Data Visualization | Fall Semester, 2019 | Group Project

Spotify | Music Trends

M20190684	Abdallah Zaher,
M20190303	Cristina Mousinho,
M20190925	Gabriel Ravi,
M20190188	Tobias Kutscher

Dashboard Available on: https://spotify-dash-project.herokuapp.com/

Table of contents

Introduction

2019 just ended and, just like in the previous years, Spotify's users were gifted with a playlist of their most streamed tracks. With it being the end of the decade, *"2019 Wrapped"* came with not only the usual playlist, but with interactive visualizations of a user's activity (top-streamed artists, music genres, …).

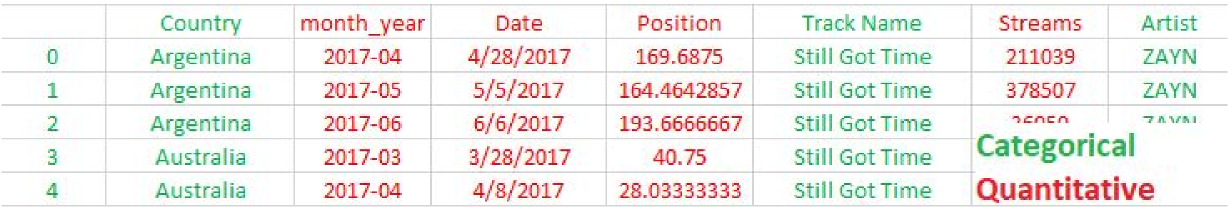
All four of us are Spotify users, from four different countries, three different continents, and with individual music tastes. We now know about our preferences, but not about the rest of the world's. And that's exactly what inspired us to do a dashboard about music trends.

So we extracted some real data from Spotify's official website over a specific period of time (from January 2017 to March 2019) for a variety of 62 different countries aligned with the top 200 monthly tracks per each one of them.

> **Dataset from:** [Spotify | Charts]

Visualizations: interacting and reading

We should first acknowledge our data from their data type to their usage. For instance, we're going to provide this example to reveal a summary about it.



	Country	month_year	Date	Position	Track Name	Streams	Artist
0	Argentina	2017-04	4/28/2017	169.6875	Still Got Time	211039	ZAYN
1	Argentina	2017-05	5/5/2017	164.4642857	Still Got Time	378507	ZAYN
2	Argentina	2017-06	6/6/2017	193.6666667	Still Got Time	26050	ZAYN
3	Australia	2017-03	3/28/2017	40.75	Still Got Time		
4	Australia	2017-04	4/8/2017	28.03333333	Still Got Time		

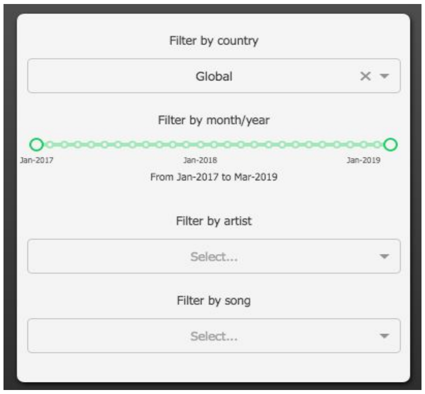
Categorical

Quantitative

Figure 1: Data and attribute overview

When looking at our dashboard, you will immediately see that it is divided into four main containers. As for design, we combined the black background and labels with the green color which were inspired by Spotify's official logo, a simple combination of colors that is clear and user friendly.

The first container (top left), is composed of three dropdown menus where you are able to choose a country, an artist and a song. In addition, a range slider was added so the user can set a period of time (of at least one month) that falls between January 2017 and March 2019. Note that when choosing a country, some filtering mechanisms were placed to reduce the options regarding the artists and the songs dropdown menus where only the artists that have been streamed in this specific country will be visible.


Filter by country

Global

Filter by month/year

Jan-2017 Jan-2018 Jan-2019

From Jan-2017 to Mar-2019

Filter by artist

Select...

Filter by song

Select...

Figure 2: Selection Box

Our data is spatial, as it is related to the distribution of the streams over the world. For this specific reason, we decided to go for a choropleth map in the second container aligned with the channel of the color scale to assist the user interpreting the dense of the streams per country. We kept the color varying from light green to black-greenish to maintain the coherence in the dashboard. The more times a country's users stream a certain artist and/or song, the darker the color will be. Color saturation was chosen as the channel for the map, even though it is not one of the most effective channels in general. Changing the respective country size on the map in regards to the attribute, for example, would not be appropriate for the target audience of the dashboard, which are mainly all spotify users, as it would be too complex to understand. As we only represent one attribute (the number of streams) color saturation has been justified for representing the attribute effectively, especially with the combination of the hover-over data presentation box.

