Demissie

Spring 2021, Big Data Analytics Lab # 1

Git Download

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

:

Demissie

For Mac Users:

Installing on macOS

There are several ways to install Git on a Mac. The easiest is probably to install the Xcode Command Line Tools. On Mavericks (10.9) or above you can do this simply by trying to run `git` from the Terminal the very first time.

```
$ git --version
```

If you don't have it installed already, it will prompt you to install it.

If you want a more up to date version, you can also install it via a binary installer. A macOS Git installer is maintained and available for download at the Git website, at https://git-scm.com/download/mac.

Figure 7. Git macOS Installer.

You can also install it as part of the GitHub for macOS install. Their GUI Git tool has an option to install command line tools as well. You can download that tool from the GitHub for macOS website, at https://desktop.github.com.

For Windows users:

Installing on Windows

There are also a few ways to install Git on Windows. The most official build is available for download on the Git website. Just go to https://git-scm.com/download/win and the download will start automatically. Note that this is a project called Git for Windows, which is separate from Git itself; for more information on it, go to https://gitforwindows.org.

To get an automated installation you can use the Git Chocolatey package. Note that the Chocolatey package is community maintained.

Another easy way to get Git installed is by installing GitHub Desktop. The installer includes a command line version of Git as well as the GUI. It also works well with PowerShell, and sets up solid credential caching and sane CRLF settings. We'll learn more about those things a little later, but suffice it to say they're things you want. You can download this from the GitHub Desktop website.

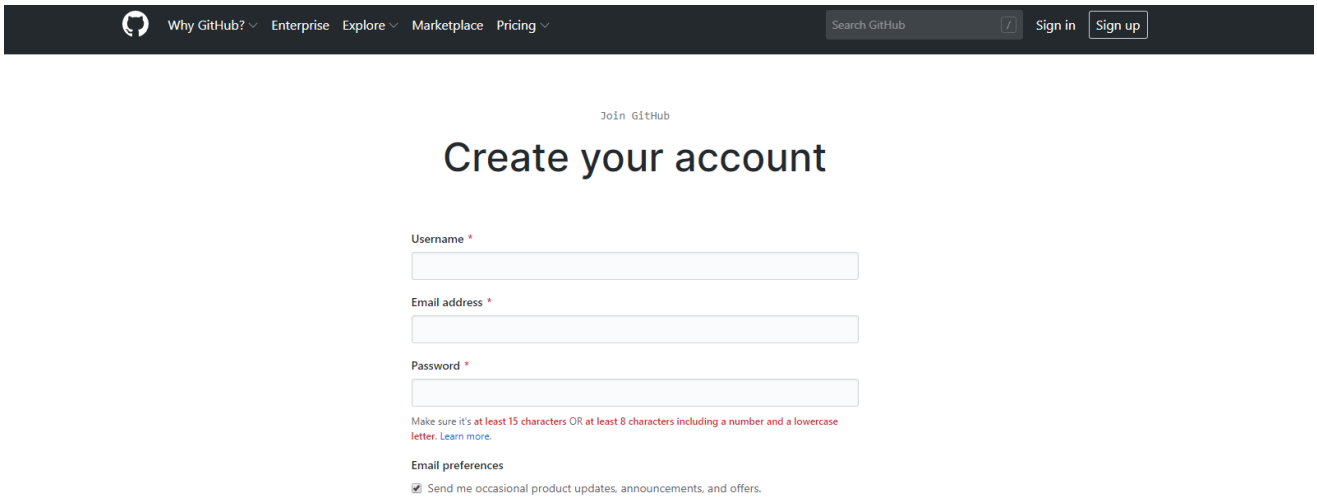
Demissie

Git Installation

Just follow the default settings

GitHub

Once you've down with Git, create a Free GitHub account at this link:
https://github.com/join
Then, you will be navigated to this webpage:



Git Tutorial

Step 1: Create a local git repository

When creating a new project on your local machine using git, you'll first create a new repository.

