

Git fundamentals

Command or concept	Description
git init	
change set	Git doesn't work with files. It works with change sets
git init	Initializes a repo from where ever the command is executed
git status	Shows the contents of the staging area as well as lists the contents of untracked files
git add	Adds files to the staging index. The staging index can be thought of as a “pre-commit” area. Files are committed from the staging area to the repo. git add . adds all the changes in the working directory to the index.
git rm	Removes items from the staging index. To remove a directory from the index: git rm --cached -r directory-name https://stackoverflow.com/questions/1143796/
git commit	Commits what is currently in the staging index to the project repo. The -m option is used to add the commit message which should be less than 80 chars. You can edit the most recent commit message using: git commit --amend https://help.github.com/articles/changing-a-commit-message/
git log	
git remote add	Adds a remote to the local git system. The example below adds a remote repo used for my Getting and Cleaning Data Coursera project: git remote add origin https://github.com/MichaelSzczepaniak/HumanActivityRecognition.git The newly added remote is stored in the .git/config file of the project.
git remote	Lists the remotes the local git project knows about. Using -v option lists urls to push and fetch (which are usually the same).
git remote rm origin	Removes the origin remote
git push	Pushes local changes to a remote. The example below is what is typically used for pushing changes from the local master branch up to a remote named origin : git push -u origin master If you need to revert a commit already pushed to the remote, use: git push origin +abc123^:master where abc123 is the SHA of the commit you want to remove. This forces the

	master branch of the origin remote repo to the parent of abc123. Reference: http://christoph.ruegg.name/blog/git-howto-revert-a-commit-already-pushed-to-a-remote-reposit.html
git clone <url>	<p>Pulls in a remote repo and uses the project name as the folder. Execute this from the parent dir under which you want to pull the project in under.</p> <p>You can rename the directory that clone creates by passing the name you want after the url. See the following video for more details: ...\\Dropbox\\sw_dev\\training\\Lynda\\Lynda - GIT Essentials\\13. Remotes\\05. Remotes-Cloning a remote repository.mov</p>
git fetch	Pulls updates from a remote into the tracking branch (typically called origin/master). A fetch syncs the remote with the tracking branch, but before these changes can be used by the local master branch, a merge must be done (typically a fast-forward merge)
git branch	List the branches in the current project
git checkout -b <branch name>	Creates a new branch from the current branch and puts you into that branch.
Git branch -D <branch name>	Forces the remove of a local branch
git push origin --delete <branch_name>	Removes a remote branch. Reference: https://stackoverflow.com/questions/2003505
.gitignore	<p>Add this file to your project to specify which files should be ignored by the git version control system. Windows Explorer doesn't allow you to create files that start with . (dot) so use the following work around:</p> <p>1) Hold SHIFT, right click the folder you're in, then select Open command window here</p>

[illegible]

Git learnings and common tasks

Date: mm/dd/20yy

Date: 10/20/2017

To enter a multi-line commit message in git, use the following format:

```
git commit -m '$first line\nsecond line'
```

Words of the form `$'string'` are treated specially. The word expands to `string`, with backslash-escaped characters replaced as specified by the ANSI C standard.

Reference:

<https://stackoverflow.com/questions/5064563/add-line-break-to-git-commit-m-from-command-line#5065387>

Date: 10/20/2017

After cloning a repository that had 3 branches (master, enh_configfile, and enh_qualplots), I only had master showing up when I did a `git branch`. I wanted to track all three branches. How to do this was described here:

<https://stackoverflow.com/questions/379081/track-all-remote-git-branches-as-local-branches>
I upvoted answer given by slm on 12/1/2014. I copied the following text into git bash and it worked just fine:

```
for i in $(git branch -r | grep -vE "HEAD|master"); do  
    git branch --track ${i#*/} $i; done
```

Date: 10/15/2017

Rename a local and remote branch in git

1. Rename your local branch. If you are on the branch you want to rename:

```
git branch -m new-name
```

If you are on a different branch:

```
git branch -m old-name new-name
```

2. Delete the old-name remote branch and push the new-name local branch.

```
git push origin :old-name new-name
```

3. Reset the upstream branch for the new-name local branch.
Switch to the branch and then:

```
git push origin -u new-name
```

Reference: <https://multiplestates.wordpress.com/2015/02/05/rename-a-local-and-remote-branch-in-git/>

Date: 12/2/2015

Remove a file from a Git repository without deleting it from the local file system:

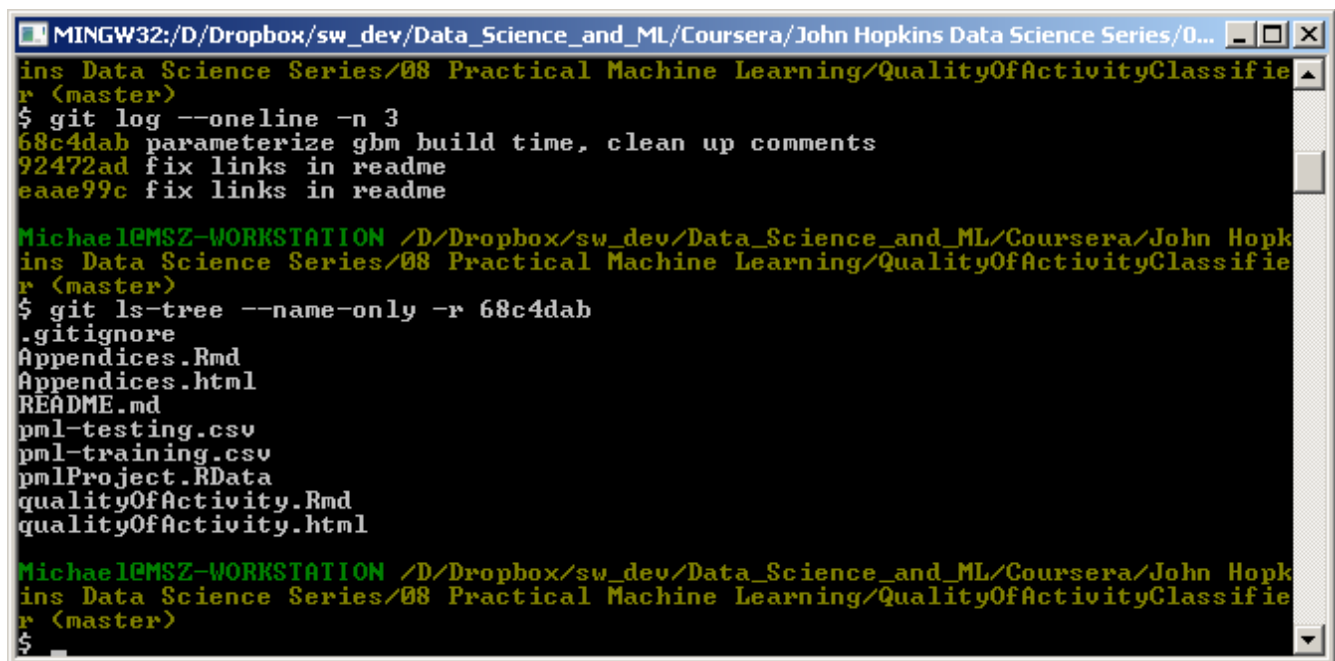
For a file: `git rm --cached mylogfile.log`

For a directory: `git rm --cached -r mydirectory`

Reference: <http://stackoverflow.com/questions/1143796>

Date: 12/2/2015

How to list all the files in the last commit. Use: `git ls-tree --name-only -r <commit-ish>`
WARNING!!! DO NOT COPY AND PASTE THE ABOVE COMMAND DIRECTLY INTO A GIT BASH CLI!!! IF YOU DO, YOU MAY SEE STRANGE BEHAVIOR BECAUSE OF HOW THIS CONSOLE HANDLES VARIOUS FONTS. TO BE SAFE, COPY AND PASTE INTO SOMETHING LIKE NP++ AND THEN COPY & PASTE FROM NP++.



```
MINGW32:/D:/Dropbox/sw_dev/Data_Science_and_ML/Coursera/John Hopkins Data Science Series/0...
ins Data Science Series/08 Practical Machine Learning/QualityOfActivityClassifie
r (master)
$ git log --oneline -n 3
68c4dab parameterize gbm build time, clean up comments
92472ad fix links in readme
eaae99c fix links in readme

Michael@MSZ-WORKSTATION /D:/Dropbox/sw_dev/Data_Science_and_ML/Coursera/John Hopk
ins Data Science Series/08 Practical Machine Learning/QualityOfActivityClassifie
r (master)
$ git ls-tree --name-only -r 68c4dab
.gitignore
Appendices.Rmd
Appendices.html
README.md
pml-testing.csv
pml-training.csv
pmlProject.RData
qualityOfActivity.Rmd
qualityOfActivity.html

Michael@MSZ-WORKSTATION /D:/Dropbox/sw_dev/Data_Science_and_ML/Coursera/John Hopk
ins Data Science Series/08 Practical Machine Learning/QualityOfActivityClassifie
r (master)
$
```

reference: <http://stackoverflow.com/questions/424071/how-to-list-all-the-files-in-a-commit>

Date: 5/11/2014

Fix for how to exit after doing a **git diff** in windows 7. It was hanging as shown below:

```
Michael@MSZ-WORKSTATION /c/dev/play/warehouse (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:   app/views/products/details.scala.html

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

Michael@MSZ-WORKSTATION /c/dev/play/warehouse (master)
$ git diff
diff --git a/app/views/products/details.scala.html b/app/views/products/details.scala.html
index c49f302..902a4d6 100644
--- a/app/views/products/details.scala.html
+++ b/app/views/products/details.scala.html
@@ -4,7 +4,7 @@

  @main("Product form") {
    <h1>Product form</h1>
-   @helper.form(action = routes.Products.save()) {
+   @helper.form(action = routes.Products.save()) { <!-- specify the save() action in Products contr
    <fieldset>
      <legend>Product (@productForm("name").valueOr("New"))</legend>
      @helper.inputText(productForm("ean"), '_label -> "EAN")
  }
}

(END)
```

Simply type **Q** to exit.

