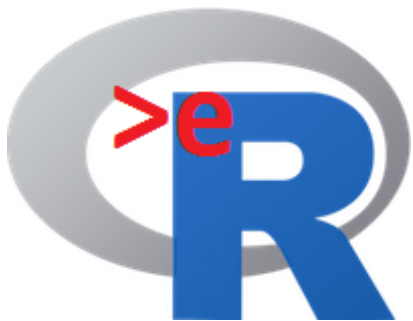




This course was developed as a part of the VLIR-UOS Cross-Cutting projects:

- Statistics: 2011-2016, 2017.
- Statistics: 2017.
- Statistics for development : 2018-2020.



The >eR-Biostat initiative  
Making R based education materials in  
statistics accessible for all

## An introduction to R: Short Version (2017)

Developed by  
Dan Lin (Hasselt University) and Ziv Shkedy (Hasselt University)

LAST UPDATE: 15/10/2017



Visit us on  
Facebook

ER-BioStat

Email: [erbiostat@gmail.com](mailto:erbiostat@gmail.com)



<https://github.com/eR-Biostat>



@erbiostat

# Overview

A (very) quick start: the cars data

Two sample t-test.

basic plots

1. Basic programming in R: objects in R
2. Reading external datasets
3. Basic plots functions
4. Programming in R: a for loop
5. Statistical modeling in R: simple linear regression
6. Statistical modeling in R: one-way ANOVA
7. Statistical modeling in R: logistic regression
8. Programming in R: user functions
9. Two-way ANOVA
10. More about two-way ANOVA,
11. More about linear regression
12. Application of a for loop: bootstrap.

# **A (very) quick Start**

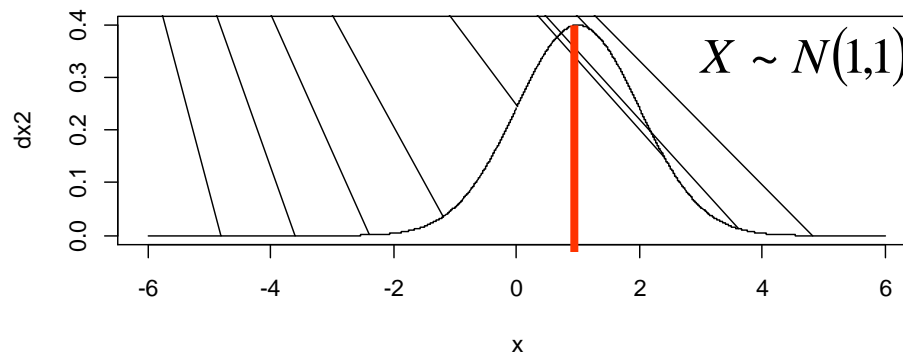
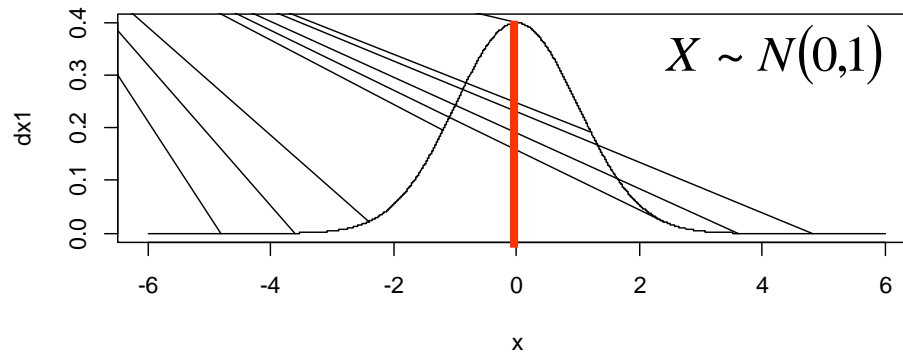
# The R environment

- Open R.
- Open a new script window.

# The normal distribution: location

Density function of a normal distribution

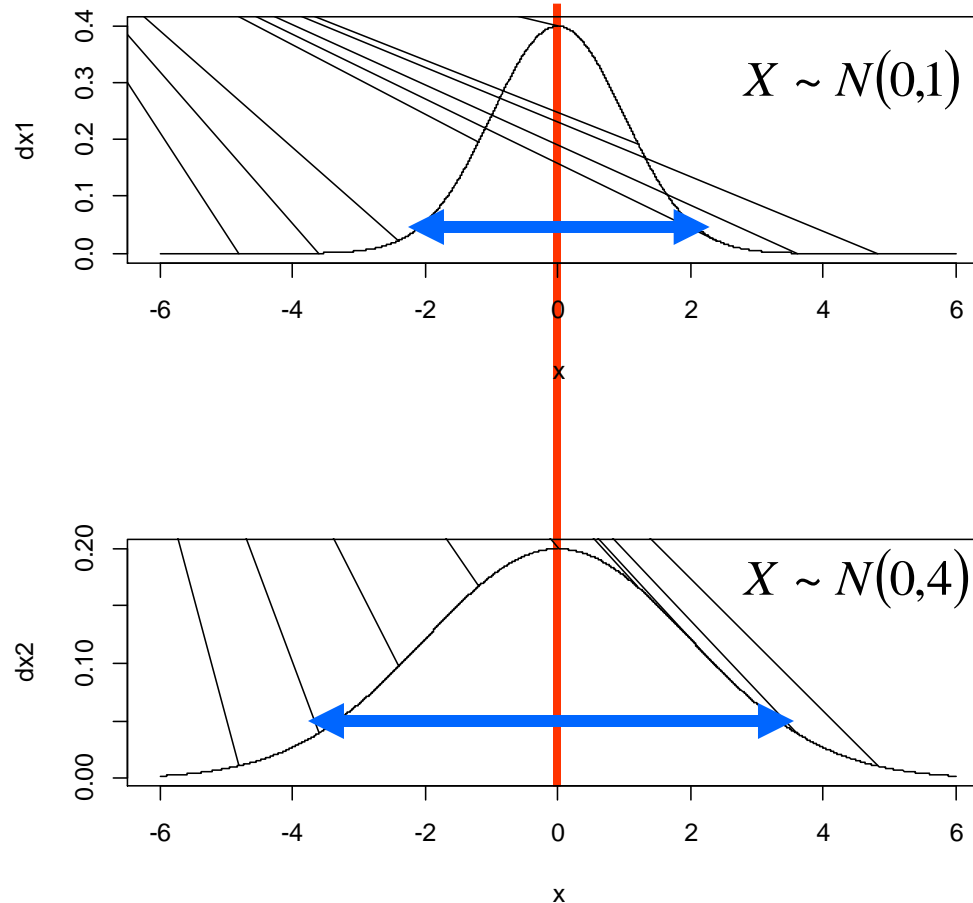
$$X \sim N(\mu, \sigma^2)$$



# The normal distribution: variability

Density function of a normal distribution

$$X \sim N(\mu, \sigma^2)$$



# Random sample from a normal distribution in R

Draw a random sample of size 100 from a normal distribution with mean – and variance 1

$$X \sim N(\mu, \sigma^2)$$

In R

**rnorm**(sample size, mean, standard deviation)

$$X \sim N(0,1)$$

**rnorm**(100, 0, 1)



# Random sample from a normal distribution in R

Draw a random sample of size 100 from a normal distribution with mean – and variance 1

$$X \sim N(\mu, \sigma^2) \Rightarrow X \sim N(0,1)$$

```
> rnorm(100,0,1)
```

```
[1] -0.173911348 -0.463196096 -1.084838332  2.373958677 -1.685884982
[6] -1.952672126 -0.055601310 -0.241913096 -0.999586206  0.308335895
[11]  0.556993818  2.337451275  0.778734465 -0.501354458  0.004525392
[16] -1.468709822  0.109901143  0.109103689  0.662434110 -0.177097648
[21] -1.442033566  0.615239368  0.254080126  1.152977602 -0.089559002
[26]  0.065022482  0.300405204 -0.190196930 -0.244365328  0.886735849
[31] -0.667671228 -1.009209277  0.388362272 -0.041883373  0.750480061
[36] -2.103109677 -1.515839684 -0.477250540 -0.344581482  0.072570862
[41] -0.364485234 -0.920898769  1.148778190  1.092225688 -0.832389361
[46] -1.914844153 -0.384265110  0.528078353  1.319149374  0.226817654
[51] -0.605867376 -0.658048328  0.086126314  0.711404951  1.190303122
[56]  2.499314086  2.201924724  0.591527333 -0.733622099 -0.656031690
[61] -0.194759316  0.864421699  0.813854743 -0.628803589  0.362077258
[66]  0.312250497  1.451227963  1.107136623  0.680487861  1.585879056
[71] -0.249983835 -1.436293634 -0.470710524 -2.330088808  0.265551343
[76] -0.847238216 -1.199413581 -1.866542460  0.826973063 -0.592073631
[81] -1.751735134  0.077115620 -0.306869702  0.120083596 -0.303521155
[86] -0.644268518  0.295067198  2.004409939  0.310290927  0.221898330
[91] -1.450606907 -1.264043444 -0.257282348  0.078120141 -0.902925645
[96]  0.499980835 -0.596173525 -1.085097601 -0.773094391  0.693319162
```

# Creating an R object

```
> x<-rnorm(100,0,1)
```

An R object contain the results

```
> x
```

```
[1] -1.91083203  1.04955497 -2.40884482  0.33493954  1.45434660 -2.42198672  
[7]  0.44232862 -0.73804911 -0.36354587  0.39064194 -0.31993512 -1.30809569  
[13]  0.11409195  0.43549125 -0.29501115  0.29197212  0.50983934 -0.80452037  
[19] -0.61008244  1.80780477  1.31535974 -1.33155401  0.29044725 -0.63380504  
  
[85]  1.03861350  0.89381884  0.86323215 -0.24199953  1.64380126  0.45445204  
[91]  1.90708641  0.34088349 -0.25727644 -0.26498359  0.80095645  1.42711451  
[97]  1.27998167 -0.54106317 -1.29443674  0.36046722
```

Print the R object

# Summary statistics

```
> mean(x)  
[1] 0.02149641
```



Sample mean

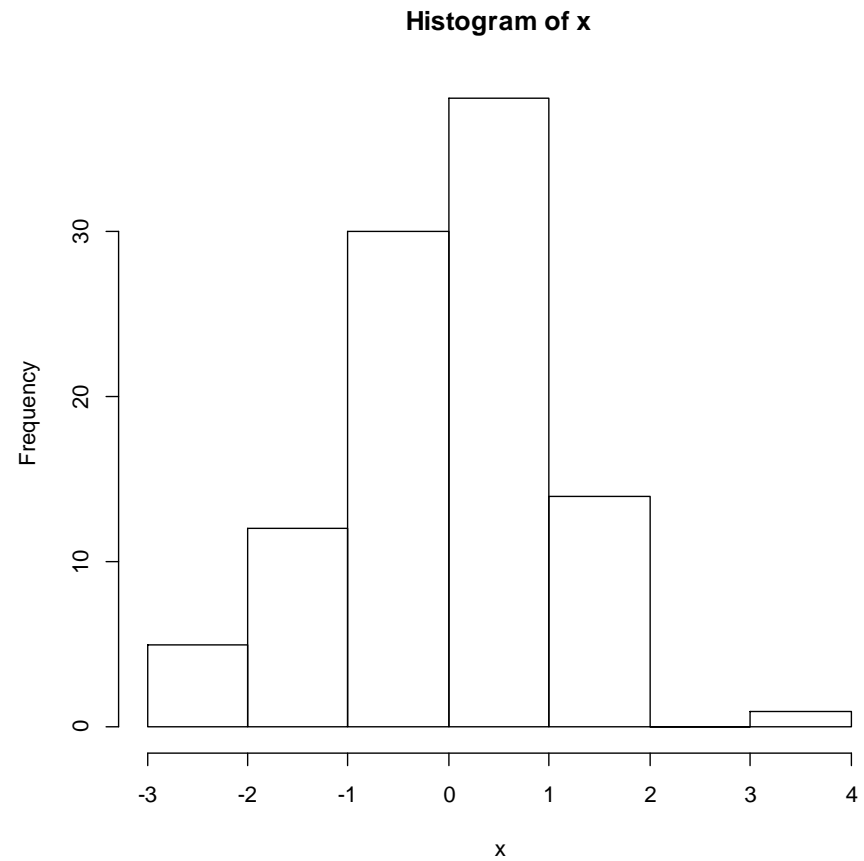
```
> var(x)  
[1] 1.061159
```



Sample variance

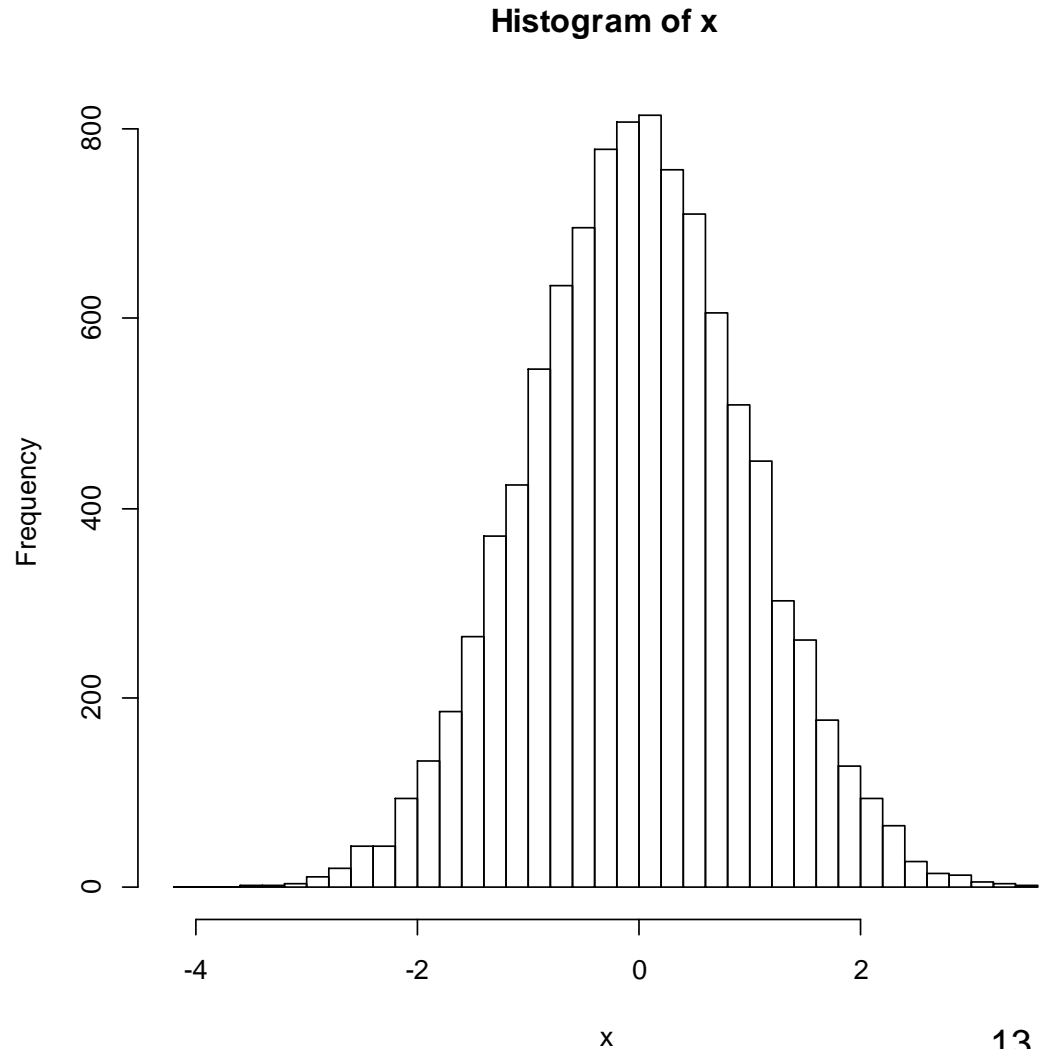
# Histogram of the sample

```
> hist(x)
```



# Histogram of the sample

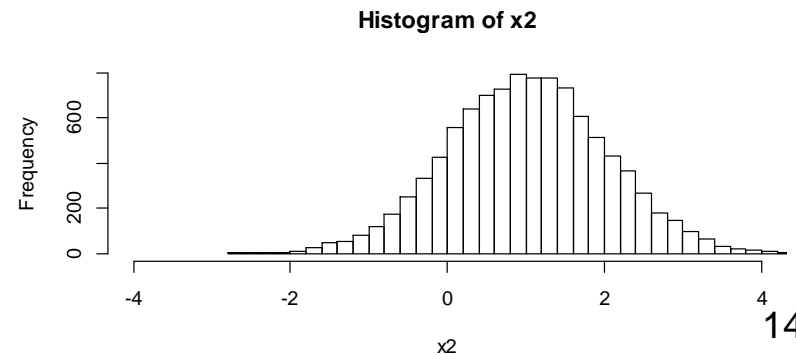
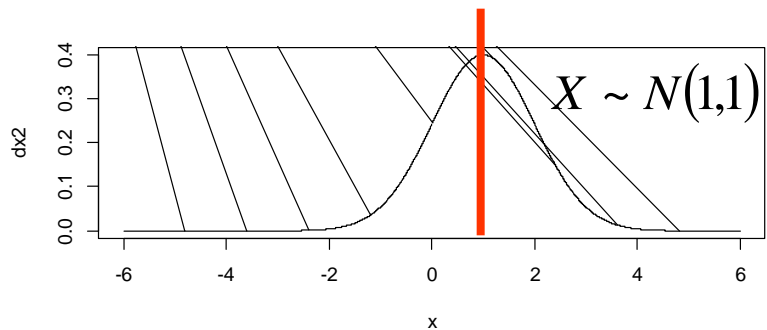
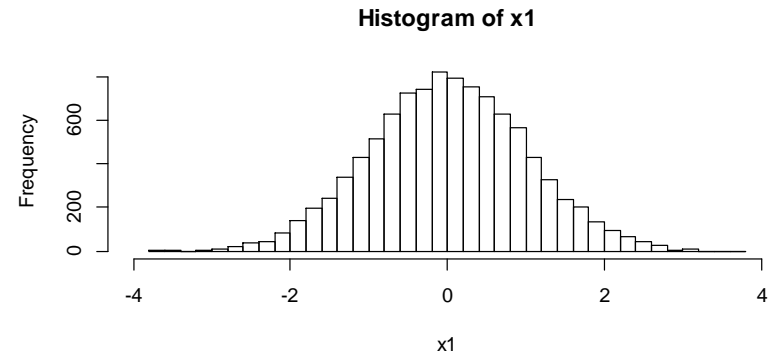
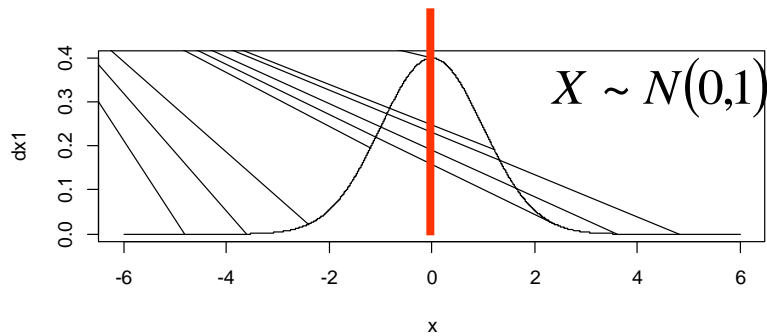
```
> x<-rnorm(10000,0,1)
> mean(x)
[1] -0.01259969
> var(x)
[1] 0.9871957
> hist(x,nclass=50)
```



