

Version control, &c.

Einar Holsbø 251120

**We'll learn how to
track changes to your own code, make it available to others**

We'll learn how to track changes to your own code, make it available to others

- Create a **repository** (repo) for some code

We'll learn how to track changes to your own code, make it available to others

- Create a **repository** (repo) for some code
- Track code and its changes within your projects

We'll learn how to track changes to your own code, make it available to others

- Create a **repository** (repo) for some code
- Track code and its changes within your projects
- Publish a repo on GitHub

We'll learn how to track changes to your own code, make it available to others

- Create a **repository** (repo) for some code
- Track code and its changes within your projects
- Publish a repo on GitHub
- Work with repos on GitHub

We won't learn how to
(ie. things I'd like you to know at some point)

We won't learn how to

(ie. things I'd like you to know at some point)

- Collaborate on repositories with branches, pull requests, etc.

We won't learn how to

(ie. things I'd like you to know at some point)

- Collaborate on repositories with branches, pull requests, etc.
- Provide proper documentation

We won't learn how to

(ie. things I'd like you to know at some point)

- Collaborate on repositories with branches, pull requests, etc.
- Provide proper documentation
- Structure code by use of functions

We won't learn how to

(ie. things I'd like you to know at some point)

- Collaborate on repositories with branches, pull requests, etc.
- Provide proper documentation
- Structure code by use of functions
- Collect and share useful functions by packaging them

Prelude

Why it is impossible to get anything done in NOWAC

There are R scripts floating around but

Why it is impossible to get anything done in NOWAC

There are R scripts floating around but

- Multiple versions of same script

Why it is impossible to get anything done in NOWAC

There are R scripts floating around but

- Multiple versions of same script
- No way to identify author from the script

Why it is impossible to get anything done in NOWAC

There are R scripts floating around but

- Multiple versions of same script
- No way to identify author from the script
- Several authors adding to same script

Why it is impossible to get anything done in NOWAC

There are R scripts floating around but

- Multiple versions of same script
- No way to identify author from the script
- Several authors adding to same script
- No timestamps

Why it is impossible to get anything done in NOWAC

To figure out what is going on you can

Why it is impossible to get anything done in NOWAC

To figure out what is going on you can

- Ask around

Why it is impossible to get anything done in NOWAC

To figure out what is going on you can

- Ask around
- Make guesses

Why it is impossible to get anything done in NOWAC

To figure out what is going on you can

- Ask around
- Make guesses
- Find out that the Key Person Who Might Know is on leave

Why it is impossible to get anything done in NOWAC

To figure out what is going on you can

- Ask around
- Make guesses
- Find out that the Key Person Who Might Know is on leave
- Find out that the KPWMK is now working elsewhere

Lack of **provenance**, ie., who did
what when

Lack of **documentation**, ie., why
did they do it

I concluded that it would probably take less time for me to write my own scripts, despite that I have to learn R first.

