Lesson 0

David John Baker

04/05/2020

R, RStudio, and Tidyverse

R

You can download R for your computer by going to CRAN and selecting the appropriate `Download and Install R` links.

Make sure to install R first before installing RStudio.

The Comprehensive R Archive Network

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2018-12-20, Eggshell Igloo) R-3.5.2.tar.gz, read what's new in the latest version.

- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).

- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.

- Source code of older versions of R is available here.

- Contributed extension packages

Figure 1: CRAN Homepage

RStudio

RStudio is an integrated development environment (IDE) for R[1]. RStudio is basically your workbench where you can access everything you need for managing your scripts, data, and project structure. By using RStudio, you also can use a host of other features ranging from Markdown documents (like this one!), interactive data dashboards like Shiny, and the tidyverse.

---

[1]https://www.rstudio.com/products/RStudio/

Figure 2: RStudio

rproj

One thing that I will introduce now, but not talk much about is the `.rproj` file. What this little file does (you can open it up as a text file and see how little it is) is bascially wall off your directory from everything else on your computer so you don't have to write obnoxiously long absolute paths. They make it easier for your work to run on other's computers.

Packages and the tidyverse

What makes R different than programs like Excel or SPSS is that R as you download it does not come with everything installed. Downloading R from CRAN gives you what is referred to as **base R**. The idea is that since you do not need all the software, all the time, you just load in what you need. R can be used just by itself, but the real advantage of it is that so many people use it and write software for it, if there is something that you need to do that is somewhat common, chances are that someone has written software for it already. The external software that you load in are referred to as **packages** which are kept in your **library**. Some packages are very small and simple, others have extensive teams developing them. One of the most important packages in R is the tidyverse.

The tidyverse is a collection of packages that were developed to make manipulating data more intuitive. As noted on its homepage,

> All packages share an underlying design philophy, grammar, and data structures.

While this might seem trivial now, having your data be `tidy` opens up an entire world of data manipulation and modeling. Once you you get over the initial learning curve of R, the tidyverse makes it so that you can pretty much take off and learn very q quickly.

RStudio Environment

Once you now have R and RStudio installed, it's time to open up RStudio. By opening RStudio, you are also starting R. R will be running under the hood of RStudio. After installing R, you can run it on it's own

by typing `R` into your terminal on a Unix machine (Mac, Linux). Though after seeing how RStudio works, you would realize why doing this is basically masochistic. (If you do this, you can quit out of the terminal R with `quit()` followed by `n`).

When you first open RStudio will see a few different panels. In it's default settings, the bottom left is the Console. The top right has your Environment, History, and version control commands. The bottom right has your Viewer, Library for your packages, and a system to navigate your files. The top left will be where you write your code.
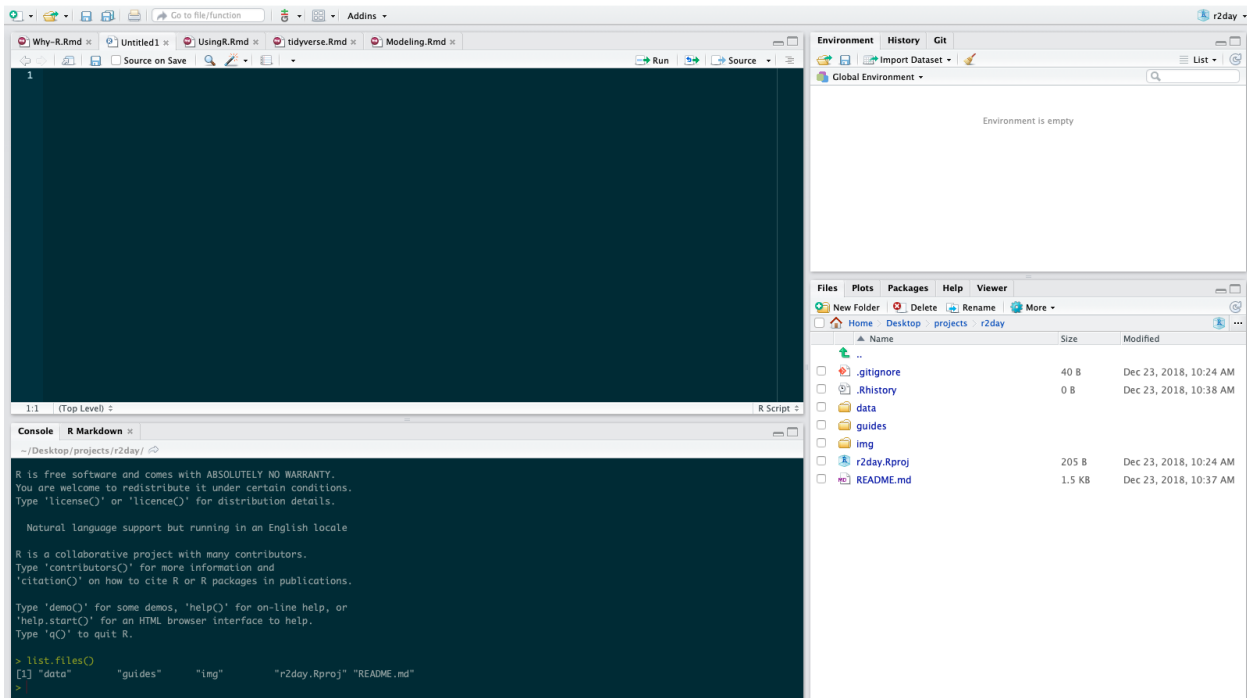

