Introduction to R

Donatello, Roualdes, Lytal, Guo

2025-01-01

Table of contents

Preface & Course Overview

This notebook was created to host learning materials for Math 130: Introduction to R at California State University, Chico.

Math 130 is designed as a short-course "primer" to be taken before, or concurrently with, an upper division or graduate level Statistics course such as Math 315 or Math 615.

The goal of Math 130 is to get the complete novice up and running with the basic knowledge of how to use the statistical programming language R in an environment that emphasizes reproducible research and literate programming for data analysis. We recognize there are many topics and many approaches to teach an introduction to the R programming language. We have picked specific topics and methods that we believe learners will need to succeed in subsequent classes.

The target audience is anyone who wants to do their own data analysis. The course will culminate with an exploratory data analysis on either a pre-specified data set or your data set of choice.

Logistics

This course may be offered as in person or (a)synchronous online. Regardless of the mode of instruction, all learning materials are found on this website. Class time (when applicable) is spent expanding on ideas and concepts, working through assignments as a class or in pairs.

The logistics of the course varies slightly by instructor and mode of instruction. See below for your specific section.

Syllabus for current term

- HTML webpage
- Downloadable PDF

  move these to sub folder

- Fall 2025 Dr. Donatello - In person

where to put discord & common help?

Navigating this notebook

Throughout this notebook you will see various colored callout boxes, weirdly highlighted text, and emojis. Here is what each of those mean.

> **blue boxes**
>
> Blue boxes are simple notes or pieces of information for you to consider. Examples include lesson learning objectives.

> **yellow boxes**
>
> means caution

> **green box**
>
> is a tip or an example

> **red box**
>
> Red boxes are warnings or advice on what *not* to do.

- :cinema: associated video
- :pencil2: You try it
- :x: Don't do this thing

Motivation - Tool choices

R + R Studio = Success

- The term **R** refers to both the programming language and the software that interprets the scripts written using it.
- **RStudio** is currently a very popular way to not only write your R scripts but also to interact with the R software.
- We will be programming in the R language, using the R Studio platform. You will have to install both onto your computer. Setup instructions are discussed later in section `FIXME`

Why use R?

- Open source, cross-platform, and free
- Great for reproducibility
- Flexible and extensible. Doesn't do something you want? Create a custom function for yourself.
- Tons of learning resources
- Currently R is used in all of Chico's upper division Statistics & Data Science courses, as well as in some Science, Public Health, Economics, Finance, and some graduate courses.
- Does not involve lots of pointing and clicking (that's a good thing!)
- Works on data of all shapes and sizes
- Produces high-quality graphics
- Large and welcoming community

Why use R Studio?

- Customizable workspace that docks all windows together.
- Notebook formats that allow for easy sharing of code and output, and integration with other languages (Python, C++, SQL, Stan)
- Syntax highlighting, warning errors when missing a closing parentheses.
- Cross-platform interface. Also works on Windows/iOS/Linux.
- Tab completion for functions. Forget the syntax or a variable name? Popup helpers are available.
- Free training videos available from the developers directly.
- One button publishing of reproducible documents such as reports, interactive visualizations, presentations (like this one!), websites.

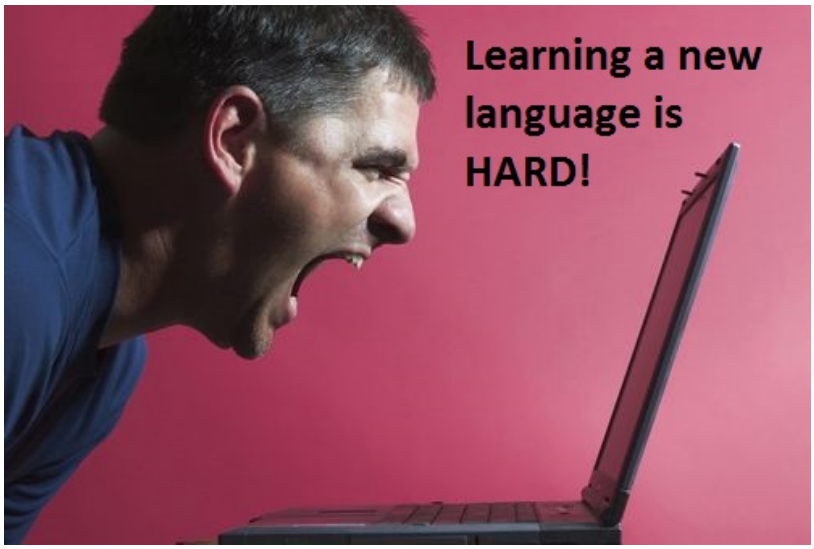
make this side by side and add appropriate alt text

