Crib sheets

Duncan Golicher

2019-03-29

Contents

Chapter 1

How to use the 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.

Chapter 2

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.

2.1 Un-paired T-test

2.1.1 Data formats

2.1.1.1 Long format

```
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

2.1.1.2 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)
dt(d2)
```

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

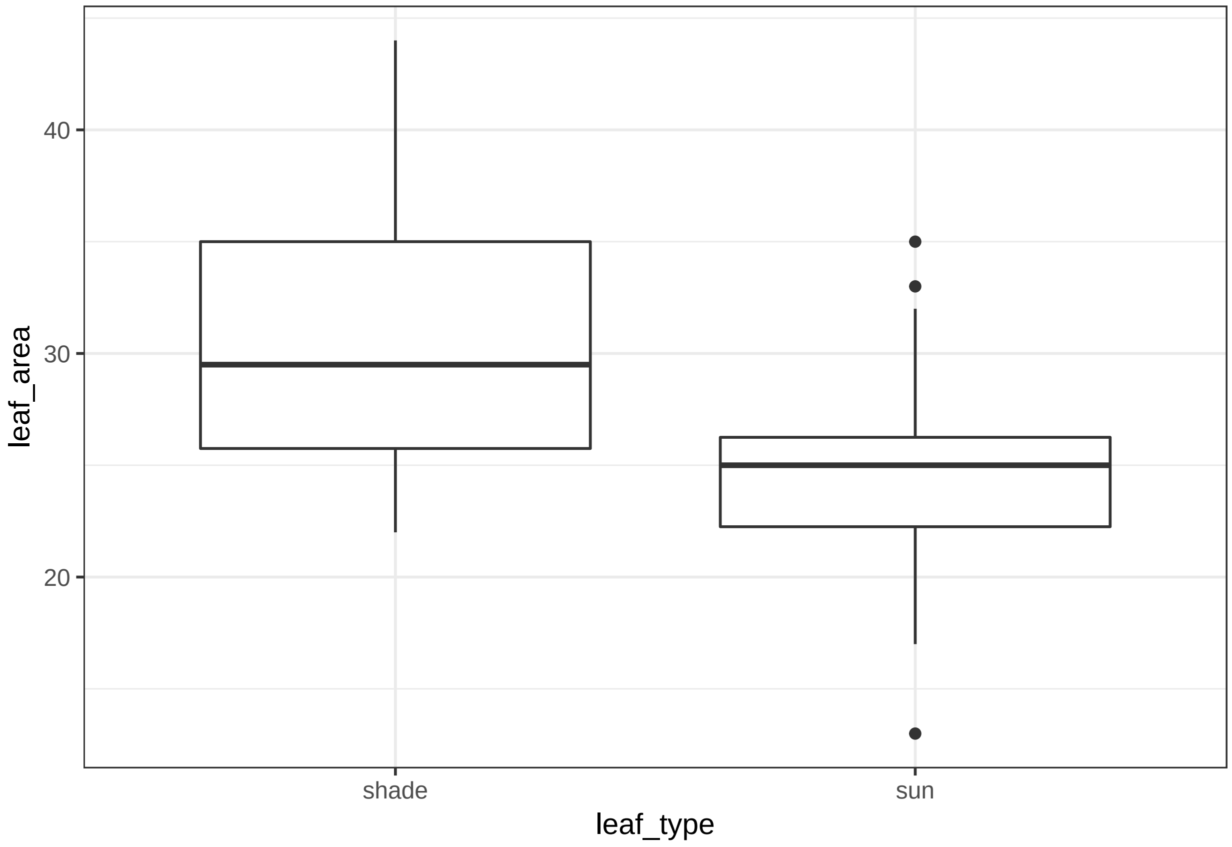
Showing 1 to 10 of 40 entries Previous 1 2 3 4 Next

```
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
```

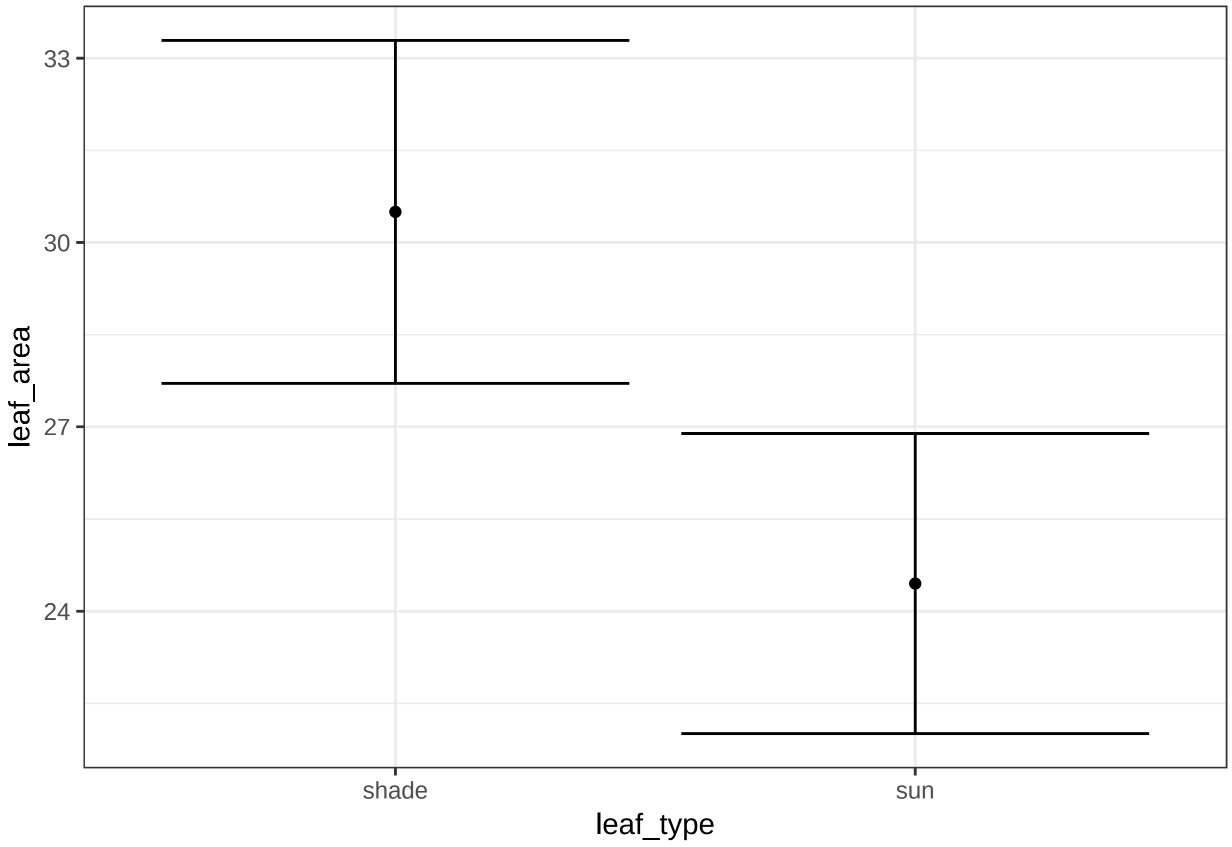
2.1.2 Boxplot

```
g0<-ggplot(d,aes(x=leaf_type,y=leaf_area))
g_box<- g0 +geom_boxplot()
g_box
```



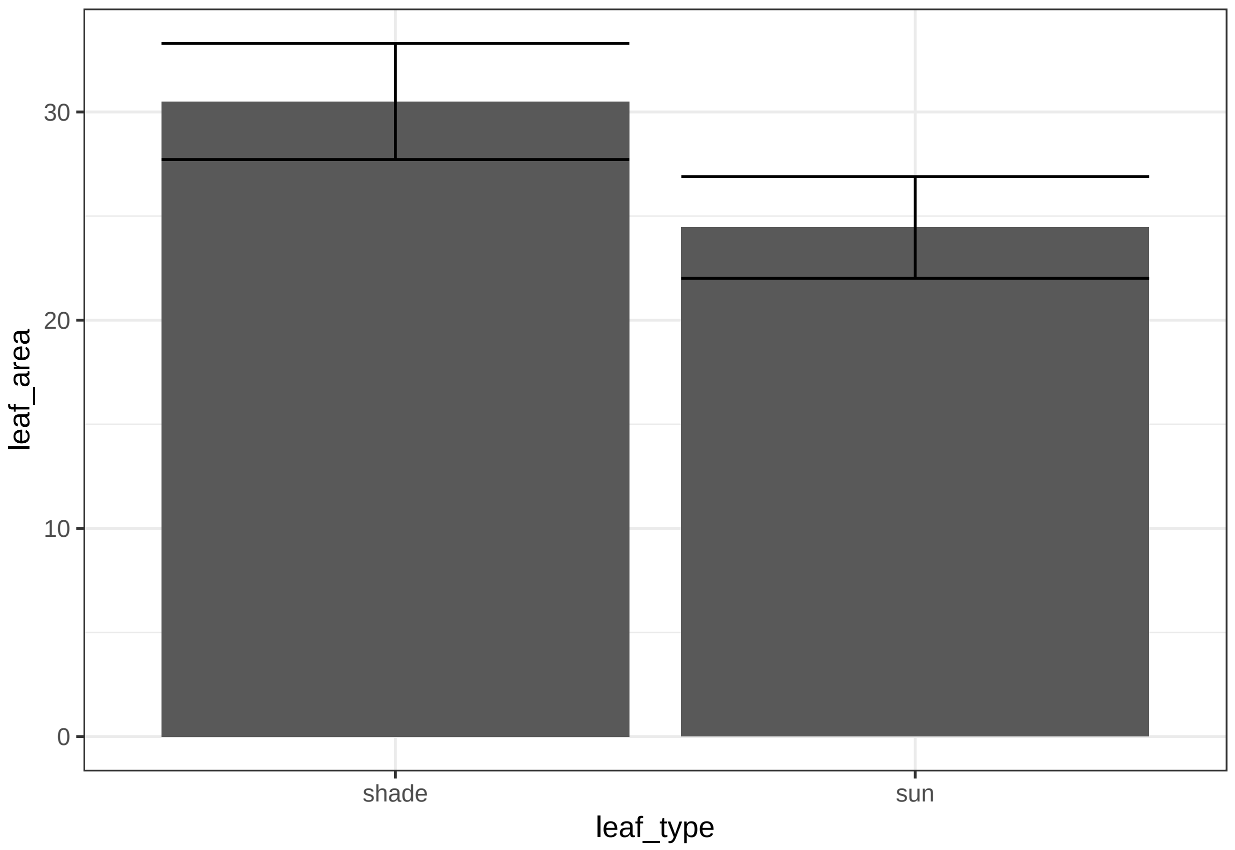
2.1.3 Confidence interval plot

```
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
```



2.1.4 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
```



2.1.5 T-test

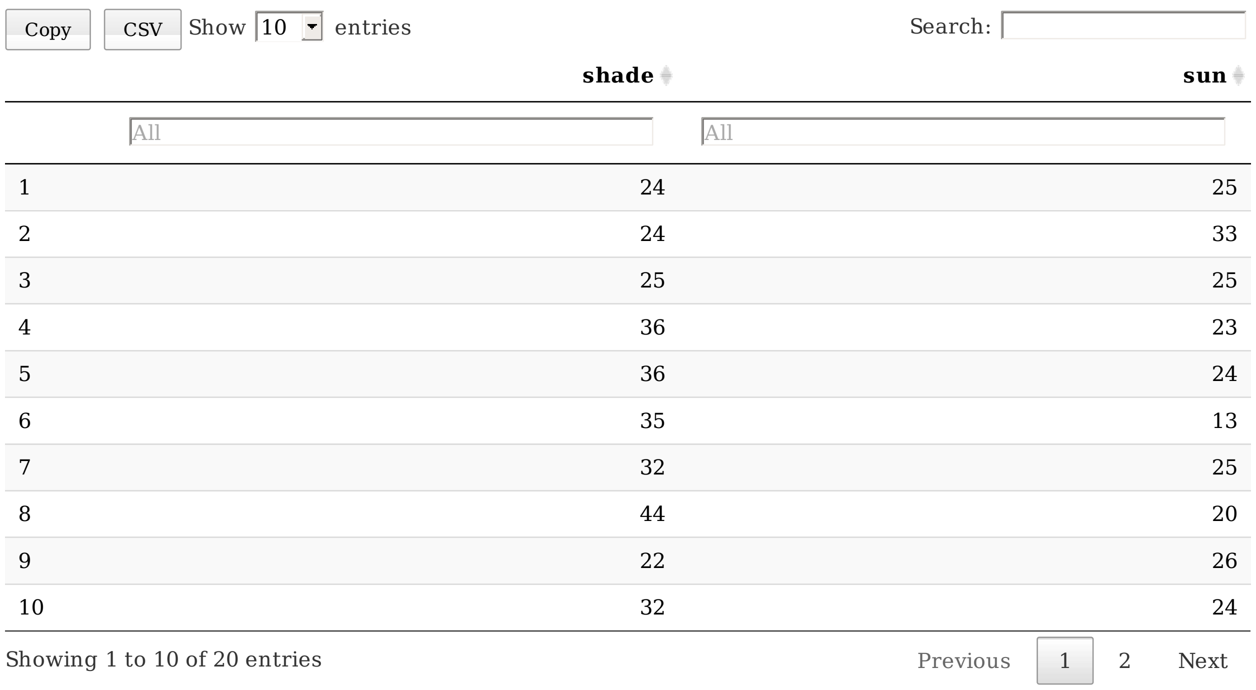
```
t.test(d$leaf_area~d$leaf_type)
```

```
##
##  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
```

2.2 Wilcoxon test (also known as 'Mann-Whitney' test)

2.2.1 Wide format

```
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

2.2.2 Wilcoxon test

```
wilcox.test(d2$shade,d2$sun)
```

```
## Warning in wilcox.test.default(d2$shade, d2$sun): cannot compute exact p-
## value with ties
```

```
##
##  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
```

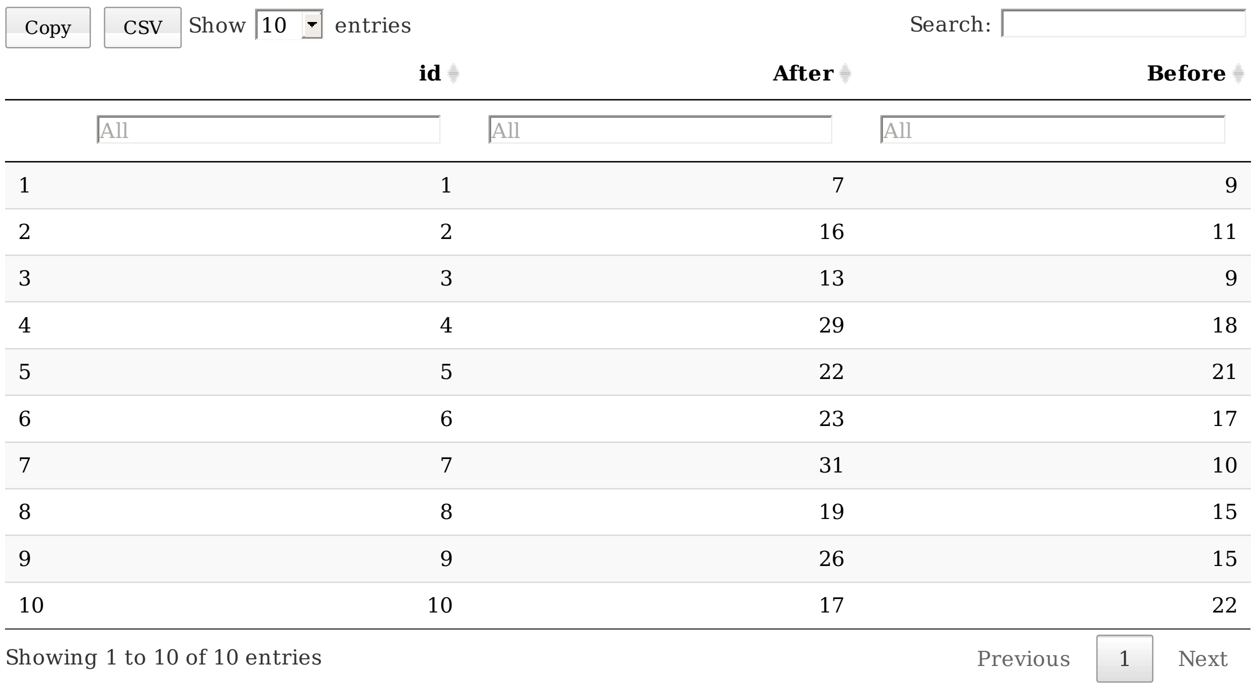
2.3 Paired T-test

2.3.1 Data formats

2.3.1.1 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:
All
All
All
Previous
1
Next

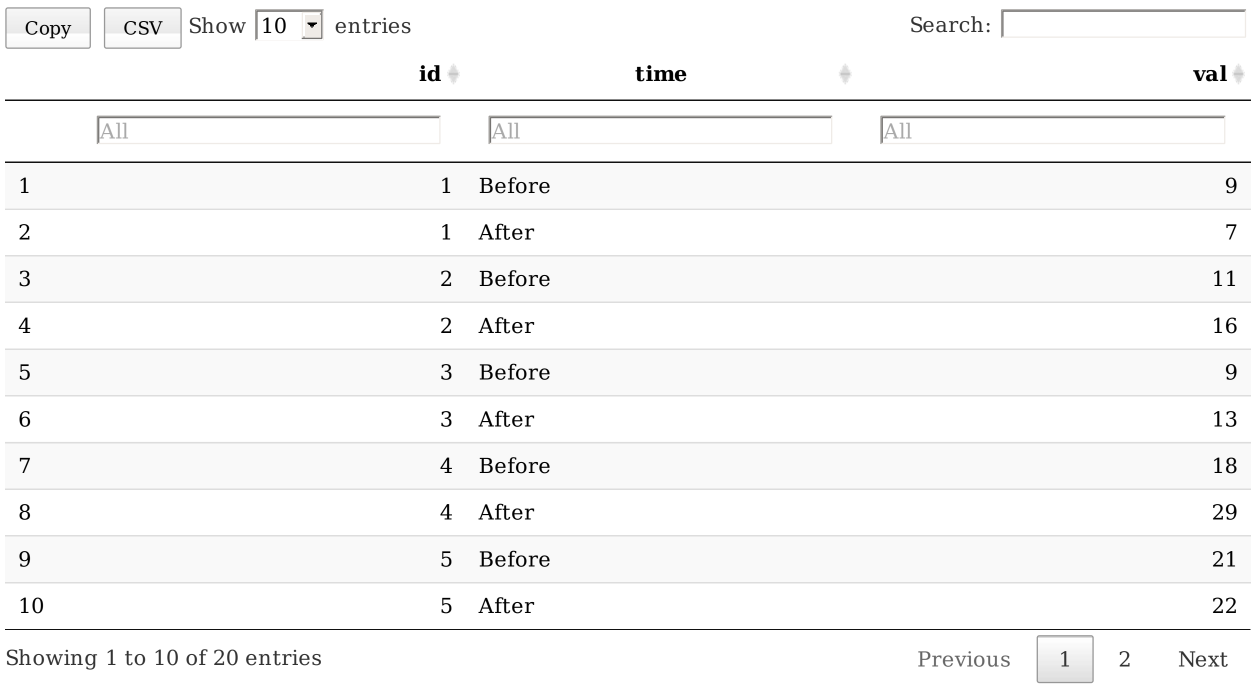
	id	After	Before
1	1	7	9
2	2	16	11
3	3	13	9
4	4	29	18
5	5	22	21
6	6	23	17
7	7	31	10
8	8	19	15
9	9	26	15
10	10	17	22

Showing 1 to 10 of 10 entries

2.3.1.2 Long format

For the classic paired t-test function the long format is best changed to wide.

```
d<-read.csv("/home/aqm/data/paired1.csv")
dt(d)
```


Copy
CSV
Show 10 entries
Search:
All
All
All
Previous
1
2
Next

	id	time	val
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

```
## 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

2.3.2 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
```

2.4 Paired Wilcoxon test

2.4.1 Wide format

The wide format seems the natural one to use in this case.

```r
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

2.4.2 Wilcoxon test

```r
wilcox.test(d2$Before,d2$After, paired=)
```

```
## Warning in wilcox.test.default(d2$Before, d2$After, paired = ): cannot
## compute exact p-value with ties
```

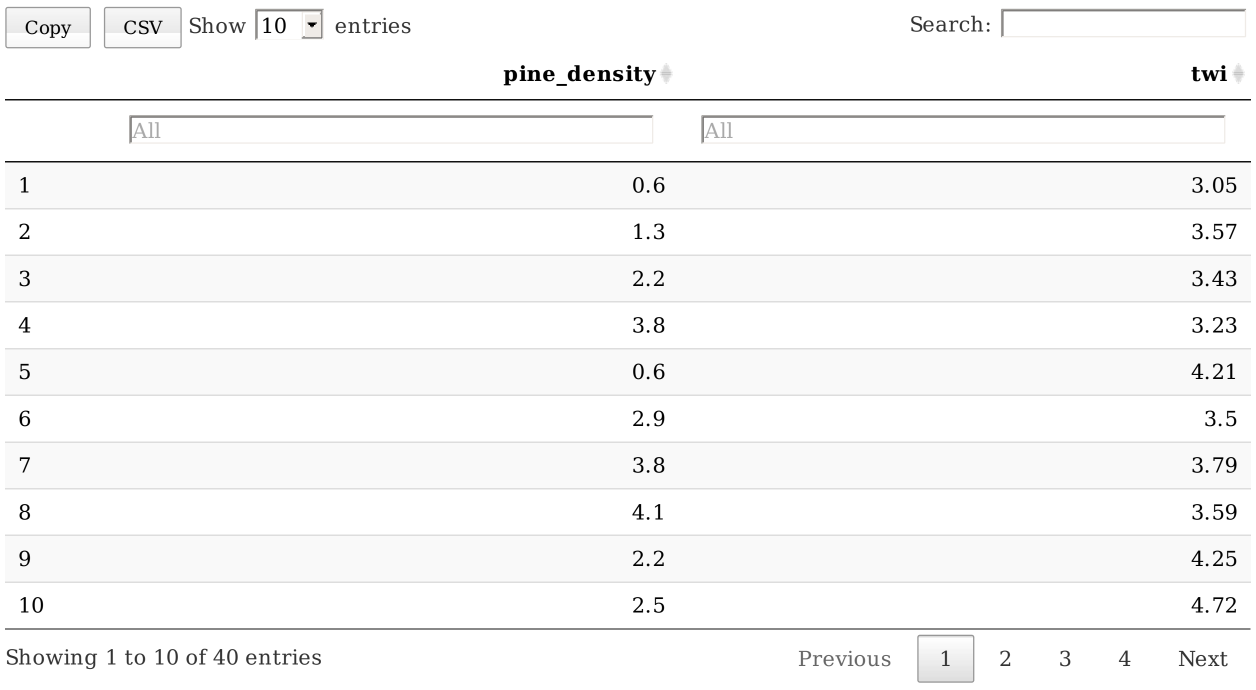
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  d2$Before and d2$After
## W = 26, p-value = 0.07522
## alternative hypothesis: true location shift is not equal to 0
```

