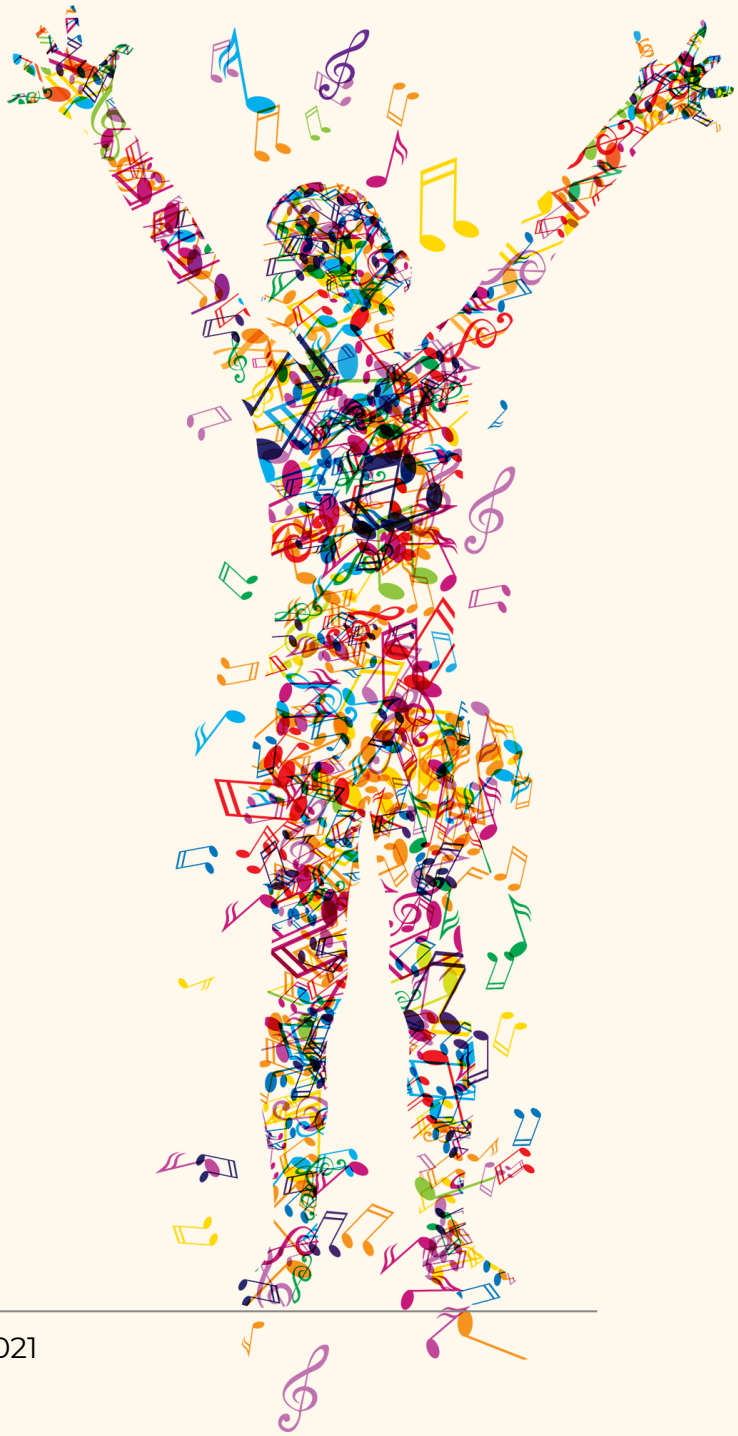
Process Book

# Music

# From Turntable to Spotify

**Lilia Ellouz**
**Faten Ghali**
**Yassine Khalfi**



COM480 Data Visualization - Spring semester 2021

EPFL

# Introduction

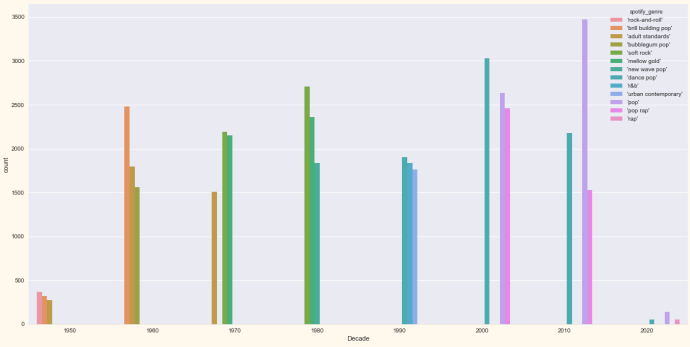
On average in the world, we listen to music for 2.5 hours a day; there are very few things we spend more time on than that every day. We all have a history with music and music has its own history. From the Beatles to Drake to the Bee Gees to Madonna to Mariah Carey, the music we listen to over the decades has changed. In our visualization we will try to tell you about the evolution of music through the decades and it will also be an opportunity to listen to some of the greatest hits of the last decades. This evolution of music will be studied through genres, musical features, milestones, and lyrics.

# Data Analysis

## Data Gathering

The initial dataset used for the visualizations is "**Billboard Hot weekly charts**" created by Sean Miller and hosted on data.world. It consists of every weekly Hot 100 singles chart from Billboard.com, as well as the audio features for each track provided by the Spotify Web API. We merged both files, and each row of the resulting dataset represents a song, the corresponding position on that week's chart, and several audio features such as 'Tempo', 'Danceability', or 'Speechiness'.

Several Billboard songs (*5'800*) did not have any corresponding lines in the Spotify features dataset, so we decided to directly scrape them from the Spotify Web API and managed to recover 98% of them. This scraping step required simplifying the names of the tracks and artists: characters like "&" or words like "Featuring" were discarded to get more matches between the rows.



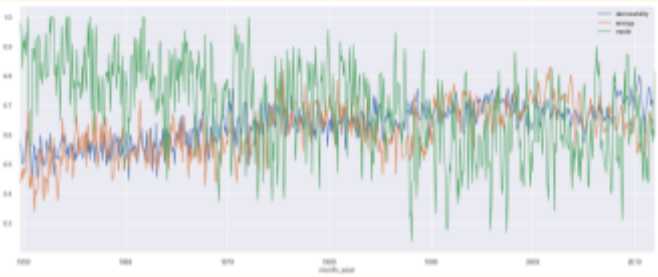

The lyrics were fetched from the Genius API. The Genius website is a bookcase containing song lyrics, news stories, sources, poetry, and various documents. For the sake of simplicity, we decided to get the lyrics of only the top 100 songs per decade. This step was challenging because a number of songs did not exist on the Genius platform and so the responses of the API were wrong. The errors of the Genius API (~5% of total songs) were fixed manually.

