

"FINAL".doc



FINAL.doc!



FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



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



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



JORGE CHAM © 2012

Git Started

*with GitHub &
RStudio*

Althea ArchMiller
September 9, 2016



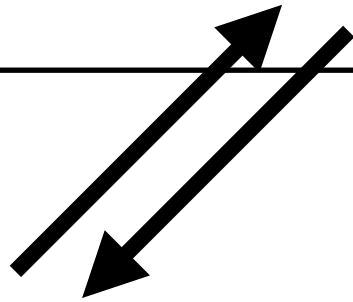
GitHub Website

- Version Control Center
 - store pushed versions from RStudio
- Read/Write Code Directly
- Control Accounts
- Accept/Reject Pull Requests



GitHub Website

- Version Control Center
 - store pushed versions from RStudio
- Read/Write Code Directly
- Control Accounts
- Accept/Reject Pull Requests



RStudio Desktop (Owner)

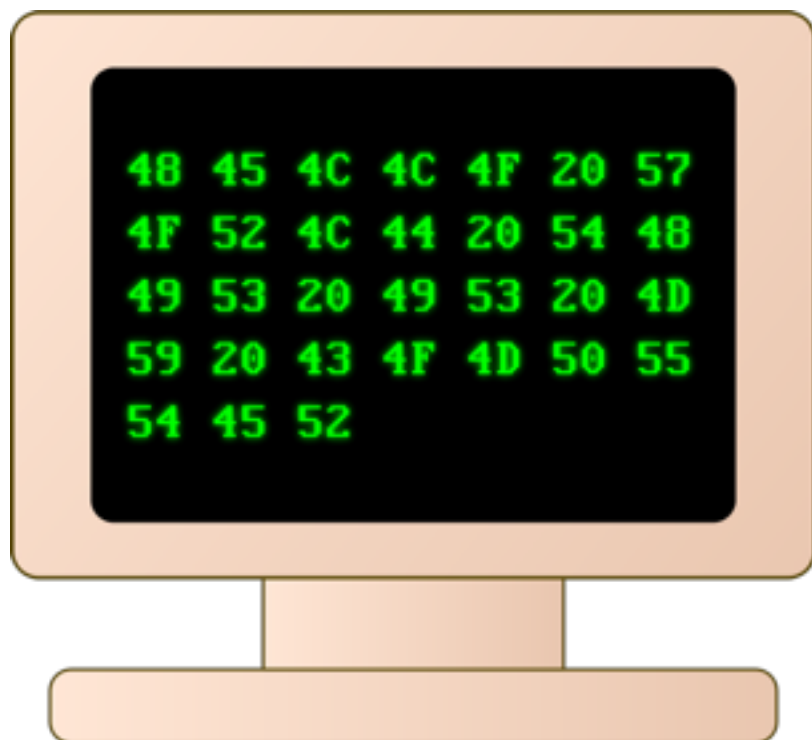


- Read/Write Documents
- Analyses in R
 - Run code
 - Make figures
 - Spin html/pdfs etc
- Commit Changes
- Push/Pull from GitHub to Master Branch Directly

GitHub Website

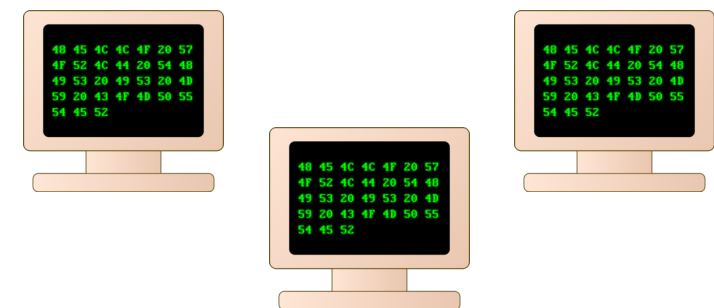
- Version Control Center
 - store pushed versions from RStudio
- Read/Write Code Directly
- Control Accounts
- Accept/Reject Pull Requests

RStudio Desktop (Owner)



- Read/Write Documents
- Analyses in R
 - Run code
 - Make figures
 - Spin html/pdfs etc
- Commit Changes
- Push/Pull from GitHub to Master Branch Directly