Figure 3: RStudio Environment

Environment

The top left has information about your current Environment. As you make new things in an R session you can track them here. There is also a History tab here that keeps track of code you wrote. Additionally there is a Git tab that will eventually allow you to do version control. You don't have to know what that is, but one day you might read about it.

Viewer

The bottom right is your File Explorer/Finder window. Try to click around on the **Files** tab. When you click **Plots** there should be nothing there as you have not made any plots yet. Your **Packages** tab will have a listing of software that you can load into R. Notice that if you click one of the package names, it will navigate you to the **Help** tab. Lastly, the Viewer tab will let you display any documents that you make while writing in R. This could be markdown documents or maybe a website that you are writing eventually.

> It is important to note that you will probably "break" R and RStudio many times when learning. Know that this is OK and the some of the best advice for learning how to program is by just seeing what happens when you change something and Googling your problems.

An Example

Just so we get a bit of practice *doing* something in R, let's import a dataset, then save it after making small modification to it. It's not super important you get the syntax quite yet, that will come next lesson!

Imagine your colleagues is teaching a lesson with the `tips.csv` dataset to group of students from different countries (England, Japan, India) who are about to take their first visit to the USA. To get them used to the idea of how much they are expected to tip their servers, we need to make new columns that convert the USD column to pounds, yen, and Australian dollars.

Note we're going to do this the `tidyverse` way, which will be the focus of this short course!

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------ tidyverse

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  3.0.0     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0


## -- Conflicts --------------------------------------------------------------------- tidyverse_confl
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
tips <- read_csv("tips.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   total_bill = col_double(),
##   tip = col_double(),
##   sex = col_character(),
##   smoker = col_character(),
##   day = col_character(),
##   time = col_character(),
##   size = col_double()
## )
```

```r
converted_tips <- tips %>%
  select(-X1) %>%
  mutate(gbp_total= total_bill * 0.81) %>%
  mutate(gbp_tips = tip * 0.81) %>%
  mutate(yen_total = total_bill * 106) %>%
  mutate(yen_tips = tip * 106) %>%
  mutate(aus_total = total_bill * 0.64) %>%
  mutate(aus_tips = tip * 0.64)

# Vectorization
tips$total_bill
```

4

```
##   [1] 16.99 10.34 21.01 23.68 24.59 25.29  8.77 26.88 15.04 14.78 10.27 35.26
##  [13] 15.42 18.43 14.83 21.58 10.33 16.29 16.97 20.65 17.92 20.29 15.77 39.42
##  [25] 19.82 17.81 13.37 12.69 21.70 19.65  9.55 18.35 15.06 20.69 17.78 24.06
##  [37] 16.31 16.93 18.69 31.27 16.04 17.46 13.94  9.68 30.40 18.29 22.23 32.40
##  [49] 28.55 18.04 12.54 10.29 34.81  9.94 25.56 19.49 38.01 26.41 11.24 48.27
##  [61] 20.29 13.81 11.02 18.29 17.59 20.08 16.45  3.07 20.23 15.01 12.02 17.07
##  [73] 26.86 25.28 14.73 10.51 17.92 27.20 22.76 17.29 19.44 16.66 10.07 32.68
##  [85] 15.98 34.83 13.03 18.28 24.71 21.16 28.97 22.49  5.75 16.32 22.75 40.17
##  [97] 27.28 12.03 21.01 12.46 11.35 15.38 44.30 22.42 20.92 15.36 20.49 25.21
## [109] 18.24 14.31 14.00  7.25 38.07 23.95 25.71 17.31 29.93 10.65 12.43 24.08
## [121] 11.69 13.42 14.26 15.95 12.48 29.80  8.52 14.52 11.38 22.82 19.08 20.27
## [133] 11.17 12.26 18.26  8.51 10.33 14.15 16.00 13.16 17.47 34.30 41.19 27.05
## [145] 16.43  8.35 18.64 11.87  9.78  7.51 14.07 13.13 17.26 24.55 19.77 29.85
## [157] 48.17 25.00 13.39 16.49 21.50 12.66 16.21 13.81 17.51 24.52 20.76 31.71
## [169] 10.59 10.63 50.81 15.81  7.25 31.85 16.82 32.90 17.89 14.48  9.60 34.63
## [181] 34.65 23.33 45.35 23.17 40.55 20.69 20.90 30.46 18.15 23.10 15.69 19.81
## [193] 28.44 15.48 16.58  7.56 10.34 43.11 13.00 13.51 18.71 12.74 13.00 16.40
## [205] 20.53 16.47 26.59 38.73 24.27 12.76 30.06 25.89 48.33 13.27 28.17 12.90
## [217] 28.15 11.59  7.74 30.14 12.16 13.42  8.58 15.98 13.42 16.27 10.09 20.45
## [229] 13.28 22.12 24.01 15.69 11.61 10.77 15.53 10.07 12.60 32.83 35.83 29.03
## [241] 27.18 22.67 17.82 18.78
```