2.5 Correlation test

2.5.1 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.

```r
d<-read.csv("/home/aqm/data/arne_pines_simple.csv")
dt(d)
```

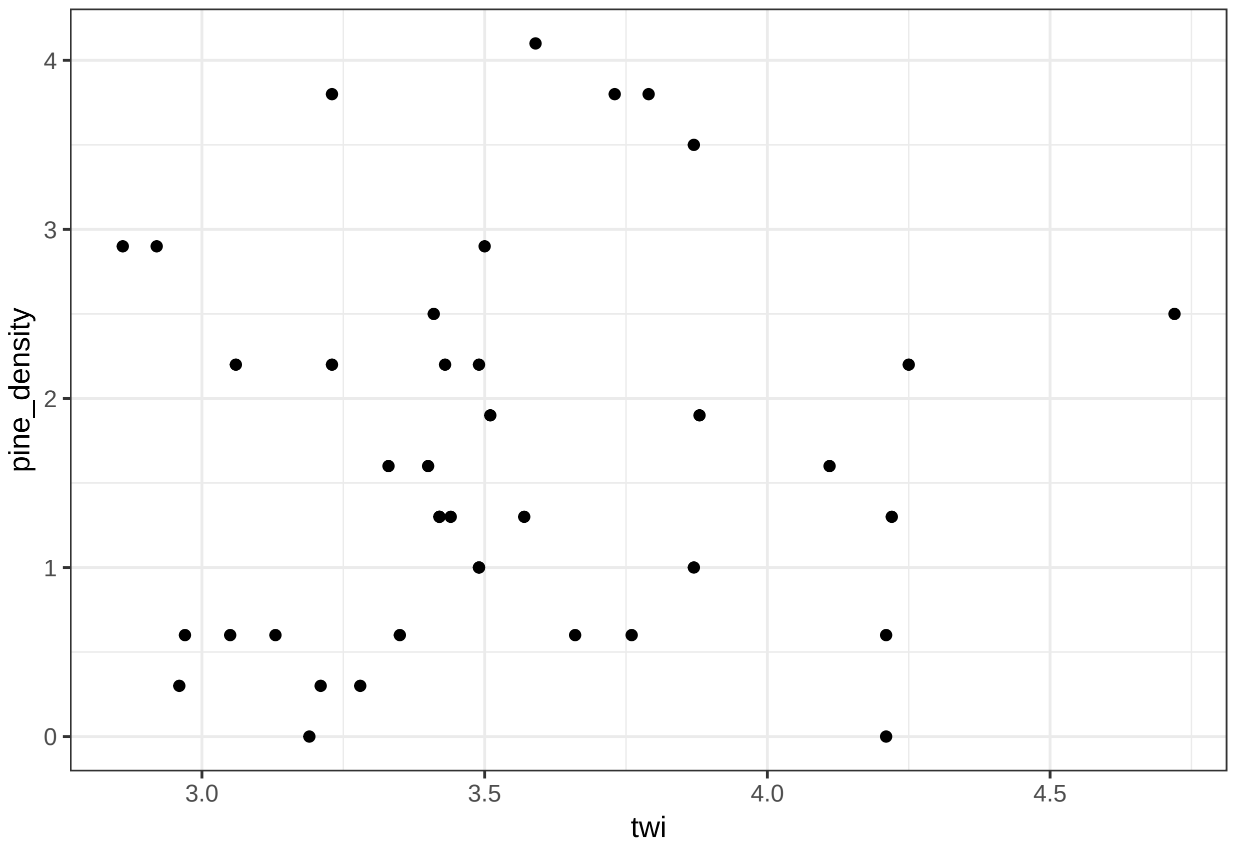


	pine_density	twi
1	0.6	3.05
2	1.3	3.57
3	2.2	3.43
4	3.8	3.23
5	0.6	4.21
6	2.9	3.5
7	3.8	3.79
8	4.1	3.59
9	2.2	4.25
10	2.5	4.72

Showing 1 to 10 of 40 entries

2.5.2 Scatterplot

```r
g0<-ggplot(d,aes(x=twi,y=pine_density)) +geom_point()
g0
```



2.5.3 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
```

2.5.4 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")
```

```
## Warning in cor.test.default(d$pine_density, d$twi, method = "spearman"):
## Cannot compute exact p-value with ties

##
```

```
##  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
```

2.6 Chi squared contingency tables

2.6.1 Data formats

2.6.1.1 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

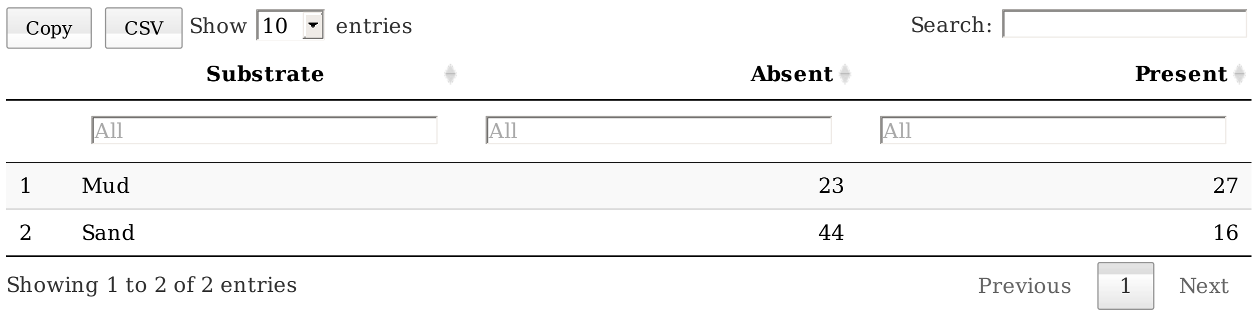
Showing 1 to 10 of 110 entries Previous 1 2 3 4 5 … 11 Next

2.6.1.2 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

2.6.2 Table of counts

2.6.2.1 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

2.6.2.2 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

2.6.3 Table of Proportions

2.6.3.1 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

2.6.3.2 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

2.6.3.3 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

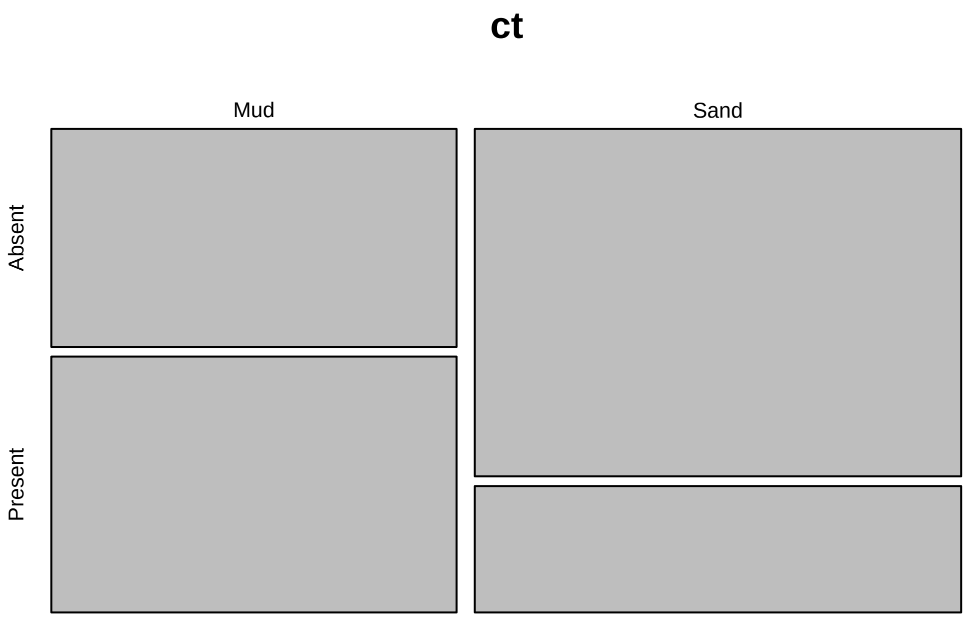
65.7

37.2

2.6.4 Plots

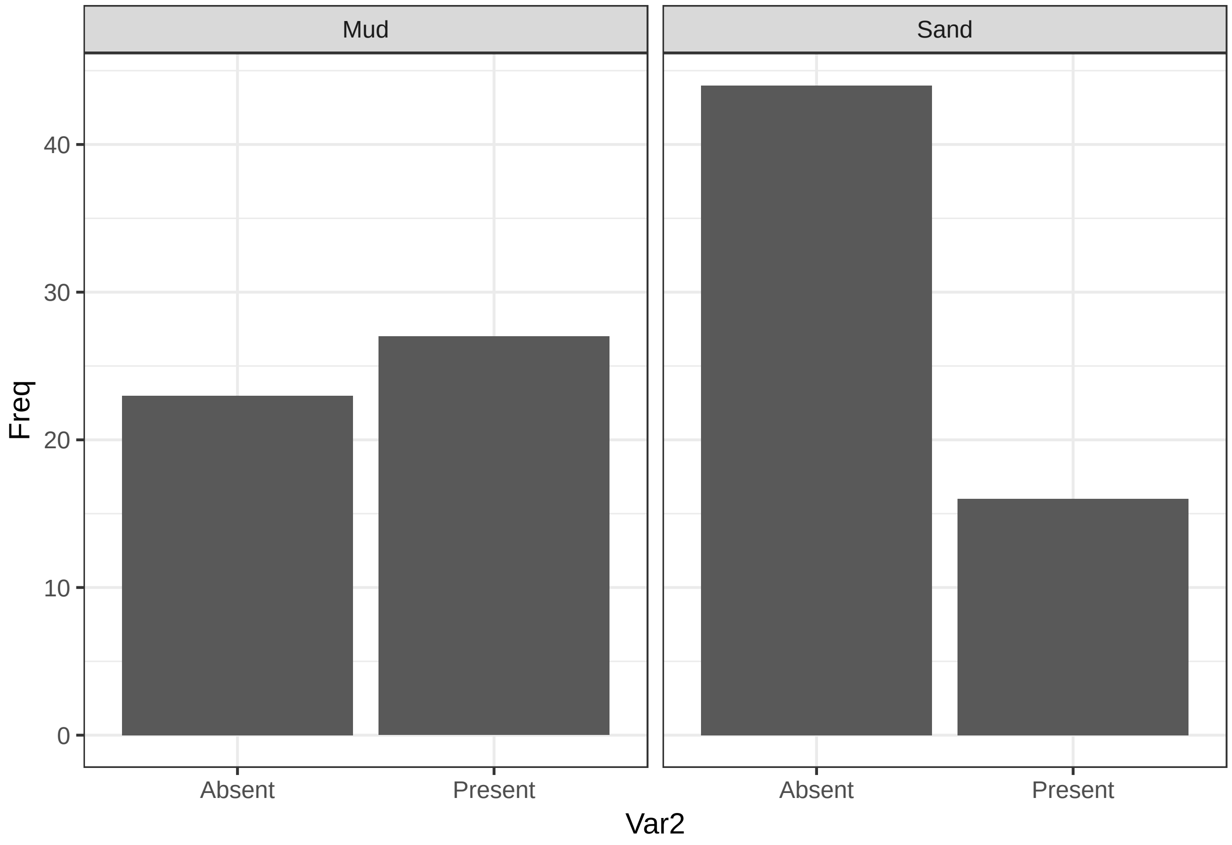
2.6.4.1 Mosaic plot

```
mosaicplot(ct)
```



2.6.4.2 Barplot

```
ct_d<-data.frame(ct)

bc<-ggplot(ct_d,aes(x=Var2,y=Freq))+geom_bar(stat="identity")
bc + facet_wrap(~Var1)
```



2.6.5 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
```

2.6.6 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 3

Regression and ANOVA

3.1 Packages needed

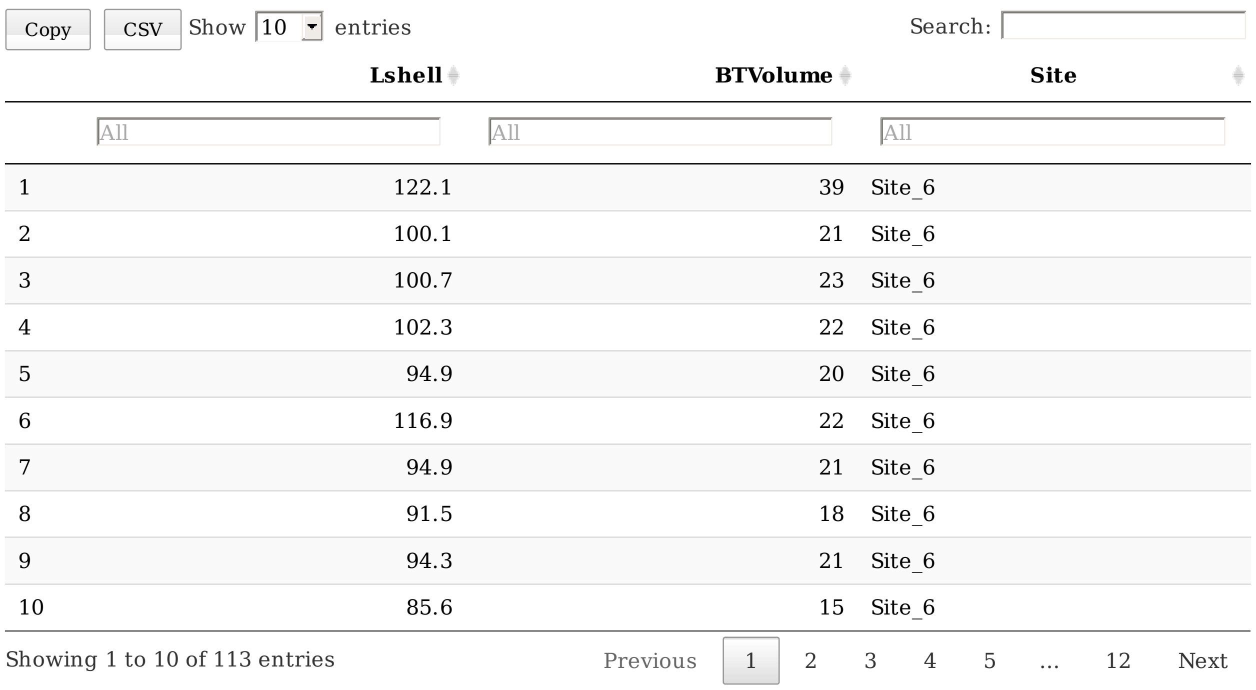
Include this chunk at the top of you analysis to ensure that you have all the packages. It also includes the wrapper to add buttons to a data table if you want to use this. Remember that data tables can only be included in HTML documents.

```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
  ))
```

3.2 Univariate

3.2.1 Data

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



	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

3.3 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

```
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
```

3.4 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
```

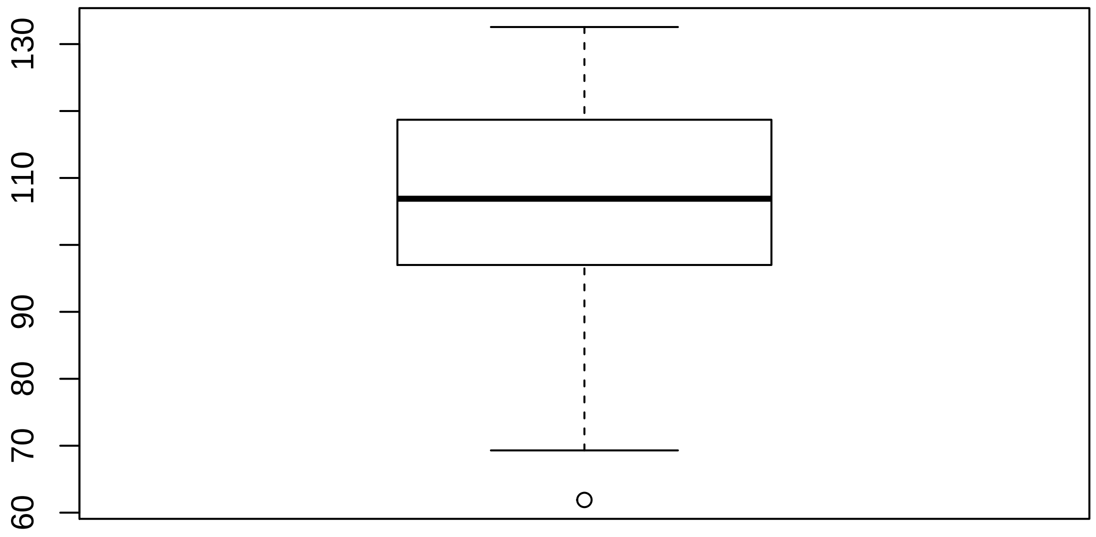
```
var(d$Lshell, na.rm=TRUE)
```

```
## [1] 220.3397
```

3.5 Simple boxplot of one variable

Useful for your own quick visualisation.

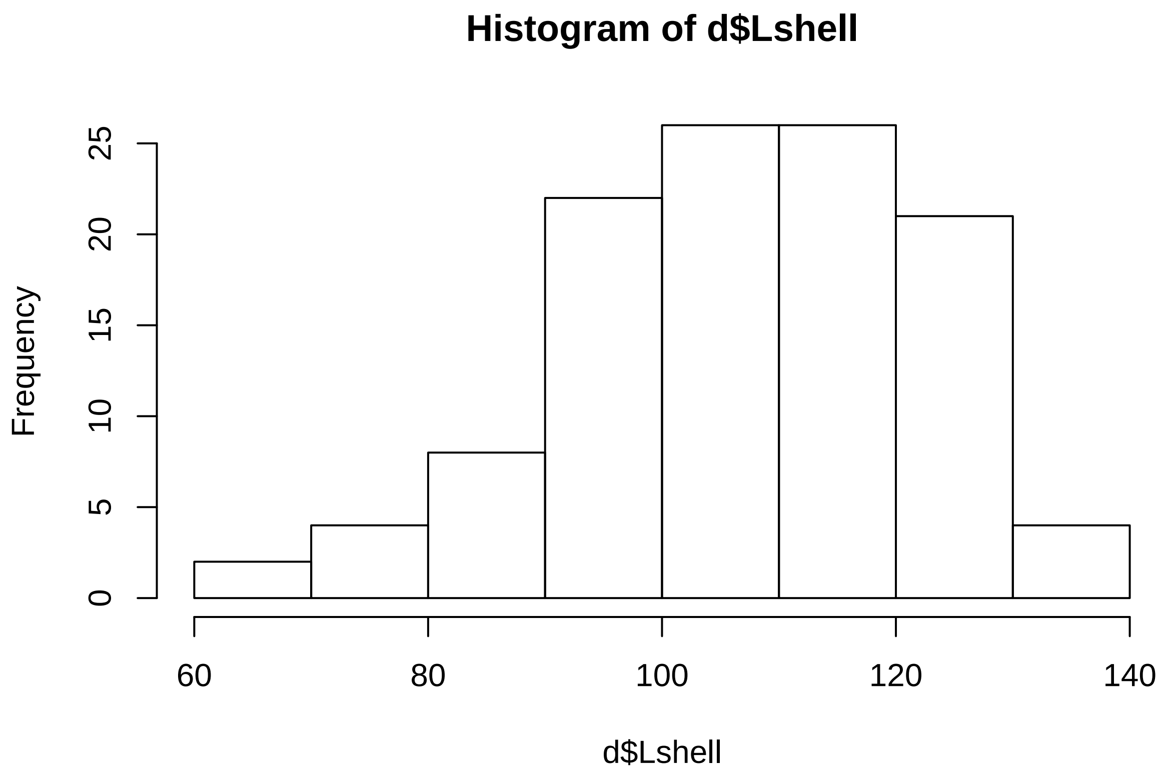
```
boxplot(d$Lshell)
```



3.6 Simple histogram of one variable

Useful for your own quick visualisation.

