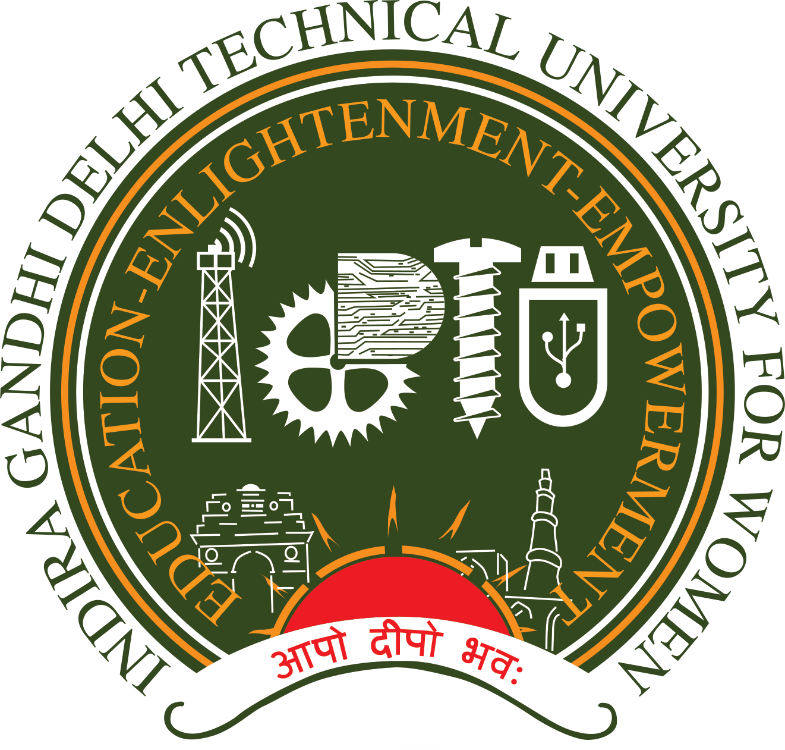
**Indira Gandhi Delhi Technical University for Women**

**(Established by Govt. of Delhi vide Act 09 of 2012)**

**Kashmere Gate, Delhi - 110006**



**REPORT FILE**

**For**

**IT WORKSHOP (BAI-108)**

**Department of Information Technology**

**MUSIC RECOMMENDATION SYSTEM IN R PROJECT**

SUBMITTED BY: SUBMITTED TO:

1) PALAK MANGLA (043) Mr. SANTANOO PATTNAIK

2) PARUL DHARIWAL (044) IT WORKSHOP LAB

3) PALAK JETHWANI (042)

B.Tech / AI & ML

Sem: 2

**INDEX**

**1) ACKNOWLEDGEMENT**

**2) OBJECTIVE**

**3) DATA ACQUISITION**

**4) DATA EXPLORATION**

**5) VISUALISATION OF DATA**

**6) CORRELATION ANALYSIS**

**7) RECOMMENDATION MODEL**

**8) MUSIC RECOMMENDATION SYSTEM R PROJET CODE**

**9) CONCLUSION**

**10) SCOPE OF IMPROVEMENT**

**ACKNOWLEDGEMENT**

We want to convey our appreciation to Mr. Santanoo sir for his guidance and support in assisting us throughout the project titled "*MUSIC RECOMMENDATION SYSTEM IN R PROJECT*" enabling us to successfully complete it. Additionally, we would like to express our gratitude to our fellow batchmates for their assistance and continuous support during the project's duration. Lastly, we are thankful to Indira Gandhi Delhi Technical University for providing us with the opportunity to work on this project.

**MUSIC RECOMMENDATION SYSTEM IN R PROJECT**

**OBJECTIVE**

Many new music platforms are emerging as a result of the rapid growth of online and mobile platforms. These platforms provide songs lists from all across the world. Every person has a unique preference for music. The majority of individuals use online music streaming services such as Spotify, Apple Music, Google Play, or Pandora.

However, it can be challenging to choose between millions of tracks. Therefore, music service providers require an efficient approach to organize songs and assist their customers in choosing music by providing quality recommendations. As a result, an effective recommendation system is essential.

Many music streaming services, such as Gaana and Spotify, are currently focused on developing high-precision commercial music recommendation systems. These businesses make money by assisting their consumers in choosing suitable music and charging them for the quality of their suggestion service. As a result, there is a strong market for good music recommendation system.

The purpose of this data analysis project is to explore the music dataset and perform various analyses to gain insights into the data. The dataset contains information about different songs, including their titles, artists, genres, and various attributes such as BPM, danceability, duration, and popularity.

The objective of this project is to develop a music recommendation system that provides personalized song recommendations to users based on their preferences and song features. The system aims to enhance the user experience by suggesting songs that align with their musical tastes.

**METHODOLOGY**

**DATA ACQUISITION**

Dataset taken from the site:

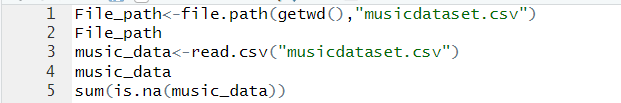
<https://www.kaggle.com/datasets/iamsumat/spotify-top-2000s-mega-dataset>

Content

A LITTLE ABOUT THE DATASET:

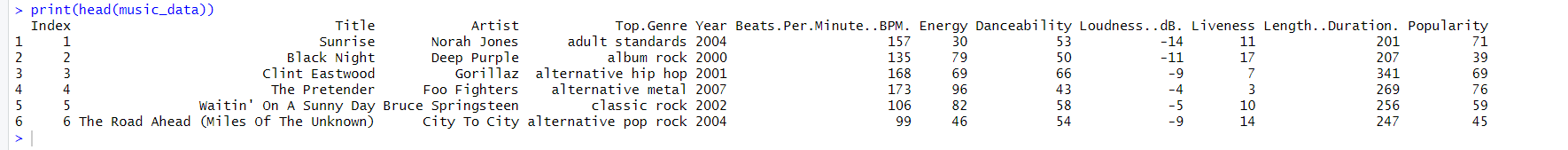
* **Index**: ID
* **Title**: Name of the Track
* **Artist**: Name of the Artist
* **Top Genre**: Genre of the track
* **Year**: Release Year of the track
* **Beats per Minute (BPM)**: The tempo of the song
* **Energy**: The energy of a song - the higher the value, the more energetic. song
* **Danceability**: The higher the value, the easier it is to dance to this song.
* **Loudness**: The higher the value, the louder the song.
* **Length**: The duration of the song.
* **Liveness**: The higher the value the more spoken words the song contains
* **Popularity**: The higher the value the more popular the song is.

The dataset was loaded into R using the **read.csv** function, to ensure data quality missing values in the dataset were examined using the sum(is.na(music\_data)) command. The preprocessing phase involved several steps, including data cleaning and feature engineering, to prepare the dataset for the recommendation system.



**DATA EXPLORATION**

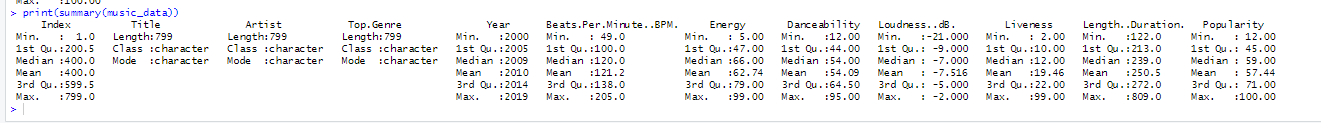
* The **head** function was used to display the first few rows of the dataset, giving an overview of the data.



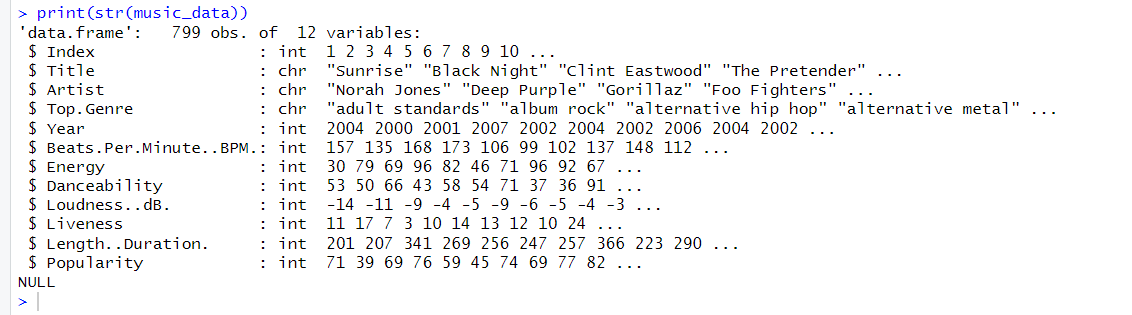
* The **tail** function was used to display the last few rows of the dataset.



* The **summary** function provided summary statistics such as minimum, maximum, mean, and quartiles for each column.

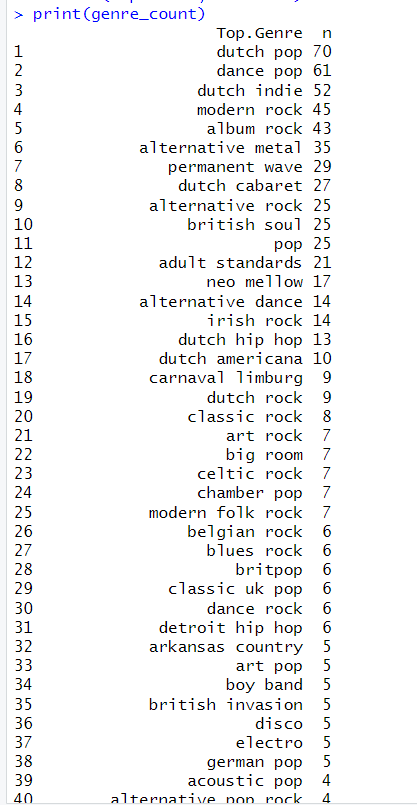


* The **str** function provided the structure of the dataset, including the data types of each column.



