Tooling ourselves up

2. Introduction to git and github

Pierre Walthéry

August 2024

Plan

- Three presentations:

  (1. Basic of document creation with Quarto)

  2. Using Git and GitHub for version control

  3. Online version control with GitHub

Today:

1. What is git?
2. The git grammar
3. A worked example
4. Introduction to GitHub
5. Publishing online content using GitHub
6. More examples

1. What is git?

What does git even mean?

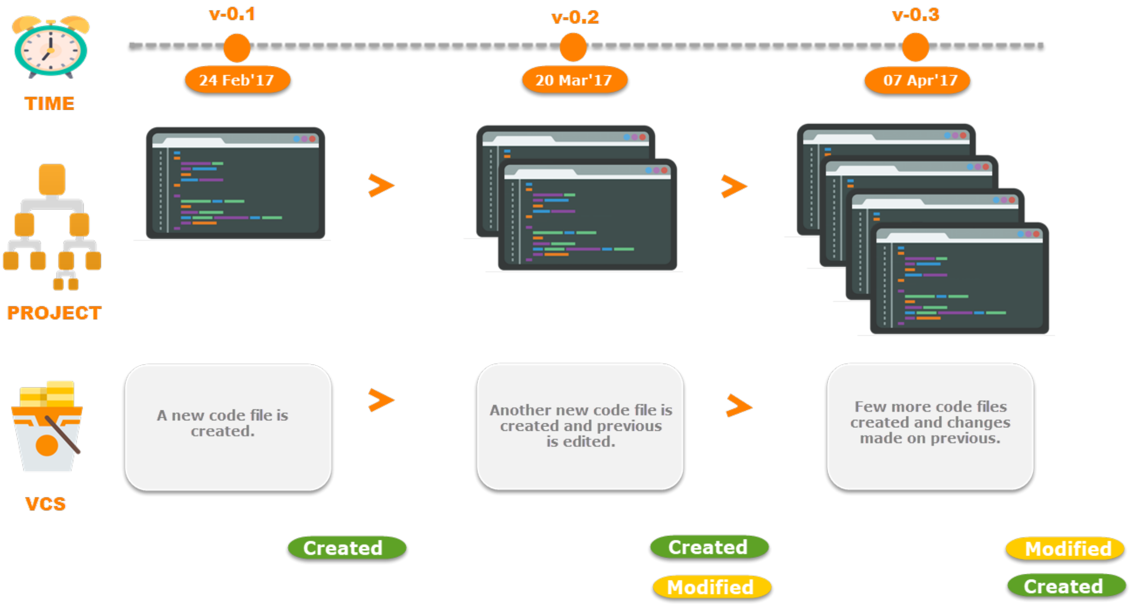
- From Wikipedia:
  - "Global Information Tracker": you're in a good mood, and it actually works for you. Angels sing, a light suddenly fills the room.
  - (random sequence of rude words): when it breaks.

- More seriously, git is a …
  - Distributed…
  - … Version Control
  - … System
- Some estimate that >90% of the version control operations globally are carried out with git

