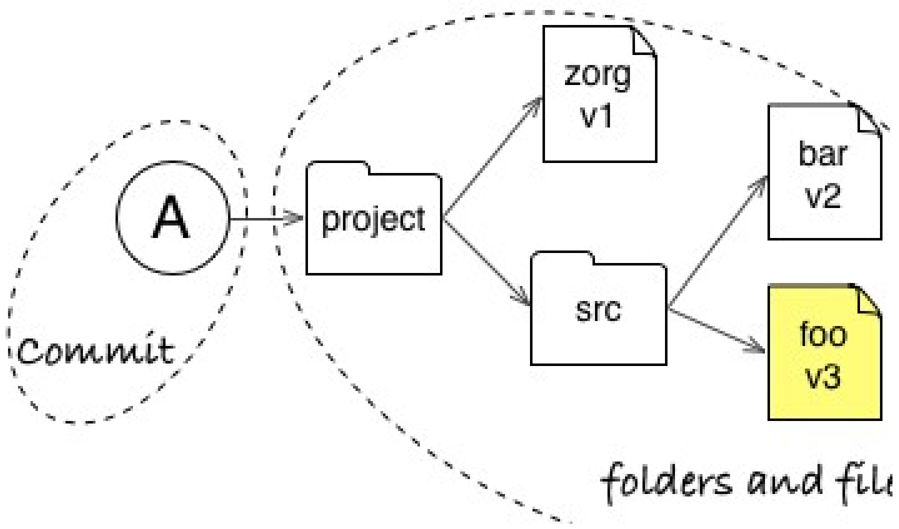
Core concepts in Git

Module 7-2

Core Concepts

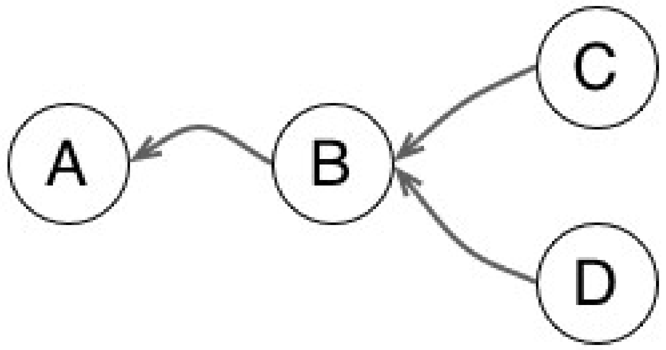
Git stores snapshots (commits) of your repository



http://sogilis.com/blog/demystifying-git-concepts-to-understand/

Core Concepts

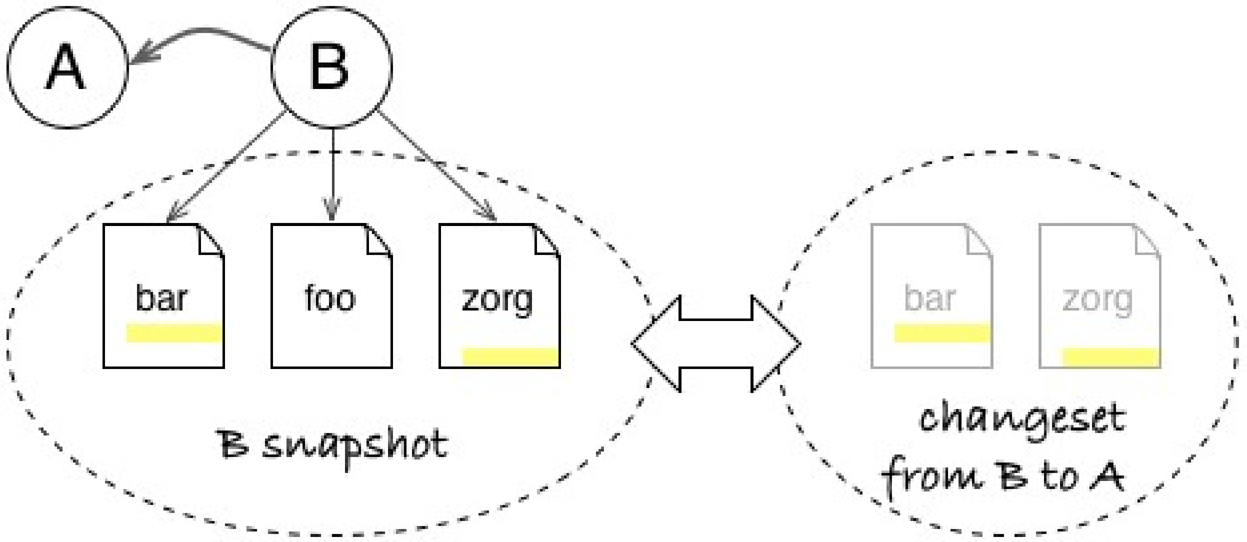
Git represents relationships between commits as a graph



http://sogilis.com/blog/demystifying-git-concepts-to-understand/

Core Concepts

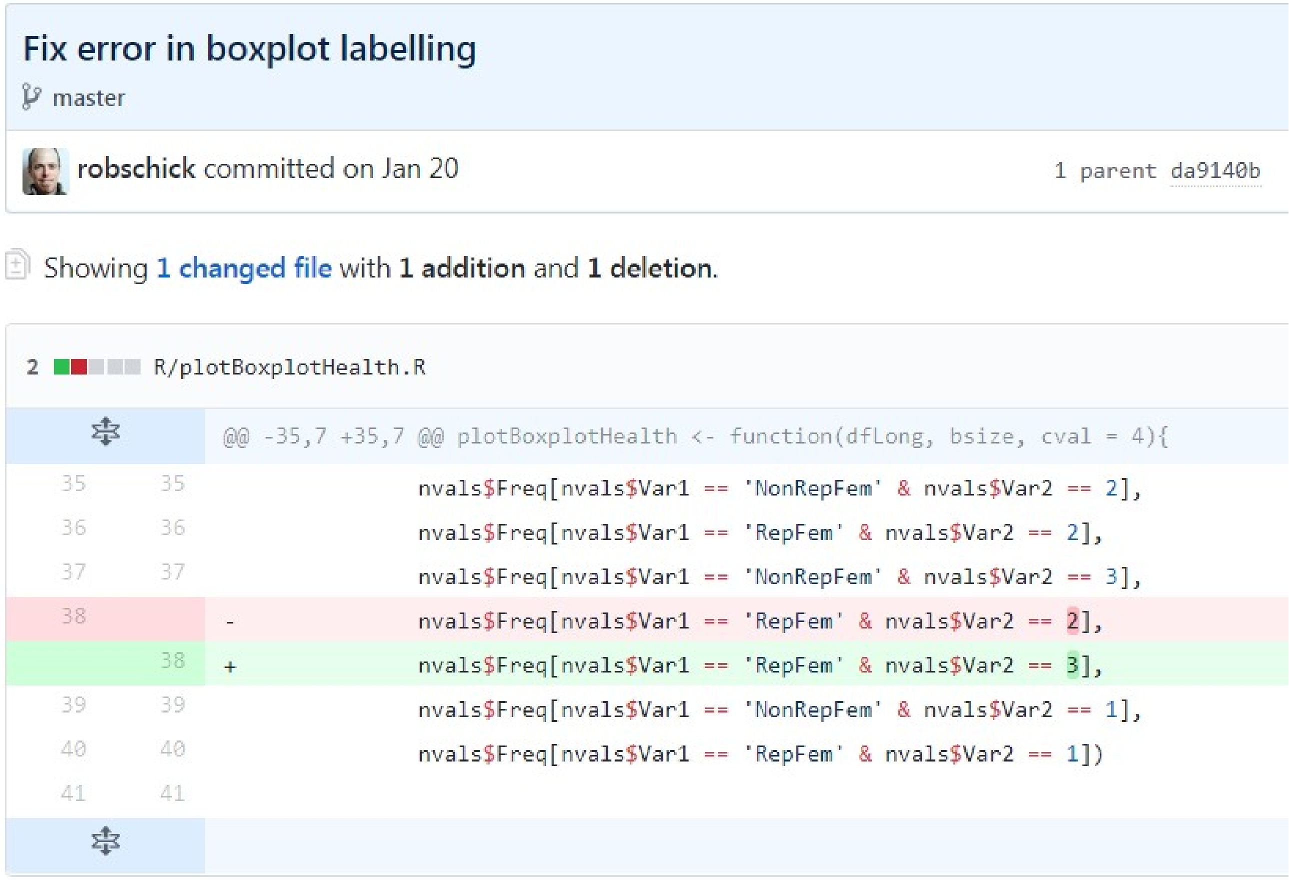
Git can compute changesets between any two commits of your project



http://sogilis.com/blog/demystifying-git-concepts-to-understand/

Core Concepts

Git sees changes at the level of lines in a text file



Fix error in boxplot labelling

master

robschick committed on Jan 20 1 parent da9140b

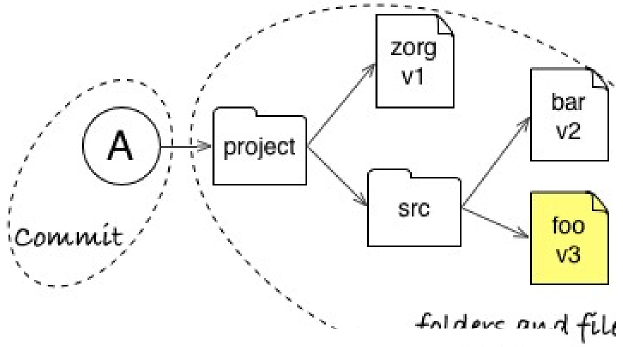
Showing 1 changed file with 1 addition and 1 deletion.

```
2 R/plotBoxplotHealth.R

@@ -35,7 +35,7 @@ plotBoxplotHealth <- function(dfLong, bsize, cval = 4){
 35  35                 nvals$Freq[nvals$Var1 == 'NonRepFem' & nvals$Var2 == 2],
 36  36                 nvals$Freq[nvals$Var1 == 'RepFem' & nvals$Var2 == 2],
 37  37                 nvals$Freq[nvals$Var1 == 'NonRepFem' & nvals$Var2 == 3],
 38      -              nvals$Freq[nvals$Var1 == 'RepFem' & nvals$Var2 == 2],
     38  +              nvals$Freq[nvals$Var1 == 'RepFem' & nvals$Var2 == 3],
 39  39                 nvals$Freq[nvals$Var1 == 'NonRepFem' & nvals$Var2 == 1],
 40  40                 nvals$Freq[nvals$Var1 == 'RepFem' & nvals$Var2 == 1])
 41  41
```

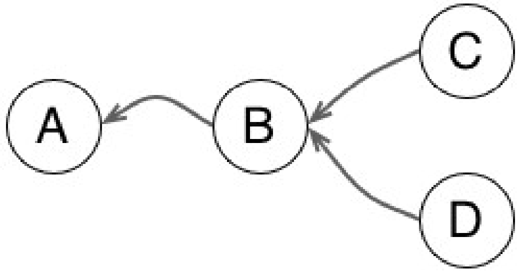
http://sogilis.com/blog/demystifying-git-concepts-to-understand/

