Applied Time Series Econometrics: Introduction to R

Oxana Malakhovskaya, NRU HSE

September 03, 2019

What is R software?

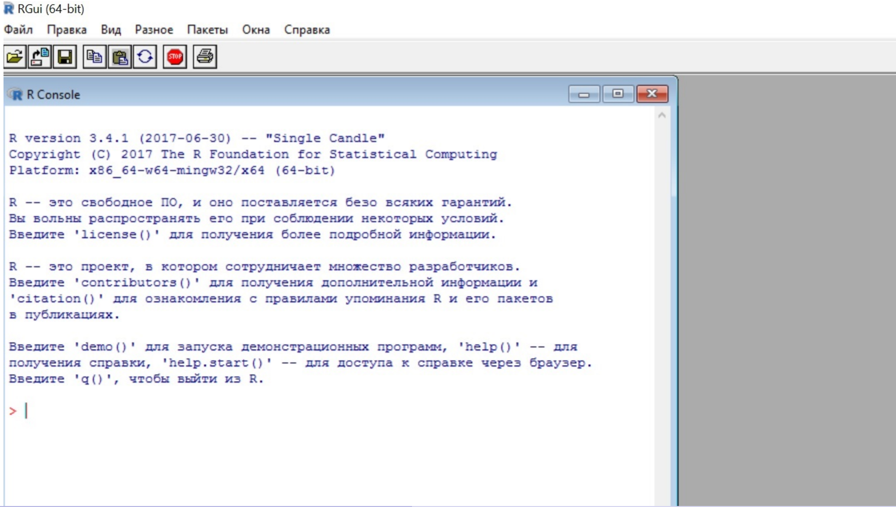
- R is a powerful open-source software for statistics, econometrics, graphics and so on.
- R is a serious competitor to all excellent commercial statistical packages in spite of the fact that its implementation is a collective accomplishment of thousands of volunteers.

Downloading and installation

- To download R go to: www.cran.r-project.org
  and choose a distribution according to your computer's operating system. CRAN stands for an abbreviation from Comprehensive R Archive Network
- Use the system's manager to download and install the software.
- Double-click on the R icon on your desktop to run R.

Running R

The main part of the window on your screens is called R console where we can enter commands.



```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 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.

> |
```

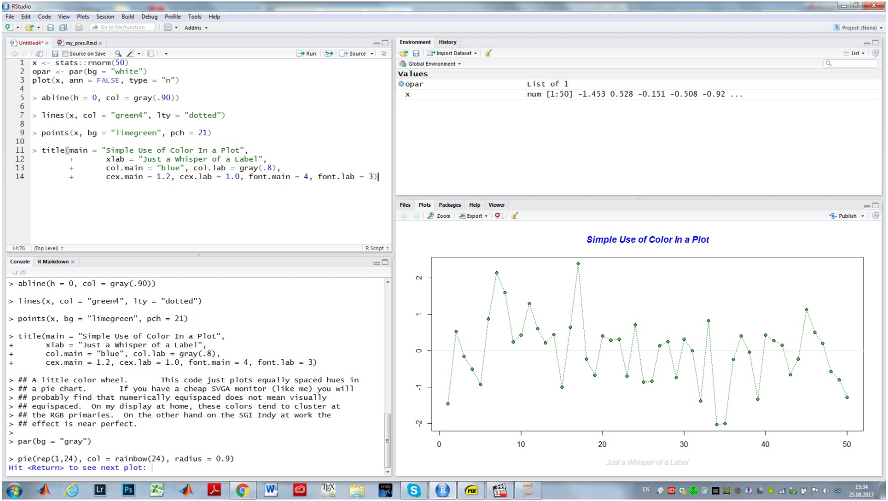
What is Rstudio?

- To make programming a little bit easier and the interface user-friendlier we also download and install Rstudio.
- Rstudio is a free and open-source integrated development environment for R.
- Owing to Rstudio, we can take advantage of an intuitive user interface for making scripts, performing graphs and obtaining the software help.
- It exists in two editions: Desktop edition and Rstudio server. We need the Desktop edition.

Downloading and installation

- To download Rstudio go to: www.rstudio.com/products/rstudio/download and choose a free desktop edition.
- Use the system's manager to download and install the software.
- Double-click on the Rstudio icon in your Taskbar to run Rstudio.

