Project 1 Instructions

Vignettes

Vignettes are explanations of some concept, package, etc. with text, code, and output interweaved. We already know how to make them with R Markdown!

Project Objectives

This project involves creating a vignette (HTML with table of contents) about reading in data and exploring it. Everyone is paired with another student from your group of four and will work with their partner (see below for group info).

Group	JSON Partners	XML Partners
Group A	Evan Elms, Cameron Evangelista	Ariel Macari, Xiyou Zhou
Group B	Kelly Craig, Sejin Lim	Anchal Saxena, Michael Yan
Group C	Jesse DeLaRosa, Sophia Melenikiotis	Steven Miller, Clayton Ramsey
Group D	Heather Barker, Shantel Butler	Jin Hong, Ben Pfeffer
Group E	Kristin Calvert, Xingyue Gong	Phillip Rodriguez-Lebron, Adam Weimerskirch
Group F	Greg Janesch, Hongliang Liu	Ryan McGovern, Steve Sortijas
Group G	Kyle Beard, Laura Clebone	Nick King, Quin Nelson
Group H	Tyler Brigman, Stuart Cary	Erkang Ou, Melissa Rineer
Group I	Avisek Choudhury, Pierre Cobb	Yan Shi, Thom Teears

The JSON group will create a vignette about reading JSON data into R with an example that includes basic exploratory data analysis (summaries and graphs - more details on this later). The XML group will do the same with an XML data set.
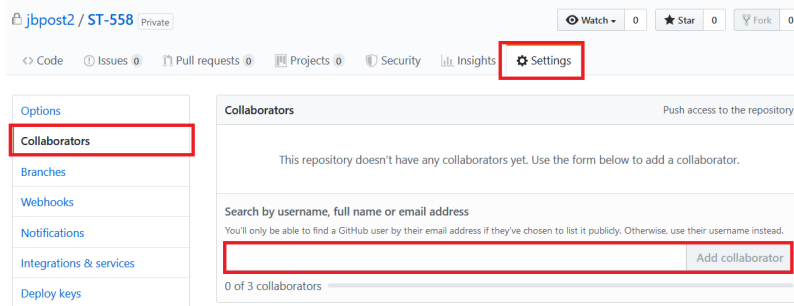
## Group Work

Each group will collaborate through GitHub.

- One **partner** should make a repository
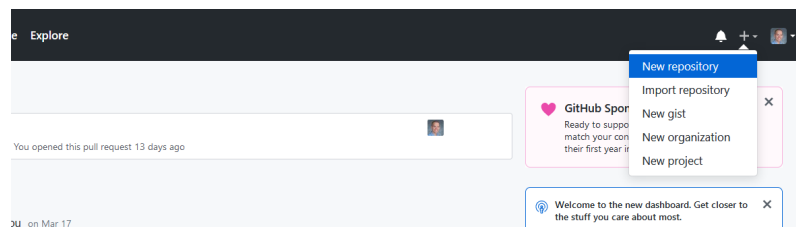
- That person should add all **three** other group members to the project
  - Go to settings, collaborators, and search for your groupmates (usernames may need to be shared)

- All changes to documents should be done on the repositories (this way when grading I can see how much each group member contributes :)

- The JSON partners will work independently of the XML partners but incoroporate feedback given by their counterparts (and vice-versa).

  - At some point the JSON partners should ask the XML partners for feedback (with a reasonable amount of time for them to respond). Each of the XML partners must submit at least one pull request on the vignette (and vice-versa).

 - To do so, the XML members should first `fork` the repository
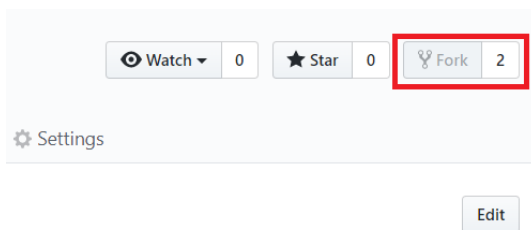
  - Clone the repo locally (i.e. grab the clone URL and start a new project in R Studio (version control -> git) and put in the repo URL)
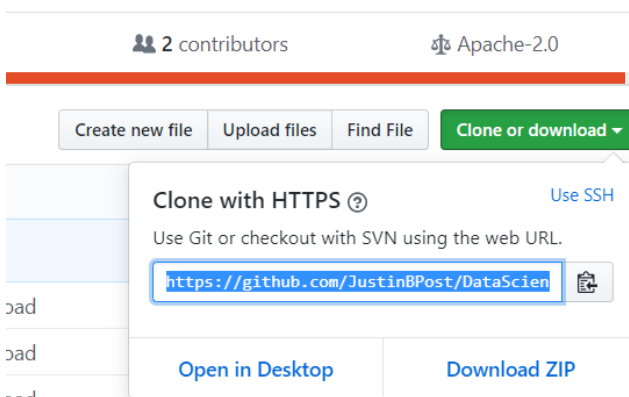
  - Make comments/edits locally and commit to your forked repo

