Class 2: Introduction to R and Data Plotting

MGSC 310

Prof. Jonathan Hersh

Class 2: Outline

1. R Studio Introduction
2. Why Use R Studio Projects?
3. Lab Class 2
    - Basics of R
    - Plotting with ggplot2

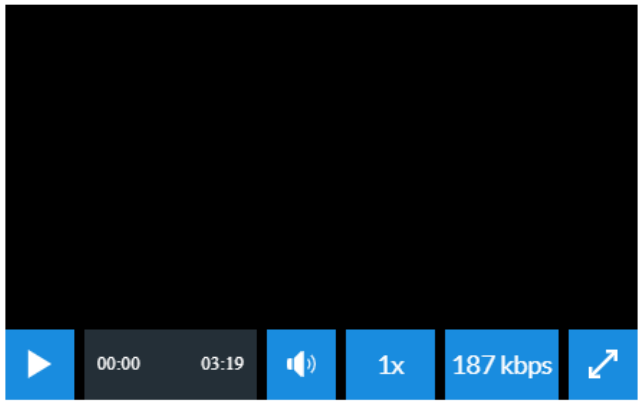
Please Follow Instructions to Install Computer Tools

Below is a list of software you will need for this course. Follow the links to install the software.

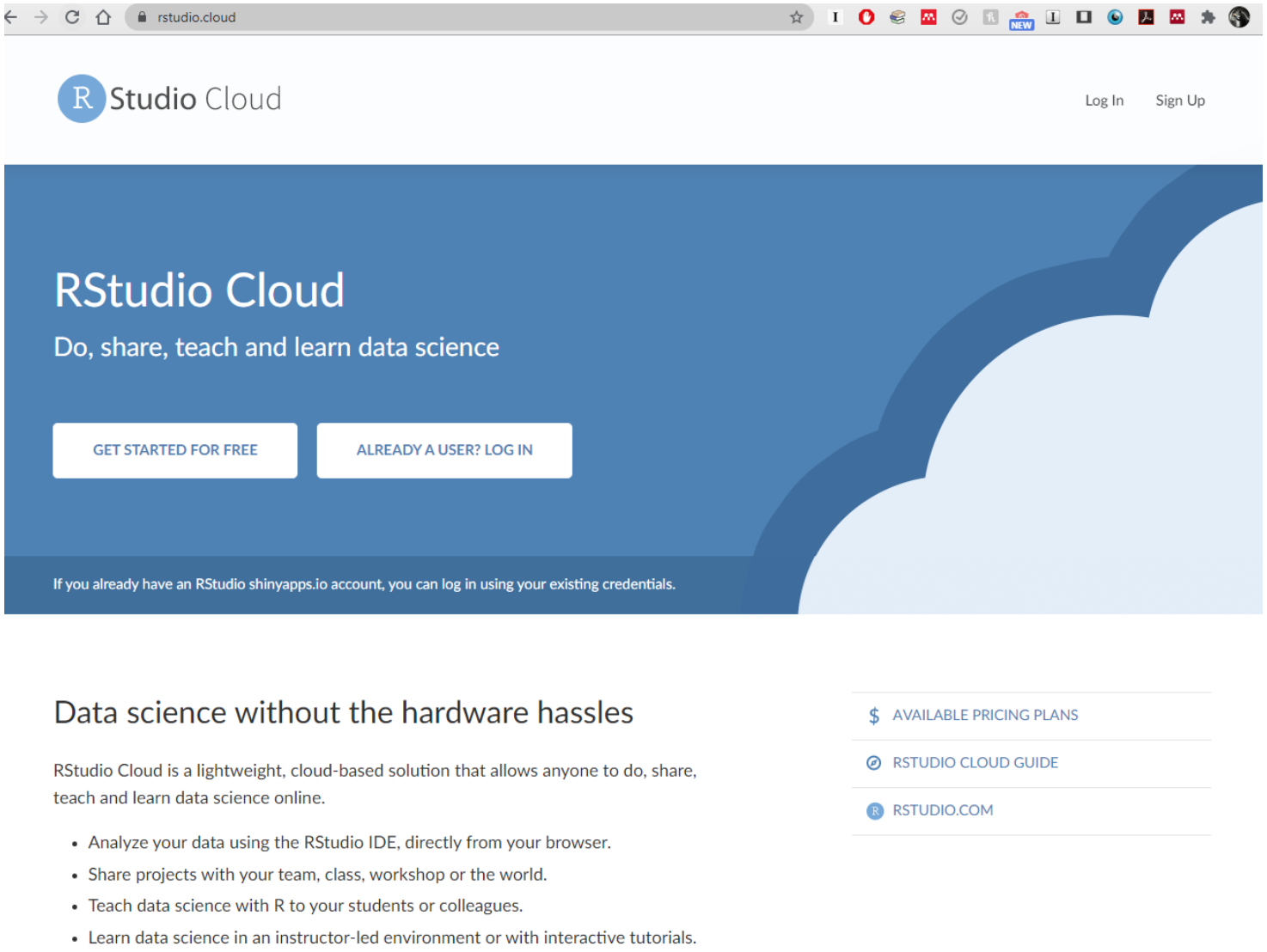
- R 4.0.2:
  - Window Download
  - Mac Download
- Rstudio v.1.3.1073:
  - Download
- Miktex (needed to produce pdf output):
  - Download
- Compiler tools (needed to load certain packages)
  - Windows
    - RTools
  - Mac
    - Xcode and GFortran

