ESMA 3102: Introduction to Statistics II

Dr. Wolfgang Rolke

August 12, 2018

Contents

0.1 Syllabus

Professor: Dr. Wolfgang Rolke

The web address is http://academic.uprm.edu/wrolke/esma3102.

The official prontuario for the course is available from the usual site. If there is any difference between the prontuario and the information on the webpage use the webpage.

Time and Place:

Tuesday, Thursday 12:30-1:45pm SH005

Textbook: Statistics, Informed Decisions using Data, Michael Sullivan (**highly recommended but not required**)

Office hours:

Tuesday, Thursday OF407 12-12:30, 3:15-3:45 Wednesday 1-4:00pm via email

email: wolfgang[dot]rolke[at]upr[dot]edu
when you send me an email ALWAYS start the subject line with ESMA3102

Grading:

1. Quizzes: 35%
2. Partial Exams 35%
3. Final 30%

All quizzes and exams will be done using moodle. To get to the quizzes go to https://ecourses.uprm.edu/, log on with your UPR ID and password. The first time use the enrollment key: Esma 3102 - 066.

output: html_document: default pdf_document: fig_caption: no —

1 Using the Computer and R

This page contains some basic information on how to use the computer and the R program.

To log on to computers in Ch115:

Username: .\esma (important: do not forget to include ". " before the word esma)
Password: Mate1234 (important: uppercase letter"M")

To log on to computers in SH005:

Username: Estudiante
Password: salon005

The class webpages are at http://academic.uprm.edu/wrolke/esmaXXXX (3015, 3101, 3102, 6661 etc)

At the end of each session log off

1.0.1 General Info

You can get a free version of R for your computer from a number of sources. The download is about 70MB and setup is fully automatic. Here are some links:

Windows

MacOS

After the installation is finished close R (if it is open). From now on ALWAYS open R by clicking on the link to to the RESMA3 file on top of the homepage. You can also download and save that file to your own computer and start R from there. The first time you do this the program will download a number of additional stuff, just let it. Also a window might pop up and ask whether to save something, if so click on yes.

Note

- You might be asked at several times whether you want to do something (allow access, run a program, save a library, . . .), always just say yes!
- You will need to connect to a reasonably fast internet for these steps.
- This will take a few minutes, just wait until the > sign appears.

FOR MAC OS USERS ONLY

There are a few things that are different from MacOS and Windows. Here is one thing you should do:

Download XQuartz - XQuartz-2.7.11.dmg
Open XQuartz
Type the letter R (to make XQuartz run R)

Hit enter Open R Run the command .First()
Then, every command should work correctly.

1.0.2 RStudio

there is a program called RStudio that a lot of people like to use to run R. You can download it at RStudio. Before you can use RStudio with Resma3 you need to run Resma3 JUST ONCE from R itself.

So do this

1) follow **ALL** the instructions above
2) only if everything is running correctly install RStudio.

For the purpose of the class R itself is enough, we don't need RStudio.

1.0.3 Troubleshooting

if you try to run a command and get an error

could not find function "ggplot"

(or grid or shiny)

first try this: run the command

```
ls()
```

You should see a listing of many things (over 200). If you do not Resma3 did not load correctly. Close R and restart it by clicking on the link to Resma3 on the homepage.

If you do see the listing, type

```
one.time.setup()
```

A number of things should be happening, just wait until you see the > again and see whether that fixes the problem.

If this does not work turn off R and restart it with a new version of Resma3 from the top of the class homepage.

If this also does not work send me an email with the explanation of the problem. The best thing to do is to include a screenshot. Here is how:

Windows

MacOS

You can also just use your cell phone to take a picture of the screen, but make sure it is is readable!

I often get an email saying that something is not working, and my answer is simply:

RGDM

this means: **R**ead the **G**od-**D**amn **M**anual!

that is the answer to your problem is somewhere on these pages, and you should have found it there before sending an email!

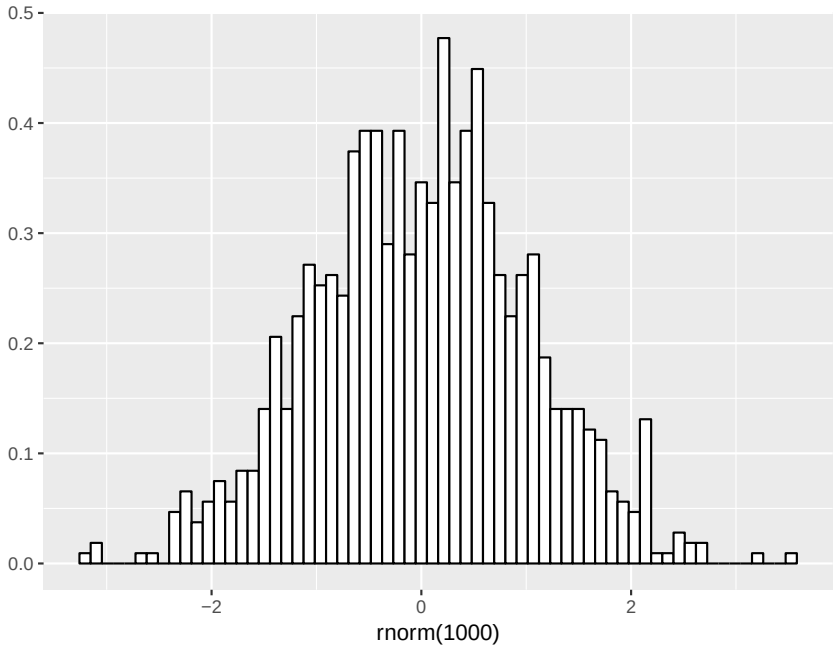
Throughout this class when you see something like this:

```
text
```

it means commands you should type (or copy-paste) into R.

To see whether everything is installed correctly copy-paste the following line into R and hit enter:

```
hplot(rnorm(1000))
```



You should see a graph like this (called a histogram)

For a much more extensive introduction to R go here

Once you have started a session the first thing you see is some text, and then the > sign. This is the **R prompt**, it means R is waiting for you to do something. Sometimes the prompt changes to a different symbol, as we will see.

Let's start with

```
ls()
```

shows you a "listing"" of the files (data, routines etc.)

If you have worked for a while you might have things you need to save, do that by clicking on

File > Save Workspace

If you quit the program without saving your stuff everything you did will be lost. R has a somewhat unusual file system, everything belonging to the same project (data, routines, graphs etc.) are stored in just one file, with the extension .RData.

To quit R, type

```
q()
```

or click the x in the upper right corner.

R has a nice recall feature, using the up and down arrow keys. Also, typing

```
history()
```

shows you the most recent things entered.

R is case-sensitive, so a and A are two different things.

Often during a session you create objects that you need only for a short time. When you no longer need them use **rm** to get rid of them:

```
x <- 10
x^2
```

```
## [1] 100
```

```
rm(x)
```

the **<-** is the *assignment* character in R, it assigns what is on the right to the symbol on the left.

1.1 Data Entry

1.1.1 With the keyboard

For a few numbers the easiest thing is to just type them in:

```
x <-  c(10, 2, 6, 9)
x
```

```
## [1] 10  2  6  9
```

c() is a function that takes the objects inside the () and **c**ombines them into one single object (a vector).

1.1.2 idataio

This section can be left out unless data i/o is discussed in class

idataio won't run in CH115 until computers have been updated

We have data on the age and the position of people. So there were 10 old people in the first position, and so on:

Age	First	Second	Third
Old	10	16	21
Young	15	12	26

To get this into R use the the routine **idataio**.

CAREFUL: **idataio** currently does not work in CH115 because of old version of R!

It can be used to enter the values directly from the keyboard, a table that was copied to the clipboard or read it from a file like an excel worksheet.

Say we want to get the table above into R. Here are three ways to do this using idataio:

```
x <- idataio()
```

this will bring up the browser with a spreadsheet and you can just enter the values. Change Number of Cases to 2 and Number of Variables to 4. Type the column names (Age First Second Third) in the box on the right and enter the values in the spreadsheet. Click on the button Close App to return to R.

2) use the mouse to highlight the whole table, switch to R and run

```
x <- idataio()
```

select the Copy from Clipboard option. Change Number of Variables to 4. Highlight the table in the browser and right-click Copy. Hit Go! and see whether the table appears correctly. If not maybe you need to play around a bit with the Number ofr cases etc. When it is ok hit the Close App button on top.

copying from an Excel worksheet works exactly the same way.

NOTE : the current version does not allow for empty cells. If there are any enter NA first. Also any names can not include spaces.

3) Open Microsoft Excel and enter the info as usual. Save the file as an excel spreadsheet (with the xlsx extension). Now run idataio and choose the Read data from file option.

1.1.3 Getting Data from Moodle Quizzes

Most moodle quizzes will require you to transfer data from the quiz to R. This is done with the command *get.moodle.data()*. There are two steps:

- in moodle use the mouse to highlight the data. If it is a table with several columns ALWAYS include the column headers (names of variables).
- switch to R and run

```
get.moodle.data()
```

Now the data should be in R. It is called x. You can always check by typing x and ENTER.

```
x
```

```
## [1] 10  2  6  9
```

Here are some examples:

a) single set of numbers:

101.6 115.0 100.9 103.8 77.6 102.6 99.6 108.5 100.8 92.5 101.8 81.6 103.7 94.9 103.3 86.7 101.6 106.6 101.5 96.9

highlight the data with the mouse, copy it, go to R and type

```
get.moodle.data()
x
```

```
##  [1] 101.6 115.0 100.9 103.8  77.6 102.6  99.6 108.5 100.8  92.5 101.8
## [12]  81.6 103.7  94.9 103.3  86.7 101.6 106.6 101.5  96.9
```

this also works if the data is not numbers:

Old Old Young Old Young Young

```
get.moodle.data()
```

```
## [1] "Old"   "Old"   "Young" "Old"   "Young" "Young"
```

sometimes parts of the data are separated by some symbol, for example a comma. In that case you can use the *sep* argument:

1.5, 2.3, 5.3, 2.4, 7.9, 8.1, 2.7, 4.2

```
get.moodle.data(sep = ",")
```

```
## [1] 1.5 2.3 5.3 2.4 7.9 8.1 2.7 4.2
```

b) data is in the form of a table with several columns:

Age	Gender
25	Female
21	Male
18	Male
20	Female
18	Male
21	Female
18	Male
22	Male
18	Female
25	Female

```
get.moodle.data()
```

```
##    Age Gender
## 1   25 Female
## 2   21   Male
```

```
## 3   18   Male
## 4   20 Female
## 5   18   Male
## 6   21 Female
## 7   18   Male
## 8   22   Male
## 9   18 Female
## 10  25 Female
```

Note if the data is a single vector it is given the name x, and you can now do things like

```
mean(x)
```

if the data is a table it is immediately attached and you can use the column names, for example

```
mean(Age)
```

Note on rare occasions the routine can fail if the data is a table but everyting is text. In that case use the argument is.table=TRUE.

Note sometimes you might get a warning from R, as long as the data is transfered correctly you can ignore that.

1.1.4 Data Types in R

the most basic type of data in R is a **vector**, simply a list of values.

Say we want the numbers 1.5, 3.6, 5.1 and 4.0 in an R vector called x, then we can type

```
x <- c(1.5, 3.6, 5.1, 4.0)
x
```

```
## [1] 1.5 3.6 5.1 4.0
```

Often the numbers have a structure one can make use of:

```
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
10:1
```

```
##  [1] 10  9  8  7  6  5  4  3  2  1
```

```
1:20*2
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
```

```
c(1:10, 1:10*2)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10  2  4  6  8 10 12 14 16 18 20
```

Sometimes you need parentheses:

```
n <- 10
1:n-1
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

```
1:(n-1)
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

The *rep* ("repeat") command is very useful:

```
rep(1, 10)
```

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

```
rep(1:3, 10)
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
rep(1:3, each=3)
```

```
## [1] 1 1 1 2 2 2 3 3 3
```

```
rep(c("A", "B", "C"), c(4,7,3))
```

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

what does this do?

```
rep(1:10, 1:10)
```

1.1.4.1 Commands for Vectors

To find out how many elements a vector has use the *length* command:

```
x <- c(1.4, 5.1, 2.0, 6.8, 3.5, 2.1, 5.6, 3.3, 6.9, 1.1)
length(x)
```

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

The elements of a vector are accessed with the bracket [ ] notation:

```
x[3]
```

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

```
x[1:3]
```

```
## [1] 1.4 5.1 2.0
```

```
x[c(1, 3, 8)]
```

```
## [1] 1.4 2.0 3.3
```

```
x[-3]
```

```
## [1] 1.4 5.1 6.8 3.5 2.1 5.6 3.3 6.9 1.1
```

```
x[-c(1, 2, 5)]
```

```
## [1] 2.0 6.8 2.1 5.6 3.3 6.9 1.1
```

Instead of numbers a vector can also consist of characters (letters, numbers, symbols etc.) These are identified by quotes:

```
c("A", "B", 7, "%")
```

```
## [1] "A" "B" "7" "%"
```

A vector is either numeric or character, but never both (see how the 7 was changed to "7").

You can turn one into the other (if possible) as follows:

```
x <- 1:10
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
as.character(x)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```
x <- c("1", "5", "10", "-3")
x
```

```
## [1] "1"  "5"  "10" "-3"
```

```
as.numeric(x)
```

```
## [1]  1  5 10 -3
```

A third type of data is logical, with values either TRUE or FALSE.

```
x <- 1:10
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
x > 4
```

```
##  [1] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

these are often used as conditions:

```
x[x>4]
```

```
## [1]  5  6  7  8  9 10
```

This, as we will see shortly, is EXTREMELY useful!

1.1.5 Data Frames

data frames are the basic format for data in R. They are essentially vectors put together as columns.

The main thing you need to know about working with data frames are the following commands:

1.1.5.1 Case Study: UPR Admissions

consider the **upr** data set . This is the application data for all the students who applied and were accepted to UPR-Mayaguez between 2003 and 2013.

```
dim(upr)
```

```
## [1] 23666    16
```

tells us that there were 23666 applications and that for each student there are 16 pieces of information.

```
colnames(upr)
```

```
##  [1] "ID.Code"        "Year"           "Gender"         "Program.Code"
##  [5] "Highschool.GPA" "Aptitud.Verbal" "Aptitud.Matem"  "Aprov.Ingles"
##  [9] "Aprov.Matem"    "Aprov.Espanol"  "IGS"            "Freshmen.GPA"
## [13] "Graduated"      "Year.Grad."     "Grad..GPA"      "Class.Facultad"
```

shows us the variables

```
head(upr, 3)
```

```
##      ID.Code Year Gender Program.Code Highschool.GPA Aptitud.Verbal
## 1 00C2B4EF77 2005      M          502           3.97            647
## 2 00D66CF1BF 2003      M          502           3.80            597
## 3 00AB6118EB 2004      M         1203           4.00            567
##   Aptitud.Matem Aprov.Ingles Aprov.Matem Aprov.Espanol IGS Freshmen.GPA
## 1           621          626         672           551 342         3.67
## 2           726          618         718           575 343         2.75
## 3           691          424         616           609 342         3.62
##   Graduated Year.Grad. Grad..GPA Class.Facultad
## 1        Si       2012      3.33           INGE
## 2        No         NA        NA           INGE
## 3        No         NA        NA       CIENCIAS
```

shows us the first three cases.

Let's say we want to find the number of males and females. We can use the table command for that:

```
table(Gender)
```

```
## Error: object 'Gender' not found
```

What happened? Right now R does not know what Gender is because it is "hidden" inside the upr data set. We need to make it visible to R first:

```
attach(upr)
table(Gender)
```

```
## Gender
##     F     M
## 11487 12179
```

there is also a detach command to undo an attach, but this is not usually needed because the attach goes away when you close R.

Note: you need to attach a data frame only once in each session working with R.

Note: Say you are working first with a data set "students 2016" which has a column called Gender, and you attached it. Later (but in the same R session) you start working with a data set "students 2017" which also has a column called Gender, and you are attaching this one as well. If you use Gender now it will be from "students 2017".

1.2 Subsetting of Data Frames

Consider the following data frame (not a real data set):

```
students
```

```
##    Age GPA Gender
## 1   22 3.1   Male
## 2   23 3.2   Male
## 3   20 2.1   Male
## 4   22 2.1   Male
## 5   21 2.3 Female
## 6   21 2.9   Male
## 7   18 2.3 Female
## 8   22 3.9   Male
## 9   21 2.6 Female
## 10  18 3.2 Female
```

Here each single piece of data is identified by its row number and its column number. So for example in row 2, column 2 we have "3.2", in row 6, column 3 we have "Male".

As with the vectors before we can use the [ ] notation to access pieces of a data frame, but now we need to give it both the row and the column number, separated by a ,:

```
students[6, 3]
```

```
## [1] "Male"
```

As before we can pick more than one piece:

```
students[1:5, 3]
```

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

```
students[1:5, 1:2]
```

```
##   Age GPA
## 1  22 3.1
## 2  23 3.2
```

```
## 3  20 2.1
## 4  22 2.1
## 5  21 2.3
```

```
students[-c(1:5), 3]
```

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

```
students[1, ]
```

```
##   Age GPA Gender
## 1  22 3.1   Male
```

```
students[, 2]
```

```
##  [1] 3.1 3.2 2.1 2.1 2.3 2.9 2.3 3.9 2.6 3.2
```

```
students[, -3]
```

```
##    Age GPA
## 1   22 3.1
## 2   23 3.2
## 3   20 2.1
## 4   22 2.1
## 5   21 2.3
## 6   21 2.9
## 7   18 2.3
## 8   22 3.9
## 9   21 2.6
## 10  18 3.2
```

1.3 Vector Arithmetic

R allows us to apply any mathematical functions to a whole vector:

```
x <- 1:10
2*x
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
x^2
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100
```

```
log(x)
```

```
##  [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101
##  [8] 2.0794415 2.1972246 2.3025851
```

```
sum(x)
```

```
## [1] 55
```

```
y <- 21:30
```

```
x+y
```

```
##  [1] 22 24 26 28 30 32 34 36 38 40
```

```
x^2+y^2
```

```
##  [1]  442  488  538  592  650  712  778  848  922 1000
```

```
mean(x+y)
```

```
## [1] 31
```

Let's try something strange:

```
c(1, 2, 3) + c(1, 2, 3, 4)
```

```
## [1] 2 4 6 5
```

so R notices that we are trying to add a vector of length 3 to a vector of length 4. This should not work, but it actually does!

When it runs out of values in the first vector, R simply starts all over again.

In general this is more likely a mistake by you, check that this is what you really wanted to do!

1.4 Subsetting

One of the most common tasks in Statistic is to select a part of a data set for further analysis. There is even a name for this: **data wrangling**.

1.4.1 Case Study: New York Air Quality Measurements

Description: Daily measurements of air quality in New York, May to September 1973.

A data frame with 154 observations on 6 variables.

Ozone: Mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island

Solar.R: Solar radiation in Langleys in the frequency band 4000-7700 Angstroms from 0800 to 1200 hours at Central Park

Wind: Average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport

Temp: Maximum daily temperature in degrees Fahrenheit at La Guardia Airport.

Source: The data were obtained from the New York State Department of Conservation (ozone data) and the National Weather Service (meteorological data).

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

Let's say that instead of looking at the whole data set we want to consider only the months of August and September. Those have Month = 8, 9 and we can select this part of the data set with the [,] notation we discussed earlier:

```
attach(airquality)
airAugSept <- airquality[Month>=8, ]
head(airAugSept)
```

```
##    Ozone Solar.R Wind Temp Month Day
## 93    39      83  6.9   81     8   1
## 94     9      24 13.8   81     8   2
## 95    16      77  7.4   82     8   3
## 96    78      NA  6.9   86     8   4
## 97    35      NA  7.4   85     8   5
## 98    66      NA  4.6   87     8   6
```

This task of data wrangling is so important, there are quite a lot of routines that are helping with it. One of them is **isubset**.

Here is what you do:

```
airAugSept<- isubset(airquality)
```

The app lets you use up to three conditions, we just have one (Month ≥ 8), so we can leave that alone. Now choose the condition and then hit "Click when ready to run"

Here is a screenshot:

now hit Close App and return to R.

In this example we used a very simple condition: Month ≥ 8. These conditions can be much more complicated using & (AND), | (OR) and !(NOT).

Let's say what we want only those days in August and September with a Temperature less than 80:

```
airAugSeptTemp80 <- isubset(airquality)
```

Finally let's say we want only either those days in August and September with a Temperature less than 80, or days with Wind>10:

Let's get back to the days in August and September. What we want to do with those days is to find the mean Ozone level:

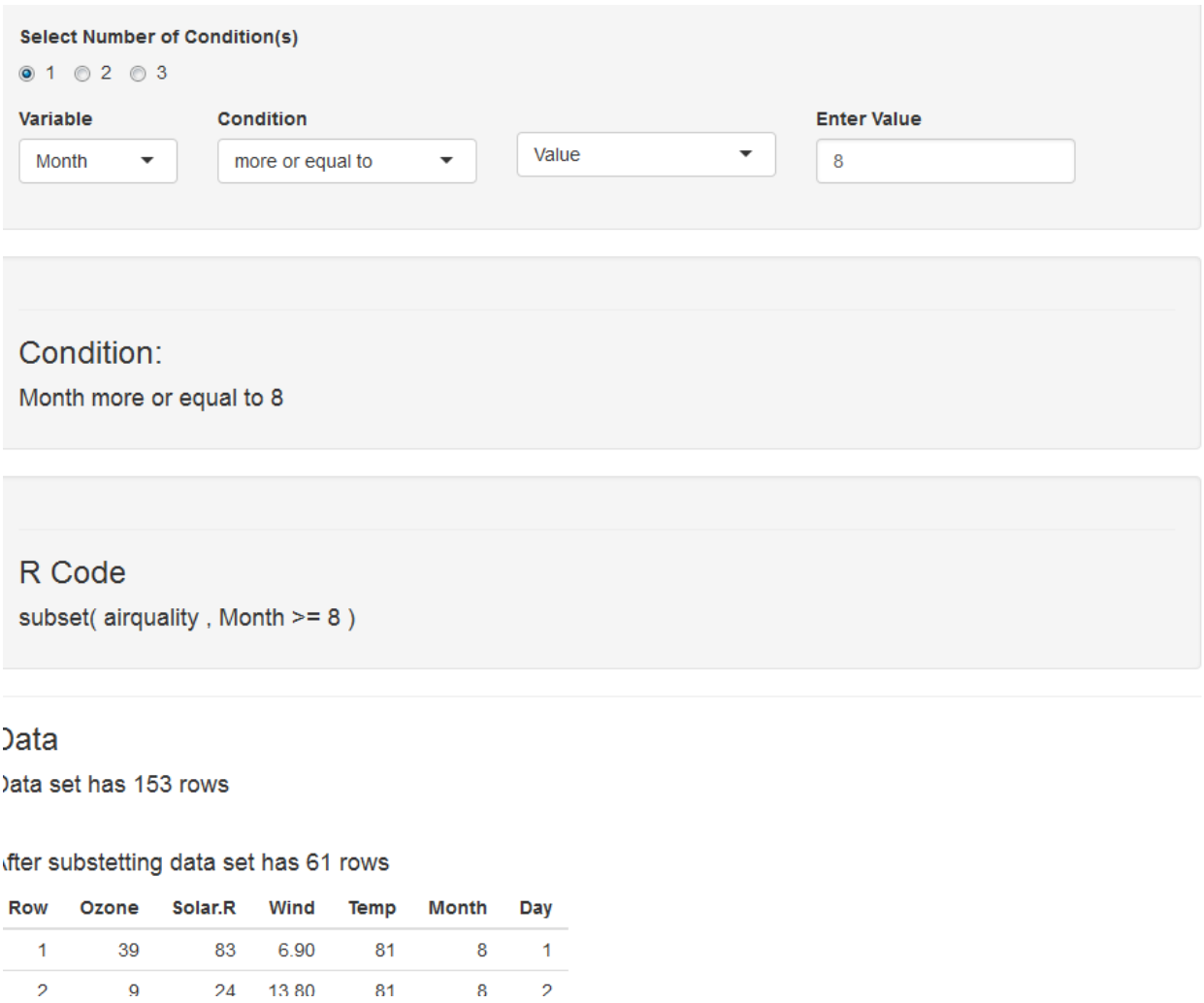

Select Number of Condition(s)
1 2 3
Variable
Month
Condition
more or equal to
Value
Enter Value
8
Condition:
Month more or equal to 8
R Code
subset(airquality , Month >= 8)
Data
Data set has 153 rows
After substetting data set has 61 rows
Row Ozone Solar.R Wind Temp Month Day
1 39 83 6.90 81 8 1
2 9 24 13.80 81 8 2

Figure 1:

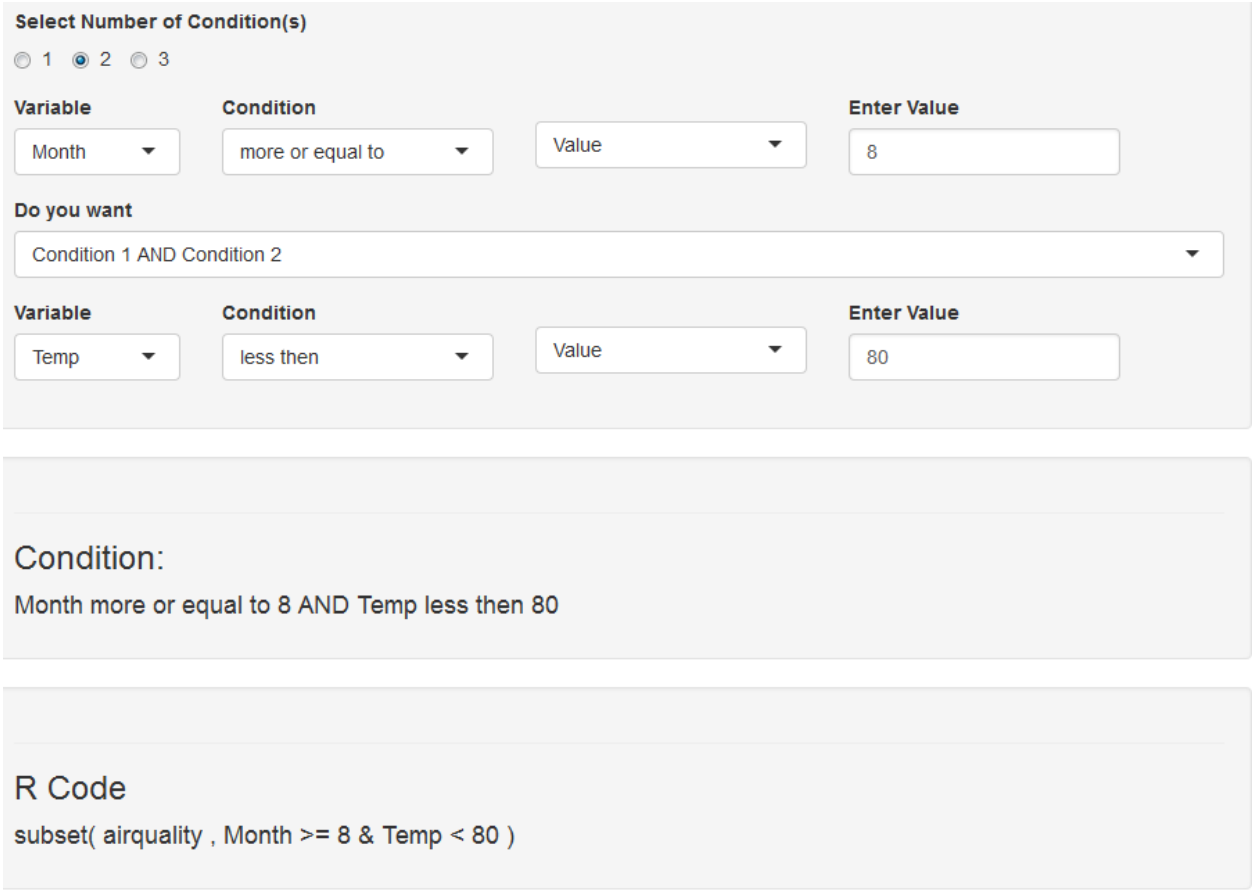

Select Number of Condition(s)
1
2
3
Variable
Condition
Value
Enter Value
Month
more or equal to
Value
8
Do you want
Condition 1 AND Condition 2
Variable
Condition
Value
Enter Value
Temp
less then
Value
80
Condition:
Month more or equal to 8 AND Temp less then 80
R Code
subset(airquality , Month >= 8 & Temp < 80)

Figure 2:

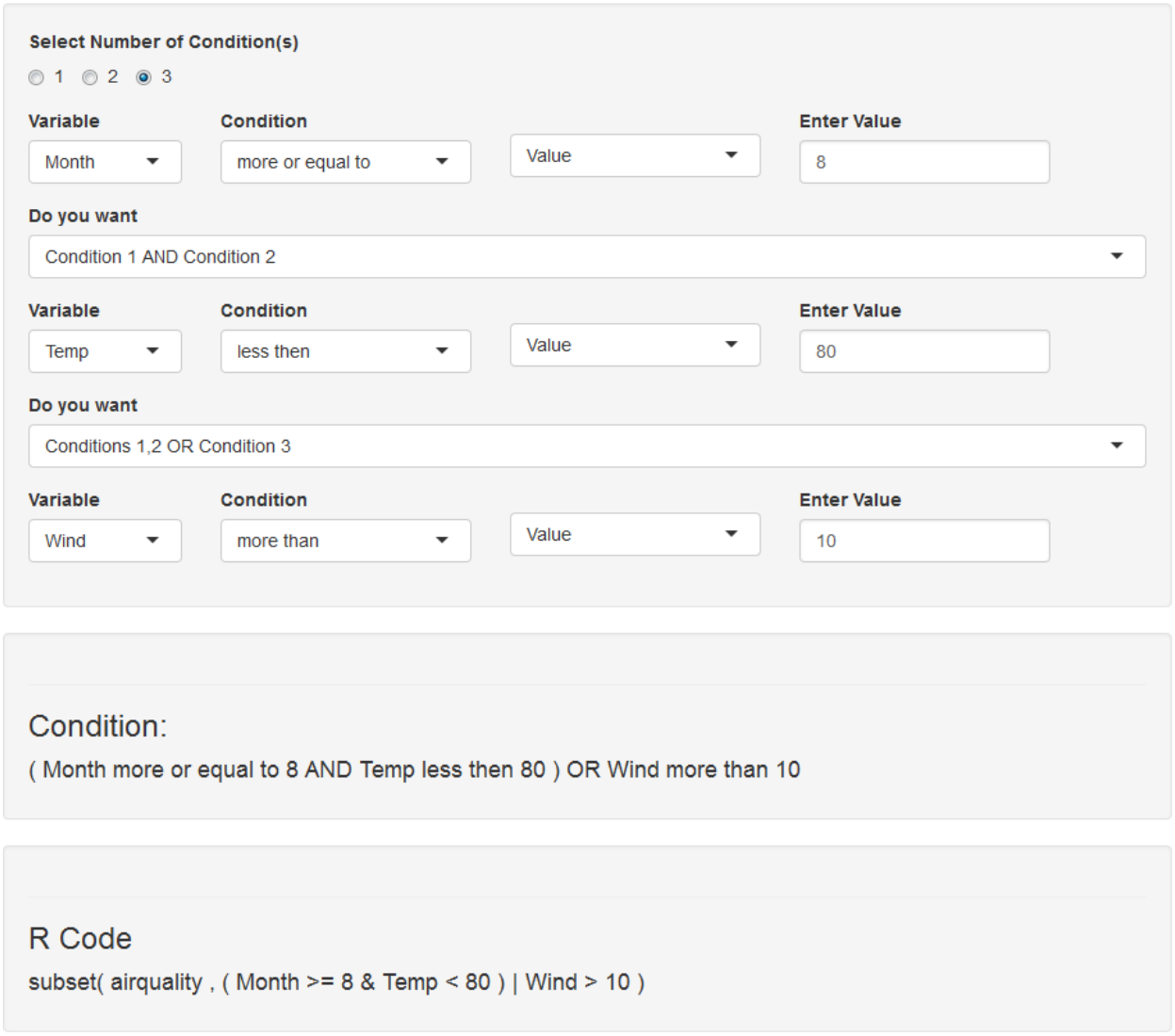

Select Number of Condition(s)
1 2 3
Variable
Condition
Value
Enter Value
Month
more or equal to
Value
8
Do you want
Condition 1 AND Condition 2
Variable
Condition
Value
Enter Value
Temp
less then
Value
80
Do you want
Conditions 1,2 OR Condition 3
Variable
Condition
Value
Enter Value
Wind
more than
Value
10
Condition:
(Month more or equal to 8 AND Temp less then 80) OR Wind more than 10
R Code
subset(airquality , (Month >= 8 & Temp < 80) | Wind > 10)

Figure 3:

```
airAugSept <- isubset(airquality)
mean(Ozone)
```

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

Oh! Something went wrong! The problem is that the column Ozone has *missing values*, which R codes as NA. These are just what it says, for some days the Ozone level was not measured and so is missing. One way to go is to tell R to ignore the missing values:

```
mean(Ozone, na.rm=TRUE)
```

```
## [1] 42.12931
```

or we could use:

```
stat.table(Ozone)
```

```
## Warning:  37  missing values were removed!
```

```
##       Sample Size Mean Standard Deviation
## Ozone         116 42.1                 33
```

OK!

But wait a minute: we are told there are 37 missing values and 116 "good" ones, for a total of 37+116=153. But there are supposed to be only 61 rows (or observations) in airAugSept. Let's check:

```
length(Ozone)
```

```
## [1] 153
```

```
nrow(airAugSept)
```

```
## [1] 61
```

What's wrong?

The problem is that Ozone still comes from the original airquality data set, but our Ozone is still hidden inside airAugSept. One solution would be to

```
attach(airAugSept)
```

but as R is warning us, now there are two Ozones, and it can get quite confusing. To be sure we work with the correct data we can do this:

```
detach(airquality)
stat.table(Ozone)
```

```
## Warning:  6  missing values were removed!
```

```
##       Sample Size Mean Standard Deviation
## Ozone          55 44.9                35.2
```

1.4.2 Case Study: Age and Gender in Puerto Rico in 2000

Breakdown of the population of USA and Puerto Rico by age and gender, according to the 2000 Census

```
head(agesex)
```

```
##           Age  Male Female
## 1 Less than 1 29601  28442
## 2           1 29543  28130
## 3           2 30252  28881
## 4           3 30643  28867
## 5           4 31248  29799
## 6           5 31621  29696
```

```
tail(agesex)
```

```
##           Age Male Female
## 98         97  282    418
## 99         98  189    296
## 100        99  123    196
## 101 100 - 104  258    448
## 102 105 - 109   47     59
## 103  Over 110   17     27
```

shows us that the data set consists of three vectors: the ages, the number of males and the number of females. The first one is a character vector ("less than 1") and the other two are numeric.

Let's answer a few questions about the age and gender in PR in 2000:

- What was the number of men and women in PR in 2000?

```
attach(agesex)
sum(Male)
```

```
## [1] 1833577
```

```
sum(Female)
```

```
## [1] 1975033
```

- How many people where there in PR?

Simple:

```
sum(Male)+sum(Female)
```

```
## [1] 3808610
```

we will need the column with the Male and Female counts a few more times, so maybe we should do it this way:

```
People <- Male + Female
head(People)
```

```
## [1] 58043 57673 59133 59510 61047 61317
```

```
sum(People)
```

```
## [1] 3808610
```

Note

we now have another variable called People among the data sets, as we can see with

```
ls()
```

It will stay there until we close R. If we want to keep it for the next time we use R we need to save everything with File > Save Workspace. If we want to save the workspace but not this variable we first have to

```
rm(People)
```

- How many newborns were there?

```
People[1]
```

```
## [1] 58043
```

- How many teenagers were there?

teenagers (Age from 13 to 19) are in rows 14 - 20, so

```
sum(People[14:20])
```

```
## [1] 433764
```

- What percentage of the population was male, rounded to 1 digit behind the decimal point?

```
sum(Male)/sum(People)*100
```

```
## [1] 48.14294
```

```
round(sum(Male)/sum(People)*100, 1)
```

```
## [1] 48.1
```

- In how many age groups were there more males than females?

Let's start with

```
Male > Female
```

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

```
##  [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [34] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [100] FALSE FALSE FALSE FALSE
```

and now we can find

```
sum(Male > Female)
```

```
## [1] 21
```

- What age group had the largest population?

```
max(People)
```

```
## [1] 64795
```

```
People==max(People)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [34] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [100] FALSE FALSE FALSE FALSE
```

```
Age[People==max(People)]
```

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

Note == is the symbol for "is equal to". The others are

- < "is less than"
- <= "is less or equal to"

- > "is greater than"
- >= "is greater or equal to"

So the age group of 10 year olds is the largest. Why is this answer a bit strange?

Here is another way to do this:

```
order(People, decreasing = TRUE)
```

```
##   [1]  11  21  19  18  20  10   6   8  17   5  22  23  16   7  13  12  15
```

```
##  [18]  14   9   4   3  24   1   2  25  26  30  35  36  29  31  37  28  38
##  [35]  27  41  40  34  39  33  32  43  44  46  42  45  51  53  47  48  54
##  [52]  50  49  52  55  56  57  58  59  61  60  62  63  64  66  65  68  67
##  [69]  69  70  72  71  73  74  75  76  77  78  79  80  81  82  83  84  85
##  [86]  86  87  88  89  90  91  92  93  94  95  96  97 101  98  99 100 102
## [103] 103
```

```
head( agesex[ order(People, decreasing = TRUE), ])
```

```
##    Age  Male Female
## 11  10 33188  31607
## 21  20 32441  32154
## 19  18 32216  31705
## 18  17 32735  31070
## 20  19 32038  31744
## 10   9 31798  30101
```

another useful command is **sort**, which we can use to order one variable, by default from smallest to largest:

```
sort(People)
```

```
##   [1]    44   106   319   485   700   706   847  1122  1332  1728  2285
##  [12]  2694  3640  4466  5261  6278  7279  8414  8726  9132 10436 11659
##  [23] 13449 14211 15293 16657 17514 19403 19673 20588 21421 21865 23123
##  [34] 24982 25596 26222 26929 30387 30552 30690 32035 32737 34118 34715
##  [45] 36268 38544 39146 40807 44265 45004 45280 45875 45926 46155 46311
##  [56] 46579 48142 48987 49262 49499 50003 50009 50828 50951 51259 52213
##  [67] 52395 52553 52795 52807 53293 53573 53709 54352 54815 55124 55313
##  [78] 55754 56337 57673 58043 58725 59133 59510 60020 60112 60216 60221
##  [89] 60456 60695 60707 60748 60786 61047 61221 61231 61317 61899 63782
## [100] 63805 63921 64595 64795
```

- What was the mean age of the population?

Because the data is grouped the mean is found as follows:

$$\frac{(0 \times \text{newborns} + 1 \times \text{one year olds} + 2 \times \text{two year olds} + ...+)}{\text{total population}}$$

Age is a character variable but we need a quantitative one to do arithmetic, so let's make one as close to Age as possible:

```
Ages  <- c(0:99, 102, 107, 112)
Ages
```

```
##   [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  99 102 107
## [103] 112
```

```
round(sum(Ages*People)/sum(People), 1)
```

```
## [1] 34
```

2 Short List of Important R Commands

- **head**: show the first k elements of a dataset

```
head(agesex, 3)
```

```
##           Age  Male Female
## 1 Less than 1 29601  28442
## 2           1 29543  28130
## 3           2 30252  28881
```

- **ls**: list of all elements of the RData file

```
head(ls())
```

```
## [1] "acorn"      "Ages"       "agesex"     "agesexUS"   "aids"
## [6] "airAugSept"
```

- **attach**: make column names of a data frame usable

```
table(Gender)
```

```
## [1] "Error: object Gender not found"
```

```
attach(upr)
head(Gender)
```

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

- **args**: show arguments of a routine

```
args(stat.table)
```

```
## function (y, x, Mean = TRUE, Sort = FALSE, ndigit = 1)
## NULL
```

this does not always work:

```
args(mean)
```

```
## function (x, ...)
## NULL
```

in that case use

- **?**: show details of routine

```
?mean
```

will open a help file in a browser.

- **length**: number of elements of a vector
- **dim**: number of row and columns of a data frame
- **colnames**: names of columns of a data frame
- **rownames**: names of rows of a data frame
- **sum, mean, sd**
- **table**: count the number of occurances

```
table(Gender)
```

```
## Gender
##     F     M
## 11487 12179
```

```
table(Gender, Year)
```

```
##        Year
## Gender 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013
##      F 1102 1040 1162 1137 1208 1219 1180  958  853  769  859
##      M 1151 1118 1138 1098 1256 1219 1237 1073  919  979  991
```

3 Routines in Resma3

3.0.1 Interactive Apps

idataio input and output of data into R.

isummary - graphs and numerical summaries, with or without groups.

ihist - histogram

isplot - scatterplot, with or without groups

isubset - data subsetting

3.0.2 Routines

barchart - Barcharts, one or two Variables

bplot - Boxplot

change.order - Change Ordering of Categorical Variable

chi.gof.test - Chisquare Goodnes-of-fit Test

chi.ind.test - Chisquare Test for Independence

ci.mean.sim - Simulation of Confidence Intervals for one Mean

dlr - Least Squares Regression with one Dummy Variable

dlr.predict Prediction for SLR with Dummy Variable

fivenumber - Five Number Summary

flplot - Fitted Line Graph

get.moodle.data - read data from moodle quizzes

hplot - Histogram

iplot - Interaction Plot

kruskalwallis - Kruskal-Wallis test

mallows - Best Subset Regression

mlr Multiple Regression

mlr.predict - Prediction for Multiple Regression

mplot - Marginal Plot

multiple.graphs - Combine Several Graphs into one

nplot - Normal Probability Plot

one.sample.t - Infrerence for one Mean

one.sample.prop - Inference for one Proportion

one.sample.wilcoxon - Wilcoxon Rank Sum Test, non parametric alternative to one.sample.t

oneway - One-way ANOVA

pearson.cor - Test and Interval for Correlation

prop.ps - Power and Sample Size for one Proportion

slr - Regression for One Predictor

slr.predict Prediction for Regression with one Predictor

splot Scatterplot, also with groups

stat.table - Summary Statistics

t.ps - Power and Sample Size for one Mean

test.mean.sim - Simulation of Hypothesis esting for one Mean

tukey - Tukey Multiple Comparison, one or two Factors

twoway - Two-way ANOVA

3.1 Interactive Apps

These are apps that open a new window and then allow the user to do all the work using (mostly) point and click.

Most of these apps are called with data sets as arguments. They will accept any number of arguments, which can be either vectors, matrices or data frames. If any of the the later arguments do not match the first one in length they are ignored. Some apps also return a data set.

Most of the apps also show the commands that could be used in R directly to produce the same results, either with the Resma3 commands or without them.

3.1.1 idataio

Routine to read data into R and export data to a file. It allows for

- data entered from the keyboard into a spreadsheet
- data read from a file
- data downloaded from the internet
- data copied from another program such as a browser or an Excel spreadsheet

Almost all standard file formats are supported, such as csv, excel, html, etc. For a complete list see

Examples:

```
dta <- idataio()
```

3.1.2 isummary

graphical and numerical summaries of one numerical vector, optionally rouped by a categorical variable

Examples

```
attach(mtcars)
isummary(mtcars)
isummary(mpg)
isummary(mpg, gears)
```

3.1.3 ihist

draws histograms

Examples

```
ihist(mtcars)
```

3.1.4 isplot

scatterplots

Examples

```
isplot(mtcars)
isplot(mpg, disp, gear, cyl)
```

3.1.5 isubset

subsetting a data frame or vector

Examples:

```
new.mtcars <- isubset(mtcars)
```

4 General Comments on Resma3 Routines

The routines I wrote for this course all use the following standard (where it makes sense)

first argument y is a numeric vector ("Response")

second argument x is either a numeric or categorical vector or matrix ("Predictor" or "Factor")

Sometimes there is a third argument z, always a categorical vector ("Group")

Obvious exceptions: routines for categorical data analysis (barchart, chi.ind.test, chi.gof.test)

Many of the routines have the following arguments:

return.result=FALSE (Optional): if TRUE returns results as vector for further use. This allows storing the results, for example to do simulation.

You can get all the routines and data sets by downloading and opening Resma3.RData

---

sometimes you might make a mistake entering the data, or you want to change a few values. In that case use

```
students <- edit(students)
```

This brings up the spreadsheet and you can do the changes there!

4.1 Standard R Routines

4.1.1 attach

Arguments
x: a data frame
makes column names "visible" to R

Examples:

```
attach(mothers)
mean(Length)
```

Note: you need to do this only once in any R session, it will stay until you close R.

4.1.2 mean, median, sd, IQR, quantile, cor

Summary statistics for quantitative data

Arguments
x: a numeric vector
na.rm = FALSE

Examples:

```
mean(Length)
```

```
median(Length)
```

```
sd(Length)
```

```
 IQR(Length)
```

```
quantile(Length, c(0.25,0.75))
```

Note: all these routines have an argument na.rm = FALSE, so if the data set has missing values (NA) the result is NA. Simply use na.rm = TRUE

4.1.3 table

Tables and cross-tabulation for categorical data

Arguments:
x: either a categorical vector or a data frame with two categorical columns
y: a second categorical vector (if x is a vector as well)

Examples:

```
head(rogaine,3)
```

```
table(rogaine)
```

4.1.4 cor

Pearson's correlation coefficient **Arguments:**
x: either a numeric vector or a data frame with two or more numeric columns
y: a second numeric vector (if x is a vector as well)
use = "everything", set to use="complete.obs" if NA's in the data

Examples:

```
x <- rnorm(50)
y <- rnorm(50)
cor(x, y)
```

```
cor(cbind(x,y))
```

4.1.5 subset

find a subset of a data set based on some condition(s)

Arguments:

x: a data frame

cond: some logical condition

select (Optional): which columns should be returned, default is all of them

drop=FALSE, if just one column is selected as output use drop=TRUE

Examples:

```
head(subset(wrinccensus, Satisfaction>=4, select=Income),3)
head(subset(wrinccensus, Satisfaction>=4 & Gender=="Male"),3)
head(subset(wrinccensus, Satisfaction>=4 & Gender=="Male", select=c(Income,Job.Level)),3
head(subset(wrinccensus, Satisfaction>=4 & Gender=="Male", select=Income),3)
```

Note that the last one results in a data frame with one column. You might want it as a numeric vector:

```
head(subset(wrinccensus, Satisfaction>=4 & Gender=="Male", select=Income, drop=TRUE),3)
```

NOTE: see also interactive app **isubset**

5 Resma3 routines

5.0.1 get.moodle.data

read data from moodle quizzes

highlight the data, use mouse to copy, switch to R and run

```
get.moodle.data()
```

5.1 Routines for Summary Statistics

5.1.1 stat.table

tables of summary statistics, with or without groups **Arguments** y: numeric vector (Required) x: categorical variable (Optional) Mean=TRUE: if set to FALSE table finds medians and IQRs **Examples:**

```
stat.table(Length)
stat.table(Length,Status)
stat.table(Length,Status,Mean=FALSE)
```

5.2 Routines for One Variable

5.2.1 fivenumber

five number summary and IQR, with or without groups

Arguments:

y: quantitative vector x: (optional) categorical vector

Example:

```
fivenumber(Length)
```

5.2.2 one.sample.t

Confidence interval or hypothesis test for one mean

Arguments:

y: either a vector with numbers or the sample mean of the data shat, n: standard deviation and sample size (only needed if y is sample mean)

mu.null: mean in null hypothesis (if missing confidence interval is found)

alternative = "equal": alternative hypothesis

conf.level = 95

ndigit = 1 (number of digits for rounding)

Examples:

```
one.sample.t(Length, conf.level=90)
one.sample.t(49.55, 3.38, 94, conf.level=90, ndigit=2)
one.sample.t(Length, mu.null=50, alternative="less")
```

5.2.3 t.ps

power and sample size calculations for one mean

Arguments:

n: sample size diff: difference in means
sigma: standard deviation
power: power of test
E (optional): error of confidence interval (for sample size calculation only)
conf.level=90: confidence level of confidence interval (for sample size calculation only)
alpha = 0.05: type I error probability
alternative = "equal": alternative hypothesis
routine finds whatever argument is left out (n, diff or power)

Examples:

```
t.ps(n=100, diff=1.23, sigma=5, alpha=0.1, alternative="greater")
t.ps(power=90, d=1, sigma=13, alpha=0.1, alternative="greater")
t.ps(sigma= 0.5, E=0.125, conf.level=99)
```

5.2.4 wilcoxon

Wilcoxon rank sum test for one quantitative variable - non parametric alternative to one.sample.t

Arguments:

y: quantitative vector
mu.null: mean in null hypothesis (if missing confidence interval is found)
alternative = "equal": alternative hypothesis
conf.level = 95

Examples:

```
wilcoxon(Length, conf.level=90)
wilcoxon(Length, mu.null=50, alternative="greater")
```

5.2.5 one.sample.prop

Confidence interval or hypothesis test for one proportion (percentage, probability)

Arguments:

x: number of successes
n: number of trials
pi.null: proportion in null hypothesis (if missing confidence interval is found)
alternative = "equal": alternative hypothesis
conf.level = 95

Examples:

```
one.sample.prop(40, 100, conf.level=90)
one.sample.prop(40, 100, pi.null=0.5, alternative=less)
```

5.2.6 prop.ps

Power and sample size calculations for one proportion

Arguments:

n: sample size phat: alternative proportion
pi.null: proportion under null hypothesis
power: power of test
E (optional): error of confidence interval (for sample size calculation only)
conf.level=90: confidence level of confidence interval (for sample size calculation only)
alpha = 0.05: type I error probability
alternative = "two.sided": alternative hypothesis
routine finds whatever argument is left out (n, phat or power)

Examples:

```
prop.ps(n=100, phat=0.65, pi.null=0.5)
prop.ps(power=90, phat=0.65, pi.null=0.5)
```

5.2.7 chi.gof.test

Chisquare test for multinomial proportions

Arguments:

x: observed counts p: hypothesized proportions

Example

```
chi.gof.test(c(12, 17, 20, 15, 10, 26), rep(1,6)/6)
```

5.3 Routines for Two Variables

5.3.1 pearson.cor

Confidence interval and hypothesis test for Pearson's correlation coefficient

Arguments:

y: quantitative vector
x: quantitative vector
rho.null (if missing confidence interval is found, only rho.null = 0 accepted)

conf.level = 95 confidence level of interval

Note: when the routine is run R sometimes gives a

Warning message:
Continuous x aesthetic – did you forget aes(group=...)?

just ignore this

Example:

```
pearson.cor(Draft.Number, Day.of.Year, rho.null = 0)
```

5.4 Routines for Simulations

5.4.1 ci.mean.sim

does a simulation for coverage of the t test confidence intervals

Arguments:

n : sample size mu: mean sigma: standard deviationconf.level: nominal coverage

Example:

```
ci.mean.sim(n=500,mu=75,sigma=30,conf.level=99)
```

5.4.2 test.mean.sim

does a simulation of the p value of the t test. If mu.null=mu it finds the true type I error α, otherwise the power of the test. In either case it draws the histogram of p values.

Arguments:

n : sample size
mu: mean
mu.null=mu: value of mean under null hypothesis
sigma: standard deviation
alpha: nominal alpha

Examples:

```
test.mean.sim(n=20, mu=5, sigma=1, alpha=0.1)
test.mean.sim(n=20, mu=5, mu.null=5.5, sigma=1, alpha=0.1)
```

5.5 Routines for Graphs

5.5.1 barchart

bar charts

Arguments:

y: a table (often from a call to the table routine)

Percent: if missing graph uses counts. Other values are “Grand”, “Row” or “Column” for respective percentages

new.order: for changing the order of the bars

Polygon = FALSE if TRUE adds polygon

Examples:

```
attach(rogaine)
barchart(table(Growth))
barchart(table(Growth), Percent="Grand")
barchart(table(Growth), Percent="Grand",Polygon=TRUE)
barchart(table(rogaine))
barchart(table(rogaine),Percent="Row")
```

5.5.2 hplot

Histogram, if desired with fitted density

Arguments:

x: numerical data
f: name of distribution (Optional)
par: parameters of distribution(Optional)
n: number of bins (Optional) label_x, main_title: x axis label and graph title (Optional)

Examples:

```
hplot(Length)
hplot(Length, label_x = "Length of Babies (cm)", main_title = "Mothers, Babies and Cocai
hplot(Length, f = "norm", par = c(mean(Length), sd(Length)))
```

5.5.3 bplot

Boxplot / do.violinplot

Arguments:

y: numeric vector or matrix or data frame
x: catagorical vector (Optional)
do.violin = FALSE: if TRUE does violin plot
orientation=“vertical”, if orientation=“horizontal” boxplot is drawn horizontally
new_order: change the order of the boxes. Either a vector of position numbers or “Sort”, then sorted from smallest mean to largest.
label_x, label_y, main_title: axes labels and graph title (Optional)
Examples:

```
bplot(Length)
bplot(Length, Status)
bplot(Length, Status, label_y  = "Length of Babies (cm)",
```

```
      label_x = "Drug Status",
      main_title = "Mothers, Babies and Cocain Use")
```

5.5.4 splot

Scatterplot, possibly with groups and fits

Arguments:

y: numeric vector , y axis
x: numeric vector, x axis
z: catagorical variable (Optional)
w: second catagorical variable (Optional)
plot.points=TRUE: if FALSE dots are not plotted add.line = 0: adds lines, if add.line=1 least squares regression line, if add.line=2 LOESS, if add.line=3 it does the line graph

jitter = FALSE: if true jitters dots
use.facets = FALSE: if TRUE usess facets instead of colors for z
errorbars = FALSE: if TRUE adds error band to fit
label_x, label_y, label_z, main_title: axes labels and graph title (Optional)
add.text, add.text_x, add.text_y: add text to graph (Optional)
plotting.size = 1: size of plotting symbols
plotting.symbols: change plotting symbols. can use either symbols added on keyboard or numbers corresponding to R symbols key(Optional)

plotting.colors: change colors, can use either numbers corresponding to R color key or explicit text : pcolor="red" (Optional)

ref_x, ref_y: add reference lines (Optional)

log_x = FALSE, log_y = FALSE: change to log scale

no.legends = FALSE: rmove all alegends

Examples:

```
attach(salaries)
splot(Salary,Years)
splot(Salary,Years, add.line=1)
splot(Salary,Years, Level, add.line=1)
splot(Salary,Years, add.line=3)
```

```
attach(upr)
splot(y = Freshmen.GPA, x = IGS, z = Gender, use.facets = TRUE, add.line = 1, label_y =
```

NOTE: see also ineractive app **isplot**

5.5.5 mplot

Marginal plot with scatterplot and boxplots

Arguments:

y: numeric vector , y axis

x: numeric vector, x axis

z: catagorical variable (Optional)

add.line = 0: adds lines, if add.line=1 least squares regression line, if add.line=2 LOESS, if add.line=3 it does the line graph

Examples:

```
mplot(Salary, Years)
```

Note: when the routine is run R sometimes gives a Warning message: Continuous x aesthetic – did you forget aes(group=...)? Just ignore that

5.5.6 flplot

Fitted line plot, allows for log transforms or polynomial fitting

Arguments:

y: numeric vector , y axis

x: numeric vector, x axis

z: catagorical variable (Optional)

additive = FALSE: if true fits parallel lines

logx = FALSE, logy = FALSE: if true applies log transforms

polydeg = 1: degree of polynomial to be fit

jitter = FALSE: if true jitters dots

Examples:

```
attach(longjump)
flplot(LongJump, Year)
flplot(LongJump, Year, polydeg=2)
attach(elusage)
flplot(elusage[,3], elusage[,4], logx=TRUE, logy=TRUE)
```

5.5.7 nplot

Normal probability plot

Arguments:

y: numerical vector

x: categorical vector (Optional)

Examples:

```
nplot(euros[,1])
```

5.5.8 iplot

Interaction plot

Arguments:

y: numerical vector

x and z: categorical vectors

Examples:

```
attach(fermentation)
iplot(Ethanol, Sugar, Oxygen)
```

5.5.9 multiple.graphs

combine (up to four graphs) in one

5.5.9.1 Arguments:

ggplt objects, likely generated using other graph functions with the argument returnGraph=TRUE

titles (Optional) titles for each graph

Examples:

```
attach(gasoline)
plt1 <- bplot(MPG, Gasoline, returnGraph=TRUE)
plt2 <- bplot(MPG, Automobile, returnGraph=TRUE)
multiple.graphs(plt1,plt2)
```

```
x<-rnorm(1000)
multiple.graphs(
  hplot(x, n=10, returnGraph=TRUE),
  hplot(x, n=25, returnGraph=TRUE),
  hplot(x, n=50, returnGraph=TRUE),
  hplot(x, n=100, returnGraph=TRUE),
  titles =  paste(c(10, 25, 50, 100), "bins")
  )
```

5.6 Routines for Testing with two or more Variables

5.6.1 chi.ind.test

Chisquare test of independence

Arguments:

x: a table of counts

Examples:

```
chi.ind.test(table(rogaine))
```

5.6.2 oneway

ANOVA with one factor

Arguments:

y: numeric vector

x: categorical vector

ndigit = 1: rounding answer to 1 digit

var.equal = TRUE: assume equal variance

conf.level = 95: in the case of a categorical variable with 2 levels finds a 95% confidence interval for the difference in means

Examples:

```
oneway(Length, Status)
```

5.6.3 kruskalwallis

Non-parametric ANOVA

Arguments:

y: numeric vector

x: categorical vector

Examples:

```
kruskalwallis(Length, Status)
```

5.6.4 twoway

ANOVA with two factors

Arguments:

y: numeric vector

x, z: categorical vectors

with.interaction = TRUE: assume interaction is present (defaults to FALSE if there are no repeated measurements)

Examples:

```
attach(gasoline)
twoway(MPG, Gasoline, Automobile)
twoway(MPG, Gasoline, Automobile, with.interaction="FALSE")
```

5.6.5 tukey

Tukey's Multiple Comparison in ANOVA

Arguments:

y: numeric vector

x : categorical vector

z : second categorical vector (Optional)

with.interaction = TRUE: assume interaction is present (defaults to FALSE if there are no repeated measurements)

which="first": do comparison for first categorical variable (x), or change to which="second" or which="interaction"

Examples:

```
tukey(mothers[,2], mothers[,1])
tukey(MPG, Gasoline, Automobile, which="first")
tukey(MPG, Gasoline, Automobile, which="interaction")
```

5.6.6 slr

Linear Regression with one predictor, including polynomial regression

Arguments:

y, x: numerical vectors

no.intercept = FALSE: fit intercept?

polydeg = 1: fit polynomial of higher degree?

show.tests=FALSE: if TRUE t tests for coefficients are shown

Examples:

```
slr(wine[,3],wine[,2])
slr(wine[,3],wine[,2],polydeg=2)
slr(log(wine[,3]),wine[,2],polydeg=2)
```

5.6.7 slr.predict

Prediction for simple linear regression

Arguments:

same as slr. In addition:

newx = x: predict for values for x (can be vector). If missing predict for values in data set.

interval: either "PI" for prediction intervals or "CI" for confidence intervals

conf.level = 95

Examples:

```
slr.predict(wine[,3], wine[,2],newx=c(2,2.5,3), interval="PI", conf.level=90)
```

5.6.8 mlr

Linear Regression with more than one predictor

Arguments:

y: numerical vector

x: numeric matrix with predictors in columns

show.tests=FALSE: if TRUE t tests for coefficients are shown

returnModel=FALSE, if TRUE fit object is returned (and can be used in other routines)

Examples:

```
mlr(houseprice[,1], houseprice[, -1])
```

5.6.9 mlr.predict

Prediction for regression with more than one predictor

Arguments:

same as slr.predict but here x and newx are matrices

Examples:

```
newx <- cbind(c(2000, 2100, 2200), rep(1, 3), rep(2, 3), rep(2, 3))
mlr.predict(houseprice[,1], houseprice[, -1], newx=newx, interval="PI", conf.level = 99)
```

5.6.10 mallows

Best subset regression with Mallow's Cp

Arguments:

same as mlr

Examples:

```
mallows(houseprice[,1], houseprice[, -1] )
```

5.6.11 dlr

Linear regression with one dummy variable

Arguments:

y: numerical vector

x: numeric vectorz: categorical vector

additive = FALSE: if parallel lines set to TRUE

show.tests=FALSE: if TRUE t tests for coefficients are shown

Examples:

```
dlr(salaries[,1], salaries[,2], salaries[,3])
dlr(salaries[,1], salaries[,2], salaries[,3], additive=T)
```

5.6.12 dlr.predict

Prediction for regression with a dummy variable

Arguments:

same as slr.predict but also needs newz: values of categorical variable for prediction

Examples:

```
dlr.predict(salaries[, 1], salaries[, 2], salaries[, 3],
  newx=5, newz="Low", interval="PI")
```

5.7 Miscellaneous Routines

5.7.1 change.order

Change the order of a categorical variable

Arguments:

z: categorical variable

NewOrder: can be a numeric vector specifying a certain order or a categorical vector with ordered values of z

Examples:

```
bplot(Length, Status)
bplot(Length, change.order(Status,c(2,1,3)))
bplot(Length, change.order(Status,c("Throughout","First Trimester","Drug Free")))
```

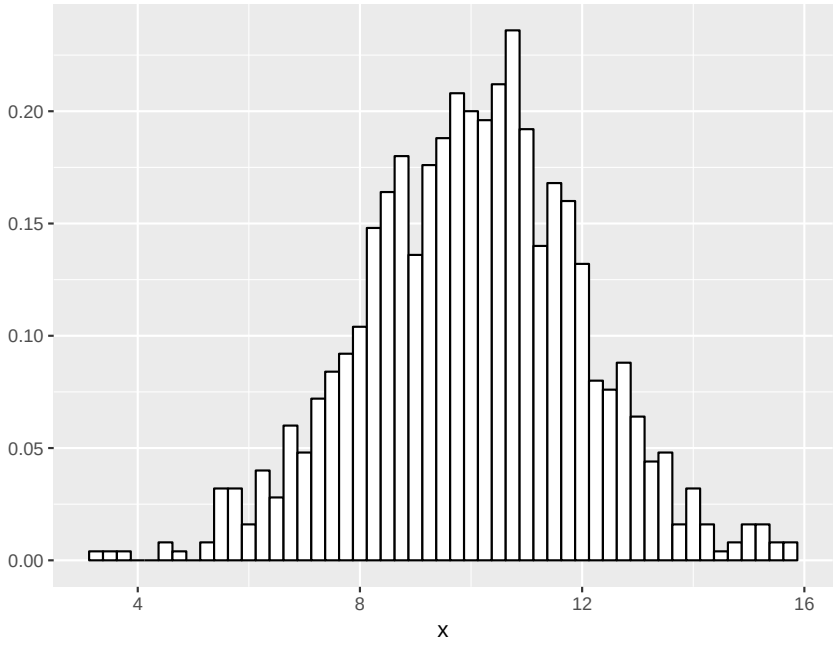
6 Resma3 vs Basic R

In this section we will see how some of our problems could be done with base R.

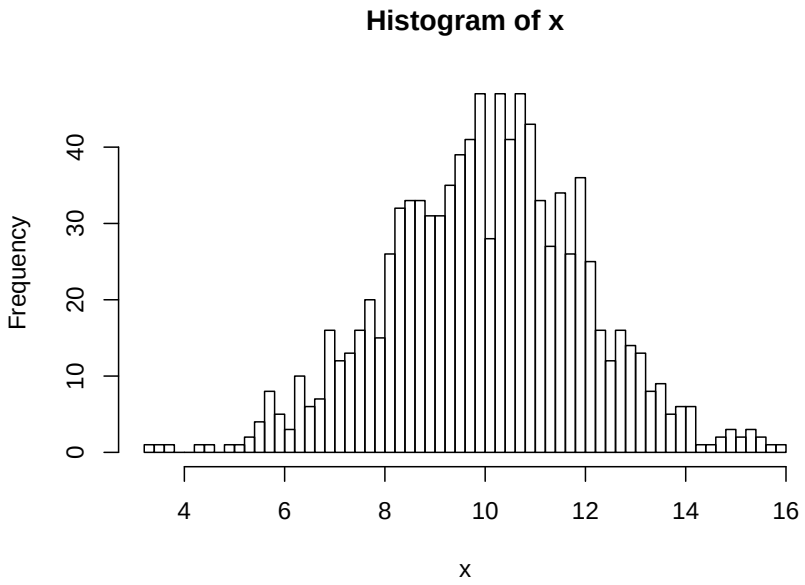
6.1 Graphs

6.1.1 Histogram

```
x <- rnorm(1000, 10, 2)
hplot(x, n=50)
```

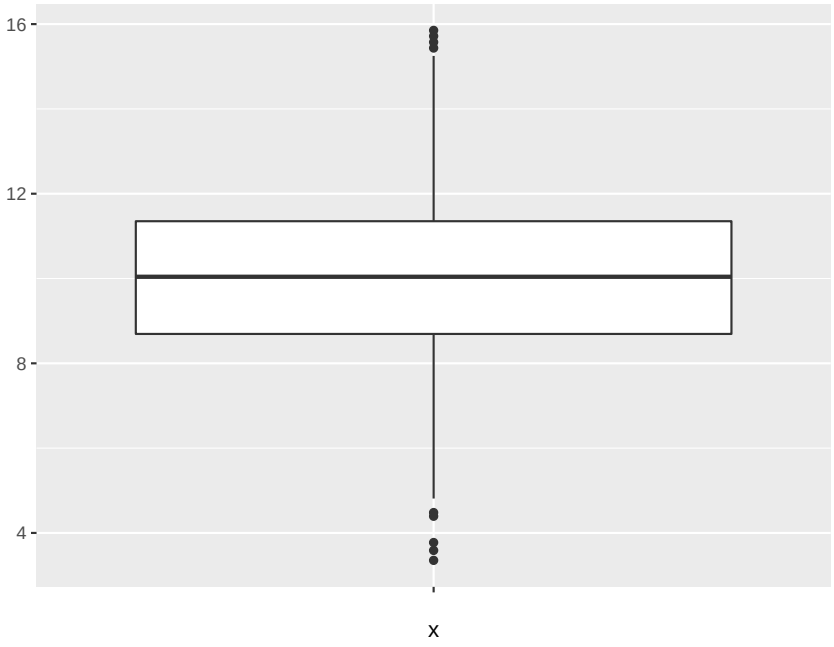



```
hist(x, 50)
```

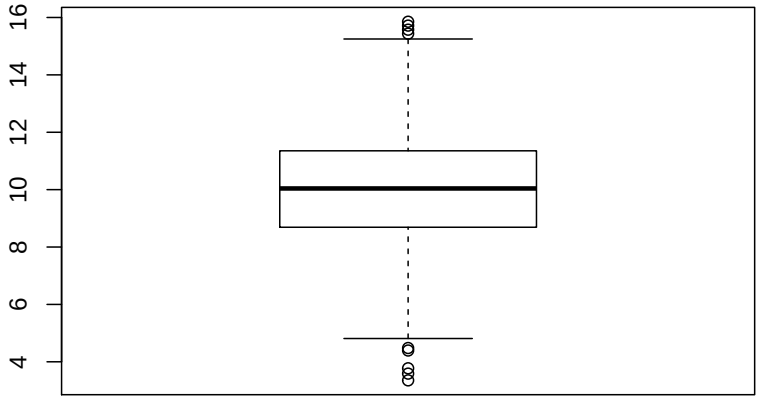


6.1.2 Boxplot

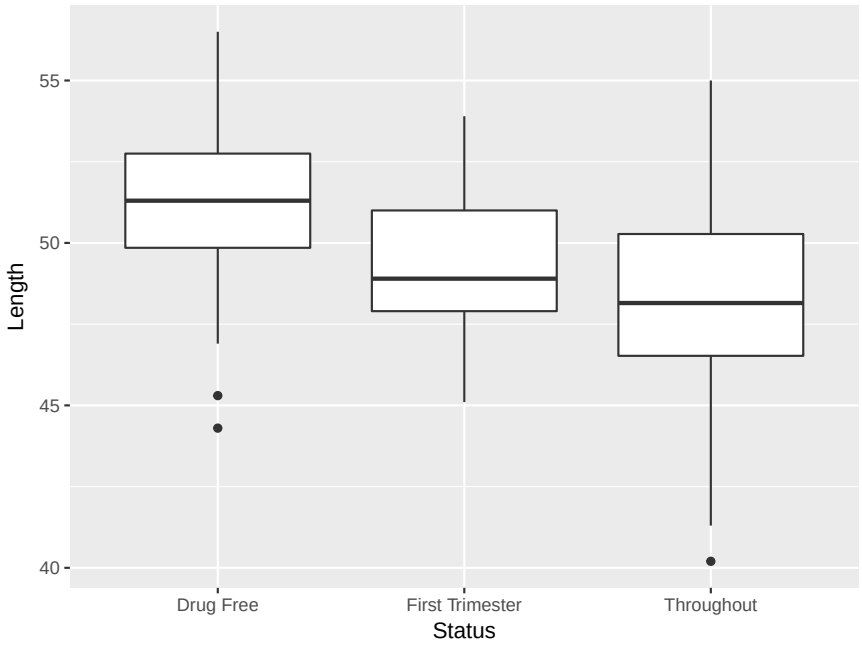
```
bplot(x)
```



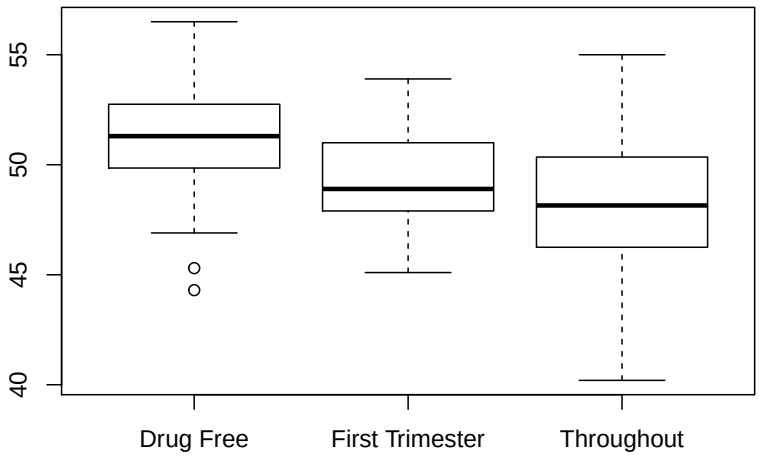
```
boxplot(x)
```




```
attach(mothers)
bplot(Length, Status)
```

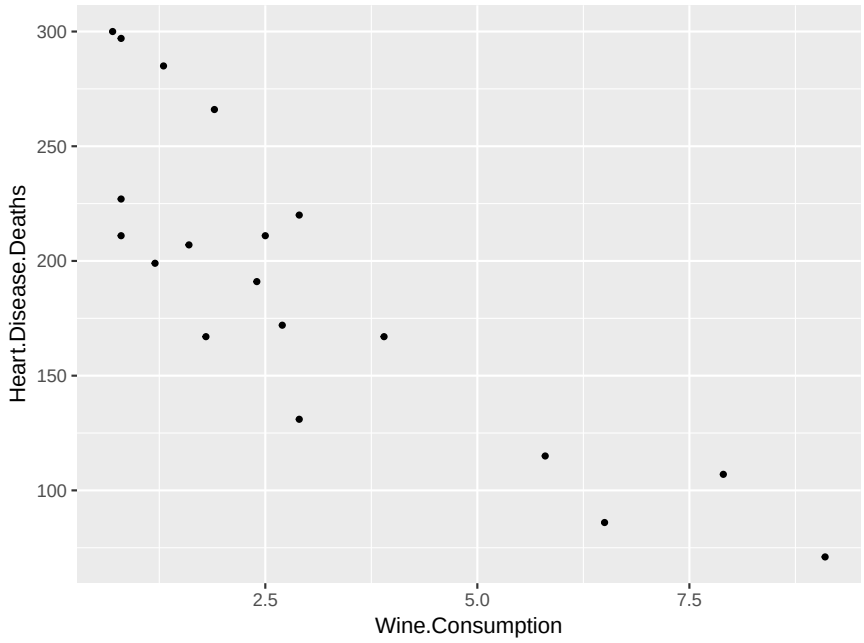



```
boxplot(Length~Status)
```

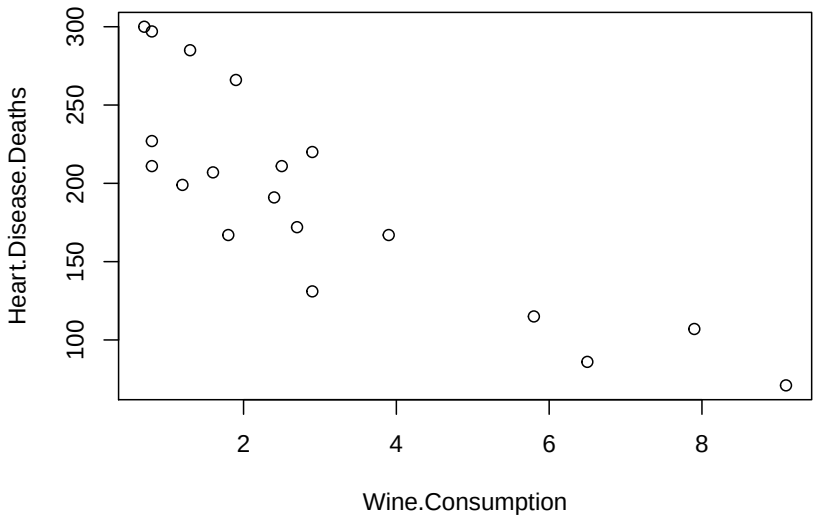


6.1.3 Scatterplot

```
attach(wine)
splot(Heart.Disease.Deaths, Wine.Consumption)
```




```
plot(Wine.Consumption, Heart.Disease.Deaths)
```



6.2 Summary Statistics

```
fivenumber(x, ndigit = 2)
```

```
##  Minimum   Q1 Median    Q3 Maximum
##     3.35 8.69  10.04 11.35   15.85
## IQR =  2.66
```

```
round(c(min(x), quantile(x, 0.25), median(x), quantile(x, 0.75), max(x)), 2)
```

```
##         25%         75%
##  3.35  8.69 10.04 11.35 15.85
```

```
stat.table(x, ndigit = 2)
```

```
##   Sample Size  Mean Standard Deviation
## x        1000 10.02                1.98
```

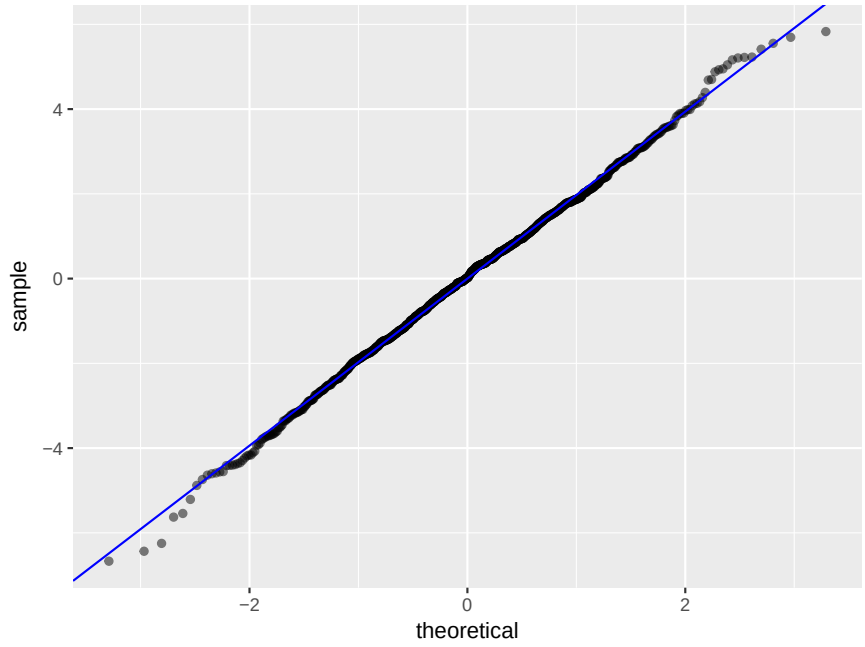
```
round(c(length(x), mean(x), sd(x)), 2)
```

```
## [1] 1000.00   10.02    1.98
```

6.3 Confidence Intervals/Hypothesis Tests

6.3.1 Mean

```
one.sample.t(x, conf.level = 90,  ndigit = 3)
```

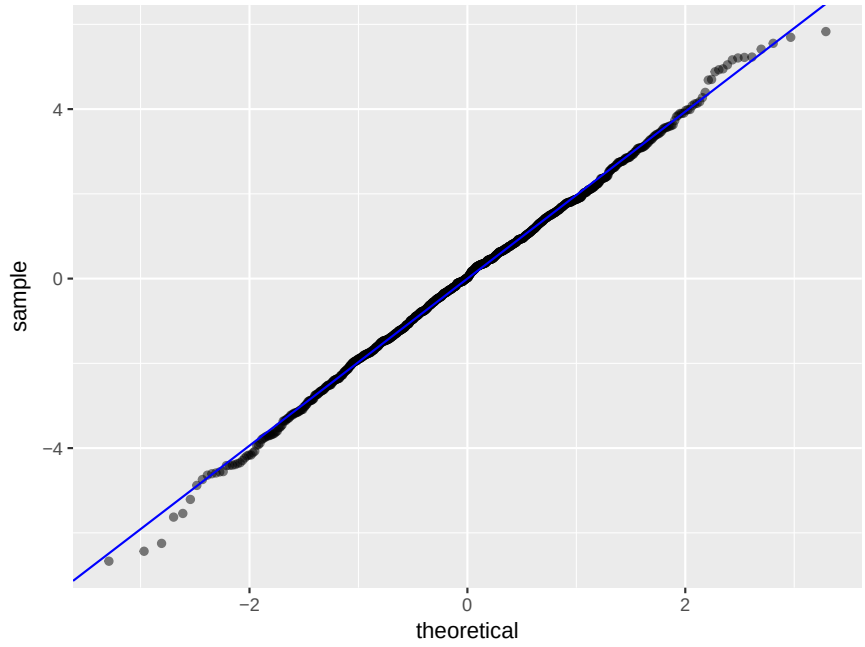



```
## A 90% confidence interval for the population mean is (9.919, 10.125)
```

```
t.test(x, conf.level = 0.9)
```

```
##
##  One Sample t-test
##
## data:  x
## t = 160.32, df = 999, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##   9.919363 10.125213
## sample estimates:
## mean of x
##  10.02229
```

```
one.sample.t(x, mu.null = 10,
             alternative = "greater", ndigit = 3)
```




```
## p value of test H0: mu=10 vs. Ha: mu > 10:  0.3608
```

```
t.test(x, mu=10, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  x
## t = 0.35652, df = 999, p-value = 0.3608
## alternative hypothesis: true mean is greater than 10
## 95 percent confidence interval:
##  9.919363      Inf
## sample estimates:
## mean of x
##  10.02229
```

the *t.ps* command does not exist in base R.

6.3.2 Proportion

```
one.sample.prop(60, 100, conf.level = 90,  ndigit = 3)
```

```
## A 90% confidence interval for the population proportion is (0.513, 0.682)
```

```
prop.test(60, 100, conf.level = 0.9)
```

```
##
##  1-sample proportions test with continuity correction
##
```

```
## data:  60 out of 100, null probability 0.5
## X-squared = 3.61, df = 1, p-value = 0.05743
## alternative hypothesis: true p is not equal to 0.5
## 90 percent confidence interval:
##  0.5127842 0.6816248
## sample estimates:
##   p
## 0.6
```

```
one.sample.prop(60, 100, pi.null = 0.5,
             alternative = "greater", ndigit = 3)
```

```
## p value of test H0: pi=0.5 vs. Ha: pi > 0.5:  0.0287
```

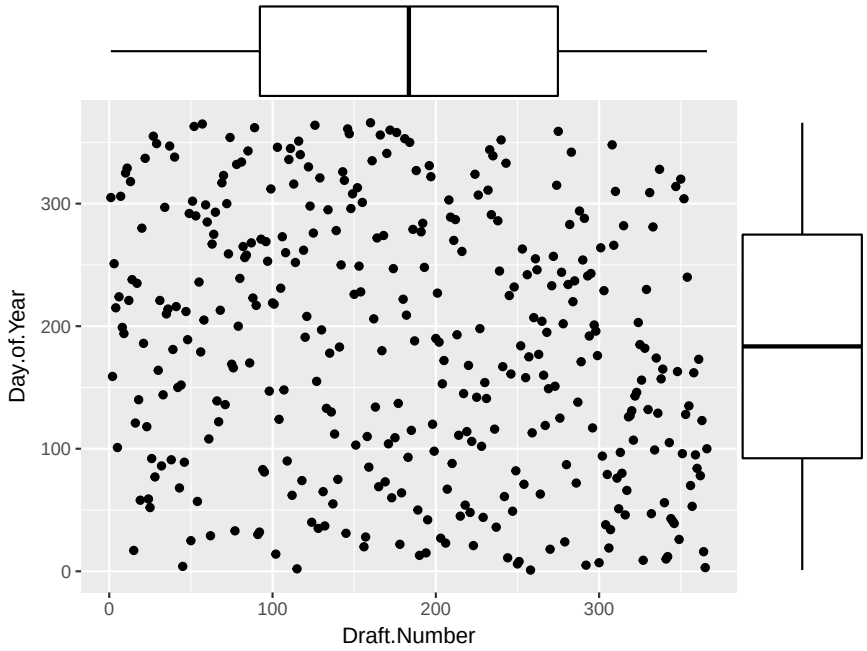
```
prop.test(60, 100, p=0.5, alternative = "greater")
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  60 out of 100, null probability 0.5
## X-squared = 3.61, df = 1, p-value = 0.02872
## alternative hypothesis: true p is greater than 0.5
## 95 percent confidence interval:
##  0.5127842 1.0000000
## sample estimates:
##   p
## 0.6
```

the *prop.ps* command does not exist in base R.

6.3.3 Correlation

