Data Visualization for Scientists

Final Report for the PhD Course

Luca Giuliani
Department of Computer Science and Engineering
luca.giuliani13@unibo.it

1 Introduction

Data The data used in this report were retrieved from the Spotify for Artists web application. It consists in the daily *streams*, *listeners*, and *followers* of a single artist. Since the web application does not provide an API to download the data, four different `.csv` files were manually retrieved: one of them – `public.csv` – contains the overall number of streams, listeners, and followers for the artist profile; the remaining three – `bolo-by-fight.csv`, `asfalto.csv`, and `natale-con-i-tuoi.csv` – contain the number of daily streams per song, respectively. These files are publicly available at the following link: `https://github.com/giuluck/SpotiViz`.

Code Along with the data, the previous github repository contains the code used to generate the visualizations. The data were processed using the `numpy` and `pandas` packages for python, and eventually plotted leveraging `matplotlib` and `seaborn`. All the plots were exported in `.pdf` format in order to preserve high quality. We use the `whitegrid` theme and `poster` context from `seaborn` along with its default *Sans Serif* font type and a custom color palette whose choice is motivated later on in this section. The choice of the `poster` context is motivated by larger font sizes, which improve readability especially in case the plots are printed on paper. All the additional plot-specific choices – e.g., wider ticks, integer ticks, x and y labels – have been made to increase readability and ease of understanding.

Motivation The Spotify for Artists web application lacks customizable visualization tools. In fact, it is only possible to visualize the evolution of song's streams separately, or to visualize the total number of daily streams without any information about the source. In addition, the web application offers no possibility for smoothing data, and it allows for three possible visualization periods only, i.e., *last 7 days*, *last 28 days*, and *from 2015*. Finally, Spotify does not have access to further information which can be valuable in detecting anomalous patterns such as promotional periods, live shows, social media trends, etc. For all of these reasons, the idea of using this data source aims to enable better analysis of song streaming patterns through simple visual inspection.

Common Choices Among all the proposed visualizations, three common choices were made. First, the color palette has been customized according to the song covers. Figure 1 shows the three covers, hence the palette is composed of *Magenta* (`#FF0072`), *Cyan* (`#00A2A4`) and *Yellow* (`#FFC801`), respectively – data merged from all the songs are instead represented in black. Second, two different plots are reported for each of the four views. They represent, respectively, the evolution of the raw data – on a daily basis –, and of the same data smoothed via a centered 7-days moving window with cosine moving average. The purpose of this choice is to allow for inspection of both high-frequency patterns – e.g., rapid spikes in the number of streams –, and more stable patterns that do not

suffer from regular weekly fluctuations. Additionally, the same *y-axis* is used for the two parallel plots to facilitate comparison. Third, since this is a time series, the *x-axis* always represents the time dimension. However, while in some plots time is absolute and thus represented as a *date* object, in others time is relative to each song, hence represented as the number of *days* elapsed from the release date.

Figure 1: The three songs covers, from which the color palette was built.

2 Visualizations

Daily Listeners Figure 2 shows the number of *total* daily listeners – since the original data are not segmented for source, i.e., the songs. This simple *lineplot* is enhanced by two additional information: (1) the release date and (2) the promotional periods of each song. The purpose of the highlighted information is to allow easier understanding of spike patterns in the data, which clearly exhibit strong correlations with these events. Furthermore, the importance of having both a raw and an averaged view becomes evident depending on what is being analyzed. Indeed, if we are considering the evolution of song streams around release dates, the raw preprocessing is more informative, as it allows us to see the audience response on a fine-grained, day-by-day scale. On the contrary, smoothing out with a 7-days moving window makes it easier to quantify the promotional impact, as it cancels out the weekly trends that result in a drop of listeners in the middle of the promotional period – indeed, such a drop is likely to fall on a working day, while the number of listeners is expected to peak during the weekend.

