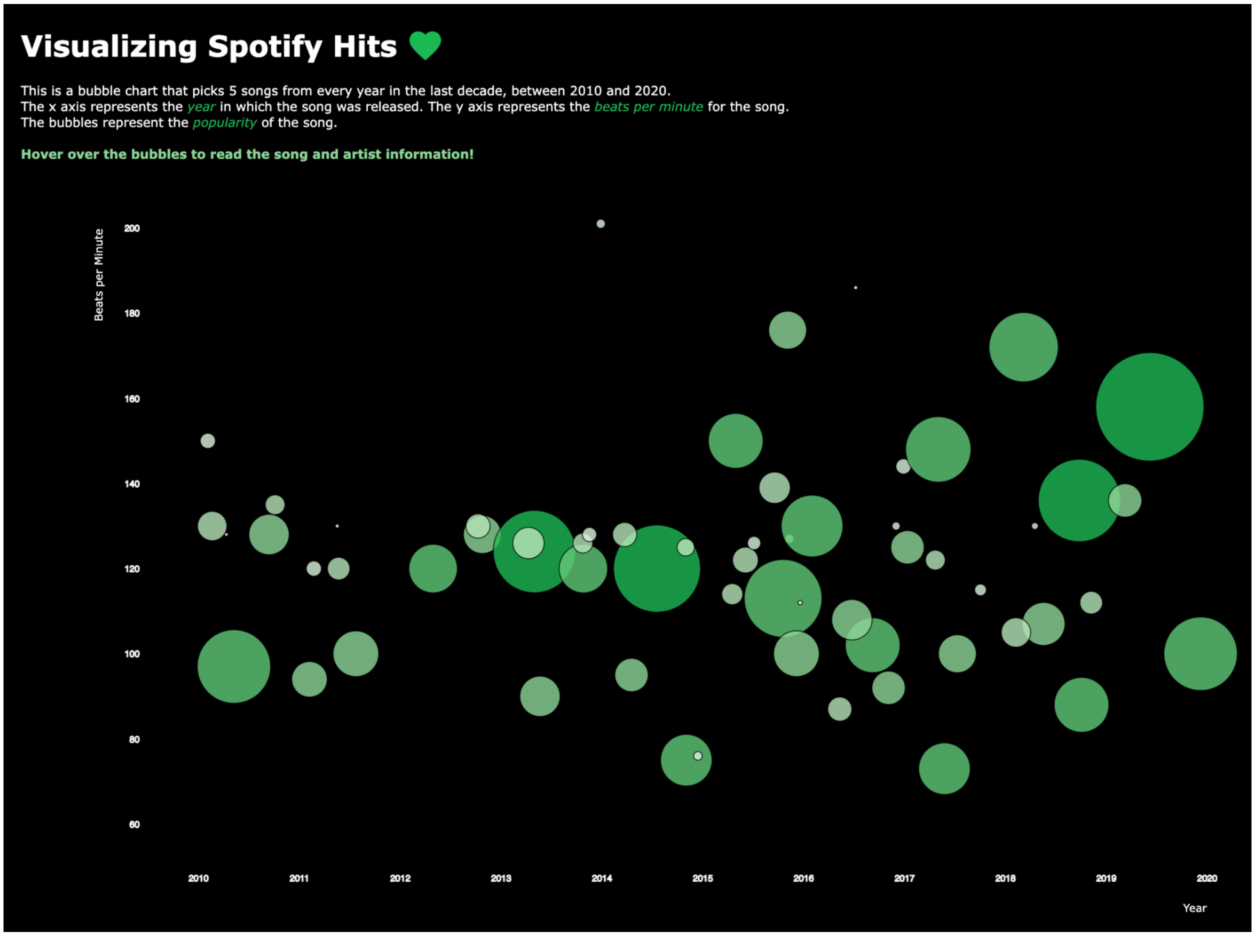


Visualizing Spotify Hits 💚

This is a bubble chart that picks 5 songs from every year in the last decade, between 2010 and 2020.
The x axis represents the *year* in which the song was released. The y axis represents the *beats per minute* for the song.
The bubbles represent the *popularity* of the song.

Hover over the bubbles to read the song and artist information!

