

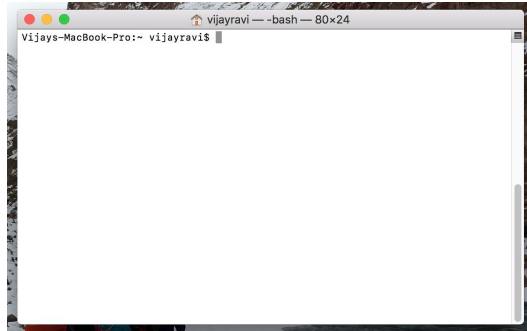
STATS 202A : STATISTICS PROGRAMMING

HOMEWORK 8: UNIX COMMANDS

NAME: VIJAY RAVI

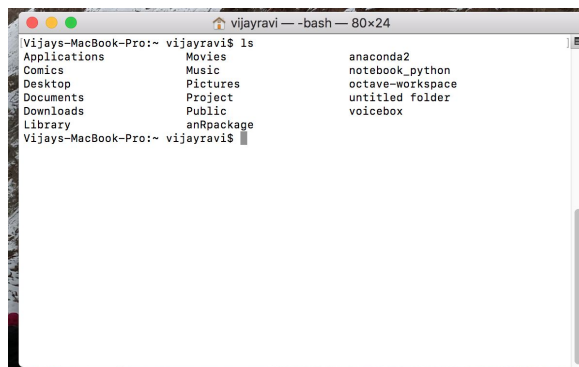
UID: 805033666

Command Terminal on MAC where UNIX commands are executed.

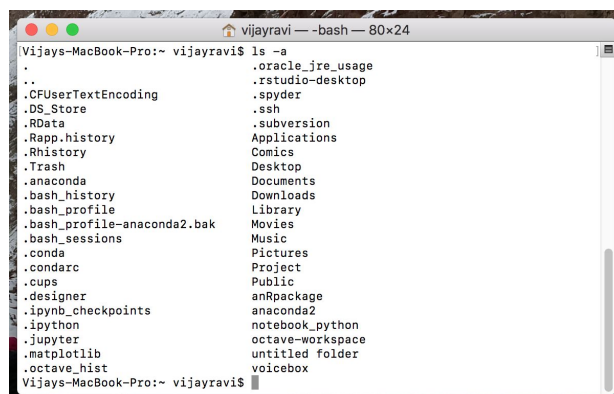


The following UNIX commands have been tried. Attached is the screenshot with each and brief explanation of each command functionality.

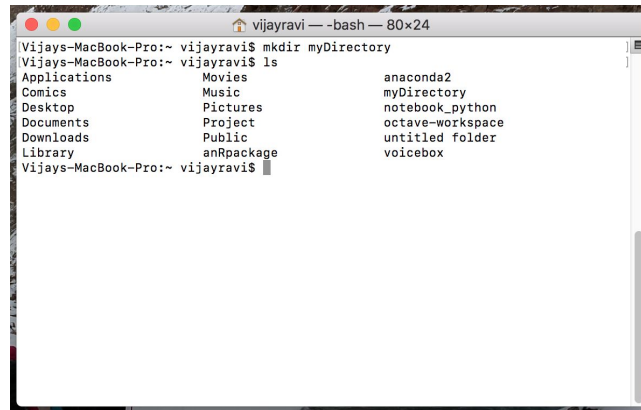
1. **'ls'**: The **ls** command (lowercase L and lowercase S) lists the contents of your current working directory.



2. **'ls -a'** : List all files including the hidden files.

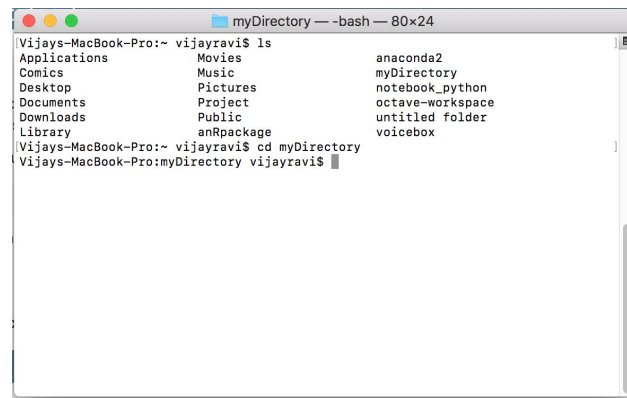


3. **'mkdir'**: make a directory. Creating a directory called 'myDirectory'. Using 'ls' to check if directory has been created.



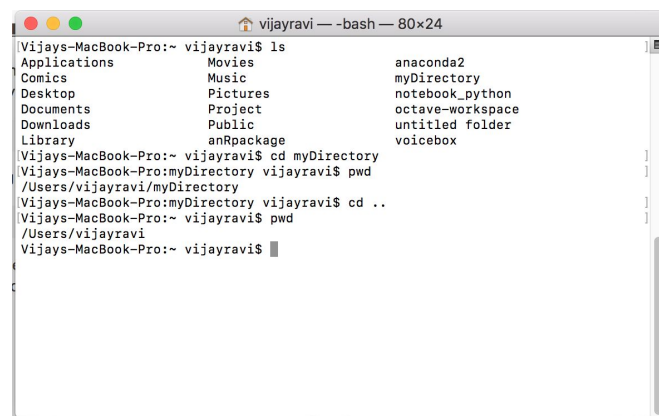
```
Vijays-MacBook-Pro:~ vijayravi$ mkdir myDirectory
Vijays-MacBook-Pro:~ vijayravi$ ls
Applications      Movies            anaconda2
Comics            Music             myDirectory
Desktop           Pictures          notebook_python
Documents         Project          octave-workspace
Downloads        Public            untitled folder
Library          anRpackage       voicebox
Vijays-MacBook-Pro:~ vijayravi$
```

4. **'cd directory'**: change working directory to the named **directory**. Changed the working directory to the newly created directory.



```
Vijays-MacBook-Pro:~ vijayravi$ ls
Applications      Movies            anaconda2
Comics            Music             myDirectory
Desktop           Pictures          notebook_python
Documents         Project          octave-workspace
Downloads        Public            untitled folder
Library          anRpackage       voicebox
Vijays-MacBook-Pro:~ vijayravi$ cd myDirectory
Vijays-MacBook-Pro:myDirectory vijayravi$
```