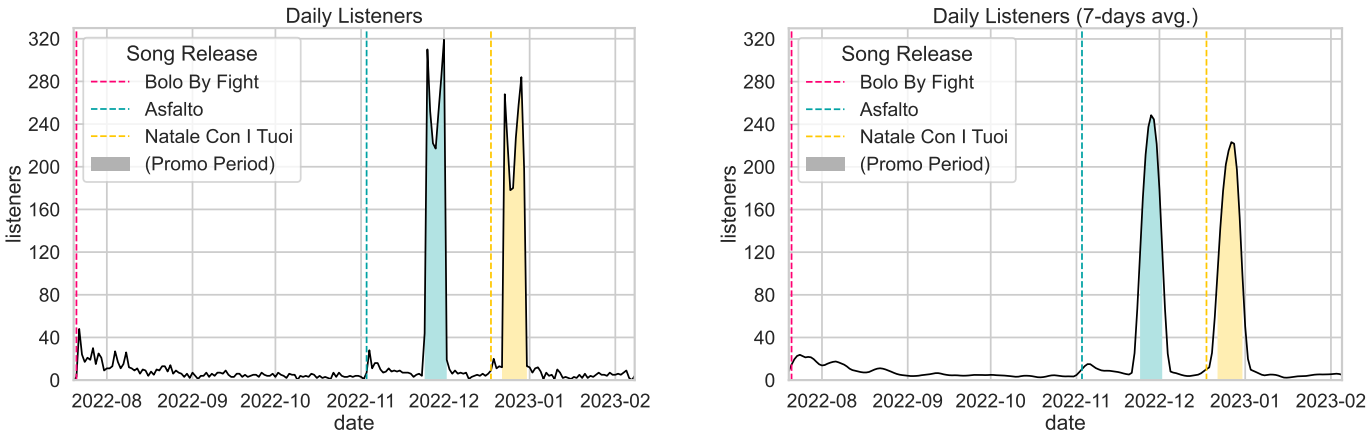

Figure 2: Number of daily listeners for the entire set of songs with (*right*) and without (*left*) 7-days cosine moving average. Release dates are highlighted with a dotted vertical line, while promotional periods by filling the area under the curve with (a lighter version of) the color representing the promoted song.

Daily Streams As a second visualization, Figure 3 shows similar data to those above, this time relating to individual streams rather than listeners. Comparing the silhouettes in Figure 3 with that in Figure 2, it can be seen that the data show similar patterns. However, the peaks are more pronounced in this second visualization, particularly at release dates. This can be interpreted as a sign that, during these periods, audiences are more likely to play the same song multiple times – another evidence comes from the fact that this effect is less distinct in the averaged data. Also, unlike previous data, this was segmented by source. The visualization benefits from this additional information, as data coming from each songs can be stacked using different colors. The daily streams are stacked vertically for two main reasons: on the one hand, stacking the data in front of each other would result in a less readable plot; on the other hand, this choice allows for an upper silhouette that represents the number of *total* streams – across all songs – while still having an insight of which one is the most listened to. The vertical dotted lines present in the previous plots are no longer necessary since the same information can be retrieved from the stacked data; instead, the promotional periods have been suppressed for ease of visualization.