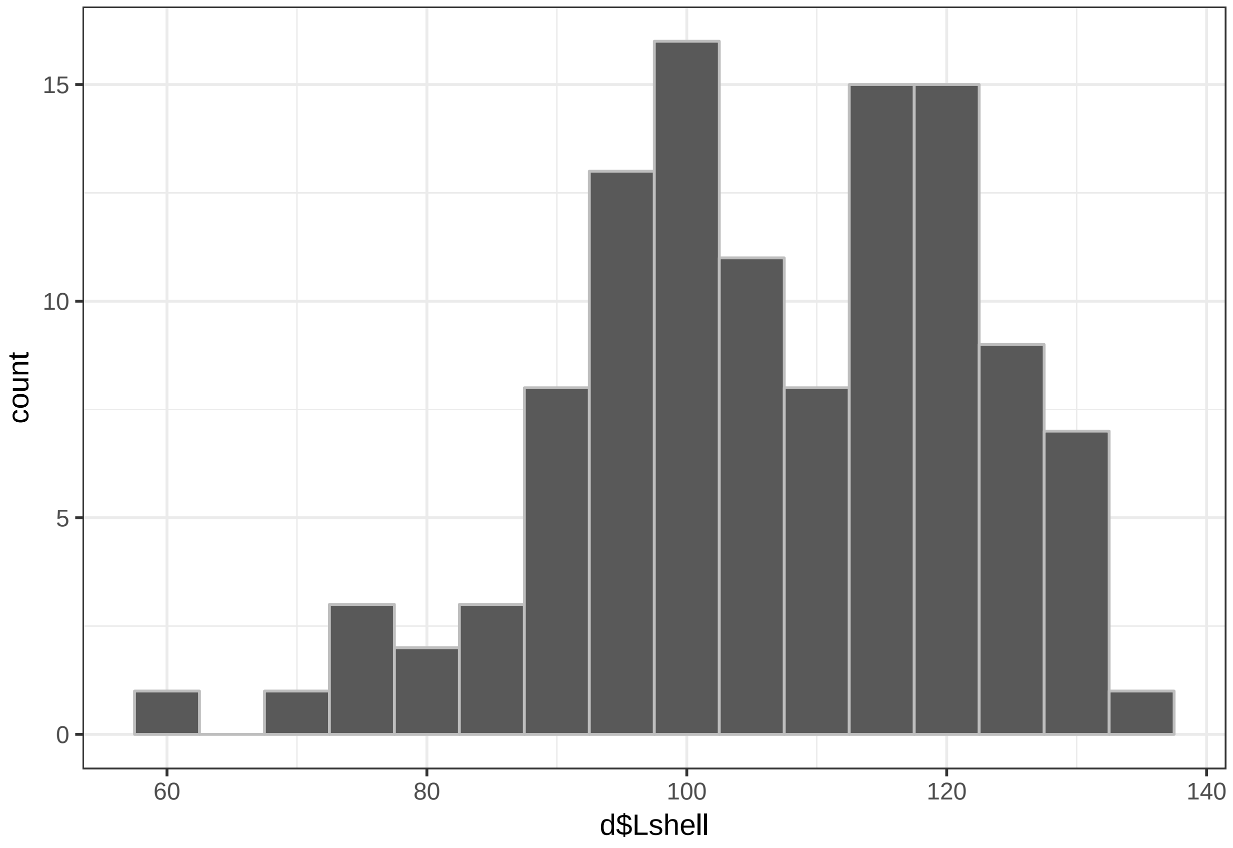
```
hist(d$Lshell)
```



3.7 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)
```



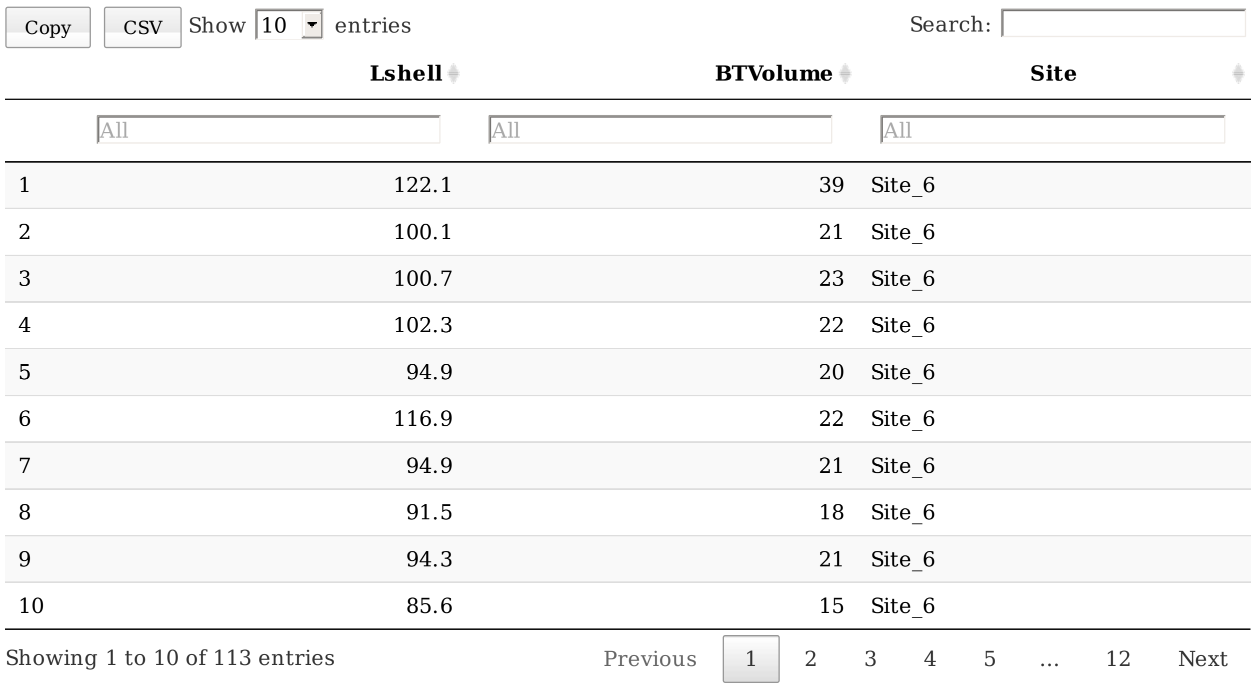
Chapter 4

Regression

4.1 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:

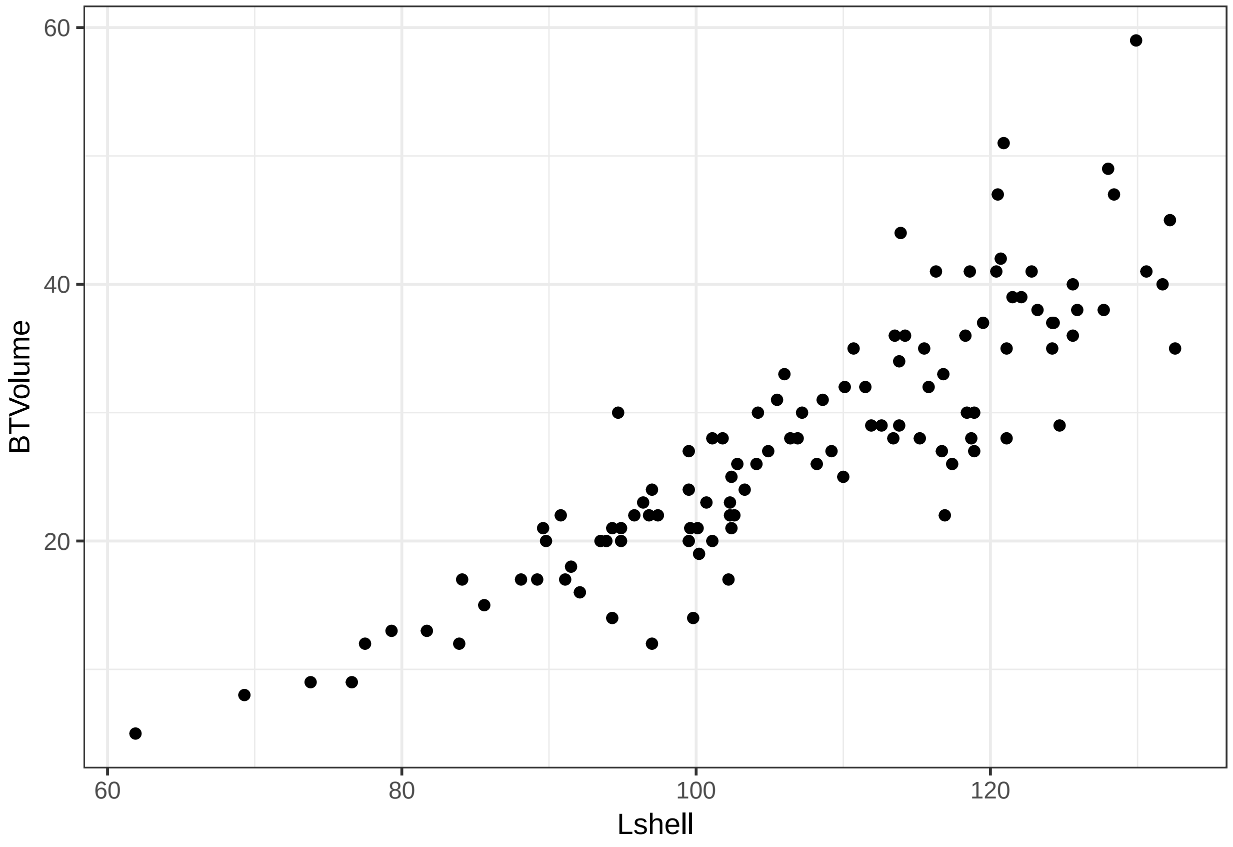
	Lshell	BTVolume	Site
	All	All	All
1	122.1	39	Site_6
2	100.1	21	Site_6
3	100.7	23	Site_6
4	102.3	22	Site_6
5	94.9	20	Site_6
6	116.9	22	Site_6
7	94.9	21	Site_6
8	91.5	18	Site_6
9	94.3	21	Site_6
10	85.6	15	Site_6

Showing 1 to 10 of 113 entries

Previous 1 2 3 4 5 … 12 Next

4.2 Scatterplot without fitted line

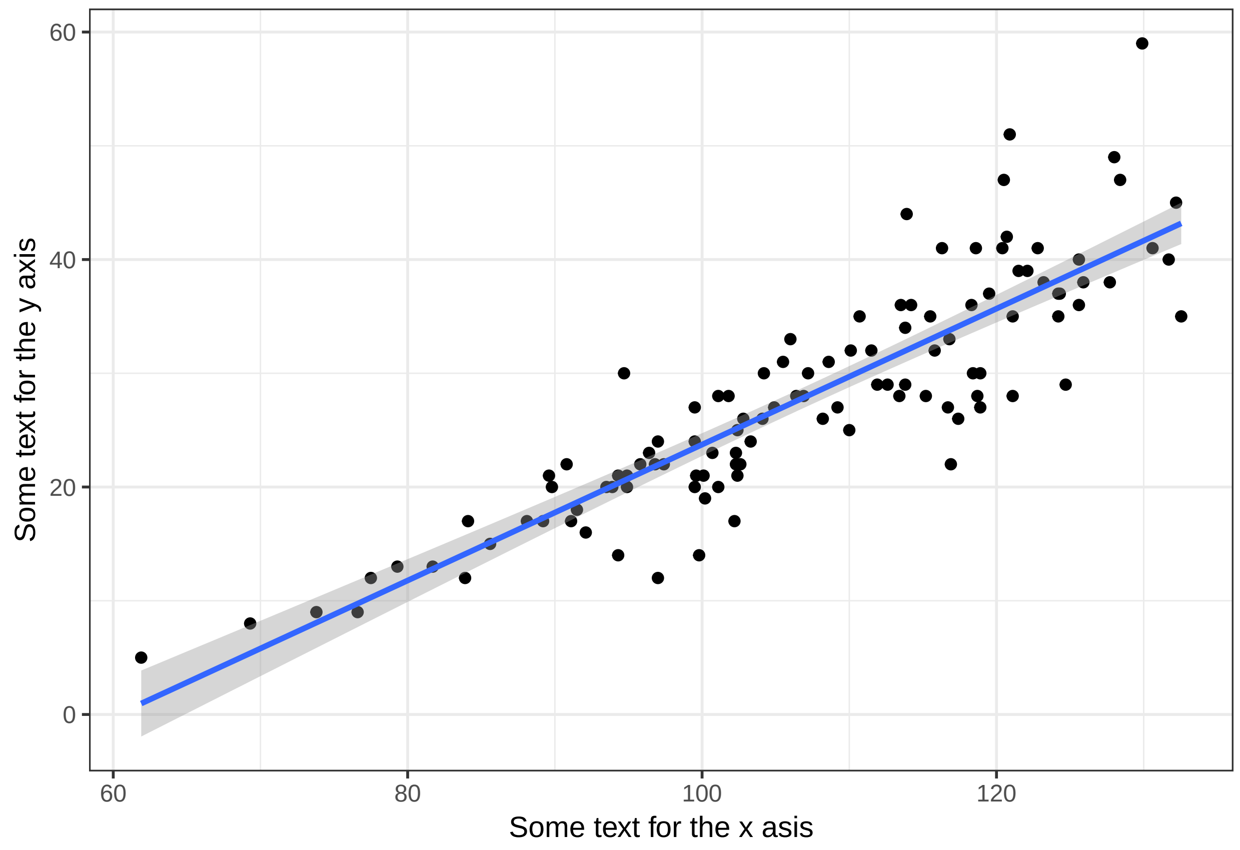
```
g0<-ggplot(d,aes(x=Lshell,y=BTVolume))
g0+geom_point()
```



4.3 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")
```



4.4 Fitting a model

Change the names of the variables in the first line.

```
mod<-lm(data= d, BTVolume~Lshell)
```

4.4.1 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
```

4.4.2 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
```

4.4.3 Confidence intervals for the model parameters

```
confint(mod)
```

```
##                   2.5 %      97.5 %
## (Intercept) -42.6406346 -29.4070662
## Lshell        0.5361881   0.6588891
```

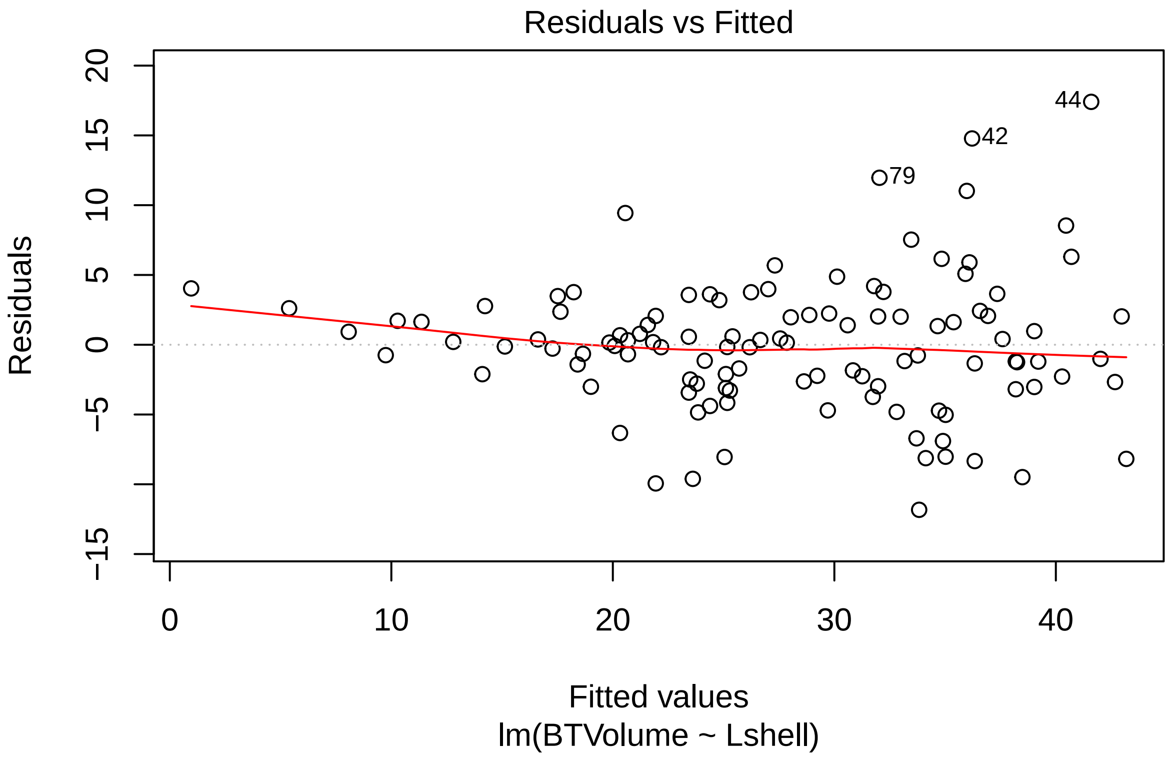
4.4.4 Extracting residuals

```
d$residuals<-residuals(mod)
```

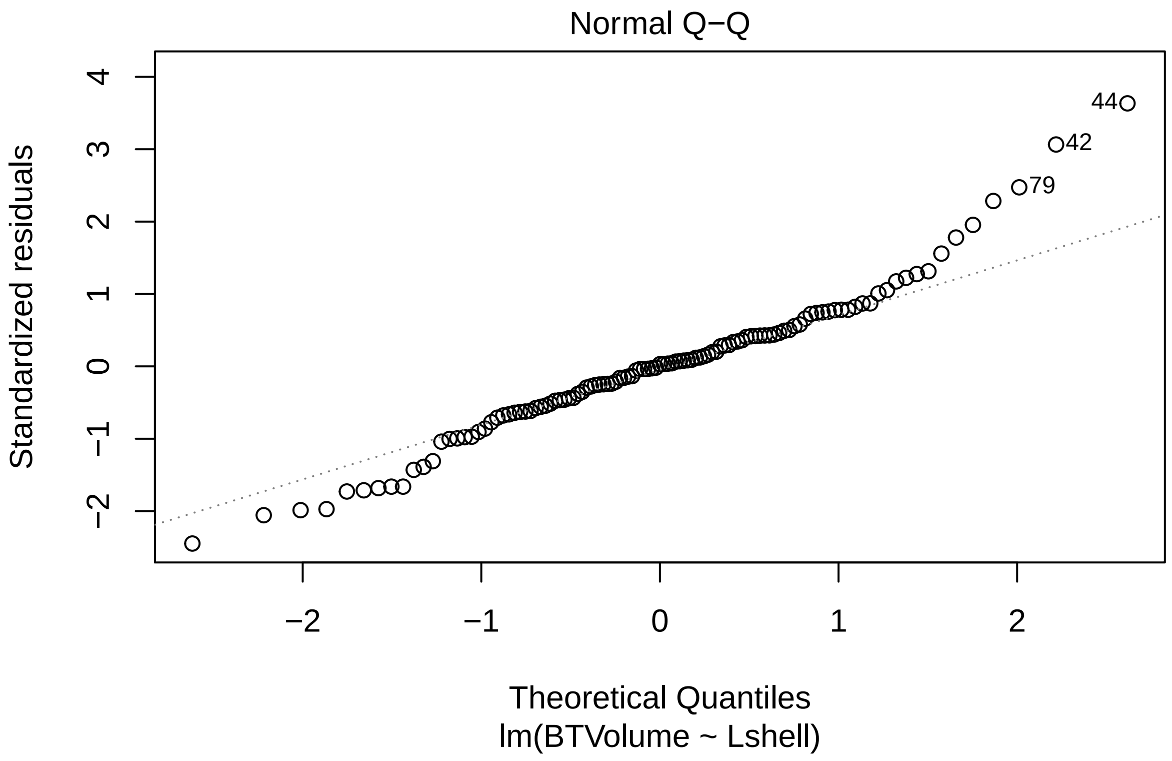
4.4.5 Model diagnostics

Look at the regression handout to understand these plots.

