INTRO TO DATA SCIENCE
LECTURE 10: VERSION CONTROL WITH GIT

Rob Hall
DAT9 SF // September 11, 2014

LAST TIME:

- CLUSTERING
- K-MEANS
- K-MEANS CLUSTERING IN PYTHON

QUESTIONS?

I. INTRO TO VERSION CONTROL

II. LAB: GIT INSTALLATION

III. SUBLIME TEXT TUTORIAL

IV. PYTHON DEBUGGER TUTORIAL

V. WEB SCRAPING TUTORIAL

VI. IN-CLASS EXERCISE ON 3, 4, 5 ABOVE

VII. LAB: GITHUB SETUP (IF TIME ALLOWS)

INTRO TO VERSION CONTROL

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it?*

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it? (of course not)*

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it? (of course not)*
- *But ever had a co-worker do that...? ;-)*

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it? (of course not)*
- *But ever had a co-worker do that...? ;-)*
- *...and then pushed that code to your production website?!?!*

Why do we care about version control?

- *Ever had a hard drive crash?*

- *Ever had a computer stolen?*

- *Ever deleted the wrong file?*

- *Ever modified code and unintentionally broken it? (of course not)*

- *But ever had a co-worker do that...? ;-)*

- *...and then pushed that code to your production website?!?!*

That's why we care about version control!!

Version control is a system that records changes to a file or set of files over time so that we can recall specific versions later.
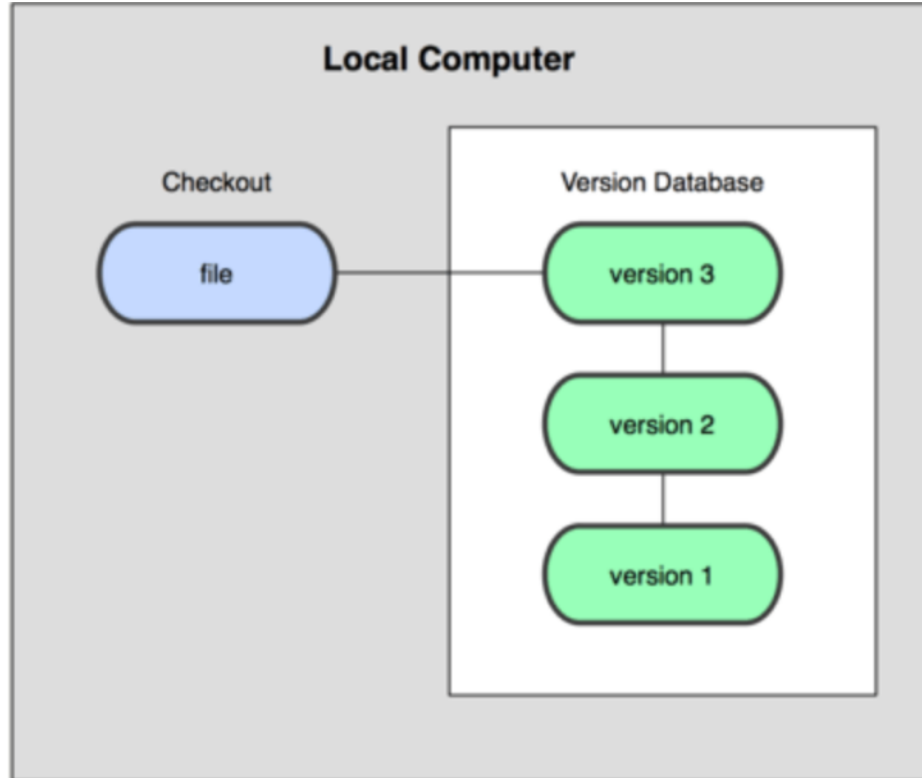
(Think of Time Machine for your Mac)

