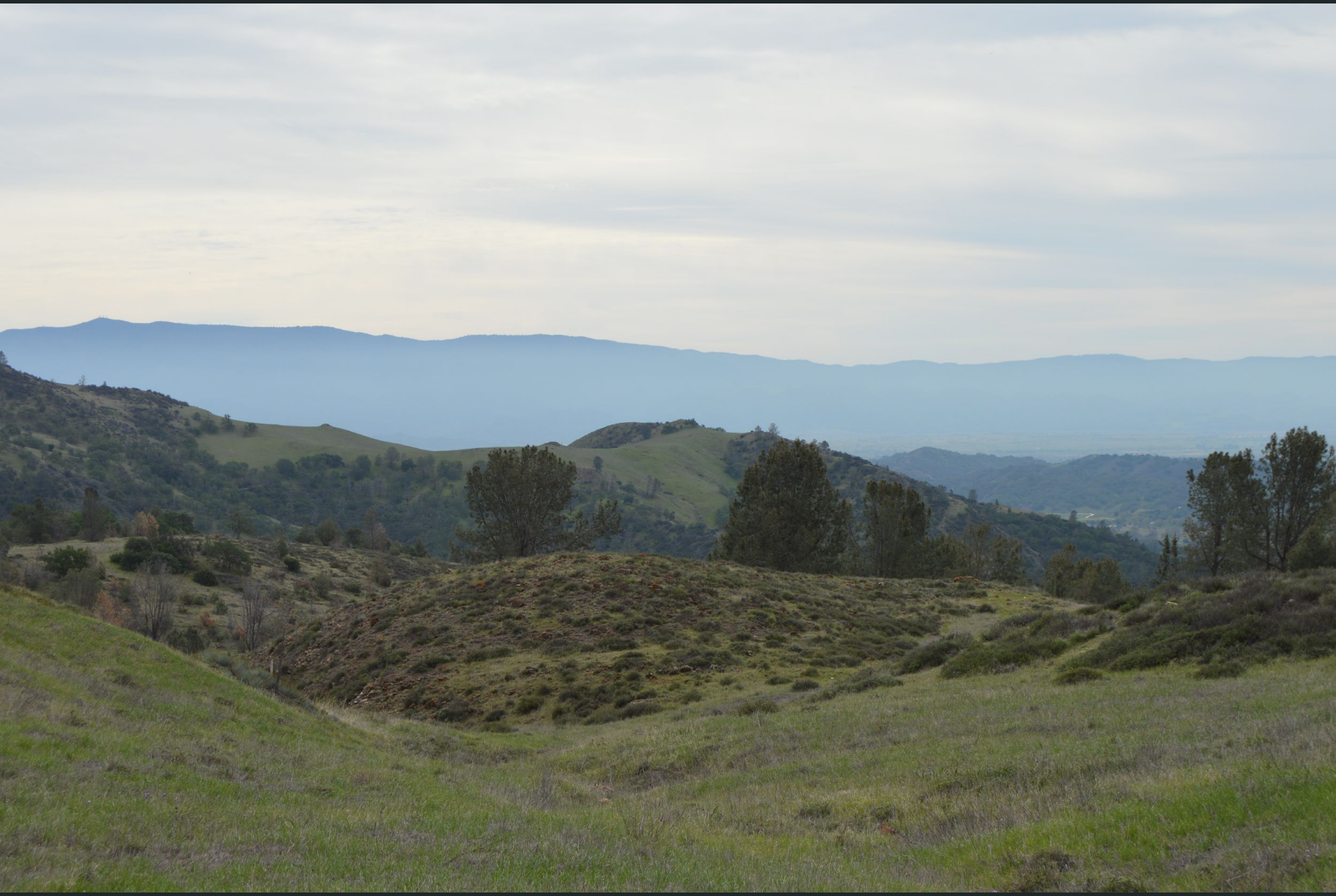
Create a github account;

Follow the instructions at http://gauravsk.github.io/eeb177-W17/2017/01/03/setup-github.html to get Git set up on your virtualbox

(link should be accessible from an email...)

Version control with git and github

Gaurav Kandlikar
11 January 2017



any general course questions?

get to know your neighbors on both sides

you will get much more out of this course if you are willing to ask questions and share what you know with your classmates.

evening hacking sessions

tentatively 6-8 pm on tuesdays in
life science building, room 3215

---

gaurav's office hours
tentatively 4.30-5.30 on monday
time subject to change due to fieldwork...

I encourage you to space out your practice sessions.

Review article

The effect of distributed practice: Neuroscience, cognition, and education

Emilie Gerbier [a,*], Thomas C. Toppino [b,1]

[a] Laboratoire d'Etudes des Mécanismes Cognitifs – EA 3082, Université de Lyon , Université Lumière Lyon 2, 5 Avenue Pierre Mendès France, 69500 Bron, France
[b] Villanova University, 800 Lancaster Avenue, Villanova, PA 19087, USA

CrossMark

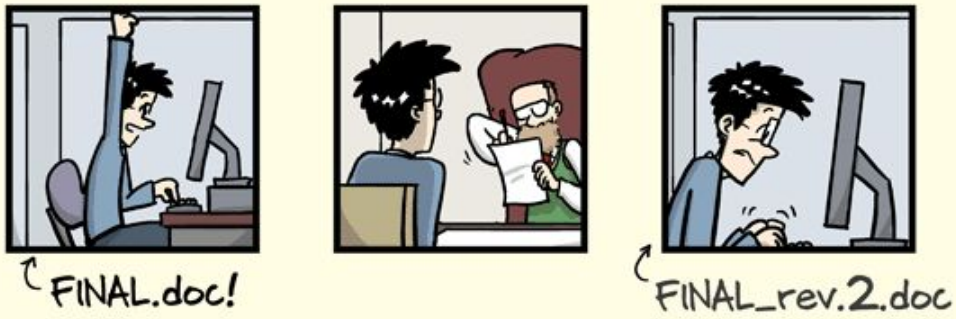
"That is, **it is best to study something by scheduling relatively short study sessions that can be repeated after an appropriate period of time** rather than by devoting the same total amount of time to a single study session or to a number of repeated study sessions that occur in immediate succession."

pourquoi version control?

