

Week 1 Tutorial: Introduction to Jupyter, CLI, Git

POP77001 Computer Programming for Social Scientists

Integrated Development Environments (IDEs)

- There is a number of integrated development environments (*IDEs*) available for:
 - R (RStudio) and
 - Python (Spyder, PyCharm)
- As well as text editors with R/Python-specific extensions (Visual Studio Code, Sublime Text, Vim)
- Try different ones and choose what works best for you!

Jupyter Notebook

- Jupyter Notebook is a language-agnostic web-based interactive computational environment.
- It is available with backends (*kernels*) for different programming languages (**J**ulia, **P**ython, **R** = **Jupyter**)
- Can be used both locally and remotely.
- Combine both code (with comments) and code output.
- Good for ad-hoc data analysis and visualization.

Jupyter Notebook vs Scripts

- Written code is often distributed as **scripts**:
 - `.R` files - R scripts
 - `.py` files - Python scripts
- Scripts are easier to work with in text editor.
- But do not include output.

Jupyter Notebook Cells

- Notebooks allow writing, executing and viewing the output of Python code within the same environment
- All notebook files have `.ipynb` extension for **interactive python notebook**
- The main unit of notebook is *cell*, a text input field (Python, Markdown, HTML)
- Output of a cell can include text, table or figure

Jupyter Notebook Installation

- To work with Jupyter Notebooks we recommend installing JupyterLab Desktop, a cross-platform desktop application for Jupyter Notebooks.
- Alternatively, you can try using Kaggle Code, an online platform for hosting Jupyter Notebooks.
- Its interface is slightly different and you need to register on Kaggle or have a Google account, but it does not require any local installations.
- However, for this module and course more broadly we recommend installing toolchain for working with Jupyter Notebooks locally.

Starting Jupyter

- To start Jupyter, open CLI/Terminal and type `jupyter notebook`
- This will open a browser window with Jupyter Notebook displaying the directory, in which you executed the command above.
- To create a new notebook press `New` and select `Python` from the drop-down menu

Using Jupyter

- In order to run a Python command, create a new cell:
 - Press **+** in the toolbar or click **Insert, Insert Cell Below**
 - Make sure that in the drop-down menu on the toolbar you select **Code**
 - Press CTRL+ENTER to run a command
- Rather than running a Python command, you can also write Markdown in the cell (e.g. to create slides)
 - Select **Markdown** in the drop-down menu on the toolbar
 - Write Markdown (check **Markdown Cheatsheet**)
 - Press CTRL+ENTER to render Markdown cell

Jupyter Notebook Demonstration

The screenshot shows the JupyterLab web interface. At the top, there's a browser window with the address bar showing a URL with a token. Below the browser window, the JupyterLab logo is on the left, and 'Quit' and 'Logout' buttons are on the right. The main area has three tabs: 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser. The file browser has a header with 'Select items to perform actions on them.' and buttons for 'Upload', 'New', and a refresh icon. Below the header, there's a list of files and folders. A dropdown menu is open next to the 'New' button, showing options: 'Notebook:', 'Julia 1.0.4', 'Python 3' (which is highlighted), 'R', and 'Other:'. Below the 'Other:' section, there are options for 'Text File', 'Folder', and 'Terminal'. A dark overlay with the text 'Create a new notebook with Python 3' is positioned over the 'Python 3' option.

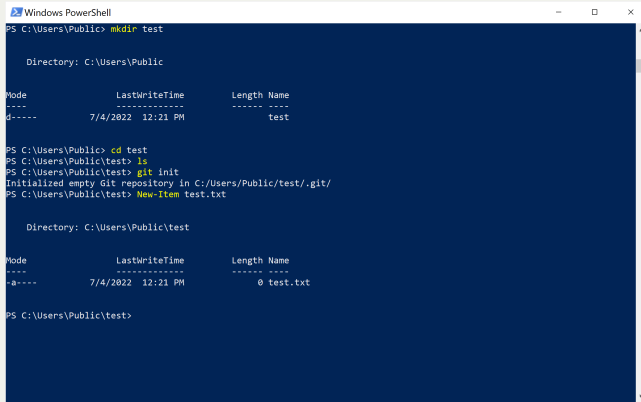
Jupyter Notebook Demonstration

The screenshot displays a web browser window with a Jupyter Notebook interface. The browser's address bar shows the URL `localhost:8888/notebooks/lectures/Untitled.ipynb`. The Jupyter interface includes a top bar with the notebook title "Untitled - Jupyter Notebo...", a "Logout" button, and a status message "Last Checkpoint: an hour ago (unsaved changes)". Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A toolbar contains icons for saving, creating new cells, deleting, copying, pasting, undo, redo, and running code. The "Run" button is highlighted. The main area shows a code cell with the text `1 + 1`. Below the code cell, the output is displayed as `In [1]: 1 + 1` and `Out[1]: 2`. The interface also shows a "Trusted" status and a "Kernel" button.