Looking for unique genres present in the dataset:

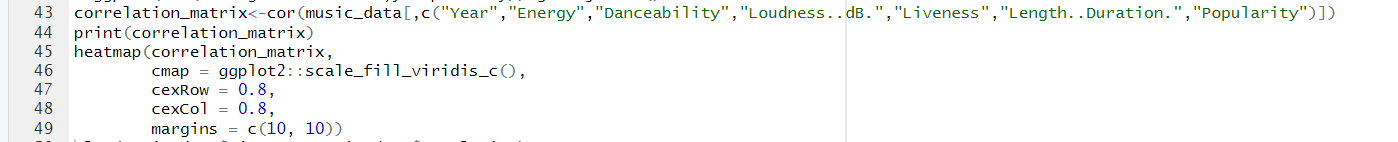


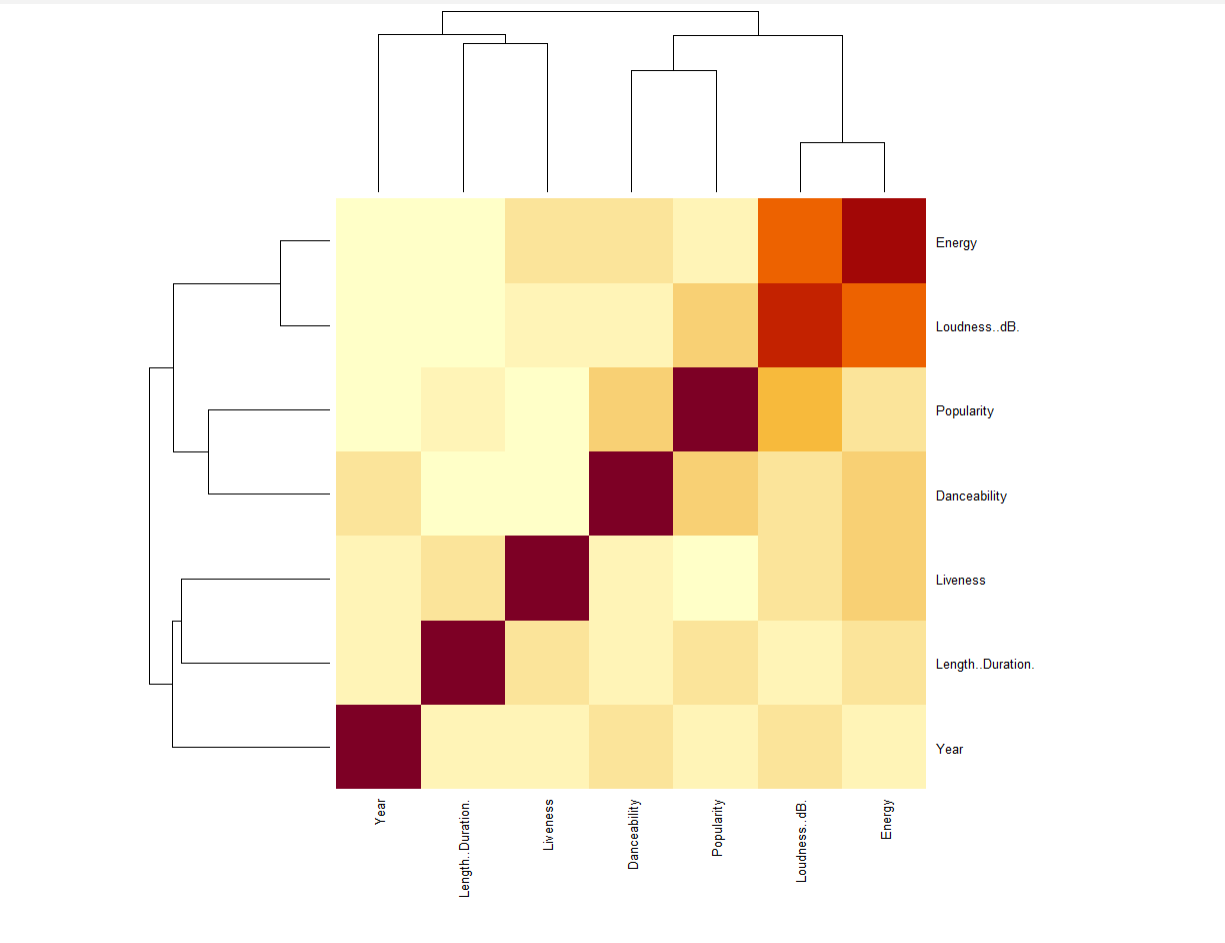


**VISUALIZATION OF THE DATA**

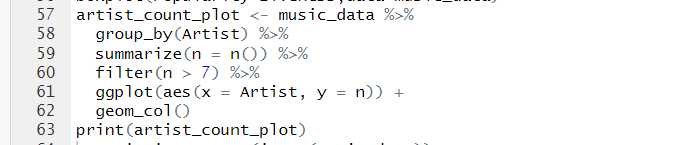
1) Correlation Matrix: The correlation matrix quantifies the relationships between different attributes of songs. The matrix provides correlation coefficients between variables such as year, energy, danceability, loudness, liveness, length/duration, and popularity. It helps identify the strength and direction of these relationships.

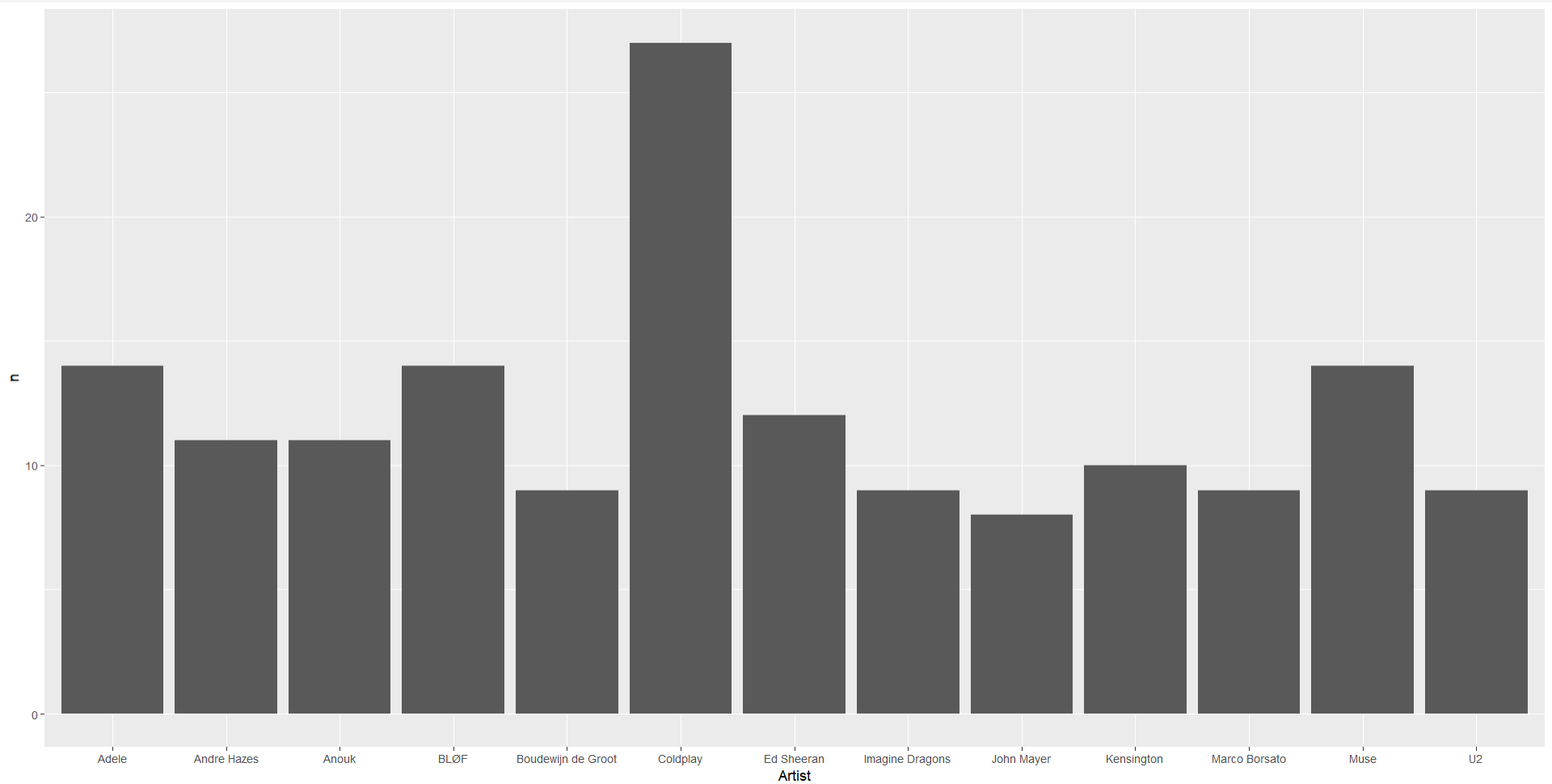
Heatmap of Correlation Matrix: The heatmap visually represents the correlation matrix using colors. It allows for a quick assessment of the strength and patterns of correlations between variables.



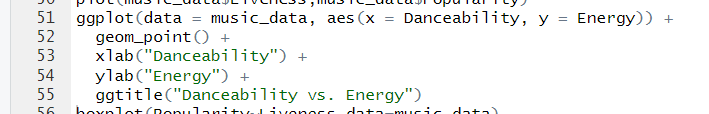


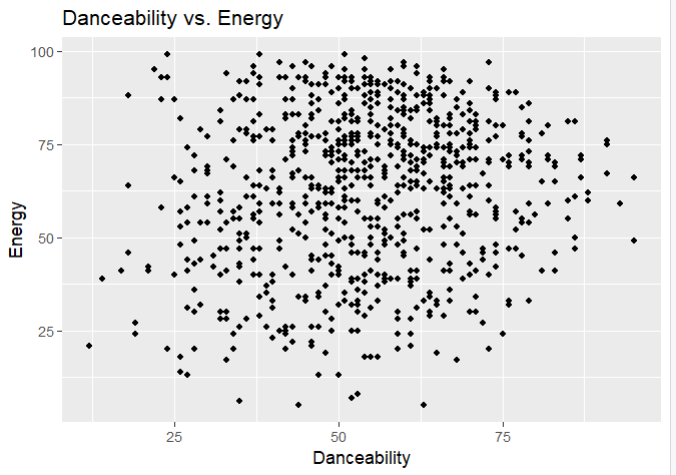
2) Artist Count Plot: The plot represents the count of songs for each artist. Artists with more than 6 songs are included. It gives an insight into the number of songs contributed by different artists.



