**Setup (only needs to be done once per new user):**

1. Install R and RStudio on your computer. R is an open source statistical programming language. RStudio is an Integrated Development Environment (IDE) for developing R code.

For Windows:

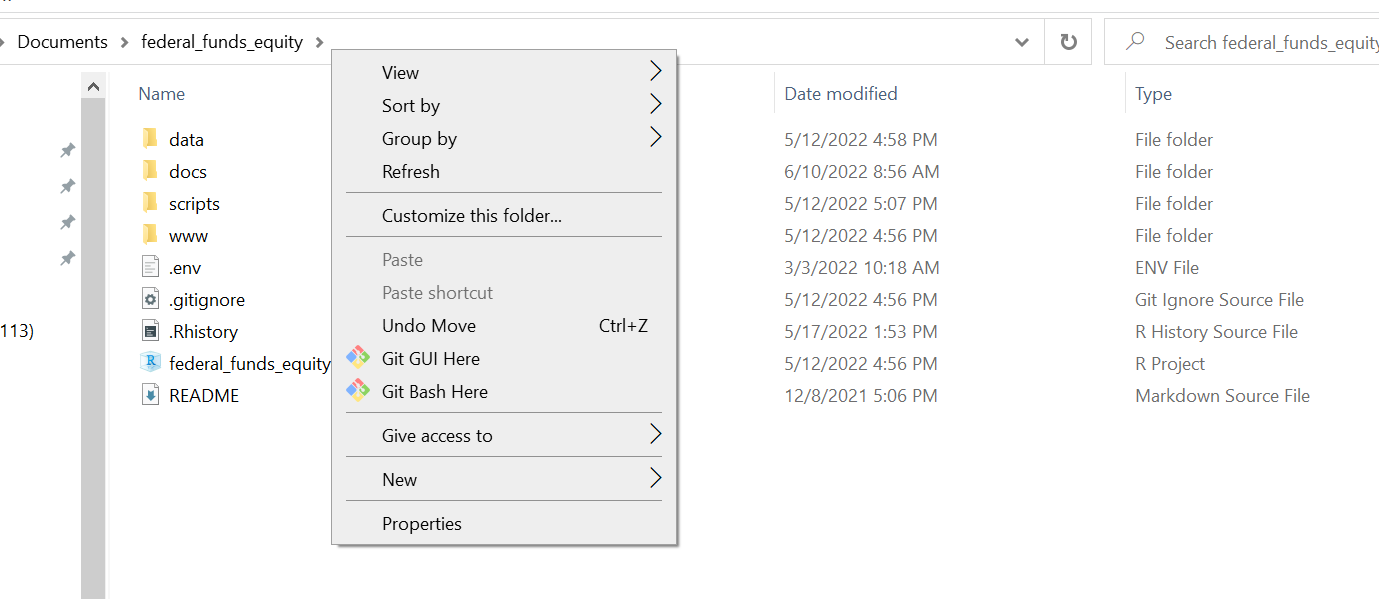
* 1. Install R
     1. Navigate to the [CRAN website](https://cran.r-project.org/).)
     2. Click "Download R for Windows"
     3. Click "base"
     4. Click "Download R 4.X.X for Windows"
     5. Follow the installation instructions. Accept all defaults and install R.
  2. Install RStudio
     1. Navigate to the [RStudio website](https://rstudio.com/products/rstudio/download/#download)
     2. Download the RStudio Desktop installer for Windows. It should be something similar to "RStudio-1.4.1717.exe".
     3. Follow the installation instructions. Accept all defaults and install RStudio.
     4. Open RStudio. If successful, then R and RStudio are installed.

For Mac:

1. Install R
   1. Navigate to the [CRAN website](https://cran.r-project.org/).)
   2. Click "Download R for (Mac) OS X"
   3. Select the .pkg link under "Latest Release"
   4. Follow the installation instructions. Accept all defaults and install R.
2. Install RStudio
   1. Navigate to the [RStudio website](https://rstudio.com/products/rstudio/download/#download)
   2. Download the RStudio Desktop installer. It should be something similar to "RStudio-1.4.1717".
   3. Follow the installation instructions. Accept all defaults and install RStudio.
3. Install Git

For Windows:

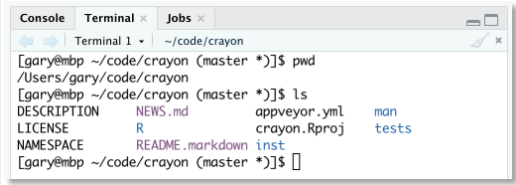
1. Install [Git for Windows](https://git-scm.com/downloads). Keep all default selections and **be sure to install Git Bash**. Git Bash is a command line for Windows that is useful for using Git. After installation, you should be able to right click in a finder window and select “Git Bash Here” as shown in the screenshot below.