## Data Processing

**Genre study**: As a first step, we wanted to see how the most popular genres had evolved from the 1950's to nowadays, so we had a look at each decade's dominating genre.

**Feature evolution study**: Our initial inspiration for this part was this graph. We started by computing the average value of the numerical audio features for each month on the Billboard for the top 10 songs. However, that resulted in very noisy curves. To eliminate the noise, we decide to compute the average value *per year*, and for all 100 songs of each week's chart. This gave us smoother graphs.

**Longest lasting artists/hits/albums**: To generate the statistics used later on in the



*Average value per month for top 10 songs*

*Average value per year for top 100 songs*

visualizations, we simply grouped the complete data by artists, song IDs and albums and counted the most occurring ones. We did this for all ranks, combined, as well as for N°1 position on the charts.

**Sentiment analysis:** We carried out a semantic analysis based on categories of sentiments. We chose 5 categories of sentiment that are the most obvious: Love, Hate, Pain, Sadness and Joy. This analysis returns the percentage of appearance of the lexicon related to the chosen category among the input text. Our input text is the lyrics of songs of one decade combined. We plot the results to compare between the decades under each category.

**Word cloud:** We get the count of the 100 most common words in the lyrics of each decade after eliminating stop words. This analysis gives more insight on how the language evolves over time.

# Story

After our work on the data, we discussed how to turn the data we had into a story. Knowing that our objective is to give our users insights on the evolution of music, we chose to make visualisations that take into account the time parameter so that our users can see the evolution of genres and music.

