

The Joy of Version Control

Jae Yeon Kim

Objectives

"FINAL".doc



FINAL.doc!



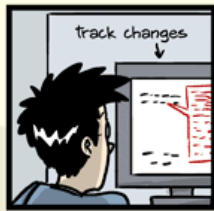
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.##\$%WHYDID
ICOMETOGRADSCHOOL?????.doc



JORGE CHAM © 2012

- 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

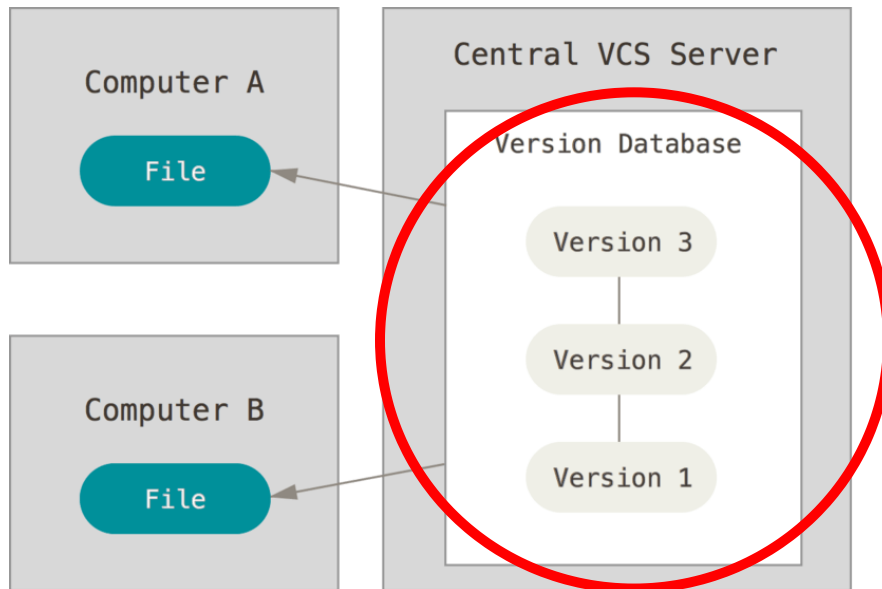
1. Which changes were made?
2. Who made the changes?
3. When were the changes made?
4. Why were changes needed?

**What if you're
working in a team?**

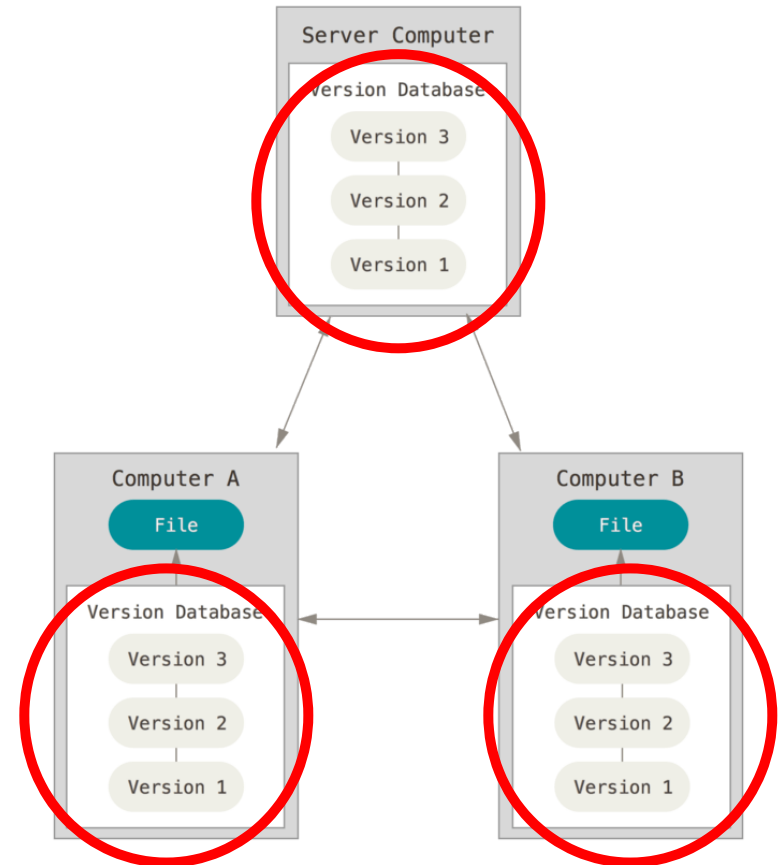
Solution:

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.

