

Getting started

The 01-getting-started.pdf file gives step-by-step instructions for installing R and RStudio. Where installation links are provided, right click on the link to go to the download site. Instructions for obtaining additional files and packages required to get up and running for the course are also included here. You can also go to the “Getting started” chapter in the bookdown version of this file found here: [2024_IntroR_and_RStudio](#).

Installing R

The first step is to install R. You can download and install R from the [Comprehensive R Archive Network \(CRAN\)](#), for MacOS [here](#) and for Windows [here](#).

Note: if you have a Mac M1 arm64 computer, DON'T install the R-4.4.1-arm64.pkg, instead install R-4.4.1-x86_64.pkg for Intel because the arm64 version doesn't support the Bioconductor packages that we will be using.

For MacOS, the next things you need to install are XQuartz and fortran tools.

Download and install [XQuartz](#) (version 2.8.5 or later)

The download link for fortran tools:

[gfortran-12.2-universal.pkg](#)

If you are on a Mac, you also need to install XCode. Open the terminal app (in the Finder under the GO dropdown click Utilities and in that folder click on 'Terminal.app') and then type:

```
sudo xcode-select --install
```

There is dash between xcode and select follow by a space and then a double dash before the word install which may not be visible in the pdf file. This installs Xcode command line tools which are sufficient to build R.

Installing RStudio

The next step is to install RStudio, instructions are here:

[RStudio](#)

RStudio is a typical integrated development environment (IDE) in that it provides additional tools to make the process of developing scripts easier. It has a graphical user interface (GUI) that displays things without necessarily having to enter lines of commands.

Additional course material: Cloning from Github

What is Github? GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. It's fairly sophisticated and we are not going to learn a lot about it here but it is well worth your time to learn it if you don't know about it already. A good introduction to using Git with R can be found [here](#). For this course we will use github primarily just to “clone” course material from a “repository” onto your local machine.

There are three course repositories for the course that you need to clone. To clone the first one so that you get your own copy in Rstudio, create a new RStudio Project.

Be intentional about where you create this Project. You should make a folder where you keep all of your RProjects, for example, your home directory /Users/your username/Documents/RProjects is a good one.

Start RStudio, then go to:

File > New Project > Version Control > Git. In the “repository URL” paste the URL of your new GitHub repository:

https://github.com/gurinina/2024_IntroR_and_RStudio

You don't need to put anything in the Project directory name field.

In the Create as a sub-directory field navigate to your RProjects folder.

Hit Create Project and your 2024_IntroR_and_RStudio project will be generated.

If you have already created the 2024_IntroR_and_RStudio project but have closed RStudio, you can open it by starting Rstudio, going to Open Project in the File menu, navigating to the 2024_IntroR_and_RStudio folder and opening it. Alternatively, you should be able to find it in the Recent Projects list on the RStudio start page.

Running R code; installing packages

The first R command we will run is `install.packages`. R only includes a basic set of functions. It can do much more than this, and many of these functions are stored in the [CRAN](#). Note that these packages are vetted: they are checked for common errors and they must have a dedicated maintainer. You can easily install packages from within R if you know the name of the packages. As an example, we are going to install the three required packages.

You'll need to install `bookdown`, `knitr` and `rmarkdown` packages to compile the 01-getting-started.Rmd file in R. R Markdown is a file format for making dynamic documents with R. An R Markdown document is written in markdown (an easy-to-write plain text format) and contains chunks of embedded R code, like the ones below. The notation at the beginning of each chunk gives specific instructions for the chunk. For example, “include = FALSE” in the

following chunk prevents code from appearing in the finished file and “eval = FALSE” tells R not to run that chunk. Chunks need to be inserted using three single backticks (below the ‘~’ on your keyboard to the left of number 1) followed by {r} in closed parentheses at the beginning of the chunk and three more backticks at the end of a chunk. You can insert a chunk automatically by clicking on the insert chunk at the top of an Rmd document (click the green +C square icon and choose R in the dropdown menu). To create a document (usually HTML or PDF) from rmarkdown, you need to Knit the document using the ‘knitr’ package. Knitting a document simply means taking all the text and code and creating a nicely formatted document in either HTML, PDF, or Word. ‘bookdown’ allows you to compile files into a book, like the ones we’ll be using in class, available on my bookdown page at [2024_IntroR_and_RStudio](#).

