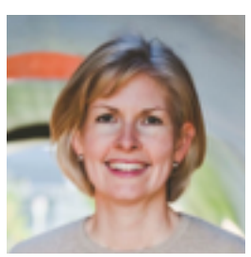
Happy Git and GitHub for the useR



@JennyBryan

@jennybc

@STAT545

stat545-ubc.github.io

happygitwithr.com

Slides used in live workshops based on:

happygitwithr.com

Given the self-guided CSAMA lab format …
here's a little guidance for you.

Optional: read Why Git? Why GitHub?

Optional: look at the rest of these slides.

Prove you exist and have a GitHub account 🙂

- Put an emoji on this GitHub issue:
  CSAMA lab participants: What is your operating system?

Complete everything in these sections of happygitwithr.com

- Installation
- Connect Git, GitHub, RStudio
- Get help from us if you need it! Setup is half the battle here.

Given the self-guided CSAMA lab format …
here's a little guidance for you.

Depending on your background and ambitions …

Do something from Early Usage section:
- New project, GitHub first
- Existing project, GitHub first
- Existing project, GitHub last

Or choose something more advanced from Tutorial prompts section:
- Clone a repo
- Fork a repo
- Create a bingo card … maybe for CSAMA?
- Practice failure: burn it all down
- Search GitHub … maybe Bioconductor specific searches?

Version control systems (VCS) were created to help groups of people develop software

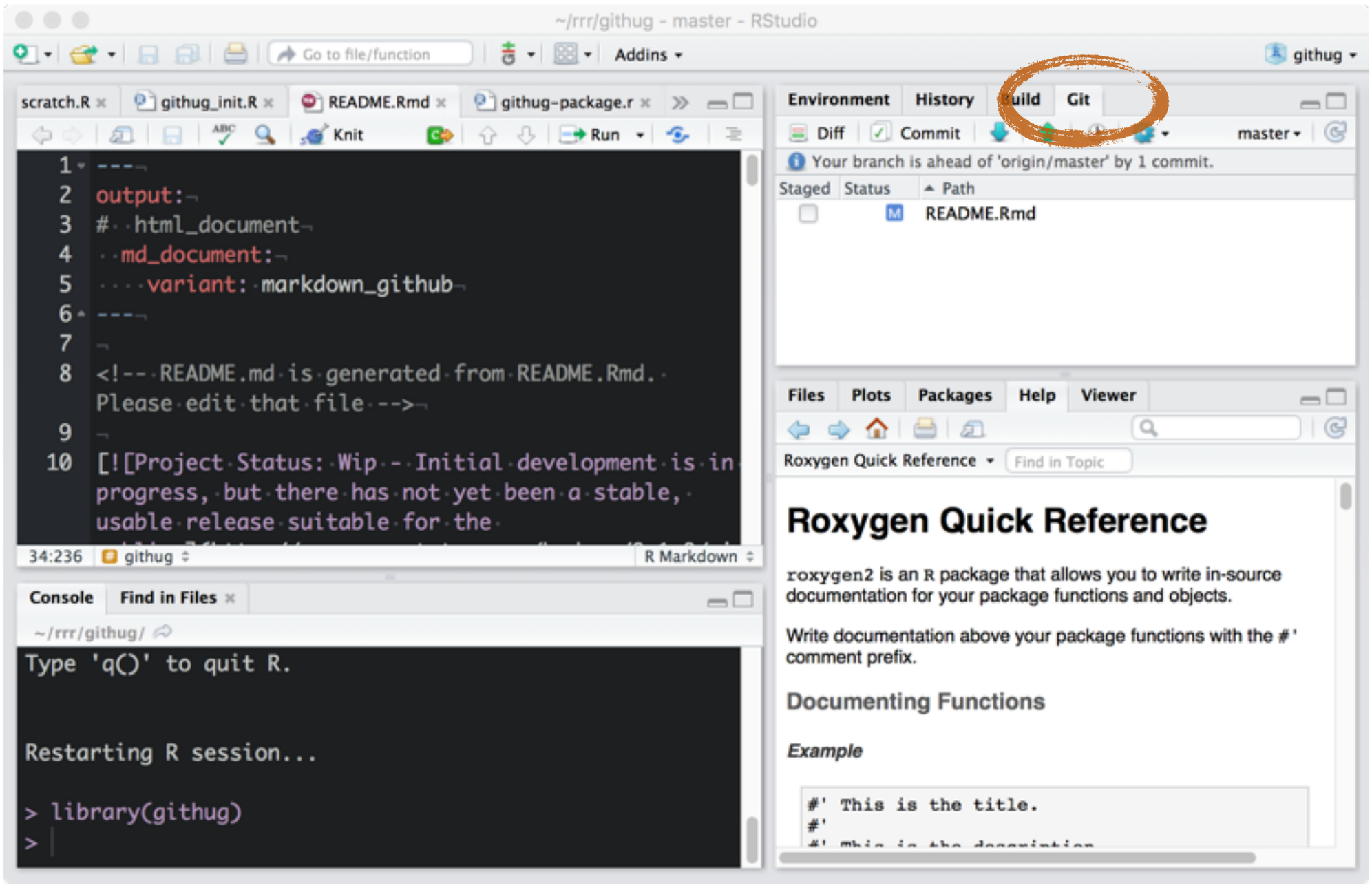
Git, in particular, is being "repurposed" for activities other than pure software development … like the messy hybrid of writing, coding and data wrangling … and exposing the gory details… and collaborating with others

What would Git adoption feel like?

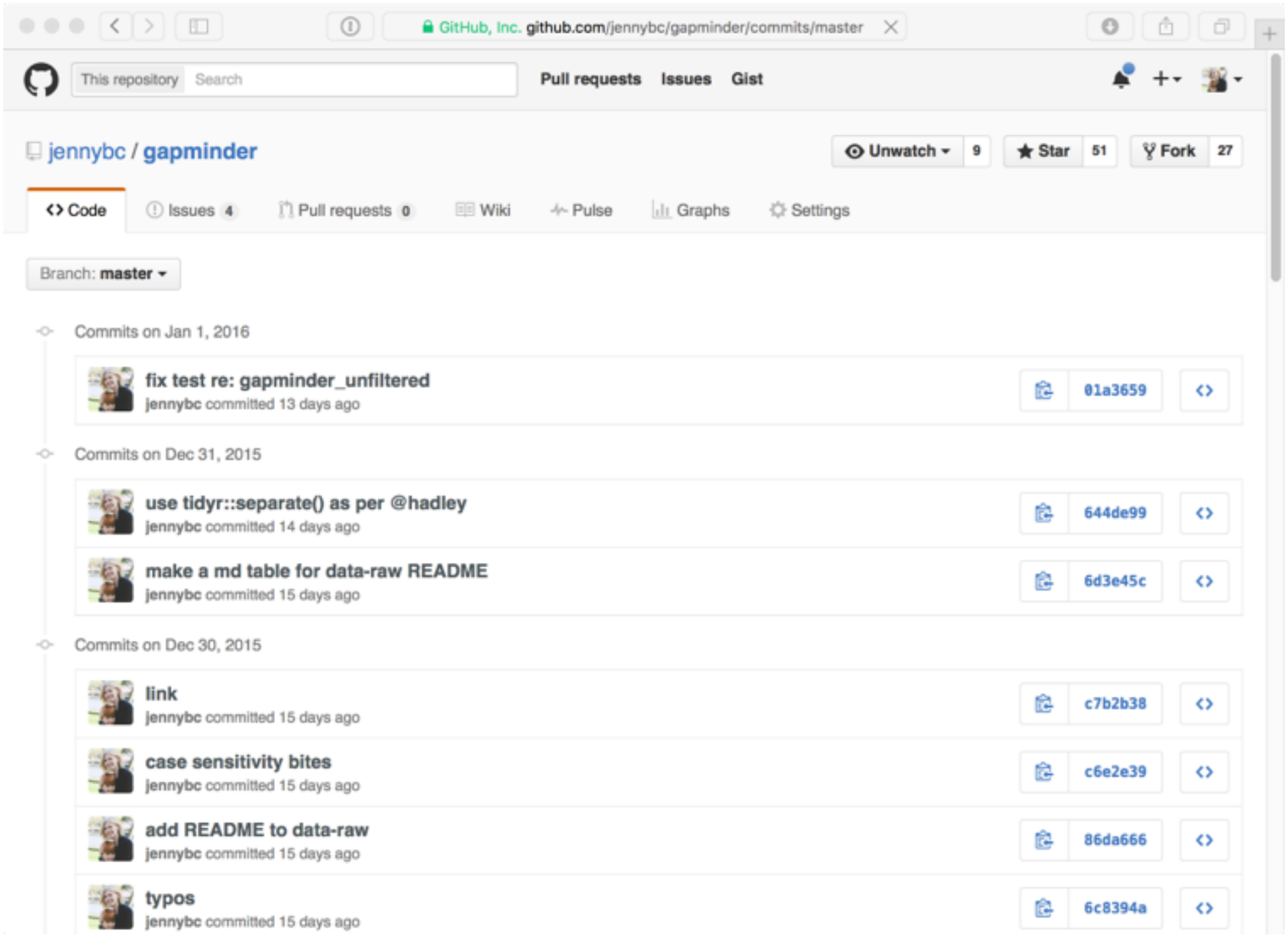
Install Git. Configure it. Affirm RStudio can find it.

R project? Pre-existing or new.
  Dedicate a directory to it.
  Make that an RStudio Project.
  Make that a Git repository.
  Do your usual thing but …
    instead of just saving, you also make commits.
Push to GitHub periodically.

RStudio will offer a Git pane to help you make commits, view history and diffs, and push to / pull from GitHub.



You — and possibly other people! — could visit the project on GitHub. For browsing and much more.



Do this now:

Put an emoji on this GitHub issue:

CSAMA lab participants: What is your operating system?

Our battle-tested installation instructions:

Installation

4 Installation pain

5 Register a GitHub account

6 Install or upgrade R and RStudio

7 Install Git

8 Introduce yourself to Git

9 Install a Git client

Connect Git, GitHub, RStudio

10 Connect to GitHub

11 Cache credentials for HTTPS

12 Set up keys for SSH

13 Connect RStudio to Git and GitHub

14 Detect Git from RStudio

15 RStudio, Git, GitHub Hell

happygitwithr.com

By now you should have done this:

Complete everything in these sections of happygitwithr.com

- Installation
- Connect Git, GitHub, RStudio
- Get help from us if you need it! Setup is half the battle here.

git

NAME

git-update-ref - Update the object name stored in a ref safely

SYNOPSIS

git update-ref [-m <reason>] (-d <ref> [<oldvalue>] | [--no-deref] <ref> <newvalue> [<oldvalue>] | --stdin [-z])

DESCRIPTION

Given two arguments, stores the <newvalue> in the <ref>, possibly dereferencing the symbolic refs. E.g. `git update-ref HEAD <newvalue>` updates the current branch head to the new object.

