Git and GitHub

Roadmap

1. Git vs. GitHub
2. Vocabulary
3. GitHub process

Git vs. GitHub

Git

- A piece of software that is installed locally on your computer
- Version control system; allows tracking changes in your code

Git vs. GitHub

GitHub

- Website that hosts/stores git repositories
- Facilitates sharing and collaborating on code
- Allows you to set rules about who can access a project, track bugs in your code, and more

Vocabulary

Repository

Clone

Commit

Push

Pull

Adapted from this post

Vocabulary

Repository (also called repo)

This is a group of files and/or folders that you’re tracking with git. It is the location where your code is saved.

Vocabulary

Clone

Cloning a repo copies a remote repository (i.e. a repo on GitHub) onto your computer.

Vocabulary

Commit

Committing is like saving your work. When you commit, you're basically putting your code in its current state into a time capsule. The commit is exists only on your own computer until you push to GitHub.

Vocabulary

Push

Committing is like placing your files in their current state into a time capsule. When you push, that time capsule is sent to GitHub.

Vocabulary

Pull

Pulling takes code from a remote repository (i.e. a repo on GitHub) and puts it onto your computer. If you are pulling from a repo, you need to have already cloned that repo.

Summary: Vocabulary

Repository

Clone

Commit

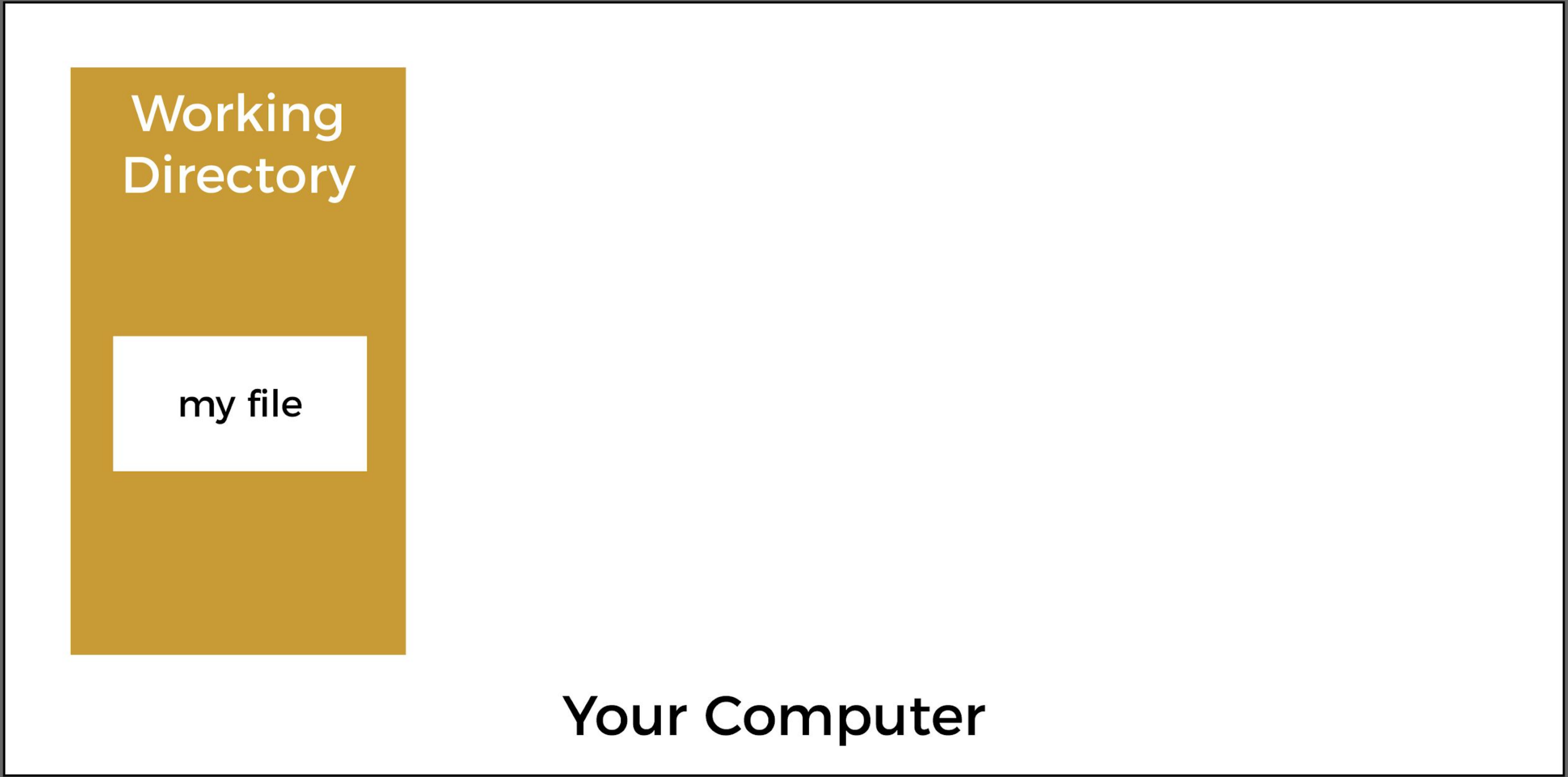
Push

Pull

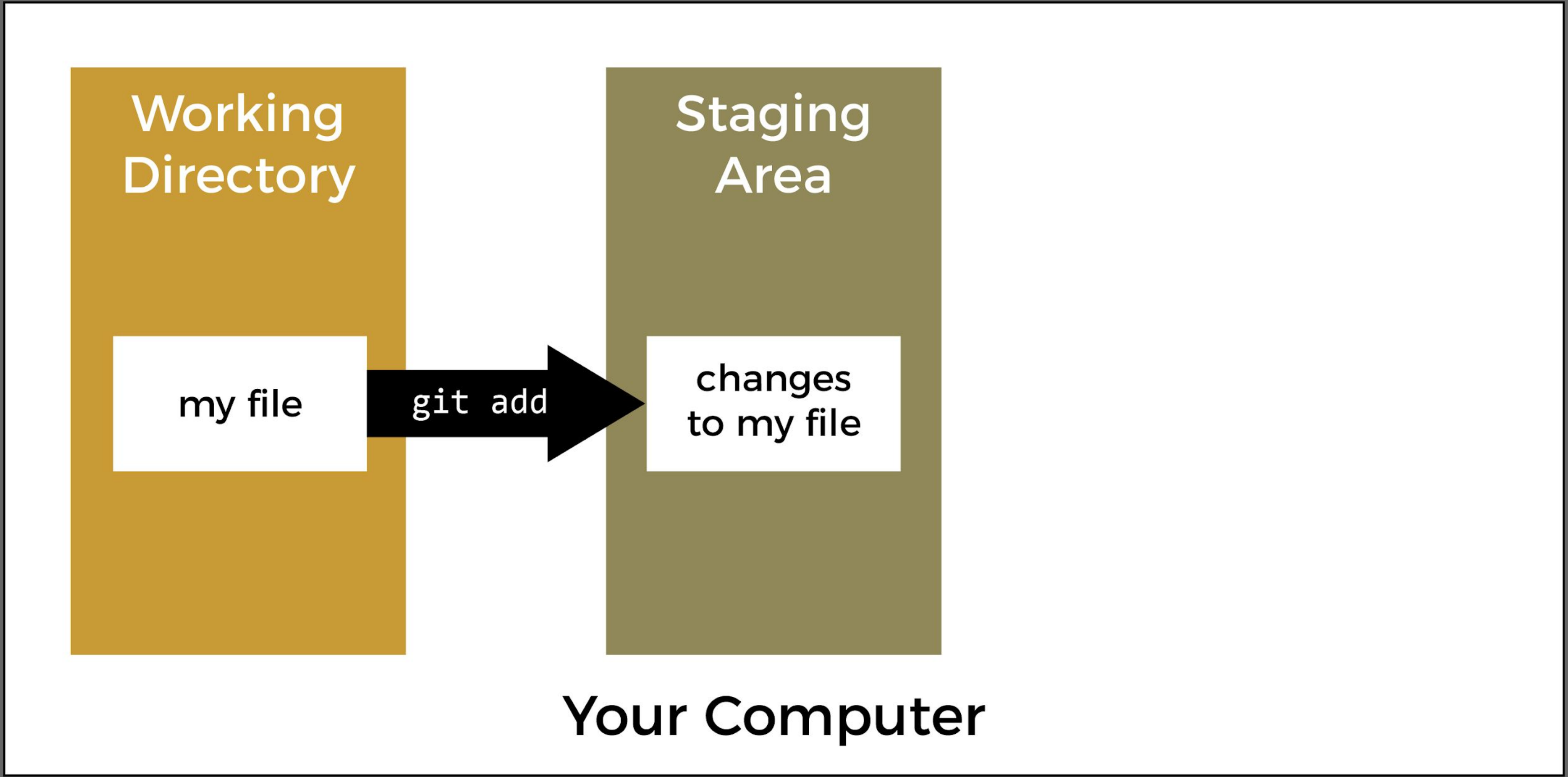
Adapted from this post

GitHub Process

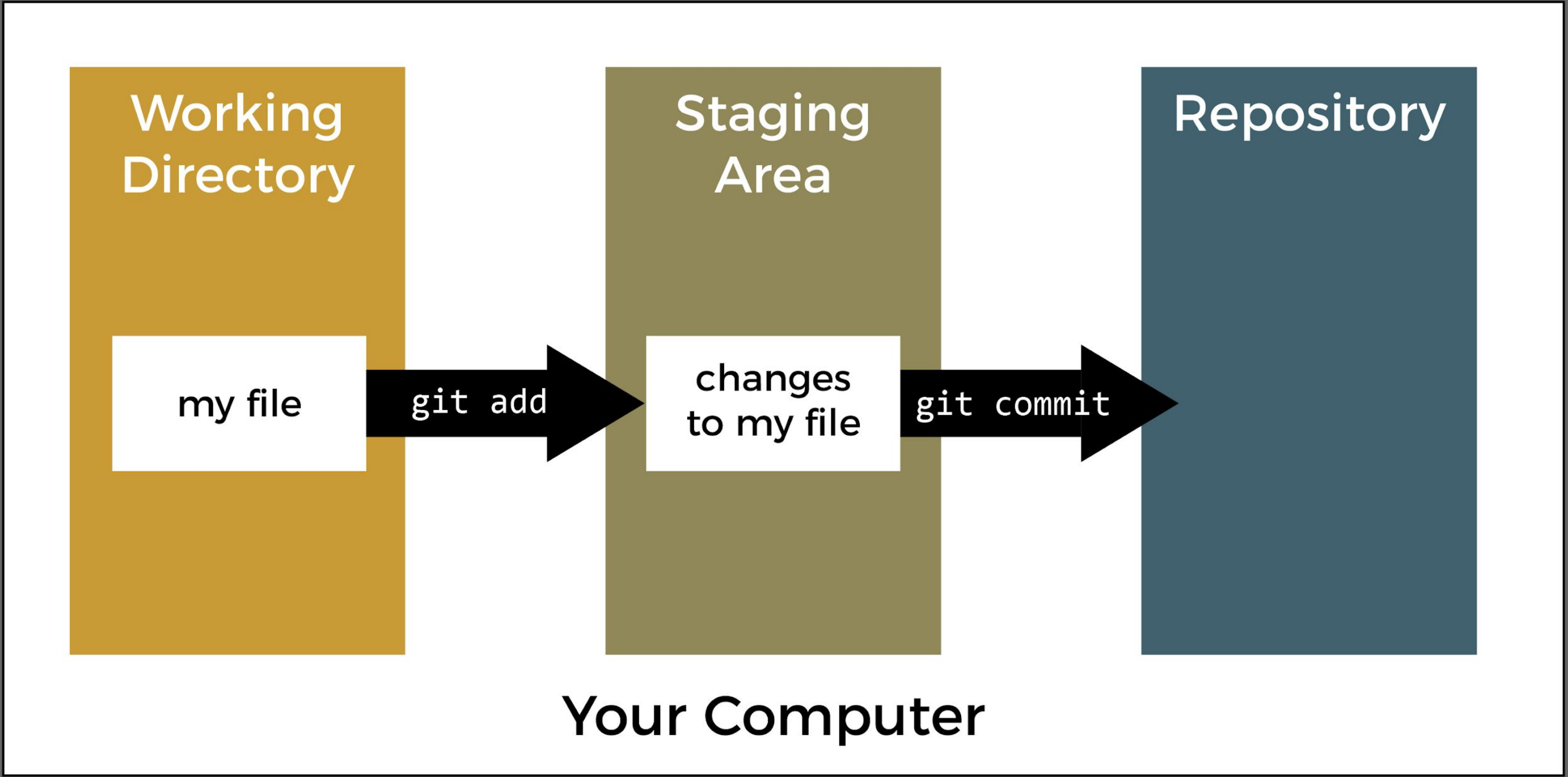
add → commit → push



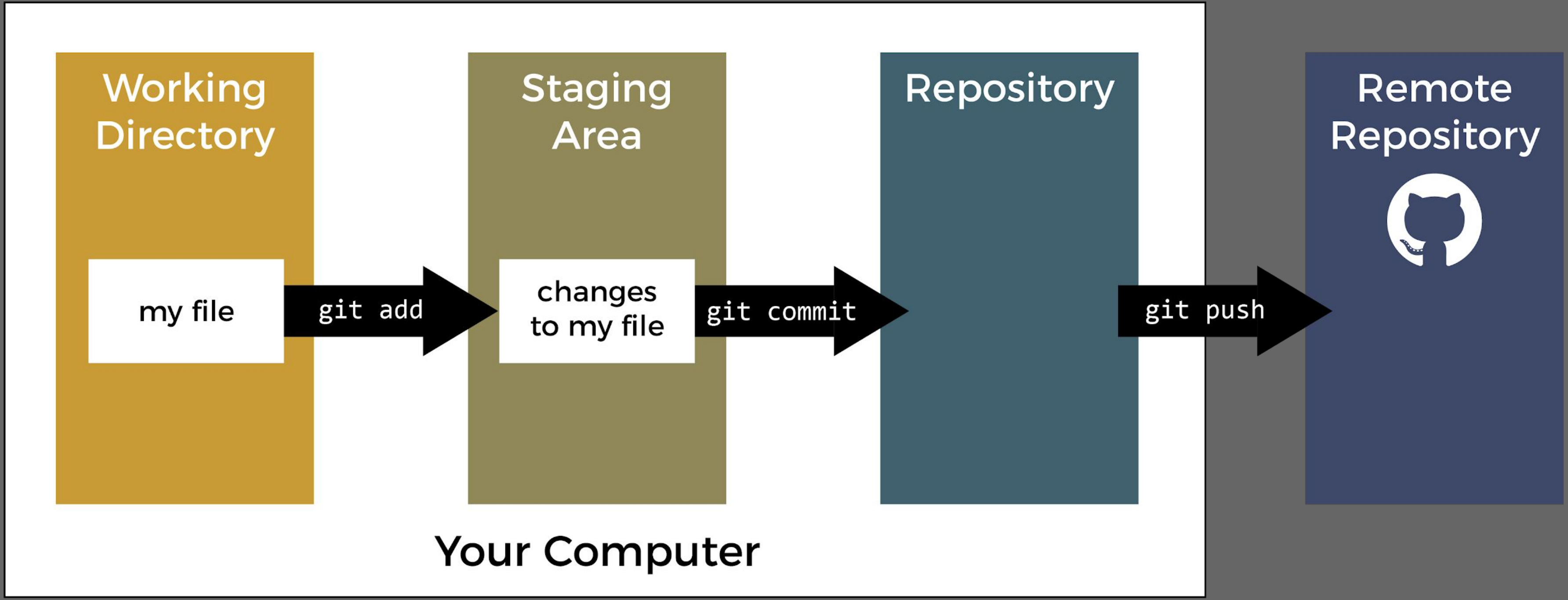
Adapted from: https://www.slideshare.net/LorenzoBaracchi/git-presentation-44341864



Adapted from: https://www.slideshare.net/LorenzoBaracchi/git-presentation-44341864



Adapted from: https://www.slideshare.net/LorenzoBaracchi/git-presentation-44341864

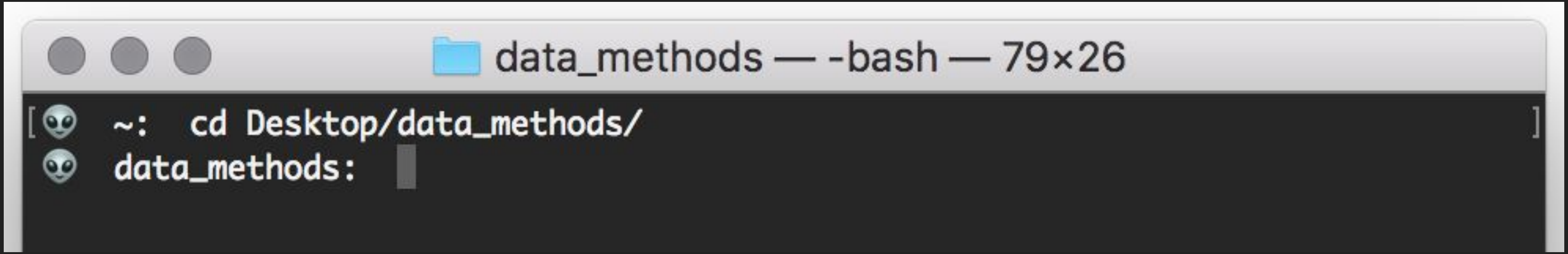


Adapted from: https://www.slideshare.net/LorenzoBaracchi/git-presentation-44341864