Version control systems (VCS) can be:
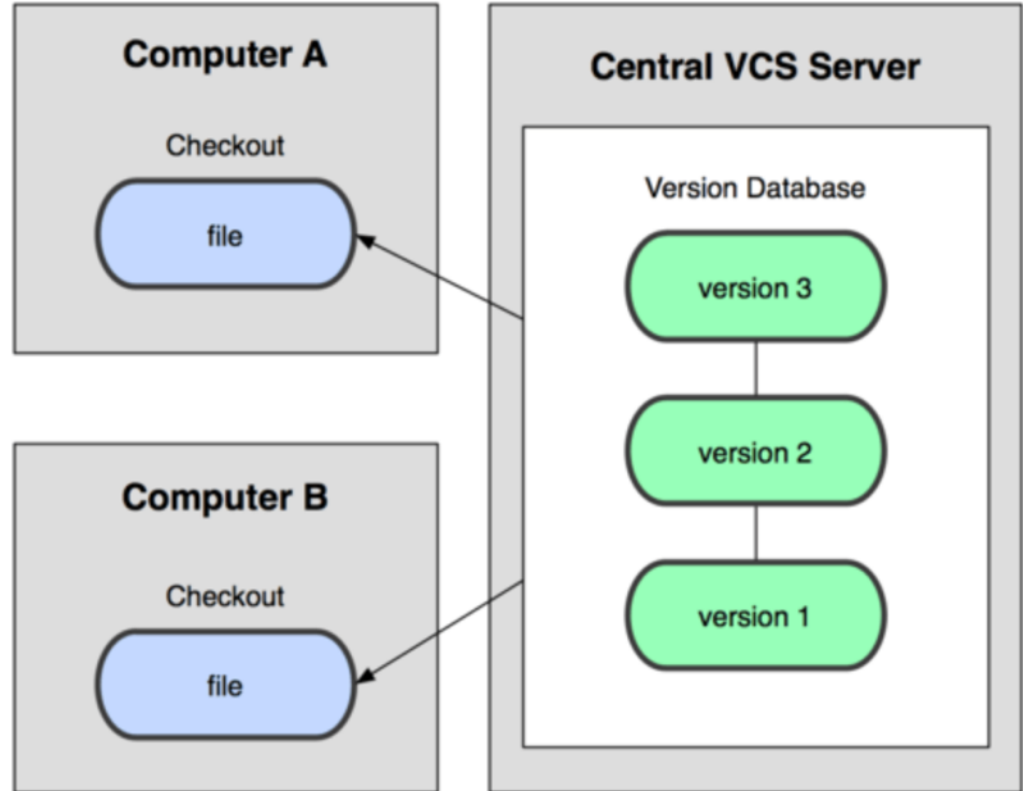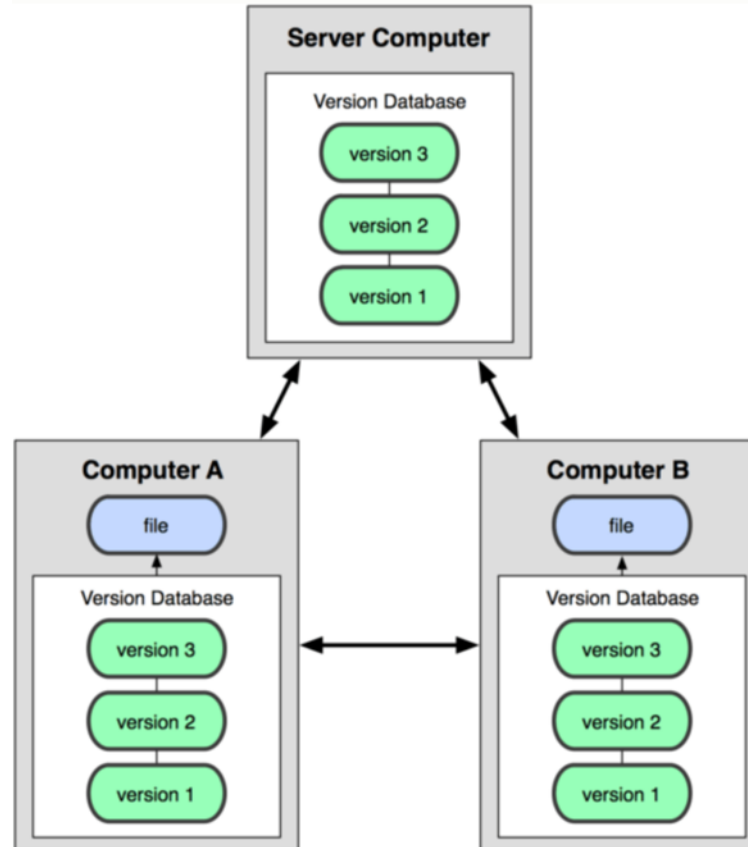
- *Local*
- *Centralized*
- *Distributed*