# The normal distribution: location

$$X \sim N(\mu, \sigma^2)$$

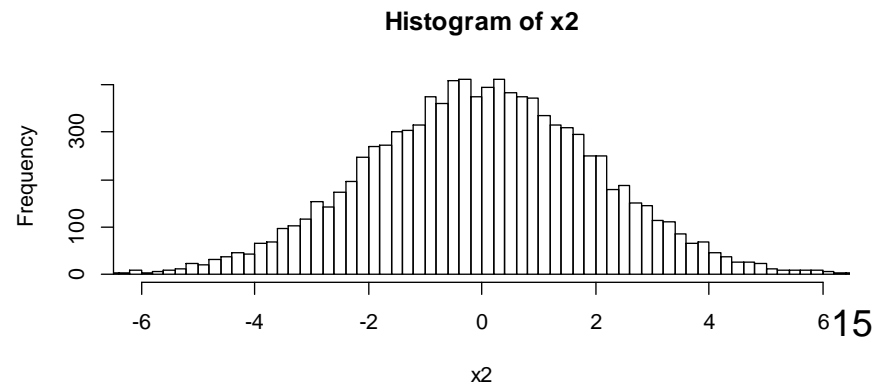
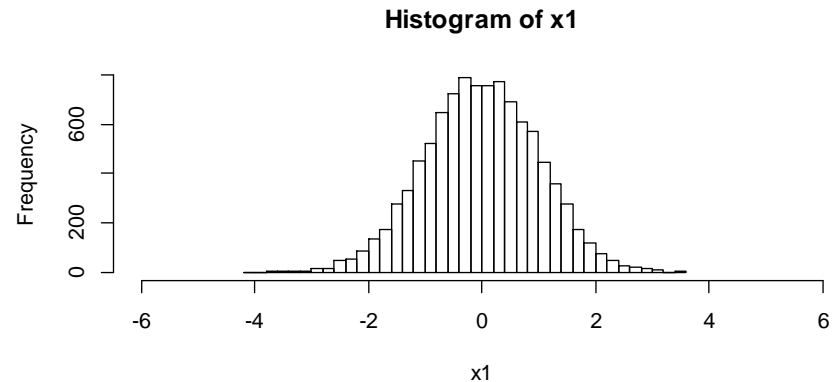
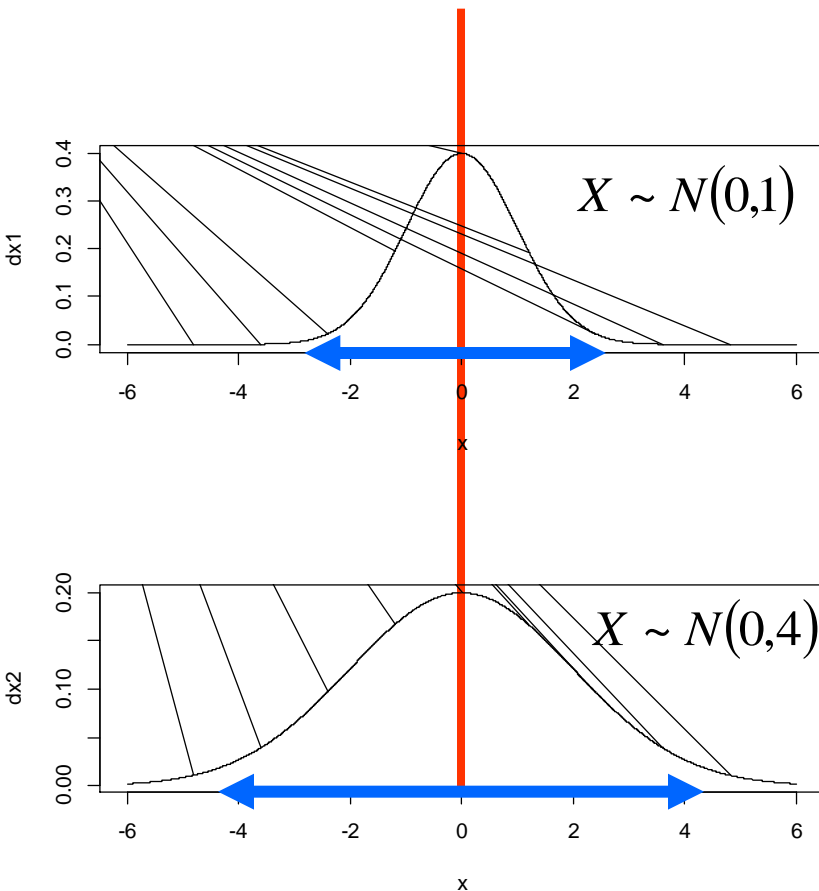
```
> x1<-rnorm(10000,0,1)
> x2<-rnorm(10000,1,1)
> par(mfrow=c(2,1))
> hist(x1,nclass=50,xlim=c(-4,4))
> hist(x2,nclass=50,xlim=c(-4,4))
```



# The normal distribution: variability

$$X \sim N(\mu, \sigma^2)$$

```
> x1<-rnorm(10000,0,1)
> x2<-rnorm(10000,0,2)
> par(mfrow=c(2,1))
> hist(x1,nclass=50,xlim=c(-6,6))
> hist(x2,nclass=100,xlim=c(-6,6))
```



# Working with data



# The cars Data set in R

1. Write **cars** in the script window.
2. Submit

```
> cars
```

	speed	dist
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10
.	.	.
.	.	.
48	24	93
49	24	120
50	25	85

```
> help(cars)
```

```
>
```

## Speed and Stopping Distances of Cars Description

The data give the speed of cars and the distances taken to stop. Note that the data were recorded in the 1920s.

```
[,1] speed    numeric    Speed (mph)
```

```
[,2] dist     numeric    Stopping distance (ft)
```

# The cars Data set in R: the \$ sign

```
> speed
```

```
Error: object 'speed' not found
```

```
>
```

```
> cars$speed
```

```
[1] 4 4 7 7 8 9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14 15 15  
[26] 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 20 22 23 24 24 24 24  
25
```

```
>
```

# The cars Data set in R: creating a new object

```
> cars[,1]
```

```
[1] 4 4 7 7 8 9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14 15 15  
[26] 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 20 22 23 24 24 24 24  
25
```

```
> x=cars[,1]
```

```
> print(x)
```

```
[1] 4 4 7 7 8 9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14 15 15  
[26] 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 20 22 23 24 24 24 24  
25  
>
```

# Basic plot and descriptive statistics

- What is the average speed of the cars ?
- What is the variance of the cars' speed ?
- What is the min. (max.) speed ?
- What is the association between speed and stopping distance ?

# Descriptive statistics

```
> mean(cars$speed)  
[1] 15.4
```

```
> max(cars$speed)  
[1] 25
```

```
> min(cars$speed)  
[1] 4
```



The variable speed  
in the dataset cars

```
> head(cars)  
  speed dist  
1     4    2  
2     4   10  
3     7    4  
4     7   22  
5     8   16  
6     9   10
```

# attach(data)

```
> attach(cars)
```

Tells R to work with the dataset cars.

```
> mean(speed)
```

```
[1] 15.4
```

```
> max(speed)
```

```
[1] 25
```

```
> min(speed)
```

```
[1] 4
```

We can work with the variables by using their names.

```
> detach(cars)
```

Stop using the dataset cars.

# Descriptive statistics

Correlation between the variables speed and stopping distance.

```
> cor(cars)
      speed      dist
speed 1.0000000 0.8068949
dist  0.8068949 1.0000000
```

# R functions

```
function(data)
```

A procedure that was programmed in R that uses data to produce output.

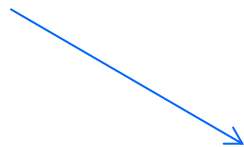
```
> var(cars$speed)  
[1] 27.95918
```

Calculate the variance.

```
> var(cars$speed)
```



function



data



# R functions

```
> print(cars)
  speed dist
1      4    2
2      4   10
3      7    4
4      7   22
45     23   54
46     24   70
47     24   92
48     24   93
49     24  120
50     25   85
```

```
> cor(cars)
           speed      dist
speed 1.0000000 0.8068949
dist  0.8068949 1.0000000
```

# R functions

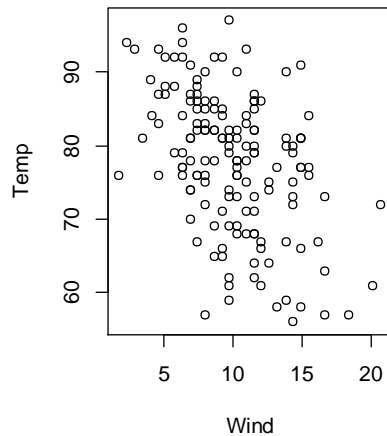
- Can be used for
  - Data analysis: descriptive statistics, testing, modeling, etc.
  - Data manipulation: selection of cases, variables...
  - Data management: reading and writing datasets into/from R.
  - Visualization: plots for the data.
  - .....

# Discussion

- R Objects: data frame.
- R functions.
- \$.

# Practical session 1

- The **airquality** is a dataset available in R.
- How many variables there are in the data ?
- Define an R object which contain the information about the wind speed.
- Calculate the mean, and variance for the wind speed.



# Two-sample t-test

# The sleep data in R

```
> help(sleep)
```

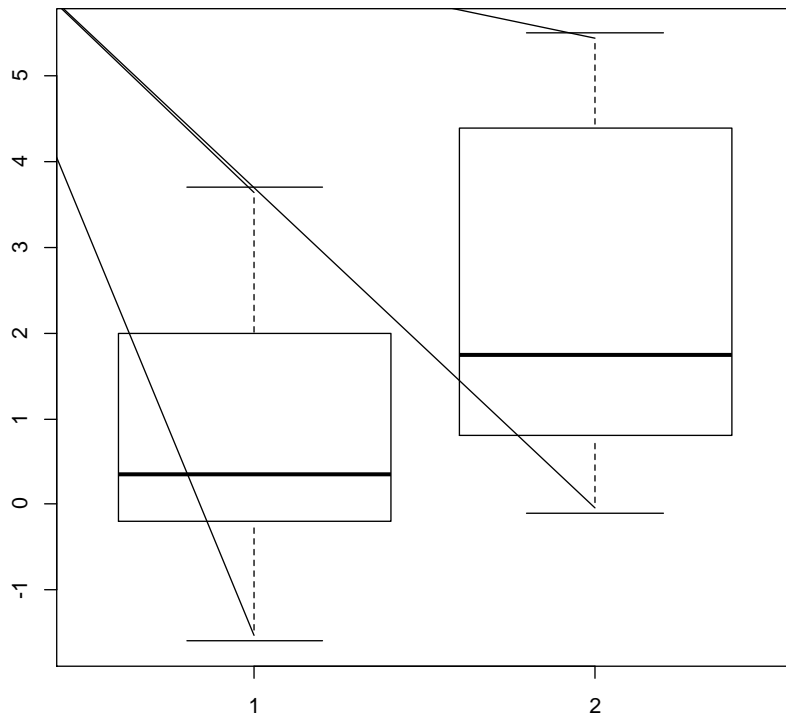
Data which show the effect of two soporific drugs (increase in hours of sleep compared to control) on 10 patients.

extra	numeric	increase in hours of sleep
group	factor	drug given
ID	factor	patient ID

```
> sleep
```

	extra	group	ID
1	0.7	1	1
2	-1.6	1	2
3	-0.2	1	3
4	-1.2	1	4
5	-0.1	1	5
.	.	.	.
14	0.1	2	4
15	-0.1	2	5
16	4.4	2	6
17	5.5	2	7
18	1.6	2	8
19	4.6	2	9
20	3.4	2	10

# Two samples t-test



```
> extra=sleep$extra  
> group=sleep$group  
> boxplot(split(extra,group))
```

The aim of the analysis:

Test for a difference between the two soporific drugs

# Two samples t-test

```
> t.test(extra~group,var.equal=TRUE)
```

Two Sample t-test

data: extra by group

t = -1.8608, df = 18, p-value = 0.07919

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-3.363874 0.203874

sample estimates:

mean in group 1	mean in group 2
0.75	2.33

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

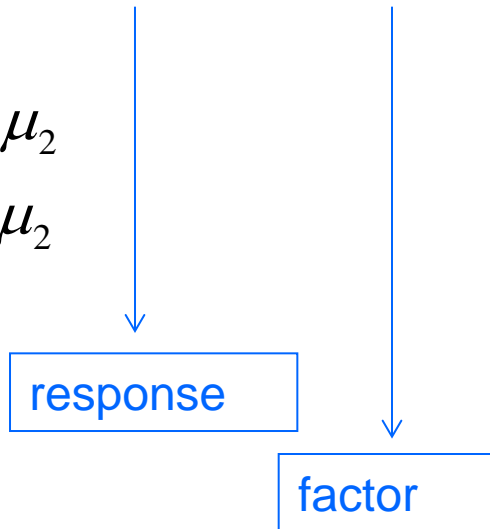


# Two samples t-test

> **t.test(extra~group, var.equal=TRUE)**

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$



# R object for the output

```
> t.obj=t.test(extra~group,var.equal=TRUE)  
> summary(t.obj)
```

	Length	Class	Mode
statistic	1	-none-	numeric
parameter	1	-none-	numeric
p.value	1	-none-	numeric
conf.int	2	-none-	numeric
estimate	2	-none-	numeric
null.value	1	-none-	numeric
alternative	1	-none-	character
method	1	-none-	character
data.name	1	-none-	character

# R object for the output

```
> print(t.obj)
```

Two Sample t-test

data: extra by group

t = -1.8608, df = 18, p-value = 0.07919

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-3.363874 0.203874

sample estimates:

mean in group 1 mean in group 2

0.75

2.33

```
> t.obj$p.value
```

```
[1] 0.07918671
```

```
> t.obj$statistic
```

t

```
-1.860813
```

```
>
```

# Discussion

- R Objects: output of the analysis.
- R functions: t.test
- \$.

## Practical session 2

- The `ToothGrowth` is a dataset available in R.
- Use `help(ToothGrowth)` for more details.
- The response variable is the Tooth length.
- Test if the Supplement type has an effect on the tooth length.

```
t.test(response ~ group, data = ...)
```

# Basic plots

# The faithful data in R

```
> help(faithful)
```

Waiting time between eruptions  
and the duration of the eruption  
for the Old Faithful geyser in  
Yellowstone National Park,  
Wyoming, USA.

```
> Faithful
```

	eruptions	waiting
1	3.600	79
2	1.800	54
3	3.333	74
4	2.283	62
5	4.533	85
6	2.883	55
7	4.700	88
8	3.600	85
9	1.950	51

# The faithful data in R

```
> faithful$eruption
```

```
[1] 3.600 1.800 3.333 2.283 4.533 2.883 4.700 3.600 1.950  
4.350 1.833 3.917 , ..., .4.750 4.117 2.150 4.417 1.817 4.467
```

```
> mean(faithful$eruption)
```

```
[1] 3.487783
```

```
faithful$eruption
```

```
> mean(x)
```

```
[1] 3.487783
```

```
> median(x)
```

```
[1] 4
```

```
> range(x)
```

```
[1] 1.6 5.1
```

```
> min(x)
```

```
[1] 1.6
```

```
> max(x)
```

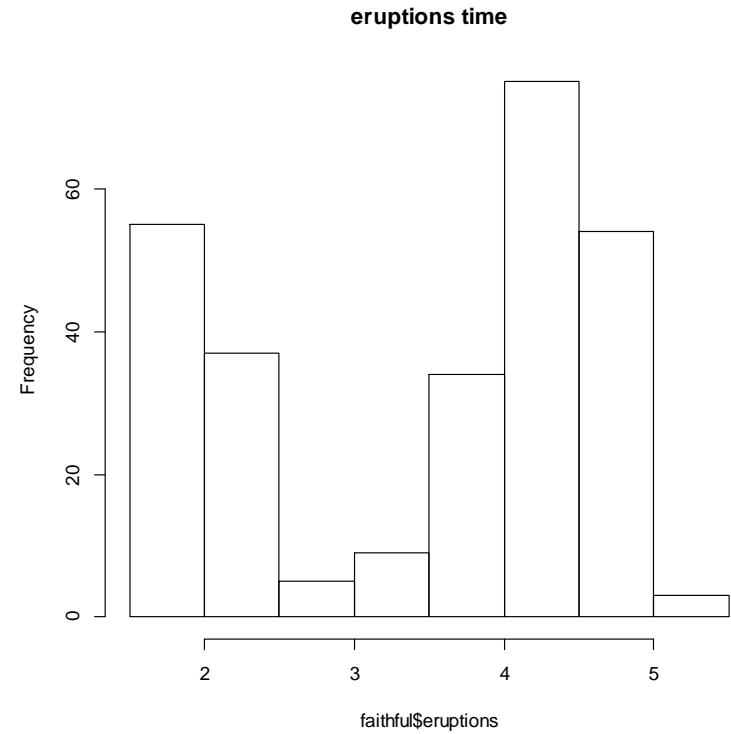
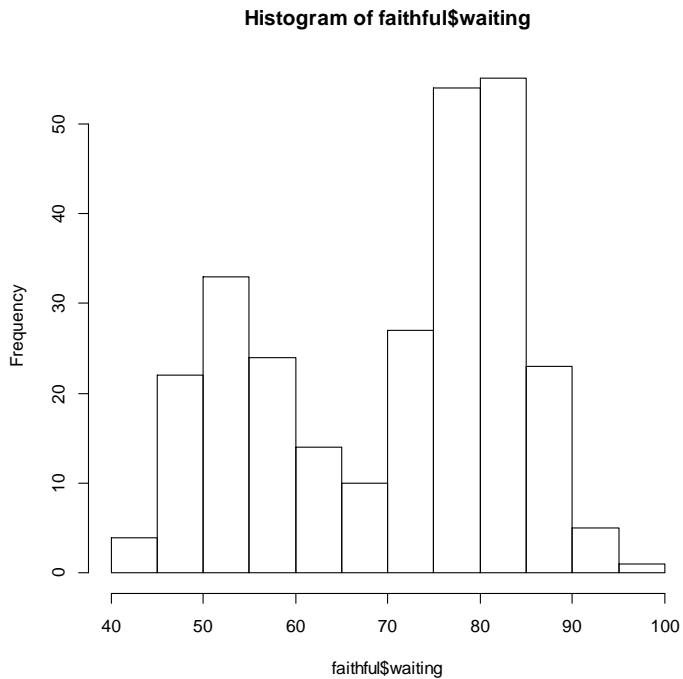
```
[1] 5.1
```



# Basic plot

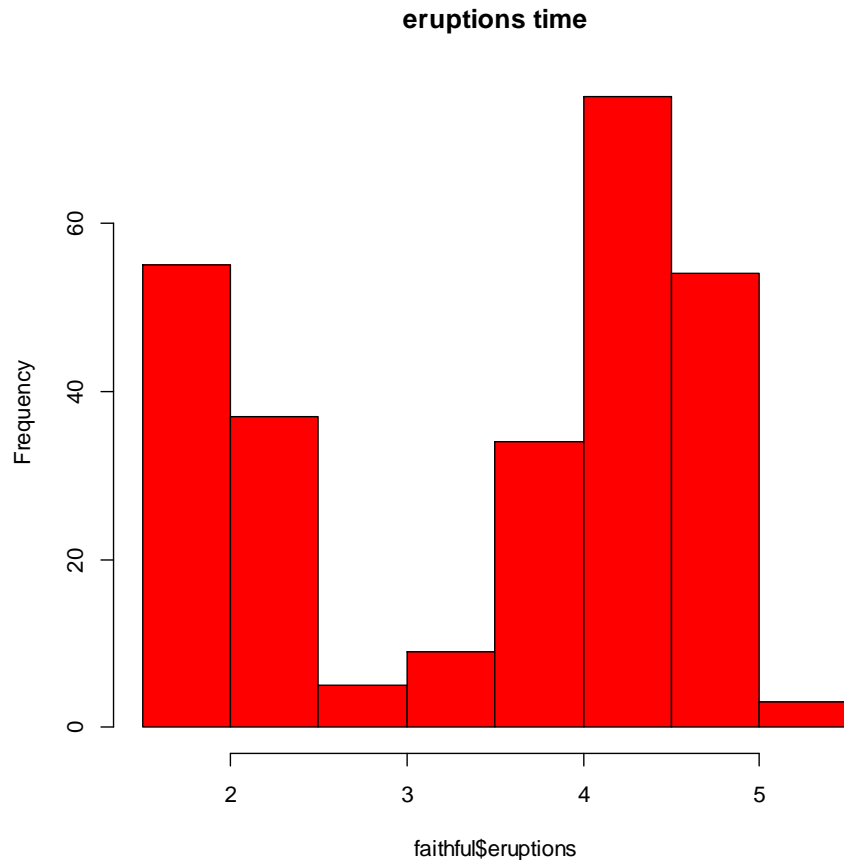
```
> hist(faithful$waiting)
```

```
> hist(faithful$eruptions,  
      main="eruptions time")
```