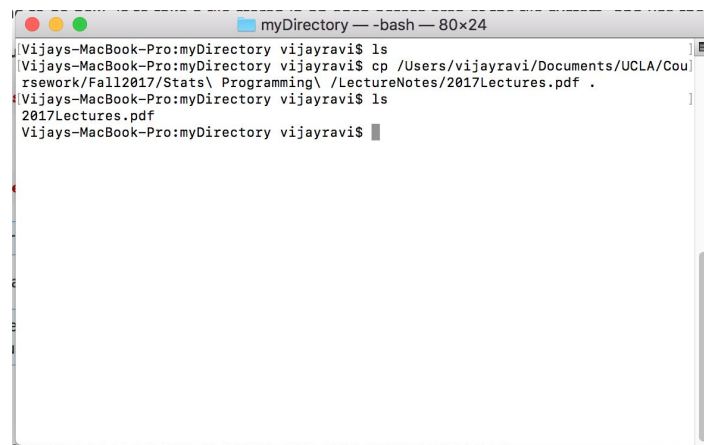
5. **'cd'**: Change working directory to home directory.
6. **'cd ..'**: Change working directory to parent directory.
7. **'pwd'**: Display the path of the current working directory.



```
Vijays-MacBook-Pro:~ vijayravi$ ls
Applications      Movies            anaconda2
Comics            Music             myDirectory
Desktop           Pictures          notebook_python
Documents         Project          octave-workspace
Downloads        Public            untitled folder
Library          anRpackage       voicebox
Vijays-MacBook-Pro:~ vijayravi$ cd myDirectory
Vijays-MacBook-Pro:myDirectory vijayravi$ pwd
/Users/vijayravi/myDirectory
Vijays-MacBook-Pro:myDirectory vijayravi$ cd ..
Vijays-MacBook-Pro:~ vijayravi$ pwd
/Users/vijayravi
Vijays-MacBook-Pro:~ vijayravi$
```

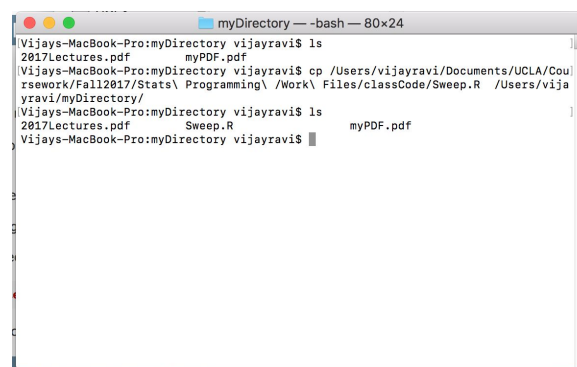
Changed the working directory back to parent directory. Confirmed the change by printing the path.

8. 'Cp file1 file2': Copy file1 and rename it as file2. Here I am copying the lecture notes from its file location to the current working directory. Using 'ls' to verify if file exists in copied location. Using the (.) notation for current working directory. **myDirectory** is empty before copying but has the **2017Lectures.pdf** after file has been copied.



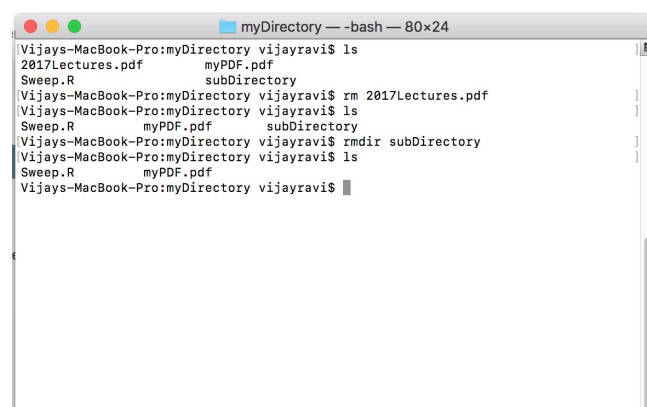
```
myDirectory -- -bash -- 80x24
Vijays-MacBook-Pro:myDirectory vijayravi$ ls
Vijays-MacBook-Pro:myDirectory vijayravi$ cp /Users/vijayravi/Documents/UCLA/Cou
rsework/Fall2017/Stats\ Programming\ /LectureNotes/2017Lectures.pdf .
Vijays-MacBook-Pro:myDirectory vijayravi$ ls
2017Lectures.pdf
Vijays-MacBook-Pro:myDirectory vijayravi$
```

9. 'mv file1 file2': moves (or renames) **file1** to **file2**. Here, I am moving Sweep.R from it's location to 'myDirectory'. If the file is in the same directory, mv command can be used to rename the file.



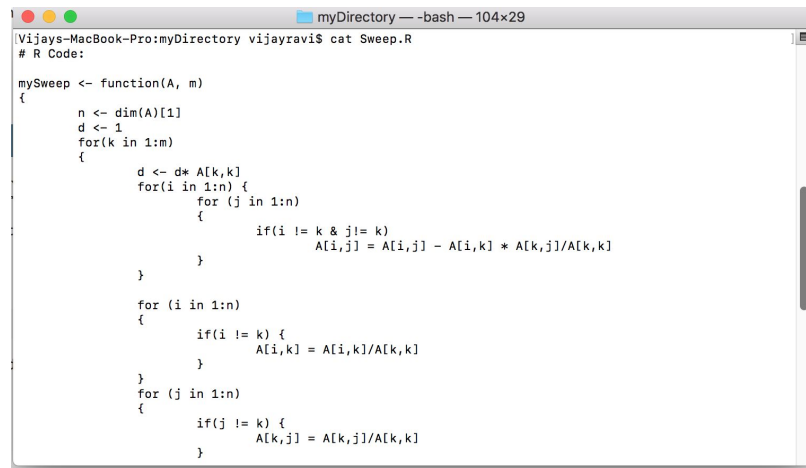
```
myDirectory -- -bash -- 80x24
Vijays-MacBook-Pro:myDirectory vijayravi$ ls
2017Lectures.pdf      myPDF.pdf
Vijays-MacBook-Pro:myDirectory vijayravi$ cp /Users/vijayravi/Documents/UCLA/Cou
rsework/Fall2017/Stats\ Programming\ /Work\ Files/classCode/Sweep.R /Users/vija
yravi/myDirectory/
Vijays-MacBook-Pro:myDirectory vijayravi$ ls
2017Lectures.pdf      Sweep.R      myPDF.pdf
Vijays-MacBook-Pro:myDirectory vijayravi$
```

