New tools and workflows for data analysis

Dr. Jennifer (Jenny) Bryan
Dept. of Statistics & Michael Smith Laboratories, UBC

Workshop on Visualization for Big Data @ Fields Institute

jenny@stat.ubc.ca
http://stat545-ubc.github.io
http://www.stat.ubc.ca/~jenny/

@JennyBryan
@STAT545
https://github.com/jennybc

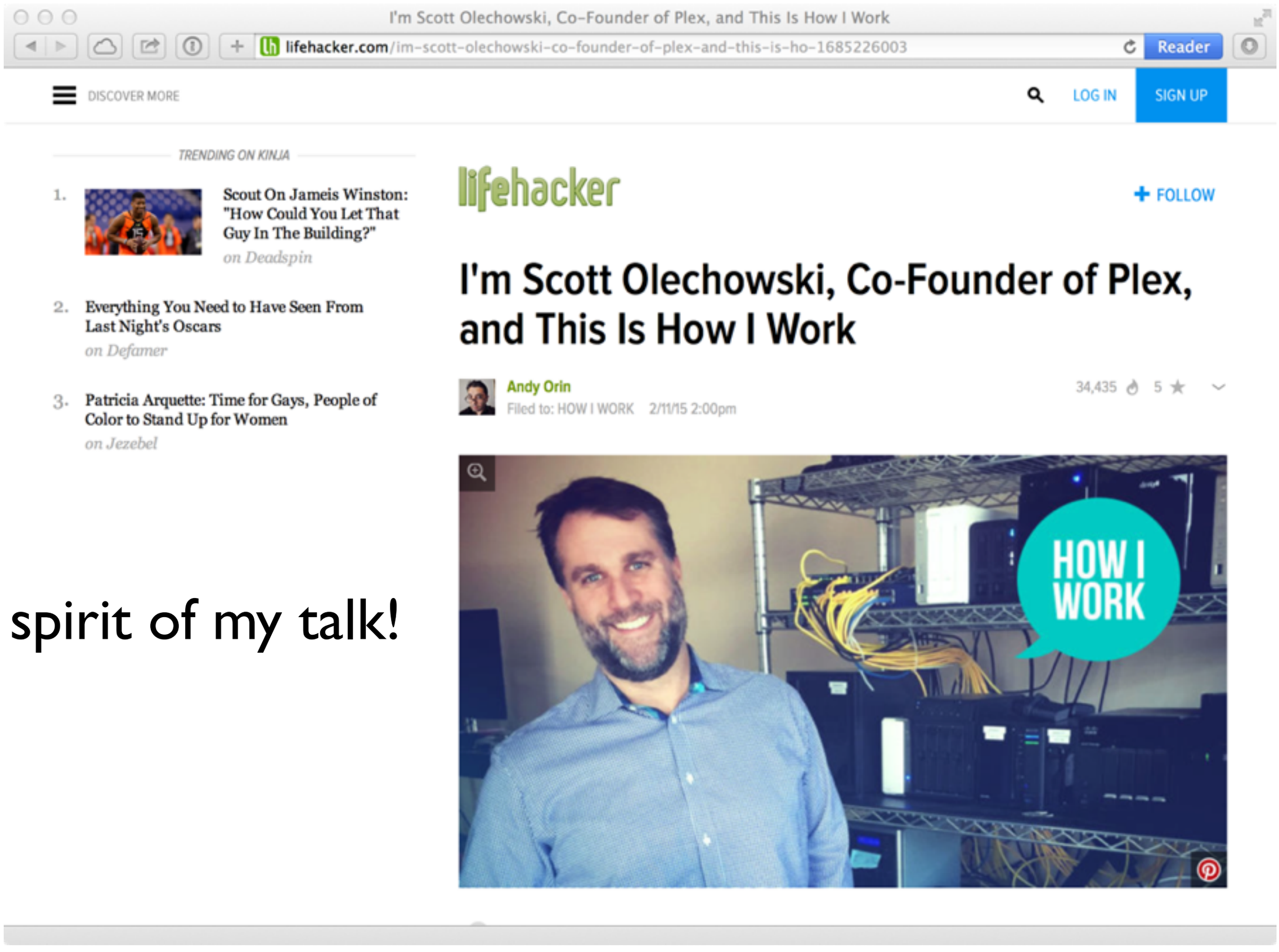
in a wide array of academic fields,

the **ability to effectively process data**

is superseding other more classical modes of research

The Big Data Brain Drain: Why Science is in Trouble
http://jakevdp.github.io/blog/2013/10/26/big-data-brain-drain/

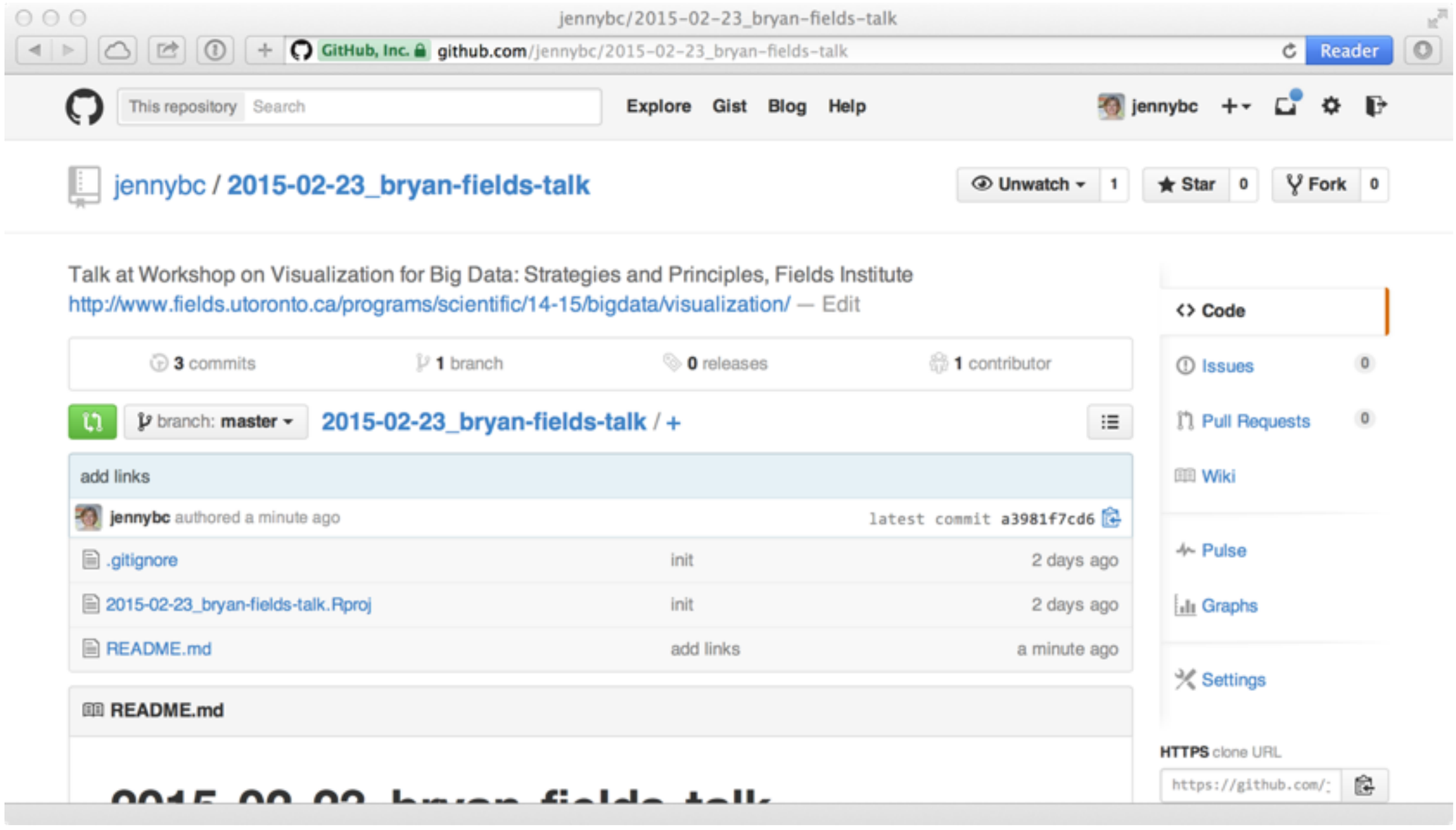
spirit of my talk!



Fact: I don't work for these companies. I don't represent them. I am not an author of these packages.

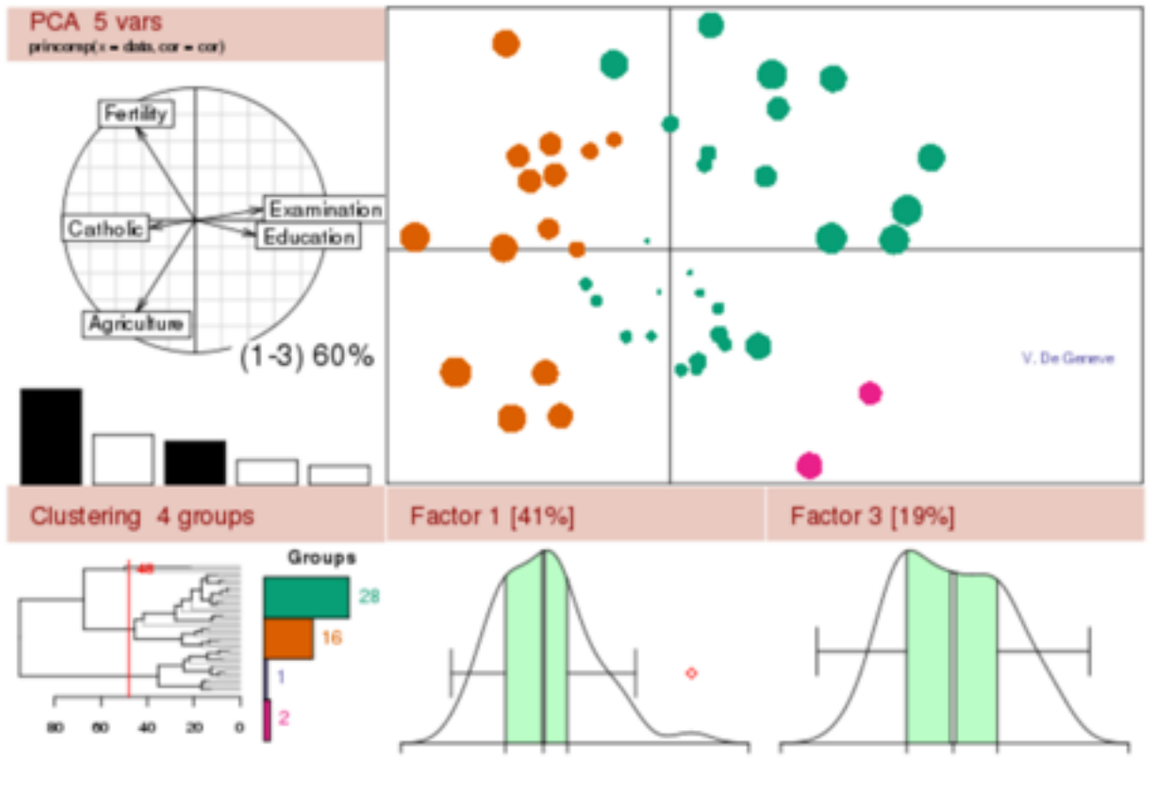
GitHub

links, files, etc. available here



https://github.com/jennybc/2015-02-23_bryan-fields-talk

The R Project for Statistical Computing



About R
What is R?
Contributors
Screenshots
What's new?

Download, Packages
CRAN

R Project
Foundation
Members & Donors
Mailing Lists
Bug Tracking
Developer Page
Conferences
Search

Documentation
Manuals
FAQs
The R Journal
Wiki
Books
Certification
Other

Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.
- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

Douglas Bates
John Chambers
Peter Dalgaard
Seth Falcon
Robert Gentleman
Kurt Hornik
Ross Ihaka
Michael Lawrence
Friedrich Leisch
Uwe Ligges
Thomas Lumley
Martin Maechler
Martin Morgan
Duncan Murdoch
Paul Murrell
Martyn Plummer
Brian Ripley
Deepayan Sarkar
Duncan Temple Lang
Luke Tierney
Simon Urbanek

Natural Sciences and Engineering Research Council of Canada | Conseil de recherches en sciences naturelles et en génie du Canada | Canada

NSERC

www.nserc-crsng.gc.ca

Contact Us | Help

eConsole >

Main Menu
Logout

Proactive Disclosure
Proactive Disclosure

eConsole

Version 5.52

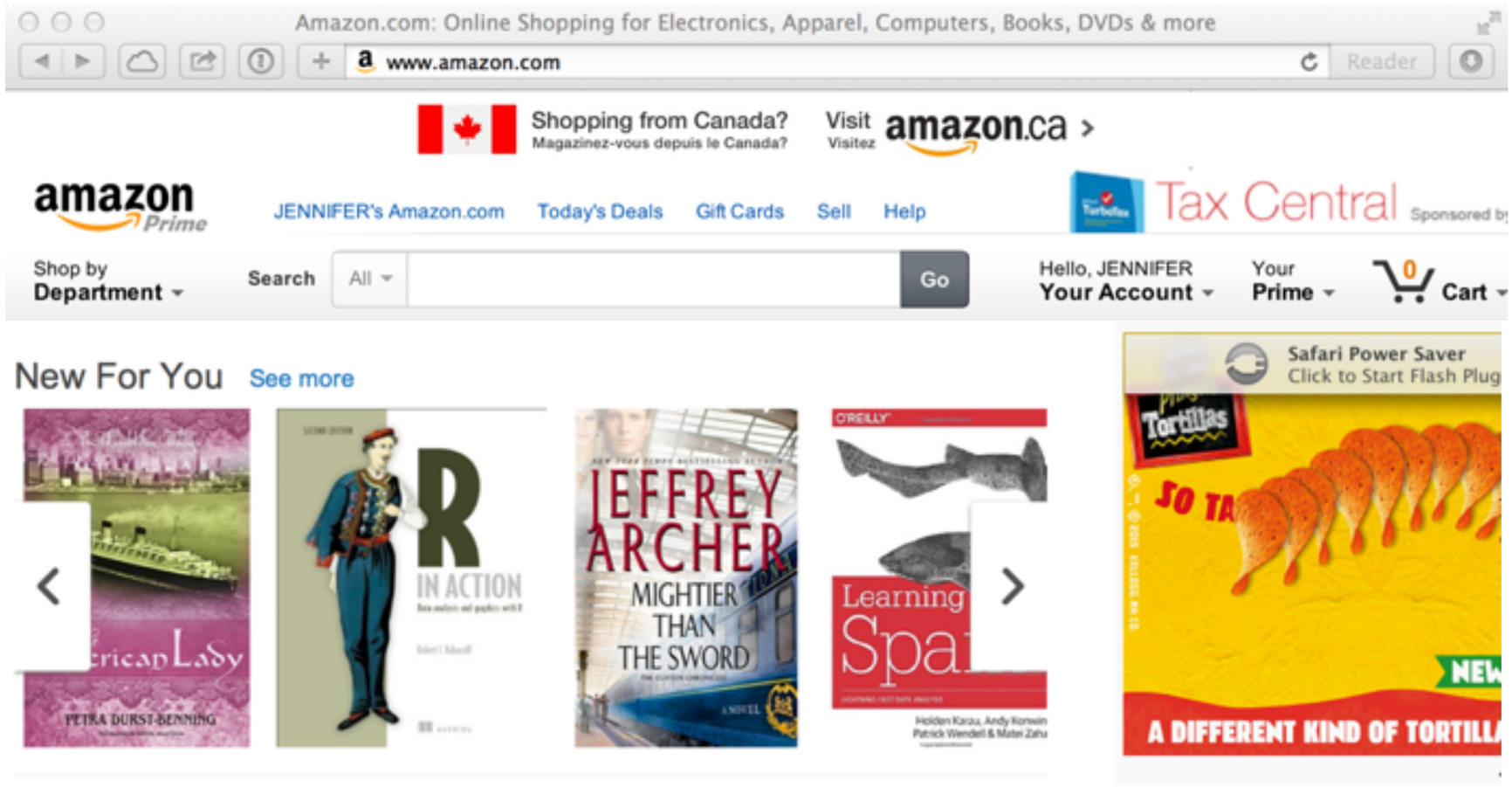
Welcome Jennifer Bryan

Users of the eSubmission system will no longer be required to periodically change their passwords. However, for their own protection, users are encouraged to change their passwords regularly. Keep your password safe and confidential; do not divulge it to anyone. NSERC will not be held liable for any loss of your data should you neglect to protect your password.

