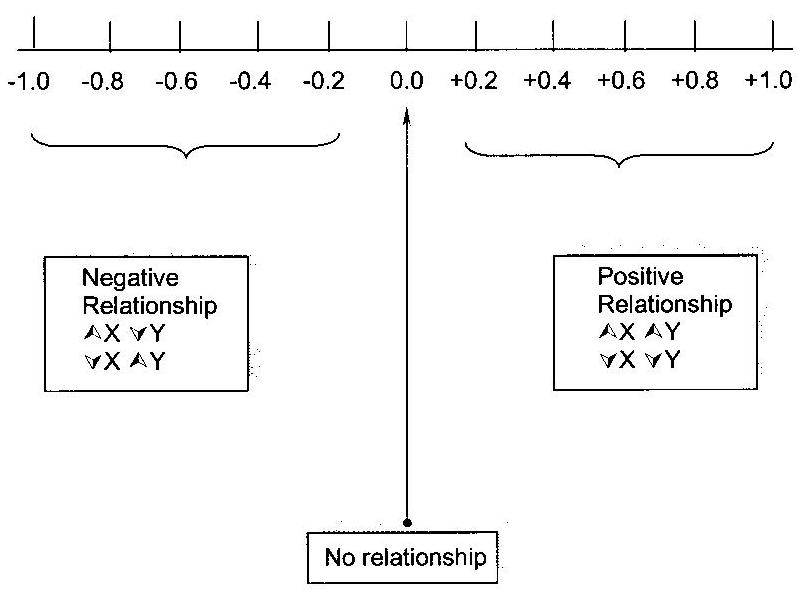
cov(X,Y)==sum_(i==1)^N((x_i-x^_)(y_i-y^_))/N.

A positive covariance indicates that as one variable deviates from the mean, the other one deviates in the same direction, in other word if one variable goes up the other one goes up as well.

The problem with the covariance is that its value depends upon the scale of measurement used, so you won’t be able to compare covariance between datasets unless both data are measures in the same units. To standardise the covariance, it is divided by the SD of the 2 variables. It gives you the most widely-used correlation coefficient: the Pearson product-moment correlation coefficient “r”.



Of course, you don’t need to remember that formula but it is important that you understand what the correlation coefficient does: it measures the magnitude and the direction of the relationship between two variables. It is designed to range in value between 0.0 and 1.0.



The 2 variables do not have to be measured in the same units but they have to be proportional (meaning linearly related)

One last thing before we go back to our example: the coefficient of determination r2: it gives you the proportion of variance in Y that can be explained by X, in percentage.

In our case, we may want to see if the level of correlation differs between boys and girls. To do so, we type:

>cor(exam.anxiety[exam.anxiety$Gender=="Male", c("Exam","Anxiety","Revise")])



Because we did not specify which columns we are interested in, we get the correlations coefficients for the combinations between all 3 variables. If we want to build the equation of the line, we can type:

>coef(lm(Anxiety~Revise,exam.anxiety,exam.anxiety$Gender == "Male"))



So it goes: Anxiety=84.19-0.53\*Revise

Same thing for the girls:

>cor(exam.anxiety[exam.anxiety$Gender == "Female", c("Exam","Anxiety","Revise")])



>coef(lm(Anxiety~Revise,exam.anxiety,exam.anxiety$Gender == "Female"))

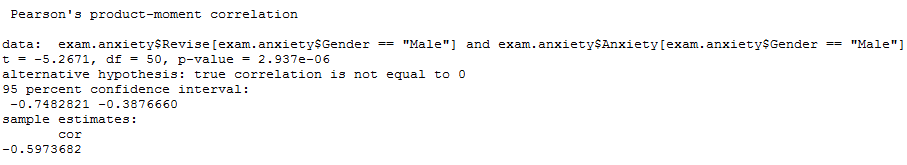
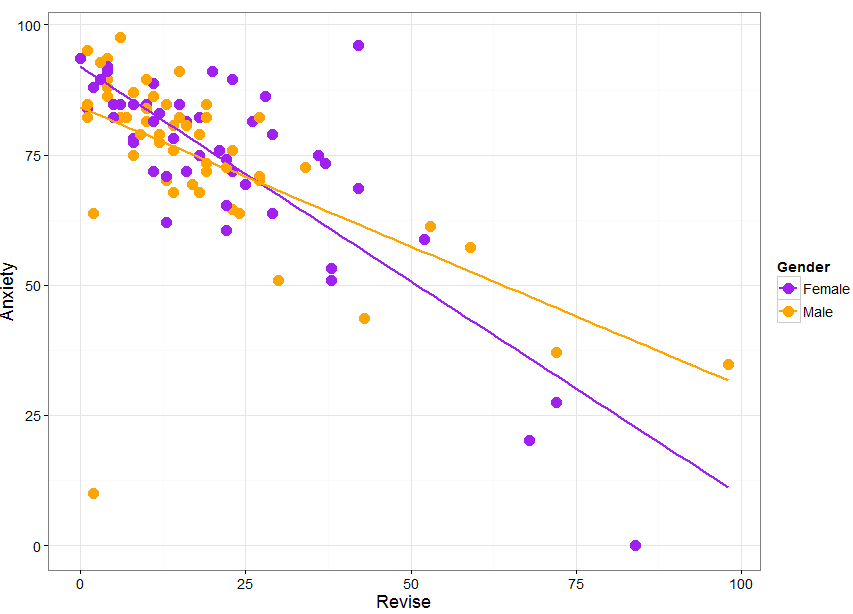