10. 'rm ' and 'rmdir': These commands are used to remove/delete a file or a directory. Here I am removing the file **2017Lectures.pdf** and the directory '**subDirectory**'.



```
myDirectory -- -bash -- 80x24
Vijays-MacBook-Pro:myDirectory vijayravi$ ls
2017Lectures.pdf      myPDF.pdf
Sweep.R              subDirectory
Vijays-MacBook-Pro:myDirectory vijayravi$ rm 2017Lectures.pdf
Vijays-MacBook-Pro:myDirectory vijayravi$ ls
Sweep.R      myPDF.pdf      subDirectory
Vijays-MacBook-Pro:myDirectory vijayravi$ rmdir subDirectory
Vijays-MacBook-Pro:myDirectory vijayravi$ ls
Sweep.R      myPDF.pdf
Vijays-MacBook-Pro:myDirectory vijayravi$
```

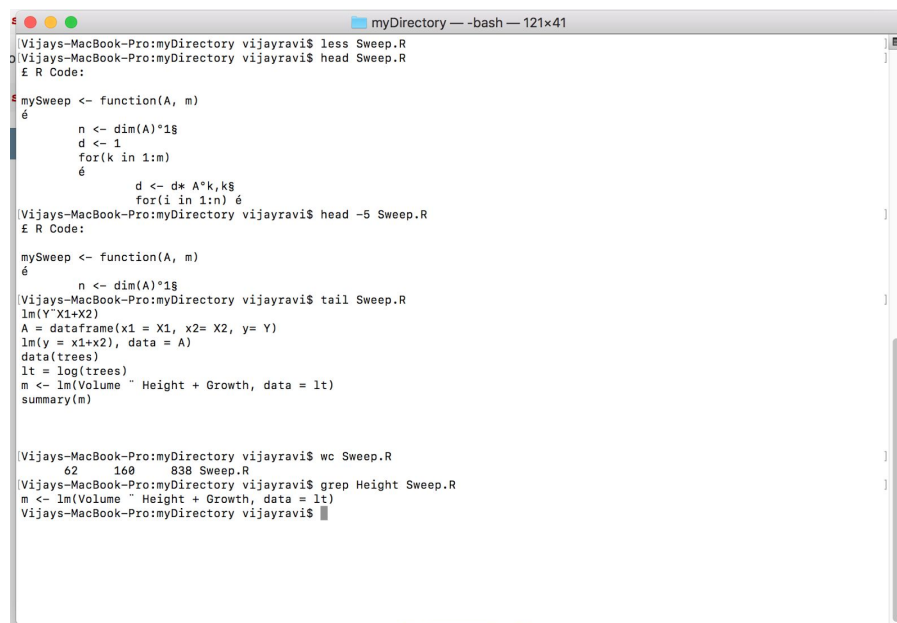
11. 'clear': clears the command prompt/terminal. History is still accessible by up/down cursor key.
12. 'cat': The command cat can be used to display the contents of a file on the screen. Here, I am displaying the contents of the sweep.R file.



```
myDirectory — -bash — 104x29
[Vijays-MacBook-Pro:myDirectory vijayravi$ cat Sweep.R
# R Code:

mySweep <- function(A, m)
{
  n <- dim(A)[1]
  d <- 1
  for(k in 1:m)
  {
    d <- d * A[k,k]
    for(i in 1:n) {
      for (j in 1:n)
      {
        if(i != k & j != k)
          A[i,j] = A[i,j] - A[i,k] * A[k,j]/A[k,k]
      }
    }
    for (i in 1:n)
    {
      if(i != k) {
        A[i,k] = A[i,k]/A[k,k]
      }
    }
    for (j in 1:n)
    {
      if(j != k) {
        A[k,j] = A[k,j]/A[k,k]
      }
    }
  }
}
```