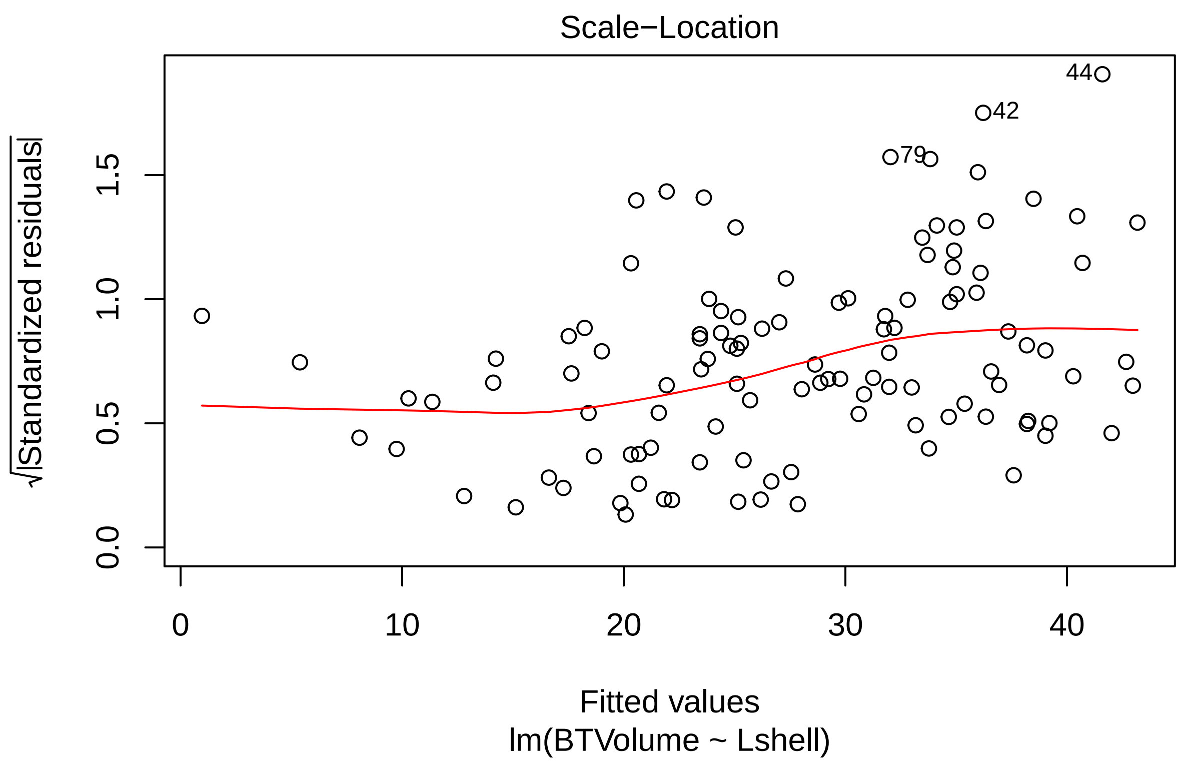
```
plot(mod,which=1)
```



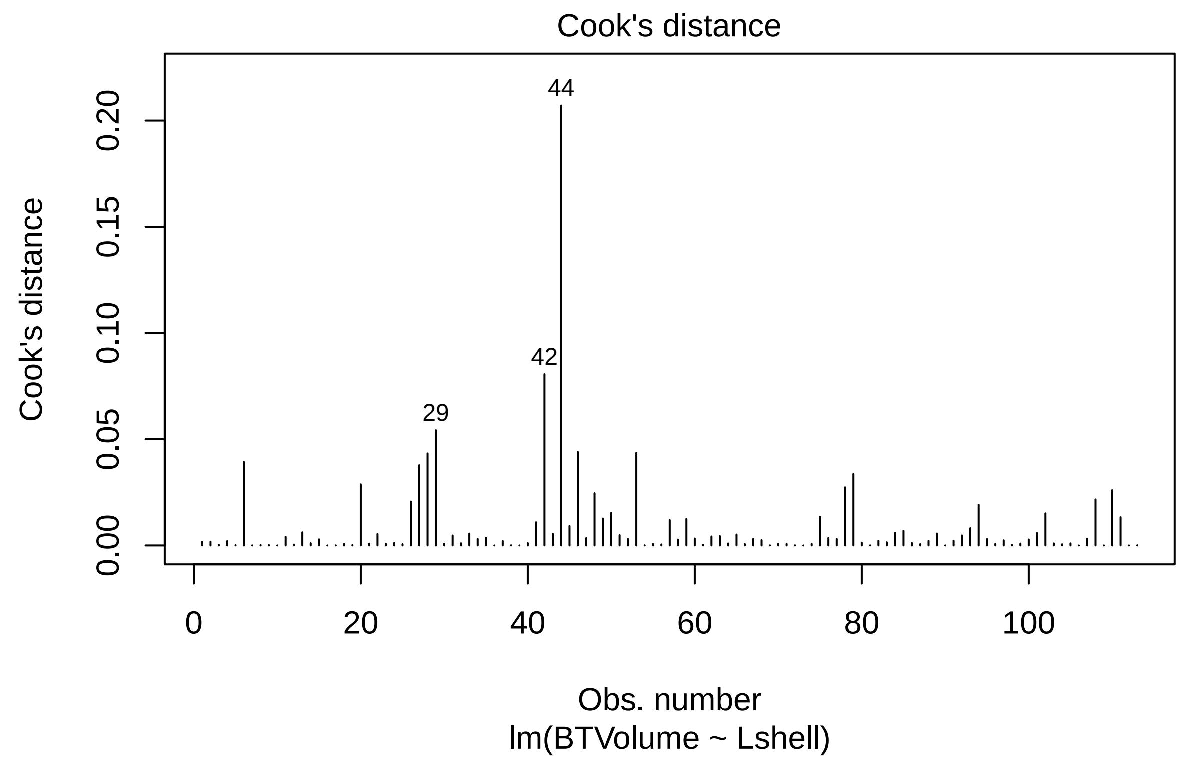
```r
plot(mod,which=2)
```



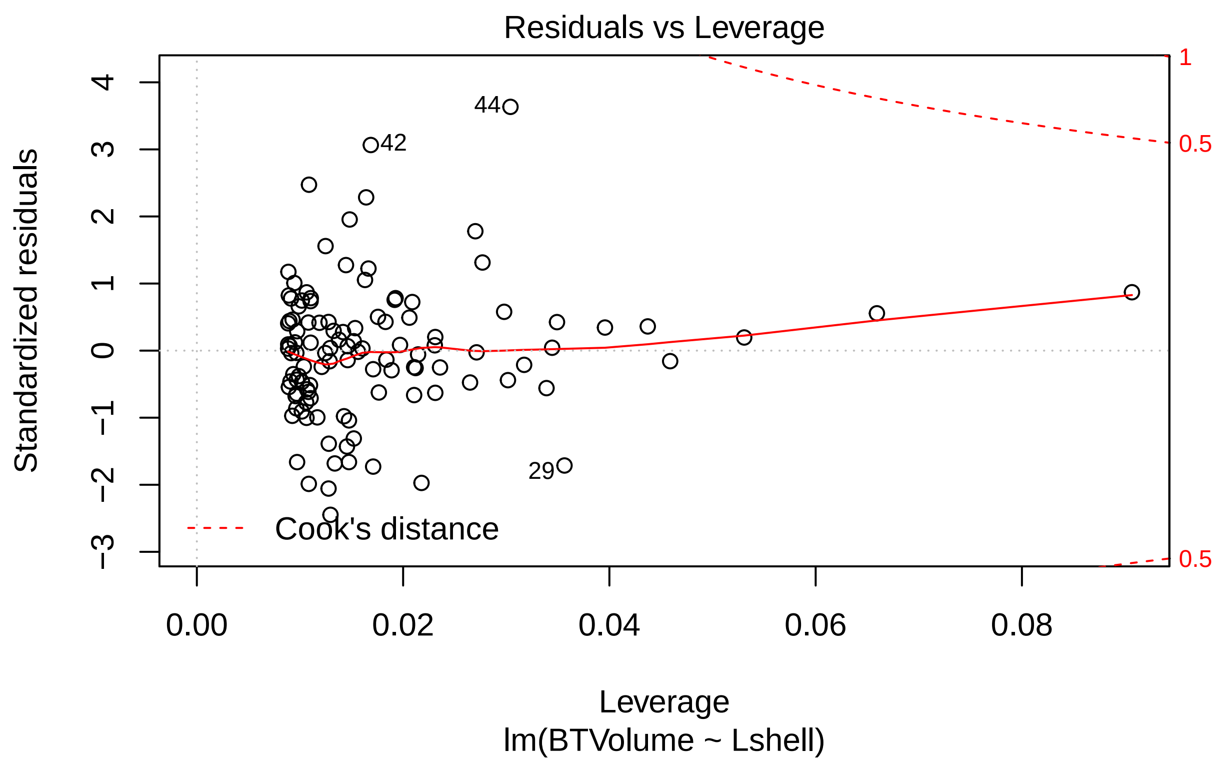
```r
plot(mod,which=3)
```



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



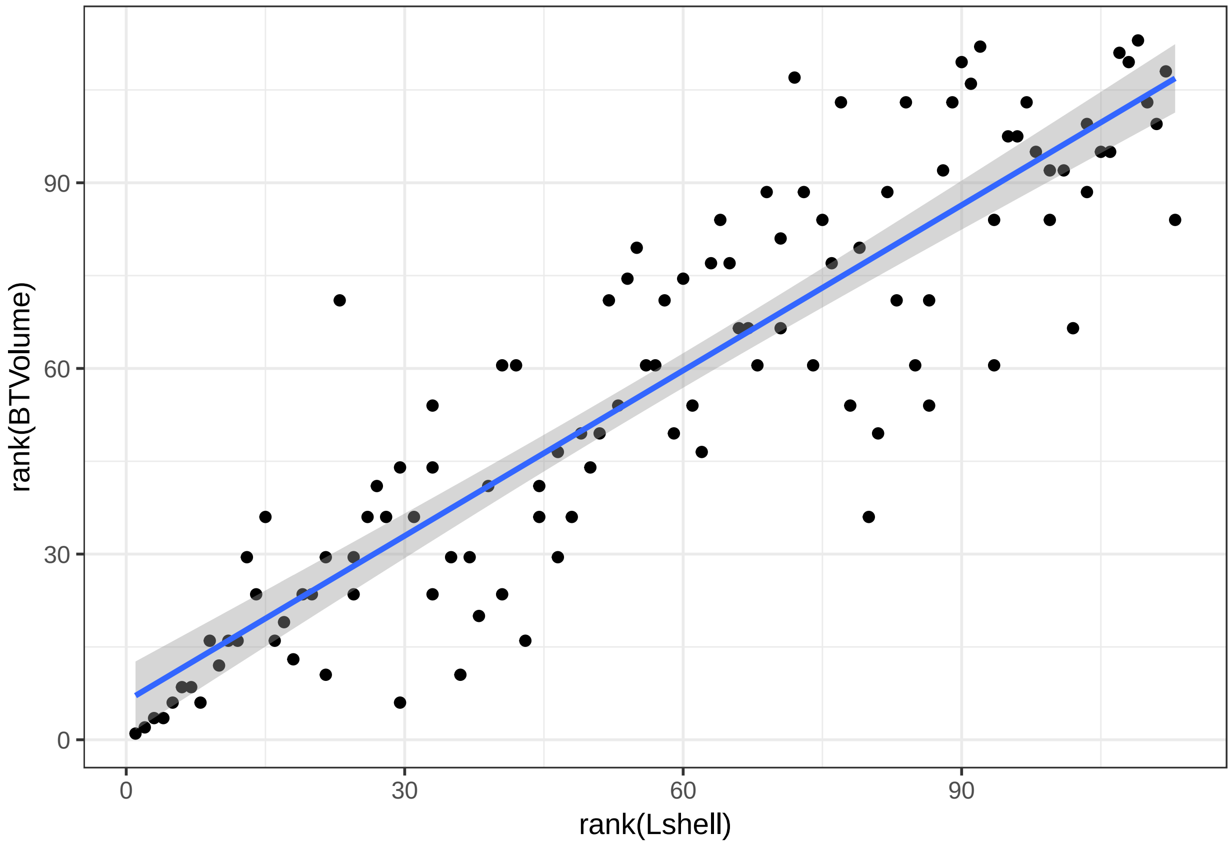
```r
plot(mod,which=5)
```



4.4.6 Spearman's rank correlation

Used if all else fails. Not needed with these data, but included for reference.

```r
g0<-ggplot(d,aes(x=rank(Lshell),y=rank(BTVolume)))
g0+geom_point() + geom_smooth(method="lm")
```



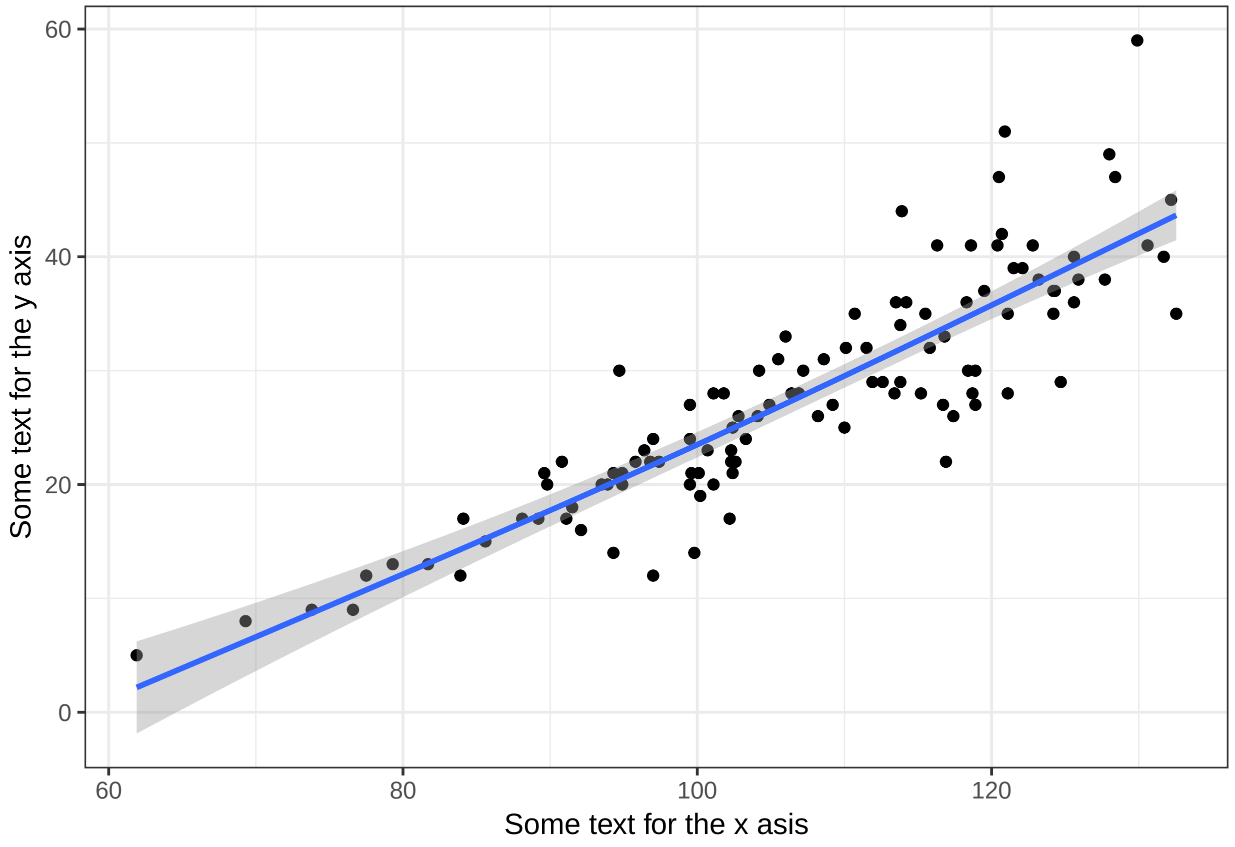
```r
cor.test(d$Lshell,d$BTVolume,method="spearman")
```

```
##
##  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
```

4.5 Fitting a spline

Only use if you suspect that the relationship is not well described by a straight line.

```r
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.

```
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 5

One way ANOVA

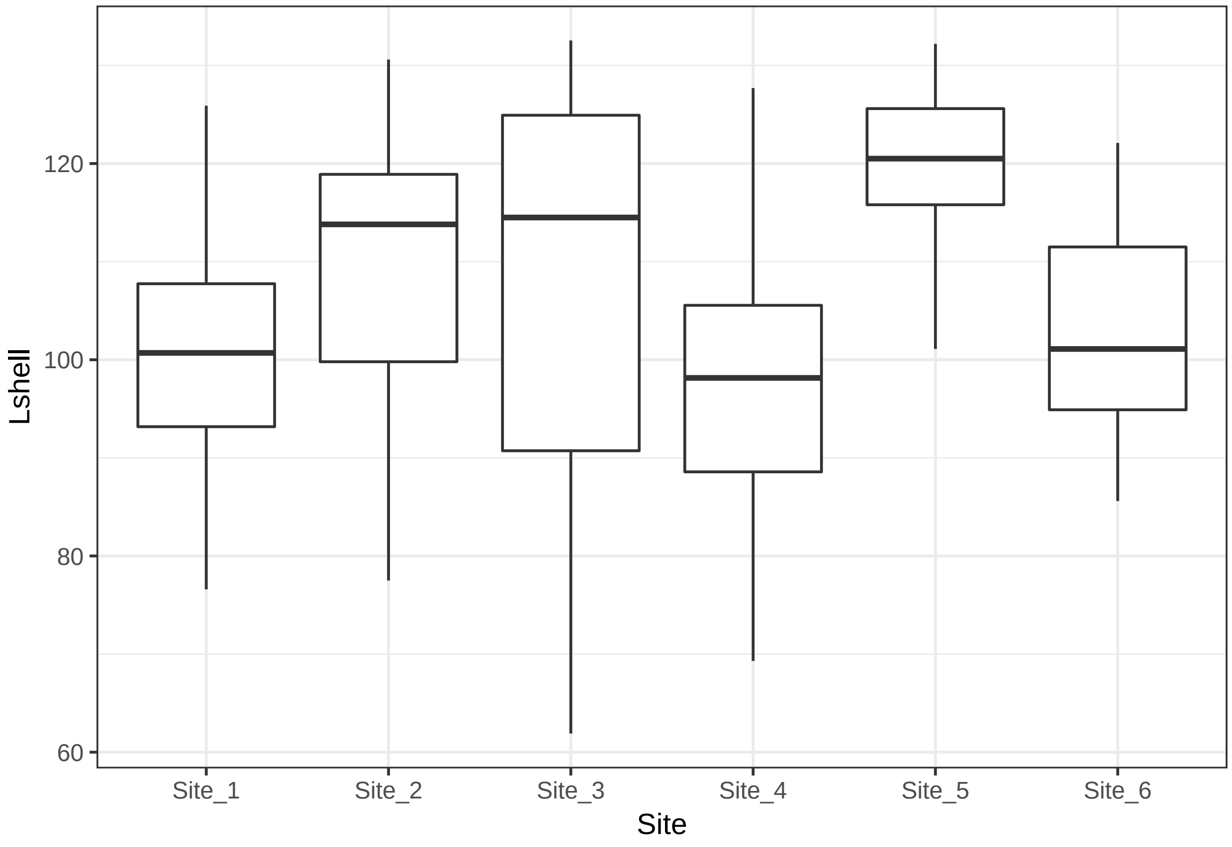
The purpose of one way anova is

1. Test whether there is greater variability between groups than within groups
2. Quantify any differences found between group means

5.1 Grouped boxplots

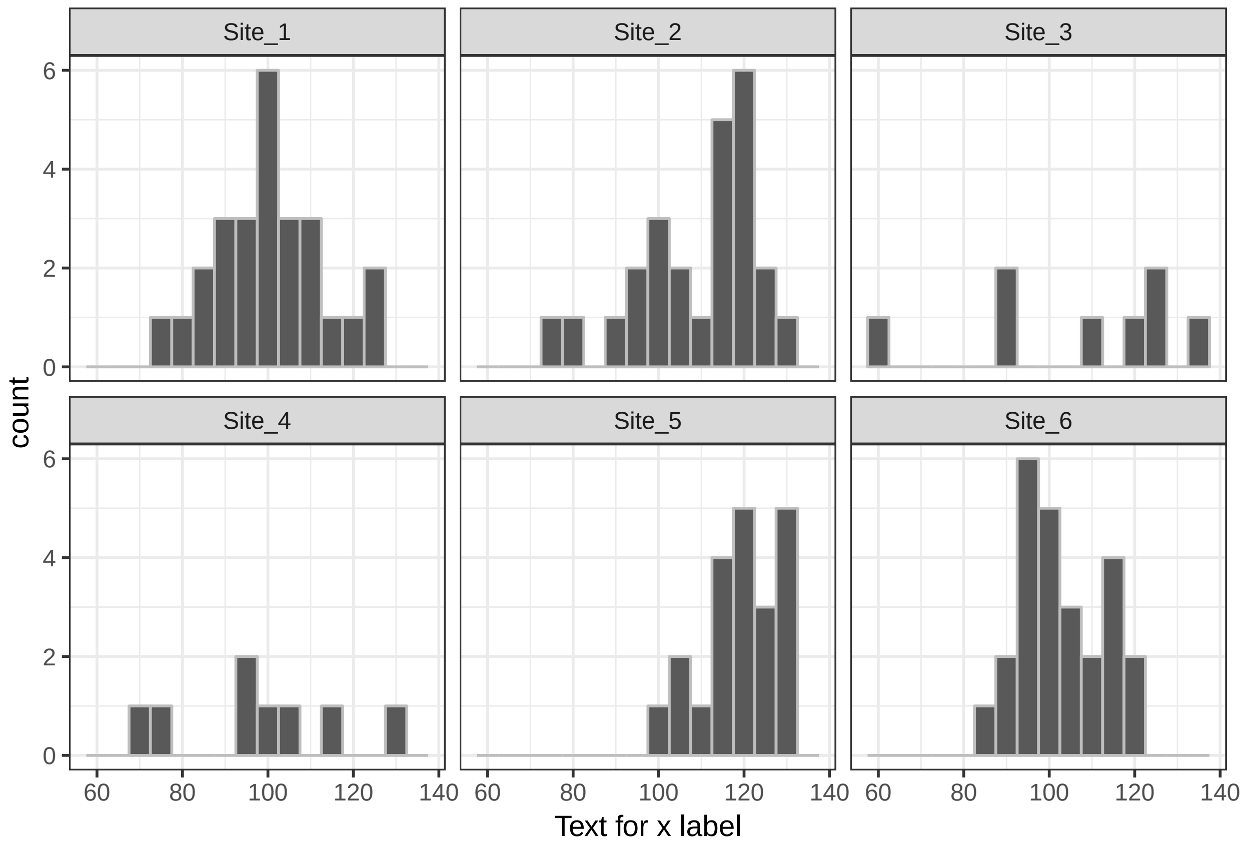
Exploratory plots

```
g0<-ggplot(d,aes(x=Site,y=Lshell))
g0+geom_boxplot()
```



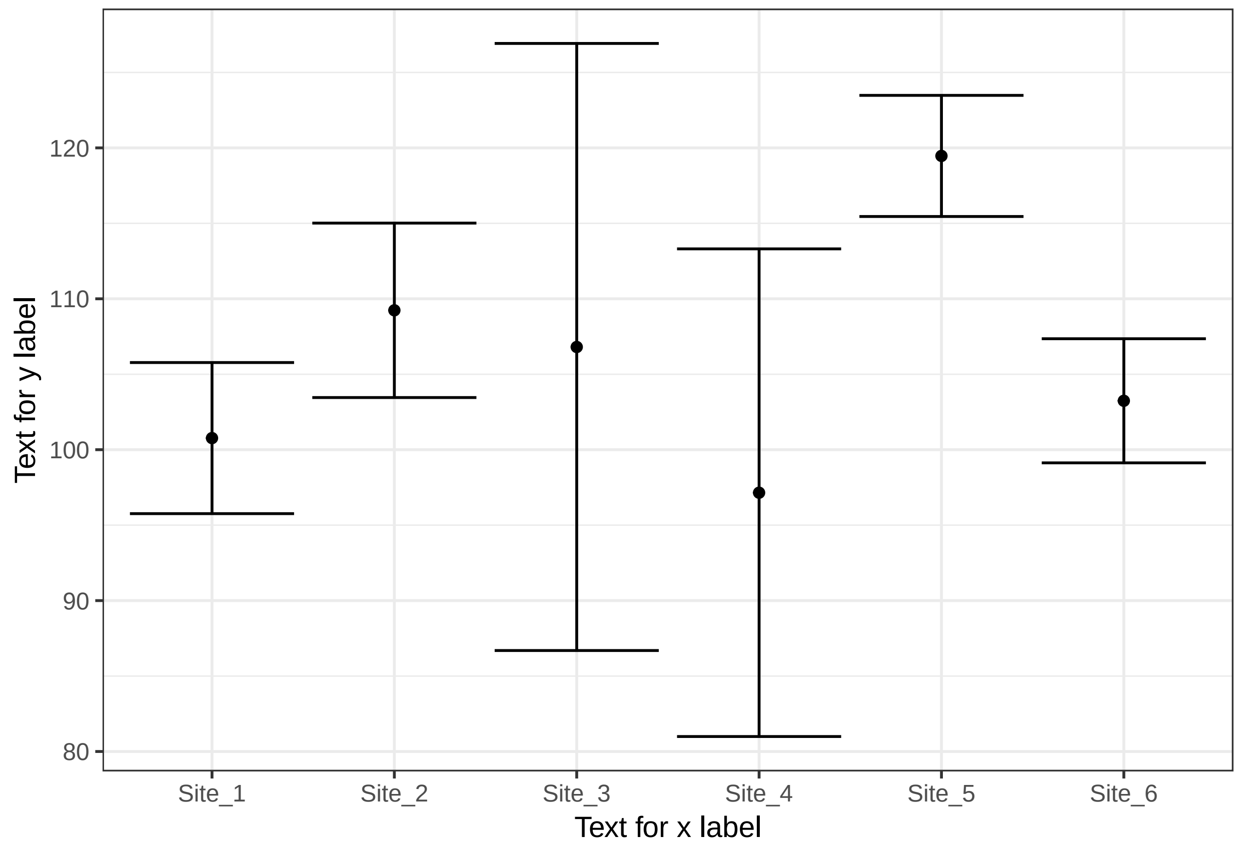
5.2 Histograms for each factor level

```
g0<-ggplot(d,aes(x=d$Lshell))
g1<-g0+geom_histogram(color="grey",binwidth = 5)

g1+facet_wrap(~Site) +xlab("Text for x label")
```



