Advanced Quantitative methods

Duncan Golicher

2019-03-21

Contents

Chapter 1

Introduction

In this unit you wil learn to apply a range of statistical methods that are considered to be more advanced than are typically taught on an introductory course in data analysis. However the unit aims to provide more than a simple set of recipes for applying methods. Advances in computing power have resulted in an explosion of new methods for analysing ecological data. It would be impossible to design a course that includes examples of all the possible methods that might be relevant to the analysis of a data set that forms the subject of a master's dissertation. Data analysis does not simply involve applying statistical methods. Good data management and effective data manipulation are just as important as good analysis. So an important element of the course wil focus on understanding the nature of data and the ways data can be manipulated.

1.1 The new statistics

The underlying philosophy of the course is based on a contemporary concept of good statistical practice. This has sometimes been called "the new statistics" (Hobbs and Hilborn, 2006)

The aim of the new statistics is to evaluate the relative strength of evidence in data for hypotheses represented as models. Traditionally, models used by ecologists for statistical inference have been limited to a relatively small set of linear forms. The functional forms and definitions of parameters in these models were chosen for statistical reasons; that is, they were not constructed to explicitly symbolize biological states and processes. Consequently, composing models to represent hypotheses has traditionally played a relatively minor role in developing testable statements by most ecological researchers. Statistical models were used to represent verbal hypotheses, and little thought was applied to the model building that ultimately supported inference.

The new statistics require deliberate, thoughtful specification of models to represent competing ecological hypotheses

1.2 Statistical models vs statistical tests

"New statistics" has developed not just as a response to advances in computing power, although that has played an important role. A fundamental element of contemporary thinking with regard to the use of statistics is the avoidance, or at the least the downplaying, of null hypothesis **tests* (NHST). This can be confusing for students who have taken introductory courses in statistics in which the words "test" and "p-value" constantly were emphasised. Many students (and many researchers) may be unaware that the whole basis of null hypothesis testing was in fact controversial from the outset. Many prominent statisticians have argued against NHST (Nickerson et al., 2000). Some of the criticisms that can be found in Nickerson's excelent review have been severe. For example "Null hypothesis testing provides researchers with no incentive to specify either their own research hypotheses or competing hypotheses.... it is surely the most bone-headedly misguided procedure ever institutionalized in the rote training of science students" (Gigerenzer, 1998). The biggest problem with NHST is that it does not actually test a hypothesis of genuine interest. Some of the other arguments against NHST made by statisticians are quite technical in nature, although they are well worth trying to understand. A reasonably non-technical review is provided by Goodman (2008) in which 12 common misconceptions are laid out clearly. A very influential and highly cited paper that restates many of the criticisms of NHST made by statisticians in the context of ecology is Johnson (1999). I won't repeat all the arguments made by Johnson. It it well worth reading the paper.

1.3 Bayesian vs frequentist approaches to inference

It is my *belief* (and the word belief is important in this context) that all students should be aware that a vigorous debate arose between two schools of thought regarding the nature of statistical inference at the time that many of the conventional, text book, statistical tests were being developed. It is fascinating to read the words of R A Fisher writing in the "Design of experiments", first published in 1935 . Fisher stated that "*I shall not assume the truth of Bayes' axiom*". What is most remarkable about that particular statement is that Bayes' axiom is a simple mathematical consequence of applying the rules of probability. There is no controversy whatsoever regarding the axiom's mathematical validity. Fisher was aware that implementing what was known as "inverse probability" would be mathematically unfeasible in the context of the sort of problems he was working on at the time. Integration formulas to calculate marginal likelihoods are horrendously complex in even very simple cases involving continuous ditributions, and are totally intractable for most designs. However his main objection to the use of Bayes' theorem was that to him it appeared to introduce an element of subjectivity, and thus did not allow researchers to reach clear conclusions. It is unfortunate that this led to what some studies of the history of scientific thought have described as a "holy war" between two schools of thought. The unfortunate consequence of this was that efforts to remove the incorporation of subjective prior beliefs into analyses led to the conventional analytical device of NHST in which **all** prior information was ignored. This was never intended by Fisher himself. It occurred almost by accident, as a result of the development of NHST by Pearson and Neyman. Fisher originally intended p-values to be indications that further research was worth conducting, and not as decision rules regarding the truth of either a null hypothesis nor any alternative hypothesis. However for a while this got overlooked in statistical courses that emphasised the logic and mathematical basis of NHST without placing the null in the context of any field of study. Students being told to state null hypotheses in the form H_0 "There is no difference in species richness between heathland and grassland" would be quite correct in pointing out the illogical and unhelpful nature of H_0. We already know *a priori* that there must be a difference, so why on earth would we ever test this? The answer is that we clearly should not. In an applied setting we would aim to develop a much richer and more informative analysis of the difference between the two vegetation types, conditional on the data that we have available. There is no need to place subjective prior probabilities on the size of the difference in order to do this, but to ignore all relevant knowledge regarding the two systems would clearly be quite wrong. Finding evidence against H_0 is only worth stating as an objective of a study such if H_0 is inherently credible. This can be reasonable in the context of a well planned experiment, but In all other cases there will be much more information to be obtained from the data when H_0 is not considered to be an objective of the analysis.

1.4 Using p-values with discretion

In the quantiative and spatial analysis unit I introduced a pragmatic approach towards NHST that will continue to be used on this unit. This approach emphasises confidence intervals as the most important result of applying technique based on frequentist inference. It accepts the use of p-values associated with null hypothesis tests, but only as the "statistic of last resort". The actual numerical values of confidence intervals match those produced by Bayesian methods with non-informative priors in many cases. So, confidence intervals can informally be interpreted as measures of uncertainty regarding the parameters of interest, even if formally they are not. When taking this approach, when running any regresion analysis the most important result would be the slope of the line, and associated confidence interval. The next most valuable output is the coeficient of determination (R squared) as a measure of the signal to scatter ratio. If all else fails, then a p-value can be reported, with due discretion and care with regard to the wording of the interpretation, as a "test" of any **detectable** association between variables conditional on the data obtained. Although power analysis is highly advisable when planning any study in order to assess whether the level of replication is likely to be sufficient to provide statistical signficance, when adopting this pragmetic approach you should not set out with the initial intention of "testing" a null hypothesis. Instead you should aim to estimate effect sizes. The most important element of any study is the scientific question that is of interest. Many questions can only be answered through estimating the size of effects and/or the shape and form of any response. Simply detecting a difference, an association or a correlation does not usually answer the question fully. We should, and usually can, do much better than that. This implies that non parametric methods will not be taught on this course, although it is still worth being aware of them as potential fall back methods of last resort when all else fails. Non parametric tests are easy to run and understand in R. The R code to run them is provided in this crib sheet for reference.

http://r.bournemouth.ac.uk:82/AQM/AQM_2018/Crib_sheets/Classical_statistical_tests.html

1.5 Common pitfalls

Some of the advice provided in the literature regarding the validity of statistical tests when assumptions are not met is potentially misleading. In non experimental studies the assumptions of almost any inferential method are rarely met in full. So technically almost all statistical methods might be considered to be "invalid". In reality it is not the violation of an assumption that is important, it is the degree to which an assumption is violated and the influence that any violation may have on the validity of the substantive findings that is really important. If an analysis produces a p-value of < 0.0001 when using a method that leads to minor violations of assumptions, almost any correction will still lead to rejection of the null hypothesis. So a conclusion based only on NHST will not change at all. On the other hand confidence intervals may have to be broadened when the violations are corrected for, so a modification as a result of making a correction for unmet assumptions will take place. This is a powerful additional argument against emphasising NHST alone (naked p-values) that is not mede clear in all the papers cited here. Many students have learned to test "the data"" for normality. This is poor advice, for many reasons, not least of which is that students often test the wrong element in the data The assumption of normality in regresion analysis applies to the residuals, not to the raw date (Zuur et al., 2010). Rigorous diagnostics are a very important part of building statistical models, but these are best conducted through careful inspection of the patterns shown in the residuals rather than through automated statistical tests. A statistical test of normality does not actually test the data anyway. It is, in reality, testing whether the data could have been obtained from an underlying normal distribution. The only way to "test" the data themselves is to look at them carefully. As suggested by Steel et al. (2013), plot the data early and often. When looking at observational data that were not derived from a conventional experimentla design, you should adopt an incremental approach to choosing an analysis. There is nothing to prevent you using the "wrong" models as initial tools for understanding the data and (hopefully) finally choosing a better, more appropriate model as the analysis progresses.

Chapter 2

R programming

During the course of this unit you will develop the skills to be able to apply R code to the analysis of a range of data sets. Although crib sheets will be provided with model code for all the analytical techniques shown on the course, in order to manipulate data effectively and apply novel techniques it is important to begin to become more comfortable with basic R programming techniques and concepts. This week we will revise some of the fundamentals of R programming.

2.1 Using R to simulate data

If you understand data structures you can usually choose and run the right analysis. If you do not consider the basic properties of your data then not only will you not know which analysis to use, but you will probably not even be able to import the data into any software in order to run it!

2.2 Simple R programming

Try running some simple commands by stepping through them in code chunks. When working in RStudio notice that the results are shown under the code chunk when

```r
1+1
```

```
## [1] 2
```

Experiment some more. For example ..

```r
5*10
```

```
## [1] 50
```

```r
12/4
```

```
## [1] 3
```

```r
3^2
```

```
## [1] 9
```

There is a lot more maths functionality built into the R language that you can find out about as you go on. However to follow the primer you do not need to learn any more commands than are shown in this document.

Note that when you type maths in the console the output is just printed to the screen. This is not stored anywhere in the computer's memory. So we need to bring in the concept of data objects. A data object "holds" the numbers and can be manipulated by commands. This is how R manages to run analyses on your data. When you import a file containing data it will be stored into a data object.

The simplest data object is a variable. If a variable only contains one number it is known formally as a scalar. A variable with many numbers is a vector. So let's assign a single value to a variable to form a scalar. We do that in R using the <- operator which is typed as a combination of the less than sign < with a horizontal bar -.

```
x<-5
```

Nothing appears to have happened! However you now have your first data object, a scalar called x. This is stored in memory. So try this.

```
x*2
```

```
## [1] 10
```

Notice that this would not have worked if you had typed X*2 as R is case sensitive.

So, how do we form a vector with multiple values? When you are analysing your own data the answer is that you usually won't need to. You import all the values from a file. But if you wish to form a vector in the console you must use the concatenation operator "c". So this gives the variable x a series of values.

```
x<-c(1,4,5,9,10,11,12,34,56,67,100,123,45)
```

Now see what happens if you type

```
x*2
```

```
##  [1]   2   8  10  18  20  22  24  68 112 134 200 246  90
```

You can carry out any sort of mathematical operation that involves x and all the values in the vector will be used. Notice that the results are just printed out to the console and lost.

If you want to assign the results to a new variable you use the "<-" operator again. This is a very common practice when analysing data. So, say you are intending to work with the natural logarithm of x you might write.

```
logx<-log(x)
```

You can see that this has worked by writing the new variable name so that R prints out the contents to the console.

```
logx
```

```
##  [1] 0.000000 1.386294 1.609438 2.197225 2.302585 2.397895 2.484907
##  [8] 3.526361 4.025352 4.204693 4.605170 4.812184 3.806662
```

This time you can see more clearly the purpose of the indices in the output. The second line starts with the 12th number in the vector.

You can find the names of the data objects that are held in memory by typing ls().

```
ls()
```

```
## [1] "logx" "x"
```

2.2.1 Data structures

Now we can begin looking at data structures. You can ask R to tell you about the structure of any object that it has in memory by typing str().

```r
str(x)
```

```
##  num [1:13] 1 4 5 9 10 11 12 34 56 67 ...
```

```r
str(logx)
```

```
##  num [1:13] 0 1.39 1.61 2.2 2.3 ...
```

So R has told us that both x and logx are numerical vectors. If you have a lot of data you probably do not want to look at all of it at once.

How does this relate to choosing an analysis? We have seen that this particular variable contains numbers. However in statistical analysis we also use "variables"" that don't consist of numbers. They vary in another respect. If you look at any introductory statistical textbook you will see a chapter that defines types of variables in terms of interval, ordinal and scale variables and nominal and ordinal variables.

The sort of analysis that you will be able to run is determined by the sort of variable, or combination of variables, that you are working with.

Let's set up a categorical variable. Experienced users of R often want to replicate values in order to set up some test data as Dytham suggests. So R has a range of functions for making data. The rep function replicates the values.

```r
gender<-rep(c("Male","Female"),each=10)
gender
```

```
##  [1] "Male"   "Male"   "Male"   "Male"   "Male"   "Male"   "Male"
##  [8] "Male"   "Male"   "Male"   "Female" "Female" "Female" "Female"
## [15] "Female" "Female" "Female" "Female" "Female" "Female"
```

Now if we ask R about the data structure it will tell us that we have a character vector.

```r
str(gender)
```

```
##  chr [1:20] "Male" "Male" "Male" "Male" "Male" "Male" "Male" "Male" ...
```

In statistics categorical variables are referred to as factors. Factors are special character vectors with numbered levels. R automatically assumes that any column in a data file that contains non numeric values is a factor and converts the data to that form. In this case we need to tell R to turn the character vector into a factor.

```r
gender<-as.factor(gender)
str(gender)
```

```
##  Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
```

Now R tells us that gender is a factor with two levels. There is no intrinsic order to the levels. R places them in alphabetical order by default. The important element to be aware of is that gender is a variable. It varies in a known and specific way (it has two levels), but it is a variable all the same. You cannot calculate means, medians, standard deviations or any similar summary statistics using a factor. However you can, and will, use them together with numerical variables in many types of analysis.

Let's produce a numerical variable to go alongside this. In the UK the mean height of men is around 176 cm and women 164 cm. So we could produce a vector with these values using this code. What we need to do is to replicate the "expected value" for each gender to be placed alongside the factor level.

```r
height<-rep(c(176,164),each=10)
height
```

```
##  [1] 176 176 176 176 176 176 176 176 176 176 164 164 164 164 164 164 164
## [18] 164 164 164
```

```r
str(height)
```

```
##  num [1:20] 176 176 176 176 176 176 176 176 176 176 ...
```

So now we have another numerical variable. However if we really carried out a survey of people's heights we would be absolutely amazed if we got data like this. Although the numbers represent an estimate of the expected value for each of the ten men and women we know from experience that people's heights vary around this value. In fact they vary quite a lot. The standard deviation is around 6cm. So we need to add some variability. This demonstrates clearly how statistical procedures work. Data consists of two components. It has an underlying structure that we typically want to find out about. In this case there is a difference in heights which is related to gender. There is also random variability, somtimes called the stochastic component. We are usually less interested in this directly, but we need to be very careful to make justifiable assumptions about this variability in order to correctly analyse the data.

Knowing this we can now make our variable more realistic by adding in some simulated values taken from a normal distribution with this standard deviation.

```r
set.seed(1)
height<-height+rnorm(20,sd=6)
height
```

```
##  [1] 172.2413 177.1019 170.9862 185.5717 177.9770 171.0772 178.9246
##  [8] 180.4299 179.4547 174.1677 173.0707 166.3391 160.2726 150.7118
## [15] 170.7496 163.7304 163.9029 169.6630 168.9273 167.5634
```

We may want to round the values to one decimal place to make them equivalent to the sort of measurements we might make in practice.

```r
height<-round(height,1)
```

Now we have two variables that are held in R's memory. We are assuming that they both form part of a simulated data set that we could have obtained if we had measured a stratified random sample of twenty students consisting of ten men and ten women. Let's call the survey "hsurvey"" and make what is known as a data frame to hold the results.

```r
hsurvey<-data.frame(gender,height)
str(hsurvey)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ gender: Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ height: num  172 177 171 186 178 ...
```

So we have now made up data that has the same structure and similar properties to the results we would get from carrying out a simple survey of people's heights. The basic concepts used to do this can be extended to more complex situations and used to help the design of experiments.

A data frame is the basis of almost all statistical analysis. It consists of two or more columns that line up in such a way that the measurements or observations recorded have been taken from the same individual member of a sample. This is often equivalent to a single sheet in a spreadsheet. R tells us that we have one factor and one numerical variable in this case. We might have measured many other variables at the same time and produced a much wider set of data. These may have been either categorical or numerical. Providing that they all "line up"" and correspond to the same individual we have our data frame. Many standard statistical techniques use two variables together. Later on in the course you will see how we can analyse data consisting of more than two variables using multivariate analysis.

The complete data frame is shown below.

```r
hsurvey
```

```
##    gender height
## 1    Male  172.2
## 2    Male  177.1
## 3    Male  171.0
## 4    Male  185.6
## 5    Male  178.0
## 6    Male  171.1
## 7    Male  178.9
## 8    Male  180.4
## 9    Male  179.5
## 10   Male  174.2
## 11 Female  173.1
## 12 Female  166.3
## 13 Female  160.3
## 14 Female  150.7
## 15 Female  170.7
## 16 Female  163.7
## 17 Female  163.9
## 18 Female  169.7
## 19 Female  168.9
## 20 Female  167.6
```

If you want to remove all the other variables you set up from memory and leave just the data frame you could type.

```
remove(x,logx,height,gender)
ls()
```

```
## [1] "hsurvey"
```

Now if we ask R to print height or gender it will not find them. They have been placed within the data frame.

```
height
```

```
##  [1]  2.255  0.865  1.190 -1.336 -1.334 -1.036 -0.684  0.820  0.061  0.635
## [11]  0.616  0.045 -0.002  1.375 -0.060 -0.356  0.094  1.117  0.054  1.627
## [21]  1.786 -0.503  0.883  1.494  0.729  0.367  1.367  1.768  1.671 -0.201
## [31] -0.811 -0.482  2.052 -0.375  0.167 -0.976  0.460 -0.893  2.222 -0.578
## [41]  0.766 -1.005 -0.030  0.170 -0.348
```

```
gender
```

```
##  [1] Male   Male   Male   Male   Male   Male   Male   Male   Male   Male  
## [11] Female Female Female Female Female Female Female Female Female Female
## Levels: Female Male
```

We can refer to the variables by writing out the full name of the data frame and the variable we want, separated by "$"

```
hsurvey$height
```

```
##  [1] 172.2 177.1 171.0 185.6 178.0 171.1 178.9 180.4 179.5 174.2 173.1
## [12] 166.3 160.3 150.7 170.7 163.7 163.9 169.7 168.9 167.6
```

```
hsurvey$gender
```

```
##  [1] Male   Male   Male   Male   Male   Male   Male   Male   Male   Male  
## [11] Female Female Female Female Female Female Female Female Female Female
## Levels: Female Male
```

Or we can attach the data frame. This makes the variables available directly. Notice that if we have several data frames containing variables with the same name this could cause confusion (so I don't usually do this)

```
attach(hsurvey)
```

```
## The following object is masked from d (pos = 38):
##
##     height
## The following object is masked from d (pos = 54):
##
##     height
## The following objects are masked from hsurvey (pos = 69):
##
##     gender, height
```

```
height
```

```
##  [1] 172.2 177.1 171.0 185.6 178.0 171.1 178.9 180.4 179.5 174.2 173.1
## [12] 166.3 160.3 150.7 170.7 163.7 163.9 169.7 168.9 167.6
```

```
gender
```

```
##  [1] Male   Male   Male   Male   Male   Male   Male   Male   Male   Male
## [11] Female Female Female Female Female Female Female Female Female Female
## Levels: Female Male
```

2.2.2 Saving and loading data frames

Now that we have made up some data we might want to save it in the format that we will eventually use to capture our real data. The simplest, most portable data format is a CSV (Comma Separated Variable) file. Such a file can be easily read by all software.

First we must find a place to store the data. The standard practice is to make a separate data directory (folder) for each analysis that you carry out. You then set this as the working directory using the menu options in the R console. Once you have set your working directory (which could be on a USB stick for portability) you can save the data that is held in memory using an R command.

```
write.csv(hsurvey,file="hsurvey.csv",row.names=FALSE)
```

The command has three elements. The first is the name of the data frame that you wish to save. This is not quoted. The second is the name of the file into which you are going to save the data. The third tells R not to add row names to the file (these are not usually necessary).

Data frames are the standard input to all statistical analysis and are of a consistent form. Many different sorts of data summaries and tables can be generated from a data frame quickly by statistical software.

You can check the working directory using this command.

```
getwd()
```

```
## [1] "/home/rstudio/rstudio/AQM_book"
```

```
#In my case I am running R on Linux so the paths look different to Windows
```

To see a list of files in the directory type dir()

```
dir()
##[1] hsurvey.csv
```

If we remove the dataframe we formed we will be left with nothing in R's memory

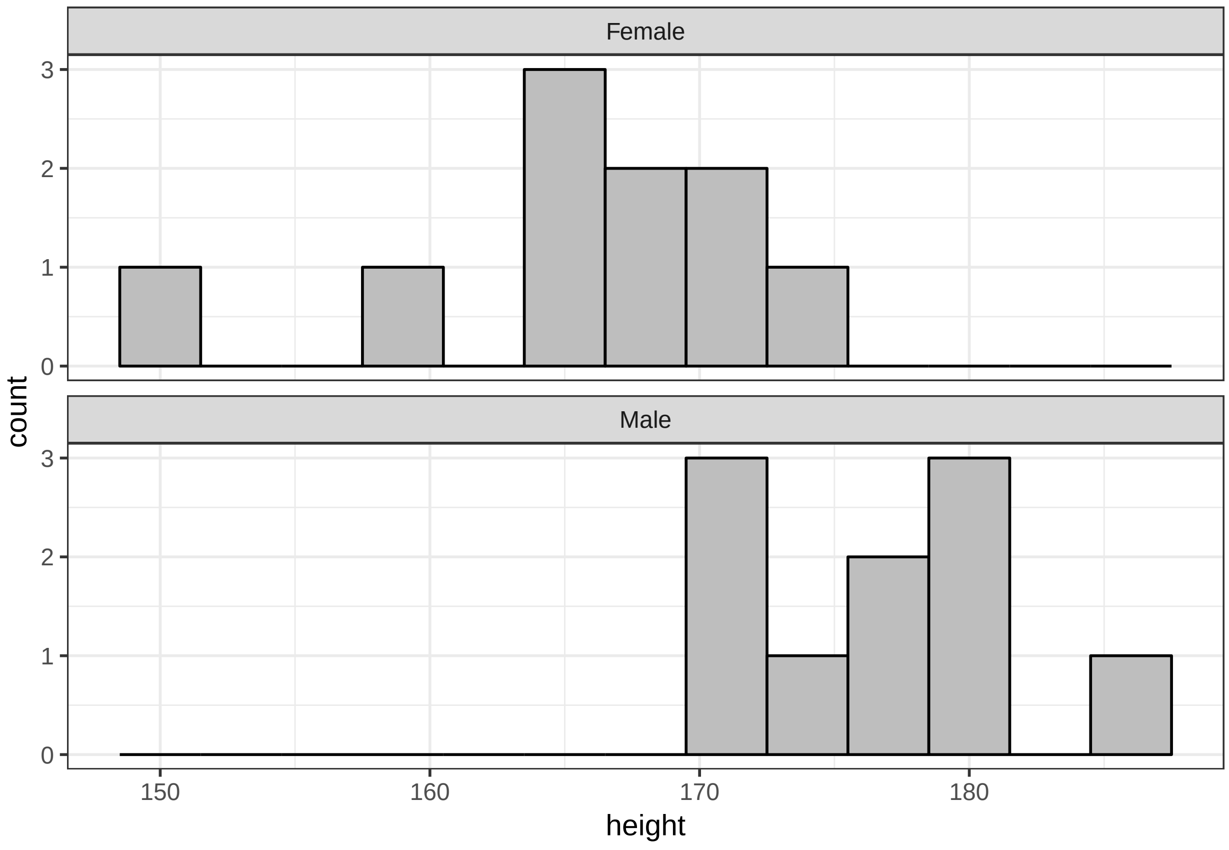
```
remove(hsurvey)
ls()
```

To load it back from the file type

```
hsurvey<-read.csv("hsurvey.csv")
str(hsurvey)
```

2.2.3 Histogram of simulated data

```
library(ggplot2)
g0<-ggplot(hsurvey,aes(x=height))
g0+geom_histogram(fill="grey",colour="black",binwidth=3)+facet_wrap(~gender,nrow=2)
```



So we have simulated some data, saved it, and reloaded the data into R. We have then looked at it as a histogram.

2.2.4 Saving a workspace

You can save the

```
save(list=ls(),file="height_study.rda")
```

2.2.4.1 Installing packages

Many of the more advanced features of R have been implemented within add on packages. These are held online at numerous mirror sites throughout the world. Packages are what makes R the important research tool that it is. Notice that the top articles from Methods in Ecology and Evolution present recently built R packages.

In order to use packages you have to install them to your hard disk. You do this once, and need to be online in order to download them. Once installed you make a package available at the start of a session using the command "library". All the most useful packages have been installed already on the PCs in the lab. You may want to add some packages to your laptop. This can be done either using the menu options at the top of the console or by writing a command (if you know the names of the packages you need).

For example the following line will install the package vegan. This provides access to a large range of useful methods for analysing biodiversity and community data.

```
install.packages("vegan")
```

Once installed you can use vegan by typing

```
library(vegan)
```

During the course I will tell you which packages are most useful, and what they are useful for. I will also show you how to get help on functions in order to work independently. For now, just be aware that if R tells you that "there is no package called vegan"" if you type "library(vegan)" it means that the package has not yet been downloaded. You would then need to install it first.

2.2.4.2 More vector functions

As we have seen functions act on the whole vector at once. This applies also to functions that return only one number as a result.

```
x<-c(1,3,5,7,4,2,7)
sum(x)
```

```
## [1] 29
```

```
mean(x)
```

```
## [1] 4.142857
```

```
length(x)
```

```
## [1] 7
```

```
var(x)
```

```
## [1] 5.47619
```

```
sd(x)
```

```
## [1] 2.340126
```

It is very important to realise that a vector with NA values can cause problems for these sorts of functions. You need to tell R explicitly to remove the NAs. If not the result itself will be an NA. For example.

```
x<-c(1,2,3,4,5,6)
mean(x)
```

```
## [1] 3.5
```

```
x<-c(1,2,3,4,5,NA)
mean(x)
```

```
## [1] NA
```

```
mean(x,na.rm=T)
```

```
## [1] 3
```

This is a very common pitfall for beginners. In general, if something does not work as you expect look for NAs!

2.2.5 Generating sequences of numbers in R

One of the most useful features of R is the ease with which you can generate sequences of numbers and simulated data sets. To produce a sequence you can use the : syntax.

```
x<-0:100
0:100
```

```
##   [1]   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
##  [18]  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33
##  [35]  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50
##  [52]  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67
##  [69]  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84
##  [86]  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
```

```
x<-30:10
x
```

```
##  [1] 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10
```

A longer but more flexible way of producing a sequence is with seq. For example to produce even numbers between 0 and 100.

```
x<-seq(0,100,by=2)
x
```

```
##  [1]   0   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32
## [18]  34  36  38  40  42  44  46  48  50  52  54  56  58  60  62  64  66
## [35]  68  70  72  74  76  78  80  82  84  86  88  90  92  94  96  98 100
```

Say we know the length of the sequence we want but have not worked out the intervals.

```
x<-seq(0,100,length=23)
x
```

```
##  [1]   0.000000   4.545455   9.090909  13.636364  18.181818  22.727273
##  [7]  27.272727  31.818182  36.363636  40.909091  45.454545  50.000000
## [13]  54.545455  59.090909  63.636364  68.181818  72.727273  77.272727
## [19]  81.818182  86.363636  90.909091  95.454545 100.000000
```

2.2.5.1 Using rep to replicate a vector

If we want ten copies of the same vector one after the other we can use.

```
x<-rep(c(1,4,9,23),times=10)
x
```

```
##  [1]  1  4  9 23  1  4  9 23  1  4  9 23  1  4  9 23  1  4  9 23  1  4  9
## [24] 23  1  4  9 23  1  4  9 23  1  4  9 23  1  4  9 23
```

However we might want each number in the vector replicated ten times before moving to the next. In this case we use each instead of times.

```
x<-rep(c(1,4,9,23),each=10)
x
```

```
##  [1]  1  1  1  1  1  1  1  1  1  1  4  4  4  4  4  4  4  4  4  4  9  9  9
## [24]  9  9  9  9  9  9  9 23 23 23 23 23 23 23 23 23 23
```

2.2.5.2 Replicating text

When designing a format to hold the results from a planned experiment or field survey it can be very useful to generate replicated sequences of text for the factor levels or grouping variables. This is also very easy.

```
x<-rep(c("Control","Treatment1","Treatment2"),each=10)
x
```

```
##  [1] "Control"    "Control"    "Control"    "Control"    "Control"   
##  [6] "Control"    "Control"    "Control"    "Control"    "Control"   
## [11] "Treatment1" "Treatment1" "Treatment1" "Treatment1" "Treatment1"
## [16] "Treatment1" "Treatment1" "Treatment1" "Treatment1" "Treatment1"
## [21] "Treatment2" "Treatment2" "Treatment2" "Treatment2" "Treatment2"
## [26] "Treatment2" "Treatment2" "Treatment2" "Treatment2" "Treatment2"
```

Or of course

```
x<-rep(c("Control","Treatment1","Treatment2"),times=10)
x
```

```
##  [1] "Control"    "Treatment1" "Treatment2" "Control"    "Treatment1"
##  [6] "Treatment2" "Control"    "Treatment1" "Treatment2" "Control"   
## [11] "Treatment1" "Treatment2" "Control"    "Treatment1" "Treatment2"
## [16] "Control"    "Treatment1" "Treatment2" "Control"    "Treatment1"
## [21] "Treatment2" "Control"    "Treatment1" "Treatment2" "Control"   
## [26] "Treatment1" "Treatment2" "Control"    "Treatment1" "Treatment2"
```

2.2.6 Logical vectors and subsetting

One of the keys to using R efficiently is the concept of logical vectors, indices and subsetting. However the concept does take a little effort to get used to. Let's take it a step at a time. Say we have a vector x which we have setup like this.

```
x<-seq(-4,10,by=2)
x<-rep(x,times=3)
x
```

```
##  [1] -4 -2  0  2  4  6  8 10 -4 -2  0  2  4  6  8 10 -4 -2  0  2  4  6  8
## [24] 10
```

Now we can ask a question that will be answered as true or false for each of the numbers. Is the element of x greater than zero?

```
x>0
```

```
##  [1] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE
```

R replies with a vector stating whether the result is true or false. We can also ask which of the numbers is greater than zero.

```
which(x>0)
```

```
##  [1]  4  5  6  7  8 12 13 14 15 16 20 21 22 23 24
```

Now R replies with the indices of the elements of the vector.

Subsetting the vector involves the use of the square brackets {[} and {]}. If you include either a logical vector with TRUE and FALSE values or numeric indices within the square brackets R will subset the vector. Either way works.

```
x[x>0]
```

```
##  [1]  2  4  6  8 10  2  4  6  8 10  2  4  6  8 10
```

```
x[which(x>0)]
```

```
##  [1]  2  4  6  8 10  2  4  6  8 10  2  4  6  8 10
```

When we move on to handling more than one variable at a time using data frames we will see that the same concept can be used to subset whole blocks of data.It is a very powerful and fundamentally simple way of manipulating data.

A more complex example is given here. This takes every second member of x using a sequence of indices as a subset.

```
x[seq(2,length(x),by=2)]
```

```
##  [1] -2  2  6 10 -2  2  6 10 -2  2  6 10
```

Could you follow how this worked? Working outwards

- length(x) gives the number of elements in x. Lets call this n
- seq(0,length(x),by=2) gives the vector of indices 2,4,6,….n.
- A subset of x is found using the square brackets.

With experience it is common to wrap up several steps in a single line. There is nothing wrong with explicitly writing

```
n<-length(x)
ind<-seq(2,n,by=2)
x[ind]
```

```
##  [1] -2  2  6 10 -2  2  6 10 -2  2  6 10
```

2.2.6.1 Sorting and ordering

We can also sort vectors. There are two ways of doing this. The first is very direct, but I really do not recommend it. It uses the sort command with the argument decreasing=TRUE or FALSE

```
sort(x,decreasing=T)
```

```
##  [1] 10 10 10  8  8  8  6  6  6  4  4  4  2  2  2  0  0  0 -2 -2 -2 -4 -4
## [24] -4
```

That's simple enough. However it is much better practice in the long run to use order. This needs a little explaining, again we will take it step by step.

```
order(x,decreasing=T)
```

```
##  [1]  8 16 24  7 15 23  6 14 22  5 13 21  4 12 20  3 11 19  2 10 18  1  9
## [24] 17
```

Order has not given us the numbers in order! But of course it should not, as sort does that. Instead order has given us the the indices in order. Notice that if there are ties, as in this case, R respects the original order for the tied numbers. So how can we use order to sort the vector? If you have followed the logic of using indices to refer to elements of a vector you might have guessed.

```
x[order(x,decreasing=T)]
```

```
##  [1] 10 10 10  8  8  8  6  6  6  4  4  4  2  2  2  0  0  0 -2 -2 -2 -4 -4
## [24] -4
```

Although this involves more typing than simply writing sort, it is more powerful. The power comes from the way indices can be used with many variables at once. When we move on to see data frames this will be clearer. For the moment look at this simple example to see the logic.

```
x<-c(1,3,2,4)
y<-c("A","B","C","D")
y[order(x,decreasing=F)]
```

```
## [1] "A" "C" "B" "D"
```

2.2.6.2 Ranks

Finding ranks is a very common procedure in statistics. This is a slightly different problem. We need a way to deal with ties, and there is more than one solution to the problem.

```
x<-c(1,1,2,3,4,5,5)
rank(x,ties="average")
```

```
## [1] 1.5 1.5 3.0 4.0 5.0 6.5 6.5
```

```
rank(x,ties="max")
```

```
## [1] 2 2 3 4 5 7 7
```

```
rank(x,ties="min")
```

```
## [1] 1 1 3 4 5 6 6
```

The last example coincides with the familiar ranks given in sports (joint gold medal followed by the bronze). Notice that there is no decreasing argument to ranks. The lowest numbers take the lowest ranks. If you really want to rank performance scores you must reverse them first by, say, subtracting all the scores from the maximum possible.

```
x<-c(1,1,2,3,4,5,5)
rank(max(x)-x,ties="min")
```

```
## [1] 6 6 5 4 3 1 1
```

2.3 Exercise

In order to practice putting together these commands in order to simulate some data try this.

A researcher is interested in looking at the differences in diameters of trees in three different woodland sites in the New Forest. At each site there are several different species. In order to simplify the task we will consider only two types of trees … conifers and broadleaves. We will also simplify the exercise by assuming that the same number of trees (50) are sampled in each woodland.

Set up a dataframe with three columns. One column represents the site. The second represents the type of tree (i.e. conifer or broadleaf). The third represents the diameters. So there will be 150 observations (rows) in all. Try to produce data in which there is a difference in mean diameter that is affected by the site from which the measurements are taken and the type of tree being measured. You can assume that the random variation in diameters is normally distributed and that measurements are taken to the nearest cm.

Chapter 3

Introduction to statistical modelling

In most cases the relationship between a traditional statistical test and the model is not explicit, although one way anova is based on an underlying statistical model. Statistical modelling can allow you to extract more information from your data. Most contemporary papers in ecology use models of some kind. Even if the nature of the data you collect limits your options, it is very important to learn to fit and interpret statistical models in order to follow the literature.

3.1 What is a statistical model?

To some extent statistical modelling is easier than other forms of ecological modelling. Building process based models requires a great deal of understanding of a system. Statistical models are built from the data themselves. Providing you do have some data to work with you can let the combination of data and prebuilt algorithms find a model. However there are many issues that make statistical modelling challenging.

A statistical model is a formal mathematical representation of the "sample space"", in other words the population from which measurements could have been drawn. It has two components.

- An underlying "deterministic" component, that usually represents a process of interest
- A stochastic component representing "unexplained" variability So, a statistical model effectively partitions variability into two parts. One part represents some form of potentially interesting relationship between variables. The other part is just "random noise". Because statistics is all about variability, the "noise" component is actually very important. Variability must be looked at in some detail on order to decide on the right model. Many of the challenges involved in choosing between statistical models involves finding a way to explain the "unexplained" variability.

3.1.1 Uses of models

The literature on statistical modelling frequently uses the terms "explanation"" and "prediction"" to describe the way models are used. Although the same model can have both roles, it is worth thinking about the difference between them before fitting and interpreting any models.

3.1.1.1 Prediction

Models can be used to predict the values for some variable when we are given information regarding some other variable upon which it depends. An example is a calibration curve used in chemistry. We know that conductivity and salinity are directly related. So if we measure the conductivity of liquids with known salinities we can fit a line. We can then use the resulting model to predict salinity at any point between the two extremes that we used when finding the calibration curve. Notice that we cannot easily extrapolate beyond the range of data we have. We will see how the same concept applies to the models we use in quantiative ecology.

3.1.1.2 Explanation

The idea that the variability in the data can be "explained" by some variable comes from the terminology that was used when interpretating experimental data. Experiments usually involve a manipulation and a control of some description. If values for some response differ between control and intervention then it is reasonable to assume that the difference is "explained" by the intervention. If you wish to obtain data that are simple to analyse and interpret you should always design an experiment. However, experiments can be costly. Ecological systems are often slow to respond to interventions, making experiments impossible within the time scale of a master's dissertation. We are often interested in systems that cannot be easily modified anyway on ethical or practical grounds. Thus in ecology we often have to interpret associations between variables as evidence of process that "explain" a relationship. Correlation is not causation. Although patterns of association can provide insight into causality they are not enough to establish it. So, when you read the words "variance explained" look carefully at the sort of data that are being analysed. In an ecological setting this may only suggest close association between variables, not a true explanation in the everyday sense of the word.

3.2 The general linear model

General linear models lie behind a large number of techniques. These have different names depending on the type of data used to explain or predict the variability in a numerical response variable.

- Regression (Numerical variable)
- Analysis of variance (One or more categorical variables)
- Analysis of covariance (Categorical variable plus numerical variable)
- Multiple regression (Multiple numerical variables)

Although these analyses are given different names and they appear in different parts of the menu in a program such as SPSS, they are all based on a similar mathematical approach to model fitting. In R the same model syntax is used in all cases. The steps needed to build any linear model are…

- Look at the data carefully without fitting any model. Try different ways of plotting the data. Look for patterns in the data that suggest that they are suitable for modelling.
- Fit a model: The standard R syntax for a simple linear regression model is **mod<-lm(y~x)** However model formulae may be much more complex.
- Look at whether the terms that have been entered in the model are significant: The simplest R syntax is *anova(mod).* Again, this part of the process can be become much more involved in the case of models with many terms.
- Summarise the model in order to understand the structure of the model. This can be achieved with* **summary(mod)**
- Run diagnostics to check that assumptions are adequately met. This involves a range of techniques including statistical tests and graphical diagnoses. This step is extremely important and must be addressed carefully in order to ensure that the results of the exercise are reliable.

3.3 Regression

3.3.1 Theory

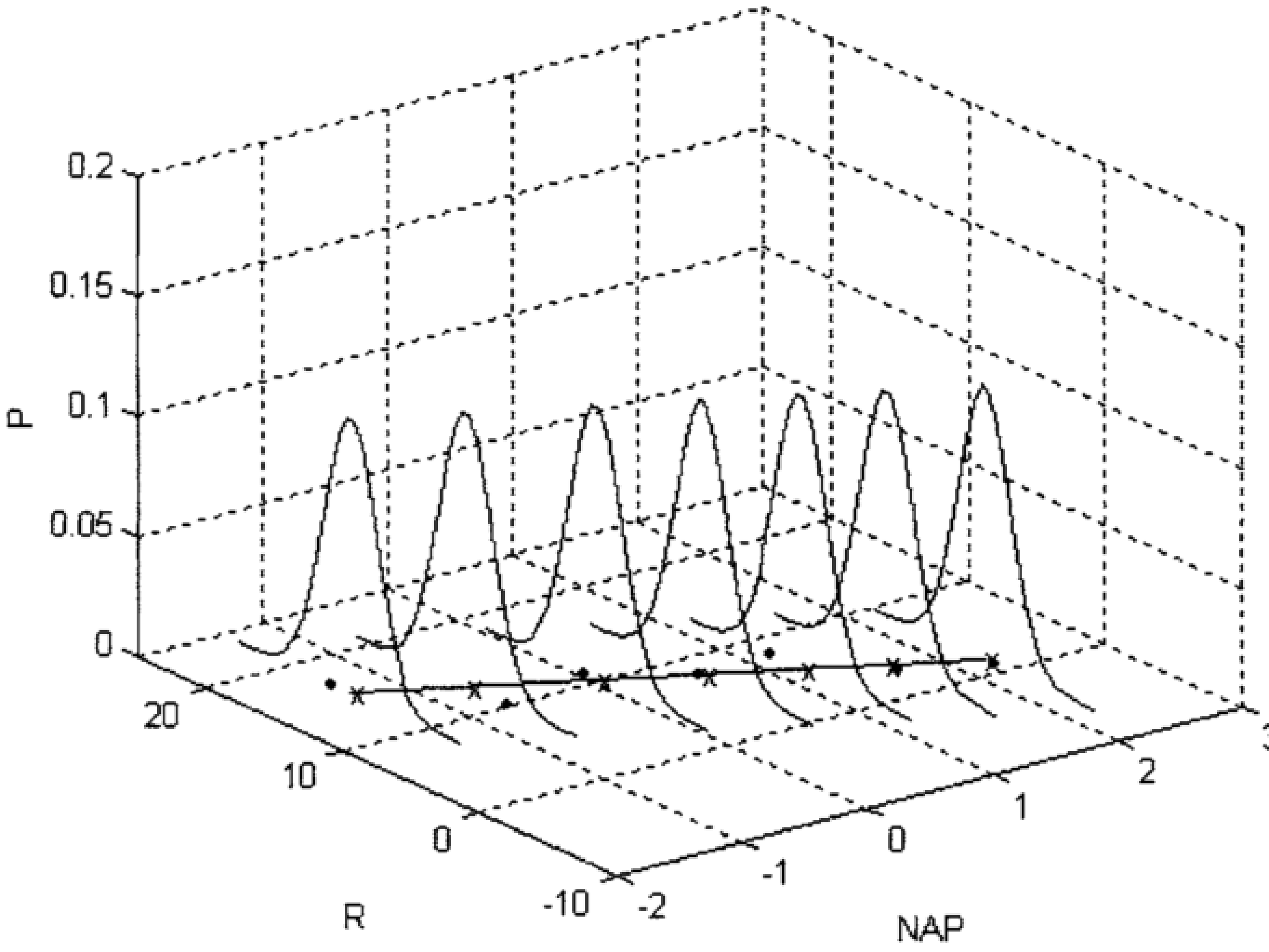
The regression equation is ..

$y = a + bx + \epsilon$ where $\epsilon = N(o, \sigma^2)$

In other words it is a straight line with a as the intercept, b as the slope, with the assumption of normally distributed errors (variability around the line)

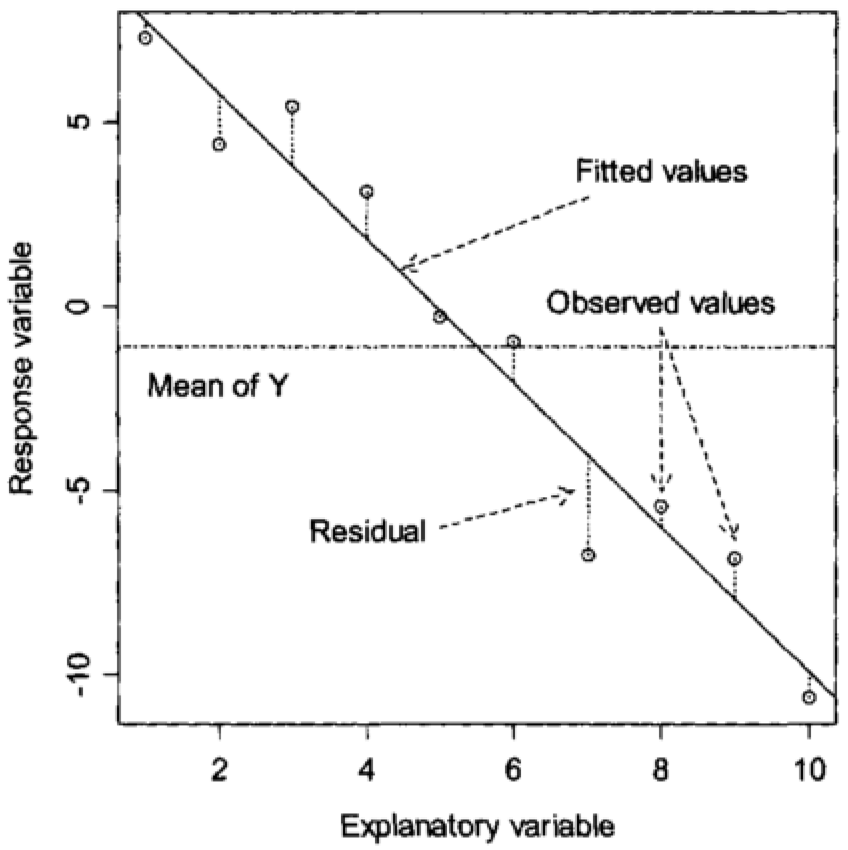
The diagram below taken from Zuur et al (2007) illustrates the basic idea. In theory, if we had an infinite number of observations around each point in a fitted model, the variability would form a perfect normal distribution. The shape and width of the normal curves would be constant along the length of the model. These normal curves form the stochastic part of the model. The strait line is the deterministic part. This represents the relationship we are usually most interested in. The line is defines by its intercept with the axis and its slope.

```r
knitr::include_graphics("figs/regression1.png")
```



For any observed value of y there will be a fitted value (or predicted value) that falls along the line. The difference between the fitted value and the observed value is known as the residual.

```
knitr::include_graphics("figs/regression.png")
```



In reality we do not have infinite observed values at each point. However if we collected all the residuals together we should get a normal distribution.

3.3.2 Example

The example is taken from Crawley's R book. It is very simplified and idealised, but serves to explain the concepts. Crawley's own ecological research involves herbivory. Here he presents an example in which the growth of insect larvae on leaves with different concentrations of tannin has been measured. Tannin concentrations slow insect growth (Crawley unfortunately does not provide the units in which this is measured). The question is, how much is growth inhibited by an increase of 1% in tannin concentration? The first step after loading the data is to produce a scatterplot.

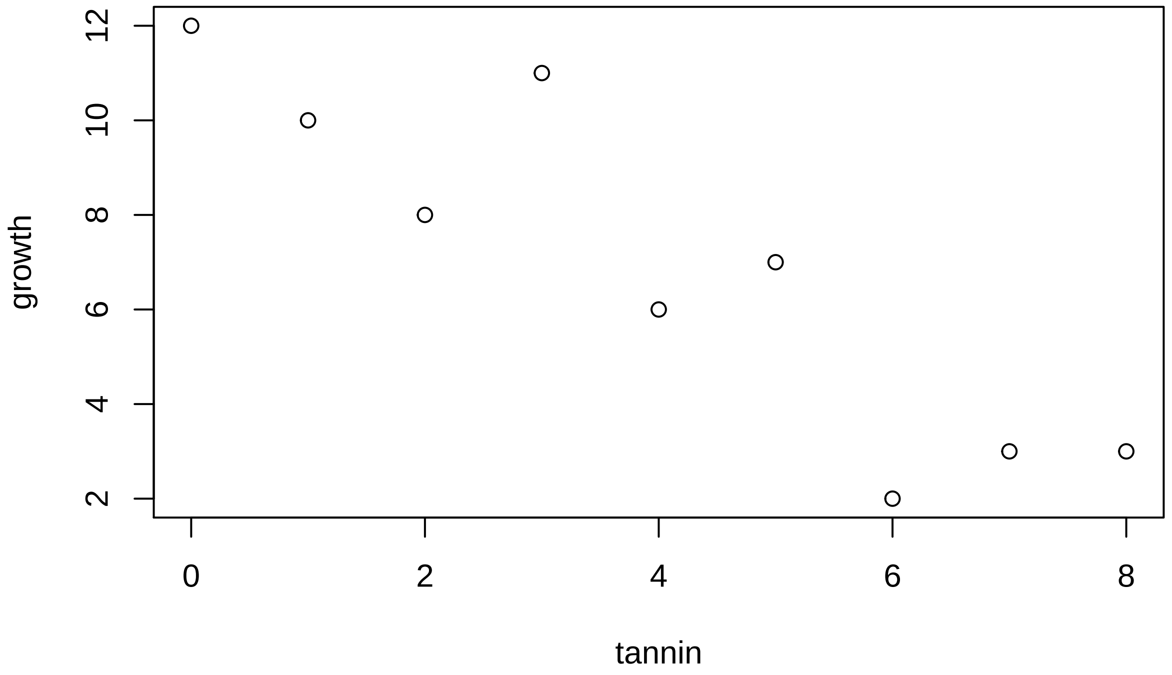
```
larvae<-read.csv("https://tinyurl.com/aqm-data/larvae.csv")
names(larvae)
```

```
## [1] "growth" "tannin"
```

```
attach(larvae)
```

```
## The following objects are masked from larvae (pos = 65):
##
##     growth, tannin
```

```r
plot(growth~tannin)
```



The scatterplot is produced using the syntax that we will use in the model. Growth is a function of (~) tannin.

We now fit a model and assign the results to an object in R. We can call this "mod". This contains all the information about the fitted model.

```r
mod<-lm(growth~tannin)
```

We can now look at the properties of the model. If we ask for an "anova". R will produce the following output.

```r
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: growth
##           Df Sum Sq Mean Sq F value    Pr(>F)
## tannin     1 88.817  88.817  30.974 0.0008461 ***
## Residuals  7 20.072   2.867
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This is similar to the table produced for an ANOVA involving categegorical variables (factors). It can be interpreted in a similar way. Just as for an ANOVA we have an F ratio that represents the amount of variability that falls along the regression line divided by the residual variation.

The numerator degrees of freedom for a simple regression is always one. The numerator degrees of freedom is n-2 as we have estimated two parameters, the slope and the intercept. Thus we have found a very significant effect of tannin concentration on growth F(1, 7) =31, p <0.001. You should report the R^2 values along with the p-value.

A correlation test would have shown the same significant relationship, but without as much detail.

```
cor.test(growth,tannin)
```

```
##
##  Pearson's product-moment correlation
##
## data:  growth and tannin
## t = -5.5654, df = 7, p-value = 0.0008461
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9796643 -0.5972422
## sample estimates:
##        cor
## -0.9031408
```

The additional information that we have with a regression concerns the two elements in the regression equation. The slope and the intercept.

```
summary(mod)
```

```
##
## Call:
## lm(formula = growth ~ tannin)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4556 -0.8889 -0.2389  0.9778  2.8944
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.7556     1.0408  11.295 9.54e-06 ***
## tannin       -1.2167     0.2186  -5.565 0.000846 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.693 on 7 degrees of freedom
## Multiple R-squared:  0.8157, Adjusted R-squared:  0.7893
## F-statistic: 30.97 on 1 and 7 DF,  p-value: 0.0008461
```

Remember the equation we used.

$y = a + bx + \epsilon$

The intercept (a) is 11.76

The slope (b) is -1.22

Which means that growth (whatever this was measured in by Crawley) is reduced by 1.22 for each increase of 1% in tannin concentration.

3.3.3 Confidence intervals

There is an issue with this way of summarising the results. If we only report the coefficients as given we have ignored the fact that there was unexplained variation in the model. This variation produces uncertainty regarding the true values of the parameters. The greater the unexplained variation (scatter or noise) around the line, the less confident we can be regarding their values. The statistical mechanism used in the calculations (that is not particularly complex, but can safely be taken on trust) allows us find confidence intervals for the parameters. If the confidence intervals do not include zero then the parameters are significant. There are only a few cases where this is very meaningful for the intercept. We are usually more interested in the slope.

```
confint(mod)
```

```
##                  2.5 %     97.5 %
## (Intercept)  9.294457 14.2166544
## tannin      -1.733601 -0.6997325
```

So now we can go a bit further. Instead of giving a point estimate for the effect of tannin on growth we can state that the 95% confidence interval for the effect of a 1% increase of tannin on growth lies between -1.73 and -0.7

3.3.4 Prediction

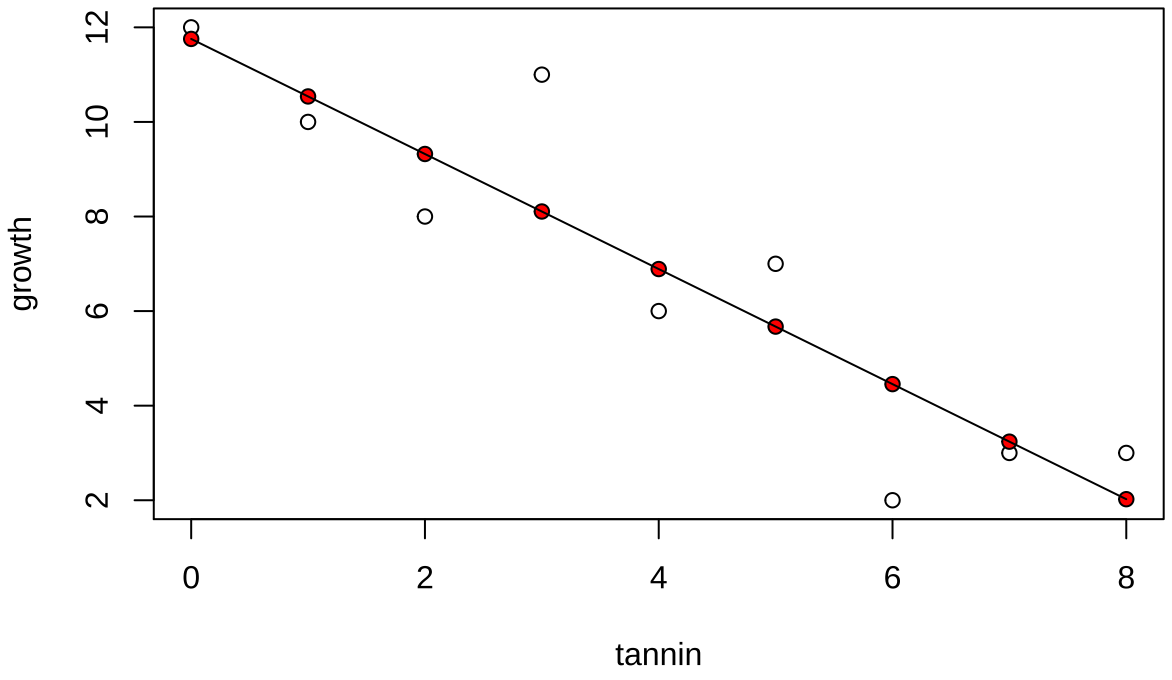
The equation can be used to predict the most likely value of y given a value of x. If we just ask R to "predict" we get the fitted values for the values of the explanatory variable that we used when fitting the model.

```
predict(mod)
```

```
##         1         2         3         4         5         6         7 
## 11.755556 10.538889  9.322222  8.105556  6.888889  5.672222  4.455556 
##         8         9 
##  3.238889  2.022222
```

So we can plot out the data again and show the predicted values as red points and draw the regression line.

```
plot(growth~tannin)
points(tannin,predict(mod),pch=21,bg=2)
lines(tannin,predict(mod))
```

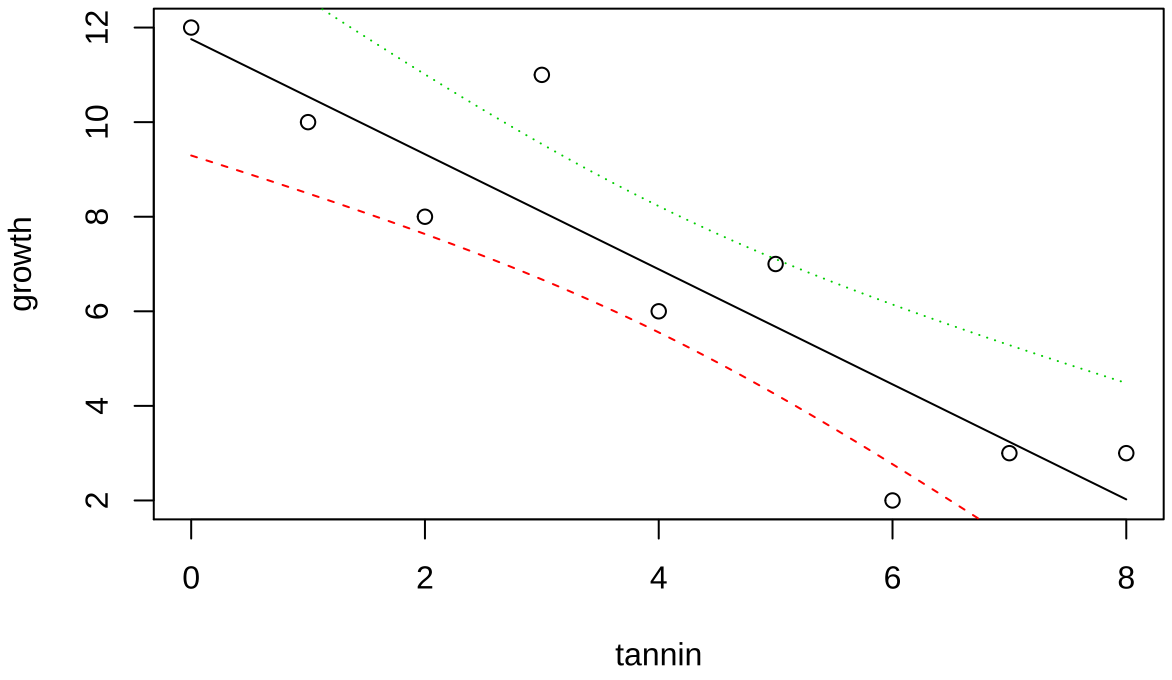


3.3.4.1 Prediction with confidence intervals

As we have seen, a statistical model takes into account uncertainty that arises as a result of variability in the data. So we should not simple look at the line of best fit as summing up a regression. We should add some indication of our confidence in the result to the figure.

To achieve this nicely in R requires a couple of extra steps. After plotting the data with plot(growth~tannin) we set up a sequence of 100 x values that lie between the minimum and the maximum. Now if we pass these to the predict function in R and ask for confidence intervals (that by default are 95%) we get the figure below.

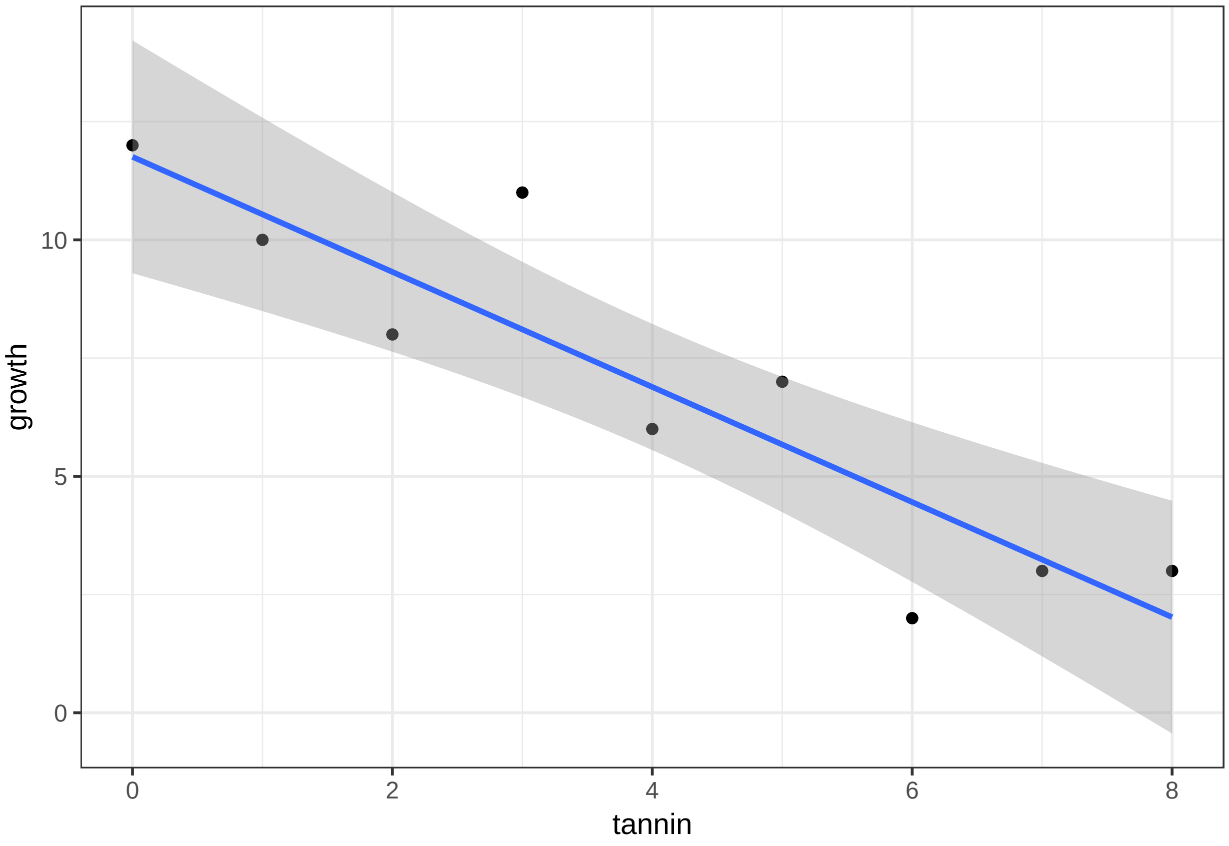
```
plot(growth~tannin)
x<-seq(min(tannin),max(tannin),length=100)
matlines(x,predict(mod,list(tannin=x),interval="confidence"))
```



The confidence bands refer to the fitted model. They show uncertainty regarding the regression line and are calculated from the standard error.

3.4 Using ggplot2 for confidence intervals.

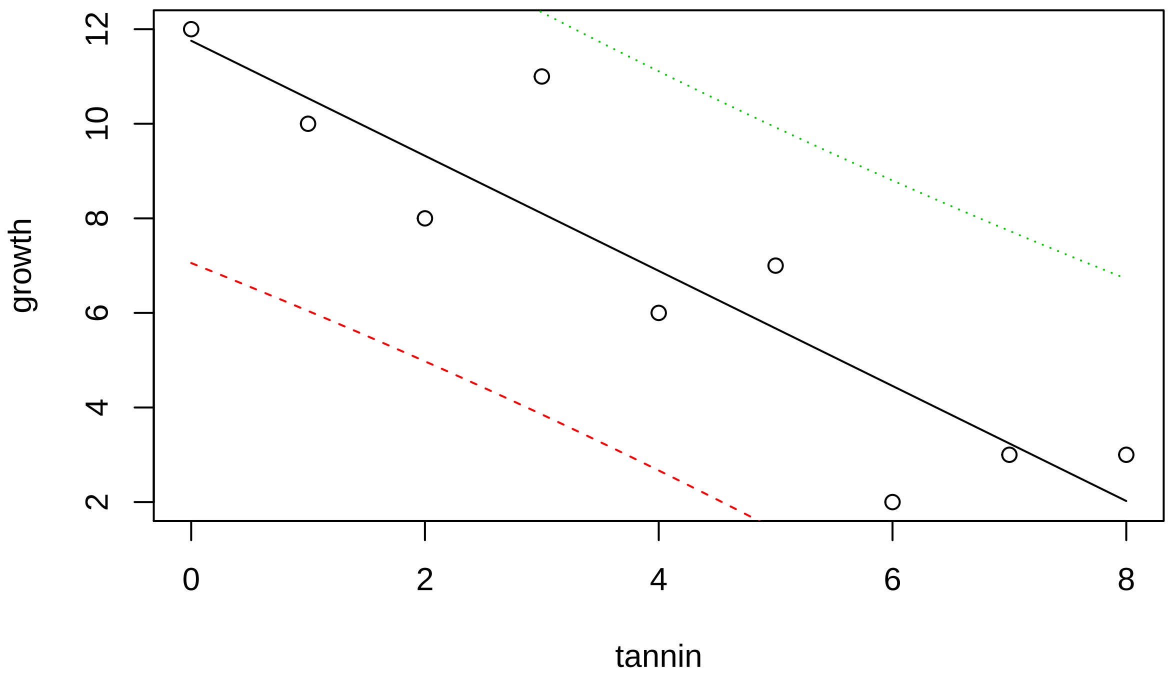
```
library(ggplot2)
g0<-ggplot(data=larvae, aes(x=tannin,y=growth))
g1<-g0+geom_point()
g1+geom_smooth(method="lm")
```



3.4.1 Prediction intervals

There is another way to look at uncertainty. If we know the value for tannin, where would we expect a single measured value for growth to lie. Notice that this is different. The confidence bands show where the mean value migh lie if we measured growth many times. But in this case we want to know where we might expect a data point to fall. This is a much broader interval, and is based on the idea of theoretical normal curves falling around the regression line with a standard deviation estimated from the data. We cut off the tails of these curves.

```
plot(growth~tannin)
x<-seq(min(tannin),max(tannin),length=1000)
matlines(x,predict(mod,list(tannin=x),interval="prediction"))
```



3.4.2 Diagnostics

It is not enough to simply fit a regression line, or any other model. We have to justify our choice of model and convince those who might use it that the assumptions have been adequately met. The question of how close the data are to meeting the assumptions requires model diagnostics.

The basic assumptions for regression

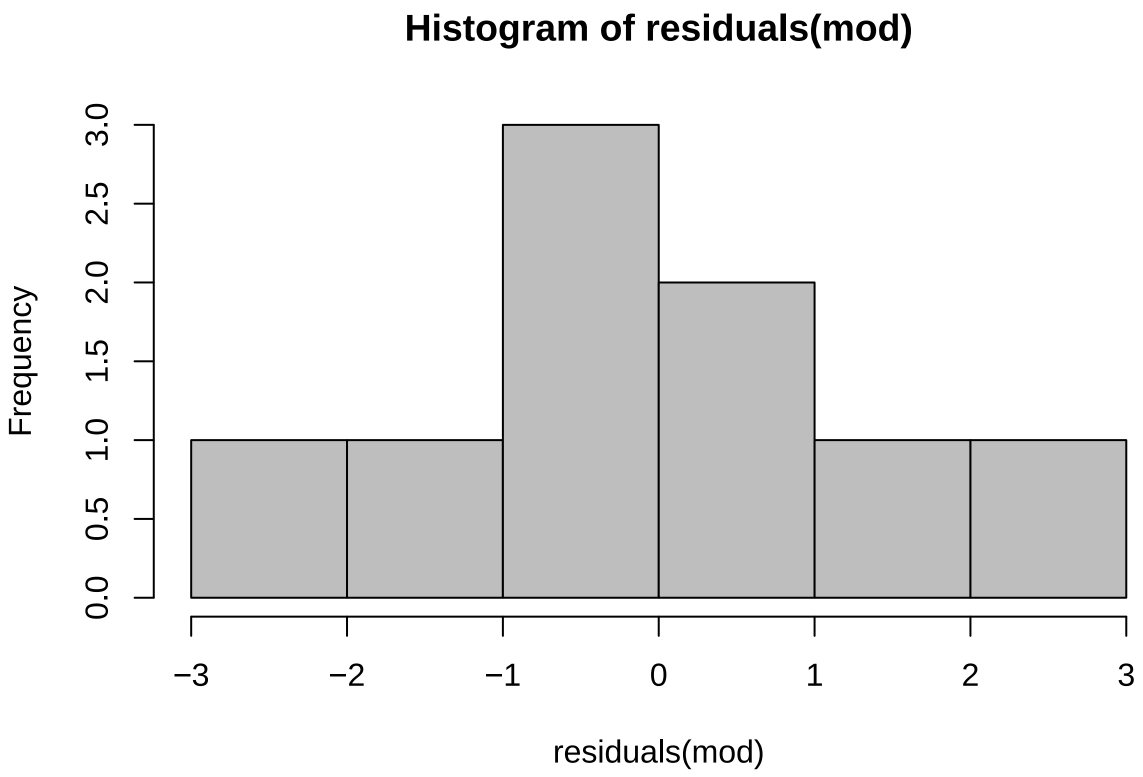
- Normally distributed errors
- Identically distributed errors over the range (homogeneity of variance)
- No undue influence of points with high leverage
- An underlying linear relationship
- Independent errors

In this rather artificial example all the assumptions are met. However real ecological data is rarely as simple as this. In order to justify the use of a regression model you must be able to show that the assumptions are not *seriously* violated. We will come on to what "seriously"" means later.

3.4.2.1 Normality

It is important to remember that the assumption of normality applies to the residuals after a model has been fitted. You do *not* test the for normality of the variable itself, as the relationship that you are modelling influences the distribution of the variable. You can look at the distribution of the residuals by plotting a histogram.

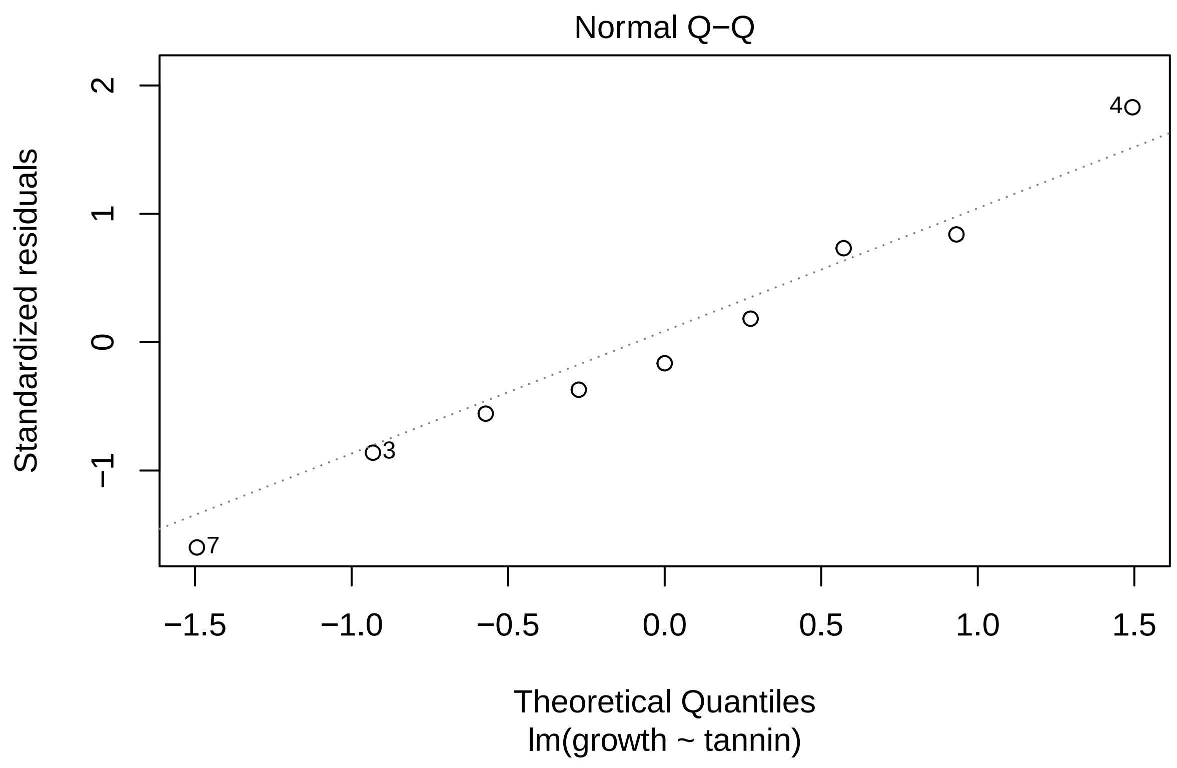
```
hist(residuals(mod),col="grey")
```



This looks good enough. A slightly more sophisticated way of spotting deviations from normality is to use a qqplot.

If we ask R to plot a fitted model, we actually get a set of diagnostic plots. There are six of these. By default R will produce four of them. The qqplot is the second in the series. It is used as a visual check of the normality assumption. If the assumption is met the points should fall approximately along the predicted straight line.

```r
plot(mod,which=2)
```



Qqplots often have a sigmoid shape. This occurs when some of the extreme values fall further along the tail of the normal distribution than expected. Providing this effect is slight it is not a problem. However deviations from the line away from the extremes does show a major departure from the assumption. Interpretation of QQplots requires some experience, but they can be very useful.

If you have doubts about your ability to interpret the plot you could try using a statistical test of normality.

```
shapiro.test(residuals(mod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(mod)
## W = 0.98794, p-value = 0.9926
```

The problem with this test is that it becomes more sensitive as the number of observations increase. Thus you are more likely to get a p-value below 0.05 if you have a lot of data. However violations of normality become much less serious as the number of observations increase. So it can be safe to ignore a significant test result if the QQplot and histogram do not suggest major problems providing you have more than around 30 observations. If the histogram shows clear skew then you should think about a data transform, weighted regression or using a generalised linear model. More about this later.

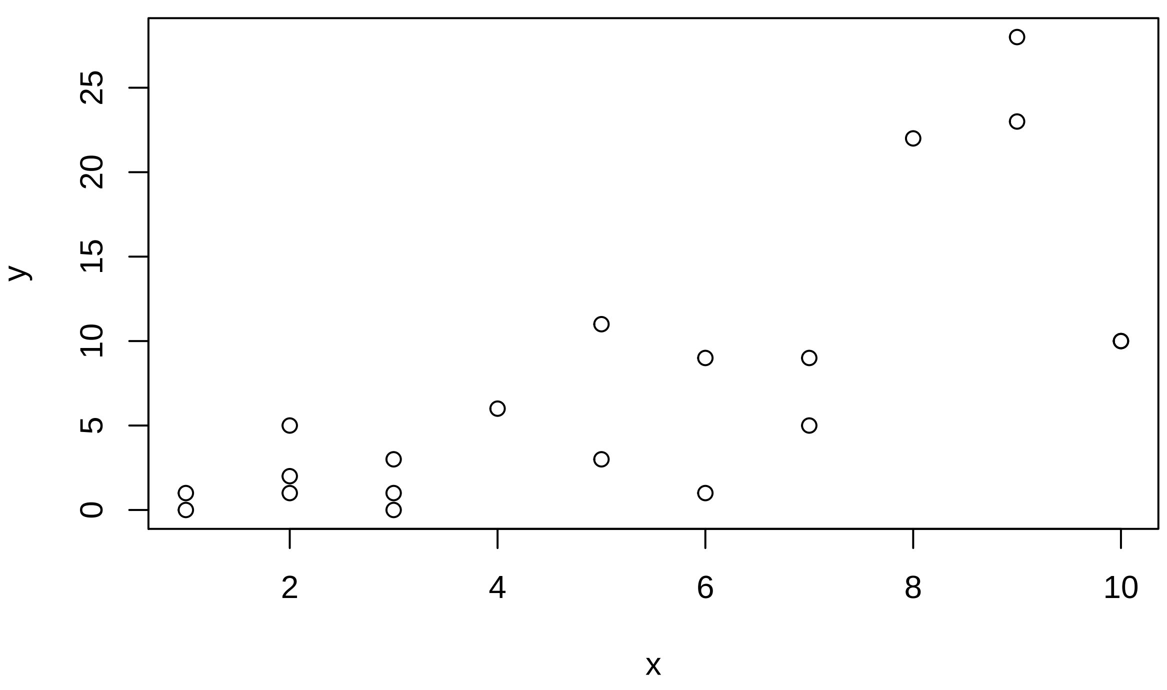
3.4.2.2 Homogeneity of variance

The assumption here is that the distance a point is likely to fall from the fitted line does not change along the x values. This is very often violated. For example if the values represent counts of individuals it is constrained to not fall below zero and will tend to be some function of the expected value. This results in residuals that follow a poisson distribution or more likely, a negative binomial. The characteristic of both these distributions is that the scatter increases as the expected value increases.

The following lines produce some data with this characteristic in order to illustrate the pattern.

```r
set.seed(5)
```

```r
library(MASS)
x<-sample(0:10,20,rep=T)
y<-rnegbin(20,mu=x,theta=2)
plot(y~x)
```

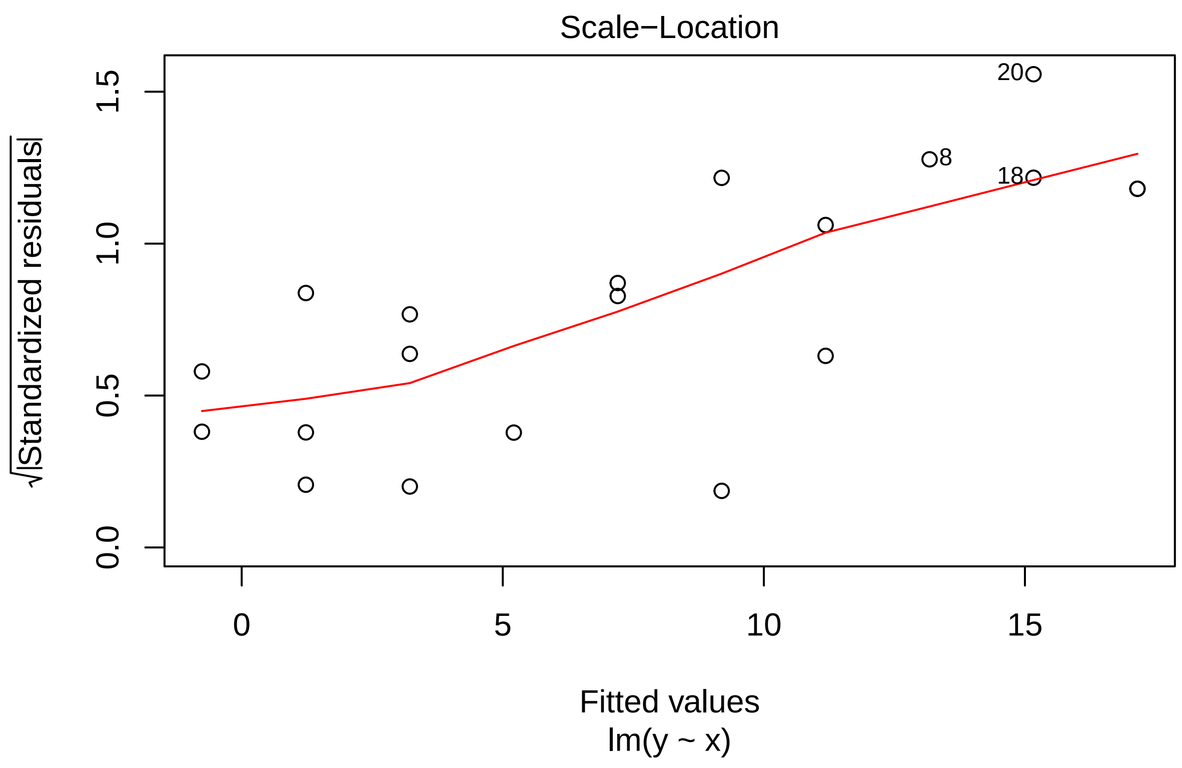


```r
mod.negbin<-lm(y~x)
```

You should be able to spot a “fan” effect.

The third diagnostic plot in the series is used to see this effect more clearly. This plot shows the square root of the standardised residuals plotted against the fitted values. Because the square root is used all the values are positive. If there is an increasing (or decreasing) trend in the absolute size of the residuals it should show up on the plot.

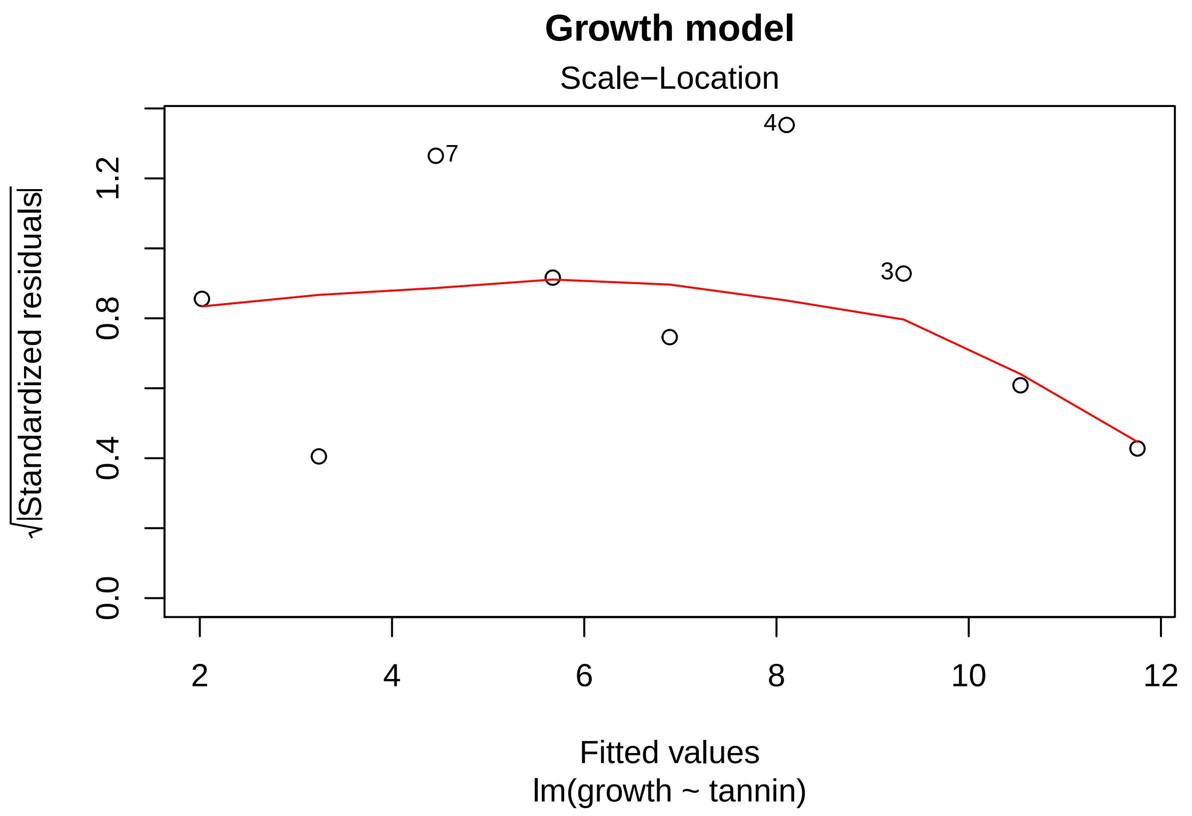
```r
plot(mod.negbin,which=3)
```



The red line is meant to guide the eye, but you should look carefully at the pattern rather than the line itself.

This is the pattern for the original model, that does not have a problem with heterogeneity of variance.

```
plot(mod,which=3,main="Growth model")
```



If you are having difficulty interpreting the results you could try a test for the significance of the trend using a correlation test between the variables shown on the diagonistic plot.

```r
cor.test(sqrt(stdres(mod.negbin)),predict(mod.negbin))
```

```
## Warning in sqrt(stdres(mod.negbin)): NaNs produced

##
##  Pearson's product-moment correlation
##
## data:  sqrt(stdres(mod.negbin)) and predict(mod.negbin)
## t = 4.8093, df = 7, p-value = 0.001945
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5071470 0.9737071
## sample estimates:
##       cor
## 0.8761687
```

```r
cor.test(sqrt(stdres(mod)),predict(mod))
```

```
## Warning in sqrt(stdres(mod)): NaNs produced

##
##  Pearson's product-moment correlation
##
## data:  sqrt(stdres(mod)) and predict(mod)
## t = -0.50019, df = 2, p-value = 0.6666
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9803574  0.9236405
## sample estimates:
##        cor
## -0.3334484
```

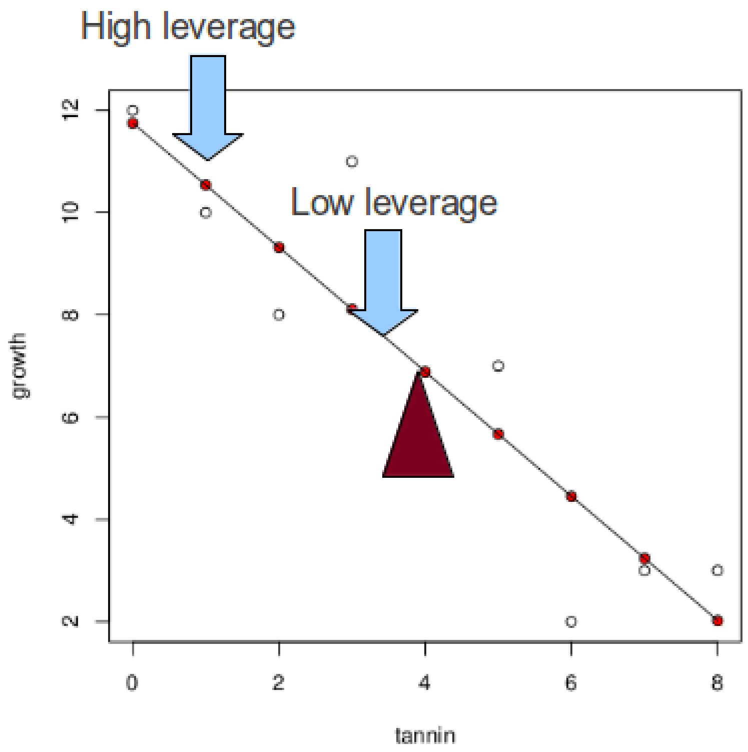
The same sort of caveats apply to the literal interpretation of this test as a decision rule as the normality test. Although most violations of homogeneity of variance have to be taken seriously, large data sets may show statistically significant, but inconsequential violations.

3.4.2.3 Leverage

In the artificial example provided by Crawley all the points are equally spaced along the x axis. This is the "classic" form for a regression and prevents problems with leverage. However in ecology we often have to analyse data that does not have this property. Some of the x values on a scatterplot may fall a long way from the centre of the distribution. Such points potentially could have an excessive influence on the form of the fitted model.

The influence a point has on the regression is a function of leverage and distance from a line fitted using all the other points.

```
knitr::include_graphics("figs/leverage.png")
```



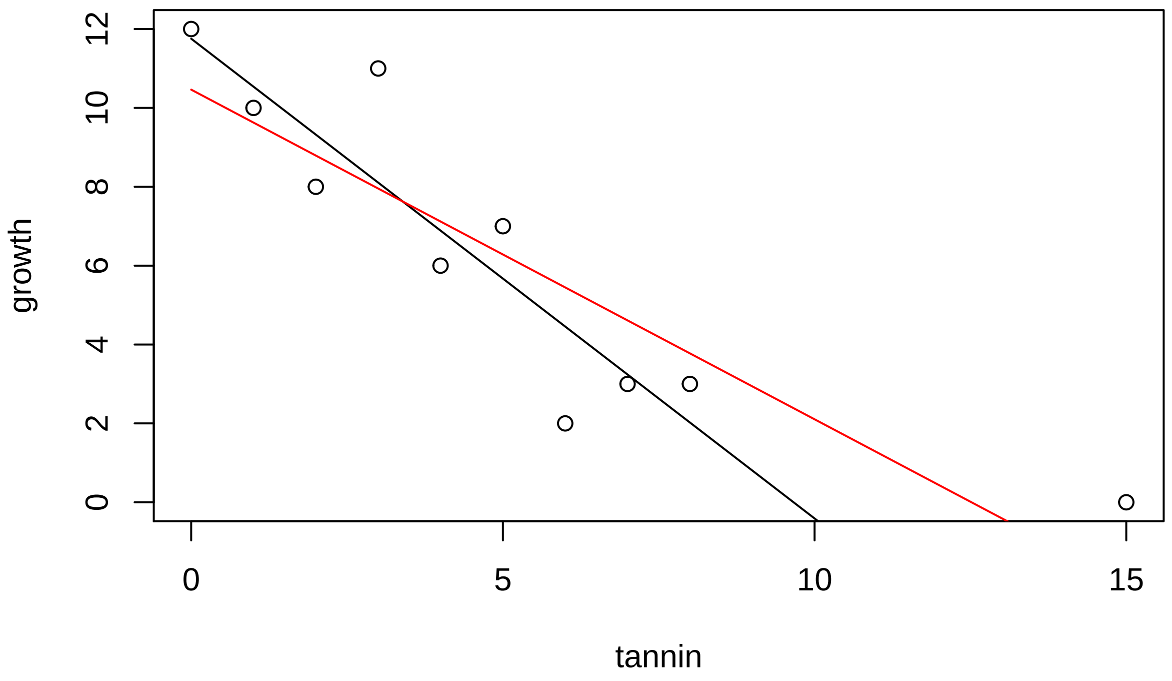
To illustrate this let's add a point to the data at a high tannin density and assume that zero growth was measured at this value. How much does this new point influence the model?

```
tannin<-c(tannin,15)
growth<-c(growth,0)
```

We can plot the new data and show the effect of this one point on the regression line. It is shown in red.

```
new.mod<-lm(growth~tannin)
plot(growth~tannin)
```

```r
lines(tannin,predict(mod,list(tannin)))
lines(tannin,predict(new.mod,list(tannin)),col=2)
```

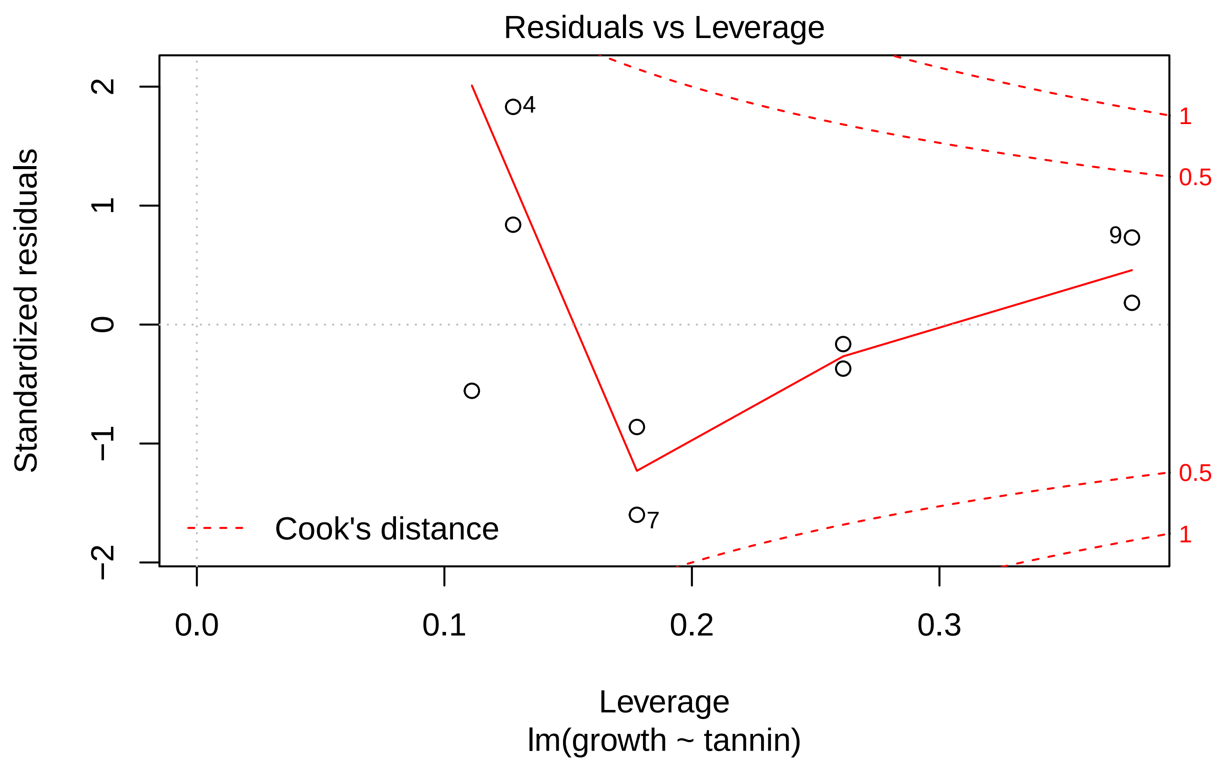


This data point has shifted the regression quite a lot due to a combination of its distance from the others (which gives it high leverage) and the fact that it lies a long way from the fitted line (high residual deviation). These two effects are captured by a quantity known as Cook's distance.

The diagnostic plot shows leverage on the x axis and standardised residuals on the y axis. Large residuals with low leverage do not affect the model, so the safe area with low values for Cook's distance (below 1) forms a funnel shape. Points falling outside this "safe area" can be flagged up.

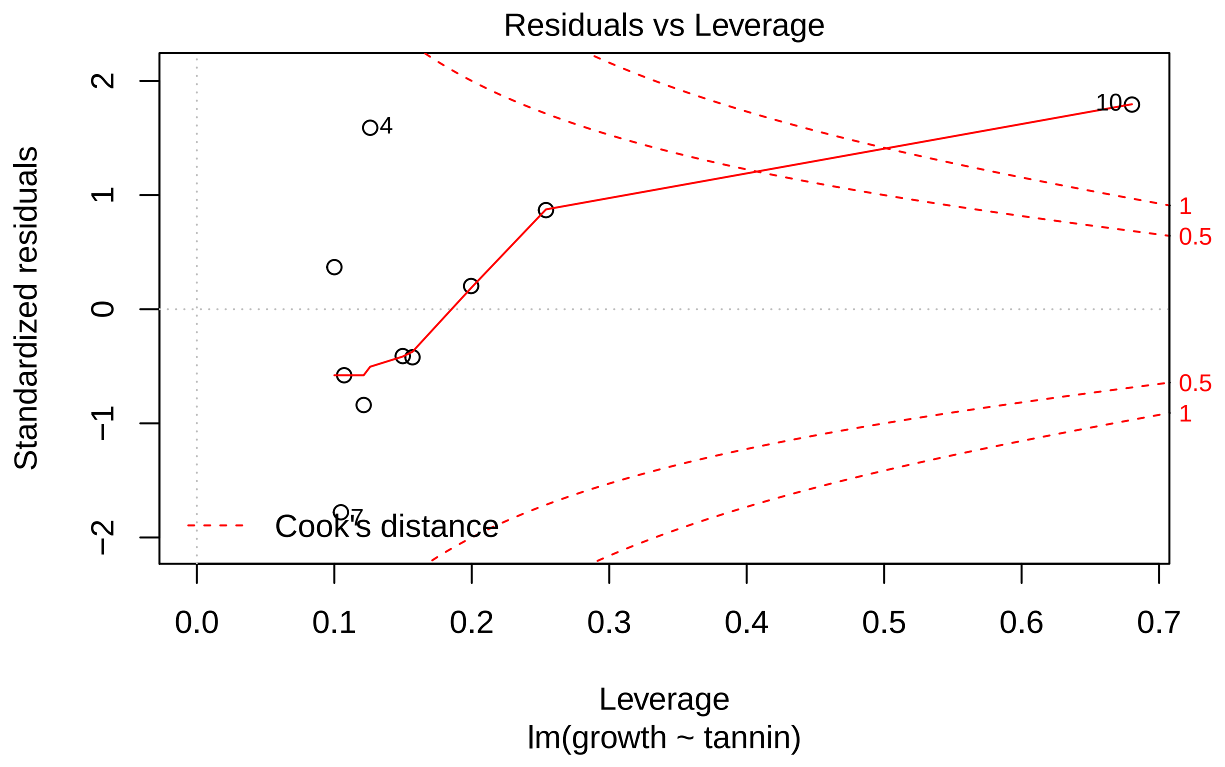
There are no problem points in the original model.

```r
plot(mod,which=5)
```



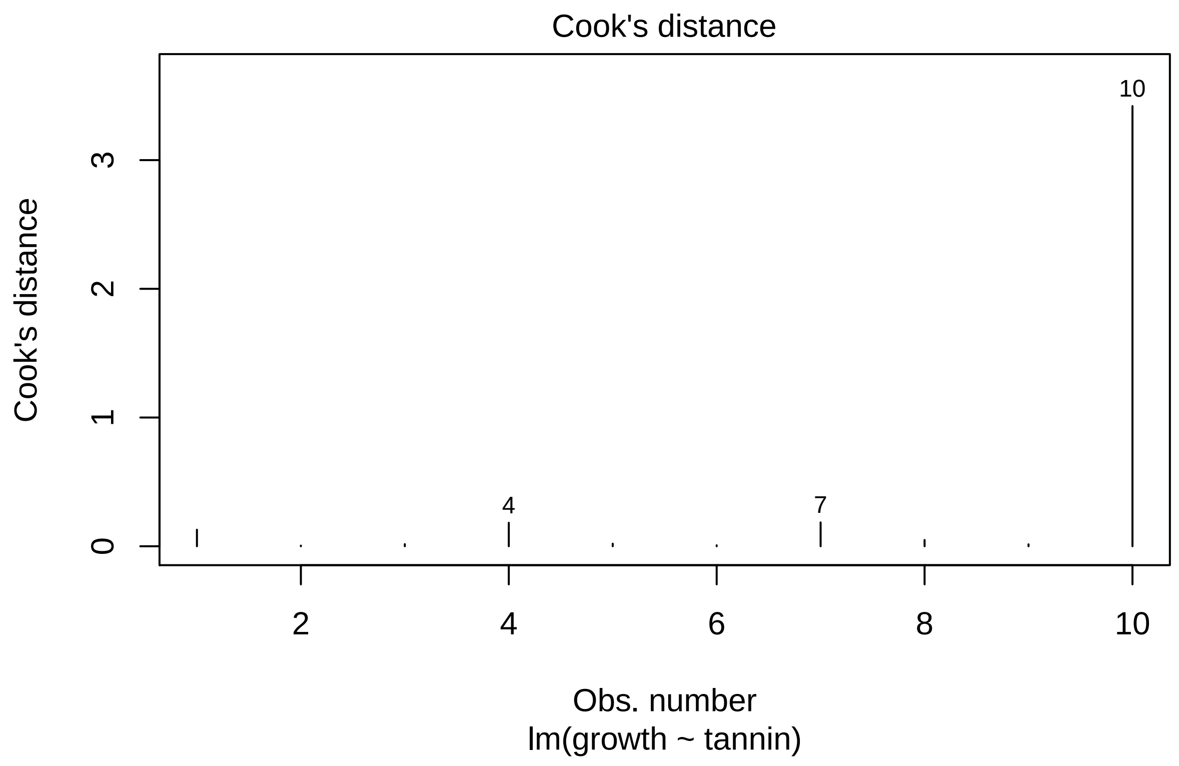
However the point that we added with both a high leverage and a high deviation from the original model shows up clearly when we diagnose the model with this point included.

```r
plot(new.mod,which=5)
```



Another way of spotting this influential points is by looking at Cook's distance directly. Values over 1 are typically considered to be extreme.

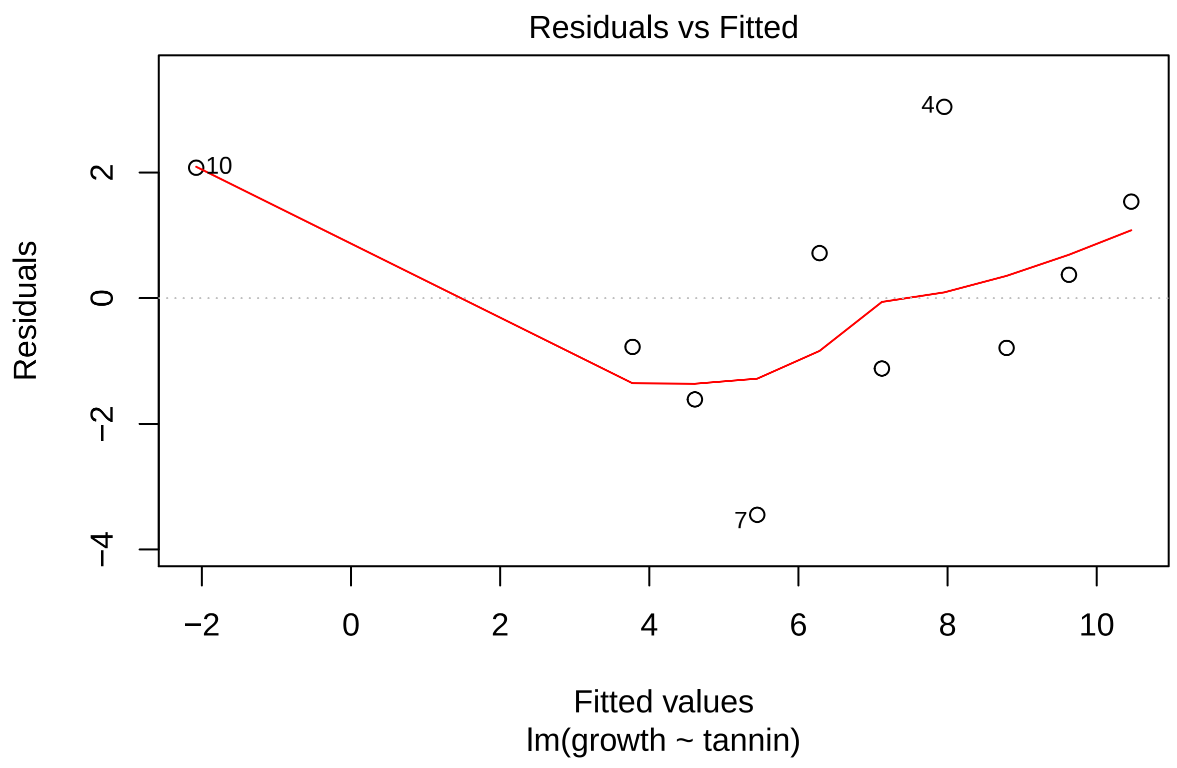
```r
plot(new.mod,which=4)
```



3.4.3 Lack of independence

The value and sign of the residual deviation should not be related in any way to that of the previous point. If residual values form "clusters" on one side of the line or another it is a sign of lack of independence. There are two causes of this. The most serious is intrinsic temporal or spatial autocorrelation. This is discussed in the next section. A less serious matter is that the shape of the model is not appropriate for the data. We can see this in the last example. While the strait line might have been OK for the range of data originally given by Crawley, a straight line is a poor model when an additional data point is added at a high tannin level. At some point tannin stops all growth, so the underlying model must be asymptotic. Thus the model with the extra point has correlation in the raw residuals. High tannin values have positive residual deviation as a result of the poor fit of the model. This can be seen by using the first diagnostic plot that R produces.

```r
plot(new.mod, which=1)
```



A common way of testing for lack of independence is the Durbin Watson test for serial autocorrelation.

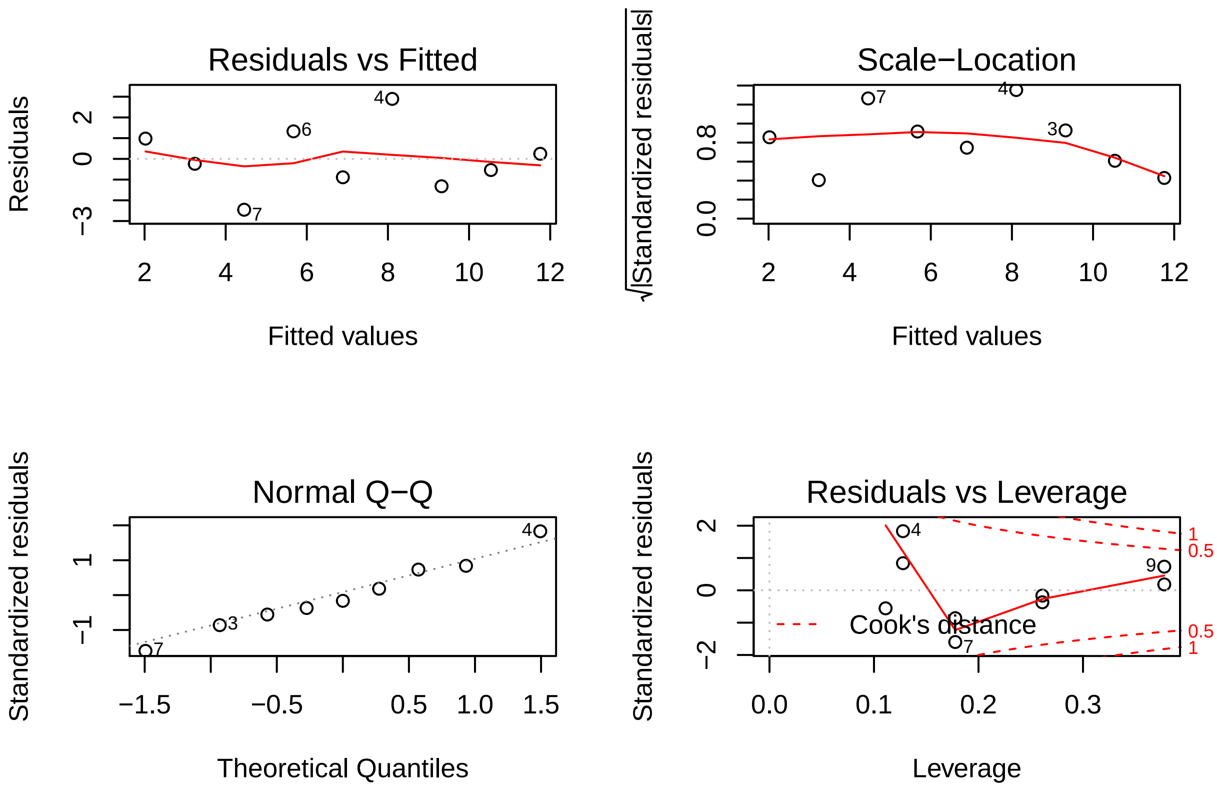
```
library(lmtest)
dwtest(growth~tannin)
```

```
##
##  Durbin-Watson test
##
## data:  growth ~ tannin
## DW = 2.0168, p-value = 0.3679
## alternative hypothesis: true autocorrelation is greater than 0
```

3.4.3.1 Putting it all together

If you just ask R to plot a model you get all four plots. With some practice these should help you spot problems that need attention. Notice that to get four plots on one page you use **par(mfcol=c(2,2))**.

```
par(mfcol=c(2,2))
plot(mod)
```



3.4.4 Violations of assumptions

What do you do if your diagnostic analysis shows that there is a problem with the model? The first thing to realise is that all models are approximations of reality. As the statistician G.E. Box famously stated “All models are wrong ... but some are useful.” If journal editors and referees only admitted papers which used the “correct” model ecological publishing would stop in its tracks. The important point to be aware of is that not all violations of the assumptions are equally serious. Almost all of them can be handled by more advanced modelling, but sometimes it is simply enough to point out that a minor violation was noticed, but it was not considered serious enough to prevent further analysis, together with a clearly stated justification.

Normality of the residuals: This is considered to be the least important assumption of the model. Slight deviations from normality will not usually affect the conclusions drawn. Furthermore some influential authors, eg Sokal and Rolf, state that the central limit theorem means that the assumption can safely be ignored for large (n>30) samples. However, be careful with this, as the assumption is that the residuals are non-normal, but homogeneous. This is unlikely to occcur.

Fixed values of the independent variable: This assumption is in fact nearly always violated in ecological studies. We never measure anything without error. The main point to be aware of is that it is error relative to the range that causes an issue. So, if we measure a variable such as tree height with an accuracy of 10cm and the range of values falls between 5m and 20m there is no problem. However if the range of heights were to be only between 10m and 11m an issue may arise. You should always aim to have a large range of values for the explanatory variable with respect to measurement errors.

Homogeneity of variance: Violations of this assumption can be quite serious. However you should be aware that fixing the problem using techniques such as weighted regression will not affect the slope of the regression line. It will affect the p-values (making them larger) and confidence intervals (making them wider). If the p-value from the uncorrected model is very small (p<0.001) then a correction for heterogeneity of variance is unlikely to result in a loss of significance. If the p-value is close to the conventional cut off (p<0.05) then the correction will almost certainly change your conclusions. In some cases the apparent heterogenity is the result of outliers that need removal. Doing this will of course change the slope.

Incorrect model form: A regression is a straight line. In many ecological situations the true responses take the form of a curve of some kind. Asymptotes are very commonly predicted both from theory and "common sense". For example, adding more nitrogen fertilizer beyond a certain point will not produce any more growth. The biggest problem for regression in these circumstances is not that the model does not fit. It may approximate quite well to one part of the curve. The issue is that the functional form is misleading and does not represent the underlying process.

Influential outliers: We have seen that an outlier does not necessarily influence the regression unless it also has high leverage. You have to think carefully before simply removing these points. If points at the extreme ends of the x axis have high residual values it may be that the model is badly formed. We have seen the potential for this in the example above. In this case you may want to restrict a linear regression to the linear part of the relationship and remove data at the ends. Flexible models such as GAMs which we will see later can deal with the issue. You may also use a data transformation to pull the extreme values in. It may well turn out that the issue simply is due to mistaken measurements, in which case the extreme values are rejected after checking. There are also other methods to formally handle the problem such as robust regression.

Independence of the errors: This is the big issue. Violations of this assumption are always serious. The main problem is that if observations are not independent the model is claiming many more degrees of freedom (replication) than is justified. This means that the model cannot be generalised. The issue affects many, if not all, ecological studies to some extent. Measurements are rarely truly independent from each other in space and time. The issue affects the interpetation of the p-value and confidence intervals of the model. While the model may still be suitable for prediction within its sample space it will not generalise beyond it. In the next class we will look at an example in detail that may help to clarify this.

3.5 Exercises

1. Climate data

When analysing time series it can be very important to take into account serial autocorrelation. However if serial autocorrelation is not particularly strong simple regression analysis can provide reliable insight regarding trends.

The following data are taken from the UK Meterological office. They represent total precipitation and mean monthly temperatures averaged over climate stations in England. Is there any evidence of trends?

```
Temp<-read.csv("https://tinyurl.com/aqm-data/Temp.csv")
Prec<-read.csv("https://tinyurl.com/aqm-data/Prec.csv")
```

3.6 Some "data wrangling"

There are now many different ways in R to reshape data frames into consistent formats. The new "tidyr" and dplyr are becoming increasingly popular. A simple way of stacking many columns into two (variable and value) is provided by the reshape package. The melt function takes the original data frame and arguments defining "id" variables and "measurement" variables.

```
library(reshape2)
Temp2<-melt(Temp[,3:15],id="Year")
str(Temp2)
```

```
## 'data.frame':    1236 obs. of  3 variables:
```

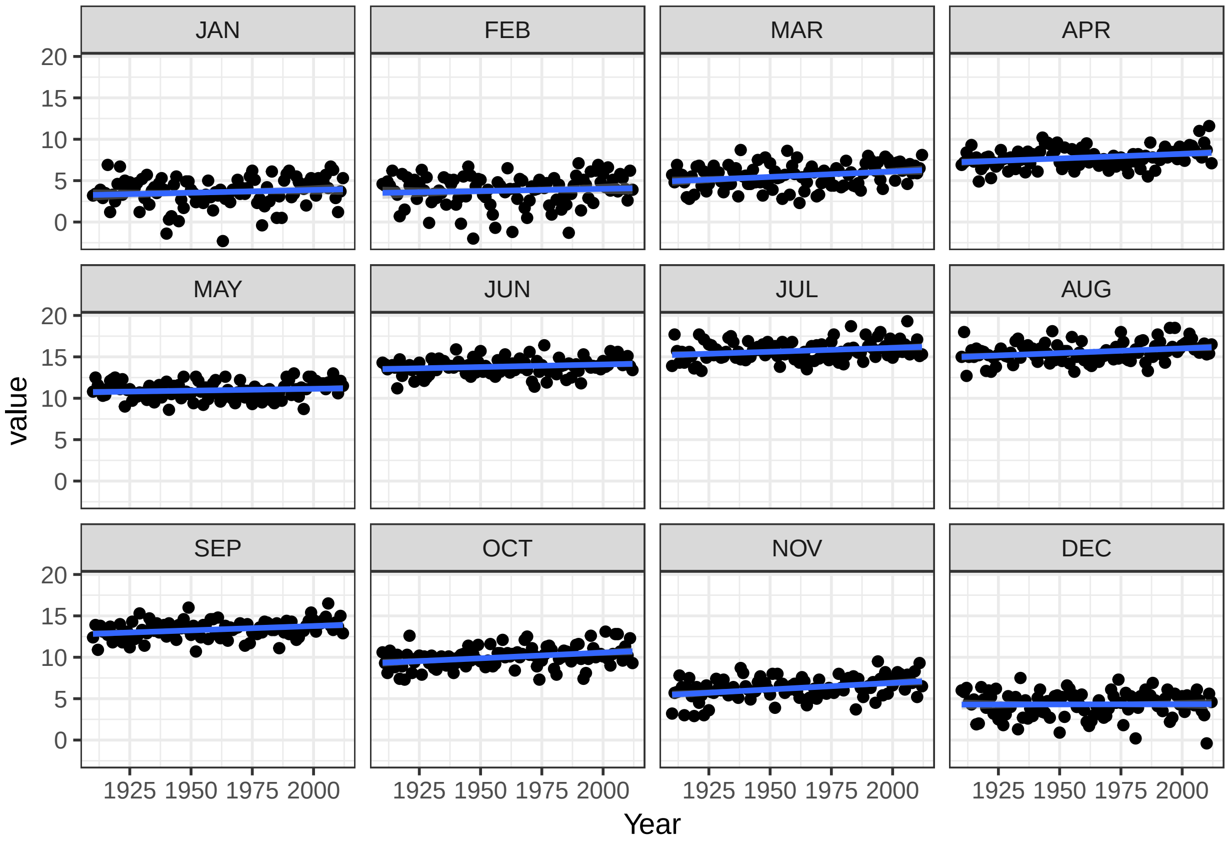
```
##  $ Year    : int  1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 ...
##  $ variable: Factor w/ 12 levels "JAN","FEB","MAR",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ value   : num  3.2 3.4 3.3 3.9 2.9 3.5 6.9 1.2 3.3 2.5 ...
```

```
head(Temp2)
```

```
##   Year variable value
## 1 1910      JAN   3.2
## 2 1911      JAN   3.4
## 3 1912      JAN   3.3
## 4 1913      JAN   3.9
## 5 1914      JAN   2.9
## 6 1915      JAN   3.5
```

All the months can now be plotted on a single figure using ggplot2.

```
g0<-ggplot(Temp2,aes(x=Year,y=value))
g0+geom_point()+geom_smooth(method="lm")+facet_wrap("variable")
```



3.7 One way to run multiple analyses

Once the data are in long format it is fairly easy to loop through the factor levels and run a separate analysis for each month. Note that the value of the factor is used to subset the data. Then within the loop the same code is run multiple times. To include output in a knitted document you need to explicitly tell R to print the results.

```
for(i in levels(Temp2$variable))
{
  d<-subset(Temp2,Temp2$variable==i)
  mod<-lm(data=d,value~Year)
  print(i)
  print(anova(mod))
  print(summary(mod))
}
```

```
## [1] "JAN"
## Analysis of Variance Table
##
## Response: value
##            Df  Sum Sq Mean Sq F value Pr(>F)
## Year        1   4.071  4.0706  1.3876 0.2416
## Residuals 101 296.298  2.9336
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9279 -0.8244  0.1514  1.2828  3.5863
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.497250  11.132331  -0.853    0.396
## Year         0.006686   0.005676   1.178    0.242
##
## Residual standard error: 1.713 on 101 degrees of freedom
## Multiple R-squared:  0.01355,    Adjusted R-squared:  0.003785
## F-statistic: 1.388 on 1 and 101 DF,  p-value: 0.2416
##
## [1] "FEB"
## Analysis of Variance Table
##
## Response: value
##            Df Sum Sq Mean Sq F value Pr(>F)
## Year        1   2.62  2.6165  0.7436 0.3906
## Residuals 101 355.40  3.5188
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7250 -0.9115  0.1946  1.3608  3.1445
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.712280  12.192223  -0.551    0.583
## Year         0.005361   0.006217   0.862    0.391
##
## Residual standard error: 1.876 on 101 degrees of freedom
## Multiple R-squared:  0.007308,   Adjusted R-squared:  -0.00252
## F-statistic: 0.7436 on 1 and 101 DF,  p-value: 0.3906
```

```
##
## [1] "MAR"
## Analysis of Variance Table
##
## Response: value
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## Year        1  15.482  15.482   7.964 0.005747 **
## Residuals 101 196.345   1.944
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3082 -0.9820  0.1701  0.9679  3.4048
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.975905   9.062175  -2.204  0.02977 *
## Year          0.013040   0.004621   2.822  0.00575 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.394 on 101 degrees of freedom
## Multiple R-squared:  0.07309,    Adjusted R-squared:  0.06391
## F-statistic: 7.964 on 1 and 101 DF,  p-value: 0.005747
##
## [1] "APR"
## Analysis of Variance Table
##
## Response: value
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## Year        1  11.266 11.2657   9.023 0.003362 **
## Residuals 101 126.104  1.2486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5791 -0.6621 -0.0448  0.6434  3.2429
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -14.011848   7.262512  -1.929  0.05650 .
## Year          0.011123   0.003703   3.004  0.00336 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.117 on 101 degrees of freedom
## Multiple R-squared:  0.08201,    Adjusted R-squared:  0.07292
## F-statistic: 9.023 on 1 and 101 DF,  p-value: 0.003362
##
## [1] "MAY"
## Analysis of Variance Table
##
## Response: value
##            Df  Sum Sq Mean Sq F value Pr(>F)
## Year        1   1.797  1.7970  1.7804 0.1851
## Residuals 101 101.944  1.0093
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.42442 -0.62451 -0.02886  0.67880  1.89335
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.257157   6.529832   0.346    0.730
## Year        0.004443   0.003329   1.334    0.185
##
## Residual standard error: 1.005 on 101 degrees of freedom
## Multiple R-squared:  0.01732,    Adjusted R-squared:  0.007593
## F-statistic:  1.78 on 1 and 101 DF,  p-value: 0.1851
##
## [1] "JUN"
## Analysis of Variance Table
##
## Response: value
##            Df Sum Sq Mean Sq F value  Pr(>F)
## Year        1  3.466  3.4664  3.7702 0.05496 .
## Residuals 101 92.860  0.9194
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.50768 -0.57619  0.01828  0.61019  2.46764
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.740239   6.232143   0.279    0.781
## Year        0.006170   0.003178   1.942    0.055 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9589 on 101 degrees of freedom
## Multiple R-squared:  0.03599,    Adjusted R-squared:  0.02644
## F-statistic:  3.77 on 1 and 101 DF,  p-value: 0.05496
```

```
##
## [1] "JUL"
## Analysis of Variance Table
##
## Response: value
##            Df  Sum Sq Mean Sq F value  Pr(>F)
## Year        1   8.144  8.1436  6.5637 0.01188 *
## Residuals 101 125.311  1.2407
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25628 -0.83798 -0.07239  0.64977  3.15598
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.827188   7.239644  -0.391   0.6970
## Year         0.009457   0.003691   2.562   0.0119 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.114 on 101 degrees of freedom
## Multiple R-squared:  0.06102,    Adjusted R-squared:  0.05172
## F-statistic: 6.564 on 1 and 101 DF,  p-value: 0.01188
##
## [1] "AUG"
## Analysis of Variance Table
##
## Response: value
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## Year        1  11.613 11.6132  9.2353 0.003023 **
## Residuals 101 127.005  1.2575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.56001 -0.78660 -0.03558  0.62243  2.98701
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.568975   7.288419  -0.901  0.36958
## Year         0.011294   0.003716   3.039  0.00302 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.121 on 101 degrees of freedom
## Multiple R-squared:  0.08378,    Adjusted R-squared:  0.07471
## F-statistic: 9.235 on 1 and 101 DF,  p-value: 0.003023
##
## [1] "SEP"
## Analysis of Variance Table
##
## Response: value
##            Df Sum Sq Mean Sq F value   Pr(>F)
## Year        1   9.92  9.9203  9.6765 0.002426 **
## Residuals 101 103.54  1.0252
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5740 -0.6080  0.1034  0.6420  2.7573
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.100940   6.580889  -1.079  0.28315
## Year         0.010438   0.003355   3.111  0.00243 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.013 on 101 degrees of freedom
## Multiple R-squared:  0.08743,    Adjusted R-squared:  0.0784
## F-statistic: 9.677 on 1 and 101 DF,  p-value: 0.002426
##
## [1] "OCT"
## Analysis of Variance Table
##
## Response: value
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Year        1  16.922 16.9225   11.95 0.0008011 ***
## Residuals 101 143.030  1.4161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.05175 -0.65856  0.02091  0.60267  3.11619
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.704930   7.734564  -2.160 0.033156 *
## Year          0.013633   0.003944   3.457 0.000801 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.19 on 101 degrees of freedom
## Multiple R-squared:  0.1058, Adjusted R-squared:  0.09694
## F-statistic: 11.95 on 1 and 101 DF,  p-value: 0.0008011
##
## [1] "NOV"
## Analysis of Variance Table
##
## Response: value
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Year        1  22.704 22.7043  15.115 0.0001812 ***
## Residuals 101 151.714  1.5021
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.96830 -0.60514  0.06336  0.76071  2.77387
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.676785   7.965901  -3.098 0.002525 **
## Year          0.015791   0.004062   3.888 0.000181 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.226 on 101 degrees of freedom
## Multiple R-squared:  0.1302, Adjusted R-squared:  0.1216
## F-statistic: 15.11 on 1 and 101 DF,  p-value: 0.0001812
##
## [1] "DEC"
## Analysis of Variance Table
##
## Response: value
##            Df Sum Sq Mean Sq F value Pr(>F)
## Year        1   0.02 0.01993  0.0084  0.927
## Residuals 101 238.63 2.36263
##
## Call:
## lm(formula = value ~ Year, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7346 -0.9670  0.3108  0.9989  3.2010
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.3941682  9.9903576   0.340    0.735
## Year        0.0004679  0.0050939   0.092    0.927
##
## Residual standard error: 1.537 on 101 degrees of freedom
## Multiple R-squared:  8.352e-05,  Adjusted R-squared:  -0.009817
## F-statistic: 0.008436 on 1 and 101 DF,  p-value: 0.927
```

This is "quick and dirty" as no diagnostics have been run for each regresson. In order to fully justify any interesting results you would need to do more work than this. In reality a correct analysis of time series data should include an investigation of serial autocorrelation, amongst other elements. We'll be looking at this later in the course.

2.. Recall that in the primer we looked at the relationship between measurements on mussel shell length and body tissue volume in ml. Load the data and analyse them using linear regression.

