# Git and Github
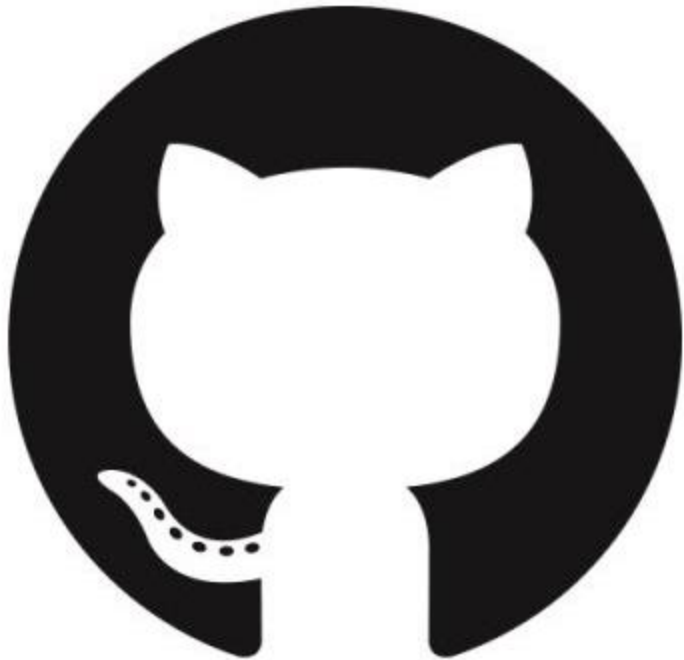Code/slides: Joel Swift
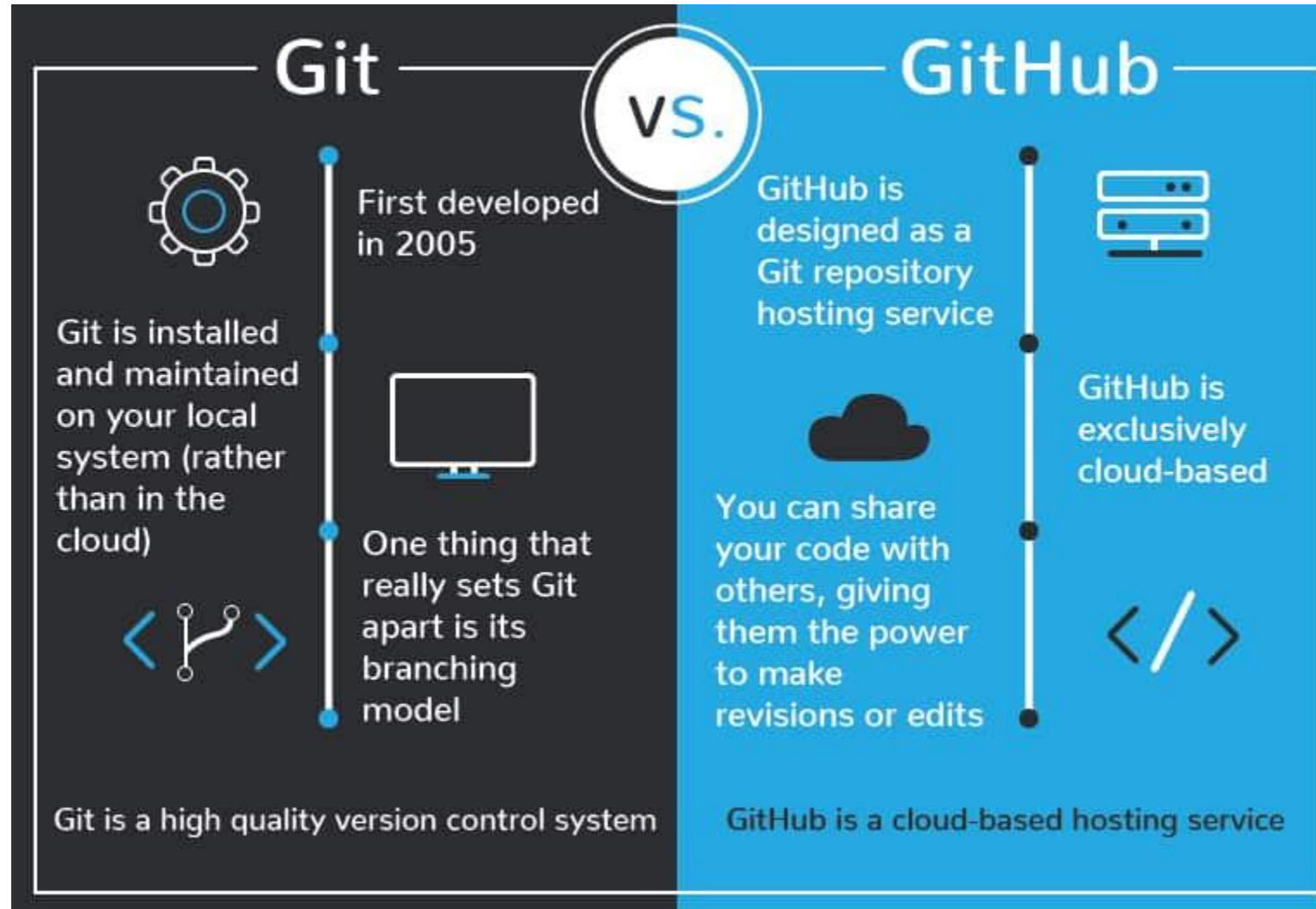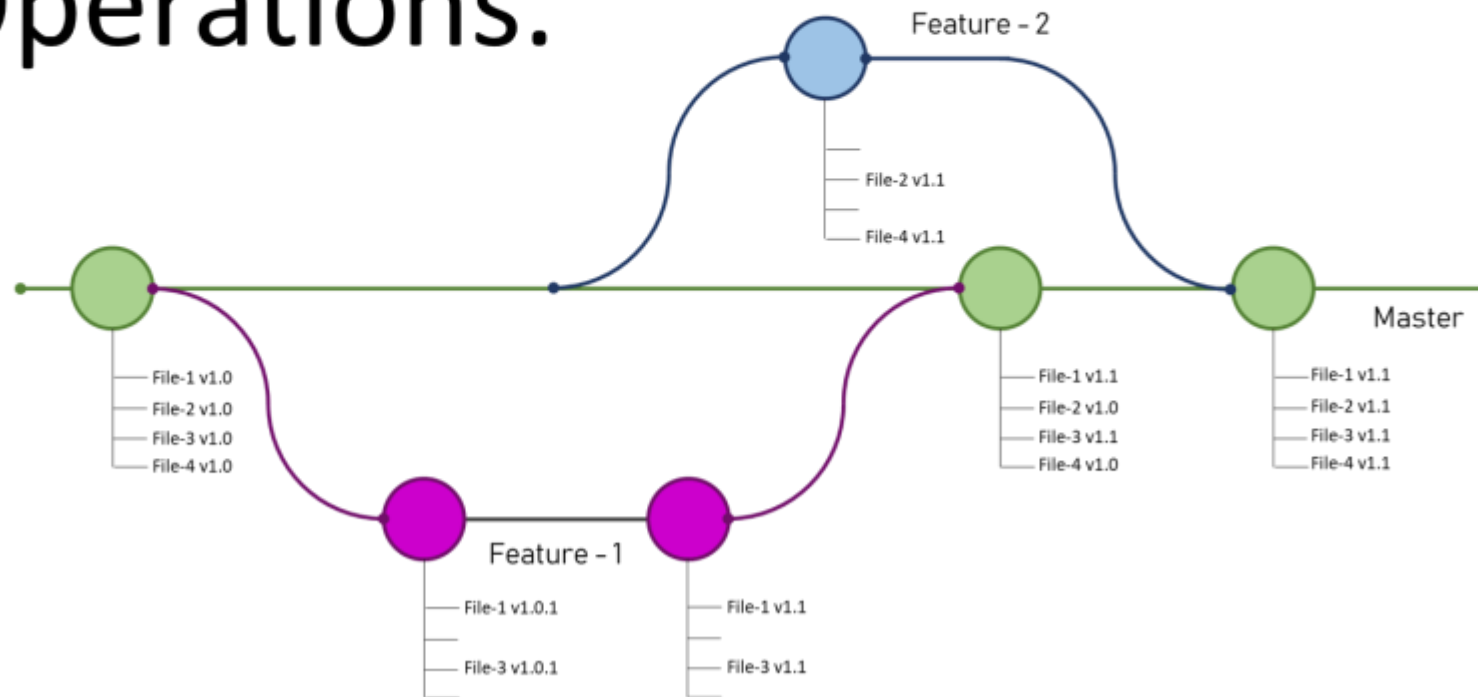
# What the hell are git and github?

