UNIVERSITÉ DU LUXEMBOURG

CENTRE DE RECHERCHE PUBLIC SANTÉ

PhD Course

Advanced Biostatistics

Lecture 1

Introduction to R.

Descriptive Statistics in R

dr. P. Nazarov

petr.nazarov@crp-sante.lu

15-12-2014

Outline

- **Information package**
- **Installation**
- **R interface (L1.3)**
  - typing commands, calling functions, embedded help and demo
- **Variables and basic operations (L1.4)**
  - variables, types of data, scalar data, vectors, matrixes, data frames, lists.
- **Data import and export (L1.5)**
  - work folders, use scan, read/write tables, load/save data
- **Control workflow and custom functions (L1.6)**
  - if, while, repeat, next, break, custom functions, use external scripts
- **Data visualization (L1.7)**
  - variables, types of data, scalar data, vectors, matrixes, data frames, lists

L1.1. Information Package

Main Web-page:
cran.r-project.org

cran.r-project.org/manuals.html
cran.r-project.org/web/packages/
cran.r-project.org/other-docs.html

R/Bioconductor
www.bioconductor.org

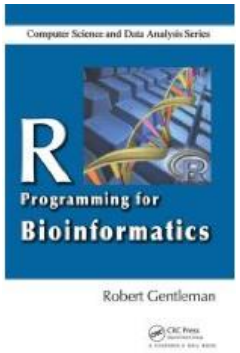
R-Project Seek Engine:
www.rseek.org

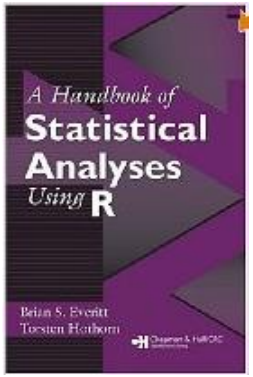
Advanced Biostatistics
edu.sablab.net/abs2014

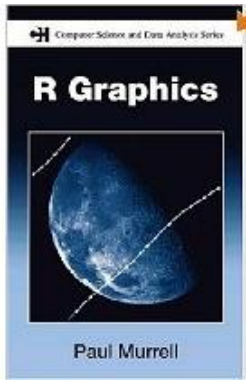
Scripts
edu.sablab.net/abs2014/scripts

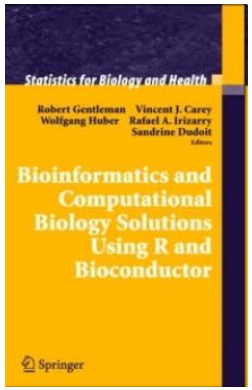
Data
edu.sablab.net/data/txt

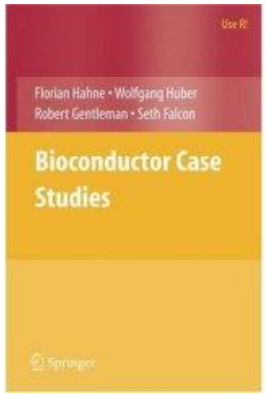
Other related courses
edu.sablab.net











L1.2. Installation

1. Download Binaries

http://cran.r-project.org/bin/

http://cran.r-project.org/bin/windows/base/ (for Windows)

2. Install R (basic packages are automatically installed)

3. Run R and install additional packages (need Internet)

```
install.packages(package_name)
install.packages("rgl")
```

4. Another method: using Bioconductor tools (more robust)

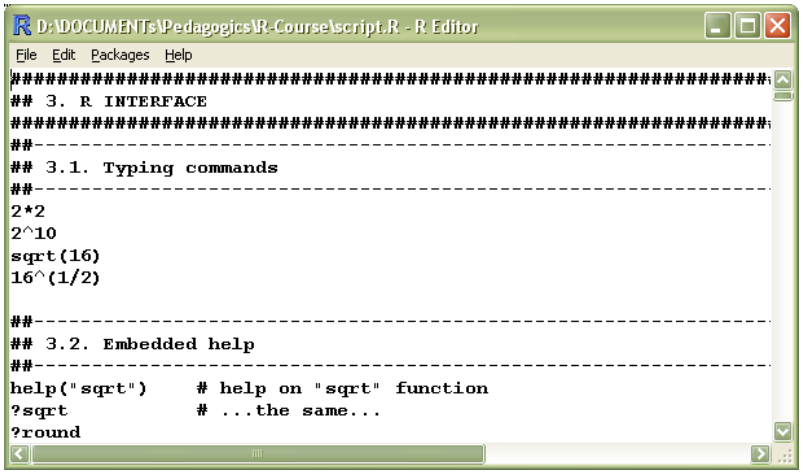
http://www.bioconductor.org/install/

```
source("http://bioconductor.org/biocLite.R")
biocLite(package_name)
```

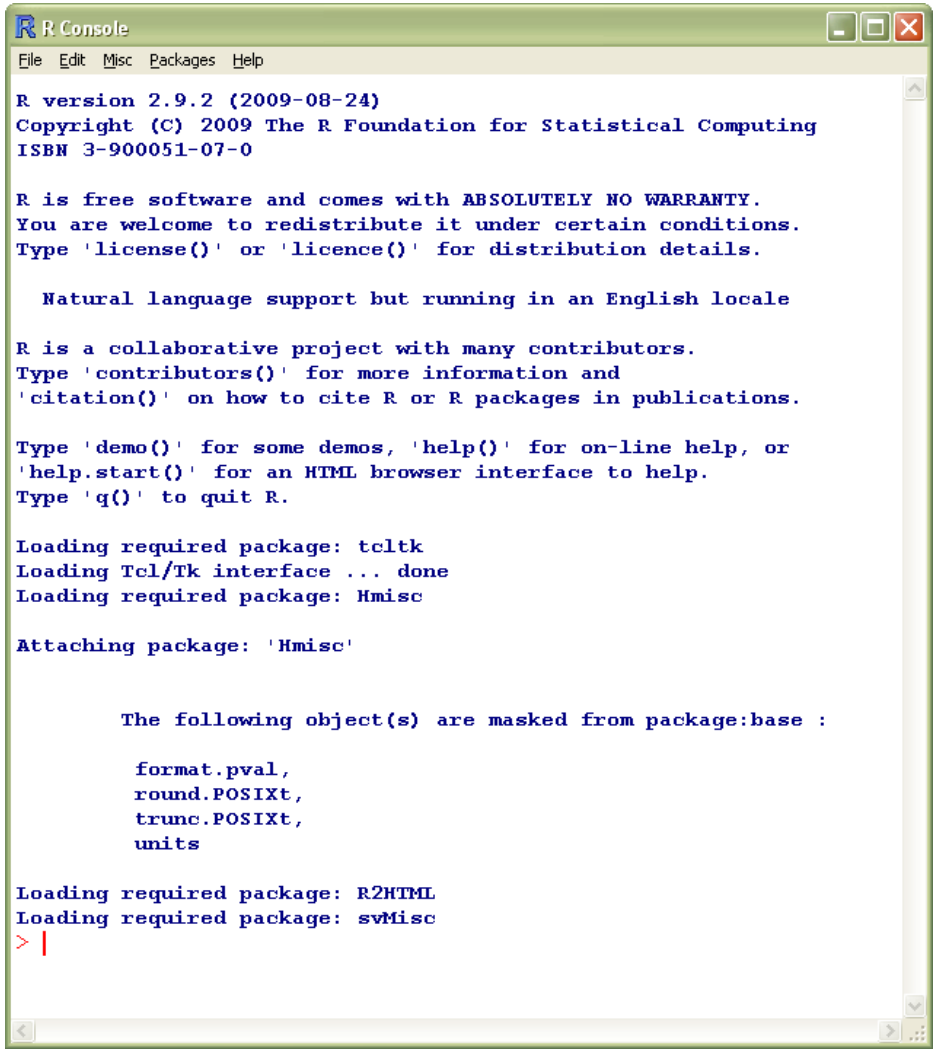
See more packages at http://cran.r-project.org/web/packages/

L1.3. R Interface

Built-in Script Editor



Console



Alternative Editors

RStudio (Win, Linux, McOS)
http://rstudio.com/

Notepad++ & NpptoR (Win)
http://notepad-plus-plus.org/
http://sourceforge.net/projects/npptor