```
mussels<-Prec<-read.csv("https://tinyurl.com/aqm-data/mussels.csv")
attach(mussels)
```

```
## The following objects are masked from mussels (pos = 61):
##
##     BTVolume, Lshell, Site
```

```
library(ggplot2)
theme_set(theme_bw())
library(multcomp)
```

Chapter 4

Some theory on the general linear model

Mathematically the equation for a one way anova is equivalent to this too. If you think about a situation in which instead of a single value for x being multipled by a single coefficient we have a matrix of values representing which level of a factor is being taken into account when calculating a value of y we have an equation with a parameter for each factor level.

4.1 Calculating the sum of squares

In both cases the formula that is dropped in to the call to fit a linear model in R is similar. The difference is that one way ANOVA uses a factor to predict fitted values and calculate residuals whereas regression uses a numerical variable.

In order to demonstrate the fundamental similarity between the two models we will set up some simulated data and then calculate the sum of squares from first principles.

4.1.0.1 ANOVA

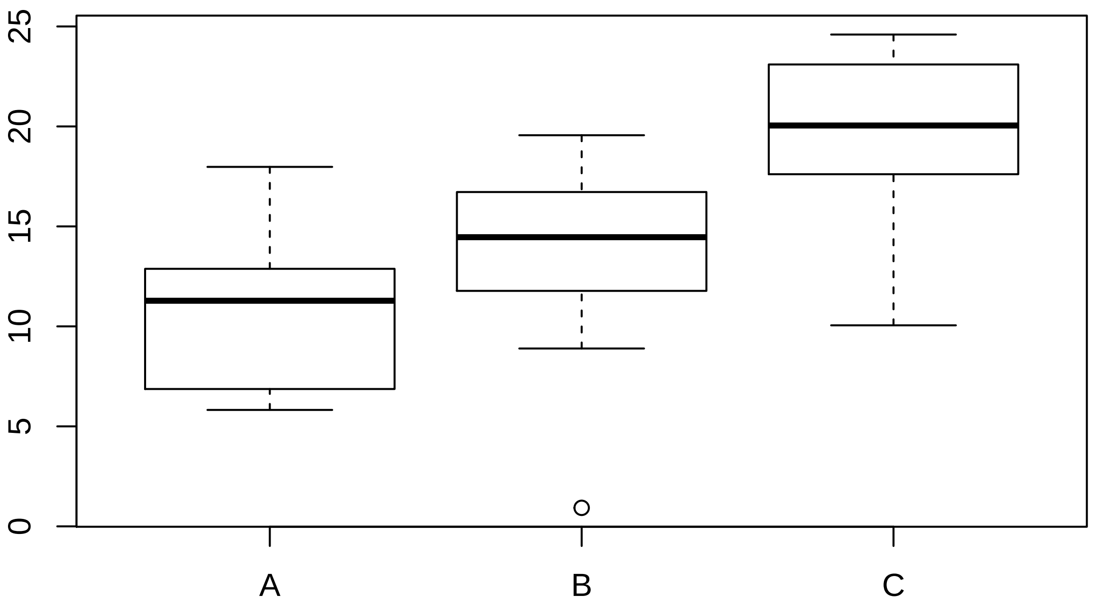
Let's make up a predictor variable. This is a factor with three levels, call them A, B and C. Now, let's assume that the fitted values (in other words the deterministic pattern of response to the three levels of this factor) are mean values of 10, 15 and 20. These are the expected responses if there were no variability apart from that between groups.

```r
set.seed(1)
predictor<-as.factor(rep(c("A","B","C"),each=10))
fitted<-rep(c(10,12,20),each=10)
```

Of course, there is going to be variability within each group. So let's add this in. We will assume that the variability can be modelled as a normal distribution with mean zero and standard deviation of 5.

So the results we actually get are derived by adding these two components together.

```r
residuals<-rnorm(30,mean=0,sd=5)
results<-fitted+residuals
d<-data.frame(predictor,results)
boxplot(results~predictor,data=d)
```



Of course, because this was a simulation experiment the actual means will be slightly different to the invested values. We can set up a data frame with true fitted and residual values like this.

```
mod<-lm(results~predictor,data=d)
d<-data.frame(predictor,results,fitted=fitted(mod),residuals=residuals(mod))
head(d)
```

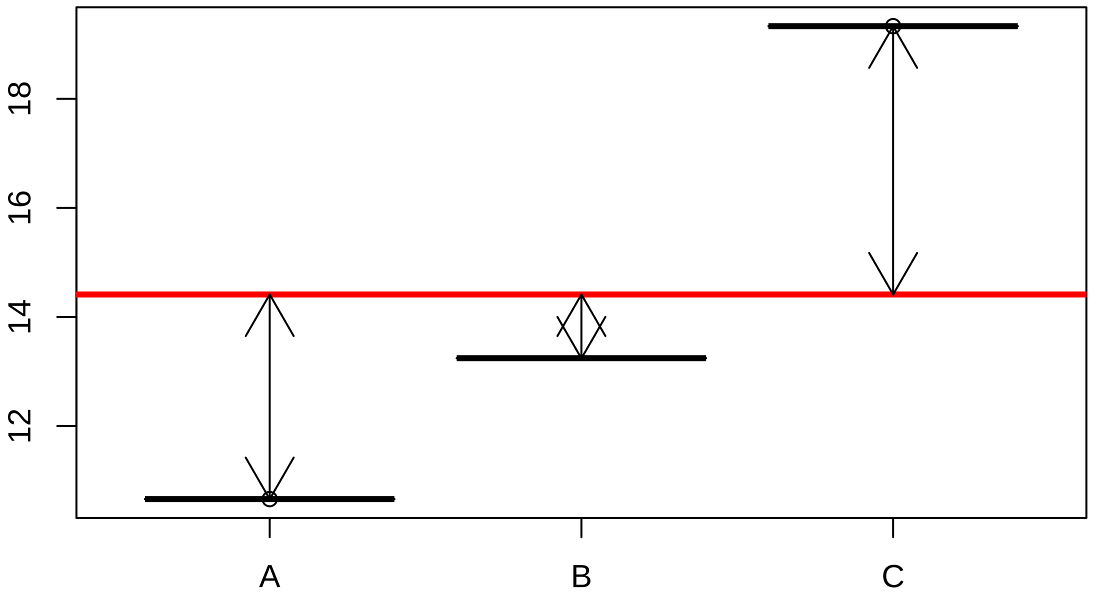
```
##   predictor   results   fitted  residuals
## 1         A  6.867731 10.66101 -3.7932830
## 2         A 10.918217 10.66101  0.2572027
## 3         A  5.821857 10.66101 -4.8391570
## 4         A 17.976404 10.66101  7.3153901
## 5         A 11.647539 10.66101  0.9865250
## 6         A  5.897658 10.66101 -4.7633558
```

4.1.0.2 The numerator sum of squares

OK, so where do the numbers in the Anova table come from? The first step is to understand that the numerator sum of squares represents all the deterministic variability in the system. This is the variability attributable to differences between groups. The sum of squares is the sum of the squared deviations around the mean. But, which mean? In this case it is the overall mean value for the respnse.

So, look at the boxplot again, but this time remove the variability and just plot the means. We can show the difference for each mean from the grand mean using arrows.

```
boxplot(fitted~predictor,data=d)
abline(h=mean(results),lwd=3,col=2)
arrows(1,10.66,1,mean(results),code=3)
arrows(2,13.24,2,mean(results),code=3)
arrows(3,19.33,3,mean(results),code=3)
```



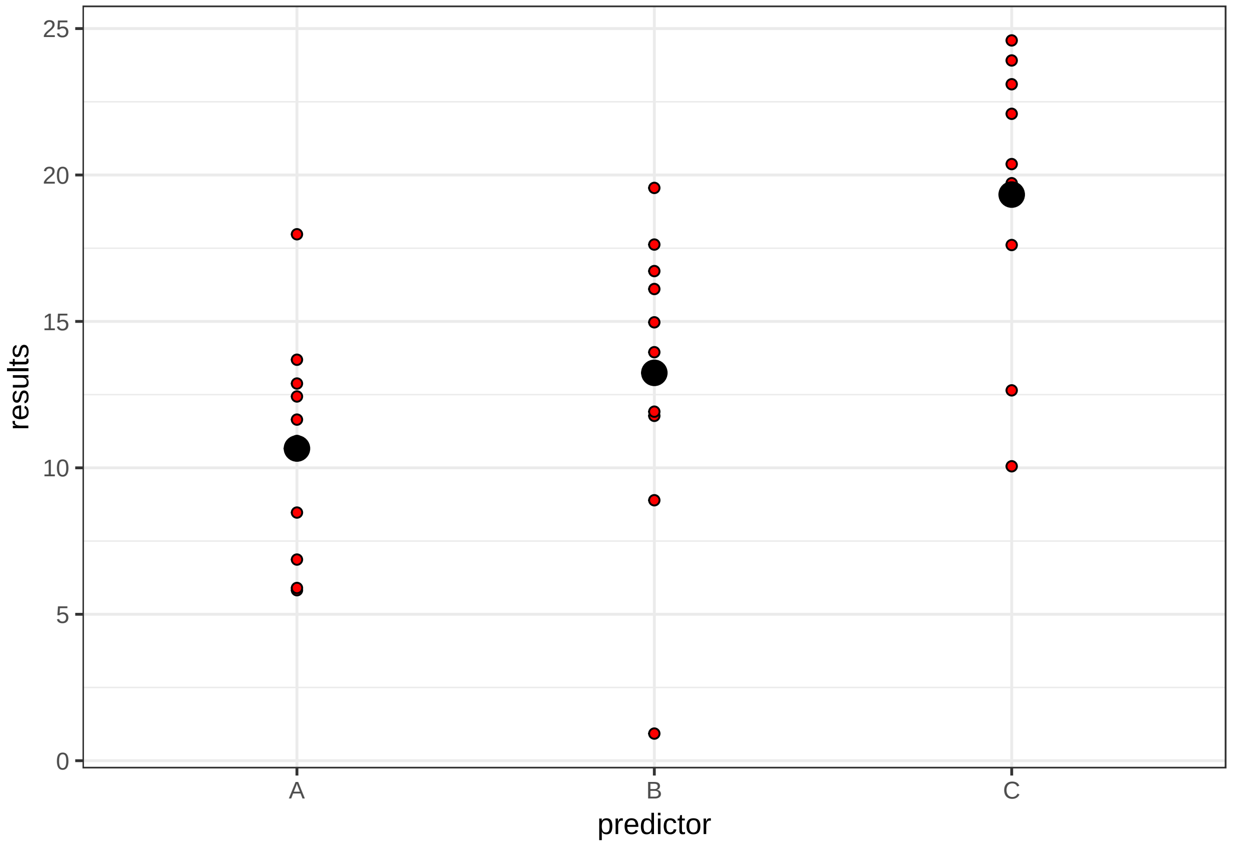
OK, so for each mean there are 10 fitted values. The sum of squares is simply

```
nsqrs<-(d$fitted-mean(results))^2
nsumsqrs<-sum(nsqrs)
```

4.1.0.3 Denominator sum of squares

The denominator sum of squares in an Anova table represents all the variability that can be attributed to the stochastic component of the model. In other words, the residual variablity. We produced that when simulating the data. The values are simply the residuals once the means for each group have been subtracted.

```
g0<-ggplot(d,aes(x=predictor,y=results))
g0+geom_point(pch=21,bg=2)+stat_summary(fun.y=mean,geom="point",size=4)
```



So each residual is the distance between the red points and the large black point representing the mean for that group.

```
dsqrs<-d$residuals^2
dsumsqrs<-sum(dsqrs)
```

```
mod<-lm(results~predictor)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: results
##           Df Sum Sq Mean Sq F value   Pr(>F)
## predictor  2 396.36  198.18  8.9192 0.001062 **
## Residuals 27 599.93   22.22
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
nsumsqrs
```

```
## [1] 396.3639
```

```
dsumsqrs
```

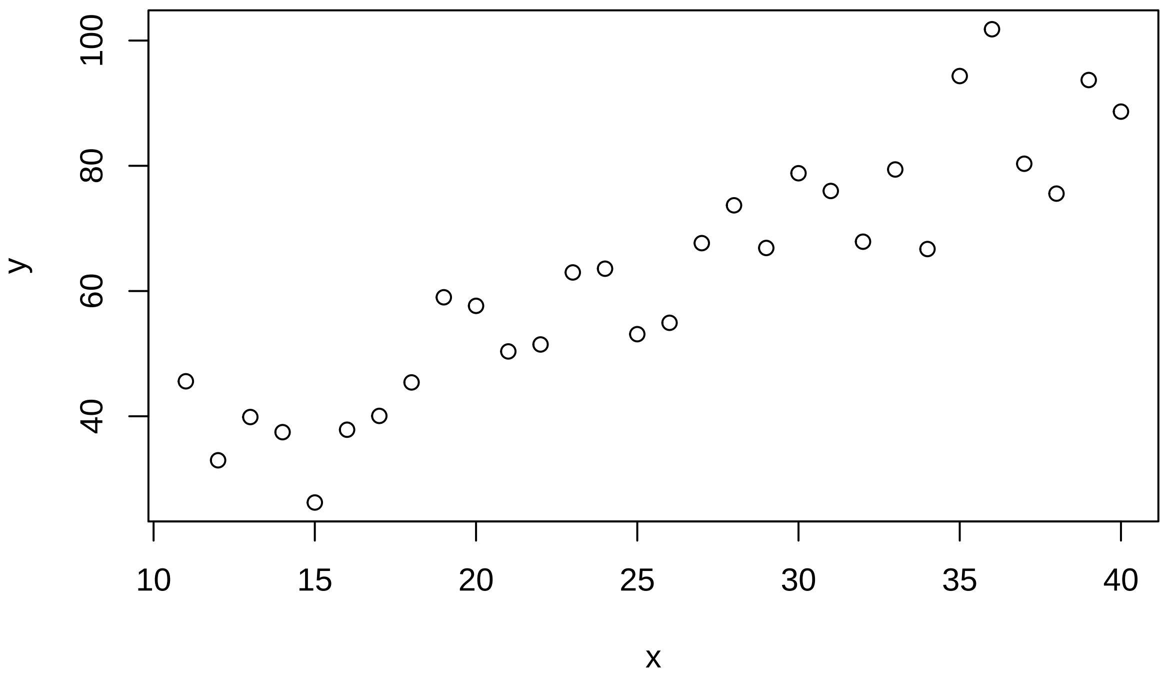
```
## [1] 599.9315
```

The degrees of freedom for the munerator sum of squares are the number of free parameters in the model. This is one less than the number of groups because we need to calculate the grand mean from the data, and thus use up degree of freedom. In order to calculate the residuals we need to calculate three means in this case, so the denominator degree of freedom is the total sample size minus three.

4.1.1 Regression

The set up for regression is, in effect, identical, apart from the fact that the fitted values are continuous rather than group means. So if we invent some data

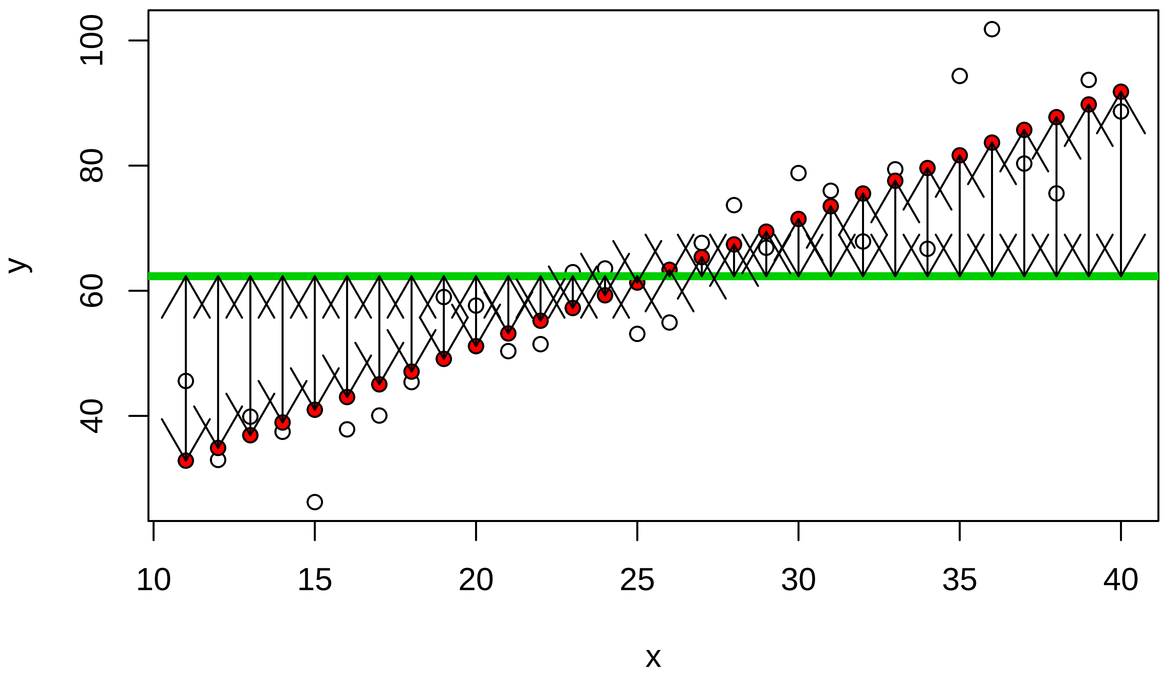
```
x<-11:40
y<-10+x*2+rnorm(30,0,10)
plot(y~x)
```



```
mod<-lm(y~x)
d<-data.frame(x,y,fitted=fitted(mod),residuals=residuals(mod))
head(d)
```

```
##    x        y   fitted  residuals
## 1 11 45.58680 32.84861  12.738183
## 2 12 32.97212 34.88166  -1.909533
## 3 13 39.87672 36.91470   2.962016
## 4 14 37.46195 38.94774  -1.485794
## 5 15 26.22940 40.98079 -14.751383
## 6 16 37.85005 43.01383  -5.163776
```

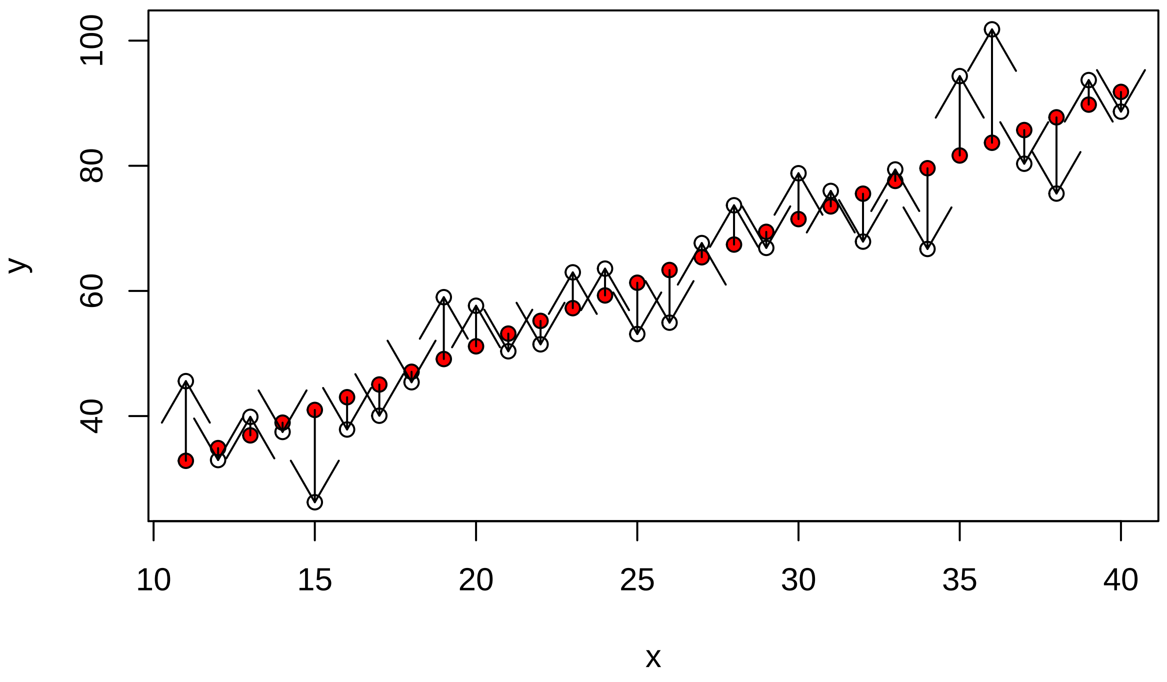
```
plot(y~x)
points(x,fitted(mod),pch=21,bg=2)
abline(h=mean(y),lwd=4,col=3)

arrows(x,fitted(mod),x,mean(y),code=3)
```



```
nsqrs<-(d$fitted-mean(y))^2
nsumsqrs<-sum(nsqrs)
```

4.1.2 Residuals

```
plot(y~x)
points(x,fitted(mod),pch=21,bg=2)
arrows(x,fitted(mod),x,y,code=2)
```



```
dsqrs<-d$residuals^2
dsumsqrs<-sum(dsqrs)
```

```
mod<-lm(y~x)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1 9289.5  9289.5  141.99 1.759e-12 ***
## Residuals 28 1831.9    65.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
nsumsqrs
```

```
## [1] 9289.517
```

```
dsumsqrs
```

```
## [1] 1831.902
```

4.2 Where does R squared (coefficient of determination) come from?

The "explained variability" in the data is often reported using R squared. High values suggest that a large proportion of the variability is "explained" by the model, whether the model is a regression or an ANOVA. Where does that fit in? Well the total sum of squares is the sum of all the squared distances that we have calculated.

If we divide the sum of squares attributable to the model by the total we get the proportion of the variability attributable to the model.

```
totsumsqrs<-nsumsqrs+dsumsqrs
nsumsqrs/totsumsqrs
```

```
## [1] 0.8352817
```

We can see that this is the same as R provides by asking for a summary.

```
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.751  -5.120  -1.579   5.365  18.129
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.4851     4.5945   2.282   0.0303 *
## x             2.0330     0.1706  11.916 1.76e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.089 on 28 degrees of freedom
## Multiple R-squared:  0.8353, Adjusted R-squared:  0.8294
## F-statistic:   142 on 1 and 28 DF,  p-value: 1.759e-12
```

4.3 Model assumptions

Assumptions of any linear model are the same.

- Normally distributed errors
- Identically distributed errors over the range (homogeneity of variance)
- Independent errors

In the case of regression we can also state that it is assumed that there is

- No undue influence of points with high leverage (regression)
- An underlying linear relationship (regression)

If these assumptions are not met then the p-values and R squared calculations based on the sums of squares may potentially be misleading. So testing assumptions critically is an important part of using and interpreting linear models. The best diagnostic tests involve looking carefully at the data rather than running an automated decision tool. Minor violations may be acceptable in some circumstances.

4.4 Some practice using linear models

The best way to become comfortable fitting models in R is to fit them several times until the procedure becomes routine.

Let's look at some classic data on the morphological characteristics of iris flowers. This is a useful data set to get to know, as it is very frequently used in the R literature to illustrate a wide range on more advanced techniques, including machine learning. We can load the data set into memory with the data command. For consistency I'll then call it d.

```
data("iris")
d<-iris
DT:::datatable(d)
```

Show 10 entries Search:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

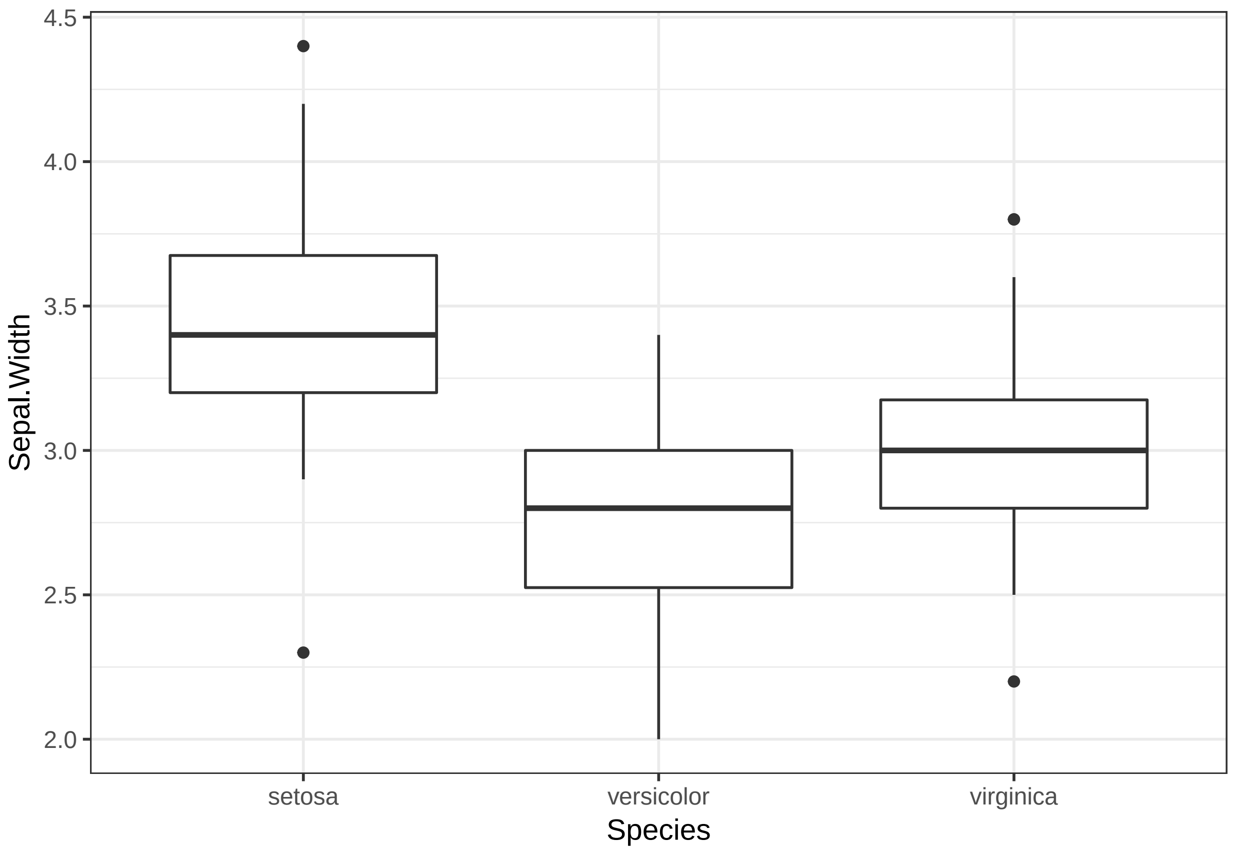
Showing 1 to 10 of 150 entries Previous 1 2 3 4 5 … 15 Next

So there are three (closely related) species with measurements made on petals and sepals.

4.5 One way Anova

So, a reminder. The first step in any analysis is to look at the data. So if we want to look at differences in sepal width between species we can plot the data as boxplots.

```r
g0<-ggplot(d,aes(x=Species,y=Sepal.Width))
g0 +geom_boxplot()
```



4.5.1 Diagnostics

The boxplots allow basic diagnostic tests for one way anova. Using them we can test for

- Normally distributed errors
- Identically distributed errors over the range (homogeneity of variance)

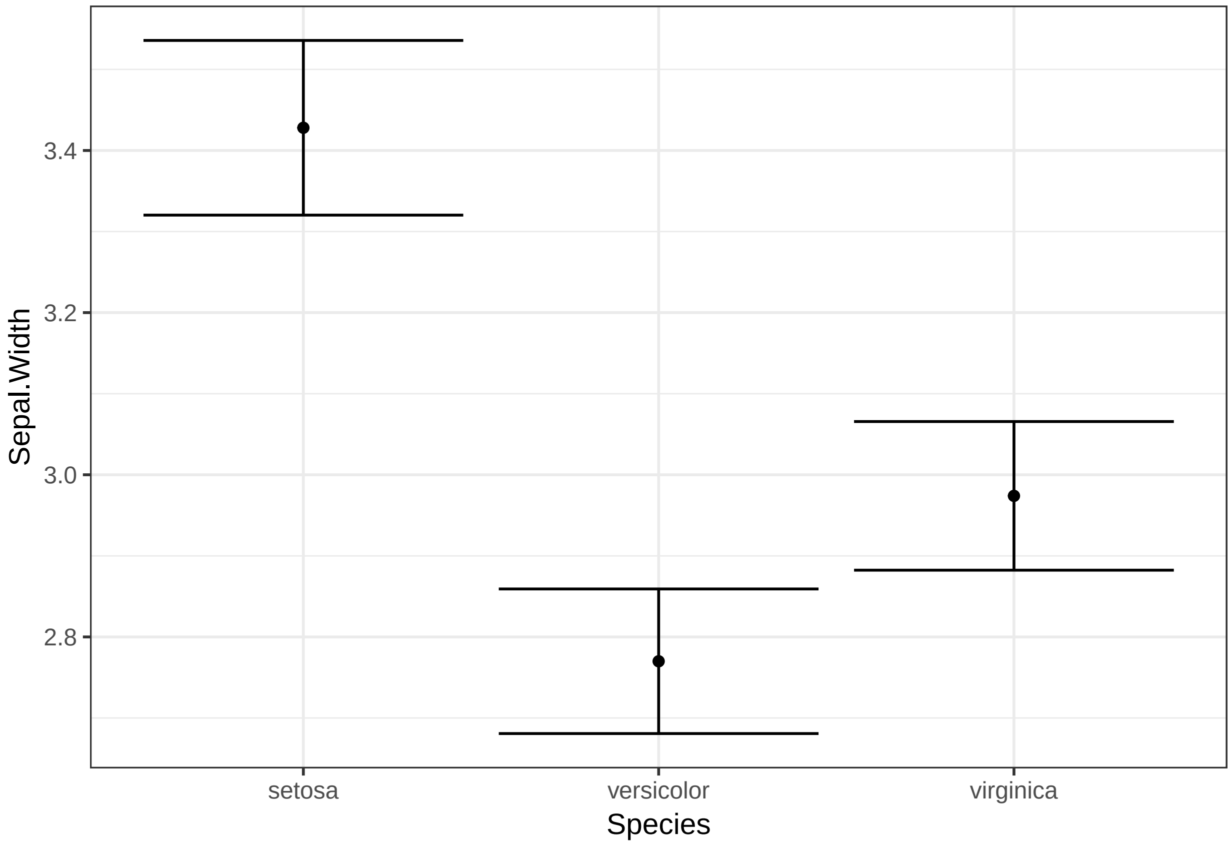
Look carefully at the boxplots. Do they look approximately symetrical? Is there a clear pattern of outliers either above or below the box? If the answers are yes and no, then the data may be approximately normally distributed.

Look at the width of the boxes for each group. Are they similar? If the answer is yes then heterogenity of variance is not likely to be a major problem.

4.6 Statistical inference

The next step is to produce confidence intervals.

```
g0 +stat_summary(fun.y=mean,geom="point") + stat_summary(fun.data=mean_cl_normal,geom="errorbar")
```



Remember that confidence intervals are a function of sample size. So if there are differences in the number of measurements in each group this may result in some showing narrower confidence intervals even if the underlying variances are similar.

4.7 Fitting a linear model

There are two commonly forms of syntax for fitting a one way anova in R. We can use either oav, or lm. The aov syntax is actually just a wrapper to lm. Lm stands for linear model. The syntax is identical to that used for regression, but the second variable in the formula is a factor.

```
mod<-lm(data=d,Sepal.Width~Species)
anova(mod)
```

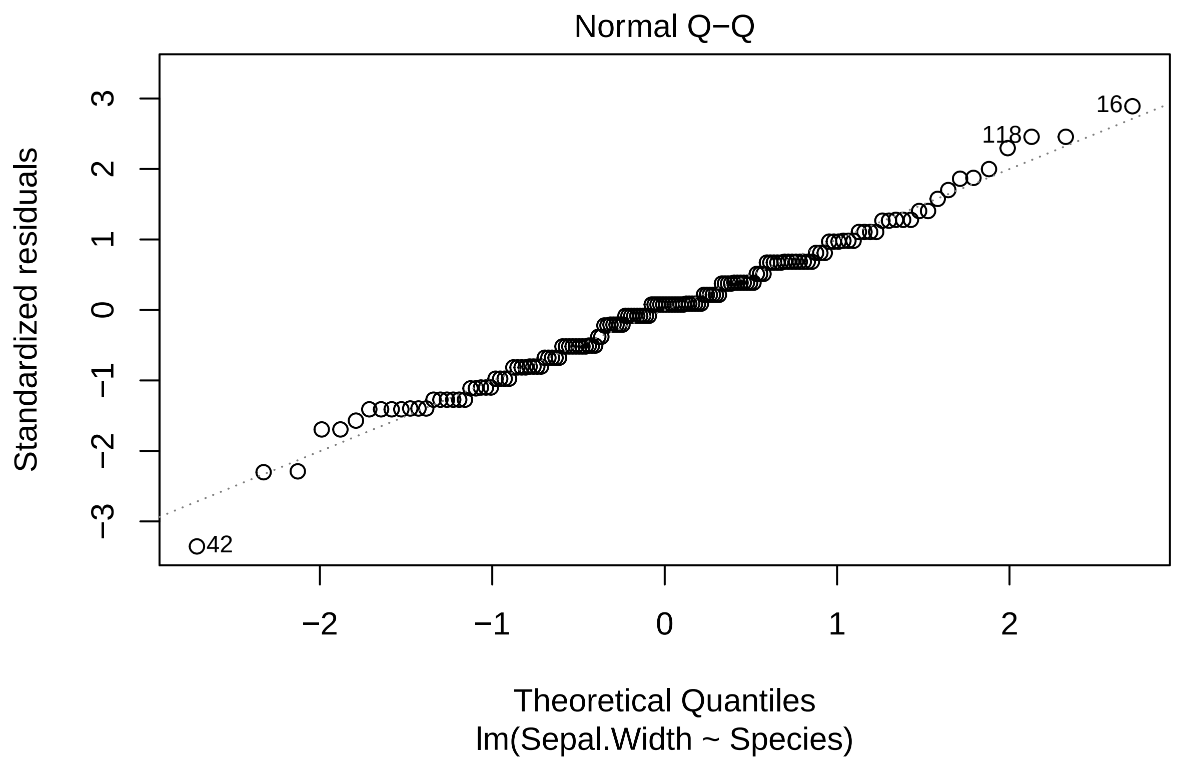
```
## Analysis of Variance Table
##
## Response: Sepal.Width
##            Df Sum Sq Mean Sq F value    Pr(>F)
## Species     2 11.345  5.6725   49.16 < 2.2e-16 ***
## Residuals 147 16.962  0.1154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

You should be able to interpret all the elements in the output now.

4.8 Diagnostics

A qqplot can be the most useful of the diagnistic plots for anova. This will help in deciding if the residuals are normally distributed. Hetereogeneity of variance is best spotted using the boxplots.

```
plot(mod,which=2)
```



If the points more or less fall along the diagonal this can be taken as a good indication that the assumption of normality is met. Minor deviations from normality are rarely important, and taken overall the assumption of "exact" normality of residuals is not the most important. Minor deviations will not invalidate the model.

You should also be very careful before talking about an "invalid" model if there are minor violations. The underlying pattern may be quite clearly shown through almost any reasonable analtsis. Finding the "best" model involves refining the way in which p-values and confidence intervals are calculated to ensure that they are justifiable. If p-values are extremely small, then refinement may be unlikely to change the overall significance of the result. If p-values are close to the traditional cut off point (0.05) then the significance of the result may be questionable, and changes in the way a model is fitted to the data may change the conclusions.

4.9 Treatment contrasts using summary

If we ask for a summary of a one way anova the default in R is to provide a table with treatment contrasts. Treatment constrasts take the first level of the factor as the reference level.

```
summary(mod)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Species, data = d)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.128 -0.228  0.026  0.226  0.972
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.42800    0.04804  71.359  < 2e-16 ***
## Speciesversicolor -0.65800    0.06794  -9.685  < 2e-16 ***
## Speciesvirginica  -0.45400    0.06794  -6.683 4.54e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3397 on 147 degrees of freedom
## Multiple R-squared:  0.4008, Adjusted R-squared:  0.3926
## F-statistic: 49.16 on 2 and 147 DF,  p-value: < 2.2e-16
```

4.10 Changing the reference leval

If the anova represented an experiment with some control group, this would be the natural choice as the reference level. We can set alternative reference levels for the summary table.

```
contrasts(d$Species)<-contr.treatment(levels(d$Species),base=3)
mod<-lm(data=d,Sepal.Width~Species)
summary(mod)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Species, data = d)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.128 -0.228  0.026  0.226  0.972
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.97400    0.04804  61.908  < 2e-16 ***
## Speciessetosa      0.45400    0.06794   6.683 4.54e-10 ***
## Speciesversicolor -0.20400    0.06794  -3.003  0.00315 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3397 on 147 degrees of freedom
## Multiple R-squared:  0.4008, Adjusted R-squared:  0.3926
## F-statistic: 49.16 on 2 and 147 DF,  p-value: < 2.2e-16
```

4.11 Multiple comparisons

In an observational study we will often want to look at the differences between pairs of groups. The "General Linear Hypothesis" from the multcomp package is useful for this.

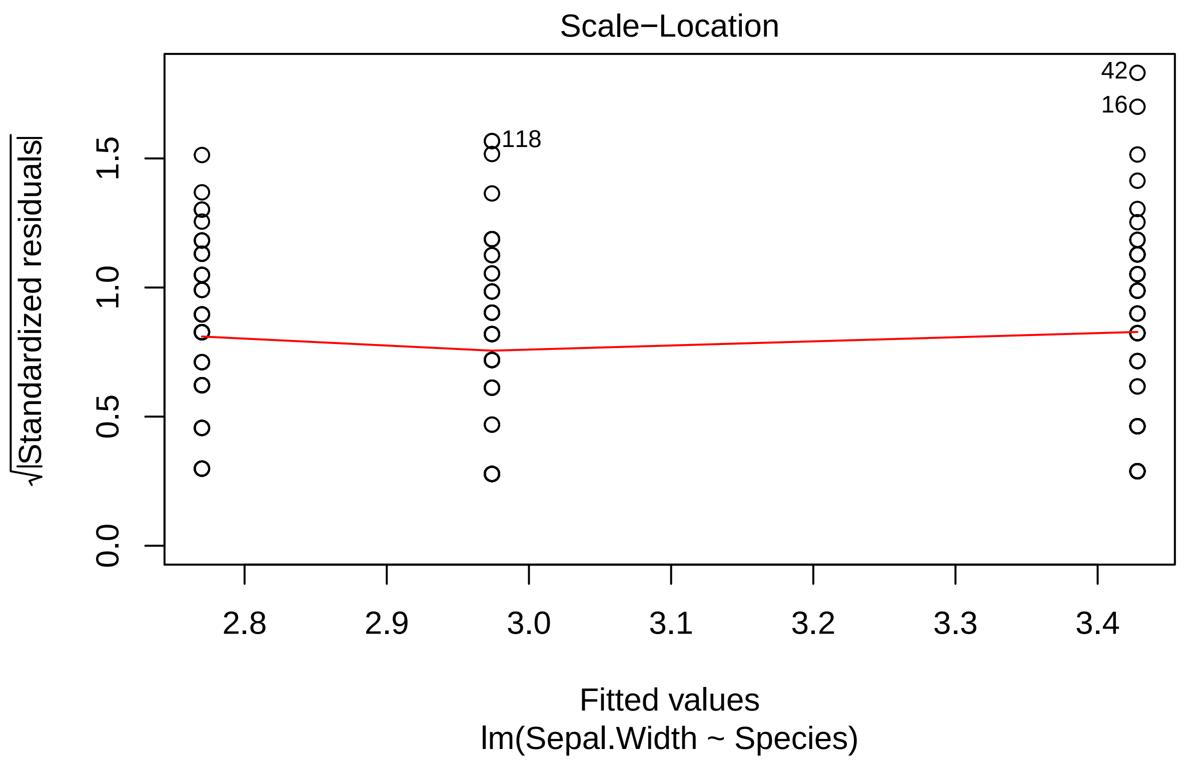
```
library(multcomp)
#plot(glht(mod, linfct = mcp(Species = "Tukey"))) ## Have to adjust the margins to show this clearly
summary(glht(mod, linfct = mcp(Species= "Tukey")))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = Sepal.Width ~ Species, data = d)
##
## Linear Hypotheses:
##                             Estimate Std. Error t value Pr(>|t|)
## versicolor - setosa == 0    -0.65800    0.06794  -9.685  < 1e-04 ***
## virginica - setosa == 0     -0.45400    0.06794  -6.683  < 1e-04 ***
## virginica - versicolor == 0  0.20400    0.06794   3.003  0.00877 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

4.12 Scale location plot

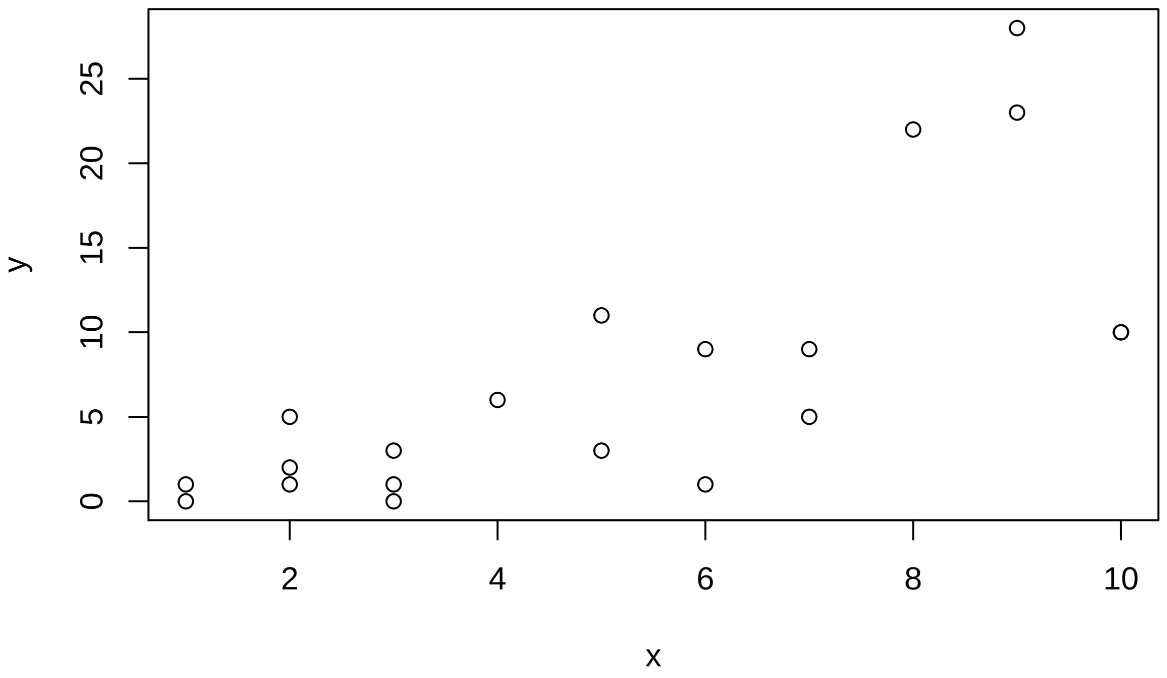
This can be useful for spotting heterogeneity of variance. If heterogeneity of variance is an issue you are likely to see a trend towards larger values on the y axis as the fitted value increases.

```
plot(mod,which=3)
```

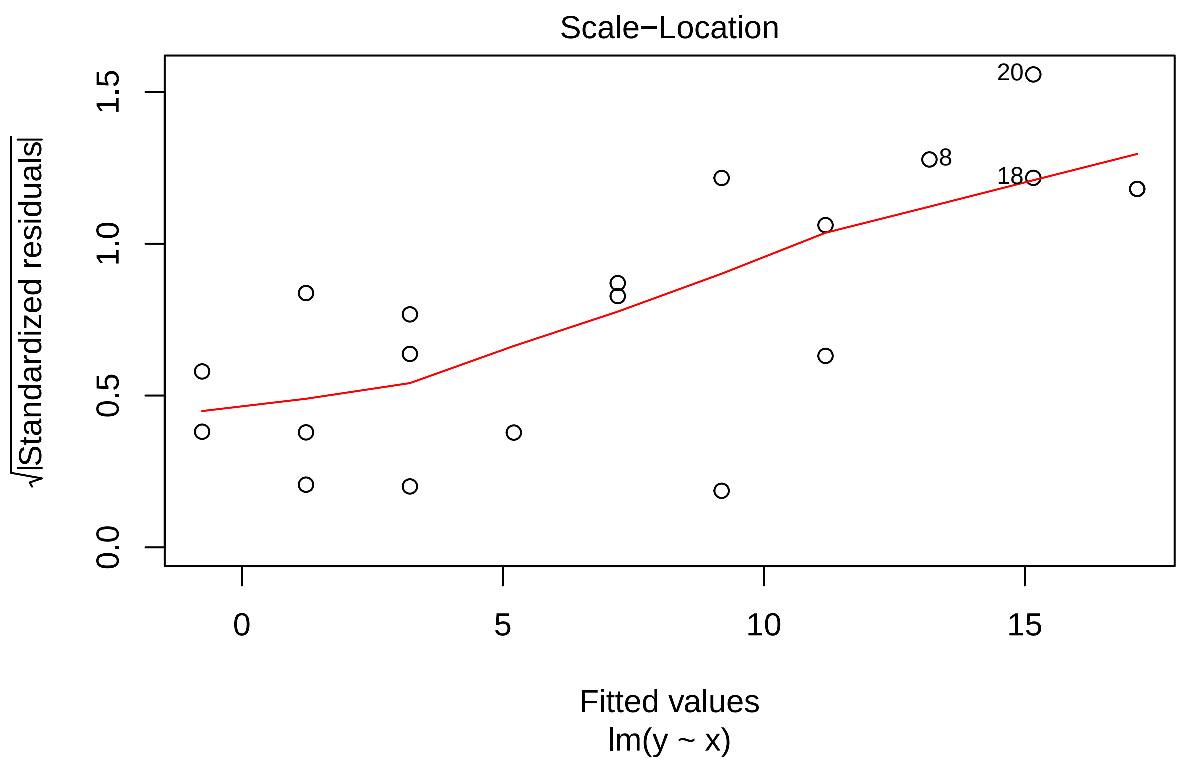


4.13 Example of heterogeniety

```
set.seed(5)
library(MASS)
x<-sample(0:10,20,rep=T)
y<-rnegbin(20,mu=x,theta=2)
plot(y~x)
```



```
mod.negbin<-lm(y~x)
plot(mod.negbin,which=3)
```



4.14 Exercises

1. Try fitting regressions to different variables within the iris data set.
2. The statistician Francis Anscombe invented four data sets in order to demonstrate some of the pitfalls involved in running regression analysis blind (without either looking at the data or carrying out diagnostics).The four data sets are provided in R for illustration.

```
data(anscombe)
str(anscombe)
```

```
## 'data.frame':    11 obs. of  8 variables:
##  $ x1: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x2: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x3: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x4: num  8 8 8 8 8 8 8 19 8 8 ...
##  $ y1: num  8.04 6.95 7.58 8.81 8.33 ...
##  $ y2: num  9.14 8.14 8.74 8.77 9.26 8.1 6.13 3.1 9.13 7.26 ...
##  $ y3: num  7.46 6.77 12.74 7.11 7.81 ...
##  $ y4: num  6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.5 5.56 7.91 ...
```

```
d<-anscombe
```

We can fit models to x1,y1, x2,y2 etc

```
mod1<-lm(data=d,y1~x1)
mod2<-lm(data=d,y2~x2)
mod3<-lm(data=d,y3~x3)
mod4<-lm(data=d,y4~x4)
```

The summaries of the models look very similar.

```
summary(mod1)
```

```
##
## Call:
## lm(formula = y1 ~ x1, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92127 -0.45577 -0.04136  0.70941  1.83882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001     1.1247   2.667  0.02573 *
## x1            0.5001     0.1179   4.241  0.00217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6665, Adjusted R-squared:  0.6295
## F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = y2 ~ x2, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9009 -0.7609  0.1291  0.9491  1.2691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125   2.667  0.02576 *
## x2             0.500      0.118   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6662, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
```

```
summary(mod3)
```

```
##
## Call:
## lm(formula = y3 ~ x3, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1586 -0.6146 -0.2303  0.1540  3.2411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0025     1.1245   2.670  0.02562 *
## x3            0.4997     0.1179   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
```

```
summary(mod4)
```

```
##
## Call:
## lm(formula = y4 ~ x4, data = d)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.751 -0.831  0.000  0.809  1.839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0017     1.1239   2.671  0.02559 *
## x4            0.4999     0.1178   4.243  0.00216 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6667, Adjusted R-squared:  0.6297
## F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

However only one of these data sets is really suitable for linear regression. Run the appropriate diagnostics to find out which.

3. Geek of the week data

For fun I have downloaded spotify detials on tracks from three bands.

```
library(spotifyr)
id <- "df4b4ced508a4ac39ea5357c3ed2d477"
secret<-"48e05067ee3e47a4b9fcedea97ffa5ae"

Sys.setenv(SPOTIFY_CLIENT_ID = id)
Sys.setenv(SPOTIFY_CLIENT_SECRET = secret)

library(spotifyr)
d1 <- data.frame(artist="Oasis",get_artist_audio_features('Oasis'))
d2 <- data.frame(artist="Blur",get_artist_audio_features('blur'))
d3 <- data.frame(artist="Arctic Monkeys",get_artist_audio_features('Arctic Monkeys'))
d<-rbind(d1,d2,d3)
d$track_name<-gsub("Remastered","",d$track_name)
d$album_name<-gsub("(Remastered)","",d$album_name)
library(dplyr)
d<-d[,c(1,3,8,10:21,24)]
write.csv(d,"/home/aqm/data/spotify.csv")
```

These data are rather like the Iris data in general format.

```r
d<-read.csv("/home/aqm/data/spotify.csv")
str(d)
```

```
## 'data.frame':    381 obs. of  17 variables:
##  $ X               : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ artist          : Factor w/ 3 levels "Arctic Monkeys",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ album_name      : Factor w/ 27 levels "13","All The People... Blur Live At Hyde Park 02/07/2009",
##  $ track_name      : Factor w/ 376 levels "(Get Off Your) High Horse Lady",..: 82 204 190 279 154 31
##  $ danceability    : num  0.312 0.367 0.327 0.214 0.355 0.389 0.399 0.426 0.486 0.366 ...
##  $ energy          : num  0.886 0.988 0.895 0.828 0.978 0.905 0.729 0.838 0.913 0.831 ...
##  $ key             : Factor w/ 12 levels "A","A#","B","C",..: 3 9 4 11 6 11 11 4 1 11 ...
##  $ loudness        : num  -3.24 -1.07 -3.06 -2.66 -2.33 ...
##  $ mode            : Factor w/ 2 levels "major","minor": 2 1 1 1 1 1 1 1 1 1 ...
##  $ speechiness     : num  0.0427 0.0706 0.0476 0.0383 0.0596 0.0352 0.0333 0.0362 0.0394 0.0429 ...
##  $ acousticness    : num  7.32e-05 4.25e-03 1.34e-01 1.42e-02 6.57e-04 1.27e-01 1.26e-02 5.32e-02 1.
##  $ instrumentalness: num  1.61e-04 7.58e-06 2.06e-03 1.61e-06 4.95e-03 0.00 3.24e-04 3.02e-06 4.98e-
##  $ liveness        : num  0.0885 0.0486 0.385 0.161 0.448 0.382 0.368 0.0829 0.446 0.484 ...
##  $ valence         : num  0.251 0.327 0.295 0.388 0.19 0.315 0.181 0.182 0.308 0.339 ...
##  $ tempo           : num  80 135 148 170 137 ...
##  $ duration_ms     : int  462227 302600 439373 356600 262427 347400 412293 288600 313200 560000 ...
##  $ track_popularity: int  0 0 0 0 0 0 0 0 0 0 ...
```

Which artist's songs have the most energy? Is it sensible to test this statistically?

Chapter 5

Fitting curves to data

In the previous class we have assumed that the underlying relationship between variables takes the form of a straight line. However in ecology the relationship between variables can have more complex forms. Sometimes the form can be predicted from theory. However we are often simply interested in finding a model that describes the data well and provides insight into processes.

5.1 Data exploration

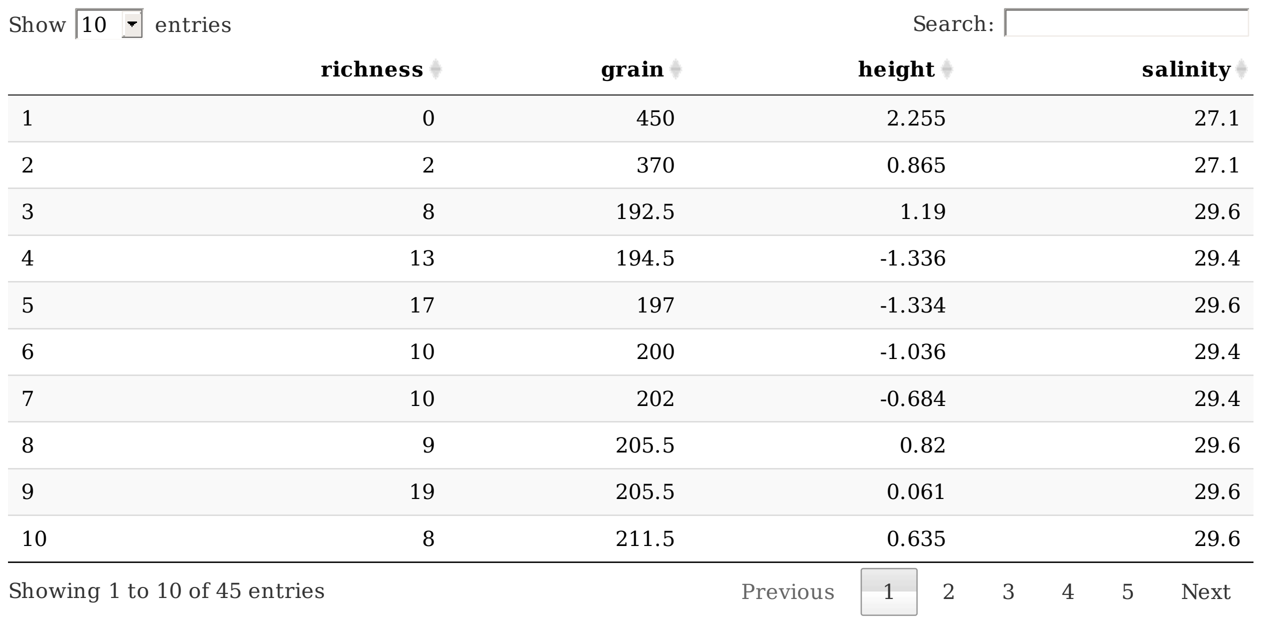
The data we will first look at is analysed by Zuur et al (2007) but here it has been modified slightly in order to illustrate the point more clearly.

The data frame we will look at contains only two variables selected from a large number of measurements taken on sediment cores from Dutch beaches. The response variable is the richness of benthic invertebrates. The explanatory variable we are going to look at here is sediment grain size in mm. The data also contains measurements on height from mean sea level and salinity.

5.1.1 Visualisation

```r
library(ggplot2)
```

```r
d<-read.csv("/home/aqm/course/data/marineinverts.csv")
DT::datatable(d)
```



Show 10 entries Search:

	richness	grain	height	salinity
1	0	450	2.255	27.1
2	2	370	0.865	27.1
3	8	192.5	1.19	29.6
4	13	194.5	-1.336	29.4
5	17	197	-1.334	29.6
6	10	200	-1.036	29.4
7	10	202	-0.684	29.4
8	9	205.5	0.82	29.6
9	19	205.5	0.061	29.6
10	8	211.5	0.635	29.6

Showing 1 to 10 of 45 entries Previous 1 2 3 4 5 Next

The first step in any analysis should be to look at the data.

Base graphics are good enough for a quick figure at this stage.

```
attach(d)
```

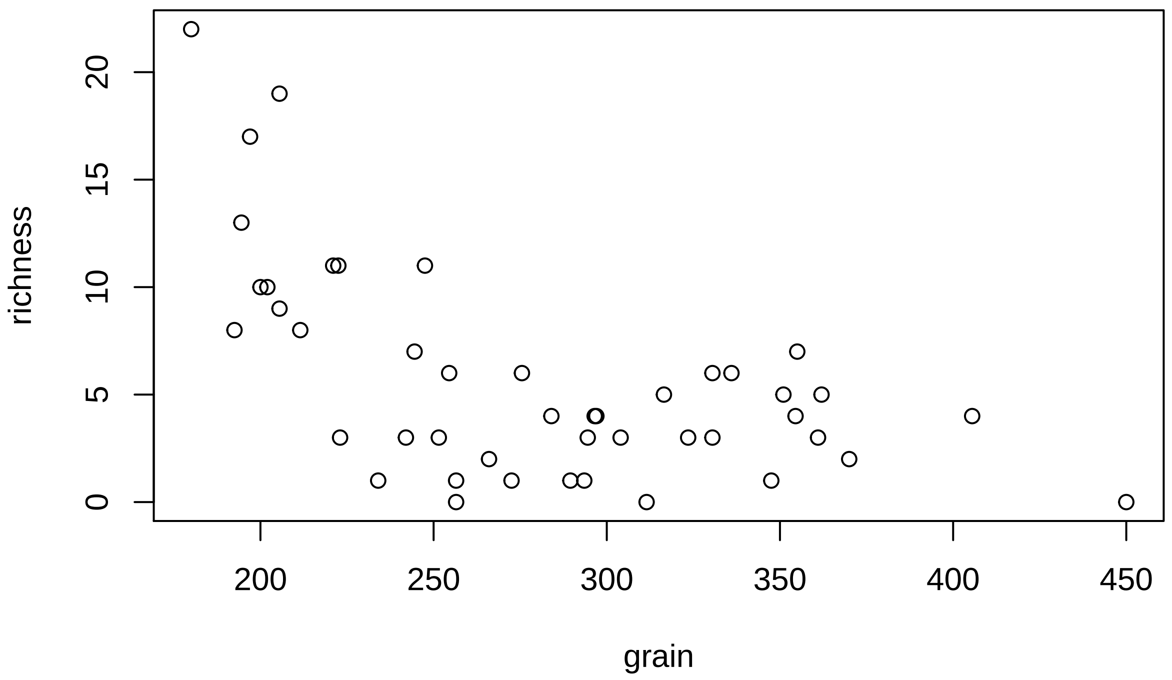
```
## The following object is masked from hsurvey (pos = 5):
##
##     height
```

```
## The following objects are masked from d (pos = 41):
##
##     grain, height, richness, salinity
```

```
## The following objects are masked from d (pos = 57):
##
##     grain, height, richness, salinity
```

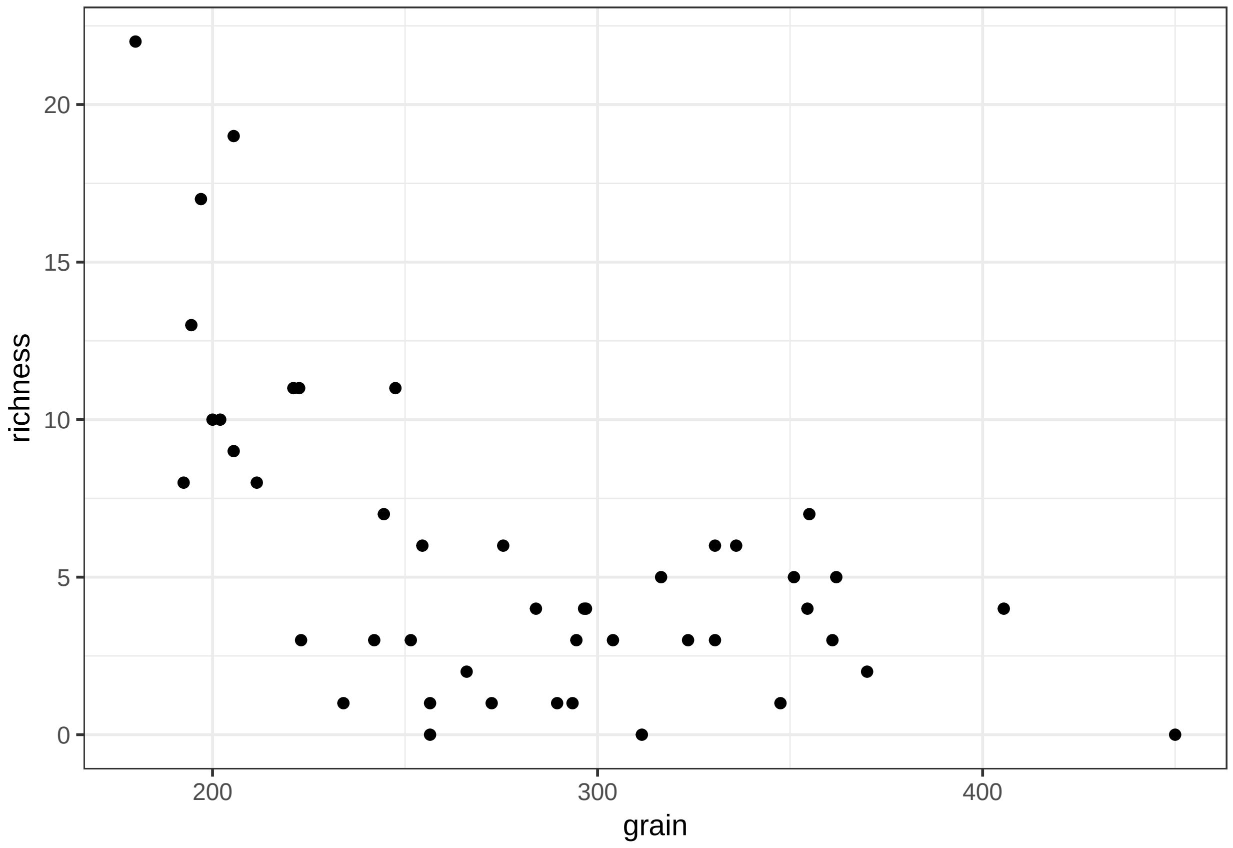
```
## The following object is masked from hsurvey (pos = 72):
##
##     height
```

```
plot(richness~grain)
```



Or in ggplot

```
theme_set(theme_bw())
g0<-ggplot(data=d,aes(x=grain,y=richness))
g1<-g0+geom_point()
g1
```

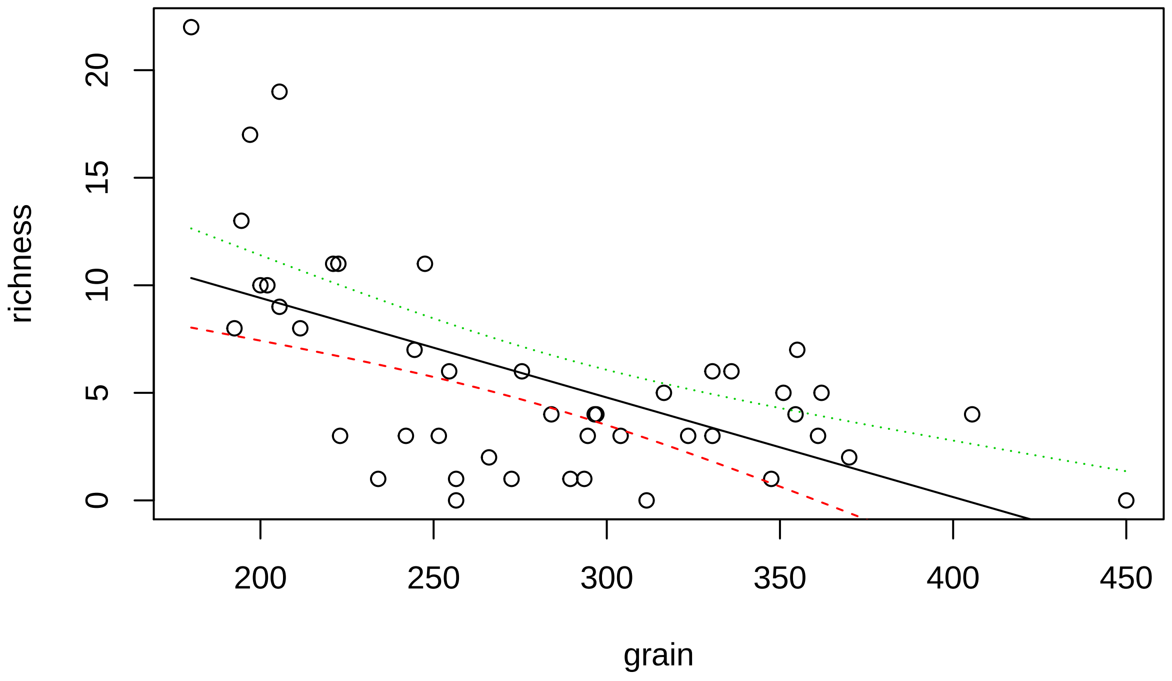


There seems to be a pattern of an intial steep decline in richness as grain size increases, followed by a plateau. Let's try fitting a linear regression to the data and plotting the results with confidence intervals.

Note This is **not** necessarily the correct analysis for these data. In fact there are many reasons why it is clearly an incorrect way of modelling the relationship. We will explore this as the class progresses, and in a later class we will revisit the data set in order to use a much better model. The exercise at this point is for illustrative, didactic purposes. You should think about the reasons for not using a simple model at every step, in order to understand why more advanced methods are needed.

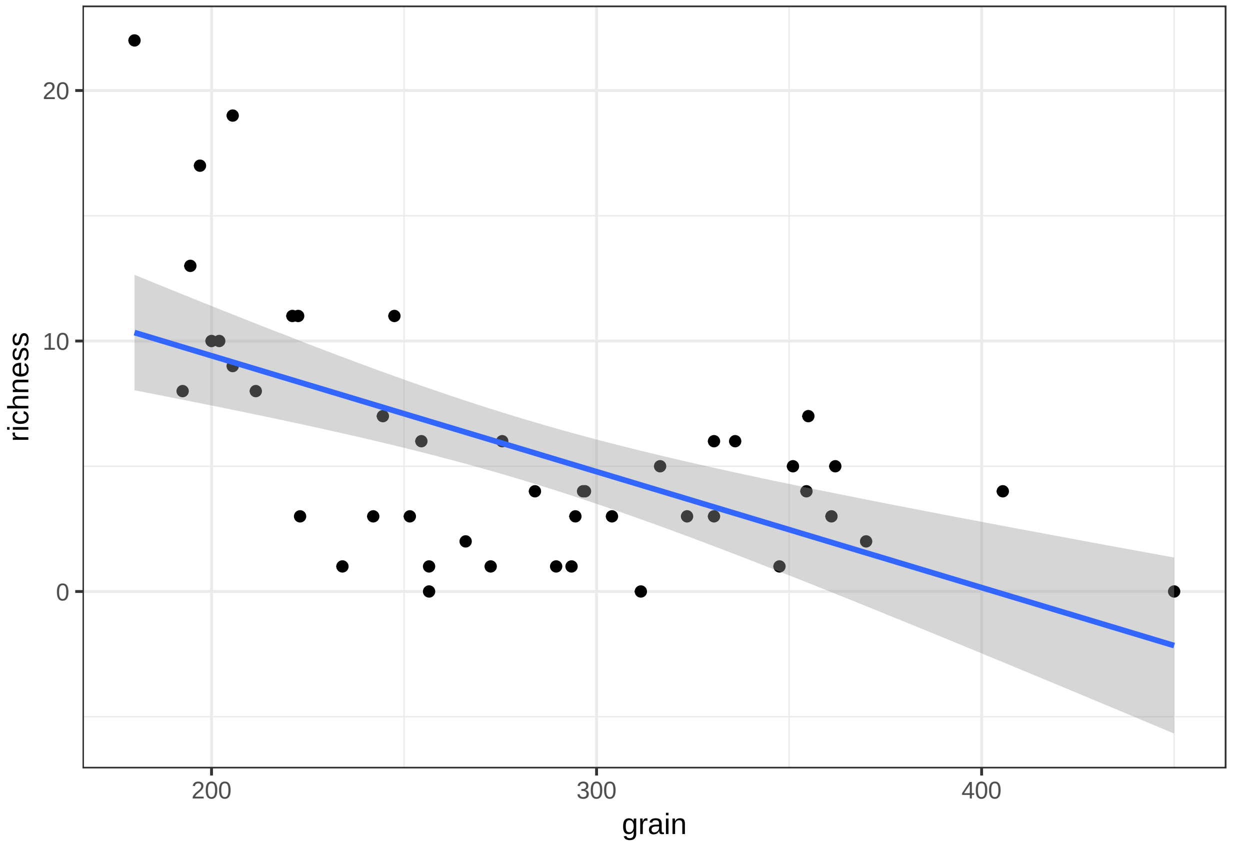
Plotting confidence intervals using base graphics requires a few steps in base graphics, but you can see explicitly that these are constructed using the model predictions.

```r
mod<-lm(richness~grain)
plot(richness~grain)
x<-seq(min(grain),max(grain),length=100)
matlines(x,predict(mod,newdata=list(grain=x),interval="confidence"))
```



It is, of course, usually much quicker and easier to use ggplots.

```r
g2<-g1+geom_smooth(method="lm")
g2
```



The relationship is modelled quite well by a straight line, but it does not look completely convincing. There are two many points above the line on the left hand side of the figure and too many below on the right. If look at the first diagnostic plot for the model this is also apparent. The residuals are correlated as a result of the model not being of the correct form.

Look at the diagnostic plots.

```
anova(mod)
```

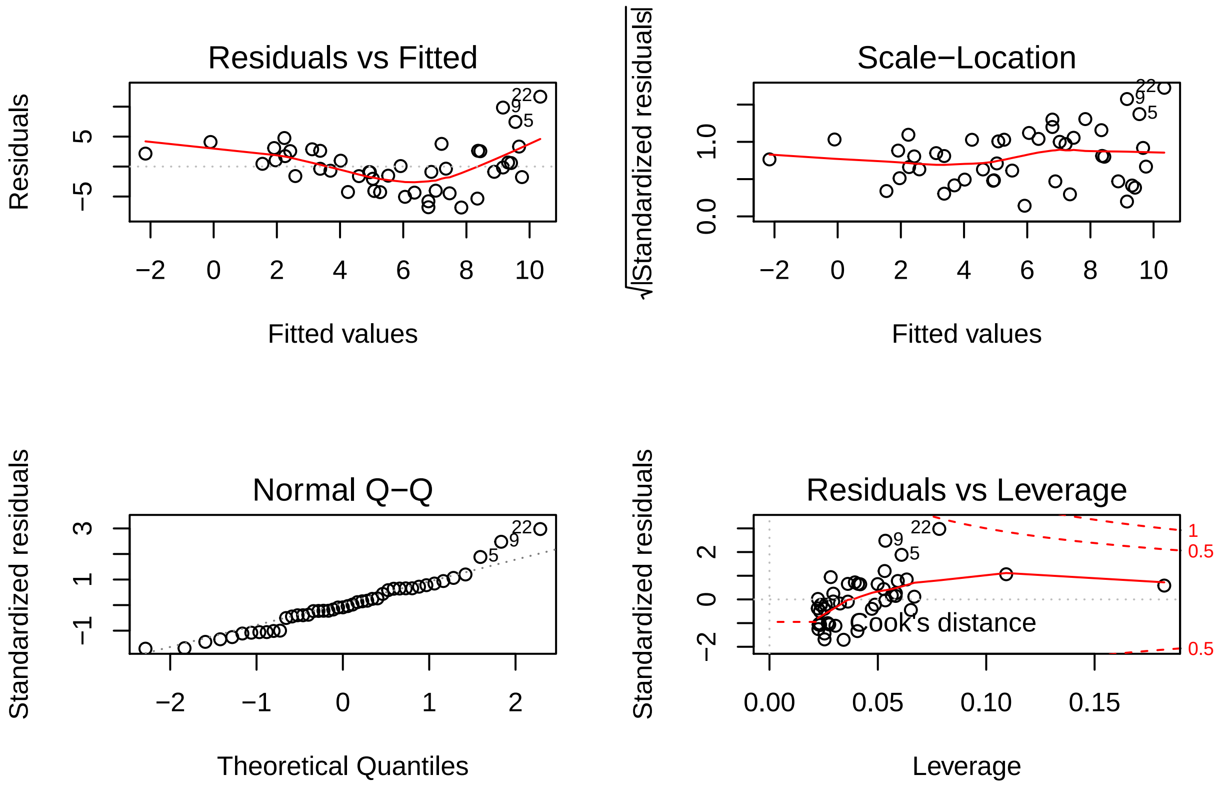
```
## Analysis of Variance Table
##
## Response: richness
##           Df Sum Sq Mean Sq F value    Pr(>F)
## grain      1 385.13  385.13  23.113 1.896e-05 ***
## Residuals 43 716.52   16.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = richness ~ grain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8386 -2.0383 -0.3526  2.5768 11.6620
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.669264   2.767726   6.745 3.01e-08 ***
## grain       -0.046285   0.009628  -4.808 1.90e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.082 on 43 degrees of freedom
## Multiple R-squared:  0.3496, Adjusted R-squared:  0.3345
## F-statistic: 23.11 on 1 and 43 DF,  p-value: 1.896e-05
```

```
par(mfcol=c(2,2))
plot(mod)
```



5.1.2 T values and significance in summary output

An element of the R output that you should be aware of at this point is the summary table. If you look at this table you will see that for each parameter there is an estimate of its value together with a standard error. The confidence interval for the parameter is approximately plus or minus twice the standard error. The T value is a measure of how far from zero the estimated parameter value lies in units of standard error. So generally speaking t values of 2 or more will be statistically significant. This may be useful, but it certainly does **not** suggest that parameter is useful in itself. To evaluate whether a parameter should be included requires taking a "whole model" approach.

Remember that a p-value only tells you how likely you would be to get the value of t (or any other statistic with a known distribution such as an F value) if the null model were true. It doesn't really tell you that much directly about the model you actually have.

5.1.3 Testing for curvilearity

We can check whether a strait line is a good representation of the pattern using the reset test that will have a low p-value if the linear form of the model is not a good fit.

```r
library(lmtest)
resettest(richness ~ grain)
```

```
##
##  RESET test
##
## data:  richness ~ grain
## RESET = 19.074, df1 = 2, df2 = 41, p-value = 1.393e-06
```

The Durbin Watson test which helps to confirm serial autocorrelation that may be the result of a misformed model will often also be significant when residuals cluster on one side of the line.

```r
dwtest(richness~grain)
```

```
##
##  Durbin-Watson test
##
## data:  richness ~ grain
## DW = 1.7809, p-value = 0.1902
## alternative hypothesis: true autocorrelation is greater than 0
```

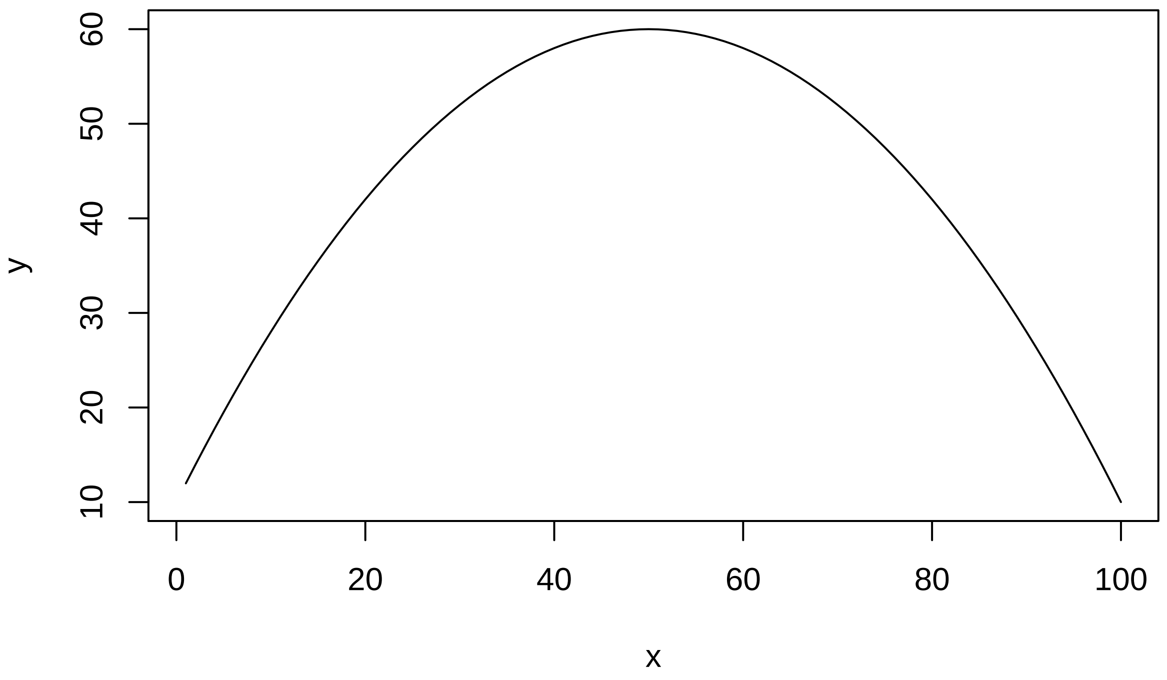
In this case it was not, but this may be because there were too few data points.

5.2 Polynomials

The traditional method for modelling curvilinear relationships when a functional form of the relationship is not assumed is to use polynomials. Adding quadratic, cubic or higher terms to a model gives it flexibility and allows the line to adopt different shapes. The simplest example is a quadratic relationship which takes the form of a parabola.

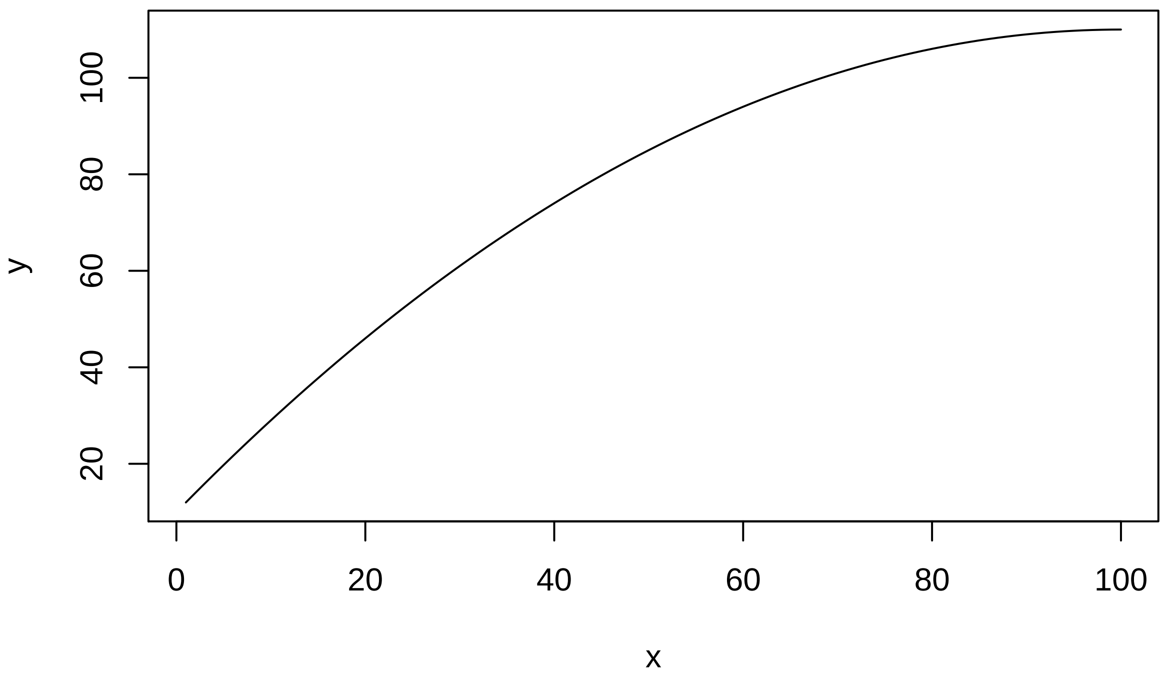
$y = a + bx + cx^2 + \epsilon$ where $\epsilon = N(o, \sigma^2)$

```r
x<-1:100
y<-10+2*x-0.02*x^2
plot(y~x,type="l")
```



If the quadratic term has a small value the curve may look more like a hyperbola over the data range.

```r
x<-1:100
y<-10+2*x-0.01*x^2
plot(y~x,type="l")
```



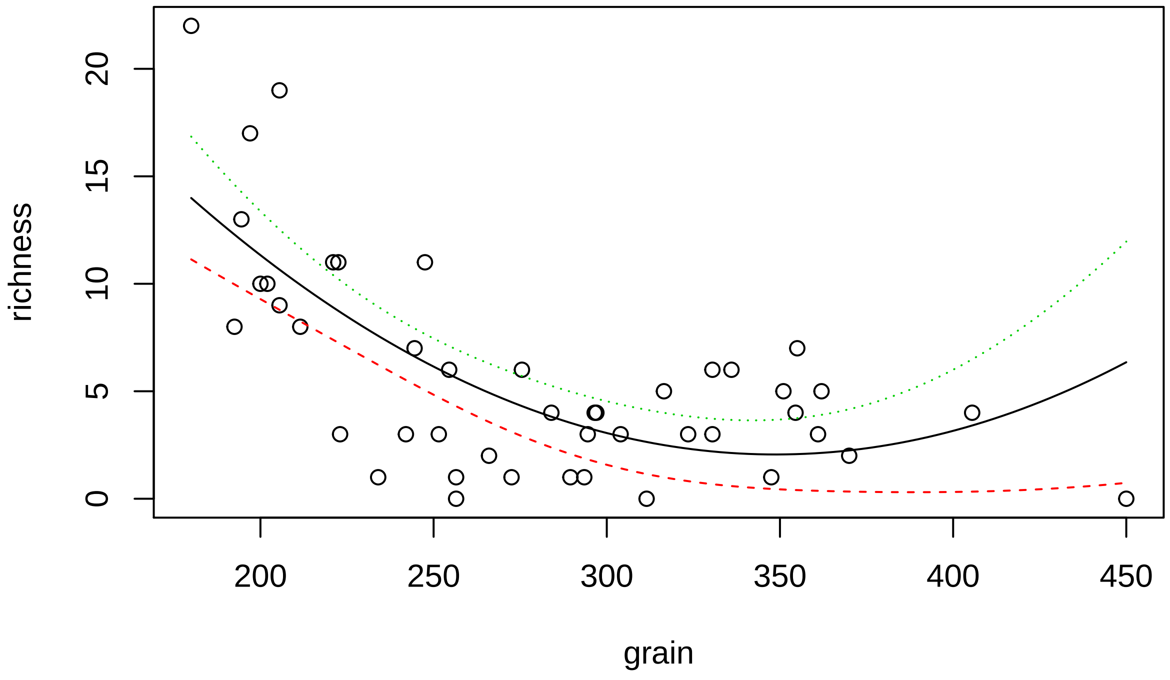
We can add a quadratic term to the model formula in R very easily.

The syntax is lm(richness ~ grain+I(grain^2)).

The “I” is used to isolate the expression so that it is interpreted literally in mathematical terms. Notice that we are still using lm, i.e. a general linear model. This is because mathematically the terms still enter in a “linear” manner. So linear models can produce curves!

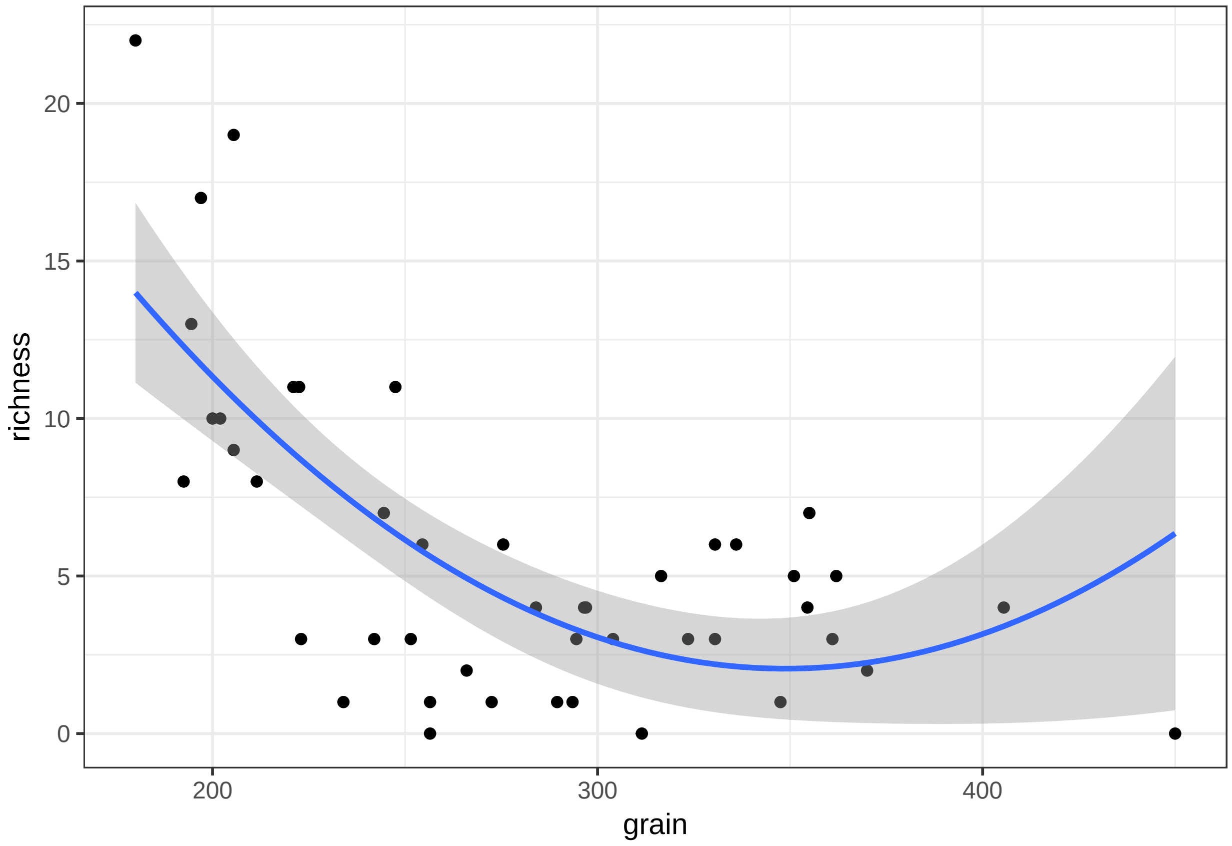
We canplot the predictions explicitly using base graphics.

```
mod2<-lm(richness~grain+I(grain^2))
plot(richness~grain)
x<-seq(min(grain),max(grain),length=100)
matlines(x,predict(mod2,newdata=list(grain=x),interval="confidence"))
```



Or using ggplot.

```
g1+geom_smooth(method="lm",formula=y~x+I(x^2), se=TRUE)
```



```
anova(mod2)
```

```
## Analysis of Variance Table
##
## Response: richness
##            Df Sum Sq Mean Sq F value    Pr(>F)
## grain       1 385.13  385.13  29.811 2.365e-06 ***
## I(grain^2)  1 173.93  173.93  13.463   0.00068 ***
## Residuals  42 542.59   12.92
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod2)
```

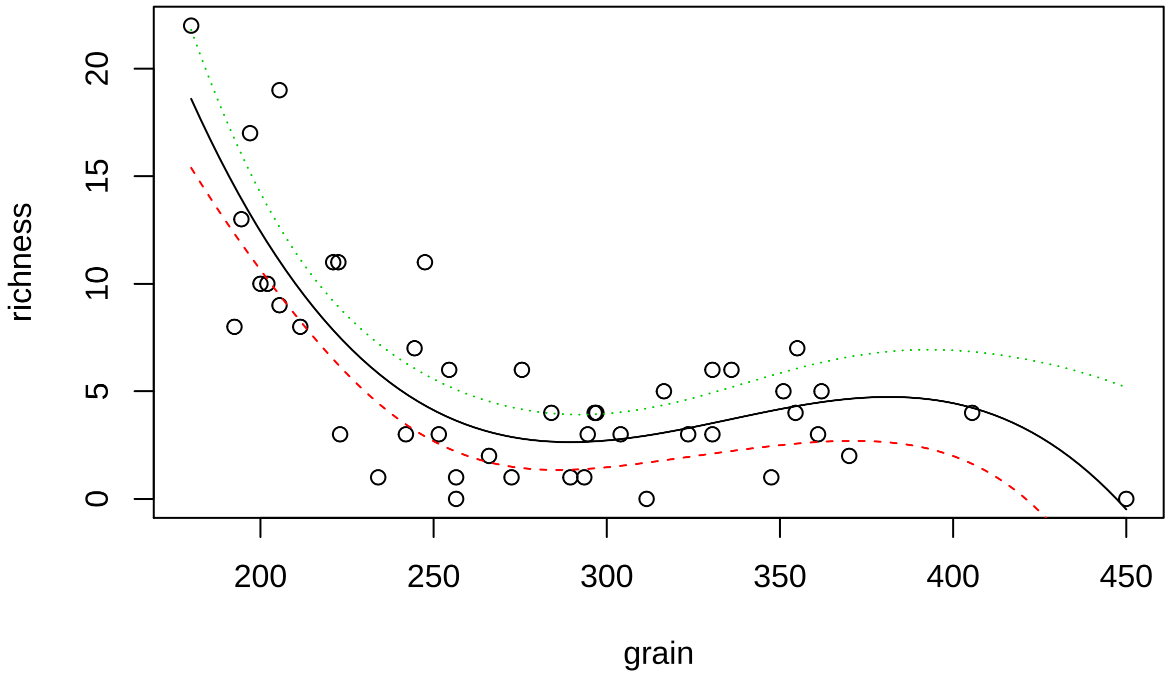
```
##
## Call:
## lm(formula = richness ~ grain + I(grain^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5779 -2.5315  0.2172  2.1013  8.3415
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 53.0133538  9.6721492   5.481 2.21e-06 ***
## grain       -0.2921821  0.0675505  -4.325 9.19e-05 ***
## I(grain^2)   0.0004189  0.0001142   3.669  0.00068 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.594 on 42 degrees of freedom
## Multiple R-squared:  0.5075, Adjusted R-squared:  0.484
## F-statistic: 21.64 on 2 and 42 DF,  p-value: 3.476e-07
```

We now have a higher value of R squared and a better fitting model.

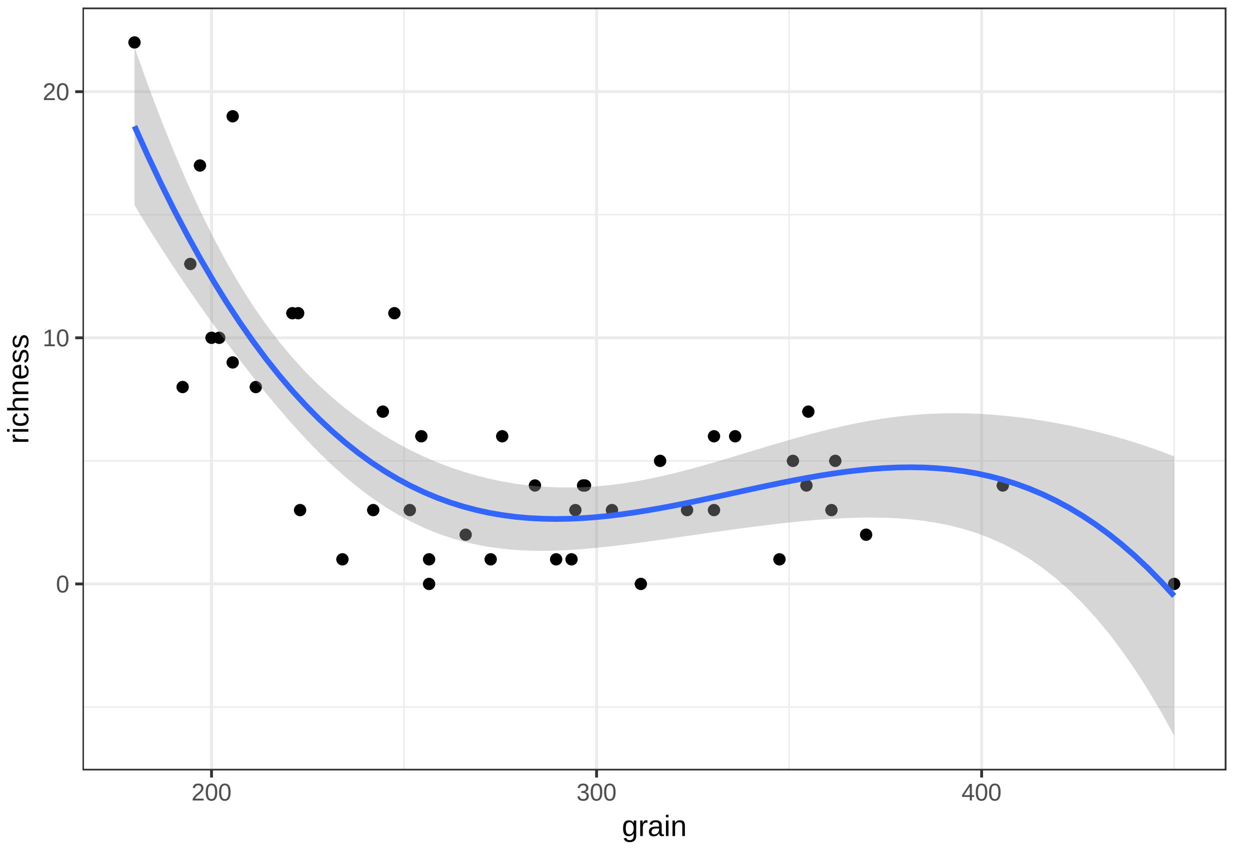
But the shape does not look right. The quadratic is constrained in form and has started to rise at the high end of the x axis. This does not make a great deal of sense.

We can give the model more flexibility by adding another term.

```
mod3<-lm(richness~grain+I(grain^2)+I(grain^3))
plot(richness~grain)
x<-seq(min(grain),max(grain),length=100)
matlines(x,predict(mod3,newdata=list(grain=x),interval="confidence"))
```



```
g1+geom_smooth(method="lm",formula=y~x+I(x^2)++I(x^3), se=TRUE)
```



```
anova(mod3)
```

```
## Analysis of Variance Table
##
## Response: richness
##             Df Sum Sq Mean Sq F value    Pr(>F)
## grain        1 385.13  385.13  42.542 7.820e-08 ***
## I(grain^2)   1 173.93  173.93  19.212 7.944e-05 ***
## I(grain^3)   1 171.42  171.42  18.936 8.768e-05 ***
## Residuals   41 371.17    9.05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod3)
```

```
##
## Call:
## lm(formula = richness ~ grain + I(grain^2) + I(grain^3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5250 -1.8841 -0.2896  2.2259  7.9431
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.955e+02  3.373e+01   5.796 8.44e-07 ***
## grain       -1.784e+00  3.474e-01  -5.134 7.27e-06 ***
## I(grain^2)   5.420e-03  1.153e-03   4.700 2.93e-05 ***
```

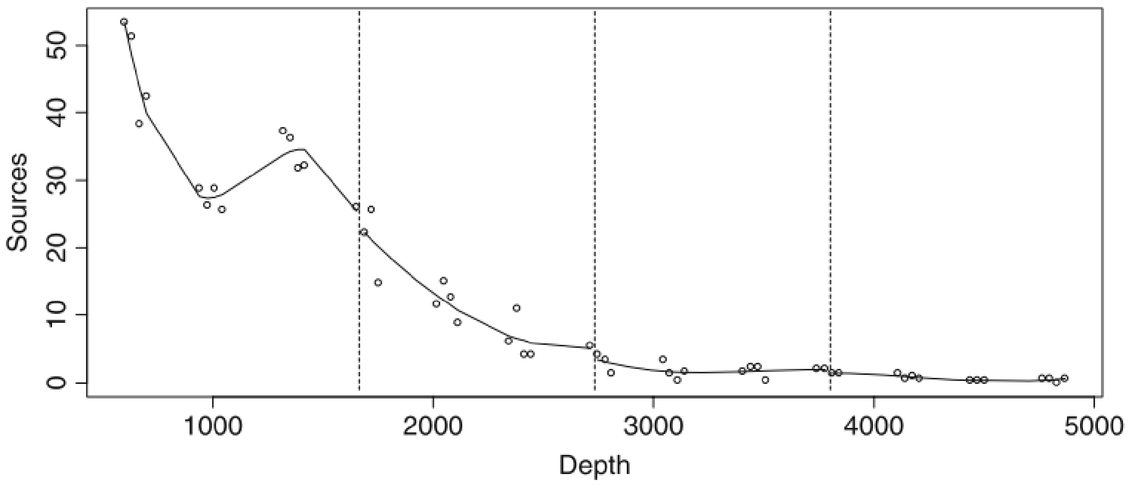


Figure 5.1: alt text

```
## I(grain^3)  -5.386e-06  1.238e-06  -4.352 8.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.009 on 41 degrees of freedom
## Multiple R-squared:  0.6631, Adjusted R-squared:  0.6384
## F-statistic:  26.9 on 3 and 41 DF,  p-value: 8.838e-10
```

This produces a better fit statistically, but things are now getting very confusing. It is not clear what the third term is actually doing. The confidence intervals are quite wide, so we could ignore the sharp downturn, as any shape within the confidence intervals is permissible. But the model still does not look right.

The advantages of polynomials is that they do result in a formula that can be written down and provided as a predictive model. The major disadvantage is that the formula is rather complex and has no intrinisic biological or ecological basis. You must also be very careful never to use these models to predict values that fall outside the range used for fitting. Also the formulae produced by fitting polynomials are often used without regard to the confidence intervals. Uncertainty is part of the statistical model and should be taken into account.

5.3 Splines

A commonly used alternative to polynomials is to fit a so called smoother of some description. There are many different ways to go about this, making the subject seem complicated. However for most practical purposes they produce similar results and we can rely on the software to make most of the decisions.

The most commonly used smoothers are splines of some type. These work by fitting curves to sections of the data and then splicing the results together. This gives the curves much greater flexibility that polynomials. Almost any shape can be fitted.

The issue with this involves complexity. If we let the curves become too flexible we could fit a line to that passed through all the data points. But this would not be useful and would leave no degrees of freedom. The degree of waviness is selected in R automatically by cross validation if we use the mgcv package. There is no guarantee that the model will be biologically meaningful, but many times the selection produces a curve that fits the data well and can be interpreted.

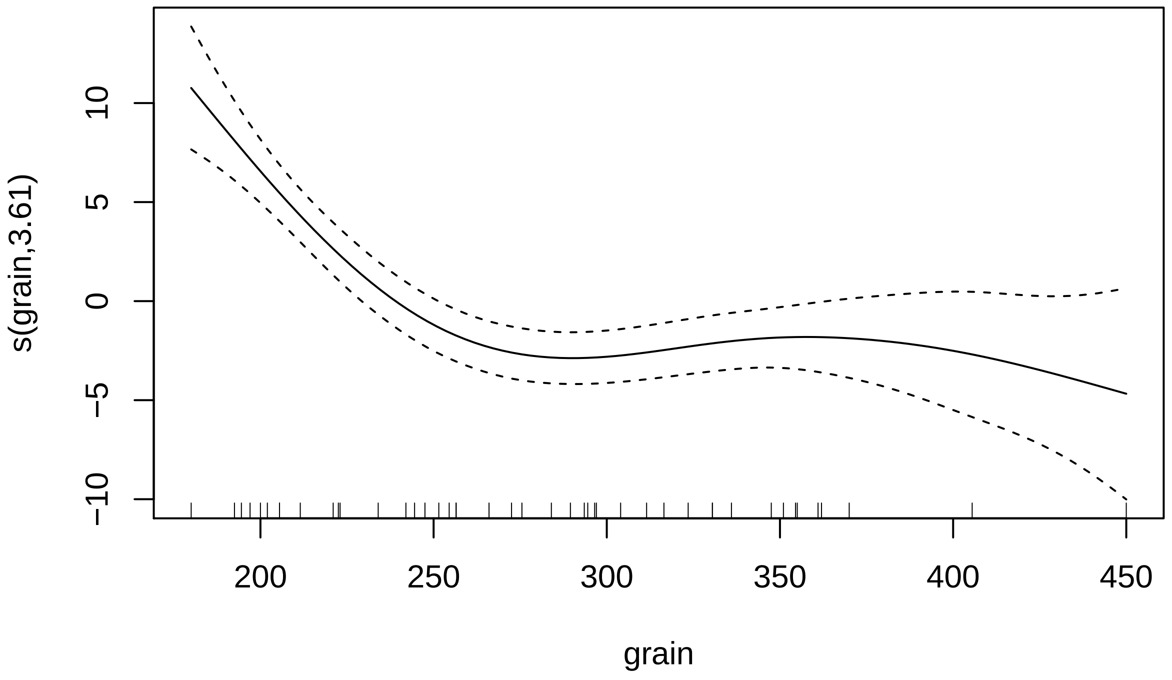
```
library(mgcv)
mod4<-gam(richness~s(grain))
summary(mod4)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## richness ~ s(grain)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.6889     0.4601   12.36  2.6e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F  p-value
## s(grain) 3.615  4.468 15.92 3.25e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.619   Deviance explained = 65.1%
## GCV = 10.616  Scale est. = 9.5269    n = 45
```

The summary gives a p-value for the term and also shows the estimated degrees of freedom. The more complex the response, the more degrees of freedom are used in fitting the model.

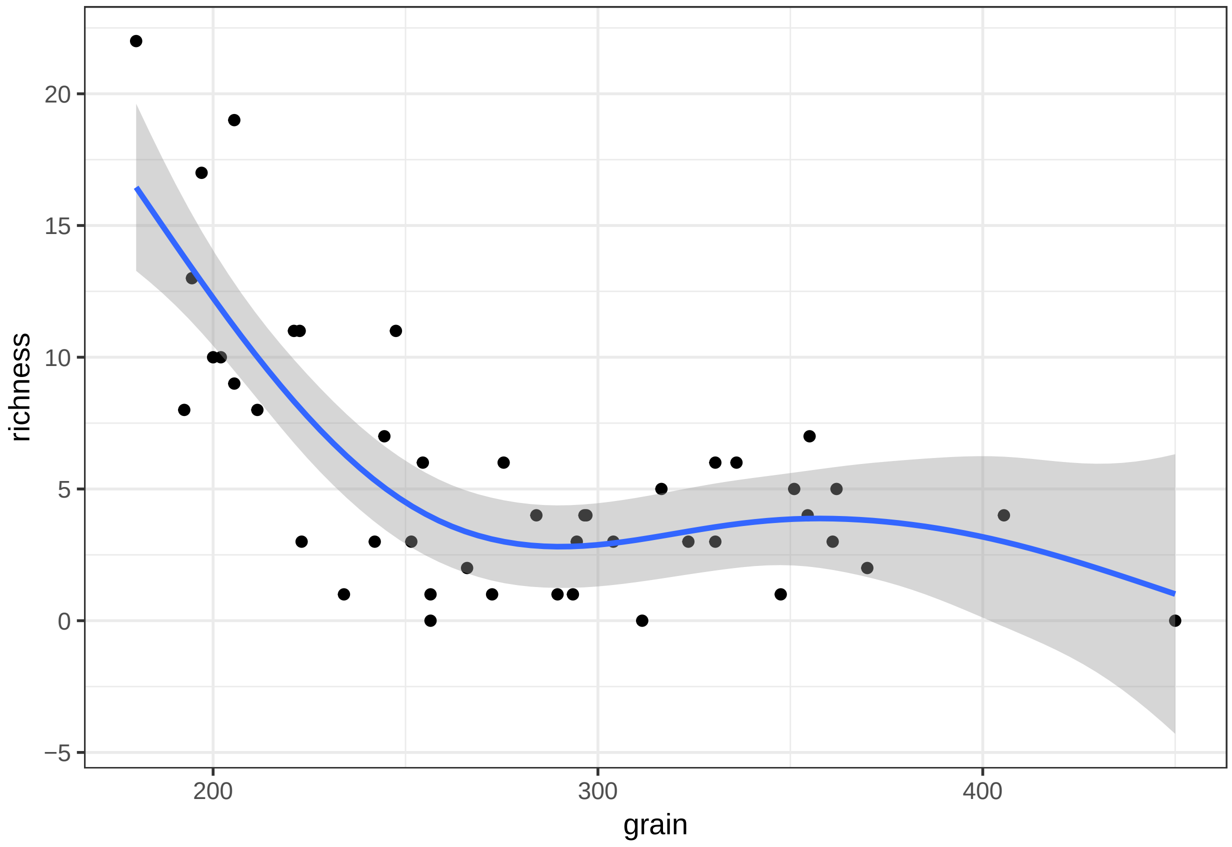
Unlike regression models there is no formula associated with the model. So it can be difficult to communicate the results. The usual way of presenting the model is graphically.

```
plot(mod4)
```



Notice that the plot shows differences from the mean value (intercept) associated with the smoothed term in the model. Splines can suffer from some of the same problems as polynomials, but they often lead to a curve that has more intrinsic meaning.

```
g1+stat_smooth(method = "gam", formula = y ~ s(x))
```



5.4 Complex shapes

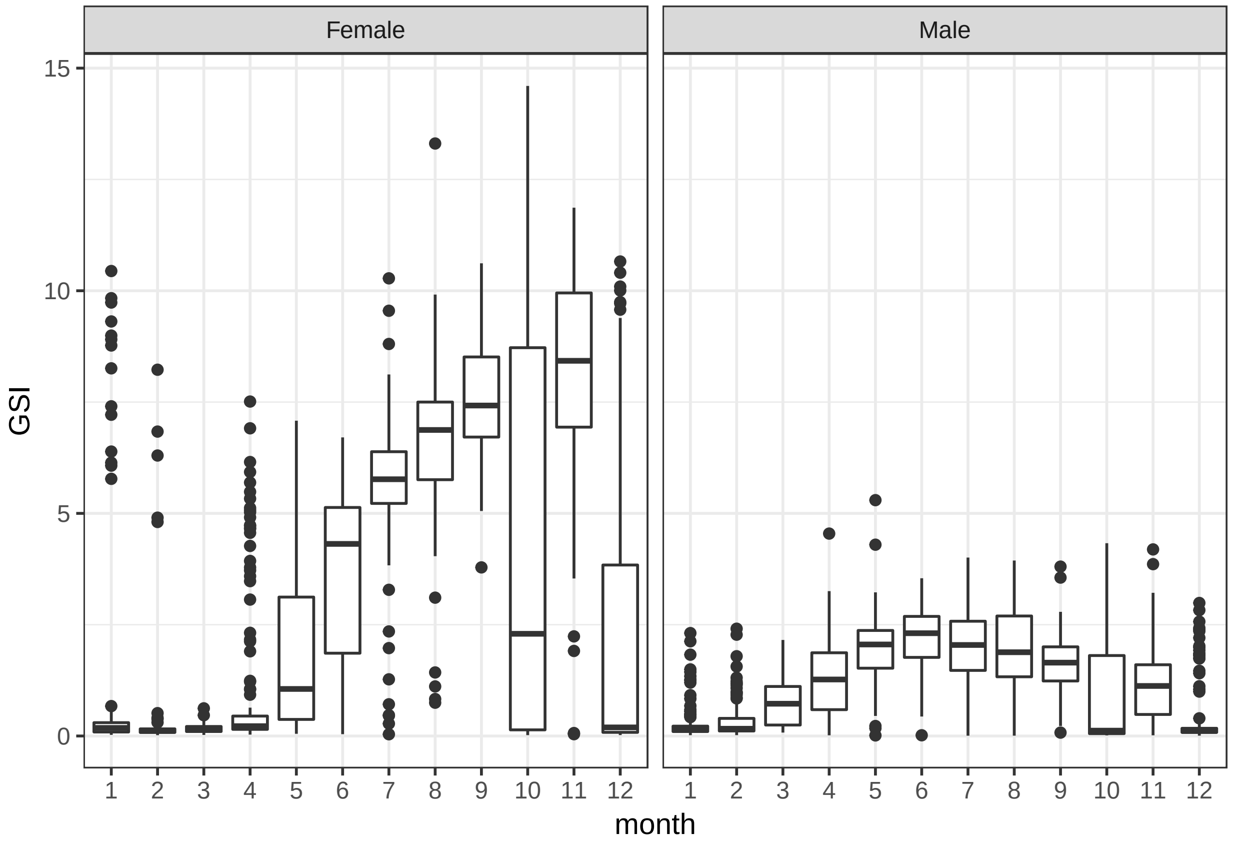
The next data set is on the Gonadosomatic index (GSI, i.e., the weight of the gonads relative to total body weight) of squid Measurements were taken from squid caught at various locations and months in Scottish waters.

```
squid<-read.csv("/home/aqm/course/data/squid.csv")
```

We can plot out the data using a conditional box and whisker plot.

```
squid$month<-as.factor(squid$MONTH)
```

```
g0<-ggplot(data=squid,aes(x=month,y=GSI))
g1<-g0+geom_boxplot()
g1+facet_wrap("Sex")
```

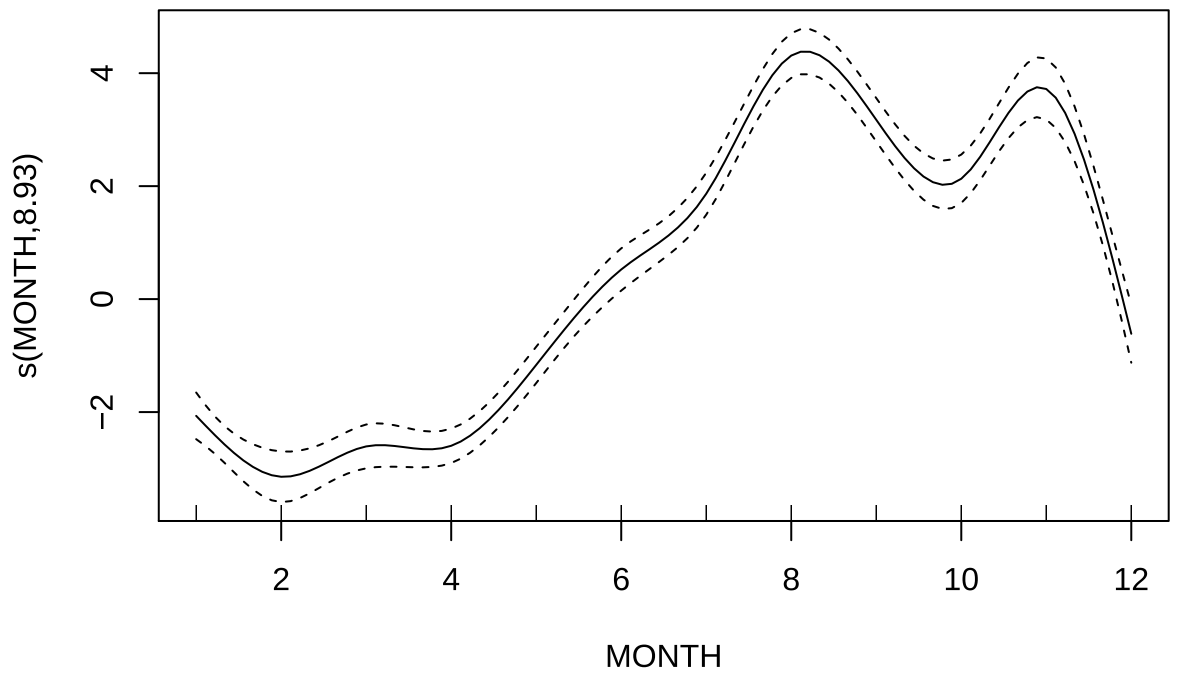


It seems sensible to split the data by gender.

```r
males<-subset(squid,Sex=="Male")
females<-subset(squid,Sex=="Female")
```

Now we can try representing the pattern of change over the year using a spline model fit using mgcv.

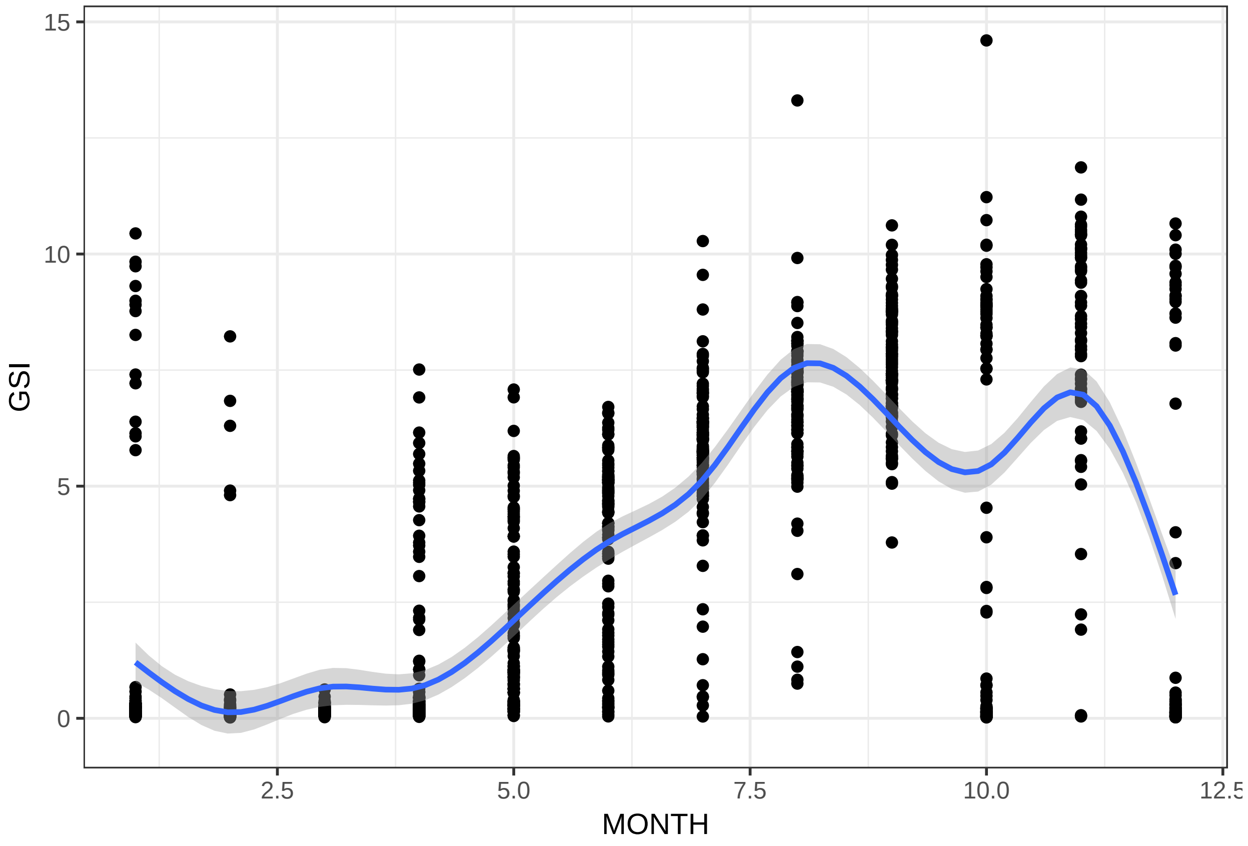
```r
mod1<-gam(GSI~s(MONTH),data=females)
plot(mod1)
```



```
summary(mod1)
```

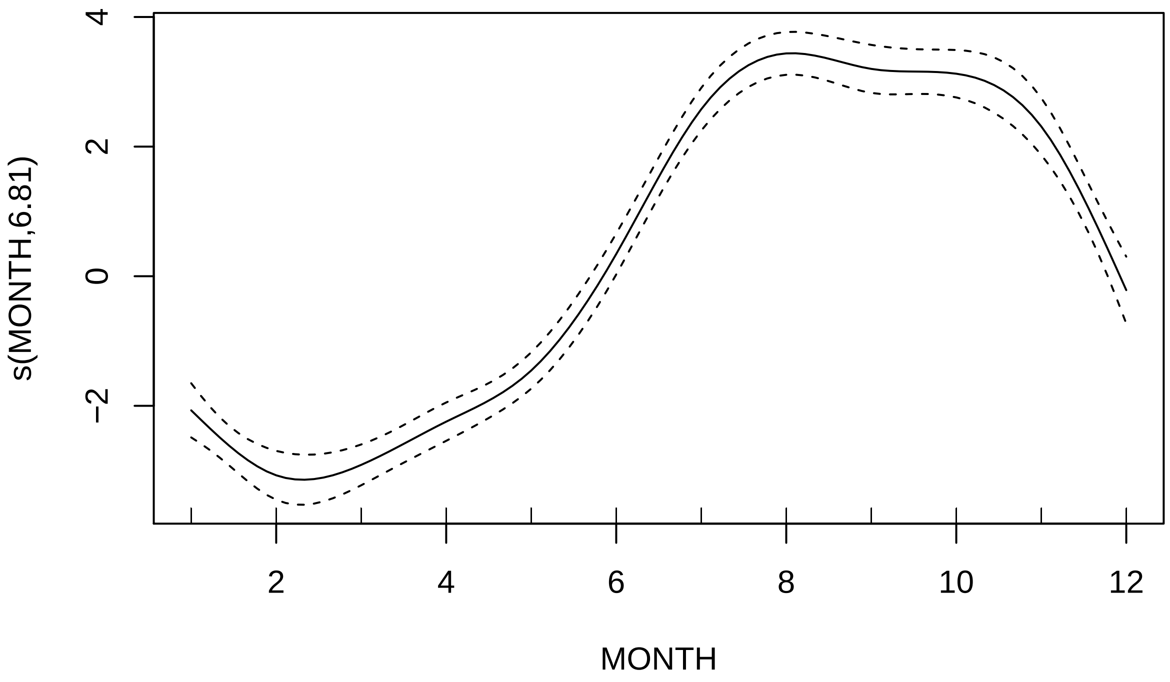
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## GSI ~ s(MONTH)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.27244    0.06651    49.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F p-value
## s(MONTH) 8.93  8.999 156.7  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.53   Deviance explained = 53.4%
## GCV = 5.5387  Scale est. = 5.4944    n = 1242
```

```
g0<-ggplot(data=females, aes(x=MONTH,y=GSI))
g1<-g0+geom_point()
g1+stat_smooth(method = "gam", formula = y ~ s(x))
```



The statistics are OK, but the biology seems wrong. The dip in the curve in October does not seem to make sense. Although the number of knots in the spline are determined by cross validation we can lower them in order to produce a simpler model.

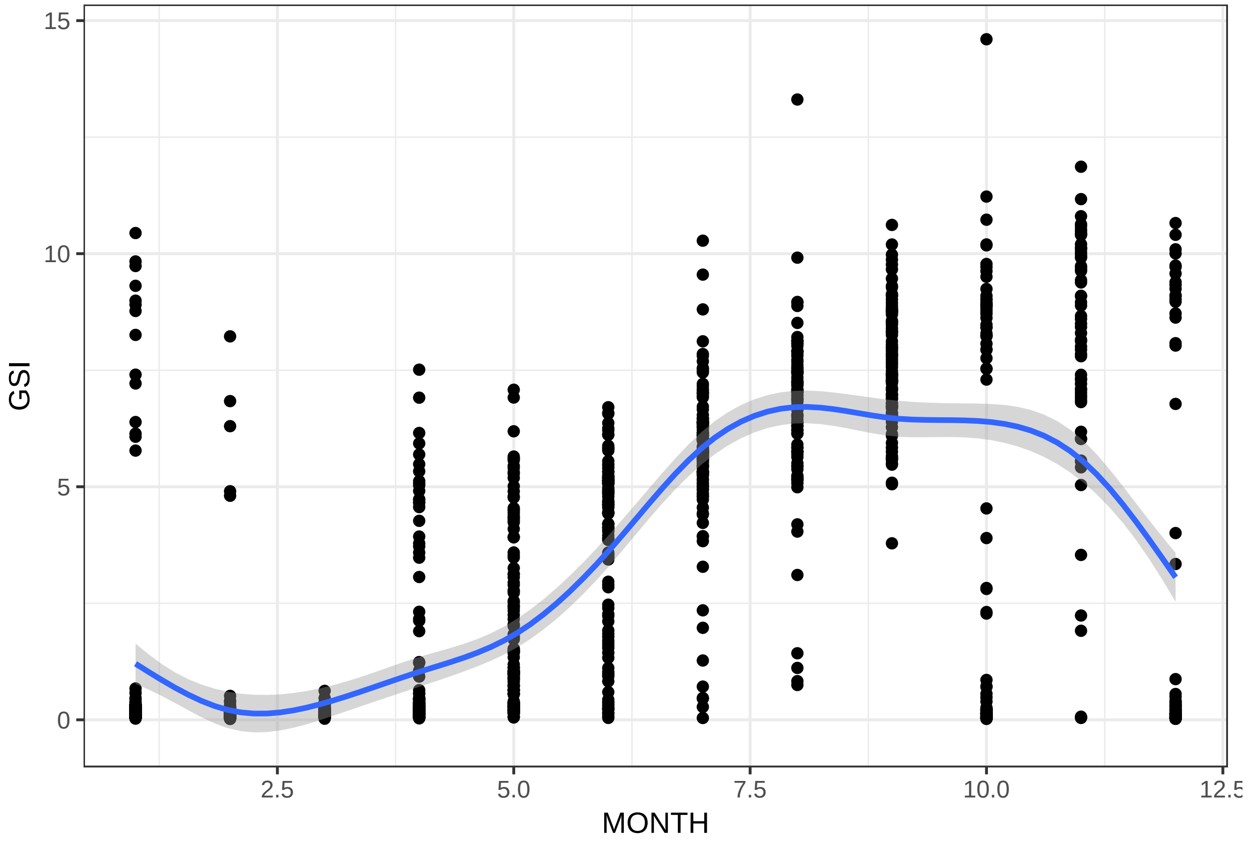
```
mod2<-gam(GSI~s(MONTH,k=8),data=females)
plot(mod2)
```



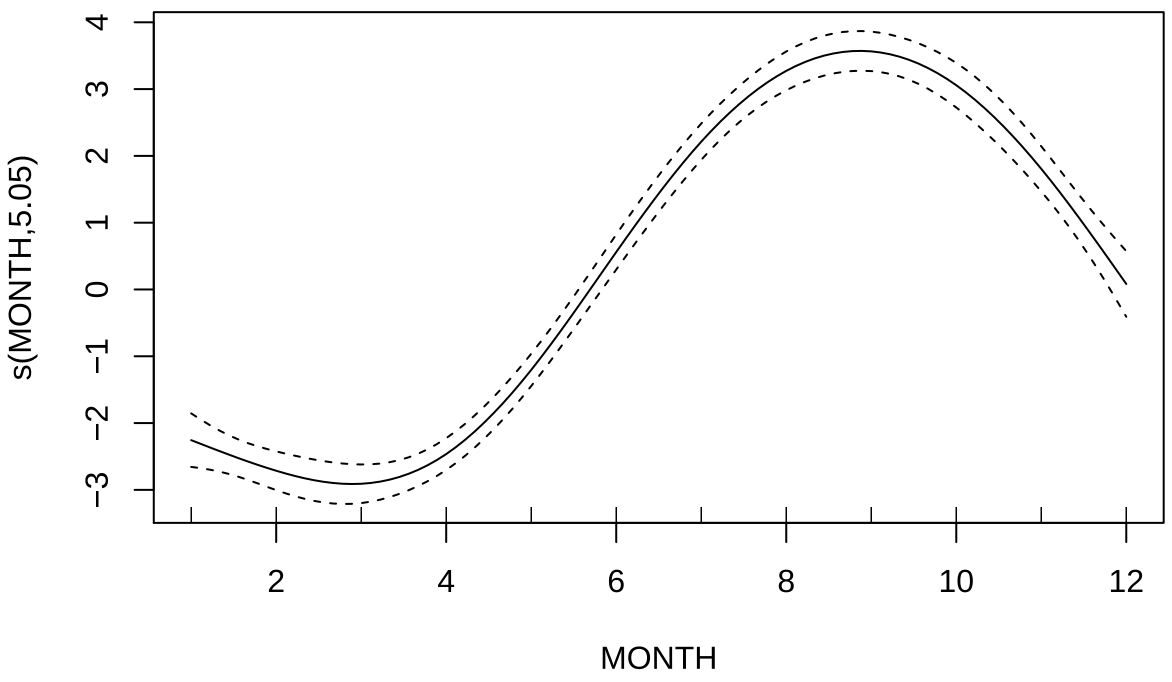
```
summary(mod2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## GSI ~ s(MONTH, k = 8)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.27244    0.06848   47.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F p-value
## s(MONTH) 6.815  6.987 179.8  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.502   Deviance explained = 50.5%
## GCV = 5.8608  Scale est. = 5.8239    n = 1242
```

```
g1+stat_smooth(method = "gam", formula = y ~ s(x,k=8))
```



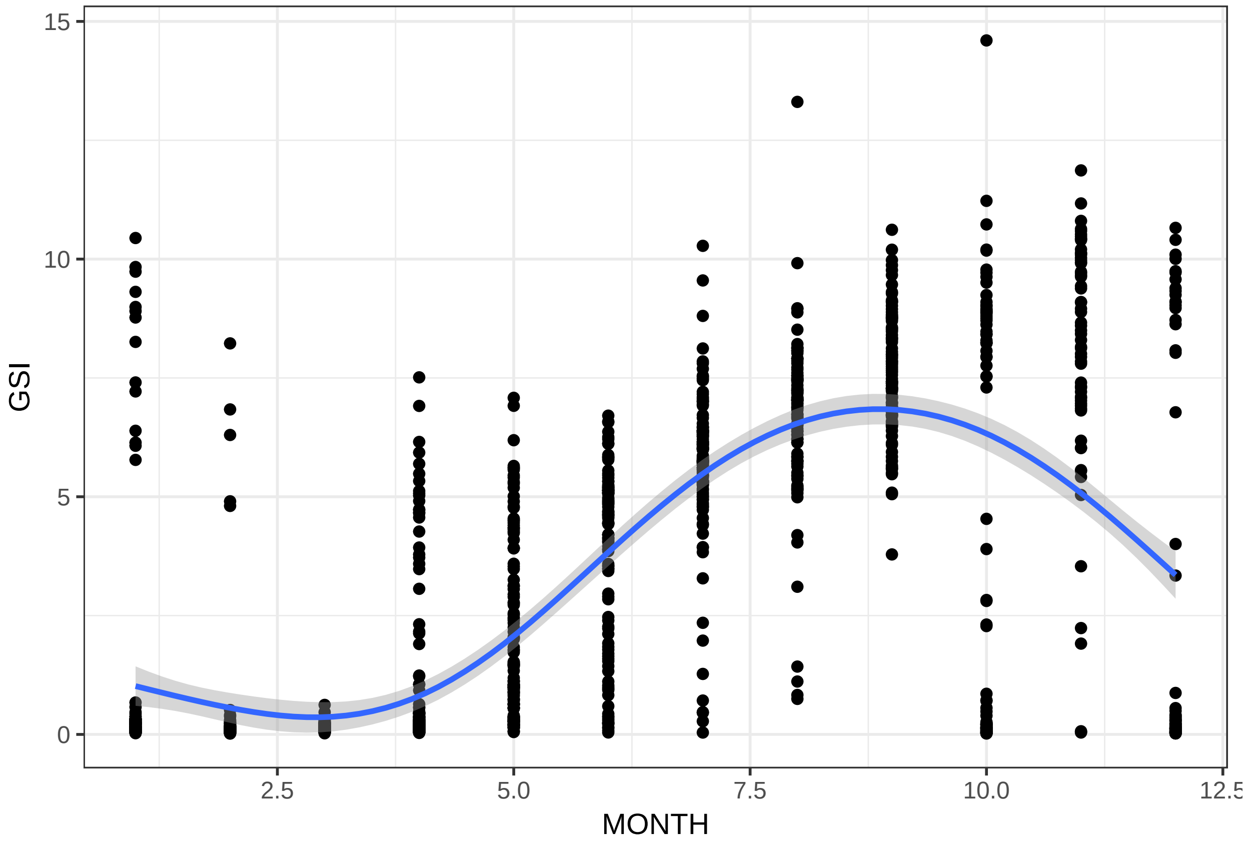
```
mod3<-gam(GSI~s(MONTH,k=7),data=females)
plot(mod3)
```



```
summary(mod3)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## GSI ~ s(MONTH, k = 7)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.27244    0.06896   47.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F p-value
## s(MONTH) 5.047  5.655 213.7  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.495   Deviance explained = 49.7%
## GCV = 5.9348  Scale est. = 5.9059    n = 1242
```

```
g1+stat_smooth(method = "gam", formula = y ~ s(x,k=7))
```



By reducing the number of knots we have models which use fewer degrees of freedom. However the fit as measured by the deviance explained (R squared in this case) is reduced.

We can test whether the first model is significantly better. We find that it is.

