Getting Started with RStudio Desktop

An optional guide to offlining your work for PH142

Gene Ho

Summer 2019

Contents

Background

As great as RStudio Cloud is, we often find ourselves patiently waiting for the workspace to load–if it ever does at all. One way to combat this is to utilize RStudio Desktop to offline our assignments and projects. In addition to loading much faster–as it is limited by the specs of your computer–you'll be able to access your files when your internet goes out at 2AM or in the jungles of South America, as long as you have your computer (and that it's charged).

This guide is especially a good idea for people (*cough cough Epi/Biostats cough cough*) who will be needing to constantly be working with data and data sets.

The biggest downside to RStudio Desktop over RStudio Cloud is that you can't access R convieniently from your browser.

Lastly, there's a decent amount of technical detail in this document. Don't worry if you don't understand it all. This document is meant to get you set up in RStudio. Feel free to read as much or as little as you would like.

Instructions

The installation instructions for R and RStudio Desktop is mostly the same between Windows and macOS. The installation of LaTeX (a program used to knit to PDF) is slightly different as the installation files come from different sources. Because of this, there will be a fork in instructions after the successful installation of RStudio Desktop.

Part I: Downloading & Installing RStudio Desktop

1. Download both the base R package installer [https://cran.cnr.berkeley.edu/] and RStudio Desktop [https://www.rstudio.com/products/rstudio/download/#download]. Filetype should be pkg and dmg for macOS; and exe for Windows
2. Install the R package first (at the time of writing is called R-3.6.1)
3. Install RStudio Desktop
4. Double-click RStudio Desktop to run it.

Part II: Installing RStudio Desktop Dependencies

When opening RStudio Desktop for the first time, you may be asked to install additional software. Depending on what programs you have, some, none or even all of this part may apply to you.

Between each step, allow the installation to complete before proceeding to the next step. Check the console for movement/scrolling–sometimes it takes a while for it to complete. You will know the task as completed/failed once you see the blue ">"

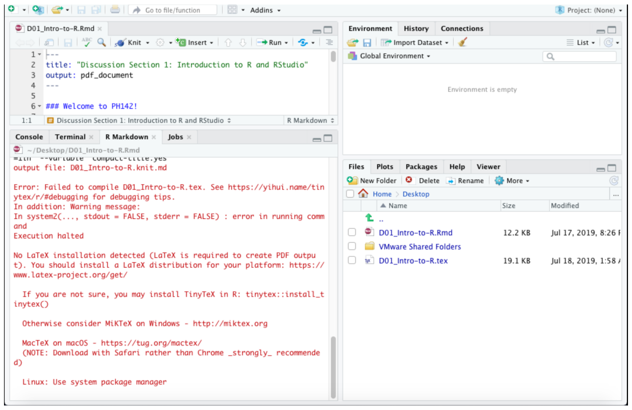
1. A popup may appear asking you to install git. Click *Install*. It is not necessary to install of Xcode.
2. Near the top of the Source (top left box), there should be a bar that says that the knitr package is not installed. Click Install.
3. Download an existing RMD file (I suggest one of the labs) from bCourses or RStudio Cloud (instructions on how to download from RStudio Cloud below).
4. Click Knit to PDF.
5. If this is successful, proceed to Part IV to install libraries. If not, continue to Part III.

Part III: Installing LaTeX

In order to Knit to PDF, you will need a LaTeX package installed. LaTeX is a document preparation system similar to Microsoft Word and Apple Pages. It is what RStudio uses to generate text in PDF documents.

1. In the console, type `tinytex::install_tinytex()`
2. It will take some time to run, if the installation fails. Install MikTex (Windows) [https://miktex.org/] or MacTex (macOS) [https://tug.org/mactex/]. **Alternatively *on macOS only*** If you get the error:

   add_link_dir_dir: destination /usr/local/bin not writable, no links from /Users/ph142-test/Library/TinyTeX/bin/x86_64-darwin.
   tlmgr: An error has occurred. See above messages. Exiting.
   add of symlinks had 1 error(s), see messages above.



