FRIENDS: The One With The InfoVis

Catarina Rodrigues
Instituto Superior Técnico
Lisbon, Portugal
catarina.rebelo.rodrigues
@tecnico.ulisboa.pt

Daniela Mendes
Instituto Superior Técnico
Lisbon, Portugal
daniela.mendes
@tecnico.ulisboa.pt

Vasco Pires
Instituto Superior Técnico
Lisbon, Portugal
vascocfpires
@tecnico.ulisboa.pt

ABSTRACT

UPDATED—15 November 2021. This paper describes a visualization dashboard regarding the infamous 90s sitcom 'Friends'. Our goal is to provide the fans of the show some answers on the six beloved main characters, aiming to expose new insights that can help users analyze the periods of the show. The final dashboard consists of an interconnected set of idioms: a Barchart, a Network graph, a Wordcloud and a Boxplot constructed in d3.js as an Information Visualization course project.

INTRODUCTION

Friends is an American television sitcom created by David Crane and Marta Kauffman, which aired from 1994 to 2004, lasting ten seasons. The show revolves around six friends in their twenties and thirties living in New York City: Chandler, Joey, Monica, Phoebe, Rachel and Ross. Friends received a lot of acclaim throughout its run, becoming one of the most popular television shows of all time [1].

As fans of the show ourselves we were eager to dissect the show and its characters a bit further. In addition, as many fans of the show, we utilize Friends as a comfort show and are constantly rewatching episodes. We idealized a tool that would help users explore specific periods of the show and maybe even help with picking the next season or episode to watch. Being the target audience made us passionate to find solutions for situations that were dear to us and satisfy curiosities we either couldn't otherwise with the resources we had come across or that we felt we could have something to add to. In order to try to avoid some of our biases we spoke to a small sample of fans of the show, relying on some user research to settle on a few ideas that were considered relevant and that we felt confident we could find data and techniques to help with bringing the concept to life. With this project we want to give fans the ability to explore and analyze the main characters, specifically their relationships and emotions, as well as some correlations with the show's ratings and the technical team involved. We found these themes seemed to be what the fans we came across were most interested in. We settled on a few questions that would help fans with their needs and that we were committed to answer:

- Who are the most active characters in a given time range?
- For a specific character, which other character do they interact with most within a given time period?
- What is the general audience's preferred episode in a given period?
- What are the most common emotions throughout the show?
- How do the episode ratings vary among the show's directors?

The title "Friends: The One with the InfoVis" came up as a pun but quickly grew on us, being a bit of a fun inside joke around the show's titles. It stuck around. Making use of the d3.js version 7 package and the knowledge acquired in the Information Visualization course, we crafted what we believe would be the best solution given our resources and skills.

RELATED WORK

During our research phase we came across interesting visualizations with related problem domains. We got inspired by some websites [2,3] as they answered some interesting questions. For example, one of the websites displayed how many scenes each character was in, consisting of showing the number of scenes in which each character appeared throughout the ten seasons, either solo or with another selected character. Yet, in this case no type of idiom was provided, as we could only click each character to reveal the number, with no ways of visualizing and comparing each character's data and no way of filtering this data through specific periods of the show. The website also contains a barchart encoding the number of lines spoken between each character pair throughout the whole show run. We got really inspired by it as both we and a few other fans thought it could be extremely relevant for our problem domain, and immediately thought of ways we could elevate this concept. For example, we felt some sort of node-link based idiom could really elevate the concept, as well as the ability to filter the various show's time periods [2]. Another website displayed the most talkative characters during the show and even how close each friend is to others, but the idiom choices lacked ways to interact and a few of them seemed to be outright poor choices when we take into account the theoretical concepts learned throughout the course [3]. We believed our data could be presented in a different way, making use of some unexplored and even more relevant metrics (according to ours and several fans' needs) and more fitting idioms, providing ways to filter, interact and allowing the exploration and analysis to be more accessible and resourceful, as well as easier to grasp.