Visualizing Music Using Spotify Data & D3.js

09.12.2022

Aparna Maya Warrier

PUI Final Project
Lab D (Venkat)

Website Info + Github Links

Website is live at
https://a-warrier.github.io/teach-pui-example/SpotifyD3js/src/spotifychart.html

Github repository link for main code
https://github.com/a-warrier/teach-pui-example/blob/main/SpotifyD3js/src/spotifychart.html

PART 1: Project Overview

The main purpose of the website is to provide a novel way to experience music that is not limited to an auditory experience. Data visualization using D3.js allows us to visualize data and gather insights using currently available Spotify data. The main information that is conveyed in this website is through the interactive bubble chart.

The chart uses information that is presented in a dataset of the top songs of the past decade - this dataset is a CSV file that is available on Kaggle (a data science community that has a provision to share datasets with the public). The dataset was cleaned and filtered using Python to create a subset of data that picks 5 songs from each year to plot on the bubble chart.

The x-axis plots the different songs over the last decade (running from 2010 to 2020). The y-axis refers to the beats per minute for each individual song. The bubble radius themselves represent the popularity of each song - bigger bubbles represent more popular songs.

The target audience for this website is anyone that has a deep interest in music through the years; the chart also helps users learn more about the elements that make a particular song more successful. It is interesting because music is usually an auditory experience - “seeing music” is a novel experience!

PART 2: Interactivity

Users can interact through the website using the interactive bubble chart to understand which songs are popular over the years, and using information about their BPM (beats per minute). Here are the steps to access this interaction:

- Hover on any of the bubbles that are plotted on the chart
- A tooltip element/box will pop up for the selected bubble that provides users information about the song and artist that is represented by that bubble

Users can interpret that the average beats per minute seems to produce the most popular music. The songs at either extreme are not as popular as the ones in the middle of the chart. As the years progress, songs have an increasing trend in popularity through the growing demand and interest in the music industry over the past decade.

PART 3: External/New Tools Used

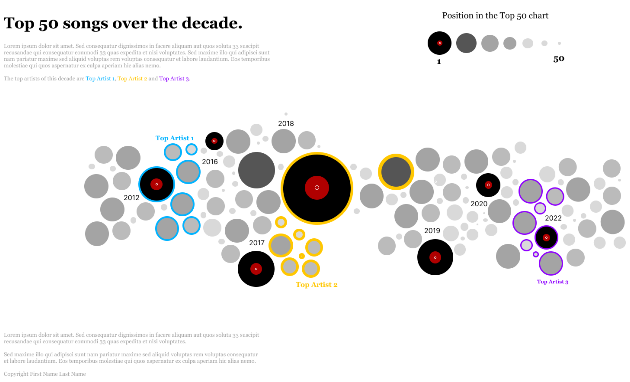
The external tool used in this project is D3.js, which is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. This was the external tool that I chose because my objective was to create a data visualization that was insightful - but also highly customizable and had a powerful visual appeal. D3.js is the right choice for this objective because it allows programmers to customize every attribute of the graph. D3.js governs the majority of the project that I created as the interactive bubble chart is the primary element of my website. It creates the interactivity and novel experience of “visualizing music” that I aimed to create using the Spotify data that was available.

PART 4: Prototype Iterations & Feedback

There are a lot of changes, improvements and iterations that I made along the way, as I built this project from the ground up. Initially, I had created a hi-fidelity prototype that had no axes - I changed this to add the x and y axes because I believed that adding this made the charts more meaningful and comprehensible for a user.

I also reduced the number of different colors being used so that it does not create unnecessary load on the user while they try to process the information on the graph. I also wanted it to be linked to the data that is being used - hence, I ended up finalizing the colors to go with the Spotify brand colors. I also switched it to a dark theme as it felt less strenuous to view the dark themed website, while also offering sufficient contrast to address any accessibility issues that may crop up.

The initial hi-fi prototype and the final screenshots are attached below.



Top 50 songs over the decade.

