Intro to Git and GitHub

Tara Henechowicz

April 25, 2022

Session Outline

1. Download and Install Git help, creating github account help (first 15 minutes)
2. Introductions (15 minutes)
3. Git Slides (see slides from U of T coders) (20 minutes)
4. Working with Git locally: edit a text file and git it
5. GitHub for Collaboration

Step 1. Configure git

```
#This step you only do it once (first time using git on yo
git config --global user.name "tarahenechowicz"
#your github username, check your github profile
git config --global user.email "tara.henechowicz@mail.utoro
#also recommend using your github account email
```

Step 2. Create a Repository on your computer with gitGit

```
#First step is we need to know where we are in our computer
#To figure out where you are (if you're lost!):
pwd

#On mac you might want to change to a place that finder can
  #windows it is called "documents"
  #to change the director use the command cd
cd Documents #name of directory
# to go backwards use cd .. to go back one level, have to d
cd ..
cd ..
#We are going to make a directory (or folder) for our proje
mkdir GitPractice2
# now let's go into the directory that we just made
cd GitPractice2
```

```
## /Users/tarahenechowicz/Documents/RTeaching/IntroGitGitHu
## bash: line 7: cd: Documents: No such file or directory
```

Step 3. Git Init (Initialize the Repo)

```
git init ## Create the repository (init = initialize)

#change the default branch name to mastera
git config --global init.defaultBranch
```

```
## Reinitialized existing Git repository in /Users/tarahene
## master
```

Adding files to the Repo

- Let's create a text file to store some data to work on

1. Create a text file named mydata.txt

```
#use touch command and create a file
touch mydata.txt
```

2. Open the file and add:

- your name
- your program
- your research area

3. Save the file.

Making our first commit

```
git status mydata.txt ## Check the activity
#we need to add it to the stage so it can be commited
git add mydata.txt
#you can also do git add all to add all the files in the r
  #git add -A
```

```
## On branch master
## Changes not staged for commit:
##   (use "git add <file>..." to update what will be commit
##   (use "git restore <file>..." to discard changes in wor
##  modified:   mydata.txt
##
## no changes added to commit (use "git add" and/or "git co
```

you can also use `git add -A` to add all the files in your repository (remember we only have one right now)

Making our first commit

Let's save our first set of changes using a commit. You need to put a message with a commit.

```
#git commit to save file to the history with a message (-m)
git commit -m "Commit 1"
```

```
## [master d827503] Commit 1
##  1 file changed, 2 insertions(+)
```

Making our second commit

Now we are going to back to the text file and make a change. - for example, I'm going to add my affiliation in the first line - you can add your programming languages and experiences - or add a sentence about why you want to use git

Check activity and changes with git status or git diff

```
#we can use git status to see the activity
git status ## Check the activity
git diff # shows the difference between the changes in our
git add -A #to add all of our changes in the text file
git status ## should be green to show that our text file is
```

```
## On branch master
## Changes not staged for commit:
##   (use "git add <file>..." to update what will be commit
##   (use "git restore <file>..." to discard changes in wor
##  modified:   .Rproj.user/665D533C/sources/prop/INDEX
##  modified:   SessionPlan.Rmd
##
## Untracked files:
##   (use "git add <file>..." to include in what will be co
##  SessionPlan.pdf
##
## no changes added to commit (use "git add" and/or "git co
```

Make the commit

```
git commit -m "modified my affiliation"
```

```
## [master 11fd656] modified my affiliation
##  3 files changed, 52 insertions(+), 36 deletions(-)
##  create mode 100644 SessionPlan.pdf
```

Let's check our log of changes

```
git log
```

```
## commit 11fd6568a4766c90d44d6cfd8085fd2912767e03
## Author: tarahenechowicz <tara.henechowicz@mail.utoronto
## Date:   Mon Apr 25 09:36:21 2022 -0400
##
##     modified my affiliation
##
## commit d827503f54eaa0efba7ceb297d0488a26760602e
## Author: tarahenechowicz <tara.henechowicz@mail.utoronto
## Date:   Mon Apr 25 09:36:21 2022 -0400
##
##     Commit 1
##
## commit 0be27699d0be6974d6173dbc046729917b37d2c3
## Author: tarahenechowicz <tara.henechowicz@mail.utoronto
## Date:   Mon Apr 25 09:21:06 2022 -0400
##
```

PART II GitHub

1.Create a repository on github that matches the name of your repository on your computer

- take our repository from our computer and "push it" to github so it can be store on their server in addition to or instead of our computer
- Why would we want to do this?

1. storage (privately or publicly)
2. Public sharing of data (open science)
3. now that its on github your team members or even general public can collaborate with you on your work and you can log the changes of multiple users!

To push our repository to github, we need to first set up the connection between the repository on our computer with the repository on github

```
#go to the repository that we just made and there is the g
  #git remote add origin https://github.com/tarahenechowicz
```

Part 4 How to use git with Rstudio

Social aspect of Github

starring following