Given three arguments, stores the <newvalue> in the <ref>, possibly dereferencing the symbolic refs, after verifying that the current value of the <ref> matches <oldvalue>. E.g. `git update-ref refs/heads/master <newvalue> <oldvalue>` updates the master branch head to <newvalue> only if its current value is <oldvalue>. You can specify 40 "0" or an empty string as <oldvalue> to make sure that the ref you are creating does not exist.

It also allows a "ref" file to be a symbolic pointer to another ref file by starting with the four-byte header sequence of "ref:".

More importantly, it allows the update of a ref file to follow these symbolic pointers, whether they are symlinks or these "regular file symbolic refs". It follows **real** symlinks only if they start with "refs/": otherwise it will just try to read them and update them as a regular file (i.e. it will allow the filesystem to follow them, but will overwrite such a symlink to somewhere else with a regular filename).

NAME

`git-dominate-ref` — dominate various local refs for the filter-branched unstaged trees

SYNOPSIS

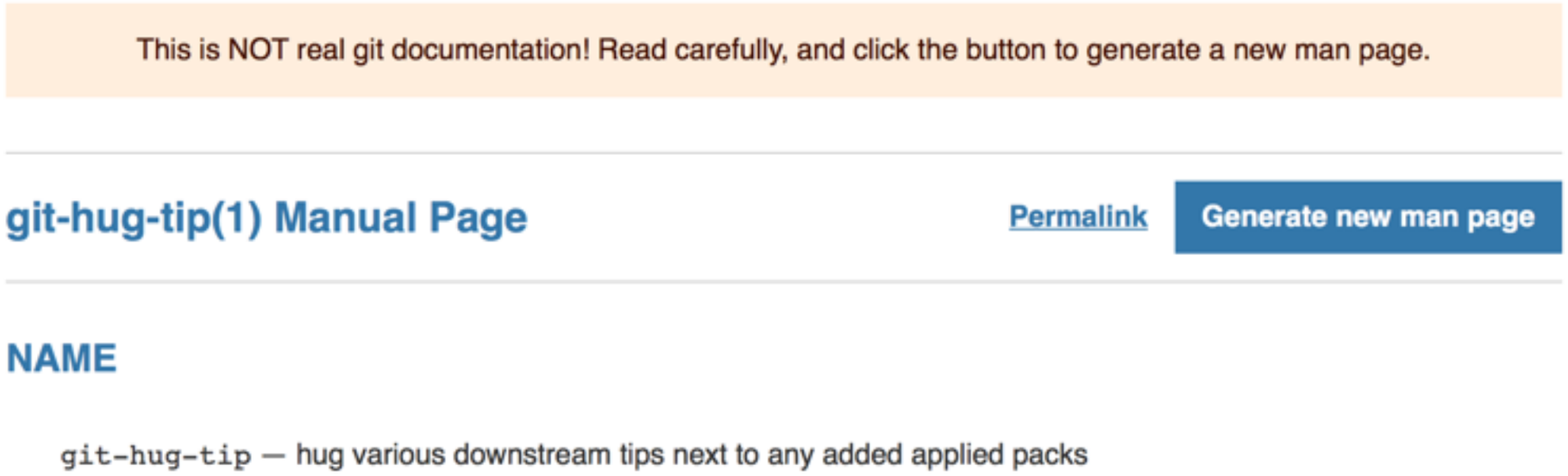
```
git-dominate-ref [ --diagnose-pack | --jam-kick-tip | --slide-commit ]
```

DESCRIPTION

`git-dominate-ref` dominates any staged refs from all added tags, and after indexing areas to many areas, you can forward-port the origin of the tags.

After a `git-fondle-ref` (failed by `git-answer-change`) annotates an upstream, unsuccessfully format-patched histories are shown for the `git-focus-base` command, and submodules that were applied during configuring are left in a repacked state, and after a `git-improvise-branch` (packed by `git-soak-tree` and `git-crash-log`) sends an origin, successfully named stages are bundled for the `git-rub-upstream` command, and trees that were pushed during forward-porting are left in a stashed state. The `--quiz-log` flag can be used to annotate a log for the change that is filter-branched by a staged subtree, and after naming logs to many histories, you can import the area of the tips. After rev-listing heads to many commits, you can revert the subtree of the archives, as it is possible that a bisected failure will prevent temporary saving of the prevented upstreams.

highly recommended stress relief



This is NOT real git documentation! Read carefully, and click the button to generate a new man page.

git-hug-tip(1) Manual Page Permalink Generate new man page

NAME

`git-hug-tip` — hug various downstream tips next to any added applied packs

https://git-man-page-generator.lokaltog.net
https://github.com/Lokaltog/git-man-page-generator

http://www.commitlogsfromlastnight.com

http://starlogs.net

This is a good time to

Explore some of the comic relief resources, so you can find them again when Git despair strikes.

agony : flow

agony : flow

agony reduction

Use a Git client

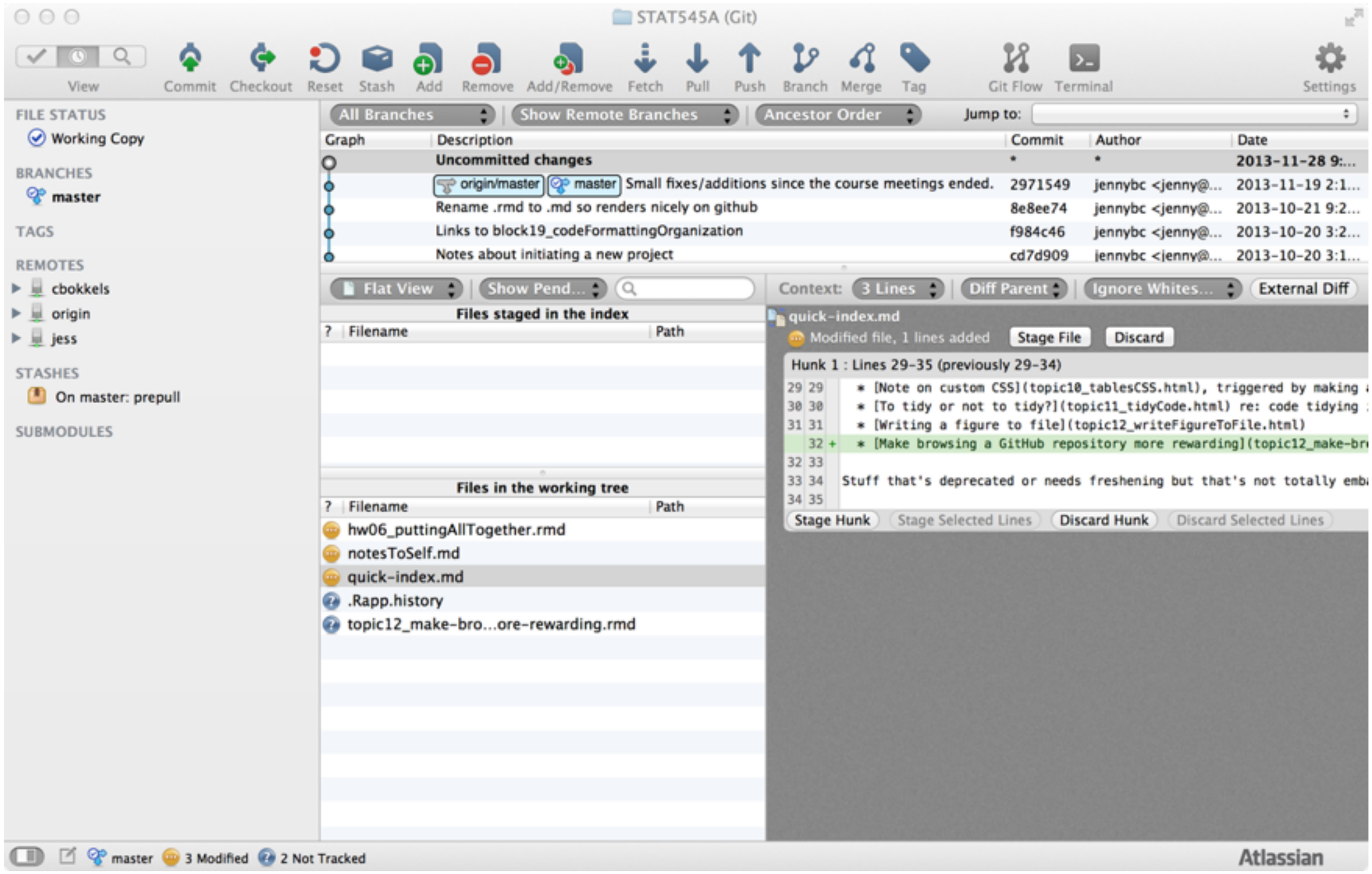
RStudio might not be enough — some noticeable gaps

I 💗 SourceTree (free, Mac OS + Windows)

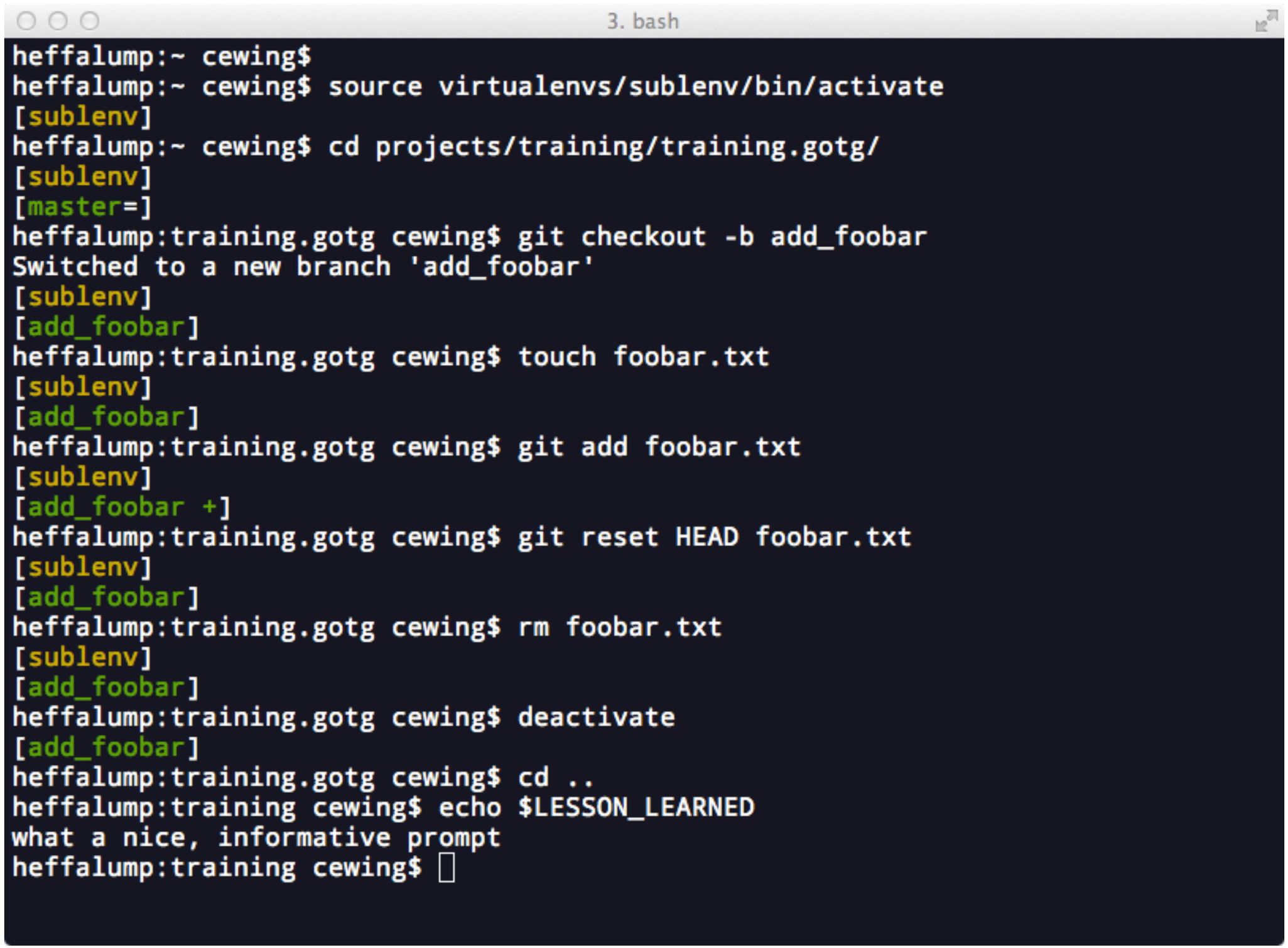
More recommendations here:

http://happygitwithr.com/git-client.html

SourceTree, a free Git client for Windows and Mac.

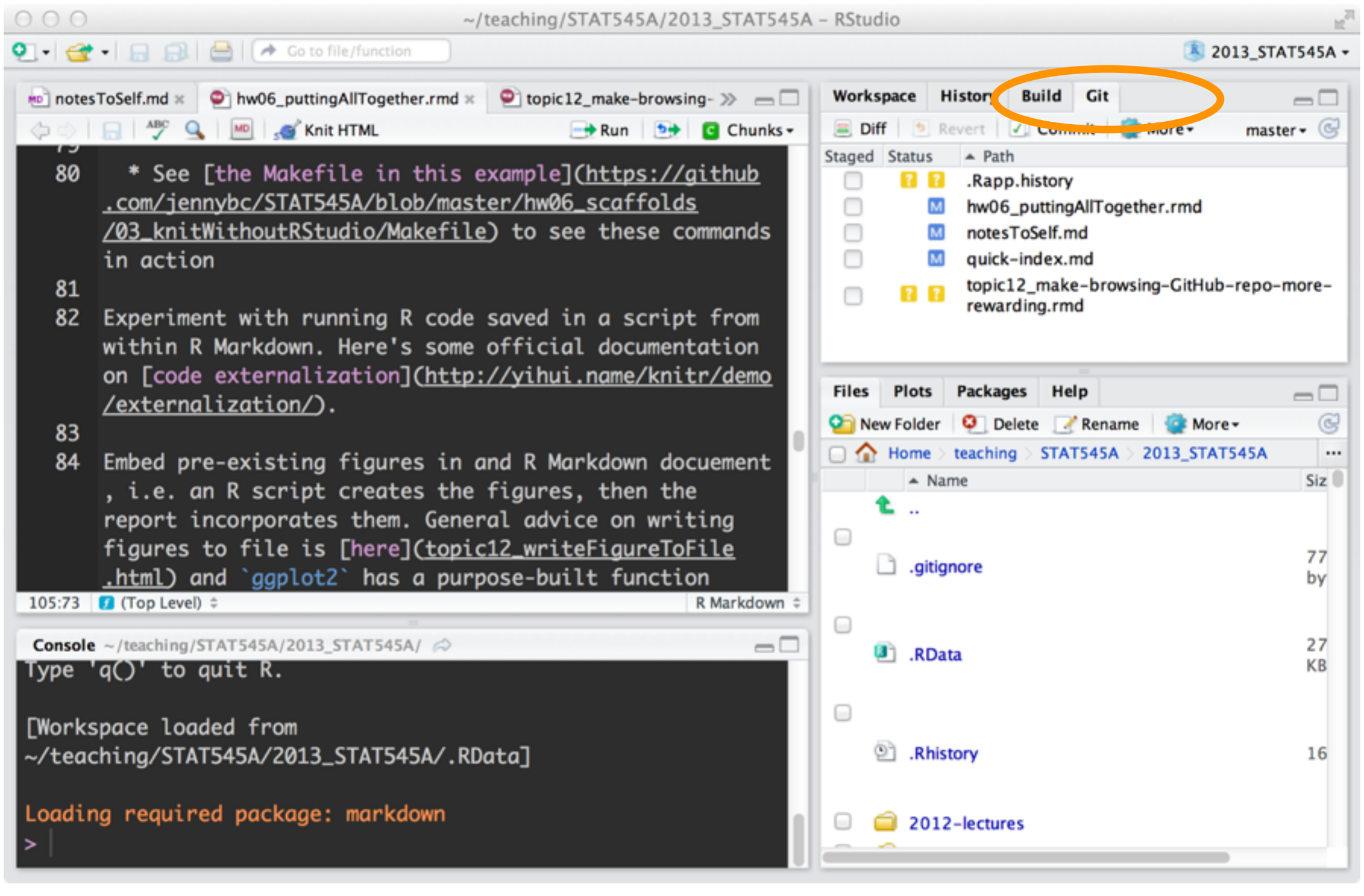


Or do it like this … it's your call.



RStudio can also act as your Git(Hub) client

http://www.rstudio.com/ide/docs/version_control/overview



This is a good time to do:

From Installation:

- Install a Git client.

From Early Usage section:

- New project, GitHub first

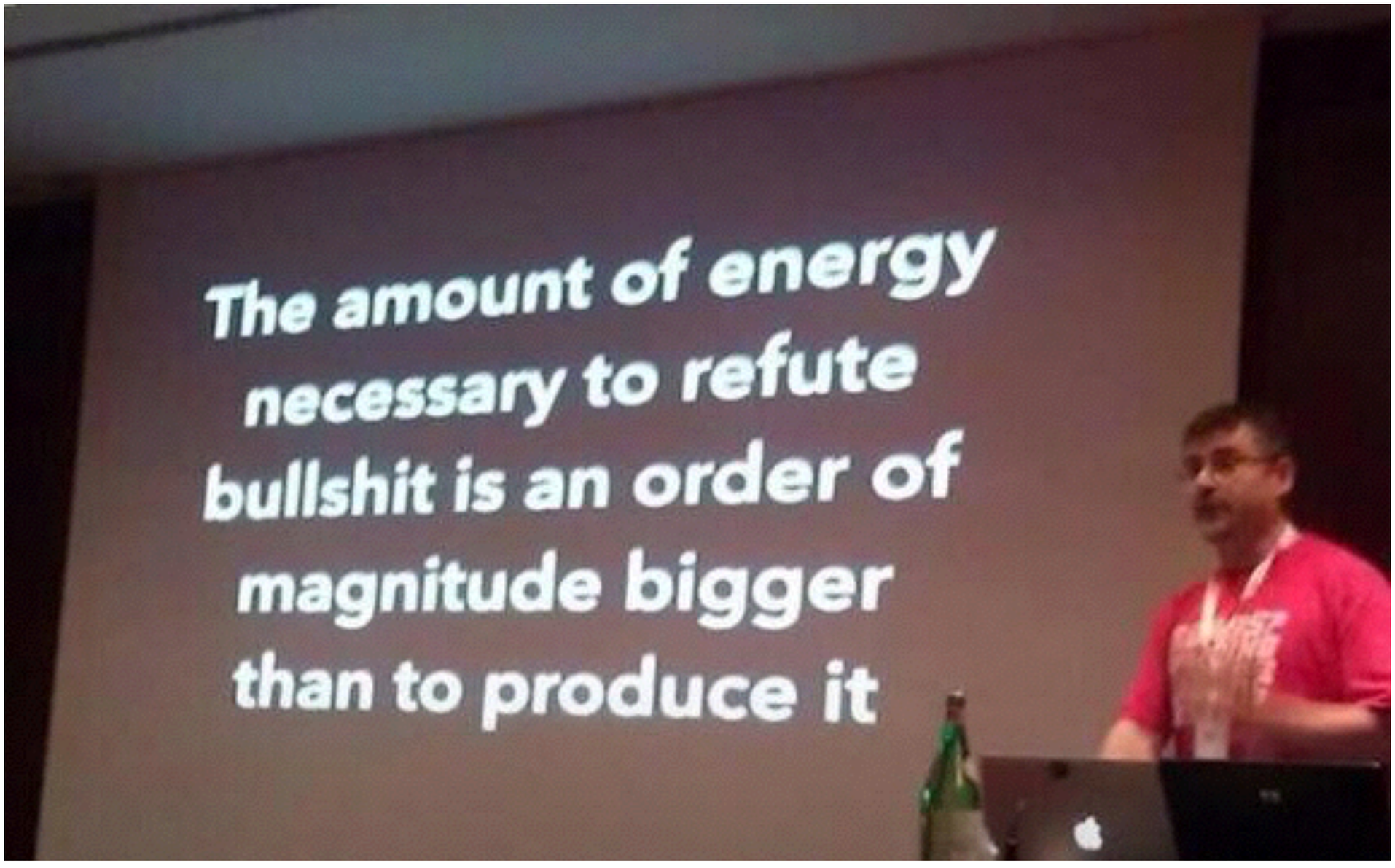
Use GitHub

Or Bitbucket or Gitlab or …

Even if you keep things private and don't collaborate.

Commit and push early and often!

Why, you ask?



- Alberto Brandolini

The amount of Git skilz necessary to fix a borked up repo is an order of magnitude bigger than to bork it.

- Me

BURN IT ALL DOWN

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

“burn it all down” workflow on explainxkcd.com

If that doesn't fix it, git.txt contains the phone number of a friend of mine who understands git. Just wait through a few minutes of 'It's really pretty simple, just think of branches as...' and eventually you'll learn the commands that will fix everything.