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

Add function for random group assignment

main

jaeyk committed 9 days ago

1 parent ea042f1 commit 4ebfe0ddf961929c97acfb1c13beb96ae762c44d

Showing 2 changed files with 29 additions and 0 deletions.

18

_logistics/group_assignment.r

...

...

@@ -0,0 +1,18 @@

1 +

2 + # Load library

3 + if (!require("pacman", quietly = TRUE)) { install.packages("pacman") }

4 +

5 + pacman::p_load(here, tidyverse)

6 +

7 + # Load file

8 + df <- read.csv(here("_logistics", "names.csv"))

9 + df <- df %>%

10 + select(x) %>%

11 + rename("name" = "x")

12 +

13 + # Randomly assign students into two groups

14 + #set.seed(1234)

15 +

16 + df\$group_id <- sample(rep(1:2, each = 5), 10, replace = FALSE)

17 +

18 + df %>% arrange(group_id)

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

11

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

```
1
2 # User name and email
3 $ git config --global user.name "Firstname Lastname"
4 $ git config --global user.email username@school.extension
```

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

```
1
2 pacman::p_load(usethis)
3 use_git_config(user.name = "<Firstname Lastname>",
4               user.email = "<username@school.extension>")
```



Let's learn Git commands

PS239T / **spring_2021**


<> **Code** ! Issues 3 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⚙ Settings

🔗 main ▾ 🔗 1 branch 🏷 0 tags


Go to file Add file ▾ **Code** ▾

 **jaeyk** update group csv


📁 .github/workflows	Update issue templates
📁 _logistics	update group csv
📁 final_projects	updated final project template
📁 lecture_notes	corrected errors in the bash slides

 **Clone** ?

HTTPS SSH GitHub CLI

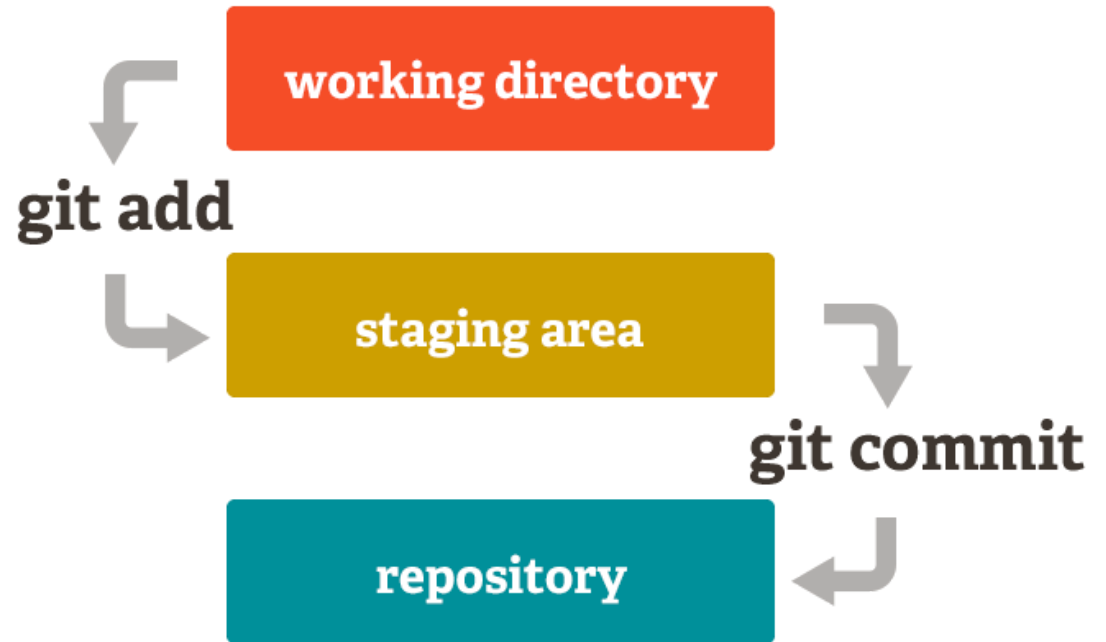


Use Git or checkout with SVN using the web URL.