RStudio server.

Update (9/2) This video will help you debug two common errors

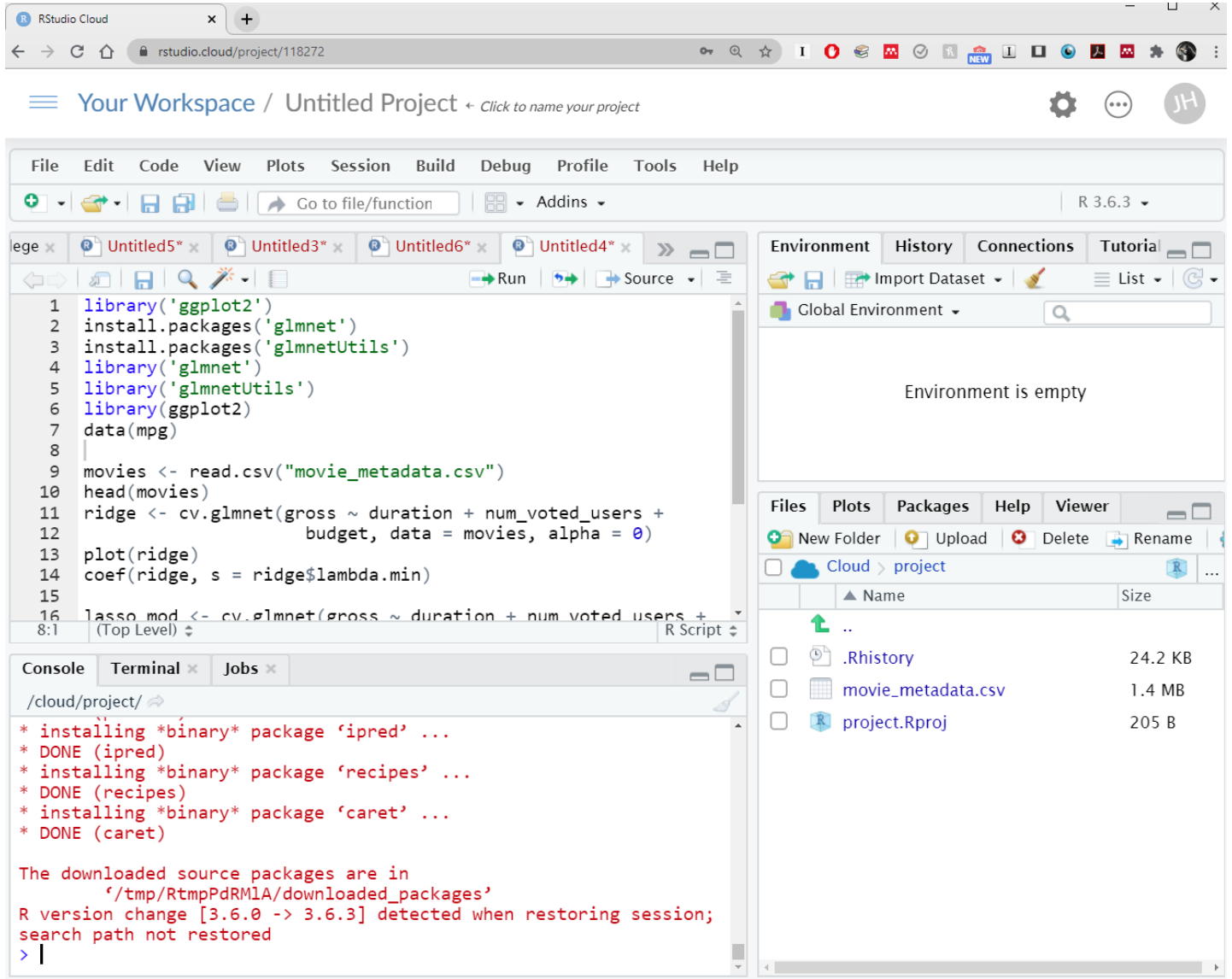


R Studio Cloud