Defining features

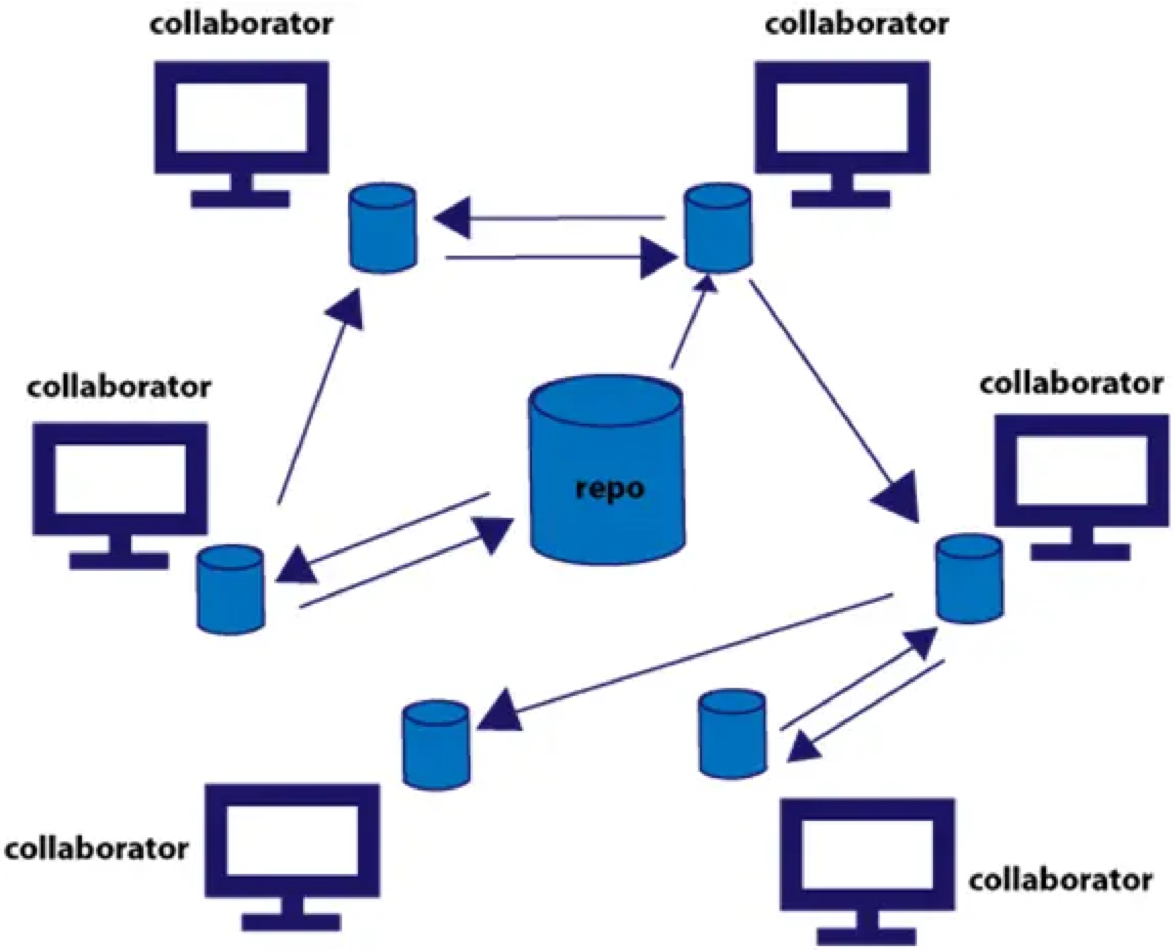
- **Version Control**:
  - < software engineering
  - … *practice of controlling (…) files and versions of files; primarily source code text files, but generally any type of file*.
  - Required when building complex software projects and when several - potentially many – developers collaborate together

What it looks like


TIME
v-0.1
24 Feb'17
v-0.2
20 Mar'17
v-0.3
07 Apr'17
PROJECT
VCS
A new code file is created.
Another new code file is created and previous is edited.
Few more code files created and changes made on previous.
Created
Created
Modified
Modified
Created

Distributed version control

- Everyone has access to all files at all times. Each user has an up to date, copy of the repository* and its version history* :
- Hierarchical client-server vs a ‘mushrooming’ approach
  - Users independently amend their copy of the repository on their local computer
  - In principle, no need for an ‘offical’ project server, but in practice often retained
- Users have to agree on changes before they are merged into a ‘master’ version.



In a nutshell…

- What git does
  - git records changes to some content (file/files) so that all team members have access to the current version on their local machine
  - git only records the changes (differences between versions) as opposed to entire versions of files
  - keeps a reversible history of a project through which users can navigate
- What it is for:
  - A single user works on a project, wants to keep track of corrected errors and/or explore further developments/ideas;
  - A team collaborates on (different sections of) a project, in parallel or sequentially;

Do we really need git at UKDS □?

