Introduction to R Course

Introductory Lecture

Wolfgang Viechtbauer

Maastricht University

2022-05-28

Introduction

- Wolfgang Viechtbauer

  Department of Psychiatry and Neuropsychology

  School for Mental Health and Neuroscience

  Maastricht University, The Netherlands
- **website:** https://www.wvbauer.com/
- **email:** wvb@wvbauer.com
- **twitter:** @wviechtb

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

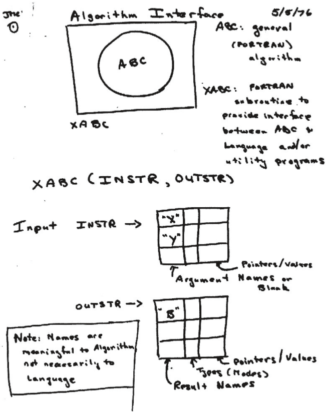
- … it began May 5, 1976 at:

Bell Laboratories, Murray Hill, New Jersey[1]

[1] Photo 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”


JMC
①
Algorithm Interface
5/5/76
ABC
ABC: general (FORTRAN) algorithm
XABC: FORTRAN subroutine to provide interface between ABC & Language and/or utility programs
XABC
XABC (INSTR, OUTSTR)
Input INSTR →
"X"
"Y"
Pointers/Values
Argument Names or Blank
OUTSTR →
"B"
Note: Names are meaningful to Algorithm not necessarily to Language
Pointers/Values
Types (Modes)
Result Names

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 Ihaka[2]

[2] Photos by Stuart Isett and Kieran Scott

Some R Milestones

- first binary of R released in 1993
- Ihaka and Gentleman (1996) publish an article about R in the *Journal of Computational and Graphical Statistics*
- source code released in 1997, CRAN is started, and the R Core Team is formed with 9 members
- version 1.0 (2000), 2.0 (2004), 3.0 (2013), 4.0 (2020)
- first useR! conference in May 2004 in Vienna, Austria
- current version: R 4.2.0 released April 2022

Other Related Developments

- Revolution Analytics founded in 2007
- 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 movements
- 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)
- a ‘functional’ and ‘object-oriented’ 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 `five` and hit Tab: `fivenum` (tada!)
- if ambiguous, can get list with possible options
- type `box` and hit Tab (twice) and 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 any text editor
- some editors have functionality for opening multiple documents, code execution, syntax highlighting, brace-matching, and other 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
- prompt: “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

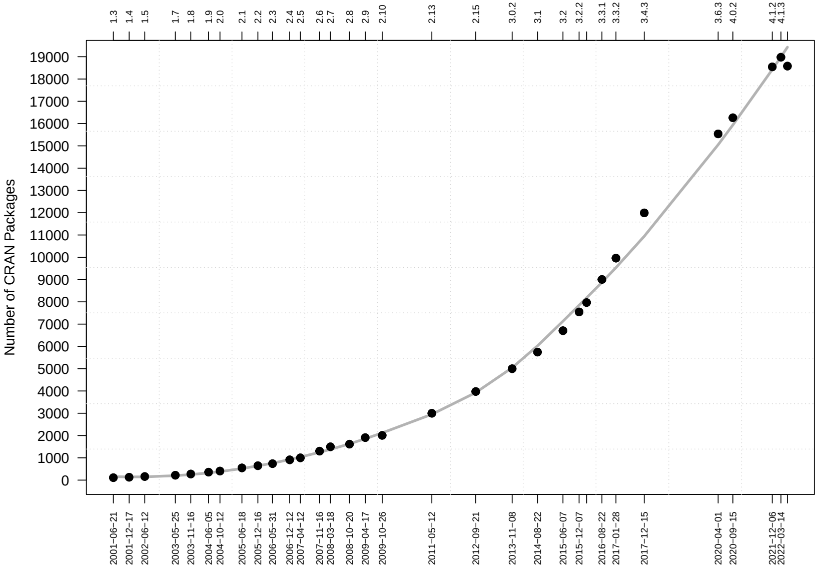
- suppose you have written a script file with the name `rcode.r` and saved it to some directory
- to set the “working directory”, click:
    - Windows: Menu File – Change Dir
    - macOS: Menu Misc – Change Working Directory
    - RStudio: Menu Session – Set Working Directory
- alternatively:
    - `setwd()`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

```
> nrow(available.packages())
[1] 18553
```

R Packages


1.3
1.4
1.5
1.7
1.8
1.9
2.0
2.1
2.2
2.3
2.4
2.5
2.6
2.7
2.8
2.9
2.10
2.13
2.15
3.0.2
3.1
3.2
3.2.2
3.3.1
3.3.2
3.4.3
3.6.3
4.0.2
4.1.2
4.1.3
19000
18000
17000
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Number of CRAN Packages
2001−06−21
2001−12−17
2002−06−12
2003−05−25
2003−11−16
2004−06−05
2004−10−12
2005−06−18
2005−12−16
2006−05−31
2006−12−12
2007−04−12
2007−11−16
2008−03−18
2008−10−20
2009−04−17
2009−10−26
2011−05−12
2012−09−21
2013−11−08
2014−08−22
2015−06−07
2015−12−07
2016−08−22
2017−01−28
2017−12−15
2020−04−01
2020−09−15
2021−12−06
2022−03−14

How We Will Proceed ...

- completely hands-on
- will show step-by-step how to do things
- if I go too fast or you have questions, let me know!