Kajsa Møllersen, NOWAC mailing list several years ago

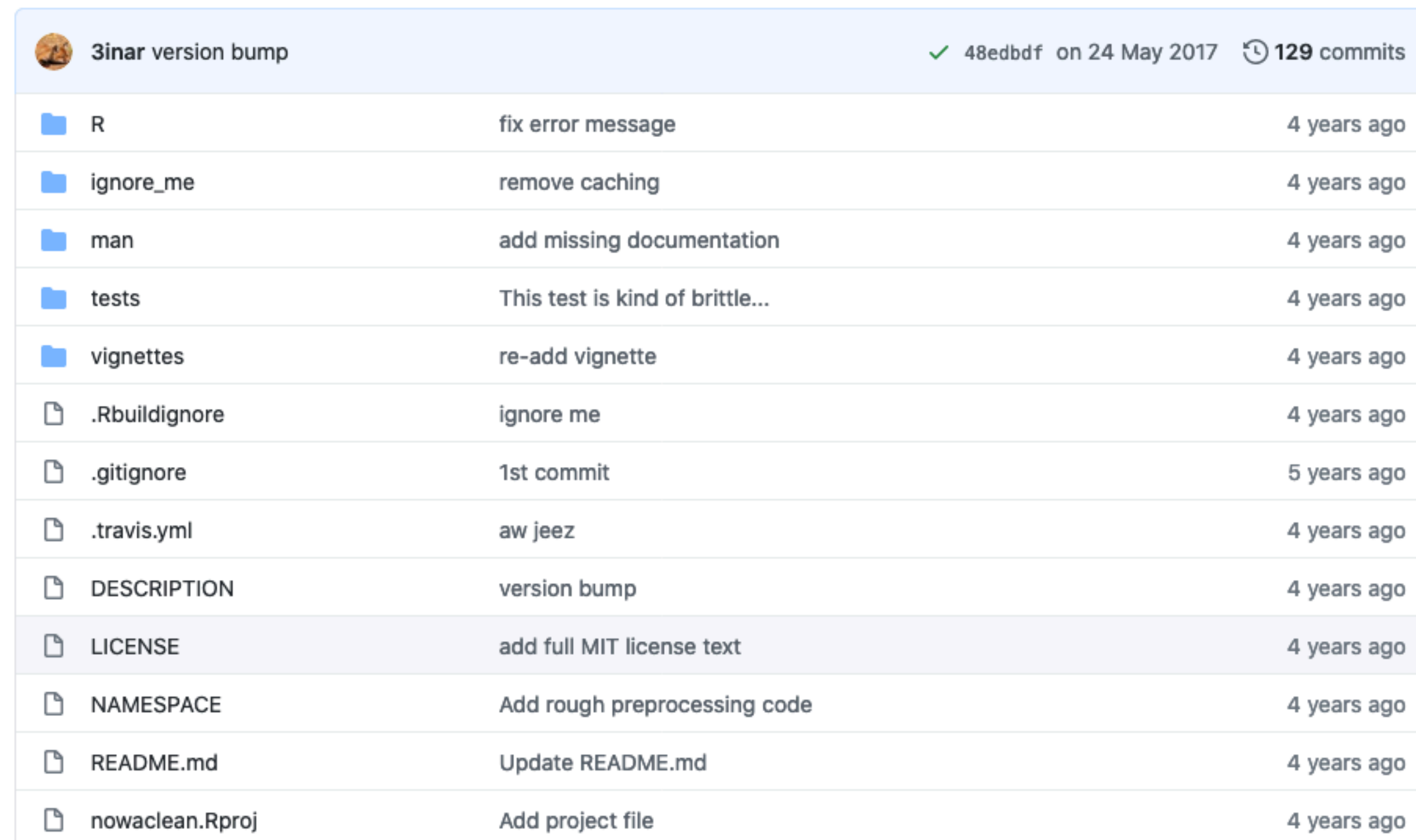
Act I: The repo

A repo is a folder with extra dressing

Solves the **provenance** problem + partly the **documentation** problem

A repo is a folder with extra dressing

Solves the **provenance** problem + partly the **documentation** problem



The screenshot shows a commit history table for a repository. The header row includes the repository name '3inar version bump', a green checkmark, the commit hash '48edbfd', the date 'on 24 May 2017', and '129 commits'. The table lists various files and folders with their respective commit messages and dates.

File/Folder	Commit Message	Time
R	fix error message	4 years ago
ignore_me	remove caching	4 years ago
man	add missing documentation	4 years ago
tests	This test is kind of brittle...	4 years ago
vignettes	re-add vignette	4 years ago
.Rbuildignore	ignore me	4 years ago
.gitignore	1st commit	5 years ago
.travis.yml	aw jeez	4 years ago
DESCRIPTION	version bump	4 years ago
LICENSE	add full MIT license text	4 years ago
NAMESPACE	Add rough preprocessing code	4 years ago
README.md	Update README.md	4 years ago
nowaclean.Rproj	Add project file	4 years ago

File structure
like a folder

A repo is a folder with extra dressing

Solves the **provenance** problem + partly the **documentation** problem

Complete record of
all changes

The screenshot displays a vertical list of commits from a GitHub repository, organized by date. Each commit entry includes a title, the author's name, the commit date, a commit hash, and a link to view the commit details. The commits are as follows:

- Commits on May 24, 2017**
 - version bump (3inar committed on 24 May 2017) - Hash: 48edbfd
 - fix error message (3inar committed on 24 May 2017) - Hash: 2e97a69
- Commits on May 21, 2017**
 - remove caching (3inar committed on 21 May 2017) - Hash: abf1758
- Commits on May 20, 2017**
 - update gitignore (3inar committed on 20 May 2017) - Hash: d280768
 - re-add vignette (3inar committed on 20 May 2017) - Hash: 9871001
 - remove old stuff (3inar committed on 20 May 2017) - Hash: dc41af0
 - update (3inar committed on 20 May 2017) - Hash: eea0069
 - Merge branch 'master' of github.com:3inar/nowaclean (3inar committed on 20 May 2017) - Hash: 2a685ac
- Commits on May 18, 2017**
 - needed some different plots back when (3inar committed on 18 May 2017) - Hash: 3d458f0

