**ESM 244 Week 1 Discussion:**

**Git reminders, branches and pull requests!**

**Intro:** Allison gives a conceptual overview of why a branch can be useful (testing stuff, features, non-disruptive collaboration, development without deployment).

**SETUP**

* Log-in to GitHub
* Create a new repo called ‘trash’ or ‘trash-repo’ or something, with a ReadMe
* Clone to create a local version controlled R Project (see screenshot)
* Graphical user interface, text, application, email

  Description automatically generated
* Initial commit (w/ .gitignore and .Rproj files added)
* Create a new RMarkdown document, delete everything past the first code chunk, attach the *tidyverse* package
* Stage, commit, pull, push to main
  + Commit sets up changes/take a snapshot of what everything looks like
  + Then pull to see whether anything has changed in the origin
    - Will do its best to combine everything
  + Then will merge changes and then push
  + If you commit then push it’s probably fine, but if there are changes on the remote it’ll say hold on you have to pull first

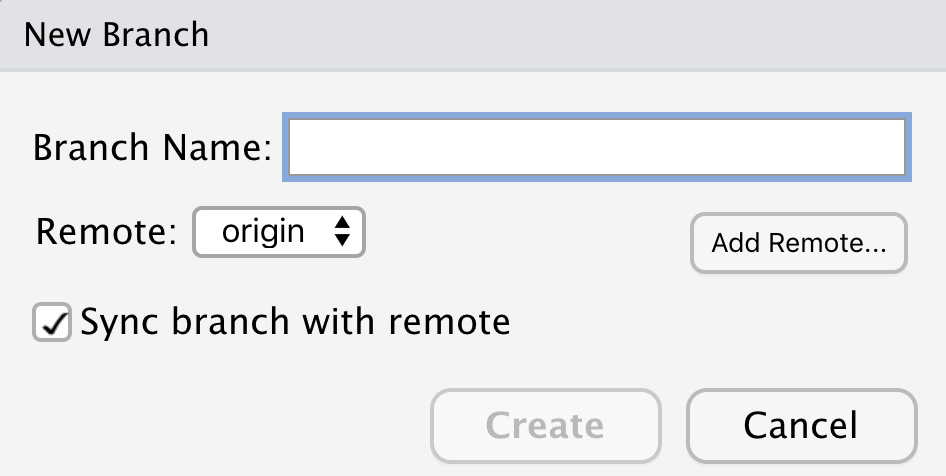
**MAKE A BRANCH (2-ways)**

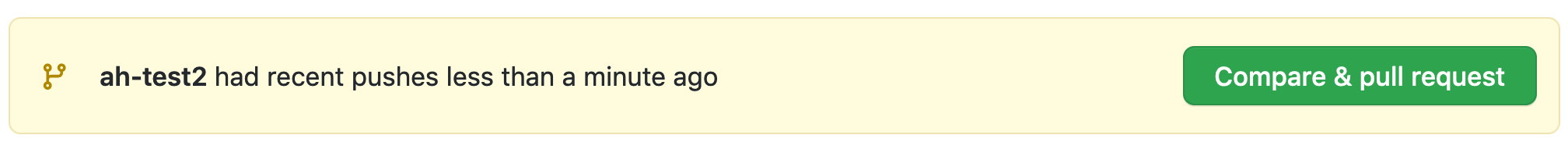
***Way 1: Friendly RStudio buttons***

* In your R project, click on the Git tab
* Then click on the purple “branch” button it the upper right:



...which opens a window that looks like this:



* Add a branch name (like initials-branch, e.g. mine is ah-branch), then click ‘Create’
* Notice that the branch you’re now in shows up in the top bar of RStudio, as well as in the top right of the Git Tab
* While still in the branch, update the RMarkdown document by…
  + Making a basic scatterplot of character mass versus height using the *starwars* data (built-in, dplyr)
  + Update the graph point color to “purple”
* Stage, commit, pull, then push
* Go back to GitHub & refresh your repo, and you will \*probably\* see a highlighted message that looks something like this:
* 
* Click on ‘Compare & pull request’ - you should see that it can be merged automatically (conflicts later…)
* Write a little descriptive note, and submit the PR
* Merge. TADA.

**GET BACK IN MAIN BRANCH**

* In the Terminal tab in RStudio (or go to Tools > New Terminal), type:
  + git checkout main
    - Note: git checkout is the command to switch between branches
    - Check that you’re in main
* Also in Terminal: git pull origin
  + (like pressing the ‘Pull’ button to get remote updates!)