- Go to rstudio.cloud if your version of R is ever not working

R Studio Cloud

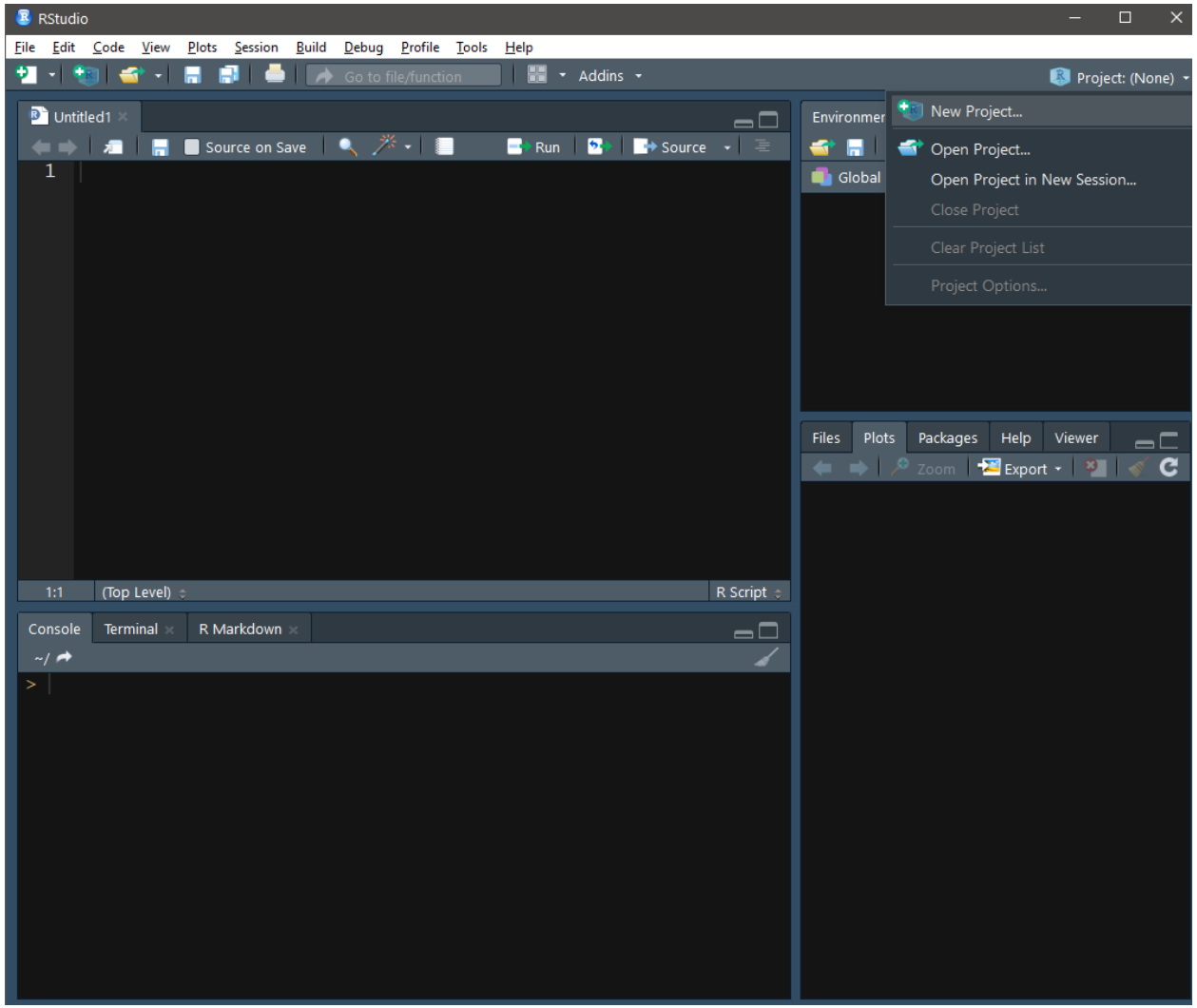


- R Studio Cloud is a full featured version of R in your browser!