3 Core Concepts

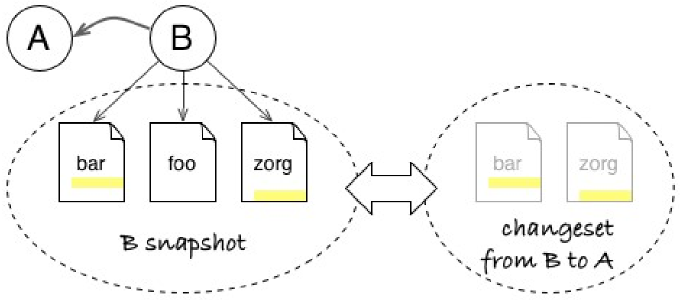
- Snapshot



- Graph



- Changeset

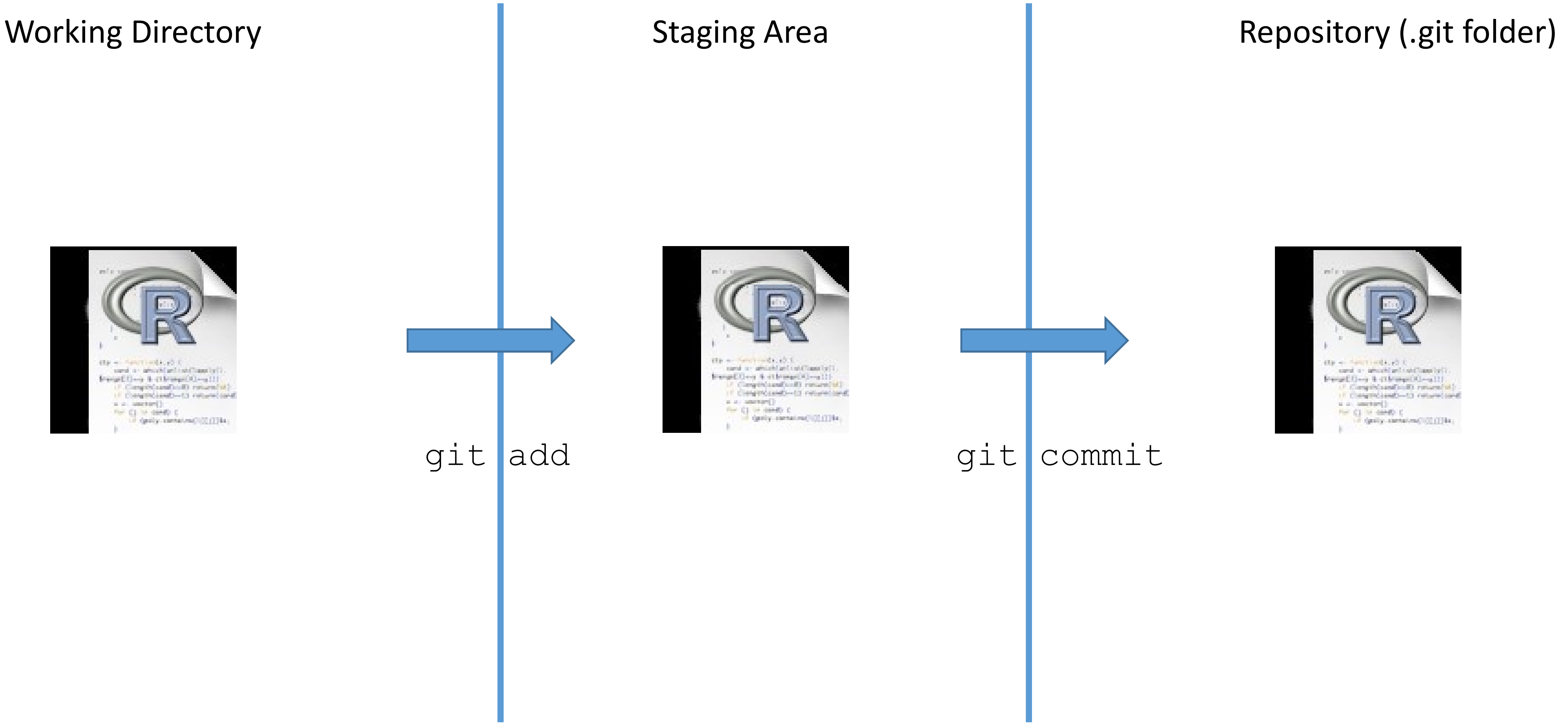


States of a git Repository

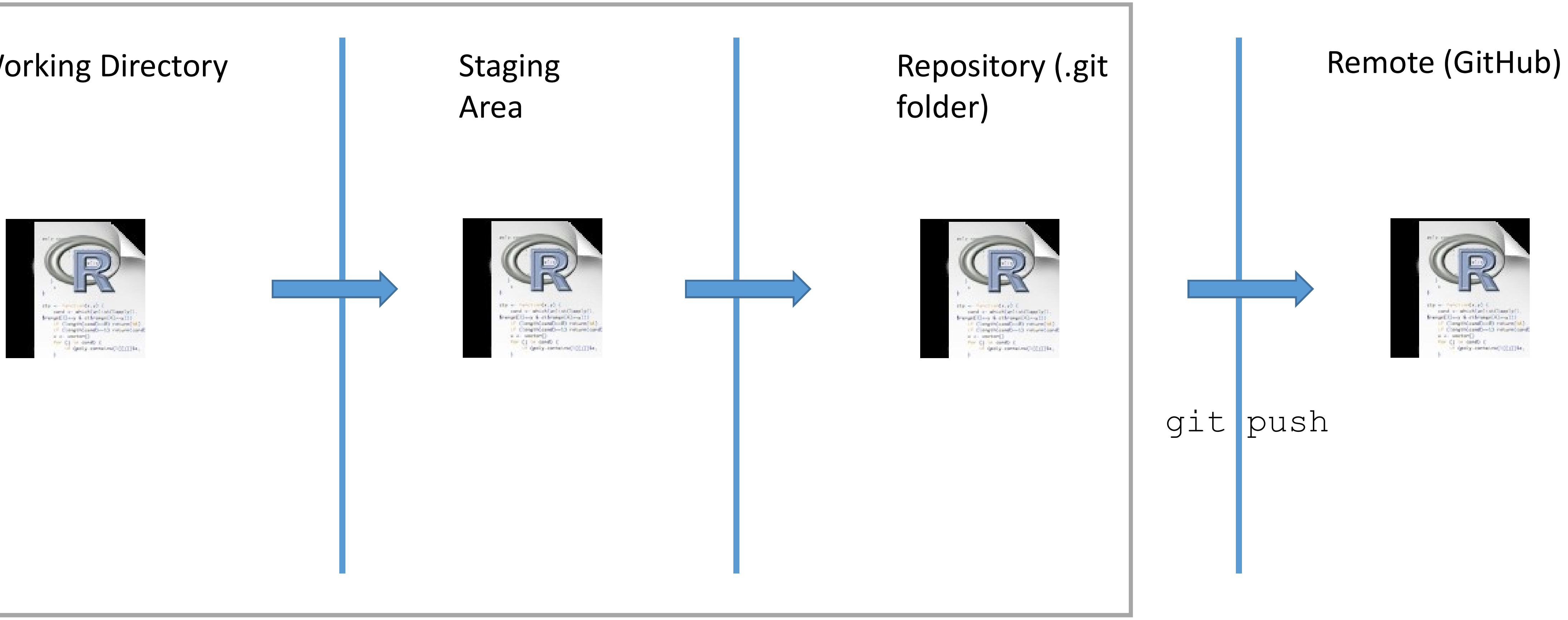
The Repository

- Collection of files managed by git
- History (all of it)
- Encompassing file on the Operating System is considered the working directory
    - Can include files managed by git
    - Files ignored by git
    - Files not yet managed by git
- Quasi-hidden **.git** folder
- Since the repo contains all the history, keep the repos narrowly focused