```
anova(mod1,mod2,mod3,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: GSI ~ s(MONTH)
## Model 2: GSI ~ s(MONTH, k = 8)
## Model 3: GSI ~ s(MONTH, k = 7)
##   Resid. Df Resid. Dev      Df Deviance      F    Pr(>F)
## 1    1232.0     6769.5
## 2    1234.0     7187.8 -2.0115  -418.25 37.844 < 2.2e-16 ***
## 3    1235.3     7299.4 -1.3317  -111.67 15.263 1.391e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So we are left with a difficulty. The statistical criteria lead to a wavy model, while common sense suggest that there is some issue that has not been taken into account. In such a situation we should look for additional variables that have not been measured. Perhaps some of the squid that were caught actually came from a separate population with a different timing of reproduction.

Chapter 6

Non-linear models

Statistical modelling involves measures of fit. However scientific modelling often brings in other elements, including theory that is used to propose a model for the data. These theoretical models are often non-linear in the statistical sense. The terms do not enter in a linear manner. Such models can be difficult to fit to real life data, but are often used when building process based ecological models.

6.1 Fitting a rectangular hyperbola

For example, resource use is commonly modelled using a function with an asymptote. The equation below is a version of Holling's disk equation that has been rewriten as a generalised rectangular hyperbola. This is identical in form to the Michaelis-Menton equation for enzyme kinetics.

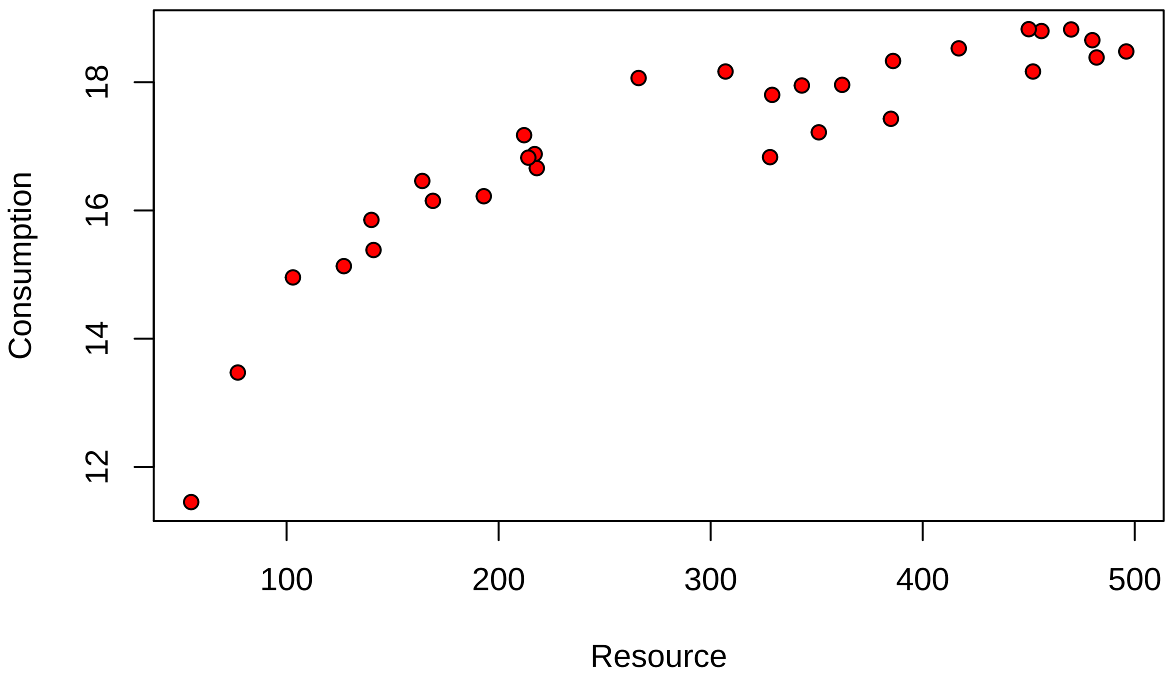
$$C = \frac{sR}{F + R}$$

Where

- C is resource consumption,
- R is the amount or density of theresource,
- s is the asymptotic value and
- F represents the density of resource at which half the asymptotic consumption is expected to occur. This model is not linear in its parameters.

The data below have been simulated from this equation in order to illustrate the model fitting process.

```r
d<-read.csv("/home/aqm/course/data/Hollings.csv")
plot(d,pch=21,bg=2)
```

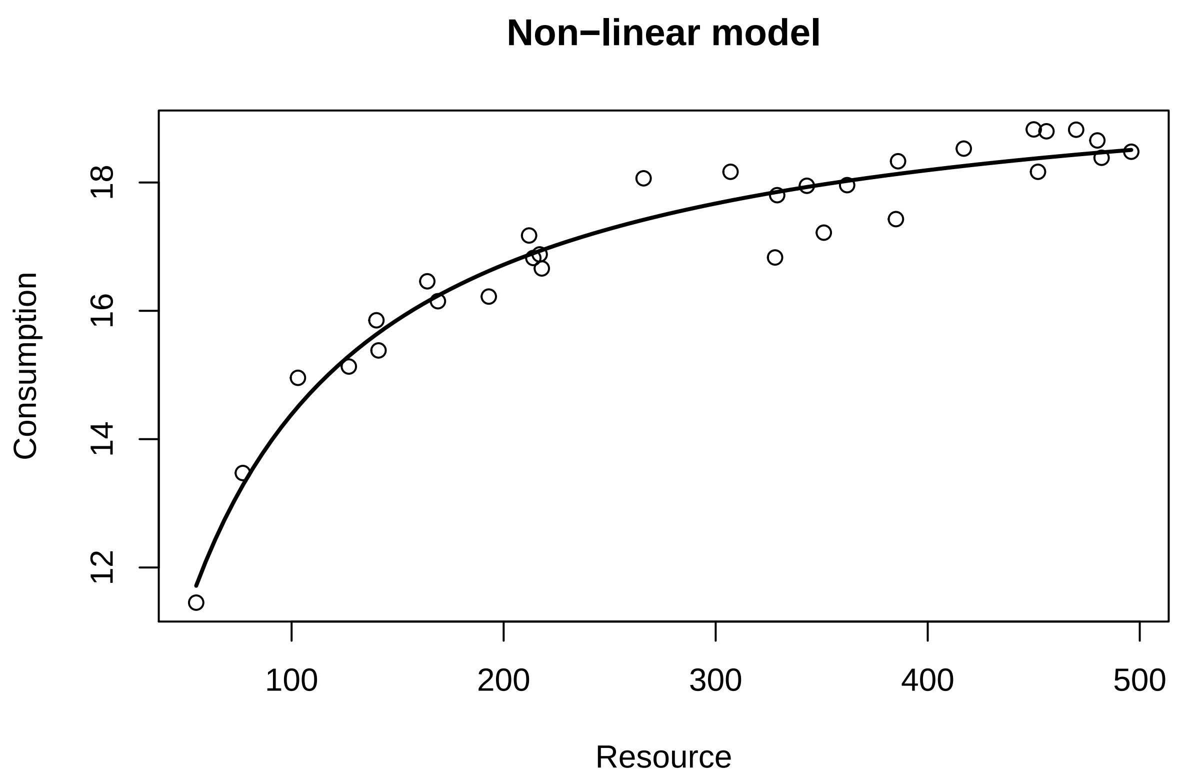


There are many ways to fit a theoretical non-linear model in R. One of the simplest uses least squares, so some of the assumptions made are similar to linear regression.

It is much harder to run diagnosis on these sorts of models. Even fitting them can be tricky. Diagnostics are arguably less important, as the interest lies in finding a model that matches our understanding of how the process works, rather than fitting the data as well as possible.

We have to provide R with a list of reasonable starting values for the model. At present R does not provide confidence bands for plotting, but this is less important in the case of non linear models. It is much more interesting to look at the confidence intervals for the parameters themselves.

```
nlmod<-nls(Consumption~s*Resource/(F+Resource),data = d,start = list( F = 20,s=20))
newdata <- data.frame(Resource=seq(min(d$Resource),max(d$Resource),l=100))
a <- predict(nlmod, newdata = newdata)
plot(d,main="Non-linear model")
lines(newdata$Resource,a,lwd=2)
```

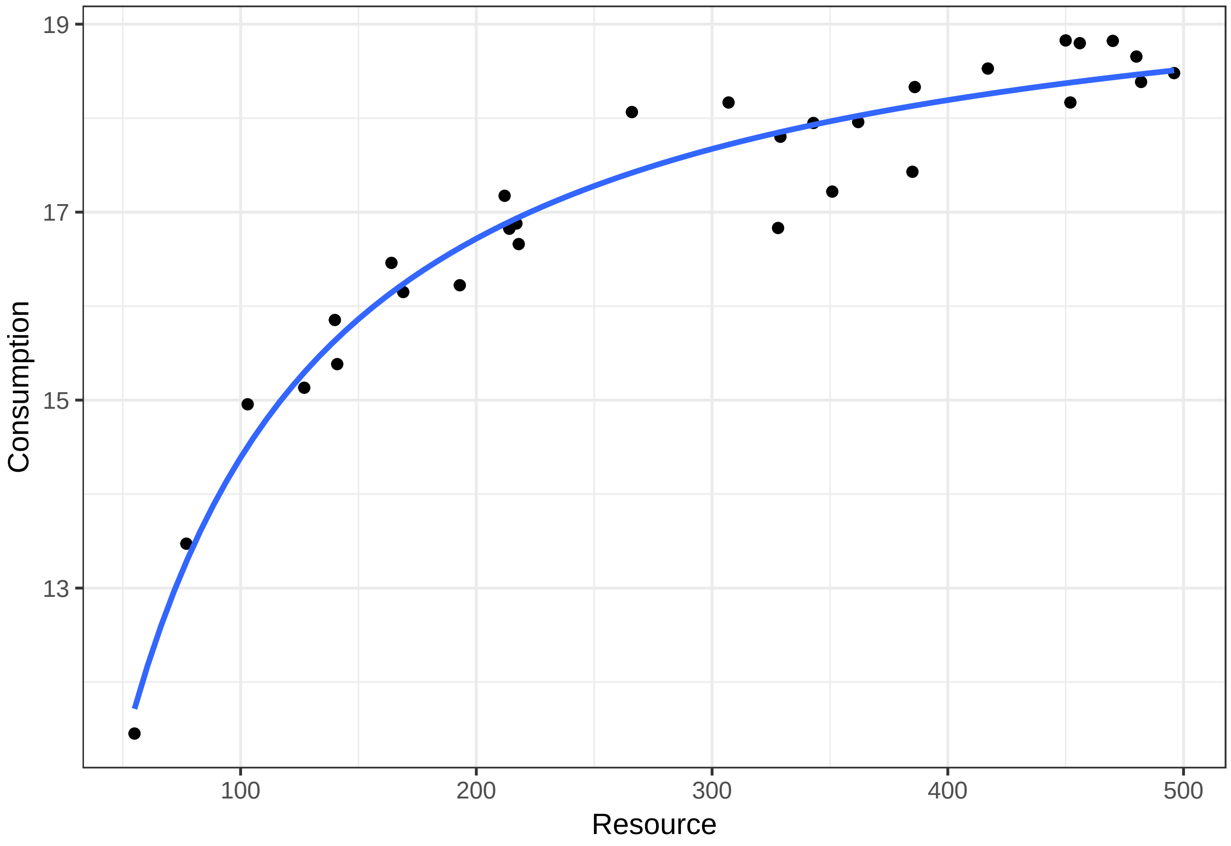


```
confint(nlmod)
```

```
## Waiting for profiling to be done...
```

```
##       2.5%    97.5%
## F 33.97300 43.66697
## s 19.58576 20.32930
```

```
g0<-ggplot(data=d,aes(x=Resource,y=Consumption))
g1<-g0+geom_point()
g1+geom_smooth(method="nls",formula=y~s*x/(F+x),method.args=list(start = c( F = 20,s=20)), se=FALSE)
```



The nice result of fitting this form of model is that we can now interpret the result in the context of the process. If the resource represented biomass of ragworm in mg m2 and consumption feeding rate of waders we could now estimate the maximum rate of feeding lies between 19.5 and 20.3 mg hr. The density at which birds feed at half this maximum rate, if the theoretical model applies, lies beyond the range of the data that we obtained (34 to 44 mg m2).

Non linear models can thus be extrapolated beyond the data range in a way that linear models, polynomials and splines cannot. However this extrapolation relies on assuming that the functional form of the model is sound.

It is common to find that parameters are estimated with very wide confidence intervals even when a model provides a good fit to the data. Reporting confidence intervals for key parameters such as the asymptote is much more important in this context than reporting R2 values. This can be especially important if the results from non linear model fitting are to be used to build process models.

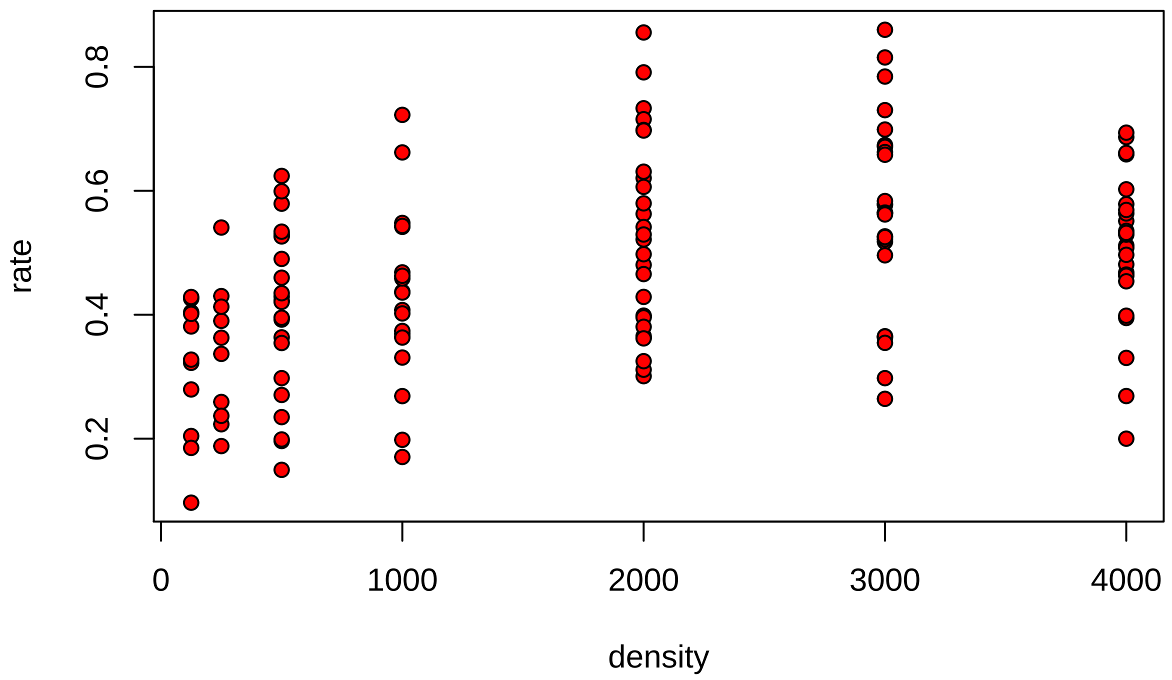
You should not usually fit a non-linear model of this type to data, unless you have an underlying theory. When the data are taken from a real life situation it is always best to explore them first using other approaches before thinking about fitting a non-linear model. When we obtain data from nature we do not know that the equation really does represent the process. Even if it does, there will certainly be quite a lot of random noise around the underlying line.

6.2 Real data

Smart, Stillman and Norris were interested in the functional responses of farmland birds. In particular they wished to understand the feeding behaviour of the corn bunting *Miliaria calandra* L, a bird species whose decline may be linked to a reduction of food supply in stubble fields.

The authors tested five alternative models of the functional responses of corn buntings. They concluded that Holling's disk equation provided the most accurate fit to the observed feeding rates while remaining the most statistically simple model tested.

```r
d<-read.csv("/home/aqm/course/data/buntings.csv")

plot(rate~density,data=d,pch=21,bg=2)
```



The classic version of Hollings disk equation used in the article is written as

$$R = \frac{aD}{1 + aDH}$$

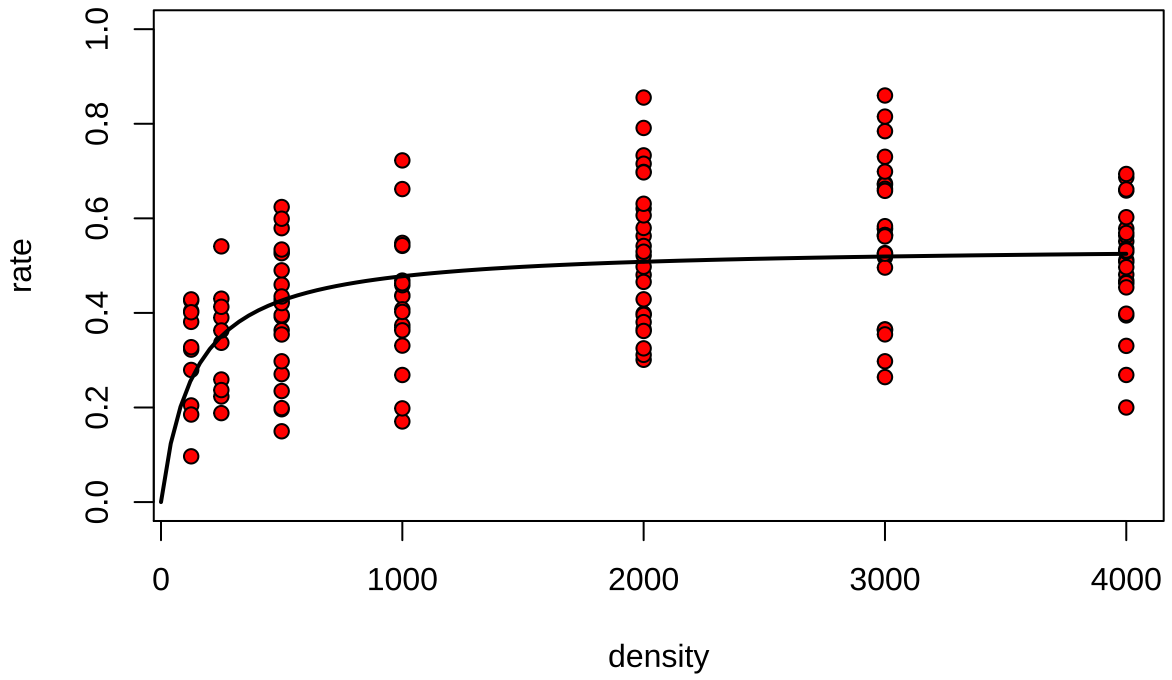
Where

- F = feeding rate (food items)
- D = food density (food items m^{-2})
- a = searching rate ($m^2 s^{-1}$)
- H = handling time (s per food item).

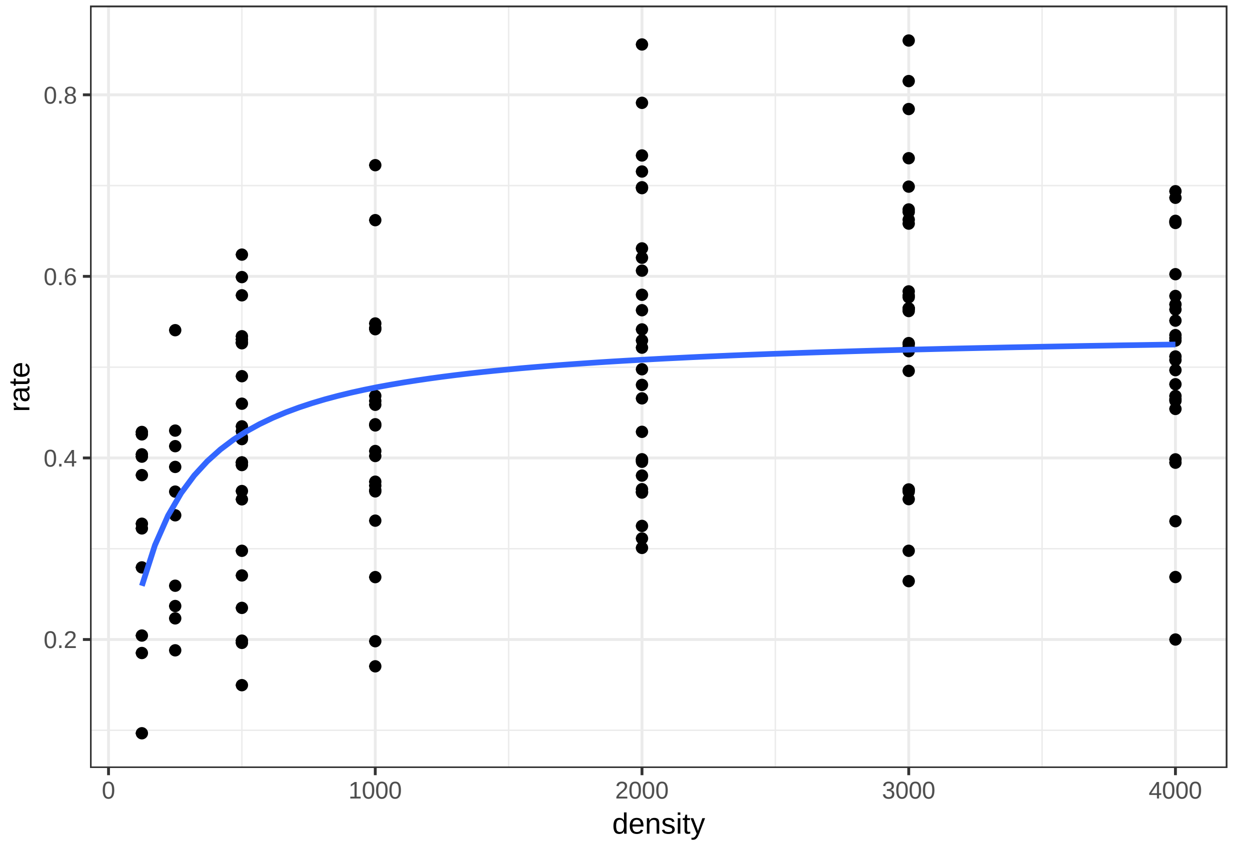
```r
HDmod<-nls(rate~a*density/(1+a*density*H),data = d,start = list(a =0.001,H=2))
confint(HDmod)
```

```
## Waiting for profiling to be done...

##          2.5%       97.5%
## a 0.002593086 0.006694939
## H 1.713495694 1.976978655
```

```r
newdata <- data.frame(density=seq(0,max(d$density),l=100))
HDpreds <- predict(HDmod, newdata = newdata)
plot(rate~density,data=d,pch=21,bg=2,ylim=c(0,1))
lines(newdata$density,HDpreds,lwd=2)
```



```
g0<-ggplot(data=d,aes(x=density,y=rate))
g1<-g0+geom_point()
g1+geom_smooth(method="nls",formula=y~a*x/(1+a*x*H),method.args=list(start = c(a = 0.01,H=2)), se=FALSE
```



The figure in the article shows the five models that were considered.

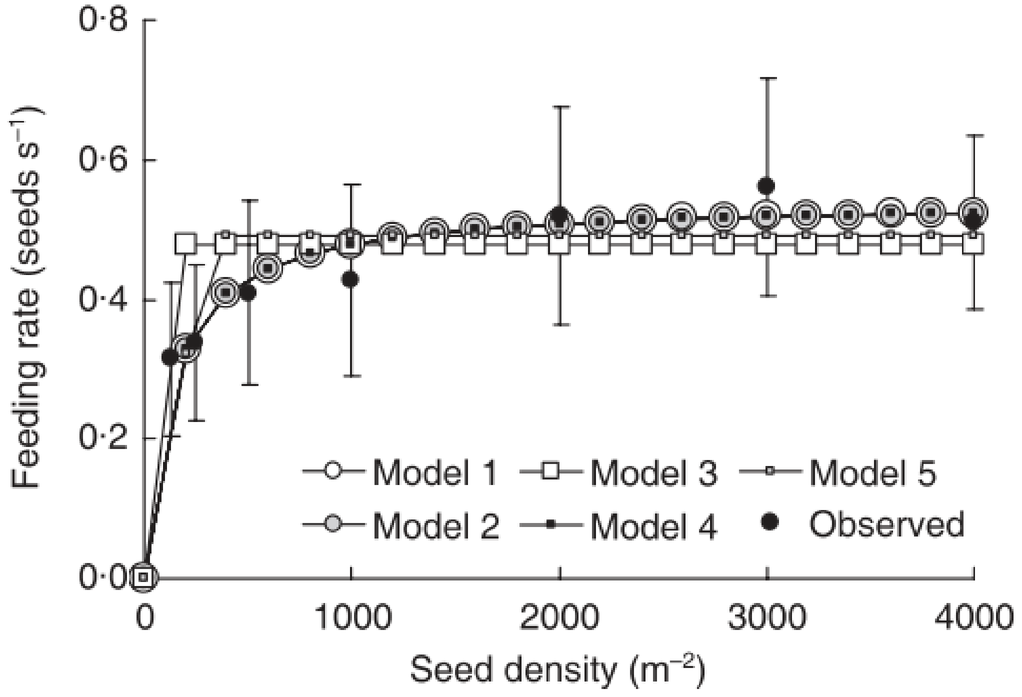


Fig. 3. Fit of each functional response model to the observed relationship between feeding rate and seed density. The solid circles show the mean observed feeding rate for each seed density (with associated standard deviation). See Table 2 for best-fitting parameter estimates for each model, which were derived from the raw data.

Figure 6.1: alt

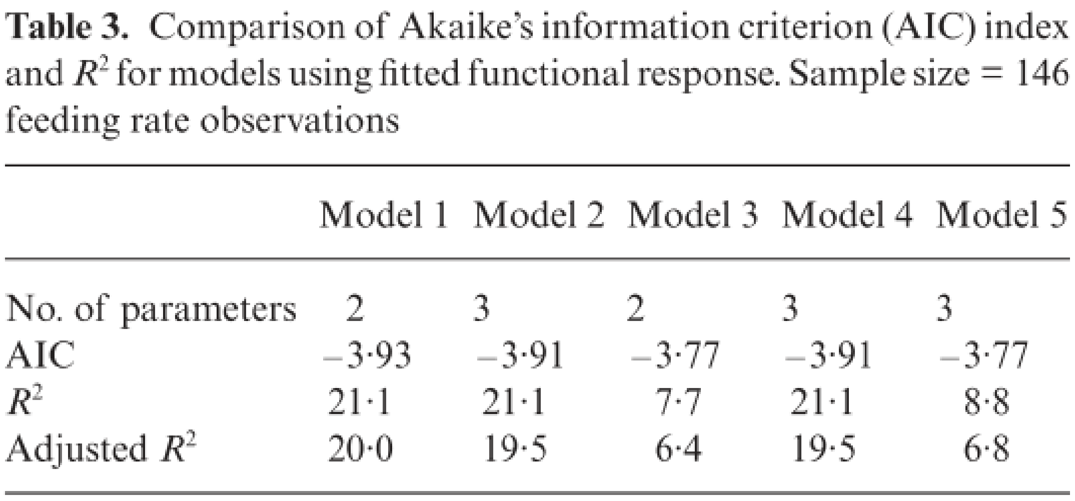


Table 3. Comparison of Akaike's information criterion (AIC) index and R^2 for models using fitted functional response. Sample size = 146 feeding rate observations

	Model 1	Model 2	Model 3	Model 4	Model 5
No. of parameters	2	3	2	3	3
AIC	−3·93	−3·91	−3·77	−3·91	−3·77
R^2	21·1	21·1	7·7	21·1	8·8
Adjusted R^2	20·0	19·5	6·4	19·5	6·8

Figure 6.2: alt

Note that models with a vigilance term could not be fit to the data directly due to lack of convergence.

6.2.1 Calculating the R squared

The fit of the models in the paper is presented below.

R does not calculate R squared for non linear models by default. The reason is that statisticians do not accept this as a valid measure. However ecologists are used to seeing R squared values when models are fit. You can calculate them easily enough from a non linear model. First you need the sum of squares that is not explained by the fitted model. This is simply the variance multiplied by n-1.

```
nullss<-(length(d$rate)-1)*var(d$rate)
nullss
```

```
## [1] 3.544056
```

We get the residual sum of squares (i.e. the unexplained variability) when we print the model

```
HDmod
```

```
## Nonlinear regression model
##   model: rate ~ a * density/(1 + a * density * H)
##    data: d
##        a        H
## 0.003966 1.841890
##  residual sum-of-squares: 2.795
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 3.807e-06
```

So, R squared is just one minus the ratio of the two.

```
Rsq<-1-2.795/nullss
Rsq*100
```

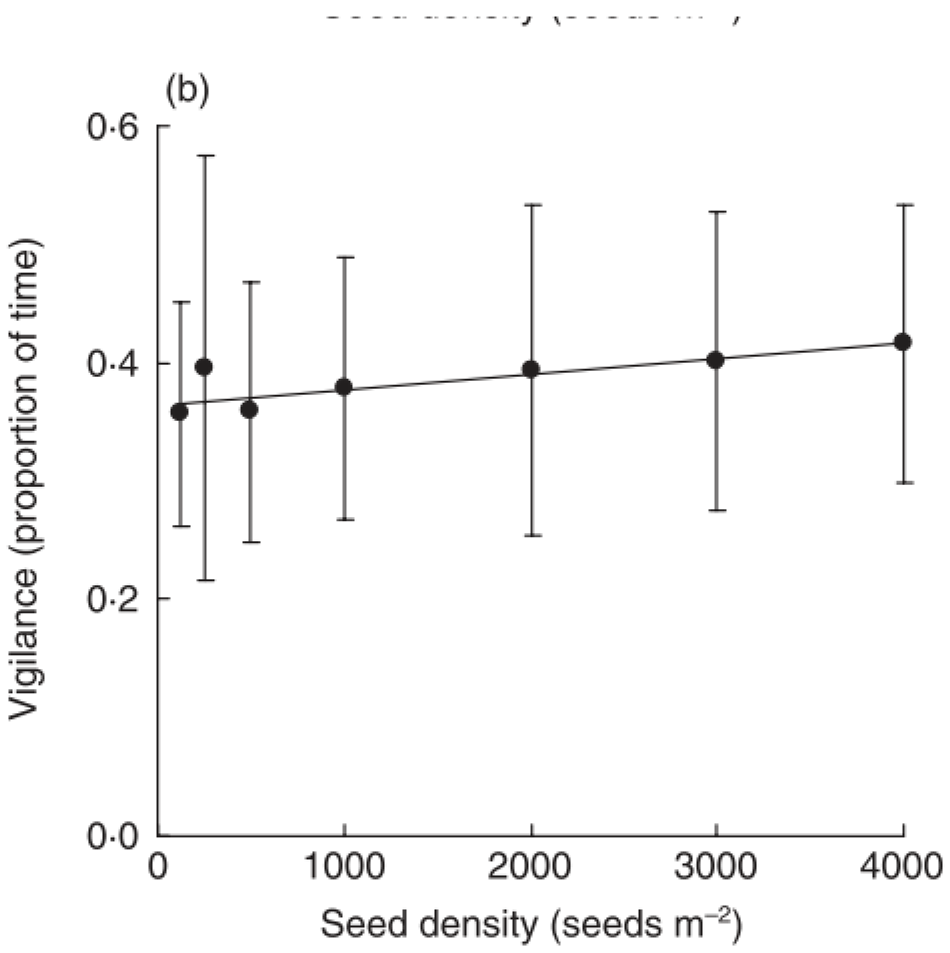


Figure 6.3: alt

```
## [1] 21.13555
```

Which agrees with the value given in the paper.

6.2.2 Including the vigilance term

A very interesting aspect of this article is that the terms of the model were in fact measured independently.

Vigilance, which is defined as the proportion of the time spent being vigilant, rather than feeding, is included in most of the models which are presented in the paper as alternatives to the classic disk equation. Most of these are rather complex conditional models, but the simplest is model 2.

$$R = \frac{(1-v)aD}{1+aDH}$$

The vigilance parameter for model2 cannot be estimated from the data. However it was also measured independently.

Notice that vigilance is fundamentally independent of seed density within this range of densities. The measured value (approximately 0.4) can be included and the model refit with this value included.

```r
Vigmod<-nls(rate~(1-0.4)*a*density/(1+a*density*H),data = d,start = list(a =0.001,H=2))
confint(Vigmod)
```

```
## Waiting for profiling to be done...
```

```
##          2.5%      97.5%
## a 0.004321811 0.01115823
## H 1.028097415 1.18618719
```

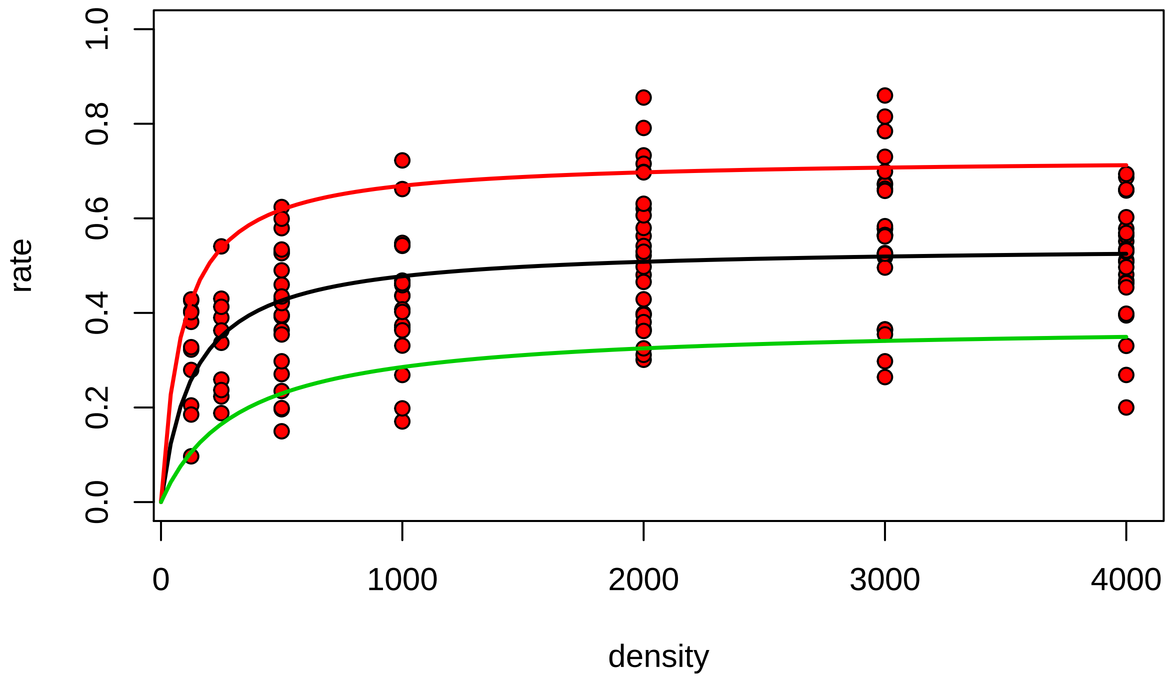
When the measured vigilance is included in the model as an offset term the handling time is reduced by about a half. This makes sense. We may assume that most of the measured vigilance is in fact reducing the rate of feeding as the seed density is so high that search time is very small. However note that we could not possibly have fitted this model without some prior knowledge of the offset value.

6.3 Quantile regression

Another way to look at the issue is to use an alternative approach to fitting statistical models. The traditional approach to fitting a model to data assumes that we are always interested in the centre of any pattern. The error term was classically assumed to be uninteresting random noise around an underlying signal. However in many situations this noise is actually part of the phenomenon we are studying. It may sometimes be attributed to *process error*, in other words variability in the process we are actually measuring, rather than error in our measurements. This may possibly be occurring here. If handling time in fact sets an upper limit on feeding rate, and if we can assume that feeding rate has been measured fairly accurately, then the upper part of the data cloud should be estimating true handling time. Birds that are feeding more slowly than this may be doing something else. For example, they may be spending time being vigilant. So there is possibly some additional information in the data that has not been used in the published paper.

We can use quantile regression to fit a non linear model around the top 10% of the data and the bottom 10%.

```r
library(quantreg)
QuantMod90<-nlrq(rate~a*density/(1+a*density*H),data = d,start = list(a =0.001,H=2),tau=0.9)
QuantMod10<-nlrq(rate~a*density/(1+a*density*H),data = d,start = list(a =0.001,H=2),tau=0.1)
QuantPreds90 <- predict(QuantMod90, newdata = newdata)
QuantPreds10 <- predict(QuantMod10, newdata = newdata)
plot(rate~density,data=d,pch=21,bg=2,ylim=c(0,1))
lines(newdata$density,HDpreds,lwd=2, col=1)
lines(newdata$density,QuantPreds90,lwd=2,col=2)
lines(newdata$density,QuantPreds10,lwd=2,col=3)
```



If handling time limits the maximum rate at which seed can be consumed, then the estimate based on the upper 10% of the data should be closer to the true handling time. So if we look at the summaries of these models we should be able to get a handle on vigilance without the prior knowledge.

```
summary(QuantMod90)
```

```
##
## Call: nlrq(formula = rate ~ a * density/(1 + a * density * H), data = d,
##     start = list(a = 0.001, H = 2), tau = 0.9, control = list(
##         maxiter = 100, k = 2, InitialStepSize = 1, big = 1e+20,
##         eps = 1e-07, beta = 0.97), trace = FALSE)
##
## tau: [1] 0.9
##
## Coefficients:
##   Value    Std. Error t value  Pr(>|t|)
## a  0.00821  0.00132    6.24178  0.00000
## H  1.37329  0.04821   28.48585  0.00000
```

```
summary(QuantMod10)
```

```
##
## Call: nlrq(formula = rate ~ a * density/(1 + a * density * H), data = d,
##     start = list(a = 0.001, H = 2), tau = 0.1, control = list(
##         maxiter = 100, k = 2, InitialStepSize = 1, big = 1e+20,
##         eps = 1e-07, beta = 0.97), trace = FALSE)
##
## tau: [1] 0.1
##
## Coefficients:
##   Value   Std. Error t value Pr(>|t|)
## a 0.00117 0.00051    2.29229 0.02334
## H 2.64950 0.29627    8.94298 0.00000
```

The upper limit (pure handling time) is 1.37 (se= 0.047)

The lower estimate that may include time spent being vigilant is estimated using quantile regression as 2.65 (se= 0.31)

Vigilance thus could, arguably, be estimated as the difference between the upper and lower estimates of handling time divided by the upper value. As the uncertainty around these values has been provided by the quantile regression in the form of a standard error we can use a montecarlo prodedure to find confidence intervals by simulating from the distributions and finding the percentiles of the result.

```
quantile((rnorm(10000,2.65,0.31)-rnorm(10000,1.37,0.047))/rnorm(10000,2.65,0.31),c(0.025,0.5,0.975))
```

```
##      2.5%       50%     97.5%
## 0.2401205 0.4807025 0.7730838
```

This is within the range of the measured value.

There are some interesting elements here that could be discussed in the context of the explanation of the study methods and results.

6.4 Summary

Many relationships in ecology do not form strait lines. If we only have data, and no underlying theory, we can fit a model to the underlying shape using traditional methods such as polynomials or more contemporary models such as splines and other forms of local weighted regression. However these models cannot be extrapolated beyond the data range.

A very different approach involves finding a model "from first principles". Often models are constrained to pass through the origin and by some fundamental limit (asymptote). Model fitting then involves finding a justifiable form for the curve lying between these two points. In some situations data can be used to "mediate" between competing models. In other situations the best we can achieve is to find justifiable estimates, with confidence intervals, for the parameters of a non linear model.

6.5 Exercise

A researcher is interested in establishing a predicitive equation to estimate the heights of trees based on measurements of their diameter at breast height. Your task is to try to find a defensible model for three different data sets.

```
pines1<-read.csv("https://tinyurl.com/aqm-data/pinus_allometry1.csv")
```

```
oaks<-read.csv("https://tinyurl.com/aqm-data/quercus_allometry.csv")
```

```
pines2<-read.csv("https://tinyurl.com/aqm-data/pinus_allometry2.csv")
```

Chapter 7

Analysis of covariance, nested data and mixed effects

7.1 Introduction

We've seen that regression is often not the best available technique to use for bivariate analysis.The books written by Zuur and associates show some of the big challenges involved in analysing real life ecological data. The idealised assumptions of linear regression are very rarely met in full.

Zuur writes **Always apply the simplest statistical technique on your data, but ensure it is applied correctly!** And here is a crucial problem. In ecology, the data are seldom modelled adequately by linear regression models. If they are, you are lucky. If you apply a linear regression model on your data, then you are implicitly assuming a whole series of assumptions, and once the results are obtained, you need to verify all of them. What should we do if we violate all the assumptions? The answer is simple: reject the model. But what do we do if we only violate one of the assumptions? And how much can we violate the assumptions before we are in trouble?

This sounds frightening. If model assumptions are rarely met, how can we ever trust them? Do we need the help of an expert statistician such as Alain Zuur to analyse all ecological data? Perhaps all we can use are simple tests. But do we really know that their assumptions hold?

The best advice to a student aiming to analyse data for an MSc dissertation is simply to always make the best possible attempt at diagnosing issues with the chosen analysis. Be aware of all the assumptions used and be critical at all times. But don't despair if some assumptions are not met. Always point out the problems when discussing the analysis. In many cases the issues will in fact turn out to be unimportant. The key message held within most data sets can often be extracted, even if models have minor violations of assumptions. In other cases there may be more serious problems. Sometimes a more advanced analysis than was orginally planned may be necessary. If you have spotted a problem and understand its implications, it could be easier than you think to build a slightly more complex model. You will not be the first person to have come across the issue. The biggest mistakes usually involve failing to account for lack of independence in the data. This chapter runs through an example of this in practice. The data initially seem simple enough.

An important issue arises when the same relationship between two variables is repeated multiple times in a data set. In the past you may have handled this situation by subsetting the data various times and repeating the analysis for each subset. However it is possible to build more oomprehensive models that look at a population of relationships.

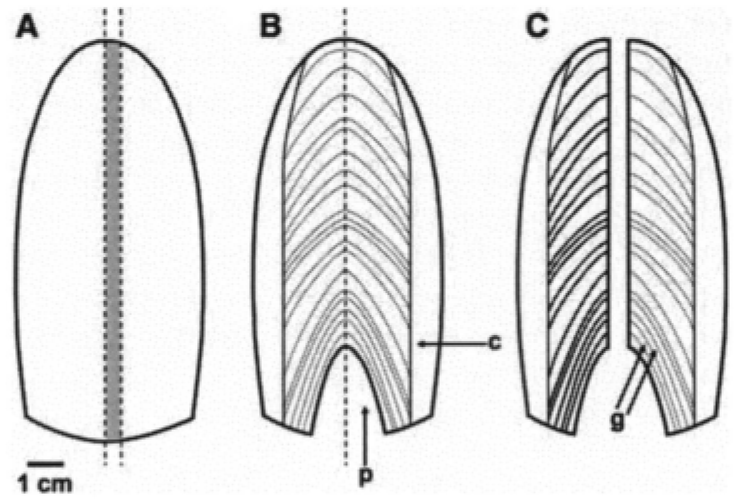


Fig. 1 a–c Diagram illustrating how teeth were cut and sampled. **a** Whole tooth showing where a thin (1–3 mm) slice was cut from the centre, along the bucco-lingual plane. **b** Thin slice cut in half to yield two symmetrical portions [cement (*c*), pulp cavity (*p*)]. **c** Two half-portions. The left one was etched and used for age estimation and the right one decalcified and sampled to obtain one sample per growth layer group (*g*)

Figure 7.1:

7.2 Whale teeth and isotope ratios

In Zuur et al. (2009) a rather complex analysis involving generalised additive models and correlated residuals is developed and justified as a means of ensuring that all model assumptions are met. The results of the analysis are also presented in a paper that shows how the statistical analysis was used to answer an interesting scientific question. It is well worth reading this paper before analysing the data.

The researchers aimed to piece together information regarding feeding behaviour and migration of Sperm whales, based on analysis of the composition of their teeth. Specifically they looked at carbon and nitrogen isotope ratios. N isotope ratios can be indicative of trophic position in marine communities. Trophic level is expected to be higher for larger individuals of the same species. Sperm whales are approximately 4m long at birth, with males growing to 18 m.The relative trophic position of the different stages in the life of several male sperm whales can be investigated through the use of N stable isotope ratios. It was expected that the trophic position would increase with age, as the animals become larger.

Whale teeth have growth rings. To obtain data the researchers analysed isotope ratios from the bands of individual teeth.

The first task in the analysis could be simply to establish that the expectation of increased N isotope ratio with age is supported by the data. We will concentrate on that before looking at more complex issues.

7.2.1 Moby's tooth

Let's first look at a single tooth that was obtained from a famous whale called Moby that was washed up on the shores of the Firth of Forth in 1997 after many rescue attempts. Moby was 15.2 metres in length and weighed 38.5 tons.

Figure 7.2:

Moby's bones went on display in the Royal Scottish Museum, Chamber's street soon after death. The original display failed to attract the public due to the appalling smell.

Moby's skull has since been de-oderised and is now redisplayed. It has even been associated with a Turner prize.

Moby is clearly a star turn. But, how does his life history compare to that of other Sperm Whales? This is the topic we will analyse in this class.

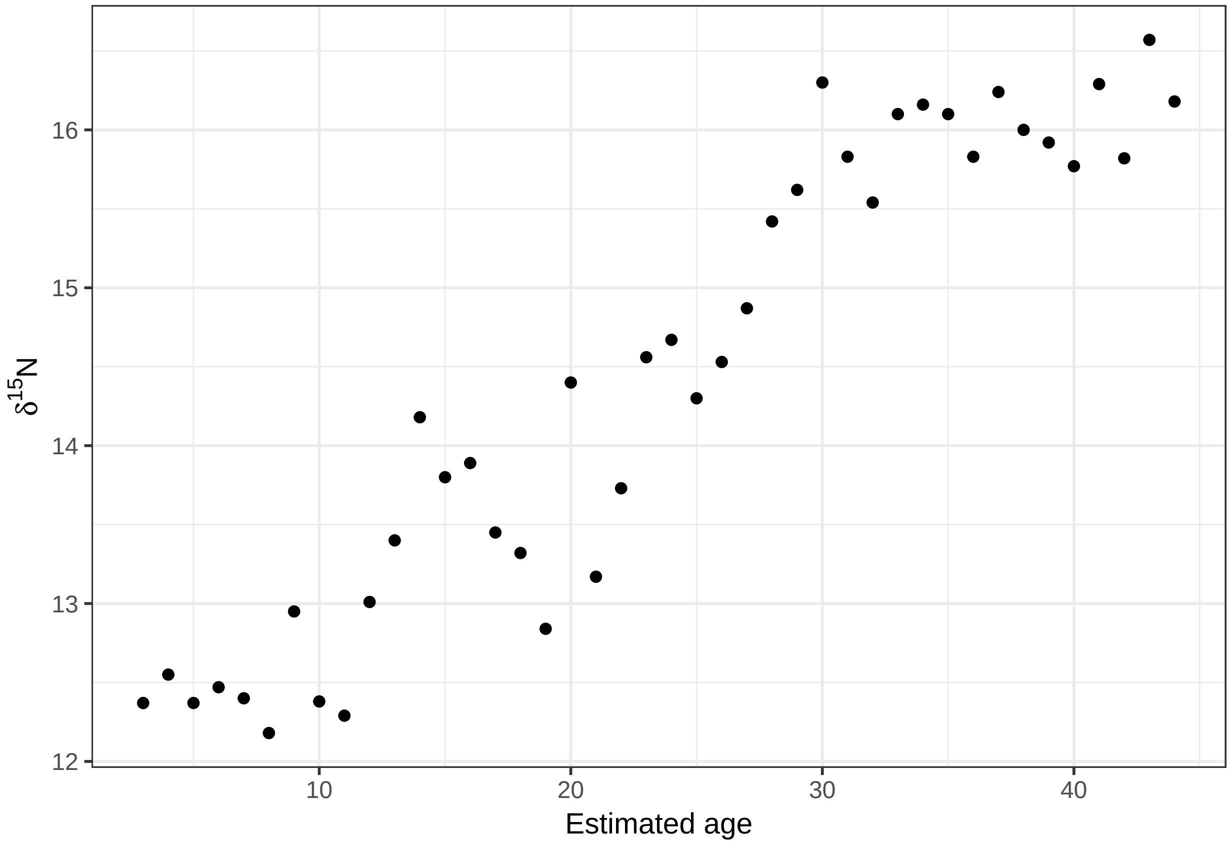
```r
Whales<-read.csv("https://tinyurl.com/aqm-data/whaleteeth.csv")
Moby<-subset(Whales,Whale=="Moby")
```

7.3 Plotting the data

We should first plot the data to look at the pattern. Notice the way a more complex label is set up for the y axis. There are many tricks such as this in R that are rather hard to discover. If you need to use Greek symbols on the axes of your own figures try adapting this code.

```r
ylabel <-expression(paste(delta^{15}, "N"))
xlabel<-"Estimated age"


library(ggplot2)
theme_set(theme_bw())
g0<-ggplot(data=Moby,aes(x=Age,y=X15N))
g1<-g0+geom_point() +xlab(xlabel) +ylab(ylabel)
g1
```

Figure 7.3:



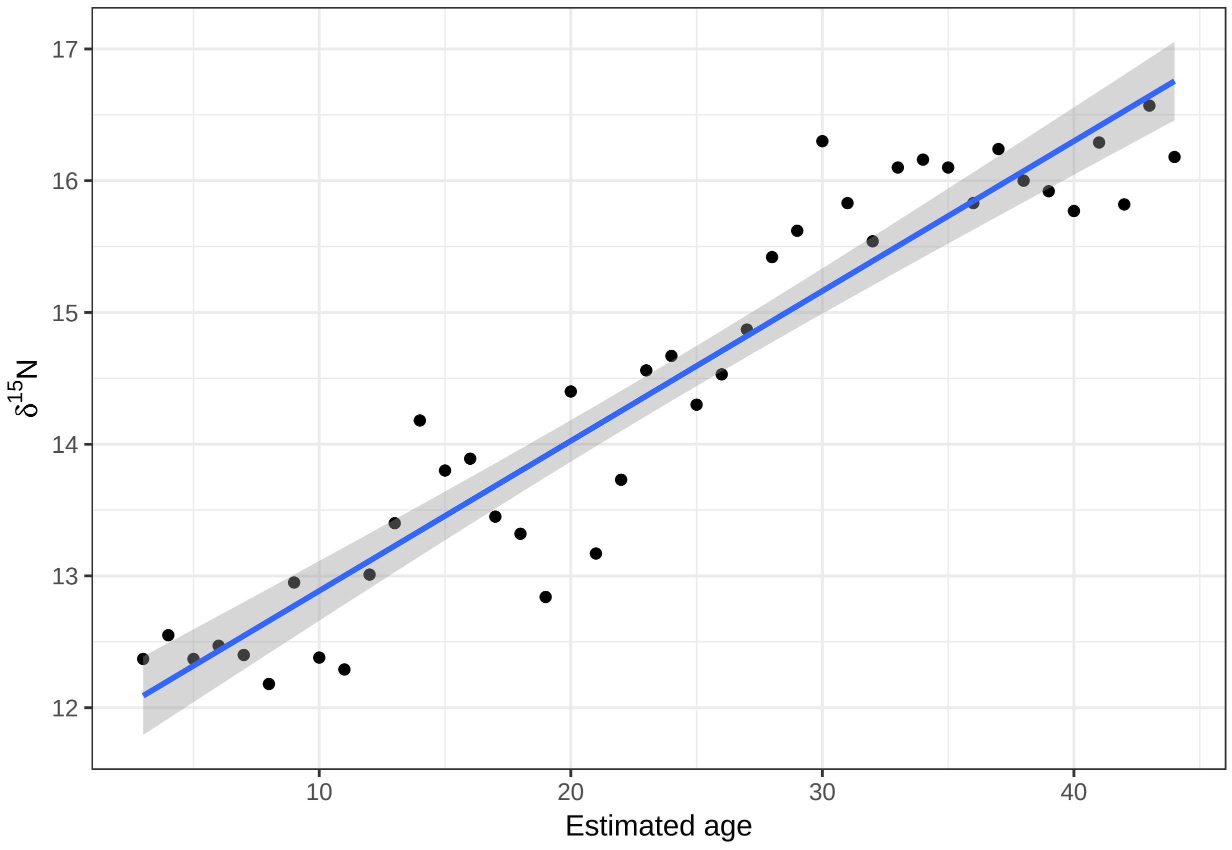
7.4 Fitting a regression

OK, so this all looks quite simple at this stage. The data points lie more or less along a straight line. So we could fit a regression.

```
mod<-lm(data=Moby,X15N~Age)
summary(mod)
```

```
##
## Call:
## lm(formula = X15N ~ Age, data = Moby)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.07102 -0.28706  0.04346  0.33820  1.13724
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.748940   0.163559   71.83   <2e-16 ***
## Age          0.113794   0.006186   18.40   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4859 on 40 degrees of freedom
## Multiple R-squared:  0.8943, Adjusted R-squared:  0.8917
## F-statistic: 338.4 on 1 and 40 DF,  p-value: < 2.2e-16
```

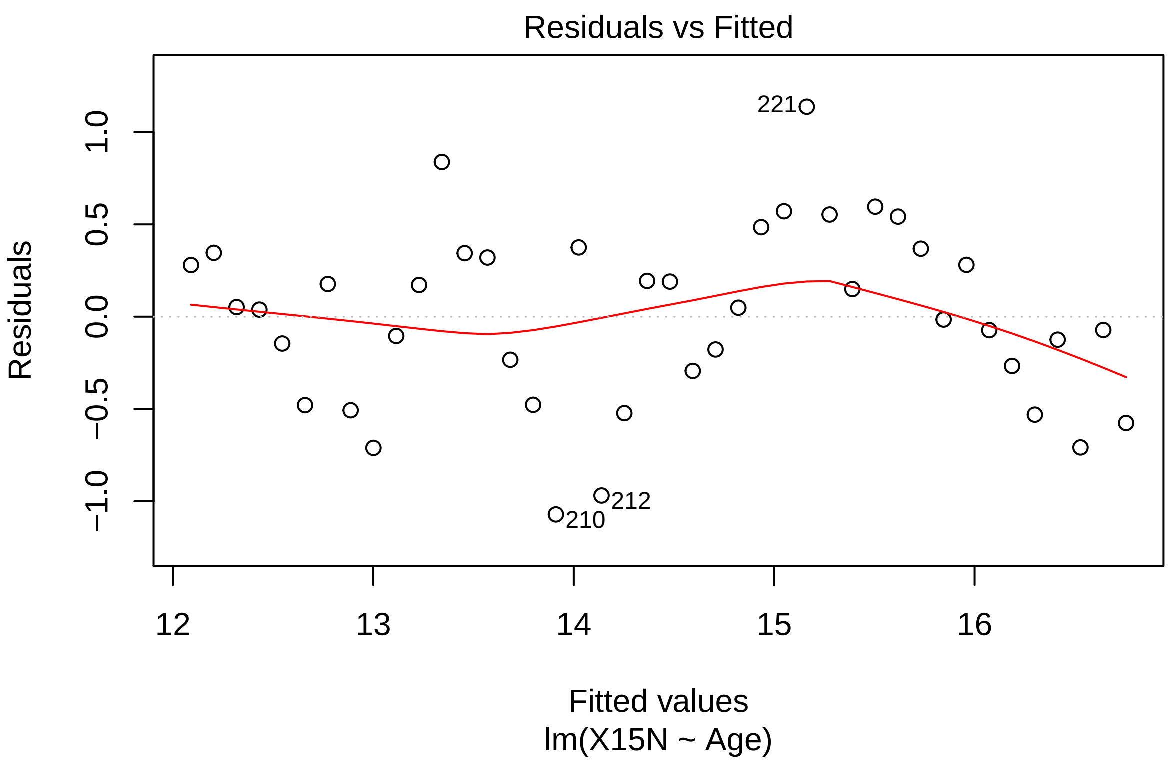
```
g1+stat_smooth(method="lm")+labs(y = ylabel,x=xlabel)
```



7.4.1 Diagnostics

Diagnostics show that most of the assumptions of a regression appear to be met. However an issue shows up when we look at the first diagnostic plot.

```
plot(mod,which=1)
```



There is a clear pattern in the residuals.

7.4.2 Testing for serial correlation

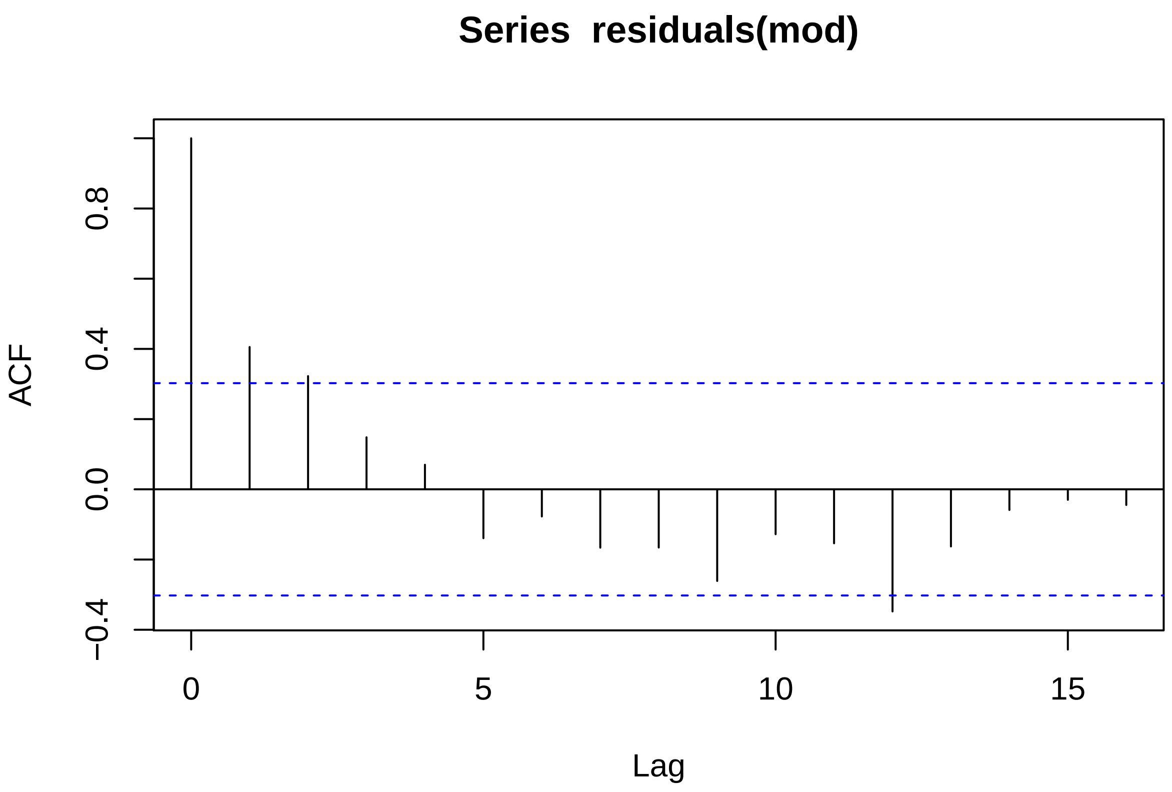
In fact the residuals are serially correlated. This can be checked using a Durbin Watson test.

```
library(lmtest)
dwtest(Moby$X15N~Moby$Age)
```

```
##
##  Durbin-Watson test
##
## data:  Moby$X15N ~ Moby$Age
## DW = 1.1458, p-value = 0.0009539
## alternative hypothesis: true autocorrelation is greater than 0
```

We can confirm this using the acf function.

```
acf(residuals(mod))
```



If any of the lines apart from the first extends above or below the blue line the data are significantly serially correlated. In other words the last value can be used to partially predict the next value. If one residual is negative, the next is also likely to be negative and vice versa. The correlation is significant at lag 1 and lag 2.

7.5 Interpreting the results

Does the serial correlation matter? Well in a purely statistical sense it could matter quite a lot. Zuur et al take the effect into account through the use of a rather cunning statistical trick using an ARIMA model. However most MSc students would be unlikely to be aware of this rather advanced method. In this context it is not really worth the effort. The only real difference between a model that includes autocorrelation and one that does not involves the size of the p-value and the width of the confidence bands. Providing that a straight line is a reasonable description of the underlying pattern (and we will come on to that later), the model is not changed.

There is something much more important to consider however. We have spotted the serial correlation (lack of independence) between years that occurs in the case of Moby. However, you may have realised that in reality all the data we have just analysed are totally lacking in independence anyway! They have been taken from a single whale. So, in one sense we have been conducting statistics with a sample size of one.

In effect, the relationship that we have established applies only to Moby. It would also be really nice to have data from some other teeth from Moby to check even this. We certainly cannot generalise it to other whales. We may have found out some interesting information along the way. The researchers were interested in understanding the trophic ecology of mature sperm whales in some detail. If they are right in assuming that high levels of delta N are associated with feeding on larger prey then there may be some indication that Moby spent several years in a row feeding on prey that may have been larger than expected for his age, followed by a switch to rather smaller prey. This is quite a speculative interpretation, but it may tell us something.

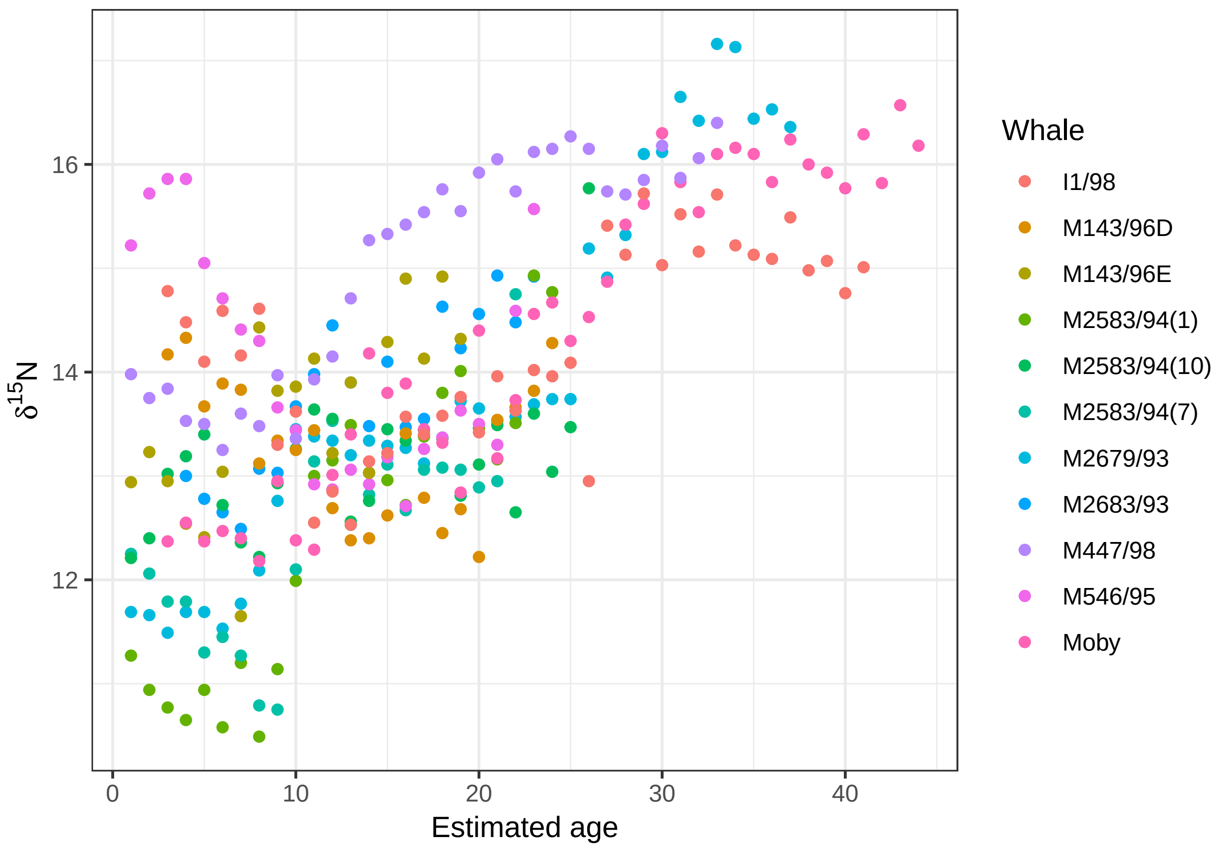
7.6 Exercise

Now try implementing an appropriate method to look at alternatives to a straight line to capture the **empirical** relationship between age and delta N measurements in Moby's tooth. Comment on your findings.

7.7 Finding a general pattern

We actually have data for fifteen whales. So we could compare the pattern shown by Moby with the population of whale teeth. So let's plot out the data from all fifteen whales at once.

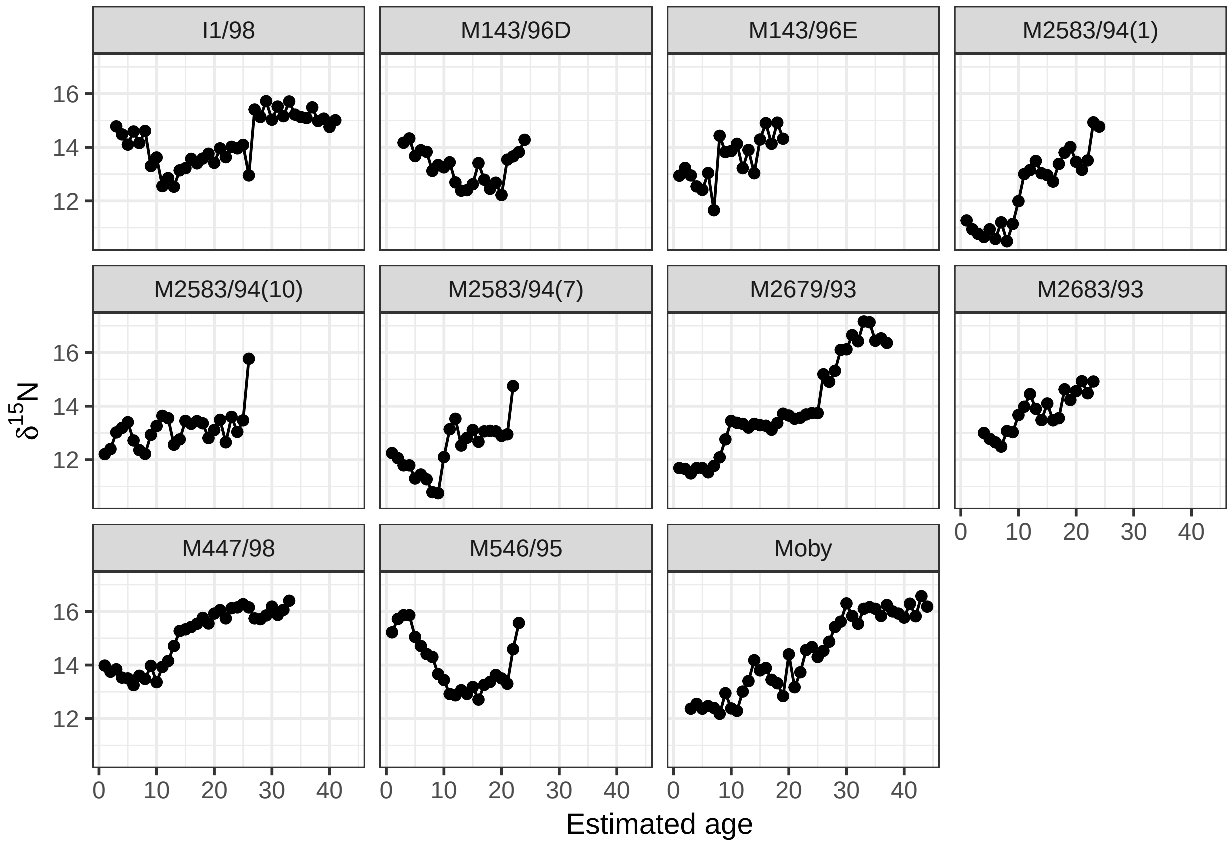
```
g0<-ggplot(data=Whales,aes(x=Age,y=X15N,col=Whale))
g1<-g0+geom_point() + xlab(xlabel) +ylab(ylabel)
g1
```



There is still a clear pattern, but with much more scatter this time. This is to be expected, as each whale has its own life history that contributes variability.

We can look at this by plotting out the data for each individual whale using a facet wrap to split the data.

```
g0<-ggplot(data=Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()+geom_line()+ labs(y = ylabel,x=xlabel)
g1+facet_wrap("Whale")
```



Each whale seems to have its own pattern. Most are approximately linear, like Moby, but some show a very different pattern

For the moment we will ignore this and continue to fit a simple regression model to all the data. Note **THIS IS THE WRONG WAY TO WORK WITH THE DATA!**

```r
mod1<-lm(X15N~Age,data=Whales)
summary(mod1)
```

```
##
## Call:
## lm(formula = X15N ~ Age, data = Whales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5041 -0.7013 -0.1141  0.6209  3.3444
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.228411   0.108598   112.6   <2e-16 ***
## Age          0.095715   0.005631    17.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9953 on 305 degrees of freedom
## Multiple R-squared:  0.4865, Adjusted R-squared:  0.4848
## F-statistic: 288.9 on 1 and 305 DF,  p-value: < 2.2e-16
```

```
confint(mod1)
```

```
##                   2.5 %     97.5 %
## (Intercept) 12.01471479 12.4421063
## Age          0.08463424  0.1067958
```

The problem with this model is that we have far too many degrees of freedom. The model has assumed that each data point is independent, which clearly is not true.

7.8 Analysis of covariance

The data that we have available to predict N15 levels consists of two variables. One is Age and the other is the identity of the individual whale from which the data was obtained. This is a categorical variable. If we continue to put to one side the issue of independence of the data points within the regression for each whale we can analyse the data using a technique known as analysis of covariance.

Analysis of covariance takes into account the differences between each group of observations. It is used to answer the following questions.

- Does the level of a categorical factor affect the response variable?
- Does the slope of a regression line differ between levels of the categorical variable?

In our case the first question implies that each whale may have a different mean value for $\delta^{15}N$. If you read the paper carefully you will see that the researchers were aware that this is a rather trivial question. They looked at more subtle differences related to time of birth and life history. However we will ignore this for the moment too and carry out the analysis in order to illustrate the point.

As mentioned previously, R has a very consistent syntax that makes model fitting very easy. All we need to do to add in an effect for each whale to the model is to write lm(X15N~Age+Whale). The line of R code preceding this sets the contrasts for the model. This determines the output from the summary of the model. In this case we want to contrast the results for other whales with Moby (number 11 in the list).

```
contrasts(Whales$Whale)<-contr.treatment(levels(Whales$Whale),base=11)
mod2<-lm(X15N~Age+Whale,data=Whales)
anova(mod2)
```

```
## Analysis of Variance Table
##
## Response: X15N
##            Df Sum Sq Mean Sq F value    Pr(>F)
## Age         1 286.21 286.215 431.039 < 2.2e-16 ***
## Whale      10 106.27  10.627  16.004 < 2.2e-16 ***
## Residuals 295 195.88   0.664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = X15N ~ Age + Whale, data = Whales)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.70699 -0.51434 -0.07552  0.47030  2.64139
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       12.256788   0.174661  70.175  < 2e-16 ***
## Age                0.092183   0.005159  17.869  < 2e-16 ***
## WhaleI1/98        -0.010461   0.181371  -0.058  0.95404    
## WhaleM143/96D     -0.229444   0.220575  -1.040  0.29909    
## WhaleM143/96E      0.385063   0.235813   1.633  0.10355    
## WhaleM2583/94(1)  -0.936579   0.216095  -4.334 2.01e-05 ***
## WhaleM2583/94(10) -0.357032   0.209785  -1.702  0.08983 .  
## WhaleM2583/94(7)  -0.901441   0.223214  -4.038 6.87e-05 ***
## WhaleM2679/93     -0.059622   0.185189  -0.322  0.74772    
## WhaleM2683/93      0.267238   0.227314   1.176  0.24069    
## WhaleM447/98       1.210339   0.192499   6.288 1.16e-09 ***
## WhaleM546/95       0.685273   0.219544   3.121  0.00198 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.8149 on 295 degrees of freedom
## Multiple R-squared:  0.6671, Adjusted R-squared:  0.6547 
## F-statistic: 53.73 on 11 and 295 DF,  p-value: < 2.2e-16
```

The anova result is very clear. There is a significant additive effect that can be attributed to the individual. So we can see that whales do differ in their baseline N15 levels. In order to interpret the model summary you need to be aware that the intercept represents the value at age zero for Moby, which was set to be the "control" or baseline for the contrasts. The other whale values are compared to the baseline. Thus if a coefficient for an individual whale is not significant it means that the intercept for that whale was not different to that of Moby.

So, the answer to the first question, just as we would expect, is a definite yes. Six of the ten whales that we compared with Moby have significantly different levels of N15 after allowing for age.

How do we answer the second question? This involves adding an interaction term to the model. The interaction implies that each whale could have it's own intercept AND it's own slope. If the interaction term is significant then we have a very complex model that cannot be easily generalised.

```
mod3<-lm(X15N~Age*Whale,data=Whales)
anova(mod3)
```

```
## Analysis of Variance Table
## 
## Response: X15N
##            Df  Sum Sq Mean Sq F value    Pr(>F)    
## Age         1 286.215 286.215 734.509 < 2.2e-16 ***
## Whale      10 106.266  10.627  27.271 < 2.2e-16 ***
## Age:Whale  10  84.828   8.483  21.769 < 2.2e-16 ***
## Residuals 285 111.055   0.390                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction term is highly significant. This is not suprising when we consider the lattice plot, which showed a series of very different patterns for each whale.

If we summarise the model now we will obtain even more complex output which shows how the coefficients depend on the identity of each whale.

```
summary(mod3)
```

```
##
```

```
## Call:
## lm(formula = X15N ~ Age * Whale, data = Whales)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58547 -0.40951 -0.01552  0.37746  2.41087
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            11.748940   0.210125  55.914  < 2e-16 ***
## Age                     0.113794   0.007947  14.320  < 2e-16 ***
## WhaleI1/98              1.464565   0.303845   4.820 2.34e-06 ***
## WhaleM143/96D           1.905086   0.376915   5.054 7.73e-07 ***
## WhaleM143/96E           0.720710   0.364725   1.976 0.049115 *
## WhaleM2583/94(1)       -1.517527   0.336649  -4.508 9.58e-06 ***
## WhaleM2583/94(10)       0.747337   0.328174   2.277 0.023512 *
## WhaleM2583/94(7)       -0.559979   0.346500  -1.616 0.107179
## WhaleM2679/93          -0.750516   0.296705  -2.529 0.011962 *
## WhaleM2683/93           0.509286   0.412830   1.234 0.218351
## WhaleM447/98            1.557803   0.305940   5.092 6.45e-07 ***
## WhaleM546/95            3.269282   0.341382   9.577  < 2e-16 ***
## Age:WhaleI1/98         -0.065573   0.011918  -5.502 8.36e-08 ***
## Age:WhaleM143/96D      -0.142106   0.022432  -6.335 9.23e-10 ***
## Age:WhaleM143/96E      -0.004390   0.027327  -0.161 0.872476
## Age:WhaleM2583/94(1)    0.065493   0.020050   3.267 0.001222 **
## Age:WhaleM2583/94(10) -0.065797   0.018155  -3.624 0.000343 ***
## Age:WhaleM2583/94(7)  -0.007142   0.022432  -0.318 0.750421
## Age:WhaleM2679/93       0.041481   0.012471   3.326 0.000996 ***
## Age:WhaleM2683/93      -0.001922   0.025478  -0.075 0.939929
## Age:WhaleM447/98       -0.012176   0.013906  -0.876 0.381993
## Age:WhaleM546/95       -0.194624   0.021171  -9.193  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6242 on 285 degrees of freedom
## Multiple R-squared:  0.8112, Adjusted R-squared:  0.7973
## F-statistic: 58.33 on 21 and 285 DF,  p-value: < 2.2e-16
```

Now we can see that six of the whales do not just have a different baseline level of $\delta^{15}N$ to Moby, the change over time when modelled as a linear response is also different for these individuals.

7.8.1 More about interactions

The model above could also be written in R longhand as:

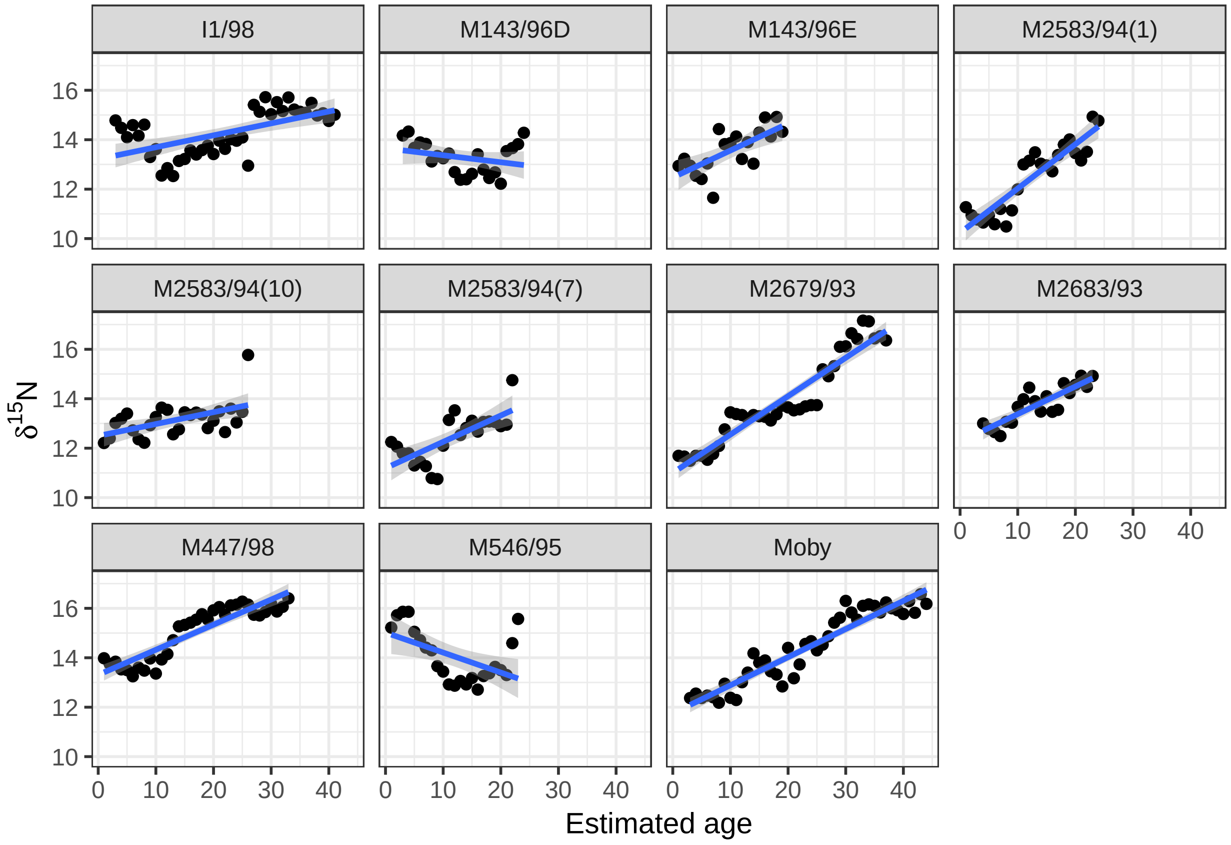
```
mod3<-lm(X15N~Age+Whale+Age:Whale,data=Whales)
```

The {*} symbol is simply short hand for this. It does not imply multiplication. The interaction component is Age:Whale. Don't confuse statistical interactions with ecological interactions such so those that occur in a food web. A statistical interaction implies that we need to know additional information in order to predict something. For example if an experiment showed an interaction between nitrogen fertilizer level and soil type we could not be sure that adding nitrogen always enhances growth in the same way. Perhaps it has no effect at all when applied to heavy clay soils, although it increases growth generally. Finding interactions suggest complexity. This can often be interesting in itself.

7.8.2 Plotting the model

The model with interactions can be plotted using ggplots as a series of individual regression lines for each whale. Notice that the slope differs between each of the panels.

```
g0<-ggplot(data=Whales,aes(x=Age,y=X15N,group=Whale))
g1<-g0+geom_point()+ labs(y = ylabel,x=xlabel) +geom_smooth(method="lm")
g1+facet_wrap("Whale")
```



Chapter 8

Introducing mixed effects modelling

Analysis of covariance is often used when there are a small number of factor levels. If we had two or three whales either subsetting for each whale or analysis of covariance would be the only appropriate methods. There would not be enough replication to think about the data as having been drawn from a larger population of whales that it could represent. Each case would be treated as being distinct. However, as we get more data analysis of covariance becomes clumsy and less appropriate as a technique. We can see that in this example. A **mixed effects** model treats each level of a factor as a **random effect**. This is a powerful technique when we have enough data as we are effectively trying to draw out some underlying population level pattern, which we may want to use when comparing the response of each individual. Mixed effects modelling also has some advantages for data sets with missing data or unbalanced numbers of observations for each group.

The example was adapted from Zuur's book on mixed effects models. Mixed effects are also known as hierarchical models. It is important to be aware of the potential usefulness of such models and the difference between fixed and random effects. The last model we fitted used a regression equation for each whale in turn. This approach assumes the identity of the whale is a fixed factor. We need to know each whale's name in order to use the model! If we want to generalise for all whales we would be much better to think of the whales as compromising a random sample from all the whales that we might have found.

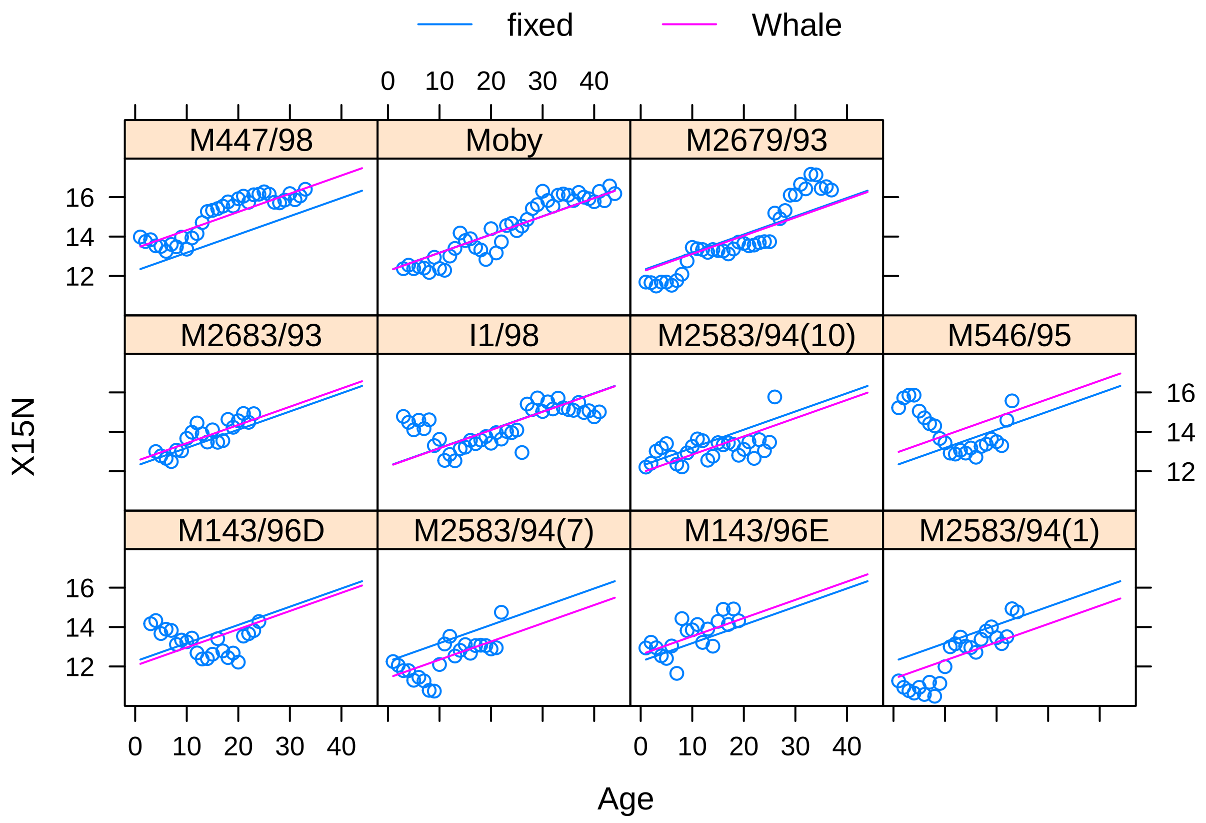
The following R code uses a mixed effects model approach to fit a separate regression line for each whale. To cut a long story short, and to simplify the overall idea, you may think of the result as being rather similar to taking the mean slope and intercept from all the individual regressions and finding confidence intervals for the pooled result. This was a traditional way of analysing data of this type and it is still used by some researchers. It is not "wrong" in essence, but modern methods are both easier to apply and more formally correct due to the manner in which unbalanced samples are handled. Look at the plot we produced for a regression fitted to each whale to get an idea of what the model is based on. Imagine averaging the slopes and intercepts shown in this figure in order to get an intuitive idea of the basis for the model.

8.1 Using the nlme package

The nlme package name stands for "non linear mixed effects". It has some poweful functions for fitting non linear models with random effects. More recently the lme4 package has become more popular for linear mixed effects as the algorithm used is prefered by mathematical statisticians. In almost all applied situations the results from using either package for linear mixed effects models are identical. Plotting the models is easy using nlme as the package has a grouping function for the data and a series of default plots that use the lattice package. Lattice plots have now largely given way to ggplots when formally presenting results as ggplots are more flexible and nicer looking. However the default in nlme is easy to use and easy to interpret.

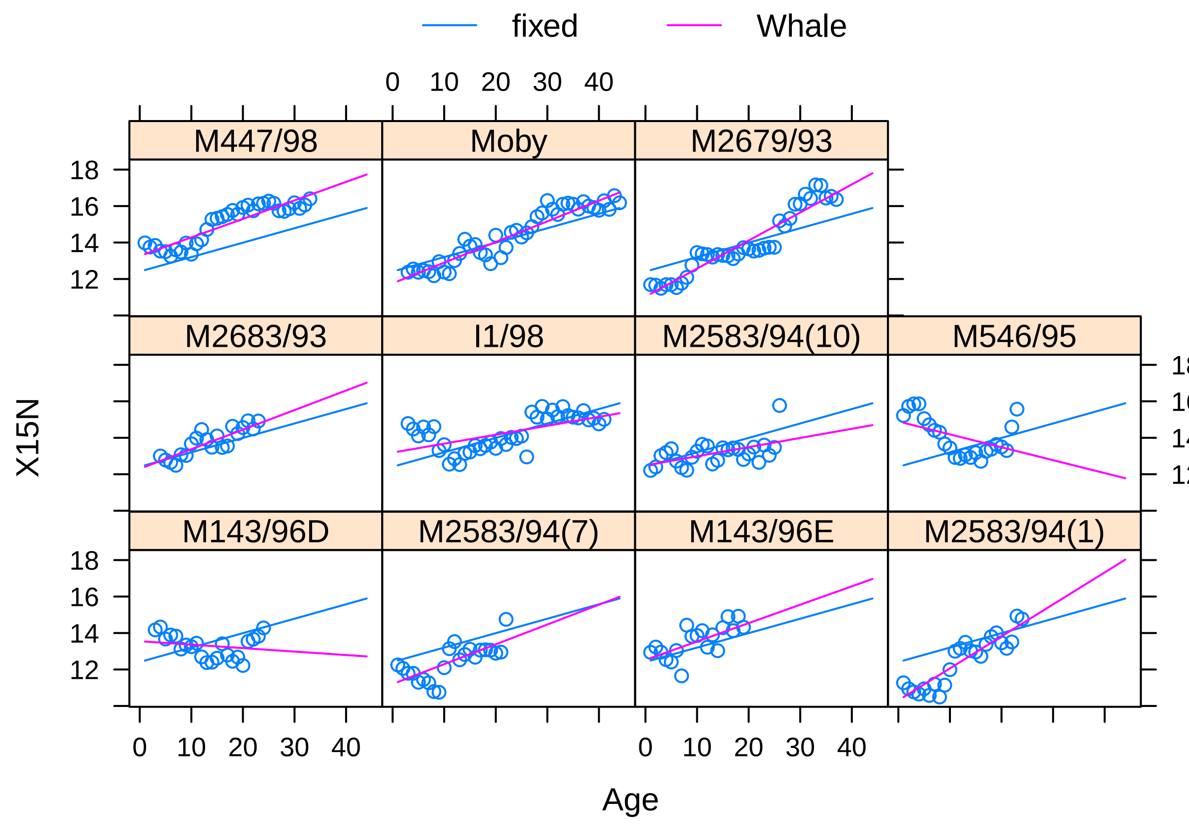
8.1.1 Intercept only model

```
library(nlme)
w<-groupedData(X15N~Age|Whale,Whales)
intercept.mod<-lme(X15N~Age,random=~1|Whale,data=w)
print(plot(augPred(intercept.mod,level=c(0,1))))
```



8.1.2 Random slopes model

```
slope.mod<-lme(fixed=X15N~Age,random=X15N~Age|Whale,data=w)
print(plot(augPred(slope.mod,level=c(0,1))))
```



The fixed effect is the underlying trend of increasing $\delta^{15}N$ levels with age. The random effect is the variation in slope and intercept between whales. We can test which model is prefered on statistical criteria by comparing them.

```
anova(intercept.mod,slope.mod)
```

```
##               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## intercept.mod     1  4 792.3965 807.2777 -392.1982
## slope.mod         2  6 669.3825 691.7044 -328.6913 1 vs 2 127.0139  <.0001
```

So the slope model provides a significantly better fit to the data, just as we found when taking whale as a fixed effect.

8.1.2.1 Confidence intervals for the effects

```
intervals(slope.mod)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                   lower        est.      upper
## (Intercept) 11.60509740 12.41013148 13.2151656
## Age          0.03376861  0.07914293  0.1245172
## attr(,"label")
## [1] "Fixed effects:"
##
##  Random Effects:
##   Level: Whale
##                            lower        est.      upper
## sd((Intercept))       0.84344384  1.33071563  2.0994926
## sd(Age)               0.04669552  0.07443052  0.1186389
## cor((Intercept),Age) -0.97238683 -0.90157346 -0.6787395
##
##  Within-group standard error:
##     lower      est.     upper
## 0.5750660 0.6242716 0.6776874
```

By taking into account the lack of independence we now get a much wider confidence intervals for the coefficients than we did when we very naively fitted a regression to the pooled data without taking into account the identity of the whale from which the data was obtained. However we now have a generalisable model based on a sample of size of n=15, rather than the model with far too many degrees of freedom. The mixed model is better than the model used in the analysis of covariance, as this could only be used for prediction if you provide the identity of the particular whale that you are interested in.

Notice that the parameters estimated for the random effects are standard deviations. If you can imagine actually fitting all 15 separate models, writing down the interecept and the slope 15 times and finding the mean and standard deviation of each you have approximated the operation carried out by the mixed effects modelling. The standard deviations are approximations, so there are also confidence intervals for these provided by the full fitted model.

There is a whole lot more to mixed effects modelling, but this result is quite simple to understand in this context. It provides a robust solution to the issue of finding confidence intervals for the mean slopes and intercepts when analysis of covariance shows that seperate regression models are needed for each level of a factor, but we do want to treat the levels of the factor as random draws from a population as there are too many factor levels to be concerned with comparisons between each one. Mixed effects also handles unbalanced samples, as if one whale has fewer observations the overall model gives the data less weight.

Finding an underlying trend in N15 levels is only part of the story told in the paper. An issue that has been set well to one side in this still simplified example is whether a linear regression is the best approach to finding an underlying pattern. In fact Zuur et al used GAMs in order to capture the non linear relationship that some individuals seem to have.

8.2 Using the package lme4

The alternative package to nlme is lme4. This uses a more sophisticated approach to actually fiting the model that is considered more reliable mathematically by some statisticians. In most real life situations there is little difference between the results from each package and they can usually be used interchangeably.

A slightly unusual aspect of lme4 is that it does not provide p-values for the test of signficance of the effects by default. This was a deliberate decision made by the author of the package (Douglas Bates). The argument is that the degrees of freedom are ill-defined for many technical reasons. As p-values are only really useful if a statistic such as t is close to the boundary of significance the argument goes that there is no point in calculating them unless they are exact. A t value of 3 is going to be significant whatever and a t value of 1 will not be. However some people demand p-values and they can be calculated quite easily. The package Lmertest adds these calculations to the output of lme4.

The syntax used to fit the models is slightly different but follows the same general pattern as nlme. A random effect for the intercept alone is written as (1|Whale).

```r
library(lmerTest)
intercept.mod2<-lmer(X15N~Age+(1|Whale),data=w)
intercept.mod2
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: X15N ~ Age + (1 | Whale)
##    Data: w
## REML criterion at convergence: 784.3965
## Random effects:
##  Groups   Name        Std.Dev.
##  Whale    (Intercept) 0.6139
##  Residual             0.8149
## Number of obs: 307, groups:  Whale, 11
## Fixed Effects:
## (Intercept)          Age
##    12.25874      0.09245
```

```
anova(intercept.mod2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##     Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## Age 216.11  216.11     1 301.39  325.46 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The slope model can be fit in the same way.

```
slope.mod2<-lmer(X15N~Age+(Age|Whale),data=w)
slope.mod2
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: X15N ~ Age + (Age | Whale)
##    Data: w
## REML criterion at convergence: 657.3826
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  Whale    (Intercept) 1.33124
##           Age         0.07446  -0.90
##  Residual             0.62427
## Number of obs: 307, groups:  Whale, 11
## Fixed Effects:
## (Intercept)          Age
##    12.41013      0.07914
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```
anova(slope.mod2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##     Sum Sq Mean Sq NumDF  DenDF F value   Pr(>F)
## Age 4.5882  4.5882     1 9.8319  11.773 0.006585 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

8.2.1 Profile confidence intervals

A more accurate, but much slower, profiling technique is used to calculate confidence intervals in lme4. The result may be slightly different from nlme.

```
# Note that this takes around 30 seconds to run as it is based on simulated profiling
confint.merMod(slope.mod2)
```

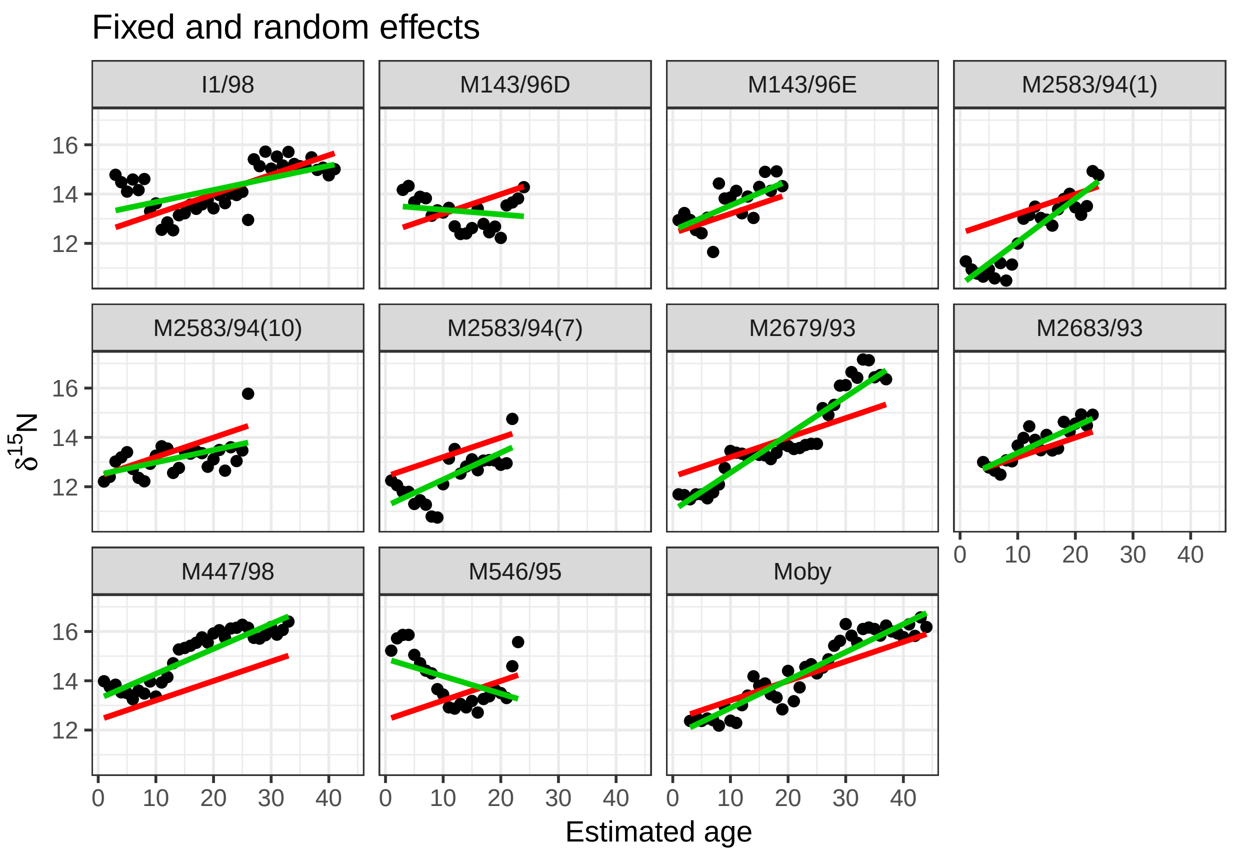
```
##                   2.5 %     97.5 %
## .sig01        0.85660240  2.0833944
## .sig02       -0.97150830 -0.6859207
## .sig03        0.04712678  0.1173118
## .sigma        0.57633298  0.6792690
## (Intercept) 11.57425698 13.2460978
## Age          0.03191633  0.1262261
```

8.3 Ggplots from lme output

The lme4 package does not include the augpred function. However it is easy to reproduce the same sort of figures using ggplots. The trick is to predict the fixed effect alone and the fixed and random effects for each data point then add the results to the figure as lines.

```
Whales$fixed<-predict(slope.mod2,re.form=NA)
Whales$rand<-predict(slope.mod2)
```

```
g0<-ggplot(Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=2,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=rand),colour=3,lwd=1)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed and random effects")
g1+facet_wrap("Whale")
```



8.4 Mixed effect gamm models

You could argue (rightly), after looking at al the data. that straight lines do not provide a good model for the pattern in the data at all. However analyses are not just about finding the best fitting model. The model has to have some underyling purpose and feed into some narrative about the data. A fitted linear trend is often used as an easily communicated baseline for comparison rather than as a perfect model for the data. So it may still be useful to be able to state how much "on avererage" you expect $\delta^{15}N$ levels to increase per year.

It is however possible to fit smoothers using a mixed effects approach. The approach could be "overkill" for some purposes. If you just need to find the overall average change in $\delta^{15}N$ per year it would not be useful. However it could also help show the differences between the pattern for each whale and so fit the overall population level pattern if the trend is not linear.

So .. and do note that this is rather advanced stuff that is not an essential part of the course, I will show how this can be done using the gamm4 package in R.

Fitting the model uses the same syntax, but with a smooth term for the fixed effects. The interepretation of the smoother is more difficult.

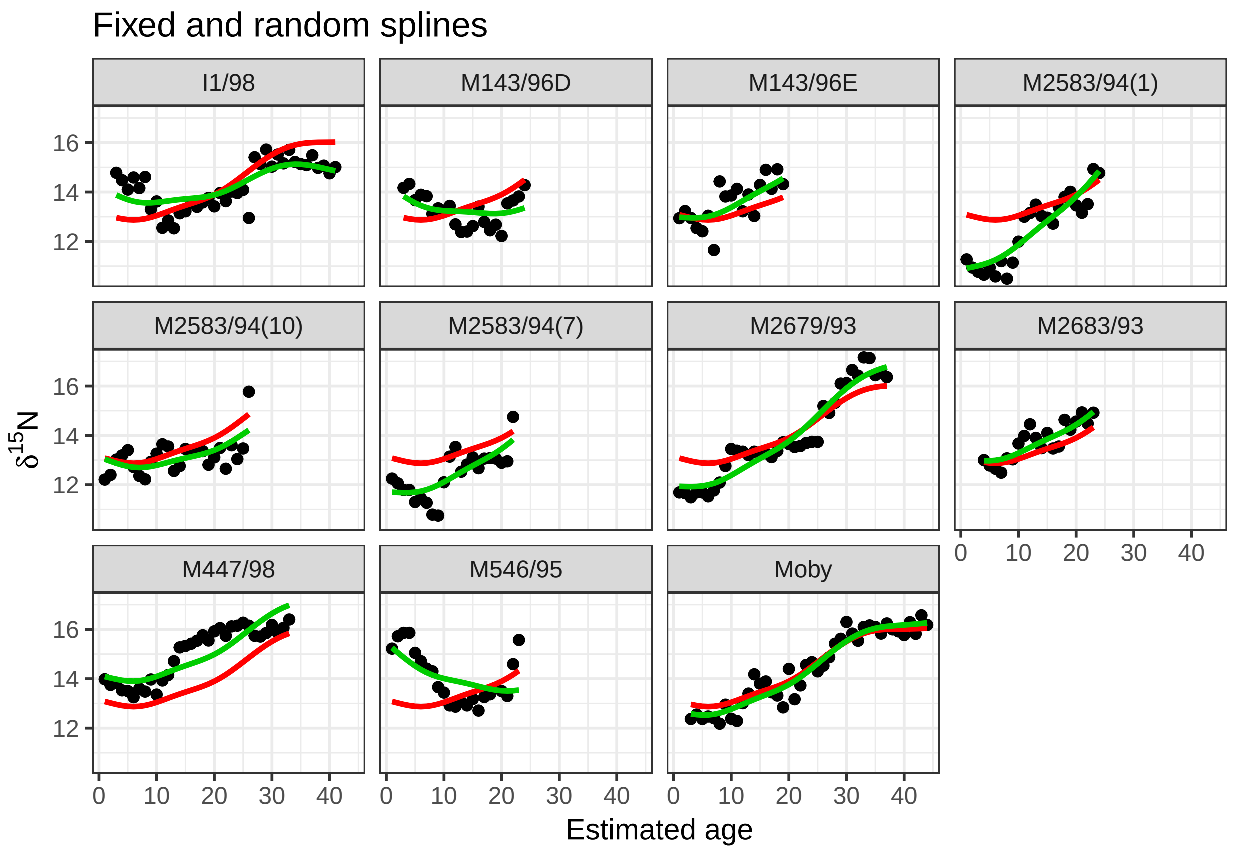
```
library(gamm4)
gamm.mod<-gamm4(X15N~s(Age),data=w,random = ~ (Age|Whale))
gamm.mod
```

```
## $mer
## Linear mixed model fit by REML ['lmerMod']
## REML criterion at convergence: 619.3
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  Whale    (Intercept) 1.35294
##           Age         0.07361  -0.91
##  Xr       s(Age)      2.66779
##  Residual             0.57755
## Number of obs: 307, groups:  Whale, 11; Xr, 8
## Fixed Effects:
## X(Intercept)    Xs(Age)Fx1
##      13.8111       -0.2626
##
## $gam
##
## Family: gaussian
## Link function: identity
##
## Formula:
## X15N ~ s(Age)
##
## Estimated degrees of freedom:
## 5.66  total = 6.66
##
## lmer.REML score: 619.3
```

The only real way to understand any model based on a smoother is to plot it out. The effects can be extracted and added to plots.

```
Whales$fixed<-predict(gamm.mod$gam)
Whales$rand<-predict(gamm.mod$mer)
```

```
g0<-ggplot(Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=2,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=rand),colour=3,lwd=1)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed and random splines")
g1+facet_wrap("Whale")
```

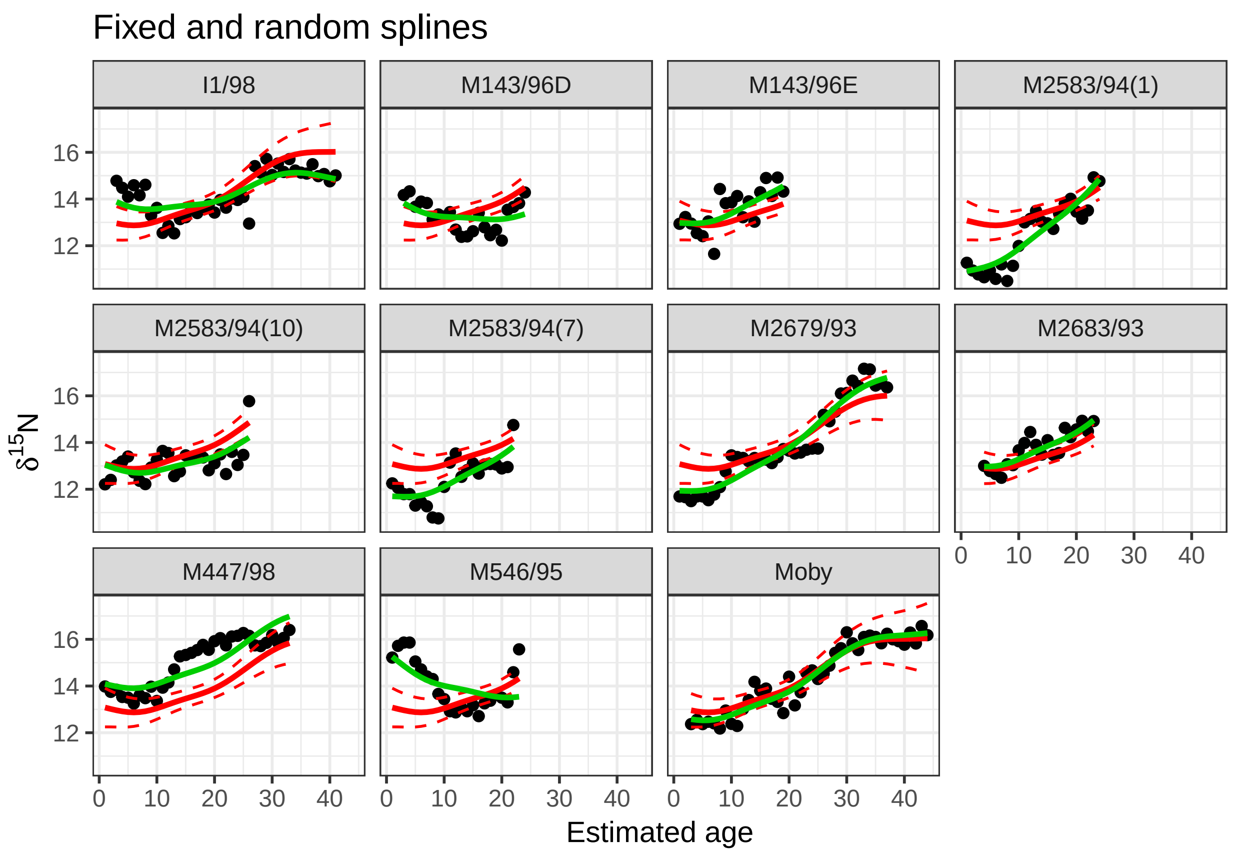


Notice that there is now an overall curve for the fixed effect with separate curves for each whale. The overall fixed effect model is the average of all these curves.

It is possible to go one step futher and add confidence intervals to a fixed effect plot, because the standard error for the fixed effects can also be found

```
Whales$fixedse<-predict(gamm.mod$gam,se=T)$se
```

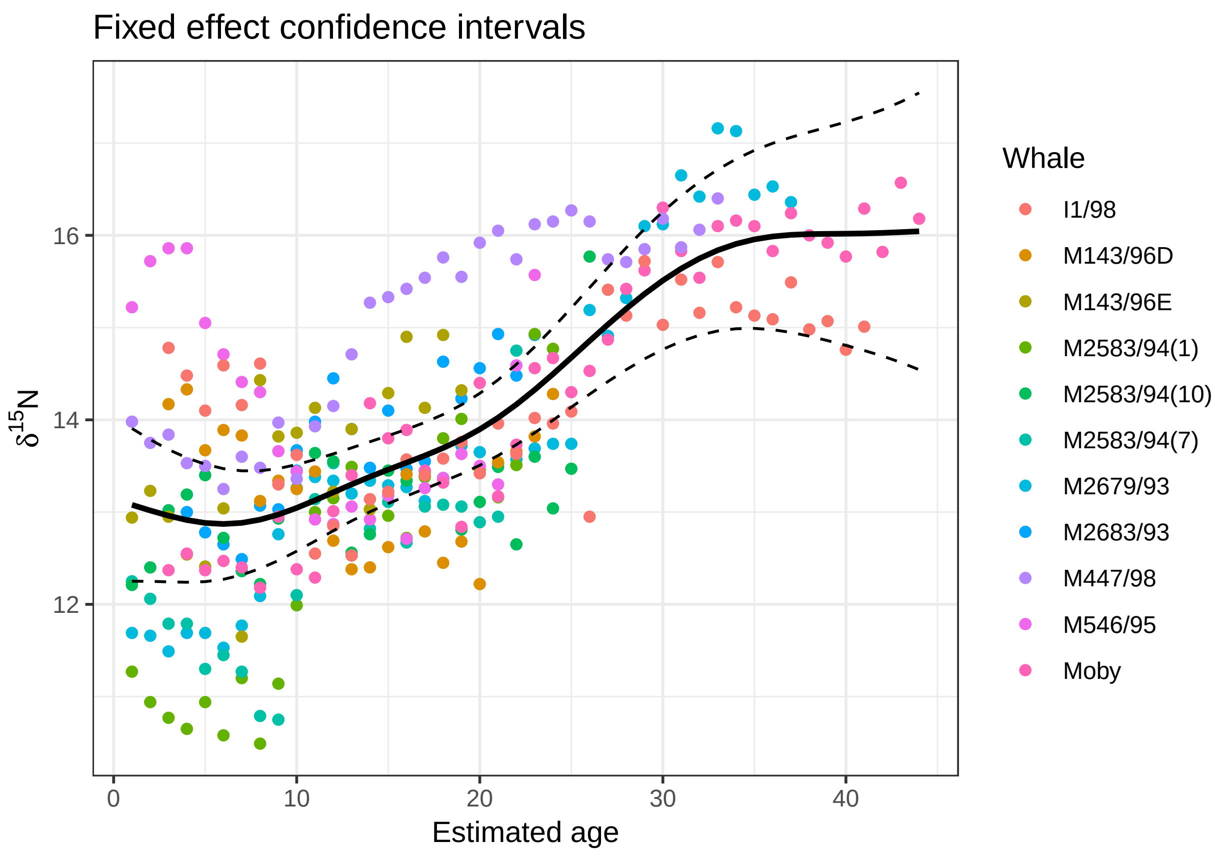
```
g0<-ggplot(Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=2,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=fixed+2*fixedse),colour=2,lty=2)
g1<-g1+geom_line(aes(x=Age,y=fixed-2*fixedse),colour=2,lty=2)
g1<-g1+geom_line(aes(x=Age,y=rand),colour=3,lwd=1)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed and random splines")
g1+facet_wrap("Whale")
```



The real reason for all this in practice would be to compare the overall pattern with that shown by each individual whale. The overall fixed effect as fitted by a spline may be a useful model as a general description of the pattern shown by the population. The normal caveats about the representativiness of the sample apply of course. In this particular case there is also the issue that the whales may have lived at different periods.

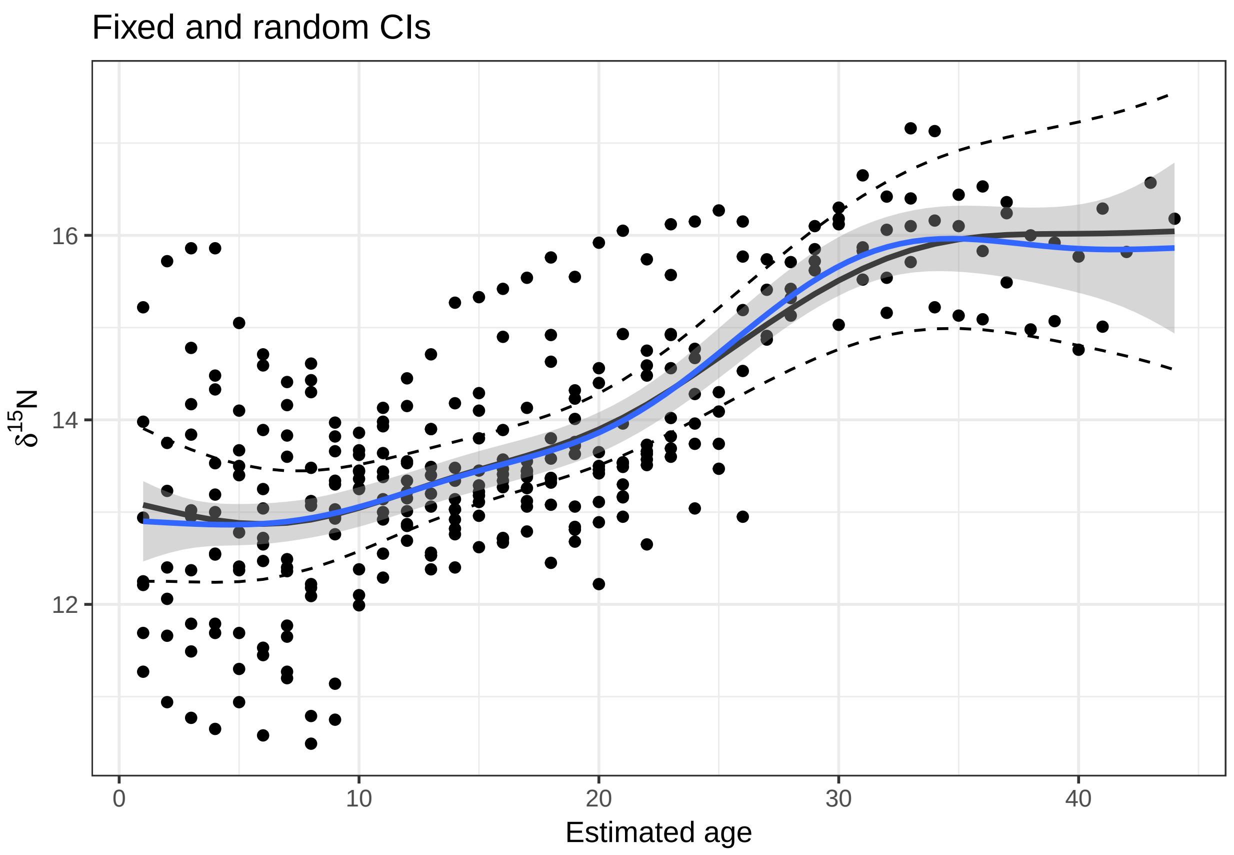
Notice that the confidence intervals for the smoother are now much wider than you would obtain if the random effect of each whale was not included in the model. This is because the sample size is effectively only 15 and the random variability between whales is being properly taken into account. The model potentially generalises to the population from which the sample (n=15) was taken.

```
g0<-ggplot(Whales,aes(x=Age,y=X15N,col=Whale))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=1,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=fixed+2*fixedse),colour=1,lty=2)
g1<-g1+geom_line(aes(x=Age,y=fixed-2*fixedse),colour=1,lty=2)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed effect confidence intervals")
g1
```



Look at the difference between the "naive" smooth that overpredicts by just using all the points as if they were all independent and the much more robust model that is based on the population level fixed effect.

```
g0<-ggplot(Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=1,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=fixed+2*fixedse),colour=1,lty=2)
g1<-g1+geom_line(aes(x=Age,y=fixed-2*fixedse),colour=1,lty=2)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed and random CIs")
g1+geom_smooth(method="gam",formula=y~s(x))
```



You should also notice that the general shape of the response is still very similar. The point of going through all the process of modelling the random effect is just to reveal how much (or how little) confidence we have that the shape of the response can be generalised. The appropriate model for the data takes into account the small sample of whales, even though there are many individual data points. The message seems to be that whales begin to feed on similar sized food items overall once they reach an age of 30 years.

8.5 Summary

This example shows that a lot can be learned about the data by first applying relatively simple (and statistically incorrect) techniques, even though the models that are first tried will later need major modification in order to prevent violations of assumptions. Once all the issues have been spotted, solutions can usually be found, although they may involve using more advanced techniques used in combination. This approach may need help from a more experienced analyst. The problem of lack of independence are very common in many data sets. You can explore the data using simple techniques in the knowledge that an ajustment may be needed before finally presenting results in order to ensure the findings are robust from a statistical perspective.

Notice that constructing the more sophisticated models doesn't necessarily change the underlying message in the data. If we divide models into an informative component, that we are most interested in, and a stochastic component (variability) that is of less scientific interest, then the statistical sophistication is largely needed mainly to produce a better model for the stochastic element. Looking at the data carefully will usually produce insight into the scientific issu. Finding the right model allows us to produce more defensible confidence intervals (or p-values), even though our basic conclusions may not change. This is a common element in statistical modelling. Don't expect complex models to necessarily "tell you more" than simpler models. This may look like a lot of work for little reward, but the fundamental reason for using more sophisticated models is to reduce the number of assumptions that are being violated and thus carry out truly reproducible research. In this case we end up with a generalised pattern based on the data that we are not overly confident about, but may be a useful summary of the general trend at the population level. So while we would not be suprised at all to find whale teeth that follow very different patterns to that shown in Moby's particular case, we would confidently expect that a new sample of fifteen whales would produce a similar pattern to that found using the mixed effects model, providing the sample has been drawn from the same potential population.

The example has also been used to introduce analysis of covariance, a technique that uses both a categorical variable (factor) and a numerical variable as explanatory variables in the model formula. You can now apply the technique to slightly simpler data in the examples below in order to practice.

8.5.1 Exercises

1. Analysis of covariance can be used to look in more detail at the mussels data again. You first have to make sure that R treats site as a factor.