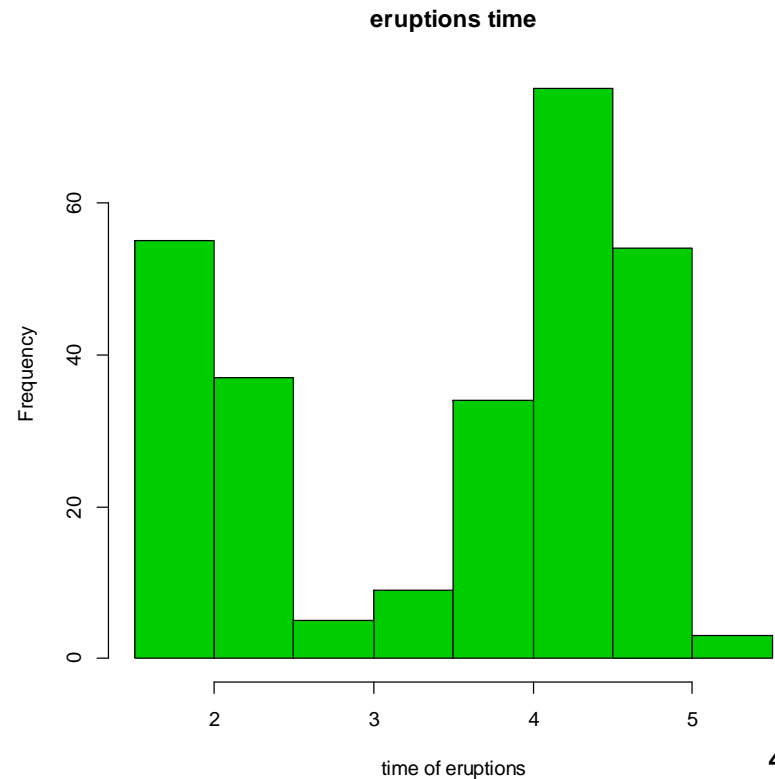


# Basic plot

```
>hist(faithful$eruptions,  
main="eruptions time",  
col=2)
```

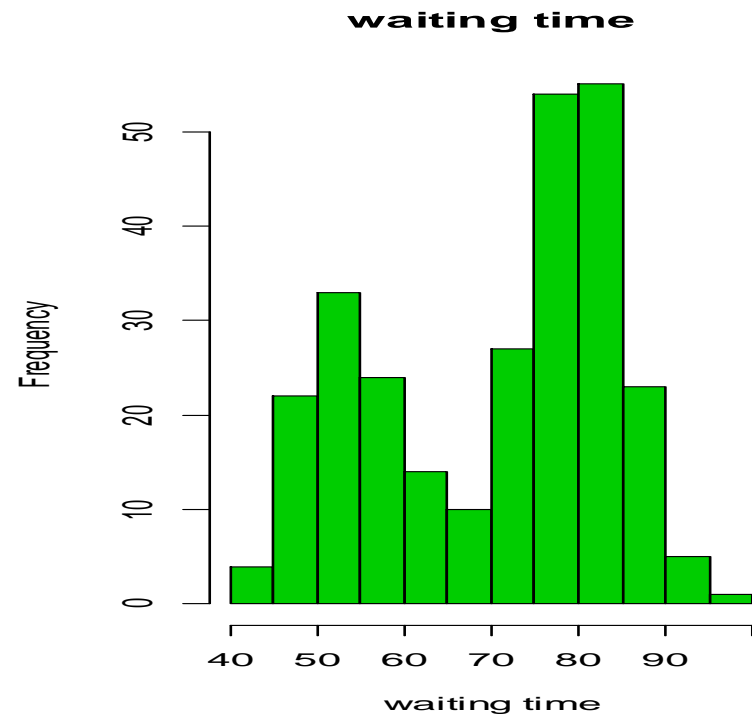
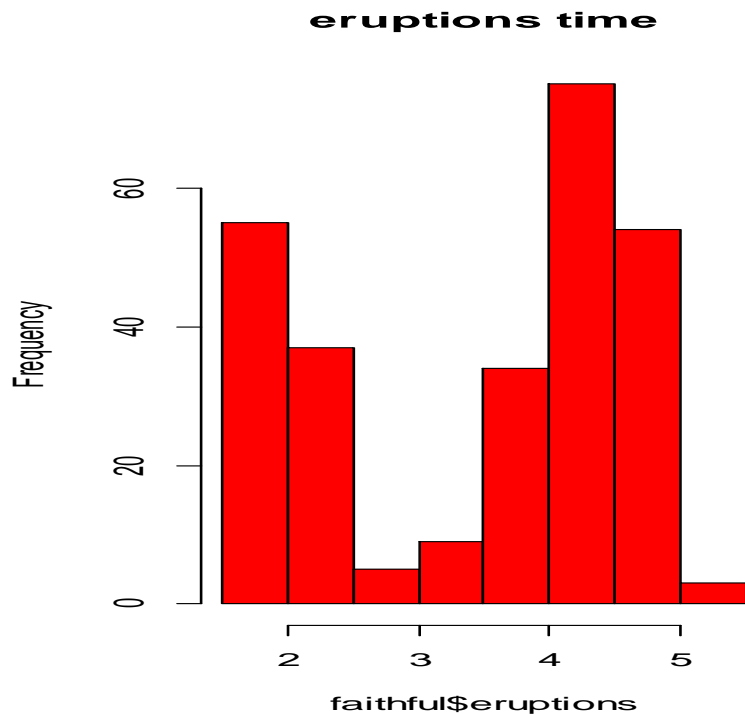


```
>hist(faithful$eruptions,  
main="eruptions time",  
col=3,  
xlab="time of eruptions")
```



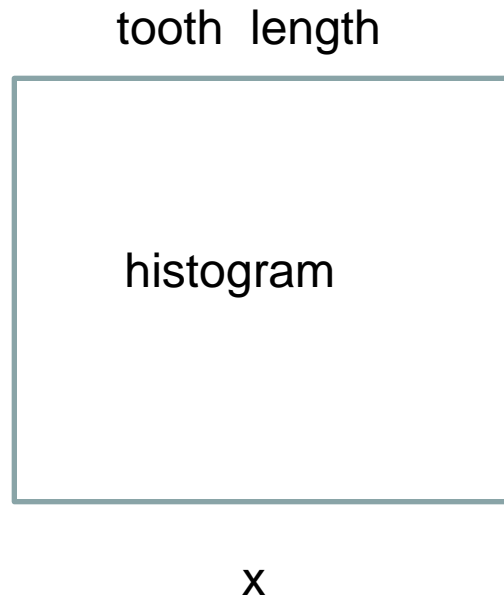
# Basic plot

```
> mfrow=c(1,2))  
> hist(faithful$eruptions,main="eruptions time",col=2)  
> hist(faithful$waiting,main="waiting  
time",col=3,xlab="waiting time")
```



## Practical session 3

- Use the **ToothGrowth** data.
- Produce an histogram for the tooth length with the following structure.



# **Chapter 1**

## **Basic programming**

### **Objects in R**

# Simple objects

Assign the value of 5 to  
the R object x



```
> x <- 5
```

```
> x
```

```
[1] 5
```

```
> x^2
```

```
[1] 25
```

```
> x + 6
```

```
[1] 11
```

# Vectors

```
> x<-c("A","A","A","A","B","B","B","B")  
> x  
[1] "A" "A" "A" "A" "B" "B" "B" "B"
```

```
> y<-c(10,11,9,15,3,5,7,2)  
> y  
[1] 10 11  9 15  3  5  7  2
```

# Index vectors

```
y[ x=="A" ]
```

All the elements in y  
for which x=A

```
> ya<-y[ x=="A" ]  
> ya  
[1] 10 11 9 15
```

```
> yb<-y[ x=="B" ]  
> yb  
[1] 3 5 7 2
```

```
> tapply(y,x,mean)  
      A      B  
11.25  4.25
```



# Data frames

A data structure  
which contains more  
than 1 object.

Objects can be  
numeric objects and  
character objects

```
> z<-data.frame(x,y)
```

```
> z
```

	x	y
1	A	10
2	A	11
3	A	9
4	A	15
5	B	3
6	B	5
7	B	7
8	B	2

# The \$

The object x in z

```
> z$x
```

```
[1] A A A A B B B B
```

```
Levels: A B
```

```
> z$y
```

```
[1] 10 11 9 15 3 5 7 2
```

# Matrix

```
> w<-c(1,2,40,2,3,9,200,4,6000)
> matw<-matrix(w,3,3)
> matw
```

	[,1]	[,2]	[,3]
[1,]	1	2	200
[2,]	2	3	4
[3,]	40	9	6000

# Rows and columns

$X_{ij}=x[i,j]$

```
> w1<-matw[1,]  
> w2<-matw[,2]  
> w1  
[1]    1    2 200  
> w2  
[1]  2  3  9
```

# The matrix reloaded

```
> matw+10
```

	[,1]	[,2]	[,3]
[1,]	11	12	210
[2,]	12	13	14
[3,]	50	19	6010

```
>
```

[1]	1	3	6000
-----	---	---	------

# The inverse matrix

```
> solve(matw)
```

	[,1]	[,2]	[,3]
[1,]	-0.687854189	0.39056517	0.0226680962
[2,]	0.453361924	0.07658141	-0.0151631184
[3,]	0.003905652	-0.00271864	0.0000382907

```
> solve(matw)%*%(matw)
```

	[ ,1]	[ ,2]	[ ,3]
[1, ]	1.0000000e+00	9.714451e-17	1.998401e-15
[2, ]	5.551115e-17	1.0000000e+00	-8.104628e-15
[3, ]	4.336809e-19	-4.336809e-19	1.0000000e+00

# Example: data frame

```
> x<-c(25,36,21)  
> gender<-c("M","M","F")  
> data.frame(x,gender)
```

	x	gender
1	25	M
2	36	M
3	21	F

# Example: an R object of a data frame

```
> x<-c(25,36,21)
> gender<-c("M","M","F")
> xdat<-data.frame(x,gender)
> xdat
  x gender
1 25      M
2 36      M
3 21      F
> xdat$gender
[1] M M F
Levels: F M
```



# Practical session 4

- Create the folowig data frame:

A	100
B	99
C	105
D	35
E	0
F	250

# **Chapter 2**

## **Reading external datasets**

## Read an external file (**text file**)

```
> spwh3<-read.table('c:\\projects\\wseda\\spwh3.txt',  
                    header=FALSE,na.strings="NA", dec=".")
```

```
> dim(spwh3)  
[1] 60 4
```

```
> spwh3<-data.frame(spwh3)  
> names(spwh3)<-c("id","y","x1","gender")
```

## Read an external file (**csv file**)

```
> gsw.ts <- read.csv("C:/projects/NBA/GSW_TS.csv",header =  
FALSE,sep =";")
```

# The data

```
> spwh3
```

	id	y	x1	gender
1	1	10.111368	1	0
2	2	9.948930	1	0
3	3	10.322560	1	0
4	4	10.241052	1	0
5	5	9.911427	1	0
6	6	9.357969	1	0
7	7	10.649141	1	0
8	8	10.150197	1	0
9	9	9.403218	1	0
10	10	8.027072	1	0
11	11	20.020056	1	1

## The sleep data in R

```
> sleep
```

	extra	group	ID
1	0.7	1	1
2	-1.6	1	2
3	-0.2	1	3
4	-1.2	1	4
5	-0.1	1	5
.	.	.	.
14	0.1	2	4
15	-0.1	2	5
16	4.4	2	6
17	5.5	2	7
18	1.6	2	8
19	4.6	2	9
20	3.4	2	10

# Two samples t-test

```
> y1<-spwh3$y[spwh3$gender==0]  
> y2<-spwh3$y[spwh3$gender==1]  
> t.test(y1,y2)
```

Welch Two Sample t-test

data: y1 and y2

t = -9.1428, df = 58, p-value = 7.715e-13

alternative hypothesis: true difference in means is not  
equal to 0

95 percent confidence interval:

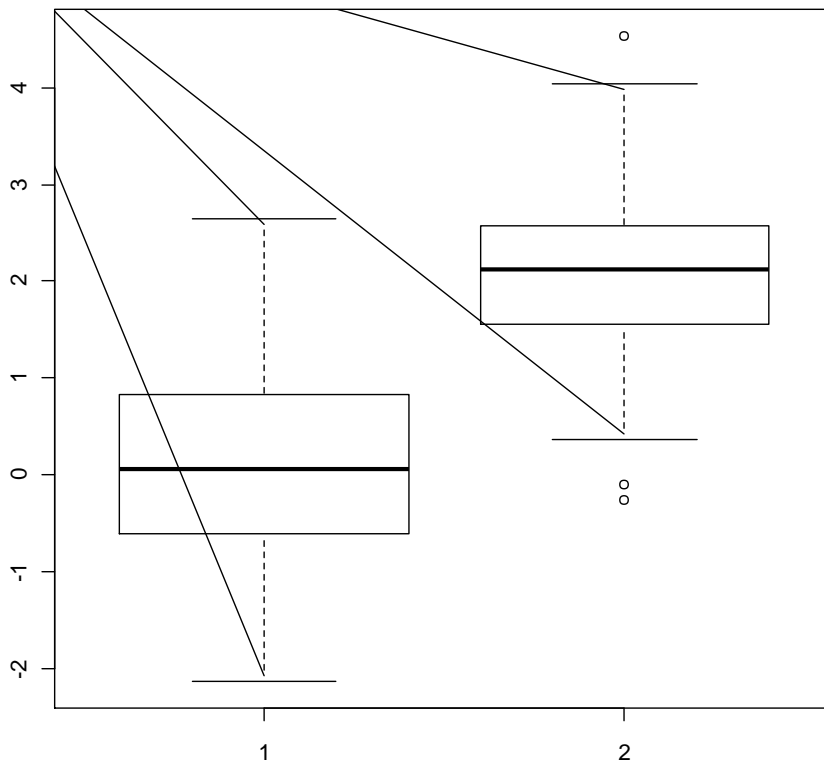
-12.229889 -7.836547

sample estimates:

mean of x mean of y

14.99933 25.03254

# Two samples t-test



```
> y1<-rnorm(100,0,1)  
> y2<-rnorm(57,2,1)  
> boxplot(y1,y2)
```

# Two samples t-test

```
> t.test(y1,y2)
```

Welch Two Sample t-test

data: y1 and y2

t = -14.2203, df = 126.176, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-2.290641 -1.730980

sample estimates:

mean of x mean of y

-0.0063866 2.0044240



# R object for the output

```
> t.t<-t.test(y1,y2)
```

```
> summary(t.t)
```

	Length	Class	Mode
statistic	1	-none-	numeric
parameter	1	-none-	numeric
p.value	1	-none-	numeric
conf.int	2	-none-	numeric
estimate	2	-none-	numeric
null.value	1	-none-	numeric
alternative	1	-none-	character
method	1	-none-	character
data.name	1	-none-	character

# R object for the output

```
> t.t
```

```
Welch Two Sample t-test
```

```
data: y1 and y2
```

```
t = -14.2203, df = 126.176, p-value < 2.2e-16
```

```
alternative hypothesis: true difference in means is not  
equal to 0
```

```
95 percent confidence interval:
```

```
-2.290641 -1.730980
```

```
sample estimates:
```

```
mean of x mean of y
```

```
-0.0063866 2.0044240
```

```
> t.t$p.value
```

```
[1] 5.570543e-28
```

```
> t.t$statistic
```

```
t
```

```
-14.22034
```

# Practical session 5

- Create the following text file:

A	100
B	99
C	105
D	35
E	0
F	250

and read it to R as an external file

# **Chapter 3**

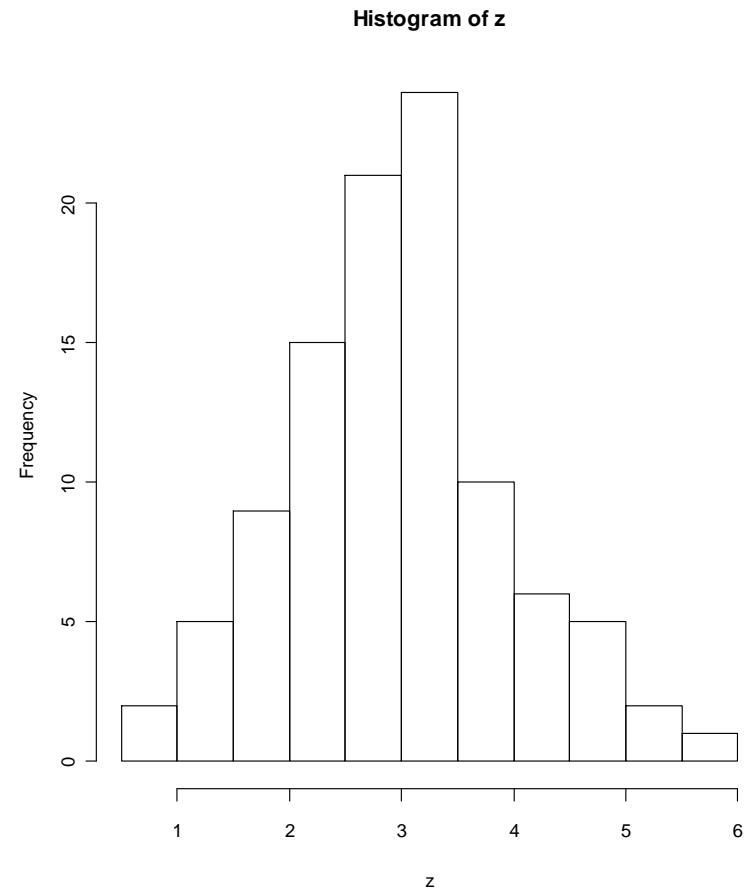
## **Basic plots functions**

# Descriptive statistics in R

```
> z<-rnorm(100,3,1) ← Generate random  
                        sample from N(3,1)  
> mean(z)  
[1] 2.979706  
> median(z)  
[1] 2.958521  
> max(z)  
[1] 5.849559  
> min(z)  
[1] 0.877219
```