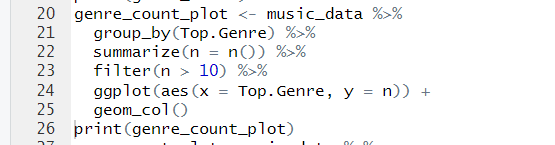


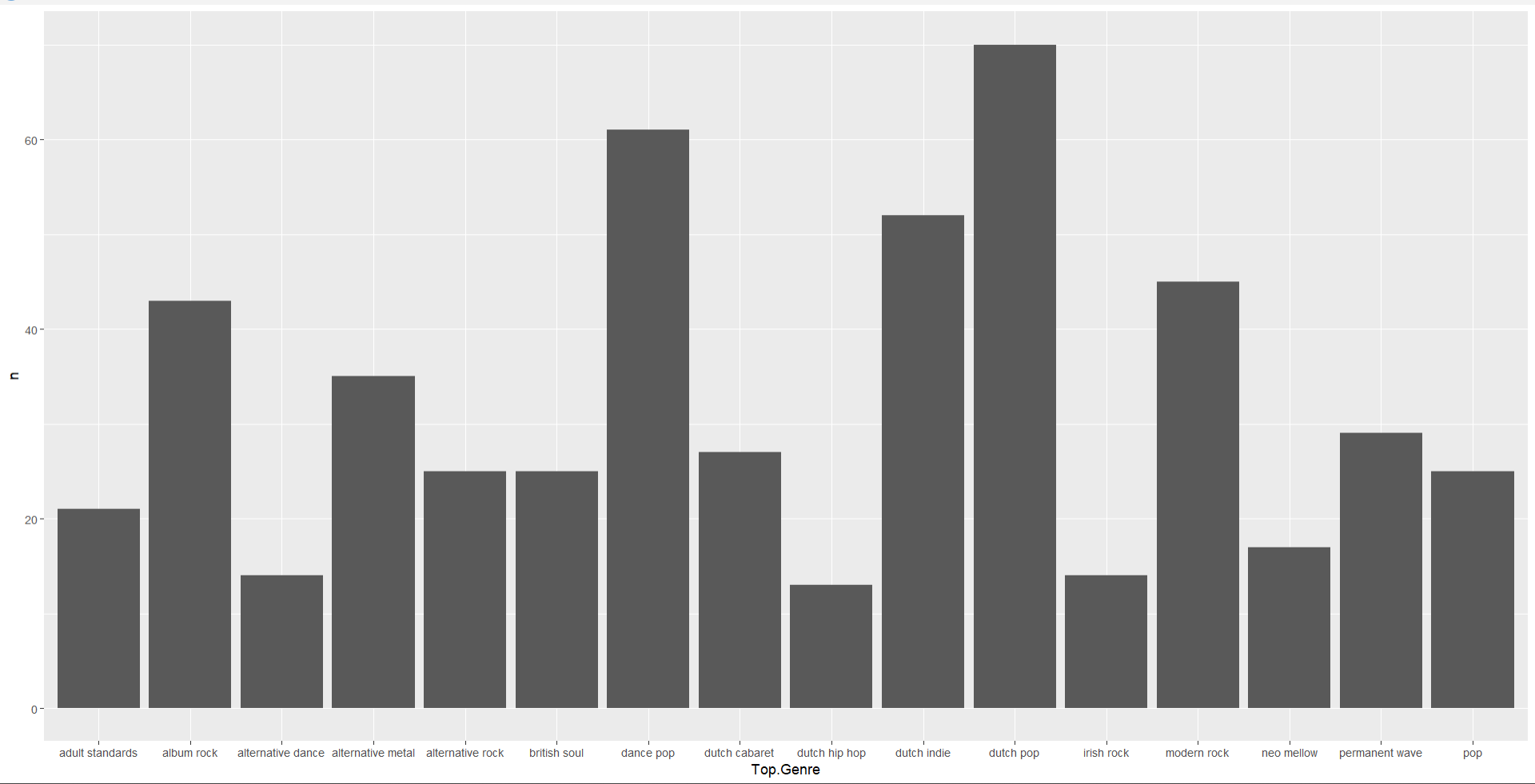
3) Danceability vs. Energy: This scatter plot displays the relationship between danceability and energy. It helps visualize if there is any correlation between these attributes and whether songs with higher danceability tend to have higher energy levels.



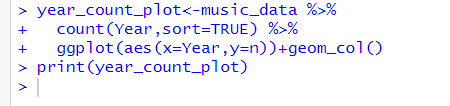


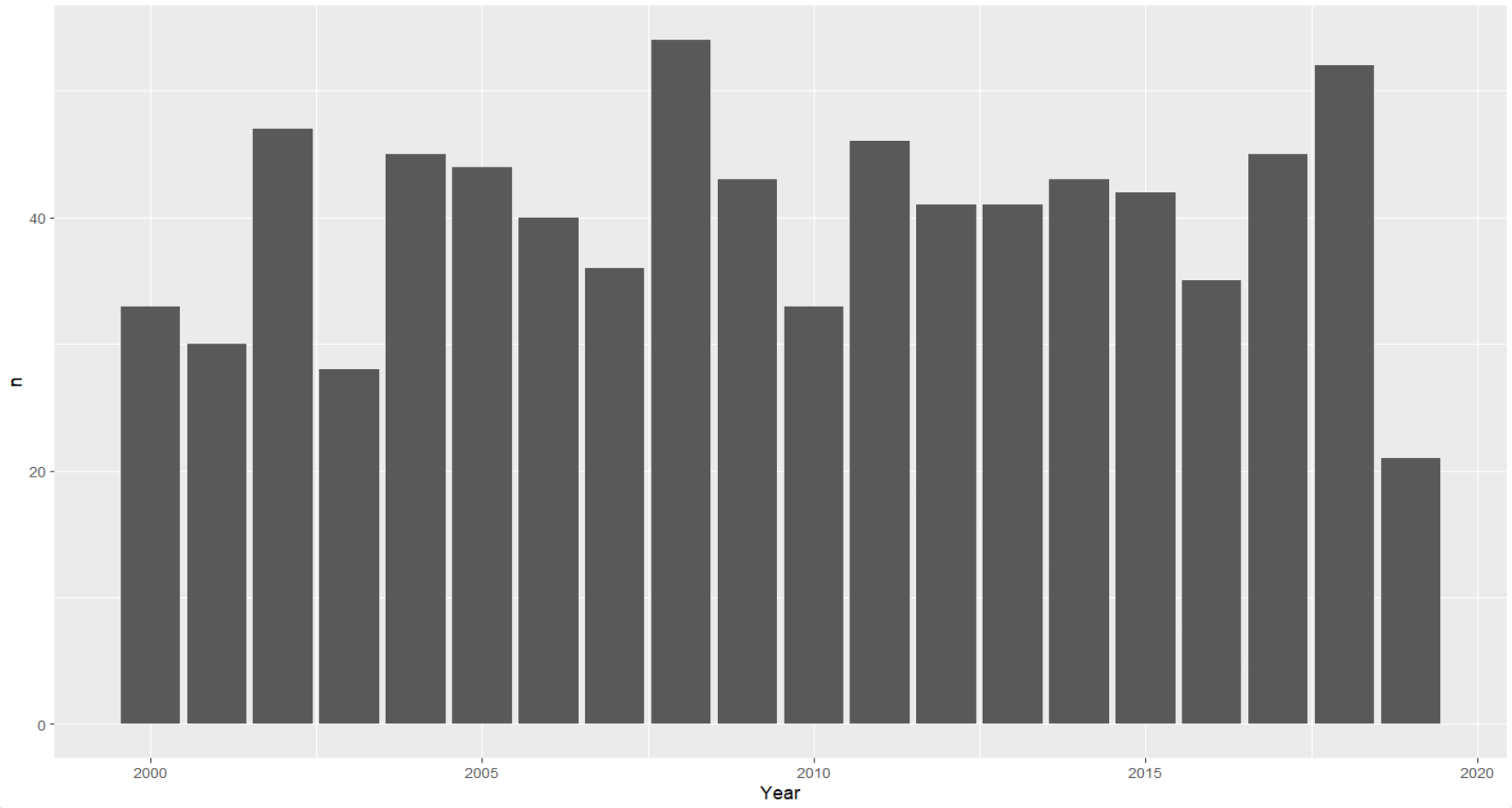
4) Genre Count Plot: The plot shows the count of songs for each top genre. Genres with more than 10 songs are included. It provides an overview of the distribution of songs across different genres.