Class 2: Outline

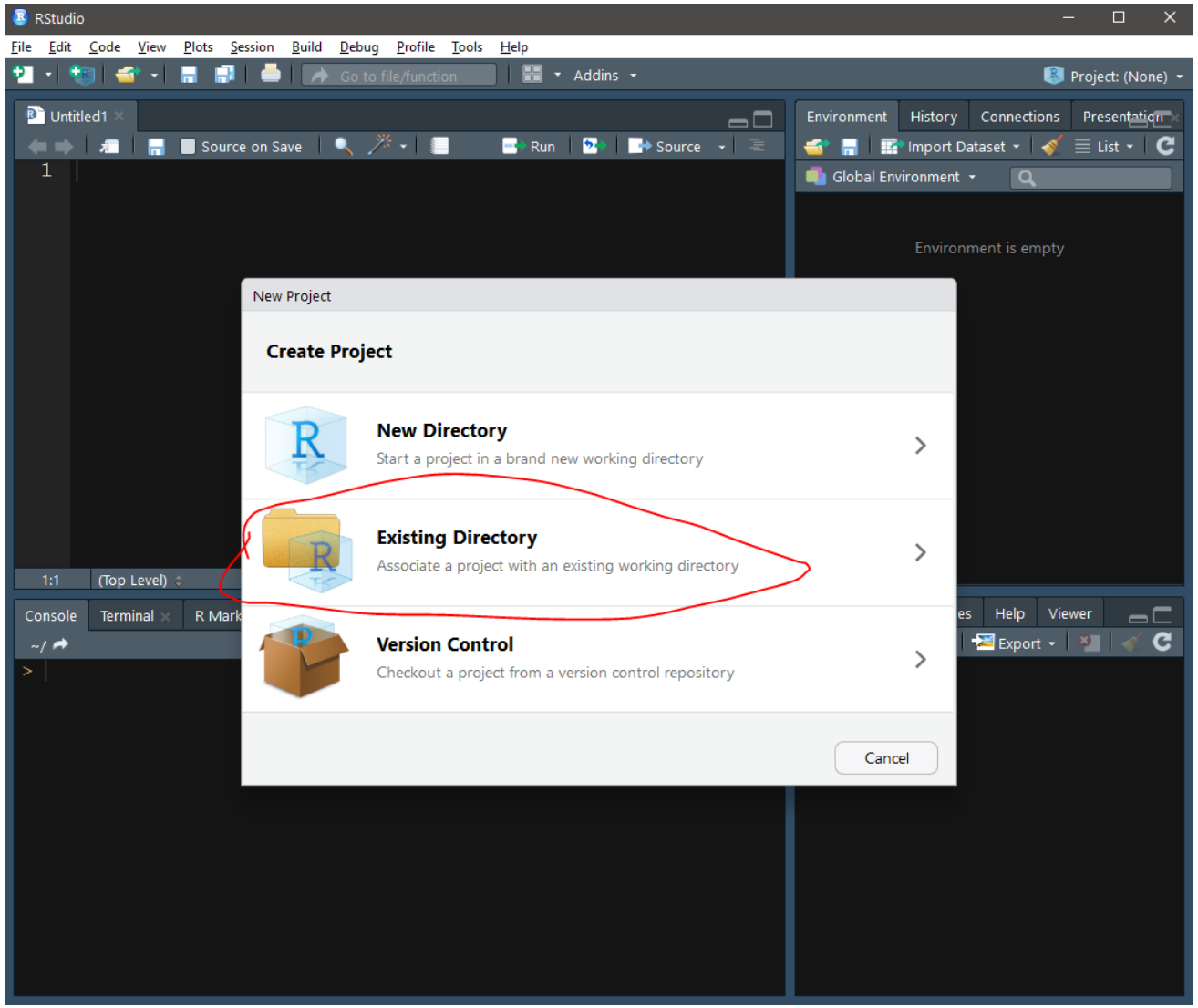
1. R Studio Introduction
2. **Why Use R Studio Projects?**
3. Lab Class 2
    - Basics of R
    - Plotting with ggplot2

