Part 1
Introduction to R

Natalia Levshina © 2017

Summer School of Linguistics
Litomyšl, August 2017

How to see the meaning?

1. Introduction to R
2. Simple Correspondence Analysis
3. Multiple Correspondence Analysis
4. Behavioural Profiles and cluster analysis
5. Semantic Vector Spaces and more cluster analysis
6. Traditional semantic maps: graphs
7. Probabilistic semantic maps: Multidimensional Scaling
8. Motion charts

Course materials

- The slides, R scripts and R data are on GitHub(files will be added daily):

  https://github.com/levshina/Litomysl

- Click on Clone or download and download the zip-archive and extract the files to some local directory.
- You can access the scripts and datasets from RStudio.

Outline

1. What is R?

2. R syntax

3. R objects

What is R?

- statistical computing environment (from *t*-test to generalized linear models, and more…)
    - core distribution “base”
    - add-on packages (> 10K as of June 2017)
- programming language
- tools for creation of publication-quality plots

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. What is R?

2. R syntax

3. R objects

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
> exp(1.609438)
[1] 5
```

Creation of objects

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

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
```

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

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
```

Outline

1. What is R?

2. R syntax

3. R objects

Important data types in R

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

Numeric vectors

```
> vnum <- 1:5 # a vector of integers from 1 to 5
> vnum
[1] 1 2 3 4 5

> vnum[2] #find the second element
[1] 2
```

If it's 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"
```

Factors

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

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
```

Summarizing the data

```
> summary(mydf)
 sex        RT
 f:2      Min.   :455.0
 m:2      1st Qu.:497.8
          Median :589.5
          Mean   :601.8
          3rd Qu.:693.5
          Max.   :773.0

> str(mydf)
'data.frame':  4 obs. of  2 variables:
 $ sex: Factor w/ 2 levels "f","m": 1 2 2 1
 $ RT: num  455 773 512 667
```

Subsetting a data frame

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

> 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
1   f 455
4   f 667

> mydf[mydf$sex != "m", ]
  sex  RT
1   f 455
4   f 667

> mydf[mydf$RT < 500,]
  sex  RT
1   f 455
```

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
```

Distance matrices

```
> eurodist
```

[output omitted: distances between several European cities]

My journey yesterday:

	Leipzig	Prague	Litomysl
Leipzig	0	255	400
Prague	255	0	160
Litomysl	400	160	0

My journey

```
> mydist <- rbind(Leipzig = c(0, 255, 400), Prague = c(255, 0, 160),
Litomysl = c(400, 160, 0))
> colnames(mydist) <- rownames(mydist)
> mydist
          Leipzig Prague Litomysl
Leipzig         0    255      400
Prague        255      0      160
Litomysl      400    160        0
```

From matrix to distance matrix and back

```
> mydist <- as.dist(mydist)
> mydist
          Leipzig Prague
Prague    255
Litomysl  400     160

> m <- as.matrix(mydist)
> m
          Leipzig Prague Litomysl
Leipzig   0       255    400
Prague    255     0      160
Litomysl  400     160    0
```

Quest

1. Compute the square root of 1681.
2. Type in R: set.seed(x), where x is the result of step 1.
3. Create a random sample of 100 numbers from 1 to 100.
4. Find the 20th element.
5. Take the yth letter in the English alphabet. Write down the letter.
6. Open the help page of the function read.table and find the subsection “See also”. Find the first R function mentioned in that subsection. Remove the first letter and write down the result.
7. Find R citation information using citation(). Take the 3rd word and write down the letter.
8. Put all words together!