- Git is a version control system.
  - Basically it allows for branched workflows, where multiple branches of a file or code can co-exist and be merged.

- Github
  - A cloud based hosting service for hosting git projects (repositories, repos).
  - GUI support.
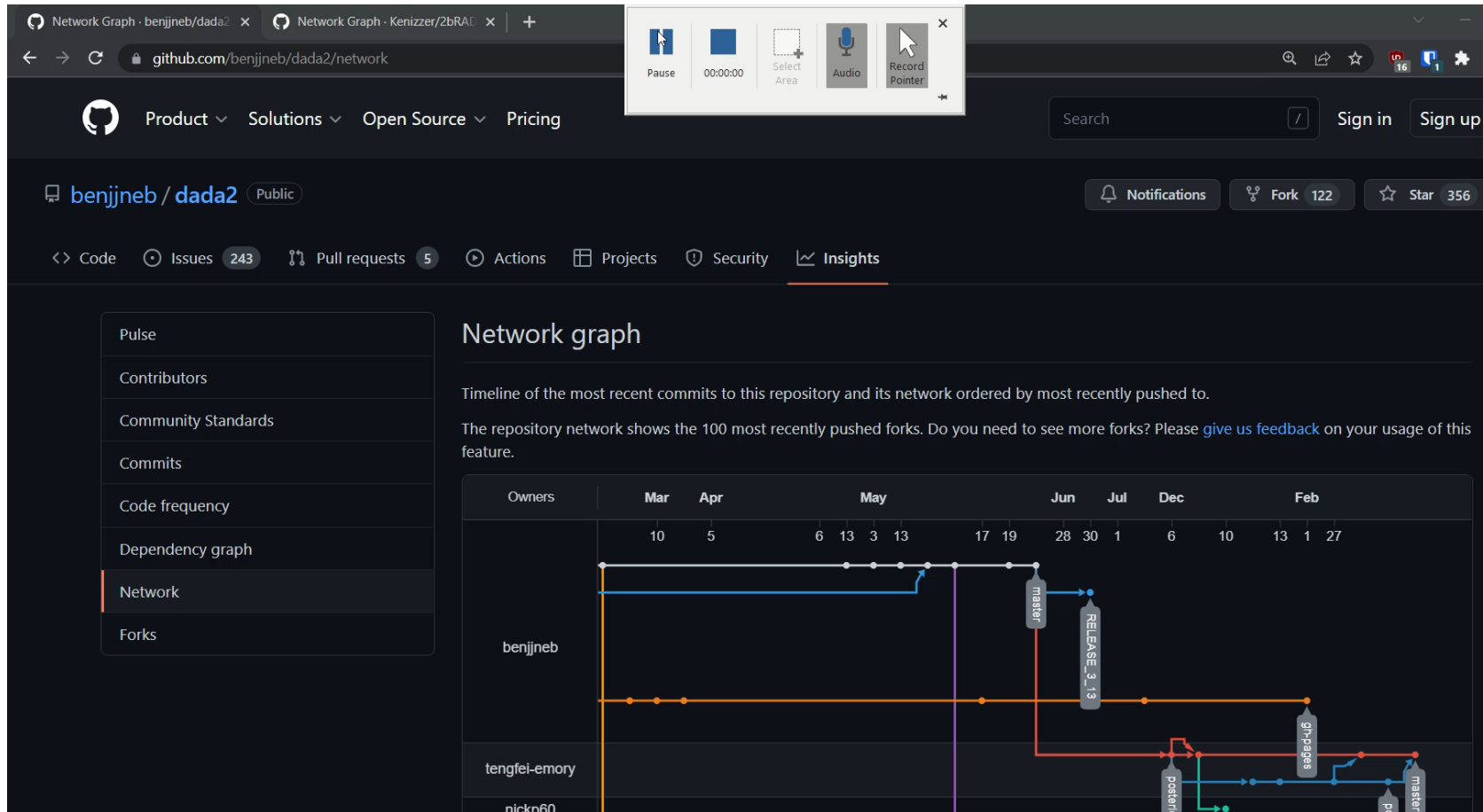  - Was purchased by Microsoft for 7.5 billion in !