13. 'less file1' : Displays the contents of file1 one page at a time. Press spacebar to navigate to the next page. Press q to exit reading the file.
14. 'head file1': Displays first 10 lines of the file.
15. 'tail file1': Displays last 10 lines of the file.
16. 'grep **keyword** file1' : Returns the occurrence of the keyword in the file.
17. 'wc file1': Returns the count number of lines/words/characters in file



```
myDirectory — -bash — 121x41
[Vijays-MacBook-Pro:myDirectory vijayravi$ less Sweep.R
[Vijays-MacBook-Pro:myDirectory vijayravi$ head Sweep.R
# R Code:
mySweep <- function(A, m)
{
  n <- dim(A)[1]
  d <- 1
  for(k in 1:m)
  {
    d <- d * A[k,k]
    for(i in 1:n) {
      for (j in 1:n)
      {
        if(i != k & j != k)
          A[i,j] = A[i,j] - A[i,k] * A[k,j]/A[k,k]
      }
    }
    for (i in 1:n)
    {
      if(i != k) {
        A[i,k] = A[i,k]/A[k,k]
      }
    }
    for (j in 1:n)
    {
      if(j != k) {
        A[k,j] = A[k,j]/A[k,k]
      }
    }
  }
}

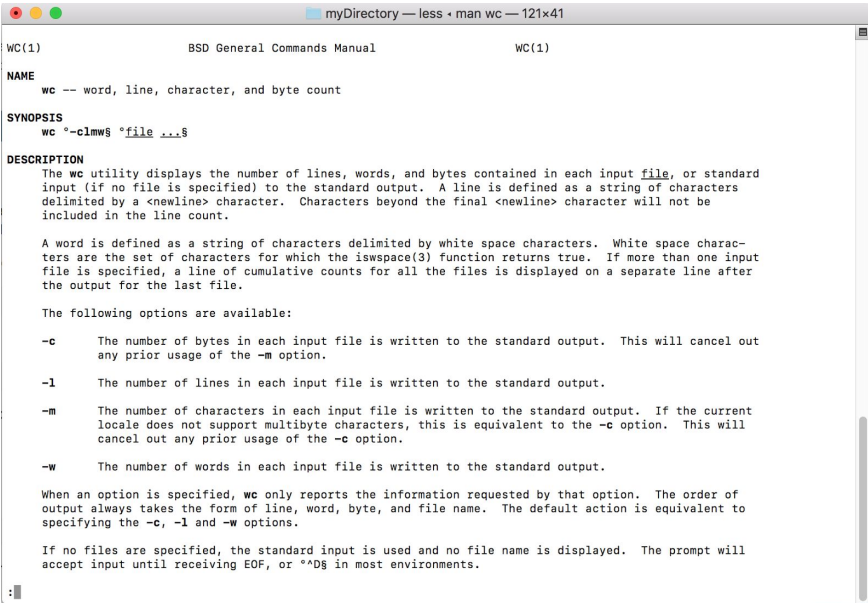
[Vijays-MacBook-Pro:myDirectory vijayravi$ head -5 Sweep.R
# R Code:
mySweep <- function(A, m)
{
  n <- dim(A)[1]
  d <- 1
  for(k in 1:m)
  {
    d <- d * A[k,k]
    for(i in 1:n) {
      for (j in 1:n)
      {
        if(i != k & j != k)
          A[i,j] = A[i,j] - A[i,k] * A[k,j]/A[k,k]
      }
    }
    for (i in 1:n)
    {
      if(i != k) {
        A[i,k] = A[i,k]/A[k,k]
      }
    }
    for (j in 1:n)
    {
      if(j != k) {
        A[k,j] = A[k,j]/A[k,k]
      }
    }
  }
}

[Vijays-MacBook-Pro:myDirectory vijayravi$ tail Sweep.R
lm(Y~X1+X2)
A = dataframe(x1 = X1, x2= X2, y= Y)
lm(y = x1+x2, data = A)
data(trees)
lt = log(trees)
m <- lm(Volume ~ Height + Growth, data = lt)
summary(m)

[Vijays-MacBook-Pro:myDirectory vijayravi$ wc Sweep.R
 62   160   838 Sweep.R
[Vijays-MacBook-Pro:myDirectory vijayravi$ grep Height Sweep.R
m <- lm(Volume ~ Height + Growth, data = lt)
[Vijays-MacBook-Pro:myDirectory vijayravi$
```

The file Sweep.R has the word 'Height' and has 62 lines, 160 words and 838 characters.

18. `'man command'` : read the online manual/Documentation page for a command. Use cursor to navigate. Press `'q'` to exit. `'man wc'` gives the following output:



```
WC(1) BSD General Commands Manual WC(1)

NAME
wc -- word, line, character, and byte count

SYNOPSIS
wc -c|lmw$ file ...$

DESCRIPTION
The wc utility displays the number of lines, words, and bytes contained in each input file, or standard input (if no file is specified) to the standard output. A line is defined as a string of characters delimited by a <newline> character. Characters beyond the final <newline> character will not be included in the line count.

A word is defined as a string of characters delimited by white space characters. White space characters are the set of characters for which the iswspace(3) function returns true. If more than one input file is specified, a line of cumulative counts for all the files is displayed on a separate line after the output for the last file.

The following options are available:

-c      The number of bytes in each input file is written to the standard output. This will cancel out any prior usage of the -m option.

-l      The number of lines in each input file is written to the standard output.

-m      The number of characters in each input file is written to the standard output. If the current locale does not support multibyte characters, this is equivalent to the -c option. This will cancel out any prior usage of the -c option.

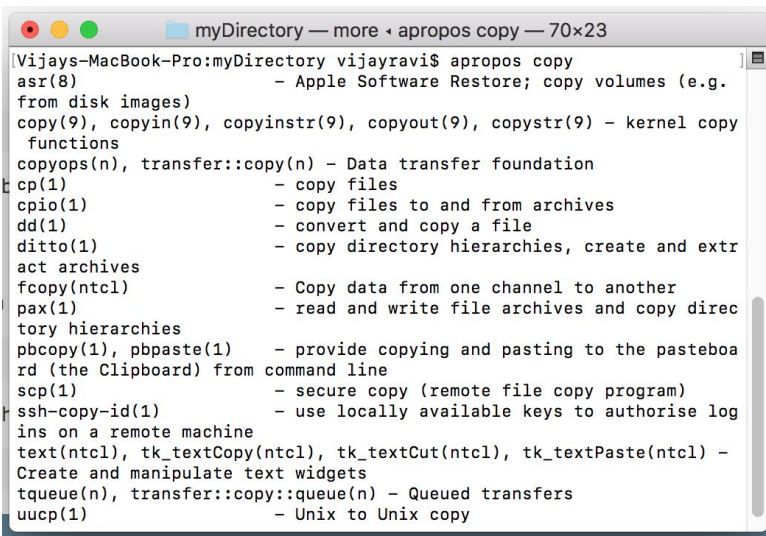
-w      The number of words in each input file is written to the standard output.

When an option is specified, wc only reports the information requested by that option. The order of output always takes the form of line, word, byte, and file name. The default action is equivalent to specifying the -c, -l and -w options.

If no files are specified, the standard input is used and no file name is displayed. The prompt will accept input until receiving EOF, or ^D$ in most environments.

:
```

