Git Basics

RSensus LabMeeting

Jeronimo Rodriguez-Escobar

2023-02-27

Git

Git is a *distributed version control* system. This **document** has a great introduction on Version control.

Git was originally developed to *to help groups of developers work collaboratively on big software projects*.

Its purpose is to manage the evolution of a group of files (known as *repo*). (Track changes on steroids)

In Data Science, it is used to manage all the files associatd with a data management project in a more structured manner:

- Keeps all files together
- Keeps track of all changes and allows to go back in time
- Useful to keep updated versions among multiple machines/users
- Facilitates teamwork and publishing

This **video** explains in very basic terms what git is about and the basic operations:

- git init
- git status
- git commit
- git branch
- git push
- git pull
- git fetch

Installation.

Find instructions to install Git **here**, in case it is not installed in your computer yet. Check if Git is installed with this command:

```
git --version
```

```
## git version 2.32.1 (Apple Git-133)
```

Git can be used directly on the command line or trough GUIs. RStudio offers a very very intuitive approach. Instructions for setting it up are **here**

Seting the user

```
git config --global user.name "Jeronimo Rodriguez"
git config --global user.email jeronimo.rodriguez@temple.edu
```

Check Settings

```
git config --list
```

```
## credential.helper=osxkeychain
## user.name=Jeronimo Rodriguez
## user.email=jeronimo.rodriguez@temple.edu
## core.editor=emacs
## credential.helper=osxkeychain
## core.repositoryformatversion=0
## core.filemode=true
## core.bare=false
## core.logallrefupdates=true
## core.ignorecase=true
## core.precomposeunicode=true
```

Setting the editor

I set EMACS as the default editor, it is not the only option, but the one I am familiar with

```
git config --global core.editor emacs
```

Starting a new repository

```
git init
```

```
## Reinitialized existing Git repository in /Users/sputnik/Documents/ARD_tuytorial/.git/
```

Check the Git Status

```
git status
```

```
## On branch master
## Changes to be committed:
##   (use "git restore --staged <file>..." to unstage)
##  new file:   ARD_tutorial copy.Rmd
##  modified:   ARD_tutorial.Rmd
##  new file:   ARD_tutorial.log
##  new file:   ARD_tutorial.tex
##  new file:   ARD_tuytorial.Rproj
##
## Untracked files:
##   (use "git add <file>..." to include in what will be committed)
##  .DS_Store
##  .gitignore
##  ARD_tutorial.html
##  Archive/
##  FkGClo1XoAEgs6O.jpg
##  Images_rmd/
##  git_tutorial.Rmd
##  git_tutorial.md
##  git_tutorial.pdf
##   git_tutorial_files/
##   matt/
```

Git Commit

```
#git commit
```

Create new branch

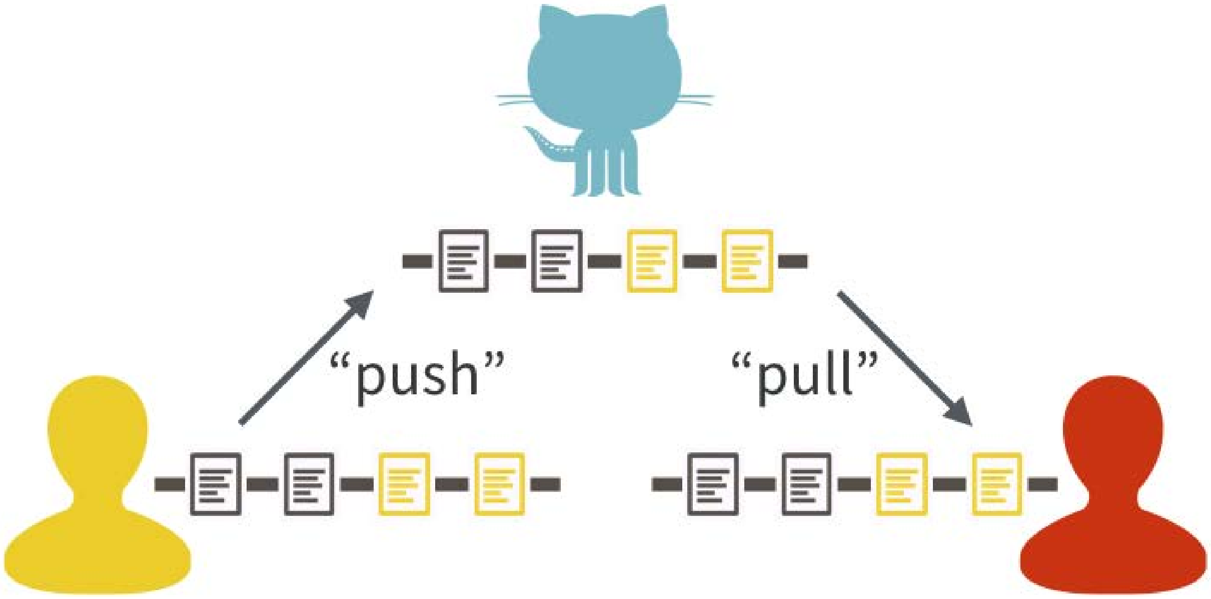
```
#git checkout -b master
```

```
#git checkout -b test1
```

