PLSC 476: Empirical Legal Studies

Christopher Zorn

January 21, 2021

Software

- Preferred: R + RStudio
- Also viable: Stata
- Possible, but harder: Python, Javascript, etc.
- Highly discouraged: SAS, SPSS/PSPP, Minitab, Statistica, etc.

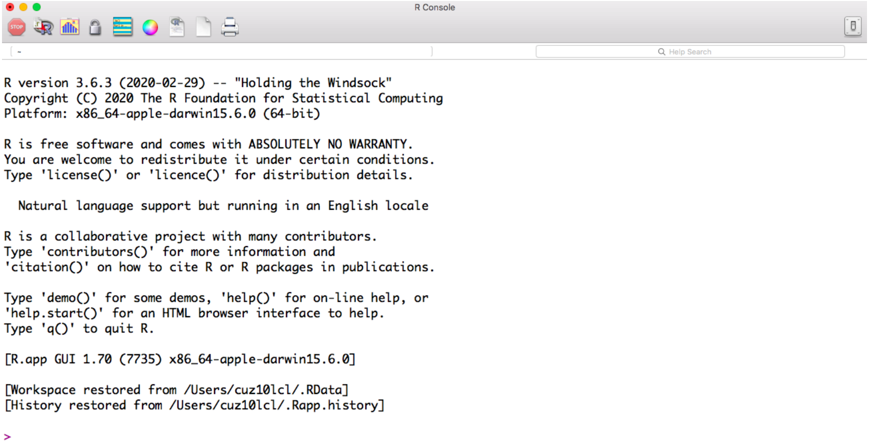
Using R / RStudio

- Simplest: Install and run R + RStudio locally
    - **R** (v. 4.0.3, from `cran.r-project.org`), and
    - **RStudio (Desktop)** (v. 1.4.1103, from `www.rstudio.com/products/rstudio`)

- Almost as simple: RStudio.cloud...

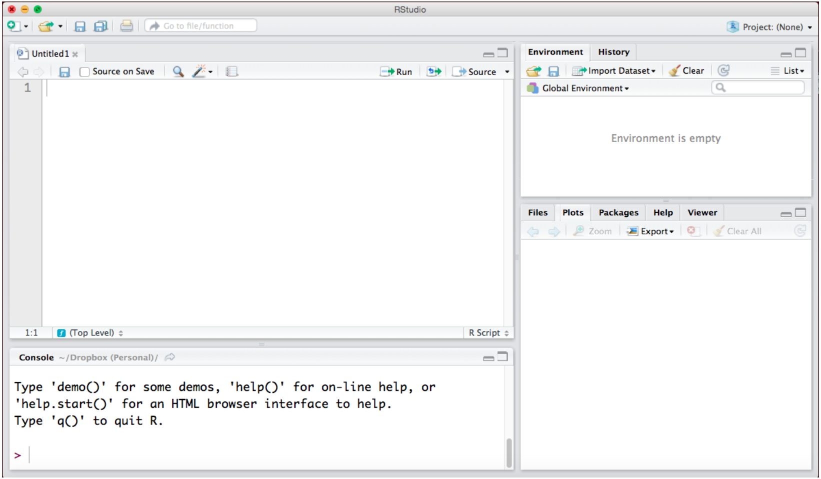
- Also possible: Access via PSU Linux clusters (e.g., like this)

R



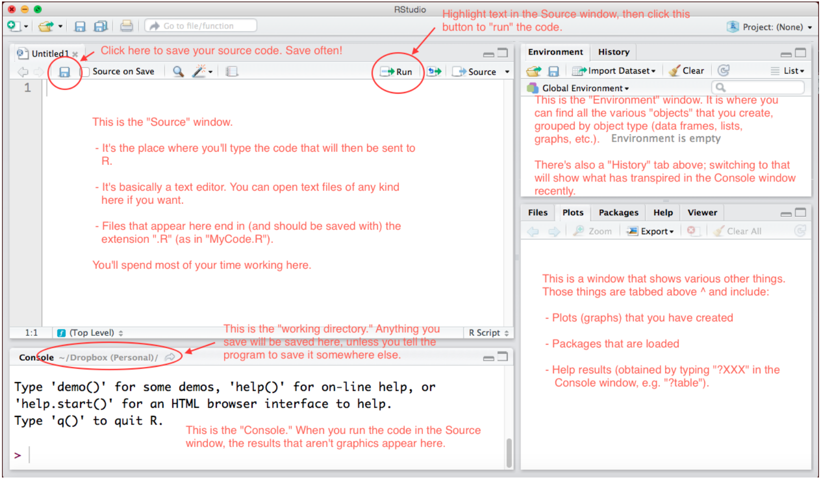
```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.70 (7735) x86_64-apple-darwin15.6.0]

[Workspace restored from /Users/cuz10lcl/.RData]
[History restored from /Users/cuz10lcl/.Rapp.history]

>
```

