A brief introduction to git & GitHub

Karl Broman & Samuel G. Younkin

Biostatistics & Medical Informatics University of Wisconsin–Madison

github.com/syounkin/GitPrimer

"FINAL".doc



CFINAL.doc!



FINAL_rev.2.doc







FINAL_rev.6.COMMENTS.doc

FINAL_rev.8.comments5. CORRECTIONS.doc











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

WWW.PHDCOMICS.COM

Methods for tracking versions

- Don't keep track
- Save numbered zip files
- ► Formal version control

Suppose it stops working...

- Don't keep track
 - good luck!
- Save numbered zip files
 - Unzip versions and diff
- Formal version control
 - Easy to study changes back in time
 - Easy to jump back and test

Why use formal version control?

- History of changes
- Able to go back
- No worries about breaking things that work
- Merging changes from multiple people

Example repository



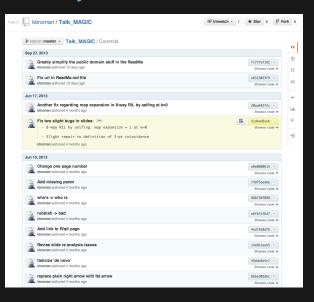
Example repository



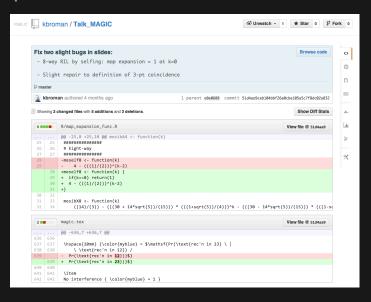
Greatly simplify the public domain stuff in the ReadMe		
kbroman authored 15 days ago		latest commit f1777ef192 🚉
Figs	Add crazy table from preCC paper	4 months ago
Perl	Add lines_of_code_by_version.csv to repository	4 months ago
m R	Another fix regarding map expansion in 8-way RIL by selfing at k=0	4 months ago
gitignore	Add lines_of_code_by_version.csv to repository	4 months ago
Makefile	Revise Readme to link to version for web	4 months ago
ReadMe.md	Greatly simplify the public domain stuff in the ReadMe	15 days ago
magic.tex	Fix two slight bugs in slides:	4 months ago

rights to "MAGIC design and other topics". This work is published from: United States.

Example history



Example commit



What is git?

- Formal version control system
- Developed by Linus Torvalds (developer of Linux)
 - used to manage the source code for Linux
- Tracks any content (but mostly plain text files)
 - source code
 - data analysis projects
 - manuscripts
 - websites
 - presentations

Why use git?

- ▶ It's fast
- You don't need access to a server
- Amazingly good at merging simultaneous changes
- ▶ Everyone's using it

What is GitHub?

- A home for git repositories
- Interface for exploring git repositories
- Real open source
 - immediate, easy access to the code
- Like facebook for programmers
- (Bitbucket.org is an alternative)
 - free private repositories

Why use GitHub?

- It takes care of the server aspects of git
- Graphical user interface for git
 - Exploring code and its history
 - Tracking issues
- Facilitates:
 - Learning from others
 - Seeing what people are up to
 - Contributing to others' code
- Lowers the barrier to collaboration
 - "There's a typo in your documentation." vs.
 "Here's a correction for your documentation."

- Change some files
- See what you've changed

```
git status
git diff
git log
```

Indicate what changes to save

```
git add
```

Commit to those changes

```
git commit
```

- Change some files
- See what you've changed

```
git status
git diff
git log
```

Indicate what changes to save

```
git add
```

Commit to those changes

```
git commit
```

Push the changes to GitHub

```
git push
```

- Change some files
- See what you've changed

```
git status
git diff
git log
```

Indicate what changes to save

```
git add
```

Commit to those changes

```
git commit
```

Push the changes to GitHub

```
git push
```

Pull changes from your collaborator

```
git pull
```

Initialize repository

- Create a working directory
 - For example, ~/GitPrimer
- Initialize it to be a git repository
 - git init
 - Creates subdirectory ~/GitPrimer/.git

```
$ mkdir ~/GitPrimer
$ cd ~/GitPrimer
$ git init
Initialized empty Git repository in ~/GitPrimer/.git/
```

Produce content

▶ Create a README file

```
Welcome to the GitPrimer repository.

Date: Tue Oct 1 14:12:47 CDT 2013

This repository contains source code for a brief git & GitHub tutorial given by Younkin & Broman at the University of Wisconsin-Madison,
Dept. of Biostatistics & Medical Informatics.

Email Samuel Younkin <syounkin@stat.wisc.edu> with questions or comments.
```

