

Data Visualization Project 3: TV Time

Group Members

Chris Laney, Yale Miller, Prabh Singh, Jake Huseman and Caleb Hendrix

Project Topic and Motivation:

Our team chose the TV show SpongeBob SquarePants which first aired on Nickelodeon in 1999 and is still releasing new episodes today. We made this decision mainly due to multiple team members having enjoyed the show while growing up, the cultural impact of this show on many people as well as the series including many memorable recurring characters. These characters interact in many settings and engage in unique exchanges of dialogue which makes for some very interesting visualization potential.

Data Collection and Processing:

The data for this project was collected in two different ways. The first was the collection of the transcripts for each individual episode and the second was collecting the list of episodes for each season. This was done simply due to what data was available online.

Episode Transcripts:

<https://www.kaggle.com/datasets/mikhailgaerlan/spongebob-squarepants-completed-transcripts?resource=download>

Sample Episode List: <https://thetvdb.com/series/spongebob-squarepants/seasons/official/1>

The list of episodes from each season was scraped from the website listed above by using the IMPORTHTML function in google sheets. These imported lists were then compiled into a single master csv file which listed all of the episodes with their season, episode number, title and airdate. This csv was used to match against the episode transcript files and then create the final csv data files with all the needed information.

This process was accomplished with a short python script which read in all of the transcript files obtained from the link above and processed them into a csv format. Using all of this information the following csv header list was arrived at:

Id, speaker, speech, context, raw-dialouge, season, episode, title, airdate, runtime

Each line of dialogue has a unique id for various purposes. The python script identified the speaker of the dialogue, what words that speaker said and then also any context clues that were included in that line. The transcripts often included bracketed context clues within the speech which indicated things like the speakers emotions or actions in the scene. These were


```

        speaker = ''.join(re.split(r'\[.*?\]', speaker))
        dialouge = line.split(":", 1)[1].strip()
        context = ' '.join(re.findall(r'\[(.*?)\]', dialouge))
        speech = ''.join(re.split(r'\[.*?\]', dialouge))
        id += 1
        data.append([str(id), speaker, speech, context, dialouge,
episode_data["season"], episode_data["episode"], episode_data["title"],
episode_data["airdate"], episode_data["runtime"]])

    with open("complete.csv", 'w', newline='') as csv_file:
        csv_writer = csv.writer(csv_file)
        csv_writer.writerow(headers)
        csv_writer.writerows(data)

    seasons = {
        '01': [],
        '02': [],
        '03': [],
        '04': [],
        '05': [],
        '06': [],
        '07': [],
        '08': [],
        '09': [],
        '10': [],
    }
    for entry in data:
        seasons[entry[5]].append(entry)

    for key, value in seasons.items():
        with open(f"season{key}.csv", 'w', newline='') as csv_file:
            csv_writer = csv.writer(csv_file)
            csv_writer.writerow(headers)
            csv_writer.writerows(value)

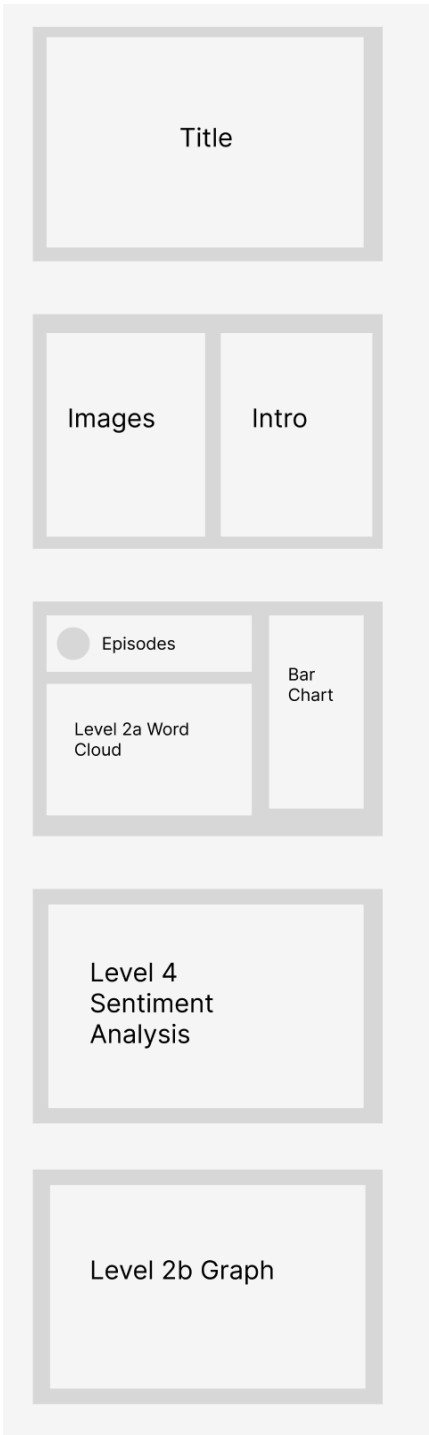
except Exception as e:
    print(f"An error occurred: {e}")

