Data Visualization Assignment

Analysis of the connection between music and globalization

Universidad Politécnica de Madrid

Cecilia Peccolo, Karla Gonzalez Romero, Federico Paschetta

cecilia.peccolo@alumnos.upm.es
karla.gonzalez@alumnos.upm.es
federico.paschetta@alumnos.upm.es

Contents

1 Problem characterization

Globalization is profoundly affecting various aspects of our society, blending new cultures and bringing significant changes to our social fabric. This impact is particularly evident in the field of music. For example, Europe has seen a surge in the popularity of reggaeton, a genre with Caribbean roots, while African beats and American hip-hop have captivated audiences well beyond their origins, echoing across continents.
Our aim was to explore how globalization has facilitated the spread of these diverse music genres across the globe. We were interested in observing how artists have begun to collaborate, merging different musical genres, languages, and styles.
This collaboration has led to the emergence of a new, flexible international musical frontier. This trend reflects the interconnected nature of our world and underscores the creative possibilities that arise when varied cultural elements intermingle. It's a testament to the power of music as a universal language, capable of transcending geographical, cultural, and linguistic barriers, creating a rich and diverse global soundscape.

2 Data abstraction

For this work, no dataset on the web was big enough or suitable, so it was necessary to fetch data based on our own will. In order to get the most accurate data possible, Spotify API has been selected as main source, fetched via Spotipy interface, integrating what was missing with musicbrainz API catalogue. At first table *artists* has been created, getting data from all artists having at least one song (or having a featuring) in Spotify top 50 playlist, at the end of October, in one of the following countries: USA, UK, Italy, France, Argentina, Mexico, Spain.

After this step, we will have a table with the following schema.

ID	Name	Popularity	Country	Genre
06HL4z0CvFAxyc27GXpf02	Taylor Swift	100	US	pop

Where *ID* corresponds to the Spotify ID of the artist, *Name* is the name he has on the platform, *Popularity* is a score computed directly by Spotify, *Country* is the "musically raising" place for the artist, data got using Musicbrainx API, and *Genre* is the artist main genre. This dataset contains 1229 artists, considered as the most influential in today's music.

Then, for each of the artist in the *artists* table, with a loop, all his albums are fetched, to get an other dataset: *albums*. *albums* contains the following features for each row: *ID*, the unique ID for the album, *Name*, corresponding to album's name *Date*, album release date and *Main artist*, album main artist (if it's a joint album we will get only one of them, but it will not represent a problem). The dataset contains 4782 rows, each corresponding to an album or

a single.

ID	Name	Date	Main Artist
4FftCsAcXXD1nFO9RFUNFO	Un Verano Sin Ti	2022-05-06	Bad Bunny

To get a final complete dataset, for each album in *albums* dataset, all songs in it are fetched, resulting in the *songs* dataset. This dataset contains as features *ID*, *Name*, *Main Artist*, *Album ID*, *Album Name* and *Popularity*, corresponding respectively to Spotify song ID, song name, main artist name (present in *artists* dataset), album ID and name (present in *albums* dataset) and song popularity, score given by Spotify to each song, based on the streams it has. Dataset contains 27272 rows.

ID	Name	Main Artist	Album ID	Album Name	Popularity
Un v58OOXk2SoU711L2IXO	Moscow Mule	Bad Bunny	4tfs2nfvSSFjvv	Un Verano Sin Ti	85

At the end, as one idiom requires all featurings between different artists, for each song featurings are fetched. It is created a dataset with the following attributes: *SongID*, ID of the song with featuring, *Song Name*, name of the song with featuring, *Artist1ID*, ID of the first artist in the song, *Artist1Name*, name of the first artist, *Artist2ID*, ID of the second artist in the song, *Artist2Name*, name of the second artist. The table contains 7497 rows, some songs will not appear in this table, while others will appear more than once, as all couples of featuring artists are needed.

