Data Visualization

Second Exercise:
Design of a New Interactive Data Analysis Tool

Music Explorer

Group Members:
Azeez Abdikarim, Béla Bönte, Eleonora Renz

2022-01-26

Table of Contents

Problem Characterization in the Application Domain

The three questions that inspired this dashboard, are the following:

1. How has the profile of music changed over time?
2. Are certain years of music more homogenous than others?
3. Are trends in popular musical styles cyclical or seasonal?
4. What attributes of a song are correlated with one another?

A dashboard providing a visualization of this data would be incredibly valuable for music labels and artists trying to identify trends in popular music. For music labels, this tool could help them provide stylistic suggestions to their artists, or even determine the artistic attributes they hope to find in future label signings. Artists could also use this tool in an exploratory manner to dissect historical trends, and see if there are elements that could be revived for the modern market. As is often said, "what was once old becomes new again", so historical analysis could help artists influence the future sound of music.

Beyond the use of artists and music industry professionals, this dashboard is also designed to appeal to the curiosity of the average person. Music has a unique ability to stimulate an individual's memories/emotions, and scientists have coined the term neural nostalgia, to refer to an idea that as people age, they tend to prefer the music of their teenage years. New music may sound like 'nonsense' in comparison to the 'classics' of their youth. While the field of psychology would suggest that this phenomenon occurs because the music of your teenage years are entangled with youthfulness and positive memories, what if this is actually the result of stylistic changes? A casual user could use this dashboard to learn about the stylistic signature of their favorite time period, evaluate how popular music has evolved since that time period, and find the closest matching time period.

Data and Task Abstraction

The dataset chosen to answer the posed questions by visual means is *Spotify-Data 1921-2020 - Audio features of songs from 1921-2020,* which can be found on Kaggle. The data is given in CSV format and contains over 160,000 songs collected from the Spotify Web API, organized as a table. Each year between 1921 and 2020 has the top 100 songs. The attributes in the dataset range from categorical, such as for example artist, key (representing octave encoding), and name, to numerical values on the characteristics of a song, such as acousticness, popularity, danceability, and many more. In total there are 19 columns and 169909 rows. A detailed explanation on all the available features can be found in the *Introduction* section of the tool.

The goal of this visualization is to illustrate the changing nature of popular music over time and the primary key used to tell this story is the attribute Year. This key can filter the dashboard to provide visualizations that summarize a particular year, and also filter in a hierarchical manner to select the window for a time series graph showing quantitative attributes. Corresponding to each key (Year) is a set of values, with each item being a row representing an individual song. For each song, certain continuous and spatial attributes are aggregated to the level of 'Year', and are used to populate the time series and time slice visualizations.

For music industry professionals, the main usages of the tool are to consume information on the music market trends, discover which attributes made music popular in a given year, and to present information on these things to artist when discussing the next song production phase. As a mid-level goal, music professionals will want to search in a lookup and browse fashion. When analyzing attributes of certain years, this information is looked up and when trying to understand the specifics of past musical years, the tool is browsed in the relevant plots. The types of queries that are of interest to musical industry professionals are to compare music profiles over years and to summarize the profiles.

For casual users, being those that would approach the tool without any preformulated questions, this tool is intended to quickly communicate the idea that the profile of "popular music" has changed over time. The default views of this dashboard provide a high level depiction of these trends, and the interactive features allow the user to explore specific time periods, as well as specific musical features. The use of this tool is to enjoy and satisfy their curiosity. This use case falls under the consumer action. The kind of searches a casual user would do are mainly explore and browse. Users compare the data visualizations across different years in their exploration, this is the type of query used by them.

Interaction Level and Visual Encoding

Task 1: How has the profile of music changed over time?
To allow the user to see if the profile has changed over the years, a graphical representation is chosen using a scatterplot and a trendline. This shows the average values of the respective attributes (e.g. danceability, acousticness) per year.

Figure 1: Scatterplot over years

In order to see if the data in general fluctuate a lot, boxplots were formed from the aggregated years.

Figure 2: Boxplots

For both these visual representations, the following properties exist:
In order to recognize the change of the data in certain time intervals, an exact time interval can be defined by using a slider.
So that the user can quickly see which value of a song attribute it is, the buttons and the attributes (scatterplot & boxplot) are in the same color. Additionally, this can be read in a legend.
To avoid information overload, attributes can be removed and added as desired using the color-coded buttons.