Account Management
- Change Password
- Maintain User Profile

Forms Management
- Forms - Researcher
- Forms - Student
- Forms - Reviewer
- Forms - Partners
- Forms - Department Head

differences in the user experience?



real
world
data

statistical
theory

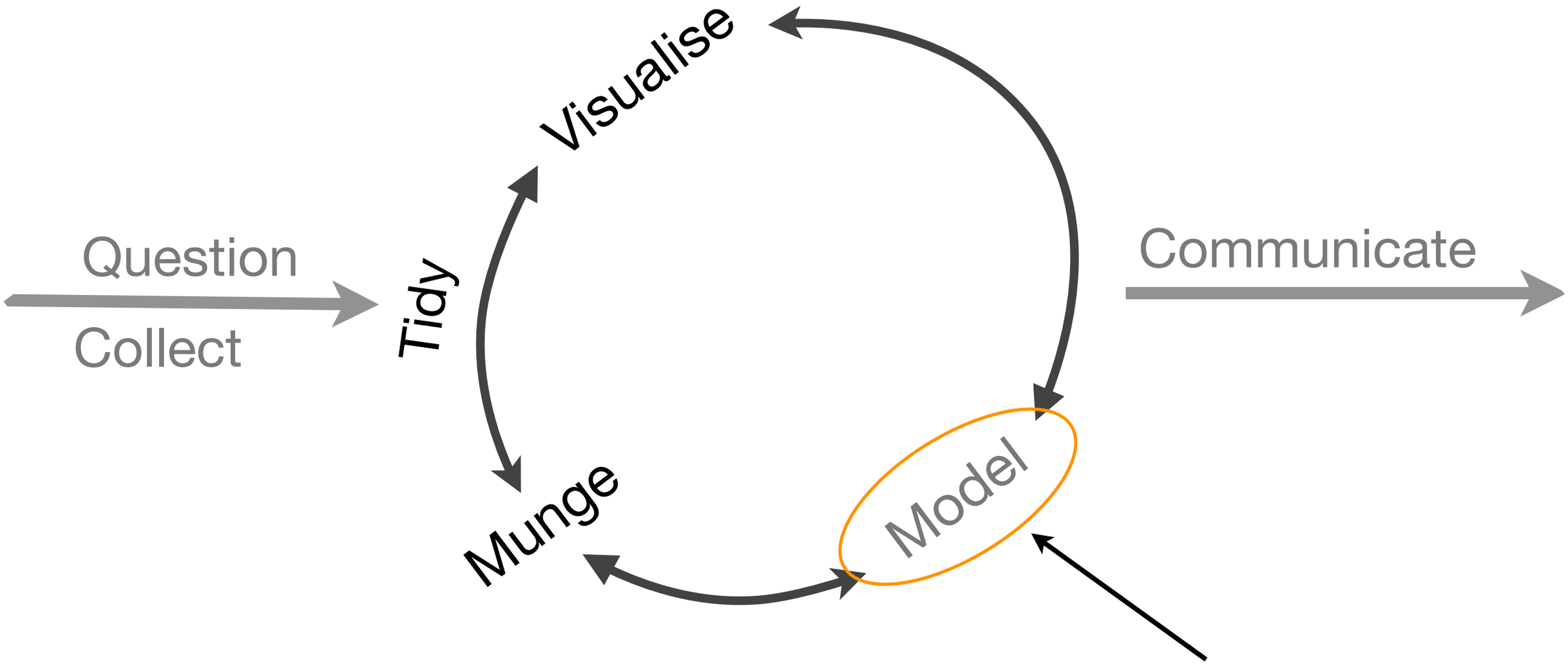
Data wrangling, exploration, and analysis with R

UBC STAT 545A and 547M

Learn how to

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

http://stat545-ubc.github.io



We can't focus just on this!

Slides from Hadley Wickham's talk in the Simply Statistics Unconference http://t.co/D931Og8mq3

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

By **STEVE LOHR** AUG. 17, 2014

Monica Rogati, Jawbone's vice president for data science, with Brian Wilt, a senior data scientist.
Peter DaSilva for The New York Times

Data scientists spend 50 - 80% of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

nytimes 2014-08-18

http://mimno.infosci.cornell.edu/b/articles/carpentry/

Data carpentry

WRITTEN BY DAVID MIMNO

The New York Times has an article titled For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights. Mostly I really like it. The fact that raw data is rarely usable for analysis without significant work is a point I try hard to make with my students. I told them "do not underestimate the difficulty of data preparation". When they turned in their projects, many of them reported that they had underestimated the difficulty of data preparation. Recognizing this as a hard problem is great.

What I'm less thrilled about is calling this "janitor work". For one thing, it's not particularly respectful of custodians, whose work I really appreciate. But it also mischaracterizes what this type of work is about. I'd like to propose a different analogy that I think fits a lot better: *data carpentry*.

Note: data carpentry seems to already be a thing

Why is woodworking a better analogy? The article uses a few other terms, like data wrangling (data as unruly beasts to be tamed?) and munging (what is that, anyway?), neither of which mean much to me. I also like *data curation* but that's also

"data science is 'just' statistics"

"data wrangling is not statistics"

if you value self-consistency, you can hold at most **one** of these opinions

Johns Hopkins University

Data Science

A Specialization on Coursera: Your Pathway to Expertise

Final Capstone Project created with:

SwiftKey

GENERAL ASSEMBLY

‹ View all in Data Analysis

DATA SCIENCE

AN 11-WEEK TECHNOLOGY COURSE SUPPORTED BY

MedStar Health

We cannot expect anyone to know anything we didn't teach them ourselves.

Sarah Bryce

To a very great degree, daily work by other people sounds easy -- certainly easier that what we have to do.

Gretchen Rubin

STAT 545 **now**

~~permission~~ requirement to invest time in setting up tools and to develop proficiency

"simple" descriptive stats
exploration through visualization

tame data from "the wild"

alpha to omega: raw data to a web page/app

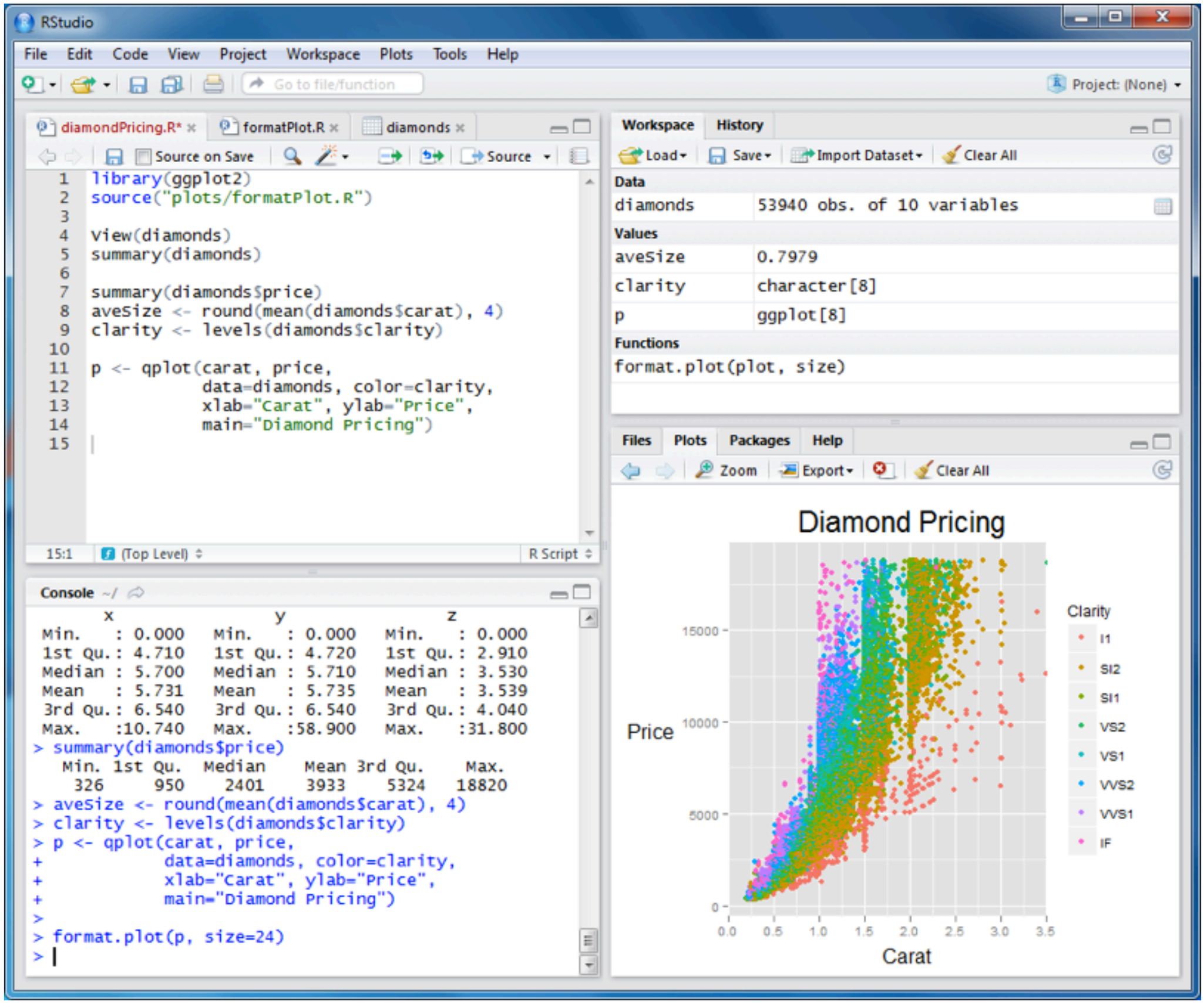
readiness for open science and automation

how to organize your work?
how to make work more pleasant for you?
how to make it navigable by others?
how to reduce tedium and manual processes?
how to reduce friction for collaboration?
how to reduce friction for communication?

specific tools and habits can build alot of this into the normal coding and analysis process

weak links in the chain: process, packaging and presentation

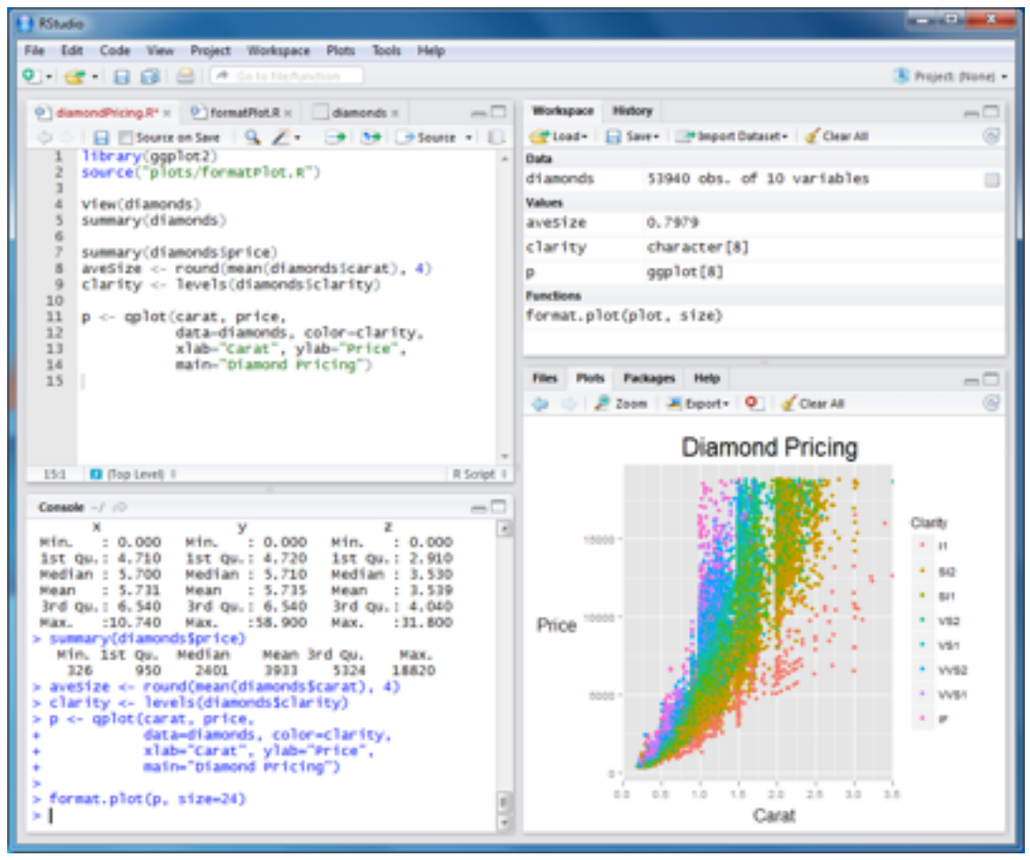
RStudio is an integrated development environment (IDE) for R



R ≠ RStudio

RStudio mediates your interaction with R; it would replace Emacs + ESS or Tinn-R, but not R itself

Rstudio is a product of -- actually, more a driver of -- the emergence of R Markdown, `knitr`, R + Git(Hub)



markdown

What is Markdown?

- Markdown is a lightweight markup language for creating HTML (or XHTML) documents.
- Markup languages are designed produce documents from human readable text (and annotations).
- Some of you may be familiar with *LaTeX*. This is another (less human friendly) markup language for creating pdf documents.
- Why I love Markdown:
  - Easy to learn and use.
  - Focus on **content**, rather than **coding** and debugging **errors**.
  - Once you have the basics down, you can get fancy and add HTML, JavaScript & CSS.

http://cpsievert.github.io/slides/markdown/#/5

Markdown → HTML

foo.md → foo.html

easy to write
(and read!)

easy to publish
easy to read in
browser

Markdown → HTML

```
Title (header 1, actually)
==================================

This is a Markdown document.

## Medium header (header 2, actually)

It's easy to do *italics* or __make things bold__.

> All models are wrong, but some are useful. An
approximate answer to the right problem is worth a
good deal more than an exact answer to an
approximate problem. Absolute certainty is a
privilege of uneducated minds-and fanatics. It is,
for scientific folk, an
you do every day matter
once in a while. We can
anything we didn't teac
Enthusiasm is a form of

Code block below. Just
we'll get to R Markdown

```
x <- 3 * 4
```

I can haz equations. Inline equations, such as ...
the average is computed as $\frac{1}{n} \sum_{i=1}
^{n} x_{i}$. Or display equations like this:

$$
\begin{equation*}
|x|=
\begin{cases} x & \text{if $x≥0$,} \\\\
-x &\text{if $x\le 0$.}
\end{cases}
\end{equation*}
$$
```

Fess up: How many of you still hand-code HTML?

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8"/>

<title>Title (header 1, actually)</title>

<!-- MathJax scripts -->
<script type="text/javascript" src="https://
c328740.ssl.cf1.rackcdn.com/mathjax/2.0-latest/
MathJax.js?config=TeX-AMS-MML_HTMLorMML">
</script>

                                    rial, sans-serif;

                                    y)</h1>

<p>This is a Markdown document.</p>

<h2>Medium header (header 2, actually)</h2>

<p>It&#39;s easy to do <em>italics</em> or
<strong>make things bold</strong>.</p>

<blockquote>
<p>All models are wrong, but some are...
<p>Code block below. Just affects formatting here
but we&#39;ll get to R Markdown for the real fun
soon!</p>

<pre><code>x &lt;- 3 * 4
</code></pre>
```

Markdown → HTML

```
Title (header 1, actually)
==================================

This is a Markdown document.

## Medium header (header 2, actually)

It's easy to do *italics* or __make things bold__.

> All models are wrong, but some are useful. An
approximate answer to the right problem is worth a
good deal more than an exact answer to an
approximate problem. Absolute certainty is a
privilege of uneducated minds-and fanatics. It is,
for scientific folk, an unattainable ideal. What
you do every day matters more than what you do
once in a while. We cannot expect anyone to know
anything we didn't teach them ourselves.
Enthusiasm is a form of social courage.

Code block below. Just affects formatting here but
we'll get to R Markdown for the real fun soon!

```
x <- 3 * 4
```

I can haz equations. Inline equations, such as ...
the average is computed as $\frac{1}{n} \sum_{i=1}
^{n} x_{i}$. Or display equations like this:

$$
\begin{equation*}
|x|=
\begin{cases} x & \text{if $x≥0$,} \\\\
-x &\text{if $x\le 0$.}
\end{cases}
\end{equation*}
$$
```

Title (header 1, actually)

This is a Markdown document.

Medium header (header 2, actually)

It's easy to do *italics* or **make things bold**.

> All models are wrong, but some are useful. An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem. Absolute certainty is a privilege of uneducated minds-and fanatics. It is, for scientific folk, an unattainable ideal. What you do every day matters more than what you do once in a while. We cannot expect anyone to know anything we didn't teach them ourselves. Enthusiasm is a form of social courage.

Code block below. Just affects formatting here but we'll get to R Markdown for the real fun soon!

