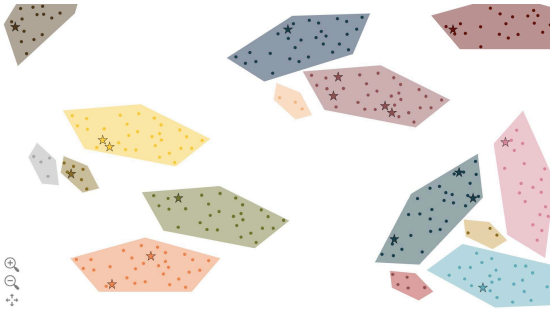
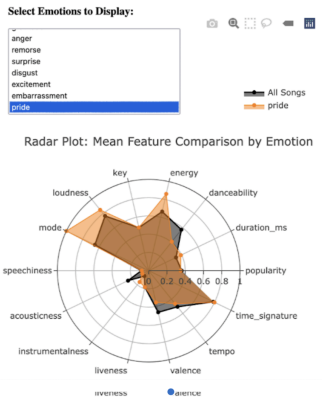
Lyricalytics - Milestone 2

We have designed 5 different visualizations to answer our project's question: **what is the impact of song lyrics' sentiments – do song emotions impact a song's popularity and characteristics?**

Our five visualizations are as follows:



1. **Overview of the songs in the emotion space through dimensionality reduction.** This is our most ambitious visualization, serving as a general presentation of our project, and capturing all dimensions of the data we worked with.
   - **MVP:** We use the various song features to form a "song embedding," and then apply Principal Component Analysis to reduce it to either 2 or 3 dimensions. We also plot the emotion clusters by encircling in a colored the songs belonging to an emotion. If the result is too overloaded in a visual sense, we will explore other ways of either plotting or reducing our data's dimensionality; we will also consider there being no drawn emotion cluster shape, and representing each cluster simply by the songs' (dots) colors. Finally, top hits are distinguished from other songs by being represented by a star instead of a dot.
   - **Extra ideas:** We want to add hover functionality to each dot (which represents a song), whereby information about that song is displayed. We would like to also add some interactivity, allowing the user to explore the plot, able to zoom in and out as well as move around.
   - **Tools:** If we decide to do a 3D visualization, we will likely use Three.js. If we deem this too complex for the time we have for this project and do a 2D visualization instead, we will either use Plotly.js or D3.js for our implementation. We will iterate quickly and test a small prototype on each library to decide which will be used for our final approach.
   - **Lectures:** Interaction, Perception-Color, Mark and Channels (especially Shape), Tabular Data, (and D3 lectures, including interactive D3, if we end up implementing with D3); Designing lecture concepts will be used to write a complementing introduction on the website.
2. **Radar chart capturing how emotions in lyrics correlate with song features (e.g., tempo).**
   - **MVP:** We use normalized numerical features from both songs and associated images to represent each emotion category, scaling values between 0 and 1. Radar plots visualize these features, enabling a multi-dimensional comparison. We also display the normalized mean feature values of the entire English song dataset as a baseline. Users can select an emotion category to compare its feature profile against the overall average.It is interactive and the user can choose to hide or visualize the average features for the songs corresponding to each emotion.

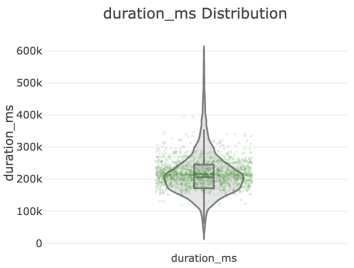


   - **Extra ideas:** Polish the current rough implementation to improve user friendliness.
   - **Tools:** Plotly.js
   - **Lectures:** Perception-Color, Mark and Channels, Tabular Data; Designing lecture concepts will be used to write a complementing introduction on the website.
3. **Word cloud visualization capturing how each emotion relates to lyrics.**
   - **MVP:** For each emotion category, we concatenate lyrics of corresponding songs and extract the 100 most frequent words (stop words are not considered). These are visualized in a word cloud, where font size reflects word frequency and each word is colored differently for better distinction. We avoid random rotation to enhance readability. Users can

interactively explore one emotion category at a time, highlighting the most common lexical elements associated with each emotion.

