DSI: Unix Shell, Git and GitHub
Assignment 2 & Quiz: Git and GitHub

Part 1

Part 1 of Assignment 2 is a quiz. Please complete to the best of your ability. Notes are permitted. Please email your responses to the Instructor.

1. Check all that are TRUE about version control:
   - [x] Can revert files to a previous state
   - [x] Can compare changes over time
   - [x] Can see who modified something last
   - [x] Can recover lost files

2. What is the difference between centralized version control systems and distributed version control systems?

Centralized version control system only has one single server for all the versioned files for all collaborators. Whereas the distributed version control system allows collaborator to mirror the repository, so if a server dies, the repositories can be resotred from any collaborator.

3. What are the three states that files can reside in?
   - a) committed, changed, waiting
   - b) saved, changed, staged
   - c) committed, modified, staged
   - d) saved, modified, staged

4. What command initializes a new repository?
   - a) `git clone`
   - b) `git branch`
   - c) `git fork`
   - d) `git init`

5. What does `git diff` do?
   - a) compares the differences between the home directory and staging area
   - b) compares the differences between the working directory and staging area
   - c) compares the differences between the working directory and what's been committed
   - d) compares the differences between the staging area and what's been committed

6. How do you add a message to your commit? (select all that apply)
   - a) `git commit -m`
   - b) `git commit -messages`
   - c) `git commit`
   - d) `git commit -message`

7. How do you add a remote repo? (select all that apply)

- a) `git remote`
- b) `git add remote`
- c) `git clone`
- d) `git add clone`

8. What is the difference between `git pull` and `git fetch`?

git fetch will allow users to get new changes in the remote repository but it will not merge it to our work, while git pull will both fetch and merge the remote rabnd to the our working branch.

9. How do you switch branches?
   - a) `git checkout`
   - b) `git checkout -b`
   - c) `git branch -c`
   - d) `git branch`

10. Why are messages important? What would make a good commit message?

messages help us to keep track of all the commit we make, so that we know what changes has been done in each version of the repository and each commit, for us and other collaborators.

Great commit messages will include a short summary of changes made, with a subject line as the first line, separated from the body.

11. Please correct the merge shown below (both codes are suitable, neither has errors):

```
<<<<<<< HEAD
df.loc[df['sex'] == 'f', 'age'].mean()
=======
df.loc[df['sex'] == 'm', 'age'].mean()
>>>>>>> branch_1
```

This is a merge conflict that would come up when trying to merge two branches that have different codes on the same line.

This question shows the two conflicting codes together, separated by "=====". We can choose one of the code for the merge or re-write. Then try to merge again.

Part 2

1. `fork` and `clone` this class GitHub repo.
2. `push` your Assignment 1 to the folder labelled “assignment-2.” Your additions should include…
   - All components necessary to run Assignment 1
   - Proper folder structure (inputs, outputs, scripts)
   - A README.md file. The README should include components discussed in the workshop. Feel free to research good READMEs and add anything that you believe will add value to your README
3. Create a `pull request` to add your additions to the class repo.

Rubric:

Component	**Yes**	**No**
1. Repo contains all necessary components to run Shell script and has the correct folder structure		
2. README is comprehensive and includes components discussed in class plus at least one component learned from outside sources		
3. Pull request has been successfully requested without any merge errors		