Left photo credit; Right photo credit. Entire figure credit: Data Carpentry R for Social Scientists.

check this link

Examples of things you can do in RStudio

Programming is scary!



check links

Learning to program has other benefits

- Improves your logical skills and critical problem solving
- Increases your attention to detail
- Increases your self reliance and empowers you to control your own research.
- Your PI will love your awesome graphics and reports.
- Some people think what you do is magic.
- Thinking graduate school? [expect to learn this on your own]
- [A few] [other lists] [of reasons]

Why no point and click?

Because it's not reproducible.

- Which boxes did you click last time?
- New data? Gotta do it all over.
- Need to expand your model? Gotta do it all over.
- Made a mistake in the data coding? Gotta do it all over…

References

Part I

Introduction to your new tools

1 Your New Tools

update obj

> **Learning Objectives**
>
> After completing this lesson learners will be able to:
>
> - Use R and R Studio on their personal computer.
> - Describe the purpose of the RStudio Script, Console, Environment, and Plots panes.
> - Organize files and directories for a set of analyses as an R Project, and understand the purpose of the working directory.
> - Execute simple commands in the console
> - Use the built-in RStudio help interface to search for more information on R functions.
> - Demonstrate how to provide sufficient information for troubleshooting with the R user community.

1.1 Setup your class folder for success! :cinema:

link to chapter video?

Using a consistent folder structure across your projects will help keep things organized, and will also make it easy to find/file things in the future. This can be especially helpful when you have multiple projects.

1.1.1 Naming Things

You also need to choose a naming convention for your class folder and stick with it. Recommended options are:

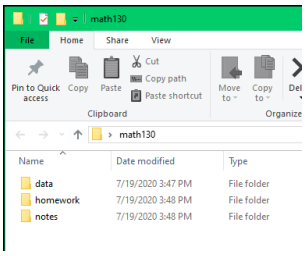
- ALL CAPS (`MATH130`)

- no caps (`math130`)
- snake_case (`math_130`)

- CamelCase (`Math130`)

1.1.2 Create Folders

1. On your computer, in an easy to find place, create a new folder named `math130`.
2. Then create three subfolders: `data`, `homework`, `notes`.



1.1.3 Adding files into your class folder

When you download a file, right click and "Save as" or "Save target as" and **actively choose** where to download this file.

:x: Do not let files live in your downloads folder.

:x: Do not open any files from your browser window after downloading.

:pencil2: Right click [this link] to download and save Assignment 1 into your `homework` folder now.

> need to fix hw1 link loc and qmd

:cinema: Windows video walk through.

> to what?

1.2 Installing R and R Studio

1.2.1 Tablet and Chromebook users

If you are using a tablet, Chromebook or otherwise do not have a computer that you can install programs on, you can use Posit Cloud for this class.

- Make a **Cloud FREE** account at https://posit.cloud/plans/free
- Start a new project by clicking on the button in the top right corner
- Then go to slide #6 to learn how to navigate RStudio.

The free account allows for 25 project hours/month, which may not be enough for this class. If you run into time limits or you are using this for another class you will have to upgrade to the **Cloud Student** plan which is $5/month.

Using the cloud is easier to initially setup, but having your own installation on your computer ensures that

- you want to keep the program free forever
- you will be able to put your files under version control
- you always have access to your code even with unstable or no internet

1.3 Download and install R

1.3.1 Step 1: Download R v 4.5+

- Windows 10 https://cran.r-project.org/bin/windows/base/
- Mac OS X page - https://cran.r-project.org/bin/macosx/
  - First link under "Latest Release" and looks like **R-4.5.0.pkg**.
  - :warning: You may later get a message about needing *X11* or *XQuartz*. The download for that program is also on this page. (Mac only)
- Choose to save the file, do not open or run.