Three Local States



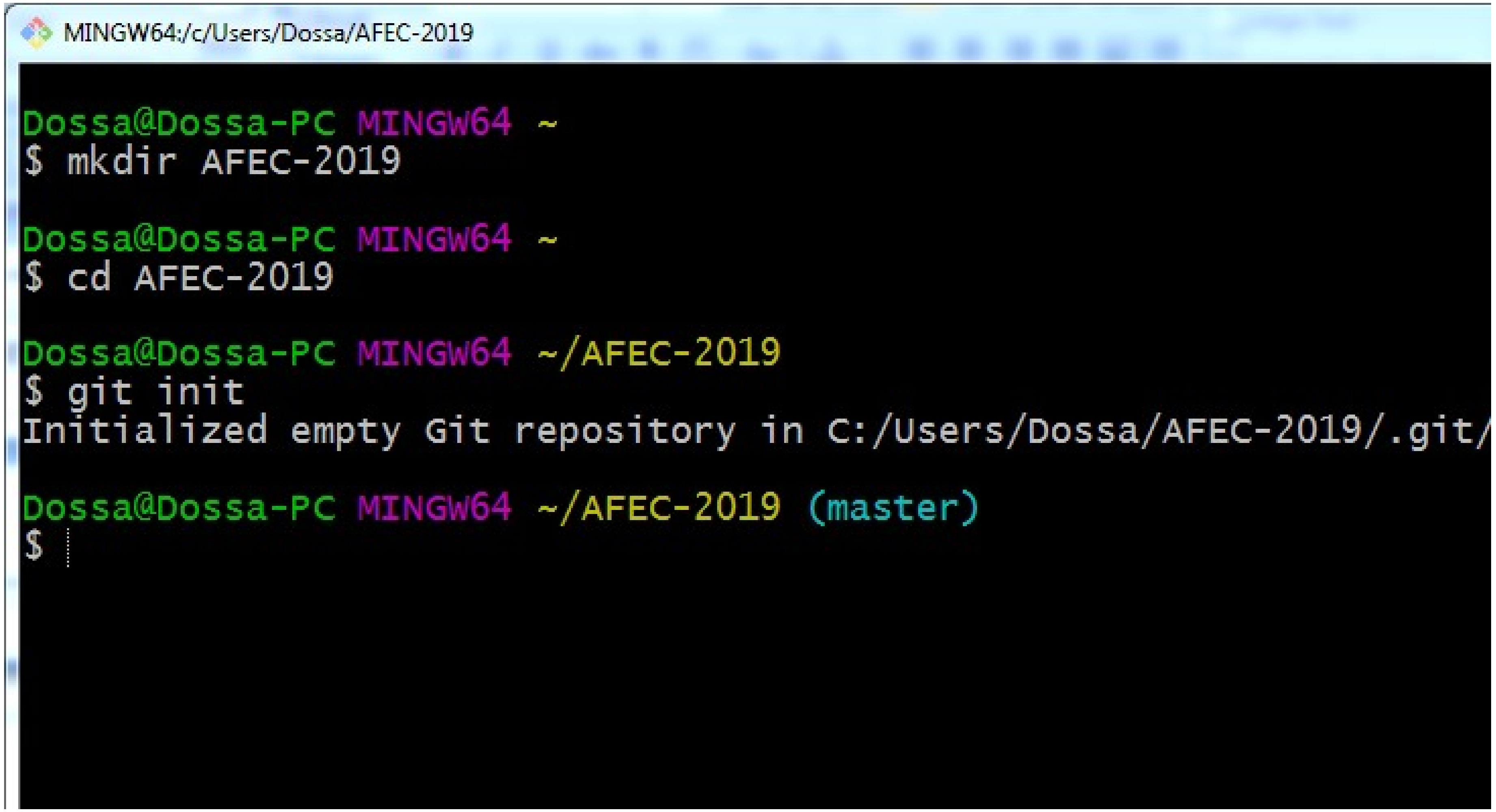
Three Local States with Remote



Basic Commands

Mastering a Basic Workflow

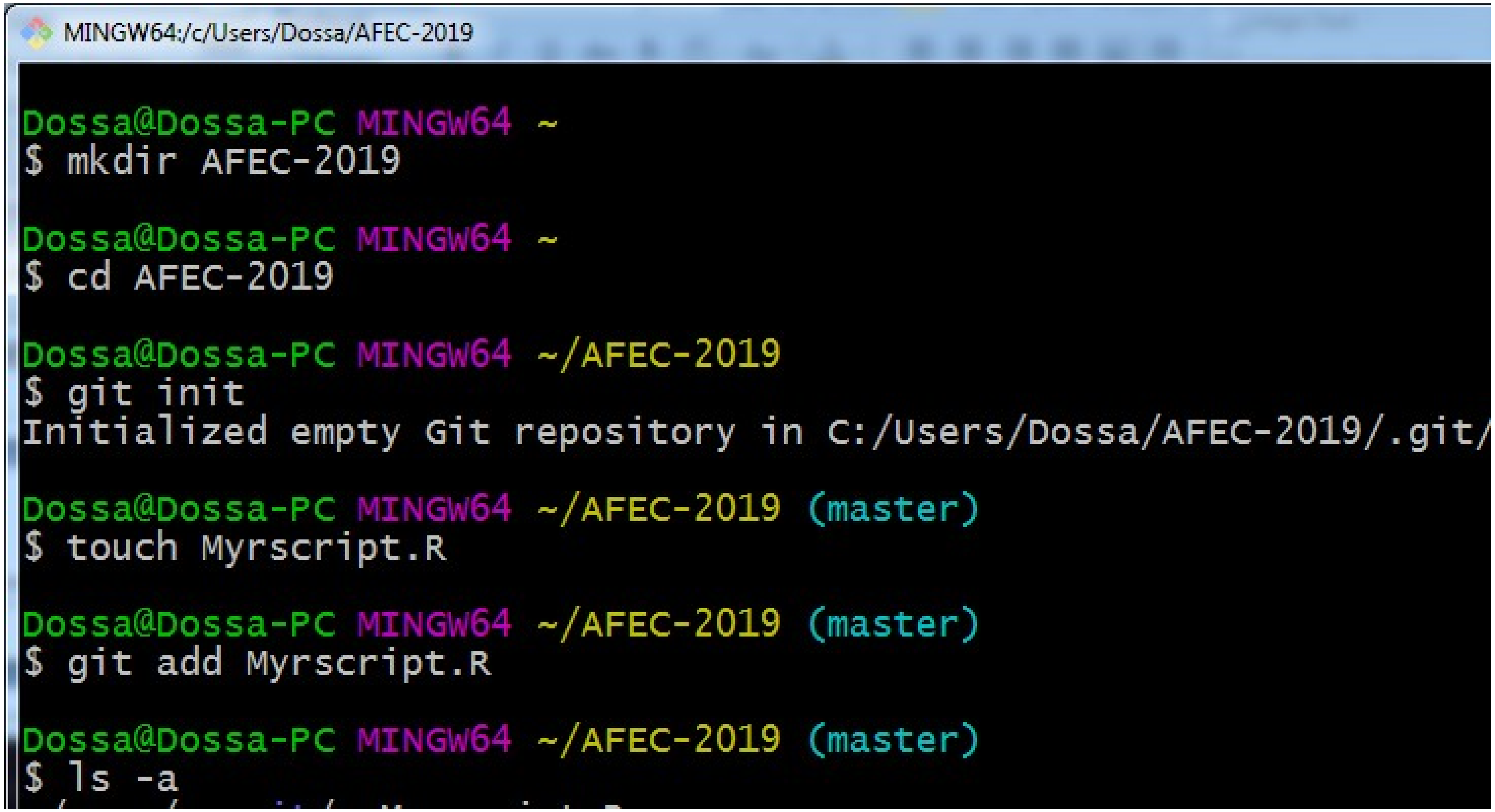
`git init` – Initialize an Empty Repo



MINGW64:/c/Users/Dossa/AFEC-2019

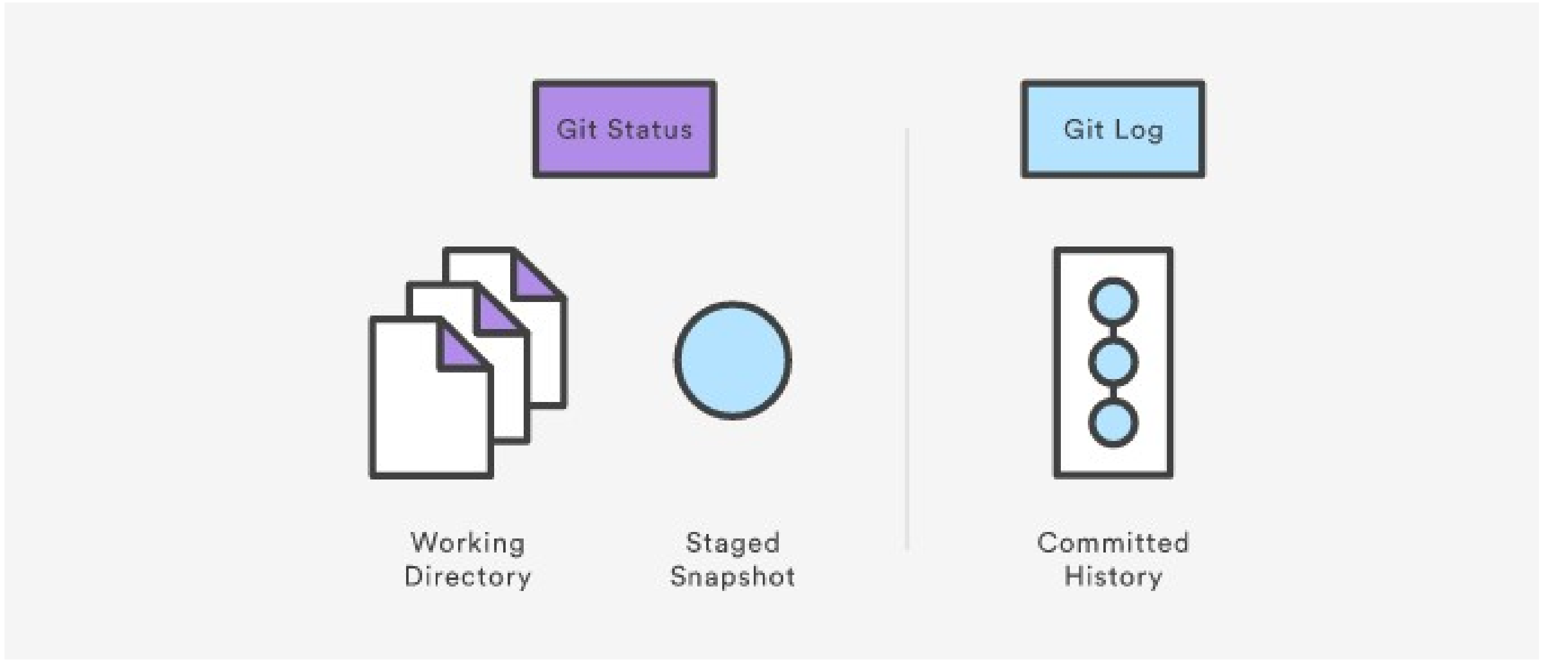
```
Dossa@Dossa-PC MINGW64 ~
$ mkdir AFEC-2019

Dossa@Dossa-PC MINGW64 ~
$ cd AFEC-2019

Dossa@Dossa-PC MINGW64 ~/AFEC-2019
$ git init
Initialized empty Git repository in C:/Users/Dossa/AFEC-2019/.git/

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$
```

`git add` – Add a Document to the Staging Area



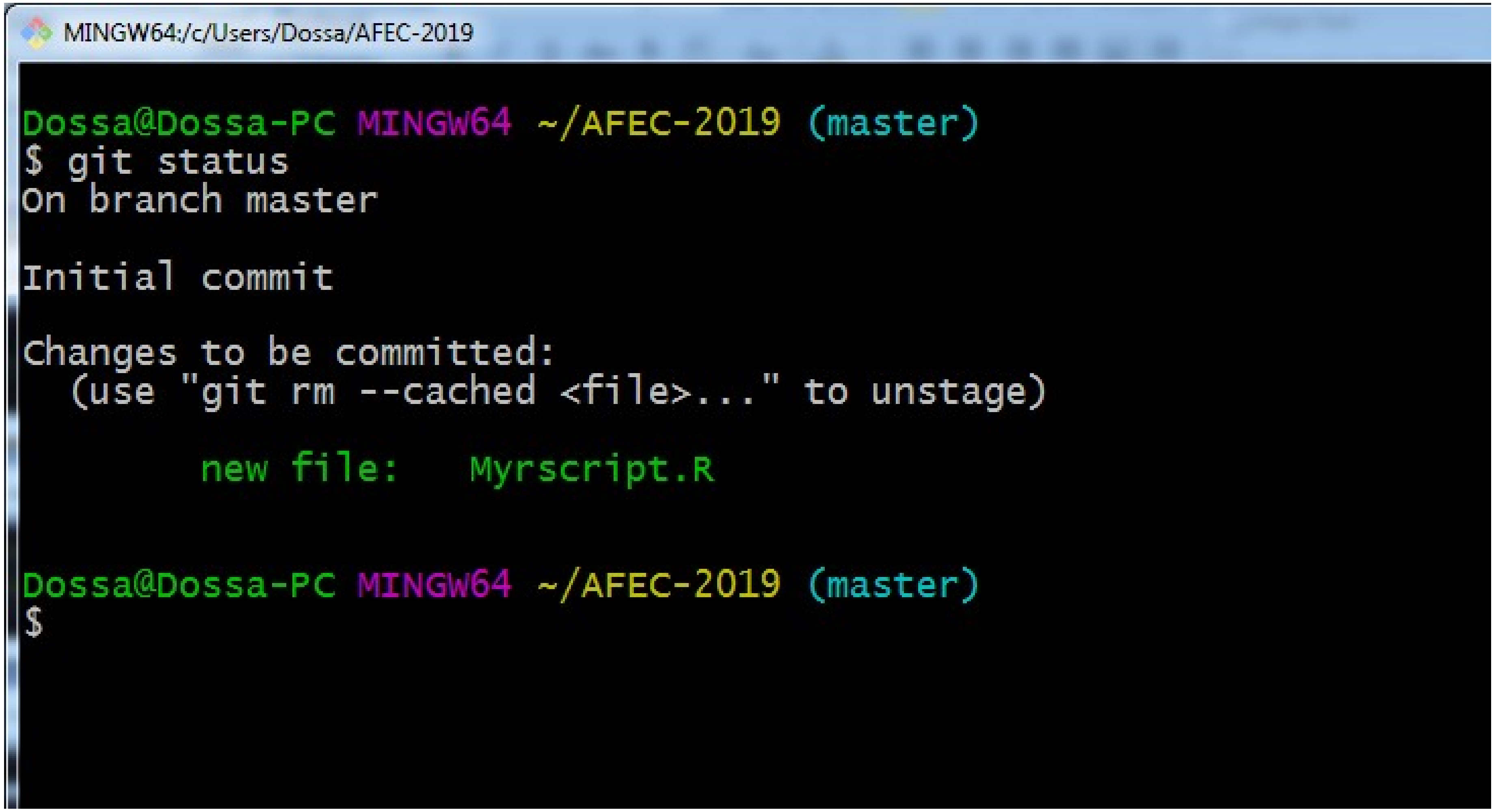
```
Dossa@Dossa-PC MINGW64 ~
$ mkdir AFEC-2019

Dossa@Dossa-PC MINGW64 ~
$ cd AFEC-2019

Dossa@Dossa-PC MINGW64 ~/AFEC-2019
$ git init
Initialized empty Git repository in C:/Users/Dossa/AFEC-2019/.git/

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ touch Myrscript.R

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git add Myrscript.R

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ ls -a
```