```r
Mussels<-read.csv("https://tinyurl.com/aqm-data/mussels.csv")
Mussels$Site<-as.factor(Mussels$Site)
```

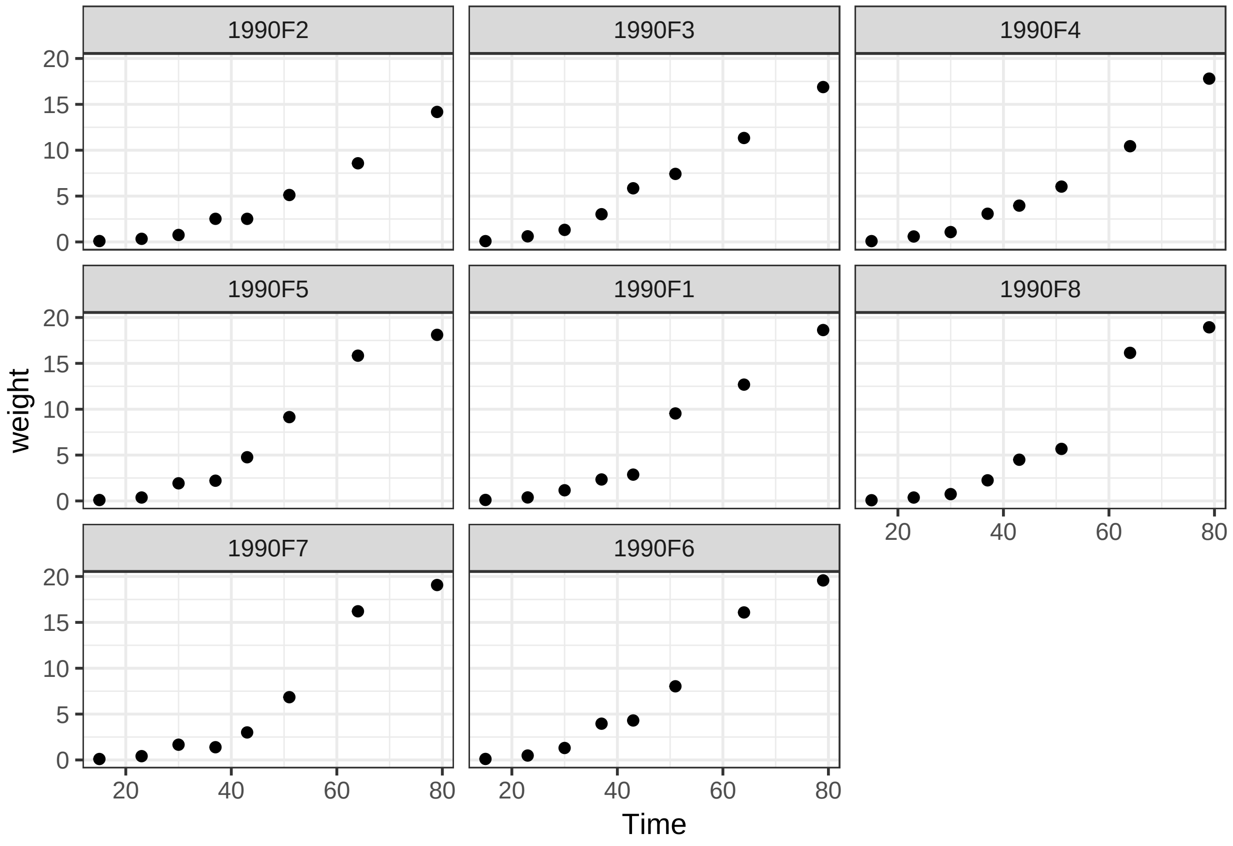
Use linear analysis of covariance.

Is there a difference in the relationship between BTvolume and shell length between sites? How did you reach your conclusion?

2. Soybean growth curves

Try these growth curves. They are best fit using **non-linear** mixed effects, but you could still try using the methods above for this exercise.

```r
data("Soybean")
d<-subset(Soybean,Soybean$Variety=="F")
d<-subset(d,d$Year==1990)
g0<-ggplot(d,aes(x=Time,y=weight))
g1<-g0+geom_point()
g1+facet_wrap("Plot")
```



3. Here is another similar but very difficult data set to practice on (optional). It has also been analysed by Zuur. The measurements are lengths and ash free dry weight (AFD) of 398 wedge clams (Donax hanleyanus) taken from a beach in Argentina during six different months (Ieno, unpublished data). The main issue is the extremely unbalanced nature of the data. Can we do anything useful at all with this tricky observational data set? There are limits that even the most sophisticated statistics cannot remove.

```
Clams<-read.table("https://tinyurl.com/aqm-data/Clams.txt",head=T)
Clams$MONTH<-as.factor(Clams$MONTH)
```

The data frame already contains log transformed versions of the variables. Try first fitting a regression to the untransformed data and then use the transformed version. Which is most suitable for modelling using regression? Now use analysis of covariance to investigate the effect of month on the relationship. Does this work? What would you do if faced with real life data of this sort?

Chapter 9

Design and analysis of experiments part 1

9.1 Introduction

Observational studies are much more common in environmental science and ecology than in other fields. When we analyse observational data we are seeking to gain evidence in support of scientific hypotheses. However this evidence tends to be indirect and open to a range of interpretations. In contrast, a well designed experiment involves a planned manipulation of nature that should allow us to test hypotheses directly. A statistical hypothesis never completely coincides with the scientific hypothesis, however in an experimental setting the two concepts may be closely linked.

Experimental studies are rare in ecology because they tend to be costly to set up, may require large areas and may need to run for many years in order to provide convincing data. So we often look for natural experiments that arise on an *ad hoc* basis to provide data. Sometimes unintentional human interventions such as oil spills, hunting, landscape fragmentation etc provide such systems that we can study. Although these rarely follow conventional experimental design, familiarity with experimental concepts may help in the selection of an appropriate analysis.

Many of the analytical methods and concepts used by experimental ecologists and environmental scientist have been derived from agricultural experiments. There are good reasons for this. The most influential statistician of all time, R A Fisher, was employed for many years at Rothamstead agricultural research centre. During that time he developed the theory and practical application of analysis of variance. The language used to describe experimental designs is based around this work. Although this helps to provide a common language it can sometimes be confusing. Terms such as “split plot” can be used for a design with no real plots, “Repeat measures” can occur when the measures are not actually taken on more than one occasion. Blocks may refer to non spatial entities. The key to understanding experimental design is to realise that the design of the experiment and the design of the statistical model used to analyse it go hand in hand. If you can understand the structure of the data and the model used to analyse it you can either design experiments to produce appropriate data or design surveys that provide observational data that can be analysed using methods designed for experiments.

9.2 Basic concepts of experimental design

9.2.1 1. Replication

The most important element to recognise when designing any form of experiment that will be analysed using statistical methods is the amount of **replication** involved. This can be easy to determine if the experimental unit is well defined. For example if a treatment is applied to an animal then the number of replicates is usually going to be the number of animals used in the experiment. However if the animals share a single environment, such as a cage or enclosure, then that may become the experimental unit thus possibly reducing the number of replicates. It can be more complex when treatment is applied to plots of land due to spatial auto correlation and lack of independence. Ecological experiments have often been criticised for involving so called "pseudoreplication". This occurs when the true number of independent replicates is much lower than the number claimed by the researcher. One cause of this may be that sub samples taken from what should properly be regarded as a single experimental unit are analysed as if they constitute true replicate samples. This may become clearer after we have looked at some examples. The amount of replication needed to establish the significance and size of the effect of some intervention depends to a large extent on the amount of natural variability in the system of study. Some laboratory based experiments can be based on extremely small numbers of replicates. For example, if drugs are administered to a set of genetically identical mice there may be no need for more than three animals in the control and treatment group in order to establish a statsitically significant effect. The reason for this is the response of all the animals without intervention would be very similar and all are likely to respond in identical ways to the treatment. However this experiment would be limited to establishing the effect of the drug on that single genotype. If a set of individuals captured at random from a wild population were used the study would have to be much larger. If the intention was to test the drug on the wild population at large it would have to be even larger in order to account for all the extraneous source of variability in the environment. Similarly if a researcher is interested in the effects of some treatment on plant growth and works with genetically similar plants grown on identical media in a greenhouse, for example using hydroponics or a very uniform potting compost, then the experiment may be need few replicates. However if the plants are grown in fields with varying soil conditions both within and between fields the experimental design will need to take this variability into account. Ecologists and environmental scientists face more challenges than other researcher as a result of the variability in natural systems. At the same time it may be difficult or impossible to find enough naturally ocurring independent replicates to use robust analytical methods.

9.2.2 2. Treatment levels

A second concept is that of **treatment levels**. The most basic type of experiment involves two levels of a treatment. The first level is simply no intervention (control). So, if we wanted to look at the effects of pesticides on pollinator abundance we would design an experiment with replicated units in which no pesticide was applied (control) and others in which the same level was applied. It is easy to see that the two level model could be extended to include more categories, such as high, medium and low levels of application. The classic analyses use fixed, categorical treatment levels. However in an observational context we often obtain continuous measurements. This influences the way we analyse data as we may substitute regression and analysis of covariance for a classic ANOVA. However linear statistical models have an underlying mathematical structure that can be used for either form of data. We can build models with combinations of both types of variables.

9.2.3 3. Randomisation

The third concept to understand is **randomisation**. When designing an experiment the units should be assigned a treatment level at random which should not depend on their current status and ideally should not even be known by the experimenter. So, for example, in drug trials the participants are assigned a treatment at random. In a blind trial they themselves do not know whether they have been given a new drug or a placebo. In a double blind trial neither does the researcher. If random selection is used there should be no criteria used to choose those that receive the treatment. As you can imagine, this can sometimes raise ethical issues, but is a fundamental feature of statistically rigorous experimental design. In many situations involving "natural" experiments there is no randomisation in the application of treatments. For example a forest fire may burn some trees and leave others intact but the selection of trees for treatment by fire is far from random. Parasites may attack hosts that are weakened by some other cause. Often spatial prximity is a key factor in selection for treatment. This leads to spatial autocorrelation and a lack of independence. Once more this makes applying the analytical methods designed for analysing true experiments more challenging when data is derived from observations of the natural system.

9.2.4 4. Interactions

We use the word **interaction** frequently in ecology to refer to effects such as competition, depredation, parasitism and other elements involved in the systems we study. However in statistics the word has a specific meaning that applies to experiments or observational studies involving more than a single factor. An interaction occurs when the effects of the factor are not additive. In other words the level of one factor influence the effect of another. Take for example an experiment in which saplings were grown at two light levels, full sun and 50% shade and two irrigation treatments applied. Let's call them high and low. Plants need both water and light to grow so we might expect the plants grown in full sun with high levels of irrigation to grow faster than those with low levels of irrigation. However what if the plants fail to respond to increased watering at low light levels, or even grow more slowly, perhaps due to waterlogged roots. In this case there would be an interaction as the two effects do not add together. The effect of irrigation is *conditional* on the level of light.

9.2.5 5. Fixed and random effects

The distinction between whether we treat factors as having **fixed** or **random** effects is rather subtle and subjective. Simply put fixed effects are those that we are most interested in while random effects represent natural variability that we may wish to quantify and control for, but which is not directly influential on our scientific hypotheses. When we analyse experimental data using R or any other statistical package we can declare some effects to be fixed and others as random. If for example we carried out the same experiment at five different sites that were selected at random from a large set of possible locations we may treat site as random effect, as we are not interested in the specific effect of any single named site. However if we deliberately selected the sites as representing different conditions that we may be interested in then the effect of site would be fixed.

When we include a variable in a model as a random effect we are only interested in the amount of variability, not the specific source of variability at the level of a single observation. So instead of looking at specfic effects we look at the variance or standard deviation of each random effect. These effects may be at a range of levels. For example we might look at the variability in yield between single plants and between fields. If no treatment has been assigned then in both cases we would only be interested in the variances. We usually fit "mixed effects" models however as we are typically interested in some form of fixed effect in addition to random variation.

It can often be rather difficult to decide which effects should be treated as random, and when it is appropriate to do so. The good news is that in many situations it doesn't really matter. The p-values and confidence intervals for the fixed effects that we are most interested in may not be influenced by the choice at all. The bad news is that in other cases the decision can be very important. Failure to declare a random effect as an error term may lead to type one errors, in other words, declaring a result to be statistically significant when it is not. In other situations, involving nested effects at different levels, the use of a mixed effects model is essential. Again, this may become clearer after looking at examples.

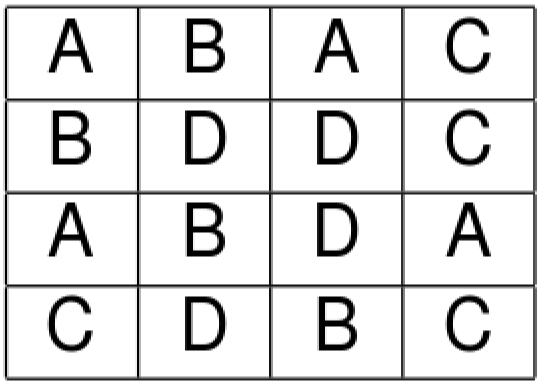


Figure 9.1: alt text

9.3 Types of design

9.3.1 Completely randomised design

This is the simplest form of experiment. In this design, all levels of the treatment or all combinations of treatment levels are assigned to experimental units completely at random. Often one of the treatment levels is considered to be a control (i.e. no intervention). If all the factor levels involve some form of treatment one of them can be considered a reference level or baseline. There must be replication of each treatment level.

The statistical model is a simple one way Anova.

Let's look at a simple experiment of this type. We'll take an agricultural example. A researcher measured soil moisture at a certain depth following irrigation using four different techniques always using the same amount of water. The idea was to find out if there were any differences between them in order to find the most efficient technique in which less moisture was lost through evaporation. Let's first assume that the moisture levels were measured once in each plot. The layout looks like this.

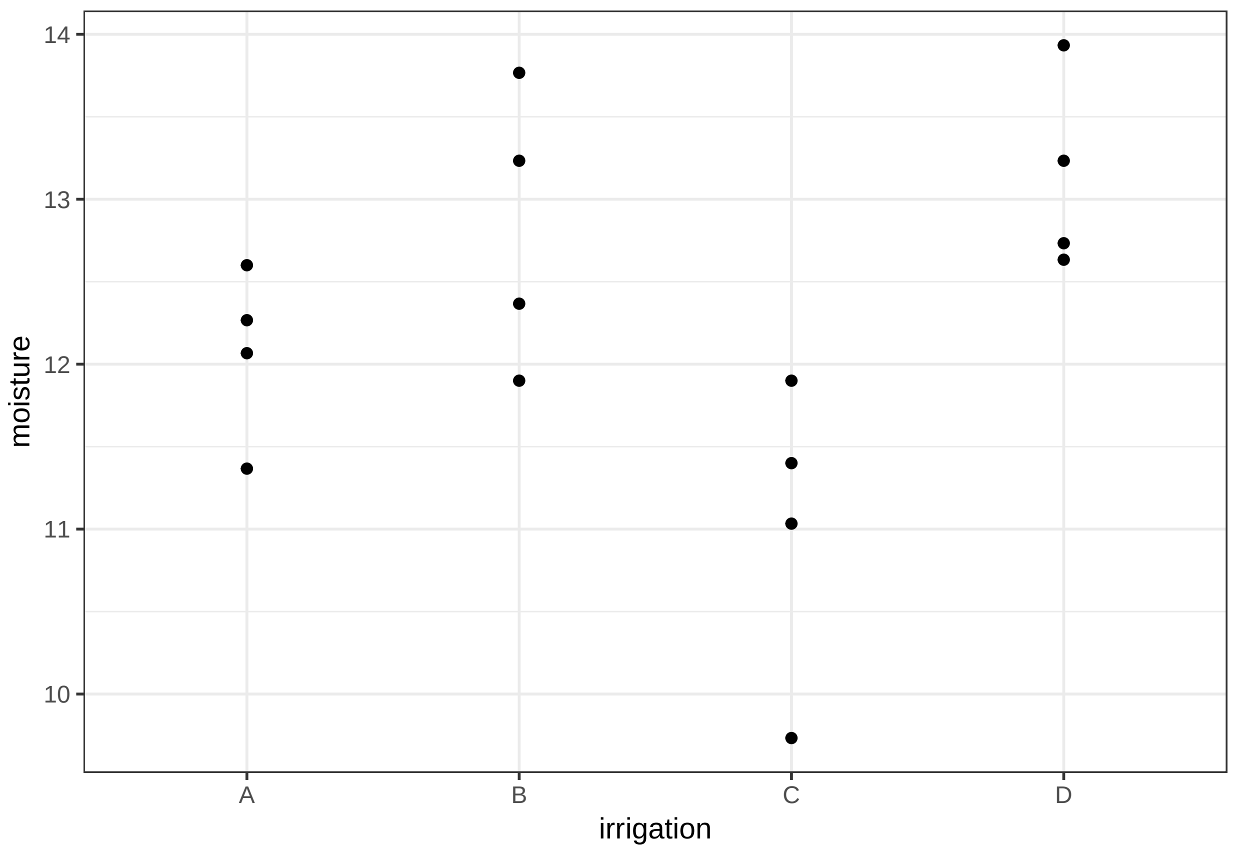
```
library(ggplot2)
library(reshape)
library(multcomp)
```

```
d<-read.csv("https://tinyurl.com/aqm-data/irrigation1.csv")
str(d)
```

```
## 'data.frame':    16 obs. of  3 variables:
##  $ plot      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ irrigation: Factor w/ 4 levels "A","B","C","D": 1 1 1 1 2 2 2 2 3 3 ...
##  $ moisture  : num  12.3 12.6 11.4 12.1 12.4 ...
```

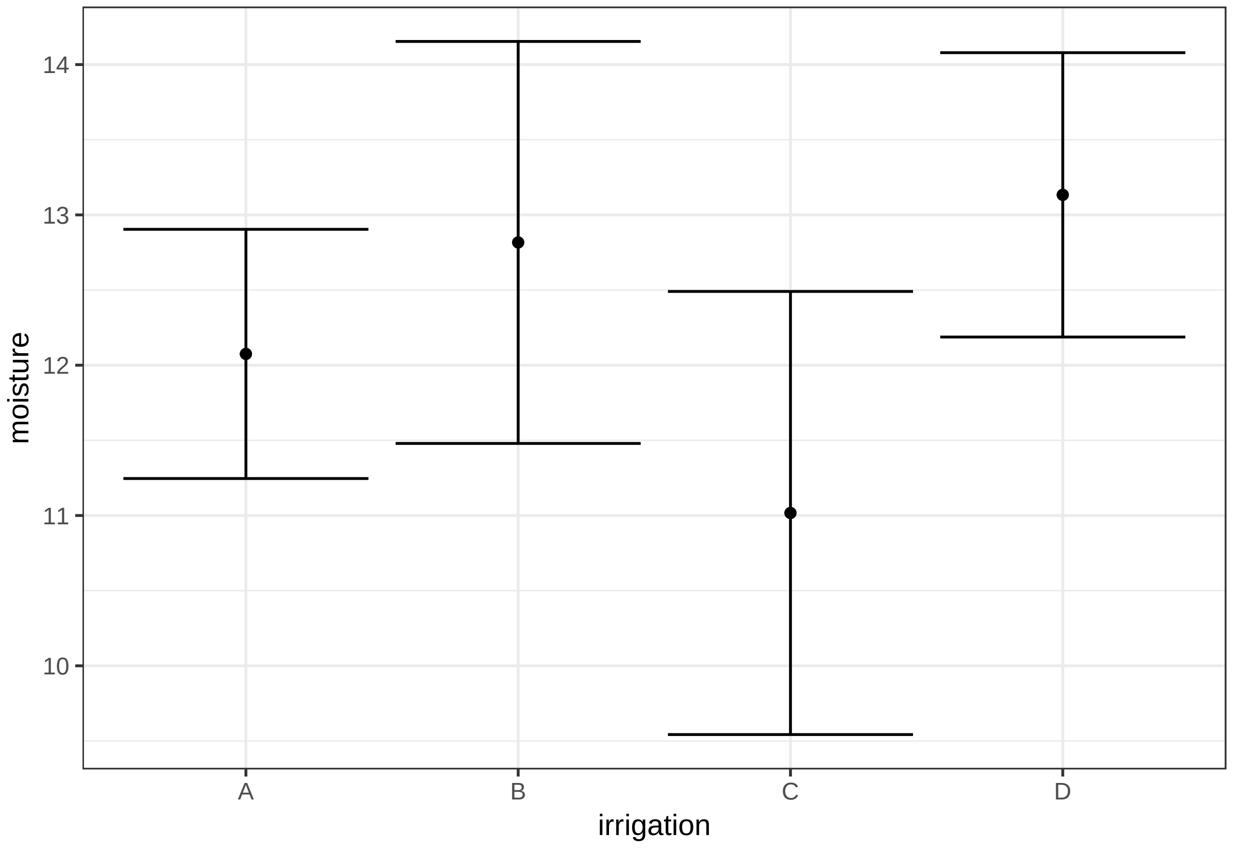
9.4 Visualising the data

```
d$plot<-factor(d$plot)
d$irrigation<-factor(d$irrigation)
g0<-ggplot(d,aes(x=irrigation,y=moisture))
g0+geom_point()
```



Plotting the means with confidence intervals is a good way of identifying patterns that can be tested for significance.

```
g1<-g0+stat_summary(fun.data=mean_cl_normal,geom="point")
g1+stat_summary(fun.data=mean_cl_normal,geom="errorbar",colour="black")
```



OK. So we can analyse the experiment using Anova as we have seen previously.

```
mod<-lm(moisture~irrigation,data=d)
anova(mod)
```

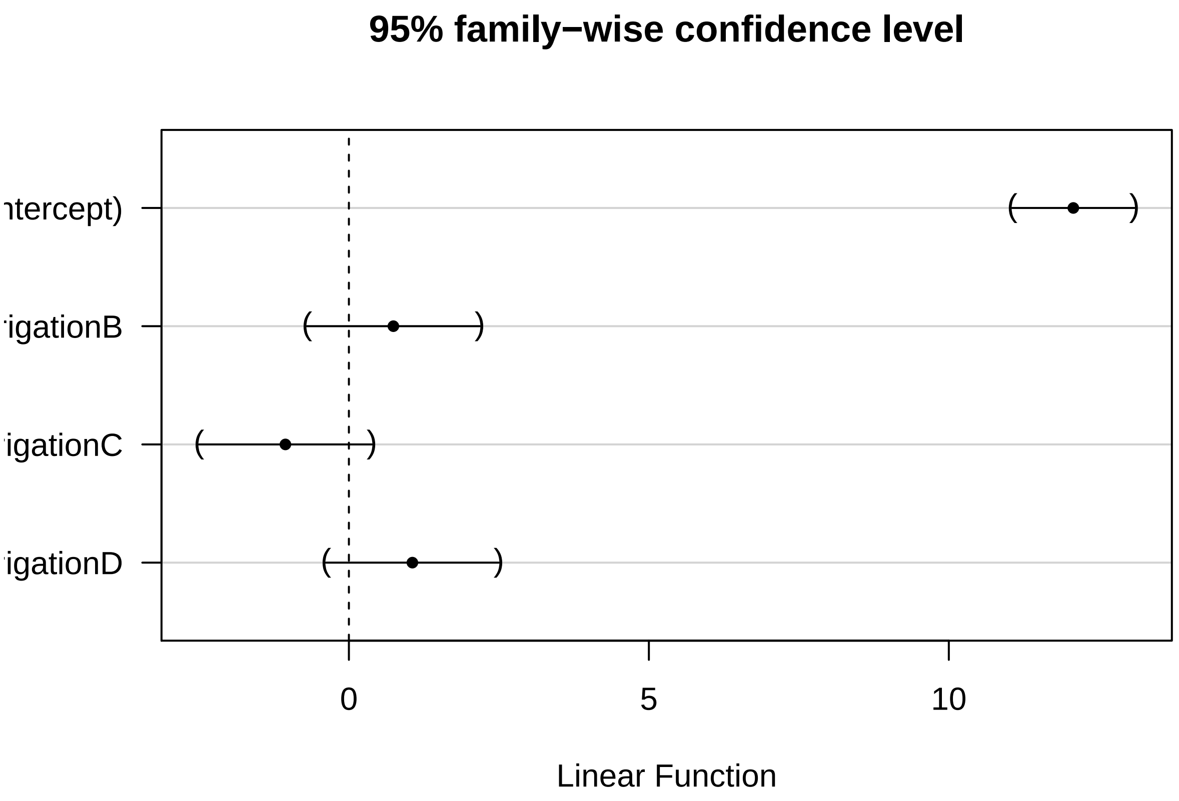
```
## Analysis of Variance Table
##
## Response: moisture
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## irrigation  3 10.6108  3.5369  6.4626 0.007504 **
## Residuals  12  6.5675  0.5473
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nothing new here. You should be able to interpret the table easily.

9.4.1 Comparisons

The multcomp package provides a useful way of summarising the results by testing the general linear hypothesis for each treatment level. By default this will use the treatment that is lowest in alphabetical order as the “control” or baseline.

```
plot(glht(mod))
```



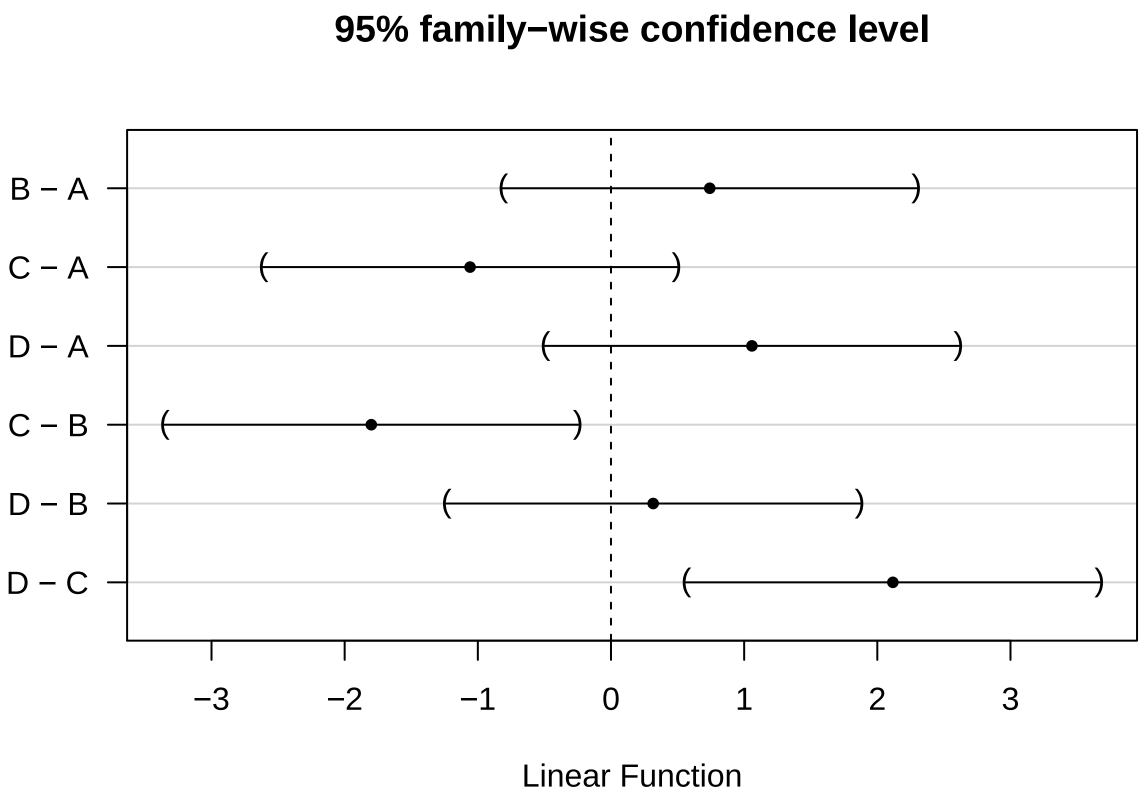
```r
summary(glht(mod))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = moisture ~ irrigation, data = d)
##
## Linear Hypotheses:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept) == 0  12.0750     0.3699  32.644   <0.001 ***
## irrigationB == 0   0.7417     0.5231   1.418    0.418
## irrigationC == 0  -1.0583     0.5231  -2.023    0.172
## irrigationD == 0   1.0583     0.5231   2.023    0.172
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

So, although the anova can be reported as having detected significant variation between treatments (F 3,12 =6.4, p<0.01) there are no significant differences between treatment A and the others.

As no treatment was a natural control the best way to proceed in this case is to look at multiple comparisons to determine where the significant difference lie. We need to make an appropriate adjustment such as Tukey's.

```r
plot(glht(mod, linfct = mcp(irrigation = "Tukey")))
```



```
summary(glht(mod, linfct = mcp(irrigation = "Tukey")))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = moisture ~ irrigation, data = d)
##
## Linear Hypotheses:
##            Estimate Std. Error t value Pr(>|t|)
## B - A == 0   0.7417     0.5231   1.418  0.51263
## C - A == 0  -1.0583     0.5231  -2.023  0.23308
## D - A == 0   1.0583     0.5231   2.023  0.23304
## C - B == 0  -1.8000     0.5231  -3.441  0.02187 *
## D - B == 0   0.3167     0.5231   0.605  0.92840
## D - C == 0   2.1167     0.5231   4.046  0.00747 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

So it appears that treatment C leads to significantly lower soil water content than either B or D. This may be the type of irrigation method to reject an inefficient. The other methods were indistinguishable based on the results from this rather small study. A larger study would be needed to detect significant differences.

9.5 Completely randomised design with subsampling

In this case, all levels of the treatment or all combinations of treatment levels are assigned to experimental units completely at random, but measurements are taken by sub sampling within the experimental unit. Ignoring the dependence that comes from sub sampling would lead to a form of pseudo-replication and this can result in falsely claiming significance (type one error).

In the case of the previous example, it reality the researchers had taken three sub samples from each experimental unit. The previous analysis used the mean that had already been calculated from these three measurements. If the original data were used there may be a temptation to analyse the measurements as if they were all independent. This would be wrong. In fact the sub samples are **nested** within the experimental units.

9.5.1 Wrong analyisis for the subsampling

Any form of sub sampling potentially can lead to type one errors (p-values too small) if it is not recognised for what it is. The reason for this is that the denominator degrees of freedom (the measure of the amount of independent replicatation) in the anova table is too large.

Look carefully at the structure of this data frame. Each level of the factor now has three sub samples. However these are all measures of the same quantity.

```
irrigation2<-read.csv("https://tinyurl.com/aqm-data/irrigation2.csv")
d<-irrigation2
```

```
str(d)
```

```
## 'data.frame':    48 obs. of  4 variables:
##  $ irrigation: Factor w/ 4 levels "A","B","C","D": 1 1 1 1 1 1 1 1 1 1 ...
##  $ plot      : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ subplot   : int  1 2 3 1 2 3 1 2 3 1 ...
##  $ moisture  : num  12.6 11.9 12.3 13 12.4 12.4 11.3 11.9 10.9 12.5 ...
```

This is what happens if we ignore the sub sampling and run an analysis using all the data.

```
d$plot<-factor(d$plot)
mod<-lm(moisture~irrigation,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: moisture
##            Df Sum Sq Mean Sq F value   Pr(>F)
## irrigation  3 31.832 10.6108  18.673 5.84e-08 ***
## Residuals  44 25.002  0.5682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice that this leads to a much smaller p-value. There are too many degrees of freedom in the denominator due to the pseudoreplication.

We have already seen a good solution. Just take the means of each set of sub samples and analyse these as measures taken from each experimental unit. This is often by far the simplest approach. It is easy to understand and easy to explain to others.

Another way of dealing with sub sampling is to account for it by adding a random effect for the experimental unit from which sub-samples are taken.

There are many different ways of doing this in R. If use the aov wrapper to fit linear models for analysis of variance we declare an error term at the plot level and add this to the model formula.

```
mod<-aov(moisture~irrigation+Error(plot),data=d)
summary(mod)
```

```
##
## Error: plot
##            Df Sum Sq Mean Sq F value Pr(>F)
## irrigation  3  31.83  10.611   6.463 0.0075 **
## Residuals  12  19.70   1.642
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 32    5.3  0.1656
```

You should see that the result is almost identical to that produced through analysing the means, as it should be.

One point to be aware of is that when you have nested sub samples within a single level of the treatment, as occurs here, you can't include plot as a fixed effect in the model. If you try to do this you will find that some coefficients cannot be estimated.

```
mod<-lm(moisture~irrigation+plot,data=d)
summary(mod)
```

```
##
## Call:
## lm(formula = moisture ~ irrigation + plot, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93333 -0.26667  0.03333  0.26667  0.66667
##
## Coefficients: (3 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.2667     0.2350  52.206  < 2e-16 ***
## irrigationB   0.9667     0.3323   2.909 0.006542 **
## irrigationC  -1.2333     0.3323  -3.712 0.000782 ***
## irrigationD   0.9667     0.3323   2.909 0.006542 **
## plot2         0.3333     0.3323   1.003 0.323319
## plot3        -0.9000     0.3323  -2.708 0.010762 *
## plot4        -0.2000     0.3323  -0.602 0.551492
## plot5        -0.8667     0.3323  -2.608 0.013719 *
## plot6        -1.3333     0.3323  -4.013 0.000338 ***
## plot7         0.5333     0.3323   1.605 0.118314
## plot8             NA         NA      NA       NA
## plot9         0.8667     0.3323   2.608 0.013719 *
## plot10        0.3667     0.3323   1.103 0.278059
## plot11       -1.3000     0.3323  -3.912 0.000448 ***
## plot12            NA         NA      NA       NA
## plot13       -0.6000     0.3323  -1.806 0.080388 .
## plot14        0.7000     0.3323   2.107 0.043084 *
## plot15       -0.5000     0.3323  -1.505 0.142206
## plot16            NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.407 on 32 degrees of freedom
## Multiple R-squared:  0.9067, Adjusted R-squared:  0.863
## F-statistic: 20.74 on 15 and 32 DF,  p-value: 2.466e-12
```

The model can also be fit using either the lmer package or the older nlme package. The differences between the two are rather technical. In general terms nlme can be more convenient for mixed effects models which involve a response to a continuous variable, particularly if the response is non-linear. The newer lme4 package can fit some very complex model involving multi-layer interactions that nlme cannot. However by default lme4 does not provide p-values due to a deliberate decision made by its author. There is a highly technical academic argument regarding the validity of the calculations. Fortunately, as we will often do want to report p-values, the relevant calculations have been added by the author of the lmerTest package.

```
library(lmerTest)
mod<-lmer(moisture~irrigation+(1|plot),data=d)
anova(mod)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##            Sum Sq Mean Sq NumDF DenDF F value   Pr(>F)
## irrigation 3.2111  1.0703     3    12  6.4625 0.007505 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-values are the same as would result from using the means of the sub samples in a one way anova. So there is no obvious advantage in using the raw data rather than pooled means.

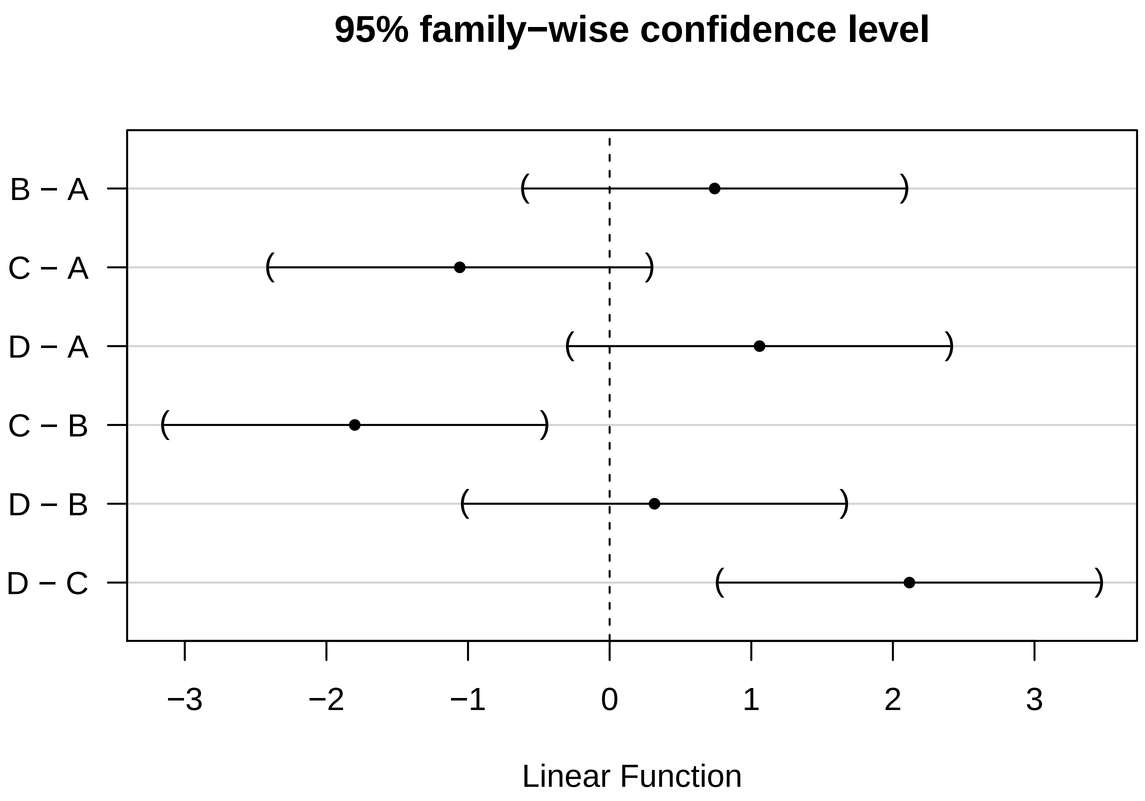
In many cases the simplest way to analyse the data is to just take the means of the sub samples and use those, as in the first example. The exception to this is if you are interested in variability **within** the sub samples. Notice that the mixed effect does provide you with an estimate of this.

```
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: moisture ~ irrigation + (1 | plot)
##    Data: d
##
## REML criterion at convergence: 83.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6115 -0.6833  0.1057  0.5153  1.3200
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  plot     (Intercept) 0.4921   0.7015
##  Residual             0.1656   0.4070
## Number of obs: 48, groups:  plot, 16
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  12.0750     0.3699 11.9998  32.644 4.32e-13 ***
## irrigationB   0.7417     0.5231 11.9998   1.418   0.1817
## irrigationC  -1.0583     0.5231 11.9998  -2.023   0.0659 .
## irrigationD   1.0583     0.5231 11.9998   2.023   0.0659 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) irrgtB irrgtC
## irrigationB -0.707
## irrigationC -0.707  0.500
## irrigationD -0.707  0.500  0.500
```

We can use the glht function with the output from fitting this model as before, and reach an identical conclusion.

```
plot(glht(mod, linfct = mcp(irrigation = "Tukey")))
```



```
summary(glht(mod, linfct = mcp(irrigation = "Tukey")))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lmer(formula = moisture ~ irrigation + (1 | plot), data = d)
##
## Linear Hypotheses:
##              Estimate Std. Error z value Pr(>|z|)
## B - A == 0    0.7417     0.5231   1.418  0.48811
## C - A == 0   -1.0583     0.5231  -2.023  0.17946
## D - A == 0    1.0583     0.5231   2.023  0.17933
## C - B == 0  -1.8000     0.5231  -3.441  0.00312 **
## D - B == 0   0.3167     0.5231   0.605  0.93041
## D - C == 0   2.1167     0.5231   4.046  < 0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Always watch out for subsampling in any data set.

Never treat subsamples as independent replicates

9.6 Randomized Complete Block Design

Blocks occur when the variability between the response within sets of experimental units is expected to be lower than the variability between sets of units. The term block comes from agricultural experiments and is easiest to understand in this context. If different fields have different inherent fertility levels we would expect the yield of wheat from parcels placed within each field to be similar. However the yield would vary between fields. The idea of blocking is to account for this variability in the design of the experiment. This can increase statistical power and reduce the number of experimental units needed.

Here is an example of a randomised complete block design. Four fields are chosen for an experiment. In each field four plots are selected and allocated a treatment level at random. Let's say the treatment involves planting four different varieties of maize and recording the total yield.

The fields are blocks. Each field may have a different soil type and thus a different intrinsic fertility. In the case of a blocked experiment we are not interested in the effect of the block. It is a confounding variable. However it adds to the variability between treatment levels and thus may make it harder to spot real differences between them.

Block effects need to be taken into account as if they are not the analysis may have too high a p-value (reduced power or type 2 error). The situation is rather complicated conceptually as we can only estimate the effects of the blocks. We don't really know what they are.

```
RCB<-read.csv("https://tinyurl.com/aqm-data/RCB.csv")
d<-RCB
```

```
str(d)
```

```
## 'data.frame':    16 obs. of  3 variables:
##  $ block: Factor w/ 4 levels "b1","b2","b3",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ treat: Factor w/ 4 levels "t1","t2","t3",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ yield: num  18.1 22.6 21.5 30.8 26 ...
```

9.6.1 Wrong analysis ignoring the effect of block

Let's first look at all the sources of variability combined at the level of the treatments.

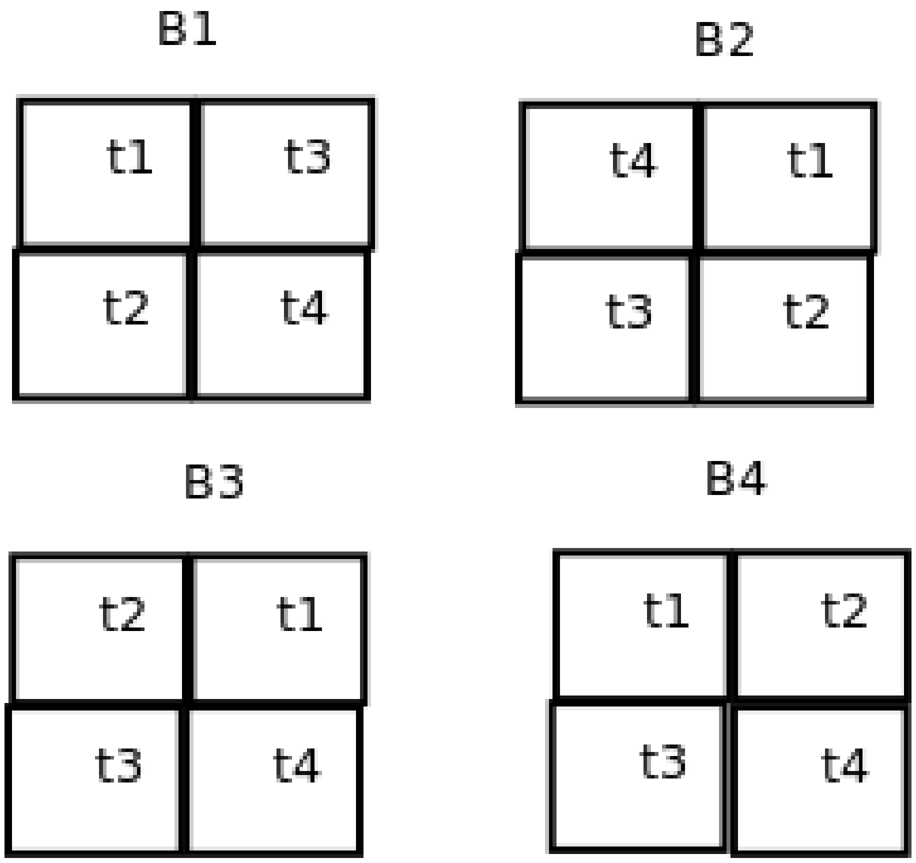
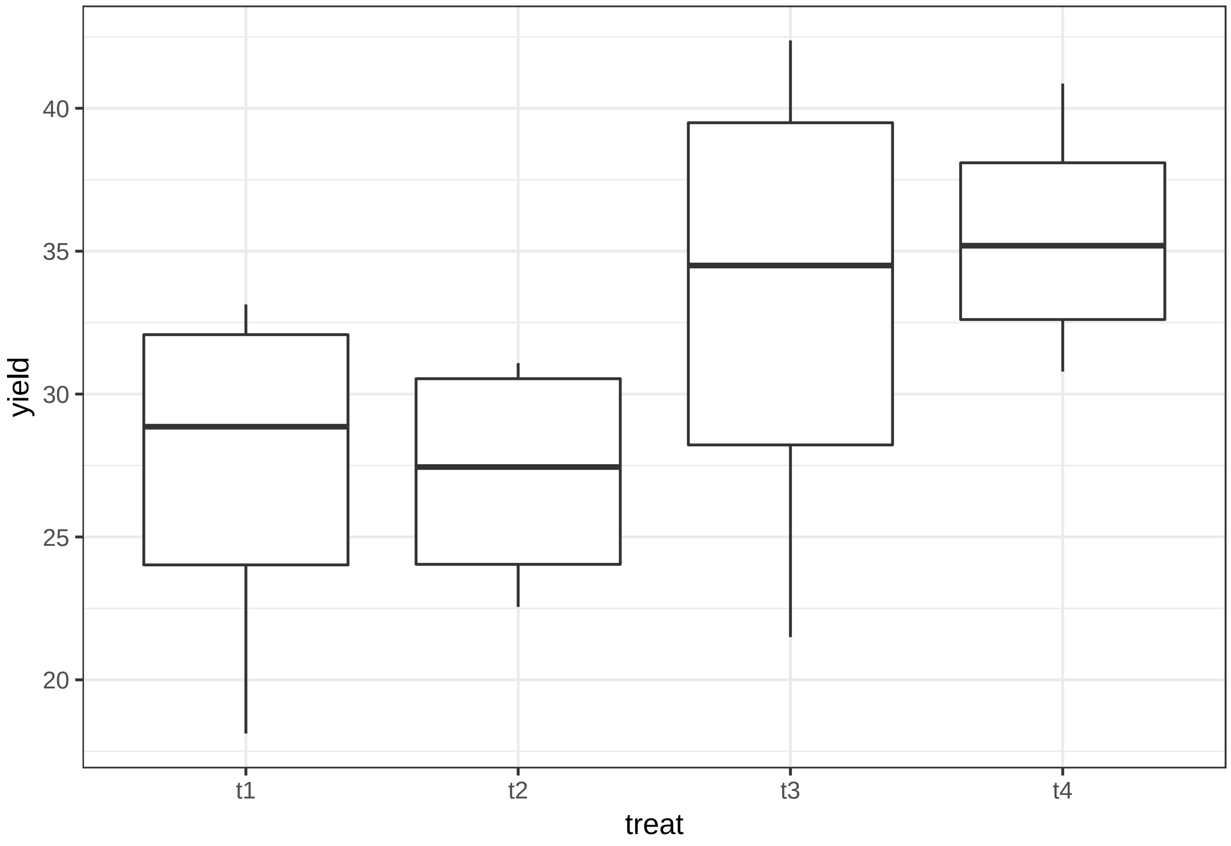
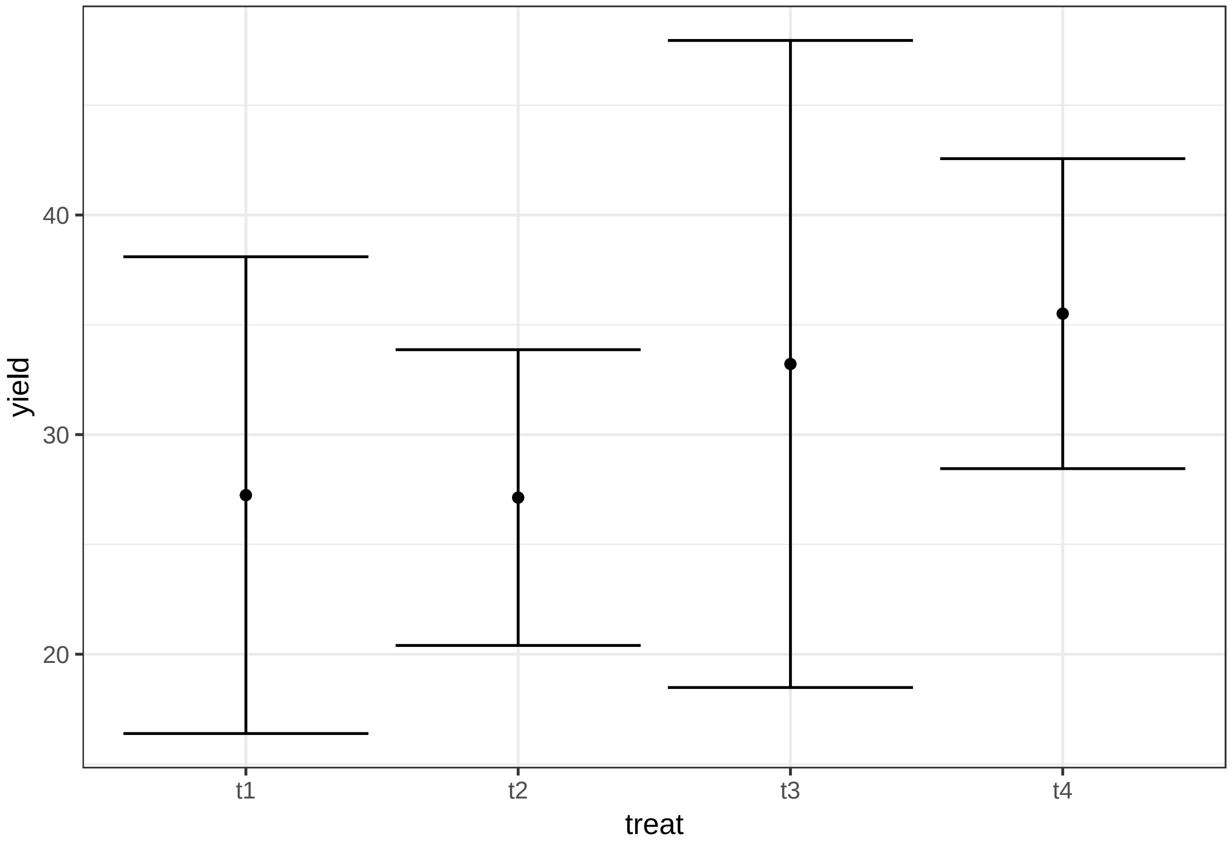


Figure 9.2: alt text



```
g1<-g0+stat_summary(fun.data=mean_cl_normal,geom="point")
g1+stat_summary(fun.data=mean_cl_normal,geom="errorbar",colour="black")
```



We can fit a model ignoring the blocking effects.

```r
mod<-lm(yield~treat,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## treat      3 216.45  72.150  1.6991 0.2201
## Residuals 12 509.56  42.463
```

Notice that this suggests that there is no significant effect of the treatments.

9.6.2 Treating block as a random effect

This time we will fit a model with an error term at the block level using aov.

```r
mod<-aov(yield~treat+Error(block),data=d)
summary(mod)
```

```
##
## Error: block
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  3  446.1   148.7
##
## Error: Within
##           Df Sum Sq Mean Sq F value  Pr(>F)
## treat      3 216.45   72.15   10.23 0.00294 **
## Residuals  9  63.49    7.05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

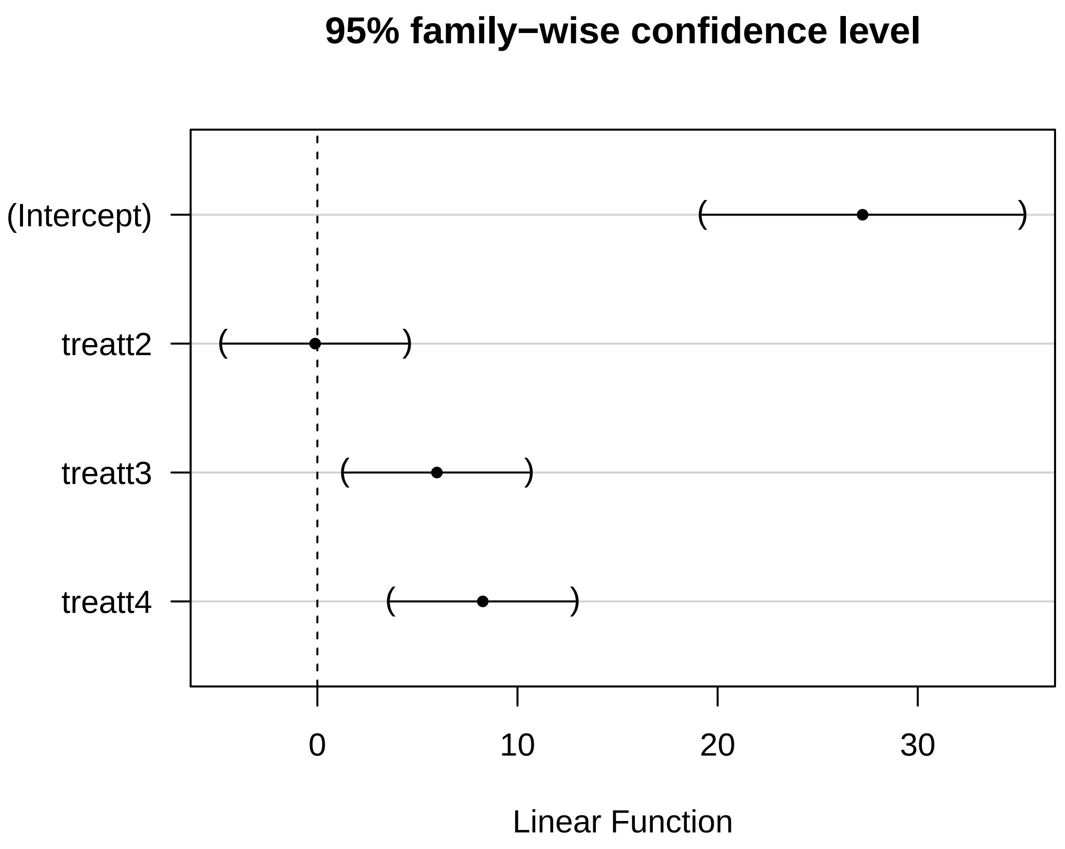
Again, the same result can be obtained using the more powerful lmerTest package.

```
mod<-lmer(yield~treat+(1|block),data=d)
anova(mod)
```

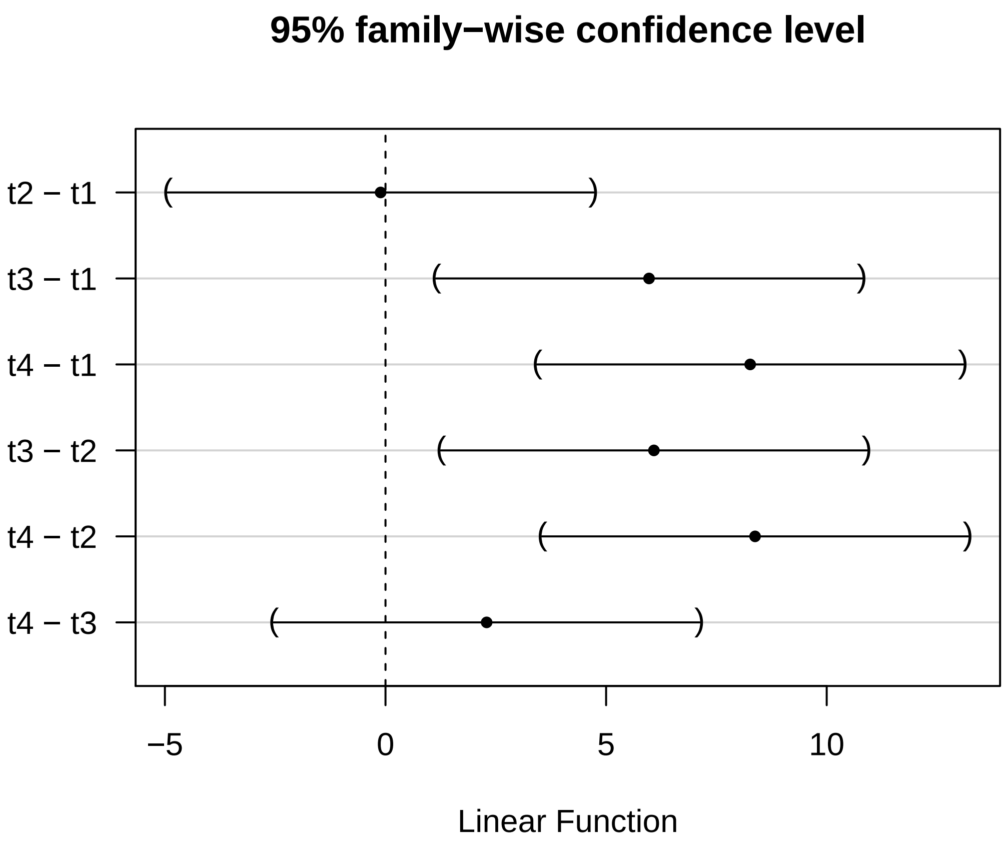
```
## Type III Analysis of Variance Table with Satterthwaite's method
##       Sum Sq Mean Sq NumDF  DenDF F value   Pr(>F)
## treat 216.45   72.15     3 9.0001  10.227 0.002941 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Just as previously we can look at the pattern of effects using the glht function from the multcomp package.

```
par(mar=c(4,8,4,2))
plot(glht(mod))
```



```
par(mar=c(4,8,4,2))
plot(glht(mod, linfct = mcp(treat = "Tukey")))
```



9.6.3 Treating block as a fixed effect

If we treat block as fixed we obtain identical results for the treatment effects. It may be advantageous to treat the blocks as fixed if we wish to identify differences between them. For example one field may have a very different soil fertility to the others. As blocking effects confound detection of treatment effects even when we allow for them in the statistical model we might decide not to use this block in further trials in order to have a more homogeneous set of initial conditions. The difference between using a fixed and random effect in this situation is arbitrary and depends on circumstances.

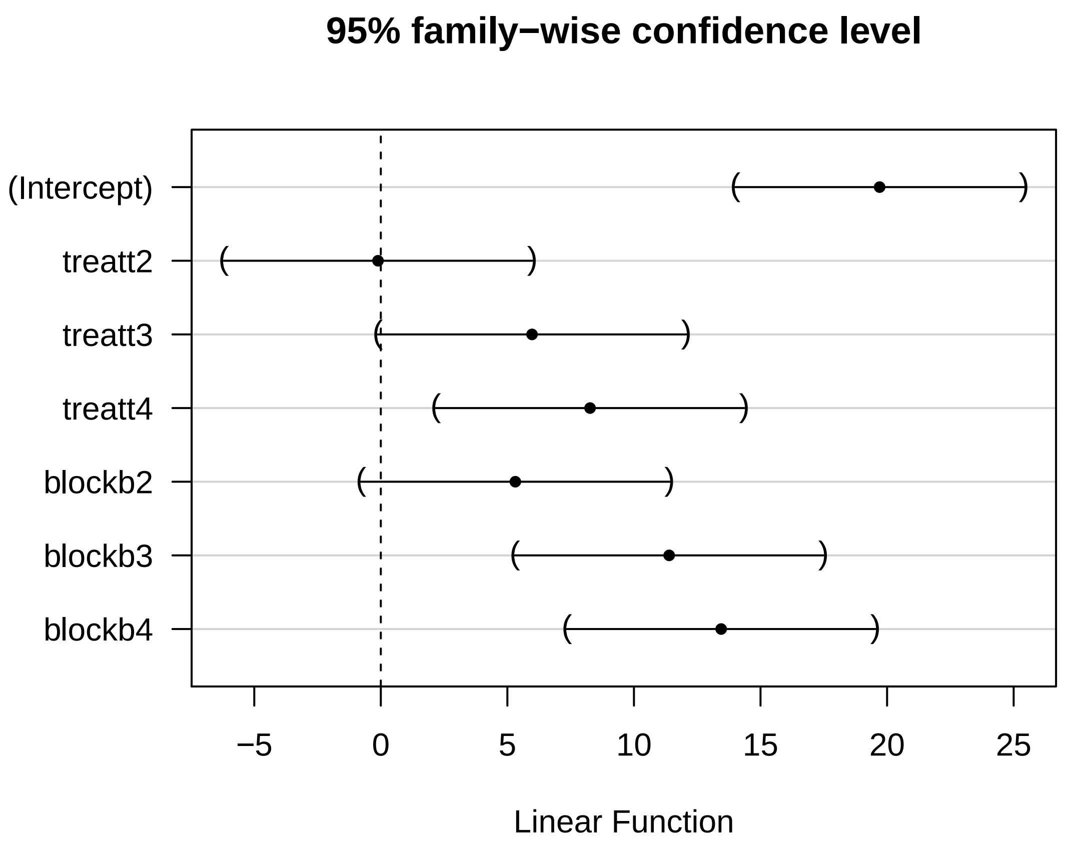
```
mod<-aov(yield~treat+block,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value    Pr(>F)
## treat      3 216.45  72.150  10.227 0.0029416 **
## block      3 446.06 148.688  21.076 0.0002086 ***
## Residuals  9  63.49   7.055
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

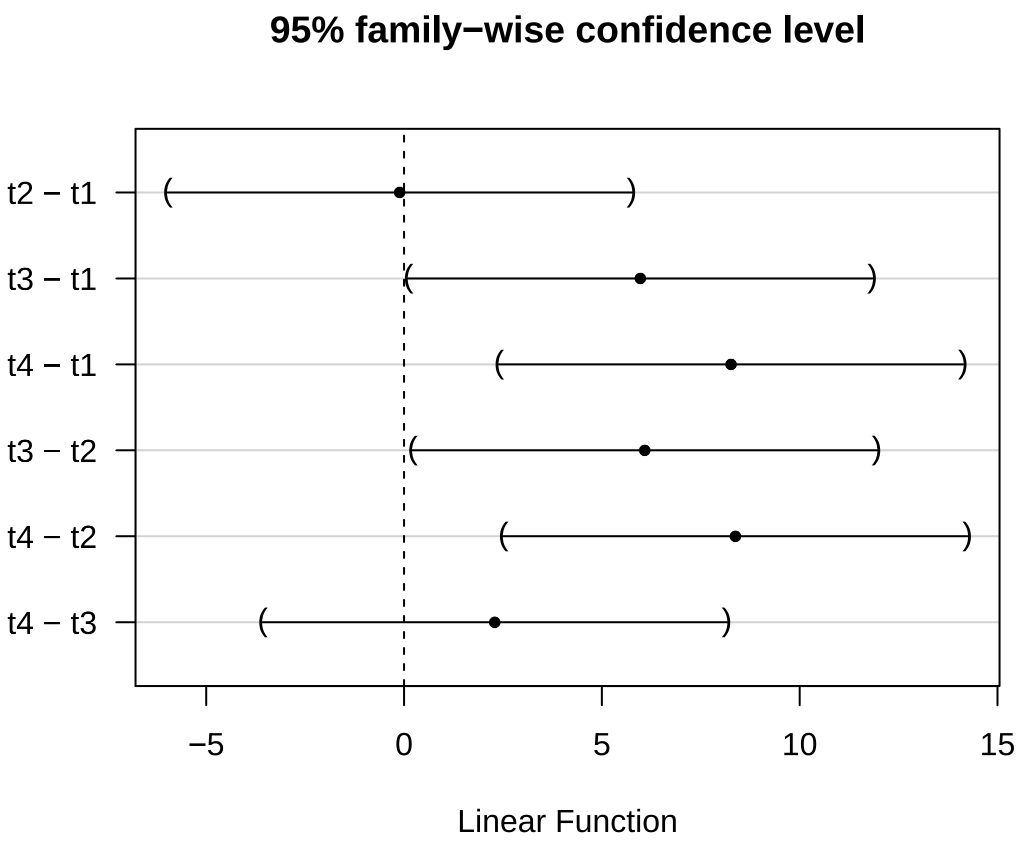
```
summary(mod)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## treat        3  216.5   72.15   10.23 0.002942 **
## block        3  446.1  148.69   21.08 0.000209 ***
## Residuals    9   63.5    7.05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

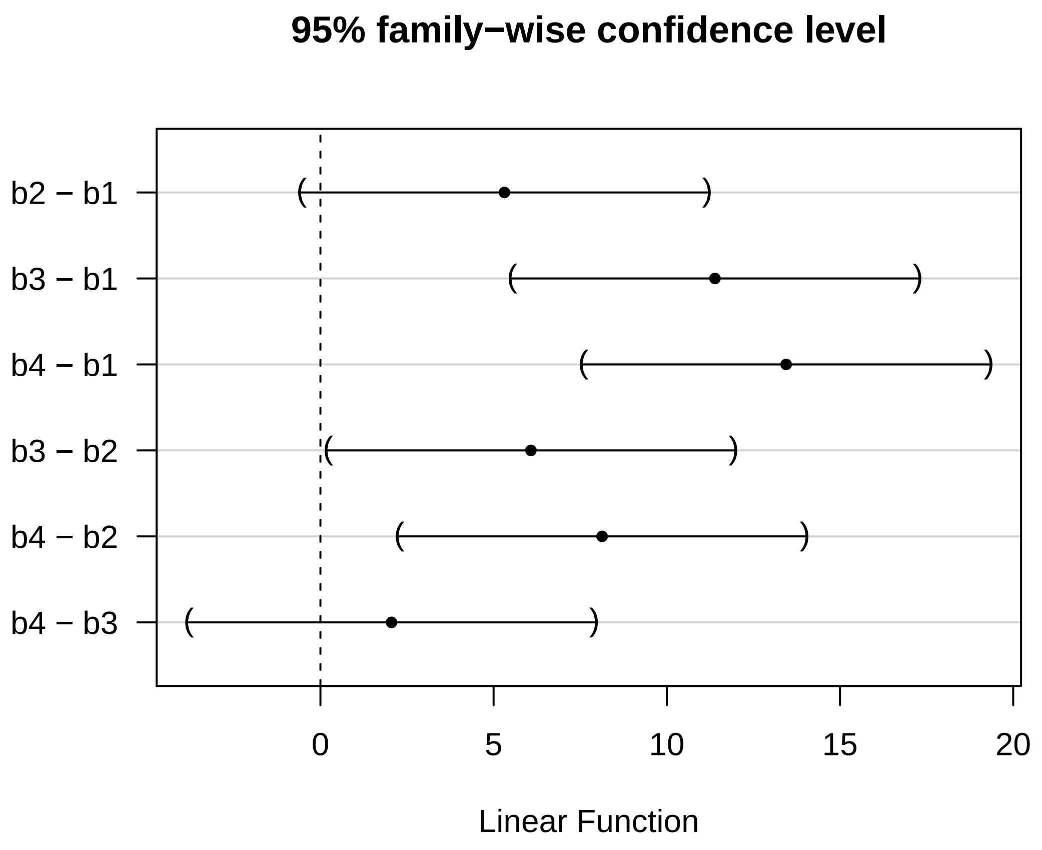
```
par(mar=c(4,8,4,2))
plot(glht(mod))
```



```
par(mar=c(4,8,4,2))
plot(glht(mod, linfct = mcp(treat = "Tukey")))
```



```r
par(mar=c(4,8,4,2))
plot(glht(mod, linfct = mcp(block = "Tukey")))
```



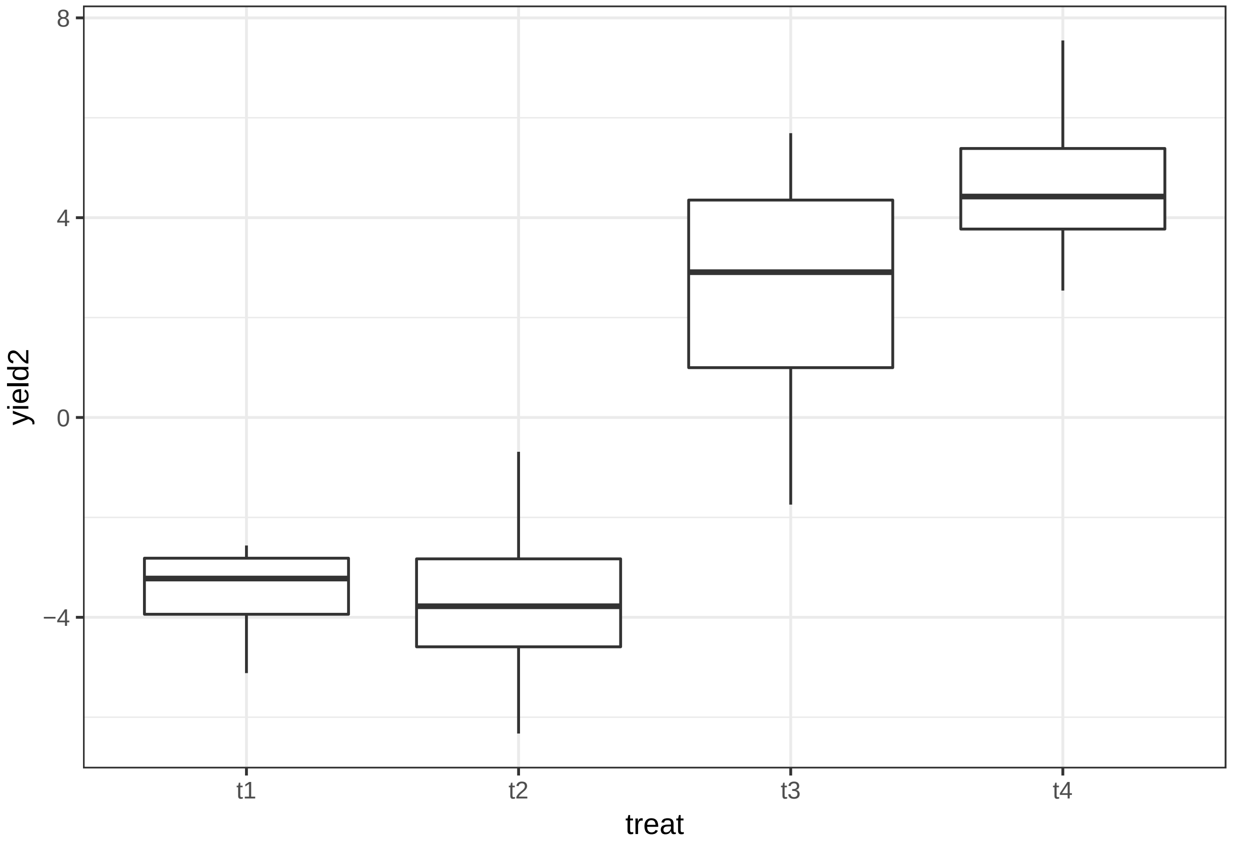
9.7 Illustation of how block effects work

We could handle the block effect "by hand". Each block of four treatments has a mean yield that we can calculate in R and add to our data frame.

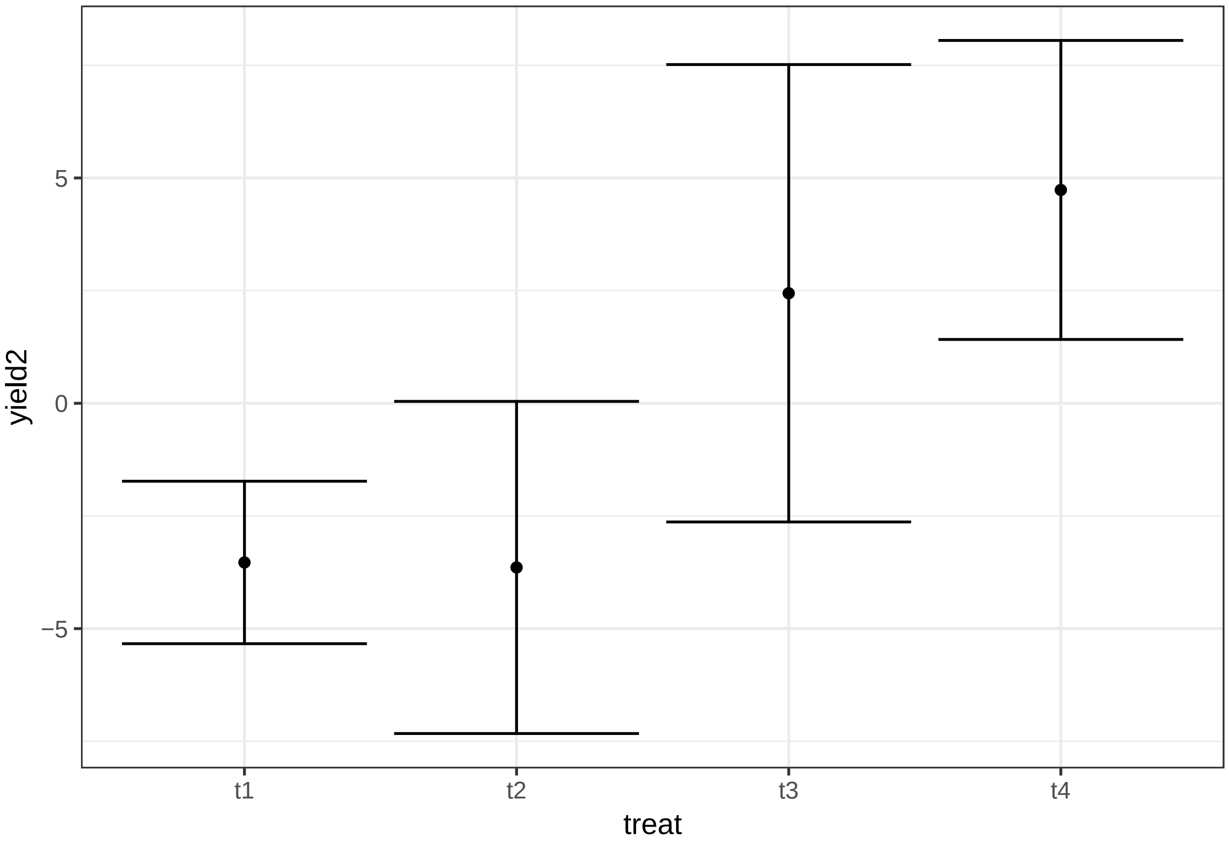
```
d$block_mean<-rep(with(d,tapply(yield,block,mean)),each=4)
```

Look at the results of this in the data table. Now we can subtract the block mean from the yields and re-plot our confidence intervals.

```
d$yield2<-d$yield-d$block_mean
g0<-ggplot(d,aes(x=treat,y=yield2))
g0+geom_boxplot()
```



```
g1<-g0+stat_summary(fun.data=mean_cl_normal,geom="point")
g1+stat_summary(fun.data=mean_cl_normal,geom="errorbar",colour="black")
```



The pattern that we detected using the analysis which took into account blocks is now clearly shown.

Now if we run an Anova you can see that the sum of squares and the mean squares are correct. The residual sum of squares is also correct. The difference is that the denominator degrees of freedom is too high as we have not compensated for the fact that the block means were estimated, by reducing the df by three, The mean squares in the denominator of the F Ratio is therefore slightly too low. Looking closely at the table should help you understand the logic.

```
mod<-lm(yield2~treat,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: yield2
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## treat      3 216.451  72.150  13.636 0.0003583 ***
## Residuals 12  63.494   5.291
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Although this produces an identical result in terms of the sum of squares it is preferable to declare the block effect as either an additional additive fixed effect or a random effect in order to clearly show that the block effects are estimated by the model and thus lead to reduced degrees of freedom.

9.8 An observational example of sub-sampling

A researcher wants to establish if algal mat cover has an effect on species richness. The mat cover is measured once but richness five times at each site.

This is an examle of a regression with sub-sampling. The issue to be dealt with is the difference in the amount of replication of the measure of algal mat and the measure of richness. It can be handled either by taking means or by using a model with a random effect included to handle nested observations.

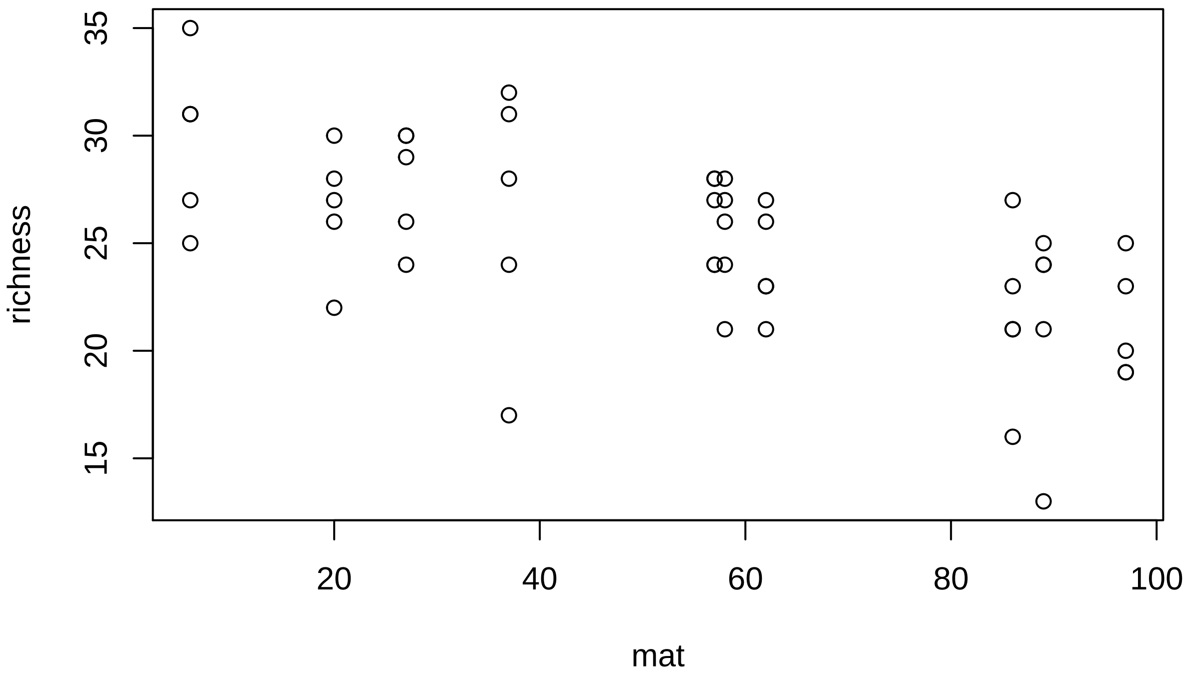
```
set.seed(1)
mat<-rep(sample(1:100,10),each=5)
mat
```

```
##  [1] 27 27 27 27 27 37 37 37 37 37 57 57 57 57 57 89 89 89 89 89 20 20 20
## [24] 20 20 86 86 86 86 86 97 97 97 97 97 62 62 62 62 62 58 58 58 58 58  6
## [47]  6  6  6  6
```

```
site<-rep(1:10,each=5)
richness<-round(30-0.1*mat+rnorm(50,0,4),0)
richness
```

```
##  [1] 24 29 30 30 26 32 28 24 17 31 24 24 28 28 27 25 24 21 13 24 28 27 22
## [24] 26 30 27 21 23 21 16 19 19 20 25 23 23 23 27 26 21 21 26 27 24 28 31
## [47] 27 31 25 35
```

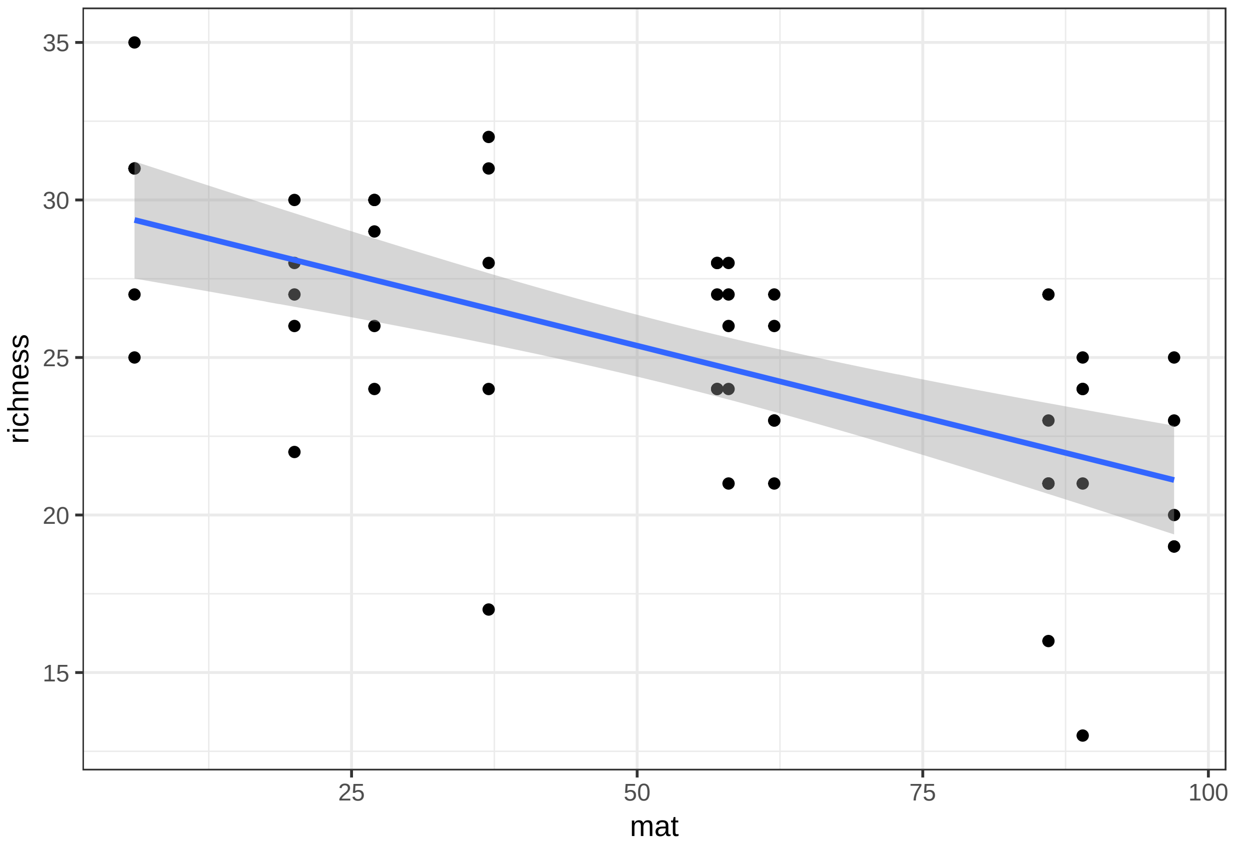
```
plot(mat,richness)
```



```
d<-data.frame(site,mat,richness)
```

9.8.1 Plot the raw data

```
g0<-ggplot(d,aes(x=mat,y=richness))
g0+geom_point()+geom_smooth(method="lm")
```



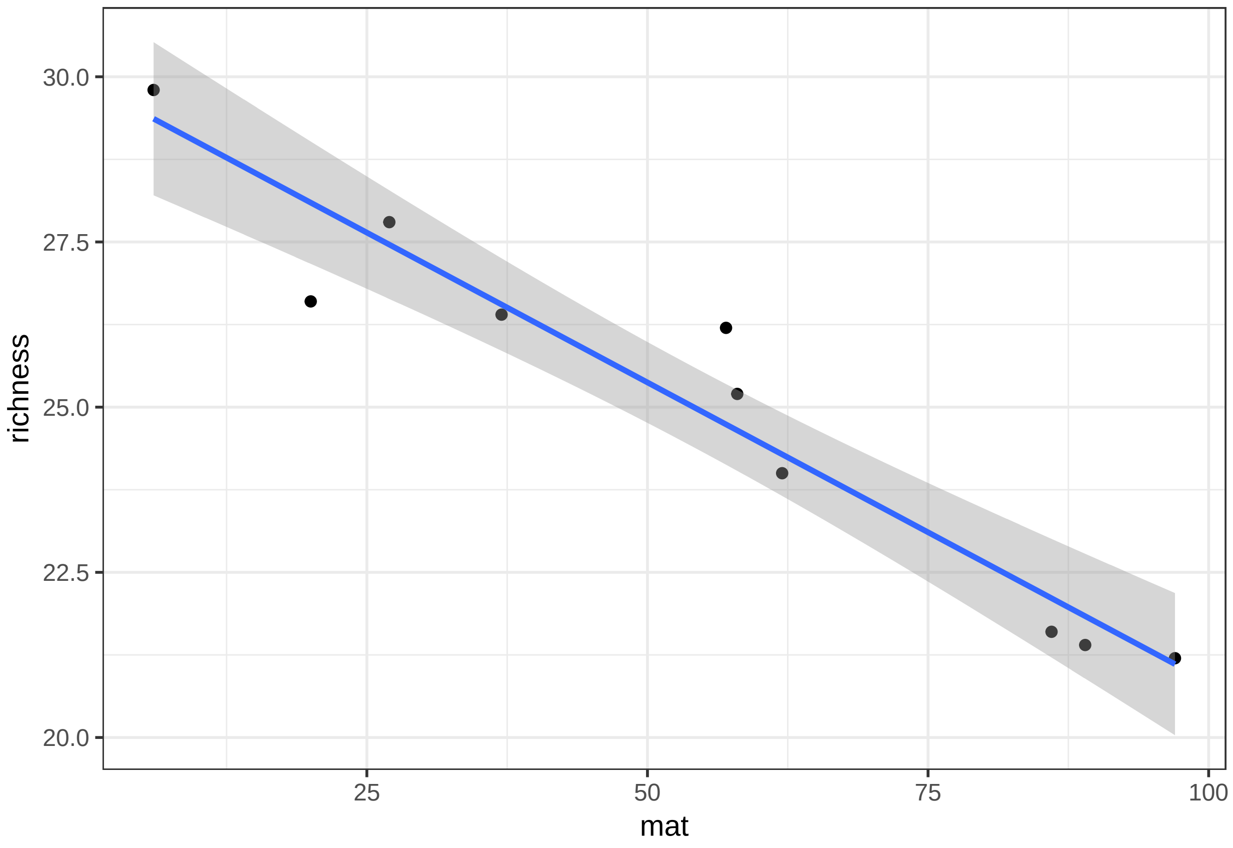
```
mod1<-lm(richness~mat,data=d)
summary(mod1)
```

```
##
## Call:
## lm(formula = richness ~ mat, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5530 -1.9363  0.3984  2.3292  5.6351
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.90919    1.01005  29.611  < 2e-16 ***
## mat         -0.09071    0.01645  -5.515 1.37e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.423 on 48 degrees of freedom
## Multiple R-squared:  0.3879, Adjusted R-squared:  0.3752
## F-statistic: 30.42 on 1 and 48 DF,  p-value: 1.366e-06
```

9.8.2 Group to take mean richness at each site with same algal coverage

```
library (dplyr)
```

```
d %>% group_by(site,mat) %>% summarise(richness=mean(richness)) ->d2


g0<-ggplot(d2,aes(x=mat,y=richness))
g0+geom_point()+geom_smooth(method="lm")
```

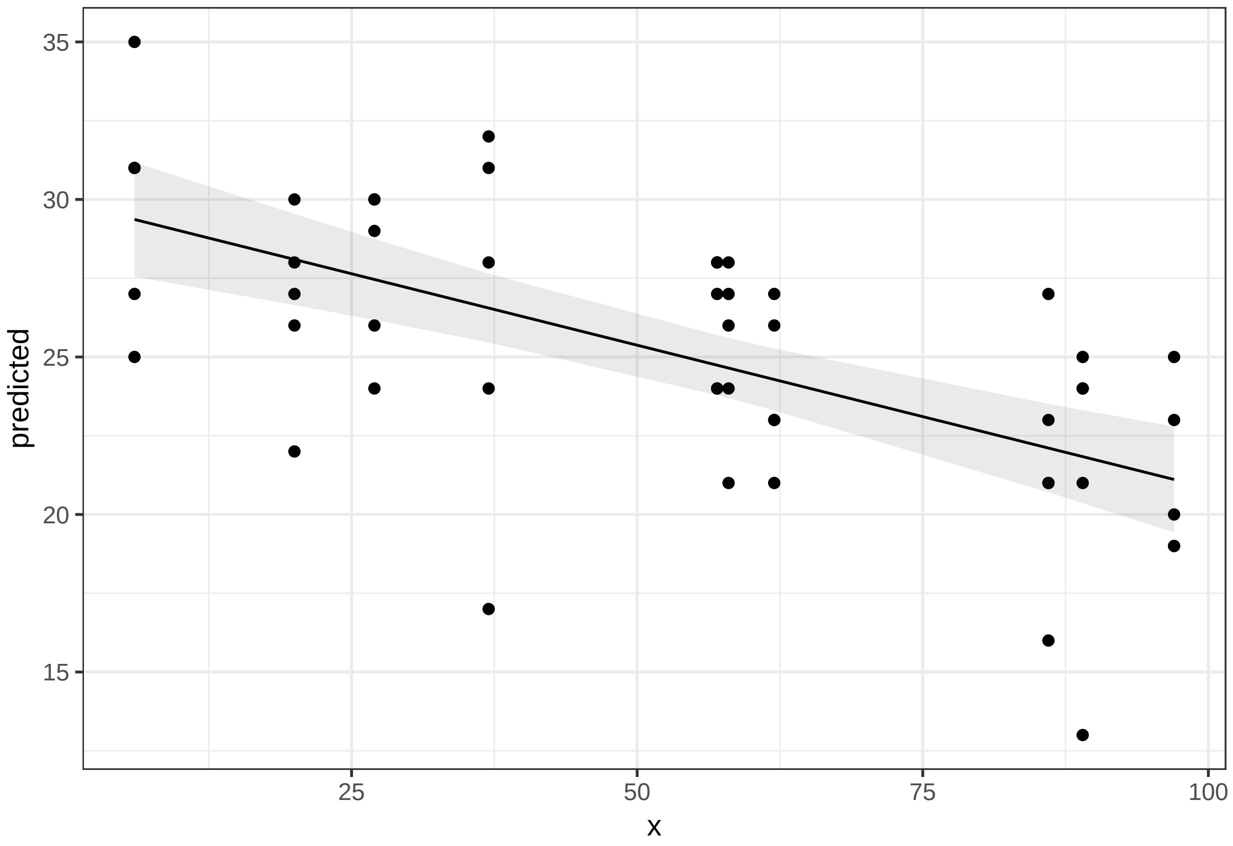


```
mod2<-lm(richness~mat,data=d2)
summary(mod2)
```

```
##
## Call:
## lm(formula = richness ~ mat, data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49502 -0.39841 -0.03172  0.41128  1.46120
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.909188   0.549077   54.47 1.43e-11 ***
## mat         -0.090708   0.008941  -10.15 7.62e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8322 on 8 degrees of freedom
## Multiple R-squared:  0.9279, Adjusted R-squared:  0.9189
## F-statistic: 102.9 on 1 and 8 DF,  p-value: 7.618e-06
```

9.8.3 Use raw data with a random effect for site

```
library(nlme)
library(merTools)
library(ggeffects)

mod3<-lme4::lmer(richness~mat +(1|site),data=d)
fit<-ggpredict(mod3, terms = "mat")
ggplot(fit, aes(x, predicted)) + geom_point(data=d,aes(y=richness,x=mat)) + geom_line() +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .1)
```



```
library(lmerTest)
mod4<-lmer(richness~mat+(1|site),data=d)
summary(mod4)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: richness ~ mat + (1 | site)
##    Data: d
##
## REML criterion at convergence: 268.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7906 -0.5656  0.1164  0.6804  1.6461
##
## Random effects:
```

```
##  Groups   Name        Variance Std.Dev.
##  site     (Intercept)  0.00    0.000
##  Residual             11.72    3.423
## Number of obs: 50, groups:  site, 10
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept) 29.90919    1.01005 48.00000  29.611  < 2e-16 ***
## mat         -0.09071    0.01645 48.00000  -5.515 1.37e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## mat -0.878
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
anova(mod4)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##     Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## mat 356.48  356.48     1    48  30.419 1.366e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

9.9 Summary

Including random effects can allow for subsampling which leads to non-independent repllication. It often will lead to similar results to the simpler technique of taking means of the subsamples. If the number of subsamples actually varies at each observational point then a random effects model is better than one using means, as the mean will be based on a different number of observations and may vary in reliability. A random effects model can take this into account. In other situations the result will be very similar. Blocking differs from subsampling as the treatment is repeated within the block (which it isn't in a nested subsampling design). Blocks can be treated as **either** fixed or random effects. The result will be the same. Treat blocks as **fixed** effects if you are interested in them for some reason. Treat them as random effects if you never want to look at individual blocks as such.

Watch out for subsampling which leads to different amounts of repication for each variable in observational studies! This often occurs and it can lead to erroneous conclusions.

9.10 Exercises

A researcher is interested in whether dark corrugated iron strips are used more frequently by sand lizards than plain corrugated iron. The researcher has 20 pieces of iron of each type and places them on five different sites at Arne. The strips are inspected every day for two weeks in spring. The total number of sandlizards found under each strip is recorded each day as the response variable (data may also be collected on weather conditions etc.. but you can ignore this).

Design a fake data set that could be used as the template for a simple analysis of the data using an appropriate analysis of variance. Run an analysis on the fake data.

Answer the folowing questions.

1. What feature of the design may be considered to lead to blocking?
2. How many levels of the factor are there?
3. How might subsampling be handled?
4. Which feature of the response variable may cause particular difficulties when working with real life data?

Chapter 10

Repeat measures designs

Repeated measures on the same experimental subject can occur for many reasons. The typical situation is when the measures are repeated in time, but repeated measures can also arise in other ways. The key to understanding when a design involves repeated measures is to realise that measures which involve different levels of the fixed factor of interest are taken from the same experimental unit. A blocked design is therefore an example of repeated measures in space if a block is regarded as an experimental unit. However blocks are usually thought of in terms of groups of experimental units. The terminology in the literature can be slightly confusing in this respect as it may depend on the discipline involved and blocked designs can be called repeated measures in some circumstances.

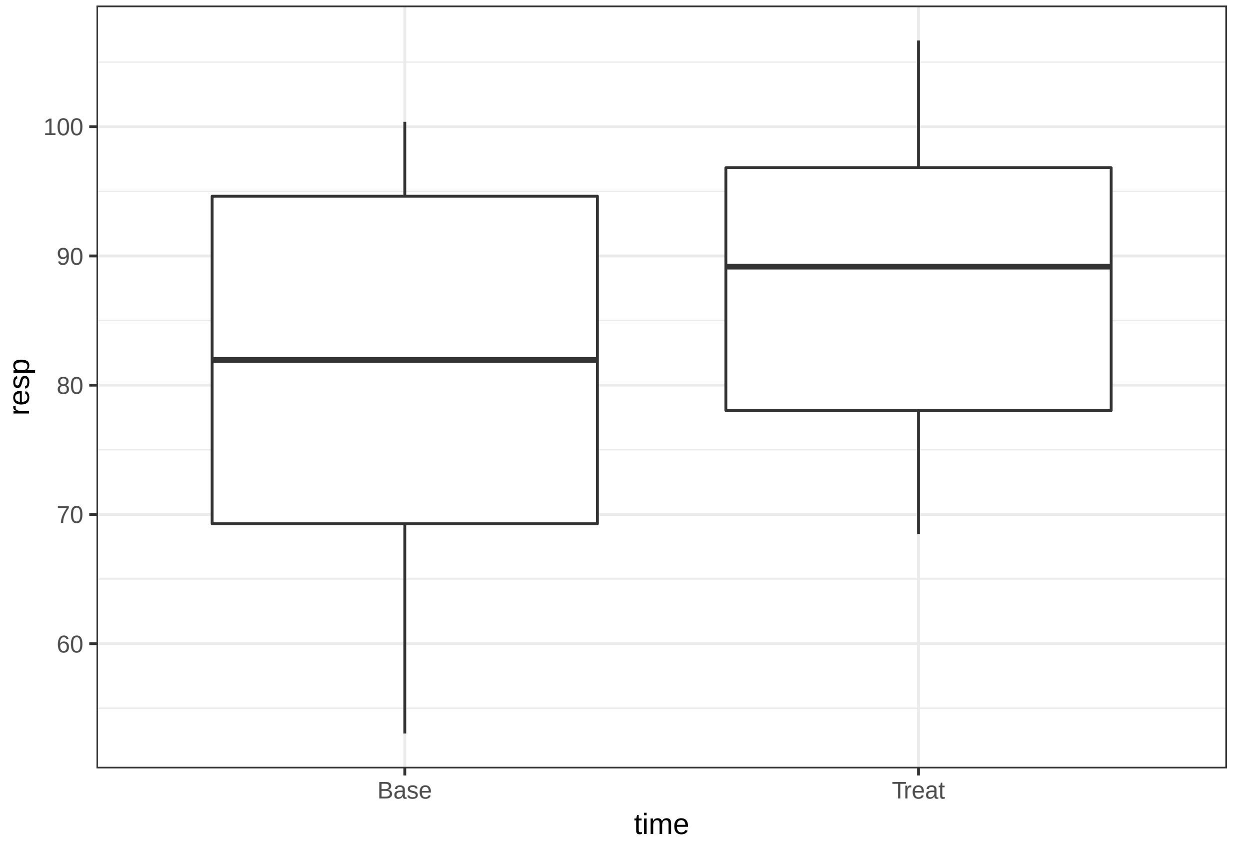
Let's think of a simple example of a repeated measurement in a laboratory setting. The blood pressure of ten rats was measured before and after the injection of a drug.

```r
library(lmerTest)
library(ggplot2)
library(effects)
library(reshape)
library(multcomp)
library(dplyr)
```

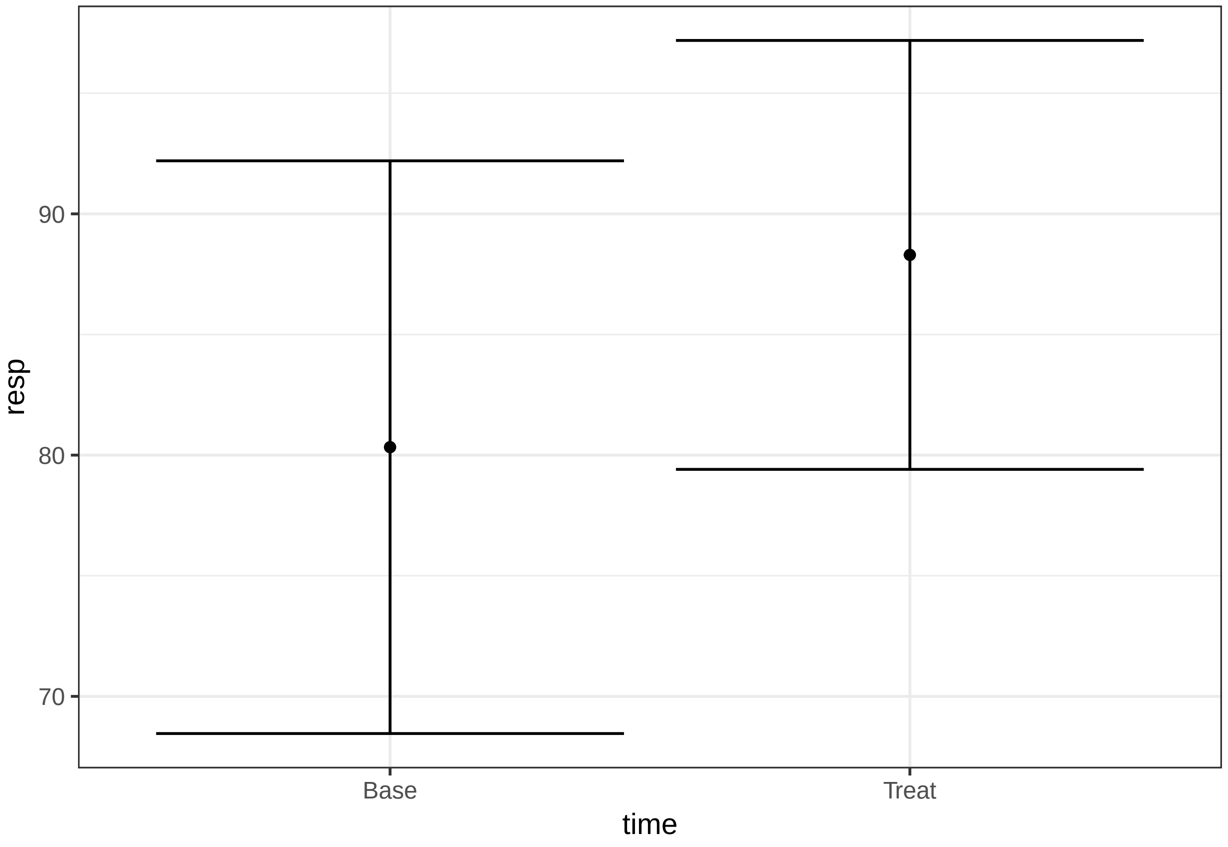
```r
d<-read.csv("https://tinyurl.com/aqm-data/rats.csv")
```

Repeat measures such as these lead to a situation that is fundamentally similar to a design with blocks. Each treatment level is replicated once within each subject. However the total variability also has a between subject component as each rat may have a different baseline blood pressure. This needs to be taken into account.

```r
g0<-ggplot(d,aes(x=time,y=resp))
g0+geom_boxplot()
```



```
g1<- g0+stat_summary(fun.data=mean_cl_normal,geom="point")
g1+stat_summary(fun.data=mean_cl_normal,geom="errorbar",colour="black")
```



10.1 Paired t-test

Just as in the blocked design failure to take into account between subject variability reduces the power of the analysis. We can see this by running two t-tests. The first is unpaired. The second is paired.

```
d2<-melt(d,id=1:2,m="resp")
d2<-cast(d2,id~time)
t.test(d2$Base,d2$Treat)
```

```
##
##  Welch Two Sample t-test
##
## data:  d2$Base and d2$Treat
## t = -1.2157, df = 16.68, p-value = 0.241
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -21.820632   5.881527
## sample estimates:
## mean of x mean of y
##  80.32928  88.29883
```

```
t.test(d2$Base,d2$Treat,paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  d2$Base and d2$Treat
## t = -4.3212, df = 9, p-value = 0.00193
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -12.141652  -3.797453
## sample estimates:
## mean of the differences
##               -7.969552
```

10.2 Mixed effects model

The design can be analysed using a mixed effect model with rat id as a random effect.

First let's see what happens if we don't include the random effect.

```
mod<-lm(resp~time,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: resp
##           Df Sum Sq Mean Sq F value Pr(>F)
## time       1  317.6  317.57   1.478 0.2398
## Residuals 18 3867.7  214.87
```

You should see that the p-value is the same as we got from the unpaired t-test.

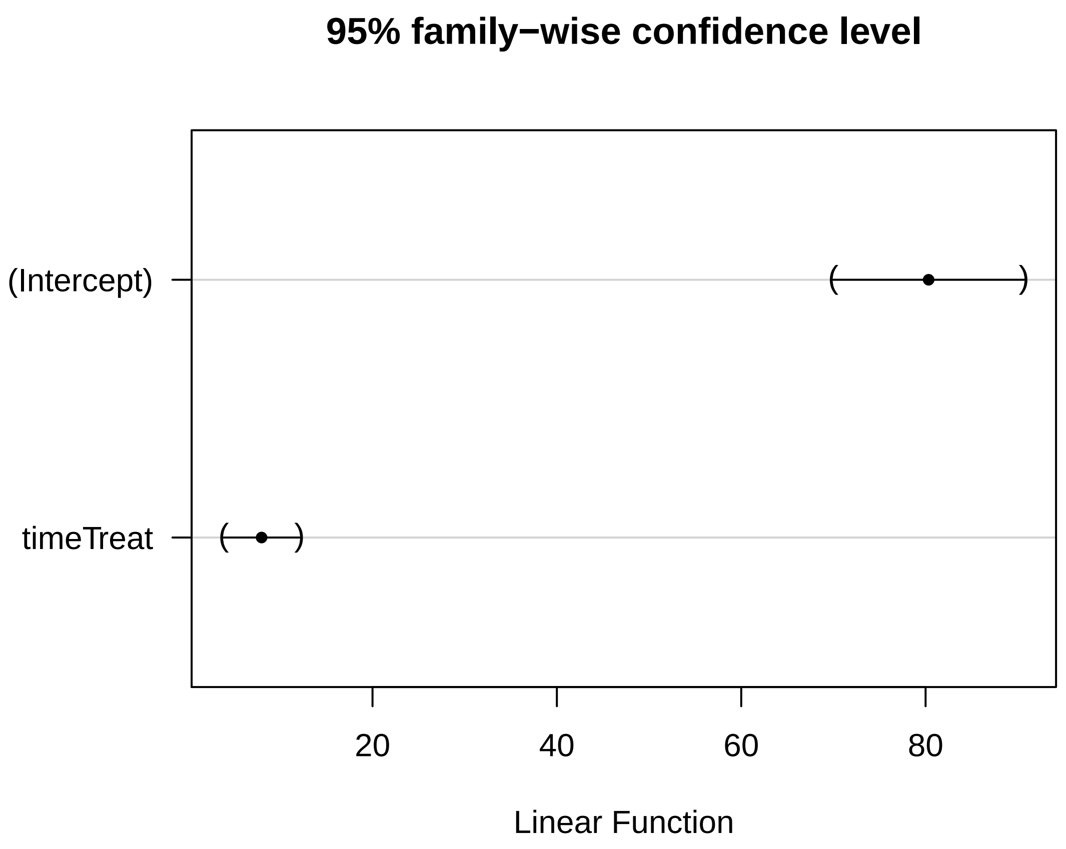
Now making rat id a random effect.

```
mod<-lmer(resp~time+(1|id),data=d)
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: resp ~ time + (1 | id)
##    Data: d
##
## REML criterion at convergence: 135.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.42938 -0.49183  0.06957  0.55204  1.08978
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 197.86   14.066
##  Residual              17.01    4.124
## Number of obs: 20, groups:  id, 10
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   80.329      4.635  9.740  17.329 1.21e-08 ***
## timeTreat      7.970      1.844  9.000   4.321  0.00193 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Correlation of Fixed Effects:
##           (Intr)
## timeTreat -0.199
```

```
par(mar=c(4,8,4,2))
plot(glht(mod))
```



Now the p-value is the same as that obtained by a paired t-test. Alternatively code this achieve the same result.

```
mod<-aov(resp~time+Error(id),data=d)
summary(mod)
```

```
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  9   3715   412.7
##
## Error: Within
##           Df Sum Sq Mean Sq F value  Pr(>F)
## time       1  317.6   317.6   18.67 0.00193 **
## Residuals  9  153.1    17.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

You should also be able to see the fundamental similarity between this situation and one involving blocking with respect to the specification of the model. However in a classic block design the blocks are considered to be **groups** of experimental units rather than individual experimental units. This can lead to confusion and model mispecification if you are not careful. A block design must have repeated measures with different levels of treatment **within** each block, just as a repeat measure design has different treatment levels within each subject. These simple examples can be extended to produce more complex designs.

10.3 Repeat measures with subsampling

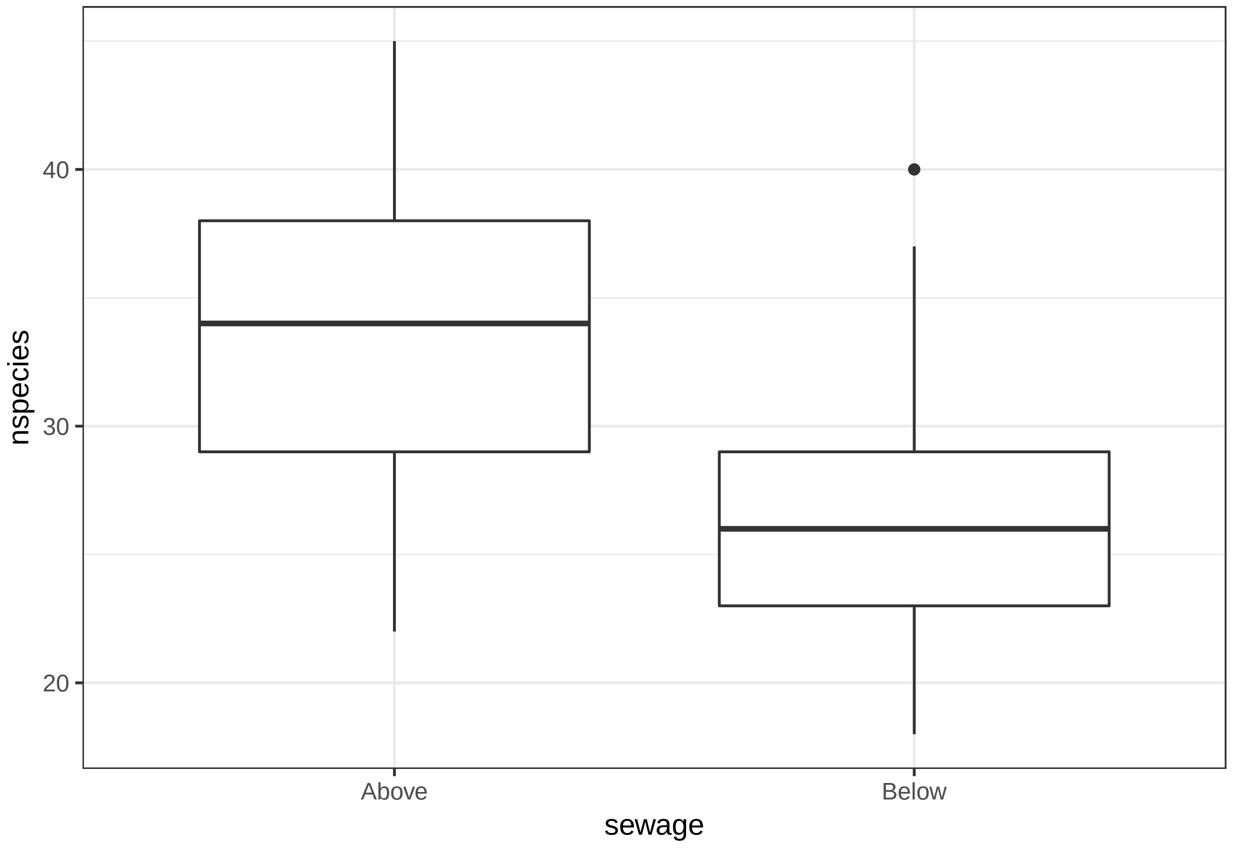
One way the repeat measures design can be extended is with sub sampling. Take this example. A researcher is interested in the impact of sewage treatment plants outfalls on the diversity of river invertebrates. In order to produce a paired design kick samples are taken 100m above the outfall and 100m below the outfall in five rivers. This leads to a total of ten sites. At each site three kicks are obtained. These three measurements are thus sub-samples.

```r
d<-read.csv("https://tinyurl.com/aqm-data/river.csv")
g0<-ggplot(d,aes(x=sewage,y=nspecies))
```

10.3.1 Visualising the data

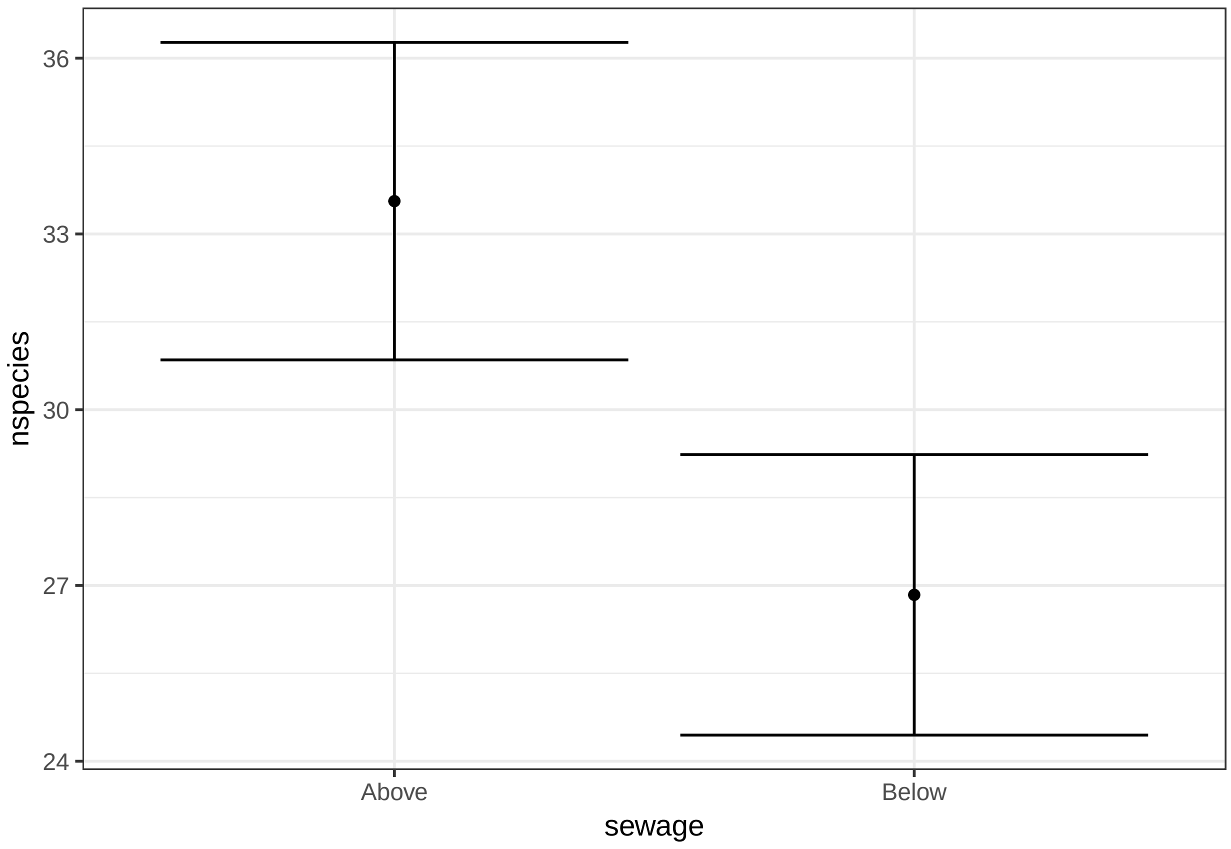
If we pool all the data points taken below and above the sewage outfall we would not be taking into account the variability between rivers. We would also not be taking into account the potential for "pseudo-replication" as a result of the sub-sampling.

```r
g0+geom_boxplot()
```



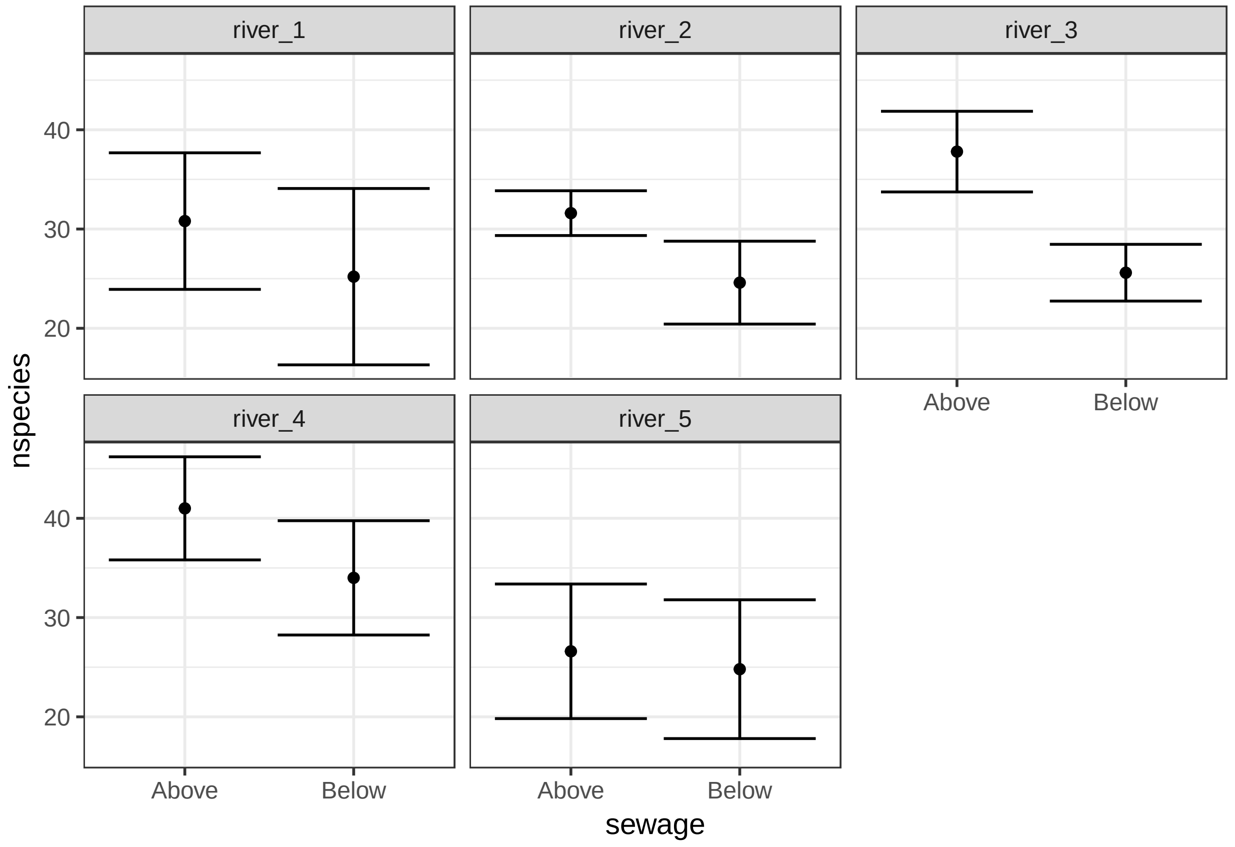
We can still obtain confidence intervals, but they would be based on a simplistic model that does not take into account the true data structure.

```
g1<-g0+stat_summary(fun.data=mean_cl_normal,geom="point")+stat_summary(fun.data=mean_cl_normal,geom="er
g1
```



We could try splitting the data by river in order to visualise the pattern of differences.

```
g1+facet_wrap(~id)
```



10.3.2 Incorrect model specification.

We could think of river id as a fixed effect. This may be reasonable if we are interested in differences between rivers. However it does not take into account the fact that the data consists of sub samples at each site. So the model below would be wrong.

```
mod<-lm(nspecies~sewage+id,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: nspecies
##           Df Sum Sq Mean Sq F value    Pr(>F)
## sewage     1 564.48  564.48 25.0596 9.454e-06 ***
## id         4 850.40  212.60  9.4382 1.344e-05 ***
## Residuals 44 991.12   22.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

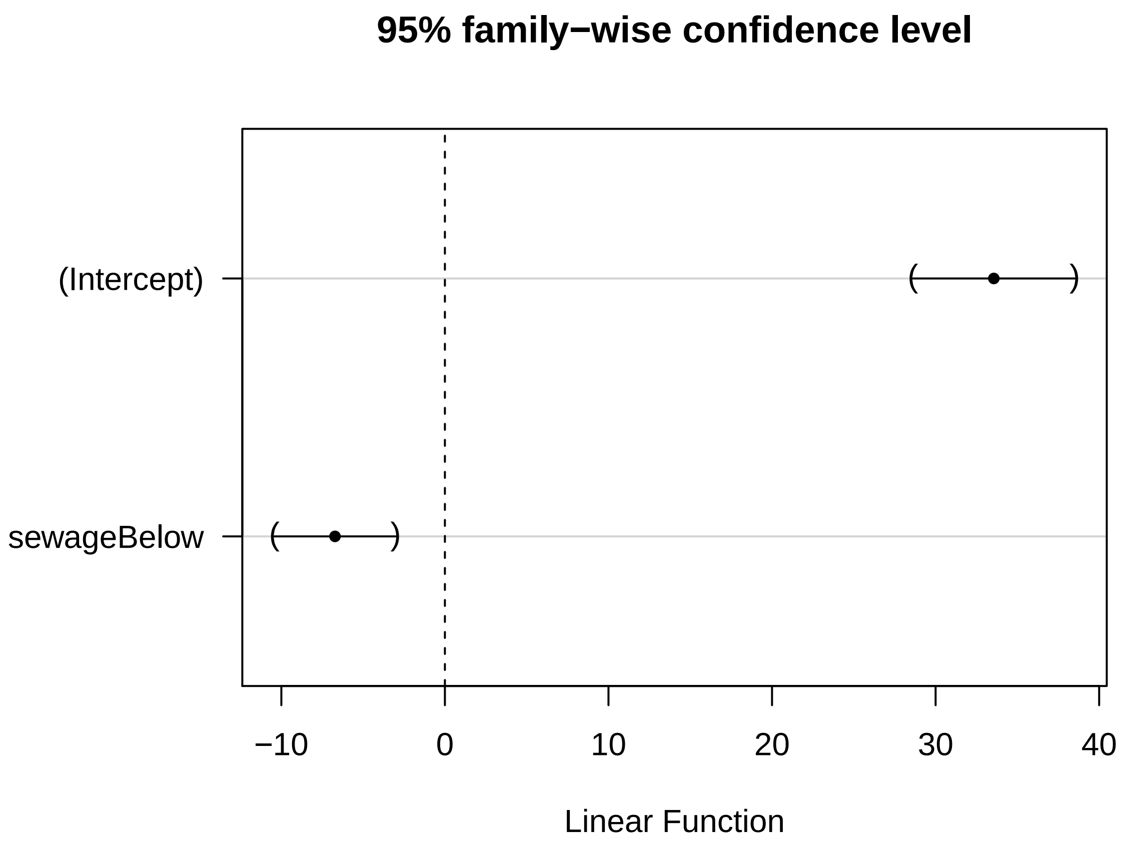
10.3.3 Mixed effect model

The data could be modelled as nested random effects. There is random variability between rivers and there is also random variability between sites at each river. The fixed effect that we are most interested in concerns whether the site is above or below the sewage outfall.