Local version control:

Centralized version control:

Distributed version control:
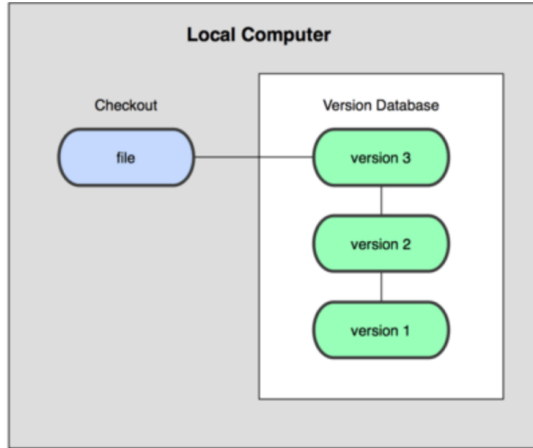
VERSION CONTROL

Local

Local Computer

Checkout — file

Version Database
- version 3
- version 2
- version 1

Centralized

Computer A

Checkout — file

Computer B

Checkout — file

Central VCS Server

Version Database
- version 3
- version 2
- version 1

Distributed

Server Computer

Version Database
- version 3
- version 2
- version 1

Computer A

file

Version Database
- version 3
- version 2
- version 1

Computer B

file

Version Database
- version 3
- version 2
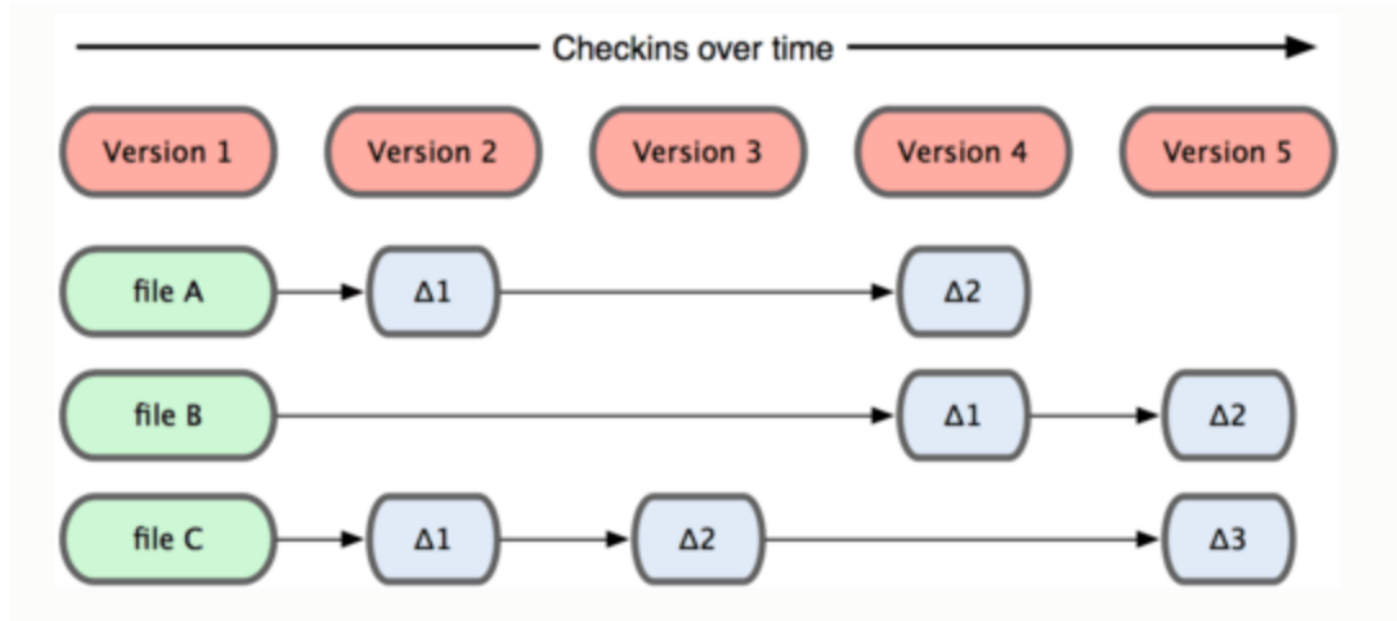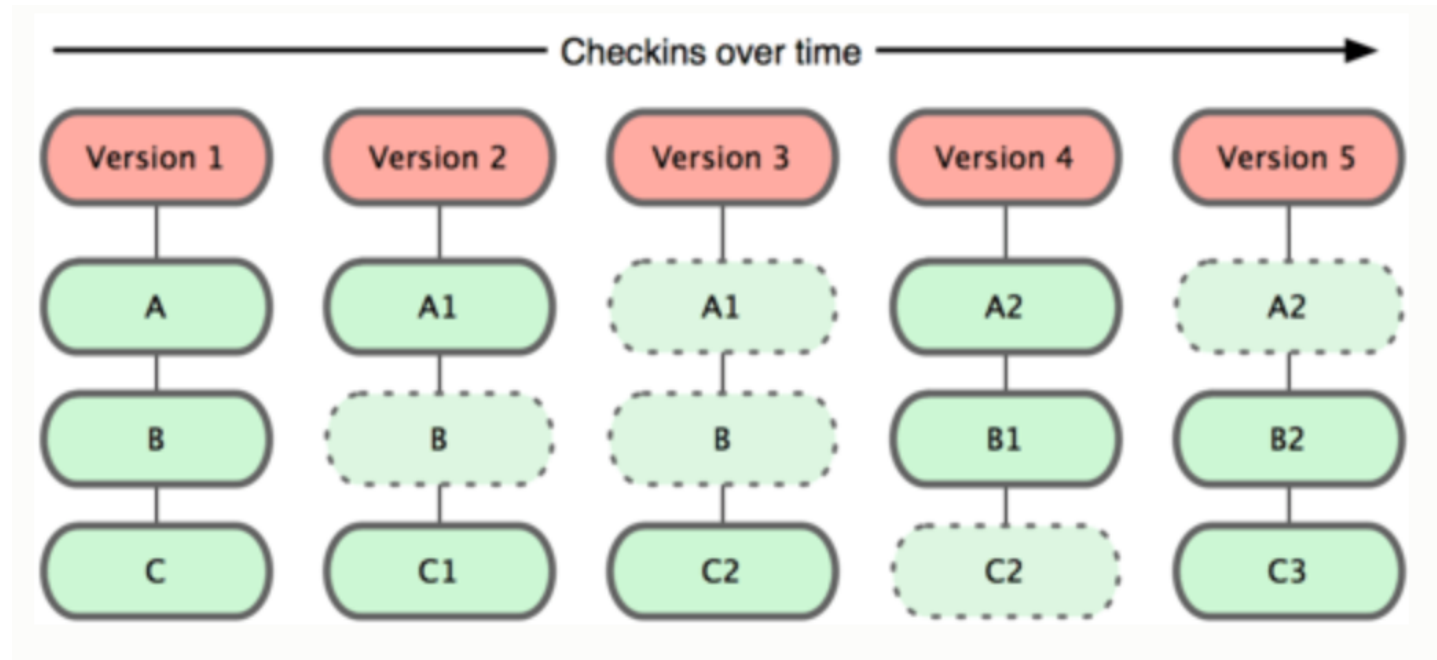- version 1

Traditional VCS' work in terms of files and differences

Git stores data as snapshots of a project over time

modified, staged, committed

Local Operations

working directory

staging area

git directory (repository)