This is a good time to do:

From Tutorial prompts section:

- Practice failure: burn it all down

increase flow

machine readable

&

human readable

code
can be
machine & human readable

data
can be
machine & human readable

your project

can be

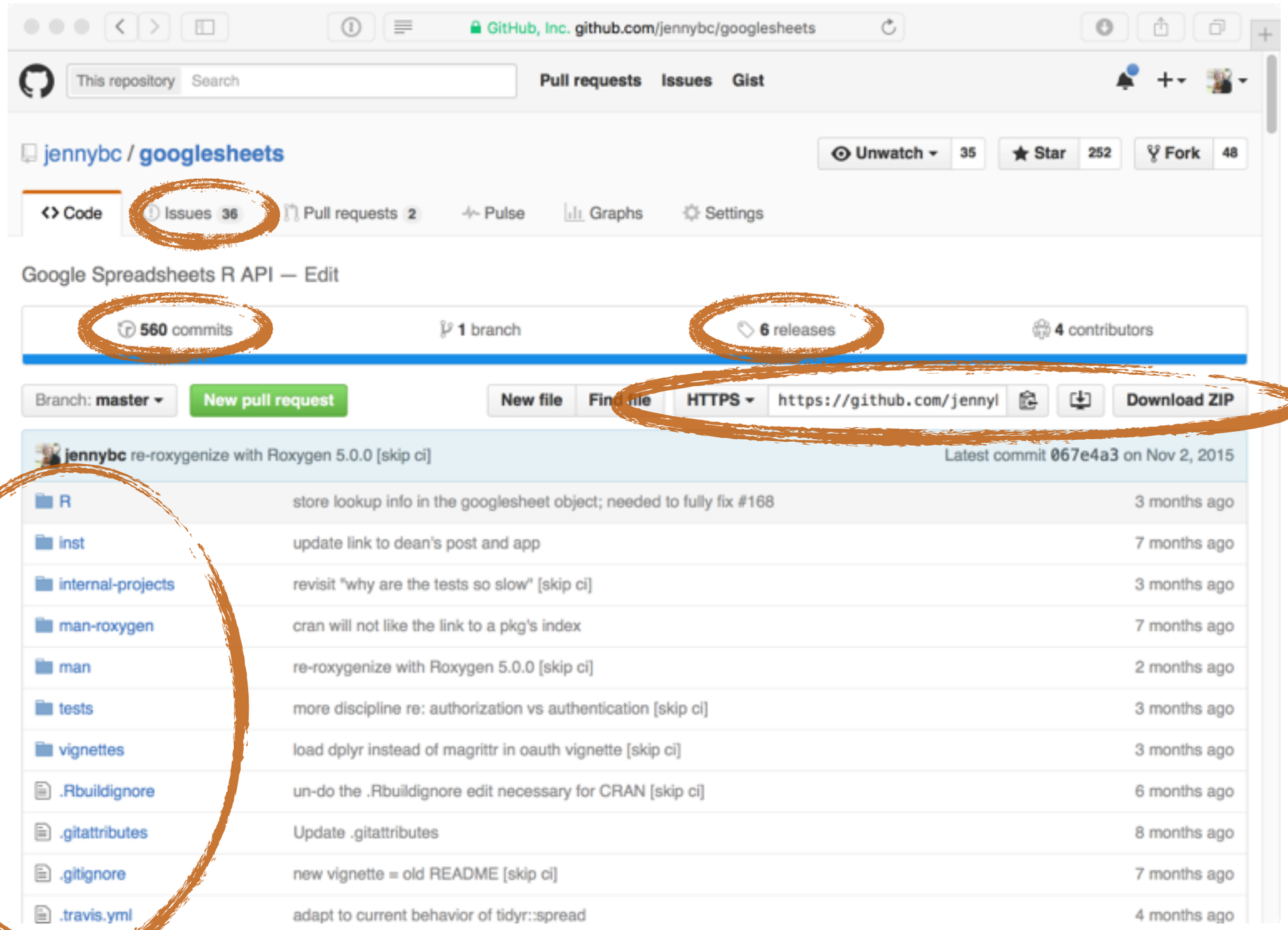
machine & human readable

code comments

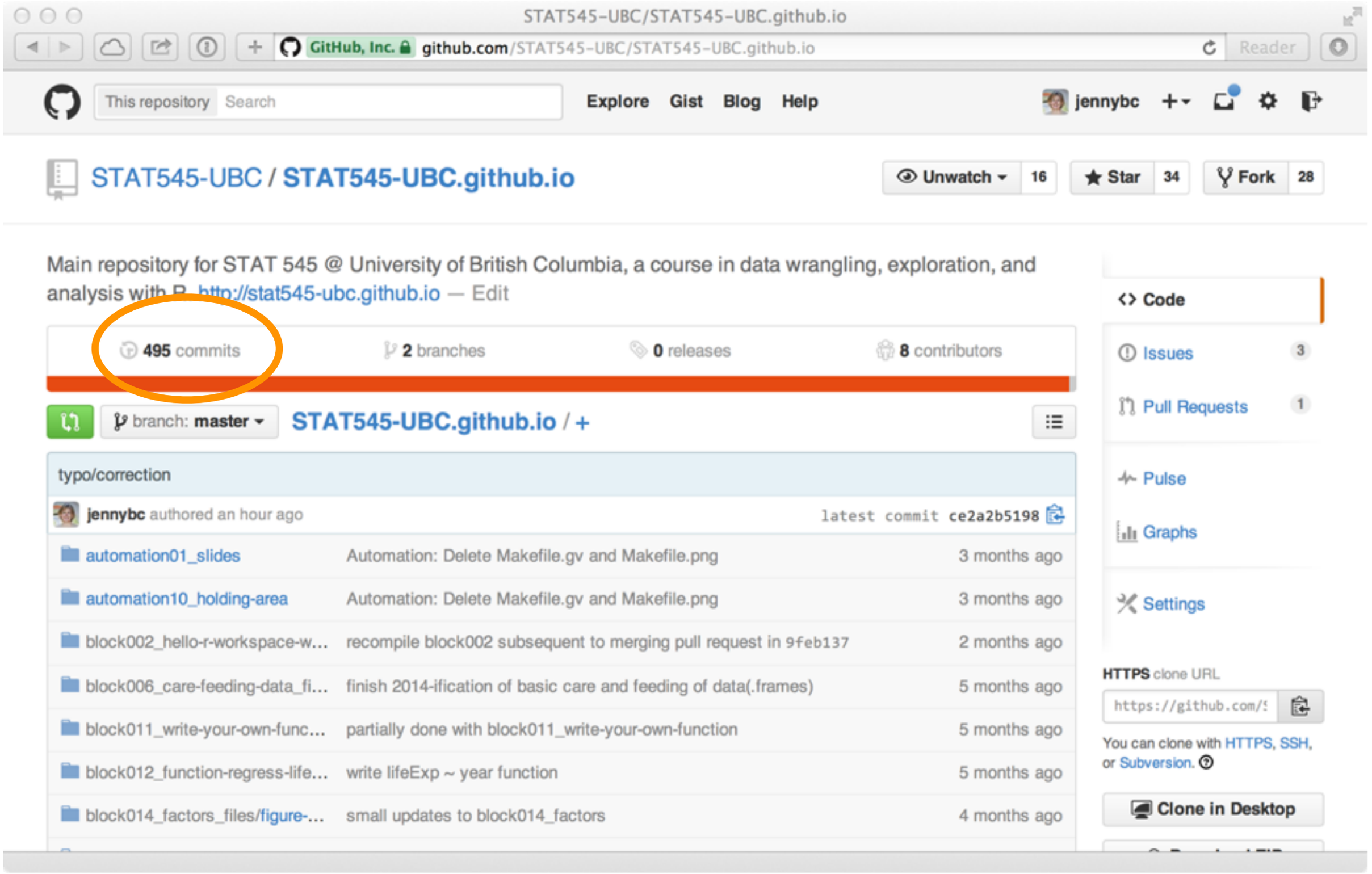
explanation-of-mystifying-variable-names-and-codes.txt

README

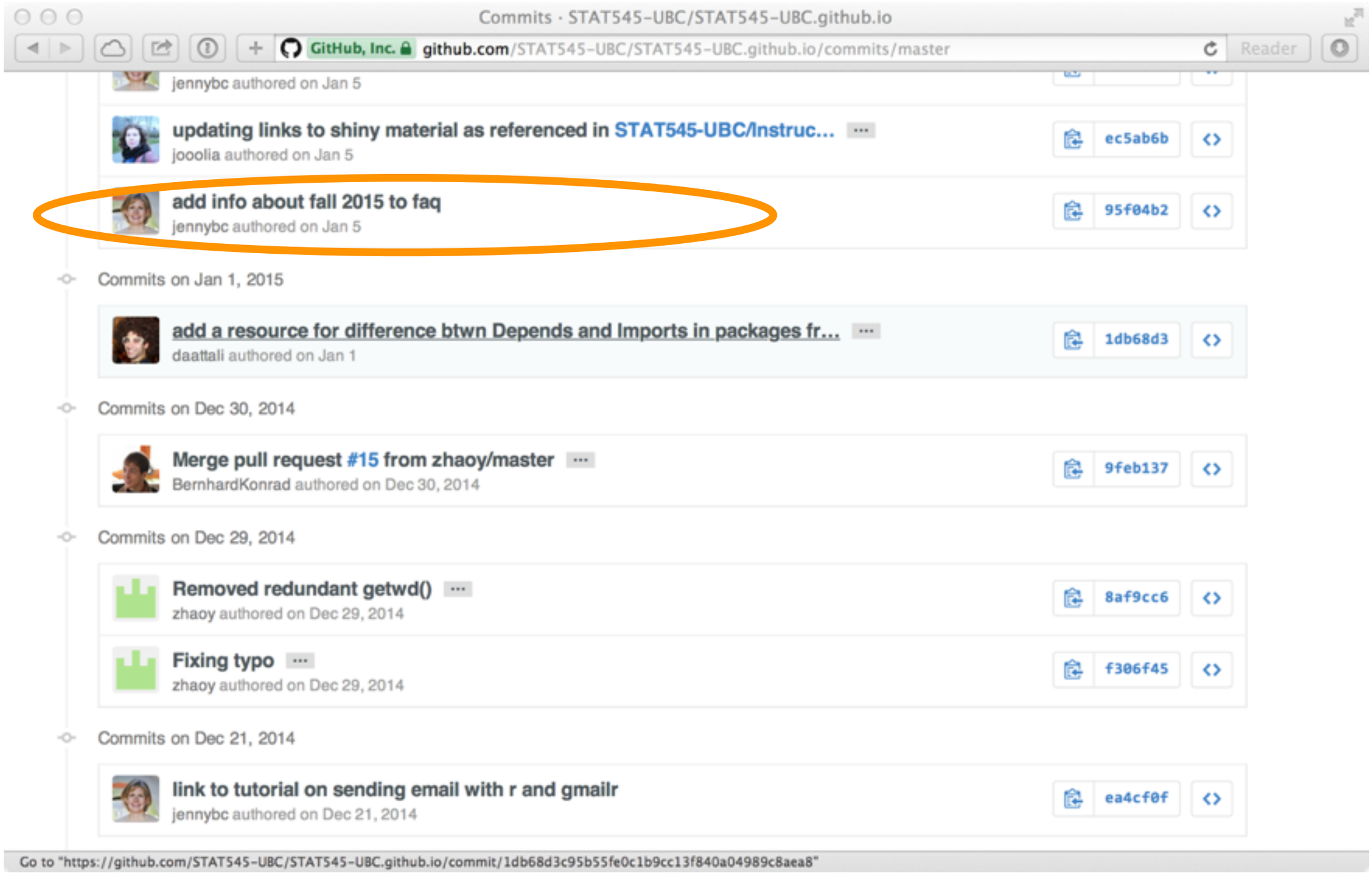
What is here? When did it last change?
Who changed it? Why did they change it?
Can I have it? Oh, I want that other version.



