Module 2.2: Learn - Coding

2.2 Coding

In this module, we are continuing to work this R tutorial in anticipation of using R code on data for the research project in Module 3:

R for Reproducible Scientific Analysis

https://swcarpentry.github.io/r-novice-gapminder/index.html (https://swcarpentry.github.io/r-novice-gapminder/index.html)

Recall that there is an option to view all episodes on one page at the bottom of the site, if that makes it easier to look back and forth.

Assignment: R Programming Tutorial, Part 2

Please continue on and do the following episodes (skip 12, 13, and 14):

1. Summary and Setup
    - If you don't remember how to fire up RStudio, refer to Module 1 Learn Coding
2. Episode 7: Control Flow
    - This episode shows you how to use conditionals and loops to execute different code based on the value of a specific variable
3. Episode 8: Creating Publication-Quality Graphics with ggplot2

- This episode shows how the basics on how to use a data visualization package called ggplot2, which we v visualize data in the CCLE

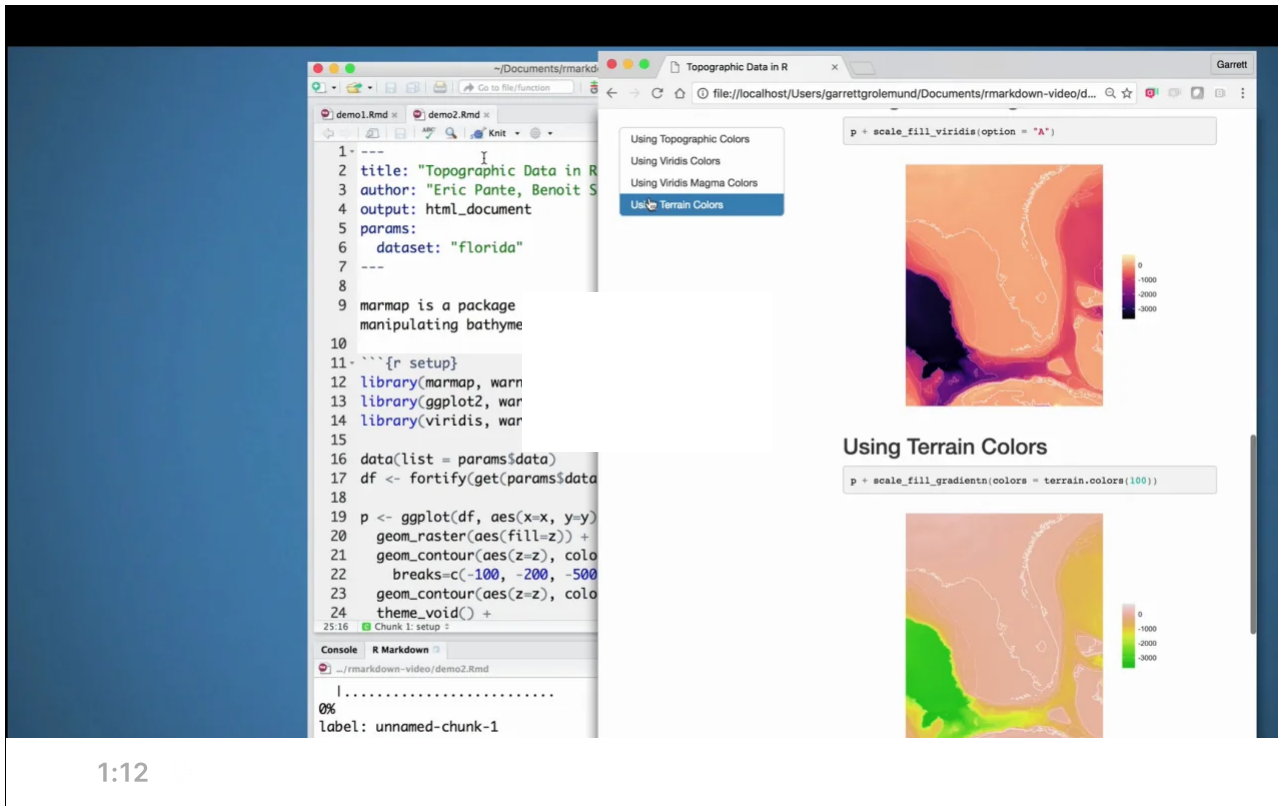
4. Episode 9: Vectorization
    - This episode shows some tricks you can do in R to do operations on vectorized data as an alternative to u through data
    - These tricks can be difficult to read and understand so the template code provided for you by the instructo them, but they are good to know about
5. Episode 10: Functions Explained
    - This episode shows you how functions are written and used in R
    - We will be using a lot of functions written by others in packages you can install from public repositories, so understand how they work
6. Episode 11: Writing data
    - This episode shows how to save output from ggplot2 and how to write out a variable into a file
7. Episode 15: Producing reports with knitr
    - This episode shows how to print reports in knitr
    - Template code written by the instructors for use in this class will be provided as R Markdown files, which y modify and print as a report to be handed in as an assignment
8. Episode 16: Writing good software
    - This episode gives good advice on how to make your code more readable, usable, and reproducible

Feel free to do the episodes we skipped for your own learning if you have time.

Resources for using R Markdown (Rmd)

RMarkdown (Rmd) is a simple formatting syntax that allows you to save, execute (or “run”), and print R code into Word documents. The user-friendly interface of RStudio gives you the ability to connect data to code that applies visualizations that help you interpret your data. In addition, when your code is complete, you can export reports re reproducing. This allows you to share your work with an audience who can then replicate exactly what was done you’re reporting.

