

Frasier-Script – Documentation

by Calvin Kinateder, Terrance Miller, and Morgan Weltzer

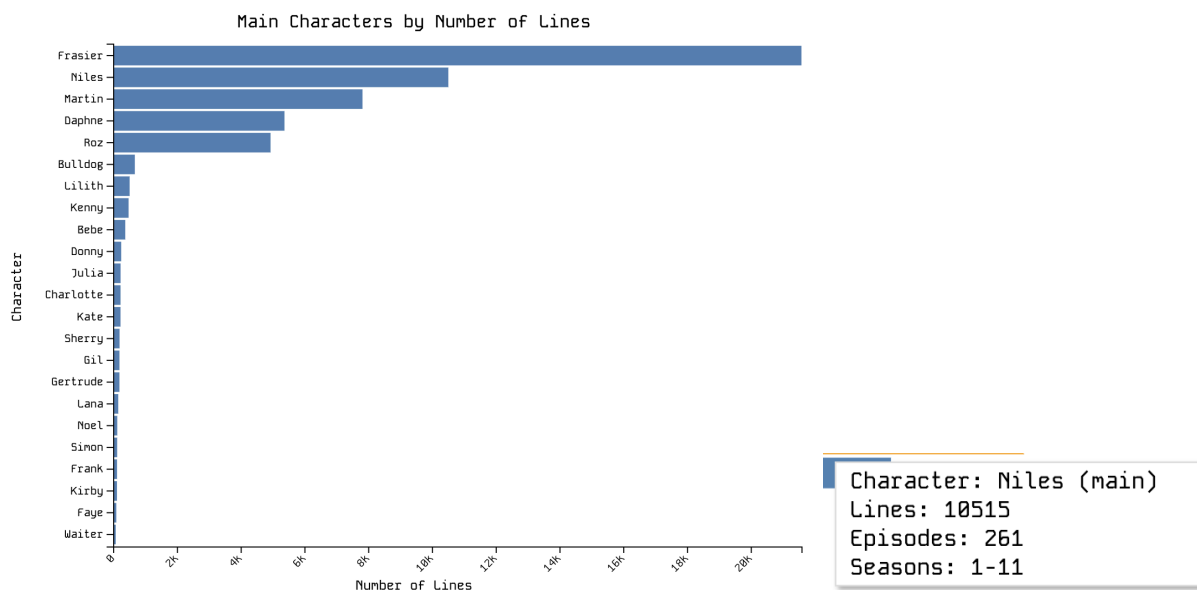
Motivation

This project is designed to let users understand the Frasier show in a less conventional form. However, it is a more fun and interactive way to understand a show and its characters than a typical summary. This visualization focuses on the characters of the show and is narrowed down to their dialogue. It gives users a scope of which characters contribute the most to the shows, and allows you to see which characters who might have made a big impact on you. It is great for both those who have never heard of the show and those who are massive fans.

Visualization Components

There are six main components to this visualization. Each helps the user to make conclusions about the data.

Main Characters by Number of Lines



This bar chart shows the number of lines spoken by each character throughout the whole show. When hovering over each bar, it shows the exact number of spoken lines. Each character is listed on the y-axis and the number of lines is measured through the x-axis in thousands. The user can hover over each bar to get more details on how many lines, seasons, and episodes the character had.

Word Cloud

Character: Season:

Word Cloud for Frasier

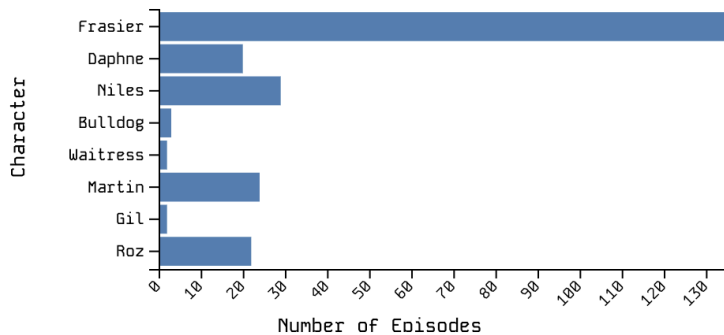


Word: here
Count: 1201
Percentage: 1.89%

This word cloud shows the words said most often by a character. The user can select which season and character to view the data for. Hovering over a point displays more information.

