Version control with Git and GitHub

Matthew Wigginton Conway
School of Geographical Sciences and Urban Planning
Arizona State University

May 1, 2020

School of Geographical Sciences and Urban Planning
Arizona State University

Slide download

`https://files.indicatrix.org/git.pdf`

What is version control?

- Track changes to your code
- Know *when* and *who* made changes to code
- Revert to old versions of your code
- Archive specific versions of your code (e.g. after publication)
- `git` is the most popular version control system, and it's free

`git` vs. GitHub

`git`

- Software that runs on your computer
- Used to manage your code locally
- Runs from the command line

GitHub

- Website for hosting software stored in `git`
- Allows collaboration with other on `git`-based projects
- Allows *public* or *private* code storage

`git` Concepts

- Repository: storage for code for a single project
- Commit: Set of changes to the code
- Branch: Version of the code you can try something in before finalizing it
- Tag: Pointer to a specific version of your code (e.g. for publication)
- Working tree: The copy of your code on your system, which you may have modified since the last commit

What should I put in version control?

- Yes
  - Code
  - Text-based documentation (TeX, Markdown, etc.)
- Maybe
  - Binary documents (Word, Excel, etc.)
- No
  - Large files (larger than a few MB)
  - Data files (shouldn’t be changing often)

Collaboration

- Don’t share your `git` repository with others via Dropbox, etc.
- Repository can become corrupted if modified simultaneously
- Instead, use a service like GitHub

Live demo

Download and unzip:

`https://files.indicatrix.org/git-demo.zip`

Open a terminal

- Mac: Open Terminal application, type `cd `, drag folder into terminal, press enter
- Windows: Right-click in folder, choose Git Bash Here
- Linux: Open a terminal, `cd ~/where/you/unzipped/folder`

`git` Setup

- `git config --global user.name "Your Name"`
- `git config --global user.email "Your Email"`
  - If you push to a public repository on GitHub, your email will be public
  - If that concerns you, skip setting email for now
  - Later, follow: How to keep your email private on GitHub

Creating a new `git` repository

- `git init`
- `git add README.txt demo.R`
  - Can use `git add` more than once
- `git commit -m "Commit message"`
  - Quotes are important!

Make some changes to the files, create new files, etc.

Seeing what’s changed since the last commit

- `git status`

```
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified: README.txt
    deleted: demo.R

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    demo2.R

no changes added to commit (use "git add" and/or "git commit -a")
```

Seeing what's changed since the last commit

- `git status`

```
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified: README.txt
    deleted: demo.R

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    demo2.R

no changes added to commit (use "git add" and/or "git commit -a")
```

Modified

Seeing what’s changed since the last commit

- `git status`

```
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified: README.txt
    deleted: demo.R

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    demo2.R

no changes added to commit (use "git add" and/or "git commit -a")
```

Modified

Deleted

Seeing what’s changed since the last commit

- `git status`

```
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified: README.txt
    deleted: demo.R

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    demo2.R

no changes added to commit (use "git add" and/or "git commit -a")
```

Modified

Deleted

New

Committing your changes

- `git add` for modified or new files
- `git rm` for deleted files
- `git commit -m "commit message"` to commit

Getting rid of your changes before you commit

- `git restore filename` for a single file
  - `git checkout filename` in older versions of `git`
- NO UNDO

Getting rid of your changes before you commit

- `git restore filename` for a single file
  - `git checkout filename` in older versions of `git`
- NO UNDO
- `git stash` for your entire working tree (except new files)
- `git stash pop` to undo

Creating a branch

- `git switch -c branch-name`
    - `git checkout -b branch-name` on older versions of `git`

Working on a branch

- Change some files
- Commit them

Switching back to your main branch

- Main branch is `master`
- To get back to it: `git switch master`
  - `git checkout master` on old versions
- Commit or restore everything before switching branches
  - Any changes not committed or restored will be retained when switching branches

Merging changes from a branch into `master`

- `git merge branch-name`
- If files have changed in both branches, `git` will try to combine the changes
- Always check that results make sense after a merge
- If the merge process doesn't work: how to resolve a merge conflict

Tagging a commit

-
  ```
  git tag tag-name
  ```
- Make sure you have no uncommitted changes

Getting back to an old tag

- `git tag` to list tags
- `git checkout <tag-name>` to restore that tag
- Check for changes using `git status` beforehand
- This *does not* change the master branch