```
x <- 3 * 4
```

I can haz equations. Inline equations, such as ... the average is computed as $\frac{1}{n} \sum_{i=1}^{n} x_i$. Or display equations like this:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$

How is the math getting typeset?

Answer: Mathjax

MathJax is an open source JavaScript display engine for mathematics that works in all browsers.

No more setup for readers. No more browser plugins. No more font installations... It just works.

How painful is that to use?

Not at all. Automagic with `knitr` and RStudio.

What happens to equations if the reader is not connected to the internet?

The LaTeX is displayed. No great harm.

Code block below. Just affects formatting here but we'll get to R Markdown for the real fun soon!

```
x <- 3 * 4
```

I can haz equations. Inline equations, such as … the average is computed as \(\frac{1}{n} \sum_{i=1}^{n} x_{i}\). Or display equations like this:

\[\begin{equation*} |x|= \begin{cases} x & \text{if $x≥0$,} \\ -x &\text{if $x\le 0$.} \end{cases} \end{equation*} \]

If I use Markdown, am I restricted to HTML output?

No.

pandoc = "swiss-army knife" of document conversion

(RStudio will gladly install and invoke for you.)

About pandoc

If you need to convert files from one markup format into another, pandoc is your swiss-army knife. Pandoc can convert documents in markdown, reStructuredText, textile, HTML, DocBook, LaTeX, MediaWiki markup, OPML, or Haddock markup to

- HTML formats: XHTML, HTML5, and HTML slide shows using Slidy, reveal.js, Slideous, S5, or DZSlides.
- Word processor formats: Microsoft Word docx, OpenOffice/LibreOffice ODT, OpenDocument XML
- Ebooks: EPUB version 2 or 3, FictionBook2
- Documentation formats: DocBook, GNU TexInfo, Groff man pages, Haddock markup
- Outline formats: OPML
- TeX formats: LaTeX, ConTeXt, LaTeX Beamer slides
- PDF via LaTeX
- Lightweight markup formats: Markdown, reStructuredText, AsciiDoc, MediaWiki markup, Emacs Org-Mode, Textile
- Custom formats: custom writers can be written in lua.

If you have an annoying process for authoring for the web ....

or

If you avoid authoring for the web, because you're not sure how ...

start writing in Markdown.

R markdown

R Markdown

````
R Markdown rocks
====================================

This is an R Markdown document.

```{r}
x <- rnorm(1000)
head(x)
```

See how the R code gets executed and a
representation thereof appears in the document?
`knitr` gives you control over how to represent all
conceivable types of output. In case you care, then
average of the `r length(x)` random normal variates
we just generated is `r round(mean(x), 3)`. Those
numbers are NOT hard-wired but are computed on-the-
fly. As is this figure. No more copy-paste ... copy-
paste ... oops forgot to copy-paste.

```{r}
plot(density(x))
```

Note that all the previously demonstrated math
typesetting still works. You don't have to choose
between having math cred and being web-friendly!

Inline equations, such as ... the average is
computed as $\frac{1}{n} \sum_{i=1}^{n} x_{i}$. Or
display equations like this:

$$
\begin{equation*}
|x|=
\begin{cases} x & \text{if $x≥0$,} \\\\
-x &\text{if $x\le 0$.}
\end{cases}
\end{equation*}
$$
````

Markdown

````
R Markdown rocks
====================================

This is an R Markdown document.


```
r
x <- rnorm(1000)
head(x)
```


```
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973
2.4157
```


See how the R code gets executed and a
representation thereof appears in the document?
`knitr` gives you control over how to represent all
conceivable types of output. In case you care, then
average of the 1000 random normal variates we just
generated is -0.081. Those numbers are NOT hard-
wired but are computed on-the-fly. As is this
figure. No more copy-paste ... copy-paste ... oops
forgot to copy-paste.


```
r
plot(density(x))
```


![plot of chunk unnamed-chunk-2](figure/unnamed-
chunk-2.png)

...
````

Markdown ⟶ HTML

````
R Markdown rocks
================================

This is an R Markdown document.


```r
x <- rnorm(1000)
head(x)
```


```
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973
2.4157
```


See how the R code gets executed and a
representation thereof appears in the document?
`knitr` gives you control over how to represent all
conceivable types of output. In case you care, then
average of the 1000 random normal variates we just
generated is -0.081. Those numbers are NOT hard-
wired but are computed on-the-fly. As is this
figure. No more copy-paste ... copy-paste ... oops
forgot to copy-paste.


```r
plot(density(x))
```

![plot of chunk unnamed-chunk-2](figure/unnamed-
chunk-2.png)

...
````

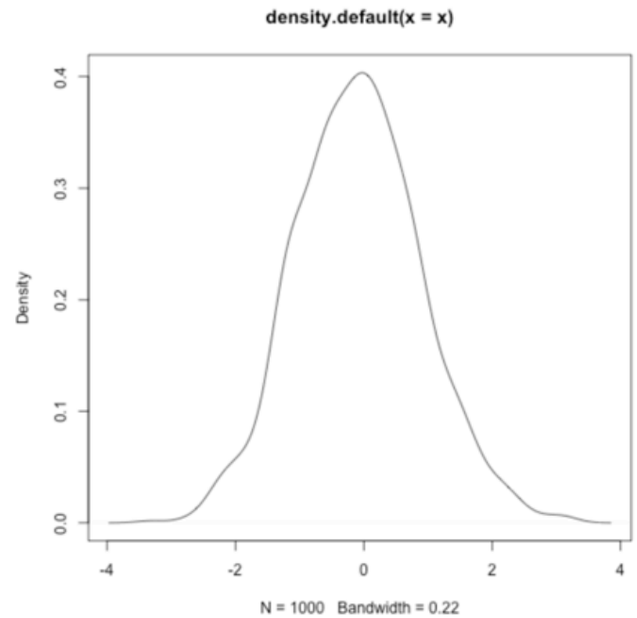
R Markdown rocks

This is an R Markdown document.

```
x <- rnorm(1000)
head(x)
```

```
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973  2.4157
```

See how the R code gets executed and a representation thereof appears in the document? `knitr` gives you control over how to represent all conceivable types of output. In case you care, then average of the 1000 random normal variates we just generated is -0.081. Those numbers are NOT hard-wired but are computed on-the-fly. As is this figure. No more copy-paste … copy-paste … oops forgot to copy-paste.



Note that all the previously demonstrated math typesetting still works. You don't have to choose between having math cred and being web-friendly!

Inline equations, such as … the average is computed as $\frac{1}{n}\sum_{i=1}^{n} x_i$. Or display equations like this:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$

R Markdown → Markdown → HTML

foo.rmd → foo.md → foo.html

easy to write
(and read!)

easy to publish
easy to read in
browser

How do to actually convert Markdown to HTML?

`knitr`, `rmarkdown` add-on packages provide user-friendly functions

RStudio makes them available via button

R Markdown → HTML

````
R Markdown rocks
====================================

This is an R Markdown document.

```{r}
x <- rnorm(1000)
head(x)
```

See how the R code gets executed and a
repres
`knitr
concei
averag
we jus
number
fly. A
paste

```{r}
plot(d
```

Note t
typese
betwee

Inline
comput
displa

$$
\begin
|x|=
\begin
-x &\text{if $x\le 0$.}
\end{cases}
\end{equation*}
$$
````

R Markdown rocks

This is an R Markdown document.

```
x <- rnorm(1000)
head(x)
```

```
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973  2.4157
```

See how the R code gets executed and a representation thereof appears in the

How to achieve at the command line:

```
> library("rmarkdown")
> render("foo.Rmd")
```

equations like this:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$

R Markdown ⟶ HTML

```
R Markdown rocks
================================

This is an R Markdown document.

```{r}
x
he
```

Se
re
`k
co
av
we
nu
fl
pa

```
\begin{cases} x & \text{if $x≥0$,} \\\\
-x &\text{if $x\le 0$.}
\end{cases}
\end{equation*}
$$
```

R Markdown rocks

This is an R Markdown document.

```
x <- rnorm(1000)
```

~/tmp/test – RStudio

Go to file/function

test

foo.Rmd

Knit HTML | Run | Chunks

```
1  ---
2  title: "Untitled"
3  author: "Jenny Bryan"
4  date: "22 February, 2015"
5  output: html_document
6  ---
7
8  This is an R Markdown document. Markdown is a simple
   formatting syntax for authoring HTML, PDF, and MS
```

1:1 (Top Level) R Markdown

Click here.

Environment | History

To Console | To Source

```
library("rmarkdown")
render("test.Rmd")
```

Files | Plots | Help | Viewer

New Folder | Delete | Rename | More

Home › tmp › test

	Name	Size	Modified
	..		
	test.Rproj	257 B	Feb 22, 2015, 9:47 PM
	foo.Rmd	723 B	Feb 22, 2015, 9:47 PM

Console ~/tmp/test/

```
'citation()' on how to cite R or R packages in publications
.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library("rmarkdown")
```

Inline equations, such as … the average is computed as $\frac{1}{n}\sum_{i=1}^{n} x_i$. Or display equations like this:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$
```

Do I have to do everything in R markdown? What about plain R scripts?

Use `rmarkdown::render()` or Rstudio's Compile Notebook button to get a satisfying stand-alone webpage based on an R script.

simple R script:
`toyline.R`

```r
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))

plot(x, y)
abline(a, b, col = "blue", lwd = 2)
```

→ HTML

toyline.R

jenny — *Sep 6, 2013, 3:15 PM*

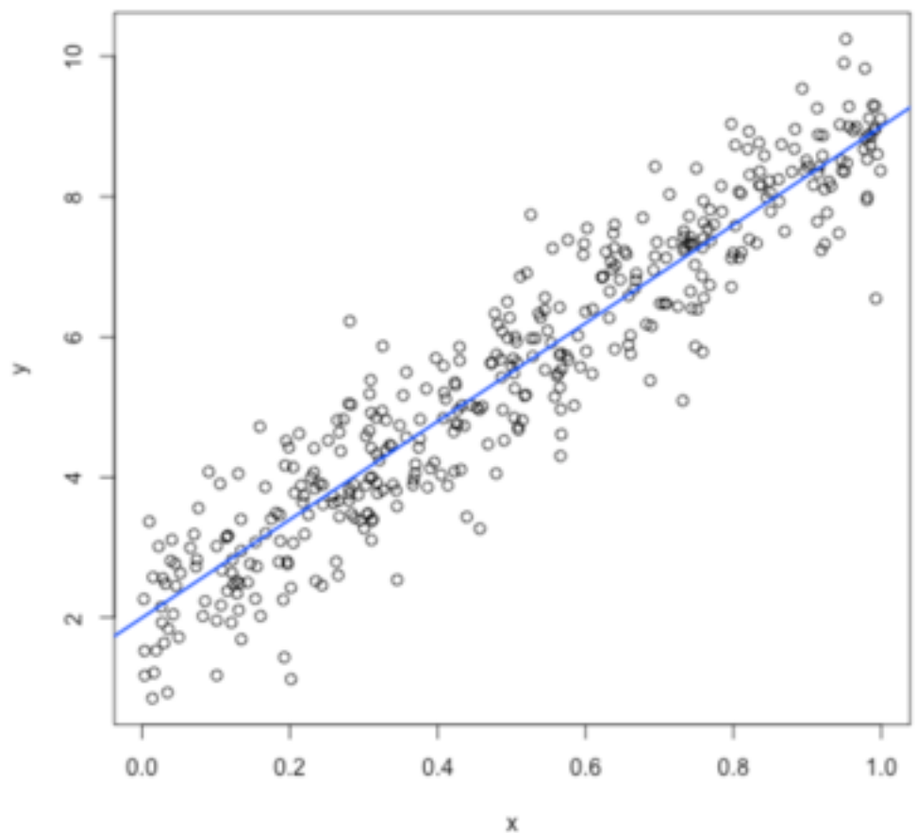
```r
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))
```

```
[1] 0.4969
```

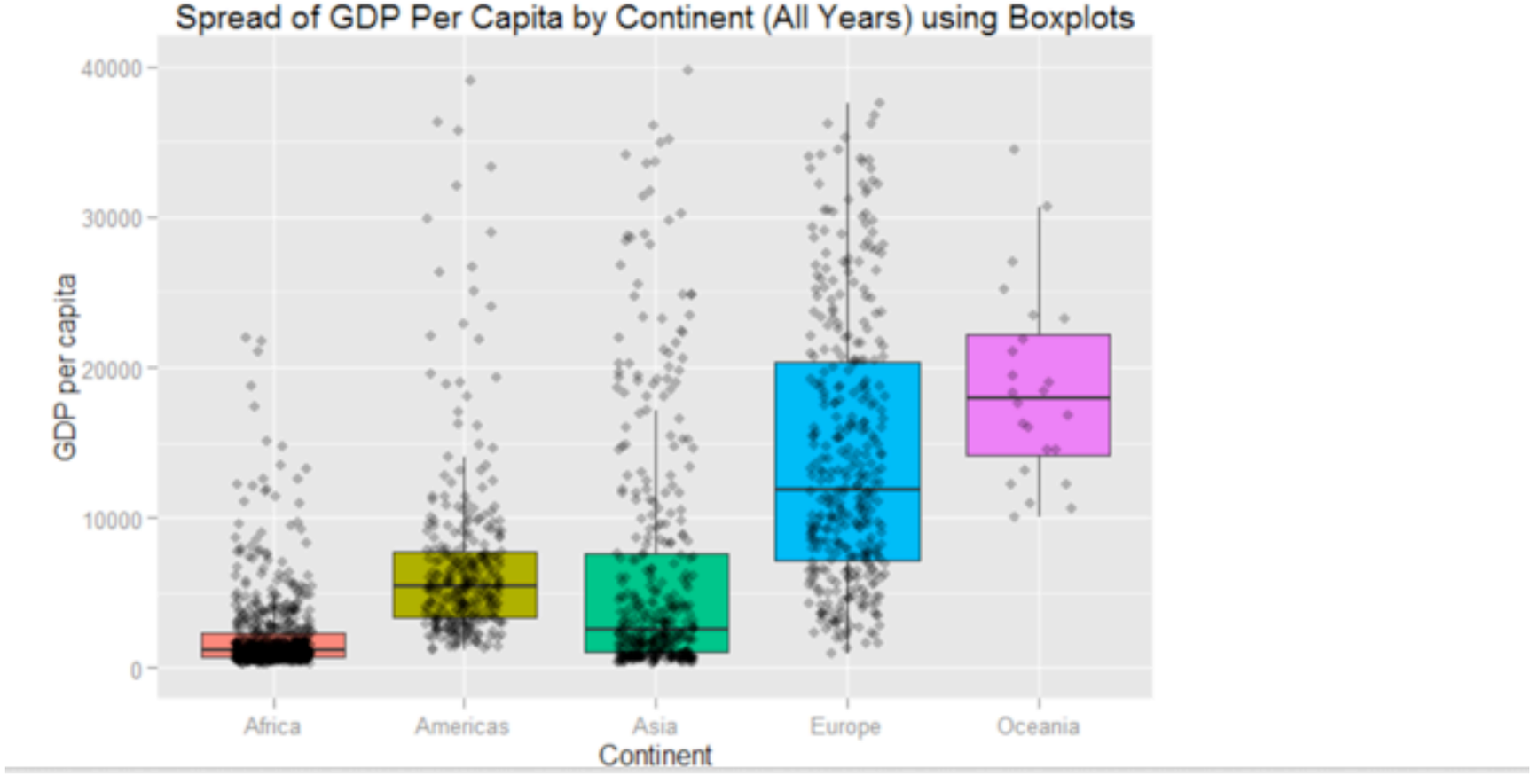
```r
plot(x, y)
abline(a, b, col = "blue", lwd = 2)
```



When I mark homework ... this is what I see.

In this section, we will use two similar graphs to visualize spread: the **boxplot** and the **violin plot**. Note that the original unmanipulated data frame `gtbl` will be used here.

