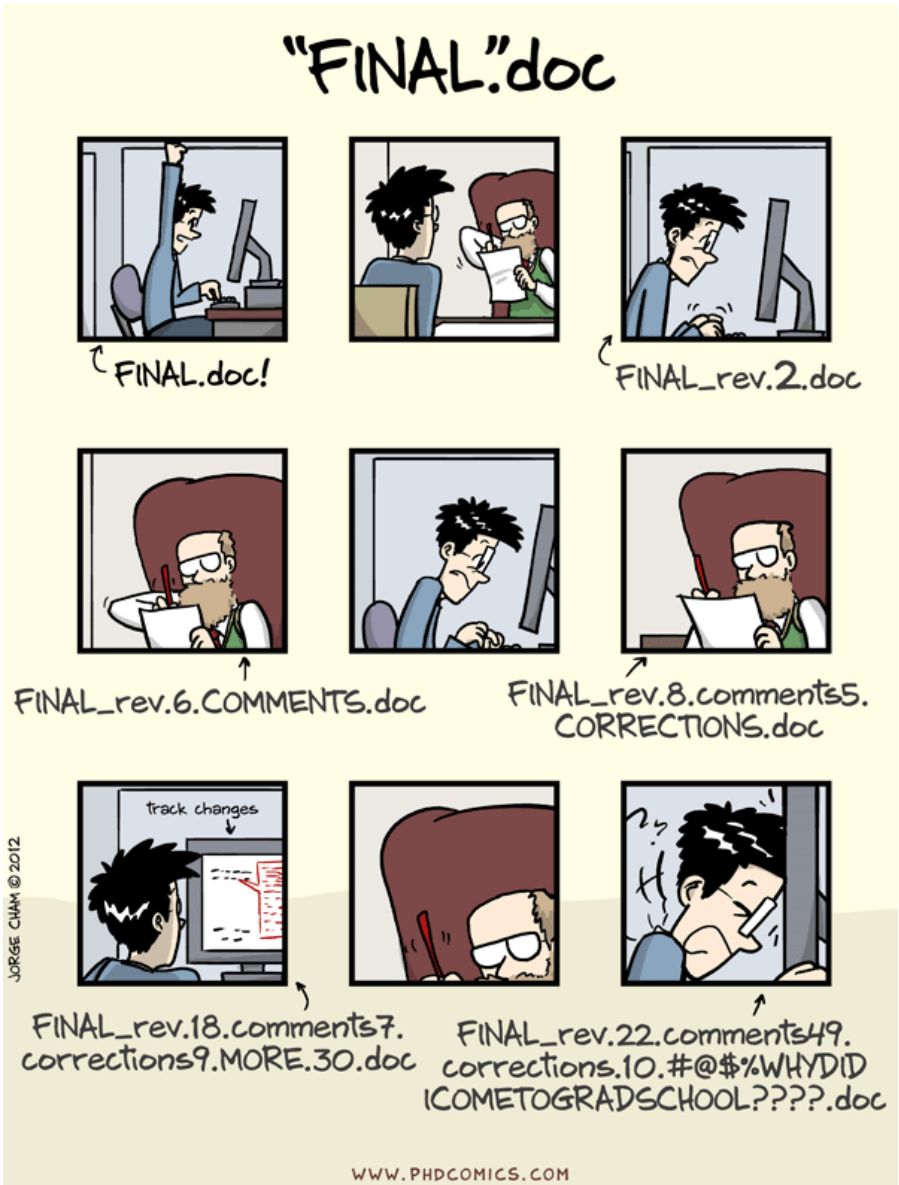
The Joy of Version Control

Jae Yeon Kim

Objectives



- Seriously, no more this (<-)
- Learning the concept of version control
  - Version control != Backup
- Practicing version using Git and GitHub
  - Git + GitHub = Time machine for computational projects

Let's be kind to your futureself, advisors, coauthors, reviewers, and so many others

Version control != Backup

- ??? I'm doing version control because I put backup data in Dropbox, Box, Google Drive, external hard drive, USB, etc ... (NOPE!)
- The only thing you need to backup is your RAW DATA (=READ ONLY)
- Everything else is subject to change and you should put them under version control.