First Lines Spoken

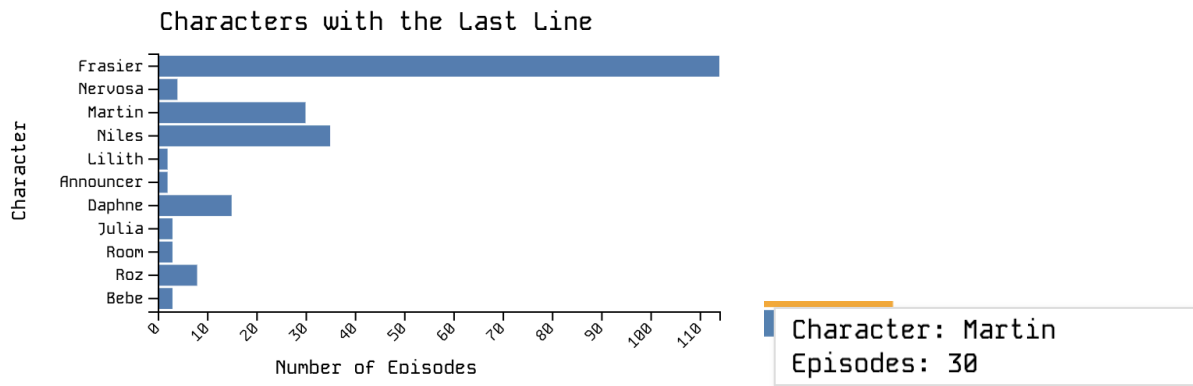
Characters with the First Line



Character: Daphne
Episodes: 20

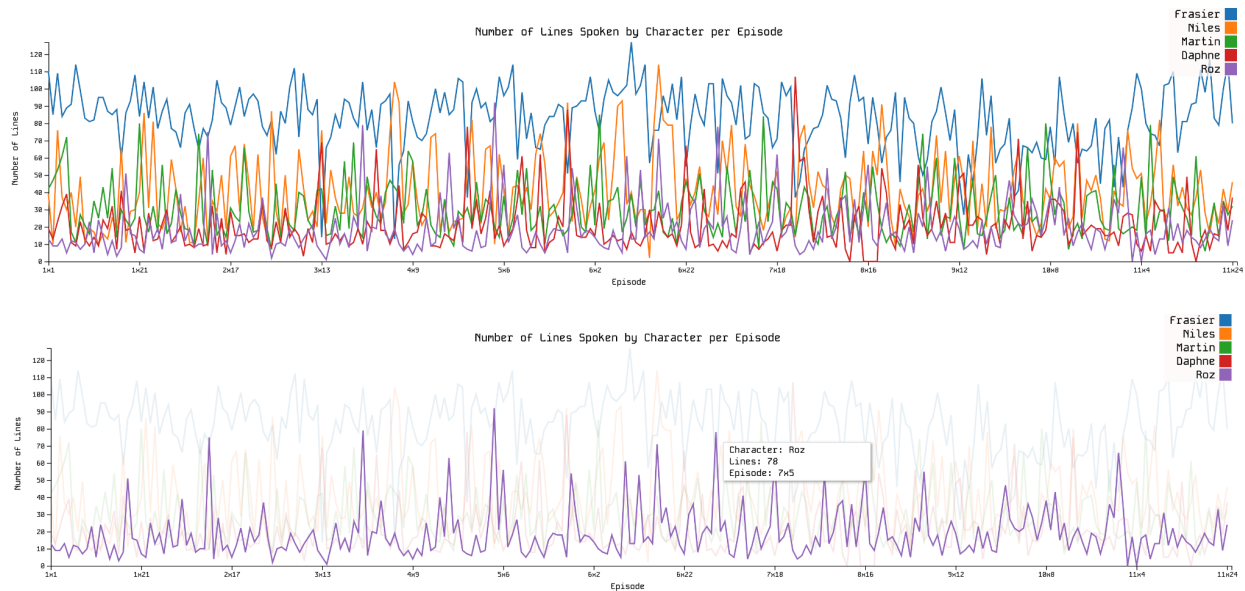
This bar graph shows the number of episodes that each character started. Hovering shows more details.