Task 2: Are certain years of music more homogenous than others?
Homogeneity within a year can be determined by looking at the boxplot of the attributes of all songs released in a given year. The closer the quartiles are to the median, the less diversity there is along that specific attribute. On the *Year Slice* tab, a user can compare the boxplots between two years, and determine whether the popular music of one year was more homogenous than that of another year.

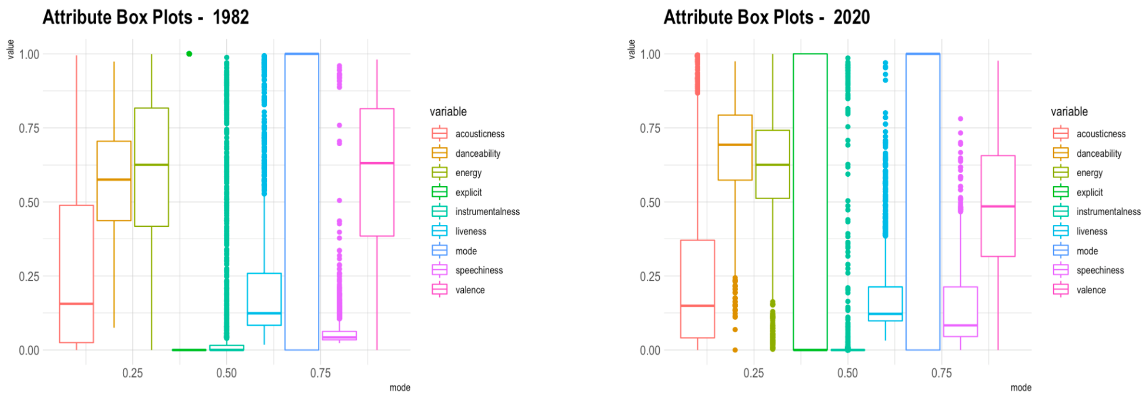

Figure 3: Boxplots comparing years

Task 3: Are trends in popular musical styles cyclical or seasonal?
A conclusion regarding this question can be arrived at by studying the default time series graph of yearly attribute averages. The highest level of data granularity here is by year,

therefore cyclical trends must be observed with the scope of multiple years, or even decades. When reviewing these attribute trends on the range of 1920-2020, certain attributes like ‘acousticness’ and ‘energy’ look as if they could be following a cyclical pattern, however after only 100 years, none have completed a full cycle.

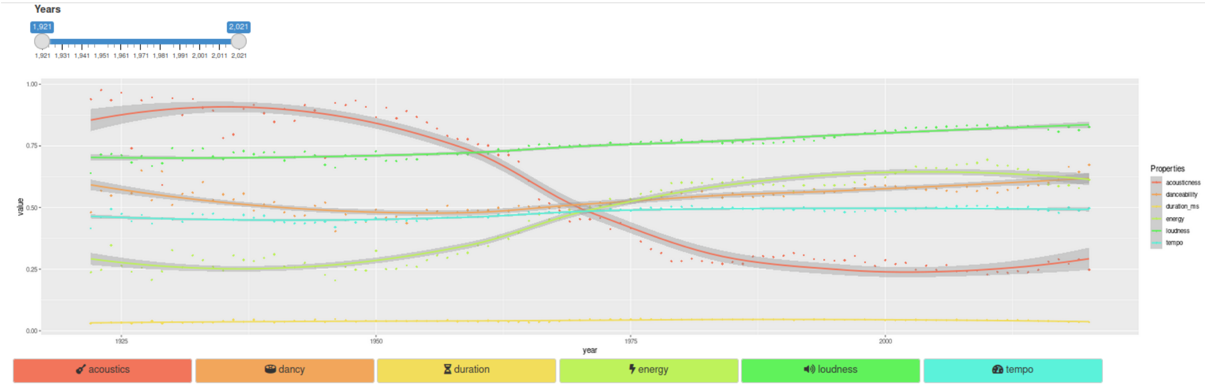

Figure 4: Scatterplot over years

Task 4: What attributes of a song are correlated with one another?
Users can investigate how attributes of songs are correlated with one another by inspecting the *Year Slice* tab. On this tab, they find a correlation plot made up of the recorded attributes of a song. This correlation chart uses a diverging color scheme to highlight the difference between positively and negatively correlated features. One of the most consistent correlations observed is that songs with high ‘acousticness’ are negatively correlated with the attributes of ‘loudness’ and ‘energy’. Consistent positive correlations that are observed are that high ‘energy’ songs also have high ‘loudness’, and high ‘valence’ (happy) songs correlate with high ‘danceability’. The validity of these correlations can be confirmed by self experience and intuition.