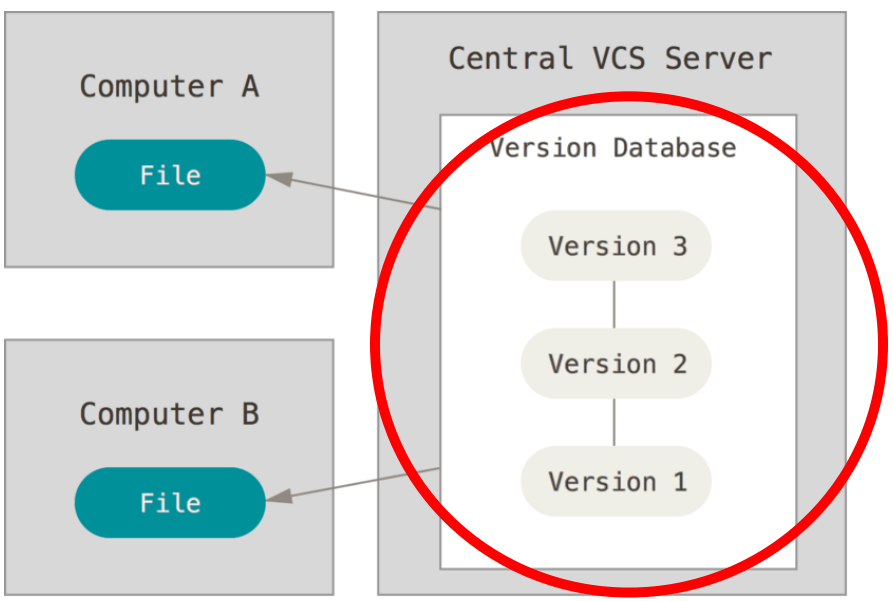
Version control = tracking the history of changes

1. Which changes were made?
2. Who made the changes?
3. When were the changes made?
4. Why were changes needed?

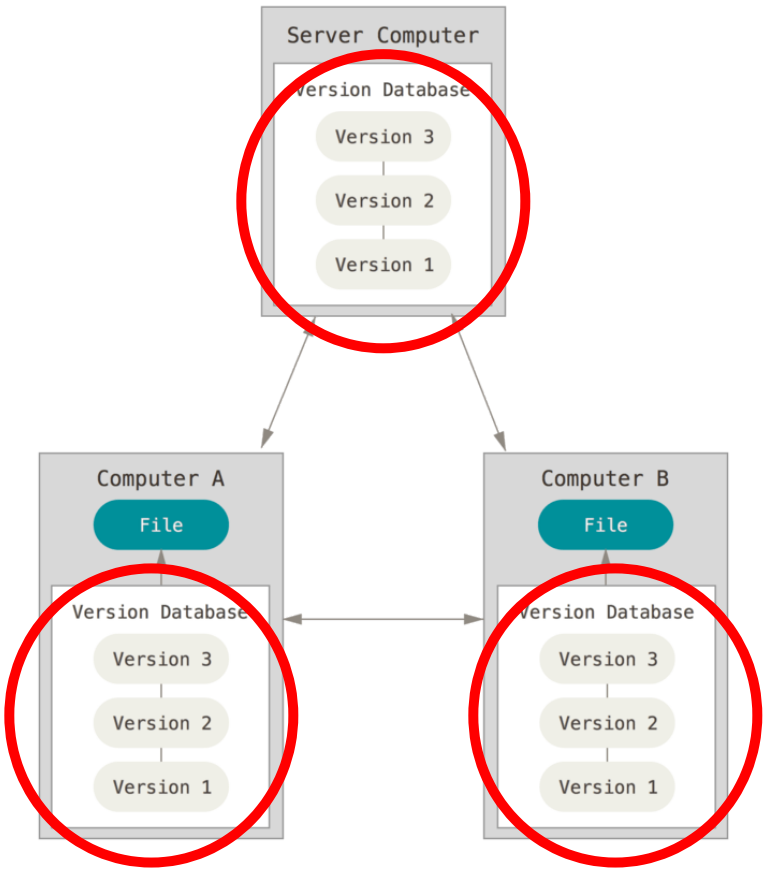
What if you're working in a team?