Getting back to an old commit

- `git log`

```
commit 64660da5ee23476e82b9c199bee6efdcb91d7969
Author: Matthew Wigginton Conway <matt@indicatrix.org>
Date: Sat Dec 28 13:43:13 2019 -0500

adjust convergence criteria to be less stringent.

commit 169a33d0860d4a7914663f01794b4af0b6e6a82d
Author: Matthew Wigginton Conway <matt@indicatrix.org>
Date: Wed Nov 27 15:21:52 2019 -0800

    don't import bayer by default, remove dep on linearmodels.
```

Getting back to an old commit

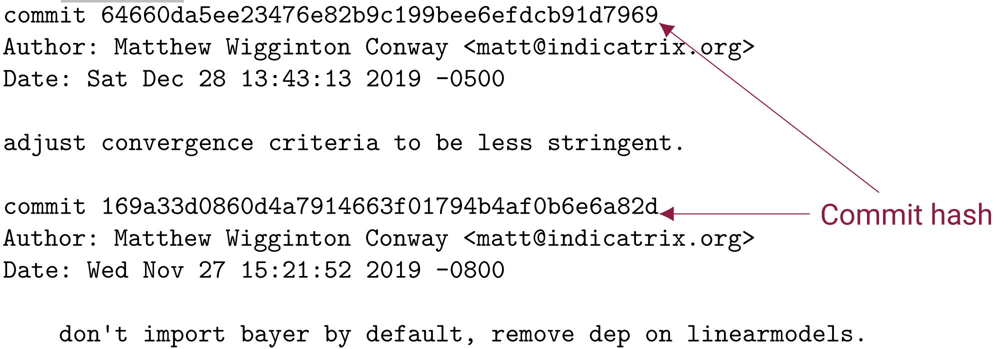
- `git log`

```
commit 64660da5ee23476e82b9c199bee6efdcb91d7969
Author: Matthew Wigginton Conway <matt@indicatrix.org>
Date: Sat Dec 28 13:43:13 2019 -0500

adjust convergence criteria to be less stringent.

commit 169a33d0860d4a7914663f01794b4af0b6e6a82d
Author: Matthew Wigginton Conway <matt@indicatrix.org>
Date: Wed Nov 27 15:21:52 2019 -0800

    don't import bayer by default, remove dep on linearmodels.
```

Commit hash

Getting back to an old commit

- `git log`



```
commit 64660da5ee23476e82b9c199bee6efdcb91d7969
Author: Matthew Wigginton Conway <matt@indicatrix.org>
Date: Sat Dec 28 13:43:13 2019 -0500

adjust convergence criteria to be less stringent.

commit 169a33d0860d4a7914663f01794b4af0b6e6a82d
Author: Matthew Wigginton Conway <matt@indicatrix.org>
Date: Wed Nov 27 15:21:52 2019 -0800

    don't import bayer by default, remove dep on linearmodels.
```

Commit hash

- To go back to commit `169a33d`: `git checkout 169a33d`
- This *does not* change the master branch

Doing more work based on an old commit

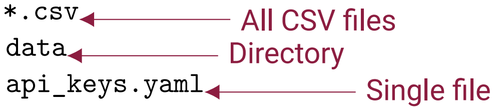
- Create new branch based on old commit or tag
- First, get to old tag or commit with `git checkout`
- Then, `git switch -c new-branch-name`
  - `git checkout -b new-branch-name` on old versions
- This creates a new branch that you can work on and commit to based on old version
- `git switch master` to get back to the most recent code
  - `git checkout master` on old versions
- Merging this branch could be messy

Getting a single file from an old commit or tag

- `git restore --source=<commit hash or tag> filename`
  - `git checkout <commit hash or tag> -- filename` on old versions
- `git add filename` then `git commit` to commit old version of file

What about files I don’t want to commit?

- Create a file `.gitignore` in your repository
- List any files or file patterns you don’t want to commit
- e.g.



```
*.csv
data
api_keys.yaml
```