Last Lines Spoken



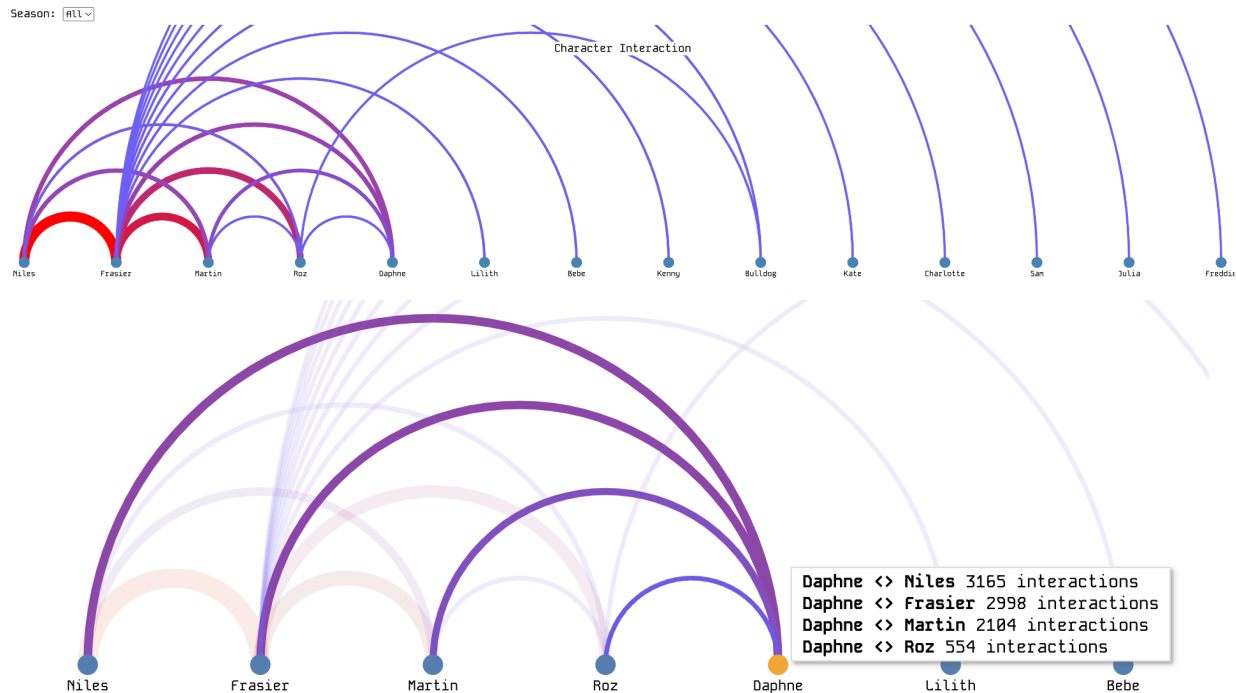
This bar graph shows the number of episodes that each character ended. Hovering shows more details.

Lines Spoken Over Time



This line chart shows the number of lines spoken over time per episode per character. Hovering enables the user to focus on each line individually. Each line is color-coded according to their respective character shown in the legend.

Character Interactions



Lastly, this chart enables the user to discover the number of interactions between each character. Lower numbers have thinner, blue arcs, and higher numbers have thicker, red arcs. Hovering highlights the interactions and displays the exact numbers.

