INTRODUCTION TO VERSION CONTROL SYSTEMS

TODAY…

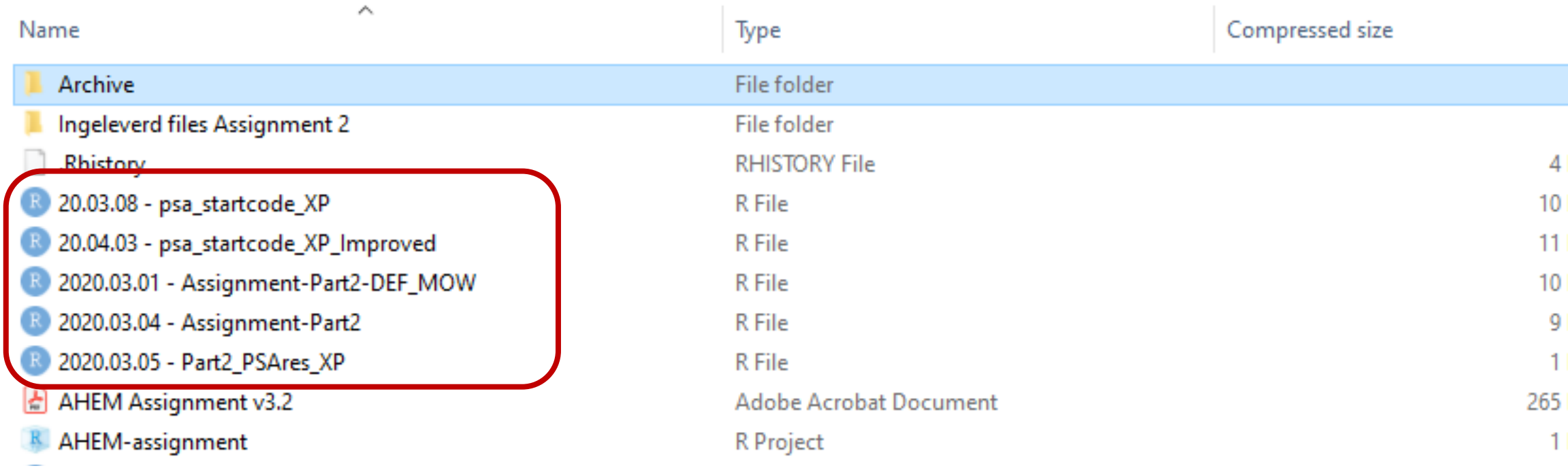
We are going to talk about a life-changing event for me…

It happened in the year 2021…

When I discovered GitHub!

SHORT CHECK

- Raise your hand if your files also look like this



SHORT CHECK

- How do you keep track of changes you make in source code / script?
- Do you have experience with collaborating on a script?
  - How did you prevent conflicts and ensured the possibility to work simultaneously?

PREREQUISITE

- Basic knowledge of the statistical software ‘R’ and R-studio

LEARNING OBJECTIVES

- After this lecture, you should be able to:
  - Explain what are version control systems
  - Explain why using version control systems
  - Explain the (dis)advantages of using version control systems

VERSION CONTROL – WHAT?

- System to:
  - Systematically track changes made in scripts (and files)
  - Build a history of changes made

VERSION CONTROL – WHY?

- Allows checking adaptations
- Allows to re-use previous versions
- Avoid files' duplication
- Prevent (damages of) deleting the 'wrong' files
- Facilitate collaboration on a project

VERSION CONTROL – HOW?

1. Log table

LOG TABLE

- Keep track manually of:
  - Version number
  - Sometimes: file of each version number
  - Changes made
  - Person who made changes
  - Date of changes
  - Status, eventually (e.g., draft, revised, final)

LOG TABLE EXAMPLES

Title				
Description				
Created By				
Date Created				
Maintained By				
Version Number	Modified By	Modifications Made	Date Modified	Status

Source: The University of Sydney

Table 1: Document Version Control Log

Version	Date	Description of changes and person responsible for making changes
1.0	23/2/2015	Initial draft summary for comments from workshop participants (Craig Sinclair)
1.1	25/2/2015	Revision incorporating feedback from workshop facilitators (Craig Sinclair)
2.0	6/3/2015	Inclusion of submissions from two facilitators (Angus Cook, Amar Varsani) and addition of feedback points raised by 13 survey respondents (Craig Sinclair)

Source: Sinclair et al. 2015

WHAT ARE THE DRAWBACK OF SUCH APPROACH?

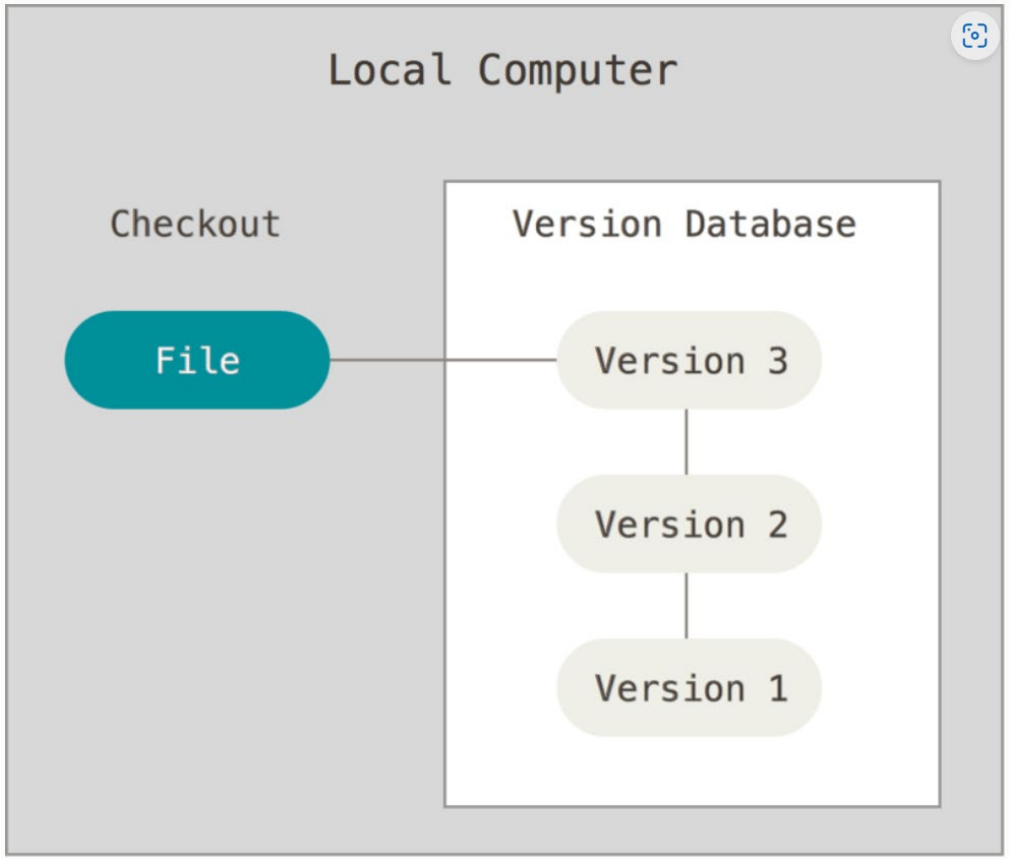
- Time consuming
- Still: save multiple version of a file
  - If not: you cannot go back to a pr
  - If yes: takes up disc spac
- You can still delet wrong files
- Collaborati
  - W p-to-date file!

