Introduction to R Programming

Slide Set 1: Course Organization and Basics

Maria Ptashkina

Barcelona GSE ITFD

September 2021

Table of Contents

Course Organization

- 2 weeks (6-17 September), 5 classes per week, 1 hour per class
- Email: maria.ptashkina@barcelonagse.eu
- Goal: learn core functionality of R, prepare to work with real data
- Disclaimer: I cannot *teach* you to use R! The key is to practice yourselves (actually, the key is Stack Overflow / Stack Exchange)
- My goal: explain to you the *logic* of R programming

By Now You Should Have...

- Finished the free on-line course about the basic concepts of R at DataCamp ▸ DataCamp
- Installed R and R Studio on your personal computers ▸ R ▸ R Studio ▸ Instructions

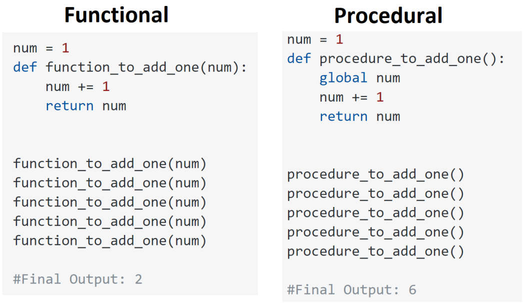
Plan

Topic	Details
1 Introduction and refreshing the basics	Variables, data types, vectors, matrices, factors, data frames, lists
2 Exploratory Data Analysis with R	Descriptive statistics, data wrangling and merging, tidyverse
3 Data Visualization	Types of data and plots, ggplot
4 R Programming 1	Conditional statements, logical expressions
5 R programming 2	Loops, functions, the 'apply' family
6 Application 1: Income inequality	Simple data analysis and plotting
7 Application 2: The effect of sugar tax	Applied study
8 Application 3: International Trade	Regressions
9 Application 4: Time series	Time series
10 Wider R functionality	API, web scraping, geospatial data (maps), text as data, R Markdown and notebooks

Table of Contents

Programming Styles

- Procedural programming: writing a list of instructions to tell the computer what to do step by step
- Functional programming: writing a mathematical function, i.e. a function that takes arguments and returns a value



Functional

```
num = 1
def function_to_add_one(num):
    num += 1
    return num


function_to_add_one(num)
function_to_add_one(num)
function_to_add_one(num)
function_to_add_one(num)
function_to_add_one(num)

#Final Output: 2
```

Procedural

```
num = 1
def procedure_to_add_one():
    global num
    num += 1
    return num


procedure_to_add_one()
procedure_to_add_one()
procedure_to_add_one()
procedure_to_add_one()
procedure_to_add_one()

#Final Output: 6
```

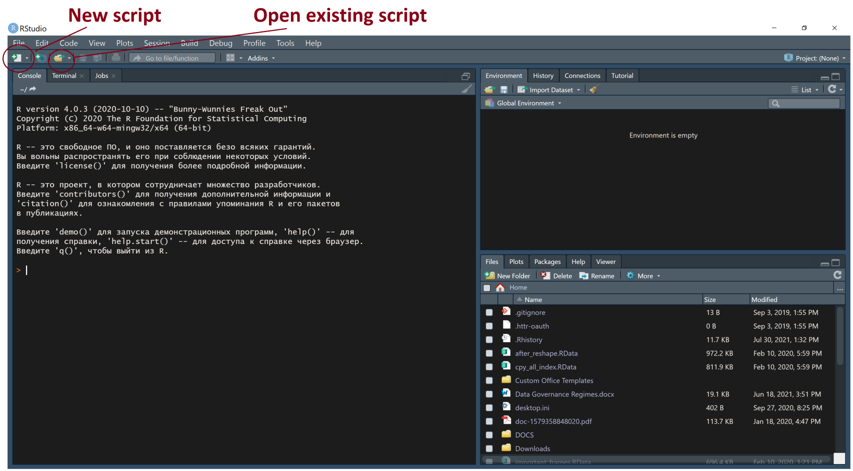
- R is a functional programming language ▸ Nerdy Curiosity 1

Why R?

- Be pragmatic: work in any language / program you like as long as you can achieve your goal
- In Economics we also use `MATLAB`, `Stata`, `Python` and `Julia`
- Generally, `MATLAB` and `Julia` are best for numerical problems and simulations, while `R` and `Python` are great at data handling (`Stata` is not a language, but is very convenient for working with data)
  Comparison
- `R` is a constantly evolving user-driven language
- `R` has a fantastic (great amazing best best ever) community online (Stack Overflow / Stack Exchange / R Bloggers, etc.)