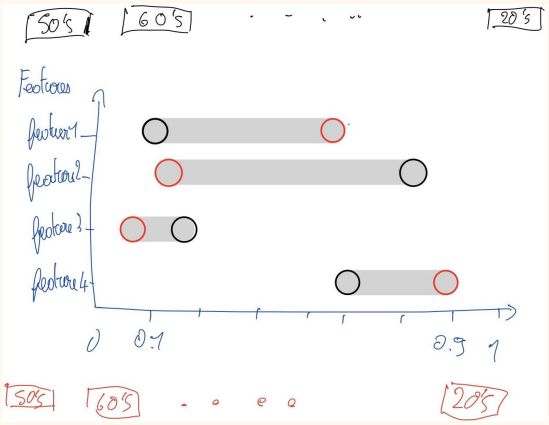
The user's journey on our site is made in several stages: we start with the playlist of the greatest hits (the opportunity to listen again or to discover some of them), then we dive into the evolution of major musical genres (pop and rock), afterwards we look at the evolution of features that have strongly evolved, and we end up being more interested in the lyrics of songs. Our aim was to tell the story of the evolution of music from different points of view, hence our choice to do it in several parts.

# Visualization

## Part 1: Genre evolution

### Initial plan

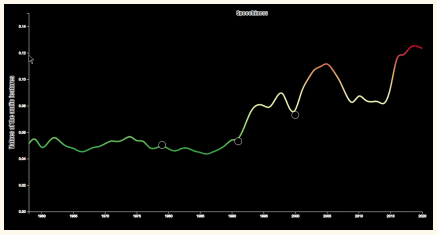
For the first visualisation we wanted to show the evolution of musical genres over the decades. After looking at various other visualisations, we quickly decided to go for a Cleveland dot plot. Depending on the two decades and the genre selected, the user sees two dots for each feature, i.e. one for each decade; showing the difference in values between the two decades. Later on we also chose a third type of button to select the genre of the music: Rock or Pop.




### Choices and implementation

For the implementation of the visualization that you can see on our website, we relied on the example of the interactive Cleveland dot plot of d3.js gallery. As far as interactivity is concerned, we have two CSV files regrouping the data of the genres and depending on what the user chooses (by clicking on the corresponding button) one of the two files is chosen. The decade buttons allow the user to choose the values to be displayed, by selecting the corresponding values within a file for a given genre. The buttons all allow the user to trigger update functions that will modify variables and redraw the graph.

## Part 2: Audio features, and what their history tells us about music

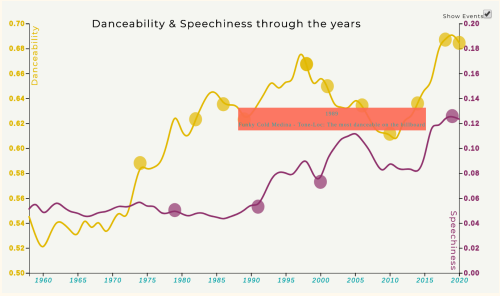


### Initial plan

This part was designed to show how audio features like song speechiness or danceability of the most successful hits changed,

on average, through the years. Initially, the idea was to overlay the main genres for each decade on the graphs so that the user can make the connection between the music type and the feature (e.g. more rap means more speechiness). However, this meant that, first, the visualisations would contain too much information, and second, some genres (like Pop) are so varied that the corresponding songs have very diverse audio features.

### Choices and implementation




On this visualization, two line graphs are present: danceability and speechiness. Since they are not on the same range, the left axis refers to danceability values, and the right one refers to speechiness. The lines transition from left to right when the visualization is loaded. As the user moves the mouse on the graph, a vertical bar appears and follows the position of the cursor, as well as the values that correspond to the year the user is pointing to.

Another feature is the interactive events: each circle corresponds to an event or a song that is linked to the displayed value (ex: electro music highlights for danceability). When hovering over the circles, an explanatory tooltip appears. For more ease of visualization, the user can choose to hide the events, and the circles faint with a slight chronological delay.

