

Lab 01 Intro to R and R Markdown

Lusine Zilfимian

February 19 (Wednesday), 2020

Contents

- Prerequisites
- Useful Things
- R Coding Style
- R Project
- R Markdown

Prerequisites

- Go to **<https://cran.r-project.org/>** and install/update the last version of **R 3.6.(1|2|3)**
- For useful **R Studio** shortcuts press *Alt + Shift + K*
- Repeat/learn, please, the basic R functions, objects (*vector, matrix, dataframe, array, list*), special values, conditions and loops, data visualization techniques
- For very beginners: **The (great) book of R**

Useful things in R that you probably didn't know about

- You can use the sign “;” between commands to execute more than one command sequentially on one line

```
exp(1), pi
```

```
## Error: <text>:1:7: unexpected ' ','  
## 1: exp(1),  
##           ^
```

```
exp(1); pi
```

```
## [1] 2.718282
```

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

- Try to play with pressed *Alt + Ctrl + Cursor*. Ne blagodarite! 😊

Assignment operators in R

- Is there any difference among these 3 assignments?

```
x = 2020 + 17
x <- 2020 + 17
x <<- 2020 + 17
2020 + 17 -> x
```

- R has 3 assignments operators “<-”, “=”, and “<<-”
- Read about the difference among AOs in “The book of R”, Chapter 2.2 Assigning Objects
- See also the following **discussion**.

Printing result

- To see the output/result type `x`, **print** `x` or use **parentheses**

```
x; print(x); (x <- 5)
```

```
## [1] 2037
```

```
## [1] 2037
```

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

- To *remove* the object `x` from the environment, run

```
rm(x); x
```

```
## Error in eval(expr, envir, enclos): object 'x' not found
```

- To *comment* large chunk of text, select the text and press *Ctrl + Shift + C*

Coding Style

- Is there any difference?

```
attach(mtcars) # Try to avoid using the function attach()  
x <- (cyl * 12) + mpg  
x<-cyl*12+mpg
```

- What about these names?

goof_name.R

bad models name.R

sum.r

mean.r

Coding Style

- “*Good coding style is like correct punctuation: you can manage without it, but it sure makes things easier to read.*”
- Yes, *sometimes* we can skip this “not important” styling issues and execute the code without problems.
- But the (=my) recommendation is to be as neat as possible. So use this **Google’s R Style Guide** to learn basic rules of coding.

About R Packages

- R has *built-in*, small number of *recommended*, and thousands of *contributed* packages.
- The *recommended* package needs to be installed only once (the easiest way to do this is using the `install.packages()`)

About R Packages

- However, the process of loading the library should be done every time you start a new R Session (by simply executing `library()`)

```
install.packages(dplyr)
library(dplyr)
```

- Do you remember how to know more about the package/function?^a

```
?sd # for function from R base package
??summarise # for function from contibuted package
?dplyr::summarise # as package is called
hepl(package = "dplyr") # help on the package
```

^a Each package on CRAN has its own webpage,
e.g. <https://cran.r-project.org/web/packages/dplyr/index.html>

R project

- We can type and execute the code in *R Console*, but cannot save it as a file
- **.R** file helps us to save, update and change the code, but execute it line by line
- **.Rmd** file helps to combine code, text, and result
- First, let's create the R Project
- *R Project* used to facilitate the working process with multiple script files. It is just a simple folder with the project file, hidden directory, and source documents
- To create the *R Project* go to *File* → *NewProject...*
- Now, you can create any type of file inside the **R Project**
- For details see [here](#).

Why do we need R Studio Project?

- While opening a project, R restores previous work and project history includes recent commands which had been used in the project.
- Source pane remembers the files which had been opened.
- You can search any word/phrase without opening the file using *Ctrl + Shift + F*.

R Markdown

- To create simple report using R amazing¹ **R Markdown** is needed².
- R Markdown is **fully reproducible**.
- R Markdown combines **text**, **code**, and **output** in one report.
- R Markdown allows using multiple languages (Bash, Python, etc.)
- R Markdown supports dozens of static and dynamic output formats (HTML, PDF, MS Word, Beamer³, etc.)
- R markdown is run in its own environment.

¹ Astonishing, beautiful, WONDERFUL, add your own word _____

² Actually, we can, also, manually copy and paste codes and outputs, for example, in MS Word, but IMHO it is not an optimal variant

³ I created this presentation using *Beamer*

Creating/Opening R Markdown file

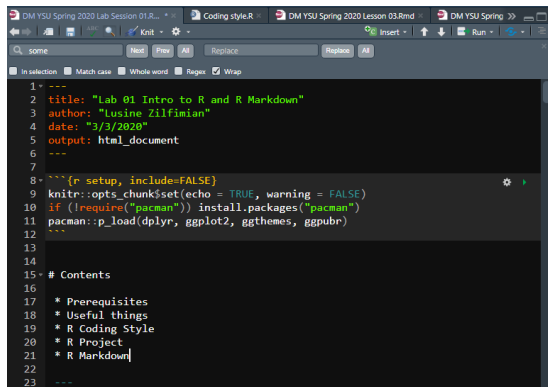
- First step is to install and load needed packages

```
install.packages(rmarkdown)
install.packages(knitr)
# Pacman allows to load libraries using one function
if (!require("pacman")) install.packages("pacman")
pacman::p_load(rmarkdown, knitr)
```