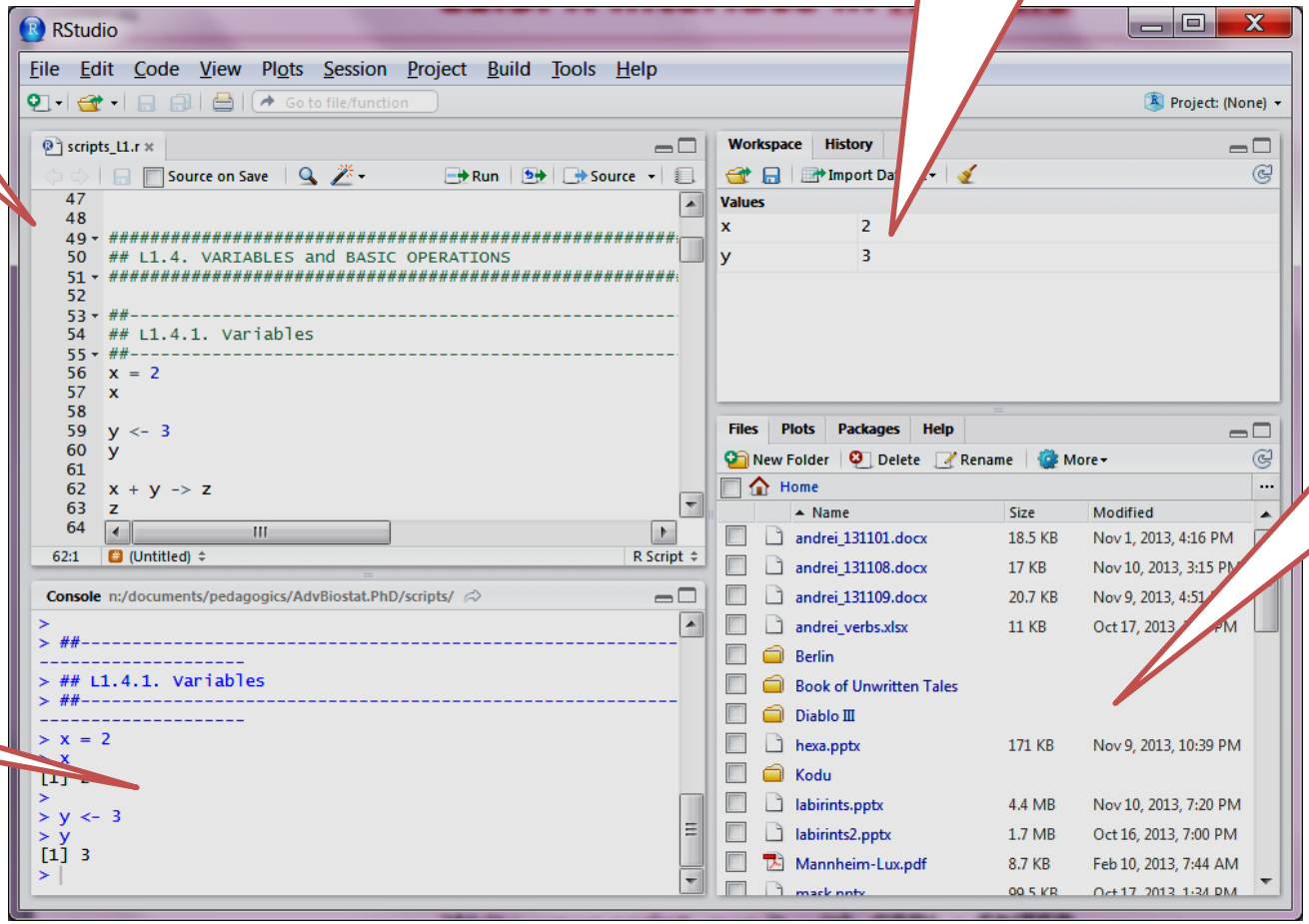
JGR (Win,Linux, McOS)
rforge.net/JGR/

L1.3. R Interface in RStudio

RStudio (Win, Linux, McOS)
http://rstudio.com/

Sripts

Variables and History



Files, Plots, Packages, Help

Console

Write your script, run it with CTRL + ENTER

L1.3.(1-3) Commands, Function and Help

```
################################################################################
## L1.2. INSTALL R PACKAGES
################################################################################
install.packages("rgl")
## if does not work:
##   a) Select all repositories in "packages" menu
##   b) if still does not work - use Bioconductor installation routine
################################################################################
## L1.3. R INTERFACE
################################################################################
##------------------------------------------------------------------------------
## 0.3.1. Typing commands
##------------------------------------------------------------------------------
2*2
2^10
sqrt(16)
16^(1/2)
##------------------------------------------------------------------------------
## L1.3.2. Calling functions
##------------------------------------------------------------------------------
log(100)
log(100, base=10)
log(100, b=10)
log(100, 10)
##------------------------------------------------------------------------------
## L1.3.3. Embedded help
##------------------------------------------------------------------------------
help("sqrt")     # help on "sqrt" function
?sqrt            # ...the same...
?round
??round          # fuzzy seach for "round" in all help topics
apropos("plot") # propose commands with the word "plot" inside the name
## Demos
demo()           # show available demos
demo("image")    # start demo "image"
demo(persp)
demo(plotmath)
```

L1.4. Types of Variables in R

Scalar Data

Numeric
- Integer
- Double

```
1
3.141593
```

Logical

```
TRUE
FALSE
```

Character

```
"Hello, world!"
```

Factor

has a sense to use only in vectors or data frames

```
> answer=factor(c("yes","no"))
> answer
[1] yes no
Levels: no yes
```

Data Containers

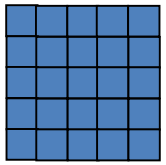
Vector

```
> x
[1] 1 2 3 4 5
```

Matrix

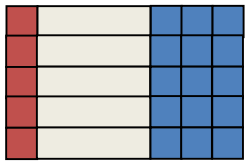
faster then data frame and list other

Array



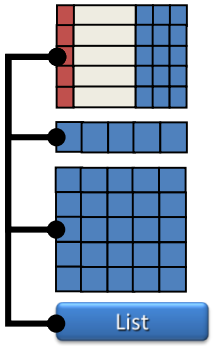
```
> A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    1
```

Data frame



```
  name marks
1 Alex    10
2 Jean     8
3 David    7
```

List



Object of a Class

- Data
- Methods

L1.4. Types of Variables in R

L1.4.(1-3). Variables, Scalar Types, Special Values

```
################################################################################
## L1.4. VARIABLES and BASIC OPERATIONS
################################################################################
##------------------------------------------------------------------------------
## L1.4.1. Variables
##------------------------------------------------------------------------------
x = 2
x
y <- 3
y
x + y -> z
z
## Variables are case-sensitive
Z
## Another way to show the data
print(z)
## show variables in memory
ls()
## remove all variables from memory
rm(list=ls())
ls()

##------------------------------------------------------------------------------
## L1.4.2. Scalar types of data
##------------------------------------------------------------------------------

##--------------------------
## Numeric (integer, double)
i=5
i
i*2
i/2
i%/%2   # integer division
i%%2    # remainder of integer division
round(1.5)
## (*) for bitwise operation install and use
##     "bitops" package and bitAnd, bitOr,...
## Double
r=1.5
r
l=pi*2*r # let us calculate the circumference for circle with r
l

#>>>>>>>>>>>>>>>>>>>>>>>>>
#> please, do Task L1.4a
#>>>>>>>>>>>>>>>>>>>>>>>>>

##--------------------------
## Boolean
b1=TRUE   # try  b1=T
b2=FALSE  # try  b2=F
b1 & b2   # logical AND
b1 | b2   # logical OR
!b1       # logical AND
xor(b1,b2)# logical XOR
r==l
r<l
```

```
##--------------------------
## Character (strings)
st = "Hello, world!"
st
paste("We say:",st) # concatination
sprintf("We say for the %d-rd time: %s..",3,st) # a more powerfull method a-la C
sprintf("By the way pi=%f, and e=%f",pi,exp(1))

sub(", world","",st)  # replace a part of the sting
                      # (*) in R the regular expression are used to define the
pattern!
casefold(st, upper=T) # change the case
nchar(st)             # number of characters
strsplit(st,"")[[1]]  # (*) transforms a string into the vector of single
characters

##--------------------------
## Factors
## ... this will be considerd in part 4.4!
## how to check who is who?
class(st)
is.character(st)
is.numeric(st)
is.numeric(pi)

##------------------------------------------------------------------------------
## L1.4.3. Special values
##------------------------------------------------------------------------------
## NA - Not-Available (missing data)
na = NA
na + 1
100>na
na==na
is.na(na)

## Inf - Infinity (+/- infinite data)
0*1/0
-1/0
is.infinite(1/0)

## NaN - Not-A-Number
0/0
is.nan(sqrt(-1))
```

Task L1.4a

L1.4. Types of Variables in R

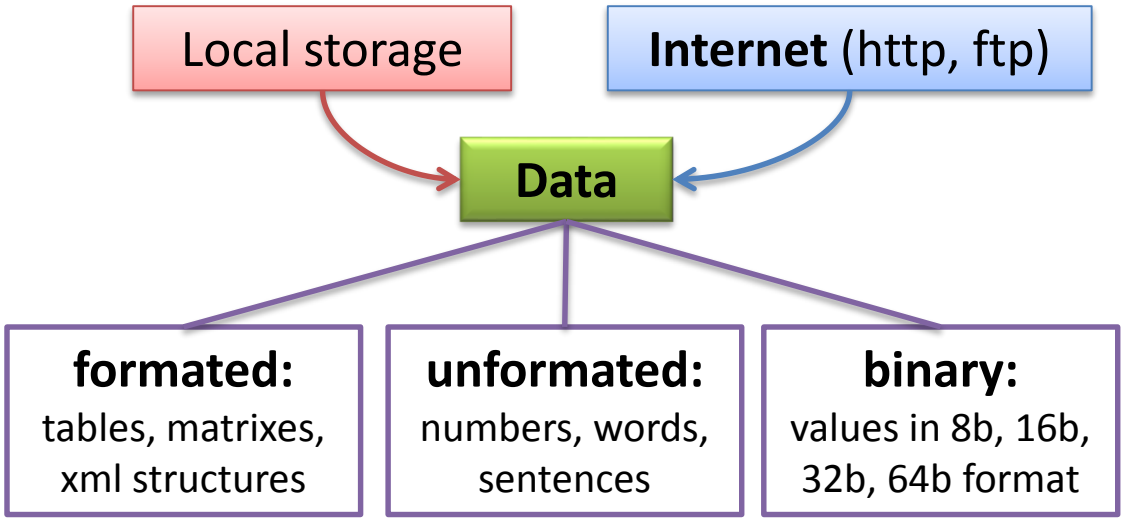
L1.4.(4-6). Vectors, Matrixes, Data Frames, Lists