Solution:
Git & GitHub

Centralized (e.g., CVS, Subversion, Perforce)



Distributed (e.g., Git, Mercurial, Bazaar)



Easier, faster, and safer: strong support for *non-linear* and *distributed* development

Git

Git is a distributed version-control system for tracking changes in any set of files, originally designed for coordinating work among programmers cooperating on source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows. **Wikipedia**

Original author(s): Linus Torvalds

Developer(s): Junio Hamano and others

Initial release: April 07, 2005

GitHub

GitHub, Inc. is provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features. **Wikipedia**

```
commit 4dc9c2bc11935c99577c2e5ad8be56df83f1534e (HEAD -> main, origin/main, origin/HEAD)
Author: Jae Yeon Kim <jaeyeonkim@berkeley.edu>
Date:   Tue Jan 26 09:37:00 2021 -0800

    update group csv

commit 01b2bc465f54282f6940e7c5386c92551cb2ef17
Author: Jae Yeon Kim <jaeyeonkim@berkeley.edu>
Date:   Mon Jan 25 12:53:54 2021 -0800

    corrected errors in the bash slides

commit 9a5040178b00f04f4786ea477ef00ebfcd77cf33
Merge: 301eba1 04a7047
Author: Jae Yeon Kim <jaeyeonkim@berkeley.edu>
Date:   Mon Jan 25 12:28:58 2021 -0800

    Add python binder`

    Merge branch 'main' of github.com:PS239T/spring_2021 into main

commit 301eba1252de54505dd578e7659f3a61d45deaf1
Author: Jae Yeon Kim <jaeyeonkim@berkeley.edu>
Date:   Mon Jan 25 12:28:53 2021 -0800

    add check_installation.sh

commit 04a7047f413de54e164feaa4a286b8e38339f4a1
Author: Jae Yeon Kim <44354133+jaeyk@users.noreply.github.com>
Date:   Sun Jan 24 16:50:46 2021 -0800
```

When & who : automatically documented

Why: explained by commit

If you already cloned the course repo, you can print the above output by typing the following.

```
cd spring_2021/ ; git log
```

Add function for random group assignment

main

Browse files

jaeyk committed 9 days ago

1 parent ea042f1 commit 4ebfe0ddf961929c97acfb1c13beb96ae762c44d

Showing 2 changed files with 29 additions and 0 deletions.

Unified Split

18 _logistics/group_assignment.r

```
@@ -0,0 +1,18 @@
+
+ # Load library
+ if (!require("pacman", quietly = TRUE)) { install.packages("pacman") }
+
+ pacman::p_load(here, tidyverse)
+
+ # Load file
+ df <- read.csv(here("_logistics", "names.csv"))
+ df <- df %>%
+   select(x) %>%
+   rename("name" = "x")
+
+ # Randomly assign students into two groups
+ #set.seed(1234)
+
+ df$group_id <- sample(rep(1:2, each = 5), 10, replace = FALSE)
+
+ df %>% arrange(group_id)
```

What: you can see this code change (+: green, -: red) from GitHub

Setup

Installation & Sign-up

1. Install Git (you can check by typing `git --version` in the terminal. You should see something like # git version x.xx.x
2. Sign up a GitHub account. (Also sign up for GitHub Student Developer Pack. Pro account for free!)
3. Access GitHub either using Hypertext Transfer Protocol Secure (HTTPS) or Secure Shell (SSH).

HTTPS

1. Create a personal access token: https://docs.github.com/en/github/authenticating-to-github/creating-a-personal-access-token
2. When you're communicating with GitHub, you need to authenticate. Use PAT in place of PW.

SSH

1. Strongly recommended: safer, and more convenient. But setting up might take more time.
2. Two keys (public and private): Public -> server (e.g., GitHub) and private -> client (e.g., your laptop). Only when the two are matched, the system unlocks.