Viewing the Repo



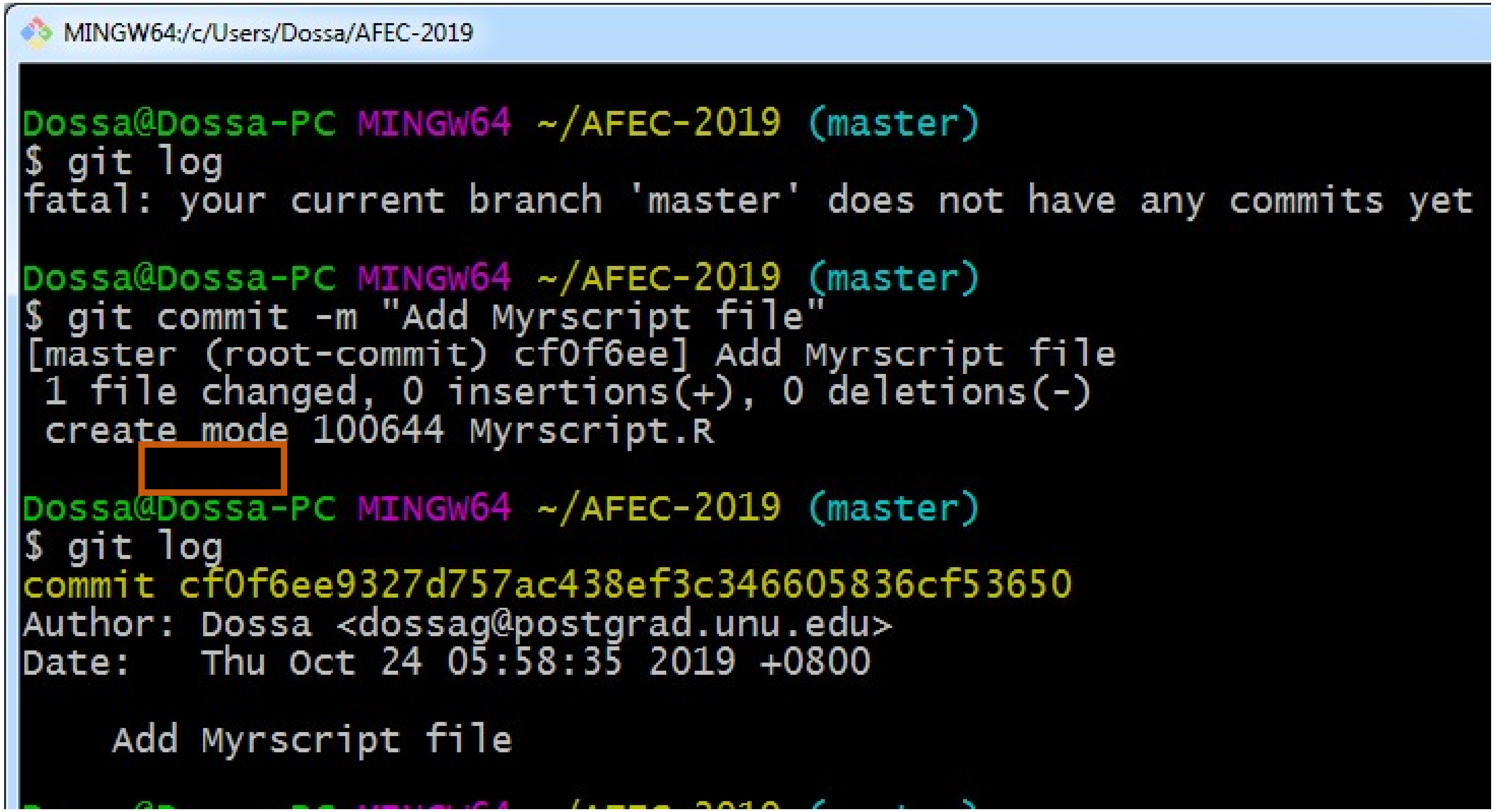
https://www.atlassian.com/git/tutorials/inspecting-a-repository

`git status` – What’s Happening?



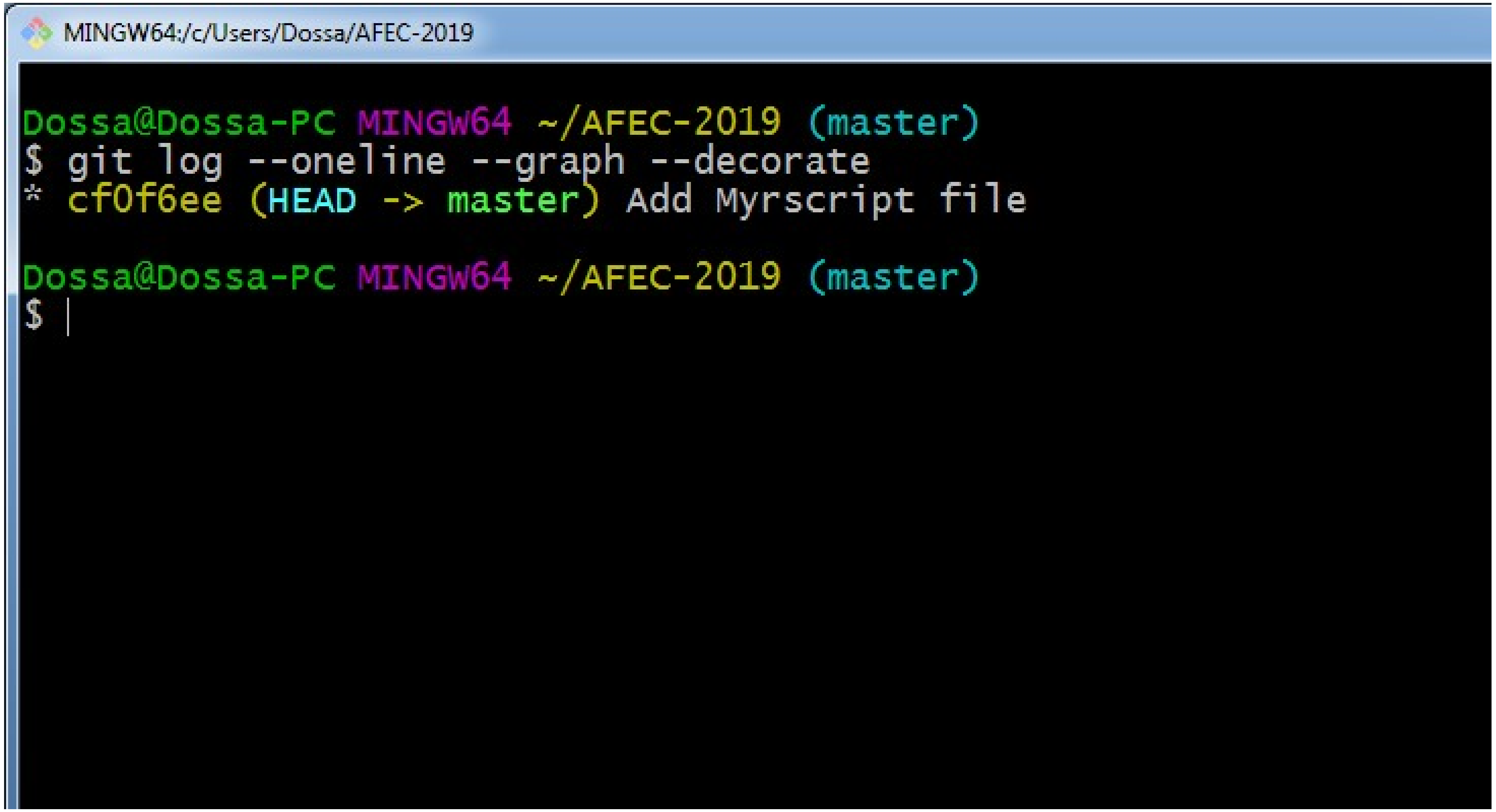
```
Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git status
On branch master

Initial commit

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

        new file:   Myrscript.R


Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$
```

git log – To view the history of Repo



```
Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git commit -m "Add Myrscript file"
[master (root-commit) cf0f6ee] Add Myrscript file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Myrscript.R

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git log
commit cf0f6ee9327d757ac438ef3c346605836cf53650
Author: Dossa <dossag@postgrad.unu.edu>
Date:   Thu Oct 24 05:58:35 2019 +0800

    Add Myrscript file
```

`git log` – With Options

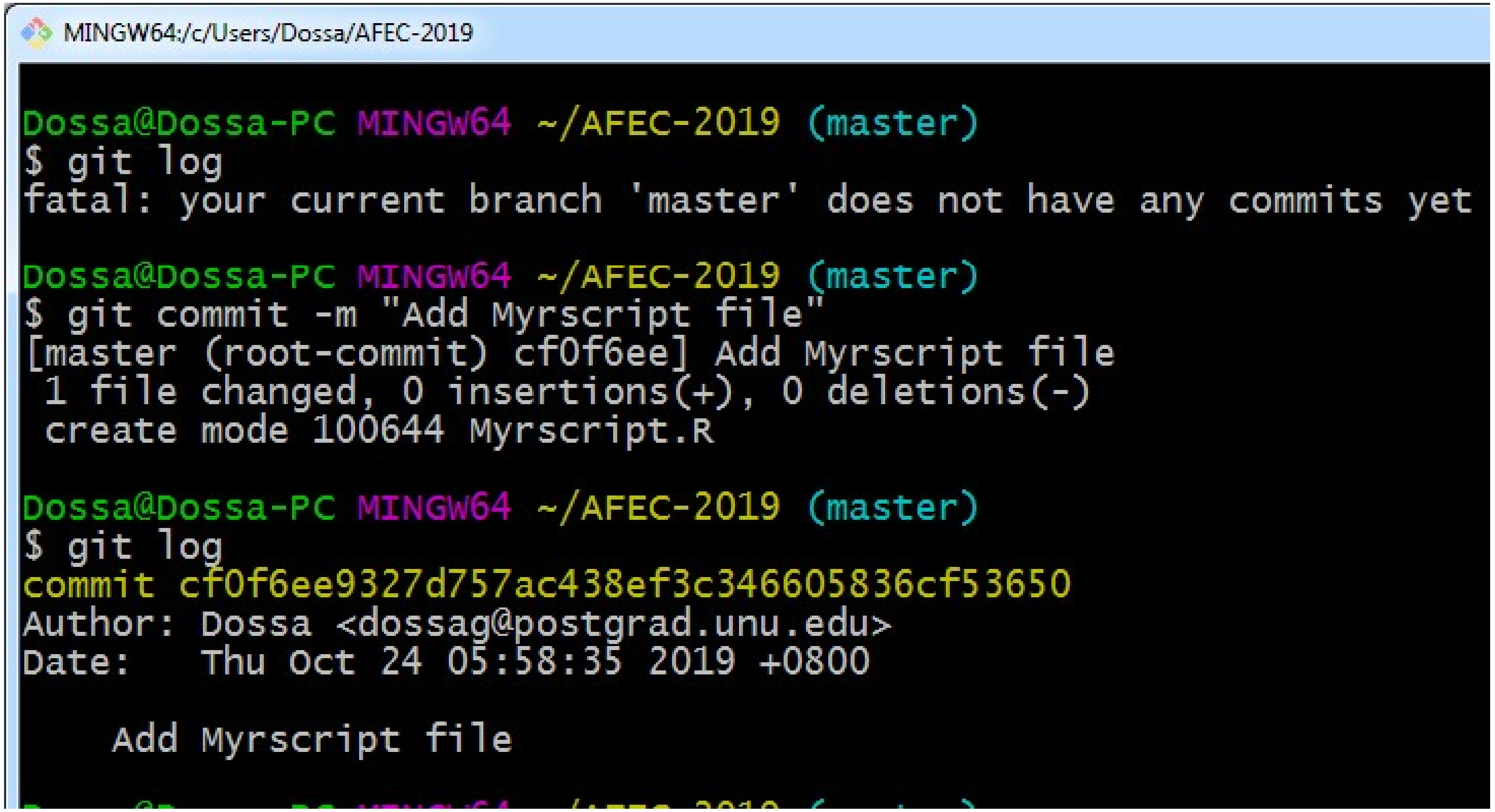


MINGW64:/c/Users/Dossa/AFEC-2019

```
Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git log --oneline --graph --decorate
* cf0f6ee (HEAD -> master) Add Myrscript file

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ |
```