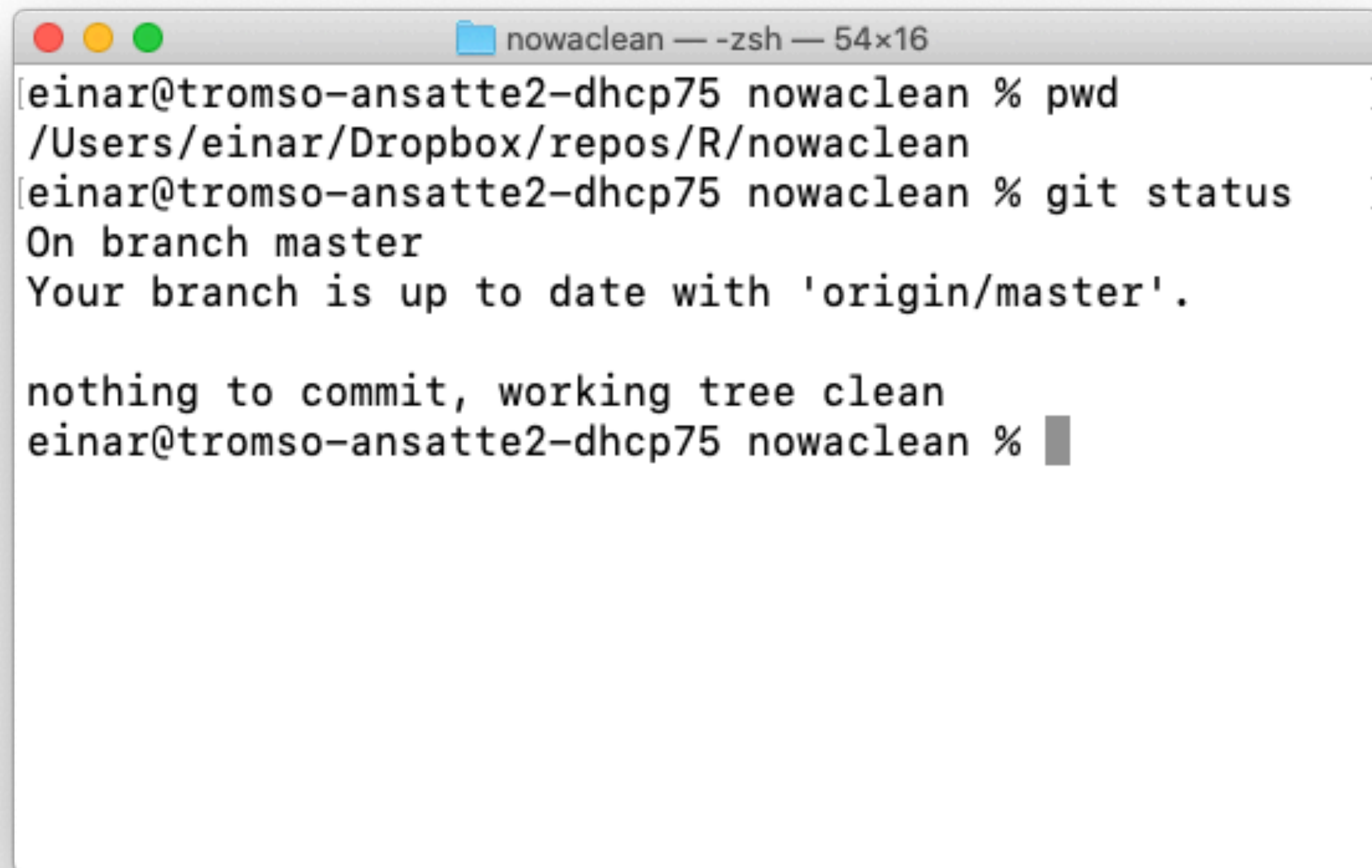
For our purpose done by a program called Git that notices when something changes in the repo

**(note to self: look at the
nowaclean repo here)**

Some ways to use Git and make the repo go

Some ways to use Git and make the repo go

Terminal window

A screenshot of a terminal window titled "nowaclean — -zsh — 54x16". The terminal shows the following commands and output:

```
einar@tromso-ansatte2-dhcp75 nowaclean % pwd
/Users/einar/Dropbox/repos/R/nowaclean
einar@tromso-ansatte2-dhcp75 nowaclean % git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
einar@tromso-ansatte2-dhcp75 nowaclean % █
```