5.3 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")
```



5.4 Fitting ANOVA

```
mod<-lm(data=d,Lshell~Site)
```

5.4.1 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
```

5.4.2 Model summary

5.4.2.1 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
```

5.4.2.2 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)
```

5.4.2.3 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)
```

5.4.2.4 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"))
```

5.4.2.4.1 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)
```

5.4.3 Tukey corrected pairwise comparisons

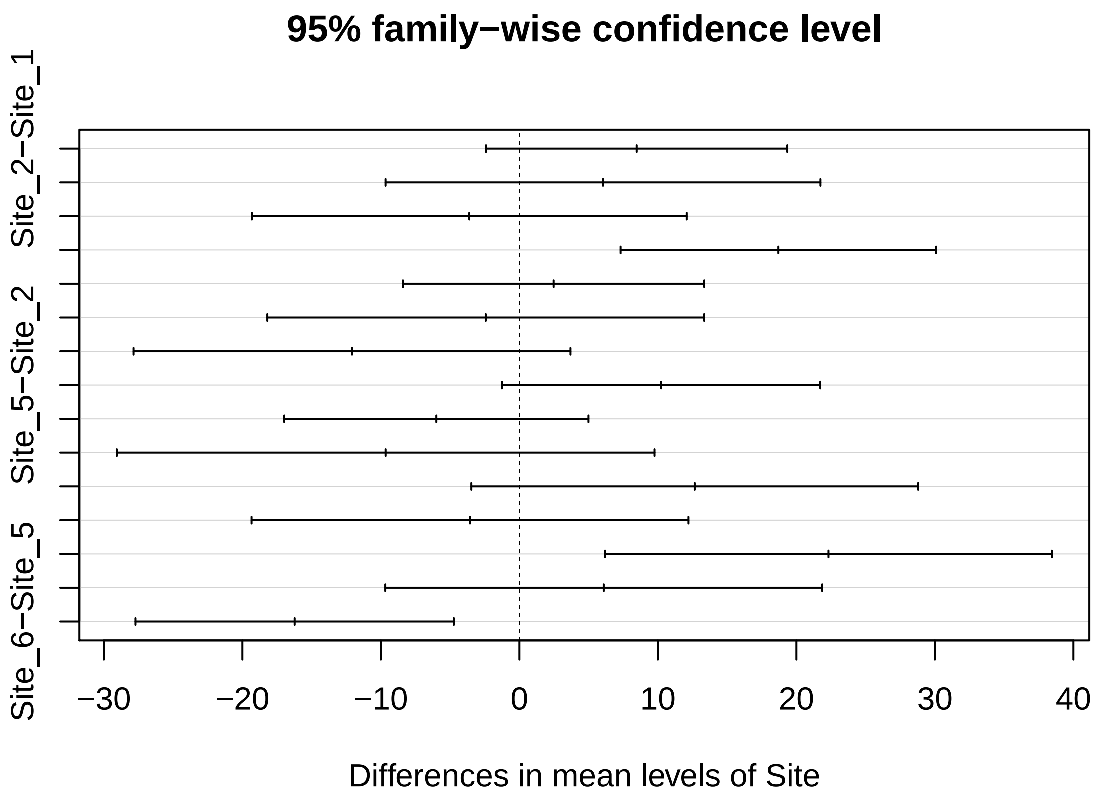
Use to find where signficant differences lie. This should confirm the pattern shown using the confidence interval plot.

```
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))
```



5.4.4 Anova with White's correction

This will give you the overall Anova table if there is heterogeneity of variance.

```r
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
```

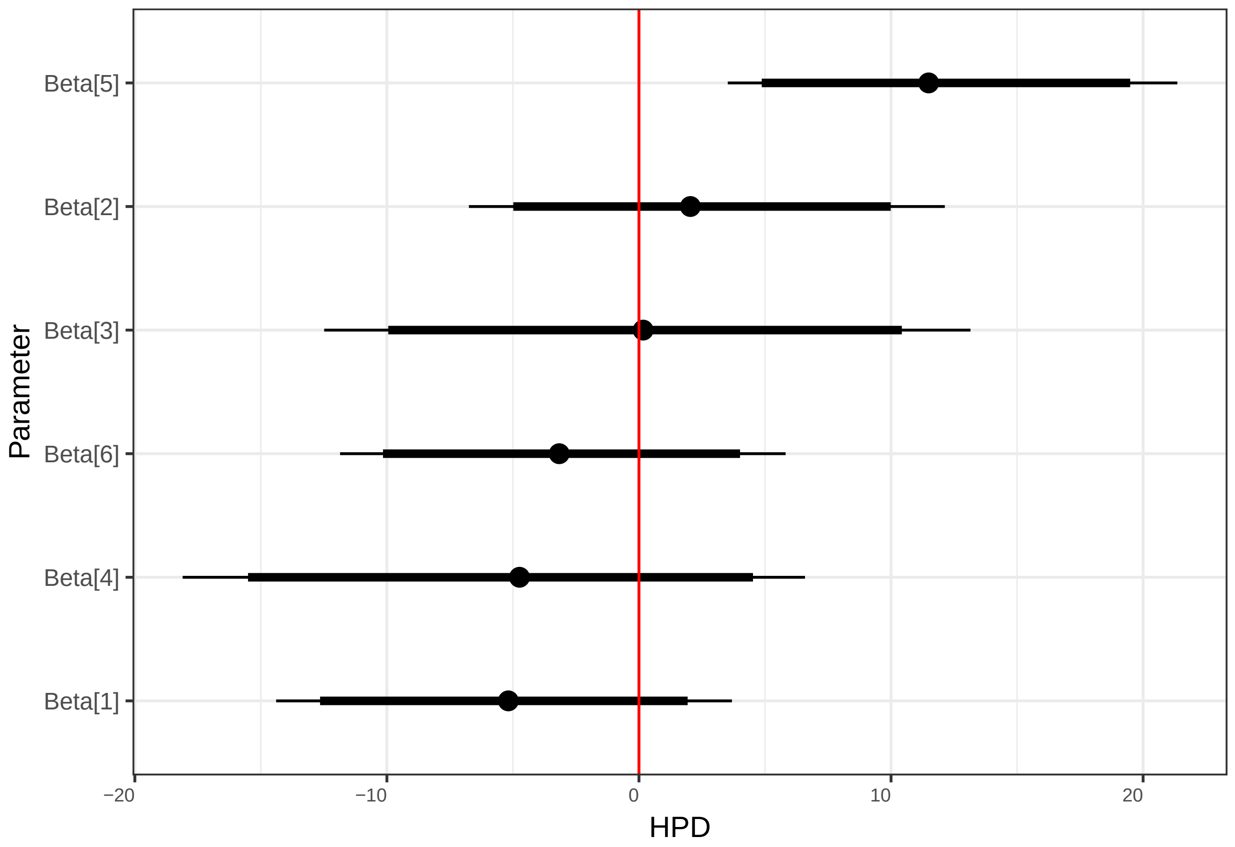
5.4.5 Bayesian model with shrinkage

Specialist model. Probably the best for these particular data, but seek guidance. **Don't do this at home kids!**

```r
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")
```



```r
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]       -5.2065 4.560  0.04560        0.08197
## Beta[2]        2.2120 4.700  0.04700        0.08533
## Beta[3]        0.1858 6.343  0.06343        0.07986
## Beta[4]       -5.0166 6.204  0.06204        0.08989
## Beta[5]       11.7369 4.561  0.04561        0.08363
## Beta[6]       -3.1205 4.457  0.04457        0.08181
## Difbeta[1,2]  -7.4184 3.640  0.03640        0.03640
## Difbeta[1,3]  -5.3923 6.478  0.06478        0.06829
## Difbeta[2,3]   2.0262 6.512  0.06512        0.06896
## Difbeta[1,4]  -0.1899 6.075  0.06075        0.06304
## Difbeta[2,4]   7.2285 6.317  0.06317        0.06818
## Difbeta[3,4]   5.2024 7.978  0.07978        0.08339
## Difbeta[1,5] -16.9434 3.336  0.03336        0.03443
## Difbeta[2,5]  -9.5250 3.424  0.03424        0.03424
## Difbeta[3,5] -11.5511 6.449  0.06449        0.06917
## Difbeta[4,5] -16.7535 6.332  0.06332        0.06757
## Difbeta[1,6]  -2.0860 3.111  0.03111        0.03111
## Difbeta[2,6]   5.3325 3.342  0.03342        0.03342
## Difbeta[3,6]   3.3063 6.382  0.06382        0.06787
## Difbeta[4,6]  -1.8961 6.026  0.06026        0.06328
## Difbeta[5,6]  14.8574 2.965  0.02965        0.02965
## mu_r         106.7106 4.195  0.04195        0.08194
## sigma_r        8.2901 3.887  0.03887        0.05050
##
## 2. Quantiles for each variable:
##
##                 2.5%       25%      50%       75%    97.5%
## Beta[1]      -14.395  -7.84891  -5.1814  -2.48557   3.6875
## Beta[2]       -6.747  -0.63448   2.0406   4.95843  12.1334
## Beta[3]      -12.488  -3.70929   0.1647   3.97258  13.1521
## Beta[4]      -18.104  -8.77090  -4.7398  -1.05371   6.5868
## Beta[5]        3.523   8.83569  11.4922  14.34445  21.3584
## Beta[6]      -11.858  -5.76622  -3.1635  -0.52198   5.8187
## Difbeta[1,2] -14.714  -9.79923  -7.3845  -5.02241  -0.3133
## Difbeta[1,3] -18.542  -9.46672  -5.3371  -1.23218   7.3429
## Difbeta[2,3] -10.998  -2.09946   1.9874   6.08006  15.2928
## Difbeta[1,4] -12.112  -4.17430  -0.2366   3.66956  12.2614
## Difbeta[2,4]  -4.950   3.16226   7.1195  11.34160  19.9193
## Difbeta[3,4]  -9.909  -0.02652   4.9860  10.17430  21.7295
## Difbeta[1,5] -23.423 -19.20651 -17.0111 -14.79280 -10.2803
## Difbeta[2,5] -16.314 -11.78930  -9.4956  -7.23590  -2.8225
## Difbeta[3,5] -24.757 -15.64663 -11.4113  -7.34284   0.9295
## Difbeta[4,5] -29.637 -20.86813 -16.7428 -12.43520  -4.6493
## Difbeta[1,6]  -8.231  -4.10564  -2.0774  -0.03864   3.9767
## Difbeta[2,6]  -1.197   3.07471   5.3421   7.59872  11.8768
## Difbeta[3,6]  -9.272  -0.71618   3.2819   7.30825  16.1412
## Difbeta[4,6] -14.053  -5.75891  -1.8764   2.03563   9.8128
## Difbeta[5,6]   8.960  12.94192  14.8784  16.81303  20.6977
## mu_r          97.861 104.39612 106.8568 109.15502 114.7138
## sigma_r        3.771   5.80926   7.4105   9.73286  18.0357
```

Chapter 6

Fitting polynomials

6.1 The data

```
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

6.2 Fitting linear model

6.2.1 Linear model fit

```
mod<-lm(data=d,richness~grain)
```

6.2.2 Linear model anova and summary

```
anova(mod)
```

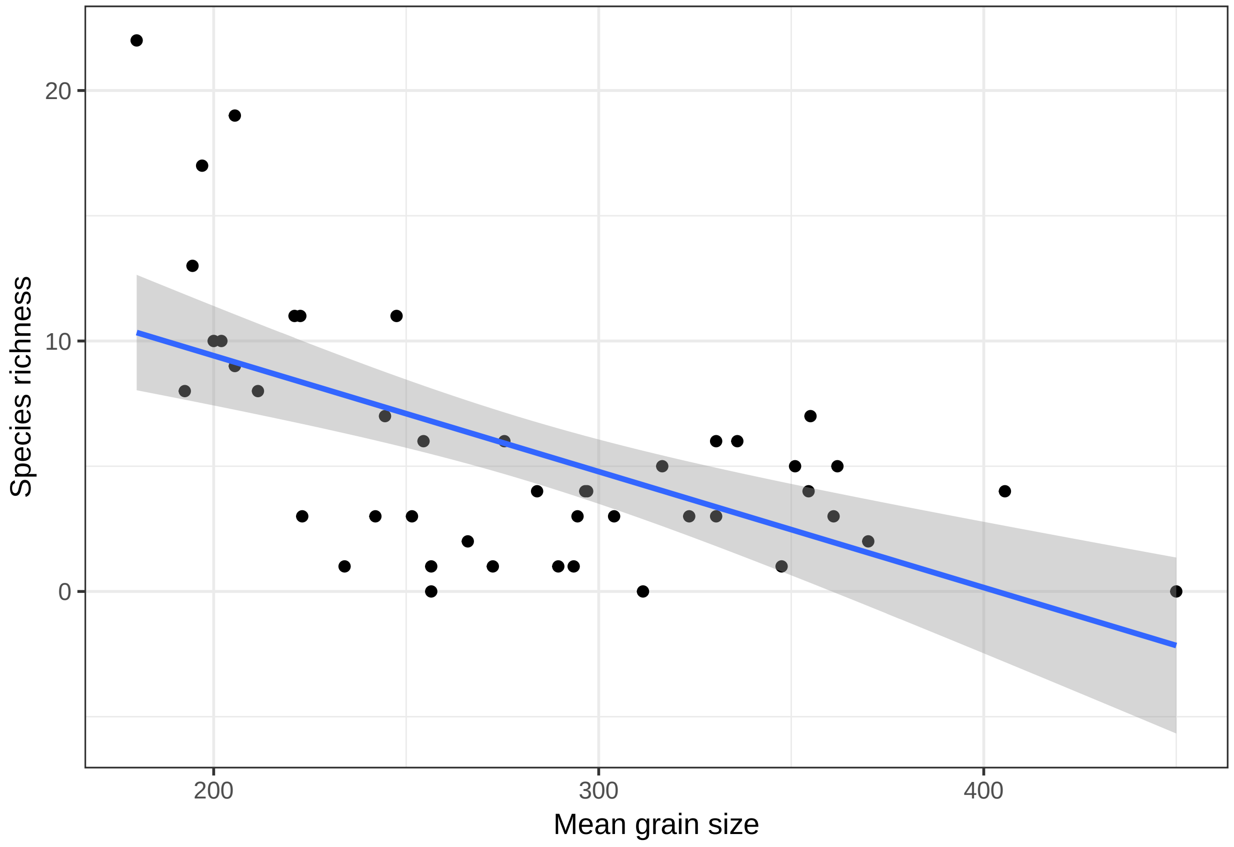
```
## Analysis of Variance Table
##
## Response: richness
##           Df Sum Sq Mean Sq F value    Pr(>F)
## grain      1 385.13  385.13  23.113 1.896e-05 ***
## Residuals 43 716.52   16.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
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
```

6.2.3 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")
```



6.2.4 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
```

6.2.5 Durbin Watson test

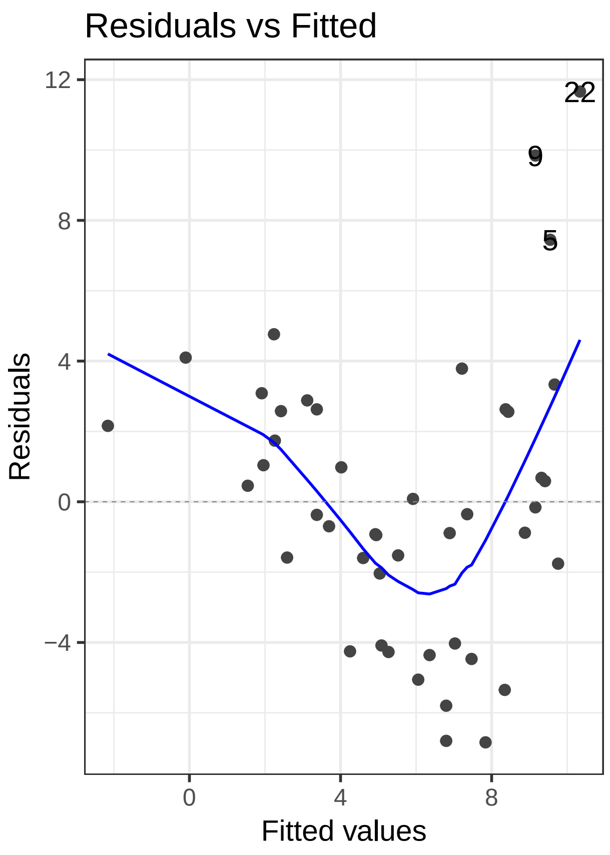
The Durbin Watson test which helps to confirm serial autocorrelation that may be the result of a misformed model will often also be significant when residuals cluster on one side of the line. In this case it was not, but this may be because there were too few data points.

```
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
```

6.2.6 Diagnostic plot after fitting linear model

```r
library(ggfortify)
theme_set(theme_bw())
autoplot(mod, which = 1)
```



6.3 Fitting quadratic model

6.3.1 Quadratic model fit

```r
mod2<-lm(data=d,richness~grain + I(grain^2))
```

6.3.2 Quadratic model anova and summary

```r
anova(mod2)
```

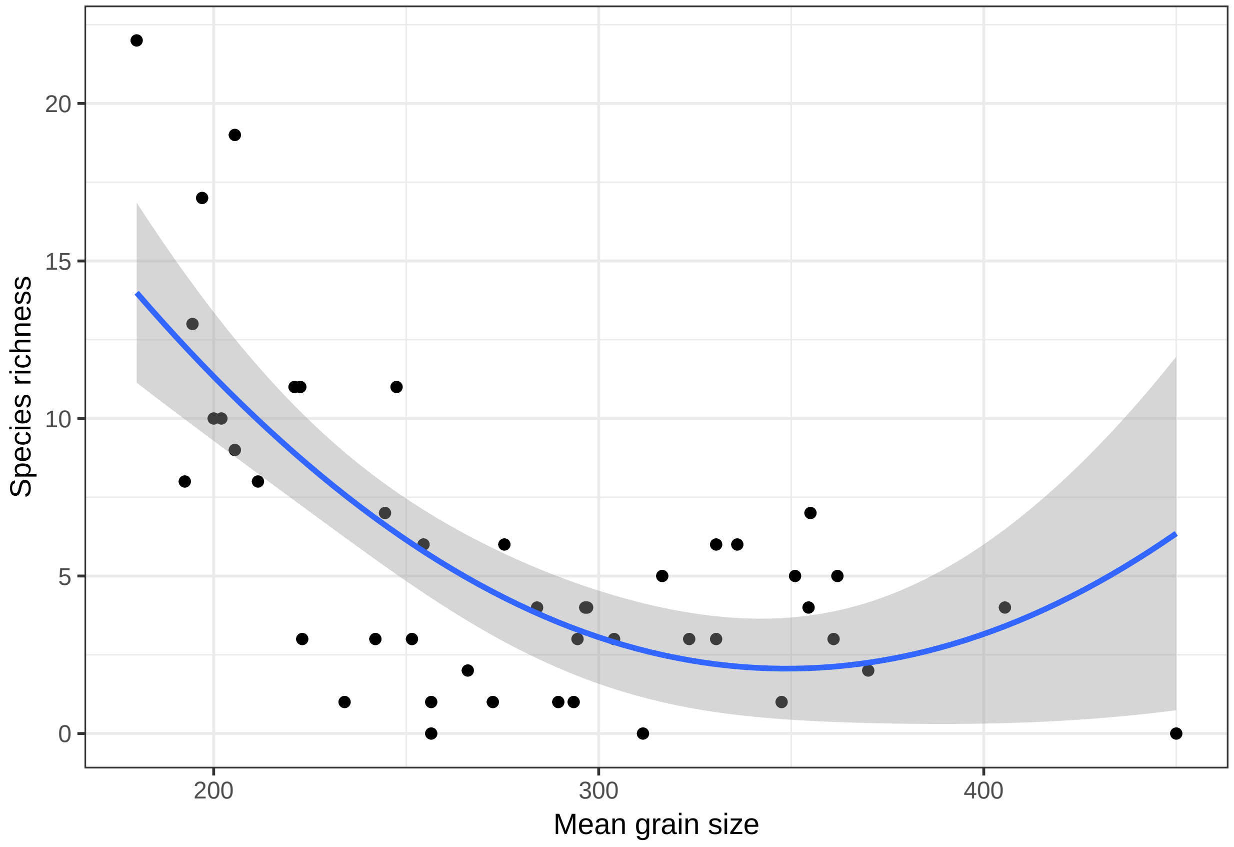
```
## Analysis of Variance Table
##
## Response: richness
##             Df Sum Sq Mean Sq F value    Pr(>F)
## grain        1 385.13  385.13  29.811 2.365e-06 ***
## I(grain^2)   1 173.93  173.93  13.463   0.00068 ***
## Residuals   42 542.59   12.92
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01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

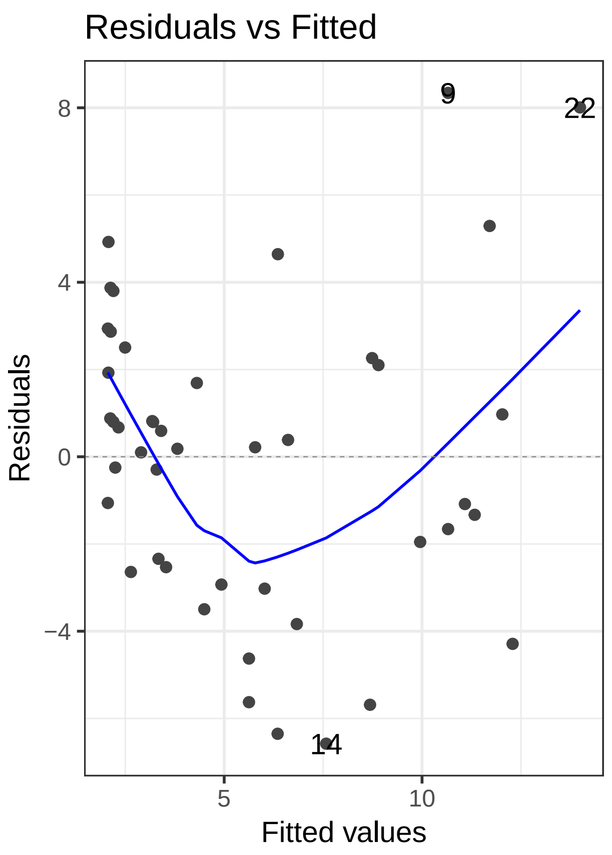
6.3.3 Quadratic model plot