- More, more diverse and more complex training materials:
  - need for unique versions of documents, easily transferable between people, computers, etc
- Increase in the number of 'software' materials we produce: ie syntax files; quarto documents; shiny apps
- Need for 'agile' publication of content at the same time as flexibility to amend them
- –> git/github as a possible way forward

Software implementation

- git command line: standalone program that runs from a console (terminal)
- Several GUI implementations:
  - GitHub Desktop, RStudio, GitKraken…
- Available on all major OS: Windows, MacOS, Linux
- *Open source*, free, created by Linus Torvald (< Linux kernel)

2. The git grammar

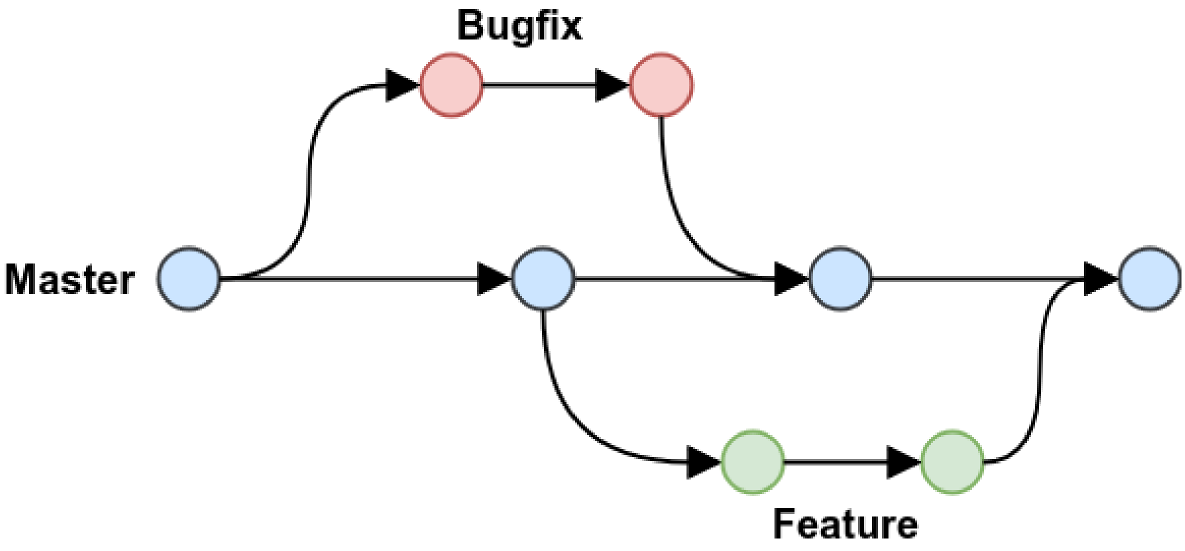
Essential concepts…

- *Working directory/tree*: contains the current file(s) of the project
- *Repository*: where tracked project files and their version history are stored
  - located in a *.git* folder inside the working directory
  - Not all files inside a working tree need to be tracked
- *Staging area*: buffer zone where (version of) files can be manipulated before being added to the repository
- *Branch*: a copy of the working directory to be worked on/modified without altering the master branch

Essential actions

- `git init:` create a new repository inside a working directory
- `git add:` adds files (new and modified ones) to the staging area
- `git commit:` records the version of the files in the staging area into the repository
  - A 'commit' is a snapshot of a working tree at a given point in time.
- `git branch:` creates/manipulate branches for working on a project without altering the current ie *master* branch
- `git switch:` moves between branches
- `git merge:` merges commits from a branch into the *master* branch
- `git status`, `git log`, `git diff`: inspect the status of repositories and files, monitor changes

The git workflow



3. A worked example

What we will cover

- Creating a single file repository
- Initial commit
- Amending content and committing changes
- Keeping track of what happened
- Amending a project via a branch