Solution?
Let a software do this for you = version control system!

VERSION CONTROL – HOW?

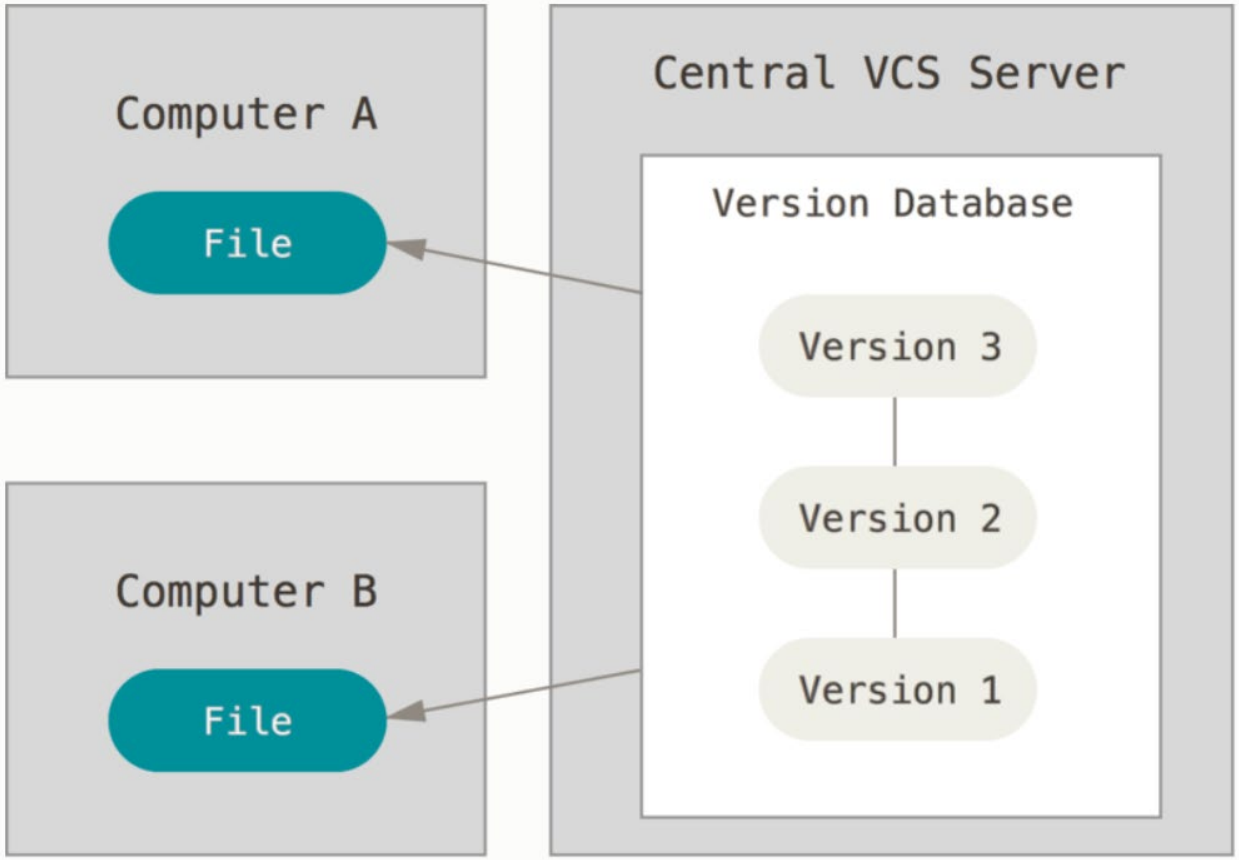
1. Log table
2. Version control systems
   - Local, centralized, **distributed**
   - Softwares, such as '**Git**', Mercurial
   - https://en.wikipedia.org/wiki/List_of_version-control_software