We were also inspired by some of the dashboard examples displayed in the Hall of Fame [4], namely the projects "Trump's Tweets" specifically the idea of having a

personalized podium for mentioned accounts (which we felt could be great for comparisons of the characters presence) and "Game Of Thrones" which inspired us to work on the realm of a tv show, yet in our case we decided it would be best fit for our problem domain to dive deeper into our characters' dynamics, with Friends having essentially six main characters, all very interconnected, as well as tapping a bit into the show's success and technical information, as these were the needs we identified.

THE DATA

Upon researching we came across a dataset consisting of a collection of all the conversations over the 10 seasons of the sitcom, made available on Github by Emory University and their Natural Language Processing team [5]. This dataset consisted of 10 JSON files, each one referent to one of the seasons, with each entry referring to a script's line, having data organized in a hierarchical tree structure, giving us access to information such as the line's speakers, transcript, the episode and season it is inserted in as well as the associated emotion of the transcript.

In order to answer some of the questions, we correlated the characters' line analysis with some of the respective episodes technical details, making use of another dataset, consisting in a single CSV file released on Kaggle by Mohammad Reza Ghari and Moulik Dhade, titled "Friends Series Dataset" [6], containing technical information for each of the 236 episodes from which we could retrieve data such as each episode's rating (populated from IMDB) or the episode's director.

Making use of the Python Pandas library we processed these 11 files, first discarding some attributes that would be of no use for our solution and renaming others for organization purposes. We ended up retrieving some information from IMDB ourselves, in order to fix some missing values from the technical dataset. We also generated some derived measures that would provide the needed data to craft the chosen idioms, such as an attribute that consists in the number of words spoken by each of the main characters in a given episode, which determines the length of the bars of one of our idioms, or an attribute consisting in the number of verbal interactions between each pair of main characters in a given season that determines the width of the lines of another idiom, among others.

We also decided that, for our problem domain, it would be best to discard the lines spoken by characters not contained in the main group, as our focus was set to be mainly around the six main characters. This way not only can we focus on what really mattered to the majority of fans but, also, we can handle some of the possible scalability issues.

In addition, decisions regarding the emotion category granularity level that we chose to work with were made having our solution in mind. Had we chosen to further segregate the emotion categories we could have run into scalability issues, as the value associated with the categorical emotions is encoded in our solution making use of size. By choosing to work with a small number of categories we prevent present (and future) scalability issues. For our final dashboard we also handled the fact that we have a somewhat large number of directors through the entire show by making use of interaction (either allowing the use of a scrollbar in the referent idiom or further filtering the subset by selecting fewer seasons to analyze). Having more directors could generate issues, as the user could have to potentially scroll a lot more, but, as the show has officially come to an end, we decided we didn't have to worry or plan for this hypothesis, as it wouldn't seem to ever become an actual possibility.

On the same note, we made the decision to work with ways to filter specific periods of the show. In this sense we ruled that filtering the data by season would consist of the perfect granularity level to be able to answer the proposed questions as well as settle for idioms that would better account for scalability issues.

The end of our initial data processing stage culminated in 2 CSV files. A few challenges worth mentioning would be the high dimension of the script line's dataset, as it wouldn't be wise to display 10 seasons worth of lines as individual entries in any idiom, paired with having the file organized in a hierarchical tree-like structure, which we hadn't had prior experience with. For this we worked on finding a way to process this file into a structure we would be more comfortable working with. In addition, one of our biggest setbacks was related to some surprise missing values for the emotion attribute (an attribute containing the emotion associated with each script line) which ended up compromising the faithfulness of the remaining attribute's data, invalidating its use. As this was an attribute we didn't want to let go of, since it allowed to answer one of our favorite questions, and even though we had no prior experience with sentimental analysis, we found and made use of a Python package named Text2Emotion [7], learning just enough to take our existing data and, from the script's output, compute derived values that allowed us to answer our question.

In this case, and in a few others at the time of crafting the dashboard, for quicker processing, data relating to a few visualization idioms was then isolated into separate files where its structure was more conducive to the generation of the idiom in question.

VISUALIZATION

Overall Description

The final dashboard can be found in Figure 1. It consists of a linked visualization of 4 different idioms: a Barchart, a Network graph, a Wordcloud and a Boxplot, with the ability to filter data using a set of buttons. In the following subsections we will go into detail on the choices that lead us to this specific solution and the reasoning behind them.

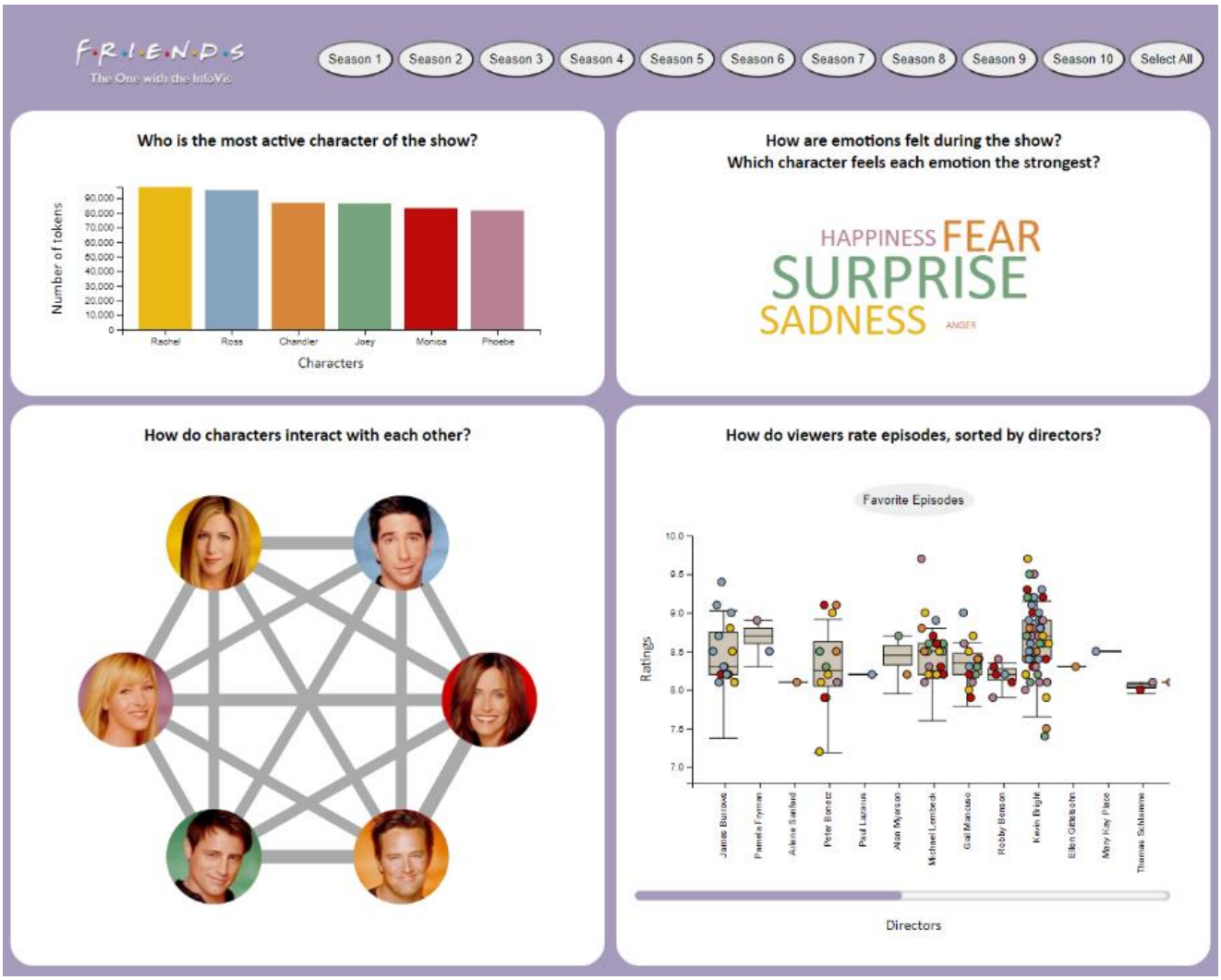

Figure 1. Overview of the dashboard

In our dashboard each main character is associated with a color. The color code is consistent throughout the idioms and is made clear through both the Barchart idiom, having each name under the designated color, as well as the Network graph, showing the characters photo on their respective colored node. Data referent to a specific character can be selected on each of the several idioms with a click, leaving only their associated color visible, setting all other colors to gray, in order to make it more salient and facilitate analysis.

Figure 2. Overview of the filtering set of buttons

Starting from the top of our dashboard, we find the visualization's logo followed by a set of buttons allowing the user to filter the data through seasons, in order to provide a way for the analysis and exploration of specific periods of the show, essentially filtering the data subset used by the several idioms. Users can click each button to either select the season (if it wasn't previously), in this case adding its data to the subset used or deselect it (if it was previously selected). An additional button denominated "Select All" allows the user to easily select data from all the seasons. To minimize confusion and avoid having empty idioms, the initial state has, as default, all seasons selected.

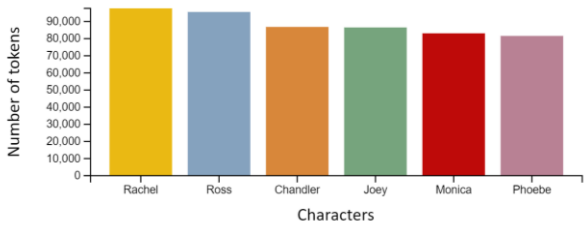

Figure 3. Overview of the Barchart idiom

Below the filter buttons, on the left, we can find a Barchart idiom, as illustrated in Figure 3, which compares each of the six main characters' presence throughout the selected period of the show.

Each bar is referent to a different character, colored with its respective hue. The length of each bar encodes the character's

presence in the selected period. For our idiom we consider presence as the sum of the tokens (the result of the tokenization process of the various spoken lines, with each token referring to a spoken word) for the respective character throughout the period in question.

The Barchart idiom is always set to ensure the reordering of the character's bars in a descending scheme, from left to right, even when we change the subset, allowing for easier comparisons.

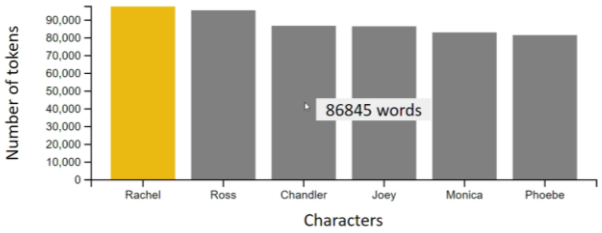

Figure 4. Hovering a bar on the Barchart idiom, having one of the characters selected

Clicking a bar will select the character's data in the several idioms, greying out the remaining colors. Hovering over a bar displays a tooltip containing the number of words spoken by the respective character as illustrated in Figure 4.

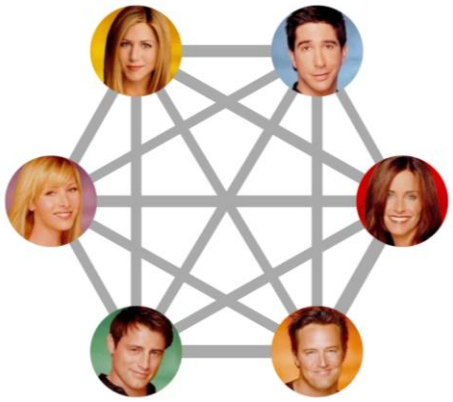

Figure 5. Overview of the Network Graph idiom

Below the Barchart, we can find a Network Graph idiom, illustrated in Figure 5, that allows the users to learn more about how much the six main characters interact with each other. We crafted this idiom in order to allow users to analyze the verbal interactions count of any pair of characters of the main group on a selected period, having verbal interactions essentially meaning that a pair of characters both had lines in a scene together.

Each node circle corresponds to a different character, being colored accordingly, as well as displaying the character's photo. Each of the edge lines connects two characters, having its stroke width proportional to the count of verbal interactions between the two characters it connects.

Clicking a specific character node will select the character's data in the several idioms, greying out the remaining colors and photos. As illustrated in Figure 6, hovering an edge line displays a tooltip in which we can consult the amount of verbal interactions that a pair of characters has had on the selected period, as well as their names.

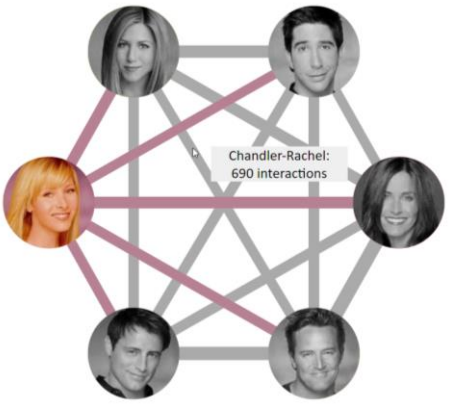

Figure 6. Hovering over a line in the Network Graph idiom

Figure 7. Overview of the Wordcloud idiom

On the right side of the screen, under the seasons' filter, we can find a Wordcloud idiom, illustrated in Figure 7, displaying the distribution of the characters emotions throughout the script, in the selected time period.

With it we can visualize the dominant emotions of the show and who portrayed them more, having the size of each word being proportional to the distribution of the emotion in the script and the color of the word set accordingly to the color associated with the character who most often portrays it during the selected period.

Figure 8. Hovering a word on the Wordcloud idiom, having one of the characters selected

As illustrated in Figure 8, hovering over a word will show the percentage distribution of the respective emotion among the six main characters. Clicking a word will select the character who portrayed it more (the character associated with the color of the word) within the selected time period.

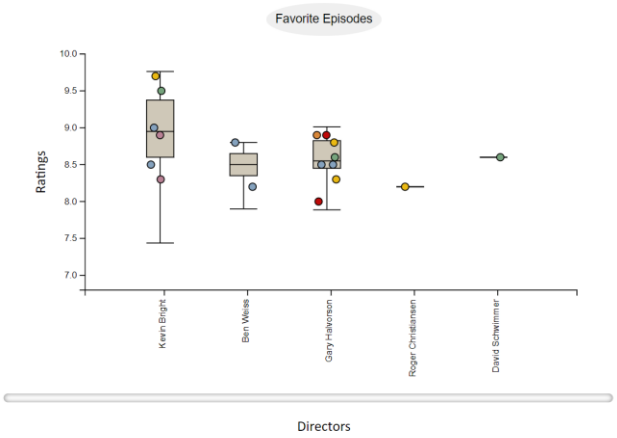

Figure 9. Overview of the Boxplot idiom

Below the Wordcloud, we can find a Boxplot idiom, illustrated in Figure 9, that opens the possibility of analyzing the variation of the episodes' ratings among directors, within a specific time period.

In this idiom each director corresponds to a different box, having each of their directed episodes encoded as a dot. The color of each dot is the color corresponding to the most active character in that specific episode (in essence the character with the largest quantity of lines in the episode in question). The position of the dot in the y-axis corresponds to the rating value of the episode.

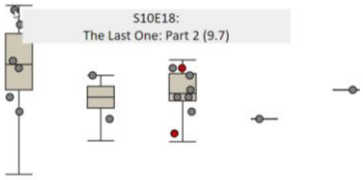

Figure 10. Hovering over a dot in the Boxplot idiom, having one of the episodes selected

Clicking a dot will select the data of that episode's most active character in the several idioms, greying out the remaining colors. As illustrated in Figure 10, hovering a dot will display some information regarding the episode in question, namely its season number, episode number, title and rating.

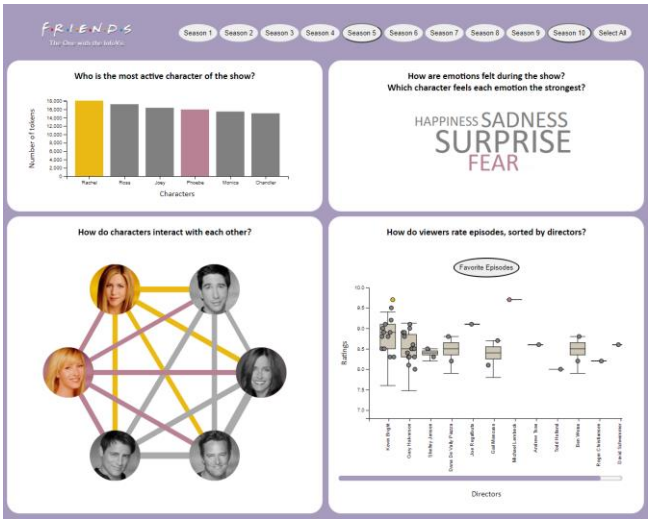

Figure 11. Result of clicking the "Favorite Episodes" button

Lastly, within the Boxplot idiom, we chose to implement a button denominated "Favorite Episodes". Clicking the "Favorite Episodes" button will select the highest rated episodes within the selected time period, as well as the characters who were most active in those episodes (in essence the characters with the largest quantity of lines in each of the episodes in question) in the several other idioms, illustrated in Figure 11.

Rationale

When choosing the above idioms, we made sure to take into account our specific dataset characteristics, our problem domain, the specific questions we wished to answer and our data abstraction. We made sure to justify every decision with the theoretical principles learned.

Starting with the barchart, we had gotten inspired by the idea of having a podium-like element to display the characters' presence, especially because humans are generally good at distinguishing the length of bars. When it comes to channels, we used different color hues to encode the categorical attributes (each of the different characters) and length to encode the presence attribute, consisting in the count of tokens, an attribute of type ratio. Regarding the presence attribute, we first decided to consider presence as the number of lines of each character in a given period. Yet, throughout the iterations of the project, we realized this approach could hinder the oral contributions of characters who tend to make lengthy speeches, as we found to be the case in a few situations, modifying it to account for the amount of words spoken in the character's lines in a period. As we made the decision to focus on the six main characters, we only made use of six different color hues throughout all of the idioms, making it a fitting channel in this case.

For the Network graph, in order to visually display the relations between the characters, we decided a node–link diagram would be fitting. We used color hue as a channel to encode the several categorical main characters and the lines

stroke to encode the ratio attribute (the count of interactions between the various character pairs).

When it comes to the Wordcloud, we used size to encode a ratio attribute (the distribution of the portrayed emotions) and color hue as a channel to encode the several categorical main characters. We are aware of the possible shortcomings of using a Wordcloud, however, we believe that in this particular case, having only a few emotion values and taking into account the theme of the idiom's data, it was our best option.

In the Boxplot we made use of different color hues as a channel to encode the several categorical main characters. The dot placement along the y-axis and essentially its position along the axis encodes the episode ratings (a ratio attribute).

Before we reached the final decisions for our idioms, as detailed above, we considered some other possibilities to answer our questions. When it came to display the interaction of pairs, we first thought of making use of a barchart encoding the number of interactions as the bar length for each character pair. Yet, due to already deciding on using a Barchart as an idiom and wanting both to make our dashboard more diverse and to try something new, as well as given the fact that the data regarding the interaction between the pairs could be considered a node-link structure, we decided that a Network Graph would display the connection between characters much better and could allow for better comparisons in a few cases.

Each of these idioms' channels, such as size, length or width, was always carefully coded in order to prevent any scalability issues, being essentially relative to the largest and smallest values of the data it encodes.

When it came to displaying each episode in our dashboard, we first thought of using a scatter plot, yet we soon realized that a boxplot would essentially also display each episode as a dot while providing us additional insights that would allow us to answer another question, regarding the distribution and variance of the episode ratings per director, something that a few fans showed interest in. In addition, a few fans go as far as showing some care or aversion towards specific directors, so this option seemed to be perfect.

At first we also set out to have an extra dashboard element to provide the answer to "What is the general audience's preferred episode in a given period?" but, as the prototype evolved, we found that this was better implemented within the Boxplot, using the "Favorite Episodes" button, becoming a more interactive option for the user (as our previous option lacked any interaction, essentially only displaying the best rated episodes within the selected time), as well as standing as a more resourceful option, as in the last iteration we now have the dots referring to the best rated episodes salient in the Boxplot idiom, additionally showcasing who directed them, adding ability for the user to visually compare these episodes' ratings to the others and having the several other idioms visually selecting the characters who are most active in the best rated episodes.

Additionally, the dashboard evolved iteratively when it comes to overall layout and design, especially when it comes to element design consistency, compartmentalization of the several idioms and the use of labels for each component.

As for the overall layout and design we tried to replicate something that gave us that 90s feel while being careful for it not to feel too old-fashioned. For the background color, we went for lilac, as it is a color that is very much associated with Friends, being the color of the apartment where the six friends spend the most time together, holding a huge significance.

Demonstrate the Potential

We were able to create a visualization that answers our proposed questions and some more. Let's explore two cases that corroborate it:

For the first scenario with the question "How do the episode ratings vary among the show's directors?", we start by filtering the seasons by clicking on the button "Select All" (in case all the seasons aren't already selected). This way we can see the dashboard supplied with data regarding all seasons.

Now, to answer our first question, we will be focusing on the boxplot area, on the bottom right of the dashboard.

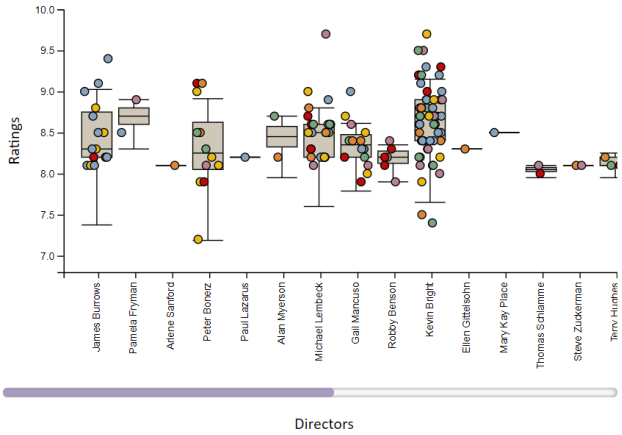

Figure 12. Boxplot, after all seasons are selected

Here we can check each episode's rating, grouped by each director, and analyze the variance for each of them. By hovering each dot, we will also be able to visualize the episode's corresponding number, season, title and respective rating. Each dot has a specific color assigned to it, representing the character with the most presence in that same episode, if you click it, additionally, you will be able to light up all the information of that character while greying out the rest.

We had anticipated we would also be able to tell which directors were the most requested (the director with most dots per box) but we did not initially anticipate we would

also be able to visualize correlations between the most present characters and the directors. In Figure 12 we can tell that James Burrows has directed episodes in which the character Ross has had the most presence, which is an interesting insight we didn't account for.

For the second scenario, answering the question "What are the most common emotions throughout the show?", we will be focusing on the Wordcloud area on the top right of the dashboard. By selecting a few seasons at the top of the dashboard, in this case 7, 8 and 9, we can choose to analyze a specific period of the show.

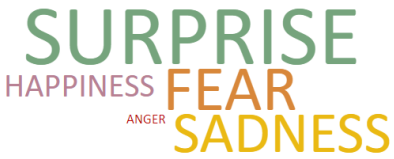

Figure 13. Wordcloud, after seasons 7, 8 and 9 are selected

By looking at the idiom, the user can visualize which emotions were felt throughout this chosen period of the show. Through the size of each word, the user can see which emotion has been felt the most. At first it wasn't in our plans to also use color to encode, for each emotion, the character who portrayed it most, yet as the opportunity arose, we figured it would help answer additional questions without compromising the ones we set out to answer in the first place. In general, actually, in order to make the most of the techniques available and link all of the idioms, we ended up answering a few more questions than we anticipated.

The most unexpected insight was to learn that independently of our filtering choices, the most common emotions were Sadness and Fear, which shocked us as Friends is not only a sitcom but generally considered a feel-good show. Another interesting insight was that there were not many differences on the distribution of each emotion amongst the six main characters. Being that the six friends seem to be very distinct it is curious that they experience relatively the same amount of each emotion throughout the various periods of the show.

IMPLEMENTATION DETAILS

When it comes to the idiom implementation, we had to overcome a few challenges. The Network Graph was a great and interesting idea but none of the ready-made options available online gave us what we were idealizing, so we had to plan and implement this idiom from scratch to fit our specific needs, making use of the d3 "forcesimulation" function. For the implementation of the Boxplot, we found we had to group each director's information ourselves, using the "groups" function, as the examples from lab classes didn't provide that functionality and the examples we found available online made use of a deprecated function. In this situation the available documentation regarding "groups" was very scarce, setting us back a good amount of time. When it comes to the Wordcloud, we made use of a d3.js package to facilitate the idiom's implementation. This choice was made due to the fact that all implementation examples we found made use of it. The package is locally stored in our project folder for offline viewing. Lastly, it is worth mentioning that none of us had any prior experience with d3.js and we ended up experiencing many setbacks due to the documentation available being rather small, which constituted an overall considerable challenge as well. For idioms such as the Barchart and the Wordcloud we could overall work on top of ready-made solutions even though we had to make them fit for our use case by modifying or adding new functionalities ourselves.

In order to filter the data throughout all idioms we created a Datachange function that created a subset of data according to the filter buttons that would be pressed, later fed into each of our idioms.

For the mechanisms used to support interconnection and linking between idioms we settled for an interesting mechanism of having attributes purposefully named differently across different datasets in ways that allowed us to, when an item is clicked, be able to know exactly what to update and where.

It is worth mentioning that our choice to develop idioms in a sequential order at first was well thought out as we were able to finish the implementation of the first idiom and then implement the remaining ones according to the mechanisms we previously defined. In addition, we started by implementing a Barchart, an idiom we had a bit of practise with, which made the process of integration with the remaining idioms and replication the process a bit easier in our opinion. Finally, it is also worth noting that we felt the idioms we chose were a bit complex for our skill level, as they were not covered in lab classes (the only exception would be the Barchart and the simple version of a Boxplot which didn't really fit our needs), which set us back for some extra work that we didn't fully anticipate.

Overall, in the end, we were able to make the best of what we had in hand, overcoming every single one of the setbacks we experienced and feel we were able to deliver the finished product we promised.

CONCLUSION & FUTURE WORK

With this project we learned how we could visualize information making use of data to answer our curiosities as Friend's super fans. We wouldn't have done it in any other way, but if we had more time and resources, the main things we would like to focus on would be conducting further User Research, as well performing Usability Tests of our prototype with our target audience, in order to evaluate what could be improved to further appeal the target users and better suit their needs. Additionally, we have only briefly tapped into the realm of Sentimental Analysis, which we ended up finding extremely fascinating, so in this sense we would love to conduct further learning and exploration in this area in order to further enrich our dashboard with additional idioms that could allow further exploration and analysis of

the characters feelings. In relation to this, the entire world of Natural Language Processing is also new to us but given our data a lot could be explored, such as conversation themes or catchphrase detection, for example.

At last, being able to answer the questions we set out to, in general it would be interesting to grow the dashboard a bit further, better separating it into several views and investing in answering additional questions that we had to leave out of this scope due this specific project objectives and duration.

REFERENCES

1. Friends (TV Series 1994–2004). (1995). IMDb. https://www.imdb.com/title/tt0108778/
2. Originals, C. (2021, May 27). Every Friends Script, Quote, and Scene Analyzed - Ceros Inspire. Ceros Inspire: Create, Share, Inspire. https://www.ceros.com/inspire/originals/friends-scripts-25th-anniversary-catchphrase-scenes-quotes/
3. Analyzing information from the popular TV show Friends. (2019). https://www.crystalywang.com/the-one-with-the-data-visualizations
4. VI2122: Hall of Fame. (2021). Information Visualization 2021/2022. https://pcm.rnl.tecnico.ulisboa.pt/moodle/login/index.php?id=1332
5. Emorynlp. (2020). character-mining/json at master · emorynlp/character-mining. GitHub. https://github.com/emorynlp/character-mining/tree/master/json
6. Friends Series Dataset. (2021, March 28). Kaggle. https://www.kaggle.com/rezaghari/friends-series-dataset?select=friends_episodes_v3.csv
7. Band, A. (2021, April 11). Text2emotion: Python package to detect emotions from textual data. Medium. https://towardsdatascience.com/text2emotion-python-package-to-detect-emotions-from-textual-data-b2e7b7ce1153