LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Introduction to R

Petr Nazarov

petr.nazarov@lih.lu

10 / 11-05-2017

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Outline

- **Information package, Installation, R interface (S1)**
- **Variables and basic operations (S2)**
- **Data import and export (S3)**
- **Control workflow and custom functions (S4)**
- **Data visualization (S5)**

Optional

- Descriptive statistics (S6)
- Statistical tests (S7)
- Linear models (S8)
- PCA and Clustering (S9)

S1. Information Package

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

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

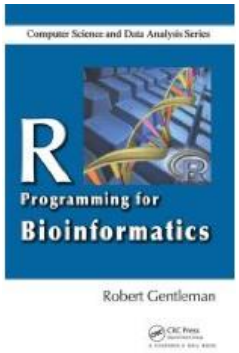
R-Project Seek Engine:
www.rseek.org

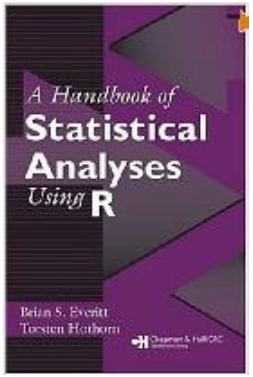
Advanced Biostatistics
edu.sablab.net/r2017

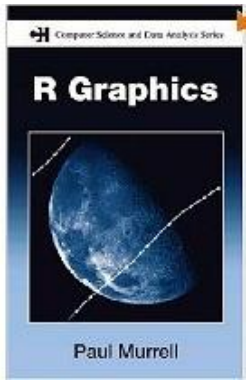
Scripts
edu.sablab.net/r2017/scripts

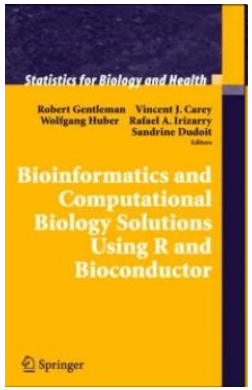
Data
edu.sablab.net/data/txt

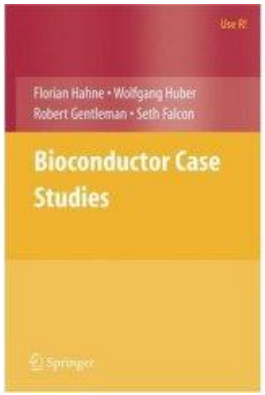
Other courses (statistics)
edu.sablab.net







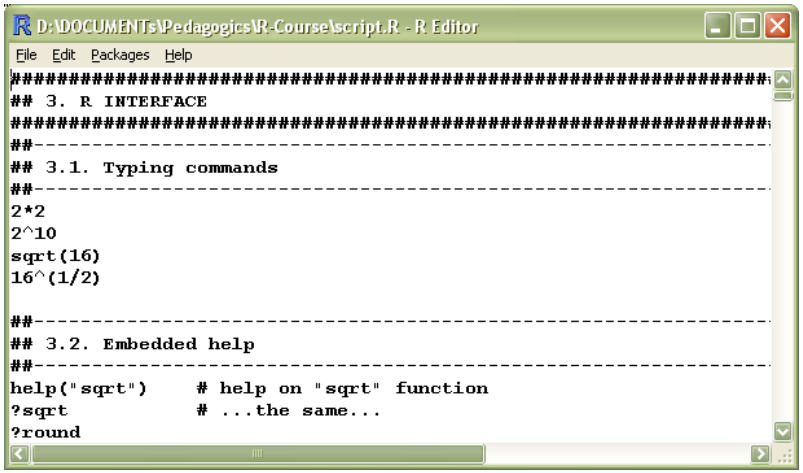




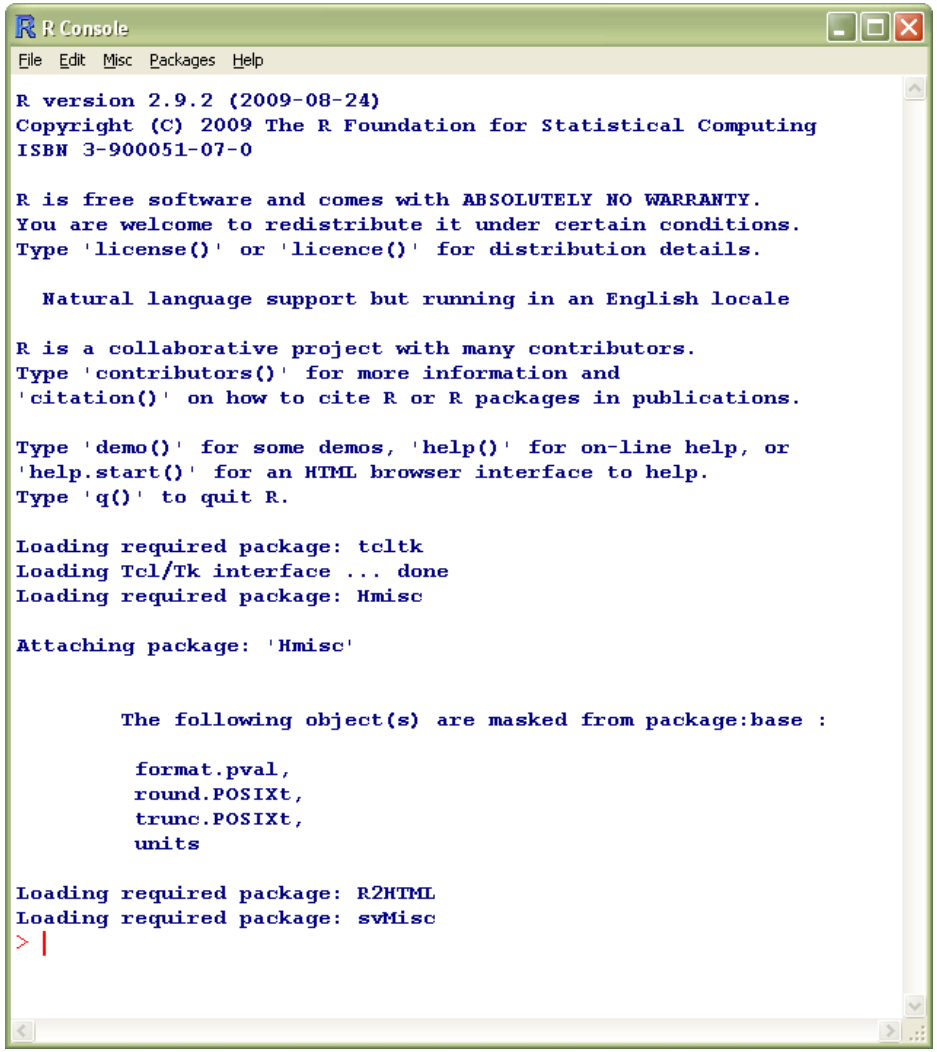
S1. R Interface

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

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/

S1. Installation

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

lihguest
welcometolih

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 (sometimes 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/

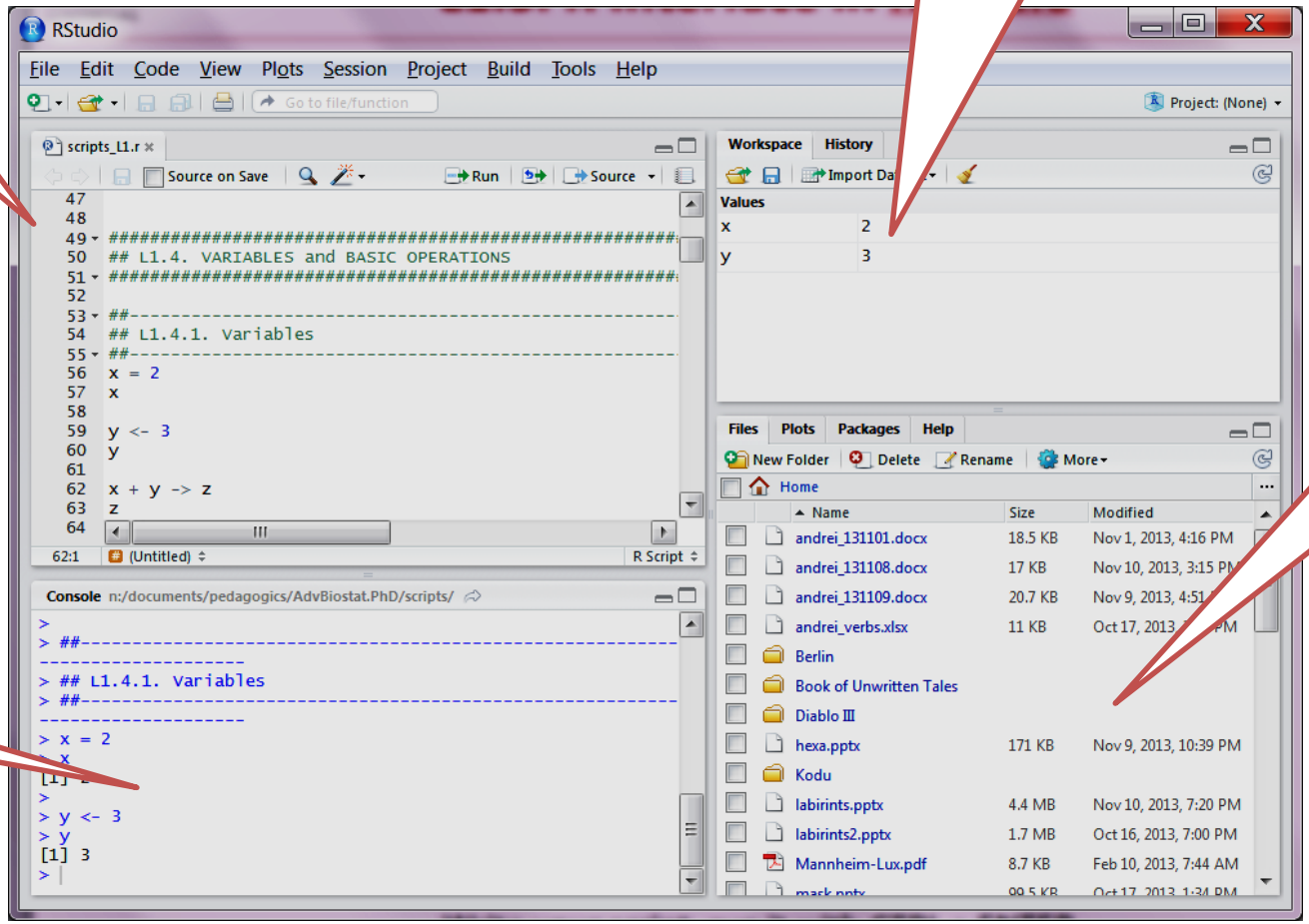
S1. R Interface in RStudio

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

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

Sripts

Variables and History



Files, Plots, Packages, Help

Console

Write your script, run it with CTRL + ENTER

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

S1.(1-3) Commands, Function and Help

```
################################################################################
## S1. 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
################################################################################
## S1. R INTERFACE
################################################################################
##------------------------------------------------------------------------------
## S1.1. Typing commands
##------------------------------------------------------------------------------
2*2
2^10
sqrt(16)
16^(1/2)
##------------------------------------------------------------------------------
## S1.2. Calling functions
##------------------------------------------------------------------------------
log(100)
log(100, base=10)
log(100, b=10)
log(100, 10)
##------------------------------------------------------------------------------
## S1.3. 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)
```

S2. Types of Variables in R

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

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

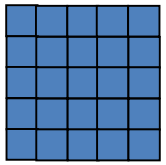
Vector

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

Matrix

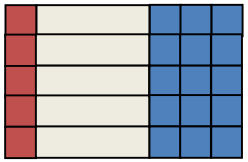
faster then data frame and list other

Array



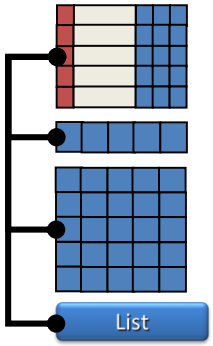
```
> A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    1
```