Rstudio Projects



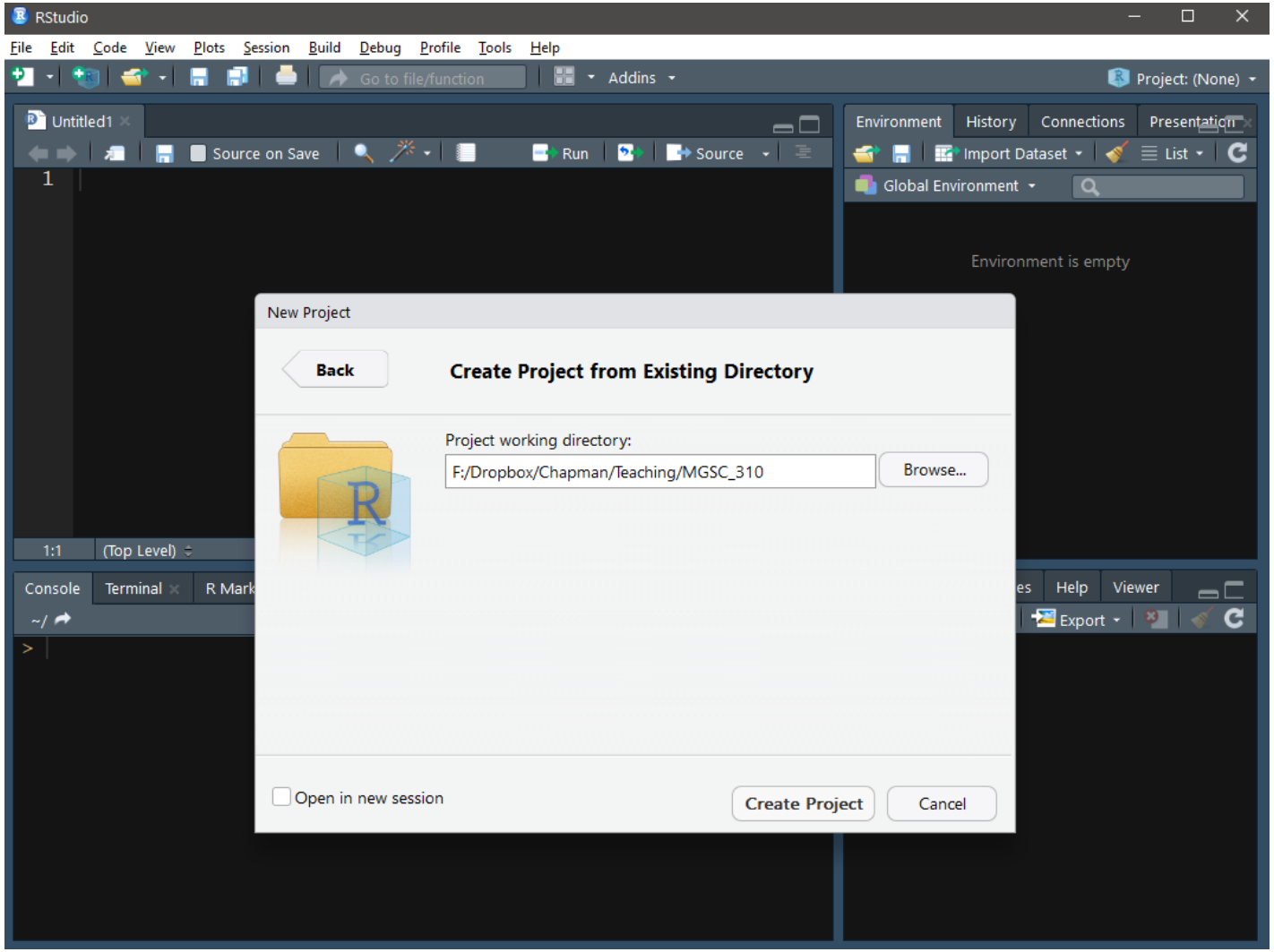
- Rstudio Projects are environments for your code
- They make your life easier, especially across teams and platforms
- Always use Rstudio Projects!