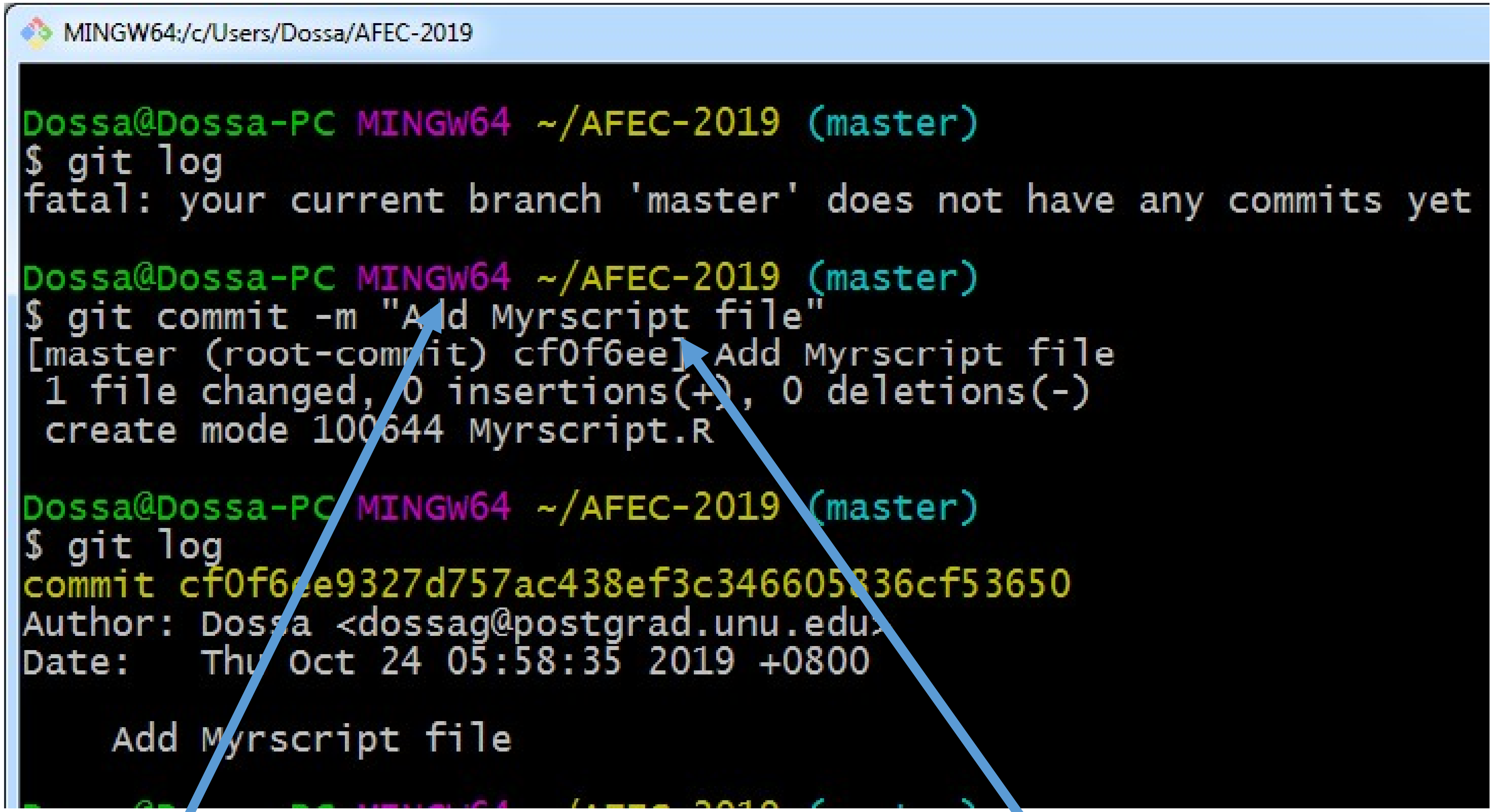
`git commit` – Records changes in the Repo

- `git commit -m “Second check in of my R Script"`



```
Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git commit -m "Add Myrscript file"
[master (root-commit) cf0f6ee] Add Myrscript file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Myrscript.R

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git log
commit cf0f6ee9327d757ac438ef3c346605836cf53650
Author: Dossa <dossag@postgrad.unu.edu>
Date:   Thu Oct 24 05:58:35 2019 +0800

    Add Myrscript file
```

Ok, What Just happened?


MINGW64:/c/Users/Dossa/AFEC-2019

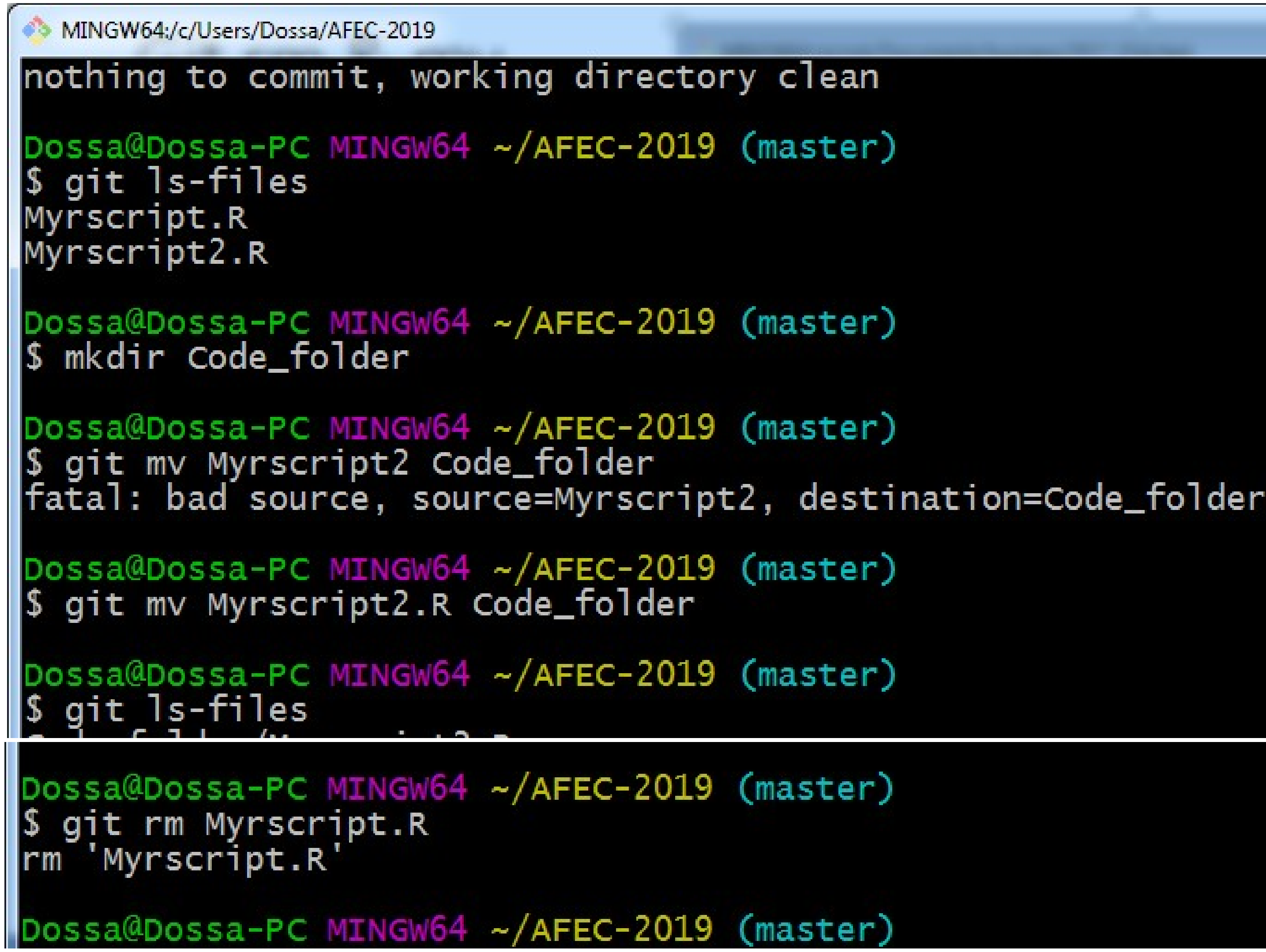
```
Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git commit -m "Add Myrscript file"
[master (root-commit) cf0f6ee] Add Myrscript file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Myrscript.R

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git log
commit cf0f6ee9327d757ac438ef3c346605836cf53650
Author: Dossa <dossag@postgrad.unu.edu>
Date:   Thu Oct 24 05:58:35 2019 +0800

    Add Myrscript file
```

sha-1 commit

What's Changed

Git rm

- Don’t delete or rename tracked files with the OS; use:
- `git rm`
- `git mv`



MINGW64:/c/Users/Dossa/AFEC-2019

```
nothing to commit, working directory clean

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git ls-files
Myrscript.R
Myrscript2.R

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ mkdir Code_folder

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git mv Myrscript2 Code_folder
fatal: bad source, source=Myrscript2, destination=Code_folder

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git mv Myrscript2.R Code_folder

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git ls-files
```

```
Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
$ git rm Myrscript.R
rm 'Myrscript.R'

Dossa@Dossa-PC MINGW64 ~/AFEC-2019 (master)
```

Git mv

- Don’t delete or rename tracked files with the OS; use:

- `git mv`

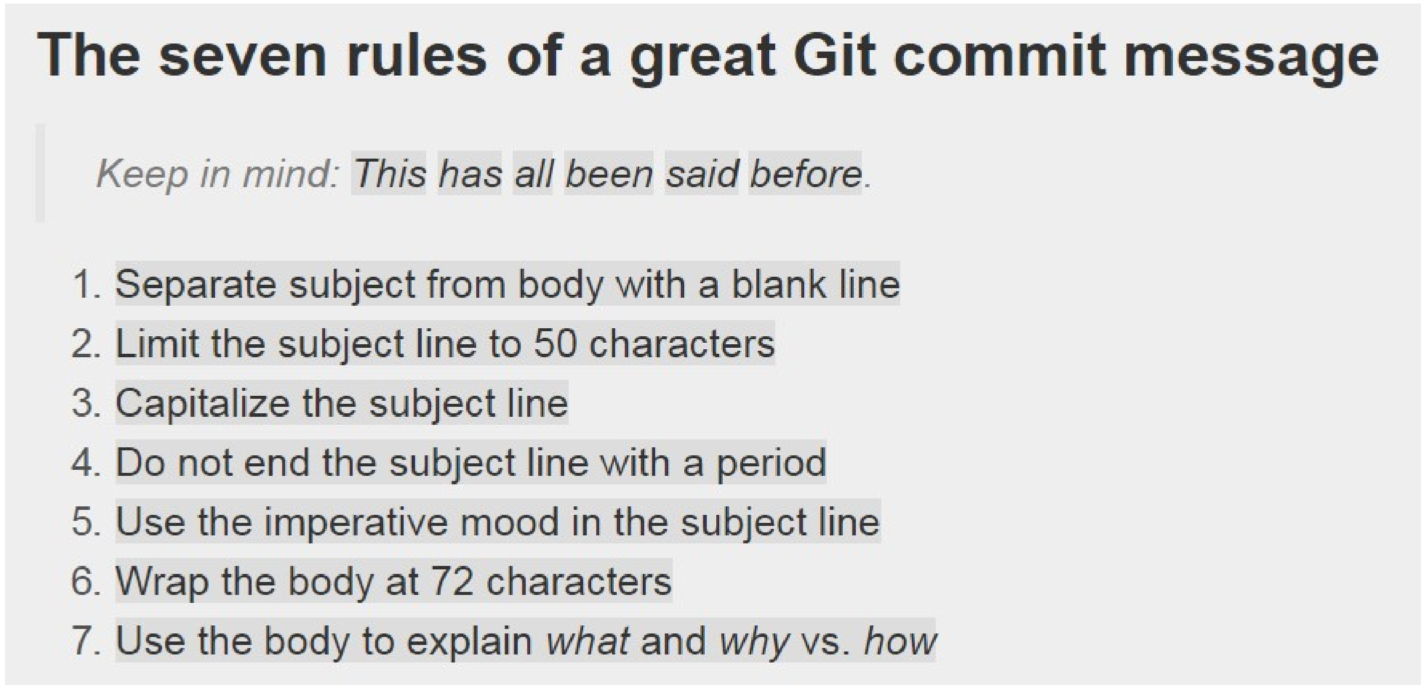
Lather, Rinse, Repeat

Good Commit Messages*

- **Be concise, yet evocative.** At a glance, you should be able to see what a commit does. But there should be enough detail so you can remember (and understand) what was done
- **Describe the why, not the what.** Since you can always retrieve the diff associated with the commit, the message doesn’t need to say exactly what changed. Instead it should provide a high-level summary that focuses on the reasons for the change