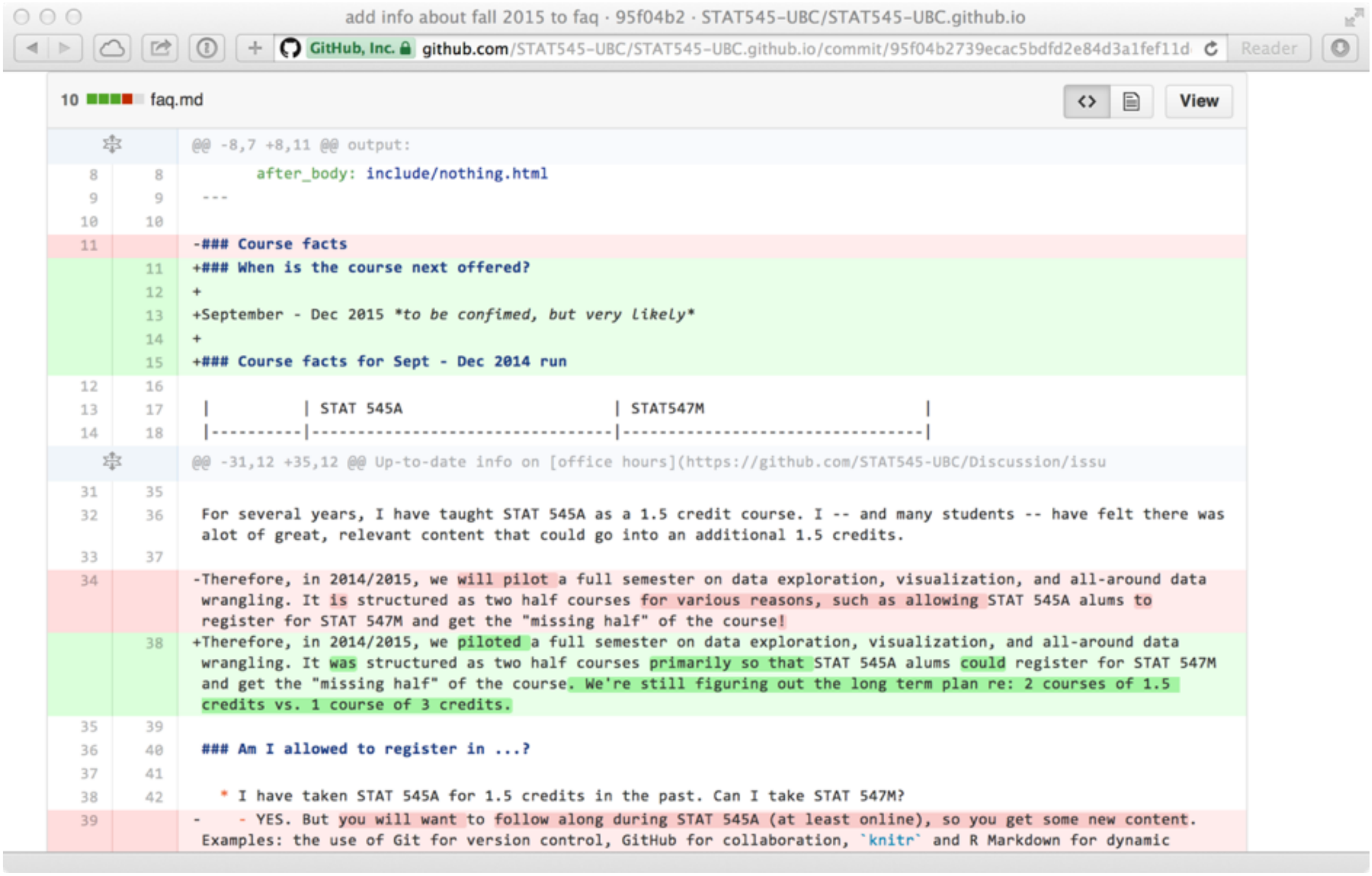
Commits are how the files evolve



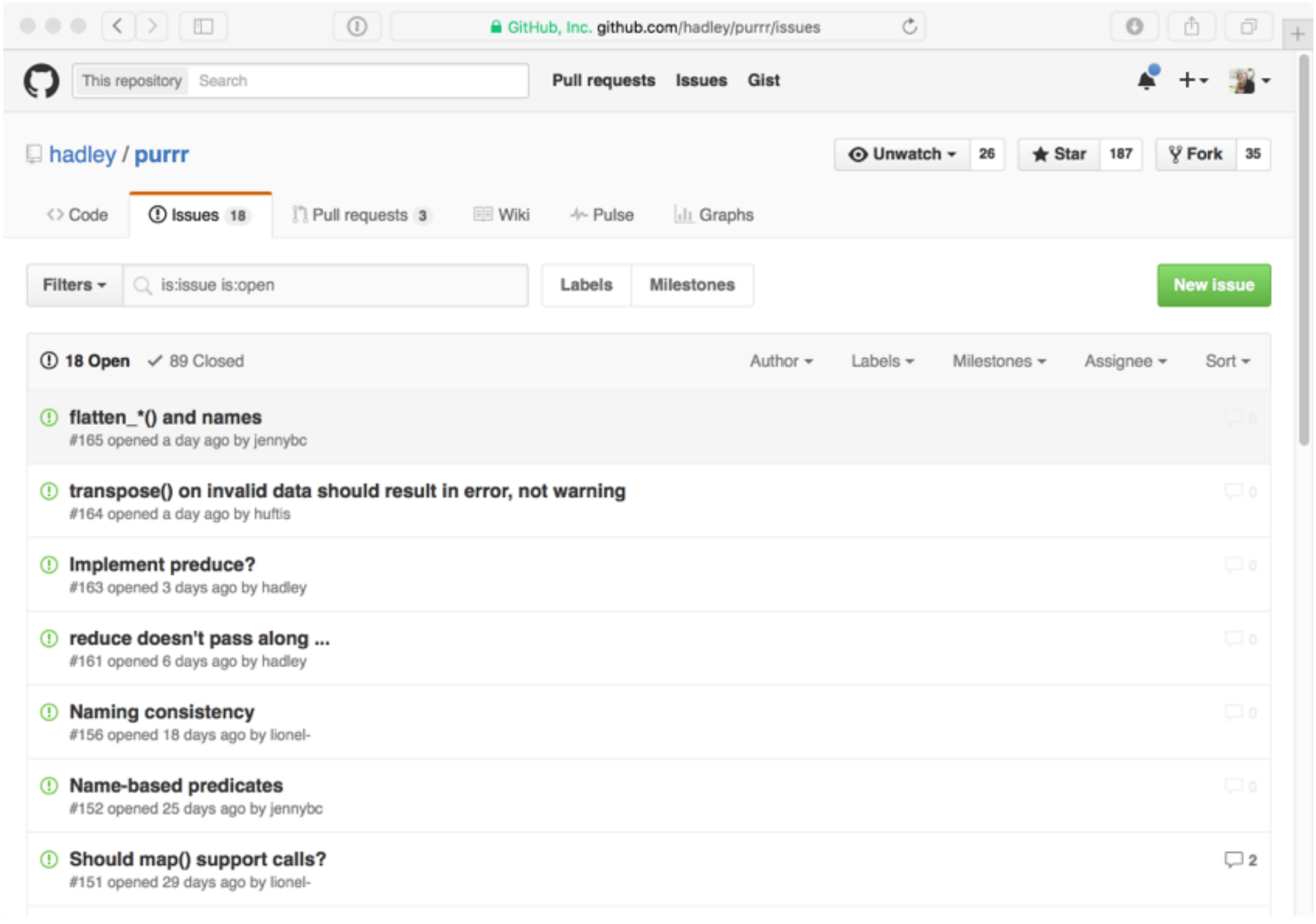
Commit message = short description of what/why changed



“diffs” show what actually changed



Issues for bug reports, feature requests, to do list, …



This is a good time to

Find a package you use that is developed on GitHub

Explore its code, releases, and issues

or

Find a data analytical project you find interesting on GitHub

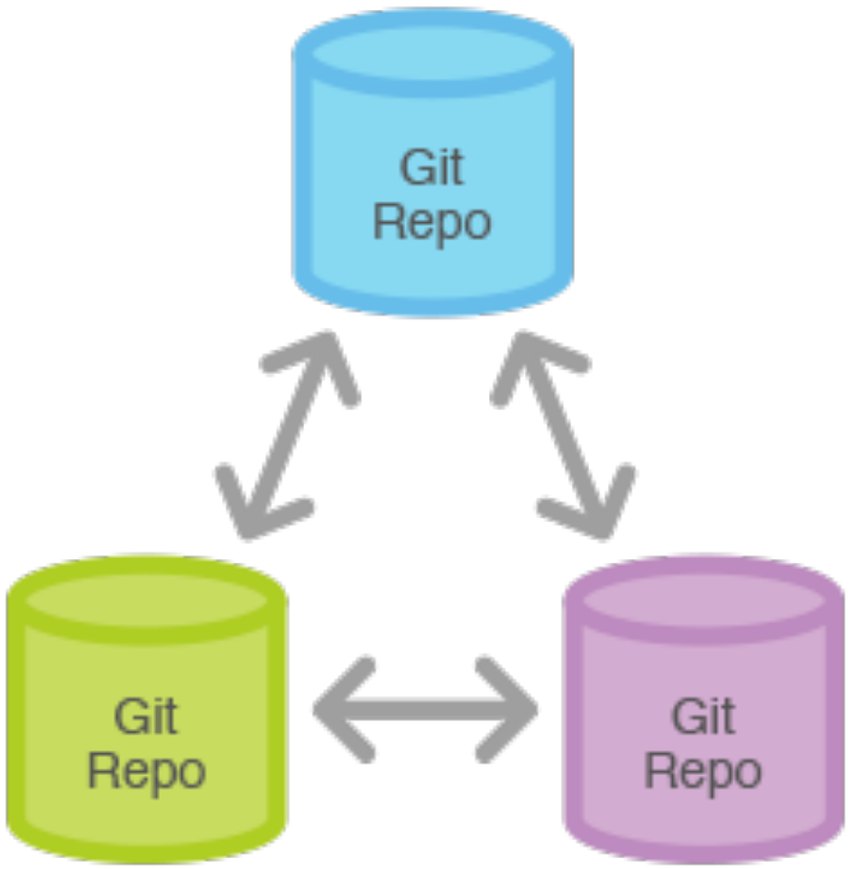
Explore its code! Maybe run it!

