

# GITHUB WORKFLOW

APPLIED STATISTICAL ANALYSIS/QUANTITATIVE METHODS I

JEFFREY ZIEGLER, PHD

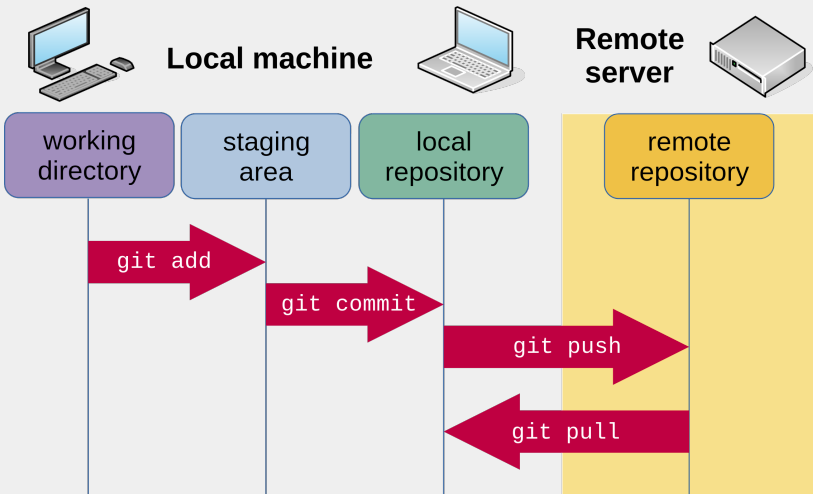
ASSISTANT PROFESSOR IN POLITICAL SCIENCE & DATA SCIENCE  
TRINITY COLLEGE DUBLIN

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
<code>git init &lt;project name&gt;</code>	Create a new local repository
<code>git clone &lt;project url&gt;</code>	Download a project from remote repository
<code>git status</code>	Check project status
<code>git diff &lt;file&gt;</code>	Show changes between working directory* and *staging area
<code>git add &lt;file&gt;</code>	Add a file to the staging area
<code>git commit -m "&lt;commit message&gt;"</code>	Create a new <i>commit</i> from changes added to the staging area
<code>git pull &lt;remote&gt; &lt;branch&gt;</code>	Fetch changes from <i>remote</i> and merge into *merge
<code>git push &lt;remote&gt; &lt;branch&gt;</code>	Push local branch to <i>remote</i> 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 committed
- 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'

The screenshot shows the GitHub 'Quick setup' page. At the top, it says 'Quick setup — if you've done this kind of thing before'. Below this, there are two tabs: 'HTTPS' (selected) and 'SSH'. The URL field contains 'https://github.com/tpaskhalis/test.git'. Below the URL field, it says 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.' Below this, there are three sections: 1. '...or create a new repository on the command line' with a copy icon and a code block containing: 

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/tpaskhalis/test.git
git push -u origin main
```

 2. '...or push an existing repository from the command line' with a copy icon and a code block containing: 

```
git remote add origin https://github.com/tpaskhalis/test.git
git branch -M main
git push -u origin main
```

 3. '...or import code from another repository' with a note 'You can initialize this repository with code from a Subversion, Mercurial, or TFS project.' and an 'Import code' button.

Quick setup — if you've done this kind of thing before

or **HTTPS** **SSH**

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/tpaskhalis/test.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/tpaskhalis/test.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

# 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)