*http://r-pkgs.had.co.nz/git.html#commit-best-practices

Good Commit Messages



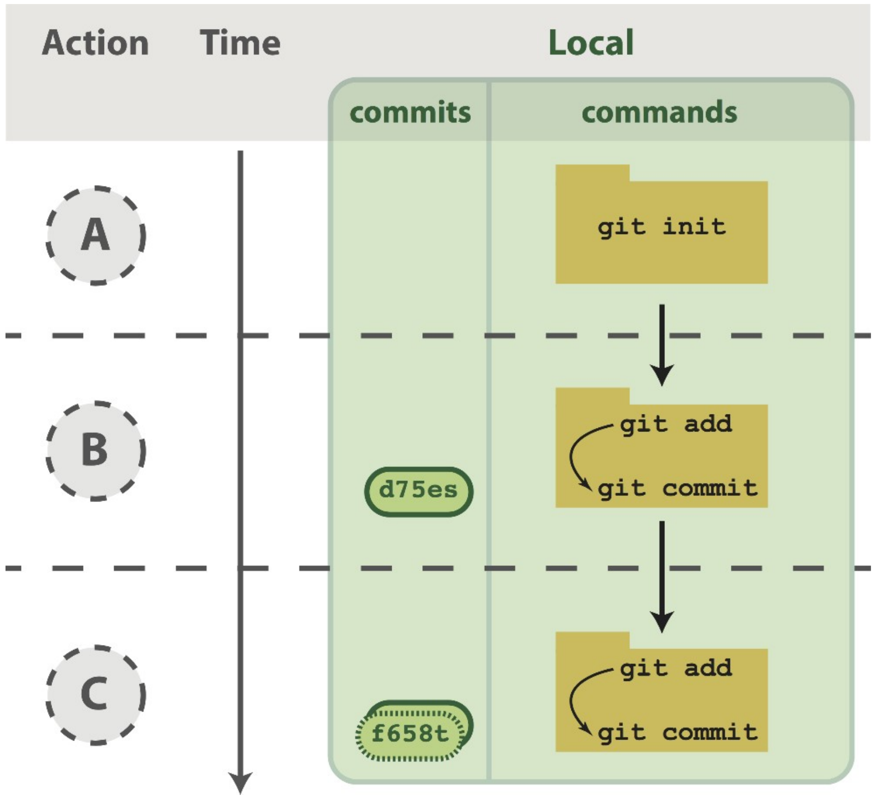
The seven rules of a great Git commit message

> *Keep in mind: This has all been said before.*

1. Separate subject from body with a blank line
2. Limit the subject line to 50 characters
3. Capitalize the subject line
4. Do not end the subject line with a period
5. Use the imperative mood in the subject line
6. Wrap the body at 72 characters
7. Use the body to explain *what* and *why* vs. *how*

https://chris.beams.io/posts/git-commit/

Workflow Visualized



Blischak, John D., Emily R. Davenport, and Greg Wilson. 2016.
"A Quick Introduction to Version Control with Git and GitHub."
PLoS Computational Biology 12 (1): e1004668.

Lifecycle of status

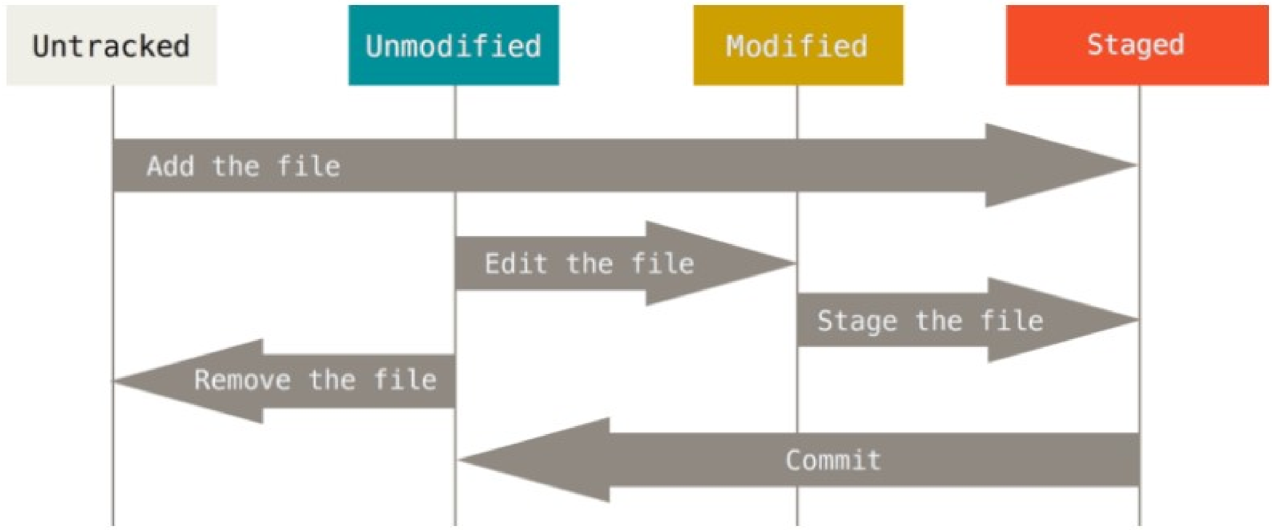
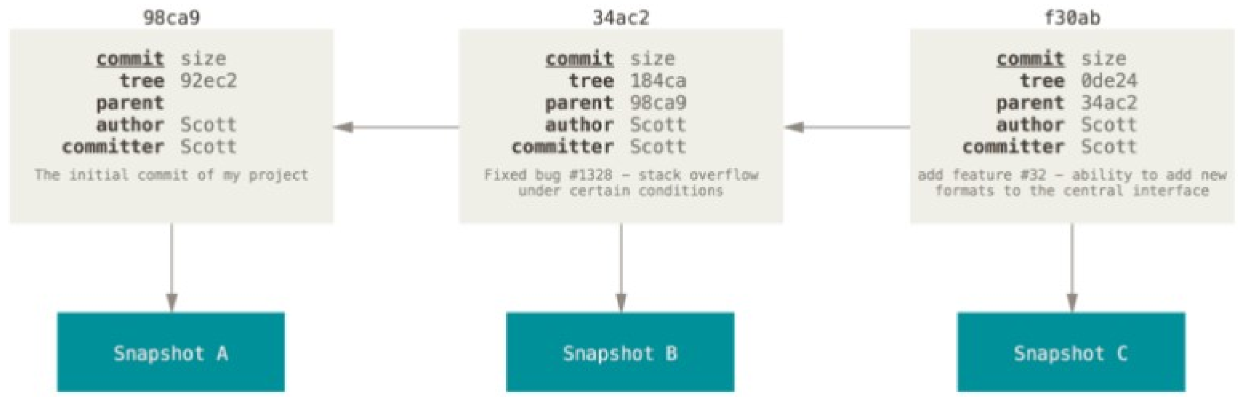


FIGURE 2-1

The lifecycle of the status of your files.

Chacon, S., and B. Straub. 2014. Pro Git. The Expert's Voice. Apress.

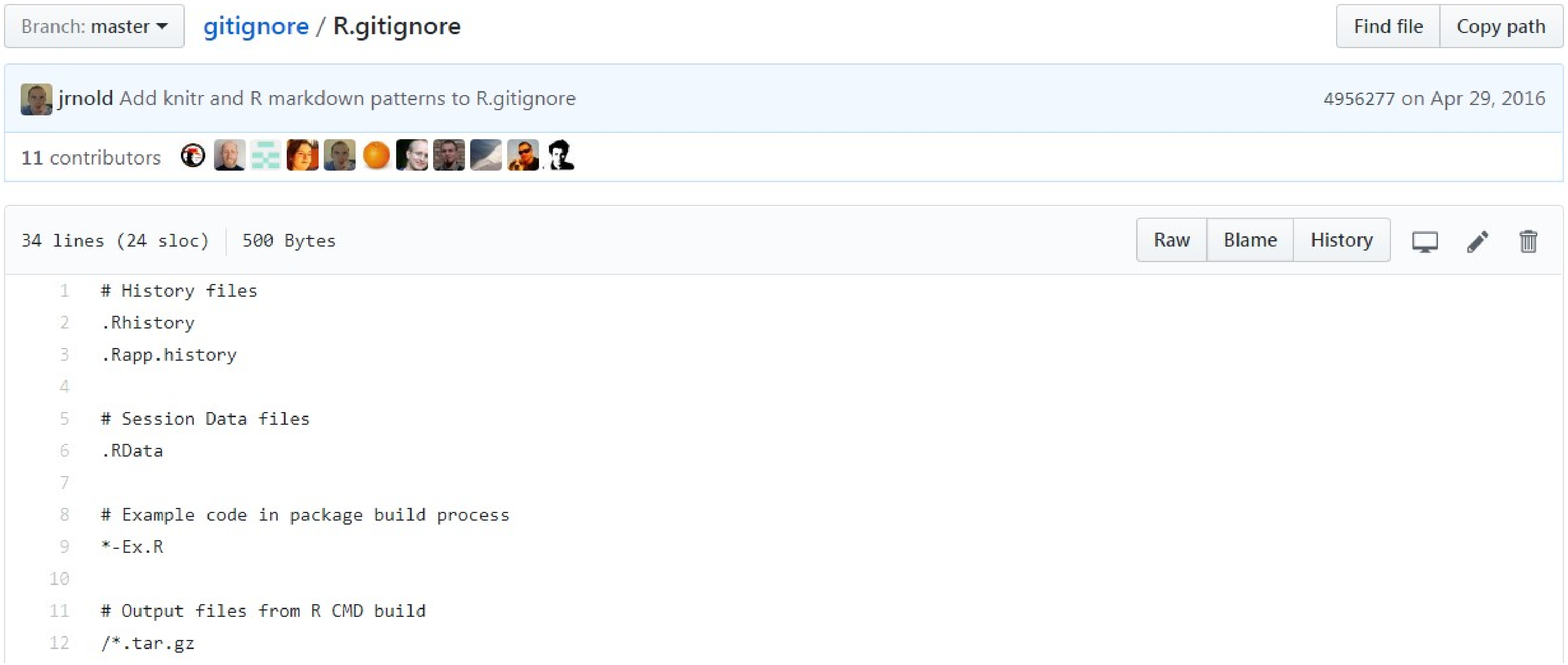
Commit Graph Visualized



A branch in Git is simply a lightweight movable pointer to one of these commits. The default branch name in Git is `master`. As you start making commits, you're given a `master` branch that points to the last commit you made. Every time you commit, it moves forward automatically.

Chacon, S., and B. Straub. 2014. Pro Git. The Expert's Voice. Apress.