Stopping Jupyter Notebook

- First, make sure you saved your work (!) by pressing Command+S / CTRL+S
- You can close the running notebook by clicking **File** and then **Close and Halt**
- Jupyter Notebook runs as a server
- Which means that closing its tabs/web browser does not stop it
- You need to press **Quit** in the upper right corner of your main Jupyter tab (located at <http://localhost:8888/>)
- Alternatively, you can press CTRL+C in the terminal window

CLI Examples



```
Windows PowerShell
PS C:\Users\Public> mkdir test

Directory: C:\Users\Public

Mode                LastWriteTime         Length Name
----                -
d-----          7/4/2022 12:21 PM             test

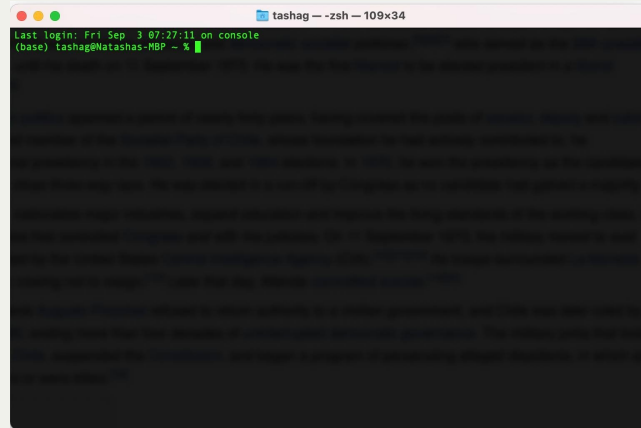
PS C:\Users\Public> cd test
PS C:\Users\Public\test> ls
PS C:\Users\Public\test> git init
Initialized empty Git repository in C:/Users/Public/test/.git/
PS C:\Users\Public\test> New-Item test.txt

Directory: C:\Users\Public\test

Mode                LastWriteTime         Length Name
----                -
-a-----          7/4/2022 12:21 PM             0 test.txt

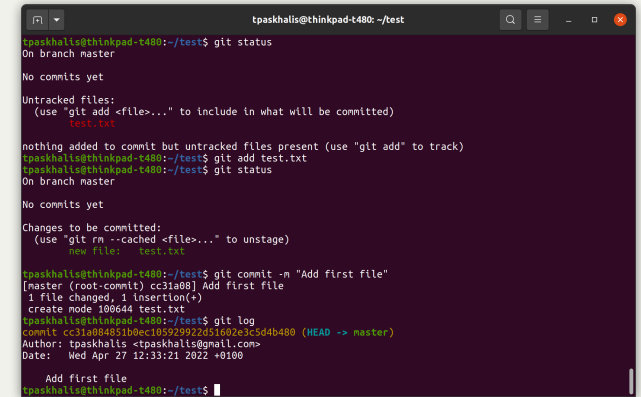
PS C:\Users\Public\test>
```

Microsoft PowerShell
(Windows)



```
tashag -- zsh -- 109x34
Last login: Fri Sep 3 07:27:11 on console
(base) tashag@Natashas-MBP ~ %
```

Z shell, zsh (macOS)



```
tpaskhalis@thinkpad-t480: ~/test
On branch master
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  test.txt

nothing added to commit but untracked files present (use "git add" to track)
tpaskhalis@thinkpad-t480:~/test$ git add test.txt
tpaskhalis@thinkpad-t480:~/test$ git status
On branch master
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   test.txt

tpaskhalis@thinkpad-t480:~/test$ git commit -m "Add first file"
[master (root-commit) cc31a08] Add first file
1 file changed, 1 insertion(+)
 create mode 100644 test.txt
tpaskhalis@thinkpad-t480:~/test$ git log
commit cc31a084851b0ec105929922d51692e3c5d4b480 (HEAD -> master)
Author: tpaskhalis <tpaskhalis@gmail.com>
Date:   Wed Apr 27 12:33:21 2022 +0100

    Add first file
tpaskhalis@thinkpad-t480:~/test$
```

bash (Linux/UNIX)