SongID	Song Name	Artist1ID	Artist1Name	Artist2ID	Artist2Name
3gE4eQH3K83Sght0ZLvuBK	MIA	4q3ewBCX7sLwd24euuV69X	Bad Bunny	3TVXtAsR1Inumwj472S9r4	Drake

Due to the necessity of having only main genres (like pop, latin, hip-hop), instead of specific ones (like uk pop and contemporary pop for pop), in the *artist* dataset it's been added one other column, named *Main Genre*, representing the genre family.

3 Interaction and visual encoding

To analyze our problem, we divided our analysis into three parts, each with a different representation. Once you open the *ShinyApp*, it will appear a menu with the possibility of selecting one of them for analysing three different aspects:

- **Analysis of International Collaborations (Featuring) Over Time:** The first part of our analysis examines the number of songs created through international collaborations (featuring) by artists of different nationalities, up to the year 2023. We counted the number of collaborative songs among artists from three geographical divisions: the United States, Latin America and the Caribbean, and Europe. Our representation includes a *dropdown menu* that allows the selection of specific origins to compare. When "All" is selected, it shows the general trend of international collaborations among all nationalities. The data clearly indicates that from

the early 2000s to 2023, the number of collaborations has grown exponentially, with a significant spike after 2020, as shown in the Figure 1. This representation was chosen for its clarity in showing the frequency of collaborations over time. The geographical division of artists was selected to highlight the most striking musical differences, thus making the representation clear and simple.
As the following images represent, the number of featurings between these continents are increasing exponentially in the last years, which confirm the thesis that the globalization is having a big impact in the music landscape. In the pictures 2, 3, 4, it is possible to observe this relationships.
The line chart is the ideal graphic to represent the trend of a phenomenon over a fixed period of time. We chose to connect the points to make different scores more comparable, as the slope of the line can help understand whether one point is higher or lower than another. This approach also provides a more immediate understanding of the overall trend.

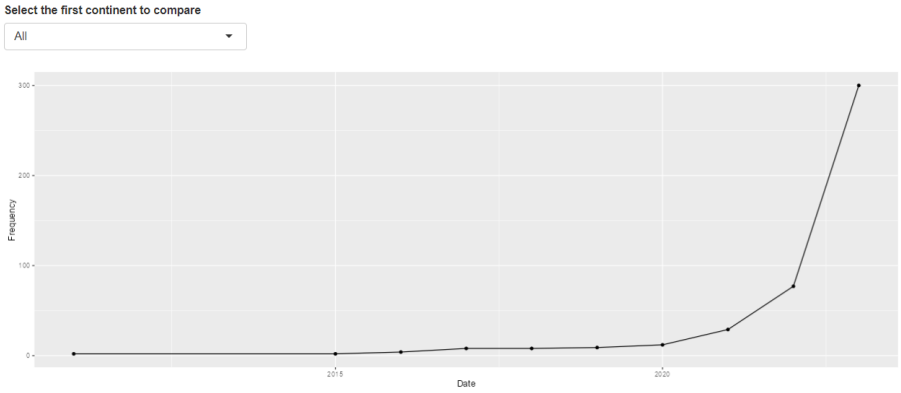

Figure 1: Line plot of the number of featurings over the years

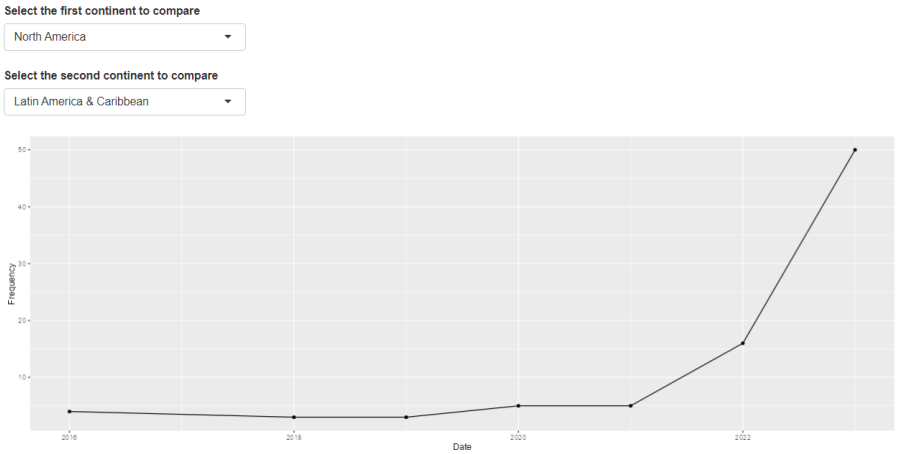

Figure 2: Line plot of the number of featurings between North America and Latin America

