UNIVERSITÄT LEIPZIG

European Research Council

Introduction to R

Natalia Levshina © 2019

Tallinn University
May 14-18 2019

Outline

1. Introduction to R
2. Basics of R syntax
3. Main objects in R
4. Creating and importing your data into R

What is R?

- statistical computing environment (from *t*-test to generalized linear models, and more…)
    - core distribution “base”
    - add-on packages
- programming language
- tools for creation of publication-quality plots (e.g. ggplot2)

Where to get R?

- Distribution and packages: CRAN (Comprehensive R Archive Network) http://cran.r-project.org/
- Information: http://www.r-project.org/

RStudio

- Highly recommended (easy to manage projects, packages, data, graphs, etc.)!
- Available from http://www.rstudio.com/products/RStudio/

Outline

1. Introduction to R
2. Basics of R syntax
3. Main objects in R
4. Creating and importing your data into R

Input and output

```
> 2 + 2
[1] 4

> sample(100, 25) #random sampling of 25 elements
from integers 1 to 100
[1] 49 45 70 51 54  5  7 19 60 82 35 55  6 76 93
89 44
[18]  8 48 87 53 34 86 96 63
```

Basic arithmetic functions

```
> 25^2
[1] 625
> 625^0.5
[1] 25
> sqrt(625)
[1] 25
> log(5)
[1] 1.609438
```

Creation of objects

```
> a <- 3
> a
[1] 3
> a + 5
[1] 8
```

Exercise

- Create two vectors with 1 element in each:
  a) the population of Tallinn
  b) the population of Helsinki
- Compute their sum.
- Compute their difference.
- By how many times is the population of Tallinn larger than that of Helsinki?

Beware: = and ==

```
> a = 3 # creates an object a with the value 3, an alternative to "a <- 3"
> a == 3 # tests if a equals 3
[1] TRUE
> a == 10 # tests if a equals 10
[1] FALSE
```

Exercise

- Make R test whether the population of Tallinn is equal to that of Helsinki, using the vectors.

R is case-sensitive!

```
> b <- 7
> a + b
[1] 10
> a + B
Error: object 'B' not found
```

Managing your objects

```
> ls() #returns a list of objects
 [1] "a"         "b"

> rm(b) #removes an object
> ls()
 [1] "a"
```

Saving your workspace

1. Click on the cross or type

```
> q()
```

Select the action (to save or not to save).

```
> getwd() #to find out where your workspace will be saved
[1] "C:/Users/Your/Directory"
> setwd("C:/Users/Your/Directory") #to change it, if you like
```

2. Next session: restart R or, if you have many different workspaces, click on the R from the directory; alternatively:

```
> load("yourDirectory/yourFile.RData")
```

Getting help

```
> ?cor #to open a help file with information about function 'cor'

> ??correlation #returns a list of functions that contain this expression
```

Exercise

- Get help on the function summary().

Errors

```
> x <- 1:10 # creates a numeric vector with integers from 1 to 10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> meann(x) # we want to compute the mean value of x: a typo
Error: could not find function "meann"
> mean(x) # correct
[1] 5.5
```

Warning messages

```
> mytable <- rbind(c(1, 2), c(3, 4)) #create a 2-by-2 table
> mytable
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> chisq.test(mytable)

        Pearson's Chi-squared test with Yates' continuity correction

data:  mytable
X-squared = 0, df = 1, p-value = 1

Warning message:
In chisq.test(mytable) : Chi-squared approximation may be incorrect
```

A quest

Complete these operations and tell me the sentence.

1. Open the help page of the function aggregate(). Go to the Description section. Find the 8^{th} word.

2. Which function in R helps you to identify the class of an object?

3. Take the 3^{rd} argument of the function apply().

Outline

1. Introduction to R
2. Basics of R syntax
3. Main objects in R
4. Creating and importing your data into R

Important data types in R

- Numeric vectors
- Character vectors
- Factors
- Data frames
- Contingency tables
- Matrices
- Distance matrices

Numeric vectors

```
> vnum <- 1:5 # a vector of integers from 1 to 5
> vnum
[1] 1 2 3 4 5
> is(vnum)
 [1] "integer"                    "numeric"
"vector"
```

[….]