Open the 01-getting started.Rmd in the lessons folder and navigate to the code below. Copy and paste the following chunk outputs into your console (bottom left corner of RStudio) without the header and footer backticks.

```
# type the following into your console or run the chunk in the .Rmd file if you  
# managed to open the document in R, after setting eval to TRUE, by clicking on  
# the little green arrow next to the chunk.  
  
install.packages(c("rmarkdown", "bookdown", "knitr", "devtools",  
"rsconnect"))
```

To load the packages after installation copy and paste the following chunk output into your console again without the header and footer backticks:

```
library(bookdown)  
library(knitr)  
library(rmarkdown)  
library(devtools)  
  
## Loading required package: usethis  
  
library(rsconnect)  
  
##  
## Attaching package: 'rsconnect'  
  
## The following object is masked from 'package:devtools':  
##  
##      lint
```

Once these packages have been installed and loaded, you should be able to run the .Rmd version of this file. So you can switch from following the 01-getting-started.html file to following the 01-getting-started.Rmd file. In this file, you run chunks of R code by clicking the little green arrow next to the chunk or by clicking the Run button at the top of the script window and selecting “run current chunk” in the dropdown menu. If want to keep using the

.html file, just copy and paste the chunk outputs (light yellow bits) into your console (bottom left corner of RStudio).

Note that eval is often set to FALSE in my working copy of the .Rmd files. This is usually because I have already run these commands once and don't need to run them again, nor should you after you have run them once (it's time consuming). Typically, eval = FALSE is set for chunks that install packages that have already been installed.

Packages can also be installed from [Bioconductor](#), another repository of packages which provides tools for the analysis and comprehension of high-throughput **genomic data**. We'll use Bioconductor most of the time when installing packages.

To install from Bioconductor, you first need to install BiocManager. **This only needs to be done once ever for your R installation.** We will be installing several packages from Bioconductor using BiocManager (see below).

Set eval = TRUE if you haven't yet installed BiocManager.

```
install.packages("BiocManager")
```

For the other packages you'll need, run the following (you should set eval = TRUE):

```
devtools::install_github("gurinina/GOenrichment", force = TRUE)

BiocManager::install(c("gplots", "tidyverse", "RColorBrewer", "DESeq2",
  "pheatmap",
  "ggplot2", "ggrepel", "org.Hs.eg.db", "clusterProfiler", "enrichplot",
  "fgsea ",
  "purrr"))
```

Running R code; compiling the book

Optional Once the rmarkdown, bookdown and knitr packages have been installed and the libraries loaded, you can try running the following command **in your console**:

There are two ways to render the book. The first is to run the following command in the console. The second is to run the render_all.R file in the 2024_IntroR_and_RStudio folder. You can run this by opening the file in RStudio and clicking the "source" button at the top of the script window.

Setting eval = FALSE tells R not to run the code in the chunk.

```
# DO NOT RUN THIS IN RMARKDOWN, DO NOT SET EVAL = TRUE, COPY AND PASTE JUST THE
# RENDER_BOOK COMMAND INTO YOUR CONSOLE AND RUN IT THERE -- ELSE IT WILL
# TRIGGER AN ENDLESS LOOP
render_book(input = "index.Rmd", output_format = "bookdown::gitbook")
```

If it runs successfully, you will have a “_book” folder that has appeared in your working directory. Open it and then click on “index.html”. This will open up in your browser and then you should have your own version of the ‘2024_IntroR_and_RStudio’ book.