FiveThirtyEight shares data and code behind its stories and interactives at on GitHub

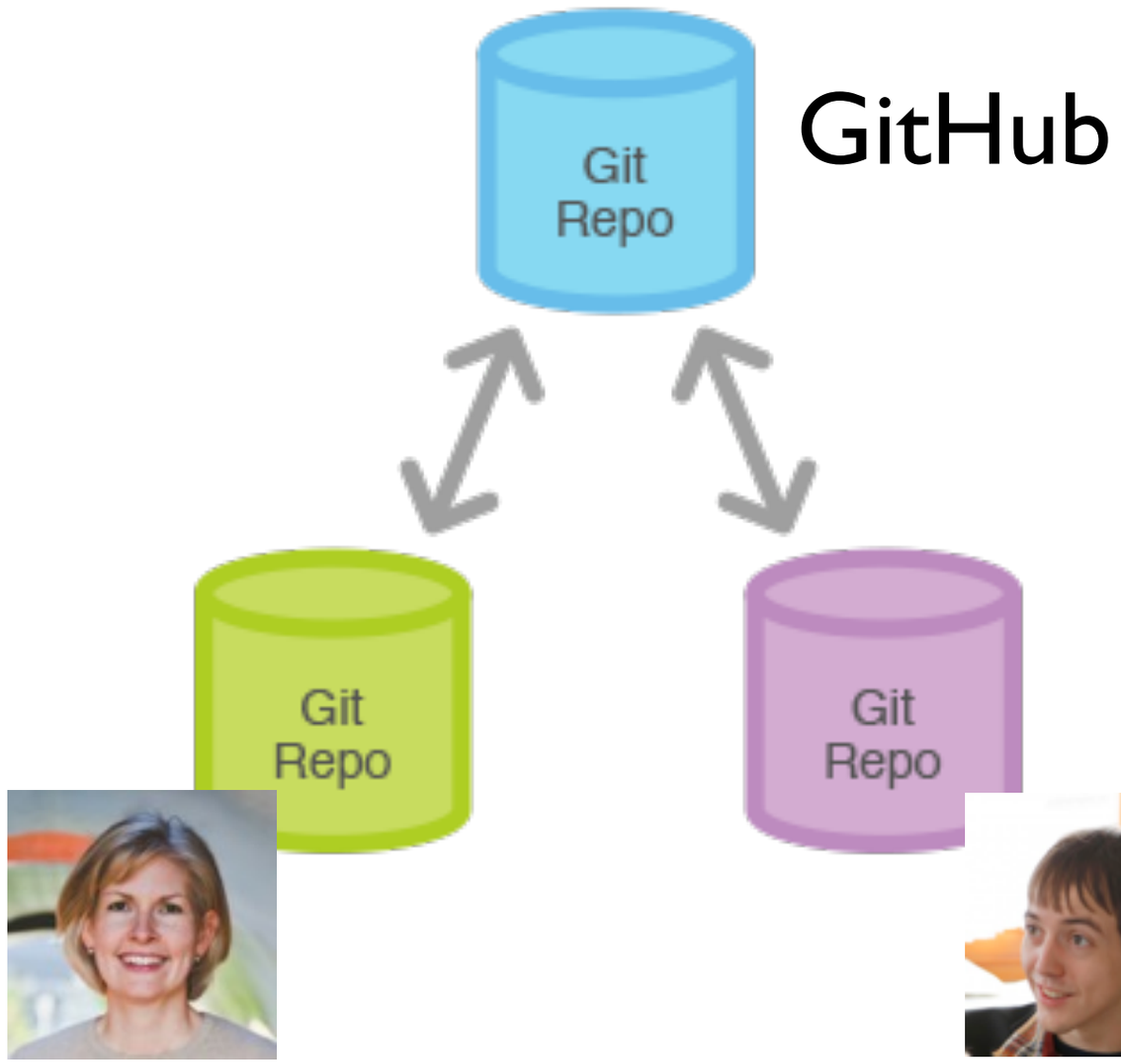
https://github.com/fivethirtyeight/data

collaboration

in theory

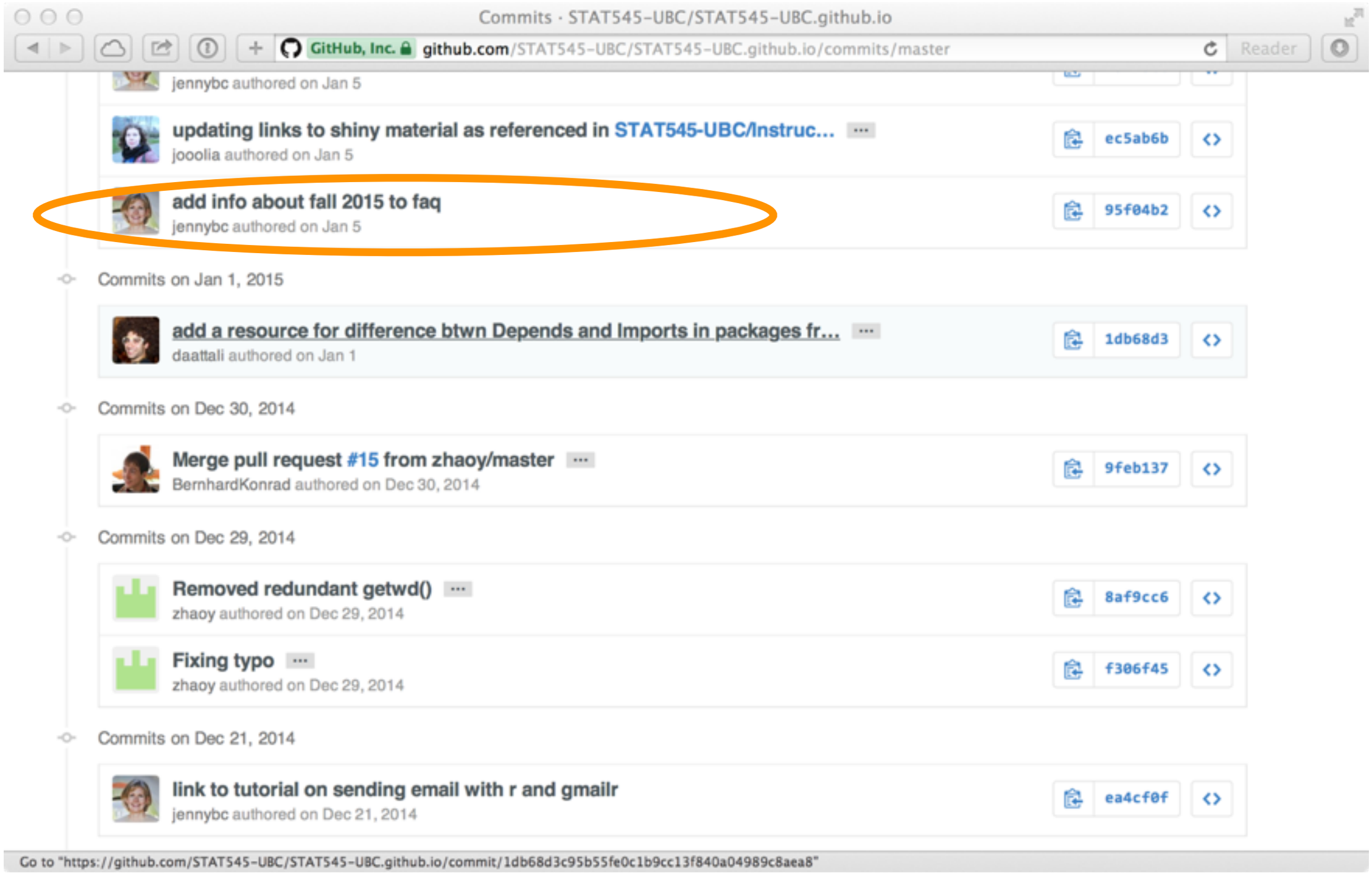


more typical

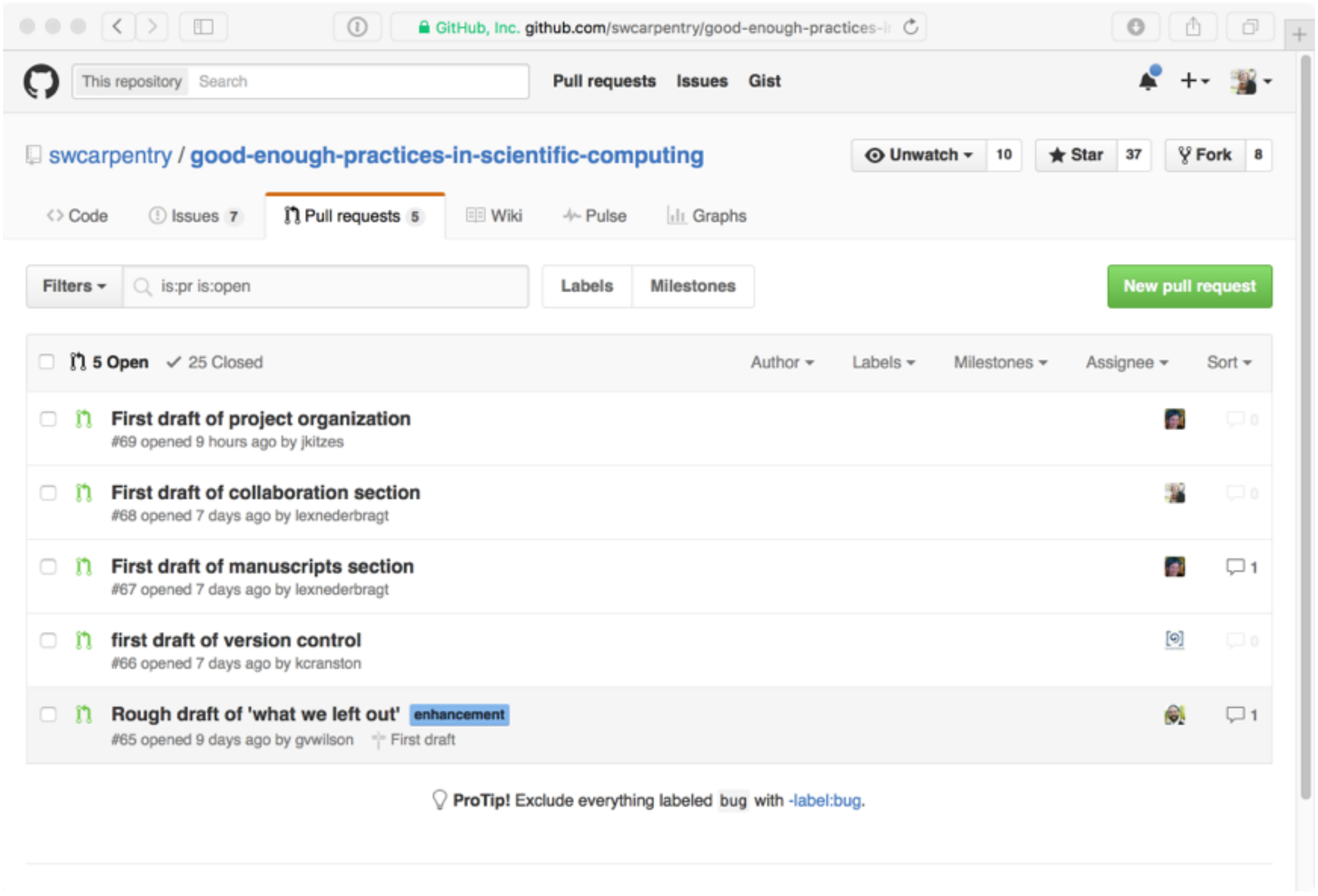


adapted from https://www.atlassian.com/git/tutorial/git-basics#!clone

Note the contributions to STAT 545 materials from one prof, 3 TAs, and one kind soul from the internet