```
attach(draft)
pearson.cor(Draft.Number, Day.of.Year, conf.level = 90)
```




```
## A 90% confidence interval for the
## population correlation coefficient is ( -0.306, -0.143 )
```

```
cor.test(Draft.Number, Day.of.Year, conf.level = 0.9)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Draft.Number and Day.of.Year
## t = -4.4272, df = 364, p-value = 1.264e-05
## alternative hypothesis: true correlation is not equal to 0
## 90 percent confidence interval:
##  -0.3061994 -0.1427007
## sample estimates:
##        cor
## -0.2260414
```

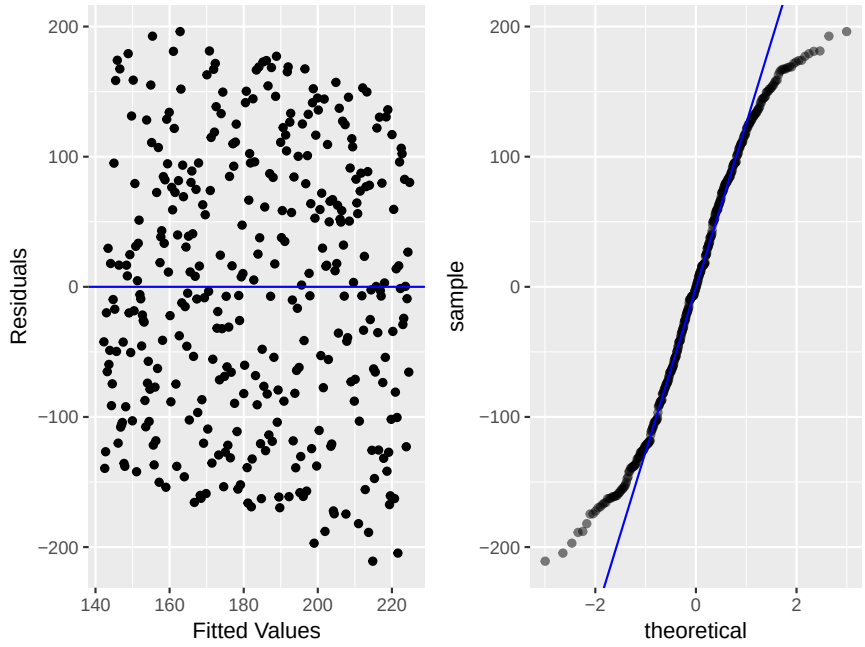
```
pearson.cor(Draft.Number, Day.of.Year, rho.null = 0)
```

```
## p value of test H0: rho=0 vs. Ha: rho <> 0:  0.000
```

6.3.4 Regression

- Simple Regression

```
slr(Draft.Number, Day.of.Year)
```



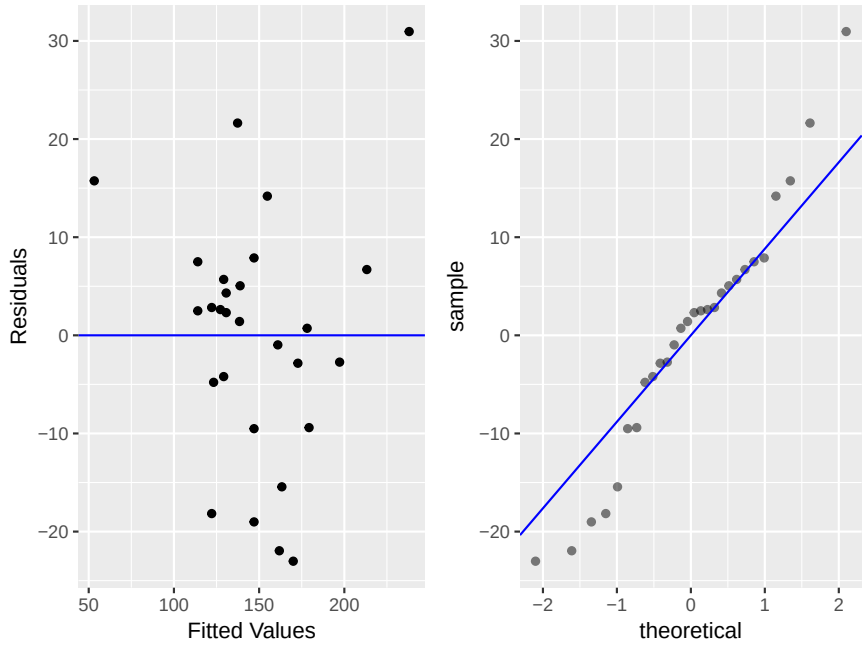

```
## The least squares regression equation is:
##  Draft.Number  = 225.009 - 0.226 Day.of.Year
## R^2 = 5.11%
```

```
summary(lm(Draft.Number~Day.of.Year))
```

```
##
## Call:
## lm(formula = Draft.Number ~ Day.of.Year)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -210.837  -85.629   -0.519   84.612  196.157
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 225.00922   10.81197  20.811  < 2e-16
## Day.of.Year  -0.22606    0.05106  -4.427 1.26e-05
##
## Residual standard error: 103.2 on 364 degrees of freedom
## Multiple R-squared:  0.05109,    Adjusted R-squared:  0.04849
## F-statistic:  19.6 on 1 and 364 DF,  p-value: 1.264e-05
```

- Multiple Regression

```
attach(houseprice)
mlr(Price, houseprice[, -1])
```




```
## The least squares regression equation is:
##  Price  =  -67.62 + 0.086 Sqfeet - 26.493 Floors - 9.286 Bedrooms + 37.381 Baths
## R^2 = 88.6%
```

```
summary(lm(Price ~ ., data=houseprice))
```

```
##
## Call:
## lm(formula = Price ~ ., data = houseprice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.018  -5.943   1.860   5.947  30.955
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -67.61984   17.70818  -3.819 0.000882
## Sqfeet        0.08571    0.01076   7.966 4.62e-08
## Floors      -26.49306    9.48952  -2.792 0.010363
## Bedrooms     -9.28622    6.82985  -1.360 0.187121
## Baths        37.38067   12.26436   3.048 0.005709
##
## Residual standard error: 13.71 on 23 degrees of freedom
## Multiple R-squared:  0.8862, Adjusted R-squared:  0.8665
## F-statistic:  44.8 on 4 and 23 DF,  p-value: 1.558e-10
```

- Best Subset Regression

```
library(leaps)
mallows(Price, houseprice[, -1])
```

```
##  Number of Variables Cp   Sqfeet Floors Bedrooms Baths
##  1                   8.83 X
##  2                   8.81 X                     X
##  3                   4.85 X      X              X
##  4                   5    X      X      X       X
```

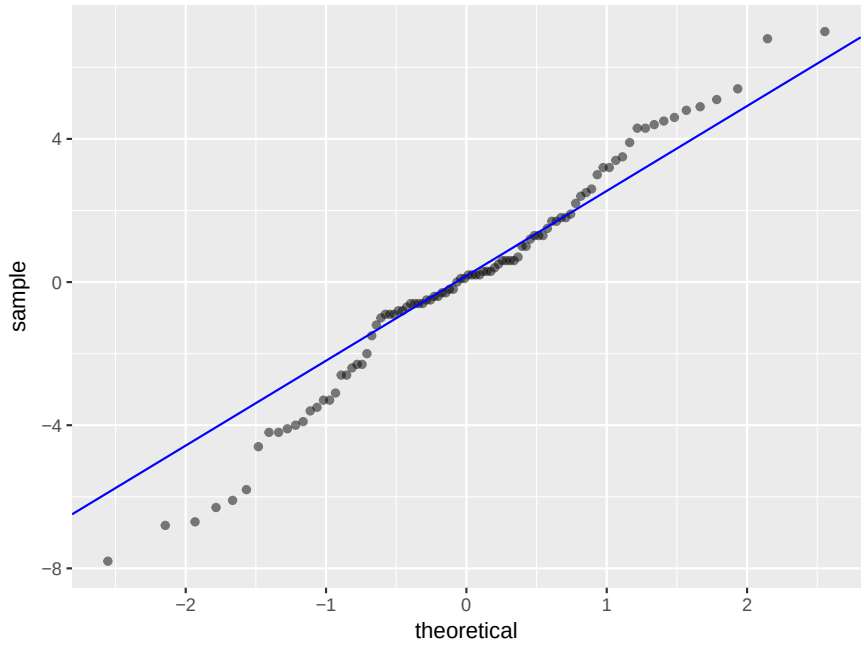
```
leaps(houseprice[, -1], Price)
```

```
## $which
##       1     2     3     4
## 1  TRUE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE  TRUE
## 1 FALSE FALSE  TRUE FALSE
## 1 FALSE  TRUE FALSE FALSE
## 2  TRUE FALSE FALSE  TRUE
## 2  TRUE  TRUE FALSE FALSE
## 2  TRUE FALSE  TRUE FALSE
## 2 FALSE FALSE  TRUE  TRUE
## 2 FALSE  TRUE FALSE  TRUE
## 2 FALSE  TRUE  TRUE FALSE
## 3  TRUE  TRUE FALSE  TRUE
## 3  TRUE FALSE  TRUE  TRUE
## 3  TRUE  TRUE  TRUE FALSE
## 3 FALSE  TRUE  TRUE  TRUE
## 4  TRUE  TRUE  TRUE  TRUE
##
## $label
## [1] "(Intercept)" "1"           "2"           "3"           "4"
##
## $size
##  [1] 2 2 2 2 3 3 3 3 3 4 4 4 4 5
##
## $Cp
##  [1]   8.834171  92.088525 104.303380 161.057329   8.812489  10.306028
##  [7]  10.812154  66.886236  77.214388  87.881962   4.848657  10.794275
## [13]  12.289752  66.450032   5.000000
```

6.3.5 ANOVA

- oneway

```
oneway(Length, Status)
```




```
## p value of test of equal means: p = 0.000
## Smallest sd:  2.5    Largest sd : 3.6
```

```
summary(aov(Length~Status))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Status       2  181.4   90.69   9.319 0.000208
## Residuals   91  885.6    9.73
```

- twoway

```
attach(gasoline)
twoway(MPG, Gasoline, Automobile)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            3 25.405   8.468  90.464 3.21e-13
## z            2  0.527   0.263   2.813   0.0799
## x:z          6  0.909   0.151   1.618   0.1854
## Residuals   24  2.247   0.094
##                   [,1]
## Gasoline  p =   0.0000
## Automobile  p = 0.0799
## Interaction p = 0.1854
```

```
G <- as.factor(Gasoline)
A <- as.factor(Automobile)
summary(aov(MPG ~ G * A))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## G            3 25.405   8.468  90.464 3.21e-13
## A            2  0.527   0.263   2.813   0.0799
```

```
## G:A          6  0.909   0.151   1.618   0.1854
## Residuals   24  2.247   0.094
```

```
twoway(MPG, Gasoline, Automobile, with.interaction = FALSE)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            3 25.405   8.468  80.510 1.89e-14
## z            2  0.527   0.263   2.504   0.0987
## Residuals   30  3.156   0.105
##                    [,1]
## Gasoline  p =   0.0000
## Automobile  p = 0.0987
```

```
summary(aov(MPG ~ G + A))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## G            3 25.405   8.468  80.510 1.89e-14
## A            2  0.527   0.263   2.504   0.0987
## Residuals   30  3.156   0.105
```

7 Introduction to ESMA 3102

This page discusses some general concepts of ESMA 3102.

7.1 3101 vs. 3102

In ESMA 3101 (3015) we were mainly concerned with answering questions about one variable at a time. We considered problems like these:

- What is the average height of men in Puerto Rico? (Find the mean or median, or draw a histogram or boxplot, or find a confidence interval)
- Are men in Puerto Rico on average taller than 5'10"? (do a hypothesis test)
- Has the average income in Puerto Rico gone up in the last ten years? (hypothesis test)

In ESMA 3102 we are going to study two (or more) variables simultaneously, and we are really interested in their relationships:

- Is the average height of men in Puerto Rico different from men in the USA and from men in Europe?
- How does the average height of men relate to things like their economic status (income), their race, their diet, et.
- How does the average income in Puerto Rico depend on the economic policies of the Government?

7.2 Categorical vs. Quantitative Variables

We categorize variables as follows:

7.2.1 Quantitative

data is numeric, and arithmetic makes sense (adding, multiplying etc.)

Examples:

1) Yearly income of a family in Puerto Rico
2) Temperature in Mayaguez at 12 Noon
3) Amount paid for the phone bill

7.2.2 Categorical

everything else

Examples:

1) A students major
2) in an experiment to grow wheat three different fertilizers were labeled 1,2 and 3
3) Your student id number

Note: often whether a variable is categorical or quantitative depends on how (and how precisely) it is measured.

Example Our variable is "rain yesterday"

- Did it rain at all yesterday? "Yes" or "No" $\rightarrow$ categorical
- We put a cup outside. The cup has marks for each cubic inch of rain. Our data is the number of cubic inches. Values will be 0, 0.1, 0. 2 etc. $\rightarrow$ quantitiative

Categorical data comes in one of two versions - ordered or unordered:

Examples

1) grades in a course: A, B, C, D, W - ordered
2) gender: Male, Female - unordered
3) Treatments in a clinical trial: A, B, C - unordered
4) Treatments in a clinical trial: 1, 2, 3 - unordered
5) blood pressure: low medium high - ordered
6) directions: north east south west - unordered

One consequence of having an ordering is that it should be used in graphs, tables etc.

Recognizing what the type of your data is has to be the first thing you do with any data set. It will determine everything that you do later. Getting this wrong likely means anything you do is wrong.

For more on data types see page 32 of the textbook.

7.3 Predictor - Response Paradigm

It is often useful to think of the problems we discuss in this class as trying to use one (or more) variables to predict another

Predictor	Response
Gender	Grade in Course
Gender	Income
GPA in high school, points on college	GPA after the freshmen year i
Whether fertilizer was used or not	Yield of crop
Size of lot, size of house, number of bedrooms, quality of neighborhood	Price of House

7.4 Types of Problems in 3102

Depending on the type of data we need to use different methods of analysis. Here is a table to help with this:

Predictor(s)	Response	Method
Categorical	Categorical	Categorical Data Analysis
All Categorical	Categorical	ANOVA
At least one quantitative	Quantitative	Correlation and Regression

7.4.1 Warning

This table maybe the most important item for you to learn - understand - memorize - use. Without it you can not pass this class, or do Statistics in real live!

8 Graphs

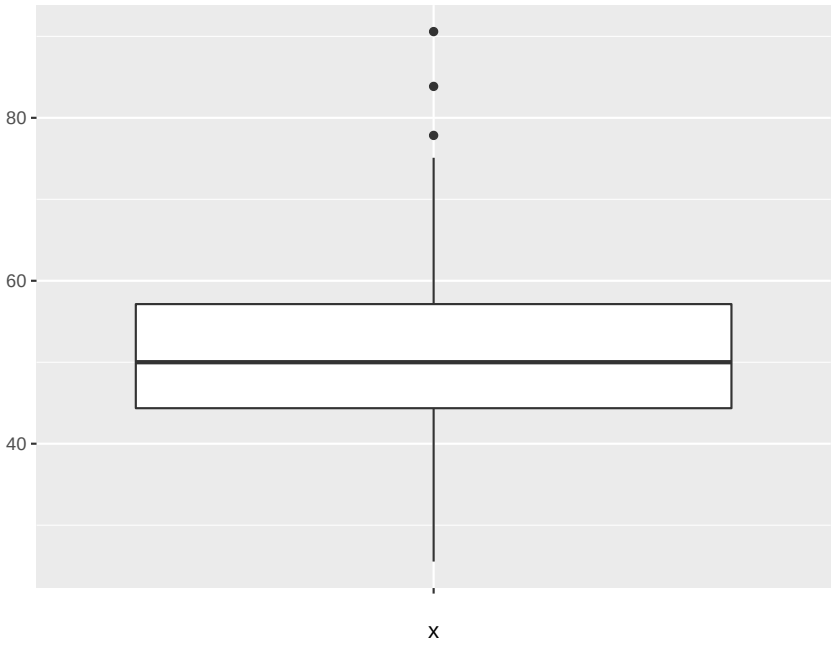
Graphs play a major role in Statistics. Here are some of the commonly used graphs:

8.1 Boxplot

This is a graph for quantitative data. It shows the numbers from the 5-number summary:

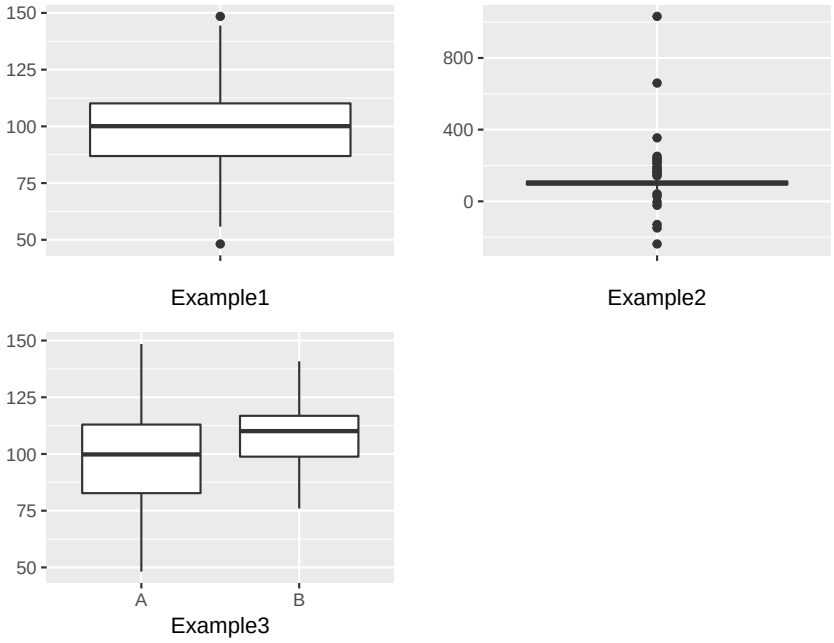
Minimum | Q1 | Median | Q3 | Maximum

plus some rules for identifying observations that are shown as stars. The boxplot is done with the command **bplot**. The first argument has to be the quantitative variable and the second one (if needed) the categorical one.



Here are several examples:

```
plt1 <- bplot(Example1, return.graph = TRUE)
plt2 <- bplot(Example2, return.graph = TRUE)
plt3 <- bplot(x, y, label_x="Example3", label_y="",
      return.graph = TRUE)
multiple.graphs(plt1, plt2, plt3)
```



note the use of the *multiple.graphs* command to combine several graphs into one. We also need the the additional argument return.graph = TRUE.

We use the boxplot mostly for these:

8.1.1 Are there any unusually large or small observations?

These appear as stars in the boxplot, compare the graph in the upper right corner with the one in the upper left corner.

Connected to this is the following question:

8.1.2 Do the observations come from a normal distribution?

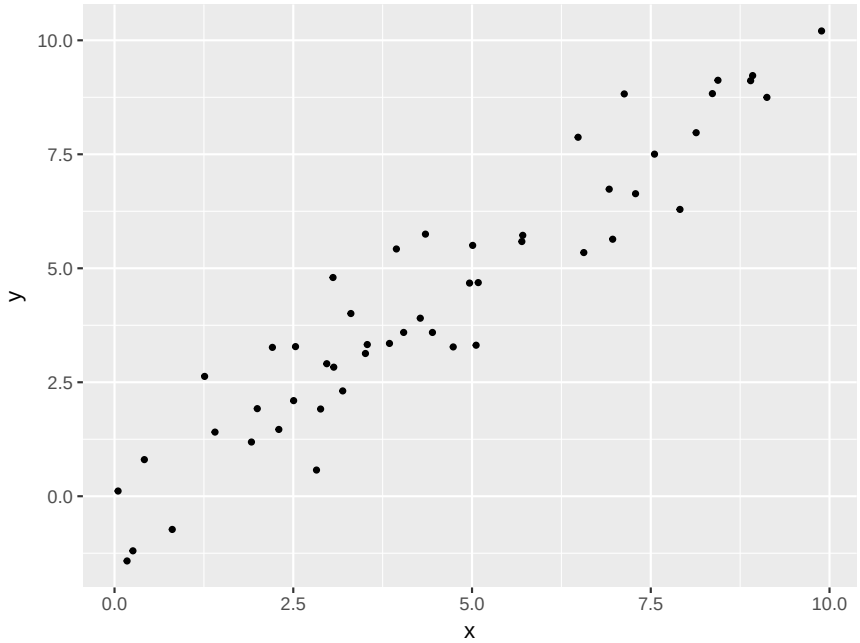
Most of the methods discussed in this course require that the observations of a variable come from a normal distribution. Problems with this assumption can often be seen in the boxplot. If there are stars (far away) from the box the normal assumption is wrong.

8.1.3 We have observations from several groups, how do they compare?

check out graph in lower left corner

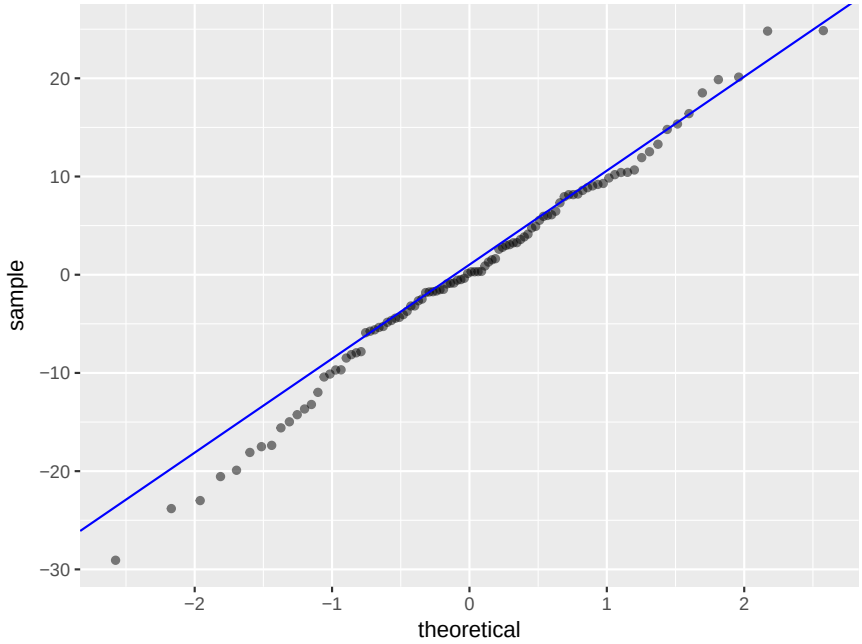
8.2 Scatterplot

This graph is for two quantitative variables. It is just a Cartesion coordinate system with the observations plotted as points. We have the command **splot(y, x)** to do the graph. Here y is the variable that goes on the y axis.



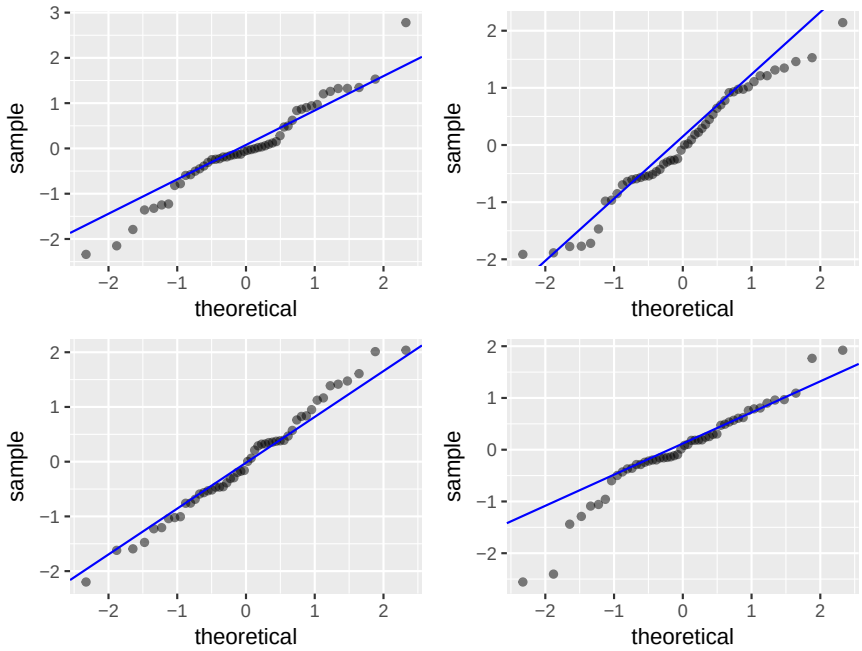
8.3 Normal Plot

This is a graph specifically designed to check whether the observations follow a normal distribution. If this is true the dots should (roughly) be on a line:

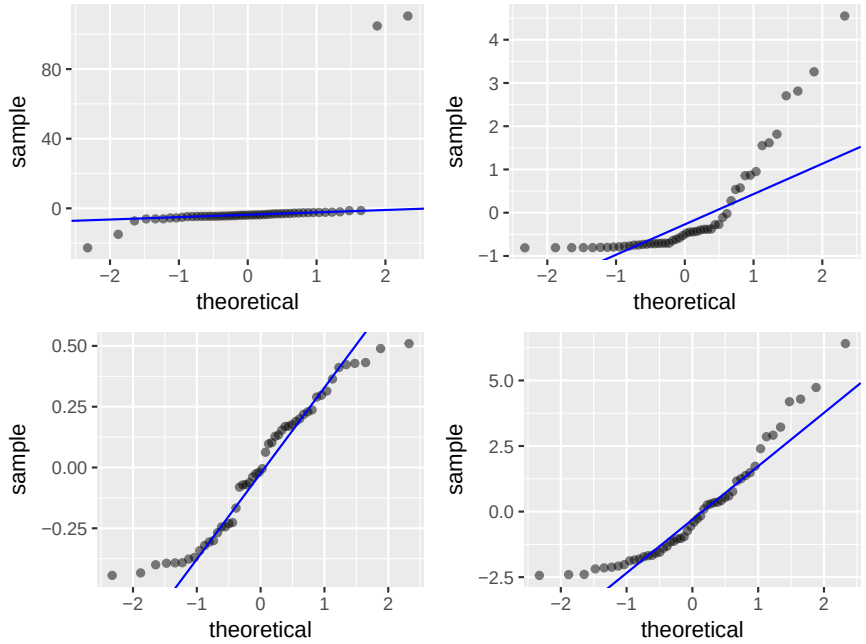


we have the command **nplot** to do the graph.

8.3.0.1 Normal Assumption is correct:



8.3.0.2 Normal Assumption is wrong:



8.4 Case Study: Euro Coins

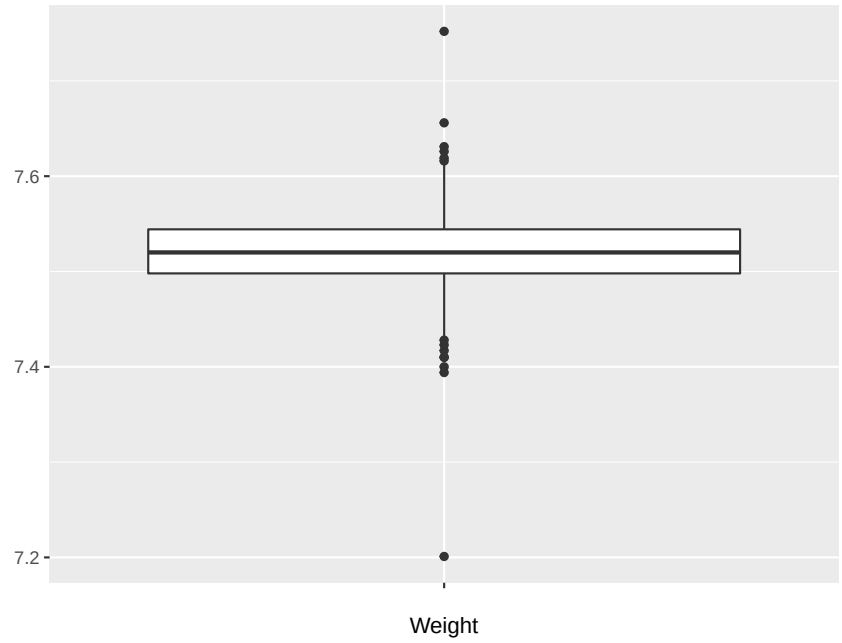
The data were collected by Herman Callaert at Hasselt University in Belgium. The euro coins were "borrowed" at a local bank. Two assistants, Sofie Bogaerts and Saskia Litiere weighted the coins one by one, in laboratory conditions on a weighing scale of the type Sartorius BP 310s.

Here is the boxplot and the normal plot of the weights:

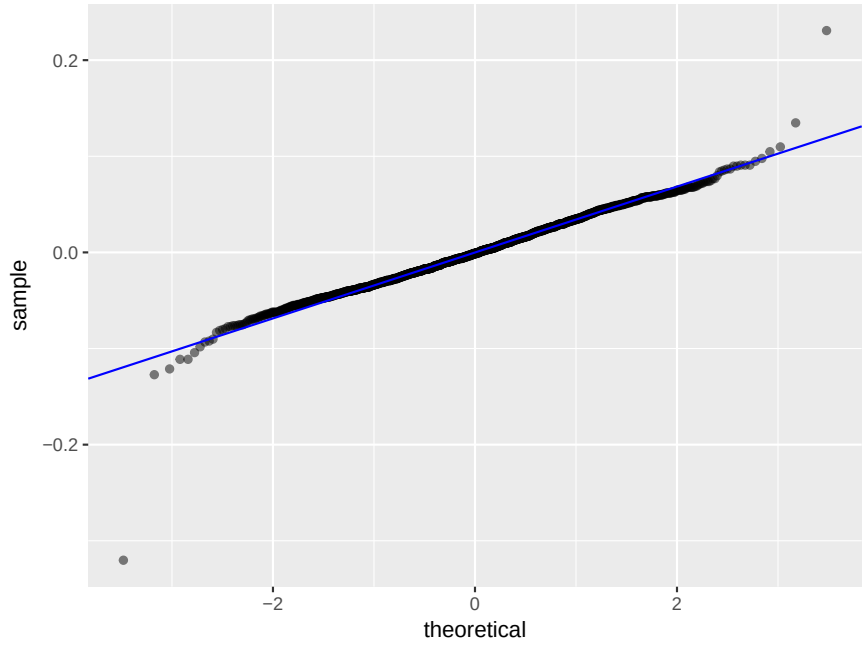
```
attach(euros)
head(euros)
```

```
##   Weight Roll
## 1  7.512    1
## 2  7.502    1
## 3  7.461    1
## 4  7.562    1
## 5  7.528    1
## 6  7.459    1
```

```
bplot(Weight)
```




```
nplot(Weight)
```

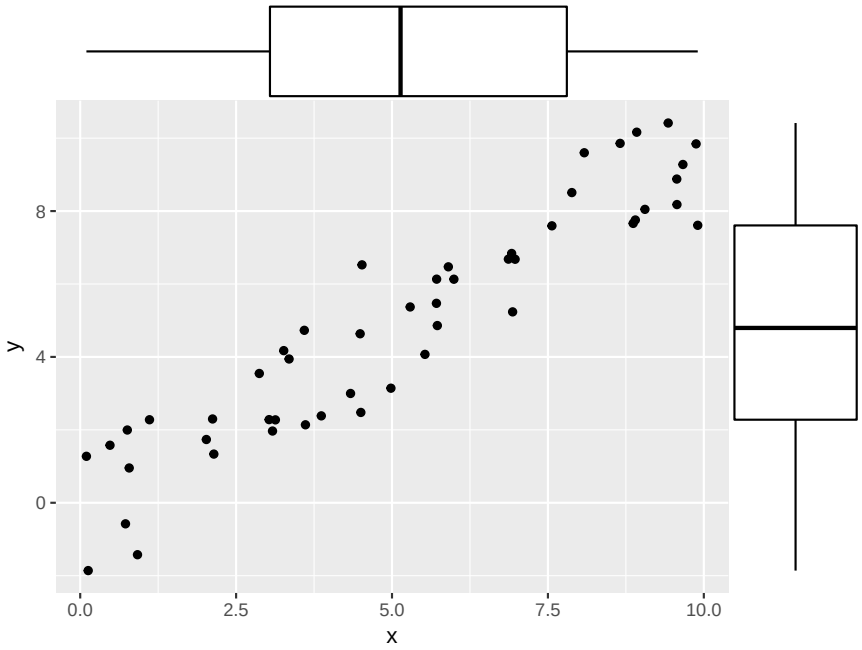


It appears that the weights do not come from a normal distribution.

8.5 Marginal Plot

There is a nice graph that combines a scatterplot and boxplots called the marginal plot. We can do it with the **mplot(y, x)** command:

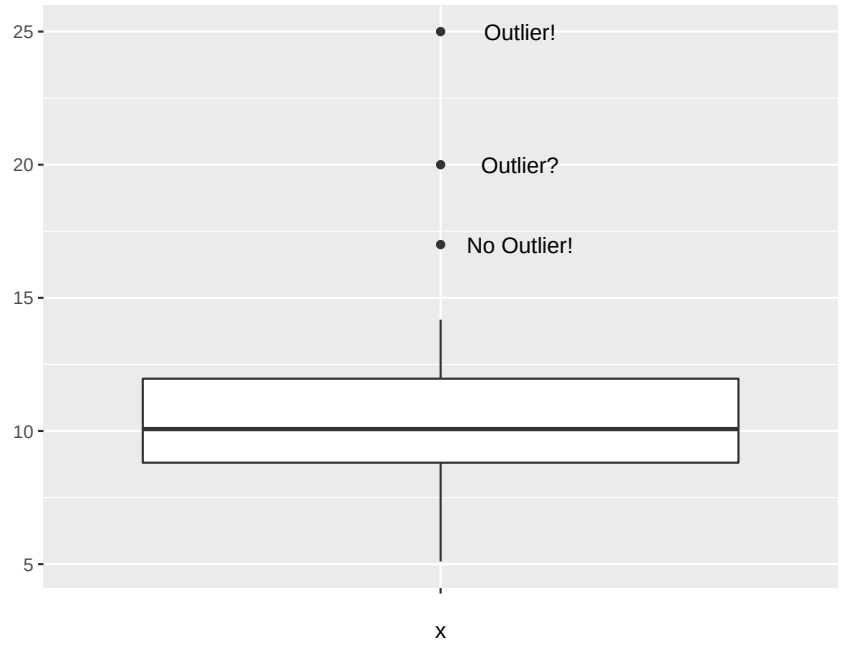
```
mplot(y, x)
```



9 Outliers - Detection and Treatment

Many of the methods discussed in this class don't work well if the dataset has **outliers**. An outlier is any observation that is in some way unusual/strange/weird.

We have already seen that an observation that is unusual with respect to one variable appears as a separate dot in an R boxplot:



Unfortunately there are no hard rules exactly when an observation becomes an outlier. To a large part that depends on the method of analysis we want to use, some methods are **sensitive** to outliers, others are more **robust**.

In addition to the case discussed above, there are other ways in which an observation can be an outlier:

9.0.1 Case Study: Alcohol vs. Tobacco Expenditure

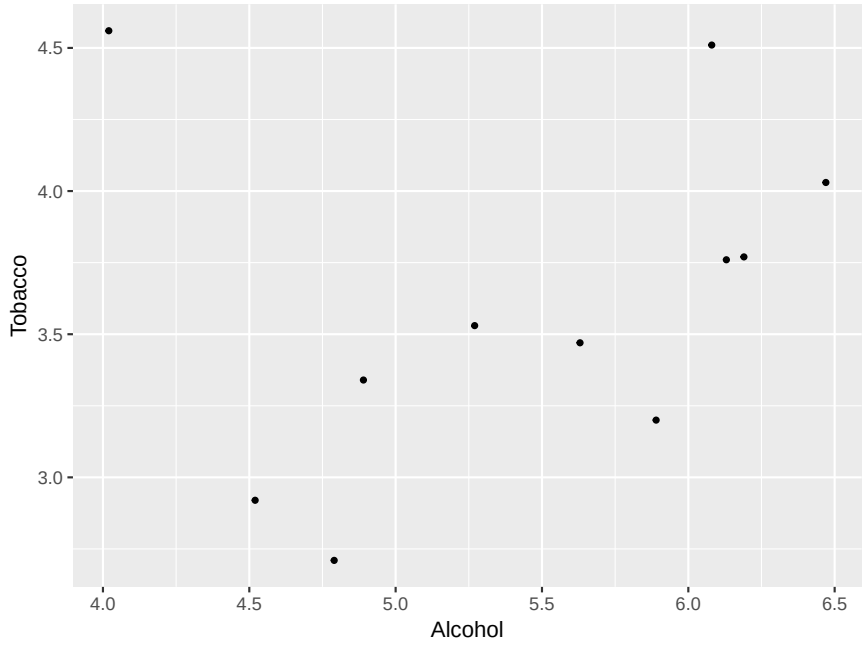
Data from a British government survey of household spending may be used to examine the relationship between household spending on tobacco products and alcoholic beverages. The numbers are the average expenditure for each of the 11 regions of England.

```
alcohol
```

```
##              Region Alcohol Tobacco
## 1             North    6.47    4.03
## 2         Yorkshire    6.13    3.76
## 3         Northeast    6.19    3.77
## 4     East_Midlands    4.89    3.34
## 5     West_Midlands    5.63    3.47
## 6       East_Anglia    4.52    2.92
## 7         Southeast    5.89    3.20
## 8         Southwest    4.79    2.71
## 9             Wales    5.27    3.53
## 10         Scotland    6.08    4.51
## 11 Northern_Ireland    4.02    4.56
```

Here we have two quantitative variables, so the obvious thing to do is draw the scatterplot:

```
attach(alcohol)
splot(Tobacco , Alcohol)
```



There seems to be generally a positive relationship, but also one case that does not fit. It seems it has the smallest value for Alcohol, and so we can easily find out which observation it is:

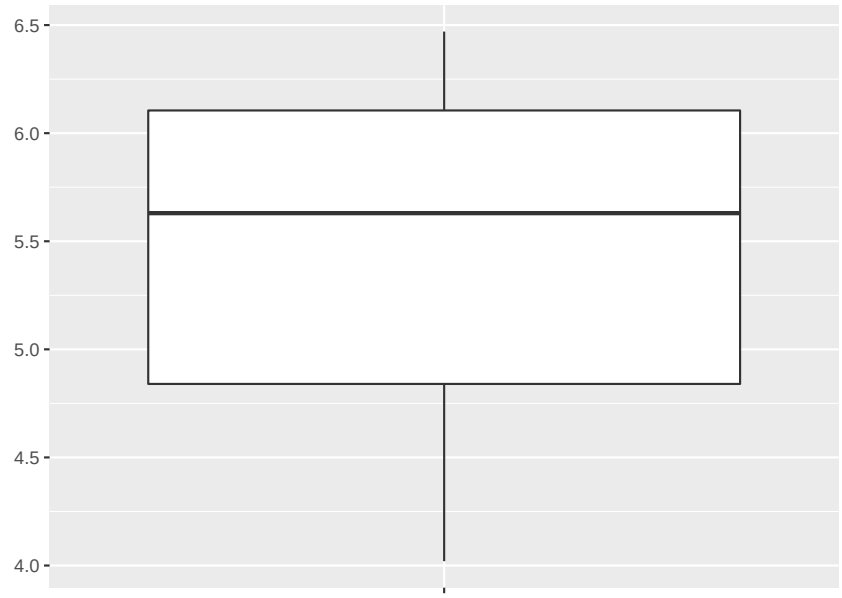
```
alcohol[which(Alcohol==min(Alcohol)), ]
```

```
##              Region Alcohol Tobacco
## 11 Northern_Ireland    4.02    4.56
```

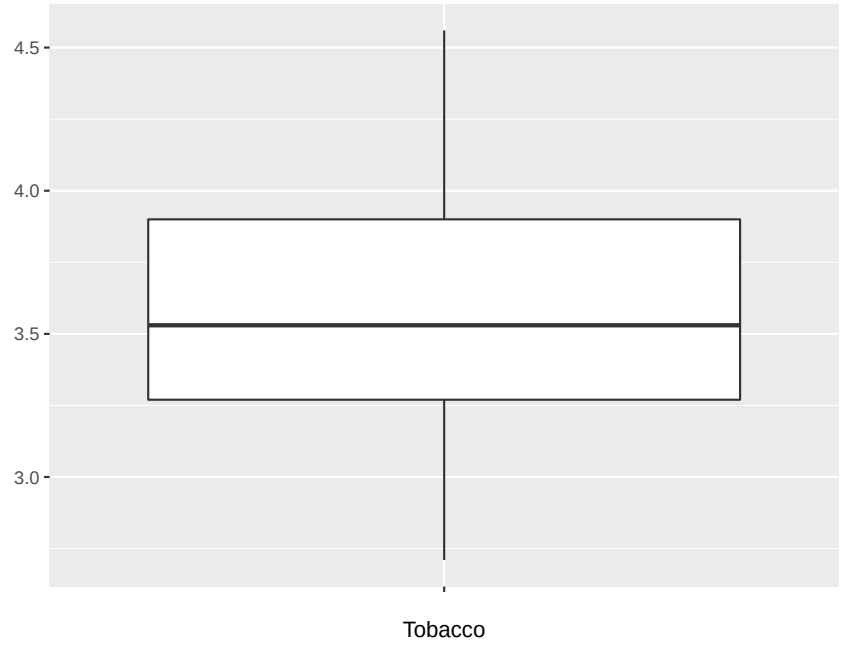
So it is Northern Ireland, where there is a fairly high expenditure on Tobacco but not on Alcohol (???)

Note that neither Alcohol nor Tobacco have any outliers by themselves:

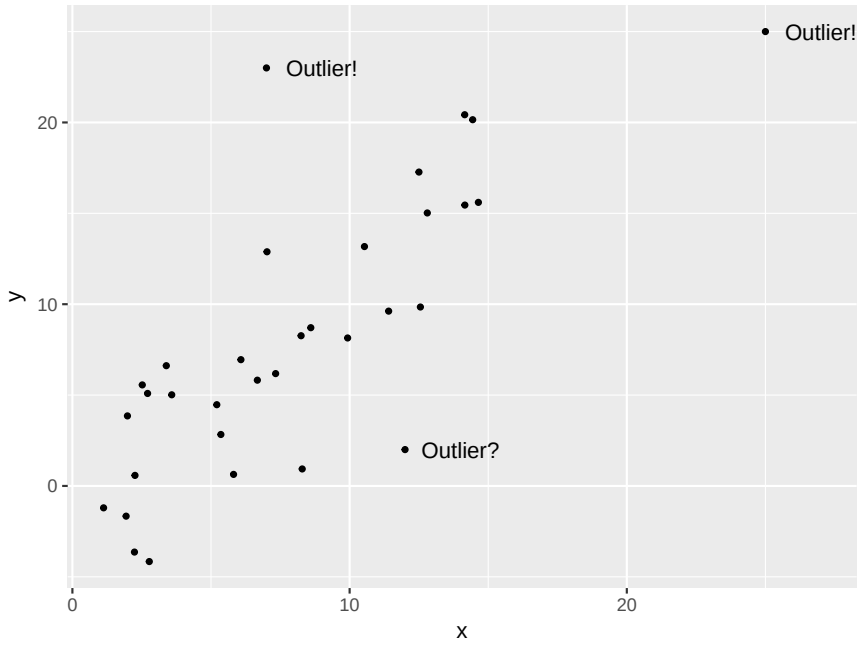
```
bplot(Alcohol)
```




```
bplot(Tobacco)
```



Again, it is not always obvious when an observation becomes an outlier:

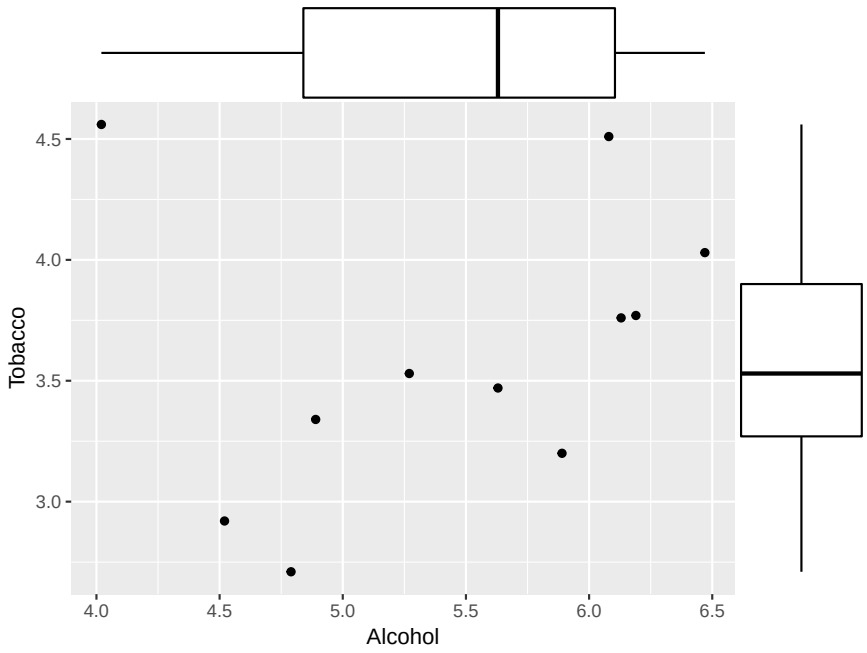


If we have two quantitative variables an outlier can happen in one of three ways:

- in the x variable, which we can check in the boxplot of x
- in the y variable, which we can check in the boxplot of y
- in the relationship between the xand the y variable, which we can check in the scatterplot of x and y

In fact we can do all three in one step:

```
mplot(Tobacco, Alcohol)
```



9.1 Treatment of Outliers

If we have an outlier in a dataset, what do we do then? First and foremost, **don't ignore them!** Most statistical methods are very sensitive to outliers, often they simply don't work.

Example Is there a relationship between Alcohol and Tobacco expenditures in England? Because we have two quantitative variables we might use Pearson's correlation coefficient to answer this question:

```
cor(Tobacco, Alcohol)
```

```
## [1] 0.2235721
```

```
cor(Tobacco[-11], Alcohol[-11])
```

```
## [1] 0.7842873
```

So with Northern Ireland we find a weak positive correlation, but without Northern Ireland it is a fairly strong positive correlation.

Which one is right? Clearly the first one is wrong because of the outlier!

So, if there are outliers, what do we do?

1. Learn as much as you can about the "story" behind the data and understand why there is an outlier. Is it an error? Is it something we should expect to see in this kind of data? etc.
2. Find a method that is not sensitive to outliers. For example, alternatives to Pearson's correlation coefficient include Spearman's rank correlation coefficient and Kendall's coefficient of concordance , although neither of them works any better here.
3. Try and "adjust" the outliers. We know what "caused" the Alcohol number for Northern Ireland to be off, so maybe we can adjust it.
4. If all else fails, eliminate the outlier(s)

10 Describing a Population: Probability Distributions

Recall the following definitions from Esma 3101:

Population: all of the entities (people, events, things etc.) that are the focus of a study

Sample: any subset of the population

Parameter: any numerical quantity associated with a population

Statistic: any numerical quantity associated with a sample

How can one describe a population? Sometimes this can be done by **enumeration (counting)**:

Example: Say we are interested in the age of the undergraduate students at the Colegio. So we go to the Registrar's office and ask them for help. They give us a computer file with the ages of all the students:

21 21 19 20 23 25 19 19 . . . ..

from this we can find a table:

Age	Counts
17	21
18	402
19	2109
20	2957
21	2089
22	1908
23	1105
24	788
25	208

and this table is a complete description of our population! Now it would be easy to find various numbers for this population:

- what percentage of the students is 19 years old?

2109/11587 *100% = 18.2%

- what is the 80^{th} percentile of the ages?

22 years

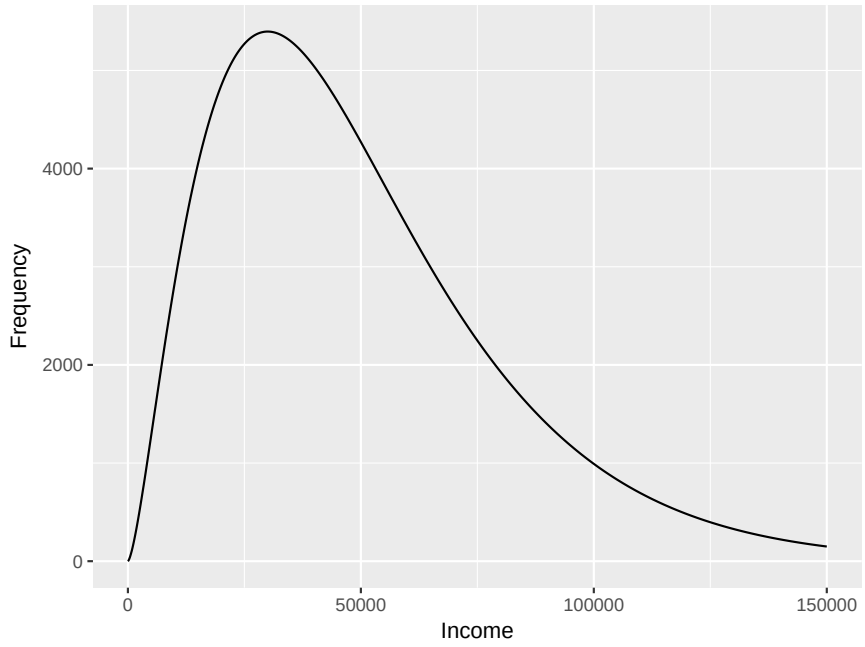
- what is the mean age of the students:

(17*21+18*402+..+25*208)/11587 = 20.9

These are numbers computed using the whole **population**, so these are **parameters**.

This simple way of describing a population works very rarely, in most cases we need to do something different:

Example: We want to study the incomes of families in Puerto Rico. Now it turns out that this kind of study has been done numerous times in many places, and Economists have worked hard to develop theories, and when we put it all together it seems reasonable that the **income distribution** in Puerto Rico looks like this:

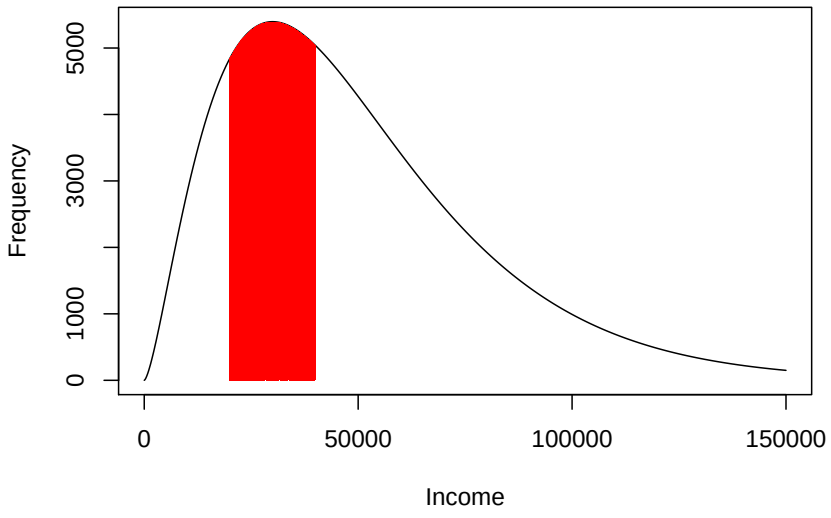


Here we are ignoring anyone with an income over $150000.

What is such a curve telling us? As before we can find from it many numbers of interest. For example probabilities about the population:

Say we randomly select a family in Puerto Rico. What is the probability that this family has an income between $20000 and $40000?

This probability is given by the area under the curve from 2 to 4:



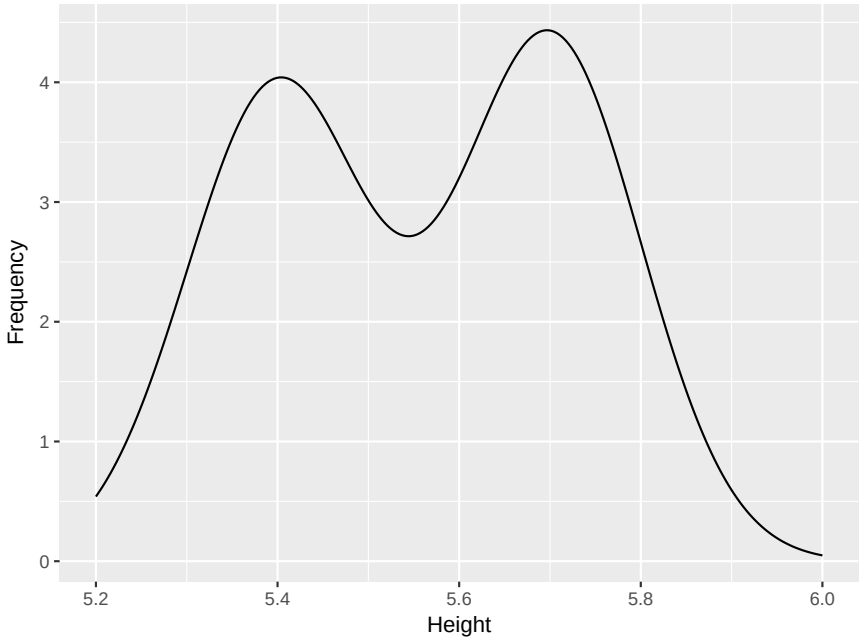
Finding this analytically would require some advanced math (the answer is 42.4%)

The basic idea here is: populations can be described by probability distributions, that is theoretical curves. Once such a distribution is known, anything one wants to know can be calculated from it (at least in theory, in practise the math might be quite difficult).

Notice that this is a straight-forward generalization of the enumeration method we used for the ages of the undergraduates, only there the math is very easy because we know how to find areas of rectangles!

Often in real live we have some theoretical reason to suspect that a certain distribution has a certain shape:

Example say we randomly select an undergraduate student from the Colegio and measure their height. Then in seems reasonable that the corresponding distribution looks like this:



Now life as a Statistician would be quite difficult if we would have to invent a new distribution everytime we study a new experiment. Instead it turns out that there are a number of basic distributions that we encounter time and again. These have been studied in great detail, and so we now have a lot of formulas easily available.

Example: say we randomly select an undergraduate student. We want to know whether the student is male or female.

The crucial part here is that there are only two possibilities: male or female. Any such experiment is called a **Bernoulli trial**. Here some other examples:

- student is in Arts&Sciences, or not
- student has a GPA of 3.5 or higher, or not
- student was born in Puerto Rico or not

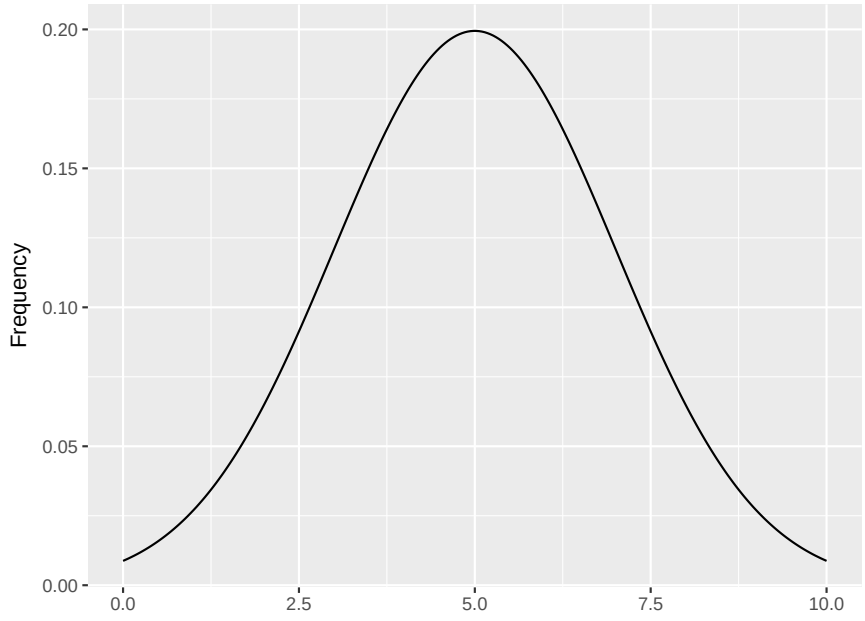
All these experiments share the feature that there are only two answers. Of course they differ in the probability that the answer is "yes":

- probability that a student is male: 50.8%
- probability that a student is a Arts & Sciences: 41.6%?
- student has a GPA of 3.5 or higher, or not: 21.8%
- student was born in Puerto Rico or not: no idea

And this is an important feature here: often from the description of the experiment we can make a guess what the general shape of the distribution is. The exact shape, though, depends on some numbers we often don't know. So what do we do? Statistics!

- student was born in Puerto Rico or not: we now need to find a sample of students, find the percentage of students born in Puerto Rico in the sample, and use that as a guess for the percentage in the whole population.

Although it is not at all obvious why this should be so, it turns out that the most important of these basic distributions is the **Normal Distribution**, characterized by a bell-shaped histogram:



A normal distribution has two parameters: the mean μ (which is where the peak of the curve is) and the standard deviation σ, which tells out how far out the curve goes.

11 Confidence Intervals

One way to find out what a statistic is telling us about the corresponding parameter is to find a **confidence interval**, that is a range of likely values for the parameter.

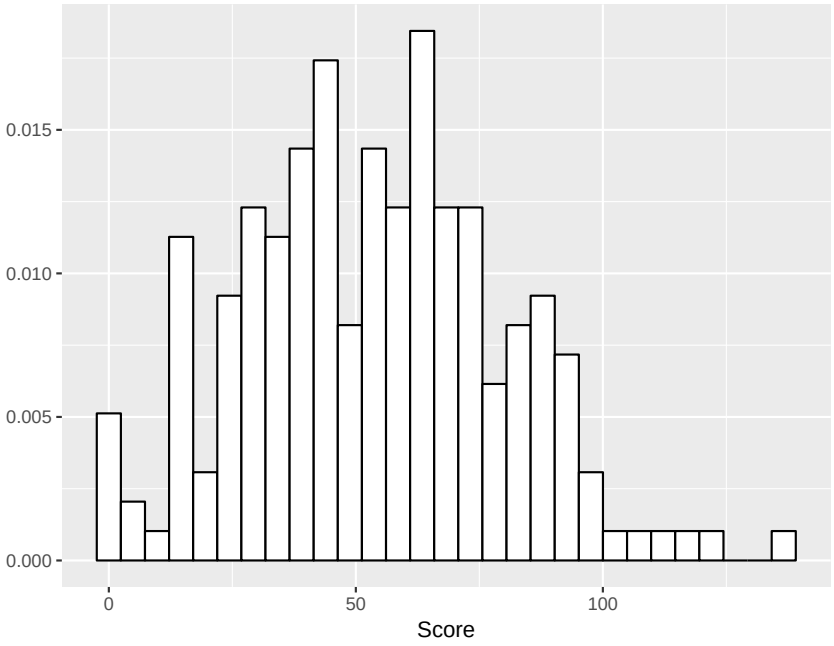
11.0.1 Case Study: Exam Scores

Here we will consider the variable Score, the score on two exams takes by students in a course What can we say about the average score?

```
attach(examscores)
head(examscores)
```

```
##     Exam Score
## 1 Exam 1   115
## 2 Exam 1    35
## 3 Exam 1    15
## 4 Exam 1    80
## 5 Exam 1    50
## 6 Exam 1    45
```

```
hplot(Score)
```



shows that the histogram of the scores is roughly bell-shaped, with the peak at around 50. More precisely we find

```
mean(Score)
```

```
## [1] 53.78
```

But this is based on exactly these 200 randomly selected scores, if we repeat the experiment we get different ones, and therefore a (somewhat) different mean. What can we say about the mean of all possible scores? (the **population** mean)

```
one.sample.t(Score)
```

```
## A 95% confidence interval for the population mean is (50.1, 57.4)
```

So what does it mean to say **a 95% confidence interval for the population mean score of (50.1, 57.4)**? The idea here is that we now know that with a high likelihood the true population mean score is between 50.1 and 57.4.

Notice I used the word **likelihood**. It is very tempting (and many people do it in real live) to use the word **probability** here.

Unfortunately for technical reasons to difficult for us to discuss this would be wrong!

The correct interpretation of a confidence interval is this: suppose that over the next year statisticians (and other people using statistics) all over the world compute 100,000 95% confidence intervals, some for the mean, others maybe for medians or standard deviations or ..., than about 95% or about 95,000 of those intervals will actually contain the parameter that is supposed to be estimated, the other 5,000 or so will not.

If the chance of getting a bad interval of 5% is to high, we can change that easily, say be finding a 99% confidence interval:

```
one.sample.t(Score, conf.level = 99)
```

```
## A 99% confidence interval for the population mean is (49, 58.6)
```

So this interval has a likelihood of 99% so we will get a wrong one just 1 in 100 times. But there is a price:

```
58.6 - 49
```

```
## [1] 9.6
```

```
57.4 - 50.1
```

```
## [1] 7.3
```

so this interval is larger than the 95% one, which means there is a larger uncertainty on exactly what the value is. Finding confidence intervals always involves a trade-off:

size of interval vs confidence level

11.0.2 App: confint

```
run.app(confint)
```

This illustrates the idea of confidence intervals.

What to do:

As the app starts the page on the right is empty, there is no data yet. In the panel on the left you can choose the population parameters that you want.

Next move the slider to 1. Now on the Single Experiment tab you get on simulated dataset, the Summary Statistics and the confidence interval calculations. You can now run the movie and see a sequence of simulated datasets.

You can also play around and see the effects of a

a) larger sample size n $\rightarrow$ smaller intervals

b) change population mean $\mu \rightarrow$ changes location of interval but not its size

c) increase population standard deviation $\sigma \rightarrow$ increases range of data, increases length of interval.

d) increase confidence level $\alpha \rightarrow$ increases length of interval.

2) on Many Experiment tab

no matter how n, μ or σ are changed, the percentage of good intervals always matches the chosen confidence level

12 Hypothesis Testing - An Introduction

Often a specific problem falls under the following very general heading:

we have a theory

Example 1: Is the new treatment better than the old one?

Example 2: Is the theory of evolution correct?

Example 3: Are coins fair?

and we want to do an experiment to see whether the theory is true

Example 1: Take a number of patients. Give some of them the old treatment and some of them the new one. See whether there is a difference.

Example 2: Ask a biologoist what would make a good experiment

Example 3: Flip the coin and count the number of heads and tails.

Finally we need to compare the results of the experiment with the theory to see whether they agree. This is what a hypothesis test does.

Now the first thing we need to recognize is

One of the principles of Science is that it is impossible to prove that a theory is correct but it is always possible to prove that the theory is false (a theory can be falsified)

Example 1: If the new treatment is actually better but only by a tiny little bit it might be impossible to be sure. If it is much better it will be obvious.

Example 2: Biologists have been inventing new ways to test the theory of evolution for 150 years. None of them has ever proven the theory wrong, but that does not mean the next one won't do so. (Even though by now that is very unlikely!)

Example 3: If we flip the coin 100 times and get 50 heads and 50 tails, does this prove the coin is fair? No because the same result is quite likely if the probability of heads where 0.51 (say). Of course by now we can be sure the coin is almost fair, but that is not the same as exactly fair!

Example 4: Theory: the tooth fairy does not exist

Now I am sure most of us are quite certain this theory is correct. After all no one has actually seen her. But maybe that is because she has lived on the planet Zoloff for the last 500 years? And of course, if she shows up on TV tomorrow (and can prove that she is indeed the tooth fairy) our theory has been proven false immediately!

Example 5: Theory: all swans are white

That is what everybody thought in Europe, until in the 18th century they found black swans in Australia!

Because of this a hypothesis test is set up so the data can proof the theory to be false:

Example 1: Null Hypothesis H_0: the new treatment is NOT better than the old one.

Example 2: Null Hypothesis H_0: the theory of evolution is correct

Example 3: Null Hypothesis H_0: the coin is fair

but NOT proving the theory is false is not the same as accepting the theory as true!

That is why we say we fail to reject the null hypothesis instead of just saying we accept the null hypothesis.

Let's have a closer look at example 3. Here the experiment to check the theory is very simple, in fact I have a shiny app that will do it for us:

12.0.1 App: coin

```
run.app(coin)
```

The app flips a coin 100 times and shows the results. By default it is a fair coin (p=0.5) but we can change that on the left. Recall we have

$$H_0 : p = 0.5 \text{ (coin is fair)}$$
$$H_a : p \neq 0.5 \text{ (coin is not fair)}$$

Next we can decide on the **Rejection Region**:

In this experiment we would reject the theory of a fair coin if the number of heads is far from 50.

What do you think this should be? Set the sliders accordingly.

Now click the Run! button and repeat the experiment 20 times. Each time you can see the number of heads and whether or not we would reject the theory.

Doing this one at a time is a bit slow, and we really should do this many times, so switch to the Many Runs tab. Here we see the results of 100000 runs of this experiment.

Now move the sliders for the Rejection Region to 45 and 55. The cases in blue are those where the 100 flips resulted in a number of heads between 45 aand 55, and we would not reject the theory of a fair coin.

In red we have all the cases with either less than 45 or more than 55 heads, and so here we would reject the theory of a fair coin. As we see that happens about 27% of the time.

But we are still flipping a fair coin, so we should not reject the theory at all, doing so is an error. Soon we will call this the **type I error**. The 27% will be called the **type I error probability** α.

Committing an error in about 1 in 4 cases (~27%) does not sound like a good idea. So let's make this much smaller. Move the slider to 40-60, and then then we have $\alpha = 3.5\%$, much better.

But there is also a downside to this. Let's select a **Slightly unfair (p=0.6)** coin. Now the coin is NOT fair, and we should reject the theory. But we are doing so only 46% of the time, the other 54% of the runs wrongly make the theory look ok.

This mistake is called the **type II error**. The 54% is called the **type II error probability** β.

The percentage of runs that correctly reject the theory is called the **power** of the test.

Now if we go back to Rejection Region 45-55 the percentage of correctly rejected false theories goes up to 82.3%, much better.

Of course in real life we do not know whether the coin is fair or not, so we how do we choose the Rejection Region? We do it by choosing a type I error probability α that seems acceptable to us. Often this is about 5%, and that leads to 41-59.

Once we have decided on α we can the do some math to see what the β might be, but this will depend on how unfair the coin might be.

13 Hypothesis Testing

13.0.1 Case Study: A New Treatment for Skin Cancer

Let's consider the following (artificial) example. A pharmaceutical company has developed a new treatment for a certain type of skin cancer. In order for the treatment to be approved by the Food and Drug Administration for use they have to show it is save (that is it has only acceptable side-effects) and it works (that is it is as better than the existing drugs). Let's say they have shown the safety, and now want to show the effectiveness.

How can they do that?

Obviously they need to carry out a clinical trial: find a number of people with this type of skin cancer, give them the new treatment and see what happens.

Question: how many subjects do they need?

Let's say that they decide to use 40 subjects.

What should they measure? In general there are many possibilities: time until cure, time until death, number of cancer cells, some feature of the blood, etc. Let's say this is a very agressive form of skin cancer which usually is deadly, so they will measure the time until death. They find that the subjects in the study survive on average **517** days.

Is this a long time? By itself this question cannot be answered, we need something to compare it with. Let's say that using the current best treatment the survival time is known to have a mean of **485** days. So this looks good for the new treatment: 517 days is better than 485 days.

But the 517 days is the **sample mean** of this specific sample of 40 people, if we randomly choose another group of 40 subjects and give them the new treatment, their mean survival is NOT going to be 517 days again. It might be lower but it also might be higher.

Could it be as low as 485 days?

Is the difference of 517-485 = 32 days due to the fact that the new treatment is better than the old one, or is it due to random fluctutation?

In very general terms what we have here is a question which has an answer of either **yes or no**. Yes, the new treatment is better than the old one or No, it is not. In Statistics, if we have such a yes-no question we usually answer it by doing a **hypothesis test**.

Let's assume for the moment that the new treatment is NOT better than the old one, so the increase of 32 days was just due to random chance. Let's find the probability that if we then select a new group of 40 subjects and use the new treatment, their mean survival time is again 517 days, or even longer?

This probability is called the **p-value** of the the test.

If this probability is small (?) one of two events happened:

- our experiment had an outcome that was very unlikely
- our assumption that the new treatment is not better is false. The new treatment is better, and that is why the subjects lived longer.

Let's say the p-value is small, and so the hypothesis test says that the new treatment is better than the old one. But maybe the p-value was small because in our specific group of 40 subjects several survived a long time, so their mean survival time was unusually high.

Say that in reality (but unknown to us!) the new treatment is NOT better than the old one, but because of random fluctutation the test decided wrong. This means we committed the **type I error**.

In a hypothesis test we decide before the test is done how small a p-value will lead us to reject the null hypothesis. This cut-off probability is usually denoted by α and is often chose as 5% (but not always!)

Of course it might also work out the other way: let's say the p-value is NOT so small, and so the hypothesis test says that the new treatment is NOT better than the old one. But

maybe the p-value was not so small because in our specific group of 40 subjects several died quickly for other reasons, so their mean survival time was unusually low. Actually, the new treatment is better than the old one, but because of random fluctutation the test decided wrong. This means we committed the **type II error**.

So you see that this hypothesis test has a lot of moving parts: the parameter of the test (here the mean time of survival), the value for the sample (517), the value to which we should compare this (485), the acceptable type I error α, type I and type II errors, their probabilitites, the p-value, the decision of the test (accept or reject).

13.1 Parts of a Hypothesis Test

1) Parameter of interest

2) Method of analysis

3) Assumptions of Method

4) Type I error probability α

5) Null hypothesis H_0 (in plain language and in terms of a parameter, if appropriate)

6) Alternative hypothesis H_a

7) Find p value (using R)

8) Decision and Conclusion,**in plain language**.

The decision on whether or not to reject the null hypothesis is easy:

p $< \alpha \rightarrow$ reject H_0

p $> \alpha \rightarrow$ fail to reject H_0

Example: We flip a coin 100 times and get 58 heads.

Question: Is this a fair coin?

1) Parameters of interest: a proportion (or percentage or probability)

2) Method of analysis: 1 proportion

3) Assumptions of Method: none

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis $H_0 : \pi = 0.5$ (coin is fair)

6) Alternative hypothesis $H_a : \pi \neq 0.5$ (coin is not fair)

7) p value = 0.133

8) $0.133 > 0.05$, so we fail to reject H_0, the coin could be fair.

13.2 More on the p-value

The p-value is the probability of repeating the experiment and observing the same result or something even more unlikely (assuming the null hypothesis is true).

Example: coin: p value is the probability to flip the coin again 100 times and get again 58 head or even more **if in truth the coin is fair**.

If the p-value is **small** (say <0.05) we should reject the null hypothesis!

Note: actually because we are testing the alternative $\pi \neq 0.5$ the p value is the probability of number of heads ≥ 58 or ≤ 42, but those are technical details which R will take care of for us.

13.2.1 App pvalue

```
run.app(pvalue)
```

This app illustrates the concept of the p value. The parameter of interest here is the mean μ.

As the app starts the page on the right is shows the chosen type I error probability α, the null and the alternative hypothesis. There is no data yet.

This illustrates one important fact about hypothesis testing: α, H_0 and H_a do NOT depend on the data, they come from the problem/experiment we are working on.

Next move the slider to 1. Now on the Single Experiment tab you get one simulated dataset (from a normal distribution with mean μ and standard deviation σ), the p-value of the corresponding test and the decision on the test (reject/ fail to reject H_0). You can now run the movie and see a sequence of simulated datasets.

Switch to the Many Experiment tab

Case I: μ=10, H_0 is true This shows the histogram of 1000 hypothesis tests just like the one on the Single Experiment tab. In each test if $p < \alpha$ (drawn in red) we reject H_0, otherwise we fail to reject H_0. The app shows the percentage of tests with $p < \alpha$, which should be close to α!

Changing the sample size n or the population standard deviation σ does not change any of this.

Changing α changes the percentage of rejected tests so that it always matches α.

Case I: $\mu \neq 10$, H_0 is false

Move slider to $\mu = 11.0$. Now the number of tests with $p < \alpha$ is much higher (63%), which is good because this means we would correctly reject this false H_0 63% of the time. Move the slider to $\mu = 12.0$ and now almost all the test have $p < \alpha$.

Move slider to μ=11.0 and see that

- larger sample size n (=100) $\rightarrow$ reject more tests (91.3% vs 63%)
- increase population standard deviation σ (=9) $\rightarrow$ reject fewer tests (12.4% vs 63%)

d) increase confidence level $\alpha = 0.1 \rightarrow$ reject more tests (74.4% vs 63%)

13.3 What you can conclude from the outcome of a hypothesis test

After carrying out a hypothesis test, what can you conclude? There are always the following possibilities:

- If we rejected the null hypothesis:
  — H_0 is in fact false
  and we made the correct decision
  — H_0 is in fact true and we made the wrong decision, so we committed the type I error (but we know the probability of doing so - α)
- If we failed to reject the null hypothesis:
  — H_0 is actually true
  and so we made the correct decision
  — H_0 is in fact false, so we made the wrong decision, we committed the type II error
  — H_0 is in fact false, so we made the wrong decision, specifically because our sample size was to small!

13.3.1 "fail to reject H_0" vs "accept H_0"

If we carry out a hypothesis test and at the end find pα, we say that we** fail to reject **the null hypothesis. Why do we not just say that we** accept **the null hypothesis? Let's illustrate the difference with an example:

Example say you pick a coin out of your pocket. It is a perfectly ordinary coin, but for some reason you wish to test whether the probability of heads is 0.4. You flip the coin 25 times and find 13 heads (a quite realistic outcome). Now we have

```
one.sample.prop(13, 25, pi.null=0.4)
```

```
## p value of test H0: pi=0.4 vs. Ha: pi <> 0.4:  0.3074
```

1) Parameter of interest: 1 Proportion π

2) Method of analysis: test for binomial π

3) Assumptions of Method: none

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis H_0: $\pi = 0.4$

6) Alternative hypothesis H_a: π $ne 0.4$

7) p-value = 0.2273

8) $p > \alpha$, we fail to reject the null hypothesis

so, should we now conclude that $\pi = 0.4$? Remember, this is a completely normal coin, almost certainly a fair coin with (just about) $\pi = 0.5$, not 0.4 So the null hypothesis is pretty much certain to be wrong, and us "accepting" it would be wrong!

The problem is of course that flipping the coin just 25 times was not enough. Say we had flipped the coin 100 times and got 52 heads (for the same proportion 13/25 = 52/100), but now p-value=0.0184 $< \alpha$ and we would reject H_0. So

We never accept a null hypothesis, we can only fail to reject a null hypothesis

13.3.2 Power of a Test

One of the most important considerations in a hypothesis test is its **power**. It is defined as the probability to **CORRECTLY** reject a **FALSE** null hypothesis.

13.3.3 App power

```
run.app(power)
```

The app generates n observations from a normal distribution with mean μ and standard deviation σ. Then it does the test

H_0: μ=10.0 vs H_a: μ>10.0

To start μ=11.0, so H_0 is false and should be rejected. Run the Movie and see what happens if we do this 100 times. About half the time we make the right decision, and so the power of this test is 50%. Select the Show Power Curve button, and you get the theoretical power curve with the actual power, closely matching the simulation result.

Now you can change the situation by changing the true μ to 12.0, the standard deviation from 3 to 5, the sample size from 25 to 50 and the type I error probability α from 5% to 10%. Observe how each of these changes affects the power.

The power of a test is (among other things) a tool that helps us when we are planning an experiment. It can help us understand whether a certain experiment is likely to be successful (statistically!) and it can help us decide how large the experiment needs to be (the sample size)

14 The Lady tasting tea

In 1935 Sir R.A. Fisher wrote a book with the title *The Design of Experiments*. This book and several others that he wrote were so important that today Fisher is often called the father of Statistics.

In the book he tells the following story: one day one of his collegues at the Rothemstead Experimental Station, Muriel Bristol (Ph.D), claimed she could tell whether in a cup of tea the tea had been poured into the cup before the milk, or vice versa.

Fisher devised an experiment to test that claim as follows: He filled eight identical cups with milk and tea, four with the milk first and four with the tea first. Then he randomly put them on a table and asked Muriel to pick the four with the tea poured first. Muriel was told the experimental setup, so she knew there were four cups of each kind.

What can we say about this experiment? Let's write down one possible arrangement. Here T is a cup where the tea has been poured first (of course without Muriel knowing this!), whereas M is one with the milk first:

T M M T T T M M

Let's also say that the first four cups are those the Lady has identified as the one with the milk poured first. So in this case she got two correct and two wrong. Not very good!

Of course this is what we would expect to see if indeed the Lady knows what she is doing:

M M M M T T T T

Now Fisher decided to only accept Muriels claim if indeed she could identify all four cups with milk poured first correctly. If she was just guessing, how likely was it she would get that lucky? Well, how many possible arrangements of the cups she picks are there? Here they are:

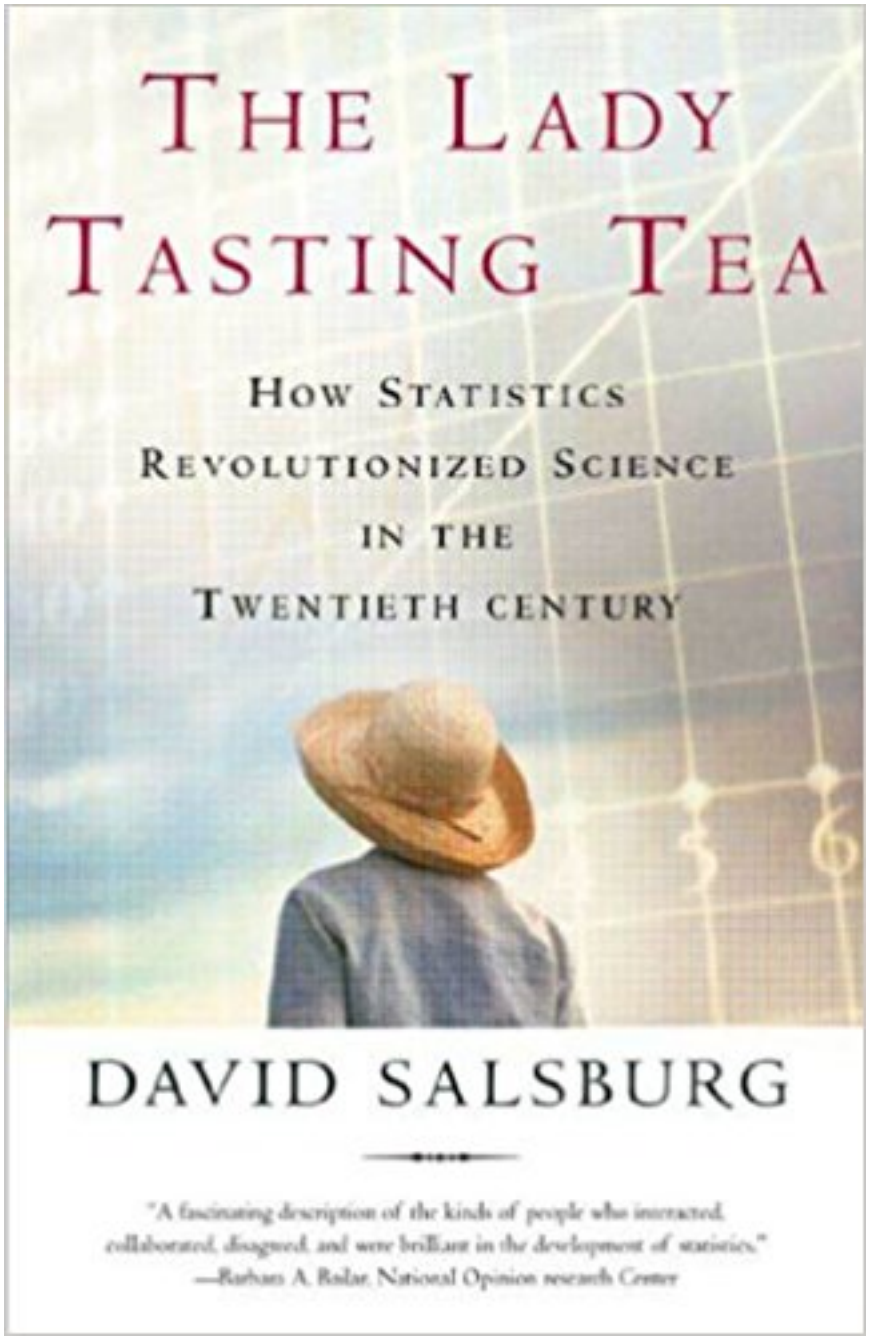

THE LADY TASTING TEA
HOW STATISTICS REVOLUTIONIZED SCIENCE IN THE TWENTIETH CENTURY
DAVID SALSBURG
"A fascinating description of the kinds of people who interacted, collaborated, disagreed, and were brilliant in the development of statistics."
—Barbara A. Bailar, National Opinion research Center

Figure 4:

M M M M	M M &
M M M T	M M &
M M M T	M M &
M M M T	M M &
M M M T	M M &
M M M T	M M &
M M M T	M M &
M M M T	M T &
M M M T	M T &
M M T T	M T &
M M T T	M T &
M M T T	M M &
M M T T	M M &
M M T T	M M &
M M T T	M M &
M M M T	M M &
M M M T	M M &
M M M T	M M &
M M M T	M M &
M M T T	M M &
M M T T	M M &
M M T T	M M &
M M T T	M M &
M M T T	M M &

Want to count them? Sometimes it helps to know some math: if there are $2n$ cups there are $\binom{2n}{n}$ such arrangements, or here:

$$\binom{8}{4} = \frac{8!}{4!4!} = \frac{40320}{24 \times 24} = 70$$

only the first one (M M M M) is correct, so her chances of getting it right if she were to just guess randomly were $\frac{1}{70} = 0.0143$.

Let's view this experiment in light of our previous discussion on hypothesis testing:

14.0.1 1) Parameter of Interest

One can view this as an experiment with two variables (actually tea first/actually milk first and identified as tea first/identified as milk first) and whether the two are independent or not. In this sense the parameter is a correlation, or as we say for categorical data, an association.

14.0.2 2) Method of Analysis

this idea of counting the total number of possible answers is now know as Fisher's Exact test.

14.0.3 3) Assumptions of the Method

the experiment has to be set up as described above. For example, it is very important that Muriel knew that exactly four cups had the milk poured first.

14.0.4 4) Type I error probability α

$\alpha = 0.05$

14.0.5 5) Null hypothesis H_0

H_0: Muriel is just guessing.

Notice that common feature of many hypothesis tests, namely to pick the "negative option" as the null.

14.0.6 6) Alternative hypothesis H_a

This is where it gets interesting, because there isn't one!

The idea of an alternative hypothesis was invented a bit later by Jerzy Neyman and Egon Pearson (son of Karl Pearson of correlation fame). Fisher never liked it. They had some very good fights over this!

One consequence of not having an alternative hypothesis is that one can not find the power of the test.

14.0.7 7) p-value

So, how did Muriel do? In fact she was perfect, she got all eight cups correct! Therefore we have $p = 1/70 = 0.0143$.

14.0.8 8) Decision of the test

$p = 0.0143 < 0.05 = \alpha$, and so we reject the null hypothesis.

14.0.9 9) Conclusion

Muriel certainly proved her claim.

Figure 5:

14.0.10 Type I error:

Type I error =

reject the null hypothesis although it is true =

conclude that Muriel knows what she is doing although she was just guessing.

14.0.11 Type II error:

Type II error =

fail to reject the null hypothesis although it is false =

conclude that Muriel was lying although she actually knows her tea (but unfortunately made a mistake).

14.1 Historical Importance

Using this simple experiment, Fisher established most of the fundamental principles for hypothesis testing, which contributed to major advances across biological and physical sciences. A careful read of the original text shows a precise use of terms, in a concise and unambiguous presentation, in contrast with many textbooks written later that were more confusing than helpful.

15 Methods for Univariate Data

15.1 Population Mean

The R command for testing and to find confidence intervals for a population mean μ is *one.sample.t*. The assumptions for this method are: either the data comes from a normal distribution or the sample size is large enough.

15.1.1 Case Study: Resting Period of Monarch Butterflies

Some Monarch butterflies fly early in the day, others somewhat later. After the flight they have to rest for a short period. It has been theorized that the resting period (RIP) of butterflies

flying early in the morning is shorter because this is a thermoregulatory mechanism, and it is cooler in the mornings. The mean RIP of all Monarch butterlfies is 133 sec. Test the theory at the 10% level.

Research by Anson Lui, Resting period of early and late flying Monarch butterflies Danaeus plexippus, 1997

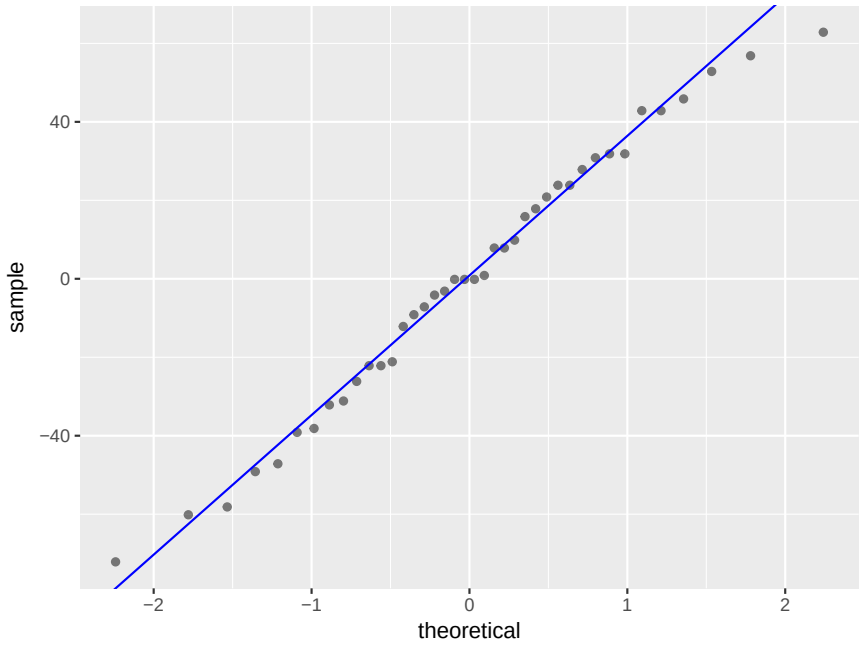
A) find a 90% confidence interval for the mean length of the flight times.

First we check the assumptions:

```
attach(butterflies)
sort(RIP.sec.)
```

```
##  [1]  52  64  66  75  77  85  86  92  93  98 102 102 103 112 115 117 120
## [18] 121 124 124 124 125 132 132 134 140 142 145 148 148 152 155 156 156
## [35] 167 167 170 177 181 187
```

```
nplot(RIP.sec.)
```



The graph shows that the data comes from a normal distribution.