Data frame



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

List



Object of a Class

- Data
- Methods

S2. Types of Variables in R

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

S2.(1-3). Variables, Scalar Types, Special Values

```
################################################################################
## S2. VARIABLES and BASIC OPERATIONS
################################################################################
##------------------------------------------------------------------------------
## S2.1. 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()

##------------------------------------------------------------------------------
## S2.2. 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 S4a
#>>>>>>>>>>>>>>>>>>>>>>>>>

##--------------------------
## 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)

##------------------------------------------------------------------------------
## S2.3. 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 S4a

S2. Types of Variables in R

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

S2.(4-6). Vectors, Matrixes, Data Frames, Lists

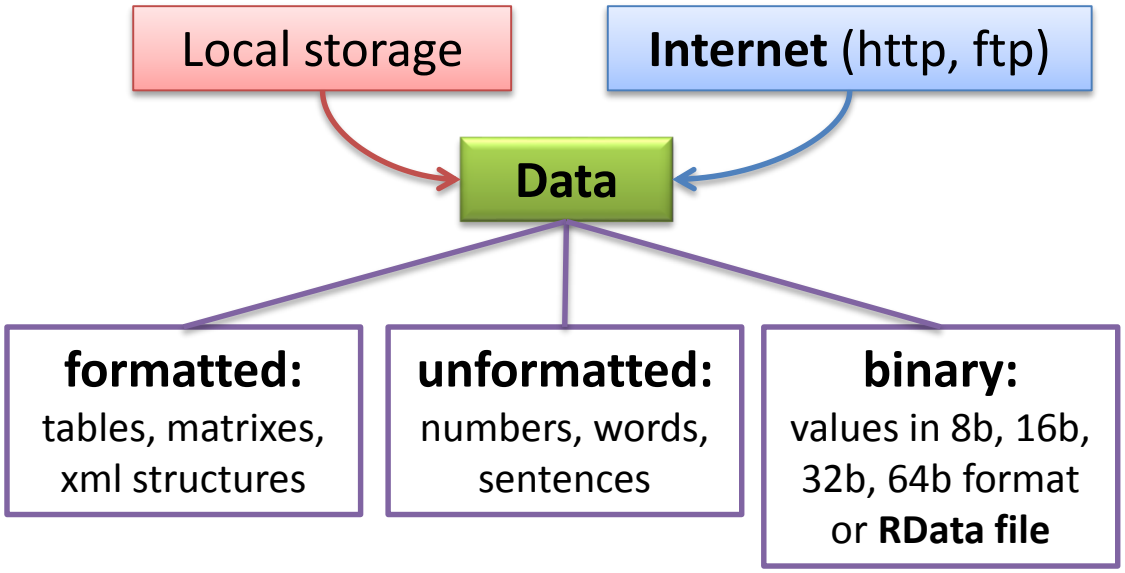
```
##-----------------------------------------------------------------------------
## S2.4. Vectors
##-----------------------------------------------------------------------------
## Vector creation
a = c(1,2,3,4,5)
a
a[1]+a[4]
b=5:9
a+b   #(*) try b=5:10. Can you explain the effect? (ans: "!tfihs ralucriC" :)
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  S4b,c,d
#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

##-----------------------------------------------------------------------------
## S2.5. 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
```

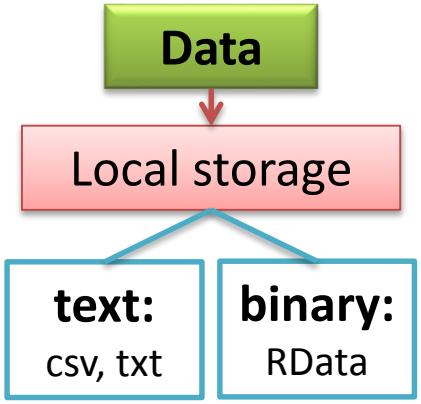
```
## put in the data manualy
Data$name=sprintf("Mouse_%d",1:5)
Data$sex=c("Male","Female","Female","Male","Male")
Data$weight=c(21,17,20,22,19)
Data$age=c(160,131,149,187,141)
Data$survival=c(T,F,T,F,T)
Data$code = 1:nrow(Data)
Data

## 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

##-----------------------------------------------------------------------------
## S2.6. 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 S4b,c,d

S3. Data Import and Export

Ways to Import and Export Data





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

S3. Data Import and Export

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

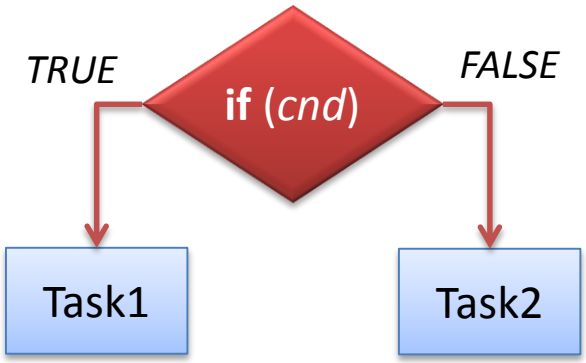
Script

```
################################################################################
# S3. DATA IMPORT AND EXPORT
################################################################################
##------------------------------------------------------------------------------
## S3.1. Current folder
##------------------------------------------------------------------------------
getwd()   ## shows current folder
dir()     ## shows files in the current folder
setwd("E:/DOCUMENTs/Pedagogics/R-Course_2010/Data") ## sets folder

##------------------------------------------------------------------------------
## S3.2. 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

##------------------------------------------------------------------------------
## S3.3. 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)

##------------------------------------------------------------------------------
## S3.4. "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")
```

```
##------------------------------------------------------------------------------
## S3.5. 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 S5a, S5b
#>>>>>>>>>>>>>>>>>>>>>>>>>
```

Tasks S3 a,b

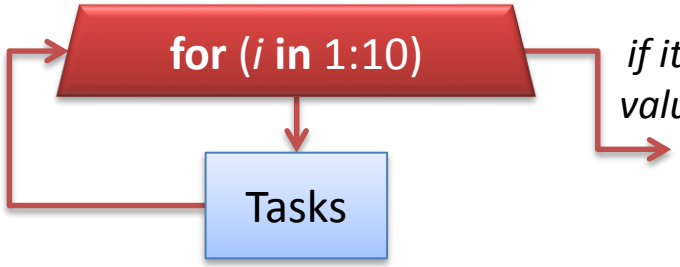
Main Workflow Control Methods




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

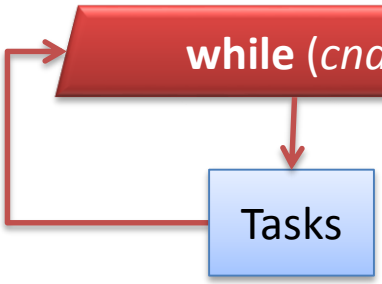
Functional version:

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



*if iterator **i** took all values of the vector*

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



if cnd = FALSE

```
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.*

S4. Control Workflow and Custom Functions

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Optional

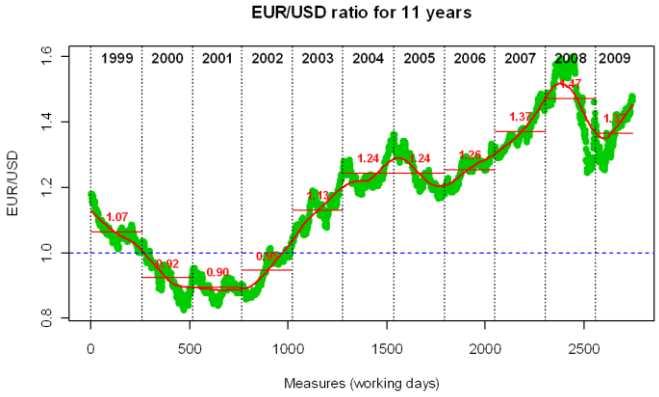
```
##############################################################################
## S4. CONTROL WORKFLOW and CUSTOM FUNCTIONS
##############################################################################
Shop = read.table("http://edu.sablab.net/data/txt/shop.txt",header=T,sep="\t")
a=1
b=2
##------------------------------
## 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
```

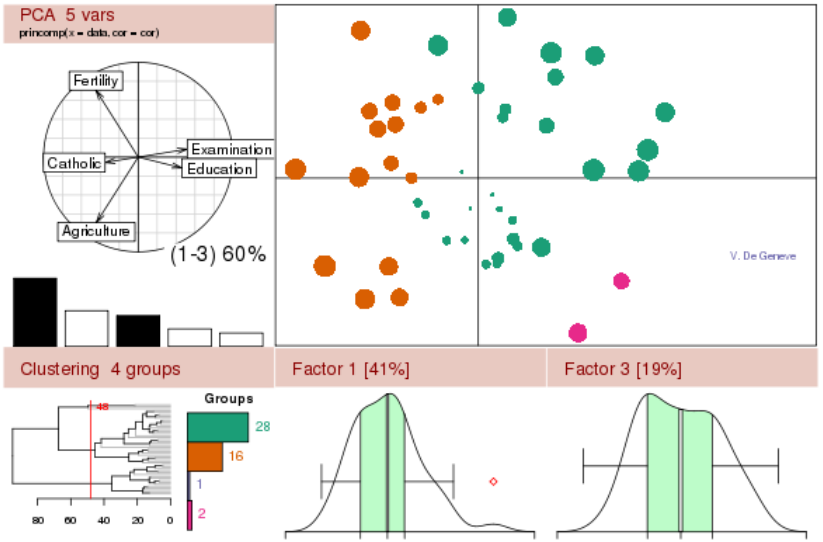
```
##------------------------------
## Custom functions
##------------------------------

## 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()
```

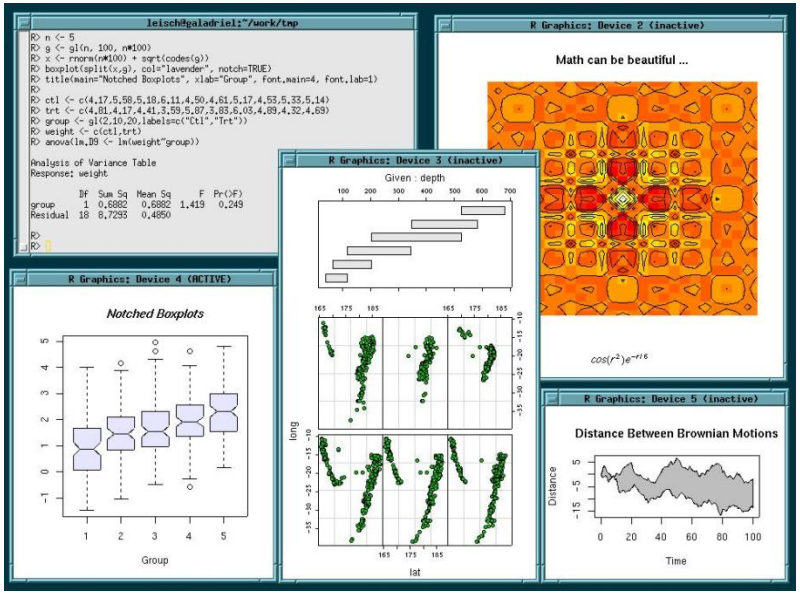
Tasks S4a,b

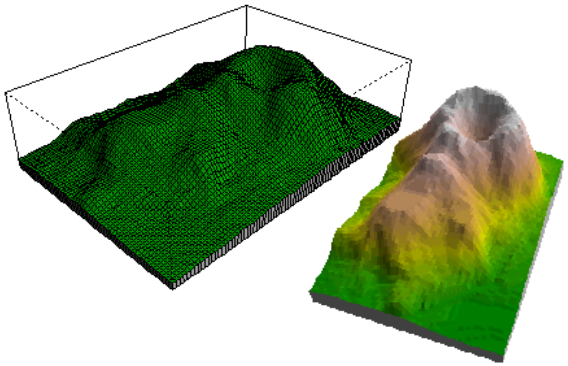
S5. Data Visualization

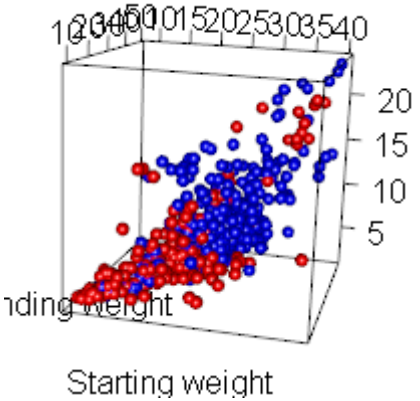
Various Figures Generated in R





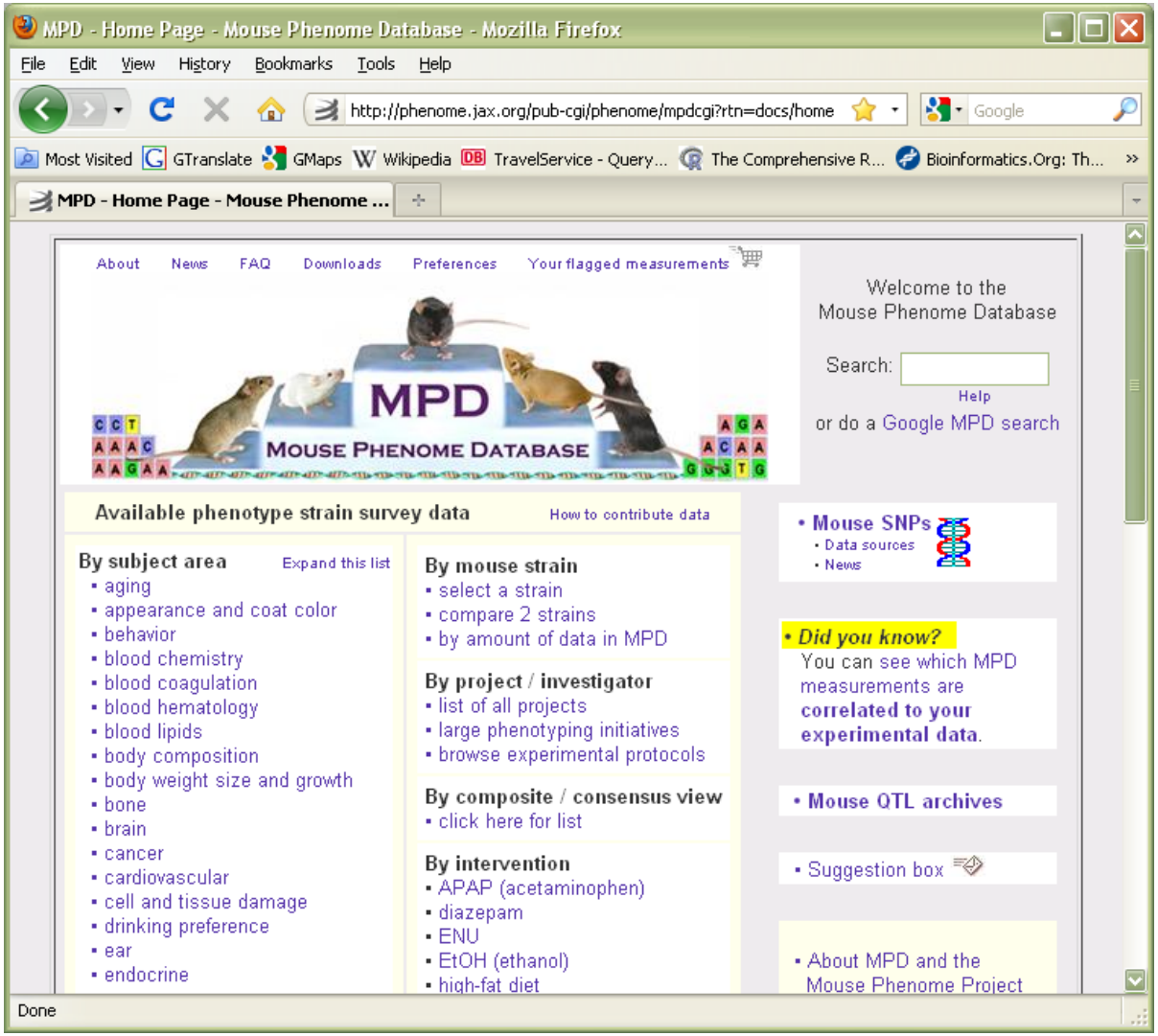






S5. Data Visualization

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

S5. Data Visualization

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

