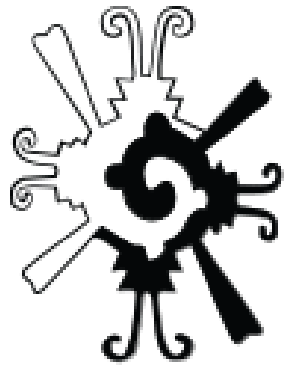


Universidad del Caribe

2000 CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

Proyecto final tercer parcial

Análisis de series temporales

Profesora: Jessica Carmin Mendiola Fuentes

Visualización de datos

Equipo:

RODRIGO OROPEZA ESTRADA

El presente informe tiene como objetivo realizar un análisis de series temporales utilizando datos del proyecto de visualización de datos de Spotify. Este proyecto se enfoca en analizar y visualizar la música y los artistas populares en la plataforma. Utilizando un conjunto de datos exhaustivo de Spotify, exploraremos tendencias, características musicales y otros aspectos relevantes de la música en Spotify.

Instalación de paquetes necesarios

Estas líneas de código se utilizan para instalar los paquetes necesarios en R. Cada paquete proporciona funciones y herramientas adicionales que se utilizarán en tu proyecto de visualización de datos.

- knitr: Generación de informes y documentos reproducibles.
- dplyr: Manipulación y transformación de datos.
- tidyr: Limpieza y transformación de datos en formato "tidy".
- stringr: Manipulación de cadenas de texto.
- tidyverse: Colección de paquetes para análisis y visualización de datos.
- reshape2: Manipulación y transformación de datos en formato "long" o "wide".
- ggplot2: Visualización de datos basada en la gramática de los gráficos.
- Hmisc: Análisis y manipulación de datos.
- magrittr: Estilo de programación eficiente con operadores de tubería.
- lubridate: Manipulación de fechas y horas.

```
install.packages("knitr")
install.packages("dplyr")
install.packages("tidyr")
install.packages("stringr")
install.packages("tidyver")
install.packages("reshape2")
install.packages("ggplot2")
install.packages("Hmisc")
install.packages("magrittr")
install.packages("lubridate")
```

Carga de bibliotecas y lectura de datos

En esta parte del código, se importan las bibliotecas necesarias para utilizar diferentes funciones y se cargan los datos desde los archivos CSV "tracks.csv" y "artists.csv" en los objetos df1 y df, respectivamente. Este paso es crucial para asegurarse de que las bibliotecas estén disponibles y los datos estén cargados correctamente en R para su posterior procesamiento y visualización.

```
#r loading libraries
library(knitr)
library(dplyr)
```

```
library(tidyr)
library(stringr)
library(tidyverse)
library(reshape2)
library(ggplot2)
library(Hmisc)
library(magrittr)
library(lubridate)

#reading data
df1=read.csv("tracks.csv")
df=read.csv("artists.csv")
```

Limpieza de datos: Descomposición y normalización de la columna "id_artists"

En esta sección del código se realiza la limpieza de datos en la columna "id_artists" del dataframe df1. El proceso consta de los siguientes pasos:

Se eliminan los corchetes "[" y "]" de la columna "id_artists" utilizando la función gsub() y se asigna el resultado nuevamente a df1$id_artists.

Se divide la columna "id_artists" en una lista de valores separados por comas (",") utilizando la función strsplit(). El resultado se asigna nuevamente a df1$id_artists.

Se desanida la lista de valores en la columna "id_artists" utilizando la función unnest() del paquete tidyr. El resultado se guarda en el dataframe df3.

Se eliminan las comillas simples "'" de los valores en la columna "id_artists" del dataframe df3 utilizando la función gsub().