 **Download ZIP**

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

Workflow



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)

```
1 # Stage specific file
2 git add <specific file name>
3
4 # Stage all files
5 git add -A
6
7 # Stage updated file
8 git add -u
9
10 # Stage
11 git add .
12
13 # Not sure?
14 git add --help
```

Workflow

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

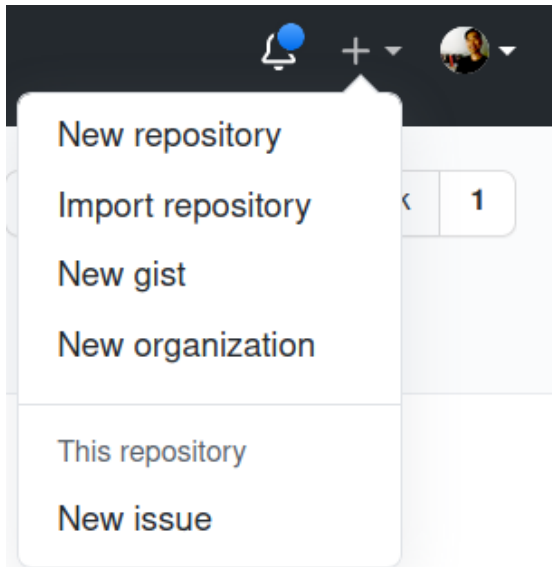
```
(base) jae@jae-X705UDR:~/hateasiancovid$ cat .gitignore
.Rproj.user
.Rhistory
.RData
.Ruserdata
*.csv
*.tsv
*.jsonl
*.rds
*.pdf
*.log
```

Workflow

Making a repo (command-line)

```
1 mkdir code_exercise
2 cd code_exercise
3
4 git init
```

Making a repo (GUI)



```
1 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

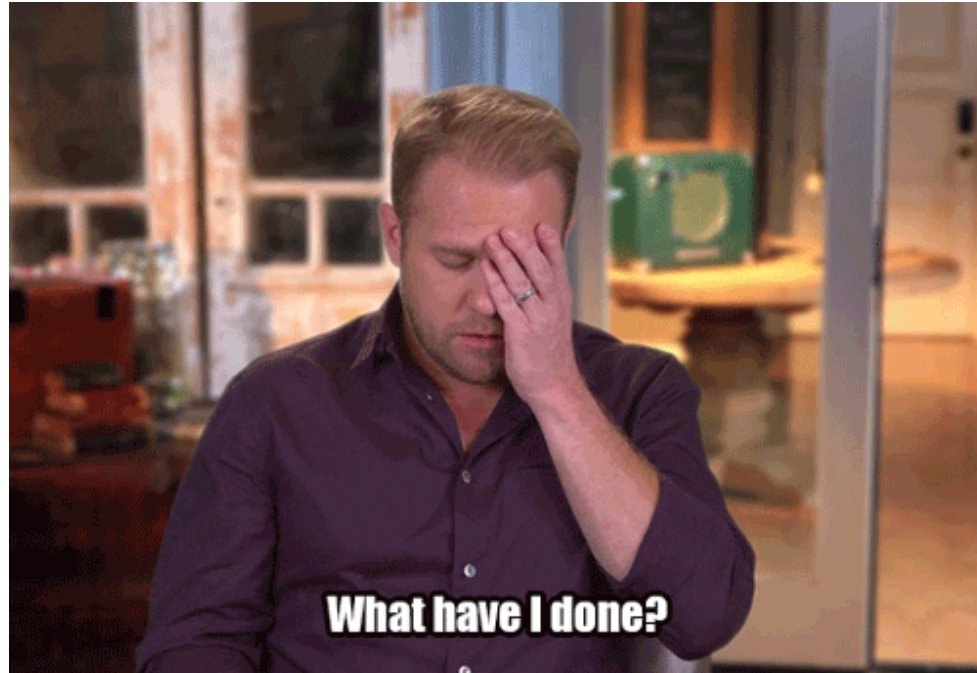
The preprint version of this project is available at <https://osf.io/preprints/socarxiv/dvm7r/>.

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