LOCAL VERSION CONTROL SYSTEM



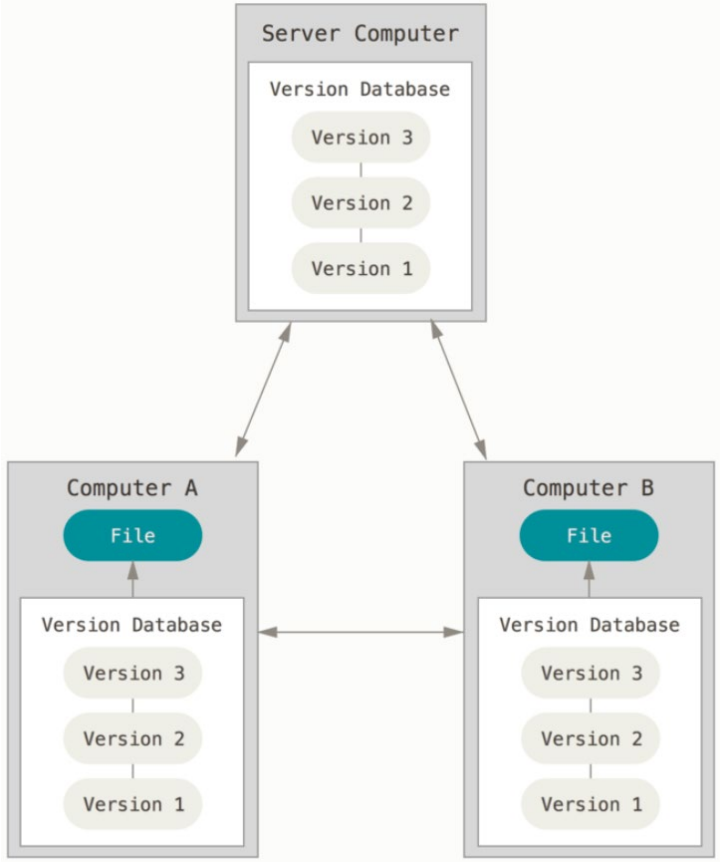
Source: Chacon & Straub

CENTRALISED VERSION CONTROL SYSTEM



Source: Chacon & Straub

DISTRIBUTED VERSION CONTROL SYSTEM



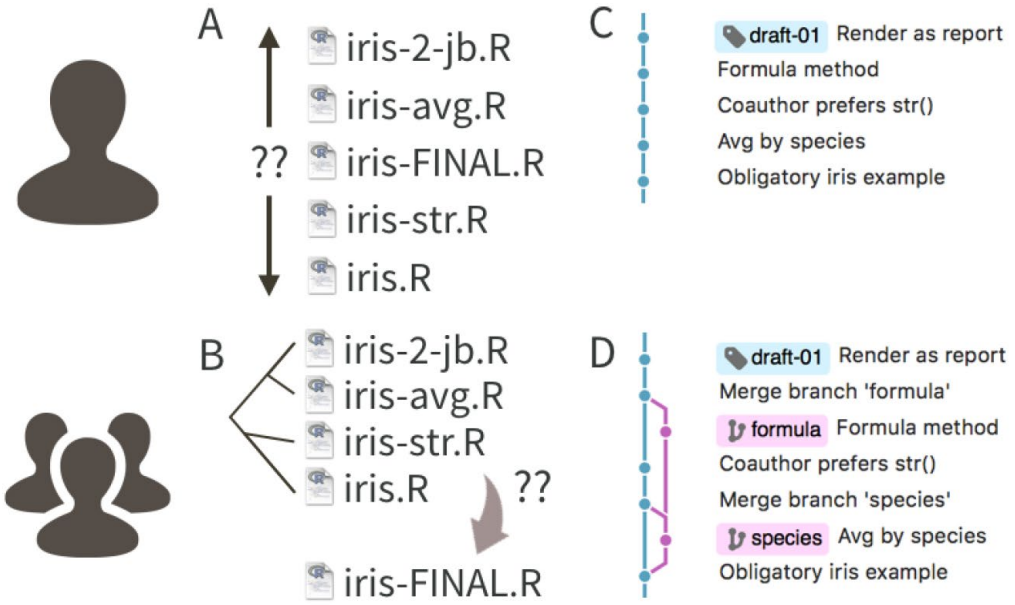
Source: Chacon & Straub

SOFTWARE – GIT

- Git
  - Open-source **distributed** version control system / software
  - Make a local copy of a repository
  - Multiple people can work simultaneously on the same project
  - Allows remote and offline repository / code adaptation
  - Synchronisation with repository on server
  - Integrated in all major script-based development softwares (such as R)
  - Components:
    - repos - repositories
    - branches
    - commits
    - push / pull
    - ...

WHAT IS GITHUB?

- https://www.youtube.com/watch?v=pBy1zgt0XPc
- Advantages of Git & GitHub >> integration in R!
- Difference Git & GitHub
  - Git = software
  - GitHub = web-host Git system
- Alternatives to GitHub:
  - GitLab
  - Codeberg (European)

