Module 3 – lesson 04: Working with R packages

Script

So far we’ve looked at customizing our documents by changing various options either in the code chunks or in the YAML header. In this lesson, we’re going to customize our document by adding parameters to our documents.

To learn more about adding parameters to your documents, see <http://rmarkdown.rstudio.com/lesson-6.html> and <http://rmarkdown.rstudio.com/developer_parameterized_reports.html>

For this lesson, we’re going to work with a dataset from the fivethirtyeight R package. Go ahead and install this package – go to Tools/Install Packages and type in fivethirtyeight to install this package.

You can learn more about the fivethirtyeight package at the CRAN repository site (which is a reliable place from which to install R packages – I’ll explain this in more detail in a moment)

<https://cran.r-project.org/web/packages/fivethirtyeight/index.html>

and at the Github repository where the authors of the package host the package which is still under development

<https://cran.r-project.org/web/packages/fivethirtyeight/index.html>

LEARNING ABOUT R PACKAGES

I’ve mentioned this briefly before, but when you install R and RStudio you get only the basic functionality of the core base functions and packages built into the base R software. The bulk of the functionality of R comes from user contributed R packages that you install as you need them. The biggest advantage of this is that you literally have the whole world of R developers working to improve R every day.

The biggest disadvantage of open source software is there is no single central group that checks and validates these R packages – so in the open source software arena you should always check to see how reliable the authors and the package is before you use it. Investigate how long the package has been around, how many people use it, have you seen it used in publications. You can also get an idea of how much code testing and validation has been done based on the repository hosting the R package.

There are typically 4 places where you can find R packages:

1. The CRAN repository <https://cran.r-project.org/> where you can explore submitted packages alphabetically and by “Task Views” There are currently over 11,000 packages hosted by CRAN. These packages have all been through some validation and testing. So, these packages are considered to have been “released” for production. That said, it is still possible that these packages have errors or bugs in them so always be sure to double check that the package is working the way you expect it to.

2. Another very reliable repository for R packages is Bioconductor <https://www.bioconductor.org/> . Bioconductor hosts R packages primarily for use in the computational biology research arena. However, the R packages hosted here go through even more rigorous vetting and testing before being accepted by the repository. But it still a good idea to test these packages to make sure they are working like you expect. There are currently over 1400 R packages hosted by Bioconductor.

3. The next most common place to find R packages is on Github. In most cases, the R packages found on Github are considered to be still “in development”. As such, it is more common to find errors and bugs in these R packages. However, you can often get the leading edge (or bleeding edge) of the latest and greatest functionality. But you need to be even more certain that the R package is working as you expect. There is no accurate count for the number of R packages currently hosted on Github but it is well into the many tens of thousands.

4. Finally, virtually any other file server, repository hosting service (such as Bitbucket), or other mode of transferring files or groups of files can be used to distribute an R package. Obviously, only accept files from sources that you trust.

So, let’s take a moment and explore the fivethirtyeight package that we’ll use for this and future lessons. Go to the PACKAGES tab in the lower right window. This TAB lists all of the R packages installed on your local drive. These are all of the packages you currently have access to on your local drive for use when writing R code to do analyses and manipulate data files. Scroll down until you see the fivethirtyeight package and click the hypertext link. This will open a HELP page for the package. This R package mainly consists of datasets that have been used over the past several years at 538.com for the purposes of writing news articles.

For example, for this lesson we’re going to work with the steak\_survey dataset. Click on the link for this dataset. This dataset was used for the article 538.com published in May 2014 on “How Americans Like Their Steak”. There is a link provided in the HELP menu for this article published online. Take a few minutes to read through this article to get an idea of how 538.com uses data to support their journalism. We’re going to explore this dataset similar to how they did. While we’re not going to try and replicate their results, it is good to know you’re working with the same dataset they used when they published this article.

