

Git

Joseph Longworth

Session Outline



- What is Version Control Software and it's benefit
- Fundamentals of Git interaction (Work Locally)
- Git Hosting Services and Collaboration

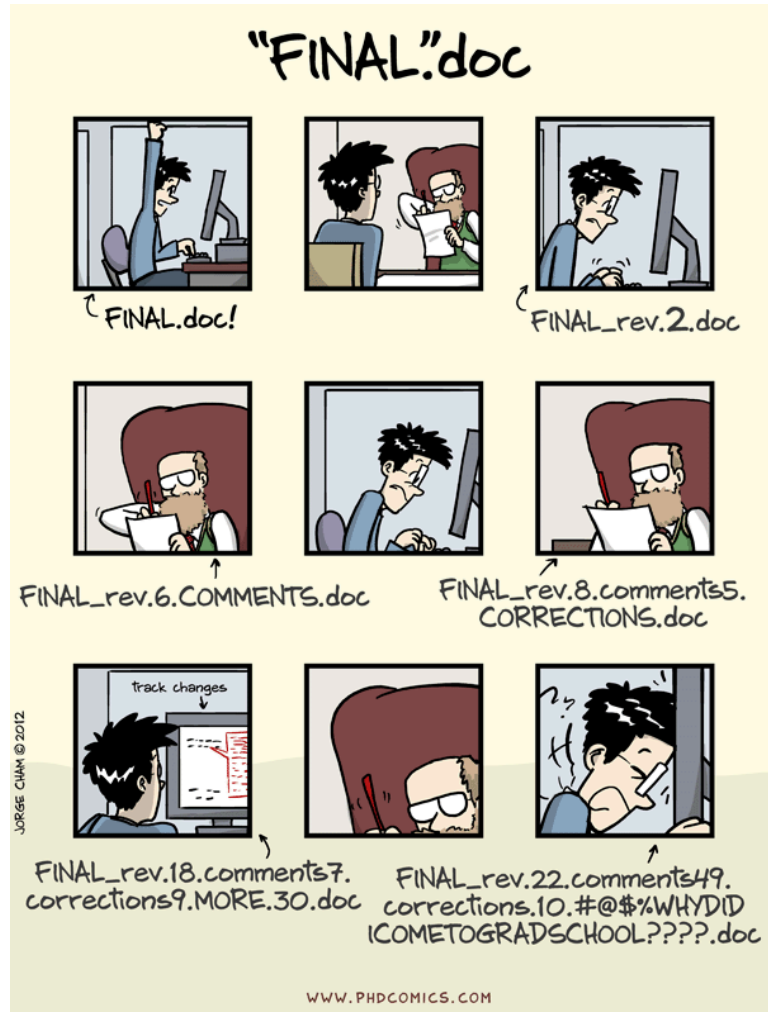
Who is this?



- Anyone remember who this was?
- Linus Torvalds
- Having created Linux in 1991 in 2005 he created Git.

Version Control

The Chaos of Uncontrolled Code



- Renaming our files with version “indicators” quickly gets confused.
- This is complex when working alone when working collaboratively it’s a 🤬
- Lost changes, confusion, and frustration become commonplace.

Introducing Version Control

Think about:

- How have you tracked your files in the past
- How would describe you edits over time
- How would you explain it to someone else

*5 min video on version control
general concept*

<https://git-scm.com/video/what-is-version-control>

Git Fundamentals

How to Start with Git

Think about:

4 min video Git

- We should have git installed already 🙌
- What `git configs` are needed to get started
- Look for git `git init`
- What is `git add` and `git commit`

<https://git-scm.com/video/get-going>

Git Commands Recap

```
git init
```

Creates a new Repository

```
git add
```

Moves files to staging area

```
git commit -m ""
```

Commits staged files to git history

```
git status
```

Current State of our stage

```
git log --oneline
```

History view (condensed)

Class Activity 1

Class Activity 1.1

1. Open you linux terminal that we had form the last session
2. Change directory `cd` to the home directory
3. Run git configs
 1. `git config --global user.name "MeMario"`
 2. `git config --global user.email "@gmail.com"`
4. run `git init first_git`
5. change directory into the new git tracked directory `cd ???`
6. Create a file with `touch file_1.txt`
7. Check `git status`

Class Activity 1.2

1. `git add file_1.txt` to add the file to the 'stage'
2. Check `git status`
3. `git commit -m "Describe your commit"`
4. Repeat creating 2 more commits (use nano to edit your file)
5. run `git log --oneline` what can you see?