If not a sequence:

```
> RT <- c(455, 773, 512, 667) #reaction times in an
experiment
> RT
[1] 455 773 512 667
```

Character vectors

```
> sex <- c("f", "m", "m", "f")
> sex
[1] "f" "m" "m" "f"
> is(sex)
[1] "character"              "vector"
[…]
```

Factors

```
> sex.f <- factor(sex)
> sex.f
[1] f m m f
Levels: f m

> is(sex.f)
 [1] "factor"                    "integer"
[…]
```

Data frames

```
> mydf <- data.frame(sex, RT) #char. vectors turn
into factors
> mydf
     sex    RT
1    f              455
2    m              773
3    m              512
4    f              667
> is(mydf)
[1] "data.frame" "list" […]
```

Exercise

1. Create a character vector with the names of your fellow students.

2. Create a vector with the number of languages they can speak.

3. Combine the vectors in one data frame.

Summarizing the data

```
> summary(mydf)
 sex          RT           dialect
 f:2   Min.   :455.0   Length:4
 m:2   1st Qu.:497.8   Class :character
       Median :589.5   Mode  :character
       Mean   :601.8
       3rd Qu.:693.5
       Max.   :773.0
> str(mydf)
'data.frame': 4 obs. of  3 variables:
 $ sex    : Factor w/ 2 levels "f","m": 1 2 2 1
 $ RT     : num  455 773 512 667
 $ dialect: chr  "BrE" "AmE" "AmE" "BrE"
```

Selecting observations

```
> mydf[1,]
  sex  rt   dialect      #the fist row
1   f 455   BrE

> mydf[,2]
[1] 455 773 512 667 #the second column

> mydf[1,2]
[1] 455 #the element in the fist row,  second
column
```

Using logical operators

```
> mydf[mydf$sex == "f",]
  sex  RT   dialect
1   f 455   BrE
4   f 667   BrE
> mydf[mydf$sex != "m", ]
  sex  RT   dialect
1   f 455   BrE
4   f 667   BrE
> mydf[mydf$RT < 500,]
  sex  RT   dialect
1   f 455   BrE
```

Exercise

- Make a subset of your data frame with all students who can speak more than 2 languages.
- How many rows (students) does the data frame contain?

Contingency tables

- Let's add another factor to the dataframe, *dialect*:

```
> mydf$dialect <- c("BrE", "AmE", "AmE", "BrE")
> mydf
  sex          RT     dialect
1   f  455            BrE
2   m  773            AmE
3   m  512            AmE
4   f  667            BrE
> table(mydf$sex, mydf$dialect)

     AmE BrE
  f    0      2
  m    2      0
```

Exercise

1. Do you think there's a relationship between the preference for dogs or cats and introversion or extraversion?

2. Add a factor to your data frame with your colleagues' psychological types ("intro", "extra" or something like that).

3. Add another factor with the information whether one prefers dogs or cats("dogs" or "cats", or …)

4. Cross-tabulate the factors and interpret the numbers.

Matrices

```
> m <- cbind(1:5, 10:6)
> m
     [,1] [,2]
[1,]    1   10
[2,]    2    9
[3,]    3    8
[4,]    4    7
[5,]    5    6
> is(m)
[1] "matrix"    "array"   […]
```

Distance matrices

```
> eurodist
```

[output omitted: distances between several European cities]

My journey this summer:

	Frankfurt	Stockholm	Tampere
Frankfurt	0	1186	1572
Stockholm	1186	0	395
Tampere	1572	395	0