Open R

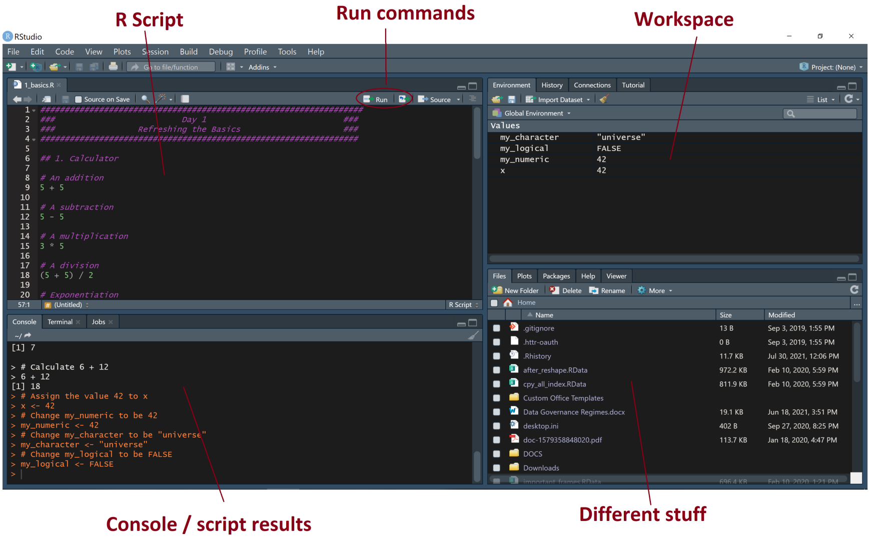
- Reproducible research: we work with R scripts


New script
Open existing script
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Console Terminal Jobs
~/
Environment History Connections Tutorial
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R -- это свободное ПО, и оно поставляется безо всяких гарантий.
Вы вольны распространять его при соблюдении некоторых условий.
Введите 'license()' для получения более подробной информации.

R -- это проект, в котором сотрудничает множество разработчиков.
Введите 'contributors()' для получения дополнительной информации и
'citation()' для ознакомления с правилами упоминания R и его пакетов
в публикациях.

Введите 'demo()' для запуска демонстрационных программ, 'help()' -- для
получения справки, 'help.start()' -- для доступа к справке через браузер.
Введите 'q()', чтобы выйти из R.

> |
```

Name	Size	Modified
.gitignore	13 B	Sep 3, 2019, 1:55 PM
.httr-oauth	0 B	Sep 3, 2019, 1:55 PM
.Rhistory	11.7 KB	Jul 30, 2021, 1:32 PM
after_reshape.RData	972.2 KB	Feb 10, 2020, 5:59 PM
cpy_all_index.RData	811.9 KB	Feb 10, 2020, 5:59 PM
Custom Office Templates		
Data Governance Regimes.docx	19.1 KB	Jun 18, 2021, 3:51 PM
desktop.ini	402 B	Sep 27, 2020, 8:25 PM
doc-1579358848020.pdf	113.7 KB	Jan 18, 2020, 4:47 PM
DOCS		
Downloads		
important_frames.RData	696.4 KB	Feb 10, 2020, 1:21 PM

Working in R



```
########################################################################
###                         Day 1                                    ###
###                   Refreshing the Basics                          ###
########################################################################

## 1. Calculator

# An addition
5 + 5

# A subtraction
5 - 5

# A multiplication
3 * 5

# A division
(5 + 5) / 2

# Exponentiation
```

```
~/
[1] 7

> # Calculate 6 + 12
> 6 + 12
[1] 18
> # Assign the value 42 to x
> x <- 42
> # Change my_numeric to be 42
> my_numeric <- 42
> # Change my_character to be "universe"
> my_character <- "universe"
> # Change my_logical to be FALSE
> my_logical <- FALSE
>
```

