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 TRUE
   * Can compare changes over time TRUE
   * Can see who modified something last TRUE
   * Can recover lost files TRUE
2. What is the difference between centralized version control systems and distributed version control systems?

Centralized version control systems (CVCS) have a single server that contains all the versioned files, allow some level of transparency to others’ work and give Administrators a level of control over what developers can and can’t, and if the server goes down, all collaboration halt

If server dies, any of collaborators’ repositories can be copied back to the server to restore it in Distributed version control.

1. 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
2. What command initializes a new repository?
   * a) git clone
   * b) git branch
   * c) git fork
   * d) git init
3. 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
4. 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
5. How do you add a remote repo? (select all that apply)
   * a) git remote
   * b) git add remote Should be git remote add
   * c) git clone
   * d) git add clone
6. What is the difference between git pull and git fetch?

Git fetch will get any new changes but it wont merge it to our work or modify our work.

Gill pull will automatically fetch and merge a remote branch to our current branch

1. How do you switch branches?
   * a) git checkout
   * b) git checkout -b
   * c) git branch -c
   * d) git branch
2. Why are messages important? What would make a good commit message?

They act as a reminder for what our commit includes, and also tell our collaborators what we did last. Good commit messages would be: Short 950 chars or less) summary of changes, more detailed explanatory text if necessary about 72 characters or so, bullet points if needed but typically a hyphen or asterisk are used for the bullt

1. 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

df.loc[df['sex'] == 'f', 'age'].mean()

df.loc[df['sex'] == 'm', 'age'].mean()

# Part 2

1. fork and clone [this class GitHub repo](https://github.com/delipouya/DSI-workshop-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 |  |  |