Activity 1.1

# GitHub and Markdown

Requirements:

* GitHub account

Goals:

* Navigate and visualize the commit history of various repositories on GitHub
* Demonstrate using the typical Git workflow
* Merge repository branches via pull requests on GitHub, resolving conflicts if necessary

## Introduction

This activity is designed to introduce you to Git and GitHub, which we will be using during this semester for file management, version control, and a collaboration tool! Before we get into collaborating with GitHub, we need to understand some of the basics. That is, many of the tasks in this activity are done individually, but talk with your team members and help each other out! You are also provided some discussion and reflection prompts throughout to talk about as a team.

## Getting Started

1. Navigate to the [discussion](https://github.com/sta518/discussion) repository.
2. Click “ Watch ” near the upper-right corner of the repo, so that you get email notifications.
   * You should consider bookmarking this page, but it is also linked on the left-hand toolbar of the [course website](https://sta518.github.io) ( Help)
3. Comment on the Issue I created called “Introduce yourself!” In your post
   * Say who your are,
   * Share one “thing of interest” about you,
   * Tag your neighbor(s) using @username, and
   * Ask any question(s) you have regarding the [Syllabus and Course Policies](https://sta518.github.io/syllabus/)
4. If you haven’t yet, add a profile picture to your profile to help make the STA 418/518 community feel more personable. You can do this from your profile homepage.

## Big Picture

To give me some more time to set-up some class materials (GitHub Classroom), explore these blog posts that were made with R:

* [Text analysis of Trump’s tweets](http://varianceexplained.org/r/trump-tweets/)
* [A year of fitbit](https://livefreeordichotomize.com/2017/12/27/a-year-as-told-by-fitbit/)
* [R-Ladies global tour](https://livefreeordichotomize.com/2017/12/27/a-year-as-told-by-fitbit/) (with a gif created in R at the end!)

## Team Work

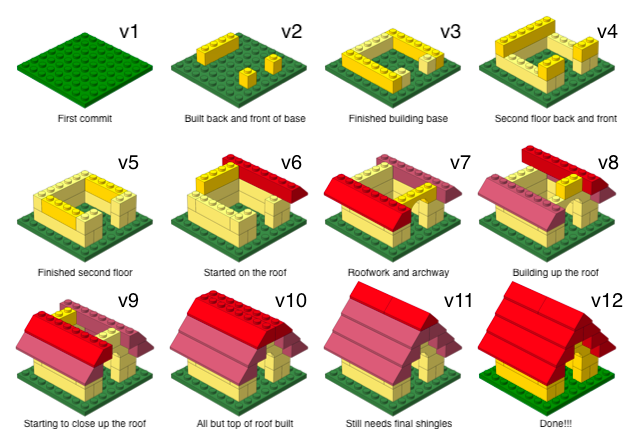
In this course, we will make extensive use of students working in teams. Most of you have had experiences, some positive and some negative, with group work.

As a team, brainstorm a list of Pros and Cons for group work. A team is composed of team members. Identify at least three (3) and no more than five (5) traits that an ideal team member would have. When you have your pros and cons and traits, have a member write them in the corresponding “column” on the white board.

### Version Control with Git

The statistical programming language that we will use is R. We will interface with R using the software RStudio - these are the focus of every activity *after* this first one.

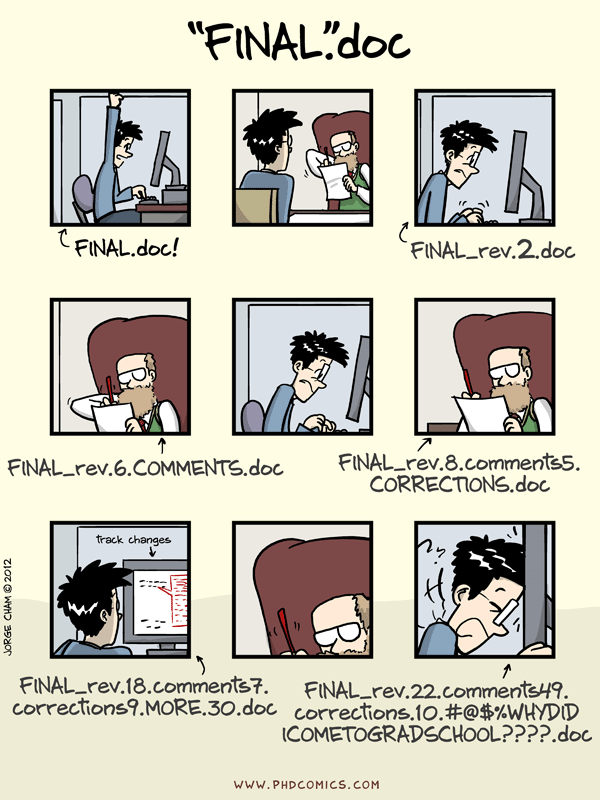
There are number of ways that I could get you the course materials and assignments, but we will use GitHub as our platform for collaboration and version control.