Esoteric but powerful

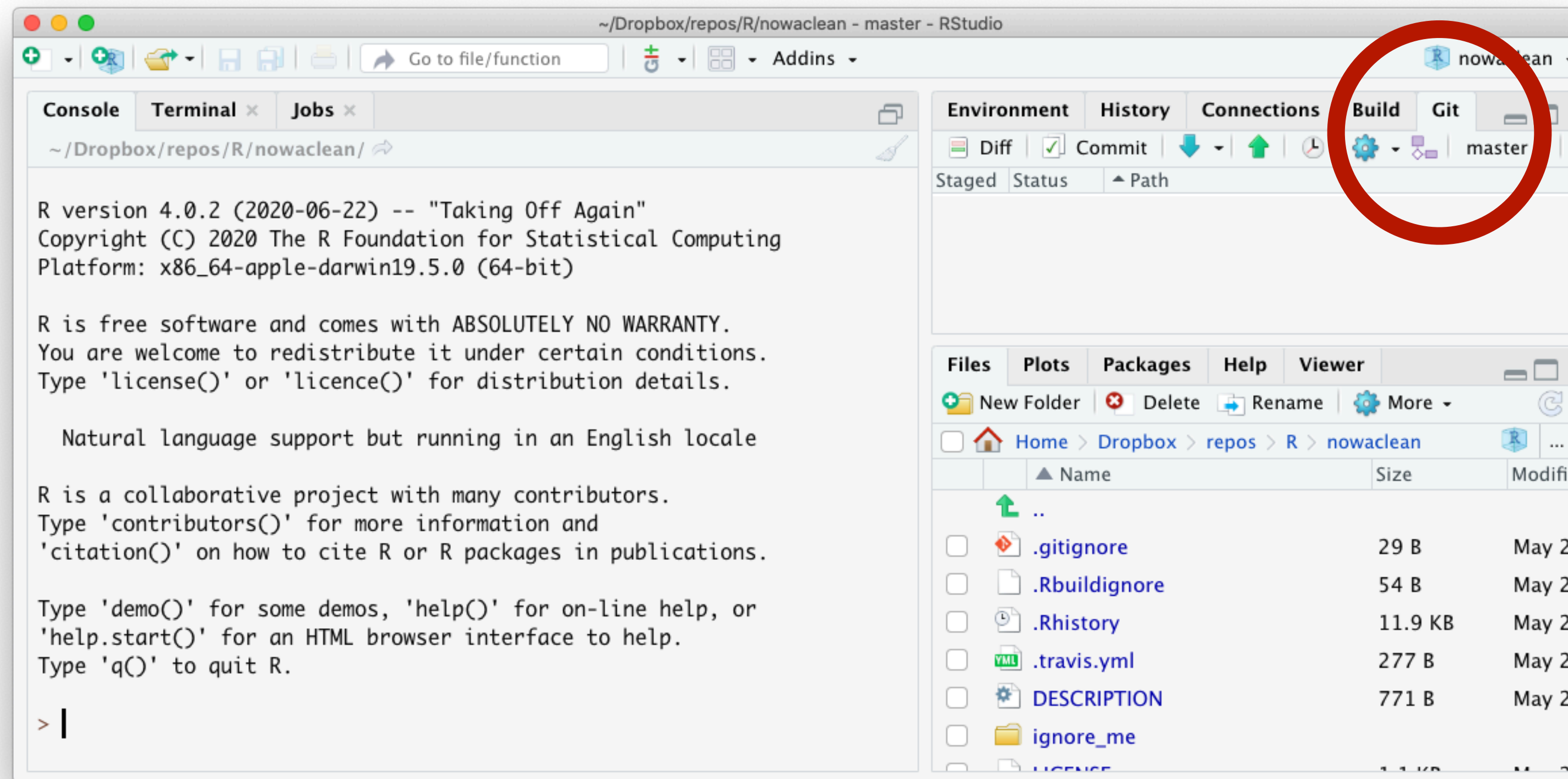
Some ways to use Git and make the repo go

GitHub app

Very nice actually



Some ways to use Git and make the repo go RStudio builtin



**Not great but useful
because inside RStudio**

Practical: creating a new repo

(note to self: short break?)

Act II: Tracking code changes

Basic workflow:

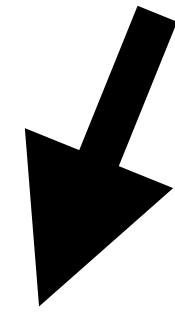
- Make a moderate, self-contained piece of work (make sure it works!!!!!!!)
- **Stage changes**
- **Commit changes** with short description of what is going on

Practical: Our first few commits

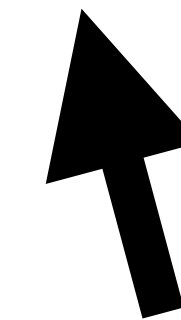
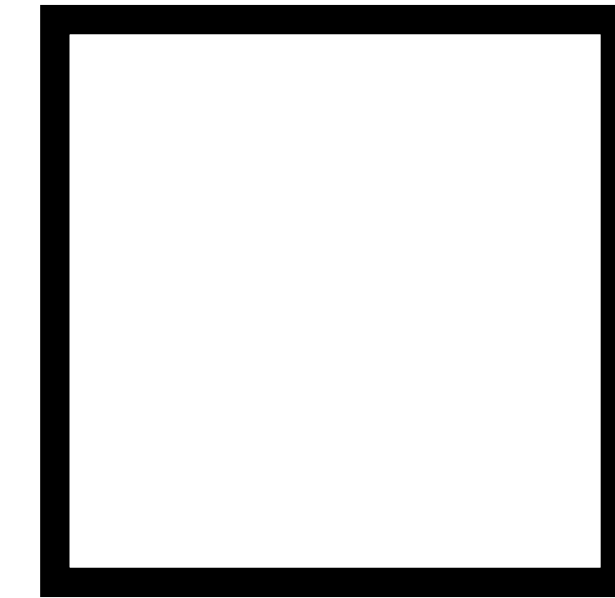
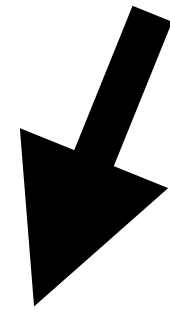
Act III: Going online