```
ggplot(gtbl, aes(continent, gdpPercap))+
  geom_boxplot(aes(fill = continent), outlier.shape = NA)+
  geom_jitter(alpha = 0.3, position = position_jitter(width = 0.2))+
  xlab("Continent")+
  ylab("GDP per capita")+
  ylim(c(0,40000))+
  theme(legend.position = "none")+
  ggtitle("Spread of GDP Per Capita by Continent (All Years) using Boxplots")
```



R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics
Technology Innovations in Statistics Education, 8(1)

Baumer, Ben, Smith College
Cetinkaya-Rundel, Mine, Duke University
Bray, Andrew, Smith College
Loi, Linda, Smith College
Horton, Nicholas J., Amherst College

Publication Date:
2014

Permalink:
https://escholarship.org/uc/item/90b2f5xh

How do I show the world all these awesome dynamic HTML reports I'm creating?

Easiest: Rpubs

Or do whatever you usually do to get HTML on the web.

Or use GitHub ....

Big picture, so far:

web-friendly is good

various hosting platforms make it easy to share web-ready products with minimal effort

embedding analysis and logic in source document for a report is good

- huge win for reproducibility
- also excellent for communication and documentation

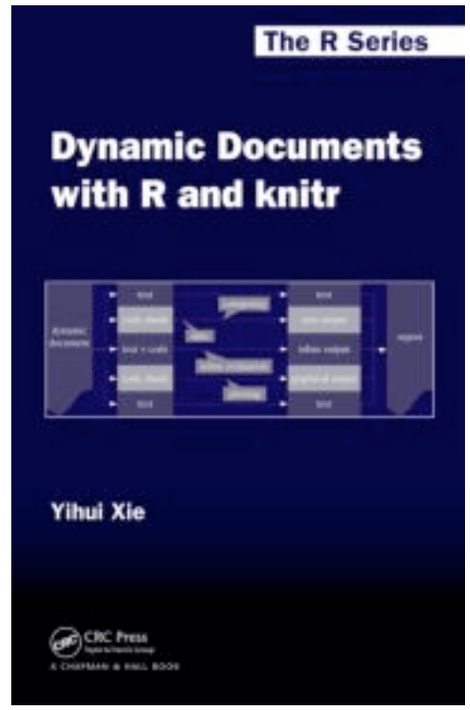
(R) Markdown + `knitr` (+ RStudio) make it very easy to author dynamic reports that are ready for the web

disclaimer:

knitr is **not limited** to executing R code
knitr is **not limited** to processing R Markdown

I just chose to focus on R and R Markdown

Read more in the book or on the web:
Dynamic documents with R and knitr by Yihui Xie, part of the CRC Press / Chapman & Hall R Series (2013). ISBN: 9781482203530.



OK you've got a collection of ...

R scripts
R package
R Markdown files
input data
intermediate results
figures
output tables
compiled reports

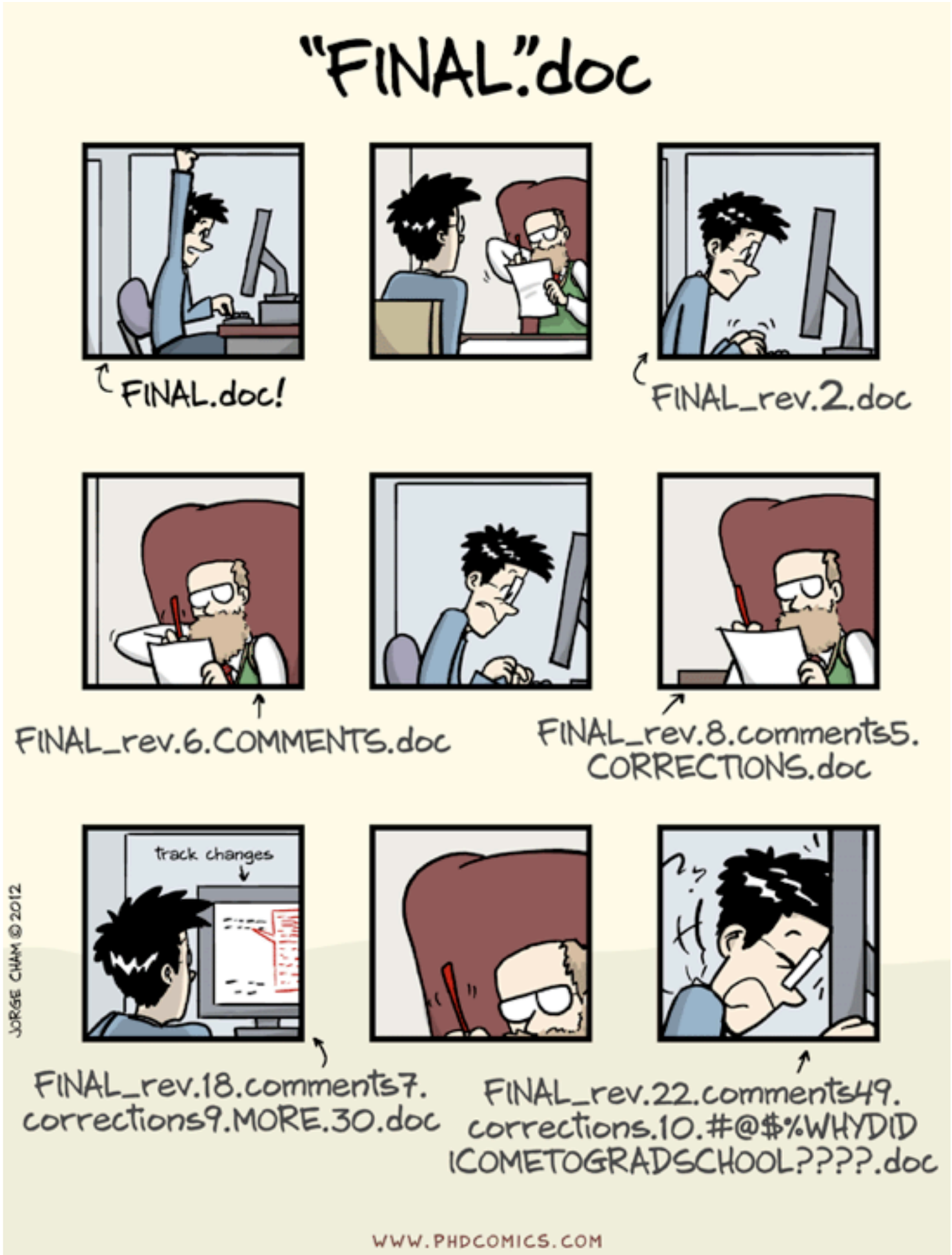
all evolving over time

how do you keep track of this?