Rstudio Projects



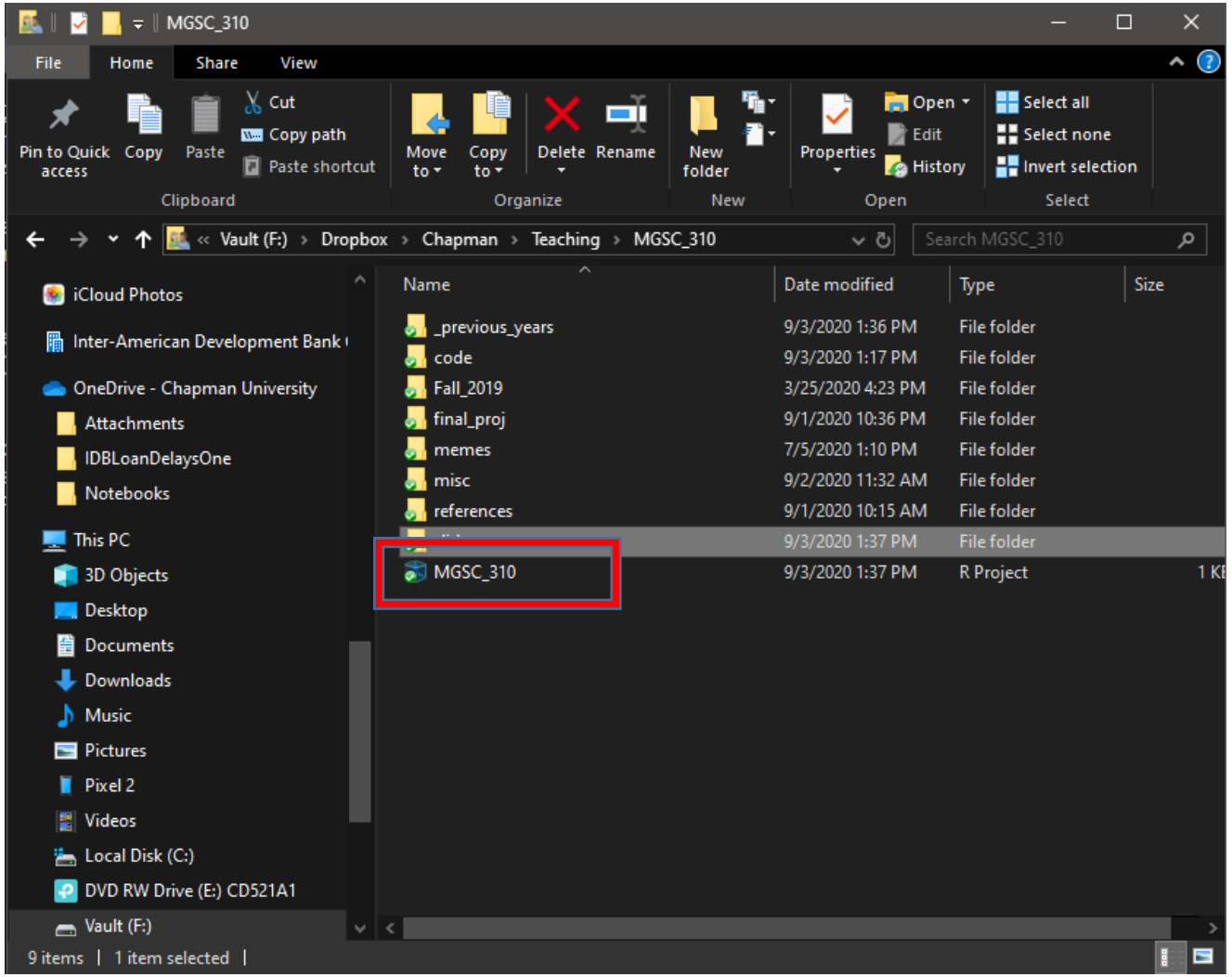
- I recommend you create a new project in your MGSC310 Folder
- Select file -> new project

