ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

PROCESS BOOK

DATA VISUALIZATION COM-480

Théo Lemonnier, Caroline Verchère, Théo Houle

30th May 2025

CHAPTER 1

INTRODUCTION

For this project, we quickly decided to work on a music-related topic. Music is universal, and everyone has listened to some. But there is a lot of music worldwide, so we thought it would be interesting to provide some way of visualising this massive amount of data in a simple and comprehensible way.

To do this we spent quite some time looking a what already exists and inspired ourselves from it. Notably, we saw a website that allows diving through all types of genres and subgenres, and for each, a short preview of a typical song of that subgenre could be played.

Initially, we wanted to be able to explicitly show links between genres and niche genres, showing how they influence each other. But we realised that this was not as easy as we thought, and had to simplify. Instead, we decided to let the user browse through lyrics of songs, what are the most common words based on the genre, etc, browse through the top 200 songs and artists each year according to Spotify and browse through the genres and sub-genres. This approach, although not as deep an analysis as we would have wanted, is still well-rounded and allows looking at multiple facets of the music industry.

In the following, we will discuss in detail the implementation of the different visualisations and, at the end, we will include a breakdown of the parts of the project completed by each team member.

CHAPTER 2

VISUALISATIONS

2.1 SPOTIFY CHARTS TOP 10

The goal of this visualisation is for the user to be able to see the most listened to songs in each country, each year. We originally planned to use this Spotify Charts kaggle dataset but quickly realized that there was quite a lot of data missing (for some countries, the top200 or top50 was not complete). So we had to resort to another method. We discovered the Spotify Charts website which holds for each week, the top 200 songs, albums and artists. Thus, in order to use this data, we created a scraping bot. We only scraped the songs and artists as it is quite and lengthy process and that artist and songs are already enough for this animation.

The scraping was in done in python using the `request` package. Check out the file `crawler.ipynb` for more information. It allowed us to have two new datasets: song and artist rankings. They contain a line for each entry in the top 200 for each week of each year in each country. There are 74 different countries and the data spans from 2017 to 2025 for the songs and from 2021 to 2025 for the artists.

This animation starts with a map of the world (see figure 2.1). The user can choose a year and a "type" (i.e. song or album), and then the countries are colored based on the top artist (or song). Then when hovering over the country, the name of the country and the artist (or song) is displayed. Upon clicking, the top 10 artists (or songs) are displayed for that country that year. Just below this is the same ranking but world-wide.

For this first animation, we needed to find a way to aggregate the top 200 rankings of each weak of that year to find the overall top 10. For this we decided to simply add up the ranking of all artists (or songs) and keep the top 10 with the smallest value. This acts as a weighted sum. This might not be the best idea but it works.

The next step of the animation was to have a pie chart of the genres, but since we decided to extend this visualisation to song and artist, it did not really make sense anymore. So we dropped this idea.

Finally, we wanted to have a sort of bump chart showing the evolution of the top 10 in that country over the year. This turned out to be a lot more complicated than we anticipated because we need to manage artists (or songs) going in and out of the top 10. Instead, we decided to simplify by plotting for the overall top 10, their ranks throughout the year. In the end, it looks a lot like what we wanted initially because it turns out that the top 10 is pretty stable through the year. Unfortunately, though, this animation does not work as well with the songs because a lot of the time, they will appear in the top 200 in the middle of the year and the animation looks a bit ugly. But it still works as intended.