## Git **VS.** GitHub

**Git**

Git is installed and maintained on your local system (rather than in the cloud)

First developed in 2005

One thing that really sets Git apart is its branching model

Git is a high quality version control system

**GitHub**

GitHub is designed as a Git repository hosting service

You can share your code with others, giving them the power to make revisions or edits

GitHub is exclusively cloud-based

GitHub is a cloud-based hosting service

https://devmountain.com

# *Branching model of Git*

# Real world examples (dada2 and my repo)

# Github… make an account

- Make an account

- Setup 2FA (w/ mobile or hardware if you have it)

# *Github Desktop*

- https://desktop.github.com/
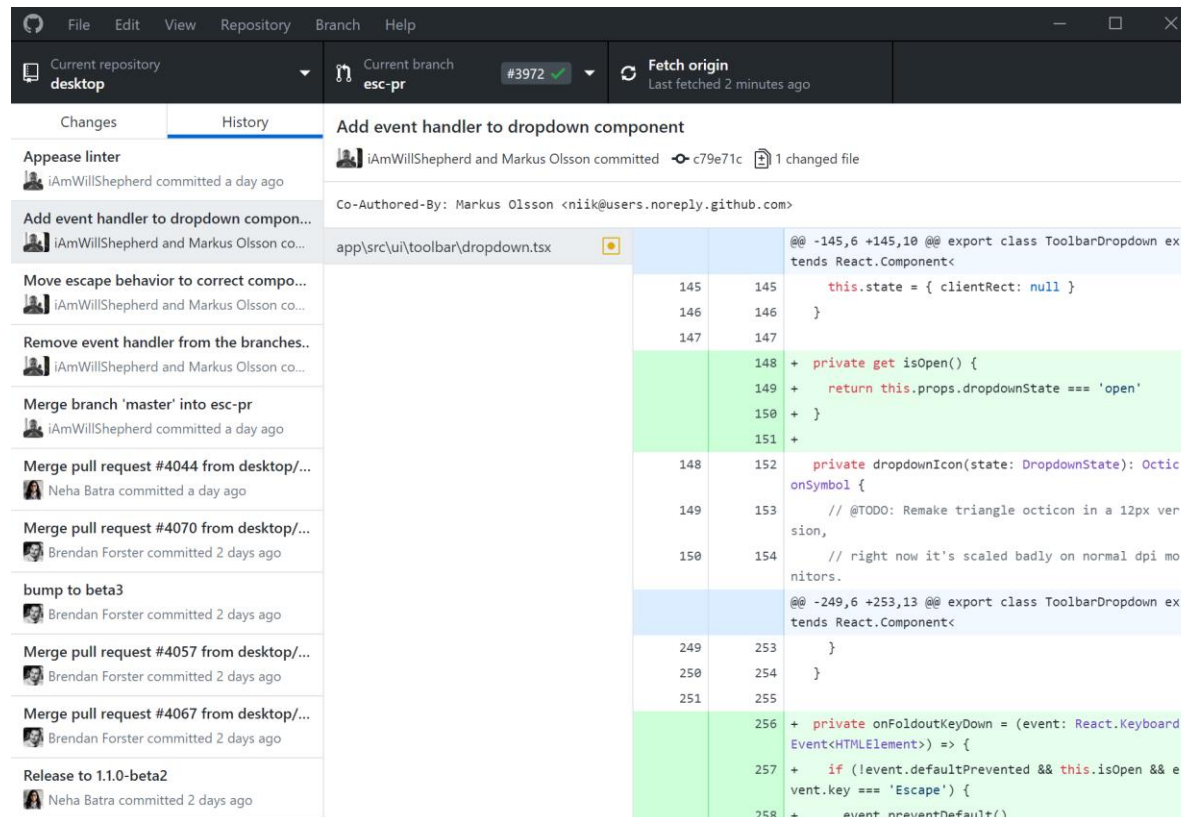
# *Why use github?*

- Collaborative coding is greatly simplified in a git/github environment.
- To share the code required to ensure your analysis is reproducible.
- Its often a journal requirement prior to publication.
- To better the environment for others that come after you, sometimes the solution to ones problems have been solved before…

```
### 30 year average (1990-2021) of percipitation for KS counties
county_precip_KS <- read.csv('KS_precipitation1895to2021.csv', sep
which(colnames(county_precip_KS)=="X1990") # index 99
which(colnames(county_precip_KS)=="X2021") # index 130
# Average the rainfall annually for 1990-2021
county_precip_KS$AVG_30_years <- rowMeans(county_precip_KS[,99:130
# Make a simpler data frame for plotting
county_fip_30yrAVG <- data.frame('county_fips' = county_precip_KS$
# Fips # to character to play nice with plotting function
county_fip_30yrAVG$county_fips <- as.character(county_fip_30yrAVG$
# Sanity check
county_fip_30yrAVG[1:5,]
summary(county_fip_30yrAVG)

# Color palette
# Taken from the colors in https://climate.k-state.edu/basics/
colfunc<-colorRampPalette(c("#fffe7a",
```

Fitzpatrick, Connor R., et al. "Assembly and ecological function of the root microbiome across angiosperm plant species." *PNAS* 115.6 (2018)

```
1591 ##### 4) Differential abundance testing #####
1592 # Thank you to Maggie Wagner for making her R code publically available! #
1593 # Many of the analytical operations were based on her code #
1594 # Also thank you to Michael Love for patient responses on the DESeq
1595 # Will use DeSeq2 to estimate fold change differences across factors and test
1596 # for differential abundance.
```