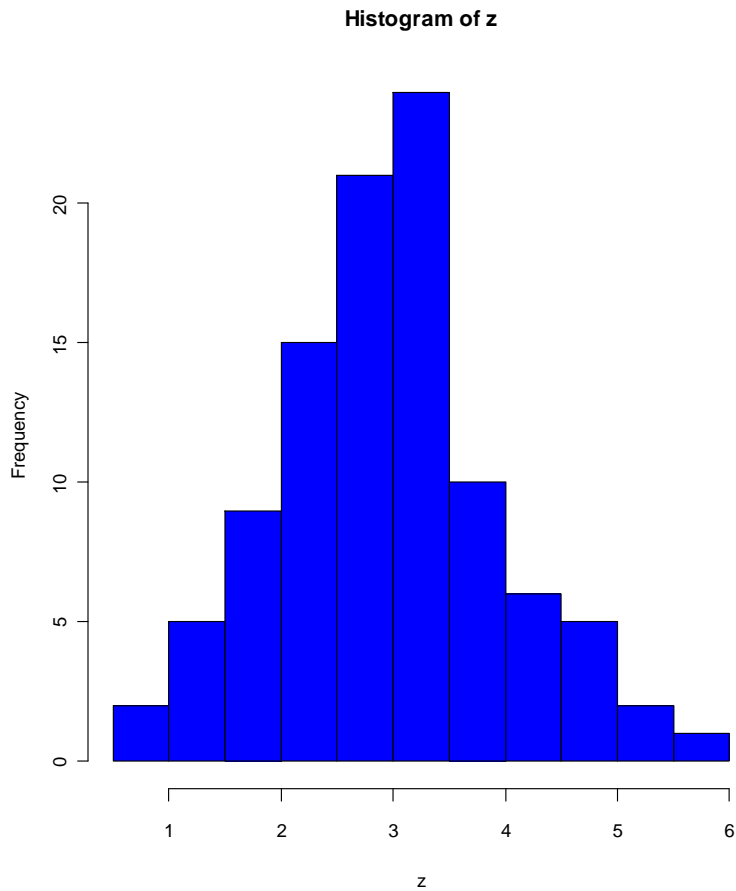
# Histogram

```
> hist(z)
```

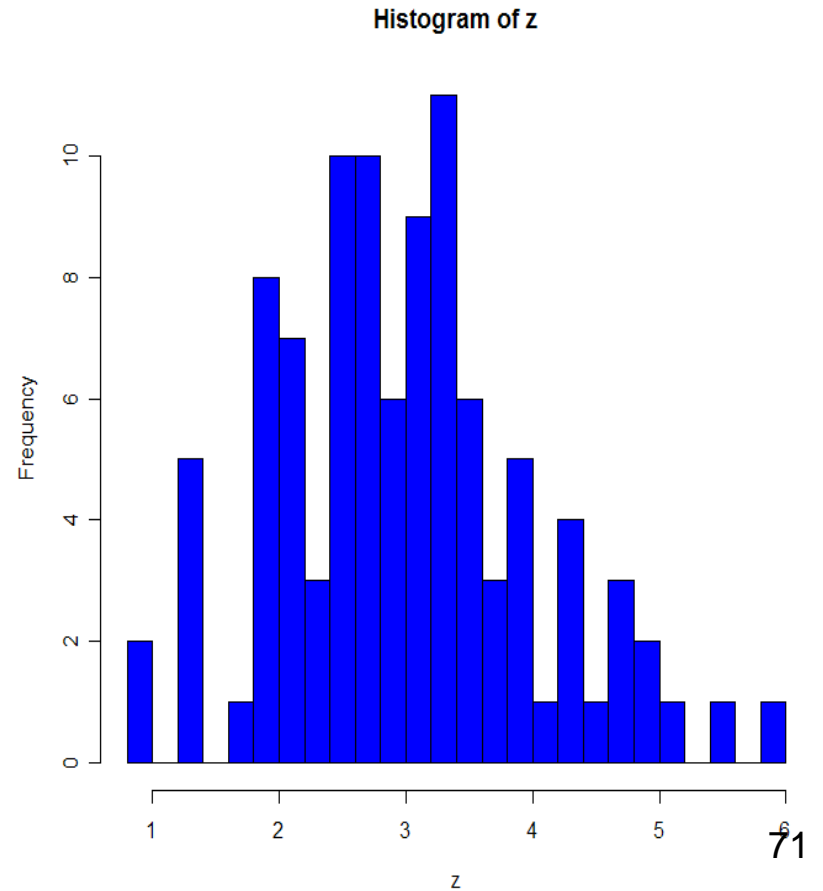


# Plot options

```
> hist(z,col=4)
```

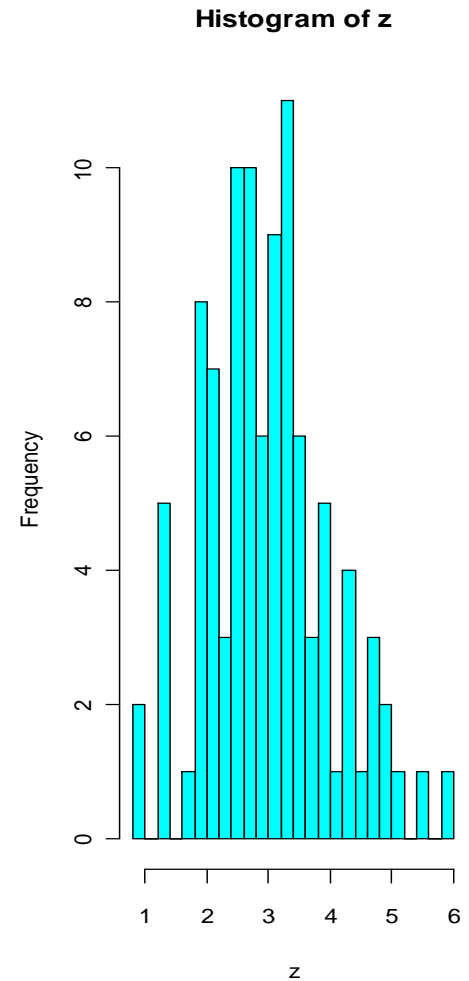
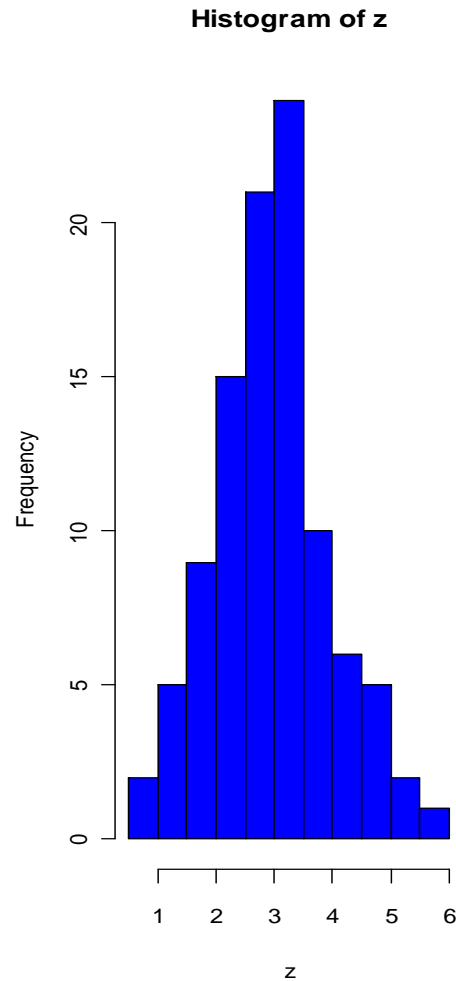
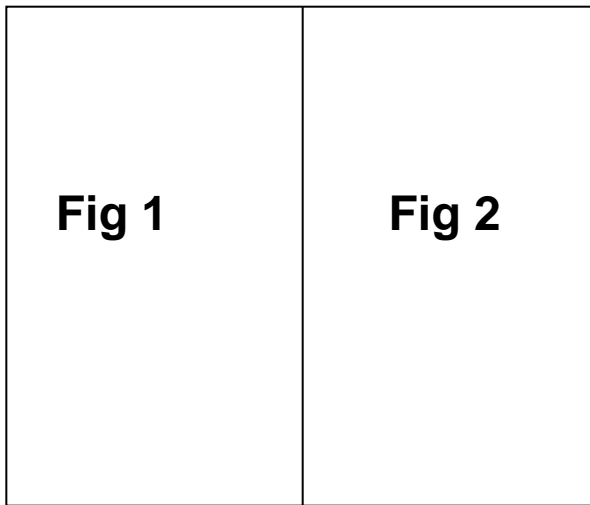


```
> hist(z,col=4,nclass=25)
```



# Controlling the graphical output: the par() function

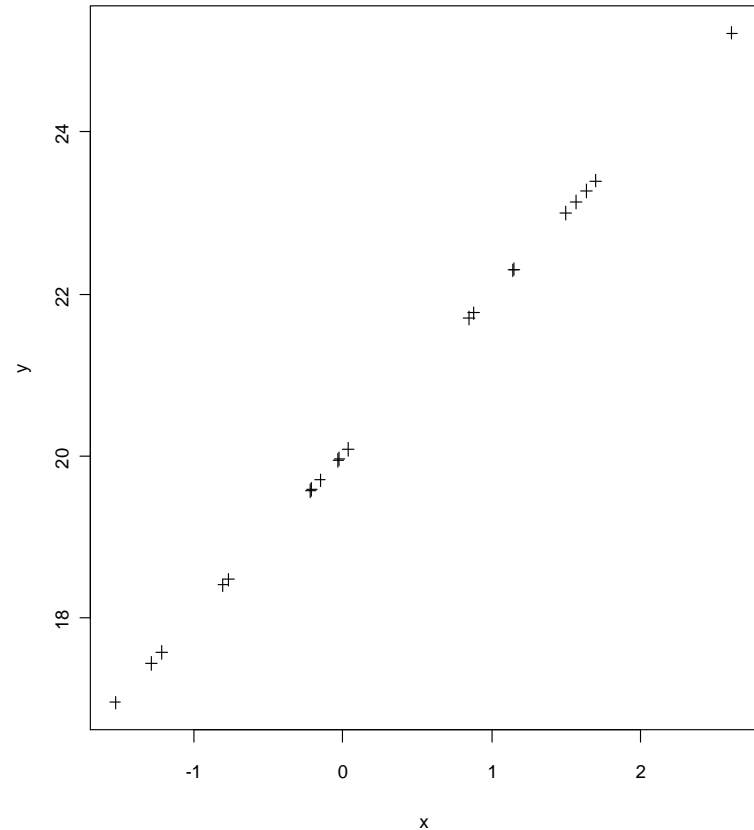
```
> par(mfrow=c(1,2))  
> hist(z,col=4)  
> hist(z,col=5,nclass=25)
```





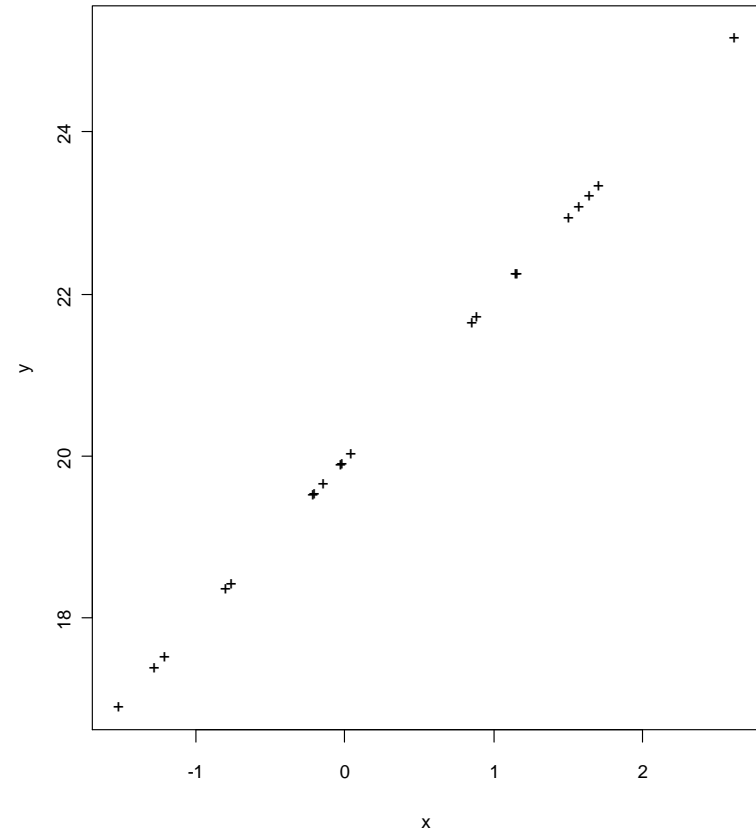
# Type of points: the pch() function

```
> y<-2*x+20  
> plot(x,y)  
> plot(x,y,pch=3)
```



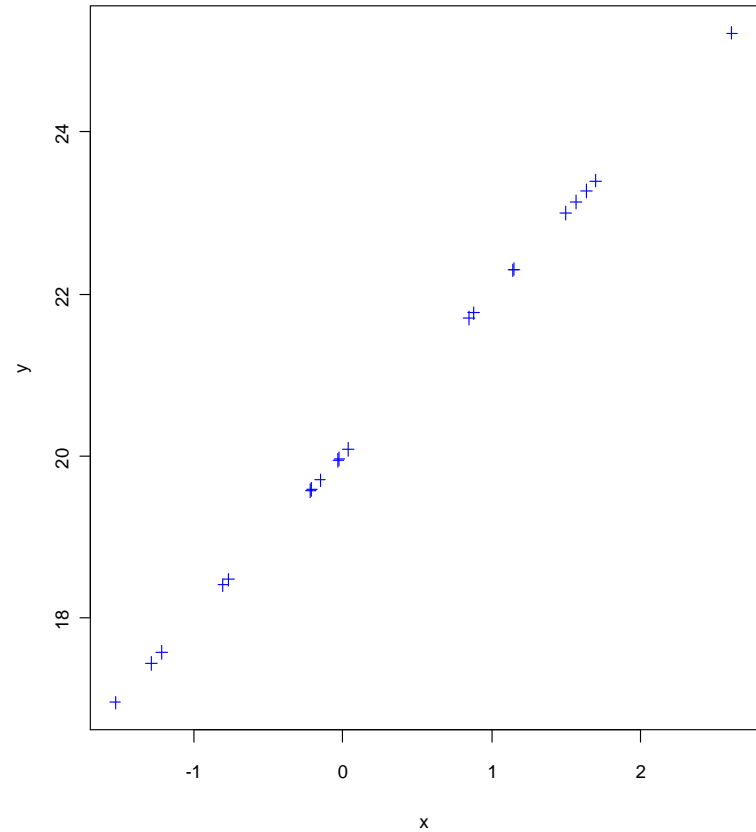
# Type of points: the pch() function

```
>plot(x,y,pch="+")
```



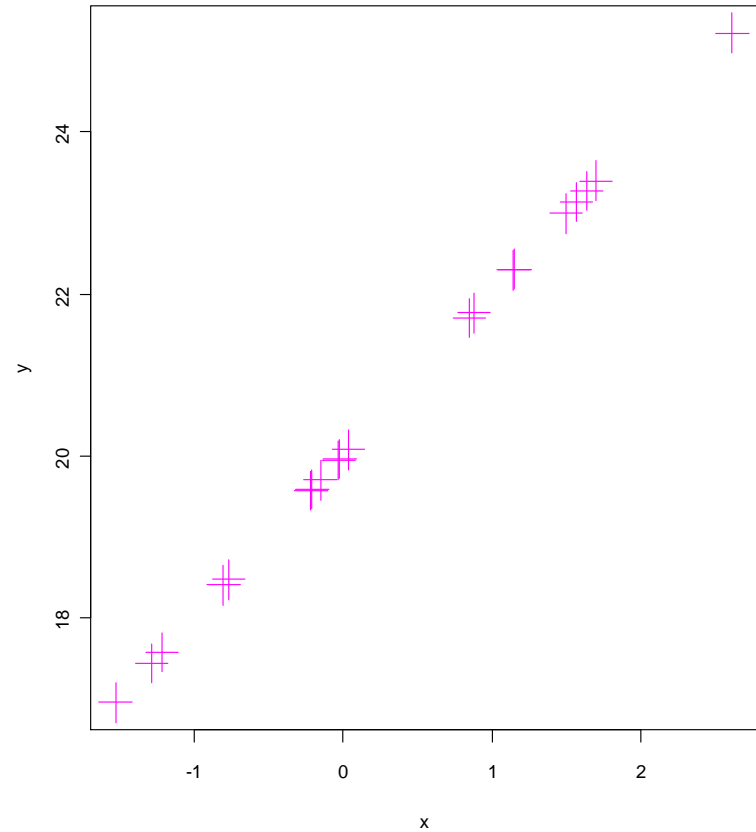
# Colors: the option col

```
> plot(x,y,pch=" ")
> points(x,y,col=4,pch=3)
```



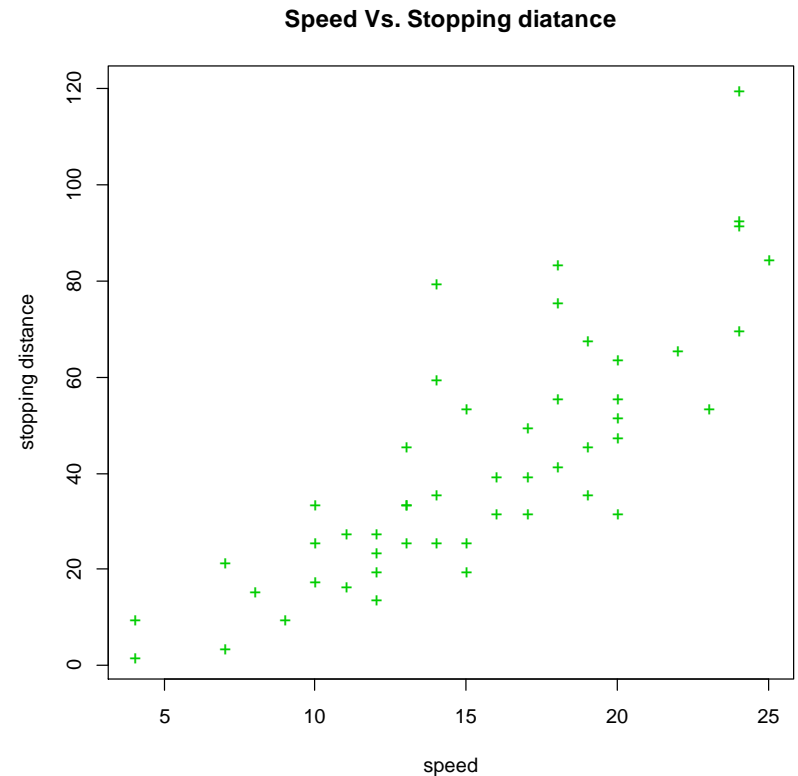
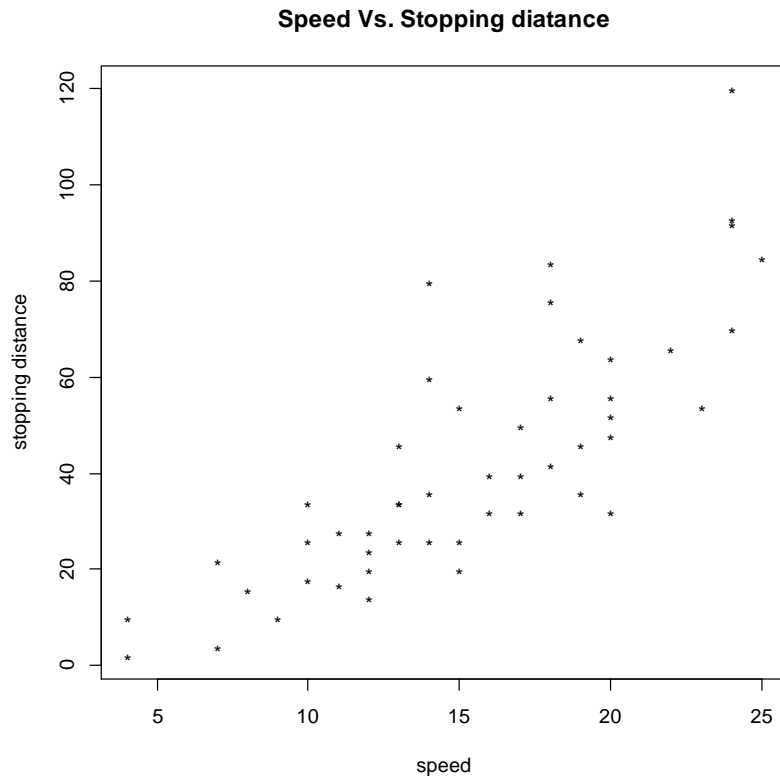
# Size: the option cex

```
> plot(x,y,pch=" ")
> points(x,y,col=6,pch=3,cex=3)
```



# Practical session 6

- Produce the following figures in R (use the cars data)



# Practical session 7

- Use the `airquality` in R.
- Produce the following plots:
  - Histogram for the ozone level
  - Boxplot for the ozone level.
- What is the mean and the median of the ozone level.
- What is the minimum and maximum ozone levels.

# **Chapter 4**

## **Programming I: A for loop**

A for loop

```
for(i in 1:B)  
{
```

*Here you ask from R to do the same  
thing B times.....*

```
}
```



# Generate 1000 samples from $N(2,1)$

```
> x<-rnorm(10,2,1)
```

```
> x
```

```
[1] 2.1531462 2.4426189 0.8080064 1.4051178 1.9392356 0.6466574  
[7] 0.7519918 -0.1097367 2.3338487 3.7598694
```

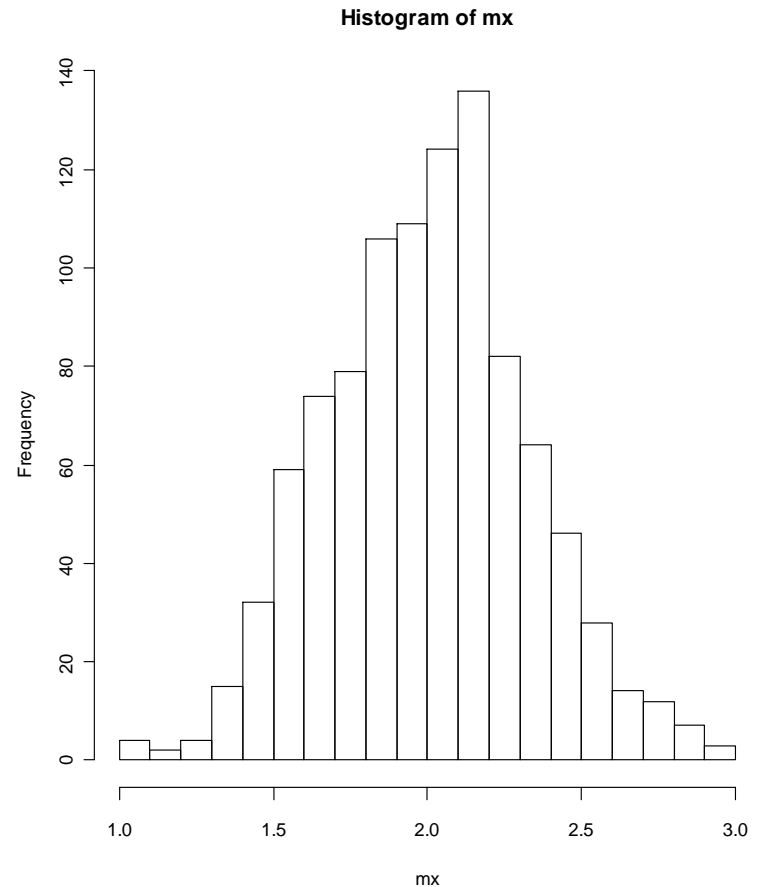
```
> x<-rnorm(10,2,1)
```

```
> x
```

```
[1] 2.9694328 1.1065506 1.5612572 0.3904008 1.6890423 3.7319756 0.9026146  
[8] 1.7763012 2.4356002 0.9643299
```

# Generate 1000 samples from $N(2,1)$

```
> mx<-c(1:1000)
> for(i in 1:1000)
+ {
+ x<-rnorm(10,2,1)
+ mx[i]<-mean(x)
+ }
> hist(mx,nclass=25)
```



# Example: distribution of the minimum in uniform distribution

- Generate 1000 samples ( $n=50$ ) from  $U(0,1)$ .
- Calculate the minimum of each sample.
- Estimate the density of the minimum.