how do I put my stuff on the web?

for the world or select collaborators?

Advice to preserve sanity:
Stop doing this via email, attachments, and tracking changes in Word. Get that stuff into plain text, put it under version control and get it out on the web.



http://www.phdcomics.com/comics/archive.php?comicid=1531 via Ram, 2013 doi:10.1186/1751-0473-8-7

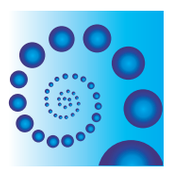
Version control systems (VCS) were created to help groups of people develop software

Git, in particular, is being "repurposed" for activities other than pure software development ... like the messy hybrid of writing, coding and data wrangling

"Git, provides a lightweight yet robust framework that is ideal for managing the full suite of research outputs such as datasets, statistical code, figures, lab notes, and manuscripts."

"... this tool can be leveraged to make science more reproducible and transparent, foster new collaborations, and support novel uses."

Ram *Source Code for Biology and Medicine* 2013, **8**:7
http://www.scfbm.org/content/8/1/7



SOURCE CODE FOR
BIOLOGY AND MEDICINE

BRIEF REPORTS Open Access

Git can facilitate greater reproducibility and increased transparency in science

Karthik Ram

GitHub repository for this paper: https://github.com/karthik/smb_git

Ram: Git can facilitate greater reproducibility and increased transparency in science. Source Code for Biology and Medicine 2013 8:7. doi:10.1186/1751-0473-8-7

collaboration = the “killer app” of version control

Learning Git has been -- and continues to be -- painful. But not nearly as crazy-making as the alternatives:

- documents as email attachments
- uncertainty about which version is “master”
- am I working with the most recent data?
- archaelogical “digs” on old email threads
- uncertainty about how/if certain changes have been made or issues solved