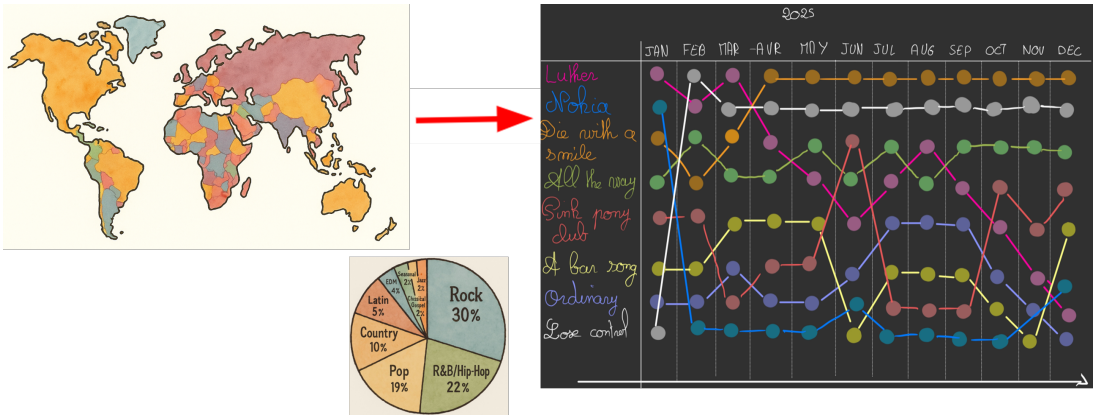

FIGURE 2.1
Initial visualisation sketch submitted in Milestone 2

2.2 GENRE BUBBLES

This visualization invites users to explore the universe of music—literally and figuratively. Just like stargazing through a telescope, users can zoom into different musical constellations, dive into clusters of genres, and discover the unique sound of each. The interface is designed as a cosmic journey through three immersive levels: main genres, subgenres, and finally, individual tracks, each offering both visual and auditory discovery.

We started by cleaning and aggregating a large dataset of Spotify tracks, including metadata like popularity, energy, and genre tags. From this, we calculated average values per genre and subgenre (e.g., number of tracks, mean popularity), enabling us to build layered, bubble-based visualizations using D3.js.

2.2.1 LEVEL 1: MAIN GENRES

The first screen displays a floating map of main genres, visualized as colorful, animated bubbles drifting through a galaxy-themed background. This visual metaphor sets the tone: music is a universe waiting to be explored.

- **Bubble size** = number of tracks in the genre
- **Color** = average popularity

Bubbles bounce gently, collide, and float with smooth, continuous motion—an effect customized from D3's force simulation to create an endless, organic drift.

In early prototypes, we included X and Y axes to display musical attributes such as danceability and energy. While informative, this approach introduced too much visual complexity and constrained the playful nature of the experience. Ultimately, we decided to remove the axes entirely. This allowed the bubbles to move more freely, creating a more dynamic and joyful interface. The visual became lighter, more engaging, and better aligned with the goal of exploration rather than analysis.

2.2.2 LEVEL 2: SUBGENRES

Clicking on a genre bubble zooms into a second map: a set of subgenres for the selected genre. These follow the same visual logic but are colored differently to signal the deeper level of detail.

From here, users can click on any subgenre to access a hand-picked selection of up to 10 representative tracks—opening the door to real listening exploration.

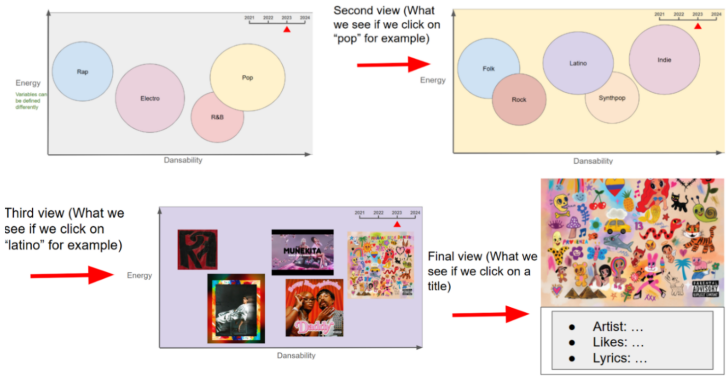

FIGURE 2.2
Initial visualisation sketch submitted in Milestone 2

2.2.3 LEVEL 3: TRACKS

The final view reveals a floating set of album covers—tracks within the selected subgenre. For each track, we retrieved via the Spotify API:

- The album art (displayed as the bubble)
- The track and artist name
- An audio clip

The magic happens here: each bubble represents a sound you can hear. Users can click to preview, listen, then return to the map, click another subgenre, compare, and discover something completely different. The freedom to navigate between styles and instantly hear what they sound like makes this more than just a data visualization—it's a musical exploration tool.

