

Computational Social Science

File management and Github

Dr. Thomas Davidson

Rutgers University

February 1, 2024

Plan

- ▶ File management
- ▶ Github
- ▶ Homework 1

File management



Source: The Verge, 2021.

File management

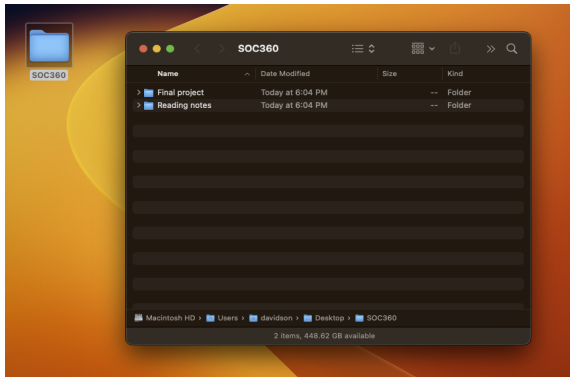
- ▶ File management in this class
 - ▶ Keeping track of course materials
 - ▶ Working on homework assignments
 - ▶ Organizing final projects
 - ▶ Reproducibility!

Organizing your files

- ▶ Make a directory to contain all materials for this class
- ▶ Store this somewhere practical
 - ▶ e.g, `/Users/me/Documents/SOC360`,
`/Users/me/Desktop/Classes/SOC360`
 - ▶ Do not just leave files in your Downloads directory!

Organizing your files

- ▶ Within this directory, make a separate directory for the class materials, readings, and homework assignments. It might look something like this.



Github

- ▶ Github is a version-control system
 - ▶ This allows you to easily control and manage changes to your code (similar to Track Changes in Word)
 - ▶ It can facilitate collaboration
 - ▶ Version-control helps to ensure reproducibility
 - ▶ It makes it easy to share code
- ▶ Github is *not* designed as a place to store large datasets (100Mb file size limit)

Terminology

- ▶ A Github *repository* (or *repo* for short) contains all files and associated history
 - ▶ A repository can be public or private
 - ▶ Files should be organized into folders
 - ▶ Github can render Markdown files (suffix `.md` in Markdown), useful for documentation
- ▶ Github repositories exist online and you can *clone* them to your local computer

Using Github

- ▶ You can interact with Github in several different ways:
 - ▶ RStudio integration (recommended)
 - ▶ Github Desktop (redundant if using RStudio)
 - ▶ Through your browser (not recommended, but viewing is fine)
 - ▶ Using the command line (recommended for advanced users)

Using Github

Follow the instructions on the course website to set up Github with RStudio: <https://github.com/t-davidson/SOC360-CSS/>

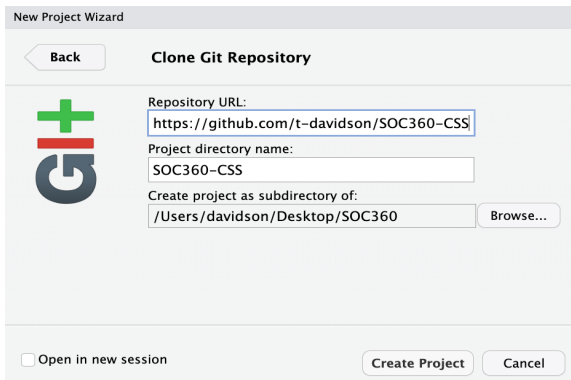
1. Register for a Github account
2. Install Git
3. Sync your Github account with RStudio

Cloning the course materials

- ▶ Once you have this set up, navigate to the course website on Github and copy the URL:
 - ▶ <https://github.com/t-davidson/SOC360-CSS>

Cloning the course materials

- ▶ In RStudio, click File > New Project > Version Control > Git




The screenshot shows the 'New Project Wizard' dialog box in RStudio, specifically the 'Clone Git Repository' step. The dialog has a title bar 'New Project Wizard' and a 'Back' button. On the left is the Git logo (a green plus sign over a grey 'G'). The main area contains three text input fields: 'Repository URL:' with the value 'https://github.com/t-davidson/SOC360-CSS', 'Project directory name:' with the value 'SOC360-CSS', and 'Create project as subdirectory of:' with the value '/Users/davidson/Desktop/SOC360'. A 'Browse...' button is next to the last field. At the bottom, there is a checkbox 'Open in new session' which is unchecked, and two buttons: 'Create Project' and 'Cancel'.

New Project Wizard

Back

Clone Git Repository



Repository URL:

Project directory name:

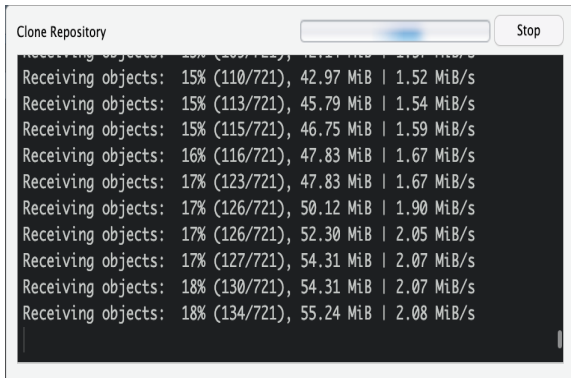
Create project as subdirectory of:

☐ Open in new session

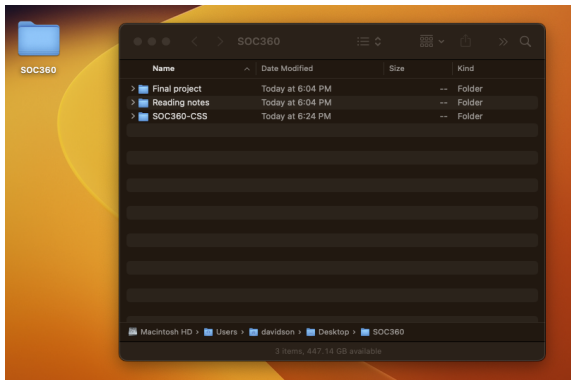
Cloning the course materials

- ▶ Paste the URL into the Repository URL field
- ▶ Write a suitable name in the Project directory name field
 - ▶ This will be the name of the folder that is created on your computer
- ▶ Choose a location to store the repository on your computer
 - ▶ Recommend using the folder we created earlier
- ▶ Then click Create Project

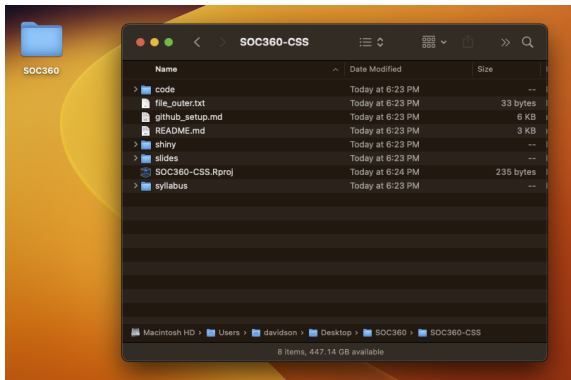
Cloning the course materials



Cloning the course materials

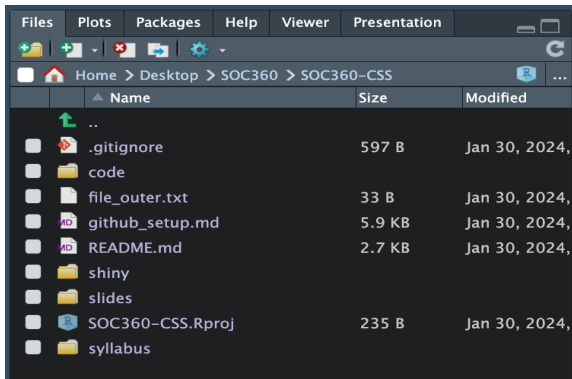


Cloning the course materials



Navigating files using RStudio

You can also use the Files pane in RStudio to navigate folders.



Opening this .Rmd file

- ▶ Navigate to `slides/` and open `lecture5-file-management-and-github.Rmd` by double-clicking the file

Navigating files using RStudio

Run the `getwd()` command to show the current working directory in R.

```
getwd()
```

Navigating files using RStudio

Run the `setwd()` command to change your working directory. Try going one step back from the current directory using `..`. You can then run `getwd()` to verify it has changed.

```
setwd("..")  
getwd()
```

Navigating files using RStudio

- ▶ When a `.Rmd` file is opened RStudio defaults to the directory where the file is contained.
 - ▶ If you run `setwd()` it will change within a chunk, but other chunks will revert back to the working directory.

Navigating files using RStudio

The working directory is important when considering loading files. Navigate to the code directory. You can use the `list.files` function to see a list of the files in the current directory.

```
setwd("../code/")  
list.files()
```

Using file paths

- ▶ The working directory is critical because it defines a path we need to use to load files. Different files will fall into one of three groups:
 1. File contained in the working directory.
 2. Files contained in outer directories.
 3. Files contained in inner, nested directories.

Files contained in the working directory

If a file exists in the working directory, you will see it when running `list.files`. It can be loaded by using the file name.

```
library("tidyverse")  
read_file("file.txt") %>%  
  print()
```


Files contained in an outer directory

If a file is contained in an upper level directory, you need to use the `..` to escape the current directory. For each step out from the current directory you need to add another `../` to the file path. In this case, the file is contained one level above the current directory.

```
read_file("../file_outer.txt") %>%  
  print()
```

Files contained in an inner directory

If a file is contained in an inner directory, you need to add the name of the directory to the file path.

```
read_file("nested/file_nested.txt") %>%  
  print()
```

Files contained in an inner directory

If a file is contained in an inner directory, you need to add the name of the directory to the file path.

```
read_file("nested/nested2/file_nested2.txt") %>%  
  print()
```

Exercise

Modify the path below to find and print the contents of the hidden file in the course repository.

```
read_file("") %>%  
  print()
```

File navigation

Common errors

- ▶ It is common to get errors if you misspecify a path when trying to load a file. You will get an error like the following:
 - ▶ Error: 'filename' does not exist in the current working directory ('directory').
- ▶ If this happens, check whether the file is in your current working directory.
 - ▶ You can always use your Files tab or your normal file viewer to verify the location.

Homework 1

Github classroom

- ▶ Homework 1 released on Github Classroom
 - ▶ Follow link on Canvas in the Module for this week
 - ▶ Select your Rutgers NetID
 - ▶ This will take you to a Github repository

Homework 1

Cloning the repository

- ▶ Clone the repository using the same process as above and store within the class folder
 - ▶ Copy the URL
 - ▶ Start a new project and paste the URL
 - ▶ Store it somewhere sensible
 - ▶ DO NOT store this inside the repository for the course materials

Homework 1

Working on the homework

- ▶ The homework is contained in a `.Rmd` file. All instructions are located in this file.

Homework 1

Submitting the homework

- ▶ Submission instructions are included at the bottom of the homework file
- ▶ There is also a guide on the Github wiki on the course website

Next week

- ▶ Application Programming Interfaces (APIs)
- ▶ Collecting and analyzing data from Spotify and other platforms
 - ▶ If you don't have Spotify, please make a free account before class