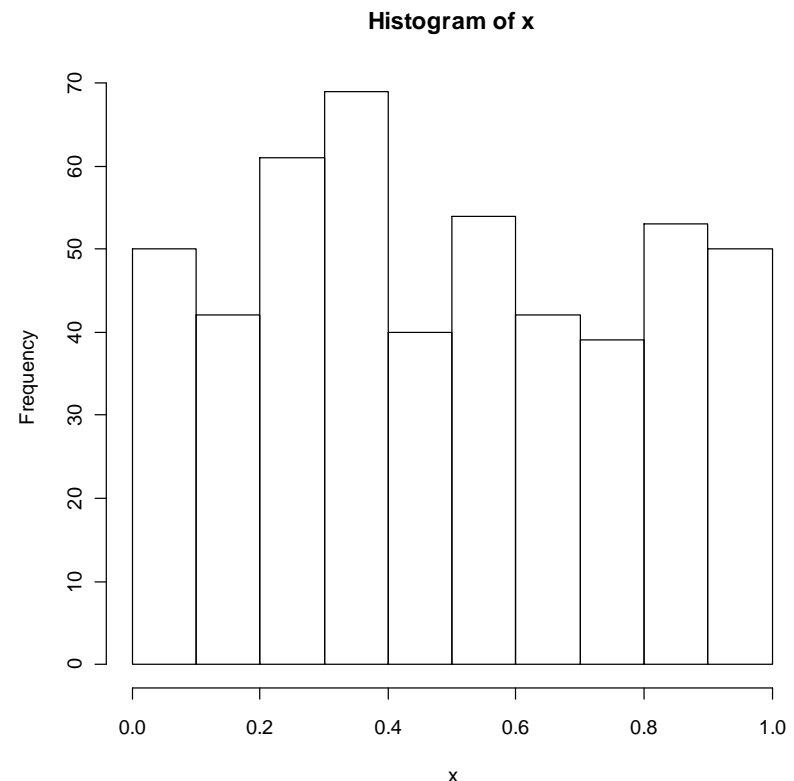
# Example: distribution of the minimum in uniform distribution

- Generate 1000 samples ( $n=50$ ) from  $U(0,1)$ .
- Calculate the minimum of each sample.
- Estimate the density of the minimum.

# Example: distribution of the minimum in uniform distribution

- Generate 1000 samples ( $n=50$ ) from  $U(0,1)$ .
- Calculate the minimum of each sample.

```
> x<-runif(500,0,1)
> hist(x)
> min(x)
[1] 0.004631357
```



# Example: distribution of the minimum in uniform distribution

- Estimate the density of the minimum.

```
for(i in 1:B)  
{
```

Generate 1000 samples ( $n=50$ ) from  $U(0,1)$ .  
Calculate the minimum of each sample.

```
}
```

# Example: distribution of the minimum in uniform distribution

- Estimate the density of the minimum.

```
for(i in 1:B)
{
  Generate 1000 samples (n=50) from U(0,1).
  Calculate the minimum of each sample.
}
```

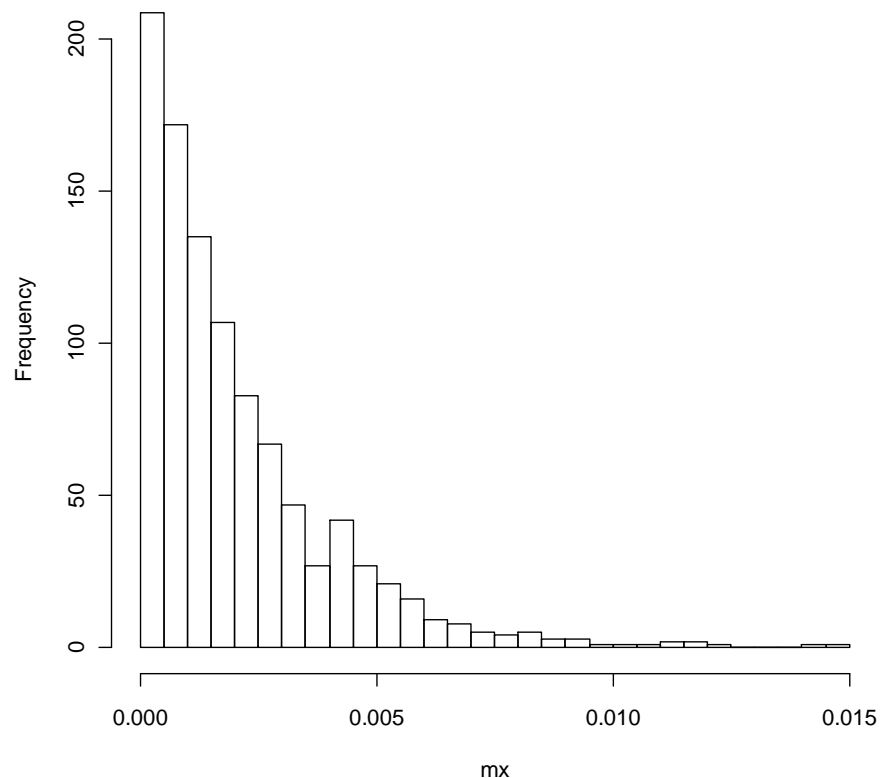
```
> mx<-c(1:1000)
> for(i in 1:1000)
+ {
+   x<-runif(500,0,1)
+   mx[i]<-min(x)
+ }
```

# Example: distribution of the minimum in uniform distribution

- Estimate the density of the minimum.

```
> mx<-c(1:1000)
> for(i in 1:1000)
+ {
+ x<-runif(500,0,1)
+ mx[i]<-min(x)
+ }
>hist(mx)
```

Histogram of mx





# Practical session 8

- Make a for loop that print your name 500 times.

# **Chapter 5**

## **Statistical modeling 1: Simple linear regression**

# Reading the cars data

```
> carsdat<-read.table('c:\\projects\\wseda\\Rintro\\cars.txt',  
  header=FALSE,na.strings="NA", dec=".")  
> dim(carsdat)  
[1] 50  3
```

The data is available in R, use, `help(cars)`

# The cars data

```
> help(cars)
```

```
cars                                package:datasets                R Documentation
```

```
Speed and Stopping Distances of Cars
```

```
Description:
```

```
    The data give the speed of cars and the distances taken to stop.  
    Note that the data were recorded in the 1920s.
```

```
Usage:
```

```
cars
```

```
Format:
```

```
    A data frame with 50 observations on 2 variables.
```

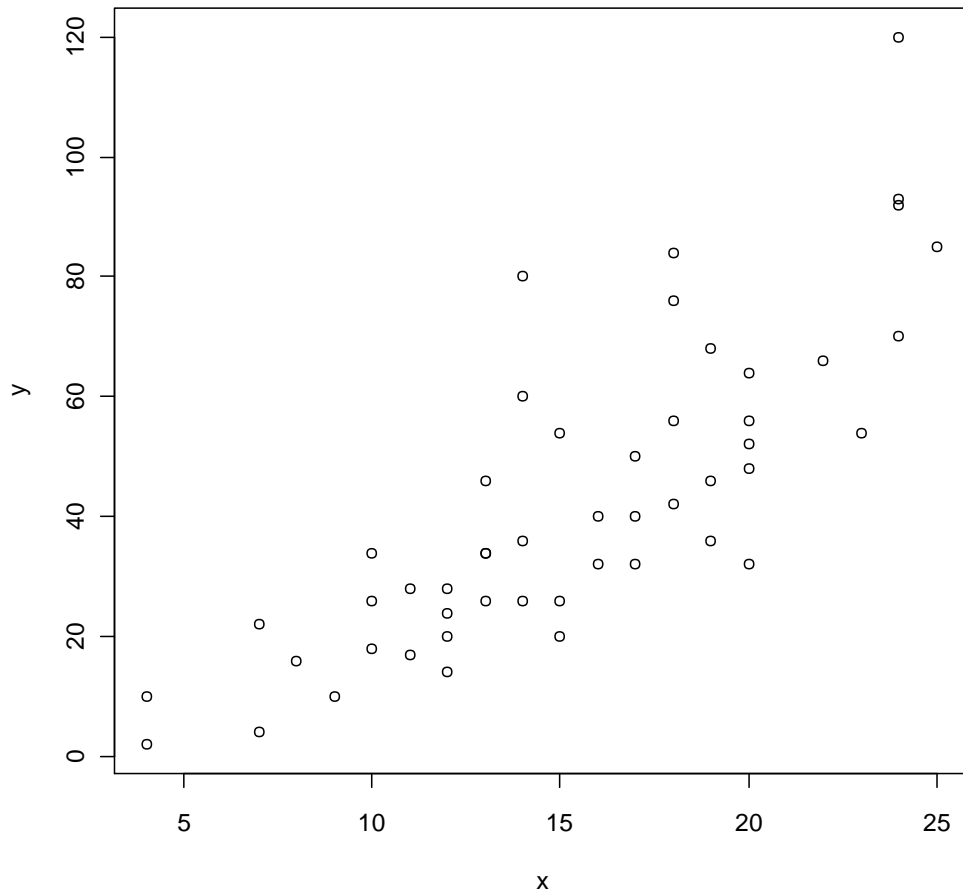
```
    [,1] speed  numeric  Speed (mph)  
    [,2] dist   numeric  Stopping distance (ft)
```

```
Source:
```

```
    Ezekiel, M. (1930) Methods of Correlation Analysis.  Wiley.
```

# The cars data

```
> x<-carsdat[,2]  
> y<-carsdat[,3]  
> plot(x,y)
```



# The lm() function

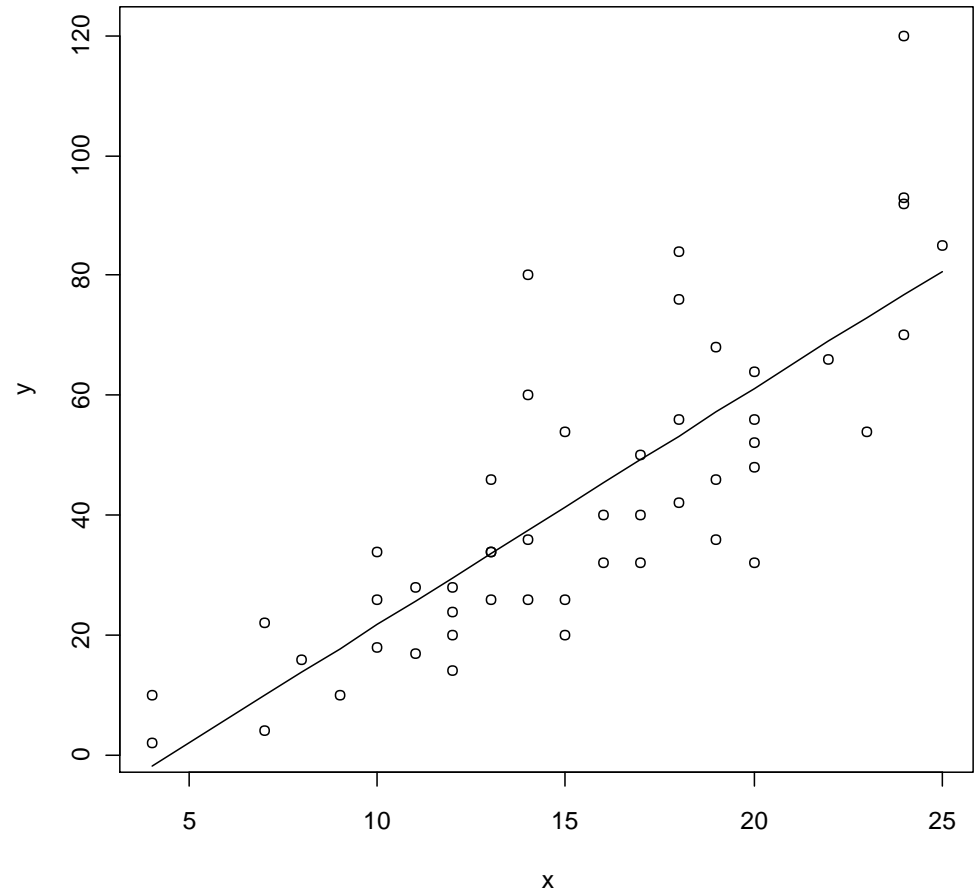
$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

General call of the `lm()` function

```
lm(response~predictor)
```

# Data and predicted model

```
> fit.1<-lm(y~x)
> plot(x,y)
> lines(x,fit.1$fit)
```



# The “output”

## ANOVA table for the model

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x	1	21186	21185.5	89.567	1.490e-12
***					
Residuals	48	11354	236.5		

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’  
0.05 ‘.’ 0.1 ‘ ’ 1

```
> summary(fit.1)
```

Call:

```
lm(formula = y ~ x)
```

Residuals:

Min	1Q	Median	3Q	Max
-29.069	-9.525	-2.272	9.215	43.201

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-17.5791	6.7584	-2.601	0.0123 *
x	3.9324	0.4155	9.464	1.49e-12 ***

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’  
0.1 ‘ ’ 1

Residual standard error: 15.38 on 48 degrees of freedom

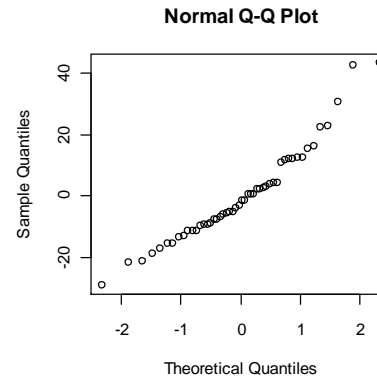
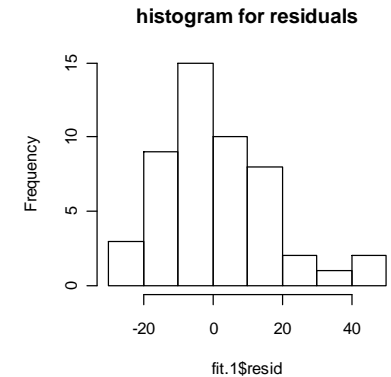
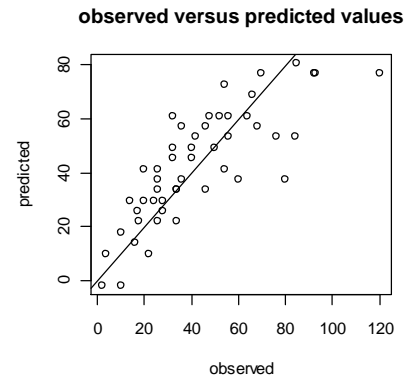
Multiple R-squared: 0.6511, Adjusted R-squared:  
0.6438

F-statistic: 89.57 on 1 and 48 DF, p-value: 1.490e-12



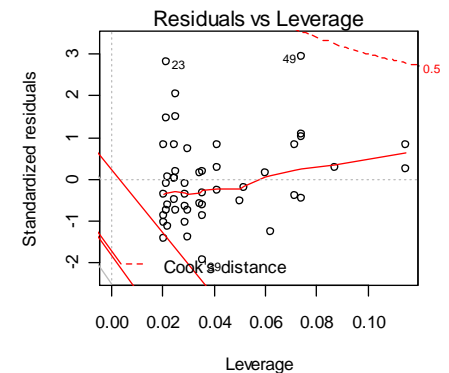
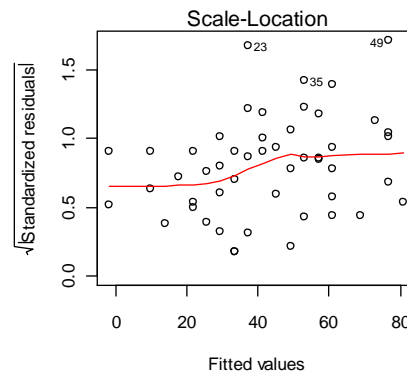
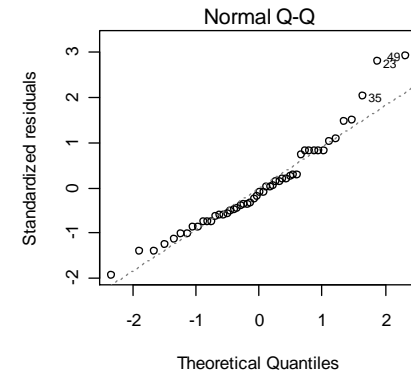
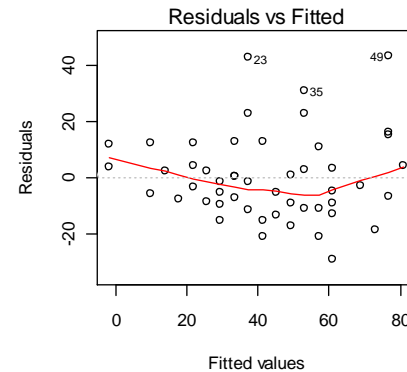
# Graphical output

```
> par(mfrow=c(2,2))
> plot(y,fit.1$fit,xlab="observed",
      ylab="predicted")
> abline(0,1)
> title("observed versus predicted
      values")
> hist(fit.1$resid,col=0,main=" ")
> title("histogram for residuals")
> qqnorm(fit.1$resid)
```



# Default plots

```
> plot(fit.1)
```



## Practical session 9

- The **airquality** is a dataset available in R.
- Fit a simple linear regression model in which the ozone level is the response and the wind speed is the predictor.
- Test the hypothesis that the slope is zero.
- Use the default plots of an `lm()` object to produce the diagnostic plot.

# Chapter 6

## Statistical modeling 2: One way ANOVA

Examples:

The chick data

The cash data

# Example 1: The chick dataset in R

```
> chickwts
```

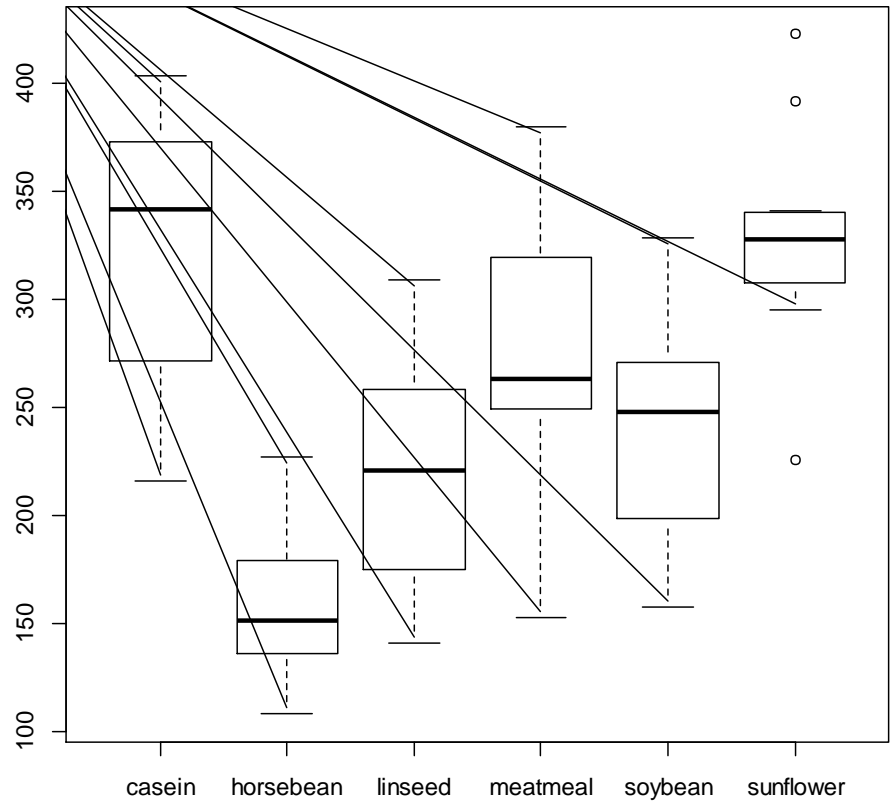
	weight	feed
1	179	horsebean
2	160	horsebean
3	136	horsebean
4	227	horsebean
16	203	linseed
17	148	linseed
18	169	linseed
23	243	soybean
24	230	soybean
25	248	soybean

```
> help(chickwts)
```

An experiment was conducted to measure the effectiveness of various feed supplements on the growth rate of chickens.

# Boxplot by group

```
> w<-chickwts[,1]  
> feed<-chickwts[,2]  
> boxplot(split(w,feed))
```



# Mean by group

```
> tapply(w, feed, mean)
```

casein	horsebean	linseed	meatmeal	soybean	sunflower
323.5833	160.2000	218.7500	276.9091	246.4286	328.9167

# One-Way ANOVA model: model formulation

$$Y_{ij} = \mu_i + \varepsilon_{ij}$$

Parameters: fixed but unknown and needed to be estimated

Random error, assumed to follow normal distribution with constant variance.

$$\varepsilon_{ij} \sim N(0, \sigma^2)$$

Model assumptions are:

1. The random error is normal distributed.
2. The variance is constant across the factor levels.



# The Null Hypothesis: No diet effect

- For a model in which the factor has 5 (the diet group) levels we wish to test the null hypothesis:

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6$$

- This means that we want to test if the means across all factor levels are equal.
- Mind that: we test if the parameters ( $\mu_j$ ) are equal, not is the sample means ( $\bar{Y}_j$ ).