- hair-raising ZIP archives containing file salad

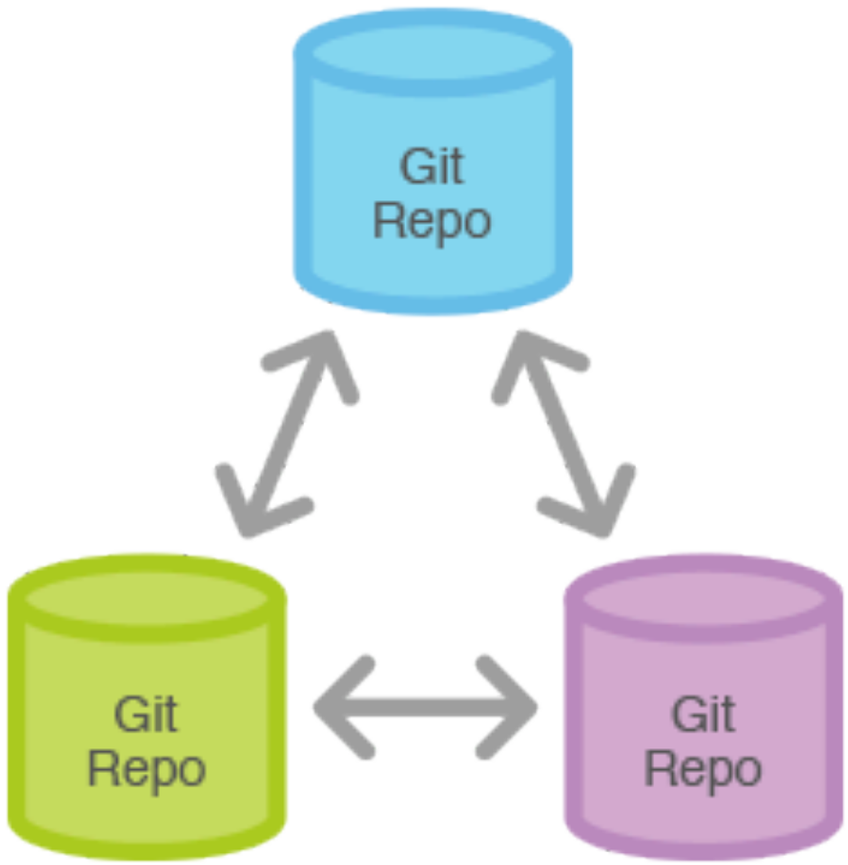
Git repository = a bunch of files you want to manage in a sane way

repo = repository

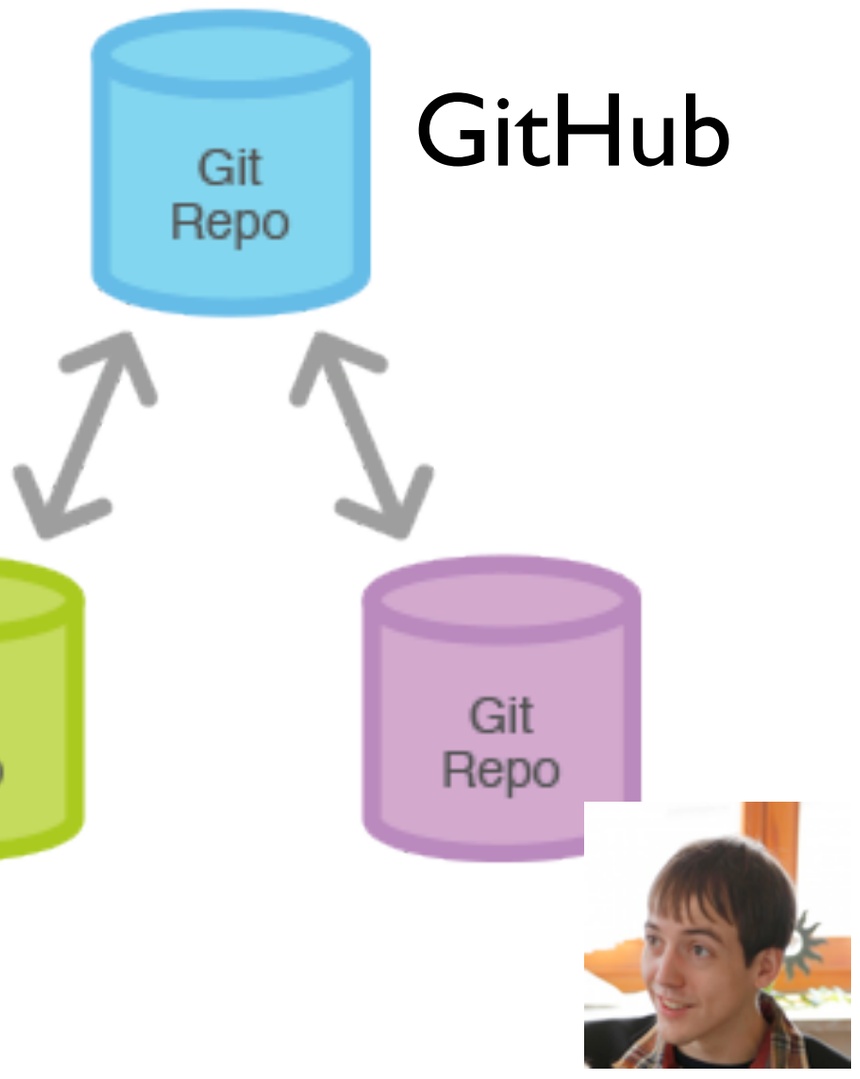
you can set up repo ... then start your work

or you can make a set of existing files and make them into a repo

in theory



more typical



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

Git server

repo

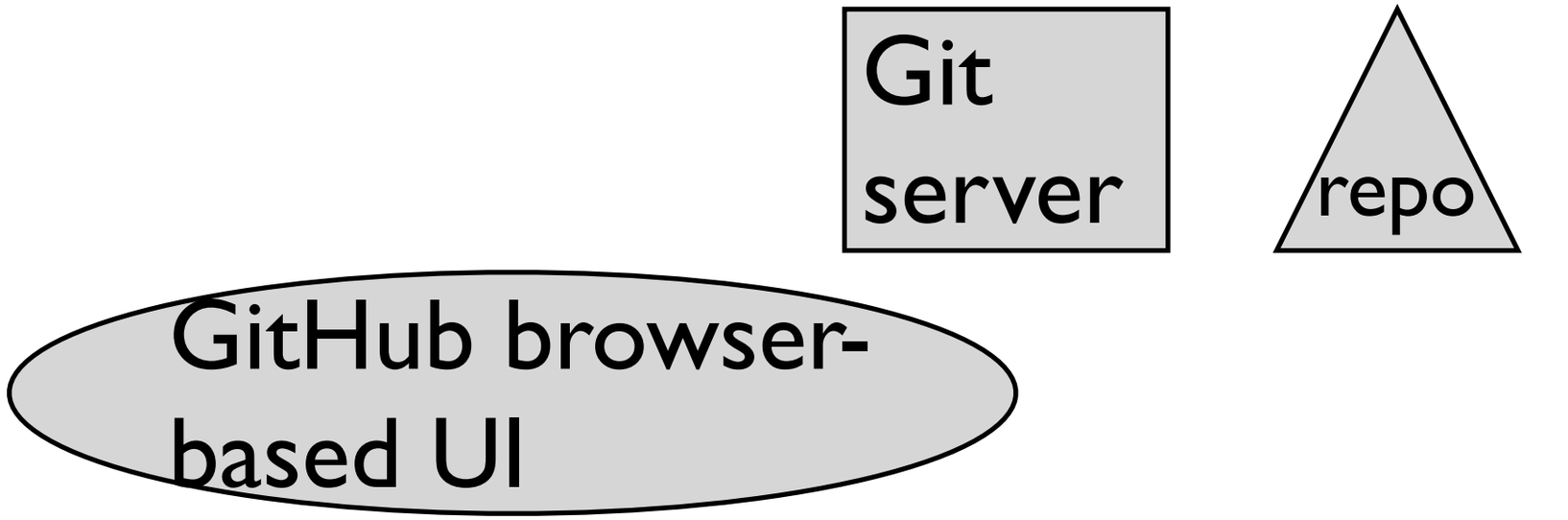
GitHub browser-based UI

Using a Git client to commit (a local operation)

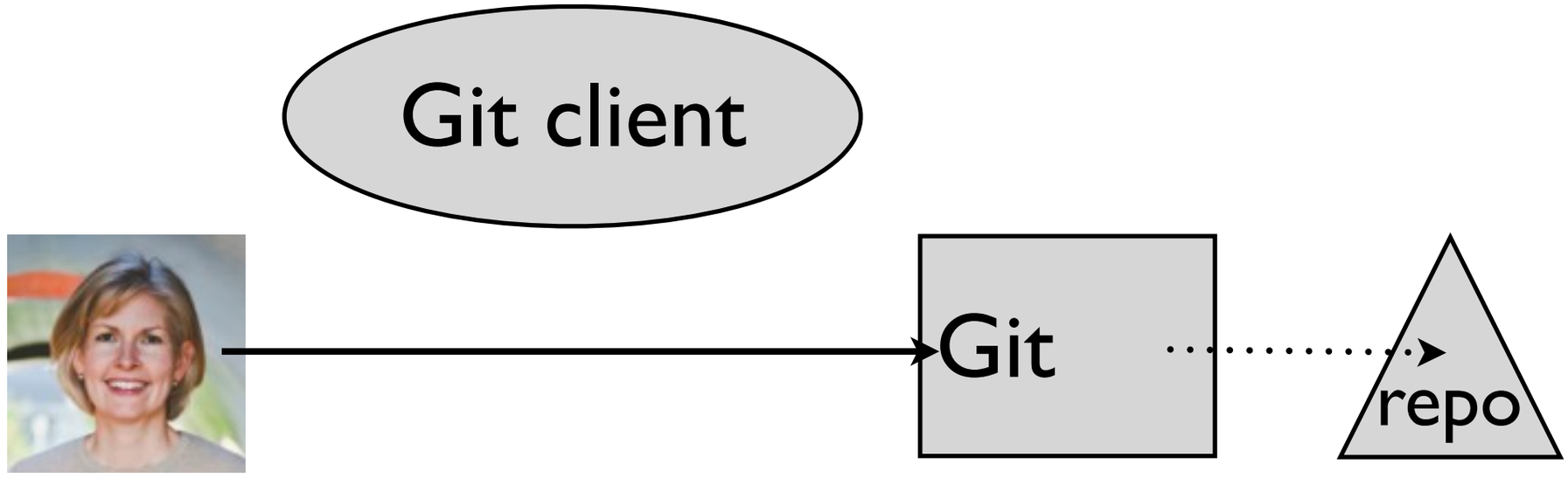
Git client

Git

repo



Using Git at the command line to commit (a local operation)



Git server

repo

GitHub browser-based UI

Using a Git client to push (local ⇒ remote)

Git client

Git

repo

Git server

repo

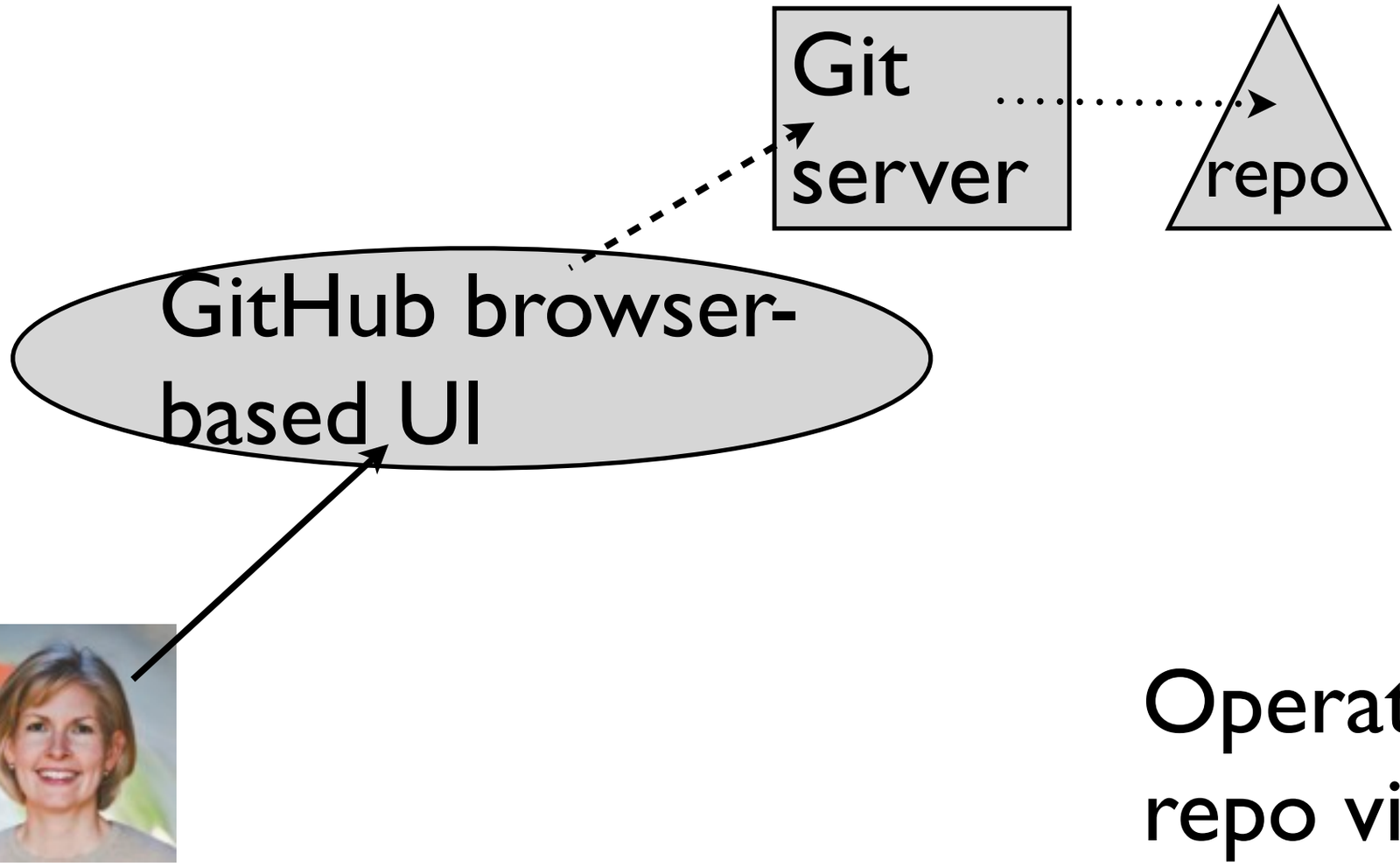
GitHub browser-based UI

Using Git at the command line to push (local ⇒ remote)

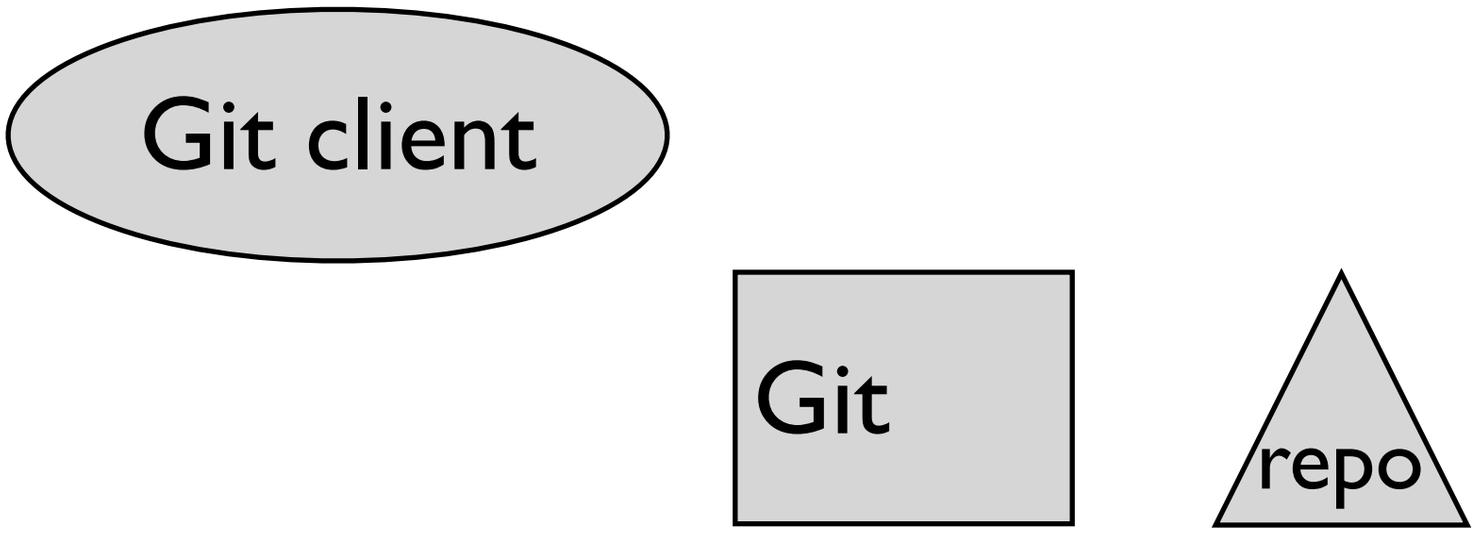
Git client

Git

repo



Operating on a Git repo via GitHub in the browser



Git server

repo

GitHub browser-based UI

Using a Git client to pull (remote ⇒ local)

Git client

Git

repo

Git server

repo

GitHub browser-based UI

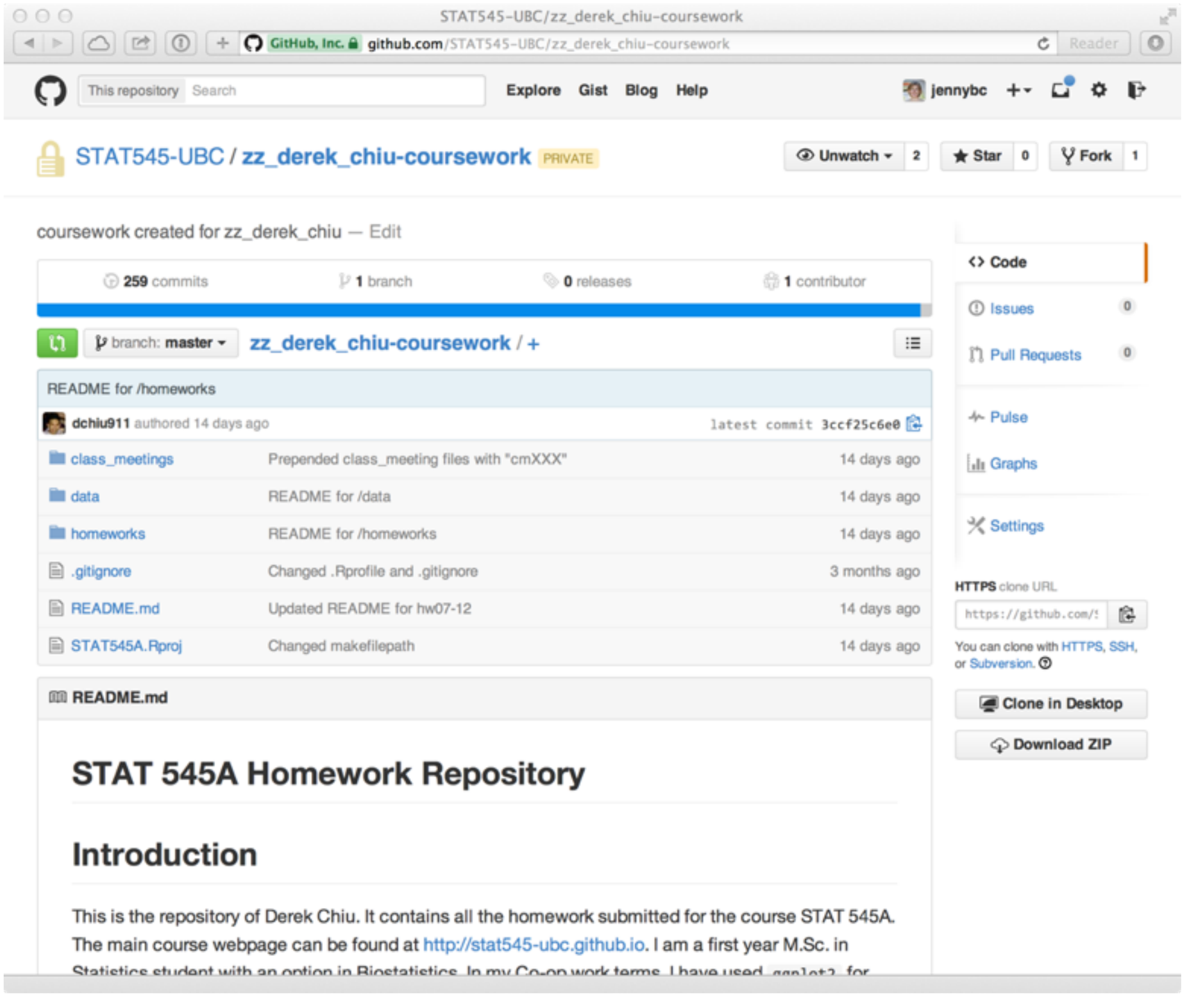
Using Git at the command line to pull (remote ⇒ local)

Git client

Git

repo

GitHub = a place to host Git repositories on the web
GitHub ≠ Git



Many R packages are developed in the open on GitHub

Nice option when someone tells you to "read the source"!

Many government agencies, media outlets, academic labs, etc. put their stuff on GitHub

https://github.com/WhiteHouse
https://github.com/chicago
https://github.com/fivethirtyeight
https://github.com/TheUpshot
https://github.com/propublica/
http://ncip.github.io (NCI's informatics program)
https://github.com/LSST (Large Synoptic Survey Telescope)
https://github.com/ctb (Titus Brown lab)
