Class 3: Research Reproducibility II

Kim Johnson

February 1, 2018

Lecture Outline

- Version control
- Git using GitBash
- Github

Learning Objectives

1. Be able to understand what version control is and how it can potentially make your life easier
2. Be able to create a Github account
3. Understand and utilize some basic git version control *operations* and **words** (*add*, *commit*, *status*, *log*, **origin**, **master**, *push*, *pull*,*fork*)

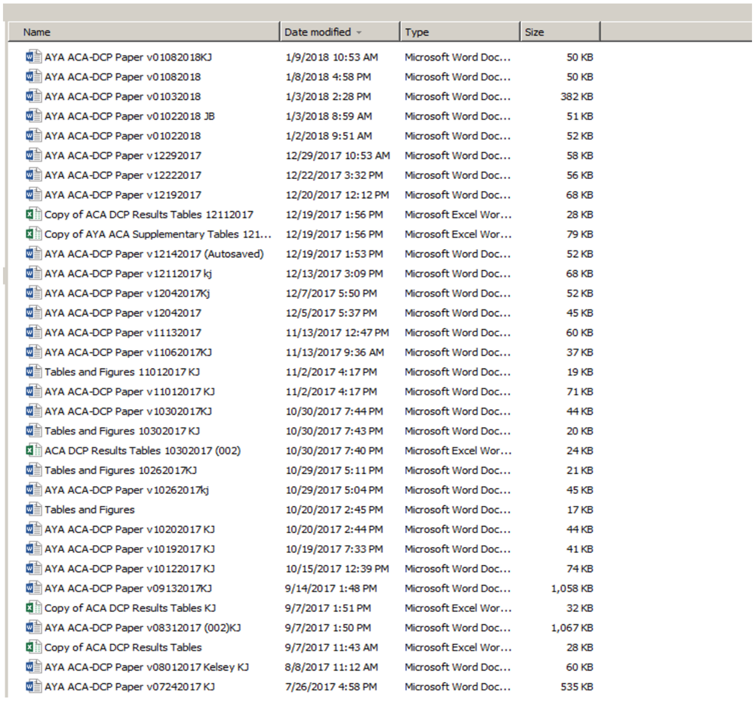
Version control

- How many of you have worked on assignments where multiple versions of a project file were simultaneously being altered by different people?
- Exchanging files by email?
- Working on the wrong version and then having to add your changes to a different version?

So what is version control?

(with credit to https://www.visualstudio.com/learn/what-is-version-control/)

- Software for tracking changes you make to code or any files for that matter
- In the old days and perhaps now, version control involved (involves) multiple versions of the same document on your computer with version numbers or dates.



