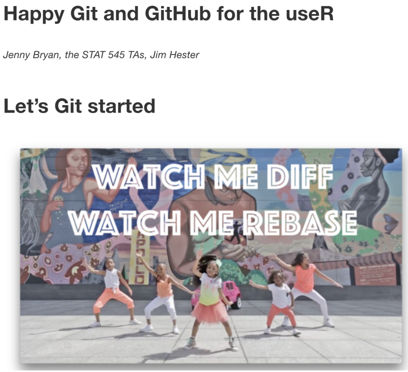
Setup

1. Go to www.github.com and make a free account

2. Make sure you have a recent version (v1.1 or later) of RStudio https://www.rstudio.com/products/rstudio/download/#download

3. Keep www.happygitwithr.com open

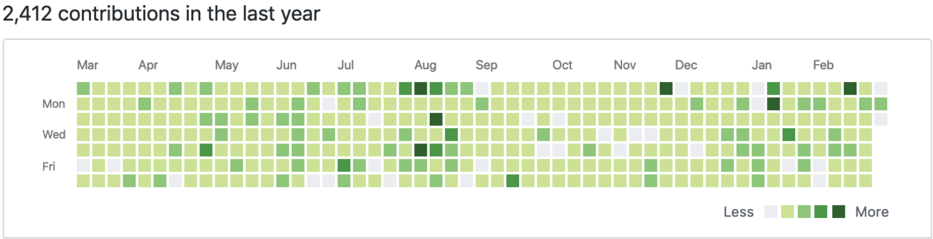
4. Download these slides via: https://github.com/kuriwaki/github-demo/raw/master/presentation-slides/kuriwaki_github.pdf



Happy Git and GitHub for the useR

Jenny Bryan, the STAT 545 TAs, Jim Hester

Let's Git started



2,412 contributions in the last year

Introduction to git for social science students

(not software developers)

Shiro Kuriwaki

March 5, 2019

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: Political data analytics (where I learned git from Annie Wang)

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: Political data analytics (where I learned git from Annie Wang)
- I do some software development,
- but most of my work is applied ("substantive")

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: Political data analytics (where I learned git from Annie Wang)
- I do some software development,
- but most of my work is applied ("substantive")

My perspective

- Version control is mandatory for programmers (and professional data scientists)

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: Political data analytics (where I learned git from Annie Wang)
- I do some software development,
- but most of my work is applied ("substantive")

My perspective

- Version control is mandatory for programmers (and professional data scientists)
- but does it make sense for *applied* researchers who ...
- work with datasets that are

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: Political data analytics (where I learned git from Annie Wang)
- I do some software development,
- but most of my work is applied ("substantive")

My perspective

- Version control is mandatory for programmers (and professional data scientists)
- but does it make sense for *applied* researchers who ...
- work with datasets that are **large**,

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: Political data analytics (where I learned git from Annie Wang)
- I do some software development,
- but most of my work is applied ("substantive")

My perspective

- Version control is mandatory for programmers (and professional data scientists)
- but does it make sense for *applied* researchers who ...
- work with datasets that are **large**, **unstructured**,

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: Political data analytics (where I learned git from Annie Wang)
- I do some software development,
- but most of my work is applied ("substantive")

My perspective

- Version control is mandatory for programmers (and professional data scientists)
- but does it make sense for *applied* researchers who ...
- work with datasets that are **large**, **unstructured**, **prone to change**,

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: Political data analytics (where I learned git from Annie Wang)
- I do some software development,
- but most of my work is applied ("substantive")

My perspective

- Version control is mandatory for programmers (and professional data scientists)
- but does it make sense for *applied* researchers who ...
- work with datasets that are **large**, **unstructured**, **prone to change**, **with collaborators**

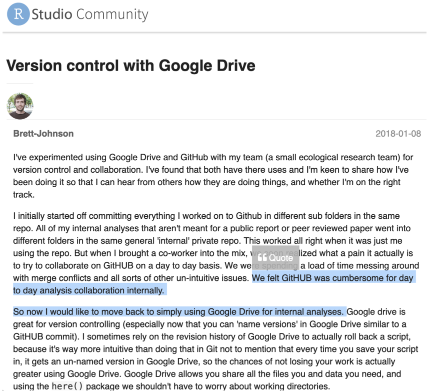
Setting Expectations: Is it worth it?

What do Gentzkow and Shapiro say?

Definitely:

> *"It will probably take you a couple days to set up a repository and learn how you want to interact with [Version Control]. You will break even on that time investment within a month or two."*[1]

but also see[2]



RStudio Community

Version control with Google Drive

Brett-Johnson 2018-01-08

I've experimented using Google Drive and GitHub with my team (a small ecological research team) for version control and collaboration. I've found that both have there uses and I'm keen to share how I've been doing it so that I can hear from others how they are doing things, and whether I'm on the right track.

