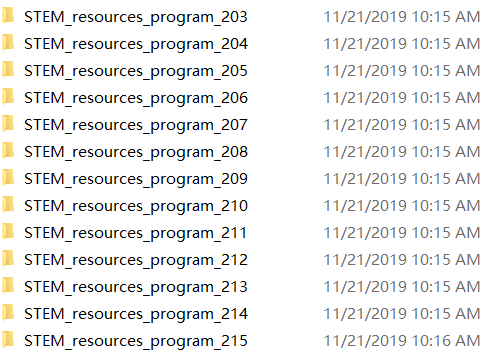
Sweet Cheetos ReadMe

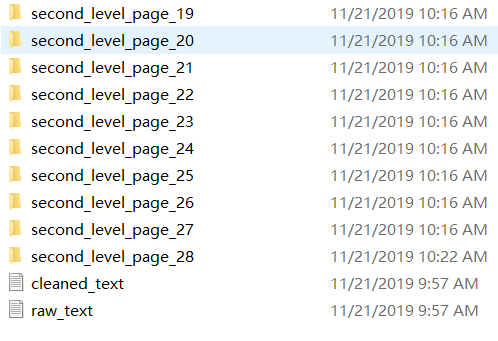
**Project Narrative**

For our project, our group was assigned the UC Davis Stem Portal. Nowadays, anything can be considered STEM. Originally, only topics in the science, technology, engineering, and math areas would be considered as STEM subjects. The definition of STEM also ranges from institution to institution, but we wanted to focus on UC Davis specifically. One example of the expansion of STEM topics in UC Davis is that the economics degree is now considered a STEM degree. With these recent changes, we made our goal for this project to try to understand how UC Davis interprets STEM. At the end of this entire research process, we discovered that while the traditional STEM topics are still all included under the STEM resources, our data visualization also showed a couple of unexpected topics such as business and economics. This shows that what topics and majors UC Davis considers as STEM is gradually expanding.

**Organization, Relationships, and Provenance**

For this process, we needed to use many R libraries, including stringr, rvest, XML, NLP, and tm. The first level we scraped was <https://stem.ucdavis.edu/resources>. From this page, we iterated through 22 pages and created a folder for each individual resource, which came to a grand total of 215 resources/programs. We made each resource into a folder, which was named “STEM\_resources\_program\_(number)” followed by the order the folder was created. We then scraped the text and put it into two separate text files named “cleaned\_text” and “raw\_text”. After this, we went into each resource and scraped the links from them that contained “ucdavis.edu”. We did this so that we would only get sub links that were directly applicable to our research question regarding STEM in UC Davis specifically. We did not want to get any unwanted external links about STEM that were unrelated to UC Davis. As our process continues, we created more folders for every sublink that was found inside the resource folder. Each resource folder now had about 20-30 folders in them, with each folder containing two text files, “**second\_cleaned\_text**” and “**second\_raw\_text**”. At the end of all this, we concluded with 4966 folders and 9932 files. Below is an example of how we organized the files and what our final result looked like.





When we cleaned our text, we also made sure to change everything to lowercase so that everything would be uniform and same words that had an uppercase letter would not be considered a different word as the ones with a lowercase letter. Similarly, we used the removed the punctuation. In our code, we used a nested for loop to accomplish this. We had an outer loop that ran through all of the pages of resources. We then had another loop to get the links inside each of the resources. We would then clean each file inside the loop. Another thing we needed to add in our loops was a try-catch method. We needed this method to avoid any errors or incorrect URLs. Initially, this was difficult because we had so many errors and since there were so many links and pages, there was no way for us to find every single error that our program had encountered, so adding the try-catch method made this a lot easier. We included a blacklist as well to clean out some of the junk strings that we kept seeing repeatedly. We also changed the numbers on the pages into #numeric so that instead of many different numbers showing up in the word frequency, they would all be put into one large category together.

The ReadMe, along with our code and our final presentation can all be located in the shared google drive file under our group name Sweet Cheetos.

**Naming Conventions**

* Variables leading with t: t\_blacklisted, t\_nonNumeric, t\_Punctuation, etc.
  + variables that have to do directly with cleaning the text
* Variables with the word “deeper”
  + indicate which level we were scraping
* Variables with the word “final”
  + storing and writing the second level of our text

**Contributions**

Shawn worked on the majority of the code, as well as the methods portion of the final presentation. Kelley worked on the ReadMe and the results and interpretations portion of the final. Jonathan worked on the ReadMe and the introduction portion of the final presentation.