https://github.com/lh3 (Heng Li lab)

STAT 545 is an Organization on GitHub

all course materials are posted there (public repo)

all course development was done there (private repo for instructors only)

each student had his/her own repo for coursework (visible only within the Organization)

rough notes on set-up

When I mark homework ... this is what I see.

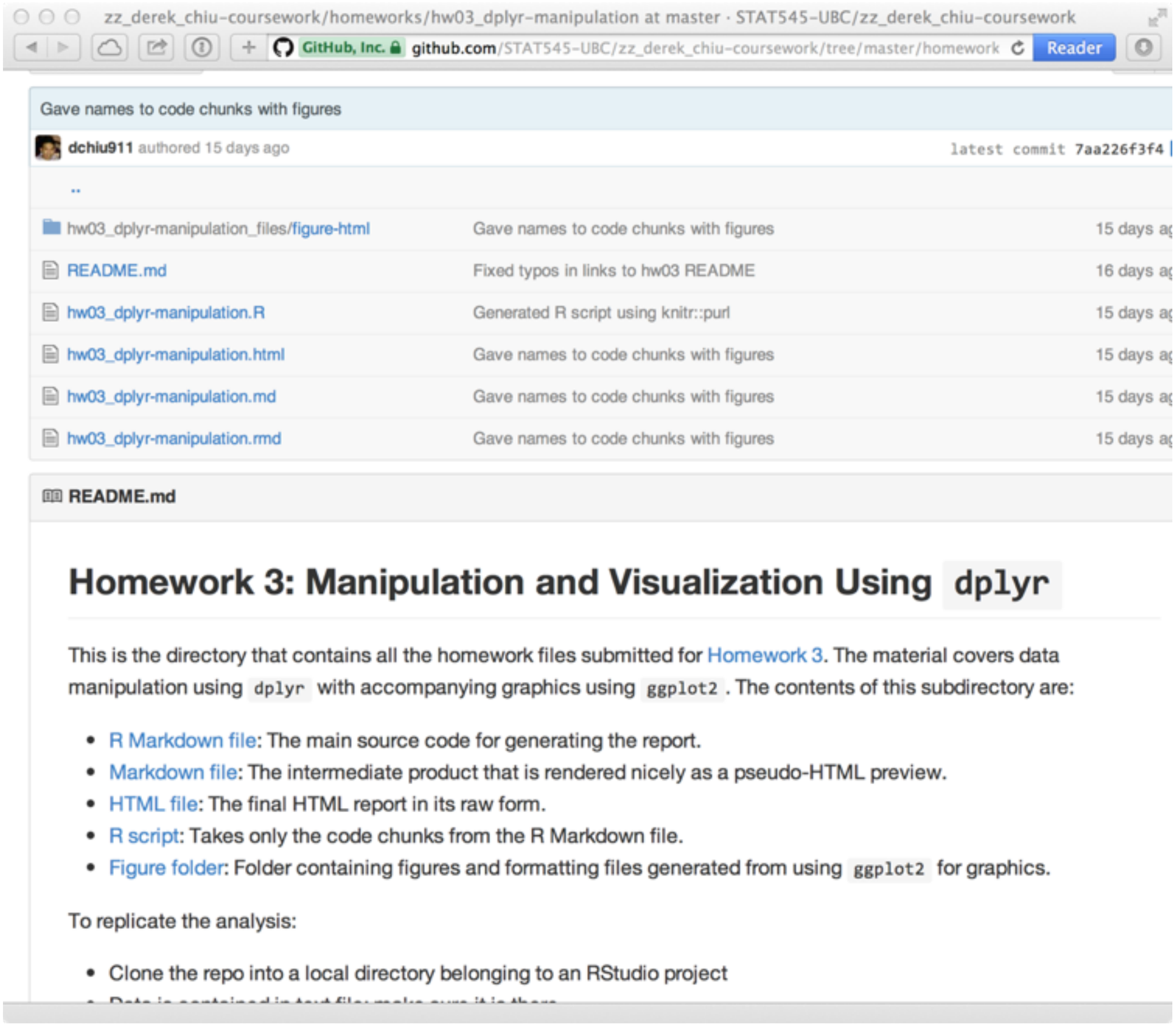
STAT 545A

- Homework 1: Edit README.md and experiment with Markdown
- Homework 2: Exploring the Gapminder Dataset
- Homework 3: Manipulation and Visualization Using `dplyr`
- Homework 4: Writing and Testing Functions
- Homework 5: Factor Control and File I/O
- Homework 6: *Optional:* Transition Activities

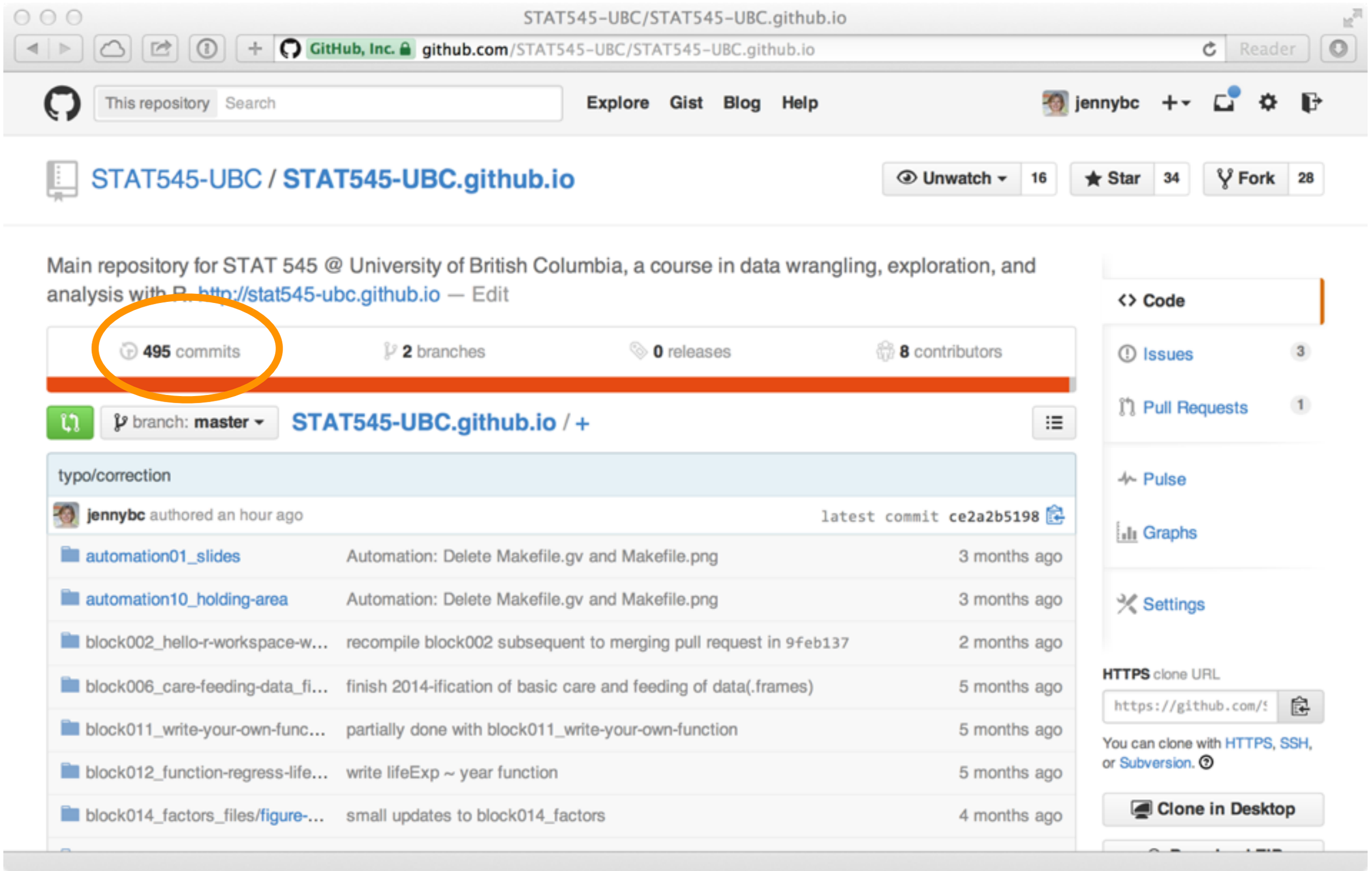
STAT 547M

- Homework 7: Data Wrangling Grand Finale
- Homework 8: Data Cleaning
- Homework 9: Automating Data Analysis Pipelines
- Homework 10: Building an R Package
- Homework 11: Building a Shiny App
- Homework 12: Getting Data off the Web

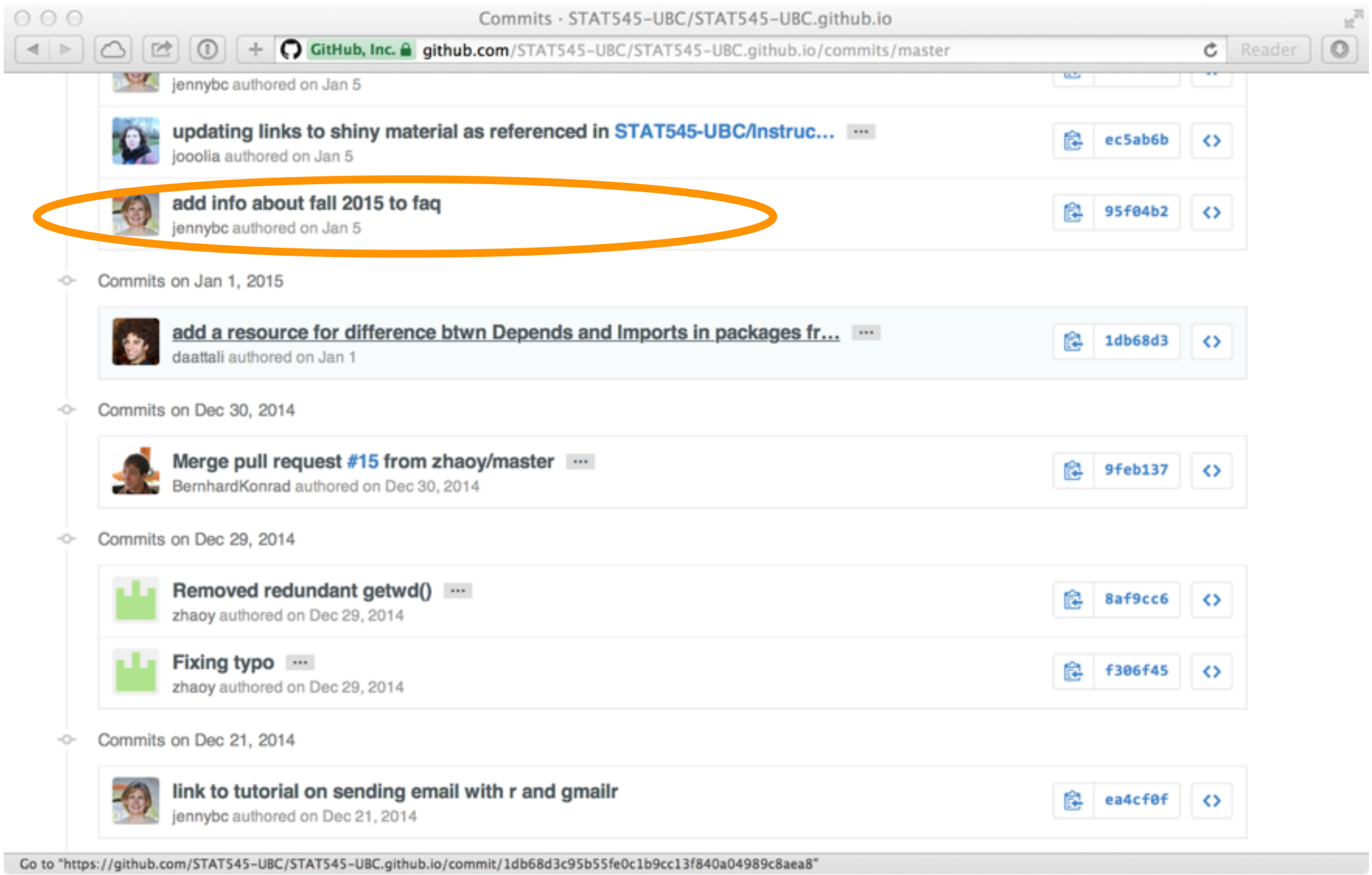
When I mark homework ... this is what I see.



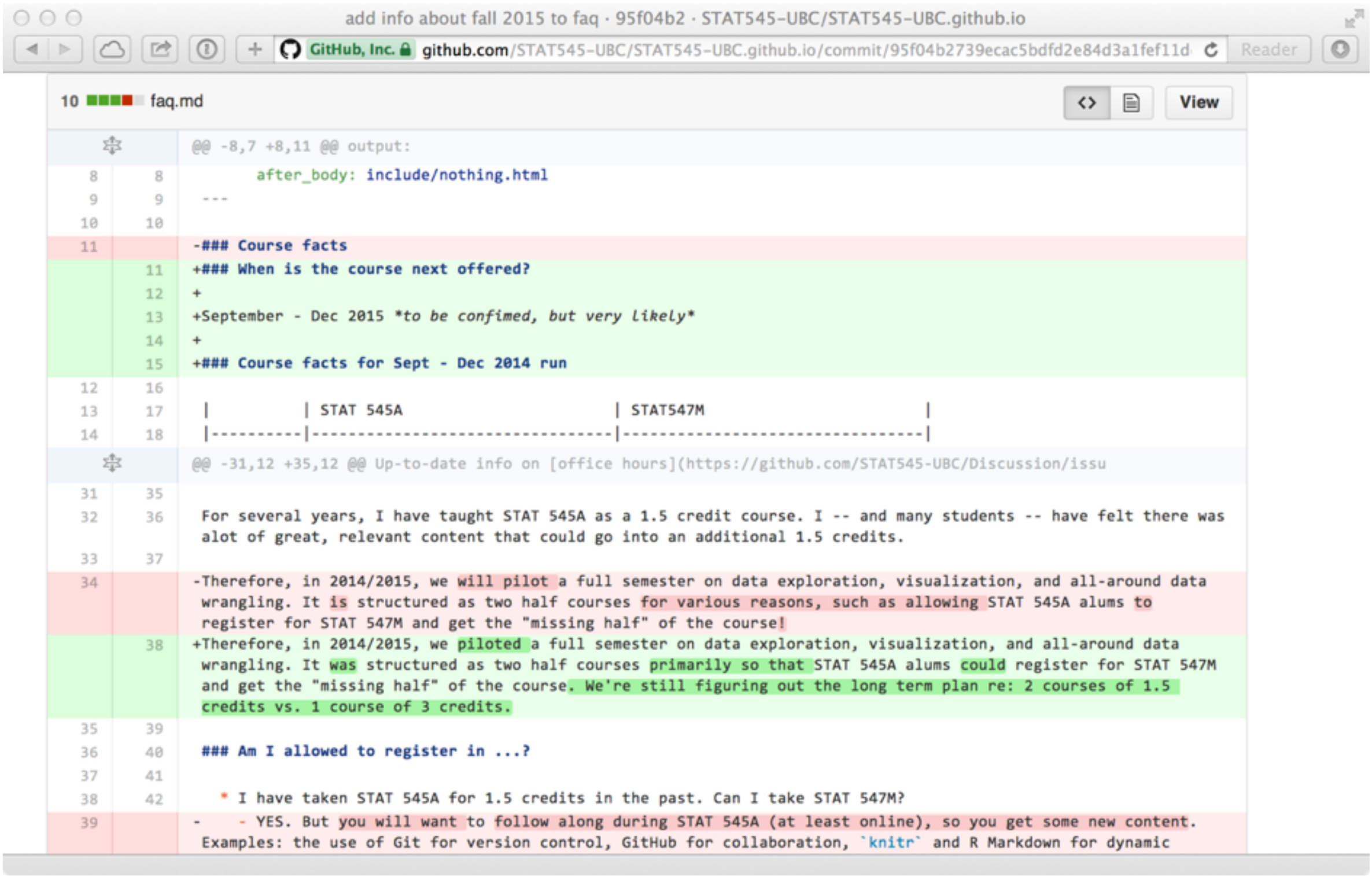
Commits are how the files evolve



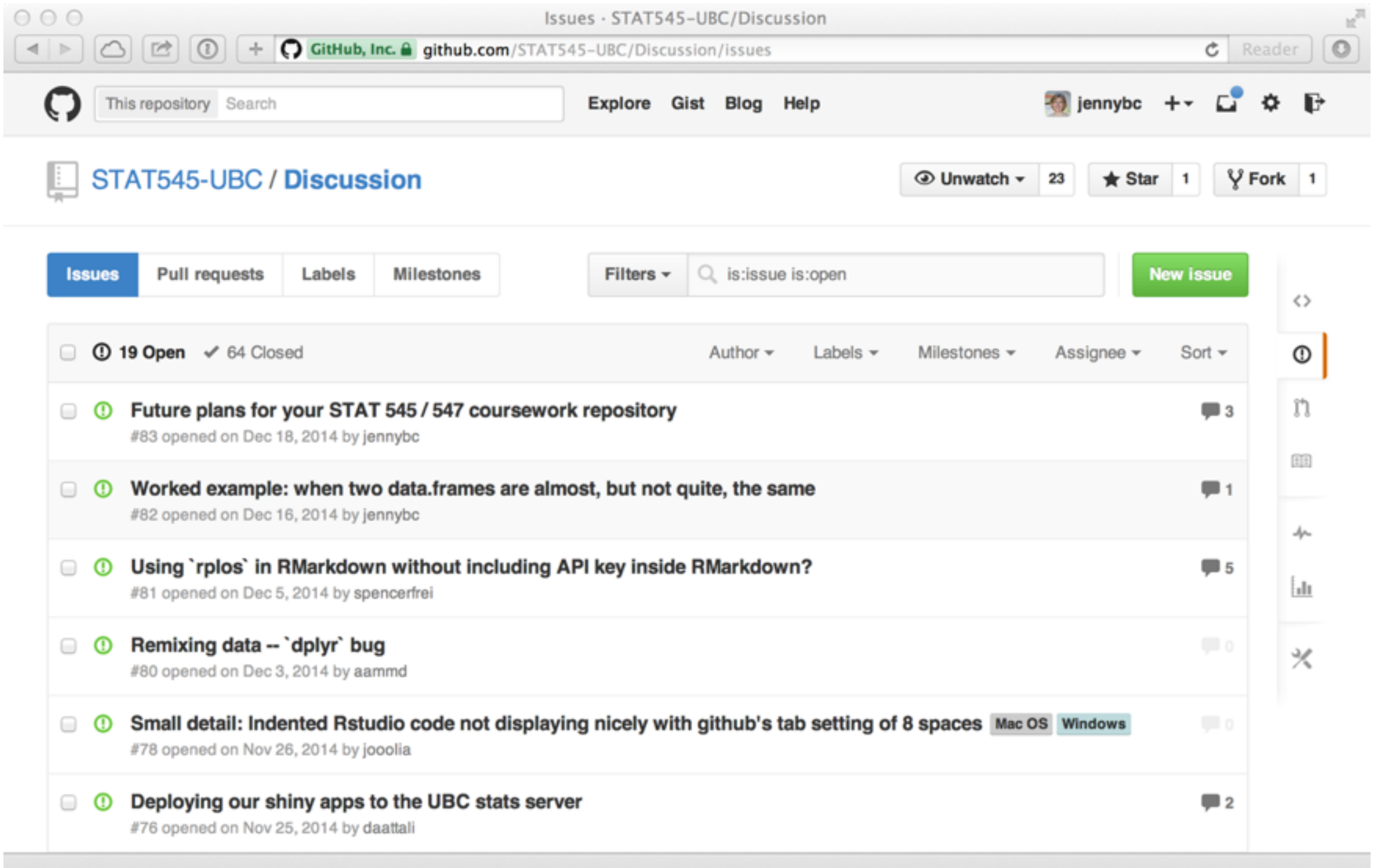
Commit message = short description of what/why changed



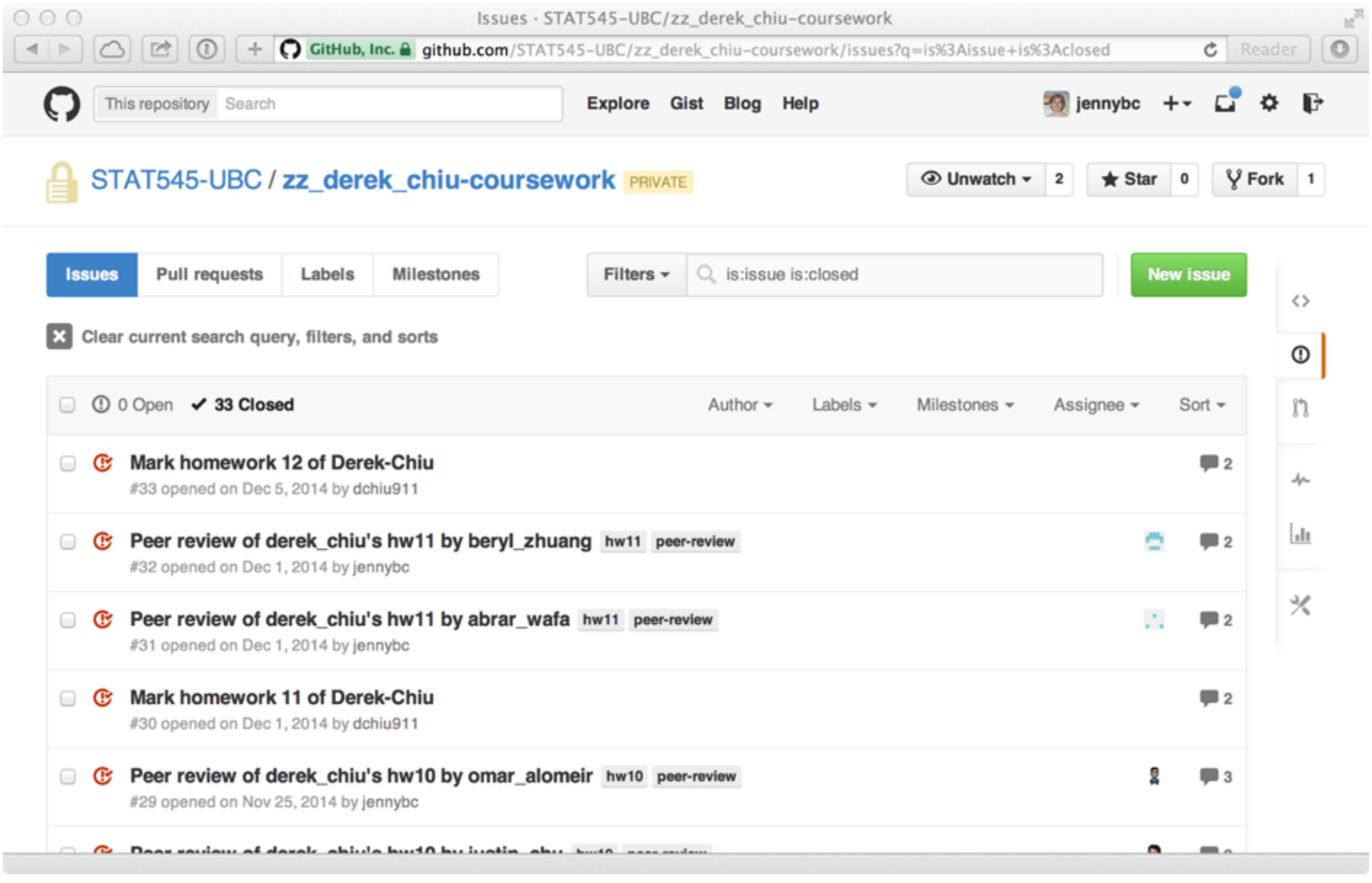
“diffs” compare a file then vs. now



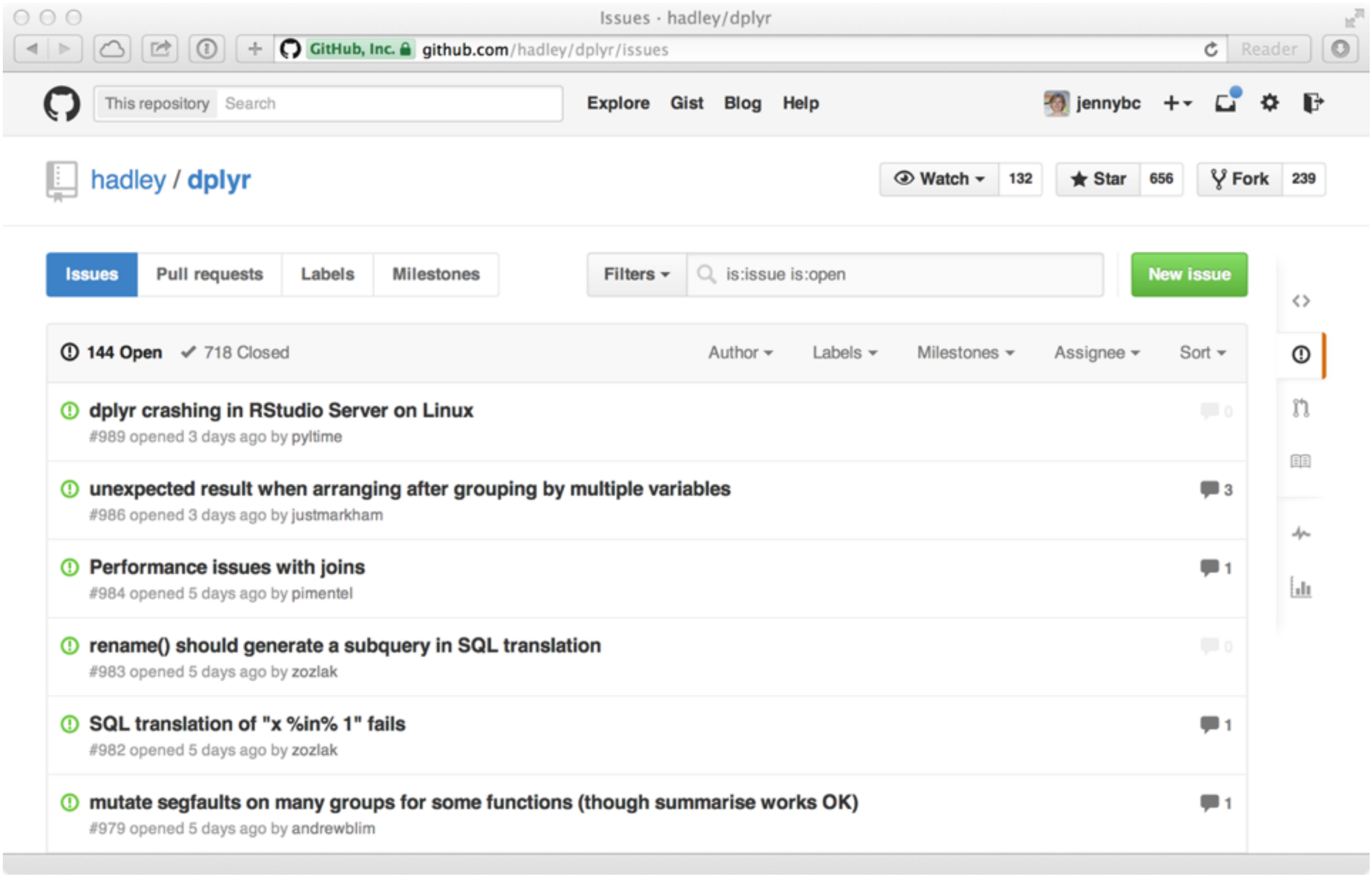
GitHub repositories can have *issues*: think discussion forum.



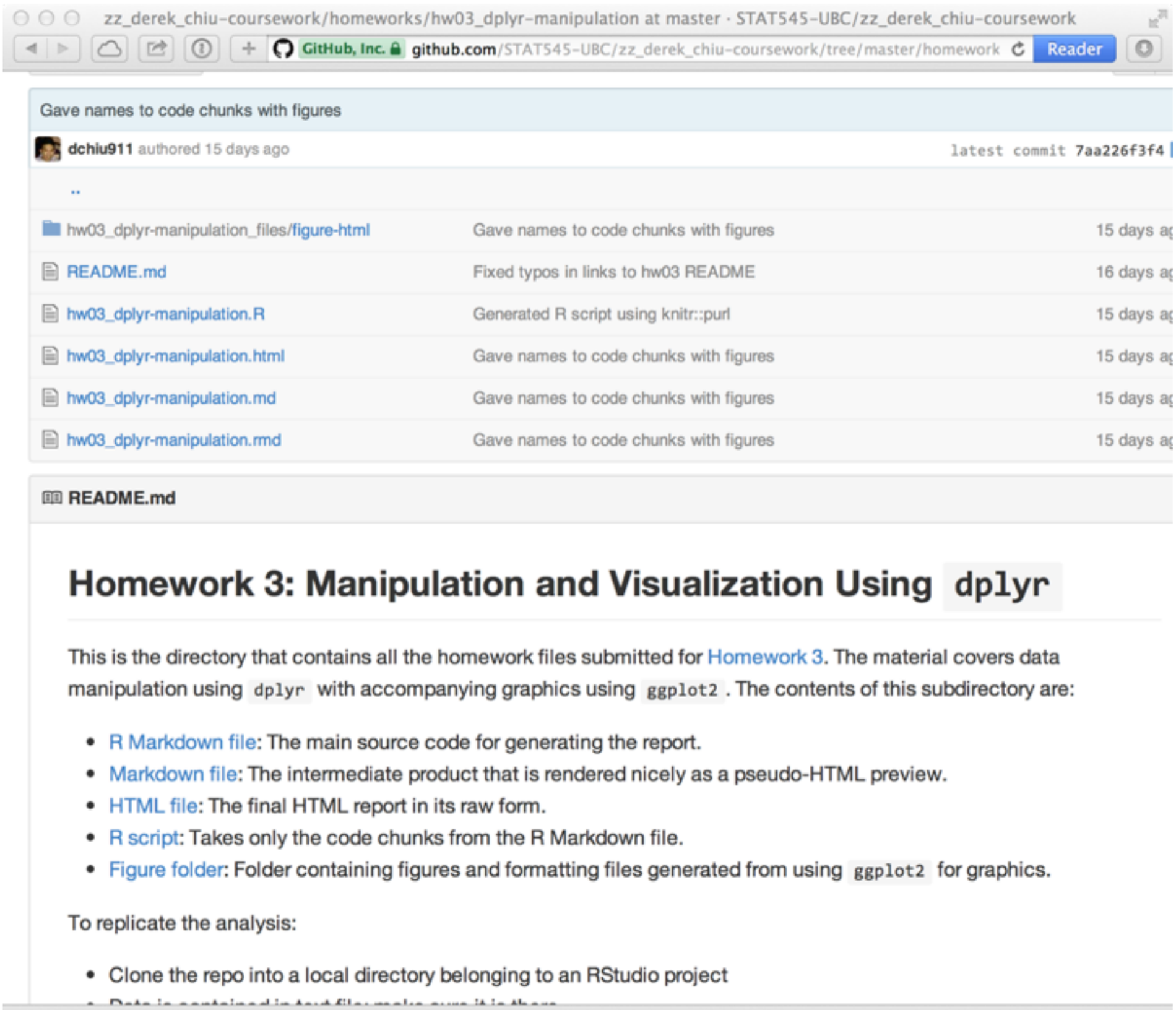
GitHub repositories can have *issues*: think "to do list"



GitHub repositories can have *issues*: think "bug tracker"



Markdown files are automatically rendered nicely in GitHub repositories



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

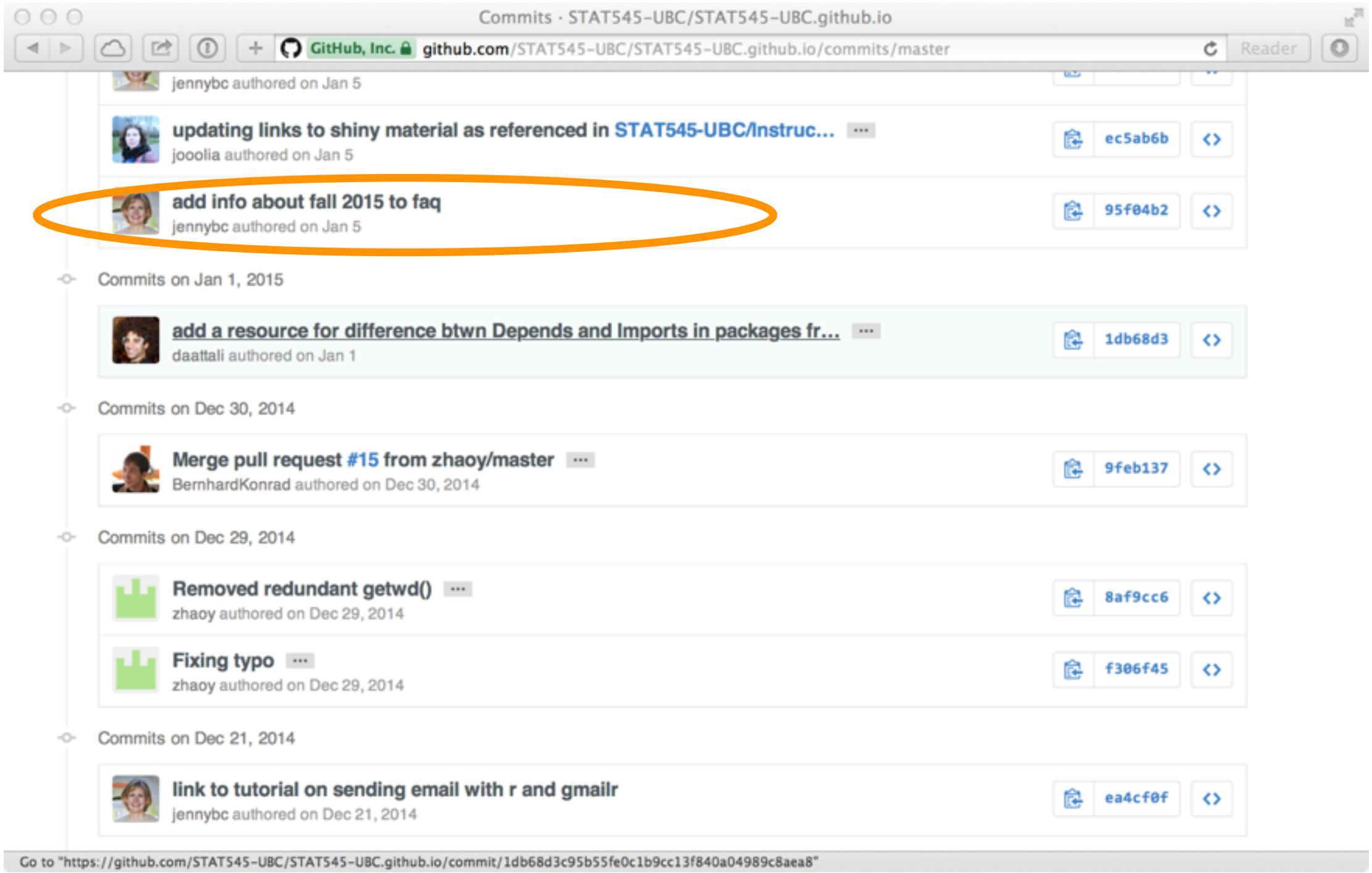
1 contributor

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