```
mod<-lmer(nspecies~sewage+(1|id/site),data=d)
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: nspecies ~ sewage + (1 | id/site)
##    Data: d
##
## REML criterion at convergence: 300.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.59547 -0.71758 -0.05326  0.60059  2.36669
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  site:id  (Intercept)  2.694   1.641
##  id       (Intercept) 17.789   4.218
##  Residual             21.300   4.615
## Number of obs: 50, groups:  site:id, 10; id, 5
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   33.560      2.225  5.272  15.086 1.54e-05 ***
## sewageBelow   -6.720      1.668  4.001  -4.029   0.0157 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## sewageBelow -0.375
```

```
par(mar=c(4,8,4,2))
plot(glht(mod))
```



We could also treat river as a fixed effect if we are specifically interested in differences between rivers.

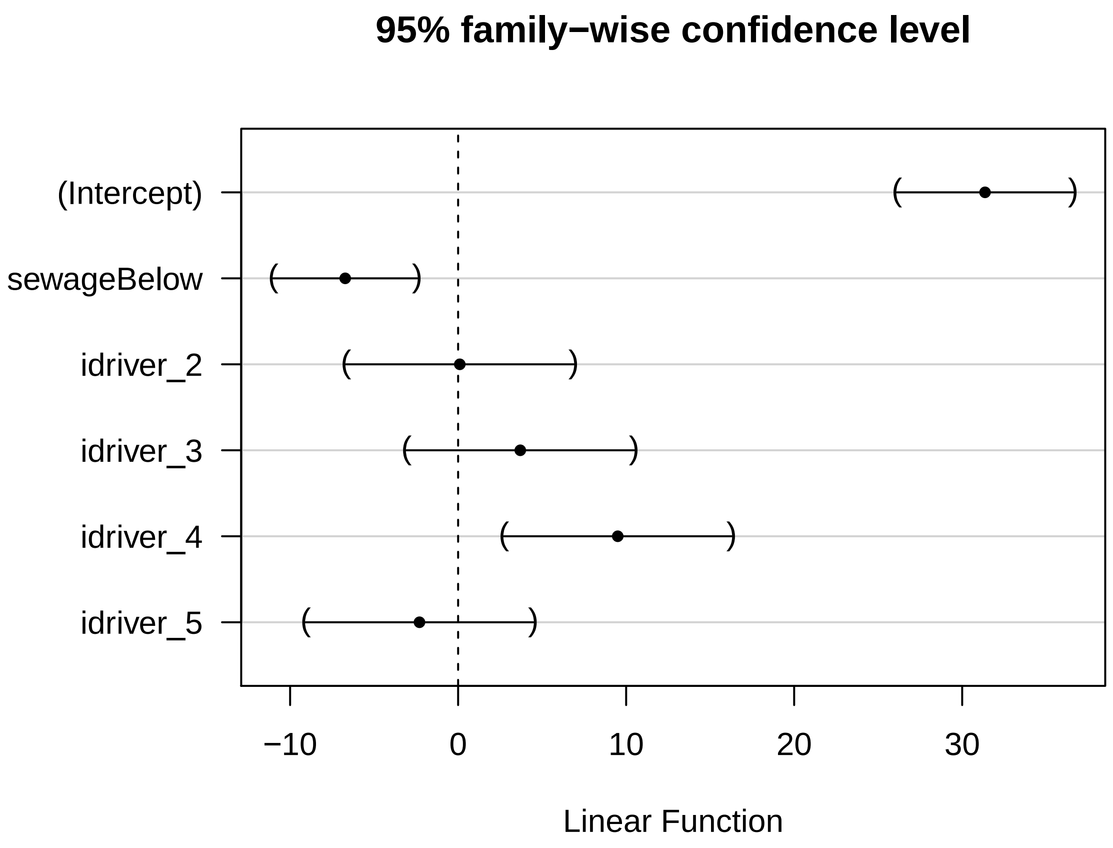
```
mod<-lmer(nspecies~sewage+id+(1|site),data=d)
anova(mod)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##        Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
## sewage  345.7   345.7     1     4 16.2300 0.01575 *
## id      520.8   130.2     4     4  6.1127 0.05374 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: nspecies ~ sewage + id + (1 | site)
##    Data: d
##
## REML criterion at convergence: 275.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.54770 -0.70432 -0.04561  0.61090  2.41441
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  site     (Intercept)  2.696   1.642
##  Residual             21.300   4.615
## Number of obs: 50, groups:  site, 10
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   31.360      2.043  4.000  15.350 0.000105 ***
## sewageBelow   -6.720      1.668  4.000  -4.029 0.015751 *
## idriver_2      0.100      2.637  4.000   0.038 0.971572
## idriver_3      3.700      2.637  4.000   1.403 0.233304
## idriver_4      9.500      2.637  4.000   3.602 0.022718 *
## idriver_5     -2.300      2.637  4.000  -0.872 0.432392
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) swgBlw idrv_2 idrv_3 idrv_4
## sewageBelow -0.408
## idriver_2   -0.645  0.000
## idriver_3   -0.645  0.000  0.500
## idriver_4   -0.645  0.000  0.500  0.500
## idriver_5   -0.645  0.000  0.500  0.500  0.500
```

```
par(mar=c(4,8,4,2))
plot(glht(mod))
```



Notice that both models provide the same p-value for the effect of the sewage outfalls.

Using nested random effects produces an almost identical result to that which would be obtained by taking the means of the sub samples, as we have seen in previous examples.

```
# dd<-melt(d,m="nspecies") reshape method of aggregation
# dd1<-cast(dd,id+sewage~variable,mean)
## Dplyr method
d %>% group_by(id,sewage) %>% summarise(nspecies=mean(nspecies)) ->dd
mod<-lm(nspecies~id+sewage,data=dd)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: nspecies
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## id         4 170.080  42.520  6.1127 0.05374 .
## sewage     1 112.896 112.896 16.2300 0.01575 *
## Residuals  4  27.824   6.956
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chapter 11

Factorial designs

Factorial designs refer to situations in which more than one factor is of interest. There are various reasons to use more than one factor in an experiment. In some cases it may save time and money to measure the effects of two factors simultaneously. This makes sense if the factors are additive, with no interactions. In other situations we may actually be interested in looking for interactions between two factors. Altering the level of one factor may alter the effect of another.

Let's look at an example. An experimenter was interested in the effect of irrigation on the yield of two varieties of wheat. The design involved assigning variety and irrigation treatment at random to 24 plots in order to have 6 replicates of each combination of treatment levels (v1:i1,v1:i2,v2:i1 and v2:i2)

For the moment let's ignore the possibilities of setting up the experiment within blocks and assume that the experimental units are uniform prior to treatment and are independent of each other.

```r
d<-read.csv("https://tinyurl.com/aqm-data/factorial.csv")
```

We can look at the results as box plots by conditioning on one of the factors using facet wrapping.

```r
g0<-ggplot(d,aes(x=irrigation,y=yield))
g0+geom_boxplot()+facet_wrap(~variety)
```

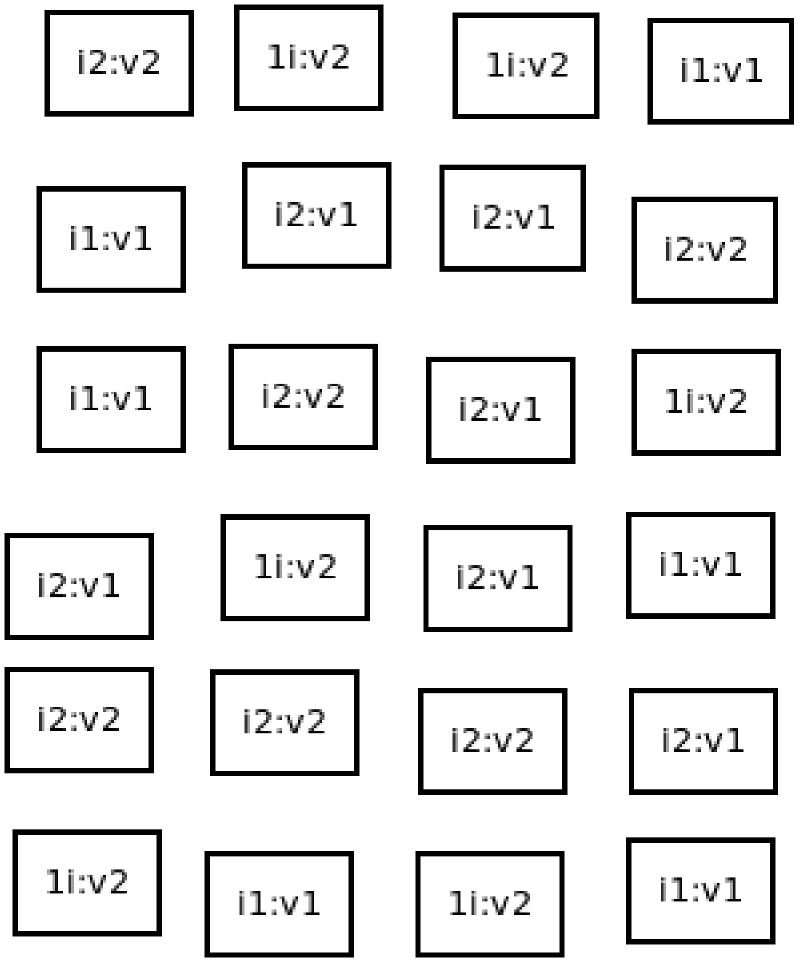
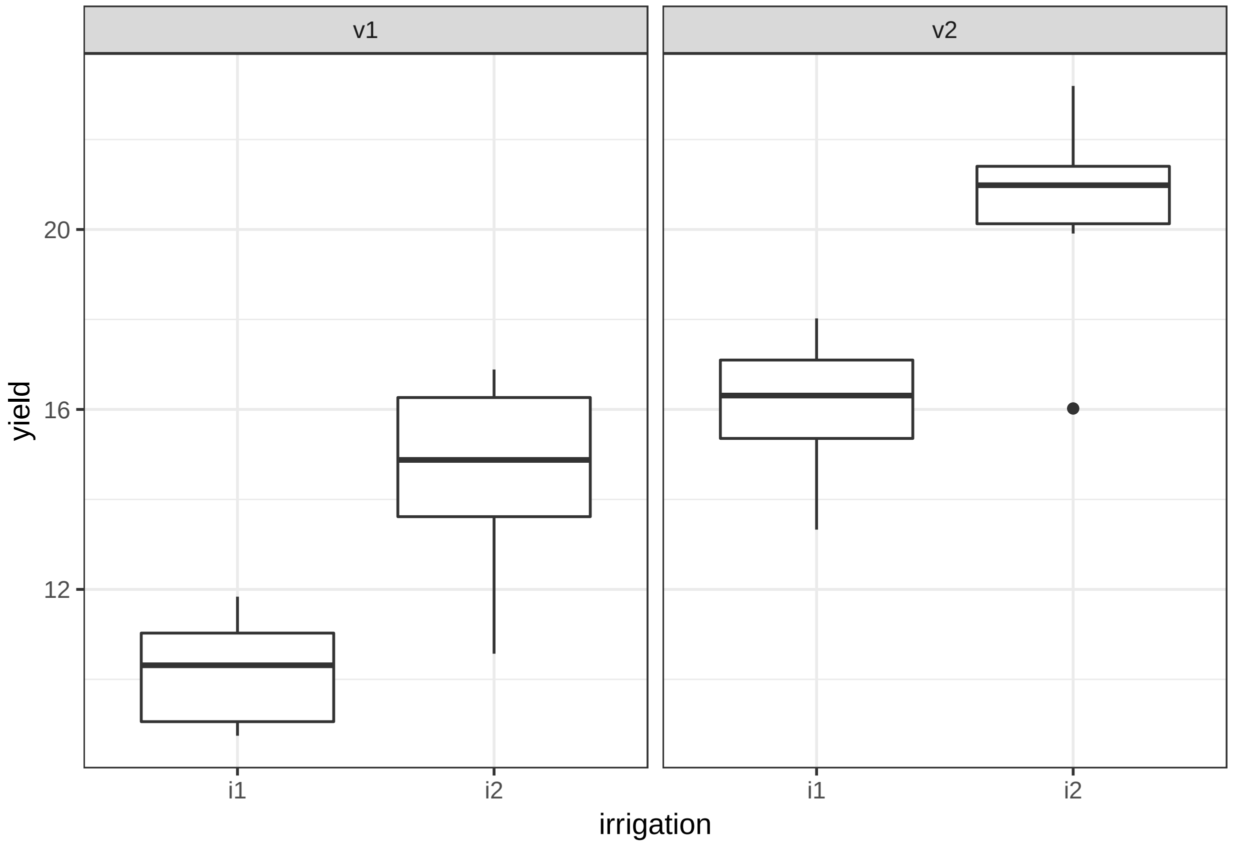
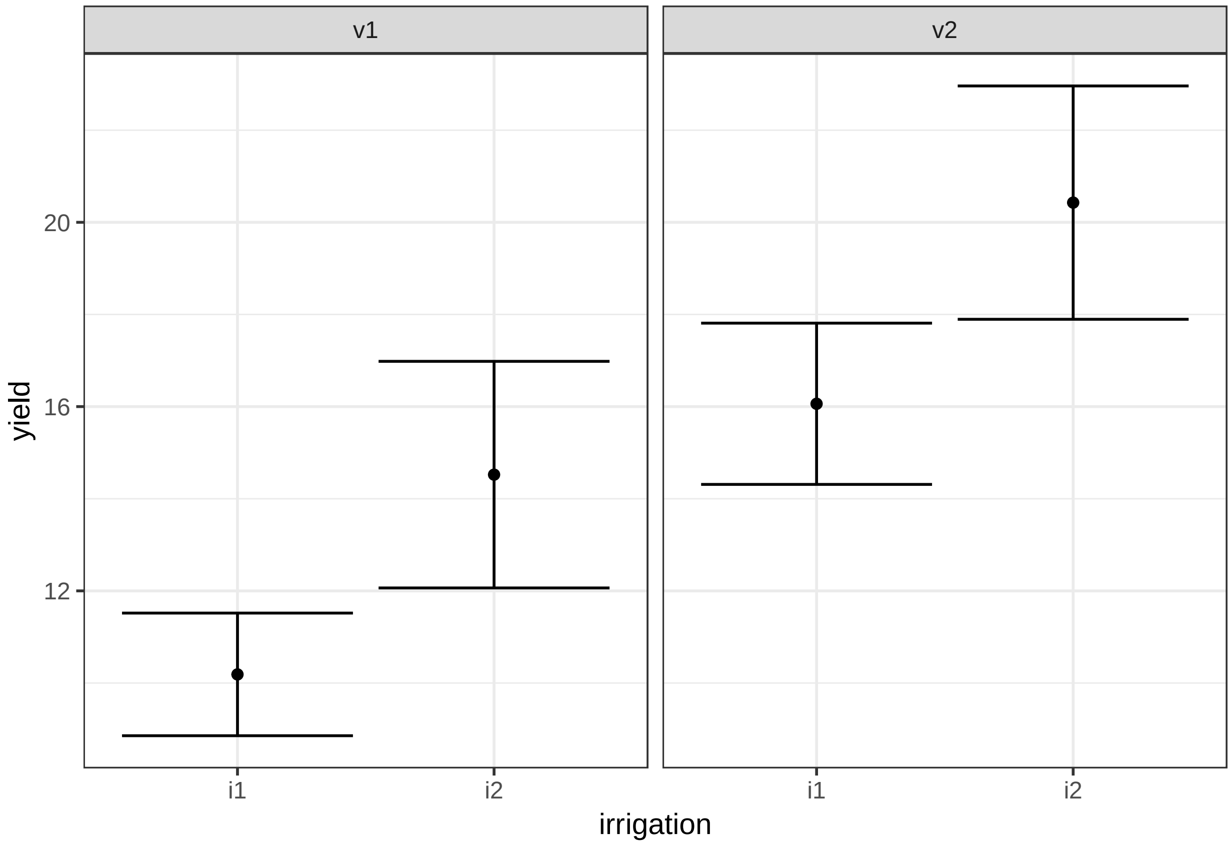

i2:v2
1i:v2
1i:v2
i1:v1
i1:v1
i2:v1
i2:v1
i2:v2
i1:v1
i2:v2
i2:v1
1i:v2
i2:v1
1i:v2
i2:v1
i1:v1
i2:v2
i2:v2
i2:v2
i2:v1
1i:v2
i1:v1
1i:v2
i1:v1

Figure 11.1:



We can also look at confidence intervals.

```
g1<-g0+stat_summary(fun.data=mean_cl_normal,geom="point")
g1<-g1+stat_summary(fun.data=mean_cl_normal,geom="errorbar",colour="black")
g1+facet_wrap(~variety)
```



11.1 Model fitting

There are no random effects in this design, so a simple linear model can be used. Typing an asterix (*) in the model formula fits a model with main effects and an interaction.

```
mod<-lm(yield~irrigation*variety,data=d)
anova(mod)
```

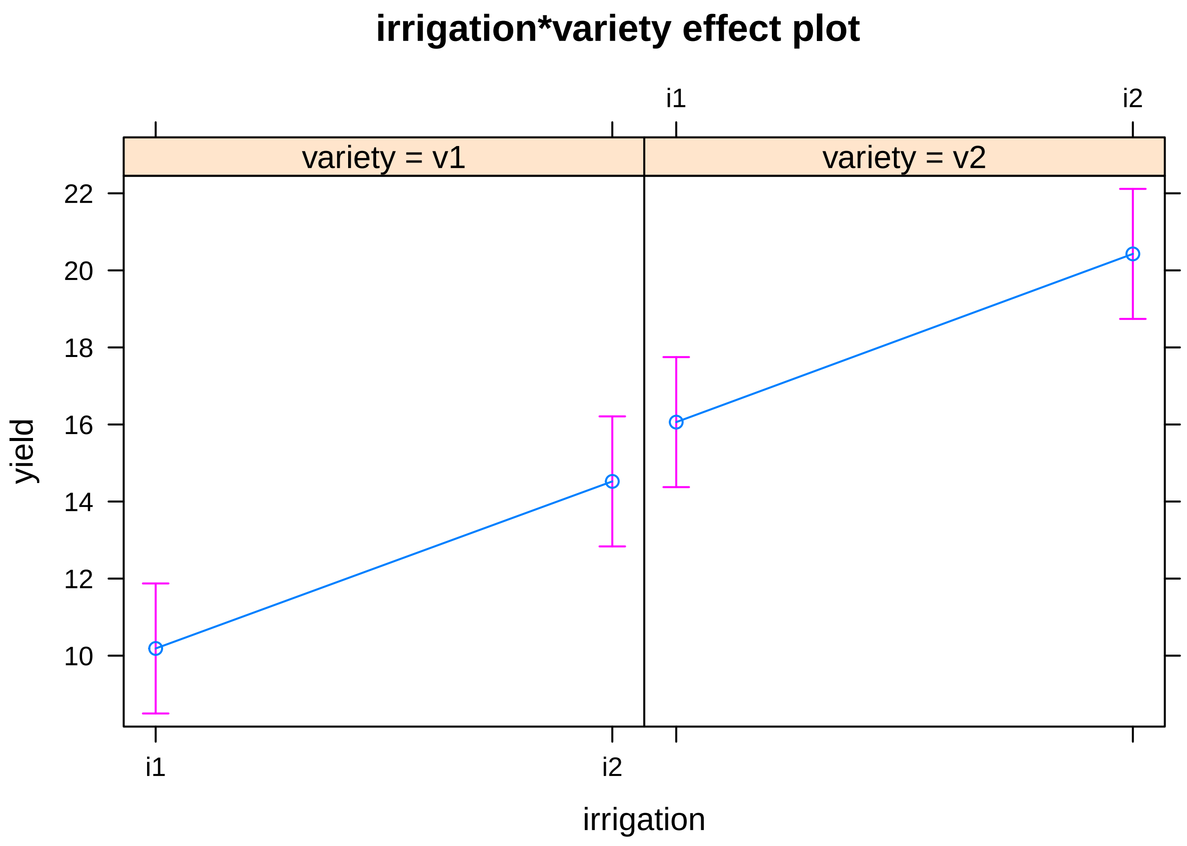
```
## Analysis of Variance Table
##
## Response: yield
##                    Df  Sum Sq Mean Sq F value    Pr(>F)
## irrigation          1 113.600 113.600 28.9293 2.897e-05 ***
## variety             1 208.128 208.128 53.0020 4.845e-07 ***
## irrigation:variety  1   0.001   0.001  0.0003    0.9854
## Residuals          20  78.536   3.927
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = yield ~ irrigation * variety, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4064 -0.9752  0.4121  1.0838  2.7629
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            10.18679    0.80899  12.592 5.78e-11 ***
## irrigationi2            4.33623    1.14409   3.790  0.00115 **
## varietyv2               5.87464    1.14409   5.135 5.05e-05 ***
## irrigationi2:varietyv2  0.03003    1.61798   0.019  0.98538
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.982 on 20 degrees of freedom
## Multiple R-squared:  0.8038, Adjusted R-squared:  0.7744
## F-statistic: 27.31 on 3 and 20 DF,  p-value: 2.836e-07
```

You should be able to see that the interaction term in this case is not significant. The effects package allows us to look at the pattern of effects visually.

```
e <- Effect(c("irrigation","variety"), mod)
plot(e)
```



You can see that the lines between the two levels of the factors follow the same slope but are moved up. This is indicative of additive effects. We could therefore fit a simpler model

```
mod<-lm(yield~irrigation+variety,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
```

```
## Response: yield
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## irrigation  1 113.600  113.60  30.375 1.811e-05 ***
## variety     1 208.128  208.13  55.651 2.478e-07 ***
## Residuals  21  78.537    3.74
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = yield ~ irrigation + variety, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3989 -0.9827  0.4196  1.0876  2.7704
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10.1793     0.6837  14.888 1.24e-12 ***
## irrigationi2    4.3512     0.7895   5.511 1.81e-05 ***
## varietyv2       5.8897     0.7895   7.460 2.48e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.934 on 21 degrees of freedom
## Multiple R-squared:  0.8038, Adjusted R-squared:  0.7851
## F-statistic: 43.01 on 2 and 21 DF,  p-value: 3.747e-08
```

11.2 Experiment with interactions

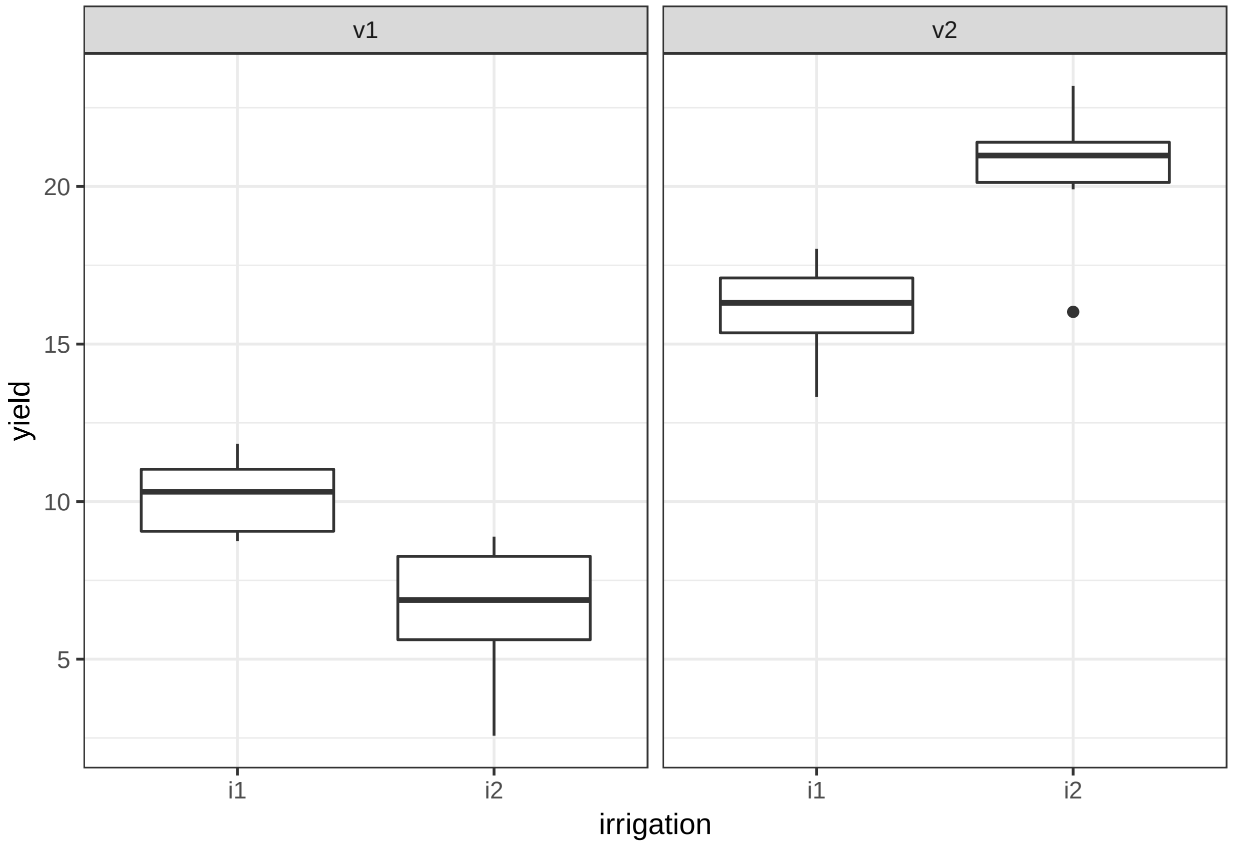
Let's look at data from the same experimental set up, but in this case there **is** an interaction.

```
d<-read.csv("https://tinyurl.com/aqm-data/factorial2.csv")
head(d)
```

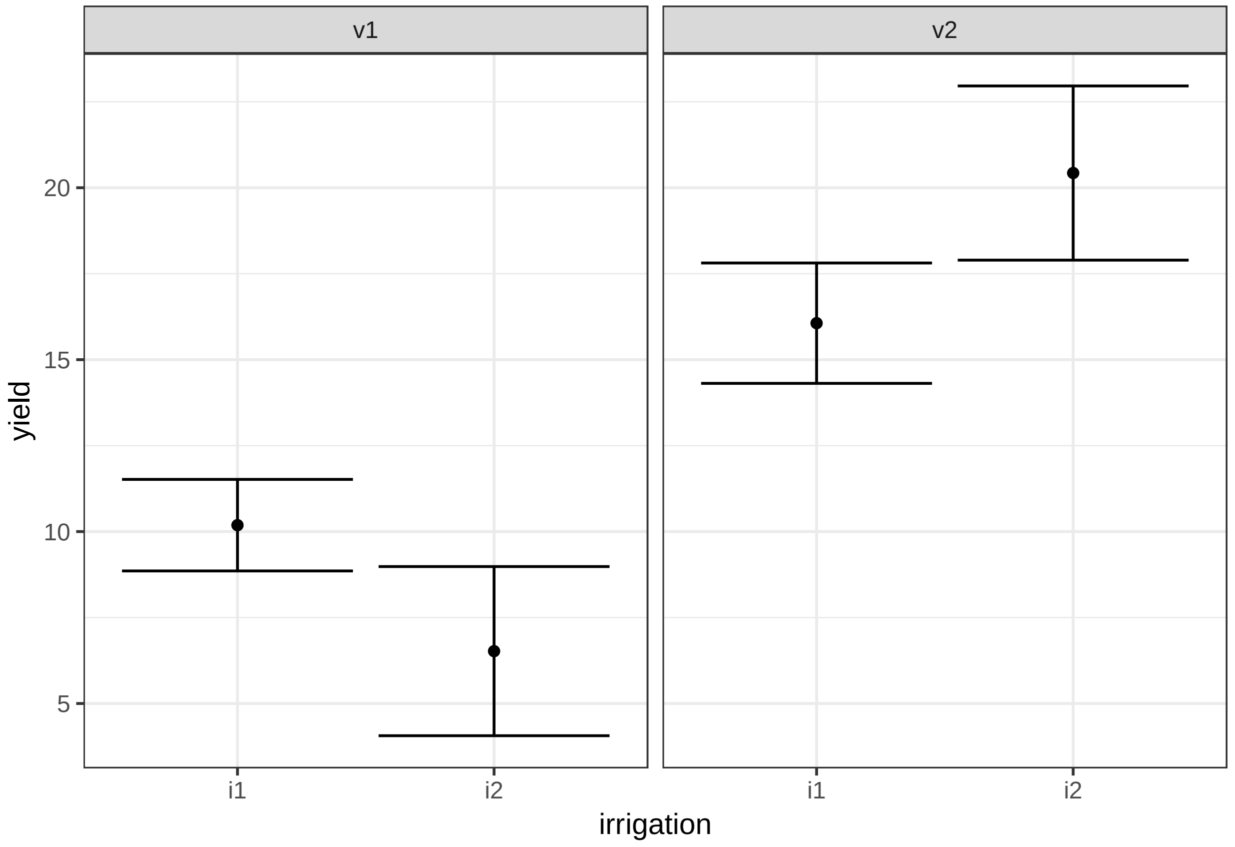
```
##   variety irrigation     yield
## 1      v1         i1  8.747092
## 2      v1         i2  7.367287
## 3      v2         i1 13.328743
## 4      v2         i2 23.190562
## 5      v1         i1 10.659016
## 6      v1         i2  5.359063
```

The pattern is apparent in the box plots.

```
g0<-ggplot(d,aes(x=irrigation,y=yield))
g0+geom_boxplot()+facet_wrap(~variety)
```



```
g1<-g0+stat_summary(fun.data=mean_cl_normal,geom="point")
g1<-g1+stat_summary(fun.data=mean_cl_normal,geom="errorbar",colour="black")
g1+facet_wrap(~variety)
```



Now if we fit a model we will find a very significant interaction.

```r
mod<-lm(yield~irrigation*variety,data=d)
anova(mod)
```

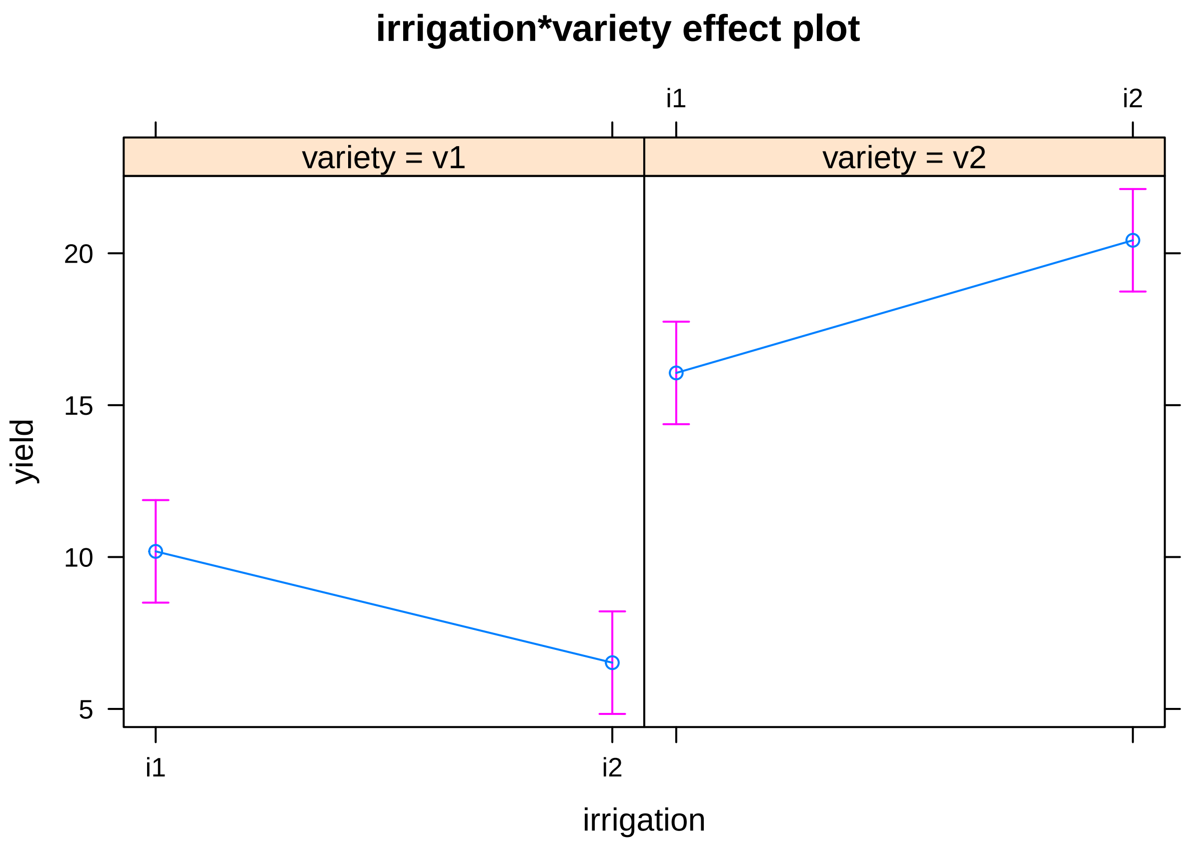
```
## Analysis of Variance Table
##
## Response: yield
##                    Df Sum Sq Mean Sq  F value    Pr(>F)
## irrigation          1   0.74    0.74   0.1885    0.6688
## variety             1 586.83  586.83 149.4427 9.789e-11 ***
## irrigation:variety  1  96.72   96.72  24.6313 7.484e-05 ***
## Residuals          20  78.54    3.93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mod)
```

```
##
## Call:
## lm(formula = yield ~ irrigation * variety, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4064 -0.9752  0.4121  1.0838  2.7629
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                10.187      0.809  12.592 5.78e-11 ***
## irrigationi2            -3.664      1.144  -3.202  0.00447 **
## varietyv2                5.875      1.144   5.135 5.05e-05 ***
## irrigationi2:varietyv2   8.030      1.618   4.963 7.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.982 on 20 degrees of freedom
## Multiple R-squared:  0.897,  Adjusted R-squared:  0.8816
## F-statistic: 58.09 on 3 and 20 DF,  p-value: 4.713e-10
```

The first variety does not respond positively to irrigation. In fact yield may be reduced through over watering. In this case the effects are not additive. They are conditional on the level of one of the factors. In this situation the main effects of irrigation or variety can be difficult to interpret as they represent the average effect taken over both levels of the other factor.

```
e <- Effect(c("irrigation","variety"), mod)
plot(e)
```



Interactions are very common in many ecological situations. Although some textbooks on experimental design treat interactive effects as undesirable, finding interactions is an interesting result. In this case the discussion of the results could concentrate on finding an explanation for the difference in response between the two varieties.

11.3 Full factorial with blocking

The previous example treated each experimental unit as independent. In many situations there is some spatial dependence between experimental units.

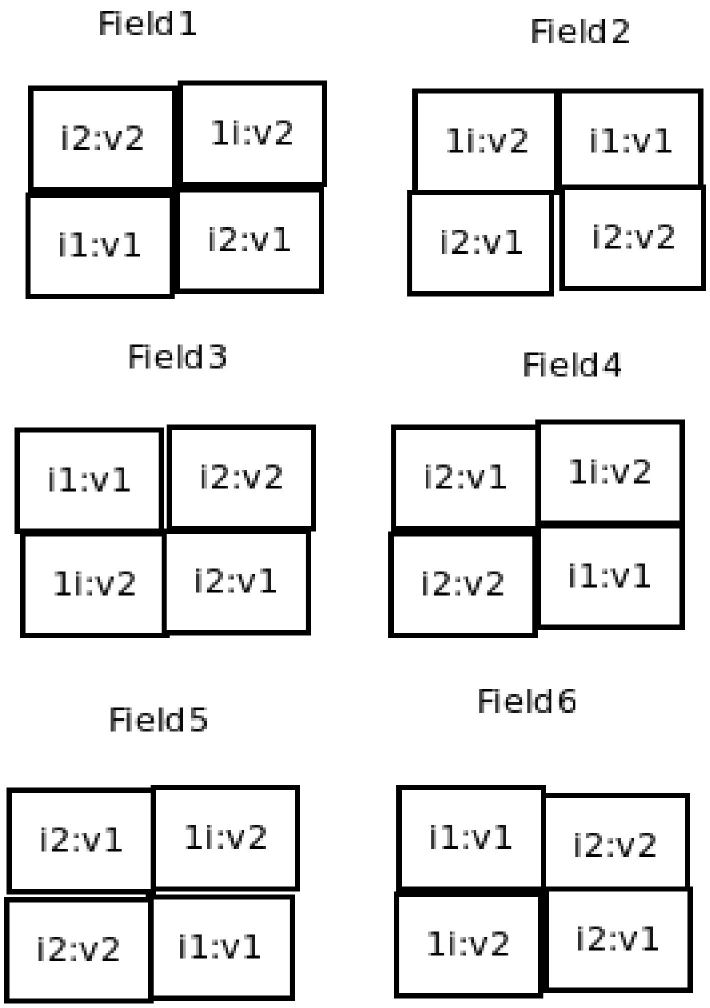


Figure 11.2: alt text

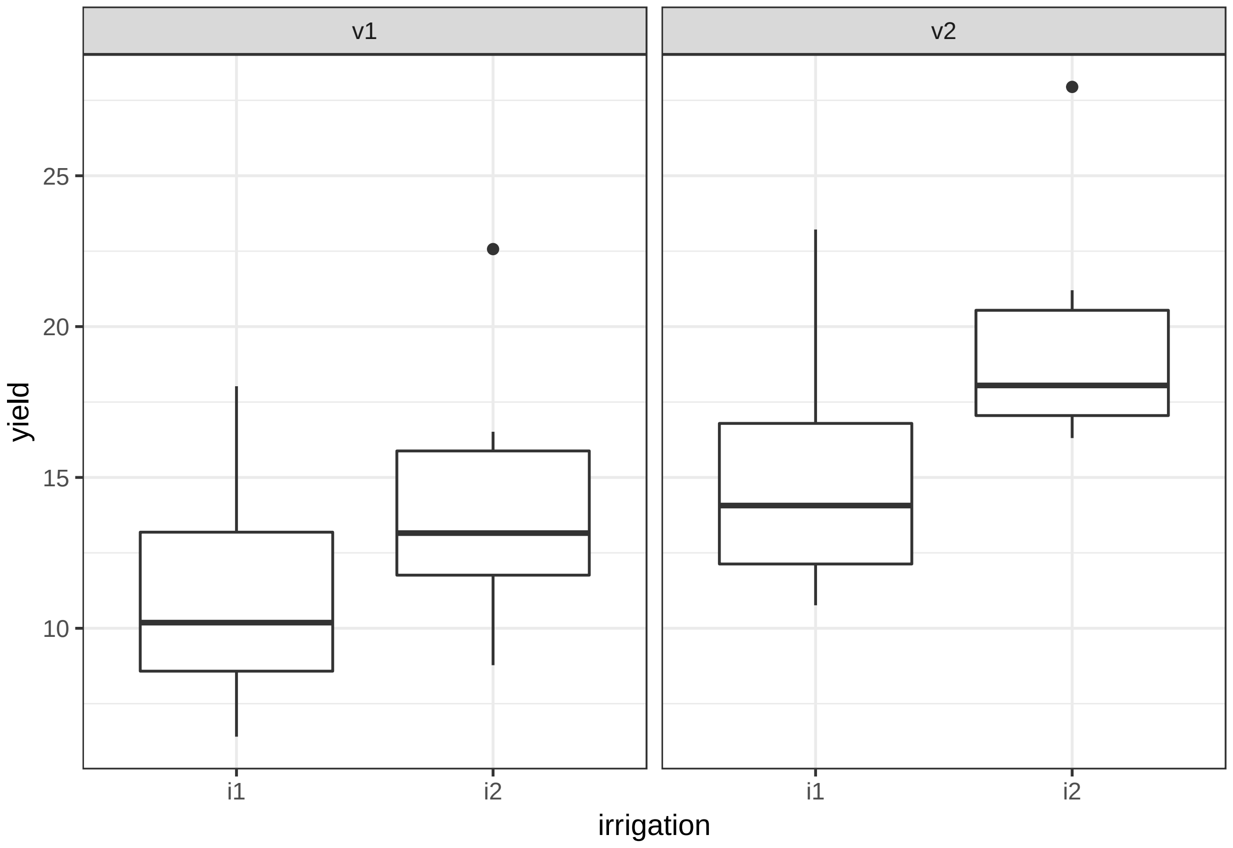
```r
d<-read.csv("https://tinyurl.com/aqm-data/factorial_block.csv")
head(d)
```

```
##   variety   field field_effect irrigation     yield
## 1      v1 field_1   -2.5058152         i1  8.469043
## 2      v1 field_1   -2.5058152         i2 13.970834
## 3      v2 field_1   -2.5058152         i1 13.645747
## 4      v2 field_1   -2.5058152         i2 16.883408
## 5      v1 field_2    0.7345733         i1 13.758136
## 6      v1 field_2    0.7345733         i2 16.514260
```

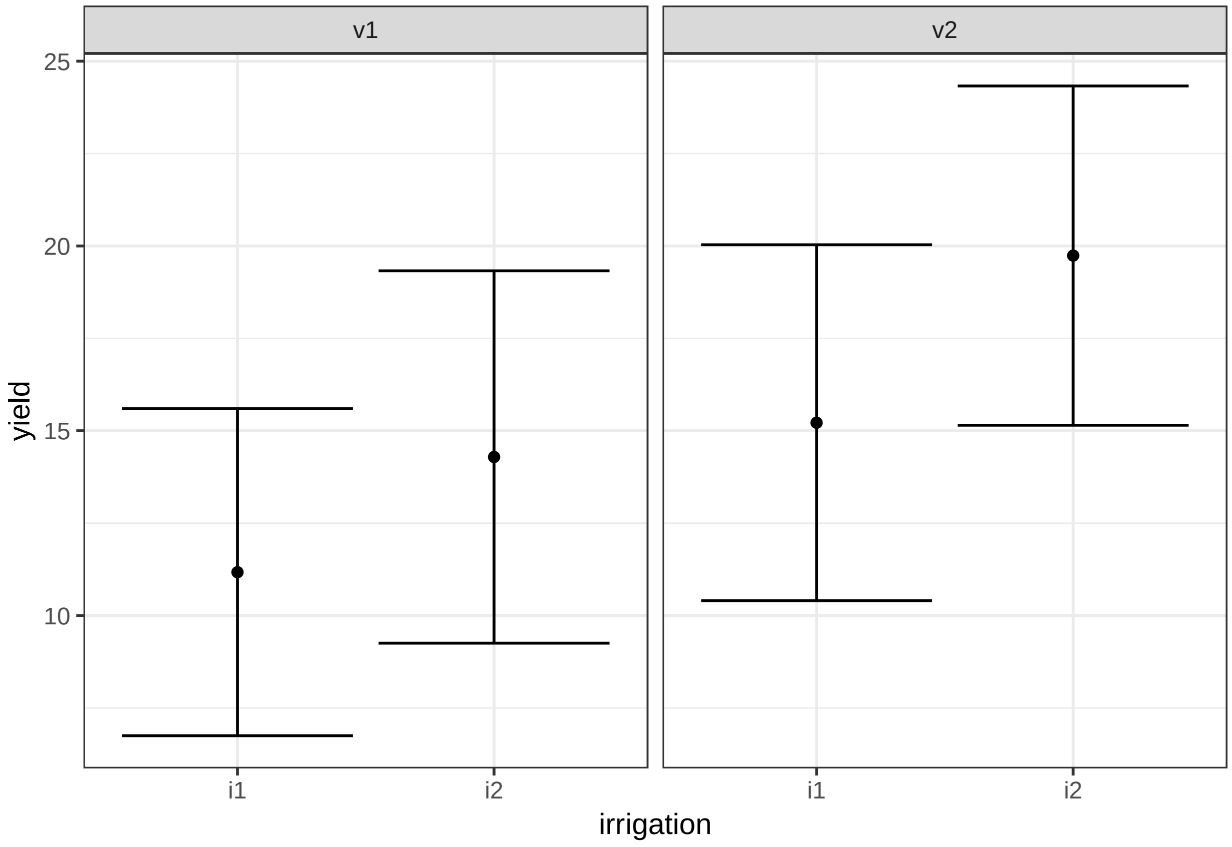
Once again, the potential advantage of taking into account blocking is to increase the power of the analysis. If the intrinsic variability between blocks is contributing to the variability in the response of the experimental units then accounting for it in the statistical model will increase power.

This can be seen in this example by first plotting the raw data.

```r
g0<-ggplot(d,aes(x=irrigation,y=yield))
g0+geom_boxplot()+facet_wrap(~variety)
```



```r
g1<-g0+stat_summary(fun.data=mean_cl_normal,geom="point")
g1<-g1+stat_summary(fun.data=mean_cl_normal,geom="errorbar",colour="black")
g1+facet_wrap(~variety)
```



11.3.1 Model not taking into account blocks

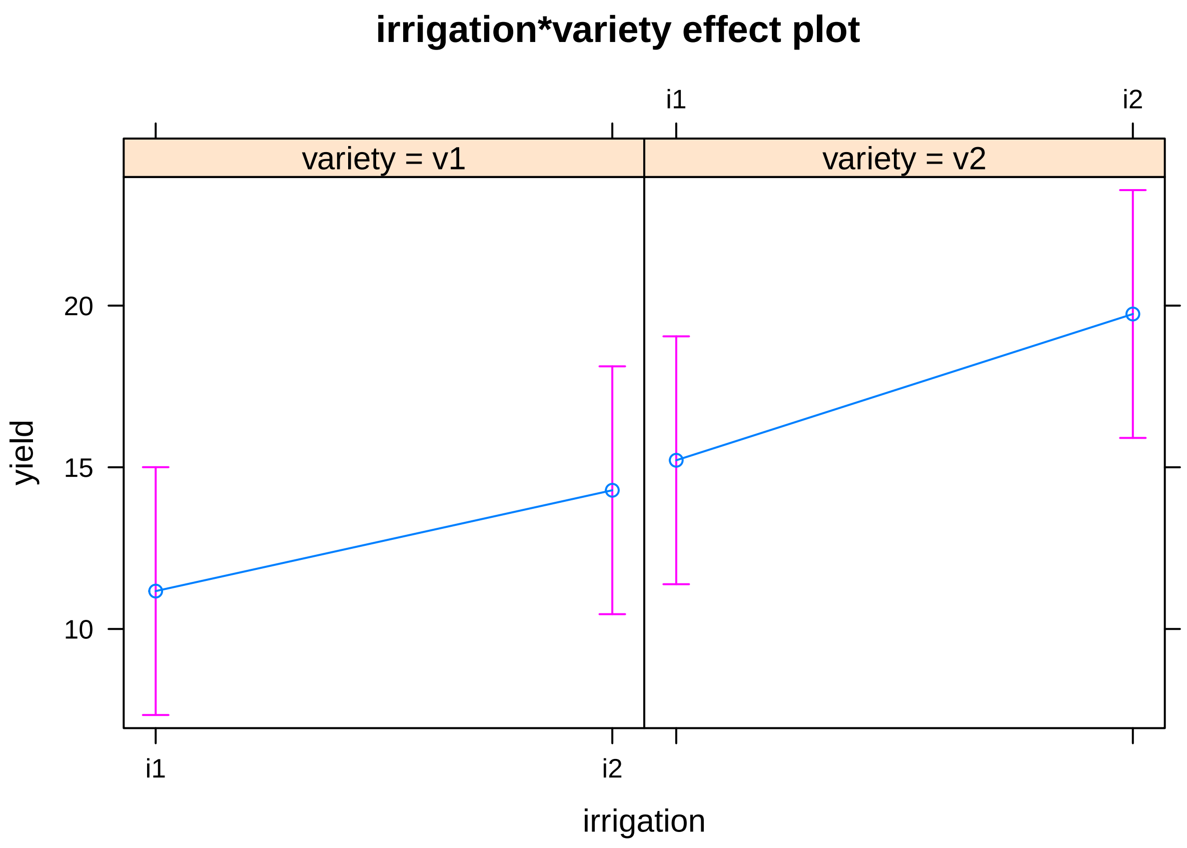
If we don't take into account the variability attributable to the blocks in the model we lose statistical power.

```
mod<-lm(yield~variety*irrigation,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: yield
##                    Df Sum Sq Mean Sq F value  Pr(>F)
## variety             1 135.24 135.238  6.6790 0.01772 *
## irrigation          1  87.57  87.567  4.3246 0.05064 .
## variety:irrigation  1   2.96   2.962  0.1463 0.70614
## Residuals          20 404.97  20.248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Effects plot.

```
e <- Effect(c("irrigation","variety"), mod)
plot(e)
```



11.3.2 Model with block as a random effect.

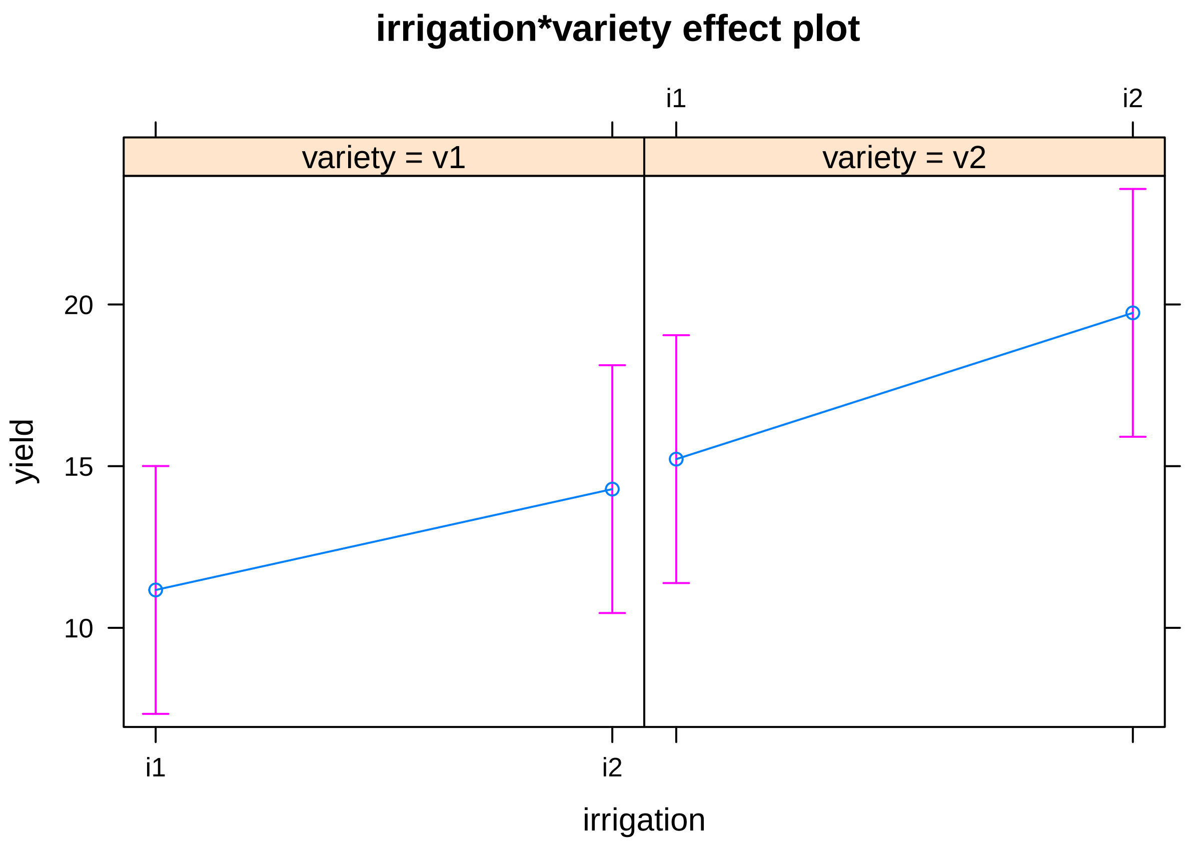
Block can be fitted as a random effect using lmer.

```r
mod<-lmer(yield~variety*irrigation+(1|field),data=d)
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: yield ~ variety * irrigation + (1 | field)
##    Data: d
##
## REML criterion at convergence: 103
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.98922 -0.49404 -0.04189  0.55337  1.34688
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  field    (Intercept) 16.991   4.122
##  Residual              3.257   1.805
## Number of obs: 24, groups:  field, 6
##
## Fixed effects:
##                          Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)                11.172      1.837  6.426   6.082 0.000695 ***
## varietyv2                  4.045      1.042 15.000   3.882 0.001474 **
## irrigationi2               3.118      1.042 15.000   2.992 0.009119 **
## varietyv2:irrigationi2     1.405      1.474 15.000   0.954 0.355380
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) vrtyv2 irrgt2
## varietyv2   -0.284
## irrigation2 -0.284  0.500
## vrtyv2:rrg2  0.201 -0.707 -0.707
```

Now notice how the effects plot shows more effects of the treatment.

```
e <- Effect(c("irrigation","variety"), mod)
plot(e)
```



11.4 Split plot

In all the previous examples there have been alternatives available to the use of mixed models with random and fixed effects. In the case of situations involving sub sampling the dependence between sub samples can be dealt with by taking means for each experimental unit. In the cases in which variability occurs between blocks it is possible to include blocks in a model as a fixed effect. Repeat measures designs can use differences within subjects as responses. However there are some more complex situations in which the sum of squares attributed to the variability attributed to different treatments should be compared with different error terms. A classic example is the split plot design in agriculture. The split plot design arose simply as a form of reducing the cost and difficulty of setting up field experiments. When treating field plots it can

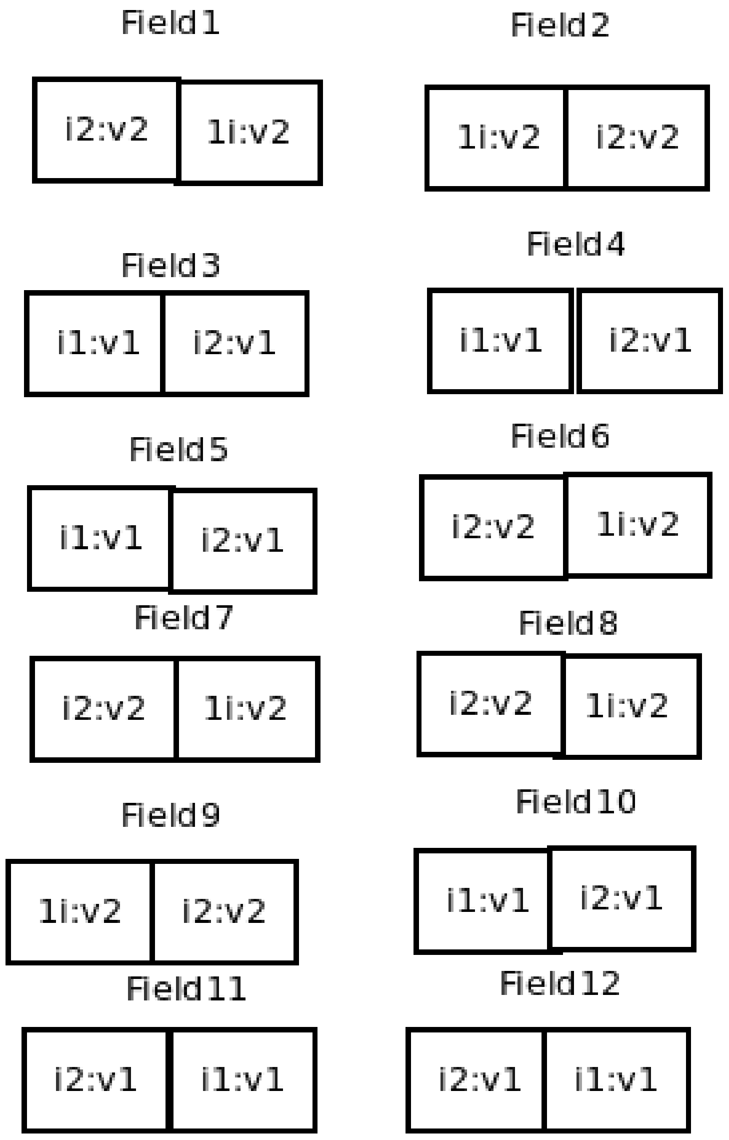


Figure 11.3:

sometimes be much easier to set up some treatments for larger areas than a smaller one. Imagine the case in the yield experiment we have been looking at in which whole fields are planted with single varieties of wheat and each field is split into two forms of irrigation treatment.

In some respects you could argue that not much has changed from the block design. However if each main field has a different response then the error term for variety will have fewer degrees of freedom than that for irrigation as the same variety has been planted in each field which is then "split" into two levels of irrigation. So effectively there is less independent replication of variety than of irrigation.

If this is not taken into account there may be an accusation of "pseudo replication" with respect to the effect of variety as the anlysis would potentially treat all subplots as independent replicates without taking into account the fact that they are nested within field. However this is a problem if there are shared effects at the field level. If it were possible to ensure as much independence between subplots within the same field as between subplots as a whole it wouldn't matter.

Here is a dataframe with results from the split plot experiment shown above.

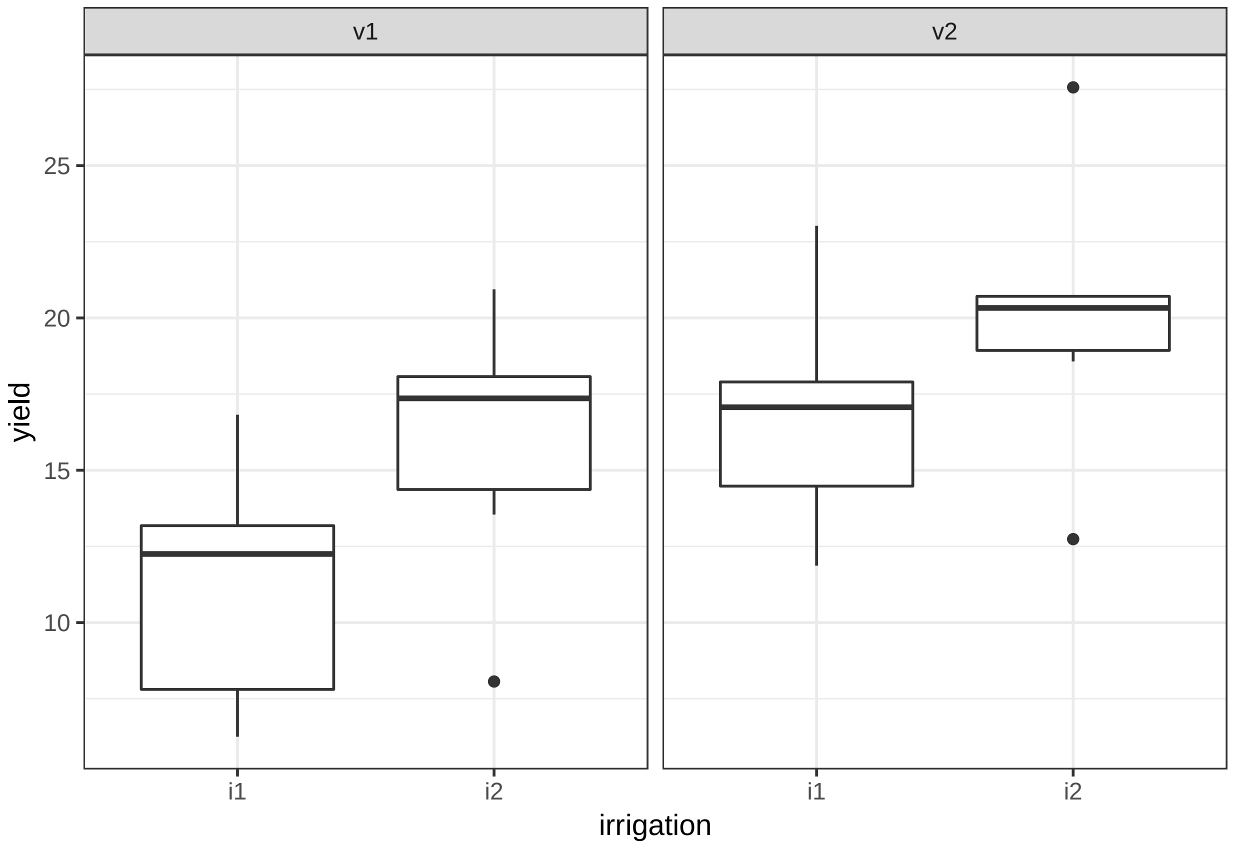
```
d<-read.csv("https://tinyurl.com/aqm-data/split_plot.csv")
str(d)
```

```
## 'data.frame':    24 obs. of  5 variables:
##  $ variety     : Factor w/ 2 levels "v1","v2": 1 1 2 2 1 1 2 2 1 1 ...
##  $ field       : Factor w/ 12 levels "field_1","field_10",..: 1 1 5 5 6 6 7 7 8 8 ...
##  $ field_effect: num  -2.506 -2.506 0.735 0.735 -3.343 ...
##  $ irrigation  : Factor w/ 2 levels "i1","i2": 1 2 1 2 1 2 1 2 1 2 ...
##  $ yield       : num  6.25 8.06 17.98 20.64 6.63 ...
```

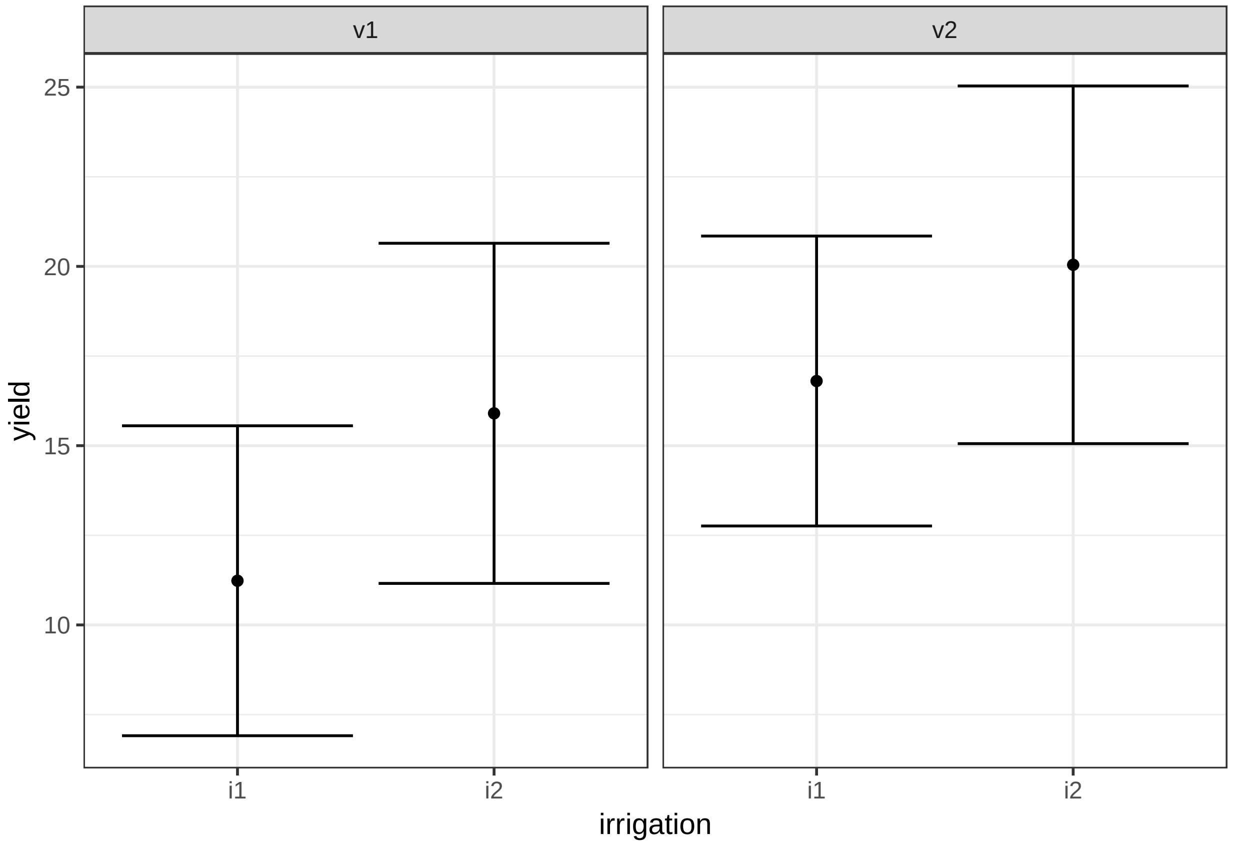
11.4.1 Visualising the data

We can look at the data as boxplots.

```
g0<-ggplot(d,aes(x=irrigation,y=yield))
g0+geom_boxplot()+facet_wrap(~variety)
```



```
g1<-g0+stat_summary(fun.data=mean_cl_normal,geom="point")
g1<- g1+stat_summary(fun.data=mean_cl_normal,geom="errorbar",colour="black")
g1+facet_wrap(~variety)
```



11.4.2 Incorrect model

The naive way of analysing the data would overlook the potential for a field effect.

```
mod<-lm(yield~variety*irrigation,data=d)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: yield
##                    Df Sum Sq Mean Sq F value  Pr(>F)
## variety             1 141.55 141.550  7.5656 0.01233 *
## irrigation          1  93.88  93.875  5.0175 0.03661 *
## variety:irrigation  1   3.06   3.059  0.1635 0.69027
## Residuals          20 374.19  18.710
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

11.4.3 Correct model

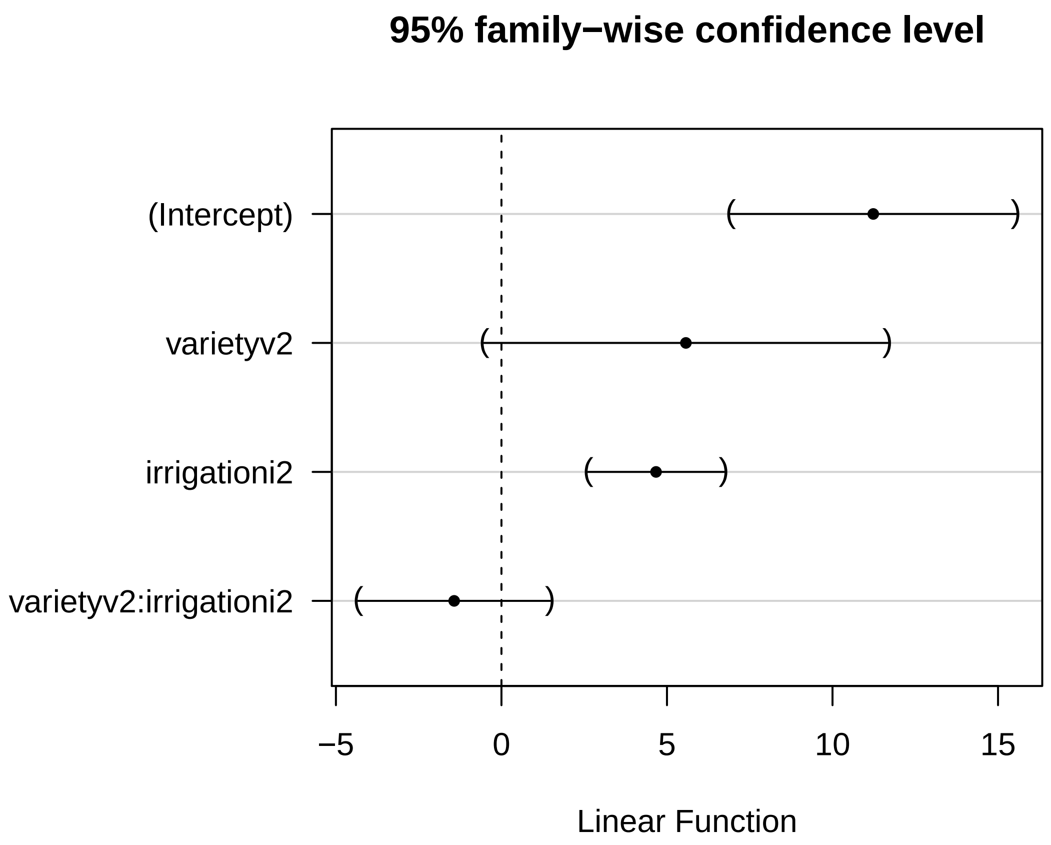
```
mod<-lmer(yield~variety*irrigation+(1|field),data=d)
anova(mod)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                    Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## variety             8.503   8.503     1    10  4.0100   0.07308 .
## irrigation         93.875  93.875     1    10 44.2701 5.684e-05 ***
## variety:irrigation  3.059   3.059     1    10  1.4424   0.25743
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: yield ~ variety * irrigation + (1 | field)
##    Data: d
##
## REML criterion at convergence: 107.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.31217 -0.32739  0.08055  0.44805  1.21676
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  field    (Intercept) 16.589   4.073
##  Residual              2.121   1.456
## Number of obs: 24, groups:  field, 12
##
## Fixed effects:
##                        Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)             11.2319     1.7659 11.1971   6.361 4.95e-05 ***
## varietyv2                5.5711     2.4973 11.1971   2.231 0.047051 *
## irrigationi2             4.6695     0.8407 10.0000   5.554 0.000243 ***
## varietyv2:irrigationi2  -1.4279     1.1890 10.0000  -1.201 0.257430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) vrtyv2 irrgt2
## varietyv2   -0.707
## irrigation2 -0.238  0.168
## vrtyv2:rrg2  0.168 -0.238 -0.707
```

```
par(mar=c(4,12,4,2))
plot(glht(mod))
```



```
mod<-aov(yield~variety*irrigation+Error(field),data=d)
summary(mod)
```

```
##
## Error: field
##           Df Sum Sq Mean Sq F value Pr(>F)
## variety    1  141.5   141.6    4.01 0.0731 .
## Residuals 10  353.0    35.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##                    Df Sum Sq Mean Sq F value   Pr(>F)
## irrigation          1  93.88   93.88  44.270 5.68e-05 ***
## variety:irrigation  1   3.06    3.06   1.442    0.257
## Residuals          10  21.21    2.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chapter 12

Generalised linear models

So far we have assumed throughout that the variability in our models takes an approximately normal form. This is the assumption used in the classical parametric statistical tests and in regression, ANOVA and ANCOVA. Violations of the assumption often lead to the adoption of simple non parametric tests instead of more informative model based procedures due to worries about not meeting the assumptions needed for parametric modelling. However the set of models, known as Generalised Linear Models (GLMs) can use any known distribution for the errors. These are very powerful techniques. They are not much more difficult to apply using R than the methods that you have already seen. However careful thought is required in order to find the correct form for the model.

12.1 Poisson regression

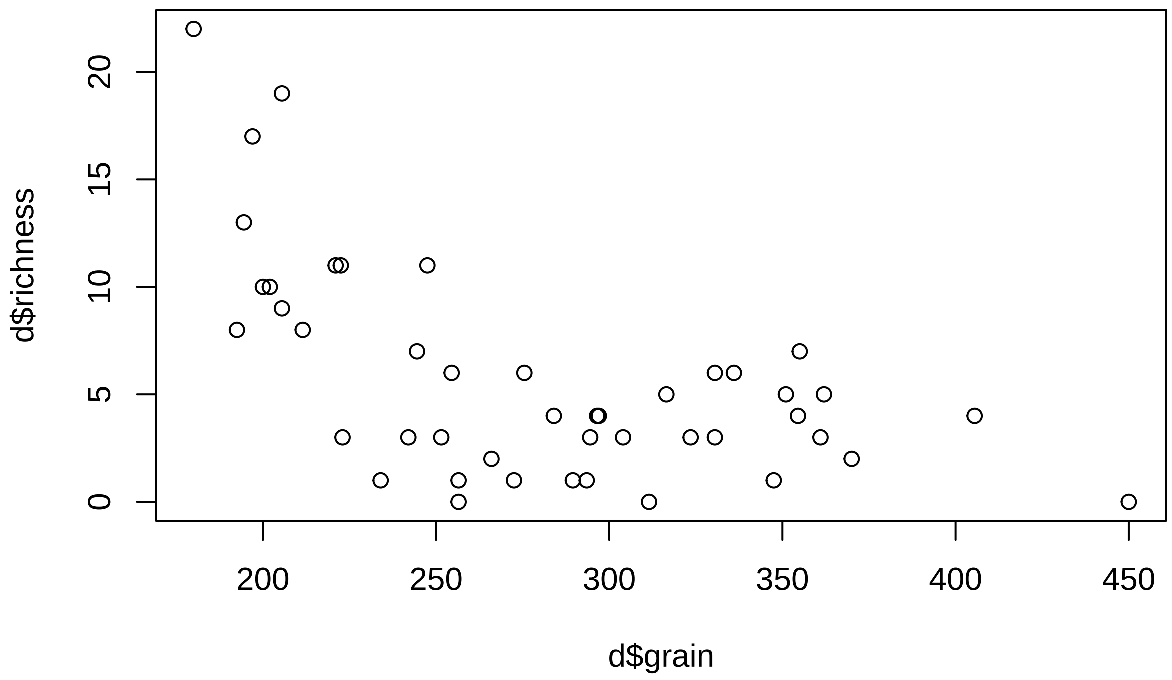
Let's look at the marine invertebrates data that we saw earlier.

```
d<-read.csv("https://tinyurl.com/aqm-data/marineinverts.csv")
str(d)
```

```
## 'data.frame':    45 obs. of  4 variables:
##  $ richness: int  0 2 8 13 17 10 10 9 19 8 ...
##  $ grain   : num  450 370 192 194 197 ...
##  $ height  : num  2.255 0.865 1.19 -1.336 -1.334 ...
##  $ salinity: num  27.1 27.1 29.6 29.4 29.6 29.4 29.4 29.6 29.6 29.6 ...
```

Plotting species richness against grain size again.

```
attach(d)
plot(d$richness~d$grain)
```



In the previous analysis we looked at how allowing the model to adopt a curved form led to a better fit. However the issue of the inappropriate use of the normal distribution to represent the error term was ignored.

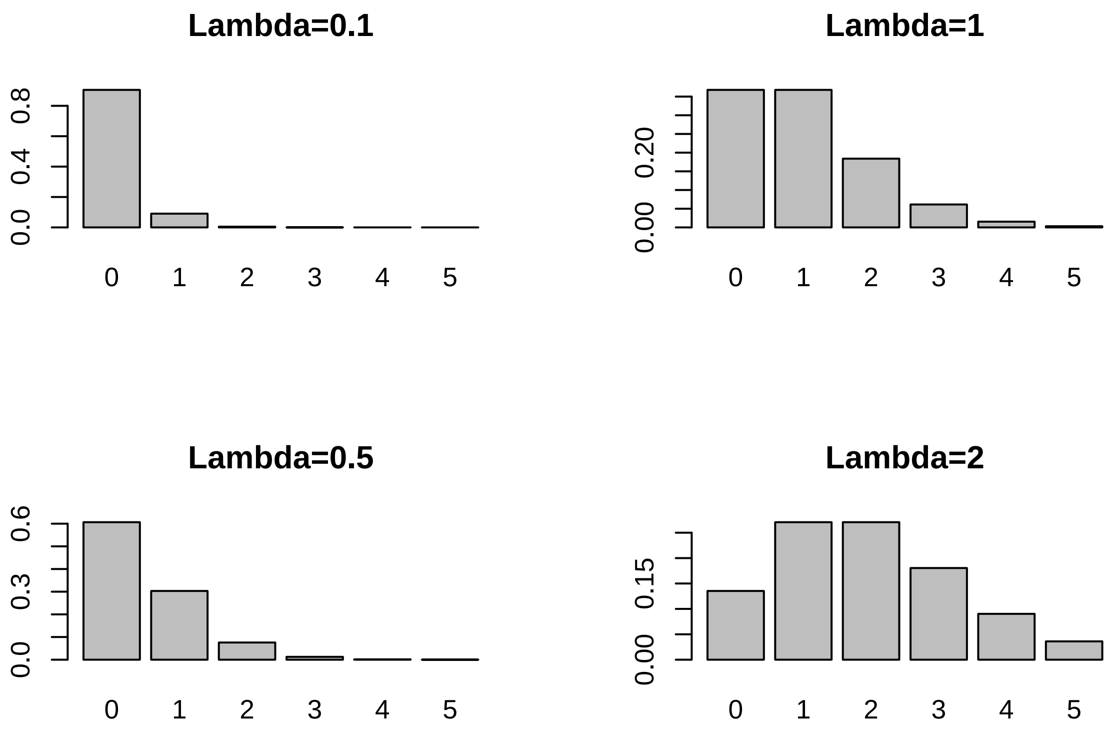
One way of thinking about the situation is to remember that the description of a regression line includes some statement about the errors.

$y = a + bx + \epsilon$ where $\epsilon = N(o, \sigma^2)$

This equation should be able to describe the process that leads to each data point. The model has a deterministic component (the regression line) and a stochastic component (the error term). However when the points are counts a continuous error term is incorrect. Although the mean value (trend) does not have to be an integer value, the actual data values do. So the errors around the trend should be discrete.

The poisson distribution can represent this. For any value of lambda (which is continuous) the probability distribution of values is discrete. The poisson distribution automatically builds in heterogeniety of variance as the variance of a poisson distribution is in fact equal to lambda.

```r
par(mfcol=c(2,2))
barplot(dpois(0:5,lambda=0.1),names=0:5,main="Lambda=0.1")
barplot(dpois(0:5,lambda=0.5),names=0:5,main="Lambda=0.5")
barplot(dpois(0:5,lambda=1),names=0:5,main="Lambda=1")
barplot(dpois(0:5,lambda=2),names=0:5,main="Lambda=2")
```

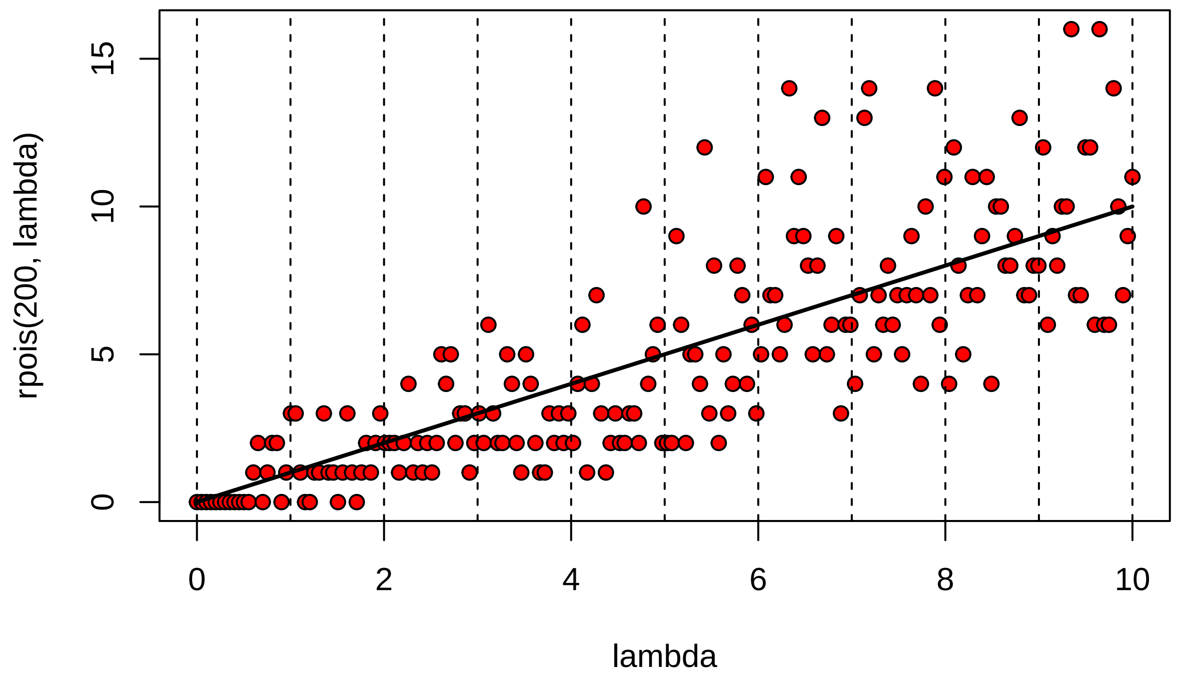


Let's think of a regression line with poisson errors with a=0, and b=1.

$y = a + bx + \epsilon$ where $\epsilon = poisson(lambda = y)$

Something interesting happens in this case. Lambda is a measure of the central tendency, but for most of the regression line no observations can actually take the value of lambda. A point can only fall on the line when lambda happens to be an integer.

```
lambda<-seq(0,10,length=200)
plot(lambda,rpois(200,lambda),pch=21,bg=2)
lines(lambda,lambda,lwd=2)
abline(v=0:10,lty=2)
```



This is the motive for fitting using maximum likelihood. A point that falls a long way away from the deterministic component of a model contributes more to the model's deviance than one that is close. A model with a low total deviance has a higher likelihood than one with a high deviance. The probabilities (that contribute to the deviance) are determined from assumptions regarding the form of the stochastic component of the model. The normal distribution is only one form of determining these probabilities. There are many other possible distributions for the error term.

So let's fit the model again, this time using poisson regression. By default this uses a log link function. This is usually appropriate for count data that cannot fall below zero. In this case the logarithmic link function also deals nicely with the problem of curvilinearity of the response.

```
mod1<-glm(data=d,richness ~ grain,family=poisson)
summary(mod1)
```

```
##
## Call:
## glm(formula = richness ~ grain, family = poisson, data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4828  -1.3897  -0.0732   0.8644   2.5838
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.238393   0.299033  14.174  < 2e-16 ***
## grain       -0.009496   0.001179  -8.052 8.16e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 179.75  on 44  degrees of freedom
## Residual deviance: 105.35  on 43  degrees of freedom
## AIC: 251.35
```

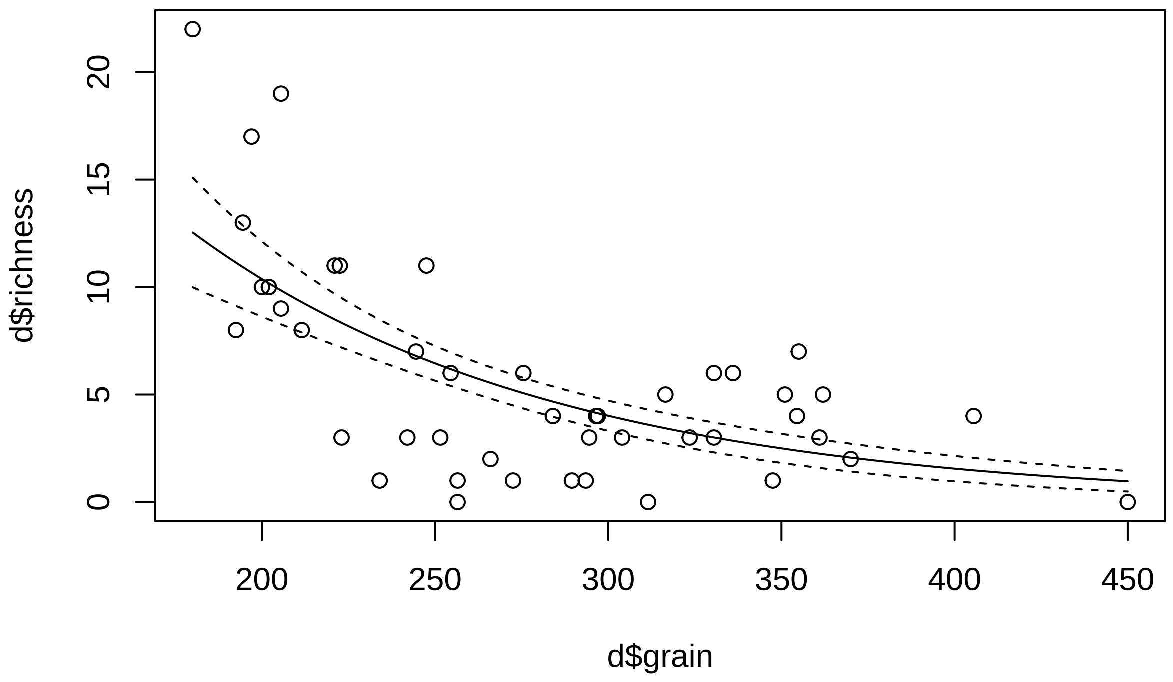
```
##
## Number of Fisher Scoring iterations: 5
```

```
confint(mod1)
```

```
##                    2.5 %       97.5 %
## (Intercept)  3.65451714  4.827458978
## grain       -0.01185141 -0.007224939
```

Plotting the model shows it's form. Note that with when fitting a GLM in R we can ask for the standard errors and produce approximate confidence intervals using them.

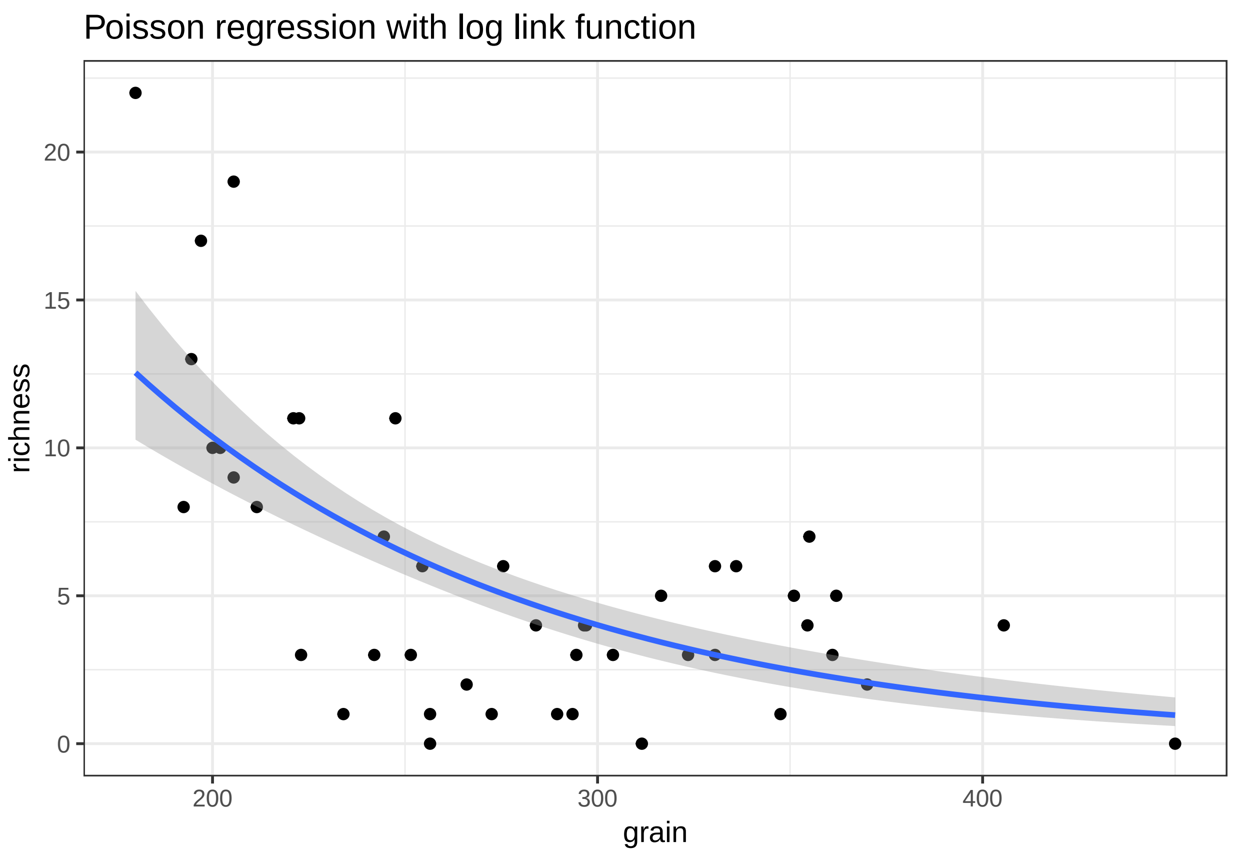
```
plot(d$richness ~ d$grain)
x<-seq(min(d$grain),max(d$grain),length=100)
a<-predict(mod1,newdata=list(grain=x),type="response",se=T)
lines(x,a$fit-2*a$se.fit,lty=2)
lines(x,a$fit+2*a$se.fit,lty=2)
lines(x,a$fit)
```



12.2 GGplot

Its easy to add a glm to a ggplot scatterplot. However be careful to add in the methods.args.

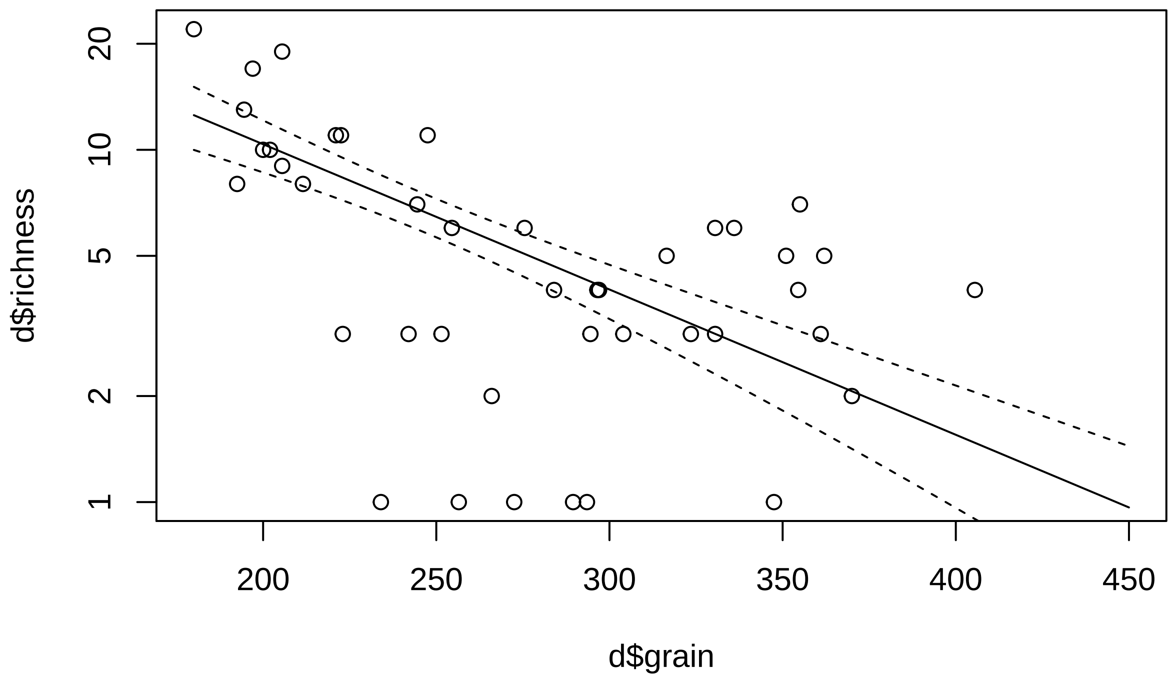
```
library(ggplot2)
g0<-ggplot(d,aes(x=grain,y=richness))
glm1<-g0+geom_point()+stat_smooth(method="glm",method.args=list( family="poisson"), se=TRUE) +ggtitle("
glm1
```



12.3 Showing the results with logged y

This is **not** a good approach, as the zeros are lost, but it demonstrates the idea.

```
plot(d$richness ~d$grain, log="y")
x<-seq(min(grain),max(grain),length=100)
a<-predict(mod1,newdata=list(grain=x),type="response",se=T)
lines(x,a$fit-2*a$se.fit,lty=2)
lines(x,a$fit+2*a$se.fit,lty=2)
lines(x,a$fit)
```



12.4 Log link function explained

The coefficients of the model when we ask for a summary are rather hard to undertand.

```
summary(mod1)
```

```
##
## Call:
## glm(formula = richness ~ grain, family = poisson, data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4828  -1.3897  -0.0732   0.8644   2.5838
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.238393   0.299033  14.174  < 2e-16 ***
## grain       -0.009496   0.001179  -8.052 8.16e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 179.75  on 44  degrees of freedom
## Residual deviance: 105.35  on 43  degrees of freedom
## AIC: 251.35
##
## Number of Fisher Scoring iterations: 5
```

The slope is given as -0.009. What does this mean? Unlike a regeression slope it is **NOT** the change in y for a unit change in x, as we are using a logarithmic link function.

In generalized linear models, there is always some sort of link function, which is the link between the mean of Y on the left and the predictor variable on the right. It is possible to use the identity link, which leaves the result the same, but typically some other link function is used. The identity link is not a good choice for data with many zeros.

Formally the link function is ..

$f(y|x) = a + bx$

I.e. Some function of the conditional value of y on x, ignoring the residual error, is predicted by a regression equation rather than simply y.

The log link function exponentiates the linear predictors. It **does not** actually involve a log transform of the outcome variable.

$y = exp(a + bx)$

Which could be also written as ..

$y = e^{a+bx}$

As the logarithm used is the natural logarithm this implies that expected value of y is **multiplied** by $exp(b)$ as we increase the value of x by 1 unit.

This is not intuitive.

Exponentiating the coefficients in R for the model above produces this result..

```
exp(coef(mod1))
```

```
## (Intercept)       grain
##  69.2963902   0.9905485
```

So, the intercept,for a grain size of zero is 69.3 and for each unit increase in grain size the diversity is changed by 99.055 % of the previous value. This is a process of exponential decay, as the richness is falling away steadily with each unti increase in grain size, but the model never leads to a predicted species richness below zero.

One way to make all this a little more understandable is to divide the natural logarithm of 2 (0.69) by the raw slope coefficient, which was found to be -0.009.

```
log(2)/(coef(mod1)[2])
```

```
##     grain
## -72.99001
```

This is using the formula for the half life, or doubling time, in an expenential decay or growth model.

So, in order to double the expected species richness we therefore would have to change the grain size by -72.99 units.

When presenting the results to a mathematically sophisticated audience you can safely place the coefficients within the equation and expect the audience to make sense of it.

$y = e^{a+bx}$

When explaining the result in words you can say that a change in grain size of -72.99 leads to doubling of expected species richness.

Showing a scatterplot with the fitted line is usually the easiest way to visualise the model and to make sense of it intuitively.

12.4.1 Likelihood and deviance

In order to fully understand all the elements used when analysing a GLM we also need at least an intuitive understanding of the concepts of likelihood and deviance.

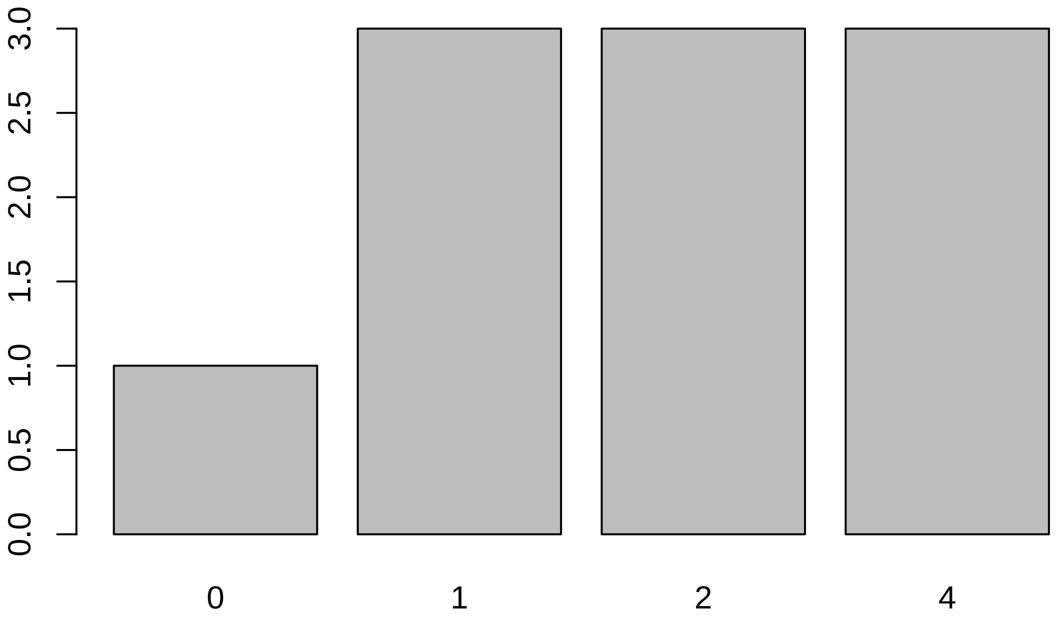
Models are fit by maximising the likelihood. But, what is the likelihood?

To try to inderstand this, let's first obtain some simple count data. We can simulate the counts from a poisson distribution.

```r
set.seed(1)
x<-rpois(10,lambda=2)
x
```

```
##  [1] 1 1 2 4 1 4 4 2 2 0
```

```r
barplot(table(x))
```



We can fit a simple model that just involves the intercept (mean) using R. This is.

```r
mod<-glm(x~1,poisson(link="identity"))
summary(mod)
```

```
##
## Call:
## glm(formula = x ~ 1, family = poisson(link = "identity"))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.04939  -0.84624  -0.06957   0.85560   1.16398
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.1000     0.4583   4.583 4.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
```

```
##
##     Null deviance: 10.427  on 9  degrees of freedom
## Residual deviance: 10.427  on 9  degrees of freedom
## AIC: 36.066
##
## Number of Fisher Scoring iterations: 3
```

```
coef(mod)
```

```
## (Intercept)
##         2.1
```

```
confint(mod)
```

```
##    2.5 %   97.5 %
## 1.325155 3.130620
```

```
lam<-coef(mod)
```

Now, under the poisson model we can calculate a probability of getting any integer from a poisson distribution with a mean of lambda using a standard formula that is built into R. So the probability of getting a zero is dpois(0,lambda=2.1)

dpois(0,lambda=2.1)=0.122

dpois(1,lambda=2.1)=0.257

dpois(2,lambda=2.1)= 0.27

dpois(3,lambda=2.1)=0.189

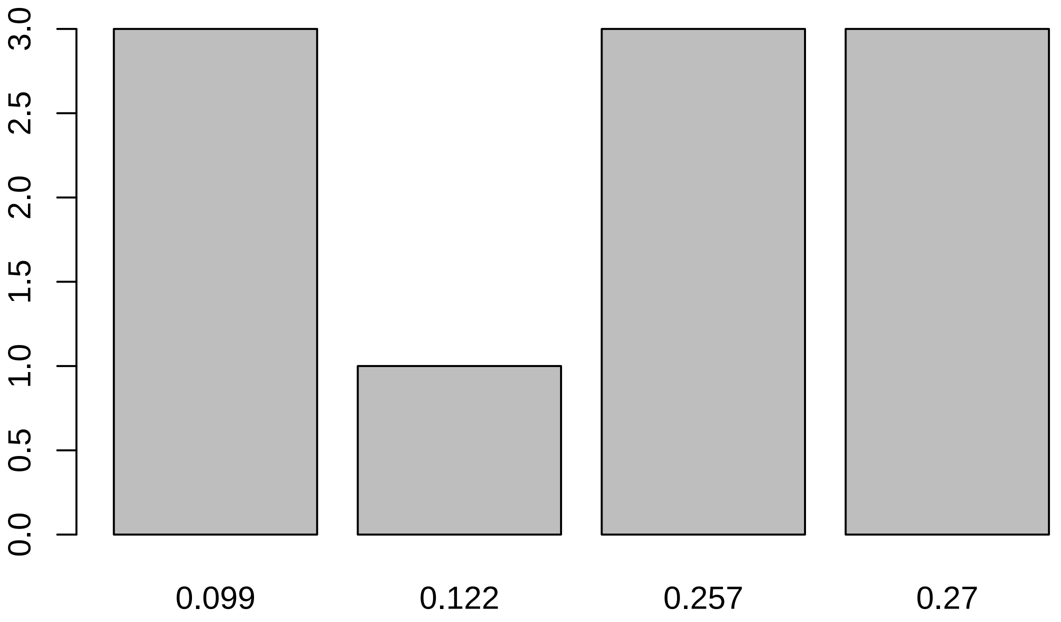
dpois(4,lambda=2.1)=0.099

What this means is that we have a probability (likelihood) for each of the data points given the model parameter (lambda). We can look at this as a barplot of counts of each probability value.

```
dx<-dpois(x,lambda=lam)
dx
```

```
##  [1] 0.25715850 0.25715850 0.27001642 0.09923104 0.25715850 0.09923104
##  [7] 0.09923104 0.27001642 0.27001642 0.12245643
```

```
barplot(table(round(dx,3)))
```



The probability of getting **EXACTLY** the data that we have is the product of all these probabilities, as we find the combined probability of independent events by multiplying them together. Because this is going to result in very small numbers it is usually easier to work with logarithmns and add them together. Hence the term log likelihood that you will see used in all treatments of GLMs.

```r
loglik<-sum(log(dx))
loglik
```

```
## [1] -17.03292
```

```r
logLik(mod)
```

```
## 'log Lik.' -17.03292 (df=1)
```

OK, so that was not too difficult. Notice as well that this calculation gave us the maximum likelihood. If we had used any other value as an estimate for lambda we would have got a lower value expressed as a negative value.

```r
sum(log(dpois(x,lambda=1)))
```

```
## [1] -21.6136
```

```r
sum(log(dpois(x,lambda=lam)))
```

```
## [1] -17.03292
```

```r
sum(log(dpois(x,lambda=3)))
```

```
## [1] -18.54274
```

In order to simplify matters further we remove the sign and work with -2 log likelihood.

```r
-2*sum(log(dpois(x,lambda=lam)))
```

```
## [1] 34.06584
```

The AIC which we will look at later in the course as a way of comparing two models combines the -2 log likelihood with the number of parameters (k). In this case we have just one parameter so AIC adds 2 to the number we previously calculated.

AIC=2k-2ln(L)

```
AIC(mod)
```

```
## [1] 36.06584
```

Now finally, what does the deviance refer to?

Well, even a model which has a separate parameter for each data point will still have a likelihood below one. The deviance refers to the difference in -2 log likelihood between the fully saturated model and the actual model. We can get the -2 log likelihood for this model as well.

```
dpois(x,lambda=x)
```

```
##  [1] 0.3678794 0.3678794 0.2706706 0.1953668 0.3678794 0.1953668 0.1953668
##  [8] 0.2706706 0.2706706 1.0000000
```

```
satmod<--2*sum(log(dpois(x,lambda=x)))
satmod
```

```
## [1] 23.63838
```

Just to confirm, this should give us the deviance.

```
-2*loglik-satmod
```

```
## [1] 10.42746
```

```
deviance(mod)
```

```
## [1] 10.42746
```

Notice that we had ten data points (residual degrees of freedom = n-1 = 9) and a residual deviance that is around 10. This is an indication that the assumption of Poisson distributed residuals is a reasonable one as for mathematical reasons that we need not go into we would expect an addition of just under 1 to the deviance for each additional data point.

Going back to the summary of the model

```
summary(mod)
```

```
##
## Call:
## glm(formula = x ~ 1, family = poisson(link = "identity"))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.04939  -0.84624  -0.06957   0.85560   1.16398
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.1000     0.4583   4.583 4.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 10.427  on 9  degrees of freedom
## Residual deviance: 10.427  on 9  degrees of freedom
## AIC: 36.066
##
## Number of Fisher Scoring iterations: 3
```

We can see that in this artificial case the null deviance and the residual deviance are identical. This is because the null is "true". There is nothing to report in the model apart from the intercept, i.e. a single value for lambda. If we use this concept in a model with a predictor variable we should see a difference between these two numbers. The larger the diffence, the more of the deviance is "explained" by our predictor. We want to reduce the deviance bt fitting a model, so if there is a relationship the residual deviance should always be lower than the null deviance.

12.4.1.1 Overdispersion

If the residual deviance is larger than residual degrees of freedom we have overdispersion (extra, unexplained variation in the response).

```r
summary(mod1)
```

```
##
## Call:
## glm(formula = richness ~ grain, family = poisson, data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4828  -1.3897  -0.0732   0.8644   2.5838
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.238393   0.299033  14.174  < 2e-16 ***
## grain       -0.009496   0.001179  -8.052 8.16e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 179.75  on 44  degrees of freedom
## Residual deviance: 105.35  on 43  degrees of freedom
## AIC: 251.35
##
## Number of Fisher Scoring iterations: 5
```

This means that in fact the measured variance in the data, after taking into account the regression line, is still larger than the lambda values over the range of the regression. This is extra variability that goes beyond that expected under the assumption that the residuals are poisson distributed.

This is the diagnostic tool which is used in Poisson regression. The point is that under a poisson distribution the variance is fixed. It is always identical to the mean (lamda). This may not be a reasonable assumption, but it is the assumption being made. If it is not met we will need to make some compensation for this in order to produce a more justifiable model.

12.4.1.2 Quasi-poisson regression

A simple way of dealing with over dispersion is to use so called quasi-poisson regression. This finds a weight so that instead of assuming that the variance is equal to lambda the assumption is made that it is equal to some multiple of lambda. The multiple is estimated from the data. The effect is to reduce the significance of the regression term and widen the confidence intervals. It is a rather outdated technique that has some problems, but we'll try it anyway.

```
mod2<-glm(richness ~ grain,family=quasipoisson)
```

```
## Error in model.frame.default(formula = richness ~ grain, drop.unused.levels = TRUE): variable length
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = richness ~ mat, data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49502 -0.39841 -0.03172  0.41128  1.46120
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.909188   0.549077   54.47 1.43e-11 ***
## mat         -0.090708   0.008941  -10.15 7.62e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8322 on 8 degrees of freedom
## Multiple R-squared:  0.9279, Adjusted R-squared:  0.9189
## F-statistic: 102.9 on 1 and 8 DF,  p-value: 7.618e-06
```

```
AIC(mod2)
```

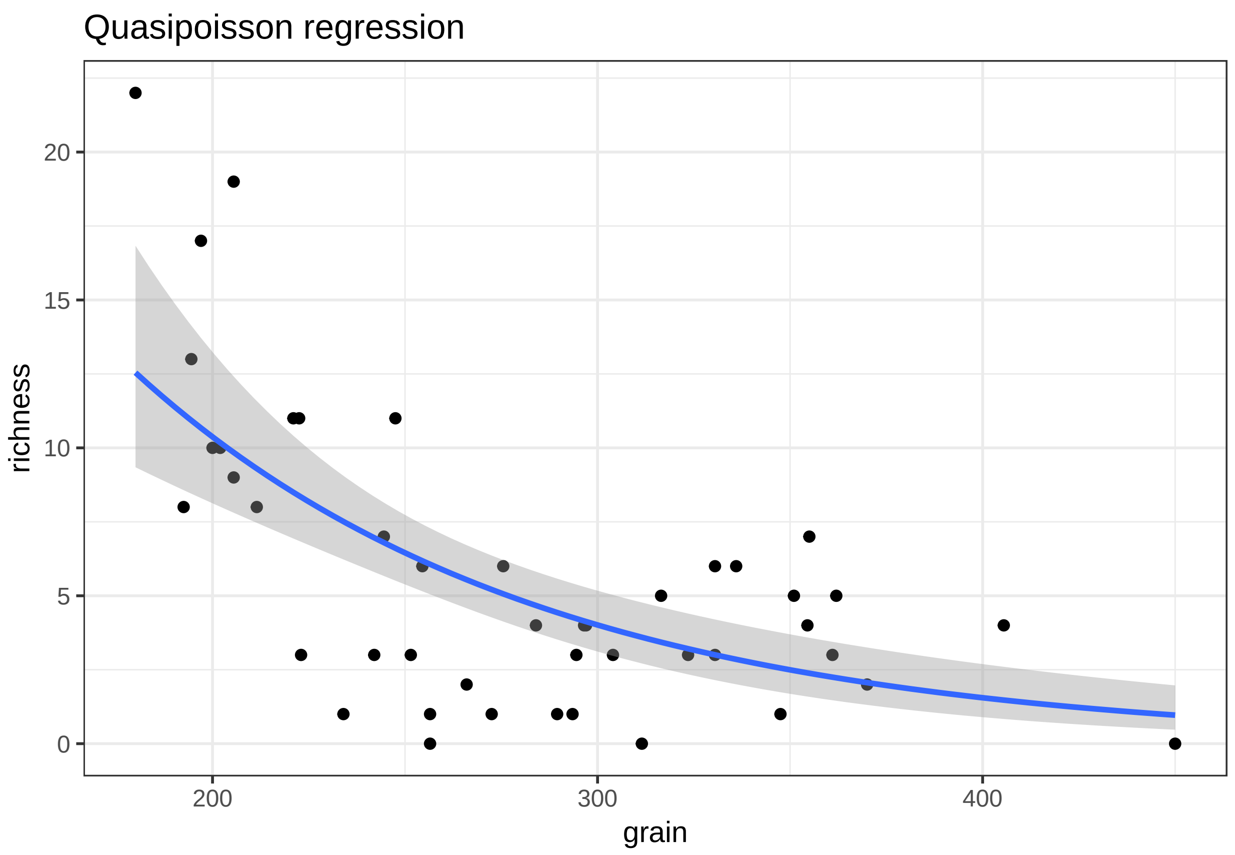
```
## [1] 28.47456
```

```
confint(mod2)
```

```
##                  2.5 %      97.5 %
## (Intercept) 28.6430130 31.17536233
## mat         -0.1113255 -0.07009151
```

Notice that the confidence intervals are wider. However we cannot obtain a value for AIC from a quasi model as the likelihood function is not fully defined. This limits the application of quasi poisson models, so we'll pass on quickly to a rather more useful approach..

```
glm2<-g0+geom_point()+geom_smooth(method="glm", method.args=list(family="quasipoisson"), se=TRUE) + ggt
glm2
```



12.4.2 Negative binomial regression

As we have seen, there is a problem with quasi poisson regression.There is no defined form for the likelihood. Therefore it is impossible to calculate AIC. This makes it difficult to run model comparisons using quasi poisson models. An alternative is to fit the model assuming a negative binomial distribution for the error terms. This is a well defined model for over dispersed count data.

```
library(MASS)
mod3<-glm.nb(richness ~ grain)
```

```
## Error in model.frame.default(formula = richness ~ grain, drop.unused.levels = TRUE): variable length
```

```
summary(mod3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: richness ~ mat + (1 | site)
##    Data: d
##
## REML criterion at convergence: 268.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7906 -0.5656  0.1164  0.6804  1.6461
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  site     (Intercept)  0.00    0.000
##  Residual             11.72    3.423
## Number of obs: 50, groups:  site, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 29.90919    1.01005  29.611
## mat         -0.09071    0.01645  -5.515
##
## Correlation of Fixed Effects:
##     (Intr)
## mat -0.878
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
confint(mod3)
```

```
##                  2.5 %      97.5 %
## .sig01       0.0000000  1.18505144
## .sigma       2.7904493  4.13658715
## (Intercept) 27.9316552 31.88672006
## mat         -0.1229084 -0.05850855
```

Notice that the AIC for the negative binomial model is much lower than that for the (incorrect) poisson model. The residual deviance is now not much larger than the residual degrees of freedom. It is very important to include the overdispersion rather than use the assumption that the variance is equal to lambda that is built into poisson regression.

```r
AIC(mod1)
```

```
## [1] 251.3523
```

```r
AIC(mod3)
```

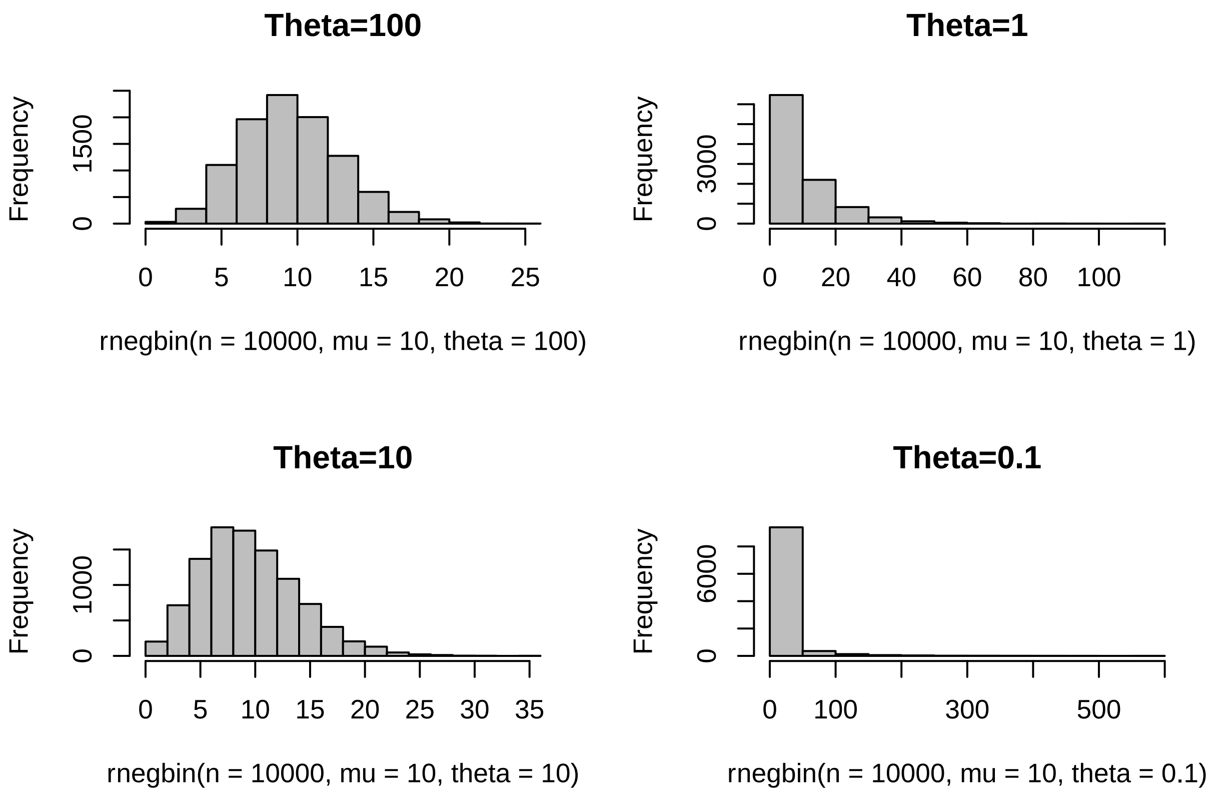
```
## [1] 276.9441
```

The variance of the negative binomial is

$var = \mu + \frac{\mu^2}{\theta}$

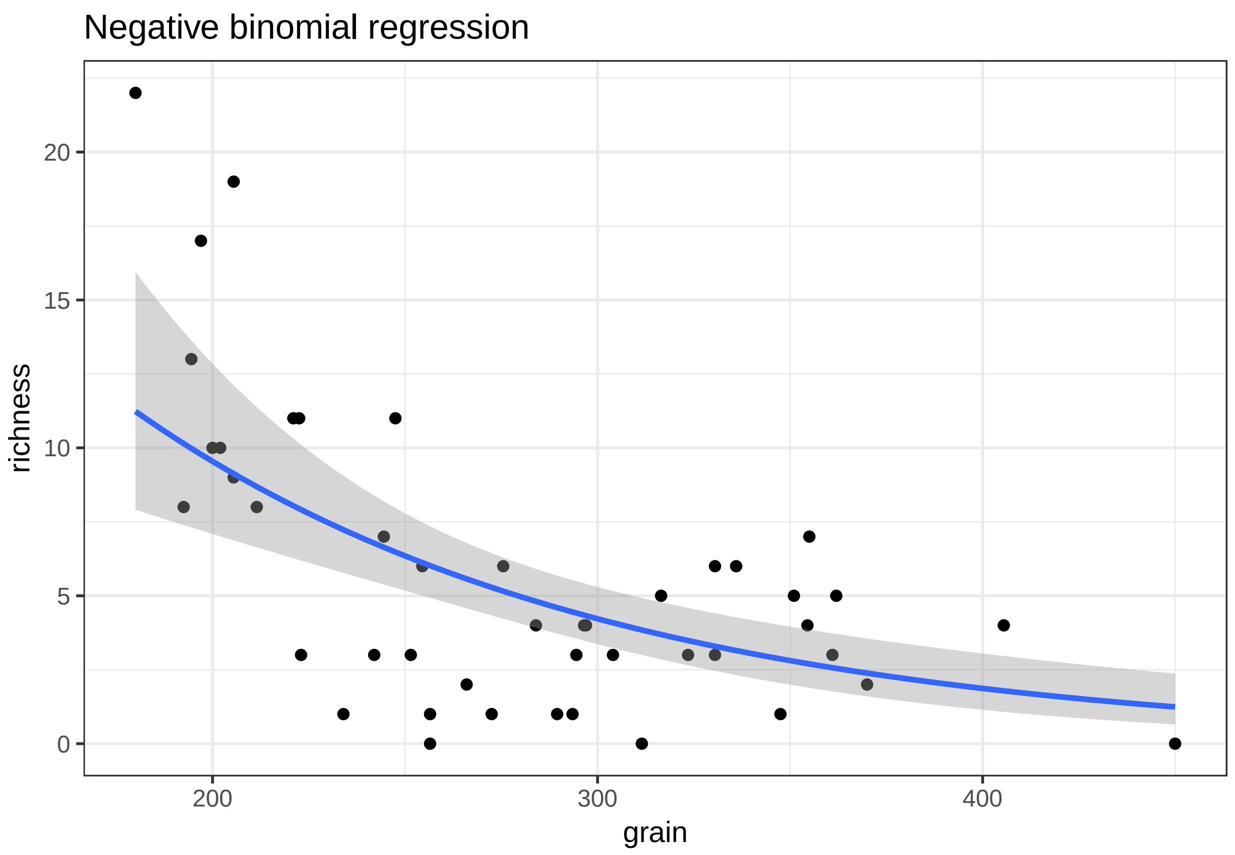
So theta controls the excess variability compared to Poisson. The smaller the value of theta the more skewed the distribution becomes.

```r
par(mfcol=c(2,2))
hist(rnegbin(n=10000,mu=10,theta=100),main="Theta=100",col="grey")
hist(rnegbin(n=10000,mu=10,theta=10),main="Theta=10",col="grey")
hist(rnegbin(n=10000,mu=10,theta=1),main="Theta=1",col="grey")
hist(rnegbin(n=10000,mu=10,theta=0.1),main="Theta=0.1",col="grey")
```



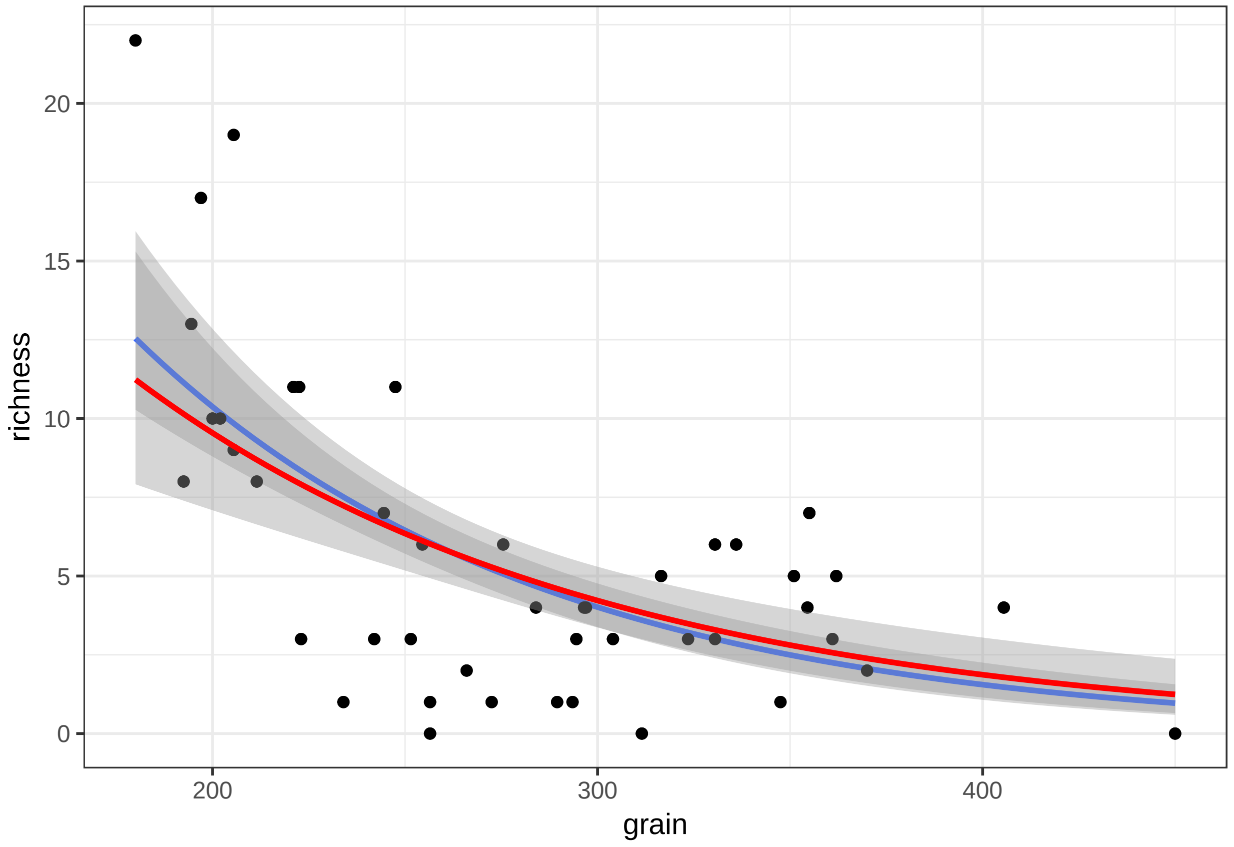
Plotting the model produces a very similar result to that shown by the quasipoisson model.