This video introduces the basic parts of an Rmd and the benefits of using it:



Video. What is RMarkdown?

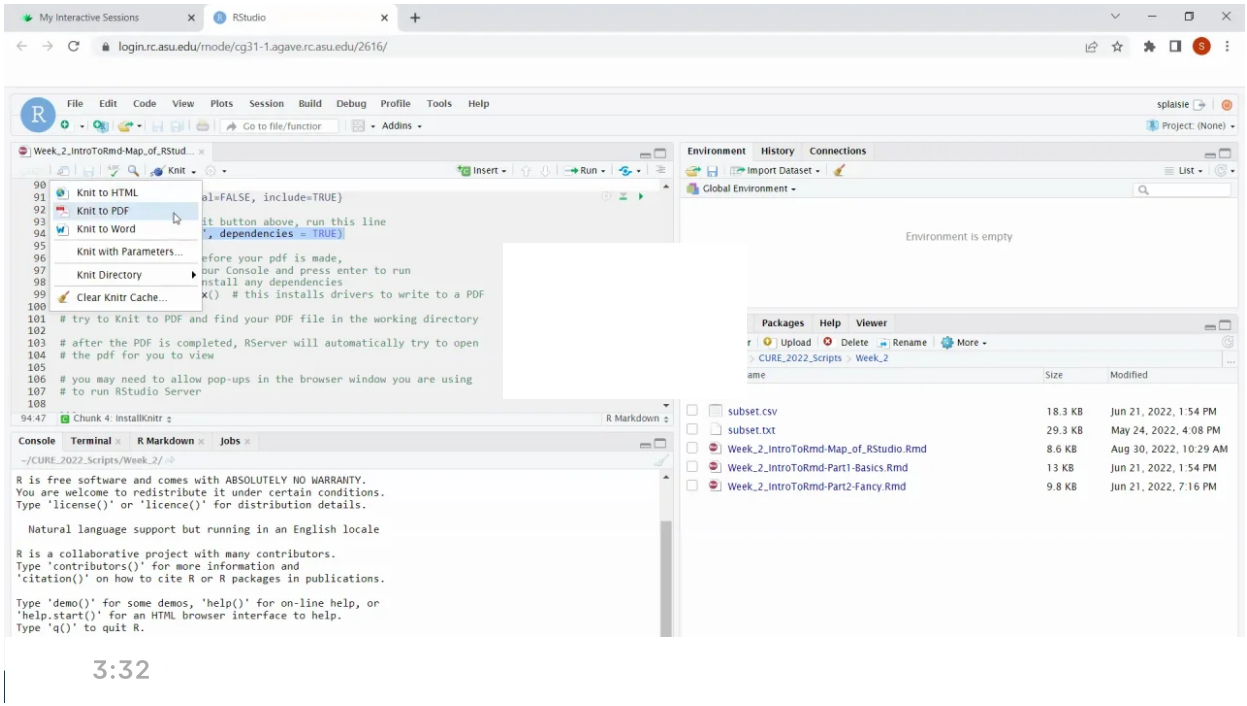
This video discusses the RMarkdown authoring framework, types of content to run and display text or code, and how you can use it to replicate and share your work in different formats.

View transcript. (https://canvas.asu.edu/courses/122165/files/54792246?wrap=1) **(https://canvas.asu.edu/courses/122165/files/54792246/download?download_frd=1)**

Creating Reports from R Markdowns (Rmds)

By writing your R code in an R Markdown format, not only can you include important descriptive information to he understand your code and you can also easily print a report that can be shared. This is done using the Knitr pack specific rendering packages to help print images of the outputs you generate. Knitr runs all the code in your Rmd

description, code, and output to a report. Knitr can print reports in many file formats, which can be specified in the header.



Video. Knitting Reports in RMarkdown.

This video shows how to knit a RMarkdown to save in different formats.

View Transcripts. (https://canvas.asu.edu/courses/122165/files/54792243?wrap=1) **(https://canvas.asu.edu/courses/122165/files/54792243/download?download_frd=1)**

The video above shows that you can click the Knit button (Knit) at the top of the tab and then select the desi most common are pdf, html, and Word document. If you look in your Files tab, you should see the report saved ir the Rmd file with the same name as the Rmd file but with the file suffix changed to indicate the output format (.htı option that is best to use if you are new to Rmd or if printing to other formats is giving you errors is HTML, since t any other packages or dependencies to work.

The above video describes printing to PDF, but if you would rather print to another format, just use the pull down button to change your output format or change the output option in the header of the Rmd. If you would like to pr

errors, it could be because you do not have the underlying functions needed to generate a PDF installed properly to install `tinytex`, a package that allows you to create PDFs.

```
install.packages('knitr', dependencies = TRUE)
tinytex::install_tinytex()
```

Notes

It is important to note that if you change your code and press Knit to print a new report, the same output type will automatically and the old report will be overwritten. If you want to save multiple versions of a printed report, you c report before knitting again using the button with a blue arrow in the Files tab to rename it so it is not overwritten.

In Rmarkdown it will ask you to denote in-line code chunk with the following: `` `r `` and `` ` ``.

Note that they are referring to a forward tick `` ` `` and not a single quotation mark `'`.