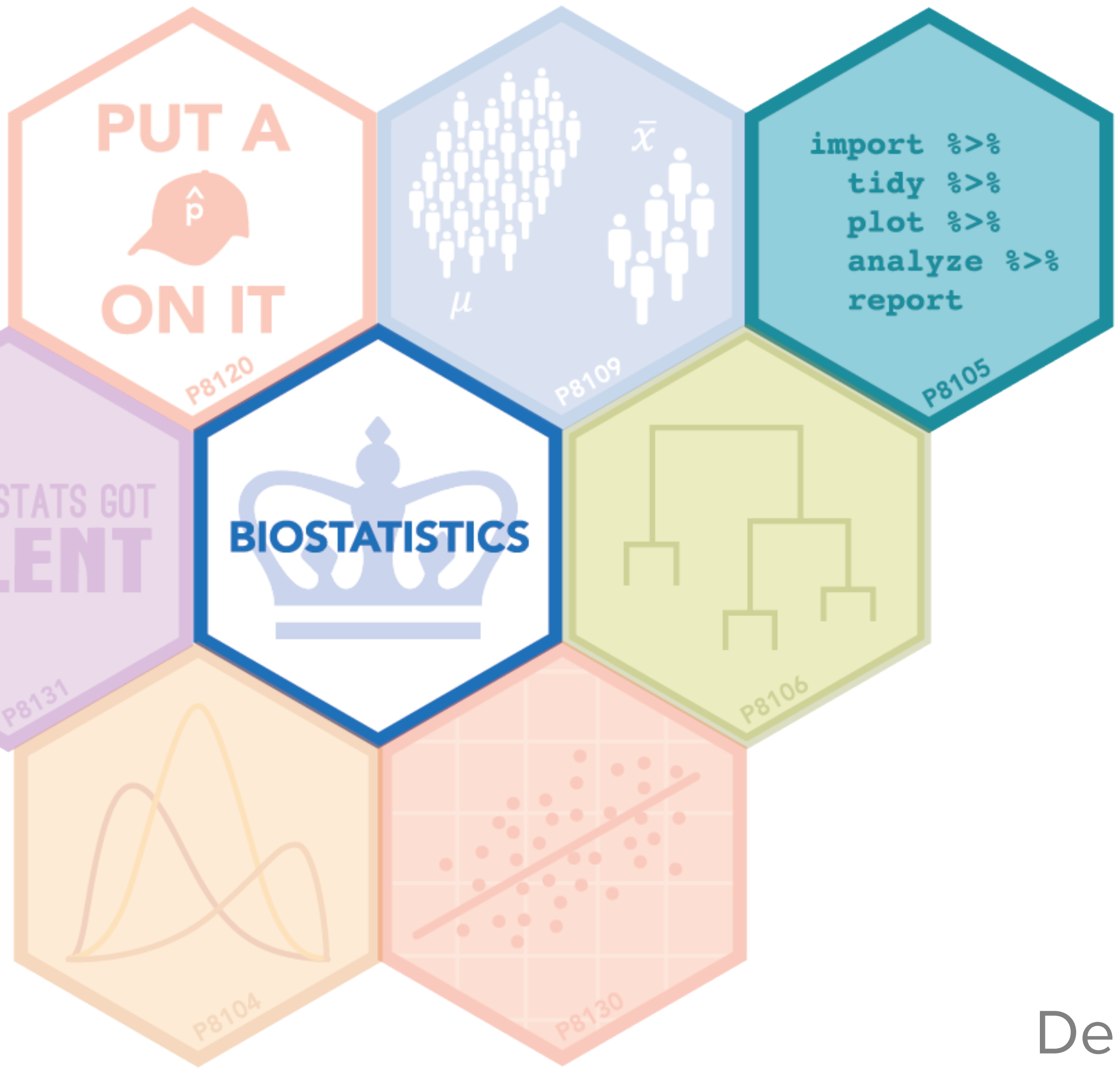


GIT

Jeff Goldsmith, PhD
Department of Biostatistics

Is Git awesome?

- Yes

Is Git awful?