All CSV files

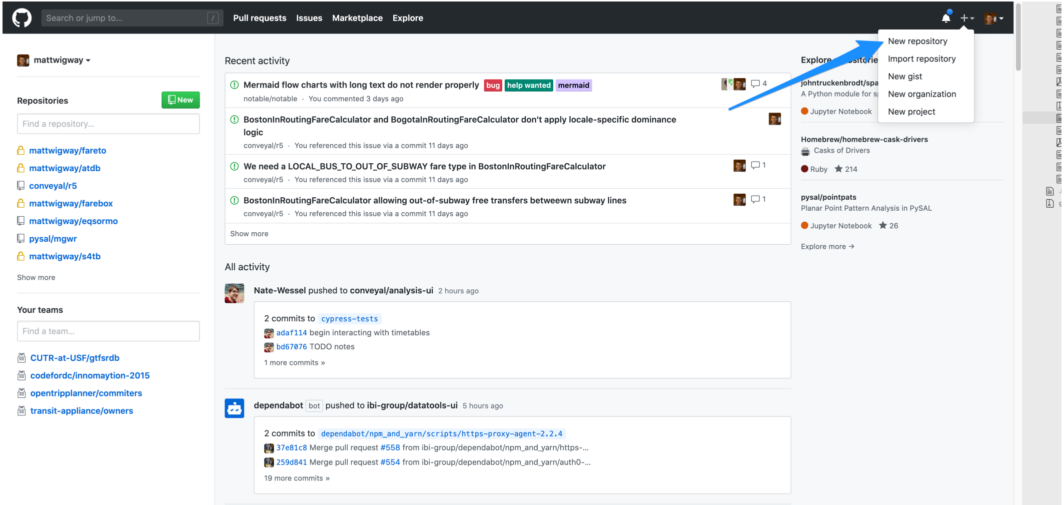
Directory

Single file

Pushing to GitHub

- You don’t have to use use GitHub to use `git`
- Repositories can be *public* or *private*
- In public repositories, all files and their history can be seen by anyone
  - Just deleting sensitive data before making a repository public doesn’t cut it...

Creating a repository



Search or jump to... /

Pull requests Issues Marketplace Explore

New repository
Import repository
New gist
New organization
New project

mattwigway

Repositories New

Find a repository...

- mattwigway/fareto
- mattwigway/atdb
- conveyal/r5
- mattwigway/farebox
- mattwigway/eqsormo
- pysal/mgwr
- mattwigway/s4tb

Show more

Your teams

Find a team...

- CUTR-at-USF/gtfsrdb
- codefordc/innomaytion-2015
- opentripplanner/commiters
- transit-appliance/owners

Recent activity

Mermaid flow charts with long text do not render properly bug help wanted mermaid 4
notable/notable · You commented 3 days ago

BostonInRoutingFareCalculator and BogotaInRoutingFareCalculator don't apply locale-specific dominance logic
conveyal/r5 · You referenced this issue via a commit 11 days ago

We need a LOCAL_BUS_TO_OUT_OF_SUBWAY fare type in BostonInRoutingFareCalculator 1
conveyal/r5 · You referenced this issue via a commit 11 days ago

BostonInRoutingFareCalculator allowing out-of-subway free transfers betweewn subway lines 1
conveyal/r5 · You referenced this issue via a commit 11 days ago

Show more

All activity

Nate-Wessel pushed to conveyal/analysis-ui 2 hours ago

2 commits to `cypress-tests`
`adaf114` begin interacting with timetables
`bd67076` TODO notes
1 more commits »

dependabot bot pushed to ibi-group/datatools-ui 5 hours ago

2 commits to `dependabot/npm_and_yarn/scripts/https-proxy-agent-2.2.4`
`37e81c8` Merge pull request #558 from ibi-group/dependabot/npm_and_yarn/https-...
`259d841` Merge pull request #554 from ibi-group/dependabot/npm_and_yarn/auth0-...
19 more commits »

Explore r ositorie

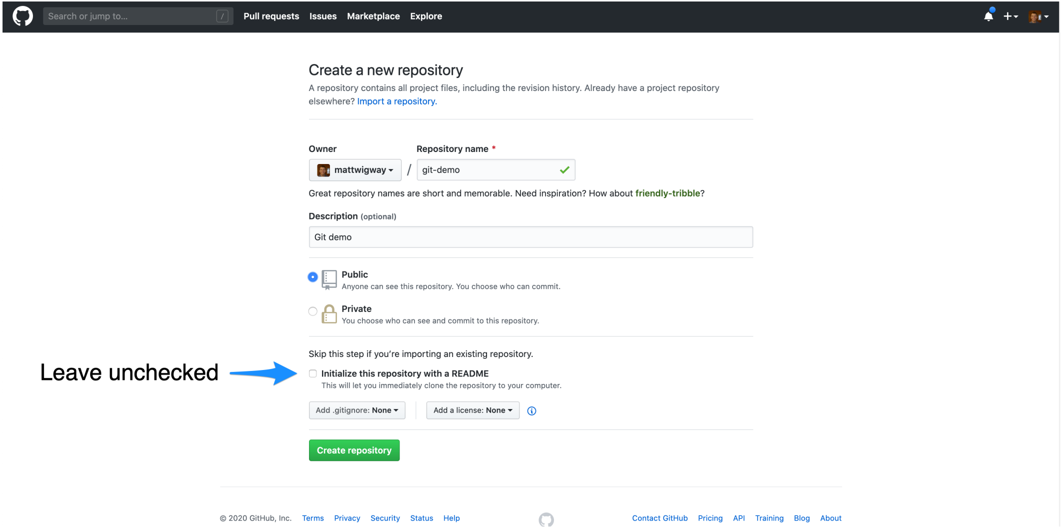
johntruckenbrodt/spa
A Python module for s
● Jupyter Notebook