Data

We gathered the data from data.world. This is set up as a very intensive dataset. Each row has a character, episode, season, line spoken, and more. We did very minimal preprocessing – only trimming with a simple python script (shown below), and manually removing a few duplicates.

```
def parse_columns(
    path: str = "transcripts.csv",
    keep: list = [
        "character",
        "charactername",
        "charactername",
        "charactername",
        "gender",
        "title",
        "lines",
        "season",
        "episode",
    ],
) -> pd.DataFrame:
    """Load a csv file and keep only the columns we need
```

```

    Args:
        path (str, optional): input path. Defaults to "transcripts.csv".
        keep (list, optional): to keep. Defaults to [ "character", "charactername",
"character", "gender", "title", "lines", ].

    Returns:
        pd.DataFrame: cleaned
    """
    df = pd.read_csv(path)
    df = df[keep]

    # remove duplicate lines
    df = df.drop_duplicates()

    return df

if __name__ == "__main__":
    df = parse_columns()
    print(df.head())
    print(df.info())
    print(df.describe())
    df.to_csv("small.csv", index=False)

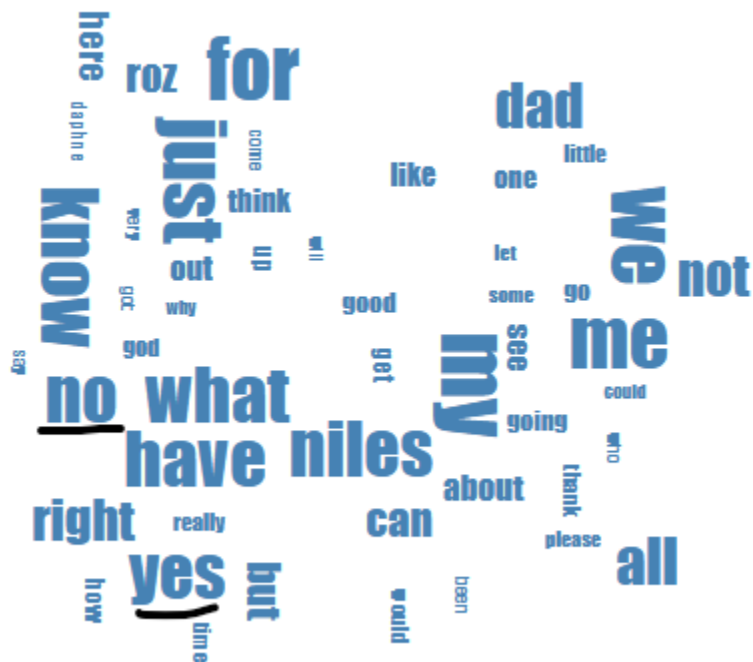
```

In the code, we do some minimal sorting and processing to filter by season, episode, and/or character.

Discoveries

One cool discovery we made was through the word cloud. Every side character has a drastically higher no to yes ratio than the main character, Frasier. This makes sense, as he drives the plot, and the side characters are often there to create conflict for the main character.