# Test Statistic

Within group sum of squares

$$SSW = \sum_{i=1}^I \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i.})^2$$

Between group sum of squares

$$SSB = \sum_{i=1}^I n_i (\bar{Y}_{i.} - \bar{Y}_{..})^2$$

$$F = \frac{SSB / (I - 1)}{SSW / (N - I)} = \frac{MSB}{MSW}$$

The test statistic,  $F$ , is the ratio between the mean of the between sum of squares (SSB) and the mean of the within sum of squares.

# The aov() function

$$Y_{ij} = \mu_i + \varepsilon_{ij}$$

```
aov(response ~ factor)
```

```
> a.model=aov(w~feed)  
> summary(a.model)
```

# Test Statistic

Between group sum of squares/dgree of fredom

Within group sum of squares/dgree of fredom

$$\frac{SSB / (I - 1)}{SSW / (N - I)} = \frac{MSB}{MSW} = F$$

```
> a.model=aov(w~feed)
```

```
> summary(a.model)
```

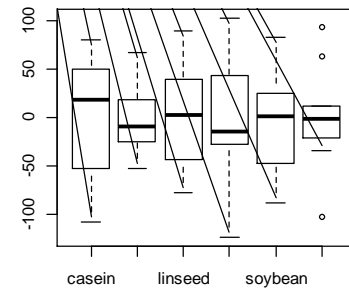
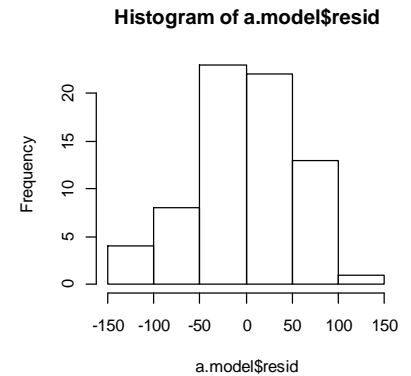
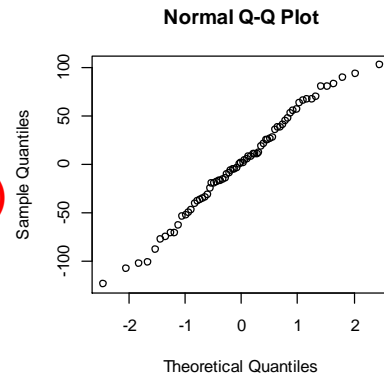
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
feed	5	231129	46226	15.37	5.94e-10 ***
Residuals	65	195556	3009		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Diagnostic plot

```
> par(mfrow=c(2,2))  
> qqnorm(a.model$resid)  
> hist(a.model$resid,col=0)  
> boxplot(split(a.model$resid,feed))
```



# One-Way ANOVA model: alternative model formulation

$$Y_{ij} = \mu_0 + \alpha_i + \varepsilon_{ij}$$

Mean of the  
reference group

Diet effect

$$\sum_{i=1}^I \alpha_i = 0$$

Random error,  
assumed to follow  
normal distribution  
with constant  
variance.

$$\varepsilon_{ij} \sim N(0, \sigma^2)$$

Model assumptions are:

1. The random error is normal distributed.
2. The variance is constant across the factor levels.

## Estimation of the model in R

$$Y_{ij} = \mu_0 + \alpha_i + \varepsilon_{ij}$$

`lm(response~predictor)`

```
> lm.fit<-lm(w~feed)
```

# Estimation of the model in R

```
> summary(lm.fit)
```

Call:

```
lm(formula = w ~ feed)
```

Residuals:

Min	1Q	Median	3Q	Max
-123.909	-34.413	1.571	38.170	103.091

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	323.583	15.834	20.436	< 2e-16	***
feedhorsebean	-163.383	23.485	-6.957	2.07e-09	***
feedlinseed	-104.833	22.393	-4.682	1.49e-05	***
feedmeatmeal	-46.674	22.896	-2.039	0.045567	*
feedsoybean	-77.155	21.578	-3.576	0.000665	***
feedsunflower	5.333	22.393	0.238	0.812495	
---					

323.583: the mean of  
the casein

323.583 - 163.383 =  
160.2000, the mean of  
the horsebeen group

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 54.85 on 65 degrees of freedom  
Multiple R-squared: 0.5417, Adjusted R-squared: 0.5064  
F-statistic: 15.36 on 5 and 65 DF, p-value: 5.936e-10



# The AVOVA table

Residual standard error: 54.85 on 65 degrees of freedom  
Multiple R-squared: 0.5417, Adjusted R-squared: 0.5064  
F-statistic: 15.36 on 5 and 65 DF, p-value: 5.936e-10

```
> anova(lm.fit)
```

Analysis of Variance Table

Response: w

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
feed	5	231129	46226	15.365	5.936e-10 ***
Residuals	65	195556	3009		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

$$54.85 = \sqrt{3009}$$

## Example 2: Reading the cash data

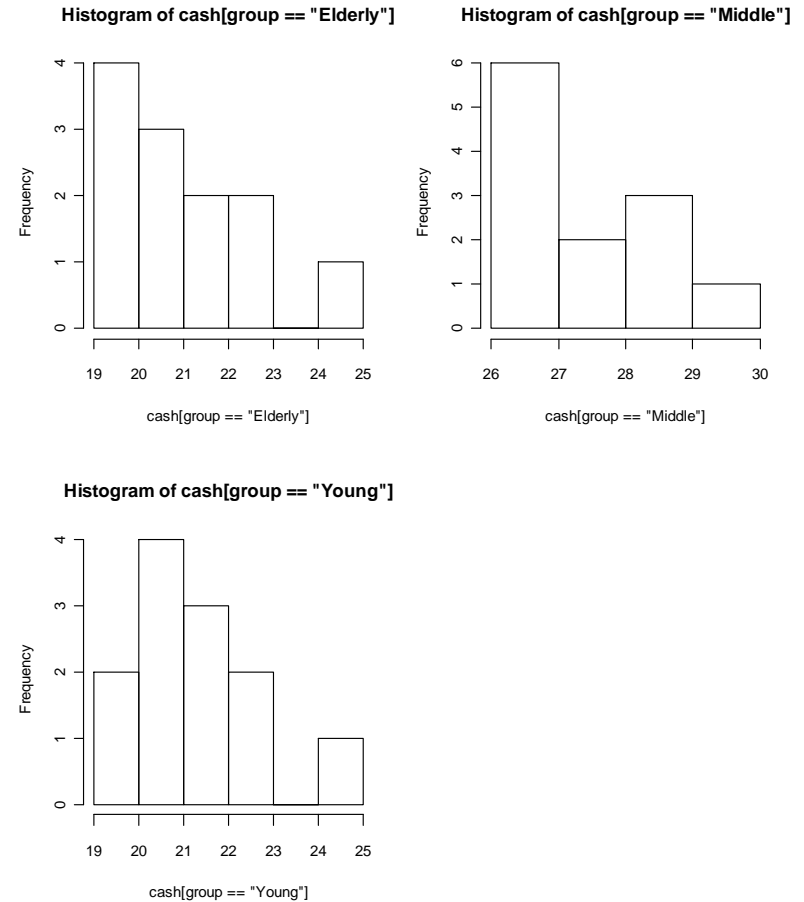
```
> cashdat<-  
  read.table('c:\\projects\\wseda\\Rintro\\cashdat.txt',  
    header=FALSE,na.strings="NA", dec=".")  
> dim(cashdat)  
[1] 36  2  
> names(cashdat)<-c("cash","group")  
> attach(cashdat)
```

# The data

```
> print(cashdat)
   cash  group
1    23  Young
2    25  Young
.      .      .
.      .      .
11   21  Young
12   21  Young
13   28 Middle
.      .      .
.      .      .
24   29 Middle
25   23 Elderly
26   20 Elderly
35   22 Elderly
36   21 Elderly
```

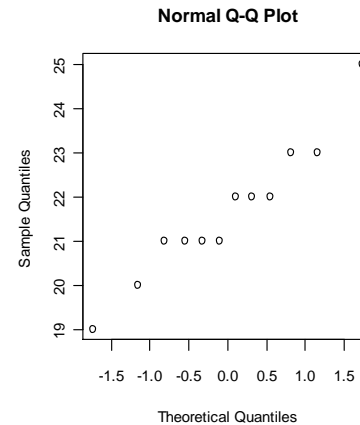
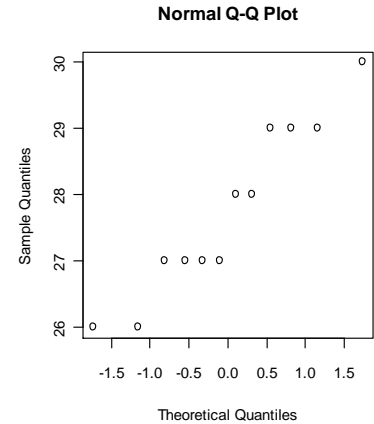
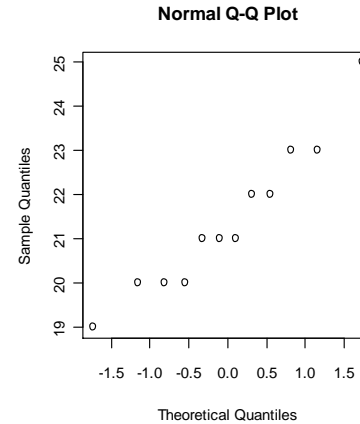
# Histograms by group

```
> par(mfrow=c(2,2))  
> hist(cash[group=="Elderly"],col=0)  
> hist(cash[group=="Middle"],col=0)  
> hist(cash[group=="Young"],col=0)
```



# qq normal plots by group

```
> par(mfrow=c(2,2))  
> qqnorm(cash[group=="Elderly"])  
> qqnorm(cash[group=="Middle"])  
> qqnorm(cash[group=="Young"])
```

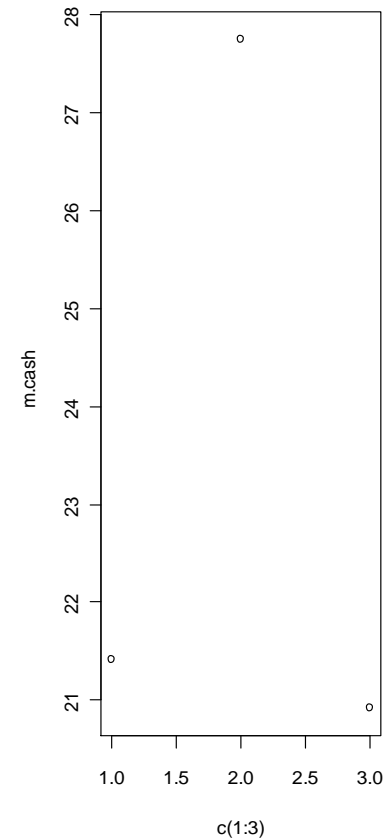
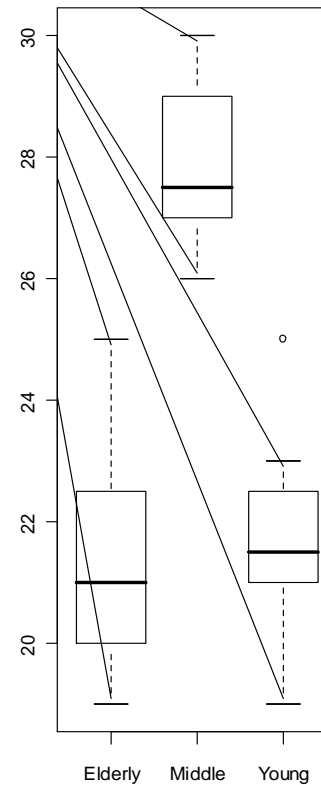


# Boxplot and dotplot

```
> par(mfrow=c(1,2))  
> boxplot(split(cash,group))
```

```
> tapply(cash,group,mean)  
Elderly    Middle    Young  
21.41667  27.75000  21.66667
```

```
> m.cash<-c(21.41667,27.75,20.91667)  
> names1<-c("Elderly","Middle","Young")  
> plot(c(1:3),m.cash)
```



# One-Way ANOVA model: model formulation

$$Y_{ij} = \mu_i + \varepsilon_{ij}$$

Parameters: fixed but unknown and needed to be estimated

Random error, assumed to follow normal distribution with constant variance.

$$\varepsilon_{ij} \sim N(0, \sigma^2)$$

Model assumptions are:

1. The random error is normal distributed.
2. The variance is constant across the factor levels.

# The Null Hypothesis: No treatment effect

- For a model in which the factor has three levels we wish to test the null hypothesis:

$$H_0 : \mu_1 = \mu_2 = \mu_3$$

- This means that we want to test if the means across all factor levels are equal.
- Mind that: we test if the parameters ( $\mu_j$ ) are equal, not is the sample means ( $\bar{Y}_j$ ).



# Test Statistic

Within group sum of squares

$$SSW = \sum_{i=1}^I \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i.})^2$$

Between group sum of squares

$$SSB = \sum_{i=1}^I n_i (\bar{Y}_{i.} - \bar{Y}_{..})^2$$

$$F = \frac{SSB / (I - 1)}{SSW / (N - I)} = \frac{MSB}{MSW}$$

The test statistic,  $F$ , is the ratio between the mean of the between sum of squares (SSB) and the mean of the within sum of squares.

# The aov() function

$$Y_{ij} = \mu_i + \varepsilon_{ij}$$

```
aov(response ~ factor)
```

```
>Fit.aov<-aov(cash~group)  
>summary(Fit.aov)
```

# Test Statistic

Between group sum of squares/dgree of fredom

Within group sum of squares/dgree of fredom

$$F = \frac{SSB / (I - 1)}{SSW / (N - I)} = \frac{MSB}{MSW}$$

```
> summary(Fit.aov)
```

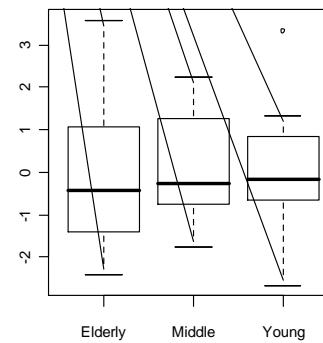
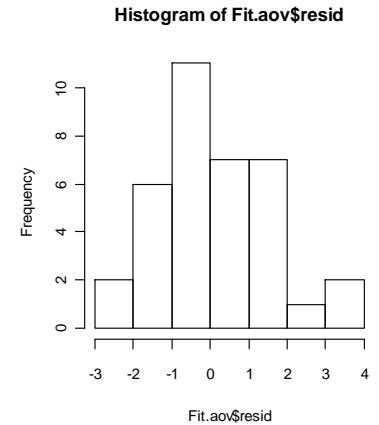
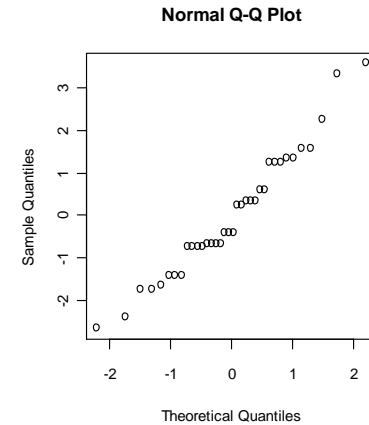
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
group	2	308.722	154.361	67.172	2.322e-12 ***
Residuals	33	75.833	2.298		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Diagnostic plot

```
> par(mfrow=c(2,2))  
> qqnorm(Fit.aov$resid)  
> hist(Fit.aov$resid,col=0)  
> boxplot(split(Fit.aov$resid,group))
```



## Practical session 10 (a)

- Create the following data frame in R

	y	treatment
1	10	A
2	12	A
3	13	A
4	15	A
5	10	B
6	9	B
7	9	B
8	11	B
9	10	C
10	15	C
11	13	C
12	8	C

- Use one-way ANOVA model to test the null hypothesis of no treatment effect

## Practical session 10 (b)

- Create the following data frame in R

	y	treatment
1	10	A
2	12	A
3	13	A
4	15	A
5	10	B
6	9	B
7	9	B
8	11	B
9	10	C
10	15	C
11	13	C
12	8	C

- Use one-way ANOVA model to test the null hypothesis of no treatment effect

# Chapter 7

## Statistical modeling 3: Logistic regression

Examples:

Serological data

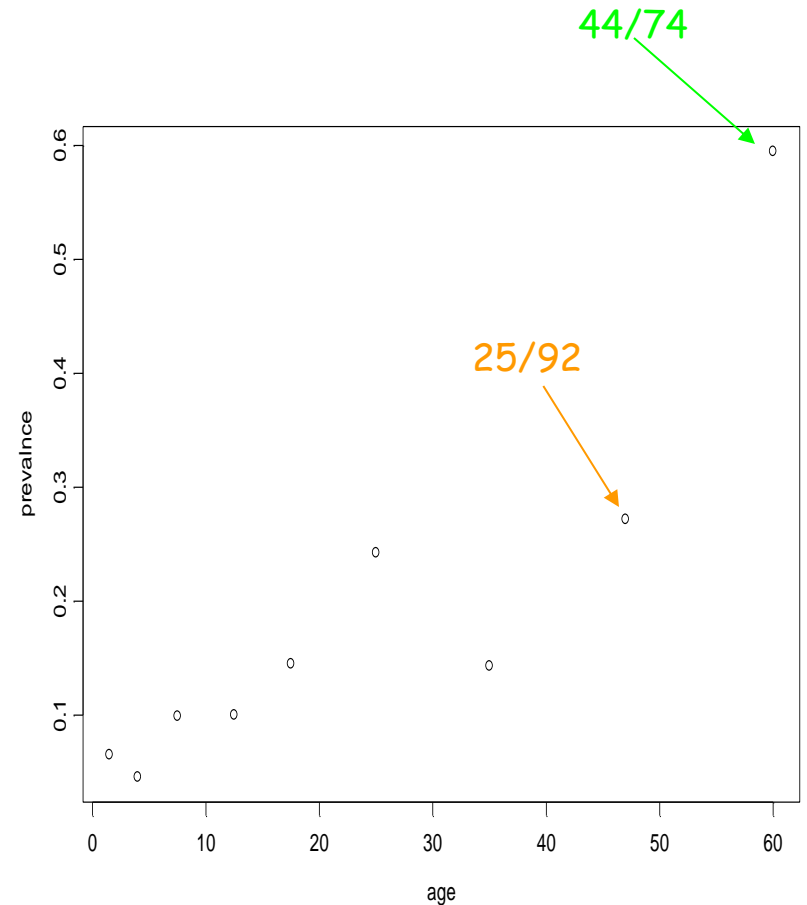
# Example : Serological data of malaria

- In this example the information about each subject in the experiment is the disease status (infected or not by malaria) and the age group of the subject.
- The variables are: the sample size, the number of sero-positive at each sample size (=the number of infected subjects) and the age.



# Example : serological data

Age group	Mid age	Sero positive	Sample size
	1.5	8	123
	4.0	6	132
	7.5	18	182
	12.5	14	140
	17.5	20	138
	25.0	39	161
	35.0	19	133
	47.0	25	92
	60.0	44	74



# Reading the data

```
> sero<-read.table('c:\\projects\\wseda\\Rintro\\sero1.txt',  
header=FALSE,na.strings="NA", dec=".")
```

```
> print(sero)
```

	V1	V2	V3	V4
1	1	1.5	123	8
2	2	4.0	132	6
3	3	7.5	182	18
4	4	12.5	140	14
5	5	17.5	138	20
6	6	25.0	161	39
7	7	35.0	133	19
8	8	47.0	92	25
9	9	60.0	74	44

# Example : serological data

Mid age	Sero positive	Sample size
1.5	8	123
4.0	6	132
7.5	18	182
12.5	14	140
17.5	20	138
25.0	39	161
35.0	19	133
47.0	25	92
60.0	44	74

$$Z_i = \begin{cases} 1 & \text{sero pos.} \\ 0 & \text{sero neg.} \end{cases}$$

$$Y_i = \sum Z_i$$

Number of sero-positive at each age group

$$Y_i \sim B(n_i, P_i)$$

$n_i$ : sample size at each age group

$P_i$  is the probability to be infected (the prevalence). We use logistic regression in order to model the prevalence as a function of age

$$\log it(P_i) = \alpha + \beta \times \text{age}$$

# The probability of infection

If  $\beta > 0$  then there is a positive association between the probability and age. This means that the probability of infection increase with age.

$$P = \frac{e^{\alpha + \beta \text{ age}}}{1 + e^{\alpha + \beta \text{ age}}}$$

If  $\beta < 0$  then there is a negative association between the probability and age. This means that the probability of infection decrease with age.