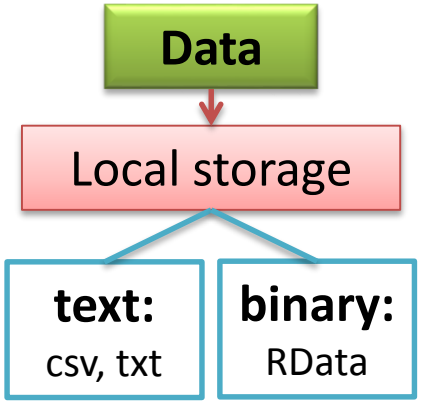
```
##-----------------------------------------------------------------------------
## L1.4.4. Vectors
##-----------------------------------------------------------------------------
## Vector creation
a = c(1,2,3,4,5)
a
a[1]+a[4]
b=5:9
a+b   #(*) try b=5:10. Can you explain the effect? (ans: "!tfihs ralucriC" :)
seq(from=1,to=10,by=0.5) #sequence
seq(1,10,0.5)
rep(1:4, 2)       # same as rep(1:4, times=2)
rep(1:4, each=2)  # not the same
txt = c(st, "Let's try vectors", "bla-bla-bla")
txt
boo = c(T,F,T,F,T)
boo

##!!!!!!!!!!!!!!!!!!!!!!!!
## Extremely important !!
##!!!!!!!!!!!!!!!!!!!!!!!!
## Vector indexes
a
a[1:3] # take a part of vector by index numbers
a[boo] # take a part of vector by logical vector
a[a>2] # take a part by a condition
a[-1]  # removes the first element

#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
#>  Please, do tasks  L1.4b,c,d
#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

##-----------------------------------------------------------------------------
## L1.4.5. Matrixes and Data Frames
##-----------------------------------------------------------------------------
A=matrix(0,nrow=5, ncol=5)
A
A=A-1   # add scalar
A
A=A+a   # add vector
A
t(A)    # transpose
B=A+t(A)# add matrix
B
B*B     # by-element product
B%*%B   # matrix product

##-----------------------------------
## Data frame
Data = data.frame(A)  # alternatively:  D=data.frame(matrix(nr=5,nc=5))
Data
## let us add a column to Data
mice = sprintf("Mouse_%d",1:5)
Data = cbind(mice,Data)
## put the names to the variables
names(Data) = c("name","sex","weight","age","survival","code")
Data

## put in the data manualy
Data$name=sprintf("Mouse_%d",1:5)
Data$sex=c("Male","Female","Female","Male","Male")
Data$weight=c(21,17,20,22,19)
Data$age=c(160,131,149,187,141)
Data$survival=c(T,F,T,F,T)
Data$code = 1:nrow(Data)
Data

## visualize data as a table
fix(Data)

## see the structure of the objects
str(Data)

## see the head of the objects
head(Data)

## summary on the data
summary(Data)

##------------------------
## Factors

## Let's use factors
Data$sex = factor(Data$sex)
summary(Data)

## usefull commands when working with factors:
levels(Data$sex)        # returns levels of the factor
nlevels(Data$sex)       # returns number of levels
as.character(Data$sex)  # transform into strings

##-----------------------------------------------------------------------------
## L1.4.6. Lists
##-----------------------------------------------------------------------------

L=list()
L$Data=Data
L$descr = "A fake experiment with virtual mice"
L$num = nrow(Data)
str(L)

## how to access the fields? Simple!
L$Data
L$"Data"
L$num
## or
L[[1]]
L[[3]]


## clear all
ls()
rm(list=ls())
ls()
```

Tasks L1.4b,c,d

L1.5. Data Import and Export

Way of Data Import and Export





currency.txt

R Data Editor

File Edit Help

	Date	EUR
1	1999-01-04	1.1789
2	1999-01-05	1.179
3	1999-01-06	1.1743
4	1999-01-07	1.1632
5	1999-01-08	1.1659
6	1999-01-11	1.1569
7	1999-01-12	1.152
8	1999-01-13	1.1744
9	1999-01-14	1.1653
10	1999-01-15	1.1626
11	1999-01-18	1.1612
12	1999-01-19	1.1616
13	1999-01-20	1.1575
14	1999-01-21	1.1572
15	1999-01-22	1.1567
16	1999-01-25	1.1584
17	1999-01-26	1.1582
18	1999-01-27	1.1529
19	1999-01-28	1.141
20	1999-01-29	1.1384

L1.5. Data Import and Export

Script

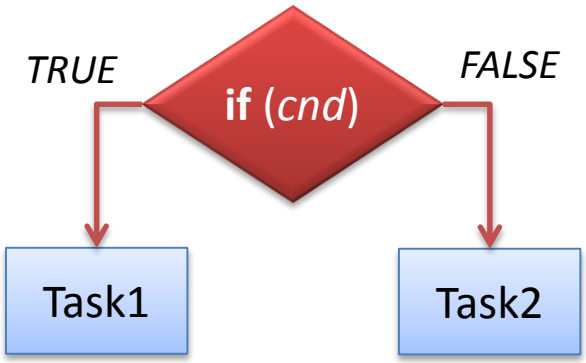
```
################################################################################
# L1.5. DATA IMPORT AND EXPORT
################################################################################
##------------------------------------------------------------------------------
## L1.5.1. Current folder
##------------------------------------------------------------------------------
getwd()   ## shows current folder
dir()     ## shows files in the current folder
setwd("E:/DOCUMENTs/Pedagogics/R-Course_2010/Data") ## sets folder

##------------------------------------------------------------------------------
## L1.5.2. Scanf - reads arbitrary data
##------------------------------------------------------------------------------
## File from Internet / disk
SomeData = scan("http://edu.sablab.net/data/txt/currency.txt",
                what = character(0))
SomeData

## HTML from Internet
Google = scan("http://google.com",what = character(0))
Google

##------------------------------------------------------------------------------
## L1.5.3. Read table (from Internet or local folder)
##------------------------------------------------------------------------------
Currency = read.table("http://edu.sablab.net/data/txt/currency.txt",
                      header=T, sep="\t")
str(Currency)

## let's ask to do not transfere strings to factors
Currency = read.table("http://edu.sablab.net/data/txt/currency.txt",
                      header=T, sep="\t", as.is=T)
str(Currency)
head(Currency)
summary(Currency)
fix(Currency)
## first plot :)
plot(Currency$EUR)

##------------------------------------------------------------------------------
## L1.5.4. "GE" a big dataset: use "download.file" and "load"
##------------------------------------------------------------------------------
download.file("http://edu.sablab.net/data/all.Rdata",
              destfile="all.Rdata",mode = "wb")
## check the current folder for ".Rdata" file
getwd()                 # show current folder
dir(pattern=".Rdata")   # show files in the current folder
load("all.RData")       # load the data
ls()
str(GE.matrix)
## see the annotation for dimentions
attr(GE.matrix,"dimnames")
```

```
##------------------------------------------------------------------------------
## L1.5.5. Data export
##------------------------------------------------------------------------------
write.table(Shop,"shop.txt",sep = "\t",
            eol = "\n", na = "NA", dec = ".",
            row.names = F,
            qmethod = c("escape", "double"));

save(Shop,file="shop.Rdata")

getwd()
dir()
## if you need to set working folder, use
setwd("...put here desired path...")

## clear all
rm(list=ls())

#>>>>>>>>>>>>>>>>>>>>>>>>>
#> please, do Tasks L1.5a, L1.5b
#>>>>>>>>>>>>>>>>>>>>>>>>>
```

Tasks L1.5a,b

L1.6. Control Workflow and Custom Functions

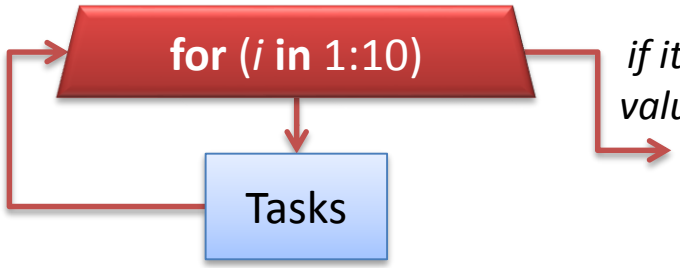
Main Workflow Control Methods



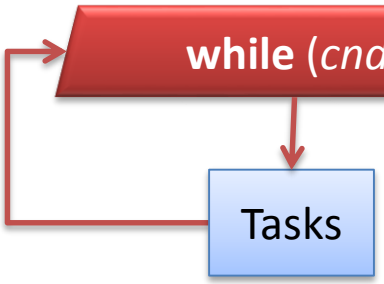

```
if (r > 1) {
    fc = r     ## task 1
} else {
    fc = -r    ## task 2
}
```

Functional version:

```
fc = ifelse(r>1,r,-r)
```




```
for (i in 1:10){
  print(2^i)
}
```




```
i=1
while (i<=10){
  print(2^i)
  i=i+1
}
```

*Command **next** finishes current iteration and starts a new one.*

*Command **break** allows going out from a loop immediately.*

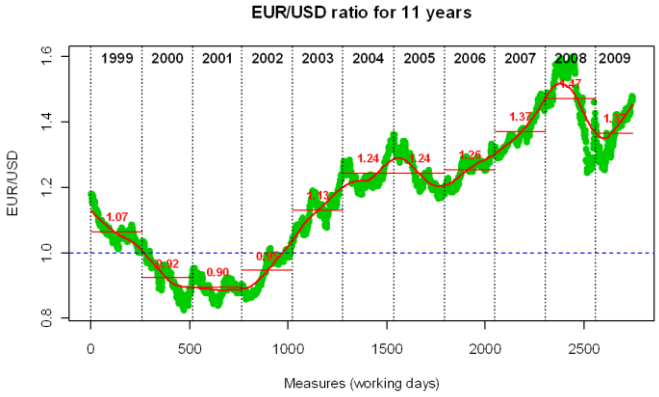
L1.6. Control Workflow and Custom Functions

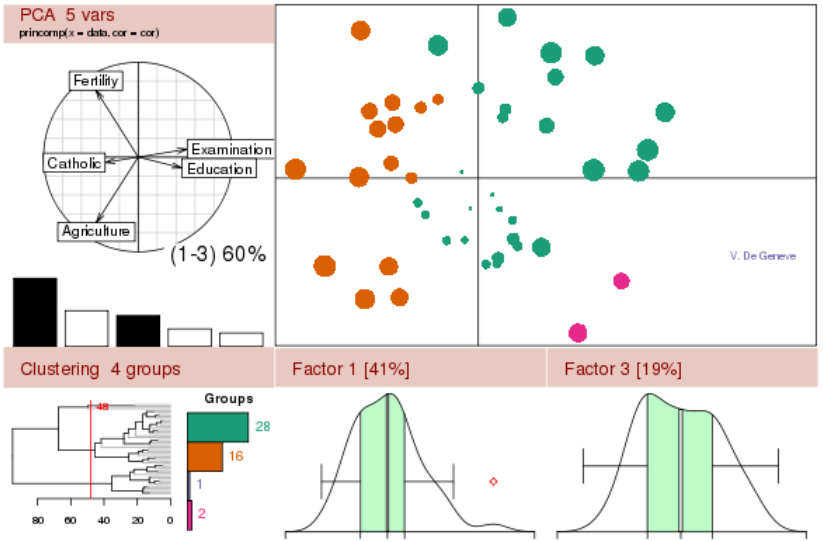
```
################################################################################
## L1.6. CONTROL WORKFLOW and CUSTOM FUNCTIONS
################################################################################
Shop = read.table("http://edu.sablab.net/data/txt/shop.txt",header=T,sep="\t")
a=1
b=2
##------------------------------
## IF condition
if (a==b) {
  print("a equals to b")
} else {
  print("a is not equal to b")
}

## use if in-a-line
ifelse(a>b, a, b)

##------------------------------
## FOR loop

## print all information for the first client
for (i in 1:ncol(Shop))
  print(Shop[1,i])

##------------------------------
## WHILE loop

## print all information for the first client
i=1;
while (i <= ncol(Shop)){
  print(Shop[1,i])
  i=i+1
}

##------------------------------
## REPEAT loop

i=1
repeat {
  print(i)
  i=i+1
  if (i>10) break
}
## "break" and "next" - help to control flow
```