GitHub Process

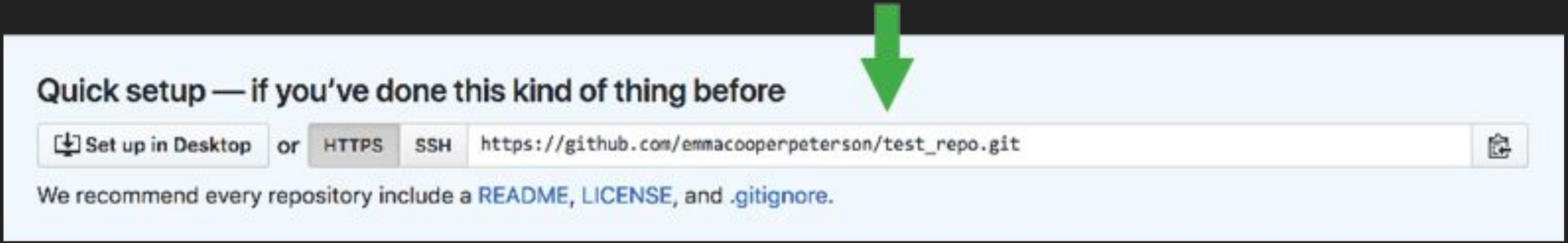
1. Clone a repo from GitHub
2. Push a file to the repo
3. Work on the same file from the RStudio server
4. Pull your work on the RStudio server onto your computer

In your terminal, use `cd` to navigate to the directory in which you want your repo to live. When you `git clone` later, it will create a new folder in this directory.



```
[👽  ~:  cd Desktop/data_methods/                                          ]
 👽  data_methods:
```

Accept the assignment on GitHub. Copy this link at the top of the page after accepting.

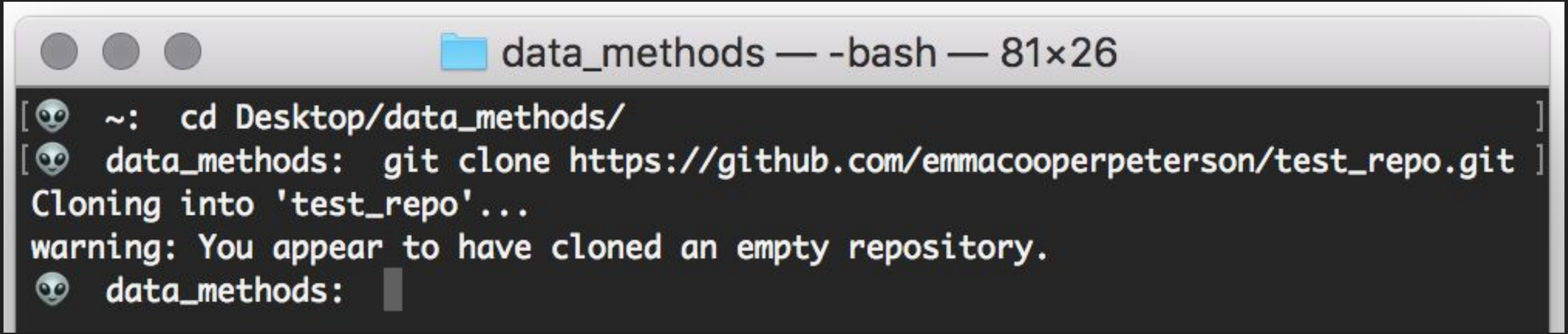


Quick setup — if you've done this kind of thing before

```
https://github.com/emmacooperpeterson/test_repo.git
```

We recommend every repository include a README, LICENSE, and .gitignore.

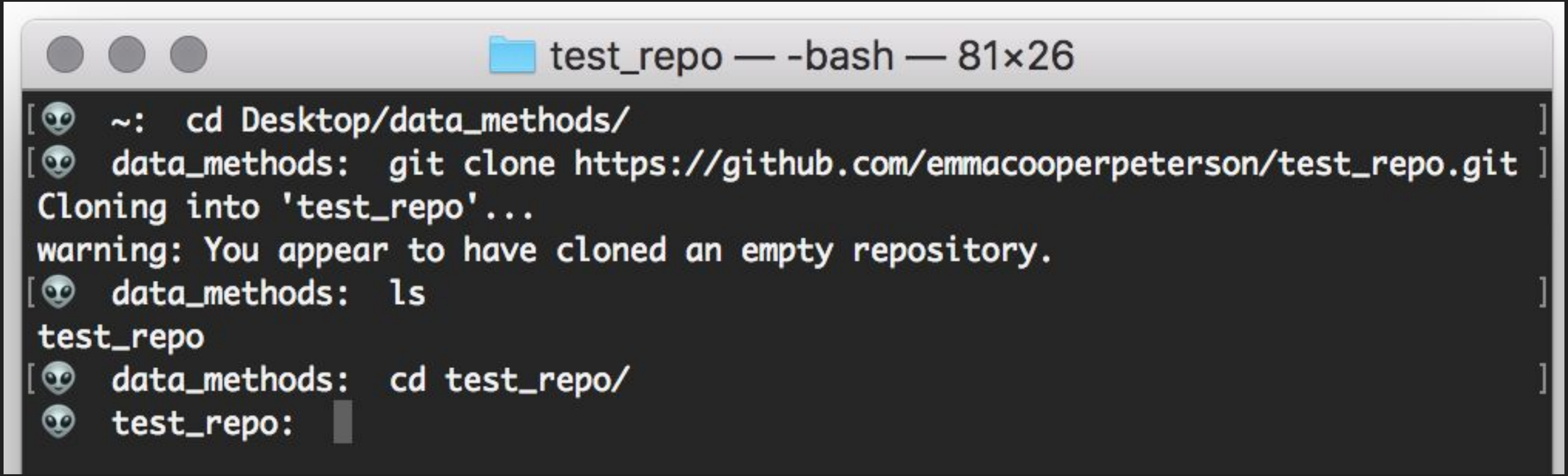
Return to your terminal. Type `git clone` and then paste the link you just copied.



```
[👽  ~:  cd Desktop/data_methods/                                                  ]
[👽  data_methods:  git clone https://github.com/emmacooperpeterson/test_repo.git ]
Cloning into 'test_repo'...
warning: You appear to have cloned an empty repository.
 👽  data_methods:
```