1.3.2 Step 2: Install R

- Install R by double clicking on the downloaded file and following the prompts.
  - Default settings are OK.
  - Delete any desktop shortcuts that was created (looks like the icon above.)

! Video Tutorials for both R and R Studio.

- Windows
- Mac

1.4 Download and install R Studio

1.4.1 Step 1: Download the most recent version from

- https://posit.co/download/rstudio-desktop/
- Your operating system should be automatically detected. If not, scroll down and choose your version.

1.4.2 Install the program

- Windows: Double click on the downloaded file to run the installer program.
- Mac: Double click on the downloaded file, then drag the R Studio Icon into your Applications folder.
  - After you are done, eject the "Drive" that you downloaded by dragging the icon to your trash.

1.5 Navigating R Studio

We will be interacting with the programming language R *only* through R Studio. Not by itself. There are four panes, or windows, in R Studio.

update to newest carpentry instructions

yellow for videos?

Watch one of the following short videos to learn how to navigate R Studio.

- R Girls on YouTube
- Dr. D's overview

1.5.1 Setting preferences in R Studio

AKA: Retain sanity while troubleshooting

side by side/text image

1. Open R Studio and go to the file menu go to *Tools* then *Global Options*.
2. Uncheck "Restore .RData into workspace at startup"
3. Where it says "Save workspace to .RData on exit:" Select "Never" "
4. Click apply then ok to close that window.

This will ensure that when you restart R you do not "carry forward" objects such as data sets that you were working on in a prior assignment.

To effectively restart R, go to the file menu and click *Session* , then "Restart R", or "Restart R and clear output". :cinema: Link to video walk through

2 Programming with R

write objectives

2.1 Terminology

The basis of programming is that we write down instructions for the computer to follow, and then we tell the computer to follow those instructions.

We write, or *code*, instructions in R because it is a common language that both the computer and we can understand.

We call the instructions *commands* and we tell the computer to follow the instructions by *executing* (also called *running*) those commands.

The **console** pane is the place where commands written in the R language can be typed and executed immediately by the computer. It is also where the results will be shown for commands that have been executed.

You can type commands directly into the console and press `Enter` to execute those commands, but they will be forgotten when you close the session.

2.2 Code appearance

In these notes, code is displayed like this:

```
2+2
```

where the output or result of the code is displayed with two pound signs (`##`)

2.3 R is an overgrown calculator

can I use webR for this page?

:pencil: In the console type the following code, then press `Enter`.

```
2+2
```

```
[1] 4
```

Now try a more complicated equation.

```
2 + 5*(8^3)- 3*log10)
```

Uh oh, we got an Error. Nothing to worry about, errors happen all the time.

:pencil: Put a open parenthesis `(` before `log10` to fix it and try again.

3 > R is waiting on you...

:pencil: In the console type the following code, then press `Enter`.

```
2 + 5*(8^3)- 3*log(10
```

Notice the console shows a `+` prompt. This means that you haven't finished entering a complete command.

This is because you have not 'closed' a parenthesis or quotation, i.e. you don't have the same number of left-parentheses as right-parentheses, or the same number of opening and closing quotation marks.

When this happens, and you thought you finished typing your command, click inside the console window and press `Esc`; this will cancel the incomplete command and return you to the `>` prompt.

3.1 Packages

"Where the real money from the movie is made."

R is considered an **Open Source** software program. That means many (thousands) of people contribute to the software. They do this by writing commands (called functions) to make a particular analysis easier, or to make a graphic prettier.

When you download R, you get access to a lot of functions that we will use. However these other *user-written* packages add so much good stuff that it really is the backbone of the customizability and functionality that makes R so powerful of a language.

For example we will be creating graphics using functions like `boxplot()` and `hist()` that exist in base R. But we will quickly move on to creating graphics using functions contained in the `ggplot2` package. We will be managing data using functions in `dplyr` and reading in Excel files using `readxl`. Installing packages will become your favorite past-time.