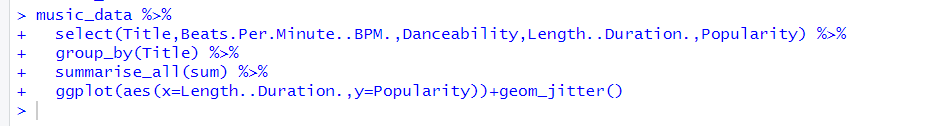


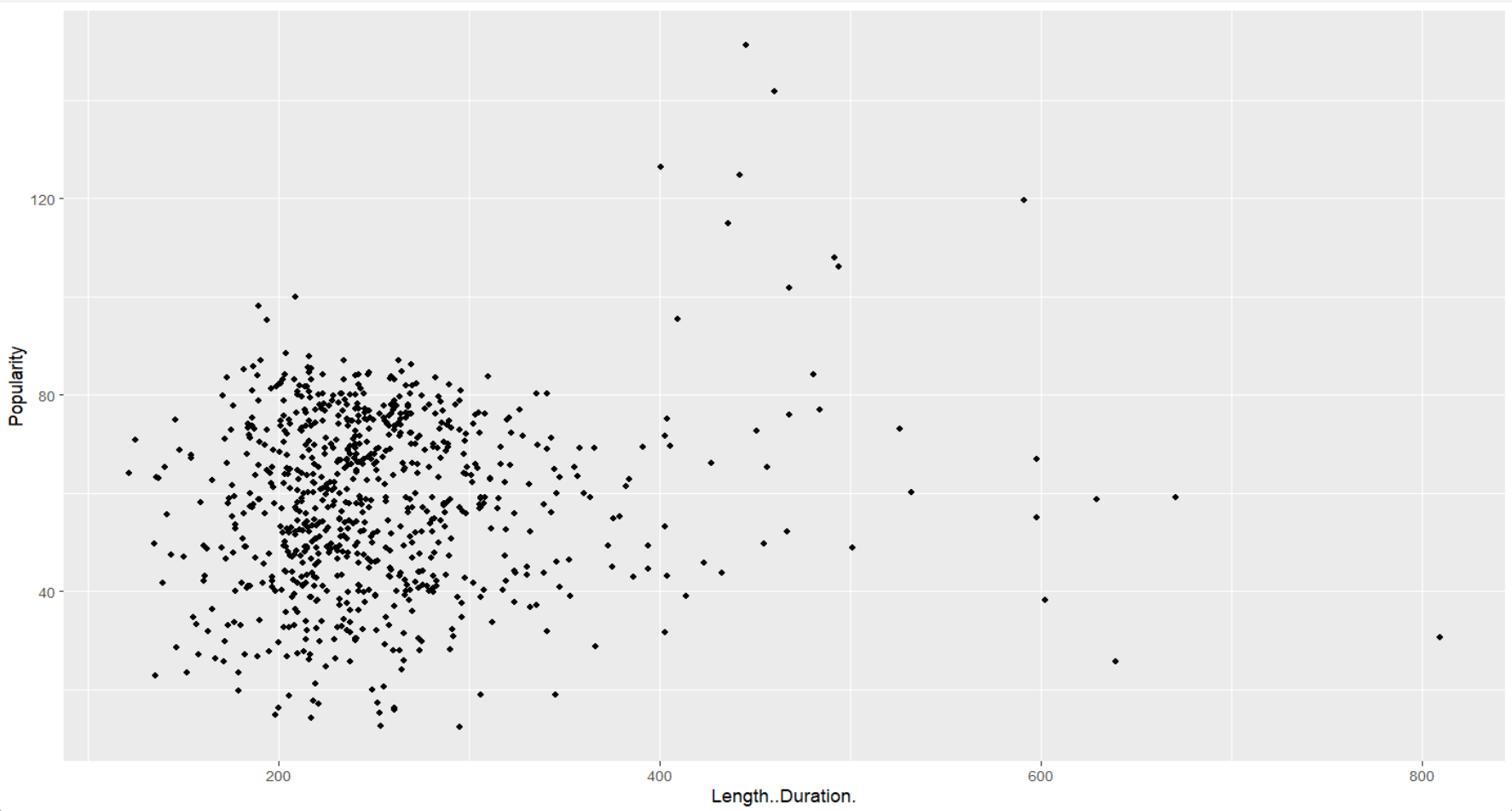
5) Year Count Plot: This plot displays the count of songs for each year. It helps visualize the distribution of songs over time, indicating which years have a higher or lower number of releases.





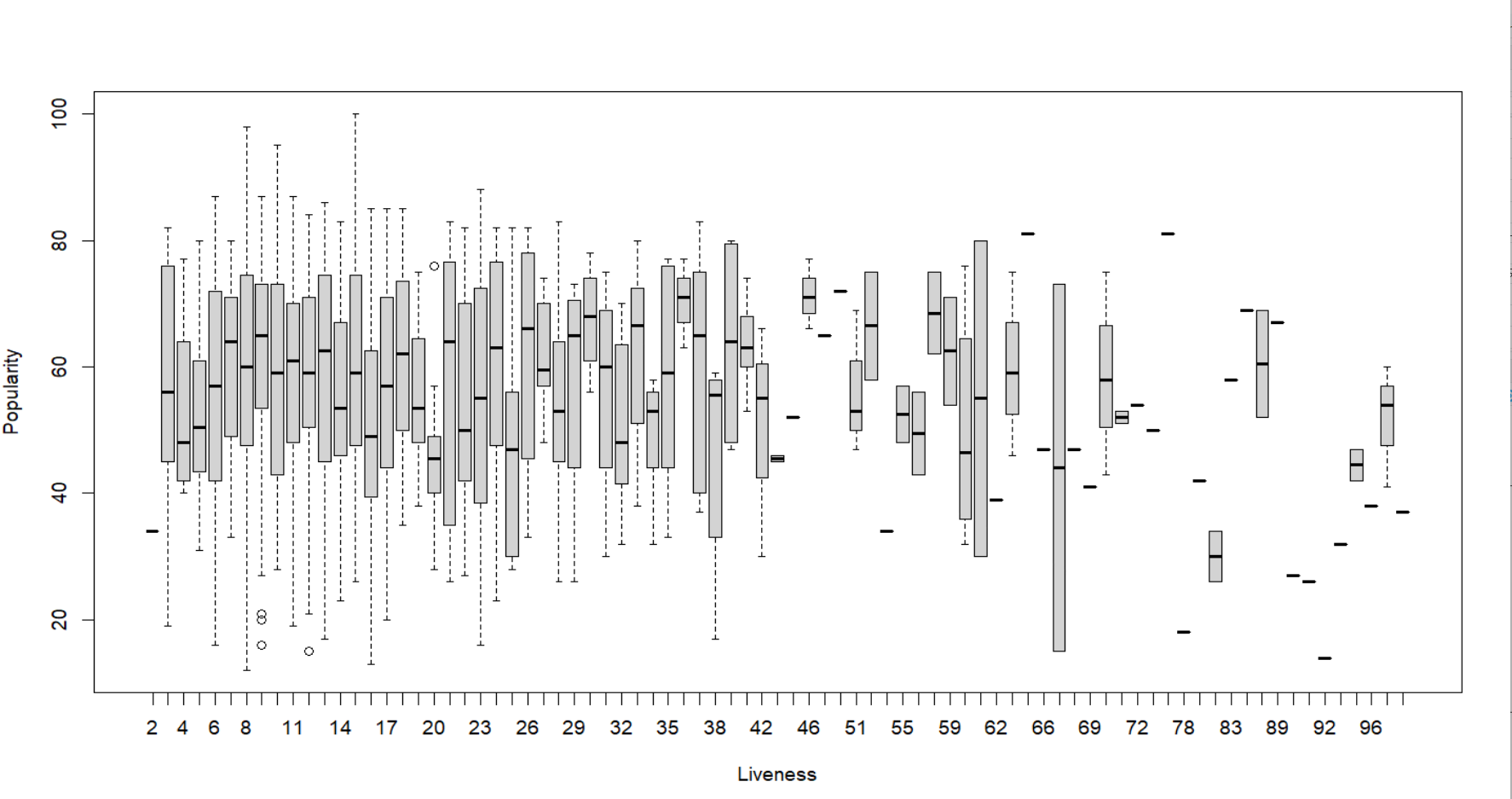
6) Length vs. Popularity: The jitter plot shows the relationship between song length and popularity. It gives a scattered representation of how the duration of a song relates to its popularity.





7) Popularity vs. Liveness (Boxplot): The boxplot shows the distribution of popularity across different levels of liveness. It helps compare the median, quartiles, and potential outliers in popularity for different liveness categories.

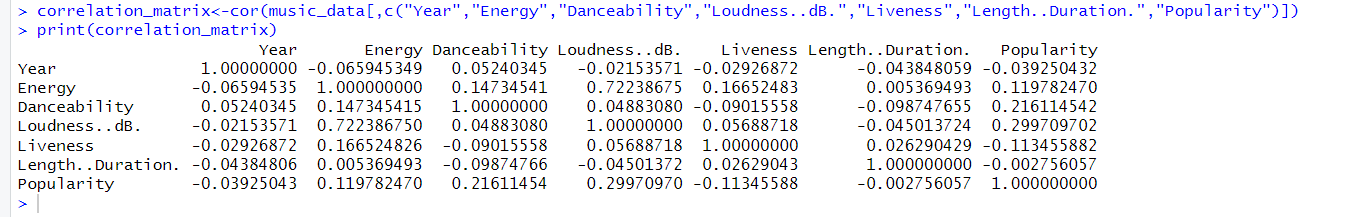




Overall, these graphs provide insights into the distribution, relationships, and patterns within the music dataset. They assist in understanding variables, identifying trends, and exploring potential correlations between attributes.

**CORRELATION ANALYSIS**

A correlation matrix was calculated to explore the relationships between various numerical variables in the dataset. The cor() function was used to compute the correlation coefficients, and the results were stored in the correlation\_matrix variable.



|  |  |  |
| --- | --- | --- |
| VARIABLE NAME | DEPENDENT / INDEPENDENT | DEPENDENT VARIABLES |
| Year | INDEPENDENT | NIL |
| Energy | DEPENDENT | Loudness(0.72), Popularity(0.12) |
| Danceability | DEPENDENT | Popularity(0.22) |
| Loudness..dB. | DEPENDENT | Energy(0.72) |
| Liveness | INDEPENDENT | NIL |
| Length..Duration. | INDEPENDENT | NIL |
| Popularity | DEPENDENT | Energy(0.12), Danceability(0.22), Loudness(0.30) |

By looking at the correlation matrix, we can make the following observations about the dependency of variables on each other:

Year: There is a very weak negative correlation between the year and the other variables. This suggests that the year of the music release has little to no impact on the other attributes.

Energy: There is a positive correlation between energy and loudness (0.72), indicating that songs with higher energy tend to be louder. There is also a positive correlation between energy and popularity (0.12), suggesting that more energetic songs may be more popular.

Danceability: There is a positive correlation between danceability and popularity (0.22), implying that songs that are more danceable have a higher likelihood of being popular.

Loudness: There is a strong positive correlation between loudness and energy (0.72), suggesting that louder songs tend to have higher energy levels.

Liveness: There is a weak negative correlation between liveness and popularity (-0.11), implying that live recordings or songs with higher liveness may be less popular.

Length/Duration: There is no significant correlation between the length/duration of a song and the other variables (-0.003 to 0.026), indicating that song duration does not strongly influence the other attributes.

Popularity: Popularity shows a weak positive correlation with energy (0.12), danceability (0.22), and loudness (0.30), suggesting that these attributes may have some influence on a song's popularity.

It's important to note that correlation does not imply causation, and the strength and significance of these correlations may vary. Further statistical analysis or modeling would be required to determine the precise relationships and their significance.