Other Accounts



- Check out "branch"
- Edit and Commit changes
- Create Pull Requests

Agenda

- Tour of GitHub website and “Git Started Tutorial”
- Add GitHub Accounts to existing “Git Started” Project
- Connect RStudio & GitHub
- Roll Call exercise
- Unicorn Biology exercise
- 20 Questions exercise



Connect RStudio & GitHub

RStudio

1. New Project > Version Control
2. Clone Git Repository > Url:

[https://github.com/username/
Git_Started_Tutorial](https://github.com/username/Git_Started_Tutorial)

This
should be
easy



Connect RStudio & GitHub

RStudio

3. Tools > Global Options > Git/SVN
 1. Ensure path to git.exe is correct
 2. Create RSA Key
 3. Enter GitHub password, close
 4. View Public Key > copy

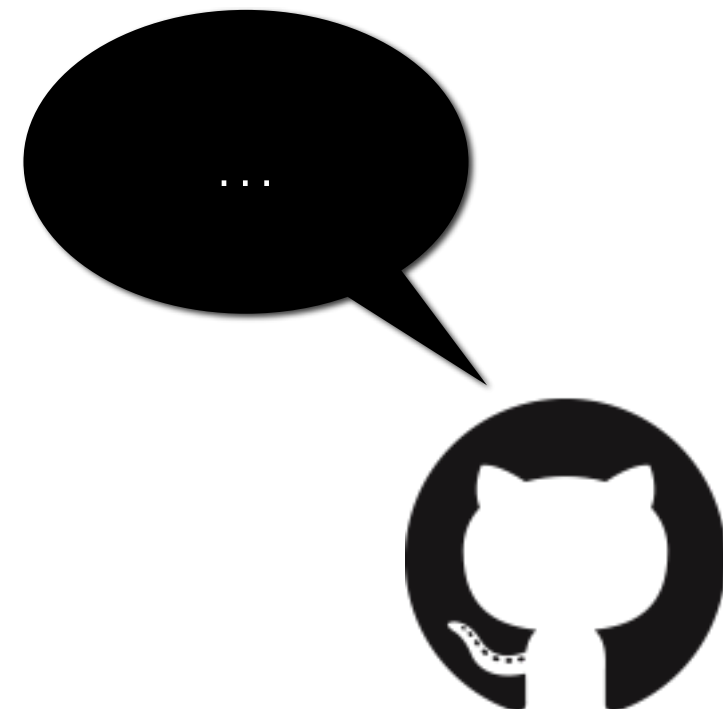
Got it!
What's next?



Connect RStudio & GitHub

GitHub website

1. Account Settings > SSH keys
2. Add SSH Key
3. Name example: “My Laptop”
4. Paste in key from RStudio



Connect RStudio & GitHub

RStudio

1. Tools > Shell...

```
git config --global user.email "email@email.com"
```

```
git config --global user.name "username"
```

```
git config remote.origin.url
```

```
git@github.com:username/Git_Started_Tutorial
```

BTW, all of
this on 1 line



Connect RStudio & GitHub

RStudio

2. Git tab > Pull

3. (One by One) Open “programs/roll_call.R”

1. Add name & handle

2. Git tab > Commit

3. Click checkbox by “programs/roll_call.R”

...We're not
done yet?



Connect RStudio & GitHub

RStudio

4. “Roll Call-name” in commit message, Commit & Close

5. PUSH

6. If you get a warning message, type in “yes” and OK

7. Enter passphrase and OK

Whew - We
made it!



Unicorn Biology

- Now, we'll do some short coding and see how collaborating would work
- The goal is to determine a bit about unicorns using (very fake) data provided in the project
- Ready ... Set ...



Go!

Hint: Look at the Readme file

20 Questions

Let's see who can guess the other team's famous person with the fewest questions (while getting used to pulling, committing, & pushing)

- Get into 2 teams
- Flip a coin to see which team starts ("Alpha team")
- Open "programs/twenty_questions_round1.R"
- Alpha team: Write a question in the box for question 1. Commit and push.
- Beta team: Pull. Answer question. Commit and push.
- Repeat previous two steps until you can guess the famous person

Bob Saget?



Additional Resources

- <http://lifehacker.com/5983680/how-the-heck-do-i-use-github>
- <http://r-bio.github.io/intro-git-rstudio/>
- Other links provided in the email...



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.