Rstudio main window



```
x <- stats::rnorm(50)
opar <- par(bg = "white")
plot(x, ann = FALSE, type = "n")

> abline(h = 0, col = gray(.90))

> lines(x, col = "green4", lty = "dotted")

> points(x, bg = "limegreen", pch = 21)

> title(main = "Simple Use of Color In a Plot",
        +       xlab = "Just a Whisper of a Label",
        +       col.main = "blue", col.lab = gray(.8),
        +       cex.main = 1.2, cex.lab = 1.0, font.main = 4, font.lab = 3)
```

Values

```
opar                          List of 1
 x                            num [1:50] -1.453 0.528 -0.151 -0.508 -0.92 ...
```

```
> abline(h = 0, col = gray(.90))

> lines(x, col = "green4", lty = "dotted")

> points(x, bg = "limegreen", pch = 21)

> title(main = "Simple Use of Color In a Plot",
+       xlab = "Just a Whisper of a Label",
+       col.main = "blue", col.lab = gray(.8),
+       cex.main = 1.2, cex.lab = 1.0, font.main = 4, font.lab = 3)

> ## A little color wheel.       This code just plots equally spaced hues in
> ## a pie chart.       If you have a cheap SVGA monitor (like me) you will
> ## probably find that numerically equispaced does not mean visually
> ## equispaced.  On my display at home, these colors tend to cluster at
> ## the RGB primaries.  On the other hand on the SGI Indy at work the
> ## effect is near perfect.
>
> par(bg = "gray")

> pie(rep(1,24), col = rainbow(24), radius = 0.9)
Hit <Return> to see next plot:
```

Rstudio layout

The main window of R studio is splitted into four parts.

- The top-left panel is a text editor where we can type the scripts.
- The top-right panel contains the Environment (the variables being used in the current session) and the History (the list of all recent commands).
- The bottom-right panel contains Files (the files system), Plots (the pane where the plots are displayed), Packages (the pane where the packages can be installed or updated), Help (the pane where the documentation can be found) and Viewer (used to view local web content).
- The bottom-left panel is the console where all the messages and results are displayed.

Quit or interrupt the computation

- To quit Rstudio type q() or press `Ctrl + Q` or go to `File` → `Quit Session`.
- To interrupt a long-running computational process click on the stop-sign icon (upper right corner of the console).

Looking for help

- To read the documentation supplied with R type `help.start()` or go to `Help` → R help.
- "Packages" and “Search Engine and Keywords” are probably the most useful.
- To see documentation for a function type `help(functionname)` or `?(functionname)`
- To see arguments of a function: `args(functionname)`
- To see examples of a function usage: `example(functionname)`

Looking for the help(2)

- If the help function reports that it can find no documentation about a function, use `help.search("functionname")` or `?? functionname`. It works when a package that contains the function is installed but not loaded for the current session.
- To get help about an installed package type `help(package = "packagename")`
- A convenient search on the web is provided by www.rseek.org It displays only R-connected results. A comprehensive index of R packages and documentation can be found at www.rdrr.io.
- An alternative is to use `RSiteSearch()` function directly in the console.

Setting a working directory

- A working directory is a default location where R looks for your data sets and scripts.
- By default, this is the same location where R save your script and your workspace.
- To get a working directory we use the function `getwd()`
- To set a working directory go to: `Session` → `Set working directory`

  The usage of a function `setwd()` is also possible.

```
setwd("D:/Job/BAC/TimeSeries/2019/Seminar1")
```

- To set a default working directory go to `Tools` → `Global options`.

R as a calculator: expressions(1)

- To get an instant reply from the program just type your task in the console. Enter any expression next to a command prompt > and see the result.

```
2 + 5
```

```
## [1] 7
```

```
3 ^ 2 * 4
```

```
## [1] 36
```

[1] means the first element of a vector.

R as a calculator: some useful operators

```r
8 / 4 #division
9 %/% 2 #integer division
9 %% 2 #remainder of integer division
sqrt(16) #squared root
```

- In case of expressions the result is instant but the value is lost.

Creating and deleting variables: assignments

- To assign a value to a variable we use the assignment operator <- (the equality sign = and the right assignment operator -> are also possible but not recommended)

```
x <- 5 + 2
```

- Note the appearance of a newly defined variable in the workspace (`Environment` tab).
- To have a list of all objects in the workspace we use `ls()` or `ls.str()` functions.