```
one.sample.t(RIP.sec., conf.level = 90)
```

```
## A 90% confidence interval for the population mean is (115, 133.3)
```

Note the one.sample.t command does the normal plot automatically.

B) test at the 5% level whether the true mean resting period is less than 133 minutes.

```
one.sample.t(RIP.sec., mu.null = 133, alternative = "less")
```

```
## p value of test H0: mu=133 vs. Ha: mu < 133:  0.0558
```

1) Parameter of interest: population mean
2) Method of analysis: one sample t
3) Assumptions of Method: normal data or large sample
4) $\alpha = 0.05$

5) H_0: $\mu = 133$

6) H_a: $\mu < 133$

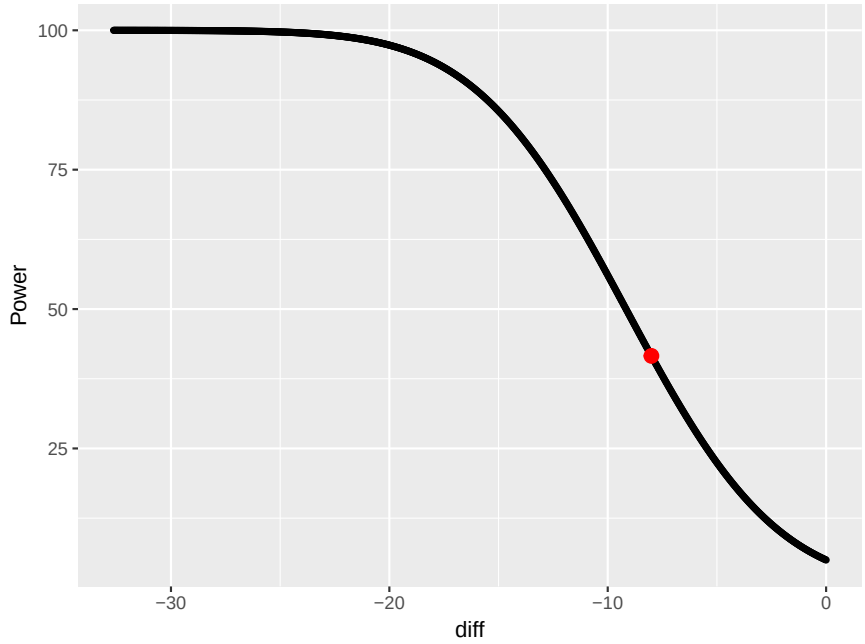
7) p = 0.0558

8) $p > \alpha$, we fail to reject the null hypothesis, the waiting time might well be 133 minutes.

C) for the test in B) what was the power of the test if the true resting period is 125 minutes?

We have the routine t.ps for power and sample size calculations:

```
t.ps(n = 40, diff = 125-133, sigma = sd(RIP.sec.),
     alternative="less")
```




```
## Power of Test = 41.6%
```

Note for this method the actual values of 125 and 133 are irrelevant, only the difference 125-133 matters!

Note the argument name for the standard deviation is sigma because ideally we would want this to be the **population** standard deviation. In practice we often only have the **sample** standard deviation, though. Usually that is ok.

Note The routine doesn't just find the power for the given diff but does a whole curve, for lots of differences!

D) If the true resting period is 125 minutes and we tested at the 10% level, what sample size would be needed for the test to have a power of 90%?

The same command will find the sample size if we give it the required power, and leave the sample size n out:

```
t.ps(power = 90, diff = 125-133, sigma = sd(RIP.sec.),
     alternative="less")
```

```
## Sample size required is  160
```

15.1.2 Case Study: Educational Achievements

A sociologist wants to study the educational level in some population. She randomly selects 120 people and asks them for their education, specifically the number of years (for example, someone who finishes high school is a 12 and someone with a bachelors degree a 16). She finds a mean of 12.3 with a standard deviation of 1.7. She wants to find a 90% confidence interval for the population mean.

Here we don't have the data itself, but the information given is all we need:

```
one.sample.t(y=12.3, shat=1.7, n=120, conf.level=90)
```

```
## A 90% confidence interval for the population mean is (12, 12.6)
```

Note One problem with this situation is that without the data we can not do a normal plot and verify that the data came from a normal distribution.

If she wanted the interval to have an error of 0.2, what sample size would she need?

```
t.ps(E=0.2, sigma=1.7, conf.level=90)
```

```
## [1] "Sample size required is  196"
```

15.2 Population Proportion (Percentage / Probability)

Note: proportion = probability = percentage/100

The R command for testing and to find confidence intervals for a population proportion π is *one.sample.prop*. This method has **no** assumptions!

15.2.1 Case Study: John Kerrichs Coin

The South African mathematician John Kerrich was in a German POW camp during World War II. During his time there he flipped a coin 10000 times, getting 5067 heads and 4933 tails.

A) Test at the 5% level of significance whether 5067 heads in 10000 flips are compatible with a fair coin.

```
one.sample.prop(x=5067, n=10000, pi.null=0.5)
```

```
## p value of test H0: pi=0.5 vs. Ha: pi <> 0.5:  0.1835
```

1) Parameter: proportion π

2) Method: exact binomial

3) Assumptions: None

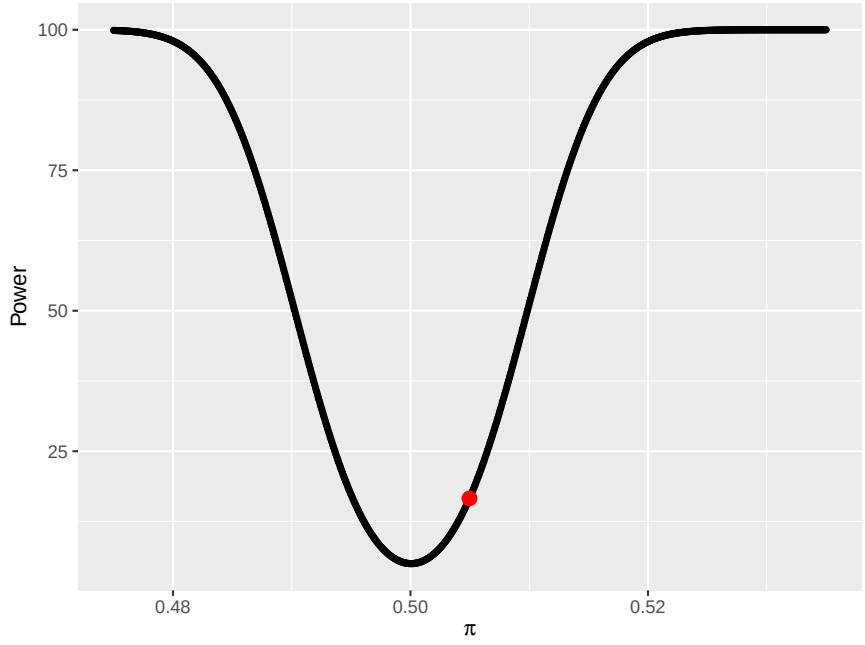
4) $\alpha = 0.05$
5) H_0: $\pi = 0.5$ (50% of flips result in "Heads", coin is fair)
6) H_a: $\pi \neq 0.5$ (coin is not fair)
7) p = 0.1835
8) p = 0.1835 > 0.05, so we fail to reject the null hypothesis, it appears John Kerrich's coin was indead fair.

Note problems for proportions do not include a standard deviation σ. (actually, they do, but it is "included" in the π)

B) If the true probability of heads for his coin was 0.505, what would have been the power of the test?

for power and sample size calculations for percentages/proportions/probabilities we have the prop.ps command:

```
prop.ps(n=10000, phat=0.505, pi.null=0.5)
```




```
## [1] "Power of Test = 16.6%"
```

Note that in the case of a proportion we need both phat and pi.null, not just the difference.

```
prop.ps(n=100, phat=0.5, pi.null=0.6)
```

```
## [1] "Power of Test = 53.7%"
```

```
prop.ps(n=100, phat=0.6, pi.null=0.5)
```

```
## [1] "Power of Test = 46.0%"
```

C) If the true probability of heads for his coin was 0.505, how often would he have had to flip the coin so that the chance for rejecting the null would have been 90%?

```
prop.ps(phat=0.505, pi.null=0.5, power=90)
```

```
## [1] "Sample size required is  105281"
```

15.2.2 Case Study: Failure Rates

The Professors of some course are worried about the failure rates (F's and W's) in some course. They are planning to randomly select students from the last few years and then find a 99% confidence interval for the failure rate. What sample size will they need if they want the interval to have an error of 7.5%?

```
prop.ps(E=0.075, conf.level=99)
```

```
## [1] "Sample size required is  295"
```

Note here we didn't mention phat. In that case prop.ps will use phat=0.5:

```
prop.ps(E=0.075, phat=0.5, conf.level=99)
```

```
## [1] "Sample size required is  295"
```

Say they know that the failure rate is around 40%, what would the sample size be then?

```
prop.ps(phat=0.4, E=0.075, conf.level=99)
```

```
## [1] "Sample size required is  284"
```

Note with phat=0.5 we get 295, with phat=0.4 it is n=284. In fact phat=0.5 will always yield a sample size larger than for any other phat. A larger sample size is always safer (yield a smaller confidence interval).

15.2.3 Case Study: Election polling

When you read the results os an election poll it often says that the poll has *a margin of error of* $\pm 3\%$. This means that the error of the confidence interval is 0.03.

In the next election for governor of Puerto Rico, what sample size will be needed in a poll to have an error of $\pm 3\%$?

```
prop.ps(E=0.03)
```

```
## [1] "Sample size required is  1068"
```

In the next election for president of the US, what sample size will be needed in a poll to have an error of $\pm 3\%$?

```
prop.ps(E=0.03)
```

```
## [1] "Sample size required is  1068"
```

Whu is this quite remarkable?

15.3 Pearson Chisquare Goodness-of-Fit Test

This test compares proportions from a sample with proportions from a population. The test has the assumption that all expected counts be at least five.

15.3.1 Case Study: Rolling a fair die

In order to test whether a certain die is fair it was rolled 100 times. The results were

x	p
12	1/6
17	1/6
20	1/6
15	1/6
10	1/6
26	1/6

Let's enter the data in R and run the test:

```
x  <- c(12, 17, 20, 15, 10, 26)
p <- c(1, 1, 1, 1, 1, 1)/6
chi.gof.test(x, p)
```

```
## p value of test p=0.0741
```

$p = 0.0741 > 0.05 = \alpha$, so we fail to reject the null, the die does not appear to loaded.

15.3.2 Case Study: Gregor Mendels Genetic Experiments

Experiments in Plant Hybridization (1865) by Gregor Mendel is one of the most famous papers in all of Science. His theory of genetics predicted that the number of Smooth yellow, Wrinkled yellow, Smooth green and Wrinkled green peas would be in the proportions 9/16, 3/16, 3/16 and 1/16. In one of his experiments he oberved 315, 101, 108 and 32. Does this agree with his theory?

```
x  <- c(315, 101, 108 , 32)
p  <- c(9, 3, 3, 1)/16
chi.gof.test(x, p)
```

```
## p value of test p=0.9254
```

$p = 0.9254 > 0.05$, the data from the experiment is in agreement with Mendel's theory.

15.3.3 Case Study: Eggs of Birds

a biologist has a theory regarding the number of eggs laid by a certain species of birds. He does a survey of these birds and finds:

Number of Eggs	Observed	Theory (in%)
0	15	25
1	24	30
2	8	20
3	11	20
More than 3	4	5

Test at the 5% level whether his data agrees with his theory

```
x <- c(15, 24, 8, 11, 4)
p <- c(25, 30, 20, 20, 5)/100
chi.gof.test(x, p)
```

```
## Warning : some expected counts<5
## p value of test p=0.4681
```

Notice the warning regarding the expected counts < 5. This is because the last category doesn't have enough data. The solution is to combine some categories:

Number of Eggs	Observed	Theory (in%)
0	15	25
1	24	30
2	8	20
More than 2	15	25

Now rerun that test

```
x <- c(15, 24, 8, 11+4)
p <- c(25, 30, 20, 20+5)/100
chi.gof.test(x, p)
```

```
## p value of test p=0.3674
```

15.3.4 Coin Flipping

you flip a coin 100 times and get 60 heads. Test at the 5% level whether this is a fair coin.

Solution A:

each flip is either "heads"" or "tails", so it is a Bernoulli trial. Therefore we can use

```
one.sample.prop(60, 100, pi.null = 0.5)
```

```
## p value of test H0: pi=0.5 vs. Ha: pi <> 0.5:  0.0574
```

$p > \alpha$, so we fail to reject the null hypothesis.

Solution B:

We can use the chisquare goodness-of-fit test:

```
x <- c(40, 60)
p <- c(0.5, 0.5)
chi.gof.test(x, p)
```

```
## p value of test p=0.0455
```

$p < \alpha$, so we reject the null hypothesis.

What is going on? To begin with, both methods are perfectly ok. They are not the same methods, so we should not expect them to give the (exactly) same answer. Also, in either case the p values are close to 0.05, in which case one should not really either reject or fail to reject the null hypothesis but if possible collect more data.

16 Bayesian Statistics

Say you pick a coin from your pocket. It's just any coin, nothing special. You flip it 10 times and get 3 heads . What can we conclude about this coin?

Now each flip is a Bernoulli trial with success parameter π. We have previously seen that the standard estimator for π is the ratio of successes to trials, so we find $\widehat{\pi} = x/n = 3/10 = 0.3$.

But wait just a minute! This is a regular coin, we all know that coins are (almost) fair, so we know that really $\pi = 0.5$! 3 head in 10 flips of a fair coin is a perfectly fine outcome, in fact the probability of 3 or less heads in 10 flips of a fair coin is 0.172, so this will happen easily.

What's going on? The problem is that the formula $\widehat{\pi} = x/n$ is completely general, it is the same whether we flip a coin (head vs tails), survey people (male vs female), check students in a class (pass vs fail) or do anything else that is a Bernoulli trial. It does not take into account that we know a lot about this experiment "flip a coin" **a priori**, that is before we ever do it, namely that (almost always) $\pi = 0.5$.

Of course there is also the issue that 10 flips is very few, just 300 heads in 1000 flips would be a very different thing. But situations with little data are quite common, and it would still be nice to have a more sensible answer than 0.3.

In fact, it is possible to include such a priori information in a statistical analysis, applying what is called **Bayesian Statistics**. The principle idea is this:

- "encode" your knowledge of the experiment before it is done in what is called a **prior distribution**.
- do the experiment and collect the data
- combine the data and the prior to get to the **posterior distribution**, which now encodes our updated knowledge of what we know after having done the experiment.

Note: the prior and the posterior are regular probability distributions like those we discussed before.

The science of Statistics comes in two flavours: Bayesian and Frequentist. There are a number of fundamental differences between them. One we have already seen: a Bayesian analysis not only can but has to begin by specifying a prior distribution . This can be a strength (as in the coin flip example above) or a weakness, mostly in situations were we really don't have much prior knowledge. A Frequentist analysis on the other hand doesn't need a prior, but also can't use one if there is one!

There are deeper differences as well, for example the very definition of what a probability is. Those issues are quite fundamental to doing Statistics but unfortunately much beyond what we can discuss in an introductory class!

So, how do we do this "combine data and prior" step? It uses something called **Bayes formula** (which is where the name comes from) and a lot of heavy math, calculus and more. This is one reason why Bayesian statistics is not yet as widely used as most Statisticians think it should be! But more and more computer programs can take care of the calculations for us.

I have written an "Interactive Bayesian Calculator for Percentages", which we can use for our problem. Run it with

```
ibayesprop()
```

when it opens it looks like this:

The first thing we need to do is specify the prior distribution.

There are several ways to do this, listed on the left side. The default option is to specify what we think the most likely value is and what the range might be. We do think this is a fair coin, so 50% is ok. The graph shows that any value between about 25% and 75% is ok. You can use the box above the graph to change that if you want.

Below the graph we see the interval (30.6%, 69.3%). If our prior distribution is reasonable for our problem than the true percentage should be inside.

Now let's enter our data, Sample Size = 10 and Number of Successes = 3:

The blue curve is the same prior distribution as before, the red is the posterior distribution, that is our best guess after having seen the result of the experiment. Notice that because there were fewer heads than we expected from a fair coin it has shifted a bit to the left.

On the bottom we see the 95% Bayesian credible interval (27.4%, 60.2%), which is our best guess after having done the experiment.

For comparison we also have the 95% confidence interval (8.1%, 64.6%).

Let's see what would happen if we actually had 300 heads in 1000 flips:

Again the red posterior curve has shifted to the left, by now it is far away from the blue prior one. The Bayesian interval is (27.2%, 33.2%).

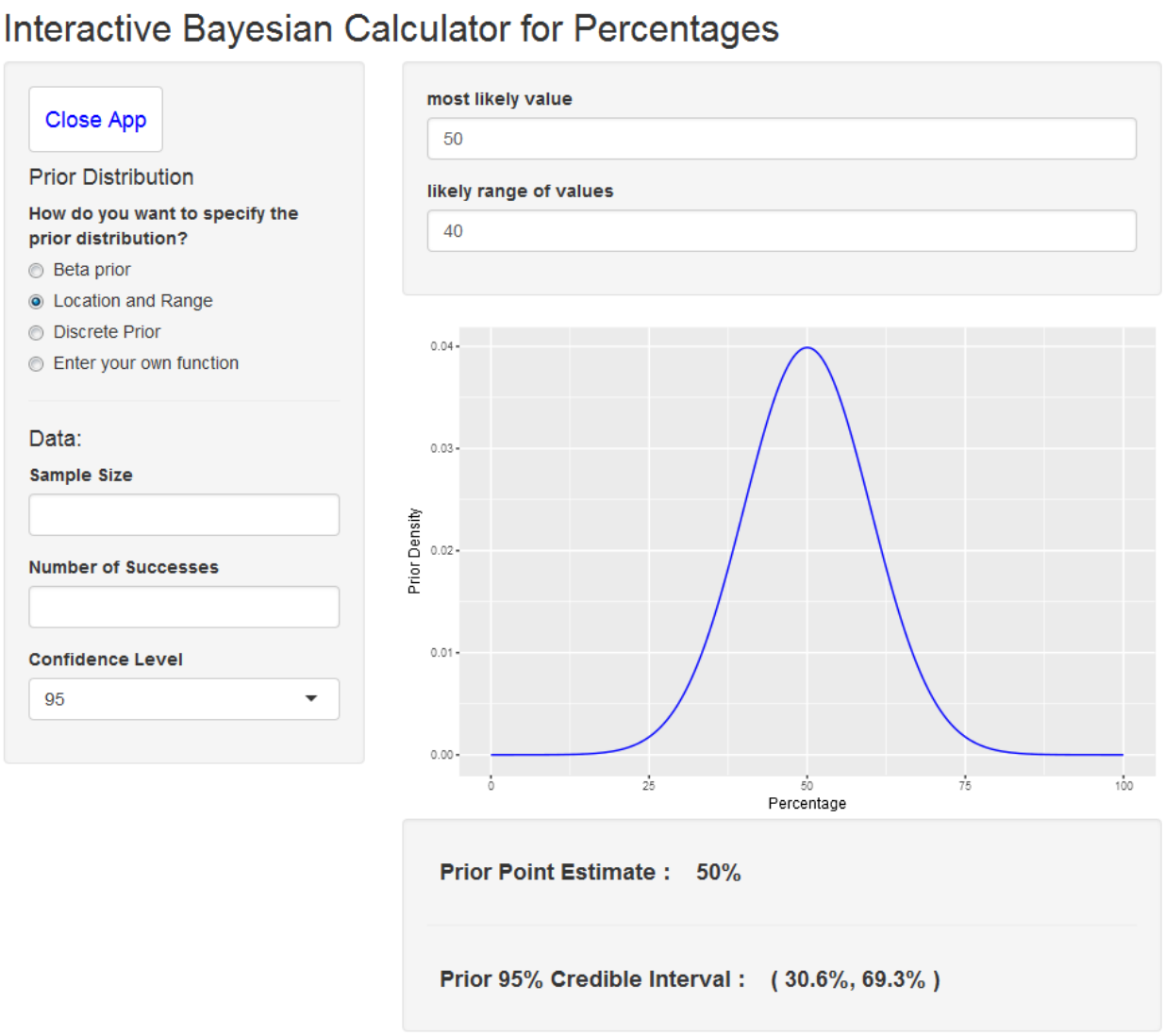

Interactive Bayesian Calculator for Percentages
Close App
Prior Distribution
How do you want to specify the prior distribution?
Beta prior
Location and Range
Discrete Prior
Enter your own function
Data:
Sample Size
Number of Successes
Confidence Level
95
most likely value
50
likely range of values
40
Prior Density
0.04
0.03
0.02
0.01
0.00
0
25
50
75
100
Percentage
Prior Point Estimate : 50%
Prior 95% Credible Interval : (30.6%, 69.3%)

Figure 6:

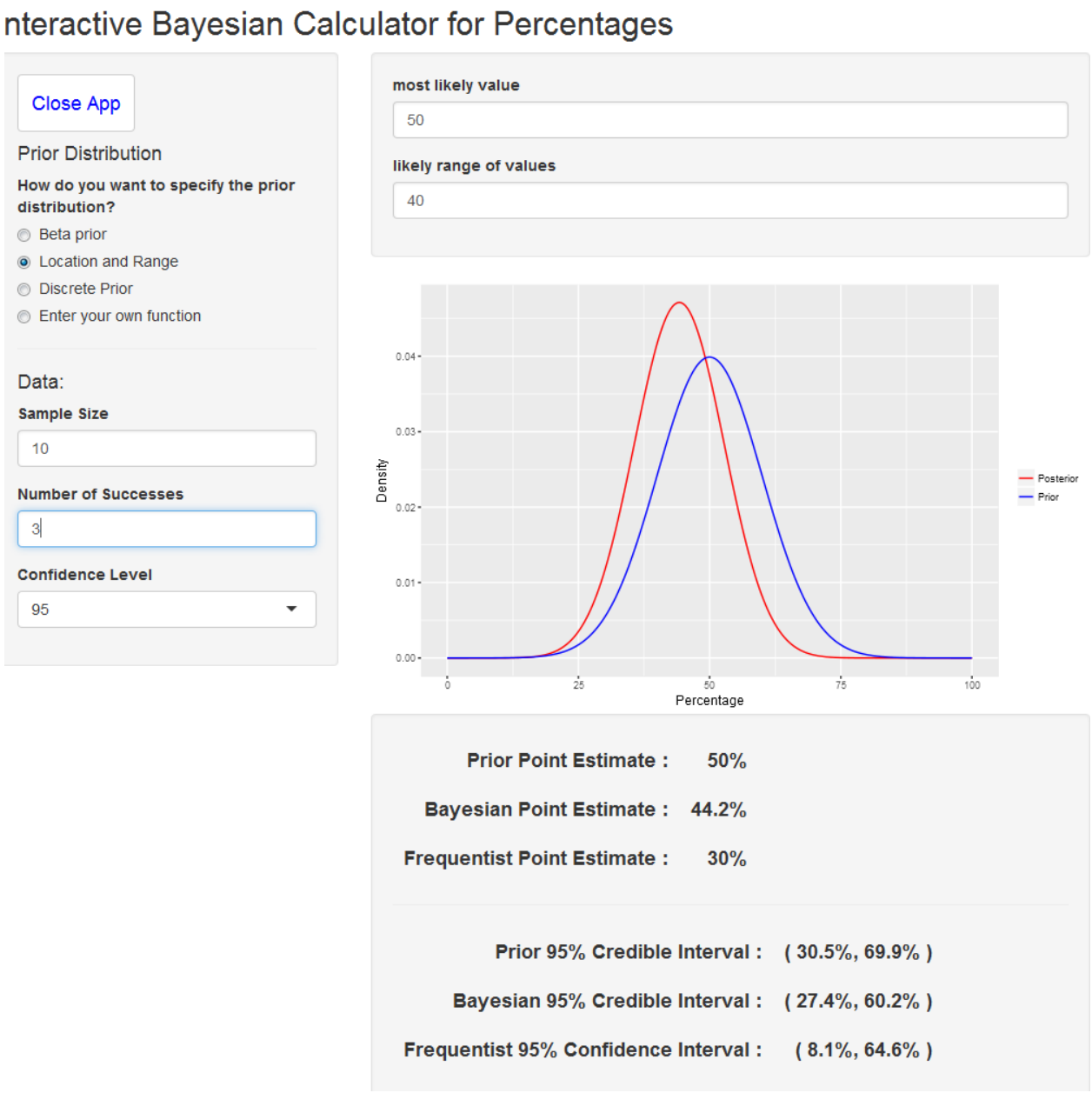

nteractive Bayesian Calculator for Percentages
Close App
Prior Distribution
How do you want to specify the prior distribution?
Beta prior
Location and Range
Discrete Prior
Enter your own function
Data:
Sample Size
10
Number of Successes
3
Confidence Level
95
most likely value
50
likely range of values
40
Density
Percentage
Posterior
Prior
Prior Point Estimate : 50%
Bayesian Point Estimate : 44.2%
Frequentist Point Estimate : 30%
Prior 95% Credible Interval : (30.5%, 69.9%)
Bayesian 95% Credible Interval : (27.4%, 60.2%)
Frequentist 95% Confidence Interval : (8.1%, 64.6%)

Figure 7:

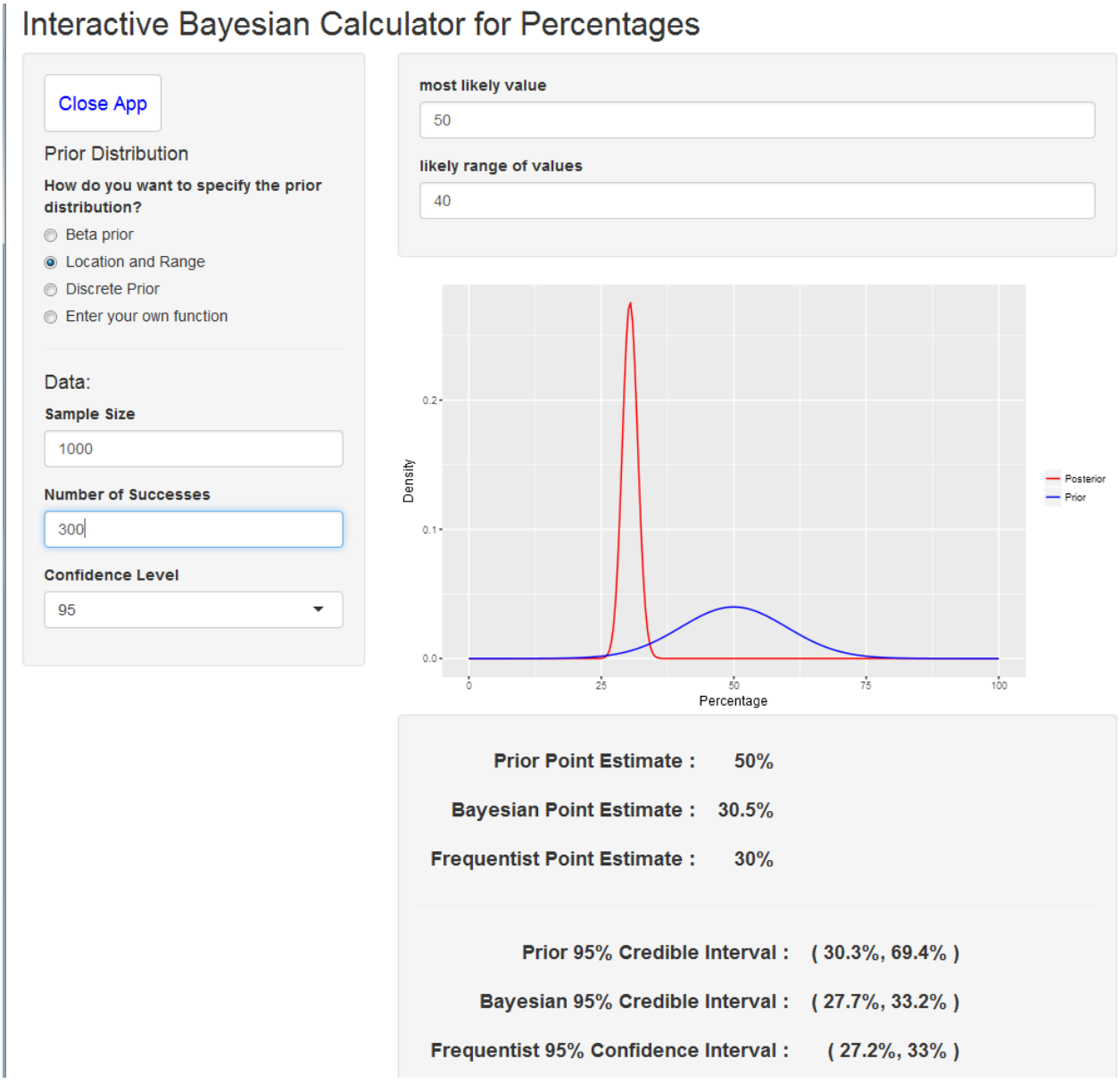

Interactive Bayesian Calculator for Percentages
Close App
Prior Distribution
How do you want to specify the prior distribution?
Beta prior
Location and Range
Discrete Prior
Enter your own function
Data:
Sample Size
1000
Number of Successes
300
Confidence Level
95
most likely value
50
likely range of values
40
Density
0.2
0.1
0.0
0
25
50
75
100
Percentage
Posterior
Prior
Prior Point Estimate : 50%
Bayesian Point Estimate : 30.5%
Frequentist Point Estimate : 30%
Prior 95% Credible Interval : (30.3%, 69.4%)
Bayesian 95% Credible Interval : (27.7%, 33.2%)
Frequentist 95% Confidence Interval : (27.2%, 33%)

Figure 8:

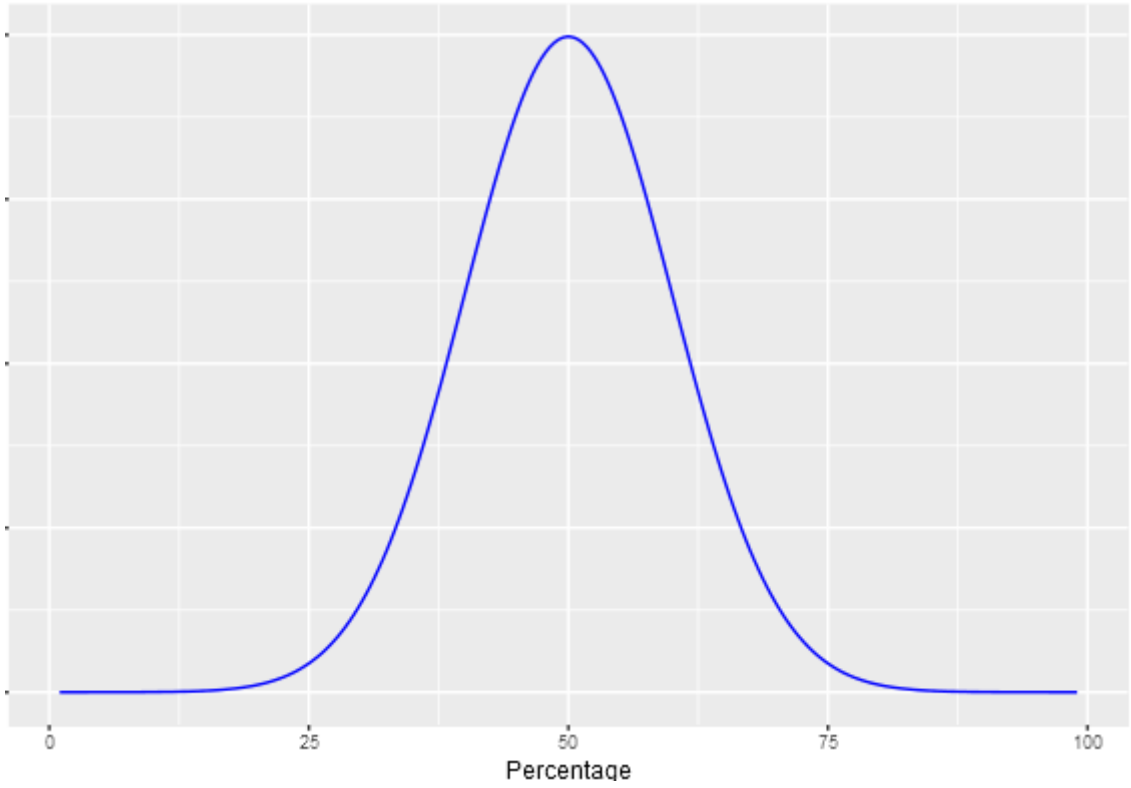


Figure 9:

Notice that it is quite similar to the Frequentist confidence interval (27.2%, 33.0%). This is something we see a lot: in cases were there is a lot of data (100 flips) the answers from a Bayesian and a Frequentist analysis tend to be very similar. This of course makes good sense: whatever our expectation was before the experiment (as encoded in the prior distribution), we will certainly change that expectation in the face of a lot of evidence (aka data).

16.0.0.1 Specifying a Prior Distribution

There is a vast literature on how to go about encoding our prior knowledge. In the app I have included four ways to do so:

- Location and Range: just as it says, decide what the most likely value is and in what range the answer should be.

Here are four examples:

1) we think the true percentage is around 50%, but it could be as low as 25% and as high as 75%
2) we think the true percentage is around 50%, but it could be as low as 0% and as high as 100%
3) we think the true percentage is small, maybe even 0, and no larger than 20%
4) we are quite certain that the true percentage is around 20%.

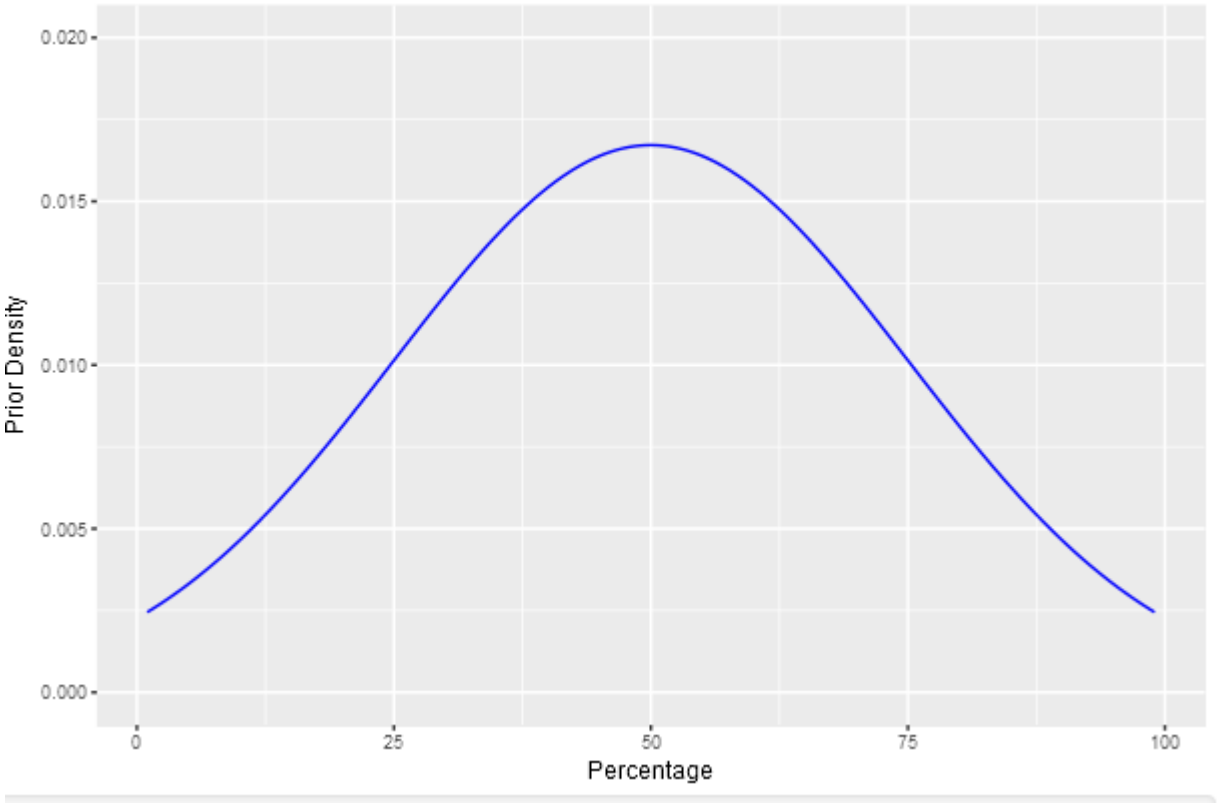

0.020
0.015
0.010
0.005
0.000
Prior Density
0
25
50
75
100
Percentage

Figure 10:

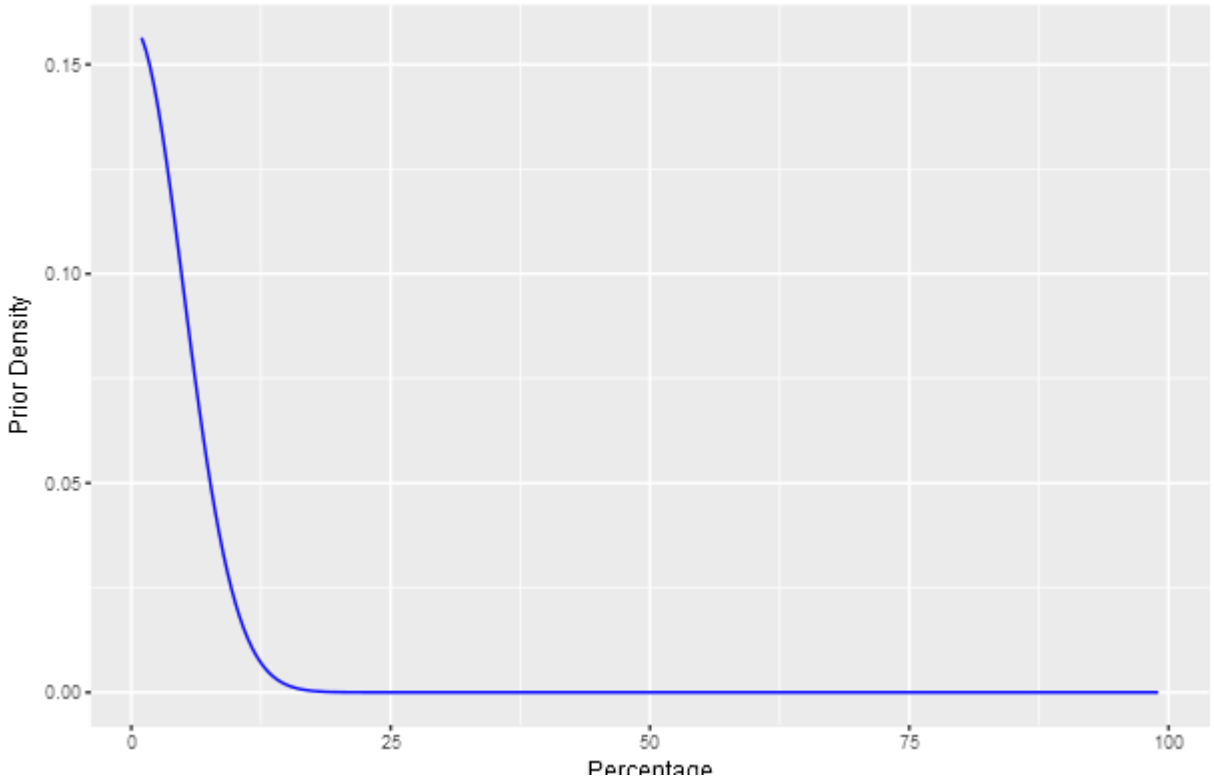


Figure 11:

- Beta prior: this is a class of distributions which has a number of advantages. It has two parameters α and β, and you can use sliders to get a shape that works for your experiment.

1) we really have no idea where π might be.

This one is the default for the Beta. It looks a little funny but has some good theoretical features (for the specialists: it is the Jeffrey's prior for the binomial)

2) we really have no idea where π might be.

Another favorite, what is called a flat prior.

3) we think the true percentage is around 50% but we are not to sure of that.

Note that here we have $\alpha = \beta$, which will always put the peak of the curve at 50%

4) we are quite certain that the true percentage is greater than 50%.

- Discrete prior: here we can specify the (relative) probabilities for 10 points in some interval.

1) we really have no idea where π might be.

2) we really have no idea where π might be, but is not likely that it is either very close to 0 or very close to 100%

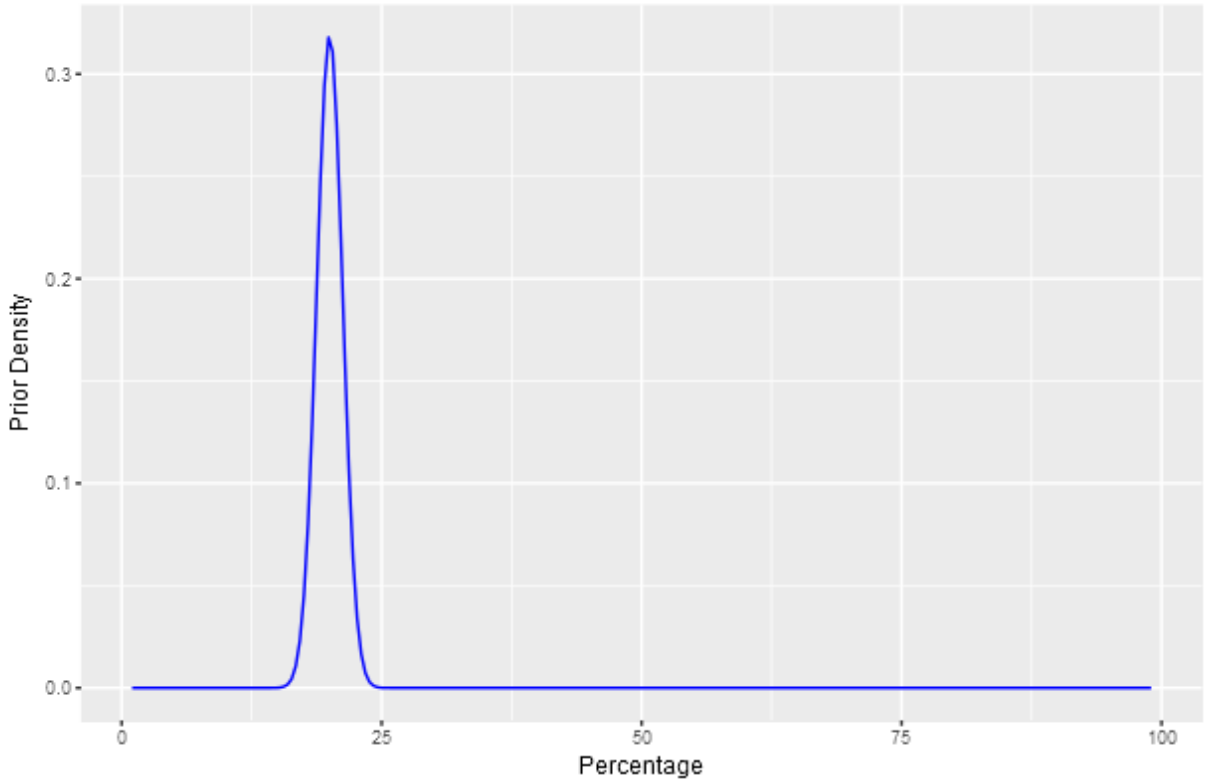


Figure 12:

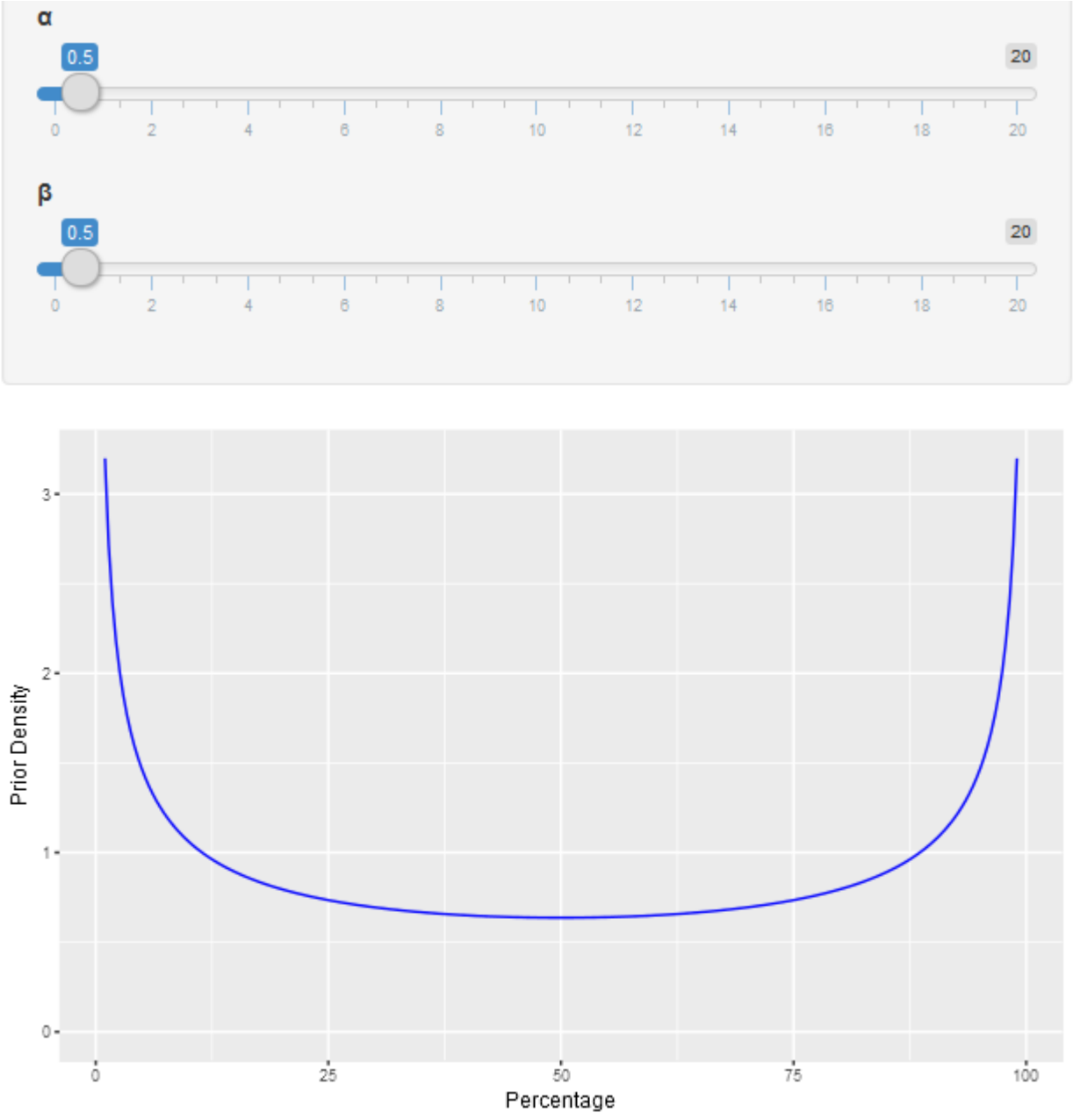

α
0.5
20
0
2
4
6
8
10
12
14
16
18
20
β
0.5
20
0
2
4
6
8
10
12
14
16
18
20
Prior Density
0
1
2
3
0
25
50
75
100
Percentage

Figure 13:

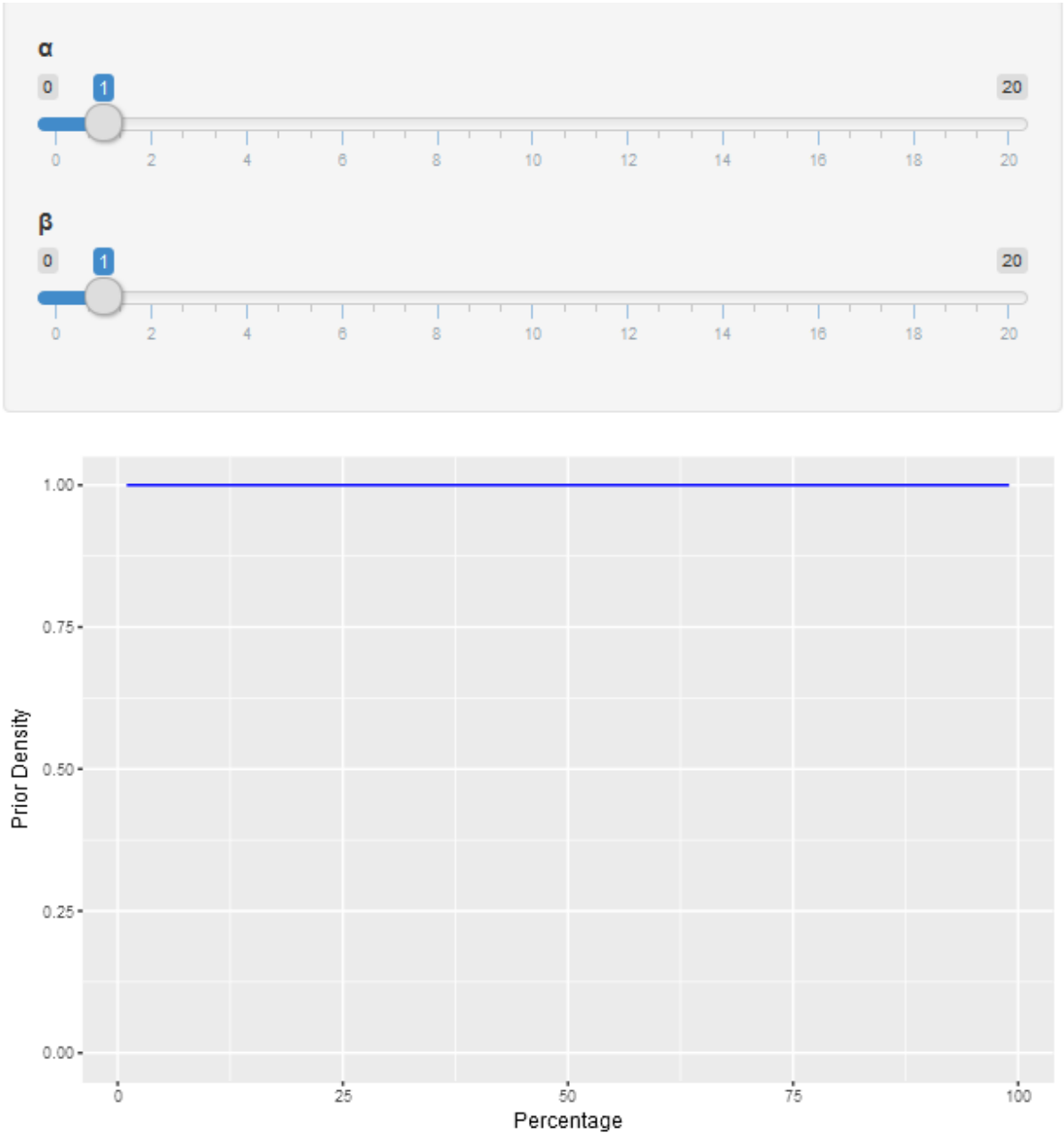

α
0
1
20
0
2
4
6
8
10
12
14
16
18
20
β
0
1
20
0
2
4
6
8
10
12
14
16
18
20
1.00
0.75
0.50
0.25
0.00
Prior Density
0
25
50
75
100
Percentage

Figure 14:

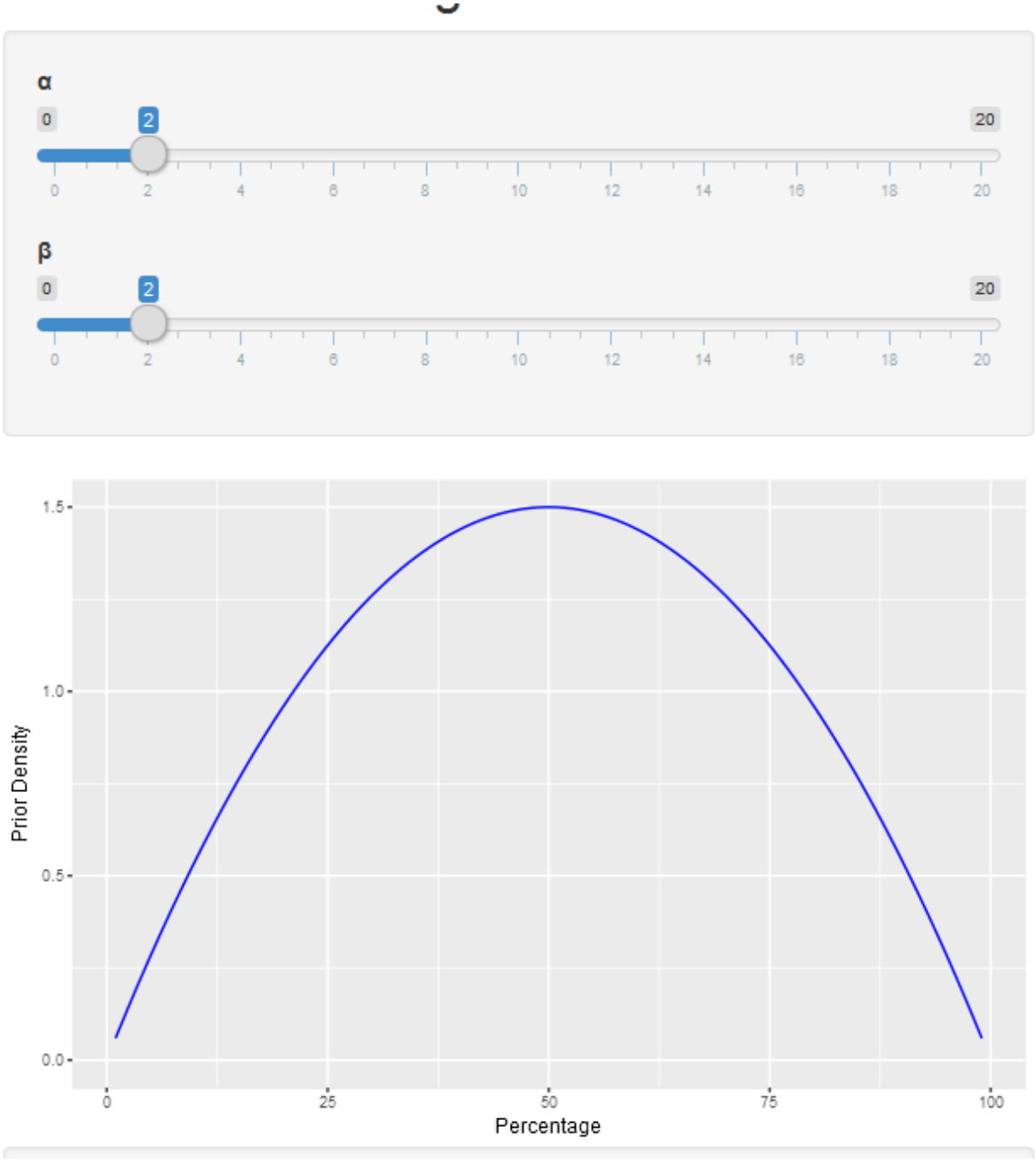

α
0
2
20
0 2 4 6 8 10 12 14 16 18 20
β
0
2
20
0 2 4 6 8 10 12 14 16 18 20
1.5
1.0
0.5
0.0
Prior Density
0
25
50
75
100
Percentage

Figure 15:

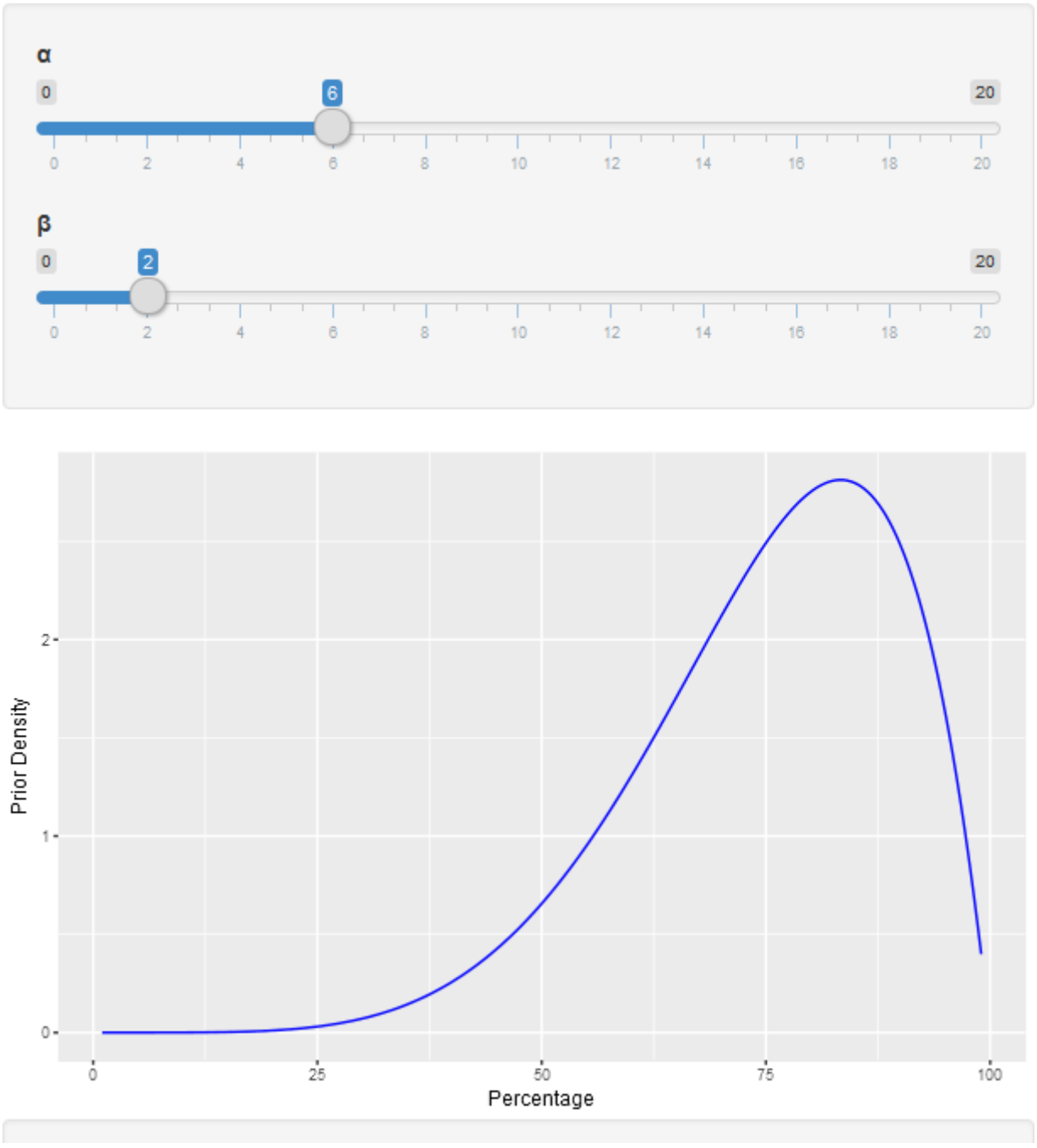

α
0
6
20
0 2 4 6 8 10 12 14 16 18 20
β
0
2
20
0 2 4 6 8 10 12 14 16 18 20
Prior Density
2
1
0
0
25
50
75
100
Percentage

Figure 16:

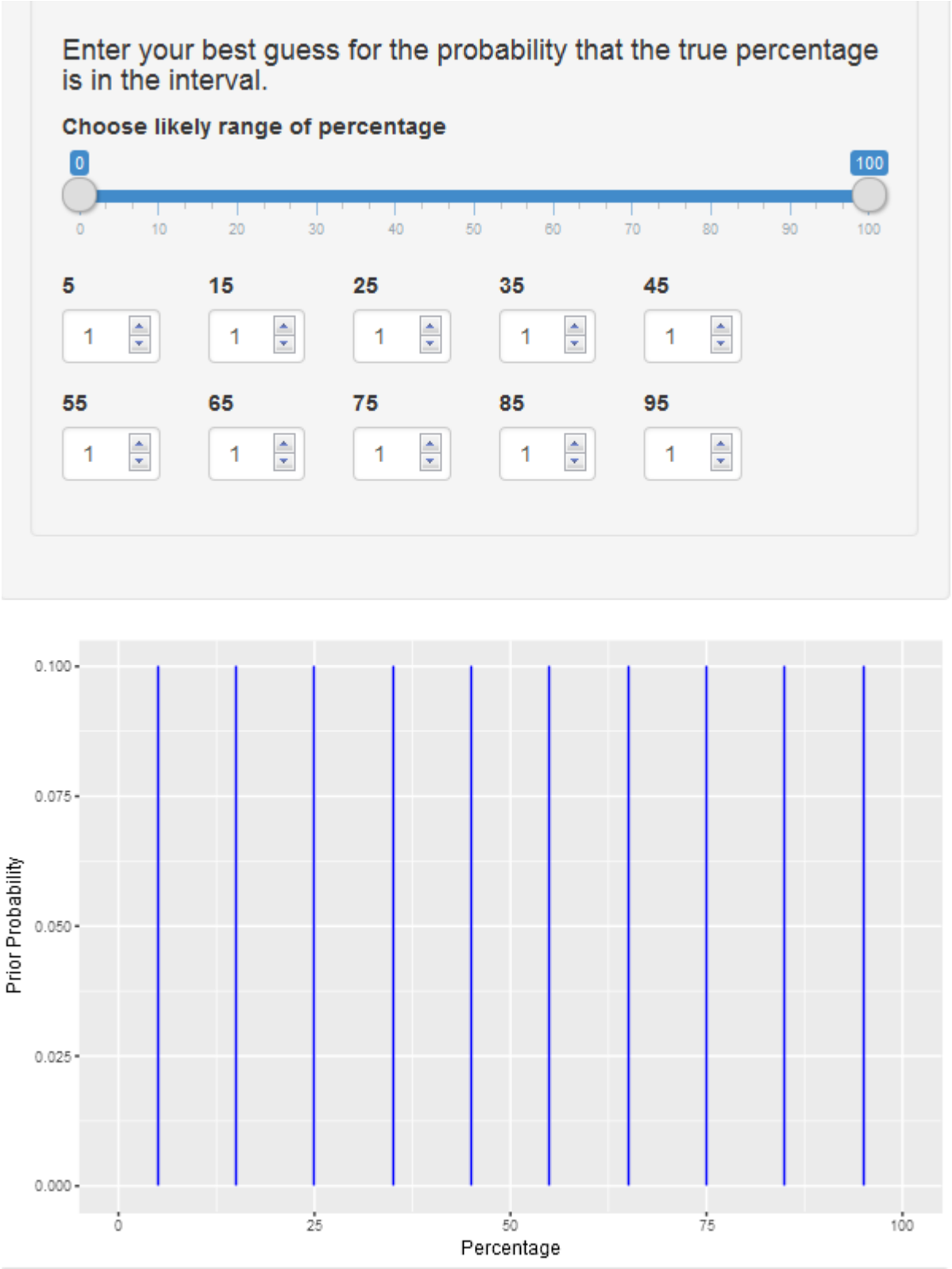
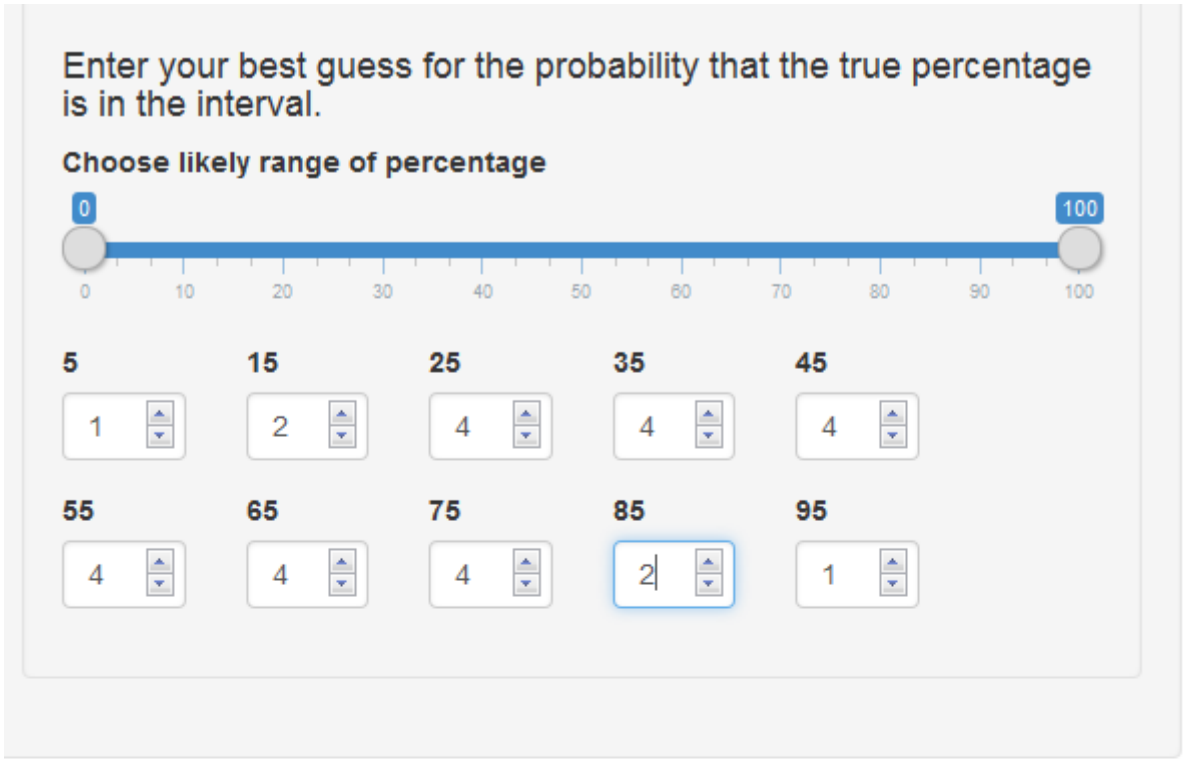

Enter your best guess for the probability that the true percentage is in the interval.
Choose likely range of percentage
0
100
0
10
20
30
40
50
60
70
80
90
100
5
15
25
35
45
1
1
1
1
1
55
65
75
85
95
1
1
1
1
1
0.100
0.075
0.050
0.025
0.000
Prior Probability
0
25
50
75
100
Percentage

Figure 17:


Enter your best guess for the probability that the true percentage is in the interval.
Choose likely range of percentage
0
100
0
10
20
30
40
50
60
70
80
90
100
5
15
25
35
45
1
2
4
4
4
55
65
75
85
95
4
4
4
2
1

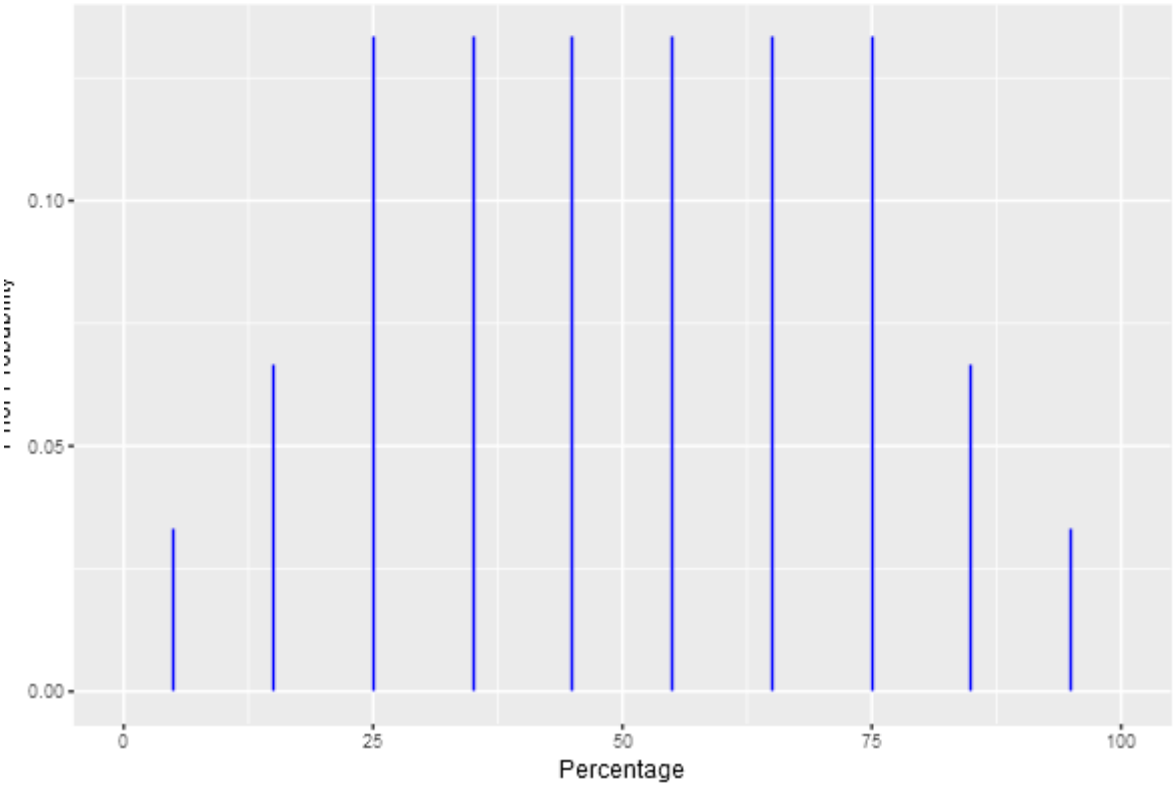
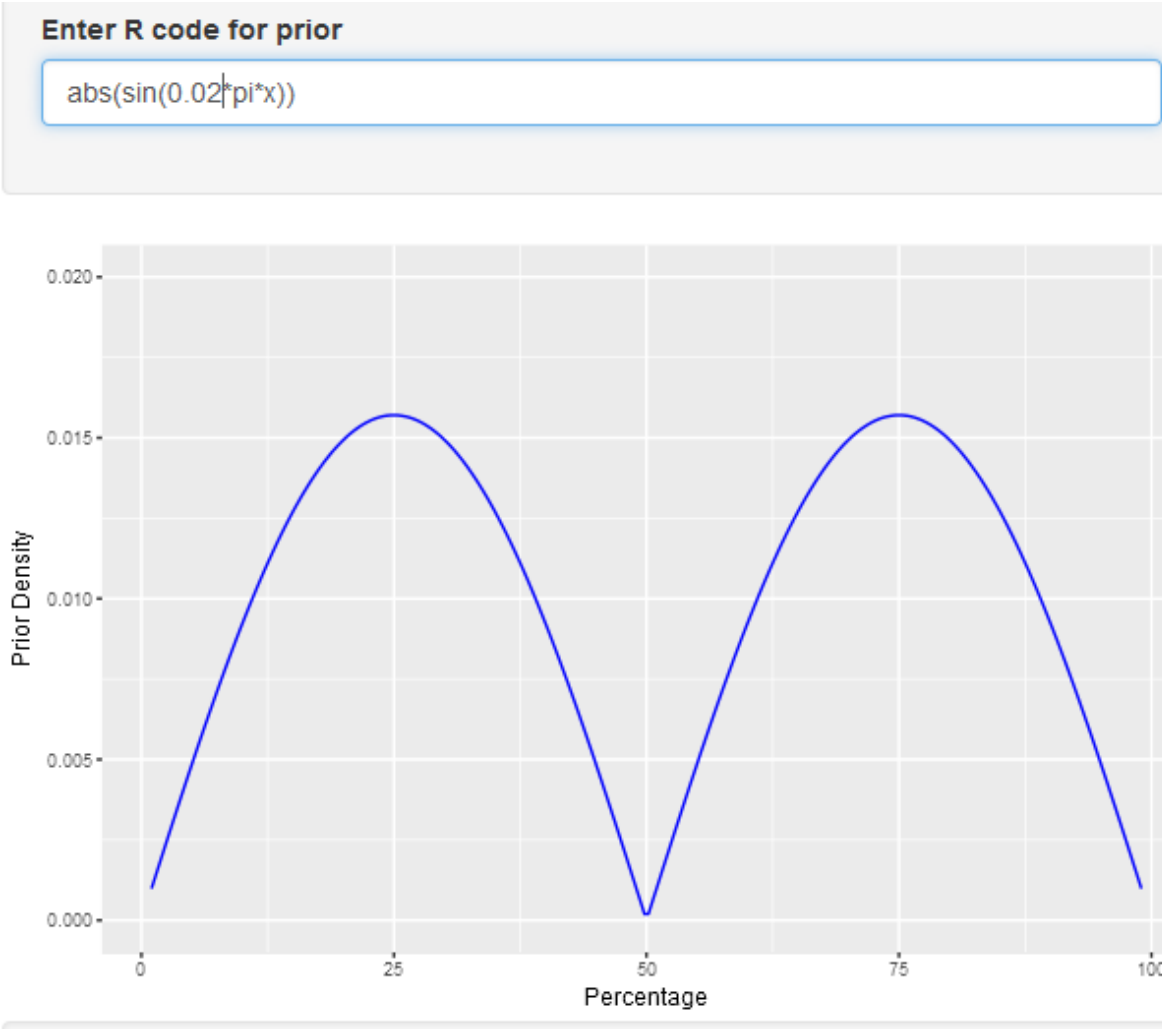

0.10
0.05
0.00
0
25
50
75
100
Percentage

Figure 18:

3) we think the true percentage is around 40%. Moreover, we are vey sure it is not less than 30% and not higher than 50%.

- Enter your own function: here you can enter any R expression for any function you like! (and that could e a prior, of course). we think the true percentage is is either around 25%



or around 75%.

Can you think of any situation where this might actually be an apprioriate prior?

16.0.0.2 Example

We have collected data from some recent classes. For each student in each class we we found their gender. What would be an appropriate prior to use here?

Actually it will depend on the class. For example, if this is a class in engineering, the percentage of females is likely smaller than 50% but if it is a course in nursing it likely larger than 50%. If we don't know what class it is we should use a prior which allows for some range. So maybe Location and Range with most likely value = 50 and likely range of values = 60

16.0.0.3 Example

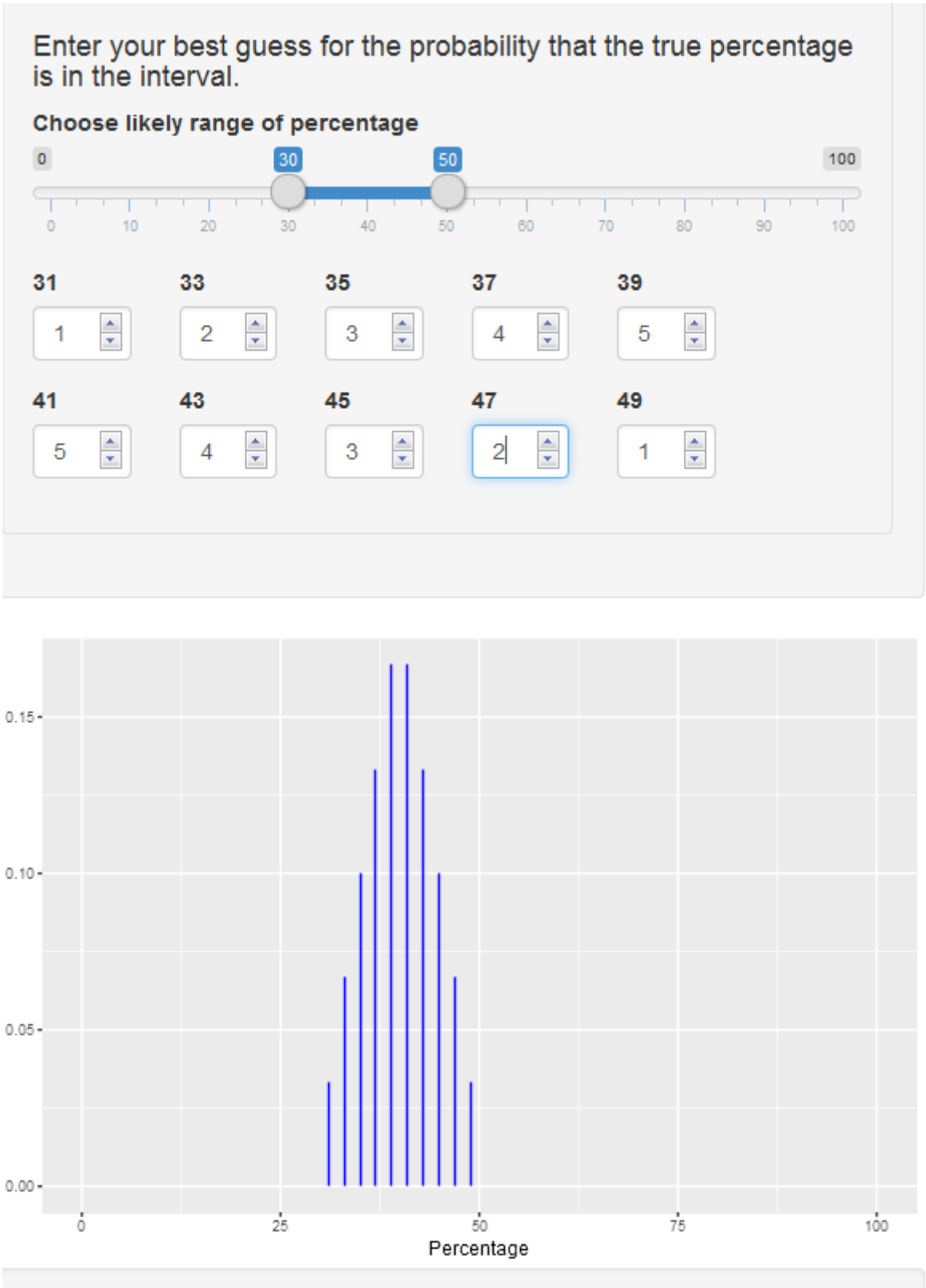

Enter your best guess for the probability that the true percentage is in the interval.
Choose likely range of percentage
0
30
50
100
0 10 20 30 40 50 60 70 80 90 100
31
1
33
2
35
3
37
4
39
5
41
5
43
4
45
3
47
2
49
1
0.15
0.10
0.05
0.00
0
25
50
75
100
Percentage

Figure 19:

We have collected data from some recent introductory statistics classes. For each student in each class we we know whether they got an A or not. What would be an appropriate prior to use here?

Here a prior with a peak at 10% seems appropriate. Moreover, any number above 20% is highly unlikely.

Try Beta prior with $\alpha = 2$ and $\beta = 20$.

16.0.0.4 Example

We have collected data from some experiment. We know the following:

- the percentage is definitely between 70% and 90%
- the percentage is most likely between 78% and 82%
- the percentage is twice as likely to be less than 78% than it is to be over 82%.

Here is one way to encode this prior knowledge with the Discrete prior option:

16.0.0.5 Example

So, how about our coin? What should we do here?

There are really two possibilities: either the coin is fair, so π is just about 0.5, and that is most likely the case. Or it is not fair, and then π could really be anything at all.

Here is one way to encode this:

The prior (blue) curve is flat from 0 to 100 but moves up sharply between 48 and 52. (This is often called Lincoln's hat function!) Under the posterior (red) curve this is still most likely a fair coin, but there is little higher chance that it has a bias towards tails.

17 Exercise Problems 1

In this page we will practise some of the things we have learned before. You should try to solve each problem on your own before looking at the solutions.

17.0.1 Problem 1

Some years ago an insurance company did a study of their policies and found that 30% of them where for cars. They randomly select 532 of their current policies and find that 131 of them are car insurances.

a. Test at the 1% level whether the percentage of car insurance policies is now smaller than 30%

b. If the true percentage of car insurance policies is 25%, what was the power of this test?

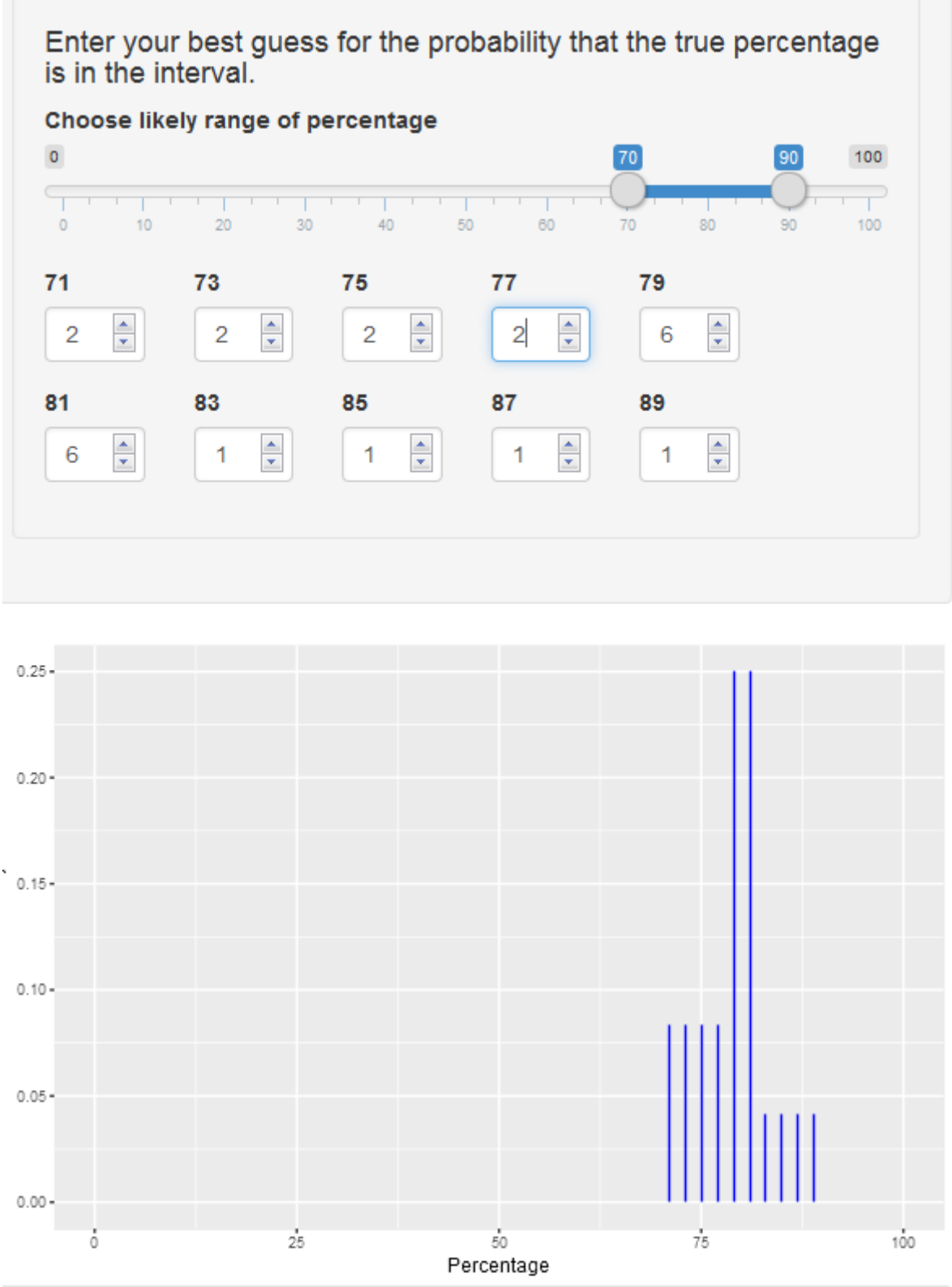

Enter your best guess for the probability that the true percentage is in the interval.
Choose likely range of percentage
0
70
90
100
0 10 20 30 40 50 60 70 80 90 100
71
73
75
77
79
2
2
2
2
6
81
83
85
87
89
6
1
1
1
1
0.25
0.20
0.15
0.10
0.05
0.00
0
25
50
75
100
Percentage

Figure 20:

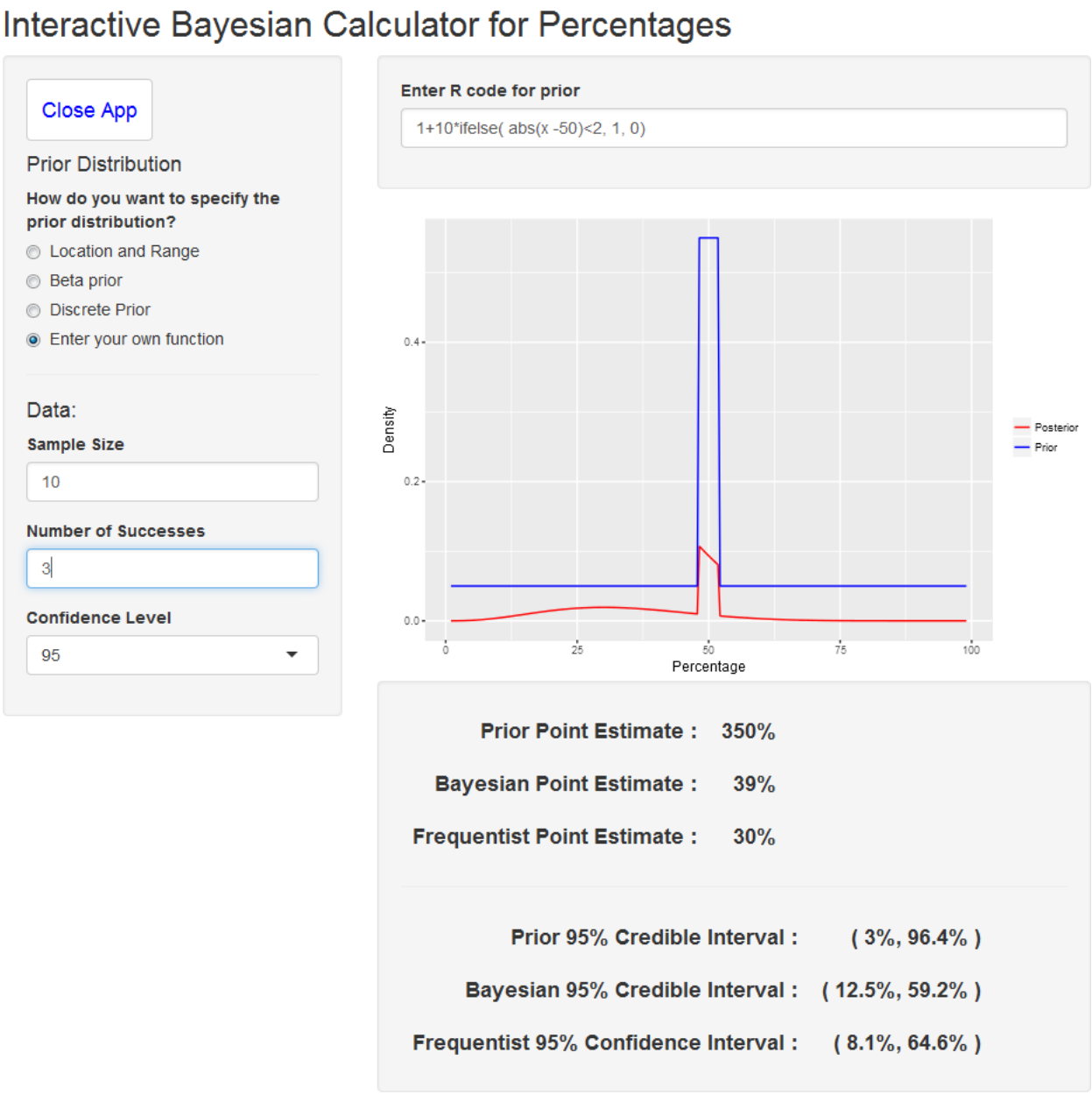

Interactive Bayesian Calculator for Percentages
Close App
Prior Distribution
How do you want to specify the prior distribution?
Location and Range
Beta prior
Discrete Prior
Enter your own function
Data:
Sample Size
10
Number of Successes
3
Confidence Level
95
Enter R code for prior
1+10*ifelse(abs(x -50)<2, 1, 0)
Density
0.4
0.2
0.0
0
25
50
75
100
Percentage
Posterior
Prior
Prior Point Estimate : 350%
Bayesian Point Estimate : 39%
Frequentist Point Estimate : 30%
Prior 95% Credible Interval : (3%, 96.4%)
Bayesian 95% Credible Interval : (12.5%, 59.2%)
Frequentist 95% Confidence Interval : (8.1%, 64.6%)

Figure 21:

c. If the true percentage of car insurance policies is 25%, what sample size is needed to have a power of 95%

17.0.2 Problem 2

An insurance company is interested in the amount of money they pay on average on insurance claims. They randomly select 42 policies and find the payouts:

1400 32400 27400 22400 3500 8300 33600 17000 9600 20500 4900 9900 33100 9600 23200 22400 3600 12400 14900 29100 4500 13500 12100 12700 16600 17000 21700 29200 16200 4000 16500 17000 41000 19000 23900 37300 32100 5200 19700 21600 400 18700

Find a 95% confidence interval for the true mean amount of payouts.

17.0.3 Problem 3

In a certain store the average sales is $48. The store ran an add in a newspaper, and they want to see whether the add has worked. They randomly select some recent sales and find

52.35 48.34 72.55 71.55 44.68 56.64 52.43 73.98 60.55 60.29
42.33 46.84 59.45 63.04 59.04 33.28 50.67 62.3 54.13 68.65
34.21 58.72 60.99 68.24 52.45 56.88 57.81 56.29 79.69 46.88
48.73 53.77 65.79 73.71 44.55 58.53 51.48 52.95 46.26 50.98

a. Test at the 5% level whether the add was a success

b. if the add raised the mean sales to $50, what is the power of the test?

c. if the add raised the mean sales to $50, what sample size is needed to have a power of 99%?

17.0.4 Problem 4

According to a web site of the Red Cross 57% of Hispanics have blood type O, 31% have type A, 10% type B and 2% type AB. A sample of 250 people resulted in the following blood types:

O	A	B	AB
147	71	28	4

Test at the 10% level whether the blood types suggest that these people were hispanics

17.0.5 Problem 5

The data set studentsurvey has the replies of students to some questionnaire.

a. find a 90% confidence interval for the mean score

b. test at the 10% level whether there are equally many male and female students.

c. test at the 5% level whether the mean GPA is less than 2.5

d. test at the 5% level whether the poplation has equally many Freshman, Junior, Senior and Sophomore

e. find a 90% confidence interval for the mean age of the students.

17.0.6 Problem 6

In class we talked about Bernoulli trials, that is experiments which have only two possible outcomes. Often one is interested in how often a certain outcome happens when the experiment is carried out a number of times. This is then called a Binomial distribution, and probabilities can be found with the R command dbinom(k, n, p), where n is the number of trials, k how often the outcome happens and p its probability. For example, if we want to know the probability of 2 sixes in 10 rolls of a fair die, it is