**RECOMMENDATION MODEL**

The recommendation model used in this project is based on content-based recommender. Content-based recommender is a commonly used technique in recommendation systems that uses the commonly used method cosine similarity method.

Feature Extraction: Identify the relevant features or attributes that you want to use for similarity calculation. These features will be used to create a profile for each item (in this case, songs) in the dataset. For example, you may choose to use attributes like genre, artist, and song duration.

Vectorization: Convert the extracted features into a numerical representation for each song. This can be done using various methods such as one-hot encoding or numerical scaling. The goal is to represent each song as a vector in a multi-dimensional feature space.

Before building the recommendation model, the music dataset underwent preprocessing steps to transform it into a suitable format for collaborative filtering. This included extracting keywords from song titles and creating a binary matrix to indicate the presence of each music genre. The resulting feature matrix represented the features of each song in terms of keywords and genre presence. The data was transformed into a suitable format for modeling or algorithm implementation. Genre and artist data was converted to genre matrix and artist matrix. Further a similarity matrix was created.

User Profile: To generate personalized recommendations, the model required information about the user's preferences. In this case, the user's favorite music genres were defined and used to create a user profile matrix. The user profile matrix represented the user's preferences in terms of genre presence, aligning with the feature matrix.

Similarity Calculations:

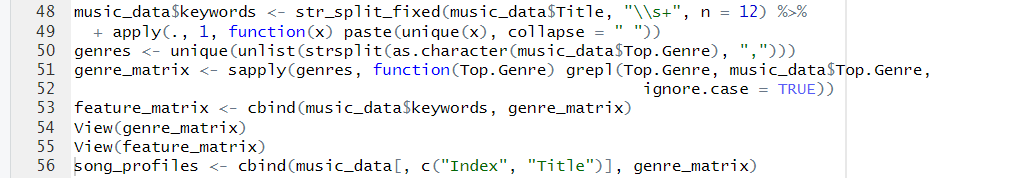
The next step involved calculating the similarity between the feature matrix and the user profile matrix. Similarity scores were computed using the cosine similarity measure, which measures the cosine of the angle between two vectors. It provides a measure of similarity between two vectors, with values ranging from -1 to 1, where 1 indicates perfect similarity.

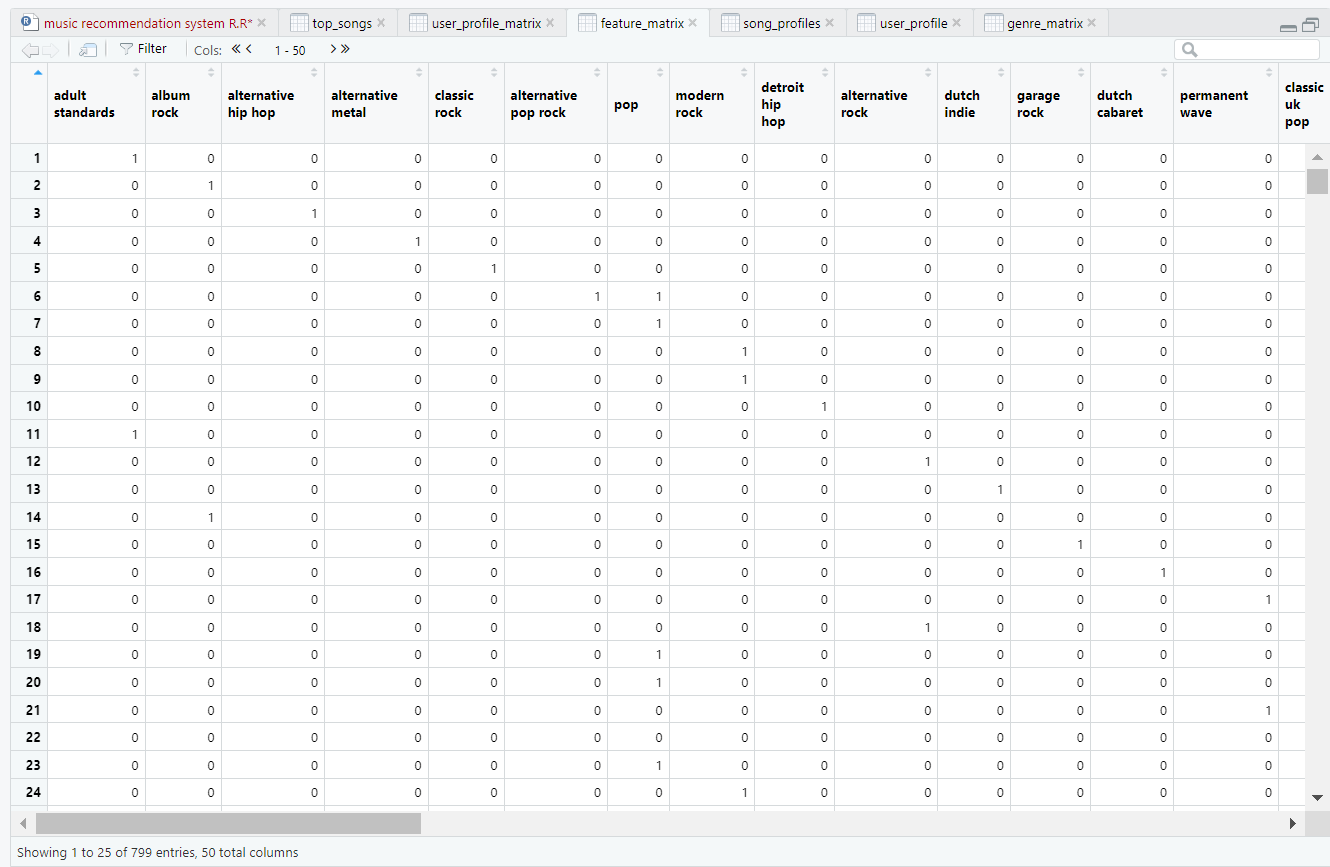
Recommendation Generation:

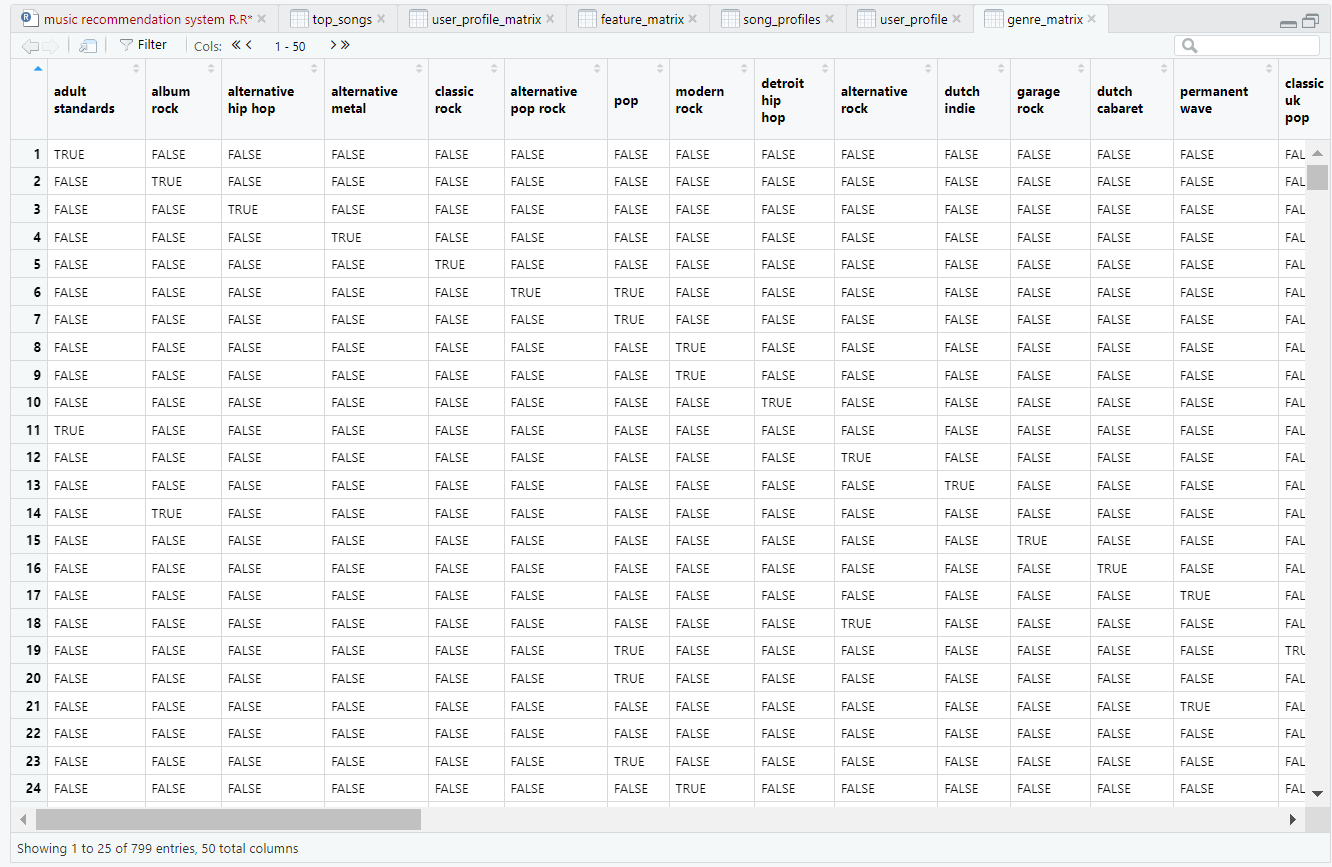
Once the similarity scores were calculated, they were merged with the original dataset to associate each song with its corresponding similarity score. The dataset was then sorted based on the similarity scores in descending order. The top N recommended songs were selected from the sorted dataset and presented to the user as personalized recommendations.