The interface lets the bubbles float freely, and some overlap naturally occurs as part of the dynamic, organic motion. The galactic background, clean typography, and smooth transitions support the visual metaphor of music as a universe to explore.

2.2.4 TECHNICAL AND DESIGN HIGHLIGHTS

One of the biggest challenges was handling Spotify's API rate limits while fetching metadata. We implemented error handling and token-based retry logic. All collected data was preprocessed and saved to static JSON files to ensure fluid interaction and fast load times.

On the design side, several challenges emerged as we refined the interface:

- **Label visibility**: Bubble labels were difficult to read at first due to motion and low contrast. We improved them using clearer fonts, outlines, and better positioning to ensure readability.
- **Information overload**: Displaying too many tracks per subgenre created visual clutter. We chose to limit each subgenre to 10 curated tracks to make the experience more focused and digestible.
- **Bubble motion**: Fast or erratic movements made the view hard to follow. We adjusted the force simulation parameters to slow down the drift and maintain a calm, readable dynamic.
- **Color saturation**: An overly colorful palette resulted in cognitive overload. We opted for a more restrained and coherent color scheme, allowing album art to provide the necessary visual richness without overwhelming the user.

2.3 LYRICS

Overview of the Visualization

This view invites users to explore the lyrical landscape of popular music through two complementary lenses: which words appear most often, and how wordy different genres tend to be. It's a dive into the vocabulary of music, revealing what artists sing about, and how much they say. At the top of the interface, a clean control panel allows users to dynamically filter the dataset. Sliders adjust key musical dimensions like *liveness*, *danceability*, and *release year*. A genre selector lets users focus on specific styles or compare across the full musical spectrum. Finally, a word count slider controls how many words are displayed in the word cloud, helping users zoom into core themes or uncover more nuanced language patterns.

Details of the Visual Components

The first visualization is a dynamic and colorful word cloud generated from the lyrics of all tracks matching the current filters. Each word is sized by frequency and assigned a consistent color via hashing, ensuring a visually varied but structured layout. Words like *love*, *yeah*, *baby*, and *get* often appear prominently, but as users adjust filters by genre or time period, new themes emerge. The cloud updates in real-time, providing immediate insight into how lyrical language changes with musical context.

Complementing this, a horizontal bar chart presents the average number of words per song across all genres. This quantitative layer adds depth to the exploration by showing not only what artists say, but how much they tend to say. Bars are sorted from most to least verbose genres, with selected ones highlighted to allow quick comparison. Hovering over a bar reveals precise average word counts, offering surprising insights into the balance of lyrical richness across different musical traditions.

