Problem Set 2

Overview:

One of the goals for this quarter is to get you comfortable using git and GitHub. In this problem set we will be practicing more git/GitHub workflow basics, manipulating data in R, and creating GitHub issues. We are asking you to create a git repository on your local computer which you will later connect to a remote repository on GitHub. This local repository will have an `.R` file where you will read in data and practice manipulating this data. We recommend doing the reading prior to completing the problem set.

Part I: Concepts & Definitions (2 pts)

1. What hidden directory is created whenever a git repository is created?

   `.git/`

2. Describe what git objects are, what they are identified by, and where they are stored.

   `Git objects stores all data in the repository. They are identified by a unique hash and are stored`

3. What are the 4 types of git objects?

   `blob, tree, commit, tag`

Part II: Command line & Git (command-line bullshittery) (6 pts)

1. Using your command line interface (CLI) (e.g. Git Bash, terminal), create a new folder called **lastname_ps2**. Be intentional about where you create this folder (hint: change directories to where you want to save this folder first). Then, change directory into the **lastname_ps2** folder.

   Write the commands you used here (to create the folder and change directory):

   ```
   mkdir lastname_ps2
   cd lastname_ps2
   ```

2. Turn **lastname_ps2** into a git repository and write the command you used here:

   `git init`

3. Use the `echo` command to output the text `"# YOUR NAME HERE"` and redirect it using `>` to a file called `problemset2.R` (hint: refer to example code in lecture). Write the command you used here:

   `echo "# YOUR NAME HERE" > problemset2.R`

4. Check the status of your repository. Write the command you used here:

   `git status`

   According to the output, under which heading is `problemset2.R` listed under?

   `Untracked files`

5. What is the git command to check what changes (i.e., differences) were made to `problemset2.R`?

   `git diff problemset2.R`

If you run this command now, do you see an output? Why or why not?

```
No output because this file has never been added to "index" before.
```

6. Add `problemset2.R` to the staging area and check the status. Write the commands you used here:

```
git add problemset2.R
git status
```

According to the output, under which heading is `problemset2.R` listed under?

```
Changes to be committed
```

7. Use a git command to compute the hash ID for `problemset2.R`. Write the command you used here:

```
git hash-object problemset2.R
```

What is the hash of the blob object?

```
60cb3d64770c4ef0fa511b6d16a4c481e53262ab
```

8. Use a git command to get the content, type, and size of the blob object. Write the commands you used and the outputs you got here:

```
git cat-file -p 60cb3d64770c4ef0fa511b6d16a4c481e53262ab
# YOUR NAME HERE
```

```
git cat-file -t 60cb3d64770c4ef0fa511b6d16a4c481e53262ab
blob
```

```
git cat-file -s 60cb3d64770c4ef0fa511b6d16a4c481e53262ab
17
```

9. Commit the file and check the commit log. Write the commands you used here:

```
git commit -m "initial commit"
git log
```

According to the output, what is the hash of your commit?

```
b56376f73c037e34704b1ffd3cefb1b2a3b68b66
```

10. Use a git command to get the content, type, and size of the commit object. Write the commands you used and the outputs you got here:

```
git cat-file -p b56376f73c037e34704b1ffd3cefb1b2a3b68b66
tree 23d89277afe72b5ef82a11b71e413f32bb194259
author cyouh95 <25449416+cyouh95@users.noreply.github.com> 1586079139 -0700
committer cyouh95 <25449416+cyouh95@users.noreply.github.com> 1586079139 -0700

initial commit
```

```
git cat-file -t b56376f73c037e34704b1ffd3cefb1b2a3b68b66
commit
```

```
git cat-file -s b56376f73c037e34704b1ffd3cefb1b2a3b68b66
217
```

Part III: Manipulating data in R (4 pts)

1. Open `problemset2.R` in RStudio to edit the file and remove the comment containing your name at the top of the file.

2. Load data on off-campus recruiting events by public universities:

`load(url("https://github.com/Rucla-ed/rclass2/raw/master/_data/recruiting/recruit_school_somevars.R`

Each observation (row) in the data is a high school. The columns are various characteristics of the high school. There are also columns indicating the number of times the high school has been visited by a public university:

- `visits_by_100751` = University of Alabama
- `visits_by_126614` = University of Colorado Boulder
- `visits_by_110635` = UC Berkeley

3. Take some time to investigate the data:

   - How many rows and columns are there?