```
Values
  my_character          "universe"
  my_logical            FALSE
  my_numeric            42
  x                     42
```

Name	Size	Modified
.gitignore	13 B	Sep 3, 2019, 1:55 PM
.httr-oauth	0 B	Sep 3, 2019, 1:55 PM
.Rhistory	11.7 KB	Jul 30, 2021, 12:06 PM
after_reshape.RData	972.2 KB	Feb 10, 2020, 5:59 PM
cpy_all_index.RData	811.9 KB	Feb 10, 2020, 5:59 PM
Custom Office Templates		
Data Governance Regimes.docx	19.1 KB	Jun 18, 2021, 3:51 PM
desktop.ini	402 B	Sep 27, 2020, 8:25 PM
doc-1579358848020.pdf	113.7 KB	Jan 18, 2020, 4:47 PM
DOCS		
Downloads		
important_frames.RData	696.4 KB	Feb 10, 2020, 1:21 PM

Basics

- R is case sensitive
- To run one line `Ctrl + Enter`
- To run the entire code `Ctrl + Shift + Enter`
- # sign to add comments

R as Calculator

- Addition: +
- Subtraction: −
- Multiplication: ∗
- Division: /
- Exponentiation: ˆ
- Modulo: %% - returns the remainder of the division of the number to the left by the number on its right

Variable assignment

- A basic concept in (statistical) programming
- Allows to store a value (e.g. 4) or an object (e.g. a function description) in R
- Assignment operator is typically `<-`
- In case you're wondering why it's not = ▸ Nerdy Curiosity 2

Basic data types in R

- Numerics: Decimal values and integers
- Logical: boolean values (TRUE or FALSE)
- Characters: text (or string) values (denoted using `""` quotation marks)
- Super important to know which type of data you're working with: check using `class()` function

Vector

- Vector is a one dimensional array to store different types of data
- To create a vector use 'combine' function `c()`, list elements separated by comma
- To assign names to the elements of a vector use `names()` function
- Sum of two vectors in `R` is element-wise
- Sum of all elements of a single vector is calculated using `sum()` function
- Vector selection
  - Use square brackets to select by number (Note! The first element in a vector has index 1, not 0 as in many other programming languages)
  - Select a sub-vector using `c()`
  - Select by using names

Relational Operators

- `<` for less than
- `>` for greater than
- `<=` for less than or equal to
- `>=` for greater than or equal to
- `==` for equal to each other
- `!=` not equal to each other

Matrix

- A matrix is a collection of elements of the *same* data type arranged into a fixed number of rows and columns
- Construct a matrix using `matrix()`
- Add names to row or colums using `rownames()` and `colnames()`
- Calculate sum across rows or columns using `colSums()` and `rowSums()`
- Append matrices or vectors using `cbind()` or `rbind()`
  - Note! `R` will give you a warning but still run the commands if the matrix dimensions or data types don't match!
- Element selection is similar to vectors, but now we have two dimensions
- `+`, `-`, `/`, `*`, etc. work in an element-wise way on matrices
  - Note that this is not matrix multiplication! For that you should be using `%*%`

Factors

- Statistical data type used to store categorical variables (as opposed to continuous)
- Create factors using `factor()` (encodes the vector as a factor)
- Nominal vs. an ordinal categorical variables
- To change the names of the levels use `levels()`

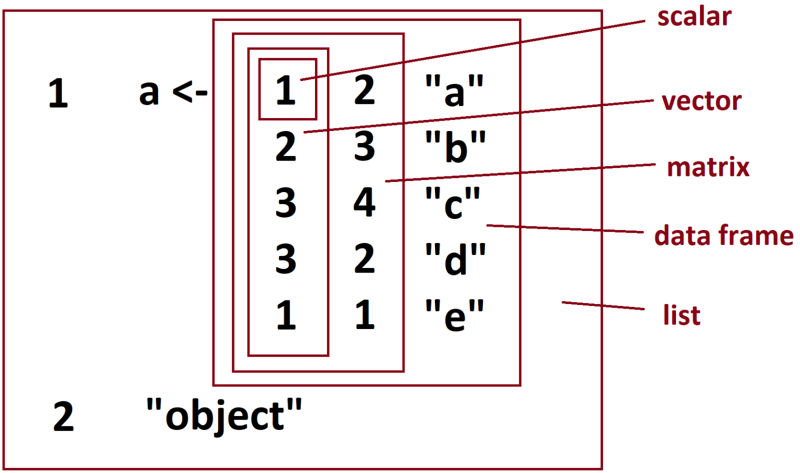
Data Frame

- Data frame is a data set of different data types
- To take a look at the data frame, use `head()`, `tail()`, `str()`
- To create a data frame use `data.frame()`
- If columns have names, use $ to select a whole column
- To choose parts of a data frame use `subset(df, some condition)`
- To sort a data frame use `order()`
- For now we are focusing on 'classic' approach to learning `R`, but if you're interested you can read a bit about tibbles and tidyverse
  ▸ Tibbles?? ▸ Tidyverse?? ▸ Eh!?

Lists

- List in `R` allows you to gather a variety of objects under one name
- To create a list use `mylist <- list(comp1, comp2 ...)` where components can be matrices, vectors, other lists
- To select a component use the numbered position of that component and double square brackets `[[]]`, or the $ sign

Data Structures in R



Homework

- Pre-install `tidyverse` package on your computers using the following code:
  `install.packages("tidyverse")`
- R will download the packages from CRAN (the **c**omprehensive **R** **a**rchive **n**etwork) and install them on to your computer
- If you have problems installing, make sure that you are connected to the internet, and that nothing blocked by your firewall or proxy
- If you still have problems, contact me via email

References and Resources

References

- DataCamp Introduction to R ▸ DataCamp

Resources

- Code and Data for the Social Sciences: A Practitioner's Guide ▸ Guide