Configurations

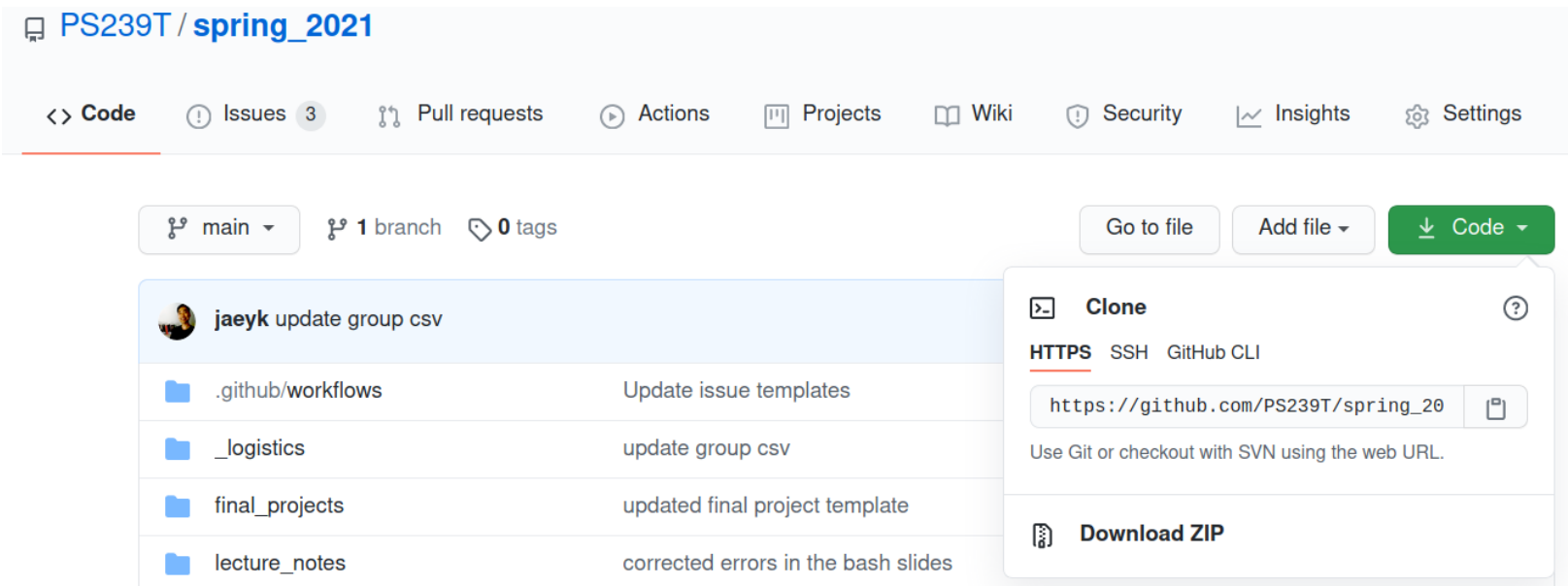
Method 1: Using the terminal

```

# User name and email
$ git config --global user.name "Firstname Lastname"
$ git config --global user.email username@school.extension
```

Method 2: Using RStudio (if you insist using R)

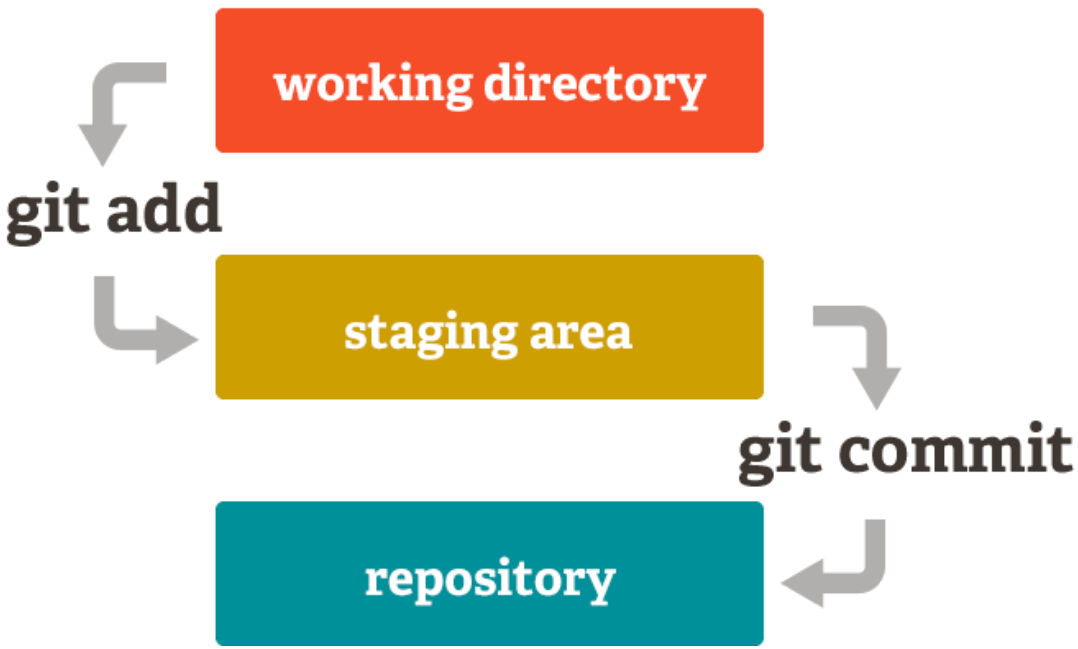
```

pacman::p_load(usethis)
use_git_config(user.name = "<Firstname Lastname>",
               user.email = "<username@school.extension>")
```

