**Github in class activities**

**I. Creating a repository with code and data**

1. Go to the United States Diabetes Atlas (<https://gis.cdc.gov/grasp/diabetes/DiabetesAtlas.html>), click on Social Determinants of Health, click on table above the map, and download the CSV file
2. Put the CSV file in a directory/folder on your desktop
3. Initialize that directory/folder for version control using the command line
4. Put the CSV file under version control using the command line
5. Upload the CSV file to a US diabetes project repository you create on Github using the command line with a commit message
6. Edit the file in your local directory/folder removing rows 1 and 2
7. Commit those changes back to Github using the command line with a message that indicates what was changed
8. Upload that file to Github using the command line
9. View the changes on Github. What do you see?
10. View your repository to make sure all is in order (you should see the CSV file with two versions)
11. Create an R Markdown code file for the project in R Studio with the title of the project, your name, and the date
12. Import the diabetes data file into R studio from your Github site. How many rows and columns are there?
13. Using code, find the county with the highest diabetes rate (annotate the code to indicate what it does)
14. Publish the code on Github in your project repository using the command line
15. Create a README.Rmd file that gives a description of the repository file, where it came from, and when it was downloaded, publish that file on Github in your project repository using the command line.

**II. Collaborative Coding**

**Parts for pairs:** Owner, collaborator 1

1. Pick someone to collaborate with at your table and assign owner and collaborator parts.
2. The owner will use their diabetes repository and add the collaborator
3. The collaborator will clone the diabetes repository
4. Collaborator 1 will make a change to the coding file, writing code to take the mean of the overall SVI and push it back up
5. The owner will pull the changes, make another change to the file (e.g. delete Alabama) and push it back up with a commit message
6. Collaborator 1 will pull the new changes
7. **Extra optional challenge:** Collaborate 1 in consultation with the owner decides that AL should not be deleted so reverts to the previous version using commands in the command line (this may be helpful: <https://www.techtarget.com/searchitoperations/answer/How-to-roll-back-Git-code-to-a-previous-commit>)

**Parts for trios:** Owner, collaborator 1, collaborator 2

1. Pick someone to collaborate with at your table and assign parts (owner, collaborator 1 and collaborator 2)
2. The owner will use their diabetes repository and add the collaborators
3. The collaborators will clone the repository
4. Collaborator 1 will make a change to the coding file, writing code to take the mean of the overall SVI and push it back up
5. Collaborator 2 will pull the changes, make another change to the file (e.g. delete Alabama) and push it back up with a commit message
6. Collaborator 1 and the owner will pull the new changes
7. **Extra optional challenge:** The group decides that AL should not be deleted so one person is tasked to revert to the previous version using commands in the command line (this may be helpful: https://www.techtarget.com/searchitoperations/answer/How-to-roll-back-Git-code-to-a-previous-commit )

Gdshhdsghg

Ddshdhhds

Sdhdhhjsdhjhsd

Dbbdsdhhhjsd

Dhsjdjj sdhhd shsdjd

Gdsdshd sshhjsj shdjs

Nsnhjjhds bnbsnbsdb nbsndbd snbdb