If you now open the index.Rmd file in RStudio and click on the “Knit” button at the top of the script window, you should be able to compile the book in your viewer pane, which makes it easier to navigate for the work we will be doing in class.

Optional You can publish this book on bookdown.org. It requires setting up an account and then running the following command:

```
# DO NOT RUN THIS IN RMARKDOWN, COPY AND PASTE publish_book() INTO YOUR  
CONSOLE  
# AND RUN IT RUN IT THERE  
  
publish_book()
```

R will tell you: You do not currently have a bookdown.org publishing account configured on this system. Would you like to configure one now? [Y/n]: Type ‘y’ and hit enter.

A browser window should open to complete authentication.

Learn R Basics

In this book we will be using the **R programming language** for all our analysis. You will be introduced to R in the first lab. However, if you have no basic programming skills and no knowledge of R, your first homework assignment is to complete an R tutorial to familiarize yourself with the basics of programming and R syntax.

There are two approaches here. The first is to go through the [swirl](#) tutorial, which teaches you R programming and data science interactively, at your own pace and in the R console. Once you have R installed, you can install `swirl` and run it the following way:

(note this chunk won’t run unless you change `eval` to `TRUE`)

```
install.packages("swirl")  
library(swirl)  
swirl()
```

It will provide prompts and ask you what you want to learn. Choose R Programming, then choose any of these lessons except 10, 11, 13-15.

- 1: Basic Building Blocks
- 2: Workspace and Files
- 3: Sequences of Numbers
- 4: Vectors
- 5: Missing Values
- 6: Subsetting Vectors
- 7: Matrices and Data Frames

8: Logic

9: Functions 10: lapply and sapply DON'T choose this 11: vapply and tapply DON'T choose this 12: Looking at Data 13: Simulation DON'T choose this 14: Dates and Times DON'T choose this 15: Base Graphics DON'T choose this

The second approach is to start going through the lectures, which are self-standing. For that I would go to the webpage hosting the book and start going through the lessons. You can find the book version of the introduction to R here: [2024_IntroR_and_RStudio](#). Going through the initial lessons will be very really helpful for you, as we will dive into things pretty quickly and are not likely to get through all of the 2024_IntroR_and_RStudio sections in class. I recommend the ggplot2 lecture in particular after going through some of the basics.

There are also many open and free resources and reference guides for R that will be helpful to you. Several examples are listed here:

- [Quick-R](#): a quick online reference for data input, basic statistics and plots.
- [R reference card \(PDF\)](#) by Tom Short.
- The best reference for rmarkdown documents is [R Markdown Cookbook](#).
- There is also an excellent introductory R book for graduate students here: [Guide-to-R-Book](#).

Getting help

Three key things you need to know about R is that you can get help for a function by typing the function into the help menu, a tab in the Files pane (lower right corner of RStudio) or by using help or ?, like this:

```
help("install.packages")
`?`(install.packages)
```

Make sure and put the function of interest in quotes when using the help function above and the example function below.

To see examples for function usage you can type example followed the function of interest in quotes. Here we use the function sum as an example, which returns the sum of all the values present in its arguments:

```
example("sum")

##
## sum> ## Pass a vector to sum, and it will add the elements together.
## sum> sum(1:5)
## [1] 15
##
## sum> ## Pass several numbers to sum, and it also adds the elements.
```

```
## sum> sum(1, 2, 3, 4, 5)
## [1] 15
##
## sum> ## In fact, you can pass vectors into several arguments, and
everything gets added.
## sum> sum(1:2, 3:5)
## [1] 15
##
## sum> ## If there are missing values, the sum is unknown, i.e., also
missing, ....
## sum> sum(1:5, NA)
## [1] NA
##
## sum> ## ... unless we exclude missing values explicitly:
## sum> sum(1:5, NA, na.rm = TRUE)
## [1] 15
```

To get the arguments of a function such as sum type:

```
args(sum)

## function (... , na.rm = FALSE)
## NULL
```