.gitignore



jrnold Add knitr and R markdown patterns to R.gitignore 4956277 on Apr 29, 2016

11 contributors

34 lines (24 sloc) | 500 Bytes

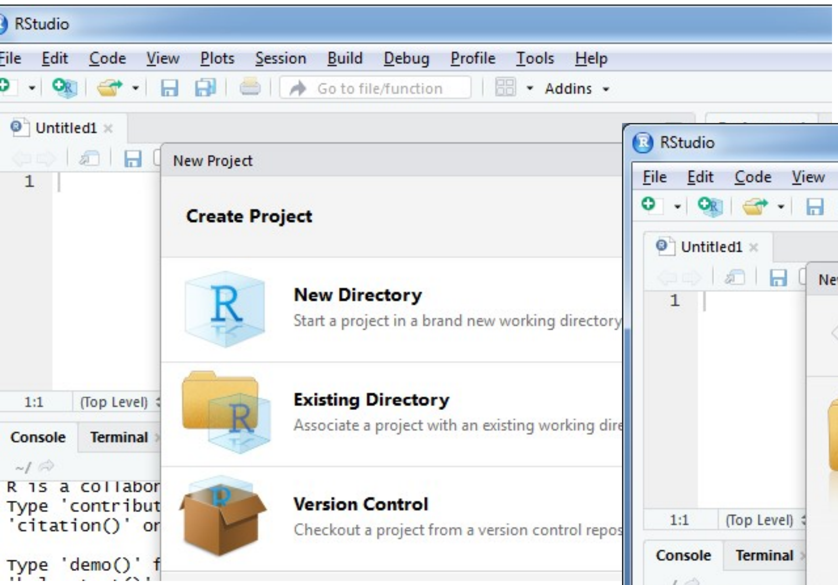
```
# History files
.Rhistory
.Rapp.history

# Session Data files
.RData

# Example code in package build process
*-Ex.R

# Output files from R CMD build
/*.tar.gz
```

`.gitignore`

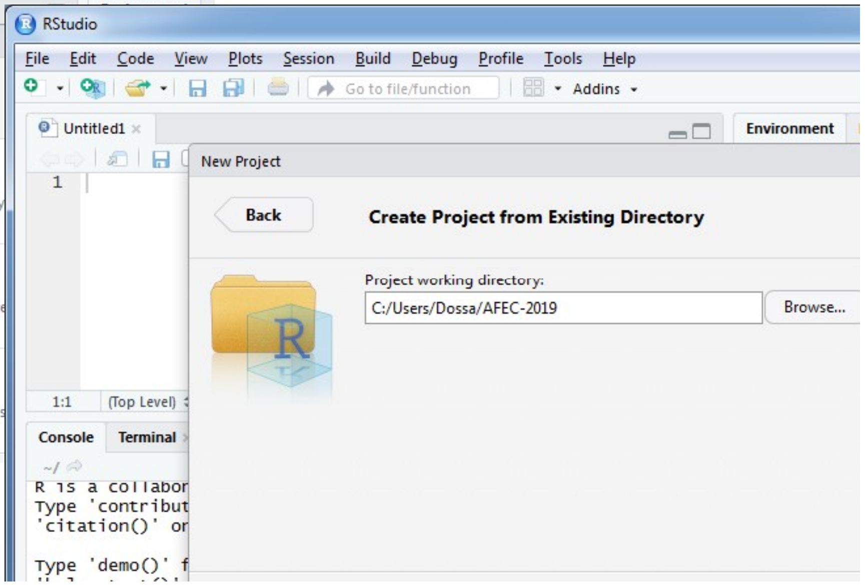
- Make the code produce a plot file, say a pdf
- Run the code
- Make a `.gitgnore` file
  - `.pdf`
  - `.Rout`
- Add & commit the `.gitignore`
- Run `ls` (you should see the `pdf` and the `Rout` file)
- Run `git ls-files` (you should *not* see the `pdf` and the `Rout` file)

Break Time?

New R Project With Your Existing Repo

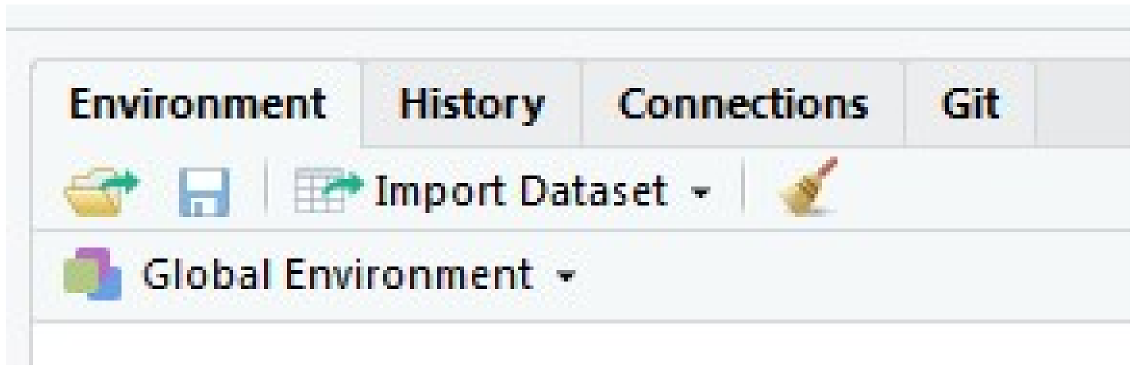

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled1
New Project
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working dire
Version Control
Checkout a project from a version control repos
Console
Terminal

```
R is a collabor
Type 'contribut
'citation()' or

Type 'demo()' f
```

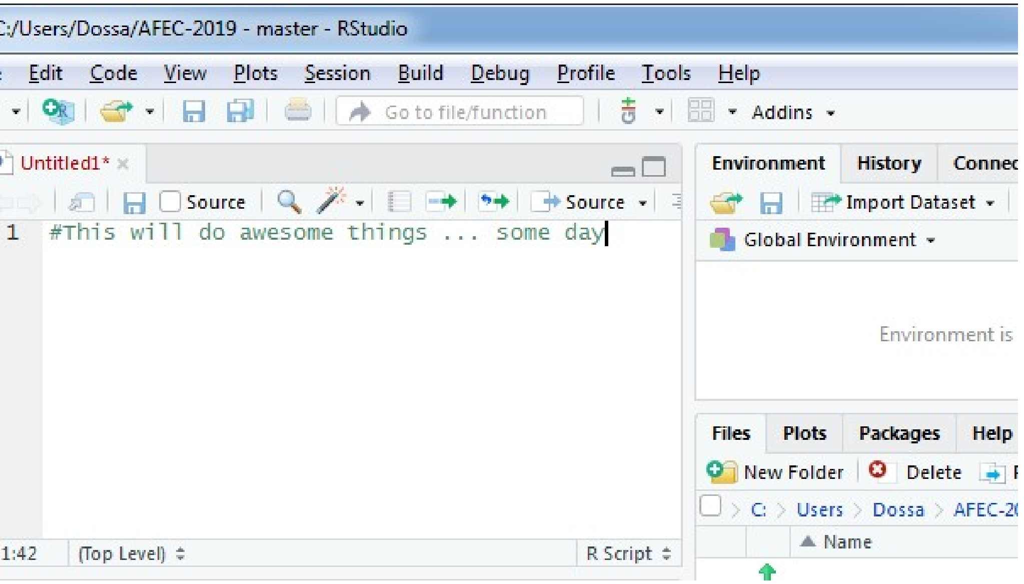

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled1
Environment
New Project
Back
Create Project from Existing Directory
Browse...
Console
Terminal

Project working directory: C:/Users/Dossa/AFEC-2019

```
R is a collabor
Type 'contribut
'citation()' or

Type 'demo()' f
```



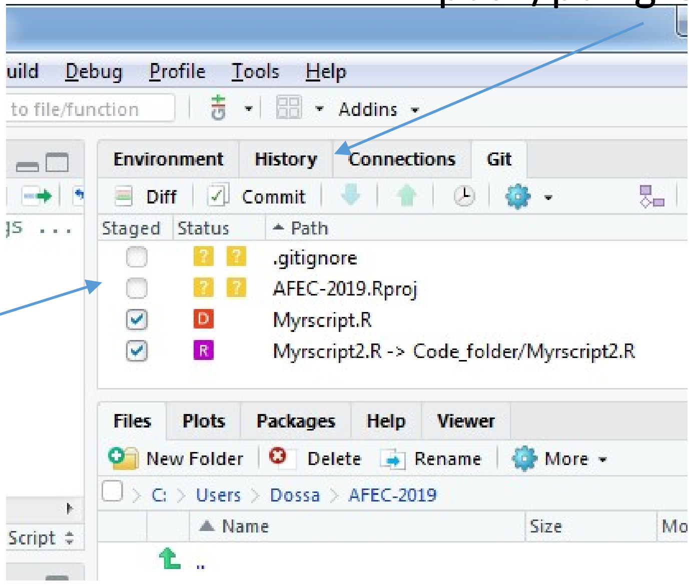
Make the Changes in RStudio



C:/Users/Dossa/AFEC-2019 - master - RStudio
Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled1*
Source
Source
Environment History Connec
Import Dataset
Global Environment
Environment is
Files Plots Packages Help
New Folder Delete
C: Users Dossa AFEC-20
Name
1:42 (Top Level) R Script

```
#This will do awesome things ... some day
```

push/pull greyed out

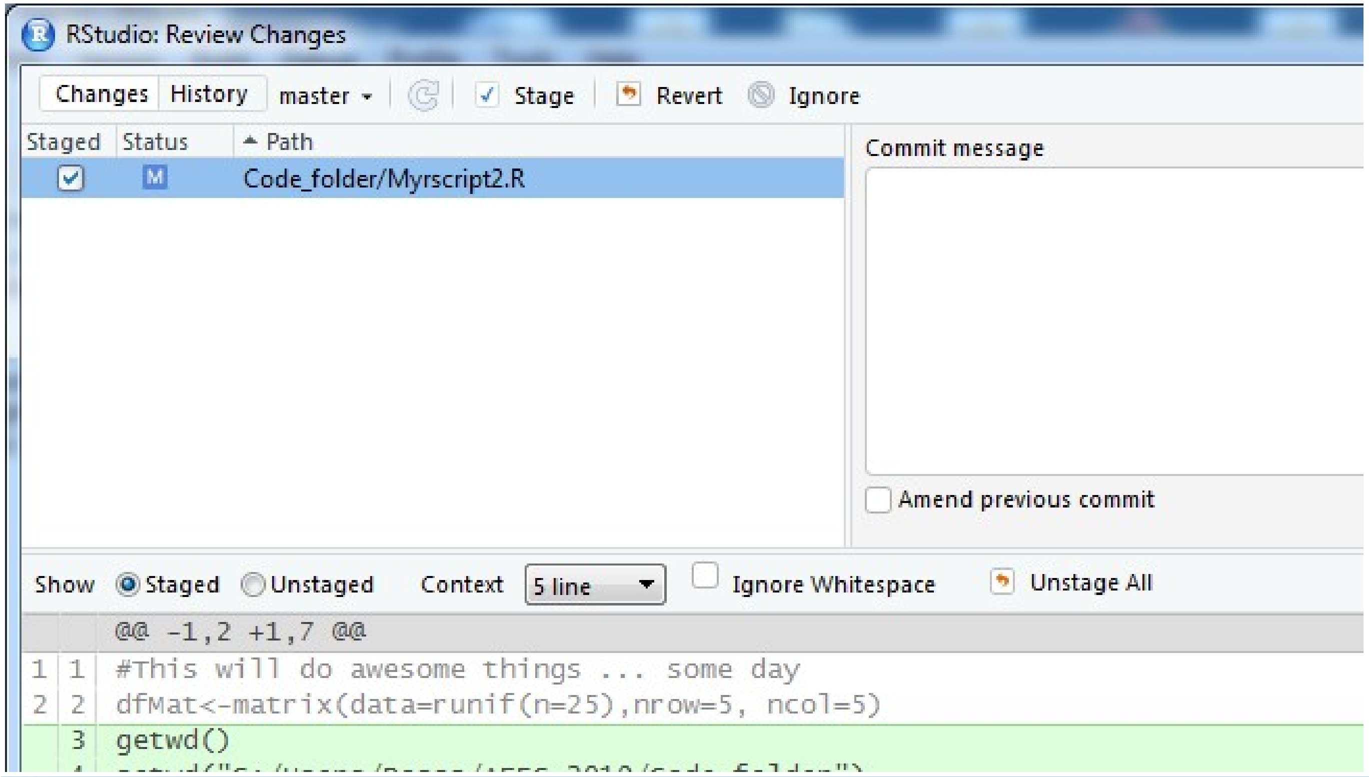


uild Debug Profile Tools Help
to file/function
Addins
Environment History Connections Git
Diff Commit
Staged Status Path
Files Plots Packages Help Viewer
New Folder Delete Rename More
C: Users Dossa AFEC-2019
Name Size Mo
Script