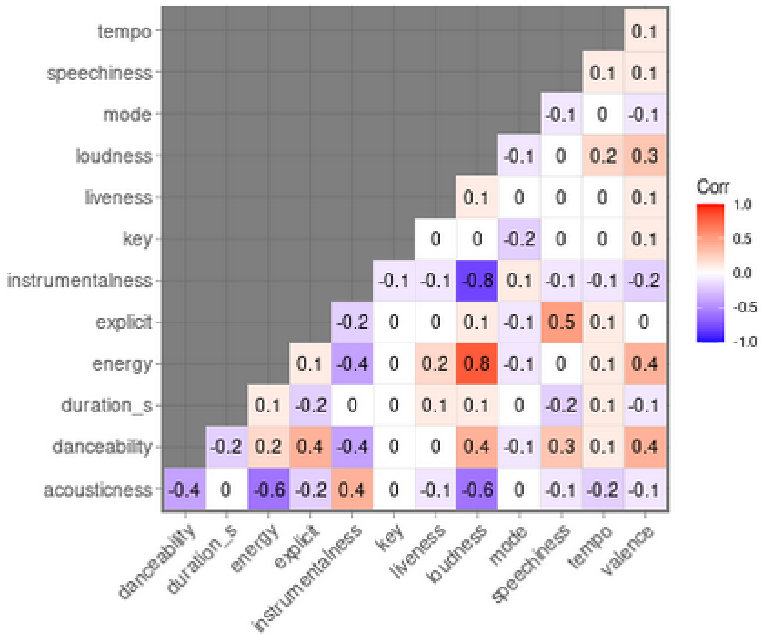

Figure 5: Correlation Plot

Algorithmic Implementation

Data Cleaning and Preparation

Min-Max-Scaling: Mix-Max scaling was used to make the values comparable. So some Spotify internal values were already encoded between 0-1 (e.g. danceability). Other values like loudness or tempo are stored in their natural metric scales. These values have been scaled so that the values can be meaningfully displayed in a plot.

Filtering and/or Aggregation

An example of a filtering technique used, that the user can interact with in the tool, is the slider bar, with which the user can change the year range displayed in several different plots. Additionally, the dataset is accessible through a filterable table.
Logically, on the programmatical side, aggregations are executed when giving the mean in the scatter plot and the box plots.
For manipulation of the visualizations, the user can select variables to focus on.

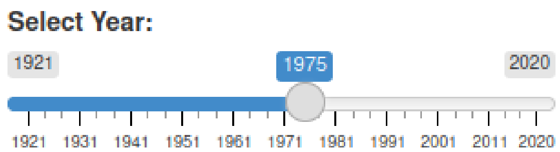

Figure 6: Filtering on year

Show 50 entries Search:

	name	artists	popularity	year	acousticness	danceability	duration_s	energy	explicit	instrumentalness	key
	All	All	A		All	All	Al			All	
1	Singende Bataillone 1. Teil	['Carl Woitschach']	0	1928	0.995	0.708	158.648	0.195	0	0.563	10
2	Fantasiestücke, Op. 111: Più tosto lento	['Robert Schumann', 'Vladimir Horowitz']	0	1928	0.994	0.379	282.133	0.0135	0	0.901	8

Figure 7: Datatables Plugin

Tables are one of the simplest and most effective tools of visualization. They help to represent individual entities. The items in a table contain attributes. With the help of the datatable, items can be sorted by a specific attribute. The attributes are sorted according to their quantitative size. Furthermore, combinations are also possible, for example, show only data from 1930 and sort descending by popularity

Use of Multiple (Synchronized) Views

The page displayed under the tab *Year Slice* shows both a radar chart along with a correlation plot, and boxplots summarizing the year's attributes. These three views are simultaneously updated in a reactive manner when the year selection slider is moved.