- To create a New R Markdown file:
- use *File* → *NewFile* → *R Markdown*
- choose your own output format
- To load your saved R markdown file from R:
- *File* → *Open File...*

Template R Markdown document

- **YAML header** - metadata, (surrounded by `---`). Here you can at least set the title, the author, the date, and the output.
- **Narrative text** - any kind of content.
- R code **chunks**, (surrounded by `"{r}"`)



```
1 ---
2 title: "Lab 01 Intro to R and R Markdown"
3 author: "Lusine Zilfimian"
4 date: "3/3/2020"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE, warning = FALSE)
10 if (!require("pacman")) install.packages("pacman")
11 pacman::p_load(dplyr, ggplot2, ggthemes, ggpubr)
12 ```
13
14
15 # Contents
16
17 * Prerequisites
18 * Useful things
19 * R Coding Style
20 * R Project
21 * R Markdown
22
23 ---
```

Adding, Labeling, and Using Chunk Options

- To add chunk use **insert** button on the right corner of the toolbar, hit \$ Ctrl + Alt + I\$ or do it manually
- Add the name to find it content of the bottom line of .Rmd file.
- Use the gear to set different options for the chunk.
- Customize R code chunks providing optional arguments after the {r}

Some options

- R Markdown would ignore any messages or warnings generated by the chunk below.

```
library(dplyr)
```

- **echo = FALSE** - R Markdown will not display the code in the final document (but it will still run the code and display its results unless told otherwise).
- **eval = FALSE** - R Markdown will not run the code or include its results, (but it will still display the code unless told otherwise).
- **results = 'hide'** - R Markdown will not display the results of the code (but it will still run the code and display the code itself unless told otherwise).

Running codes in R Markdown

- To run the line of command put the cursor at that line and press **Ctrl+Enter**
- To run the chunk, select all chunk and press **Ctrl+Enter** or just press the Run button on the right corner of the chunk.

Running Python code in R Markdown

- Insert python chunk and type the code:

```
import pandas as pd
bw_pd = pd.read_csv("bweight.csv")
print(bw_pd.head(2))
```

```
##      id  headcir  length  bweight  ...  bined_weight  lowbwt
## 0   431      12      19      4.2  ...              5      1
## 1   300      12      18      4.5  ...              5      1
##
## [2 rows x 17 columns]
```

Running Python code in R Markdown

- **Reticulate** is developed by R Studio to add the functionality of using Python and other languages in the R Studio environment.
- Access object created with python in R:

```
library(reticulate)
head(py$bw_pd, 2)
```

```
##      id headcir length bweight gestation smoker motherage mnc
## 1 431      12      19      4.2         33      1         20
## 2 300      12      18      4.5         35      1         41
##      mppwt fage fedyrs fnocig bined_weight lowbwt mage35 LowBi
## 1      109  20      10      35           5      1         0
## 2      125  37      14      25           5      1         1
```

Inline coding and LaTeX equations

- You can embed R code into the text of your document with the 'r' syntax.
- Surround the equation⁴ with one “\$” sign to embed an equation inline:

```
$`\frac{n!}{k!(n-k)!} = \binom{n}{k}$
```

- Surround the equation with two “\$” sign to embed an equation in new line:

```
$$`\frac{n!}{k!(n-k)!} = \binom{n}{k}$$
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

⁴Standard latex math symbols see here: <https://en.wikibooks.org/wiki/LaTeX/Mathematics>

Lists in R Markdown

- To make a **bulleted list**, place each item on a new line after an asterisk and a space, like this:

- * Leonardo Pisano Fibonacci
- * Augustin-Louis Cauchy

- To make an **ordered list** by placing each item on a new line after a number followed by a period followed by space, like this

- 1 Bernard Bolzano
- 2 Karl Weierstrass

In each case, you need to place a blank line between the list and any paragraphs that come before it.

Text formatting

- If you want to start from a new page use `\newpage` or `\pagebreak`
- To make the text *italic* surround the text with one asterisk, for **bold** - two asterisk:

`*italic*`

`**bold**`

- You can set Headers using “#”s
- To insert images and links you can use the following syntax:

[Basics of Markdown formatting tools] (<https://rstudio.com/wp->

! [Name of Image] ("My.PNG")

R Markdown Help/Cheatsheets

- This is only the basics of R Markdown syntax
- For more tools you can go to
- *Help* → *Cheatsheets* → *R Markdown Cheat Sheet*
- *Help* → *Cheatsheets* → *R Markdown Reference Guide*
- And/or googling 😊.