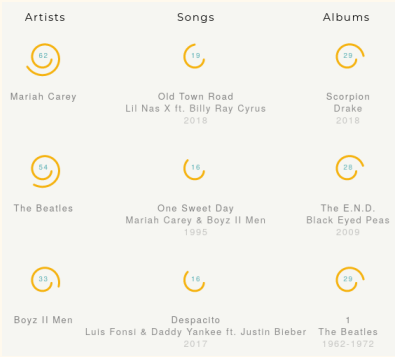
## Part 3: Key achievements & milestones in the Billboard Hot 100 chart

### Initial plan


This part intends to show in an intuitive way record-breaking numbers that artists, hits or albums have set. Our initial prospects for the visualization was that it would contain many kinds of achievements (number of singles, number of debuts, and other milestones), as shown in the hand sketch. Each spiral would represent an event, and the length of its path would depend on the number. However, we felt that keeping a unique unit (number of weeks) for all spirals would cause less confusion to the user.

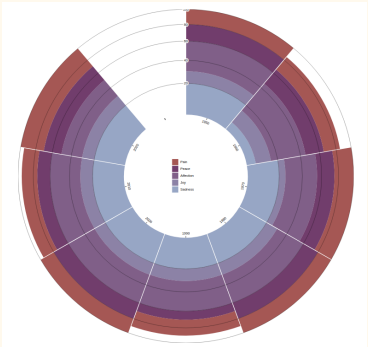
## Choices and implementation

Each spiral corresponds to the number of weeks spent on the Billboard for that artist, song or album. The longer the time spent, the longer the spiral. The first column represents artists, the second column songs, and the third is for albums. The rows go in decreasing order: the first row corresponds to more weeks than the one below, and so on. Using a radio button, the user can choose to look at the number of weeks spent on the Billboard at all positions combined, or only focus on the N°1 position. When the user changes focus, the spirals appear one by one (using .delay()) and the paths load from the center to the exterior. Using .tween and d3.interpolate, the numbers in the spirals start loading from 0 to their value until the spiral reaches its final length. During this transition, the descriptions under the spirals arrive (d3.easeBounce) at their positions.




# Part 4: Lyrics analysis

## Initial plan



A circular bar chart as shown in the figure was the initial idea of representing the sentiment analysis. Although the format is unusual and rather fun we did not opt for it because it's rather hard to read.

To look more closely at the language used, we had the idea to look at the most common words per decade and try to assign a word to each one.

## Choices and implementation

To show the evolution over the decades of sentiments in the lyrics, we chose to create a regular bar chart. It is simpler and easier to follow. The user is disposed of with five buttons representing the five categories of feelings. On click, the corresponding data CSV file is loaded, and the y-axis label is updated. We chose to keep the y-axis fixed, ranging from 0 to the maximum value of all the values (1.42%), even though some categories result in very short bars, because it also allows comparison between the categories overall (e.g. We seem to sing more about pain than about joy, all decades combined).


For the evolution of language, we opt for word clouds. They are more aesthetic and give more information about the lyrics. The word clouds are implemented using the [d3-cloud plugin] developed by Jason Davies. A CSV file is created for each decade to contain the words along with their occurences, and again, the file correct file is uploaded when the user clicks on the button of the desired decade.

## Website

Since our visualizations are more or less independent, we decided to vertically separate each section using Fullpage.js. This made our website “scrollable” from section to section. In addition to that, the first slide for a section often describes it, but the user can horizontally scroll (or use keyboard arrows) to progress through the section.
A consistent color palette was not easy to choose, as we hesitated between a light and dark color scheme, but ended up choosing a light/dark theme as (the text is more legible, and the colors stand out better / darker tones are more suited to the subject).
A challenging part was fitting all visualisations to the slides and making sure they were correctly displayable on any screen size independently of their width and height.

## Peer assessment

**Yassine**: Data Gathering, Processing songs features, analysis of musical genres present in the data, evolution of genre chart, playlist embed, website structure, module integration.

**Lilia**: Spotify scraping and merging missing data. Choice of feature analysis. Audio features chart. Research of main events for danceability and speechiness. Spiral (Key achievements) graphs & corresponding data exploration.

**Faten**: Gathering and processing lyrics. Lyrics-related visualizations: bar charts of sentiment evolution and word clouds.

**Everyone**: Overall outline of the website, process book