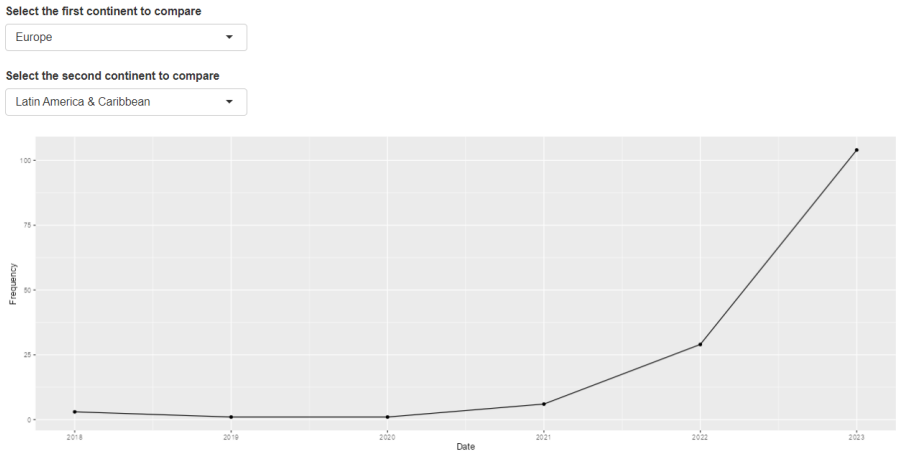

Figure 3: Line plot of the number of featurings between Europe and Latin America

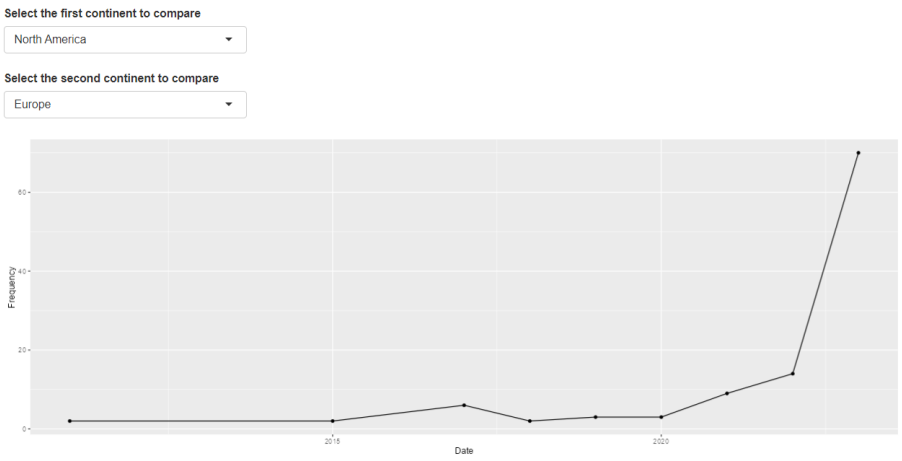

Figure 4: Line plot of the number of featurings between Europe and North America

- **Comparative Network Analysis:** The second representation involves the comparison of two networks. Each network is composed of multiple nodes representing each artist in the dataset. Two artists are connected if they have collaborated on a song: we opted for a network representation to illustrate the relationships between continents due to its intuitiveness. It quickly and clearly shows the density of relationships between nodes. The origins of the singers are distinguished by contrasting colors, enhancing the clarity of data visualization. Additionally, we chose to compare only two networks at a time because a network, in itself, can be a complex representation with numerous elements. Comparing more networks simultaneously could lead to confusion and diminish the immediacy and clarity we have achieved. Furthermore, since we are selecting very broad periods for analysis, adding more comparisons could become redundant.
  The nodes are color-coded based on the artists' geographical origins, using the same divisions as before. For visualization, we used purple for artists from the United States (a combination of red and blue, the colors of the American flag), orange for Latin countries (a mix of yellow and red, colors of the Spanish flag, often associated with these countries due to language), and green for European artists (a combination of blue and yellow, the colors of the European flag). These color choices, linked to the respective flags, create a clear distinction between the three origins.