I initially started off committing everything I worked on to Github in different sub folders in the same repo. All of my internal analyses that aren't meant for a public report or peer reviewed paper went into different folders in the same general 'internal' private repo. This worked all right when it was just me using the repo. But when I brought a co-worker into the mix, we soon realized what a pain it actually is to try to collaborate on GitHUB on a day to day basis. We were spending a load of time messing around with merge conflicts and all sorts of other un-intuitive issues. We felt GitHUB was cumbersome for day to day analysis collaboration internally.

So now I would like to move back to simply using Google Drive for internal analyses. Google drive is great for version controlling (especially now that you can 'name versions' in Google Drive similar to a GitHUB commit). I sometimes rely on the revision history of Google Drive to actually roll back a script, because it's way more intuitive than doing that in Git not to mention that every time you save your script in, it gets an un-named version in Google Drive, so the chances of not losing your work is actually greater using Google Drive. Google Drive allows you share all the files you and data you need, and using the `here()` package we shouldn't have to worry about working directories.

[1] Code and Data for Social Sciences: A Practioners Guide. 2014. https://perma.cc/5J9D-BTD6. Although I'm not sure about learning version control in "a couple of days" (I certainly couldn't!), I can guarantee reading their guide in its entirety *is* a time investment you'll break even on immediately.

[2] https://community.rstudio.com/t/version-control-with-google-drive/4032

My (recommended) setup

Terminology

- **Git** is a particular type of software for version control (Subversion is another)

- **GitHub** is an app (recently bought by Microsoft) to host git on the web (Bitbucket is another)

- A **desktop client** is an app that connects a webhost like Github to your computer and facilitates simple tasks (here **I** use **RStudio**, there are many others)

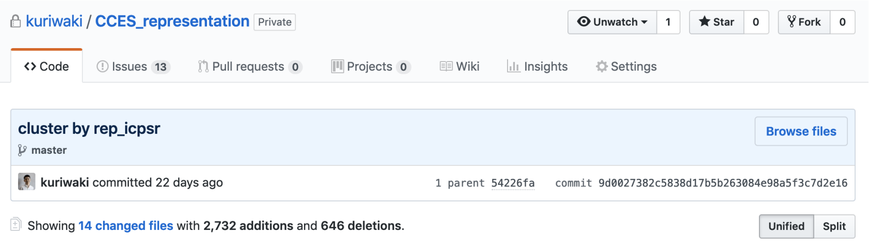
- A **repository** is the fundamental unit of a version control, like a project folder.

Terminology

- **Git** is a particular type of software for version control (Subversion is another)
- **GitHub** is an app (recently bought by Microsoft) to host git on the web (Bitbucket is another)
- A **desktop client** is an app that connects a webhost like Github to your computer and facilitates simple tasks (here I use **RStudio**, there are many others)
- A **repository** is the fundamental unit of a version control, like a project folder. Do not make a repository within a repository!

Keep Track of how your results changed

Problem: You tweak a regression specification and re-run your script, re-writing dozens of tables. You need to know how much your results changed



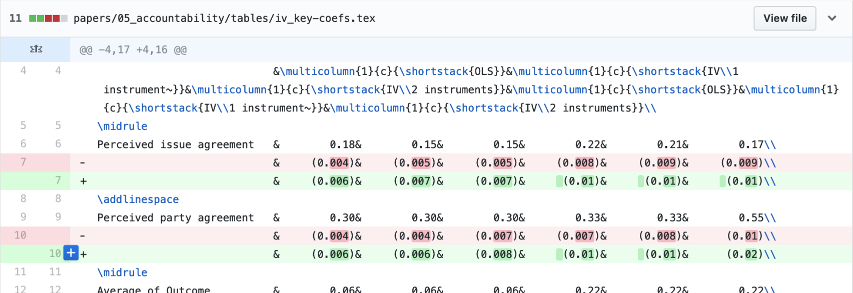
cluster by rep_icpsr

master

kuriwaki committed 22 days ago   1 parent 54226fa   commit 9d0027382c5838d17b5b263084e98a5f3c7d2e16

Showing **14 changed files** with **2,732 additions** and **646 deletions.**

...

