

Open Online Introduction to R Course

Introductory Lecture

Wolfgang Viechtbauer

Maastricht University

2020-09-16

Introduction

- Wolfgang Viechtbauer

Department of Psychiatry and Neuropsychology

School for Mental Health and Neuroscience

Maastricht University, The Netherlands

- website: <http://www.wvbauer.com/>

- email: wvb@wvbauer.com

- twitter: <https://twitter.com/wviechtb>

Course Materials

- http://www.wvbauer.com/course_oor
- lecture slides, code, data, some useful links
- website will be updated during the course
- download the files to your computer
- also available at [GitHub](#) and [GitLab](#)

What is R?

- R is a system for the manipulation, statistical and numerical analysis, and graphical display of data
- freely available under the [GNU General Public License](#) (GPL) → open-source and free/libre
- runs under Windows, MacOS, Unix/Linux, ...

History of S and R

- ... it began May 5, 1976 at:

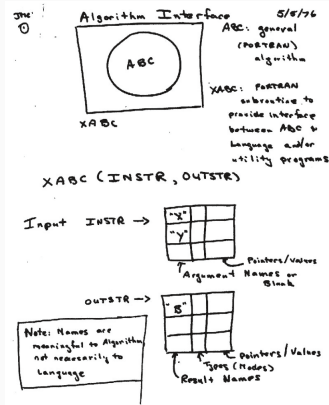


Bell Laboratories, Murray Hill, New Jersey¹

¹Photo by Alcatel-Lucent Bell Labs

History of S and R

- informal meeting to discuss development of a new system for statistical computing
- first implementation made by Rick Becker and [John Chambers](#) (and a few others)
- called “the system”



sketch of the system design
made on the first meeting

History of S and R

- “the system” → “S” (the S language)
- first UNIX version of S in 1979 (version 2)
- distributed outside Bell Labs in 1980
- source code released in 1981, then licensed in 1984 for educational and commercial purposes
- video: Rick Becker on [Forty Years of S](#)

History of S and R

- Becker & Chambers (1984). *S: An Interactive Environment for Data Analysis and Graphics*.
- Becker & Chambers (1985). *Extending the S System*.
- Becker, Chambers, & Wilks (1988): *The New S Language: A Programming Environment for Data Analysis and Graphics*.
- Chambers & Hastie (1991). *Statistical Models in S*.
- Chambers (1998). *Programming with Data: A Guide to the S Language*.

History of S and R

- **S-PLUS**, a commercial implementation of S, released in 1988 by Statistical Sciences, Inc. (now **TIBCO**)
- in 1992, **Robert Gentleman** and **Ross Ihaka** start developing a programming language “not unlike S”



Robert Gentleman and Ross Ihaka²

²Photos by Stuart Isett and Kieran Scott

Some R Milestones

- first binary of R released in 1993
- Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3), 299-314. [[link](#)]
- source code released in 1997 (CRAN is started)
- [R Core group](#) is formed in 1997 with 9 members
- version 1.0 (2000), version 2.0 (2004), version 3.0 (2013)
- first [useR! conference](#) in May 2004 in Vienna, Austria
- current version: R 4.0.2 released June 2020

Other Related Developments

- [Revolution Analytics](#) founded in 2007 (now part of MS)
- [RStudio](#) founded in 2008
- [New York Times article](#) about R in January 2009
- [R Consortium](#) founded in 2015
- [data science](#) develops as a discipline
- [open science, reproducible research](#)
- the emergence of the [tidyverse](#)

Why is it called R?

- **R**oss Ihaka and **R**obert Gentleman
- pun/play on the name of the S language
(which in turn was probably a pun based on the C programming language, also developed at Bell Labs)

Basic Concepts

- command-driven (no point-and-click interface)
- an 'object-oriented' and 'functional' language
- R console: what you see when you start R
- symbol at beginning of line (>): the 'prompt'

Modes of Interacting with R

- **interactively:** you type commands into the R console line by line and get direct feedback
- **via script files:** you type commands into a script file and then can:
 - copy-paste commands to the console
 - read in and execute all commands at once (e.g., with `source()`, `Rscript`, ...)