Name	Date modified	Type	Size
AYA ACA-DCP Paper v01082018KJ	1/9/2018 10:53 AM	Microsoft Word Doc...	50 KB
AYA ACA-DCP Paper v01082018	1/8/2018 4:58 PM	Microsoft Word Doc...	50 KB
AYA ACA-DCP Paper v01032018	1/3/2018 2:28 PM	Microsoft Word Doc...	382 KB
AYA ACA-DCP Paper v01022018 JB	1/3/2018 8:59 AM	Microsoft Word Doc...	51 KB
AYA ACA-DCP Paper v01022018	1/2/2018 9:51 AM	Microsoft Word Doc...	52 KB
AYA ACA-DCP Paper v12292017	12/29/2017 10:53 AM	Microsoft Word Doc...	58 KB
AYA ACA-DCP Paper v12222017	12/22/2017 3:32 PM	Microsoft Word Doc...	56 KB
AYA ACA-DCP Paper v12192017	12/20/2017 12:12 PM	Microsoft Word Doc...	68 KB
Copy of ACA DCP Results Tables 12112017	12/19/2017 1:56 PM	Microsoft Excel Wor...	28 KB
Copy of AYA ACA Supplementary Tables 121...	12/19/2017 1:56 PM	Microsoft Excel Wor...	79 KB
AYA ACA-DCP Paper v12142017 (Autosaved)	12/19/2017 1:53 PM	Microsoft Word Doc...	52 KB
AYA ACA-DCP Paper v12112017 kj	12/13/2017 3:09 PM	Microsoft Word Doc...	68 KB
AYA ACA-DCP Paper v12042017Kj	12/7/2017 5:50 PM	Microsoft Word Doc...	52 KB
AYA ACA-DCP Paper v12042017	12/5/2017 5:37 PM	Microsoft Word Doc...	45 KB
AYA ACA-DCP Paper v11132017	11/13/2017 12:47 PM	Microsoft Word Doc...	60 KB
AYA ACA-DCP Paper v11062017KJ	11/13/2017 9:36 AM	Microsoft Word Doc...	37 KB
Tables and Figures 11012017 KJ	11/2/2017 4:17 PM	Microsoft Word Doc...	19 KB
AYA ACA-DCP Paper v11012017 KJ	11/2/2017 4:17 PM	Microsoft Word Doc...	71 KB
AYA ACA-DCP Paper v10302017KJ	10/30/2017 7:44 PM	Microsoft Word Doc...	44 KB
Tables and Figures 10302017 KJ	10/30/2017 7:43 PM	Microsoft Word Doc...	20 KB
ACA DCP Results Tables 10302017 (002)	10/30/2017 7:40 PM	Microsoft Excel Wor...	24 KB
Tables and Figures 10262017KJ	10/29/2017 5:11 PM	Microsoft Word Doc...	21 KB
AYA ACA-DCP Paper v10262017kj	10/29/2017 5:04 PM	Microsoft Word Doc...	45 KB
Tables and Figures	10/20/2017 2:45 PM	Microsoft Word Doc...	17 KB
AYA ACA-DCP Paper v10202017 KJ	10/20/2017 2:44 PM	Microsoft Word Doc...	44 KB
AYA ACA-DCP Paper v10192017 KJ	10/19/2017 7:33 PM	Microsoft Word Doc...	41 KB
AYA ACA-DCP Paper v10122017 KJ	10/15/2017 12:39 PM	Microsoft Word Doc...	74 KB
AYA ACA-DCP Paper v09132017KJ	9/14/2017 1:48 PM	Microsoft Word Doc...	1,058 KB
Copy of ACA DCP Results Tables KJ	9/7/2017 1:51 PM	Microsoft Excel Wor...	32 KB
AYA ACA-DCP Paper v08312017 (002)KJ	9/7/2017 1:50 PM	Microsoft Word Doc...	1,067 KB
Copy of ACA DCP Results Tables	9/7/2017 11:43 AM	Microsoft Excel Wor...	28 KB
AYA ACA-DCP Paper v08012017 Kelsey KJ	8/8/2017 11:12 AM	Microsoft Word Doc...	60 KB
AYA ACA-DCP Paper v07242017 KJ	7/26/2017 4:58 PM	Microsoft Word Doc...	535 KB

Version control system benefits

(with credit to https://www.visualstudio.com/learn/what-is-version-control/)

- Manage all versions but you only see one at a time
- Every version can have a log of changes made (either specific or general)

- You can go back to any version at any time (you never lose the work)
- Facilitates collaborative coding
- You have an archive of your changes to a particular file
- Important for reproducible research!