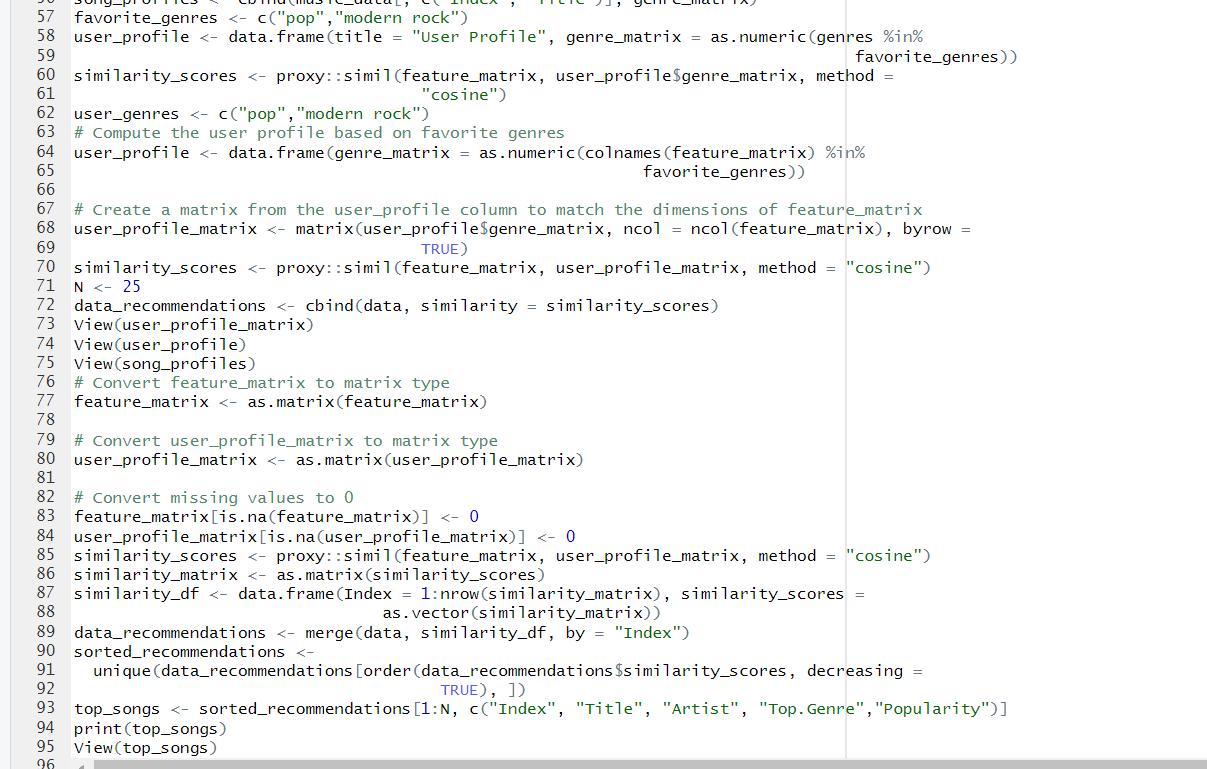
In addition to genre-based recommendations, artist-based recommendations were also provided. The process for artist-based recommendations was similar to genre-based recommendations. The presence of each artist in the dataset was identified, and a user profile matrix representing the user's favorite artists was created. Similarity scores were computed using the cosine similarity measure, and the dataset was sorted based on these scores to generate artist-based recommendations.

