Git & github

Hadley Wickham

@hadleywickham

Chief Scientist, RStudio



⊚ 9 9 9 1 February 2015

Overview

- Focus mostly on the why, not the how
- Learning git & github is definitely frustrating at first
- Two big pay offs: safety & community
- Mechanics at http://r-pkgs.had.co.nz/git.html

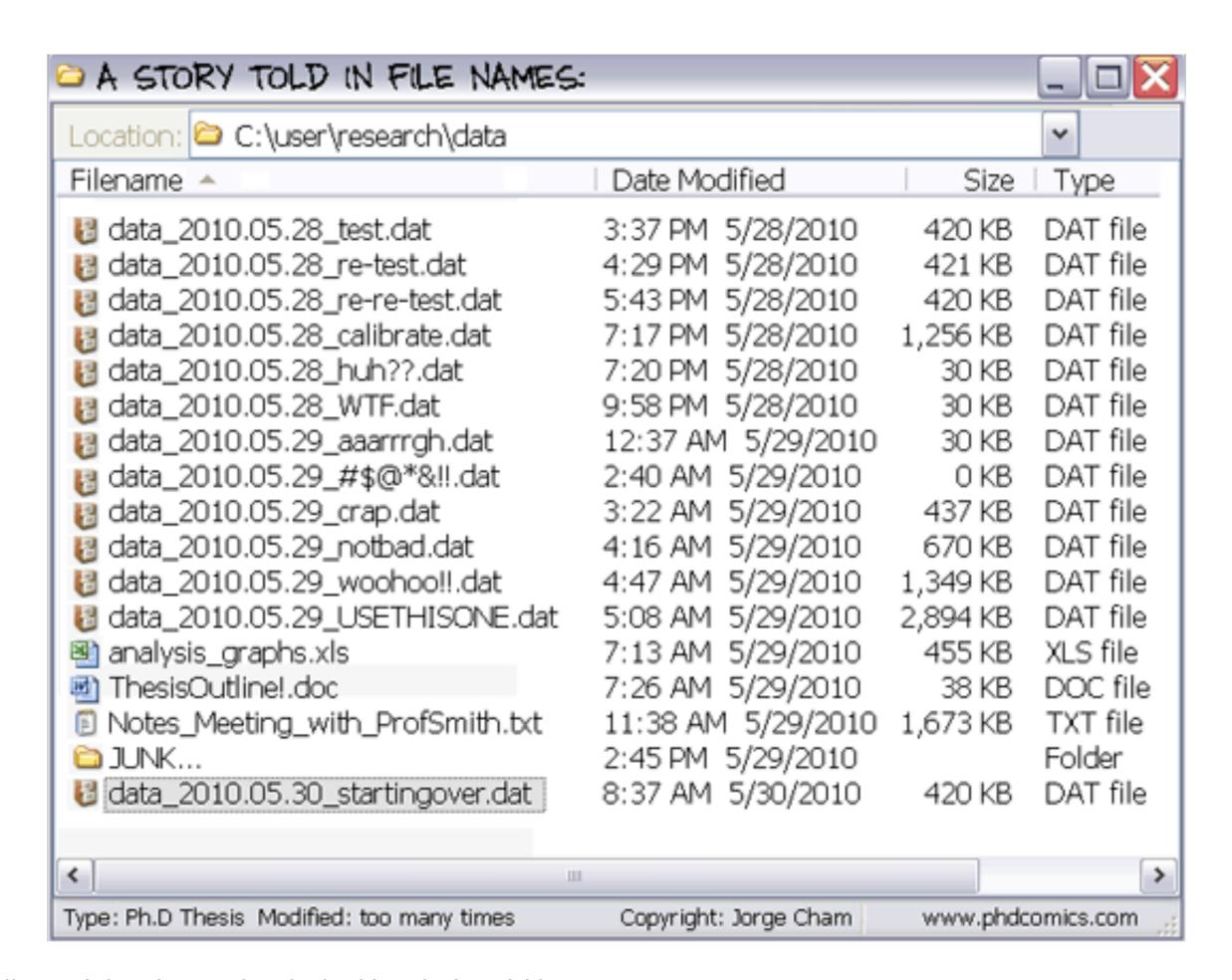
Saiety

"Failures, repeated failures, are finger posts on the road to achievement. One fails forward toward success."

- C. S. Lewis







2

"FINAL".doc



FINAL.doc!



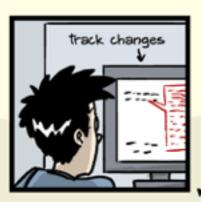
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5. CORRECTIONS.doc



JORGE CHAM @ 2012

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



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





The history of a project can be viewed as a series of changes

\bigcirc

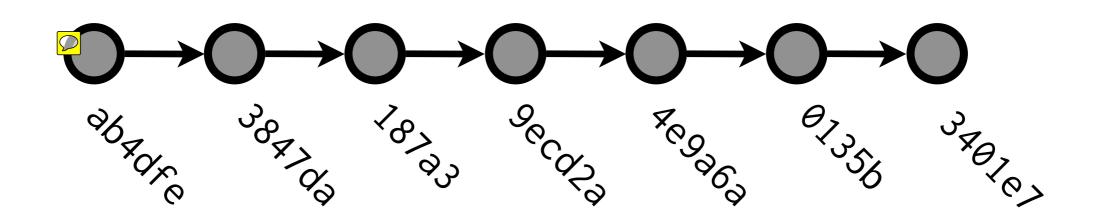
Changes

- A unique identifier
- What changed?
- When did it change?
- Who changed it?
- Why did it change?

Changes

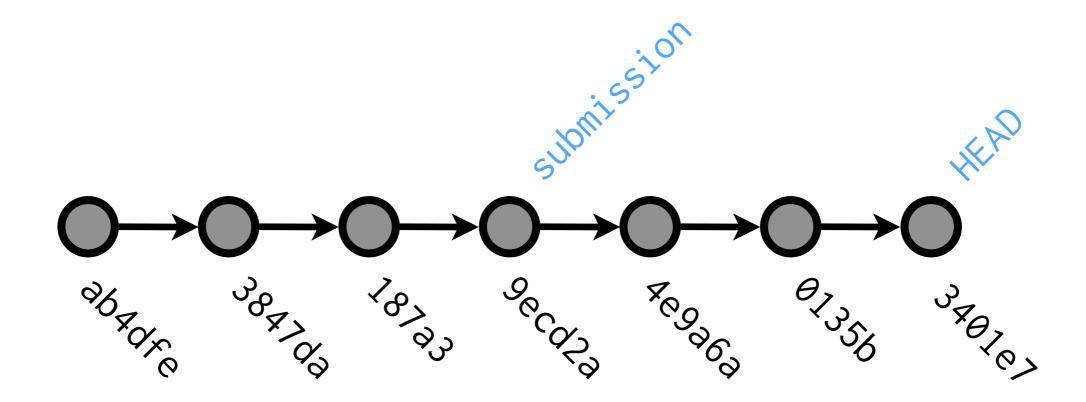
- A unique identifier
- What changed?
- When did it change?
- Who changed it?
- Why did it change?