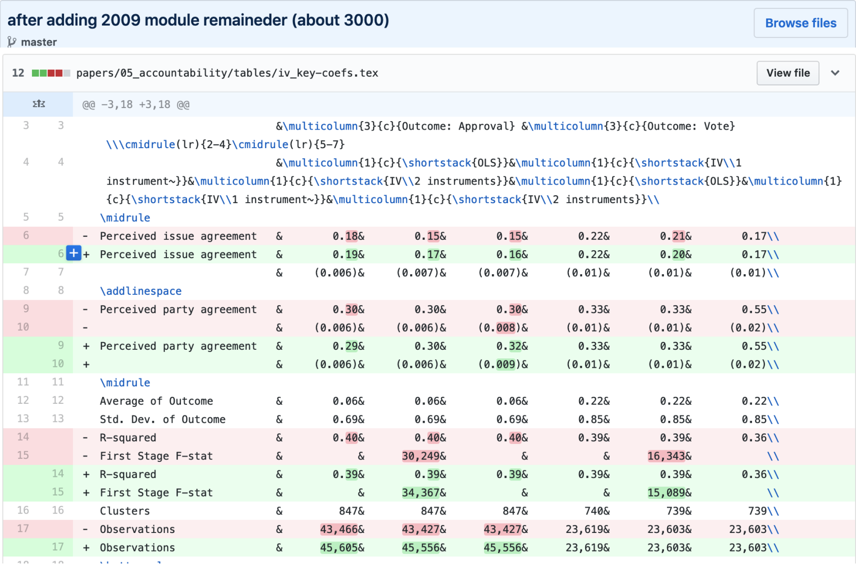


11 🟩🟩🟥🟥 papers/05_accountability/tables/iv_key-coefs.tex

```
@@ -4,17 +4,16 @@
                          &\multicolumn{1}{c}{\shortstack{OLS}}&\multicolumn{1}{c}{\shortstack{IV\\1
   instrument~}}&\multicolumn{1}{c}{\shortstack{IV\\2 instruments}}&\multicolumn{1}{c}{\shortstack{OLS}}&\multicolumn{1}
   {c}{\shortstack{IV\\1 instrument~}}&\multicolumn{1}{c}{\shortstack{IV\\2 instruments}}\\
  \midrule
  Perceived issue agreement  &       0.18&       0.15&       0.15&       0.22&       0.21&       0.17\\
-                            &    (0.004)&    (0.005)&    (0.005)&    (0.008)&    (0.009)&    (0.009)\\
+                            &    (0.006)&    (0.007)&    (0.007)&     (0.01)&     (0.01)&     (0.01)\\
  \addlinespace
  Perceived party agreement  &       0.30&       0.30&       0.30&       0.33&       0.33&       0.55\\
-                            &    (0.004)&    (0.004)&    (0.007)&    (0.007)&    (0.008)&     (0.01)\\
+                            &    (0.006)&    (0.006)&    (0.008)&     (0.01)&     (0.01)&     (0.02)\\
  \midrule
  Average of Outcome         &       0.06&       0.06&       0.06&       0.22&       0.22&       0.22\\
```

Keep Track of how your results changed

You collect more data and re-run the regressions. Now how did the results change?



after adding 2009 module remaineder (about 3000)

papers/05_accountability/tables/iv_key-coefs.tex

```
@@ -3,18 +3,18 @@
                    &\multicolumn{3}{c}{Outcome: Approval} &\multicolumn{3}{c}{Outcome: Vote}\\\cmidrule(lr){2-4}\cmidrule(lr){5-7}
                    &\multicolumn{1}{c}{\shortstack{OLS}}&\multicolumn{1}{c}{\shortstack{IV\\1 instrument~}}&\multicolumn{1}{c}{\shortstack{IV\\2 instruments}}&\multicolumn{1}{c}{\shortstack{OLS}}&\multicolumn{1}{c}{\shortstack{IV\\1 instrument~}}&\multicolumn{1}{c}{\shortstack{IV\\2 instruments}}\\
 \midrule
-Perceived issue agreement &       0.18&       0.15&       0.15&       0.22&       0.21&       0.17\\
+Perceived issue agreement &       0.19&       0.17&       0.16&       0.22&       0.20&       0.17\\
                          &    (0.006)&    (0.007)&    (0.007)&     (0.01)&     (0.01)&     (0.01)\\
 \addlinespace
-Perceived party agreement &       0.30&       0.30&       0.30&       0.33&       0.33&       0.55\\
-                          &    (0.006)&    (0.006)&    (0.008)&     (0.01)&     (0.01)&     (0.02)\\
+Perceived party agreement &       0.29&       0.30&       0.32&       0.33&       0.33&       0.55\\
+                          &    (0.006)&    (0.006)&    (0.009)&     (0.01)&     (0.01)&     (0.02)\\
 \midrule
 Average of Outcome        &       0.06&       0.06&       0.06&       0.22&       0.22&       0.22\\
 Std. Dev. of Outcome      &       0.69&       0.69&       0.69&       0.85&       0.85&       0.85\\
-R-squared                 &       0.40&       0.40&       0.40&       0.39&       0.39&       0.36\\
-First Stage F-stat        &           &     30,249&           &           &     16,343&           \\
+R-squared                 &       0.39&       0.39&       0.39&       0.39&       0.39&       0.36\\
+First Stage F-stat        &           &     34,367&           &           &     15,089&           \\
 Clusters                  &        847&        847&        847&        740&        739&        739\\
-Observations              &     43,466&     43,427&     43,427&     23,619&     23,603&     23,603\\
+Observations              &     45,605&     45,556&     45,556&     23,619&     23,603&     23,603\\
```

Tracking your text changes

Problem: You start writing up your paper, `draft.tex`

- The next day, you make a new draft. Do you overwrite?