Giving you: Anxiety=91.94-0.82\*Revise

So the slope is steeper for the girls than for the boys which is consistent with the graph.

Now you might have noticed that there were no p-values in the output above. To get p-values, we need to use another function cor.test(). For example if we want to know if the correlation between anxiety and revise in significant in male students, we type:

>cor.test(exam.anxiety$Revise[exam.anxiety$Gender=="Male"], exam.anxiety$Anxiety[exam.anxiety$Gender=="Male"],alternative= c("two.sided"),method = c("pearson"), conf.level = 0.95)



### Outliers and influential cases

Now you might have noticed that one point is really far from the others. Some cases may bias a model; these are either outliers or so-called influential cases or both. We are going to try to find out which one is this one.

To identify a case as an outlier, we look at the standardized residuals which behave like z-scores.

Quick reminder about z-score and normal distribution:



µ is the mean of the sample and σ its standard deviation. So basically z represents the distance between a raw value and the sample mean in units of standard deviation. In case of a normal distribution, 95% of the values are within +/-1.96\* σ from the mean.

Now residuals are the distance between each case and the model (here the regression line). To define a universal cut-off, we divide those residuals by their standard deviation which works in the same way as the z-scores hence the similar boundaries. You don’t want more than 5% of cases above 2 (1.96-ish) or more than 1% above 2.5. Any case with an absolute value above 3 should be considered as an outlier. That’s for the outlier’s identification.

As for the influential cases, there are several measures among which we will use:

- the Cook’s distance: Measure of aggregate impact of each observation on the group of regression coefficients. Any value above 1 may indicate an influential case.

- the leverage: Measure of how far an observation is from the others. Values 2 or 3 times bigger than the average leverage ((number of predictors+1)/sample size) can be influential.

- the covariance ratio: Measure of the impact of each observation on the variances of the regression coefficients and their covariances. Values should between the upper and lower limits of the covariance ratio: 1 +/- 3\*average leverage.

The bottom line is that first you look at the outliers, once you have identified them, you check the influence statistics and if one or more are ‘out of line’ you can then safely remove the value.

Let’s check out our misbehaving case. It is a boy who spent 2 hours revising, did not feel stressed about it (Anxiety score:10) and managed a 100% mark in his exam.

So first let’s build a data frame with just the males data:

>exam.anxiety.male<-exam.anxiety[exam.anxiety$Gender == "Male",]

Now we are going to create 4 variables to look for outliers and influential cases but first, we need to run the regression from which we will identify the misbehaving case(s).

>lm.anxiety.male<-lm(Anxiety~Revise,exam.anxiety.male)

>exam.anxiety.male$residuals<-resid(lm.anxiety.male)

>exam.anxiety.male$stan.residuals<-rstandard(lm.anxiety.male)

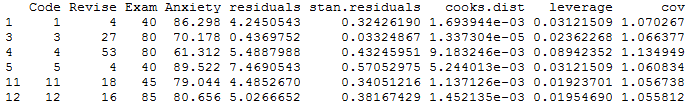
>exam.anxiety.male$cooks.dist<-cooks.distance(lm.anxiety.male)

>exam.anxiety.male$leverage<-hatvalues(lm.anxiety.male)

>exam.anxiety.male$cov<-covratio(lm.anxiety.male)

We can use the head() function to check our new file.

>head(exam.anxiety.male)



First let’s see if there are any cases that a standardised residual>3 or <-3

>exam.anxiety.male[abs(exam.anxiety.male$stan.residuals)>3,]



So one case only which happens to be the one we are worrying about. Now the Cook’s distance <1 so that’s OK, the average leverage=0.034 so that’s OK too. We are left with the covariance ratio. Our calculation tells us that the value should be between 0.89 and 1.107, with cov=0.151 this case is well below the lower limit.