**First we have to connect
RStudio & GitHub**

a file on your computer

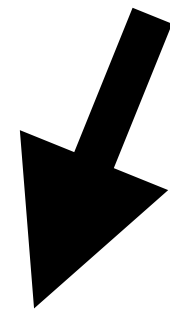


a file on your computer

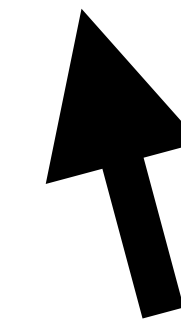
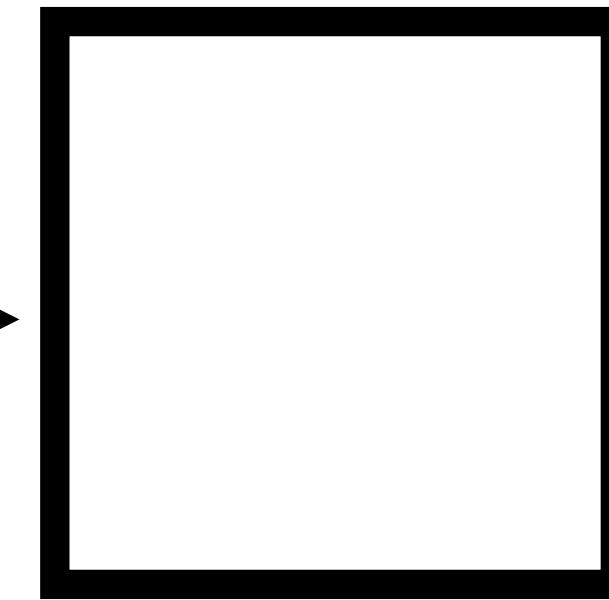


another
computer (remote)

a file on your computer

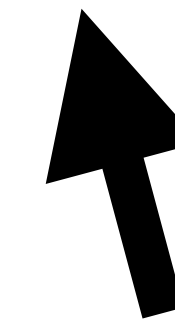
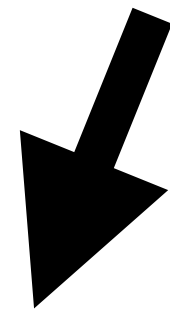


git push



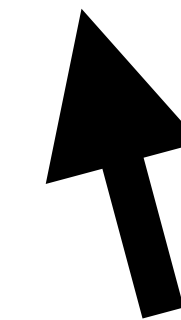
another
computer (remote)

a file on your computer

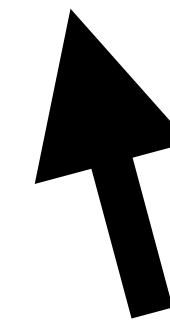


same file on another
computer (remote)

change + `commit`

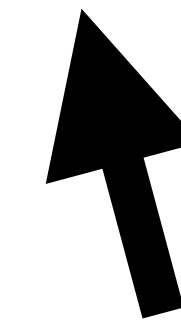


same file on another
computer (remote)

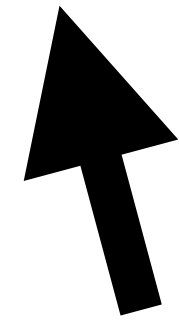


same file on another
computer (remote)