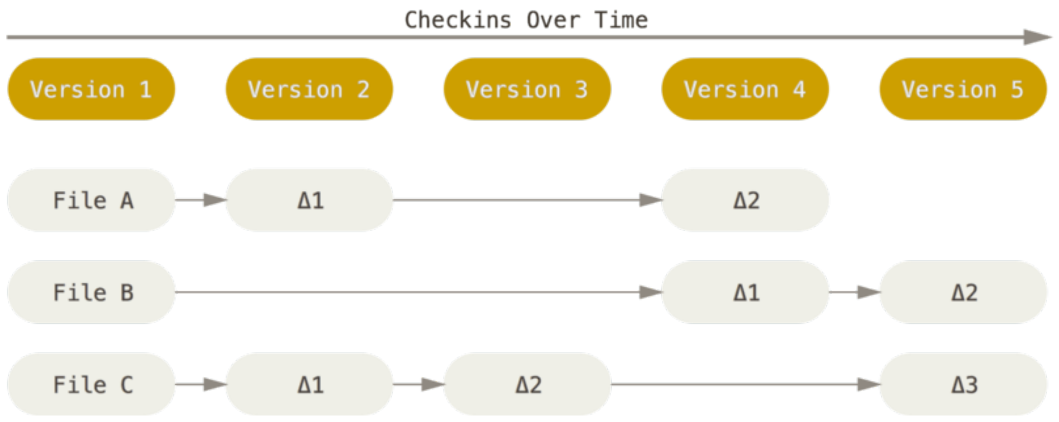
Version control systems

- CVS
- SVN (subversion)
- **GIT** (emerging star and harder to learn but we can do it!) **developed for plain text files but there are rich text file adaptations available
- Mecurial
- Bazaar

To read more: https://www.getfilecloud.com/blog/2015/02/top-5-open-source-version-control-tools-for-system-admins/#.WlZnw66nHAU

Git

- Developed in 2005 by Linus Torvalds (claim to fame: Linux open source operating system creation)
- Software that takes snapshots of files in a directory at points in time and keeps a retrievable record of plain text changes



Two main types of Git

- Command line interface (CLI, GitBash)

- Graphical user interface (GUI, Github desktop)

What is the command line?

- User issues "commands" to a program through lines of text in what is known as a shell (e.g. Unix, *Bash* is a Unix shell)
- Has been around since the 1960s
- According to Wikipedia: "Command-line interfaces to computer operating systems are less widely used by casual computer uses, who favor graphical user interfaces or menu-driven interaction"
- Other advantages (according to Wikipedia):
    - *uses fewer system resources*
    - *faster than GUIs with experience*
    - *can keep a command line history (in contrast to GUIs, which are interactive)*

Command prompts and arguments

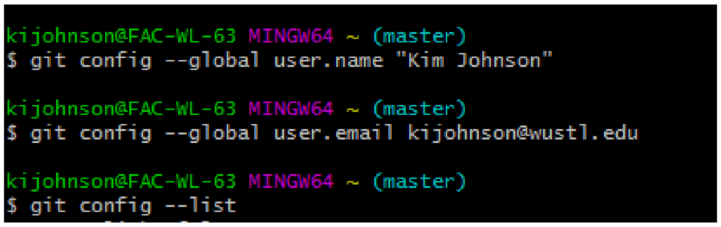
- Character that indicates the shell's readiness to take commands (for Unix it is $)
- Arguments follow the command prompt and tell the computer what to do

Git-Bash is the command line shell for Git using Unix

- Allows for local and remote version control (version control of files on your computer and Github or other remote server)
- Puts a directory on your computer under version control and send version controlled files to your Github remote repository
- Key steps to version controlling files:
    - *adding a file for tracking using git add*
    - *committing a tracked file with a message using git commit -m 'my message' (i.e. taking a snapshot)*
    - *pushing your version controlled files to a remote repository using git push remote origin master (note there are other ways to specify pushes)*

Steps for local version control

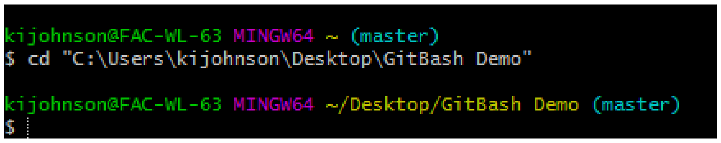
1. Open GitBash and set up user information using the *git config* command and then the *git config –list* command to check configuration settings



```
kijohnson@FAC-WL-63 MINGW64 ~ (master)
$ git config --global user.name "Kim Johnson"

kijohnson@FAC-WL-63 MINGW64 ~ (master)
$ git config --global user.email kijohnson@wustl.edu

kijohnson@FAC-WL-63 MINGW64 ~ (master)
$ git config --list
```

+

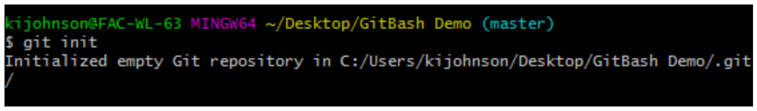
2. Change directory (*cd*) to a folder you want to put under version control (i.e. initialize a git repository)



```
kijohnson@FAC-WL-63 MINGW64 ~ (master)
$ cd "C:\Users\kijohnson\Desktop\GitBash Demo"

kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$
```