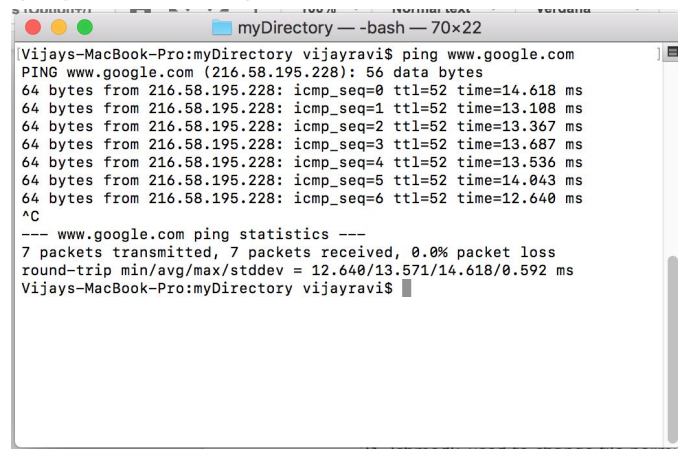
19. `'apropos keyword'` : match commands with keyword in their man(manual) pages. `'apropos copy'` gives the following output:



```
myDirectory — more · apropos copy — 70x23
Vijays-MacBook-Pro:myDirectory vijayravi$ apropos copy
asr(8) - Apple Software Restore; copy volumes (e.g.
from disk images)
copy(9), copyin(9), copyinstr(9), copyout(9), copystr(9) - kernel copy
functions
copyops(n), transfer::copy(n) - Data transfer foundation
cp(1) - copy files
cpio(1) - copy files to and from archives
dd(1) - convert and copy a file
ditto(1) - copy directory hierarchies, create and extr
act archives
fcopy(1) - Copy data from one channel to another
pax(1) - read and write file archives and copy direc
tory hierarchies
pbcopy(1), pbpaste(1) - provide copying and pasting to the pasteboa
rd (the Clipboard) from command line
scp(1) - secure copy (remote file copy program)
ssh-copy-id(1) - use locally available keys to authorise log
ins on a remote machine
text(1), tk_textCopy(1), tk_textCut(1), tk_textPaste(1) -
Create and manipulate text widgets
tqueue(n), transfer::copy::queue(n) - Queued transfers
uucp(1) - Unix to Unix copy
```

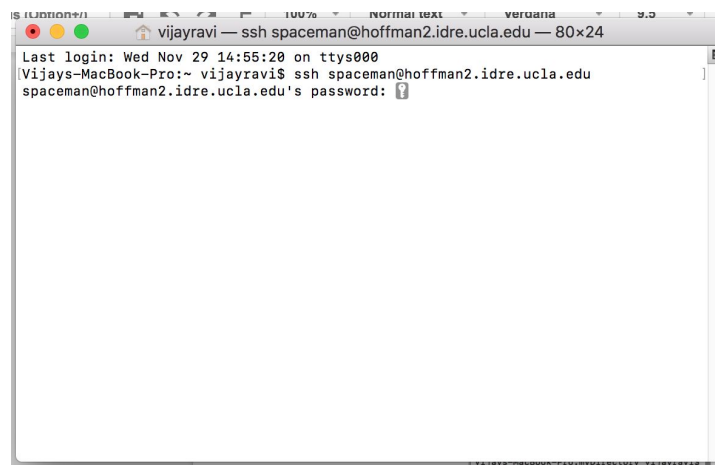
20. `'vim file1'`: Vim can be used to edit a file. Vim is a text editor.
21. `'chmod'`: used to change file permissions (read/write/executable).
Eg: `chmod 777` – read, write, execute for all

22. 'ping host' – ping host and output results. Press 'CTRL+c' to exit ping.



```
myDirectory — -bash — 70x22
Vijays-MacBook-Pro:myDirectory vijayravi$ ping www.google.com
PING www.google.com (216.58.195.228): 56 data bytes
64 bytes from 216.58.195.228: icmp_seq=0 ttl=52 time=14.618 ms
64 bytes from 216.58.195.228: icmp_seq=1 ttl=52 time=13.108 ms
64 bytes from 216.58.195.228: icmp_seq=2 ttl=52 time=13.367 ms
64 bytes from 216.58.195.228: icmp_seq=3 ttl=52 time=13.687 ms
64 bytes from 216.58.195.228: icmp_seq=4 ttl=52 time=13.536 ms
64 bytes from 216.58.195.228: icmp_seq=5 ttl=52 time=14.043 ms
64 bytes from 216.58.195.228: icmp_seq=6 ttl=52 time=12.640 ms
^C
--- www.google.com ping statistics ---
7 packets transmitted, 7 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 12.640/13.571/14.618/0.592 ms
Vijays-MacBook-Pro:myDirectory vijayravi$
```

23. 'ssh user@host' – connect to host as user.



```
vijayravi — ssh spaceman@hoffman2.idre.ucla.edu — 80x24
Last login: Wed Nov 29 14:55:20 on ttys000
[Vijays-MacBook-Pro:~ vijayravi$ ssh spaceman@hoffman2.idre.ucla.edu
spaceman@hoffman2.idre.ucla.edu's password: ]
```

- 24. Ctrl+C – halts the current command
- 25. Ctrl+Z – stops the current command, resume with fg in the foreground or bg in the background
- 26. Ctrl+D – logout of current session, similar to exit
- 27. Ctrl+W – erases one word in the current line
- 28. Ctrl+U – erases the whole line
- 29. Ctrl+R – type to bring up a recent command !! - repeats the last command
- 30. exit – log out of current session