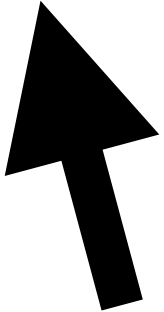
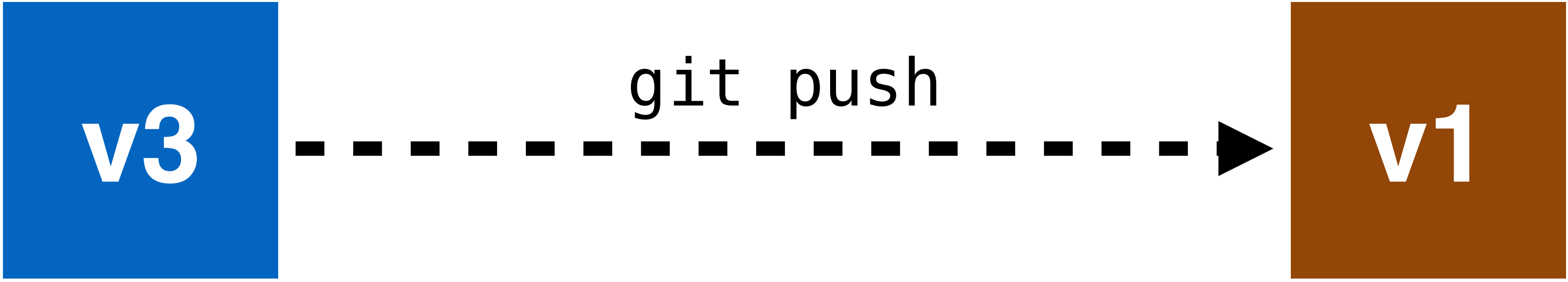
change + `commit`



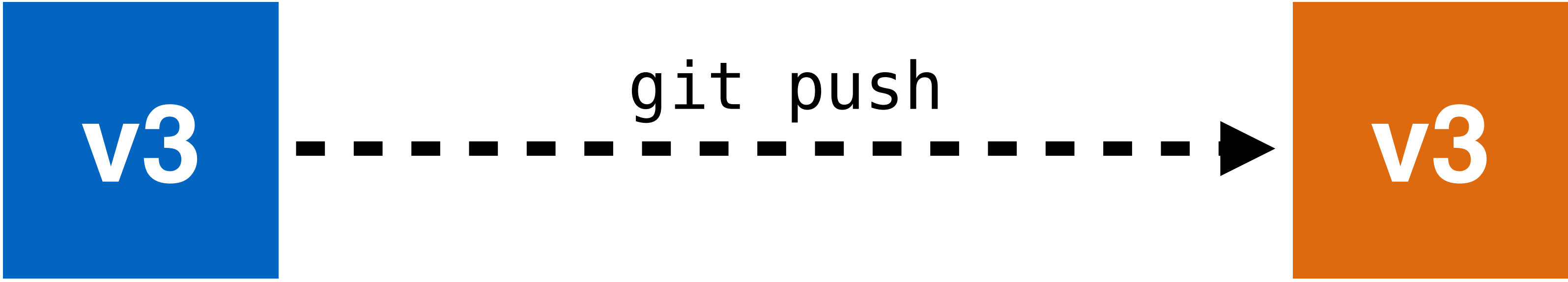
same file on another
computer (remote)



same file on another
computer (remote)



same file on another
computer (remote)



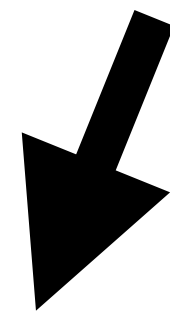


Remote is up-to-date

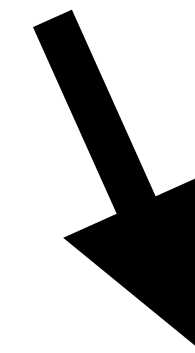


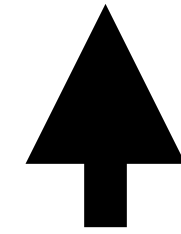
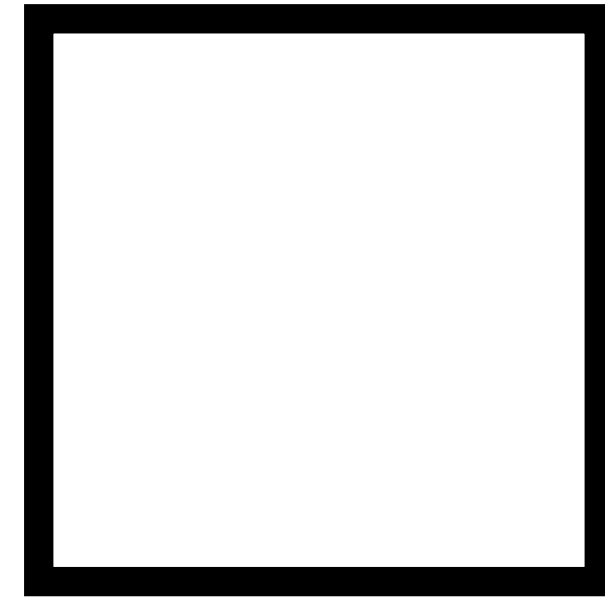
Remote is up-to-date
Also has full version history