```

The above method will read in the episodes dictionary that was just created and then loop through each episode transcript file. If an episode name is matched then that episode data is used to write the csv. The speaker, speech and context are parsed from the transcript for each line and then added to a list of data to be written. Both a complete csv with every episode's data and individual season csv files are written with this script.

Visualization Components and Layout:


Initial Sketch -




Our visualizations start off with our title page, we wanted to have a strong theme tied to the show:




Next we wanted to include a page which would introduce our audience to the data being used. If someone was unfamiliar with the show it would be important for them to have some context and look through the important characters:




SpongeBob




Patrick




Mr. Krabs




Gary




Squidward




Plankton




Sandy




Mrs. Puff




Pearl




Karen




Flying Dutchman




Mermaid Man




Narrator




Larry



Harold



Nat



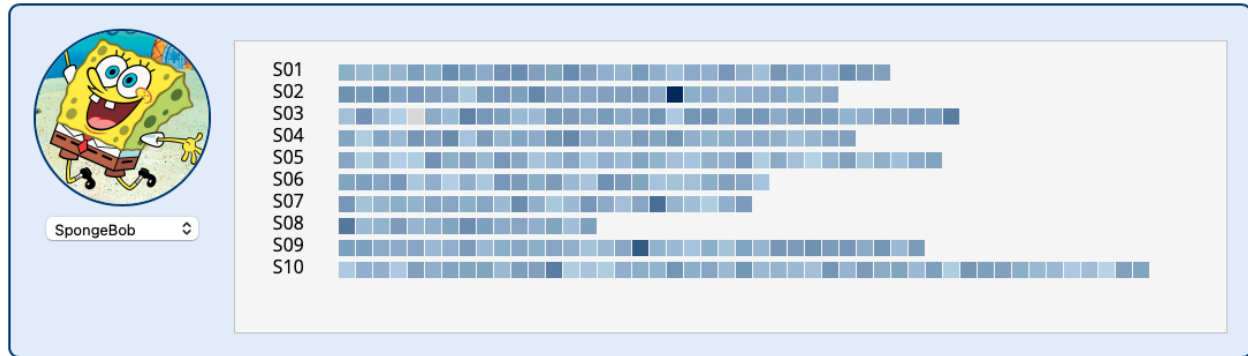
Barnacle Boy

Areee Ya Ready Kids?