Reference:

<http://stackoverflow.com/questions/9929457/how-do-i-exit-the-results-of-git-diff-in-git-bash-on-windows>

Date: 5/11/2015

How to set up NP++ as the git core editor:

```
git config --global core.editor "'C:/Program Files (x86)/Notepad++/notepad++.exe'
-multiInst -notabbar -nosession -noPlugin"
```

reference: <http://stackoverflow.com/questions/1634161/how-do-i-use-notepad-or-other-with-msysgit>

Date: 5/11/2014

Fix for the

fatal: pathspec '...' did not match any files error that I was seeing as shown below:

```

Michael@MSZ-WORKSTATION /c/dev/play/warehouse/public/bootstrap/css (master)
$ ls -la
total 384
drwxr-xr-x 10 Michael Administ 4096 May 11 17:06 .
drwxr-xr-x 5 Michael Administ 4096 May 11 13:58 ..
-rw-r--r-- 1 Michael Administ 14936 Feb 13 08:24 bootstrap-theme.css
-rw-r--r-- 1 Michael Administ 38388 Feb 13 08:24 bootstrap-theme.css.map
-rw-r--r-- 1 Michael Administ 13185 Feb 13 08:24 bootstrap-theme.min.css
-rw-r--r-- 1 Michael Administ 133503 May 11 17:06 bootstrap.2.3.2.css
-rw-r--r-- 1 Michael Administ 121220 Feb 13 08:24 bootstrap.css
-rw-r--r-- 1 Michael Administ 245960 Feb 13 08:24 bootstrap.css.map
-rw-r--r-- 1 Michael Administ 106022 May 11 15:41 bootstrap.min.2.3.2.css
-rw-r--r-- 1 Michael Administ 99961 May 11 17:05 bootstrap.min.css

Michael@MSZ-WORKSTATION /c/dev/play/warehouse/public/bootstrap/css (master)
$ git rm bootstrap.2.3.2.css
fatal: pathspec 'bootstrap.2.3.2.css' did not match any files

Michael@MSZ-WORKSTATION /c/dev/play/warehouse/public/bootstrap/css (master)
$

```

The file I was trying to remove was not being tracked (had an entry in the `.gitignore` file).

References:

<http://stackoverflow.com/questions/15931238/unable-to-remove-file-that-really-exists-fatal-pathspec-did-not-match-any>

Date: 5/11/2014

How to configure git to use bcompare as the difftool and NP++ as the default editor. Below is a copy of the User level `.gitconfig` (located in `C:\Users\Michael\.gitconfig`)

```

[user]
    name = Michael Szczepaniak
    email = golf4enlightenment@yahoo.com

[core]
    editor = "C:/Program Files (x86)/Notepad++/notepad++.exe" -multiInst -notabbar -nosession -noPlugin

[mergetool "bcompare"]
    path = "C:/Program Files (x86)/Beyond Compare 3/BComp.exe"

[merg]
    tool = bcompare

[diff]
    tool = bc3

[difftool "bc3"]
    cmd = "C:/Program Files (x86)/Beyond Compare 3/BCompare.exe" -expandall -solo %LOCAL% %REMOTE%

[color]
    ui = true

```

References:

<http://stackoverflow.com/questions/15873874>

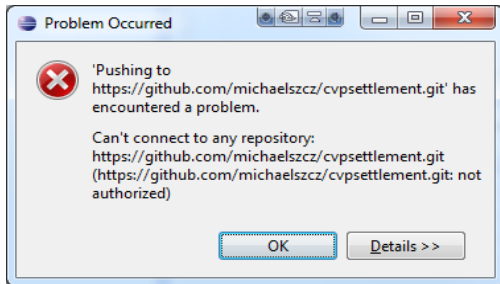
<http://www.scootersoftware.com/vbulletin/showthread.php?t=9449>

Date: 10/13/2013

Fix for the

Can't connect to any repository:... not authorized error

I was seeing in using the Egit eclipse plugin



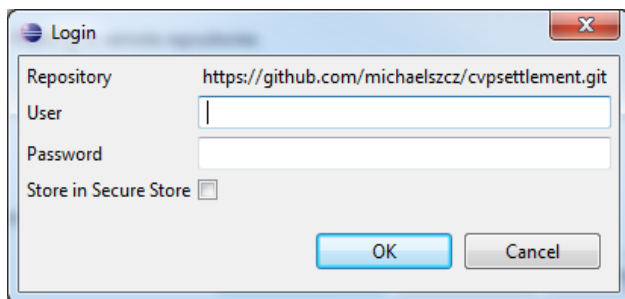
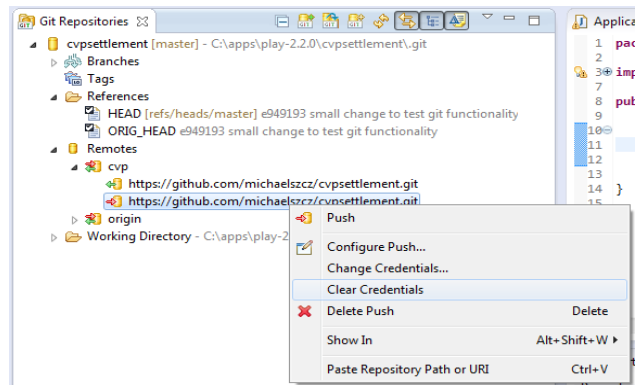
Description: When I tried to push a change to my remote git repository, I got this message:

The problem was that I had set the default repository password setting incorrectly in the **Egit** plugin and needed to change it to the correct one so the eclipse could make the connection.