With git, each change (**commit**) is given a unique identifier, called a **sha**



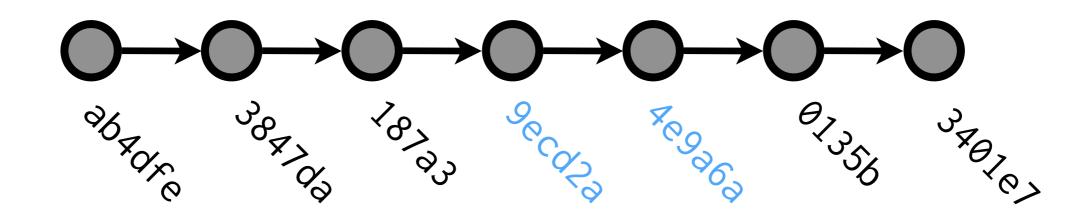
The sha is a key into a database that provides the author, date, and a description

You can also name individual commits



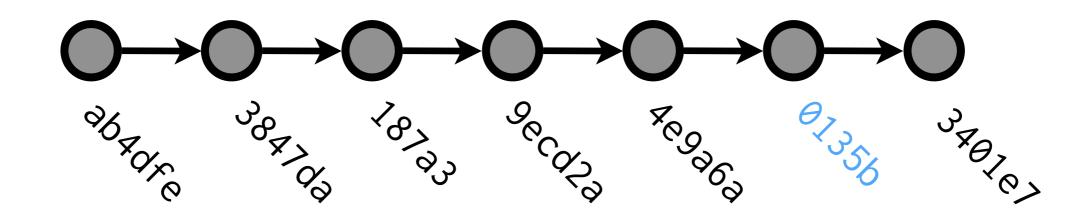
git tag submission 9ecd2a

Then see exactly what's changed



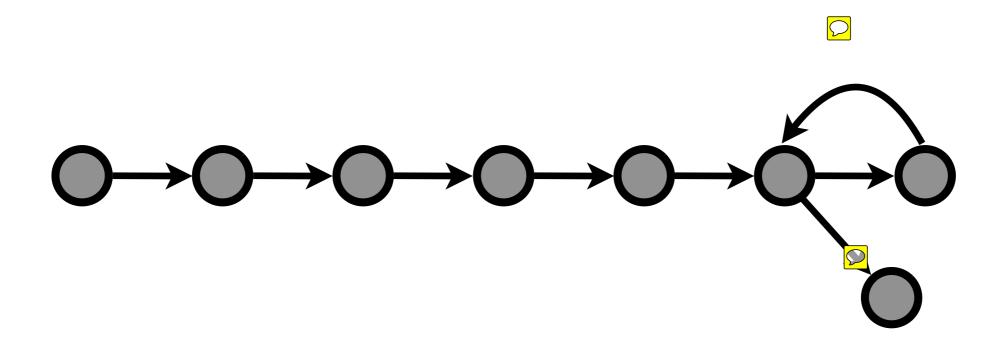
git diff 9ecd2a..4e9a6a

You can revert to a previous change with git checkout

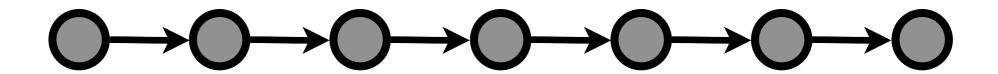


git checkout 0135b

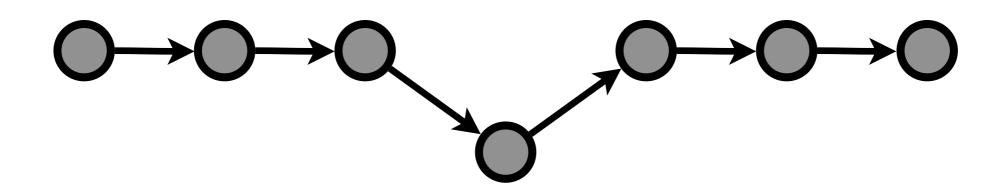
That allows you to undo mistakes



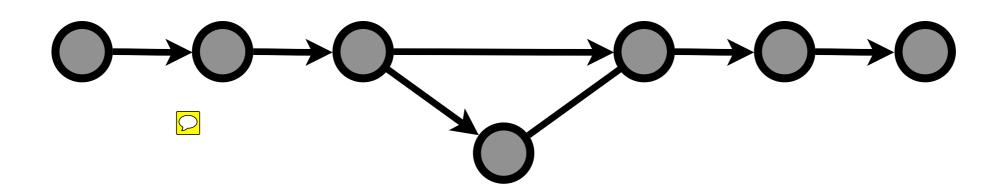
Alternatively you can go back in time and pretend mistakes never existed



Alternatively you can go back in time and pretend mistakes never existed



Alternatively you can go back in time and pretend mistakes never existed. **That's risky!**



git rebase ...