Tracking your text changes

Problem: You start writing up your paper, `draft.tex`

- The next day, you make a new draft. Do you overwrite?
- Or do you call it `draft_0305.tex` ? `draft_03052019.tex`?

Tracking your text changes

Problem: You start writing up your paper, `draft.tex`

- The next day, you make a new draft. Do you overwrite?
- Or do you call it `draft_0305.tex` ? `draft_03052019.tex`?
- The next week, you find a single typo. Do you “Save As” with a new date?

Tracking your text changes

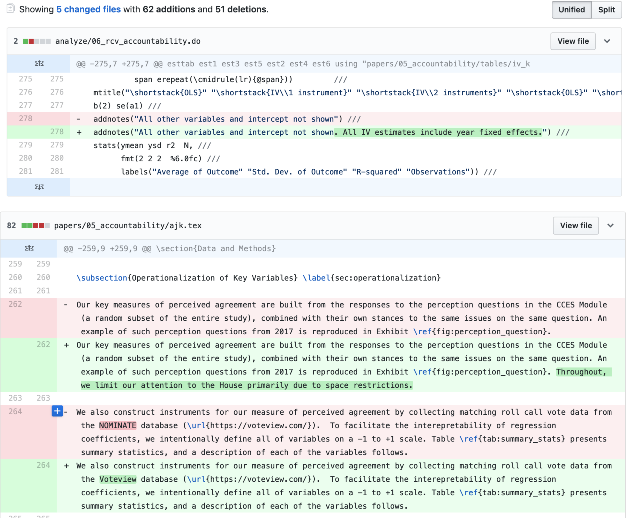
Problem: You start writing up your paper, `draft.tex`

- The next day, you make a new draft. Do you overwrite?
- Or do you call it `draft_0305.tex` ? `draft_03052019.tex`?
- The next week, you find a single typo. Do you “Save As” with a new date?
- Three weeks later, you return to your paper. Your computer indicates that the file named `draft_0305.tex` was “Last modified March 12, 2019”.

Tracking your text changes

Problem: You start writing up your paper, `draft.tex`

- The next day, you make a new draft. Do you overwrite?
- Or do you call it `draft_0305.tex` ? `draft_03052019.tex`?
- The next week, you find a single typo. Do you “Save As” with a new date?
- Three weeks later, you return to your paper. Your computer indicates that the file named `draft_0305.tex` was “Last modified March 12, 2019”.



Showing **5 changed files** with **62 additions** and **51 deletions**.

2 analyze/06_rcv_accountability.do

```
@@ -275,7 +275,7 @@ esttab est1 est3 est5 est2 est4 est6 using "papers/05_accountability/tables/iv_k
               span erepeat(\cmidrule(lr){@span}))         ///
     mtitle("\shortstack{OLS}" "\shortstack{IV\\1 instrument}" "\shortstack{IV\\2 instruments}" "\shortstack{OLS}" "\short
     b(2) se(a1) ///
-    addnotes("All other variables and intercept not shown") ///
+    addnotes("All other variables and intercept not shown. All IV estimates include year fixed effects.") ///
     stats(ymean ysd r2  N, ///
           fmt(2 2 2  %6.0fc) ///
           labels("Average of Outcome" "Std. Dev. of Outcome" "R-squared" "Observations")) ///
```

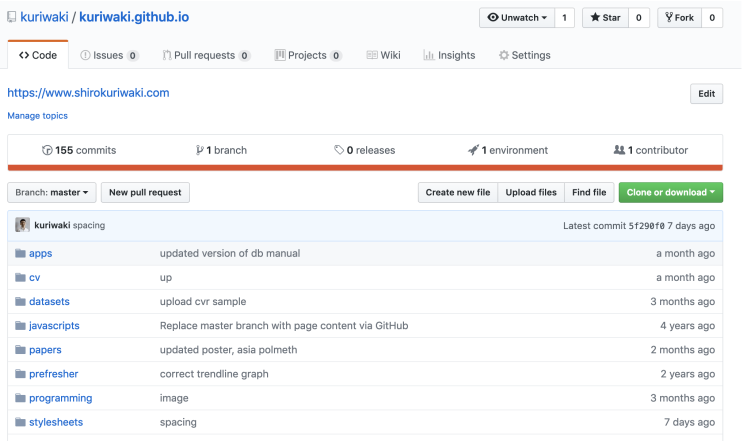
82 papers/05_accountability/ajk.tex

```
@@ -259,9 +259,9 @@ \section{Data and Methods}

  \subsection{Operationalization of Key Variables} \label{sec:operationalization}

- Our key measures of perceived agreement are built from the responses to the perception questions in the CCES Module
   (a random subset of the entire study), combined with their own stances to the same issues on the same question. An
   example of such perception questions from 2017 is reproduced in Exhibit \ref{fig:perception_question}.
+ Our key measures of perceived agreement are built from the responses to the perception questions in the CCES Module
   (a random subset of the entire study), combined with their own stances to the same issues on the same question. An
   example of such perception questions from 2017 is reproduced in Exhibit \ref{fig:perception_question}. Throughout,
   we limit our attention to the House primarily due to space restrictions.

- We also construct instruments for our measure of perceived agreement by collecting matching roll call vote data from
   the NOMINATE database (\url{https://voteview.com/}).  To facilitate the interepretability of regression
   coefficients, we intentionally define all of variables on a -1 to +1 scale. Table \ref{tab:summary_stats} presents
   summary statistics, and a description of each of the variables follows.
+ We also construct instruments for our measure of perceived agreement by collecting matching roll call vote data from
   the Voteview database (\url{https://voteview.com/}).  To facilitate the interepretability of regression
   coefficients, we intentionally define all of variables on a -1 to +1 scale. Table \ref{tab:summary_stats} presents
   summary statistics, and a description of each of the variables follows.
```