STATS 202A: STATISTICS PROGRAMMING

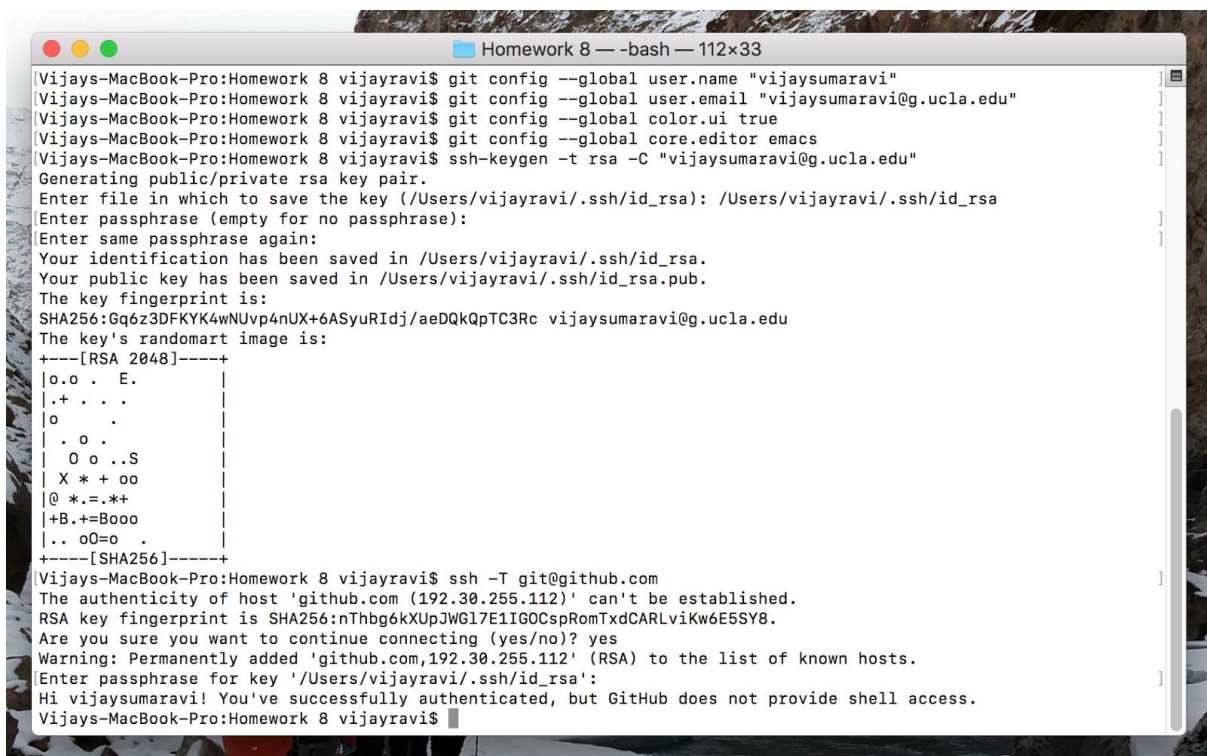
HW8: GITHUB COMMANDS

NAME: VIJAY RAVI

UID: 805033666

STEP 1: Create an account on <https://github.com/>. Account created: [vijaysumaravi](#)

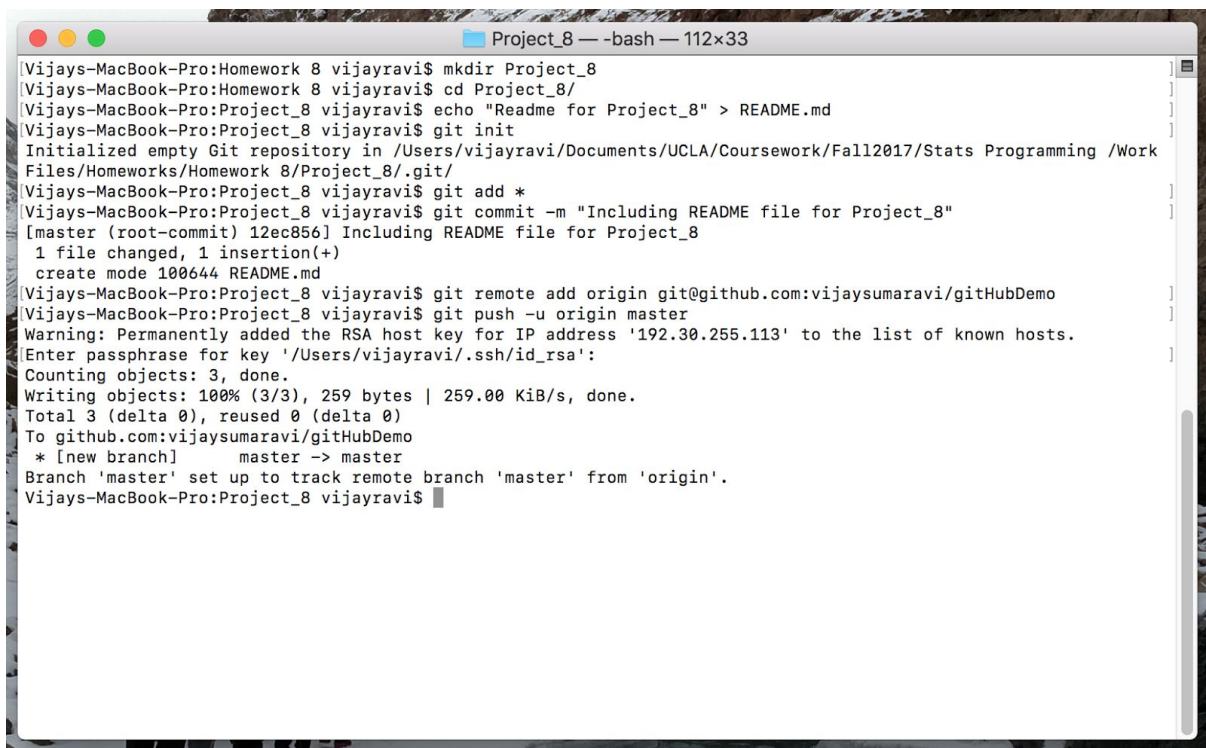
STEP 2: Use the following commands as in the screenshot to set up GITHUB on your system.



```
Vijays-MacBook-Pro:Homework 8 vijayravi$ git config --global user.name "vijaysumaravi"
Vijays-MacBook-Pro:Homework 8 vijayravi$ git config --global user.email "vijaysumaravi@ucla.edu"
Vijays-MacBook-Pro:Homework 8 vijayravi$ git config --global color.ui true
Vijays-MacBook-Pro:Homework 8 vijayravi$ git config --global core.editor emacs
Vijays-MacBook-Pro:Homework 8 vijayravi$ ssh-keygen -t rsa -C "vijaysumaravi@ucla.edu"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/vijayravi/.ssh/id_rsa): /Users/vijayravi/.ssh/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/vijayravi/.ssh/id_rsa.
Your public key has been saved in /Users/vijayravi/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Gq6z3DFKYK4wNUvp4nUX+6ASyurIdj/aeDQkQpTC3Rc vijaysumaravi@ucla.edu
The key's randomart image is:
+---[RSA 2048]---+
|o.o . E.      |
|. + . . .     |
|o .           |
| . o .        |
| O o ..S      |
| X * + oo     |
|@ *. =.*+     |
|+B. +=Boooo   |
|.. oO=o .     |
+---[SHA256]-----+
Vijays-MacBook-Pro:Homework 8 vijayravi$ ssh -T git@github.com
The authenticity of host 'github.com (192.30.255.112)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.255.112' (RSA) to the list of known hosts.
Enter passphrase for key '/Users/vijayravi/.ssh/id_rsa':
Hi vijaysumaravi! You've successfully authenticated, but GitHub does not provide shell access.
Vijays-MacBook-Pro:Homework 8 vijayravi$
```


STEP 3: Creating a new Repository from scratch

- Create a new repository in <https://github.com/>
- I have named the repo as "gitHubDemo"
- Follow the following steps in the screenshot to setup the new project, "Project_8" to the repository.