Mirroring these graphs, and on the same tab, the same visualizations are repeated with their own filtering slider. By duplicating these visualizations along with the slider input, the user can compare any two years directly with one another. Presenting these visualizations in such close proximity to one another allows one to make the comparisons easily and directly.

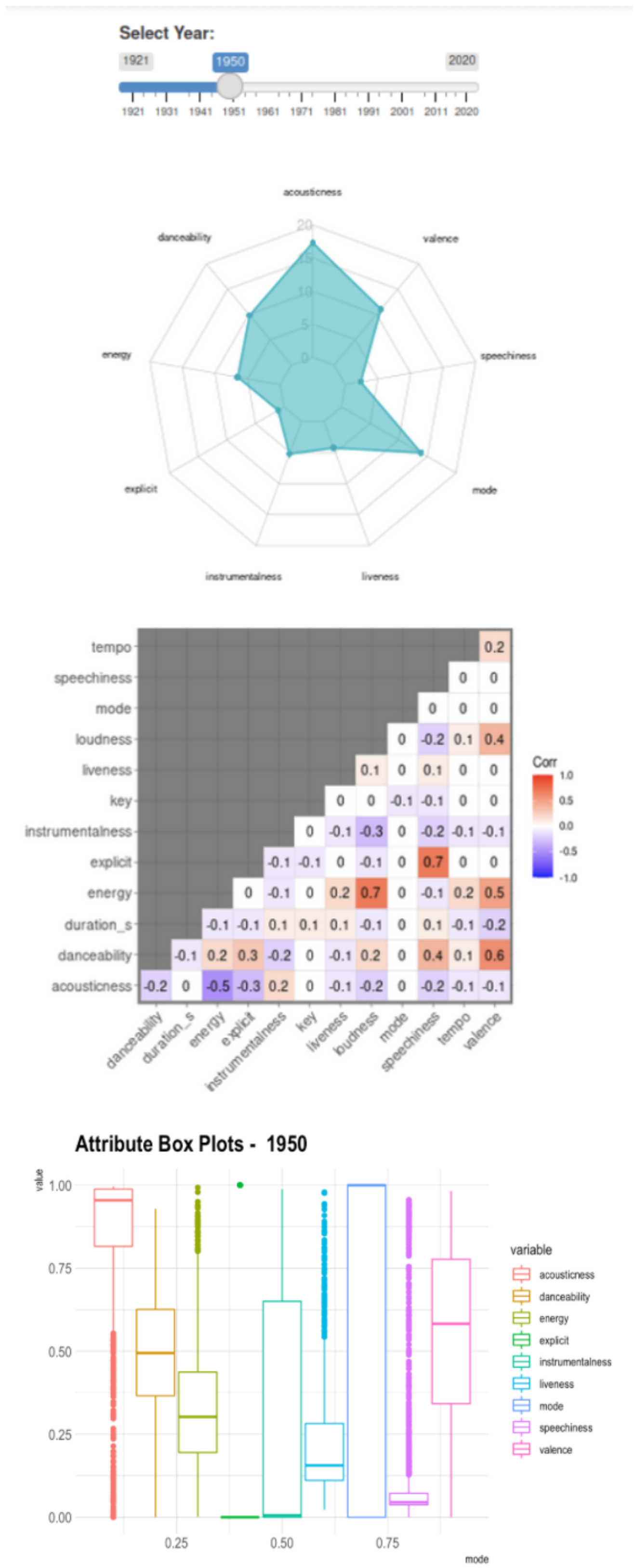

Figure 8: Multiple Views

Running the Tool

To run the tool locally, the following steps need to be followed:

1. Set the working directory to the location of the packaged tool with *setwd("path_to_tool")*
2. Install all required packages, which are listed with imported at the top of the app.R script. The full list of dependencies can also be found below in table 1.
3. Launch the app with runApp() or RStudio's keyboard shortcuts while being in the app.R script.

Alternatively, the dashboard also can be viewed through the published version on shinyapps with the link https://1shinyvismadrid.shinyapps.io/music_explorer/.

Dependencies List
car
corrplot
dplyr
DT
fmsb
GGally
ggcorrplot
ggplot2
hrbrthemes
janitor
plotly
purrrlyr
reshape2
rlang
shiny
shinyBS
shinythemes
stringr
tidyverse

Table 1: Dependencies List