Connecting to GitHub

It is very common to threat Git and Github indistinctly as the same thing, it is important to consider their relartionship:
GitHub complements Git by providing a user interface and a distribution mechanism for Git repositories

- Git is the software that records changes to a set of files.
- GitHub is a hosting service that provides a Git-aware home for such projects on the internet. (Bryan, 2017)



# Connecting with the Remote GitHub Repository

link

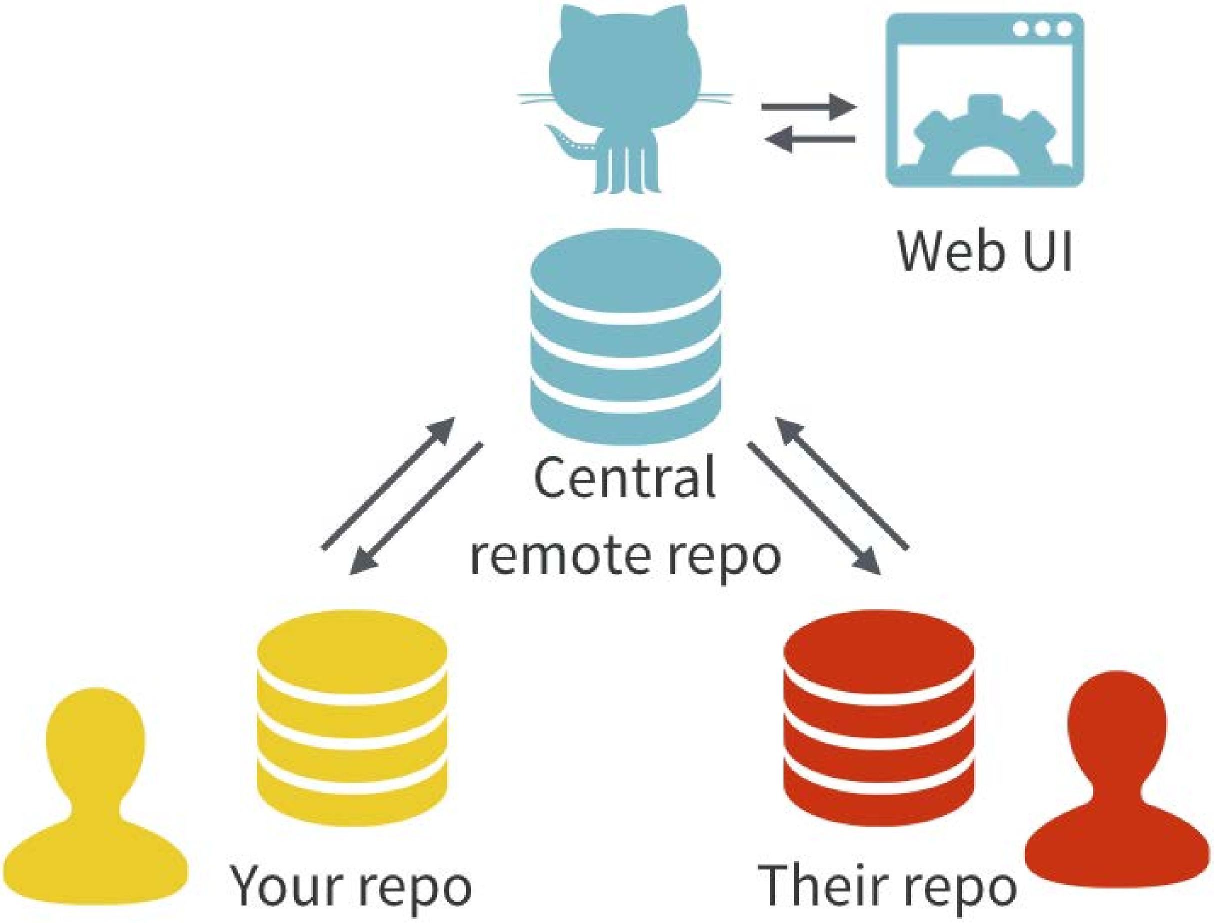


Figure 1: Git and Github