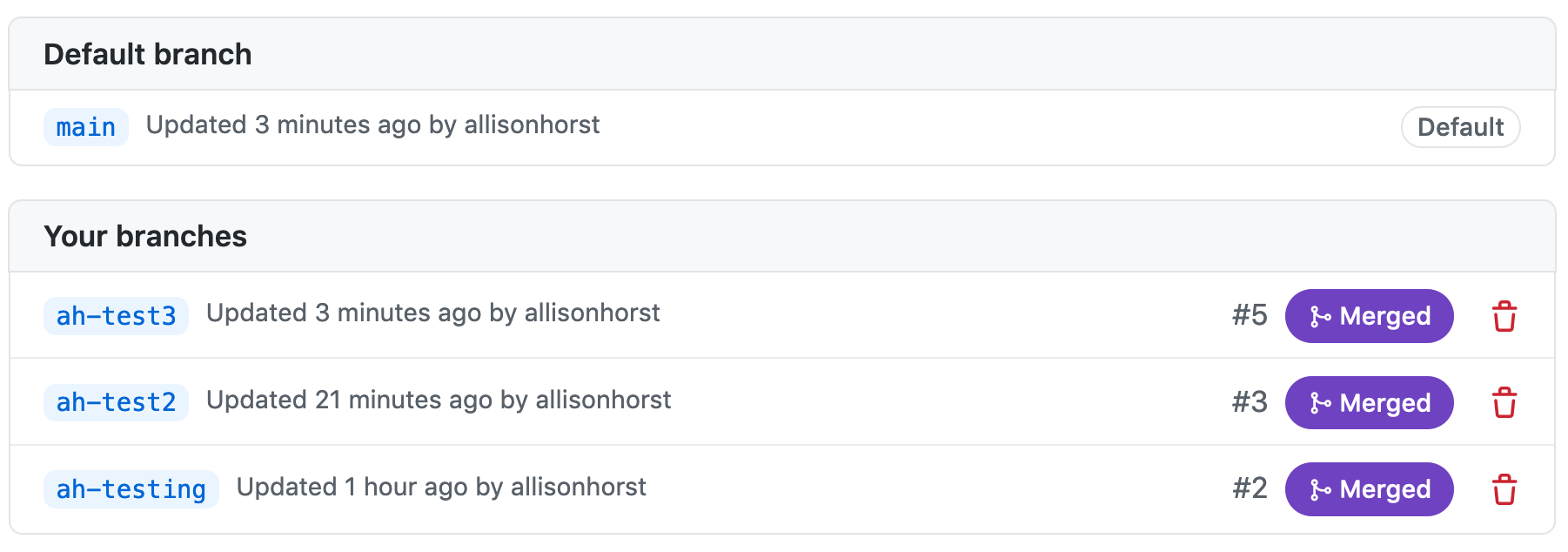
**MAKE A NEW BRANCH IN THE TERMINAL**

* *Aside from noob Julianna: Set up your account:*
  + [*https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html*](https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html)
  + *Cd into folder if you’re not doing it through R studi*o
* Type the following to create and switch to a new branch:
  + git checkout -b branch-name
  + For example, mine might be git checkout -b ah-feature
  + Press Enter - note that you’re in your newly created branch! Cool.
* Connect branch to remote (set upstream branch) by typing in the Terminal:
  + git push -u origin branch-name
* Open the .Rmd in the your new branch, now connected to your upstream branch, and update the point colors to “green”
* Stage, commit, pull, push. OK to use RStudio buttons here. If you want to try doing this from the command line, do:
  + git add . *or* git add --all (to stage everything in the current directory, like checking the boxes in R Studio)
  + git commit -m “type in a commit message”
  + git pull
  + git push
* Go back to GitHub - you will either see a notification again, OR you can click on branches from your repo home to see any unmerged updates.

That button is here:



And once you click on it you’ll see something like this:



And if you HAVE any unmerged updates to a branch, then the purple button will say “New pull request” & you can click to submit a PR then merge.

**Congratulations! You’re doing some work in git branches.**

**Question:**

*Can I do every git-related thing from the Terminal?* **YEP.** We’re just starting small with a few commands to get used to it. Moving forward we’ll keep building our command line skills through 244, too.

**END DISCUSSION 1**

***Normal workflow:***

1. ***Create repo on Git hub***
2. ***Clone it onto your computer in RStudio***