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:
   - Can revert files to a previous state T
   - Can compare changes over time T
   - Can see who modified something last T
   - Can recover lost files T

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

Centralized Version Control Systems: Single server repository; risk of single point of failure.
Distributed Version Control Systems: Every user has a full local copy of the repository; allows offline work and provides redundancy.

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

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

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 Answer: B
   - 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` Answer: A and C
   - 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` Answer: A and C
- c) `git clone`
- d) `git add clone`

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

Git pull: Downloads changes from the remote repository and immediately merges them into the local branch.
Git fetch: Downloads changes from the remote repository but does not merge them; it lets you review changes first

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

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

Messages in version control are crucial for several reasons:
1) Clarity: They provide context about what was done and why, helping team members (and future you) understand the changes.
2) Tracking: They assist in tracking down when specific changes were made and for what purpose.

A good commit message should be:
1) Concise: Brief yet descriptive.
2) Informative: Clearly states what the commit does and why, not just how.
3) Consistent: Follows the team's or project's style guidelines.
4) Contextual: Includes relevant ticket numbers, documentation, or other references when necessary.

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
```

```
# Calculate mean age for femalesmean_age_females = df.loc[df['sex'] == 'f', 'age'].mean()
# Calculate mean age for malesmean_age_males = df.loc[df['sex'] == 'm', 'age'].mean()
```

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		