  The networks are interactive, allowing for zooming and moving around the graph, and users can select a node to see the name of the artist it represents, only one a the time for clarity reason: in fact, visualising the nodes with all the names from the beginning it would have ruined the purpose of this representation which is to represent the relationship between

continents and not to the single artist.
An adjustable time slider is provided to view the relationships between continents over different periods.
The figure 5 shows an example of how the display of two networks look like, in this case a comparison was made between connections from 2017 to 2020 and connections from 2020 to 2023: although the time periods being analyzed are relatively close to each other, a striking difference can be observed in the two graphs. In the first graph, there's a noticeable pattern of connections predominantly between the United States and Latin America, indicating a strong interaction or influence between these regions during that period. However, in the second graph, a different trend emerges. It becomes evident how American culture, which initially had a significant presence in Latin America, has expanded its influence remarkably into Europe. This shift highlights the dynamic and evolving nature of cultural exchanges and influences across different continents, demonstrating how cultural trends can rapidly change and permeate different regions over time.

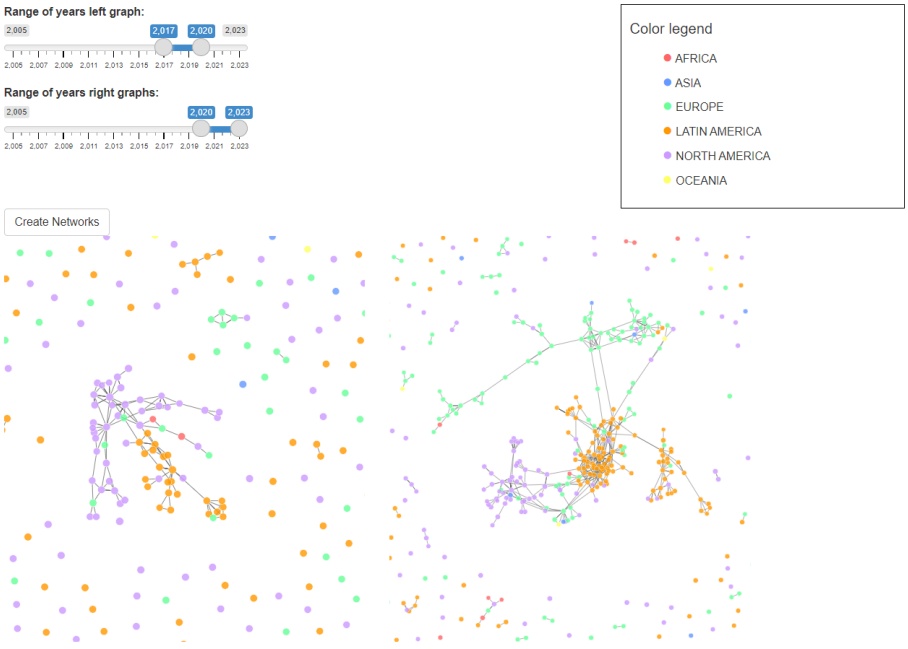

Figure 5: Comparison of the networks build between the singers in the time range 2017-2020 and 2020-2023

- **Bar Graph of Song Production by Genre:** The third chart provides a detailed look at the number of songs released in various genres over a

specific time frame, covering genres such as country, cuarteto, EDM, hip-hop, Latin music, pop, rock, and others. This data is presented using a bar plot, where the x-axis represents the different music genres, and the y-axis shows the relative frequency of songs published in each genre during a chosen time period.