Produce content

Or create a README.md file

```
Welcome to the GitPrimer repository.

Date: Tue Oct 1 14:12:47 CDT 2013

This repository contains source code for a brief git & GitHub tutorial given by [Younkin](http://www.stat.wisc.edu/-syounkin/) & [Broman](http://www.biostat.wisc.edu/-kbroman) at the University of Wisconsin-Madison, Dept. of Biostatistics & Medical Informatics.

Email Samuel Younkin <syounkin@stat.wisc.edu> with questions or comments, or submit an [Issue at the GitHub](https://github.com/syounkin/GitPrimer/issues).
```

Incorporate into repository

► Stage the changes using git add

\$ git add README

Incorporate into repository

► Now commit using git commit

```
$ git commit -m "Initial commit of README file"
[master (root-commit) 32c9d01] Initial commit of README file
1 file changed, 14 insertions(+)
create mode 100644 README
```

- ► The -m argument allows one to enter a message
- ▶ Without -m, git will spawn a text editor
- Use a meaningful message
- Message can have multiple lines, but make 1st line an overview

A few points on commits

- Use frequent, small commits
- Don't get out of sync with your collaborators
- Commit the sources, not the derived files (R code not images)
- Use a .gitignore file to indicate files to be ignored

```
*-
manuscript.pdf
Figs/*.pdf
.RData
.RHistory
*.Rout
*.Rout
*.aux
*.log
*.out
```

Removing/moving files

For files that are being tracked by git:

```
Use git rm instead of just rm Use git mv instead of just mv
```

```
$ git rm myfile
$ git mv myfile newname
$ git mv myfile SubDir/
$ git commit
```

Using git on an existing project

- ▶ git init
- ► Set up .gitignore file
- ▶ git status (did you miss any?)
- ▶ git add . (or name files individually)
- ▶ git status (did you miss any?)
- ▶ git commit

- Change some files
- See what you've changed

```
git status
git diff
git log
```

Indicate what changes to save

```
git add
```

Commit to those changes

```
git commit
```

Push the changes to GitHub

```
git push
```

Pull changes from your collaborator

```
git pull
```

Getting started with GitHub

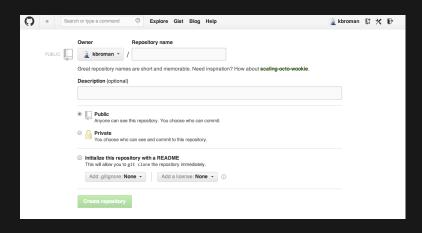
- Get an account
- Set up ssh keys
 - Look for files ~/.ssh/id rsa and ~/.ssh/id rsa.pub
 - ssh-keygen -t rsa -C "your_email@example.com"
 - Copy contents of ~/.ssh/id_rsa.pub
- Add SSH key at GitHub
 - Account settings
 - SSH Keys
 - Add SSH key
 - Paste contents of ~/.ssh/id_rsa.pub
- Similar thing at BitBucket

Set up GitHub repository

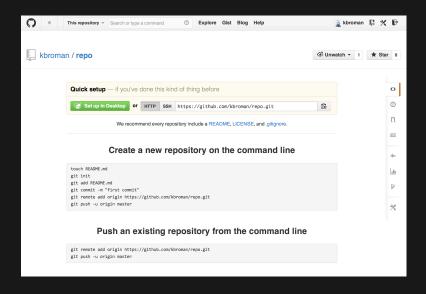
- Click the "Create a new repo" button
- Give it a name and description
- Click the "Create repository" button
- Back at the command line:

```
git remote add origin git@github.com:username/repo git push -u origin master
```

Set up GitHub repository



Set up GitHub repository



Configuration file

Part of a .git/config file:

```
[remote "origin"]
url = git@github.com:kbroman/qtl.git
fetch = +refs/heads/*:refs/remotes/origin/*

[branch "master"]
remote = origin
merge = refs/heads/master

[remote "brian"]
url = git://github.com/byandell/qtl.git
fetch = +refs/heads/*:refs/remotes/brian/*
```

Issues and pull requests

- Problem with or suggestion for someone's code?
 - Point it out as an Issue
- Even better: Provide a fix
 - Fork
 - Clone
 - Modify
 - Commit
 - Push
 - Submit a Pull Request

Suggest a change to a repo

Go to the repository:

http://github.com/someone/repo

► Fork the repository

Click the "Fork" button

Clone your version of it

git clone git@github.com:username/repo

- ► Change things locally, git add, git commit
- Push your changes to your GitHub repository git push
- Go to your GitHub repository
- Click "Pull Requests" and "New pull request"