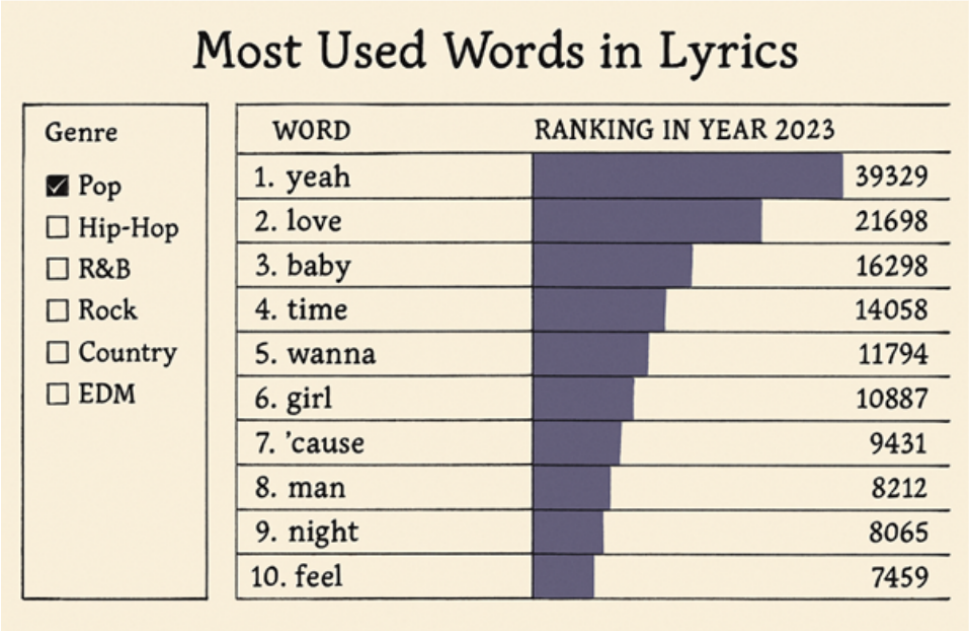

FIGURE 2.3
Initial visualisation sketch submitted in Milestone 2

Evolution from Initial Sketch

Initially, the plan was to keep it simple: a basic chart showing the most frequently used words in lyrics, filtered by genre. The idea was to recreate a classic "top 10 words in pop music" type of ranking, static, focused, and easy to interpret. However, as we developed the system, we realized the potential to go much

further. We decided to enrich the interface with dynamic filters for musical features such as *danceability*, *energy*, and *liveness*, to better connect lyrical patterns with the actual sound of the music. We also introduced *release year* as a key dimension, allowing users to explore how the vocabulary of popular music evolves over time. Finally, we added control over the number of words shown, helping users reflect on the density and depth of lyrics across different styles.

Design Challenge: Artist Gender Analysis

While designing this visualization, one early idea was to explore lyrical differences based on the artist's gender, comparing, for instance, the most common words used by male versus female artists across genres. This would have allowed for a rich investigation of how identity might influence language and themes in music. Unfortunately, none of the datasets we had access to included reliable information on artist gender, and attempts to infer it from names or external sources proved inconsistent and ethically questionable. As a result, we decided to focus on features that were directly available in our data, such as genre, word count, and musical characteristics, while keeping the door open for more identity-based analysis in future iterations, should better data become available.

Final Result

What began as a simple static ranking has grown into a full-featured lyrical exploration tool, transforming language itself into a powerful lens for understanding music.

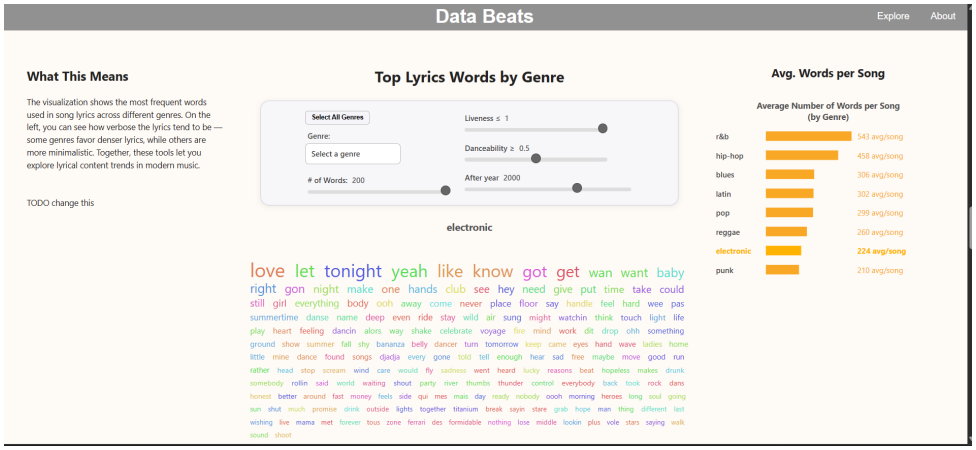

FIGURE 2.4
Final visualisation submitted in Milestone 3

2.4 GENRES-SUBGENRES

Genre to Subgenre: A Visual Flow of Musical Styles

This Sankey diagram visualizes the relationship between **main musical genres** (on the left) and their **subgenres** (on the right). Each colored flow represents how a genre connects to one or more subgenres, with color indicating the genre of origin.