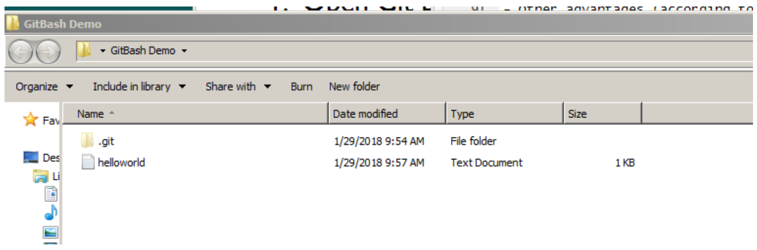
+

3. Initialize a git repository to that directory with the command *git init*



```
kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$ git init
Initialized empty Git repository in C:/Users/kijohnson/Desktop/GitBash Demo/.git
/
```

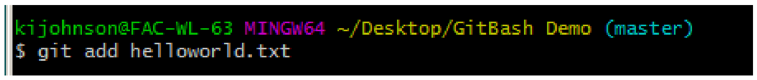
4. Create a text file using a text editor (e.g. notepad) titled *helloworld.txt* and save it to your version controlled directory



```
Name          Date modified       Type            Size
.git          1/29/2018 9:54 AM   File folder
helloworld    1/29/2018 9:57 AM   Text Document   1 KB
```

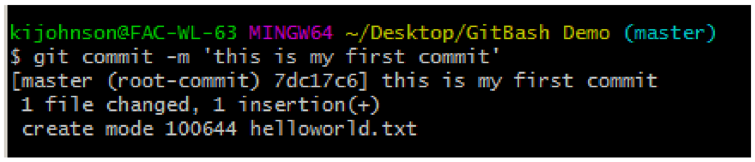
Steps for local version control (cont.)

5. Put that file under version control using the *git add filename* command (this file is said to now be **staged** for version control)



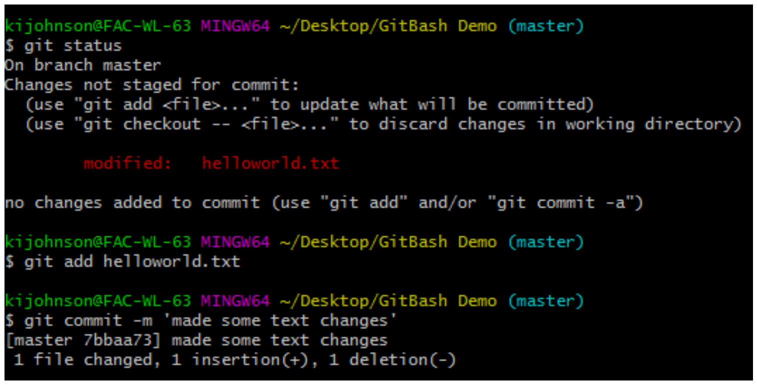
```
kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$ git add helloworld.txt
```

6. *Commit* that version with a message describing changes using the *git command -m 'my message'* command (this stores that version of the file safely in your database– i.e. takes a snapshot)



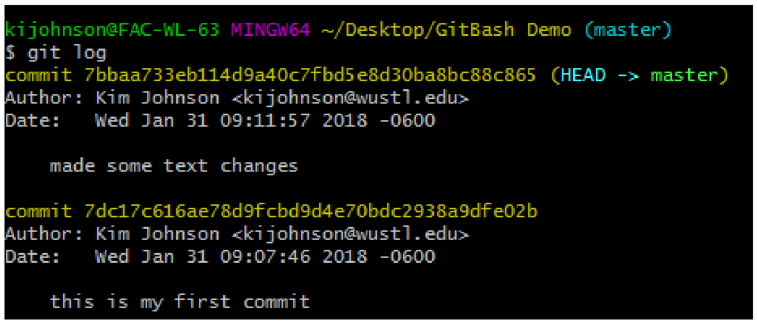
```
kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$ git commit -m 'this is my first commit'
[master (root-commit) 7dc17c6] this is my first commit
 1 file changed, 1 insertion(+)
 create mode 100644 helloworld.txt
```