Staged	Status	Path
☐	? ?	.gitignore
☐	? ?	AFEC-2019.Rproj
✓	D	Myrscript.R
✓	R	Myrscript2.R -> Code_folder/Myrscript2.R

Add / made changes

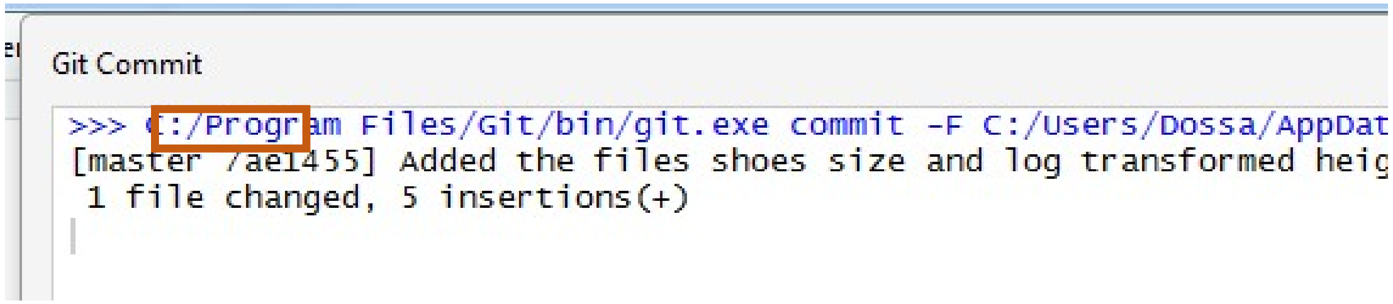
Commit the Changeset



Staged: ✓ Status: M Path: Code_folder/Myrscript2.R

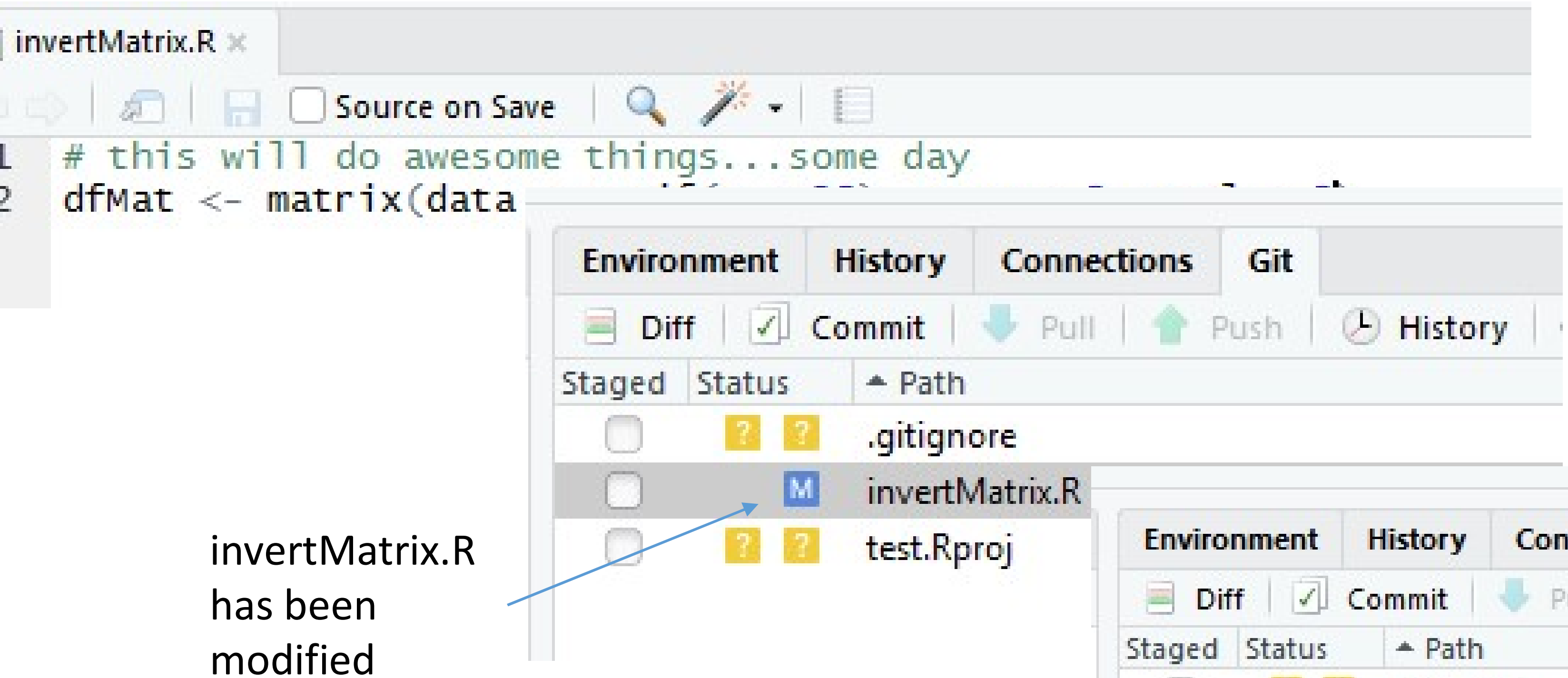
Show: [✓ Staged] Unstaged Context: 5 line

```
@@ -1,2 +1,7 @@
#This will do awesome things ... some day
dfMat<-matrix(data=runif(n=25),nrow=5, ncol=5)
getwd()
```



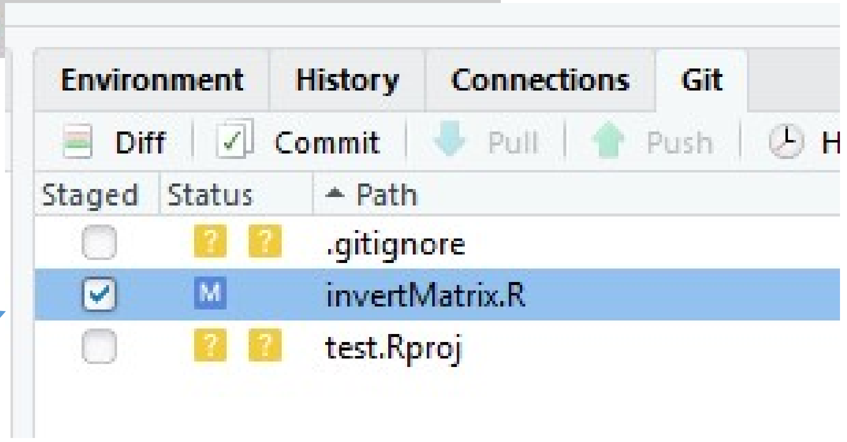
```
>>> C:/Program Files/Git/bin/git.exe commit -F C:/Users/Dossa/AppDat
[master 7ae1455] Added the files shoes size and log transformed heig
 1 file changed, 5 insertions(+)
```

Make Some Modifications



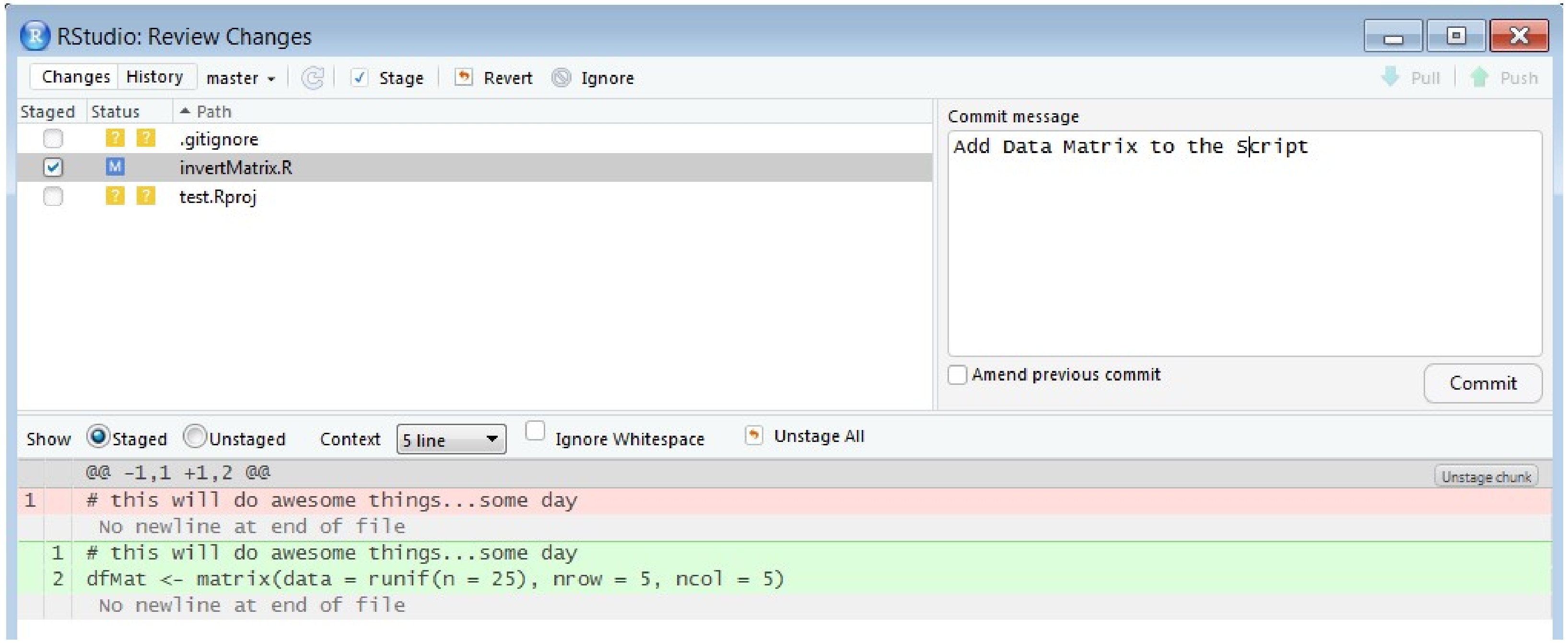
```
# this will do awesome things...some day
dfMat <- matrix(data
```

invertMatrix.R has been modified



invertMatrix.R has been added

Commit the Modifications


RStudio: Review Changes
Changes History master Stage Revert Ignore Pull Push
Show Staged Unstaged Context 5 line Ignore Whitespace Unstage All
Unstage chunk
Amend previous commit
Commit

Staged	Status	Path
☐	? ?	.gitignore
✓	M	invertMatrix.R
☐	? ?	test.Rproj

Commit message

Add Data Matrix to the Script

```
@@ -1,1 +1,2 @@
1     # this will do awesome things...some day
       No newline at end of file
   1  # this will do awesome things...some day
   2  dfMat <- matrix(data = runif(n = 25), nrow = 5, ncol = 5)
       No newline at end of file
```

Git in RStudio Covers *Most* of Your Needs

- If RStudio does all of this, why bother with the command line?
- *Most* of the time you won’t need to, but when you need it, you need it
    - Merge conflicts, for example

Code Time