My journey this summer

```
> mydist <- rbind(Frankfurt = c(0, 1186, 1572), Stockholm = c(1186, 0, 395), Tampere = c(1572, 395, 0))
> colnames(mydist) <- rownames(mydist)
> mydist
           Frankfurt Stockholm Tampere
Frankfurt          0      1186    1572
Stockholm       1186         0     395
Tampere         1572       395       0
> is(mydist)
[1] "matrix"    "array"     "mMatrix"
"structure" "vector"
```

From matrix to distance matrix

```
> mydist <- as.dist(mydist)
> mydist
          Frankfurt Stockholm
Stockholm      1186
Tampere        1572       395
> is(mydist)
[1] "dist"
```

… and back

```
> m <- as.matrix(mydist)
> m
           Frankfurt Stockholm Tampere
Frankfurt          0      1186    1572
Stockholm       1186         0     395
Tampere         1572       395       0
```

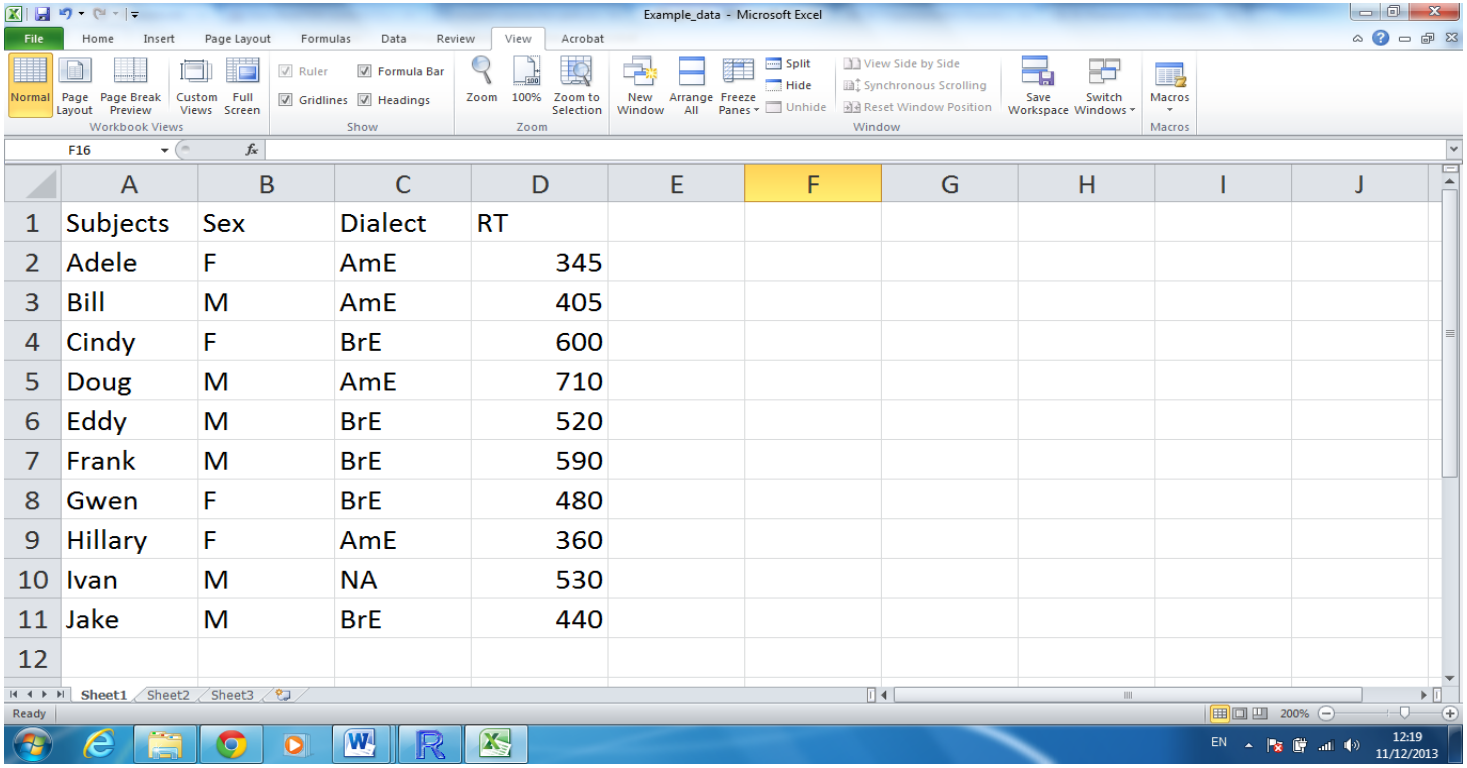
Exercise

- Make your own distance matrix with your own travel destinations.