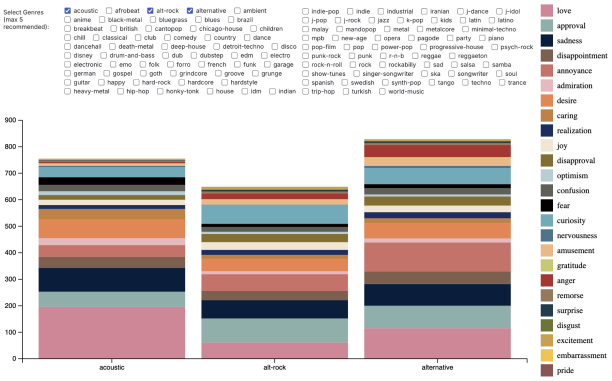
- **Extra ideas:** Make the visualization interactive, starting with the most common words across all songs, and allow users to include or exclude emotion categories dynamically. Enable comparison by displaying multiple word clouds side-by-side (e.g., joy vs. sadness). Create word clouds based on noun–adjective pairs to better capture emotional nuances (as seen in Text Viz lecture, slide 22).
- **Tools:** D3.js (https://observablehq.com/@d3/word-cloud)
- **Lectures:** D3 lectures, Interaction, Perception-Color, Text Visualization; Designing lecture concepts will be used to write a complementing introduction on the website.

4. **Scatter-enhanced violin plot capturing how song features relate to popularity.**



   - **MVP:** We construct a scatter-enhanced violin plot using 12 carefully selected numerical features. Binary features like "mode" were excluded due to their limited distributional information. Violin plots were chosen for their ability to show both summary statistics and the underlying density distribution. Top-hit songs are highlighted in green to visualize their position within the overall feature distribution. Initially, non–top-hit songs were also plotted in red, but their high quantity and variability overwhelmed the plots, so we focused solely on top-hits for clarity.
   - **Extra ideas:** Having a single unified view, whereby we explore all features on a single plot, rather than our current 8 separate ones (this might require rethinking the y-axis). Find a better way to represent top-hits and non–top-hits to improve readability, maybe by replacing the violin plot by a full scatter plot (with some x-axis variance so a similar shape will be retained). Explore whether there are outlying emotions, and if so, visualize their feature correlation using a third color.
   - **Tools:** Plotly.js
   - **Lectures:** Perception-Color, Mark and Channels, Tabular Data; Designing lecture concepts will be used to write a complementing introduction on the website.

5. **Stacked bar chart visualization for genre and lyrical sentiments.** It explores the distribution of emotions across genres, offering users an interactive and comparative view of sentiment trends.



   - **MVP:** An interactive stacked bar plot where users can select up to five genres using checkboxes. Each selected genre is shown as a stacked bar, with the y-axis initially representing raw counts and a legend indicating emotions.
   - **Extra ideas:** 1) Improve genre selection with dropdown menus or grouped interfaces. 2) Display the percentage of top hits below each bar, with a brief explanation. 3) Switch the y-axis to percentages to address large count differences. 4) Show emotion names and counts within each bar layer for clarity.
   - **Tools:** D3.js (https://observablehq.com/@d3/stacked-bar-chart/2)
   - **Lectures:** D3 lectures, Interaction, Perception-Color, Mark and Channel, Tabular Data; Designing lecture concepts will be used to write a complementing introduction on the website.

Note regarding lectures for all our visualizations: Lectures such as “Dos and Don’ts”, “Designing”, and “Storytelling” are kept in mind but not directly applied. Sound visualization concepts are unfortunately not applicable to our data, as we do not have access to music files, melodies, etc. but instead just have tabular data.