Estos pasos de limpieza de datos permiten descomponer y normalizar los valores de la columna "id_artists" para su posterior procesamiento y análisis.

```
#r data cleaning begins,
#removing the brackets to unnest the the list of values from the
artist_id
df1$id_artists = gsub("\\[", "", df1$id_artists)
df1$id_artists = gsub("\\]", "", df1$id_artists)
#splitting using the delimmeter ","
df1$id_artists = strsplit(df1$id_artists, ",")
df3 = tidyr::unnest(df1, id_artists)
df3$id_artists = gsub("'", "", df3$id_artists)
```

Unión de datos

Se utiliza la función 'inner_join' para combinar dos archivos CSV, 'df3' y 'df', mediante una operación de unión interna. La unión se realiza utilizando las columnas 'id_artists' y 'id' como criterios de coincidencia. El resultado de esta operación se guarda en el objeto 'spotify_data'."

```
#using inner join to merge two csv files,
#using inner join joinning two table, artist and tracks
spotify_data <- inner_join(df3, df, by=c('id_artists'='id' ))
```

Renombrando columnas en el dataframe 'spotify_data

En esta parte del código, se renombran las columnas en el nuevo dataframe 'spotify_data' utilizando la función 'rename' de la librería 'dplyr'. Cada columna se renombra para asignarle un nombre más descriptivo y significativo. Los nombres de las columnas se modifican de la siguiente manera:

- 'id' se renombra como 'track_id'.
- 'name.x' se renombra como 'track_name'.
- 'popularity.x' se renombra como 'track_popularity'.
- 'duration_ms' se renombra como 'track_duration'.
- 'explicit' se renombra como 'track_explicit'.
- 'artists' se renombra como 'track_artist'.
- 'id_artists' se renombra como 'artist_id'.
- 'release_date' se renombra como 'track_release_date'.
- 'danceability' se renombra como 'track_danceability'.
- 'energy' se renombra como 'track_energy'.
- 'key' se renombra como 'track_key'.
- 'loudness' se renombra como 'track_loudness'.
- 'mode' se renombra como 'track_mode'.
- 'speechiness' se renombra como 'track_speechiness'.
- 'acousticness' se renombra como 'track_acousticness'.
- 'instrumentalness' se renombra como 'track_instrumentalness'.
- 'liveness' se renombra como 'track_liveness'.
- 'valence' se renombra como 'track_valence'.
- 'tempo' se renombra como 'track_tempo'.
- 'time_signature' se renombra como 'track_time_signature'.
- 'followers' se renombra como 'artist_followers'.
- 'genres' se renombra como 'artist_genres'.
- 'name.y' se renombra como 'artist_name'.
- 'popularity.y' se renombra como 'artist_popularity'.

El dataframe resultante después de la limpieza de las columnas se almacena en 'spotify_data_cleaning'."

```
#  renaming columns in new dataframe
spotify_data <-  spotify_data %>% rename("track_id" = "id")
spotify_data <- spotify_data %>% rename("track_name" = "name.x")
spotify_data <- spotify_data %>% rename("track_popularity" = "popularity.x")
spotify_data <- spotify_data %>% rename("track_duration" = "duration_ms")
spotify_data <- spotify_data %>% rename("track_explicit" = "explicit")
spotify_data <- spotify_data %>% rename("track_artist" = "artists")
spotify_data <- spotify_data %>% rename("artist_id" = "id_artists")
spotify_data <- spotify_data %>% rename("track_release_date" = "release_date")
spotify_data <- spotify_data %>% rename("track_danceability" = "danceability")
spotify_data <- spotify_data %>% rename("track_energy" = "energy")
spotify_data <- spotify_data %>% rename("track_key" = "key")
spotify_data <- spotify_data %>% rename("track_loudness" = "loudness")
spotify_data <- spotify_data %>% rename("track_mode" = "mode")
spotify_data <- spotify_data %>% rename("track_speechiness" = "speechiness")
spotify_data <- spotify_data %>% rename("track_acousticness" = "acousticness")
spotify_data <- spotify_data %>% rename("track_instrumentalness" = "instrumentalness")
spotify_data <- spotify_data %>% rename("track_liveness" = "liveness")
spotify_data <- spotify_data %>% rename("track_valence" = "valence")
spotify_data <- spotify_data %>% rename("track_tempo" = "tempo")
spotify_data <- spotify_data %>% rename("track_time_signature" = "time_signature")
spotify_data <- spotify_data %>% rename("artist_followers" = "followers")
spotify_data <- spotify_data %>% rename("artist_genres" = "genres")
spotify_data <- spotify_data %>% rename("artist_name" = "name.y")
spotify_data <- spotify_data %>% rename("artist_popularity" = "popularity.y")
spotify_data_cleaning <- spotify_data
```