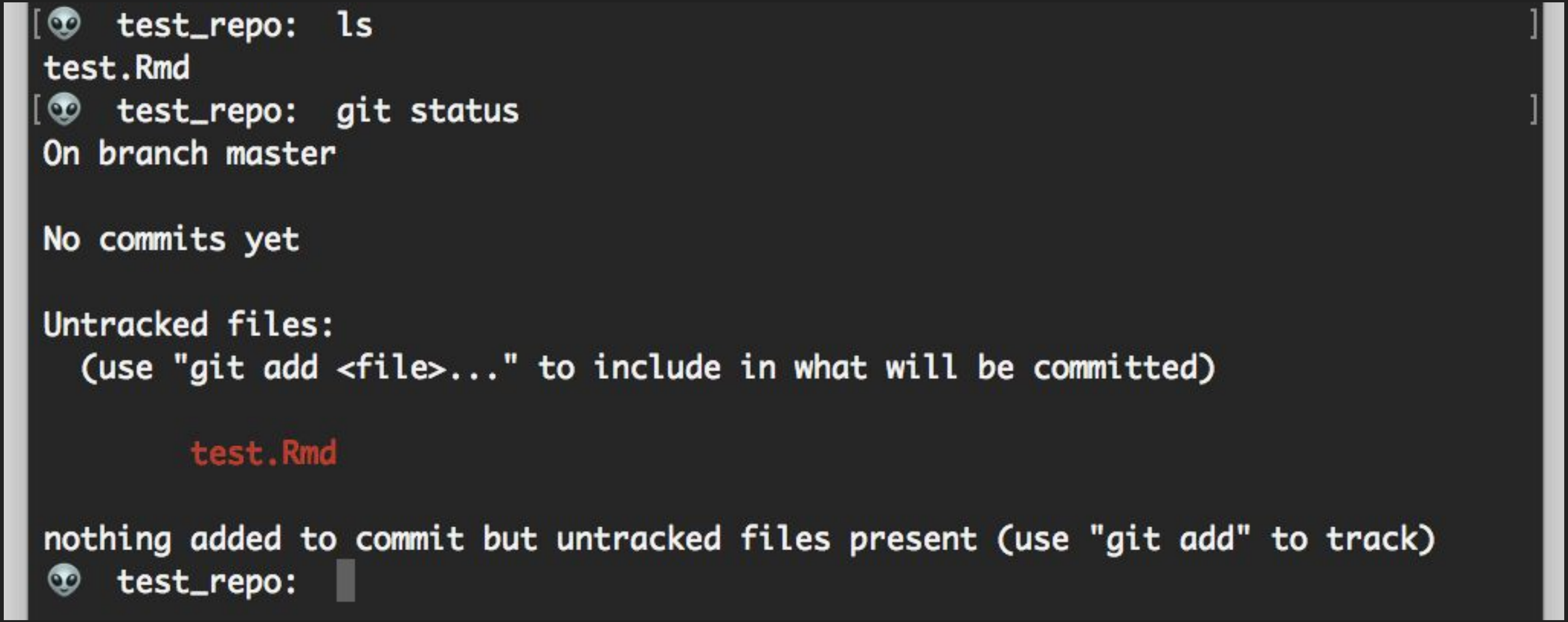
Note: `git clone` goes through several processes, one of which is `git init`. Therefore, once you've cloned, you do not need to use `git init`.

Type `ls` to see that a new folder was created. Navigate into that folder with `cd`. This is where you should save all of your homework-related files.



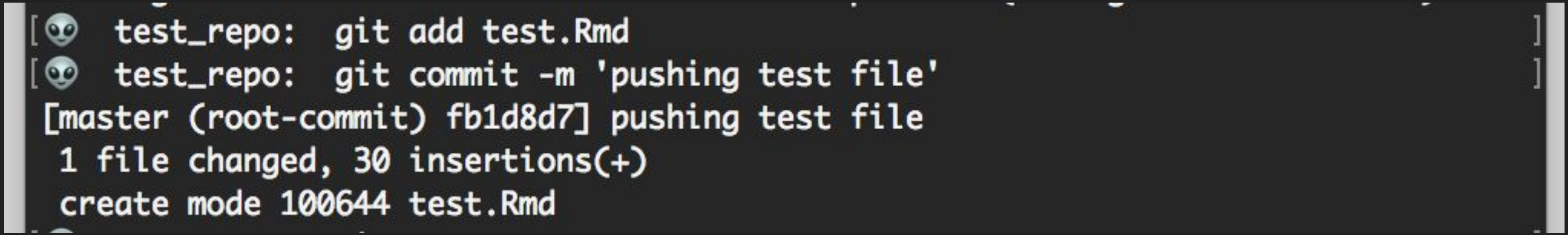
```
👽  ~:  cd Desktop/data_methods/
👽  data_methods:  git clone https://github.com/emmacooperpeterson/test_repo.git
Cloning into 'test_repo'...
warning: You appear to have cloned an empty repository.
👽  data_methods:  ls
test_repo
👽  data_methods:  cd test_repo/
👽  test_repo:
```

I’ve saved a test.Rmd file in my test_repo folder. `ls` shows that the file is in my test_repo folder, and `git status` tells me that the file is ‘untracked’ and that I need to use git add in order to track it.



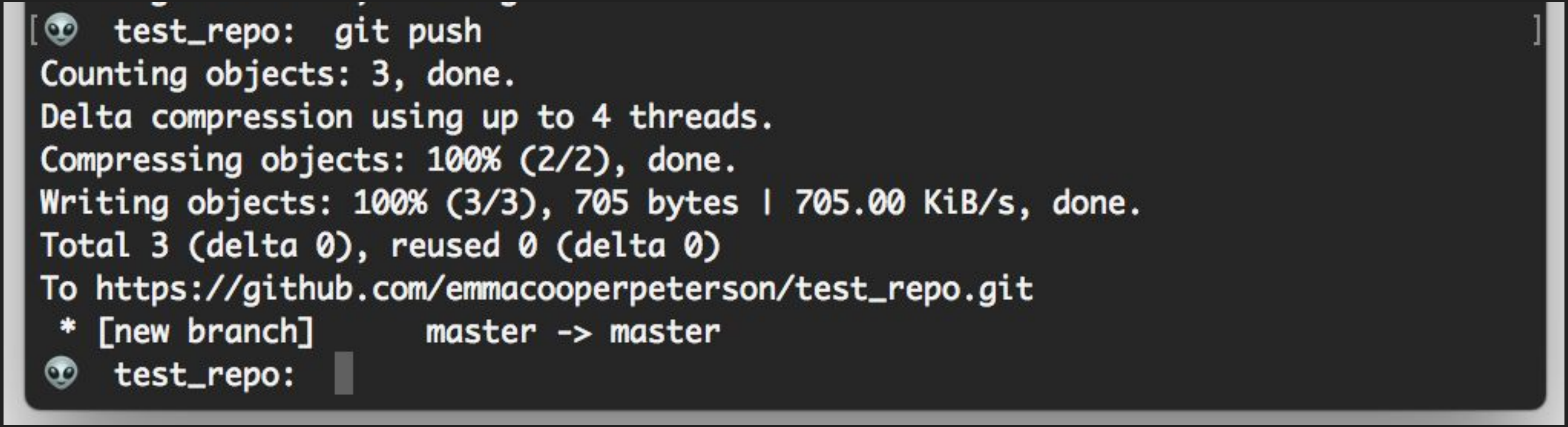
```
👽  test_repo:  ls
test.Rmd
👽  test_repo:  git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.Rmd

nothing added to commit but untracked files present (use "git add" to track)
👽  test_repo:
```