```
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")
```



6.3.4 Diagnostic plot after fitting quadratic

```
library(ggfortify)
theme_set(theme_bw())
autoplot(mod2, which = 1)
```



6.3.5 Comparing fits

```r
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 7

Fitting splines using mgcv

7.1 Fiting model

```
library(mgcv)
mod3<-gam(data=d,richness~s(grain))
```

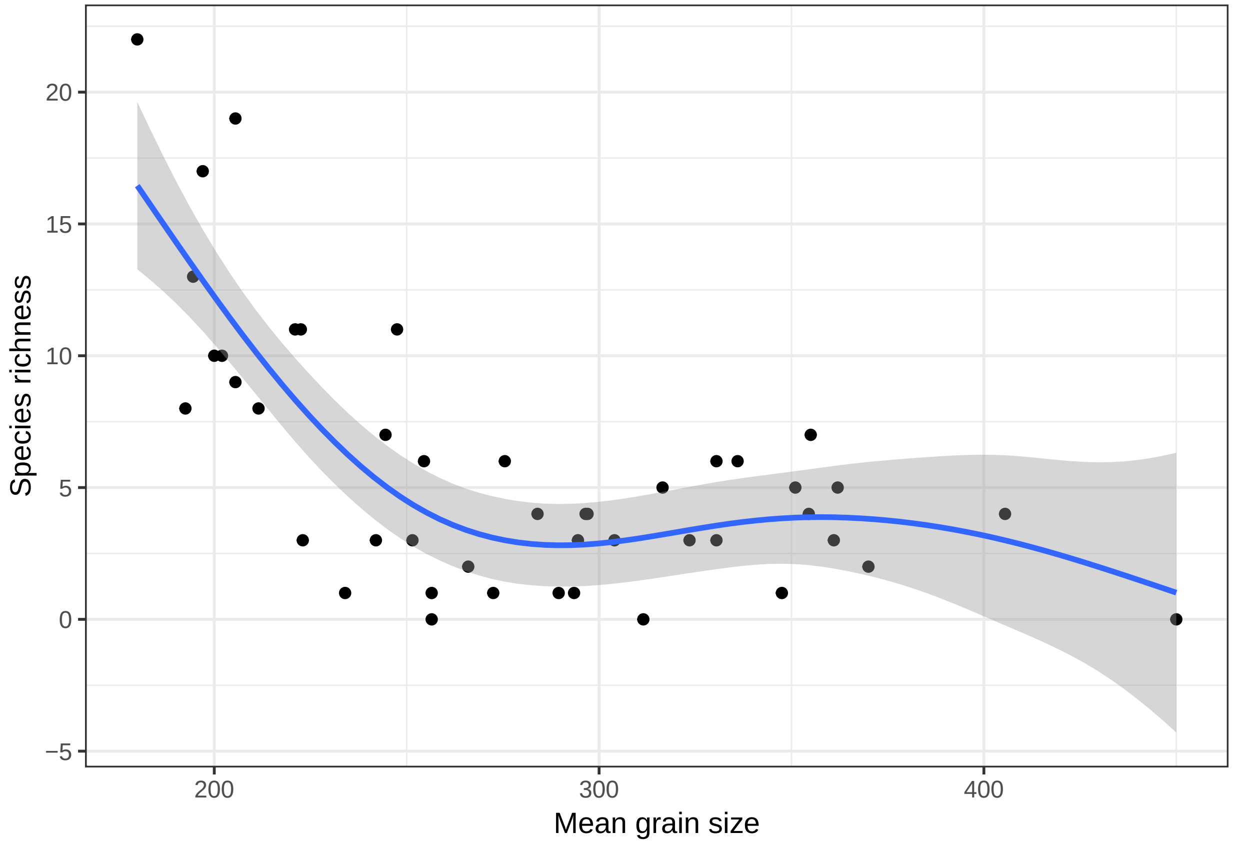
7.2 Summary model

```
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
```

7.3 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 8

Non linear model

8.1 Rectangular hyperbola of the Michaelis-Menten form

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

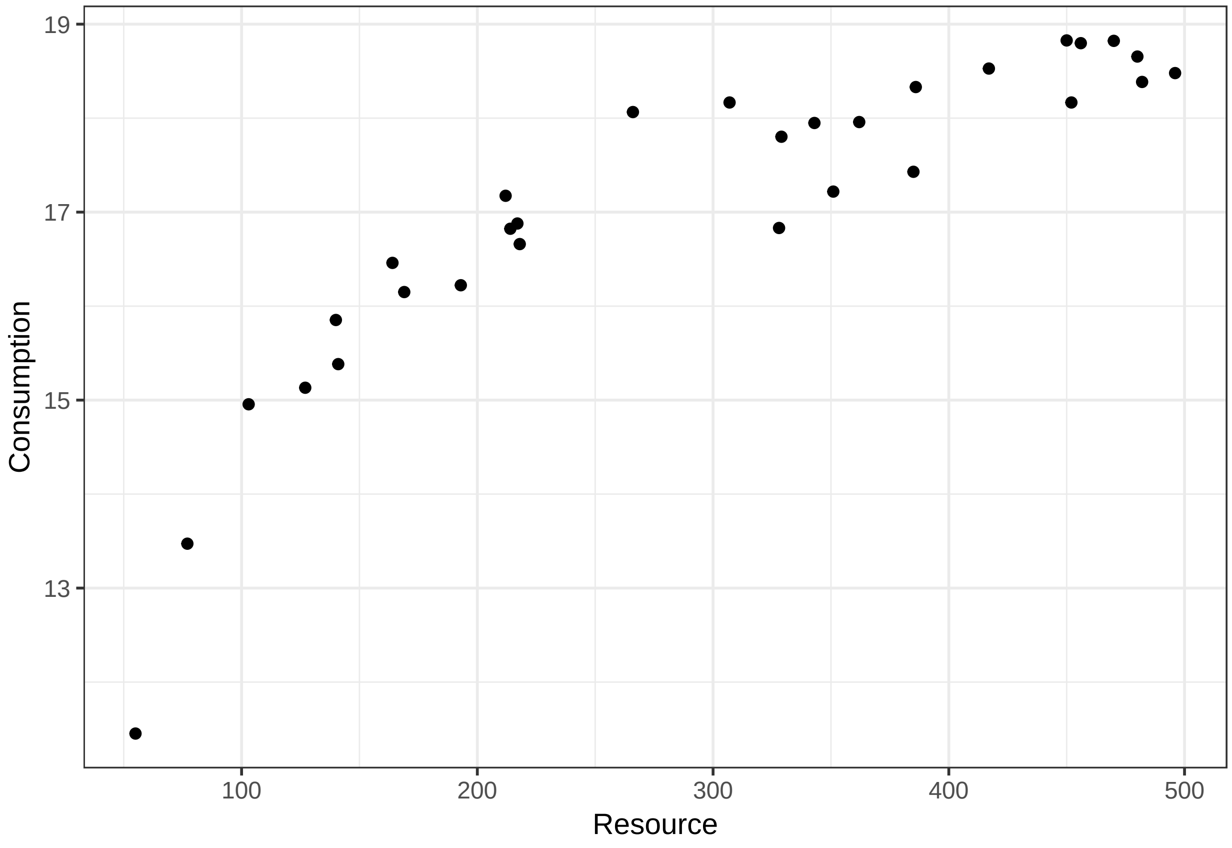
Where

- 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.

8.1.1 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
```

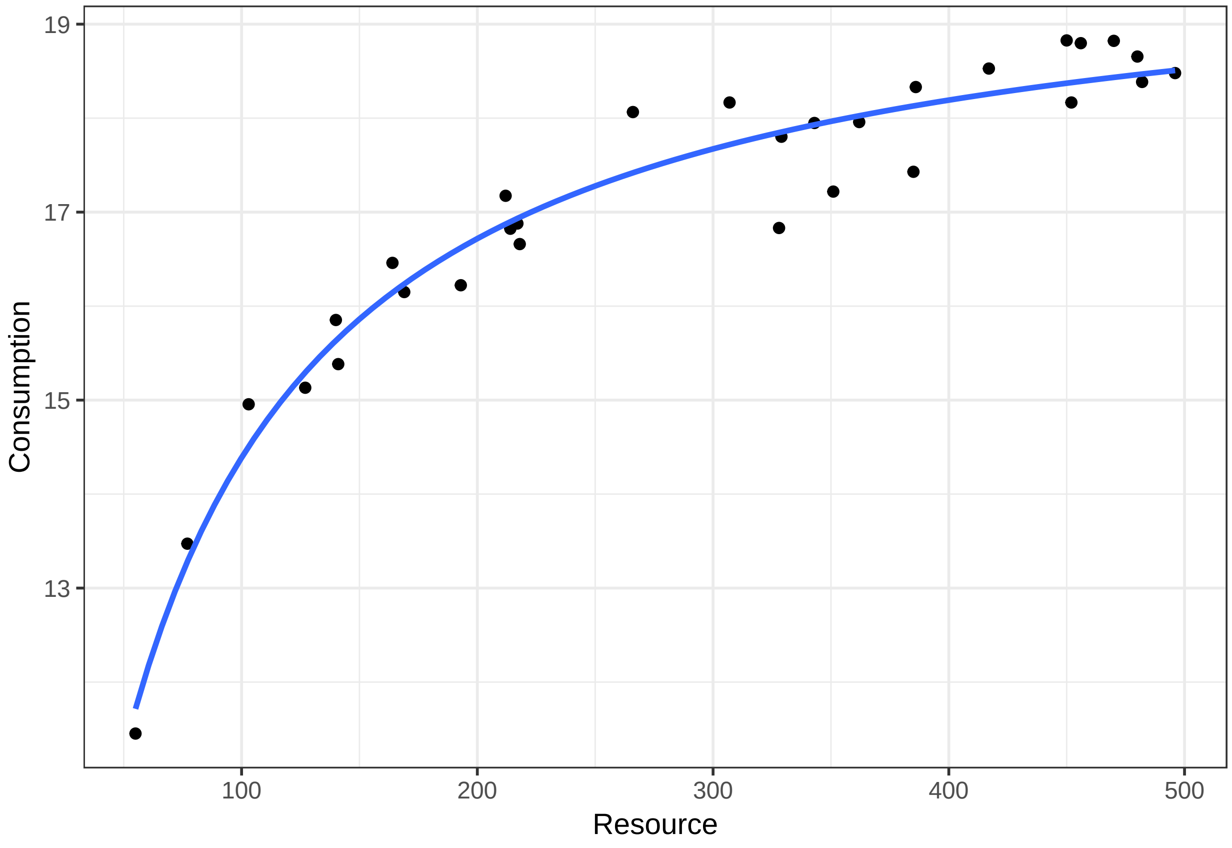


8.1.2 Fitting the model

```
nlmod<-nls(Consumption~s*Resource/(F+Resource),data = d,start = list( F = 20,s=20))
```

8.1.3 Curve fit

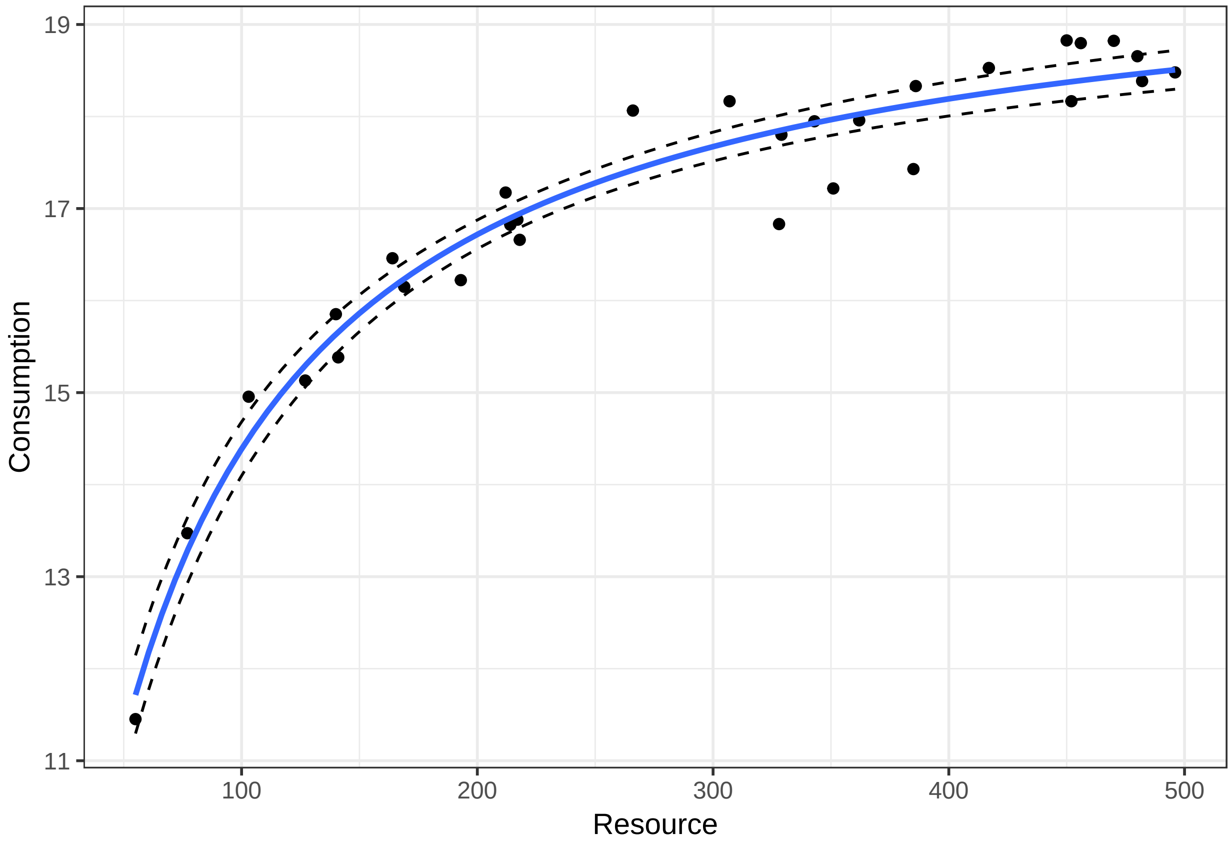
```
g0<-ggplot(data=d,aes(x=Resource,y=Consumption))
g1<-g0+geom_point()
g2<-g1+geom_smooth(method="nls",formula=y~s*x/(F+x),method.args=list(start = c( F = 20,s=20)), se=FALSE
g2
```



8.1.4 Curve fit with confidence intervals

This needs the propagate package.

```
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
```



8.2 Holling's disc equation

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

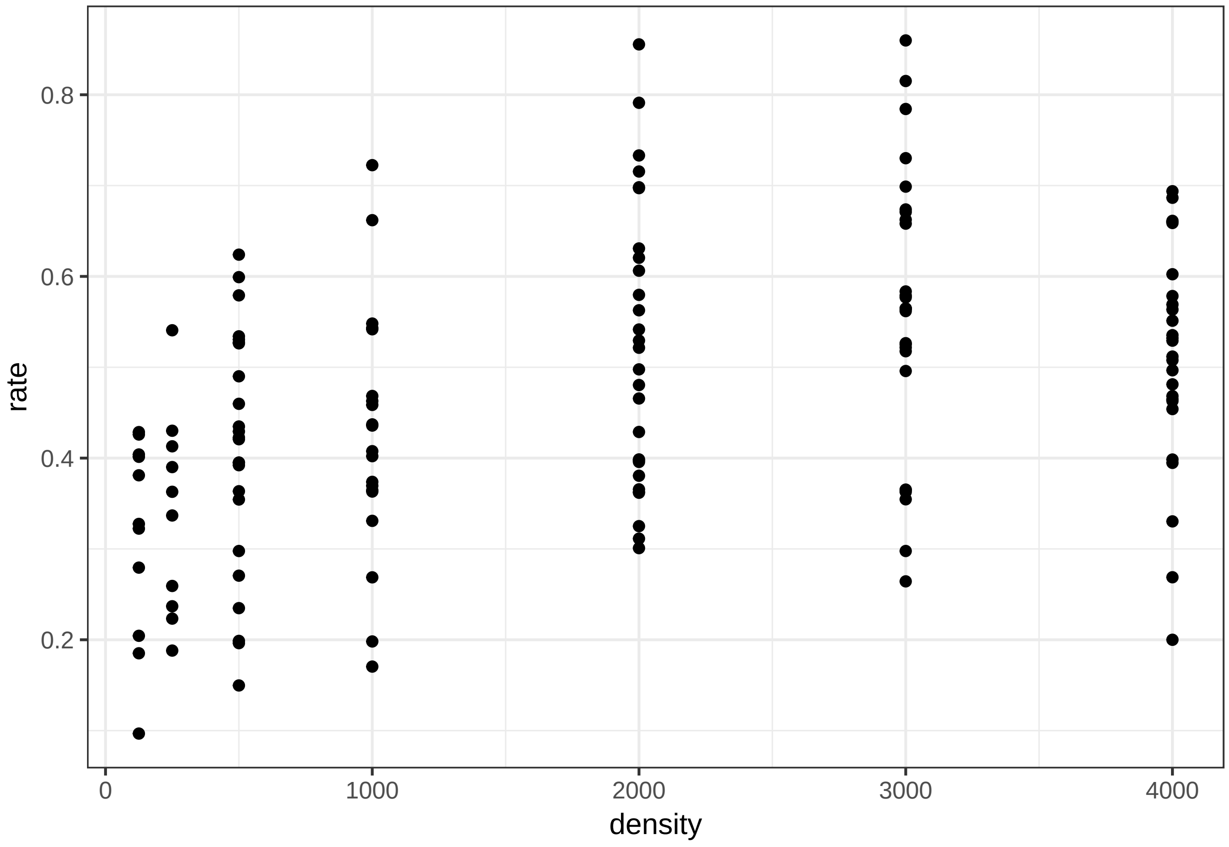
$R = \frac{aD}{1+aDH}$

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).

8.2.1 The data

```r
d<-read.csv("/home/aqm/course/data/buntings.csv")
g0<-ggplot(data=d,aes(x=density,y=rate)) + geom_point()
g0
```



8.2.2 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))
```

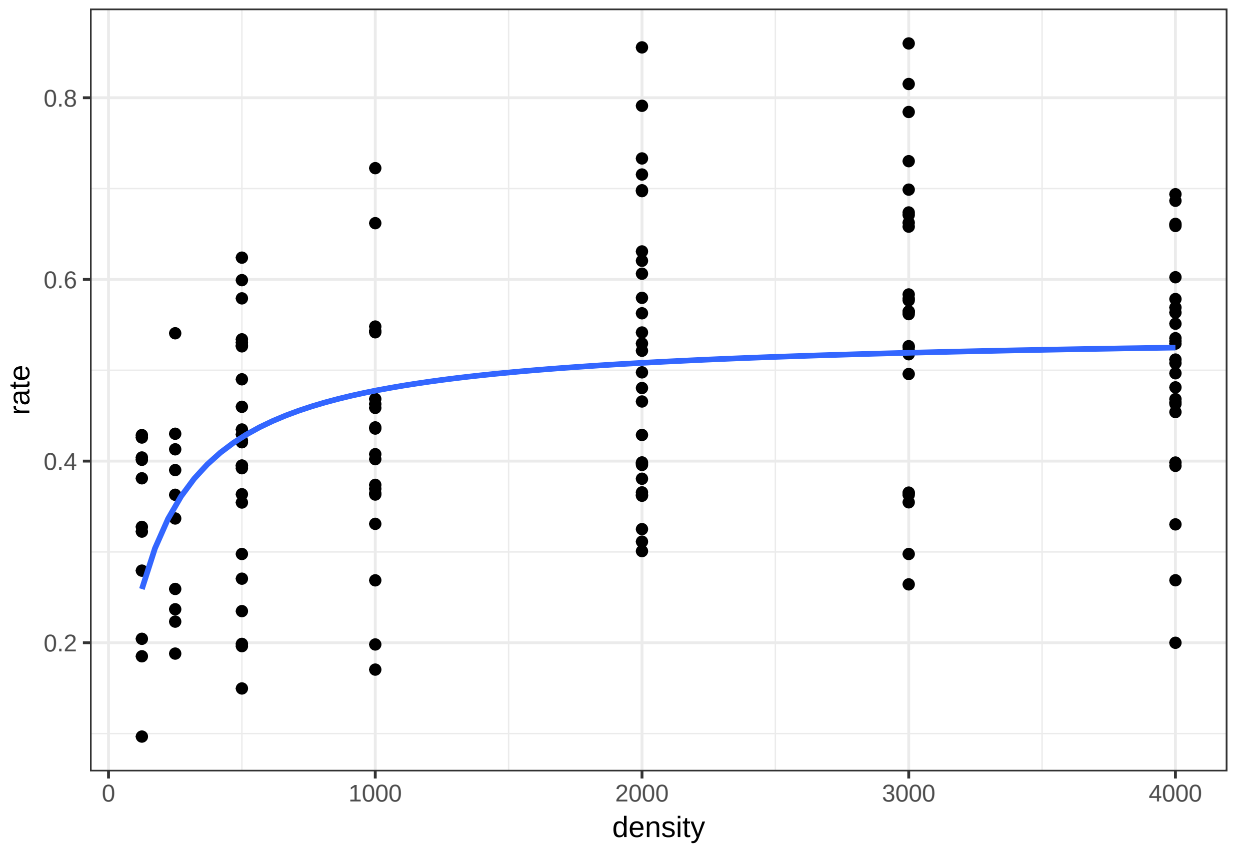
8.2.3 Confidence intervals for parameters

```
confint(HDmod)
```