```
1 # Download contnet from a remote repo
2 git fetch origin
3
4 # Going back to origin/master
5 git reset --hard origin/master
6
7 # Remove local files
8 git clean -f
```

Creating multiple time lines (called branching)

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

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

The screenshot shows the RStudio interface with Git integration. The left pane is titled "Review Changes" and shows a list of commits. The selected commit is "add check_installation.sh" by Jae Yeon Kim, with SHA 301eba12. Below the commit list, the details for this commit are shown, including the parent commit (64a6e3c52d40c287cd032432d8c9a5be3df624bf) and the files changed (check_installation.sh, lecture_notes/week1/exercises/china.csv). The file content for check_installation.sh is displayed in a code editor, showing a script that checks the version of R, git, python, nano, pdflatex, and pandoc. The right pane shows the "Environment" and "Files" panes. The "Files" pane lists the files in the project, including A_Syllabus.html, A_Syllabus.pdf, A_Syllabus.Rmd, A_Syllabus.tex, group_assignment.r, names.csv, and tweet_screenshot.png.

Review Changes (HEAD detached from 4dc9c2b) - RStudio

Subject	Author	Date	SHA
test commit	Jae Yeon Kim <jaeyeonkim@berkeley.ec>	2021-01-28	413a0689
update group csv	Jae Yeon Kim <jaeyeonkim@berkeley.ec>	2021-01-26	4dc9c2bc
corrected errors in the bash slides	Jae Yeon Kim <jaeyeonkim@berkeley.ec>	2021-01-25	01b2bc46
Add python binder	Jae Yeon Kim <jaeyeonkim@berkeley.ec>	2021-01-25	9a504017
add check_installation.sh	Jae Yeon Kim <jaeyeonkim@berkeley.ec>	2021-01-25	301eba12
Update README.md	Jae Yeon Kim <44354133+jaeyk@users>	2021-01-25	04a7047f
update notebook	Jae Yeon Kim <jaeyeonkim@berkeley.ec>	2021-01-25	64a6e3c5

SHA 301eba1252de54505dd578e7659f3a61d45dea1
Author Jae Yeon Kim <jaeyeonkim@berkeley.edu>
Date 2021-01-25 20:28
Subject add check_installation.sh
Parent 64a6e3c52d40c287cd032432d8c9a5be3df624bf

check_installation.sh
lecture_notes/week1/exercises/china.csv

check_installation.sh

```
@@ -0,0 +1,13 @@  
1 #!/bin/bash  
2  
3 R --version | head -n 1  
4  
5 git --version | head -n 1  
6  
7 python --version | head -n 1  
8  
9 nano --version | head -n 1  
10  
11 pdflatex --version | head -n 1  
12  
13 pandoc --version | head -n 1
```

lecture_notes/week1/exercises/china.csv

```
@@ -0,0 +1,159 @@  
1 "Wang Jianlin", "64", "2014", "1988", "Dalian Wanda Group", "founde
```

Environment History Connections Build Git Tutorial

Diff Commit Pull Push History More

New Branch (HEAD detached from 4dc9c2b)

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home spring_2021 _logistics

Name	Size	Modified
A_Syllabus.html	631 KB	Jan 25, 2021, 10:33 AM
A_Syllabus.pdf	174.6 KB	Jan 25, 2021, 10:33 AM
A_Syllabus.Rmd	13.4 KB	Jan 24, 2021, 3:52 PM
A_Syllabus.tex	19.4 KB	Jan 25, 2021, 10:33 AM
group_assignment.r	384 B	Jan 26, 2021, 8:55 AM
names.csv	222 B	Jan 26, 2021, 9:35 AM
tweet_screenshot.png	75.1 KB	Dec 15, 2020, 1:20 PM

**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.

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'main', '1 branch', and '0 tags'. To the right are buttons for 'Go to file', 'Add file', 'Code', and 'Use this template'. Below this is a commit history table with two entries: 'jaeyk Update README.md' (2 days ago, 7 commits) and 'Update README.md' (2 days ago). Below the table is a file list with 'README.md' (Update README.md, 2 days ago) and 'check_installations.sh' (Add shell script, 2 months ago). The 'README.md' file is open, showing the title 'Assignment 1 - Check setup' and a list of five steps for setting up the assignment. The steps are: 1. Install required softwares following this guideline: https://github.com/PS239T/spring_2021/blob/main/B_Install.md; 2. Make the shell script executable by typing `chmod +x check_installations.sh`; 3. Save the shell script output by typing `./check_installations.sh > output`; 4. Check whether every required software was installed properly; 5. Add and commit the change. Below the steps is a code block with the following commands: `git add .`, `git commit -m "Check setup" # Free to change the commit message; but make it informative`, and `git push`.

Commit	Author	Message	Time
7132b5b	jaeyk	Update README.md	2 days ago
			7 commits

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

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 commit the change.

```
git add .

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

git push
```

Questions or comments?