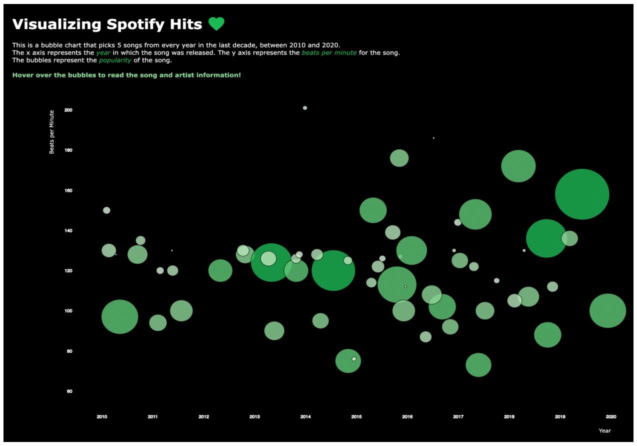
Lorem ipsum dolor sit amet. Sed consequatur dignissimos in facere aliquam aut quos soluta 33 suscipit recusandae qui consequatur commodi 33 quas expedita et nisi voluptates. Sed maxime illo qui adipisci sunt nam pariatur maxime sed aliquid voluptas rem voluptas consequatur et labore laudantium. Eos temporibus molestiae qui quos aspernatur ex culpa aperiam hic alias nemo.

The top artists of this decade are Top Artist 1, Top Artist 2 and Top Artist 3.

Lorem ipsum dolor sit amet. Sed consequatur dignissimos in facere aliquam aut quos soluta 33 suscipit recusandae qui consequatur commodi 33 quas expedita et nisi voluptates.

Sed maxime illo qui adipisci sunt nam pariatur maxime sed aliquid voluptas rem voluptas consequatur et labore laudantium. Eos temporibus molestiae qui quos aspernatur ex culpa aperiam hic alias nemo.

Copyright First Name Last Name



Visualizing Spotify Hits 💚

This is a bubble chart that picks 5 songs from every year in the last decade, between 2010 and 2020.
The x axis represents the *year* in which the song was released. The y axis represents the *beats per minute* for the song.
The bubbles represent the *popularity* of the song.

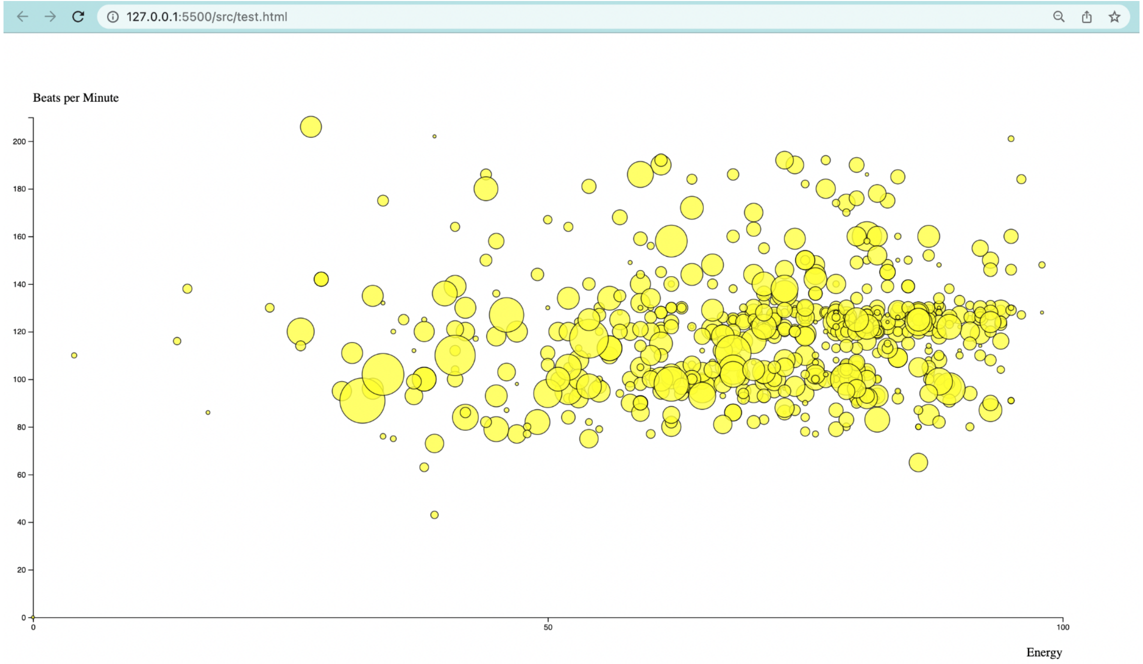
Hover over the bubbles to read the song and artist information!