# *Repositories/Repos on github*
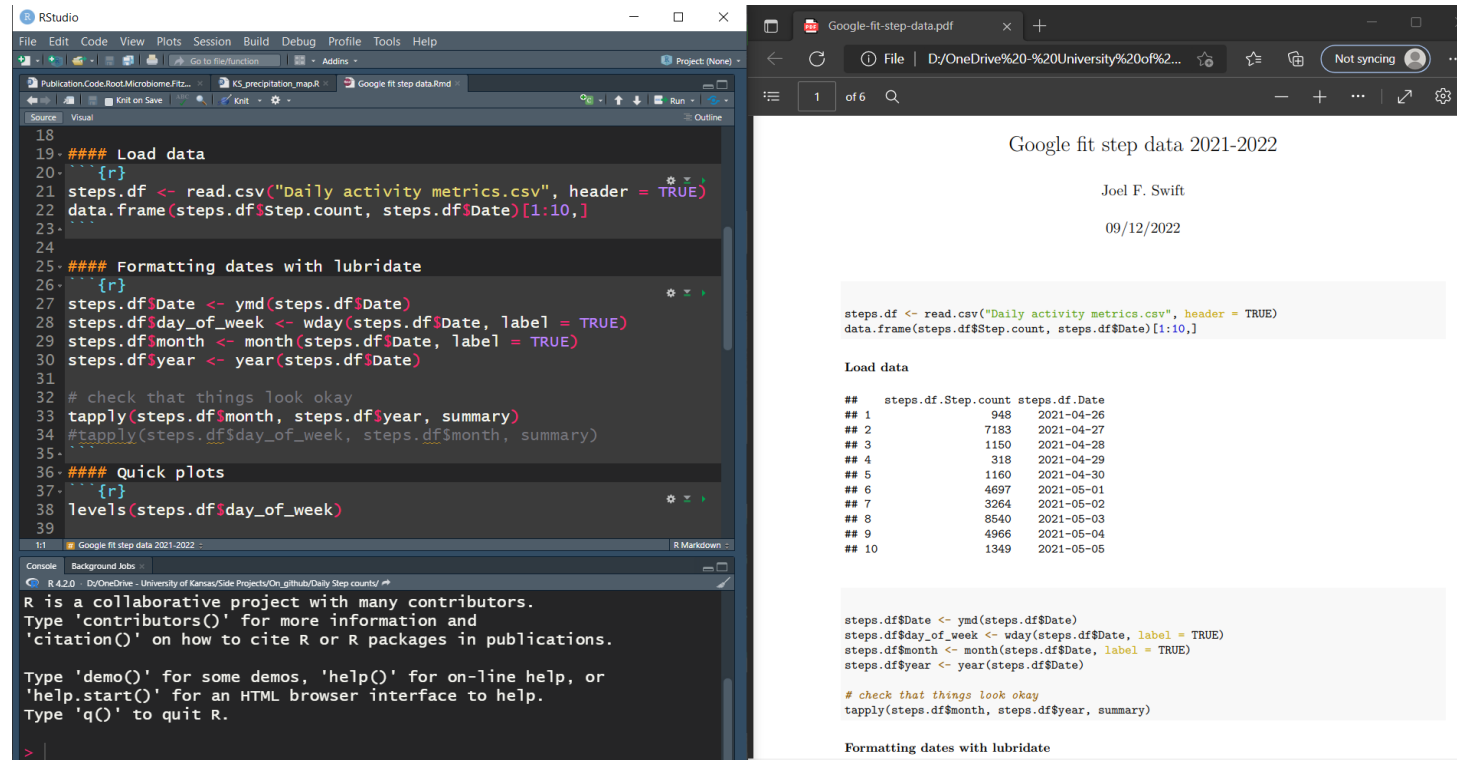
https://github.com/Kenizzer/California_Transect_Microbiome

# *Repositories/Repos on github*

- The basics are:
  - A readme file that will be kind of like the home page of the repo.
  - Something you are wanting to share (i.e., code/analyses)

- Readme files:
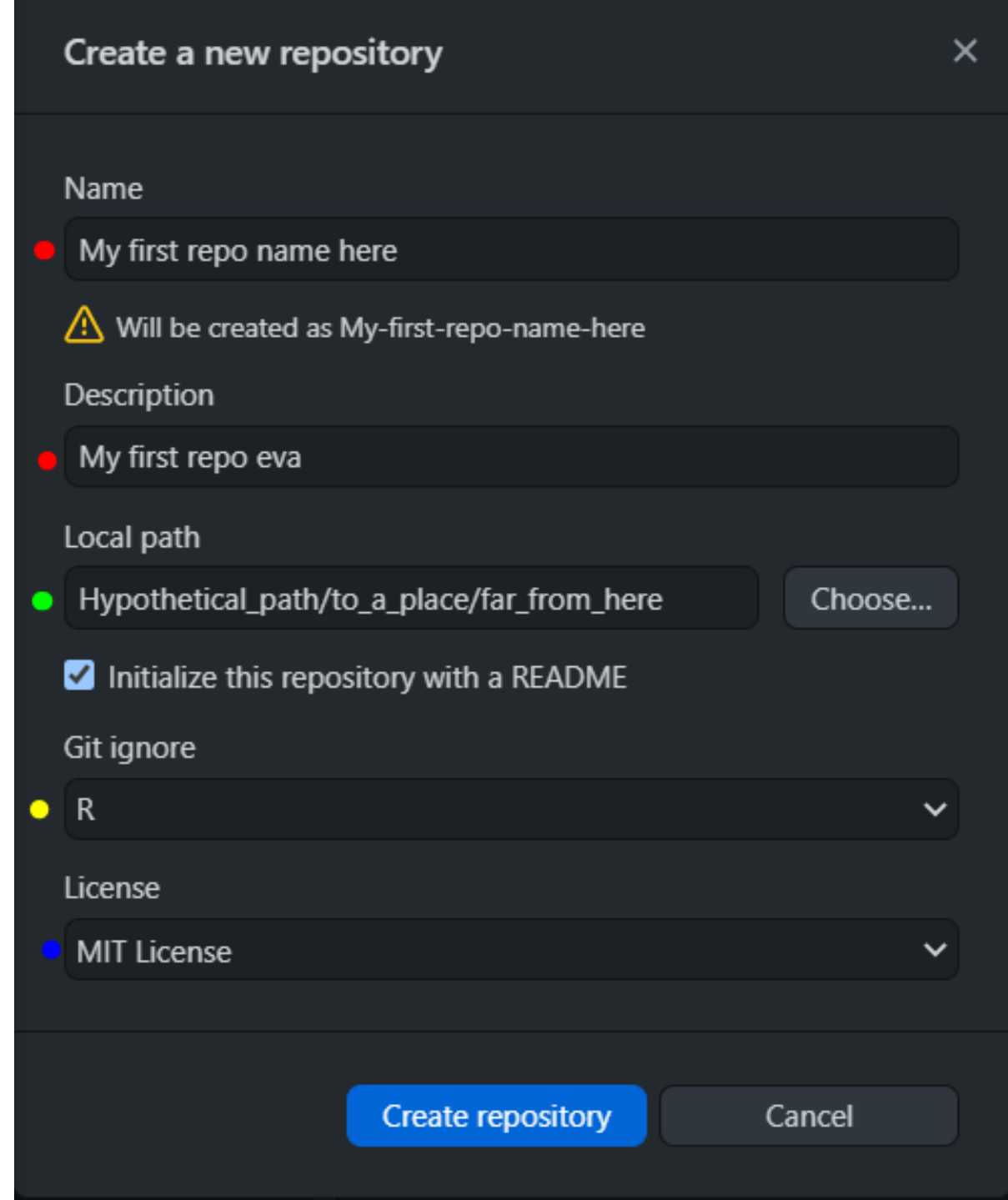  - These are typically written in markdown (not covered today)

# *Example repo – Google fit step count data*

- I wanted to look for trends in the number of steps I took per day.

