**GitHub Desktop basics**

**Creating a new repo**

Let’s create a brand-new repo to store our R project in.

1. Go to GitHub.com and select “New”   
   A screenshot of a computer

   Description automatically generated
2. Create the repo:
   1. Give the repo a name and say something about it.
   2. You can set it to private or public, your choice
   3. Add a readme file
   4. Select “Create repository”  
      A screenshot of a computer

      Description automatically generated
3. You can now clone the repo to your computer: Code > Open with GitHub Desktop
   1. Select where on your computer to store the repo
4. You’ve got a few options for how to access this repository’s data. You should see this in the middle of your Desktop session:   
   A screenshot of a computer

   Description automatically generated
5. Choose the second option to have a File Explorer session open up to the folder where this repo is located.
6. From you can open up your Readme file or even and make a change. Make sure to save your work!
   1. Save your changes to the file, then head back to GitHub Desktop.
7. You will see a list of changed files on the left, along with the contents of what’s changed to the right.
   1. Add a commit message and summary to the lower-left, then select “Commit to main.”  
      A screenshot of a computer

      Description automatically generated
8. You’ve committed or staged your changes to the main branch. You could continue committing locally if you’d like, but let’s go ahead and push this to origin.:  
   A screenshot of a computer

   Description automatically generated
9. Click on “View on GitHub” or otherwise get back to your repo to see the changes:   
   A screenshot of a computer

   Description automatically generated
10. Now let’s try this in reverse: make a change on GitHub.com and have it reflect on your computer. Go ahead and make some tweak to your readme file:  
    A screenshot of a computer

    Description automatically generated
11. Go back to your GitHub and select “Fetch origin” to check for any changes on GitHub.com (In this case, that’s the origin; you’re the remote!)  
    A screenshot of a computer

    Description automatically generated
12. Now click on “Pull Origin” (sound familiar)?  
    A screenshot of a computer

    Description automatically generated
13. You can now view these changes locally.
14. Let’s do the same thing with some R files and an R project!

**R Markdown: Helpful hints**

* The first part of the document is called the YAML (Yet Another Markup Language). This contains metadata about the file. You can give the document a title and author. As you export your file to different formats, those will display in the output section.  
  A screenshot of a computer

  Description automatically generated
* As you insert headers into the document, you will see button to the left to expand/contract the file:  
  A screenshot of a computer

  Description automatically generated
* The C and Run icons will let you, among other things, insert and run chunks, respectively:  
  A screenshot of a computer

  Description automatically generated
* You can customize the settings of chunks; we’ll use the following (as in previous screenshot). See more at the [official cheat sheet](https://rmarkdown.rstudio.com/lesson-15.html):
  + echo = FALSE will stop code from being shown in the output document
  + warning = FALSE will stop warnings from being shown in document
  + include = FALSE will run the chunk, but no code or output will be shown in the document.
  + message = FALSE will stop any messages being shown in the output document
* To view a preview of the document side-by-side with the R Notebook, click on the gearwheel of your document and select “Preview in Viewer Pane.” Then click “Preview” above your doc. The preview will show up in the Viewer pane.
  + This preview does not update on its own; to see subsequent changes to the doc, click Preview again or save the doc.   
    A screenshot of a computer

    Description automatically generated
* When you are ready to export the document, click under Preview > Knit to PDF (for example). You will see a log of the export process in the R Markdown tab. The document by default will be exported to wherever your R Notebook is.   
  A screenshot of a computer

  Description automatically generated
* The tidy() function can be used to present the result of many statistical tests in a more tabular format:  
  A screenshot of a computer

  Description automatically generated
* You can use in-line blocks for dynamically-rendered text using `r objectname` like so:  
  A screenshot of a computer

  Description automatically generated
* You can add text-only comments that aren’t rendered in the doc or run by R by using the following notation (comes from HTML):

<!---

Comments go b/w the lines

--->

They’ll look like this in the notebook:

A screenshot of a computer

Description automatically generated

* If you’re just looking to try out some code before including it in the document, try using the console:  
  A computer screen shot of a computer

  Description automatically generated
* If you are having an issue with changes to your document not being shown in preview, there is likely an error with your notebook causing it not to build properly. Check the header of your doc for an error message or knit the file to get further details.  
  A screenshot of a computer

  Description automatically generated
* If you have set up everything correctly, all the code runs, etc, you should be able to render your document as PDF, HTML and more.