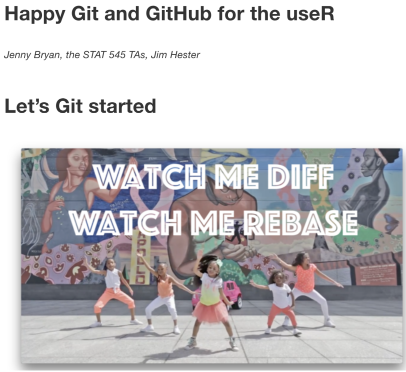
Setup

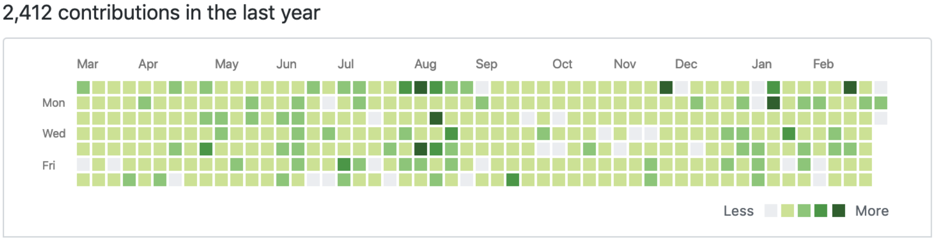
1. Go to `www.github.com` and make a free account
2. Make sure you have a recent version (v1.1 or later) of RStudio `https://www.rstudio.com/products/rstudio/download/#download`
3. Keep `www.happygitwithr.com` open



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: A year in 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: A year in 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: A year in 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

Thanks for having me

About me

- G-4 in Government
- American Politics, elections and representation
- Before: A year in 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
- 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: A year in 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
- 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: A year in 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
- 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: A year in 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
- 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: A year in 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
- but does it make sense for *applied* researchers
- who work with datasets that are **large**, **unstructured**, **prone to change**, **with collaborators**

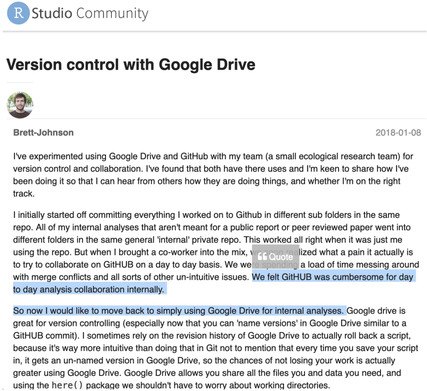
Setting Expectations: Is it worth it?

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!

Summary and wider context

Preference for local taxation and public goods provision is not partisan.

Summary and wider context

Preference for local taxation and public goods provision is not partisan.

This project brings vote choice data into the debate of policy representation in a Federalist system:

“The Increasingly United States”

- State representatives (Rogers 2018) and voters (Tausanovitch and Warshaw 2014) oriented along national partisan lines
- Accelerated by the nationalization of media (Hopkins 2018)
- But other dimensions do exist (in California?) (Snyder 1996, Gerber and Lewis 2004)

“Something for Something”

- Precincts vote for and against redistribution according to their needs (Sances 2018), and school elections hinge on school performance (Payson 2017, Kogan, Lavertu, Peskowitz 2015)
- Individual (voter-file) data reveals insights about who participates and how (Kogan, Lavertu, Peskowitz 2018)