And more cool stuff like

Getting a free, customizable, add-free website

(instead of a click-and-drag Wordpress/Squarespace website)



kuriwaki / kuriwaki.github.io

https://www.shirokuriwaki.com

Manage topics

155 commits 1 branch 0 releases 1 environment 1 contributor

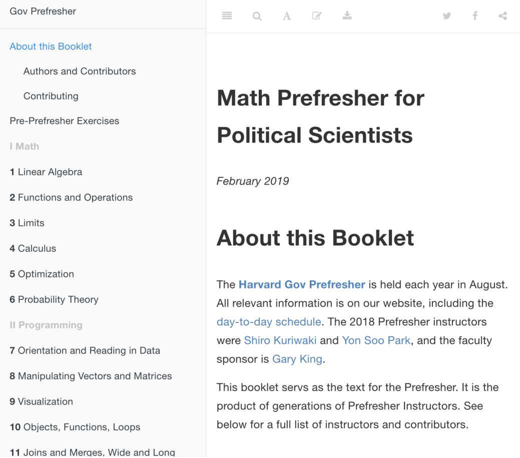
kuriwaki spacing Latest commit 5f290f0 7 days ago

apps	updated version of db manual	a month ago
cv	up	a month ago
datasets	upload cvr sample	3 months ago
javascripts	Replace master branch with page content via GitHub	4 years ago
papers	updated poster, asia polmeth	2 months ago
prefresher	correct trendline graph	2 years ago
programming	image	3 months ago
stylesheets	spacing	7 days ago

And more cool stuff like

Work on a collaborative workbook

(instead of needing to add people to your Dropbox)



Gov Prefresher

Math Prefresher for Political Scientists

February 2019

About this Booklet

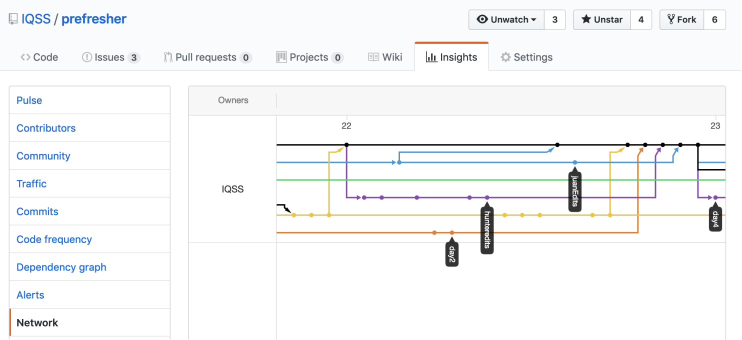
The **Harvard Gov Prefresher** is held each year in August. All relevant information is on our website, including the day-to-day schedule. The 2018 Prefresher instructors were Shiro Kuriwaki and Yon Soo Park, and the faculty sponsor is Gary King.

This booklet servs as the text for the Prefresher. It is the product of generations of Prefresher Instructors. See below for a full list of instructors and contributors.

And more cool stuff like

Work on a collaborative workbook

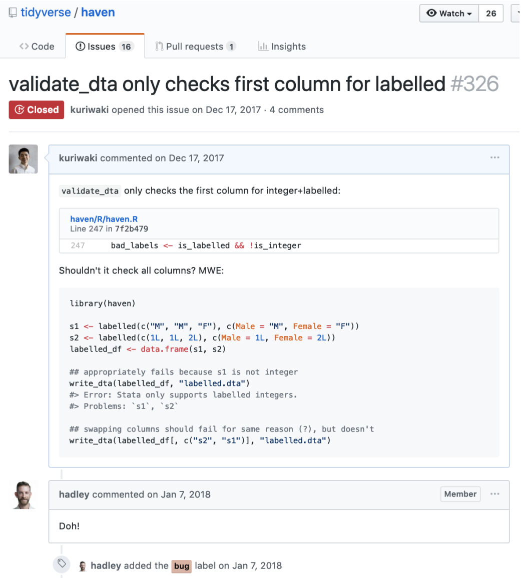
(instead of needing to add people to your Dropbox)



