

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

JP Courneya

May, 2023

# Table of contents

<b>Preface</b>	<b>3</b>
Acknowledgements . . . . .	3
<b>1 Introduction to R and RStudio</b>	<b>4</b>
1.1 Learning Objectives . . . . .	4
1.2 Why learn R? . . . . .	4
1.3 Starting out in R . . . . .	5
1.3.1 Downloading, Installing and Running R . . . . .	5
1.3.2 RStudio . . . . .	6
1.3.3 Posit Cloud (formerly RStudio Cloud) . . . . .	6
1.4 Using this book . . . . .	7
1.5 Working in the Console . . . . .	8
1.6 Working in the Terminal . . . . .	9
<b>2 Introduction</b>	<b>10</b>
<b>3 Summary</b>	<b>11</b>
<b>References</b>	<b>12</b>

# 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.

mrp-bioinformatics/MRP\_git\_training is licensed under the Creative Commons Zero v1.0 Universal

## 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>.

# 1 Introduction to R and RStudio

The following chapter will provide you with a hands on opportunity to familiarize yourself with RStudio. Learning RStudio 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.

## 1.1 Learning Objectives

- Navigate RStudio
- Use Posit Cloud (formerly RStudio Cloud)

## 1.2 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.

## 1.3 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.

### 1.3.1 Downloading, Installing and Running R

#### Download

R can be downloaded from [CRAN \(The Comprehensive R Archive Network\)](#) for Windows, Linux, or Mac.

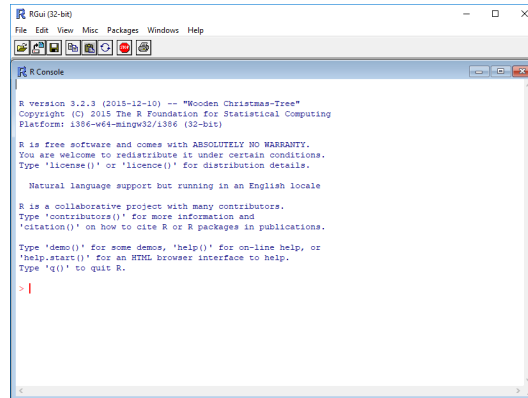
#### Install

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:



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.

### 1.3.2 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:

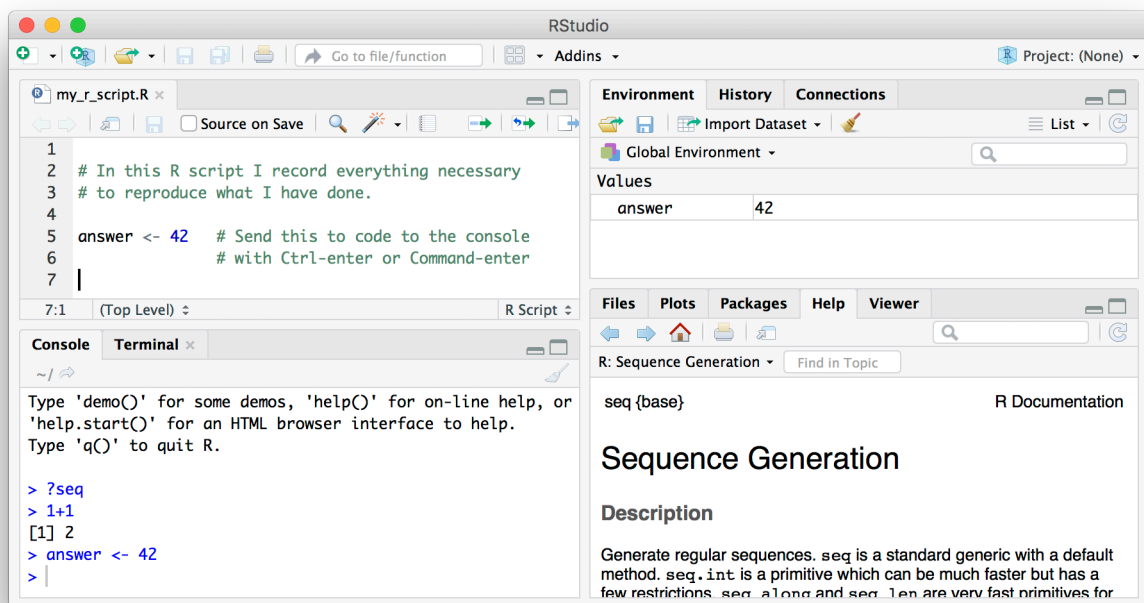
Studio is divided into four “panes”. The placement of these panes and their content can be customized (see menu, Tools -> Global Options -> Pane Layout).

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

### 1.3.3 Posit Cloud (formerly RStudio Cloud)

Posit 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. To use Posit Cloud a user account is required. While we recommend downloading RStudio for routine R programming use, we will be using RStudio Cloud for this training so we can easily share files and packages with you.



To access Posit Cloud proceed to <https://posit.cloud/> in a new browser window or tab.

## 1.4 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
```

## 1.5 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 basic mathematics:

### Arithmetic operators

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

### Logical Operators

operator	meaning
<code>==</code>	exactly equal
<code>!=</code>	not equal to
<code>&lt;</code>	less than
<code>&lt;=</code>	less than or equal to
<code>&gt;</code>	greater than
<code>&gt;=</code>	greater than or equal to
<code>x y</code>	x or y
<code>x&amp;y</code>	x and y
<code>!x</code>	not x

Spaces can be used to make code easier to read.



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

```
[1] TRUE
```

You can also run commands in the console for working with your computer's filesystem.

```
getwd() # similar to UNIX PWD
```

## 1.6 Working in the Terminal

The embedded Terminal in RStudio is a command-line interface within the IDE, allowing users to execute system commands and interact with the operating system directly. \* It shares the same working directory as the RStudio session and supports various commands for file management, package installation, and more. \* Integration into the IDE streamlines workflows by eliminating the need to switch between applications, enhancing productivity and enabling seamless interaction between R programming and system administration tasks.

## 2 Introduction

This is a book created from markdown and executable code.

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

```
1 + 1
```

```
[1] 2
```

## 3 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>.