- Data was collected by the phone application google fit.

- ~04-2021 to 07-2022.
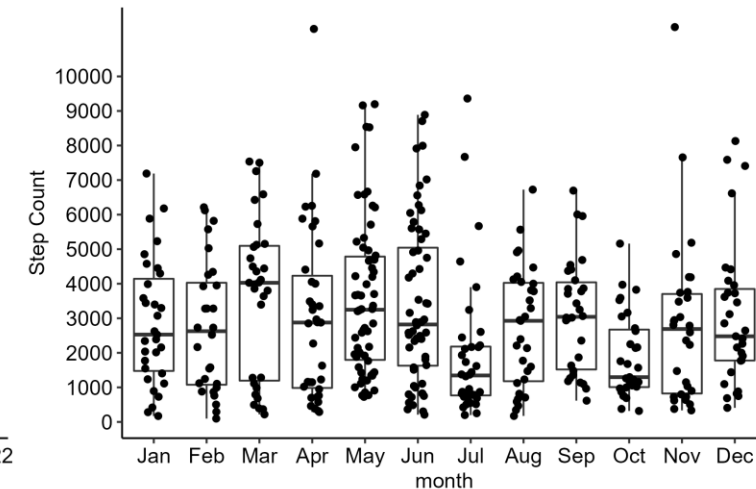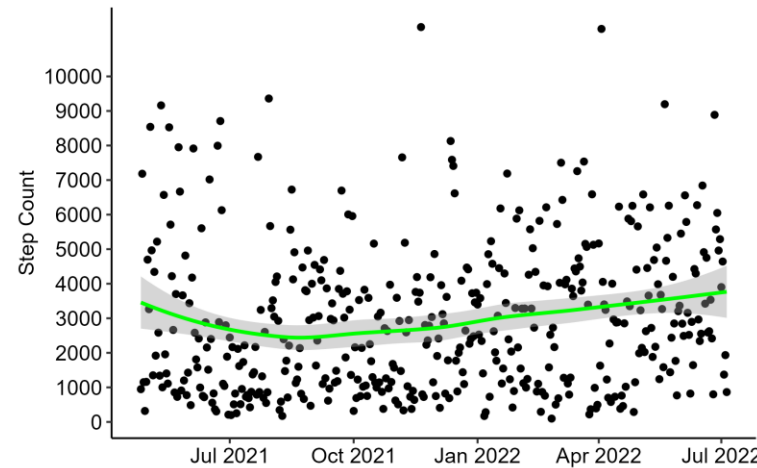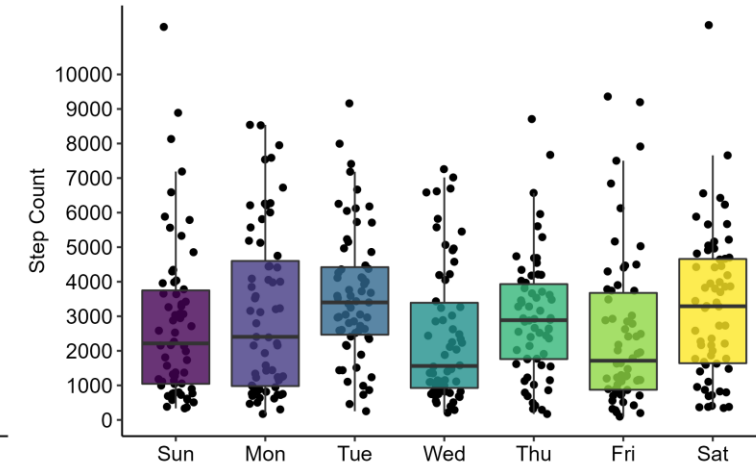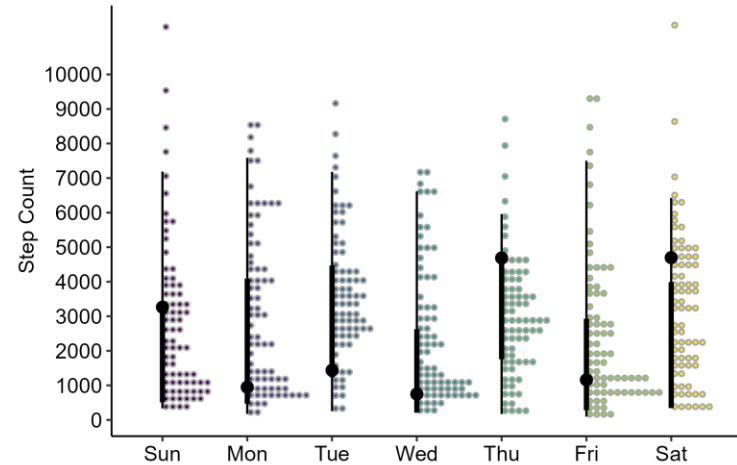
# *Adding a Repo*

- Name and description.

- Local path to the repo (where the files are on your computer).

- Git ignore template to use (in our case R will work).

- License to use for the repo MIT and GNU are likely fine (particularly important for patentable research)

# Committing changes to a Repo

- Once you have a repo, adding any files into the repo folder will show up as a change in github desktop.