  - Perform a pull request to merge your branch to the original repo master branch
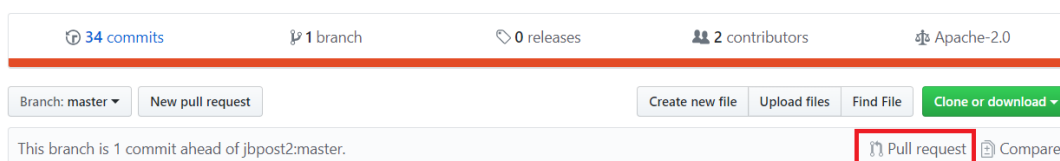
– Now the JSON group can view pull requests and choose which changes to include
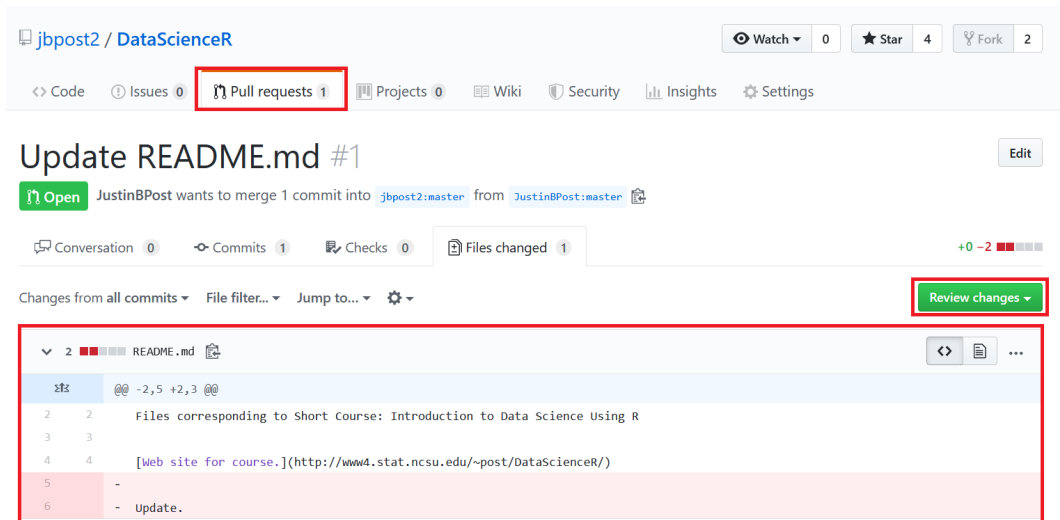
Vignette Content Details

The components of your vignette that must be present include:

- Describe your type of data (JSON or XML). What is it, where does it get used, and why is it a good way to store data? This should be detailed enough that someone that hasn't seen that type of idea would have a good idea what they are dealing with. You should link to references where applicable.

- Discussion of possible packages/functions that are available for reading your type of data into R. Choose one and explain why you've chosen it.

- Find a dataset of your type (JSON or XML) and describe where you found the data, how the data was collected, what the variables are, etc.

    – The data you read in should have at least two categorical variables and two quantitative variables.

- Read in the data set describing the options your package's functions allow.

- Perform basic exploratory data analysis that reveals a meaningful idea that you would reasonable want to investigate further. Not all things reported need to show something meaningful (i.e. graphs that show no relationship are fine) but you should end up with a solid lead that you would pursue further.

    – At some point you should create a useful function(s) to do something meaningful with the data or customize the way you read the data in.

    – You should create a new variable.

    – You should create some contingency tables and numeric summaries by some of your categorical variables

    – You should create some plots (at least a side-by-side bar plot, side-by-side box plots, and scatter plots with coloring)

When you are finished, you should post a link to you/your partner's repository for us to use along with a rating score for each of your three other group members. (0 - implying that member didn't contribute in their assigned way to 100 - implying that member contributed fully in their assigned role.)

All vignettes will be shared with the class after the due date and grading are finished.

Rubric for Grading

Item	Points	Notes
Collaboration and use of git/gitHub	–	See below.
General use of Good Programming Practices	15	All code should be indented, follow naming conventions, etc.
Use of headings, table of contents, chunk options, etc.	5	Worth either 0, 3, or 5
Data type (JSON or XML) description	15	Worth either 0, 5, 10, or 15 points
Discussion of relevant R packages	10	Worth either 4, 7, or 10
Description of Data Set	15	Worth either 0, 5, 10, or 15 points
Custom function (best practices should be used)	10	Worth either 4, 7, or 10
Creation of relevant new variable(s)	5	Worth either 0, 3, or 5
Calculation of relevant numeric & graphical summaries	25	Worth either 0, 5, ..., or 25 points

Notes on grading:

- For collaboration and git/gitHub use we will look at the commit history and group ratings to judge. If someone is deemed not to have contributed or to not have contributed much, they will of course receive little to no credit.

- For items worth say 0, 5, 10, or 15 points, we will generally move you down one level for each syntax, logical, or other error present in the code. The same holds true for missing a required item or lacking in a description.