Let's learn
Git commands



Copy and paste the URL depending on your choice of authentication (HTTPS/SSH).

Workflow


working directory
git add
staging area
git commit
repository

Workflow

Clone a repo. Fork: server -> server. Clone: server -> client

```
1 git clone git@github.com:PS239T/spring_2021.git
```

Pull changes made by others. You can skip the [] part. If the current branch is correct (you can check by git checkout).

```
1 git pull [origin main] # used to be master
```

Make a change.

```
1 echo "something" > git_test
```

Commit and push.

```
1 git add git_test ; git commit -m "test commit" ;
  git push
```

Workflow

More on git add (staging)

```
# Stage specific file
git add <speicific file name>

# Stage all files
git add -A

# Stage updated file
git add -u

# Stage
git add .

# Not sure?
git add --help
```

Workflow

Also, you can make git automatically ignore a certain type of files.

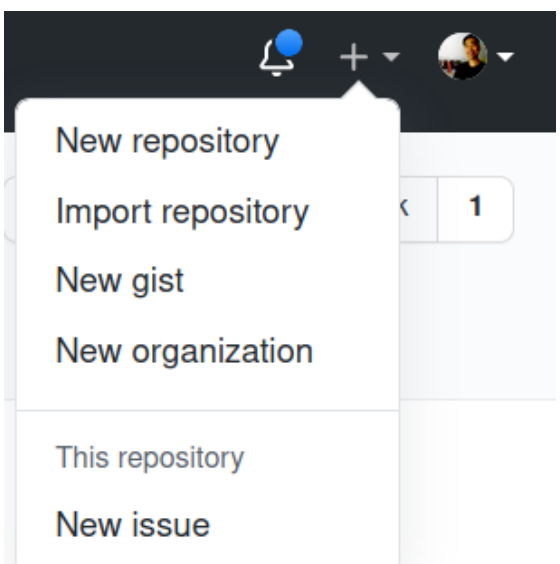
```
(base) jae@jae-X705UDR:~/hateasiancovid$ cat .gitignore
.Rproj.user
.Rhistory
.RData
.Ruserdata
*.csv
*.tsv
*.jsonl
*.rds
*.pdf
*.log
```

Workflow

Making a repo (command-line)

```
mkdir code_exercise
cd code_exercise

git init
```

Making a repo (GUI)




```
git clone /path/to/repository
```

Workflow

README.md is a default tool for documentation. I will talk more about this next week.