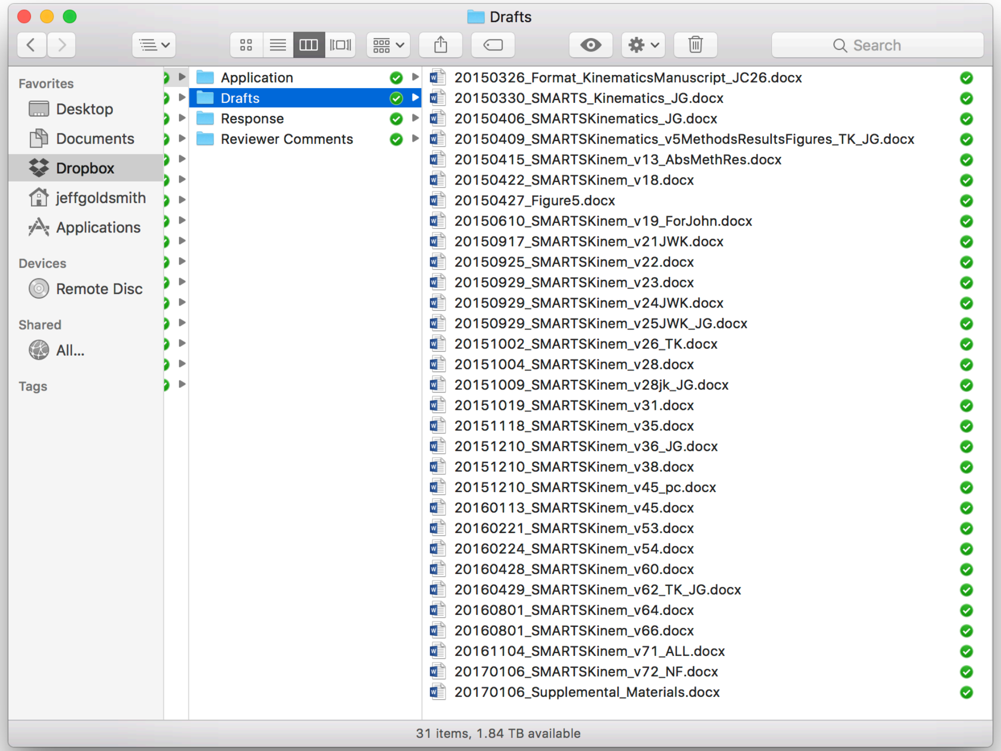
- Also yes

?????

- The good generally outweighs the bad
  - But there is some bad

So … what is Git?

- Kinda like Google Docs / Dropbox / track changes
- The goal is to avoid this:



20150326_Format_KinematicsManuscript_JC26.docx
20150330_SMARTS_Kinematics_JG.docx
20150406_SMARTSKinematics_JG.docx
20150409_SMARTSKinematics_v5MethodsResultsFigures_TK_JG.docx
20150415_SMARTSKinem_v13_AbsMethRes.docx
20150422_SMARTSKinem_v18.docx
20150427_Figure5.docx
20150610_SMARTSKinem_v19_ForJohn.docx
20150917_SMARTSKinem_v21JWK.docx
20150925_SMARTSKinem_v22.docx
20150929_SMARTSKinem_v23.docx
20150929_SMARTSKinem_v24JWK.docx
20150929_SMARTSKinem_v25JWK_JG.docx
20151002_SMARTSKinem_v26_TK.docx
20151004_SMARTSKinem_v28.docx
20151009_SMARTSKinem_v28jk_JG.docx
20151019_SMARTSKinem_v31.docx
20151118_SMARTSKinem_v35.docx
20151210_SMARTSKinem_v36_JG.docx
20151210_SMARTSKinem_v38.docx
20151210_SMARTSKinem_v45_pc.docx
20160113_SMARTSKinem_v45.docx
20160221_SMARTSKinem_v53.docx
20160224_SMARTSKinem_v54.docx
20160428_SMARTSKinem_v60.docx
20160429_SMARTSKinem_v62_TK_JG.docx
20160801_SMARTSKinem_v64.docx
20160801_SMARTSKinem_v66.docx
20161104_SMARTSKinem_v71_ALL.docx
20170106_SMARTSKinem_v72_NF.docx
20170106_Supplemental_Materials.docx

31 items, 1.84 TB available

That still doesn't explain it.

- Git watches repositories – folders / directories – for changes
- It asks that you describe changes when they're made
- It remembers old versions if you need them
- It also keeps an eye out for conflicts, and forces you to resolve them
- It allows multiple people to contribute to the same repository, and does all of the above for everyone at once

And GitHub?

- Git lives on your computer; GitHub is a web-based platform for storing repositories
  - Think DropBox, but with Git in your folders (watching you)
- GitHub is a great platform for disseminating work
  - You can easily create and host reports; websites; R packages; …

What about RStudio

- Git is something you should be doing, and RStudio tries to make it easy for you to do
- R Projects can initialize Git with a mouse click
- Then, everything in the project is being watched

And a Git client?

