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

Centralized Version Control Systems (CVCS) like SVN use a single, central server to store the repository, requiring a network connection for commits and making the server a single point of failure. Users only check out the latest version of the files. In contrast, Distributed Version Control Systems (DVCS) like Git allow each user to have a complete local copy of the repository, enabling offline commits and reducing reliance on a central server. This structure enhances data redundancy and flexibility in collaboration, as changes can be merged from different local repositories. DVCS offers more resilience against data loss and supports diverse workflows.

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

1. What command initializes a new repository?

a) git clone

b) git branch

c) git fork

* + d) git init

1. 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

1. 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

1. 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

1. What is the difference between git pull and git fetch?

**git fetch** downloads changes from a remote repository to your local repo without modifying your working files, allowing you to review changes before integration. **git pull,** on the other hand, not only fetches changes from the remote but also automatically merges them into your current branch. Essentially, fetch is for updating your local repository, while pull is for both updating and integrating

1. How do you switch branches?
   * a) git checkout
   * b) git checkout -b

c) git branch -c

d) git branch

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

Commit messages are crucial as they provide context and a clear explanation for changes in a project, aiding in tracking history and facilitating collaboration. A good commit message should include a concise summary of the changes and, if necessary, a detailed explanation of what was changed and why.

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

Assuming both are neede, we can keep both and assign it to variables.

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

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

# Part 2

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

A screenshot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

1. 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 screen shot of a computer screen

Description automatically generated

* + 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

A screenshot of a computer screen

Description automatically generated

1. 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 | yes |  |
| 2. README is comprehensive and includes components discussed in class plus at least one component learned from outside sources | yes |  |
| 3. Pull request has been successfully requested without any merge errors | yes |  |