PART 5: Challenges

I experienced many challenges that allowed me to learn through the process of building this interactive chart. One of the challenges was learning to create advanced customizations and visual elements in D3.js. It is a library that we had not come across before - the code and syntax of this library is quite different from simple JS. Hence, I spent a lot of time browsing and learning through online documentation and video tutorials online.

Another challenge that I faced was filtering out data and learning more about the variations within that data to make the chart more comprehensive. Initially, the Kaggle CSV file consisted of a large collection of songs (around 600 songs in total). This made the initial bubble chart look very cramped and complex for users to understand.

Here is a screenshot of the initial chart that I had tried to plot with the unfiltered data:



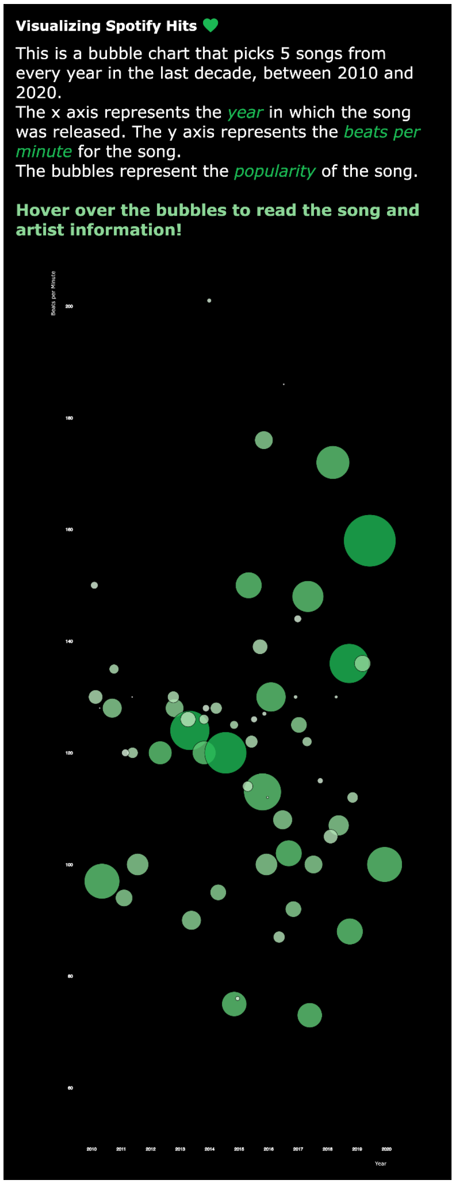
This was then improved by cleaning and filtering out data using Python. The python code can be accessed here. Essentially, the code creates a dispersed column (using random fractions) for the year column, so that the songs are more spaced out in the graph. It also filters the CSV data to pick just 5 songs per year for the last ten years, thereby reducing the number of songs that are plotted on the bubble chart. This results in a much cleaner and simpler graph that is represented by the final result on the live website.

(Responsiveness and accessibility related information in the following pages...)

PART 6: Responsive Design

The responsiveness of the website is handled by using the window: resize event that calculates the device dimensions that the website is currently running on. This was tested using the dimensions of a mobile, tablet and desktop. One point to note is while visualizations are best viewed on the desktop, it can be analyzed on other devices as well.

Testing responsiveness toggle device toolbar - with iPhone 14 (390 x 844) dimensions:



Visualizing Spotify Hits 💚

This is a bubble chart that picks 5 songs from every year in the last decade, between 2010 and 2020.
The x axis represents the *year* in which the song was released. The y axis represents the *beats per minute* for the song.
The bubbles represent the *popularity* of the song.

Hover over the bubbles to read the song and artist information!