- We can choose to ignore a file (aka don't change or add to repo) if we like.
  - An example might be a dataset that contains GPS data you don't want to share
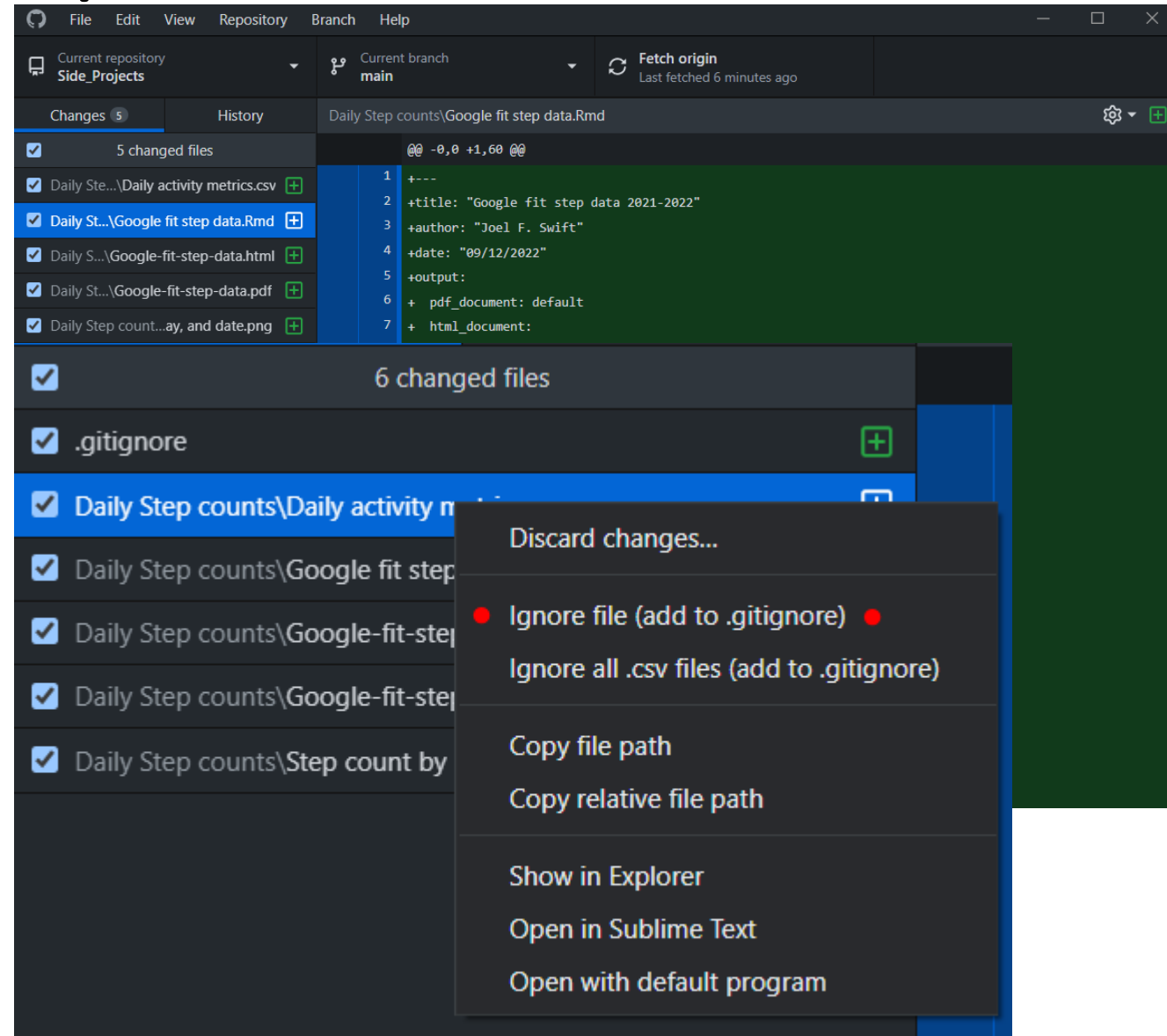
# Committing changes to a Repo

- Commit messages can be important to remember what changes were made.



Added new project, google fit step count

Google fit data was downloaded for 2021 and 2022. I looked at trends in the data by day of the week, month, and date.

Data is included in .gitignore as it contains GPS data I'd rather not share :)

Commit to **main**



BOB, ALL YOUR COMMIT MESSAGES SAY "SMALL CHANGES"!

NEXT TIME, PLEASE WRITE A MEANINGFUL MESSAGE

MEANINGFUL MESSAGE...
MEANINGFUL MESSAGE...
MEANINGFUL MESSAGE...
MEANINGFUL MESSAGE...
MEANINGFUL MESSAGE...

GIT COMMIT -M "WHAT IS LIFE IF FULL OF CARE. IF WE HAVE NO TIME TO STAND AND STARE..."
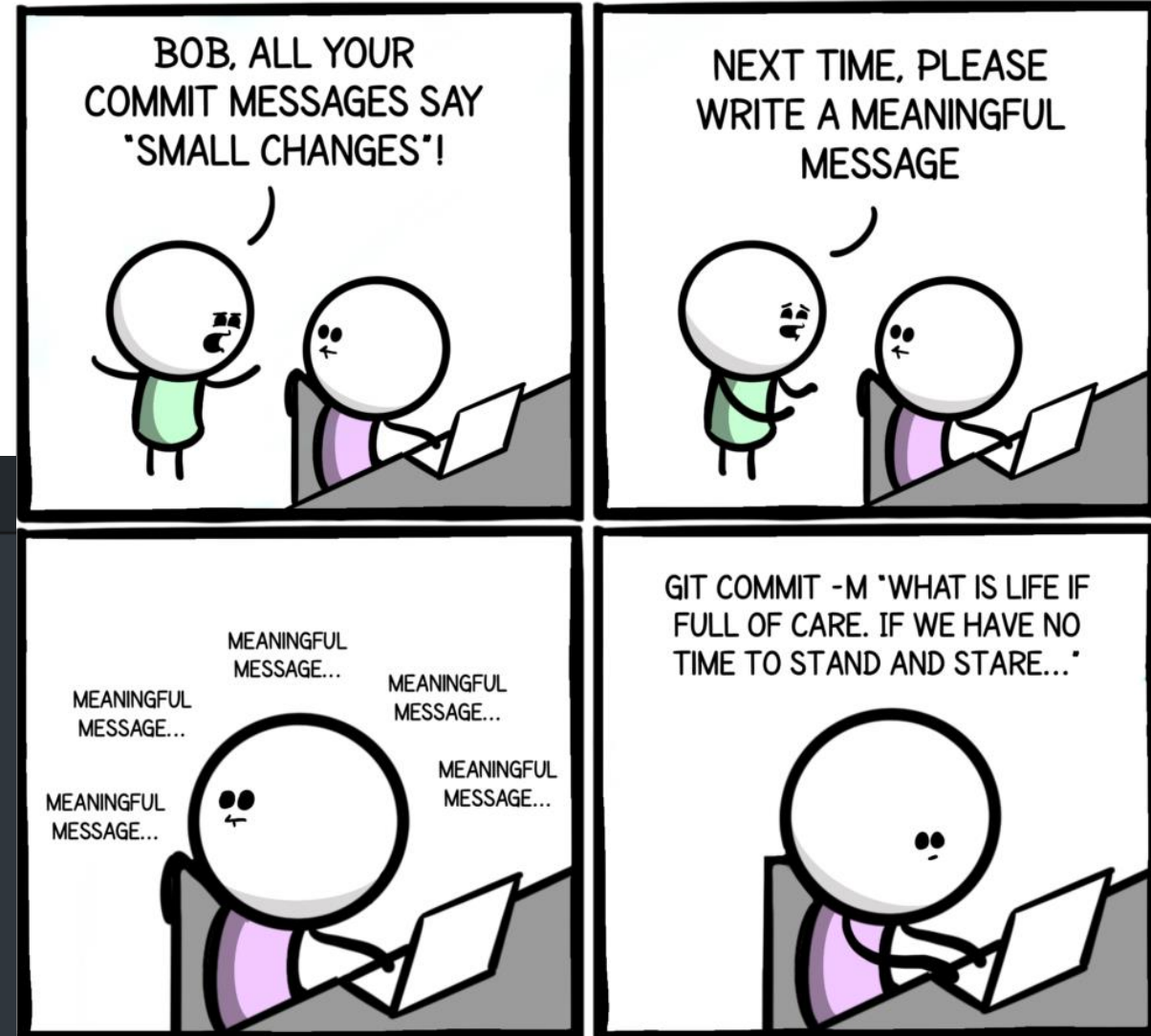
@_workchronicles
follow on *Twitter* & *Instagram* for more (and don't forget the **underscore**)