```
################################################################################
# S5. DATA VISUALIZATION
################################################################################

##-------------------------------------------------------------------------------
## S5.1. 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;
}

##-------------------------------------------------------------------------------
## S5.2. 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")

##-------------------------------------------------------------------------------
## S5.3. Show all data frame at once
##-------------------------------------------------------------------------------

plot(Mice)

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

##-------------------------------------------------------------------------------
## S5.4. 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 S5 ab
#>>>>>>>>>>>>>>>>>>>>>>>>>
```

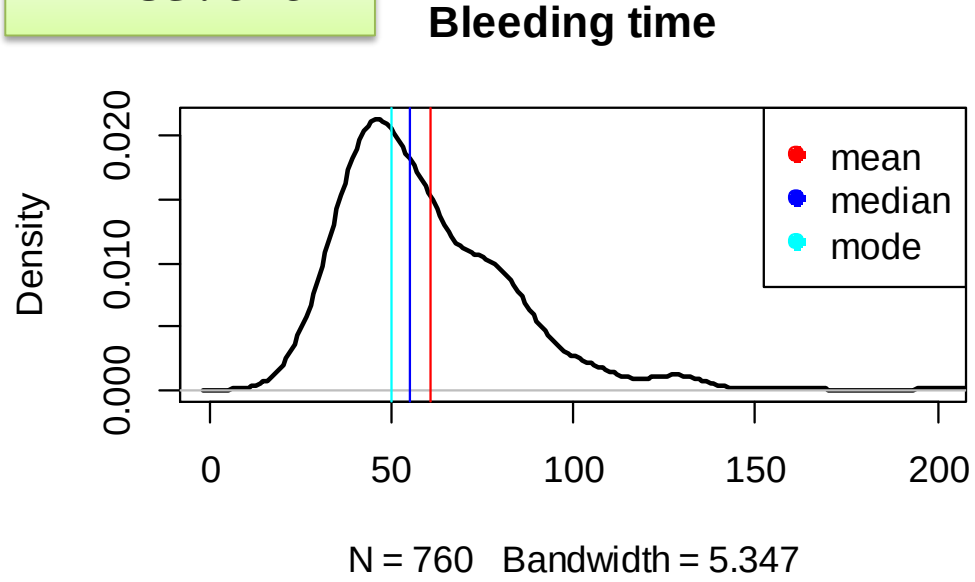
Tasks S5 a,b

Measures of Location

In R use the following functions:

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

`mice.txt`



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

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
```

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

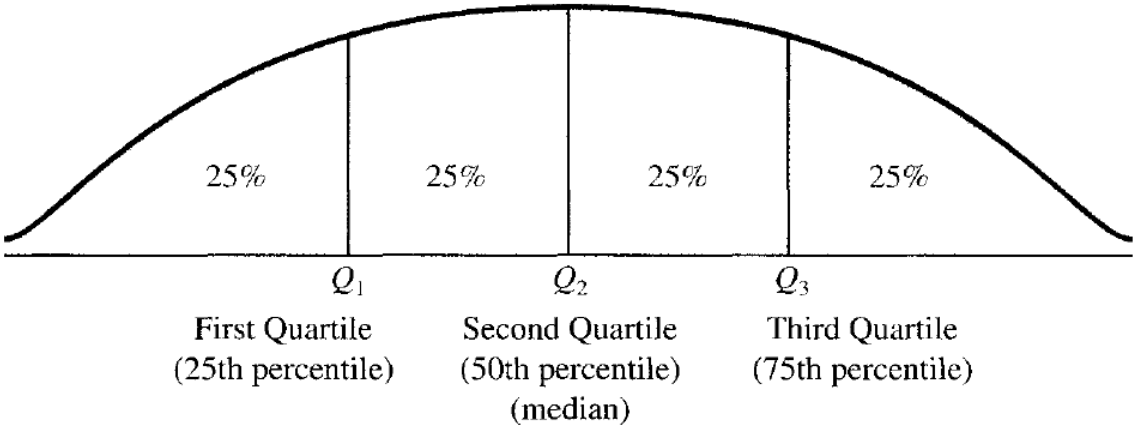
Quantiles, Percentiles and Quartiles

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

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

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

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

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)`

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Measures of Variation

Coefficient of variation

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

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

$C_V = 57\%$

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

Median absolute deviation (MAD)

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

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

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

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

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

In R use:

- `mad (x,...)`

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

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.

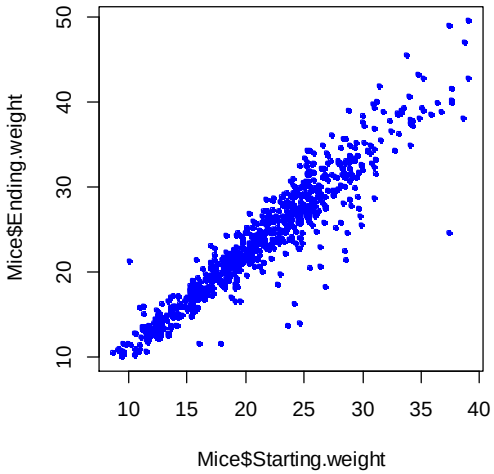
population

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

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,...)`

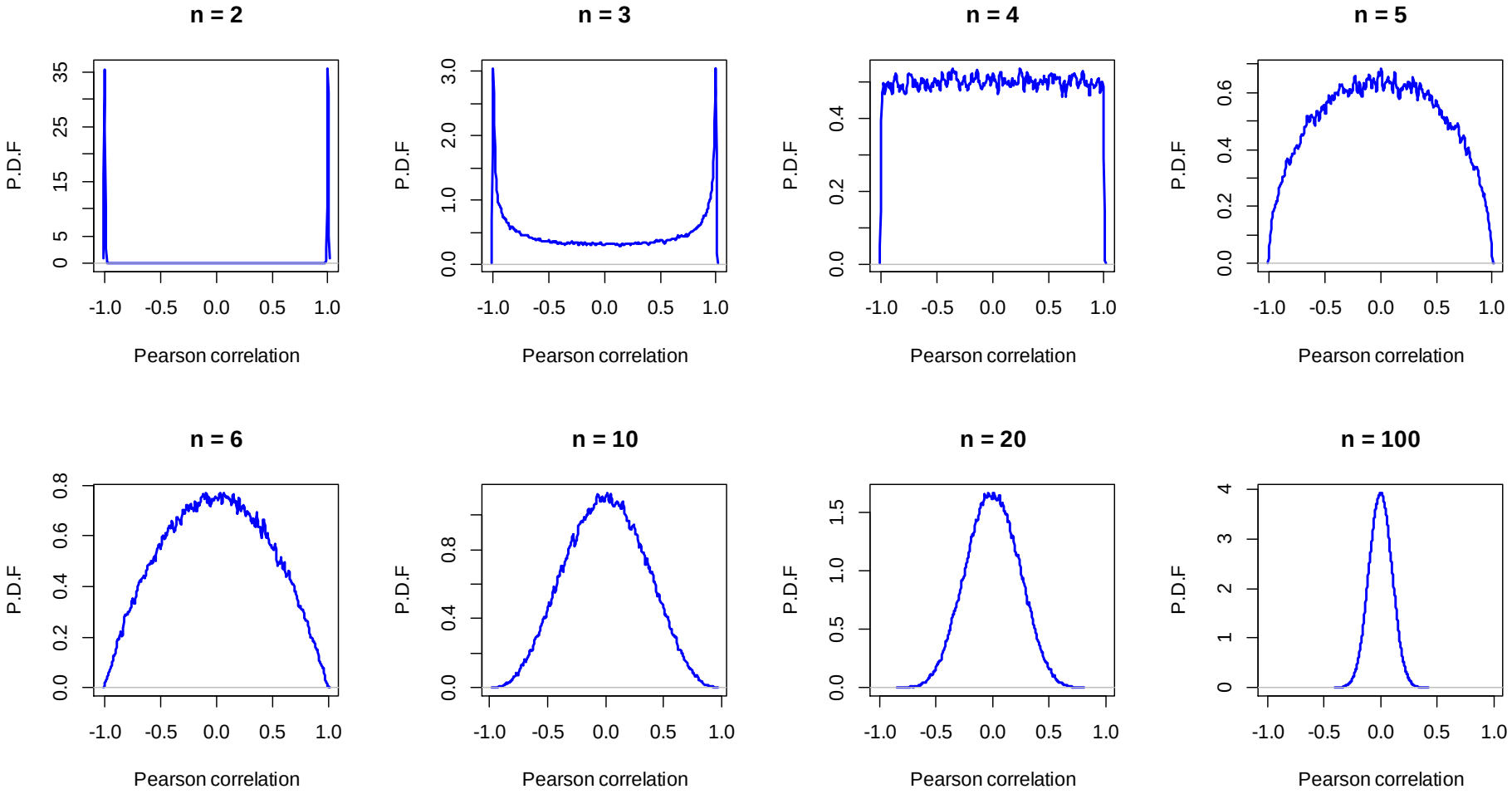
For missing data add parameter
`use = "pairwise.complete.obs"`

In Excel use function:

- `=CORREL(data)`

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Pearson Correlation: Effect of Sample Size



LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

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"`

S6. Descriptive Statistics in R

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

```
#############################################################################
# S6. 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)

##---------------------------------------------------------------------------
## S6.1. 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)

##---------------------------------------------------------------------------
## S6.2. 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)
```

```
##---------------------------------------------------------------------------
## S6.3. 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 S6
#>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Task S6

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Test Hypotheses about a Population Mean

Null Hypothesis: The hypothesis tentatively assumed true in the hypothesis testing procedure, H_0

"There is nothing interesting…"

Alternative hypothesis: The hypothesis concluded to be true if the null hypothesis is rejected, H_a

"Something interesting happens!"

"greater"	"less"	"two.sided"
H_0: $\mu \leq$ const	H_0: $\mu \geq$ const	H_0: $\mu =$ const
H_a: $\mu >$ const	H_a: $\mu <$ const	H_a: $\mu \neq$ const

Parametric:

- `t.test(x, mu =` μ_0, alternative =…)

Non-parametric:

- `wilcox.test(x, mu =` μ_0, alternative =…)

alternative = c("two.sided", "less", "greater")

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Test Hypothesis about a Population Proportion

Proportions

π – population proportion

p – sample proportion

π_0 – testing value (const)

H_0: π = const	H_0: $\pi \geq$ const	H_0: $\pi \leq$ const
H_a: $\pi \neq$ const	H_a: $\pi <$ const	H_a: $\pi >$ const

In R use:

- `prop.test(x, n, p = π0 )`
- `binom.test(x, n, p = π0 )`

Is used with big **n**

Exact test, works always (but is 10 times slower than prop.test)

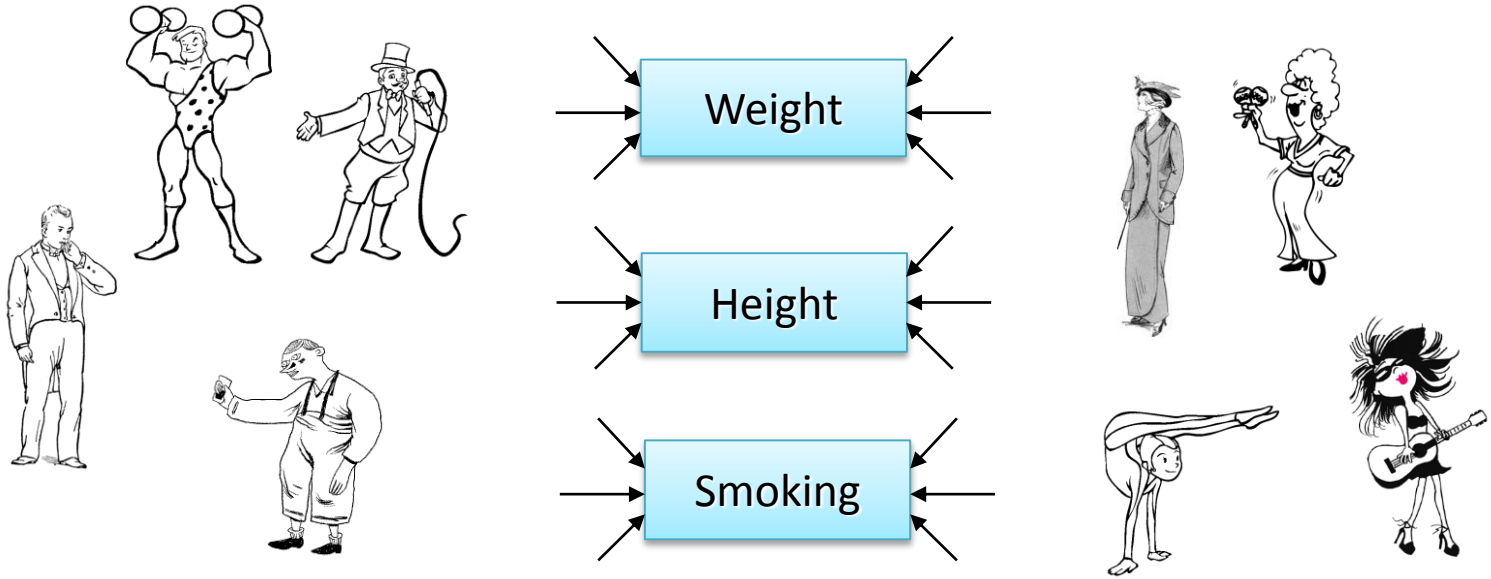
LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Independent Samples

Independent samples

Samples selected from two populations in such a way that the elements making up one sample are chosen independently of the elements making up the other sample.

$H_0: \mu_1 \leq \mu_2$	$H_0: \mu_1 \geq \mu_2$	$H_0: \mu_1 = \mu_2$
$H_a: \mu_1 > \mu_2$	$H_a: \mu_1 < \mu_2$	$H_a: \mu_1 \neq \mu_2$



LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Matched Samples

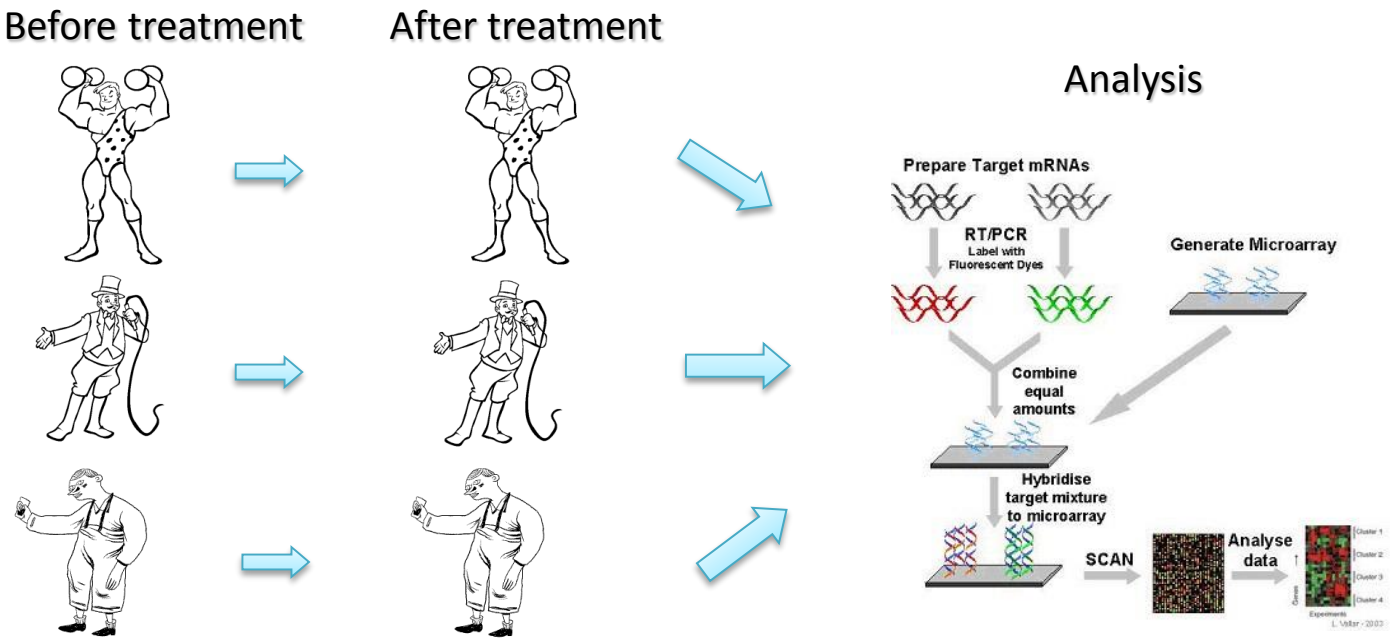
Matched samples

Samples in which each data value of one sample is matched with a corresponding data value of the other sample.

$H_0: \mu_{(x1-x2)} \leq 0$
$H_a: \mu_{(x1-x2)} > 0$

$H_0: \mu_{(x1-x2)} \geq 0$
$H_a: \mu_{(x1-x2)} < 0$

$H_0: \mu_{(x1-x2)} = 0$
$H_a: \mu_{(x1-x2)} \neq 0$



LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Two Sample Tests about Mean / Proportion

Parametric:

- `t.test( x, y, alternative=…)`
- `t.test(x, y, paired = TRUE)`

Non-parametric:

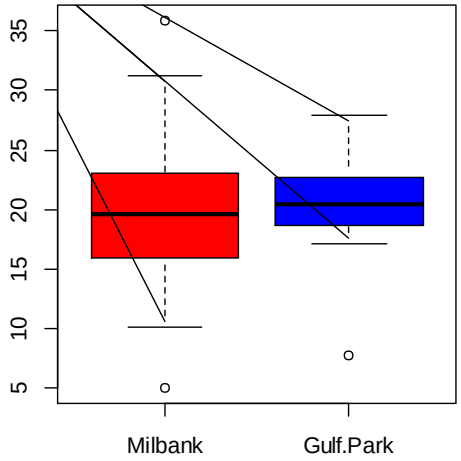
- `wilcox.test ( x, y, alternative=…)`
- `wilcox.test(x, y, paired = TRUE)`

Proportions:

- `prop.test(...)`

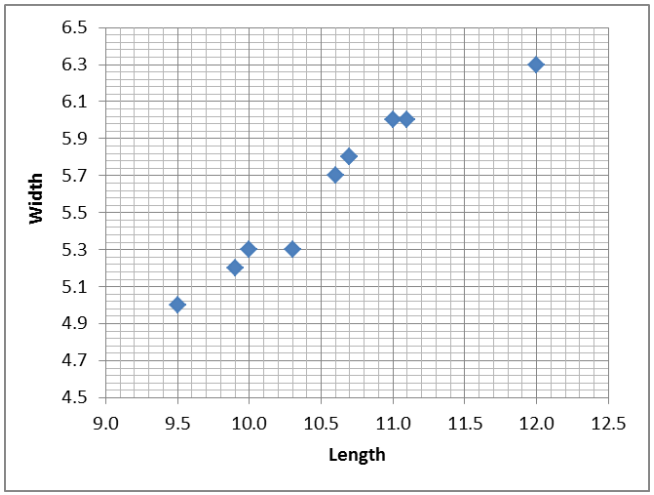
```
# comparing 9/19 (47%) and 15/23 (65%)
prop.test(c(9,15),n=c(19,23))
```

Other Tests



Variance:

- `var.test(x,y)`



Correlation:

- `cor.test (x, y)`

S7. Statistical Tests

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

```
################################################################################
# S7. TESTING HYPOTHESES
################################################################################
## clear memory
rm(list = ls())
##-------------------------------------------------------------------------------
## S7.1. Hypotheses about mean
##-------------------------------------------------------------------------------
##Number of living cells in 5 wells under some conditions are given below.
##In a reference literature source authors claimed a mean quantity of 5000
##living cells under the same conditions. Is our result significantly different?

## put number of cells
x =c(5128,4806,5037,4231,4222)
t.test(x,mu=5000)
##-------------------------------------------------------------------------------
## S7.2. Hypotheses about proportion
##-------------------------------------------------------------------------------
##During a study of a new drug against viral infection, you have found that 70
##out of 100 mice survived, whereas the survival after the standard therapy is
##60% of the infected population. Is this enhancement statistically significant?
p  = 0.7
p0 = 0.6
n = 100

## make analysis by prop.test()
prop.test(x=70,n=100,p=0.6,alternative="greater") ## Approximate!

## make analysis by binom.test()
binom.test(x=70,n=100,p=0.6,alternative="greater")## Exact!

##-------------------------------------------------------------------------------
## S7.3. Unmatched 2 samples
##-------------------------------------------------------------------------------
Mice=read.table("http://edu.sablab.net/data/txt/mice.txt",header=T,sep="\t")
str(Mice)

## Perform t.tests (by default - unpaired, two.sided)
xm = Mice$Ending.weight[Mice$Sex=="m"]
xf = Mice$Ending.weight[Mice$Sex=="f"]
t.test(xm,xf)
wilcox.test(xm,xf)

xm = Mice$Weight.change[Mice$Sex=="m"]
xf = Mice$Weight.change[Mice$Sex=="f"]
t.test(xm,xf)
wilcox.test(xm,xf)

xm = Mice$Bleeding.time[Mice$Sex=="m"]
xf = Mice$Bleeding.time[Mice$Sex=="f"]
t.test(xm,xf)
wilcox.test(xm,xf)
```

```
##-------------------------------------------------------------------------------
## S7.4. Matched samples
##-------------------------------------------------------------------------------

BP=read.table("http://edu.sablab.net/data/txt/bloodpressure.txt",header=T,sep="\t"
str(BP)

## Unpaired
t.test(BP$BP.before,BP$BP.after)
wilcox.test(BP$BP.before,BP$BP.after)

## Paired
t.test(BP$BP.before,BP$BP.after,paired=T)
wilcox.test(BP$BP.before,BP$BP.after,paired=T)

##-------------------------------------------------------------------------------
## S7.5. Proportions
##-------------------------------------------------------------------------------
Mice=read.table("http://edu.sablab.net/data/txt/mice.txt",header=T,sep="\t")
str(Mice)
x = Mice$Sex[Mice$Strain == "SWR/J"]
y = Mice$Sex[Mice$Strain == "MA/MyJ"]

prop.test(c(sum(x=="m"),sum(y=="m")),n=c(length(x),length(y)))

##-------------------------------------------------------------------------------
## S7.6. Variances
##-------------------------------------------------------------------------------
Bus=read.table("http://edu.sablab.net/data/txt/schoolbus.txt",header=T,sep="\t")
str(Bus)
var.test(Bus[,1], Bus[,2])

