More R Markdown

Tate, Levi

10/27/2020

## RMarkdown Coding Tips

Markdown is an **easy to use** format for writing reports. It resembles what you naturally write every time you compose an email. In fact, you may have already used markdown *without realizing it*. These websites all rely on markdown formatting

* [Github](https://github.com)
* [StackOverflow](https://stackoverflow.com)
* [Reddit](https://www.reddit.com)

### Chunks of Code

It is always a good idea to include titles for your chunks of R code. This allows other users to quickly find the code you may be referencing when working with other developers. Additionally, within each chunk of code, there is a number of options for how your code will be handled when you knit a file. These are can be included in “{r }” where you include the title.

Options include:

* **eval = FALSE** prevents the code from being evaluate (not run and no output created). This is useful if you want to stop a chunk of code from running but you will probably not use this in the course.
* **include = FALSE** runs the code, but doesn’t show the code or results in the final document. Use for setup code that you don’t want cluttering your report.
* **echo = FALSE** prevents code (but not results) from appearing in the finished report. Useful when presenting to management so the code doesn’t show on report.
* **message = FALSE** or **warning = FALSE** prevents messages or warnings from appearing in the output file
* **results = ‘hide’** hides printed output.
* **fig.show = ‘hide’** hides plots.
* **error = TRUE** causes the render to continue even if the code returns an error (rarely used).

###Packages and Data Imports in R

A good habit is to include all the libraries and data import statements at the beginning of your R Markdown document. This is commonly the first chunk of code that you create in the Markdown file.

A few tips when working with packages in Markdown include:

* Don’t install a package in your Markdown file. This will install the package everytime the file is knit. To load new packages, do this in the console outside of the Markdown file.
* If a new package does need to be installed, include this in the comments at the top of your R Markdown file.
* You only need to load the library once in the Markdown file. So, if you are using tidyverse, you will only have library(tidyverse) included once at the beginning.
* When you load some packages, they actually have multiple packages included in the load. For example, when you load tidyverse, this will load ggplot2, dplyr, tidyr, readr, etc.

### Examples

For this project, we will be using the tidyverse which you just installed in a previous assignment. Remember that when you load a package in R, it will remain loaded throughout the program (you don’t need to continually load the package).

Below is an example of loading the package and importing the data:

library(readr, tidyverse)  
US\_Population <- read\_csv("US\_Population.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## .default = col\_character(),  
## ESTIMATESBASE2010 = col\_double(),  
## POPESTIMATE2010 = col\_double(),  
## POPESTIMATE2011 = col\_double(),  
## POPESTIMATE2012 = col\_double(),  
## POPESTIMATE2013 = col\_double(),  
## POPESTIMATE2014 = col\_double(),  
## POPESTIMATE2015 = col\_double(),  
## POPESTIMATE2016 = col\_double(),  
## POPESTIMATE2017 = col\_double(),  
## POPESTIMATE2018 = col\_double(),  
## POPESTIMATE2019 = col\_double(),  
## NPOPCHG\_2010 = col\_double(),  
## NPOPCHG\_2011 = col\_double(),  
## NPOPCHG\_2012 = col\_double(),  
## NPOPCHG\_2013 = col\_double(),  
## NPOPCHG\_2014 = col\_double(),  
## NPOPCHG\_2015 = col\_double(),  
## NPOPCHG\_2016 = col\_double(),  
## NPOPCHG\_2017 = col\_double(),  
## NPOPCHG\_2018 = col\_double()  
## # ... with 11 more columns  
## )  
## ℹ Use `spec()` for the full column specifications.

View(US\_Population)