Frasier's Word Cloud



Niles's Word Cloud

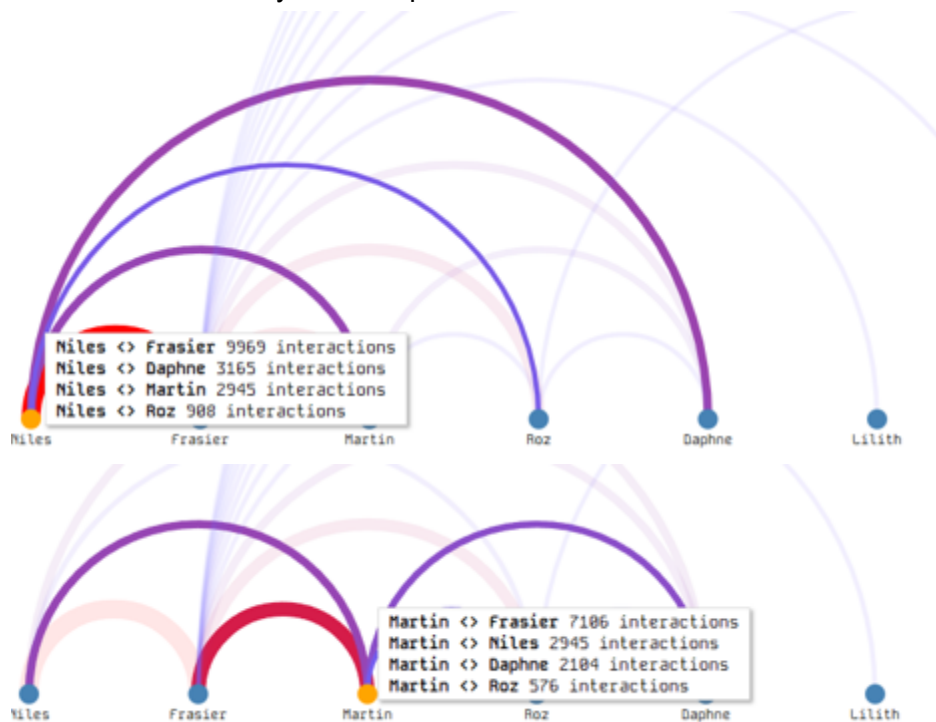


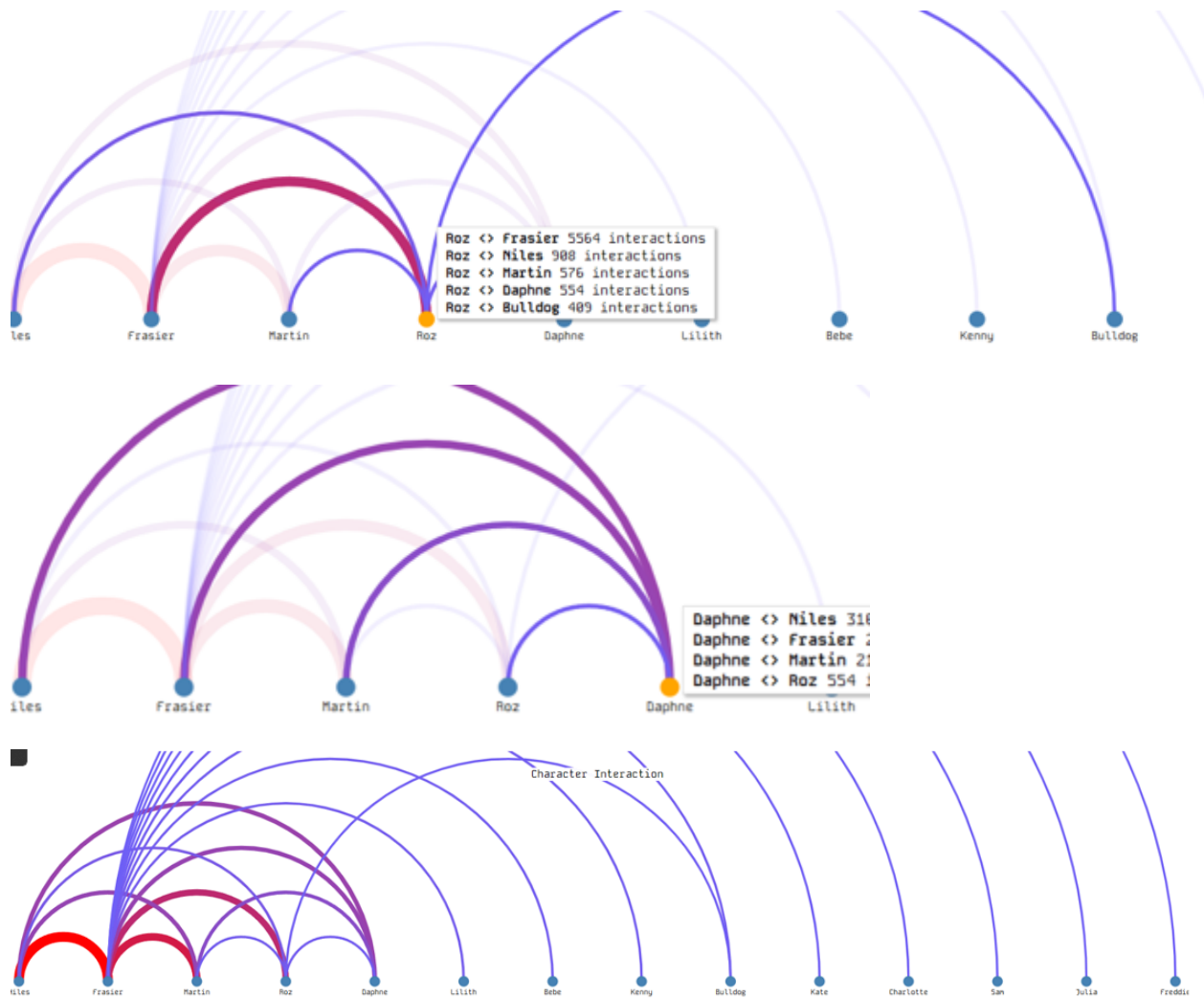
Martin's Word Cloud



The word “yes” does not even appear on Niles’ word cloud.