```
dbinom(2,10,1/6)
```

```
## [1] 0.29071
```

a. if a group consists of 100 men and 90 women and if three people are chosen at random, what is the probability all of them are men?

b. if a fair coin is flipped 10 time, what is the probability of at most 3 heads?

c. if the probability of having an accident on any one mile of road is 0.001, what is the probability of having at least one accident when driving 10000 miles in one year?

d. if a fair coin is flipped 100 times, what is the probability of getting between 40 and 60 heads (include 40 and 60) ? (Compare that our discussion of the coin app)

17.0.7 Problem 7

The mean score in the final exam of a Calculus course over many years was 72.3. The University wants to decide whether or not to change the text book, and so they are planning to teach several sections of the course with a new text book. Then they will test H_0: $\mu = 72.3$ vs H_a: $\mu > 72.3$ at the 5% level. If it is true that with this new text book the mean score will go up to 75.5 points, how many students do they need to have so that the hypothesis test has a power of 80%? (assume the standard deviation is 15.0)

17.0.8 Problem 8

At a certain moment in time the national unemployment rate was 6.9%. In one city among 250 randomly selected people 30 said they were unemployed. Test at the 10% level to check whether in this city the unemplyoment rate differs from that nationwide.

17.1 Solutions

17.1.1 Problem 1

Some years ago an insurance company did a study of their policies and found that 30% of them where for cars. They randomly select 532 of their current policies and find that 131 of them are car insurances.

Variables: 1 Proportion

Problem: Hypothesis test

a. Test at the 1% level whether the percentage of car insurance policies is now smaller than 30%

```
one.sample.prop( x = 131, n =  533, pi.null = 0.3, alternative="less")
```

```
## p value of test H0: pi=0.3 vs. Ha: pi < 0.3:  0.0036
```

1) Parameter: proportion π

2) Method: exact binomial

3) Assumptions: none
Assumption is ok

4) $\alpha = 0.01$

5) H_0: $\pi = 0.3$

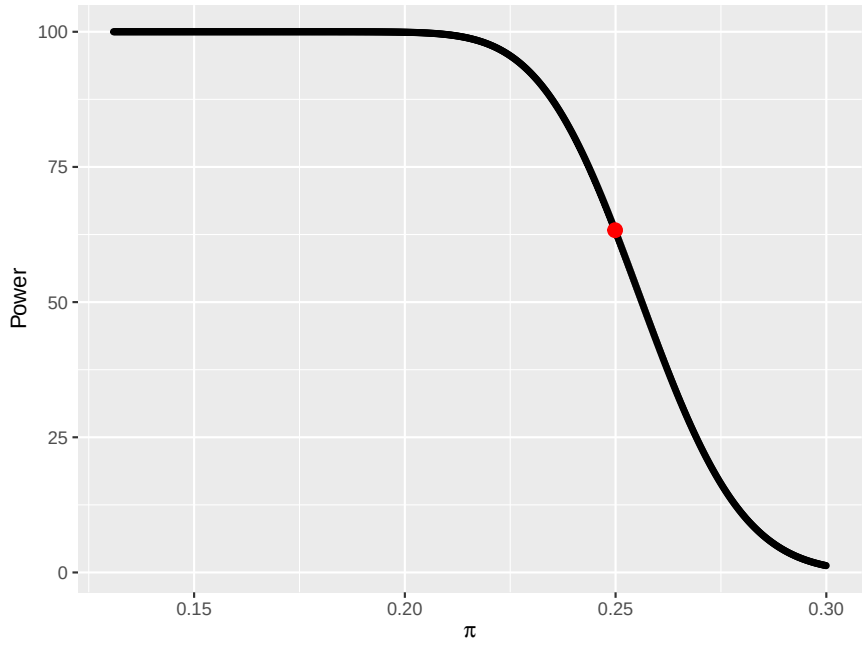
6) H_a: $\pi < 0.3$

7) p-value = 0.0036

8) p-value = 0.0036 $< \alpha$, so we reject the null hypothesis, the true percentage of car insurance claims is statistically significantly smaller than 30%

b. If the true percentage of car insurance policies is 25%, what was the power of this test?

```
prop.ps( n = 533, phat = 0.25, pi.null = 0.3, alpha = 0.01, alternative = "less")
```




```
## [1] "Power of Test = 63.3%"
```

c. If the true percentage of car insurance policies is 25%, what sample size is needed to have a power of 95%

```
prop.ps( power = 95, phat = 0.25, pi.null = 0.3, alpha = 0.01, alternative = "less")
```

```
## [1] "Sample size required is  1233"
```

17.1.2 Problem 2

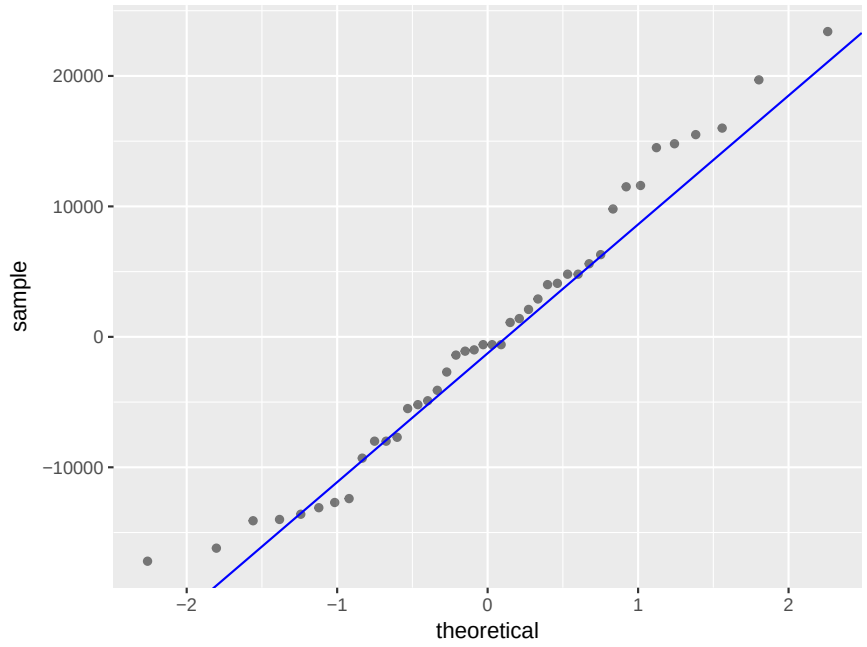
Find a 95% confidence interval for the true mean amount of payouts.

Variables: 1 mean
Problem: confidence interval

highlight the data, then in R

```
one.sample.t(x)
```

Assumptions are ok (checked boxplot and normal plot)




```
## A 95% confidence interval for the population mean is (14369.7, 20825.5)
```

17.1.3 Problem 3

Variables: 1 mean
Problem: hypothesis test

a. Test at the 5% level whether the add was a success highlight the data, then in R

Assumptions are ok (checked boxplot and normal plot)

1) Parameter of interest: population mean

2) Method of analysis: one sample t

3) Assumptions of Method: normal data or large sample

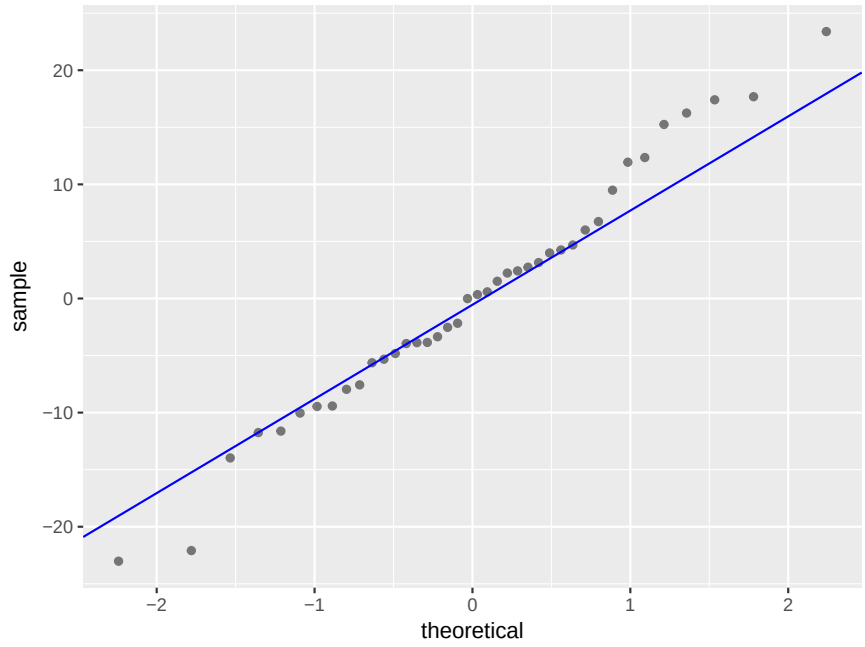
4) Type I error probability $\alpha = 0.05$

5) H_0: $\mu = 48$

6) H_a: $\mu > 48$

7) p value = 0.000

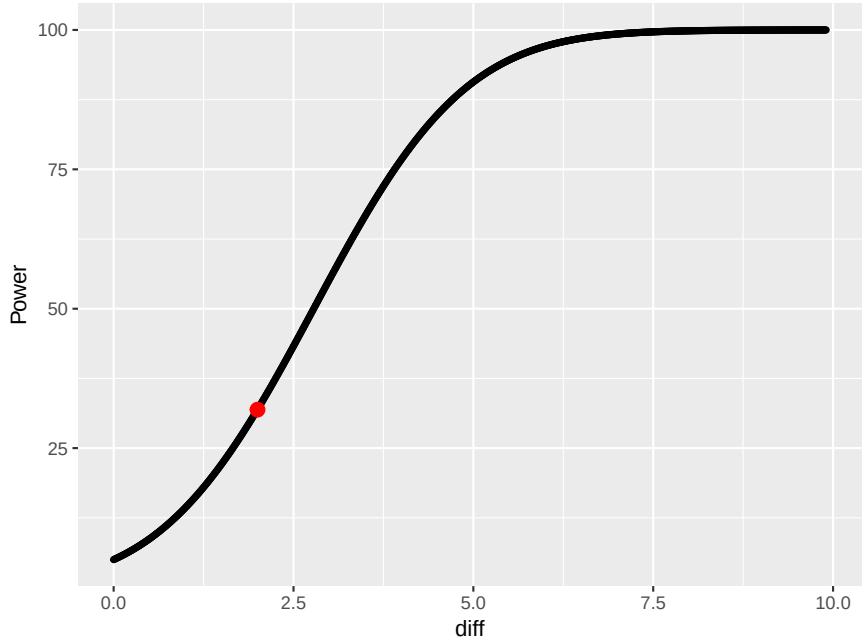
```
one.sample.t( x, mu.null = 48, alternative = "greater")
```




```
## p value of test H0: mu=48 vs. Ha: mu > 48:  0.000
```

8) p<α, we fail to reject the null hypothesis, the add was a success

b. if the add raised the mean sales to $50, what is the power of the test?




```
## Power of Test = 31.9%
```

```
t.ps( n = 40, diff = 50-48, sigma = sd(x), alternative = "greater")
```

c. if the add raised the mean sales to $50, what sample size is needed to have a power of 99%?

```
t.ps( power = 99, diff = 50-48, sigma = sd(x), alternative = "greater")
```

```
## Sample size required is  433
```

17.1.4 Problem 4

Variable: 1 categorical
Problem: hypothesis test

```
chi.gof.test(c(147,71,28,4), c(57,31,10,2))
```

```
## p value of test p=0.7417
```

so we fail to reject the proportions, these people might well have been hispanics

17.1.5 Problem 5

The data set studentsurvey has the replies of students to some questionnaire.

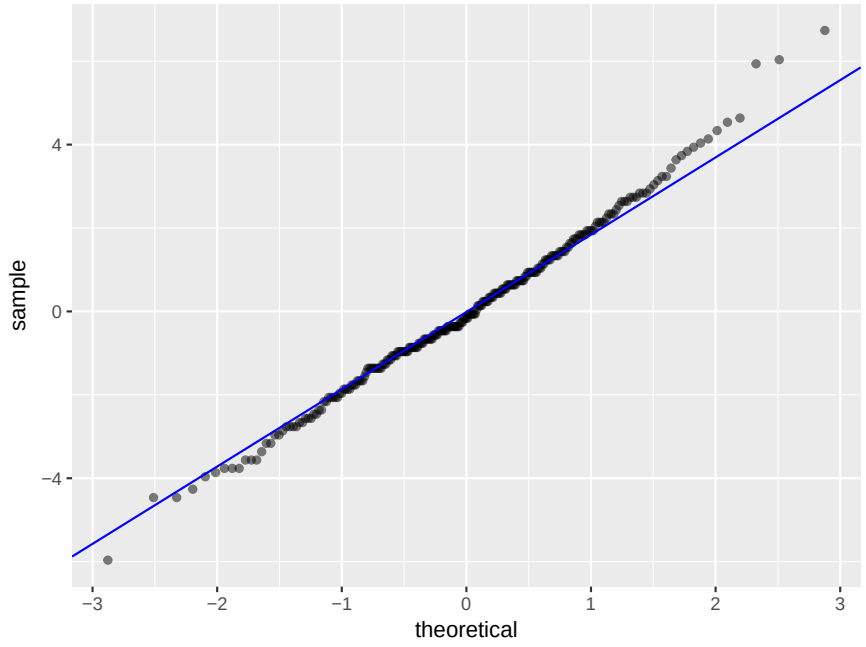
```
attach(studentsurvey)
```

a. find a 90% confidence interval for the mean score

Variable: 1 mean
Problem: confidence interval

Assumptions are ok (checked boxplot and normal plot)

```
one.sample.t(Score, conf.level = 90)
```




```
## A 90% confidence interval for the population mean is (6.1, 6.5)
```

b. test at the 10% level whether there are equally many male and female students.

Variables: 1 Proportion
Problem: Hypothesis test

```
table(Gender)
```

```
## Gender
## Female   Male
##    111    138
```

1) Parameter: proportion π

2) Method: exact binomial

3) Assumptions: none
Assumption is ok

4) $\alpha = 0.1$

5) H_0: $\pi = 0.5$

6) H_a: $\pi \neq 0.5$

7) p-value = 0.0994

```
one.sample.prop( x = 111 , n = 111+138, pi.null = 0.5)
```

```
## p value of test H0: pi=0.5 vs. Ha: pi <> 0.5:  0.0994
```

8) p-value = 0.0994 $< \alpha$, we reject the null hypothesis

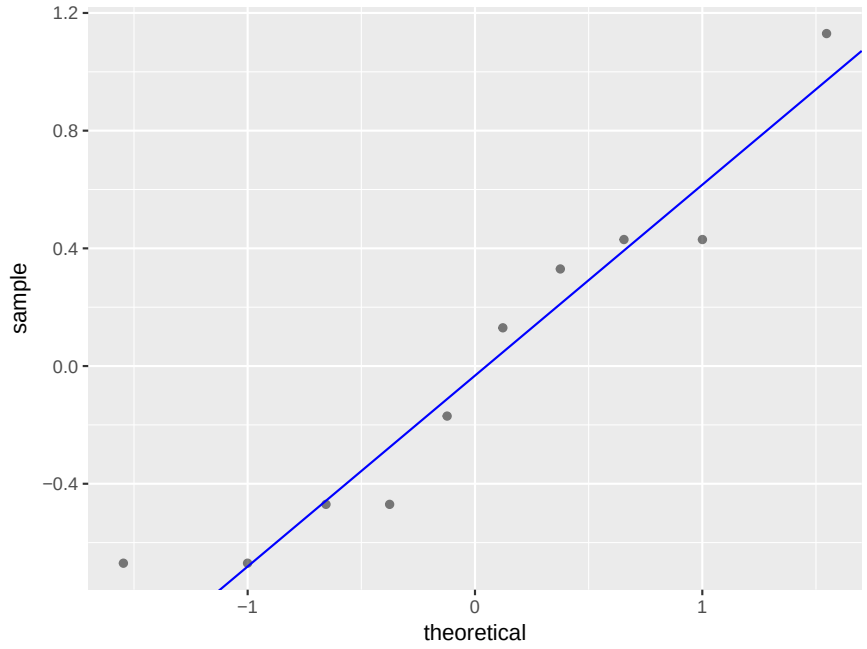
9) it appears there are slightly fewer female students (but this was a very close call, we would have failed to reject the null at the 5% level!)

c. test at the 5% level whether the mean GPA is less than 2

Variables: 1 mean
Problem: hypothesis test

Assumptions are ok (checked boxplot and normal plot) 1) Parameter of interest: population mean
2) Method of analysis: one sample t
3) Assumptions of Method: normal data or large sample
4) Type I error probability $\alpha = 0.05$
5) H_0: $\mu = 2.5$
6) H_a: $\mu < 2.5$
7) p value = 0.000

```
one.sample.t( GPA, mu.null = 2.5, alternative="less")
```




```
## p value of test H0: mu=2.5 vs. Ha: mu < 2.5:  0.0911
```

8) p $< \alpha$, we fail to reject the null hypothesis

9) the population mean GPA is almost certainly less than 2.5

d. test at the 5% level whether the poplation has equally many Freshman, Junior, Senior and Sophomore

Variable: 1 categorical
Problem: hypothesis test

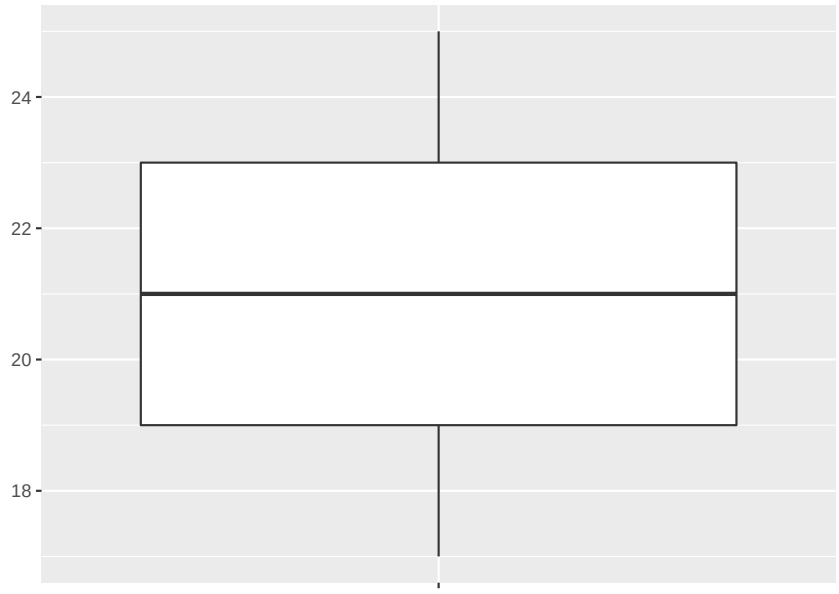
```
chi.gof.test(table(Year), c(1,1,1,1)/4)
```

```
## p value of test p=0.0544
```

p value = 0.0544 > 0.05, so we fail to reject the proportions, but by just a bit.

e. find a 90% confidence interval for the mean age of the students. The boxplot of Age shows a severe outlier. Further investigation shows this to be observation #220. We should remove this observation from the calculation.

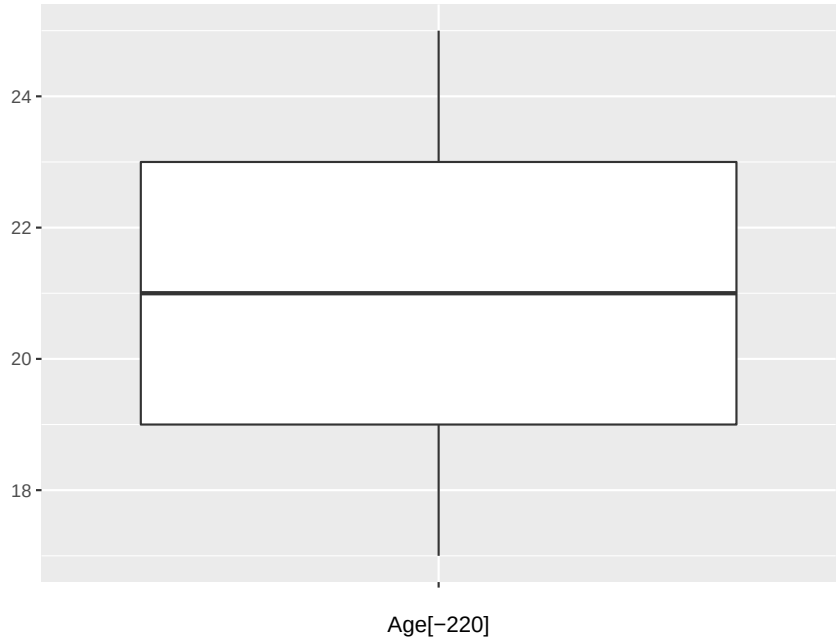
```
bplot(Age)
```



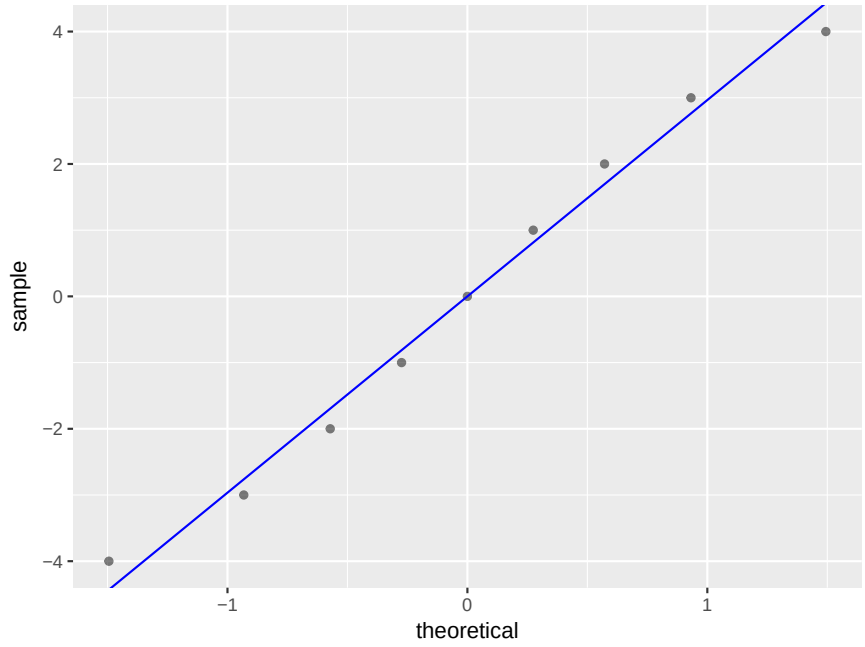

```
which(Age==max(Age))
```

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

```
bplot(Age[-220])
```




```
one.sample.t(Age[-220], conf.level = 90)
```




```
## A 90% confidence interval for the population mean is (19.3, 22.7)
```

17.1.6 Problem 6

In class we talked about Bernoulli trials, that is experiments which have only two possible outcomes. Often one is interested in how often a certain outcome happens when the experiment is carried out a number of times. This is then called a Binomial distribution, and probabilities can be found with the R command dbinom(k, n, p) where n is the number of trials, k how often the outcome happens and p its probability. For example, if we want to know the probability of 2 sixes in 10 rolls of a fair die, it is

```
dbinom(2, 10, 1/6)
```

```
## [1] 0.29071
```

a. if a group consists of 100 men and 90 women and if three people are chosen at random, what is the probability all of them are men?

```
dbinom(3, 3, 100/190)
```

```
## [1] 0.1457938
```

b. if a fair coin is flipped 10 time, what is the probability of at most 3 heads?

at most 3 means either 0 or 1 or 2 or 3, so

```
dbinom(0, 10, 1/2) + dbinom(1, 10, 1/2) + dbinom(2, 10, 1/2) + dbinom(3, 10, 1/2)
```

```
## [1] 0.171875
```

or quicker:

```r
sum(dbinom(0:3, 10, 1/2))
```

```
## [1] 0.171875
```

c. if the probability of having an accident on any one mile of road is 0.001, what is the probability of having at least one accident when driving 10000 miles in one year?

Prob(at least one accident) = 1-Prob(0 accidents)

```r
1-dbinom(0, 10000, 0.0001)
```

```
## [1] 0.632139
```

d. if a fair coin is flipped 100 times, what is the probability of getting between 40 and 60 heads (include 40 and 60) ? (Compare that our discussion of the coin app)

```r
sum(dbinom(40:60, 100, 1/2))
```

```
## [1] 0.9647998
```

17.1.7 Problem 7

The mean score in the final exam of a Calculus course over many years was 72.3. The University wants to decide whether or not to change the text book, and so they are planning to teach several sections of the course with a new text book. Then they will test H_0: μ=72.3 vs H_a: μ>72.3 at the 5% level. If it is true that with this new text book the mean score will go up to 75.5 points, how many students do they need to have so that the hypothesis test has a power of 80%? (assume the standard deviation is 15.0)

```r
t.ps(diff=75.5-72.3, sigma=15.0, power=80, alternative="greater")
```

```
## Sample size required is  138
```

17.1.8 Problem 8

At a certain moment in time the national unemployment rate was 6.9%. In one city among 250 randomly selected people 30 said they were unemployed. Test at the 10% level to check whether in this city the unemplyoment rate differs from that nationwide.

1) Parameter: proportion π

2) Method: exact binomial

3) Assumptions: none
Assumption is ok

4) $\alpha = 0.1$

5) H_0: $\pi = 0.069$

6) H_a: $\pi \neq 0.069$

7) p-value=0.00223

```
one.sample.prop(x=30, n=250, pi.null=0.069)
```

```
## p value of test H0: pi=0.069 vs. Ha: pi <> 0.069:  0.0022
```

8) p-value = 0.00223 $< \alpha$, so we reject the null hypothesis

9) the true unemployment rate in this city is not 6.9%

17.2 Case Study: Rogaine - Treatment for Hair Loss

Rogaine is the first treatment for hair loss approved by the Food and Drug Administration. Here we have the results of one of the studies that were done to show that rogaine works. A randomized clinical trial was carried out. 1431 bald men were randomly assigned to two groups. The men in the treatment group received Rogaine, the men in the control group received a placebo. After some time the men were examined and assigned to one of 5 groups:

- No Growth = no difference in amount of hair
- New Vellus = some hair follicles
- Min Growth = minimal hair growth
- Mod Growth = moderate hair growth
- Den Growth = dense hair growth

Basic Question: Does rogaine work?**

Type of variables:

```
head(rogaine)
```

```
##      Growth     Group
## 1 No Growth Treatment
## 2 No Growth Treatment
## 3 No Growth Treatment
## 4 No Growth Treatment
## 5 No Growth Treatment
## 6 No Growth Treatment
```

Growth: Values = No Growth, .., Dense Growth are text, therefore categorical

Group: Values: Treatment, Control, are text, therefore categorical

two categorical variables → categorical data analysis

Usually the first thing to do is to just count the number of times each combination has happened:

```
attach(rogaine)
table(rogaine)
```

```
##             Group
## Growth       Control Treatment
##   No Growth      423       301
##   New Vellus     150       172
##   Min Growth     114       178
##   Mod Growth      29        58
##   Den Growth       1         5
```

So, does rogaine work? This is a yes-no question, so we need to do a hypothesis test. The most popular method here is the **chisquare test for independence**. It has

H_0: Classifications are independent (here: Rogaine does **not** work)

H_a: Classifications are dependent (here: Rogaine works)

But why are "Classifications are independent" and "Rogaine does not work" the same thing? Consider: say we randomly choose one of the 1431 men that were part of this study. We do not know whether he received Rogaine or the placebo. What is the probability that the man had no growth? Well:

$724/1431 * 100 = 50.6\%$

Let's assume for the moment that rogaine is useless, it does no better than the placebo. In that case it would make no difference if we were also told that he used Rogaine, we should make the same guess of 50.6%. So knowing the value of the predictor (Rogaine or placebo) does not make any difference for the response (Hair growth)

they are **independent**

What are the assumptions of this method? They are that none of the **expected counts** is less than 5.

To run the test use the **chi.ind.test** command. The argument has to be a table, so we run

```
chi.ind.test(table(rogaine))
```

```
## Some expected counts < 5!
## p value of test p=0.000
```

so the p value of the test is 0, and we should reject the null hypothesis of no relationship.

There is however also a warning:

Some expected counts < 5!

This part is because there is a problem with the expected counts.

This generally happens when there is not enough data for some combinations. Looking at the table above it seems the number in the row Den Growth are to small. We can fix that by combining the Den Growth and the Mod Growth groups:

```
new.rogaine.table <- cbind(c(423, 150, 114, 29+1), c(301, 172, 178, 58+5))
new.rogaine.table
```

```
##      [,1] [,2]
## [1,]  423  301
## [2,]  150  172
## [3,]  114  178
## [4,]   30   63
```

```
chi.ind.test(new.rogaine.table)
```

```
## p value of test p=0.000
```

So, here is the test:

1. Parameters of interest: measure of association
2. Method of analysis: chi-square test of independence
3. Assumptions of Method: all expected counts greater than 5
4. $\alpha = 0.05$
5. H_0: Classifications are independent = Rogaine does **not** work
6. H_a: Classifications are dependent = Rogaine works
7. p = 0.000
8. we reject the null hypothesis, there is a statistically significant difference between Rogaine and Placebo, Rogaine works better than the Placebo (or doing nothing)

Note when we made the new table by combining the last two categories I didn't bother to add the row and column names, because the chi.ind.test command doesn't use them anyway. It would of course have been easy to do so:

```
colnames(new.rogaine.table) <- c("Control", "Treatment")
rownames(new.rogaine.table) <- c("No Growth", "New Vellus", "Min Growth", "Some Growth")
new.rogaine.table
```

```
##             Control Treatment
## No Growth       423       301
## New Vellus      150       172
## Min Growth      114       178
## Some Growth      30        63
```

17.2.1 Case Study : Drownings in Los Angeles

Data is from O'Carroll PW, Alkon E, Weiss B. Drowning mortality in Los Angeles County, 1976 to 1984, JAMA, 1988 Jul 15;260(3):380-3.

Drowning is the fourth leading cause of unintentional injury death in Los Angeles County. They examined data collected by the Los Angeles County Coroner's Office on drownings that occurred in the county from 1976 through 1984. There were 1587 drownings (1130 males and 457 females) during this nine-year period

```
##                          Male Female
## Private Swimming Pool     488    219
## Bathtub                   115    132
## Ocean                     231     40
## Freshwater bodies         155     19
## Hottubs                    16     15
## Reservoirs                 32      2
## Other Pools                46     14
## Pails, basins, toilets      7      4
## Other                      40     12
```

Basic Question: **is there a difference between men and women and the method of drowning?**

First notice that here the data is already in the form of a table. The "original" data would have been something like this:

Female - Private Swimming Pool
Female - Bathtub Male - Ocean

and so on

Type of variables:

This is not so trivial, at first glance it seems we have numerical data, but in fact this is already a table, the original ("raw") data was two pieces of information for each subject, namely

Gender: Values: "Male", "Female" are text, therefore **categorical**

Method: Values: "Private Swimming Pool", .., "Other", are text, therefore **categorical**

two categorical variables → categorical data analysis

Notice also an added difficulty here: at first glance it appears that more than twice as many men drown in Private Swimming Pools than do women (488 vs 219), but remember, there are twice as many men who drowned altogether, so if there were no differences between men and women we would expect twice as men as women to drown in Private Swimming Pools. What we need to do is calculate the percentages:

```
round(Male/sum(Male)*100, 1)
```

```
## [1] 43.2 10.2 20.4 13.7  1.4  2.8  4.1  0.6  3.5
```

```
round(Female/sum(Female)*100, 1)
```

```
## [1] 47.9 28.9  8.8  4.2  3.3  0.4  3.1  0.9  2.6
```

and we see that the difference between men and women who drown in Private Swimming Pools is not very large (43.2% vs. 47.9%)

Notice this was not necessary in the Rogaine data because there the groups Treatment and Control had (almost) the same size (714 and 717). This is often the case in a **designed** experiment, whereas in an **observational** study such the drowning example we often have different sample sizes.

Generally, using percentages for tables (and graphs) is rarely wrong but using counts can be (and would be here in the drowning experiment) Finally, the test.

```
chi.ind.test(drownings)
```

```
## Some expected counts < 5!
## p value of test p=0.000
```

Again we get the warning message. This time it is the Pails, basins, toilets group, which we should combine with the Other:

```
newmale <- c(drownings[1:7, 1], 7+40)
newfemale <- c(drownings[1:7, 2], 4+12)
newdrown <- cbind(newmale, newfemale)
newdrown
```

```
##      newmale newfemale
## [1,]     488       219
## [2,]     115       132
## [3,]     231        40
## [4,]     155        19
## [5,]      16        15
## [6,]      32         2
## [7,]      46        14
## [8,]      47        16
```

```
chi.ind.test(newdrown)
```

```
## p value of test p=0.000
```

1. Parameters of interest: measure of association

2. Method of analysis: chi-square test of independence

3. Assumptions of Method: all expected counts greater than 5

4. Type I error probability α=0.05

5. H_0: Classifications are independent = there is no difference in the method of drowning between men and women.

6. H_a: Classifications are dependent = there is some difference in the method of drowning between men and women.

7. p-value = 0.000

8. p<α, we reject the null hypothesis, there is a statistically significant difference between men and women and where they drown.

Note In this case combining the Pails, .. and the Other category did the trick. But in the Reservoire category there is also a small number (2). We could also have tried to combine that one with Other. Try it to see whether it would have worked!

17.2.2 Case Study

A psychologist has developed a new method for treating depression. She wants to know whether it works equally well on men and women. She randomly selects 25 patients and after applying her new treatment she "measures" their level of improvement. She finds:

	Male	Female
No Improvement	5	6
Some Improvement	7	8
Improvement	7	5

We want to test at the 5% level whether there is a difference between men and women and how the method works.

Type of variables:

Gender: Values: "Male", "Female" are text, therefore **categorical**

Method: Values: "No Improvement", "Some Improvement", "Improvement" are text, therefore **categorical**

two categorical variables → categorical data analyis

First we need to get the data into R. In the case of a small table like this the quickest way to do this is to just type it in:

```
x <- cbind(c(5, 7, 7), c(6, 8, 5))
```

Notice that for the *chi.ind.test* command we only need the numbers, not the names, so this is enough.

Now

```
chi.ind.test(x)
```

```
## p value of test p=0.7823
```

1. Parameters of interest: measure of association

2. Method of analysis: chi-square test of independence

3. Assumptions of Method: all expected counts greater than 5

4. Type I error probability α=0.05

5. H_0: Classifications are independent = there is no difference in the method between men and women.

6. H_a: Classifications are dependent = there is some difference in the method between men and women.

7. p-value = 0.7823
8. p<α, we fail to reject the null hypothesis, there is no statistically significant difference between men and women and how the method works.

17.2.3 Always be careful with the details

Consider the following two experiments:

Experiment 1: we randomy select 200 subjects. For each we record their gender and ask them whether they have ever been accused of a crime. We find

Gender	Yes	No
Male	30	70
Female	17	83

Experiment 2: we randomly select 100 men and 100 women and ask them whether they have ever been accused of a crime. We find

Gender	Yes	No
Male	30	70
Female	17	83

Note that the data in both experiments is exactly the same, but they are in fact very different experiments. In experiment 1 the fact that we had 100 men was accidental, it could just as easily have been 104 (say). On the other hand in experiment 2 we decided ahead of time how many men and how many women we want in the study.

How an experiment is done is always an important consideration. For example, if we wanted to answer the question whether there is a relationship between gender and accused, we could use the chisquare test for experiment 1 but not for experiment 2, at least not in the form presented here. This leads to a whole subfield of statistics called **experimental design**.

For more on the chisquare test for independence see section 12.2 of the textbook.

Figure 22:

18 Quantitative Predictor - Quantitative Response: Pearson's Correlation Coefficient

The method discussed here was originally developed by Karl Pearson

18.0.1 Case Study: The 1970's Military Draft

In 1970, Congress instituted a random selection process for the military draft. All 366 possible birth dates were placed in plastic capsules in a rotating drum and were selected one by one. The first date drawn from the drum received draft number one and eligible men born on that date were drafted first. In a truly random lottery there should be no relationship between the date and the draft number.

CBS TV Broadcast

Basic question: **Did the 1970 draft work the way it was supposed to?**

```
head(draft[, 4:5])
```

```
##   Day.of.Year Draft.Number
## 1           1          305
## 2           2          159
## 3           3          251
## 4           4          215
```

```
## 5           5           101
## 6           6           224
```

18.0.2 Type of variables:

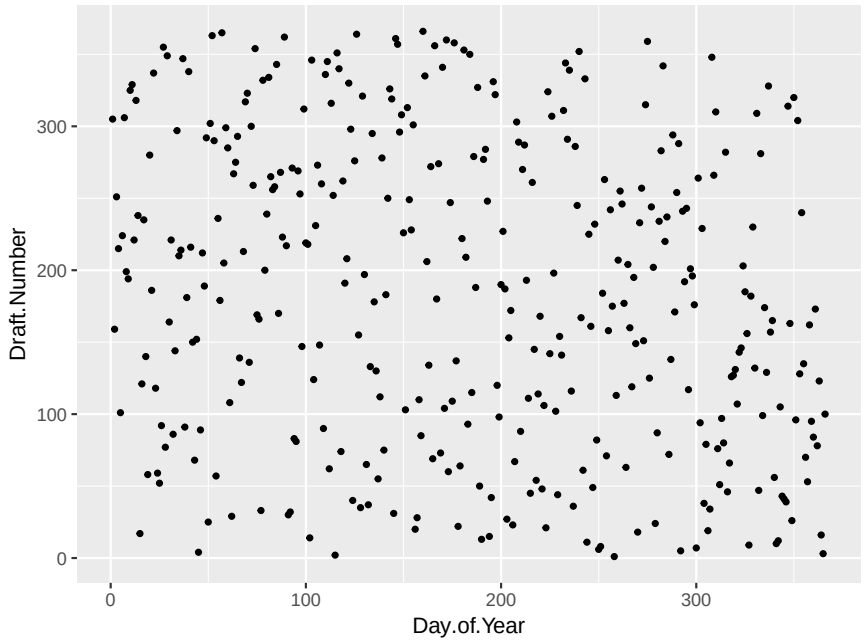
Day of Year : Values: 1, 2, 3, .. ,366 are numerical, therefore **quantitative**

Draft Number.: Values: 305, 159, 251, 215, ... are numerical, therefore **quantitative**

two quantitative variables → correlation

Whenever we want ot study the relationship between two quantitative variables we should start with the scatterplot. But before we do this, let's consider what we expect to see. The draft was designed as a lottery to make it fair, that is any man in the US should have had the same chance to be picked (or not!). In terms of their birthdays, each of the 365 days should have had the same chance of getting picked early, sometime in the middle or late. So some of the days in January should have a small Draft Number, some a large one and some in the middle. So in the scatterplot on the left (January=small Day number) we should see some dots on the bottom (small Draft) some in the middle and some on top. And exactly the same should be true for any other month:

```
attach(draft)
splot(Draft.Number, Day.of.Year)
```



so far, so good.

Now, this graph shows that there is **no obvious** relationship between the variables, but we need a bit more, we want there to be **no relationship**, we want Day of Year and Draft Number to be **independent**. A graph such as this one is not quite enough. So let's

calculate a statistic that measures the relationship between two quantitative variables, namely **Pearson's correlation coefficient**:

```
cor(Draft.Number, Day.of.Year)
```

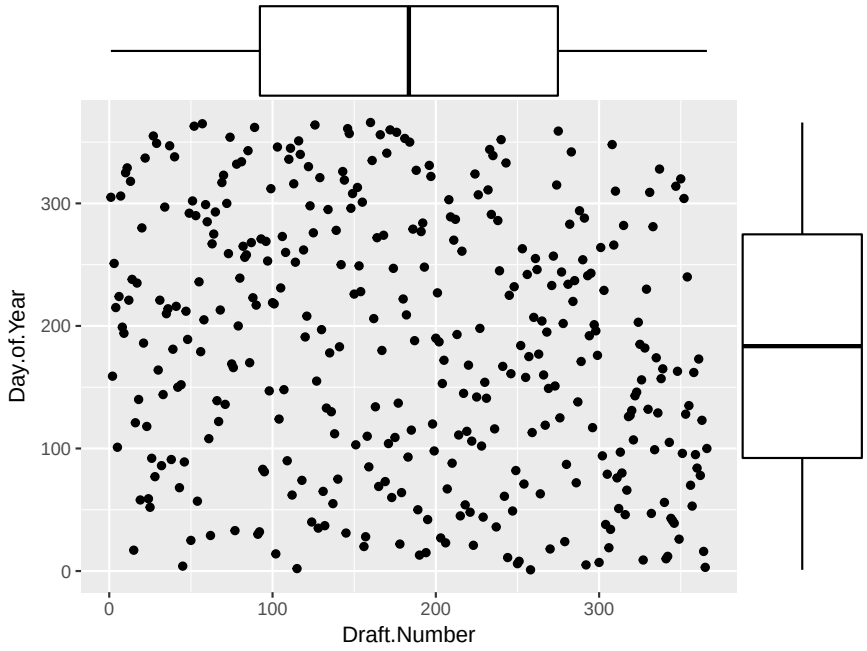
```
## [1] -0.2260414
```

Recall some of the properties of Pearson's correlation coefficient: - always -1 $\leq$ r $\leq$ 1

- r close to 0 means very small or even no correlation (relationship)
- r close to $\pm$ 1 means a very strong correlation
- r = -1 or r = 1 means a perfect linear correlation (that is in the scatterplot the dots form a straight line)
- r $<$ 0 means a negative relationship (as x gets bigger y gets smaller)
- r $>$ 0 means a positive relationship (as x gets bigger y gets bigger)
- r treats x and y symmetricaly, that is cor(x,y) = cor(y,x)

r is a **statistic** (a number calculated from a **sample**) so it has a corresponding **parameter** (a number describing a **population**) The parameter is usually denoted by ρ. If the lottery worked and was fair, then we should have $\rho = 0$. So the question becomes: if r = -0.226 could we still have $\rho = 0$? Again this is answered by a hypothesis test:

```
pearson.cor(Draft.Number, Day.of.Year, rho.null = 0)
```




```
## p value of test H0: rho=0 vs. Ha: rho <> 0:  0.000
```

1. Parameter of interest: Pearson's correlation coefficient ρ

2. Method of analysis: test based on normal theory

3. Assumptions of Method: relationship is linear, there are no outliers.
4. $\alpha = 0.05$

5. H_0: $\rho = 0$ (no relationship between "Day of Year" and "Draft Number")

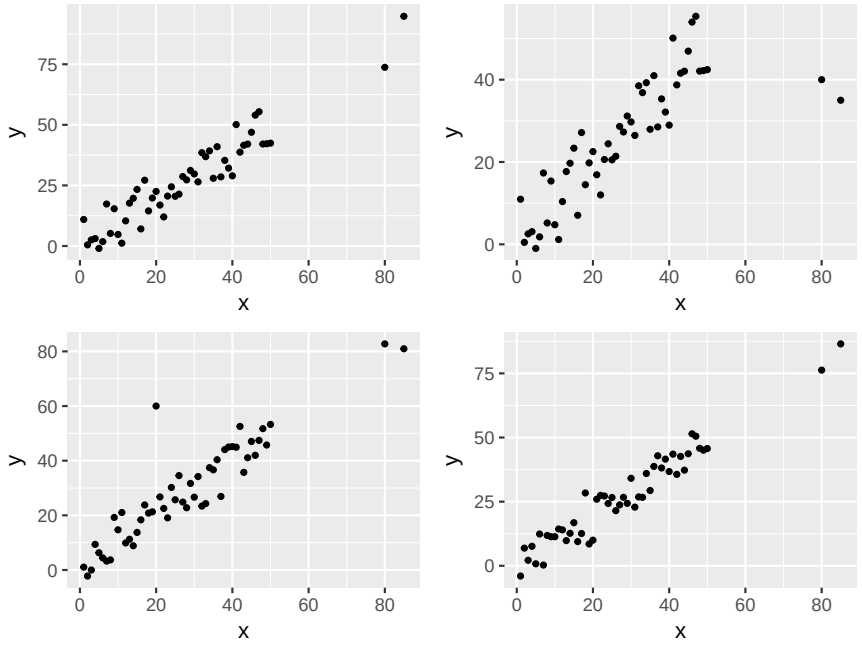
6. H_a: $\rho \neq 0$ (some relationship between "Day of Year" and "Draft Number")

7. p = 0.000

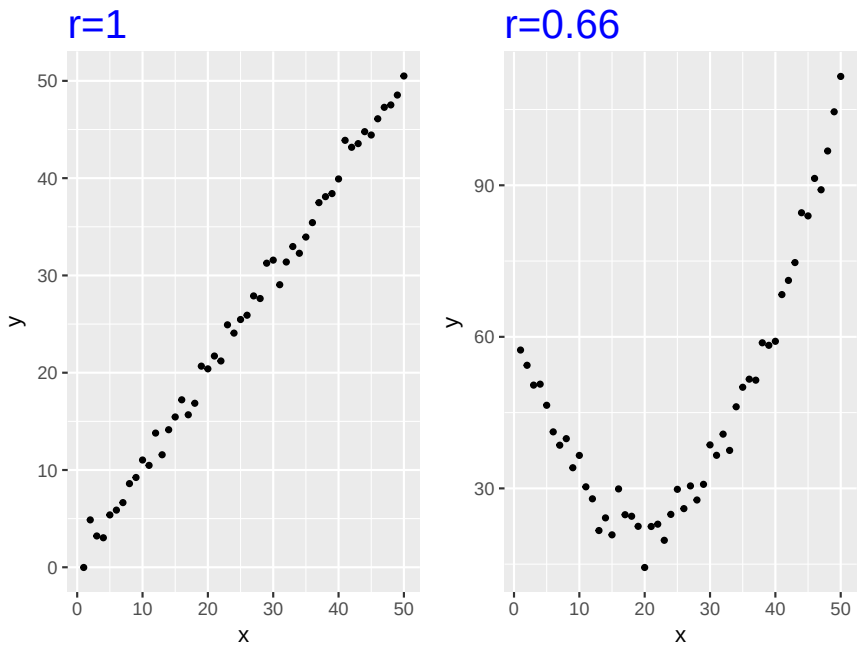
8. $p < \alpha$, so we reject H_0, there is some relationship between "Day of Year" and "Draft Number", something went wrong in the 1970 draft.

How about the assumptions? we can check them using the marginal plot, which looks just fine.

Here are some cases were Pearson's correlation coefficient would not work:



Also very important is the fact that Pearson's correlation coefficient works only for **linear relationships**:



18.0.3 App: correlation and correlation2

these apps illustrate the correlation coefficient

correlation What to do:

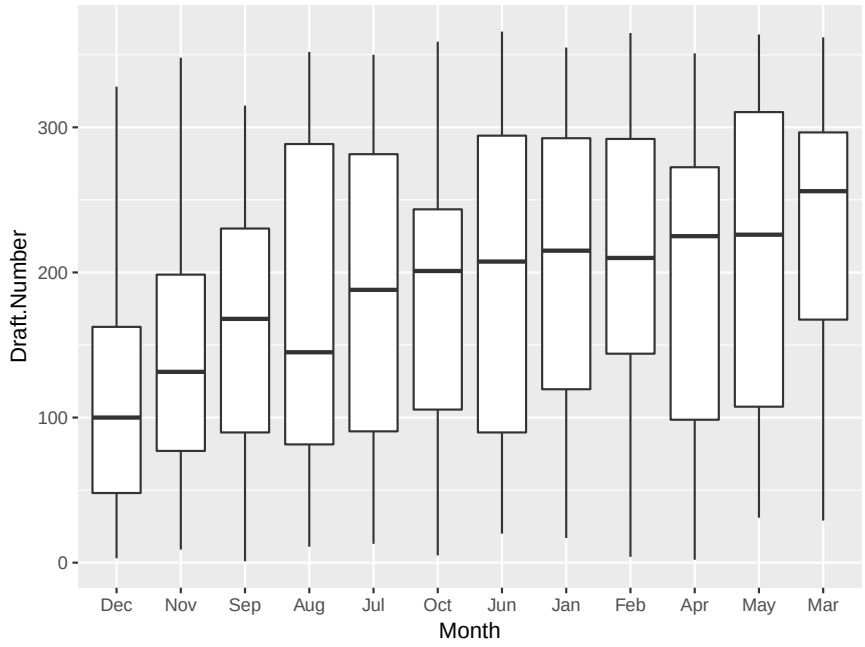
Move slider around to see different cases of the scatterplot of correlated variables include a few outliers and see how that effects that "look" of the scatterplot and the sample correlation coefficient On the Histogram tab set ρ = -0.23 and observe that we need a sample size of about 60 to have some reasonable chance to reject the null hypothesis of no correlation.

correlation2 What to do

click inside graph and watch the correlation

So, now that we know that there is indeed a relationship between Day of Year and Draft Number, can we visualize it in some way? Here is an idea: let's look at the boxplot of Draft Number by Month:

```
bplot(Draft.Number, Month,
  new_order = "Size")
```

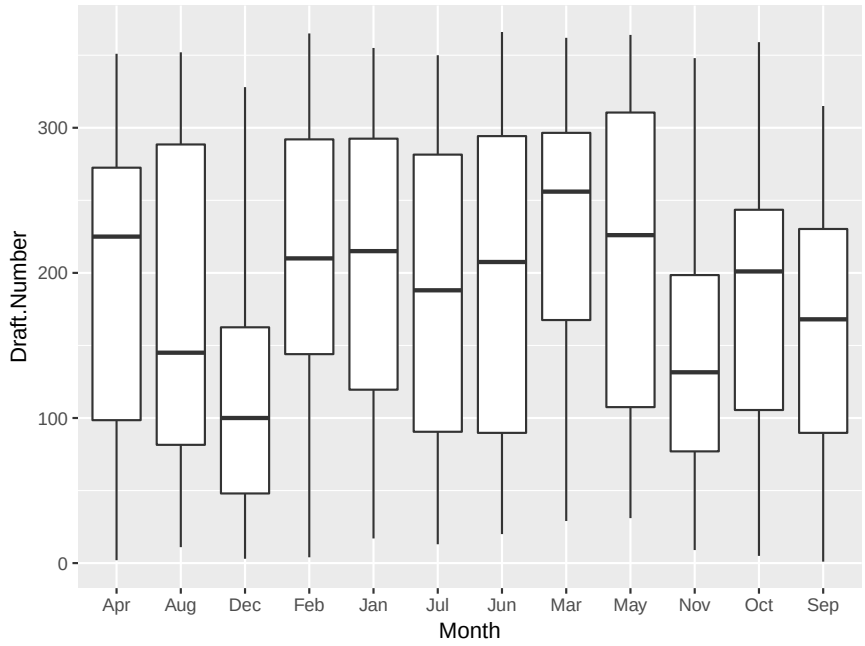


and here we can see that there is a tendency for the Draft Numbers to be lower for month later in the year.

Note

if we simply run

```
bplot(Draft.Number, Month)
```



the routine arranges the boxed alphabetically. Here we want them arranged in order. We can always do the graph in any order we want with the new_order argument.

Again notice the similarities and the differences between this analysis and those we have done before: in each case we had the basic question of whether or not there is a relationship between two variables, in each case we did the hypothesis test with the null hypothesis

H_0: there is **no** relationship

but then we used different methods depending on the type of data:

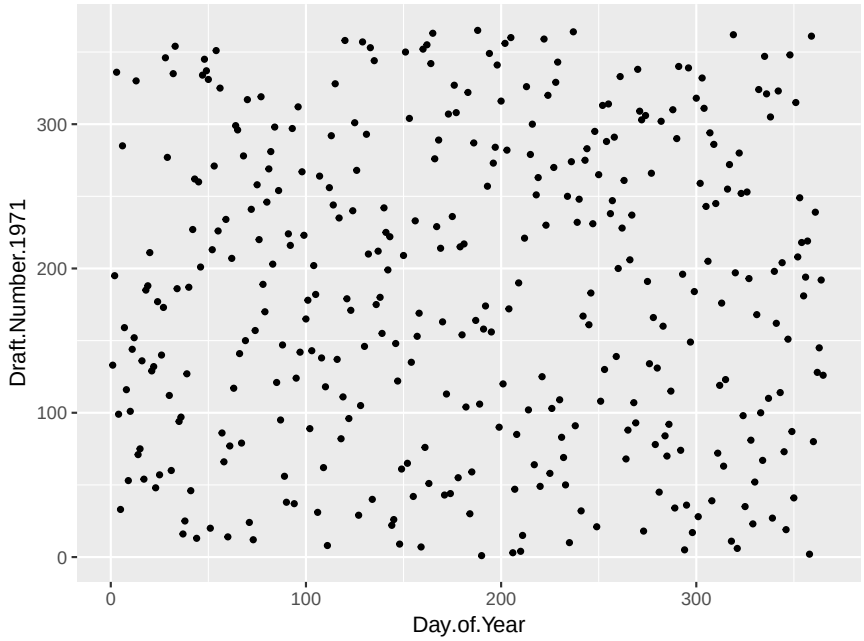
- Categorical Predictor - Categorical Response: Chi-square test for independence
- Categorical Predictor - Quantitative Response: ANOVA
- Quantitative Predictor - Quantitative Response: Pearson's Correlation Coefficient

For the last two there are even more similarities: for each of these methods there was some assumption of normal distributions

18.0.4 Case Study: The 1971 Military Draft

let's see what happened the year after:

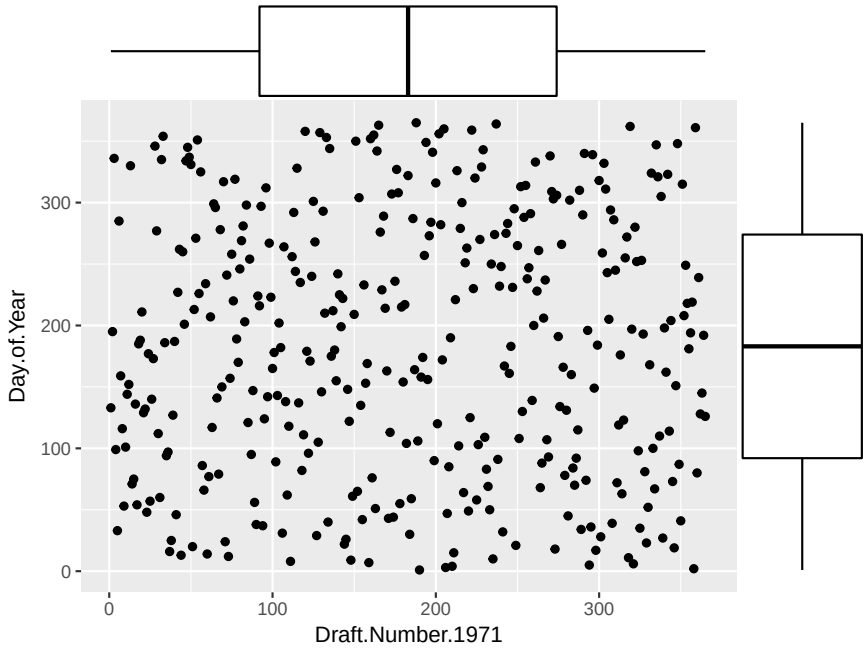
```
splot(Draft.Number.1971, Day.of.Year)
```



so there is no hint of a problem here (but again, that is what we thought before as well).

And the test:

```
pearson.cor(Draft.Number.1971, Day.of.Year, rho.null = 0)
```

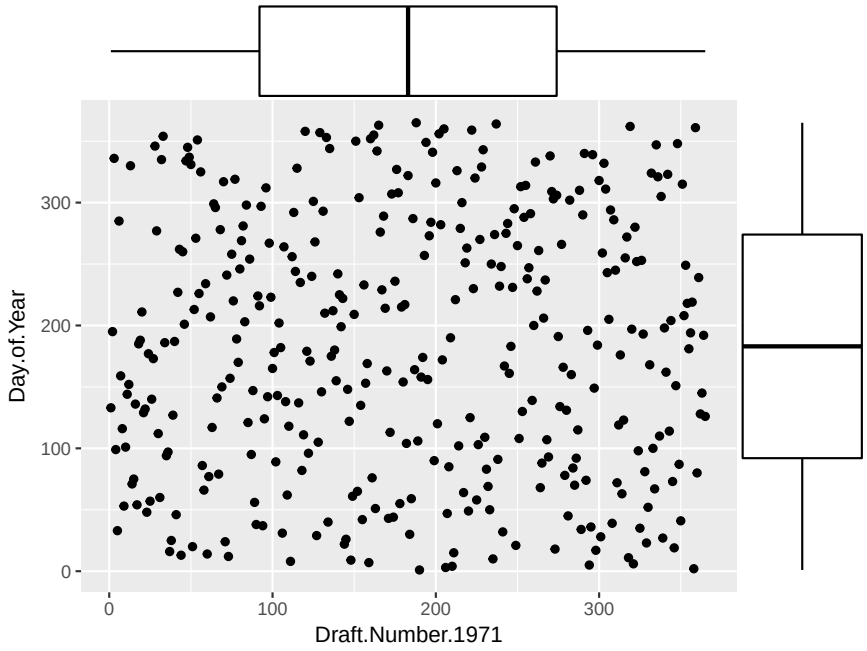



```
## p value of test H0: rho=0 vs. Ha: rho <> 0:  0.7861
```

1. Parameter of interest: Pearson's correlation coefficient ρ
2. Method of analysis: test based on normal theory
3. Assumptions of Method: relationship is linear, there are no outliers
4. α=0.05
5. H_0: $\rho = 0$ (no relationship between "Day 1971" and "Draft Number 1971")
6. H_a: $\rho \neq 0$ (some relationship between "Day 1971" and "Draft Number 1971")
7. p = 0.7861
8. p $> \alpha$, so we fail to reject H_0, there is no relationship between "Day 1971" and "Draft Number 1971". The marginal plot shows no outliers or a non linear relationship, so the assumptions are ok.

The same command can also be used to find a confidence interval. This is done when the rho.null argument is left off:

```
pearson.cor(Draft.Number.1971, Day.of.Year, conf.level = 90)
```




```
## A 90% confidence interval for the
## population correlation coefficient is ( -0.072, 0.1 )
```

19 Correlation vs. Causation

Say we have found correlation between variables "x" and "y". How can we understand and interpret that relationship?

Toy Example

x = Number of fireman responding to a fire

y = damages done by the fire.

say there is a positive correlation between x and y (and in real live there will be!)

does this mean x causes y?

Please note saying x causes y is not the same as x determines y. There are usually many other factors besides x that influence y, maybe even some more important than x. For example say x="Time studied for Exam" and y="Score on Exam". Let's assume that there is a positive correlation between x and y. It is reasonable to conclude that x causes y, that is studying longer improves the scores. But of course there are also many other factors such as general ability, previous experience, being healthy on the day of the exam, exam anxiety, having a hang-over etc.

Confusing correlation with causation can sometimes lead to very strange conclusions:

Bizzare Correlations

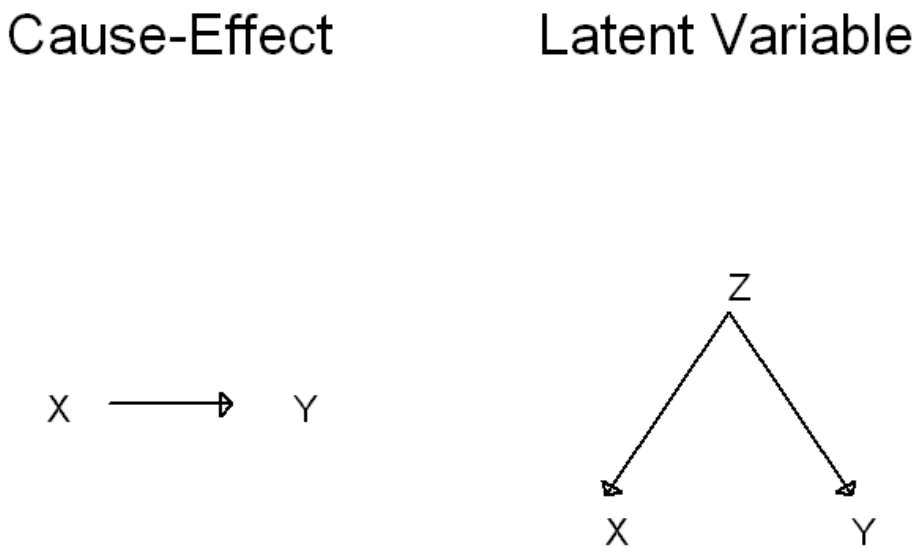


Figure 23:

but the real danger are cases were ther is no causation but it is not so obvious. In fact trying to determine whther an observed correlation is due to causation is one of the genuinely hard problems in Statistics:

19.0.1 Case Study: Smoking and Lung Cancer

There have been hundreds of studies all over the world that have shown a correlation between smoking rates and lung cancer deaths, usually with correlations of about 0.5 to 0.7. And yet, none of these studies has shown that smoking causes lung cancer because all of the were observational studies, not clinical trial.

The only perfectly satisfactory way to establish a causation is to find a random sample, for example to do a **clinical trial**. An **observational study** is always somewhat suspect because we never know about hidden biases. Nevertheless, even only using observational studies the evidence for cause-effect can be quite strong: Things to look for when trying to establish a causation:

- correlation is strong - the correlation between smoking and lung cancer is very strong
- correlation is consistent over many experiments - many studies of different kinds of people in different countries over a long time period all have shown this correlation
- higher doses are associated with stronger responses - people who smoke more have a higher chance of lung cancer
- the cause comes before the response in time - lung cancer develops after years of smoking. The number of men dying of lung cancer rose as smoking became more common, with a lag of about 30 years. Lung cancer kills more men than any other form of cancer. Lung cancer was rare among women until women started to smoke. Lung cancer in women rose along with smoking, again with a lag of about 30 years, and has now passed breast cancer as the leading cause of cancer deaths among women.
- the cause is plausible - lab experiments on animals show that nicotin causes cancer.

This issue goes by the name of "Correlation does not imply Causation", but it is not specific to quantitative data (where we use the word "correlation"), it comes up anytime we compare two variables:

Example Rogaine: we rejected the null hypothesis of no relationship between the treatment (Rogaine or Placebo) and hair growth. Does this mean Rogaine causes hair to grow? There is nothing in the statistical analysis that says so, but because this was a clinical trial with random assignments to treatment and control there can be no latent variable, so the answer is yes.

Example Mothers and Cocain use: we rejected the null hypothesis of no relationship between the drug use of the mother and the length of the babies. Does this mean the drug use is the cause for the difference?

Again there is nothing in the statistical analysis that says so, and this was an observational study, so there is the possibility of a latent variable. It is difficult to imagine what that would be, though.

20 Non-Normal Residuals, No Equal Variance - Transformations

20.1 Categorical - Quantitative

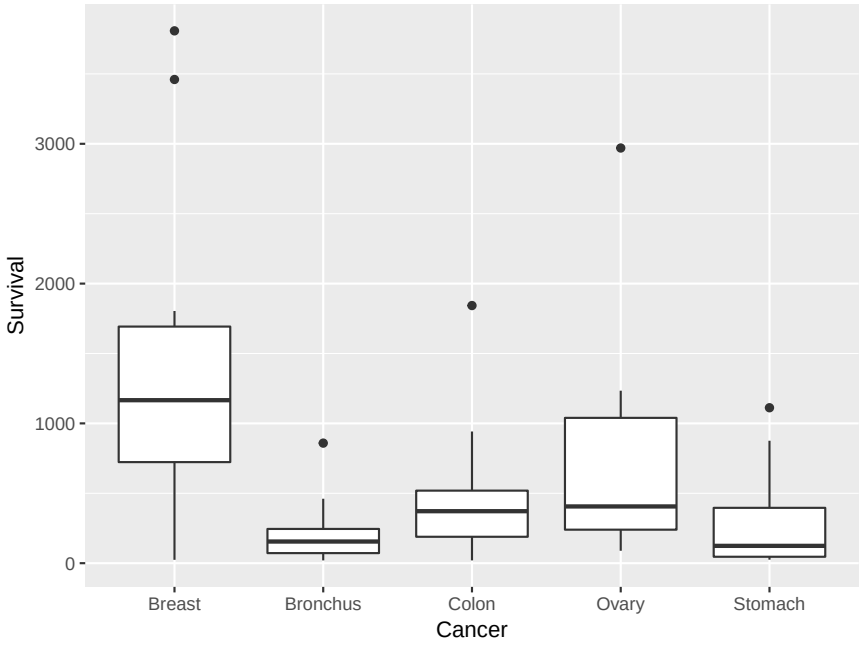
20.1.1 Case Study: Cancer Survival

As we saw before, the boxplot of this data shows some severe outliers:

```
attach(cancersurvival)
head(cancersurvival)
```

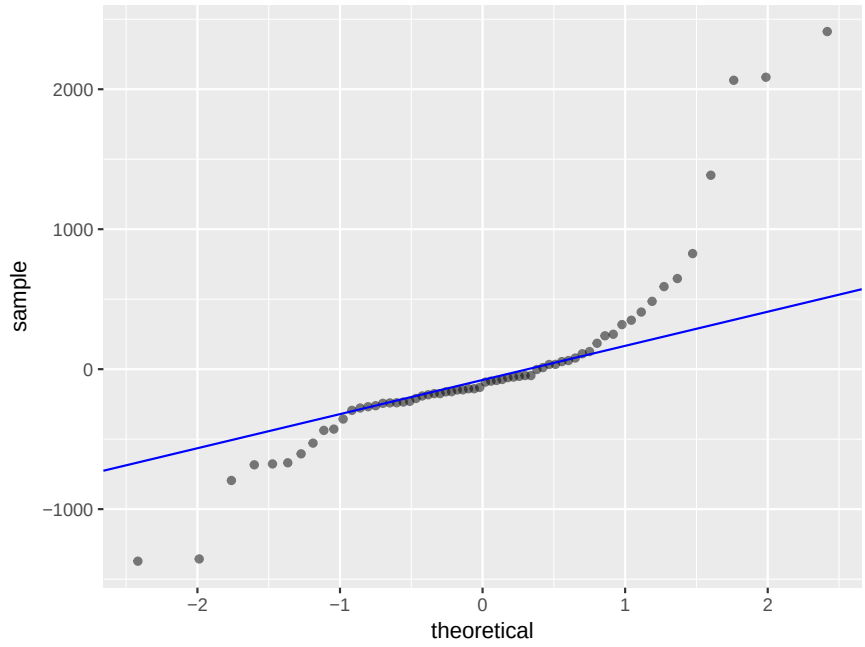
```
##   Survival  Cancer
## 1      124 Stomach
## 2       42 Stomach
## 3       25 Stomach
## 4       45 Stomach
## 5      412 Stomach
## 6       51 Stomach
```

```
bplot(Survival, Cancer)
```



These are often an indication that there is a problem with the assumption of normaly distributed residuals. In fact, when we run the ANOVA and check the normal plot we can see that this is the case:

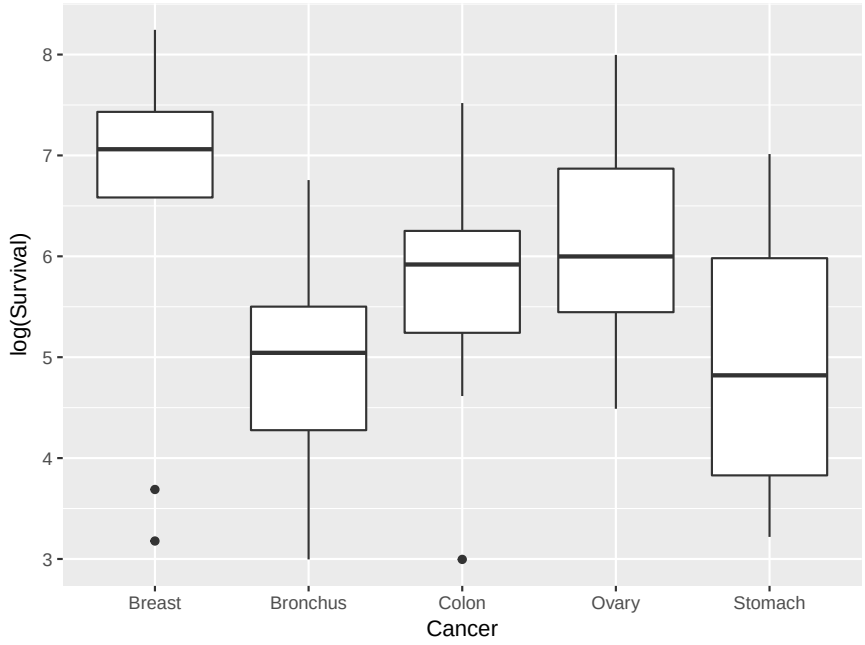
```
oneway(Survival,Cancer)
```




```
## p value of test of equal means: p = 0.000
## Smallest sd:  209.9    Largest sd : 1239
```

So, what can we do? One possible solution is to use a **log transformation**:

```
bplot(log(Survival), Cancer)
```



This takes care of (most) of the outliers.

Outliers often have another effect:

```
stat.table(Survival, Cancer)
```

```
##           Sample Size   Mean Standard Deviation
## Stomach            13  286.0              346.3
## Bronchus           17  211.6              209.9
## Colon              17  457.4              427.2
## Ovary               6  884.3             1098.6
## Breast             11 1395.9             1239.0
```

shows we also have a problem with the equal variance: smallest stdev=210, largest stdev=1239, 3*210=630 $<$ 1239.

In this class we will use the log transform only. In real live there are a number of other transforms one can try, such as square root and inverse.

Note sometimes in a quantitative variable some values are 0, but log(0) does not exist!. In this case use log(x+1). Even worse, sometimes numbers are negative, and again log(negative number) does not exist. In that case use log(x+a) so that all x+a>0

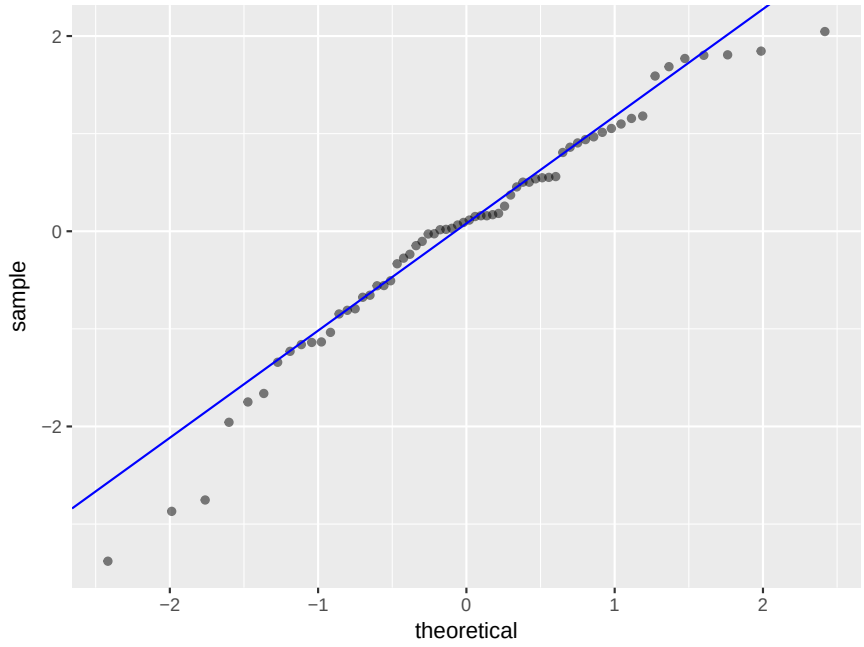
We already know that outliers have a strong effect on the mean and the standard deviation. It might therefore be better to use a summary table based on median and iqr:

```
stat.table(Survival, Cancer, Mean=FALSE)
```

```
##           Sample Size Median   IQR
## Stomach            13    124 350.0
## Bronchus           17    155 173.0
## Colon              17    372 330.0
## Ovary               6    406 799.8
## Breast             11   1166 969.5
```

Now we can finish the analysis of this dataset:

```
oneway(log(Survival),Cancer)
```




```
## p value of test of equal means: p = 0.0041
## Smallest sd:  1    Largest sd : 1.6
```

1. Parameters of interest: group means

2. Method of analysis: ANOVA

3. Assumptions of Method: residuals have a normal distribution, groups have equal variance

4. $\alpha = 0.05$

5. Null hypothesis H_0: $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$ (groups have the same means)
6. Alternative hypothesis H_a: $\mu_i \neq \mu_j$ (at least two groups have different means)

7. p value = 0.0041

8. $0.0041 < 0.05$, there is some evidence that the group means are not the same, there are differences in the survival times. Assumptions: a normal plot of residuals ok
   b smallest stdev = 1.0, largest stdev = 1.6, 3*1.0 = 3.0 > 1.6 ok

Notice that the transformation solves both the problem of the normal residuals as well as the problem of unequal variances! This is quite often the case, though not always.

20.1.2 Case Study: Capacity of Wells

The specific capacity of wells in the Appalachian mountain region of Pennsylvania has been measured in four rock types. (Knopman 1990) The rock types are dolomite, limestone, siliclastic and metamorphic. The capacities are recorded in gal/min/ft.

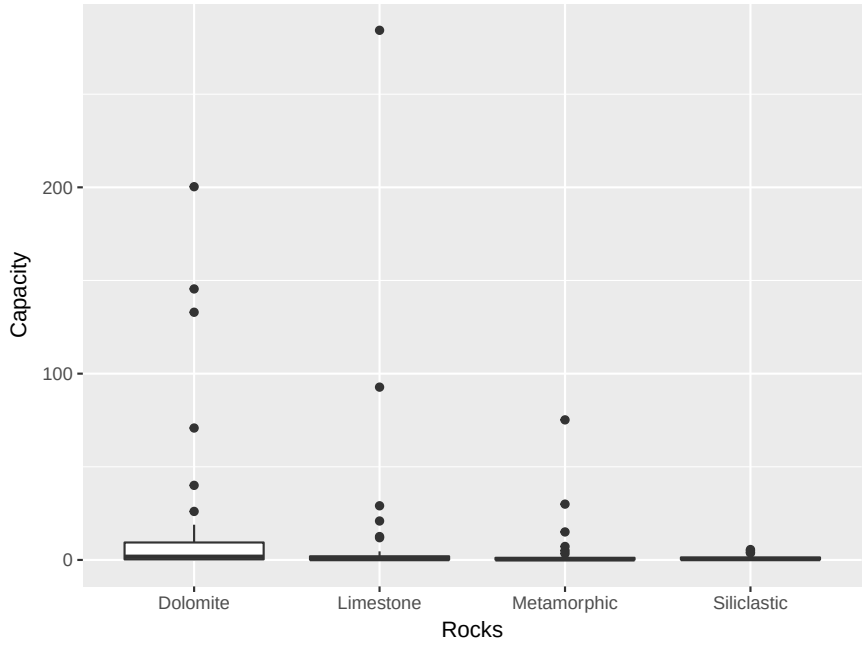
```
attach(rocks)
head(rocks)
```

```
##      Rocks     Capacity
## 1 Dolomite 132.95355630
## 2 Dolomite   0.03995506
## 3 Dolomite   3.09565649
## 4 Dolomite   9.97418198
## 5 Dolomite   4.61817669
## 6 Dolomite   1.50681778
```

```
table(Rocks)
```

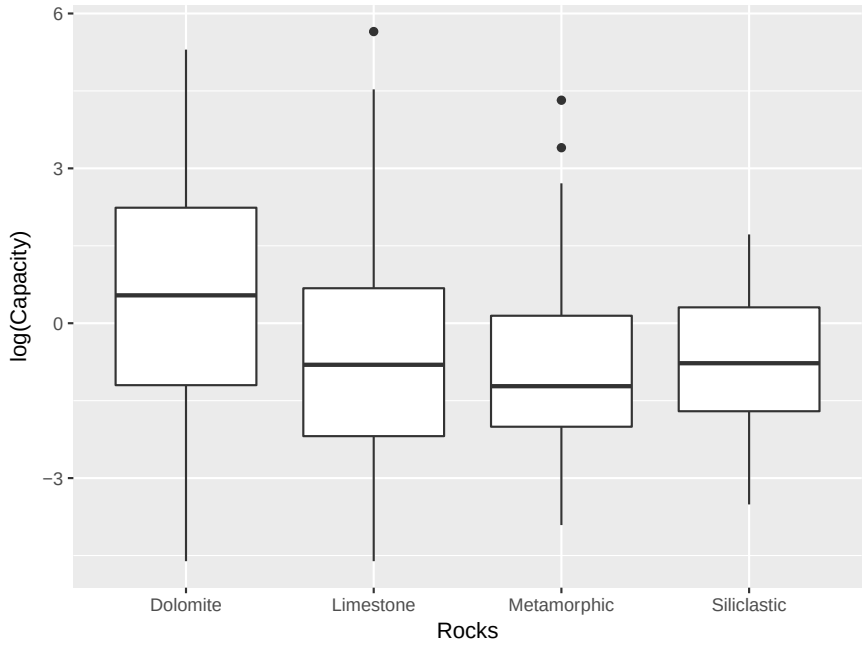
```
## Rocks
##    Dolomite   Limestone Metamorphic Siliclastic
##          50          50          50          50
```

```
bplot(Capacity, Rocks)
```



Clearly there some serious outliers. Let's try the log transform:

```
bplot(log(Capacity), Rocks)
```



and this looks much better.

Summary Statistics

Because we used a transformation we will use the median and IQR

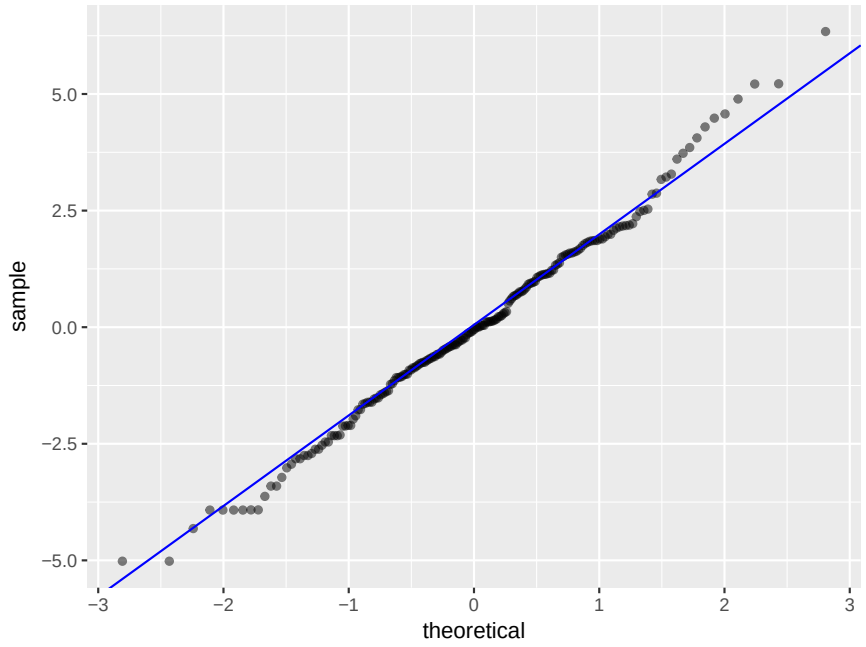
```
stat.table(Capacity, Rocks, Mean=FALSE)
```

```
##             Sample Size Median IQR
## Dolomite             50    1.7 9.1
## Limestone            50    0.5 1.9
## Siliclastic          50    0.5 1.2
## Metamorphic          50    0.3 1.0
```

Note that the estimates of the variation differ by quite a lot (1.0 vs 9.1). This again is due to the fact that we have many outliers in the dataset.

Now the test:

```
oneway(log(Capacity), Rocks)
```




```
## p value of test of equal means: p = 0.0067
## Smallest sd:  1.4    Largest sd : 2.6
```

1. Parameters of interest: group means
2. Method of analysis: ANOVA
3. Assumptions of Method: residuals have a normal distribution, groups have equal variance
4. $\alpha = 0.05$
5. H_0: $\mu_1 = \mu_2 = \mu_3 = \mu_4$ (no difference in the mean Capacity for different Rocks)
6. H_a: $\mu_i \neq \mu_j$ for some i and j (some differences in the mean Capacitys for different Rocks)
7. p-value = 0.0067
8. p $< \alpha$, we reject H_0, there are some differences in the mean Capacity for different Rocks.

Assumptions: Normal plot looks ok

smallest stdev of log(Capacity): 0.61, largest stdev: 1.11, 3*0.61 = 18.3 > 1.11, ok

Warning

If we had not done a transformation the results would have been quite different. For example, rocks would not have been stat. significant (p-value = 0.06)

20.2 Quantitative - Quantitative

20.2.1 Case Study: Brain and Body Weight of 62 Mammals

Brain and Body Weight (in kg) of 62 Mammals.