And more cool stuff like

Contributing to / getting the latest on actual software packages

(Github issues is the de facto communication of open-source developers)



validate_dta only checks first column for labelled #326

Closed kuriwaki opened this issue on Dec 17, 2017 · 4 comments

kuriwaki commented on Dec 17, 2017

`validate_dta` only checks the first column for integer+labelled:

haven/R/haven.R
Line 247 in 7f2b479

```
247       bad_labels <- is_labelled && !is_integer
```

Shouldn't it check all columns? MWE:

```
library(haven)

s1 <- labelled(c("M", "M", "F"), c(Male = "M", Female = "F"))
s2 <- labelled(c(1L, 1L, 2L), c(Male = 1L, Female = 2L))
labelled_df <- data.frame(s1, s2)

## appropriately fails because s1 is not integer
write_dta(labelled_df, "labelled.dta")
#> Error: Stata only supports labelled integers.
#> Problems: `s1`, `s2`

## swapping columns should fail for same reason (?), but doesn't
write_dta(labelled_df[, c("s2", "s1")], "labelled.dta")
```

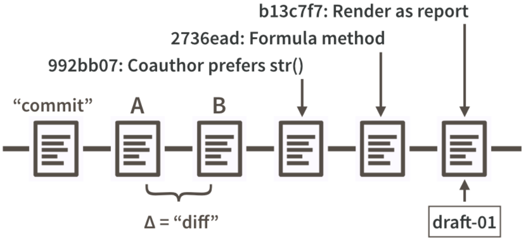
hadley commented on Jan 7, 2018 Member

Doh!

hadley added the bug label on Jan 7, 2018

Terminology

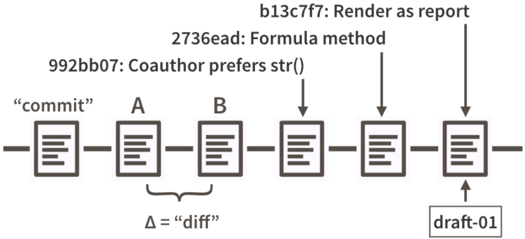
- Files increment by **commit**s. The line-by-line changes from commits are called **diffs**.



Terminology

- Files increment by **commit**s. The line-by-line changes from commits are called **diffs**.

- Commits have a human-readable **message**, and a serial code called a **SHA** (like `992bb07`).

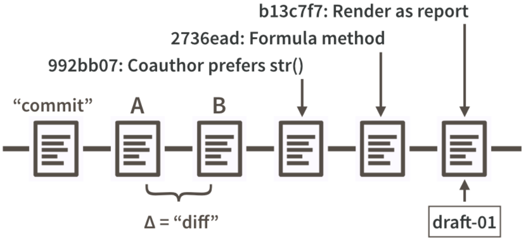


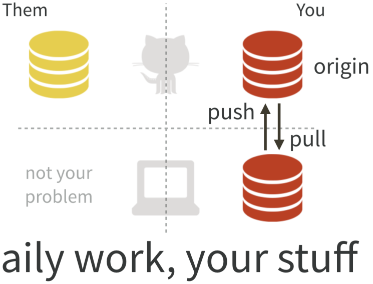
Terminology

- Files increment by **commits**. The line-by-line changes from commits are called **diffs**.

- Commits have a human-readable **message**, and a serial code called a **SHA** (like `992bb07`).

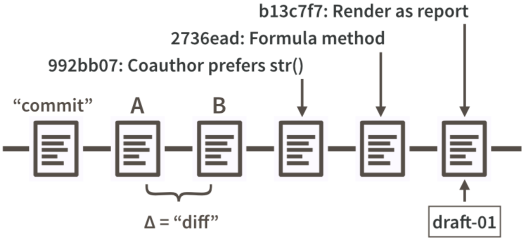
- At least two copies of your repository exist: the **local** on your computer, and a **remote** (hosted on Github, with URL `https://github.com/user/repo.git`), which has the name **origin**

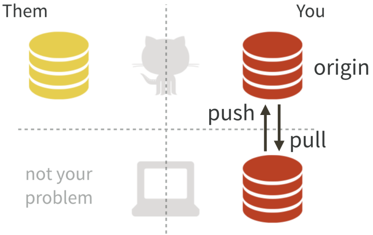




Terminology

- Files increment by **commits**. The line-by-line changes from commits are called **diffs**.
- Commits have a human-readable **message**, and a serial code called a **SHA** (like `992bb07`).
- At least two copies of your repository exist: the **local** on your computer, and a **remote** (hosted on Github, with URL `https://github.com/user/repo.git`), which has the name **origin**
- Once you make commits on your local, you **push** them to your remote. (The opposite of this is a **pull**)





daily work, your stuff

Now, some caveats

Only plain-text files get tracked

This rules out:

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs,

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg's,

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg’s, ,
Microsoft Word,

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg’s, , Microsoft Word, any Microsoft Office product,

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg’s, , Microsoft Word, any Microsoft Office product,Google Docs

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg's, , Microsoft Word, any Microsoft Office product,Google Docs, `.sav`, `.por`, `.dta`, `.Rds`, `RData`...