7. Make a modification to the *helloworld.txt* file, check status using the *git status* then *git add* and *git commit* commands again



```
kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   helloworld.txt

no changes added to commit (use "git add" and/or "git commit -a")

kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$ git add helloworld.txt

kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$ git commit -m 'made some text changes'
[master 7bbaa73] made some text changes
 1 file changed, 1 insertion(+), 1 deletion(-)
```

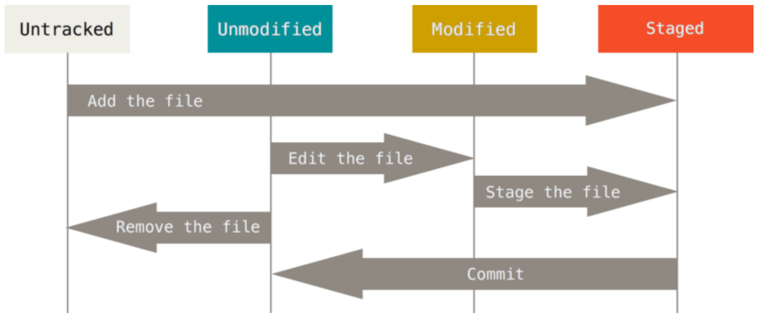
8. Check changes between the two versions using the *git log* command



```
kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$ git log
commit 7bbaa733eb114d9a40c7fbd5e8d30ba8bc88c865 (HEAD -> master)
Author: Kim Johnson <kijohnson@wustl.edu>
Date:   Wed Jan 31 09:11:57 2018 -0600

    made some text changes

commit 7dc17c616ae78d9fcbd9d4e70bdc2938a9dfe02b
Author: Kim Johnson <kijohnson@wustl.edu>
Date:   Wed Jan 31 09:07:46 2018 -0600

    this is my first commit
```

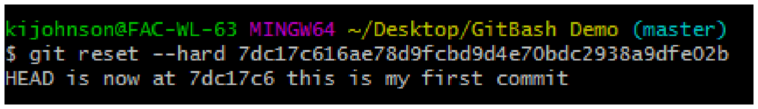
Other useful git commands/further explanation

- *git status*: allows you to check what state your files are in



- *git log - -pretty=format:"%h %s" - -graph*: To get a graph of commits with abbreviated hash numbers for each commit (for output options, see: https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History)

- *git reset - -hard commit hash*: To reset a file version to an earlier commit. Note for commit hash, you need to insert the long or abbreviated alphanumeric number for the commit



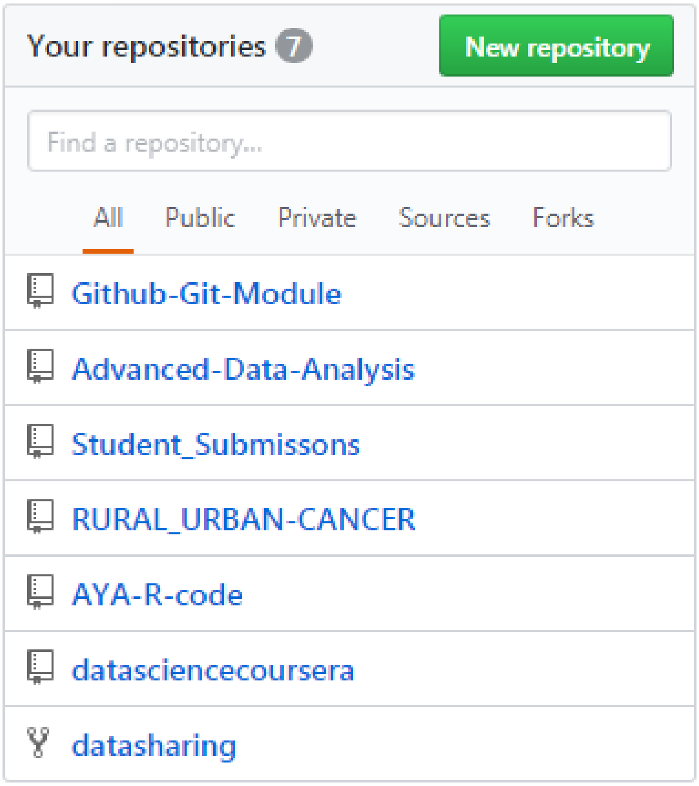
```
kijohnson@FAC-WL-63 MINGW64 ~/Desktop/GitBash Demo (master)
$ git reset --hard 7dc17c616ae78d9fcbd9d4e70bdc2938a9dfe02b
HEAD is now at 7dc17c6 this is my first commit
```