Demo

http://r-pkgs.had.co.nz http://github.com/hadley/r-pkgs

Demo

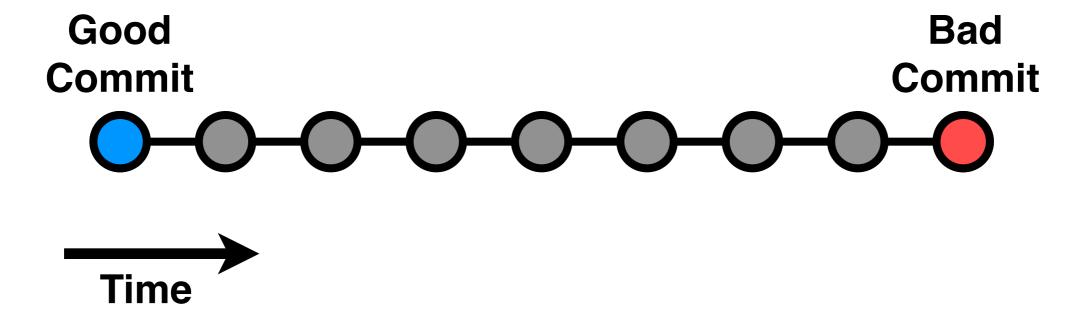
Vows package & roxygen2



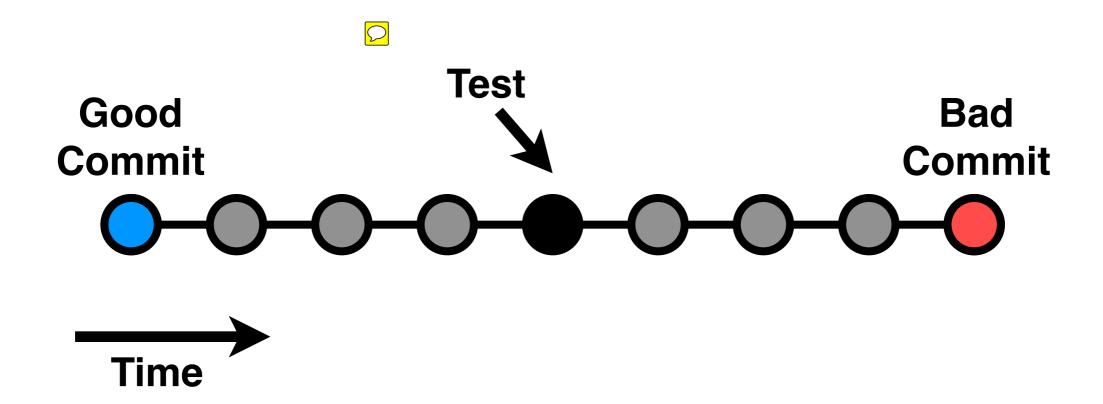
Other benefits

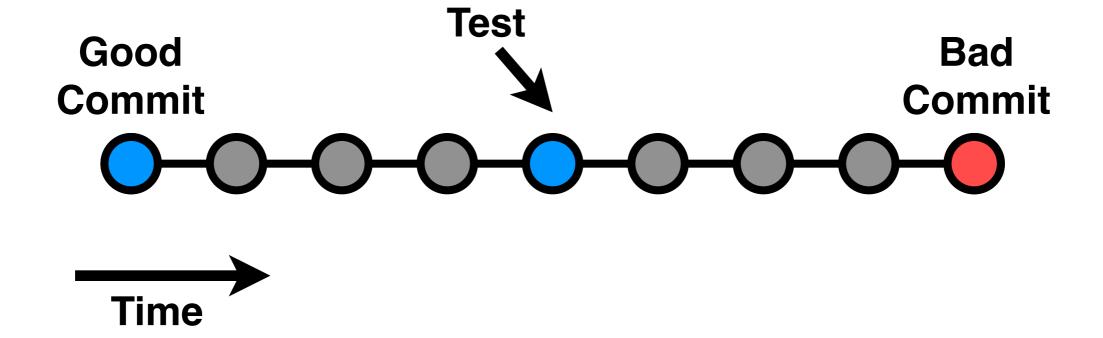
git bisect

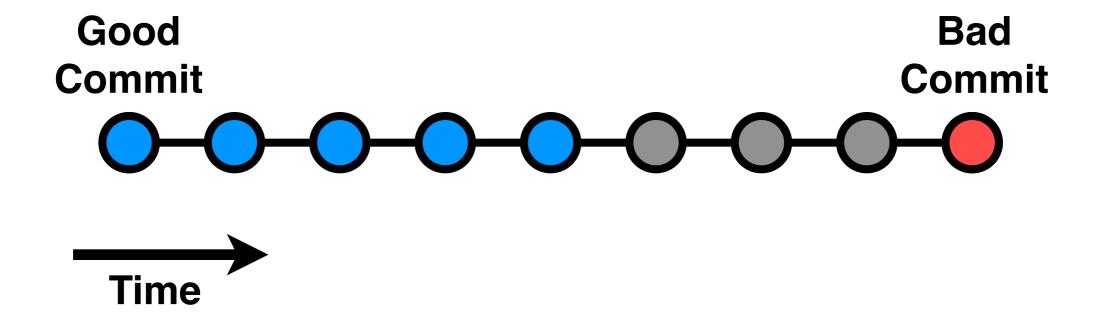
https://github.com/wch/bisectr

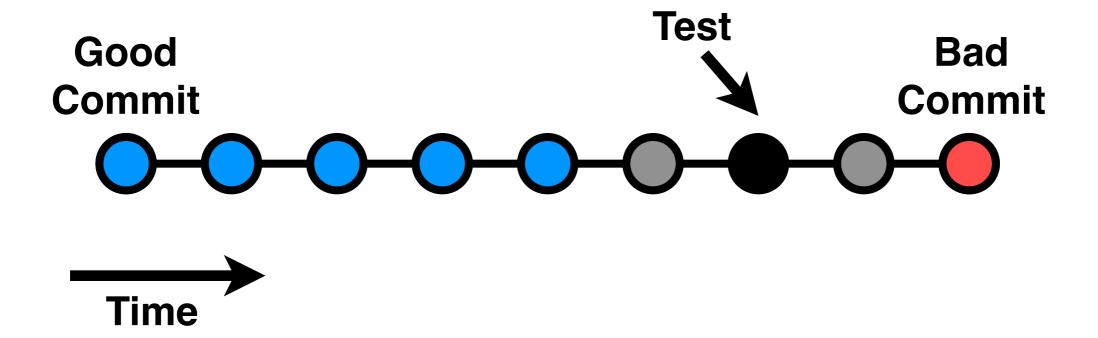


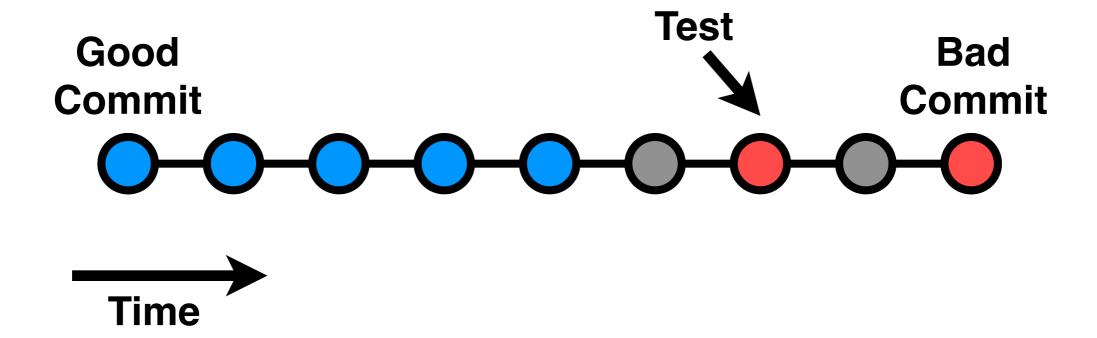
Where did the problem happen?