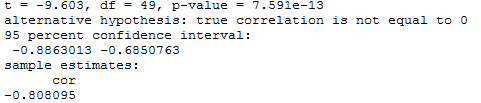
If we re-run the linear regression without this case:

>lm.anxiety.male<lm(Anxiety~Revise,exam.anxiety.male[abs(exam.anxiety.male$stan.residuals)<3,])



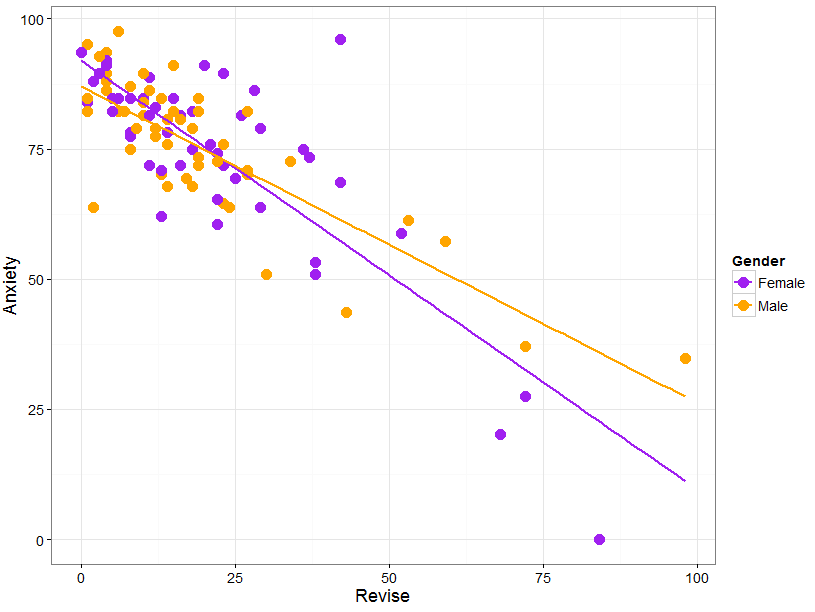
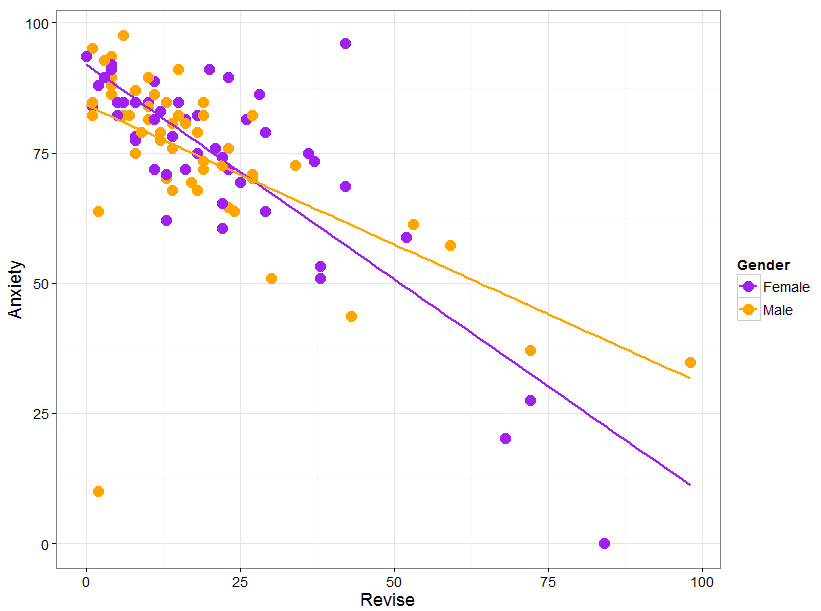
We get, as expected, a steeper slope. We can also re-run the correlation analysis.

>cor.test(exam.anxiety.male$Revise,exam.anxiety.male$Anxiety, abs(exam.anxiety.male$stan.residuals)<3],alternative= c("two.sided"),method = c("pearson"), conf.level = 0.95)



We have a better correlation now: lower p-value, narrower confidence interval and bigger correlation coefficient in absolute values.

>graph.revise.anxiety<-ggplot(exam.anxiety[exam.anxiety$Code!=78,], aes(Revise,Anxiety,colour=Gender))+geom\_point(size=5)+scale\_colour\_manual(values=c("purple","orange"))+geom\_smooth(method=lm,se=F, fullrange=T,size=1)



**References**

Cumming G., Fidler F. and Vaux D.L. Error bars in experimental biology*.* *The Journal of Cell Biology*, Vol. 177, No.1, 7-11.

Field A. 2012. *Discovering statistics using R* (1st Edition). London: Sage.

McKillup S. 2005. *Statistics explained*. Cambridge: Cambridge University Press.