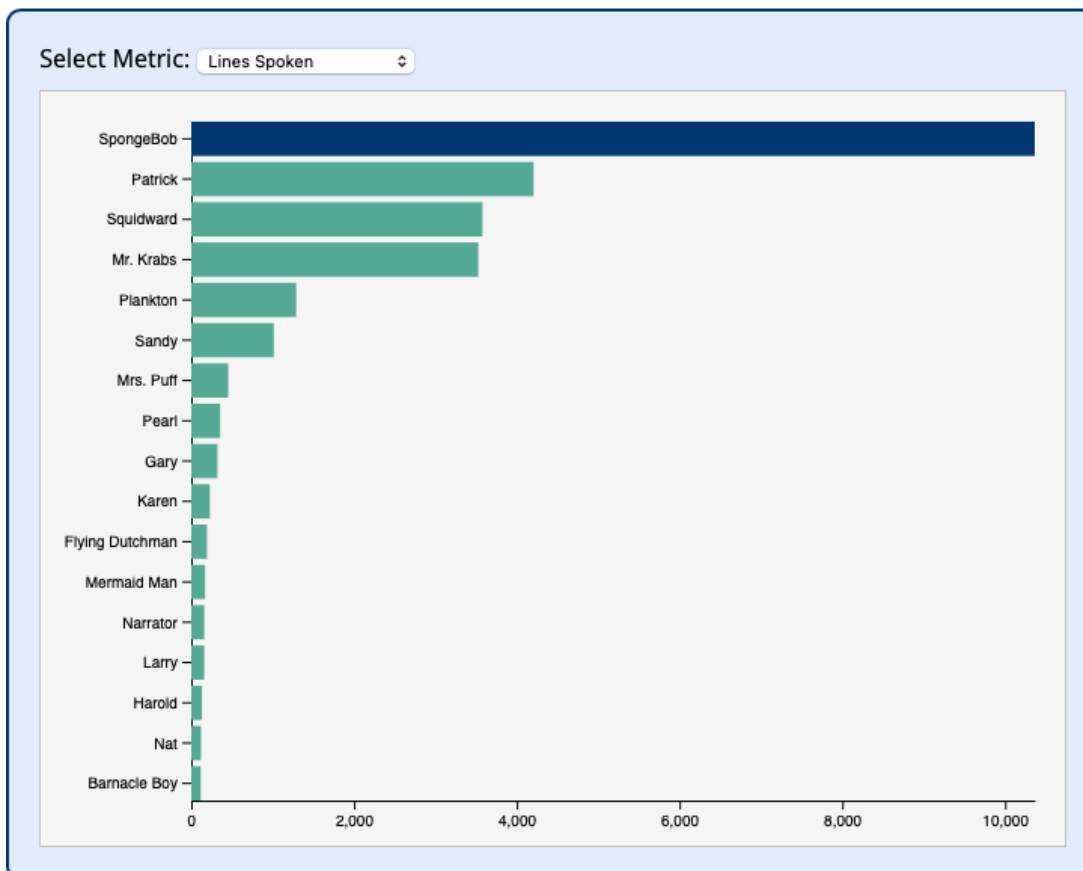
SpongeBob SquarePants is one of the most popular animated shows to ever exist. Created by Nickelodeon, the show features a zany crew of aquatic characters. This visualization explores the top 17. Keep scrolling to get started!

17	10	262
Characters	Seasons	Episodes

The purpose of our first visualization is to allow a user to explore line metrics for each of the characters. We wanted to allow someone to select a character and see how many lines were spoken by the character across the different episodes and seasons. A user can hover on an episode so that they can see the title and the number of lines spoken:



As an additional view for this character selection, we wanted to show some broader metrics for all the characters in comparison to the selected character. In this bar graph, one can see the selected character highlighted and compare their total lines spoken to the lines spoken of the other characters. If the user is not interested in the lines spoken metric, they could also choose to see the number of episodes appeared in or words spoken metrics using the drop down menu:



A scatter plot showing the performance of various Atari games over 300 episodes. The Y-axis represents 'Lines' (0 to 10,000) and the X-axis represents 'Episodes' (0 to 300). The data points are represented by small circular icons of the game covers. The plot shows a general upward trend in performance over episodes, with some games reaching high scores (up to 10,000 lines) and others plateauing at lower scores (around 4,000 lines).