Interactive Mode

- to use R as a “calculator on speed”

```
> x <- c(4,2,3,6)
```

```
> mean(x)
```

```
[1] 3.75
```

- useful for spontaneous exploration of data

Tab Completion and Scrolling

- when typing in commands, can use ‘tab completion’ (esp. useful for long commands)
- type `sq` and hit Tab: `sqrt` (tada!)
- if ambiguous, can get list with possible options
- type `ex` and hit Tab: get nothing, but hit Tab again, get list of options
- with `↑` `↓` keys, scroll through command history
- hit ESC (vigorously) if you are ‘stuck’ somewhere

Commands Over Multiple Lines

- start typing:
`> mean(`
- hit return
- command is syntactically not complete
- continue on next line (prompt is now a `+ sign`)
`> mean(
+ x)`
- hit return
`[1] 3.75`

Always Use Script Files

- promotes:
 - organized programming/analyses
 - code reuse
- increases replicability
- easier to fix errors/mistakes
- can write/edit script files with:
 - the built-in editor
 - an external editor
 - an [integrated development environment](#) (IDE)

The Built-In R Editor

- on Windows: rudimentary editor for script files
- on MacOS: a multidocument editor with 'syntax highlighting' and 'brace-matching'
- start new script: Menu File – New Script
- put cursor in line to be executed and hit Ctrl-R (Windows) or Command-Return (MacOS)
- or highlight parts to be executed
- can save/load scripts (usually .r or .R extension)

External Text Editors

- script files are just plain-text files
- can therefore write them with your favorite text editor
- some editors have functionality for opening multiple documents, code execution, syntax highlighting, brace-matching, and other useful features
- I personally use [Sublime Text](#) with some plug-ins (i.e., [Terminus](#), [SendCode](#), [Shell Exec](#), [Origami](#), ...)

Exiting R / Saving the Workspace

- can quit R with:
 `> quit()`
 or `q()` or by just closing the window
- you will get a prompt asking if you want to “Save workspace image? [Yes/No/Cancel]”
- if you choose yes: R will save the state of your workspace to the current working directory (into the files `.RData` & `.Rhistory`)
- my recommendation: never do this (choose ‘no’)

RStudio

- [RStudio](#) has created an IDE for R (same name)
- open source (commercial edition also available)
- runs on Windows, MacOS, and Linux
- some of the useful features:
 - syntax highlighting
 - code completion
 - bracket matching
 - object list and command history
 - can run session remotely

Some RStudio Keyboard Shortcuts

Description	Windows	MacOS
Start new script	Ctrl+Shift+n	Command+Shift+n
Open script	Ctrl+o	Command+o
Save script	Ctrl+s	Command+s
Close script	Ctrl+w	Command+w
Show keyboard shortcuts	Alt+Shift+k	Option+Shift+k
In Script Files:		
Run current line / selection	Ctrl+Enter	Command+Enter
Run entire script file	Ctrl+Shift+Enter	Command+Shift+Enter
Tab completion	Tab	Tab
Show help for function	F1	F1

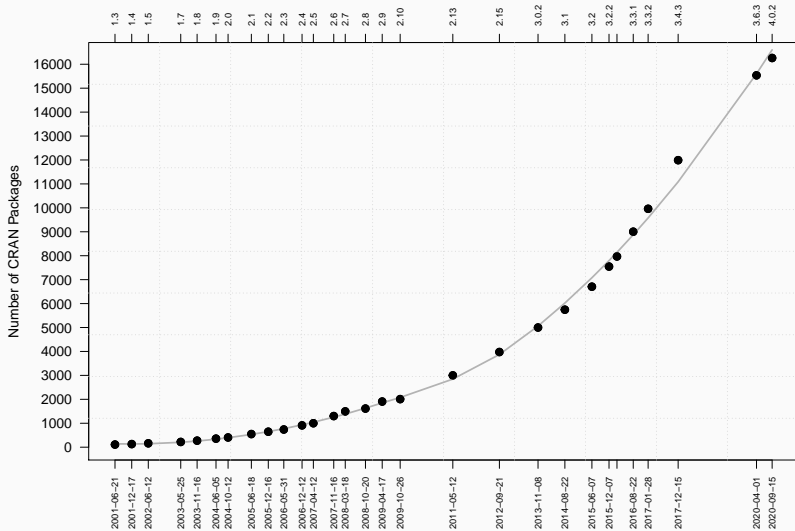
Working Directory

- suppose you have written a script file with the name `rcode.r` and saved it to some directory
- to set the “working directory”, click:
 - **Windows:** Menu File – Change Dir
 - **MacOS:** Menu Misc – Change Working Directory
 - **RStudio:** Menu Session – Set Working Directory
- alternatively:
 - `setwd()` – set the working directory
 - `getwd()` – get the current working directory

R Packages

- an extensive number of add-on “packages” have been contributed by users over the years
- one of the main strengths of R
- many statisticians have adopted R as their primary programming platform → many advanced statistical methods available in R
- [Comprehensive R Archive Network](#) (CRAN): repository for R packages – packages currently available:
> `nrow(available.packages())`
[1] 16271

R Packages



How We Will Proceed ...

- completely hands-on
- will show step-by-step how to do things
- if I go too fast, please let me know!
- if you have questions, please ask!
- keep in mind that there is a stream delay
- anybody can answer questions in the chat
- please fill out the questionnaire (see course website)
- also install all packages we will need during the course before the second day (see course website)

http://www.wvbauer.com/course_oor