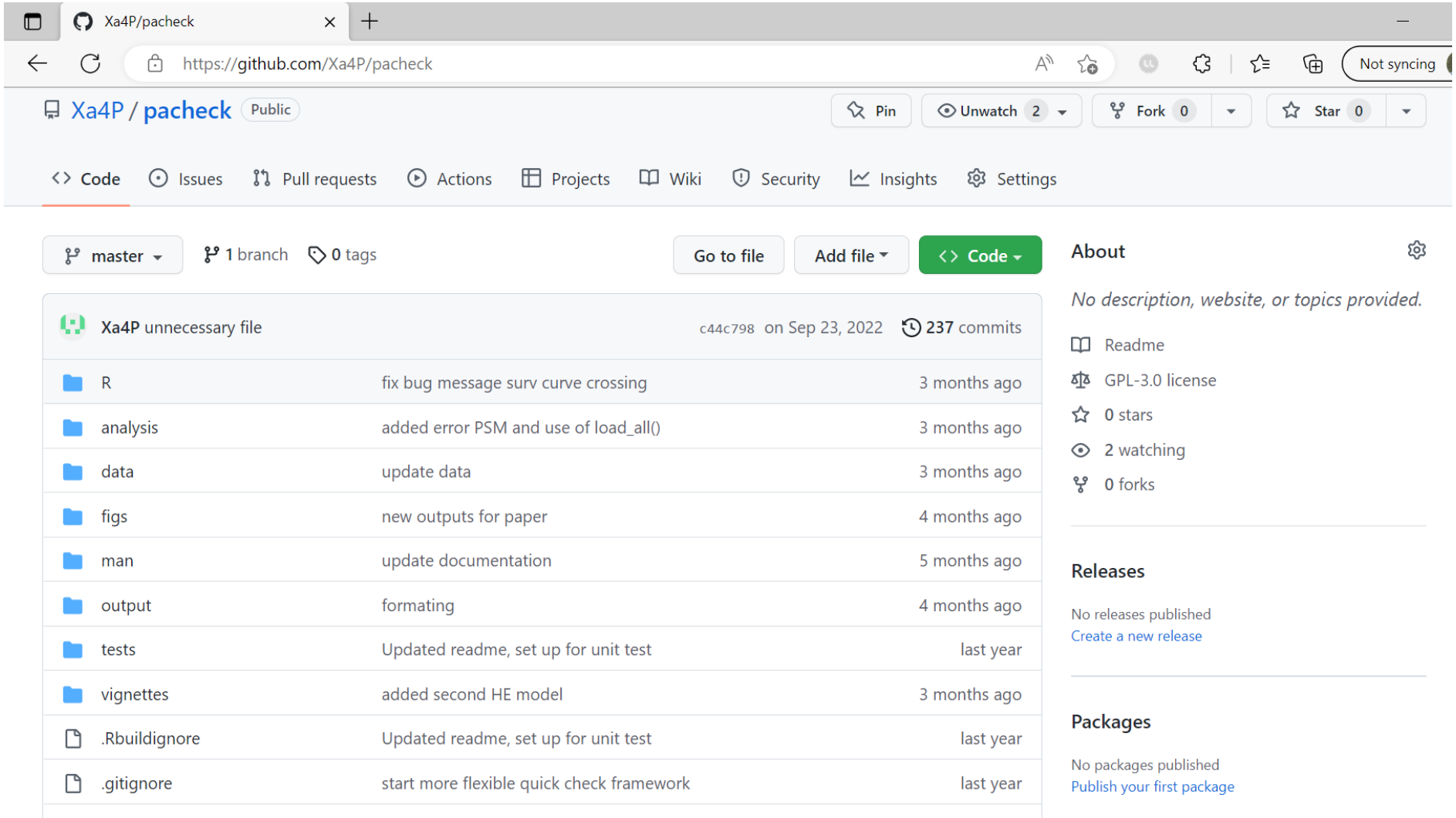


Source: Bryan – 2017

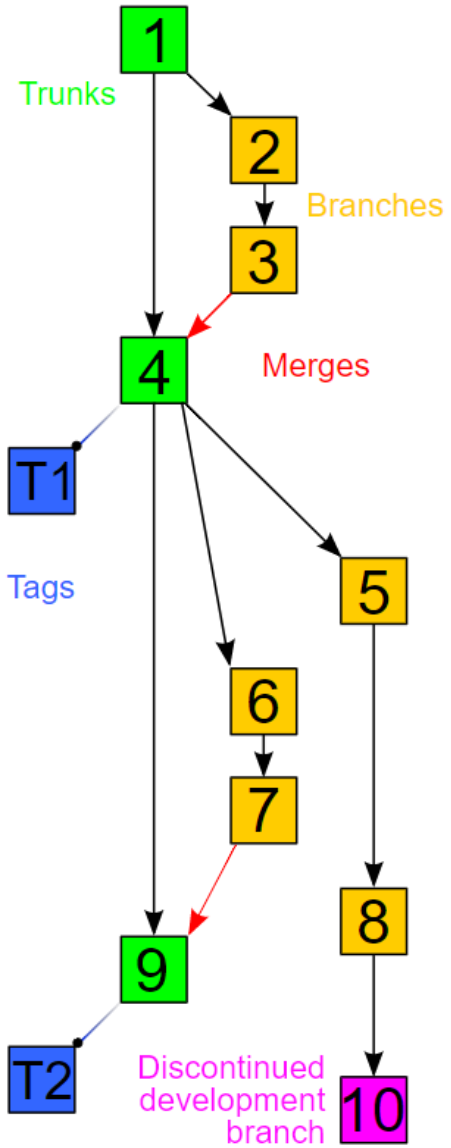
GIT(HUB) BASICS

- Repos – repositories
  - Contains all the files of your project
  - ‘Equal’ to R-project
- Branch
  - A (local) copy of the (main) repository
- Commit
  - A stamped version of a branch
- Pull
  - Action to ‘download’ a version (commit) of the online repository
- Push
  - Action to ‘upload’ your version (commit) of the repository

HOW DOES IT LOOK LIKE? REPOSITORY



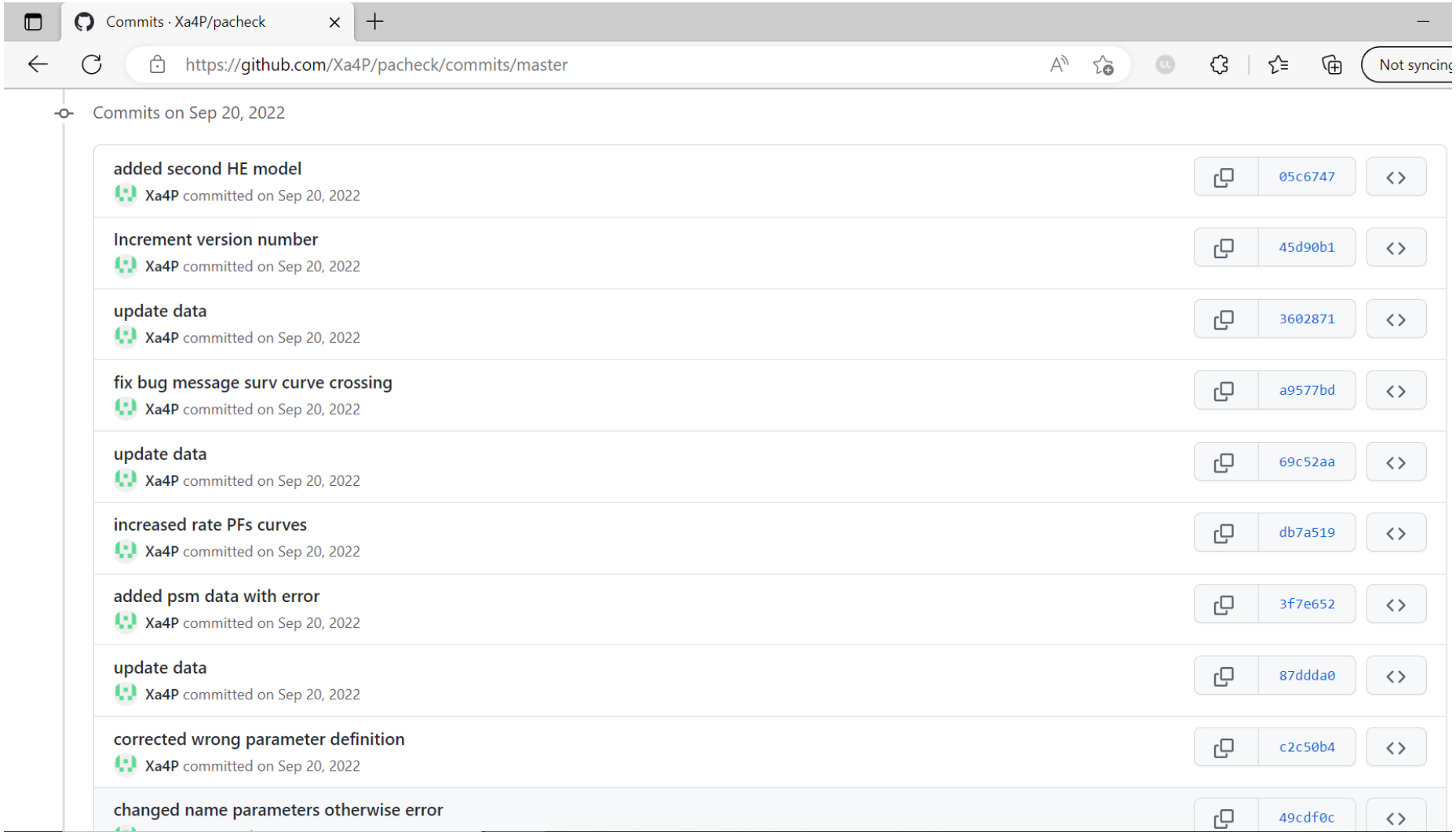
HOW DOES IT LOOK LIKE? BRANCHES



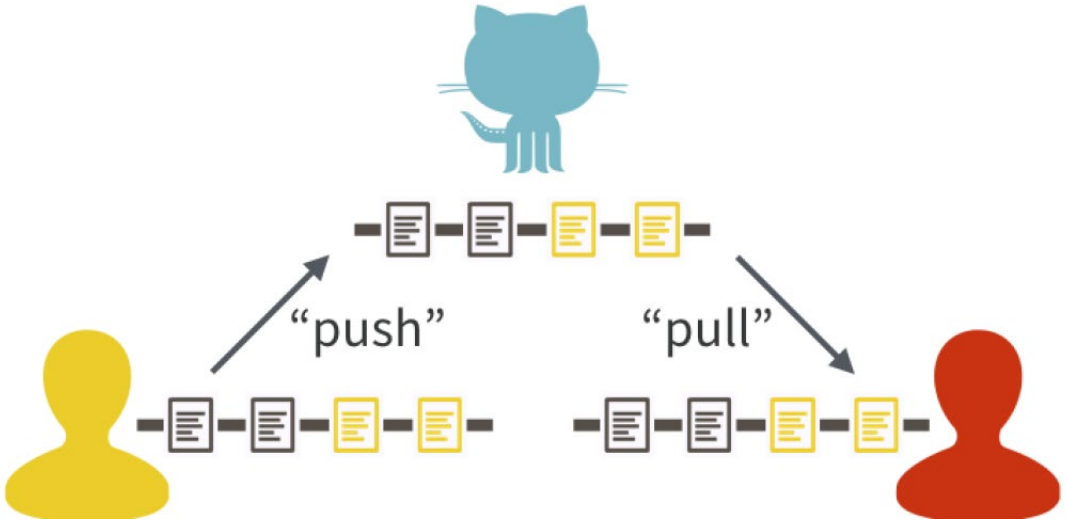
Source: https://commons.wikimedia.org/wiki/File:Revision_controlled_project_visualization-2010-24-02.svg