- Git is a command-line tool
- Git clients let you do most Git-related stuff in a GUI
  - Git client is to git as RStudio is to R

Workflow

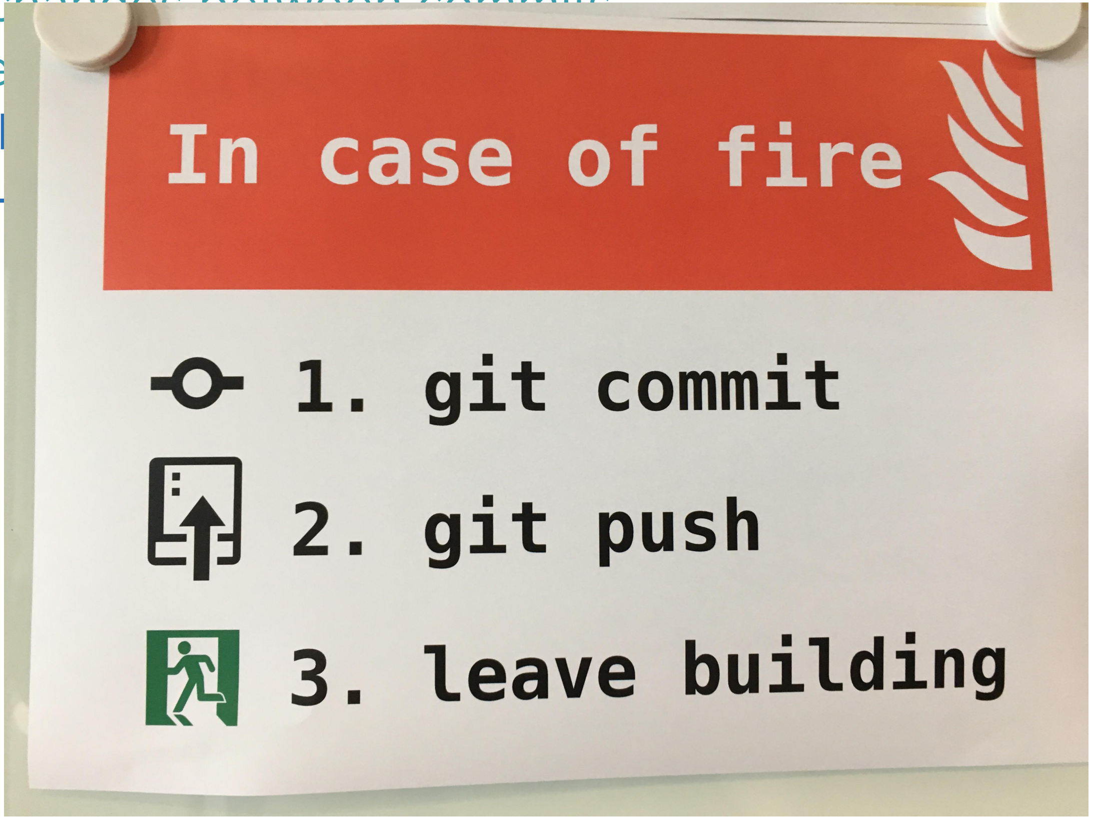
- When starting a new analysis / project / whatever, I
  - Create a directory
  - Start a new R Project in that folder
  - Initialize Git
  - Add to my Git client (and create the GitHub repository)
  - Start doing stuff

"Doing stuff" in a git repo

- You do whatever you would usually do
- Once you've done some amount of stuff, you commit the changes
  - "commit" = "fancy save"
  - Git will keep track of changes between commits
  - Your commit message will summarize what's different
- Then you do more stuff, then you commit, then you do more stuff …
- Push changes to GitHub – more on that soon

"Doing stuff" in a git repo

- You do whatever you would usually do
- Once you've done some amount of stuff, you commit the changes
  - "commit" = "fancy save"
  - Git will keep track of changes between commits
  - Your commit message
- Then you do more stuff, tl
- Push changes to GitHub -



Pros:

- You can revert to earlier commits if you mess something up
- You can quickly review the development process
- You can see what collaborators are doing, where they're doing it, and why
- You're forced to resolve conflicts (two people changing the same thing at the same time) as they arise

Cons:

- There is a lot of overhead, and it’s worst at the beginning
- “Resolving conflicts” can be awful
- Everyone on a project is required to stick with the same development pipeline

Vocab

- Repository
- Commit
- Push / Pull
- Branch / Merge
- Cloning? Forking??

Not going to cover

- Messaging
- Issue tracking

Confidentiality

- You have to watch out for data confidentiality – GitHub is public!