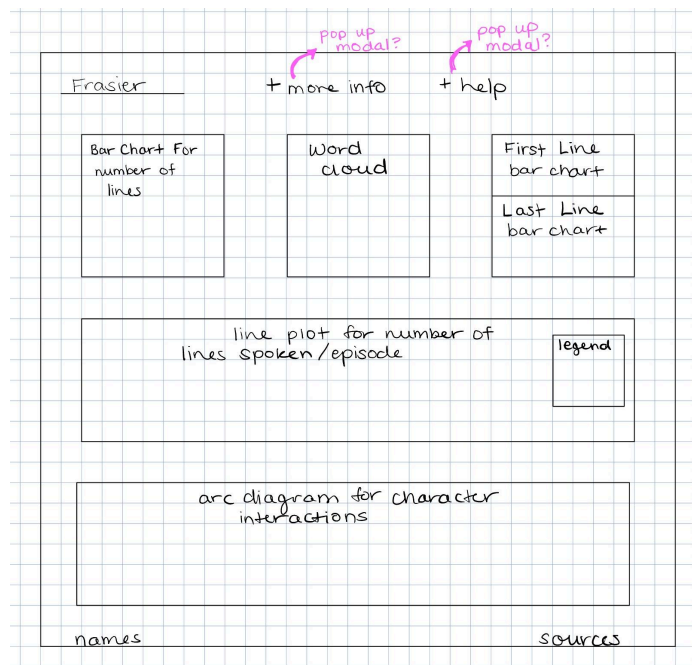
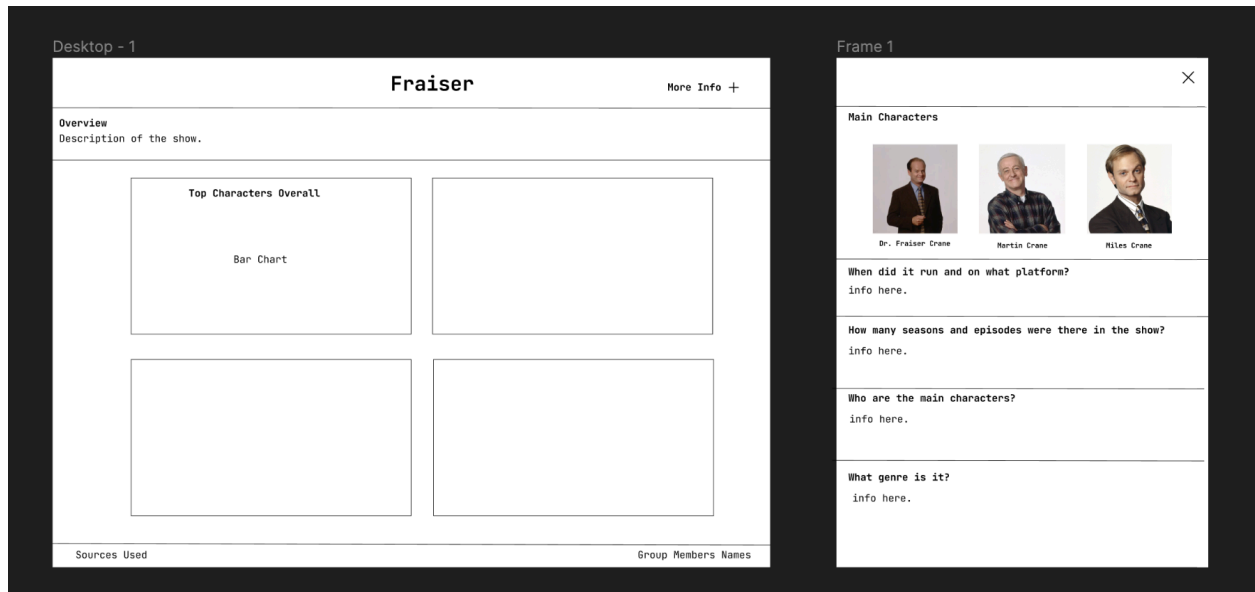
Another cool discovery we made through our application is that most of the main characters don't interact with anyone except Frasier.