Given that the variable of interest is qualitative and the goal was to visualize frequencies, the bar plot emerged as the most obvious choice. The genres in the initial dataset were personally categorized into major ones to simplify comparison. Absolute frequency was chosen to make different time periods comparable and provide a clearer idea of proportions. The 'Others' category is positioned last, as it is of less interest since users cannot know which genres have been classified under this category. Meanwhile, other categories are presented in alphabetical order rather than frequency order to facilitate a more organized interactive experience.

A key feature of this chart is its interactivity, allowing users to select a time range from 1977 to 2023. This interactive element enables users to explore changes in music trends over nearly five decades, offering insights into the evolution and shifts in musical preferences and influences over the years.

As we can see in the Figure 6, Latin songs produced in the last 8 years account for about 27% of the total songs, which is a significantly high number considering the size of the countries of origin compared to the rest of the world. However, if we look at Figure 7, it shows that Latin genres did not even make up 15% of the music from 1990 to 2010.

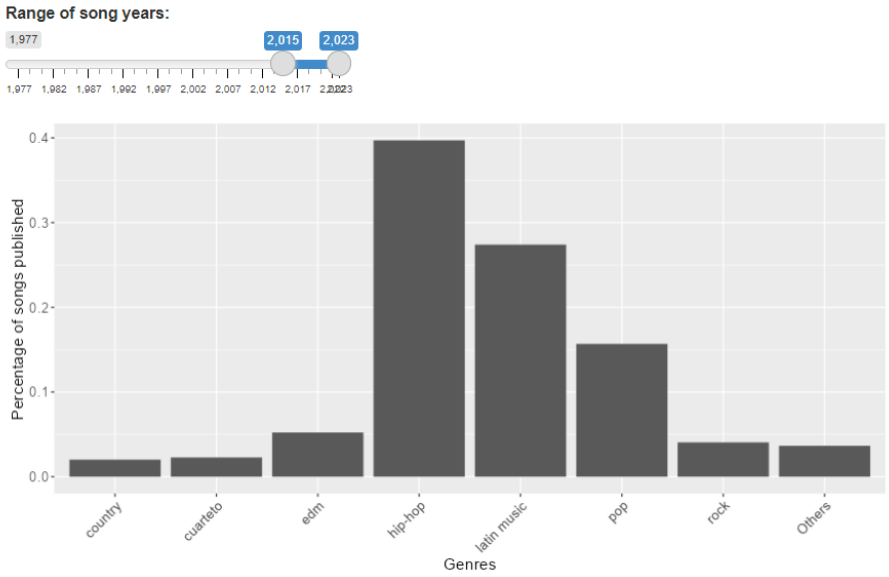

Figure 6: Proportion of songs genre published between 2015 and 2023

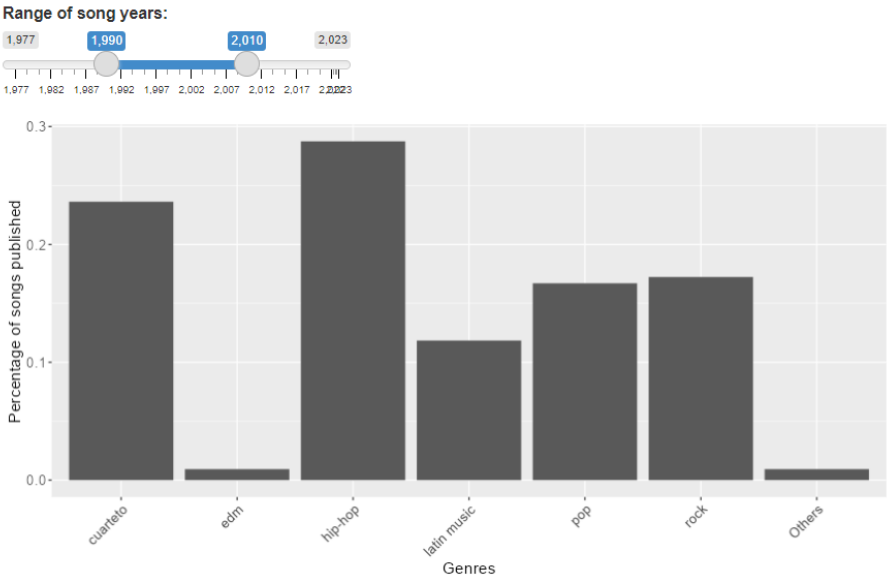

Figure 7: Proportion of songs genre published between 1990 and 2010

4 Algorithmic implementation

4.1 Work Environment and Libraries

Project has been developed using *RStudio IDE* and, to create the design application, *Shiny* package. Other libraries used in the visualization tool are *ShinyJS*, to handle conditional visualization in the first view, *dplyr*, to manipulate easily dataframes, *networkD3*, to visualize a network, *data.table*, to create easily a new dataframe, *countrycode*, to get continents from artist countries, *lubridate*, to handle easily Date data type, *ggplot2*, to create charts and *forcats*, in order to manipulate data with factor variables.

4.2 Data manipulation

Before building visualizations, we started creating data to be visualized. At first with *fread* function starting csv files have been read, then for each entry in artists table it's added an id and it's associated the continent related to the artist with *countrycode* function, with parameter destination set to *continent*. This function sets continent to 'Americas', while we need to divide artists belonging to that zone, so, again, with *countrycode* function with destination set to *region*, we can divide American artists in North America and Latin America & Caribbean. N Then we