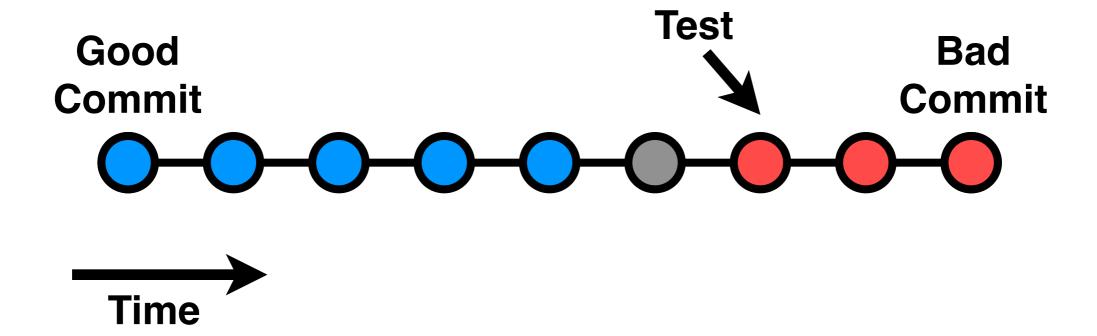






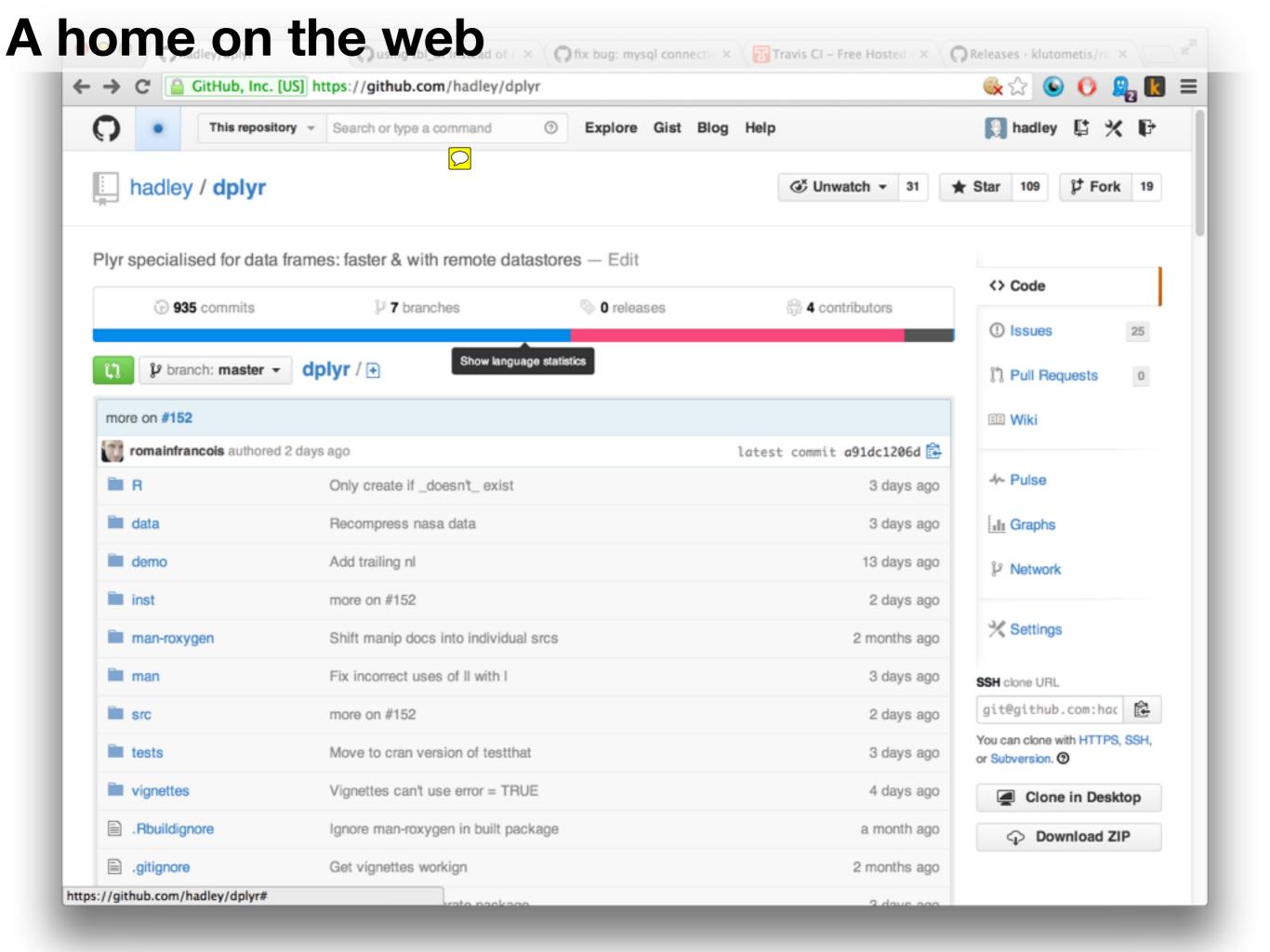








Community



GitHub, Inc. [US] https://github.com/hadley/dplyr







dplyr

build failing

dplyr is the next iteration of plyr, focussed on tools for working with data frames (hence the a in the name). It has three main goals:

- · Identify the most important data manipulation tools needed for data analysis and make them easy to use from R.
- Provide blazing fast performance for in-memory data by writing key pieces in C++.
- . Use the same interface to work with data no matter where it's stored, whether in a data frame, a data table or database.

dplyr is not yet available on CRAN, but you can install it from github with:

devtools::install_github("dplyr")

To get started, read the notes below, then read the intro vignette:

vignette("introduction", package = "dplyr") . To make the most of dplyr, I also recommend that you familiarise yourself with the principles of tidy data: this will help you get your data into a form that works well with dplyr, ggplot2 and R's many modelling functions.