Which requires a switch to plain-text

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg’s, , Microsoft Word, any Microsoft Office product,Google Docs, `.sav`, `.por`, `.dta`, `.Rds`, `RData`...

Which requires a switch to plain-text

Markdown (`.md`)and TeX (`.tex`) for writing,

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg's, , Microsoft Word, any Microsoft Office product,Google Docs, `.sav`, `.por`, `.dta`, `.Rds`, `RData`...

Which requires a switch to plain-text

Markdown (`.md`)and TeX (`.tex`) for writing, Code (`.R`, `.py`) centered dataset generation, small datasets in `.csv` or `.txt`,

Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg's, , Microsoft Word, any Microsoft Office product,Google Docs, `.sav`, `.por`, `.dta`, `.Rds`, `RData`...

Which requires a switch to plain-text

Markdown (`.md`)and TeX (`.tex`) for writing, Code (`.R`, `.py`) centered dataset generation, small datasets in `.csv` or `.txt`, interweavers like `.Rmd`,

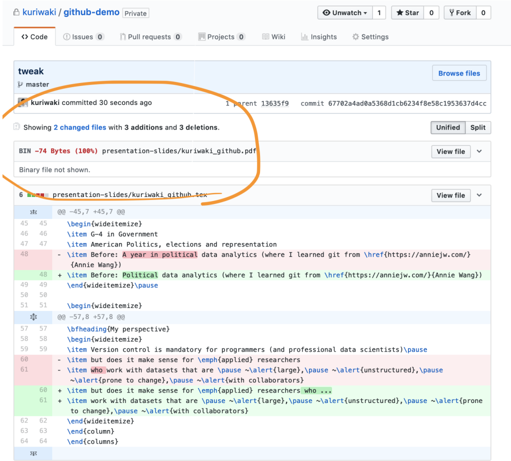
Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg's, , Microsoft Word, any Microsoft Office product,Google Docs, `.sav`, `.por`, `.dta`, `.Rds`, `RData` ...

Which requires a switch to plain-text

Markdown (`.md`)and TeX (`.tex`) for writing, Code (`.R`, `.py`) centered dataset generation, small datasets in `.csv` or `.txt`, interweavers like `.Rmd`, *Kieran Healy, "The Plain Person's Guide to Plain Text Social Science."*



tweak
master
kuriwaki committed 30 seconds ago 1 parent 13635f9 commit 67702a4ad0a5368d1cb6234f8e58c1953637d4cc

Showing 2 changed files with 3 additions and 3 deletions.

```
BIN -74 Bytes (100%) presentation-slides/kuriwaki_github.pdf
Binary file not shown.

6 presentation-slides/kuriwaki_github.tex

@@ -45,7 +45,7 @@
   \begin{wideitemize}
   \item G-4 in Government
   \item American Politics, elections and representation
 - \item Before: A year in political data analytics (where I learned git from \href{https://anniejw.com/}
    {Annie Wang})
 + \item Before: Political data analytics (where I learned git from \href{https://anniejw.com/}{Annie Wang})
   \end{wideitemize}\pause

   \begin{wideitemize}
@@ -57,8 +57,8 @@
   \bfheading{My perspective}
   \begin{wideitemize}
   \item Version control is mandatory for programmers (and professional data scientists)\pause
 - \item but does it make sense for \emph{applied} researchers
 - \item who work with datasets that are \pause ~\alert{large},\pause ~\alert{unstructured},\pause
    ~\alert{prone to change},\pause ~\alert{with collaborators}
 + \item but does it make sense for \emph{applied} researchers who ...
 + \item work with datasets that are \pause ~\alert{large},\pause ~\alert{unstructured},\pause ~\alert{prone
    to change},\pause ~\alert{with collaborators}
   \end{wideitemize}
   \end{column}
   \end{columns}
```

Git is **not** built for storing data!

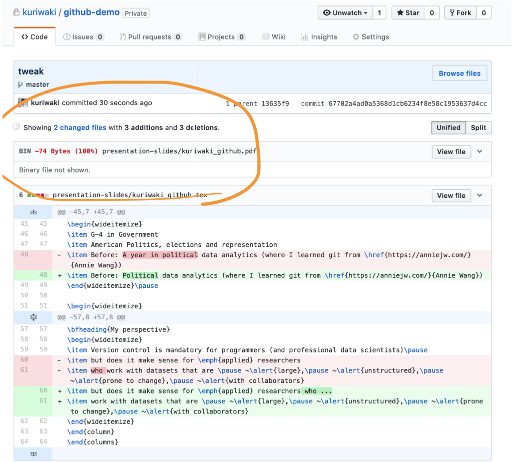
Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg's, , Microsoft Word, any Microsoft Office product,Google Docs, `.sav`, `.por`, `.dta`, `.Rds`, `RData` ...