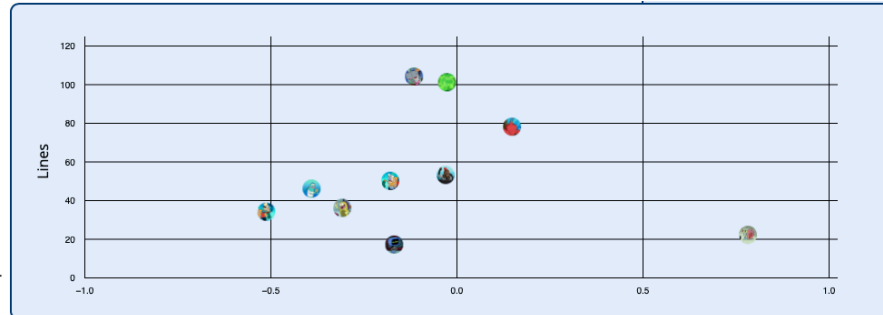
Game (Icon)	Episodes (X)	Lines (Y)
Seaquest	0	~500
Enduro	~10	~500
Private Eye	~20	~500
Robotank	~30	~500
Seaquest	~40	~500
Enduro	~50	~500
Robotank	~60	~1,500
Seaquest	~70	~1,200
Enduro	~80	~500
Robotank	~90	~500
Seaquest	~100	~500
Enduro	~200	~3,500
Robotank	~210	~4,200
Seaquest	~230	~3,500
Enduro	~300	~10,000

For a special visualization we thought it would be interesting to see the calculated sentiment of the lines spoken between the characters of the show and the main character SpongeBob. We used some sentiment analysis and machine learning tools to create these sentiment findings and then plotted the characters sentiments:

Spongey Sentiment

SpongeBob has a bubbly personality, but not all characters in the series show him the same amount of love.

We used DistilBERT and other Machine Learning techniques to analyze the **sentiment** of each character when interacting with SpongeBob. Check out our results [here!](#)



Throughout our visualizations we wanted to make the views large enough and in a meaningful arrangement. Because of this we chose to make our application span multiple pages instead of trying to get everything onto a single dashboard. This way we could introduce the data and then walk a user through the visualizations for each character. We also chose to stay on theme with the ocean and made the color theming and color gradients all shades of blue and light green.

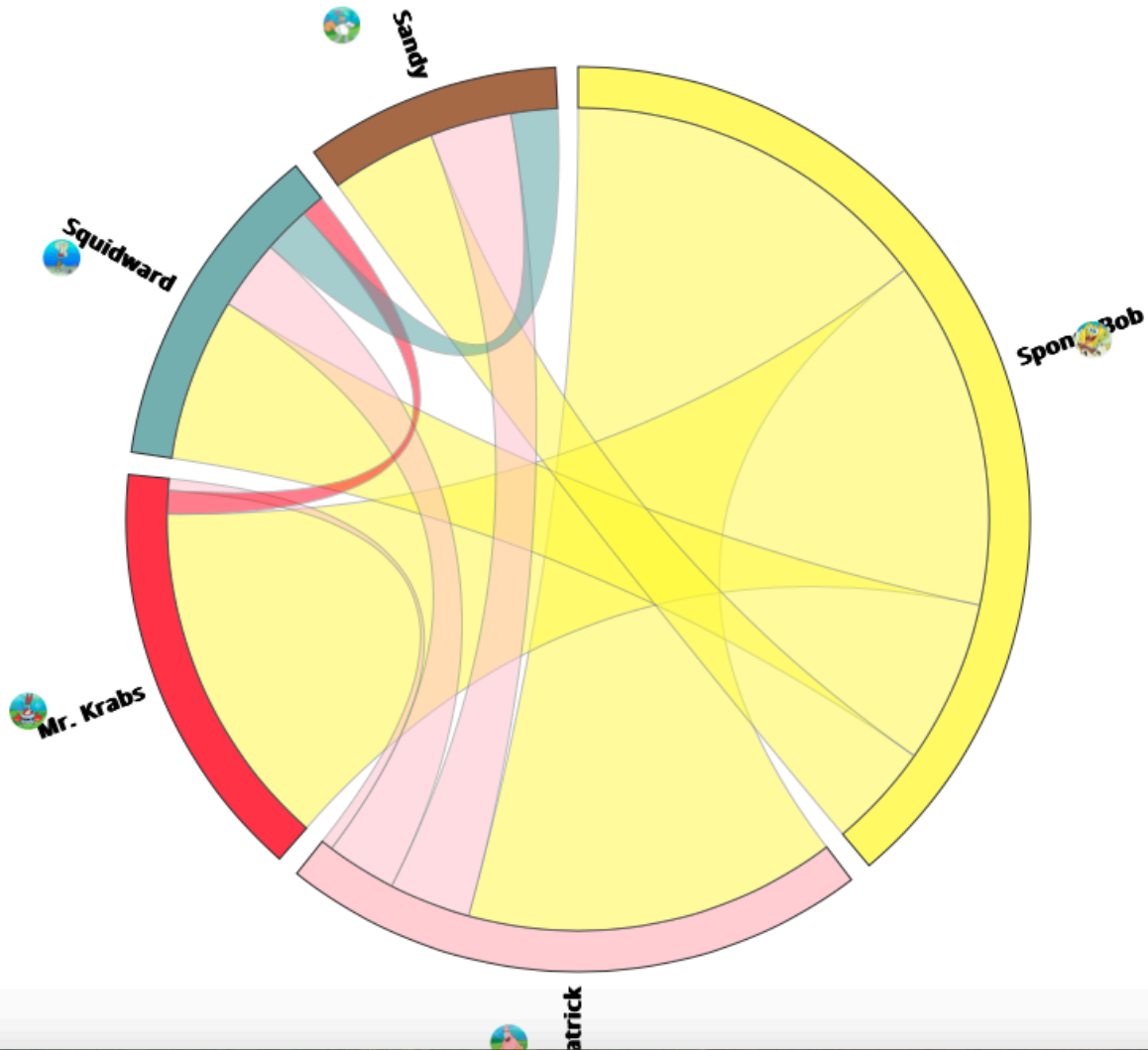
Finally, we created a visualization for showing the interactions between characters in the show on a per episode basis. The connections between characters are weighted by the number of interactions and a tooltip can be displayed when hovering the connections to show this number:

SpongeBob Character Interactions

Season 1, Episode 14

◀ Prev

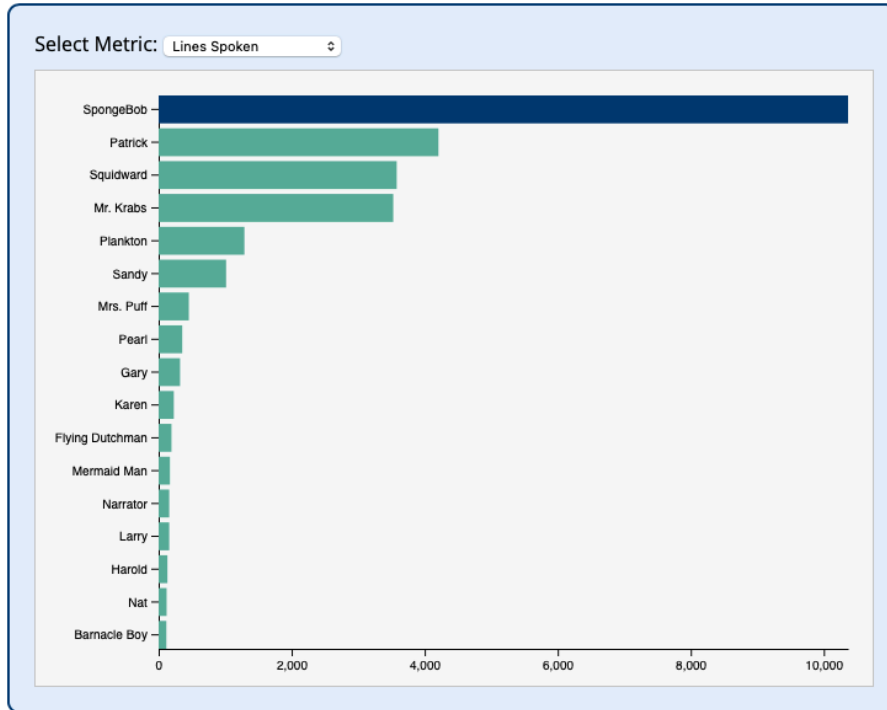
Next ▶



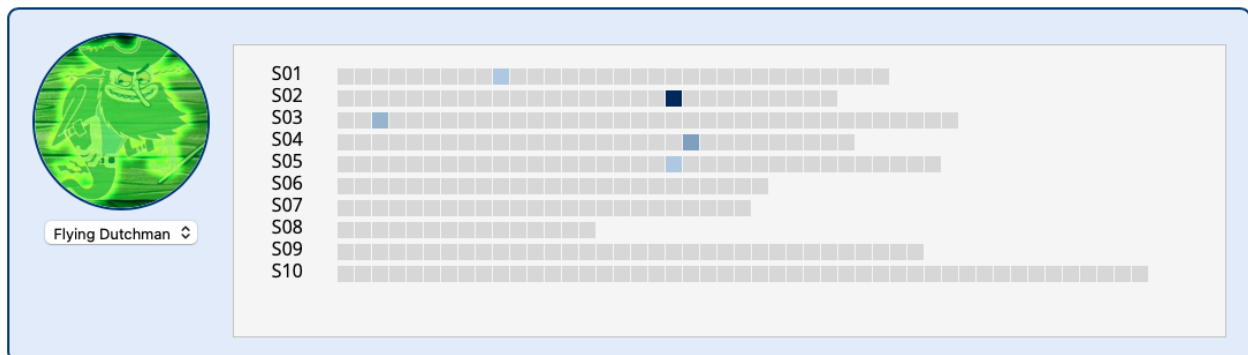
Discoveries:

Our application makes it very easy to make discoveries about the lines