```
head(brainsize)
```

```
##                       Animal body.wt.kg brain.wt.g
## 1           African elephant   6654.000     5712.0
## 2 African giant pouched rat      1.000        6.6
## 3                 Arctic Fox      3.385       44.5
## 4    Arctic ground squirrel       0.920        5.7
## 5             Asian elephant   2547.000     4603.0
## 6                     Baboon     10.550      179.5
```

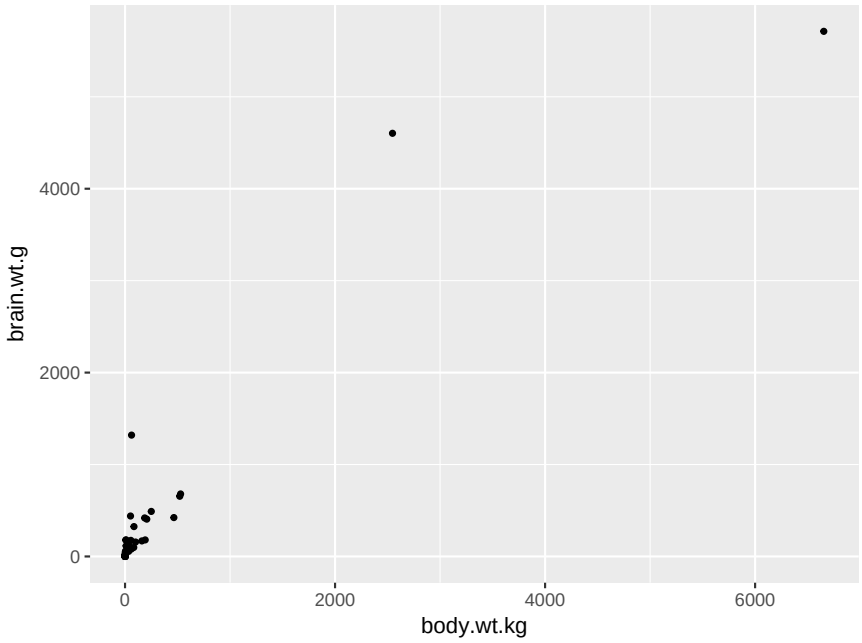
We have two quantitative variables, so we should start with the scatterplot:

```
attach(brainsize)
brainsize
```

```
##                       Animal body.wt.kg brain.wt.g
## 1           African elephant   6654.000    5712.00
## 2  African giant pouched rat      1.000       6.60
## 3                 Arctic Fox      3.385      44.50
## 4     Arctic ground squirrel       0.920       5.70
## 5             Asian elephant   2547.000    4603.00
## 6                     Baboon     10.550     179.50
## 7               Big brown bat      0.023       0.30
## 8             Brazilian tapir    160.000     169.00
## 9                         Cat      3.300      25.60
## 10                Chimpanzee     52.160     440.00
## 11                Chinchilla      0.425       6.40
## 12                       Cow    465.000     423.00
## 13           Desert hedgehog      0.550       2.40
## 14                    Donkey    187.100     419.00
## 15     Eastern American mole      0.075       1.20
## 16                   Echidna      3.000      25.00
## 17         European hedgehog      0.785       3.50
## 18                    Galago      0.200       5.00
## 19                     Genet      1.410      17.50
## 20           Giant armadillo     60.000      81.00
## 21                   Giraffe    529.000     680.00
## 22                      Goat     27.660     115.00
## 23            Golden hamster      0.120       1.00
## 24                   Gorilla    207.000     406.00
## 25                 Gray seal     85.000     325.00
## 26                 Gray wolf     36.330     119.50
## 27           Ground squirrel      0.101       4.00
```

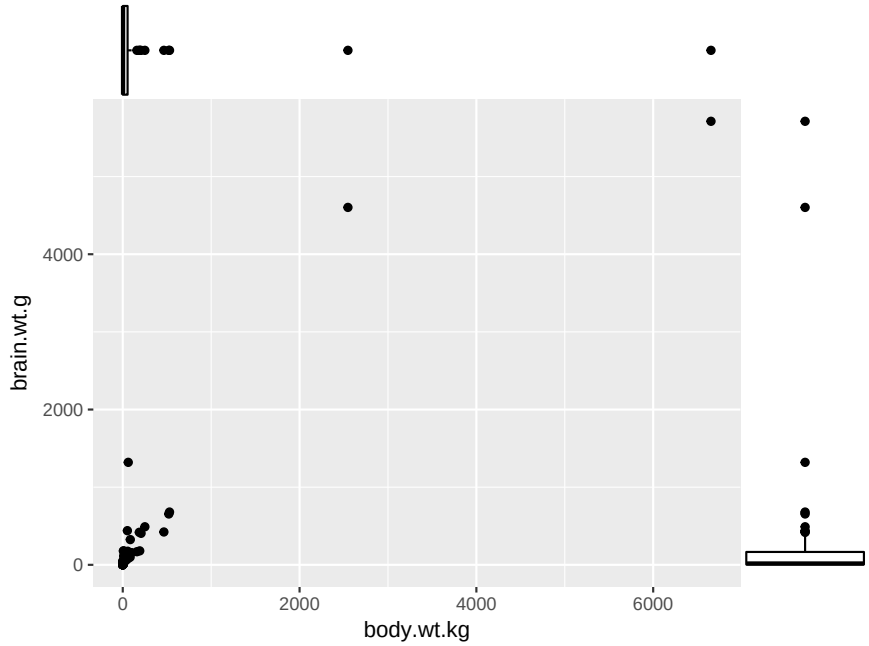
```
## 28                Guinea pig      1.040      5.50
## 29                     Horse    521.000    655.00
## 30                    Jaguar    100.000    157.00
## 31                  Kangaroo     35.000     56.00
## 32 Lesser short-tailed shrew      0.005      0.14
## 33          Little brown bat      0.010      0.25
## 34                       Man     62.000   1320.00
## 35                  Mole rat      0.122      3.00
## 36           Mountain beaver      1.350      8.10
## 37                     Mouse      0.023      0.40
## 38                 Musk shrew      0.048      0.33
## 39       N. American opossum      1.700      6.30
## 40     Nine-banded armadillo      3.500     10.80
## 41                     Okapi    250.000    490.00
## 42                Owl monkey      0.480     15.50
## 43              Patas monkey     10.000    115.00
## 44                Phanlanger      1.620     11.40
## 45                       Pig    192.000    180.00
## 46                    Rabbit      2.500     12.10
## 47                   Raccoon      4.288     39.20
## 48                       Rat      0.280      1.90
## 49                   Red fox      4.235     50.40
## 50             Rhesus monkey      6.800    179.00
## 51    Rock hyrax (Hetero. b)      0.750     12.30
## 52 Rock hyrax (Procavia hab)      3.600     21.00
## 53                   Roe deer     83.000     98.20
## 54                      Sheep     55.500    175.00
## 55                Slow loris      1.400     12.50
## 56           Star nosed mole      0.060      1.00
## 57                     Tenrec      0.900      2.60
## 58                Tree hyrax      2.000     12.30
## 59                Tree shrew      0.104      2.50
## 60                     Vervet      4.190     58.00
## 61             Water opossum      3.500      3.90
## 62     Yellow-bellied marmot      4.050     17.00
```

```
splot(brain.wt.g, body.wt.kg)
```



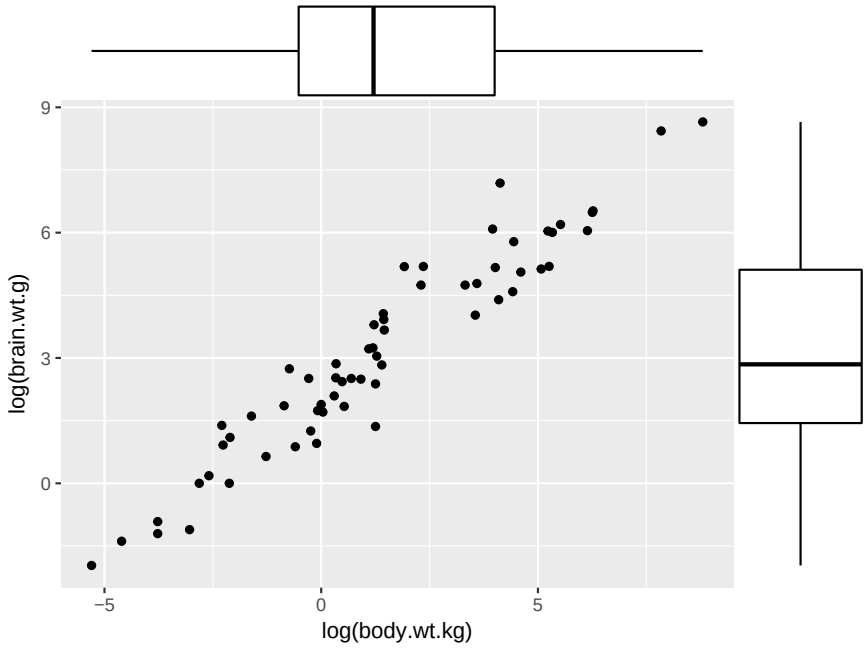
unfortunately almost all the "space" in the graph is taken up by a few outliers, it is not even possible to determine if there is a relationship between the variables. Drawing the marginal plot show that the problem are outliers in both variables:

```
mplot(brain.wt.g, body.wt.kg)
```



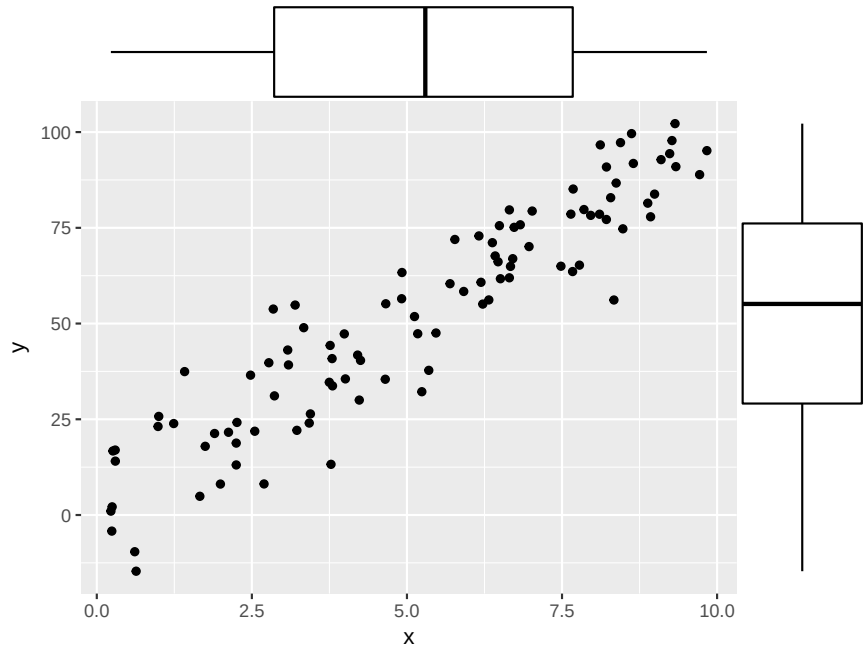
As before we can try and fix this problem by using a **log transformation**:

```
mplot(log(brain.wt.g), log(body.wt.kg))
```



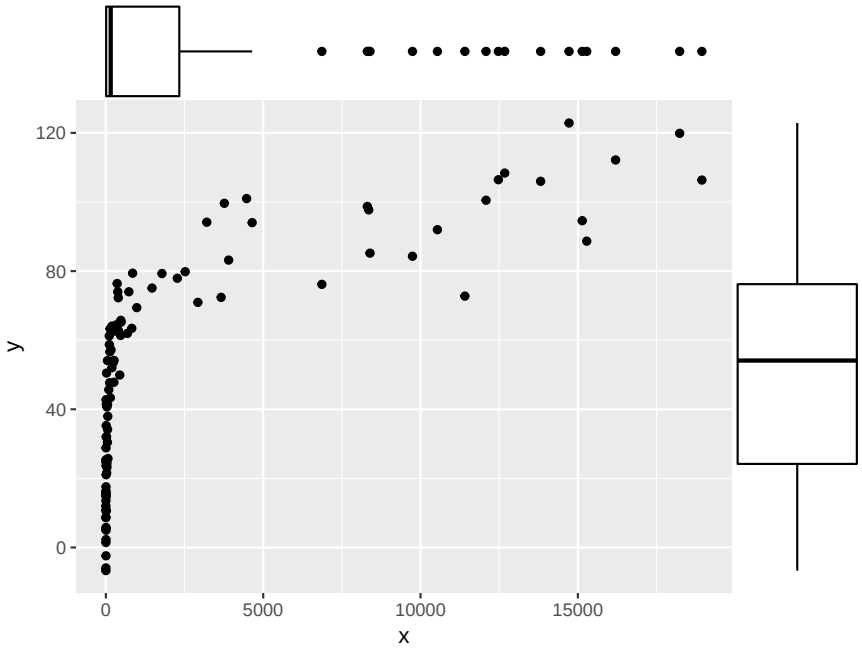
which nicely fixes the problem.

Because now we have two quantitative variables the log transform could be applied to x, to y or to both. In general we might see any of these combinations:



→ no transformations needed

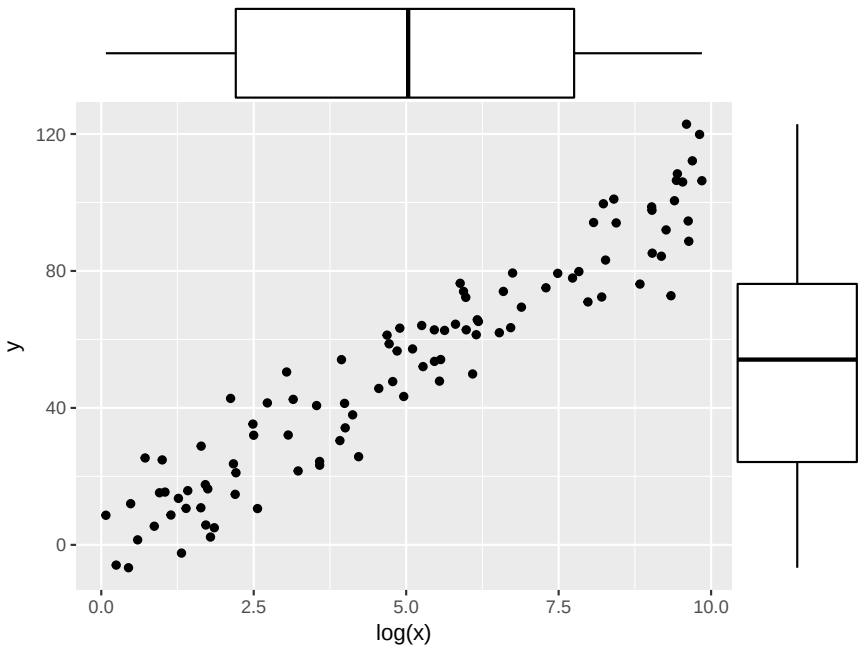
```
mplot(y, x)
```



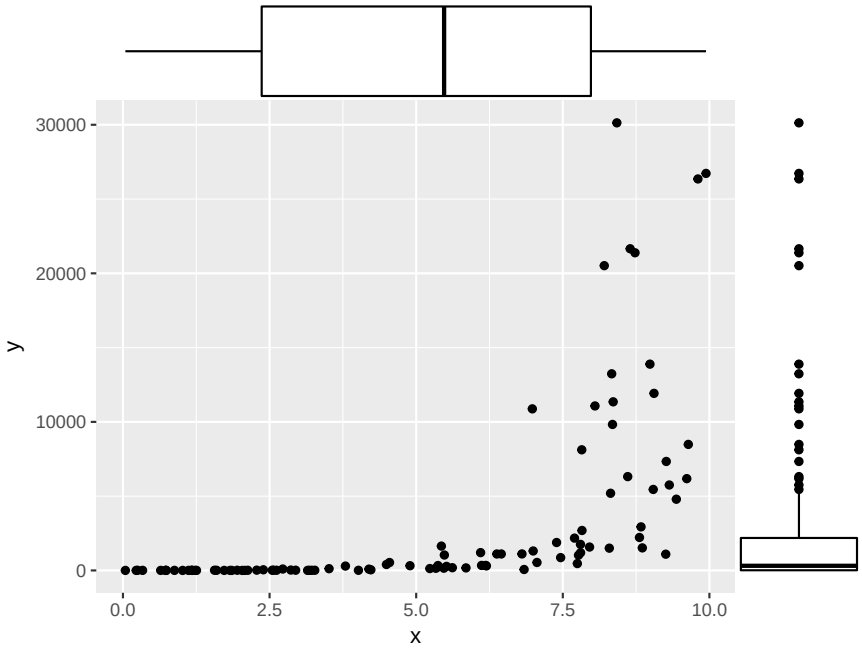
x variable is bad, y variable is ok

→ log transform x, leave y alone:

```
mplot(y, log(x))
```



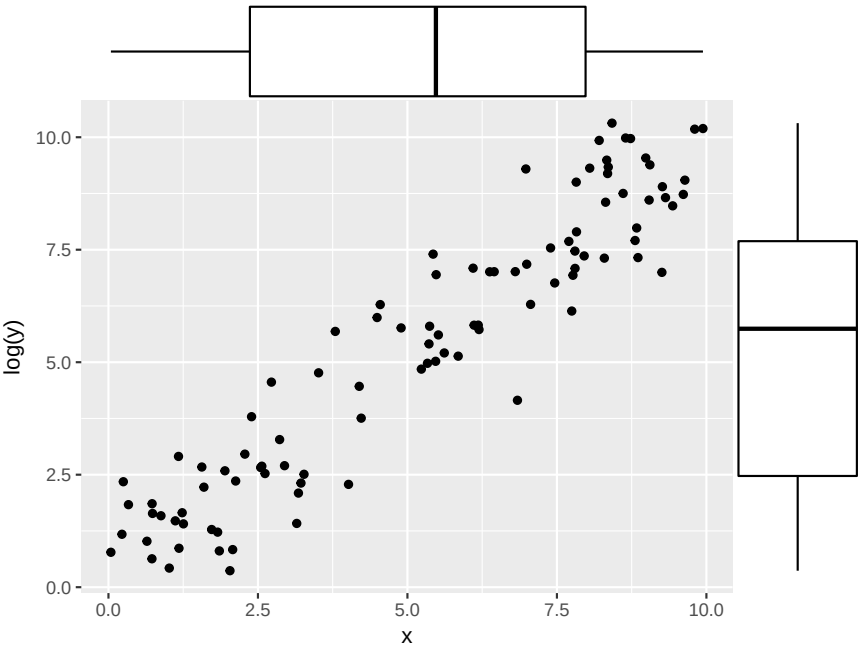
```
mplot(y, x)
```



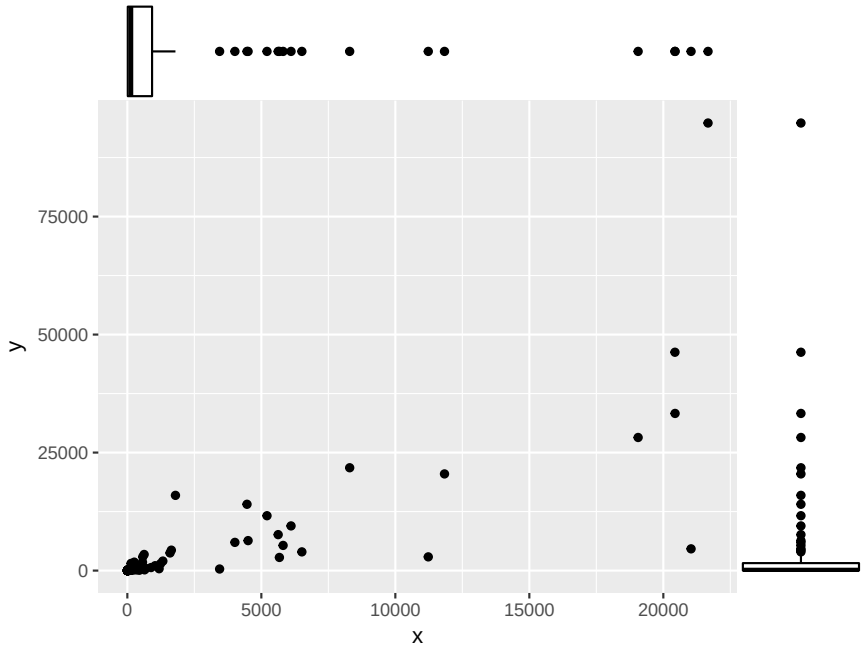
y variable is bad, x variable is ok

→ log transform y, leave x alone:

```
mplot(log(y), x)
```



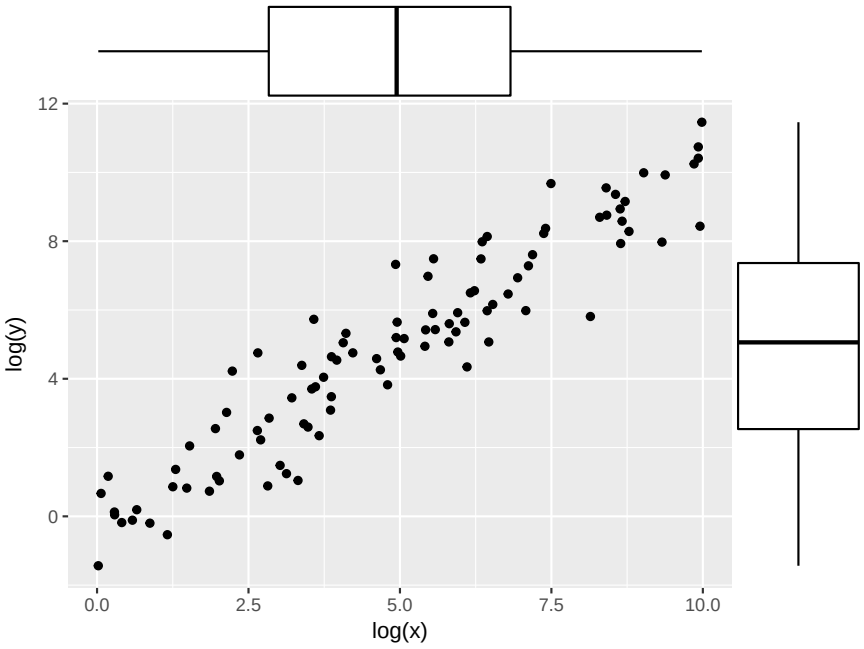
```
mplot(y, x)
```



both x and y variables are bad

→ log transform x and y

```
mplot(log(y), log(x))
```



It is clear from the scatterplot that we have a strong linear relationship between log(Brain) and log(Body), but if we want to we can now also find Pearson's correlation coefficient:

```
cor(log(body.wt.kg), log(brain.wt.g))
```

```
## [1] 0.958817
```

Doing so for the original data would have been wrong!

21 Non-Normal Residuals, No Equal Variance - Non-Parametric Methods

21.1 One Quantitative Variable (Alternative to 1 Sample t)

21.1.1 Case Study: Euro Coins

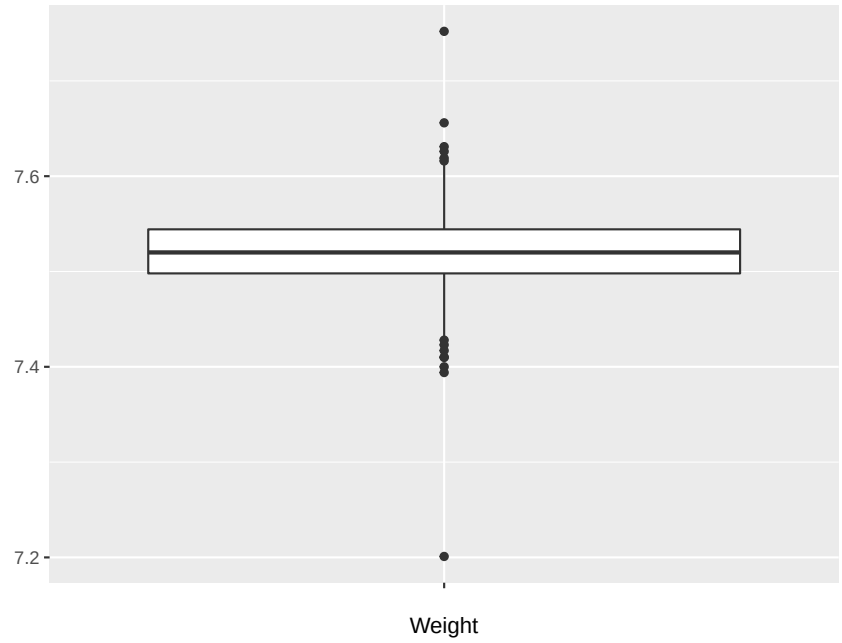
Data set: **euros**
say we are told that a one euro coin is supposed to weigh 7.5 grams. Does the data in support that claim?

The boxplot of Weight shows severe outliers, so the usual 1 sample t test won't work. Unfortunately the log transfromations does not work here either:

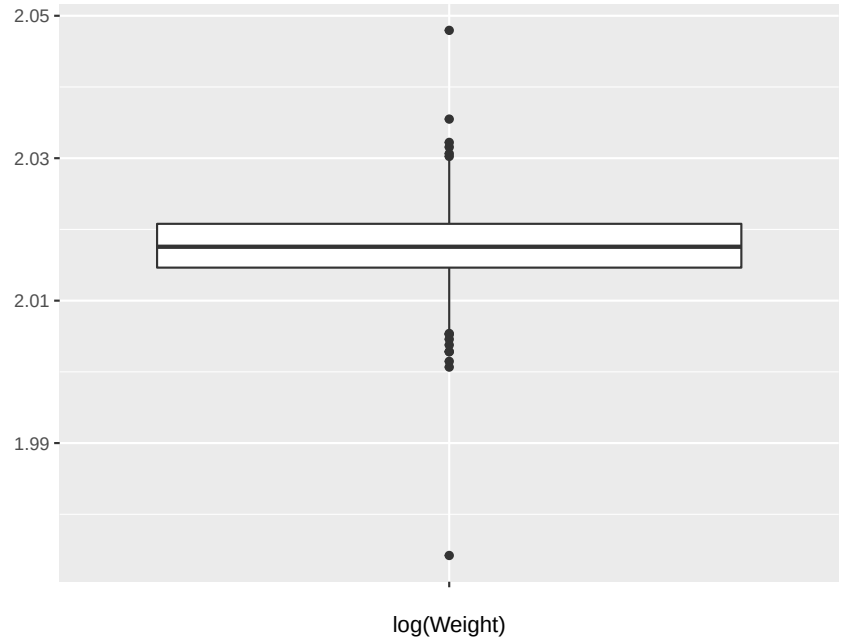
```
attach(euros)
head(euros)
```

```
##   Weight Roll
## 1  7.512    1
## 2  7.502    1
## 3  7.461    1
## 4  7.562    1
## 5  7.528    1
## 6  7.459    1
```

```
bplot(Weight)
```




```
bplot(log(Weight))
```



This is not a surprise, by the way, because the outliers are on both sides of the box.

So, what now? For this situation we have a set of methods called **non-parametric**, which make no assumptions, especially not the one of the 1 sample t test, namely normal distribution. The name of the test that works here is **Wilcoxon Signed Rank Test**.

The details are

```
one.sample.wilcoxon(Weight, med.null=7.5)
```

```
## p value of test H0: median=7.5 vs. Ha: median <> 7.5:  0.000
```

1) Parameter of interest: 1 median

2) Method of analysis: Wilcoxon Signed Rank test

3) Assumptions of Method: **none**

4) $\alpha = 0.05$

5) Null hypothesis H_0: M=7.5 (median weight is 7.5 grams)

6) Alternative hypothesis H_a: M $\neq$ 7.5 (median weight is not 7.5 grams)

7) p value = 0.000

8) 0.000<0.05, so we reject H_0, it seems the median weight is not 7.5 grams.

Actually, in this data set we could still have used the usual 1-sample t test (also with a p-value of 0.000) because we have a very large sample (n=2000), but in general it is never clear exactly how large a sample needs to be to "overcome" some outliers, so these non-parametric tests are always a safe alternative.

21.1.1.1 Why not always use the non-parametric test?

If using the t test sometimes is wrong but the Wilcoxon Rank Sum test always works, why not just always use this test and be safe? The answer is that the t test has a larger power:

In real life the power of the nonparametric tests is often almost as high as the power of the standard tests, so they should always be used if there is a question about the normal assumption.

The arguments of the *one.sample.wilcoxon* routine are the same as those of the *one.sample.t* command. For example if we wanted to test

5) Null hypothesis H_0: M=7.5 (median weight is 7.5 grams)

6) Alternative hypothesis H_a: M>7.5 (median weight is higher than 7.5 grams)

we could run

```
one.sample.wilcoxon(Weight, med.null=7.5, alternative = "greater")
```

```
## p value of test H0: median=7.5 vs. Ha: median > 7.5:  0.000
```

If we wanted a 90% confidence interval for median we could use

```
one.sample.wilcoxon(Weight, conf.level=90, ndigit=4)
```

```
## A 90% confidence interval for the population
##                   median is (7.5195, 7.5224)
```

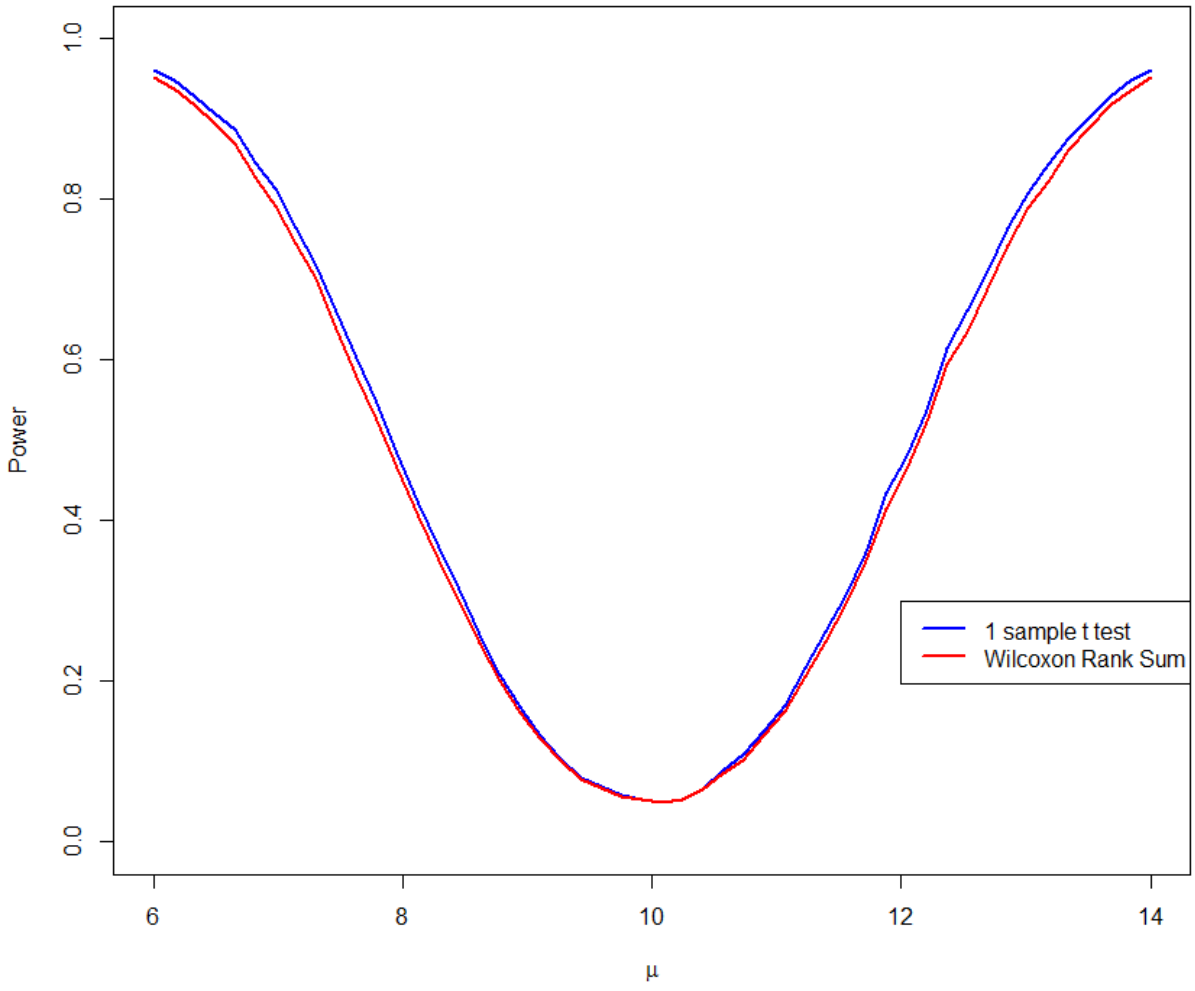

Power
0.0
0.2
0.4
0.6
0.8
1.0
6
8
10
12
14
μ
1 sample t test
Wilcoxon Rank Sum

Figure 24:

21.2 Categorical - Quantitative (Alternative to ANOVA)

21.2.1 Case Study: Euro Coins

Say we want to know whether the coin in the 8 different rolls have the same average weight. The non-parametric alternative to the oneway ANOVA is the **Kruskal-Wallis test:**

```
kruskalwallis(Weight, Roll)
```

```
## p value of test of equal means: p = 0.000
```

1) Parameters of interest: **medians**

2) Method of analysis: Kruskal-Wallis

3) Assumptions of Method: **none**

4) $\alpha = 0.05$

5) Null hypothesis H_0: M_1=..=M_8 (group **medians** are the same)

6) Alternative hypothesis H_a: $M_i \neq M_j$ for some i, j(group medians are not the same)

7) p value = 0.00

8) $0.00 < 0.05$, so we reject H_0, it seems the group medians are not the same

21.2.2 Case Study: Cultural Differences in Equipment Use

A US company manufactures equipment that is used in the production of semiconductors. The firm is considering a costly redesign that will improve the performance of its equipment. The performance is characterized as mean time between failures (MTBF). Most of the companies customers are in the USA, Europe and Japan, and there is anectotal evidence that the Japanese customers typically get better performance from the users in the USA and Europe.

Data: MTBF for randomly selected users in the USA, Europe and Japan.

Data set: **culture**

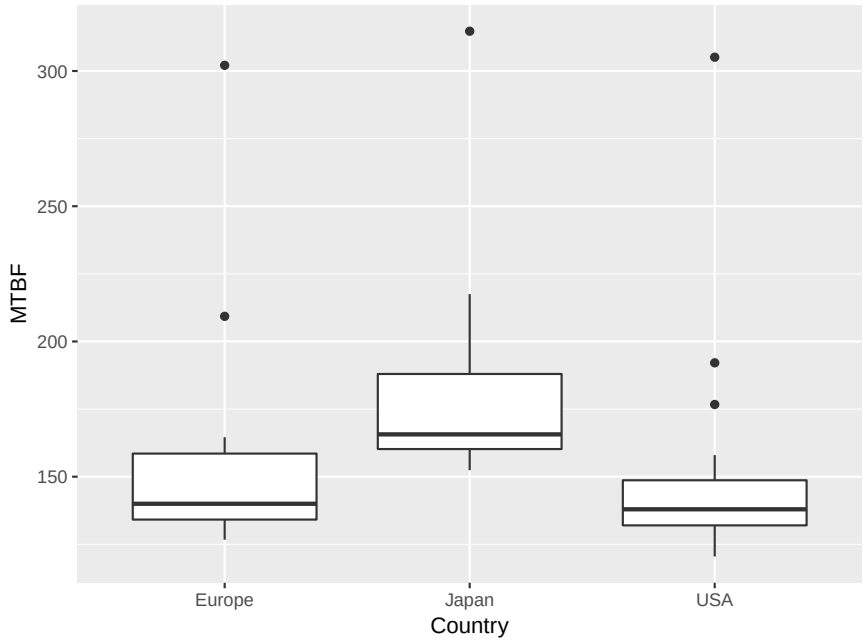
```
attach(culture)
head(culture)
```

```
##   Country  MTBF
## 1     USA 120.5
## 2     USA 127.1
## 3     USA 128.1
## 4     USA 129.7
## 5     USA 130.8
## 6     USA 132.4
```

```
table(Country)
```

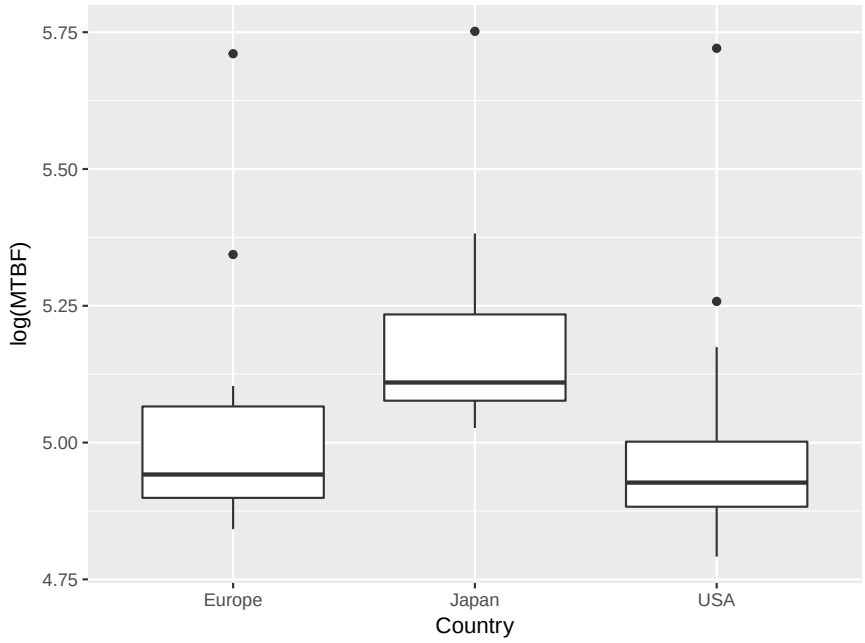
```
## Country
## Europe  Japan    USA
##     15     12     20
```

```
bplot(MTBF, Country)
```



There is a problem with the normal assumption. We can try to fix this with the log transform, but as we can see again this does not work:

```
bplot(log(MTBF), Country)
```



Summary Statistics

We could not even find a transformation, so we will use the median and IQR:

```
stat.table(MTBF, Country, Mean=FALSE)
```

```
##         Sample Size Median  IQR
## USA              20  138.0 16.7
## Europe           15  140.0 24.4
## Japan            12  165.7 27.8
```

Because none of the transformations worked we will use the non-parametric Kruskall-Wallis test:

```
kruskalwallis(MTBF, Country)
```

```
## p value of test of equal means: p = 0.00100488597092579
```

1) Parameters of interest: medians

2) Method of analysis: Kruskal-Wallis

3) Assumptions of Method: **none**

4) α0.05

5) Null hypothesis H_0: $M_1 = M_2 = M_3$ (group medians are the same)

6) Alternative hypothesis H_a: $M_i \neq M_j$ for some i, j (group medians are not the same)

7) p value = 0.001

8) $0.001 < 0.05$, so we reject H_0, it seems the group medians are not the same, the MTBF is different in diffferent countries

If we had just done the ANOVA Country would not have been stat. significant (p-value = 0.098) but if you remember to check the normal plot you will see that there is a problem with this analysis.

21.3 Quantitative - Quantitative

If the transformations fail in a regression problem things become very tricky, and far beyond the scope of this class. Talk to a professional!

22 Exercise Problems 2

22.0.1 Case Study: Survey of Students

This is a fake data set I made up for this exercise. It is supposed to be from a survey of students at some College. The data is in **studentsurvey**

The variables are

```
colnames(studentsurvey)
```

```
## [1] "Score"    "Gender"   "Year"     "GPA"      "Distance" "Major"
## [7] "Age"
```

Score is a combination of several questions designed to measure how "happy" they are to study at the College. A high number means more happiness. Distance is how far they live from the College. Make sure your answers are **complete**.

Problem 1 Is there a relationship between Score and Gender?

Problem 2 Is there a relationship between Score and Year?

Problem 3 Is there a relationship between Score and GPA?

Problem 4 Is there a relationship between Score and Distance?

Problem 5 Is there a relationship between Score and Age?

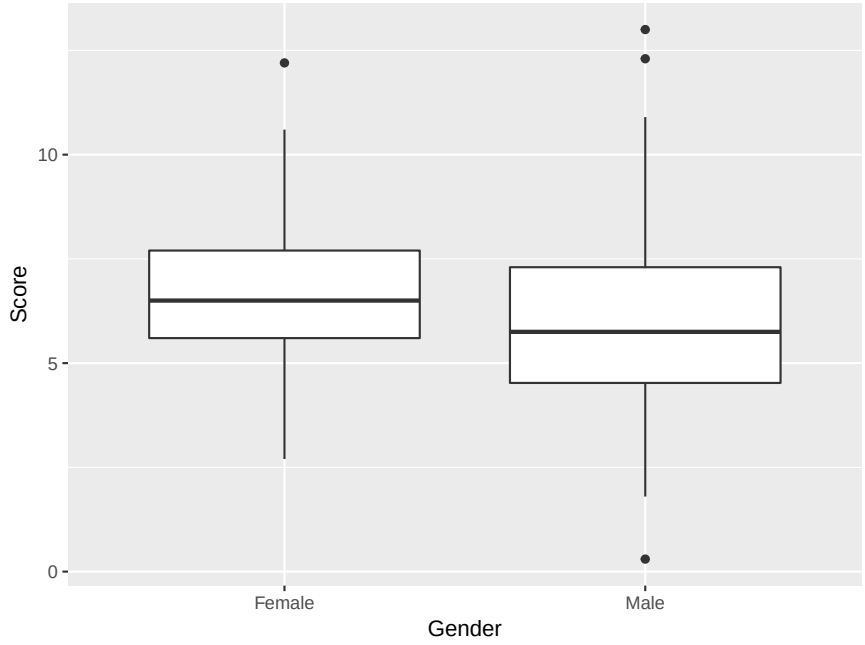
Problem 6 Is there a relationship between Gender and Major?

Problem 7 Is there a relationship between Gender and Age?

```
attach(studentsurvey)
```

Problem 1 Is there a relationship between Score and Gender? Score is a quantitative variable and Gender is a categorical variable with two values, so this is a problem for ANOVA

```
bplot(Score,Gender)
```



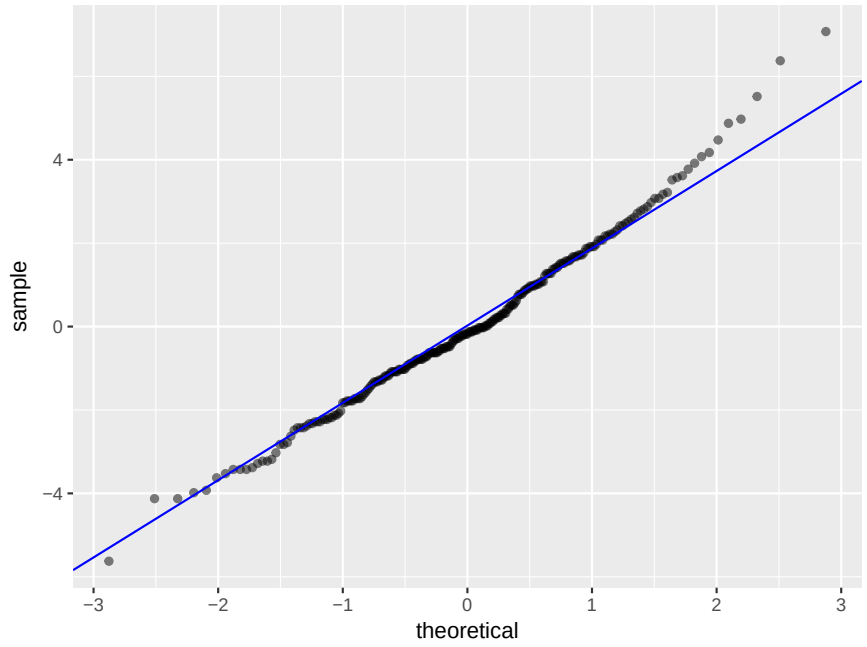
The boxplot shows a slight difference between the genders. There are a few slight outliers, but they are no problem.

```
stat.table(Score, Gender)
```

```
##        Sample Size Mean Standard Deviation
## Female         111  6.7                1.7
## Male           138  5.9                2.2
```

The test:

```
oneway(Score, Gender)
```




```
## p value of test of equal means: p = 0.0033
## Smallest sd:  1.7    Largest sd : 2.2
## A 95% confidence interval for the difference in group means is (0.3, 1.2)
```

1) Parameters of interest: means of scores of men and women

2) Method of analysis: two sample t test

3) Assumptions of Method: residuals have a normal distribution, or sample sizes are large enough

4) $\alpha = 0.05$

5) Null hypothesis H_0: $\mu_1 = \mu_2$ (groups have the same mean)

6) Alternative hypothesis H_a: $\mu_1 \neq \mu_2$ (groups have different means)

7) p value = 0.0033

8) $0.0033 < 0.05$, there is some evidence that the group means are not the same, the women tend to score higher than the men.

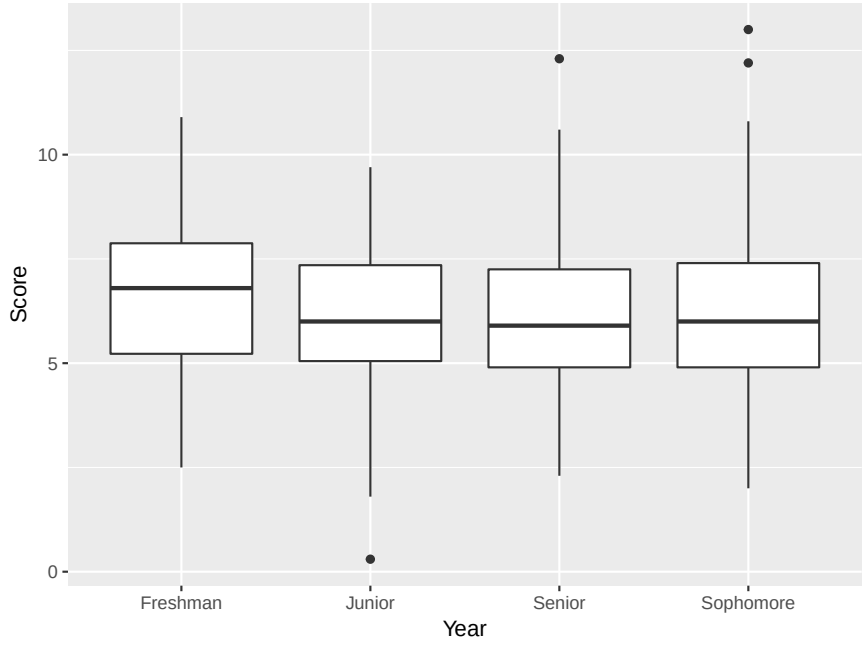
Assumptions:

1) Normal residuals: normal plot looks ok.

2) equal variance: 3*1.71 = 5.13 > 2.21, ok

Problem 2 Is there a relationship between Score and Year? Score is a quantitative variable and Year is a categorical variable, so this is a problem for ANOVA

```
bplot(Score, Year)
```



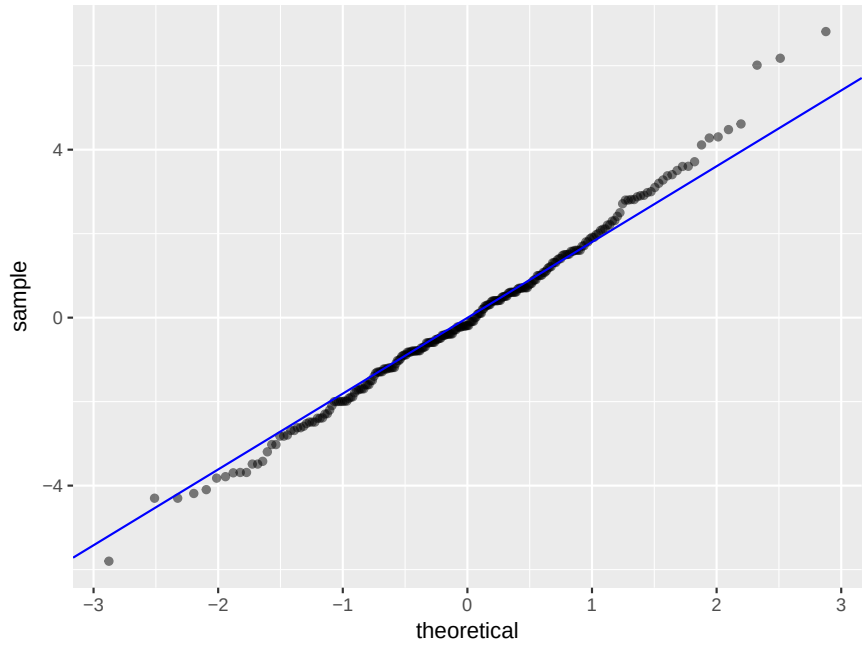
The table of summary statistics is

```
stat.table(Score, Year)
```

```
##           Sample Size Mean Standard Deviation
## Junior             47  6.1                2.0
## Freshman           66  6.6                1.8
## Sophomore          77  6.2                2.2
## Senior             59  6.1                2.1
```

The test:

```
oneway(Score, Year)
```




```
## p value of test of equal means: p = 0.4855
## Smallest sd:  1.8    Largest sd : 2.2
```

1) Parameters of interest: means of scores by year

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, or sample sizes are large enough

4) $\alpha = 0.05$

5) Null hypothesis H_0: $\mu_1 = \mu_2 = \mu_3 = \mu_4$ (groups have the same mean)

6) Alternative hypothesis H_a: $\mu_i \neq \mu_j$ (some groups have different means)

7) p value = 0.4855

8) $0.4855 > 0.05$, there is no evidence that the group means are not the same.

Assumptions:

1) Normal residuals. looks ok.

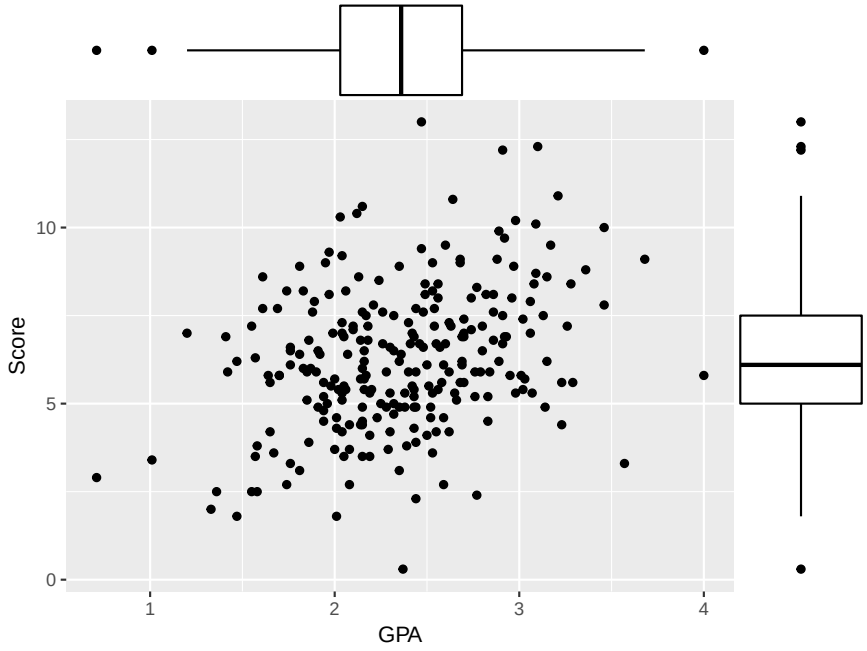
2) equal variance: 3*1.8 = 5.4 > 2.2

Problem 3 Is there a relationship between Score and GPA? Score and GPA are both quantitative variables, so this is a problem for the **Pearson's Correlation Coefficient** .

The marginal plot shows some increase in Score as the GPA increases.

```
rm(Score)
rm(GPA)
```

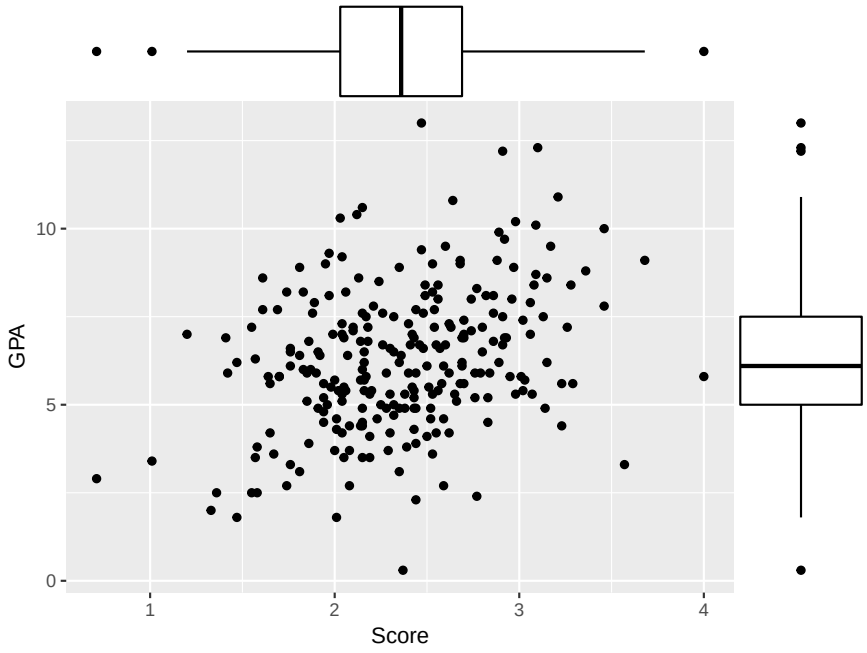
```
mplot(Score, GPA)
```



There are a few slight outlier, no problem, though.

The test:

```
pearson.cor(Score, GPA, rho.null = 0)
```



```
## p value of test H0: rho=0 vs. Ha: rho <> 0:  0.000
```

1) Parameter of interest: Pearson's correlation coefficient ρ

2) Method of analysis: test based on normal theory

3) Assumptions of Method: relationship is linear, there are no outliers

4) $\alpha = 0.05$

5) H_0: $\rho = 0$ (no relationship between Score and GPA)

6) H_a: $\rho \neq 0$ (some relationship between Score and GPA)

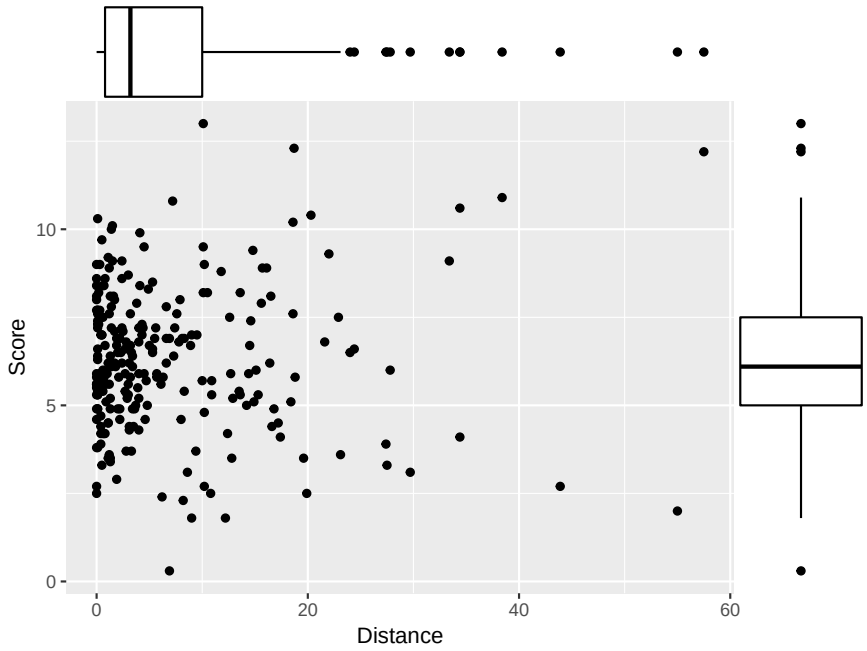
7) p = 0.000

8) $0 < 0.05$, so we reject H_0, there is a relationship between the Score and the GPA, apparently students with a higher GPA are happier.

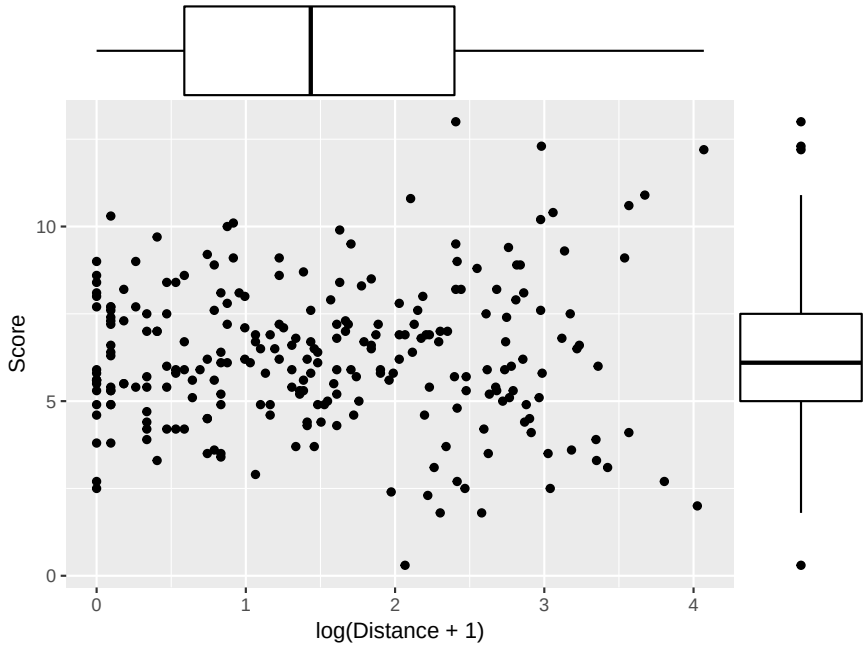
Problem 4 Is there a relationship between Score and Distance? Score and Distance are both quantitative variables, so this is a problem for the **Pearson's Correlation Coefficient**.

The marginal plot shows some slight outliers. The log transform fixes it.

```
mplot(Score, Distance)
```



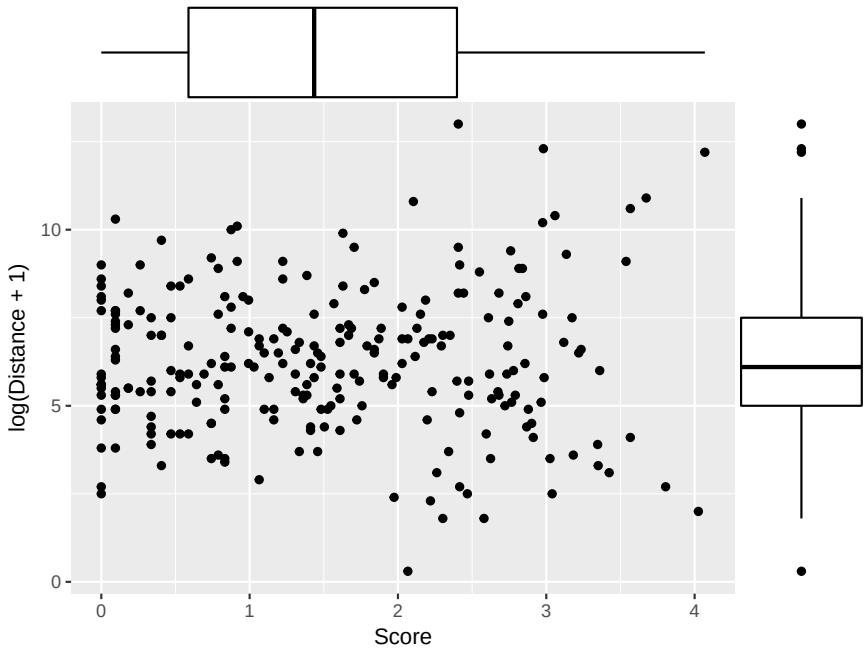

```
mplot(Score, log(Distance+1))
```



some students apparently live very close to the school, Distance=0. For this reason we use log(Distance+1).

The test:

```
pearson.cor(Score, log(Distance+1), rho.null = 0)
```




```
## p value of test H0: rho=0 vs. Ha: rho <> 0:  0.7305
```

1) Parameter of interest: Pearson's correlation coefficient ρ

2) Method of analysis: test based on normal theory

3) Assumptions of Method: relationship is linear, there are no outliers

4) $\alpha = 0.05$

5) H_0: $\rho = 0$ (no relationship between Score and Distance)

6) H_a: $\rho \neq 0$ (some relationship between Score and Distance)
7) p = 0.7305

8) $p > \alpha$, so we fail to reject H_0, there is no evidence of a relationship between the Score and the Distance.

Problem 5 Is there a relationship between Score and Age? The marginal plot shows one severe outlier:

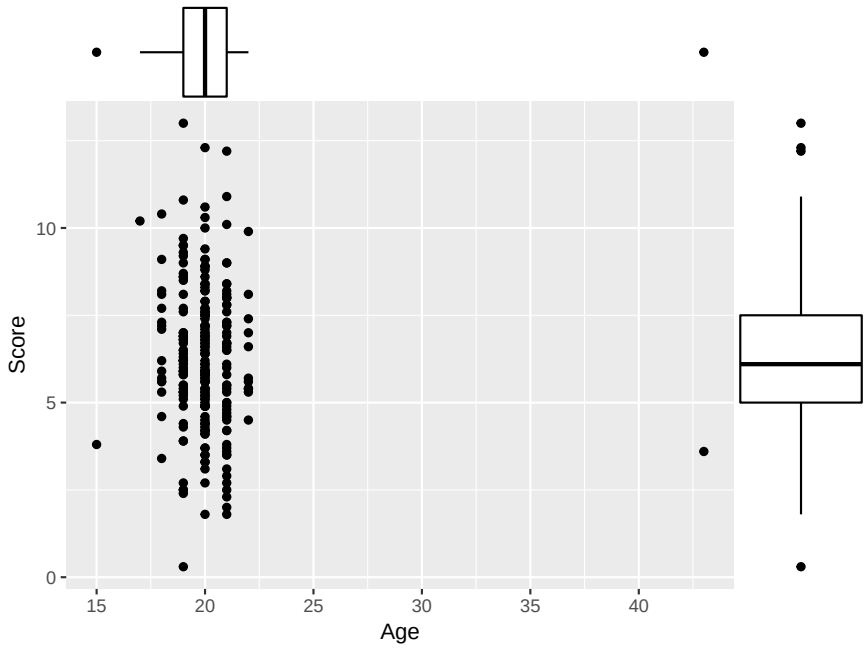
```
length(Score)
```

```
## [1] 249
```

```
length(Age)
```

```
## [1] 249
```

```
mplot(Score, Age)
```



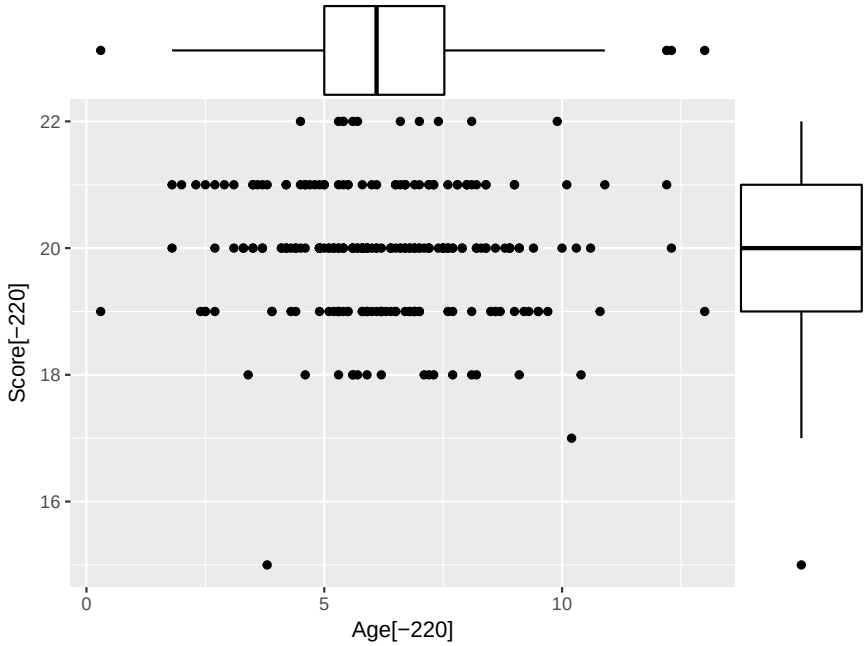
Unfortunately the log transform does not help, so the only way to preceed is to eliminate the outlier.

```
which(Age==max(Age))
```

```
## [1] 220
```

The test:

```
pearson.cor(Age[-220], Score[-220], rho.null = 0)
```




```
## p value of test H0: rho=0 vs. Ha: rho <> 0:  0.3052
```

1) Parameter of interest: Pearson's correlation coefficient ρ

2) Method of analysis: test based on normal theory

3) Assumptions of Method: relationship is linear, there are no outliers

4) $\alpha = 0.05$

5) H_0: $\rho = 0$ (no relationship between Score and Distance)

6) H_a: $\rho \neq 0$ (some relationship between Score and Distance)

7) p = 0.3052

8) $p > \alpha$, so we fail to reject H_0, there is no evidence of a relationship between the Score and the Age.

Problem 6 Is there a relationship between Gender and Major? Gender and Major are both categorical variables, so this is a problem for the **Chisquare test of Independence**.

```
table(Gender, Major)
```

```
##         Major
## Gender   Biology English Physics Psychology Spanish
##   Female      19      23      23          24      22
##   Male        24      24      31          29      30
```

```
chi.ind.test(table(Gender, Major))
```

```
## p value of test p=0.9664
```

1) Parameters of interest: measure of association

2) Method of analysis: chi-square test of independence

3) Assumptions of Method: all expected counts greater than 5

4) $\alpha = 0.05$
5) H_0: Classifications are independent = Gender and Major are independent

6) H_a: Classifications are dependent = Gender and Major are not independent

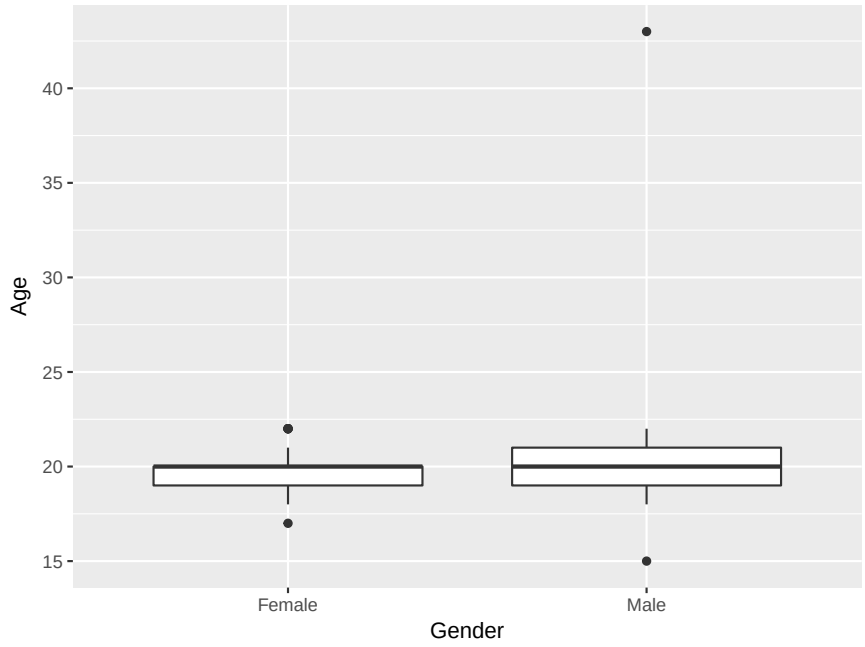
7) $p = 0.9664$

8) $0.9664 > 0.05$, there is no evidence of a relationship between gender and major.

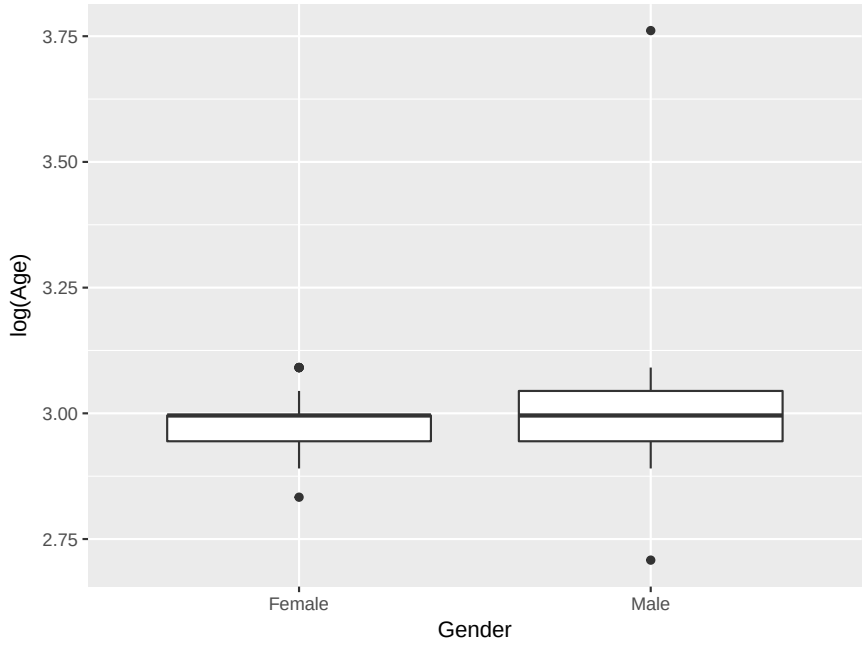
Problem 7 Is there a relationship between Gender and Age? Age is a quantitative variable and Gender is a categorical variable with two values, so this is a problem for the **ANOVA**.

The boxplot shows a few serious outliers. One could try transformations, but because the outliers are from especially small and large observations these won't work.