checkout the project
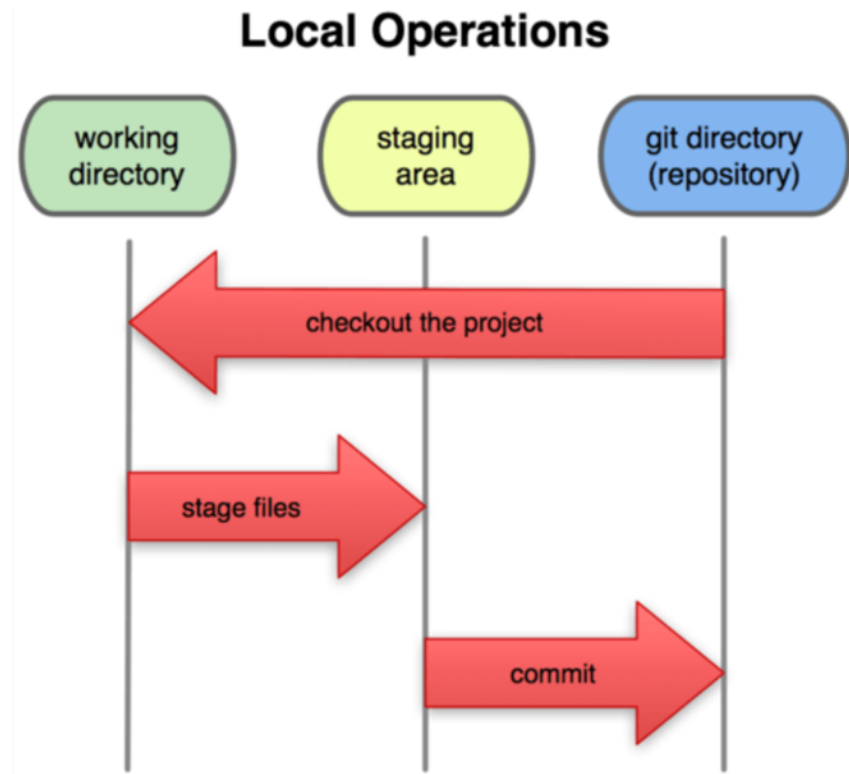
stage files

commit

The basic Git workflow

1. *You modify files in your working directory.*

2. *You stage those files by adding snapshots of the files to your staging area.*

3. *You do a commit, which takes the files in the staging area and stores that snapshot permanently to your Git directory.*

GIT COMMANDS

Main

git clone – clone a repo

git status – get status

git add – add changes to be pushed

git commit – commit the change with a comment

git push – push the change to github

git pull – pull remote changes from github

Others

git branch – see all branches

git checkout – checkout a branch

git merge – merge in another branch

git stash – stash changes

pull request – remote changes requested to be merged in

LAB: GIT INSTALLATION & CONFIGURATION

To install git on your machine:

1. *Navigate to* http://git-scm.com/download
2. *Download the installer for your platform (e.g. Mac)*
3. *Double-click on the DMG file (in your downloads directory)*
4. *Read the README file*
5. *CTRL-click on the PKG file and click the button to open (Mac)*
6. *Git installs. Wait for it to complete. Then…*

To verify git installation on your machine:

7. *Open a new Terminal window and type:*
    git -- version

8. *The system responds with the version of git being used. Is this the same version number you just installed? Or is it the version that comes installed on Mac OS X?*

```
Last login: Fri Mar  7 14:55:50 on ttys002
Robs-MacBook-Pro:~ rob$ git --version
git version 1.8.3.4 (Apple Git-47)
```


Latest source release
1.9.0
Release Notes (2014-02-14)

Download for Mac
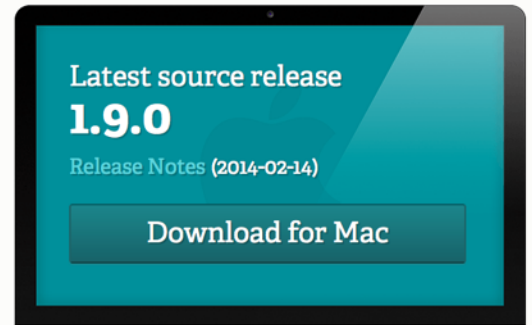
9. *If it is not the same version, then update your PATH by typing the following ALL ON ONE LINE:*

```
echo "export PATH=/usr/local/git/bin:/usr/local/bin:/usr/local/sbin:$PATH" >> ~/.bash_profile
```

10. *Then type:*

```
source ~/.bash_profile
```

11. *Now check the version again:*

```
git --version
```

```
Robs-MacBook-Pro:~ rob$ git --version
git version 1.9.0
```

12. *Now, we proceed to configuration of git:*

```
# Set your username
$ git config --global user.name "Your Name Here"

# Set your email address
$ git config --global user.email "your_name@domain.com"
```

13. *Check your configuration settings:*

```
$ git config --global user.name
$ git config --global user.email
```

14. *#winning!*

Atlassian has a great git tutorial:

https://www.atlassian.com/git/tutorials/