Enhancing Research Reproducibility and Collaboration with RStudio, Git, and GitHub

JP Courneya

Invalid Date

Table of contents

Preface

Welcome to the Malaria Research Program at The University of Maryland Baltimore - Center for Vaccine Development and Global Health https://www.medschool.umaryland.edu/malaria/.

These training materials are developed and made publicly available for increasing awareness of reproducible science and enhancing data and programming skills.

Acknowledgements

Git and Github lessons adapt material from:

- Happy Git with R

This is a Quarto book. To learn more about Quarto books visit https://quarto.org/docs/books.

Introduction to R and RStudio

The following chapter will provide you with a hands on opportunity to learn R and RStudio. Learning R is a big topic and we will not be able to cover everything, by the end of this session we hope that you will feel comfortable starting to use R on your own for working with Git and GitHub.

Learning Objectives

- Navigate RStudio
- Use Posit Cloud (previously known as RStudio cloud)

Why learn R?

- **R is free, open-source, and cross-platform.** Anyone can inspect the source code to see how R works. Because of this transparency, there is less chance for mistakes, and if you (or someone else) find some, you can report and fix bugs. Because R is open source and is supported by a large community of developers and users, there is a very large selection of third-party add-on packages which are freely available to extend R's native capabilities.
- **R code is great for reproducibility**. Reproducibility is when someone else (including your future self) can obtain the same results from the same dataset when using the same analysis. R integrates with other tools to generate manuscripts from your code. If you collect more data, or fix a mistake in your dataset, the figures and the statistical tests in your manuscript are updated automatically.
- **R relies on a series of written commands, not on remembering a succession of pointing and clicking.** If you want to redo your analysis because you collected more data, you don't have to remember which button you clicked in which order to obtain your results; you just have to run your script again.
- **R is interdisciplinary and extensible** With 10,000+ packages that can be installed to extend its capabilities, R provides a framework that allows you to combine statistical approaches from many scientific disciplines to best suit the analytical framework you

need to analyze your data. For instance, R has packages for image analysis, GIS, time series, population genetics, and a lot more.

- **R works on data of all shapes and sizes.** The skills you learn with R scale easily with the size of your dataset. Whether your dataset has hundreds or millions of lines, it won't make much difference to you. R is designed for data analysis. It comes with special data structures and data types that make handling of missing data and statistical factors convenient. R can connect to spreadsheets, databases, and many other data formats, on your computer or on the web.
- **R produces high-quality graphics.** The plotting functionalities in R are endless, and allow you to adjust any aspect of your graph to convey most effectively the message from your data.
- **R has a large and welcoming community.** Thousands of people use R daily. Many of them are willing to help you through mailing lists and websites such as Stack Overflow, or on the RStudio community. Questions which are backed up with short, reproducible code snippets are more likely to attract knowledgeable responses.

Starting out in R

R is both a programming language and an interactive environment for data exploration and statistics.

Working with R is primarily text-based. The basic mode of use for R is that the user provides commands in the R language and then R computes and displays the result.

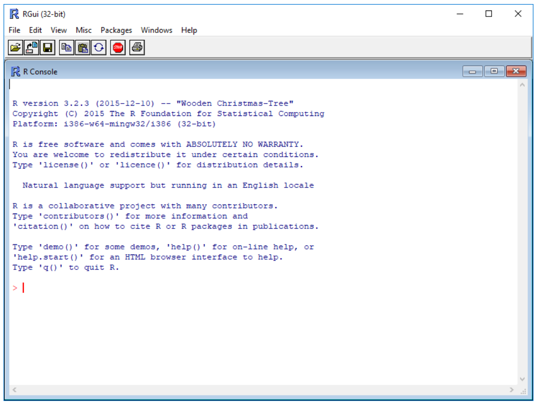
Downloading, Installing and Running R

Download
R can be downloaded from CRAN (The Comprehensive R Archive Network) for Windows, Linux, or Mac.

Install
Installation of R is like most software packages and you will be guided. Should you have any issues or need help you can refer to R Installation and Administration

Running
R can be launched from your software or applications launcher or When working at a command line on UNIX or Windows, the command `R` can be used for starting the main R program in the form `R`