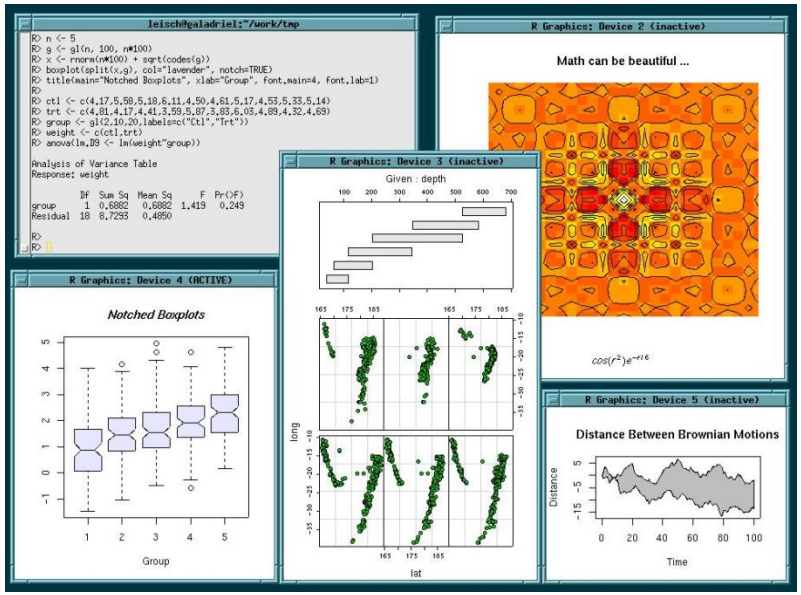
```
##------------------------------
## Custom functions
##------------------------------

## Let us write a function to print vectors
printVector = function(x, name=""){
  print(paste("Vector",name,"with",length(x),"elements:"))
  if (length(x)>0)
    for (i in 1:length(x))
      print(paste(name,"[",i,"] =",as.character(x[i])))
}

printVector(Shop$Payment, "Payment")


##----------------------------------
## Run script, saved in other files
##----------------------------------

source("http://sablab.net/scripts/getFiles.r")

ls()
```

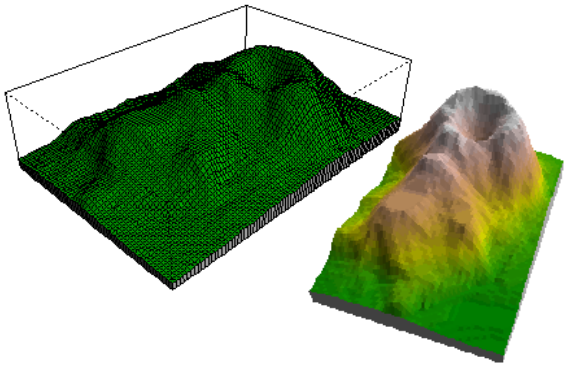
Tasks L1.6a,b

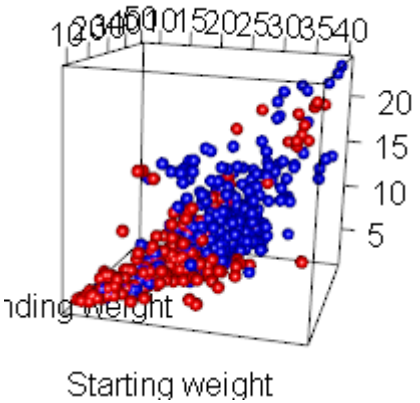
Various Figures Generated in R



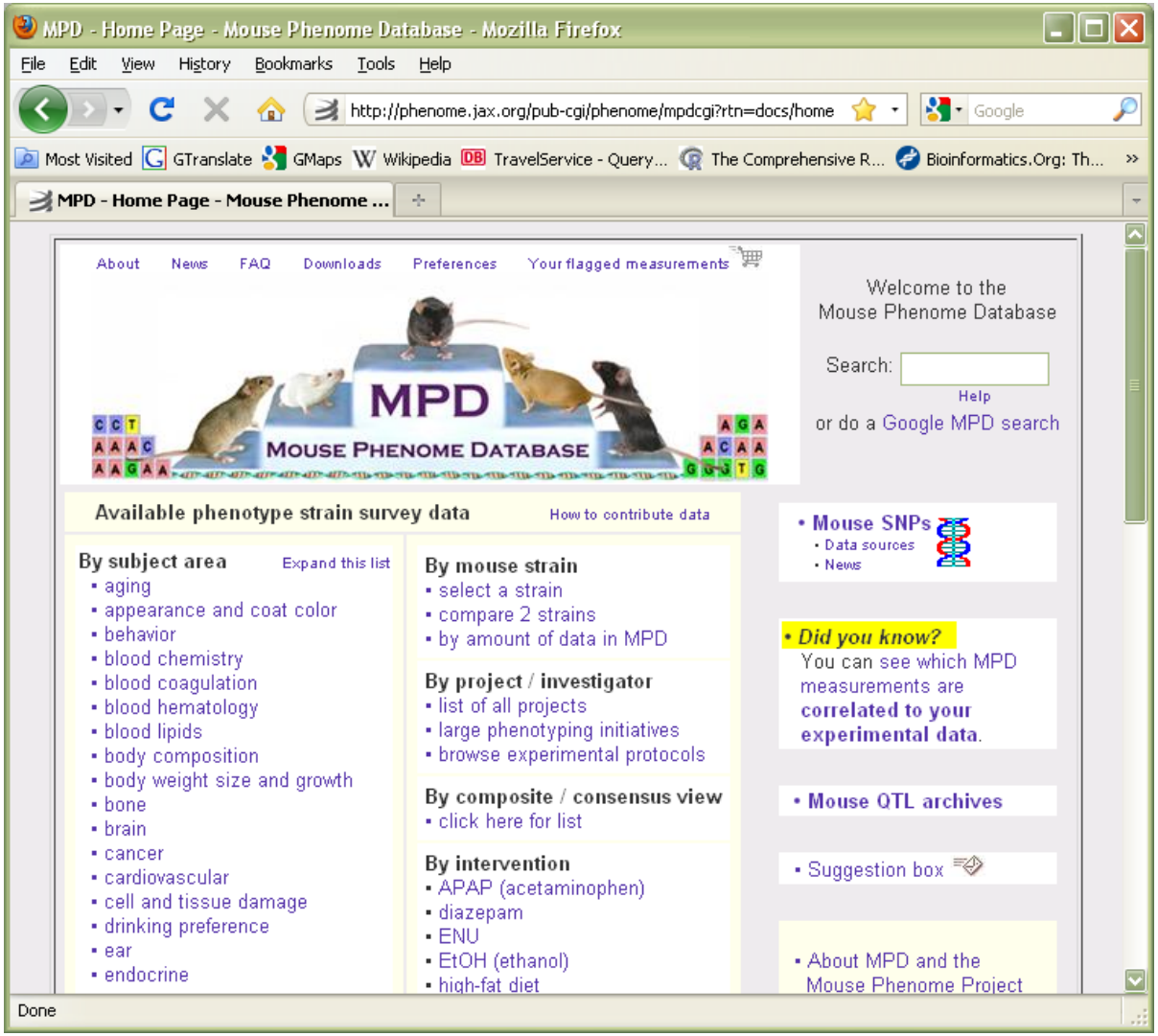








Mouse Phenome Data



mice.txt

Tordoff MG, Bachmanov AA
Survey of calcium & sodium intake and metabolism with bone and body composition data
Project symbol: Tordoff3
Accession number: MPD:103

790 mice from
40 different strains

http://phenome.jax.org

parameter
Starting age
Ending age
Starting weight
Ending weight
Weight change
Bleeding time
Ionized Ca in blood
Blood pH
Bone mineral density
Lean tissues weight
Fat weight

L1.7. Data Visualization

```
################################################################################
# L1.7. DATA VISUALIZATION
################################################################################

##------------------------------------------------------------------------------
## L1.7.1. Plot time-series and smooth
##------------------------------------------------------------------------------
## get data
Currency = read.table("http://edu.sablab.net/data/txt/currency.txt",
                      header=T,as.is=T)

## initiate window
windows(8,5)  # try x11()
## plot the currency behaviour for the last 10 years
plot(Currency$EUR)


## let's make it more beautiful
windows(8,5)
plot(Currency$EUR,col=3,pch=19,
     main="EUR/USD ratio for 11 years",
     ylab="EUR/USD",
     xlab="Measures (working days)")

## add smoothing. Try different "f"
smooth = lowess(Currency$EUR,f=0.1)
lines(smooth,col=2,lwd=2)
## add 1 level
abline(h=1,col=4,lty=2)

## (*) add years
year=1999  # an initial year
while (year<=2009){  # loop for all the years up to now
  idx=grep(paste("^",year,sep=""),Currency$Date)  # take the indexes of the measures for the "year"
  average=mean(Currency$EUR[idx]) # calculate the average ratio for the "year"
  abline(v=min(idx),col=1,lty=3)  # draw the year separator
  lines(x=c(min(idx),max(idx)),y=c(average,average),col=2)  # draw the average ratio for the "year"
  text(median(idx),max(Currency$EUR),sprintf("%d",year),font=2) # write the years
  text(median(idx),average+0.05,sprintf("%.2f",average),col=2,font=2,cex=0.8) # write the average ratio
  year=year+1;
}

##------------------------------------------------------------------------------
## L1.7.2. Mouse phenome :)
##------------------------------------------------------------------------------
## load data
Mice=read.table("http://edu.sablab.net/data/txt/mice.txt",
                header=T,sep="\t")
str(Mice)

## initiate window
windows(10,8)
par(mfrow=c(2,2))

## plot a factorial data
plot(Mice$Strain,las=2,
     col=rainbow(nlevels(Mice$Strain)),cex.names =0.7)
title("Number of mice from each strain")


## plot a factorial data as pie
pie(summary(Mice$Sex), col=c("pink","lightblue"))
title("Gender composition (f:female, m:male)")


## try to use special command "barplot" as well
## a histogram
hist(Mice$Starting.weight,probability = T,
     main="Histogram and p.d.f. approximation",
     xlab="weight, g")
lines(density(Mice$Starting.weight),lwd=2,col=4)


## (!) a box-plot of the population on the basis of sex
boxplot(Starting.weight~Sex,data=Mice,col=c("pink","lightblue"))
title("Weight by sex (f:female, m:male)",
      ylab="weight, g",xlab="sex")

##------------------------------------------------------------------------------
## L1.7.3. Show all data frame at once
##------------------------------------------------------------------------------

plot(Mice)

plot(Mice[,-(1:3)])

##------------------------------------------------------------------------------
## L1.7.4. 3D visualization and custom functions
##------------------------------------------------------------------------------

## see demo
demo(persp)

## use RGL library
library(rgl)

x=Mice$Starting.weight
y=Mice$Ending.weight
z=Mice$Fat.weight
plot3d(x,y,z)

## make it more beautiful
color = as.integer(Mice$Sex)*2
plot3d(x,y,z,
       col=color,type="s",radius=0.5,
       xlab="Starting weight",
       ylab="Ending weight",
       zlab="Fat weight")


#>>>>>>>>>>>>>>>>>>>>>>>>>
#> please, do Tasks L1.7ab
#>>>>>>>>>>>>>>>>>>>>>>>>>
```