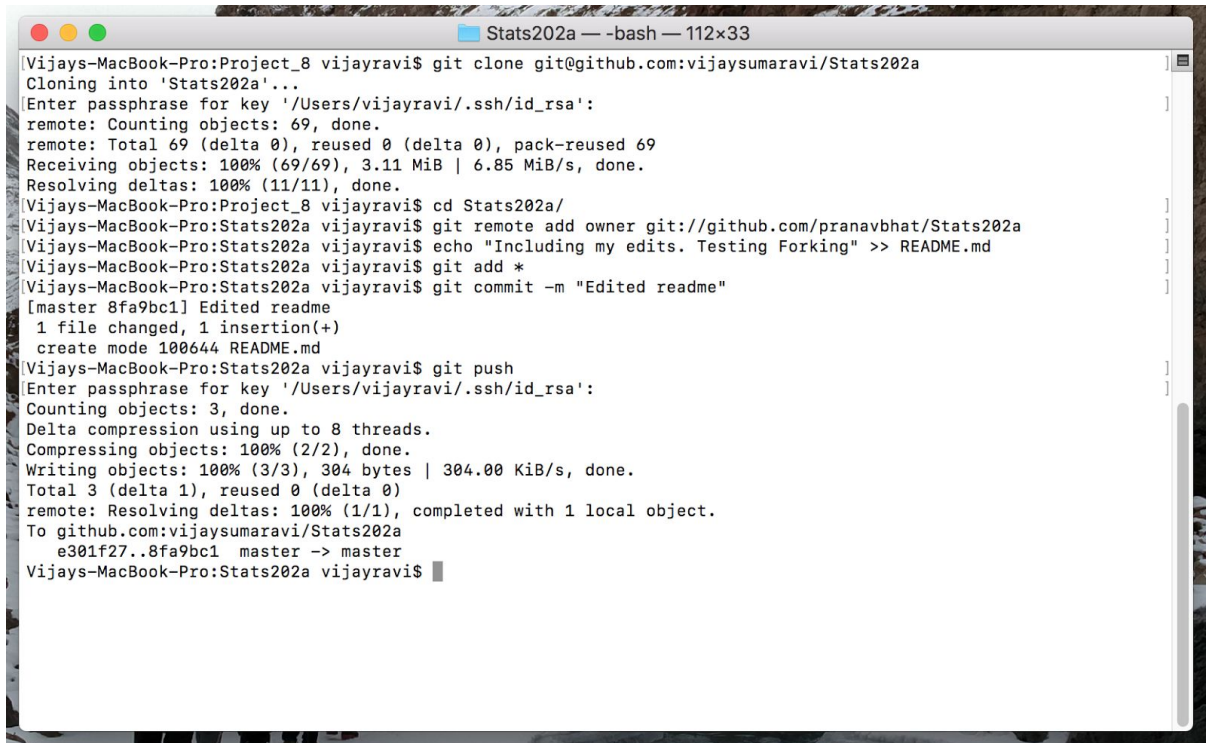
```
Vijays-MacBook-Pro:Homework 8 vijayravi$ mkdir Project_8
Vijays-MacBook-Pro:Homework 8 vijayravi$ cd Project_8/
Vijays-MacBook-Pro:Project_8 vijayravi$ echo "Readme for Project_8" > README.md
Vijays-MacBook-Pro:Project_8 vijayravi$ git init
Initialized empty Git repository in /Users/vijayravi/Documents/UCLA/Coursework/Fall12017/Stats Programming /Work
Files/Homeworks/Homework 8/Project_8/.git/
Vijays-MacBook-Pro:Project_8 vijayravi$ git add *
Vijays-MacBook-Pro:Project_8 vijayravi$ git commit -m "Including README file for Project_8"
[master (root-commit) 12ec856] Including README file for Project_8
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
Vijays-MacBook-Pro:Project_8 vijayravi$ git remote add origin git@github.com:vijaysumaravi/gitHubDemo
Vijays-MacBook-Pro:Project_8 vijayravi$ git push -u origin master
Warning: Permanently added the RSA host key for IP address '192.30.255.113' to the list of known hosts.
Enter passphrase for key '/Users/vijayravi/.ssh/id_rsa':
Counting objects: 3, done.
Writing objects: 100% (3/3), 259 bytes | 259.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:vijaysumaravi/gitHubDemo
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
Vijays-MacBook-Pro:Project_8 vijayravi$
```

STEP 4: Continuing development and keeping the repository in sync with local development environment.

- Once a piece of developed code is tested and stable follow the below mentioned steps to push the changes to the repository.
- git add filename.extension
- git commit -m "Pushing new feature X"
- git push

STEP 5: Forking someone else's repository.