##-------------------------------------------------------------------------------
## S7.7. Significance of correlation
##-------------------------------------------------------------------------------
Chiton =read.table("http://edu.sablab.net/data/txt/chiton.txt",header=T,sep="\t")
str(Chiton)
cor.test(Chiton$Length,Chiton$Width)
```

Task S7

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Why ANOVA ?

Means for more than 2 populations

We have measurements for 5 conditions. Are the means for these conditions equal?

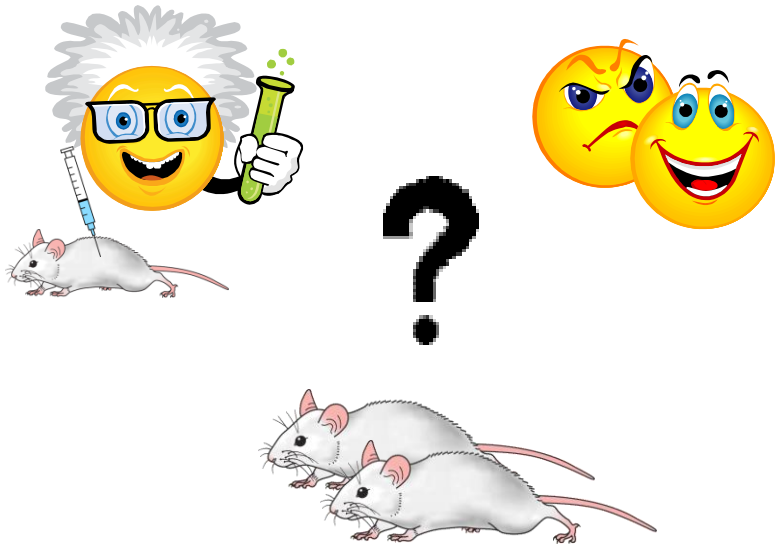
Validation of the effects

We assume that we have several factors affecting our data. Which factors are most significant? Which can be neglected?

If we would use pairwise comparisons, what will be the probability of getting error?

Number of comparisons: $C_2^5 = \frac{5!}{2!3!} = 10$

Probability of an error: $1-(0.95)^{10} = 0.4$



ANOVA
example from Partek™

http://easylink.playstream.com/affymetrix/ambsymposium/partek_08.wvx

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

ANOVA Example (1 way)

As part of a long-term study of individuals 65 years of age or older, sociologists and physicians at the Wentworth Medical Center in upstate New York investigated the relationship between geographic location and depression. A sample of 60 individuals, all in reasonably good health, was selected; 20 individuals were residents of Florida, 20 were residents of New York, and 20 were residents of North Carolina. Each of the individuals sampled was given a standardized test to measure depression. The data collected follow; higher test scores indicate higher levels of depression.

Q: Is the depression level same in all 3 locations?

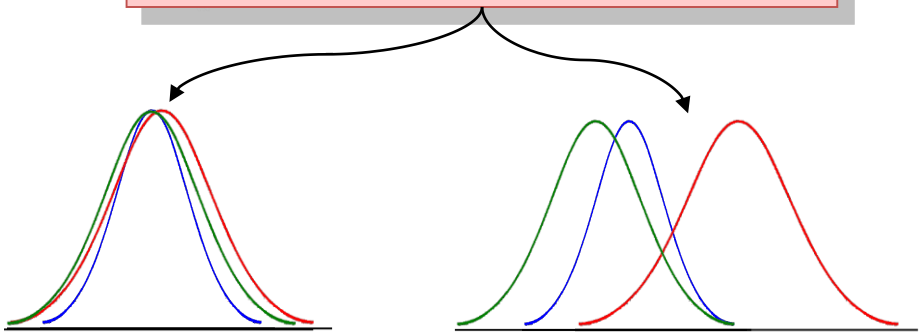
`depression.txt`

1. Good health respondents

Florida	New York	N. Carolina
3	8	10
7	11	7
7	9	3
3	7	5
8	8	11
8	7	8
...	...	...

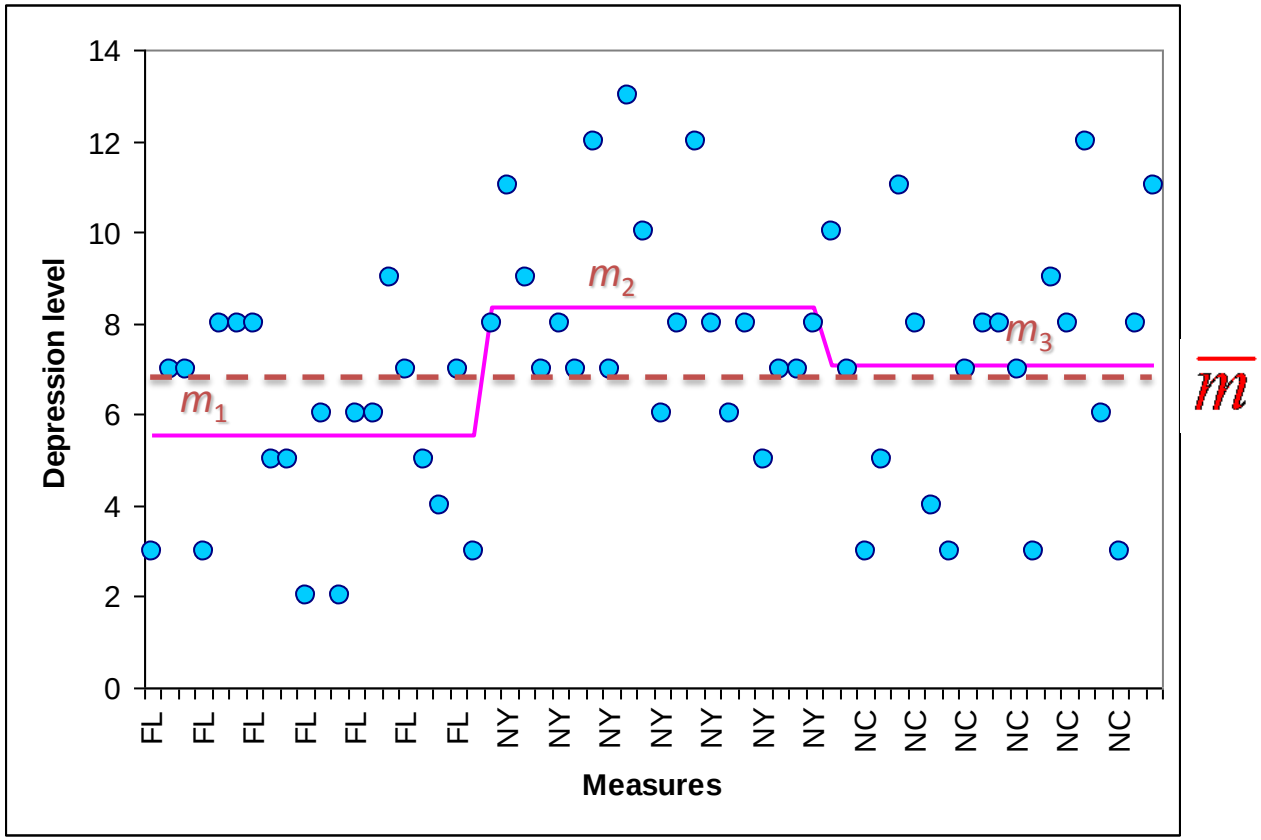
H_0: $\mu_1 = \mu_2 = \mu_3$

H_a: not all 3 means are equal



LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Meaning



Depression = μ + Location + ε

Multi-factor ANOVA

Factor
Another word for the independent variable of interest.

Factorial experiment
An experimental design that allows statistical conclusions about two or more factors.

Treatments
Different levels of a factor.

`depression2.txt`

Factor 1: Health → good health, bad health

Factor 2: Location → Florida, New York, North Carolina

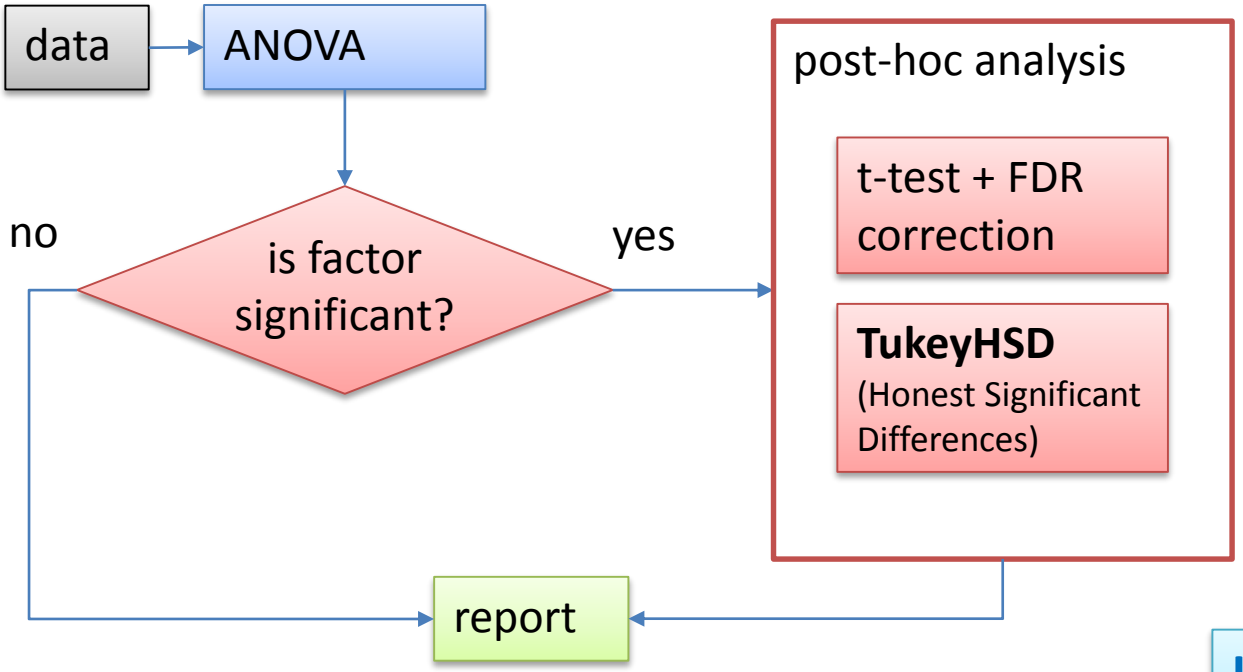
$$\text{Depression} = \mu + \text{Health} + \text{Location} + \underline{\text{Health}\times\text{Location}} + \varepsilon$$

Interaction
The effect produced when the levels of one factor interact with the levels of another factor in influencing the response variable.

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Post-hoc Analysis

Post-hoc analysis
allows for additional exploration of significant differences in the data, when significant effect of the factor was already confirmed (for example, by ANOVA).



In R use:

- `TukeyHSD(...)`

S8. Linear Models

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

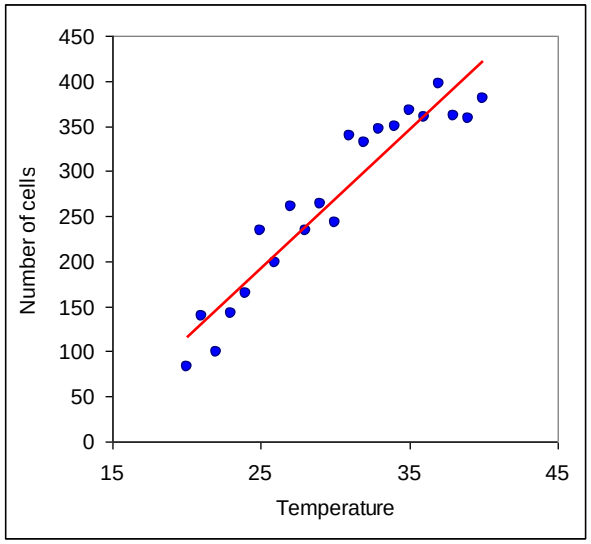
ANOVA

```
#############################################################################
# S8. LINEAR MODELS
#############################################################################
## clear memory
rm(list = ls())

## load data
Data = read.table("http://edu.sablab.net/data/txt/depression2.txt",header=T,sep="\t")
str(Data)

## let's take goo health people
DataGH = Data[Data$Health == "good",]

## build 1-factor ANOVA model
res1 = aov(Depression ~  Location, DataGH)
summary(res1)
#==================================
## build the 2 factor ANOVA model
res2 = aov(Depression ~  Location + Health + Location*Health, data = Data)

## show the ANOVA table
summary(res2)

## build one more model - without interaction
res2ni = aov(Depression ~  Location + Health, data = Data)

## get p-values
pv=(summary(res2)[[1]][,5])
names(pv) = rownames(summary(res2)[[1]])
barplot(-log(pv),ylab="-log10(p-value)")