Next, I add and commit the file.



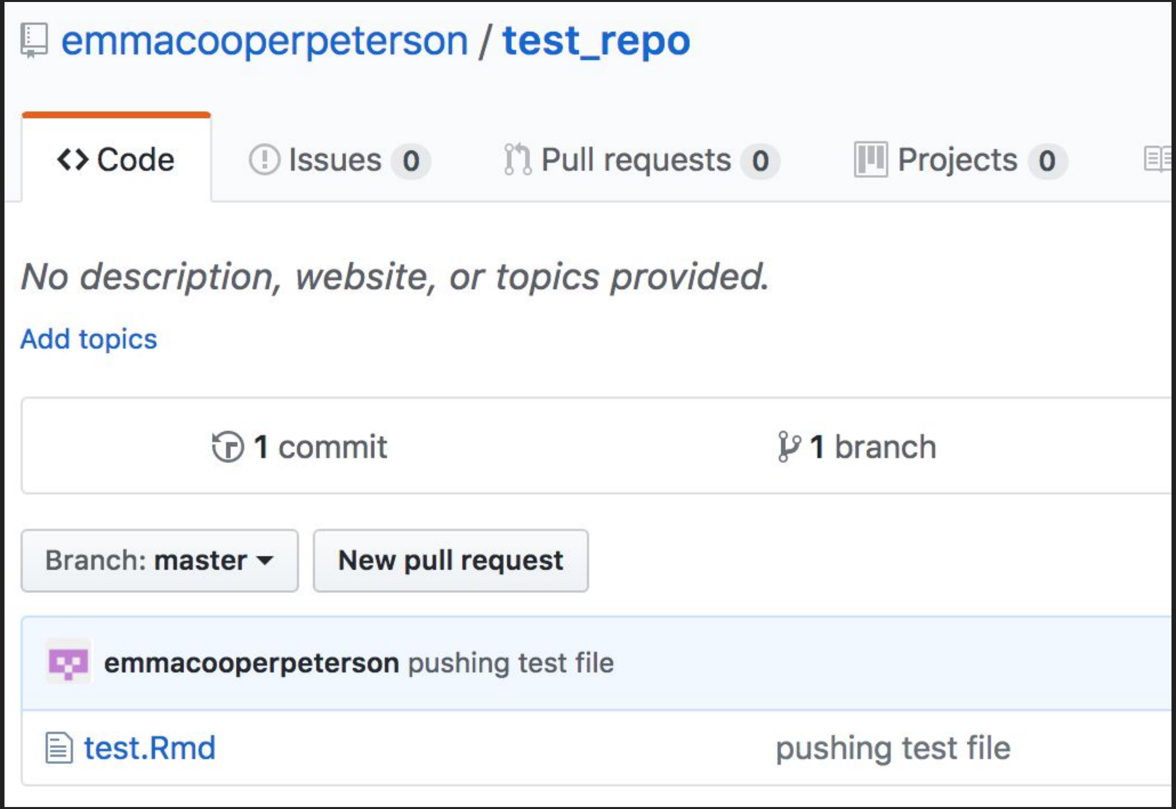
```
[👽  test_repo:  git add test.Rmd                                   ]
[👽  test_repo:  git commit -m 'pushing test file'                  ]
[master (root-commit) fb1d8d7] pushing test file
 1 file changed, 30 insertions(+)
 create mode 100644 test.Rmd
```

Finally, we’re ready to push.



```
[👽  test_repo:  git push                                                         ]
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 705 bytes | 705.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/emmacooperpeterson/test_repo.git
 * [new branch]      master -> master
👽  test_repo:
```

Now the file is there in my repo on GitHub.

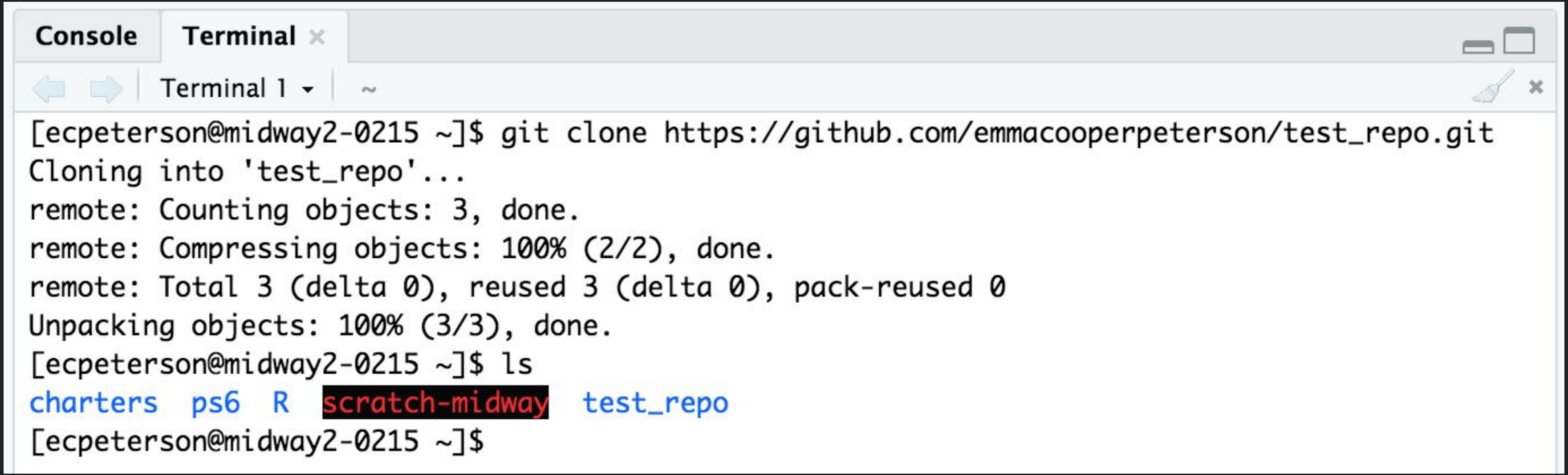

emmacooperpeterson / test_repo

No description, website, or topics provided.

