Data Analysis:

1. When first constructing the main visualizer for my project, I had initially envisioned the data to primarily be generated by the user interacting with the interface. However, I realized that the kind of data that I would get was limited and not particularly useful or insightful. This led me to look for more external datasets to supplement the interactions available to the users. As a result, I explored datasets related to keystrokes, trigrams, bigrams, character frequencies, and the entirety of Wikipedia.
2. An interesting insight I came across was how bigrams make trigrams. It seems obvious that frequent trigrams have frequent bigrams as their substrings, however, how the bigrams overlapped with the trigrams was interesting to me. Particularity when looking at the bigrams "he" and "th" that are substrings for one of the most frequent trigrams "the"
3. A couple of the datasets I was investigating were quite large. The ones containing all the text from Wikipedia took roughly 3 hours to parse and output a corresponding json file. Additionally, there was a bigram keystroke dataset that was very interesting but simply too large to be analyzed (136 million entries). Due to this I ended up focusing more on the wikipedia and the trigram/bigram datasets in my project.

Design Decisions:

Parallel Coordinates for Bigrams/Trigrams:

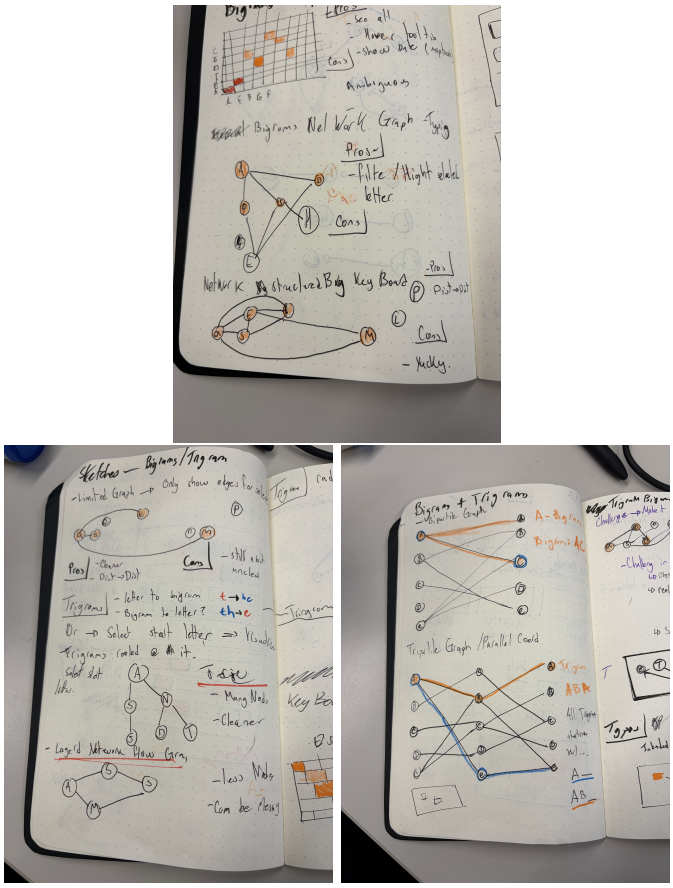

Figure 1: Sketches

One challenge I encountered during this process was figuring out how to visualize the bigrams/trigrams in such a way that maintained the "ordering" of the characters but also enabled the user to interact with specific ones. My initial design consisted of a network graph, but it quickly became evident that due to the shear volume of bigrams and trigrams the edges would make the graph extremely dense. I then tried sketching out more structure visualizations such as trees, and tries but this again lost a lot of clarity due to the number of connections (especially with trigrams). I eventually settled on bipartite/tripartite graphs or a parallel coordinate visualization because this gave a clear ordering to the characters but also allowed me to visualize all the connections without the view becoming overly cluttered.

Preprocessing and adding limits to data retrieval:
As mentioned above, many of the datasets I used were very large and could not be processed / dealt with in the browser. For the Wikipedia text data, I wrote and ran a python script that parsed the content and outputted a json file that could be processed in the browser. For the trigrams, I had initially been rendering all of them, but this was extremely slow, so I ended up capping it at the top 1500 trigrams.

The keyboard is a component with nested components (keys):
When constructing the keyboard visualization, I decided to make each key its own sub-component. This enabled me to create more control interactions between the individual keys rather than the keyboard as a whole. For example, when a key is click, it handles this click event and passes it value up to its parent's parent (the keyboard's parent which is in this case the page body). This is a lot easier to handle than if the keyboard key track of which of its keys was clicked and it enabled me to implement features such as the trigram prefix selection.

Reflection:
When this project was initially announced, I had the idea of presenting it within a github repository. I quickly realized that this was overly ambitious and not defined enough for me to pursue it. I changed to investing character frequencies and visualizing this on a keyboard. Looking back, I wish I had asked for help when trying to decide which project to pursue. I think this would have greatly helped me come up with a clear vision and plan for what I wanted to do for my final project. Even after pivoting, there was still a lot of ambiguity surrounding what my end project would look like. This led to me spending a lot of my time toiling over the details of the project instead of doing a preliminary data analysis and understanding where I stood. Ultimately, I did not effectively make use of the time given for the project and produced a lackluster and incomplete final product. If there is one thing I can take away from this project, it would be to focus less on the "how" of a visualization and put more time into the "why" of the visualization. Effective and meanigul visualization is hard and is something that takes time.