Tasks L1.7a,b

Descriptive Statistics

Outline

- **Descriptive statistics in R (L1.1)**
  - sum, mean, median, sd, var, cor, *etc*.
- **Detection of outliers (L1.2)**
  - z-score, Iglewicz-Hoaglin, Grubb's test

L1.1. Descriptive Statistics in R

Population and Sample

Population parameter
A numerical value used as a summary measure for a population (e.g., the population mean μ, variance σ^2, standard deviation σ)

POPULATION

μ – mean
σ^2 – variance
N – number of elements (usually $N=\infty$)

SAMPLE

$m, \bar{x}$ – mean
s^2 – variance
n – number of elements

Sample statistic
A numerical value used as a summary measure for a sample (e.g., the sample mean m, sample variance s^2, and sample standard deviation s)

All existing laboratory *Mus musculus*

mice.txt

790 mice from different strains

http://phenome.jax.org

ID	Strain	Sex	Starting age	Ending age	Starting weight	Ending weight	Weight change	Bleeding time	Ionized Ca in blood	Blood pH	Bone mineral density	Lean tissues weight	Fat weight
1	129S1/SvImJ	f	66	116	19.3	20.5	1.062	64	1.2	7.24	0.0605	14.5	4.4
2	129S1/SvImJ	f	66	116	19.1	20.8	1.089	78	1.15	7.27	0.0553	13.9	4.4
3	129S1/SvImJ	f	66	108	17.9	19.8	1.106	90	1.16	7.26	0.0546	13.8	2.9
368	129S1/SvImJ	f	72	114	18.3	21	1.148	65	1.26	7.22	0.0599	15.4	4.2
369	129S1/SvImJ	f	72	115	20.2	21.9	1.084	55	1.23	7.3	0.0623	15.6	4.3
370	129S1/SvImJ	f	72	116	18.8	22.1	1.176		1.21	7.28	0.0626	16.4	4.3
371	129S1/SvImJ	f	72	119	19.4	21.3	1.098	49	1.24	7.24	0.0632	16.6	5.4
372	129S1/SvImJ	f	72	122	18.3	20.1	1.098	73	1.17	7.19	0.0592	16	4.1
4	129S1/SvImJ	f	66	109	17.2	18.9	1.099	41	1.25	7.29	0.0513	14	3.2
5	129S1/SvImJ	f	66	112	19.7	21.3	1.081	129	1.14	7.22	0.0501	16.3	5.2
10	129S1/SvImJ	m	66	112	24.3	24.7	1.016	119	1.13	7.24	0.0533	17.6	6.8
364	129S1/SvImJ	m	72	114	25.3	27.2	1.075	64	1.25	7.27	0.0596	19.3	5.8
365	129S1/SvImJ	m	72	115	21.4	23.9	1.117	48	1.25	7.28	0.0563	17.4	5.7
366	129S1/SvImJ	m	72	118	24.5	26.3	1.073	59	1.25	7.26	0.0609	17.8	7.1
367	129S1/SvImJ	m	72	122	24	26	1.083	69	1.29	7.26	0.0584	19.2	4.6
6	129S1/SvImJ	m	66	116	21.6	23.3	1.079	78	1.15	7.27	0.0497	17.2	5.7
7	129S1/SvImJ	m	66	107	22.7	26.5	1.167	90	1.18	7.28	0.0493	18.7	7
8	129S1/SvImJ	m	66	108	25.4	27.4	1.079	35	1.24	7.26	0.0538	18.9	7.1
9	129S1/SvImJ	m	66	109	24.4	27.5	1.127	43	1.29	7.29	0.0539	19.5	7.1

Measures of Location

Mean
A measure of central location computed by summing the data values and dividing by the number of observations.

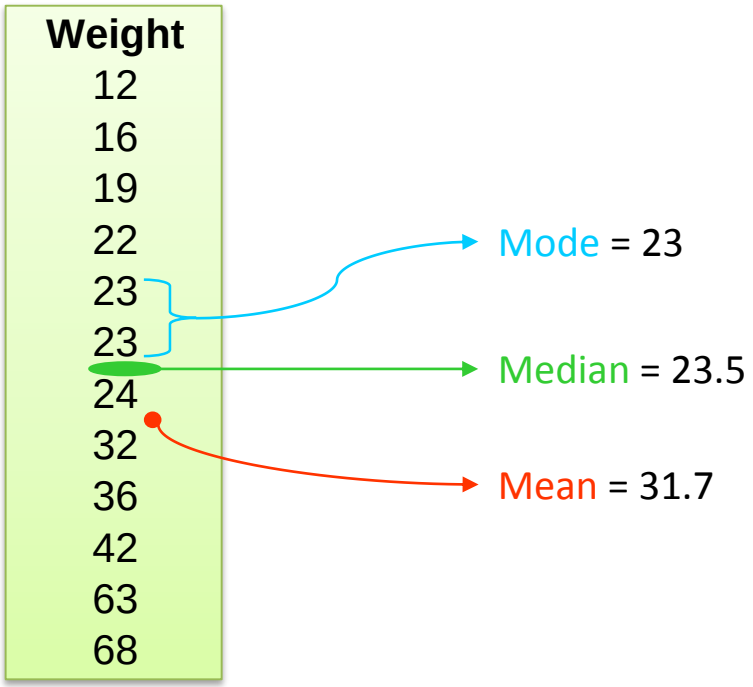
Median
A measure of central location provided by the value in the middle when the data are arranged in ascending order.

Mode
A measure of location, defined as the value that occurs with greatest frequency.

$$m = \bar{x} = \frac{\sum x_i}{n}$$

$$\mu = \frac{\sum x_i}{N}$$

$$p = \frac{\sum (x_i = TRUE)}{n}$$



L1.8. Descriptive Statistics in R

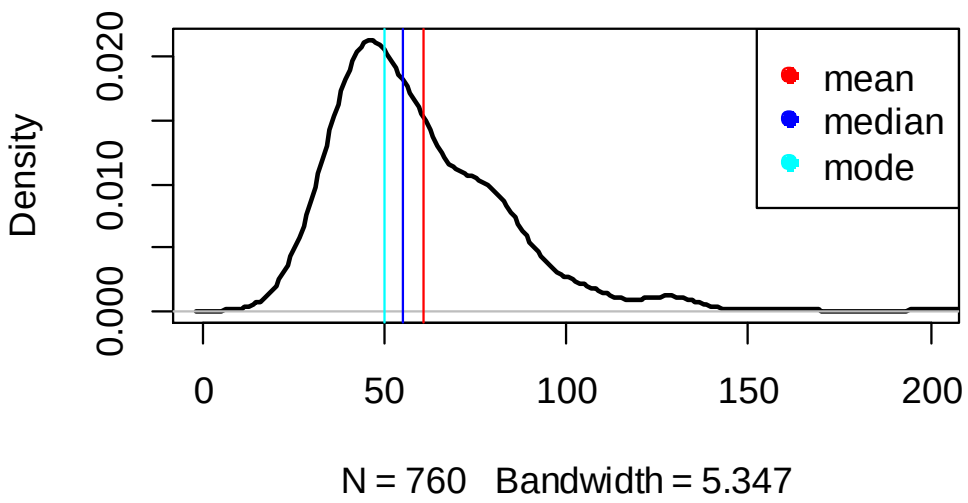
Measures of Location

In R use the following functions:

- `mean(x, na.rm=T)`
- `median(...)`
- `library(modeest)`
  `mlv(...)$M`

To by applied to data with missing elements (NA), use parameter: `..., na.rm = T`

`mice.txt`



In Excel use the following functions:

- `=AVERAGE(data)`
- `=MEDIAN(data)`
- `=MODE(data)`

To calculate proportion – count occurrence and divide by total number of elements:

```
prop.f = sum(Mice$Sex=="f")/nrow(Mice)
> 0.501
```

L1.8. Descriptive Statistics in R

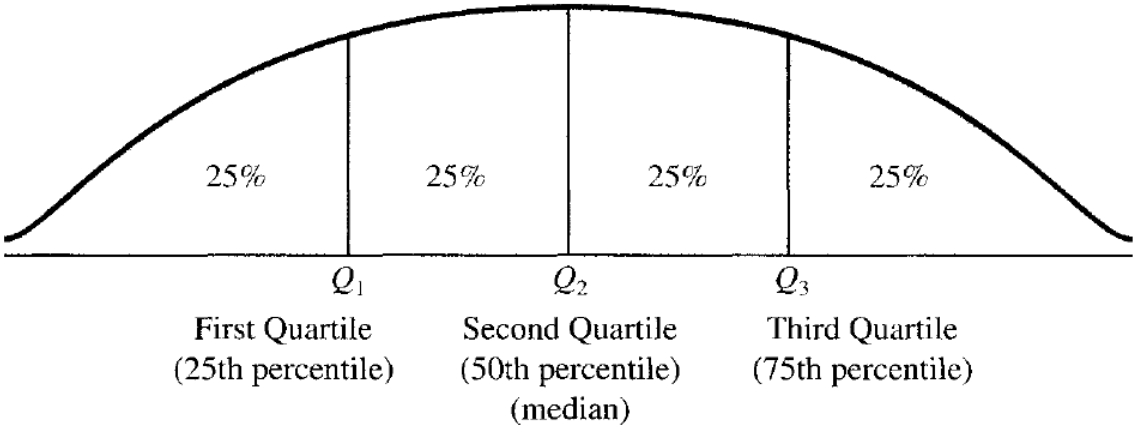
Quantiles, Percentiles and Quartiles

Percentile

A value such that at least p% of the observations are less than or equal to this value, and at least (100-p)% of the observations are greater than or equal to this value. The 50-th percentile is the *median*.