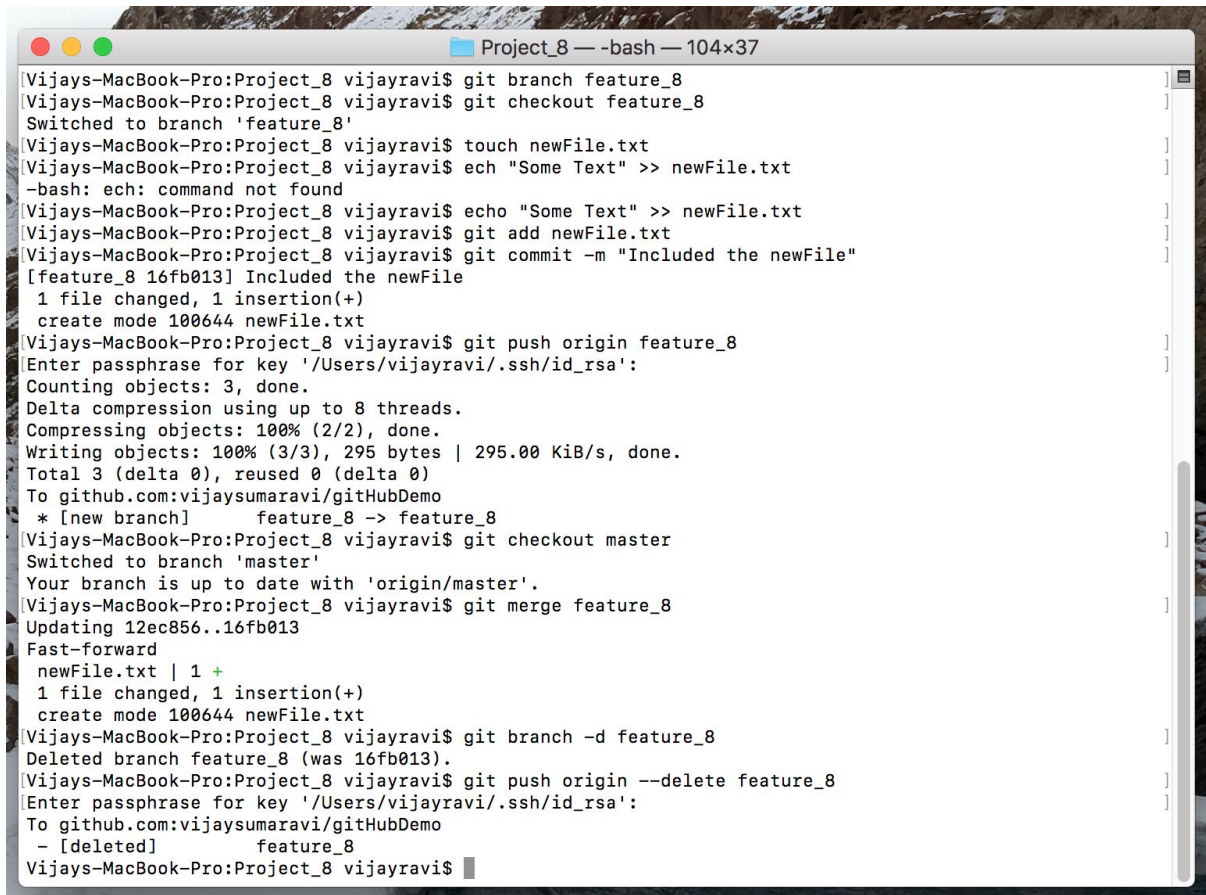
- Fork the repository on <https://github.com>
- Clone the repository to the local system and continue development, push changes.
- I have forked "Stats202a" from pranavbhat and changed the README.md file.



```
Stats202a — -bash — 112x33
Vijays-MacBook-Pro:Project_8 vijayravi$ git clone git@github.com:vijaysumaravi/Stats202a
Cloning into 'Stats202a'...
Enter passphrase for key '/Users/vijayravi/.ssh/id_rsa':
remote: Counting objects: 69, done.
remote: Total 69 (delta 0), reused 0 (delta 0), pack-reused 69
Receiving objects: 100% (69/69), 3.11 MiB | 6.85 MiB/s, done.
Resolving deltas: 100% (11/11), done.
Vijays-MacBook-Pro:Project_8 vijayravi$ cd Stats202a/
Vijays-MacBook-Pro:Stats202a vijayravi$ git remote add owner git://github.com/pranavbhat/Stats202a
Vijays-MacBook-Pro:Stats202a vijayravi$ echo "Including my edits. Testing Forking" >> README.md
Vijays-MacBook-Pro:Stats202a vijayravi$ git add *
Vijays-MacBook-Pro:Stats202a vijayravi$ git commit -m "Edited readme"
[master 8fa9bc1] Edited readme
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
Vijays-MacBook-Pro:Stats202a vijayravi$ git push
Enter passphrase for key '/Users/vijayravi/.ssh/id_rsa':
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 304 bytes | 304.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:vijaysumaravi/Stats202a
 e301f27..8fa9bc1 master -> master
Vijays-MacBook-Pro:Stats202a vijayravi$
```

STEP 6: Branching

- Creating a new feature branch.
- Developing and Testing in the branch.
- Merging development in the new branch on to the master branch.
- Deleting the new feature

A terminal window titled "Project_8 — -bash — 104x37" showing a series of Git commands and their outputs. The user creates a new branch 'feature_8', checks it out, creates a new file 'newFile.txt', adds it to the commit, and pushes it to the origin. Then, they checkout the 'master' branch, merge the 'feature_8' branch, and finally delete the 'feature_8' branch. The terminal output shows the progress of each step, including file changes, commit messages, and push status.

```
Vijays-MacBook-Pro:Project_8 vijayravi$ git branch feature_8
Vijays-MacBook-Pro:Project_8 vijayravi$ git checkout feature_8
Switched to branch 'feature_8'
Vijays-MacBook-Pro:Project_8 vijayravi$ touch newFile.txt
Vijays-MacBook-Pro:Project_8 vijayravi$ echo "Some Text" >> newFile.txt
-bash: ech: command not found
Vijays-MacBook-Pro:Project_8 vijayravi$ echo "Some Text" >> newFile.txt
Vijays-MacBook-Pro:Project_8 vijayravi$ git add newFile.txt
Vijays-MacBook-Pro:Project_8 vijayravi$ git commit -m "Included the newFile"
[feature_8 16fb013] Included the newFile
 1 file changed, 1 insertion(+)
 create mode 100644 newFile.txt
Vijays-MacBook-Pro:Project_8 vijayravi$ git push origin feature_8
Enter passphrase for key '/Users/vijayravi/.ssh/id_rsa':
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 295 bytes | 295.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:vijaysumaravi/gitHubDemo
 * [new branch]      feature_8 -> feature_8
Vijays-MacBook-Pro:Project_8 vijayravi$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Vijays-MacBook-Pro:Project_8 vijayravi$ git merge feature_8
Updating 12ec856..16fb013
Fast-forward
 newFile.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 newFile.txt
Vijays-MacBook-Pro:Project_8 vijayravi$ git branch -d feature_8
Deleted branch feature_8 (was 16fb013).
Vijays-MacBook-Pro:Project_8 vijayravi$ git push origin --delete feature_8
Enter passphrase for key '/Users/vijayravi/.ssh/id_rsa':
To github.com:vijaysumaravi/gitHubDemo
 - [deleted]      feature_8
Vijays-MacBook-Pro:Project_8 vijayravi$
```