   - Check if there are missing values in the data
   - What variable(s) uniquely identify the data?

```
#load libraries
library(tidyverse)

#load data
load(url("https://github.com/Rucla-ed/rclass2/raw/master/_data/recruiting/recruit_school_somevars.RData

#glimpse(df_school) 21,301 rows & 26 columns

#Check missing values
sapply(df_school, function(x) sum(is.na(x)))
```

```
##         state_code        school_type            ncessch               name
##                  0                  0                  0                  0
##            address               city           zip_code          pct_white
##                  0                  0                  0                  0
##          pct_black       pct_hispanic          pct_asian     pct_amerindian
##                  0                  0                  0                  0
##          pct_other       num_fr_lunch     total_students      num_took_math
##                  0                158                  0               4103
##      num_prof_math       num_took_rla       num_prof_rla avgmedian_inc_2564
##               4251               4099               4219                624
##   visits_by_110635   visits_by_126614   visits_by_100751        inst_110635
##                  0                  0                  0                  0
##        inst_126614        inst_100751
##                  0                  0
```

```
#Variable that uniquely identify data = ncessch
df_school %>%
  group_by(ncessch) %>%
  summarise(n_per_group = n()) %>%
  ungroup() %>%
  count(n_per_group)
```

```
## # A tibble: 1 x 2
##   n_per_group     n
##         <int> <int>
## 1           1 21301
```

4. Pick 1 university and perform the following data manipulation:

   - Create a 0/1 dummy variable called `visited` that indicates whether the high school received a visit from the university of your choice (`0`=received no visits, `1`=received 1 or more visits)

- Filter observations to keep only high schools that are located in the same state as the university (hint: see `state_code` for high school state code and `inst_[univ]` for university state code)
- Subset your dataframe to include the following variables: `school_type`, `ncessch`, `name`, `total_students`, `avgmedian_inc_2564`, `visits_by_[univ]`

```
df_school <- df_school %>%
  mutate(visited = ifelse(visits_by_100751 > 0, 1, 0)) %>%
  filter(state_code == inst_100751 ) %>%
  select(school_type, ncessch, name, total_students, avgmedian_inc_2564, visits_by_100751, visited)
```

Part IV: GitHub & Git (5 pts)

1. Check the changes (i.e., differences) made to `problemset2.R`. How can you tell if a line has been added or removed?

   `(git diff problemset2.R)`

   `There is a "-" before any line that has been removed and a "+" before any line that has been added.`

2. Check the status of your repository. Write the command you used here:

   `git status`

   According to the output, under which heading is `problemset2.R` listed under?

   `Changes not staged for commit`

3. Add and commit `problemset2.R`. Write the commands you used here:

   ```
   git add problemset2.R
   git commit -m "second commit"
   ```

4. Modify `problemset2.R` again by adding a comment at the end of your file where you write down your guilty pleasure. Add the file to the staging area.

   Now let's say you are having second thoughts about committing this change. What command would you use to unstage this file?

   `(git add problemset2.R)`

   `git reset HEAD problemset2.R`

5. But in the end, you decide to go ahead and commit this change anyway. Re-add `problemset2.R` to the staging area and make a commit with the message `my guilty pleasure`.

   You regret it instantly! You remember that `git reset` and `git revert` are two commands to undo changes from a commit. What is the difference between them?

   `git reset undoes the commits and removes them from history, while git revert makes a new commit tha`

6. Let's say you decided to use `git revert`. Revert the `my guilty pleasure` commit and write the command you used here:

   ```
   git revert 1fd4a1d  # This is the hash of the commit you want to remove
   ```

   `NOTE: We can see this step in https://github.com/Rucla-ed/lastname_ps2/commits/master`

7. Log in to your GitHub account online and create a new private repository here: https://github.com/organizations/Rucla-ed/repositories/new

   Name it **lastname_ps2** and do NOT initialize it with a `README.md` file. Paste the link to your repository here:

`https://github.com/Rucla-ed/lastname_ps2`

8. Connect your local **lastname_ps2** repository to the remote and push your changes (hint: refer to Section 4.2.3 in the lecture). Write the commands you used here:

```
git remote add origin git@github.com:Rucla-ed/lastname_ps2.git
git push -u origin master
```

Part V: I got issues (2 pts)

1. Navigate to the issues tab for the **rclass2** repository here: https://github.com/Rucla-ed/rclass2/issues

   You can either:

   - Create a new issue posting a question you have about the class/problem set
   - Answer a question that another student posted
   - Create a new issue posting about something new you learned or figured out from this class
     - you must mention thw other members of your homework group, and assign yourself as assigne (please close the issue within 1 week).

   Paste the link to the issue you contributed to here:

Part VI: Wrapping up (1 pt)

1. Finally, add and commit this file you are working on (`problemset2.Rmd`) to your repository and push to the remote repository.
2. How much time did you spend on this problem set?