NB! All variables are case-sensitive

Workspace

- The workspace can be saved for a future session when you quit R or at any time with the function `save.image()` or by clicking on the diskette icon in the `Environment` tab.
- The workspace is saved in a file with the `.Rdata` extension in the working directory.
- To remove a variable/variables from the workspace we use `rm(variablename)` function.
- To clear the entire workspace use `rm(list=ls())` or just click on an icon with the broom.

Command History

- The command history (`History` tab) can be also saved for a future session with the function `savehistory()` or by clicking the diskette icon in the `History` tab.
- The command history is saved in a file with the `.Rhistory` extension in the working directory.
- The saved workspace and command history are reloaded automatically when a new session starts in the same working directory.

Some tips

- An elementary command can be an expression (evaluated, printed) or an assignment (not printed automatically).
- To print a variable use the function print() or just type the variable in the command prompt.
- To clear the console press `Ctrl + L`.
- Comments start with a hashtag '#'
- In case of an incomplete command, a different prompt (+) appears in the following line.
- Vertical arrows keys can be used to call a command from a command history.

Data types

R works with several data types (classes). Some of them are:

- numeric

```
a <- 6.87
```

- logical

```
c <- FALSE # or TRUE
```

- character

```
d <- "green"
```

To check a data type of a variable we use a `class(variable)` function.

Scripts

- If we have a code that we frequently use or a code we want to save for whatever reason, it is reasonable not to type it line by line in the command prompt but to create a script.
- To create a new script click on an icon with the plus sign in an upper menu or choose File → New File → R script.
- To execute one line of the code we place the cursor on this line and press `Ctrl+Enter` or click Run in the right upper corner of the Editor.
- To run the entire code we click `Source` in the right upper corner of the Editor or type `source("scriptname.R")` in the command prompt.

Vectors: general information

- To create a vector we use `c()` operator.
- A vector can contain numbers, strings, logical values but not mixtures of them. So all elements of a vector must have the same type.
- The same operator can be applied to vectors to concatenate vectors.
- Some simple functions that are often applied to vectors are `sum()`, `mean()`, `median()`, `sd()`, `var()`, `cov()`, `cor()`.

Vector arithmetic operations

- Standard arithmetic operators with vectors perform on an element-by-element basis

```
v <- c(5,3,4)
w <- c(1,9,2)
v + w
```

```
## [1]  6 12  6
```

```
v * w
```

```
## [1]  5 27  8
```

```
v / w
```

```
## [1] 5.00 0.33 2.00
```

Vector arithmetic operations(2)

- Many functions operate for entire vectors by default so we do not need to write a loop to apply the same function to all elements of a vector.

```
v + 1
```

```
## [1] 6 4 5
```

```
sqrt(w)
```

```
## [1] 1.0 3.0 1.4
```

```
log(v)
```

```
## [1] 1.6 1.1 1.4
```

Sequences

- To create a sequence with an increment equal to ±1 we use a colon between the first and the last elements of a sequence.

```
1:5
```

```
## [1] 1 2 3 4 5
```

- We use a the function `seq()` for other increments.

```
seq(from = 1, to = 5, by = 2)
```

```
## [1] 1 3 5
```

Sequences

- In place of the increment we can determine the number of elements in the sequence.

```r
seq(from = 0, to = 4, length.out = 3)
```

```
## [1] 0 2 4
```

- In case of a simple sequence consisting of a number repeated several times we use the function repeat.

```r
rep(10, times = 3)
```

```
## [1] 10 10 10
```

Vector comparison

- We can perform an element-by-element comparison of a vector with another vector or compare the elements of a vector against a scalar.
- Usual operators `==, !=, >, <, >=, <=` are used for comparison.
- Logical vector consisting of `TRUE` and `FAlSE` statements is returned as a result of the comparison.

```
v <- c(5,3,4)
w <- c(1,9,2)
v > w
```

```
## [1]  TRUE FALSE  TRUE
```

Vector element selection

- To select an element or elements of a vector we use square brackets.

```
v[3] # to select a third element of a vector v
```

- To print a vector without an element or elements negative index values are used

```
v[-4] # to remove a forth element of a vector v.
```

- To print a vector without multiple elements we use sequences or vectors

```
v[-(2:3)] # to print a vector without a 2nd and
          # a 3rd elements
v[c(5, 8, 9)] # to select 5th, 8th and 9th elements of v.
```

Vector element selection (2)

- To select elements under certain condition we use standard logical operators.

```
v[v < 3] # to select all elements of a vector v
       # that are less than 3
```

- Using names for selecting or removing variables is also possible.