```
glm3<-g0+geom_point()+geom_smooth(method="glm.nb", se=TRUE) +ggtitle("Negative binomial regression")
glm3
```



12.5 Comparing the results

```
library(ggplot2)
g0<-ggplot(d,aes(x=grain,y=richness))
glm1<-g0+geom_point()+stat_smooth(method="glm",method.args=list( family="poisson"), se=TRUE)
glm3<-glm1+geom_point()+geom_smooth(method="glm.nb", se=TRUE,color="red")
glm3
```



```r
library(pscl)
modh<-hurdle(d$richness~grain,dist="negbin")
summary(modh)
```

```
##
## Call:
## hurdle(formula = d$richness ~ grain, dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.5804 -0.8495 -0.1085  0.6236  2.0657
##
## Count model coefficients (truncated negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.915924   0.458802   8.535  < 2e-16 ***
## grain       -0.008220   0.001724  -4.767 1.87e-06 ***
## Log(theta)   1.510256   0.510987   2.956  0.00312 **
## Zero hurdle model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.34139    3.39460   2.163   0.0306 *
## grain       -0.01531    0.01005  -1.524   0.1275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta: count = 4.5279
## Number of iterations in BFGS optimization: 13
## Log-likelihood: -114.5 on 5 Df
```

```
AIC(modh)
```

```
## [1] 238.9305
```

```
AIC(mod3)
```

```
## [1] 276.9441
```

```
confint(modh)
```

```
##                           2.5 %       97.5 %
## count_(Intercept)  3.01668827  4.815159689
## count_grain       -0.01159973 -0.004840422
## zero_(Intercept)   0.68811013 13.994679574
## zero_grain        -0.03500187  0.004381010
```

```
modzi <- zeroinfl(data=d,richness~grain,dist="negbin")
summary(modzi)
```

```
##
## Call:
## zeroinfl(formula = richness ~ grain, data = d, dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.5740 -0.8408 -0.1085  0.6161  2.0427
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.896314   0.446411   8.728  < 2e-16 ***
## grain       -0.008132   0.001659  -4.902 9.49e-07 ***
## Log(theta)   1.514259   0.514694   2.942  0.00326 **
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.37854   11.19100  -0.481    0.631
## grain        0.00447    0.03878   0.115    0.908
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 4.5461
## Number of iterations in BFGS optimization: 35
## Log-likelihood: -114.6 on 5 Df
```

```
AIC(modh)
```

```
## [1] 238.9305
```

```
AIC(modzi)
```

```
## [1] 239.1852
```

```
AIC(mod3)
```

```
## [1] 276.9441
```

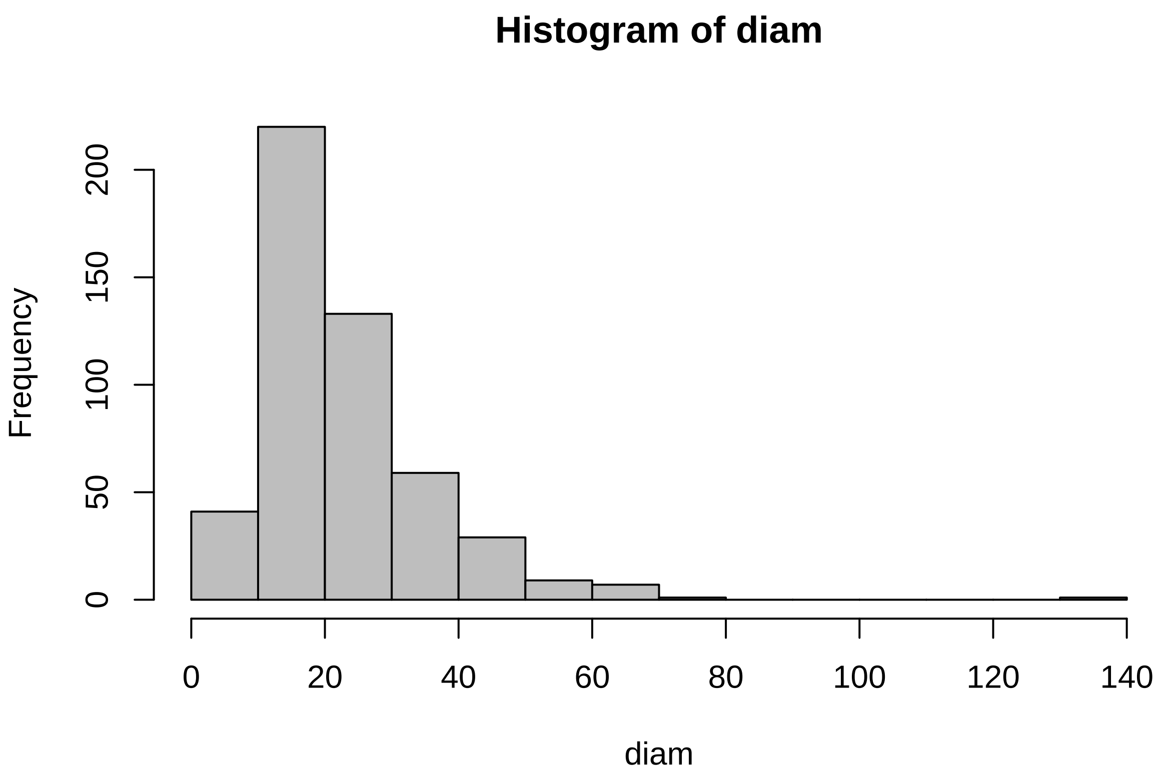
12.6 Models with binomial errors

The most commonly used GL is probably logistic regression. In this particular model the response can only take values of zero or one. Thus it is clear from the outset that errors cannot be normal. Let's set up a simple simulated data set to show how this works. Imagine we are interested in mortality of pine trees following a ground fire. We might assume that the population of tree diameters are log normally distributed with a mean of twenty.

```
set.seed(1)
diam<-sort(rlnorm(500,mean=log(20),sd=0.5))
summary(diam)
```

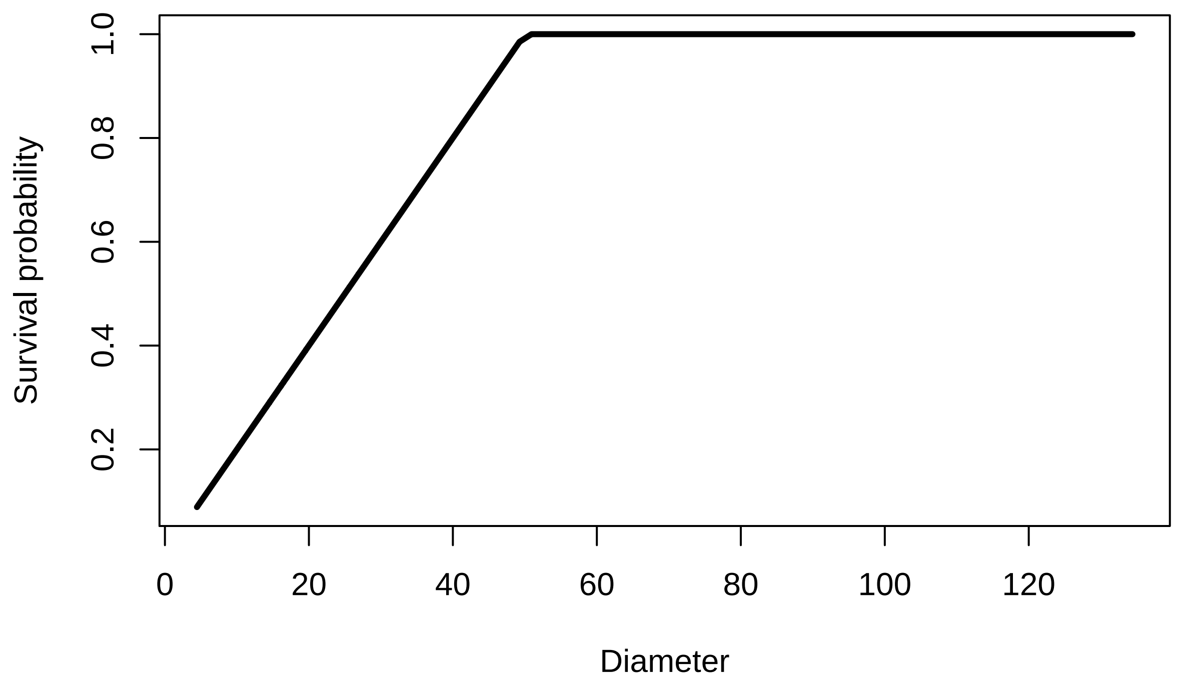
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.445  14.660  19.636  23.018  28.079 134.407
```

```
hist(diam,col="grey",breaks=10)
```



Let's simulate some response data based on an extremely simple underlying pattern for tree mortality. We might assume that trees with diameters of over 40 cm have bark that has reached a thickness that prevents the tree being killed by the fire. We might also assume a simple linear relationship between diameter and mortality up to this threshold and build a simple rule based vector of the probability that a tree survives the fire as a function of its diameter.

```
p<-diam/50
p[p>1]<-1
plot(diam,p,ylab="Survival probability",xlab="Diameter",type="l",lwd=3)
```

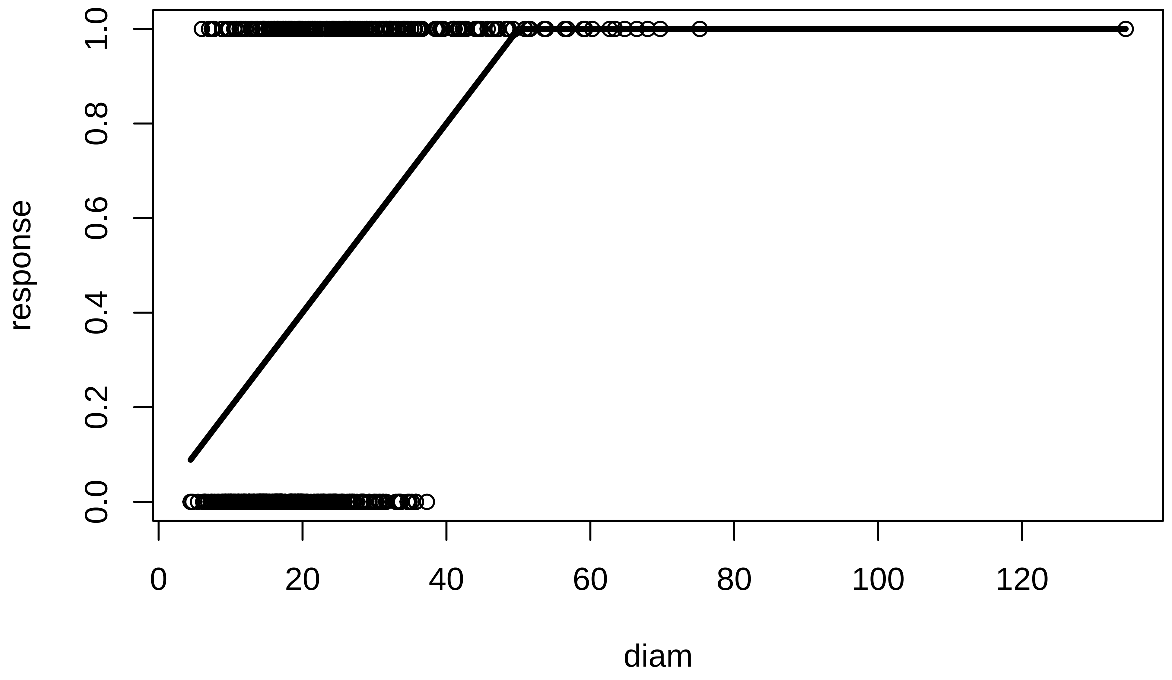


Although we have a very simple underlying deterministic model, we will not see this directly when we collect data. Any individual tree will be either alive or dead. Thus our response will be zeros and ones. This is the problem that logistic regression deals with very neatly without the need to calculate proportions explicitly.

```r
f<-function(x)rbinom(1,1,x)
response<-as.vector(sapply(p,f))
head(response)
```

```
## [1] 0 0 0 1 0 0
```

```r
d<-data.frame(diam,response)
plot(diam,response)
lines(diam,p,lwd=3)
```



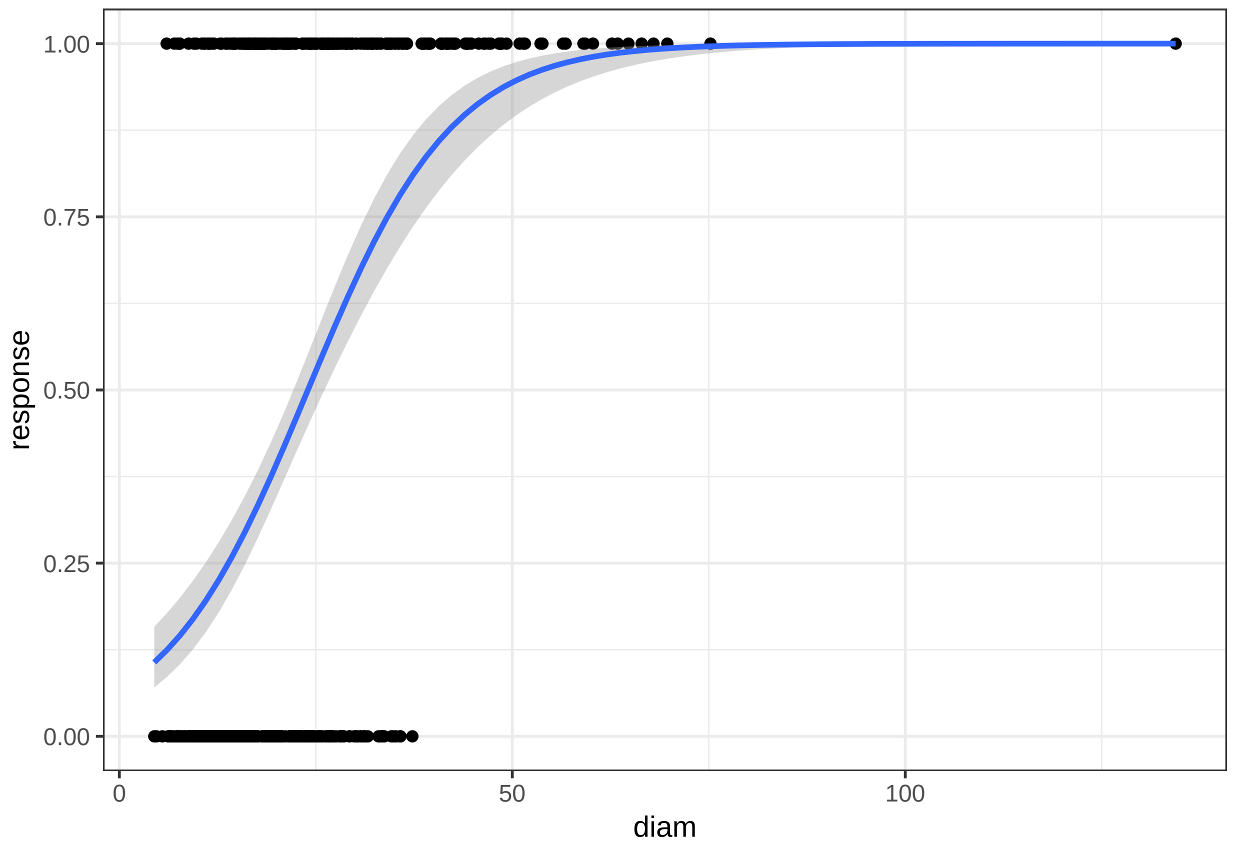
The task for the statistical model is to take this input and turn it back into a response model. Generalised linear models do this using a link function. In R it is very easy to specify the model. We simply write a model using the same syntax as for a linear model (one with gaussian errors) but we state the family of models we wish to use as binomial.

```
mod1<-glm(response~diam,family="binomial")
summary(mod1)
```

```
##
## Call:
## glm(formula = response ~ diam, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8202  -0.8891  -0.6053   1.0175   2.0428
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.60771    0.28033  -9.302   <2e-16 ***
## diam         0.10869    0.01217   8.929   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 688.91  on 499  degrees of freedom
## Residual deviance: 565.51  on 498  degrees of freedom
## AIC: 569.51
##
## Number of Fisher Scoring iterations: 5
```

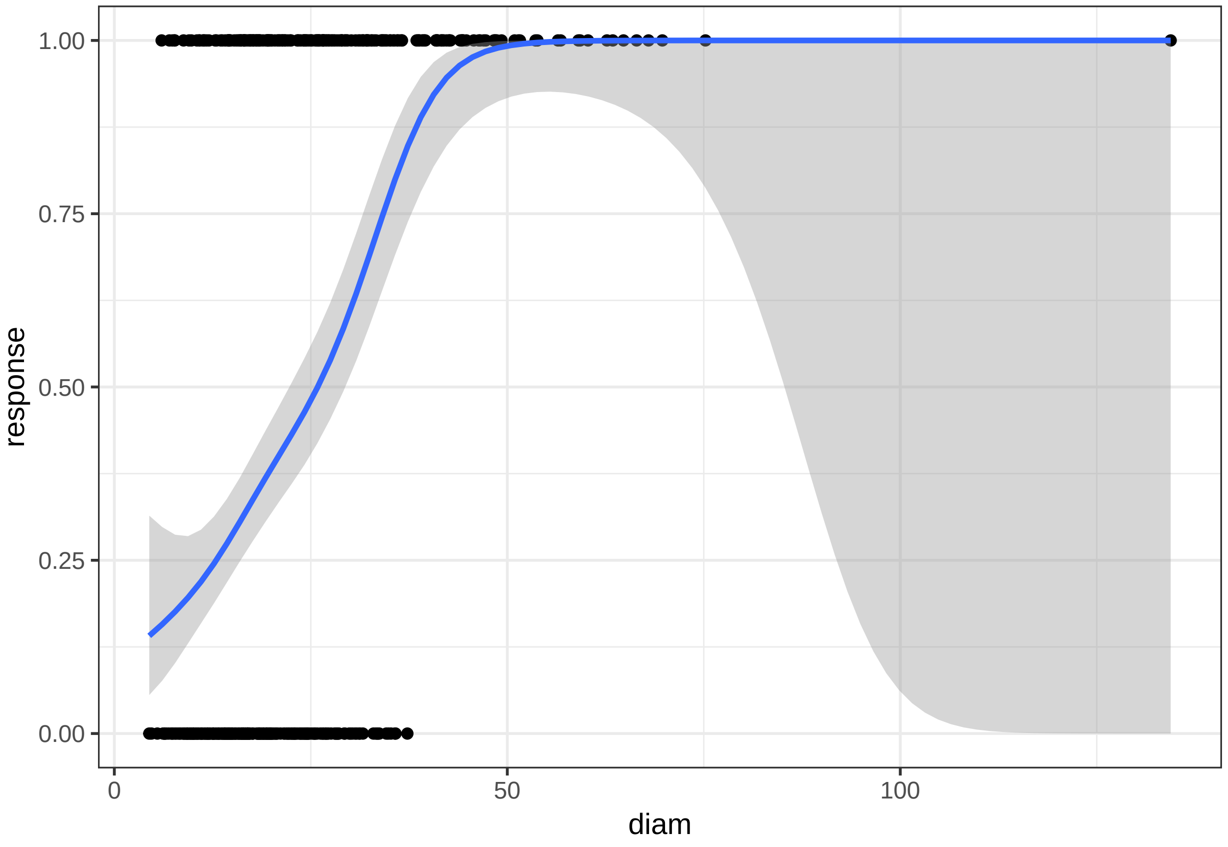
We can see that R does find a model that matches the underlying pattern very well by using the model for prediction. Again we visualise the model in order to understand it. This is always preferable to trying to understand a model from a table of numbers. Visualisation is particularly important for models with parameters expressed on a logit scale as this is not intuitive.

```
g0 <- ggplot(d,aes(x=diam,y=response))
g1<-g0+geom_point()+stat_smooth(method="glm",method.args=list(family="binomial"))
g1
```



If we wanted to check whether there was a response shape that differed from that assumed by the general linear model we could try a general additive model with a smoother.

```
library(mgcv)
g1<-g0+geom_point()+stat_smooth(method="gam",method.args=list(family="binomial"),formula=y~s(x))
g1
```



The curve is very similar. Note that as the smoother uses a form of “local” regression the confidence intervals expand in areas where there is little data.

In some cases the response would take a different form. This could happen if there were some optimum point at which some response occurred, for example the occurence of a species along an altitudinal gradient or shoreline. In this case the gam model would fit the data better than the linear model. We will look at how this can be tested formally later. A quick test is to calculate the AIC. If this is much lower for the gam it indicates that the gam may be a better fit.

```
glm_mod<-glm(data=d, response~diam, family=binomial)
gam_mod<-gam(data=d, response~s(diam), family=binomial)

AIC(glm_mod)
```

```
## [1] 569.5078
```

```
AIC(gam_mod)
```

```
## [1] 566.4594
```

In this case it is very slightly lower, but not enough to suggest the use of a gam.

12.7 The logit link function

The logit link function used in binomial glms makes the slope of the line quite difficult to understand. In most cases this doesn’t matter much, as you can concentrate on the sign and signficance of the parameter and show the line as a figure. However when analysing differences in response as a function of levels of a factor you do need to understand the logit link.

To illustrate let's take a very simple example. Ten leaves are classified as being taken from shade or sun and classified for presence of rust.

```
library(purrr)
set.seed(1)
light<-rep(c("shade","sun"),each=10)
presence<-1*c(rbernoulli(10,p=0.5),rbernoulli(10,p=0.1))
d<-data.frame(light,presence)
```

We can get a table of the results easily.

```
table(d)
```

```
##        presence
## light   0 1
##   shade 4 6
##   sun   9 1
```

So 6 of the leaves in the shade had rust present and 4 did not. The odds of rust are therefore 6 to 4. Odds are used in the logit transform rather than simple proportions because odds can take values between 0 and infinity, while proportions are bounded to lie between zero and one. Taking the logarithm of the odds leads to an additive model.

There are two factor levels, shade and sun. The default reference when a model is fitted will be the factor that is first in alphabetical order, i.e. shade. So after fitting a model the intercept will be the log of the odds in the shade. The effect of light will be the log odds in the sun minus the log odds in the shade.

```
odds_shade<-6/4
odds_sun<-1/9
log(odds_shade)
```

```
## [1] 0.4054651
```

```
log(odds_sun)-log(odds_shade)
```

```
## [1] -2.60269
```

We can see that this coincides with the model output.

```
mod<-glm(data=d,presence~light,family="binomial")
summary(mod)
```

```
##
## Call:
## glm(formula = presence ~ light, family = "binomial", data = d)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.354  -0.459  -0.459   1.011   2.146
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.4055     0.6455   0.628   0.5299
## lightsun     -2.6027     1.2360  -2.106   0.0352 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 25.898  on 19  degrees of freedom
## Residual deviance: 19.962  on 18  degrees of freedom
## AIC: 23.962
##
## Number of Fisher Scoring iterations: 4
```

If the coeficients are exponentiated then the first coeficient represents the baseline odds and the second coeficient represesnts this value divided by the odds for the "treatment". As binomial models are often used in epidemiology this explains why we could hear statements such as "eating processed meat increases the odds of contracting bowel cancer by a factor of 2". This is a literal interpretation of the exponentiated coeficient.

```
exp(coef(mod))
```

```
## (Intercept)    lightsun
##  1.50000000  0.07407408
```

```
odds_shade
```

```
## [1] 1.5
```

```
odds_sun/odds_shade
```

```
## [1] 0.07407407
```

To convert the odds into proportions divide the odds by 1 plus the odds.

```
odds_shade/(1+odds_shade)
```

```
## [1] 0.6
```

```
odds_sun/(1+odds_sun)
```

```
## [1] 0.1
```

So this gives the proportions as estimated by the model.

```
exp(coef(mod)[1])/(1+exp(coef(mod)[1]))
```

```
## (Intercept)
##         0.6
```

```
exp(coef(mod)[1] + coef(mod)[2])/(1+exp(coef(mod)[1] + coef(mod)[2]))
```

```
## (Intercept)
##         0.1
```

12.8 Exercises

1. GLMS can also be used when the explanatory variable is a factor. Here is a very simple data set that consists of counts of ragworm in two types of substrate, classified simply into mud and sand. Analyse the data using both a **general** linear model and a **generalised** linear model. Comment on the differences between the two aproaches.

```
d<-read.csv("/home/aqm/course/data/HedisteCounts.csv")
```

2. Binomial (prensence/absence) model

In some cases the actual numbers of organisms counted can be a poor choice of response variable. If organisms are highly aggregated then presence vs absence is a better choice. Reanalyse the ragworm data, this time using presence as the response.

```
d$pres<-1*(d$Count>0) ## This sets up a variable consisting of ones and zeros
```

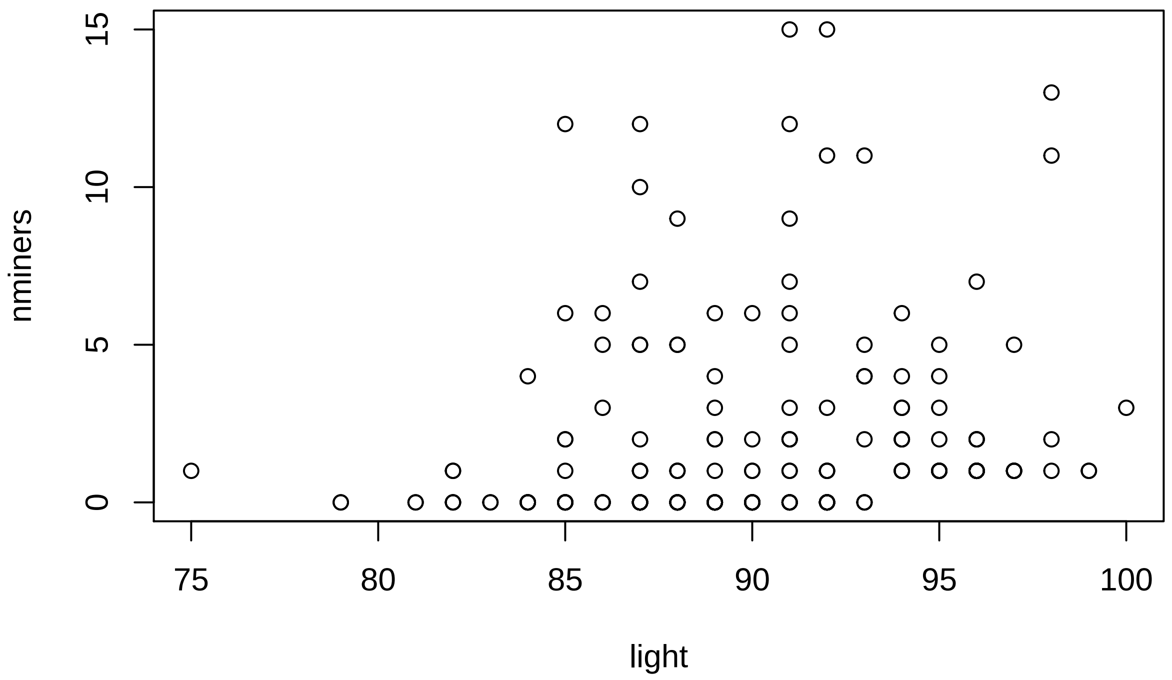
3. Leafminers and leaf exposure to light

The number of leaf miners were counted on 200 leaves exposed to different levels of ambient light, measured as a percentage of full exposure.

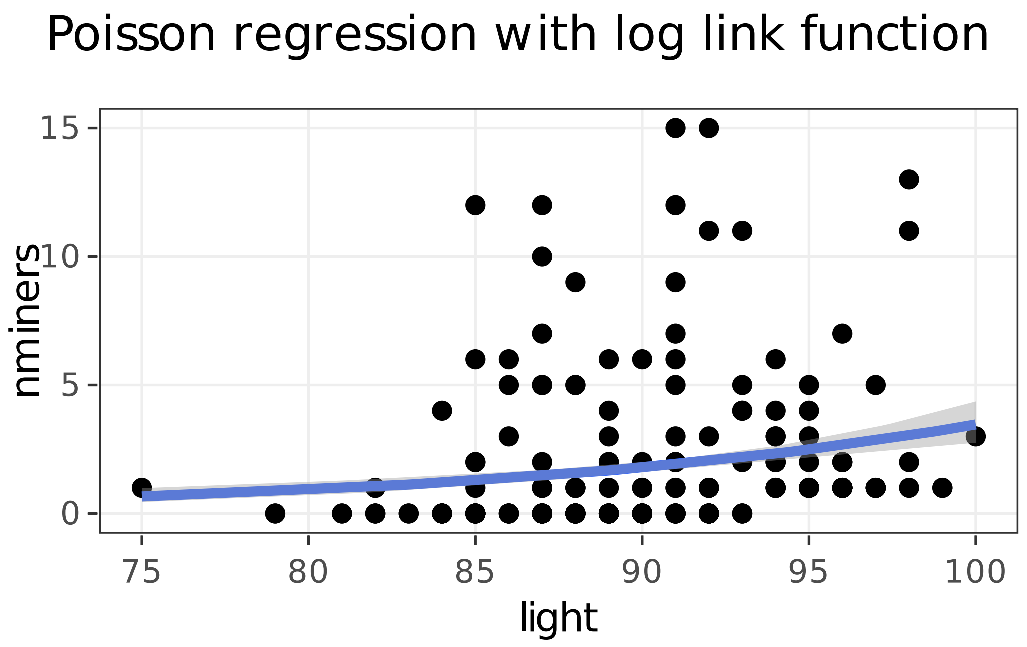
Analyse these data using an appropriate GLM.

```
d<-read.csv("/home/aqm/course/data/leafminers.csv")
```

```
plot(d)
```



```
library(plotly)
g0<-ggplot(d,aes(x=light,y=nminers))
glm1<-g0+geom_point()+stat_smooth(method="glm",method.args=list( family="poisson"), se=TRUE) +ggtitle("
ggplotly(glm1)
```



```
mod<-glm(data=d,nminers~light,family="poisson")
summary(mod)
```

```
##
## Call:
## glm(formula = nminers ~ light, family = "poisson", data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0888  -1.7275  -1.1676   0.0864   5.9691
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.36721    1.09865  -4.885 1.03e-06 ***
## light        0.06610    0.01203   5.496 3.88e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 751.31  on 199  degrees of freedom
## Residual deviance: 720.43  on 198  degrees of freedom
## AIC: 1025.3
##
## Number of Fisher Scoring iterations: 6
```

```
log(2)/coef(mod)[2]
```

```
##    light
## 10.48647
```

Chapter 13

Modelling with multiple variables

13.1 Introduction

When looking at simple regression models we have used a single numerical variable to explain and/or predict the variability in a second variable. Analysis of co-variance, uses one numerical variable and a factor. What if we have two or more numerical variables that could explain and/or predict some variable that we are interested in. Can we build models from them?

From a mathematical perspective using more variables in the model is straightforward. However as we have seen, the complex nature of ecological data has to be considered carefully.

Multiple regression implies all the same assumptions as regression. Simple intepretation of multiple regression models also relies on an aditional assumption. That is that the explanatory variables do not display a high degree of multiple co-linearity. In other words they should not be correlated with each other. This is rarely the case in ecology. Multiple co-linearity does not prevent the use of such models, but it does raise some very tricky issues.

13.2 Example data

The effect of fragmentation of habitat as a result of human activities is a common theme in ecology. We will look at an example presented by Zuur et al (2010). Forest bird abundances expressed as an index (details of how this was measured are not given) were observed in 56 forest patches in south-eastern Victoria, Australia. The aim of the study was to relate the index of forest bird abundance to six habitat variables; size of the forest patch, distance to the nearest patch, distance to the nearest larger patch, mean altitude of the patch, year of isolation by clearing, and an index of stock grazing history (1 = light, 5 = intensive).

Zuur's analysis is given in the appendix of the book. In our analysis the grazing index will be treated as a numerical variable on an ordinal scale. Zuur treats it as a factor. This does not alter the main conclusions of the analysis, and helps to clarify and illustrate some addional issues.

```
d<-read.csv("https://tinyurl.com/aqm-data/loyn.csv")
```

13.3 Muliple regression

The three steps in building a regression model with many explanatory variables are.

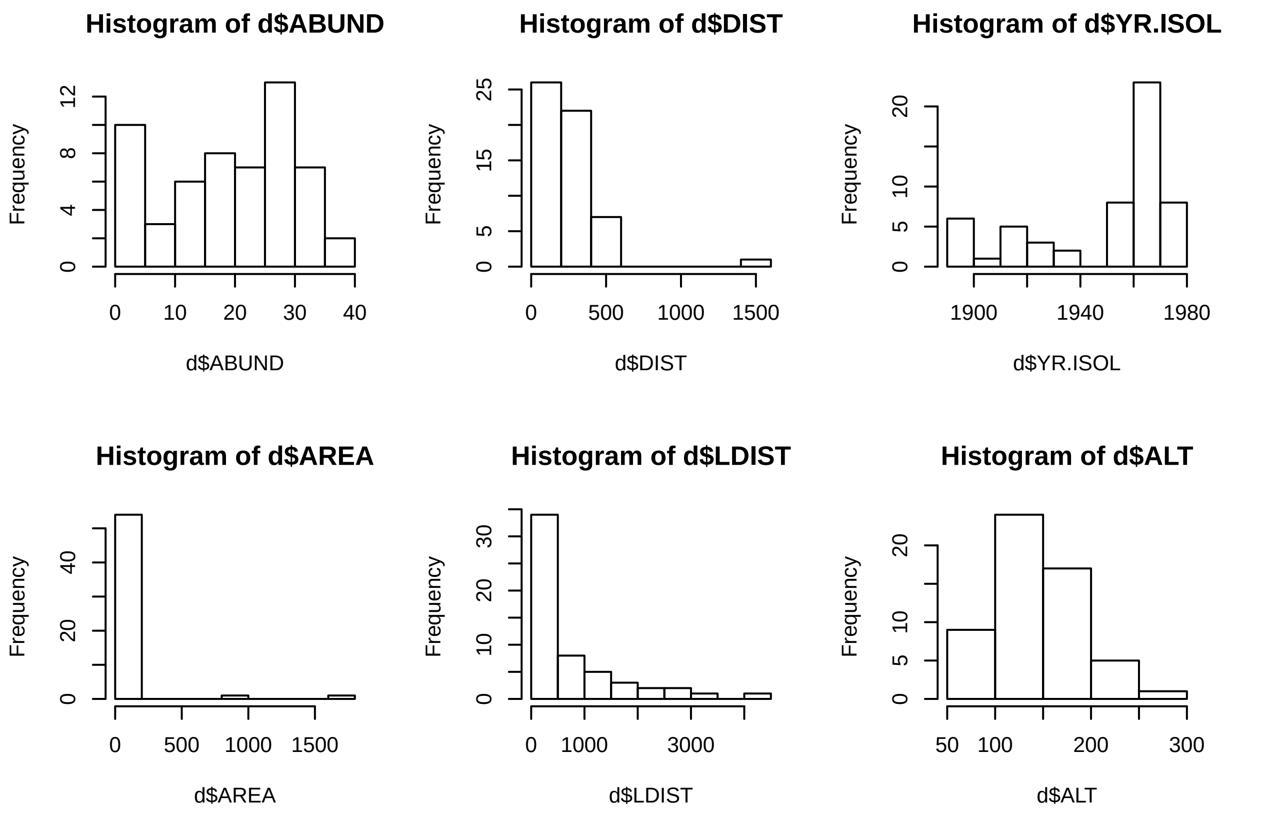
1. Look at the distributions of the explanatory and response variables with particular attention to influential outliers.
2. Look for collinearity of the explanatory variables.
3. Investigate the relationship between the response variable and the explanatory variables

13.3.1 Analysis of the distribution of variability

The first step in any analysis involving regression is to investigate the distribution of each variable and look for potential issues with influential outliers. Rembember that the assumption of normality in regression aplies to the residuals, not the explanatory variables themselves. Equally spaced observations (i.e. forming a flat, uniform distribution) would be ideal. We more or less have this for grazing, although the range is slightly limited and the measurements will have error due to the subjective judgment of grazing intensity. A symetrical,more or less normal distribution would also be suitable. However highly skewed distributions cause serious problems.

Let's first look at the histograms.

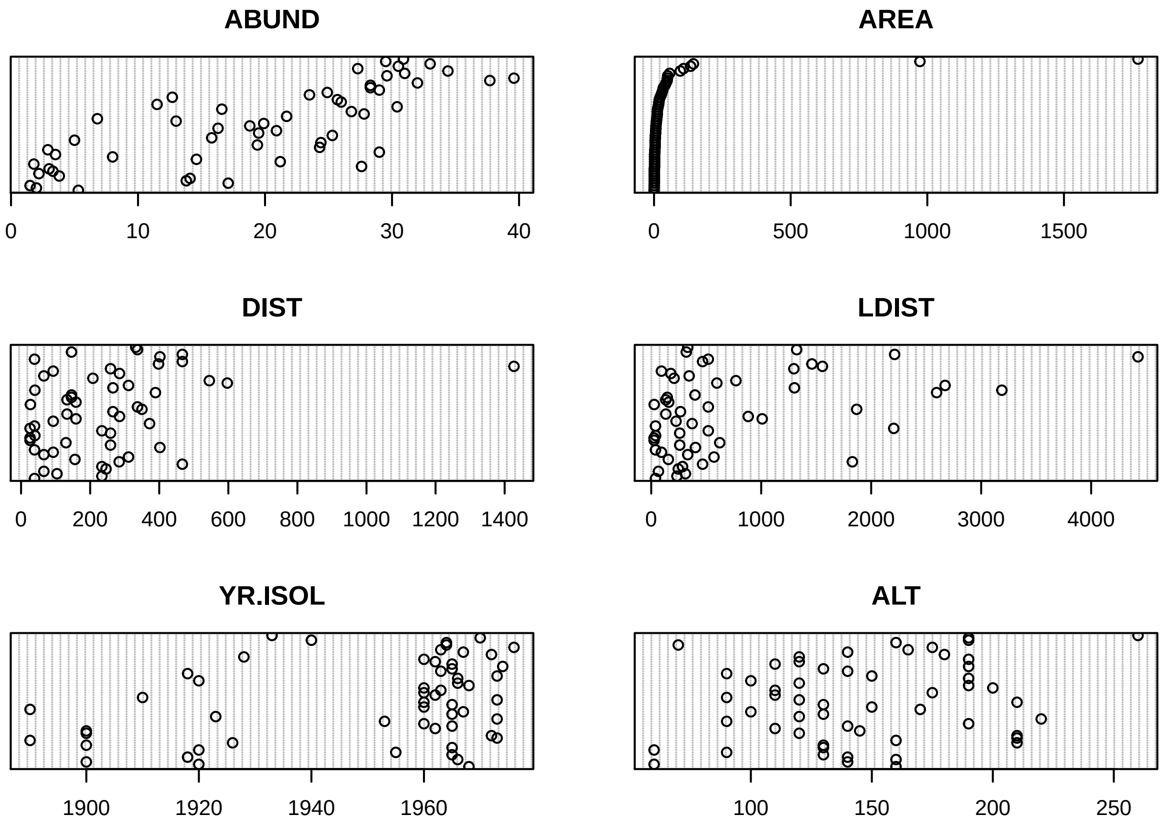
```r
par(mfcol=c(2,3))
hist(d$ABUND)
hist(d$AREA)
hist(d$DIST)
hist(d$LDIST)
hist(d$YR.ISOL)
hist(d$ALT)
```



There are issues with both the measures of distance and area. However in this case the outliers are not the result of errors. A few forest patches are much larger than the rest. This type of pattern is very common. We do not usually want to remove these sort of outliers, as they are part of the phenomenon that we are studying. However if a single observation falls in the tail for several of the variables at once it will have very high leverage and potentially exert a great deal of influence over the result.

Another way of looking for patterns in the outliers is to form Cleveland dotplots. These are very simple diagnostic tools. The value of the observation is simply plotted against the order of the data. This can help to show points that fall far from the rest of the values for several variables.

```
par(mfrow = c(3, 2), mar = c(3, 3, 3, 1))
dotchart(d$ABUND, main = "ABUND")
dotchart(d$AREA, main = "AREA")
dotchart(d$DIST, main = "DIST")
dotchart(d$LDIST, main = "LDIST")
dotchart(d$YR.ISOL, main = "YR.ISOL")
dotchart(d$ALT, main = "ALT")
```



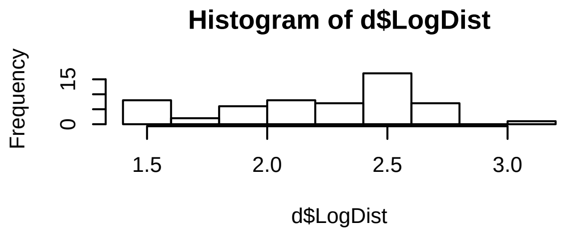
We can see that there are two observations with high AREA values, one observation with a high DIST value, and a couple of observations with high LDIST values. These are all different forest patches. If the same patch had much larger values of all variables, then it probably should be dropped from the analysis, as it could exert too much influence on the results. At the very least the analysis should be conducted twice, once with the oulier included and then with the outler removed.

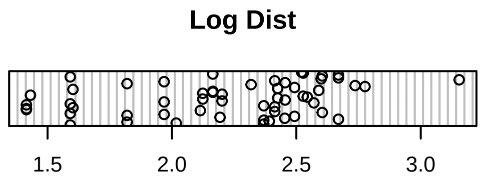
13.3.2 Transformation

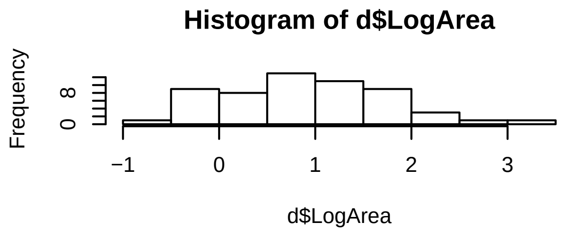
The alternative to dropping outliers is to apply a transformation. This is aimed at “pulling in”” the tail of the distribution in order to give the variables better statistical properties for modelling. We will try a log transformation and look at the results.

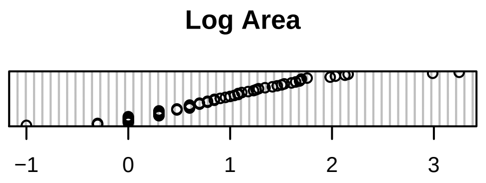
```
par(mfcol=c(3,2))
d$LogDist<-log10(d$DIST)
```

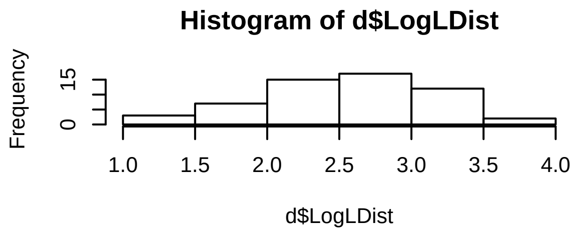
```
d$LogArea<-log10(d$AREA)
d$LogLDist<-log10(d$LDIST)
hist(d$LogDist)
hist(d$LogArea)
hist(d$LogLDist)
dotchart(d$LogDist,main="Log Dist")
dotchart(d$LogArea,main="Log Area")
dotchart(d$LogLDist,main="Log LDist")
```

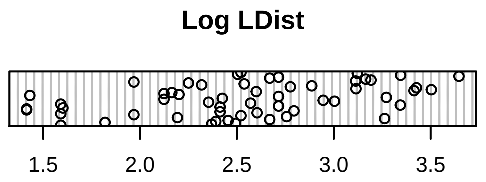












The transformations seem to have effectively neutralised the outliers, so we can proceed to the next step using all the data.

13.4 Collinearity

To assess collinearity, we will use three tools: Pairwise scatterplots, correlation coefficients, and variance inflation factors (VIF). The first two can be combined in one graph with some R code that is modified from the pairs help file. The modified function can be loaded as a script from the course site.

```
source("https://tinyurl.com/aqm-data/QEScript")
```

We first select the variables that we are interested in. You may want to look at the data frame again with str to check the order. We can quickly subset the data to only include the 2nd and the 6th to 11th columns using the command below.

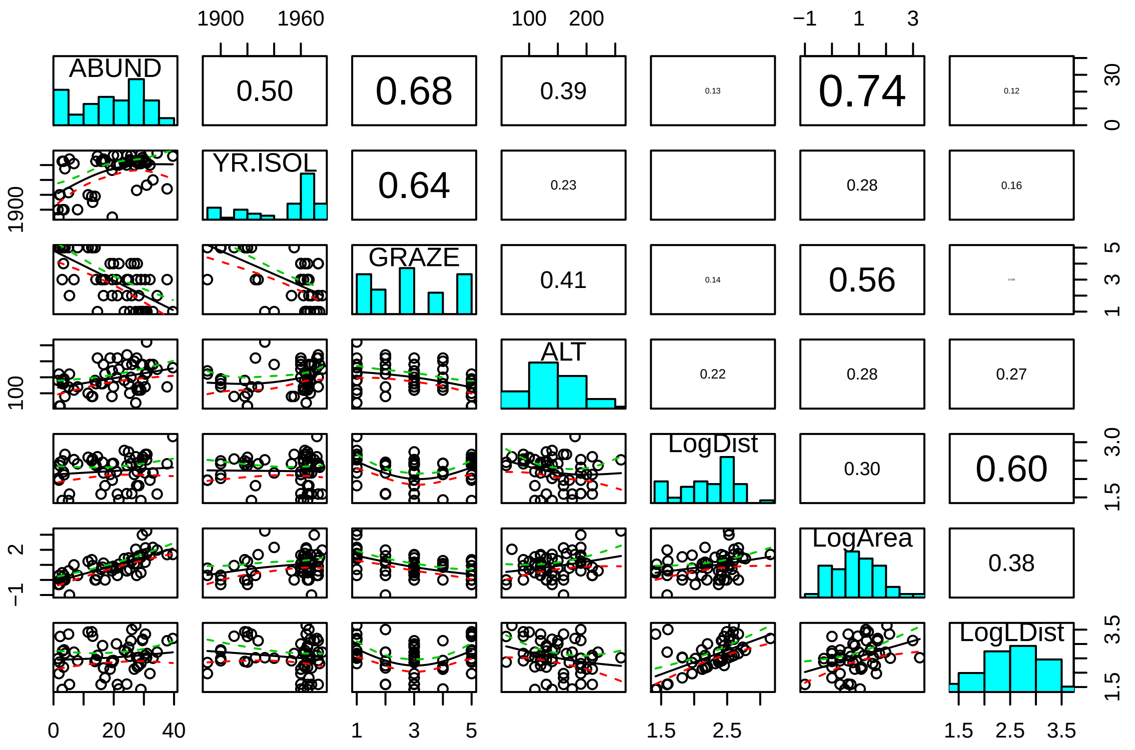
```
d1<-d[,c(2,6:11)]
str(d1)
```

```
## 'data.frame':    56 obs. of  7 variables:
```

```
## $ ABUND   : num  5.3 2 1.5 17.1 13.8 14.1 3.8 2.2 3.3 3 ...
## $ YR.ISOL : int  1968 1920 1900 1966 1918 1965 1955 1920 1965 1900 ...
## $ GRAZE   : int  2 5 5 3 5 3 5 5 4 5 ...
## $ ALT     : int  160 60 140 160 140 130 90 60 130 130 ...
## $ LogDist : num  1.59 2.37 2.02 1.82 2.39 ...
## $ LogArea : num  -1 -0.301 -0.301 0 0 ...
## $ LogLDist: num  1.59 2.37 2.49 1.82 2.39 ...
```

Now the function we have loaded produces a scatterplot of each variable against each of the rest, and shows the correlation coefficient in a font that is proportional to its size.

```
Xpairs(d1)
```



So there is a strong correlation between abundance, grazing and the logarithm of area. There is a weaker correlation with the year of isolation. These are the relationships we are interested in. However there are also correlations between the logarithm of the distance to the nearest patch and the logarithm of the distance to the nearest large patch. This is not suprising if the nearest patch is also a large patch. We will only need to worry about this if either of the terms are included in a model.

Grazing is also correlated with year of isolation and log area. The problem with these correlations is that they potentially confound the interpretation. The degree of confounding depends on the strengthof the correlation. If, for example, all the fragments that are heavily grazed were also small it would be very hard to clearly attribute low abundance to grazing or to area.

You can look at the significance of the correlations using the function cor.prob that was also included in the script loaded above. This places the correlation coeficient below the diagonal in the matrix and its significance above. In this particular data set correlation coeficients of above 0.3 (log Area with Altitude) are significant.

```
round(cor.prob(d1),2)
```

```
##          ABUND YR.ISOL GRAZE  ALT LogDist LogArea LogLDist
## ABUND     1.00    0.00  0.00 0.00    0.35    0.00     0.39
## YR.ISOL   0.50    1.00  0.00  0.08    0.89    0.04     0.24
## GRAZE    -0.68   -0.64  1.00  0.00    0.29    0.00     0.80
## ALT       0.39    0.23 -0.41  1.00    0.10    0.04     0.04
## LogDist   0.13   -0.02 -0.14 -0.22    1.00    0.02     0.00
## LogArea   0.74    0.28 -0.56  0.28    0.30    1.00     0.00
## LogLDist  0.12   -0.16 -0.03 -0.27    0.60    0.38     1.00
```

We will look at the variance inflation factor after fitting some models.

13.5 Model selection

One of the most interesting uses of multiple regression is to establish how many variables might be involved in determining the response. Every time we use regression with observational data such as these we need to remember that correlation is not causation. We cannot unequivocally attribute a cause to the effect. However we can analyse the strength of the association and interpret this carefully. In order to achieve this we need to be aware of some of the pitfalls that arise as a result of multiple colinearity. Let's first look at the grazing and area effects.

We can fit an additive model simply by typing the names of the terms. We can then test their significance using anova.

```
lm.mod1<-lm(ABUND~GRAZE+LogArea,data=d1)
anova(lm.mod1)
```

```
## Analysis of Variance Table
##
## Response: ABUND
##           Df Sum Sq Mean Sq F value    Pr(>F)
## GRAZE      1 2952.3 2952.35  71.094 2.301e-11 ***
## LogArea    1 1184.6 1184.64  28.527 1.977e-06 ***
## Residuals 53 2200.9   41.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Grazing and LogArea are quite closely correlated. So we have a problem. If we type the model formula in a different order we get different p-values for the terms!

```
lm.mod2<-lm(ABUND~LogArea+GRAZE,data=d1)
anova(lm.mod2)
```

```
## Analysis of Variance Table
##
## Response: ABUND
##           Df Sum Sq Mean Sq F value    Pr(>F)
## LogArea    1 3471.0  3471.0  83.583 1.757e-12 ***
## GRAZE      1  666.0   666.0  16.038 0.0001946 ***
## Residuals 53 2200.9    41.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both terms are still significant, but the result is quite different.

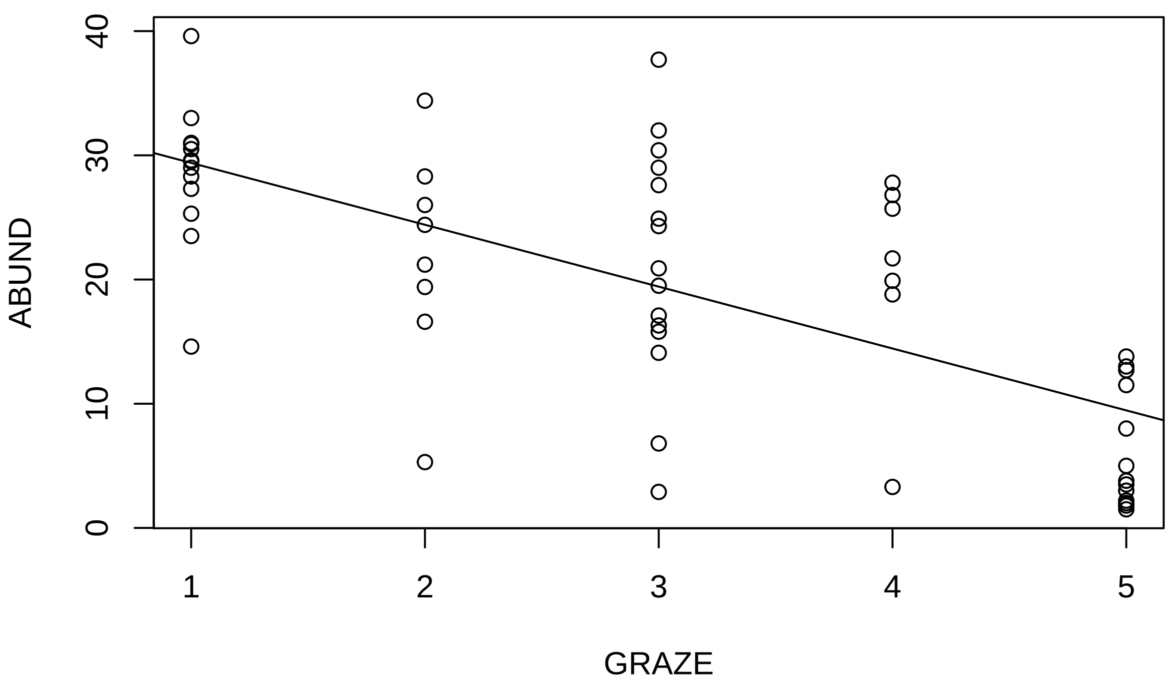
Why is this?

Let's break down the analysis in steps. First let's look at the relationship with grazing alone.

```
plot(ABUND~GRAZE,data=d1)
lm.mod.g<-lm(ABUND~GRAZE,data=d1)
anova(lm.mod.g)
```

```
## Analysis of Variance Table
##
## Response: ABUND
##           Df Sum Sq Mean Sq F value    Pr(>F)
## GRAZE      1 2952.3  2952.3   47.09 6.897e-09 ***
## Residuals 54 3385.6    62.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
abline(lm.mod.g)
```

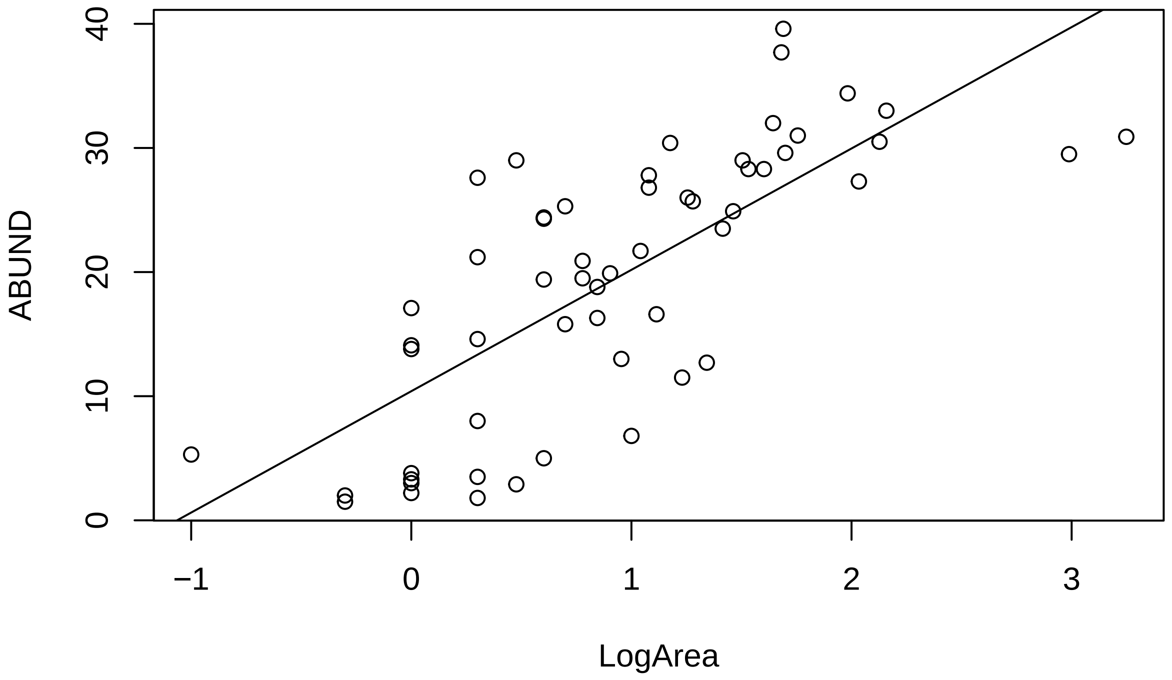


Now, imagine that we first looked at the strongest relationship that was apparent from the pairs plot. This is the relationship with log area. We could fit a simple regression model.

```
plot(ABUND~LogArea,data=d1)
lm.mod.a<-lm(ABUND~LogArea,data=d1)
anova(lm.mod.a)
```

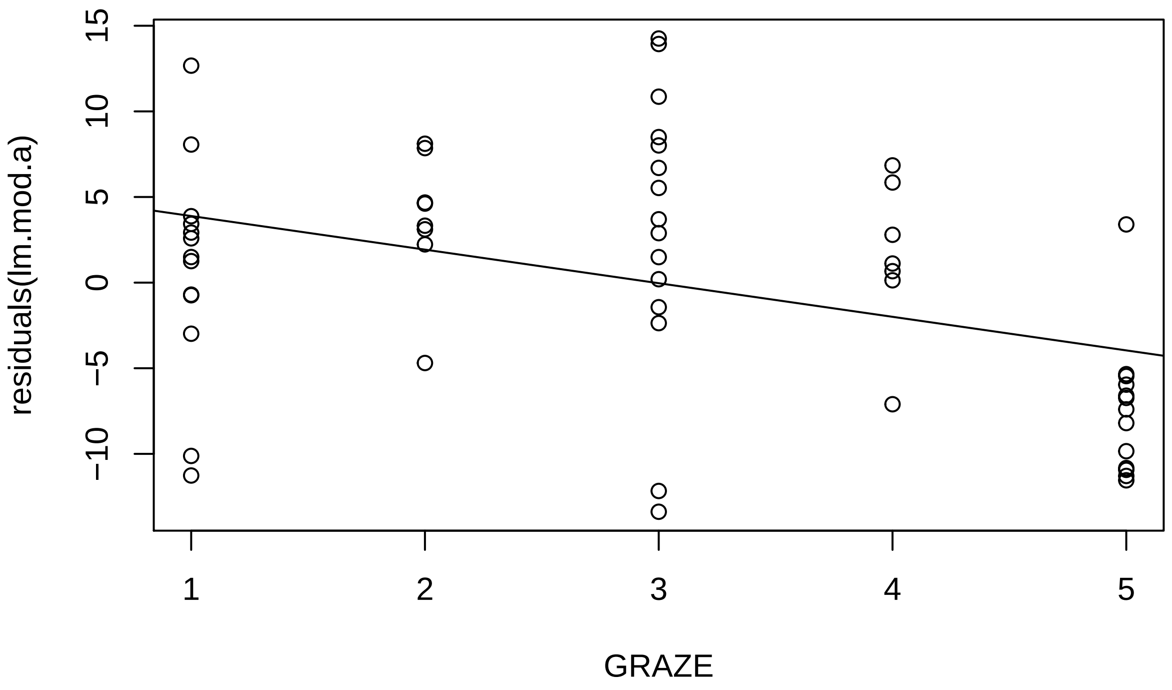
```
## Analysis of Variance Table
##
## Response: ABUND
##           Df Sum Sq Mean Sq F value    Pr(>F)
## LogArea    1 3471.0  3471.0  65.377 7.178e-11 ***
## Residuals 54 2866.9    53.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
abline(lm.mod.a)
```



The residuals from the regression is the variability that is not explained by area. We could then take this unexplained variability and see if any of this could still be explained by grazing. This would involve fitting a second model.

```
plot(residuals(lm.mod.a)~GRAZE,data=d1)
lm.mod.g2<-lm(residuals(lm.mod.a)~GRAZE,data=d1)
abline(lm.mod.g2)
```



```
anova(lm.mod.g2)
```

```
## Analysis of Variance Table
##
## Response: residuals(lm.mod.a)
##           Df  Sum Sq Mean Sq F value   Pr(>F)
## GRAZE      1  457.82  457.82  10.262 0.002279 **
## Residuals 54 2409.12   44.61
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The significance of grazing is greatly reduced after we have taken into account the effect of area. The first variable "soaks up"" a lot of the variablity that is also correlated with grazing. So less of the variability in the residuals can be explained by grazing. So, the order in which we analyse the variables is important. Although the sum of squares and the p-values are slightly different when we fit both terms together using the model formula for multiple regression, the general effect is the same. This does not occur if the variables are completely uncorrelated (orthogonal). Hence there is a need to look at the data very carefully whenever you build a multiple regression model.

13.5.1 Dropping terms

In the simple case of two explanatory variables we can get around the problem by dropping each of the terms in turn, refitting the model and looking at the difference. The results for mod1 (grazing first) and mod2 (log area first) are now identical.

```
drop1(lm.mod1,test="F")
```

```
## Single term deletions
##
## Model:
## ABUND ~ GRAZE + LogArea
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                2200.9 211.59
## GRAZE    1     666.0 2866.9 224.40  16.038 0.0001946 ***
## LogArea  1    1184.6 3385.6 233.71  28.527 1.977e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
drop1(lm.mod2,test="F")
```

```
## Single term deletions
##
## Model:
## ABUND ~ LogArea + GRAZE
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                2200.9 211.59
## LogArea  1    1184.6 3385.6 233.71  28.527 1.977e-06 ***
## GRAZE    1     666.0 2866.9 224.40  16.038 0.0001946 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

13.5.2 Stepwise model selection

The question underlying the analysis is to find out which of the set of explanatory variables are most closely associated with bird abundance. In one sense we already have an answer from the pairs plot. Some of the variables are clearly associated with abundance when used in a model on their own. But we want to explain as much of the variability as possible. How many terms are useful?

One way of addressing this is to fit a model with all the terms and then drop each in turn to check for significance.

```
lm.mod.full<-lm(ABUND~.,data=d1)
drop1(lm.mod.full,test="F")
```

```
## Single term deletions
##
## Model:
## ABUND ~ YR.ISOL + GRAZE + ALT + LogDist + LogArea + LogLDist
##          Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                1996.8 214.14
## YR.ISOL   1    108.83 2105.7 215.11  2.6705   0.10864
## GRAZE     1    131.07 2127.9 215.70  3.2163   0.07908 .
## ALT       1     27.02 2023.9 212.90  0.6630   0.41945
## LogDist   1      4.68 2001.5 212.27  0.1149   0.73609
## LogArea   1   1059.75 3056.6 235.98 26.0049 5.494e-06 ***
## LogLDist  1      3.80 2000.7 212.25  0.0933   0.76130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

However, again there is a problem with this if there is any collinearity. This initial analysis suggests that we can drop any one of the variables from the model with the exception of log Area without significantly reducing the amount of variance explained. However the problem is that if we dropped more variables from the model some of these variables would become significant again as they picked up variability that was explained by the lost variables.

13.5.3 The variance inflation factor

The variables that are more closely correlated with all the rest are those that are likely to have the least significance when dropped from the full model. We can rank the variables according to their colinearity with the rest by calculating the variance inflation factor.

```
library(car)
sort(vif(lm.mod.full))
```

```
##      ALT  LogDist  YR.ISOL  LogArea LogLDist    GRAZE
## 1.467937 1.654553 1.804769 1.911514 2.009749 2.524814
```

The variance inflation factor is quite a simple measure to understand. If we remove abundance from the data frame we are left only with the explanatory variables.

```
expl<-d1[,-1]
```

If we fit a model using all these variables in order to explain the variability in any one of the other explanatory variables we can get a value for R2. If the variable is closely correlated with all the rest this will be large. If we subtract this from one we get the ammount of variability not explained. The vif is simply the reciprocal of this.

For example for grazing.

```r
vif.mod<-lm(GRAZE~.,data=expl)
Rsq<-summary(vif.mod)$r.squared
Rsq
```

```
## [1] 0.6039312
```

```r
vif<-1/(1-Rsq)
vif
```

```
## [1] 2.524814
```

High values of vif are problematical, but there is no concensus of what is a high value. Zuur (2007) states that some statisticians suggest that values higher than 5 or 10 are too high (Montgomery and Peck 1992). In this case none of the vif values are that large, but the problem of lack of explanatory power in the presence of other variables is still apparent. Therefore a good approach is to rank the vif values as we have done here and think about the problem in context. Variables that do not explain much of the variability in the response variable are not going to be important whatever their vif. However a variable with a high vif value could explain a lot of the variability when used on its own in a model, but very little in combination with others. Grazing seems to be this sort of variable. The point here is that from an ecological perspective we would suspect that high grazing values should have an effect on bird density by altering habitat. So we would not want to drop the term from the model as a result of artefacts arising as a result of collinearity.

An alternative to using p-values for model selection is the use of AIC. AIC is quite conservative, in other words it tends to allow models to retain more parameters than some other methods for model selection.

We can use AIC for backward model selection using the step function in R. Backwards model selection using AIC is based on the same principle as the drop1 function, but terms are retained if they reduce AIC by more than 2 points. The process is repeated until dropping any further terms does not reduce AIC by more than 2 points.

```r
lm.mod.step<-step(lm.mod.full,test="F")
```

```
## Start:  AIC=214.14
## ABUND ~ YR.ISOL + GRAZE + ALT + LogDist + LogArea + LogLDist
##
##            Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - LogLDist  1      3.80 2000.7 212.25  0.0933   0.76130
## - LogDist   1      4.68 2001.5 212.27  0.1149   0.73609
## - ALT       1     27.02 2023.9 212.90  0.6630   0.41945
## <none>                  1996.8 214.14
## - YR.ISOL   1    108.83 2105.7 215.11  2.6705   0.10864
## - GRAZE     1    131.07 2127.9 215.70  3.2163   0.07908 .
## - LogArea   1   1059.75 3056.6 235.98 26.0049 5.494e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=212.25
## ABUND ~ YR.ISOL + GRAZE + ALT + LogDist + LogArea
##
##           Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - LogDist  1     12.64 2013.3 210.60  0.3159   0.57661
## - ALT      1     35.12 2035.8 211.22  0.8778   0.35331
## <none>                 2000.7 212.25
## - YR.ISOL  1    121.64 2122.3 213.55  3.0399   0.08739 .
## - GRAZE    1    132.44 2133.1 213.84  3.3098   0.07486 .
## - LogArea  1   1193.04 3193.7 236.44 29.8161 1.489e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=210.6
## ABUND ~ YR.ISOL + GRAZE + ALT + LogArea
##
##           Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - ALT      1     57.84 2071.1 210.19  1.4653   0.23167
## <none>                 2013.3 210.60
## - GRAZE    1    123.48 2136.8 211.94  3.1280   0.08294 .
## - YR.ISOL  1    134.89 2148.2 212.23  3.4169   0.07033 .
## - LogArea  1   1227.11 3240.4 235.25 31.0846 9.412e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=210.19
## ABUND ~ YR.ISOL + GRAZE + LogArea
##
##           Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                 2071.1 210.19
## - YR.ISOL  1    129.81 2200.9 211.59  3.2590   0.07682 .
## - GRAZE    1    188.45 2259.6 213.06  4.7315   0.03418 *
## - LogArea  1   1262.97 3334.1 234.85 31.7094 7.316e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(lm.mod.step)
```

```
##
## Call:
## lm(formula = ABUND ~ YR.ISOL + GRAZE + LogArea, data = d1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5159  -3.8136   0.2027   3.1271  14.5542
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -134.26065   86.39085  -1.554   0.1262
## YR.ISOL        0.07835    0.04340   1.805   0.0768 .
## GRAZE         -1.90216    0.87447  -2.175   0.0342 *
## LogArea        7.16617    1.27260   5.631 7.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.311 on 52 degrees of freedom
## Multiple R-squared:  0.6732, Adjusted R-squared:  0.6544
## F-statistic: 35.71 on 3 and 52 DF,  p-value: 1.135e-12
```

Notice how grazing became significant once more once altitude was dropped. This is probably because fragments that are more heavily grazed are in the valleys. We do have a useful model. But we need to be careful in the intepretation of the intercept. Because we used year of isolation in the model the intercept refers to the value at year zero, altitude zero, grazing zero and log area zero. This is not really helpful. However the slopes are interpretable. They represent the expected change in bird density for each unit change in the variables, when holding all other variables constant. We should report these values along with their confidence intervals, or standard errors.

```
confint(lm.mod.step)
```

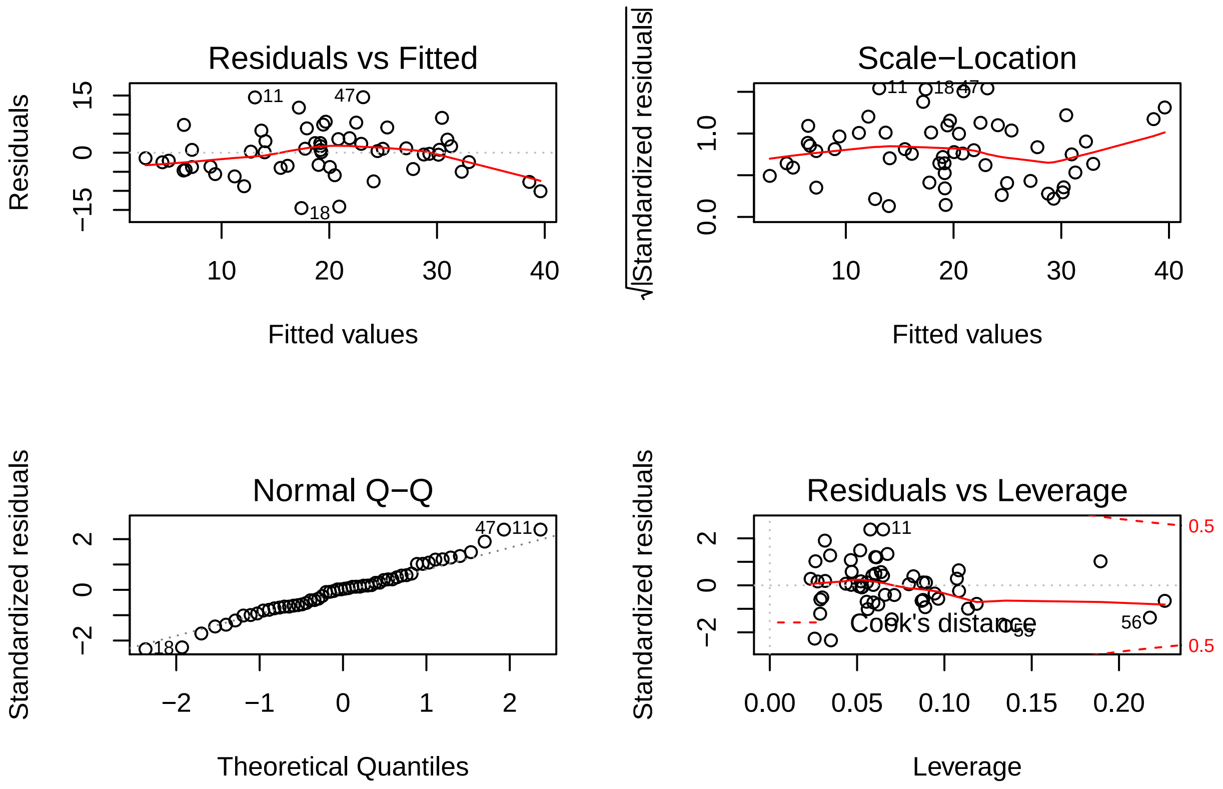
```
##                     2.5 %     97.5 %
## (Intercept) -3.076166e+02 39.0952733
## YR.ISOL     -8.739958e-03  0.1654462
## GRAZE       -3.656915e+00 -0.1473963
## LogArea      4.612500e+00  9.7198311
```

Notice that although the stepwise procedure based on AIC retained year of isolation in the model, the confidence interval includes zero and we cannot be sure if the effect is positive or negative at the 95% level. This is another way of looking at statistical significance. This term is not significant at the usual 0.05 cutoff. You might consider dropping it from the model. More on this later.

13.5.4 Diagnostics

Once the model has been decided on the usual diagnostics should be conducted.

```
par(mfcol=c(2,2))
plot(lm.mod.step)
```



The plots suggest that the main assumptions are not seriously violated. However you should be aware that spatial data such as these could lack independence due to autocorrelation. In order to analyse this we would need the coordinates, which are not provided in this data set.

13.6 Generalised Additive Models

In a previous class we looked at how flexible models can be fitted todata in order to capture more complex responses that are not well modelled by regression. General additive models can be used in a very similar way to multiple regression. They will show up any non linear response when plotted.

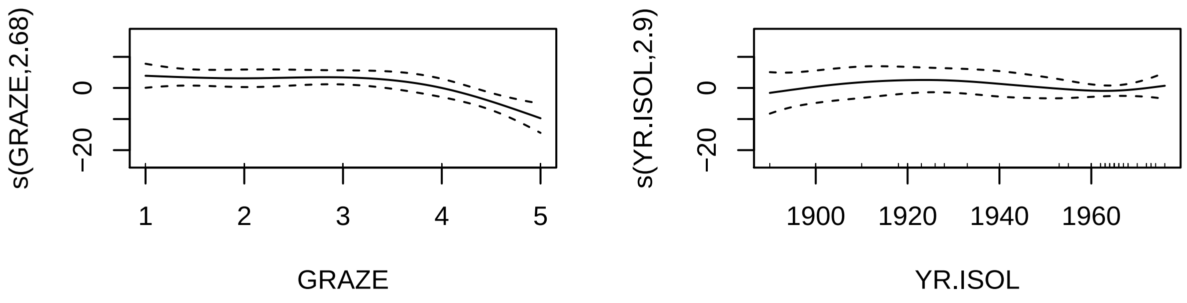
13.6.1 Fitting a gam with multiple variables

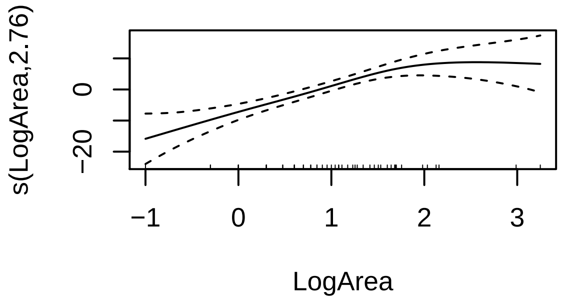
```
library(mgcv)
gam.mod1<-gam(ABUND~s(GRAZE,k=4)+s(LogArea)+s(YR.ISOL),data=d1)
summary(gam.mod1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ABUND ~ s(GRAZE, k = 4) + s(LogArea) + s(YR.ISOL)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.5143     0.7369   26.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(GRAZE)   2.682  2.909  6.078  0.00141 **
## s(LogArea) 2.760  3.479 11.485 3.12e-06 ***
## s(YR.ISOL) 2.900  3.542  0.842  0.55123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.736   Deviance explained = 77.6%
## GCV = 36.499  Scale est. = 30.41     n = 56
```

Note that in the case of grazing it was necessary to set the number of knots to four. This is because the default in mgcv is to begin with 10 and reduce the complexity through crossvalidation. We can't have 10 knots for grazing as there are only 5 values.

```
par(mfcol=c(2,2))
plot(gam.mod1)
```





13.6.2 Quick model selection for Gams

The cross validation algorithm used when fitting gams does not allow stepwise term deletion to be carried out. One quick method of model selection is to add “select=T”” when fitting a model with all the terms. This allows the smoother to reduce to having no knots, which effectively eliminates a term from the model.

```
gam.mod.sel<-gam(ABUND~s(YR.ISOL)+s(GRAZE,k=4)+s(ALT)+s(LogDist)+s(LogArea)+s(LogLDist),select=T,data=d
anova(gam.mod.sel)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ABUND ~ s(YR.ISOL) + s(GRAZE, k = 4) + s(ALT) + s(LogDist) +
##     s(LogArea) + s(LogLDist)
##
## Approximate significance of smooth terms:
##                   edf    Ref.df     F  p-value
## s(YR.ISOL)  1.104e-11 9.000e+00 0.000    0.710
## s(GRAZE)    2.181e+00 3.000e+00 8.207 1.36e-05
## s(ALT)      6.048e-01 3.000e+00 0.492    0.118
## s(LogDist)  4.183e-11 9.000e+00 0.000    1.000
## s(LogArea)  2.247e+00 8.000e+00 5.729 2.63e-09
## s(LogLDist) 8.506e-12 9.000e+00 0.000    0.962
```

13.6.3 Akaike weighting, relative strength of evidence approach.

As mentioned previously, in recent years information criteria (AIC) based approaches have become increasingly used for model selection by ecologists. This has been largely due to a string of influential papers by Burnham and Anderson. AIC is used in the automated stepwise procedure that we have seen already. However the popularity of the Burnham and Anderson model comparison approach arises as a result of the underlying philosophy. Burnham and Anderson do not like stepwise procedures, which are carried out without thought. They insist that simply letting the computer decide on a best model is a poor strategy. The alternative is to suggest a small subset of candidate models that make sense within the context of the study. AIC is then used to evaluate the strength of the evidence provided by the data for each model.

AIC stands for "Akaike's Information Criteria". To simplify a long story AIC is based on the concept of penalised likelihood. The (-2log) likelihood is a measure of the fit of the model, as is R2, and the penalty paid is 2 points for each parameter used in the model. All other things being equal models with lower AIC scores are better than those with high AIC scores, and a difference of 2 points in the (-2log) likelihood is worth paying for one parameter.

To use AIC in model selection using the protocol laid out by Burnham and Anderson there are a number of things to keep in mind.

- All the models in the set of candidates must use exactly the same set of observations and therefore be based on the same sample size n.
- All the models must use exactly the same response variable. For example do not compare one model using abundance with another using log(abundance)
- Models must use the same methods to calculate likelihoods. This is a technical issue, but in cases where normally distributed errors are asumed this can be taken as true.

13.6.4 The method.

- Choose a subset of models which can be justified as good candidates for the best model on both statistical and biological grounds.
- Calculate AIC for all models. Burnham and Anderson suggest using an adjusted value called AICc if AIC is small. In fact this could be used for all analyses.
- Identify the model with the smallest AIC. Denote its AIC as AIC_{min}. This is the best model. We could stop at this point but there is more information to extract from the models.
- Calculate the AIC differences, for each model.$\Delta_i = AIC_i - AIC_{min}$
- Compute the relative likelihood for each model. $RL_i = exp(-0.5\Delta_i)$
- Compute Akaike weights for each model. These are the normalized relative likelihoods. $w_i = \frac{RL_i}{\sum RL}$. The Akaike weights can be interpreted as probabilities that the given model is the best model.If we were to go back and obtain more data from the population and refit the same models again, then the Akaike weight gives the probability that a given model would be judged the best model on repeated sampling.

The package MuMIn in R will do steps 2 to 6 for us.

```r
library(MuMIn)
gam.mod1<-gam(ABUND~s(GRAZE,k=4)+s(LogArea),data=d1)
gam.mod2<-gam(ABUND~s(GRAZE,k=4)+s(LogArea)+s(YR.ISOL),data=d1)
gam.mod3<-gam(ABUND~s(GRAZE,k=4)+s(LogArea)+s(ALT),data=d1)
gam.mod4<-gam(ABUND~s(YR.ISOL)+s(GRAZE,k=4)+s(ALT)+s(LogDist)+s(LogArea)+s(LogLDist),data=d1)
model.sel(gam.mod1,gam.mod2,gam.mod3,gam.mod4)
```

```
## Model selection table
##          (Int) s(GRA,4) s(LgA) s(YR.ISO) s(ALT) s(LgD) s(LLD) df   logLik
## gam.mod1 19.51        +      +                                  6 -173.231
## gam.mod3 19.51        +      +                +                8 -171.390
## gam.mod2 19.51        +      +         +                      10 -169.964
## gam.mod4 19.51        +      +         +      +      +      + 13 -167.720
##           AICc delta weight
## gam.mod1 361.9  0.00  0.510
## gam.mod3 362.3  0.40  0.417
## gam.mod2 365.9  3.95  0.071
## gam.mod4 372.9 10.98  0.002
## Models ranked by AICc(x)
```

So, if we take this information theoretic approach the gam model which includes only grazing and log area has a 51% probabliity of being the best within the set of models we proposed, while the model with an additional term for altitude has a 42% probability.

We can also compare GAM's with linear models, as the likelihoods are calculated in the same way.

```
model.sel(gam.mod1,lm.mod.step)
```

```
## Model selection table
##               (Int) s(GRA,4) s(LgA)    GRA   LgA  YR.ISO class df   logLik
## gam.mod1      19.51        +      +                        gam  6 -173.231
## lm.mod.step -134.30                 -1.902 7.166 0.07835    lm  5 -180.555
##              AICc delta weight
## gam.mod1    361.9   0.0  0.995
## lm.mod.step 372.3  10.4  0.005
## Models ranked by AICc(x)
```

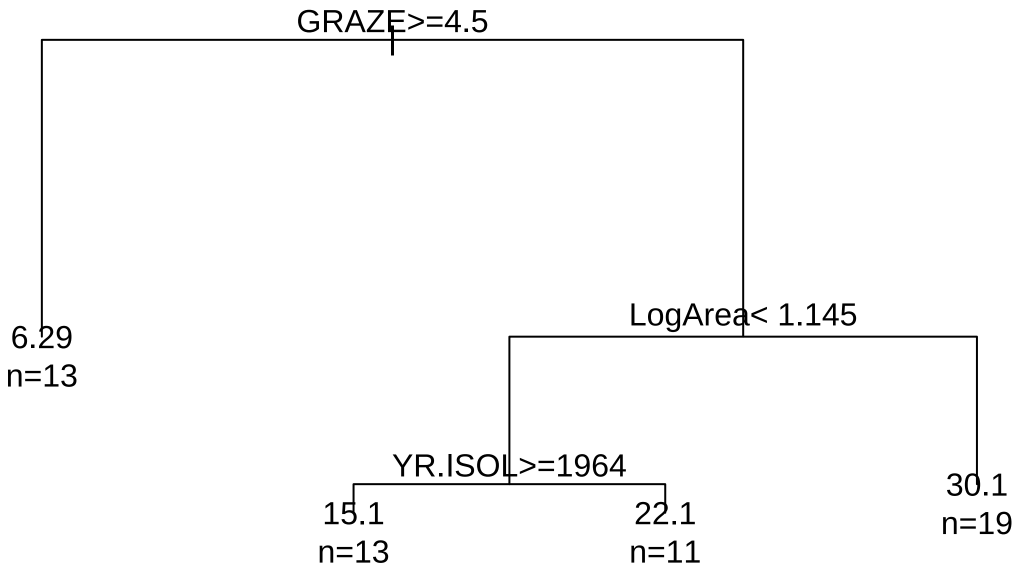
The GAM wins easily. This is as a result of the GAM finding the correct form for the grazing response rather than assuming a straight line.

13.7 Tree models

One of the problems with multiple regression and generalised additive models is that possible interactions between variables are not included. Zuur et al suggest that grazing could be included as a categorical factor, which allows some interactive effects to be looked at by taking a similar approach to the one we used for analysis of covariance. However interactions between two numerical variables cannot be easily captured. One easy way around this is to use tree models. The idea behind a tree model is to find a series of binary splitting rules that divide the response data into the most homogeneous subsets. The easiest way to follow this is with an example.

13.7.1 Fitting and plotting a tree model

```
library(rpart)
tree.mod<-rpart(ABUND~.,data=d1)
par(xpd=T)
plot(tree.mod)
text(tree.mod,use.n=T,digits=3)
```



This has a direct interpretation. We read the tree as a set of rules. If a rule is true we take the left hand branch, if it is false the right hand branch. The first split is based on the variable that can explain most of the deviance when used in this way. This is grazing. So if grazing is >=4.5 i.e. takes values of 5 on the ordinal scale we do not need to consider any other variables. The mean abundance levels for these sites is 6.3 based on 13 observations. If grazing pressure is low, some more explanatory variables are important. Large fragments with an area over log10(1.145)= 14 hectares have a mean abundance of 30.1. The final split for smaller fragments is less easy to explain. Fragments isolated after 1964 have a slightly lower mean abundance than those isolated earlier. However R starts off using an arbitrary degree of complexity. Maybe this tree has overfit to the data? We can test this by seeing if the tree can be pruned.

13.7.2 Pruning the tree

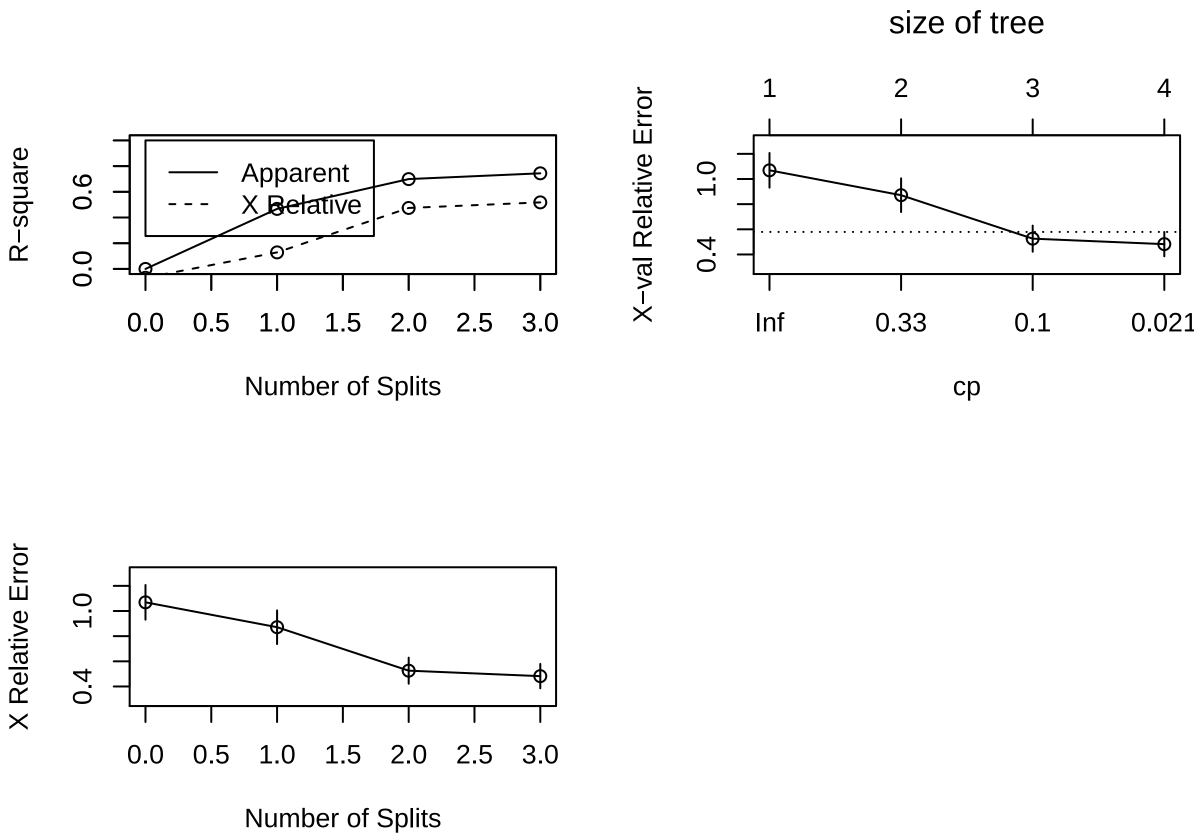
The theory behind tree models is presented in Zuur and Crawley. The most difficult element to understand is how to select an optimum tree based on the complexity parameter. The method has some similarity to the use of AIC for model selection. It is based on penalised likelihood (measured as the deviance). We want a tree that fits the data as well as possible, but uses as few parameters (splits) as possible. The default stopping rule that produced the first tree we looked at may have gone too far. We can get an R squared value (variability, or deviance explained) for each split along with the relative error calculated directly or as a result of cross validation. Relative error is 1-R squared. Cross validation produces confidence intervals for this error. We do not need to go into any more gory details regarding how cross validation is run, apart from mentioning that the cross validation error will always be higher than the apparent error as it is based on simulations using part of the data to fit a model and the rest to validate it.

```r
par(mfcol=c(2,2))
rsq.rpart(tree.mod)
```

```
##
## Regression tree:
## rpart(formula = ABUND ~ ., data = d1)
##
## Variables actually used in tree construction:
## [1] GRAZE   LogArea YR.ISOL
##
## Root node error: 6337.9/56 = 113.18
##
## n= 56
```

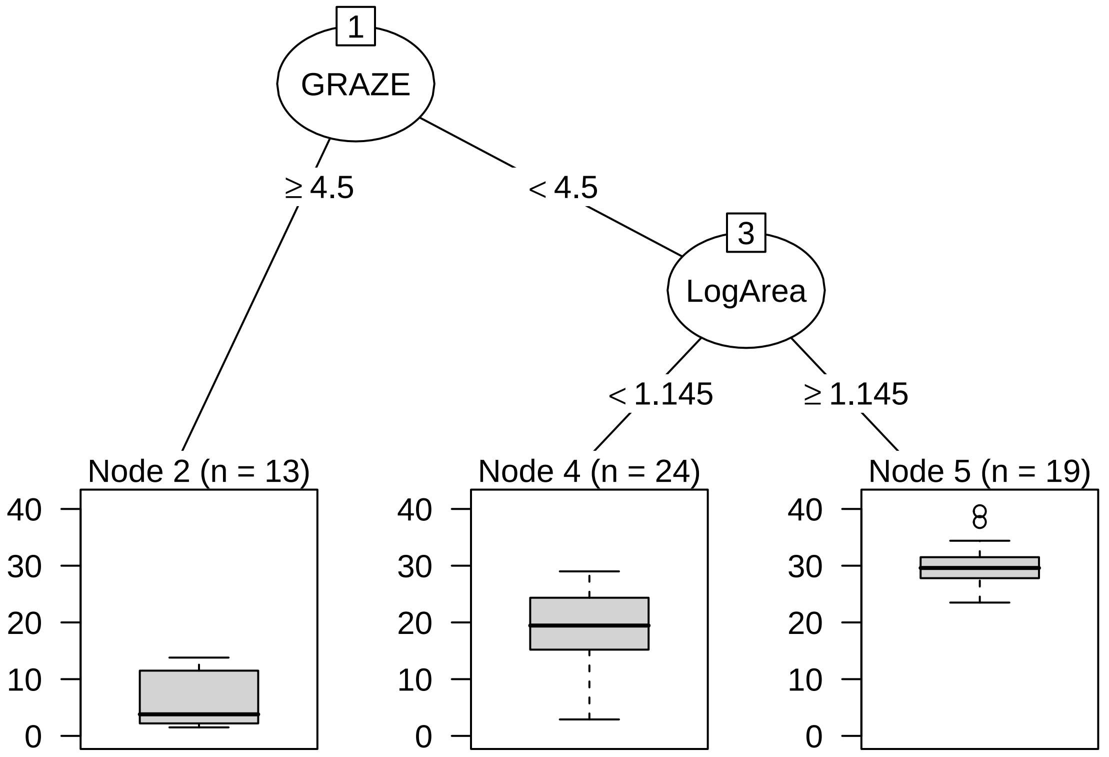
```
##
##         CP nsplit rel error  xerror     xstd
## 1 0.466991      0   1.00000 1.06900 0.137732
## 2 0.232025      1   0.53301 0.87130 0.134078
## 3 0.044967      2   0.30098 0.52570 0.103799
## 4 0.010000      3   0.25602 0.48218 0.096976
```

```
plotcp(tree.mod)
```



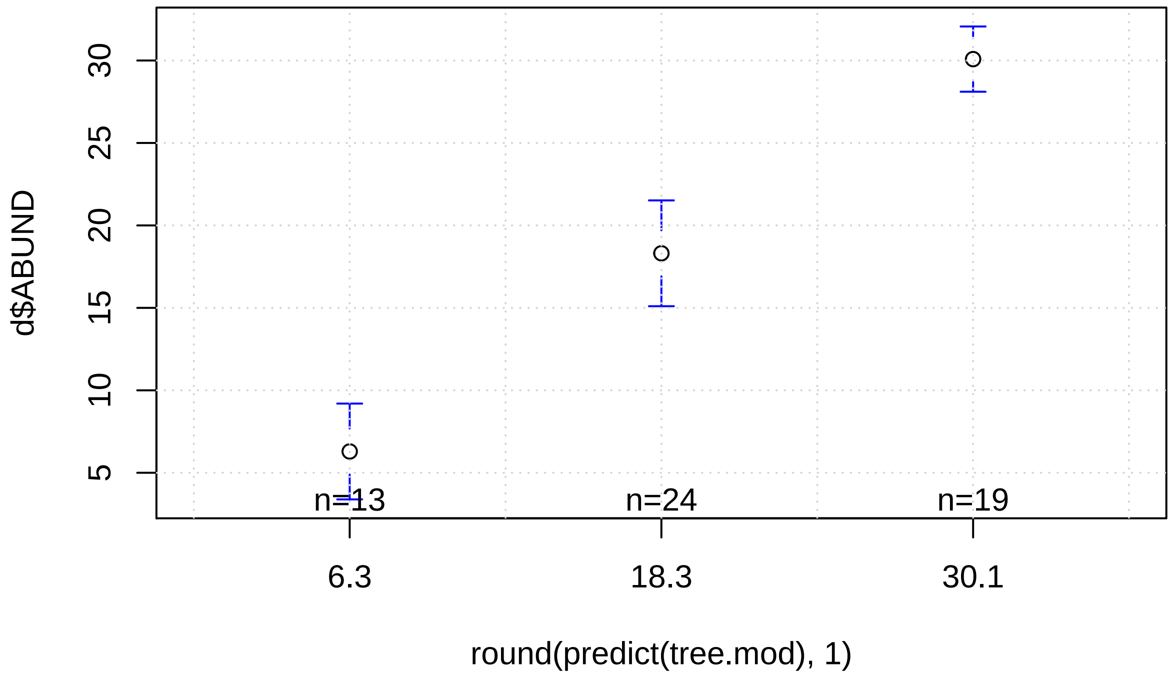
It should be apparent from the figures that adding a third split does not result in much gain in variance explained (R squared) or much reduction in relative error. So we can prune the tree using a complexity parameter of 0.1 (see the third figure above). We can also get a nicer figure using the partykit library.

```
tree.mod<-prune(tree.mod,cp=0.1)
library(partykit)
plot(as.party(tree.mod), digits=3)
```



We can also look at confidence intervals for the node values. These should not overlap if the splits are significant.

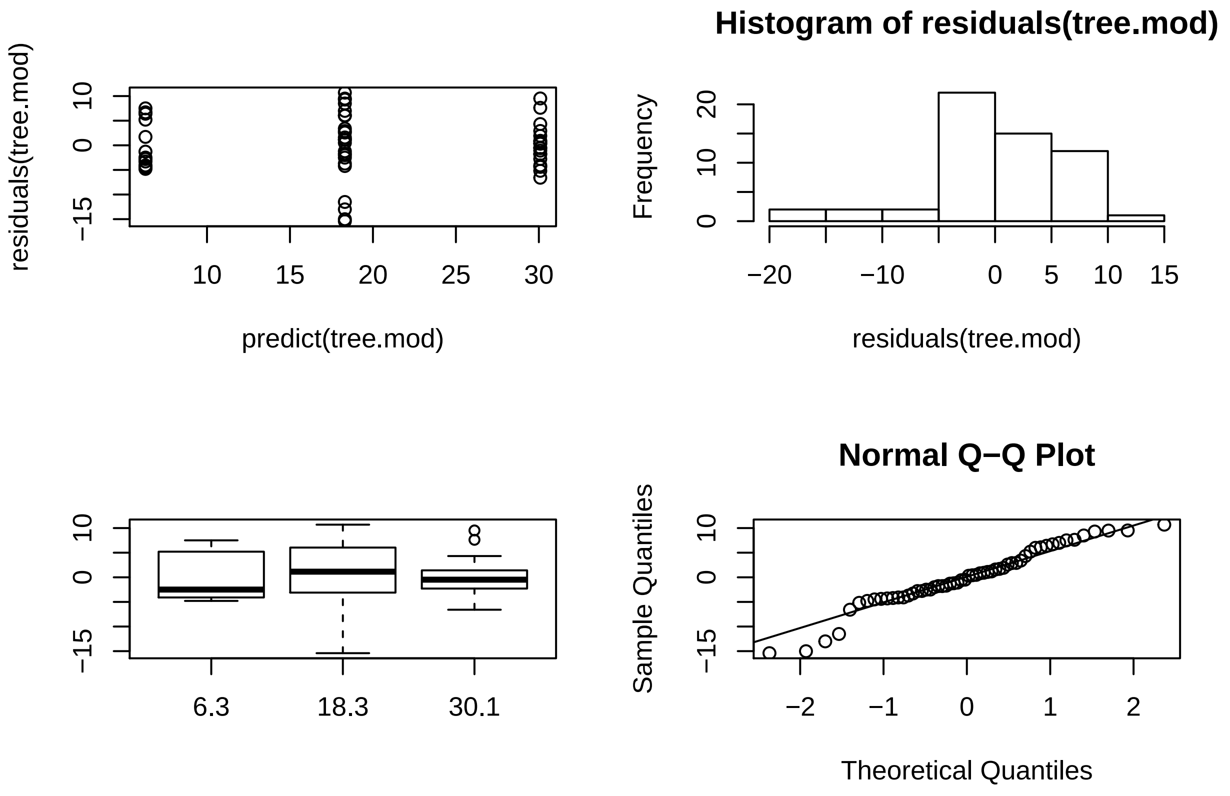
```
library(gplots)
plotmeans(d$ABUND~round(predict(tree.mod),1),connect=F)
grid()
```



So, the analysis leads to similar conclusions regarding the importance of the variables as multiple regresion. Date of isolation does not seem to be worth keeping in the model. While conservative fitting procedures may retain this term, the effect is not strong enough to be considered important. It is also ambiguous and hard to interpret in a meaningful way.

A point to be aware of is that the deviance is calculated using the sum of squares. Outliers and points with high leverage can still influence the results of the analysis. Homogeneity of variance is not assumed, as variance explained is calculated for each split separately. However it is a good idea to carry out some basic diagnistics in order to identify problematic observations that are not well predicted by the model and to check that the residuals are at least approximately normally distributed.

```r
par(mfcol=c(2,2))
plot(residuals(tree.mod)~predict(tree.mod))
boxplot(residuals(tree.mod)~round(predict(tree.mod),1))
hist(residuals(tree.mod))
qqnorm(residuals(tree.mod))
qqline(residuals(tree.mod))
```



The R squared of the tree model is 1-0.3= 0.7. This is comparable to the value we got using GAM and it is higher than the value for conventional multiple regression. The GAM and tree models show clearly that the effect of grazing is only important at the highest level of the index. The tree model also points out a possible interaction. At high levels of grazing there seems to be no further effect of area on abundance. This may be because high levels of grazing have altered the forest habitat so much that it is no longer suitable for many types of forest bird.

Tree models are particularly useful for quickly establishing which variables are most important in large data sets. Sometimes only a single split is found. If the same variable is used to split the data repeatedly the tree is effectively carrying out a regression on that variable, but with a series of steps instead of a smooth line.

13.8 Which technique to use?

The three techniques led to more or less the same conclusion regarding the relative importance of the explanatory variables. The only slight point of contention was whether altitude or year since isolation could have some effect. The GAM model and the tree model point out that the effect of grazing is only important at the highest level. This might also have been identified if we had used grazing as a factor in a linear model, which we did not try here. You may want to read Zuur et als analysis of these data.

Each of the modelling techniques have their own strengths and weaknesses.

- Multiple regression is a classic methodology that is understood by most researchers. It is simple to report the results in a conventional style (see this paper by the researcher who collected the data we have just analysed for an example). Many relationships are approximated by a straight line, so regression coefficients form a useful summary of effects. Using linear regression forces you to investigate the nature of the data thoroughly in order to ensure that the assumptions of the technique are not severely violated. With care, data sets with a high degree of multiple colinearity can be analysed, providing the consequences of colinearity are recognised when selecting the model. However the technique can lead to curvilinear responses being missed, along with interactions.
- Generalised additive models have become very popular in the ecological literature over the last few years. They allow complex responses to be identified. The relative importance of each explanatory variable can be assessed using GAMS just as in multiple regression. Linear regression can fail to spot curvilinear responses which can sometimes lead to misspecified models that do not capture an important element in the data. However GAM models are difficult to communicate to others. They can only really be plotted. So unless the data and the fitted model are provided they cannot be used for predicting new cases after the report has been written up.
- Tree models are simple to interpret, make relatively few assumptions and can show up interactions. Providing trees are not too complex the results can be easily communicated as a set of verbal rules. However the assumption that responses take the form of clear break points is often very misleading. Although tree models find the best break points, they are often still arbitrary and do not necessarily represent any real process.

There is no particular reason to prefer a single method over the others. In fact, until you have looked carefully at your data you are unlikely to know which technique will produce the most insight. It is important to end up with a defensible model which most closely meets the assumptions.

This class has shown that with practice and a little guidance it can be possible to combine all three techniques in order to find out as much as possible about the data. This is a sensible approach to take, even though it requires a bit more work. A write up of the analysis would sumarise the main findings from the technique that produced the most defensible model in the main body of the report, and include appendices showing the key results from alternative approaches. Ideally the data and R code used to fit the models would also be supplied so that a reader could check the validity of the conclusions.

Do be aware that none of these methods (with the possible exception of tree models) will work well in the case of data with large numbers of zeros, or with large and heterogeneous variability.

13.9 References

Anderson, D. R., Burnham, K. P. (2001). Kullback-Leibler information as a basis for strong inference in ecological studies. Wildlife Research, 28(2), 111-120.

Burnham, K. P, Anderson, D. R., Link, W. A., Johnson, D. H. (2001). Suggestions for presenting the results of data analyses. Journal of Wildlife Management, 65(3), 373-378.

Thompson, W. L., Anderson, D. R., Burnham, K. P. (2000). Null hypothesis testing: problems, prevalence, and an alternative. The Journal of Wildlife Management, 912-923.

Chapter 14

Analysis of multi-species data

One of the commonest tasks that quantitative ecologists face is the analysis of site by species matrices. Such data are routinely produced as a result of many types of ecological survey. Because multiple species are involved such data is always challenging to analyse. There are a lot of questions that can be addressed with multispecies data. Many of these involve analysis of species diversity in some respect.

Species diversity is classically regarded as consisting of three components.

$gamma = alpha + beta$

Alpha diversity is measured at the scale of some sampling unit (i.e. plot or quadrat). Gamma diversity is the overall diversity over the whole area that is being measured. Beta diversity is the difference between the two. Or is it? The issues involved in analysing diversity are extremely complex and could form the basis of a whole taught unit.

For the moment let's accept this simplified version of the issue for the pragmatic purpose of finding some methods that will allow us to work with a sites by species matrix.

14.1 Working with the sites by species matrix

One approach to working with the species by sites matrix would be to produce a measure that could be used as a response variable in some other model. In this class we will look at some simple measures of species diversity. The next class will apply some more complex methods in order to look at species composition. However you first need to become familiar with the format of the matrix in order to work with the data comfortably.

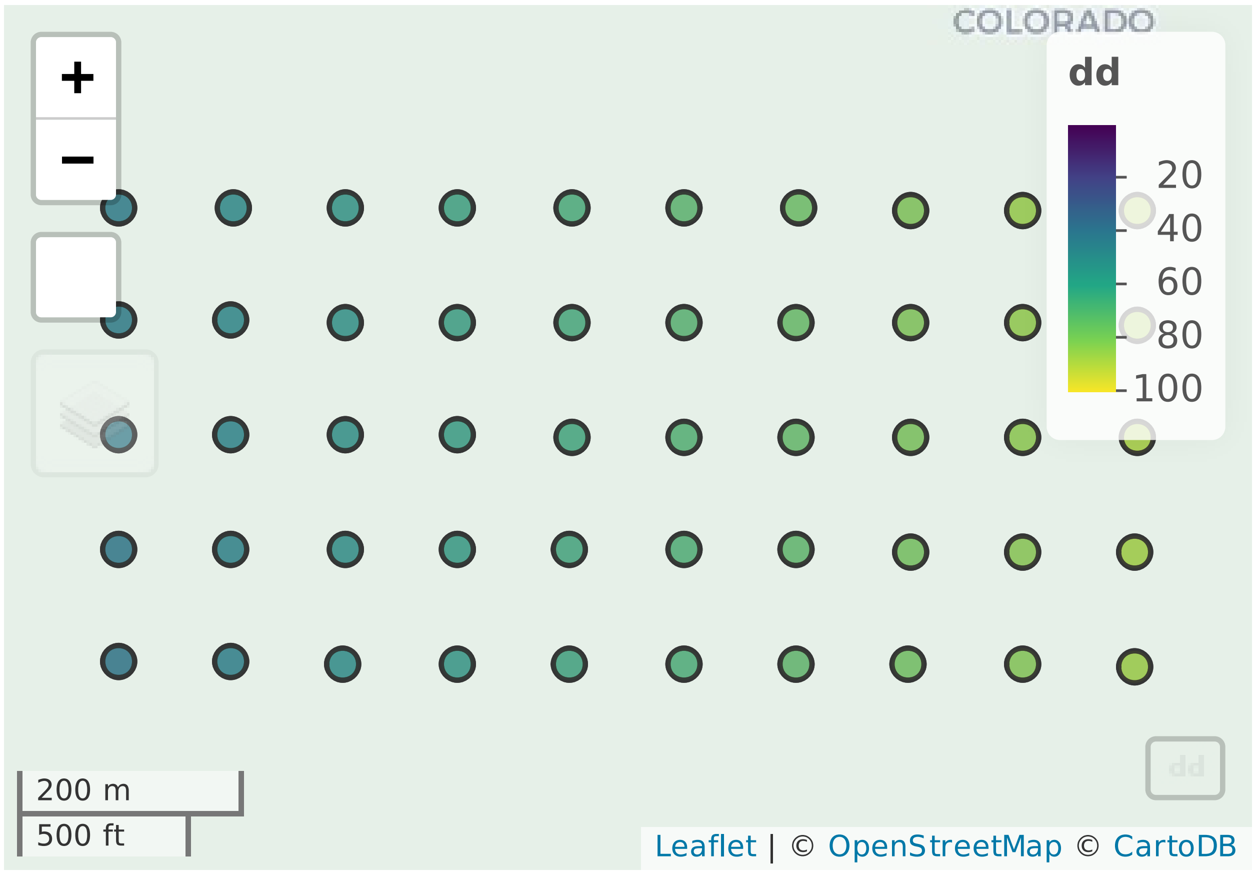
14.1.1 BCI data

Let's look at some realistically complex data from a classic study of tropical diversity. The Smithsonian institute has a permanent sample plot of 50 hectares of tropical forest located on the Island of Barro Colorado in Panama.

The data set consists of a grid of completely censused plot of 100m x 100m within which all trees over 5cm in diameter were counted and identified to species.

```r
library(rgdal)
x<- rep(seq(625754, 626654, by=100),each=5)
y<- rep(seq(1011569,  1011969, by=100),len=50)
coords<-data.frame(x,y)
```

```
dd<-data.frame(coords,id=1:100)
coordinates(dd)<-~x+y
proj4string(dd)<-CRS("+init=epsg:32617")
dd<-spTransform(dd, CRS("+init=epsg:4326"))
library(mapview)
m<-mapview(dd)
library(leaflet.extras)
m@map %>% addFullscreenControl()
```



```
library(vegan)
library(reshape)
data(BCI)
```

You can look at this matrix using

```
str(BCI)
```

```
## 'data.frame':    50 obs. of  225 variables:
##  $ Abarema.macradenia          : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Vachellia.melanoceras       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acalypha.diversifolia       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acalypha.macrostachya       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Adelia.triloba              : int  0 0 0 3 1 0 0 0 5 0 ...
##  $ Aegiphila.panamensis        : int  0 0 0 0 1 0 1 0 0 1 ...
##  $ Alchornea.costaricensis     : int  2 1 2 18 3 2 0 2 2 2 ...
##  $ Alchornea.latifolia         : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Alibertia.edulis            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Allophylus.psilospermus        : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Alseis.blackiana               : int  25 26 18 23 16 14 18 14 16 14 ...
##  $ Amaioua.corymbosa              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Anacardium.excelsum            : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Andira.inermis                 : int  0 0 0 0 1 1 0 0 1 0 ...
##  $ Annona.spraguei                : int  1 0 1 0 0 0 0 1 1 0 ...
##  $ Apeiba.glabra                  : int  13 12 6 3 4 10 5 4 5 5 ...
##  $ Apeiba.tibourbou               : int  2 0 1 1 0 0 0 1 0 0 ...
##  $ Aspidosperma.desmanthum        : int  0 0 0 1 1 1 0 0 0 1 ...
##  $ Astrocaryum.standleyanum       : int  0 2 1 5 6 2 2 0 2 1 ...
##  $ Astronium.graveolens           : int  6 0 1 3 0 1 2 2 0 0 ...
##  $ Attalea.butyracea              : int  0 1 0 0 0 1 1 0 0 0 ...
##  $ Banara.guianensis              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Beilschmiedia.pendula          : int  4 5 7 5 8 6 5 9 11 14 ...
##  $ Brosimum.alicastrum            : int  5 2 4 3 2 2 6 4 3 6 ...
##  $ Brosimum.guianense             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Calophyllum.longifolium        : int  0 2 0 2 1 2 2 2 2 0 ...
##  $ Casearia.aculeata              : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Casearia.arborea               : int  1 1 3 2 4 1 2 3 9 7 ...
##  $ Casearia.commersoniana         : int  0 0 1 0 1 0 0 0 1 0 ...
##  $ Casearia.guianensis            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Casearia.sylvestris            : int  2 1 0 0 0 3 1 0 1 1 ...
##  $ Cassipourea.guianensis         : int  2 0 1 1 3 4 4 0 2 1 ...
##  $ Cavanillesia.platanifolia      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cecropia.insignis              : int  12 5 7 17 21 4 0 7 2 16 ...
##  $ Cecropia.obtusifolia           : int  0 0 0 0 1 0 0 2 0 2 ...
##  $ Cedrela.odorata                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ceiba.pentandra                : int  0 1 1 0 1 0 0 1 0 1 ...
##  $ Celtis.schippii                : int  0 0 0 2 2 0 1 0 0 0 ...
##  $ Cespedesia.spathulata          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chamguava.schippii             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chimarrhis.parviflora          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Maclura.tinctoria              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chrysochlamys.eclipes          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chrysophyllum.argenteum        : int  4 1 2 2 6 2 3 2 4 2 ...
##  $ Chrysophyllum.cainito          : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Coccoloba.coronata             : int  0 0 0 1 2 0 0 1 2 1 ...
##  $ Coccoloba.manzinellensis       : int  0 0 0 0 0 0 0 2 0 0 ...
##  $ Colubrina.glandulosa           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cordia.alliodora               : int  2 3 3 7 1 1 2 0 0 2 ...
##  $ Cordia.bicolor                 : int  12 14 35 23 13 7 5 10 7 13 ...
##  $ Cordia.lasiocalyx              : int  8 6 6 11 7 6 6 3 0 4 ...
##  $ Coussarea.curvigemma           : int  0 0 0 1 0 2 1 0 1 1 ...
##  $ Croton.billbergianus           : int  2 2 0 11 6 0 0 4 2 0 ...
##  $ Cupania.cinerea                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cupania.latifolia              : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Cupania.rufescens              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cupania.seemannii              : int  2 2 1 0 3 0 1 2 2 0 ...
##  $ Dendropanax.arboreus           : int  0 3 6 0 5 2 1 6 1 3 ...
##  $ Desmopsis.panamensis           : int  0 0 4 0 0 0 0 0 0 1 ...
##  $ Diospyros.artanthifolia        : int  1 1 1 1 0 0 0 0 0 1 ...
##  $ Dipteryx.oleifera              : int  1 1 3 0 0 0 0 2 1 2 ...
##  $ Drypetes.standleyi             : int  2 1 2 0 0 0 0 0 0 0 ...
##  $ Elaeis.oleifera                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Enterolobium.schomburgkii       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Erythrina.costaricensis         : int  0 0 0 0 0 3 0 0 1 0 ...
##  $ Erythroxylum.macrophyllum       : int  0 1 0 0 0 0 0 1 1 1 ...
##  $ Eugenia.florida                 : int  0 1 0 7 2 0 0 1 1 3 ...
##  $ Eugenia.galalonensis            : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Eugenia.nesiotica               : int  0 0 1 0 0 0 5 4 3 0 ...
##  $ Eugenia.oerstediana             : int  3 2 5 1 5 2 2 3 3 3 ...
##  $ Faramea.occidentalis            : int  14 36 39 39 22 16 38 41 33 42 ...
##  $ Ficus.colubrinae                : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Ficus.costaricana               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.insipida                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.maxima                    : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.obtusifolia               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.popenoei                  : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Ficus.tonduzii                  : int  0 0 1 2 1 0 0 0 0 0 ...
##  $ Ficus.trigonata                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.yoponensis                : int  1 0 0 0 0 1 1 0 0 0 ...
##  $ Garcinia.intermedia             : int  0 1 1 3 2 1 2 2 1 0 ...
##  $ Garcinia.madruno                : int  4 0 0 0 1 0 0 0 0 1 ...
##  $ Genipa.americana                : int  0 0 1 0 0 0 1 0 1 1 ...
##  $ Guapira.myrtiflora              : int  3 1 0 1 1 7 3 1 1 1 ...
##  $ Guarea.fuzzy                    : int  1 1 0 1 3 0 0 2 0 3 ...
##  $ Guarea.grandifolia              : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Guarea.guidonia                 : int  2 6 2 5 3 4 4 0 1 5 ...
##  $ Guatteria.dumetorum             : int  6 16 6 3 9 7 8 6 2 2 ...
##  $ Guazuma.ulmifolia               : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Guettarda.foliacea              : int  1 5 1 2 1 0 0 4 1 3 ...
##  $ Gustavia.superba                : int  10 5 0 1 3 1 8 4 4 4 ...
##  $ Hampea.appendiculata            : int  0 0 1 0 0 0 0 0 2 1 ...
##  $ Hasseltia.floribunda            : int  5 9 4 11 9 2 7 6 3 4 ...
##  $ Heisteria.acuminata             : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ Heisteria.concinna              : int  4 5 4 6 4 8 2 5 1 5 ...
##  $ Hirtella.americana              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Hirtella.triandra               : int  21 14 5 4 6 6 7 14 8 7 ...
##  $ Hura.crepitans                  : int  0 0 0 0 0 2 1 1 0 0 ...
##  $ Hieronyma.alchorneoides         : int  0 2 0 0 0 0 0 0 1 0 ...
##   [list output truncated]
##  - attr(*, "original.names")= chr  "Abarema.macradenium" "Acacia.melanoceras" "Acalypha.diversifolia
```

```r
dim(BCI)
```

```
## [1]  50 225
```

The researchers therefore found 225 species in the 50 plots. The area is quite homogeneous, but there is a very high diversity of tree species. There are over 3000 species of tree in Panama.

Whether 225 is considered as a measure of gamma diversity (for the 50 plots) or alpha diversity (for 50 ha plots within an area with a higher gamma diversity clearly depends on the scale at which diversity is being observed and studies. There are complex issues here.

14.1.2 Reshaping the site by species matrix

Note that there are many zeros in the site by species matrix. As we have seen before, this is not strictly a "raw" data format, as it can be derived from a more compact table format. Here is how to interchange the two formats using the reshape package.

```
# Make a data frame with Site ID as one of the columns
bci<-data.frame(Site=1:50,BCI)
#Melt the data frame using the Site ID as the identifier.
bci<-melt(bci,id="Site")
# Remove zeros
bci<-bci[bci$value>0,]
```

To reshape this long format back into a sites by species matrix form, with a column for the site name you can run this line of code.

```
bci2<-data.frame(cast(bci,Site~variable,fill=0))
```

Check the format. You may often have to remove some of the columns in order to work with a matrix that only consists of species abundence. In this case

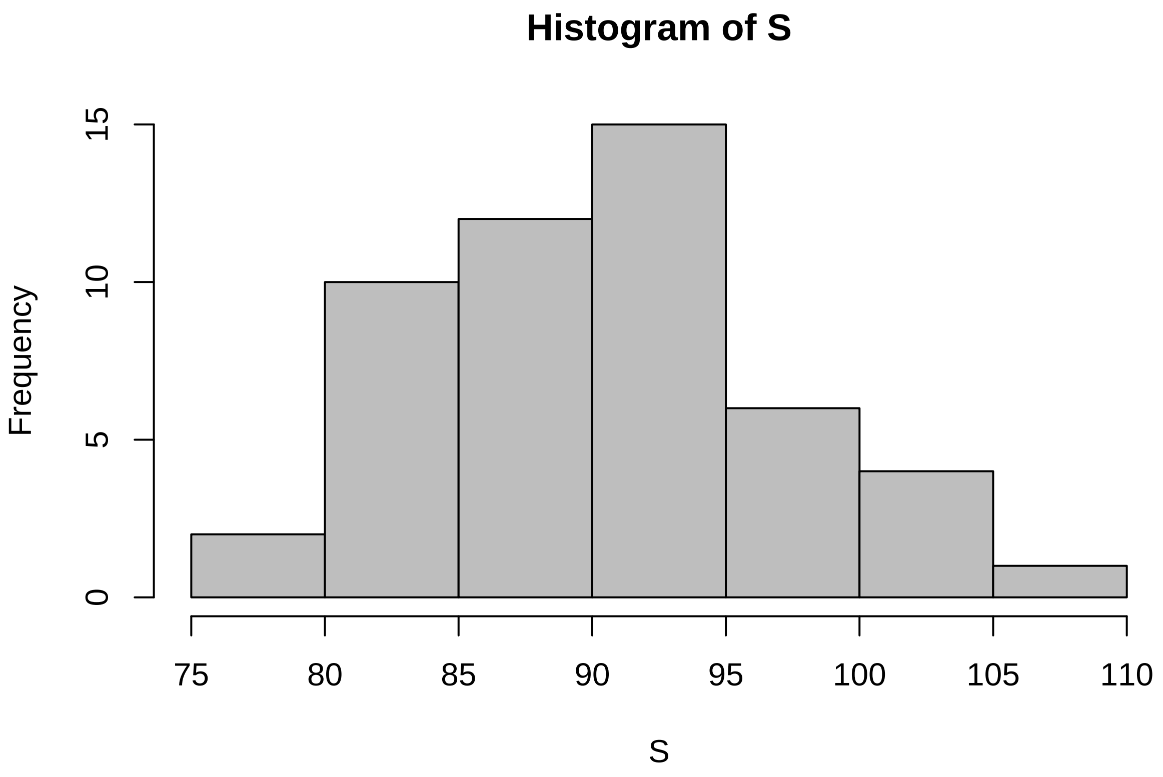
```
bcimat<-as.matrix(bci2[,-1])
```

Let's look at how many species are found in each plot. We can produce a vector of species richness using the specnumber function.

```
S<-specnumber(BCI)
summary(S)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   77.00   86.00   91.00   90.78   94.00  109.00
```

```
hist(S,col="grey")
```



So, species number could perhaps be used as a response variable if we had some explanatory variables for each sub plot. We will come back to this issue later.

14.1.3 Working with apply

Many of the methods for working with quadrat type data work directly on the broad, site by species matrix.

Although it is usually possible to use data frame objects in R for matrix based operations there is a difference between a data frame and a true matrix. A matrix can only hold numbers, while a data frame can hold any type of values. The "apply" command will work on any matrix, but would produce odd results on some data frames.

When you ask R to apply a function to a matrix it is like writing a function at the end of a set of columns or rows on an Excel spreadsheet and then dragging them across. This is only going to work if all the columns or rows contain the same sort of values. If we want to apply a function (say sum) to rows of a matrix (say BCI) we use a 1 to refer to rows and write apply(BCI,1,sum). If we want to apply the function to the columns we write apply(BCI,2,sum) So, for example to find the abundances of the species. We can sum over all the subplots to produce a vector of abundances.

```
abun<-apply(BCI,2,sum)
```

A few abundant species contribute a relatively large number of individuals to the total.

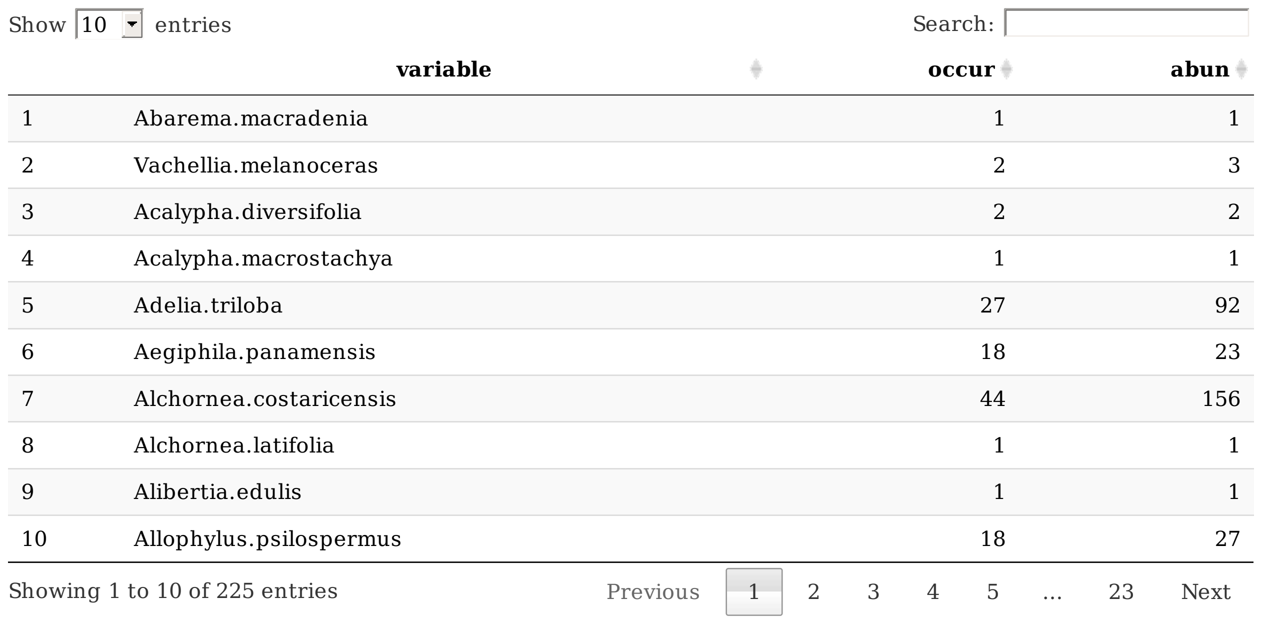
```
head(sort(abun,dec=T))
```

```
##   Faramea.occidentalis  Trichilia.tuberculata       Alseis.blackiana
##                   1717                   1681                    983
##      Oenocarpus.mapora       Poulsenia.armata Quararibea.asterolepis
##                    788                    755                    724
```

14.1.4 Working with the long format data frame

It can be more convenient to use dplyr to produce tables. To do this we need to work with the reshaped long data. When the matrix was put into the "molten" database form by the reshape operation species name became the variable and the count of individual trees the value. So this code will produce a table.

