

Polish Language(s) and Digital Humanities Using R

G. Moroz

2020

Contents

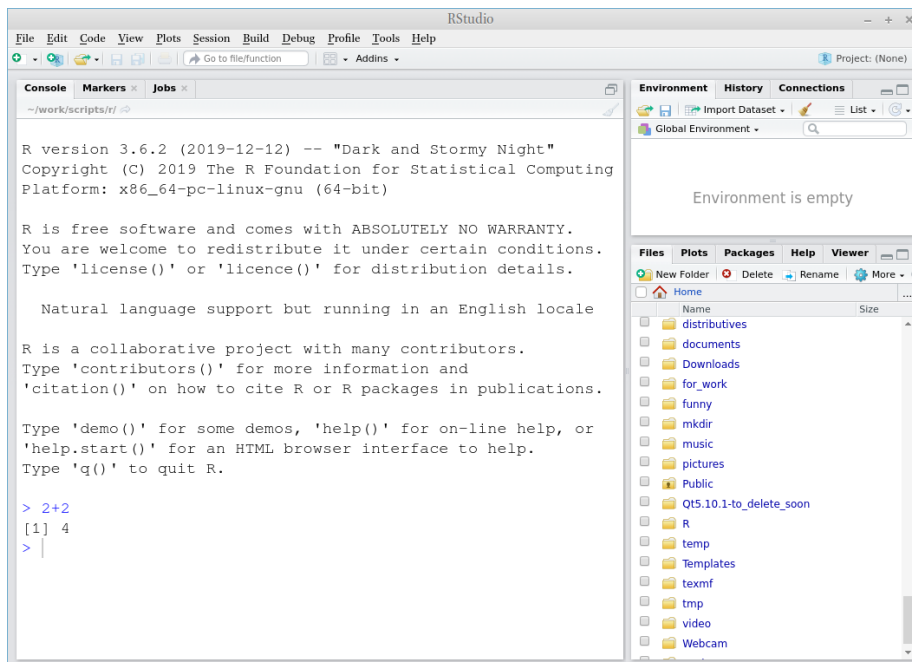
1	Prerequisites	1
2	Introduction to R and RStudio	3
2.1	Introduction	3
2.2	Introduction to RStudio	4
2.3	R as a calculator	5
2.4	Comments	6
2.5	Functions	6
2.6	Variables	9
2.7	Variable types	11
2.8	Vector	11
2.9	Dataframe (tibble)	11
2.10	Packages	11
2.11	Data import	11
2.12	Rmarkdown	11
3	Data manipulation: dplyr	13
4	Data visualisation: ggplot2	15
5	Strings manipulation: stringr	17
6	Text manipulation: gutenbergr, tidytext, udpipe	19
7	Stylometric analysis: stylo	21

Chapter 1

Prerequisites

Before the classes I would like to ask you to follow the instructions mentioned below to prepare your device for the class work:

- install **R** from the following link: <https://cloud.r-project.org/>
- install **RStudio** from the following link: <https://rstudio.com/products/rstudio/download/#download> (FREE version, no need to pay!)
- after the installation run the RStudio program, type `2+2`, and press **Enter**.



If you see something like this, then you are well prepared for classes.

- Go to the <https://rstudio.cloud/> website and sign up there. This is optional, but it will be a backup version, if something will not work on your computer.

Chapter 2

Introduction to R and RStudio

2.1 Introduction

2.1.1 Why data science?

Data science is a new field that actively developing lately. This field merges computer science, math, statistics, and it is hard to say how much science in data science. In many scientific fields a new data science paradigm arises and even forms a new sub-field:

- Bioinformatics
- Crime data analysis
- Digital humanities
- Data journalism
- Data driven medicine
- ...

There are a lot of new books “Data Science for ...”:

- psychologists (Hansjörg, 2019)
- immunologists (Thomas and Pallett, 2019)
- business (Provost and Fawcett, 2013)
- public policy (Brooks and Cooper, 2013)
- fraud detection (Baesens et al., 2015)
- ...

Data scientist need to be able:

- gather data
- transform data

- visualize data
- create a statistical model based on data
- share and represent the results of this work
- organize the whole workflow in the reproducible way

2.1.2 Why R?

R (R Core Team, 2019) is a programming language with a big infrastructure of packages that helps to work in different fields of science and computer technology.

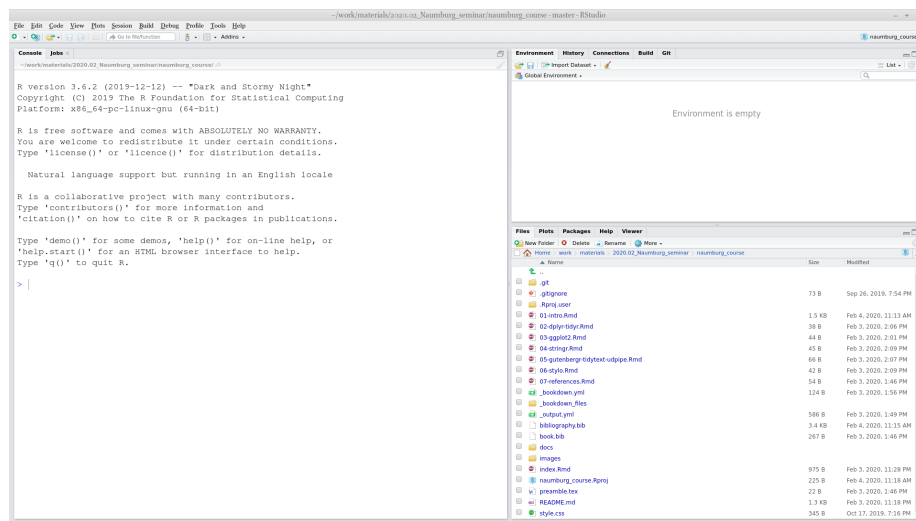
There are several alternatives:


- Python (VanderPlas, 2016; Grus, 2019)
- Julia (Bezanson et al., 2017)
- bash (Janssens, 2014)
- java (Brzustowicz, 2017)
- ...