1. Set your default terminal in RStudio to Git bash (this is only necessary for Windows). Open RStudio and go to **Tools -> Global Options -> Terminal**. In the dropdown box for **New terminals open with**, select Git Bash.

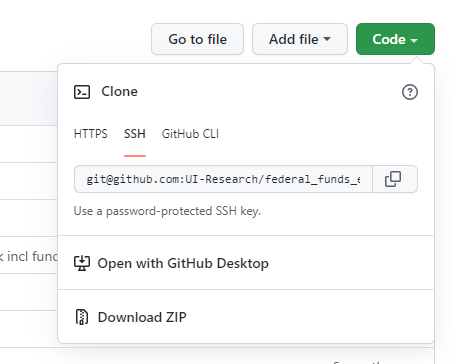
For Mac:

1. Install [Git for Mac OSX.](https://git-scm.com/downloads) Mac already has the terminal. Open the terminal from applications and type “git” and hit enter. If everything is properly installed, a list of possible command will show up.
2. Configure Git in RStudio
   1. Open R Studio and navigate to the Terminal tab which is next to the Console tab as shown below. Click on the Terminal tab and a new terminal session will be created (if there isn't one already). If the tab isn't visible, go to **Tools -> Terminal -> New Terminal**. *If you don’t see a dollar sign before the cursor in your terminal on Windows, you may need to restart RStudio for the global setting of creating new terminals with Git Bash to take effect.*

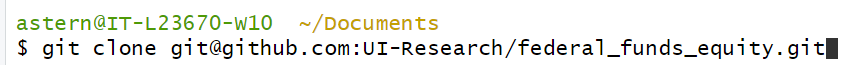


* 1. Configure your GitHub account by entering the code below in your terminal window. Use your GitHub user name for the user.name, and the email address linked to your GitHub account for the user.email.
  2. Create GitHub credentials (SSH keys) following [these instructions](https://happygitwithr.com/ssh-keys.html) from Jenny Bryan to create our key and add it to GitHub. This key is what will give you permission to make edits to the files on our GitHub repository. You will copy and paste the commands from the instructions into your terminal in RStudio.
     1. When you get to the step “**Provide public key to GitHub**” we recommend that you obtain your public key by running the following in your terminal: “cat ~/.ssh/id\_ed25519.pub”. This will print the public key to the terminal. You can simply highlight and copy the public key and then paste into GitHub following the instructions [here](https://happygitwithr.com/ssh-keys.html#on-github).

1. Clone GitHub Repository
   1. Go to the github repository URL in your browser and click on the green “Code” button in the upper right corner. Copy the text starting with [git@github.com](mailto:git@github.com) (you can just click on this button  to copy the text to your clipboard).



* 1. Go back to RStudio and in your terminal, type the following command, where you’ll type “git clone” and then paste in the text you just copied. Note that this will create a folder with the name of the repository that contains its files in the folder shown by the file path in your terminal (in the case below, this would be the “Documents” folder). If you want to copy the files in a different location, you can change to that location using the [“cd” command.](https://stackoverflow.com/questions/8961334/how-to-change-folder-with-git-bash)

