Installing R and RStudio

Why install these programs?

For this class you have been provided with some nifty modules embedded in the online textbook that allow you to practice the code for statistical concepts that were just covered. Additionally, there is a sandbox that you can use that gives you a little more freedom and flexibility. However, as you progress into the class and beyond it, you might find yourself wanting to do and try more, and those modules will feel limiting. R is a fantastic open-source language for statistical programming that is completely free, and with a powerful IDE (integrated development environment) like RStudio, it is fun to play and work with.

Installing R

It is recommended that you install R before you install RStudio. If you have a Linux operating systems, it is likely that you already have R installed, but Mac and Windows users should head to https://cran.rstudio.com/ to get started. The first section at that link contains links to download pages for Linux, Windows, and Mac installations. Choose the link appropriate for your computer and follow the download instructions given.

Mac OS

When you click the download link for Mac OS, you should see some short sections describing the version releases, and then a little bit down the page a section called *Files*. Download the most recently published file (the highest in the list, with the highest numbering; R-3.4.4.pkg at the time of writing this document). Leave all of the default settings when installing.

Windows

When you click the download link for Windows, you should see a heading called *Subdirectories* with a link called *base*. Follow that link and you should find a page that has a larger link that says "Download R" followed by a version number ("Download R 3.4.4 for Windows" at the time of writing this document). Click the download link, and leave all of the default settings when installing.

Installing RStudio

With R installed, the next step is to get RStudio installed. RStudio makes working with R a lot more fun. It will do things like autocomplete function and variable names, allow you to view full datasets as tables, keep track of your objects, and more. You can download RStudio at https://www.rstudio.com/products/rstudio/download/#download. Once you download and install RStudio you are almost done!

Creating a workspace for this course

Now that you have base R and RStudio installed, there are a few R packages that are used in this class that do not download automatically. These packages provide functions that make working with R a lot easier, and you have been using a number of them like `arrange()` and `select()` already. The packages you need to use all the functions we use in class are `dplyr`, `ggplot2`, `ggformula`, `mosaic`, `supernova`, and `lsr`. Additionally, we use some datasets that are found in the packages `Lock5Data`, `Lock5withR`, `okcupiddata`, `dslabs`, and `fivethirtyeight`.

To use an R package, you need to do three things:

1. download the package
2. install the package
3. load the package

Downloading and Installing packages

For most cases, you can use the `install.packages()` function to both download and install packages, like this:

```
# install a single package
install.packages("dplyr")

# install multiple packages at once
install.packages(pkgs = c(
  "dplyr", "ggplot2", "ggformula",
  "mosaic", "supernova", "lsr",
  "Lock5Data", "Lock5withR",
  "okcupiddata", "dslabs",
  "fivethirtyeight"
))
```

Note that when you run the code above, R might install other packages as well. The other packages are dependencies meaning that the packages you want to install need them to function.

Loading packages

Now that everything has been downloaded and installed, you need to load the packages before you can use them. Note that you need to load the packages each time you start up R. To load a package, you need to call `library()` with the package name. Because we need to do this every time R starts, it is helpful to save the code to start up every package into a script, and then just have RStudio run that script for us when we need it. To create a new script, open RStudio and click *File* | *New File* | *R Script*. This will open a script in the editor pane. The next block of code will load all of the packages we use in this class, so paste that code into the script and then save it.

```
library(dplyr)
library(ggplot2)
library(ggformula)
library(supernova)
library(lsr)
library(mosaic)
library(Lock5Data)
library(Lock5withR)
library(okcupiddata)
library(dslabs)
library(fivethirtyeight)
```

To run all of this code, you can do one of two things, either click *Source* or highlight all of the code and click *Run*. When you run the code, you should see a lot of package startup messages that you can more-or-less ignore. If you would like to suppress these messages, use the `suppressPackageStartupMessages()` function like so:

```
# note the additional { } brackets
suppressPackageStartupMessages({
  library(dplyr)
  library(ggplot2)
  # etc.
})
```

Installing other packages

Most of the time, R packages are hosted on what is called CRAN. CRAN is the Comprehensive R Archive Network, and has mirrors all around the world making sure it is available to those that need to download R or its packages. When a package is hosted on CRAN, you can install it just as we have done so far. However, some developers choose not to use CRAN, and often use GitHub or another code repository site instead. For instance, Twitter provides the

AnomalyDetection package on GitHub. To install a package like this, you will need to use the `devtools` package's function `install_github()` like so:

```
# note that devtools must be installed first
library(devtools)
install_github("twitter/AnomalyDetection")
```