Quartiles

The 25th, 50th, and 75th percentiles, referred to as the **first quartile**, the **second quartile** (median), and **third quartile**, respectively.



In R use:

- `quantile(x,...)`

R provides up to 7 methods to estimate quantiles. Use parameter:

`type =` put a number 1-7

Weight	12	16	19	22	23	23	24	32	36	42	63	68

$Q_1 = 21$ $Q_2 = 23.5$ $Q_3 = 39$

In Excel use the following functions:

- `=PERCENTILE(data,p)`

figure is adapted from Anderson et al Statistics for Business and Economics

Measures of Variation

Interquartile range (IQR)

A measure of variability, defined to be the difference between the third and first quartiles.

$$IQR = Q_3 - Q_1$$

Variance

A measure of variability based on the squared deviations of the data values about the mean.

population

$$\sigma^2 = \frac{\sum(x_i - \mu)^2}{N}$$

sample

$$s^2 = \frac{\sum(x_i - \mu)^2}{n - 1}$$

Standard deviation

A measure of variability computed by taking the positive square root of the variance.

In R use:

- `IQR (x,...)`
- `sd (x,...)`
- `var (x,...)`

Weight	12	16	19	22	23	23	24	32	36	42	63	68

IQR = 18 *Variance* = 320.2 *St. dev.* = 17.9

In Excel use the following functions:

- `= STDEV(data)`
- `= VAR(data)`

Measures of Variation

Coefficient of variation

A measure of relative variability computed by dividing the standard deviation by the mean.

$$C_V = \frac{\sigma}{\mu}$$

Weight	12	16	19	22	23	23	24	32	36	42	63	68

C_V = 57%

Median absolute deviation (MAD)

MAD is a robust measure of the variability of a univariate sample of quantitative data.

$$MAD = 1.4826 \cdot med(|x_i - med(x)|)$$

In R use:

- `mad (x,...)`

Set 1	Set 2
23	23
12	12
22	22
12	12
21	21
18	81
22	22
20	20
12	12
19	19
14	14
13	13
17	17

Constant 1.4826 is introduced to ensure that MAD $\rightarrow \sigma$ for normal distribution. Can be modified by `constant = ...`

	Set 1	Set 2
Mean	17.3	22.2
Median	18	19
St.dev.	4.23	18.18
MAD	5.93	5.93

Box-plot

Five-number summary

An exploratory data analysis technique that uses five numbers to summarize the data: smallest value, first quartile, median, third quartile, and largest value

`children.txt`

Min. :	12
Q_1 :	25
Median:	32
Q_3 :	46
Max. :	79

In R use:

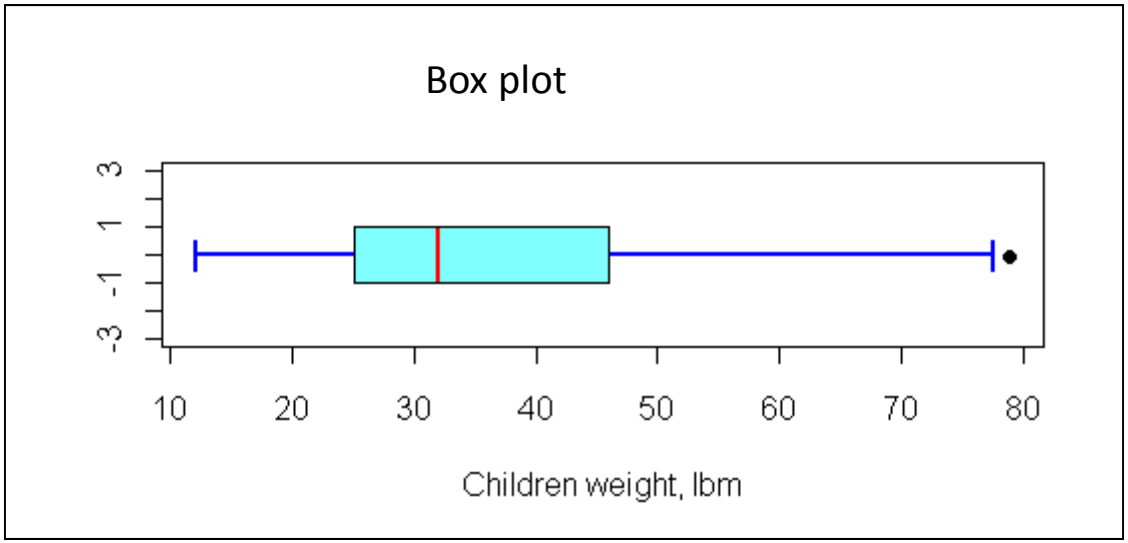
- `summary(x)`

Box plot

A graphical summary of data based on a five-number summary

In R use:

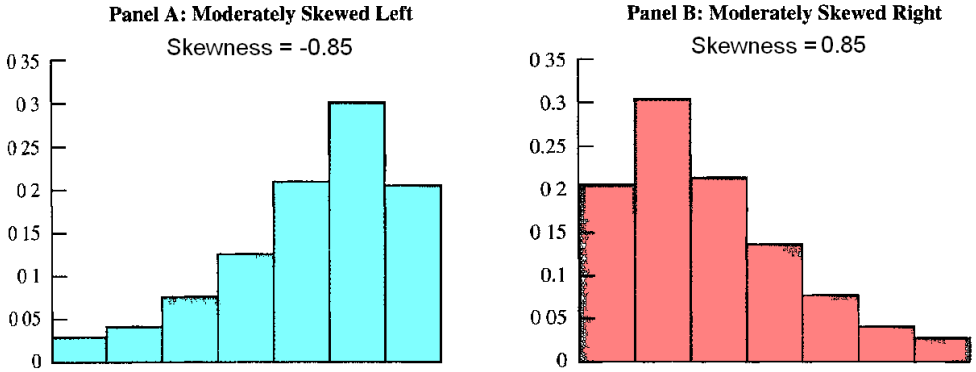
- `boxplot(...)`



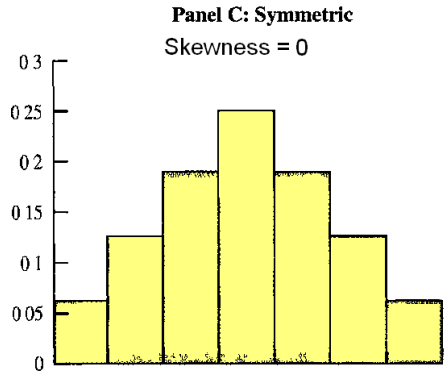
Other Parameters

Skewness

A measure of the shape of a data distribution. Data skewed to the left result in negative skewness; a symmetric data distribution results in zero skewness; and data skewed to the right result in positive skewness.



$$skw = \frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - m}{s} \right)^3$$



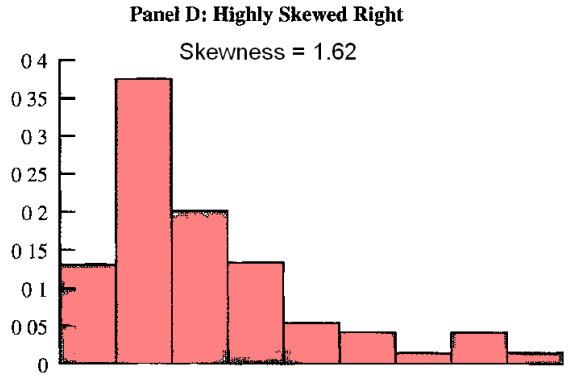


figure is adapted from Anderson et al Statistics for Business and Economics

In R use:

- `library(e1071)`
  `skewness(x,...)`
- `library(modeest)`
  `skewness(x,...)`

Measure of Association between 2 Variables

Pearson Correlation (Pearson product moment correlation coefficient)

A measure of linear association between two variables that takes on values between -1 and +1. Values near +1 indicate a strong positive linear relationship, values near -1 indicate a strong negative linear relationship; and values near zero indicate the lack of a linear relationship.

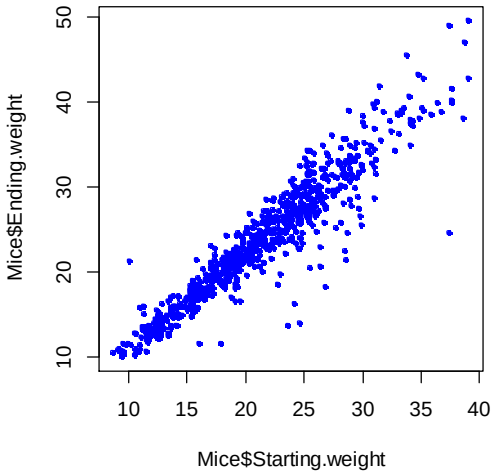
population

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = \frac{\sum(x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y N}$$

sample

$$r_{xy} = \frac{s_{xy}}{s_x s_y} = \frac{\sum(x_i - m_x)(y_i - m_y)}{s_x s_y (n-1)}$$

`mice.xls`



r_{xy} = 0.94

In R use:

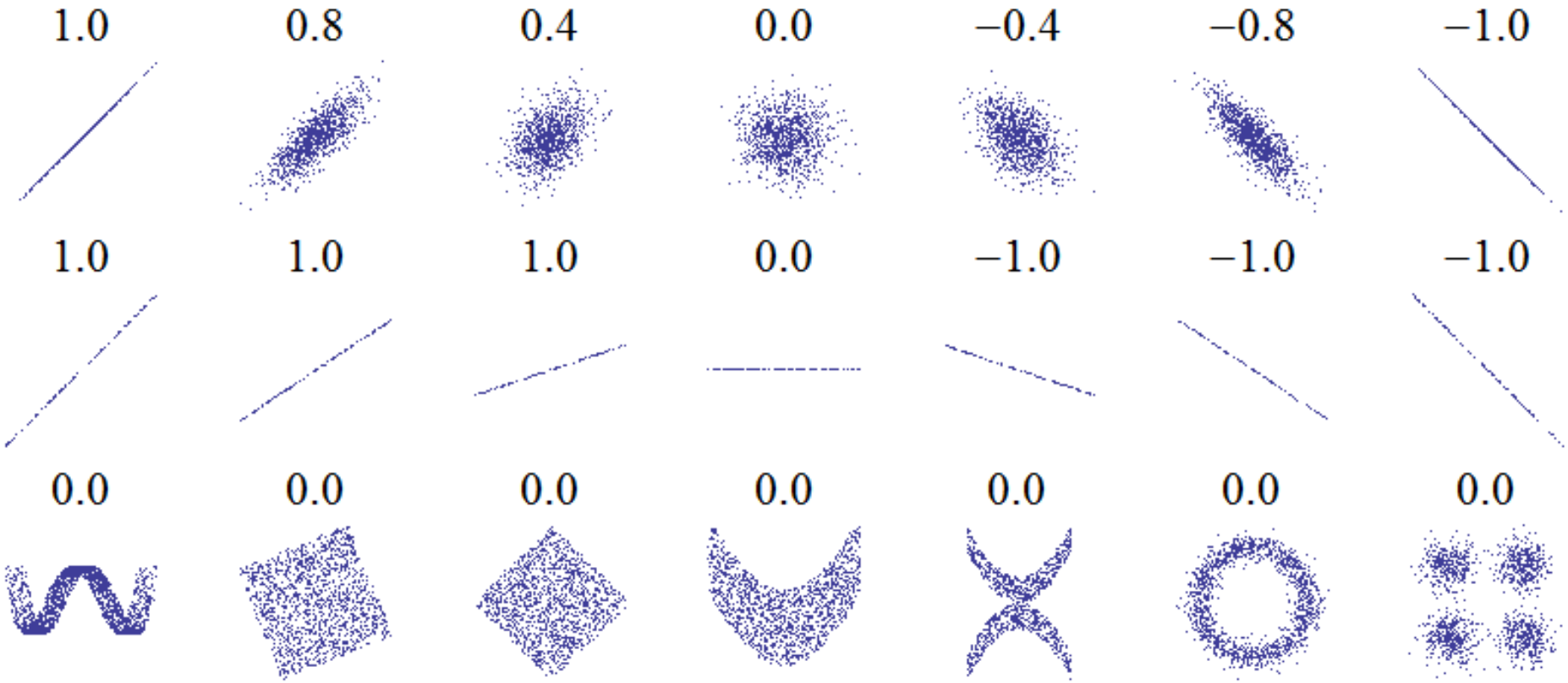
- `cor (x,...)`

For missing data add parameter
`use = "pairwise.complete.obs"`

In Excel use function:

- `=CORREL(data)`

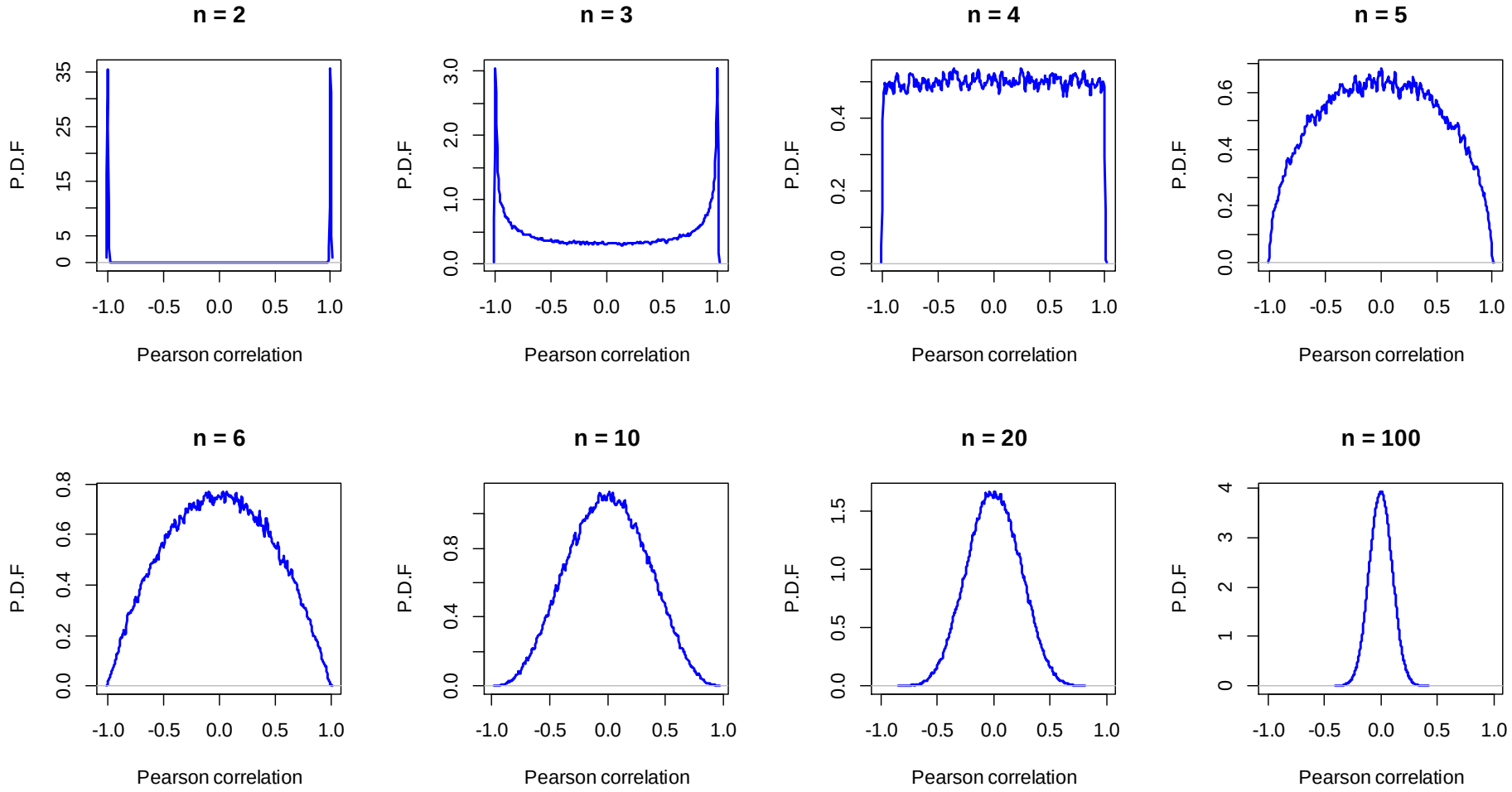
Pearson Correlation



Wikipedia

? If we have only 2 data points in *x* and y datasets, what values would you expect for correlation b/w *x* and y ?

Pearson Correlation: Effect of Sample Size



L1.8. Descriptive Statistics in R

Nonparametric Measures of Association

Kendal Correlation, τ (Kendall tau rank correlation)

a non-parametric measure of rank correlation: that is, the similarity of the orderings of the data when ranked by each of the quantities.

All combination of data pairs (x_i, y_i), (x_j, y_j) are checked.

2 pairs are concordant if:

$$(x_i - x_j)(y_i - y_i) > 0$$

2 pairs are discordant if:

$$(x_i - x_j)(y_i - y_i) < 0$$

In case of = 0 pair is not considered.

Let number of corresponding pairs be $n_{concordant}$ and $n_{discordant}$

$$\tau = 2\frac{n_{concordant} - n_{discordant}}{n(n-1)}$$

Spearman's Correlation, ρ (Spearman's rank correlation)

a non-parametric measure of statistical dependence between two variables. It assesses how well the relationship between two variables can be described using a monotonic function.

Data (x_i, y_i) are replaced by their ranks, let's denote them (X_i, Y_i). Then Person correlation is measured b/w ranks:

$$\rho_{xy} = \frac{\sum(X_i - m_X)(Y_i - m_Y)}{\sum(X_i - m_Y)\sum(Y_i - m_Y)}$$

In R use:

- `cor(x,method="kendal",...)`
- `cor(x,method="spearman",...)`

For missing data add parameter

`use = "pairwise.complete.obs"`

L1.8. Descriptive Statistics in R

```
##############################################################################
# L1.8. DESCRIPTIVE STATISTICS
##############################################################################

## clear memory
rm(list = ls())

## load data
Mice=read.table("http://edu.sablab.net/data/txt/mice.txt",
                header=T,sep="\t")
str(Mice)

##----------------------------------------------------------------------------
## L1.8.1. Measures of center
##----------------------------------------------------------------------------
summary(Mice)

## mean and median
mean(Mice$Ending.weight)
median(Mice$Ending.weight)
## for mode you should add a library:
library(modeest)
mlv(Mice$Ending.weight, method = "shorth")$M


## mean and median if NA values present: add na.rm=T
mn = mean(Mice$Bleeding.time, na.rm=T)
md = median(Mice$Bleeding.time, na.rm=T)
mo = mlv(Mice$Bleeding.time, method = "shorth", na.rm=T)$M

## let us plot them
x11()
plot(density(Mice$Bleeding.time, na.rm=T), xlim=c(0,200),lwd=2,main="Bleeding time")
abline(v = mn,col="red")
abline(v = md,col="blue")
abline(v = mo ,col="cyan")
legend(x="topright",c("mean","median","mode"),col=c("red","blue","cyan"),pch=19)

prop.f = sum(Mice$Sex=="f")/nrow(Mice)

##----------------------------------------------------------------------------
## L1.8.2. Measures of variation
##----------------------------------------------------------------------------

## quantiles, percentiles and quartiles
quantile(Mice$Bleeding.time,prob=c(0.25,0.5,0.75),na.rm=T)

## standard deviation and variance
sd(Mice$Bleeding.time, na.rm=T)
var(Mice$Bleeding.time, na.rm=T)

## stable measure of variation - MAD
mad(Mice$Bleeding.time, na.rm=T)
mad(Mice$Bleeding.time, constant = 1, na.rm=T)
```

```
##----------------------------------------------------------------------------
## L1.8.3. Measures of dependency
##----------------------------------------------------------------------------

## covariation
cov(Mice$Starting.weight,Mice$Ending.weight)

## correlation
cor(Mice$Starting.weight,Mice$Ending.weight)

## coefficient of determination, R2
cor(Mice$Starting.weight,Mice$Ending.weight)^2

## kendal correlation
cor(Mice$Starting.weight,Mice$Ending.weight,method="kendal")
## spearman correlation
cor(Mice$Starting.weight,Mice$Ending.weight,method="spearman")


#>>>>>>>>>>>>>>>>>>>>>>>>>>
#> please, do Task L1.8
#>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Task L1.8

L1.9. Detection of Outliers

z-score and Chebyshev's Theorem

z-score

A value computed by dividing the deviation about the mean (x_i x) by the standard deviation s. A *z-score* is referred to as a standardized value and denotes the number of standard deviations x_i is from the mean.

$$z_i = \frac{x_i - m}{s}$$

In R use:

- `scale(x,...)`

Weight	z-score
12	-1.10
16	-0.88
19	-0.71
22	-0.54
23	-0.48
23	-0.48
24	-0.43
32	0.02
36	0.24
42	0.58
63	1.75
68	2.03

Chebyshev's theorem

For **any data set**, at least $(1 - 1/z^2)$ of the data values must be within z standard deviations from the mean, where z – any value > 1.

For ANY distribution:

- At least 75 % of the values are within z = 2 standard deviations from the mean
- At least 89 % of the values are within z = 3 standard deviations from the mean
- At least 94 % of the values are within z = 4 standard deviations from the mean
- At least 96% of the values are within z = 5 standard deviations from the mean

Outliers

For bell-shaped distributions:

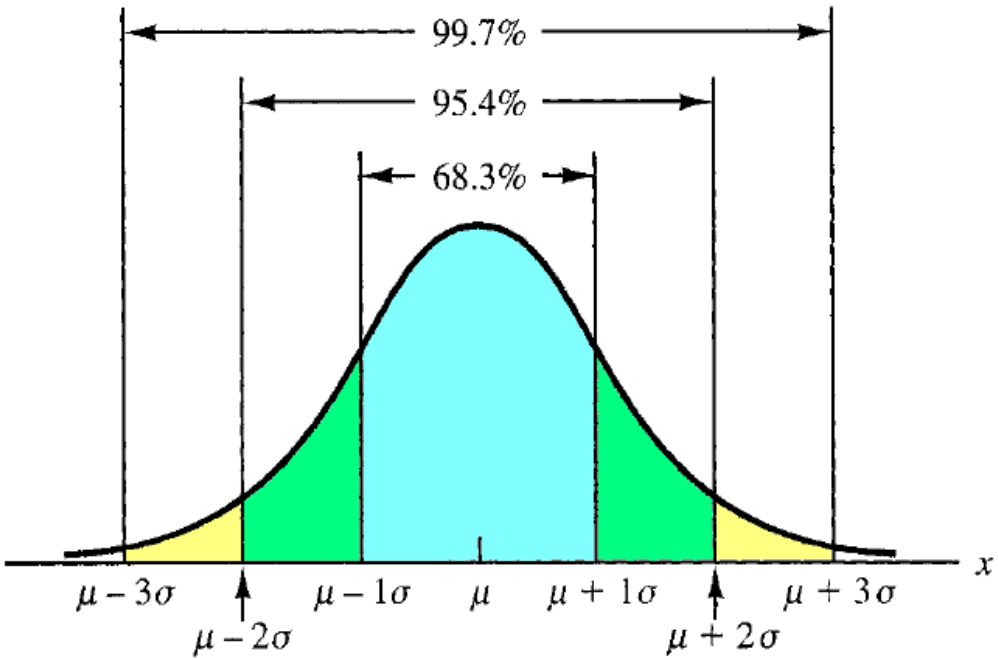
- ~ 68 % of the values are within 1 st.dev. from mean
- ~ 95 % of the values are within 2 st.dev. from mean
- Almost all data points are inside 3 st.dev. from mean

Outlier
An unusually small or unusually large data value.

For bell-shaped distributions data points with |z|>3 can be considered as outliers.

Weight	z-score
23	0.04
12	-0.53
22	-0.01
12	-0.53
21	-0.06
81	**3.10**
22	-0.01
20	-0.11
12	-0.53
19	-0.17
14	-0.43
13	-0.48
17	-0.27

Example: Gaussian distribution



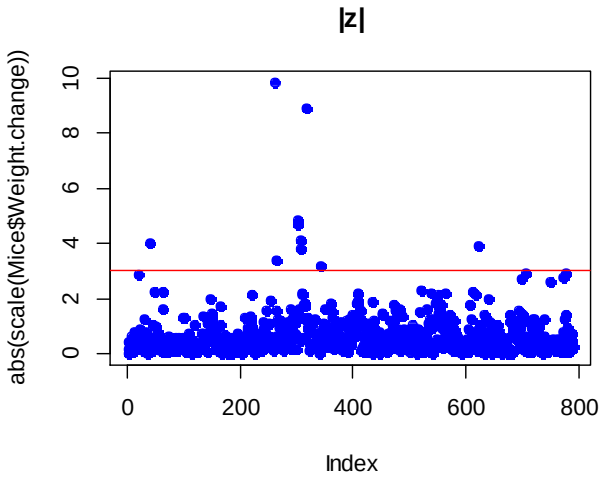
Simplest Method to Detect Outliers

mice.xls

Try to identify outlier mice on the basis of *Weight change* variable

$$z_i = \frac{x_i - m}{s}$$

For bell-shaped distributions data points with |z|>3 can be considered as outliers.



- Calculate z-score by **`scale(...)`**
- Measurements with z-score > 3 are potential outliers

L1.9. Detection of Outliers

Iglewicz-Hoaglin Method

$$z_i = \frac{x_i - med(x)}{MAD(x)}$$

***med*(x)** – median

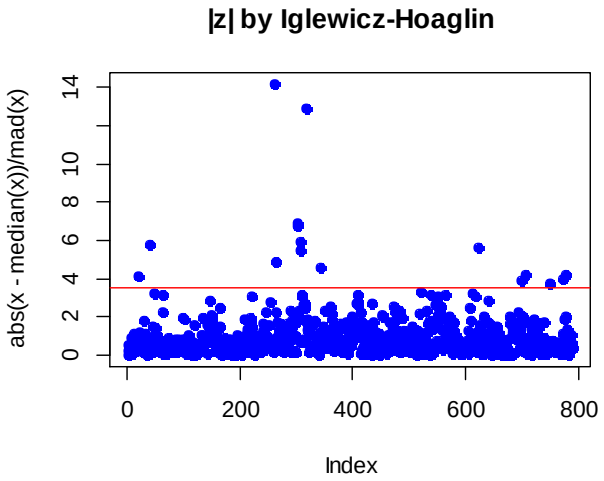
***MAD*(x)** – median absolute deviation with constant = 1.4826

$$if \ \ |z_i| > 3.5 \Rightarrow x_i \ - outlier$$

`mice.xls`

In R use:

- `abs(x-median(x))/mad(x)`



Boris Iglewicz and David Hoaglin (1993), "Volume 16: How to Detect and Handle Outliers", The ASQC Basic References in Quality Control: Statistical Techniques, Edward F. Mykytka, Ph.D., Editor

http://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm

Grubb's Method

Grubbs' test is an iterative method to detect outliers in a data set assumed to come from a normally distributed population.

Grubbs' statistics at step k+1:

$$G_{(k+1)} = \frac{\max|x_i - m_{(k)}|}{s_{(k)}} = \max|z_i|_{(k)}$$

(k) – iteration k
m – mean of the rest data
s – st.dev. of the rest data

The hypothesis of no outliers is rejected at significance level α if

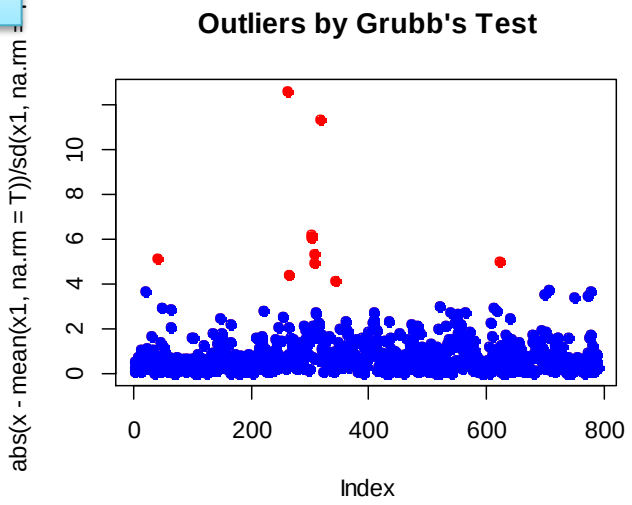
$$G > \frac{n-1}{\sqrt{n}} \sqrt{\frac{t^2}{n-2+t^2}}$$

where $t^2 = t^2_{\alpha/(2n),\ d.f.=n-2}$

t – Student statistics

In R use:

```
library(outliers)
x1=x
while(grubbs.test(x1)$p.value<0.05)
   x1[x1==outlier(x1)]=NA
```



Remember!

Generally speaking, removing of outliers is a dangerous procedure and cannot be recommended!

Instead, potential outliers should be investigated and only (!) if there are other evidences that data come from experimental error – removed.

L1.9. Descriptive Statistics in R

```
################################################################################
# L1.9. DETECTION OF OUTLIERS
################################################################################

## clear memory
rm(list = ls())

## load data
Mice=read.table("http://edu.sablab.net/data/txt/mice.txt",
                header=T,sep="\t")
str(Mice)

x=Mice$Weight.change
##------------------------------------------------------------------------------
## L1.9.1. z-score
##------------------------------------------------------------------------------
plot(abs(scale(x)),pch=19,col=4,main="|z|")
abline(h=3,col=2)

##------------------------------------------------------------------------------
## L1.9.2. Iglewicz-Hoaglin method
##------------------------------------------------------------------------------
plot(abs(x-median(x))/mad(x),pch=19,col=4,main="|z| by Iglewicz-Hoaglin")
abline(h=3.5,col=2)

##------------------------------------------------------------------------------
## L1.9.3. Grubb's method
##------------------------------------------------------------------------------
library(outliers)
x1=x
while(grubbs.test(x1)$p.value < 0.05){
  x1[x1==outlier(x1)]=NA
}
plot(abs(x-mean(x1,na.rm=T))/sd(x1,na.rm=T),pch=19,col=2,main="Outliers by Grubb's Test")
points(abs(x1-mean(x1,na.rm=T))/sd(x1,na.rm=T),pch=19,col=4)

#>>>>>>>>>>>>>>>>>>>>>>>>>
#> please, do Task L1.9
#>>>>>>>>>>>>>>>>>>>>>>>>>
```

Task L1.9

Questions ?

Thank you for your attention

to be continued…