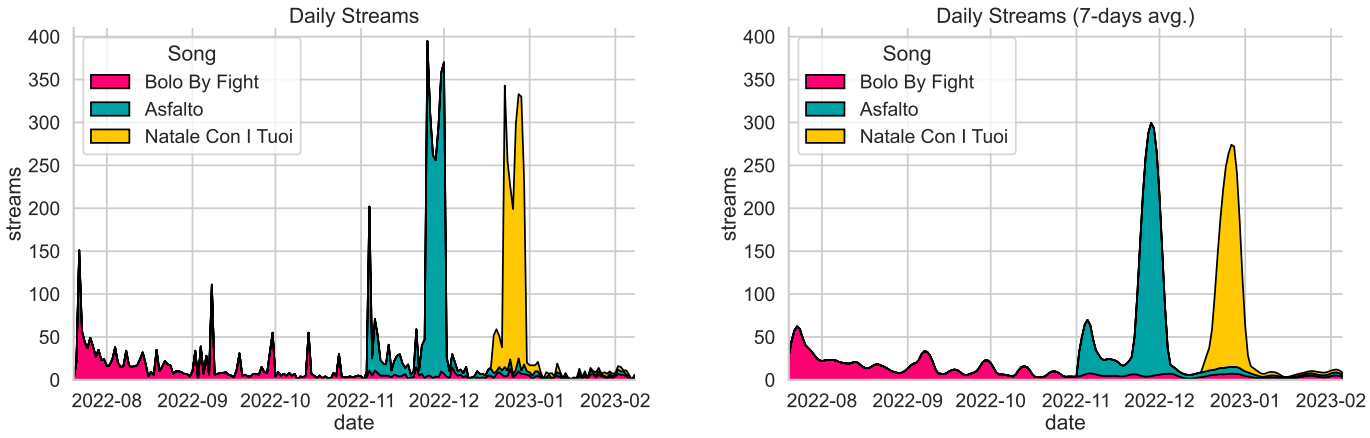

Figure 3: Number of daily listeners stacked per source song. The *left* plot is obtained from the raw data, while the *left* one is obtained via preprocessing the data using a centered 7-days moving window with cosine moving average.

Compared Streams from Release Date While the previous visualizations focused on the evolution of engagement for all songs, the following two are designed to allow for a *comparison* between songs. For this purpose, *three different lines* are drawn in Figure 4, representing the number of streams obtained by each song in the first *six weeks* from release date, respectively. All lines start from the value of *zero streams* on *day zero* – representing the day before release – and proceeds for the next 42 days. The promotional periods are highlighted as in Figure 2, and the ticks on the *x-axis* are set at one for each seven days in order to clearly mark the weekly evolution from the release date – i.e., day one. This visualization allows us to draw some considerations about the songs. Indeed, the last released song – *Natale Con I Tuoi* – shows significantly lower engagement in the first days after its release. This can be explained by two main factors: first, the physiological dampening of interest for subsequent single releases; and second, a piece of information that cannot be inferred from the plots, namely that this song was released on a Monday rather than on a Friday, as it is usually the case. Finally, after one month from the release date, all songs stabilize on a small but steady flow of streams per day.

Followers Gained from Release Date The final visualization, shown in Figure 5, displays the cumulative number of followers gained since the release date. Clearly, while streams are song-dependent and are inherently segmented, this is not the case for followers as well. Nevertheless, it is interesting to compare these data for the same reasons as before, i.e., to see whether there are interesting patterns around release dates and during promotional periods

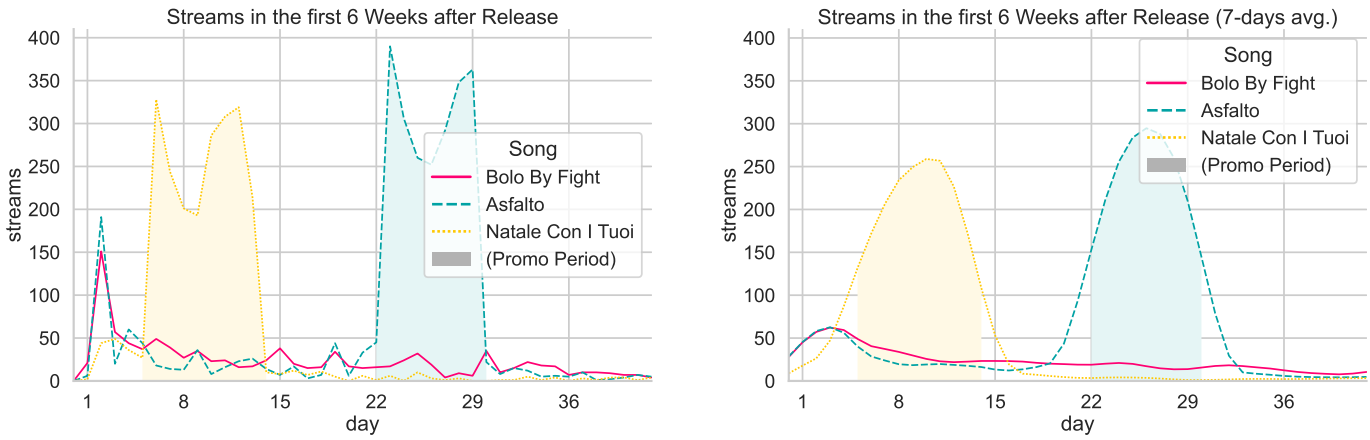

Figure 4: Comparison of daily streams of all songs in the first six weeks after release date. Again, the *left* plot shows raw data while the *right* one shows averaged data, and promotional periods are highlighted by color filling.

