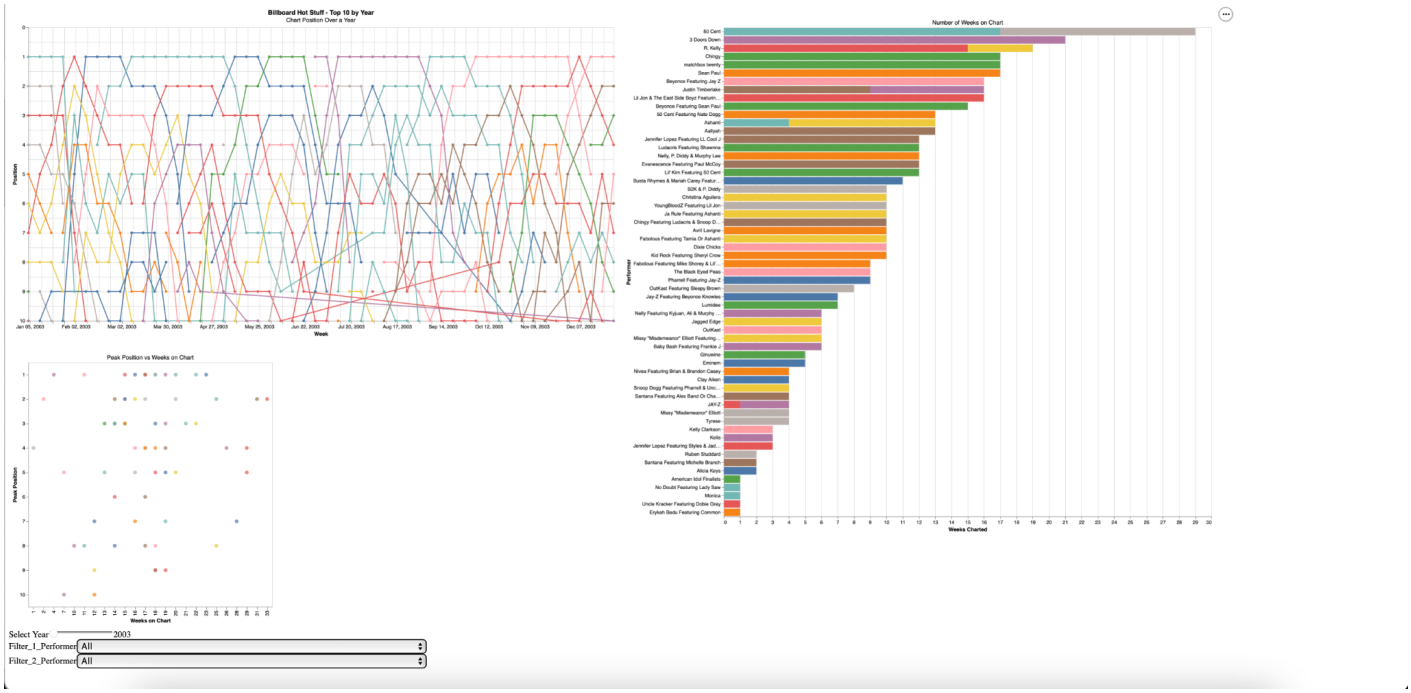
The Triad: Group 4

Jenny Fotso, Rayann Jamoussi, Felicia Li

Overview

In order to visualize the performance of different artists over the years, the top 10 billboard songs since the year 2003 were analyzed. Our visualizations include three linked visualizations: a line chart, bar graph, and scatter plot, all with the ability to sort on artists and year. The visualizations respectively show the total weeks the artists charted in the top 10 that year, the position by week of each song over the year, and the peak position vs weeks charted of each song entry. The colors encoded allow the user to better compare the two performers that they choose to filter on. Additionally, a tooltip on all of the charts allow the user to find the specific song and performer that corresponds to that data point. All together, our user should easily be able to compare the performer's rankings over time, peak positions, overall time spent on the chart, and number of songs that were charted in the top 10 for a designated year.



Data Description

Data: https://query.data.world/s/ngekjwbb3ap4kj3tqls2fmg535heai

The data we used was found online at data.world, and is a table of the billboard top 100 songs from the year 1958 until the year 2021. The table's fields include:

- the billboard chart url refers back to the song's billboard page containing its position during the specific week and the other 99 entries, and serves as a unique identifier for every entry in the table
- the week ID specifies the week during which each entry was recorded
- the week's position is the position each song held during a specific week
- the song name
- the song's performers
- the song ID is a concatenation of the song title and performer that also serves as a unique identifier for each song
- the instance is used to separate continuous breaks on the chart for a given song or, more simply put, each time a song has (re)entered the chart
- the previous week position
- the peak week position, not overall, but as of the corresponding week
- the amount of weeks a song has spent on the chart as of the corresponding week and for the specific instance

The original dataset had almost 350,000 entries. However, due to our environment's processing power, we scaled down to about 10,000 records.

The data set had no duplicates, so removing any non-unique values was not necessary. Our first data engineering step was making sure our attributes were the correct object types- Week ID was converted to the proper date-time type and week position to a numeric object. Once the data was converted, we used the weekID to create a new 'Year' attribute to make filter implementation more straightforward. To create the scatter plot, we extracted the minimum value from the 'Peak Position' attribute and the maximum value from the 'Weeks on Chart' attribute. Finally, due to its scale, we decided to filter the data to include only entries from 2003 and onwards. Additionally, only songs that were ranked in the top 10 were included in the final data set.

The main challenge was working with the ‘Performer’ column because entries were not normalized. We could not run code using the ‘split()’ method that would add featuring artists to another column based on a separator because some used the word ‘feat’ or ‘featuring,’ punctuation like an ampersand or a comma and sometimes just listed out the artists with specific demarcation.

Goals and Tasks Description

The project was based on how our specific user would get utility out of our visualization. Our main user is a teenage super fan of music, often called a stan, determined to prove that their favorite artist is the best, or better than another, based on the information and metrics provided in our visualization.

It is the year 2019 and Ariana Grande is at the top of every chart; the music video for her hit single “thank u, next” has attracted more than 55 million views in its first 24 hours and you cannot escape her voice on the speaker at your favorite grocery store and on radio stations either. Ariana Grande receives so much acclaim that she is now referred to as the queen of pop by many, but especially her fanbase, a title previously associated with pop artist Taylor Swift. Our user is Sarah- a fourteen year old die-hard Taylor Swift stan who runs twitter fan page ‘@taylorsforehead.’ She has found herself in an intense twitter feud with Ariana Grande fan account ‘@ari_s_meowmeow’ after stating the title of ‘queen of pop’ only belongs to one artist and that is still Taylor Swift. Sarah is now passionately googling music statistics to prove her favorite artist is the best.

Our main goal was to make information that is often cumbersome and strenuous to find and read about, easily accessible and understandable so the user can leave with more knowledge than they came with. Every user need not be a die-hard fan of music, but anyone with even the slightest interest in popular music can use our visualization to learn about their artist of interest. The visualization allows our user to browse through information about billboard hot 100 entries and, most importantly, filter the data to extract information on a particular artist or two.

Sarah would use the drop down menus on our visualization to be shown data points relating to Ariana Grande and Taylor Swift exclusively. She would find that Taylor Swift has been in the industry longer than her competitor Ariana Grande, but both artists have had comparable commercial success that would in fact leave the general public leaning towards Ariana Grande

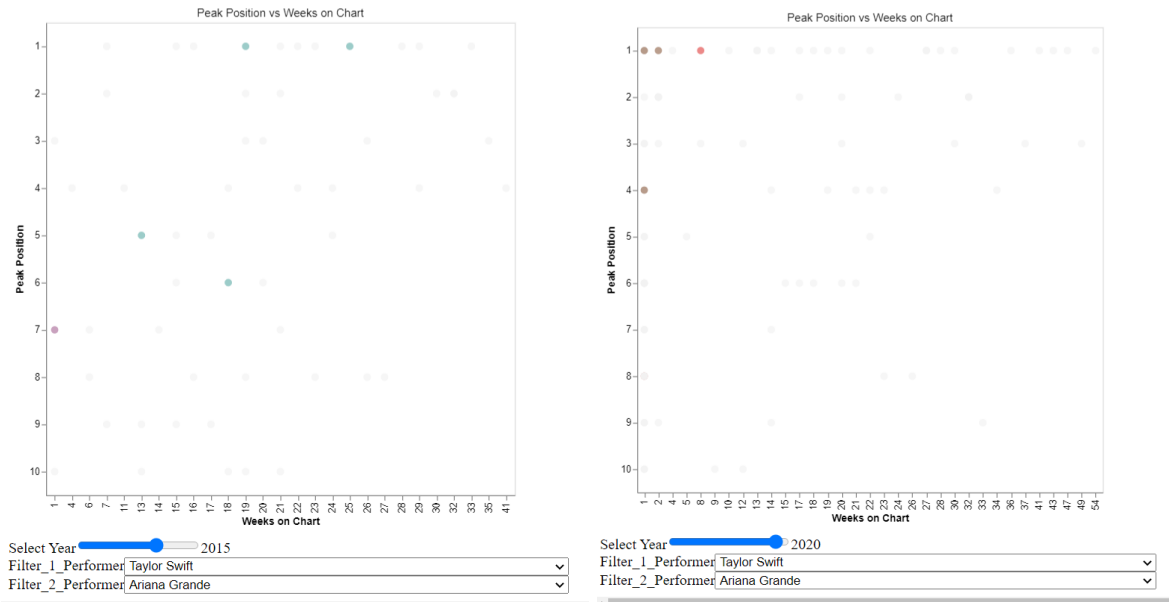
as the ‘queen of pop’ due to the recency effect. To confirm her bias for her favorite pop singer, Sarah must find a metric that would clearly leave Taylor Swift on top and in first place. While clicking through our visualization and using her analytical skills, Sarah found that Taylor Swift songs spend more time on the charts. With this information in hand, Sarah logs onto twitter to proclaim that the most important attribute the ‘queen of pop’ must possess is longevity. In this regard, Taylor Swift holds that title and Ariana Grande has yet to surpass her. Sarah receives the most likes she’s ever gotten on a single tweet and gains 84 new followers on her fan account.

In the same manner as Sarah, anyone interested in popular music could use our visualization to learn about the commercial success, via chart placement, of their song of interest.

Visualization Building

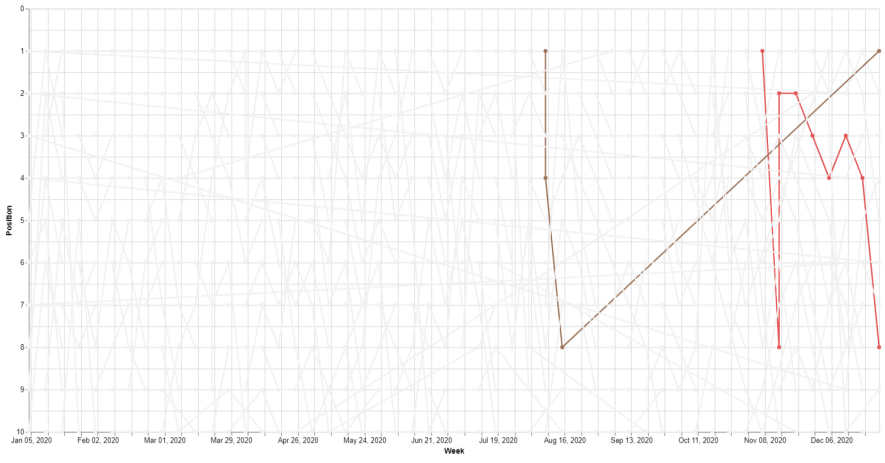
Visualization: https://www.cs.utexas.edu/~chaney/spring2022/group_4/all_together.html

Our visualization interface is set up to allow users to compare the relative success of two artists. The drop down filters located at the bottom left quadrant allow the user to filter out the other data points, leaving only the two selected artists colored on the three idioms while the unselected artists are grayed out. The third filter is a slider that allows users to toggle the data shown by year. With this slider, the user can decide if they want to see the performance of artists during a certain year or the user can slide between years to observe the growth or decline of the performer over time. We decided to include these filters since the data presented without filters was an incomprehensible mass of color for our line chart and scatter plot. Furthermore, we wanted more flexibility for the user to select what data they want to view for tasks they wish to accomplish.



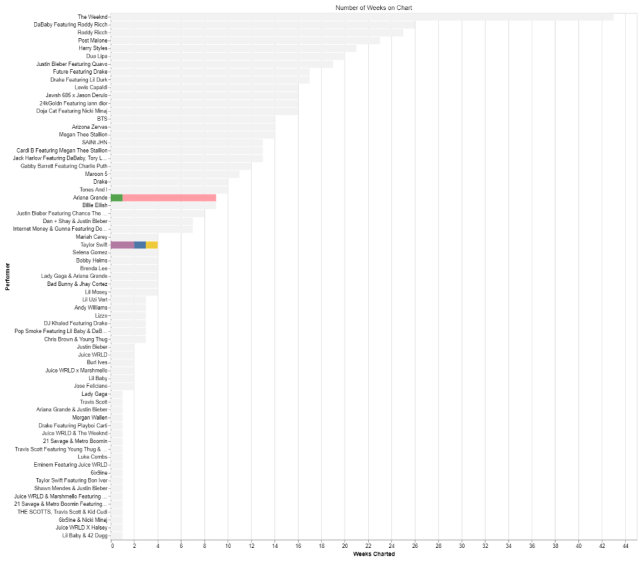
[*Filters at work* - Arianna Grande vs Taylor Swift scatter plot, 2015 vs 2020]

Our line chart shows the top 10 songs of the selected year. We chose this idiom as it allows our user to track the fluctuations in rank of charted songs of the selected artists. Something to note is that the chart shows the colors based on artists only, not songs. We tried coloring by songs at first, but since there were way more songs than colors, the colors began to repeat and became meaningless. Additionally, the color encoding the artists allows the user to easily compare the performance of the artists holistically over time rather than by song. If the user prefers to identify the specific songs, the tooltip can be used. The tooltip is active on all three idioms and allows the user to see the artist and song of the data point they are hovering over.



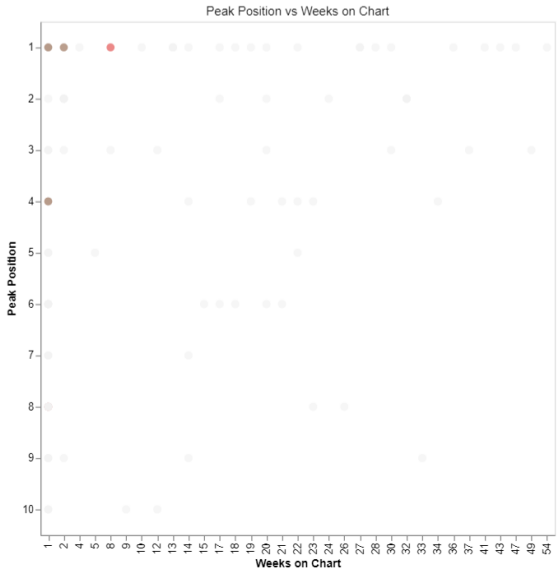
[*Line Graph at work* - Arianna Grande vs Taylor Swift, 2020]

Our bar graph shows each artist in the year (y axis) and how many weeks they charted total (x axis). The graph is also sorted in descending order for the user to easily identify which artist charted in the top 10 for longer. For this graph, the colors are separated by song. These design decisions were made to allow the user to easily identify how many total songs the artist had charting in the top 10 for that year. For example, one song can chart for 10 weeks while five songs can chart for a total of two weeks. It's up to the user to decide which metric is more valuable to them.



[*Bar Graph at work* - Arianna Grande vs Taylor Swift, 2020]

Finally, our scatter plot shows the top 10 songs of the selected year and their peak position on the chart (y axis) and the number of weeks the song was on the chart (x axis). Like the line chart the data points are colored by artist rather than song to allow the user to visually compare the chart performance of the two selected artists. We chose this idiom and comparison because a number one chart topper doesn't mean the song's guaranteed longevity. At the same time, longevity on the Billboard charts doesn't mean the artist reaches the top.



[*Scatter plot at work* - Arianna Grande vs Taylor Swift, 2020]

Reflection

Our base idea of visualizing Billboard top 100 data has been pretty consistent throughout the length of the project. The first thing to change was our dataset after our initial proposal, since we got feedback that the dataset we started out with, data of the Billboard 100 songs for 2000 only, wasn't robust enough to meet our user tasks. So we switched over to an extensive dataset that included data from 1958 until 2021.

Our user tasks and idioms naturally changed to be cohesive with our assignment parameters, our own technical skills, and of course our user. For our visualization, we stuck to idioms we covered in class that would be very straightforward and easy for our user to digest: line chart, bar chart, and k-means clustering. The bar chart evolved to be a stacked bar chart and the k-means clustering became a scatter plot in our final implementation to better suit the user tasks we had decided on.

We didn't really have any major changes to our technical goals either, so overall our project proposal was quite feasible. A problem we had was the fact that we were encoding color to songs, and there weren't enough colors to go around which led to color losing its ability to be a visual identifier. But we decided to keep the colors so that it would at least be a contrast for a song from the song next to it on the visualization. Something we wanted to implement was a

gradient color scale for the stacked bar chart as a way for users to see how many total songs charted for an artist. But alas, there was no time for further exploration.

Another problem we ran into was a slight loss of accuracy since the code would count songs an artist featured on as a unique artist, so this definitely skewed the numbers for artists that collaborate a lot. If we had more time, we would've tried to create a list of unique artists then split the featured songs into multiple entries, one entry per artist. Google Colab wasn't able to handle the full range of dataset we wanted it to (2000 to 2021), so we reduced our range to 2003 to 2021. But if we were to redo the project again, we might try using a different software to run our code that can process a greater amount of data than Google Colab. This way we can run our visualization for the entirety of the 2000s.

Team Assessment

In the beginning stages of the project we completed the milestones together up until the design rough out, but as the work got more complex we found it more efficient to delegate and give every member an area of focus based on their skills and interests.

Jenny came up with the initial idea for our project. She cleaned our data and even created a mini table for us to play with and code more efficiently before replacing it with the real dataset once our visualizations were complete. She coded the line chart, engineered the data to create a new table for the scatter plot, and coded the scatter plot. She also wrote the "Data Description," "Goals and Tasks" sections and reviewed the report before submission.

Rayann did the coding for the stacked bar chart on our visualization. She also wrote the "Overview," "Data Description," and "Visualization Building" sections of the report.

Felicia created the theme for our presentation slides and wrote the report outline milestone. She did the coding for the filters, interactions and finishing properties for the visualization. She also wrote the "Reflection," "Team Assessment," and "Visualization Building" sections of the report.