You will see a console similar to this appear:



```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

While it is possible to work solely through the console or using a command line interface, the ideal environment to work in R is RStudio.

RStudio

We will be working in RStudio. The easiest way to get started is to go to RStudio Cloud and create a new project.

The main way of working with R is the *console*, where you enter commands and view results. RStudio surrounds this with various conveniences. In addition to the console panel, RStudio provides panels containing:

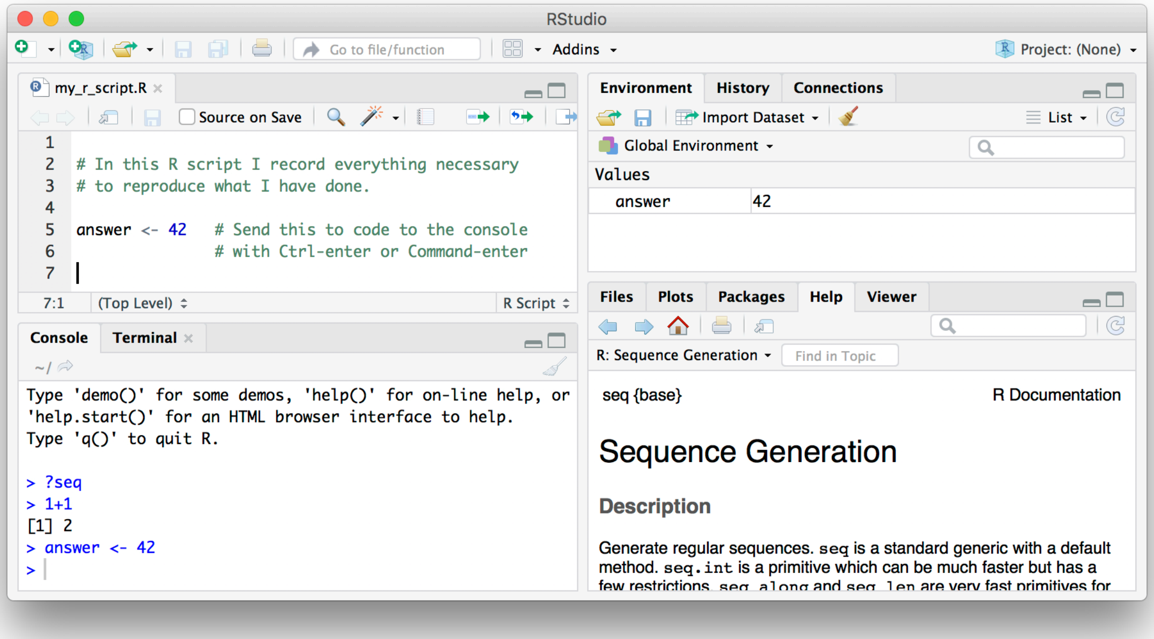
Studio is divided into four "panes". The placement of these panes and their content can be customized (see menu, Tools -> Global Options -> Pane Layout).

The Default Layout is:

- Top Left - **Source**: your scripts and documents
- Bottom Left - **Console**: what R would look and be like without RStudio
- Top Right - **Environment/History**: look here to see what you have done
- Bottom Right - **Files** and more: see the contents of the project/working directory here, like your Script.R file

RStudio Cloud

RStudio Cloud is a browser-based version of RStudio. It will allow you to use RStudio without needing to download anything to your computer. You can also easily share your R projects with others. While we recommend downloading RStudio for regular use, we will be using RStudio Cloud for these workshops so we can easily share files and packages with you.



```

# In this R script I record everything necessary
# to reproduce what I have done.

answer <- 42   # Send this to code to the console
               # with Ctrl-enter or Command-enter

```

Values

answer 42

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?seq
> 1+1
[1] 2
> answer <- 42
> 
```

seq {base} R Documentation

Sequence Generation

Description

Generate regular sequences. `seq` is a standard generic with a default method. `seq.int` is a primitive which can be much faster but has a few restrictions. `seq_along` and `seq_len` are very fast primitives for

Using this book

For these instructions code will appear in the gray box as follows:

```
fake code
```

To run the code you can copy and paste the code and run it in your RStudio session console at the prompt > which looks like a greater than symbol.

```
> fake code
```

The code can also be added to an R Script to be run.

When the code is run in RStudio the console prints out results like so:

```
[1] Result
```

In this tutorial results from code will appear like so:

```
## [1] Result
```

Working in the Console

The console is an interactive environment for RStudio, click on the “Console” pane, type `3 + 3` and press enter. R displays the result of the calculation.

```
3 + 3
```

```
[1] 6
```

`+` is called an operator. R has the operators you would expect for for basic mathematics:

Arithmetic operators

operator	meaning
+	plus
-	minus
*	times
/	divided by
^	exponent

Logical Operators

operator	meaning
==	exactly equal
!=	not equal to
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
x|y	x or y
x&y	x and y
!x	not x

Spaces can be used to make code easier to read.

```
2 * 2 == 4
```

```
[1] TRUE
```

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```r
1 + 1
```

```
[1] 2
```

2 Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```

References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.