README.md

Large-scale Twitter Analysis on COVID-19 and Anti-Asian Climate

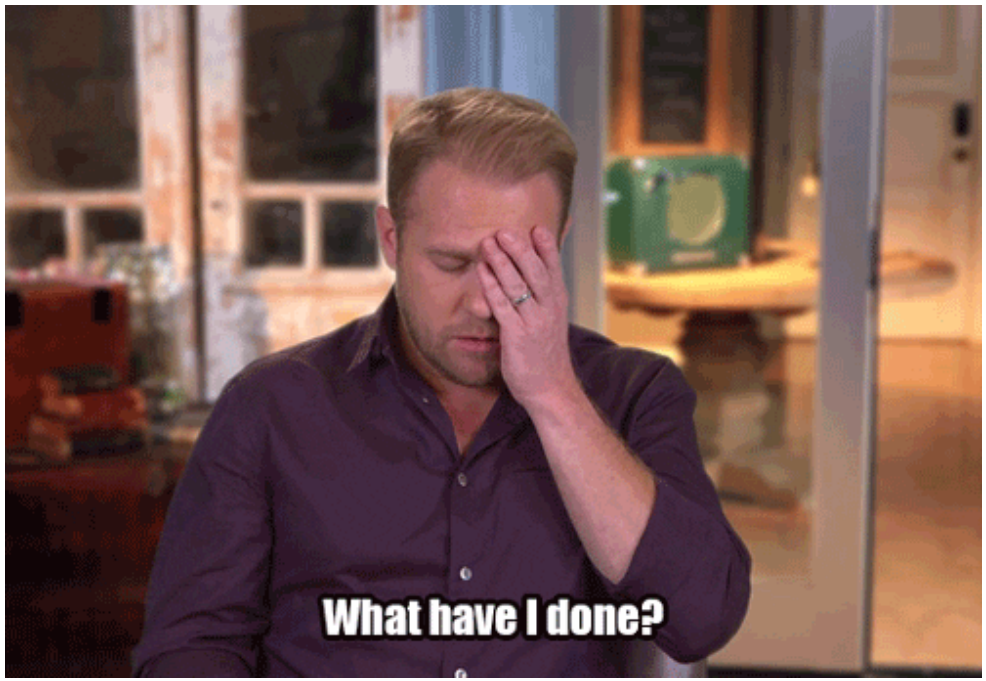
The preprint version of this project is available at https://osf.io/preprints/socarxiv/dvm7r/.

This analysis traces how COVID-19 has shaped an anti-Asian climate on Twitter, drawing on more than 1 million US-located tweets. The other part of the project is based on multi-racial survey data. This is a joint work with Nathan Chan (UCI) and Vivien Leung (UCLA). The paper version will be presented at the 2020 American Political Science Association annual meeting.

The objective of this article is to document the data collection, analysis, and visualization process for my future self, co-authors, and other researchers. In the research process, I also developed an R package called tidytweetjson, which could be useful to social scientists interested in using social media data for their own research. The entire research process is computationally reproducible. All the code used in the analysis is available in this Git repository. I automated parts that could be automatable by writing functions and putting some of these functions as a package.

I welcome any suggestions, comments, or questions. Please feel free to create issues in this Git repository or send an email to jaeyeonkim@berkeley.edu.

The Power of Time Machine



Going back to t-1

```
1 git reset --soft HEAD~1 # if you still want to keep the change, but you go
  back to t-1
2 git reset --hard HEAD~1 # if you're sure the change is unnecessary
```

