


```

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==1
r<1

##-----
## Character (strings)
st = "Hello, world!"
st
paste("We say:",st) # concatenation
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)

##-----
## 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))

##-----
## 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 4b, 4c, 4d
#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

##-----
## 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 manually
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

##-----
## 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()
```