RUHR-UNIVERSITÄT BOCHUM

INTRODUCTION TO R

Simon Ress
Faculty of Social Science
Chair of Social Science Data Analysis & Chair of Comparative Political Science

What is R?

- R is a comprehensive statistical environment and programming language for professional data analysis and graphical display
- It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories
- Webpage: http://www.r-project.org

Advantages:

- R is free
- It runs on a variety of platforms including Windows, Unix and MacOS.
- New statistical methods are usually first implemented in R
- It has state-of-the-art graphics capabilities
- Lots of help due to collaborative project

Disdvantages:

- R has a steep learning curve

What is RStudio?

- Powerful IDE (integrated development environment) for R
  - Some features:
    - Tab-completion of filenames, function names and arguments
    - Full-featured text editor (e.g. syntax highlighting, parenthesis and bracket matching, find/replace with regular expressions)
    - The graphical workspace
    - Seamless integration of Rmarkdown, knitr, GIT and other development tools
- Its free and open source, and works on Windows, Mac, and Linux and over the web.
- Webpage: https://www.rstudio.com/

Interaction between R and RStudio

- Before you can ask your computer to save some numbers, you'll need to know how to talk to it
- That's where R and RStudio come in
- RStudio gives you a way to talk to your computer
- R gives you a language to speak in
- Usually, you don't interact with R directly, you just have to have it installed
- To get started, open RStudio just as you would open any other application on your computer

- WHEN DO WE COMPILE?
  - In some languages, like C, Java, and FORTRAN, you have to compile your human-readable code into machine-readable code (often 1s and 0s) before you can run it
  - If you've programmed in such a language before, you may wonder whether you have to compile your R code before you can use it.
  - The answer is no. R is a dynamic programming language, which means R automatically interprets your code as you run it.

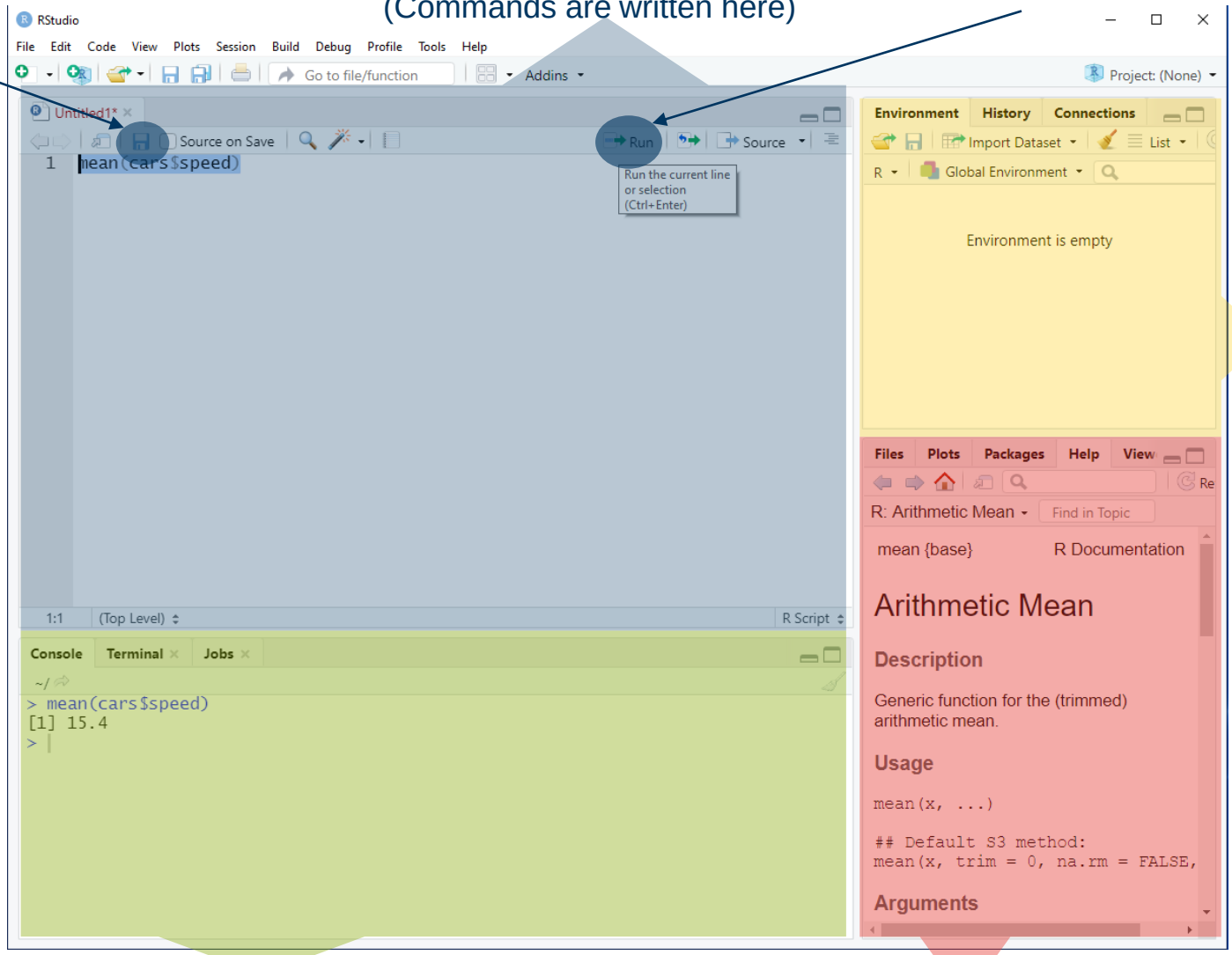
Typical workflow

1. Open Rstudio
2. Open a new or pre-existing script in Rstudio (extension .R)
3. Set working directory with setwd("path2directory")
4. Load (and install) required libraries Install with install.packages("name") - only once need to specify CRAN mirror Load with library(name) - each session if required
5. Comment your script with # really important
6. Write and execute your commands
   - Two options for execution:
     - Button: “Run”
     - Keyboard shortcut: “Ctrl”(german keyboard:“Strg”) + “Enter”
7. Save outputs in your working directory (or specify another folder)
8. Save changes to your R-script
   - Button: „Save current document“
   - Keyboard shortcut: “Ctrl”(german keyboard:“Strg”) + “S”

R-Script
(Commands are written here)

Run current line or marked section

Save current R-Script



Environment
(see datasets, vectors, functions, …)

Console
(Results of executed code & alternative for executing commands)

Multifunctional panel
(See files within working directory, plots, …)

Information on notation

Commands are always represented one-to-one in the way they are written. Words within <> mean that something has to be written at this point without the <>.

- For example, the abstract representation of the **help(<command>)** command in R would be applied like this: **help(mean)**

Getting help

- Several ways to get help for arising problems
  - Using in-R documentation by typing help(<command>) or ?<command>
  - Searching for help on stackoverflow
  - Using a web search engine („R …..“)

Contact

Published by

Simon Ress
Ruhr-Universität Bochum
Faculty of Social Science
Chair of Social Science Data Analysis & Chair of Comparative Political Science

Phone: +49(0)234 32 - 19672
E-Mail: Simon.Ress@rub.de