To begin, check the default working directory using the pwd command. Move to where you want to place the project on your local machine using the cd command.

To initialize a git repository in the root of the folder, run the git init command.

```
ypaph@DESKTOP-4JOKKUO MINGW64 ~
$ mkdir BYGB7990

ypaph@DESKTOP-4JOKKUO MINGW64 ~
$ cd BYGB7990/

ypaph@DESKTOP-4JOKKUO MINGW64 ~/BYGB7990
$
```

```
ypaph@DESKTOP-4JOKKUO MINGW64 ~/BYGB7990
$ git init
Initialized empty Git repository in C:/Users/ypaph/BYGB7990/.git/
```

Step 2: Registration

```
ypaph@DESKTOP-4JOKKUO MINGW64 ~/BYGB7990 (master)
$ git config --global user.email "zyang173@fordham.edu"

ypaph@DESKTOP-4JOKKUO MINGW64 ~/BYGB7990 (master)
$ git config --global user.name "Zhihan"
```

Step 3: Add a new file to the repository

Go ahead and add a new file to the project using any text editor you like or just running a touch command.

However, you could always copy and paste folders or files directly to your git working directory.

Once you've added or modified files in a folder containing a git repo, git will notice that changes have been made inside the repo. Git won't officially keep track of the file (that is, put it in a commit - we'll talk more about commits next) unless you use the add command

Then, you can use git status command to check the status.

```
ypaph@DESKTOP-4JOKKUO MINGW64 ~/BYGB7990 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Group project/NBA_Player_Types-master.zip
        new file:   Group project/NBA_Player_Types-master/NBA_Player_Types-maste
r/.gitignore
        new file:   Group project/NBA_Player_Types-master/NBA_Player_Types-maste
r/.ipynb_checkpoints/step_1_data_acquisition-checkpoint.ipynb
        new file:   Group project/NBA_Player_Types-master/NBA_Player_Types-maste
r/.ipynb_checkpoints/step_2_machine_learning_lda-checkpoint.ipynb
        new file:   Group project/NBA_Player_Types-master/NBA_Player_Types-maste
r/.ipynb_checkpoints/step_3_eda-checkpoint.ipynb
        new file:   Group project/NBA_Player_Types-master/NBA_Player_Types-maste
r/.ipynb_checkpoints/step_4_lineups_data_cleaning-checkpoint.ipynb
        new file:   Group project/NBA_Player_Types-master/NBA_Player_Types-maste
r/README.md
        new file:   Group project/NBA_Player_Types-master/NBA_Player_Types-maste
r/assets/BR_data_tables.png
```

Step 4: Create a commit

Use the git commit -m “logging” command to commit your previous changes. Notice here “logging” can be changed to whatever you want to input.

```
ypaph@DESKTOP-4JOKKUO MINGW64 ~/BYGB7990 (master)
$ git commit -m "BYGB7978 Group"
[master (root-commit) bfd872f] BYGB7978 Group
 62 files changed, 14501 insertions(+)
 create mode 100644 Group project/NBA_Player_Types-master.zip
 create mode 100644 Group project/NBA_Player_Types-master/NBA_Player_Types-maste
r/.gitignore
```

Step 5: Create a new branch

Branches allow you to make a new feature but are worried about making changes to the main project to the main project while developing the feature. This is where git branches come in.

Branches allow developers to move back and forth between ‘states’ of a project. For instance, if you want to add a new page to website you can simply create a new branch without affecting the main part of the project. Once you’re done with the page, you can then merge changes from your branch into the master branch.

