Lesson 02

Introduction to your new tools

Last updated: 2020-07-22

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.

Icon Definitions

- ▶ Forward arrow: Task for you to do
- ▾ Down arrow: A download link is present.
- ✕ Red X: Something you should NOT do
- ⚠ Warning triangle: this is something you should be aware of or pay attention to.

Installing programs

⚠ If you are using a tablet, Chromebook or otherwise do not have a computer that you can install programs on, head over to http://rstudio.cloud.com, make an account and start a new project. Then skip to slide #6 to learn how to navigate RStudio.

While the cloud is easier to initially setup, I want you to only use it if you absolutely have to. Having your own installation on your computer ensures that

- you likely want to customize your programs
- you will be able to put your files under version control
- you always have access to your code even with unstable or no internet

R is the programming language that we will be using. R Studio is the program we will *use* R through.

You must install both.

Download and install R

▼ Download R v 4.0+

- 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-4.0.2.pkg**.
- Choose to save the file, do not open or run.

▶ Install

- 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

Download and install R Studio

Download v1.3 +

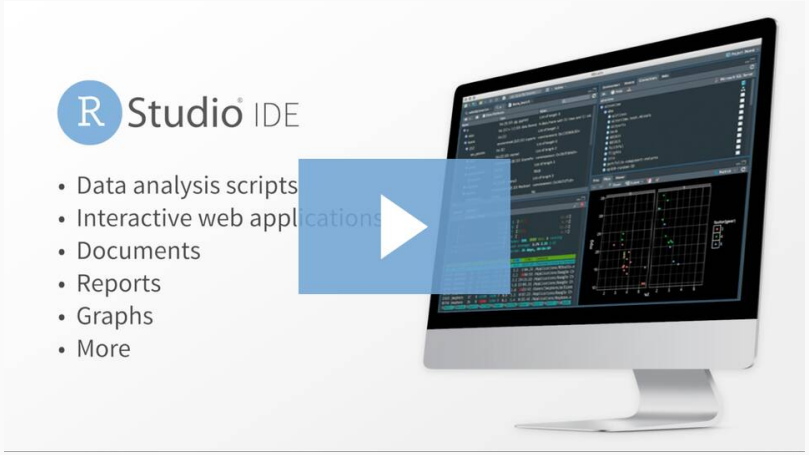
- Download R Studio (v 1.3+) from https://www.rstudio.com/products/rstudio/download/#download
- Choose the download link that corresponds to your operating system.

Install

- 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.

Navigating R Studio

▶ Watch the following short video to learn how to navigate R Studio.



RStudio IDE Overview

Setting preferences in R Studio

Retain sanity while troubleshooting

▶

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

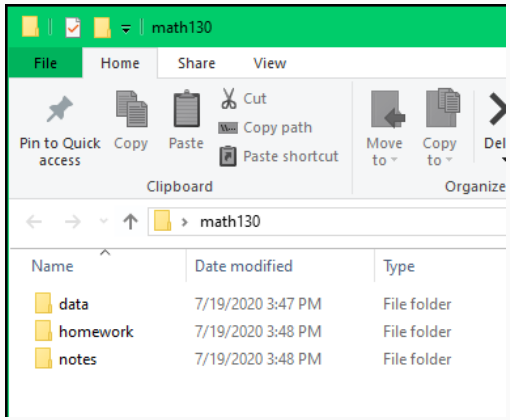
This will ensure that when you restart R you do not "carry forward" objects such as data sets that you were working on in a prior assignment.

Downloading class materials

Setup your folder for success!

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.

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



You 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`)

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.

❌ Do not let files live in your downloads folder.

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

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

Programming with R

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.

Code appearance

In these notes, code is displayed like this:

```
2+2
```

```
## [1] 4
```

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

R is an overgrown calculator

▶ 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)
```

```
## Error: <text>:1:21: unexpected ')'
## 1: 2 + 5*(8^3)- 3*log10)
##                         ^
```

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

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

> R is waiting on you...

▶ 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.

Packages

Where the real money from the movie is made.

All the fun functions are in packages

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.

Installing Packages

▶ Start by typing the following in the console to install the `ggplot2` package.

```
install.packages("ggplot2")
```

When the download and install is complete, you should see a message simliar to:

```
The downloaded binary packages are in
    C:\Users\Robin\AppData\Local\Temp\Rtmpi8NAym\downloaded_packages
```

⚠ R is case sensitive and spelling matters. If you get an error message like the following:

```
Warning in install.packages :
  package 'ggplot' is not available (for R version 3.5.1)
```

The correct package name is `ggplot2`, not `ggplot`.

Install all the things!

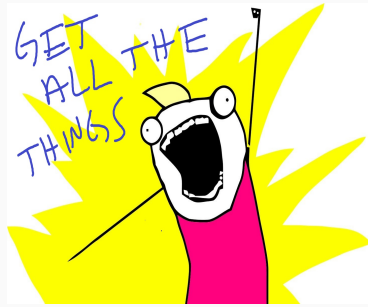
Now that you're a package installing pro, go ahead and install the following packages that we will be using in the next few weeks.

▶ Type these commands into the console one at a time and wait for it to finish before entering the next command.