RStudio



```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

RStudio (annotated)



Highlight text in the Source window, then click this button to "run" the code.

Click here to save your source code. Save often!

This is the "Source" window.

- It's the place where you'll type the code that will then be sent to R.
- It's basically a text editor. You can open text files of any kind here if you want.
- Files that appear here end in (and should be saved with) the extension ".R" (as in "MyCode.R").

You'll spend most of your time working here.

This is the "working directory." Anything you save will be saved here, unless you tell the program to save it somewhere else.

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

This is the "Console." When you run the code in the Source window, the results that aren't graphics appear here.

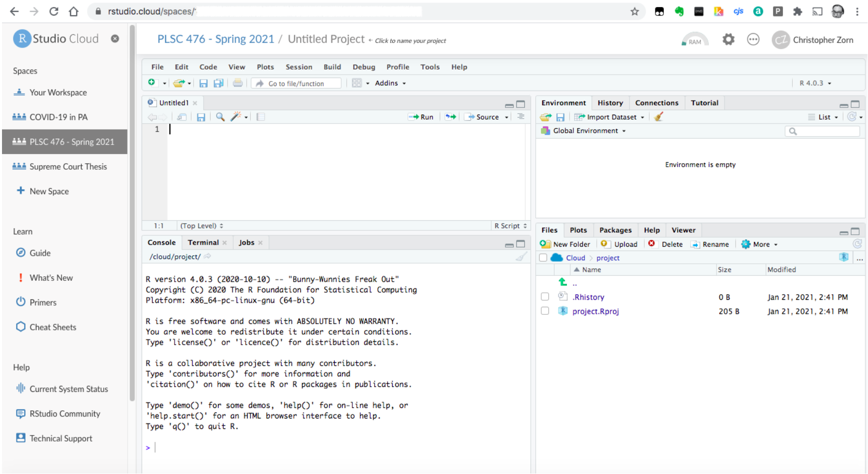
This is the "Environment" window. It is where you can find all the various "objects" that you create, grouped by object type (data frames, lists, graphs, etc.).

There's also a "History" tab above; switching to that will show what has transpired in the Console window recently.

This is a window that shows various other things. Those things are tabbed above ^ and include:

- Plots (graphs) that you have created
- Packages that are loaded
- Help results (obtained by typing "?XXX" in the Console window, e.g. "?table").

RStudio.cloud



Environment is empty

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Name	Size	Modified
..		
.Rhistory	0 B	Jan 21, 2021, 2:41 PM
project.Rproj	205 B	Jan 21, 2021, 2:41 PM

Inside the Source Window

This:

```
> table(df$X)
```

... means “Type the phrase ‘table(df$X)’ on the command line,” or – equivalently – “Type the phrase ‘table(df$X)’ into your Source code, and then run it.”

Inside the Source Window

More often, you'll see:

```
with(df, plot(Y~X,pch=19,col="red"))  # draw a scatterplot
abline(h=0,lty=2)                     # add a horizontal line at zero
abline(v=0,lty=2)                     # add a vertical line at zero
text(df$X,df$Y,labels=df$names,pos=1) # add labels
```

... which means "Put this block of text into your Source code, and then run it."

Note:

- R / RStudio ignores line breaks and white space
- Anything to the right of a "#" is a comment

R: First Things

1. The “working directory”
    - Where R “looks for” and “keeps” all the things
    - It defaults to some version of the “root” directory on the system
    - You probably want to change it more-or-less immediately, to something more useful...

E.g.:

```
> setwd("~/Documents/PLSC476")      # OS-X / Linux
> setwd("C:/My Documents/PLSC476") # Windows
```

R: First Things

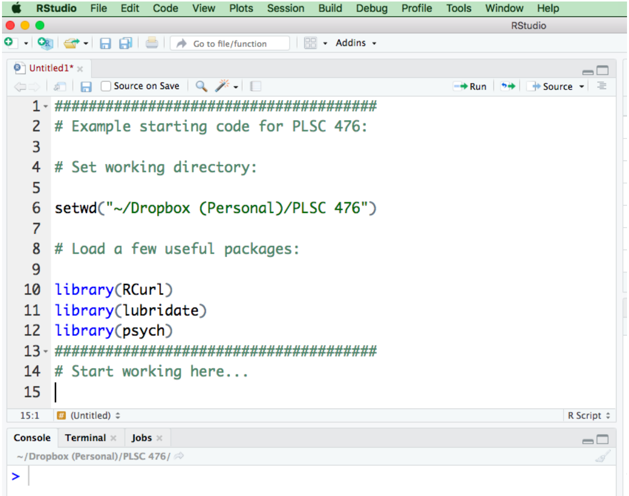
2. Packages
    - Bundles of functions, commands, data, and documentation that do specific things (data management, statistics, graphics, etc.)
    - To be used, packages first have to be *installed*, using the `install.packages()` command
    - Once a package is installed, it needs to be *loaded* for use, using `library()`
    - If you try to use a command from a package, and the package is not installed and loaded, you'll get an error (see below)

E.g.:

```
> qqp(SCOTUS$LiberalPct)
Error in qqp(df) : could not find function "qqp"
> install.packages("car")   # install
> library(car)              # load for use
> qqp(SCOTUS$LiberalPct)    # <-- works
```

A Source Code Header

Tip: Keep these things at the beginning of your source code, e.g.:



```r
######################################
# Example starting code for PLSC 476:

# Set working directory:

setwd("~/Dropbox (Personal)/PLSC 476")

# Load a few useful packages:

library(RCurl)
library(lubridate)
library(psych)
######################################
# Start working here...

```

Getting Data

1. You can enter data into R/RStudio *by hand* (but you'd rather not)
2. You can read data from a *local file*. The simplest way to do that is if the data is in standard `.csv` format:
    - values in each row separated by commas,
    - rows separated by line breaks, and
    - a single row with comma-separated variable names at the top

E.g.:

```
> DF <- read.csv("Data/SCOTUS-votes.csv")
```

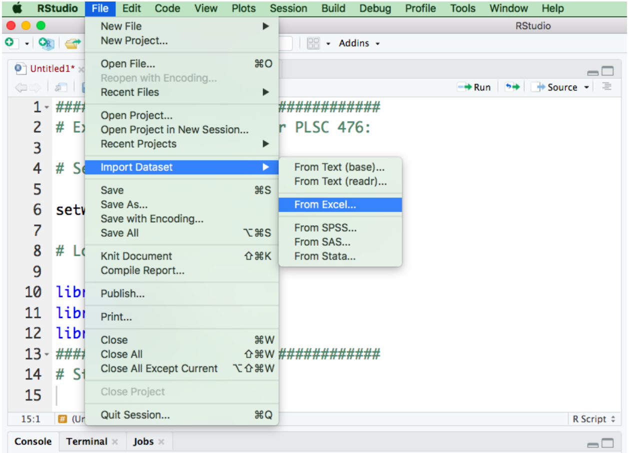
Getting Data

3. You can read a .csv (or, for that matter, a .xls, or other format file) directly from the web, via the file's URL, using the RCurl package:

```
> install.packages("RCurl")
> library(RCurl)
> foo <- getURL("http://mqscores.lsa.umich.edu/media/2019/justices.csv")
> MQScores <- read.csv(text = foo)
```

Getting Data

4. In RStudio, you can read data interactively, via “File → Import Dataset”



Getting Data

5. You can import data dynamically via RESTful (and other) APIs...
    - Simplest via the `httr` and `jsonlite` packages
    - Usually requires knowledge of JSON-formatted data
    - A basic tutorial is here.

Summary:

Data Formats and Packages

Format	Useful Command(s) / Package(s)
Plain text (`.txt`, etc.)	`read.table`, `readr`
Comma-separated (`.csv`)	`read.csv`
Excel (`.xls`, etc.)	`readxl`, `xlsx`
SPSS (`.sav`, etc.)	`haven`, `foreign`
Stata (`.dta`)	`haven`, `foreign`
SAS (`.ssd`, etc.)	`haven`, `foreign`
JSON (`.json`)	`jsonlite`
Databases (`.adb`, `.sql`, etc.)	`DBI`, `odbc`

Examples...

(see `PLSC476-Second-Day-Software.R`)

Help For Learning R(Studio)

In rough order of preference:

- Quick-R (`http://www.statmethods.net/`)
- The “Level-Zero” R Tutorial (doesn’t integrate RStudio, but is otherwise very good)
- Statistics with R
- The Do It Yourself Introduction to R
- Also be sure to consult the PLSC 476 “Useful R Resources” guide (on GitHub).

How to Figure Things Out in R

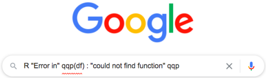
1. One leading question mark (`"?foo"`) = "help." So, if you want help with the details of the `reshape` command, in the console type:

   ```
   > ?reshape
   ```

2. Two leading question marks (`"??foo"`) = "search." So, if you want to figure out how you might use R to *reshape* your data, in the console type:

   ```
   > ??reshape
   ```

3. Google is your friend:*

   R "Error in" qqp(df) : "could not find function" qqp

4. Stack Overflow really *is* your friend...