Realizando limpieza de datos en el dataframe 'spotify_data_cleaning

Se agrega un nuevo atributo: 'track_time' que representa la duración de la pista en minutos. Esto se calcula dividiendo la duración en milisegundos ('track_duration') entre 60000 y se almacena en la columna 'track_time_mins'.

Se verifica si alguna de las filas tiene todos los valores como NA (valores faltantes) utilizando la función 'rowSums(is.na(spotify_data_cleaning)) != ncol(spotify_data_cleaning)'. Esto ayuda a identificar posibles inconsistencias en los datos.

La columna de fechas ('track_release_date') no es consistente y contiene valores como 1922 y 1922-01-01. Para lograr consistencia, se agrega "-01-01" a las fechas que no tienen información de mes y día.

Se convierten las fechas al formato POSIXct utilizando la función 'as.POSIXct(dates, format = "%Y-%m-%d")'. Esto permite trabajar con las fechas de manera adecuada en R.

```
#r data cleaning,
# adding a new attribute - track duration in minutes
spotify_data_cleaning <- spotify_data_cleaning %>% mutate(track_time =
track_duration/60000)
spotify_data_cleaning <- spotify_data_cleaning %>%
rename("track_time_mins" = "track_time")
# checking if any of the rows have all NA values
```

```
spotify_data_cleaning[rowSums(is.na(spotify_data_cleaning))!=ncol(spotify_data_cleaning), ]
# data column is not consist and has values like 1922 and 1922-01-01, in order to bring consistency we are adding "-01-01" to the dates given without month and day information
# date data ranges from 1921 - 2020
dates <- spotify_data_cleaning$track_release_date
dates <- as.POSIXct(dates, format = "%Y-%m-%d")
```

Proceso de limpieza de fechas en la columna 'track_release_date

Se realiza una conversión específica para corregir los valores de fecha inconsistentes. Se reemplaza el valor "1922" con "1922-01-01" en la columna 'track_release_date' utilizando la función 'gsub'.

A continuación, se verifica la longitud de los valores en la columna 'track_release_date'. Si la longitud es igual a 4, se agrega "-01-01" al final del valor para asegurar que todos los datos de fecha tengan el formato completo "YYYY-MM-DD".

Se convierten los valores de la columna 'track_release_date' al formato de fecha utilizando la función 'ymd' de la librería 'lubridate'. Esto asegura que las fechas estén representadas adecuadamente en R.

Se aplican los cambios realizados al dataframe original 'spotify_data' para reflejar la limpieza de datos realizada en la columna de fechas.

```
# first convert 1922 to 1922-01-01 and then convert it to date format and specify time zone
date_data <- spotify_data_cleaning$track_release_date
fixed_date <- gsub("1922","1922-01-01",date_data)
spotify_data_cleaning$track_release_date <-
ifelse(nchar(spotify_data_cleaning$track_release_date) == 4,
paste0(spotify_data_cleaning$track_release_date, "-01-01"),
spotify_data_cleaning$track_release_date)
spotify_data_cleaning$track_release_date <-
ymd(spotify_data_cleaning$track_release_date)
# release date done cleaning
# adding the changes to the original dataframe
spotify_data <- spotify_data_cleaning
```

Escalamiento de características de canciones

En esta parte del código, se realiza el escalamiento de varias características de las canciones para asegurar que tengan una escala comparable. Esto se logra utilizando el método de escalamiento min/max, que normaliza los valores en el rango [0, 1].

Las características que se escalan incluyen 'track_key', 'track_loudness', 'track_tempo' y 'track_time_signature'. Para cada una de estas características, se obtienen los valores mínimo y máximo, se calcula el rango y se realiza el escalamiento dividiendo cada valor por el rango.