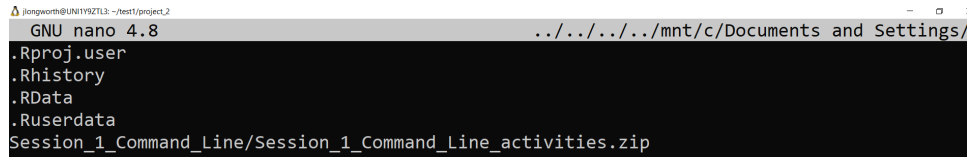


	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Extra bits

.gitignore



```
GNU nano 4.8 ..../mnt/c/Documents and Settings/
.Rproj.user
.Rhistory
.RData
.Ruserdata
Session_1_Command_Line/Session_1_Command_Line_activities.zip
```

git add -A

- Using the -A all files altered in the repository apart from those in .gitignore will be added.

Coffee Break



Git Hosting Services & Collaboration

Git Hosting Services

LIKELIHOOD YOU WILL GET CODE WORKING
BASED ON HOW YOU'RE SUPPOSED TO INSTALL IT:



Git hosting services provide a platform for storing, managing, and collaborating on code using Git.

GitHub and GitLab are powerful tools for managing your code, but they cater to slightly different styles:



- **GitHub:**

- ~ 70 million users
- **Focus:** Open source projects and individual developers.
- **Strengths:** Simple interface, huge open-source community.

- **GitLab:**

- ~30 million users
- **Focus:** Collaboration and private projects (even for teams!).
- **Strengths:** Built-in project management tools, self-hosting option.

Web vs Local Server

We can work with these services though likely we will first work with our own institutional server.

Internet

- <https://github.com/>
- <https://gitlab.com/>
- LIH
 - <https://git.lih.lu/>
 - <https://gitlab01.lih.lu/>
- UNI
 - <https://gitlab.lcsb.uni.lu/>
 - gitlab-cloud.uni.lu

Clone vs. Fork: Duplicating a Git Repository

- **Clone:** Creates a complete copy of a remote repository on your local machine
 - Ideal for: Downloading and working on a project without affecting the original.
- **Fork:** Creates a copy of a remote repository on your Git hosting service (e.g., GitHub).
 - You can make changes and propose them back to the original project through pull requests.

Class Activity 2

Class Activity 2

1. Go to Github and log in. See your current repositories
2. go to
<https://github.com/JosephLongworth/AntigenMicroarray>
3. Fork the project to your own account
4. Clone your forked project `git clone "url link found on page"` (use https unless you have the ssh set up)
5. change to text in the Readme file using `nano`
6. `git push` the changes and check the website

Class Activity 2.2

The screenshot shows the GitHub interface for a repository named 'Test' (Public), which is a fork of 'JosephLongworth/AntigenMicroarray'. The repository is on the 'master' branch, which is up to date with the upstream. The file list includes 'Sub scripts', 'data', 'www', 'Example_data_one_slide.zip', 'LICENSE', 'README.md', 'global.R', and 'server.r'. The 'Code' button is highlighted with a red circle, and the 'Clone' dropdown menu is open, showing the 'HTTPS' option also highlighted with a red circle. The URL 'https://github.com/NextImmune2-DataScienceMeeti' is visible in the dropdown, with a copy icon also highlighted.

NextImmune2-DataScienceMeetings / Test

Code Pull requests Actions Projects Wiki Security Insights Settings

Test Public
forked from JosephLongworth/AntigenMicroarray

master 1 Branch 0 Tags

Go to file Add file Code

This branch is up to date with JosephLongworth/AntigenMicroarray:master

JosephLongworth Correct Test

File	Commit Message	Time
Sub scripts	Update Scan Data I	
data	Add files via upload	
www	Add files via upload	
Example_data_one_slide.zip	Add files via upload	
LICENSE	Initial commit	
README.md	Update README.md	3 years ago
global.R	Add files via upload	3 years ago
server.r	Update server.r	3 years ago

Local Codespaces

Clone ?

HTTPS SSH GitHub CLI

https://github.com/NextImmune2-DataScienceMeeti

Clone using the web URL

Open with GitHub Desktop

Download ZIP

About

Toolset for Production and an Analysis of Antigen Microarrays

- Readme
- GPL-3.0 license
- Activity
- Custom properties
- 0 stars
- 0 watching
- 0 forks

Report repository

Releases

No releases published

Create a new release

Packages

Collaborative Working

look for:

- what is a branch
- `git pull`
- `git push`
- `git merge`

*on version control general
concept*

2:40-6min

<https://git-scm.com/video/what-is-git>

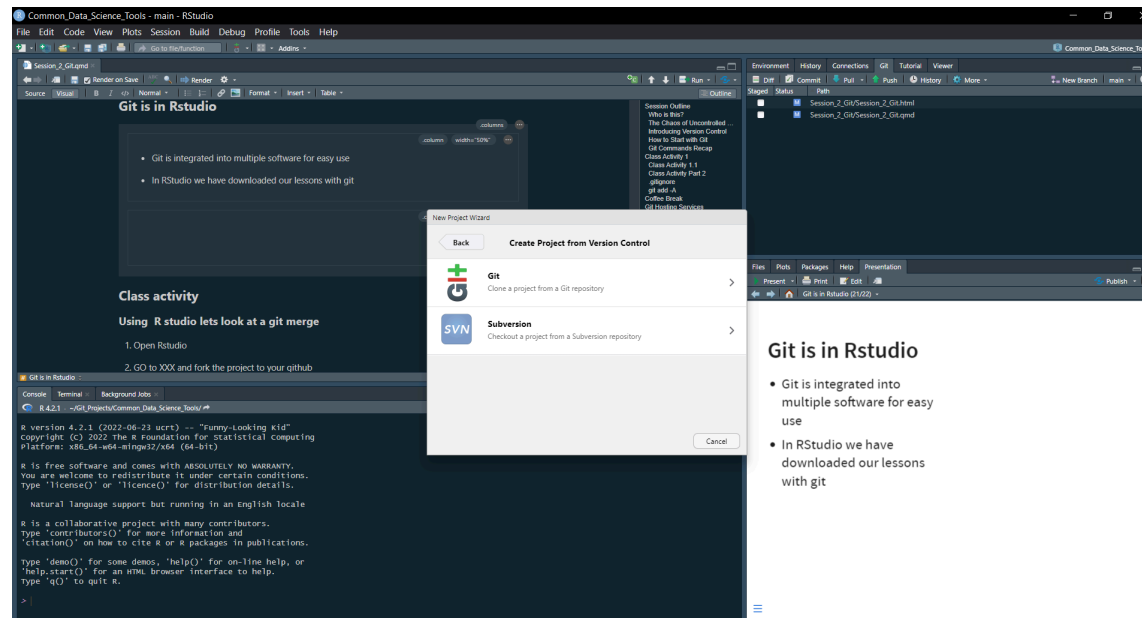
Collaboration Commands Recap

`git push` Update the remote to your current state

`git pull` Update you local version with the remote

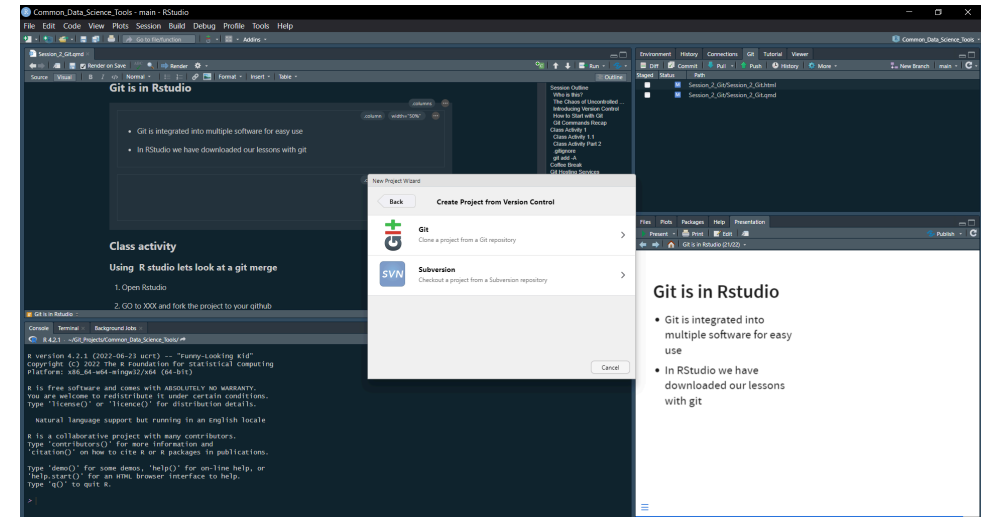
`git branch` Create a branch to work in a isolated section

`git merge` Merge branches together



Git is in Rstudio

- Git is integrated into multiple software for easy use
- In RStudio we have downloaded our lessons with git



Class Activity 3

Class Activity 3 Merging

1. Open Rstudio
2. GO to
https://github.com/JosephLongworth/Merge_Demonstartion
3. Using Projects in rstudio GUI clone the repository
4. Switch branches in rstudio
5. from the main branch run `git merge branch1` in the Terminal
6. look at results what could and couldn't be 'auto merged'
7. Fix conflicts and push back

Fin