Licence: CC BY-SA 3.0

HOW DOES IT LOOK LIKE?
COMMIT HISTORY



GIT(HUB) & COLLABORATION

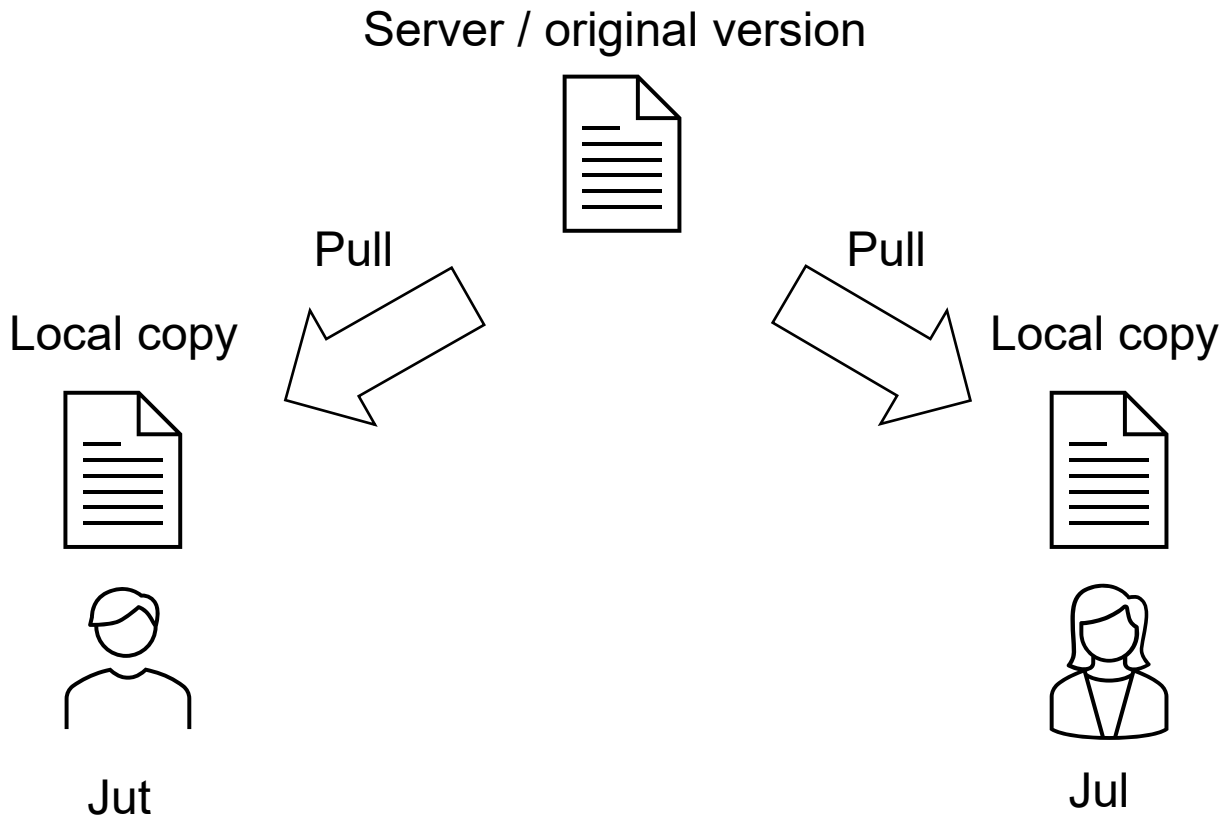


Source: Bryan – 2017

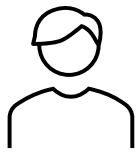
Users of your repo can:

- Download your version of the repo
- Use it
- Raise issues
- Create pull requests
  - Make changes and discussing it before merging in main branch