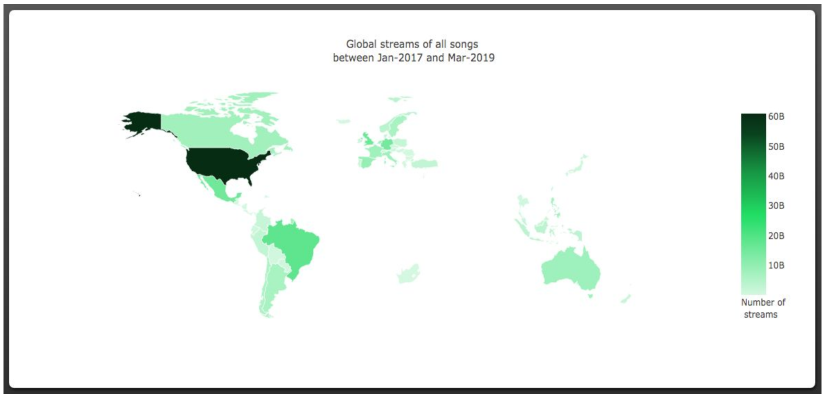


Figure 3: Choropleth World map

Because we're talking about a world map, we're also encoding data using space: white lines separate country borders. If an artist is chosen, you'll be able to see in which countries their tracks were streamed. Same goes with songs. Every time a new country is chosen, all the data in the dashboard will reset and action will be taken upon this specific new selected country. As a result, the time period is set to be between the first month and the last month that is related to this country as provided in the dataset, and the artist/song values are wiped. But that's not all. A tap feature was implemented to enhance the user's interaction with the dashboard. Whenever the user clicks on a specific country on the map, this country will become the selected one and the dashboard will interact with this selection accordingly.

On the third container (bottom left), a bar chart will give you the opportunity to check a country's top ten artists. With an artist chosen, you can check their streamed tracks in that specific country. The bar chart presents them in descending order of stream amount, meaning the first bar will represent the most played song in the country.

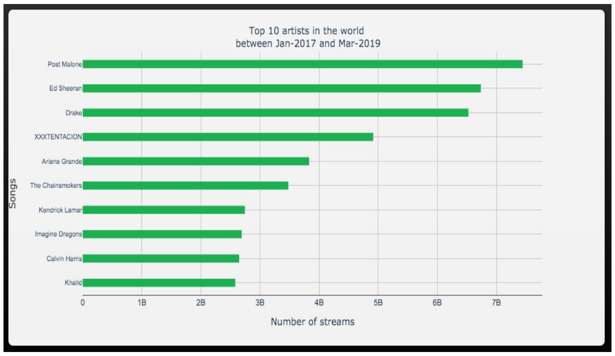


Figure 4: Barchart

Number of streams and artist name/song title are encoded using a line mark with the vertical spatial region channel for the first, and the horizontal spatial position channel for the latter. The line area (depending only on length since all bars have the same width) helps to visualize differences in stream quantity. If you then decide to choose a specific song, it will be highlighted in a dark grey (instead of green) color, meaning a new categorical attribute is encoded by adding color to the chart. The color channel was used as a "popout".

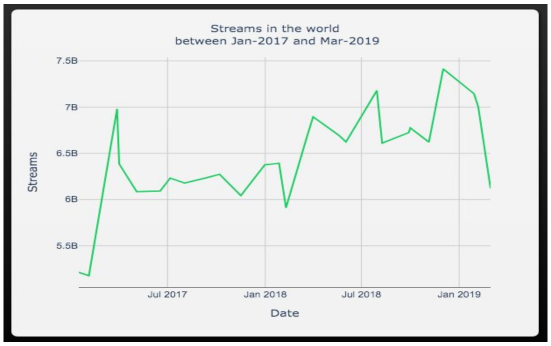


Figure 5: Line chart

And finally, on the fourth one (bottom right), you can analyze how stream amount changes over time. With a country chosen, you'll see how its user's stream amount changes over time. That time frame can then be reduced with the help of the range slider. With an artist chosen, the line chart will update to the number of streams their tracks got (in total) over time. After choosing a song, the plot updates to the number of streams it got over time. Our data encoding choices are the usual for line charts: point marks are ordered by the month/year they represent, and connected by lines between them. Vertical spatial position relates to stream amount. A color was added to keep the coherence with the 'hover' option to know the exact value if requested by the user. If no values are provided, all plots show Global values from January 2017 to March 2019, with no relation to a specific artist or track. In conclusion, we used the position on common scale with length for ordered attributes and spatial region with color hue for categorical attributes. Those attributes allowed us to achieve:

- Accuracy: The user has the option to know all the exact values by hovering over the graphs or the map.
- Discriminability: Specific values are generated upon the user's decision.
- Separability: All the graphs are related based on what the user chooses. But every graph is encoded separately and changes by the variation of the country, year, artist, or song. The channels used do not interfere with each other, as all plots only represent one attribute.
- Popout: Hover option is added over all the graphs plotted so the user has the option to extract the exact data number. When selecting a song, the barchart uses color popout to represent the selection
- Grouping: As mentioned before, all the graphs are related based on what does the user choose and they change by the decision variation of the country, year, artist, or song. The line chart uses grouping by connecting the dots of the 2-dimensional space, which are representing the amount of streams, using lines.

Technical aspects

As far as our project implementation, we followed the requirements, using Plotly with Dash software. When opening our project folder, you'll be able to examine our code in the file "app.py". The text file "requirements" contains the packages versions needed to run the code. As for the folders: "assets" has our CSS file (style.css) and Spotify's logo (used in the dashboard); "data" only holds the used dataset; and "screenshots" is composed by a screenshot of our dashboard, so you are sure you're facing the expected appearance. Regarding our code, we began by cleaning and preparing our dataframe using Pandas library. We moved on to building our app layout. As we have previous experience with HTML and CSS, it was easier to do it and to style it just as we desired. Plotly has default styling that sometimes can't be changed using only the plotly layout parameters, so we had to overwrite some class styling properties. To ensure our users' choices of country, time period, artist, and song don't clash, we had to develop a series of functions that guarantee that every time a new value is chosen, all the available options for the range slider and dropdown values are updated accordingly. And finally, we built our three plots (choropleth map, bar chart and line chart), all of which change based on the user's input.

> **Code available in: https://github.com/TobiasKutscher/spotify-dash**

Validation

To guarantee our dashboard behaves as expected and leads to a clear interpretation of the information it represents, we do consider some validation levels to be relevant.

- Visual encoding/interaction idiom: as far as the first one goes, we've justified our choices above in this report. To find potential ways our dashboard might not work, an evaluation with experts may be a good option. They might also lead us to better representation approaches. We also think we could benefit in doing user testing: asking someone how many times Ed Sheeran's "Perfect" was streamed by portuguese users during the year 2018 would give us many insights. How many clicks does the user need? How long do they take? Do they use the dropdown menu or the map to select Portugal? If they use the dropdown menu, do they scroll or do they type the country and select it? Is it easy to set the range slider from January 2018 to December 2018? How many attempts do they need? Do they look at all plots or do they immediately realize the answer will come from the bar chart? Do they hover over the bar for an exact answer or do they estimate it?

- Though in a further step, task and data abstraction might assure us that our dashboard provides new knowledge that is both interesting and relevant to Spotify's users.
- The performance of the dashboard can be checked by measuring the time for loading the data for every selection.

Discussion

The ability to choose and use a Spotify's dataset for sure worked in our favor. We were passionate about our project theme from the beginning, so picturing our dashboard was always easy. However, at times we felt like the software limited our capability to develop some of the envisioned aspects. For example, at a certain point, we realized that when a country is chosen, no changes are produced in the choropleth map. So we tried adapting what we did with the bar chart, and change a country's color when selected. Sadly, that feature isn't yet developed by Plotly, and our CSS knowledge wasn't enough to tackle it either.

That is definitely a task that we might be able to accomplish in the future, if our web developing skills progress. Putting validation in practice would also be interesting, as we would be able to see how others put it to use. In the meantime, we'll continue exploring our dashboard and gaining more knowledge on music trends all over the world!

References

Zaher, A., Mousinho, C., Santos, G., & Kutscher, T. (2019). Group project - Spotify | Music Trends. Heroku.
https://spotify-dash-project.herokuapp.com/

Kruchten, N. (2019). Plotly datasets. GitHub.
https://github.com/plotly/datasets/blob/master/gapminder_with_codes.csv

Spotify: Top 200. (n.d.). Spotify | Charts. Retrieved December 13, 2019, from https://spotifycharts.com/regional

Teal, S. (2019). How to visualize Spotify music trends in Tableau. Tableau.
https://www.tableau.com/about/blog/2019/7/how-visualize-spotify-music-trends-tableau

Gonçalves, D., Fonseca, M. J., & Campos, P. (2017). Introdução ao Design de Interfaces (Third). FCA.
https://www.fca.pt/pt/catalogo/informatica/design-multimedia/introducao-ao-design-de-interfaces

Munzner, T. (2014). Visualization Analysis and Design (First). A K Peters/CRC Press.