```
library(dplyr)
library(DT)
bci %>% group_by(variable) %>% summarise(occur=n(),abun=sum(value)) -> abunds
datatable(abunds)
```



Show 10 entries Search:

	variable	occur	abun
1	Abarema.macradenia	1	1
2	Vachellia.melanoceras	2	3
3	Acalypha.diversifolia	2	2
4	Acalypha.macrostachya	1	1
5	Adelia.triloba	27	92
6	Aegiphila.panamensis	18	23
7	Alchornea.costaricensis	44	156
8	Alchornea.latifolia	1	1
9	Alibertia.edulis	1	1
10	Allophylus.psilospermus	18	27

Showing 1 to 10 of 225 entries Previous 1 2 3 4 5 … 23 Next

We can calculate the relative abundances of these top ten species as percentages using R

```r
round(100*head(sort(abun,dec=T))/sum(abun),2)
```

```
##   Faramea.occidentalis  Trichilia.tuberculata       Alseis.blackiana
##                   8.00                   7.83                   4.58
##      Oenocarpus.mapora       Poulsenia.armata Quararibea.asterolepis
##                   3.67                   3.52                   3.37
```

So even though there are 225 species recorded, 20% of the observations are from only three species.

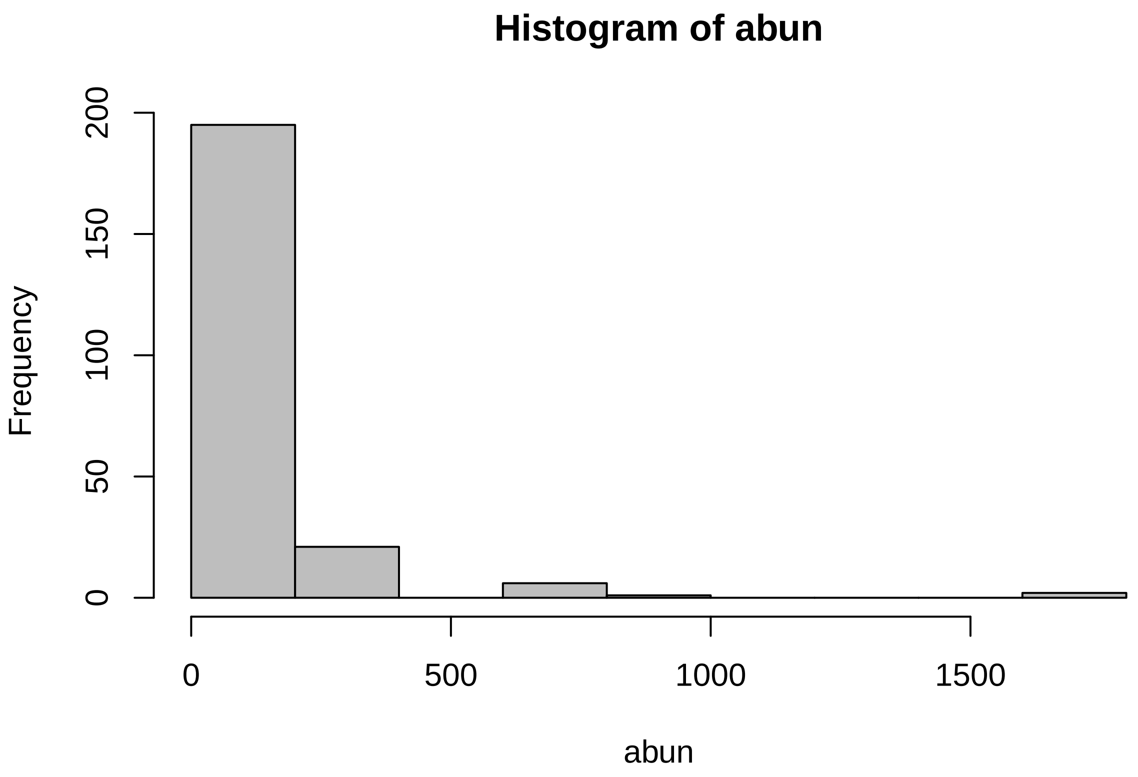
```r
summary(abun)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    7.00   25.00   95.36   82.00 1717.00
```

The median abundance is 25. 50% of the species have 25 or fewer individuals. This is a very typical pattern in diversity data, whether in the tropics or elsewhere. Most *individuals* belong to a handful of common species. However, most *species* have few individuals. To put it in a delberately perverse way common species turn out to be rare and rare species are common!

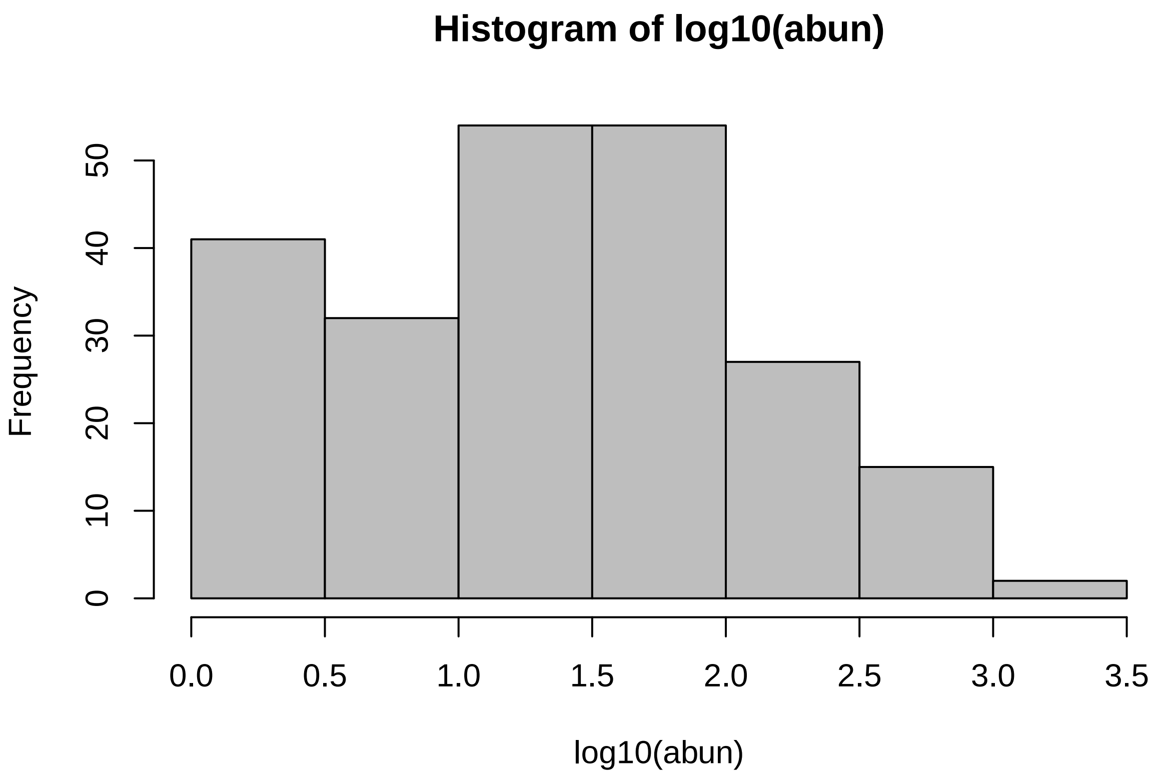
This has generated a huge literature based on attempts to describe this pattern mathematically and find underlying reasons for it. If we plot a histogram of abundances we can see the pattern.

```r
hist(abun,col="grey")
```



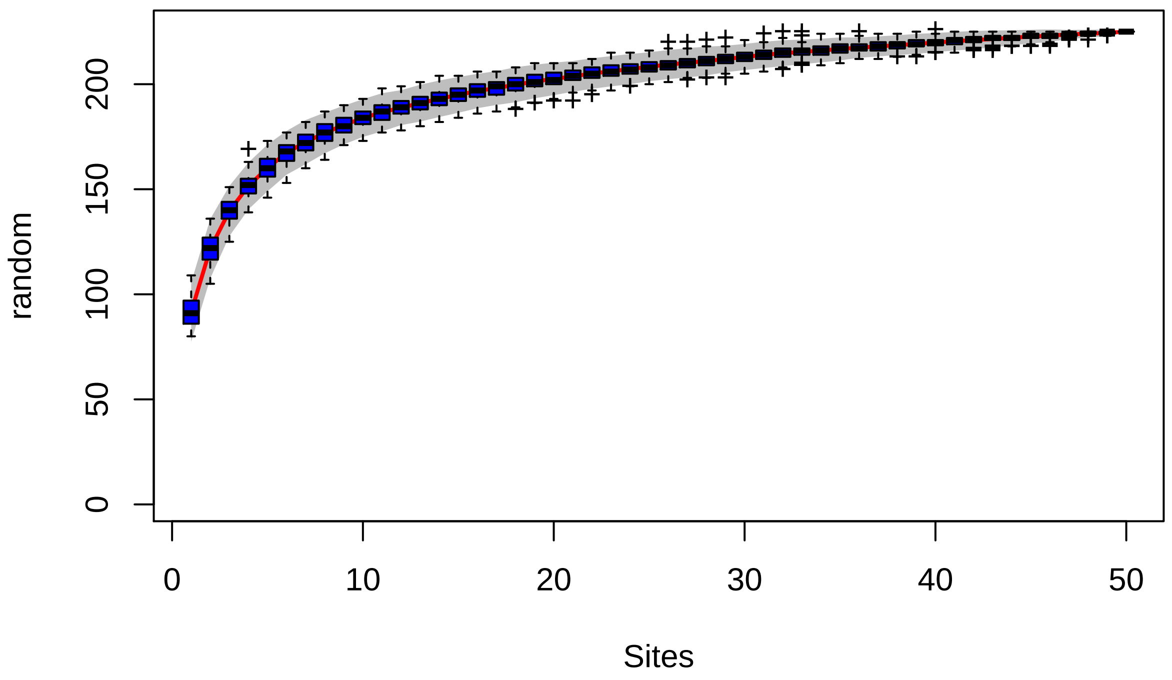
Taking logarithms

```r
hist(log10(abun),col="grey")
```



This may explain why there is a difference between the number of species found in each subplot and the total number of species. It is simply chance. As there are so many rare species you would not expect eachsubplot to contain all the species. We can look at the pattern ofaccumulation of species that we obtain if we aggregate plots randomly into larger units using the specaccum function.

```
AcCurve<-specaccum(BCI,"random")
plot(AcCurve,ci.type="poly", col="red", lwd=2, ci.lty=0, ci.col="grey")
boxplot(AcCurve, col="blue", pch="+",add=T)
```



```
specpool(BCI)
```

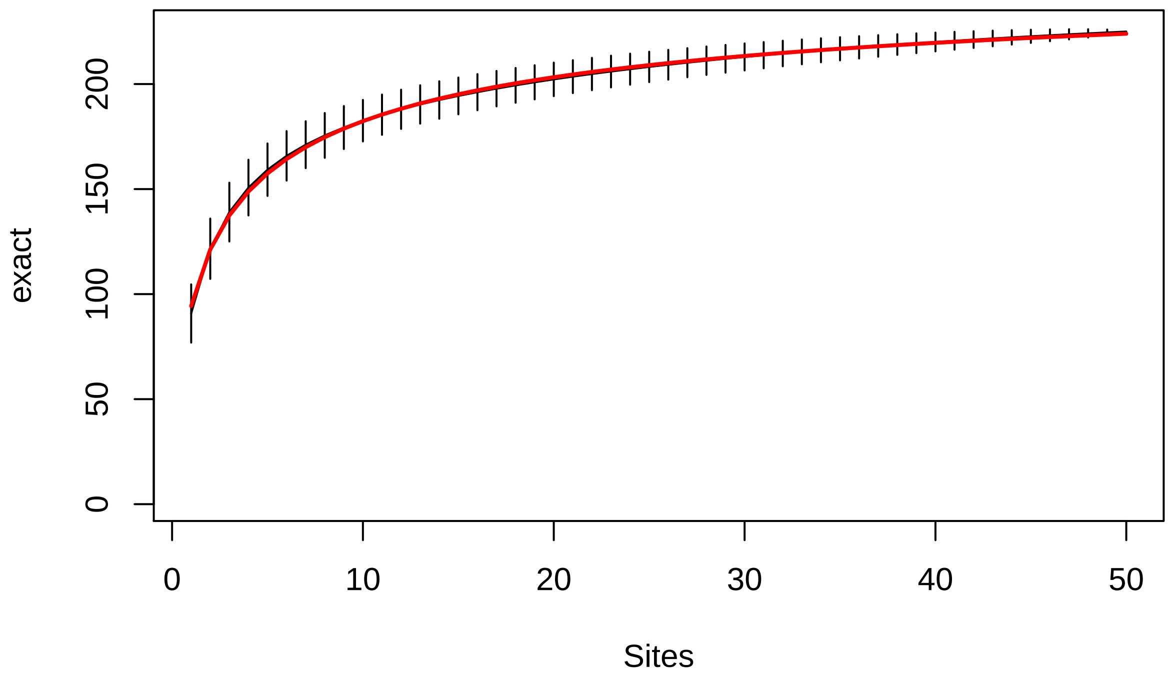
```
##     Species     chao  chao.se  jack1 jack1.se    jack2     boot  boot.se  n
## All     225 236.3732 6.54361 245.58 5.650522 247.8722 235.6862 3.468888 50
```

So species richness accumulates in a non linear manner. This has been modelled as a hyperbolic function in order to estimate the asymptotic(maximum) species richness. Fitting a Lomolino curve actually produces an estimate of the maximum number of species.

```
## Fit Lomolino model to the exact accumulation
AcCurve<-specaccum(BCI,"exact")
mod1 <- fitspecaccum(AcCurve, "lomolino")
coef(mod1)
```

```
##       Asym       xmid      slope
## 258.440682   2.442061   1.858694
```

```
plot(AcCurve)
## Add Lomolino model using argument 'add'
plot(mod1, add = TRUE, col=2, lwd=2)
```



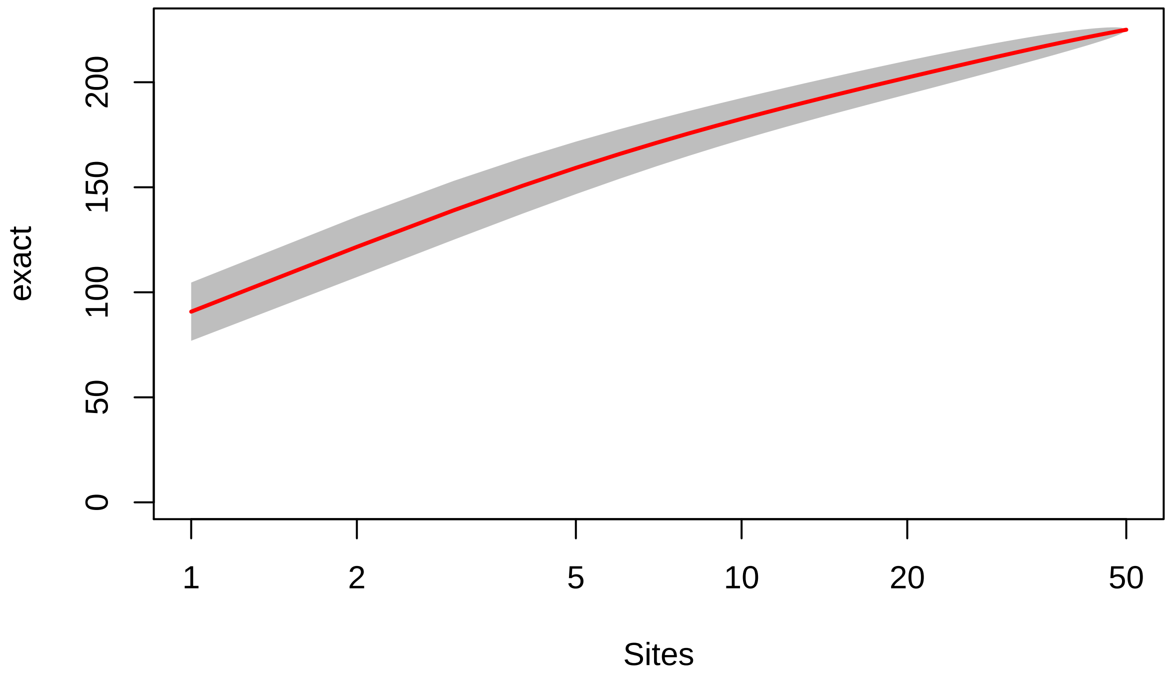
There are a range of “non parametric” estimators of species richness that claim to do the same thing.

```
pool<-apply(BCI,2,sum)
estimateR(pool)
```

```
##      S.obs    S.chao1   se.chao1      S.ACE     se.ACE
## 225.000000 237.214286   7.434824 238.217659   7.118588
```

However if we look at the accumulation on a log scale we may be less convinced that it is flattening off at all.

```
plot(AcCurve, ci.type="poly", col="red", lwd=2, ci.lty=0, ci.col="grey",log="x")
```



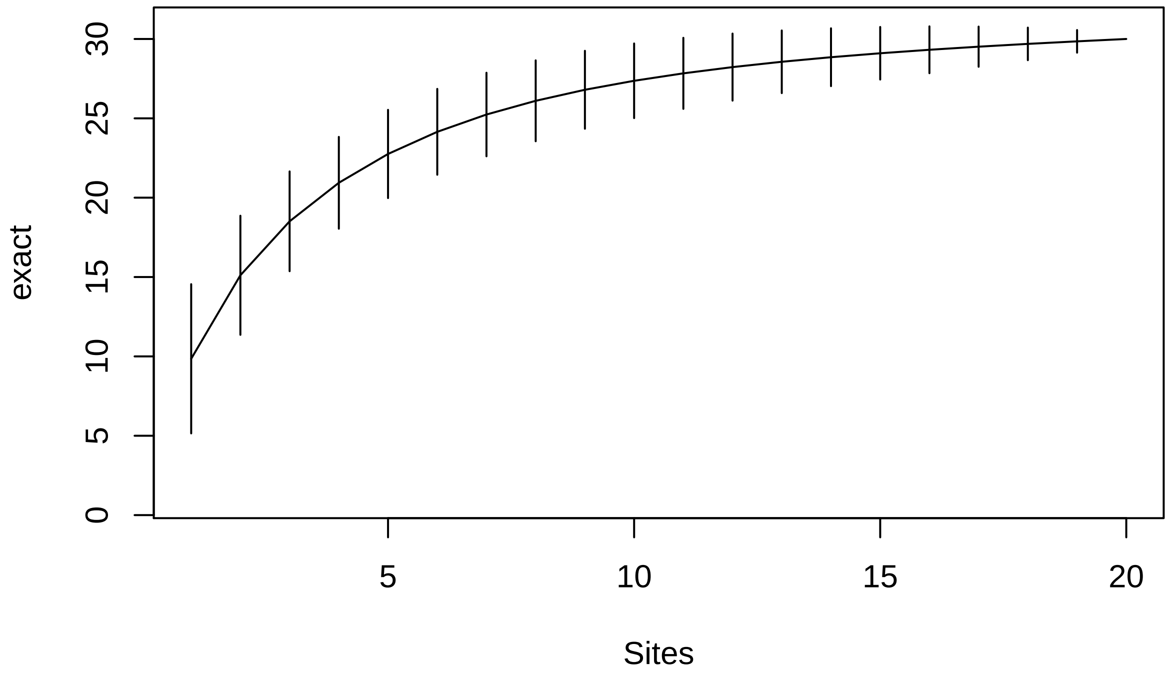
The practical implications are that when planning field work you may quickly observe at least half the species that you are going to find in any given area. However rare species will continue to add to the list, and there may be no real limit to this process.

Because the accumulation is more or less linear on a logarithmic scale there are clearly diminishing returns. If, say, six more species are added by doubling the number of sites, four times the number of sites will be needed to add six more, eight more for the next six and so on.

In this case you should notice that you expect to find around half the total number of species if you draw two sites at random from the data set and count the numbers of species. This may be useful when designing future studies.

The dune meadows data set found in the vegan package uses quadrat cover classes, so the numbers do not represent counts. The default settings will be OK for this sort of data.

```
data(dune)
AcCurve<-specaccum(dune)
plot(AcCurve)
```

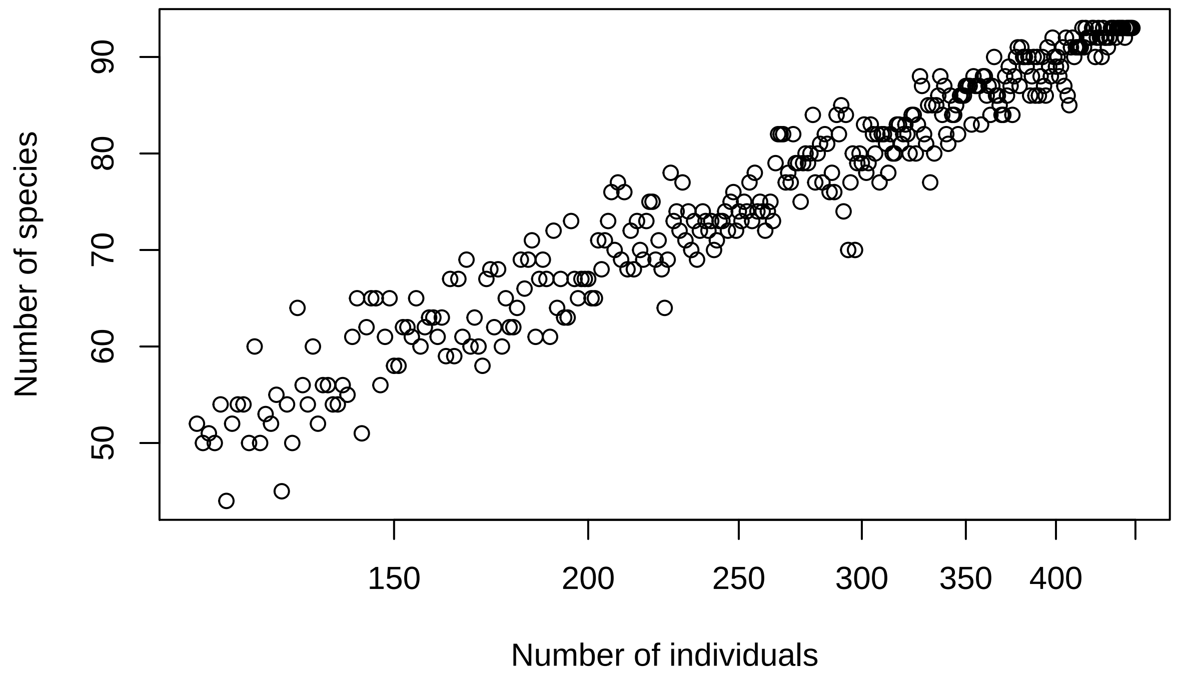


14.2 Resampling individuals

Some of the methods for resampling species accumulation curves (eg. Colemans, and rarefaction) are based on individuals. We can try an alternative form of resampling at the level of each plot.

```
resamp2 <-
function(X, rep=1,plot=FALSE) {
 require(vegan)
  Samp <- function(size, Indivs) {  length(table(sample(Indivs, size,
          replace=F)))  }
   N<-X[X>0]
   TotSp <- length(N)
   TotInds<-sum(N)#### Accumulation curve up to the maximum number of
   #individuals
   Sp <- 1:TotSp
   Inds <- rep(1:TotSp, N)
   Size <- rep(floor(TotInds*0.25):TotInds, rep) ###Note its(TotInds*0.25):TotInds
   sp.count <- sapply(Size, Samp, Inds)
   if(plot)plot(Size,sp.count,xlab="Number of individuals",ylab="Number of species",log="x")
   Sp.Ac1<-lm(sp.count~log(Size))
   Sp.Ac2<-lm(sp.count~log2(Size))
   fa<-fisher.alpha(N)
   res<-list(Logslope=Sp.Ac1$coefficients[2],Log2slope=Sp.Ac2$coefficients[2],Alpha=fa)
   return(res)
}
```

```
resamp2(BCI[1,],plot=TRUE)
```



```
## $Logslope
## log(Size)
##  30.50043
##
## $Log2slope
## log2(Size)
##   21.14129
##
## $Alpha
## [1] 35.67297
```

```
ans<-apply(BCI,1,resamp2)
res<-do.call("rbind",ans)
datatable(res) %>% formatRound(columns=c(1:3), digits=1)
```

Show 10 entries Search:

	Logslope	Log2slope	Alpha
1	30.6	21.2	35.7
2	27.8	19.3	31.0
3	30.9	21.4	33.3
4	30.2	21.0	33.9
5	34.6	24.0	38.0
6	30.4	21.1	32.5
7	26.5	18.4	30.6
8	28.4	19.7	33.4
9	32.6	22.6	35.7
10	31.8	22.0	34.8

Showing 1 to 10 of 50 entries Previous 1 2 3 4 5 Next

Once more, this pattern has some useful practical implications and some deeper theoretical ones. By looking at the slope after log transforming to base 2 we obtain an estimate of the number of new species which would be added each time we double the number of individuals counted.

14.2.1 Simpson's and Shannon's indices

Ok, we have seen that species abundance distributions tend to be very highly skewed, with a few common species and many more rare species. The counts of individuals in any one sampling unit will reflect this. Intuitively there is a difference between a sampling unit that contains 10 individuals of species A, 10 of species B and 10 of species C when it is compared to a sampling unit that has 28 of species A and only one individual of each of the other species.

There is a very long tradition in Ecology of measuring diversity using indices that combine measures of both species richness and equitability. There are two widely used indices of diversity, and around 30 that are less well known.

14.2.1.1 Simpson's index

The simplest diversity index to understand is Simpson's index.

Let's take the example of an equitable community with ten individuals of five species. The proportional abundance for each species is 0.2.

```
eq<-c(10,10,10,10,10)
prop<-eq/sum(eq)
prop
```

```
## [1] 0.2 0.2 0.2 0.2 0.2
```

Now imagine we draw an individual at random from the community. It could be of any one of the five species with an equal probability of 0.2. Let's say that it is species A. If we replace the individual and draw another, what is the probability that it will also be of species A? The answer must also be 0.2. So the probability of drawing two individuals of species A is 0.2 x 0.2 = 0.2^2 = 0.04.

The same applies to all the other species.

So the overall probability of drawing two individuals of the same species is given by the sum of p^2. We can call this D. If we subtract D from 1 we have the probability of drawing two individuals at random that are of different species. Alternatively we can find the reciprocal of D which represents the number of equally abundant species that would provide the probability obtained.

$D = \sum p^2$

$simp = 1 - D$

$invsimp = \frac{1}{D}$

```
sprop<-prop^2
sprop
```

```
## [1] 0.04 0.04 0.04 0.04 0.04
```

```
D<-sum(sprop)
1-D
```

```
## [1] 0.8
```

```
D
```

```
## [1] 0.2
```

Now what do we find for a much less equitable community?

```
uneq<-c(100,10,2,2,1)
prop<-uneq/sum(uneq)
prop
```

```
## [1] 0.869565217 0.086956522 0.017391304 0.017391304 0.008695652
```

In this case we are much more likely to draw an individual of the common species first. We are then very likely to draw another. So the probability of getting two individuals of the same species is much higher.

```
sprop<-prop^2
sprop
```

```
## [1] 7.561437e-01 7.561437e-03 3.024575e-04 3.024575e-04 7.561437e-05
```

```
D<-sum(sprop)
1-D
```

```
## [1] 0.2356144
```

```
1/D
```

```
## [1] 1.30824
```

The probability of obtaining two individuals of the same species is higher, as we are likely to draw two individuals of the very common species. Thus the "effective" number of species as measured by the inverse of Simpson's index is lower.

Simpson's index is influenced by both the number of species and the equitability of the distribution.

We can obtain a measure of pure equitability that takes a value between zero and 1 by dividing the inverse of simpsons's by the number of species.

```
Simp<-diversity(BCI,"simp")
Simp
```

```
##         1         2         3         4         5         6         7
## 0.9746293 0.9683393 0.9646078 0.9716117 0.9678267 0.9627557 0.9672014
##         8         9        10        11        12        13        14
## 0.9671998 0.9534257 0.9663808 0.9658398 0.9550599 0.9692075 0.9718626
##        15        16        17        18        19        20        21
## 0.9709057 0.9686598 0.9545126 0.9676685 0.9655820 0.9748589 0.9686058
##        22        23        24        25        26        27        28
## 0.9548316 0.9723529 0.9694268 0.9726152 0.9709567 0.9669962 0.9499296
##        29        30        31        32        33        34        35
## 0.9481041 0.9602659 0.9635807 0.9565267 0.9586946 0.9607876 0.7983976
##        36        37        38        39        40        41        42
## 0.9648567 0.9565015 0.9365144 0.9360204 0.9137131 0.9731442 0.9731849
##        43        44        45        46        47        48        49
## 0.9569632 0.9578733 0.9528853 0.9646728 0.9672083 0.9676412 0.9609552
##        50
## 0.9679784
```

```
InvSimp<-diversity(BCI,"invsimp")
InvSimp
```

```
##         1         2         3         4         5         6         7
## 39.415554 31.584877 28.254778 35.225771 31.081658 26.849731 30.489077
##         8         9        10        11        12        13        14
## 30.487609 21.471056 29.744868 29.273803 22.251827 32.475442 35.539830
##        15        16        17        18        19        20        21
## 34.371014 31.907937 21.984098 30.929617 29.054548 39.775448 31.853042
##        22        23        24        25        26        27        28
## 22.139382 36.170213 32.708387 36.516636 34.431303 30.299530 19.971863
##        29        30        31        32        33        34        35
## 19.269343 25.167317 27.457940 23.002620 24.209883 25.502106  4.960258
##        36        37        38        39        40        41        42
## 28.454909 22.989309 15.751596 15.629977 11.589250 37.235945 37.292428
##        43        44        45        46        47        48        49
## 23.235938 23.737903 21.224806 28.306797 30.495526 30.903463 25.611603
##        50
## 31.228916
```

```
Es<-InvSimp/specnumber(BCI)
Es
```

```
##          1          2          3          4          5          6
## 0.42382316 0.37601044 0.31394198 0.37474225 0.30773918 0.31587919
##          7          8          9         10         11         12
## 0.37181801 0.34645010 0.23856729 0.31643477 0.33648049 0.26490271
##         13         14         15         16         17         18
## 0.34919830 0.36265132 0.36958080 0.34309609 0.23638815 0.34752379
##         19         20         21         22         23         24
## 0.26655549 0.39775448 0.32174790 0.24328991 0.36535568 0.34429881
##         25         26         27         28         29         30
## 0.34777748 0.37836597 0.30605585 0.23496309 0.22406212 0.25945688
##         31         32         33         34         35         36
## 0.35659662 0.26139341 0.28151027 0.27719680 0.05976214 0.30929249
##         37         38         39         40         41         42
## 0.26124214 0.19209264 0.18607116 0.14486563 0.36505828 0.42864860
##         43         44         45         46         47         48
## 0.27018532 0.29306053 0.26203464 0.32914881 0.29897574 0.33959850
##         49         50
## 0.28144618 0.33579479
```

14.2.1.2 Shannon's index

Shannon's index (sometimes called the Shannon-Wienner index) is a commonly used diversity index that is rather similar in practice to Simpson's index. Unfortunately very few users of the index have an intuitive idea of it's meaning, as it is based on some rather obscure information theory.

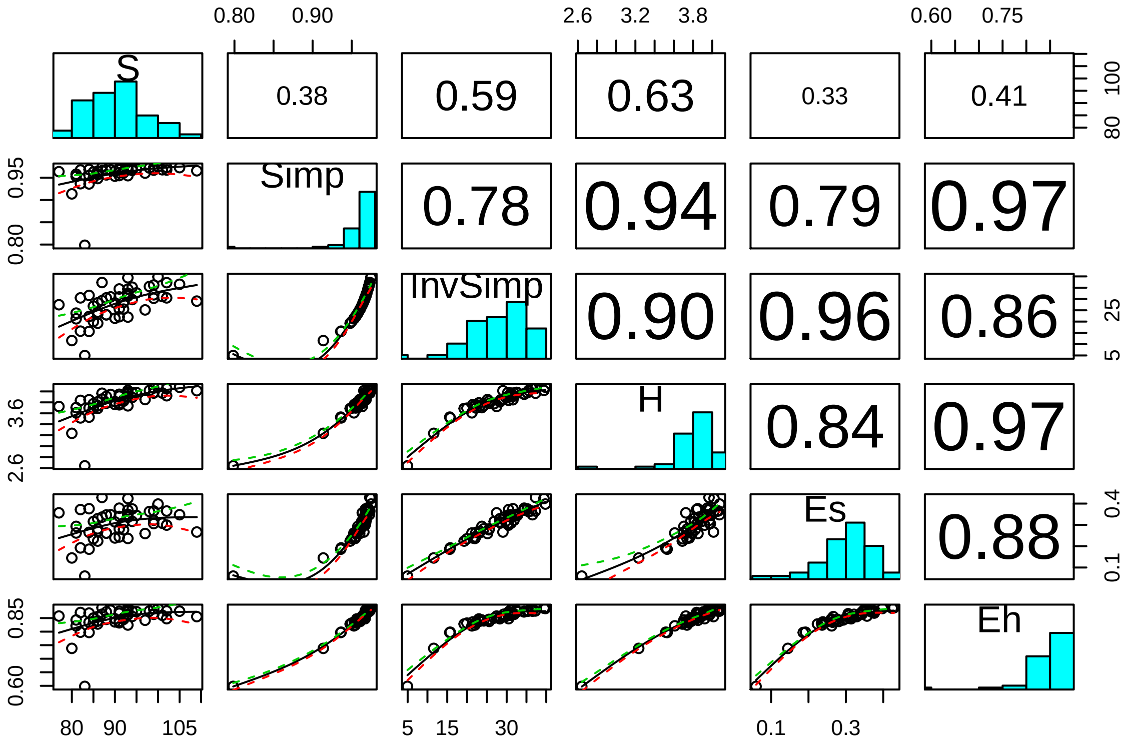
$H = -\sum p.log(p)$

The values that H can take tend to vary between 0.8 and 4. Values above 2.5 are high, but there is no simple rule. Fortunately Shannon's index can easily be converted into an index of equitability (or evenness) in a rather similar way to Simpson's by dividing it by the maximum value that it could take, which in this case is log(S). The index downweights rare species rather more than does Simpson's index.

The two indices basically measure the same thing. Simpson's is much simpler to understand and can be used by default. Shannon's is used by tradition as it was thought to be the better measure for many years. Shannon's is therefore always worth calculating and reporting for comparative purposes.

Simpson's and Shannon's indices are always highly correlated. An empirical correlation between species richness per se and evenness also tends to occur, but this is not automatic.

```
H<-diversity(BCI,"shannon")
Eh<-H/log(specnumber(BCI))
div<-data.frame(S,Simp,InvSimp,H,Es,Eh)
source("https://tinyurl.com/aqm-data/QEScript")
Xpairs(div)
```



The main reason that quantitative ecologists calculte diversity indices in most cases is to relate them to environmental variables. The two components of diversity are species richness and equitability (evenness). Either or both of these can take be modelled as functions of environmental variables. A significant reduction in evenness can, in appropriate circumstances, be taken as a symptom of environmental degradation. Low evenness scores can be the result of a community being dominated by a few species that are tolerant to, or favoured by, anthropogenic pollution or disturbance. However scores must be placed in context. The size of the site (quadrat or soil core) can have a great influence on diversity indices and this must always be taken into account.

14.2.2 Exercises

14.2.2.1 Mexican trees

The first exercise involves data taken from three forest types in Mexico. The abundances are counts of individual trees. What differences are there in species diversity between the forest types?

```r
mexveg<- read.csv("http://tinyurl.com/QEcol2013/mexveg.csv")
mexmat<-mexveg[,-c(1,2)]
```

Chapter 15

Crib sheets

The crib sheets contain R code for running analyses. There is no accompanying text to explain the output nor advice on why to use the method. You must consult course material in order to decide whether it is sensible to apply the method.

In order to use the cribsheets you **must** first become completely familiar with the process of loading data into R's memory by using either read.csv, for comma separated variable files or read_excel which can import data directly from an excel spreadsheet file. You need to know how to put together your own annotated markdown files, with embedded code chunks and annotated comments.

For each analysis an example data set is provided that is loaded from the /home/aqm/data folder on the server. The file is converted into a data table in the cribsheet. This data table can be exported and then used as the template for your own analysis.

To use the cribsheet, first look carefully at the format of the example data. Download this file and modify it in Excel, changing the values and headers to match your own data. Then build a markdown file using your own data as the input. Change any names of variables to match those used in your own data set. Providing you paste in chunks from the crib sheet **in the right order** you can then build your own bespoke analysis for your data that will reproduce the anaysis shown in the crobsheet. Order of the operations is very important, as some code chnuks are precursors to others. If you understand the logic of the analysis this will not be a problem.

15.1 Classical null hypothesis tests in R

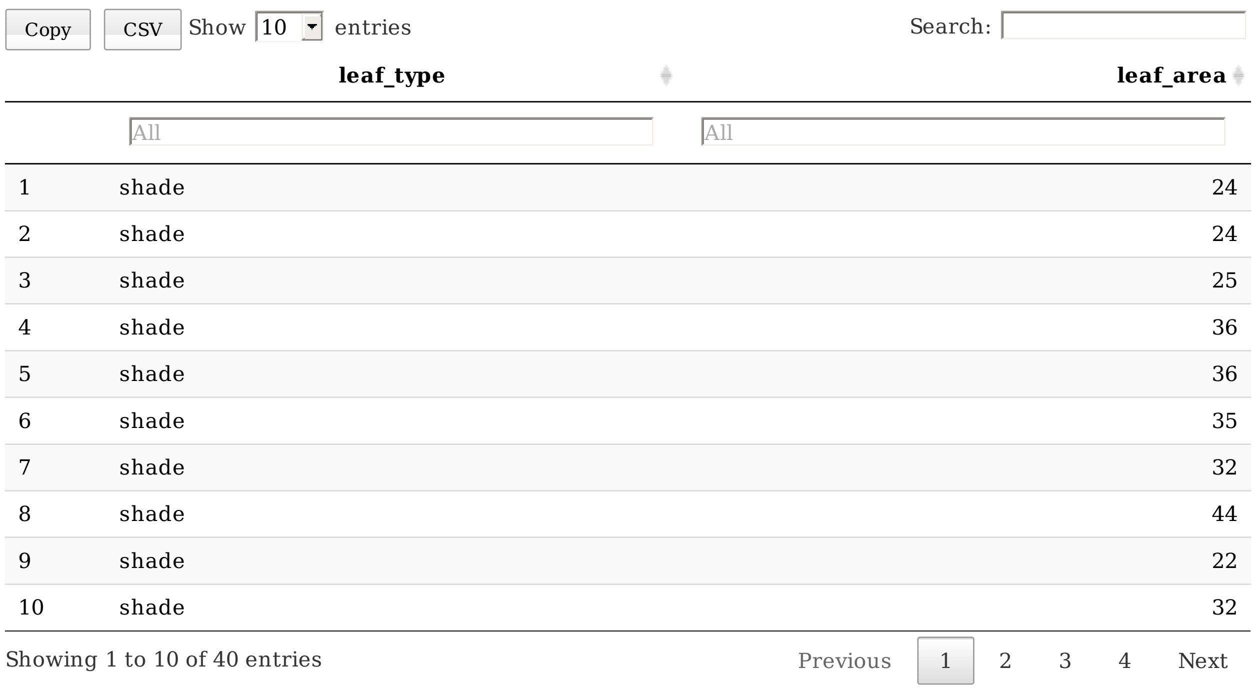
This crib sheet shows how to run simple, introductory, statistical tests. These are **very rarely** the best way to analyse your data. Model based procedures are available that produce a more informative analysis in every case, including situations when a non-parametric test is often chosen.

15.1.1 Un-paired T-test

15.1.2 Data formats

15.1.2.1 Long format

```r
d<-read.csv("/home/aqm/data/leaves.csv")
dt(d)
```


Copy CSV Show 10 entries Search:

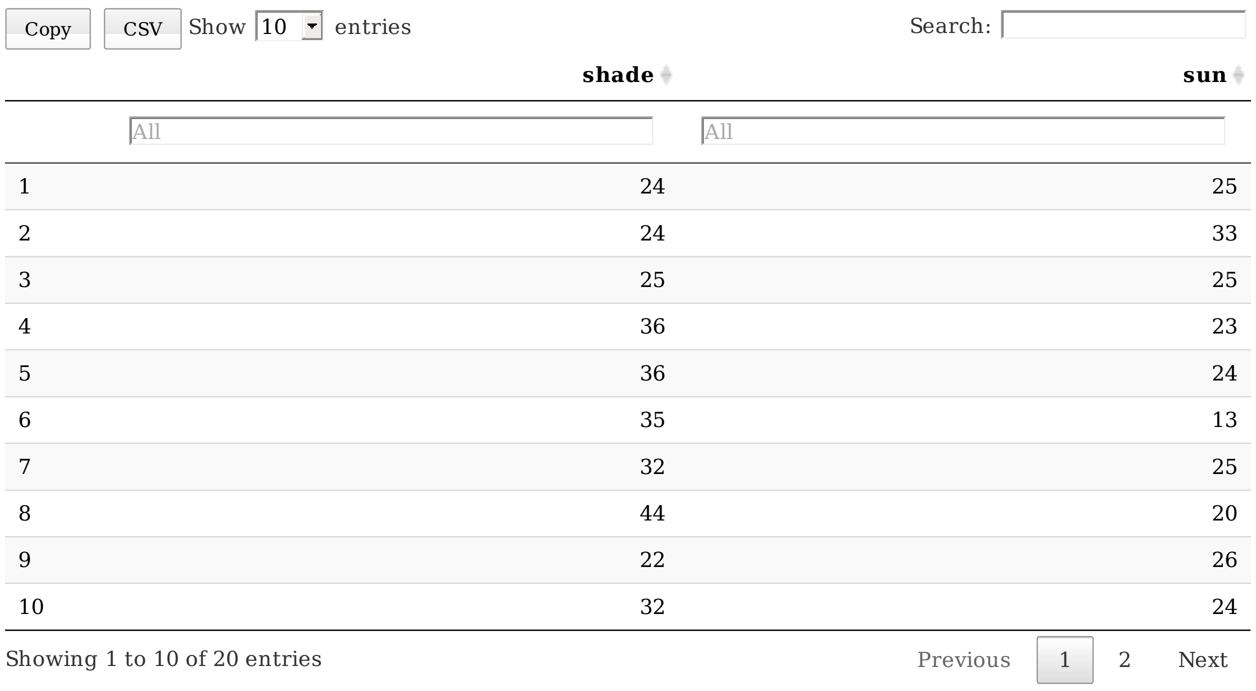
	leaf_type	leaf_area
	All	All
1	shade	24
2	shade	24
3	shade	25
4	shade	36
5	shade	36
6	shade	35
7	shade	32
8	shade	44
9	shade	22
10	shade	32

Showing 1 to 10 of 40 entries Previous 1 2 3 4 Next

15.1.2.2 Wide format

You may sometimes be given data in a wide format, with one column per group. This is not a standard data frame. The most consistent approach is to turn it into a data frame

```
d2<-read.csv("/home/aqm/data/leaves2.csv")
dt(d2)
```

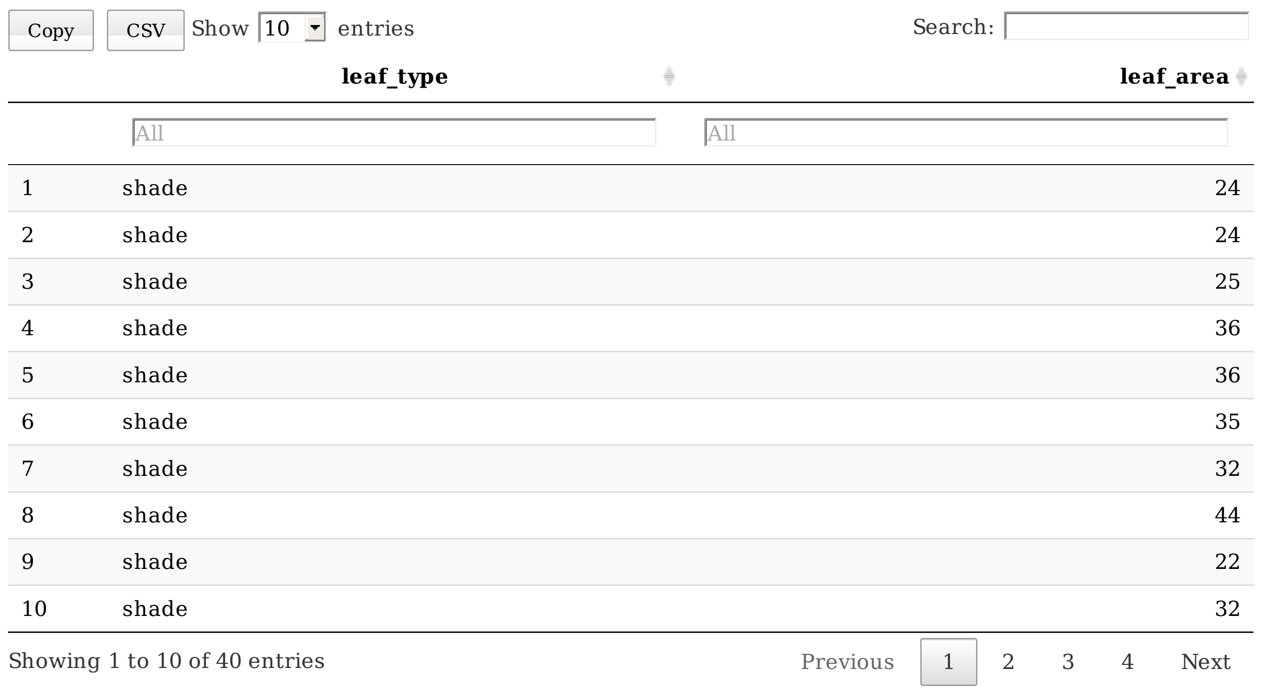

Copy CSV Show 10 entries Search:

	shade	sun
	All	All
1	24	25
2	24	33
3	25	25
4	36	23
5	36	24
6	35	13
7	32	25
8	44	20
9	22	26
10	32	24

Showing 1 to 10 of 20 entries Previous 1 2 Next

```
## To long format
d2<-gather(d2,key=leaf_type,value=leaf_area)
```

```r
dt(d2)
```



	leaf_type	leaf_area
1	shade	24
2	shade	24
3	shade	25
4	shade	36
5	shade	36
6	shade	35
7	shade	32
8	shade	44
9	shade	22
10	shade	32

Showing 1 to 10 of 40 entries

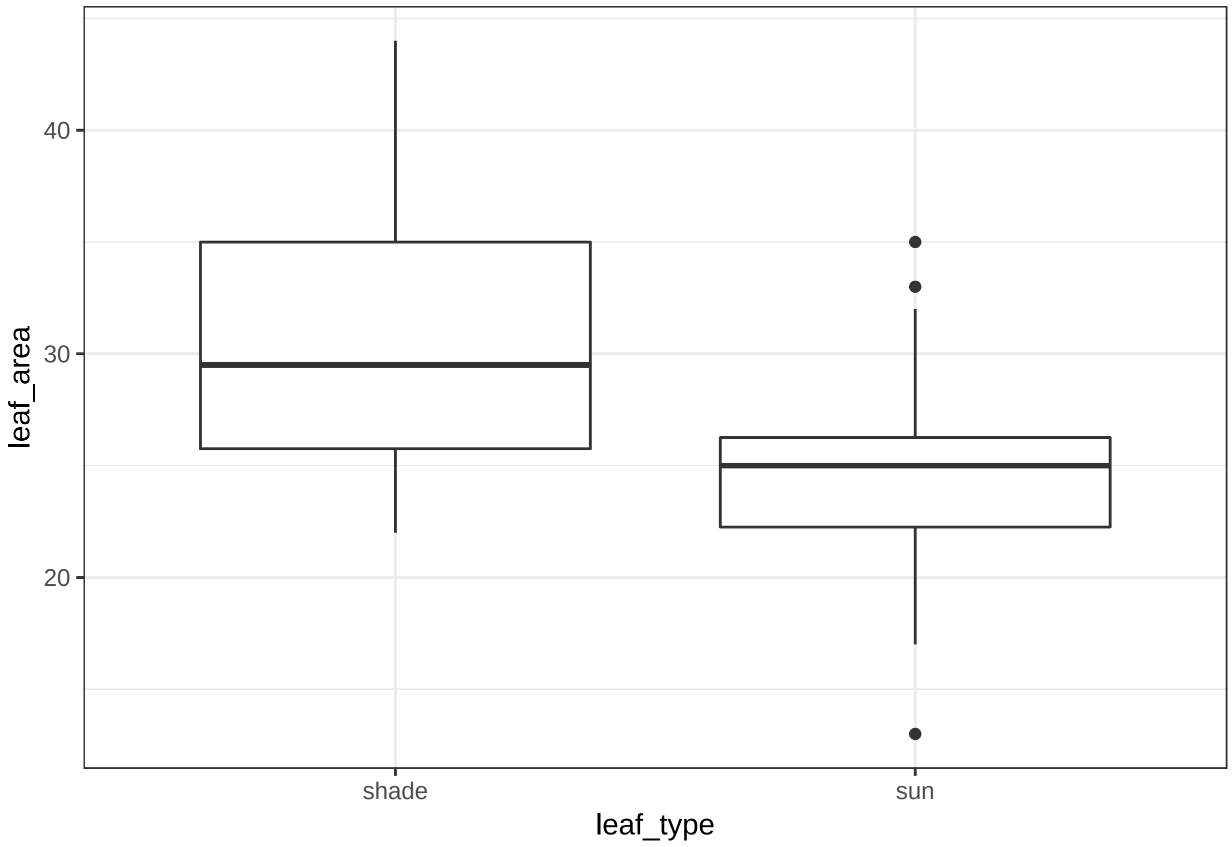
```r
d2$id<-rep(1:20,times=2)
d2 %>% spread(key=leaf_type,value=leaf_area)
```

```
##    id shade sun
## 1   1    24  25
## 2   2    24  33
## 3   3    25  25
## 4   4    36  23
## 5   5    36  24
## 6   6    35  13
## 7   7    32  25
## 8   8    44  20
## 9   9    22  26
## 10 10    32  24
## 11 11    26  17
## 12 12    28  20
## 13 13    30  19
## 14 14    26  24
## 15 15    23  35
## 16 16    29  25
## 17 17    39  27
## 18 18    35  25
## 19 19    29  32
## 20 20    35  27
```

15.1.3 Boxplot

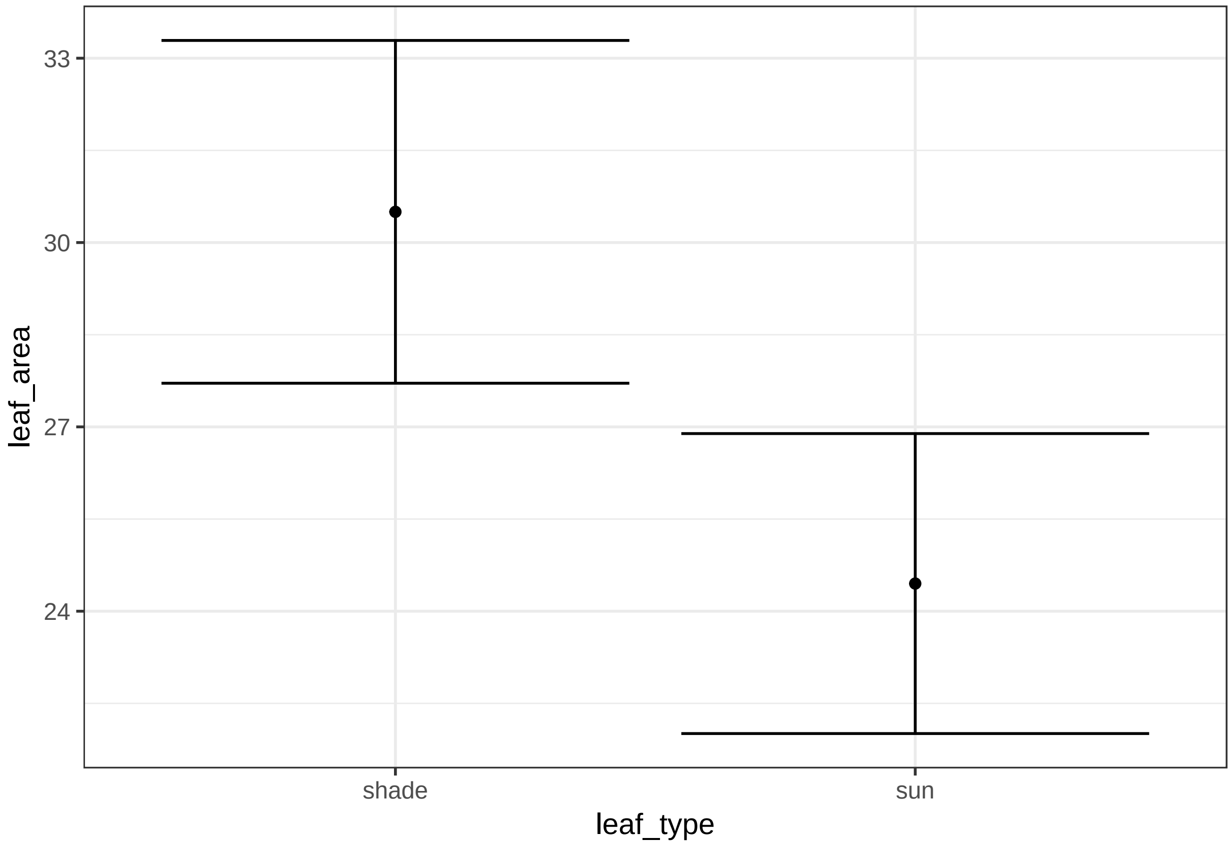
```r
g0<-ggplot(d,aes(x=leaf_type,y=leaf_area))
g_box<- g0 +geom_boxplot()
```

```
g_box
```



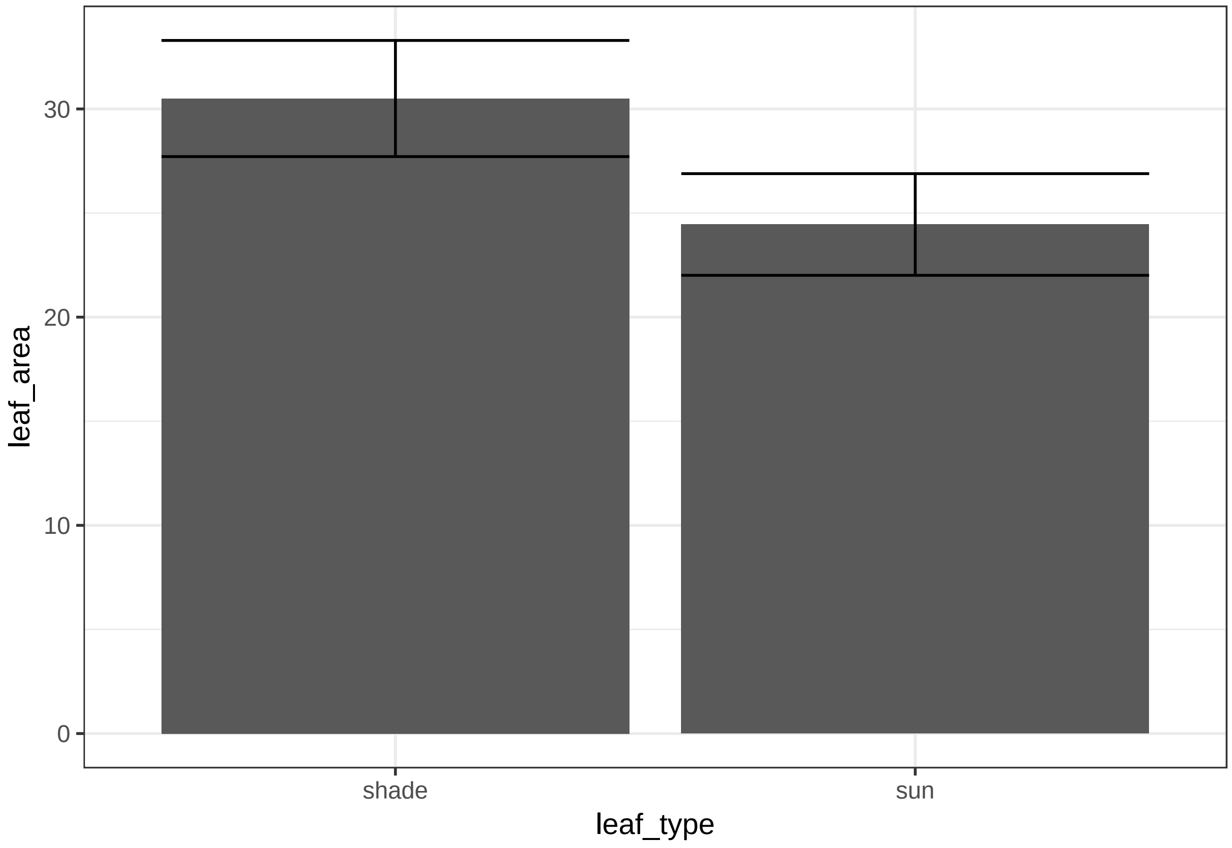
15.1.4 Confidence interval plot

```
g0<-ggplot(d,aes(x=leaf_type,y=leaf_area))
g_conf <- g0 + stat_summary(fun.y=mean,geom="point") + stat_summary(fun.data=mean_cl_normal,geom="errorb
g_conf
```



15.1.5 Dynamite plot

```
g0<-ggplot(d,aes(x=leaf_type,y=leaf_area))
g_bar <- g0 + stat_summary(fun.y=mean,geom="bar") + stat_summary(fun.data=mean_cl_normal,geom="errorbar
g_bar
```



15.1.6 T-test

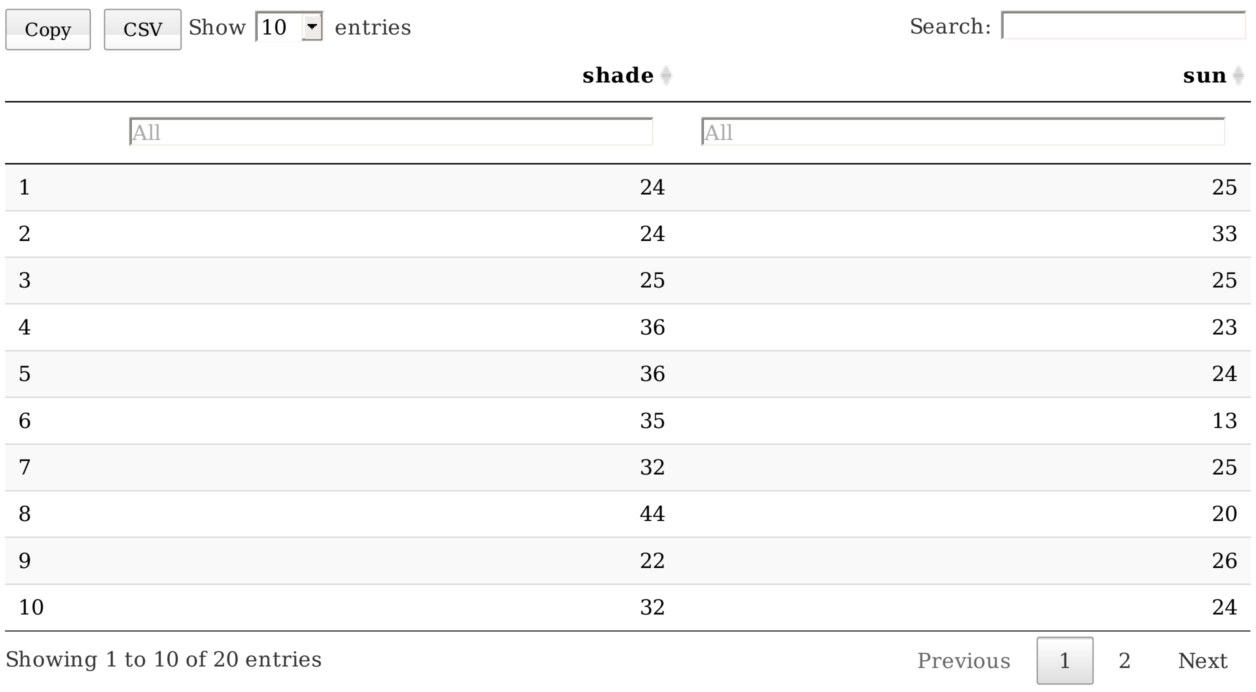
```
t.test(d$leaf_area~d$leaf_type)
```

```
##
##  Welch Two Sample t-test
##
## data:  d$leaf_area by d$leaf_type
## t = 3.416, df = 37.343, p-value = 0.001546
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.462573 9.637427
## sample estimates:
## mean in group shade   mean in group sun
##               30.50               24.45
```

15.2 Wilcoxon test (also known as 'Mann-Whitney' test)

15.2.1 Wide format

```
d2<-read.csv("/home/aqm/data/leaves2.csv")
dt(d2)
```



	shade	sun
	All	All
1	24	25
2	24	33
3	25	25
4	36	23
5	36	24
6	35	13
7	32	25
8	44	20
9	22	26
10	32	24

15.2.2 Wilcoxon test

```
wilcox.test(d2$shade,d2$sun)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  d2$shade and d2$sun
## W = 307.5, p-value = 0.003672
## alternative hypothesis: true location shift is not equal to 0
```

15.3 Paired T-test

15.3.1 Data formats

15.3.1.1 Wide format

The wide format seems the natural one to use in this case.

```
d2<-read.csv("/home/aqm/data/paired2.csv")
dt(d2)
```

Copy CSV Show 10 entries Search:

	id	After	Before
	All	All	All
1	1	7	9
2	2	16	11
3	3	13	9
4	4	29	18
5	5	22	21
6	6	23	17
7	7	31	10
8	8	19	15
9	9	26	15
10	10	17	22

Showing 1 to 10 of 10 entries Previous 1 Next

15.3.1.2 Long format

For the classic paired t-test function the long format is best changed to wide.

```r
d<-read.csv("/home/aqm/data/paired1.csv")
dt(d)
```

Copy CSV Show 10 entries Search:

	id	time	val
	All	All	All
1	1	Before	9
2	1	After	7
3	2	Before	11
4	2	After	16
5	3	Before	9
6	3	After	13
7	4	Before	18
8	4	After	29
9	5	Before	21
10	5	After	22

Showing 1 to 10 of 20 entries Previous 1 2 Next

```r
## Change to wide
d %>% spread(time,val) %>% dt()
```

Copy CSV Show 10 entries Search:

	id	After	Before
	All	All	All
1	1	7	9
2	2	16	11
3	3	13	9
4	4	29	18
5	5	22	21
6	6	23	17
7	7	31	10
8	8	19	15
9	9	26	15
10	10	17	22

Showing 1 to 10 of 10 entries Previous 1 Next

15.3.2 T-test

```
t.test(d2$After,d2$Before,paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  d2$After and d2$Before
## t = 2.3941, df = 9, p-value = 0.04028
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   0.3087225 10.8912775
## sample estimates:
## mean of the differences
##                     5.6
```

```
d %>% spread(time,val) -> d2
t.test(d2$After,d2$Before,paired=TRUE)
```

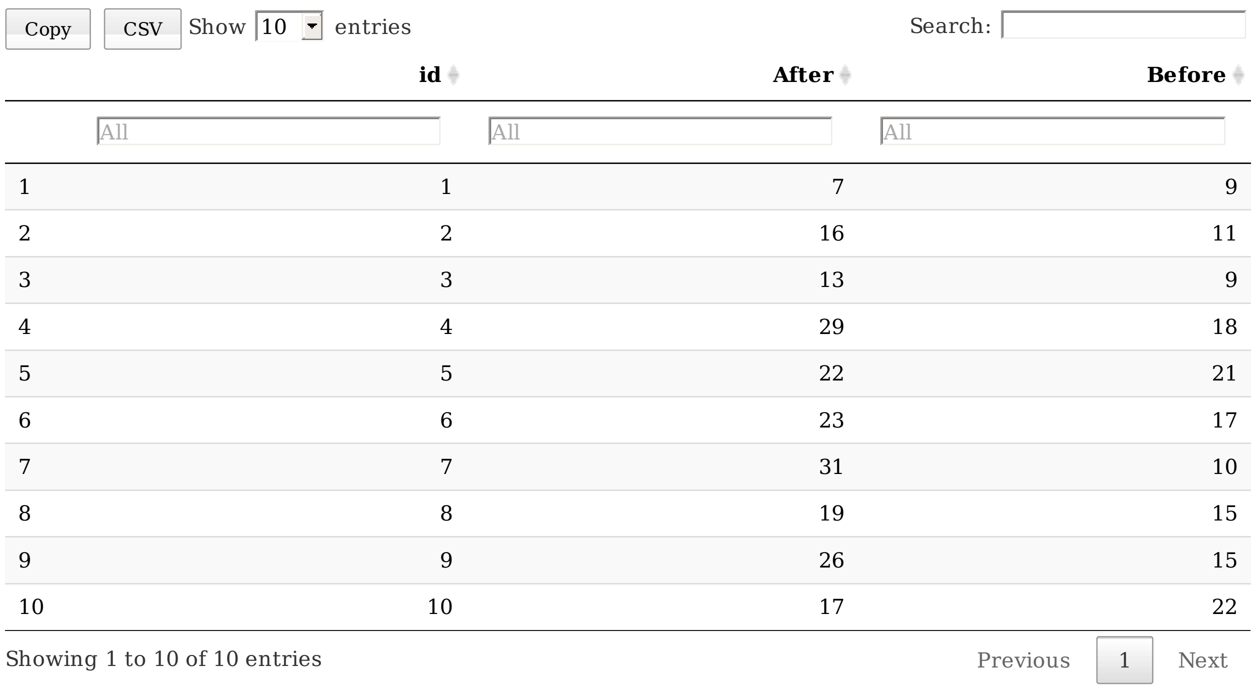
```
##
##  Paired t-test
##
## data:  d2$After and d2$Before
## t = 2.3941, df = 9, p-value = 0.04028
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   0.3087225 10.8912775
## sample estimates:
## mean of the differences
##                     5.6
```

15.4 Paired Wilcoxon test

15.4.1 Wide format

The wide format seems the natural one to use in this case.

```
d2<-read.csv("/home/aqm/data/paired2.csv")
dt(d2)
```



	id	After	Before
1	1	7	9
2	2	16	11
3	3	13	9
4	4	29	18
5	5	22	21
6	6	23	17
7	7	31	10
8	8	19	15
9	9	26	15
10	10	17	22

Showing 1 to 10 of 10 entries

15.4.2 Wilcoxon test

```
wilcox.test(d2$Before,d2$After, paired=)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  d2$Before and d2$After
## W = 26, p-value = 0.07522
## alternative hypothesis: true location shift is not equal to 0
```

15.5 Correlation test

15.5.1 Data format

Only the standard data frame format is sensible in this case. However a data frame may consist of many variables that can be correlated with each other. In this case more advanced methods avoid the need to correlate each pair in turn. Fitting regresion lines is included in other crib sheets.

```
d<-read.csv("/home/aqm/data/arne_pines_simple.csv")
dt(d)
```

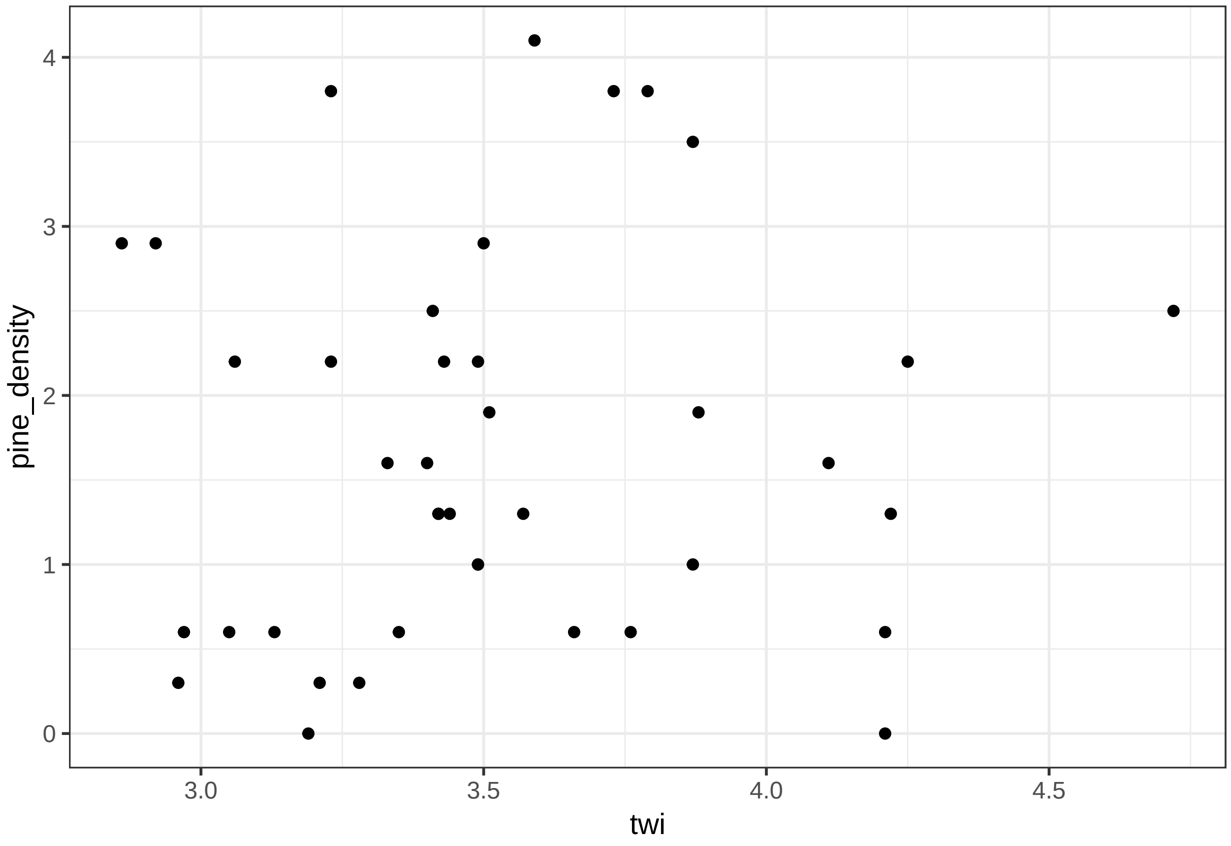
Copy CSV Show 10 entries Search:

	pine_density	twi
	All	All
1	0.6	3.05
2	1.3	3.57
3	2.2	3.43
4	3.8	3.23
5	0.6	4.21
6	2.9	3.5
7	3.8	3.79
8	4.1	3.59
9	2.2	4.25
10	2.5	4.72

Showing 1 to 10 of 40 entries Previous 1 2 3 4 Next

15.5.2 Scatterplot

```
g0<-ggplot(d,aes(x=twi,y=pine_density)) +geom_point()
g0
```



15.5.3 Pearson's correlation test

```
cor.test(d$pine_density,d$twi)
```

```
##
##  Pearson's product-moment correlation
##
## data:  d$pine_density and d$twi
## t = 0.53448, df = 38, p-value = 0.5961
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2313542  0.3874638
## sample estimates:
##        cor
## 0.08638047
```

15.5.4 Spearman's rank correlation test

There are often ties, but this does not *invalidate* the test. R produces a warning in this case.

```
cor.test(d$pine_density,d$twi,method="spearman")
```

```
##
##  Spearman's rank correlation rho
##
```

```
## data:  d$pine_density and d$twi
## S = 8988.5, p-value = 0.3339
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1567992
```

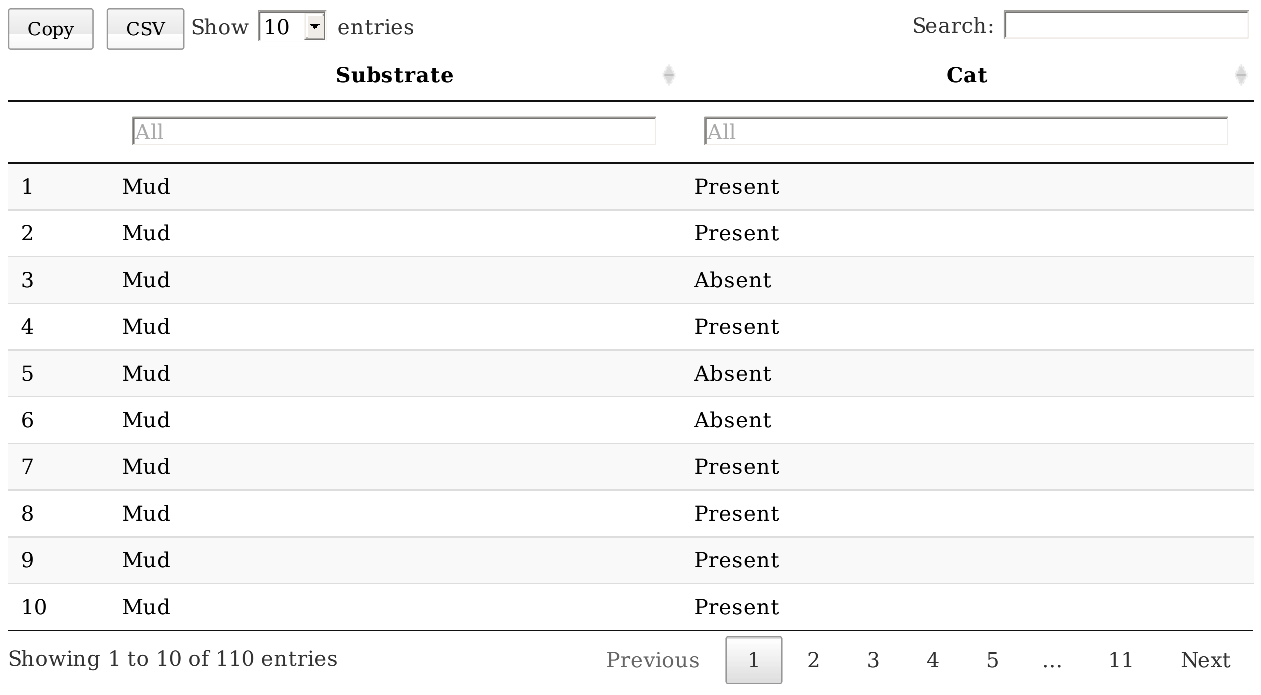
15.6 Chi squared contingency tables

15.6.1 Data formats

15.6.1.1 Long format

The data will originally have been collected though classifying each observation. So, if the data consists of mud cores that have been classified into two categories of substrate, mud or sand, and two categories depending whether ragworm are present or absent you will produces a csv file with the format as shown.

```
d<-read.csv("/home/aqm/data/HedisteCat.csv")
dt(d)
```



Copy
CSV
Show 10 entries
Search:

	Substrate	Cat
	All	All
1	Mud	Present
2	Mud	Present
3	Mud	Absent
4	Mud	Present
5	Mud	Absent
6	Mud	Absent
7	Mud	Present
8	Mud	Present
9	Mud	Present
10	Mud	Present

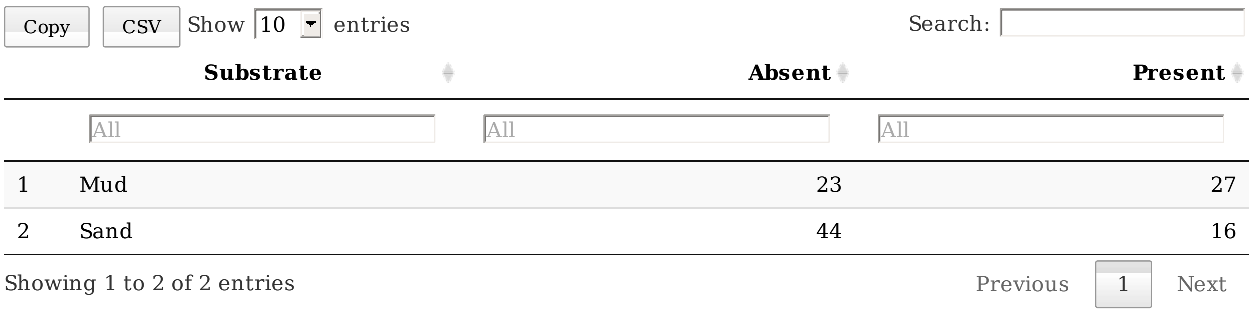
Showing 1 to 10 of 110 entries

Previous 1 2 3 4 5 … 11 Next

15.6.1.2 Tablular format

You might already have tabulated the data in Excel. Providing that the table is in the top cells of the first sheet of an Excel spreadsheet, this code will load the data.

```
library(readxl)
system ("cp contingency_table.xlsx /home/aqm/data/contingency_table.xlsx")
ct <-read_excel("/home/aqm/data/contingency_table.xlsx")
dt(ct)
```



	Substrate	Absent	Present
	All	All	All
1	Mud	23	27
2	Sand	44	16

Showing 1 to 2 of 2 entries

15.6.2 Table of counts

15.6.2.1 Table of counts using the data frame format

```
ct<-table(d)
ct
```

```
##          Cat
## Substrate Absent Present
##      Mud      23      27
##      Sand     44      16
```

```
### Formatted version for HTMl printing
ct %>% kable() %>% kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Absent

Present

Mud

23

27

Sand

44

16

15.6.2.2 Table of counts using the ct format

```
## These data are already in tabular format, but some slight rearrangement is needed to turn them into
ct <-read_excel("contingency_table.xlsx")
ct<-as.data.frame(ct)
row.names(ct) <- ct[,1]
ct<-ct[,-1]
ct<-as.table(as.matrix(ct))
ct
```

```
##      Absent Present
## Mud      23      27
## Sand     44      16
```

```
### Formatted version for HTMl printing
ct %>% kable() %>% kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Absent

Present

Mud

23

27

Sand

44

16

15.6.3 Table of Proportions

15.6.3.1 Table of proportions

```
pt<-round(prop.table(ct) *100,1)
pt
```

```
##       Absent Present
## Mud     20.9    24.5
## Sand    40.0    14.5
```

```
### Formatted version for HTMl printing
pt %>% kable() %>% kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Absent

Present

Mud

20.9

24.5

Sand

40.0

14.5

15.6.3.2 Table of proportions for rows

```
ptr<-round(prop.table(ct,margin=1) *100,1)
ptr
```

```
##       Absent Present
## Mud     46.0    54.0
## Sand    73.3    26.7
```

```
ptr %>% kable() %>% kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Absent

Present

Mud

46.0

54.0

Sand

73.3

26.7

15.6.3.3 Table of proportions for columns

```
ptc<-round(prop.table(table(d),margin=2) *100,1)
ptc
```

```
##          Cat
## Substrate Absent Present
##       Mud    34.3    62.8
##       Sand   65.7    37.2
```

```
ptc %>% kable() %>% kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Absent

Present

Mud

34.3

62.8

Sand

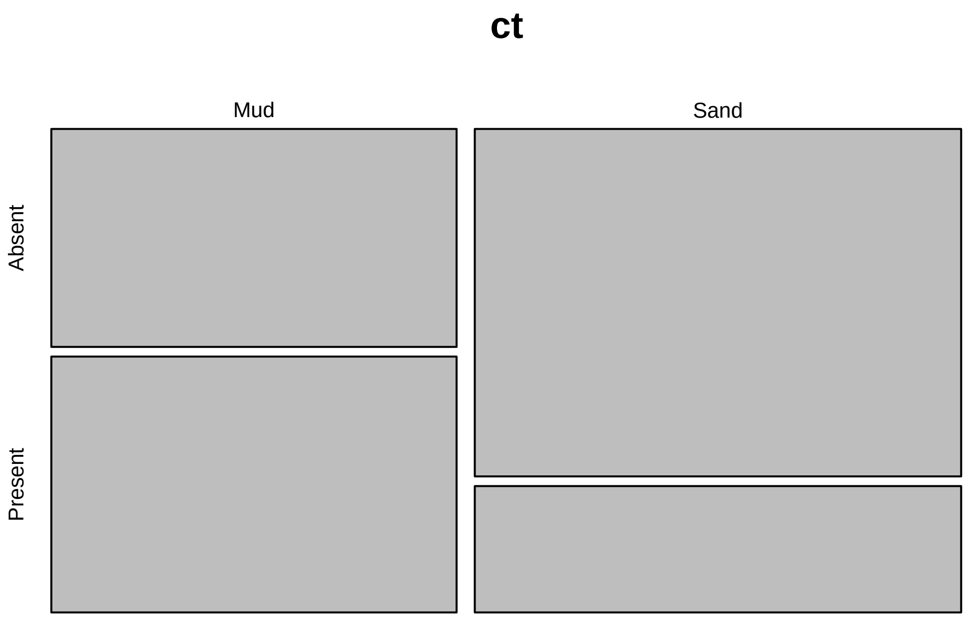
65.7

37.2

15.6.4 Plots

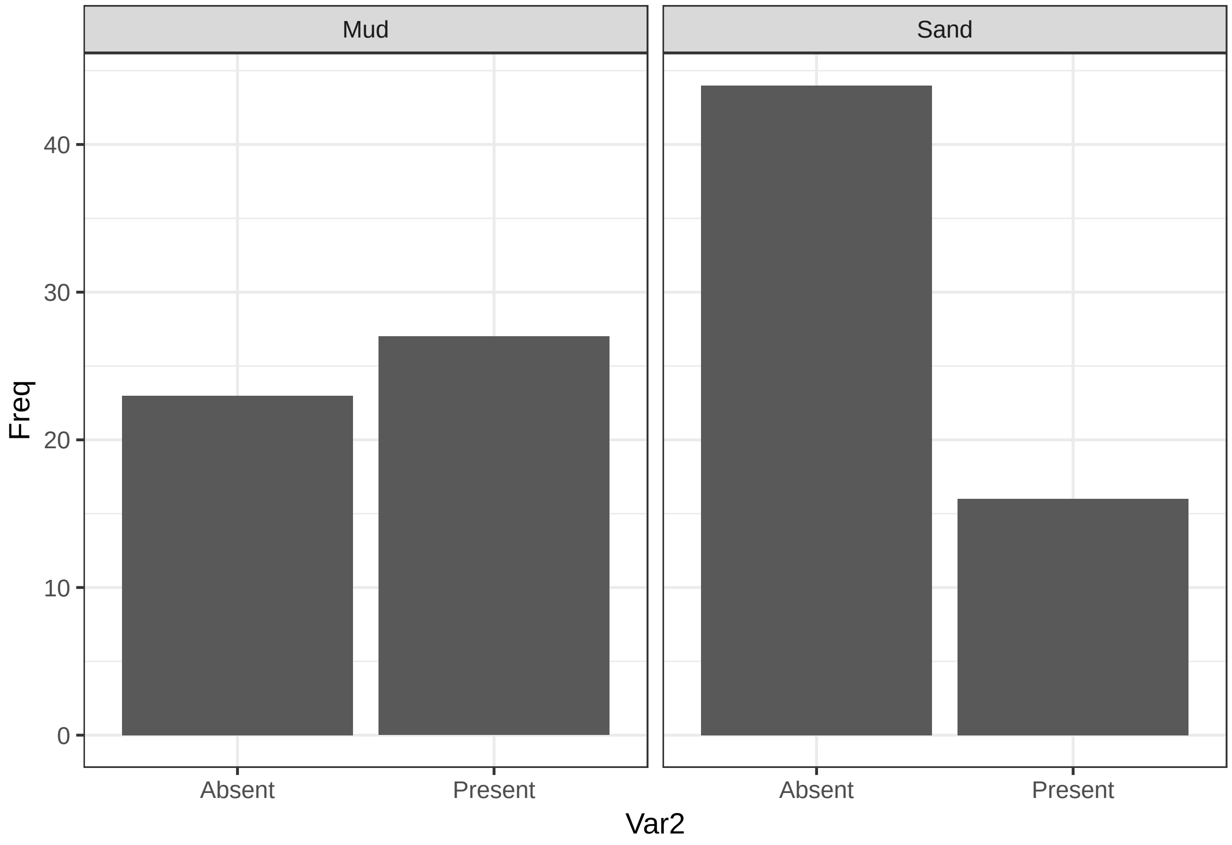
15.6.4.1 Mosaic plot

```
mosaicplot(ct)
```



15.6.4.2 Barplot

```
ct_d<-data.frame(ct)

bc<-ggplot(ct_d,aes(x=Var2,y=Freq))+geom_bar(stat="identity")
bc + facet_wrap(~Var1)
```



15.6.5 Chisq-test

```r
chisq.test(ct)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  ct
## X-squared = 7.4482, df = 1, p-value = 0.00635
```

15.6.6 Fisher's exact test

```r
fisher.test(ct)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  ct
## p-value = 0.005719
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.1287875 0.7387656
## sample estimates:
## odds ratio
##  0.3132911
```

Chapter 16

Regression and ANOVS crib sheet

This chapter provides all the code chunks that may be useful in the context of the data analysis component of the assignment. The data set used to illustrate is the mussels data, that can be analysed using one way ANOVA and regression in the context of calibrating a relationship.

You should look through ALL the handouts provided on these techniques to understand the underlying theory. This "crib sheet" simply provides access to useful code.

1. **ALWAYS CHECK THAT THE STEPS HAVE BEEN TAKEN IN THE RIGHT ORDER.**
2. **LOOK AT THE DATA YOU HAVE LOADED FIRST**
3. **USE YOUR OWN VARIABLE NAMES**
4. **PASTE IN CODE CHUNKS CAREFULLY; LEAVING GAPS BETWEEN EACH CHUNK**
5. **COMMENT ON ALL THE STEPS**

16.1 Packages needed

Include this chunk at the top of you analysis to ensure that you have all the packages. It also includes the wrapper to add buttons to a data table if you want to use this. Remember that data tables can only be included in HTML documents.

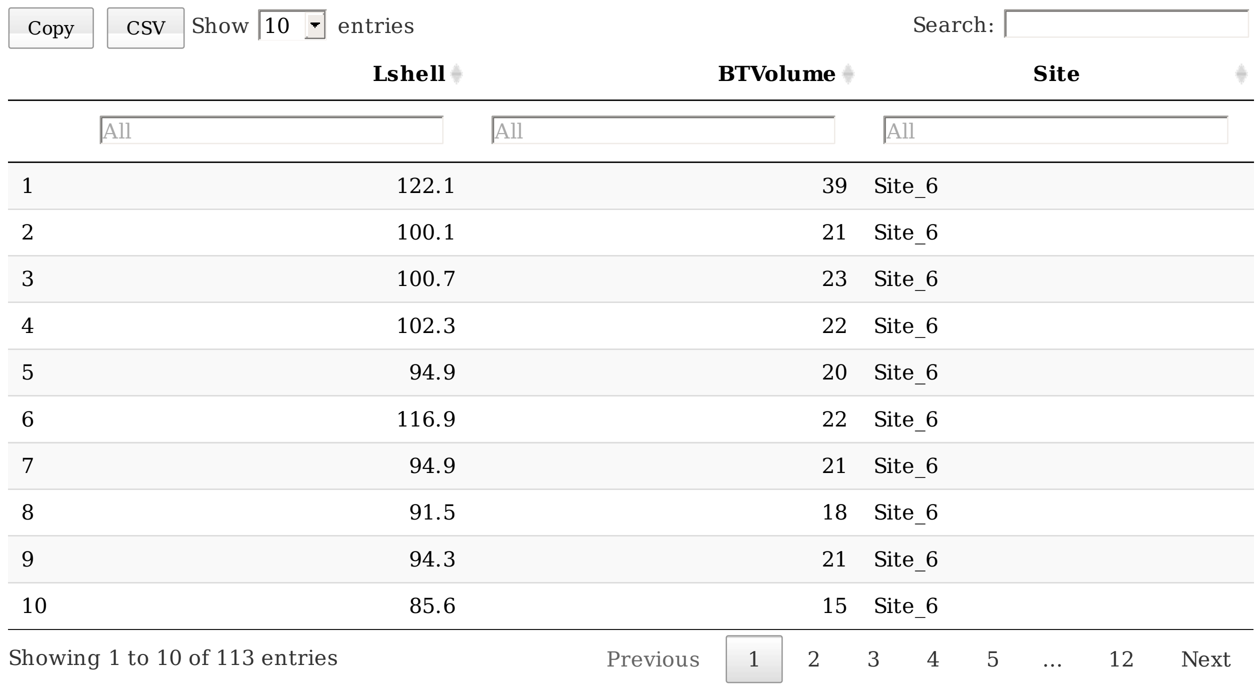
```r
library(ggplot2)
library(dplyr)
library(mgcv)
library(DT)
theme_set(theme_bw())
dt<-function(x) DT::datatable(x,
    filter = "top",
    extensions = c('Buttons'), options = list(
    dom = 'Blfrtip',
    buttons = c('copy', 'csv', 'excel'), colReorder = TRUE
  ))
```

Chapter 17

Univariate

17.1 Data

```r
d<-read.csv("https://tinyurl.com/aqm-data/mussels.csv")
dt(d)
```



	Lshell	BTVolume	Site
1	122.1	39	Site_6
2	100.1	21	Site_6
3	100.7	23	Site_6
4	102.3	22	Site_6
5	94.9	20	Site_6
6	116.9	22	Site_6
7	94.9	21	Site_6
8	91.5	18	Site_6
9	94.3	21	Site_6
10	85.6	15	Site_6

Showing 1 to 10 of 113 entries

17.2 Data summaries for individual variables

Change the name of the variable to match a numerical variable in your own data set. The command removes NAs just in case you have them

```r
summary(d$Lshell,na.rm=TRUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    61.9    97.0   106.9   106.8   118.7   132.6
```

17.3 Individual statistics for a single variable

Mean, median, standard deviation and variance.

```
mean(d$Lshell, na.rm=TRUE)
```

```
## [1] 106.835
```

```
median(d$Lshell, na.rm=TRUE)
```

```
## [1] 106.9
```

```
sd(d$Lshell, na.rm=TRUE)
```

```
## [1] 14.84384
```

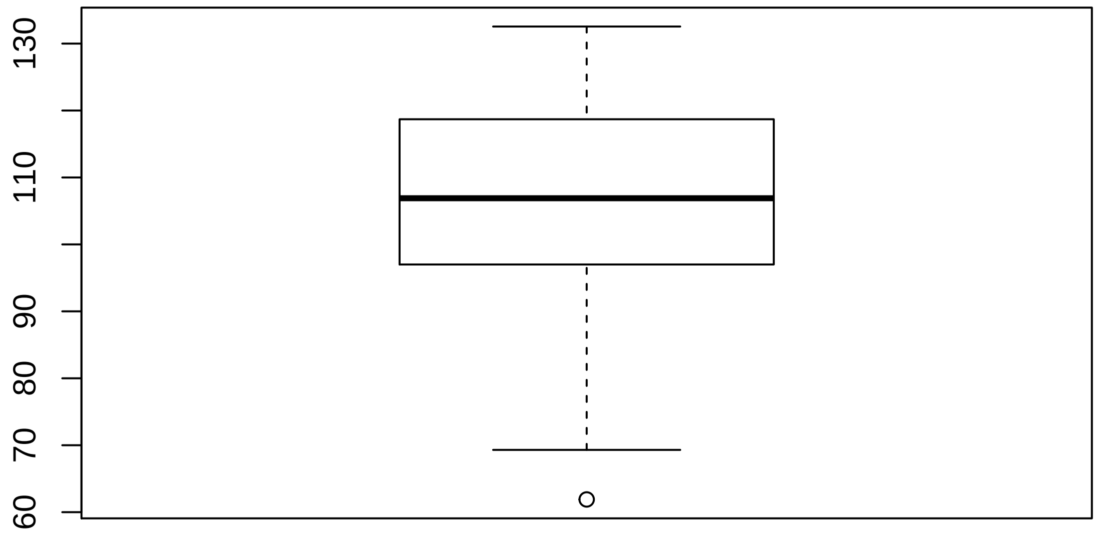
```
var(d$Lshell, na.rm=TRUE)
```

```
## [1] 220.3397
```

17.4 Simple boxplot of one variable

Useful for your own quick visualisation.

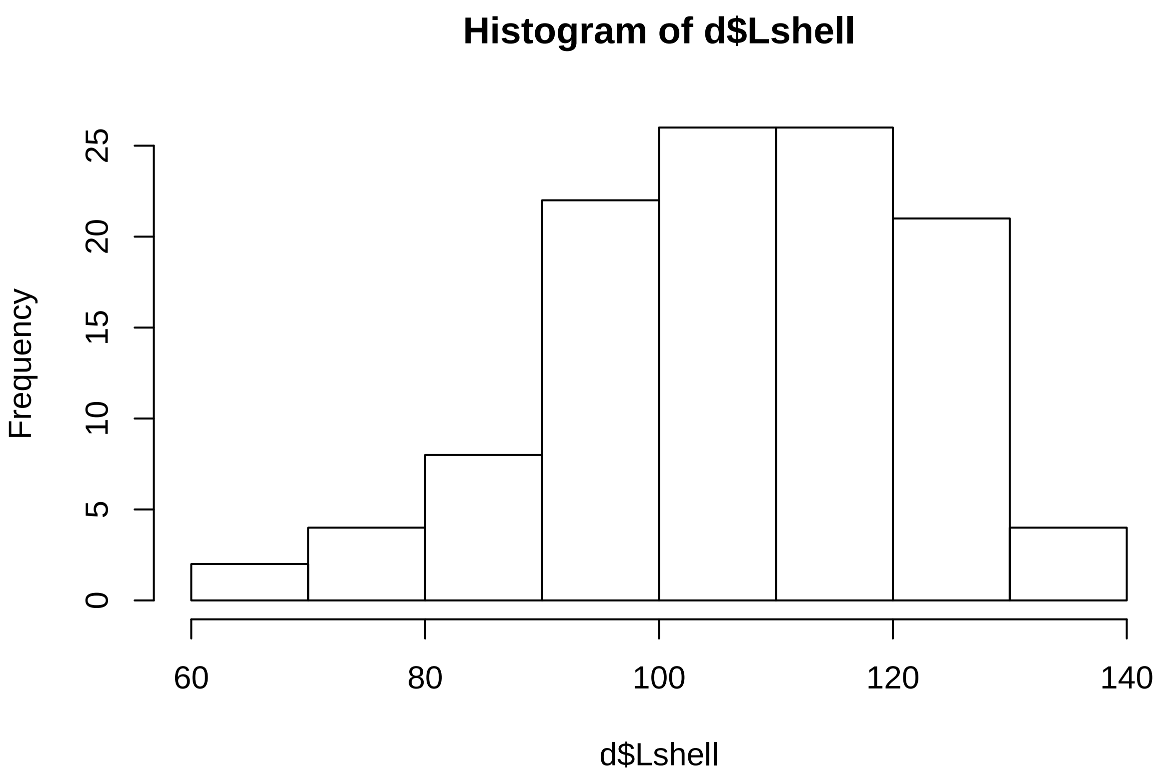
```
boxplot(d$Lshell)
```



17.5 Simple histogram of one variable

Useful for your own quick visualisation.

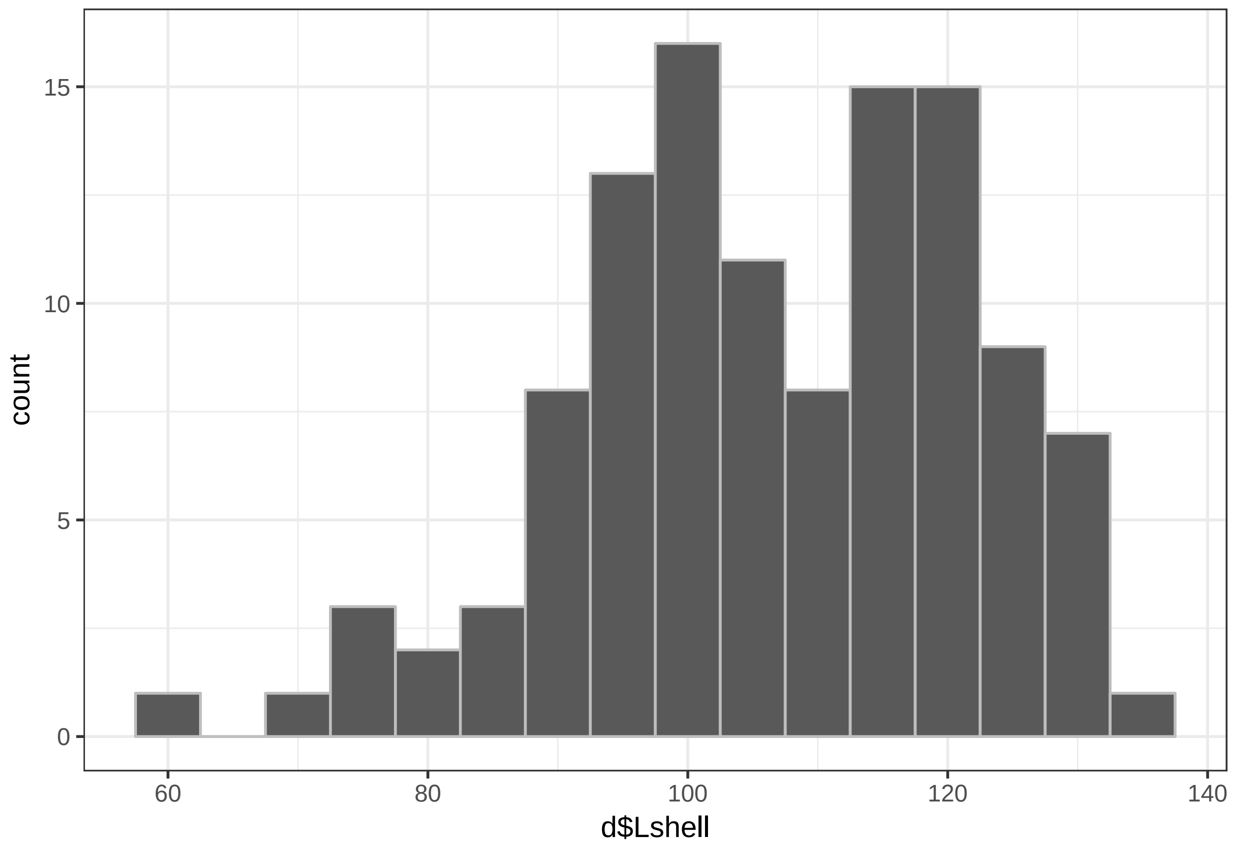
```
hist(d$Lshell)
```



17.6 Neater histogram of one variable

This uses ggplot. Change the bin width if you want to use this.

```
g0<-ggplot(d,aes(x=d$Lshell))
g0+geom_histogram(color="grey",binwidth = 5)
```


count
15
10
5
0
60
80
100
120
140
d$Lshell

Chapter 18

Regression

18.1 Data

In this data set there are two numerical variables. So we can run a linear regresion.

```
d<-read.csv("https://tinyurl.com/aqm-data/mussels.csv")
dt(d)
```

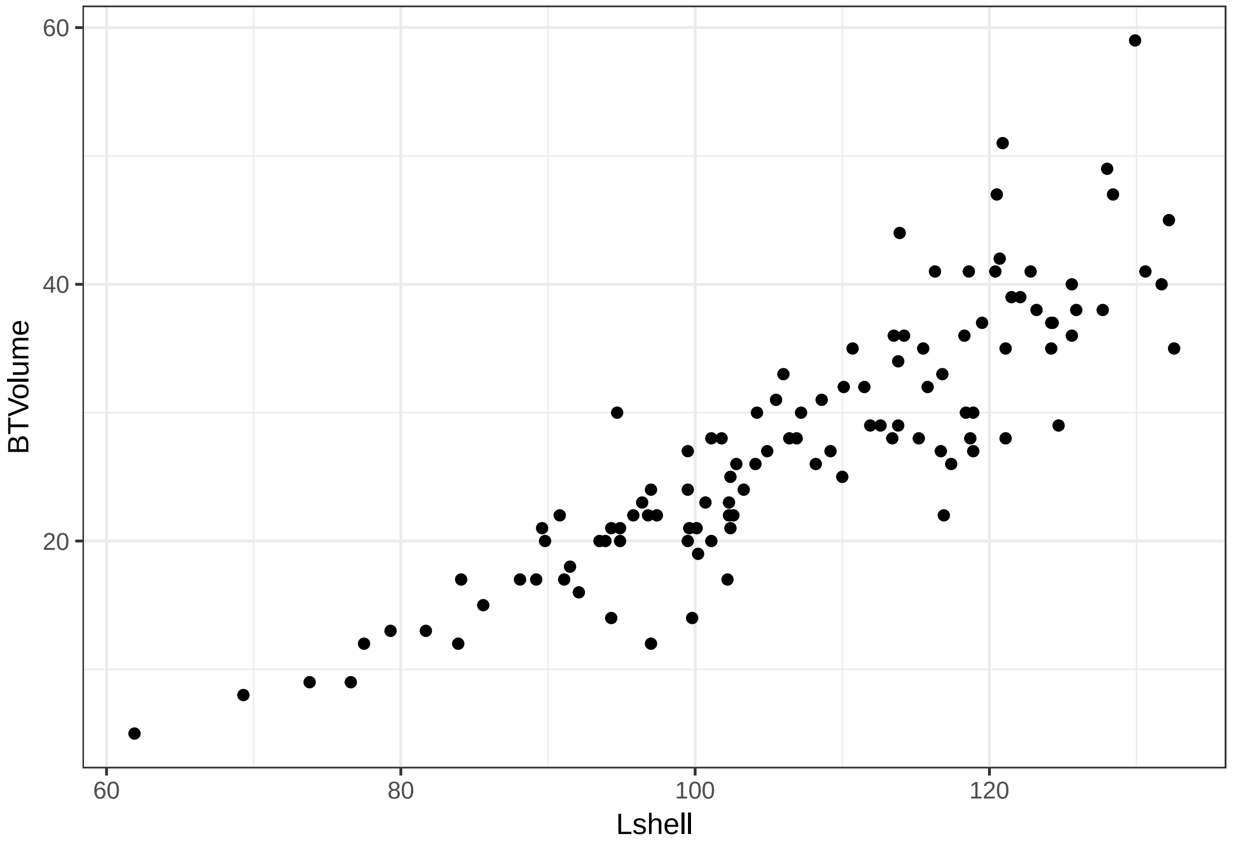
Copy CSV Show 10 entries Search:

	Lshell	BTVolume	Site
	All	All	All
1	122.1	39	Site_6
2	100.1	21	Site_6
3	100.7	23	Site_6
4	102.3	22	Site_6
5	94.9	20	Site_6
6	116.9	22	Site_6
7	94.9	21	Site_6
8	91.5	18	Site_6
9	94.3	21	Site_6
10	85.6	15	Site_6

Showing 1 to 10 of 113 entries Previous 1 2 3 4 5 … 12 Next

18.2 Scatterplot without fitted line

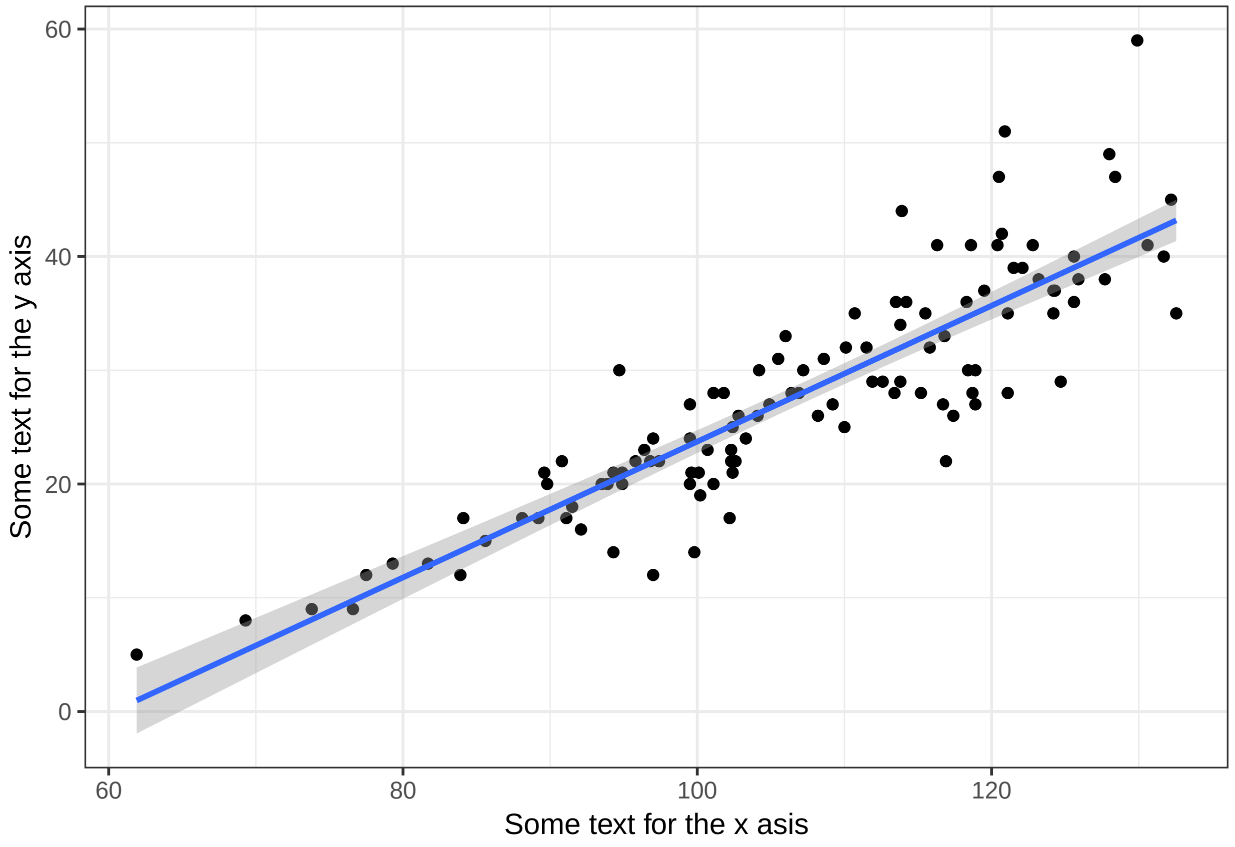
```
g0<-ggplot(d,aes(x=Lshell,y=BTVolume))
g0+geom_point()
```



18.3 Scatterplot with fitted line and labels

Type the text you want for the x and y axes to replace the variable names

```
g0<-ggplot(d,aes(x=Lshell,y=BTVolume))
g1<-g0+geom_point() + geom_smooth(method="lm")
g1 + xlab("Some text for the x asis") + ylab("Some text for the y axis")
```



18.4 Fitting a model

Change the names of the variables in the first line.

```
mod<-lm(data= d, BTVolume~Lshell)
```

18.4.1 Model summary

```
summary(mod)
```

```
##
## Call:
## lm(formula = BTVolume ~ Lshell, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.828  -2.672   0.147   2.235  17.404
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.02385    3.33917  -10.79   <2e-16 ***
## Lshell        0.59754    0.03096   19.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.864 on 111 degrees of freedom
## Multiple R-squared:  0.7704, Adjusted R-squared:  0.7684
## F-statistic: 372.5 on 1 and 111 DF,  p-value: < 2.2e-16
```

18.4.2 Model anova table

```
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: BTVolume
##            Df Sum Sq Mean Sq F value    Pr(>F)
## Lshell      1 8811.4  8811.4  372.49 < 2.2e-16 ***
## Residuals 111 2625.7    23.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

18.4.3 Confidence intervals for the model parameters

```
confint(mod)
```

```
##                   2.5 %      97.5 %
## (Intercept) -42.6406346 -29.4070662
## Lshell        0.5361881   0.6588891
```

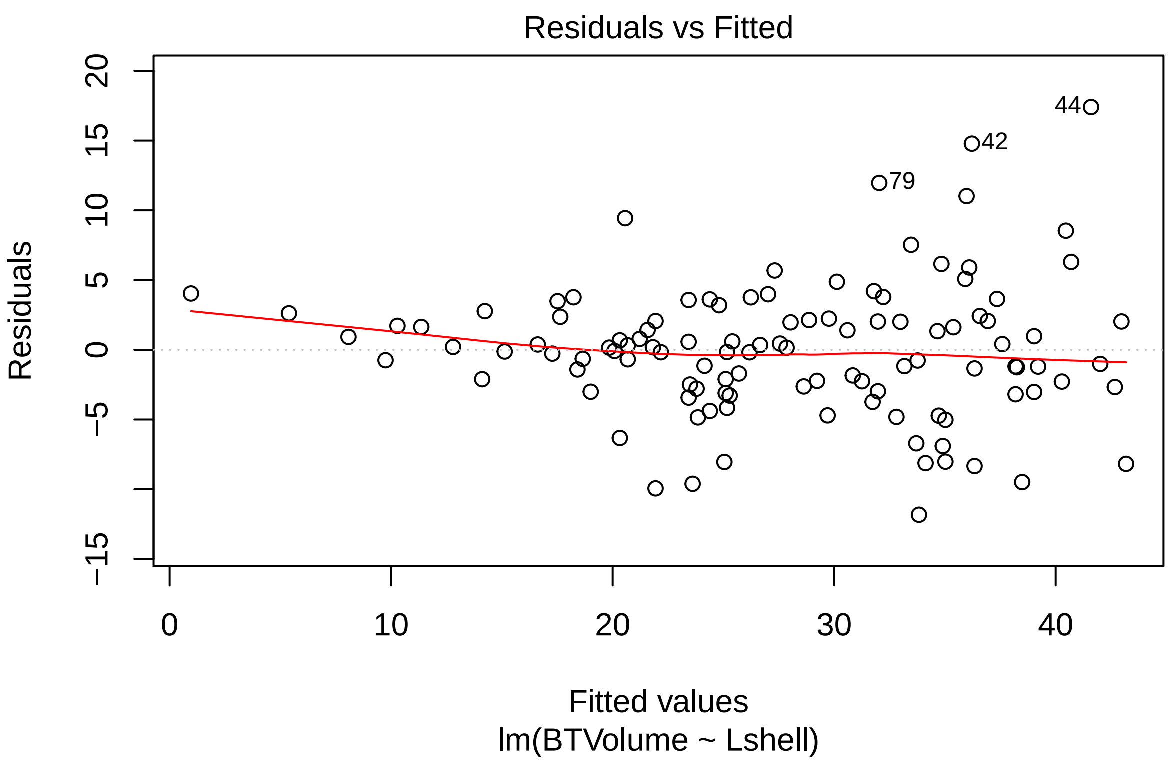
18.4.4 Extracting residuals

```
d$residuals<-residuals(mod)
```

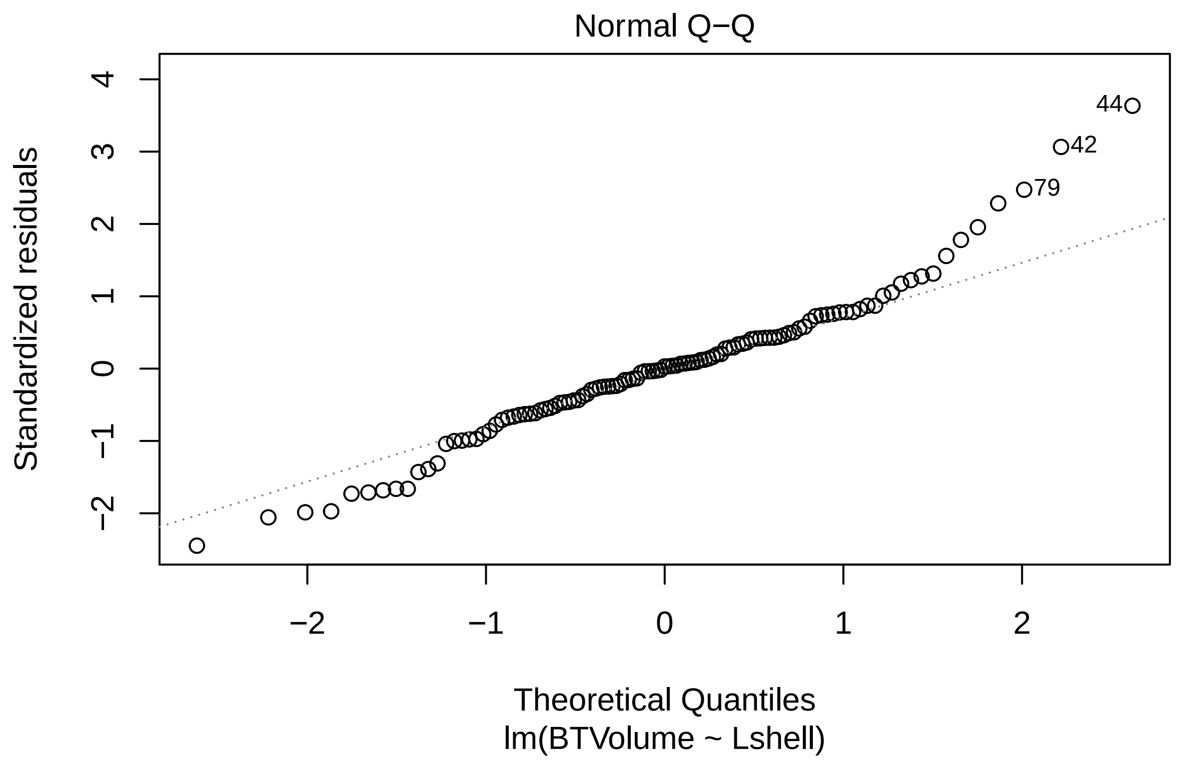
18.4.5 Model diagnostics

Look at the regression handout to understand these plots.

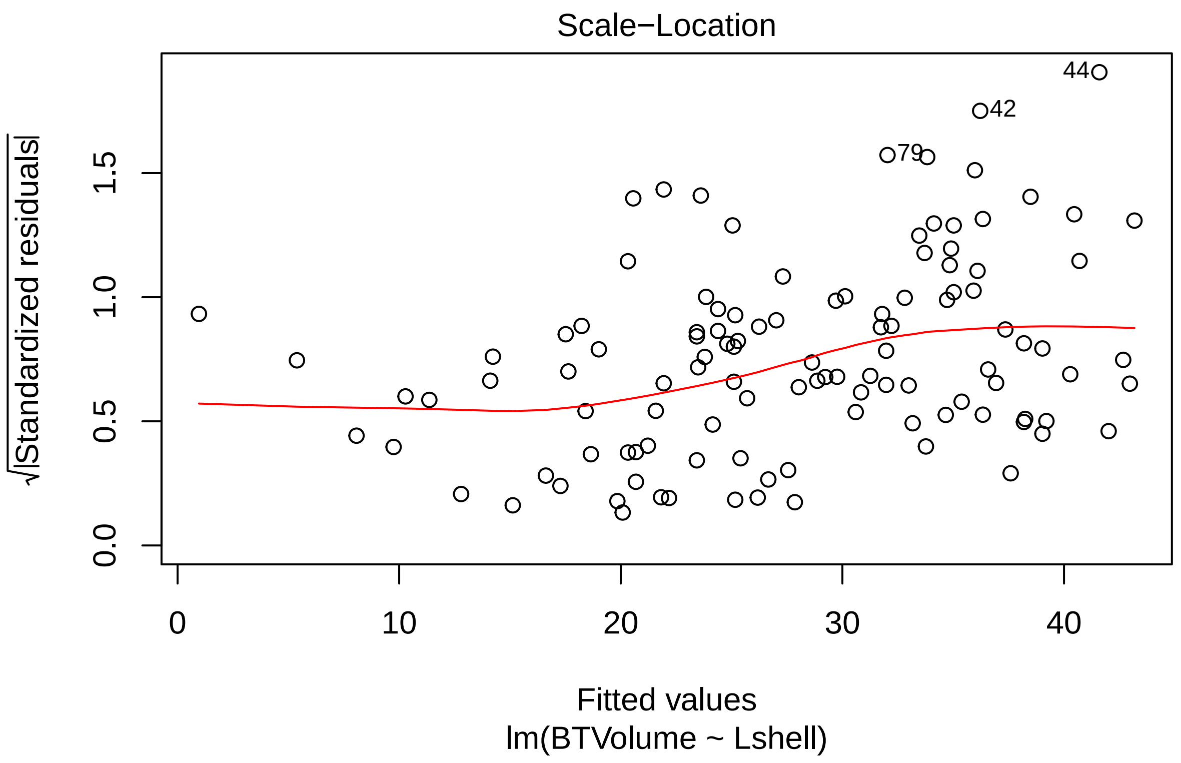
```
plot(mod,which=1)
```



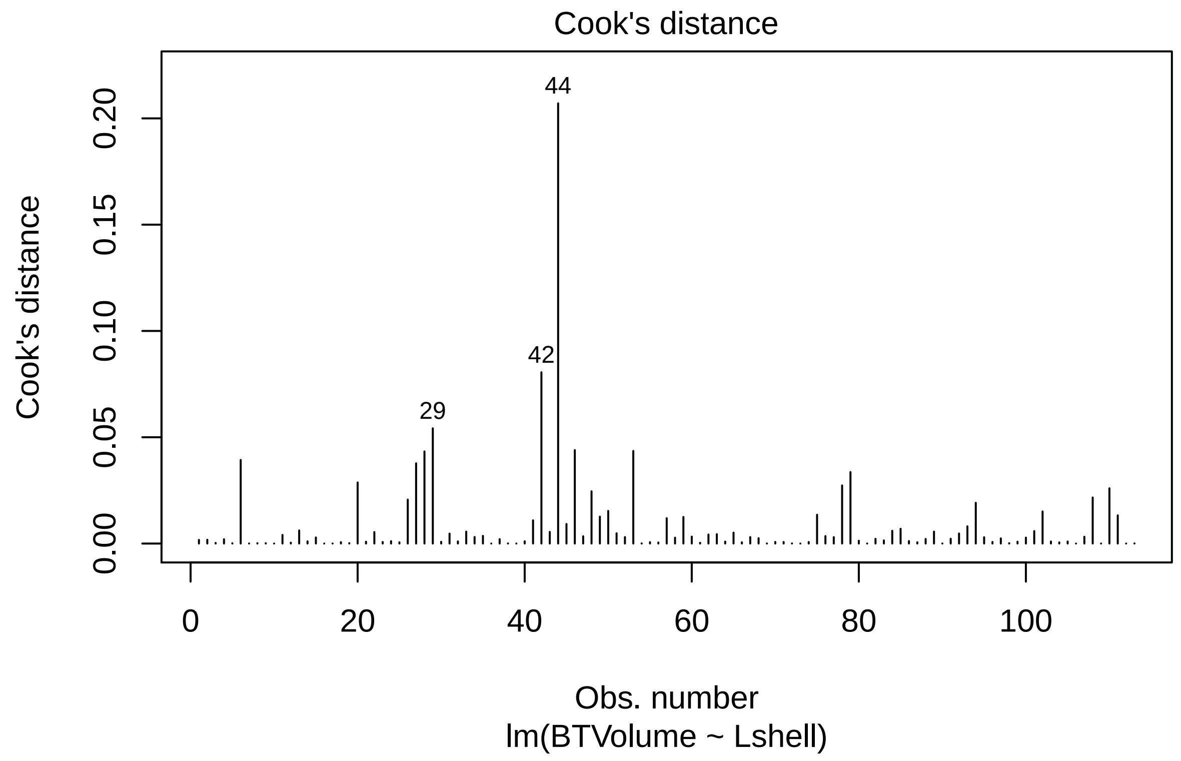
```
plot(mod,which=2)
```



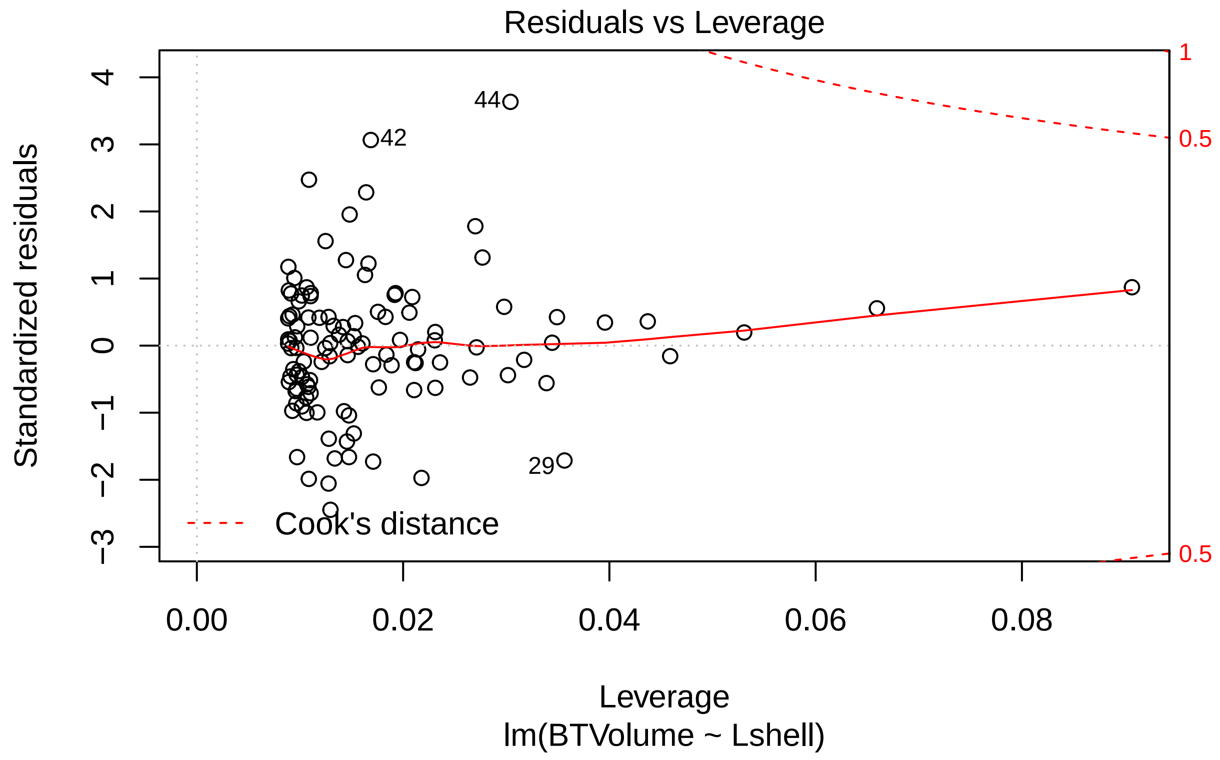
```
plot(mod,which=3)
```



```
plot(mod,which=4)
```



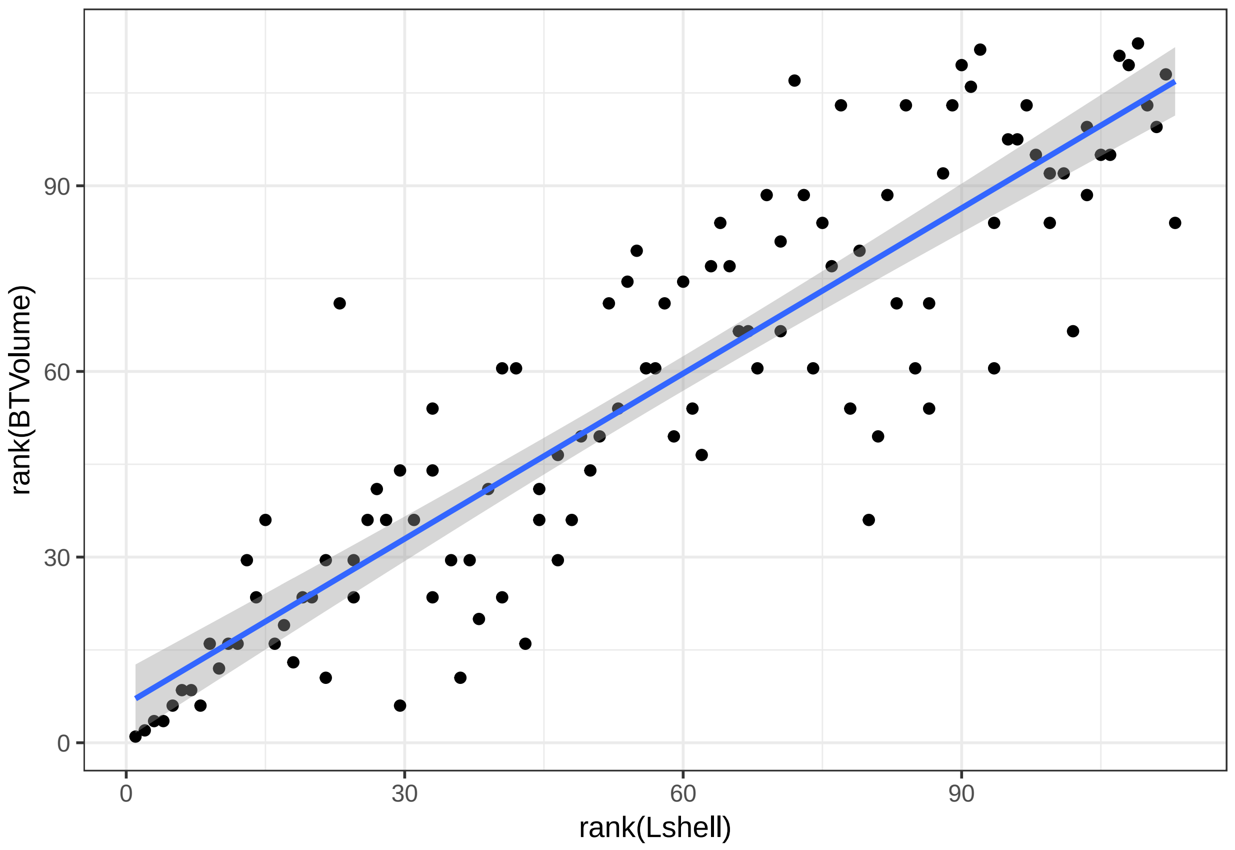
```
plot(mod,which=5)
```



18.4.6 Spearman's rank correlation

Used if all else fails. Not needed with these data, but included for reference.