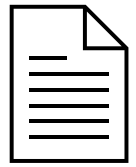
GIT(HUB) & COLLABORATION



GIT(HUB) & COLLABORATION

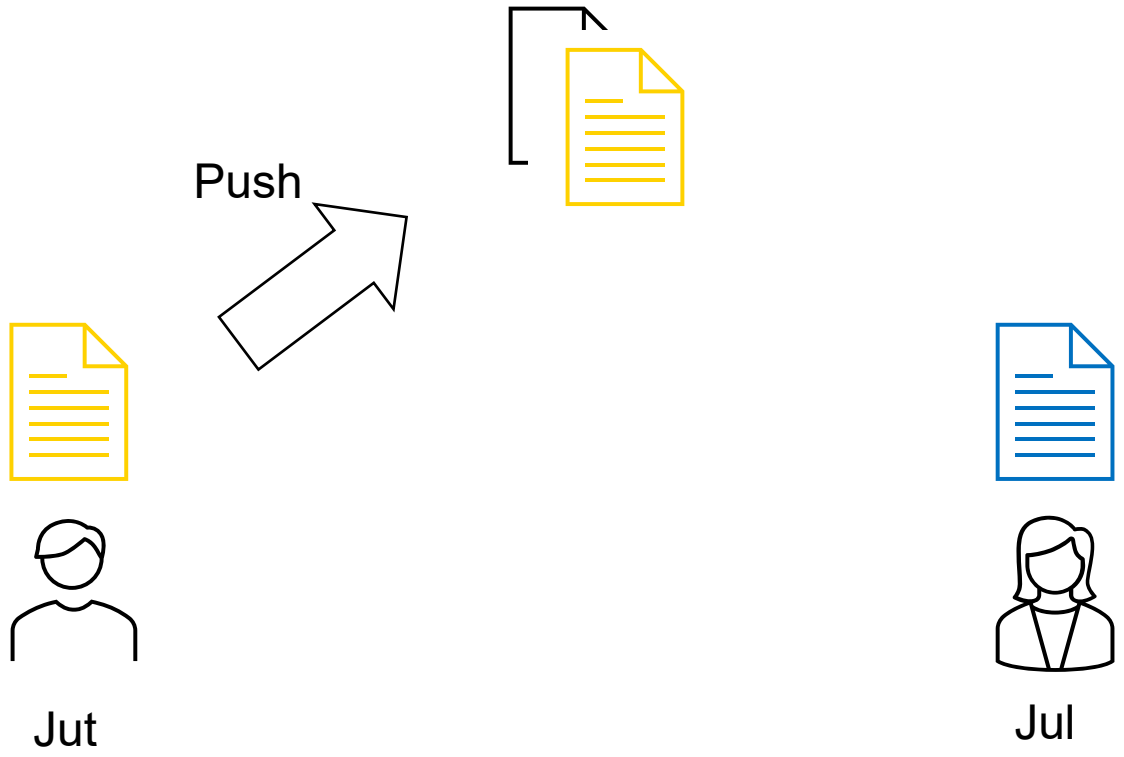


Jut

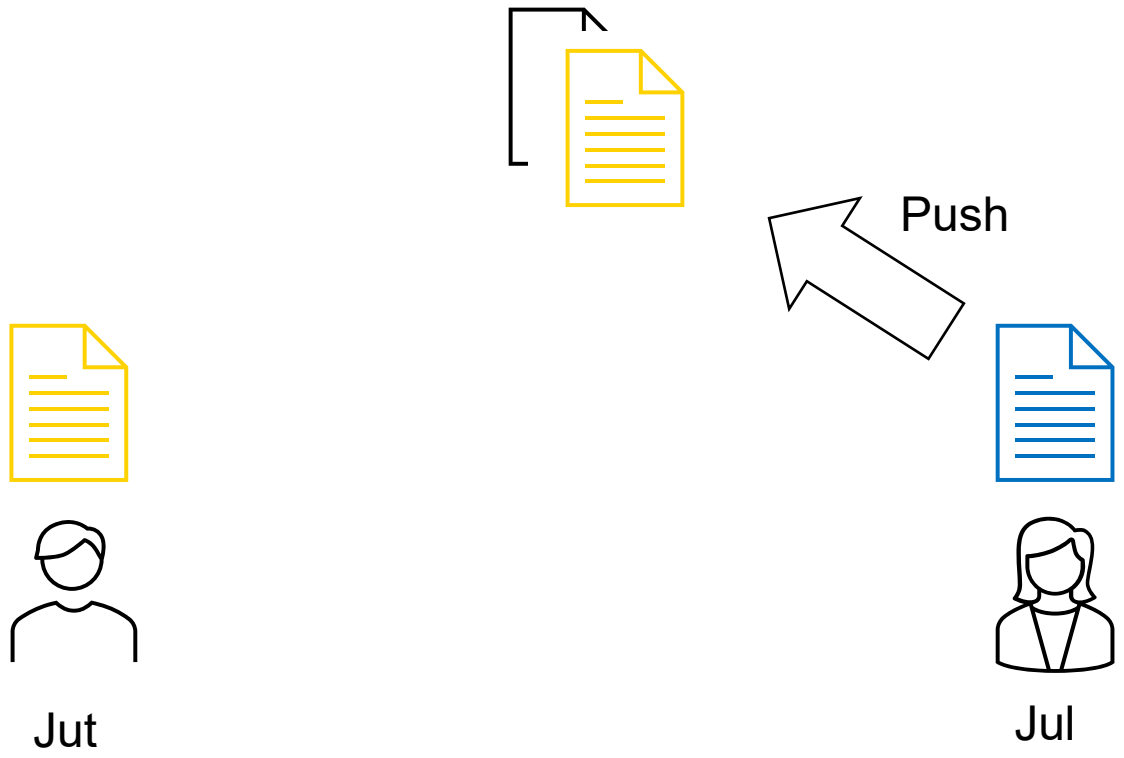


Jul

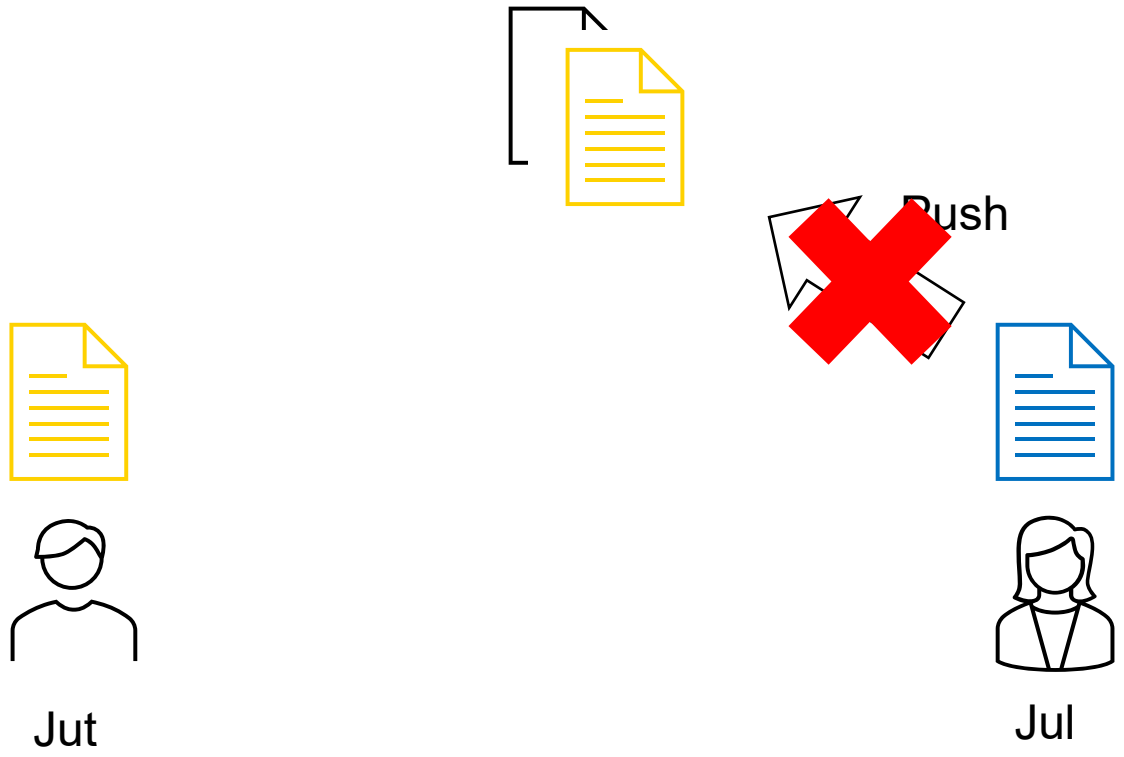
GIT(HUB) & COLLABORATION



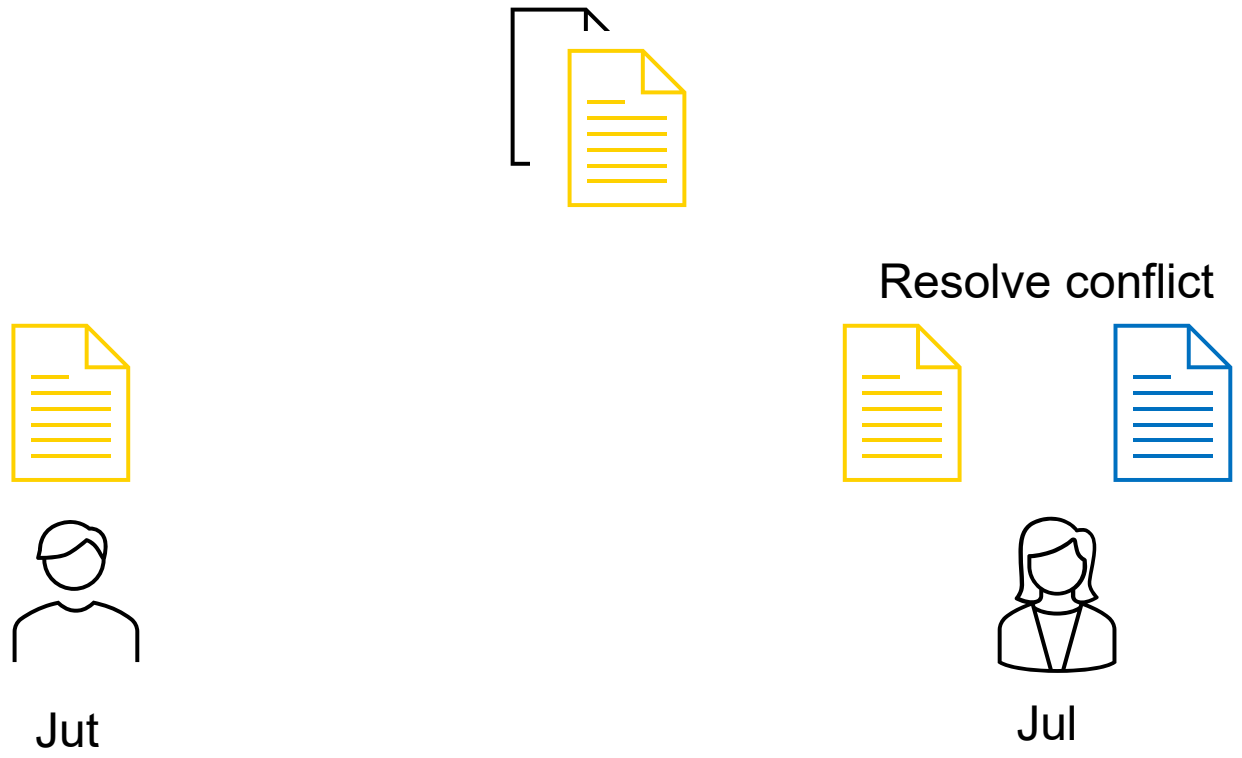
GIT(HUB) & COLLABORATION



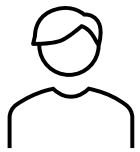
GIT(HUB) & COLLABORATION



GIT(HUB) & COLLABORATION



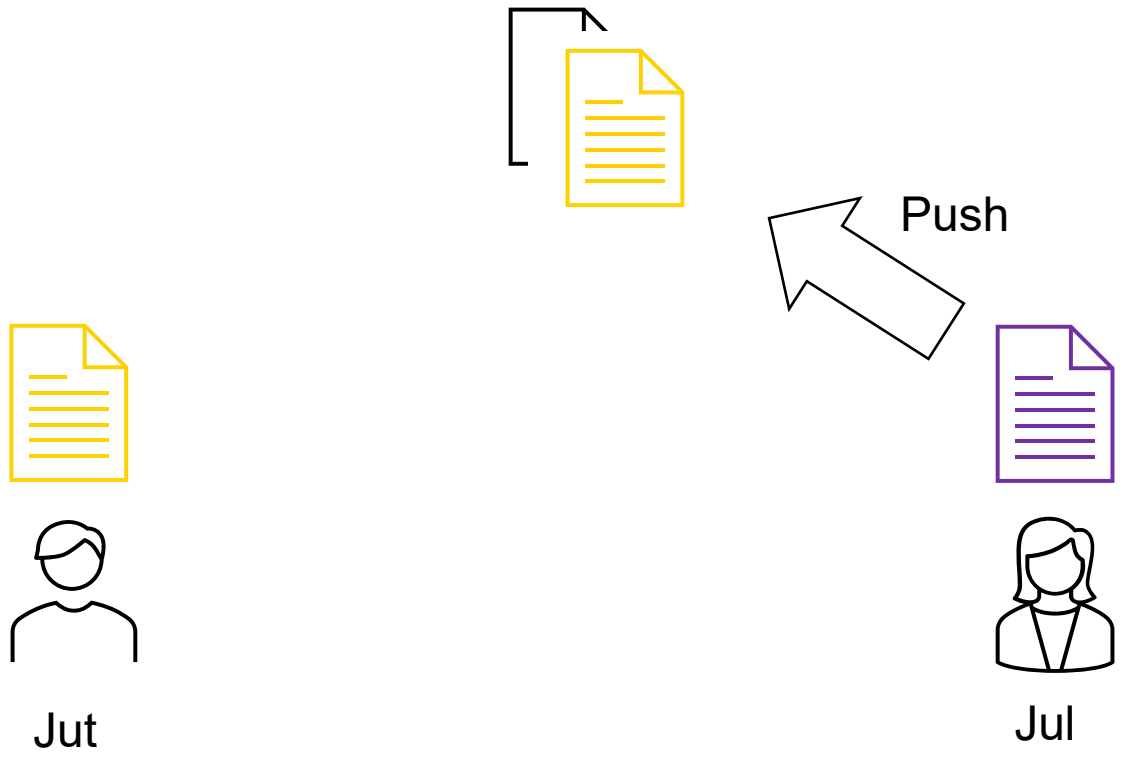
GIT(HUB) & COLLABORATION



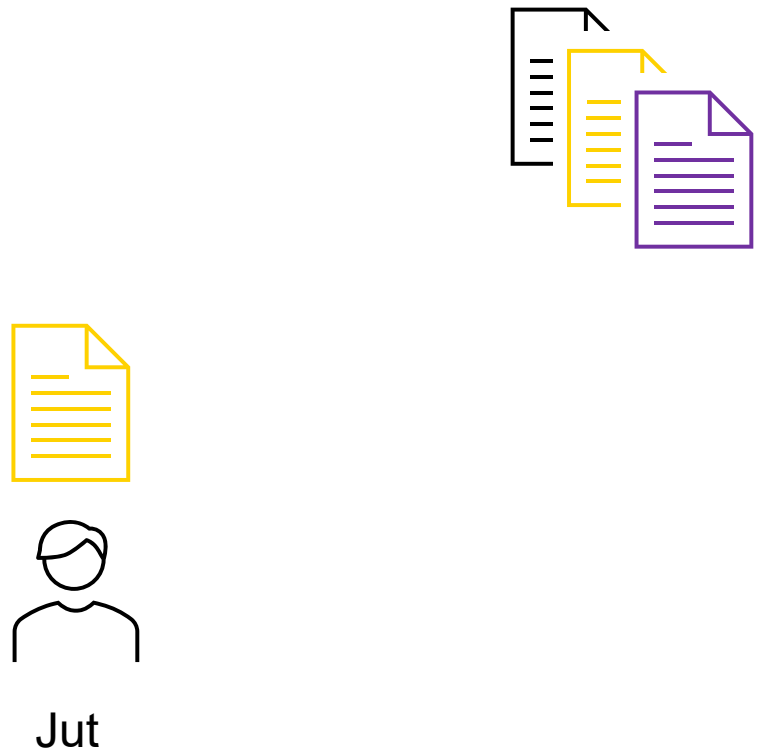
Jut

Jul

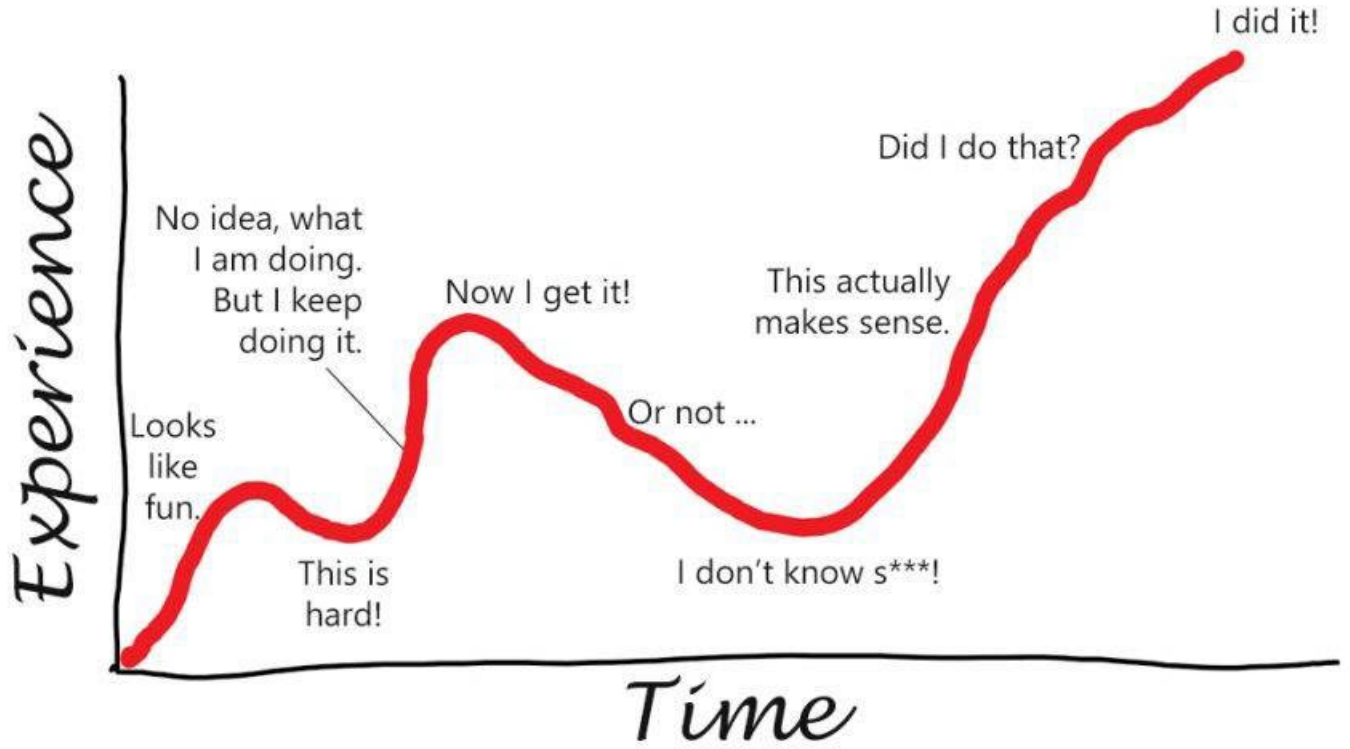
GIT(HUB) & COLLABORATION



GIT(HUB) & COLLABORATION



WARNING!



Source: https://sascha-kasper.com/the-bumpy-learning-curve/

GIT(HUB) & R

1. Create a Github / GitLab / Codeberg / … account
2. (Install or upgrade R and Rstudio)
3. Install Git (if it is not already)
4. Connect GitHub with R(studio)
5. Preferred approach when working with github:
   - Github first, then R studio

Explanation of the Github – R interaction: Bryan, J. Happy Git and GitHub for the useR: https://happygitwithr.com/index.html

WRAP UP

- Reasons why version control system will change YOUR life:
  - No more worries about:
    - Deleting, loosing, or overwriting (**CRUCIAL**) files
    - File duplication
  - It creates a history of changes
    - You can review these changes
    - You can always go back!
  - It facilitates collaboration
- But… there is a steep learning curve!

ONE FINAL NOTE

- If you use GitHub (or other) for this course, please keep the repository private

ANY QUESTION?

DO IT YOURSELF!
START USING GIT(HUB)

- GitHub official starting guide: https://docs.github.com/en/get-started/quickstart/hello-world

SOME HE – RELATED EXAMPLES

- DARTH: https://github.com/DARTH-git
- IVI: https://github.com/InnovationValueInitiative
- Synthea: https://github.com/synthetichealth/synthea

RESOURCES

- Introductions to version control system:
  - https://www.youtube.com/watch?v=zbKdDsNNOhg
  - https://www.youtube.com/watch?v=gY2JwRfin1M
- Git workflow
  - https://www.youtube.com/watch?v=3a2x1iJFJWc

RESOURCES

- Chacon, Scott, and Ben Straub. Pro git (2nd editopm) Apress. Accessed at https://git-scm.com/book/en/v2 on 3 August 2023
- Bryan, J. Excuse me, do you have a moment to talk about version control? (2017). PeerJPreprints. https://doi.org/10.7287/peerj.preprints.3159v2
- Github Foundation. GitHub & Git Foundations videos playlist. (2014). https://www.youtube.com/playlist?list=PL0lo9MOBetEHhfG9vJzVCTiDYcbhAiEqL
- Microsoft. DevOps resource center: *What is version control?* (2022). https://learn.microsoft.com/en-us/devops/develop/git/what-is-version-control accessed on 16-12-2022
- Microsoft. DevOps resource center: *What is git?* (2022). https://learn.microsoft.com/en-us/devops/develop/git/what-is-git accessed on 16-12-2022
- Sinclair C et al. Healthy Futures Forum 2015: Regional Health Research Priorities Workshop. (2015). Great Southern Science Council: Albany. [Available from http://www.greatsouthernsciencecouncil.org.au]. PDF accessed on 16-12-2022 at https://www.researchgate.net/publication/277511404_Healthy_Futures_Forum_Regional_Health_Research_Priorities_Workshop
- The University of Sydney. Version control document. https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fwww.library.sydney.edu.au%2Fresearch%2Fdata-management%2Fdownloads%2Fversion-control.docx&wdOrigin=BROWSELINK accessed on 16-12-2022