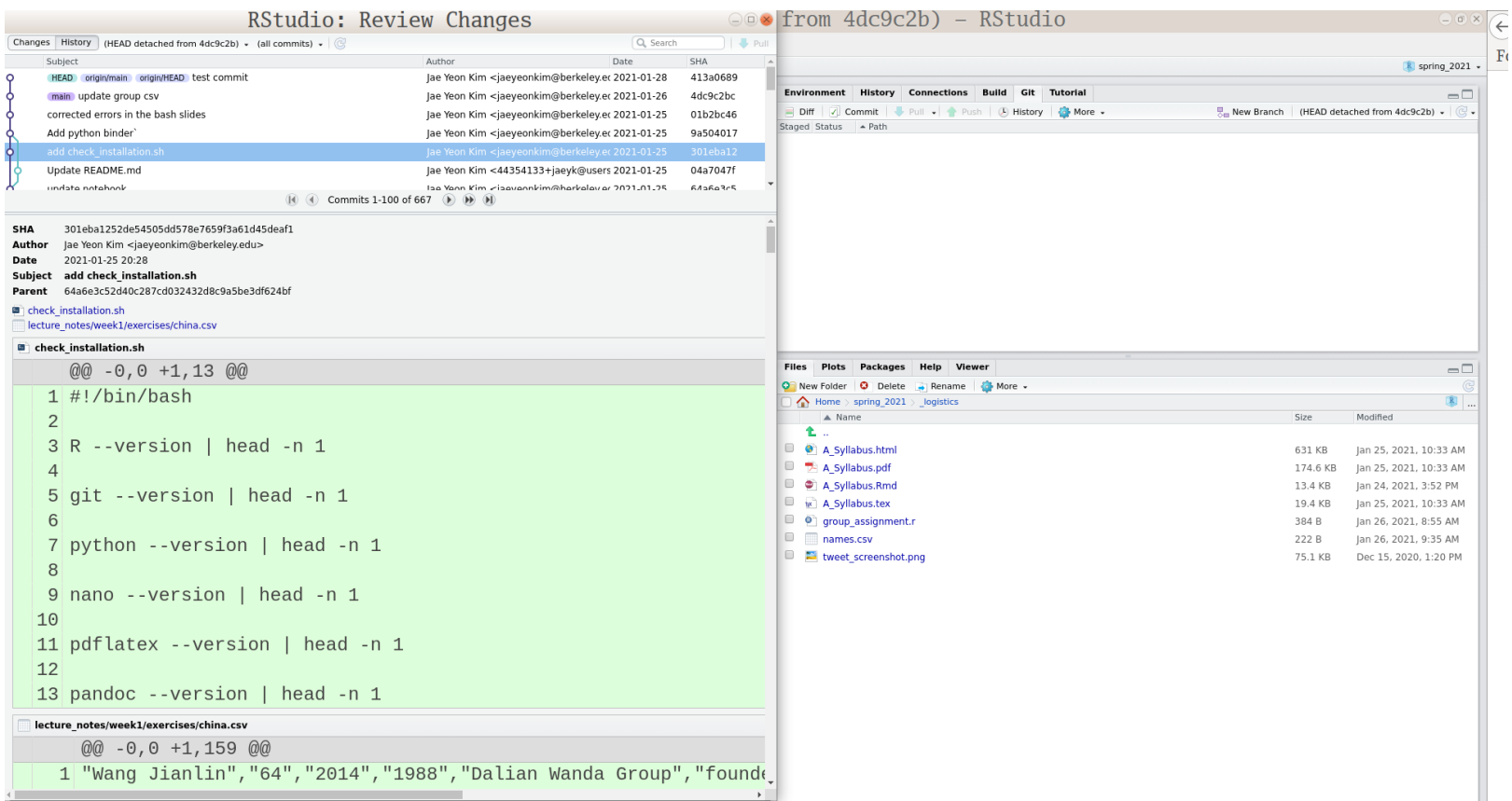
Overwriting local repo

```
# Download contnet from a remote repo
git fetch origin

# Going back to origin/master
git reset --hard origin/master

# Remove local files
git clean -f
```

Creating multiple time lines (called branching)

```
# Create a new branch
git checkout -b new_features
```

You can also use Git and GitHub in RStudio. I will talk more about this next week.



Other important stuff: issues, dashboards, pages, etc.

Demo: First assignment

GitHub Classroom

Automate your course and focus on teaching

Managing and organizing your class is easy with GitHub Classroom. Track and manage assignments in your dashboard, grade work automatically, and help students when they get stuck— all while using GitHub, the industry-standard tool developers use.

main | 1 branch | 0 tags | Go to file | Add file | Code | Use this template

jaeyk Update README.md 7132b5b 2 days ago 7 commits

README.md	Update README.md	2 days ago
check_installations.sh	Add shell script	2 months ago

README.md

Assignment 1 - Check setup

- Step 1. Install required softwares following this guideline: https://github.com/PS239T/spring_2021/blob/main/B_Install.md
- Step 2: Make the shell script executable by typing `chmod +x check_installations.sh`
- Step 3: Save the shell script output by typing `./check_installations.sh > output`
- Step 4: Check whether every required software was installed properly.
- Step 5: Add and commmit the change.

```
git add .

git commit -m "Check setup" # Free to change the commit message; but make it informative

git push
```

Questions or comments?