```
g0<-ggplot(d,aes(x=rank(Lshell),y=rank(BTVolume)))
g0+geom_point() + geom_smooth(method="lm")
```



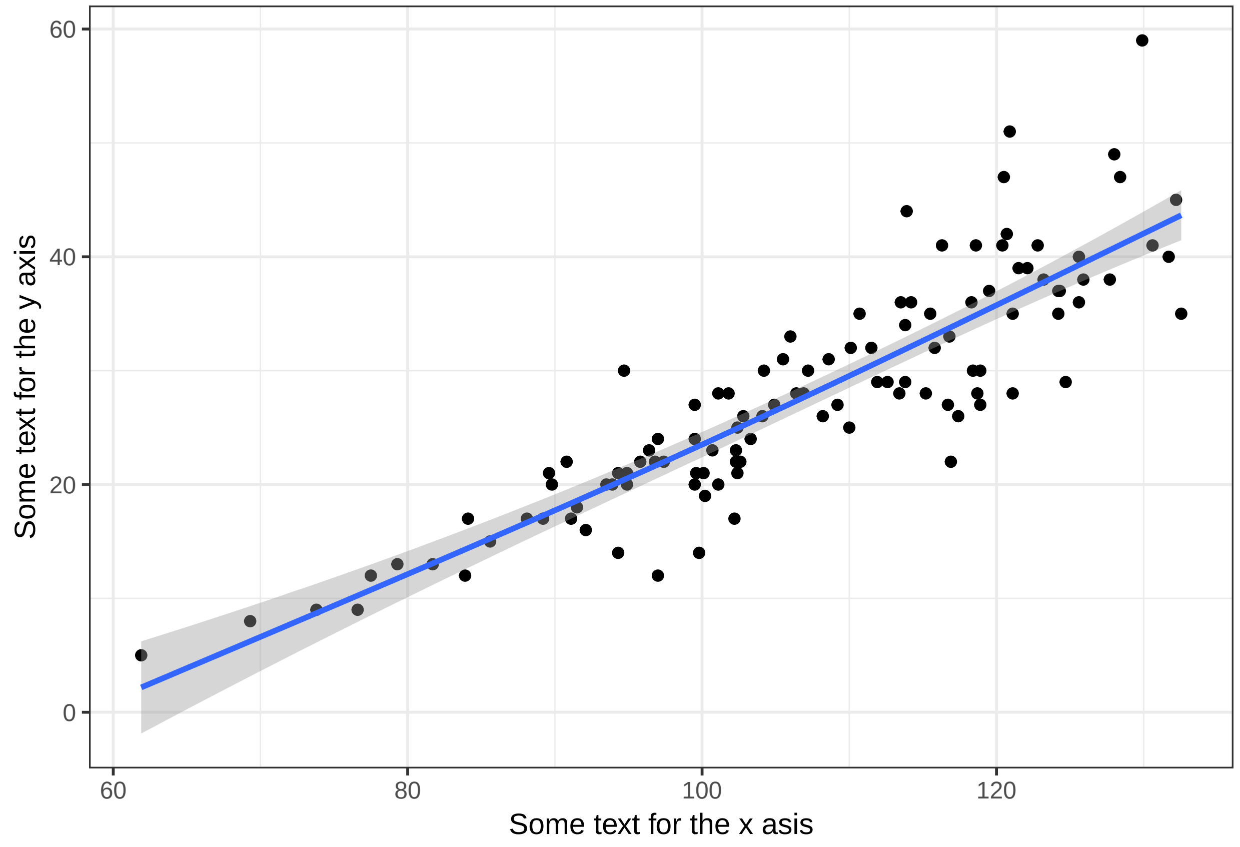
```
cor.test(d$Lshell,d$BTVolume,method="spearman")
```

```
##
##  Spearman's rank correlation rho
##
## data:  d$Lshell and d$BTVolume
## S = 26143, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8912809
```

18.5 Fitting a spline

Only use if you suspect that the relationship is not well described by a straight line.

```
library(mgcv)

g0<-ggplot(d,aes(x=Lshell,y=BTVolume))
g1<-g0 + geom_point() + geom_smooth(method="gam", formula =y~s(x))
g1 + xlab("Some text for the x asis") + ylab("Some text for the y axis")
```



In this case the line is the same as the linear model. Get a summary using this code.

```r
mod<-gam(data=d, BTVolume~s(Lshell))
summary(mod)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## BTVolume ~ s(Lshell)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.8142     0.4557   61.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df     F p-value
## s(Lshell) 1.493  1.847 198.8  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.77   Deviance explained = 77.3%
## GCV = 23.993  Scale est. = 23.463    n = 113
```

If you do use this model remember that **its only needed if you can't use linear regression**. Report the ajusted R squared value, the estimated degrees of freedom and the p-value for the smooth term (not the intercept). You **must** include the figure in your report, as that is the only way to show the shape of the response.

Chapter 19

One way ANOVA

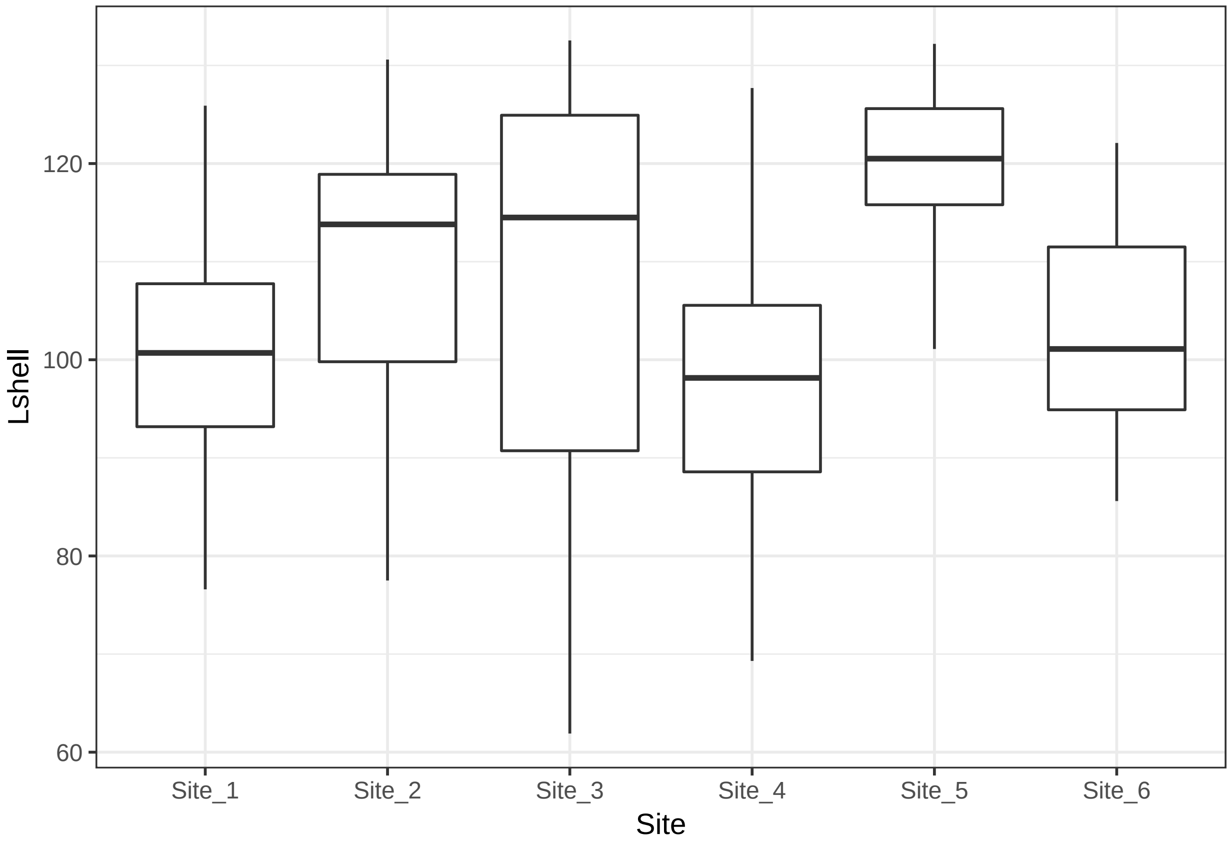
The purpose of one way anova is

1. Test whether there is greater variability between groups than within groups
2. Quantify any differences found between group means

19.1 Grouped boxplots

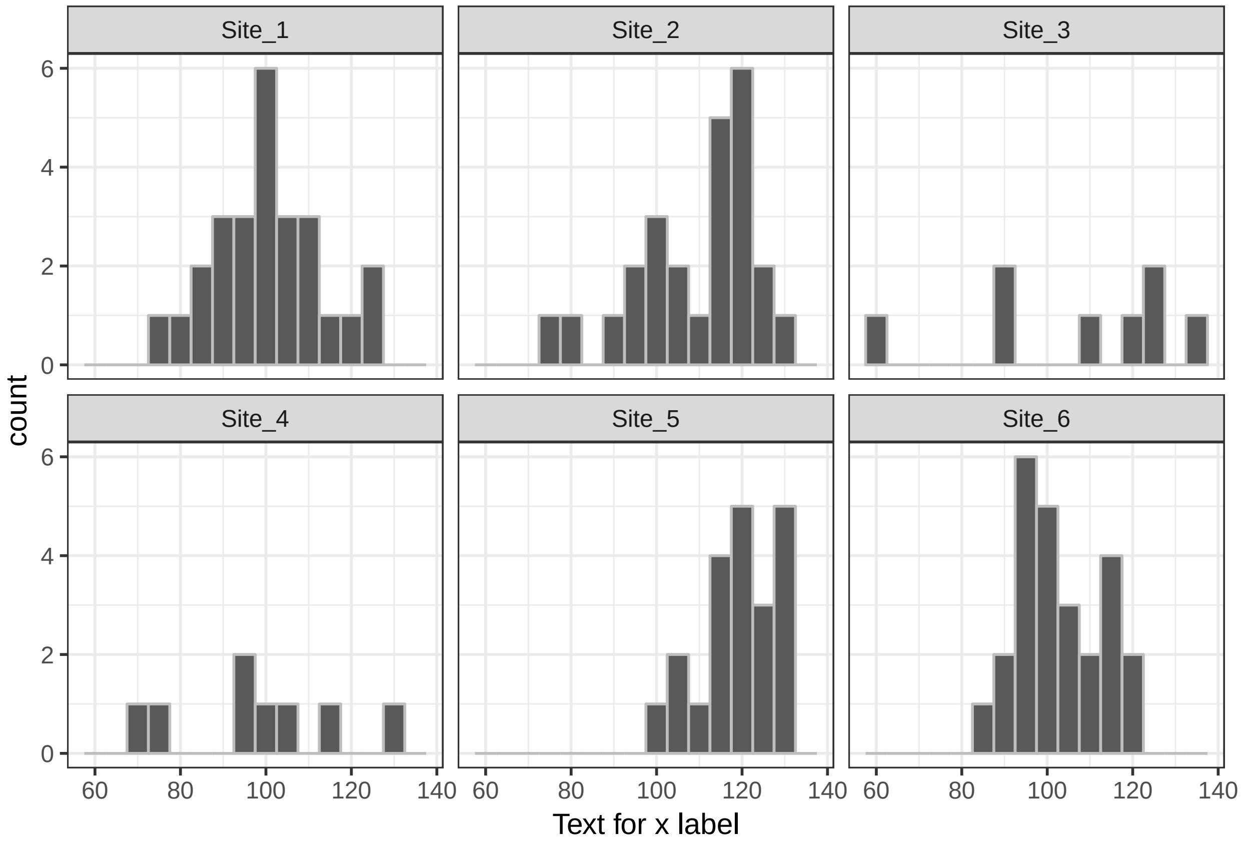
Exploratory plots

```
g0<-ggplot(d,aes(x=Site,y=Lshell))
g0+geom_boxplot()
```



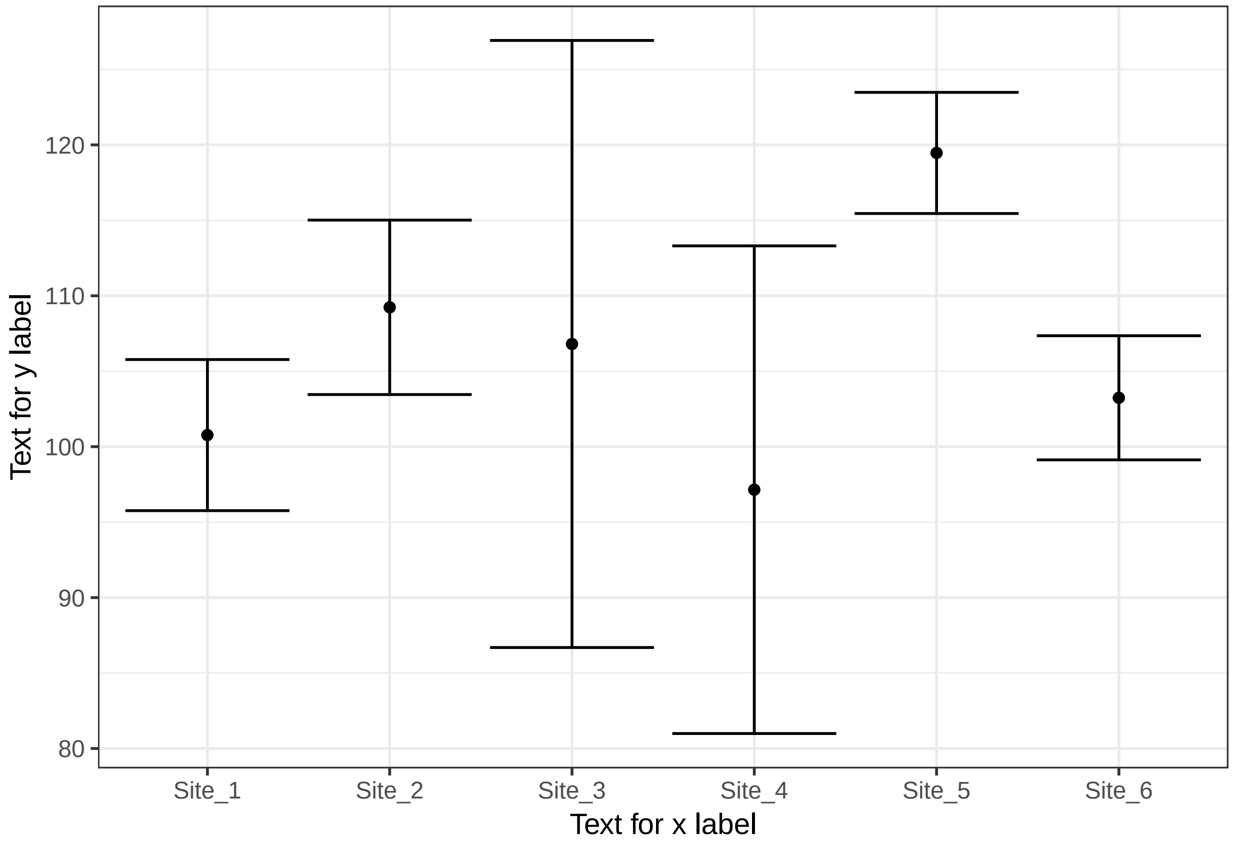
19.2 Histograms for each factor level

```r
g0<-ggplot(d,aes(x=d$Lshell))
g1<-g0+geom_histogram(color="grey",binwidth = 5)

g1+facet_wrap(~Site) +xlab("Text for x label")
```



19.3 Confidence interval plot

```
g0<-ggplot(d,aes(x=Site,y=Lshell))
g1<-g0+stat_summary(fun.y=mean,geom="point")
g1<-g1 +stat_summary(fun.data=mean_cl_normal,geom="errorbar")
g1 +xlab("Text for x label") + ylab("Text for y label")
```



19.4 Fitting ANOVA

```
mod<-lm(data=d,Lshell~Site)
```

19.4.1 Model anova

```
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: Lshell
##            Df Sum Sq Mean Sq F value    Pr(>F)
## Site        5   5525    1105  6.1732 4.579e-05 ***
## Residuals 107  19153     179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

19.4.2 Model summary

19.4.2.1 Treatment effects

```r
summary(mod)
```

```
##
## Call:
## lm(formula = Lshell ~ Site, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.906  -8.340   1.031   9.231  30.550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  100.769      2.624  38.405  < 2e-16 ***
## SiteSite_2     8.467      3.748   2.259   0.0259 *
## SiteSite_3     6.037      5.409   1.116   0.2669
## SiteSite_4    -3.619      5.409  -0.669   0.5049
## SiteSite_5    18.697      3.925   4.763 6.02e-06 ***
## SiteSite_6     2.471      3.748   0.659   0.5111
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.38 on 107 degrees of freedom
## Multiple R-squared:  0.2239, Adjusted R-squared:  0.1876
## F-statistic: 6.173 on 5 and 107 DF,  p-value: 4.579e-05
```

19.4.2.2 Change reference level

```r
slevels<-levels(d$Site)
d$Site<-relevel(d$Site,"Site_5")
mod<-lm(data=d,Lshell~Site)
summary(mod)
```

```
##
## Call:
## lm(formula = Lshell ~ Site, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.906  -8.340   1.031   9.231  30.550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  119.467      2.920  40.919  < 2e-16 ***
## SiteSite_1   -18.697      3.925  -4.763 6.02e-06 ***
## SiteSite_2   -10.231      3.960  -2.583 0.011135 *
## SiteSite_3   -12.660      5.559  -2.278 0.024739 *
## SiteSite_4   -22.317      5.559  -4.015 0.000111 ***
## SiteSite_6   -16.227      3.960  -4.097 8.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.38 on 107 degrees of freedom
## Multiple R-squared:  0.2239, Adjusted R-squared:  0.1876
## F-statistic: 6.173 on 5 and 107 DF,  p-value: 4.579e-05
```

```
d$Site <- factor(d$Site, levels=slevels)
```

19.4.2.3 Reverse levels

```
slevels<-levels(d$Site)
d$Site <- factor(d$Site, levels=rev(slevels))
mod<-lm(data=d,Lshell~Site)
summary(mod)
```

```
##
## Call:
## lm(formula = Lshell ~ Site, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.906  -8.340   1.031   9.231  30.550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  103.240      2.676  38.583  < 2e-16 ***
## SiteSite_5    16.227      3.960   4.097 8.15e-05 ***
## SiteSite_4    -6.090      5.435  -1.121    0.265
## SiteSite_3     3.566      5.435   0.656    0.513
## SiteSite_2     5.996      3.784   1.584    0.116
## SiteSite_1    -2.471      3.748  -0.659    0.511
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.38 on 107 degrees of freedom
## Multiple R-squared:  0.2239, Adjusted R-squared:  0.1876
## F-statistic: 6.173 on 5 and 107 DF,  p-value: 4.579e-05
```

```
d$Site <- factor(d$Site, levels=slevels)
```

19.4.2.4 Sum contrasts

Sum contrasts compare the effects to the mean. Notice that the last level is missing due to the way the design matrix is formed. So it can be worth running sum contrasts twice, with the order reversed, to get all the contrasts.

```
options(contrasts = c("contr.sum", "contr.poly"))
mod<-lm(data=d,Lshell~Site)
summary(mod)
```

```
##
## Call:
## lm(formula = Lshell ~ Site, data = d)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.906  -8.340   1.031   9.231  30.550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 106.1114     1.4384  73.773  < 2e-16 ***
## Site1        -5.3421     2.5804  -2.070   0.0408 *
## Site2         3.1246     2.6158   1.195   0.2349
## Site3         0.6949     4.1214   0.169   0.8664
## Site4        -8.9614     4.1214  -2.174   0.0319 *
## Site5        13.3553     2.7841   4.797 5.24e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.38 on 107 degrees of freedom
## Multiple R-squared:  0.2239, Adjusted R-squared:  0.1876
## F-statistic: 6.173 on 5 and 107 DF,  p-value: 4.579e-05
```

```
options(contrasts = c("contr.treatment", "contr.poly"))
```

19.4.2.4.1 Reverse order

```
d$Site <- factor(d$Site, levels=rev(slevels))
options(contrasts = c("contr.sum", "contr.poly"))
mod<-lm(data=d,Lshell~Site)
summary(mod)
```

```
##
## Call:
## lm(formula = Lshell ~ Site, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.906  -8.340   1.031   9.231  30.550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 106.1114     1.4384  73.773  < 2e-16 ***
## Site1        -2.8714     2.6158  -1.098   0.2748
## Site2        13.3553     2.7841   4.797 5.24e-06 ***
## Site3        -8.9614     4.1214  -2.174   0.0319 *
## Site4         0.6949     4.1214   0.169   0.8664
## Site5         3.1246     2.6158   1.195   0.2349
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.38 on 107 degrees of freedom
## Multiple R-squared:  0.2239, Adjusted R-squared:  0.1876
## F-statistic: 6.173 on 5 and 107 DF,  p-value: 4.579e-05
```

```
options(contrasts = c("contr.treatment", "contr.poly"))
d$Site <- factor(d$Site, levels=slevels)
```

19.4.3 Tukey corrected pairwise comparisons

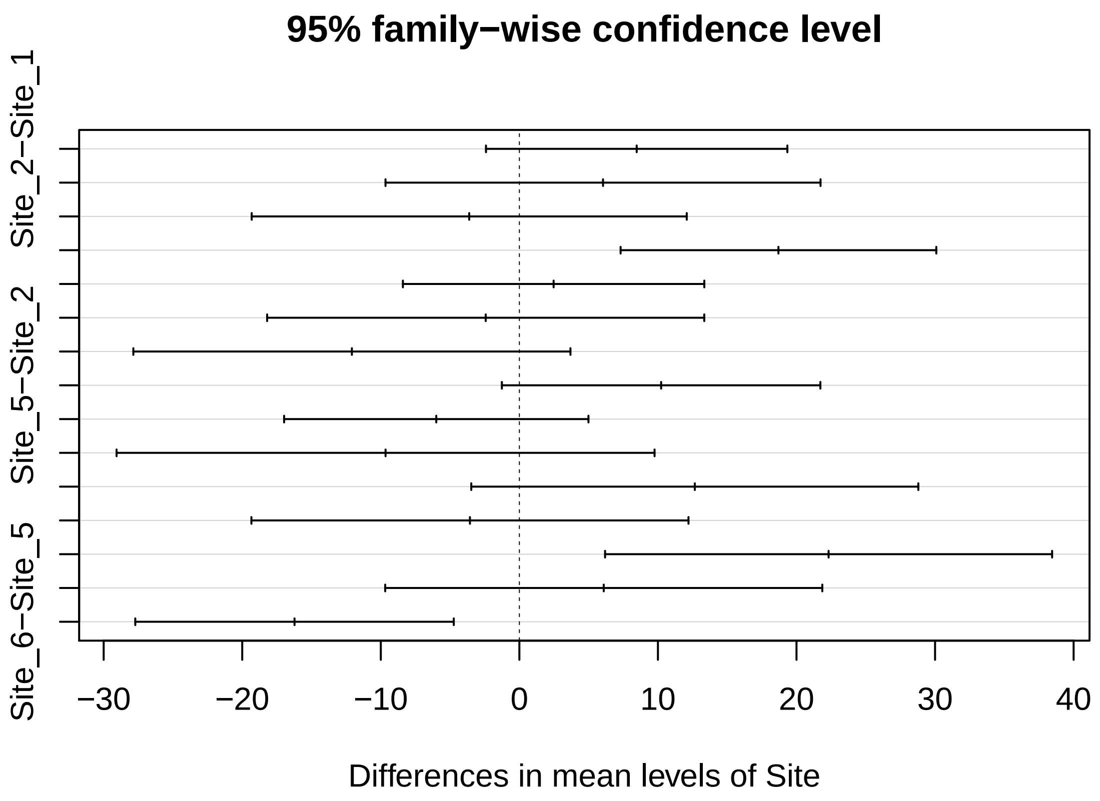
Use to find where signficant differences lie. This should confirm the pattern shown using the confidence interval plot.

```
mod<-aov(data=d,Lshell~Site)
TukeyHSD(mod)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Lshell ~ Site, data = d)
##
## $Site
##                      diff        lwr       upr     p adj
## Site_2-Site_1   8.466769  -2.408905 19.342443 0.2201442
## Site_3-Site_1   6.037019  -9.660664 21.734702 0.8737518
## Site_4-Site_1  -3.619231 -19.316914 12.078452 0.9849444
## Site_5-Site_1  18.697436   7.305950 30.088922 0.0000867
## Site_6-Site_1   2.470769  -8.404905 13.346443 0.9859123
## Site_3-Site_2  -2.429750 -18.201132 13.341632 0.9976925
## Site_4-Site_2 -12.086000 -27.857382  3.685382 0.2355928
## Site_5-Site_2  10.230667  -1.262165 21.723498 0.1103764
## Site_6-Site_2  -5.996000 -16.977781  4.985781 0.6105029
## Site_4-Site_3  -9.656250 -29.069479  9.756979 0.7004668
## Site_5-Site_3  12.660417  -3.470986 28.791819 0.2123990
## Site_6-Site_3  -3.566250 -19.337632 12.205132 0.9862071
## Site_5-Site_4  22.316667   6.185264 38.448069 0.0015143
## Site_6-Site_4   6.090000  -9.681382 21.861382 0.8718474
## Site_6-Site_5 -16.226667 -27.719498 -4.733835 0.0011239
```

Plot of results of Tukey HSD

```
plot(TukeyHSD(mod))
```



19.4.4 Anova with White's correction

This will give you the overall Anova table if there is heterogeneity of variance.

```
library(sandwich)
library(car)
mod<-lm(Lshell~Site, data=d)
Anova(mod,white.adjust='hc3')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Lshell
##            Df      F    Pr(>F)
## Site        5 9.9682 7.541e-08 ***
## Residuals 107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

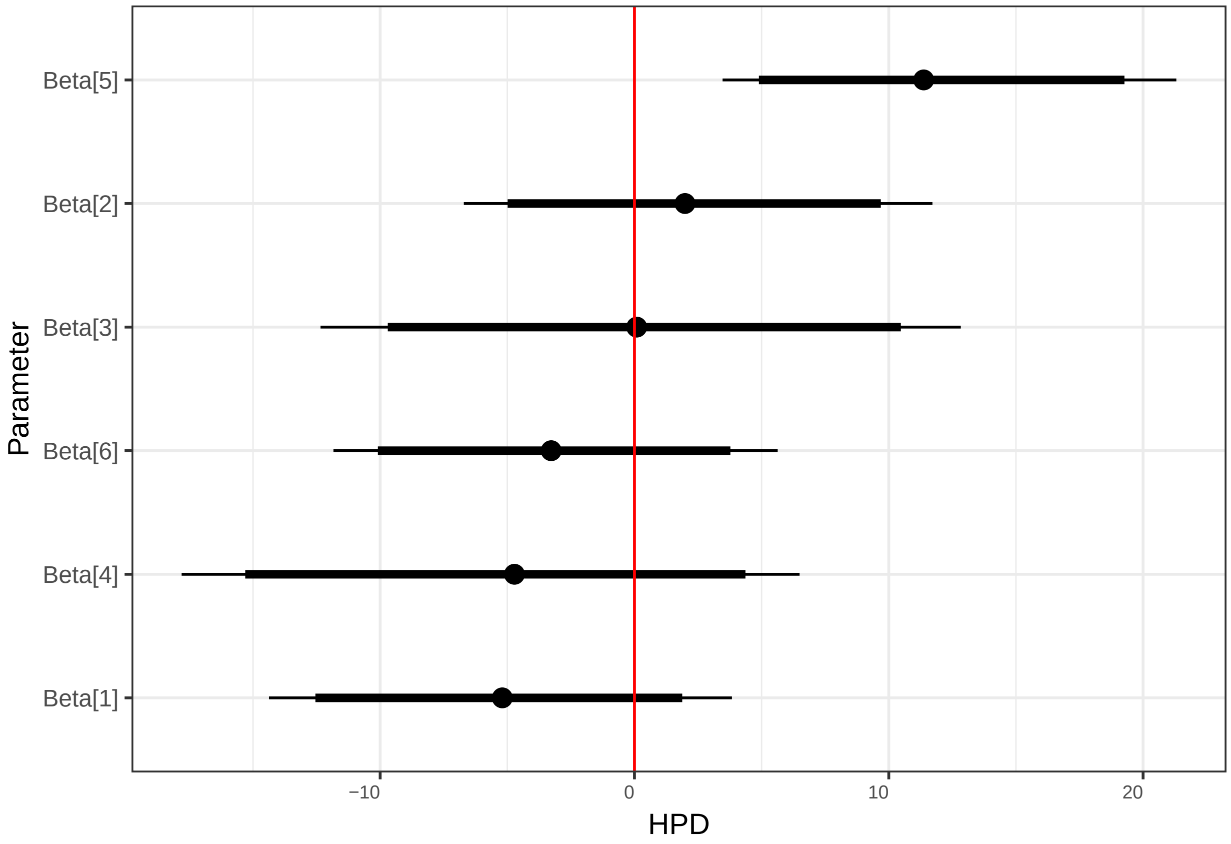
19.4.5 Bayesian model with shrinkage

Specialist model. Probably the best for these particular data, but seek guidance. **Don't do this at home kids!**

```
library(rjags)
library(ggmcmc)
rand_mod="
  model {
      ### Likeihood
      for (i in 1:N) {                  ## Loop through observations
        mu[i]<-mu_r+Beta[ind[i]]        ## Beta is added to an overall mean
        y[i] ~ dnorm(mu[i],tau[ind[i]])  ## Set an independent tau for each group agan. A

      }

    for (j in 1:p) {
     Beta[j]~dnorm(0,tau_r)            ## A single tau represents the variance between group
     tau[j] ~ dgamma(scale, rate)
      for (n in 1:(j-1)){
        Difbeta[n,j]<-Beta[n]-Beta[j]
      }
    }

    scale ~ dunif(0, 1)
    rate ~ dunif(0, 1)
    tau_r ~ dgamma(scale,rate)
    sigma_r <- 1/sqrt(tau_r)
    mu_r ~ dnorm(0,0.00001)     ## Prior for the overall mean

  }"

data=list(y=d$Lshell,
          ind=as.numeric(d$Site),
          N=length(d$Lshell),
          p=length(levels(d$Site)),
          overall_mean=mean(d$Lshell))
model=jags.model(textConnection(rand_mod),data=data)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 113
##    Unobserved stochastic nodes: 16
##    Total graph size: 288
##
## Initializing model
```

```
update(model,n.iter=1000)
output=coda.samples(model=model,variable.names=c("sigma_r","mu_r","Difbeta","Beta"),n.iter=100000,thin=
ms <-ggs(output)
mt<-filter(ms,grepl("Beta",Parameter))
ggs_caterpillar(mt) +geom_vline(xintercept = 0,col="red")
```



```
summary(output)
```

```
##
## Iterations = 2010:102000
## Thinning interval = 10
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                    Mean    SD Naive SE Time-series SE
## Beta[1]       -5.2222 4.510  0.04510        0.07726
## Beta[2]        2.1001 4.600  0.04600        0.07920
## Beta[3]        0.1333 6.228  0.06228        0.07727
## Beta[4]       -4.9979 6.131  0.06131        0.07190
## Beta[5]       11.6084 4.493  0.04493        0.08085
## Beta[6]       -3.2314 4.363  0.04363        0.07733
## Difbeta[1,2]  -7.3223 3.613  0.03613        0.03734
## Difbeta[1,3]  -5.3555 6.437  0.06437        0.06437
## Difbeta[2,3]   1.9668 6.507  0.06507        0.06507
## Difbeta[1,4]  -0.2242 6.178  0.06178        0.06594
## Difbeta[2,4]   7.0980 6.413  0.06413        0.06971
## Difbeta[3,4]   5.1312 8.081  0.08081        0.07870
## Difbeta[1,5] -16.8306 3.279  0.03279        0.03326
## Difbeta[2,5]  -9.5083 3.331  0.03331        0.03289
## Difbeta[3,5] -11.4752 6.372  0.06372        0.06715
## Difbeta[4,5] -16.6064 6.384  0.06384        0.06994
## Difbeta[1,6]  -1.9908 3.092  0.03092        0.03092
## Difbeta[2,6]   5.3315 3.341  0.03341        0.03401
## Difbeta[3,6]   3.3647 6.303  0.06303        0.06548
## Difbeta[4,6]  -1.7665 6.121  0.06121        0.06555
## Difbeta[5,6]  14.8399 2.942  0.02942        0.02942
## mu_r         106.7915 4.130  0.04130        0.07738
## sigma_r        8.2238 3.674  0.03674        0.04701
##
## 2. Quantiles for each variable:
##
##                  2.5%      25%       50%       75%    97.5%
## Beta[1]       -14.371  -7.9339  -5.19670  -2.55585   3.8319
## Beta[2]        -6.711  -0.7170   1.98651   4.75840  11.7173
## Beta[3]       -12.345  -3.7393   0.08753   3.92107  12.8327
## Beta[4]       -17.807  -8.7131  -4.71930  -1.05039   6.4925
## Beta[5]         3.465   8.7919  11.37242  14.14664  21.3074
## Beta[6]       -11.837  -5.8286  -3.27427  -0.71021   5.6321
## Difbeta[1,2] -14.379  -9.7944  -7.32246  -4.84330  -0.3078
## Difbeta[1,3] -18.333  -9.4895  -5.27167  -1.18625   7.2620
## Difbeta[2,3] -10.525  -2.2210   1.91444   6.10535  14.9385
## Difbeta[1,4] -12.335  -4.2228  -0.26517   3.73730  12.1918
## Difbeta[2,4]  -5.128   2.8516   6.97407  11.26692  20.0822
## Difbeta[3,4] -10.529  -0.2333   4.91541  10.28038  21.9897
## Difbeta[1,5] -23.157 -19.0516 -16.88190 -14.63848 -10.3687
## Difbeta[2,5] -16.090 -11.6641  -9.47519  -7.33179  -3.0101
## Difbeta[3,5] -24.781 -15.4480 -11.33631  -7.30224   0.6461
## Difbeta[4,5] -29.457 -20.8265 -16.45207 -12.32261  -4.3211
## Difbeta[1,6]  -8.105  -4.0324  -1.99638   0.04918   4.0750
## Difbeta[2,6]  -1.313   3.1251   5.35675   7.54022  11.8889
## Difbeta[3,6]  -9.318  -0.7084   3.38617   7.37996  15.8448
## Difbeta[4,6] -14.043  -5.6816  -1.69075   2.13254  10.2866
## Difbeta[5,6]   8.958  12.8888  14.85055  16.81179  20.6069
## mu_r          98.169 104.4844 106.89238 109.20635 114.8737
## sigma_r        3.744   5.8104   7.40849   9.65818  17.7280
```

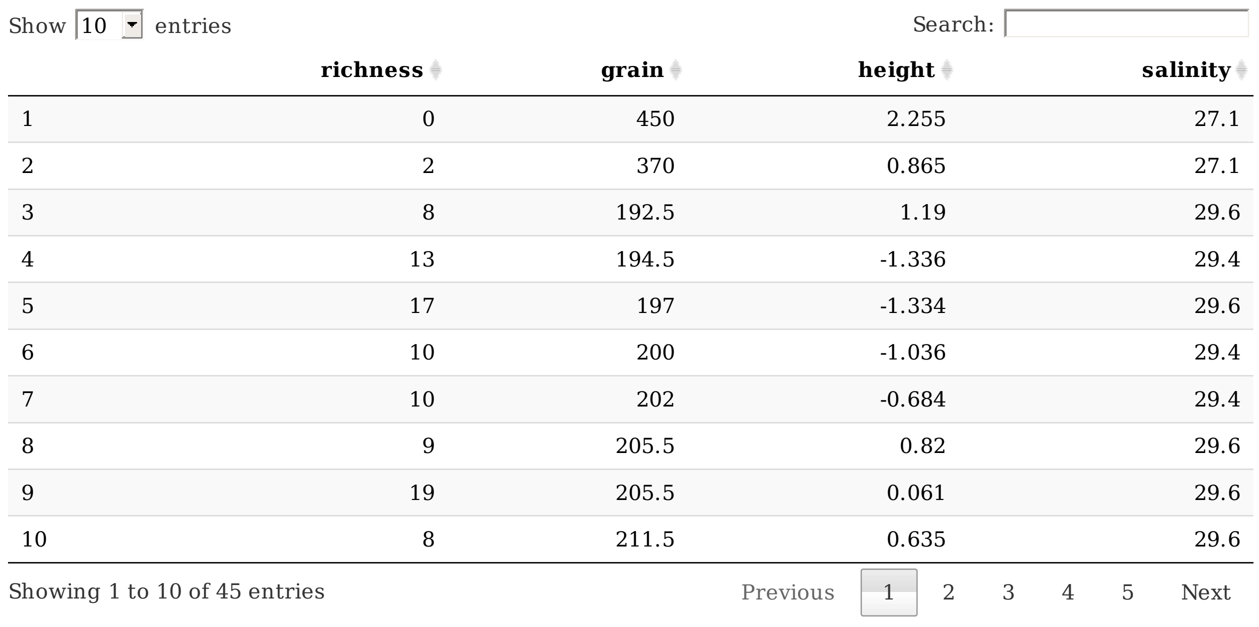
Chapter 20

Fitting curves cribsheet

20.1 Fitting polynomials

20.2 The data

```r
d<-read.csv("/home/aqm/course/data/marineinverts.csv")
DT::datatable(d)
```



Show 10 entries Search:

	richness	grain	height	salinity
1	0	450	2.255	27.1
2	2	370	0.865	27.1
3	8	192.5	1.19	29.6
4	13	194.5	-1.336	29.4
5	17	197	-1.334	29.6
6	10	200	-1.036	29.4
7	10	202	-0.684	29.4
8	9	205.5	0.82	29.6
9	19	205.5	0.061	29.6
10	8	211.5	0.635	29.6

Showing 1 to 10 of 45 entries Previous 1 2 3 4 5 Next

20.3 Fitting linear model

20.3.1 Linear model fit

```r
mod<-lm(data=d,richness~grain)
```

20.3.2 Linear model anova and summary

```
anova(mod)
```

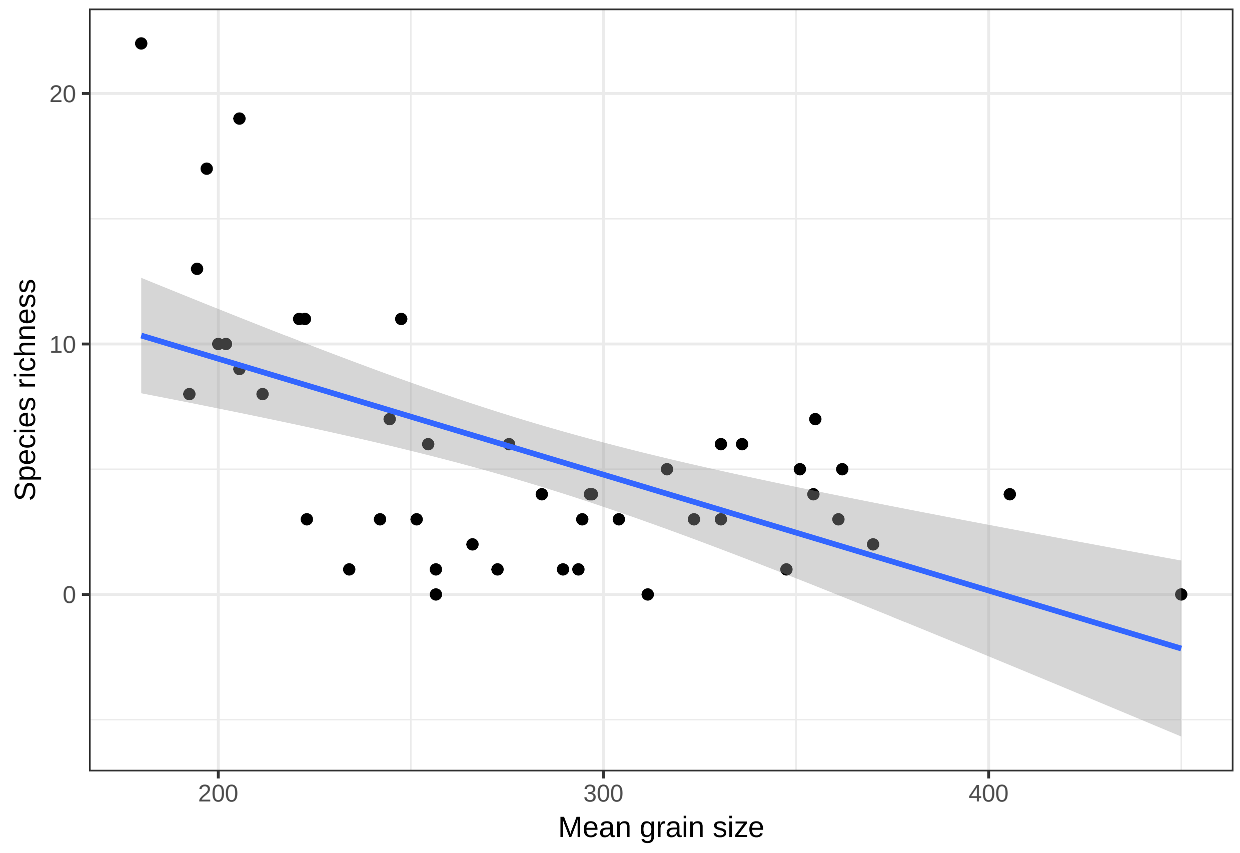
```
## Analysis of Variance Table
##
## Response: richness
##           Df Sum Sq Mean Sq F value    Pr(>F)
## grain      1 385.13  385.13  23.113 1.896e-05 ***
## Residuals 43 716.52   16.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = richness ~ grain, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8386 -2.0383 -0.3526  2.5768 11.6620
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.669264   2.767726   6.745 3.01e-08 ***
## grain       -0.046285   0.009628  -4.808 1.90e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.082 on 43 degrees of freedom
## Multiple R-squared:  0.3496, Adjusted R-squared:  0.3345
## F-statistic: 23.11 on 1 and 43 DF,  p-value: 1.896e-05
```

20.3.3 Linear model plot

```
library(ggplot2)
theme_set(theme_bw())
g0<-ggplot(d,aes(x=grain,y=richness))
g1<-g0+geom_point() + geom_smooth(method="lm")
g1 + xlab("Mean grain size") + ylab("Species richness")
```



20.3.4 Reset test

We can check whether a strait line is a good representation of the pattern using the reset test that will have a low p-value if the linear form of the model is not a good fit.

```
library(lmtest)
resettest(d$richness ~ d$grain)
```

```
##
##  RESET test
##
## data:  d$richness ~ d$grain
## RESET = 19.074, df1 = 2, df2 = 41, p-value = 1.393e-06
```

20.3.5 Durbin Watson test

The Durbin Watson test which helps to confirm serial autocorrelation that may be the result of a misformed model will often also be significant when residuals cluster on one side of the line. In this case it was not, but this may be because there were too few data points.

```
dwtest(d$richness~d$grain)
```

```
##
##  Durbin-Watson test
##
## data:  d$richness ~ d$grain
## DW = 1.7809, p-value = 0.1902
## alternative hypothesis: true autocorrelation is greater than 0
```

20.3.6 Diagnostic plot after fitting linear model

```
library(ggfortify)
```

```
## Error in library(ggfortify): there is no package called 'ggfortify'
```

```
theme_set(theme_bw())
autoplot(mod, which = 1)
```

```
## Error: Objects of type lm not supported by autoplot.
```

20.4 Fitting quadratic model

20.4.1 Quadratic model fit

```
mod2<-lm(data=d,richness~grain + I(grain^2))
```

20.4.2 Quadratic model anova and summary

```
anova(mod2)
```

```
## Analysis of Variance Table
##
## Response: richness
##             Df Sum Sq Mean Sq F value    Pr(>F)
## grain        1 385.13  385.13  29.811 2.365e-06 ***
## I(grain^2)   1 173.93  173.93  13.463   0.00068 ***
## Residuals   42 542.59   12.92
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

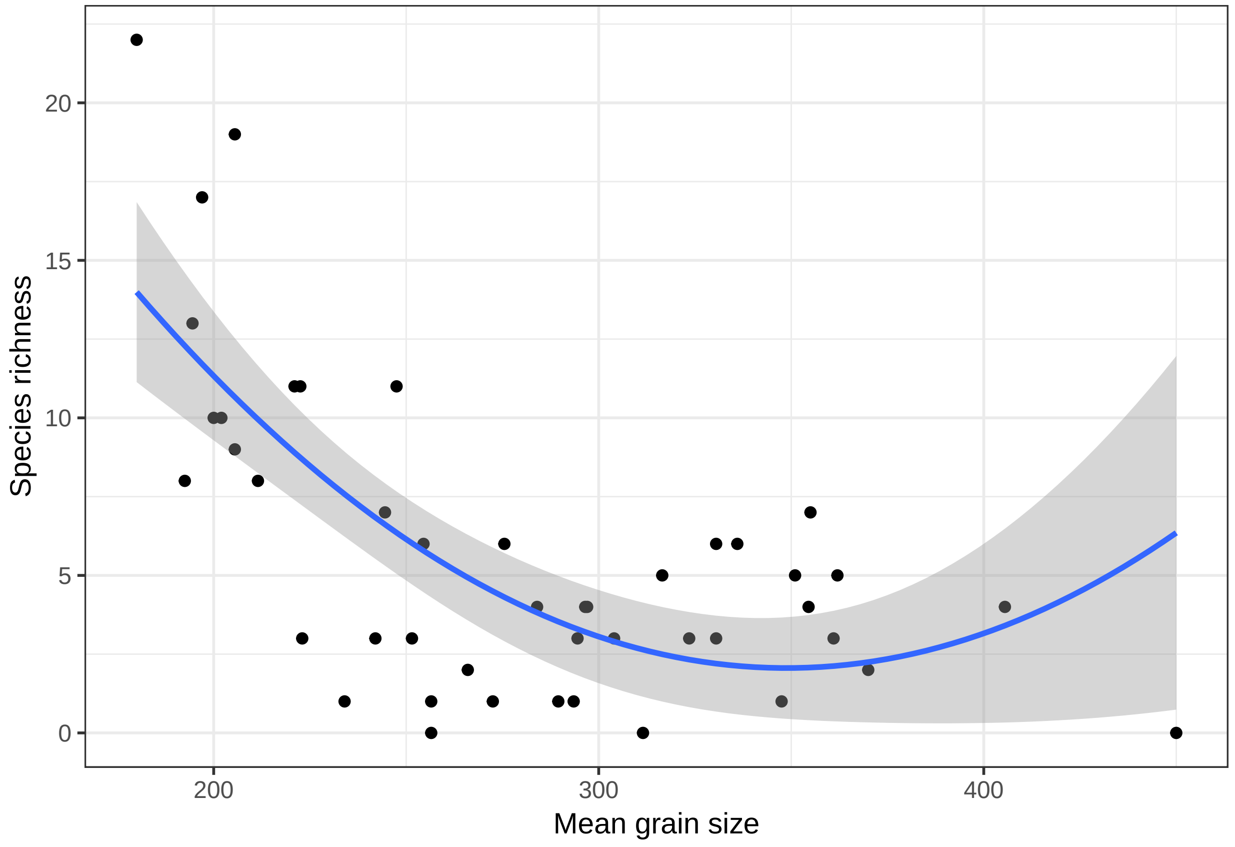
```
summary(mod2)
```

```
##
## Call:
## lm(formula = richness ~ grain + I(grain^2), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5779 -2.5315  0.2172  2.1013  8.3415
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 53.0133538  9.6721492   5.481 2.21e-06 ***
## grain       -0.2921821  0.0675505  -4.325 9.19e-05 ***
## I(grain^2)   0.0004189  0.0001142   3.669  0.00068 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.594 on 42 degrees of freedom
## Multiple R-squared:  0.5075, Adjusted R-squared:  0.484
## F-statistic: 21.64 on 2 and 42 DF,  p-value: 3.476e-07
```

20.4.3 Quadratic model plot

```
library(ggplot2)
theme_set(theme_bw())
g0<-ggplot(d,aes(x=grain,y=richness))
g1<-g0+geom_point() + geom_smooth(method="lm", formula=y~x+I(x^2), se=TRUE)
g1 + xlab("Mean grain size") + ylab("Species richness")
```



20.4.4 Diagnostic plot after fitting quadratic

```
library(ggfortify)
```

```
## Error in library(ggfortify): there is no package called 'ggfortify'
```

```
theme_set(theme_bw())
autoplot(mod2, which = 1)
```

```
## Error: Objects of type lm not supported by autoplot.
```

20.4.5 Comparing fits

```
anova(mod,mod2)
```

```
## Analysis of Variance Table
##
## Model 1: richness ~ grain
## Model 2: richness ~ grain + I(grain^2)
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     43 716.52
## 2     42 542.59  1    173.93 13.463 0.00068 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chapter 21

Fitting splines using mgcv

21.1 Fiting model

```
library(mgcv)
mod3<-gam(data=d,richness~s(grain))
```

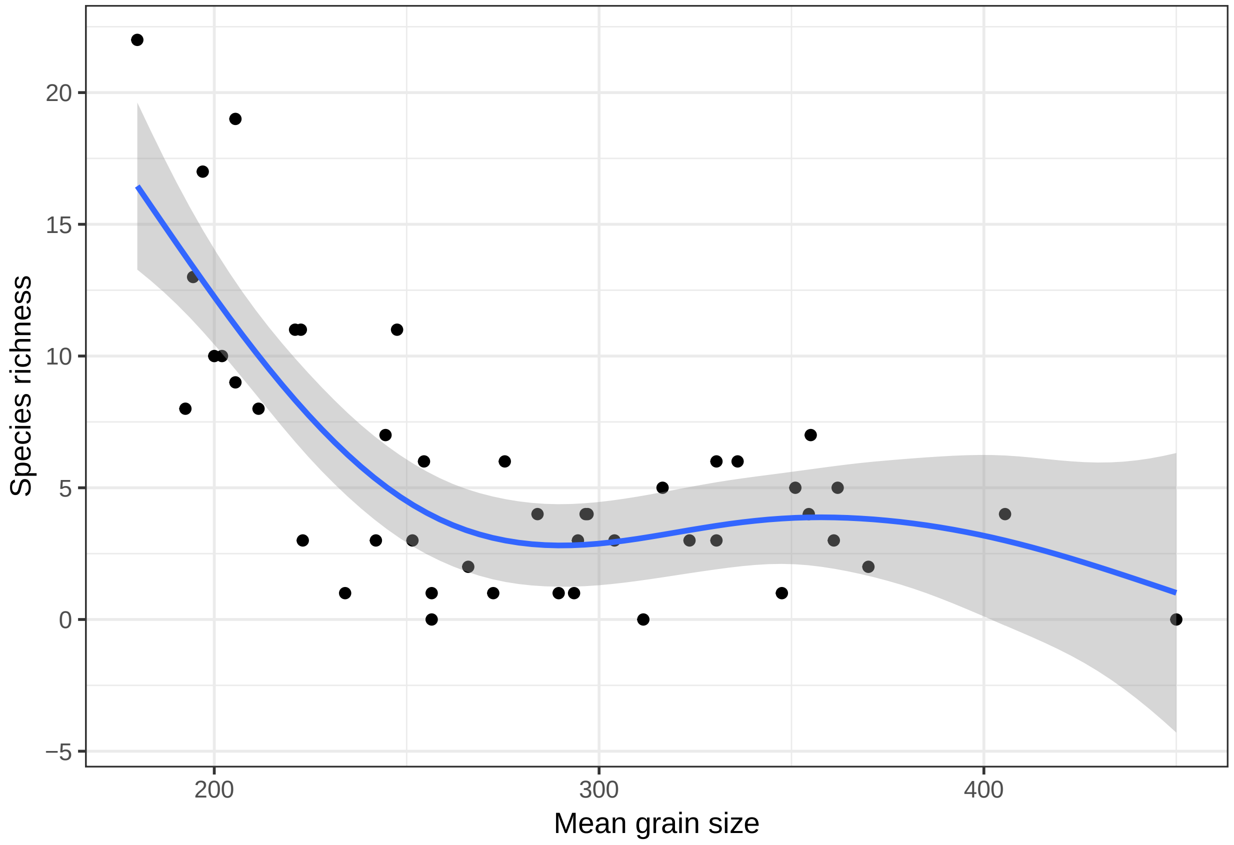
21.2 Summary model

```
summary(mod3)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## richness ~ s(grain)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.6889     0.4601   12.36  2.6e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F  p-value
## s(grain) 3.615  4.468 15.92 3.25e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.619   Deviance explained = 65.1%
## GCV = 10.616  Scale est. = 9.5269    n = 45
```

21.3 Plotting model

```
theme_set(theme_bw())
g0<-ggplot(d,aes(x=grain,y=richness))
g1<-g0+geom_point() + stat_smooth(method = "gam", formula = y ~ s(x))
g1 + xlab("Mean grain size") + ylab("Species richness")
```



Chapter 22

Non linear model

22.1 Rectangular hyperbola of the Michaelis-Menten form

$$C = \frac{sR}{F + R}$$

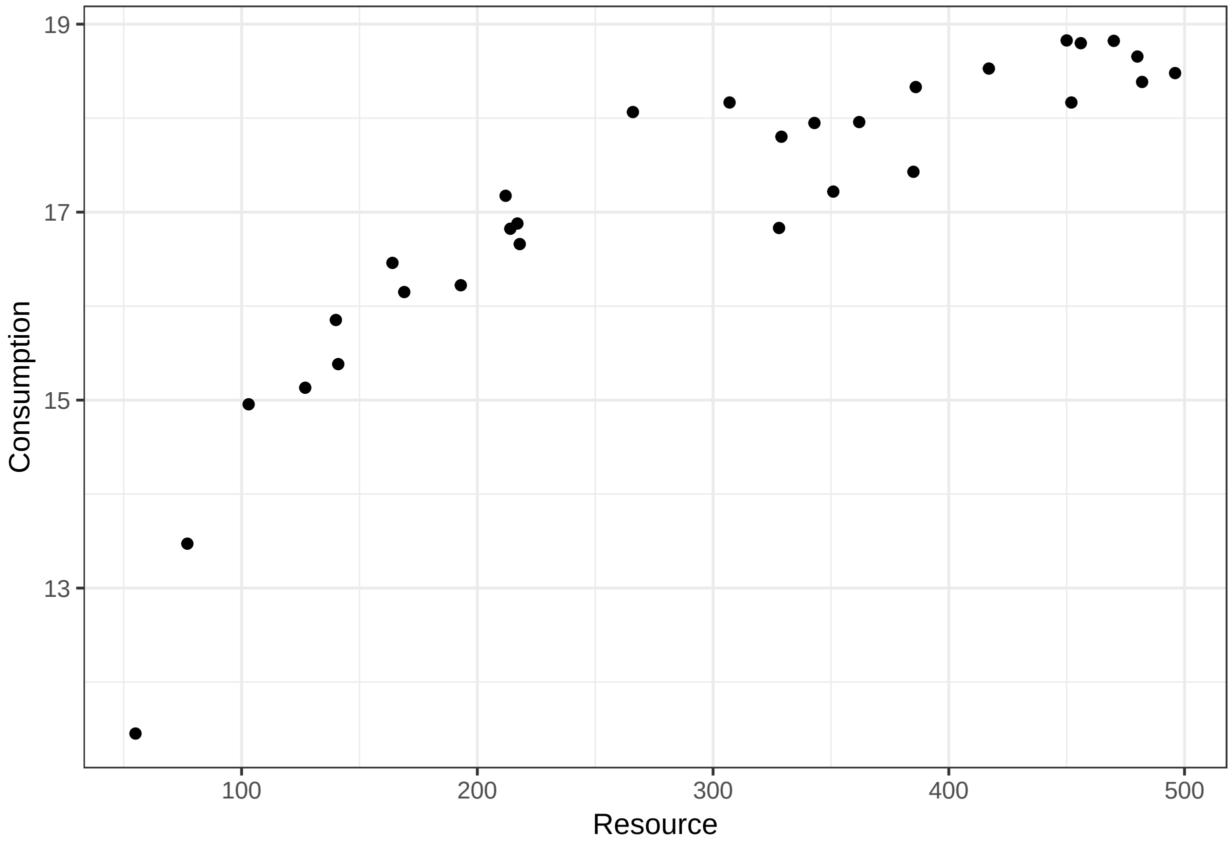
Where

- C is resource consumption,
- R is the amount or density of the resource,
- s is the asymptotic value and
- F represents the density of resource at which half the asymptotic consumption is expected to occur. This model is not linear in its parameters.

22.1.1 Fitting model

Starting values need to be provided. Plot the data first to estimate the asymptote

```
d<-read.csv("/home/aqm/course/data/Hollings.csv")
g0<-ggplot(data=d,aes(x=Resource,y=Consumption)) + geom_point()
g0
```

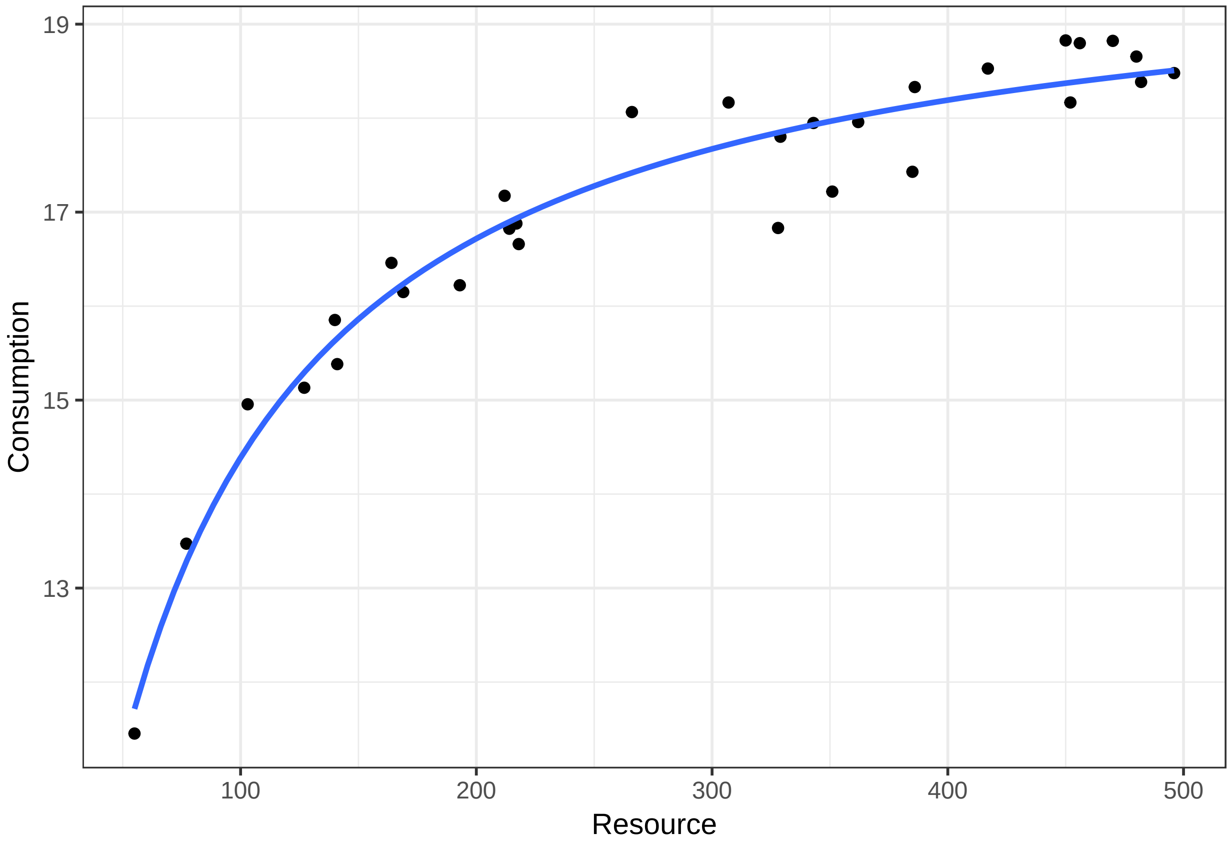


22.1.2 Fitting the model

```
nlmod<-nls(Consumption~s*Resource/(F+Resource),data = d,start = list( F = 20,s=20))
```

22.1.3 Curve fit

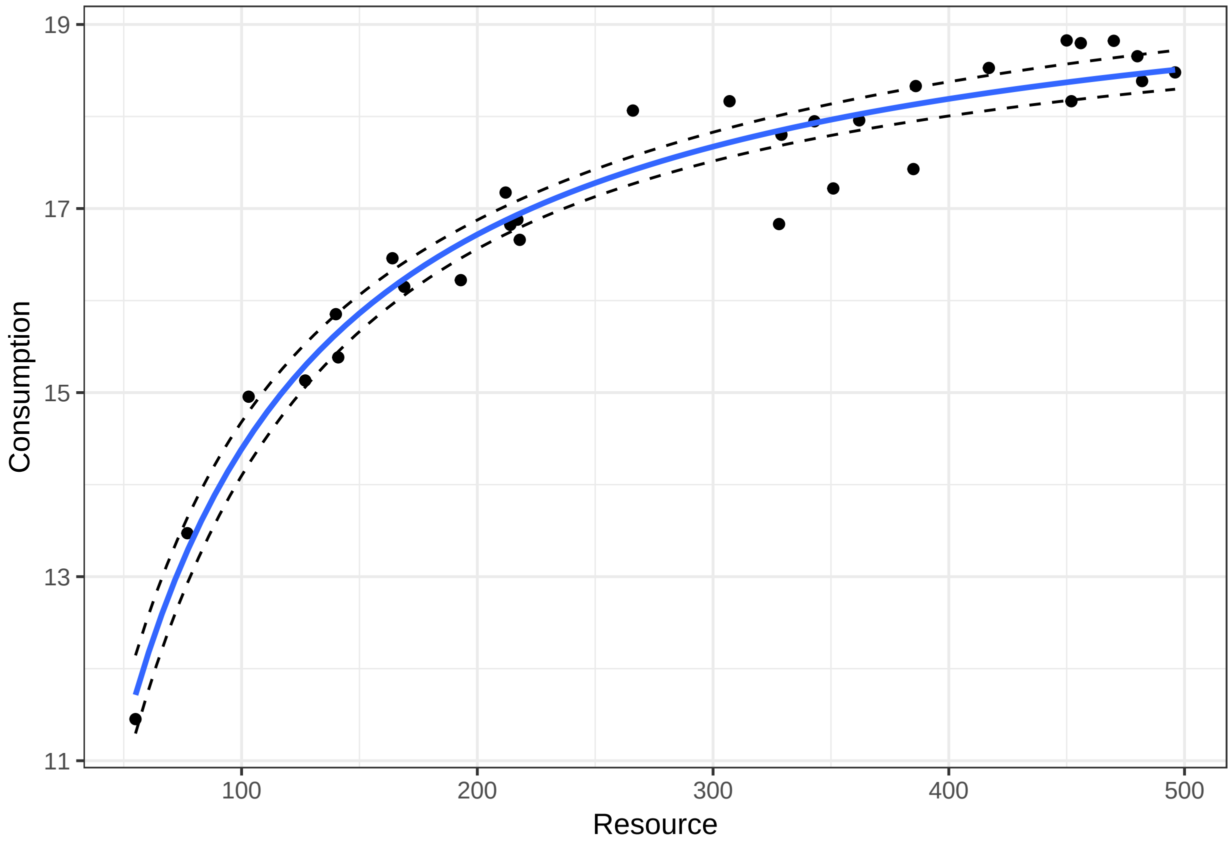
```
g0<-ggplot(data=d,aes(x=Resource,y=Consumption))
g1<-g0+geom_point()
g2<-g1+geom_smooth(method="nls",formula=y~s*x/(F+x),method.args=list(start = c( F = 20,s=20)), se=FALSE
g2
```



22.1.4 Curve fit with confidence intervals

This needs the propagate package.

```r
require(propagate)
newdata<-data.frame(Resource=seq(min(d$Resource),max(d$Resource),length=100))
pred_model <- predictNLS(nlmod, newdata=newdata,nsim = 10000)
conf_model <- pred_model$summary

newdata<-data.frame(newdata,conf_model)

g2  + geom_line(data=newdata,aes(x=Resource,y=Prop.2.5.),col="black",lty=2) + geom_line(data=newdata,ae
```



22.2 Holling's disc equation

The classic version of Hollings disk equation used in the article is written as

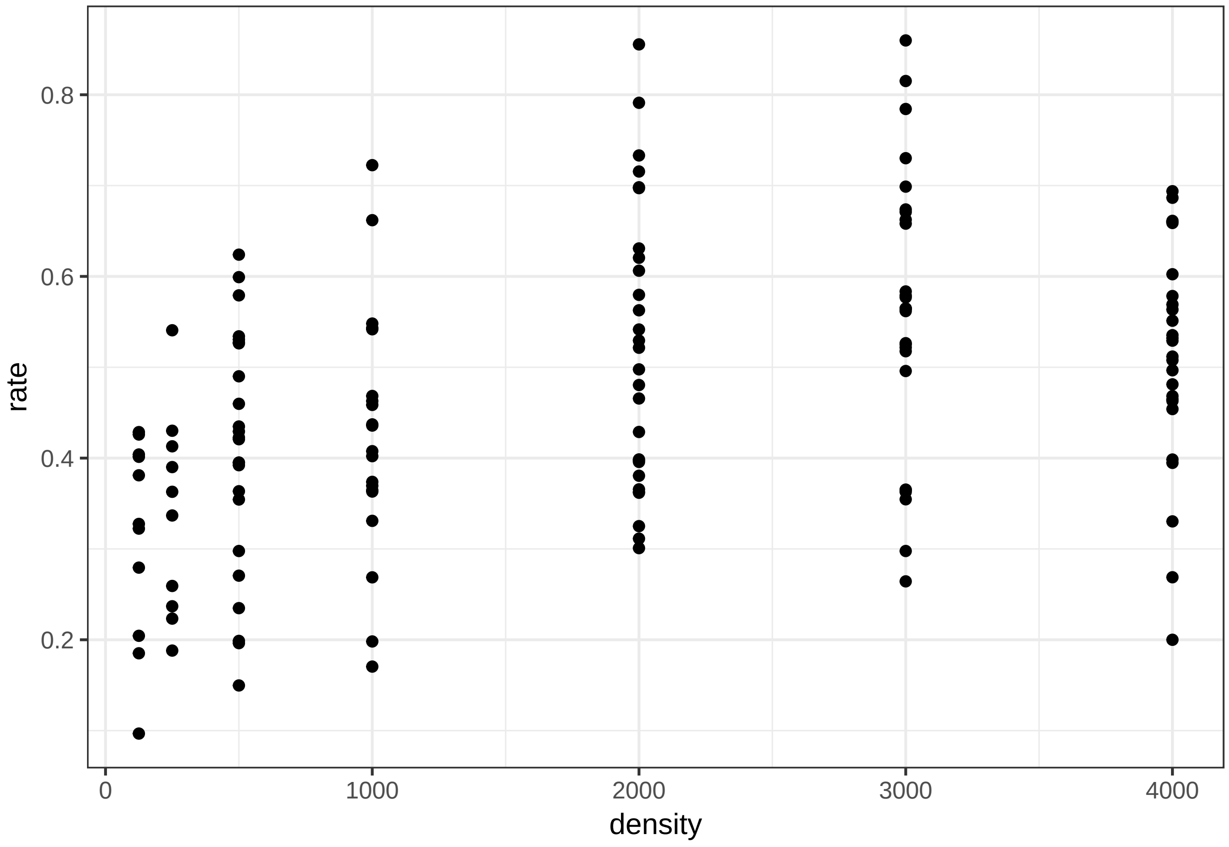
$R = \frac{aD}{1+aDH}$

Where

- F = feeding rate (food items)
- D = food density (food items m^{-2})
- a = searching rate ($m^2 s^{-1}$)
- H = handling time (s per food item).

22.2.1 The data

```
d<-read.csv("/home/aqm/course/data/buntings.csv")
g0<-ggplot(data=d,aes(x=density,y=rate)) + geom_point()
g0
```



22.2.2 Fitting the model

```
d<-read.csv("/home/aqm/course/data/buntings.csv")
HDmod<-nls(rate~a*density/(1+a*density*H),data = d,start = list(a =0.001,H=2))
```

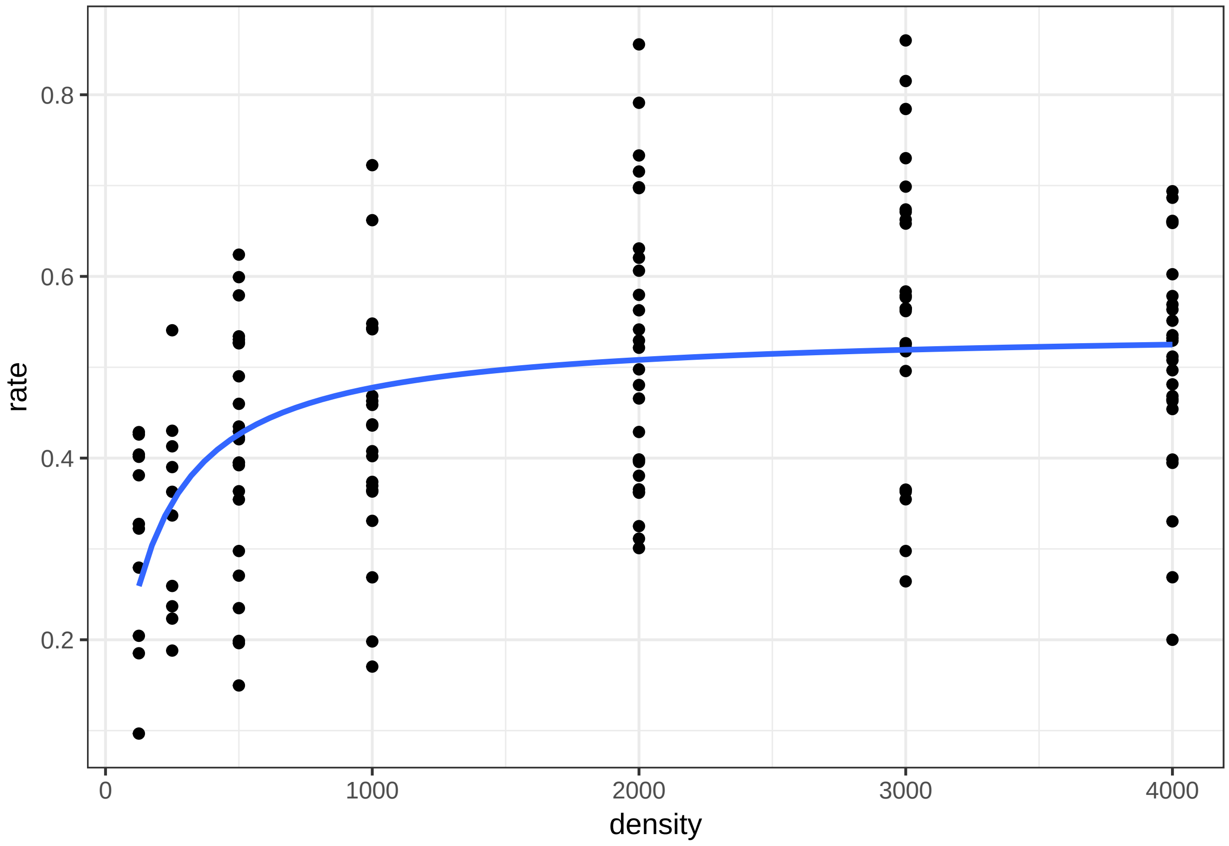
22.2.3 Confidence intervals for parameters

```
confint(HDmod)
```

```
##          2.5%       97.5%
## a 0.002593086 0.006694939
## H 1.713495694 1.976978655
```

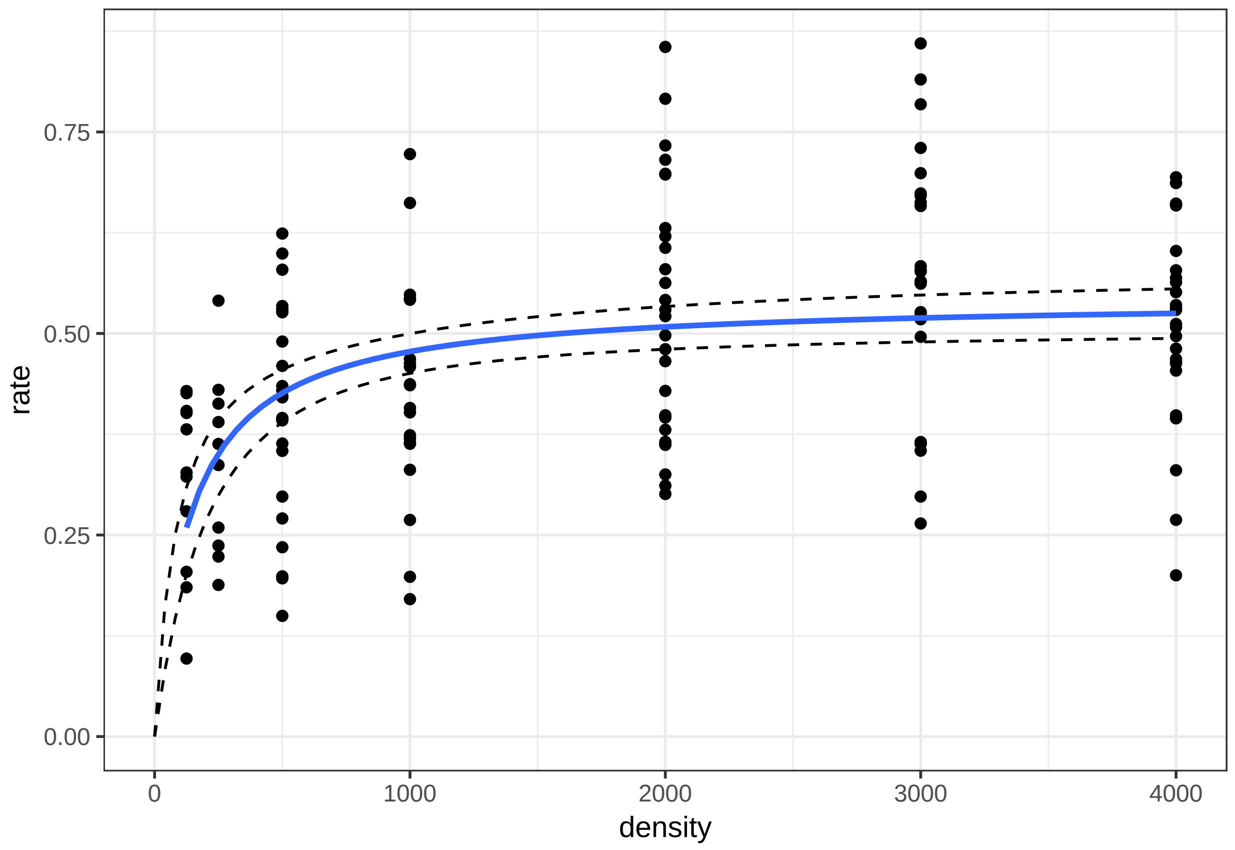
22.2.4 Plot with fitted curve

```
g0<-ggplot(data=d,aes(x=density,y=rate))
g1<-g0+geom_point()
g2<-g1+geom_smooth(method="nls",formula=y~a*x/(1+a*x*H),method.args=list(start = c(a = 0.01,H=2)), se=F
g2
```



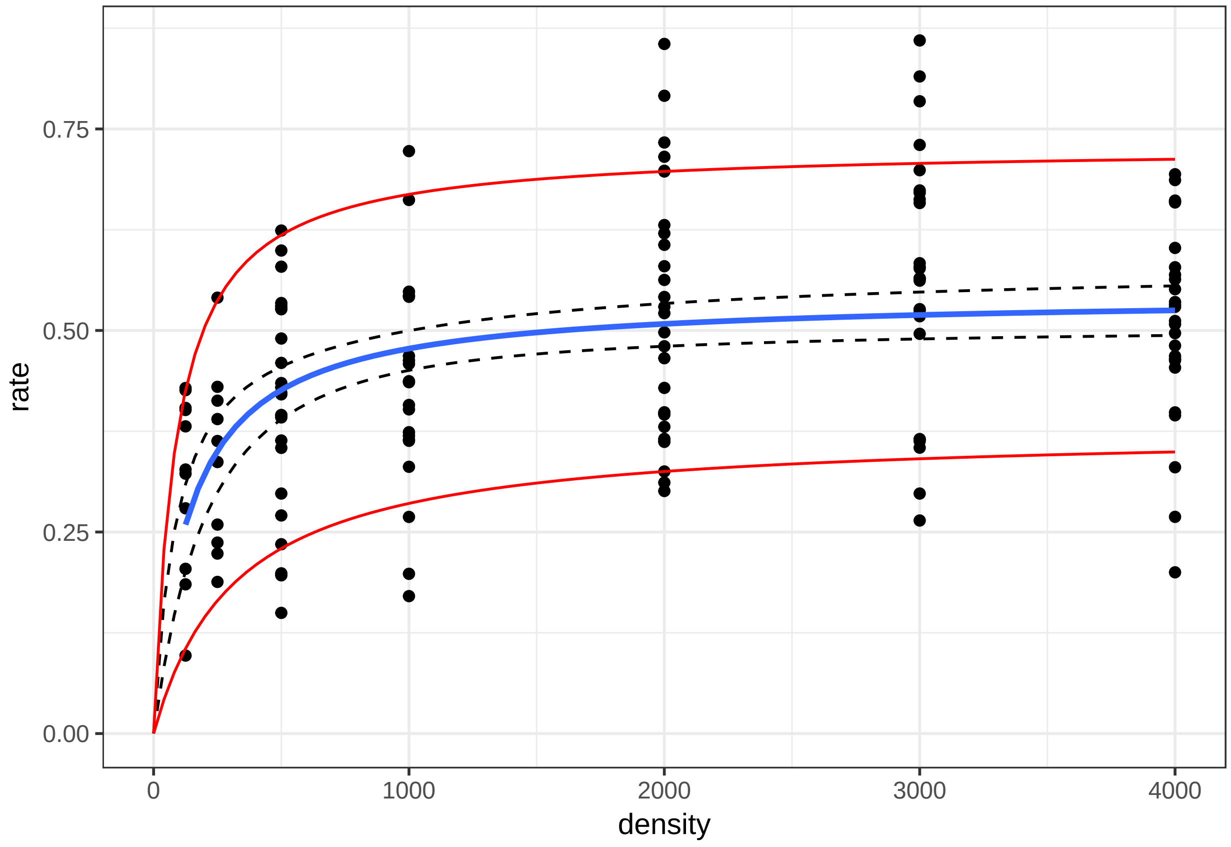
22.2.5 Plot with confidence intervals

```
require(propagate)
newdata<-data.frame(density=seq(0,max(d$density),length=100))
pred_model <- predictNLS(HDmod, newdata=newdata,nsim = 10000)
conf_model <- pred_model$summary

newdata<-data.frame(newdata,conf_model)

g3<-g2  + geom_line(data=newdata,aes(x=density,y=Prop.2.5.),col="black",lty=2) + geom_line(data=newdata
g3
```



22.2.6 Non linear quantile regression

```
library(quantreg)
QuantMod90<-nlrq(rate~a*density/(1+a*density*H),data = d,start = list(a =0.001,H=2),tau=0.9)
QuantMod10<-nlrq(rate~a*density/(1+a*density*H),data = d,start = list(a =0.001,H=2),tau=0.1)
newdata$Q90<- predict(QuantMod90, newdata = newdata)
newdata$Q10 <- predict(QuantMod10, newdata = newdata)

g3 + geom_line(data=newdata,aes(x=density,y=Q90),col="red") + geom_line(data=newdata,aes(x=density,y=Q1
```


rate
0.00
0.25
0.50
0.75
0
1000
2000
3000
4000
density

Chapter 23

Analysis of covariance and mixed effects crib

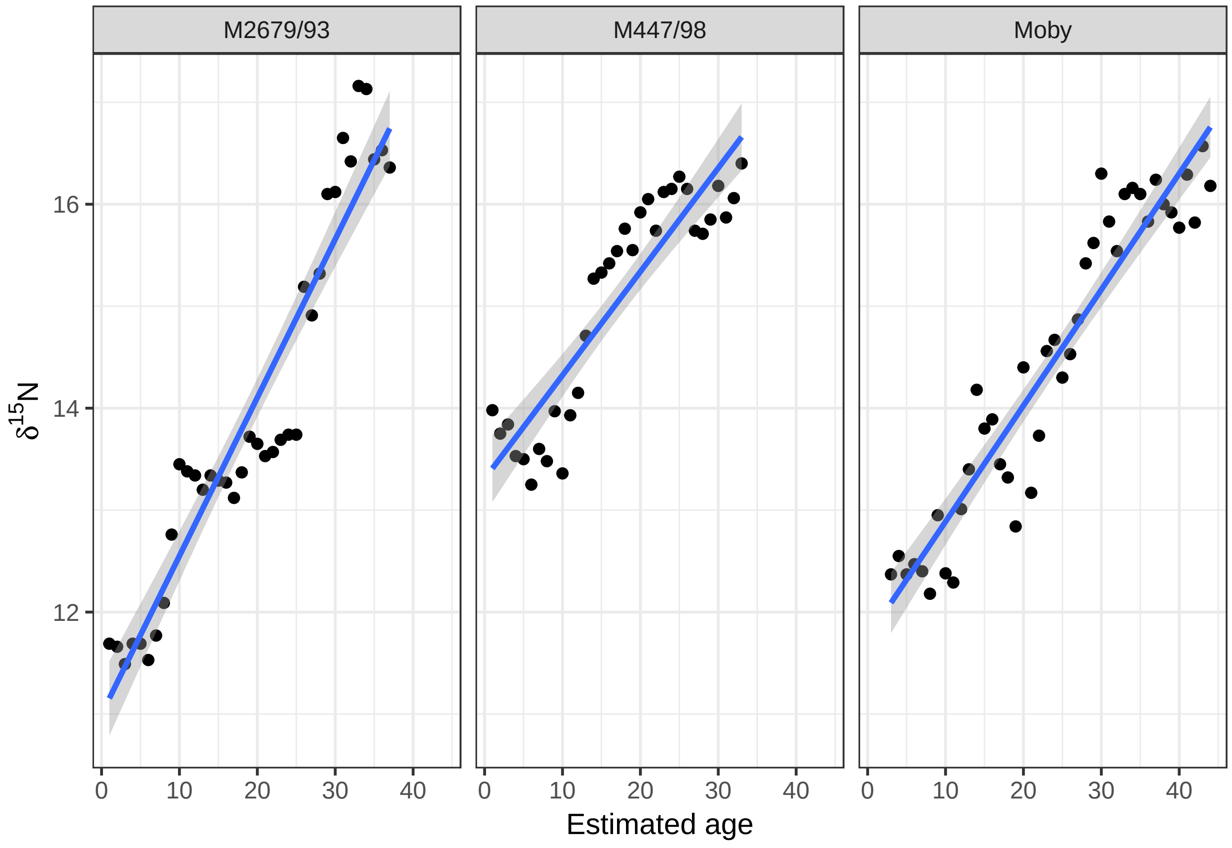
23.1 Data

Fixed effects analysis of covariance is useful when there are few groups. With many groups the groups of observations may be treated as a random effect. In this case the grouping variable is the individual whale.

```
Whales<-read.csv("https://tinyurl.com/aqm-data/whaleteeth.csv")
ylabel <-expression(paste(delta^{15}, "N"))
xlabel<-"Estimated age"
## Select just three whales
Whales %>% filter(Whale %in% levels(Whales$Whale)[c(7,9,11)]) -> Whales3
```

23.1.1 Plot the patterns grouped by individual

```
library(ggplot2)
theme_set(theme_bw())
g0<-ggplot(data=Whales3,aes(x=Age,y=X15N))
g1<-g0+geom_point()+ labs(y = ylabel,x=xlabel)
g1+facet_wrap("Whale") +geom_smooth(method="lm")
```



23.1.2 Analysis of covariance

A significant interaction shows that the slope of a linea model differs between individuals

```r
mod1 <-lm(data=Whales3,X15N~Age+Whale)
mod2 <-lm(data=Whales3,X15N~Age*Whale)
anova(mod1,mod2)
```

```
## Analysis of Variance Table
##
## Model 1: X15N ~ Age + Whale
## Model 2: X15N ~ Age * Whale
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    108 33.686
## 2    106 27.419  2    6.2671 12.114 1.827e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

23.2 Linear mixed effects

23.2.1 Intercept only

```r
library(lmerTest)
intercept.mod<-lmer(X15N~Age+(1|Whale),data=Whales)
intercept.mod
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: X15N ~ Age + (1 | Whale)
##    Data: Whales
## REML criterion at convergence: 784.3965
## Random effects:
##  Groups   Name        Std.Dev.
##  Whale    (Intercept) 0.6139
##  Residual             0.8149
## Number of obs: 307, groups:  Whale, 11
## Fixed Effects:
## (Intercept)          Age
##    12.25874      0.09245
```

```
anova(intercept.mod)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##     Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## Age 216.11  216.11     1 301.39  325.46 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

23.2.2 Slope model

```
slope.mod<-lmer(X15N~Age+(Age|Whale),data=Whales)
slope.mod
```

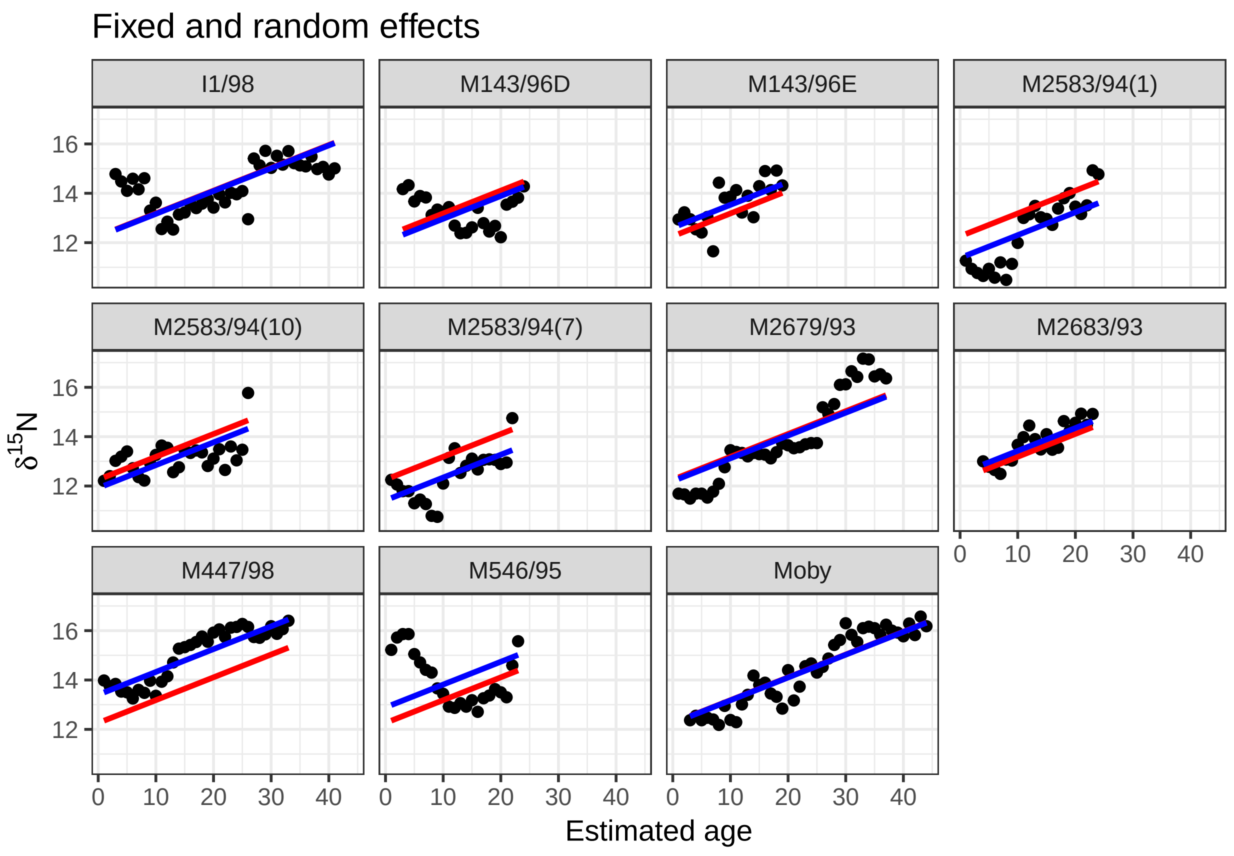
```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: X15N ~ Age + (Age | Whale)
##    Data: Whales
## REML criterion at convergence: 657.3864
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  Whale    (Intercept) 1.3138
##           Age         0.0734   -0.90
##  Residual             0.6246
## Number of obs: 307, groups:  Whale, 11
## Fixed Effects:
## (Intercept)          Age
##    12.40999      0.07915
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```
anova(slope.mod)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##     Sum Sq Mean Sq NumDF  DenDF F value   Pr(>F)
## Age 4.7217  4.7217     1 10.212  12.102 0.005751 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

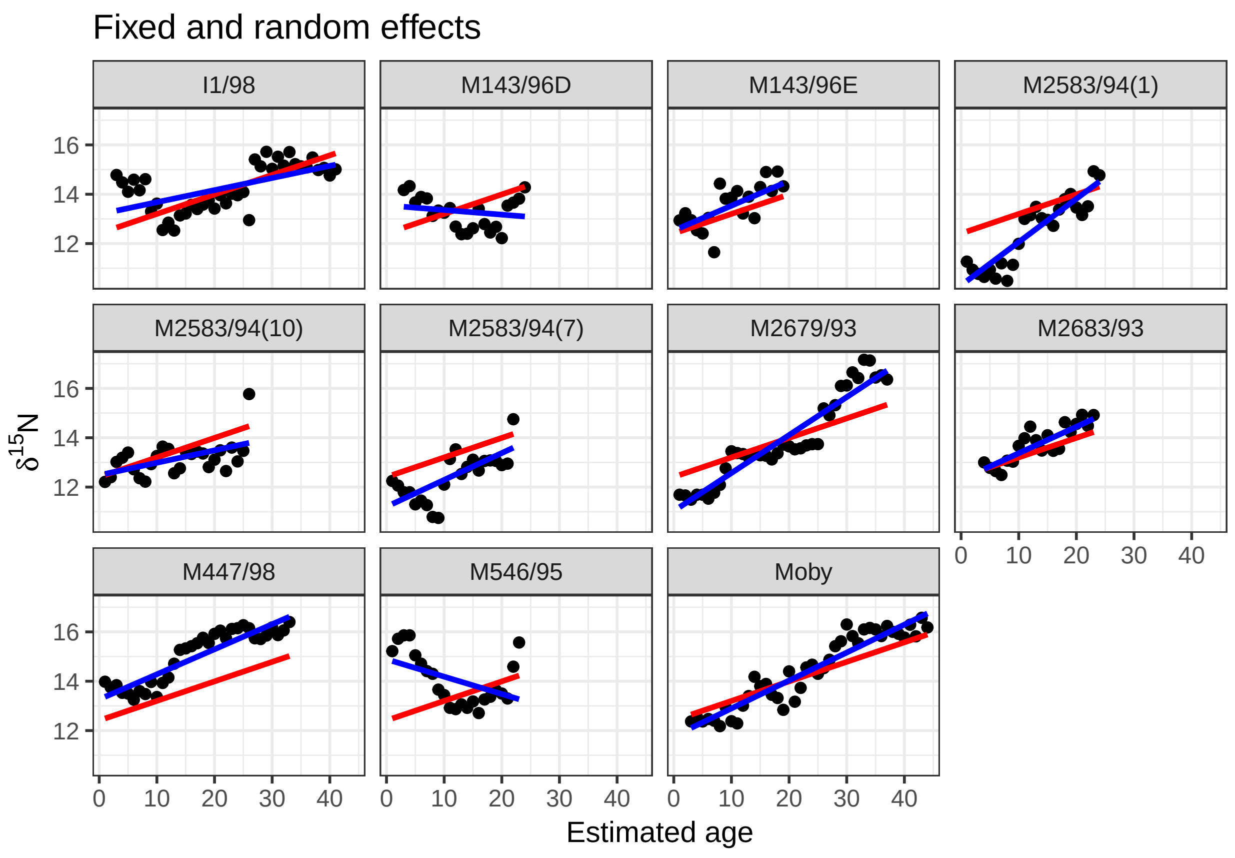
23.2.3 Plot intercept model

```
Whales$fixed<-predict(intercept.mod,re.form=NA)
Whales$rand<-predict(intercept.mod)
g0<-ggplot(Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=2,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=rand),colour=4,lwd=1)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed and random effects")
g1+facet_wrap("Whale")
```



23.2.4 Plot slope model

```
Whales$fixed<-predict(slope.mod,re.form=NA)
Whales$rand<-predict(slope.mod)
g0<-ggplot(Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=2,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=rand),colour=4,lwd=1)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed and random effects")
g1+facet_wrap("Whale")
```



23.3 GAMM model

23.3.1 Fit GAMM model

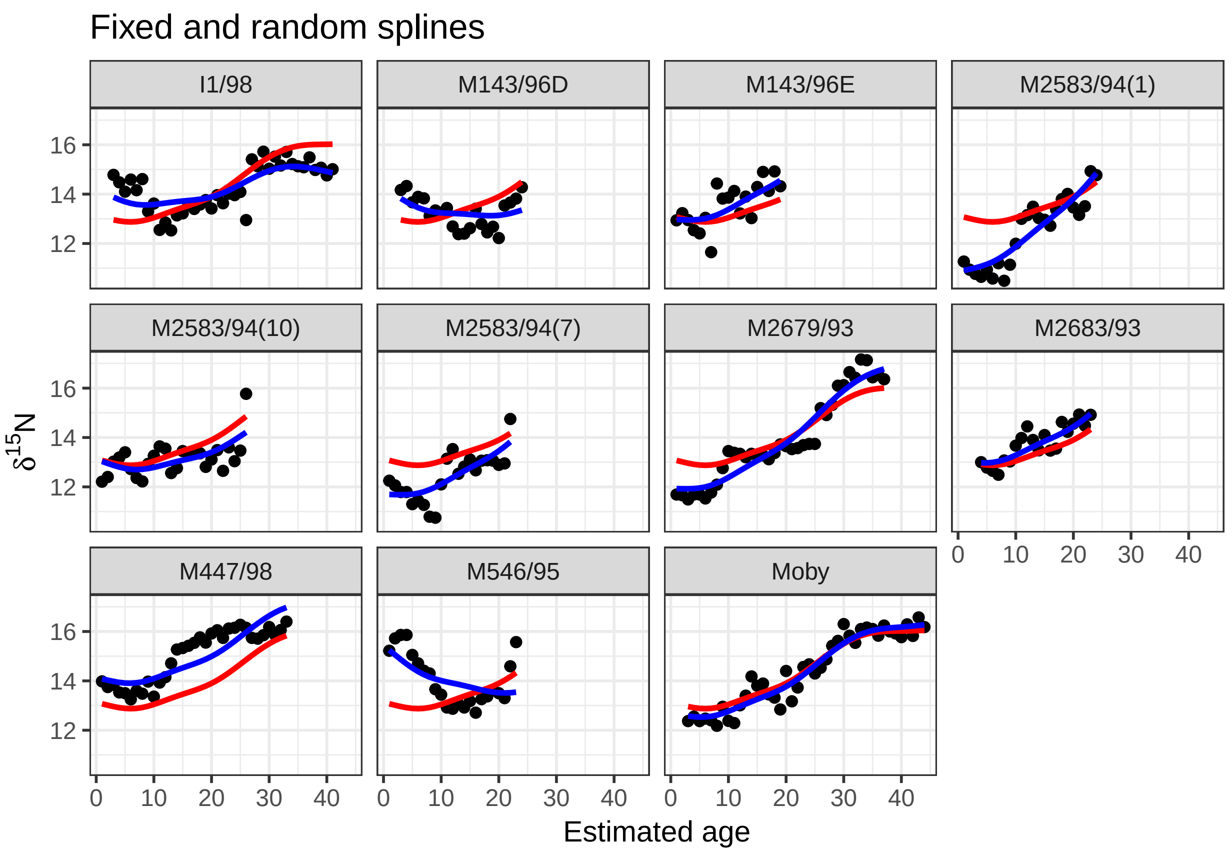
```
library(gamm4)
gamm.mod<-gamm4(X15N~s(Age),data=Whales,random = ~ (Age|Whale))
gamm.mod
```

```
## $mer
## Linear mixed model fit by REML ['lmerMod']
## REML criterion at convergence: 619.3208
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  Whale    (Intercept) 1.34561
##           Age         0.07326  -0.91
##  Xr       s(Age)      2.55534
##  Residual             0.57809
## Number of obs: 307, groups:  Whale, 11; Xr, 8
## Fixed Effects:
## X(Intercept)     Xs(Age)Fx1
##      13.8106        -0.2491
##
## $gam
##
```

```
## Family: gaussian
## Link function: identity
##
## Formula:
## X15N ~ s(Age)
##
## Estimated degrees of freedom:
## 5.56  total = 6.56
##
## lmer.REML score: 619.3208
```

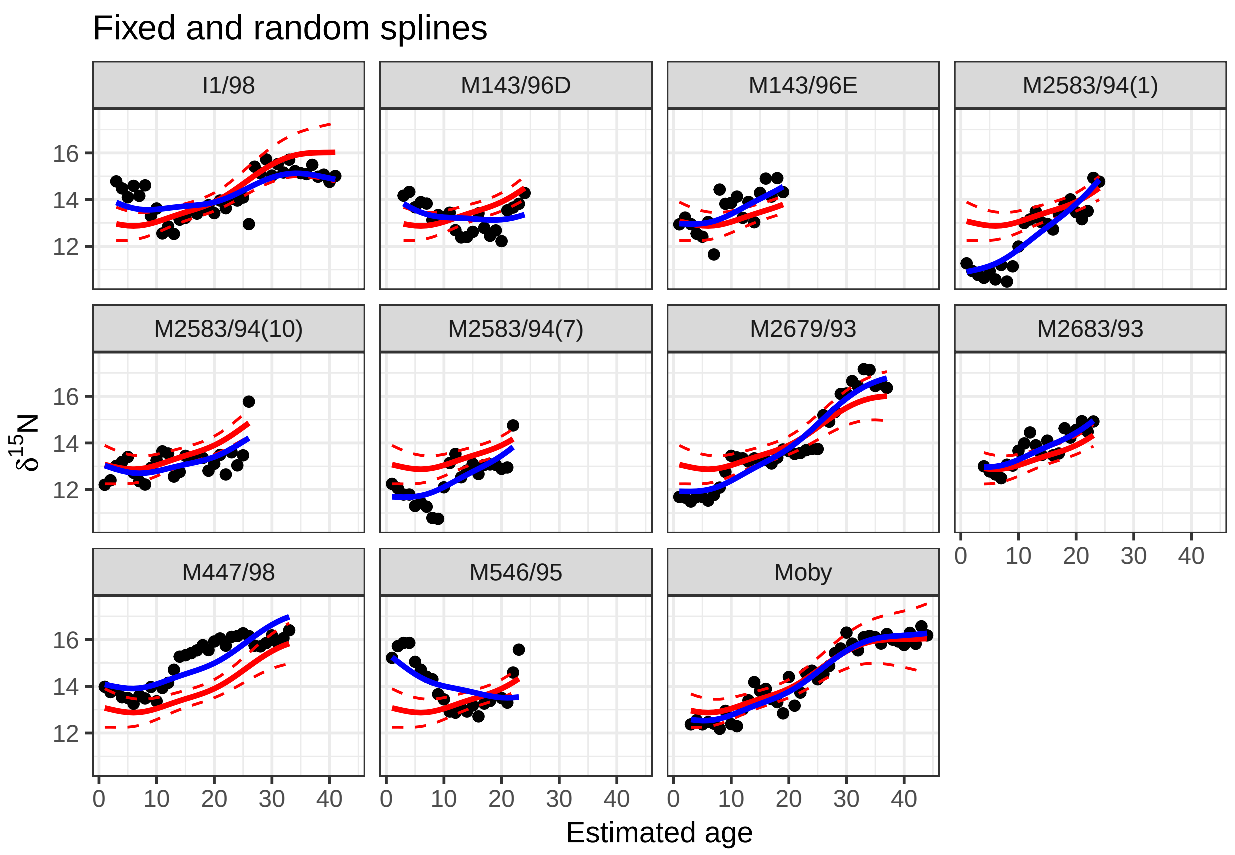
23.3.2 Plot GAMM model

```
Whales$fixed<-predict(gamm.mod$gam)
Whales$rand<-predict(gamm.mod$mer)
g0<-ggplot(Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=2,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=rand),colour=4,lwd=1)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed and random splines")
g1+facet_wrap("Whale")
```



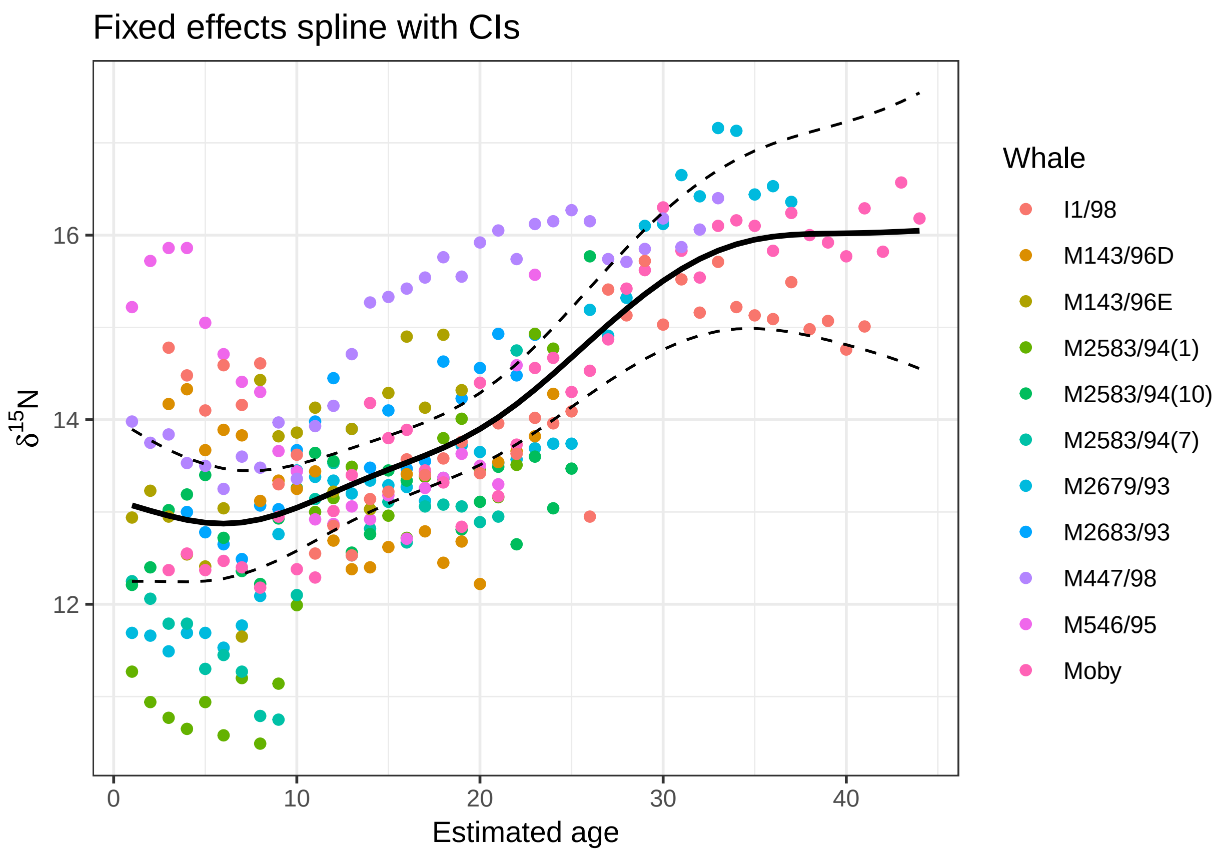
23.3.3 Plot GAMM with CIs

```
Whales$fixedse<-predict(gamm.mod$gam,se=T)$se
g0<-ggplot(Whales,aes(x=Age,y=X15N))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=2,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=fixed+2*fixedse),colour=2,lty=2)
g1<-g1+geom_line(aes(x=Age,y=fixed-2*fixedse),colour=2,lty=2)
g1<-g1+geom_line(aes(x=Age,y=rand),colour=4,lwd=1)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed and random splines")
g1+facet_wrap("Whale")
```



23.3.4 Plot fixed effects with CIS taking into account random effects

```
g0<-ggplot(Whales,aes(x=Age,y=X15N,color=Whale))
g1<-g0+geom_point()
g1<-g1+geom_line(aes(x=Age,y=fixed),colour=1,lwd=1)
g1<-g1+geom_line(aes(x=Age,y=fixed+2*fixedse),colour=1,lty=2)
g1<-g1+geom_line(aes(x=Age,y=fixed-2*fixedse),colour=1,lty=2)
g1<-g1+labs(y = ylabel,x=xlabel,title="Fixed effects spline with CIs")
g1
```


Fixed effects spline with CIs
$\delta^{15}N$
12
14
16
0
10
20
30
40
Estimated age
Whale
I1/98
M143/96D
M143/96E
M2583/94(1)
M2583/94(10)
M2583/94(7)
M2679/93
M2683/93
M447/98
M546/95
Moby

Chapter 24

Bubble plot crib sheet

Hans Roslin, who died in 2017, was considered to be one of the greatest data communicator of all time. Roslin's skill was an ability to select informative ways of displaying large data sets in order to engage the audience. There are many videos of his talks on You Tube and Ted talks.

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

Roslin's original figures were constructed by a team of data analysts, programmers and analysts working with the professor. The later figures that he used look as if they probably were first built up using Tableau and then touched up and animated by the team. Can these sophisticated figures be built easily in R? The answer is that it is surprisingly simple. The figures require very few lines of code.

24.1 Getting the data

A suitable data set is provided by the WDI package which can search, extract and format data from the World Bank's World Development Indicators.

```
library(WDI)
library(dplyr)
library(ggplot2)
library(plotly)
library(ggthemes)
```

The field names and their meanings can be searched for in the table below.

```
dd<-data.frame(WDIsearch( field='name', cache=NULL) )
DT::datatable(dd)
```

Show 10 entries Search:

	indicator	name
1	BX.TRF.PWKR.GD.ZS	Workers' remittances, receipts (% of GDP)
2	BX.TRF.PWKR.DT.GD.ZS	Personal remittances, received (% of GDP)
3	BX.TRF.MGR.DT.GD.ZS	Migrant remittance inflows (% of GDP)
4	BX.KLT.DINV.WD.GD.ZS	Foreign direct investment, net inflows (% of GDP)
5	BX.KLT.DINV.DT.GD.ZS	Foreign direct investment, net inflows (% of GDP)
6	BX.GSR.MRCH.ZS	Merchandise exports (BOP): percentage of GDP (%)
7	6.0.GDPpc_constant	GDP per capita, PPP (constant 2011 international $)
8	6.0.GDP_usd	GDP (constant 2005 $)
9	6.0.GDP_growth	GDP growth (annual %)
10	6.0.GDP_current	GDP (current $)

Showing 1 to 10 of 493 entries Previous 1 2 3 4 5 … 50 Next

24.2 Selecting some indicators

All the indicators are not available for all the years, but a useful set can be pulled from the data base. The data includes aggregates (regions and global figures) than can be filtered out to leave just the countries.

```
d <- WDI(country="all", indicator=c("NY.GDP.PCAP.CD", "SP.POP.TOTL", "SP.DYN.LE00.IN","EG.EGY.PRIM.PP.KD
names(d)[4:8]<- c("GDP_per_capita", "Population_total", "Life_expectancy","Energy_intensity_MJ_GDP","Agr
d %>% filter(region !="Aggregates") -> d ## Filter out the aggregated totals
```

24.3 Forming bubble plots

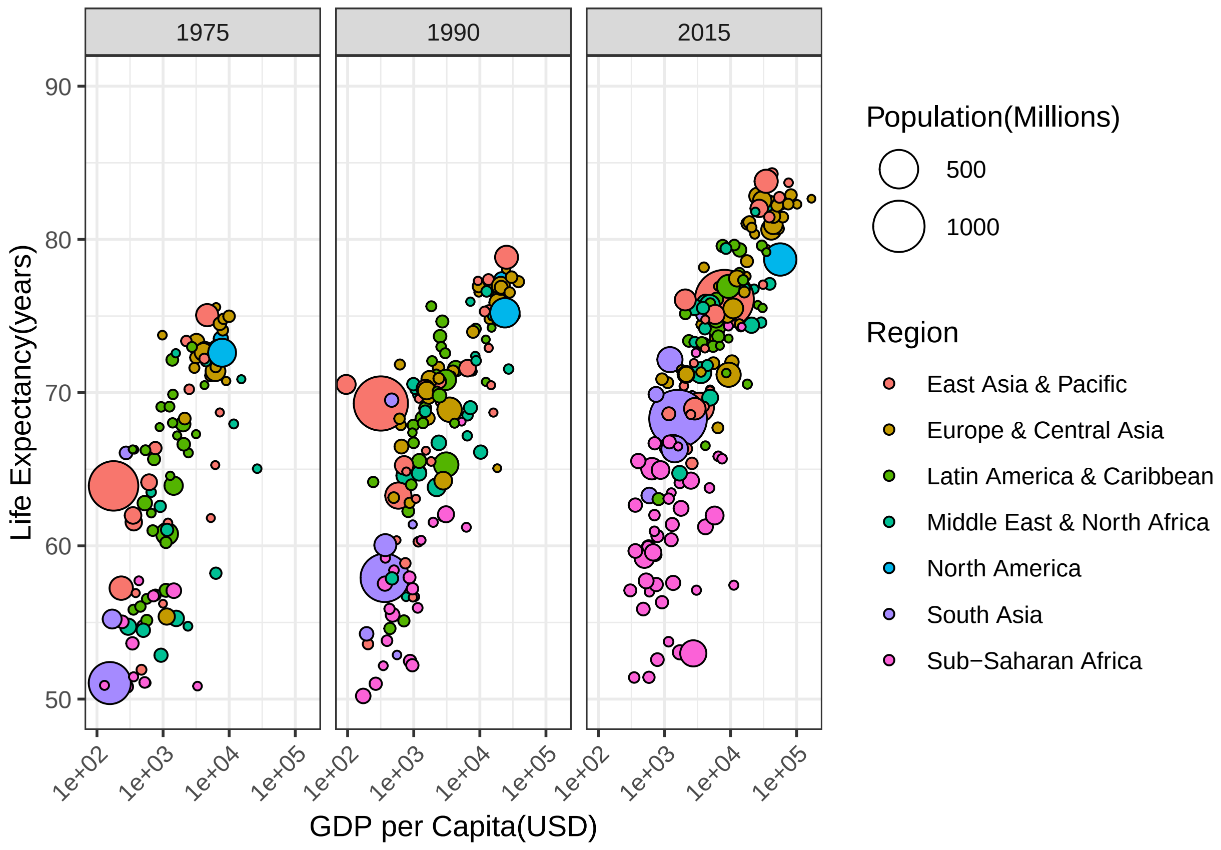
Han's Roslin's data animations effectively showed four dimensions of data in one. A third dimension was added to the scatter-plots in the form of the size of the bubble. In most of the plots this was the population size. The animation was over time, adding a fourth dimension. In order to produce a static graph a smaller subset of the years can be shown side by side.

So the tricks used in the R code are to add in a size aesthetic to represent population and to facet wrap on year. Some additional information can be added by colouring the points according to region. If a filled point style (21) is chosen the fill can be set as an aesthetic. Some tweaking of axes is necessary. A log scale on the x axis spreads out the points more effectively and the y scale can be constrained to a range to avoid one or two outlying points adding too much space.

So the tricks are

1. aes(x = GDP_per_capita, y = Life_expectancy, size = Population_total/1000000). Note that the label aesthetic is not used for a static graph as it would lead to too much clutter, but it is useful when plotly is used.
2. scale_x_log10()
3. scale_y_continuous(limits = c(50, 90))
4. facet_wrap("year")

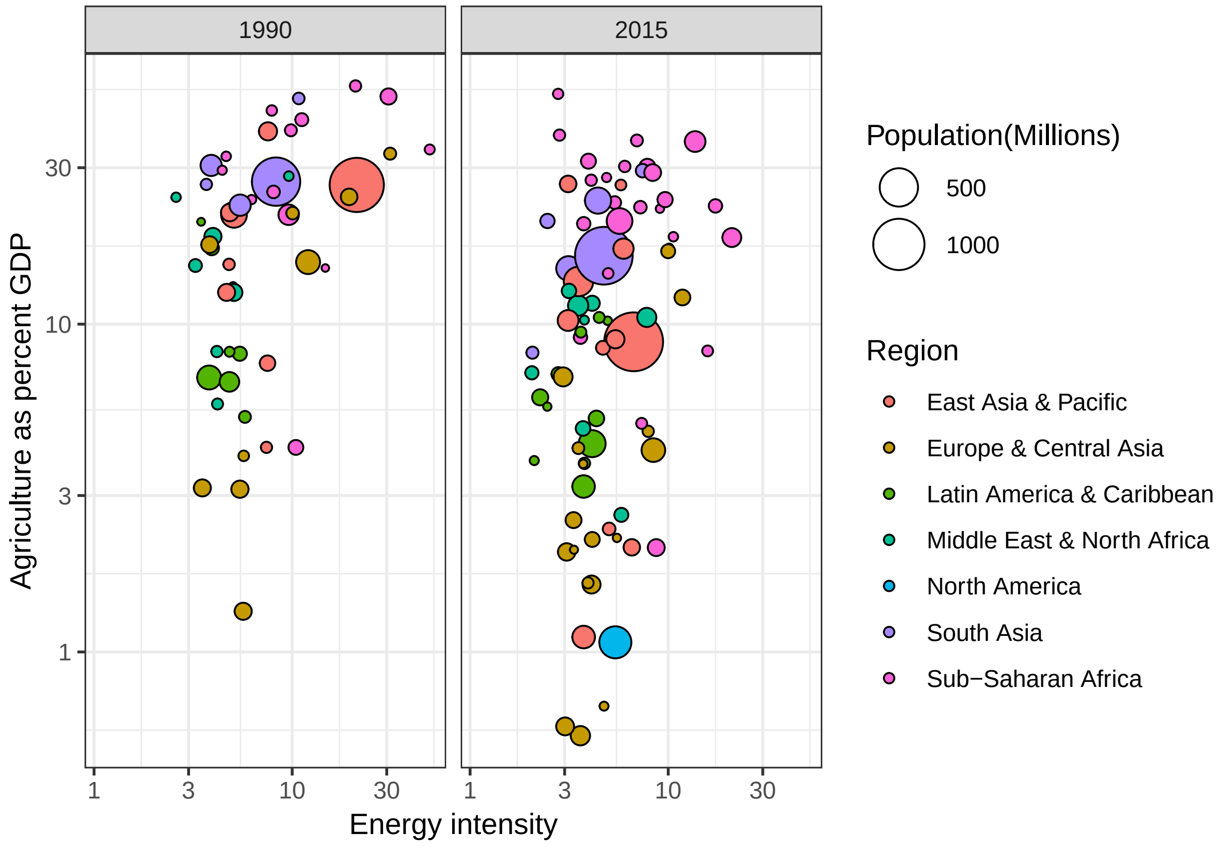
```
d %>% filter(year%in% c(1975,1990,2015)) %>% ggplot( aes(x = GDP_per_capita, y = Life_expectancy, size =
  geom_point(shape = 21) +
  labs(x = "GDP per Capita(USD)", y = "Life Expectancy(years)") +
  scale_y_continuous(limits = c(50, 90)) +
  scale_size(range = c(1, 10)) +
  labs(size = "Population(Millions)", fill = "Region") + scale_x_log10() + theme_bw() +facet_wrap("year
g1
```



Notice how sub-saharan Africa has been left behind in the general increase in life expectancy. The positions of China and India are also very striking. Looking at individual countries trajectories is possible using ggplotly.

24.4 Another example

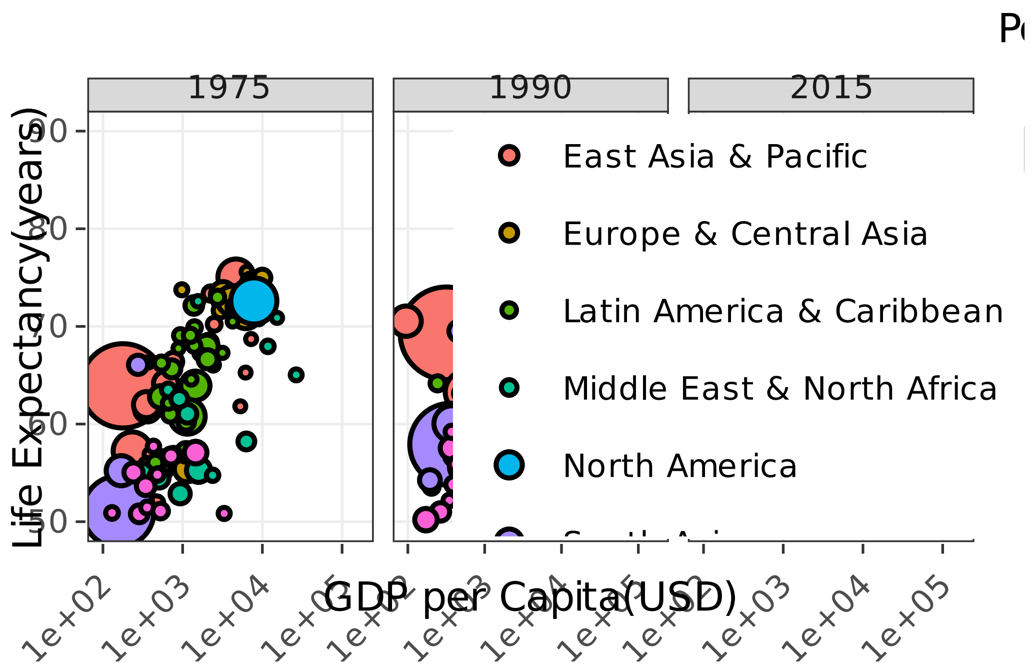
```
d %>% filter(year%in% c(1990,2015)) %>% filter(Population_total> 10000000) %>% ggplot( aes(x = Energy_i
  geom_point(shape = 21) +
  labs(x = "Energy intensity", y = "Agriculture as percent GDP") +
  scale_size(range = c(1, 10)) + scale_y_continuous(limits = c(1, 50)) +
  labs(size = "Population(Millions)", fill = "Region")  + theme_bw() +facet_wrap("year") + scale_x_log1
g2
```



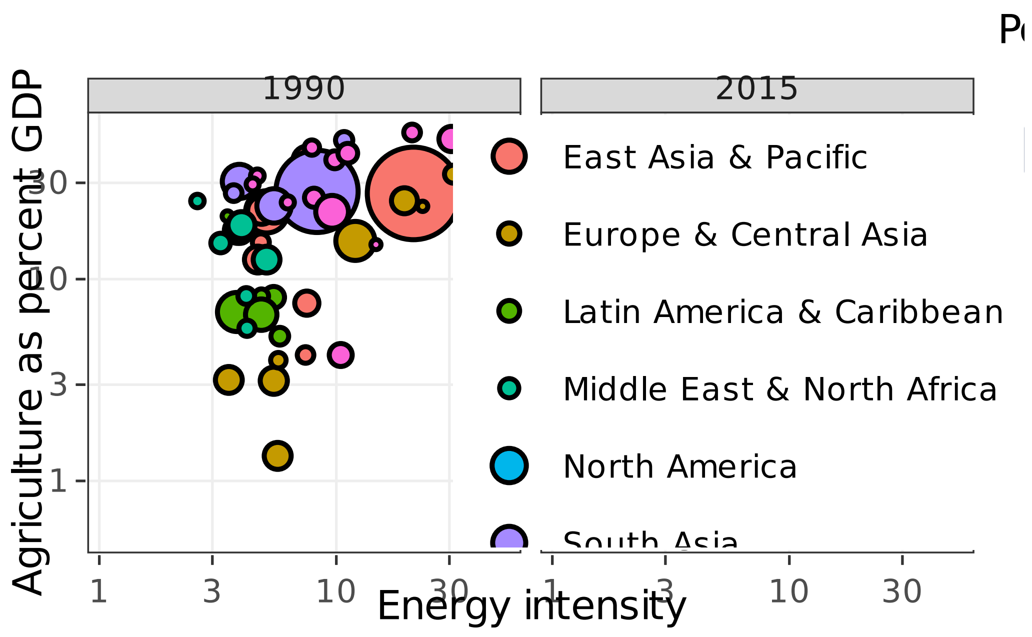
24.5 Using ggplotly

A problem with the completely static figures is that countries cannot be identified, although it is easy to deduce the identities of those with large populations such as India, China and the USA. Using ggplotly resolves this, as the label aesthetic can be seen when the mouse hovers over the country.

```r
ggplotly(g1)
```



```
ggplotly(g2)
```



The papers referred to here are all available on the server

http://r.bournemouth.ac.uk:82/AQM/AQM_2018/articles/

Bibliography

Gigerenzer, G. (1998). Surrogates for theories. *THEORY & PSYCHOLOGY*, 8(2):195–204.

Goodman, S. (2008). A dirty dozen: Twelve p-value misconceptions. *Seminars in Hematology*, 45(3):135–140.

Hobbs, N. T. and Hilborn, R. (2006). Alternatives to statistical hypothesis testing in ecology: A guide to self teaching. *ECOLOGICAL APPLICATIONS*, 16(1):5–19.

Johnson, D. H. (1999). The insignificance of statistical significance testing. *The Journal of Wildlife Management*, 63(3):763.

Nickerson, R. S., Baron, J., Chechile, R., and Es, W. (2000). Null hypothesis significance testing : A review of an old and continuing controversy. 5(2):241–301.

Steel, E. A., Kennedy, M. C., Cunningham, P. G., and Stanovick, J. S. (2013). Applied statistics in ecology: Common pitfalls and simple solutions. *Ecosphere*, 4(9):1–13.

Zuur, A. F., Ieno, E. N., and Elphick, C. S. (2010). A protocol for data exploration to avoid common statistical problems. *Methods in Ecology and Evolution*, 1(1):3–14.