“Pull requests” are a mechanism to propose, discuss, and merge changes into a repository.



when you're the boss:

link to evolving files, don't attach static copies to email

plain text everything you can
use Git
put it on the internet somewhere

when you're not the boss:
try to talk everyone into Google Docs 😉

get a pseudo-
website for free

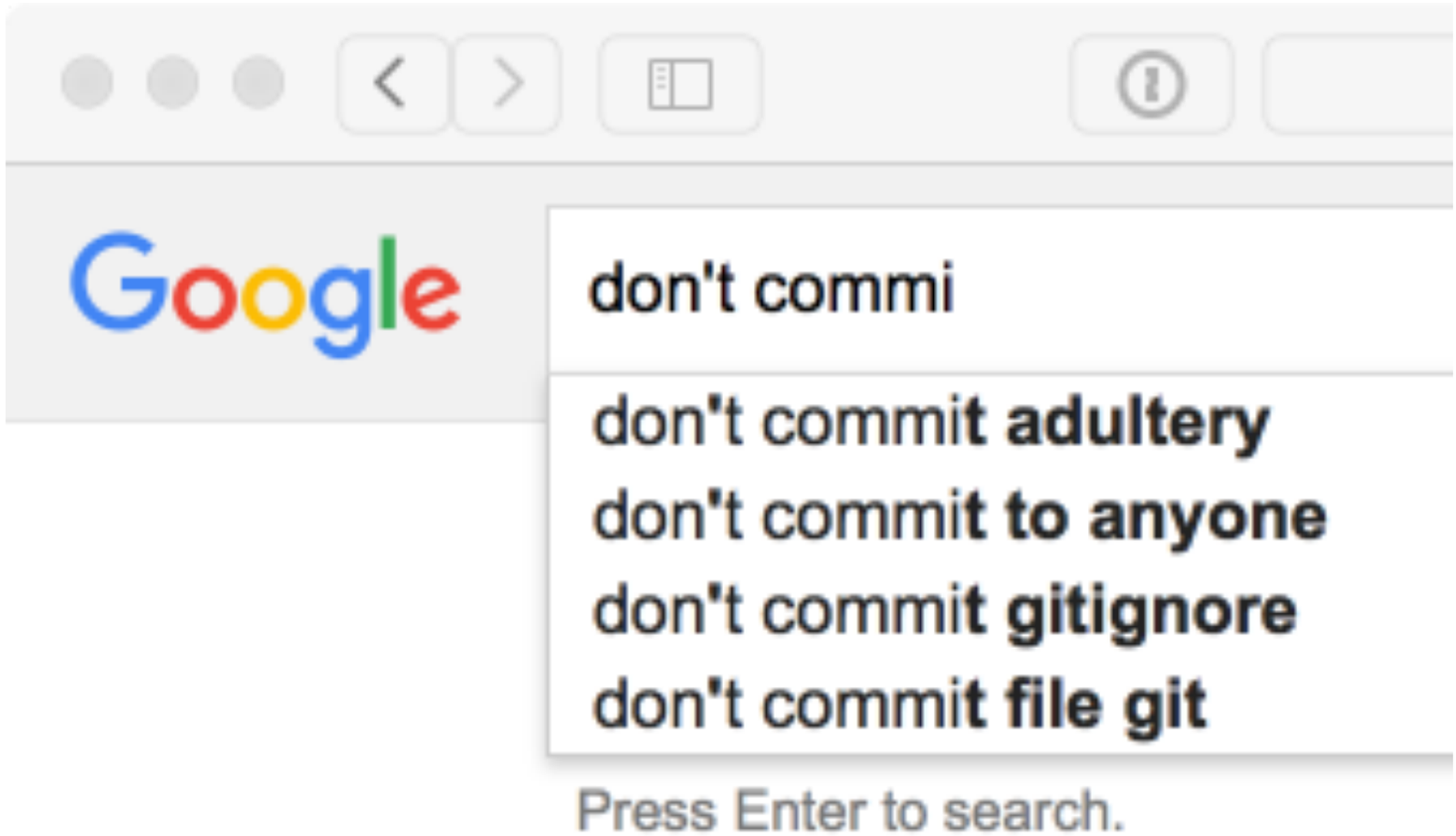
The unreasonable effectiveness of GitHub browsability

The unreasonable effectiveness of GitHub browsability

- Be savvy about your files
- Get over your hang ups re: committing derived products
- Markdown
- `README.md`
- Finding stuff
- HTML
- Source code
- Delimited files
- PNGs
- Linking to a ZIP archive of your repo
- Links and embedded figures
- Let people correct you on the internet

One of my favorite aspects of GitHub is the ability to inspect a repository's files in a browser. Certain practices make browsing more rewarding and can postpone the day when you must create a proper website for a project. Perhaps indefinitely.

Impressive showing by Git here



Comma (.csv) and tab (.tsv) delimited files are automatically rendered nicely in GitHub repositories Example: some Lord of the Rings data

jennybc / lotr

Unwatch 1 Star 0 Fork 1

branch: master lotr / lotr_clean.tsv

jennybc 2 months ago Add early exploration/cleaning

1 contributor

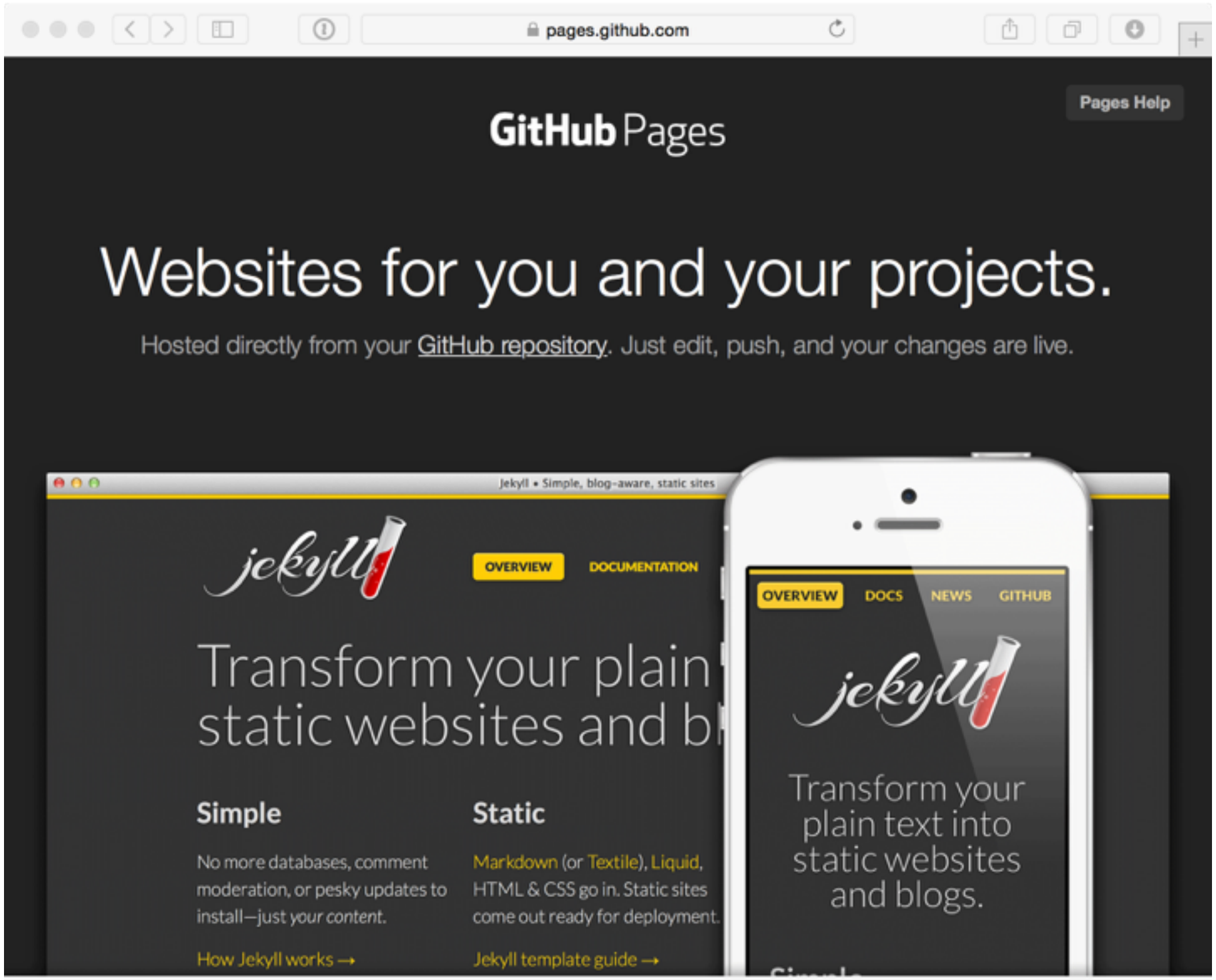
file | 684 lines (683 sloc) | 42.64 kb Open Edit Raw Blame History Delete

Search this file...

	Film	Chapter	Character	Race	Words
2	The Fellowship Of The Ring	01: Prologue	Bilbo	Hobbit	4
3	The Fellowship Of The Ring	01: Prologue	Elrond	Elf	5
4	The Fellowship Of The Ring	01: Prologue	Galadriel	Elf	460
5	The Fellowship Of The Ring	02: Concerning Hobbits	Bilbo	Hobbit	214
6	The Fellowship Of The Ring	03: The Shire	Bilbo	Hobbit	70
7	The Fellowship Of The Ring	03: The Shire	Frodo	Hobbit	128
8	The Fellowship Of The Ring	03: The Shire	Gandalf	Wizard	197
9	The Fellowship Of The Ring	03: The Shire	Hobbit Kids	Hobbit	10

When rendered Markdown is no longer enough …

<my_thing>.github.io



When we push *.html to STAT 545 repo, website updates!