a file on your computer



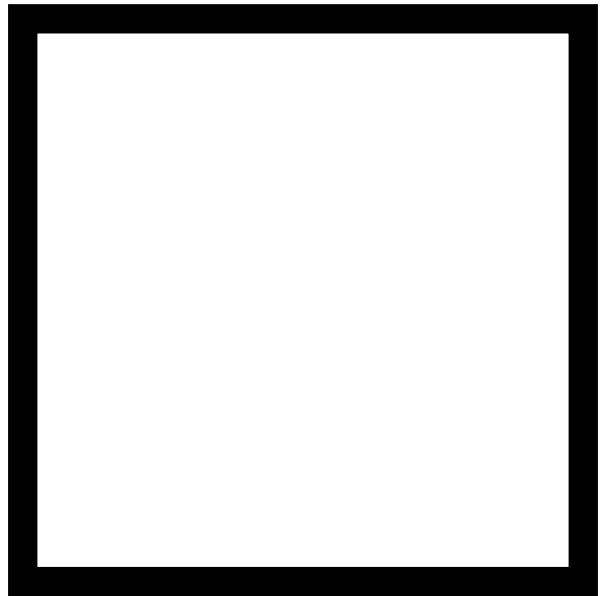
same file on another
computer (remote)





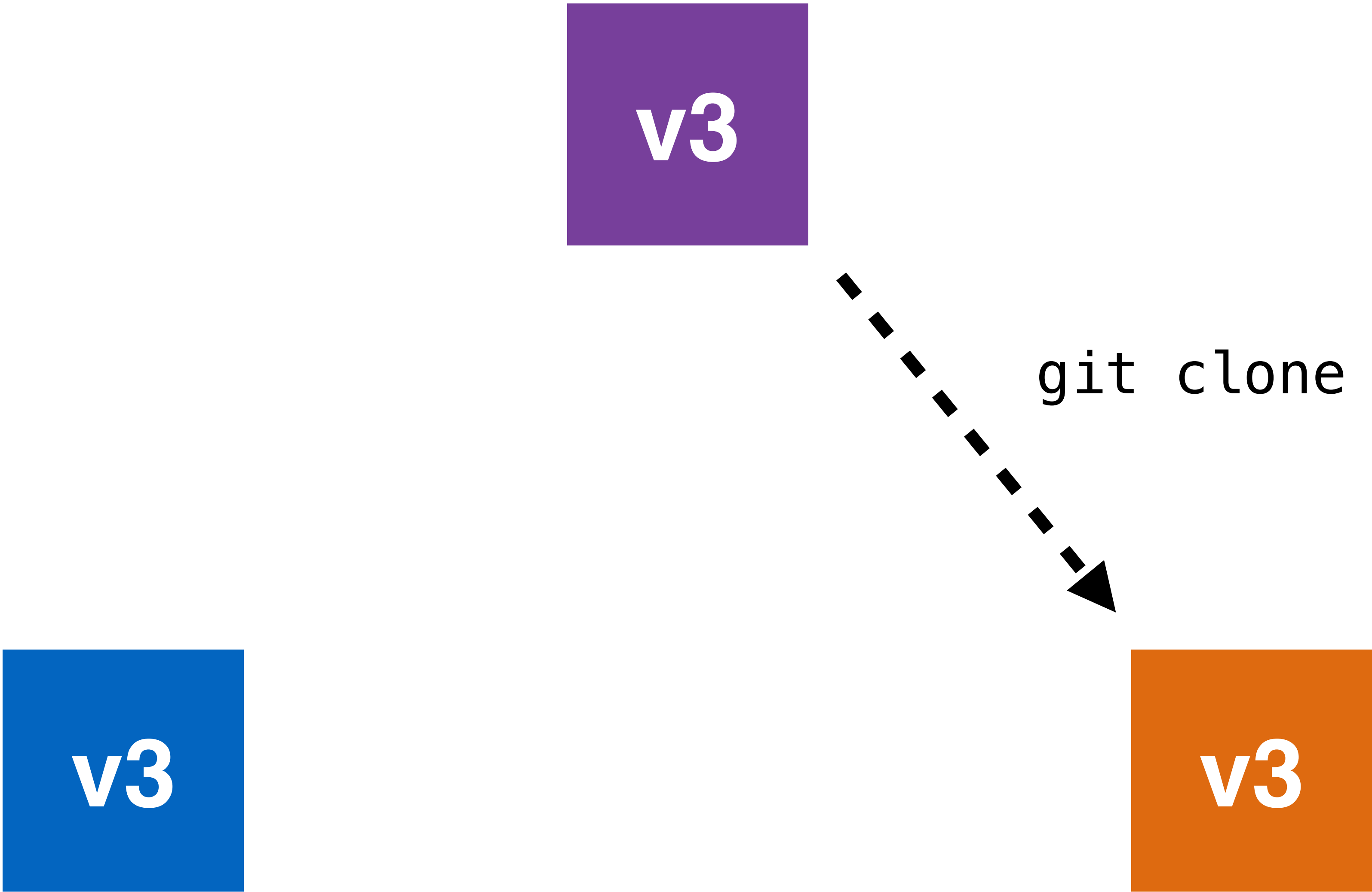
my computer doesn't have this project





git clone





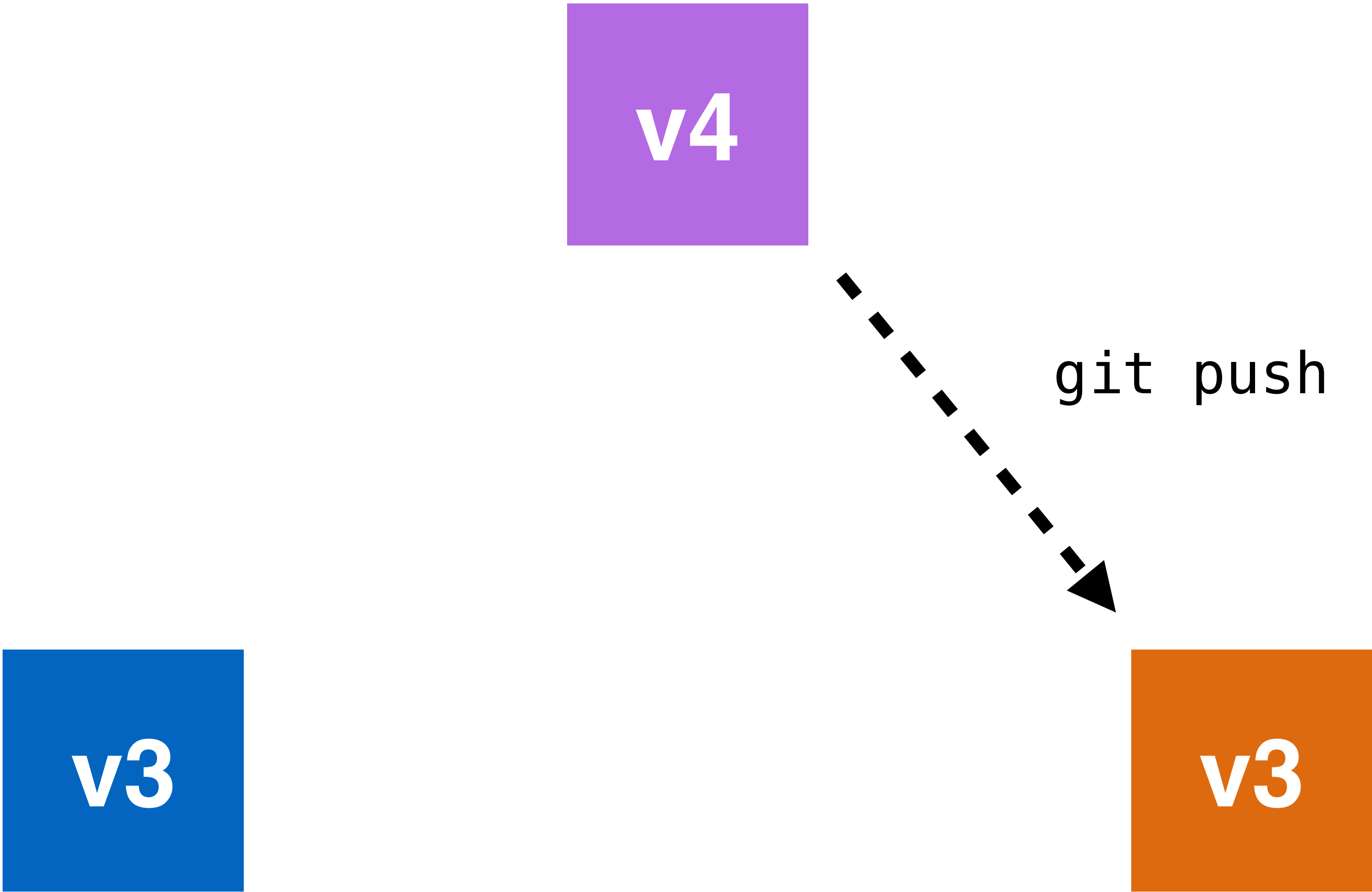
change + commit



v4

v3

v3



v4

v3

v4



git pull





git pull





REMOTE = SYNCHRONIZATION PT.



Practical: two ways of setting up a remote repo on GitHub—

- (i) start on your computer**
- (ii) start on GitHub**

Basic workflow:

- During the day you make some number of commits locally
- At the end of the day **push** your changes to remote (GitHub)

**Practical: some commits,
some pushing**

Epilogue: Three things

(i) Right after this I want you to make a repo, for a real thing you're working on. Publish it on GitHub, and use the repo from now on.

(ii) I will make these slides available and send some additional resources to you

(iii) Contact me if there's anything. Also let me know if you would like another session that goes deeper

(iv) thank you