select only most popular songs, in order to keep only most relevant data and not to overload visualization. With *inner_join* and *merge* functions we get final dataframe from the combination of four initial datasets, which has all the information of each artists and all the information of the song which has featuring from artists. For last idiom, which needs an aggregation of data, at first irrelevant variables are left out, then eight most popular genres are selected in order to be displayed.

4.3 Visualization choice page

Initial page the user see is a simple dropdown choice, asking him to select the idiom he wants to visualize. This has been done with the help of *observeEvent* function from shiny library: following the Observer-Observable pattern, the app waits for the user to select one choice and when it's selected, it fires an event in the main server function, displaying one of the three idioms with the right *appXServer* function, showing the X idiom selected.

4.4 Line Chart page

The goal is to visualize and comprehend the frequency and trend of musical collaborations throughout time, as well as how these may differ across regions or globally. The trend of these collaborations over time is represented using a line chart. The script combines multiple datasets (artists, songs, albums, and features) to produce a comprehensive perspective of musical collaborations. Primarily, the artist's country or continent information is necessary to facilitate geographical comparisons in the visualizations. The 'ID' column from the

artists dataset is essential for identifying and linking artists to their respective songs and albums. From the songs dataset, the 'Popularity' column is used to filter songs based on a set threshold, ensuring only significant or popular songs are included in the analysis. The 'Date' or 'Year' from songs or albums is crucial for the temporal aspect of the visualization, allowing trends to be observed over time. The featurings dataset provides the connections between artists through 'Artist1ID' and 'Artist2ID', indicating the collaborations that are central to this idiom. These IDs are matched with the artists' dataset to provide a comprehensive view of the collaborations.

Users can compare one or two continents. For the line chart, the selected UI elements and plot are displayed using a plotOutput. The server logic filters the data to include just the relevant collaborations based on the user's continent selections.The plotlineChart function then generates a ggplot line chart depicting the evolution of collaborations over time. The x-axis is time (in years), and the y-axis indicates the frequency of cooperation. The data is group by year and counts the number of collaborations for each year.