or not. The *x-axis* is structured as in Figure 4, with evenly spaced ticks every seven days, and the promotional periods have been highlighted with a lighter version of the song color as done in the previous visualizations. The first thing that can be noticed is that the strong spikes in daily streams exhibited in Figure 4 during promotional periods are not reflected in the number of followers, which increase by only one or two units – moreover, it should be noted that reporting an increase in the number of followers during promotional periods is a necessary but not sufficient condition to conclude that this increase is *due to* promotion. In addition, the number of followers gained near release dates decreases as new songs are released. Again, this is an expected phenomenon, since the first release always attracts more people as the Spotify profile is created from scratch.

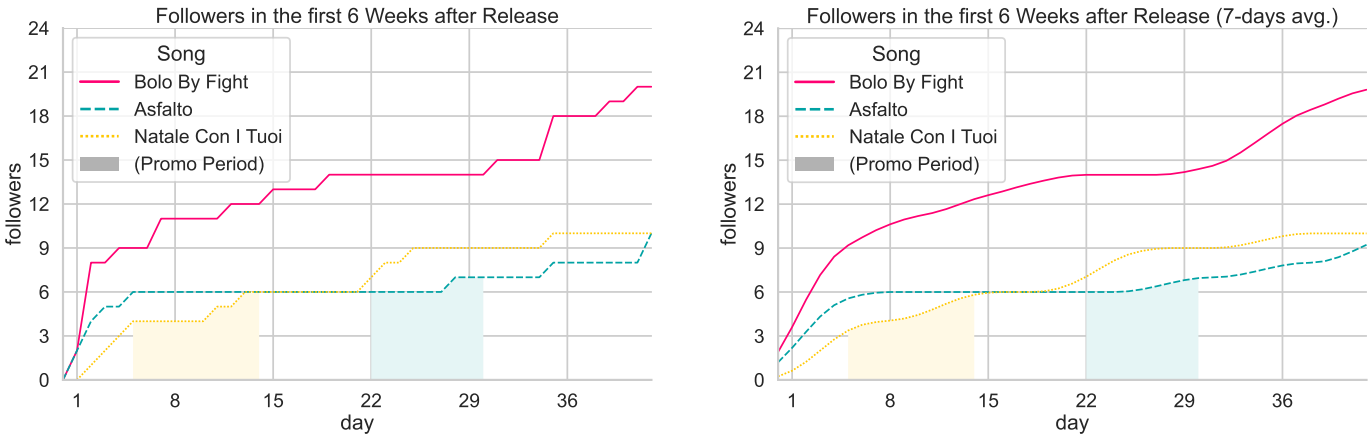

Figure 5: Cumulative number of followers gained from the first six weeks after release date. As usual, the *left* plot shows raw data while the *right* one shows averaged data, and promotional periods are highlighted by color filling.

3 Improvements

Based on the **Motivations** explained in Section 1, which guided the design choices, the proposed visualizations could benefit from two main improvements.

First of all, it might be useful to zoom in and out over different *time periods*. This is especially true because data can grow both horizontally and vertically over time, thus inspecting the trends from the first to the last day can be particularly cumbersome. Similarly, the possibility to define a custom *window size* can be convenient for analyzing long-term trends. The choice of a 7-days moving window comes naturally since the data are generated by human activities, hence it follows a strong weekly trend; however, over the long term it might be interesting to remove monthly and quarterly fluctuations in order to check for more coarse-grained patterns.

Both improvement would require the effort tp build a dynamic website where users can interact with the plots using, e.g., a *slider* to change the window size, and a *scroll bar* to inspect different periods of the year. This would guarantee a higher level of customization and more powerful means for the visual analysis.