# The glm() function

$$Y_i \sim B(n_i, P_i)$$

$$\log \text{it}(P_i) = \alpha + \beta \times \text{age}$$

glm(pos/ntot ~ age, family=binomial(link = "logit"))

# The glm() function

```
> fit.glm<- glm(pos/ntot ~ age, family=binomial(link = "logit"))
> summary(fit.glm)
```

Call:

```
glm(formula = pos/ntot ~ age, family = binomial(link = "logit"))
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.24364	-0.09726	0.01479	0.06756	0.19568

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-2.79677	1.79832	-1.555	0.120
age	0.04718	0.04668	1.011	0.312

(Dispersion parameter for binomial family taken to be 1)

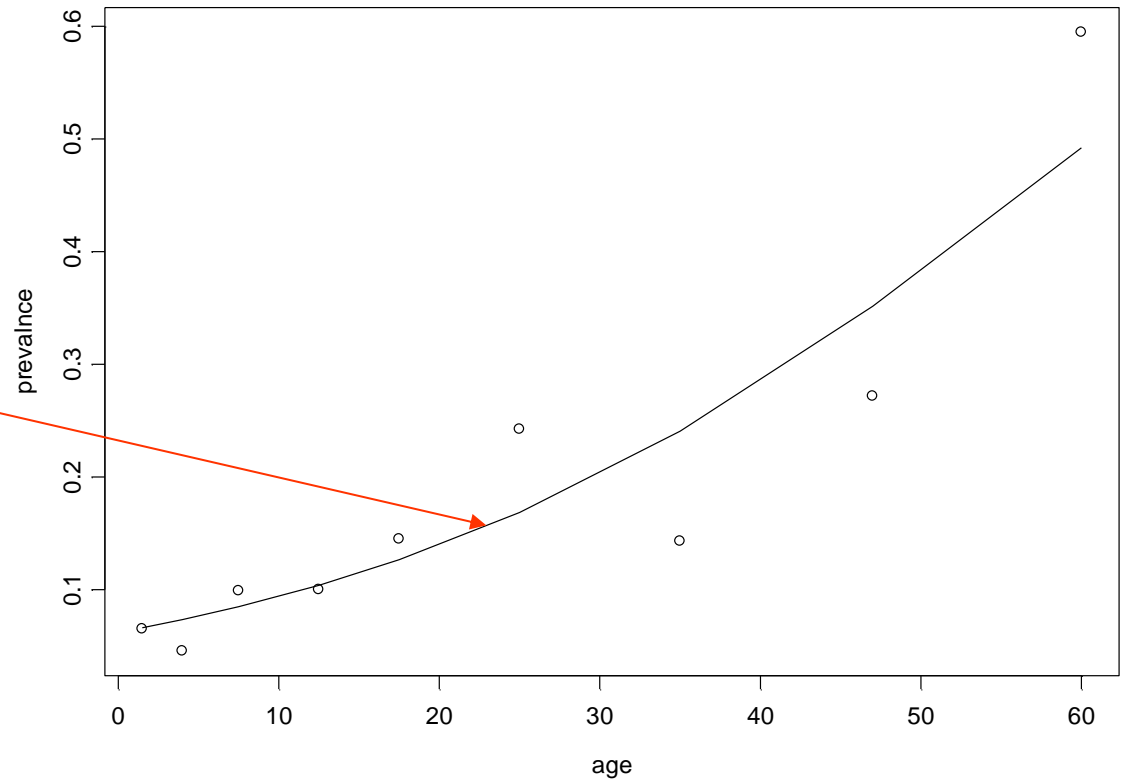
Null deviance: 1.31775 on 8 degrees of freedom  
Residual deviance: 0.18094 on 7 degrees of freedom  
AIC: 8.062

Number of Fisher Scoring iterations: 5

# Data and predicted values

$$\log it(\hat{P}_i) = -2.71 + 0.044 \times age$$

$$\hat{P}_i = \frac{e^{-2.71+0.044 \times age}}{1 + e^{-2.71+0.044 \times age}}$$



# **Chapter 8**

## **Programming in R II: User functions**



Generate a random sample of size 1000  
from  $N(0,3)$

```
> x<-rnorm(1000,0,3)
```

```
> mean(x)
```

```
[1] 0.3080260
```

```
> median(x)
```

```
[1] 0.4176008
```

```
> quantile(x)
```

0%	25%	50%	75%	100%
-5.9877043	-1.7844439	0.4176008	1.5712923	8.5930491

## A user function: general form

```
function name<-function(x)  
{
```

*R commands (what do you want that the function will do for you.....)*

```
}
```

## A user function: example

```
fch20<-function(x)
{
mean.x<-mean(x)
med.x<-median(x)
q.x<-quantile(x)
hist(x)
return(mean.x,med.x,q.x)
}
```

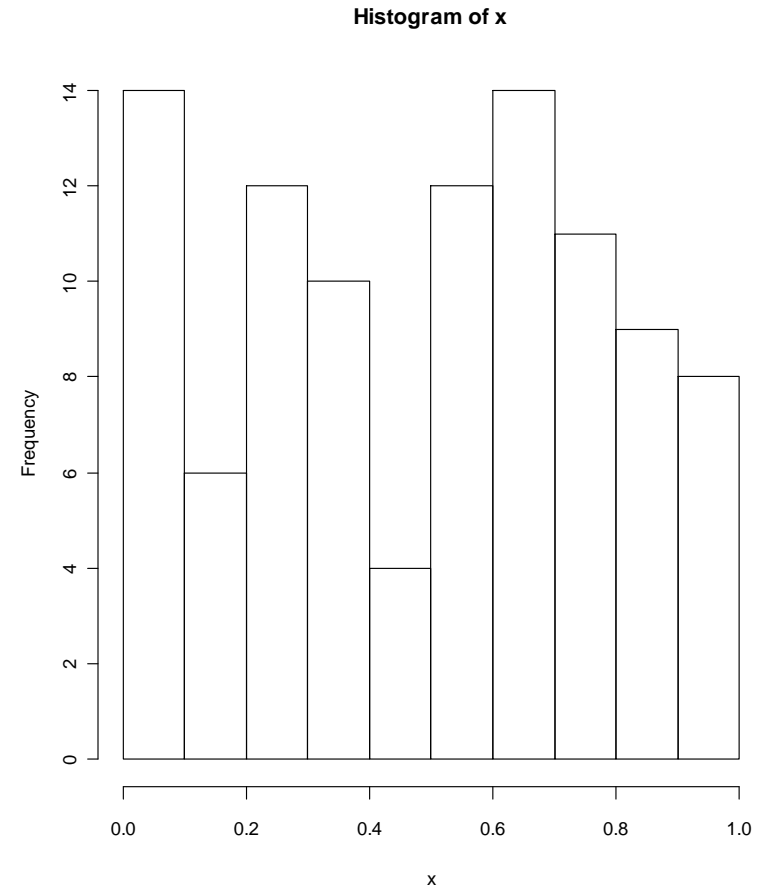
# A user function: output

```
> z<-runif(100,0,1)
> fch20(z)
$mean.x
[1] 0.4947539

$med.x
[1] 0.5291341

$q.x
      0%      25%
50%      75%     100%
0.01240262 0.24212404
0.52913405 0.72482479
0.98413912

Warning message:
In return(mean.x, med.x, q.x) :
  multi-argument returns are
  deprecated
>
```



# Practical session 11

- Write a function which receive a numerical vector as an input and calculate the mean of the vector.

## **Chapter 9:**

# **Statistical modeling : Two-way ANOVA**

# Model formulation

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \varepsilon_{ijk}$$

$\mu$  Overall mean

$\alpha_i$  Main effect of factor A

$\beta_j$  Main effect of factor B

$\alpha\beta_{ij}$  Interaction effect

$\varepsilon_{ijk}$  Random error

# Example 1: Reading the data

```
> spwh3<-read.table('c:\\projects\\wseda\\spwh3.txt',  
  header=FALSE,na.strings="NA", dec=".")  
> names(spwh3)<-c("id","y","x1","gender")
```



## Example 1: The data

```
> print(spwh3)
      id      y  x1 gender
1     1 10.111368   1     0
2     2  9.948930   1     0
3     3 10.322560   1     0
.     .      .     .     .
.     .      .     .     .
59    59 30.030490   3     1
60    60 29.541542   3     1
>
```

Both x1 and gender are numerical objects !!!!

For an ANOVA model the independent variables are suppose to be factors.

## Example 2: The data

	y	f1	f2
1	10	A1	B1
2	11	A1	B1
3	12	A1	B1
4	9	A2	B1
5	7	A2	B1
6	6	A2	B1
7	11	A1	B2
8	13	A1	B2
9	14	A1	B2
10	7	A2	B2
11	5	A2	B2
12	8	A2	B2

Two factors: f1 and f2

Three observations per combination.

```
> f1<-c("A1","A1","A1","A2","A2","A2","A1","A1","A1","A2","A2","A2")
> f2<-c("B1","B1","B1","B1","B1","B1","B2","B2","B2","B2","B2","B2")
> y<-c(10,11,12,9,7,6,11,13,14,7,5,8)
> data.frame(y,f1,f2)
```

# Which null hypotheses we test ?

$$H_0 : \alpha_1 = \alpha_2 \quad \text{No treatment effect of factor A}$$

$$H_0 : \beta_1 = \beta_2 \quad \text{No treatment effect of factor B}$$

$$\text{No interaction effects} \quad H_0 : \alpha\beta_{11} = \alpha\beta_{12} = \alpha\beta_{21} = \alpha\beta_{22}$$

## Example 1: A model without interaction

```
> fit.1<-aov(y~as.factor(x1)+as.factor(gender))
```

```
> anova(fit.1)
```

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
as.factor(x1)	2	1034.81	517.40	2244.8	< 2.2e-16 ***
as.factor(gender)	1	1509.98	1509.98	6551.3	< 2.2e-16 ***
Residuals	56	12.91	0.23		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Example 1: A model with interaction

```
fit.2<-aov(y~as.factor(x1)+as.factor(gender)
           +as.factor(x1)*as.factor(gender))
```

```
> anova(fit.2)
Analysis of Variance Table
```

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
as.factor(x1)	2	1034.81	517.40	2171.959	<2e-16	***
as.factor(gender)	1	1509.98	1509.98	6338.599	<2e-16	***
as.factor(x1):as.factor(gender)	2	0.04	0.02	0.091	0.9131	
Residuals	54	12.86	0.24			

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```
>
```

## Example 1: Testing model 1 versus model 2

```
> anova(fit.1,fit.2)
```

Analysis of Variance Table

Model 1:  $y \sim \text{as.factor}(x1) + \text{as.factor}(\text{gender})$

Model 2:  $y \sim \text{as.factor}(x1) + \text{as.factor}(\text{gender}) + \text{as.factor}(x1) * \text{as.factor}(\text{gender})$

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	56	12.9073				
2	54	12.8639	2	0.0434	0.091	0.9131



F-test for the interaction

## Example 2: A model without interaction

```
> fit.1<-aov(y~f1+f2)
> anova(fit.1)
```

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
f1	1	70.083	70.083	31.4066	0.0003325 ***
f2	1	0.750	0.750	0.3361	0.5763122
Residuals	9	20.083	2.231		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Example 2: A model with interaction

```
> fit.2<-aov(y~f1+f2+f1*f2)
> anova(fit.2)
```

### Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
f1	1	70.083	70.083	35.0417	0.0003539	***
f2	1	0.750	0.750	0.3750	0.5572922	
f1:f2	1	4.083	4.083	2.0417	0.1909016	
Residuals	8	16.000	2.000			

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1



## Example 2: Testing model 1 versus model 2

```
> anova(fit.1,fit.2)
```

Analysis of Variance Table

Model 1:  $y \sim f1 + f2$

Model 2:  $y \sim f1 + f2 + f1 * f2$

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	9	20.083				
2	8	16.000	1	4.0833	2.0417	0.1909



F-test for the interaction

## Example 2: means by factor level

```
> tapply(y,f1,mean)
```

A1	A2
11.83333	12.00000

Factor 1

```
> tapply(y,f2,mean)
```

B1	B2
9.166667	14.666667

Factor 2

```
> ind<-list(f1,f2)
```

```
> ind
```

```
[[1]]
```

```
[1] "A1" "A1" "A1" "A2" "A2" "A2" "A1" "A1" "A1" "A2" "A2" "A2"
```

```
[[2]]
```

```
[1] "B1" "B1" "B1" "B1" "B1" "B1" "B1" "B2" "B2" "B2" "B2" "B2"
```

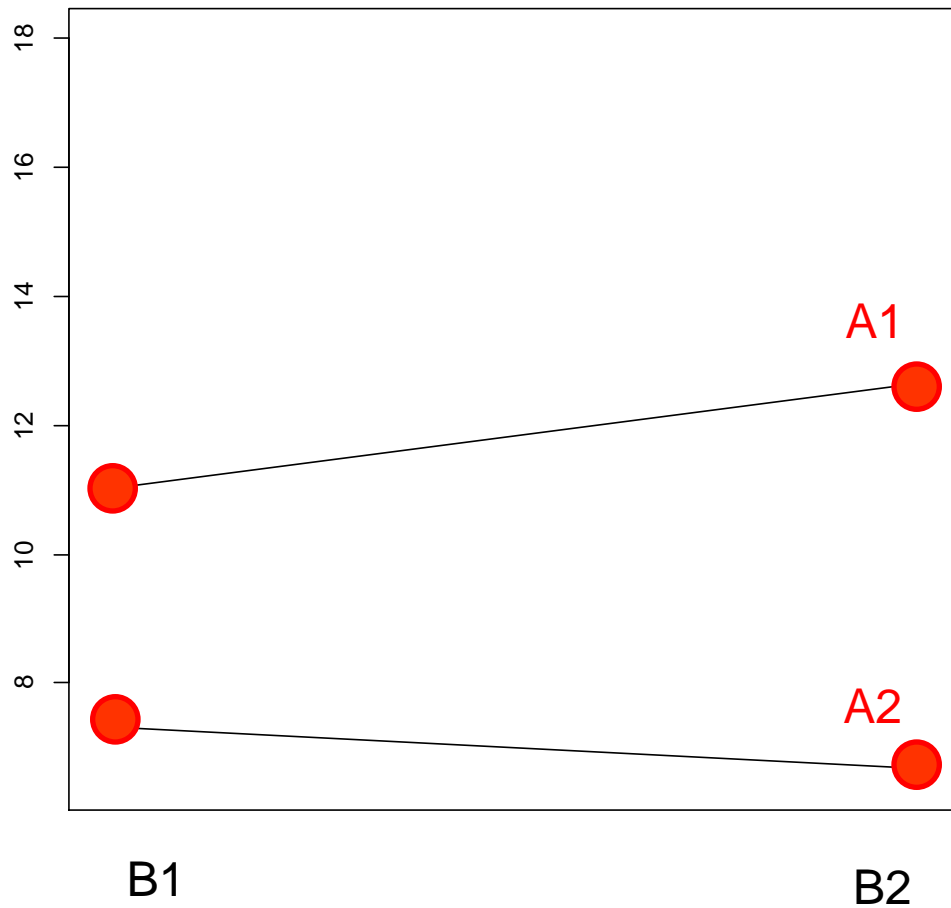
```
> m<-tapply(y,ind,mean)
```

```
> m
```

	B1	B2
A1	11.000000	12.666667
A2	7.333333	16.666667

Cell means

# Interaction plot: Example 2



Cell means

	B1	B2
A1	11.000000	12.666667
A2	7.333333	6.666667

## Example 3: The data

y	f1	f2
1	10	A1 B1
2	11	A1 B1
3	12	A1 B1
4	9	A2 B1
5	7	A2 B1
6	6	A2 B1
7	11	A1 B2
8	13	A1 B2
9	14	A1 B2
10	17	A2 B2
11	15	A2 B2
12	18	A2 B2

Two factors: f1 and f2

Three observations per combination.

```
> f1<-c("A1","A1","A1","A2","A2","A2","A1","A1","A1","A2","A2","A2")
> f2<-c("B1","B1","B1","B1","B1","B1","B2","B2","B2","B2","B2","B2")
> y<-c(10,11,12,9,7,6,11,13,14,17,15,18)
> data.frame(y,f1,f2)
```

## Example 3: A model with interaction

```
> fit.2<-aov(y~f1+f2+f1*f2)
> anova(fit.2)
```

### Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
f1	1	0.083	0.083	0.0417	0.8433536
f2	1	90.750	90.750	45.3750	0.0001471 ***
f1:f2	1	44.083	44.083	22.0417	0.0015517 **
Residuals	8	16.000	2.000		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Example 3: means by factor level

```
> tapply(y,f1,mean)
```

A1	A2
11.83333	12.00000

Factor 1

```
> tapply(y,f2,mean)
```

B1	B2
9.166667	14.666667

Factor 2

```
> ind<-list(f1,f2)
```

```
> ind
```

```
[[1]]
```

```
[1] "A1" "A1" "A1" "A2" "A2" "A2" "A1" "A1" "A1" "A2" "A2" "A2"
```

```
[[2]]
```

```
[1] "B1" "B1" "B1" "B1" "B1" "B1" "B1" "B2" "B2" "B2" "B2" "B2"
```

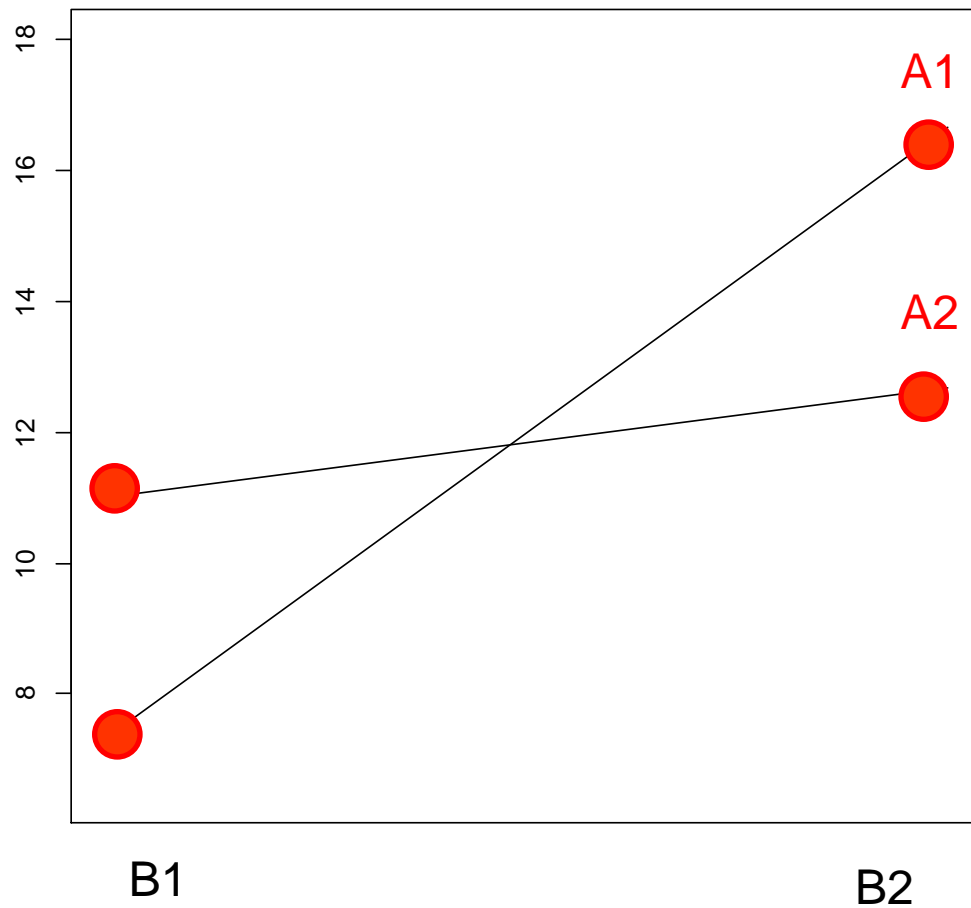
```
> m<-tapply(y,ind,mean)
```

```
> m
```

	B1	B2
A1	11.000000	12.666667
A2	7.333333	16.666667

Cell means

# Interaction plot: Example 3



Cell means

	B1	B2
A1	11.000000	12.66667
A2	7.333333	16.66667

# **Chapter 10**

## **Statistical modeling :**

### **More about two-way ANOVA**



# Reading the data

```
> spwh3<-read.table('c:\\projects\\wseda\\spwh3.txt',  
header=FALSE,na.strings="NA", dec=".")  
> names(spwh3)<-c("id","y","x1","gender")  
> attach(spwh3)
```

# Two-way ANOVA model

```
> fit.2<-aov(y~as.factor(x1)+as.factor(gender)+as.factor(x1)*as.factor(gender))  
> anova(fit.2)
```

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
as.factor(x1)	2	1034.81	517.40	2171.959	<2e-16	***
as.factor(gender)	1	1509.98	1509.98	6338.599	<2e-16	***
as.factor(x1):as.factor(gender)	2	0.04	0.02	0.091	0.9131	
Residuals	54	12.86	0.24			

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Stepwise procedure

```
> slm1 <- step(fit.2)
```

```
Start:  AIC=-80.4
```

```
y ~ as.factor(x1) + as.factor(gender) + as.factor(x1) * as.factor(gender)
```

	Df	Sum of Sq	RSS	AIC
- as.factor(x1):as.factor(gender)	2	0.043	12.907	-84.193
<none>			12.864	-80.395

```
Step:  AIC=-84.19
```

```
y ~ as.factor(x1) + as.factor(gender)
```