workchronicles.com

# *Committing changes to a Repo*

- Commit the change.

- Choose the branch.

- Push the change to the branch.
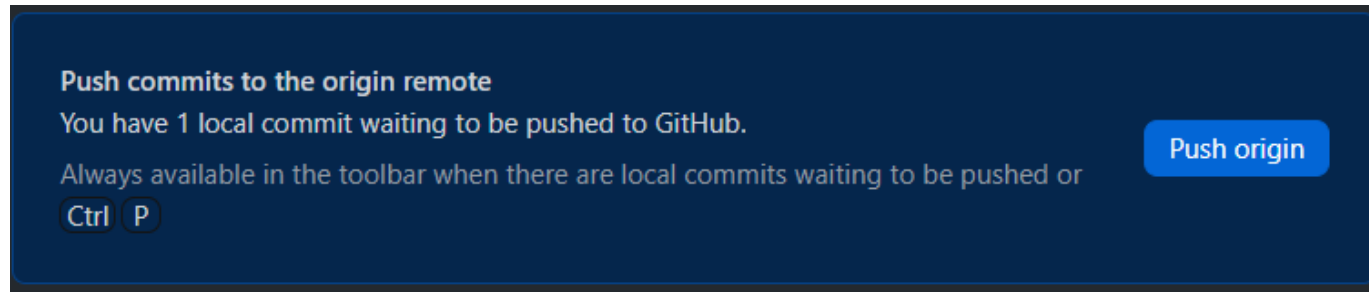
# Lets look at the repo that was changed

- [https://github.com/Kenizzer/Side_Projects](https://github.com/Kenizzer/Side_Projects)

# Working with others
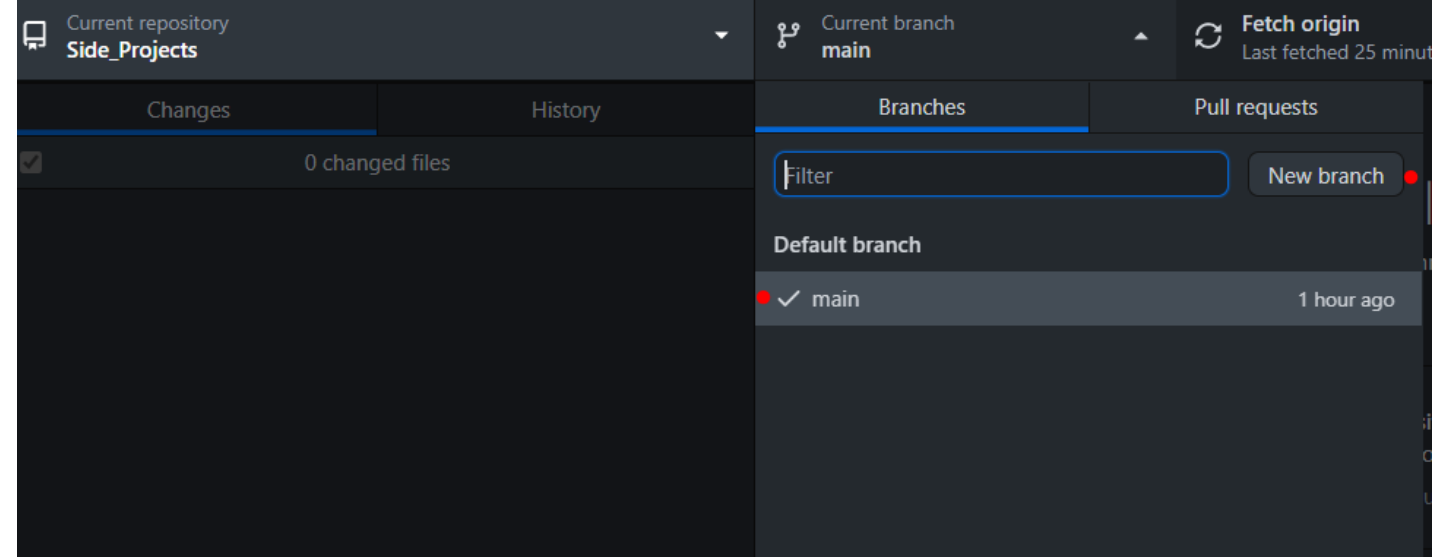# Branching and Forking



Fork vs Branch
Comparison Chart

| Fork | Branch |
|------|--------|
| A fork is nothing but a copy of the repository to your own scope without affecting the original project repo. | A branch is an isolated environment to add, modify or delete a portion of the code without messing with the main code base. |
| Forking is a cloning operation in Git that is executed on the entire repository level. | Branching is a cloning operation in Git executed on a single repository. |
| Forking creates a full copy of the original repository which sits in your account. | Branching creates a branch to implementing your changes without affecting other developers. |
| The purpose is to improve someone else's project by adding some new features to the existing repo. | The purpose is to divert from the original code base without affecting other developers' work. |

DB Difference Between.net

- Fork ... aborator on.
- Crea ... re acti ... he main cod ...