Process

We started with an initial mockup (shown below).



Obviously, our final product did not exactly resemble the mockup, but we kept with the multi-panel design and expandable info sections.

We used d3.js to build the charts. We structured the code in an object oriented fashion, with a separate class for each chart. No classes were reused. We used a lot of global variables to handle static variables. Because of the way we set up the selection dropdowns, no brushing or linking was needed, so any selections made are kept within their selected class.

The webpage is viewable [here](#), and code [here](#).

Demo

Demo is linked [here](#).

Contributions

Each team member played an integral part in completing the project.

Calvin

- Code setup
- Character bar chart
- Timeline chart
- WordCloud
- Hosting setup
- Documentation

Terrance

- First lines spoken bar chart
- Last lines spoken bar chart
- Word cloud
- Character arc chart

Morgan

- Demo video
- Mockups
- Documentation

For documentation: assume that someone is encountering your project for the first time. I strongly suggest having the documentation be publicly available on each team members personal webpage. The content for each team member can be identical, but it is important for each student to have a record of the project. You only need to submit one, for credit. Screenshots are an important tool for communicating your work to others. Including screenshots will improve your documentation grade.

- Explain the motivation for your application. What can it allow someone to understand?
- 1 section on the data: Describe the data and include a link.
 - *** Since you will have done more work for this project to obtain and process the data, describe your methods and include processing script
- 1 section on the visualization components: Explain each view of the data, the GUI, etc. Explain how you can interact with your application, and how the views update in response to these interactions.
- Include design sketches and design justifications
- 1 section on what your application enables you to discover: Present some findings you arrive at with your application. Include screen shots to illustrate.
- 1 section on your process- what libraries did you use? How did you structure your code? How can you access it and run it? Link to your code (and the live application, if it is deployed online).
- Include a 2-3 minute demo video, showing your application in action. The easiest way to record this is with a screen capture tool, which also captures audio- such as Quicktime. Use a voiceover or video captions to explain your application. Demo videos should be sufficient on their own, but can reference your documentation. Include the name of the project, your name, the project components, and how your application works. You can present it on your webpage or on youtube, but linked on your webpage.
- Document who on your team did which component of the project. Ex. If someone worked on the data, and on bar charts, list their effort on these components.