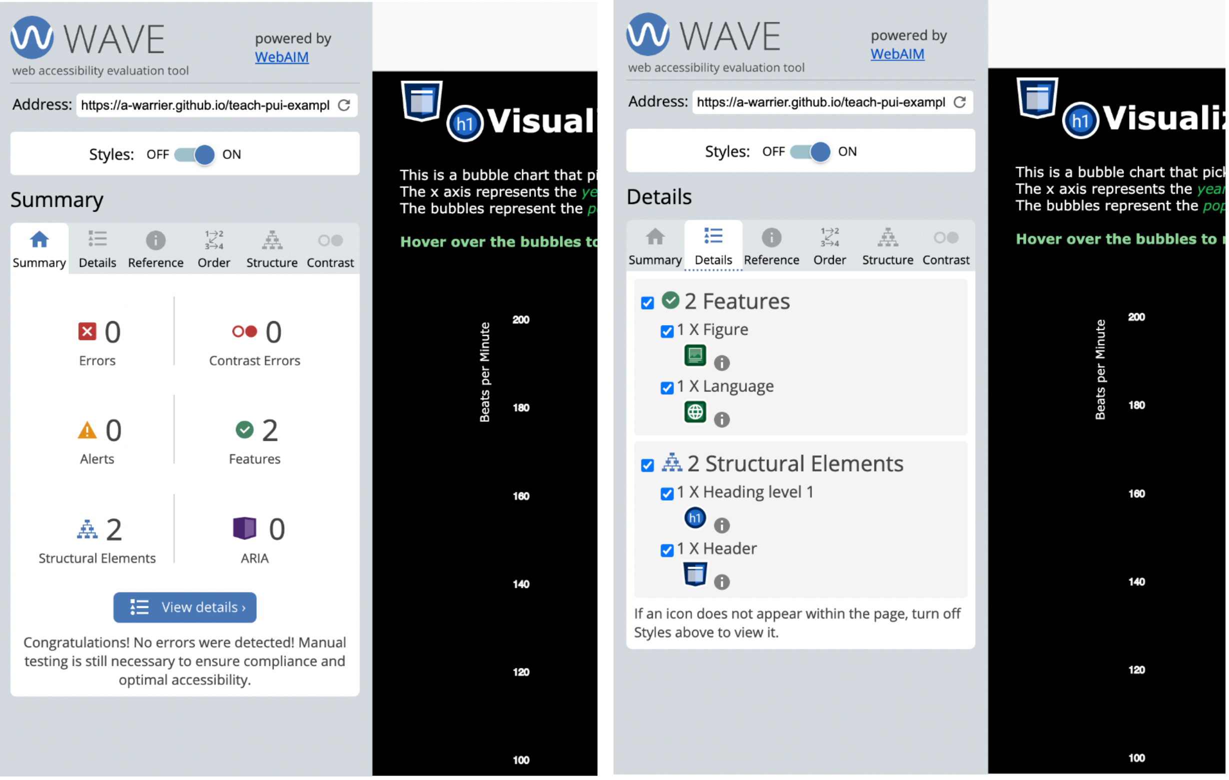
PART 7: Accessibility

There are definitely some challenges regarding accessibility of visual content, especially in the case of complex data visualizations and graphs. This is an ongoing study in the field of HCI and has taken on many routes. One way that accessibility is addressed in this project is to give comprehensive tags to the HTML elements - header, figure, figcaption etc are used, instead of normal div and class names.

The figure section refers to the bubble chart, whereas the figure caption refers to the write-up on top of the graph that summarizes the chart and its insights. I also learnt that incorporating the right headers (such as H1, H2 etc) in HTML can have additional meaning in terms of how screen readers read the website.

The WAVE tool was used to refine and test the accessibility of this website. After the changes were addressed, it reported 0 errors and 0 alerts, as shown below.

The screenshot of the WAVE tool analysis (summary and details) is attached here:



Address: https://a-warrier.github.io/teach-pui-exampl

Styles: OFF ON

Summary

0 Errors

0 Contrast Errors

0 Alerts

2 Features

2 Structural Elements

0 ARIA

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

This is a bubble chart that p
The x axis represents the ye
The bubbles represent the p

Hover over the bubbles to

Address: https://a-warrier.github.io/teach-pui-exampl

Styles: OFF ON

Details

☑ 2 Features
- ☑ 1 X Figure
- ☑ 1 X Language

☑ 2 Structural Elements
- ☑ 1 X Heading level 1
- ☑ 1 X Header

If an icon does not appear within the page, turn off Styles above to view it.

This is a bubble chart that pic
The x axis represents the year
The bubbles represent the pop

Hover over the bubbles to

Thank you for reading!