

Introduction

Utrecht University Winter School: Introduction to R



**Utrecht
University**

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

2022-01-31

Outline

Introduction Round

Open-Source Software

Open-Source Licensing

The R Statistical Programming Language

What is R?

Using R



Introduction Round

1. Name
2. Affiliation/Home base/Type of work
3. Statistical background
4. Programming background
5. (Currently) preferred statistical analysis software
6. Why do you want to learn R?
7. Favorite hobby



OPEN-SOURCE SOFTWARE



What is “Open-Source”?

R is an open-source software project, but what does that mean?

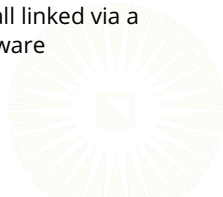
- Source code is freely available to anyone who wants it.
 - Free Speech, not necessarily Free Beer
- Anyone can edit the original source code to suit their needs.
 - Ego-less programming
- Many open source programs are also “freeware” that are available free of charge.
 - R is both open-source and freeware



Strengths of Open-Source Software

FREEDOM

- If the software you are using is broken (or just limited in capability), you can modify it in any way you like.
- If you are unsure of what the software you are using is doing, you can dig into the source code and confirm its procedures.
- If you create some software, you can easily, and independently, distribute it to the world.
 - There is a global community of potential users that are all linked via a common infrastructure that facilitates open-source software development and distribution.



Strengths of Open-Source Software

PEER REVIEW

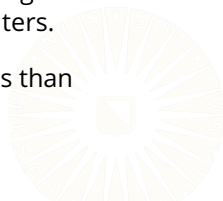
- Every user of open-source software is a reviewer of that software.
- What “bedroom programmers” lack in term of quality control procedures is overcome by the scrutiny of a large and empowered user-base.
 - When we use closed source software, we are forced to trust the honesty of the developing company.
 - We have no way of checking the actual implementation.



Strengths of Open-Source Software

ACCESSIBILITY

- Many open-source programs (like R) can be downloaded, for free, from the internet.
 - You can have R installed on all of your computers (and your mobile phone, your car's info-tainment system, your microwave, your clock-radio, ...).
 - No need to beg, borrow, or steal funds to get yourself up-and-running with a cutting-edge data analysis suite.
- Licensing legality is very simple—no worries about being sued for installing open-source software on “too many” computers.
- Open-source software tends to run on more platforms than closed-source software will.



A Note on Licensing

Some popular open-source licenses:

- The GNU General Public License (GPL)
 - <http://www.gnu.org/licenses/gpl-3.0.en.html>
- The GNU Lesser General Public License (L-GPL)
 - <http://www.gnu.org/licenses/lgpl-3.0.en.html>
- The Apache License
 - <http://www.apache.org/licenses/>
- The BSD 2-Clause License (FreeBSD License)
 - <http://opensource.org/licenses/BSD-2-Clause>
- The MIT License
 - <https://opensource.org/licenses/MIT>



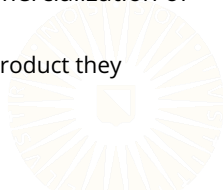
A Note on Licensing

Many open-source licenses (e.g., GPL, L-GPL) “copyleft” their products.

- Copyleft is designed to ensure that open-source software cannot be closed.
 - I can't take your copylefted software, repackage it, and sell it in violation of your original licensing terms.

Other open-source licenses (e.g., BSD-Types, Apache, MIT) are non-copyleft, “permissive” licenses.

- Many of these licenses are designed to promote commercialization of open-source products.
 - E.g., allowing a student to develop a company selling a product they developed for their dissertation



THE R STATISTICAL PROGRAMMING LANGUAGE



What is R?

R is a holistic (open-source) software system for data analysis and statistical programming.

- R is an implementation of the S language.
 - Developed by John Chambers and colleagues
 - Becker and Chambers (1984)
 - Becker, Chambers, and Wilks (1988)
 - Chambers and Hastie (1992)
 - Chambers (1998)
- Introduced by Ihaka and Gentleman (1996).
 - Currently maintained by the *R Core Team*.
- Support by thousands of world-wide contributors.
 - Anyone can contribute an R package to the *Comprehensive R Archive Network* (CRAN)
 - Must conform to the licensing and packaging requirements.



What is R?

I prefer to think about R as a *statistical programming language*, rather than as a data analysis program.

- R **IS NOT** its GUI (no matter which GUI you use).
- You can write R code in whatever program you like (e.g., RStudio, EMACS, VIM, Notepad, directly in the console/shell/command line).
- R can be used for basic (or advanced) data analysis, but its real strength is its flexible programming framework.
 - Tedious tasks can be automated.
 - Computationally demanding jobs can be run in parallel.
 - R-based research *wants* to be reproducible.
 - Analyses are automatically documented via their scripts.



Getting R

You can download R, for free, from the following webpage:

- <https://www.r-project.org/>

You will also need a proper text editor/IDE. For those who are just learning R, I recommend **RStudio**:

- <https://www.rstudio.com/>



What to Expect when Opening R

First things first, let's take a look at a few ways you can interact with R:

- Base R
- EMACS
- RStudio
- Text-only console (i.e., even more base R)



How R Works

R is an interpreted programming language.

- The commands you enter into the R *Console* are executed immediately.
- You don't need to compile your code before running it.
- In this sense, interacting with R is similar to interacting with other syntax-based statistical packages (e.g., SAS, STATA, Mplus).



How R Works

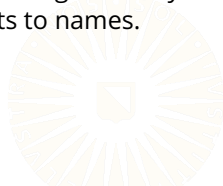
R mixes the *functional* and *object-oriented* programming paradigms.

FUNCTIONAL

- R is designed to break down problems into functions.
- Every R function is a first-class object.
- R uses pass-by-value semantics.

OBJECT-ORIENTED

- Everything in R is an object.
- R functions work by creating and modifying R objects.
- The R workflow is organized by assigning objects to names.

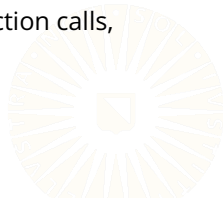


Interacting with R

When working with R, you will write *scripts* that contain all of the commands you want to execute.

- There is no “clicky-box” Tom-foolery in R.
- Your script can be run interactively or in “batch-mode”, as a self-contained program.

The primary purpose of the commands in your script will be to create and modify various objects (e.g., datasets, variables, function calls, graphical devices).



References

- Becker, R. A., & Chambers, J. M. (1984). *S: an interactive environment for data analysis and graphics*. Monterey, CA: Wadsworth and Brooks/Cole.
- Becker, R. A., Chambers, J. M., & Wilks, A. R. (1988). *The new S language*. London: Chapman & Hall.
- Chambers, J. M. (1998). *Programming with data: A guide to the S language*. New York: Springer Science & Business Media.
- Chambers, J. M., & Hastie, T. J. (1992). *Statistical models in s*. London: Chapman & Hall.
- Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3), 299–314.