## Load function
source("http://sablab.net/scripts/drawANOVA.r")
x11()
drawANOVA(res2)

#####################
## Post-hoc analysis

TukeyHSD(res1)
TukeyHSD(res2)
TukeyHSD(res2ni)
```

Regression Model and Regression Line

Regression model
The equation describing how y is related to x and an error term; in simple linear regression, the regression model is $y = \beta_0 + \beta_1 x + \varepsilon$

Regression equation
The equation that describes how the mean or expected value of the dependent variable is related to the independent variable; in simple linear regression,
$E(y) = \beta_0 + \beta_1 x$



- Model for a simple linear regression:

$$y(x) = \beta_1 x + \beta_0 + \varepsilon$$

- As we never know β - we estimate it with b_0, b_1, which are going to change from one experimental run to another

$$\hat{y} = b_1 x + b_0$$

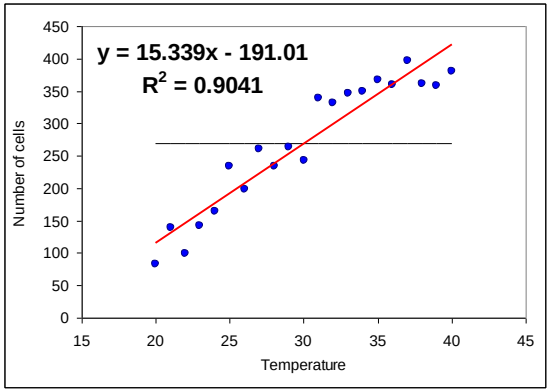
Coefficient of Determination

$$SSE = \sum (y_i - \hat{y}_i)^2$$

$$SST = \sum (y_i - m_y)^2$$

$$SSR = \sum (\hat{y}_i - m_y)^2$$

$$SST = SSR + SSE$$



Coefficient of determination

A measure of the goodness of fit of the estimated regression equation. It can be interpreted as the proportion of the variability in the dependent variable *y* that is explained by the estimated regression equation.

$$R^2 = \frac{SSR}{SST}$$

Correlation coefficient

A measure of the strength of the linear relationship between two variables (previously discussed in Lecture 1).

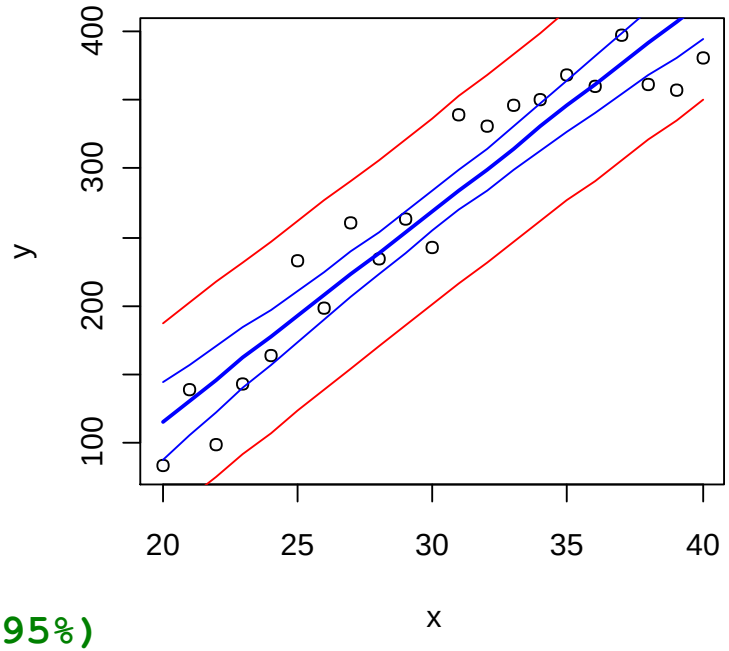
$$r = \text{sign}(b_1)\sqrt{R^2}$$

Linear Regression

```
Data = read.table("http://edu.sablab.net/data/txt/cells.txt", header=T,
                  sep="\t",quote="\"")
```

cells.txt

```
x = data$Temperature
y = data$Cell.Number
res = lm(y~x)
res
summary(res)

# draw the data
x11()
plot(x,y)
# draw the regression and its confidence (95%)
lines(x, predict(res,int = "confidence")[,1],col=4,lwd=2)
lines(x, predict(res,int = "confidence")[,2],col=4)
lines(x, predict(res,int = "confidence")[,3],col=4)
# draw the prediction for the values (95%)
lines(x, predict(res,int = "pred")[,2],col=2)
lines(x, predict(res,int = "pred")[,3],col=2)
```



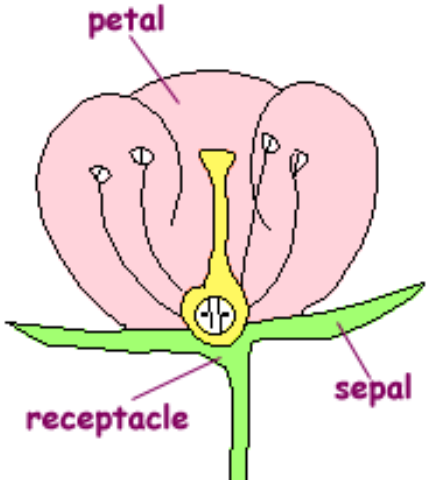
S9. PCA and Clustering

- **Principal component analysis (S9.1)**
  - Iris dataset
  - General introduction to PCA
  - PCA for data exploratory analysis
- **Clustering (S9.2)**
  - k-means clustering
  - Hierarchical clustering
  - ALL dataset

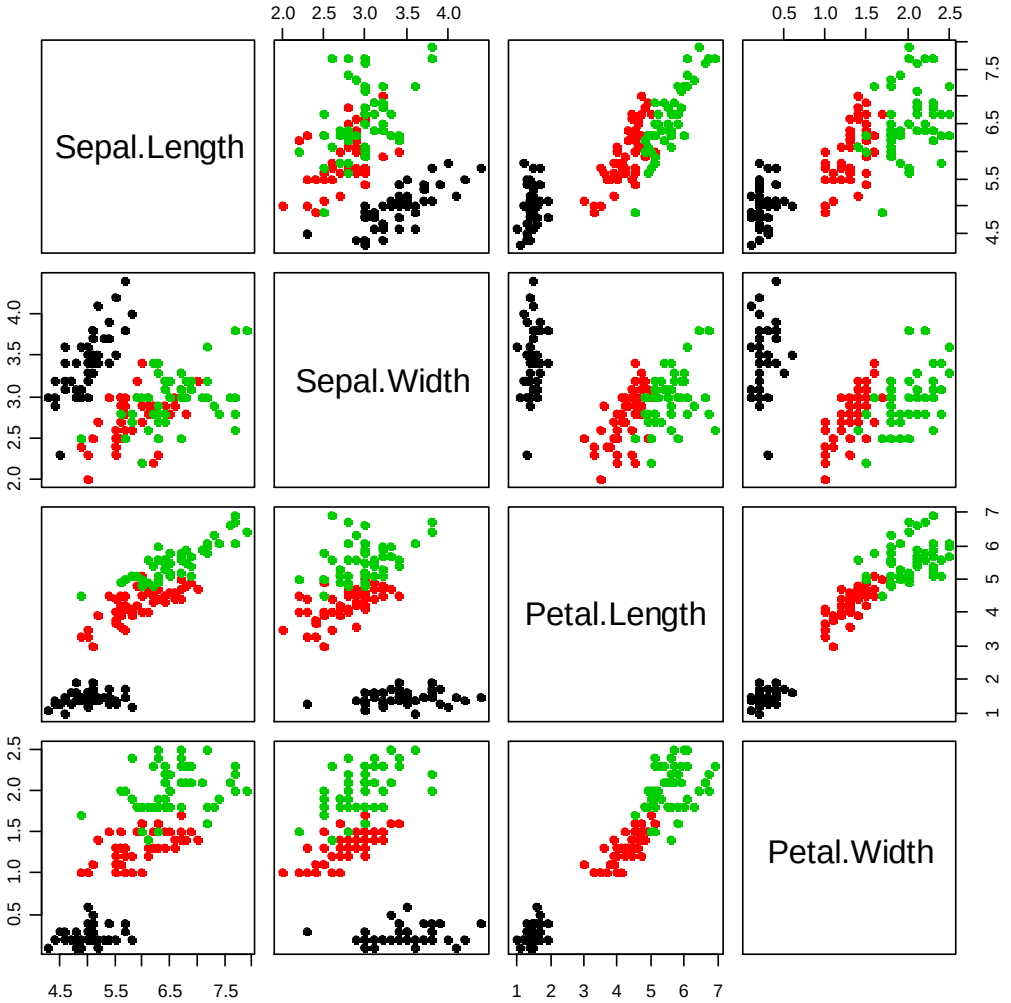
S9.1. Principal Component Analysis

Iris Dataset Presentation

```
iris
str(iris)
## plot iris data
x11()
plot(iris[,-5])
## more beautiful
plot(iris[,-5],
     col = iris[,5],pch=19)
```



http://urbanext.illinois.edu/gpe/case4/c4facts1a.html



How could we possibly represent these data on a single plot?

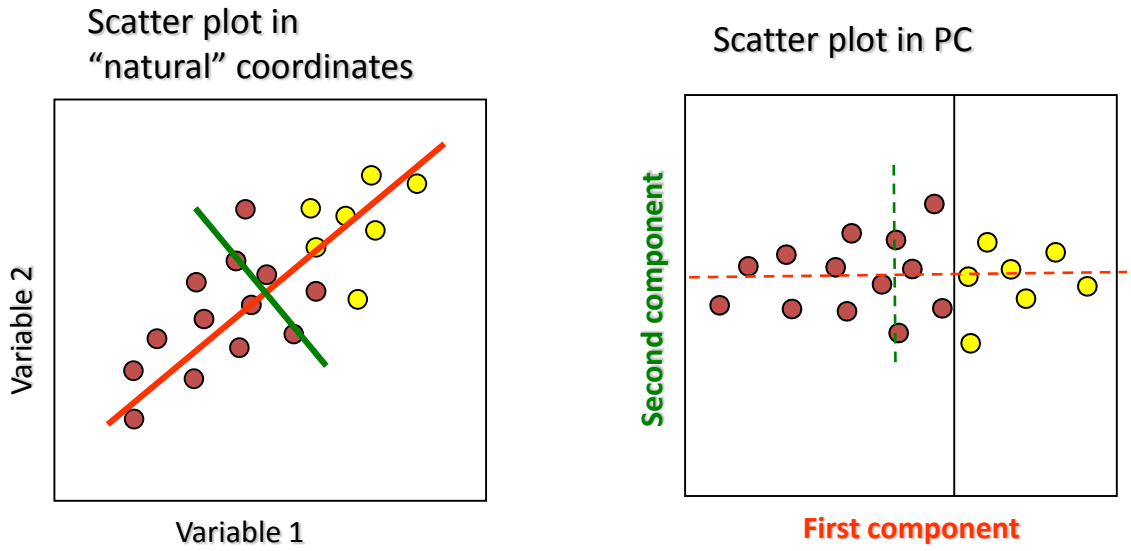
LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Principal Component Analysis (PCA)

Principal component analysis (PCA)
is a vector space transform used to reduce multidimensional data sets to lower dimensions for analysis. It selects the **coordinates along which the variation of the data is bigger**.

20000 genes → 2 dimensions

For the simplicity let us consider 2 parametric situation both in terms of data and resulting PCA.



Instead of using 2 "natural" parameters for the classification, we can use the first component!

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

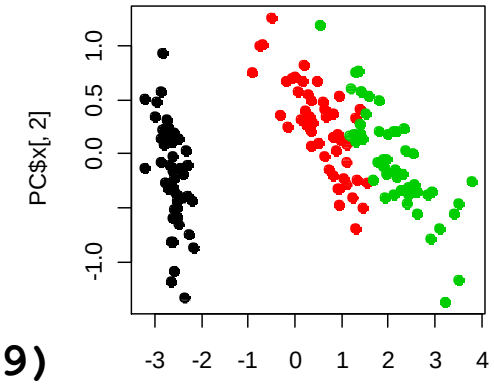
PCA for Iris Dataset

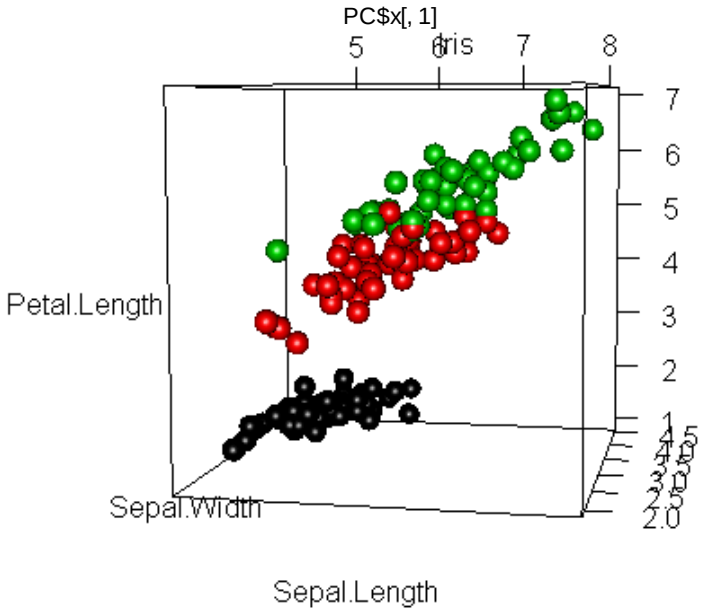
```
Data = as.matrix(iris[,-5])
row.names(Data) = as.character(iris[,5])
classes = as.integer(iris[,5])