Homebrew/homebrew-cask-drivers
Casks of Drivers
● Ruby ★ 214

pysal/pointpats
Planar Point Pattern Analysis in PySAL
● Jupyter Notebook ★ 26

Explore more →

Creating a repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: mattwigway / Repository name *: git-demo ✓

Great repository names are short and memorable. Need inspiration? How about **friendly-tribble**?

Description (optional): Git demo

● **Public**
Anyone can see this repository. You choose who can commit.

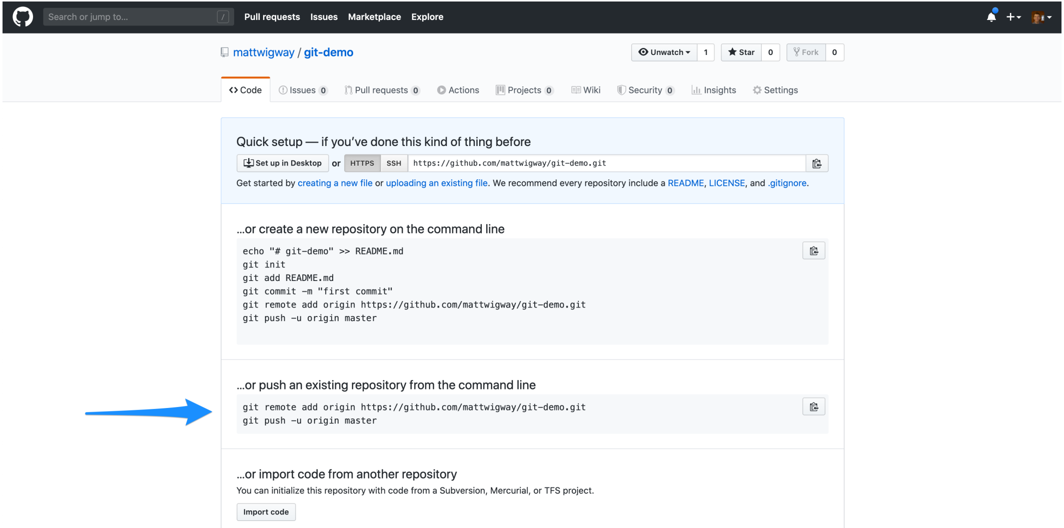
○ **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Leave unchecked → □ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** Add a license: **None**

Creating a repository



Quick setup — if you've done this kind of thing before

```
https://github.com/mattwigway/git-demo.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# git-demo" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/mattwigway/git-demo.git
git push -u origin master
```

…or push an existing repository from the command line

```
git remote add origin https://github.com/mattwigway/git-demo.git
git push -u origin master
```

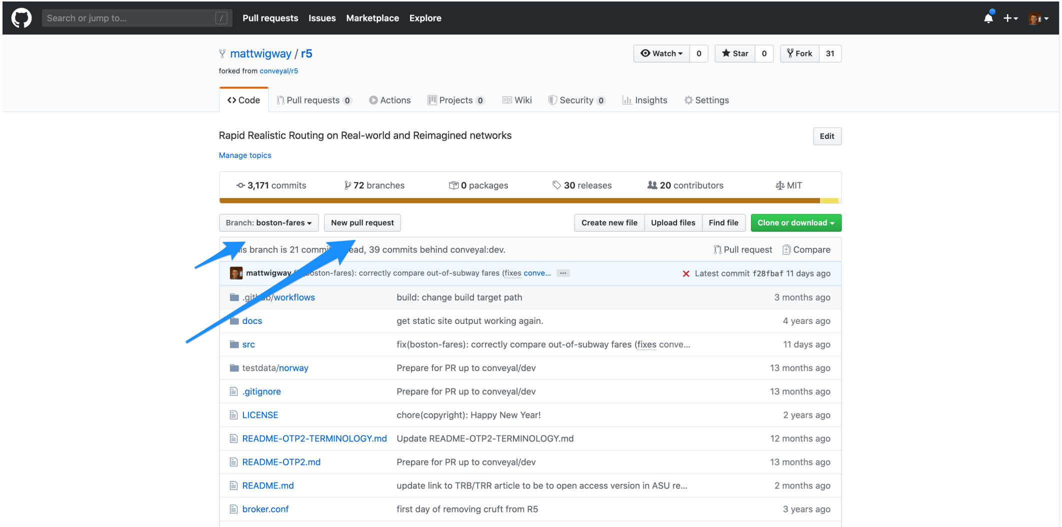
…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Working with GitHub