I SEE, YOU'VE BEEN USING GIT FOR SEVERAL YEARS NOW?

SO YOU'RE GOOD AT MERGING, FORKING, BRANCHING?

# *Branches*



- Typically there will be a single main branch (often named main or master[1]) for a repo.

- When you would like to work in a new environment, you make a new branch.
  - Why? Often its to try to fix or add things that have the potential to break other parts of the code. This way an important software can have a stable (main) branch and a dev branch for experimenting.

[1] https://www.theserverside.com/feature/Why-GitHub-renamed-its-master-branch-to-main

# *Branches*



- Change committed!

# The dreaded Merge Conflict



- What happens when we attempt to merge branches that have changes to the same file? Which version do we choose?

- Merge Conflicts!!

- Test ->

- Main ->

# *Pull request*

Add more commits by pushing to the `Test_branch` branch on **Kenizzer/Side_Projects**.

⚠ **This branch has conflicts that must be resolved**
Use the web editor or the command line to resolve conflicts.
**Conflicting files**

README.md

Merge pull request  ▾     You can also open this in GitHub Desktop or view command line instructions.

# Updated the readme for the google fit project #1

Resolving conflicts between `Test_branch` and `main` and committing changes → `Test_branch`

1 conflicting file

README.md
README.md

README.md

```
1        # Side projects
2
3        ### Google fit step data
4        <<<<<<< Test_branch
5        ***Current state: Get to stepping, needs more data***          ●
6        =======
7        ***Current state: We are tired, no more data needed***         ●
8
9        >>>>>>> main
10
```

README.md                                    1 conflict   Prev ∧   Next ∨   ⚙▾   **Mark as resolved**

```
1     # Side projects
2
3     ### Google fit step data
4     ***Current state: Get to stepping, needs more data***
5
6
7     ### Lyrics for US punk rock bands in a machine learning data object.
8     ***Current state: Machine learning data object built but needs cleaning up before analysis***
9
10
11    ### NFL margin of victory for all NFL teams 2017-Current.
12    ***Current state: Get scores for games farther back***
13
14
15    ### NFL boxscore stats scrapped from ESPN.com 2002-2019.
16    ***Current state: Making interactive vizualizations with ggplotly and exploring data***
```

- Resolved by deleting one of the lines about the google fit project.

# Commit has been merged after resolving the conflict

# Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

| Owners | lov | Dec | Feb Sep | |
|--------|-----|-----|---------|---|



---
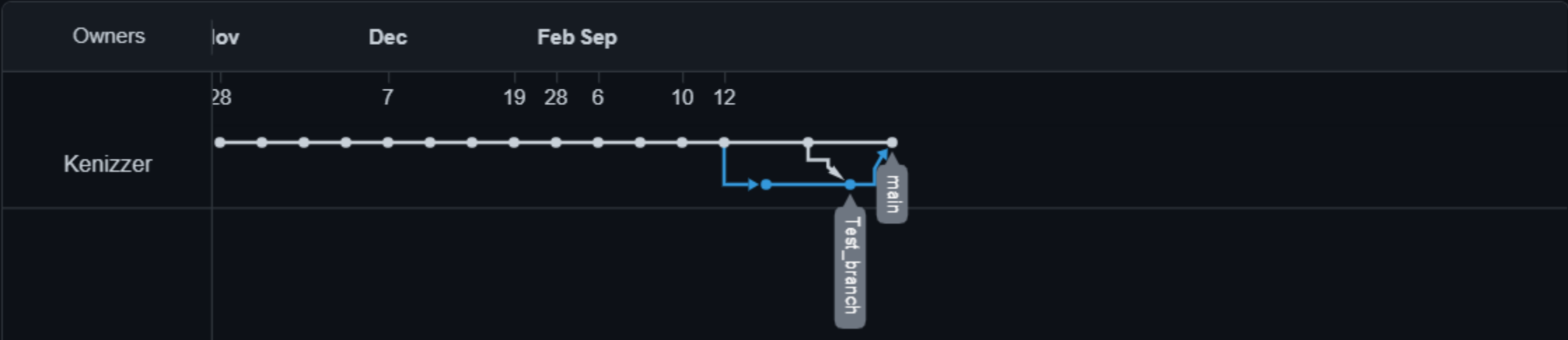


Conversation 0    Commits 1    Checks 0    Files changed 1

**Kenizzer** commented 7 minutes ago          Owner  ☺ ⋯

Small edit to readme

-○- Updated the readme for the google fit project  ⋯          616d0a3

-○- Merge branch 'main' into Test_branch          Verified  345a79e

**Kenizzer** merged commit `fec3692` into `main` 4 minutes ago          Revert

---

**Kenizzer** deleted the `Test_branch` branch now          Restore branch

# *Forks*

- Create forks when you would like to contribute to a project you did not create.

- Example: The 2022 KU symposium had an repo on github, I wanted to make the 2023 repo.

# *Forks*

- Now we clone the directory locally.

- Then we make our changes and contributions and commit and push them.

# Open a pull request

# *Open a pull request*

- Notice the message that we can auto merge?

- Why is that?

# Add E's README, deleted broken files, added 2023 folder #1

Conversation 0 | Commits 1 | Checks 0 | Files changed 18

+0 −0

**Kenizzer** commented 29 seconds ago

- Added 2023 folder with the same directory structure as 2022.
- Renamed the access instructions to README so that github will display it natively (Nice markdown/instructions E)!
- Removed just a couple werid files that I think crop up because of Mac/external harddrives.
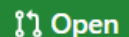- Added minutes for 9-13-2022

Add E's README, deleted broken files, added 2023 folder `...`          06f3f48

Add more commits by pushing to the `main` branch on **Kenizzer/KUCG_SymposiumPlanning**.

✓ **This branch has no conflicts with the base branch**
Only those with write access to this repository can merge pull requests.

Write | Preview

H B I ≔ <> 🔗 ☰ ⊟ ☑ @ ↗ ↩

Leave a comment

**Reviewers**

No reviews

Still in progress? Convert to draft

**Assignees**

No one assigned

**Labels**

None yet

**Projects**

None yet

**Milestone**

No milestone

**Development**

*Questions?*



GitHub is old-fashioned.