I found that I could

right click the push under the **Remotes** list in the Git perspective, clear the credentials, and then then try the push and it worked IF I HAD COMMITTED THE CHANGES FIRST as shown to the right:

After doing this, when I tried to do the commit and push, I was prompted for my git repository credentials as shown below:



I made sure the uncheck the Store in Secure Store option after providing my credentials and the push succeeded. Happy day!

Date: 7/25/2015

How do you do a diff between the different areas of git?

- **git diff**: Show differences between your working directory and the index.
- **git diff --cached**: Show differences between the index and the most recent commit.
- **git diff HEAD**: Show the differences between your working directory and the most recent commit.
- **git diff <branch1> <branch2> <file_name>**: Shows difference between two branches

Replace **diff** with **difftool** to tell git to launch your configured diff too to do the diff (e.g. bcompare)

reference: <http://www.gitguys.com/topics/git-diff/>

Date: 7/25/2015

What's the best way to rename a file in git?

Date: 7/26/2015 How do you revert a single file:

```
git checkout -- <filename>
```

reference: <http://www.norbauer.com/rails-consulting/notes/git-revert-reset-a-single-file.html>

Date: 8/5/2015 Editing/amending the last commit message

```
git commit --amend
```

Will open your editor, allowing you to change the commit message of the most recent commit. Additionally, you can set the commit message directly in the command line with:

```
git commit --amend -m "New commit message"
```

If you've already pushed your commit up to your remote branch, then you'll need to force push the commit with

```
git push <remote> <branch> --force  
# Or  
git push <remote> <branch> -f
```

Warning: force-pushing will overwrite the remote branch with the state of your local one. If there are commits on the remote branch that you don't have in your local branch, you *will* lose those commits.

Warning: be cautious about amending commits that you have already shared with other people. Amending commits essentially *rewrites* them to have different [SHA](#) IDs, which poses a problem if other people have copies of the old commit that you've rewritten.

reference: <http://stackoverflow.com/questions/179123/edit-an-incorrect-commit-message-in-git>

Date: 11/10/2015 Workflow of creating remote repo then pushing and fetching from two local repos (laptop and workstation). My thinking around this example is that knowing how to quickly do this will allow me to work from anywhere and from any machine.

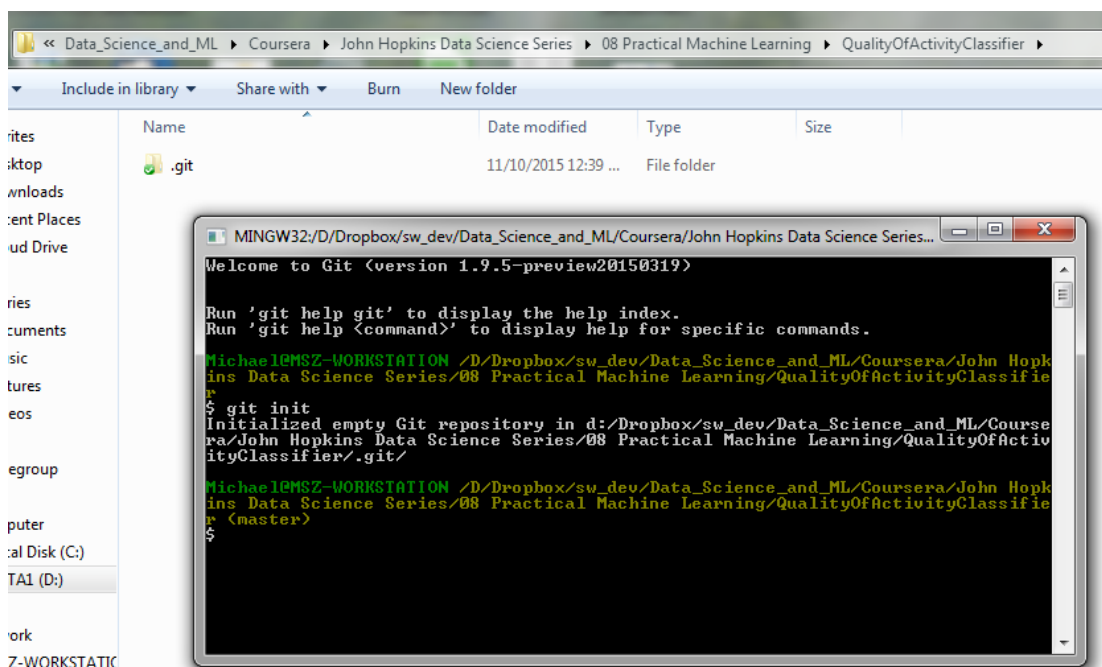
1. Create a remote repository on github as shown on the right.