spoken by the characters, the types of words used by a character and those characters interactions.

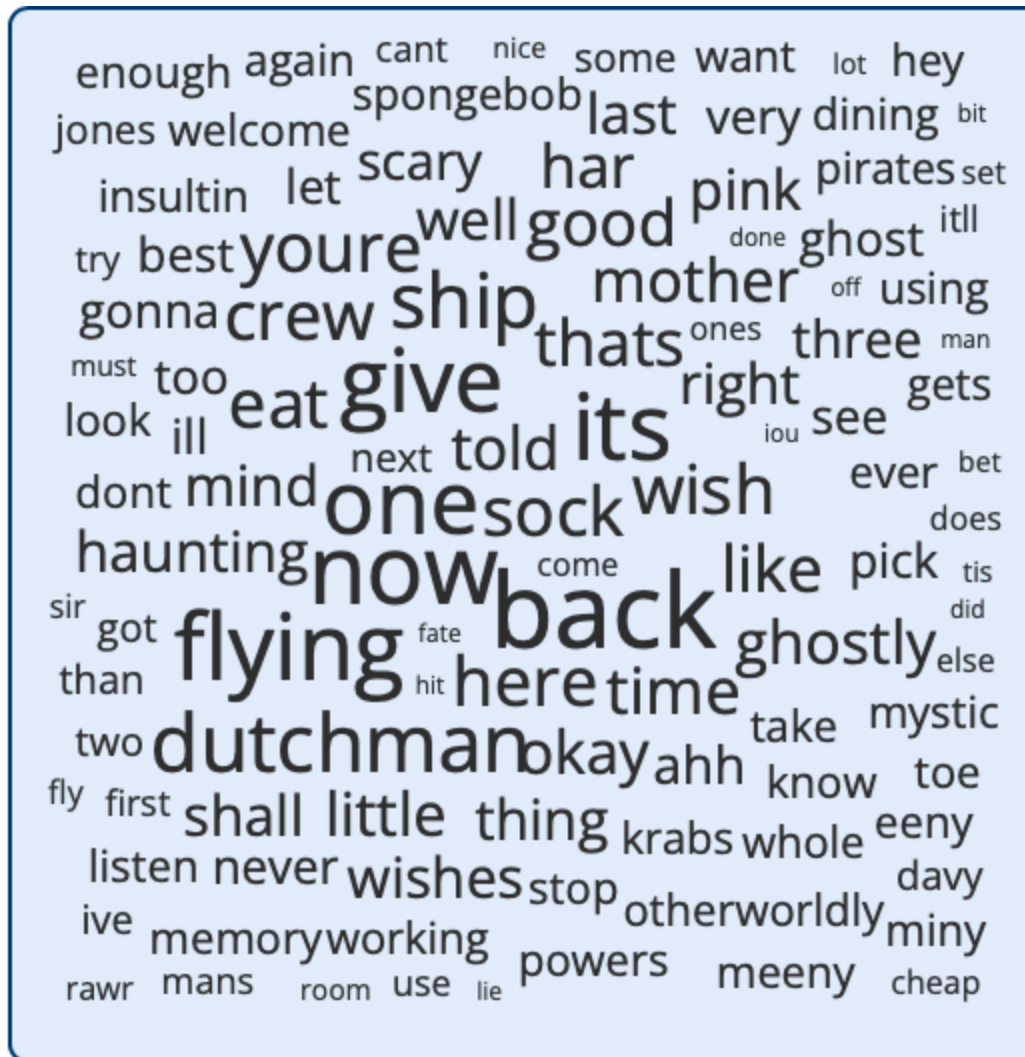
For instance, one can clearly see that Spongebob has the most lines out of any other character:



For lesser known characters such as the Flying Dutchman our application allows one to see exactly how often and in what episodes they appear in:



If someone was really into the Flying Dutchman they may also want to know what words they said most in the show. Our word cloud makes it easy to see this and we can see for this character that they say their name often and things like ghostly and ship:

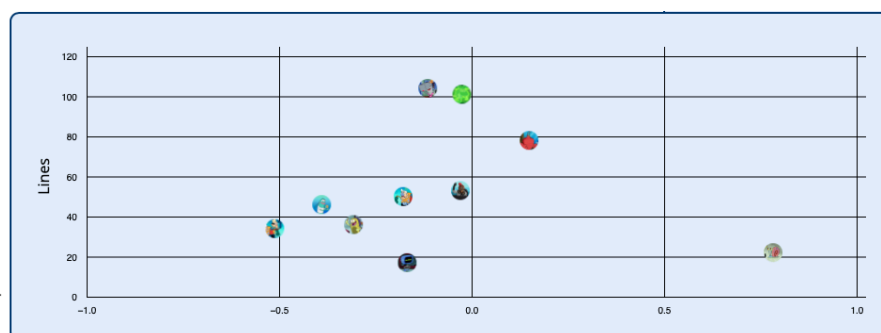


If someone was interested in what the sentiment was between the main character of the show and other characters then they could look at the sentiment analysis visualization and see that the character Gary has the most positive sentiment to SpongeBob. This makes sense as Gary is his pet:

Spongey Sentiment

SpongeBob has a bubbly personality, but not all characters in the series show him the same amount of love.

We used DistilBERT and other Machine Learning techniques to analyze the **sentiment** of each character when interacting with SpongeBob. Check out our results here!



Process:

Project Code Repo: <https://github.com/chendrix99/DataVisProject3>

Hosted Site: <https://chendrix99.github.io/DataVisProject3/>

Our project can be accessed from the above link to the hosted GitHub pages version or the code can be cloned and then built locally in VS Code.

For the data processing we used simple libraries like python's csv, os and re libraries for cleaning the data and writing to the csv files. For the sentiment analysis processing the python libraries DistilBERT were used for analysis.

The code was structured similarly to previous projects with a separate data, js, and css folders and a main index.html at the root.

Demo:

@ 📺 Screen Recording 2025-04-25 at 3.26.01 PM.mov

Contributions:

Chris Laney - Implementation of level 3 goals including the network visualization.

Yale Miller - Implementation of level 1 goals and also initial design sketches.

Prabh Singh - Implementation of level 2 goals including the word cloud visualization.

Jake Huseman - Implementation of level 4 goals including the sentiment analysis visualization.

Caleb Hendrix - Data collection and preprocessing as well as the documentation requirements.