	Df	Sum of Sq	RSS	AIC
<none>			12.91	-84.19
- as.factor(x1)	2	1034.81	1047.72	175.60
- as.factor(gender)	1	1509.98	1522.89	200.04

# Stepwise procedure

```
> summary(slm1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
as.factor(x1)	2	1034.81	517.40	2244.8	< 2.2e-16	***
as.factor(gender)	1	1509.98	1509.98	6551.3	< 2.2e-16	***
Residuals	56	12.91	0.23			

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# **Chapter 11**

## **Statistical modeling :**

### **More about Linear regression**

# Reading the data

```
> spwh2<-read.table('c:\\projects\\wseda\\spwh2.txt',
header=FALSE,
+                  ,na.strings="NA", dec=".")
> dim(spwh2)
[1] 100    5
>
> names(spwh2)<-c("id","y","x1","x2","x3")
> attach(spwh2)
```

The following object(s) are masked from spwh2 (position 3) :

```
id x1 x2 x3 y
```

# Fitting two models

```
> fit.1<-lm(y~x1+x2)
> anova(fit.1)
```

## Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x1	1	164.2	164.2	27.152	1.059e-06 ***
x2	1	7409.7	7409.7	1224.980	< 2.2e-16 ***
Residuals	97	586.7	6.0		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```
> fit.2<-lm(y~x1+x2+x3)
> anova(fit.2)
```

## Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x1	1	164.2	164.2	758.98	< 2.2e-16 ***
x2	1	7409.7	7409.7	34241.81	< 2.2e-16 ***
x3	1	566.0	566.0	2615.44	< 2.2e-16 ***
Residuals	96	20.8	0.2		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Testing model 1 versus model 2

```
> anova(fit.1,fit.2)
```

```
Analysis of Variance Table
```

```
Model 1: y ~ x1 + x2
```

```
Model 2: y ~ x1 + x2 + x3
```

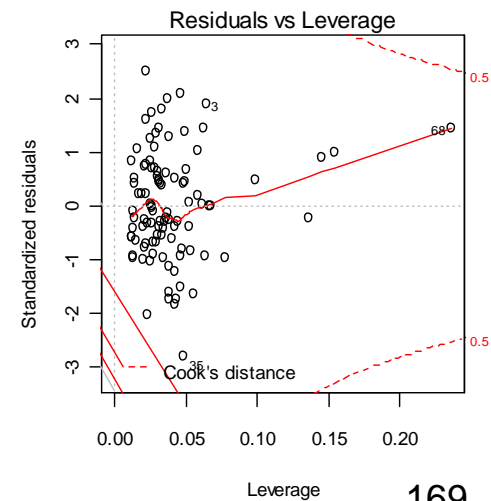
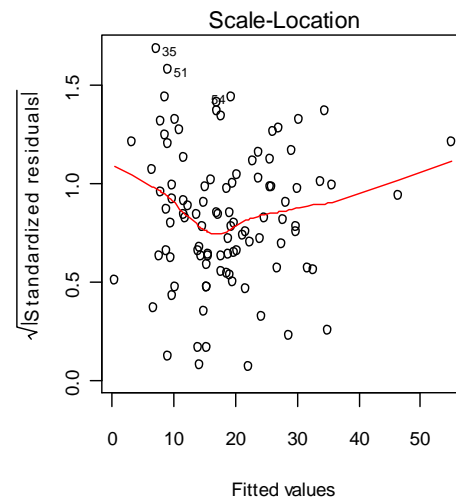
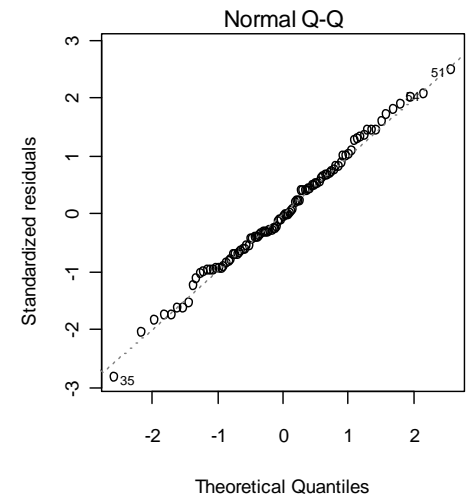
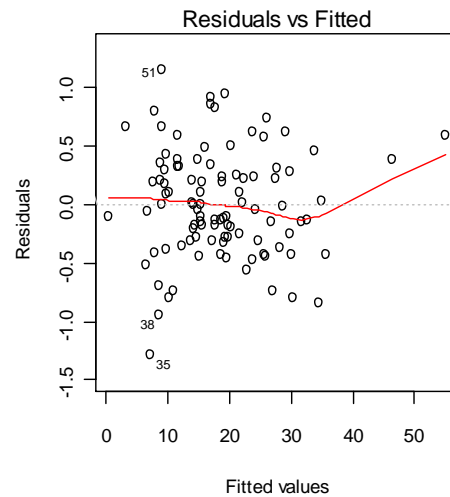
	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	97	586.74				
2	96	20.77	1	565.97	2615.4	< 2.2e-16 ***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'  
0.1 ' ' 1
```



```
> par(mfrow=c(2,2))  
> plot(fit.2)
```



# Single terms deletions

```
> drop1(fit.2, test="F")
Single term deletions
```

Model:

```
y ~ x1 + x2 + x3
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(F)
<none>			20.8	-149.1		
x1	1	76.6	97.4	3.4	354.21	< 2.2e-16 ***
x2	1	7865.3	7886.1	442.8	36347.01	< 2.2e-16 ***
x3	1	566.0	586.7	182.9	2615.44	< 2.2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# AIC and likelihood

```
> AIC(fit.2)
[1] 136.6403
> logLik(fit.2)
'log Lik.' -63.32017 (df=5)
```

# **Chapter 12**

## **Application : the for loop**

**The bootstrap estimate of the standard error for the mean**

# The observed data

A sample of 10 observations:

```
> x <- c(11.201, 10.035, 11.118, 9.055, 9.434, 9.663, 10.403, 11.662, 9.285, 8.84)
> mean(x)
[1] 10.0696
```

We wish to estimate the standard error of the sample mean

$$S.E(\bar{x}) = \frac{\sigma_F}{\sqrt{n}}$$

# Parametric and nonparametric bootstrap

## nonparametric bootstrap

$$F \rightarrow (x_1, x_2, \dots, x_n)$$

We resample from  
the empirical  
distribution

## parametric bootstrap

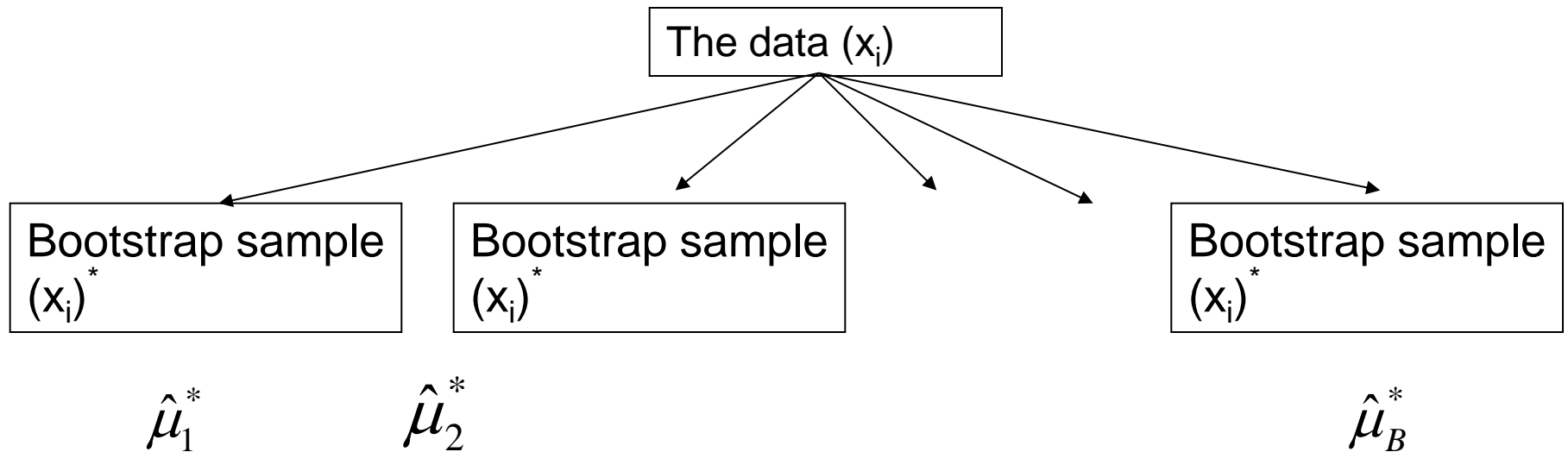
We assume a parametric  
model for  $F$

$$F(\theta)$$

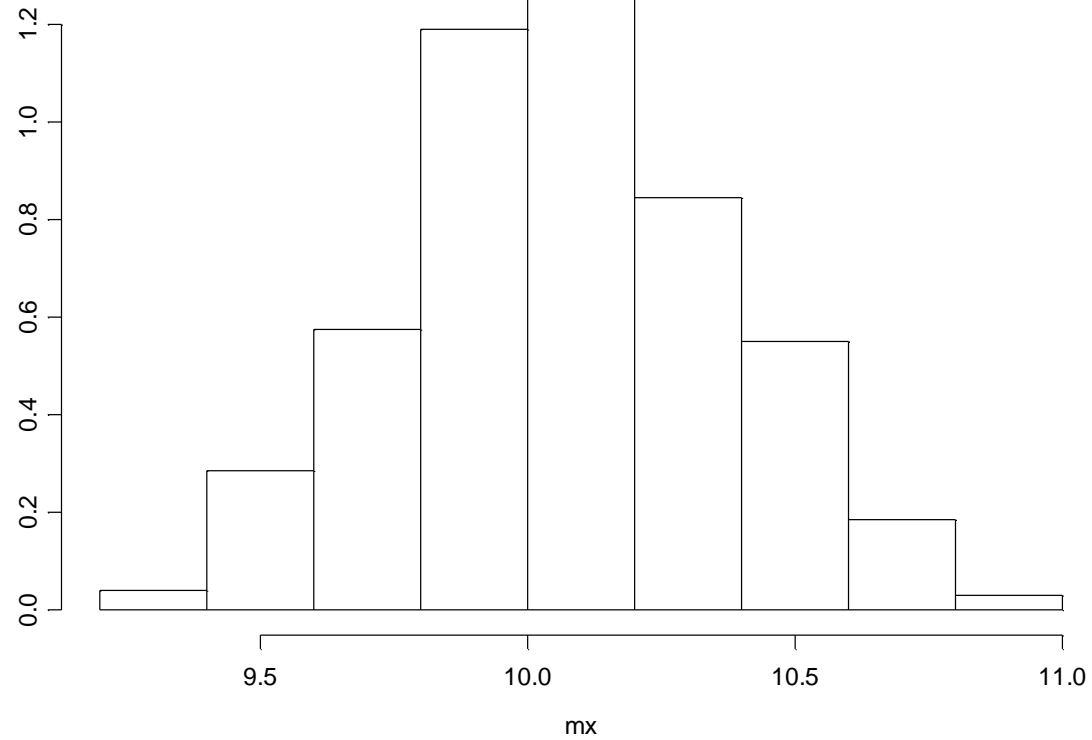
We resample from

$$F(\hat{\theta})$$

# Nonparametric bootstrap

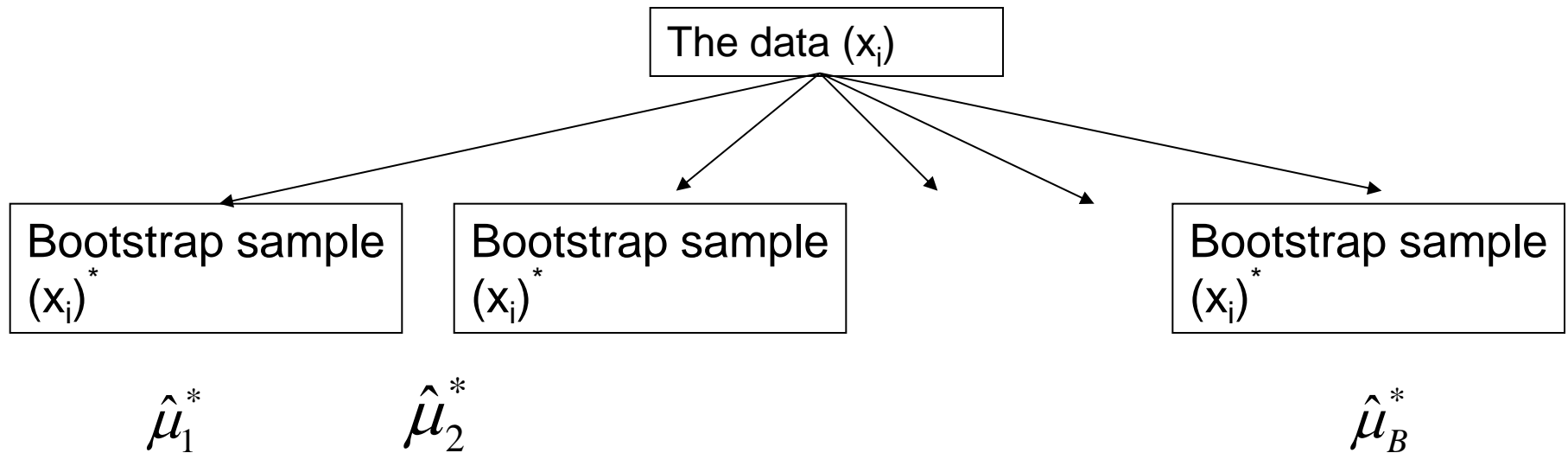


# Nonparametric bootstrap





# Nonparametric bootstrap



$$S.E.(\hat{\mu}) = \left\{ \frac{1}{B-1} \sum_{b=1}^B (\hat{\mu}_b^* - \hat{\mu}^*)^2 \right\}^{0.5}$$

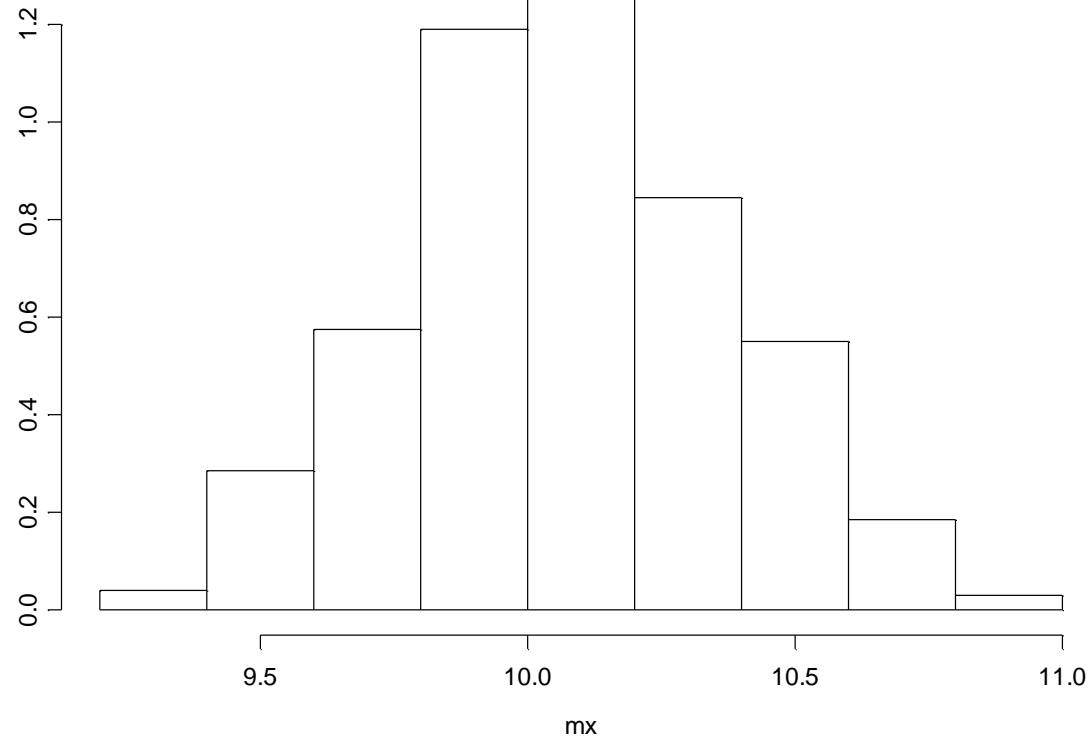
## R code

```
> var(mx)
[1] 0.09357364
```

The estimated  
standard error 0.093

```
n<-length(x)
B<-1000
mx<-c(1:B)
for(i in 1:B){
  cat(i)
  boot.i<-sample(x,n,replace=T)
  mx[i]<-mean(boot.i)
}
```

# Nonparametric bootstrap



# Parametric bootstrap

We assume a parametric model for  $F$

We estimate  $F$  by

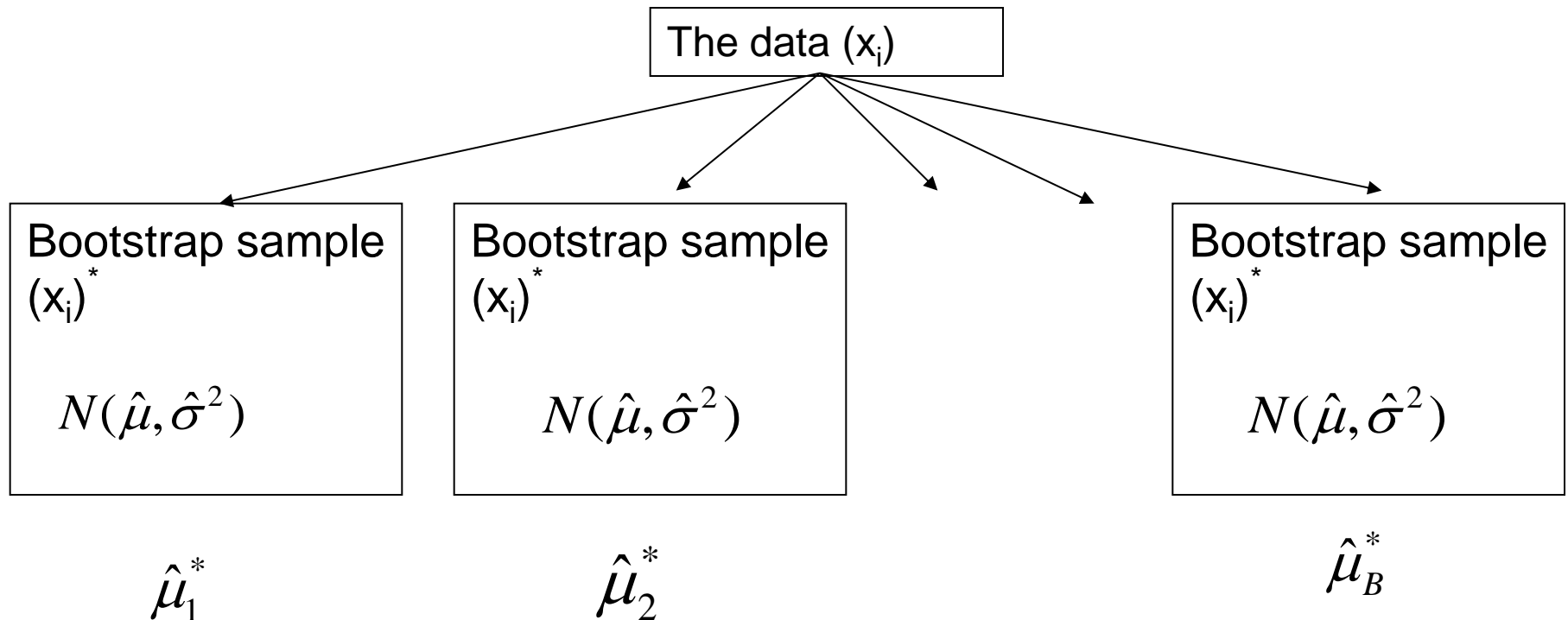
$$F = N(\mu, \sigma^2)$$

$$\hat{F} = N(\hat{\mu}, \hat{\sigma}^2)$$



We replace the unknown parameters in  $F$  with their plug-in estimates

# Parametric bootstrap



$$S.E.(\hat{\mu}) = \left\{ \frac{1}{B+1} \sum_{b=1}^B (\hat{\mu}_b^* - \hat{\mu}^*)^2 \right\}^{0.5}$$

## R code

```
> var(mx)
[1] 0.1007613
```

Bootstrap estimate for the  
standard error for the mean

```
B<-1000
MLx<-mean(x)
Varx<-var(x)
mx<-c(1:B)
for(i in 1:B){
  cat(i)
  boot.i<-rnorm(n,MLx,sqrt(Varx))
  mx[i]<-mean(boot.i)
}
```

# Parametric bootstrap