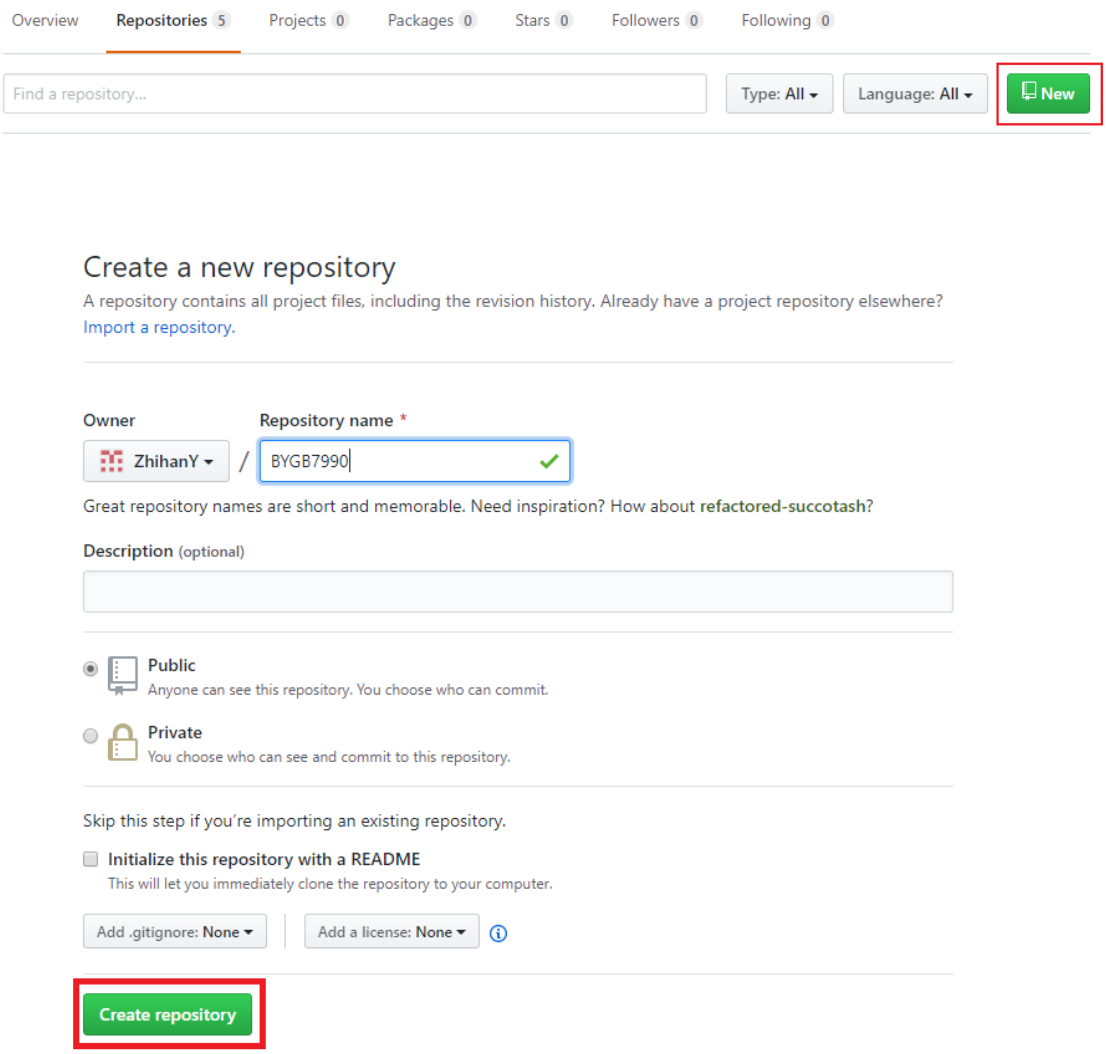
To create a new branch, run git checkout -b <branch name>, and git creates a new branch and automatically switch to that new branch. You can also use git branch <name> to create a new branch.