- After making commits on your computer: `git push`
  - To push a new branch: `git push -u origin branchname`
- After someone else has made commits: `git pull`
- After someone else has pushed a branch:
  - `git fetch`
  - `git switch <branch name>`

GitHub: using pull requests



mattwigway / r5

forked from conveyal/r5

Rapid Realistic Routing on Real-world and Reimagined networks

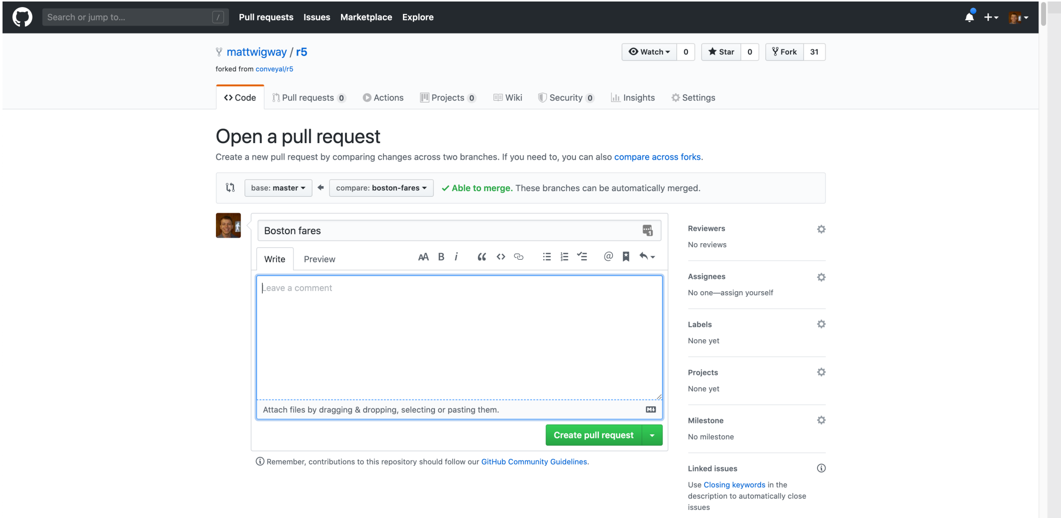
3,171 commits | 72 branches | 0 packages | 30 releases | 20 contributors | MIT

This branch is 21 commits ahead, 39 commits behind conveyal:dev.

mattwigway fix(boston-fares): correctly compare out-of-subway fares (fixes conve... ··· ✕ Latest commit f28fbaf 11 days ago

.github/workflows	build: change build target path	3 months ago
docs	get static site output working again.	4 years ago
src	fix(boston-fares): correctly compare out-of-subway fares (fixes conve...	11 days ago
testdata/norway	Prepare for PR up to conveyal/dev	13 months ago
.gitignore	Prepare for PR up to conveyal/dev	13 months ago
LICENSE	chore(copyright): Happy New Year!	2 years ago
README-OTP2-TERMINOLOGY.md	Update README-OTP2-TERMINOLOGY.md	12 months ago
README-OTP2.md	Prepare for PR up to conveyal/dev	13 months ago
README.md	update link to TRB/TRR article to be to open access version in ASU re...	2 months ago
broker.conf	first day of removing cruft from R5	3 years ago

GitHub: using pull requests



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: master ← compare: boston-fares ✓ Able to merge. These branches can be automatically merged.

Boston fares

Write | Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Linked issues: Use Closing keywords in the description to automatically close issues

GitHub: using pull requests

Live demo: `https://github.com/mattwigway/git-demo/pull/1`

Best practices for collaboration using GitHub

- Work on your own branch
- When you’re ready to share your code with the team, make a pull request
- Have others review the pull request before merging

Questions/Contact

mwconway@asu.edu
@mattwigway

Other resources:

- `git` book: `https://git-scm.com/book/`
- GitHub help: `https://help.github.com`