Steps for using Github for remote version control

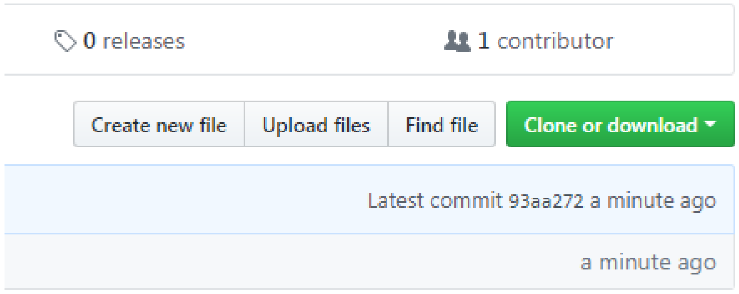
Github is a cloud repository that you can use for remote version control

1. Go to Github and create an account: https://github.com/

2. Create a repository by clicking on the green new repository button

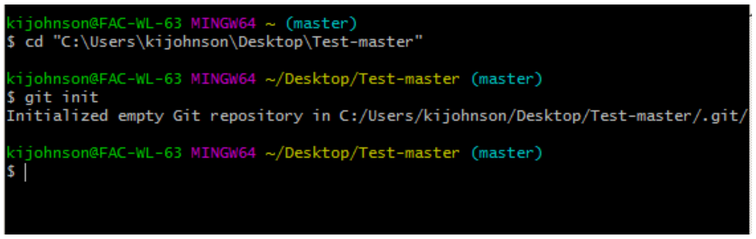


3. Check 'initialize this repository with a readme file' and *clone* that repository to your desktop by clicking on clone or download and then Download Zip. Copy the repository in the zipped file to your desktop.



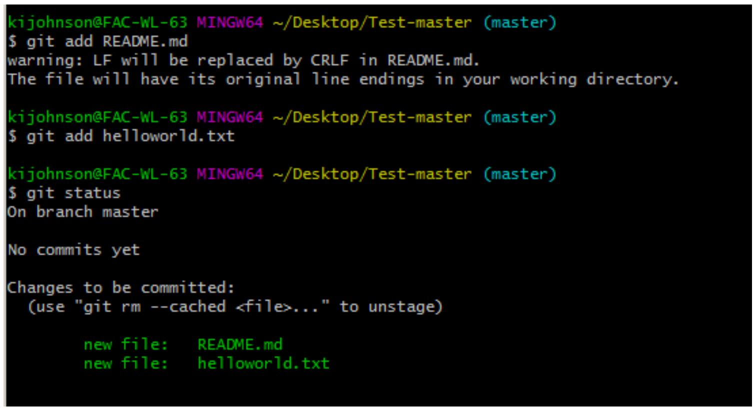
Steps for using Github for remote version control (continued)

4. Go to GitBash and put the cloned repository under local version control using *cd* to make the cloned repository the local master and use *git init* to initialize a local git repository for version control in the cloned repository



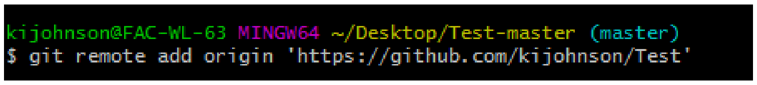
```
kijohnson@FAC-WL-63 MINGW64 ~ (master)
$ cd "C:\Users\kijohnson\Desktop\Test-master"

kijohnson@FAC-WL-63 MINGW64 ~/Desktop/Test-master (master)
$ git init
Initialized empty Git repository in C:/Users/kijohnson/Desktop/Test-master/.git/

kijohnson@FAC-WL-63 MINGW64 ~/Desktop/Test-master (master)
$ |
```