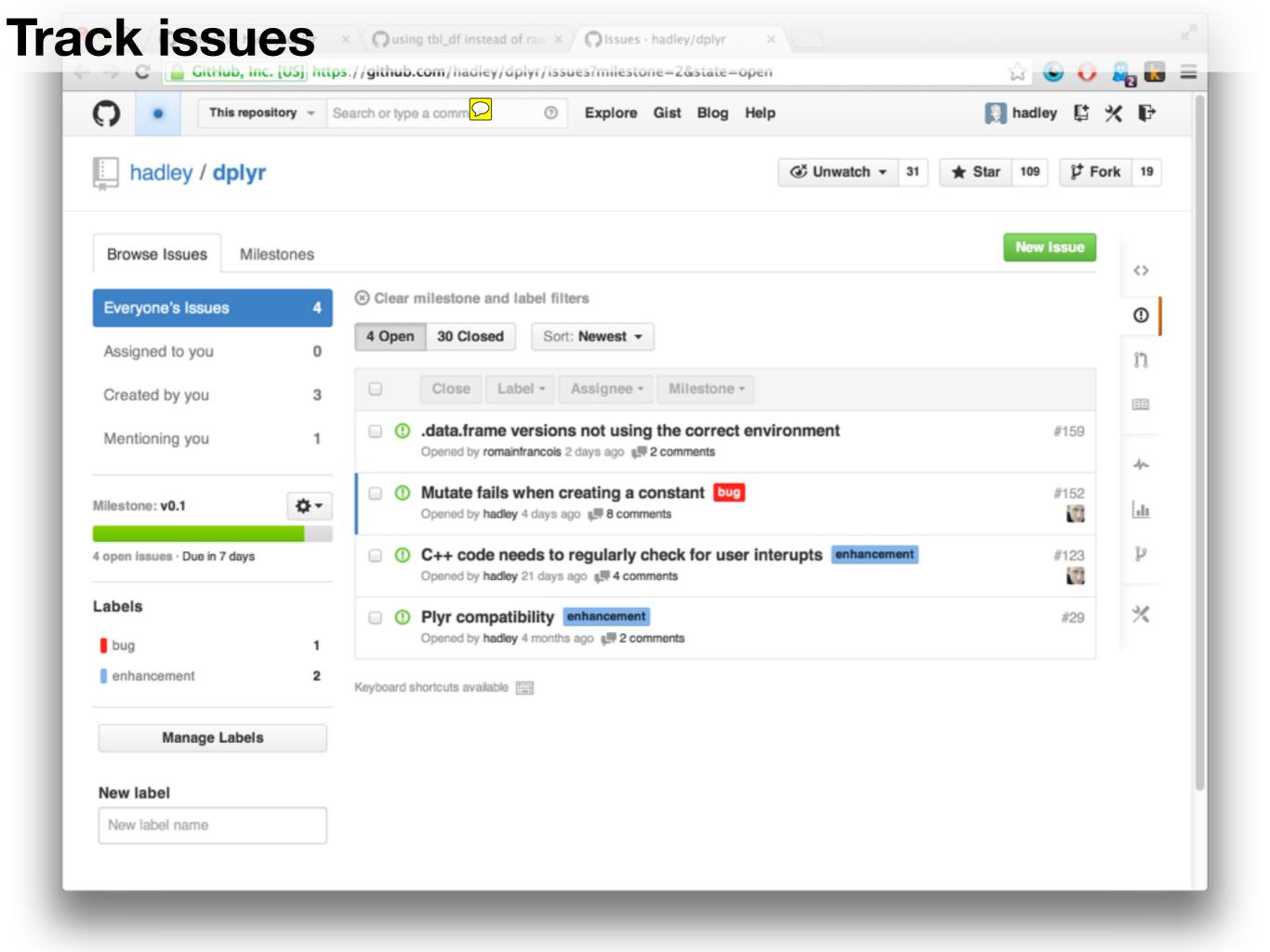
If you encounter a clear bug, please file a minimal reproducible example on github. For questions and other discussion, please use the manipulatr mailing list.

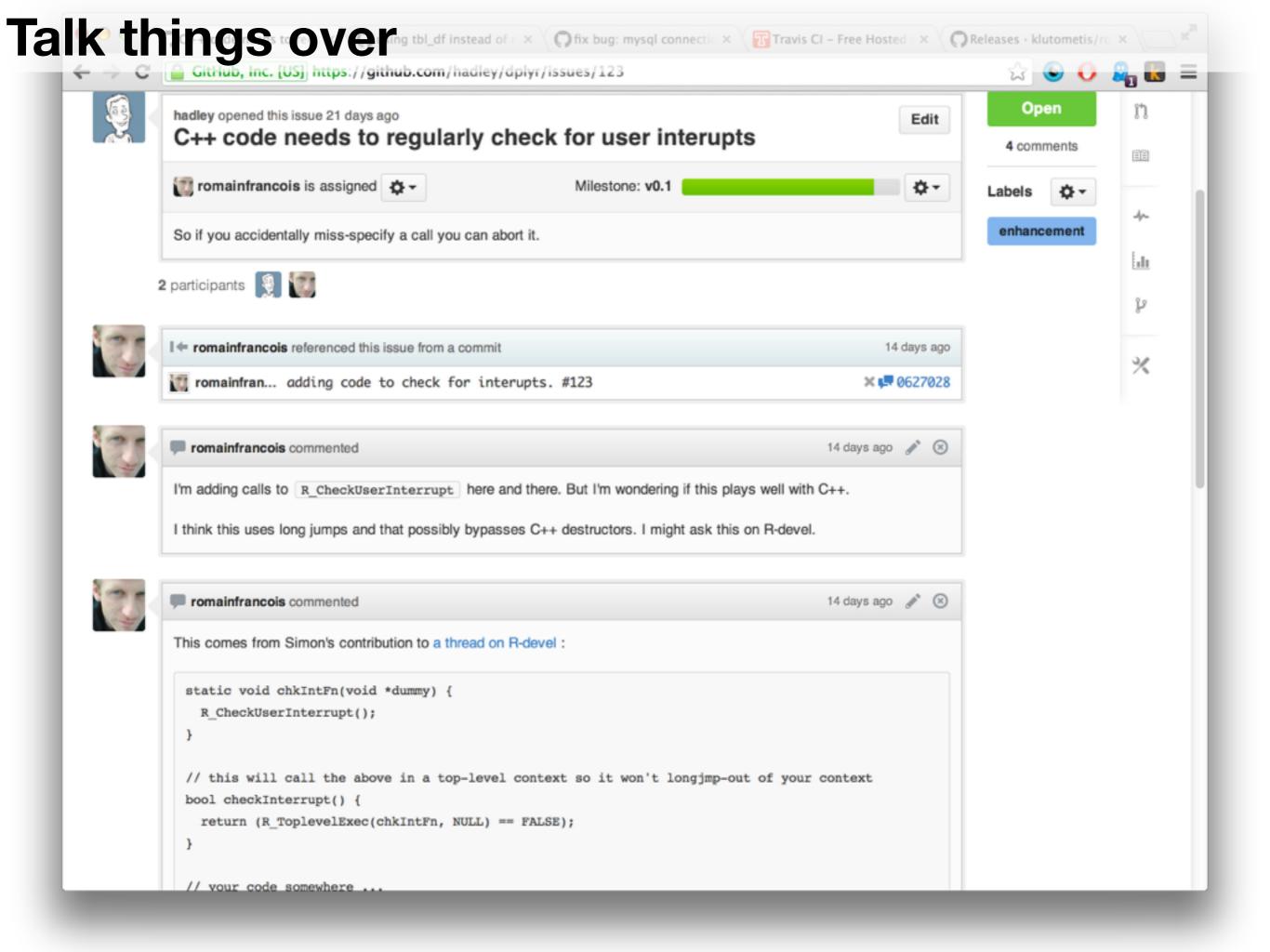
```
# Easy to install a package from github
library(devtools)
install_github("hadley/devtools")
```