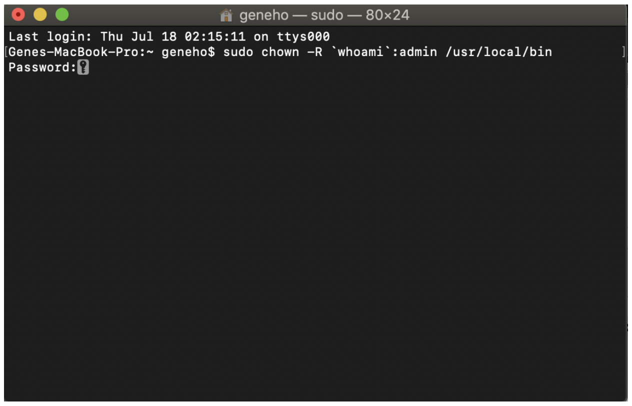
```
---
title: "Discussion Section 1: Introduction to R and RStudio"
output: pdf_document
---

### Welcome to PH142!
```

```
output file: D01_Intro-to-R.knit.md

Error: Failed to compile D01_Intro-to-R.tex. See https://yihui.name/tin
ytex/r/#debugging for debugging tips.
In addition: Warning message:
In system2(..., stdout = FALSE, stderr = FALSE) : error in running comm
and
Execution halted

No LaTeX installation detected (LaTeX is required to create PDF outpu
t). You should install a LaTeX distribution for your platform: https://
www.latex-project.org/get/

  If you are not sure, you may install TinyTeX in R: tinytex::install_t
inytex()

  Otherwise consider MiKTeX on Windows - http://miktex.org

  MacTeX on macOS - https://tug.org/mactex/
  (NOTE: Download with Safari rather than Chrome _strongly_ recommende
d)

  Linux: Use system package manager
```

Name	Size	Modified
D01_Intro-to-R.Rmd	12.2 KB	Jul 17, 2019, 8:26
VMware Shared Folders		
D01_Intro-to-R.tex	19.1 KB	Jul 18, 2019, 1:58

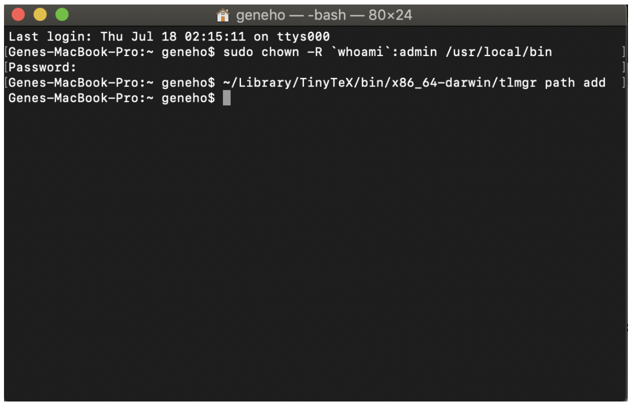
Or if you don't want to install the full TinyTex program…

3. Open **Terminal** (Applications/Utilities/Terminal or CMD+Space -> "Terminal").
4. Type `sudo chown -R `whoami`:admin /usr/local/bin` + ENTER.
5. Enter in the password you use to log into your computer + ENTER. No characters will appear, but your typing is being registered.



```
Last login: Thu Jul 18 02:15:11 on ttys000
Genes-MacBook-Pro:~ geneho$ sudo chown -R `whoami`:admin /usr/local/bin
Password:
```

6. Type `~/Library/TinyTeX/bin/x86_64-darwin/tlmgr path add` + ENTER.



```
Last login: Thu Jul 18 02:15:11 on ttys000
Genes-MacBook-Pro:~ geneho$ sudo chown -R `whoami`:admin /usr/local/bin
Password:
Genes-MacBook-Pro:~ geneho$ ~/Library/TinyTeX/bin/x86_64-darwin/tlmgr path add
Genes-MacBook-Pro:~ geneho$
```

7. Rerun `tinytex::install_tinytex()` in the **R Console**.
8. Now, TinyTex *should* successfully install.

Libraries and Packages

At the beginning of this course, we learned about the relationships between packages, libraries, and functions. "Packages" are essentially what you download, once saved in RStudio, they become known as "libraries" (e.g. `dplyr`, `ggplot2`) and its dependencies (other packages that are relied on). Libraries house functions (e.g. `head()`, `mutate()`) that you can call upon to calculate or interact with your data.