Projects take a long time to go from inception to completion. We need a system that helps manage our various versions, dead-end ideas, side-projects that grow into their own monsters, etc.

As you progress into increasingly complex projects, proper version control basically becomes a *requirement*.

"FINAL".doc



yihui / **knitr**

Watch 126 | Star 1,292 | Fork 595

Code | Issues 20 | Pull requests 5 | Projects 0 | Wiki | Pulse | Graphs

A general-purpose tool for dynamic report generation in R http://yihui.name/knitr

5,333 commits | **1** branch | **29** releases | **82** contributors

Commits on Jan 3, 2017

see if curl works
yihui committed 7 days ago ✓ — f936809

--no-check-certificate
yihui committed 7 days ago ✗ — 082d5ef

add baseurl
yihui committed 7 days ago — 448b7be

no need to prepend http://yihui.name?
yihui committed 7 days ago ✗ — 39863b0

what is wrong?
yihui committed 7 days ago — 9cc6cba

also https for XRAN
yihui committed 7 days ago ✗ — be93b38

add disqus_url
yihui committed 7 days ago — f856a18

git can be tough to digest.

But the core of the workflow is straightforward:

git add <file>

git commit -m “<commit message>”

git push

git can be tough to digest.

But the core of the workflow is straightforward:

git add <file> **# use whenever you have a new snapshot you want to save**

git commit -m “<commit message>” **# use to “save” the snapshot as a version, including a message**

git push **# use to push the updates to Github, where they can be visualized and shared**

BASIC GIT WORKFLOW

- work on some files in the repository
- committing changes is a 3 step process
  - add the files (**git add**)
  - commit the files with a description of the changes (**git commit -m**)
  - push the files to the remote repository (**git push**)

slide by mike alfaro

There are many good tutorials out there.
Go through a couple of these over the weekend.

http://rogerdudler.github.io/git-guide/ *

https://git-scm.com/book/en/v2

https://guides.github.com/activities/hello-world/ *

http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1/

http://kbroman.org/github_tutorial/

https://www.sbf5.com/~cduan/technical/git/ *

* recommended

Each project in git is a **repository**

A repository contains all the files in the directory, but also contains a series of snapshots committed along the way, as well as a history of what changed between each snapshot.

You can “revert” any file to a previous snapshot, e.g. if your program *used* to work last night but now doesn’t this morning.

Note: “repo” == “repository”

older version

newer version

28 _posts/2017-01-06-exercise1-git.markdown

View

@@ -7,16 +7,16 @@ author: Gaurav Kandlikar

```
 In this exercise, you will go through the "`git add`, `git commit`, `git push`" workflow. In Section 2 of this exercise, you will explore the use of branches and reverts to previous commits.

-Begin this (and all subsequent exercises in this course) by opening up a terminal window and navigating to your `homework` folder.

### Section 1- adding, commiting, and pushing to github

Before you begin, please read through [this helpful summary](https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control#_getting_started)<sup>0</sup> on the basics of version control.

-0. Ensure that you have set up a git repository to track your `homework` folder and that this repository is connected to your github repository ([instructions](http://gaurav.kandlikar.com/eeb177-W17/2017/01/03/setup-
```

```
 In this exercise, you will go through the "`git add`, `git commit`, `git push`" workflow. In Section 2 of this exercise, you will explore the use of branches and reverts to previous commits.

+Begin by opening up a terminal window and navigating to your `lab-work` folder.

### Section 1- adding, commiting, and pushing to github

Before you begin, please read through [this helpful summary](https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control#_getting_started)<sup>0</sup> on the basics of version control.

+0. Ensure that you have set up a git repository to track your `lab-work` folder and that this repository is connected to your github repository ([instructions](http://gaurav.kandlikar.com/eeb177-W17/2017/01/03/setup-
```

Github offers a way to visualize your repositories and code collaboratively.

Researchers can collaborate on code very easily through github- simply “pull” in the latest updates made by your collaborators, change the code, and “push” your changes back to the repository!

Collaborators (or even total strangers) can comment on each other’s code to point out errors, make improvements, etc.

I can view any homework you commit without you having to email me, etc.

A quick introduction to “git clone <repo website>”

“git clone” is an easy way to copy an existing repository that is publically available on github onto your local computer.

Note that you are doing more than just downloading the files- you are downloading the whole history of the files! To see what I mean, just type `git log` into a repo that you have recently cloned!

Today's exercise

Today you will start practicing the git workflow with dummy files. You will add, commit, and push several times from your terminal, and then visit your github page to visualize your revision history.

In Section 3 of the exercise, you will explore some slightly more advanced git concepts- how to make a new *branch*, how to merge changes, etc.

In the homework, you will make a fresh repository, push changes through to github, and clone one of my repositories onto your desktop.