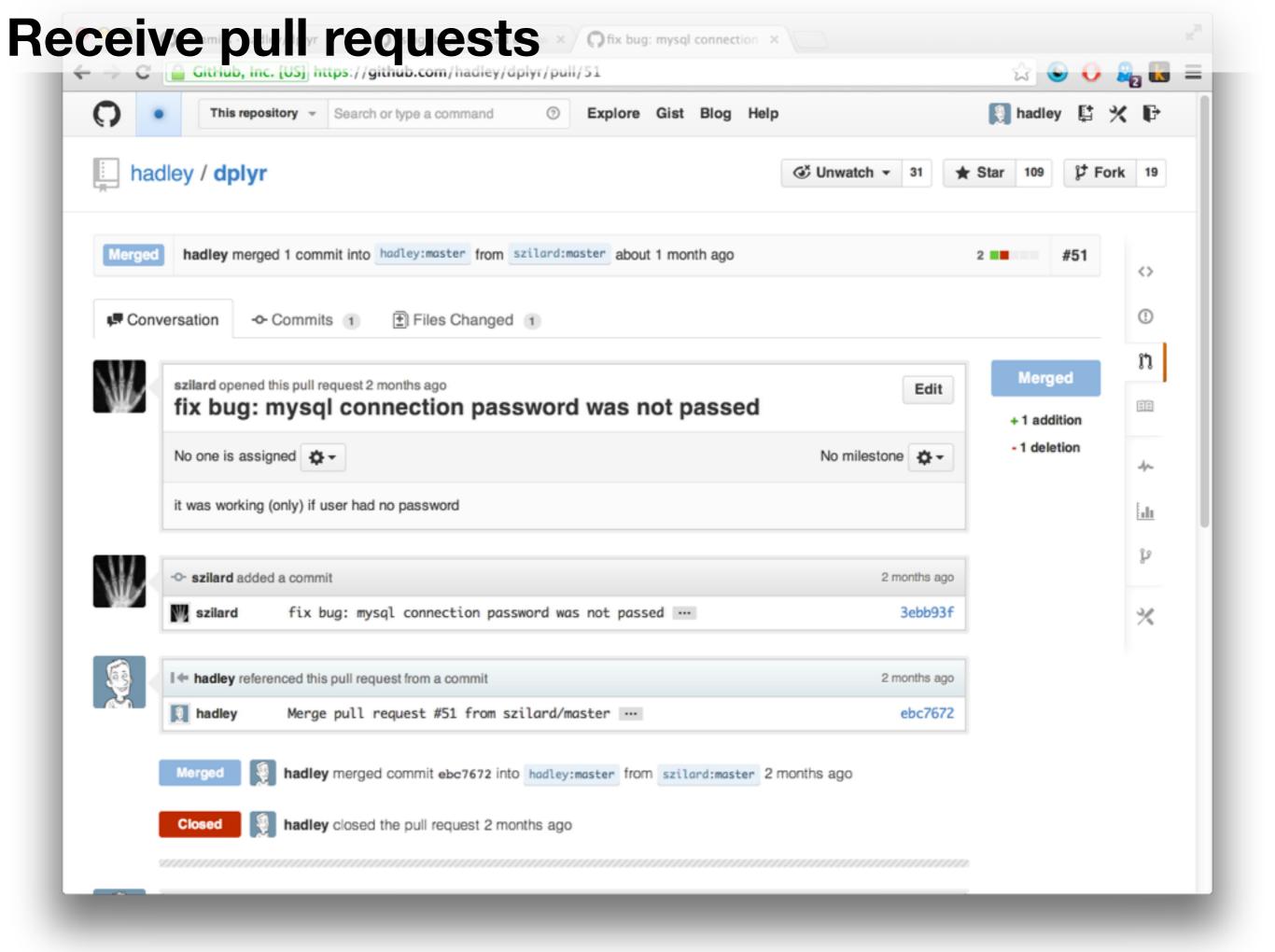
```
# Github is a viable location for package
# distribution, but it doesn't yet have the
# reach or authority of CRAN. Currently
# about ~3,000 packages on github.
```