Which requires a switch to plain-text

Markdown (`.md`)and TeX (`.tex`) for writing, Code (`.R`, `.py`) centered dataset generation, small datasets in `.csv` or `.txt`, interweavers like `.Rmd`, *Kieran Healy, "The Plain Person's Guide to Plain Text Social Science."*



tweak

master

kuriwaki committed 30 seconds ago   1 parent 13635f9   commit 67702a4ad0a5368d1cb6234f8e58c1953637d4cc

Showing 2 changed files with 3 additions and 3 deletions.

```
BIN -74 Bytes (100%) presentation-slides/kuriwaki_github.pdf
Binary file not shown.

6 presentation-slides/kuriwaki_github.tex
@@ -45,7 +45,7 @@
  \begin{wideitemize}
  \item G-4 in Government
  \item American Politics, elections and representation
- \item Before: A year in political data analytics (where I learned git from \href{https://anniejw.com/}
   {Annie Wang})
+ \item Before: Political data analytics (where I learned git from \href{https://anniejw.com/}{Annie Wang})
  \end{wideitemize}\pause

  \begin{wideitemize}
@@ -57,8 +57,8 @@
  \bfheading{My perspective}
  \begin{wideitemize}
  \item Version control is mandatory for programmers (and professional data scientists)\pause
- \item but does it make sense for \emph{applied} researchers
- \item who work with datasets that are \pause ~\alert{large},\pause ~\alert{unstructured},\pause
   ~\alert{prone to change},\pause ~\alert{with collaborators}
+ \item but does it make sense for \emph{applied} researchers who ...
+ \item work with datasets that are \pause ~\alert{large},\pause ~\alert{unstructured},\pause ~\alert{prone
   to change},\pause ~\alert{with collaborators}
  \end{wideitemize}
  \end{column}
  \end{columns}
```

Git is **not** built for storing data!

(Rely on the usual Dropbox / Google Drive / Dataverse / Cloud Servers for that)

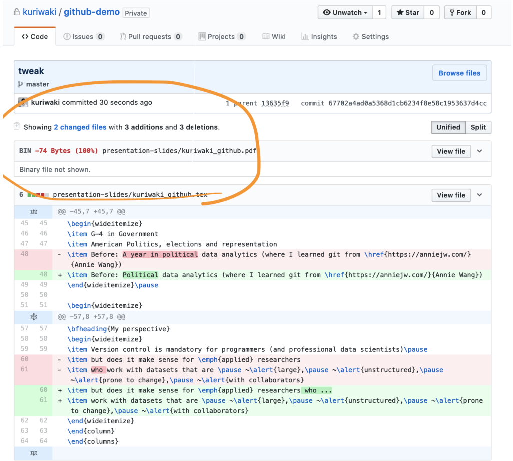
Now, some caveats

Only plain-text files get tracked

This rules out: PDFs, jpg's, , Microsoft Word, any Microsoft Office product,Google Docs, `.sav` , `.por` , `.dta` , `.Rds` , `RData` ...

Which requires a switch to plain-text

Markdown (`.md`)and TeX (`.tex`) for writing, Code (`.R` , `.py`) centered dataset generation, small datasets in `.csv` or `.txt` , interweavers like `.Rmd` ,
Kieran Healy, "The Plain Person's Guide to Plain Text Social Science."



tweak
master
kuriwaki committed 30 seconds ago   1 parent 13635f9   commit 67702a4ad0a5368d1cb6234f8e58c1953637d4cc

Showing 2 changed files with 3 additions and 3 deletions.

BIN -74 Bytes (100%) presentation-slides/kuriwaki_github.pdf

Binary file not shown.

6 presentation-slides/kuriwaki_github.tex

```
@@ -45,7 +45,7 @@
  \begin{wideitemize}
  \item G-4 in Government
  \item American Politics, elections and representation
- \item Before: A year in political data analytics (where I learned git from \href{https://anniejw.com/}
   {Annie Wang})
+ \item Before: Political data analytics (where I learned git from \href{https://anniejw.com/}{Annie Wang})
  \end{wideitemize}\pause

  \begin{wideitemize}
@@ -57,8 +57,8 @@
  \bfheading{My perspective}
  \begin{wideitemize}
  \item Version control is mandatory for programmers (and professional data scientists)\pause
- \item but does it make sense for \emph{applied} researchers
- \item who work with datasets that are \pause ~\alert{large},\pause ~\alert{unstructured},\pause
   ~\alert{prone to change},\pause ~\alert{with collaborators}
+ \item but does it make sense for \emph{applied} researchers who ...
+ \item work with datasets that are \pause ~\alert{large},\pause ~\alert{unstructured},\pause ~\alert{prone
   to change},\pause ~\alert{with collaborators}
  \end{wideitemize}
  \end{column}
  \end{columns}
```

Git is **not** built for storing data!

(Rely on the usual Dropbox / Google Drive / Dataverse / Cloud Servers for that)