## perform PCA
PC = prcomp(Data)
str(PC)
## plot PC1 and PC2 only
plot(PC$x[,1],PC$x[,2], col=classes, pch=19)
## plot 3D
library(rgl)
plot3d(PC$x[,1],PC$x[,2],PC$x[,3],
       size = 2,
       col  = classes,
       type = "s",
       xlab = "PC1",
       ylab = "PC2",
       zlab = "PC3")
```

Iris setosa (1)

Iris versicolor (2)

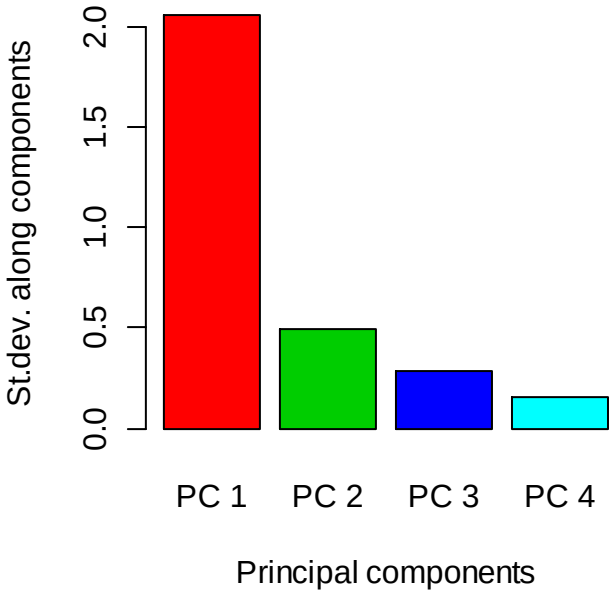
Iris virginica (3)

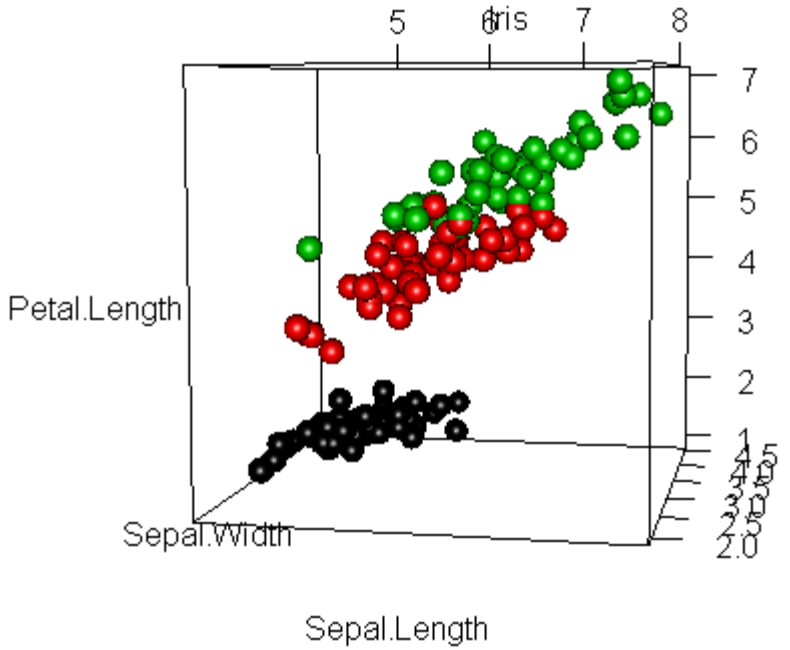




PCA for Iris Dataset

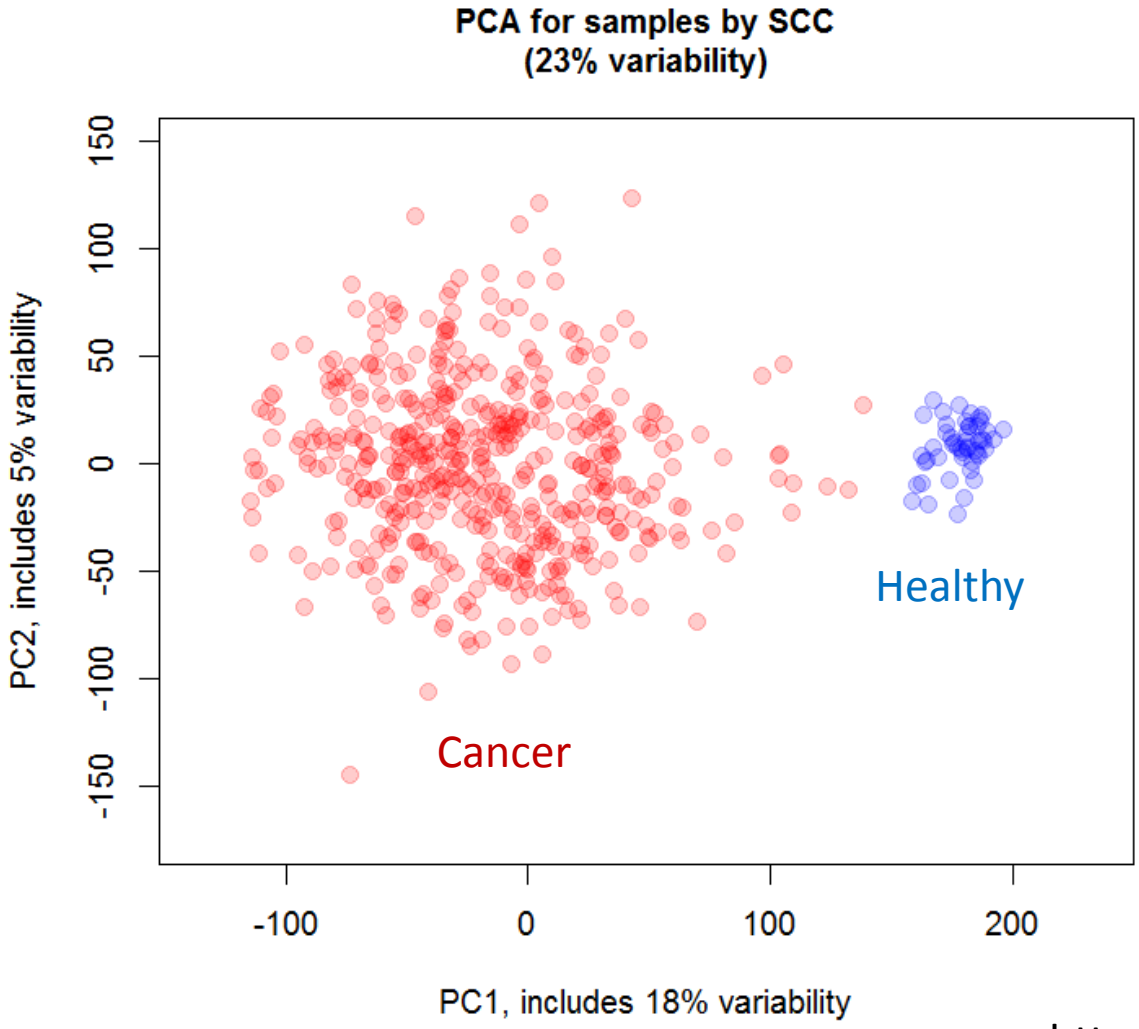
```
barplot(PC$sdev,
        names.arg = paste("PC",1:length(PC$sdev)),
        col   = (1:length(PC$sdev))+1,
        xlab = "Principal components",
        ylab = "St.dev. along components")
```





Exploratory Data Analysis

PCA in TCGA (LUSC data)



http://edu.sablab.net/data/gz/lusc.zip

S9.1. Principal Component Analysis

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

```
###############################################################################
# S9.1. PCA
###############################################################################
## clear memory
rm(list = ls())

##-----------------------------------------------------------------------------
## S9.1.1. Iris data
##-----------------------------------------------------------------------------

## show the data
iris
str(iris)

## plot iris data
x11()
plot(iris[,-5])
## more beautiful
plot(iris[,-5],col = iris[,5],pch=19)

##-----------------------------------------------------------------------------
## S9.1.2 PCA
##-----------------------------------------------------------------------------
## Let's transform data frame into numerical matrix
## data - numerical data
## classes - type of iris, 1=setosa, 2=versicolor, 3=virginica
Data = as.matrix(iris[,-5])
row.names(Data) = as.character(iris[,5])
classes = as.integer(iris[,5])

## perform PCA
PC = prcomp(Data)
str(PC)

## plot PC1 and PC2 only
plot(PC$x[,1],PC$x[,2],
     col=classes, pch=19)

## plot 3D
library(rgl)
plot3d(PC$x[,1],PC$x[,2],PC$x[,3],
       size = 2,
       col  = classes,
       type = "s",
       xlab = "PC1",
       ylab = "PC2",
       zlab = "PC3")
```

```
##-----------------------------------------------------------------------------
## S9.1.2 PCA for ALL
##-----------------------------------------------------------------------------

ALL = read.table("http://edu.sablab.net/data/txt/all_data.txt",
                 as.is=T,header=T,sep="\t")
## Transform to matrix
Data=as.matrix(ALL[,-(1:2)])
## assign colors based on column names
color = colnames(Data)
color[grep("ALL",color)]="red"
color[grep("normal",color)]="blue"
## perform PCA
PC = prcomp(t(Data))
## visualize in 3D
library(rgl)
plot3d(PC$x[,1],PC$x[,2],PC$x[,3],size = 2,col  = color,type = "s",
       xlab = "PC1",ylab = "PC2",zlab = "PC3")
## plot PC1 and PC2 only
plot(PC$x[,1],PC$x[,2],col=color,pch=19,cex=2)
text(PC$x[,1],PC$x[,2]+5,colnames(Data),cex=0.6)