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

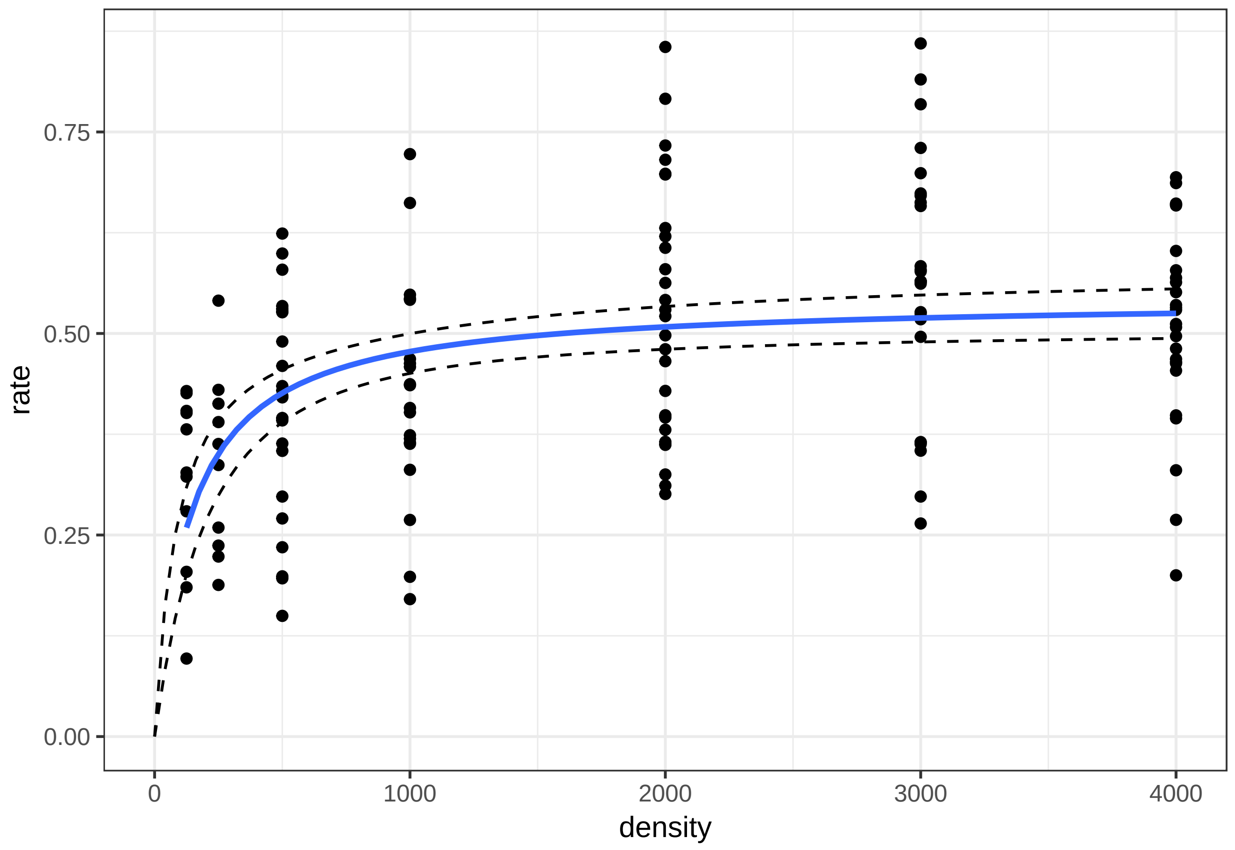
8.2.4 Plot with fitted curve

```
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
```



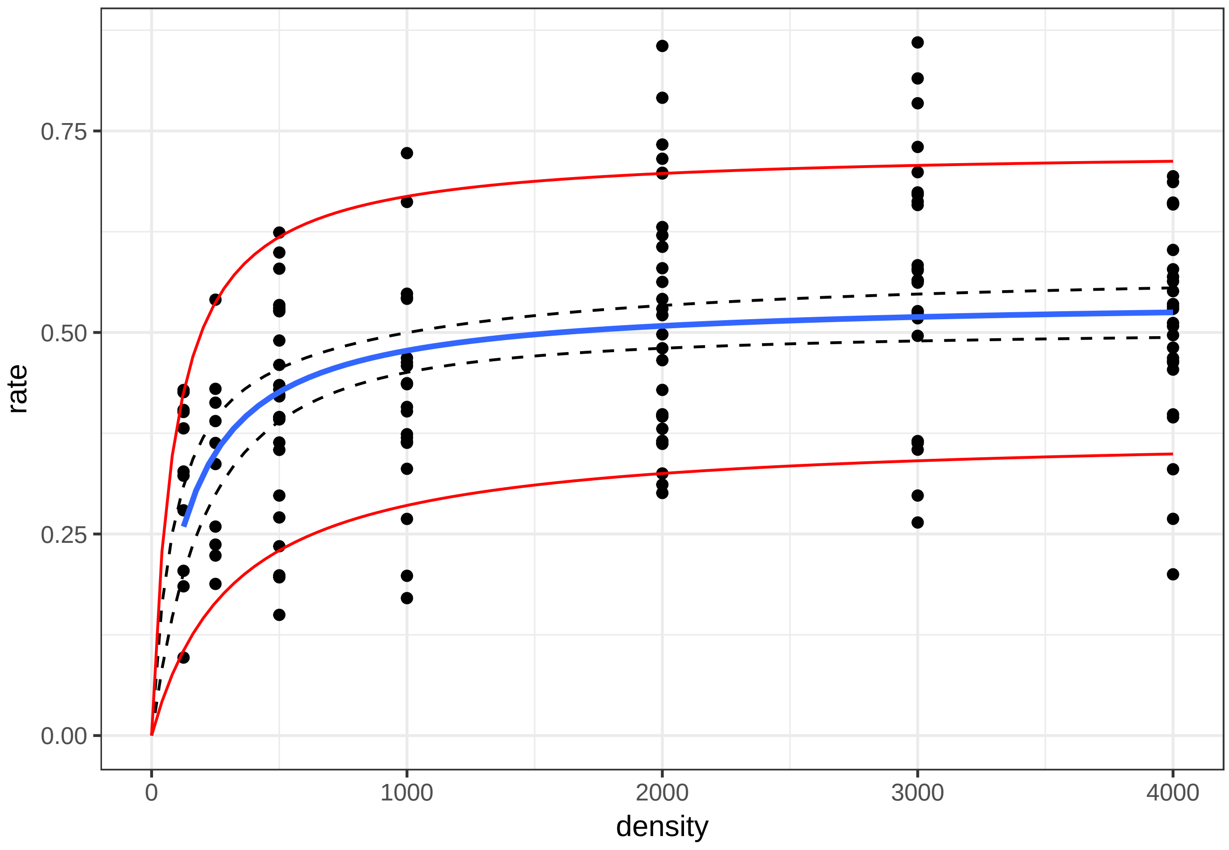
8.2.5 Plot with confidence intervals

```
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
```



8.2.6 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 9

Analysis of covariance and mixed effects models

9.1 Analysis of covariance

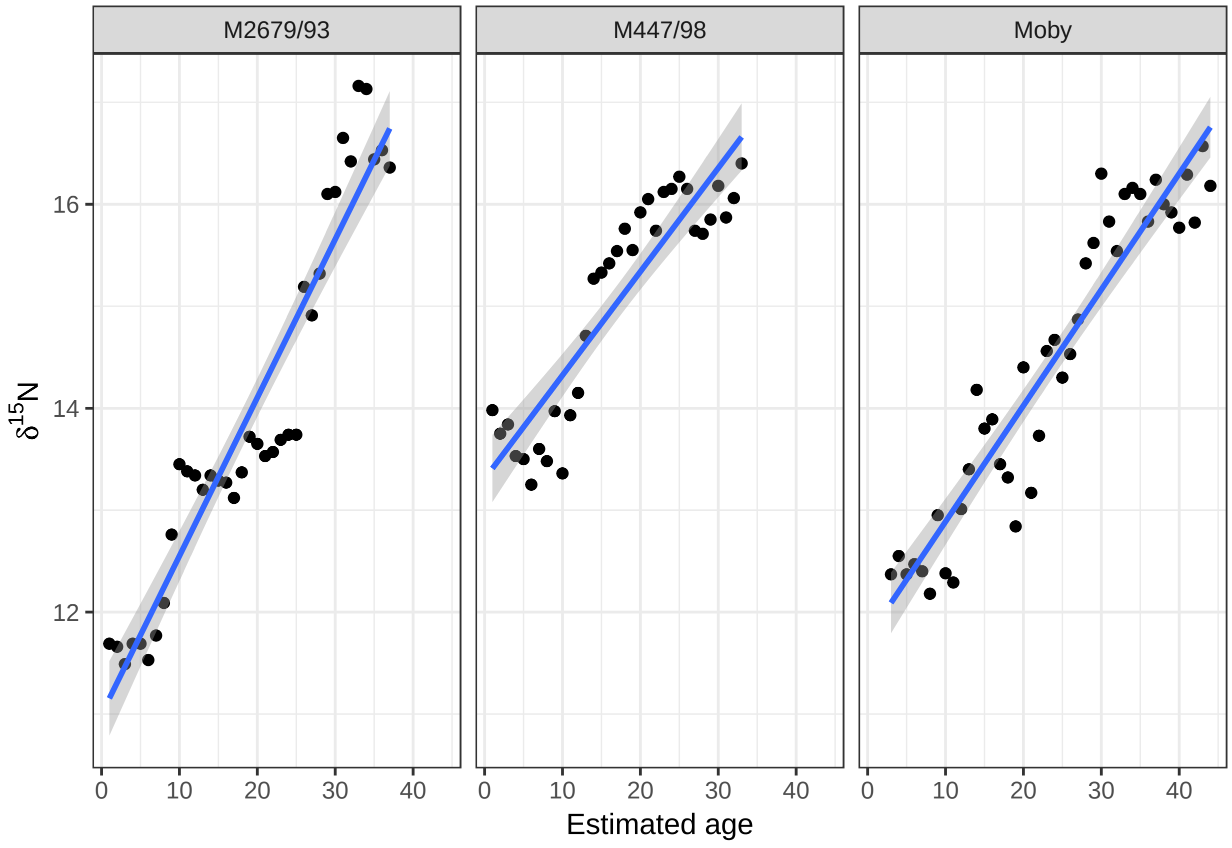
9.1.1 Data

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
```

9.1.2 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")
```



9.1.3 Analysis of covariance

A significant interaction shows that the slope of a linea model differs between individuals

```
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
```

9.2 Linear mixed effects

9.2.1 Intercept only

```
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
```

9.2.2 Slope model

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

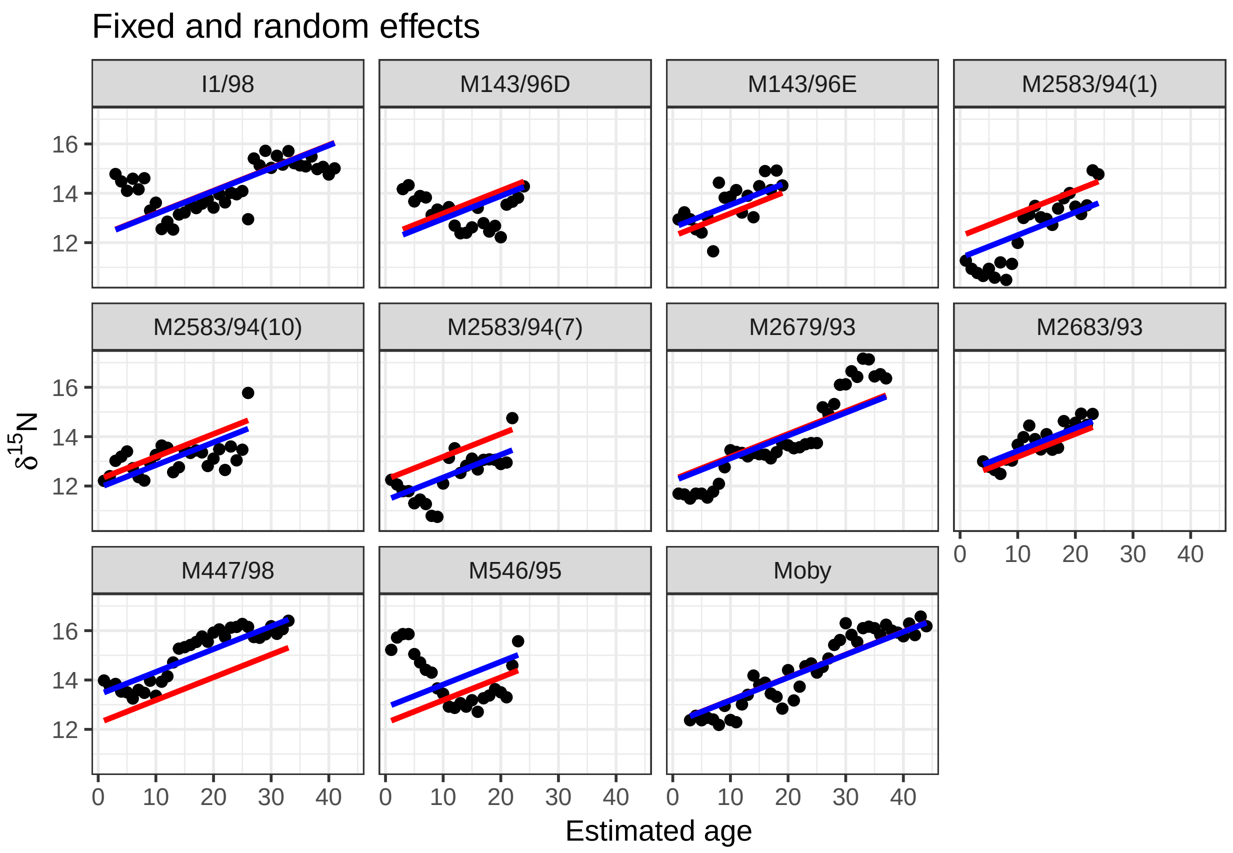
```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: X15N ~ Age + (Age | Whale)
##    Data: Whales
## REML criterion at convergence: 657.3864
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  Whale    (Intercept) 1.3138
##           Age         0.0734   -0.90
##  Residual             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
```

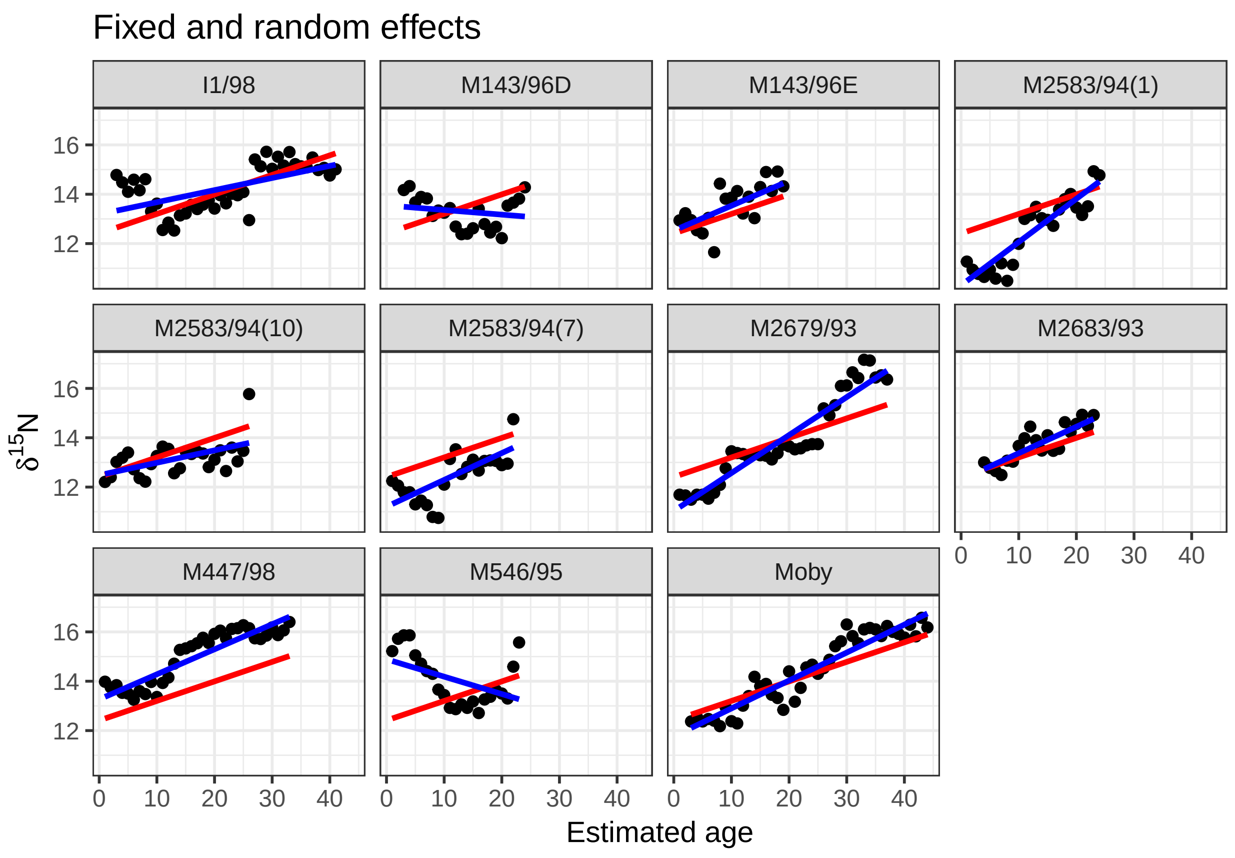
9.2.3 Plot intercept model

```r
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")
```



9.2.4 Plot slope model

```r
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")
```



9.3 GAMM model

9.3.1 Fit GAMM model

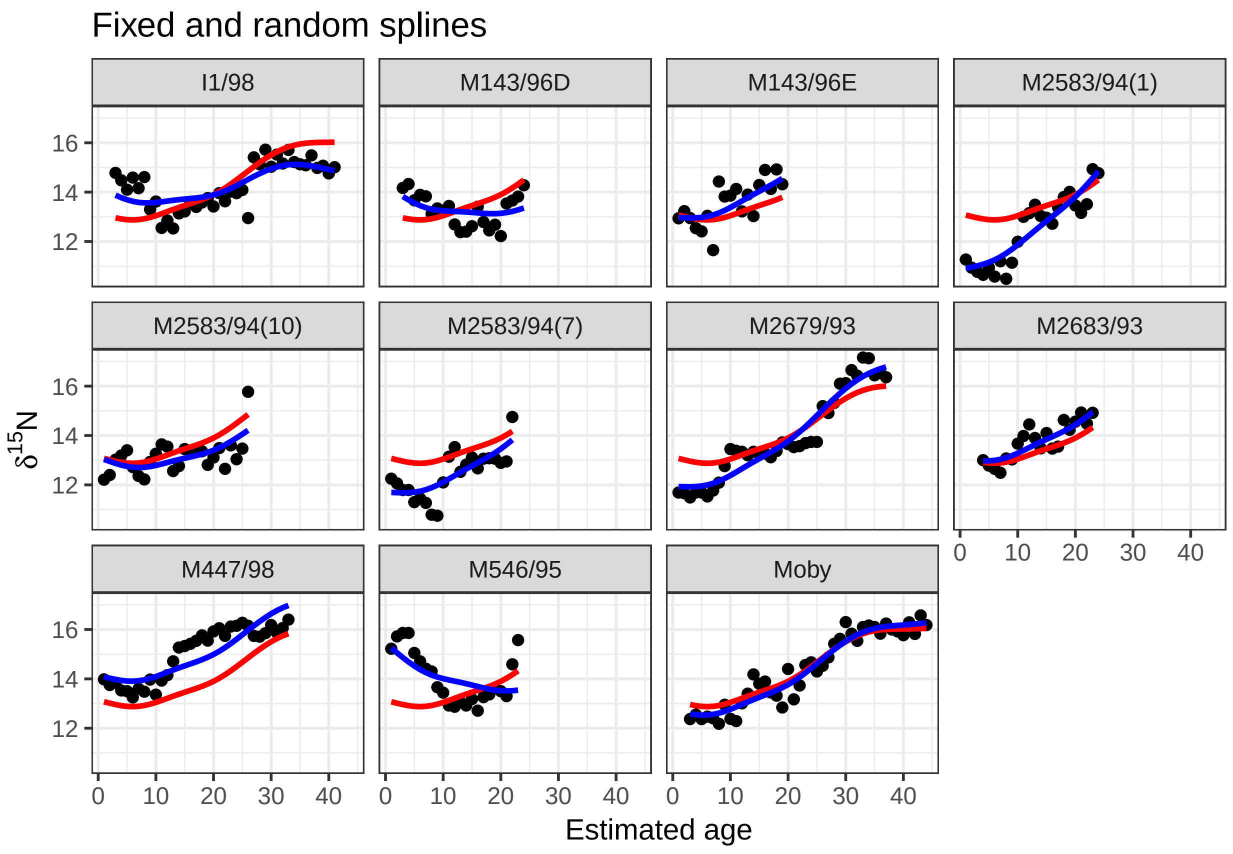
```
library(gamm4)
gamm.mod<-gamm4(X15N~s(Age),data=Whales,random = ~ (Age|Whale))
gamm.mod
```

```
## $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
```