4. GitHub

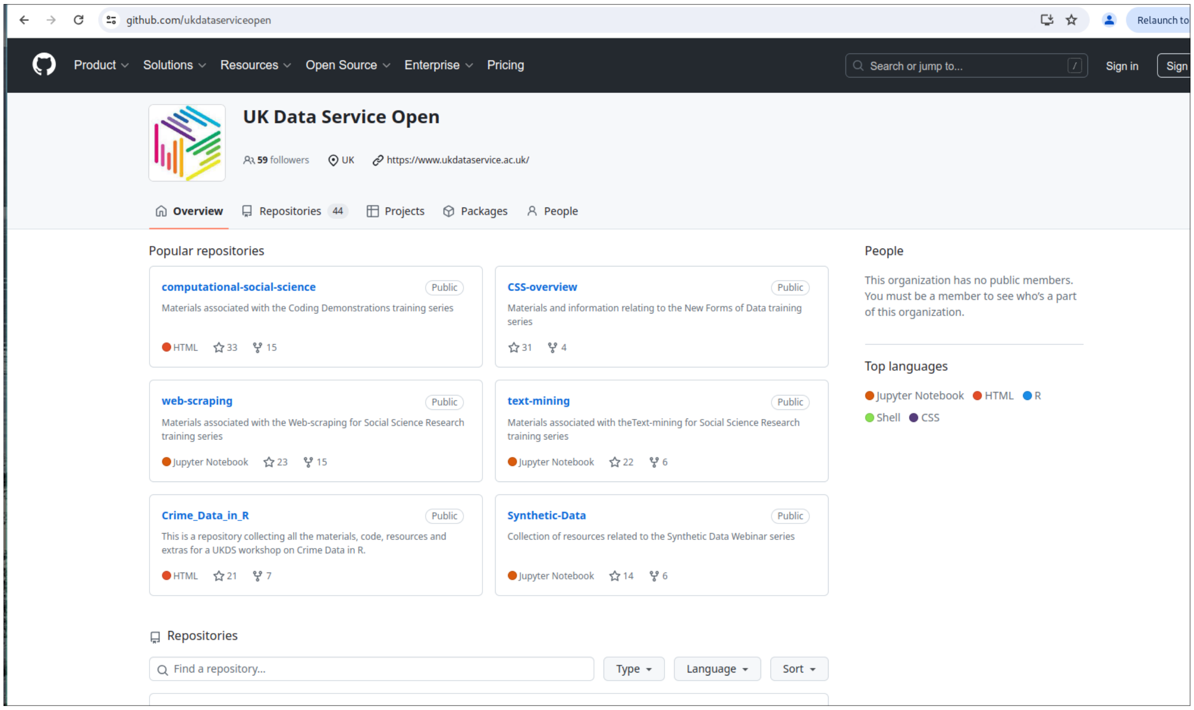
What is GitHub?

- One among several online extensions of git
  - Online VC & collaboration platform relying on git
- Other platforms include gitlab, Codeberg, gitea, etc
- Owned by Microsoft since 2018, content used for AI training.

- De facto standard, digital commons: 200,000M+ public repositories
- Convenient platform for:
  - sharing source UKDS content with a large audience
  - flexibly publish training material as web pages
  - Possibly also collecting user feedback and suggestions

Features

- All of git functionalities plus:
  - Online storage of git repositories
  - Multi-user online collaboration
  - Basic web server capabilities via GitHub Pages
  - Advanced security policies
  - Via UoM subscription: more features – private web server



UK Data Service Open

59 followers · UK · https://www.ukdataservice.ac.uk/

Popular repositories

computational-social-science Public
Materials associated with the Coding Demonstrations training series
● HTML ☆ 33 ⑂ 15

CSS-overview Public
Materials and information relating to the New Forms of Data training series
☆ 31 ⑂ 4

web-scraping Public
Materials associated with the Web-scraping for Social Science Research training series
● Jupyter Notebook ☆ 23 ⑂ 15

text-mining Public
Materials associated with theText-mining for Social Science Research training series
● Jupyter Notebook ☆ 22 ⑂ 6

Crime_Data_in_R Public
This is a repository collecting all the materials, code, resources and extras for a UKDS workshop on Crime Data in R.
● HTML ☆ 21 ⑂ 7