## check the distribution of the data
source("http://sablab.net/scripts/plotDataPDF.r")
x11()
plotDataPDF(Data,col=color,add.legend=T)
x11()
boxplot(Data,col=color,outline=F,las=2)
```

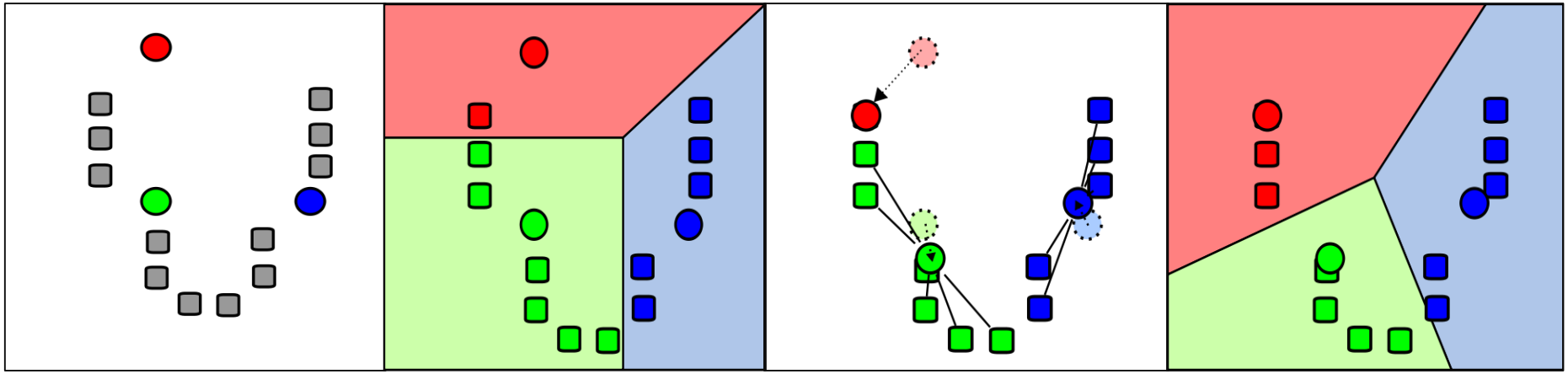
Task S9.1

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

k-Means Clustering

k-Means Clustering

k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.



1) *k* initial "means" (in this case k=3) are randomly selected from the data set (shown in color).

2) *k* clusters are created by associating every observation with the nearest mean.

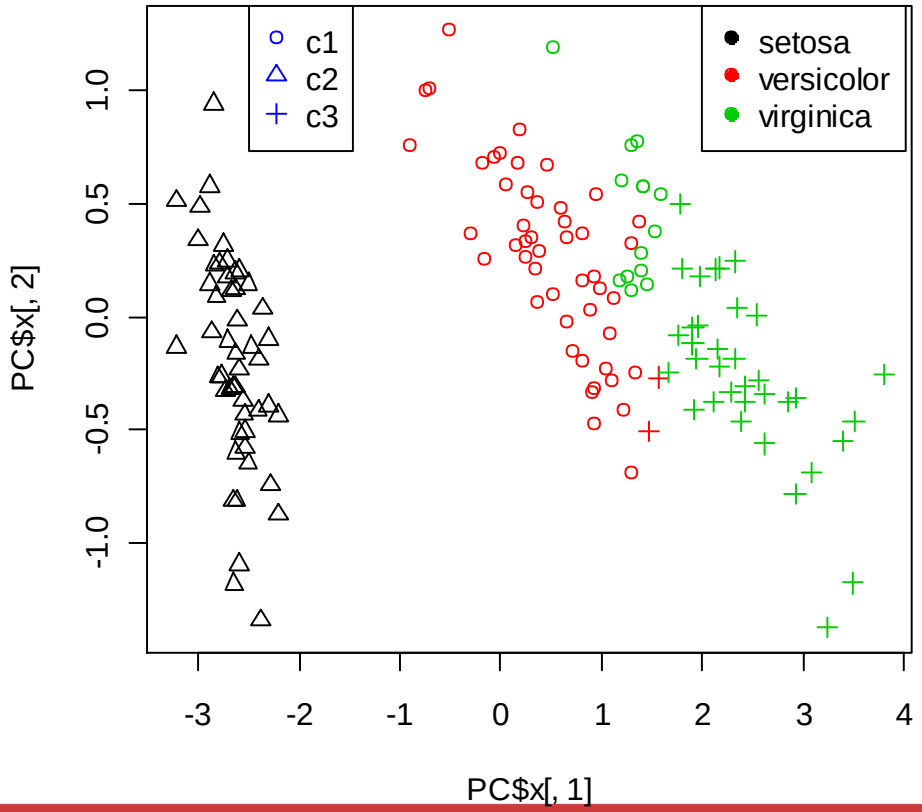
3) *The centroid of each of the k clusters becomes the new means.*

4) Steps 2 and 3 are repeated until convergence has been reached.

http://wikipedia.org

k-Means Clustering: Iris Dataset

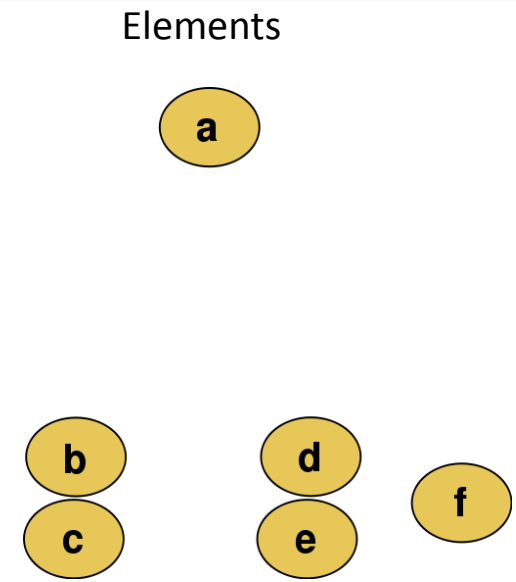
```
clusters = kmeans(x=Data,centers=3,nstart=10)$cluster
plot(PC$x[,1],PC$x[,2],col = classes,pch=clusters)
legend(2,1.4,levels(iris$Species),col=c(1,2,3),pch=19)
legend(-2.5,1.4,c("c1","c2","c3"),col=4,pch=c(1,2,3))
```

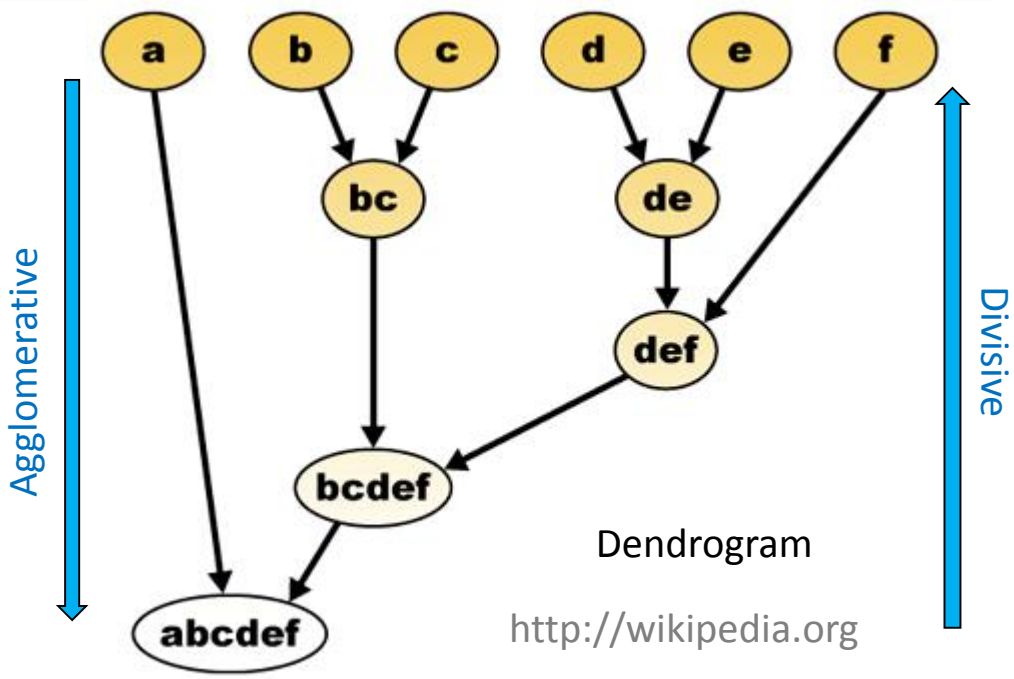


Hierarchical Clustering

Hierarchical Clustering

Hierarchical clustering creates a hierarchy of clusters which may be represented in a tree structure called a dendrogram. The root of the tree consists of a single cluster containing all observations, and the leaves correspond to individual observations.
Algorithms for hierarchical clustering are generally either agglomerative, in which one starts at the leaves and successively merges clusters together; or divisive, in which one starts at the root and recursively splits the clusters.



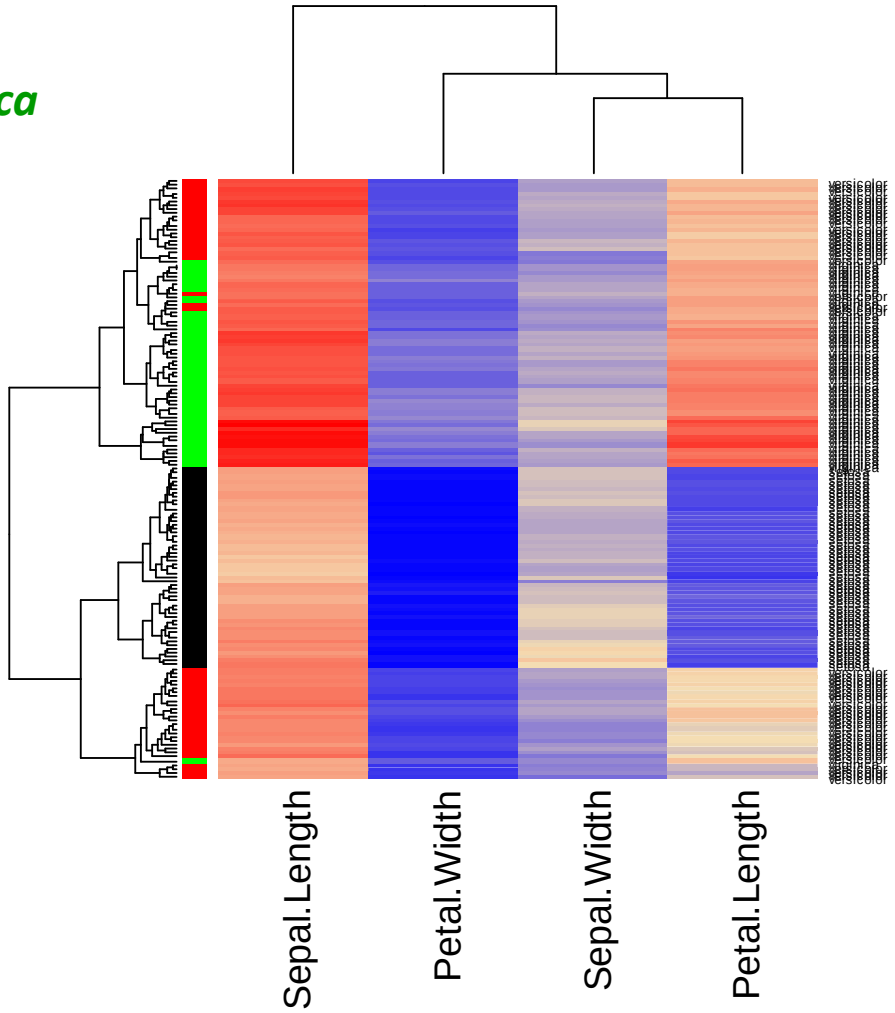


Hierarchical Clustering: Iris Dataset

Iris setosa

Iris versicolor

Iris virginica




```
## simple: use heatmap
heatmap(Data)

## use heatmap with colors
color = character(length(classes))
color[classes == 1] = "black"
color[classes == 2] = "red"
color[classes == 3] = "green"
heatmap(Data,RowSideColors=color,
        scale="none")

## modify the heatmap colors
heatmap(Data,RowSideColors=color,
    col = colorRampPalette (c("blue","wheat","red"))(1000))
```

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

Task: Clustering in ALL Dataset

1. Select the top 100 genes with the most significant variation of expression b/w normal and ALL samples. Use **`t.test(...)$p.val`** to address statistical significance.

2. Build a hierarchical clustering of the selected 100 genes and samples

```
ALL = read.table("http://edu.sablab.net/data/txt/all_data.txt",
                 as.is=T,header=T,sep="\t")
Data=as.matrix(ALL[,-(1:2)])
color = colnames(Data)
color[grep("ALL",color)]="red"
color[grep("normal",color)]="blue"
## annotate genes
rownames(Data) = ALL[,1]
## create indexes for normal and ALL columns
idx.norm = grep("normal",colnames(Data))
idx.all = grep("ALL",colnames(Data))
## perform a t-test
pv = double(nrow(Data))  ## p-value storage
for (i in 1:nrow(Data)){
  pv[i] = t.test(Data[i,idx.all],Data[i,idx.norm])$p.val
}
## select top genes
Top = Data[order(pv),][1:100,]
```

S9.2. Clustering

```
#############################################################################
# S9.2. Clustering
#############################################################################
## clear memory
rm(list = ls())

##---------------------------------------------------------------------------
## S9.2.1 k-means Clustering
##---------------------------------------------------------------------------
Data = as.matrix(iris[,-5])
row.names(Data) = as.character(iris[,5])
classes = as.integer(iris[,5])

## try k-means clustering
clusters = kmeans(x=Data,centers=3,nstart=10)$cluster

## show clusters on PCA
PC = prcomp(Data)
x11()
plot(PC$x[,1],PC$x[,2],col = classes,pch=clusters)
legend(2,1.4,levels(iris$Species),col=c(1,2,3),pch=19)
legend(-2.5,1.4,c("c1","c2","c3"),col=4,pch=c(1,2,3))

##---------------------------------------------------------------------------
## S9.2.2 Hierarchical clustering
##---------------------------------------------------------------------------

## use heatmap
heatmap(Data)

## use heatmap with colors
color = character(length(classes))
color[classes == 1] = "black"
color[classes == 2] = "red"
color[classes == 3] = "green"
heatmap(Data,RowSideColors=color,scale="none")

## modify the heatmap colors
heatmap(Data,RowSideColors=color,scale="none",
   col = colorRampPalette (c("blue","wheat","red"))(1000))

## For advanced heatmap use:
library(gplots)
heatmap.2(Data,RowSideColors=color,scale="none",trace="none",
          col = colorRampPalette (c("blue","wheat","red"))(1000))
```

```
##---------------------------------------------------------------------------
## S9.2.2 Hierarchical clustering: ALL (Task S9.2)
##---------------------------------------------------------------------------
ALL = read.table("http://edu.sablab.net/data/txt/all_data.txt",
                 as.is=T,header=T,sep="\t")
Data=as.matrix(ALL[,-(1:2)])
color = colnames(Data)
color[grep("ALL",color)]="red"
color[grep("normal",color)]="blue"
##---------------------------------
## Task S9.2a. Select top 100 genes

## annotate genes
rownames(Data) = ALL[,1]
## create indexes for normal and ALL columns
idx.norm = grep("normal",colnames(Data))
idx.all = grep("ALL",colnames(Data))

## perform a t-test
pv = double(nrow(Data))
for (i in 1:nrow(Data)){
  pv[i] = t.test(Data[i,idx.all],Data[i,idx.norm])$p.val
}
## select top genes
Top = Data[order(pv),][1:100,]

## make a heatmap
heatmap(Top,ColSideColors=color,
          col = colorRampPalette (c("blue","wheat","red"))(1000))

## scale data first and repeat heatmap
TopSc = t(scale(t(Top)))
heatmap(TopSc,ColSideColors=color,
          col = colorRampPalette (c("blue","white","red"))(1000))


##---------------------------------
## In fact, robust set of signidicant genes, identified by limma
## package of R/Bioconductor is different:
x11()
source("http://sablab.net/scripts/limmaEBS2Class.r")
res = limmaEBS2Class(data = ALL[,-c(1:2)],
                     anno =  ALL[,c(1:2)],
                     classes = sub("_.+","",names(ALL[,-c(1:2)])),
                     plotTop=100,
                     col=c("red","blue"))
```

Task S9.2