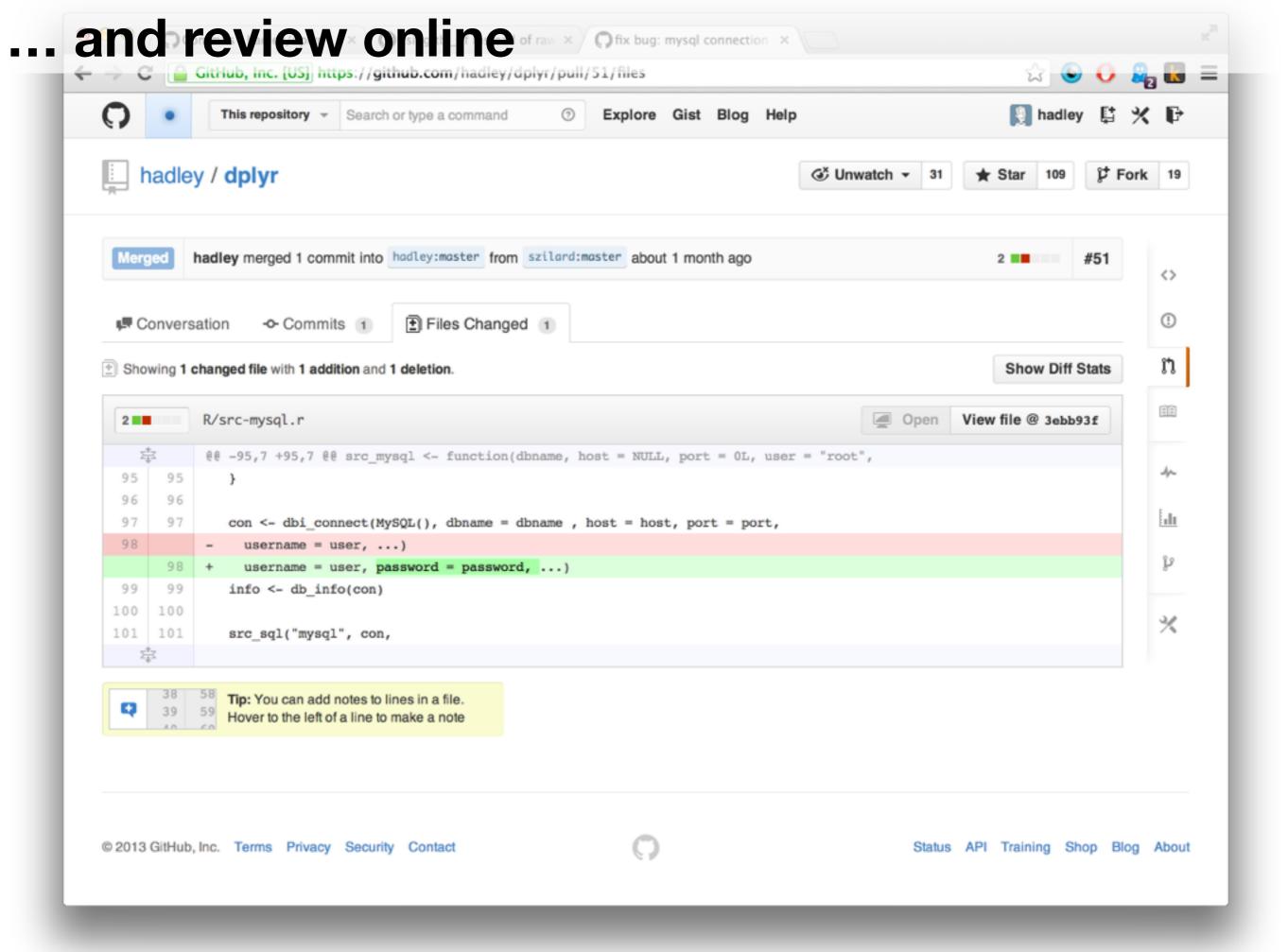
See what's happening GitHub, Inc. [US] https://github.com/hadley/dplyr/commits/master Nadley E This repository -Search or type a command Explore Gist Blog Help \bigcirc hadley / dplyr ★ Star dplyr / Commits <> Dec 07, 2013 (1) more on #152 a91dc1206d (+ romainfrancois authored 2 days ago Browse code ⇒ Dec 06, 2013 using tbl_df instead of raw data.frame for the test 864c8b7b9f > romainfrancois authored 2 days ago Browse code ⇒ fix test 58eaddc3b3 + romainfrancois authored 3 days ago Browse code ⇒ using NamedListAccumulator 0c7d40db1a + romainfrancois authored 3 days ago Browse code ⇒ handling date and time constants 457dd7e8f6 + romainfrancois authored 3 days ago Browse code ⇒ handling special cases in summarise. closes #153 eb2c3566f4 + romainfrancois authored 3 days ago Browse code ⇒ summarise handles constants (#153) ea873897a9 romainfrancois authored 3 days ago Browse code ⇒ dealing with the case summarise(df, log(z)) from #153 7d0fe6d829 romainfrancois authored 3 days ago Browse code ⇒ dealing with the case summarise(df, diff(z)) from #153 d278dfcec8 + romainfrancois authored 3 days ago Browse code ⇒

See exactly what changed GitHub, Inc. [US] https://github.com/hadley/dplyr/commit/864c8b7b9fae5329d99d818bd78b2b1686bdb167 This repository - Search or type a command Explore Gist Blog Help hadley / dplyr ★ Star using tbl_df instead of raw data.frame for the test Browse code <> ₽ master romainfrancois authored 2 days ago 1 parent 58eaddc commit 864c8b7b9fae5329d99d818bd78b2b1686bdb167 Show Diff Stats Showing 2 changed files with 3 additions and 3 deletions. inst/tests/test-summarise.r View file @ 864c8b7 幸 €€ -69,7 +69,7 €€ test that("summarise handles constants (#153)", { 69 69 today <- Sys.Date() 70 70 now <- Sys.time() 71 71 72 - res <- summarise(df, int = 1L, num = 1.0,</p> 72 + res <- summarise(tbl_df(df), int = 1L, num = 1.0, 73 str = "foo", bool = TRUE, date = today, time = now) 74 74 expect equal(res\$int, 1L) 75 75 expect equal(res\$num, 1.0) View file @ 864c8b7 4 src/dplyr.cpp 岩 00 -1024,14 +1024,14 00 SEXP summarise not grouped(DataFrame df, List args, Environment env) { 1024 1024 Rcpp::Shelter<SEXP> ; 1025 1025 for(int i=0; i<nexpr; i++){ 1026 1026 SEXP name = names[i] ; 1027 1028 boost::scoped ptr<Result> res(get handler(args[i], subsets, env)) ; 1027 + Result* res = get_handler(args[i], subsets, env);