```
bplot(Age, Gender)
```




```
bplot(log(Age), Gender)
```



Solution 1: non-parametric method

The table of summary statistics is

```
stat.table(Age, Gender, Mean=FALSE )
```

```
##        Sample Size Median IQR
## Female         111     20   1
## Male           138     20   2
```

Now

```
kruskalwallis(Age, Gender)
```

```
## p value of test of equal means: p = 0.261060279952181
```

1) Parameters of interest: 2 medians
2) Method of analysis: Kruskal-Wallis
3) Assumptions of Method: **none**
4) $\alpha = 0.05$
5) Null hypothesis H_0: M_1=M_2 (group medians are the same)
6) Alternative hypothesis H_a: $M_1 \neq M_2$ (group medians are not the same)
7) p value = 0.2611
8) $0.2611 > 0.05$, so we fail to reject H_0, it seems the group medians are the same.

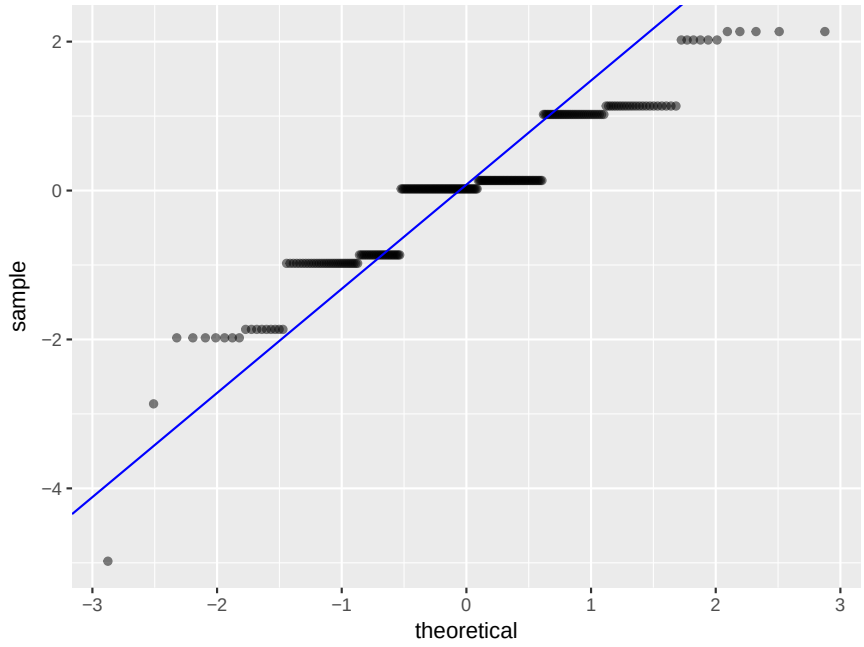
Solution 2: remove outlier

```
which(Age==max(Age))
```

```
stat.table(Age[-220], Gender[-220])
```

```
##        Sample Size Mean Standard Deviation
## Female         111 19.9                  1
## Male           137 20.0                  1
```

```
oneway(Age[-220], Gender[-220])
```




```
## p value of test of equal means: p = 0.3817
## Smallest sd:  1    Largest sd : 1
## A 95% confidence interval for the difference in group means is (-0.4, 0.1)
```

1) Parameters of interest: means of age by gender

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, or sample sizes are large enough
4) $\alpha = 0.05$

5) Null hypothesis H_0: $\mu_1 = \mu_2$ (groups have the same mean)

6) Alternative hypothesis H_a: $\mu_1 \neq \mu_2$ (some groups have different means)

7) p value = 0.3817
8) $0.3817 > 0.05$, there is no evidence that the group means are not the same. Assumptions:
9) Normal residuals ok Smallest sd: 1 Largest sd : 1 , 3*1>1, ok

22.1 Categorical Predictor - Categorical Response

In the case of two categorical variables knowing that they are somehow related is usually enough, beyond that one simply considers the percentages.

Figure 25:

22.2 Categorical Predictor - Quantitative Response

The case of a categorical predictor with 2 groups and a quantitative response is done - the two groups are different. The only other thing one might do is find a confidence interval for the differences in means, see 2-sample t method.

22.2.1 Case Study: Mothers Cocain Use and Babies Health

we have previously run the ANOVA and found that there are differences between the lengths of the babies of different groups. We can go a step further, though and ask the following questions:

- is there a difference between the Drug Free and the First Trimester group?
- is there a difference between the First Trimester and the Throughout group?

in other words, we can try to study the **pairwise differences**, which is an example of a **multiple comparison** study.

As we said before we could do this by running the 2 sample t test on each pair, but then we would be doing **simultaneous inference**. What we need is a method that does this but in such a way that the overall type I error probability is the desired α, no matter how many tests are done. R has a number of such methods implemented, we will use the one due to John Tukey, one of the founders of modern Statistics

```
attach(mothers)
tukey(Length, Status)
```

```
## Groups that are statistically significantly different:
```

```
##                  Groups p.value
## 1 Drug Free-Throughout       0
```

What does this tell us? To find out we first need to see the groups in the order of their means. We already know this here but in general a nice command to get that is

Figure 26:

```
stat.table(Length, Status, Sort=TRUE)
```

```
##                  Sample Size Mean Standard Deviation
## Throughout                36 48.0                3.6
## First Trimester           19 49.3                2.5
## Drug Free                 39 51.1                2.9
```

Now we are told that the only stat. significant difference is between Drug free and Throughout, so of course

- the difference between Drug Free and First Trimester is NOT stat. significant
- the difference between First Trimester and Throughout is NOT stat. significant

BUT: most importantly we need to remember the difference between *failing to reject* H_0 and *accept* H_0, so this does NOT say that there is no stat. significant difference between (say) Drug Free and First Trimester (why not?)

so now we have the following interpretation:

There is a stat. signif. difference between the mean lengths of the babies of Drug Free mothers and those who took cocain throughout the pregnancy. Other differences are not stat. signif., at least not at these sample sizes

Note It is theoretically possible that the oneway command find a statistically singificant difference, but Tukey does not, and vice versa! What you want to do is this: run the oneway command

- If it DOES NOT reject the null of some differences, DO NOTHING
- If is DOES reject the null, run tukey.

22.2.2 Case study: Cuckoo Eggs

That cuckoo eggs were peculiar to the locality where found was already known in 1892. A study by E.B. Chance in 1940 called The Truth About the Cuckoo demonstrated that cuckoos return year after year to the same territory and lay their eggs in the nests of a particular host species. Further, cuckoos appear to mate only within their territory. Therefore, geographical sub-species are developed, each with a dominant foster-parent species, and natural selection has ensured the survival of cuckoos most fitted to lay eggs that would be adopted by a particular foster-parent species. The data has the lengths of cuckoo eggs found in the nests of six other bird species (drawn from the work of O.M. Latter in 1902).

Basic question: is there a difference between the lengths of the cuckoo eggs of different Foster species?

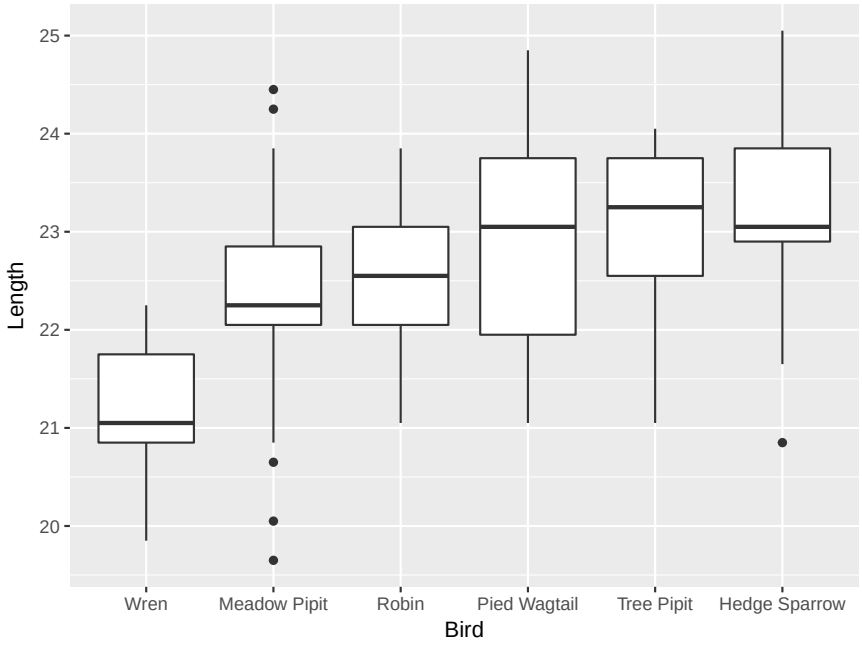
```
attach(cuckoo)
head(cuckoo)
```

```
##           Bird Length
## 1 Meadow Pipit  19.65
## 2 Meadow Pipit  20.05
## 3 Meadow Pipit  20.65
## 4 Meadow Pipit  20.85
## 5 Meadow Pipit  21.65
## 6 Meadow Pipit  21.65
```

```
table(Bird)
```

```
## Bird
## Hedge Sparrow  Meadow Pipit  Pied Wagtail         Robin    Tree Pipit
##            14            45            15            16            15
##          Wren
##            15
```

```
bplot(Length, Bird, new_order = "Size")
```



where we ordered the boxes by size because the categorical variable here has no obvious ordering.

we have some outliers in the Meadow Pipit species, but not to bad and we will ignore that.

Let's look at the table of summary statistics.

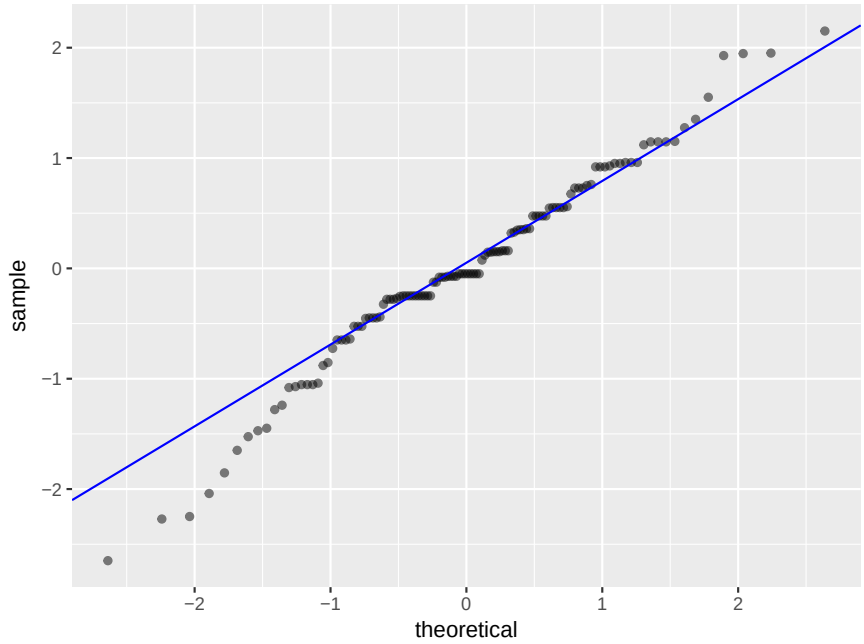
```
stat.table(Length, Bird, Sort=TRUE)
```

```
##                Sample Size Mean Standard Deviation
## Wren                    15 21.1                0.7
```

```
## Meadow Pipit          45 22.3              0.9
## Robin                 16 22.6              0.7
## Pied Wagtail          15 22.9              1.1
## Tree Pipit            15 23.1              0.9
## Hedge Sparrow         14 23.1              1.1
```

Both the graph and the table make it clear that there are some differences in the length, so the following is not really necessary:

```
oneway(Length, Bird)
```




```
## p value of test of equal means: p = 0.000
## Smallest sd:  0.7    Largest sd : 1.1
```

1) Parameters of interest: group means

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

4) $\alpha = 0.05$

5) Null hypothesis H_0: $\mu_1 = ... = \mu_6$ (groups have the same means)

6) Alternative hypothesis H_a: $\mu_i \neq \mu_j$ (at least two groups have different means)

7) p value = 0.000

8) $0.000 < 0.05$, there is some evidence that the group means are not the same, the length are different for different foster species.

Assumptions of the method:

a) residuals have a normal distribution, plot looks ok

b) groups have equal variance

smallest stdev=0.7, largest stdev=1.1, 3*0.7=2.1>1.1, ok

So, how exactly do they differ?

```
tukey(Length, Bird)
```

```
## Groups that are statistically significantly different:
```

```
##                         Groups p.value
## 1           Meadow Pipit-Wren  0.0000
## 2                  Robin-Wren  0.0000
## 3           Pied Wagtail-Wren  0.0000
## 4             Tree Pipit-Wren  0.0000
## 5          Hedge Sparrow-Wren  0.0000
## 6    Tree Pipit-Meadow Pipit  0.0475
## 7 Hedge Sparrow-Meadow Pipit  0.0429
```

so the eggs of Wrens are the smallest, and they are stat. significantly smaller than the eggs of all other birds.

Meadow Pipits are next, and they are stat. significantly smaller than the eggs of Tree Pipits and Hedge Sparrows.

no other differences are stat. significant!

On occasion one might want to see the p values of all the pairwise comparisons, for example if one wants to use an α different from 0.05:

```
tukey(Length, Bird, show.all = TRUE)
```

```
##                          Groups p.value
## 1            Meadow Pipit-Wren  0.0000
## 2                   Robin-Wren  0.0000
## 3            Pied Wagtail-Wren  0.0000
## 4              Tree Pipit-Wren  0.0000
## 5           Hedge Sparrow-Wren  0.0000
## 6           Robin-Meadow Pipit  0.9022
## 7   Pied Wagtail-Meadow Pipit  0.2325
## 8     Tree Pipit-Meadow Pipit  0.0475
## 9  Hedge Sparrow-Meadow Pipit  0.0429
## 10          Pied Wagtail-Robin  0.9155
## 11            Tree Pipit-Robin  0.6160
## 12         Hedge Sparrow-Robin  0.5726
## 13    Tree Pipit-Pied Wagtail  0.9932
## 14 Hedge Sparrow-Pied Wagtail  0.9872
## 15   Hedge Sparrow-Tree Pipit  1.0000
```

Notice that the pairs in tukey are also in the order from smallest to largest: first comes Meadow Pipit - Wren, the two birds with the smallest mean lengths.

23 Assumptions of Least Squares Regression

This page explains the assumptions behind the method of least squares regression and how to check them.

Recall that we are fitting a model of the form

$$y = \beta_0 + \beta_1 x$$

there are **three assumptions**:

1) The model is good (that is, the relationship is linear and not, say, quadratic, exponential or something else)
2) The residuals have a normal distribution
3) The residuals have equal variance (are homoscadastic)

The second and third assumption we are already familiar with from ANOVA and correlation.

We can check these assumptions using two graphs:

- Residual vs. Fits plot: this is just what it says, a scatterplot of the residuals (on y-axis) vs. the fitted values.
- Normal plot of residuals

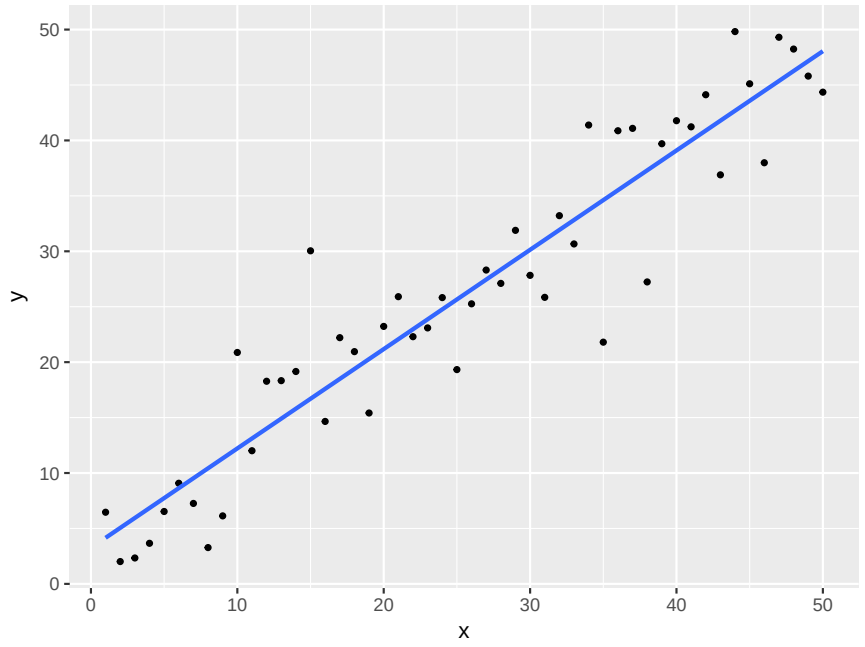
Both of these graphs are done by R automatically.

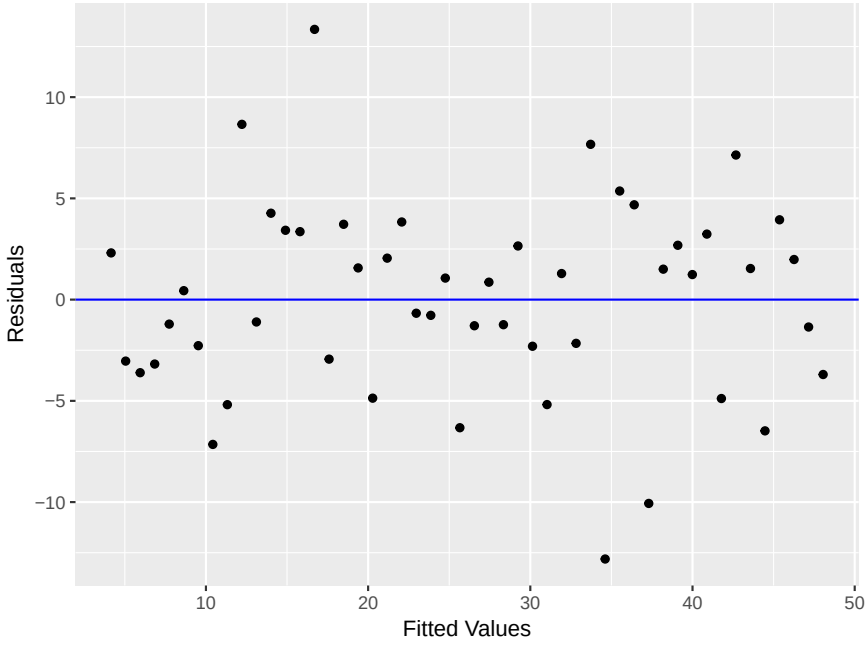
1) Good Model

For this assumption draw the Residuals vs. Fits plot and check for **any pattern**

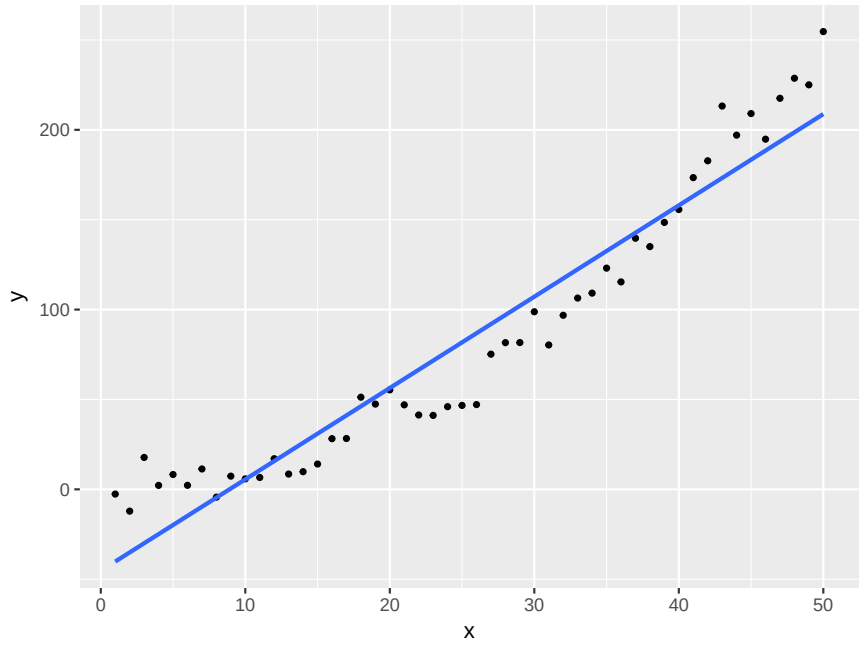
Example:

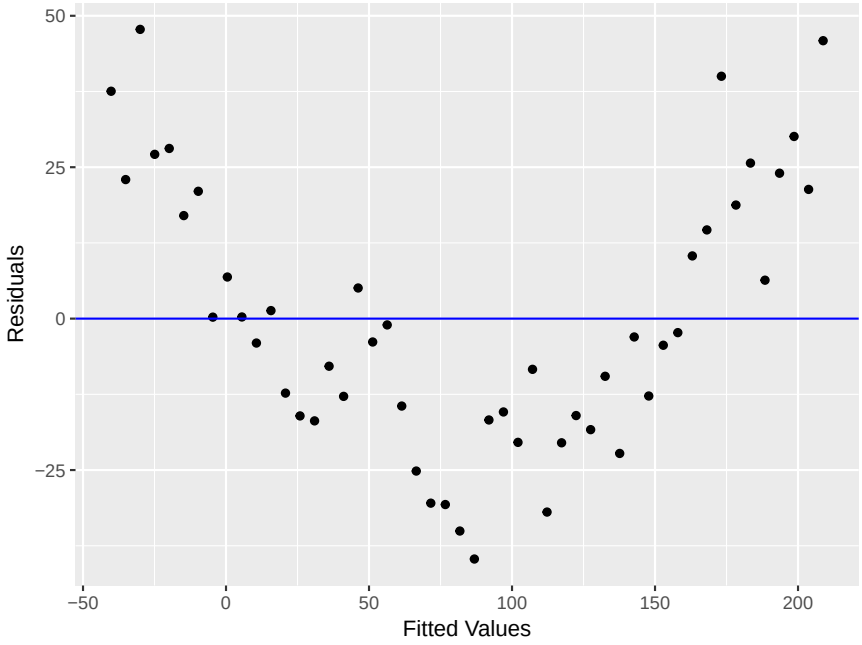
Linear model is good:





Linear model is bad:





The U shaped pattern in the residual vs. fits plot is a very common one if the linear model is bad.

2) Residuals have a Normal Distribution

For this assumption draw the normal probability plot and see whether the **dots form a straight line**, just as we have done it many times by now.

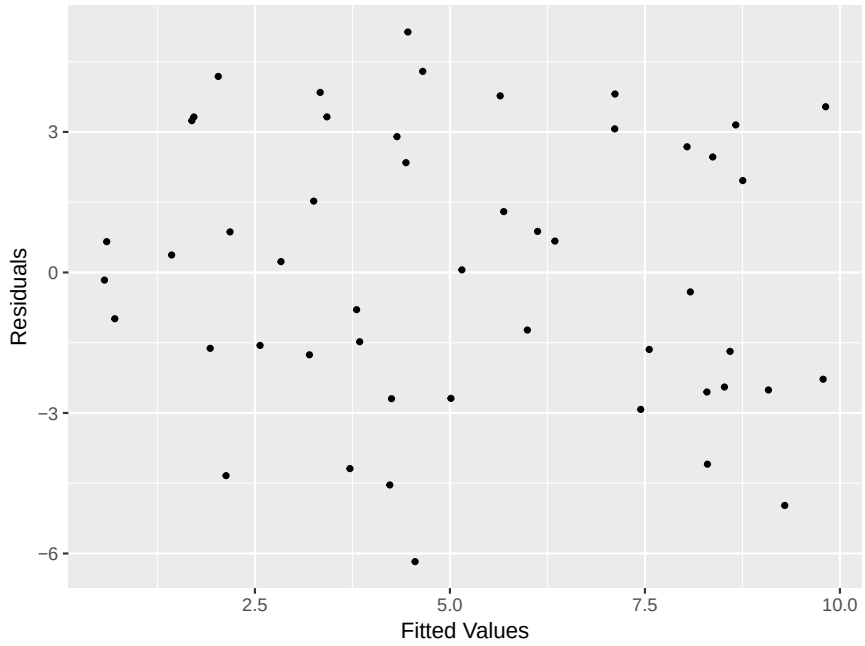
3) Residuals have Equal Variance

Previously we could check the stdev within the groups and see whether they differed by more than a factor of 3. Now, though we don't have groups. Instead we will again draw the

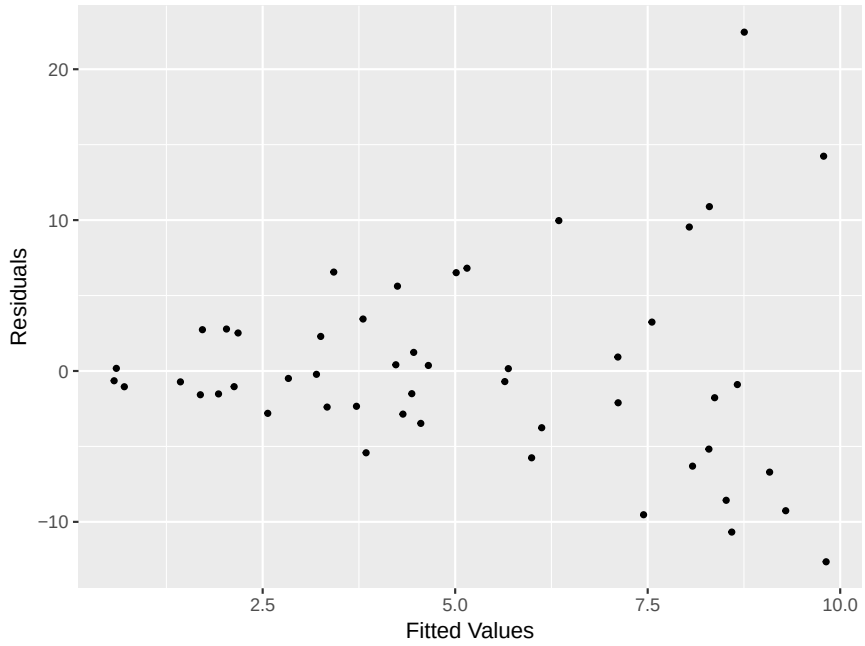
Residuals vs. Fits plot and check whether the *variance (or spread) of the dots changes as you go along the x axis.*

23.0.0.1 Example:

Equal Variance ok:



Equal Variance not ok:

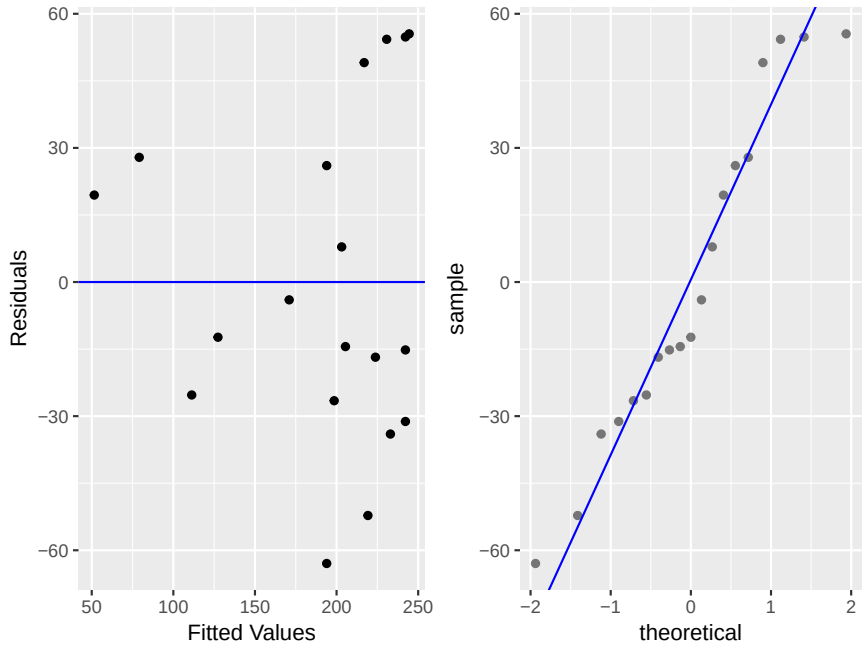


This can be a tricky one to decide, especially if there are few observations.

23.0.1 Case Study: Wine Consumption and Heart Disease

Let's check the assumptions for the wine consumption data:

```
attach(wine)
slr(Heart.Disease.Deaths, Wine.Consumption)
```




```
## The least squares regression equation is:
##  Heart.Disease.Deaths  = 260.563 - 22.969 Wine.Consumption
## R^2 = 71.03%
```

the normal plot is fine, and the residual vs. fits plot is fine as far the linear model assumption goes. There is, though, an appearance of unequal variance. This judgement is made more difficult here, though, because there is very little data in the left half of the graph, and naturaly a few dots won't have a large spread. It will take time for you to be able to judge these graphs properly. In fact this one is ok. Not great, but ok.

Note a final decision on whether the assumptions are justified is ALWAYS made based on the Residual vs. Fits Plot and the Normal plot of Residuals.

24 Hypothesis Tests in SLR

24.0.1 Case Study: Hubble's Constant

In 1929 Edwin Hubble published a paper showing a relationship between the distance and radial velocity away from Earth of "extra-galactic nebulae" (galaxies). His findings revolutionized astronomy. The "Hubble constant," the slope of the regression of velocity (Y) on distance (X), is still a subject of research and debate.
The data here are those Hubble published in his original paper. It also has data from much more recent studies.

```
head(hubble)
```

```
##   Velocity Distance Galaxy.NGC. velocity.km.s Distance.Mpc.
## 1      170    0.032         925           553          9.70
## 2      290    0.034       1326A          1831         15.81
## 3     -130    0.214        1365          1636         18.48
## 4      -70    0.263        1425          1510         20.83
## 5     -185    0.275        2090           921         11.57
## 6     -220    0.275        2541           548         12.06
```

The data set has the variables **Velocity** and **Distance**. Here is some info on these:

24.0.2 Velocity

Velocity (Speed with a sign) is measured in km/sec. How can one measure the speed with which a galaxy moves relative to earth? This is done using the **Doppler Effect**:

For more on the Doppler Effect go to Doppler Effect

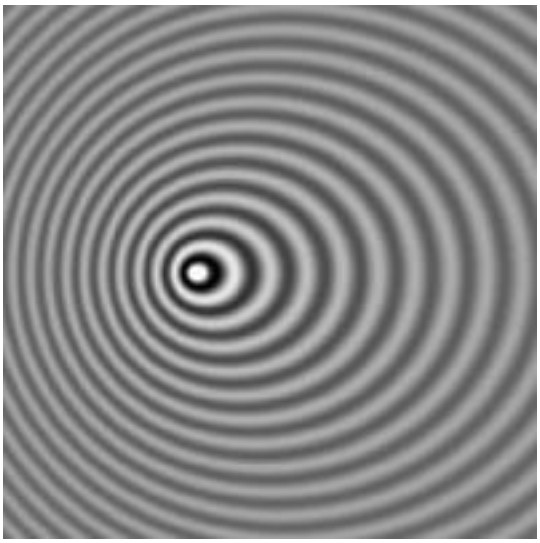


Figure 27:

24.0.3 Distance

The unit of distance in our dataset is one **Megaparsec**, or 1 million parsecs. A parsec is equal to 3.262 light years, or 19.17 billion miles. Here are some astronomical distances for illustration:

- Earth to Moon: 240000 miles (or 1.3 light seconds)
- Earth to Sun: 92 million miles (or 8.2 light minutes)
- Earth to nearest Solar System (Alpha Centauri): 8.37 billion miles (or 4.365 light years or 1.338 parsec)
- Earth to nearest galaxy (Andromeda): 48000 million million miles (or 2.5 million light years or 740 000 parsec or 0.7 megaparsec)

How does one measure the distance of a galaxy (or a star)? It is done using a method called **parallax**:

For more on parallax go here

```
attach(hubble)
splot(Velocity, Distance)
```

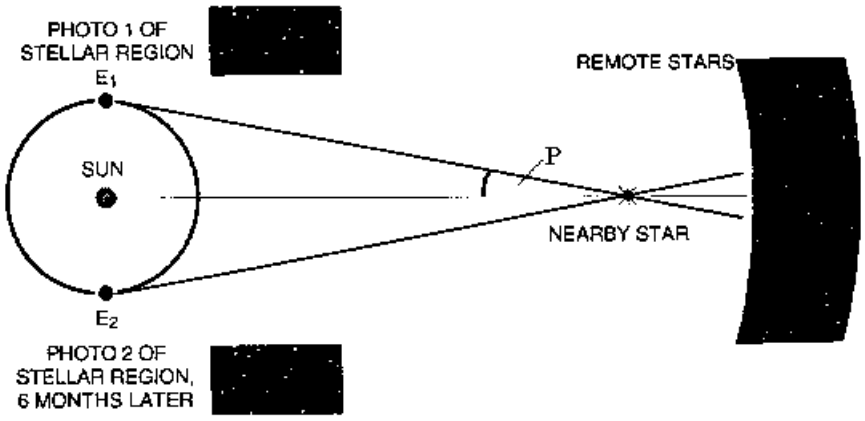
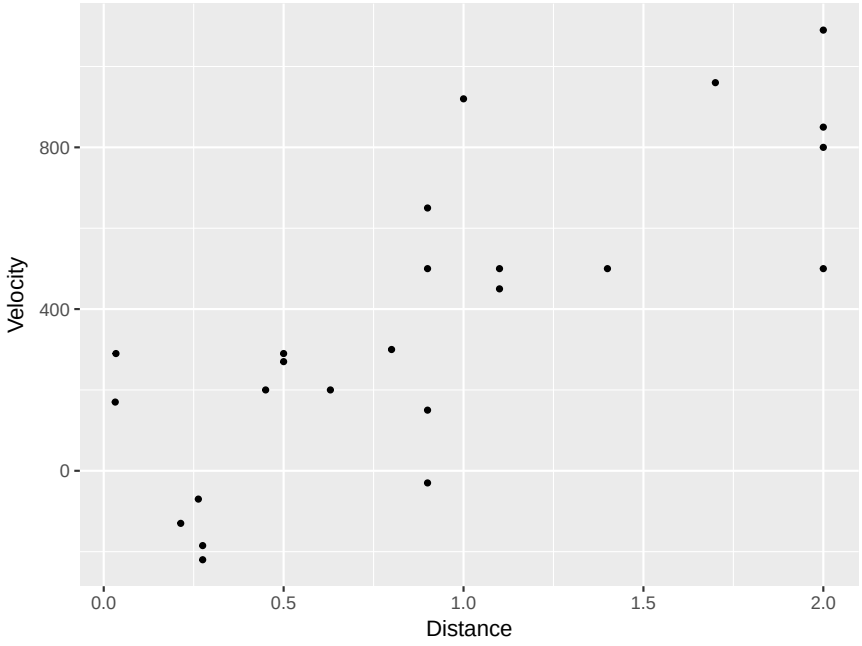
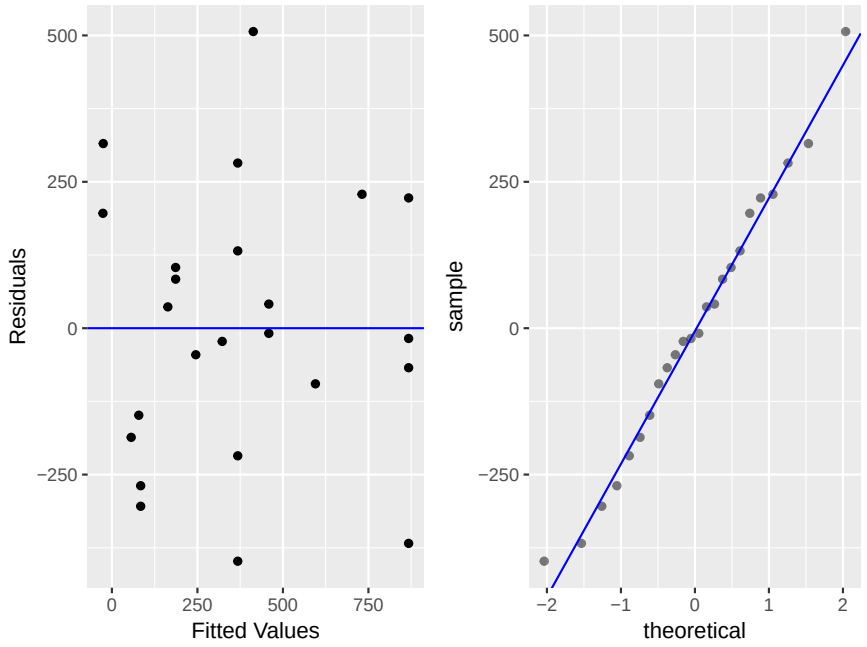


Figure 28:



The scatterplot of Velocity by Distance shows a strong relationship.

```
slr(Velocity, Distance, show.tests = TRUE)
```




```
## The least squares regression equation is:
##  Velocity  = -40.784 + 454.158 Distance
## Constant: p = 0.6298
## Distance : p = 0
## R^2 = 62.35%
```

The two graphs show us that the assumptions of LSR are justified. Let's discuss the next part of the output:

- Constant: p = 0.6298
- Distance: p = 0

Whenever there is a p value, there is a hypothesis test. Here there are two. The first one is for the

24.1 Constant:

$$H_0 : \beta_0 = 0 \text{ (intercept is zero)}$$
$$H_a : \beta_0 \neq 0 \text{ (intercept is not zero)}$$

If we fail to reject H_0, we conclude that the constant is not statistically significantly different from 0 (at the sample size of the data set!).

If we reject H_0, we conclude that the constant is statistically significantly different from 0.

Consequences:

We are fitting the model

$$y = \beta_0 + \beta_1 x$$

If H_0 is true then $\beta_0 = 0$, so the model becomes

$$y = \beta_1 x$$

this is called a **no-intercept** model. To get this model we have to rerun the regression:

```
slr(Velocity, Distance, no.intercept=TRUE)
```

```
## The least squares regression equation is:
##  Velocity  =   423.937 Distance
## R^2 = 62.35%
```

The slope of the line **423.9** is called Hubble's constant and is one of the fundamental constants in the universe!

Note the slope of the no intercept model (423.9) is NOT the same as the slope of the regular model (454.2)

Note The decision whether an intercept should be fit or not is best made based on the background of the data and whether if x=0 then y=0 makes sense.

One consequence of this model is that if x=0 then

$$y = \beta_1 x = \beta_1 0 = 0$$

so the point (0,0) is **always** on this line.

Example Say we have data with x = Number of Hurricanes in a year and y = $ Amount of Damage done by the Hurricanes. Now if x=0 (there were no hurricanes) obviously y=0 (no damage), so a no-intercept model is appropriate (even if the corresponding hypothesis test says otherwise!)

24.2 Predictor:

$$H_0 : \beta_1 = 0 \text{ (slope is zero)}$$
$$H_a : \beta_1 \neq 0 \text{ (slope is not zero)}$$

Consequences: our model is

$$y = \beta_0 + \beta_1 x$$

If H_0 is true then $\beta_1 = 0$, so the model becomes

$$y = \beta_0 + 0x = \beta_0$$

But there is no more predictor x here! So if we fail to reject H_0 it means that the predictor has no statistically significant relationship with the response (at least not for the sample size of the dataset).

If we do reject H_0 we conclude that there is a statistically significant relationship between predictor and the response y.

Note in a simple regression model such as we have here this test is the same as the test for Pearson's correlation coefficient.

24.3 Hubble data:

Constant:

1) Parameter: Intercept in SLR

2) Method: SLR

3) Assumptions: same as SLR

4) $\alpha = 0.05$

5) $H_0 : \beta_0 = 0$ (intercept is zero)

6) $H_a : \beta_0 \neq 0$ (intercept is not zero)

7) $p = 0.6298$

8) $p > \alpha$, so we fail to reject the null hypothesis, the constant is stat. consistent with 0 (at the sample size of the dataset)

Distance:

1) Parameter: Slope in SLR

2) Method: SLR

3) Assumptions: same as SLR

4) $\alpha = 0.05$

5) $H_0 : \beta_1 = 0$ (slope is 0, no relationship between distance and velocity)

6) $H_a : \beta_1 \neq 0$ (slope is not 0, some relationship between distance and velocity)

7) p = 0.000

8) $p < \alpha$, so we reject the null hypothesis, there is a statistically significant relationship between velocity and distance.

24.4 What are these tests good for?

Notice that when I ran the least squares regression command

```
slr(Velocity, Distance, show.tests = TRUE)
```

I added the argument show.tests = TRUE. Without it these tests would not be done. That is because in many ways they are useless!

- whether or not a no-intercept model is what we want should be decided by our understanding of the experiment, not the outcome of the the test for the constant
- the test for the slope is the same as Pearson's correlation test, which we likely already did!

I have discussed them here because you will see them in real live and so you should know what they are.

24.5 What it all means

What are the consequences of all this for our understanding of the universe?

25 Prediction

25.1 Categorical - Categorical

25.1.1 Case Study: Treatment for Hair Loss

Say we want to know the following: what is the percentage of men who using Rogain will grow no hair? The answer is simple: 301 of 714 for 301/714*100% = 42.2% of the men in the treatment group had no hair growth. As always though in Statistics we also want an estimate of the error in this prediction. We learned in 3101 how to do this:

```
one.sample.prop(301, 714)
```

```
## A 95% confidence interval for the population proportion is (0.385, 0.459)
```

Notice, though, that this calculation uses only the numbers 301 and 714, not any of the other results of the experiment. Moreover, if we did the same calculation for all the combinations of groups we would calculate 10 confidence intervals, and again we have a problem of **simultaneous inference**.

It turns out that this is a type of problem too difficult for this class.

25.2 Categorical - Quantitative

25.2.1 Case Study - Babies and Cocain Use by the Mother

Find 95% confidence intervals for the lengths of the babies in the Drug Free group:

```
attach(mothers)
one.sample.t(Length[Status=="Drug Free"], ndigit = 2)
```

```
## A 95% confidence interval for the population mean is (50.16, 52.04)
```

The difficulty again is if we do this for all three groups:

- Drug Free (50.16cm, 52.04cm)
- First Trimester (48.09cm, 50.51cm)
- Throughout (46.78cm, 49.22cm)

because these are individual ci's, not a collection of ci's with the correct confidence level. As above we have the problem of
simultaneous inference.

25.3 Quantitative -Quantitative

25.3.1 Case Study: Quality of Fish

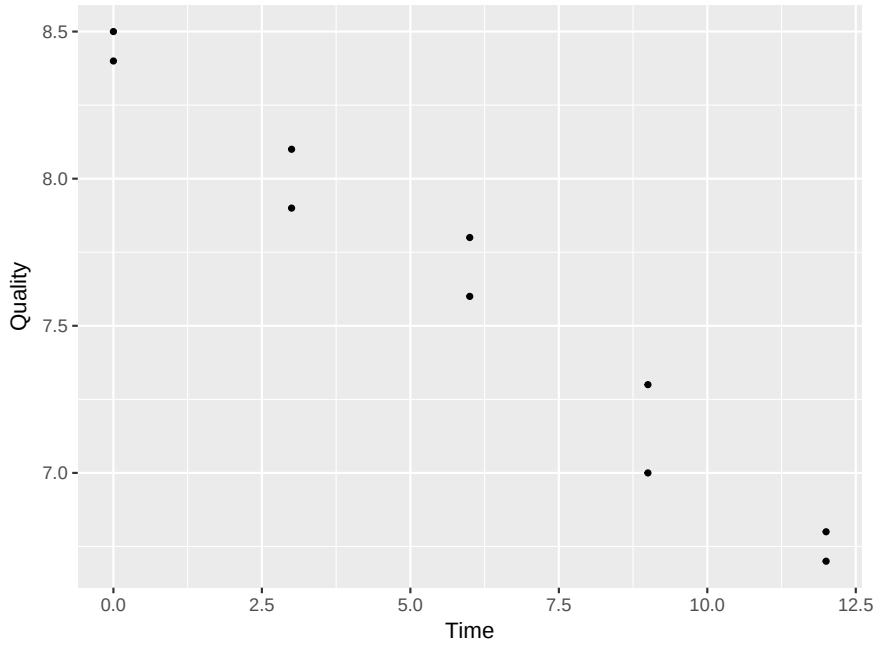
A study was conducted to examine the quality of fish after several days in ice storage. Ten raw fish of the same kind and quality were caught and prepared for storage. Two of the fish were placed in ice storage immediately after being caught, two were placed there after 3 hours, and two each after 6, 9 and 12 hours. Then all the fish were left in storage for 7 days. Finally they were examined and rated according to their "freshness".

Use this data set to estimate the quality of a fish that was put into ice 4 hours after being caught.

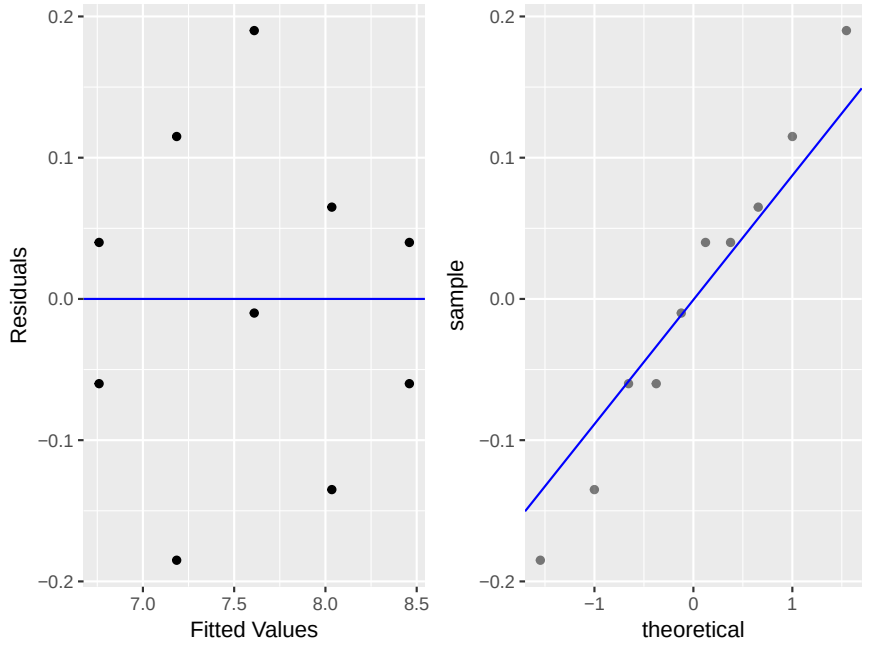
```
attach(fish)
fish
```

```
##    Time Quality
## 1     0     8.5
## 2     0     8.4
## 3     3     7.9
## 4     3     8.1
## 5     6     7.8
## 6     6     7.6
## 7     9     7.3
## 8     9     7.0
## 9    12     6.8
## 10   12     6.7
```

```
splot(Quality, Time)
```




```
slr(Quality, Time)
```




```
## The least squares regression equation is:
##  Quality  = 8.46 - 0.142 Time
## R^2 = 96.88%
```

assumptions look ok.

so we have

$$\text{Quality} = 8.46 - 0.142 * 4 = 7.9$$

We can also let R do the calculation for us:

```
slr.predict(Quality, Time, newx=4)
```

```
##  Time  Fit
##     4 7.89
```

25.3.2 Confidence vs. Prediction Intervals

Again we want an idea of the "error" in our estimate. Previously we used confidence intervals to do this. Here we will again use confidence intervals, but in the context of regression there are two types of intervals:

Confidence Interval - used to predict the **mean** response of **many** observations with the desired x value.

Prediction Interval - used to predict the **individual** response of **one** observation with the desired x value.

Warning The terminology is a little confusing here, with the same term meaning different things: Both confidence intervals and prediction intervals as found by the regression command are confidence intervals in the sense discussed before, and both are used for prediction!

They differ in what they are trying to predict, on the one hand an **individual response** (PI), on the other hand the **mean of many responses** (CI).

Example Let's consider the Quality of Fish data. Use this data set to find a 95% interval estimate for the quality of a fish that was put into storage after 4 hours.

We are talking about **one** fish, so we want a **prediction** interval:

```
slr.predict(Quality, Time, newx=4, interval="PI")
```

```
##  Time  Fit Lower Upper
##     4 7.89   7.6  8.19
```

so a 95% prediction interval for the rating of fish after 4 hours is (7.60, 8.19)

Example Again consider the Quality of Fish data. Use this data set to find a 90% interval estimate for the mean quality of fish that were put into storage after 4 hours.

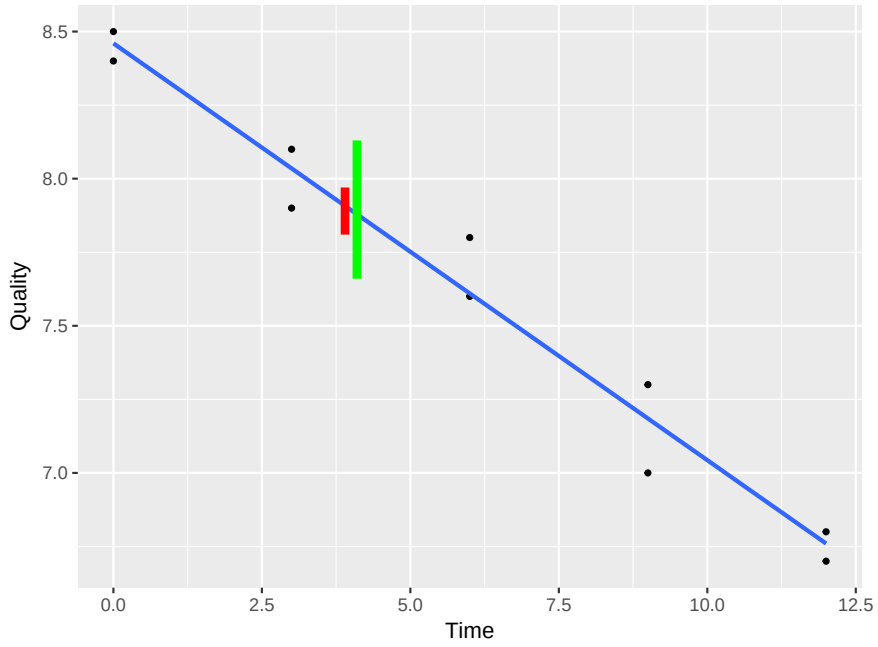
Now we are interested in the **mean** rating of many fish, so we want a **confidence** interval. Also we want a 90% interval instead of 95%:

```
slr.predict(Quality, Time, newx=4,
            interval="CI", conf.level = 90)
```

```
##  Time  Fit Lower Upper
##     4 7.89  7.81  7.97
```

so a 90% confidence interval for the mean rating of fish after 4 hours is (7.81, 7.97).

The two 90% intervals are shown in the next graph, the prediction interval in green and the confidence interval in red:



Notice that the prediction intervals are always wider than the confidence intervals. They are also the ones you want most of the time. So if you are not sure which you should use, use the prediction interval.

The slr.predict command can also be used to find a number of fits and intervals simultaneously:

```
slr.predict(Quality, Time, newx=1:10,
            interval="PI", conf.level = 90)
```

```
##  Time  Fit Lower Upper
##     1 8.32  8.07  8.57
##     2 8.18  7.93  8.42
##     3 8.04  7.79  8.28
##     4 7.89  7.66  8.13
##     5 7.75  7.52  7.99
##     6 7.61  7.37  7.85
##     7 7.47  7.23  7.70
##     8 7.33  7.09  7.56
##     9 7.19  6.94  7.43
##    10 7.04  6.80  7.29
```

If the newx argument is left off the predicition is done for the data itself:

```
slr.predict(Quality, Time,
            interval="PI", conf.level = 90)
```

```
##  Time  Fit Lower Upper
##     0 8.46  8.20  8.72
##     0 8.46  8.20  8.72
##     3 8.04  7.79  8.28
##     3 8.04  7.79  8.28
```

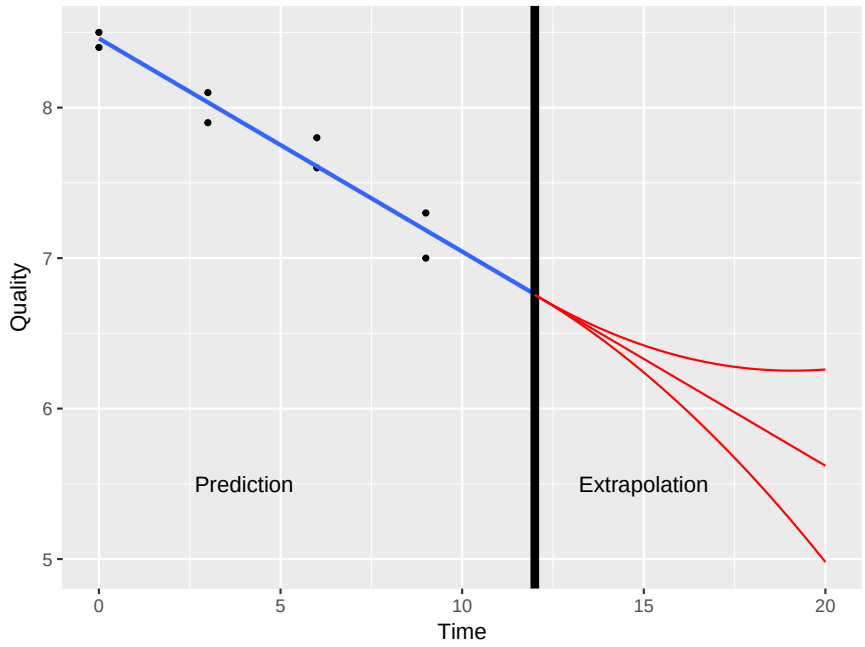
```
##      6 7.61  7.37  7.85
##      6 7.61  7.37  7.85
##      9 7.19  6.94  7.43
##      9 7.19  6.94  7.43
##     12 6.76  6.50  7.02
##     12 6.76  6.50  7.02
```

25.4 Prediction vs. Extrapolation

There is a fundamental difference between predicting the response for an x value **within** the range of observed x values (=Prediction) and for an x value **outside** the observed x values (=Extrapolation). The problem here is that the model used for prediction is only known to be good for the range of x values that were used to find it. Whether or not it is the same outside these values is generally impossible to tell.

Note Another word for prediction is **interpolation**

Example: Quality of Fish data



26 Non-Normal Residuals, No Equal Variance - Transformations

26.1 Categorical - Quantitative

26.1.1 Case Study: Cancer Survival

As we saw before we need a log transform of Survival to have normal residuals and equal variance, and then the ANOVA shows a stat. significant difference between the groups. The

only thing left is to do the multiple comparison. The main point here is that this also has to be done on the log transformed data:

```
attach(cancersurvival)
tukey(log(Survival), Cancer)
```

```
## Groups that are statistically significantly different:
```

```
##            Groups p.value
## 1 Breast-Bronchus  0.0083
## 2  Breast-Stomach  0.0158
```

26.1.2 Case Study: Capacity of Wells

Again, we have previously done everything except the multiple comparison:

```
attach(rocks)
tukey(log(Capacity), Rocks)
```

```
## Groups that are statistically significantly different:
```

```
##                 Groups p.value
## 1 Dolomite-Metamorphic  0.0105
## 2 Dolomite-Siliclastic  0.0275
## 3   Dolomite-Limestone  0.0434
```

Interpretation: Dolomite has a stat. significant larger capacity then the other rocks. Other differences are not stat. significant, at least not at these sample sizes.

26.1.3 Case Study: Cultural Differences in Equipment Use

Previously we saw that none of the transformations worked for this dataset, and we ran the Kruskal-Wallis test to see that there are differences in the MTBF by countries. Can we do a multiple comparison, as we did above? The answer is yes, but that goes beyond our discussion in 3102!

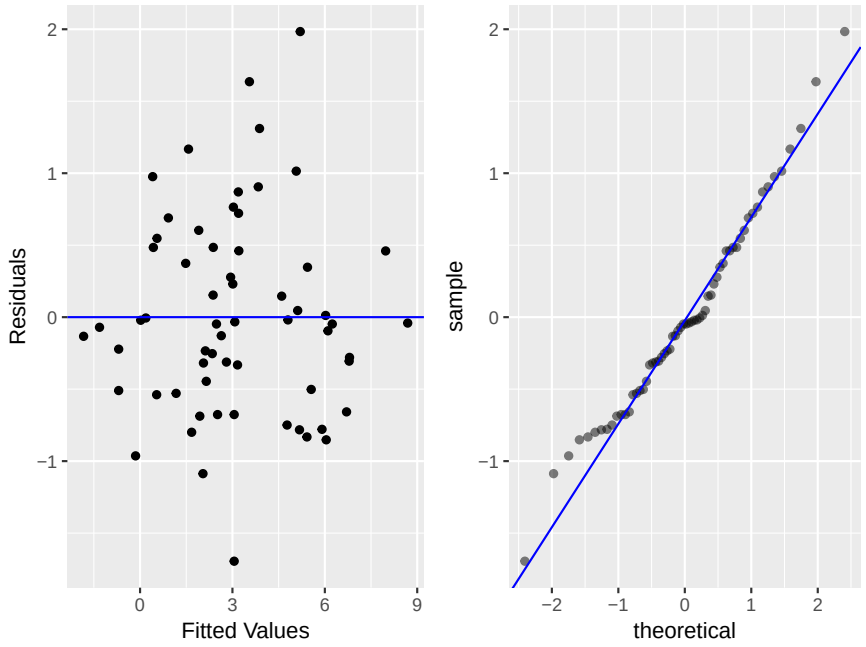
26.2 Quantitative - Quantitative

26.2.1 Case Study: Brain and Body Weight of 62 Mammals

We saw previously that we needed log transforms of both Brain and Body.

Then we find

```
attach(brainsize)
slr(log(brain.wt.g), log(body.wt.kg))
```



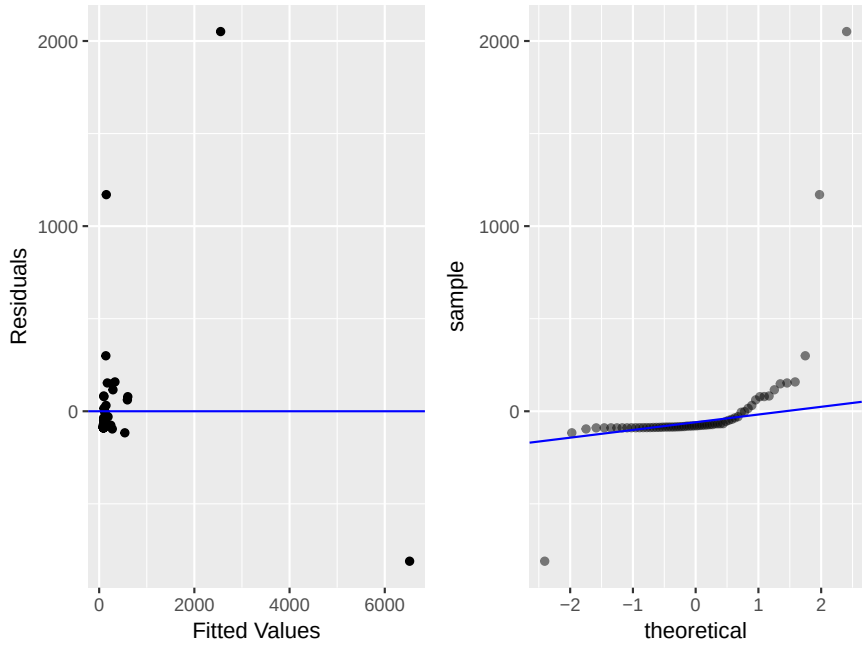

```
## The least squares regression equation is:
##  log(brain.wt.g)  = 2.121 + 0.746 log(body.wt.kg)
## R^2 = 91.93%
```

we see that now all the assumptions are ok. We find the model

$$\log(\text{brain.wt.g}) = 2.121 + 0.746 \log(\text{body.wt.kg})$$

If we had run the regression without the tranformations this is what the normal plot would have looked like:

```
slr(brain.wt.g, body.wt.kg)
```




```
## The least squares regression equation is:
##  brain.wt.g  = 89.912 + 0.967 body.wt.kg
## R^2 = 87.26%
```

Note transforming the data also has its down-side: it makes understanding the model much harder:

Model in original units: $\text{brain.wt.g} = 89.9 + 0.967\text{body.wt.kg}$

Model in transformed units: $\log(\text{brain.wt.g}) = 2.121 + 0.746\log(\text{body.wt.kg})$

the original model tells us that each extra kg of body weight roughly adds one gram of brain weight but what is the slope of 0.7 in the transformed model telling us?

However, sometimes with a bit of math one can rewrite the model in the original variables. Our log-log model turns out to be the same as the model

$$\text{brain.wt.g} = 8.339\ \text{body.wt.kg}^{0.746}$$

but this is not necessarily easier to understand.

26.3 Equal Variance

Sometimes a transformation of the response variable can help with this problem as well. Mostly, though, a more complicated method for analysing such a dataset is needed (such as weighted regression)

```
detach(cancersurvival)
detach(rocks)
detach(brainsize)
```

27 Nonlinear Regression Models

27.0.1 Case Study: Fabric Wear

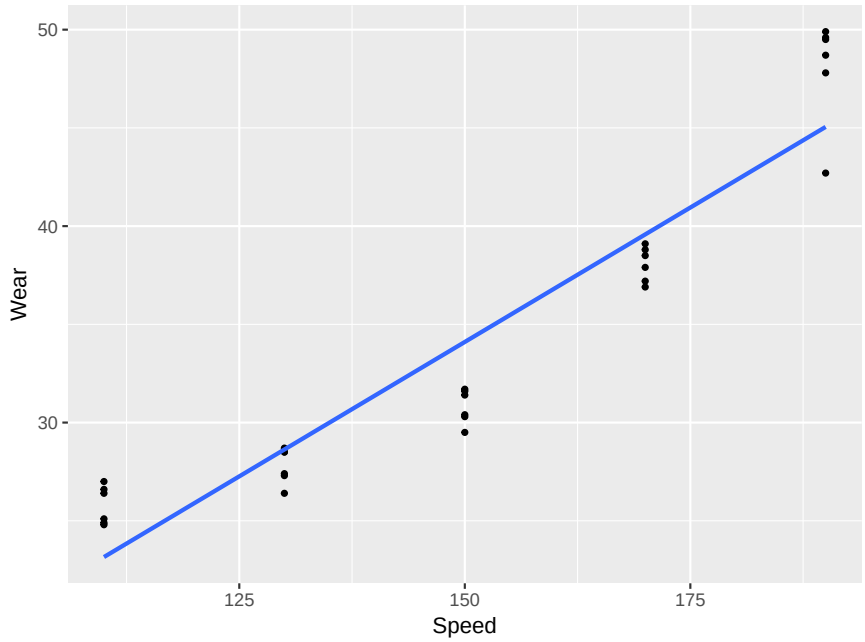
Results from an experiment designed to determine how much the speed of a washing machine effects the wear on a new fabric. The machine was run at 5 different speeds (measured in rpm) and with six pieces of fabric each.

```
head(fabricwear)
```

```
##   Speed Wear
## 1   110 24.9
## 2   110 24.8
## 3   110 25.1
## 4   110 26.4
## 5   110 27.0
## 6   110 26.6
```

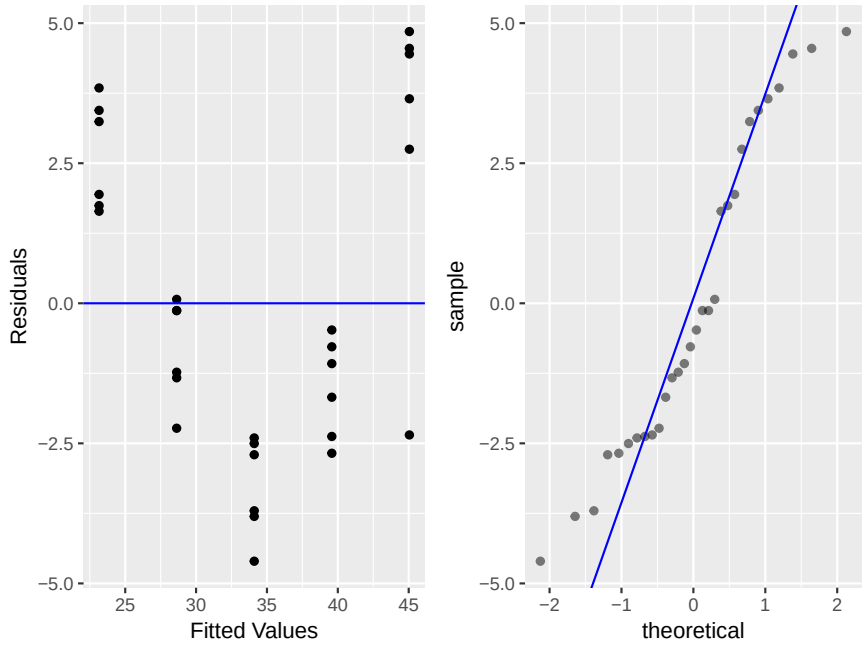
The scatterplot of wear by speed shows a strong but non-linear relationship:

```
attach(fabricwear)
splot(Wear, Speed, add.line=1)
```



How strong is a difficult question, because Pearson's correlation coefficient won't work here. If we tried slr we would see in the residual vs fits plot that there is a problem with the assumption of a linear model:

```
slr(Wear, Speed)
```

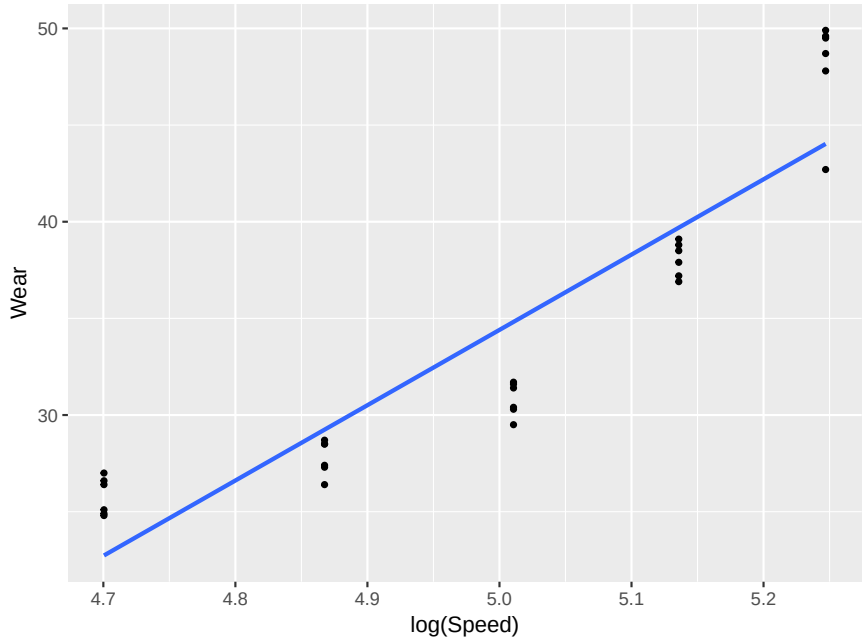



```
## The least squares regression equation is:
##  Wear  = -6.947 + 0.274 Speed
## R^2 = 88.58%
```

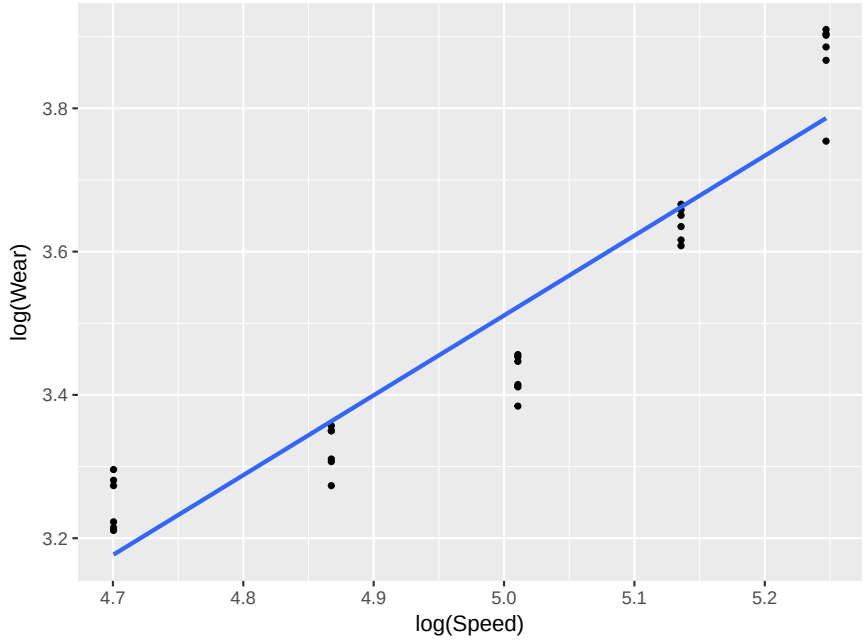
So the question is: how do fit models other than straight lines?

There are two basic things we can try. The first is something we have already done, namely the **log transformation**

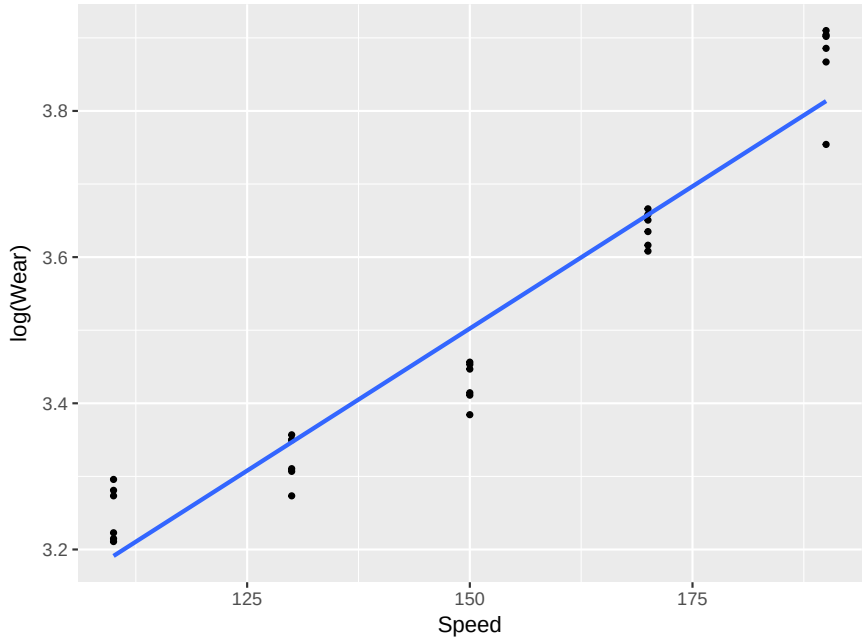
```
splot(Wear, log(Speed), add.line=1)
```



```
splot(log(Wear), log(Speed), add.line=1)
```




```
splot(log(Wear), Speed, add.line=1)
```



unfortunately non of these looks very good

Some of these have names:

- log(y) vs. x is called an **exponential model**
- log(y) vs. log(x) is called a **power model**

The other solution to our problem is to fit a **Polynomial Model**:

Linear $y = \beta_0 + \beta_1 x$

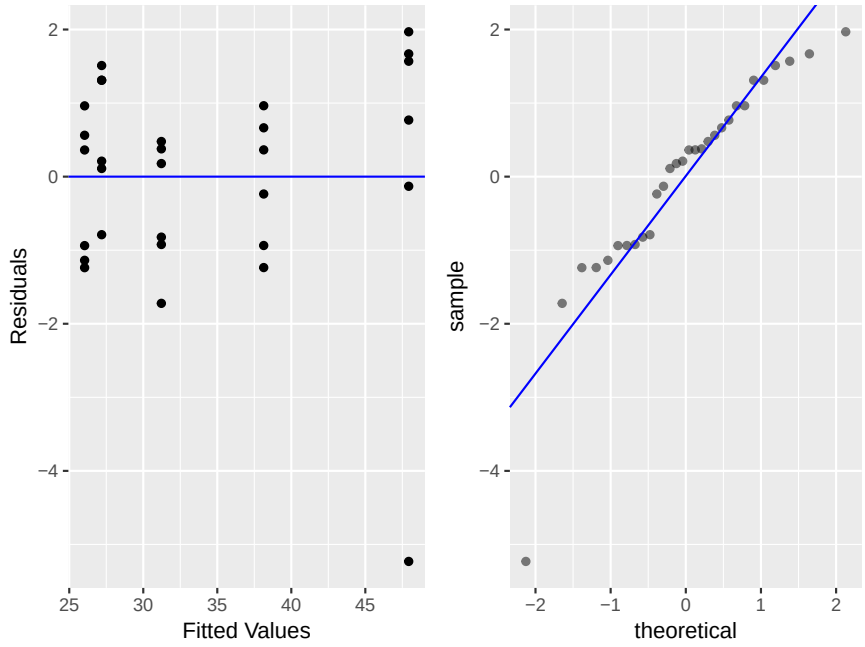
Quadratic $y = \beta_0 + \beta_1 x + \beta_2 x^2$

Cubic $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$

and so on

How do we fit such a model? We can simply use the same routine with the extra argument *polydeg=...*. For example for the quadratic model we do

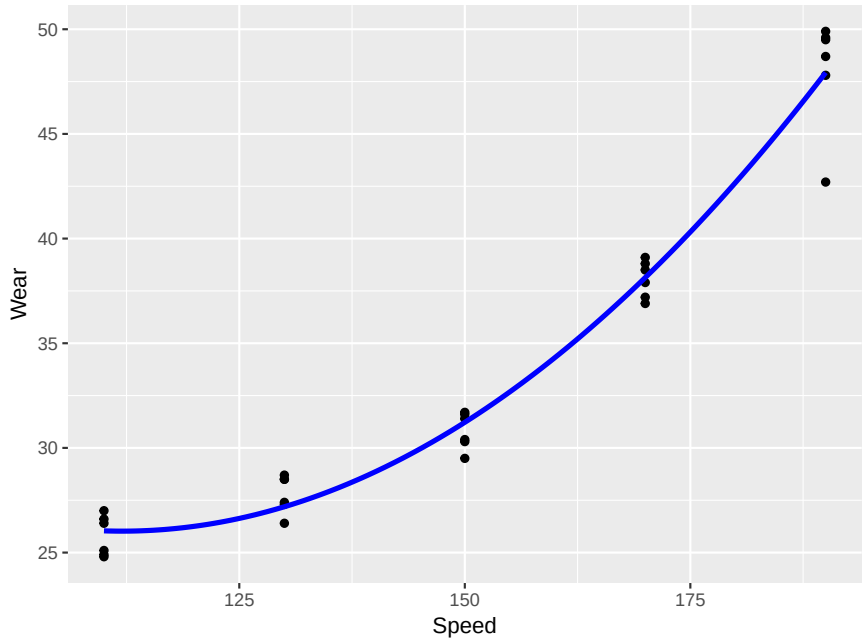
```
slr(Wear, Speed, polydeg=2)
```




```
## The least squares regression equation is:
##  Wear  = 71.199 - 0.807 Speed +0.004Speed^2
## R^2 = 97.17%
```

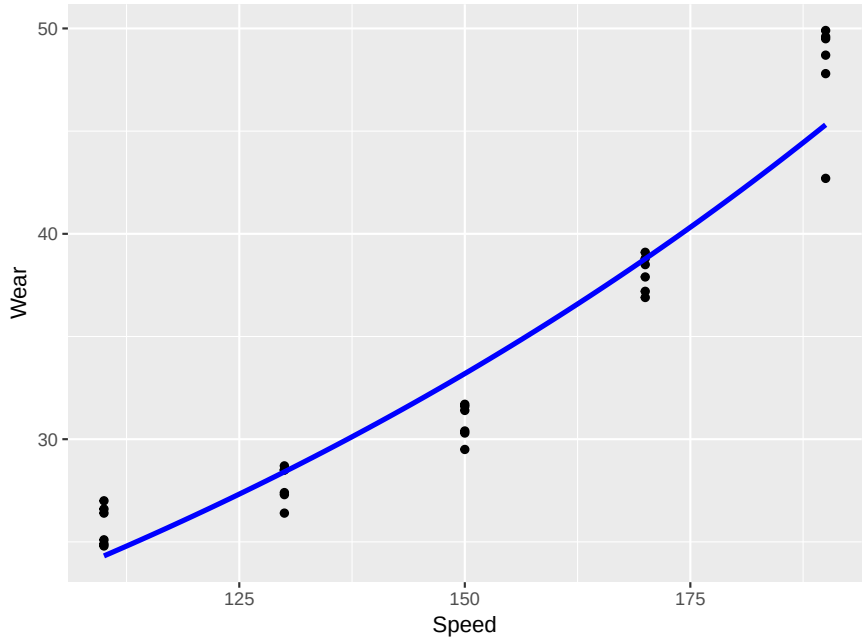
What does such a curve look like? To draw the fitted line plot, that is the scatterplot with the fitted curve, just use

```
flplot(Wear, Speed, polydeg=2)
```



This routine also does the log transform models:

```
flplot(Wear, Speed, logy=TRUE)
```



Similarly use flplot(Wear, Speed, logx=TRUE) or flplot(Wear, Speed, logx=TRUE, logy=TRUE) for the other log transforms

Note There are two big differences in the way transformations and polynomial models work:

- if we do a transformation we **replace** an old variable with a new one, if we do a polynomial model we **add** a new predictor to the model.

- we might transform the response, but a polynomial model is always a polynomial in the predictor, never the response.

Mathematical Features of these Models

What "shapes" can we fit with these models?

- Transformations might work if the relationship between x and y is **monotone**, that is in the scatterplot the dots either go up or down but never turn around.
- Polynomial models usually do turn around, quadratic models once, cubic models twice and so on. Sometimes this is not apparent because we only see the graph before the turn-around happens.

27.0.2 Prediction

Again we can use the slr.predict command to do prediction, but there are some things we need to be careful with:

Transformations

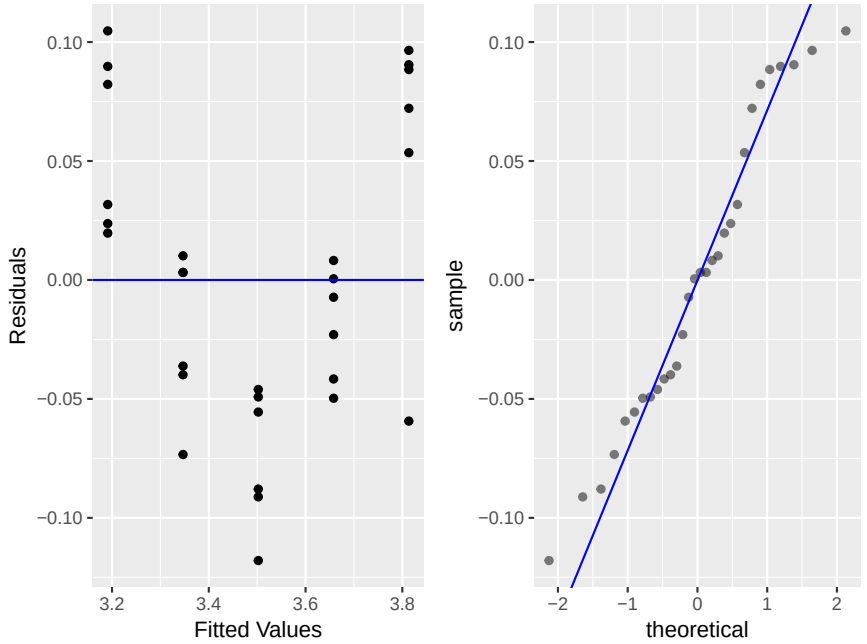
if we use a log transformation on the predictor we have to use the log transformation also on the newx:

```
slr.predict(Wear, log(Speed), newx=log(150))
```

```
##  log(Speed)   Fit
##    5.010635 34.82
```

if we use a log transformation on the response we are getting an estimate of the log of the response. To get back to the original we can do this:

```
slr(log(Wear), Speed)
```



```
## The least squares regression equation is:
##  log(Wear)  = 2.335 + 0.008 Speed
## R^2 = 92.64%
```

so we have the equation

$$\log(\text{Wear}) = 2.335 + 0.008\text{Speed}$$

and now we can get an estimate with

```
exp(2.335 + 0.008*150)
```

```
## [1] 34.29501
```

All of this works ONLY for point estimation, interval estimation is much harder and needs to be done by an expert!

In contrast, prediction using polynomials works perfectly fine as is:

```
slr.predict(Wear, Speed, newx=150, polydeg=2,
            interval="PI", conf.level= 90)
```

```
##  Speed   Fit Lower Upper
##    150 31.22 28.64 33.81
```

If you are not sure that you got the right answer, here is a quick sanity check: draw the scatterplot and do a visual guess of y.

Example: say we want to use the power model and predict the Wear for Speed=150:

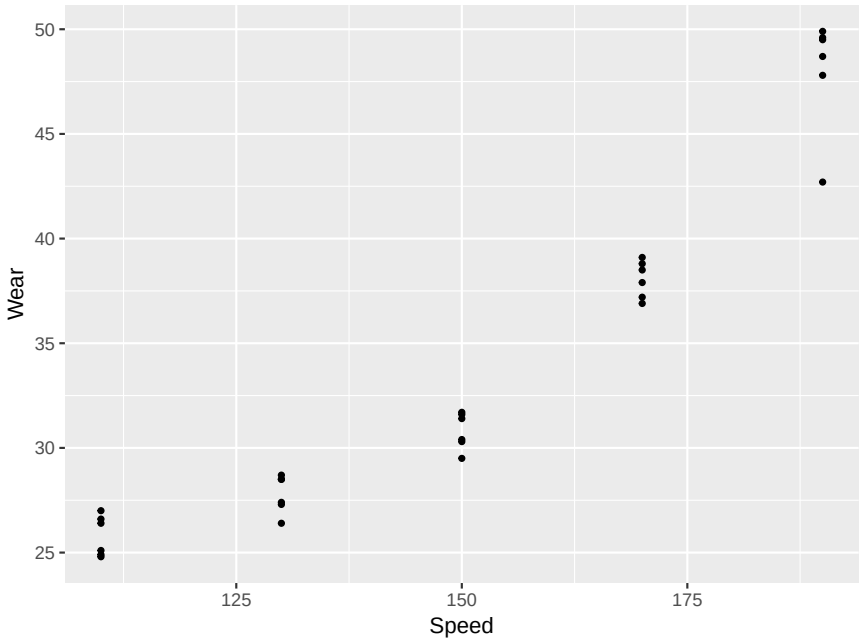
```
slr.predict(log(Wear), log(Speed), newx=150)
```

```
##  log(Speed)    Fit
##         150 165.13
```

Notice that I should have written newx=log(150).

Now if I draw the fitted line plot:

```
splot(Wear, Speed)
```



it is clear that if x=150 y should be be around 32 or so, not 165!

So I better try again:

```
slr.predict(log(Wear), log(Speed), newx=log(150))
```

```
##  log(Speed)  Fit
##    5.010635 3.52
```

and that's about right because

```
log(32)
```

```
## [1] 3.465736
```

28 Exercise Problems 3

28.0.1 Case Study: Survey of Students

This is the same data set we considered in Exercise Problems 1. The data is in **studentsurvey**

Problem 1 What can you say about the relationship between Score and Gender?

Problem 2 What can you say about the relationship between Score and GPA? Find a 95% interval estimate for a student with a GPA of 2.5. Is this an interpolation or an extrapolation?

Problem 3 What can you say about the relationship between Score and Distance? Find a 99% interval estimate for a student who lives 1.5 miles from the school. Is this an interpolation or an extrapolation?

Problem 4 What can you say about the relationship between Score and Age? Find a 90% interval estimate for the mean score of 21 year old students. Is this an interpolation or an

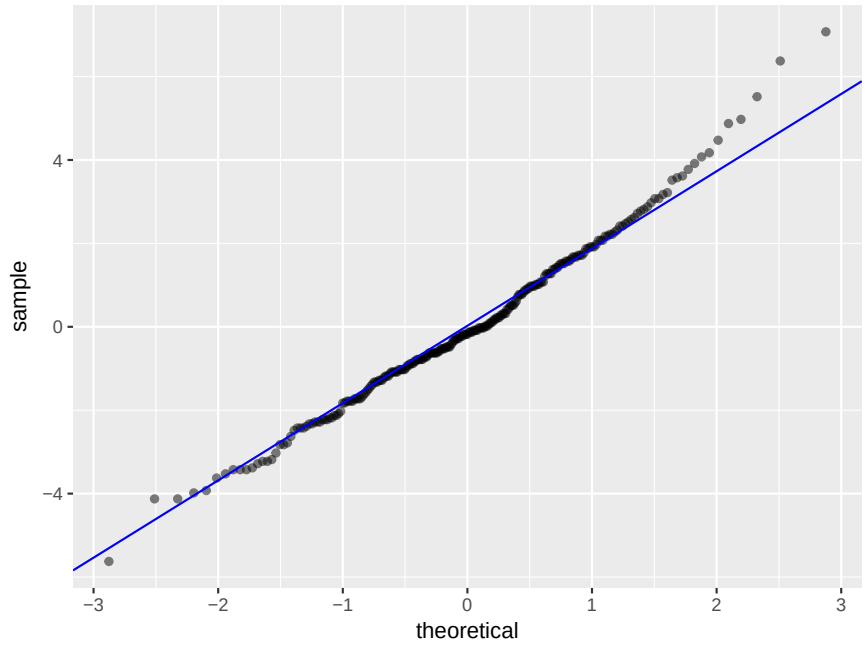
extrapolation?