Version Control with human readable messages

From [Mine Ãetinkaya-Rundel](http://www2.stat.duke.edu/courses/Spring18/Sta199/slides/lec-slides/01-meet-toolkit.html#27)

Version control is useful so we avoid:



“FINAL”.doc

As a team, think of some other version control-like software that you have used before.

Git has many commands - most of the time we will only need to use git add, git commit, git push, and git pull. Also, we will primarily use git by the built in interfaces with GitHub and RStudio.

If you Google for help with Git and come across command line instructions, skip it and move on to the next resources unless you feel comfortable trying that method.

A great in depth resource for working with Git and R (linked on the course site at [Additional Resources/GitHub](https://sta518.github.io/resources/github/)) is [Happy Git with R](https://happygitwithr.com/) - this would be a good place to start looking for help.

### GitHub

If Git/version control is like “Version history” in Google Docs, GitHub is like Google Drive - Git and GitHub being more rich in features and capabilities.

In this section, we will explore some of the basic GitHub capabilities: repos, commits, diffs, commit history, branching, and merging

#### Repositories

To demonstrate version control concepts, you are going to create a new GitHub repository.

**Create a new repository.**

Part 1:

1. If you are not still logged into GitHub, do so.
2. Click the green “ New” button
3. On the **Create a new repository** page, name this something like “sta518-test”.
4. Check the box to “Initialize this repository with a README”
5. Click on the green “Create repository” button.
6. On the main repo page, click the to edit the README - You can simply add something like, “This repo is to explore GitHub and markdown for STA 518” on line 2. - Add a descriptive commit message in the first box like, “Added repo information” - Click on the green “Commit changes” button

Part 2:

1. On [Bb](https://mybb.gvsu.edu), in “Documents” there are three documents: 1) “day1.md”, 2) “day1.html”, and 3) “day1.pdf”. Download these files and note where they were saved for the next step (probably in your Downloads folder).
2. Locate the three files you downloaded, then drag-and-drop them into the main repo page.
3. Explore each of these files within your repo. Discuss with your team members about what is view-able and what isn’t. Keep this in mind as we progress though the semester: Jenny Bryan provides some great information on [repo browsability](https://happygitwithr.com/workflows-browsability.html)

#### Commits

You already did a couple of commits in the previous section. Now we will compare commits with branching and merging. Remember that Brian Yu’s seminar talked about branching and merging and you might want to review this in more detail after class. We will talk more about the motivation behind repo branching later.

**Create a branch.**

Part 1:

1. On the main repo page, create a new branch

* Find the “Branch: master ” button and click on it
* In the text box call your new branch “test1” and create the branch

1. You should now see that you are on “Branch: master ”

* Edit this README (on the test1 branch) with a [relative link](https://github.blog/2013-01-31-relative-links-in-markup-files/) to the day1.md file in the repo from when you created the repo.

1. Explore these two branches by switching between them to see that the repo structure is different

* As a team, diagram the repo structure with a tree diagram

1. Merge your “test1” branch to the “master” branch by opening a *pull request* (one of the upper tabs)

* Close your “test1” branch
* Update your team’s tree diagram

When you open a pull request, youâre proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show diffs, or differences, of the content from both branches. The changes, additions, and subtractions are shown in green and red.

From GitHub’s [Hello World](https://guides.github.com/activities/hello-world/) activity (which we are mostly following). GitHub also has some documentation on [Pull Requests](https://help.github.com/articles/about-pull-requests/) that you may want to use to learn more outside of class.

Part 2:

1. Make a new branch call “test2”
2. Edit line 1 of the README on **both branches** to something different for each case

* For example, in the master branch add “this is the master branch” and in the test2 branch add “this is the test 2 branch”. 3 Try merging with a pull request.
* You should receive a *merge conflict* – resolve it (your choice), then merge.

Merge conflicts will more than likely come up as you work in your teams throughout the semester. To resolve them, you (the repo owner) just needs to decide which is the correct version

## Reflection

As a team, discuss your responses to these questions.

What is version control and why do we care? What is Git vs GitHub and do I need to care? What is still muddy?

## Up Next

### R and RStudio

In the next activity we will begin collaborating with GitHub (forking and pull requests within your teams) while using R and RStudio.

### Markdown

The \*.md files that we’ve seen are markdown documents. There are many [Additional Resources](https://sta518.github.io/resources/markdown/) linked on the course site and a great 10 minute tutorial that touches the basics from [CommonMark](https://commonmark.org/help/tutorial/) (this is part of your **Meeting Preparation** for the next activity).