emmacooperpeterson pushing test file

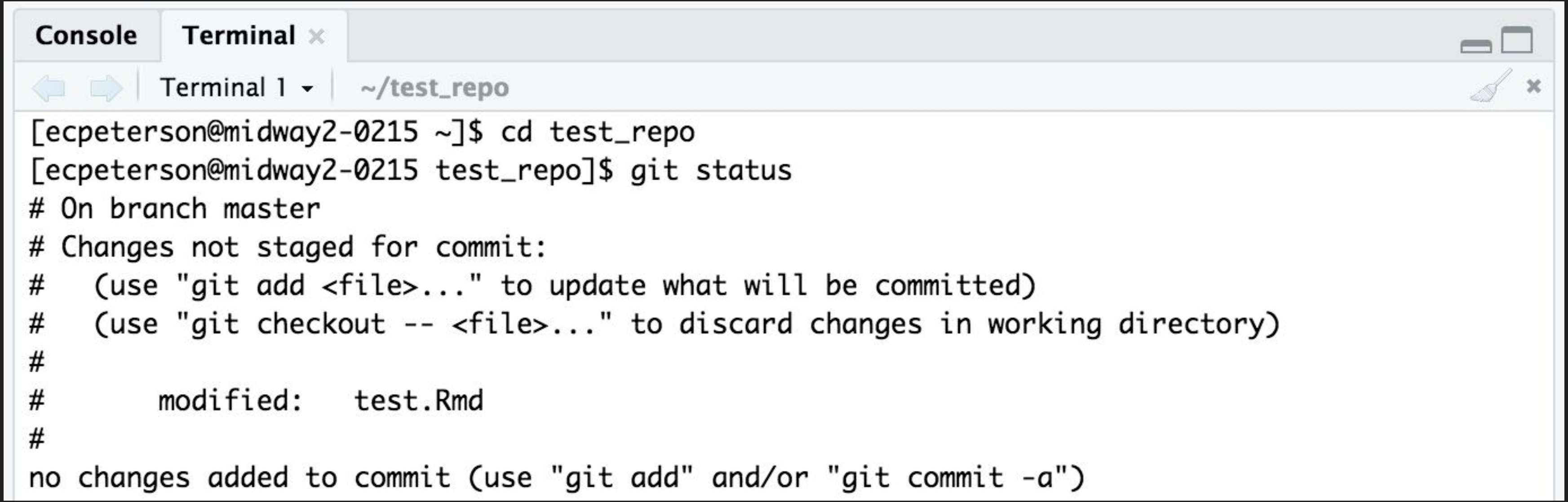
test.Rmd pushing test file

To access my repo from the server, I repeat the cloning process from the RStudio terminal on the server. After cloning, `ls` shows that the new repo was added.



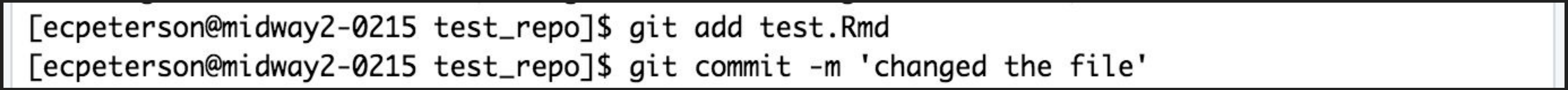
```
[ecpeterson@midway2-0215 ~]$ git clone https://github.com/emmacooperpeterson/test_repo.git
Cloning into 'test_repo'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
[ecpeterson@midway2-0215 ~]$ ls
charters  ps6  R  scratch-midway  test_repo
[ecpeterson@midway2-0215 ~]$
```

Use `cd` to move into the cloned folder. From the server, I've made a change to my test.Rmd file. `git status` shows that the file has been modified.



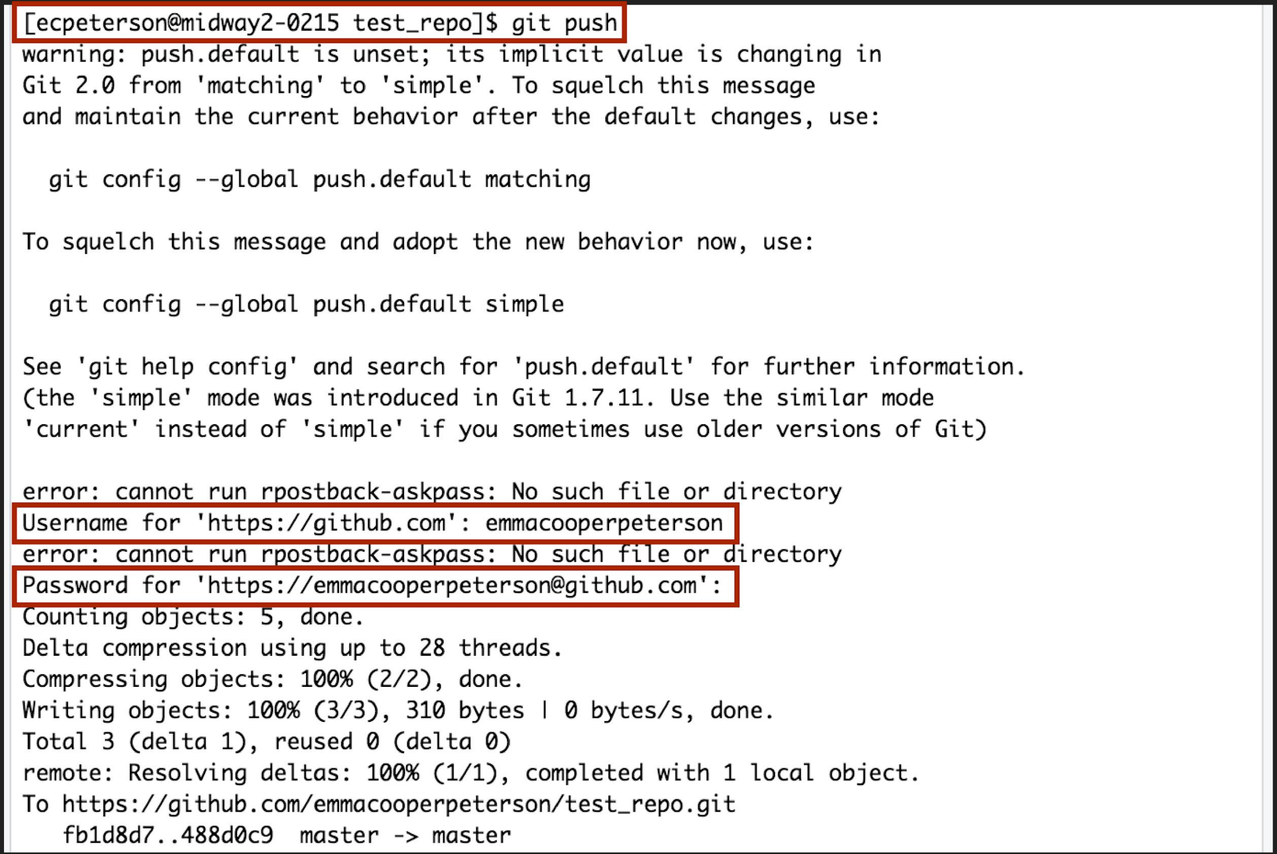
```
[ecpeterson@midway2-0215 ~]$ cd test_repo
[ecpeterson@midway2-0215 test_repo]$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   test.Rmd
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Add and commit the changes.



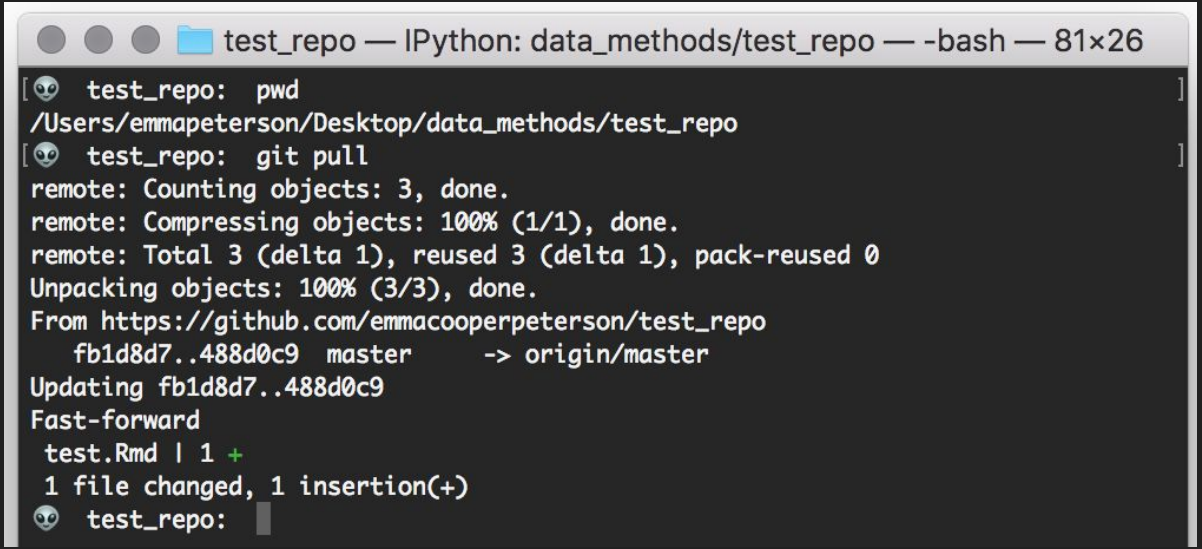
```
[ecpeterson@midway2-0215 test_repo]$ git add test.Rmd
[ecpeterson@midway2-0215 test_repo]$ git commit -m 'changed the file'
```

Push the changes and enter your GitHub username and password.



```
[ecpeterson@midway2-0215 test_repo]$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

error: cannot run rpostback-askpass: No such file or directory
Username for 'https://github.com': emmacooperpeterson
error: cannot run rpostback-askpass: No such file or directory
Password for 'https://emmacooperpeterson@github.com':
Counting objects: 5, done.
Delta compression using up to 28 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 310 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/emmacooperpeterson/test_repo.git
   fb1d8d7..488d0c9  master -> master
```

Now I want to work locally again. `pwd` shows that I have navigated to the correct directory. Use `git pull` to copy the changes that you made on the server onto your laptop.



```
[👽  test_repo:  pwd                                                            ]
/Users/emmapeterson/Desktop/data_methods/test_repo
[👽  test_repo:  git pull                                                       ]
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/emmacooperpeterson/test_repo
   fb1d8d7..488d0c9  master     -> origin/master
Updating fb1d8d7..488d0c9
Fast-forward
 test.Rmd | 1 +
 1 file changed, 1 insertion(+)
 👽  test_repo:
```

Note: If your repo on GitHub contains changes not present on your local computer, you must `git pull` before pushing anything. It's best practice to always pull first.

Additional Resources

- Tutorial (best place to start)
- Cheat sheet of common git commands
- Learn more about the concept of version control here

Vocabulary

Branching and Merging

***Not necessary for this class**

- **Branch:** The main version of your code is called the master branch. When a new branch is created, it contains a copy of the code from the master branch. Changes made to the new branch do not affect the code on the master branch.

Vocabulary

Branching and Merging

*Not necessary for this class

- **Merge:** Merging is the process by which git reconciles differences between different versions of code. Git tries to handle merging automatically, but it might need your help.