```
attach(studentsurvey)
```

Problem 1 What can you say about the relationship between Score and Gender?

In problem 1 of the Exercise Problems 2 we ran the ANOVA and found a statistically significant difference between the scores of males and females. Because there are just two groups there is no reason to run tukey, but rerunning the same command gives us a 95% confidence interval for the difference in scores:

```
oneway(Score, Gender)
```

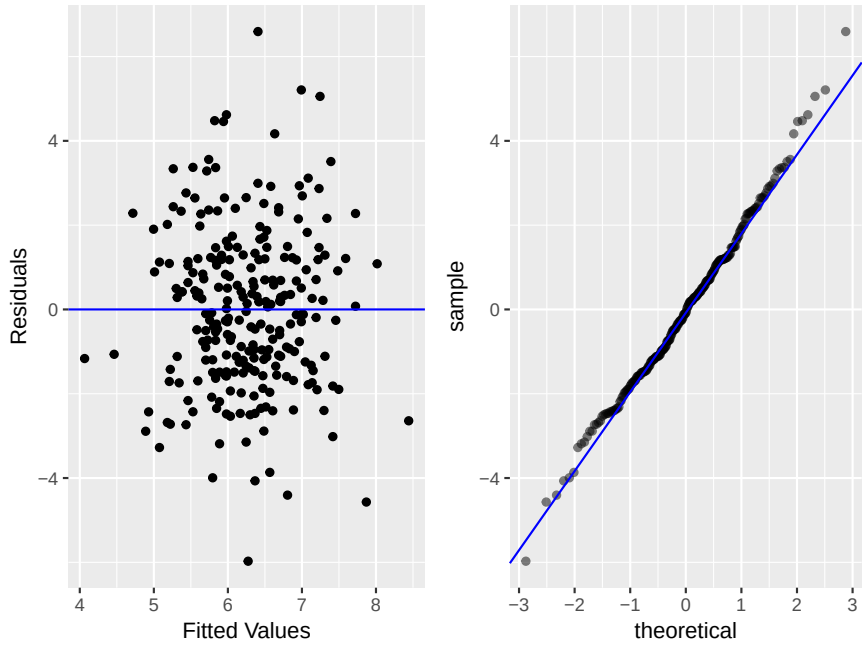



```
## p value of test of equal means: p = 0.0033
## Smallest sd:  1.7    Largest sd : 2.2
## A 95% confidence interval for the difference in group means is (0.3, 1.2)
```

Problem2 What can you say about the relationship between Score and GPA?

In problem 3 of the Exercise Problems 2 we found a statistically significant correlation between Score and GPA. Let's find a good model.

```
slr(Score, GPA)
```




```
## The least squares regression equation is:
##  Score  = 3.12 + 1.33 GPA
## R^2 = 11%
```

the residual vs fits plt and the normal plot looks good, so no problem with the assumptions. We find the model

$$\text{Score} = 3.12 + 1.33 \text{ GPA}$$

Find a 95% interval estimate for a student with a GPA of 2.5. Is this an interpolation or an extrapolation?

```
slr.predict(Score, GPA, newx=2.5, interval="PI")
```

```
##  GPA  Fit Lower Upper
##  2.5 6.45  2.65 10.24
```

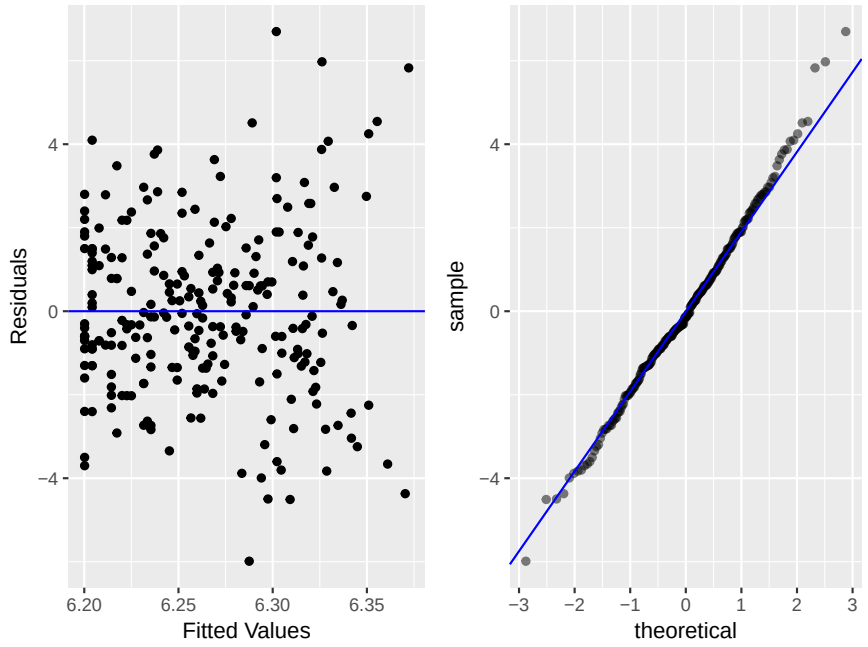
so a 95% prediction interval for a student with a GPA of 2.5 is (2.65, 10.24)

This is an interpolation because 2.5 is in the range of GPAs in the data set

Problem 3 What can you say about the relationship between Score and Distance?

In problem 5 of the Exercise Problems 2 we used a log transform on Distance. Doing so again yields

```
slr(Score, log(Distance + 1))
```




```
## The least squares regression equation is:
##  Score  = 6.2 + 0.042 log(Distance + 1)
## R^2 = 0.05%
```

the residual vs fits plt and the normal plot looks good, so no problem with the assumptions. We find the model

Score = 6.2 + 0.042 log(Distance+1)

Find a 99% interval estimate for a student who lives 1.5 miles from the school. Is this an interpolation or an extrapolation?

```
slr.predict(Score, log(Distance+1),
        newx=log(1.5+1), interval="PI", conf.level = 99)
```

```
##  log(Distance + 1)  Fit Lower Upper
##          0.9162907 6.24  0.94 11.54
```

so a 95% prediction interval for a student who lives 1.5 miles from the school is (0.94, 11.54)

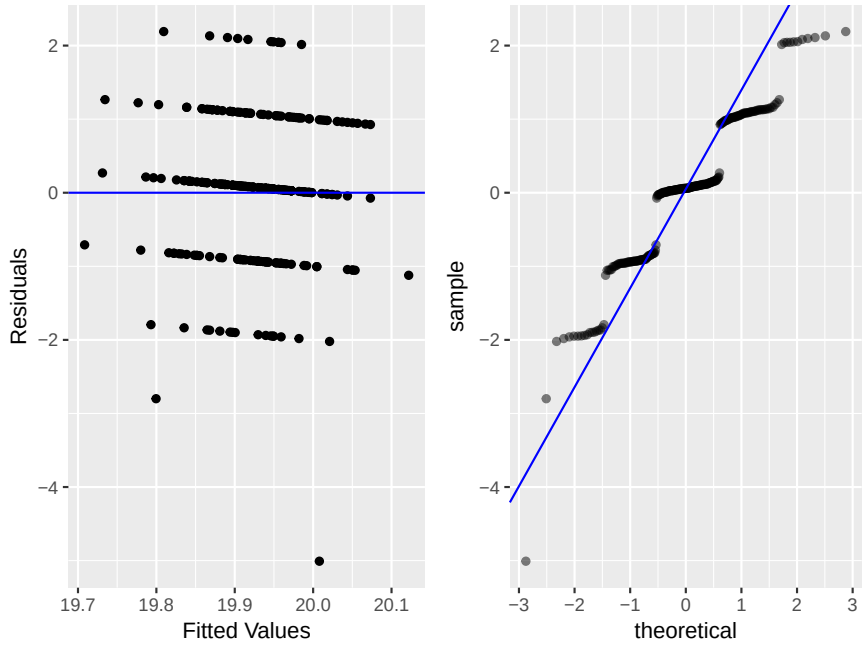
Note newx=log(1.5+1) because we have the predictor log(Distance+1).

This is an interpolation because 1.5 is in the range of Distances in the data set.

Problem 4 What can you say about the relationship between Score and Age?

Before we saw that observation #220 is an outlier and removed it. We do the same now. Then

```
slr(Age[-220], Score[-220])
```




```
## The least squares regression equation is:
##  Age[-220]  = 20.132 - 0.033 Score[-220]
## R^2 = 0.43%
```

Find a 90% interval estimate for the mean score of 21 year old students. Is this an interpolation or an extrapolation?

```
slr.predict(Score[-220], Age[-220],
            newx=21, interval="CI", conf.level = 90)
```

```
##  Age[-220]  Fit Lower Upper
##         21 6.13  5.82  6.44
```

so a 90% confidence interval for the mean score of 21 year old students is (5.82, 6.44)

This is an interpolation because 21 is in the range of Ages in the data set.

29 All Categorical - Simpson's Paradox

(or better Yule-Simpson's Paradox)

29.0.1 Case Study: Sex Discrimination in Graduate School Admissions

The famous Berkeley data on sex discrimination. In fall quarter, 1973, there were 8,442 men who applied for admission to graduate school, and 4,321 women.

Source: Freeman, D., Pisani, R., Purves, R. and Adhikiri, A. (1991) Statistics (2nd edition). WW Norton.

First we will look at the overall admittance numbers:

```
attach(berkeleyadmissions)
berkeleyadmissions[1:2, 1:3]
```

```
##   Overall  Sex Admitted
## 1    Men: 8442     3738
## 2  Women: 4321     1494
```

Let's find the percentages:

```
round(c(3738/8442, 1494/4321)*100, 1)
```

```
## [1] 44.3 34.6
```

which shows a sizable difference in admission rates. We can also do the test:

```
chi.ind.test(berkeleyadmissions[1:2, 2:3])
```

```
## p value of test p=0.000
```

1) Parameters of interest: measure of association
2) Method of analysis: chi-square test of independence
3) Assumptions of Method: all expected counts greater than 5
4) Type I error probability α=0.05
5) H_0: Classifications are independent = there is **no** difference in the admissions rates of men and women.
6) H_a: Classifications are dependent = there is some difference in the admissions rates of men and women.
7) p=0.000
8) 0.000<0.05, we reject the null hypothesis, there is some difference in the admissions rates of men and women.

Now let's consider the data with the majors

```
berM <- berkeleyadmissions[ ,5:6]
berM
```

```
##   Men.Applied Men.Admitted
## 1         825          512
## 2         560          353
## 3         325          120
## 4         417          138
## 5         191           53
## 6         373           22
```

```
round(berM[ ,2]/berM[ ,1]*100, 2)
```

```
## [1] 62.06 63.04 36.92 33.09 27.75  5.90
```

```
berF <- berkeleyadmissions[ ,7:8]
berF
```

```
##   Women.Applied Women.Admitted
## 1           108             89
```

```
## 2            25             17
## 3           593            202
## 4           375            131
## 5           393             94
## 6           341             24
```

```
round(berF[,2]/berF[,1]*100, 2)
```

```
## [1] 82.41 68.00 34.06 34.93 23.92  7.04
```

and suddenly any hint of sex discrimination is gone.

A formal hypothesis test for this is possible but outside the scope of this course.

So, we have a paradox:

- we found strong evidence (p value=0.00) of a relationship between the gender of an applicant and whether or not they were admitted to the School.
- when we broke down the data further by the major of the applicant, this relationship went away.

How is this possible?

Actually, we already know the answer: this is again an issue caused by confusing *Cause-Effect* with *Latent Variable*.

There is clearly a relationship between acceptance and gender. But saying it is due to sex discrimination is saying we have a cause - effect relationship. Instead we now know it is because of the latent variable Major.

Can we understand this in the Berkeley Admissions case?

Majors A and B are very popular with the men - 1385 men applied vs. 133 women. Majors A and B are also easy to get in - about 2 out of 3 of the applicants (men or women) get accepted. So although men and women have the same acceptance rate, 10 times as many men are accepted because 10 times as many applied.

Majors C-F are more popular with the women - 1346 men applied vs. 1702 women. But Majors C-F are hard to get in - about 1 in 4 of the applicants (men or women) get accepted. So these majors don't add much to the total student body.

If in an observational study (as opposed to a clinical trial with random assignments to "treatment" and "control") we find an relationship (association) between two variables it is usually very hard (impossible?) to decide whether it is due to a cause-effect relationship or whether there is a latent variable responsible for the relationship. In the Berkeley case it turned out that Major was a latent variable. A list of other potential latent variables includes:

1. Prior educational achievements
2. Age
3. Financial situation of parents

and so on.

Note that we could determine here that Majors is a latent variable explaining the relationship between Gender and Acceptance because we had the data to do so! So generally in a study you want to "measure" as many variables as possible because you won't know ahead of time which of them might turn out to be important.

30 Two Categorical Predictors - One Quantitative Response

30.0.1 Case Study: Testing Hearing Aids

Reference: Loven, Faith. (1981). A Study of the Interlist Equivalency of the CID W-22 Word List Presented in Quiet and in Noise. Unpublished MS Thesis, University of Iowa.

Description: Percent of a Standard 50-word list heard correctly in the presence of background noise. 24 subjects with normal hearing listened to standard audiology tapes of English words at low volume with a noisy background. They repeated the words and were scored correct or incorrect in their perception of the words. The order of list presentation was randomized.

The word lists are standard audiology tools for assessing hearing. They are calibrated to be equally difficult to perceive. However, the original calibration was performed with normal-hearing subjects and no noise background. The experimenter wished to determine whether the lists were still equally difficult to understand in the presence of a noisy background.

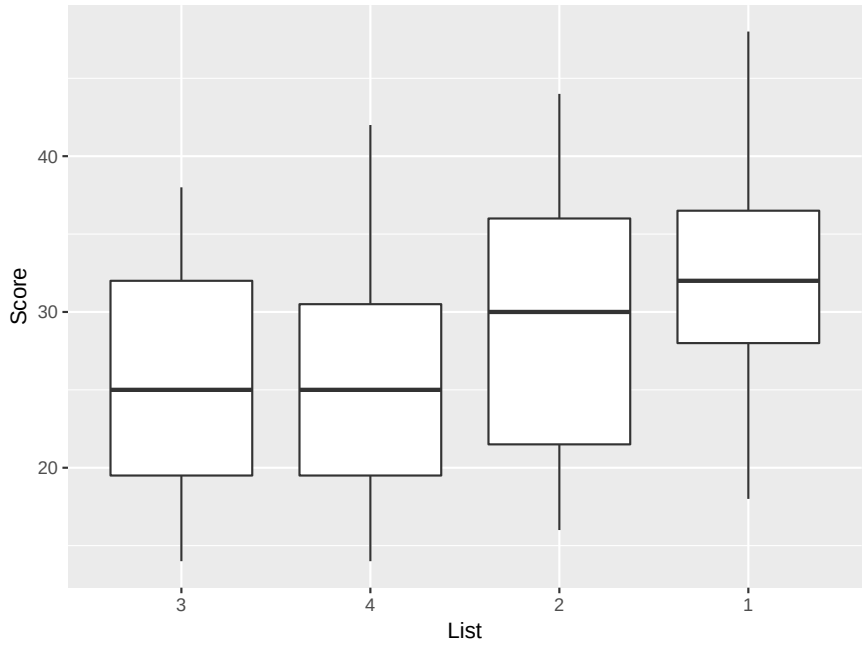
```
head(hearingaid)
```

```
##   Subject List Score
## 1       1    1    28
## 2       2    1    24
## 3       3    1    32
## 4       4    1    30
## 5       5    1    34
## 6       6    1    30
```

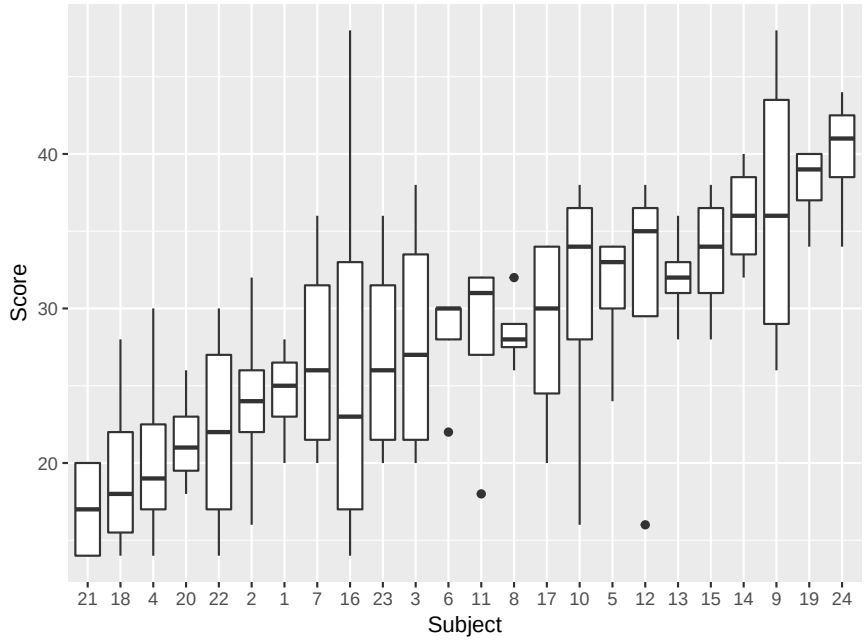
Notice that the values in both Subject and List are NOT numbers but labels, so both of them are categorical!

As long as we have one quantitative response (Score) and all the predictors (factors) are categorical (Subject, List) this is still an ANOVA problem, now called a **twoway ANOVA** More specifically, this is a **Randomized Block design** with List as the factor and Subject as a blocking variable.

```
attach(hearingaid)
bplot(Score, List, new_order = "Size")
```




```
bplot(Score, Subject, new_order = "Size")
```



The summary statistics are:

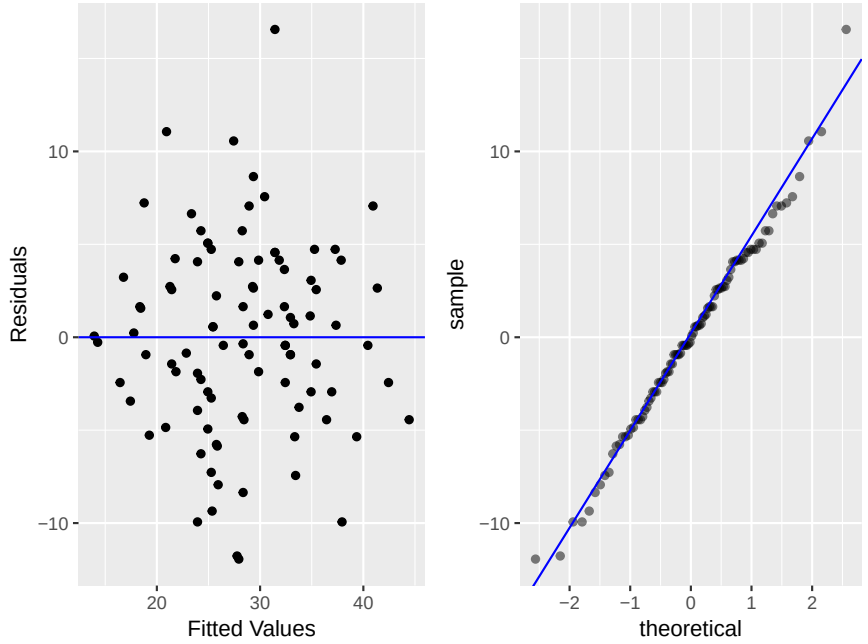
```
stat.table(Score, List, Sort = TRUE)
```

```
##   Sample Size Mean Standard Deviation
## 3          24 25.2                8.3
## 4          24 25.6                7.8
## 2          24 29.7                8.1
## 1          24 32.8                7.4
```

Because Subject is the blocking variable one would normally not include a table of summary statistics.

Now for the test, or better tests, because we can in general test for either Subject or List. The routine we will use is called twoway:

```
twoway(Score, List, Subject)
```




```
## No repeated measurement! Interaction term can not be included
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            3    920  306.82   8.446 7.41e-05
## z           23   3232  140.51   3.868 6.96e-06
## Residuals   69   2507   36.33
##               [,1]
## List  p =       0
## Subject  p =    0
```

So we have two tests, one for List and one for Subject. However, only the one for List is of interest:

1) Parameters of interest: List group means

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis H_0: $\mu_1 = .. = \mu_4$ (List groups have the same means)

6) Alternative hypothesis H_a: $\mu_i \neq \mu_j$ (at least two List groups have different means)

7) p value=0.00

8) 0.000<0.05, there is some evidence that the group means are not the same, that List means are different)

As always we need to check the assumptions. The normal plot of residuals looks fine.

Next the equal variance. In a oneway ANOVA we could just find the group standard deviations and compare them. Now (and in general if there are more than 1 factor) this is no longer a good idea mainy because there are to many factor level combinations (4*24 here) and not enough observations for each (1 here). Instead we will do the same as in the regression case, namely check the residual vs. fits plot for a change in spread from left to right.

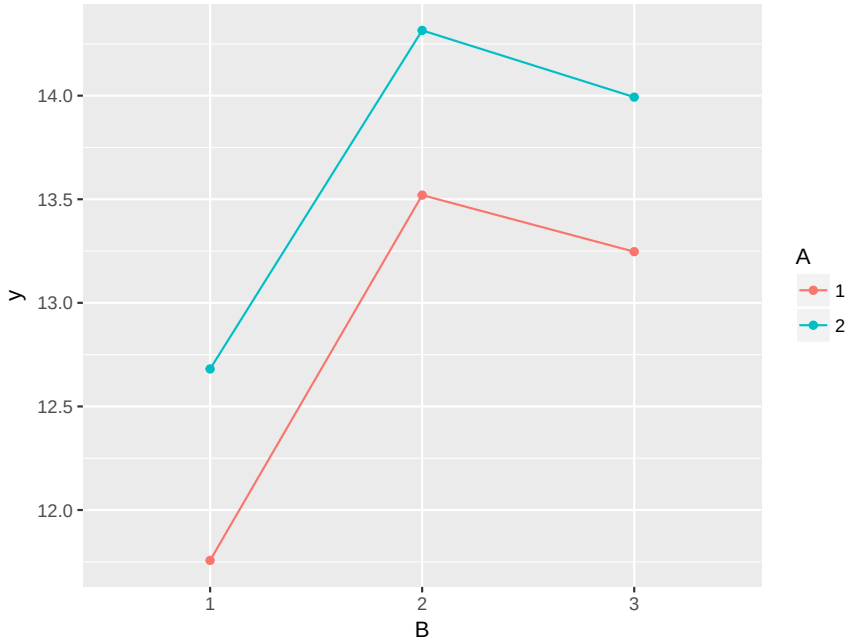
again, everything looks fine.

Notice that the ANOVA table also has the test for the Subject means. This is not very interesting, the boxplot already makes it clear that different subjects have very different hearing abilities. If that were not so, we would eliminate Subject and run a oneway ANOVA. Because we now have two factors, we need to worry about an additional problem, namely whether or not there is a relationship between the two factors. This is called

30.1 Interaction

To do so we will check the **interaction plot**

An interaction plot looks as follows:

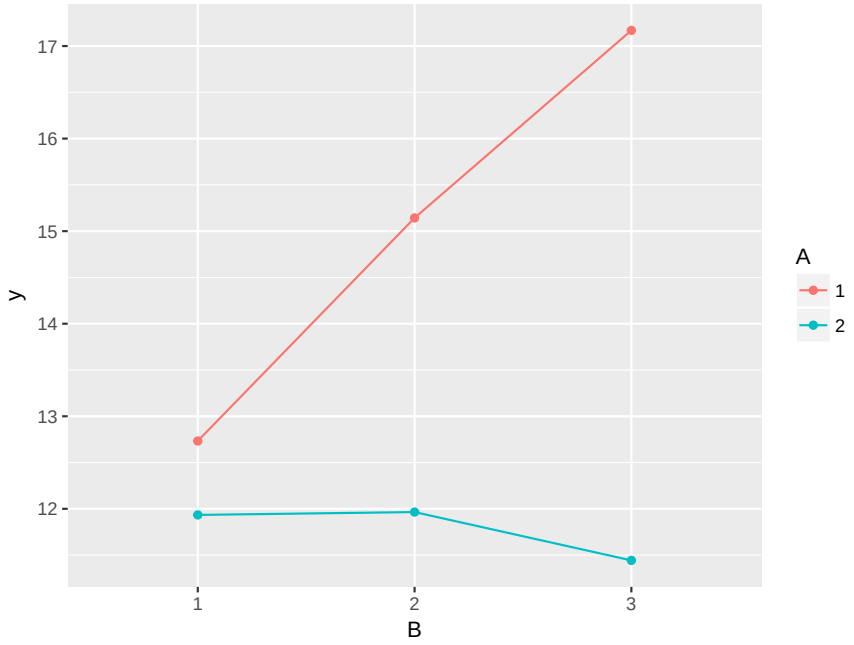


Here the line segments are allmost parallel. This implies that for any value of the factor A going from one value of B to the next adds **the same** amount to the response. So if we go

from B=1 to B=2 **both** lines move up by about 2.0, and if we go from B=2 to B=3 **both** lines move down by 0.75.

Because of this we call such a model **additive**

Now consider the following interactions plot:



Here as we go from B=2 to B=3 the line goes up by 4 **if A=1** but it goes down by 0.5 **if A=1**.

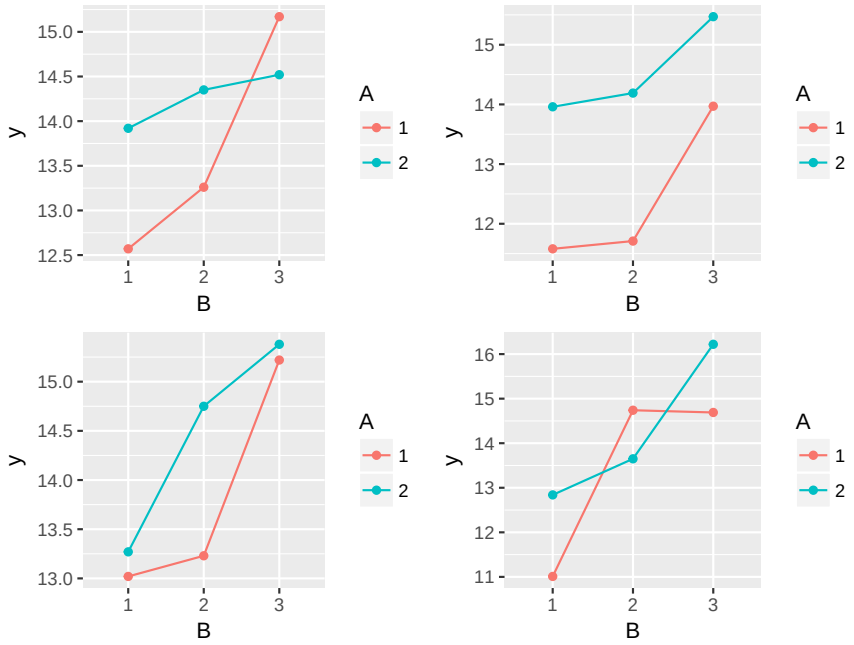
Here is another way of understanding the difference: Say you are told that you have an additive model and the following information:

	Low	High
In	2.3	3.1
Out	2.7	?

Can we make a guess for the response if Factor 1 = "high" and Factor 2 = "out"? We see that if Factor 2 = "in" and going from "low" to "high" the response goes up by 0.4 (=2.7-2.3). In an additive model that means the response should go up the **same amount** for Factor 2 = "out", that is it should go to 3.5 (=3.1+0.4).

But if there were interaction there would be no way to make any guess at all!

Deciding from the graph whether or not there is interaction is not always easy. Here are four interaction plots from a simulated data set, all guaranteed NOT to have any interaction:

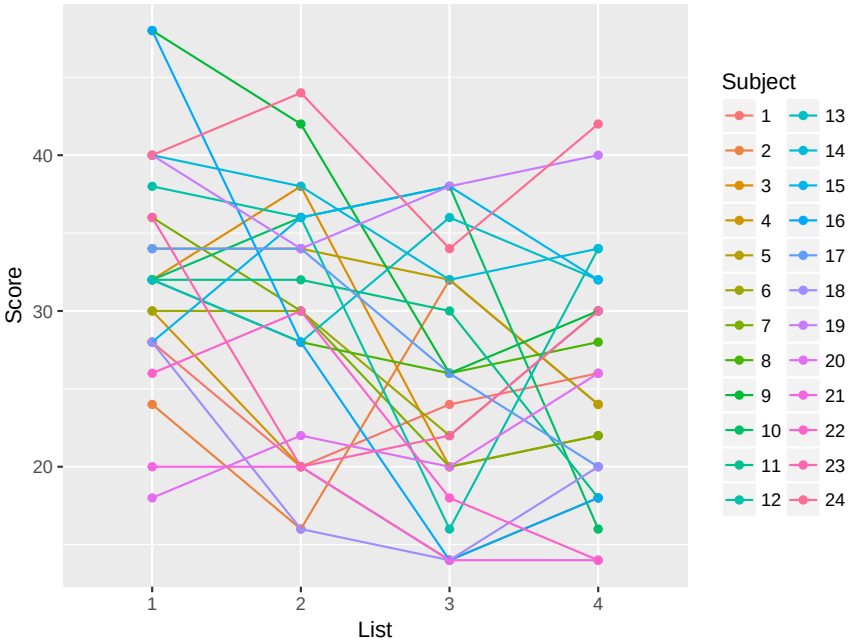


This is even worse because in ANOVA problems we often have very small data sets, so there is a great amount of variation in these graphs from sample to sample.

So it would be nice if we could actually test for interaction, but that requires **repeated measurements**.

In the hearing aid data we only have one observation for each combination of Subject and List, so we need to decide on the basis of the interaction plot:

```
iplot(Score, List, Subject )
```



There seems to be interaction between Lists and Subjects

Finally, it would of course be interesting to study just which lists are different, that is we could do a **multiple comparison**:

```
tukey(Score, List, Subject, which="first")
```

```
## No repeated measurement!
## Interaction term can not be included
##
## Groups that are statistically significantly different:
```

```
##   Groups p.value
## 1    1-3       0
## 2    1-4       0
```

so List 1 is statistically sigifcantly different from Lists 3 and 4.

No other differences are statistically significant.

Because Subject is only a blocking variable we would not due a multiple comparison for it if we wanted to we would use the command

```
tukey(Score, List, Subject, which="second")
```

```
## No repeated measurement!
## Interaction term can not be included
##
## Groups that are statistically significantly different:
```

```
##    Groups p.value
## 1   15-21  0.0394
## 2   14-21  0.0063
## 3    9-21  0.0042
## 4   19-21  0.0012
## 5   24-21  0.0000
## 6   14-18  0.0394
## 7    9-18  0.0279
## 8   19-18  0.0093
## 9   24-18  0.0019
## 10   19-4  0.0195
## 11   24-4  0.0042
## 12  19-20  0.0394
## 13  24-20  0.0093
## 14  24-22  0.0135
```

30.1.1 Case Study: Gasoline Type and Milage

In an experiment to study gas milage four different blends of gasoline are tested in each of three makes of automobiles. The cars are driven a fixed distance to determine the mpg (miles per gallon) The experiment is repeated three times for each blend-automobile combination. (Taken from Lyman Ott)

Note that the interest here is indifferent gasoline blends, automobile is a blocking variable, so this is a randomized block design.

Gasoline is numbers, but these are just codes for different blends, so it is a categorical variable or factor.

```
attach(gasoline)
head(gasoline)
```

```
##    MPG Gasoline Automobile
## 1 22.7        1          A
## 2 22.4        1          A
## 3 22.9        1          A
## 4 21.5        2          A
## 5 21.8        2          A
## 6 21.6        2          A
```

Here is an interesting calculation:

```
table(Gasoline, Automobile)
```

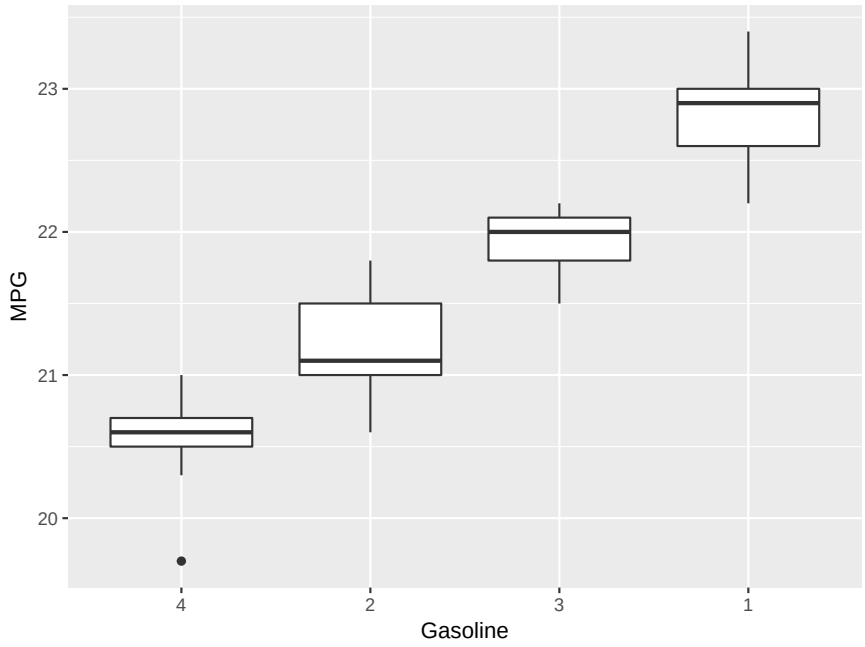
```
##          Automobile
## Gasoline A B C
##        1 3 3 3
##        2 3 3 3
##        3 3 3 3
##        4 3 3 3
```

This shows us two things:

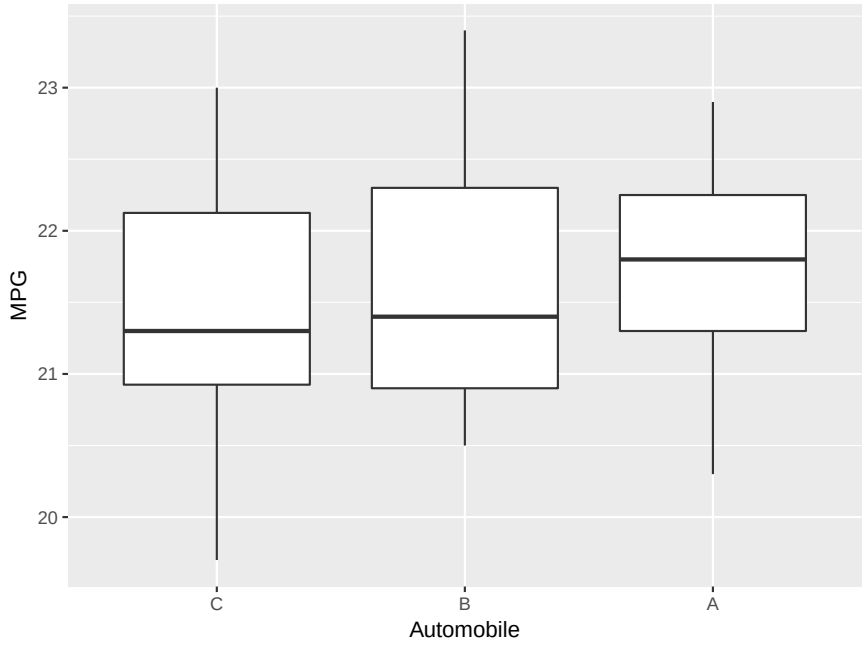
1. we have *repeated measurements* (several observations per factor-level combination)
2. we have a *balanced design* (the same number of repetitions in each factor-level combination)

This second feature used to be quite important because the calculations in a balanced design are much simpler. Nowadays with fast computers this is not important anymore. There are still good reasons why you want to design your experiment to have a balanced design if possible, though!

```
bplot(MPG, Gasoline, new_order = "Size")
```




```
bplot(MPG, Automobile, new_order = "Size")
```



the boxplots suggest a difference between blends but not between automobiles.

The summary statistics are

```
stat.table(MPG, Gasoline, Sort = TRUE)
```

```
##   Sample Size Mean Standard Deviation
## 4           9 20.5                0.4
## 2           9 21.2                0.4
```

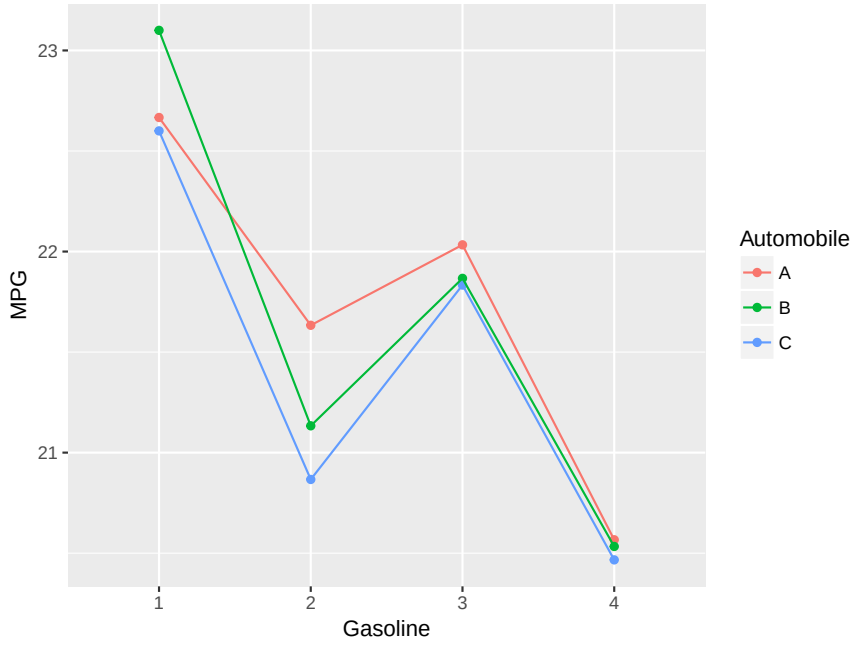
```
## 3           9 21.9                0.2
## 1           9 22.8                0.4
```

```
stat.table(MPG, Automobile, Sort = TRUE)
```

```
##   Sample Size Mean Standard Deviation
## C          12 21.4                0.9
## B          12 21.7                1.0
## A          12 21.7                0.8
```

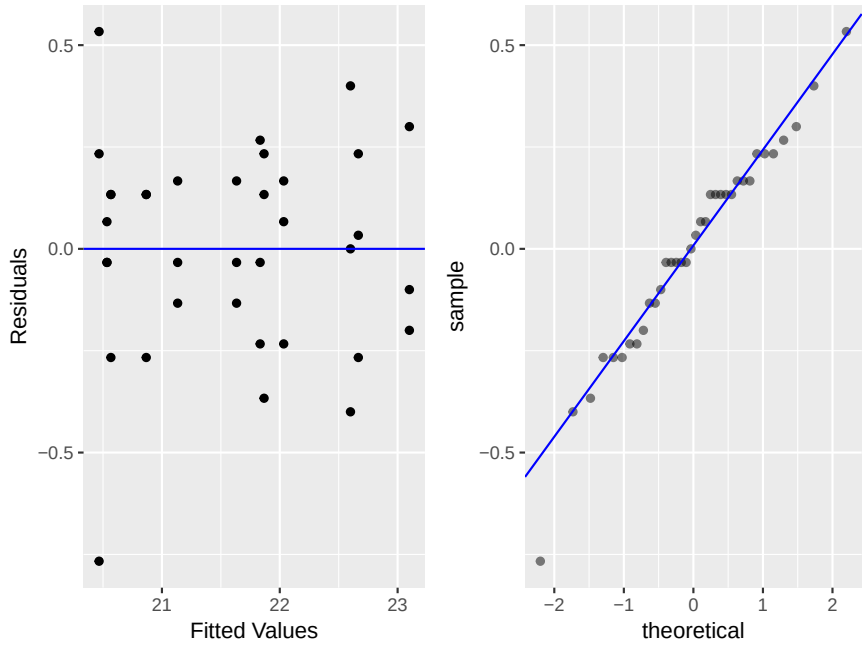
Interaction:

```
iplot(MPG, Gasoline, Automobile)
```



Lines are (almost) parallel, so there is no indication of interaction. We have **repeated measurements** (3 per factor-level combination), so we can test for this:

```
twoway(MPG, Gasoline, Automobile)
```




```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            3 25.405   8.468  90.464 3.21e-13
## z            2  0.527   0.263   2.813   0.0799
## x:z          6  0.909   0.151   1.618   0.1854
## Residuals   24  2.247   0.094
##                    [,1]
## Gasoline  p =   0.0000
## Automobile  p = 0.0799
## Interaction p = 0.1854
```

1) Parameters of interest: Interaction

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

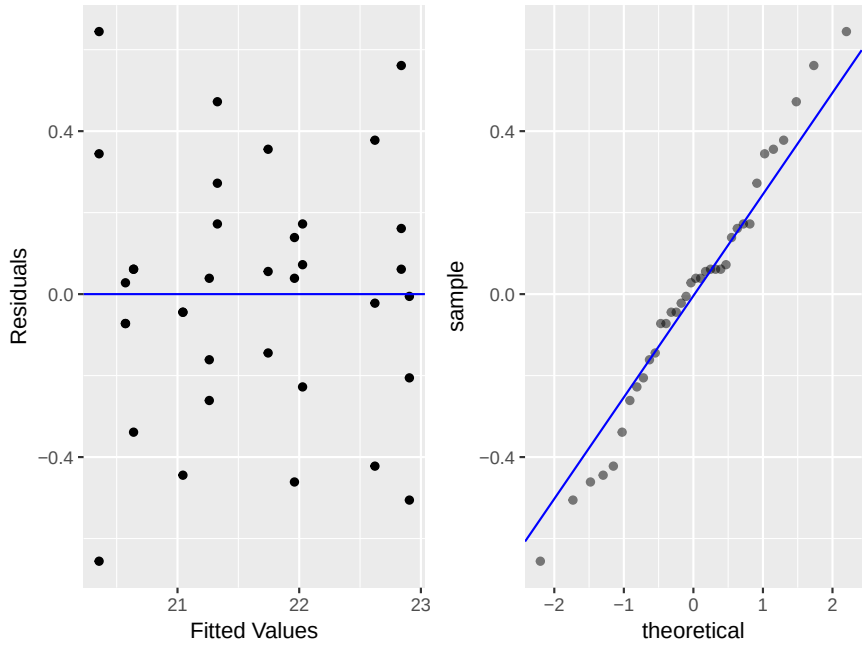
4) Type I error probability α=0.05

5) Null hypothesis H_0 : no interaction

6) Alternative hypothesis H_a: some interaction

7) p value = 0.1854
8) 0.1854 > 0.05, there is no evidence of interaction So we will now proceed without the interaction term

```
twoway(MPG, Gasoline, Automobile, with.interaction=FALSE)
```




```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            3 25.405   8.468  80.510 1.89e-14
## z            2  0.527   0.263   2.504   0.0987
## Residuals   30  3.156   0.105
##                   [,1]
## Gasoline  p =   0.0000
## Automobile  p = 0.0987
```

the plots look fine, so no problem with the assumptions.

Now let's test for the factors:

Test for Factor Gasoline:

1) Parameters of interest: means of gasoline groups

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis $H_0 : \mu_1 = .. = \mu_4$ (Gasoline groups have the same means)

6) Alternative hypothesis H_a: $\mu_i \neq \mu_j$ (Gasoline groups have different means)

7) p value=0.000

8) 0.000<0.05, there is some evidence of differences in gasoline blends

Test for Factor Automobile is not really needed because this is a blocking variable.

Notice that if we included the interaction the p-value for Automobile was 0.08, without the interaction it is 0.1. One advantage of being able to fit an additive model is that often it makes the conclusions stronger.

Because automobile is not significant and there is no interaction, we can drop automobile from the analysis and treat this as a oneway ANOVA problem:

```
tukey(MPG, Gasoline)
```

```
## Groups that are statistically significantly different:

##   Groups p.value
## 1    2-4       0
## 2    3-4       0
## 3    1-4       0
## 4    3-2       0
## 5    1-2       0
## 6    1-3       0
```

so all blends are stat. significantly different, with blend 1 having the highest miles per gallon.

30.1.2 Case Study: Film Thickness in Semiconductor Production

Chemical vapor deposition is a process used in the semiconductor industry to deposit thin films of silicon dioxide and photoresit on substrates of wafers as they are manufactured. The films must be as thin as possible and have a uniform thickness, which is measured by a process called infrared interference. A process engineer wants to evaluate a low-pressure chemical vapor deposition process that reduces costs and increases productivity. The engineer has set up an experiment to study the effect of chamber temperature and pressure on film thickness.

```
attach(filmcoatings)
filmcoatings
```

```
##    Thickness Temperature Pressure
## 1         42         Low      Low
## 2         43         Low      Low
## 3         39         Low      Low
## 4         45         Low      Mid
## 5         43         Low      Mid
## 6         45         Low      Mid
## 7         45         Low     High
## 8         44         Low     High
## 9         47         Low     High
## 10        36         Mid      Low
## 11        34         Mid      Low
## 12        37         Mid      Low
## 13        39         Mid      Mid
## 14        39         Mid      Mid
## 15        37         Mid      Mid
```

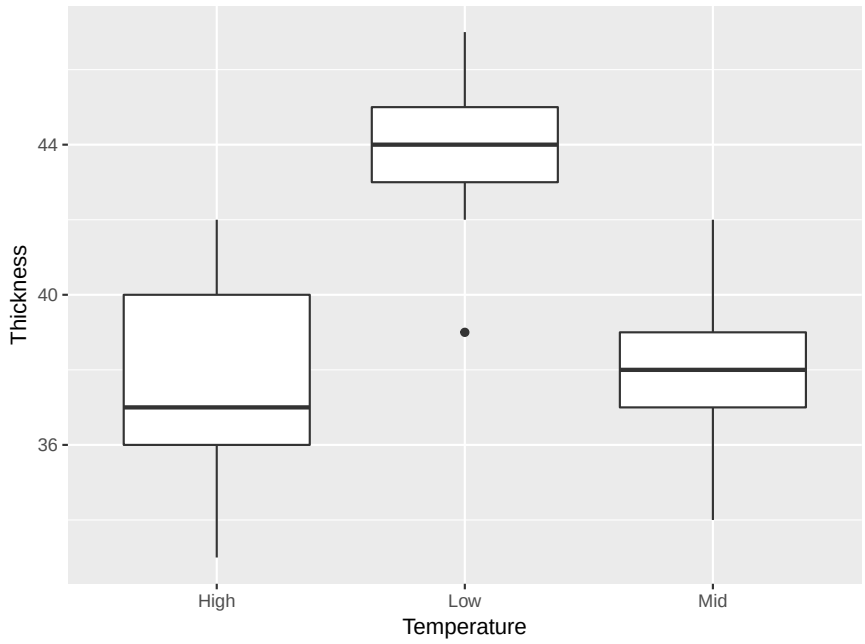
```
## 16         40          Mid     High
## 17         42          Mid     High
## 18         38          Mid     High
## 19         38         High      Low
## 20         37         High      Low
## 21         37         High      Low
## 22         35         High      Mid
## 23         36         High      Mid
## 24         33         High      Mid
## 25         40         High     High
## 26         41         High     High
## 27         42         High     High
```

```
table(Temperature, Pressure)
```

```
##             Pressure
## Temperature High Low Mid
##        High    3   3   3
##        Low     3   3   3
##        Mid     3   3   3
```

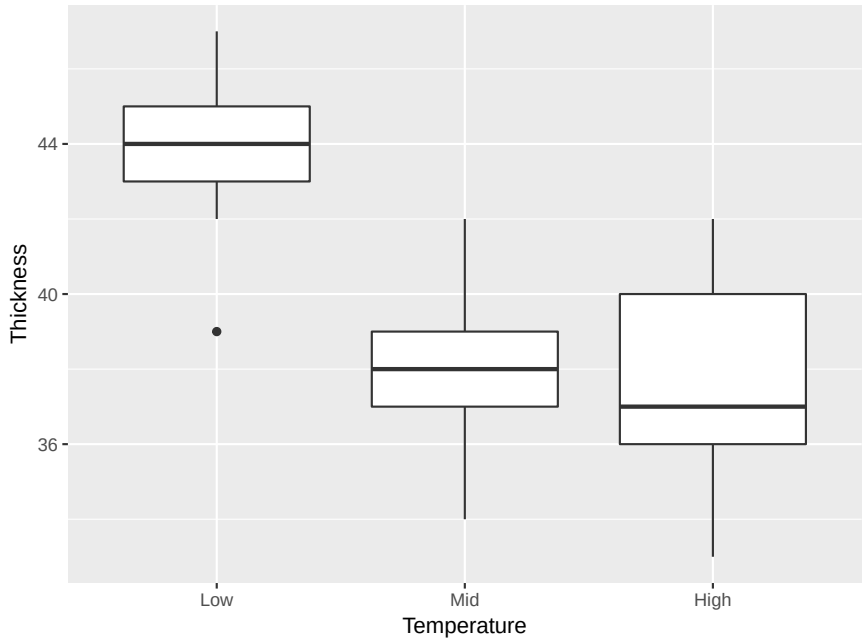
so again we have balanced design with repetead measurements

```
bplot(Thickness, Temperature)
```

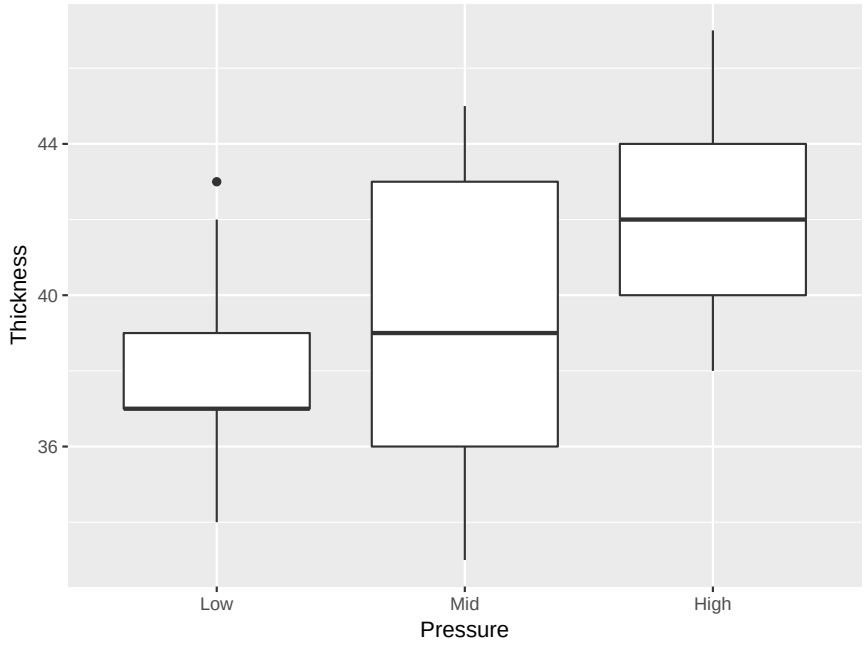


Notice that the order of the boxes is strange: High-Low-Mid. This is because R uses alphabetic ordering unless told otherise. Let's change that:

```
Temperature <- change.order(Temperature, c("Low","Mid","High"))
Pressure <- change.order(Pressure, c("Low","Mid","High"))
bplot(Thickness, Temperature)
```




```
bplot(Thickness, Pressure)
```



Unlike in the hearing aid or gasoline experiments, here we equally interested in both factors. This type of experiment is called a **factorial design** problem.

For us there is no practical difference between a randomized block design and a factorial design but the distinction can be important in other analyses.

```
stat.table(Thickness, Temperature)
```

```
##      Sample Size Mean Standard Deviation
```

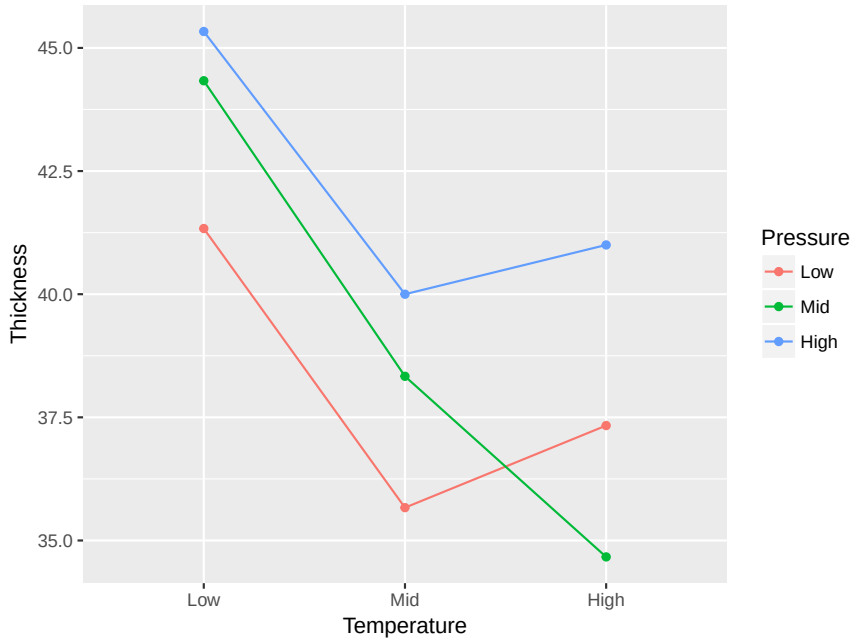
```
## Low             9 43.7                2.3
## Mid             9 38.0                2.3
## High            9 37.7                2.9
```

```
stat.table(Thickness, Pressure)
```

```
##      Sample Size Mean Standard Deviation
## Low             9 38.1                2.8
## Mid             9 39.1                4.4
## High            9 42.1                2.8
```

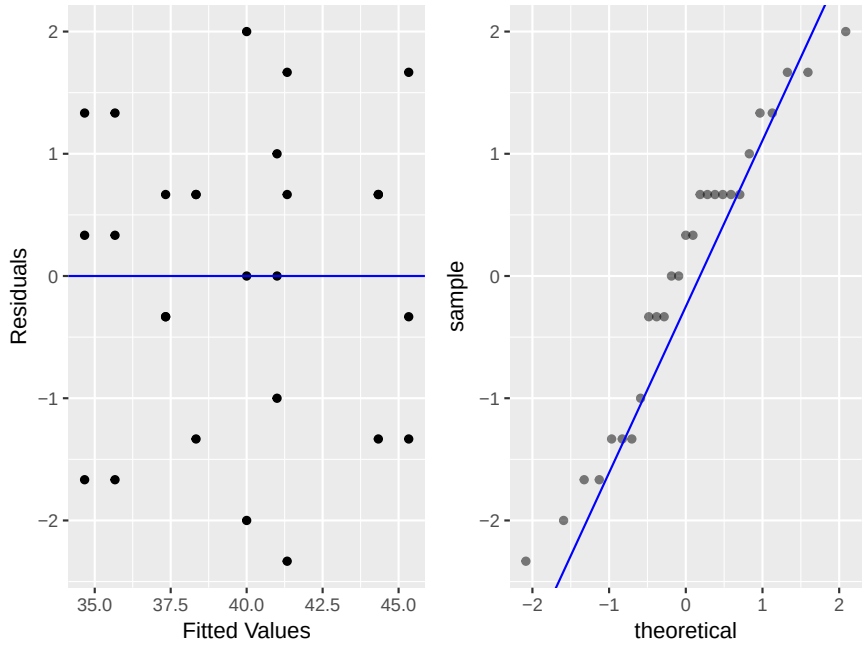
Interaction

```
iplot(Thickness, Temperature, Pressure)
```



The lines are not all parallel, so there is likely some interaction. Again we have **repeated measurements** (3 per factor-level combination), so we can actually test for this:

```
twoway(Thickness, Temperature, Pressure)
```




```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            2 204.67  102.33  47.638 6.46e-08
## z            2  78.00   39.00  18.155 4.83e-05
## x:z          4  37.33    9.33   4.345   0.0124
## Residuals   18  38.67    2.15
##                     [,1]
## Temperature  p = 0.0000
## Pressure  p =    0.0000
## Interaction p =  0.0124
```

1) Parameters of interest: Interaction

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis H_0 : no interaction

6) Alternative hypothesis H_a: some interaction

7) p value = 0.0124

8) 0.0124<0.05, there is some evidence of interaction
the graphs show that there are no problems with the assumptions

Test for Factor Temperature:

1) Parameters of interest: means of temperature groups

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3$ (Temperature groups have the same means)

6) Alternative hypothesis H_a: $\mu_i \neq \mu_j$ (Temperature groups have different means)

7) p value = 0.000

8) $0.000 < 0.05$, there is some evidence of differences in temperature
Test for Factor Pressure:

9) Parameters of interest: means of pressure groups

10) Method of analysis: ANOVA

11) Assumptions of Method: residuals have a normal distribution, groups have equal variance

12) Type I error probability $\alpha = 0.05$

13) Null hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3$ (Pressure groups have the same means)

14) Alternative hypothesis H_a: $\mu_i \neq \mu_j$ (Pressure groups have different means)

15) p value = 0.000

16) 0.000<0.05, there is some evidence of differences in pressure

Finally, what we need is to find the best combination of pressure and temperature. So what we want is a multiple comparison for Temperature and Pressure (not either of them alone!). Easily done:

```
tukey(Thickness, Temperature, Pressure, which="interaction")
```

```
## Groups that are statistically significantly different:
```

```
##                 Groups p.value
## 1    Low:Mid-High:Low  0.0000
## 2   Low:High-High:Low  0.0000
```

```
## 3     Low:Low-Mid:Low  0.0041
## 4     Low:Mid-Mid:Low  0.0000
## 5   High:High-Mid:Low  0.0072
## 6    Mid:High-Mid:Low  0.0397
## 7    Low:High-Mid:Low  0.0000
## 8    High:Mid-Low:Low  0.0000
## 9    Low:Mid-High:Mid  0.0000
## 10 High:High-High:Mid  0.0013
## 11  Mid:High-High:Mid  0.0072
## 12  Low:High-High:Mid  0.0000
## 13    Low:Mid-Mid:Mid  0.0023
## 14   Low:High-Mid:Mid  0.0000
## 15   Mid:High-Low:Mid  0.0397
## 16 Low:High-High:High  0.0397
## 17  Low:High-Mid:High  0.0072
```

This is bit hard to read. In the past we have used the tapply command to sort the groups by their means. We want to do the same here but first we need to make a new variable that combines the Temperature and the Pressure:

```
TP <- paste0(Temperature,"-",Pressure)
TP
```

```
##  [1] "Low-Low"   "Low-Low"   "Low-Low"   "Low-Mid"   "Low-Mid"
##  [6] "Low-Mid"   "Low-High"  "Low-High"  "Low-High"  "Mid-Low"
## [11] "Mid-Low"   "Mid-Low"   "Mid-Mid"   "Mid-Mid"   "Mid-Mid"
## [16] "Mid-High"  "Mid-High"  "Mid-High"  "High-Low"  "High-Low"
## [21] "High-Low"  "High-Mid"  "High-Mid"  "High-Mid"  "High-High"
## [26] "High-High" "High-High"
```

```
sort(round(tapply(Thickness,TP,mean), 1))
```

```
##  High-Mid   Mid-Low  High-Low   Mid-Mid  Mid-High High-High   Low-Low
##      34.7      35.7      37.3      38.3      40.0      41.0      41.3
##   Low-Mid  Low-High
##      44.3      45.3
```

so we see that that the combination Temperature=High, Pressure=Mid is best.

But tukey tells us that it is not stat. significantly better than either of the next three combinations (Mid Low, High Low or Mid Mid), at least not at these sample sizes.

Remember we made some new variables. If we are sure we won't need them anymore we should

```
rm(Temperature)
rm(Pressure)
rm(TP)
```

A simple idea for solving this problem seems to be this one:

1. find the best temperature:

```
sort(round(tapply(Thickness, Temperature, mean), 1))
```

```
## High  Mid  Low
## 37.7 38.0 43.7
```

so Temperature=High is best

2. find the best pressure:

```
sort(round(tapply(Thickness, Pressure, mean), 1))
```

```
##  Low  Mid High
## 38.1 39.1 42.1
```

so Pressure=Low is best

3. take the combination: Pressure=Low, Temperature=High is best! Except it is not: we saw before that Pressure=Mid, Temperature=High is best.

This simple idea does not work because of the presence of interaction.

30.1.3 Case Study: Water Quality and Mining

The effects of mining and rock type on water quality.

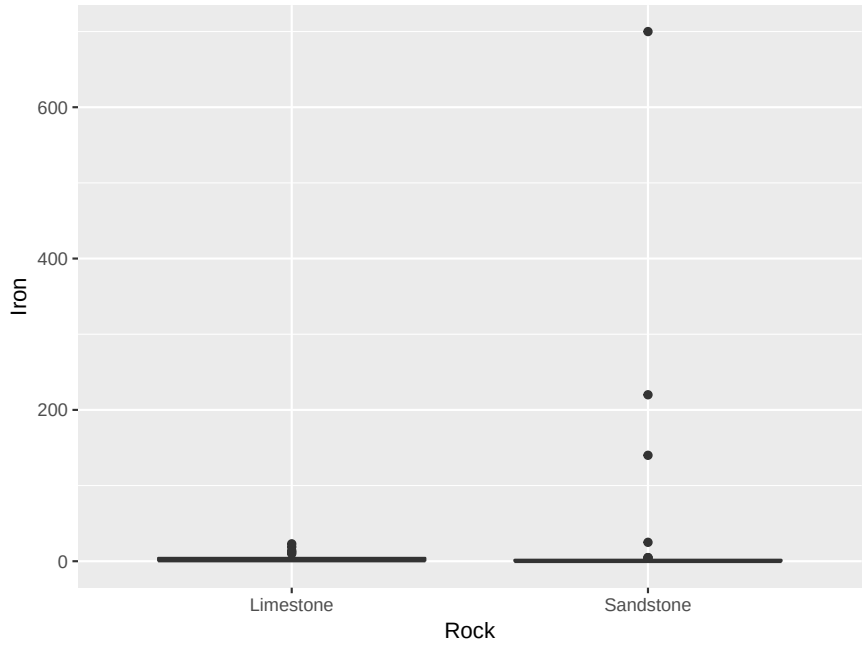
```
attach(mines)
head(mines)
```

```
##        Rock    Mine Iron
## 1 Sandstone Unmined 0.20
## 2 Sandstone Unmined 0.25
## 3 Sandstone Unmined 0.04
## 4 Sandstone Unmined 0.06
## 5 Sandstone Unmined 1.20
## 6 Sandstone Unmined 0.30
```

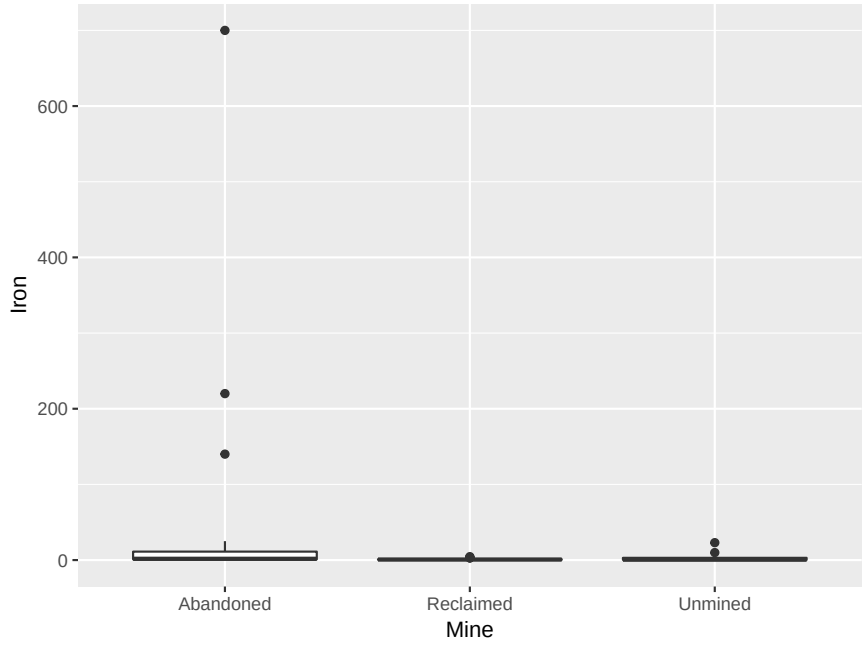
```
table(Rock, Mine)
```

```
##            Mine
## Rock        Abandoned Reclaimed Unmined
##   Limestone        13        13      13
##   Sandstone        13        13      13
```

```
bplot(Iron, Rock, new_order = "Size")
```

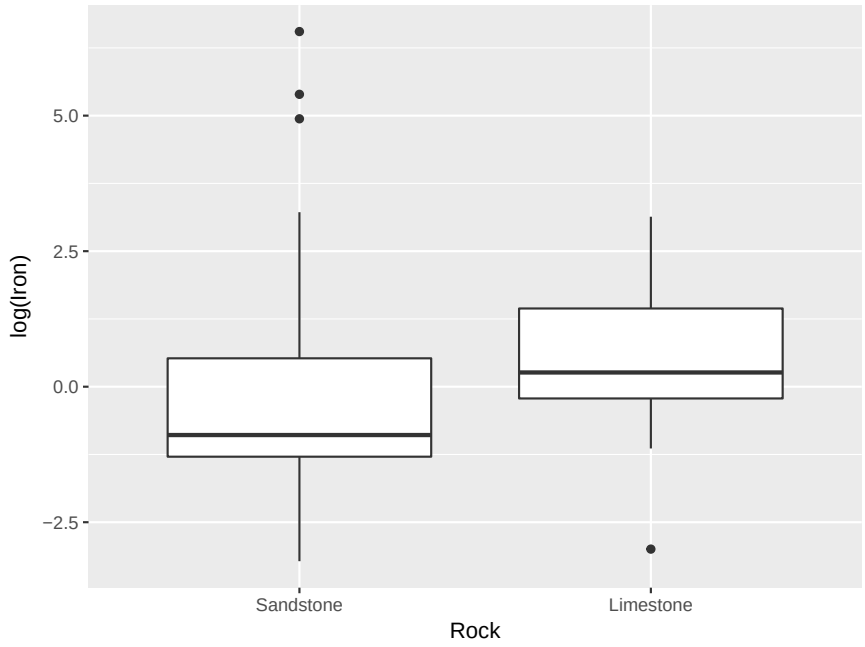



```
bplot(Iron, Mine)
```

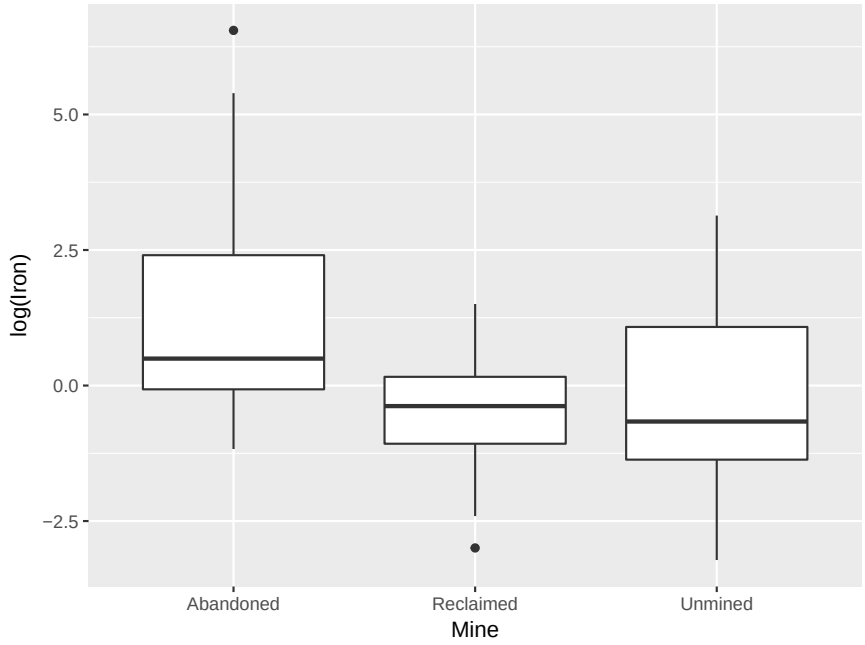


We have a clear problem with the normal assumption, so use the log transform

```
bplot(log(Iron), Rock, new_order = "Size")
```




```
bplot(log(Iron), Mine)
```



This has solved the problem, so the analysis will be based on log(Iron)

Summary Statistics

Because we use a transformation we will base the tables on Median and IQR

```
stat.table(Iron, Rock, Mean=FALSE, Sort = TRUE)
```

```
##           Sample Size Median IQR
## Sandstone          39    0.4 1.4
```

```
## Limestone          39    1.3 3.6
```

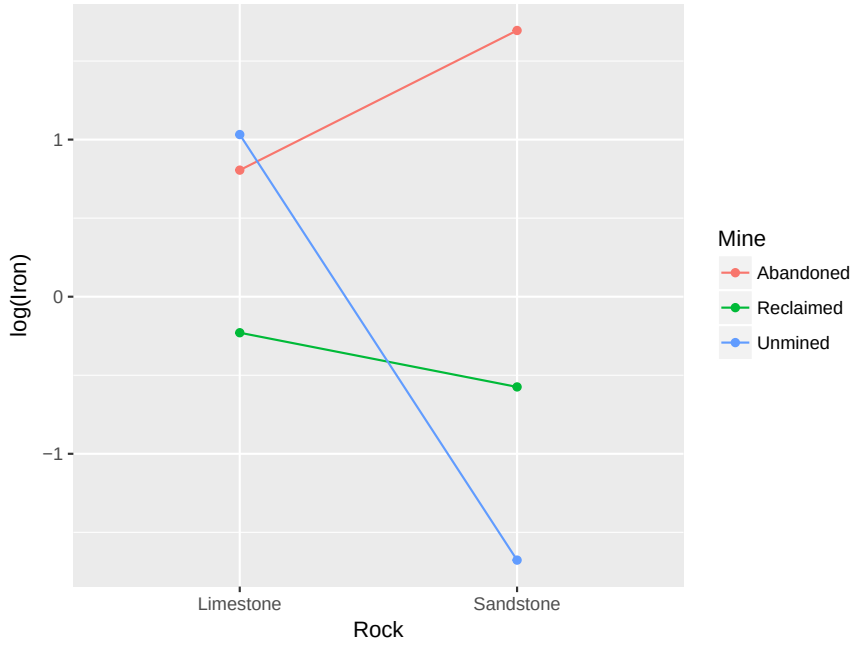
```
stat.table(Iron, Mine, Mean=FALSE)
```

```
##           Sample Size Median  IQR
## Unmined            26    0.5  2.7
## Reclaimed          26    0.7  0.8
## Abandoned          26    1.6 10.2
```

Note that the IQR's are very different. This is because this data set has a lot of outliers which still effect the IQR.

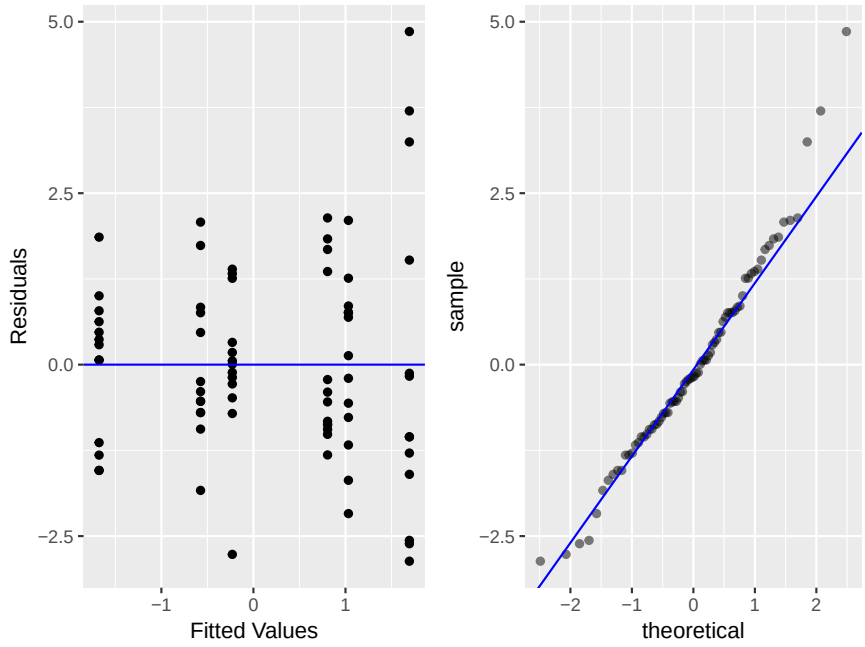
Interaction

```
iplot(log(Iron), Rock, Mine)
```



There seems to be some interaction. To confirm this test for it:

```
twoway(log(Iron), Rock, Mine)
```




```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## x             1  10.15  10.154   4.629 0.034794
## z             2  45.13  22.566  10.287 0.000118
## x:z           2  43.45  21.727   9.904 0.000159
## Residuals    72 157.95   2.194
##                     [,1]
## Rock  p =       0.0348
## Mine  p =       0.0000
## Interaction p = 0.0000
```

1) Parameters of interest: Interaction

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis H_0 : no interaction

6) Alternative hypothesis H_a: some interaction

7) p value = 0.000

8) 0.000<0.05, there is some evidence of interaction
   Check the assumptions of ANOVA: both plots look ok

Test for Factor Rock:

1) Parameters of interest: means of pressure groups

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis H_0 : $\mu_1 = \mu_2$ (Rock groups have the same means)

6) Alternative hypothesis H_a: $\mu_1 \neq \mu_2$ (Rock groups have different means)

7) p value = 0.035

8) 0.035<0.05, there is some evidence of differences in Rock types.

Test for Factor Mine:

1) Parameters of interest: means of pressure groups

2) Method of analysis: ANOVA

3) Assumptions of Method: residuals have a normal distribution, groups have equal variance

4) Type I error probability $\alpha = 0.05$

5) Null hypothesis H_0 : $\mu_1 = \mu_2 = \mu_3$ (Mine groups have the same means)

6) Alternative hypothesis H_a: $\mu_i \neq \mu_j$ (Mine groups have different means)

7) p value = 0.000

8) 0.000<0.05, there is some evidence of differences in Mine types

Multiple Comparison The main interest is in mines, so

```
tukey(log(Iron), Rock, Mine, which="second")
```

```
## Groups that are statistically significantly different:
```

```
##                 Groups p.value
## 1 Abandoned-Reclaimed       0
## 2   Abandoned-Unmined       0
```

Interpretation: There is a stat. signif. difference between the mean iron content of abondoned mines and the others. The difference between unmined and reclaimed mines is not stat. sign, at least not at these sample sizes.

31 More than One Quantitative Predictor

31.0.1 Case Study: House Prices

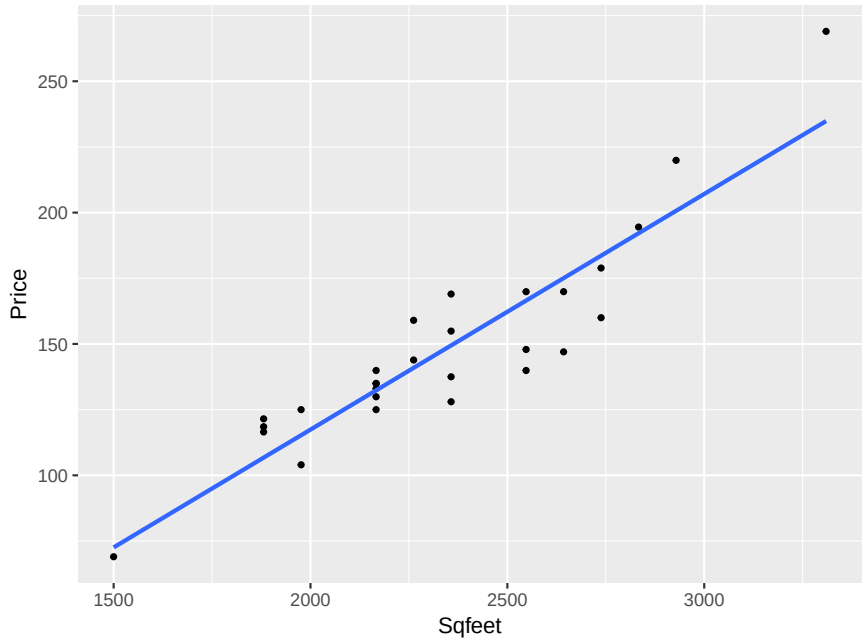
Prices of residencies located 30 miles south of a large metropolitan area with several possible predictor variables. Notice the 1.7 baths!