To explore the steak\_survey dataset from the fivethirtheight package, let’s create an R script to save our R commands. We’ll use some of these later when we build our document.

Go ahead and start RStudio and open the R project for Module 3. Once the project opens, in RStudio click File/New File/R Script.

Type in the R script provided in the read ahead materials for this lesson. I’ll explain each one in a minute.

library(fivethirtyeight)

library(tidyverse)

data("steak\_survey", package="fivethirtyeight")

sdat <- na.omit(steak\_survey) %>%

filter(region==”Mountain”)

ggplot(sdat,

aes(x = steak\_prep, fill = female)) +

geom\_bar(position="dodge") +

ggtitle(paste0("Steak Preparation Preference by Gender for ",”Mountain”," Region"))

In addition to the fivethirtheight R package, we’re also going to use the tidyverse R package developed by Hadley Wickham. The tidyverse package included a number of other packages, so when you install the tidyverse package it will take a few minutes to get everything downloaded and installed. Learn more about tidyverse at <https://www.tidyverse.org/> Tidyverse is a new approach to data manipulation and analysis that has a very good programming workflow. While the focus of this course is not on R programming per se, we will be using some R code to create objects like figures and tables for our documents. For consistency, I will be using the Tidyverse approach for R programming for these lessons.

Go ahead and take a few moments to install the tidyverse package.

Let’s go back to the R script you just created. The first 2 library statements LOAD the 2 R packages fivethirtyeight and tidyverse. Even though the packages are installed on our local drive, we have to LOAD them into our current R session for us to be able to use the datasets and functions in these packages. Highlight these 2 lines of code and click Run to load these libraries.

The next line uses the data function to load the steak\_survey dataset from the fivethirtyeight package. Highlight this line of code and click run. Then click on the ENVIRONMENT TAB at the top right to see that this dataset is now loaded in our environment – this means the dataset is now in local memory and we can see it. You’ll notice that this dataset has 550 observations (or rows) and 15 variables (or columns). You can click on the table icon at the right to view this dataset. As you can see most of the columns contain TRUE/FALSE responses for the yes/no questions asked on this steak survey. There are other responses for how people like their steak prepared, gender, age groups, income levels, education and region in the United States.

The next 2 lines use some of the functionality of the dplyr package that is loaded with the tidyverse package – namely the filter function. So, these 2 lines of code make a copy of the steak\_survey dataset but first omit the NAs (not available) which are missing data in this dataset. And then the data are piped into the filter function. The pipe is indicated by the %>% symbol. The filter function subsets the data to look only at the responses from people in the Mountain region of the United States. Highlight these 2 lines of code and click Run. You should now have a new dataset created in your ENVIRONMENT called sdat that has 24 observations and 15 variables.

Now that we have a subset of the steak survey called “sdat” we use that data to make a clustered bar chart of how people like their steak prepared (steak\_prep) by gender (female). These next lines of code use the ggplot2 package function which was also loaded as part of the tidyverse package. Again read more on the tidyverse website about the ggplot2 package. You can also learn more about using ggplot2 to make nice graphics and figures at the Cookbook for R website at <http://www.cookbook-r.com/Graphs/> Go ahead and highlight these last 4 lines of code and click Run to see the clustered bar chart graphic produced by the R code.

We’re going to use this R code script as the basis for creating a new R Markdown document using parameters for the region of the United States. In the next lesson, you will be creating a basic template for the report you want to see based on which region of the United States is selected. You will be working with parameters to further customize your templates and automate your workflow.

For now, let’s go ahead and back everything up to your Github account.

Open Git Bash and make sure you are in the correct directory:

C:\RepTemplates\Module3

Once in that directory, type in the following 4 Git commands to check the status of your local files compared to your Github cloud repository; add or stage the modified files; commit your changes; and then push the changes to your Github cloud repository.

git status

git add .

git commit –m “add R Script for fivethirtyeight package”

git push

Now go to your Github repository, refresh to see your newly committed files.