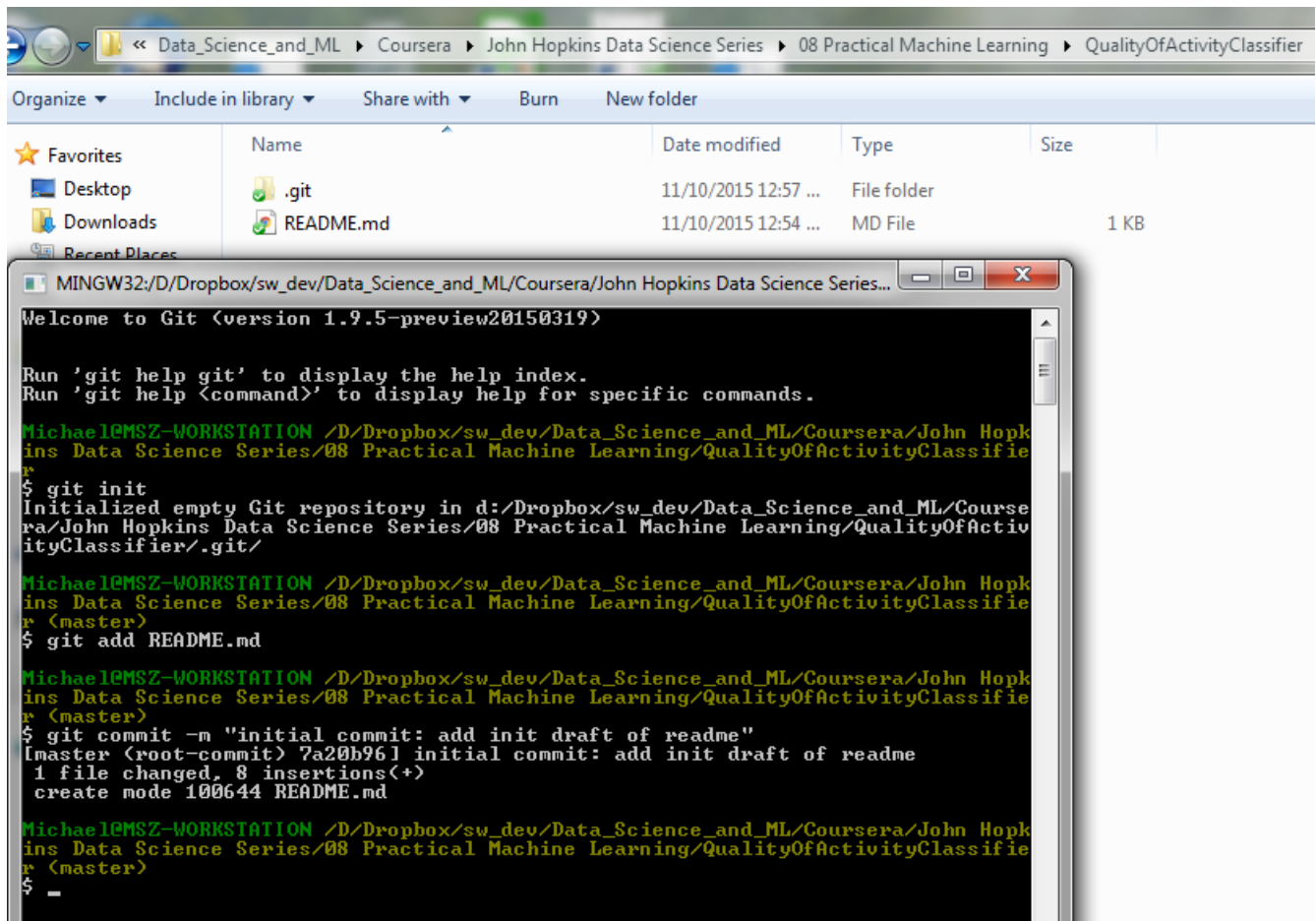
2. Create a local project directory and do a

`git init`

to initialize a repo in this location as shown below left.

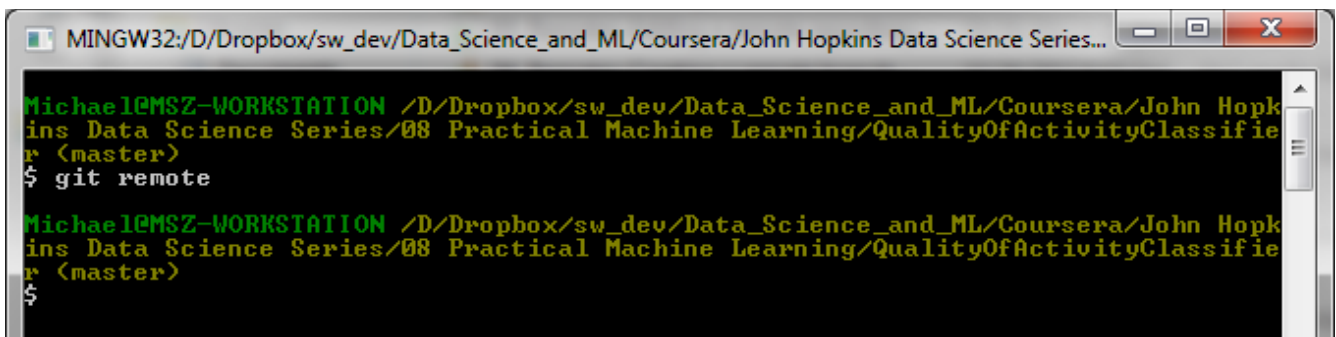


3. Create a **README.md** file and commit it to the local repo. We may use this file to make changes, push to the repository, and fetch back changes made from (the) other machine(s).