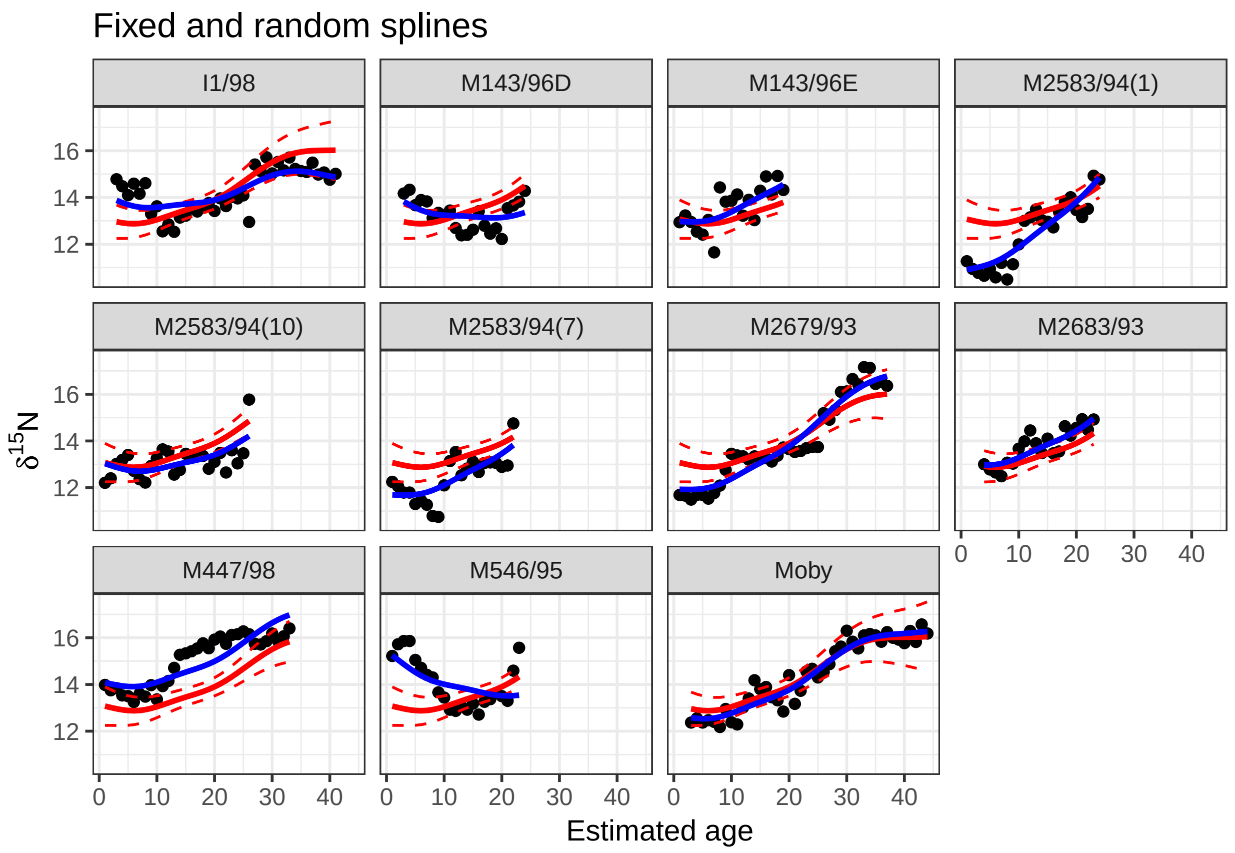
9.3.2 Plot GAMM model

```
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")
```



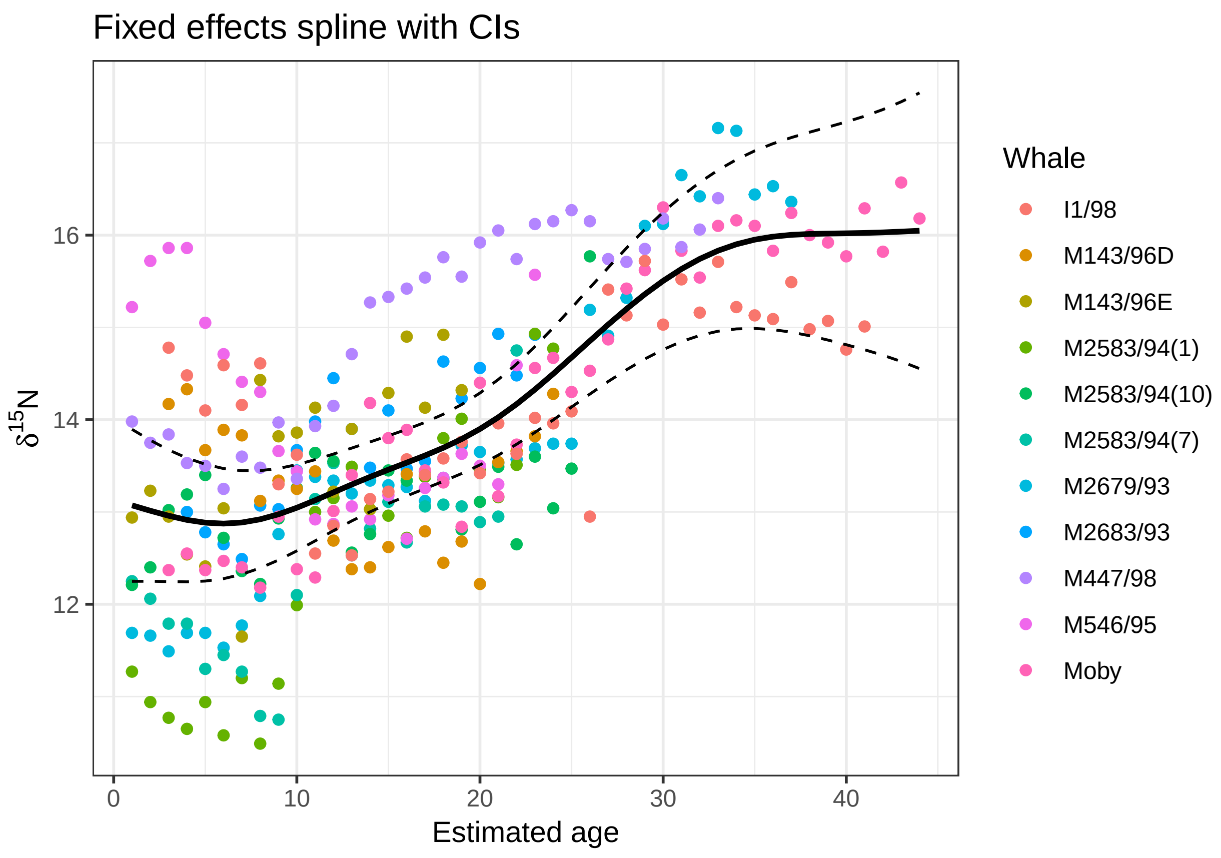
9.3.3 Plot GAMM with CIs

```
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")
```



9.3.4 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
$\delta^{15}N$
16
14
12
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 10

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.

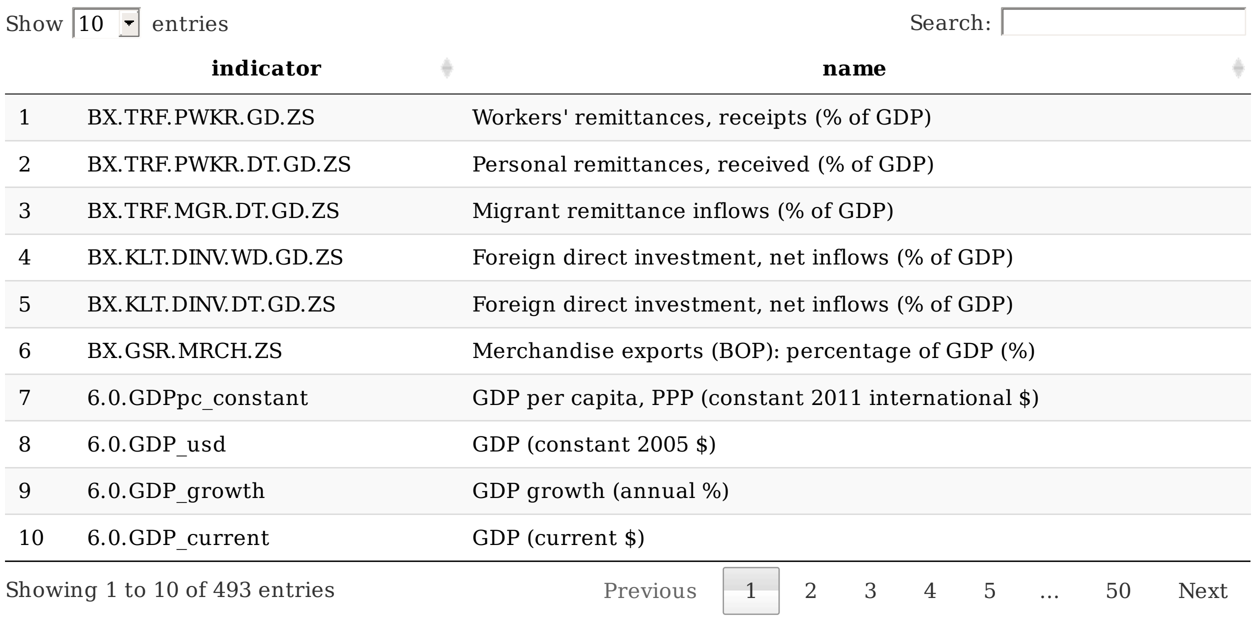
10.1 Getting the data

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.

```r
library(WDI)
library(dplyr)
library(ggplot2)
library(plotly)
library(ggthemes)
```

The field names and their meanings can be searched for in the table below.

```r
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

10.2 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.KI
names(d)[4:8]<- c("GDP_per_capita", "Population_total", "Life_expectancy","Energy_intensity_MJ_GDP","Ag
d %>% filter(region !="Aggregates") -> d ## Filter out the aggregated totals
```

10.3 Forming bubble plots

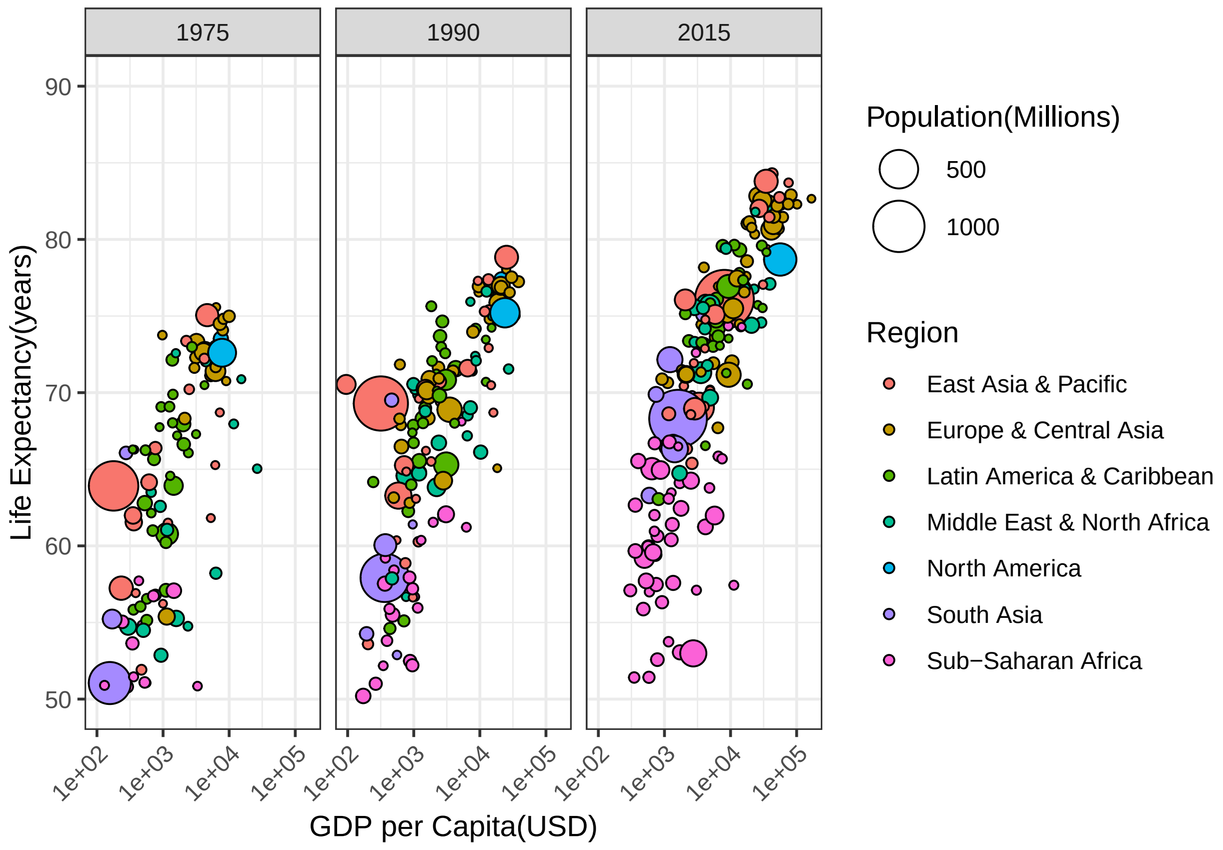
Han's Roslin's data animations effectively showed four dimensions of data in one. A third dimension was added to the scatter-plots in the form of the size of the bubble. In most of the plots this was the population size. The animation was over time, adding a fourth dimension. In order to produce a static 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")

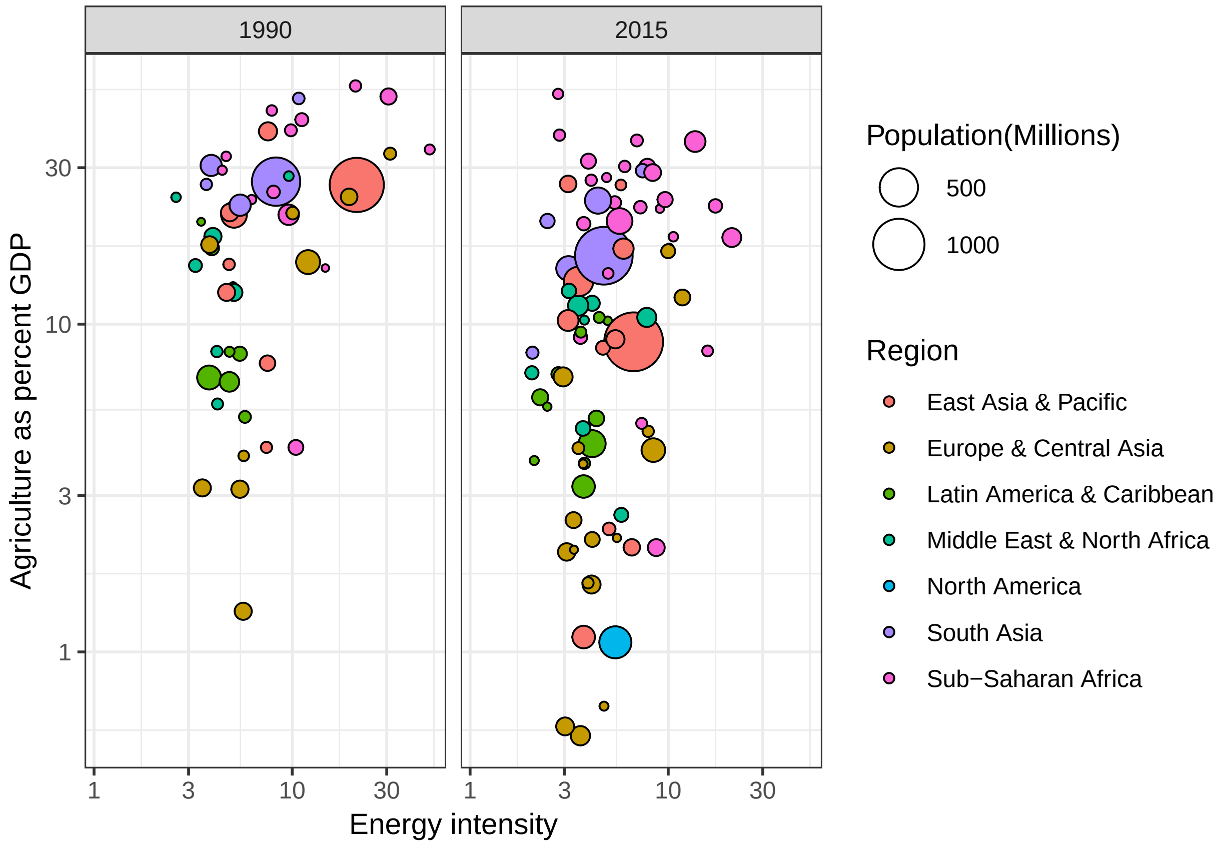
```
d %>% filter(year%in% c(1975,1990,2015)) %>% ggplot( aes(x = GDP_per_capita, y = Life_expectancy, size =
  geom_point(shape = 21) +
  labs(x = "GDP per Capita(USD)", y = "Life Expectancy(years)") +
  scale_y_continuous(limits = c(50, 90)) +
  scale_size(range = c(1, 10)) +
  labs(size = "Population(Millions)", 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.

10.4 Another example

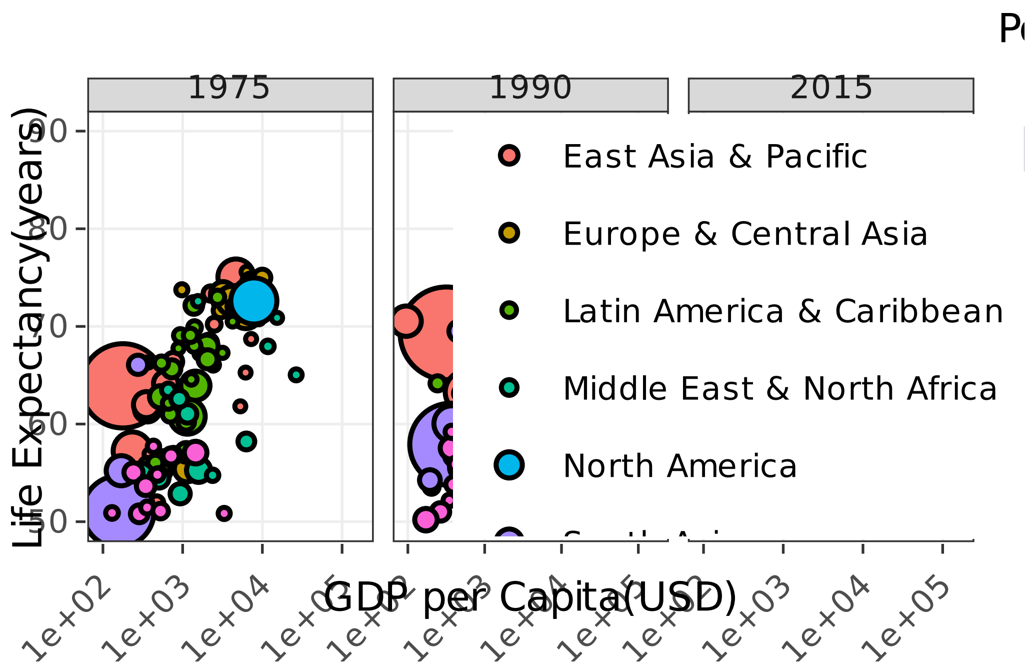
```
d %>% filter(year%in% c(1990,2015)) %>% filter(Population_total> 10000000) %>% ggplot( aes(x = Energy_i
  geom_point(shape = 21) +
  labs(x = "Energy intensity", y = "Agriculture as percent GDP") +
  scale_size(range = c(1, 10)) + scale_y_continuous(limits = c(1, 50)) +
  labs(size = "Population(Millions)", fill = "Region")  + theme_bw() +facet_wrap("year") + scale_x_log1
g2
```



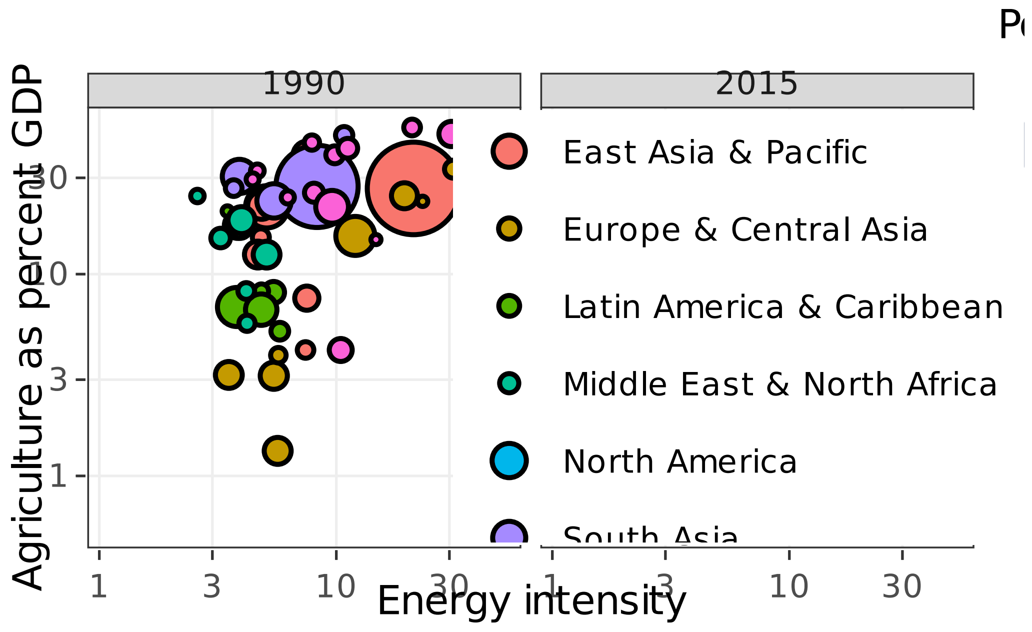
10.5 Using ggplotly

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.

```
ggplotly(g1)
```


Po
1975
1990
2015
East Asia & Pacific
Europe & Central Asia
Latin America & Caribbean
Middle East & North Africa
North America
Life Expectancy (years)
GDP per Capita (USD)
1e+02
1e+03
1e+04
1e+05

```
ggplotly(g2)
```


Po
1990
2015
East Asia & Pacific
Europe & Central Asia
Latin America & Caribbean
Middle East & North Africa
North America
South Asia
Agriculture as percent GDP
Energy intensity
1
3
10
30