4.5 Network page

The network is the most complicated idiom of the three and requires an external library, called *NetworkD3*. The *forceNetwork* function, which is necessary to create the network graph and requires nodes and edges data, is included in this package. Attributes such as name, artistId, group (which typically represents the continent), and popularity provide further details about the nodes. The source and destination characteristics of the edges, which show the relationships between artists, must match the artistId of the connected nodes. The script first gathers the information required from the artists and featured datasets to generate the nodes and edges data frames in order to build this dynamic visual. Nodes give a visual representation of the artist's origin and importance by being color-coded according to their continent and enlarged dependent on their level of popularity. Edges are generated from the featured dataset, which identifies artist collaborations, and are filtered to include just those within the user-specified year range given by two sliders. This strengthens the interactive aspect of the display by enabling a comparison view across several time intervals. After preparing the data, the forceNetwork function calculates the positions and paths for nodes and links, organizing them into a force-directed layout. In order to reduce overlaps and clearly define the network's structure, this layout disperses the nodes to resemble natural forces. Artists that collaborate extensively and are prolific are highlighted by nodes with many connections, while communities of artists who often work together are shown by clusters of nodes, which may imply future collaborations within these groups. A key component of the visualization's performance is the reactive programming approach of the Shiny app. The software reacts by redrawing the network graph to provide the updated data when users change the time range or other settings. This guarantees that the visualization stays up to date and flexible, enabling a thorough investigation of the changing terrain of musical collaborations and the complex

relationships that develop between artists over time.

4.6 Bar Chart page

Particular columns from the songs dataset, such as "Date" or "Year" and "Genre," are crucial for the third idiom, which compares music genres over time using a bar chart display. The 'Date' or 'Year' column provides for temporal filtering, while the 'Genre' column categorizes the music; both are critical for the study. Then data is grouped by eight main genres in the visualization in alphabetical order, in order to keep it consistent. The modality "Others" has been put in the last place, because, since the user will not be able of knowing which are the genres classified as such, it will not be the main focus of the representation. The 'Popularity' and 'ID' columns may help to narrow the dataset further, ensuring that the emphasis is on significant songs and proper aggregation. Genres are shown on the x-axis of the bar chart, and a column is computed for the y-axis that shows the count or proportion of songs in each genre. This idiom's algorithmic implementation begins with data preparation, in which songs are filtered by the given date range and by popularity. After classifying the data by genre, the number or percentage of songs in each genre is determined. The bar chart representation is based on this processed data. An essential part is the *plot_barChart* function, which uses user input and the processed data to produce a *ggplot* bar chart. The y-axis shows the relative frequencies or counts of the genres, while the x-axis lists the genres. This idiom's user interface, which was made using the 'app3UI' function, has a plot output area for the bar chart and a range slider for choosing the year. The server functionality is contained in the 'app3Server' function, which processes and filters the data in response to user inputs before displaying the updated bar chart. This idiom, like the others, makes use of Shiny's reactive programming approach, which updates the visualization in response to user input.

5 Instructions to run the application

5.1 Visualize Using Schinyapps.io

The Shiny app has also been published in shinyapps.io. The following link lets you visualize in the browser the visualization tools.

`https://datavisualization-karla.shinyapps.io/data_visualization_project/`

Once you open the ShinyApp, a menu will appear with the choose of selecting one of them for analysing three different aspects.

5.2 Run Locally

- Install R and RStudio: Install the recent versions of R and RStudio on your PC. RStudio is a R integrated development environment (IDE) with a user-friendly interface for running R scripts.
- Once installed open RStudio or any R environment and install the necessary libraries. This libraries can by installed by running the following command in r:
  - install.packages("networkD3")
  - install.packages("dplyr")
  - install.packages("data.table")
  - install.packages("countrycode")
  - install.packages("shiny")
  - install.packages("lubridate")
  - install.packages("ggplot2")
  - install.packages("shinyjs")
  - install.packages("forcats")
- Prepare the Data: All files ['artist.csv', 'albums.csv', 'songs.csv', 'featurings.csv'] most be prepared nad located at the specified paths in the script. If the files are stored in a different location that the script then the file paths might need to be modify.
- Run the App:
  - From the IDE: Open the app file (.R or .Rmd) in the choosen IDE, Run the script by pressing "Run APP" button
  - From the console: use command 'shiny::runApp()'
- Interact with Application: A new window should appear if the app was able to run correctly. The user can interact with the application by selecting the different options, adjusting filters and sliders. The app will update based on the interactions
- Troubleshooting: If the application doesn't run or an error is thorwn back, ensure all pacages are installed and all file paths are correct. Ensure internet connection is stable for intalling packages