Installing Packages

You only need to do this once for each package, as it saves globally within the RStudio program.

To download and install packages, you would need to run the `install.packages()` function. `install.packages` only requires one argument (what goes inside the parenteses), but you can streamline installation by separating each package with commas.

```r
# to run this code chunk, remove "eval = F" above
# this chunk installs common packages used in this course
install.packages("dplyr")
install.packages("ggplot2")
install.packages("tidyverse")
install.packages("broom")

# code below runs the same as the lines above
install.packages("dplyr", "ggplot2", "tidyverse", "broom")
```

Calling Libraries

You must do this for every single document. Traditionally, the very first code chunk calls all the libraries relevant to the document. This makes it easier for other people to view necessary libraries and install them if necessary. However, you can call the library anytime before calling functions from that library.

```r
# to run this code chunk, remove "eval = F" above
library(dplyr)
library(ggplot2)
library(tidyverse)
# ... etc
```

Importing Data

The exact process of importing data is slightly different for each file type (e.g. .csv, .xls, .json). However, once you find the appropriate package, it should be fairly simple to import. This section will briefly cover the two most common file types, csv and xlsx. For further details, check out Malvika's other document on importing datasets.

Notes About Pathnames

When working in RStudio, you have the option of creating "projects" or .Rproj files. R Project files can help organize your files and pathnames (aka file location). We **highly** suggest making use of R Project files, especially since it simplifies pathnames.

If you're running Windows, your pathname will usually look like this "C:\..." If you're running macOS, your pathname will look like this "/Users/geneho/Desktop..." or "~\Desktop". The tilde is a shortcut to your "home folder". A quick way to get the pathname is to right-click on the file, hold down Option and click "Copy...as Pathname"

If you're utilizing R Project files, you can save the headache of having to figure out colons and backslashes. Any files and folders that "live" in the same folder as the .rproj can use "/../" as the prefix to the folder path. For example, say my folders look like this

- geneho
  - Project Folder
    - MY_PROJECT.rproj
    - just_an_example.rmd
    - my_dataset.csv
    - Folder I Don't Need
      - more_data.xlsx

On a Mac, the pathname to `my_dataset.csv` might look like this "/Users/geneho/Project Folder/my_dataset.csv" or "~/Project Folder/my_dataset.csv". However, with our R project file, we can truncate it to "../my_dataset.csv" or if we need `more_data.xlsx` we could use "../Folder I Don't Need/more_data.xlsx".

Comma Separated Values

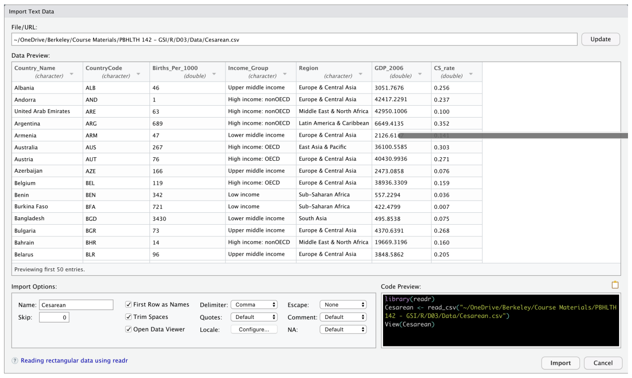
CSV files are one of the simplest file types that can hold data sets. Essentially, it has data that is usually separated by–you guessed it!–commas. Normally you can view these in a simple text editor or Excel.

```r
library(readr)
my_data <- read_csv("../Data/InfantData.csv")
```

Excel Files

```r
library(readxl)
my_data <- read_xlsx("../Data/sleep.xlsx")
```

Alternatively, you can left-click on the data file and select "Import Dataset..." It'll give you a preview of the dataset along with import options on the bottom left and a code preview on the bottom right. It is a good idea to copy and paste the code from the preview popup so that you can paste it into a code chunk. This method will only import data via the Console.