Outline

1. Introduction to R
2. Basics of R syntax
3. Main objects in R
4. Creating and importing your data into R

Importing your data to R



	A	B	C	D	E	F	G	H	I	J
1	Subjects	Sex	Dialect	RT						
2	Adele	F	AmE	345						
3	Bill	M	AmE	405						
4	Cindy	F	BrE	600						
5	Doug	M	AmE	710						
6	Eddy	M	BrE	520						
7	Frank	M	BrE	590						
8	Gwen	F	BrE	480						
9	Hillary	F	AmE	360						
10	Ivan	M	NA	530						
11	Jake	M	BrE	440						
12										

Importing your data into R

1. Create a similar table in Excel (or OpenOffice Calc). Don't forget to create a header. In case of missing values, put NA. No empty cells!

2. Save the file as a tab delimited text file (.txt).

3. Read the file in R:

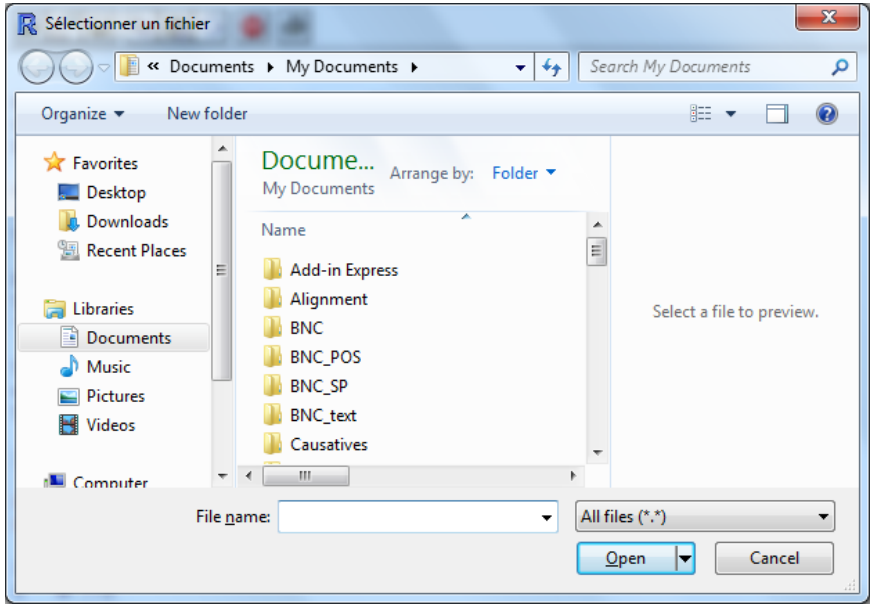
```
> mydata <- read.table(file = file.choose(), header = TRUE)
```

To be on the safe side:

```
> mydata <- read.table(file = file.choose(), header
= TRUE, sep = "\t", quote = "", comment = "")
```

The argument sep = "\t" means that the tab separates the columns. The argument quote = "" tells R to treat quotation marks as text, which may be useful for some corpus data. The argument comment = "" means that # is also treated as text, not as a special symbol that introduces some comments.

Interactive choice



Exercise

1. Go to WALS (wals.info)

2. Choose your favorite linguistic feature or language.

3. Copy and paste the data into a spreadsheet application or Notepad, edit if necessary, save as a tab-separated file.

4. Open it in R as a data frame.

Alternatively, do the same with your own data.