Step 6: Create a new repository on GitHub & Push a branch to GitHub

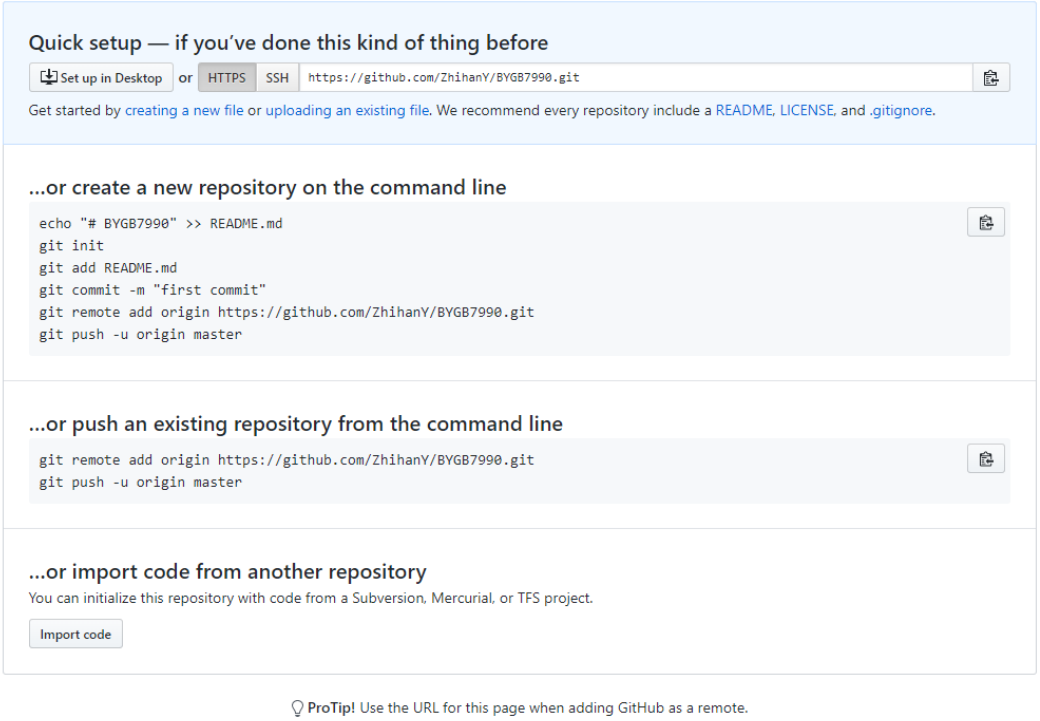
If you only want to keep track of your code locally, you don't need to use GitHub. But if you want to work with a team, you can use GitHub to collaboratively modify

the project's code.

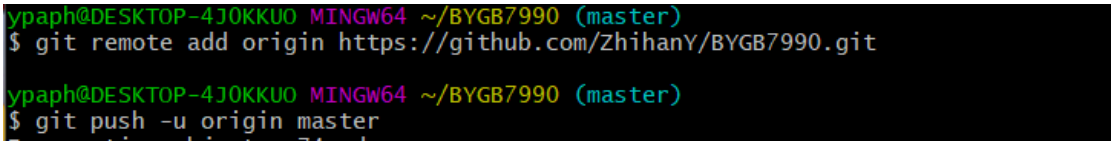
To create a new repo on GitHub, log in and go to the GitHub home page. You should see a green ‘New’ button:



After Setting up your repository, you will see a page like this:



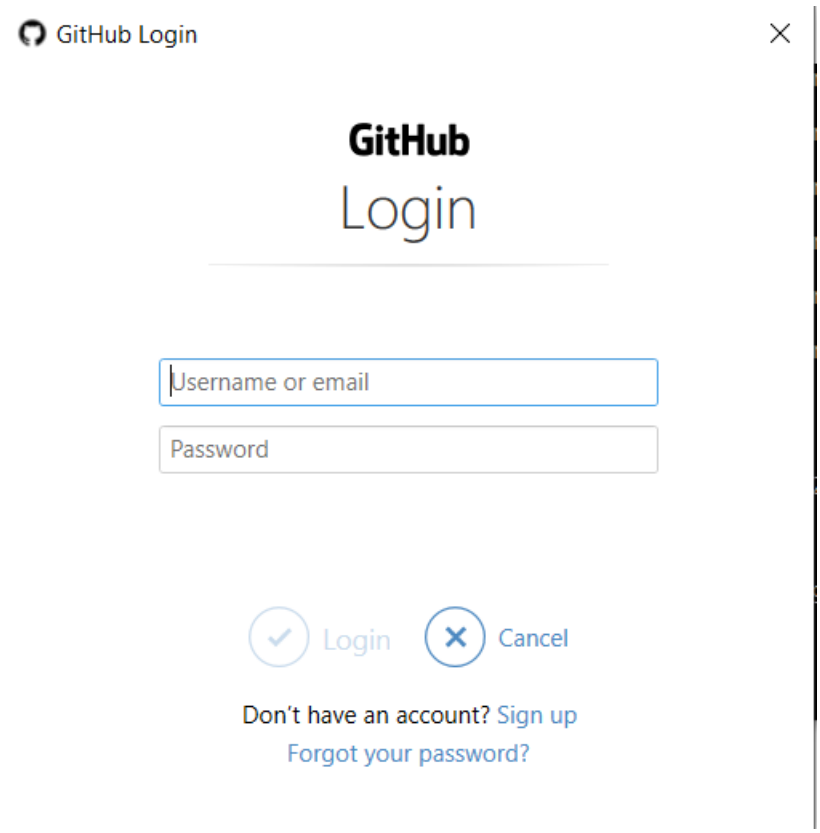
Then, type the commands shown in the middle section of the previous picture (Yours are different from mine):



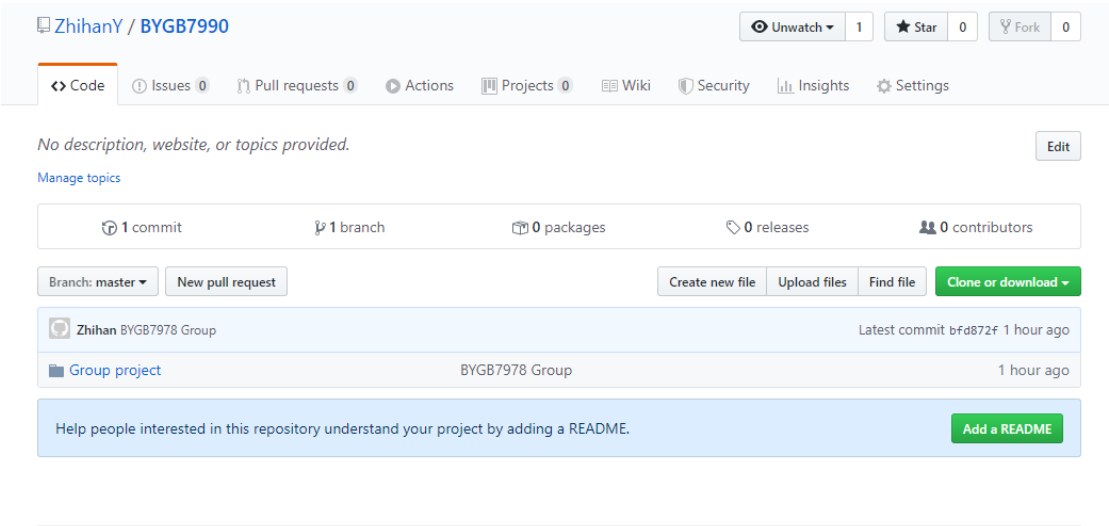
The reason why there is a -u command in the second row is that your remote repository is now empty. Next time, when you upload documents you won't need to use that so the command will be:

$ git push origin master

It will pop out a GitHub login window, enter your registered email address and password to sign in.