Rstudio Projects



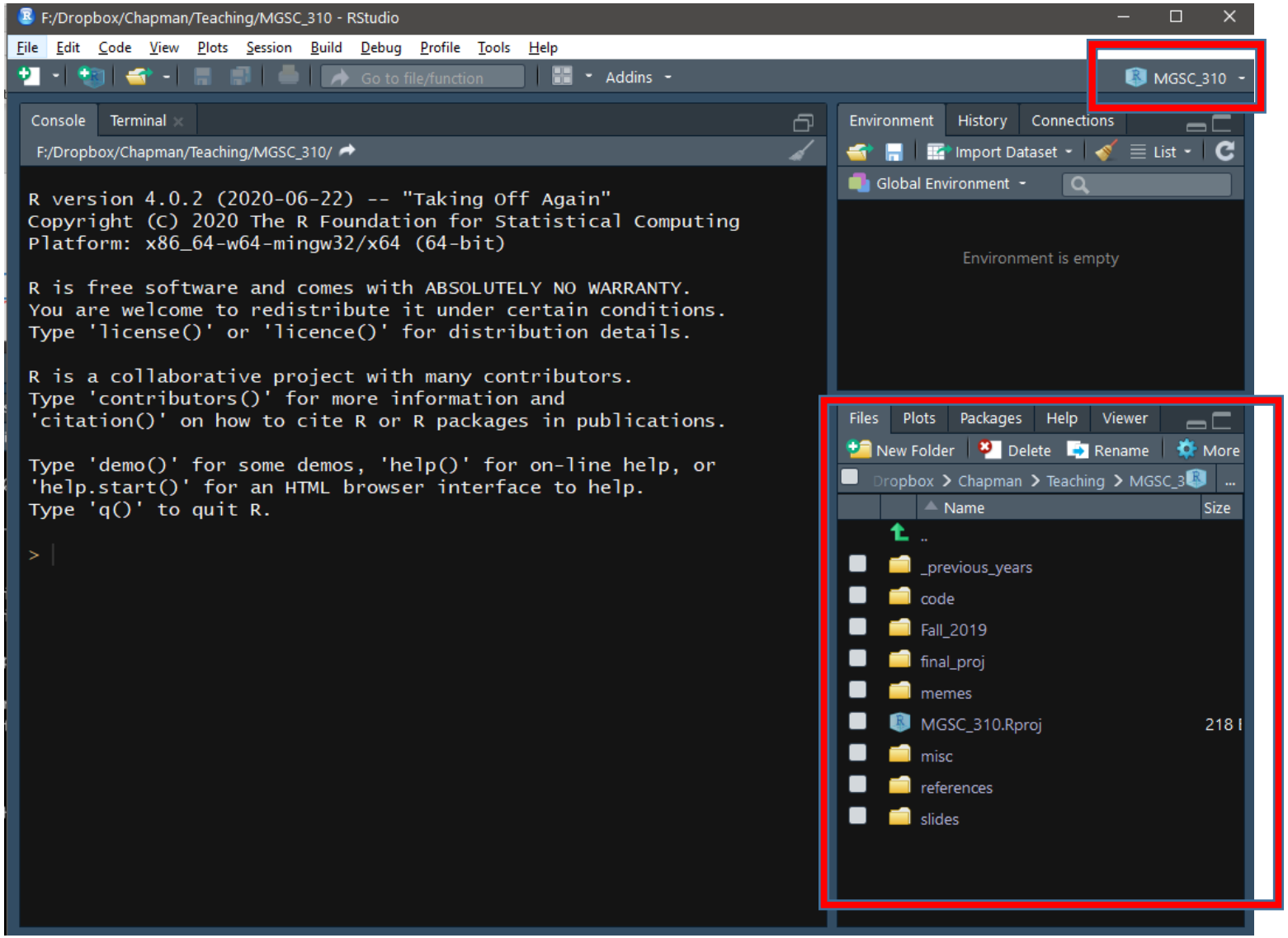
- I recommend you create a new project in your MGSC310 Folder
- Select file -> new project

Rstudio Projects



- This creates an .Rproj file you should click and open whenever you want to run code for this class