4. Configure the remote repository locally so that it knows where to **push to** and **fetch from**. Doing a **git remote**

at this point will not show anything because no remotes are set up yet:



```
Michael@MSZ-WORKSTATION /D:/Dropbox/sw_dev/Data_Science_and_ML/Coursera/John Hopkins Data Science Series/08 Practical Machine Learning/QualityOfActivityClassifier (master)
$ git remote
```

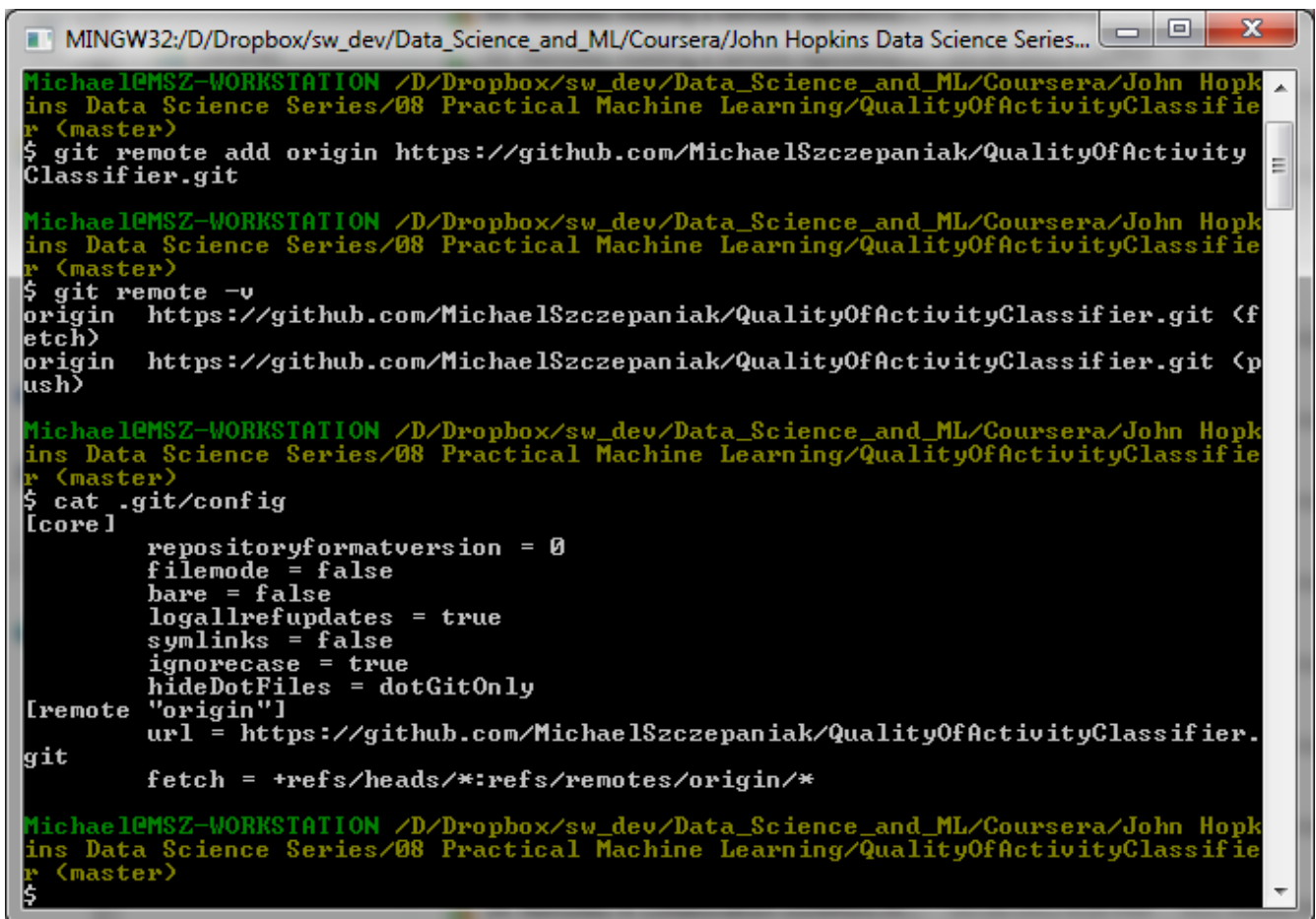
From within the project directory, execute the following command from the git command line:

git remote add origin <alias> <url of remote> For our example that would be:

git remote add origin <https://github.com/MichaelSzczechpaniak/QualityOfActivityClassifier.git>

Note: The alias “origin” is used by convention, but this could be anything.