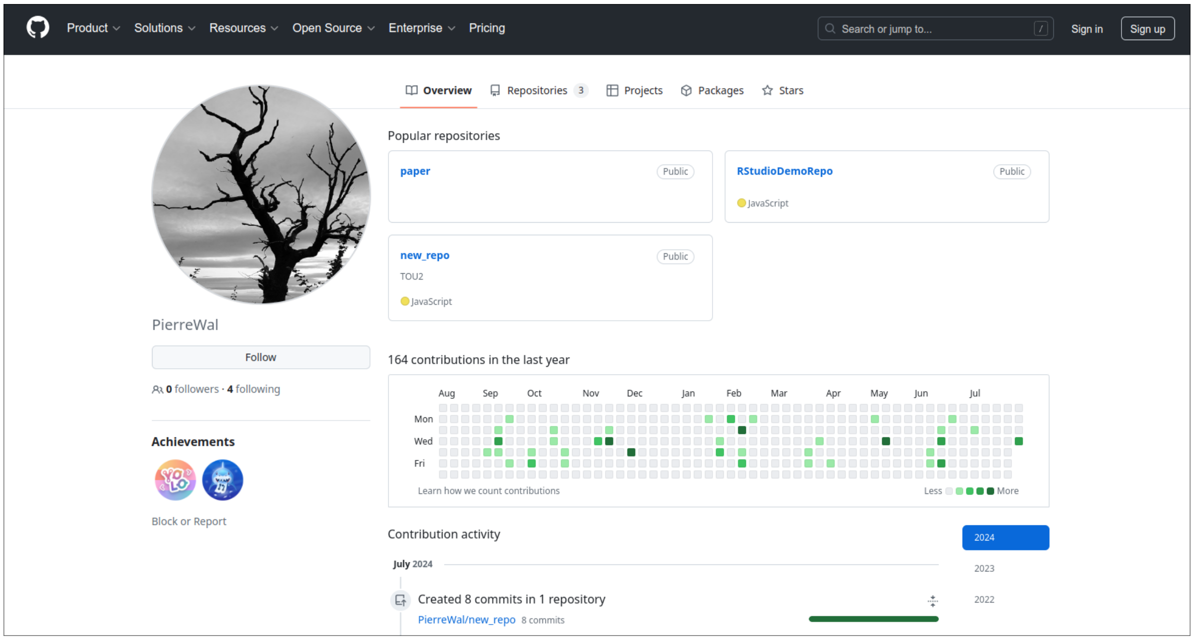
Synthetic-Data Public
Collection of resources related to the Synthetic Data Webinar series
● Jupyter Notebook ☆ 14 ⑂ 6

People

This organization has no public members. You must be a member to see who's a part of this organization.