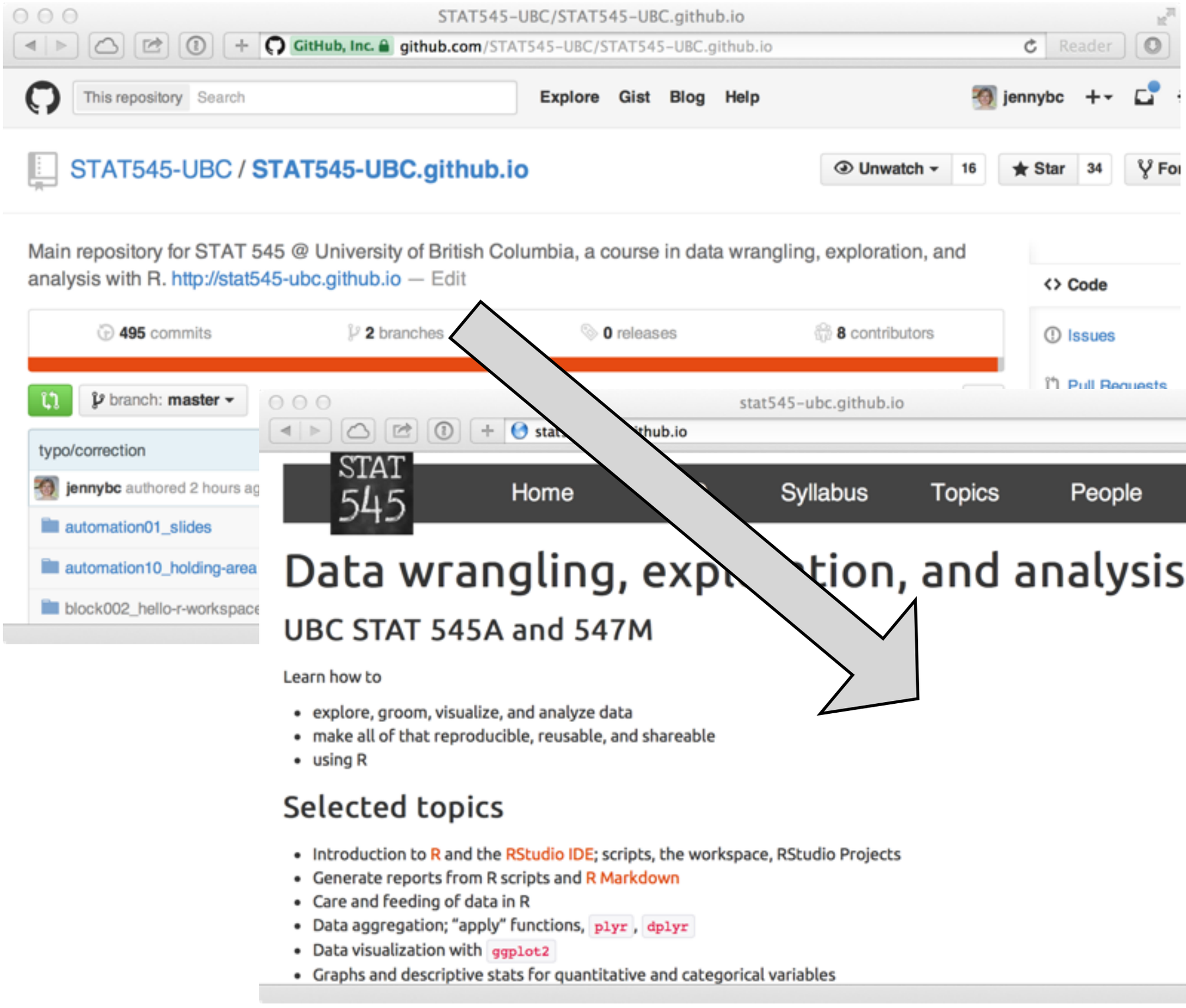
Search this file...

	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

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



When prof or TA pushes to repo, website updates!



Files in a Git repo, even one hosted on GitHub, still reside on your computer

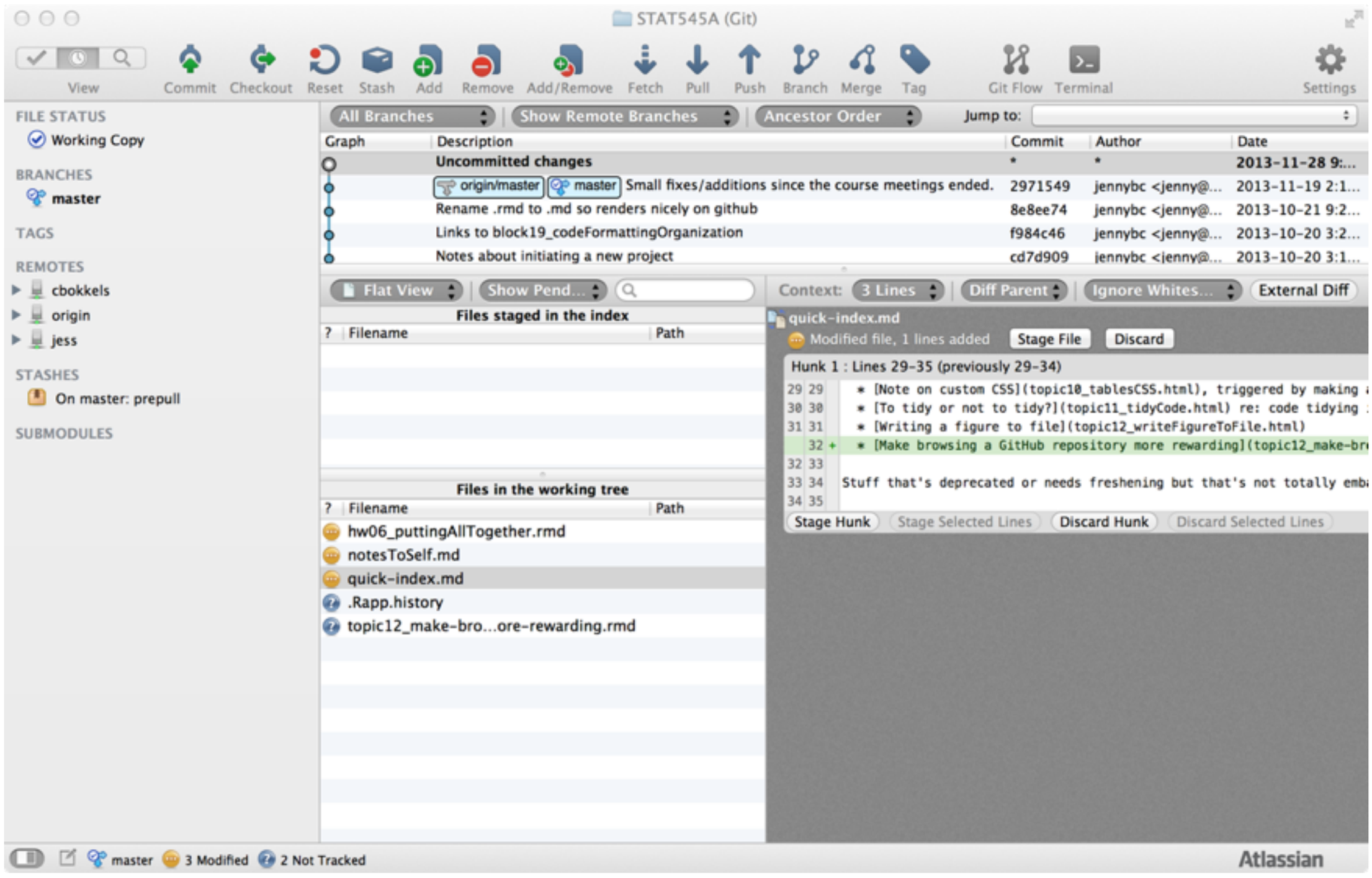
Browse and edit them all you want

Git has commands for communicating with the remote repository, e.g. the GitHub repo (push, pull, fetch, clone)

I highly recommend using a Git GUI on your computer for making commits, syncing with the remote, etc.

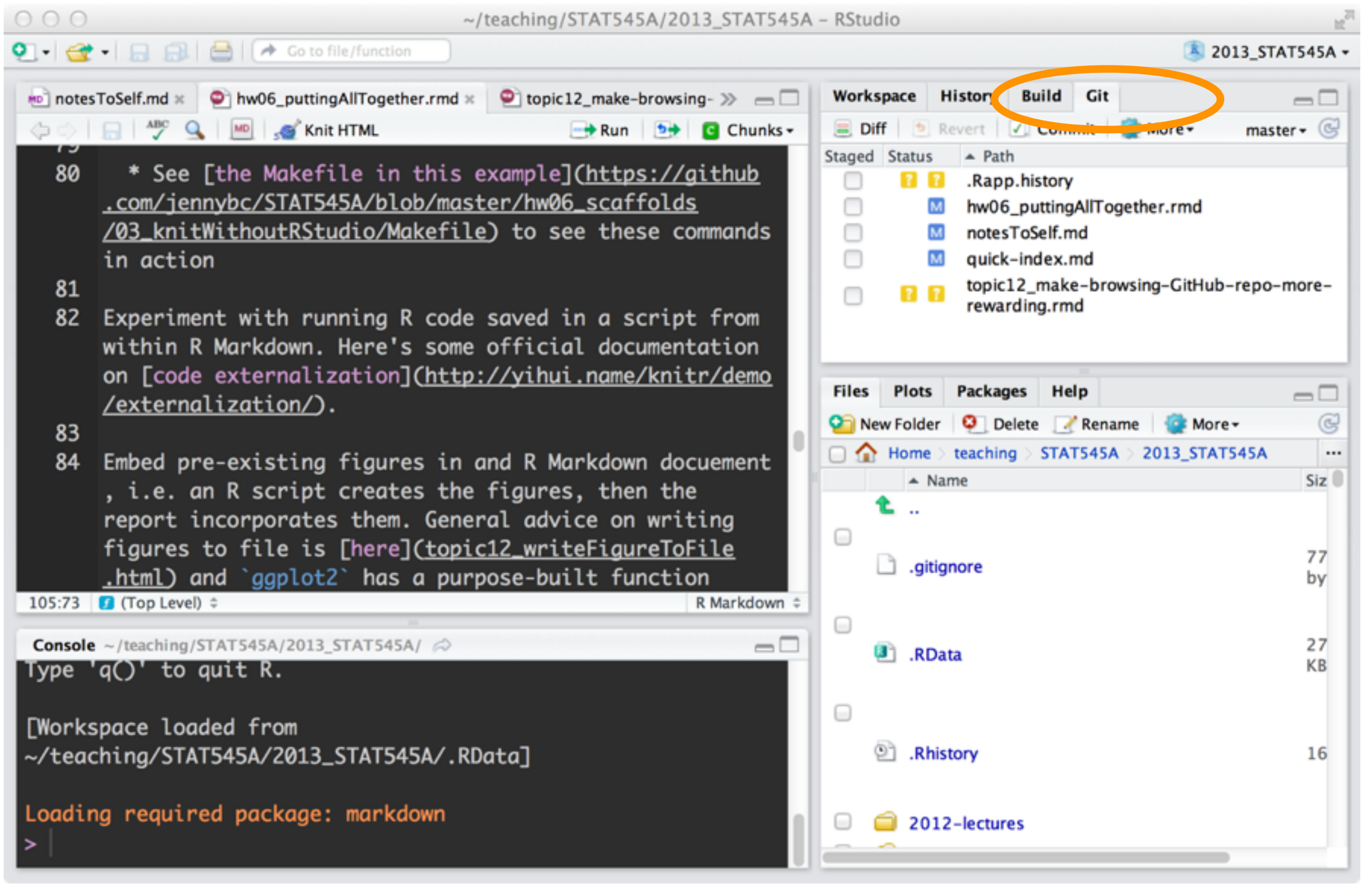
Reconciling and merging changes when two people make conflicting commits is not fun, but better than the alternatives

I recommend SourceTree, a free Git client for Windows and Mac.



RStudio can also act as your Git(Hub) client

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



Big picture, second half:

sane file and project management is good
that's what version control does

distributed file management is good
excellent for 2+ people collaborating

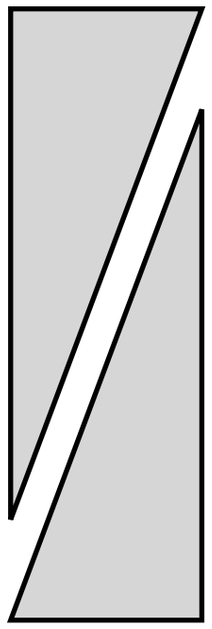
ability to browse something on the web is unreasonably powerful

Git + GitHub provide a compelling solution for collaborative file wrangling; (R) Markdown and RStudio play well with Git(Hub)

STAT 545 = 1 semester, 3 contact hours/wk

R markdown

Git(Hub)



Data wrangling, cleaning, munging

Visualization

8 weeks

(R chops, in general)

Automation & pipelines

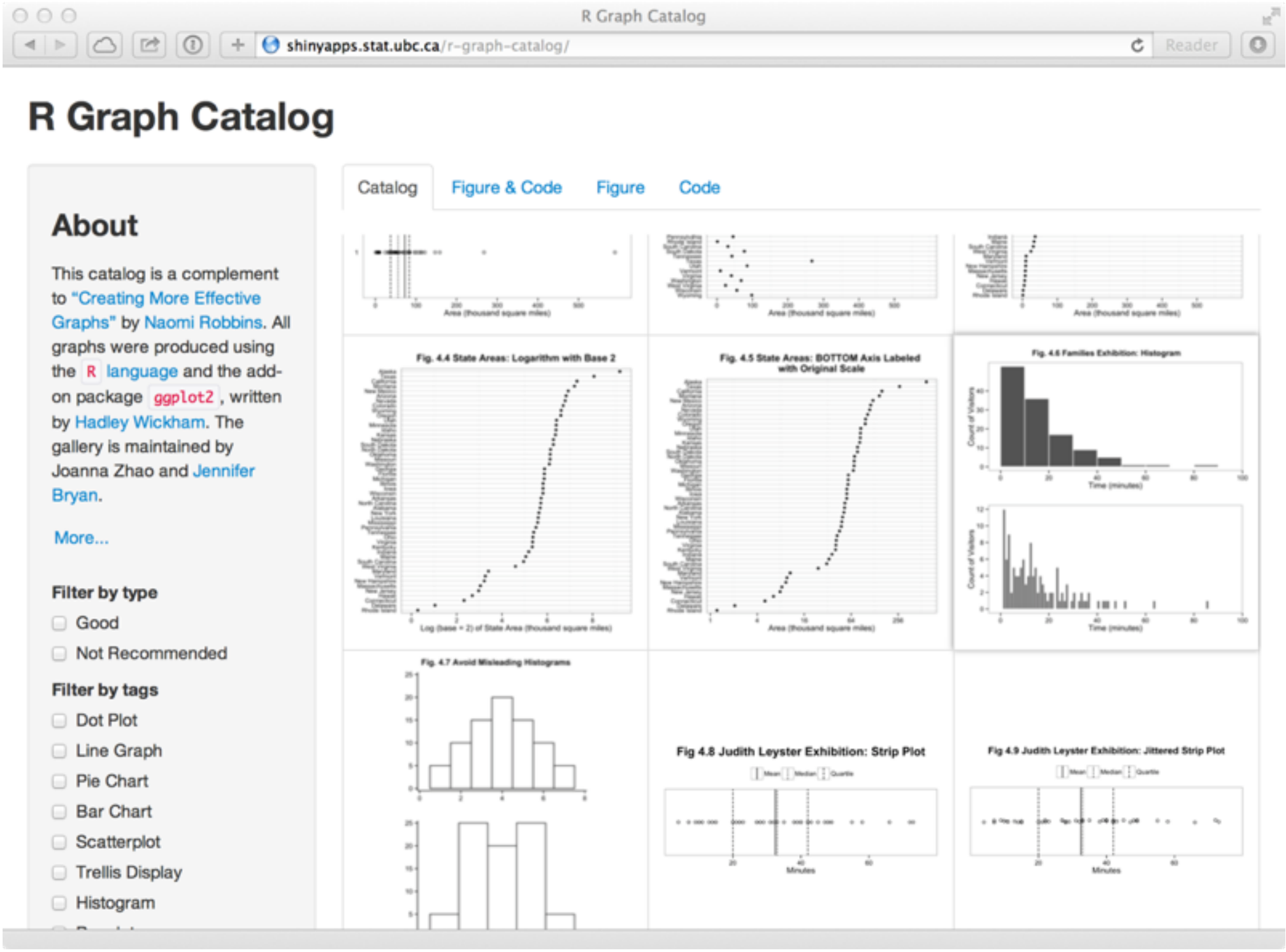
R packages

4 weeks

Shiny

Web APIs and scraping

http://shinyapps.stat.ubc.ca/r-graph-catalog/
https://github.com/jennybc/r-graph-catalog



Bottom line: do something deliberate that has a good hassle: result ratio for you.

Be open to upgrading your approach as time goes on.

Keep your eyes and ears open re: new developments.