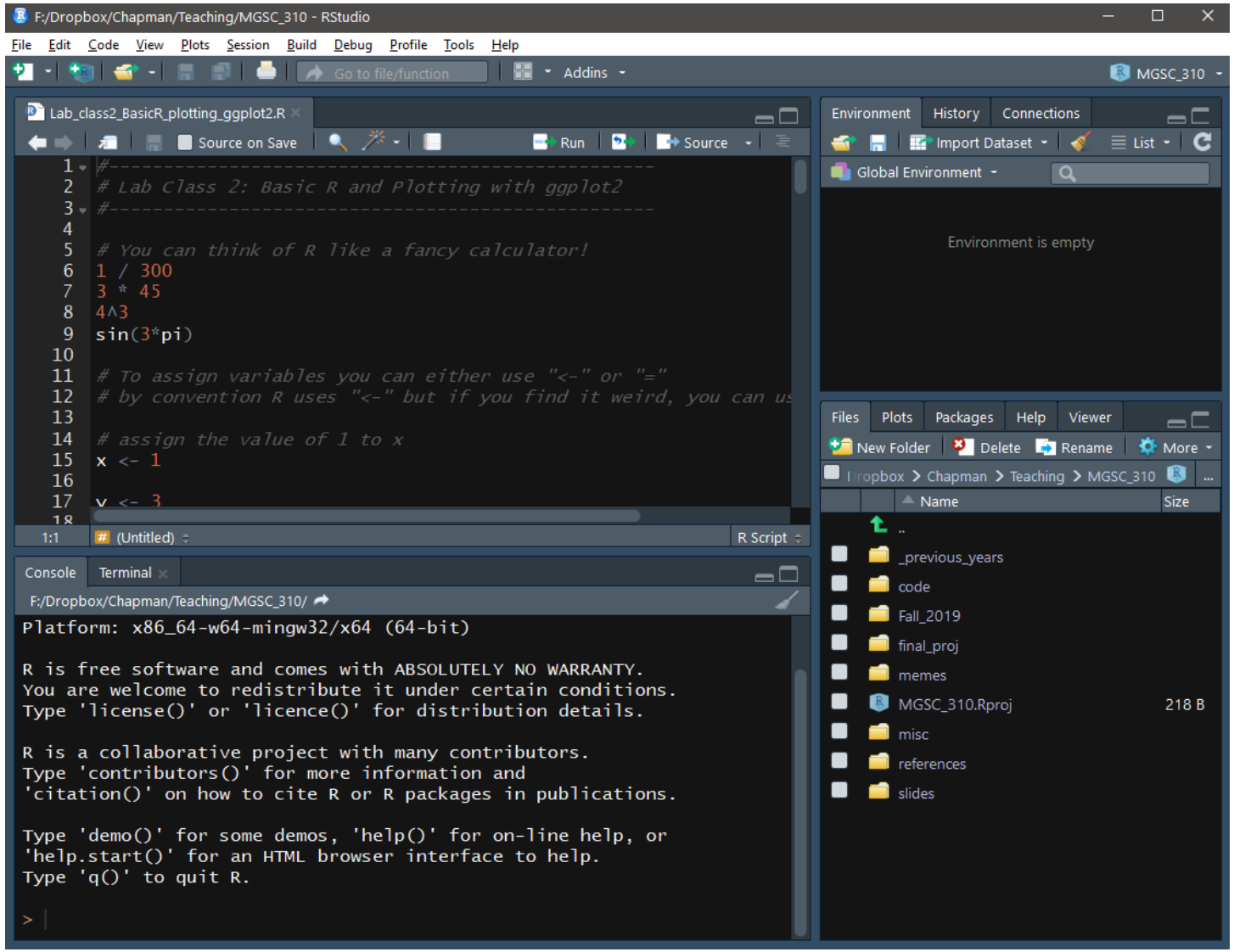
Rstudio Projects



- You should see your project name in the top right and the "root directory" of the project in the bottom right

- All links to files, datasets are relative to this root folder

Think of R as a Fancy Calculator



Plotting with ggplot2

Overview

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Installation

```
# The easiest way to get ggplot2 is to install the whole tidyverse:
install.packages("tidyverse")

# Alternatively, install just ggplot2:
install.packages("ggplot2")

# Or the development version from GitHub:
# install.packages("devtools")
devtools::install_github("tidyverse/ggplot2")
```

- ggplot2 is a package in the tidyverse that allows for high quality graphics in using a minimal amount of code

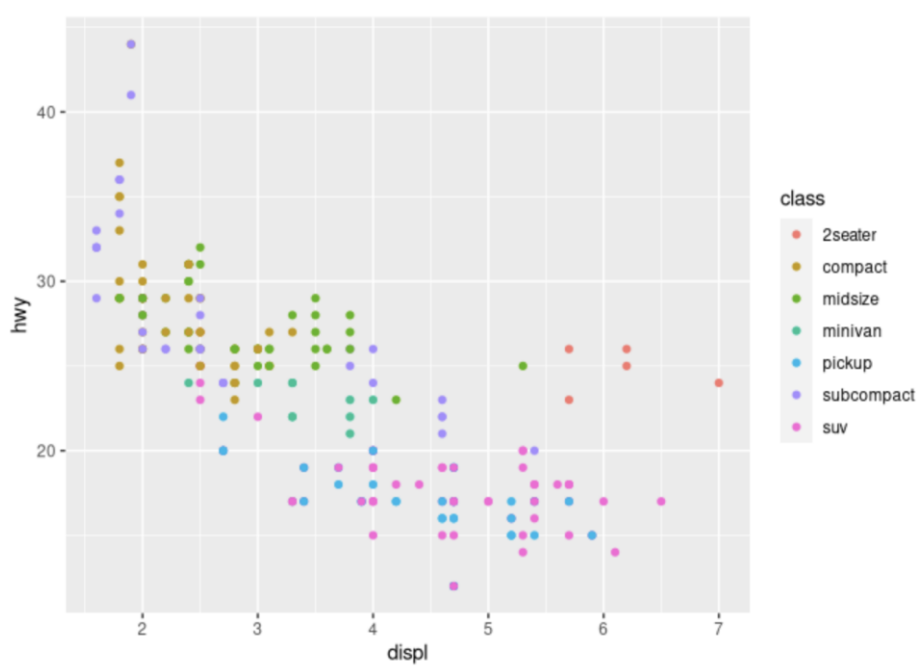
ggplot2: “Grammar of Data Manipulation”

Usage

It’s hard to succinctly describe how ggplot2 works because it embodies a deep philosophy of visualisation. However, in most cases you start with `ggplot()`, supply a dataset and aesthetic mapping (with `aes()`). You then add on layers (like `geom_point()` or `geom_histogram()`), scales (like `scale_colour_brewer()`), faceting specifications (like `facet_wrap()`) and coordinate systems (like `coord_flip()`).

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
```



- ggplot2 uses a grammar of data manipulation
- We begin with a “canvas” (usually x,y plane) and add points, lines, etc. on that canvas
- Each new part is added using a “+” operator