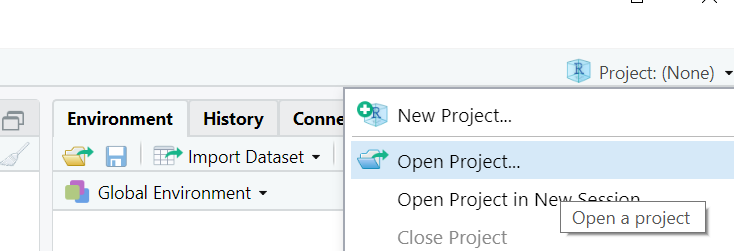


**You’re all done with setup!!**

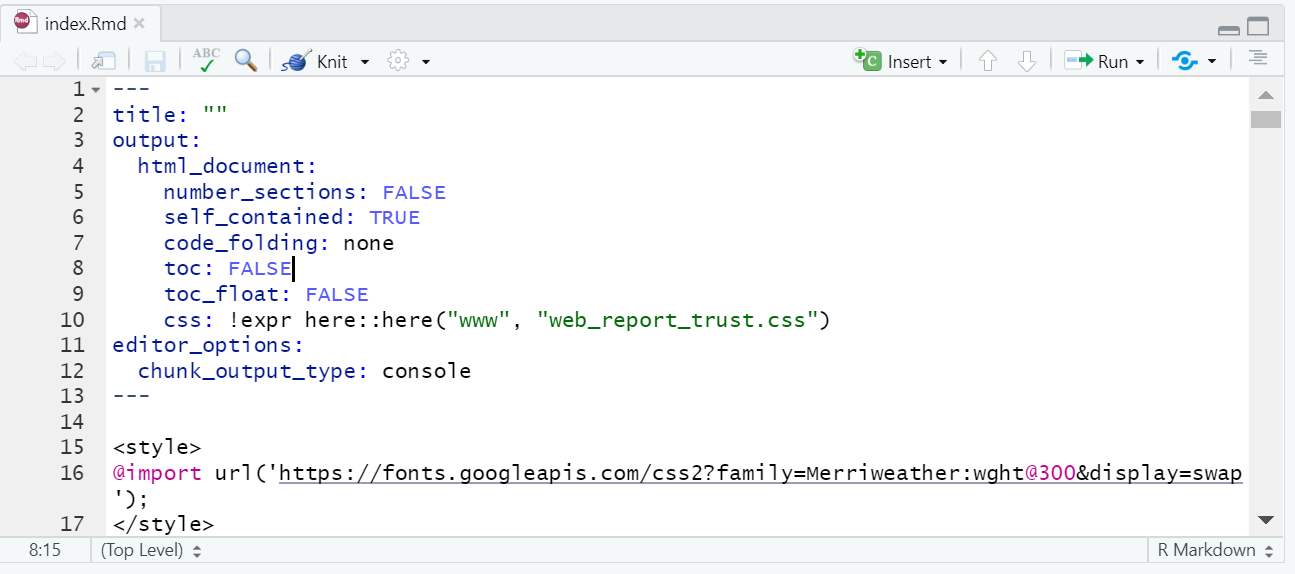
**Dashboard Update Instructions:**

*These instructions assume that you have already updated the input spreadsheet(s) that are saved in the project folder and saved them back in the same location using the same filename(s). If you change the location or filename(s) of the input files, the update will not work.*

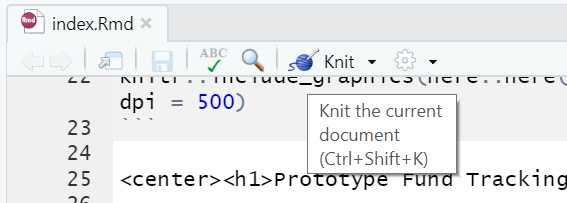
1. Open RStudio. In the top right corner, click the Project Icon (shown with the text “Project: (None)” below) and select “Open Project” as shown in the screen shot below. Navigate to the folder where you cloned the GitHub repository and select the RProject file.

****

1. Download (or “pull” in GitHub terms) the latest files from GitHub. Navigate to the terminal window following the instructions in the setup section above. By opening the project, your terminal should automatically be in the correct directory. You can pull the files by running “git pull” in your terminal. **You should always run git pull before making any edits to the dashboard or data files.**
2. In RStudio, select **File > Open File** and then select **index.Rmd** to open. The opened file should look like this:

****

1. (Optional) If you are making any changes to the dashboard itself, such as updating the introductory text or glossary, do that directly in index.Rmd and save the file when your editing is complete. Note that the document is written in markdown and thus all styling will need to be done in markdown syntax. This guide to [mastering markdown](https://guides.github.com/features/mastering-markdown/) has useful guidance on markdown syntax. See the specific example below for adding or changing a policy topic.
2. You will “knit” the index.Rmd file to re-create the dashboard with the updated data and content by clicking the “knit” icon as shown below indicated by arrow. This may take a minute or two to fully run.



1. When the file has finished knitting, you should upload, or “push” in GitHub terms, the updated data, index.Rmd file, and index.html file to GitHub. In your terminal window, run the following commands in the order given below, replacing “add a message here” in the second command with a message explaining the changes made, such as “July 2022 data updates”:
   1. git add index.Rmd index.html data/dashboard\_program\_tracker.xlsx
   2. git commit -m “add a message here”
   3. git push

After you push the files to GitHub, the GitHub page will automatically update with the latest version of the dashboard (though it may take a few moments to do so).

**You’re all done with the update!**

**Update Example: Adding or Changing a Policy Topic**

1. First, update the input data spreadsheet as outlined in this document [add link].
2. Follow steps 1-3 of the update dashboard instructions below.
3. If you’re changing the name of an existing policy topic, simply search for the name of the policy topic you wish to change using the find tool (you can open using **Edit > Find**) and replace with the new name, which must exactly match the policy topic in the spreadsheet. For example, if you wanted to change “Community Investment” to “Neighborhood Investment”, you would perform the following two find and replace operations:
   1. Find: “Community Investment”, Replace: “Neighborhood Investment” (should make 2 replacements)
   2. Find: “community\_investment”, Replace: “neighborhood\_investment” (should make 8 replacements)
4. To add a new policy topic (e.g. “Public Health”), open the file “new\_topic\_template.Rmd” and run the following find and replace operations:
   1. Find: “[New Topic]”, Replace: “Public Health” (should make 2 replacements)
   2. Find: “[new\_topic]”, Replace: “public\_health” (should make 8 replacements)
5. Copy lines 6-198 (the end of new\_topic\_template.Rmd) and paste into index.Rmd right above the following line (“## State/Local Discretionary Funding {.tabset}”).
6. Once all changes are made, follow steps 5-6 of the update dashboard instructions above.