```
tips$total_bill * 0.81
```

```
##   [1] 13.7619  8.3754 17.0181 19.1808 19.9179 20.4849  7.1037 21.7728 12.1824
##  [10] 11.9718  8.3187 28.5606 12.4902 14.9283 12.0123 17.4798  8.3673 13.1949
##  [19] 13.7457 16.7265 14.5152 16.4349 12.7737 31.9302 16.0542 14.4261 10.8297
##  [28] 10.2789 17.5770 15.9165  7.7355 14.8635 12.1986 16.7589 14.4018 19.4886
##  [37] 13.2111 13.7133 15.1389 25.3287 12.9924 14.1426 11.2914  7.8408 24.6240
##  [46] 14.8149 18.0063 26.2440 23.1255 14.6124 10.1574  8.3349 28.1961  8.0514
##  [55] 20.7036 15.7869 30.7881 21.3921  9.1044 39.0987 16.4349 11.1861  8.9262
##  [64] 14.8149 14.2479 16.2648 13.3245  2.4867 16.3863 12.1581  9.7362 13.8267
##  [73] 21.7566 20.4768 11.9313  8.5131 14.5152 22.0320 18.4356 14.0049 15.7464
##  [82] 13.4946  8.1567 26.4708 12.9438 28.2123 10.5543 14.8068 20.0151 17.1396
##  [91] 23.4657 18.2169  4.6575 13.2192 18.4275 32.5377 22.0968  9.7443 17.0181
## [100] 10.0926  9.1935 12.4578 35.8830 18.1602 16.9452 12.4416 16.5969 20.4201
## [109] 14.7744 11.5911 11.3400  5.8725 30.8367 19.3995 20.8251 14.0211 24.2433
## [118]  8.6265 10.0683 19.5048  9.4689 10.8702 11.5506 12.9195 10.1088 24.1380
## [127]  6.9012 11.7612  9.2178 18.4842 15.4548 16.4187  9.0477  9.9306 14.7906
## [136]  6.8931  8.3673 11.4615 12.9600 10.6596 14.1507 27.7830 33.3639 21.9105
## [145] 13.3083  6.7635 15.0984  9.6147  7.9218  6.0831 11.3967 10.6353 13.9806
## [154] 19.8855 16.0137 24.1785 39.0177 20.2500 10.8459 13.3569 17.4150 10.2546
## [163] 13.1301 11.1861 14.1831 19.8612 16.8156 25.6851  8.5779  8.6103 41.1561
## [172] 12.8061  5.8725 25.7985 13.6242 26.6490 14.4909 11.7288  7.7760 28.0503
## [181] 28.0665 18.8973 36.7335 18.7677 32.8455 16.7589 16.9290 24.6726 14.7015
## [190] 18.7110 12.7089 16.0461 23.0364 12.5388 13.4298  6.1236  8.3754 34.9191
## [199] 10.5300 10.9431 15.1551 10.3194 10.5300 13.2840 16.6293 13.3407 21.5379
## [208] 31.3713 19.6587 10.3356 24.3486 20.9709 39.1473 10.7487 22.8177 10.4490
## [217] 22.8015  9.3879  6.2694 24.4134  9.8496 10.8702  6.9498 12.9438 10.8702
## [226] 13.1787  8.1729 16.5645 10.7568 17.9172 19.4481 12.7089  9.4041  8.7237
## [235] 12.5793  8.1567 10.2060 26.5923 29.0223 23.5143 22.0158 18.3627 14.4342
## [244] 15.2118
```

```
write_csv(converted_tips, "converted_tips.csv")
```