Pulling a friend's changes

Add a connection

```
git remote add friend git://github.com/friend/repo
```

► Pull the changes

```
git pull friend master
```

Push them back to your GitHub repo

```
git push
```

Merge conflicts

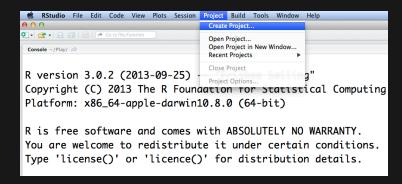
Sometimes after git pull friend master

```
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

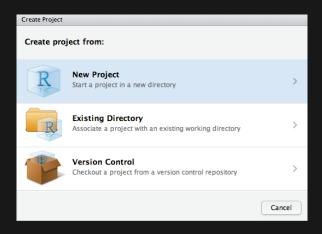
Inside the file you'll see:

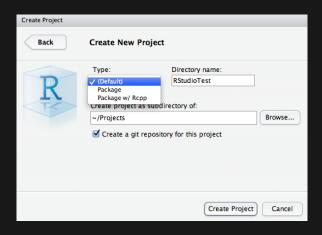
Edit, add, commit, push, submit pull request.

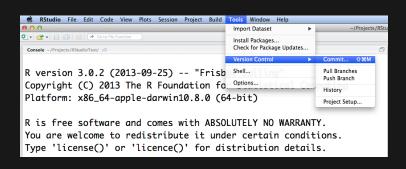
git/GitHub with RStudio

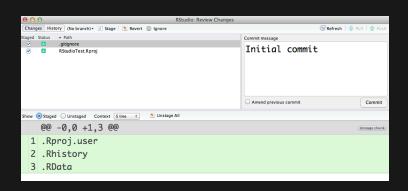


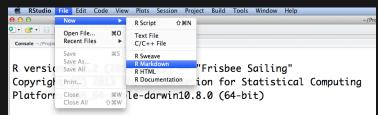
git/GitHub with RStudio



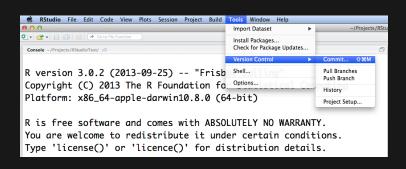


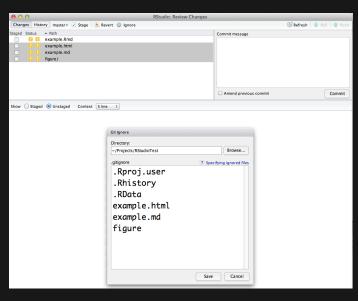


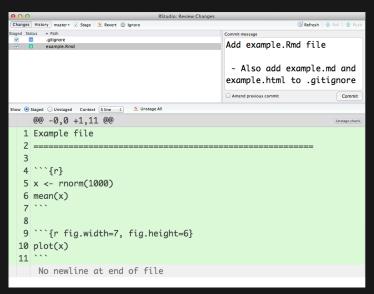


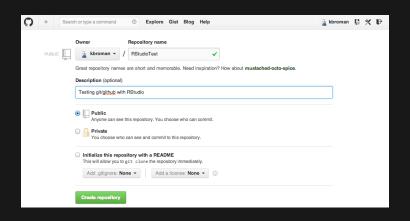


R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under certain conditions. Type 'license()' or 'licence()' for distribution details.

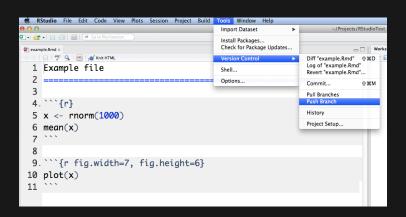


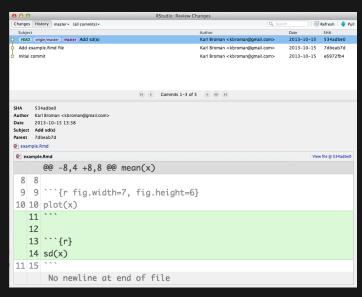




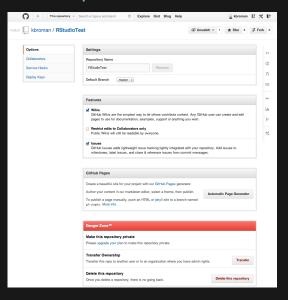


git remote add origin git@github.com:kbroman/repo git push -u origin master





Delete GitHub repo



Open source means everyone can see my stupid mistakes.

Version control means everyone can see every stupid mistake I've ever made.

Resources

- Look at others' repositories:
 - Hadley Wickham (ggplot2): https://github.com/hadley
 - Yihui Xie (knitr): https://github.com/yihui
- ► Karl's tutorial: http://kbroman.github.io/github_tutorial
- ► Karthik Ram's slides: http://karthikram.github.io/git_intro
- ► Pro Git book: http://git-scm.com/book