```
install.packages("rmarkdown")
install.packages("dplyr")
install.packages("knitr")
install.packages("forcats")
install.packages("readxl")
```

Note in this last package that is a lower case letter L at the end. For read EXCELL. It is not the number 1 (one).

Alternative Method of installing Packages:
Use the Package tab in the lower right pane in R Studio.



Seeking Help

Sometimes a second pair of eyeballs is all you need

Image credit: Intro to DS using R Workshop

Advice on asking for help

The key to receiving help from someone is for them to rapidly grasp your problem. You should make it as easy as possible to pinpoint where the issue might be.

Try to use the correct words to describe your problem. For instance, a package is not the same thing as a library. Most people will understand what you meant, but it can make things confusing for people trying to help you. Be as precise as possible when describing your problem.

⚠ Don't let not knowing exactly how to describe your problem prevent you from asking. Screenshots help temendously

When asking someone for help try to

1. Explain what thing you are trying to do
2. Explain/show the code you wrote to try to do that thing
3. Explain/show your result, and if it's not obvious explain why you feel it's not the correct result. (E.g. you expected the answer to be 5, but instead it's 10.)

⚠ Don't spend more than 20 minutes banging your head on the wall before you ask the instructor for help!

Help from inside R Studio

Use the built-in RStudio help interface to search for more information on R functions

One of the fastest ways to get help, is to use the RStudio help interface. This panel by default can be found at the lower right hand panel of RStudio. As seen in the screenshot, by typing the word `mean`, RStudio tries to also give a number of suggestions that you might be interested in. The description is then shown in the display window.

I know the name of the function I want to use, but I'm not sure how to use it

If you need help with a specific function, let's say `barplot()`, you can type:

```
?barplot
```

If you just need to remind yourself of the names of the arguments, you can use:

```
args(lm)
```

Help from inside R Studio cont.

I want to use a function that does X, there must be a function for it but I don't know which one...

If you are looking for a function to do a particular task, you can use the `help.search()` function, which is called by the double question mark `??`. However, this only looks through the installed packages for help pages with a match to your search request

```
??kruskal
```

If you can't find what you are looking for, you can use the rdocumentation.org website that searches through the help files across all packages available.

Finally, a generic Google or internet search "R \" will often either send you to the appropriate package documentation or a helpful forum where someone else has already asked your question.

I'm stuck

I get an error message that I don't understand

Start by googling the error message. However, this doesn't always work very well because often, package developers rely on the error catching provided by R. You end up with general error messages that might not be very helpful to diagnose a problem (e.g. "subscript out of bounds"). If the message is very generic, you might also include the name of the function or package you're using in your query.

If you check Stack Overflow, search using the `[r]` tag. Most questions have already been answered, but the challenge is to use the right words in the search to find the answers: http://stackoverflow.com/questions/tagged/r

⚠ Development of R moves pretty fast. When at all possible, use results from the past 1-2 years. Anything over 5 years old for packages such as `ggplot`, `dplyr`, and `forcats` are likely obsolete.

"In Person" help

Unfortunately in person 👪 help not available during Fall 20 due to COVID19 😬

However, Zoom is the next best thing.

- Your friendly classmates: if you know someone with more experience than you, they might be able and willing to help you.
- Attend Community Coding.
  - Hours can be found here: bit.ly/cc_hours
  - Drop in work session & dedicated space to work on coding projects.
  - Collaborate with your peers and learn from experts.
- The R Users Meetup group useful if you want to stay connected to the community and learn about upcoming events.
  - https://www.meetup.com/Chico-R-Users-Group/

Other Online

- In RStudio go to `Help` --> `Cheatsheets`
- R Studio Cloud interactive lessons: https://rstudio.cloud/learn/primers
- Stack Overflow: if your question hasn't been answered before and is well crafted, chances are you will get an answer in less than 5 min. Remember to follow their guidelines on how to ask a good question.
- Chico R Users Google Group.
- The R-Studio Community: it is read by a lot of people and is more welcoming to new users than the R list-serv.
- If your question is about a specific package, see if there is a mailing list for it. Usually it's included in the DESCRIPTION file of the package that can be accessed using `packageDescription("name-of-package")`. You may also want to try to email the author of the package directly, or open an issue on the code repository (e.g., GitHub).
- Twitter: #rstats @Rstudio

Written

If you're a book kinda person, there is plenty of help available as well. Many have online versions or free PDF's.

- R Markdown, the Definitive Guide: https://bookdown.org/yihui/rmarkdown/
- R for Data Science https://r4ds.had.co.nz/
- Cookbook for R http://www.cookbook-r.com/
- R Graphics Cookbook (I use this all the time) -- Chapter 8 in the above link
- The Art of R Programming https://nostarch.com/artofr.htm
- R for... http://r4stats.com/
  - Excel Users https://www.rforexcelusers.com/
  - SAS and SPSS Users http://r4stats.com/books/r4sas-spss/
  - STATA Users http://r4stats.com/books/r4stata/

Moar resources!

- The Posting Guide for the R mailing lists.
- How to ask for R help useful guidelines
- This blog post by Jon Skeet has quite comprehensive advice on how to ask programming questions.

Final Tips

Saving and closing your work.

Unless you're returning to work in R Studio in a short while, You should make a habit to save all open tabs and completely shut down R studio when you are done working. This ensures your environment is cleared. *This is a good thing.*

Restart R

To restart R without shutting the entire window down, go to the file menu bar in the top,

> *Session* --> Restart R and Clear Output

This is good to do when switching between projects/classes.

Credits