~/OneDrive/Berkeley/Course Materials/PBHLTH 142 – GSI/R/D03/Data/Cesarean.csv

Country_Name (character)	CountryCode (character)	Births_Per_1000 (double)	Income_Group (character)	Region (character)	GDP_2006 (double)	CS_rate (double)
Albania	ALB	46	Upper middle income	Europe & Central Asia	3051.7676	0.256
Andorra	AND	1	High income: nonOECD	Europe & Central Asia	42417.2291	0.237
United Arab Emirates	ARE	63	High income: nonOECD	Middle East & North Africa	42950.1006	0.100
Argentina	ARG	689	High income: nonOECD	Latin America & Caribbean	6649.4135	0.352
Armenia	ARM	47	Lower middle income	Europe & Central Asia	2126.6167	0.141
Australia	AUS	267	High income: OECD	East Asia & Pacific	36100.5585	0.303
Austria	AUT	76	High income: OECD	Europe & Central Asia	40430.9936	0.271
Azerbaijan	AZE	166	Upper middle income	Europe & Central Asia	2473.0858	0.076
Belgium	BEL	119	High income: OECD	Europe & Central Asia	38936.3309	0.159
Benin	BEN	342	Low income	Sub-Saharan Africa	557.2294	0.036
Burkina Faso	BFA	721	Low income	Sub-Saharan Africa	422.4799	0.007
Bangladesh	BGD	3430	Lower middle income	South Asia	495.8538	0.075
Bulgaria	BGR	73	Upper middle income	Europe & Central Asia	4370.6391	0.268
Bahrain	BHR	14	High income: nonOECD	Middle East & North Africa	19669.3196	0.160
Belarus	BLR	96	Upper middle income	Europe & Central Asia	3848.5862	0.205

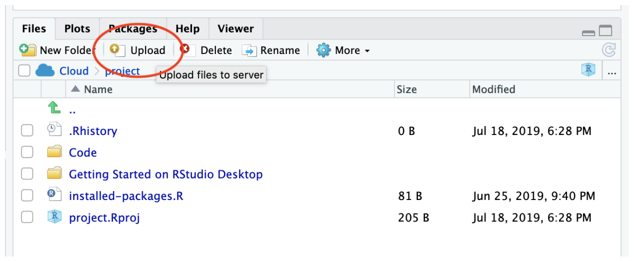
Previewing first 50 entries.

Name: Cesarean
Skip: 0
✓ First Row as Names
✓ Trim Spaces
✓ Open Data Viewer
Delimiter: Comma
Quotes: Default
Locale: Configure...
Escape: None
Comment: Default
NA: Default

```r
library(readr)
Cesarean <- read_csv("~/OneDrive/Berkeley/Course Materials/PBHLTH
142 - GSI/R/D03/Data/Cesarean.csv")
View(Cesarean)
```

Transferring Your Work from RStudio Desktop to RStudio Cloud

For the Data Skills Demonstration, we ask you to upload your code and dataset to RStudio Cloud. If all of your work is in one folder, transferring should be quick.



Name	Size	Modified
..		
.Rhistory	0 B	Jul 18, 2019, 6:28 PM
Code		
Getting Started on RStudio Desktop		
installed-packages.R	81 B	Jun 25, 2019, 9:40 PM
project.Rproj	205 B	Jul 18, 2019, 6:28 PM

All of your files are in one folder

1. Right-click on your folder and compress to a ZIP file
    a. Windows: Send to > Compressed (zipped) folder
    b. macOS: "Compress [folder]"
2. Open up RStudio Cloud and navigate to the appropriate assignment (e.g. Data-skills-demonstration)
3. In the Files corner (lower right), click "Upload"
4. Upload the ZIP folder you just created
5. Organize folders as necessary

Your files are everywhere

1. Open up RStudio Cloud and navigate to the appropriate assignment (e.g. Data-skills-demonstration)
2. In the Files corner (lower right), click "Upload"
3. Upload each file individually into RStudio Cloud