Adding the remote adds an entry into the **.git/config** file in the project directory and provides git a reference to **push to** and **fetch from**. This new entry can be seen below:



```
Michael@MSZ-WORKSTATION /D:/Dropbox/sw_dev/Data_Science_and_ML/Coursera/John Hopkins Data Science Series/08 Practical Machine Learning/QualityOfActivityClassifier (master)
$ git remote add origin https://github.com/MichaelSzczechpaniak/QualityOfActivityClassifier.git

Michael@MSZ-WORKSTATION /D:/Dropbox/sw_dev/Data_Science_and_ML/Coursera/John Hopkins Data Science Series/08 Practical Machine Learning/QualityOfActivityClassifier (master)
$ git remote -v
origin https://github.com/MichaelSzczechpaniak/QualityOfActivityClassifier.git (fetch)
origin https://github.com/MichaelSzczechpaniak/QualityOfActivityClassifier.git (push)

Michael@MSZ-WORKSTATION /D:/Dropbox/sw_dev/Data_Science_and_ML/Coursera/John Hopkins Data Science Series/08 Practical Machine Learning/QualityOfActivityClassifier (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
    hideDotFiles = dotGitOnly
[remote "origin"]
    url = https://github.com/MichaelSzczechpaniak/QualityOfActivityClassifier.git
    fetch = +refs/heads/*:refs/remotes/origin/*
```

5. With the remote configured, we can now push our changes to the remote by doing the following from the git bash CLI from within our project repository.

```
git push -u origin master
```

The `-u` option is short for `--set-upstream` and says to push our code up and also make a note of that fact because we're going to be coming back and working with this branch frequently. When we do a `git clone` it tracks the remote branch, and we can see that tracking in the `.git/config` file of the project.

6. (Update 11/28/2015) At this point, we have a remote configured, a local project configured, and we have pushed changes to that project from the workstation (machine which originally created the project). If we just worked on this project using the workstation, we would just make changes locally and then push those changes to the remote periodically.

If I want to use the laptop to work on a project originated from the workstation, I first need to clone the project, make my changes, and then push those changes back to the remote. The problem with this situation arises when additional work is done from the different laptop and workstation systems. For the rest of this example, I'll use the **AbaloneRegression** project.

The project has been cloned on the laptop as shown below:

```

MINGW32:/C:/data/dev/AbaloneRegression
Welcome to Git (version 1.9.5-preview20150319)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Michael@MICHAEL_SZ_W520 /C:/data/dev
$ git clone https://github.com/MichaelSzczepaniak/AbaloneRegression
Cloning into 'AbaloneRegression'...
remote: Counting objects: 35, done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 35 (delta 17), reused 27 (delta 9), pack-reused 0
Unpacking objects: 100% (35/35), done.
Checking connectivity... done.

Michael@MICHAEL_SZ_W520 /C:/data/dev
$ ls
AbaloneRegression

Michael@MICHAEL_SZ_W520 /C:/data/dev
$ cd AbaloneRegression/

Michael@MICHAEL_SZ_W520 /C:/data/dev/AbaloneRegression (master)
$

```

The next thing to do is make some changes from the laptop system and then...

...push those changes to the repository as shown on the right:

```

MINGW32:/C:/data/dev/AbaloneRegression

Michael@MICHAEL_SZ_W520 /C:/data/dev/AbaloneRegression (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/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:   AbaloneRegression.Rmd

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

Michael@MICHAEL_SZ_W520 /C:/data/dev/AbaloneRegression (master)
$ git commit -am"add config for laptop dev"
[master 8acc837] add config for laptop dev
1 file changed, 2 insertions(+), 1 deletion(-)

Michael@MICHAEL_SZ_W520 /C:/data/dev/AbaloneRegression (master)
$ git push
Username for 'https://github.com': MichaelSzczepaniak
Password for 'https://MichaelSzczepaniak@github.com':
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 381 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To https://github.com/MichaelSzczepaniak/AbaloneRegression
aac24a4..8acc837  master -> master

Michael@MICHAEL_SZ_W520 /C:/data/dev/AbaloneRegression (master)
$

```

Now, when I make a change on the **workstation**, commit that change and then try to push that change back to the remote, I see the following error:

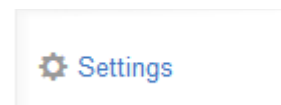

```
MINGW32:/D/dev/AbaloneRegression
Michael@MSZ-WORKSTATION /D/dev/AbaloneRegression <master>
$ git push
Username for 'https://github.com': MichaelSzczeponiak
Password for 'https://MichaelSzczeponiak@github.com':
To https://github.com/MichaelSzczeponiak/AbaloneRegression.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/MichaelSzczeponiak/AbaloneRegression.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
Michael@MSZ-WORKSTATION /D/dev/AbaloneRegression <master>
$
```

To fix this, I need to pull the changes down from the remote (fetch), merge those changes locally on the workstation, and then push those changes back up to the remote. The next steps show how to do this.

7. Next step is to pull down the remote changes.

Date: 11/10/2015 Delete a repository on github

1. Navigate to the main page of your repo on github.
2. Click on the **Settings** button on the right of the page:
3. Scroll down to the **Danger Zone** section at the bottom of the page.
4. Click **Delete this repository**.