For example, *hip-hop* branches into distinctive subgenres such as *drill*, *gangster*, and *trap*. Similarly, *latin* leads into rhythmic styles like *reggaeton*, *cumbia*, and *afro-latin*. Some genres, such as *electronic* and *pop*, link to a wide variety of subgenres, illustrating their stylistic diversity and cultural reach.

The goal of this visualization is to provide an intuitive way to explore how music genres evolve and split into more specialized subcultures. The flow layout helps emphasize both the diversity and the structure of

musical styles.

Process and Design Choices

To create this visualization, we began with a dataset of tracks labeled by genre and subgenre. We cleaned the data by filtering out broad or ambiguous categories such as *lofi*, *folk*, and *indie*, in order to keep the visualization focused and readable.

We also ensured that each musical style appears only once in the diagram, either as a genre or a subgenre, to maintain a clear and logical structure. The curated data was then converted into a JSON format compatible with D3.js and used to generate the interactive Sankey diagram.

Reflections and Future Improvements

One of the key challenges was finding the right balance between *data richness* and *visual clarity*. Including too many nodes led to visual clutter and made the diagram difficult to read. To avoid this, we limited the number of genres and subgenres to highlight the most meaningful relationships.

In the future, a *circular Sankey layout* could be explored to accommodate more connections while optimizing space. However, this comes at the cost of readability. For now, we chose a linear format to prioritize clarity and ease of exploration, especially for users less familiar with data visualization tools.

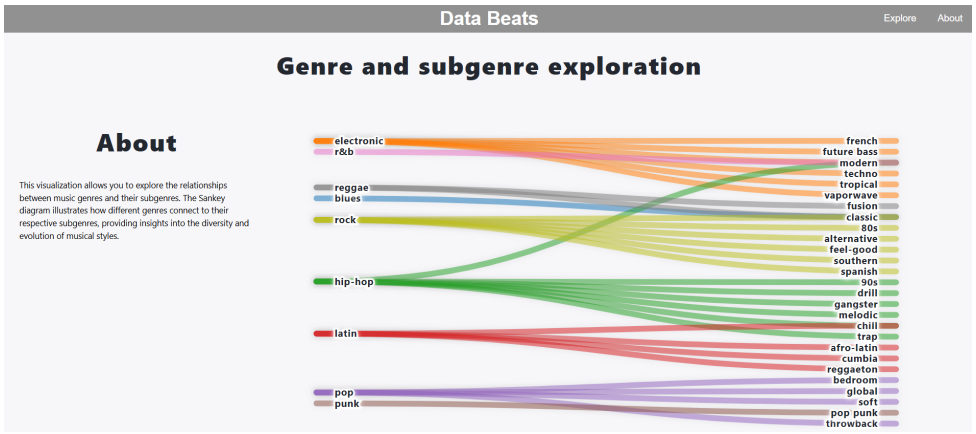

FIGURE 2.5
Sankey diagram

CHAPTER 3

PEER ASSESSMENT

We spread the work quite evenly. We all worked together in meetings for the first two milestones. Then for the final milestone, we each took one of the animations and worked on it on our own. Then in the remaining time that we had, we worked together to incorporate the visualisations into the website, do the screen cast and write this document. Here is a table summary of who did what.

Task	**Théo Houle**	**Caroline Verchère**	**Théo Lemonnier**
Website	X	X	X
Visualisation 1		X	
Visualisation 2	X		
Visualisation 3			X
Visualisation 4			X
Data Scraping	X		
Process Book	X	X	X
Screen Cast		X	X
GitHub, README	X		

TABLE 3.1
Task distribution among team members

The workload was really well balanced between the members, everyone worked to their strengths, and we all took the time to help each other when blocked.