Top languages

● Jupyter Notebook ● HTML ● R ● Shell ● CSS



PierreWal

0 followers · 4 following

Achievements

Block or Report

Popular repositories

paper Public

RStudioDemoRepo Public
JavaScript

new_repo Public
TOU2
JavaScript

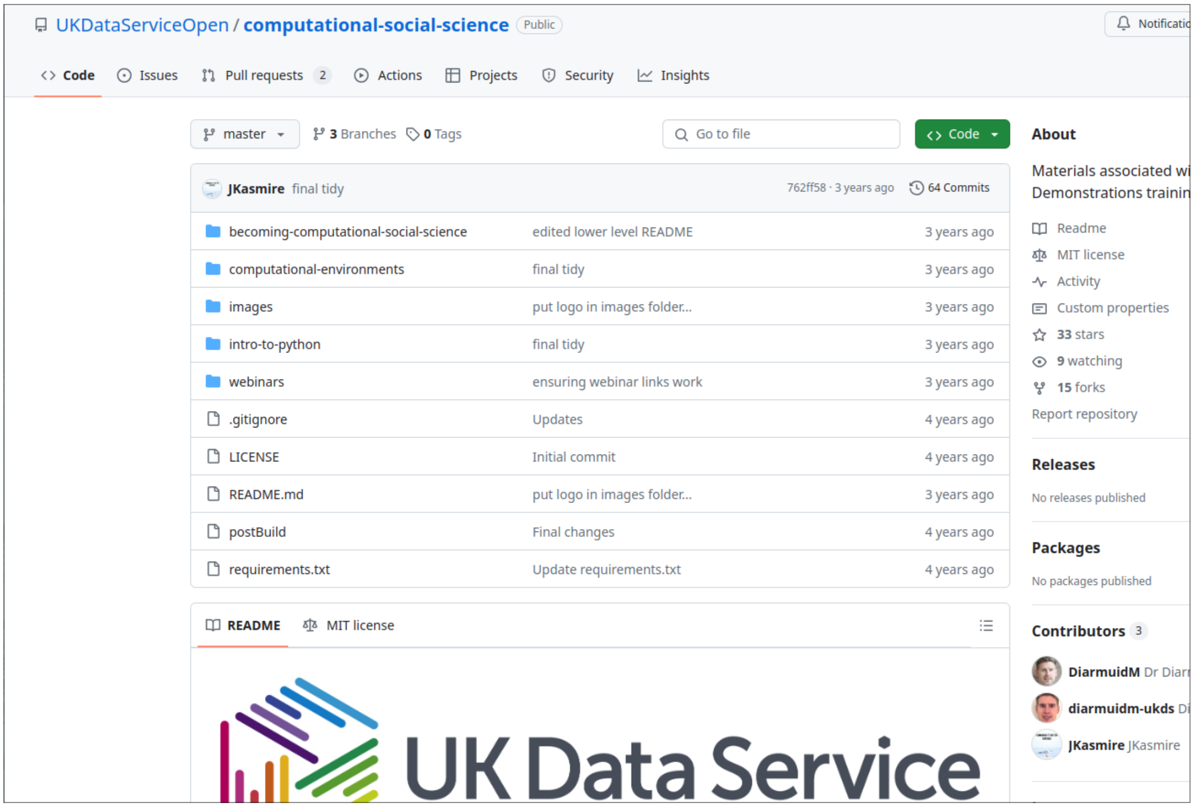
164 contributions in the last year

Contribution activity

July 2024

Created 8 commits in 1 repository

PierreWal/new_repo 8 commits



UKDataServiceOpen / **computational-social-science** Public

Notificatio

Code · Issues · Pull requests 2 · Actions · Projects · Security · Insights

master · 3 Branches · 0 Tags · Go to file · Code

JKasmire final tidy · 762ff58 · 3 years ago · 64 Commits

becoming-computational-social-science	edited lower level README	3 years ago
computational-environments	final tidy	3 years ago
images	put logo in images folder...	3 years ago
intro-to-python	final tidy	3 years ago
webinars	ensuring webinar links work	3 years ago
.gitignore	Updates	4 years ago
LICENSE	Initial commit	4 years ago
README.md	put logo in images folder...	3 years ago
postBuild	Final changes	4 years ago
requirements.txt	Update requirements.txt	4 years ago

README · MIT license

UK Data Service

About

Materials associated wi
Demonstrations trainin

Readme
MIT license
Activity
Custom properties
33 stars
9 watching
15 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 3

DiarmuidM Dr Diarı
diarmuidm-ukds Di
JKasmire JKasmire

Working with GitHub

- New conventions:
  - **Main:** local version of the repository branch (instead of *master*)
  - **Origin:** remote version of the branch on GitHub or other
- New commands:
  - To pair a new *empty* online repository with a local one:
    - `git remote add origin` * address of the online repository*
  - To upload local changes (including a new repo) to an online repo:
    - `git push origin main`
  - To retrieve locally the newest version of a project from an online repo:

        - `git pull` or `git fetch`
    - To download locally an online repo and start working on it:
        - `git clone` * address of the repository*
- These commands can be run from a terminal, from a GUI app, or RStudio

5. Workflow examples

1. Start a new project tracked on GitHub

- Create an online repository
- Clone it with RStudio
- Create content
- Add files to the staging area and commit changes
- Push the commit to the online repository

2. Upload existing work into a new online repository

- Create an empty online repository
- Initialise the local git repository;
- Add files and commit changes
- Declare the online repo
- Push the commit on GitHub

3. Publishing simple websites

- Setting up GitHub Pages
- Extra considerations:
    - Choosing the location of the content: *root* vs *docs* subdirectory
    - index.html vs specified page name
    - Using Readme.md as an landing page

Final note

- **Next time**
  - More on interfaces
  - Fetching/pulling commits
  - Interacting with other users
- **Additional resources**
  - gittutorial Basic introduction to the main functionalities of git
  - Git Tutorial on W3 School (In-depth introduction to Quarto)
  - For RStudio and GitHub, see this tutorial
  - Lots of YouTube videos

Thank you for your attention

- Any comments: pierre.walthery@manchester.ac.uk