Date: 11/10/2015 Make github HTML pages viewable in your browser:

<https://www.dropbox.com/s/yy5iv12puu6cyd5/Project%20-%20Why%20and%20How%20to%20use%20gh-pages.pdf?dl=0>

Date: 11/28/2015 Create an alias for showtool:

<http://stackoverflow.com/questions/17558221/how-can-i-view-the-output-of-git-show-in-a-diff-viewer-like-meld-kdiff3-etc/33975521#33975521>

<https://git-scm.com/book/en/v2/Git-Basics-Git-Aliases>

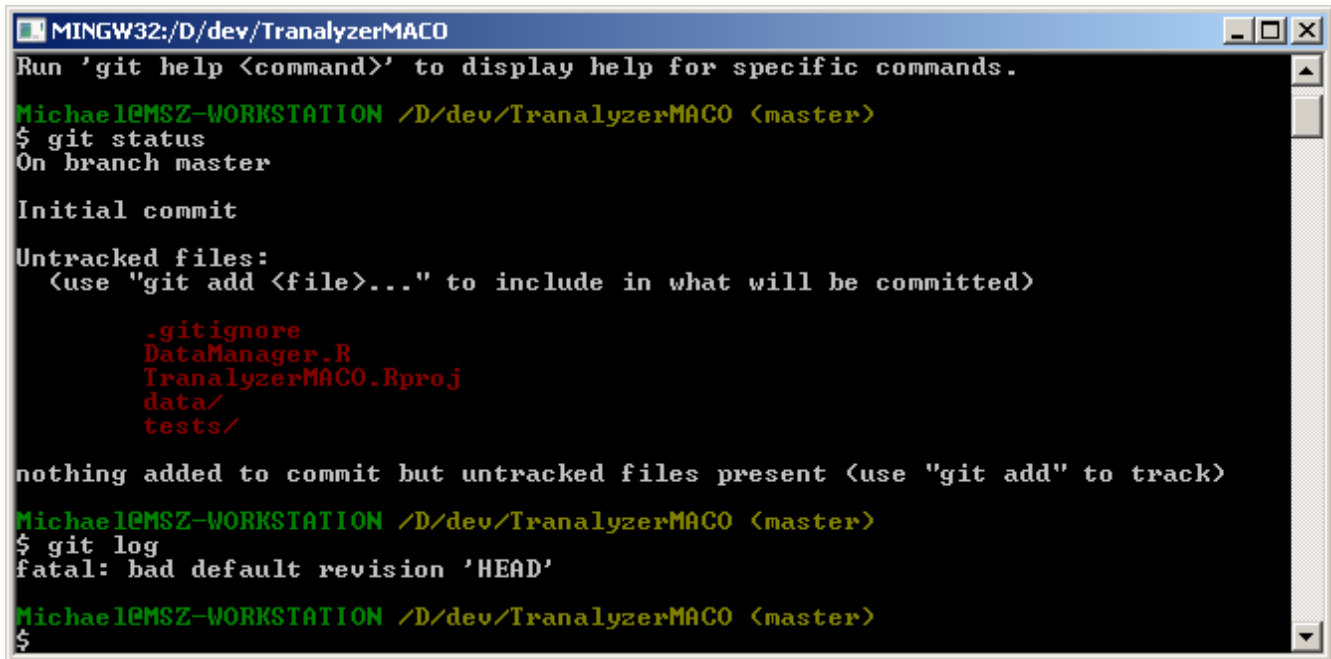
Date: 11/28/2015 Get rid of the push.default message

```
git config --global push.default simple
```

reference: <http://stackoverflow.com/questions/13148066/warning-push-default-is-unset-its-implicit-value-is-changing-in-git-2-0>

Date: 05/02/2016

Issue: When I did a git log on a newly initialized repo, I was seeing the following:

A screenshot of a Windows command prompt window titled "MINGW32:/D/dev/TranalyzerMACO". The window shows the output of several git commands. First, "git status" is run, showing "On branch master" and "Initial commit". It lists untracked files: ".gitignore", "DataManager.R", "TranalyzerMACO.Rproj", "data/", and "tests/". Then, "git log" is run, resulting in a "fatal: bad default revision 'HEAD'" error. The prompt is at the end of the line.

```
MINGW32:/D/dev/TranalyzerMACO
Run 'git help <command>' to display help for specific commands.
Michael@MSZ-WORKSTATION /D/dev/TranalyzerMACO <master>
$ git status
On branch master

Initial commit

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

        .gitignore
        DataManager.R
        TranalyzerMACO.Rproj
        data/
        tests/

nothing added to commit but untracked files present (use "git add" to track)
Michael@MSZ-WORKSTATION /D/dev/TranalyzerMACO <master>
$ git log
fatal: bad default revision 'HEAD'
Michael@MSZ-WORKSTATION /D/dev/TranalyzerMACO <master>
$
```

Solution: Just need to do a commit and this goes away. Reference: <http://stackoverflow.com/questions/15628720/fatal-bad-default-revision-head#answer-27118243>

Date: 05/17/2017 How to diff the same file in two different branches

reference: <http://stackoverflow.com/questions/4099742/how-to-compare-files-from-two-different-branches>

git difftool <branch1> <branch2> <file_name>

For example: **git difftool master disconnectbug ui.R**