Ultimately, you will see something like this:



Step 7: Write a README file

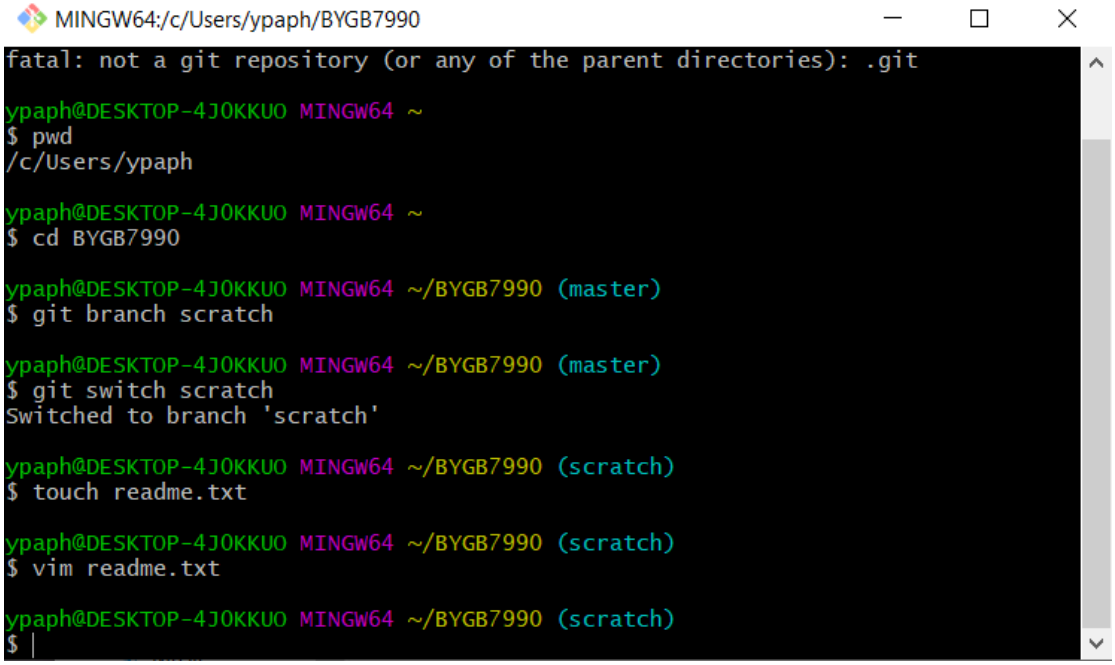
Open source community is growing rapidly. Developers release new open source projects on GitHub every day. As a result, it's becoming more and more difficult to get your own project and stand out from the community. However, you can do a few things to increase your chances of grabbing other developer's attention. One effective and

simple technique is putting up a nice-looking and helpful README file.
Good README should include enough details to help a new user get started, e.g. how to compile, how to install, and how to starting integrating.
Lastly, after learning about the writing fundamentals of README, let's dive into the styling of README which also called formatting.
Formatting is an essential part of README. You can learn about how to format your README from here and here.
README Templates:
https://gist.github.com/fvcproductions/1bfc2d4aecb01a834b46
https://github.com/dbader/readme-template

Step 8: Create a pull request

A pull request is a way to alert a repo's owner that you want to make some changes to their code. It allows them to review and make sure it looks good before putting your changes to the master branch.

Let's make a new local branch and then create a readme file:



Push the new branch to GitHub:

```
ypaph@DESKTOP-4JOKKUO MINGW64 ~/BYGB7990 (scratch)
$ git push origin scratch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 297 bytes | 99.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'scratch' on GitHub by visiting:
remote:      https://github.com/ZhihanY/BYGB7990/pull/new/scratch
remote:
To https://github.com/ZhihanY/BYGB7990.git
 * [new branch]      scratch -> scratch
```

Then you will see a Compare & pull request button where you can create your pull request: