GitHub Workflow

Applied Statistical Analysis/Quantitative Methods I

Jeffrey Ziegler, PhD

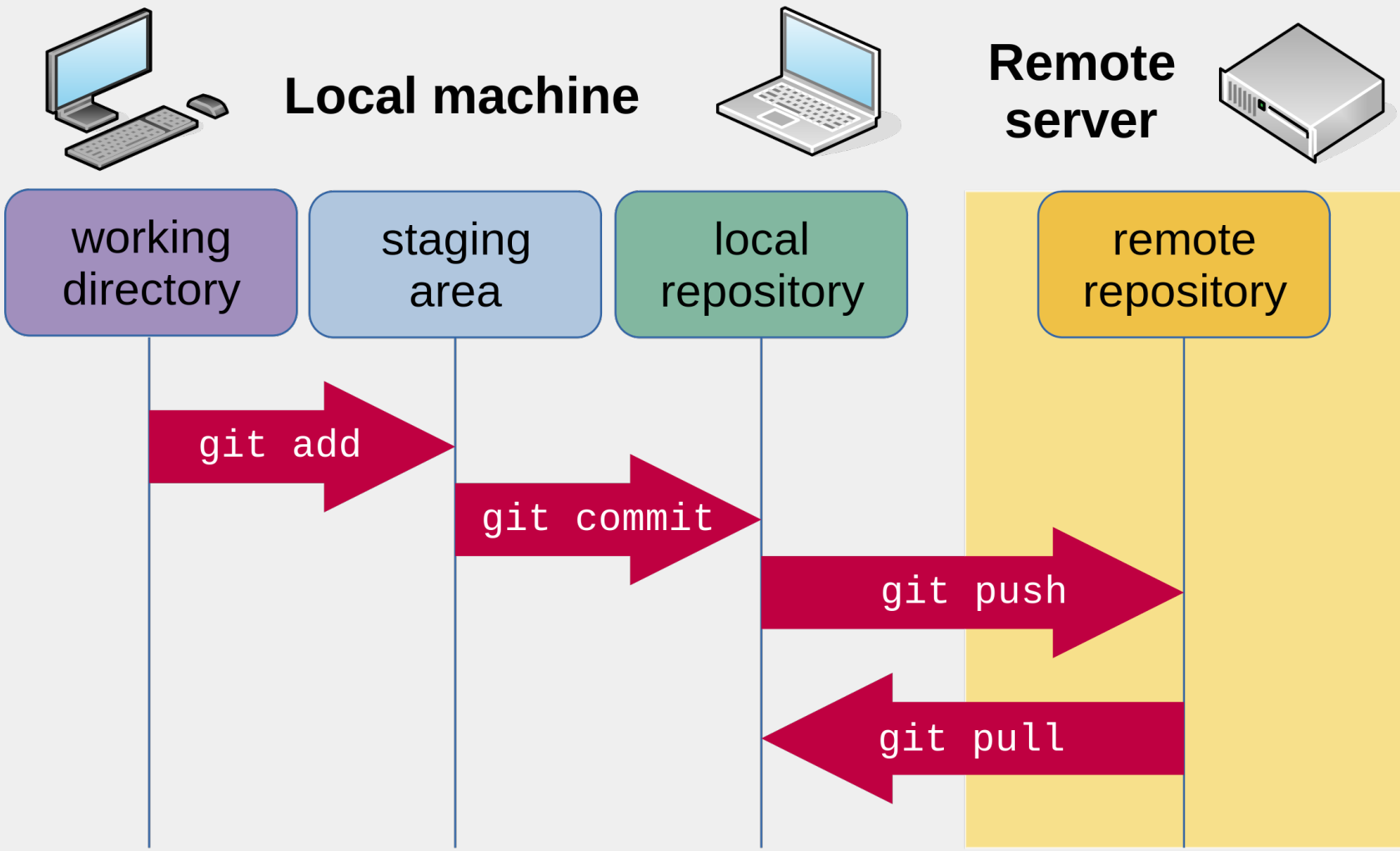
Assistant Professor in Political Science & Data Science
Trinity College Dublin

Fall 2023

Version Control and Git

- Version control systems (VCSs) allow automatic tracking of changes in files and collaboration
- Git is one of several major version control systems (VCSs, see also Mercurial, Subversion)
- GitHub is an online hosting platform for projects that use Git for version control

Git/GitHub Workflow



Some Useful Git Commands

Command (Windows)	Description
`git init <project name>`	Create a new local repository
`git clone <project url>`	Download a project from remote repository
`git status`	Check project status
`git diff <file>`	Show changes between working directory* and *staging area
`git add <file>`	Add a file to the staging area
`git commit -m “<commit message>”`	Create a new *commit* from changes added to the staging area
`git pull <remote> <branch>`	Fetch changes from *remote* and merge into *merge
`git push <remote> <branch>`	Push local branch to *remote* repository

Extra: Git Cheatsheet

Creating local Git repository

- Let's create a test project and track changes in it
- Create a test directory by typing 'mkdir test' in your CLI/Terminal
- Go into the newly created directory with 'cd test' command
- To make Git track changes run 'git init' command in this directory
- Congratulations! You now have a local repository for your test project

Making a commit: Creation to Staging

- Open your text editor of choice (Notepad, Sublime Text, Atom, Visual Studio Code, Vim, Emacs, ...)
- Create a file called ‘test.txt‘ in your local test repository
- Type whatever you like in this file
- Add this file to your staging area (make Git aware of its existence) by running ‘git add test.txt‘ command

Making a commit: Staging to Committing

- Commit this file to your local repository by running ‘git commit -m "Added first file"‘
- Note that all files that were added at the previous stage with ‘git add <file>‘ would be commited
- Check status of your repository by running ‘git status‘ (it should say ’nothing to commit, working tree clean’)
- Check history of your repository by running ‘git log‘ and make sure that you see your commit

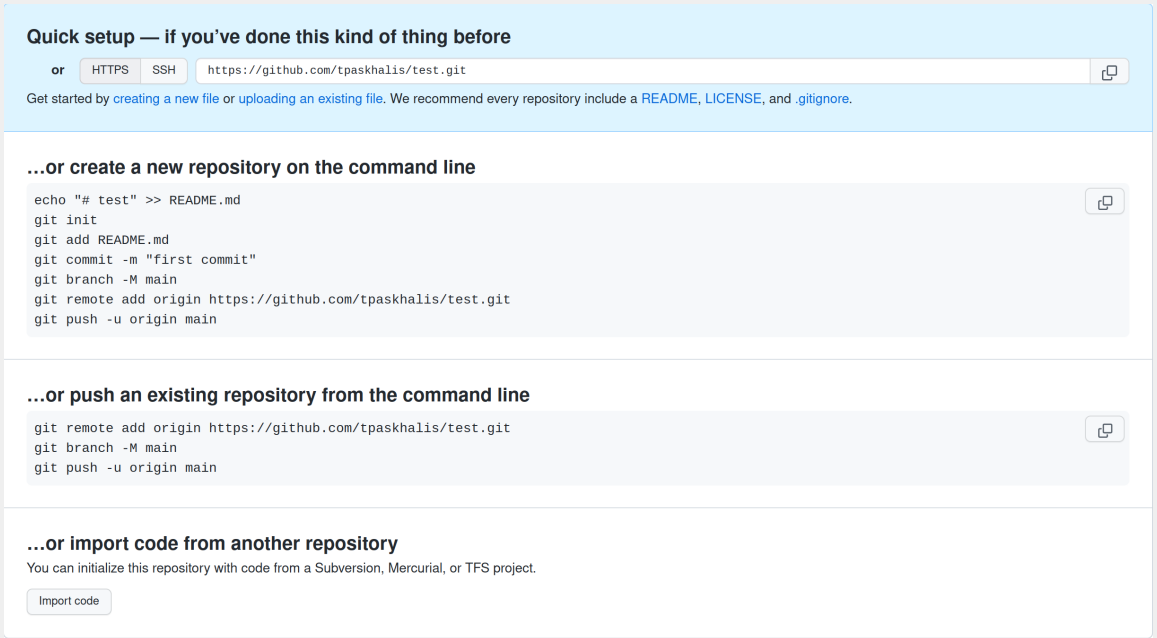
Remote Git repository: GitHub

- Hosting platform for projects that rely on Git for version control
- Bought by Microsoft in 2018
- Provides extensive tools for collaborative development and search functionality
- Helpful for troubleshooting more narrow problems (check GitHub Issues of the package/library that you have a problem with)
- GitHub is far from the only platform for hosting Git projects
- Popular alternatives to GitHub include GitLab, SourceForge

Creating remote repository on GitHub

- Register and login into your account on GitHub
- Create a new GitHub repository (choose private repository)
- You should see a similar page with the project URL of the form:

‘https://github.com/<username>/<repository_name>.git‘



Synchronising local Git repository with GitHub

- Go to your local Git repository (the one created in the previous step)
- Add link from your local Git repository to remote repository on GitHub by running: ‘git remote add origin <project_url>‘

  where:
  - ‘git remote add‘ is the command,
  - ‘origin‘ is the name given to this link (‘<remote>‘), and
  - ‘<project_url>‘ is the URL of the repository on GitHub
- Check the status of links between your local Git repository and remotes by running ‘git remote -v‘

  where:
  - ‘git remote‘ is the command, and
  - ‘-v‘ is the argument ’verbose’

Pushing local Git changes to GitHub

- Your local Git repo is now linked to remote repo hosted on GitHub
- Let's bring the changes made locally to the remote repository
- We will use the 'git push' command for that
- One last thing to check before doing so is which branch we're on
- Run 'git branch' to see name of branch you're on ('master' or 'main')
- Finally, run 'git push <remote> <branch>' (e.g. 'git push origin master')

  where:
  - 'git push' is the command,
  - '<remote>' is the name of the remote link, and
  - '<branch>' is the name of the branch
- Visit your GitHub repository to check that your commit is reflected there

Cloning module repository

- All module materials are hosted on GitHub in this repo
- You can clone this repository to your local machine by running: ‘git clone https://github.com/ASDS-TCD/StatsI_Fall2023‘
- This will create a folder called ‘https://github.com/ASDS-TCD/StatsI_Fall2023‘ within the directory where you ran this command
- To keep up to date with changes in the remote repository you can run: ‘git pull origin main‘

  where
  - ‘origin‘ is the remote address of the repository - ‘https://github.com/ASDS-TCD/StatsI_Fall2023‘
  - ‘main‘ is the name of the branch (recall the discussion about ‘main‘/‘master‘ change)