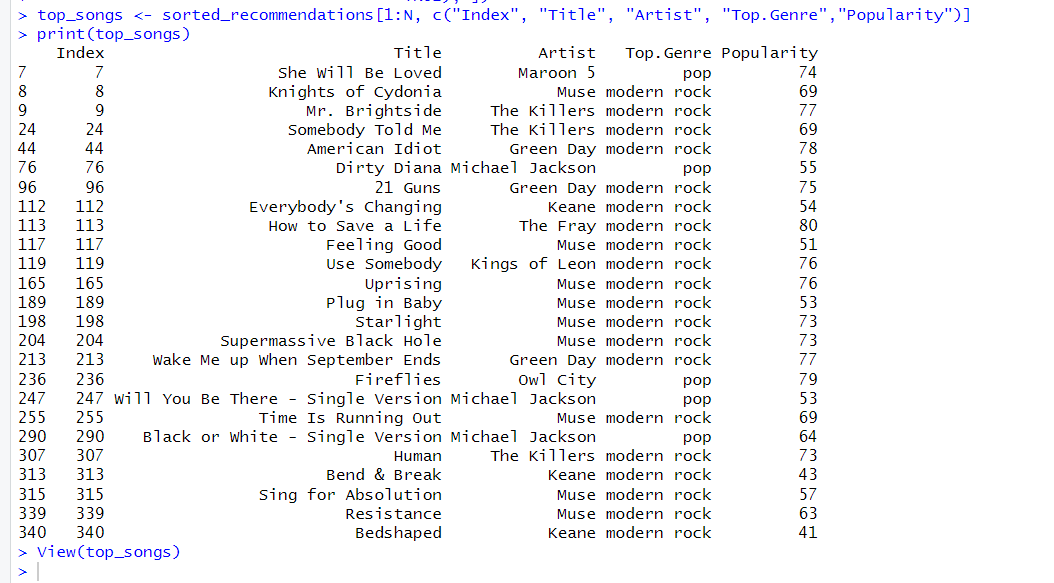
**BASED ON GENRE:**

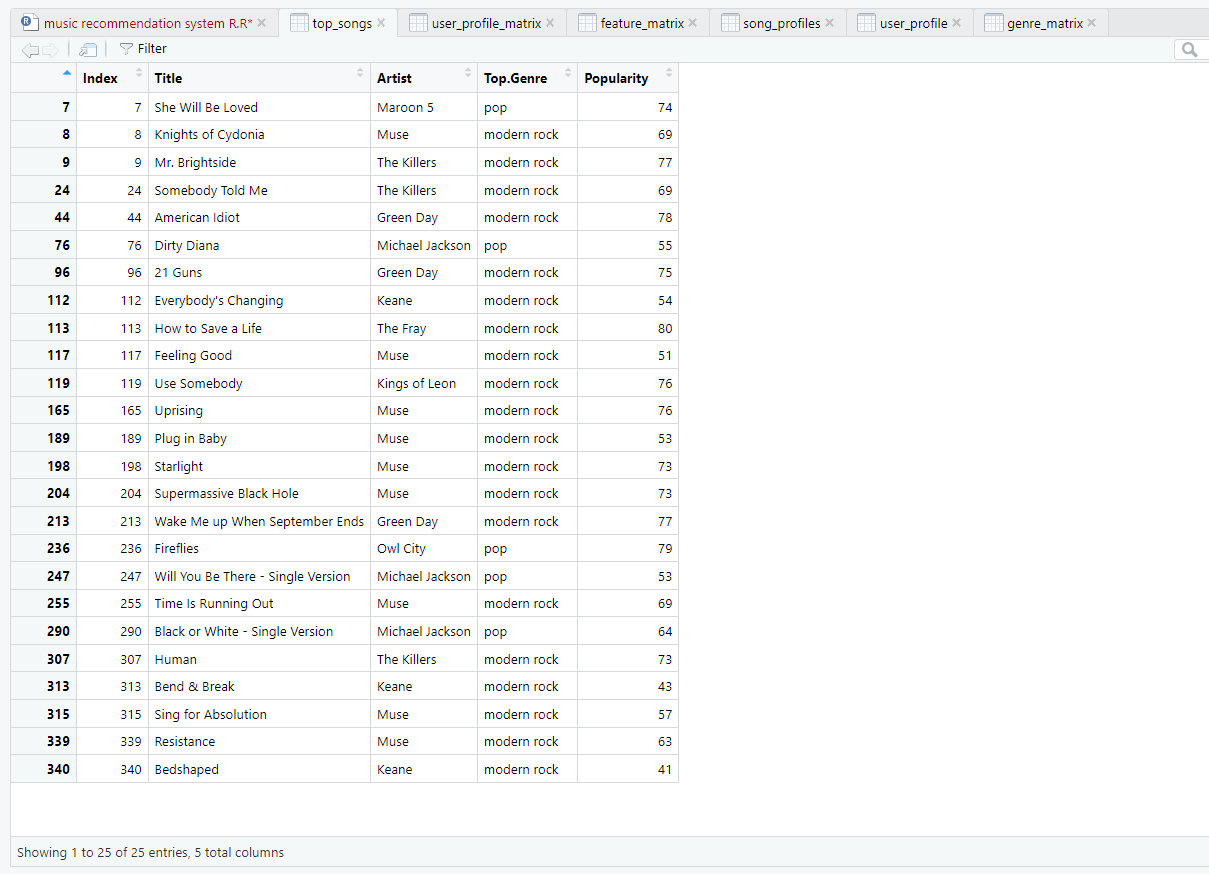




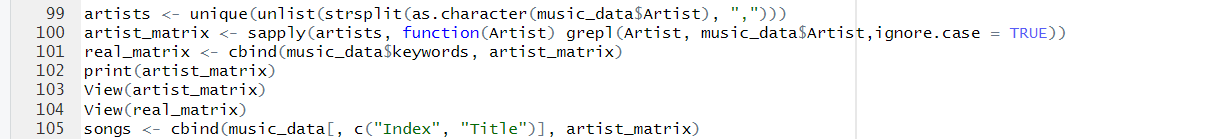


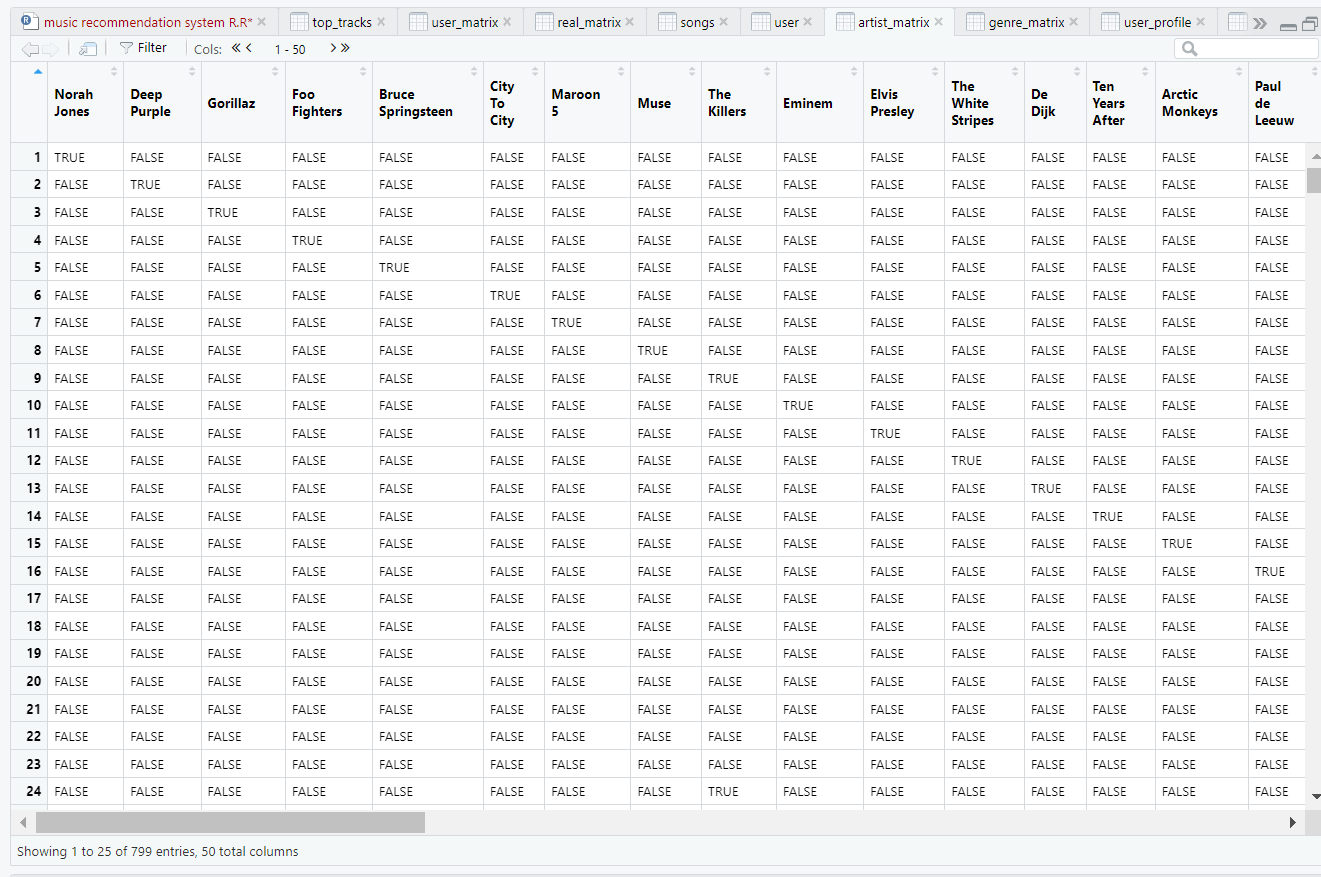


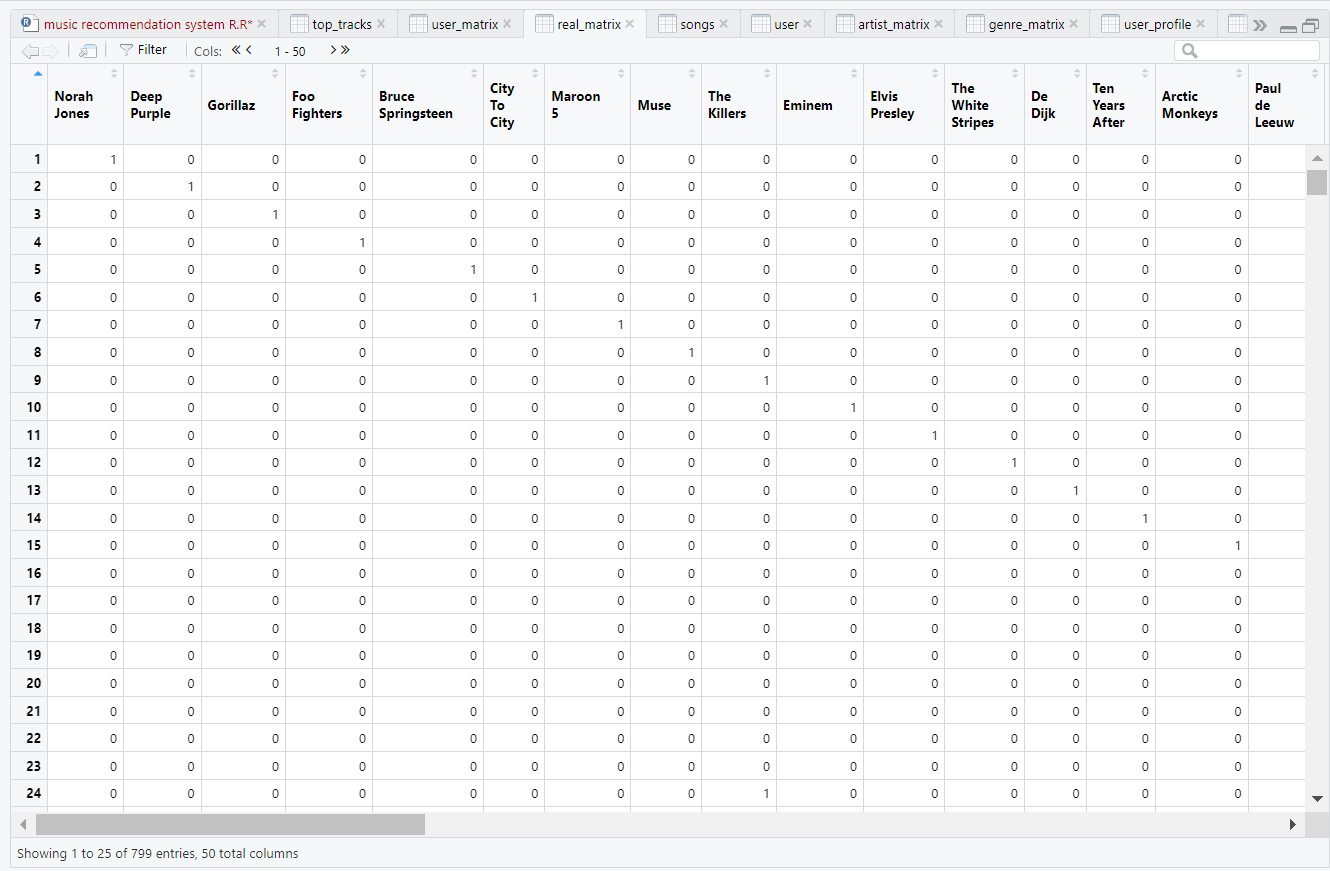




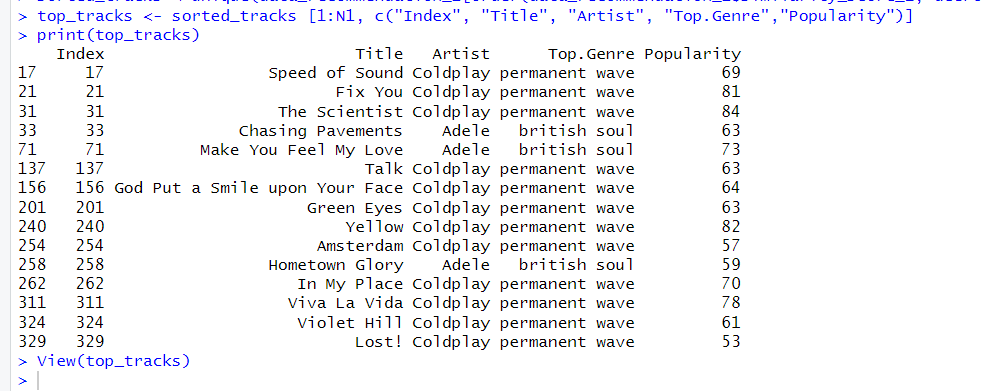
**BASED ON ARTISTS:**







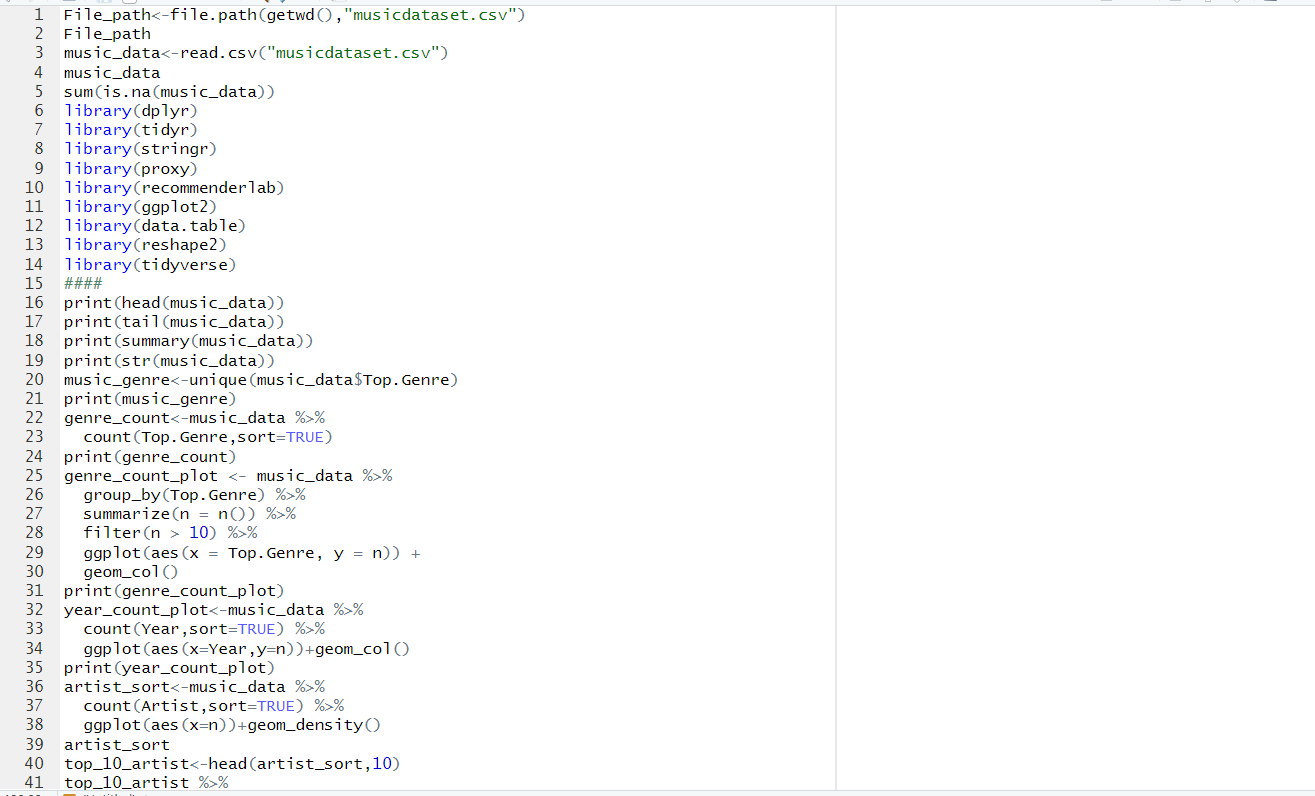


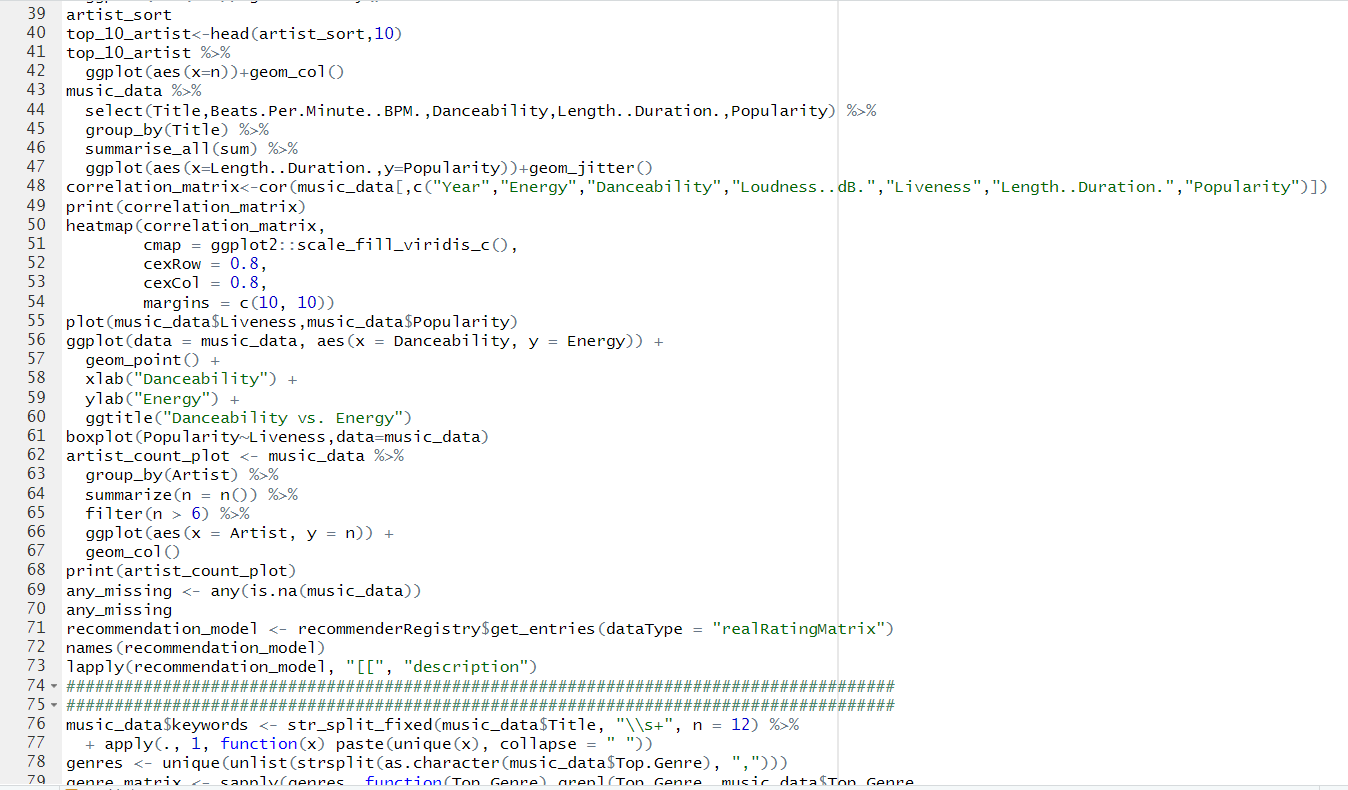


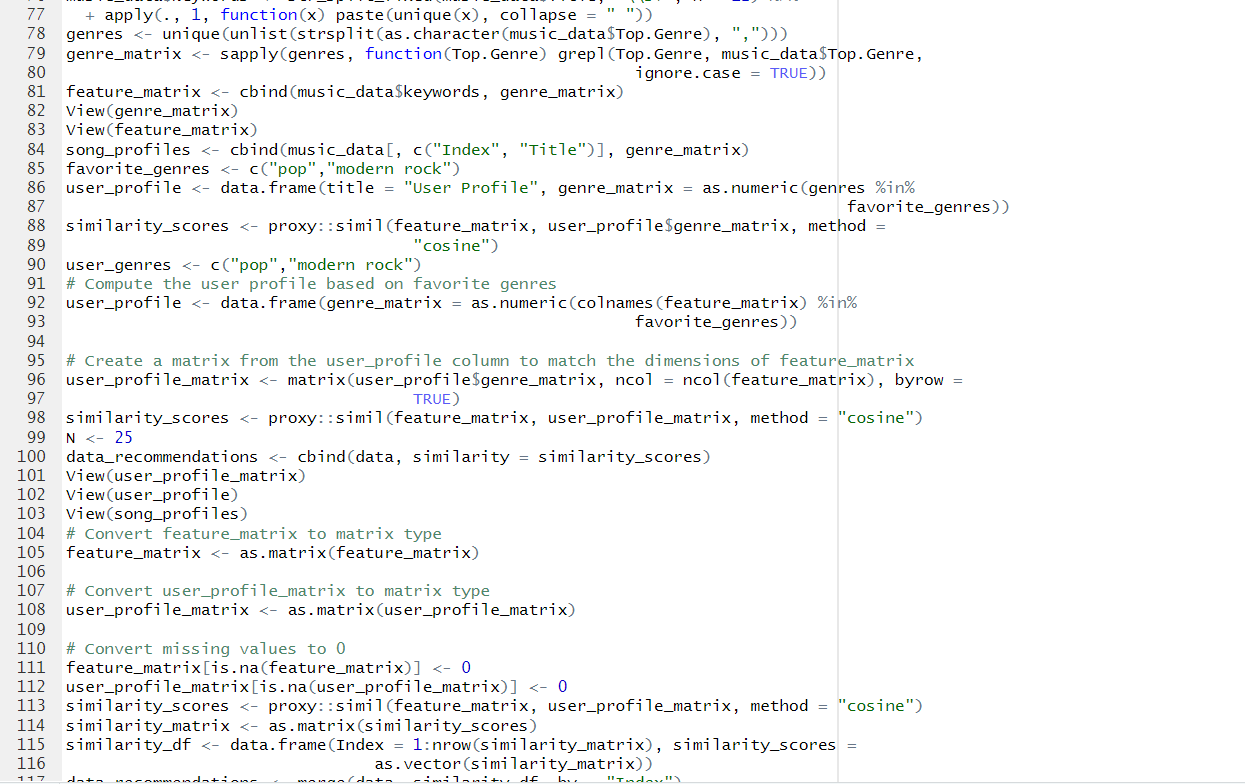


Overall, the recommendation model leveraged collaborative filtering techniques, specifically user-based collaborative filtering, to generate personalized music recommendations. By calculating similarity scores and sorting the dataset based on these scores, the model identified songs that were similar to the user's preferences in terms of genre or artist presence. This approach allows users to discover new songs that align with their musical tastes and enhances their overall music listening experience.

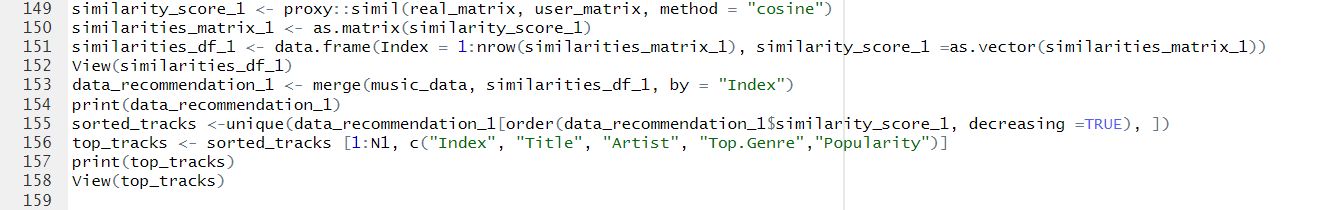
**MUSIC RECOMMENDATION SYSTEM R PROJET CODE**

****

****

****

****

****

**CONCLUSION**

In conclusion, this project successfully analyzed a music dataset and provided personalized music recommendations based on user preferences. By utilizing various data manipulation and visualization techniques, the project uncovered insights about the dataset, such as genre distribution, popularity of artists, and correlations between variables. The user-based and artist-based recommendation models enhanced the music listening experience by suggesting songs that aligned with the user's preferences. This project demonstrates the application of data analysis and recommendation systems in the music industry, providing valuable insights for music enthusiasts and industry professionals alike.

**SCOPE OF IMPROVEMENT**

Data Sparsity: The recommendation system may face challenges when dealing with sparse data, where users have rated or interacted with only a small portion of the available items. Sparse data can limit the accuracy and relevance of recommendations, particularly for niche or less popular items.

Limited Domain Coverage: The music recommendation system focuses on the analysis and recommendation of songs based on the provided dataset. It may not cover other aspects of the music domain, such as lyrics analysis, music production techniques, or cultural context, which could provide additional insights for a more comprehensive recommendation system.

Contextual Factors: The recommendation system may not consider contextual factors such as mood, location, or specific occasions, which can influence user preferences. Incorporating contextual information can enhance the relevance and personalization of recommendations.

Explainability and Transparency: Enhance the transparency and explainability of the recommendation system. Users may appreciate understanding why certain recommendations are being made, such as by providing explanations based on user preferences, similarity metrics, or music features.

Continuous Feedback and User Input: Collect feedback from users and incorporate it into the recommendation system's improvement cycle. Encourage users to provide explicit feedback, such as ratings or reviews, to further refine the system's understanding of their preferences.