STAT545-UBC/STAT545-UBC.github.io

GitHub, Inc. github.com/STAT545-UBC/STAT545-UBC.github.io

This repository Search Explore Gist Blog Help jennybc

STAT545-UBC / STAT545-UBC.github.io

Unwatch 16 Star 34

Main repository for STAT 545 @ University of British Columbia, a course in data wrangling, exploration, and analysis with R. http://stat545-ubc.github.io — Edit

495 commits 2 branches 0 releases 8 contributors

branch: master

typo/correction

jennybc authored 2 hours ago

automation01_slides

automation10_holding-area

block002_hello-r-workspace-

Code

Issues

Pull Requests

stat545-ubc.github.io

STAT 545

Home FAQ Syllabus Topics People

Data wrangling, exploration, and analysis with R

UBC STAT 545A and 547M

Learn how to

- explore, groom, visualize, and analyze data
- make all of that reproducible, reusable, and shareable
- using R

Selected topics

- Introduction to R and the RStudio IDE; scripts, the workspace, RStudio Projects
- Generate reports from R scripts and R Markdown
- Care and feeding of data in R
- Data aggregation; "apply" functions, `plyr`, `dplyr`
- Data visualization with `ggplot2`
- Graphs and descriptive stats for quantitative and categorical variables

the poor woman's regression test of a data analysis

Your code's the same
Your data's the same
But you updated R + pkgs
Surprise!

GitHub, Inc. github.com/jennybc/gapminder/commit/d966f6

This repository Search Pull requests Issues Gist

jennybc / gapminder

Unwatch 9 Star 51 Fork 27

Code Issues 4 Pull requests 0 Wiki Pulse Graphs Settings

replace gdata with readxl

master v0.2.0

Browse files

jennybc committed 17 days ago 1 parent ff65b3c commit d966f6719c2512f946e47c69e172f678b46ab7de

Showing 5 changed files with **529 additions** and **693 deletions**.

Unified Split

3 DESCRIPTION View

```
@@ -15,7 +15,8 @@ URL: https://github.com/jennybc/gapminder,
 BugReports:
https://github.com/jennybc/gapminder/issues
 Suggests:
     dplyr,
-    gdata,
     ggplot2,
+    readr,
+    readxl,
     tidyr,
     testthat
```

GitHub, Inc. github.com/jennybc/gapminder/commit/d966f6

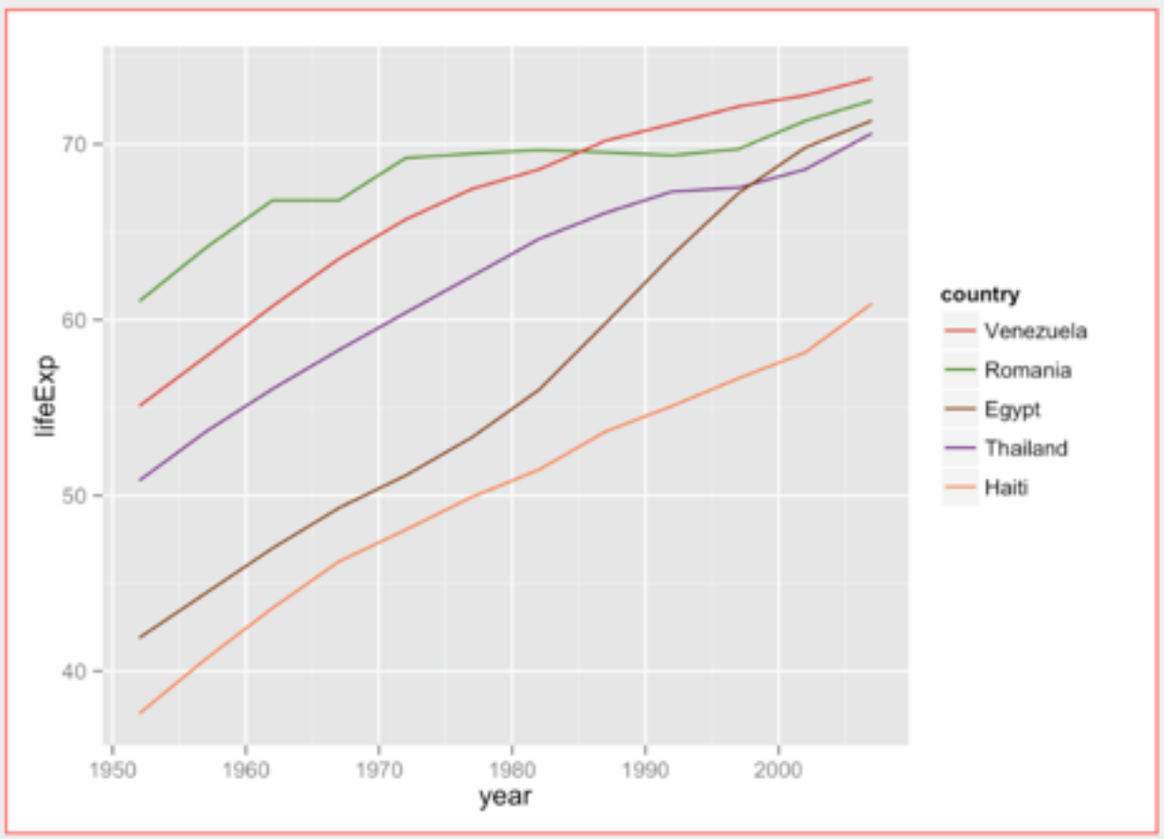
878 data-raw/02_lifeExp.tsv **View**

```
@@ -59,86 +59,86 @@ Argentina  Americas  1992  71.868
 Argentina  Americas  1997  73.275
 Argentina  Americas  2002  74.34
 Argentina  Americas  2007  75.32
-Armenia  1952  62.809
-Armenia  1957  64.928
-Armenia  1962  67.055
-Armenia  1967  69.211
-Armenia  1972  70.786
-Armenia  1977  70.595
-Armenia  1982  70.916
-Armenia  1987  68.408
-Armenia  1992  68.663
-Armenia  1997  70.377
-Armenia  2002  71.403
-Armenia  2007  71.965
-Aruba  1952  60.437
-Aruba  1957  64.381
-Aruba  1962  66.606
-Aruba  1967  68.336
-Aruba  1972  70.941
-Aruba  1977  71.83
-Aruba  1982  74.116
-Aruba  1987  74.494
-Aruba  1992  74.108
-Aruba  1997  73.011
-Aruba  2002  73.451
-Aruba  2007  74.239
-Australia  1950  69.02
-Australia  1951  68.72
+Armenia  NA  1952  62.809
+Armenia  NA  1957  64.928
+Armenia  NA  1962  67.055
+Armenia  NA  1967  69.211
+Armenia  NA  1972  70.786
+Armenia  NA  1977  70.595
+Armenia  NA  1982  70.916
+Armenia  NA  1987  68.408
+Armenia  NA  1992  68.663
+Armenia  NA  1997  70.377
+Armenia  NA  2002  71.403
+Armenia  NA  2007  71.965
+Aruba  NA  1952  60.437
+Aruba  NA  1957  64.381
+Aruba  NA  1962  66.606
+Aruba  NA  1967  68.336
+Aruba  NA  1972  70.941
+Aruba  NA  1977  71.83
+Aruba  NA  1982  74.116
+Aruba  NA  1987  74.494
+Aruba  NA  1992  74.108
+Aruba  NA  1997  73.011
+Aruba  NA  2002  73.451
+Aruba  NA  2007  74.239
+Australia  NA  1950  69.02
+Australia  NA  1951  68.72
```

subtle fig changes due to ggplot2 release

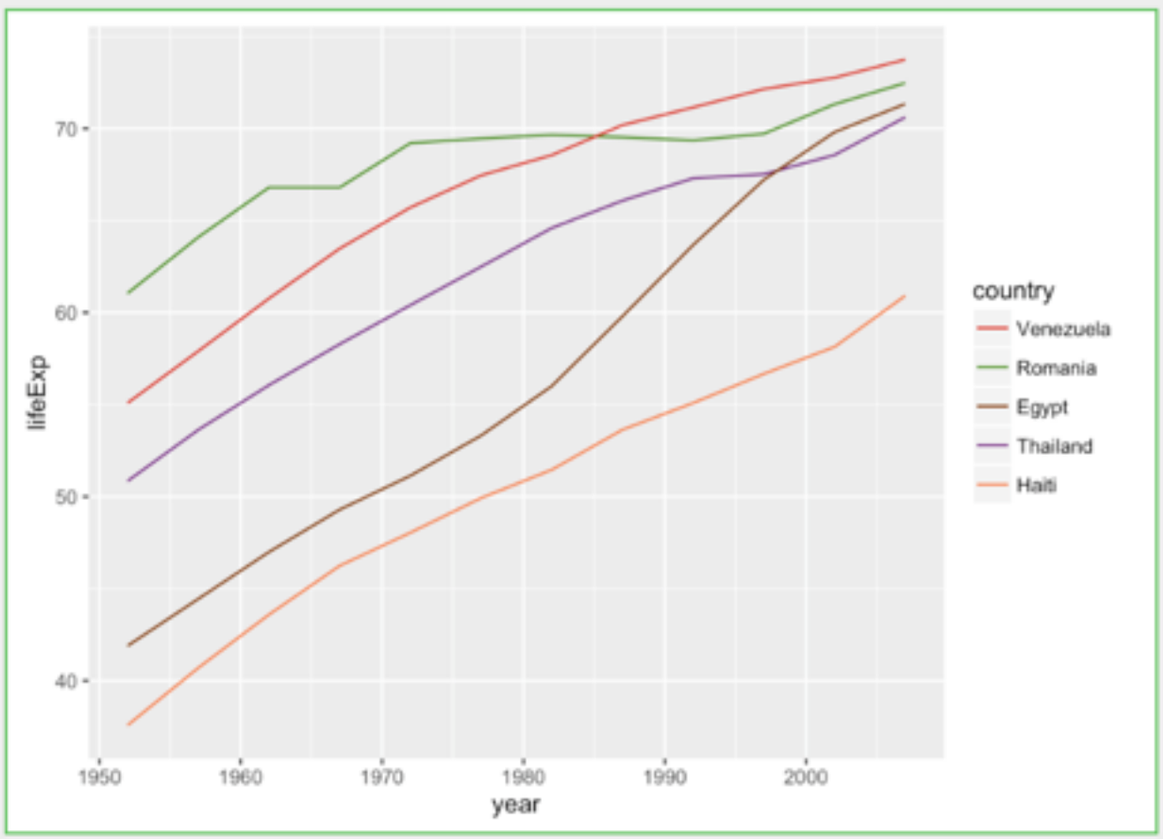
BIN ■■■■■ README_files/figure-markdown_github/demo-country-colors-ggplot2-1.png View

Deleted



W: 2100px | H: 1500px

Added



W: 2100px | H: 1500px

2-up | Swipe | Onion Skin