5. Make a copy of your "helloworld.txt" file and *add* it for version control to your cloned repository along with the README.MD file, check the status



```
kijohnson@FAC-WL-63 MINGW64 ~/Desktop/Test-master (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.

kijohnson@FAC-WL-63 MINGW64 ~/Desktop/Test-master (master)
$ git add helloworld.txt

kijohnson@FAC-WL-63 MINGW64 ~/Desktop/Test-master (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md
        new file:   helloworld.txt
```

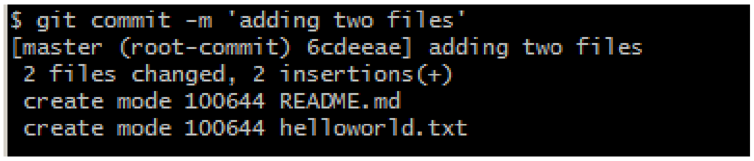
6. Link your local git repository to your github repository with the *git add remote origin 'github repository link'* command



```
kijohnson@FAC-WL-63 MINGW64 ~/Desktop/Test-master (master)
$ git remote add origin 'https://github.com/kijohnson/Test'
```

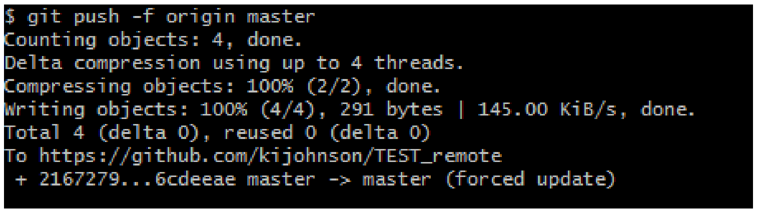
Steps for using Github for remote version control (continued)

7. Commit your local files (helloworld.txt and README.MD) with a message and add them to your Github repository



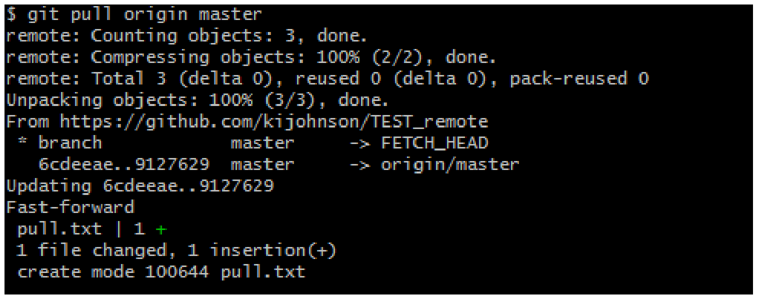
```
$ git commit -m 'adding two files'
[master (root-commit) 6cdeeae] adding two files
 2 files changed, 2 insertions(+)
 create mode 100644 README.md
 create mode 100644 helloworld.txt
```

8. *Push* the local files to your remote repository using the *git push -f origin master* command



```
$ git push -f origin master
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 291 bytes | 145.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/kijohnson/TEST_remote
 + 2167279...6cdeeae master -> master (forced update)
```

9. Create a new text file called anythingyouwant.txt (calling mine pull.txt) and add the new file to your remote repository by dropping it in and use git bash and the *git pull* command to pull that file to your local repository from your remote repository.



```
$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/kijohnson/TEST_remote
 * branch            master     -> FETCH_HEAD
   6cdeeae..9127629  master     -> origin/master
Updating 6cdeeae..9127629
Fast-forward
 pull.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 pull.txt
```

Forking (last topic)

- Forking is a method where you can fork other Github user repositories to your Github repository and work on it independently
- Fork the class repository to your repository
    - *while logged into your Github account go to the class Github website:* *https://github.com/kijohnson/Advanced-Data-Analysis*
    - *click on fork in the upper right hand corner*

For help with Git and Unix commands

- Ask Me! (This is *so cutting edge* for public health and social work that Statlab probably won't be able to help except Alina :))
- Everything you need to know (and way more): https://git-scm.com/book/en/v2
- Unix http://mally.stanford.edu/~sr/computing/basic-unix.html