```
attach(houseprice)
houseprice
```

```
##    Price   Sqfeet Floors Bedrooms Baths
## 1   69.0 1500.000      1        2   1.0
## 3  118.5 1880.952      1        2   2.0
## 4  104.0 1976.190      1        3   2.0
## 5  116.5 1880.952      1        3   2.0
## 6  121.5 1880.952      1        3   2.0
## 7  125.0 1976.190      1        3   2.0
## 8  128.0 2357.143      2        3   2.5
## 9  129.9 2166.667      1        3   1.7
## 10 133.0 2166.667      2        3   2.5
## 11 135.0 2166.667      2        3   2.5
## 12 137.5 2357.143      2        3   2.5
## 13 139.9 2166.667      1        3   2.0
## 14 143.9 2261.905      2        3   2.5
## 15 147.9 2547.619      2        3   2.5
## 16 154.9 2357.143      2        3   2.5
## 17 160.0 2738.095      2        3   2.0
## 18 169.0 2357.143      1        3   2.0
## 19 169.9 2642.857      1        3   2.0
## 20 125.0 2166.667      1        4   2.0
## 21 134.9 2166.667      1        4   2.0
## 22 139.9 2547.619      1        4   2.0
## 23 147.0 2642.857      1        4   2.0
## 24 159.0 2261.905      1        4   2.0
## 25 169.9 2547.619      2        4   3.0
## 26 178.9 2738.095      1        4   2.0
## 27 194.5 2833.333      2        4   3.0
## 28 219.9 2928.571      1        4   2.5
## 29 269.0 3309.524      2        4   3.0
```

Let's go through the list of predictors one by one:

```
splot(Price, Sqfeet, add.line = 1)
```

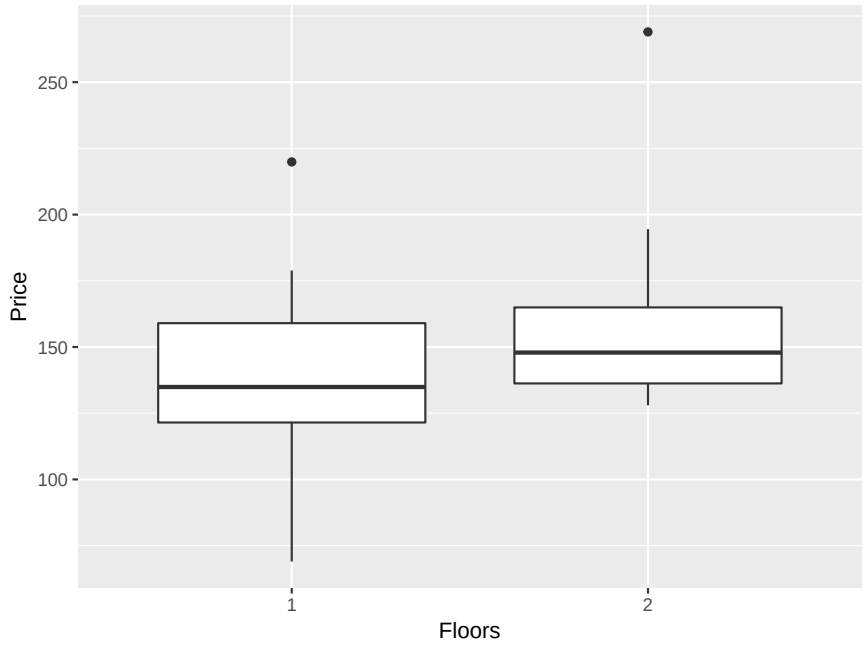



```
cor(Price, Sqfeet)
```

```
## [1] 0.9152079
```

strong positive relationship (r=0.915), could be linear

```
bplot(Price, Floors)
```



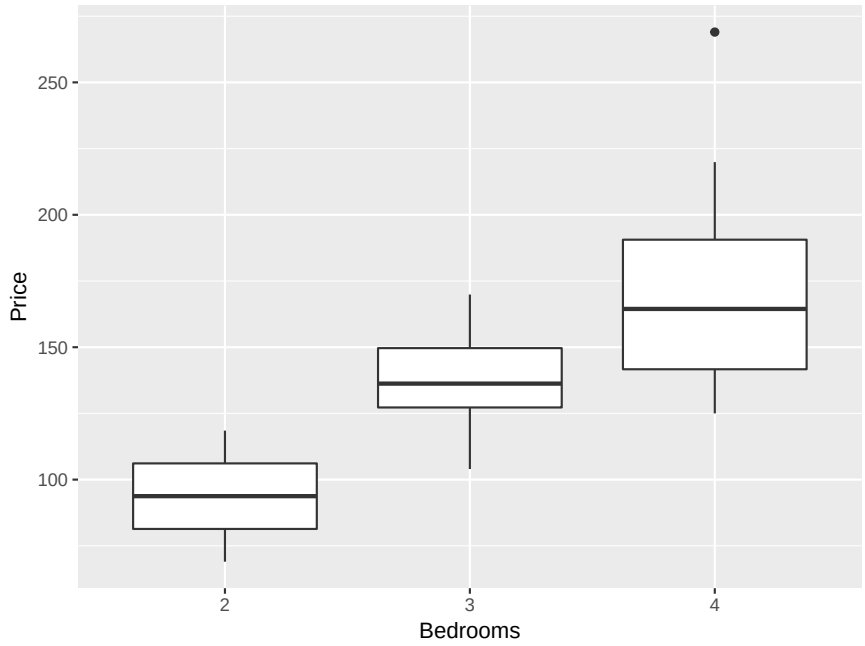

```
cor(Price, Floors)
```

```
## [1] 0.2910762
```

weak if any positive relationship (r=0.291)

Note we used the boxplot here although Floors is a quantitative predictor. If the predictor has only a few different values (2 here!) this is often a better choice.

```
bplot(Price, Bedrooms)
```

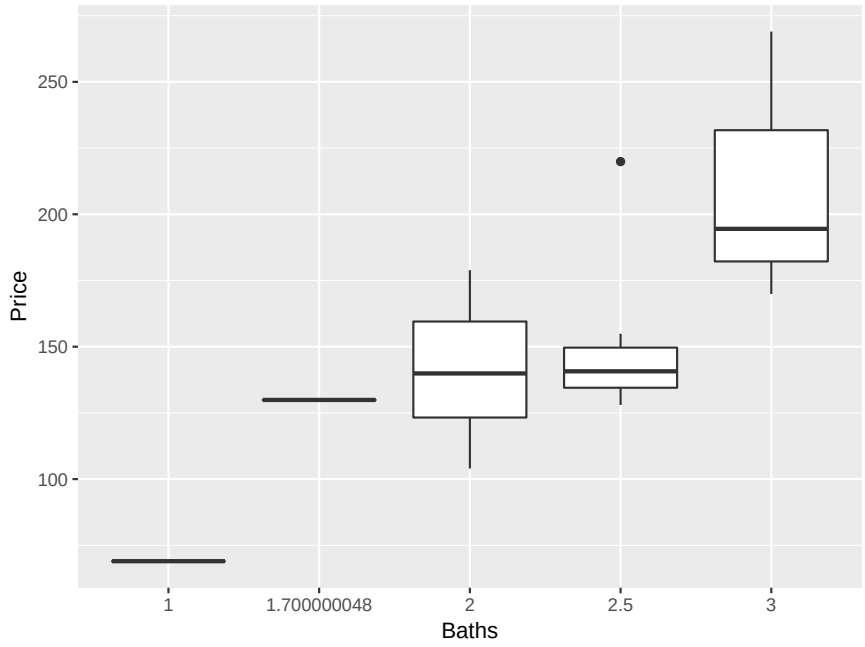



```
cor(Price, Bedrooms)
```

```
## [1] 0.6045036
```

strong positive relationship (r=0.605), could be linear

```
bplot(Price, Baths)
```




```
cor(Price, Baths)
```

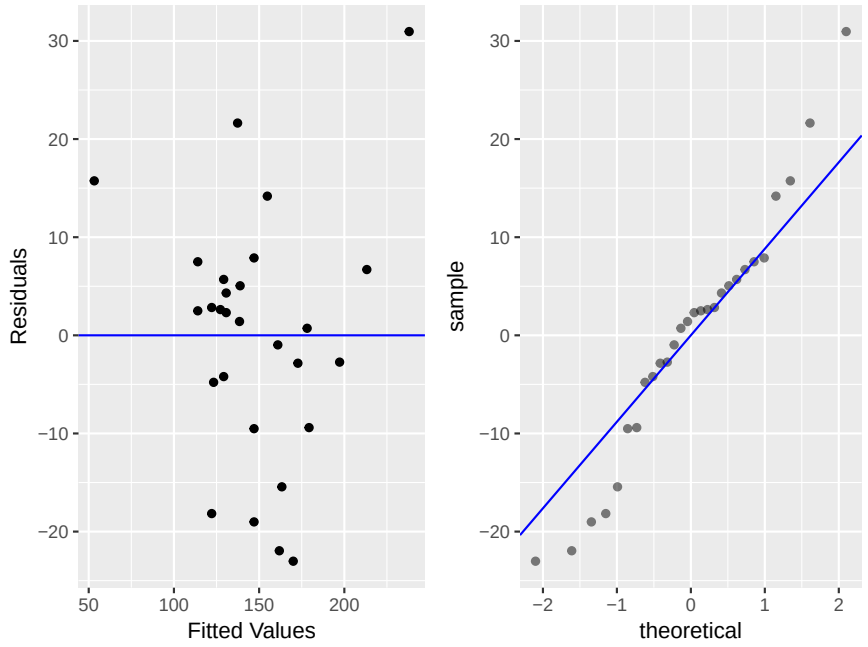
```
## [1] 0.6525626
```

strong positive relationship (r=0.652), could be linear

Note there is so far no mention of regression, residual vs. fits plot or normal plot. Making decisions about possible transformations and/or polynomial models early solely based on scatterplots and/or boxplots is usually a good idea.

Now to run the regression we have the routine *mlr*. As always the first argument is the response variable but now the second argument are all the predictors as a matrix:

```
mlr(Price, houseprice[, -1])
```




```
## The least squares regression equation is:
##  Price  =  -67.62 + 0.086 Sqfeet - 26.493 Floors - 9.286 Bedrooms + 37.381 Baths
## R^2 = 88.6%
```

For the assumptions there is nothing new, as before we need to check the residual vs. fits plot and the normal plot of residuals.

This appears to be a good model and the assumptions of normally distributed residuals with equal variance appears to be o.k.

Except,

Notice that there is something very strange about this model!

```
round(cor(houseprice[,-1]),3)
```

```
##           Sqfeet Floors Bedrooms Baths
## Sqfeet     1.000  0.370    0.652 0.628
## Floors     0.370  1.000   -0.018 0.743
## Bedrooms   0.652 -0.018    1.000 0.415
## Baths      0.628  0.743    0.415 1.000
```

The highest correlation between predictors is r=0.743 (Floors-Baths)

31.1 Best Subset Regression

We have previously talked about the fact that we want our models to be as simple as possible. Often that means a model with as few predictors as possible. So the question becomes:

Can we eliminate any of our predictors without making the model (stat. signif.) worse?

There are several things one can think of:

Choose based on R^2

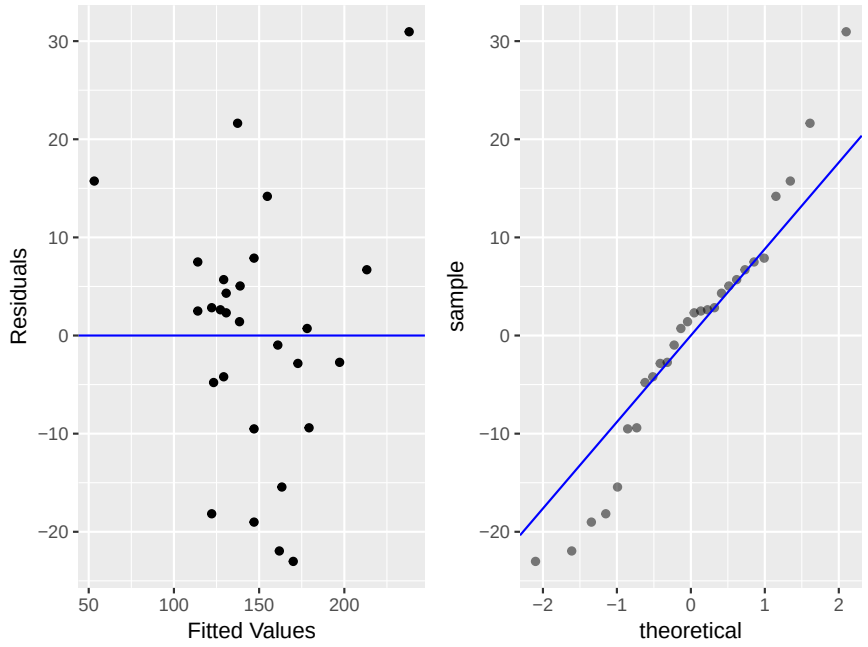
but we already know this will always lead to the model with all predictors, for the same reason that a cubic model always has an R^2 at least as high as the quadratic model.

Note:

Price by Sqfeet, Floors and Bedrooms: R^2=80.1%
Price by Floors, Bedrooms and Baths: R^2=68.4%
Price by Sqfeet, Bedrooms and Baths: R^2=83.5%
Price by Sqfeet, Floors, Bedrooms and Baths: R^2=88.2%
so model with all 4 has a higher R^2 than any of the models with just 3, **but this will always be so, even if one of the predictors is completely useless**.

Choose based on Hypothesis Tests

```
mlr(Price, houseprice[, -1], show.tests = TRUE)
```




```
## The least squares regression equation is:
##  Price  =  -67.62 + 0.086 Sqfeet - 26.493 Floors - 9.286 Bedrooms + 37.381 Baths
##  Variable p value
##  Constant   0.000
##    Sqfeet   0.000
##    Floors  0.0104
##  Bedrooms  0.1871
##     Baths  0.0057
## R^2 = 88.6%
```

so p_value of Bedrooms > 0.05, so eliminate Bedrooms.

This sounds like a good idea AND IT IS WIDELY USED IN REAL LIFE, but it turns out

to be a bad one ! The reason why is bit hard to explain, though.

Use nested models test

```
all.preds <- mlr(Price, houseprice[, -1], return.model = TRUE)
without.bedrooms <- mlr(Price, houseprice[, -c(1, 4)],
                        return.model = TRUE)
nested.models.test(all.preds, without.bedrooms)
```

```
## H0: both models are equally good.
## p value= 0.1871
```

Again, this sounds like a good idea AND AGAIN IT IS WIDELY USED IN REAL LIFE, but it turns out to be a dangerous one! To start, if we have several predictors we might want to eliminate, we immediately face the issue of *simultaneous inference*.

There are several methods in wide use that are essentially based on this idea, such as *forward selection*, *backward selection* and *stepwise regression*. These are sometimes unavoidable but need to be done with great care!

What we need is new idea:

Best Subset Regression and Mallow's C_p

We will find ALL possible models and calculate Mallow's C_p statistic for each. The model with the lowest C_p is best.

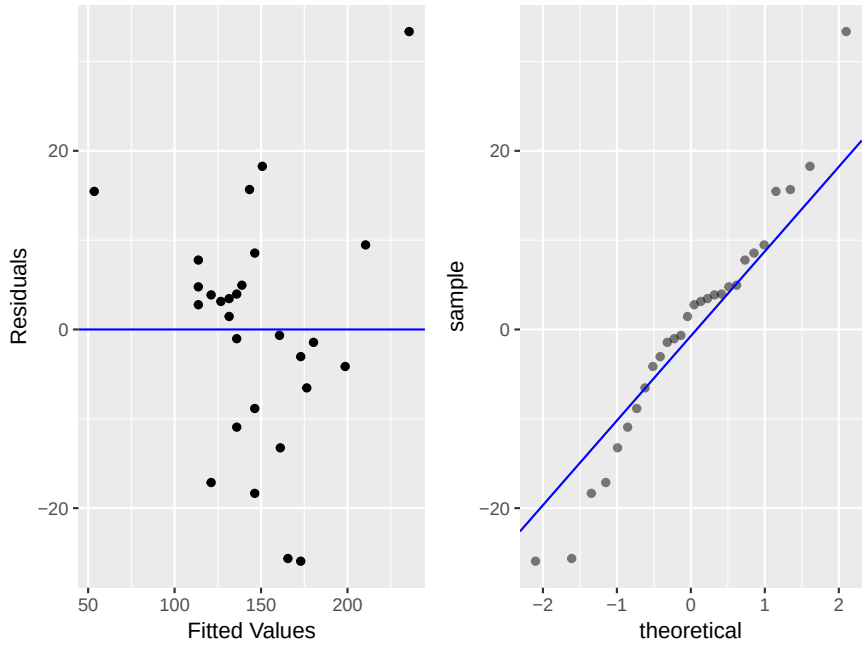
```
mallows(Price, houseprice[, -1])
```

```
##  Number of Variables Cp   Sqfeet Floors Bedrooms Baths
##  1                   8.83 X
##  2                   8.81 X                     X
##  3                   4.85 X      X              X
##  4                   5    X      X      X       X
```

so the best model uses Sqfeet, Floors and Baths.

To find the model we rerun mlr, now without Bedrooms:

```
mlr(Price, houseprice[, -c(1,4)])
```




```
## The least squares regression equation is:
##  Price  =  -73.93 + 0.078 Sqfeet - 19.68 Floors + 30.579 Baths
## R^2 = 87.7%
```

Note that the model with all four predictors has C_p=5.0. But C_p is a **statistic**, its exact value depends on the sample. So is the model with Sqfeet, Floors and Baths **statistically significantly** better than the model with all four predictors? We would need a hypothesis test to answer this question but this is not part of our course.

For more on Mallow's C_p see page 603 of the textbook.

31.1.0.1 Prediction

Prediction works just as it did for simple regression. We have the command **mlr.predict**. Say we want to find a 90% interval estimate for a house that has 2000 sqfeet, one floor and two baths. Then

```
mlr.predict(Price, houseprice[ ,-c(1, 4)],
    newx=c(2000, 1, 2), interval="PI", conf.level=90)
```

```
##  Sqfeet Floors Baths    Fit Lower  Upper
##    2000      1     2 122.98 98.07 147.89
```

and so a 90% predicition interval is ($98,070, $147,890)

If we want to do prediction for a number of cases newx has to be a matrix:

```
newx <- cbind(c(2000,2100,2200), rep(1,3), rep(2,3))
newx
```

```
##      [,1] [,2] [,3]
## [1,] 2000    1    2
```

```
## [2,] 2100    1    2
## [3,] 2200    1    2
```

```
mlr.predict(Price, houseprice[ ,-c(1, 4)],
            newx=newx, interval="PI")
```

```
##  Sqfeet Floors Baths    Fit  Lower  Upper
##    2000      1     2 122.98  92.93 153.03
##    2100      1     2 130.75 100.96 160.54
##    2200      1     2 138.52 108.86 168.18
```

Finally, not including the newx does the prediction of the dataset:

```
head(mlr.predict(Price, houseprice[ ,-c(1, 4)],
                 interval="PI"))
```

```
##    Sqfeet Floors Baths    Fit Lower  Upper
##  1500.000      1     1  53.54 18.63  88.46
##  1880.952      1     2 113.73 83.23 144.23
##  1976.190      1     2 121.13 91.01 151.25
##  1880.952      1     2 113.73 83.23 144.23
##  1880.952      1     2 113.73 83.23 144.23
##  1976.190      1     2 121.13 91.01 151.25
```

31.1.1 Case study: Air Pollution and Mortality

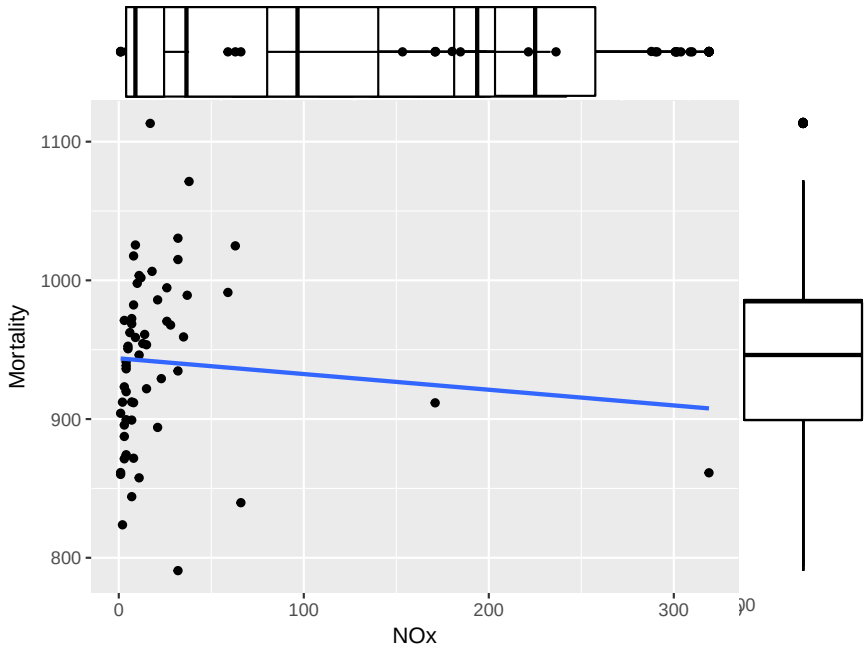
The dependent variable for analysis is age adjusted mortality (called "Mortality"). The data include variables measuring demographic characteristics of the cities, variables measuring climate characteristics, and variables recording the pollution potential of three different air pollutants.

```
attach(airpollution)
head(airpollution)
```

```
##                           Mortality JanTemp JulyTemp RelHum Rain Education
## AkronOH                      921.87      27       71     59   36      11.4
## Albany-Schenectady-TroyNY    997.87      23       72     57   35      11.0
## AllentownBethlehemPA-NJ      962.35      29       74     54   44       9.8
## AtlantaGA                    982.29      45       79     56   47      11.1
## BaltimoreMD                 1071.29      35       77     55   43       9.6
## BirminghamAL                1030.38      45       80     54   53      10.2
##                           PopDensity NonWhite WhiteCollar     Pop
## AkronOH                         3243      8.8        42.6  660328
## Albany-Schenectady-TroyNY       4281      3.5        50.7  835880
## AllentownBethlehemPA-NJ         4260      0.8        39.4  635481
## AtlantaGA                       3125     27.1        50.2 2138231
## BaltimoreMD                     6441     24.4        43.7 2199531
## BirminghamAL                    3325     38.5        43.1  883946
##                           Pop.House Income HCPot NOxPot SO2Pot NOx
```

```
## AkronOH                    3.34  29560    21     15     59  15
## Albany-Schenectady-TroyNY  3.14  31458     8     10     39  10
## AllentownBethlehemPA-NJ    3.21  31856     6      6     33   6
## AtlantaGA                  3.41  32452    18      8     24   8
## BaltimoreMD                3.44  32368    43     38    206  38
## BirminghamAL               3.45  27835    30     32     72  32
```

next we want to look at the scatterplots and the correlations. There are 15 predictors, so there are 15 graphs and correlations.



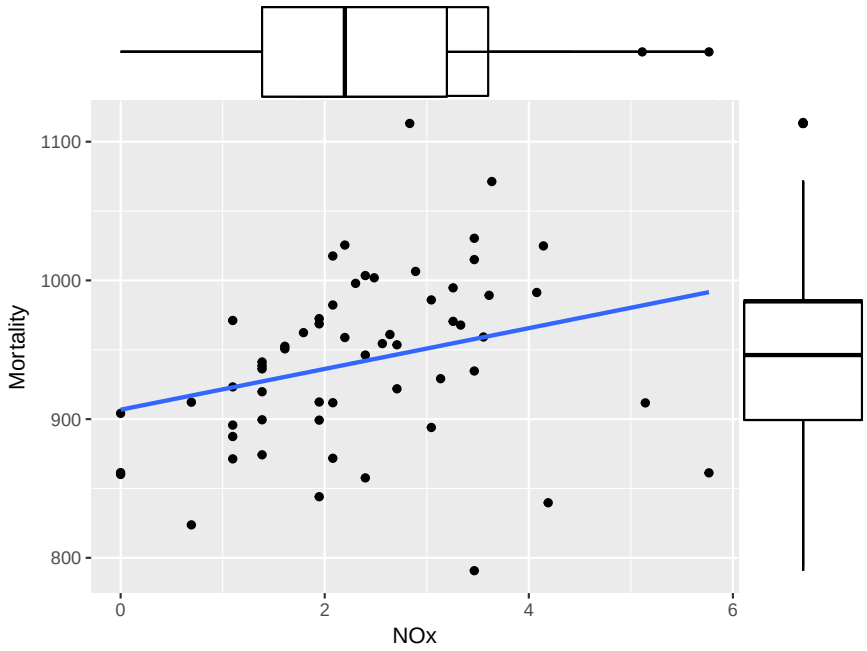

```
## [1] -0.016
## [1] 0.322
## [1] -0.101
## [1] 0.433
## [1] -0.508
## [1] 0.252
## [1] 0.647
## [1] -0.289
## Outliers! [1] 0.368
## [1] -0.283
## Outliers! Outliers! [1] 0.419
## Outliers!
```

There are problems with four predictors (Pop, HCPot, NOx, and NOxPot), Let's try the log transform and check again for those predictors:

The easiest way to do this is to make a new matrix:

```
detach(airpollution)
newair <- airpollution
newair[ ,c("Pop", "HCPot", "NOx", "NOxPot")] <- log(newair[, c("Pop", "HCPot", "NOx", "N
```

```
colnames(newair)[c(10, 13, 14, 16)] <- c("log(Pop)", "log(HCPot)", "log(NOx)", "log(NOxP
attach(newair)
for(i in c(10, 13, 14, 16)) {
   mplot(Mortality, newair[, i], add.line=1, varNames =  c(colnames(airpollution)[i], "N
   print(cor(Mortality, newair[, i]), 3)
}
```




```
## [1] 0.0853
## [1] 0.125
## [1] 0.28
## [1] 0.28
```

so in all cases the log transform worked, and we will use newair from now on.

Let's find the correlations in absolute value of the predictors with the response, in order:

```
cors <- round(cor(newair), 2)
sort(abs(cors[ ,"Mortality"]), decreasing = TRUE)[-1]
```

```
##    NonWhite   Education        Rain      SO2Pot   Pop.House    JulyTemp
##        0.65        0.51        0.43        0.42        0.37        0.32
## WhiteCollar      Income    log(NOx) log(NOxPot)  PopDensity  log(HCPot)
##        0.29        0.28        0.28        0.28        0.25        0.13
##      RelHum    log(Pop)     JanTemp
##        0.10        0.09        0.02
```

Next we look at the correlations between the predictors.

```
cors[-1, -1]
```

```
##             JanTemp JulyTemp RelHum  Rain Education PopDensity NonWhite
```

```
## JanTemp        1.00     0.32   0.09  0.06      0.11      -0.08     0.46
## JulyTemp       0.32     1.00  -0.44  0.47     -0.27      -0.01     0.60
## RelHum         0.09    -0.44   1.00 -0.12      0.19      -0.15    -0.12
## Rain           0.06     0.47  -0.12  1.00     -0.47       0.08     0.30
## Education      0.11    -0.27   0.19 -0.47      1.00      -0.24    -0.21
## PopDensity    -0.08    -0.01  -0.15  0.08     -0.24       1.00    -0.01
## NonWhite       0.46     0.60  -0.12  0.30     -0.21      -0.01     1.00
## WhiteCollar    0.21    -0.01   0.01 -0.11      0.49       0.25    -0.06
## log(Pop)       0.32     0.04  -0.02 -0.28      0.27       0.21     0.22
## Pop.House     -0.33     0.26  -0.14  0.20     -0.39      -0.17     0.35
## Income         0.20    -0.19   0.13 -0.36      0.51       0.00    -0.10
## log(HCPot)     0.23    -0.41   0.18 -0.48      0.18       0.26     0.15
## log(NOx)       0.18    -0.30   0.10 -0.39      0.03       0.34     0.21
## SO2Pot        -0.09    -0.07  -0.12 -0.13     -0.23       0.42     0.16
## log(NOxPot)    0.18    -0.30   0.10 -0.39      0.03       0.34     0.21
##             WhiteCollar log(Pop) Pop.House Income log(HCPot) log(NOx)
## JanTemp            0.21     0.32     -0.33   0.20       0.23     0.18
## JulyTemp          -0.01     0.04      0.26  -0.19      -0.41    -0.30
## RelHum             0.01    -0.02     -0.14   0.13       0.18     0.10
## Rain              -0.11    -0.28      0.20  -0.36      -0.48    -0.39
## Education          0.49     0.27     -0.39   0.51       0.18     0.03
## PopDensity         0.25     0.21     -0.17   0.00       0.26     0.34
## NonWhite          -0.06     0.22      0.35  -0.10       0.15     0.21
## WhiteCollar        1.00     0.28     -0.35   0.37       0.16     0.11
## log(Pop)           0.28     1.00     -0.26   0.41       0.48     0.50
## Pop.House         -0.35    -0.26      1.00  -0.30      -0.22    -0.12
## Income             0.37     0.41     -0.30   1.00       0.29     0.25
## log(HCPot)         0.16     0.48     -0.22   0.29       1.00     0.94
## log(NOx)           0.11     0.50     -0.12   0.25       0.94     1.00
## SO2Pot            -0.06     0.37     -0.01   0.07       0.57     0.68
## log(NOxPot)        0.11     0.50     -0.12   0.25       0.94     1.00
##             SO2Pot log(NOxPot)
## JanTemp      -0.09        0.18
## JulyTemp     -0.07       -0.30
## RelHum       -0.12        0.10
## Rain         -0.13       -0.39
## Education    -0.23        0.03
## PopDensity    0.42        0.34
## NonWhite      0.16        0.21
## WhiteCollar  -0.06        0.11
## log(Pop)      0.37        0.50
## Pop.House    -0.01       -0.12
## Income        0.07        0.25
## log(HCPot)    0.57        0.94
## log(NOx)      0.68        1.00
## SO2Pot        1.00        0.68
```

```
## log(NOxPot)   0.68        1.00
```

We find:

a) there are sizable correlations (for example cor(NonWhite,JulyTemp)=0.60)

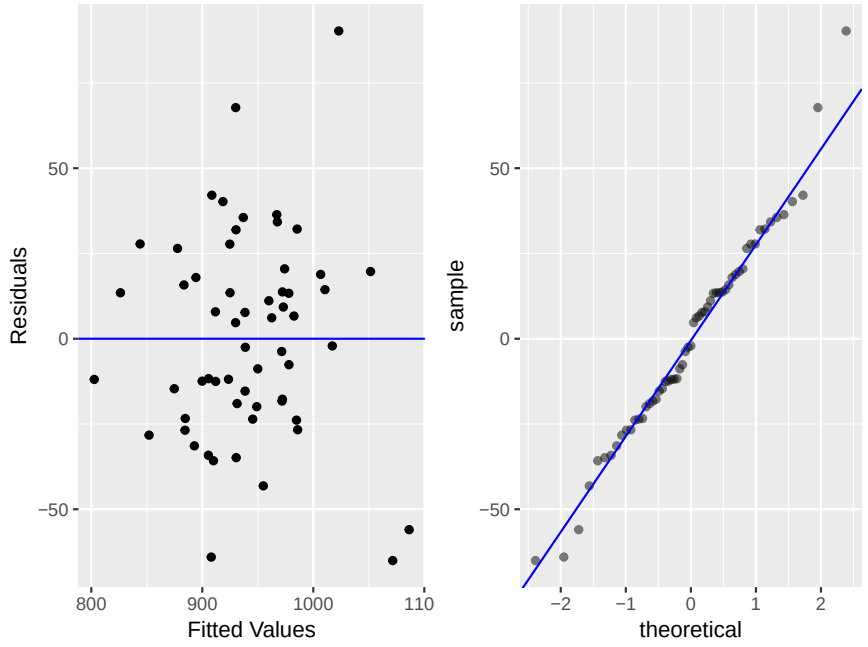
b) LOGT(NOxPot) and LOGT(NOx) are perfectly correlated. Because of a) unterpreting (understanding) the final model will be difficult.

Using perfectly correlated predictors is not possible so we eliminate one of them, say log(NOx):

```
detach(newair)
newair <- newair[ ,-16]
attach(newair)
```

Next we fit a model with all the predictors and check the assumptions:

```
mlr(Mortality , newair[, -1] )
```




```
## The least squares regression equation is:
##  Mortality  =  1230.17 - 1.885 JanTemp - 1.793 JulyTemp + 0.532 RelHum + 1.414 Rain -
## R^2 = 77%
```

The residual vs fits plot looks fine, so there is no problem with the model.

The normal plot is ok, so no problem with the nomal assumption. The residual vs fits plot looks fine, so there is no problem with the equal variance assumption.

Next we use the best subset regression to see whether we can find a model with fewer predictors.

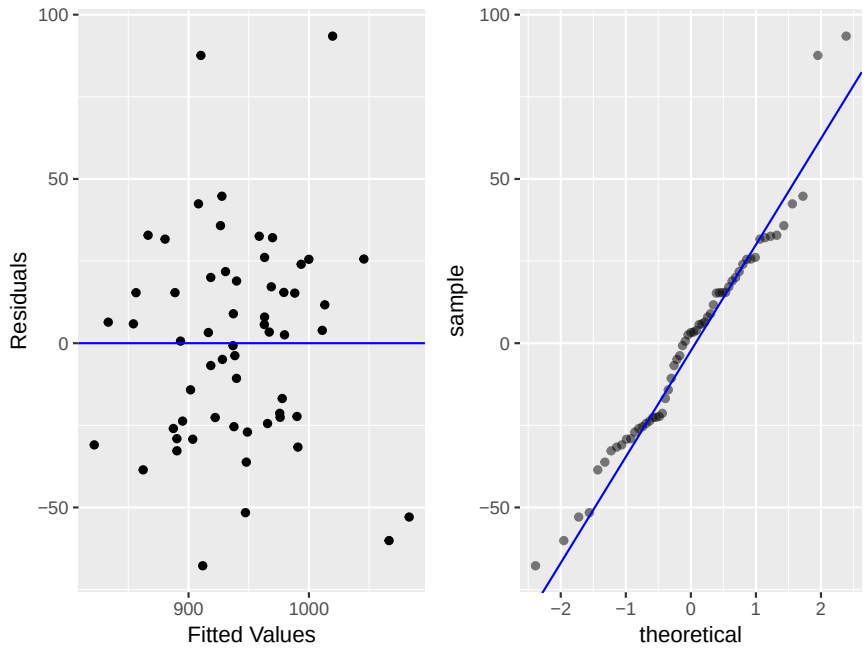
```
mallows(Mortality, newair[, -1])
```

```
##  Number of Variables Cp    JanTemp JulyTemp RelHum Rain Education
##  1                   56.46
```

```
##  2                  30.59                          X
##  3                  18.61 X                        X
##  4                  9.07  X                  X
##  5                  4.98  X                  X     X
##  6                  4.32  X                  X
##  7                  4.9   X                  X
##  8                  6.05  X       X          X
##  9                  6.98  X       X          X
##  10                 8.14  X       X          X     X
##  11                 9.55  X       X          X     X
##  12                 11.29 X       X    X     X     X
##  13                 13.14 X       X    X     X     X
##  14                 15    X       X    X     X     X
##  PopDensity NonWhite WhiteCollar log(Pop) Pop.House Income log(HCPot)
##             X
##             X
##             X
##             X
##             X
##  X          X        X
##  X          X        X                              X
##  X          X        X                              X
##  X          X        X                              X
##  X          X        X                   X          X
##  X          X        X                   X          X
##  X          X        X                   X          X
##  X          X        X                   X     X    X
##  X          X        X          X        X     X    X
##  log(NOx) SO2Pot
##
##
##
##  X
##  X
##  X
##  X
##  X
##  X        X
##  X
##  X        X
##  X        X
##  X        X
##  X        X
```

It suggests a model based on JanTemp, Rain, PopDensity, NonWhite, WhiteCollar and LOGT(NOx) with Mallow's C_p=4.32

```
mlr(Mortality,
    newair[, c("JanTemp", "Rain", "PopDensity", "NonWhite",
               "WhiteCollar", "log(NOx)")])
```




```
## The least squares regression equation is:
##  Mortality  =  944.275 - 1.942 JanTemp + 1.924 Rain + 0.006 PopDensity + 4.194 NonWhi
## R^2 = 74.2%
```

Because the best model does still include one of the pollution variables, we can conclude that pollution adds to the mortality rate.

And we are done!

32 How to include a Caterorical Predictor in a Regression

32.0.1 Case Study: Environmental, Safety and Health Attitudes

Environment, Safety and Health Attitudes of employees of a laboratory. Employees are given a questionaire, which is then collated into an average score from 1(bad) to 10(good). We also have available the length of service of the employee and their gender.

```
head(esh)
```

```
##   ES.H Yrs.Serv    Sex
## 1  7.6        5 Female
## 2  9.0       30 Female
## 3  8.0       12 Female
## 4  6.8        7 Female
## 5  7.4        7 Female
```

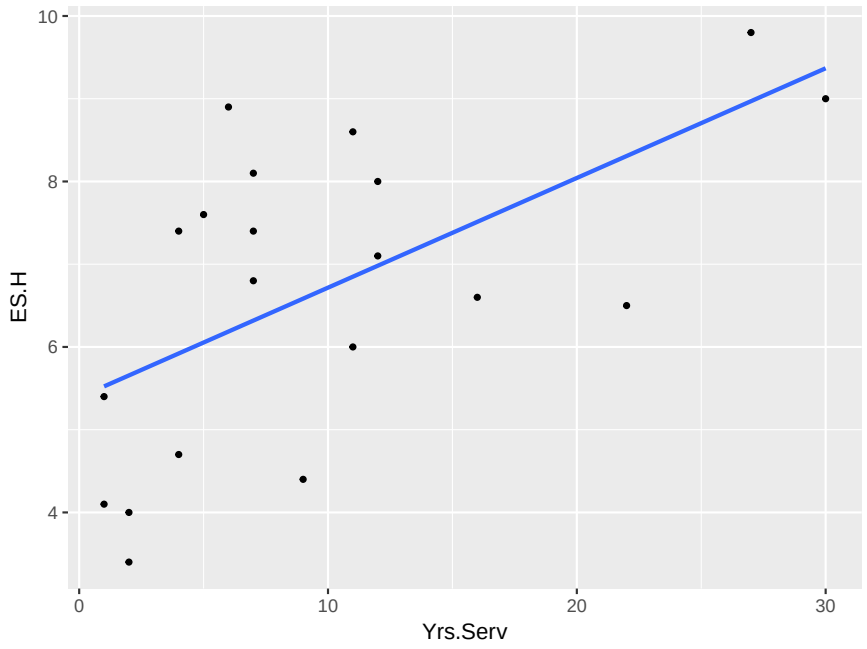
```
## 6  9.8       27 Female
```

One of the predictor variables (Sex) is actually categorical. A categorical variable used in a regression model is often refered to as a *dummy* variable.

Let's start by looking at each predictor separately.

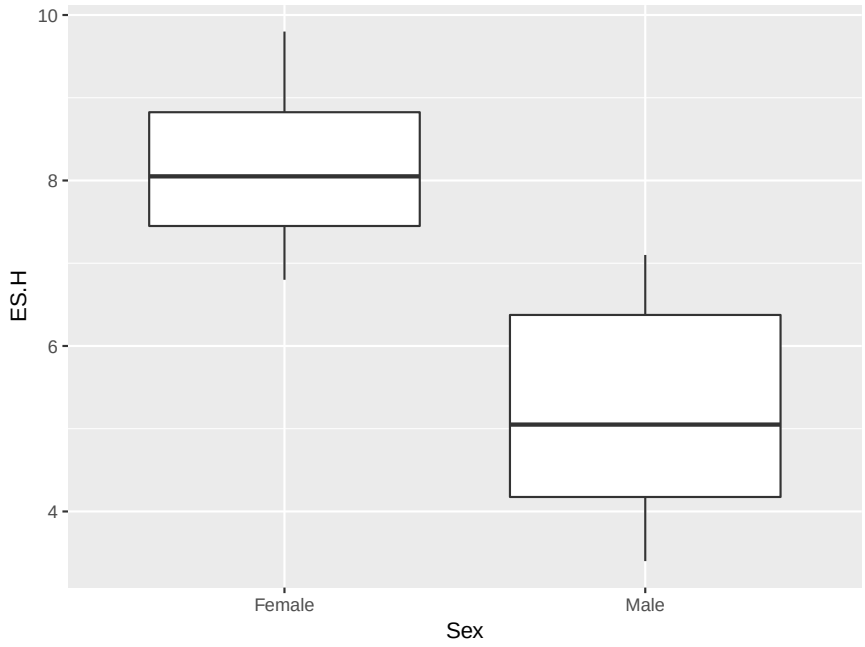
- Years is quantitative, so do the scatterplot:

```
attach(esh)
splot(ES.H, Yrs.Serv, add.line = 1)
```



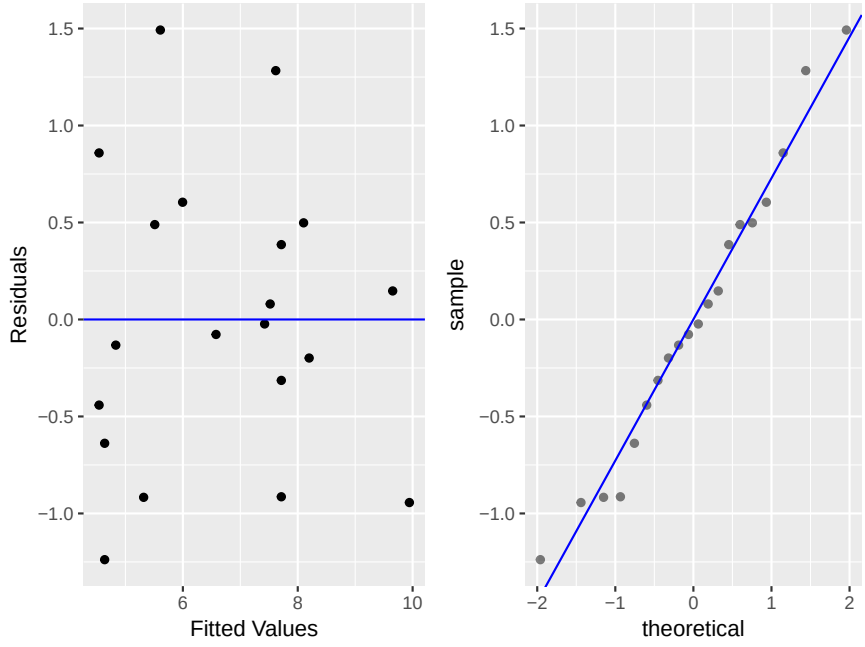
- Sex is categorical, so do the boxplot:

```
bplot(ES.H, Sex)
```



The values in Sex (Male, Female) are text but in a regression we need everything to be numeric, so in order to use Sex in a regression model we first have to *code* the variable as numbers, for example Female=0 and Male=1. Then

```
SexCode <- rep(0, length(Sex))
SexCode[Sex=="Male"] <- 1
X <- cbind(Yrs.Serv, SexCode)
mlr(ES.H, X)
```




```
## The least squares regression equation is:
##  ES.H  =  7.035 + 0.097 Yrs.Serv - 2.591 SexCode
```

```
## R^2 = 83.9%
```

The residual vs. fits and normal plot look good, so this is a good model.

Or is it?

Let's do the following: what would the equation look like if we knew the person was female? (or male). Well:

$$\begin{aligned}&\text{Female ES.H} = \\ &7.035 + 0.097\text{Yrs.Serv} - 2.591 \cdot 0 = \\ &7.035 + \mathbf{0.097}\text{Yrs.Serv}\end{aligned}$$

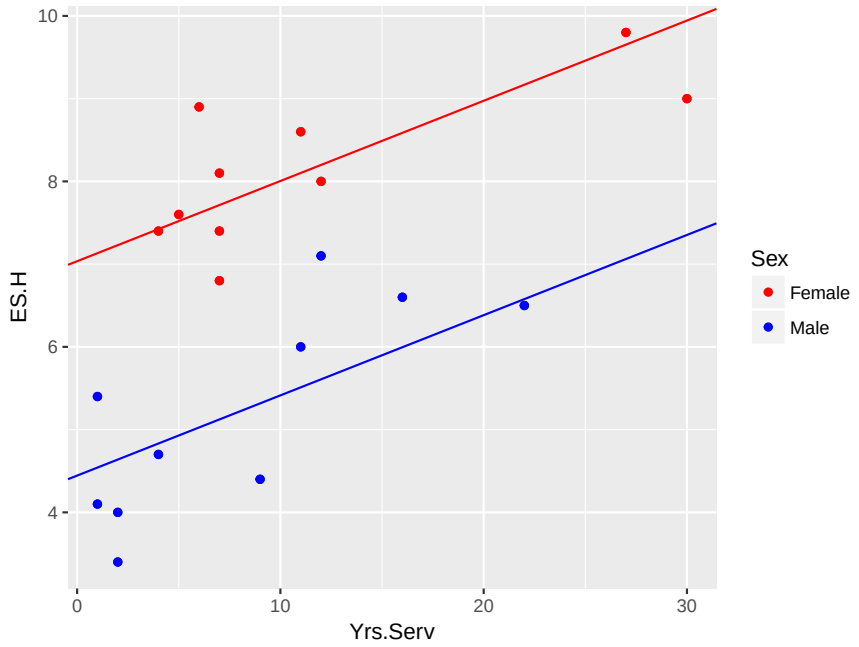
$$\begin{aligned}&\text{Male ES.H} = \\ &7.035 + 0.097\text{Yrs.Serv} - 2.591 \cdot 1 = \\ &4.444 + \mathbf{0.097}\text{Yrs.Serv}\end{aligned}$$

Notice that both equations have the same slope, so we have **parallel** lines.

Note such a model is also often called an *additive* model, similar to an ANOVA without interaction!

What does this look like? Here it is:

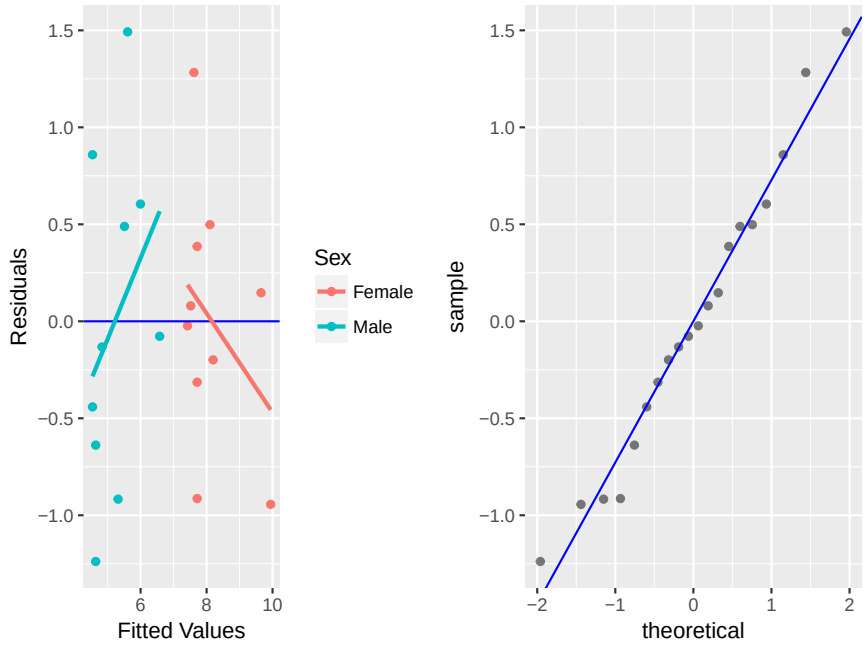
```
flplot(ES.H, Yrs.Serv, Sex, additive=TRUE)
```



Now a model with parallel line may or may not make sense for our data, but it does not have to. Except that no matter what, the way we used the categorical variable (simply code it and use it) we will **always** result in parallel lines!

Is there a way to see whether this is ok here? Yes, but it is a bit tricky: what we need is a version of the residual vs fits plot that identifies the plotting symbols by Sex. If the model is good, this residual vs fits plot should also show no pattern. We can get it easy if we use the **dlr** routine instead of the mlr:

```
dlr(ES.H, Yrs.Serv, Sex, additive=TRUE)
```

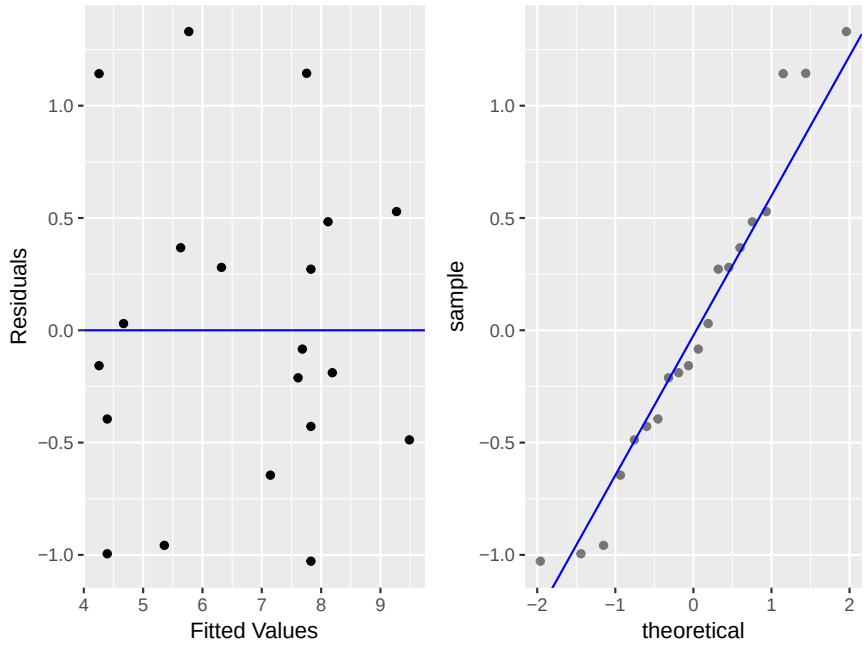



```
## The least squares regression equation is:
##  ES.H  =  7.035 + 0.097 Yrs.Serv - 2.591 Sex
## R^2 = 83.9
```

and as we can see there is a definite pattern in the colors.

So, how do we get away from parallel lines? This can be done by adding a variable Yrs.Serv*SexCode.

```
predictors <- cbind(Yrs.Serv, SexCode, Yrs.Serv*SexCode)
colnames(predictors)[3] <-  "Yrs.Serv*SexCode"
mlr(ES.H, predictors)
```




```
## The least squares regression equation is:
##  ES.H  =  7.323 + 0.072 Yrs.Serv - 3.203 SexCode + 0.065 Yrs.Serv*SexCode
## R^2 = 85.9%
```

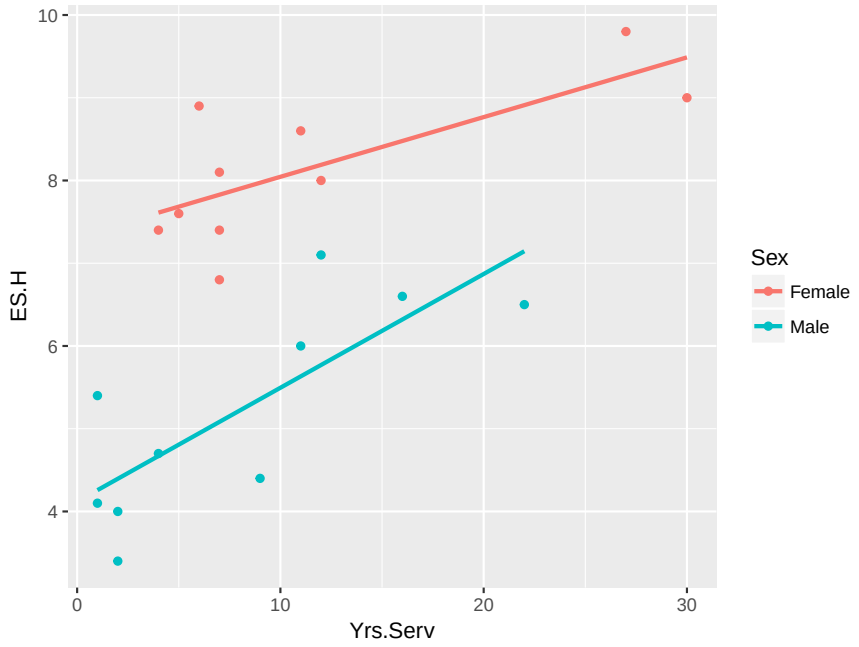
and now:

$$\begin{aligned}&\text{Female ES.H} = \\ &7.323 + 0.072\text{Yrs.Serv} - 3.203 \cdot 0 + 0.065 \cdot \text{Yrs.Serv*}0 = \\ &7.323 + 0.072\text{Yrs.Serv}\end{aligned}$$

$$\begin{aligned}&\text{Male ES.H} = \\ &7.323 + 0.072\text{Yrs.Serv} - 3.203 \cdot 1 + 0.065 \cdot \text{Yrs.Serv*}1 = \\ &4.120 + 0.138\text{Yrs.Serv}\end{aligned}$$

and so this fits *two separate lines*.

```
flplot(ES.H, Yrs.Serv, Sex)
```



Note you can get the same two equations by splitting up the dataset into two parts, the score and years of the Females and the score and years of the Males, and then doing a simple regression for both:

```
slr(ES.H[Sex=="Female"], Yrs.Serv[Sex=="Female"])
```

```
slr(ES.H[Sex=="Male"], Yrs.Serv[Sex=="Male"])
```

Doing one multiple regression has some advantages, though. For example you get one R^2 for the whole problem, not two for each part. Moreover, usually this R^2 will be higher than either of the other two.

So now we have two models:

- parallel lines: ES.H = 7.035 + 0.097 Yrs.Serv - 2.591 Sex R^2 = 83.9%
- two separate lines: ES.H = 7.323 + 0.072 Yrs.Serv - 3.203 SexCode + 0.065 Yrs.Serv*SexCode

R^2=85.85%

Clearly the second one has a higher R^2, but then the first one is a special case of the second (nested models) and so the model with parallel lines will **never** have an R^2 higher than the model with separate lines, and usually always has an R^2 a bit lower.

Of course the parallel lines model has two terms while the other one has three, and the third one is more complicated, so we would prefer the parallel lines model, if possible.

What we want to know is whether the model with two separate lines is **statistically significantly** better than the model with parallel lines. So we need a hypothesis test with:

H_0: the two separate lines model is NOT statistically significantly better than the parallel lines model.

H_a: the two separate lines model is statistically significantly better than the parallel lines model.

Notice that the parallel lines model is a special case of the ttwo independent lines model, and so we can use the *nested.models.test* to decide which is better:

```
parallel.lines <- dlr(ES.H, Yrs.Serv, Sex,
      return.model=TRUE)
independent.lines <- dlr(ES.H, Yrs.Serv, Sex, additive=TRUE,
      return.model=TRUE)
nested.models.test(independent.lines, parallel.lines)
```

```
## H0: both models are equally good.
## p value= 0.1608
```

gives a p-value of $0.1608 > 0.05$, so the parallel lines model is just as good as the model with two separate lines.

Note in this command the bigger model has to come first!

32.0.1.1 Prediction

We have the dlr.predict command to do prediction. Let's find 95% interval estimates for female employees with 0, 1, 2,..,10 years of service:

```
dlr.predict(ES.H, Yrs.Serv, Sex, newx=0:10,
    newz=rep("Female", 11), additive=TRUE, interval="PI")
```

```
##    Yrs.Serv    Sex  Fit Lower Upper
## 1         0 Female 7.04  5.21  8.86
## 2         1 Female 7.13  5.32  8.94
## 3         2 Female 7.23  5.43  9.03
## 4         3 Female 7.33  5.54  9.11
## 5         4 Female 7.42  5.65  9.20
## 6         5 Female 7.52  5.75  9.29
## 7         6 Female 7.62  5.86  9.38
## 8         7 Female 7.71  5.96  9.47
## 9         8 Female 7.81  6.06  9.56
## 10        9 Female 7.91  6.16  9.65
## 11       10 Female    8  6.26  9.75
```

32.1 Lines and Interaction

Above we explained the problem of using categorical predictors in a regression model in terms of parallel lines vs. two independent lines. But in fact this another example of the issue of *interaction*, or more generally of a relationship between the predictors. Parallel lines are ok if the categorical and the continuous predictors are essentially independent. Often terms such as Yrs Serv*SexCode are also called *interaction terms*.

For your purposes in this class (and later when doing work such as this) simply remember to include product terms when you have categorical predictors. Then you can test if that term is really needed, and drop it if it is not.

32.1.1 Case Study: Sales of Shoes

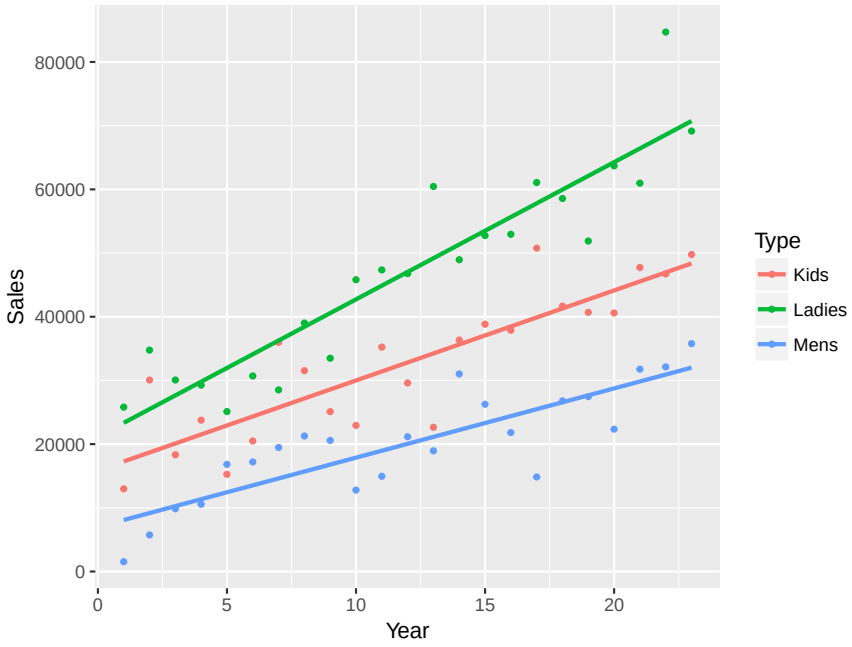
The number of shoes sold by year and type.

```
head(shoesales)
```

```
##   Sales Year   Type
## 1  1539    1   Mens
## 2 12984    1   Kids
## 3 25809    1 Ladies
## 4  5742    2   Mens
## 5 30058    2   Kids
## 6 34764    2 Ladies
```

Let's have a look at the data. Previously we used two graphs, we can also use a version of the scatterplot that identifies the dots by the categorical variable:

```
attach(shoesales)
splot(Sales, Year, Type, add.line=1)
```



We want to find a model for predicting Sales from Year and Type. Again Type is a categorical variable and so we need to code it. The most obvious thing to do would be to code:

- Mens= 0
- Kids= 1

- Ladies = 2

but that is dangerous. Unlike a categorical variable numbers always have an order and a size. So by coding in this way we are saying that Mens comes before Kids. Worse , we are saying that the "distance" from Mens to Kids is the same as the "distance" from Kids to Ladies!

Whether this matters or not dependes on the specific problem. There is however a way to include such a variable without introducing order or size:

```
d1 <- rep(0, length(Type))
d1[Type=="Kids"] <- 1
d2 <- rep(0, length(Type))
d2[Type=="Ladies"] <- 1
```

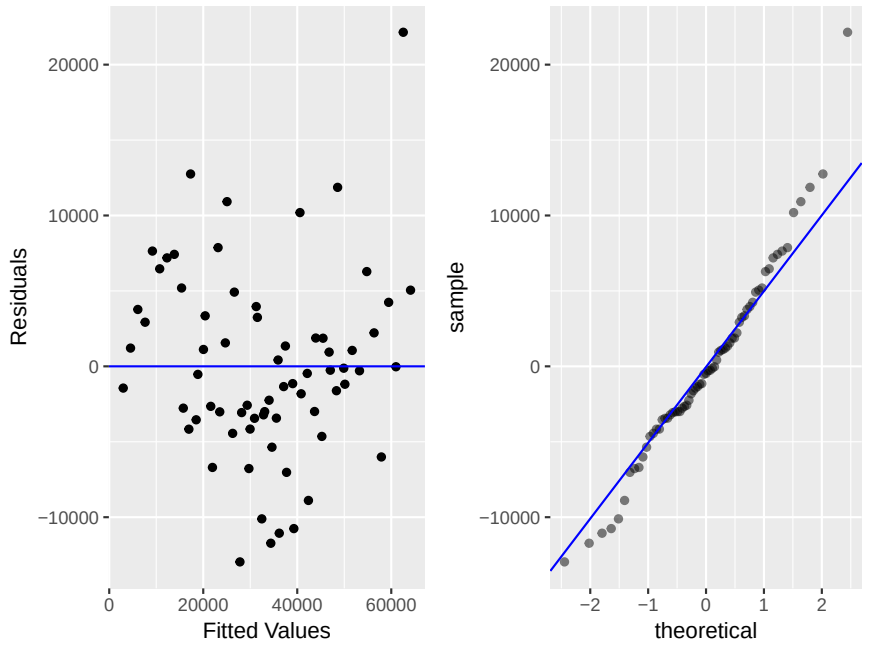
Notice that by knowing d1 and d2 we now exactly what the type is:

- d1=0, d2=0 → Mens
- d1=1, d2=0 → Kids
- d1=0, d2=1 → Ladies

so we have not lost any information, but we have also not introduced any order or size!

Now

```
predictors <- cbind(Year, d1, d2)
colnames(predictors) <- c("Year", "d1", "d2")
mlr(Sales, predictors)
```

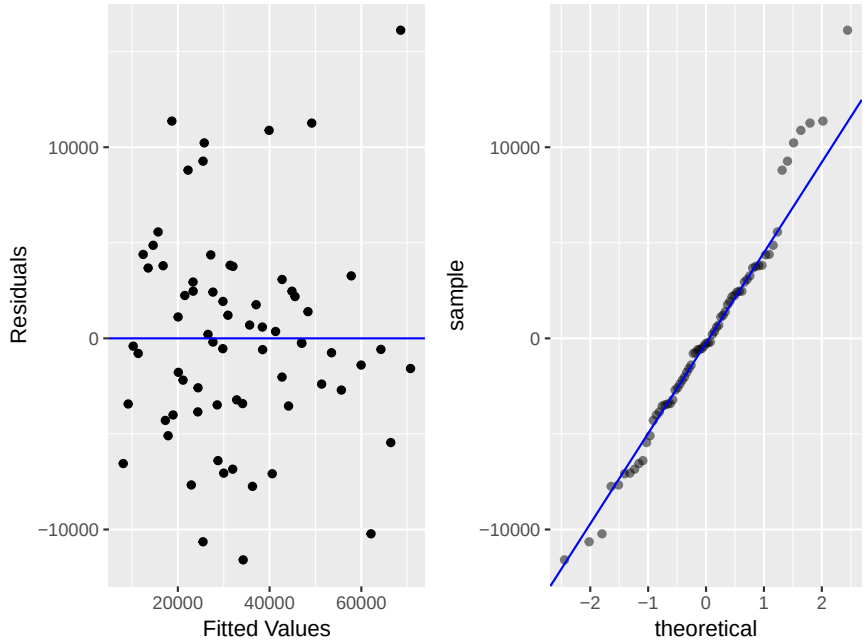



```
## The least squares regression equation is:
##  Sales  =  1429.126 + 1551.562 Year + 12774.478 d1 + 26986.87 d2
## R^2 = 85.6%
```

Note if there is one categorical predictor with just two values (binary data) we can use the dlr command. If the categorical variable has three or more values or if there is more than one categorical variable we can use the mlr command but we have to work a bit on getting the matrix of predictor variables right.

This is of course an additive model, again we should worry about interaction. But now we have two categorical predictors, so we need to add two product terms:

```
predictors <- cbind(Year, d1, d2, Year*d1, Year*d2)
colnames(predictors) <- c("Year", "d1", "d2", "Year*d1", "Year*d2")
mlr(Sales, predictors)
```




```
## The least squares regression equation is:
##  Sales  =  7000.455 + 1087.285 Year + 8862.217 d1 + 14185.146 d2 + 326.022 Year*d1 +
## R^2 = 88.9%
```

And again we can test whether the product terms are needed:

```
with.interaction <- mlr(Sales, predictors, return.model=TRUE)
without.interaction <- mlr(Sales, predictors[,1:3],
                    return.model=TRUE)
nested.models.test(with.interaction, without.interaction)
```

```
## H0: both models are equally good.
## p value= 0.000
```

and we find that here the interaction is needed (p= 0.000).

33 Exercises 4

33.0.1 Case Study: Survey of Students

This is the same data set we considerd in the previous exerceises.

Problem 1 Find the best model to predict Score from GPA and Distance.

Problem 2 Analyse the data with score as the response and Gender and Years as factors (predictors).

Problem 3 Find a 90% interval estimate for the score of a male student with a GPA of 2.15. Is this an interpolation or an extrapolation?

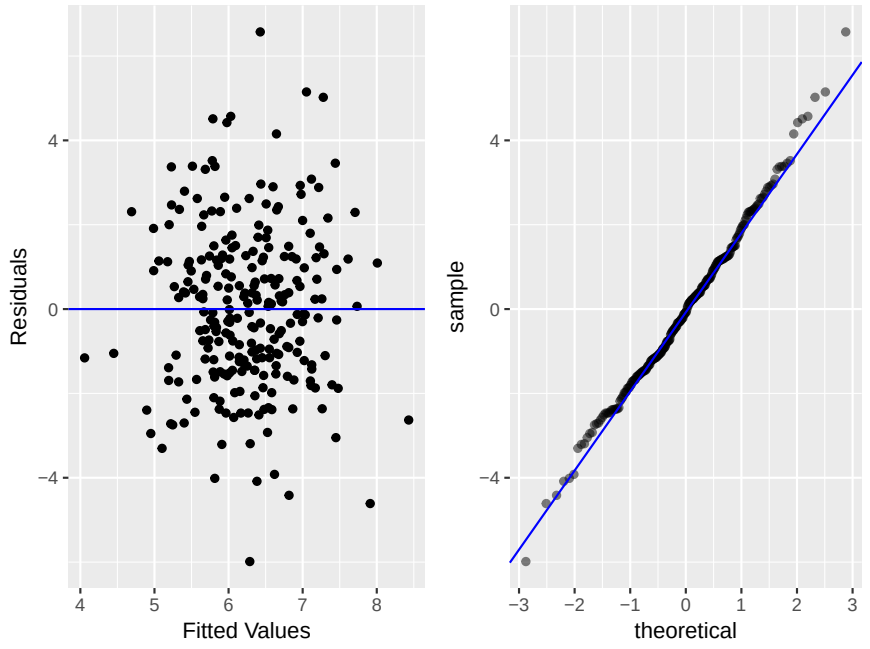
Problem 4 Find a 95% interval estimate for the score of a 20 year old male student who lives 2 miles from school. (code the variable Gender, ignore the issue of parallel lines)

```
attach(studentsurvey)
```

Problem 1 Find the best model to predict Score from GPA and Distance.

In Exercise Problems 3 we found a linear model in GPA and a log model in Distance, so let's try this:

```
mlr(Score, cbind(GPA, log(Distance + 1)))
```

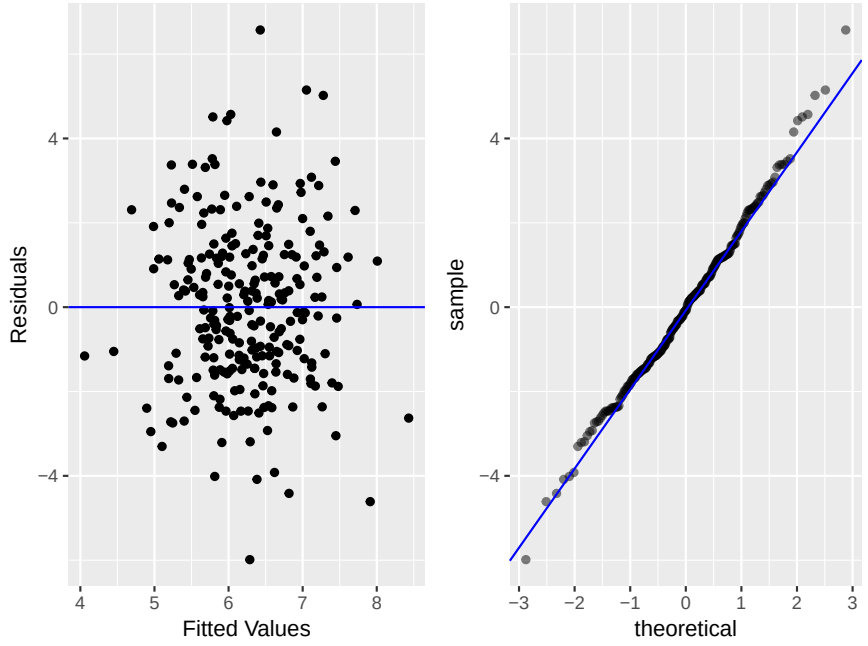



```
## The least squares regression equation is:
##  Score  =  3.089 + 1.329 GPA + 0.024
## R^2 = 11%
```

the plots look good, so no problem with the assumptions.

Notice that the name of the variable Distance is missing. If we want to fix that we can do this:

```
X <- cbind(GPA, log(Distance + 1))
colnames(X) <- c("GPA", "log(Distance + 1)")
mlr(Score, X)
```




```
## The least squares regression equation is:
##  Score  =  3.089 + 1.329 GPA + 0.024 log(Distance + 1)
## R^2 = 11%
```

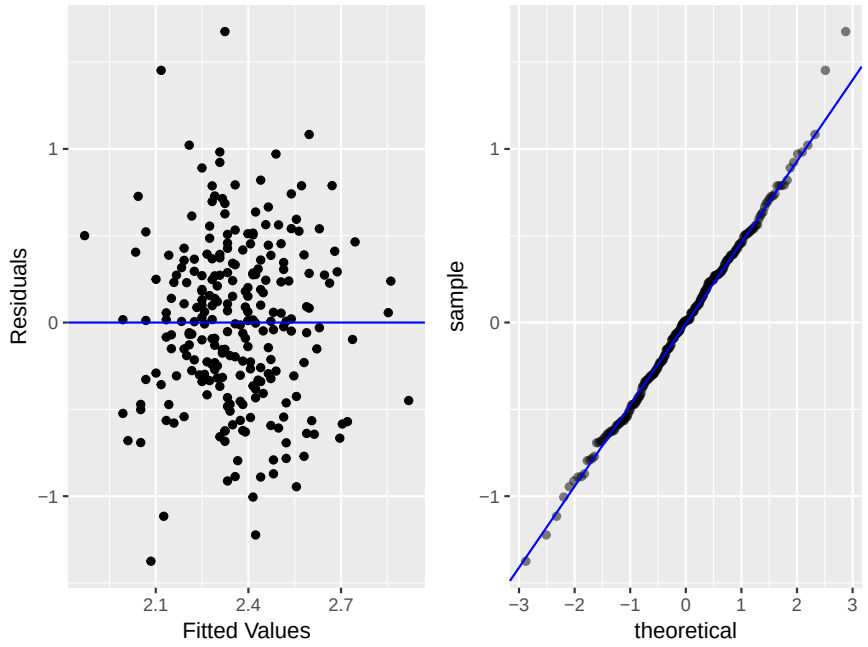
Can we simplify the model?

```
mallows(Score, X)
```

```
##  Number of Variables Cp   GPA log(Distance + 1)
##  1                   1.04 X
##  2                   3    X   X
```

the smallest C_p is for the model with GPA only, so this is best.

```
slr(GPA, Score)
```



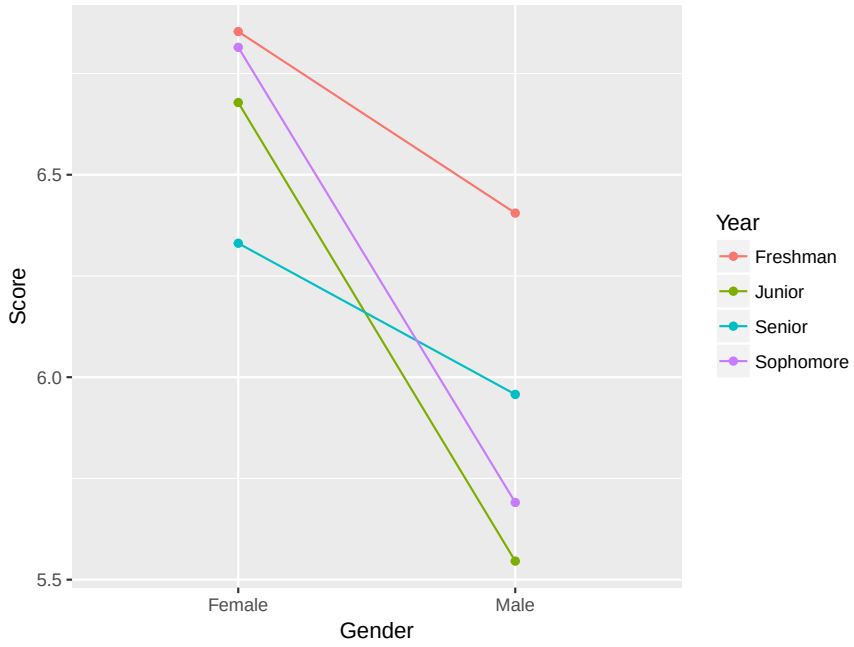

```
## The least squares regression equation is:
##  GPA  = 1.845 + 0.083 Score
## R^2 = 11%
```

Problem 2 Analyse the data with score as the response and Gender and Years as factors (predictors).

this is a twoway ANOVA problem. In Exercise Problems 2 we already looked at the boxplots and the summary statistics. Next we need to consider any possible interaction:

In problem 3 of the Exercise Problems 2 we found a statistically significant correlation between Score and GPA. Let's find a good model.

```
iplot(Score, Gender, Year)
```



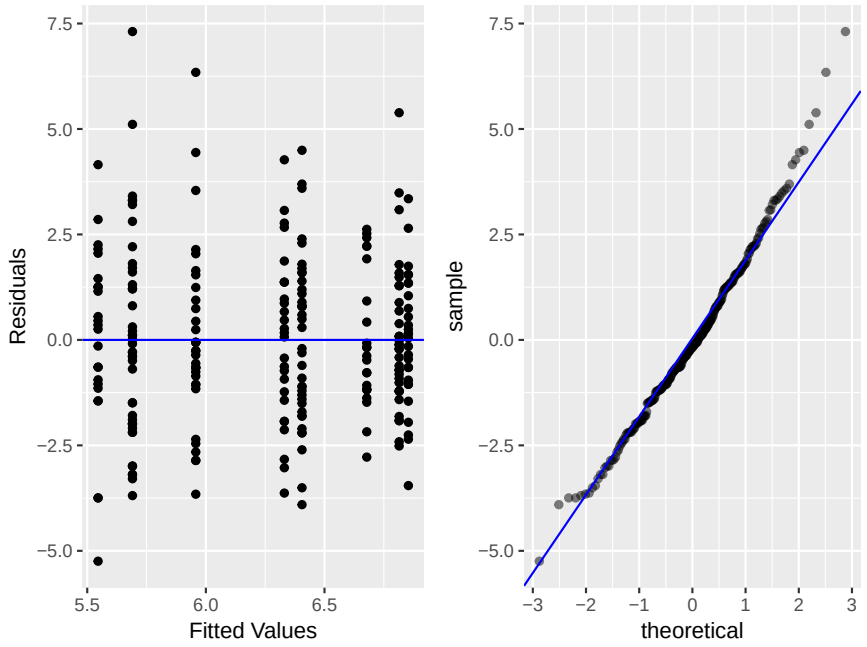
there seems to be interaction here. Can we test for it? We would need repeated measurements:

```
table(Gender, Year)
```

```
##         Year
## Gender   Freshman Junior Senior Sophomore
##   Female       28     23     26        34
##   Male         38     24     33        43
```

which we have. So

```
twoway(Score, Gender, Year)
```

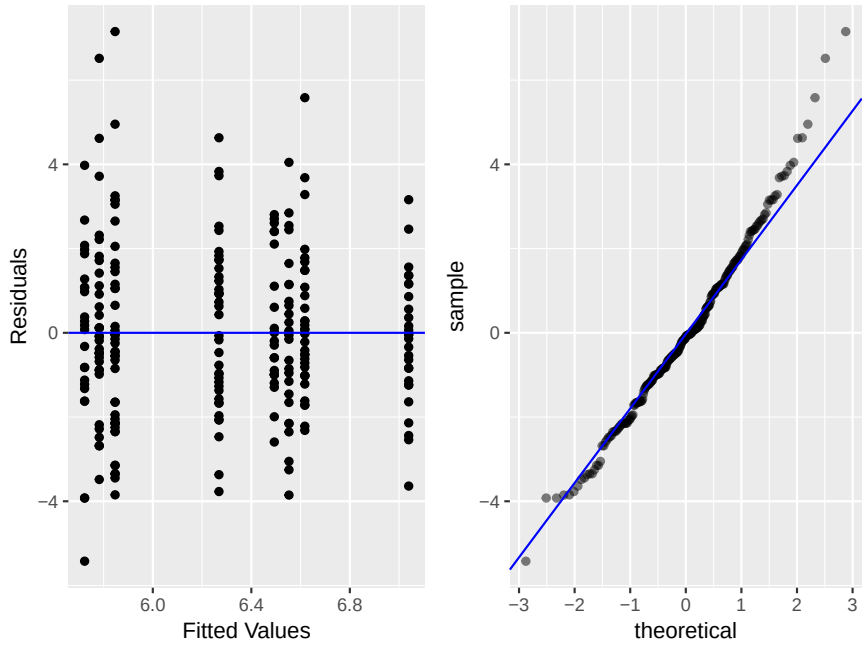



```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## x               1   35.2   35.23   8.742 0.00342
## z               3   11.4    3.79   0.940 0.42212
## x:z             3    7.9    2.63   0.652 0.58255
## Residuals     241  971.3    4.03
##                      [,1]
## Gender  p =      0.0034
## Year  p =        0.4221
## Interaction p = 0.5825
```

the plots look good, so no problems with the assumptions.

The test for interaction has p=0.5825 , so there is no evidence of interaction. We can refit with out it:

```
twoway(Score, Gender, Year, with.interaction=FALSE)
```



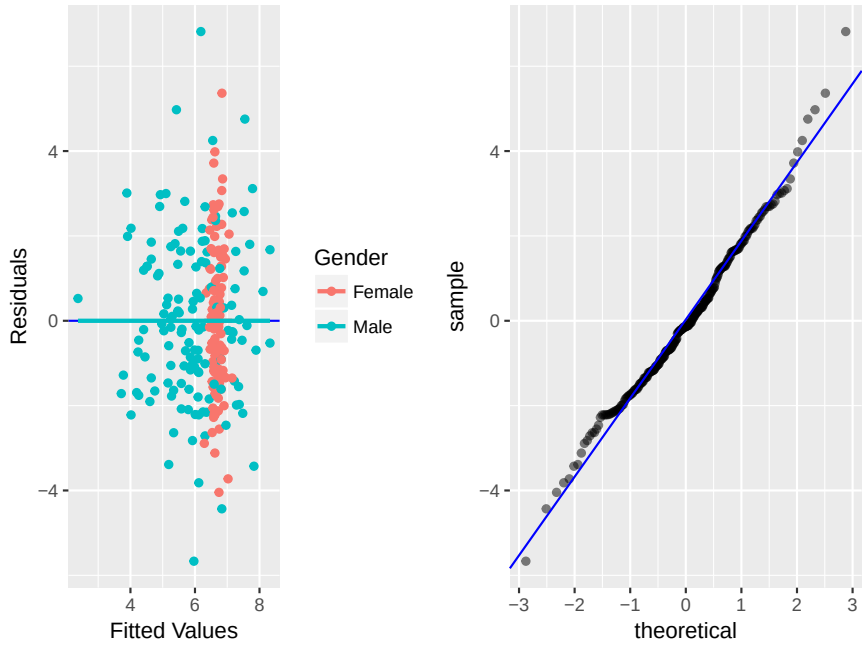

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## x               1   35.2   35.23   8.780 0.00335
## z               3   11.4    3.79   0.944 0.42013
## Residuals     244  979.2    4.01
##                 [,1]
## Gender  p = 0.0033
## Year  p =   0.4201
```

The test for Year has p=0.42, so Year does not effect score and so we can drop the term as well. We are now back to a oneway of Score by GPA, which we already analysed in Exercise Problems 2 and 3.

Problem 3 Find a 90% interval estimate for the score of a male student with a GPA of 2.15. Is this an interpolation or an extrapolation?

We have a quantitative response (Score), a quantitative predictor (GPA) and a catgorical predictor(Gender), so this is a regression problem with a dummy variable.

```
dlr(Score, GPA, Gender)
```




```
## The least squares regression equation is:
##  Score  =  6 + 0.287 GPA - 5.162 Gender + 1.877 GPA*Gender
## R^2 = 19.6
```

do we need the product term?

```
independent.lines <- dlr(Score, GPA, Gender, return.model=TRUE)
parallel.lines <- dlr(Score, GPA, Gender, additive=TRUE,
                  return.model=TRUE)
nested.models.test(parallel.lines, independent.lines)
```

```
## H0: both models are equally good.
## p value= 0.000
```

gives p=0.000, so the product term is indeed necessary.

Next we want a 90% interval estimate for the score of a male student with a GPA of 2.15:

```
dlr.predict(Score, GPA, Gender, newx=2.15,
            newz="Male", interval="PI", conf.level=90)
```

```
##    GPA Gender  Fit Lower Upper
## 1 2.15   Male 5.49  2.45  8.53
```

Is this an interpolation or an extrapolation?

```
range(GPA[Gender=="Male"])
```

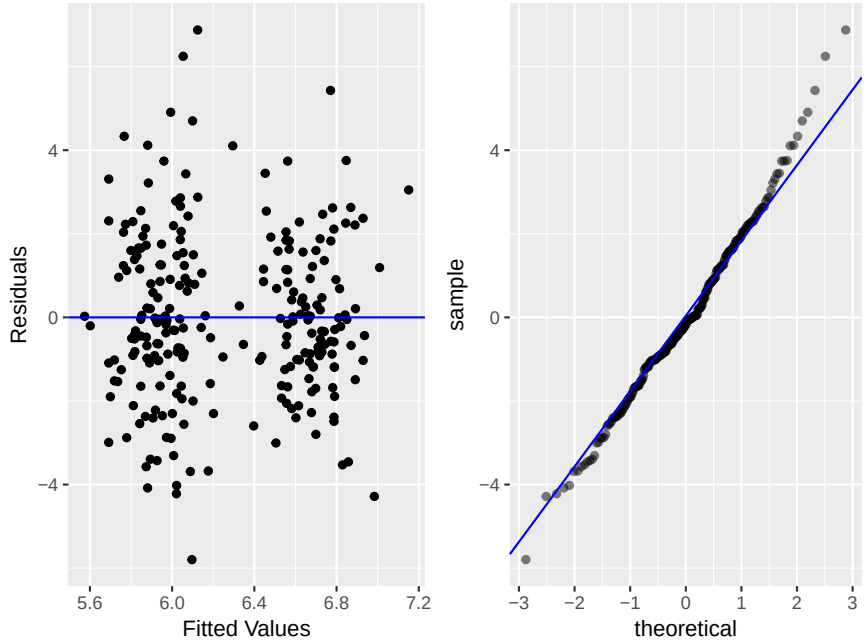
```
## [1] 0.71 3.46
```

shows that 2.15 is in the range of data values for males, so this is an interpolation.

Problem 4 Find a 95% interval estimate for the score of a 20 year old male student who lives 2 miles from school. (code the variable Gender, ignore the issue of parallel lines)

As before we need to remove observation #220 from the data set and we need to use the log tranform on Distance:

```
X <- data.frame(Distance=log(Distance[-220]+1), Age=Age[-220], Gender=ifelse(Gender[-220
mlr(Score[-220] , X)
```




```
## The least squares regression equation is:
##  Score[-220]  =  8.905 + 0.082 Distance - 0.118 Age - 0.745 Gender
## R^2 = 3.8%
```

When doing prediction there is usually no reason to simplify the model, so we won't use Mallows C_p. Now

```
newx <- cbind(Distance=log(2)+1, Age=20, Gender=1)
mlr.predict(Score[-220], X, newx=newx, interval="PI")
```

```
##  Distance Age Gender  Fit Lower Upper
##  1.693147  20      1 5.95  1.99  9.91
```