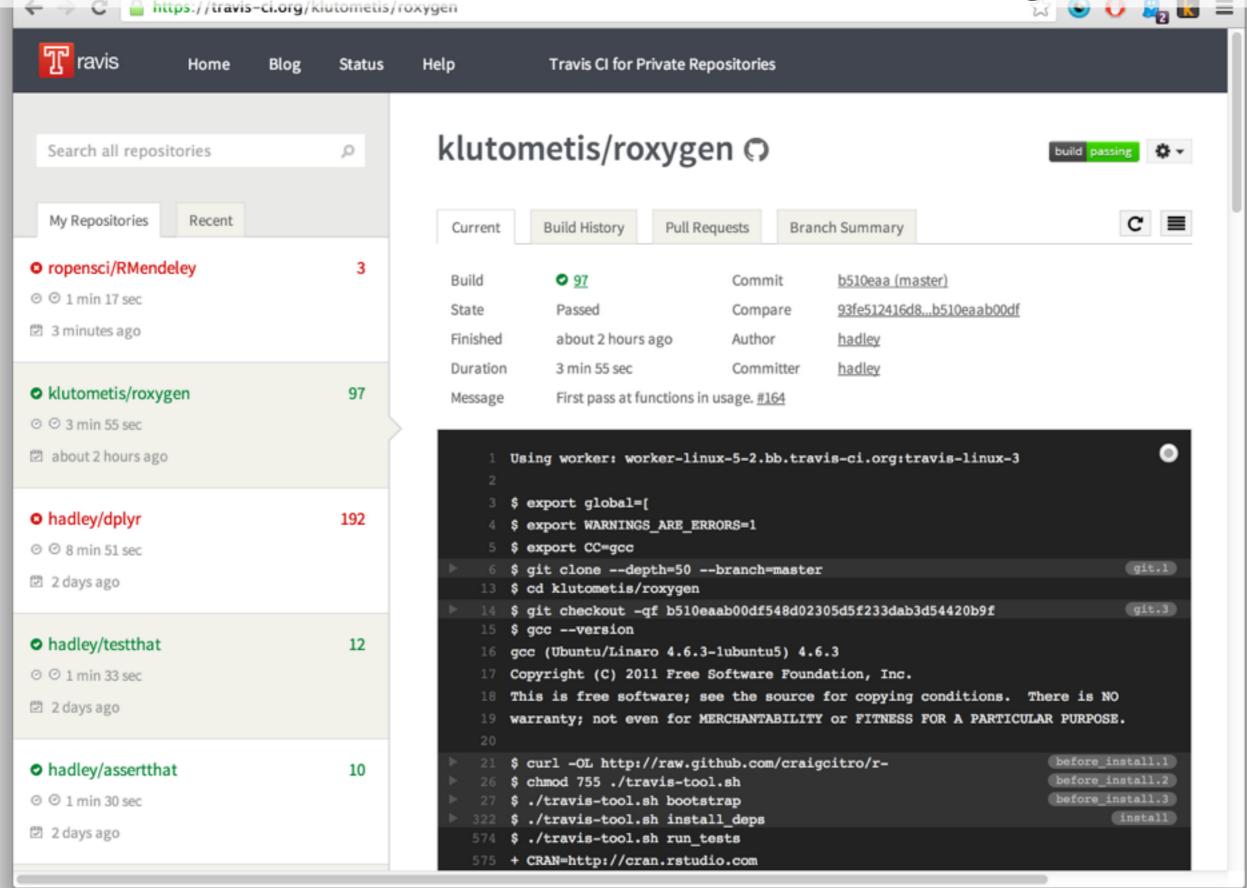


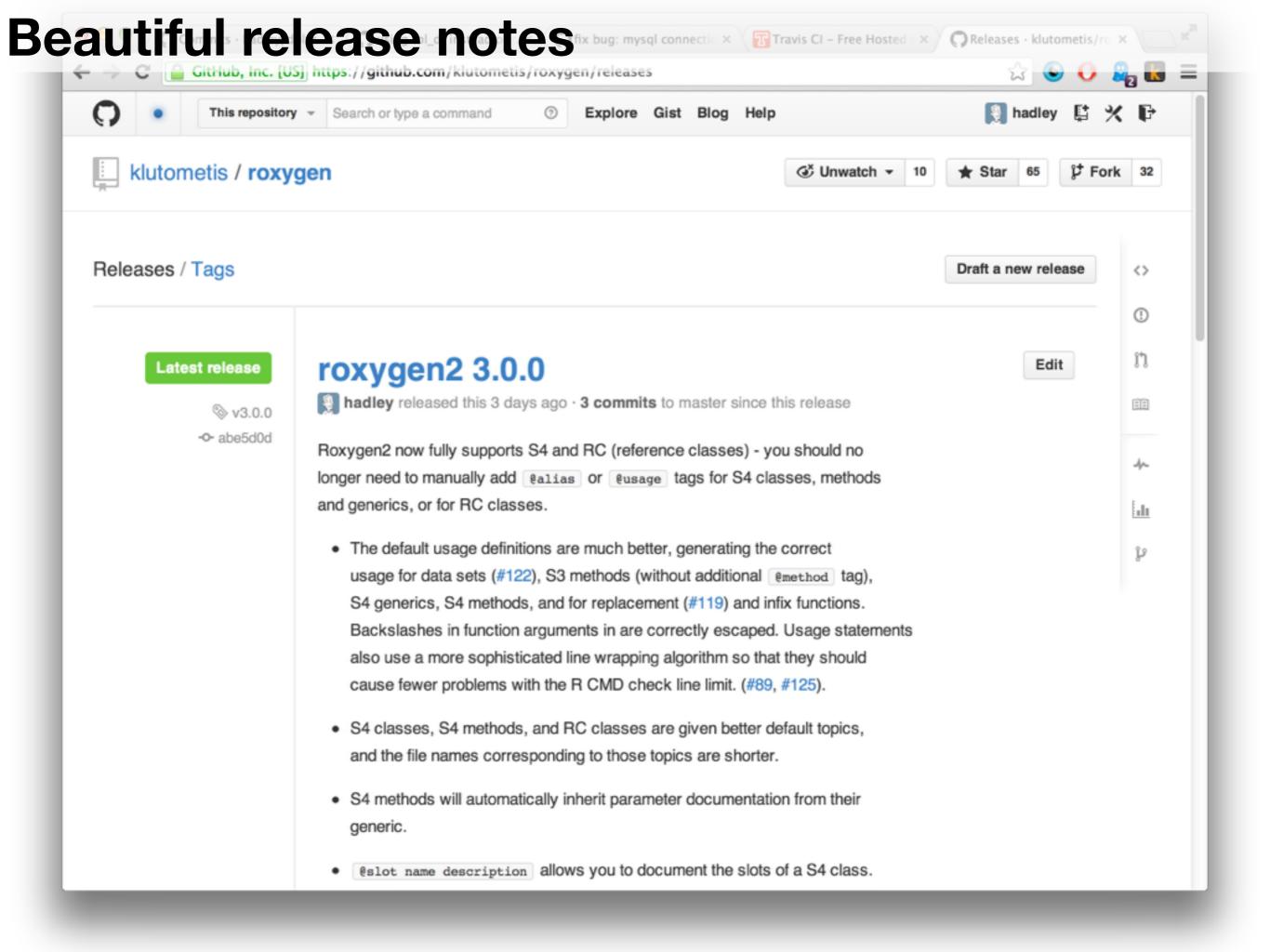






Find out as soon as you break something





Demo

Easily contribute small fixes

https://github.com/mjockers/syuzhet

My advice

Learning Git & Github

- Is frustrating at first & there's a lot of new vocab to learn.
- Pays off by increasing the pace of your coding because you don't have to worry as much about mistakes.
- Pays off by connecting you with a community of like minded people who can help make your code better.

http://r-pkgs.had.co.nz/git.html