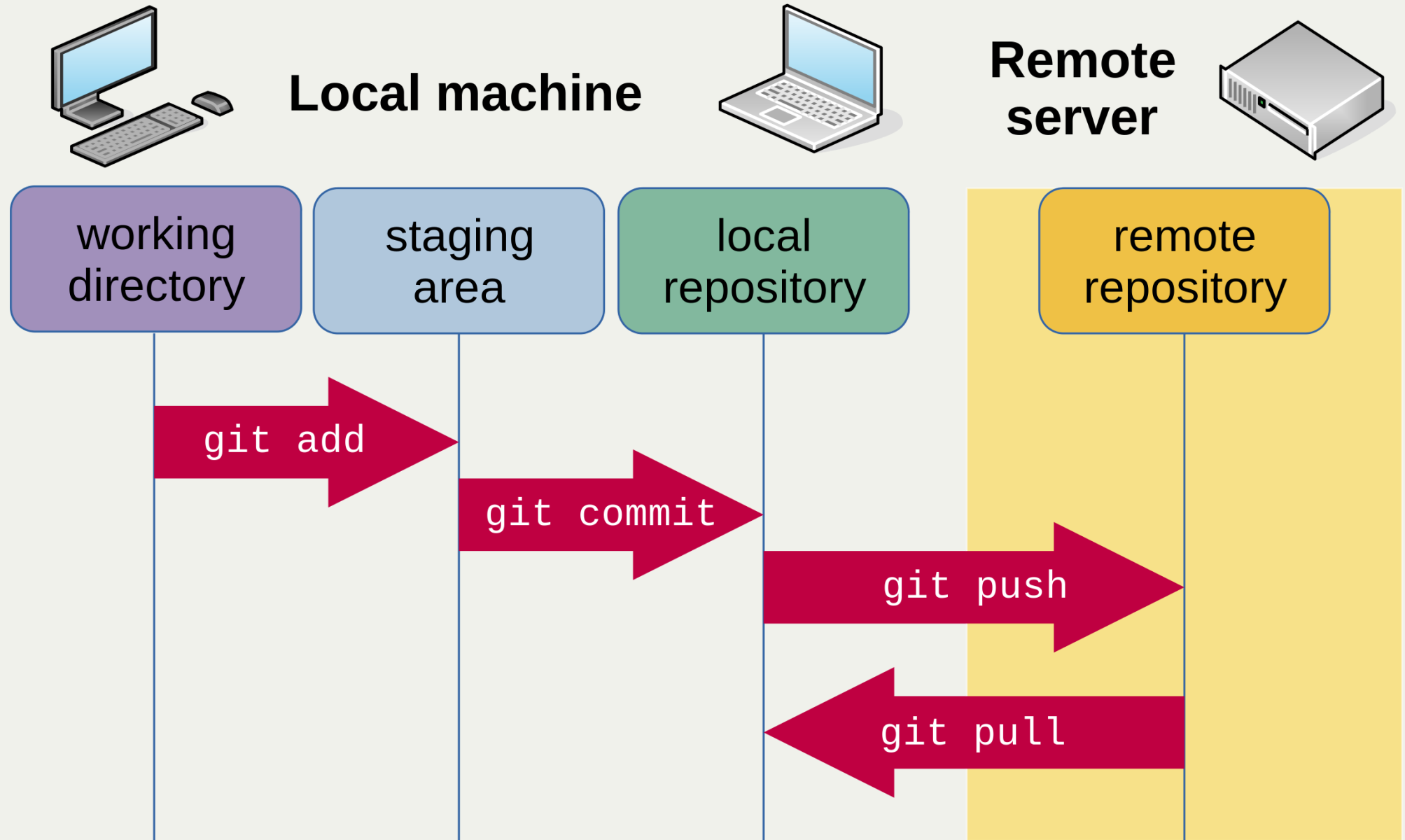
Some Useful CLI Commands

Command (Windows)	Command (macOS/Linux)	Description
<code>exit</code>	<code>exit</code>	close the window
<code>cd</code>	<code>cd</code>	change directory
<code>cd</code>	<code>pwd</code>	show current directory
<code>dir</code>	<code>ls</code>	list directories/files
<code>copy</code>	<code>cp</code>	copy file
<code>move</code>	<code>mv</code>	move/rename file
<code>mkdir</code>	<code>mkdir</code>	create a new directory
<code>del</code>	<code>rm</code>	delete a file



Introduction to CLI

Git/GitHub Workflow



Some Useful Git Commands

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



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

- Open your text editor of choice (Notepad, Sublime Text, 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
- 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 the status of your repository by running `git status` (it should say ‘nothing to commit, working tree clean’)
- Check the history of your repository by running `git log` and make sure that you see your commit

Remote Git repository: GitHub

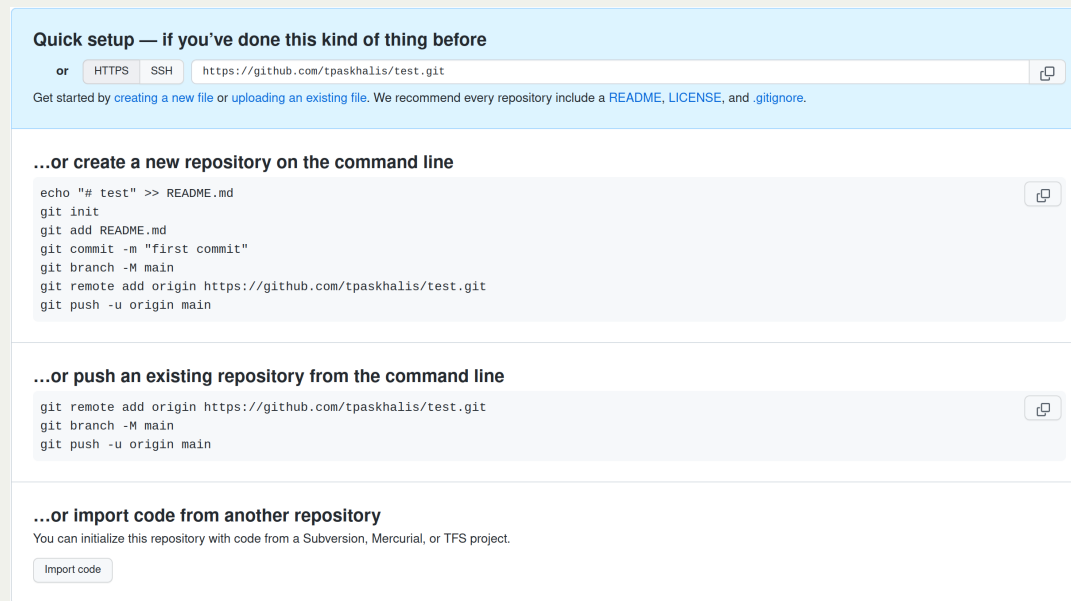


- Hosting platform for projects that rely on Git for version control
- Bought by Microsoft in 2018
- Provides extensive tools for collaborative development and search functionality
- Helpful for troubleshooting more narrow problems (check [GitHub Issues](#) of the package/library that you have a problem with)
- GitHub is far from the only platform for hosting Git projects
- Popular alternatives to GitHub include [GitLab \(UA\)](#), 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 'Quick setup' page on GitHub. At the top, it says 'Quick setup — if you've done this kind of thing before'. Below this, there are two tabs: 'HTTPS' and 'SSH'. The 'HTTPS' tab is selected, and the URL 'https://github.com/tpaskhalis/test.git' is displayed in a text box. To the right of the text box is a copy icon. Below the text box, 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 for creating a new repository on the command line. The first section is titled '...or create a new repository on the command line' and contains a list of commands: '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', and 'git push -u origin main'. To the right of the commands is a copy icon. The second section is titled '...or push an existing repository from the command line' and contains the same list of commands. To the right of the commands is a copy icon. The third section is titled '...or import code from another repository' and contains the text 'You can initialize this repository with code from a Subversion, Mercurial, or TFS project.' Below this text is a button labeled 'Import code'.

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

or **HTTPS** SSH `https://github.com/tpaskhalis/test.git` 

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 repository is now linked to the remote repository 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 are currently on.
- Run `git branch` to see the name of the branch you are on (it would be 'main' or 'master')
- Finally, run `git push <remote> <branch>` (e.g. `git push origin main`)
 - 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.

Week 1 Exercise (Unassessed)

- Create a Jupyter notebook in your local repository
- Commit it to your local repository in the same way as `test.txt` file