2.2 Introduction to RStudio

R is the programming language. RStudio is the most popular IDE (Integrated Development Environment) for R language.

When you open RStudio for the first time you can see something like this:



When you press  button at the top of the left window you will be able to see all four panels of the RStudio.

5



Lets first start with a calculator. Press in R console

9+.5

```
## [1] 9.5
```

```
pi
```

```
## [1] 3.141593
```

Remainder after division

```
10 %% 3
```

```
## [1] 1
```



So you are ready to solve some really hard equations (round it four decimal places):

$$\frac{\pi + 2}{2^{3-\pi}}$$

list of hints

Are you sure that you rounded the result? I expect the answer to be rounded to four decimal places: 0.87654321 becomes 0.8765.

Are you sure you didn't get into the brackets trap? Even though there is no any brackets in the mathematical notation, you need to add them in R, otherwise the operation order will be wrong.

2.4 Comments

All text after the hash # within the same line is considered a comment.

```
2+2 # it is four
```

```
## [1] 4
```

```
# you can put any comments here
3+3
```

```
## [1] 6
```

2.5 Functions

The most important part of R is functions: here are some of them:

```
sqrt(4)
```

```
## [1] 2
```

```
abs(-5)
```

```
## [1] 5
```

```
sin(pi/2)
```

```
## [1] 1
```

```
cos(pi)
```

```
## [1] -1
```

```
sum(2, 3, 9)
```

```
## [1] 14
```

```
prod(5, 3, 9)
```

```
## [1] 135
```

Each function has a name and zero or more arguments. All arguments of the function should be listed in parenthesis and separated by comma:

```
pi
```

```
## [1] 3.141593
```

```
round(pi, 2)
```

```
## [1] 3.14
```

Each function's argument has its own name and serial number. If you use names of the function's arguments, you can put them in any order. If you do not use names of the function's arguments, you should put them according the serial number.

```
round(x = pi, digits = 2)
```

```
## [1] 3.14
```

```
round(digits = 2, x = pi)
```

```
## [1] 3.14
```

```
round(x = pi, d = 2)
```

```
## [1] 3.14
```

```
round(d = 2, x = pi)
```

```
## [1] 3.14
```

```
round(pi, 2)
```

```
## [1] 3.14
```

```
round(2, pi) # this is not the same as all previous!
```

```
## [1] 2
```

There are some functions without any arguments, but you still should use parenthesis:

```
Sys.Date() # correct
```

```
## [1] "2020-02-05"
```

```
Sys.Date # wrong
```

```
## function ()  
## as.Date(as.POSIXlt(Sys.time()))  
## <bytecode: 0x56865248a9e8>  
## <environment: namespace:base>
```

Each function in R is documented. You can read its documentation typing question mark before the function name:

As a result, no output in the Console, and a new variable x appear in the Environment window. From now on I can use this new variable:

```
x + x
```

```
## [1] 22
```

```
sum(x, x, 7)
```

```
## [1] 29
```

All those operation don't change the variable value. In order to change the variable value you need to make a new assignment:

```
x <- 5 + 6 + 7
```

The fast way for creating `<-` in RStudio is to press `Alt -` on your keyboard.

It is possible to use equal sign `=` for assignment operation, but the recommendations are use arrow `<-` for the assignment, and equal sign `=` for giving arguments' value inside the functions.

2.7 Variable types

2.8 Vector

2.8.1 Vector coercion

2.8.2 Vector operations

2.8.3 Vector recycling

2.8.4 Indexing vectors

2.8.5 NA value

2.9 Dataframe (tibble)

2.9.1 Indexing dataframes

2.10 Packages

2.11 Data import

2.11.1 .csv files

2.11.2 .xls and .xlsx files

2.12 Rmarkdown

Chapter 3

Data manipulation: dplyr

Chapter 4

Data visualisation: ggplot2

Chapter 5

Strings manipulation: `stringr`

Chapter 6

Text manipulation:
gutenbergr, tidytext,
udpipe

Chapter 7

Stylometric analysis: `stylo`

Bibliography

- Baesens, B., Van Vlasselaer, V., and Verbeke, W. (2015). *Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection*. John Wiley & Sons.
- Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98.
- Brooks, H. and Cooper, C. L. (2013). *Science for public policy*. Elsevier.
- Brzustowicz, M. R. (2017). *Data Science with Java: Practical Methods for Scientists and Engineers*. O’Reilly Media, Inc.
- Grus, J. (2019). *Data science from scratch: first principles with python*. O’Reilly Media, Inc.
- Hansjörg, N. (2019). *Data Science for Psychologists*. self published.
- Janssens, J. (2014). *Data Science at the Command Line: Facing the Future with Time-tested Tools*. O’Reilly Media, Inc.
- Provost, F. and Fawcett, T. (2013). *Data Science for Business: What you need to know about data mining and data-analytic thinking*. O’Reilly Media, Inc.
- R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Thomas, N. and Pallett, L. (2019). *Data Science for Immunologists*. CreateSpace Independent Publishing Platform.
- VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. O’Reilly Media, Inc.