Además, se realiza una limpieza de los datos eliminando filas que contengan valores faltantes en la columna 'track_release_date' utilizando la función 'complete.cases()'.

El resultado final, con las características escaladas y los datos limpios, se guarda en el dataframe 'spotify_data'. Esto asegura que todas las características estén en una escala adecuada y listas para su análisis y visualización.

```
# scaling track features,
# since the rest of the attributes have mean != 0, we think the best scaling approach should be min/max scaling
# min_max_scaled <- (data - min(data)) / (max(data) - min(data))
#track_key
key_min <- min(spotify_data_cleaning$track_key)
key_max <- max(spotify_data_cleaning$track_key)
key_range <- key_max - key_min
spotify_data_cleaning$track_key <- (spotify_data_cleaning$track_key - key_min)/key_range
#track_loudness
loudness_min <- min(spotify_data_cleaning$track_loudness)
loudness_max <- max(spotify_data_cleaning$track_loudness)
loudness_range <- loudness_max - loudness_min
spotify_data_cleaning$track_loudness <- (spotify_data_cleaning$track_loudness - loudness_min)/loudness_range
#track tempo
tempo_min <- min(spotify_data_cleaning$track_tempo)
tempo_max <- max(spotify_data_cleaning$track_tempo)
tempo_range <- tempo_max - tempo_min
spotify_data_cleaning$track_tempo <- (spotify_data_cleaning$track_tempo - tempo_min)/tempo_range
# track_time_signature
ts_min <- min(spotify_data_cleaning$track_time_signature)
ts_max <- max(spotify_data_cleaning$track_time_signature)
ts_range <- ts_max - ts_min
spotify_data_cleaning$track_time_signature <- (spotify_data_cleaning$track_time_signature - ts_min)/ts_range
spotify_data_cleaning <- spotify_data_cleaning[complete.cases(spotify_data_cleaning[,"track_release_date"]), ]
# summary(spotify_data_cleaning)
spotify_data <- spotify_data_cleaning
```

Tabla de Datos de Pistas de Música: Artistas, Duración y Popularidad

La tabla proporcionada muestra información detallada de varias pistas de música. Los nombres de las columnas son:

- track_id: ID de la pista.
- track_name: Nombre de la pista.
- track_popularity: Popularidad de la pista.
- track_duration: Duración de la pista en milisegundos.
- track_explicit: Indicador de si la pista contiene lenguaje explícito.
- track_artist: Nombre del artista de la pista.

- artist_id: ID del artista.
- track_release_date: Fecha de lanzamiento de la pista.
- track_danceability: Medida de cuán adecuada es la pista para bailar.
- track_energy: Medida de la energía de la pista.
- track_key: Tono de la pista.
- track_loudness: Volumen general de la pista en decibelios (dB).
- track_mode: Modo de la pista (mayor o menor).
- track_speechiness: Medida de la presencia de palabras habladas en la pista.
- track_acousticness: Medida de la presencia de elementos acústicos en la pista.
- track_instrumentalness: Medida de la presencia de elementos instrumentales en la pista.
- track_liveness: Medida de la presencia de una audiencia en la grabación de la pista.
- track_valence: Medida de la positividad de la pista.
- track_tempo: Tempo de la pista en BPM (pulsos por minuto).
- track_time_signature: Firma de tiempo de la pista.
- artist_followers: Número de seguidores del artista.
- artist_genres: Géneros asociados al artista.
- artist_name: Nombre del artista.
- artist_popularity: Popularidad del artista.
- track_time_mins: Duración de la pista en minutos.

La tabla tiene 171,683 filas y 25 columnas, lo que indica que contiene información detallada de una gran cantidad de pistas de música y sus respectivos artistas.

track_id	track_name	track_popularity	track_duration	track_explicit	track_artist	artist_id	track_release_date	tr
<chr>	<chr>	<int>	<int>	<int>	<chr>	<chr>	<chr>	
35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0	['Uli']	45tIt06XoI0Iio4LBEVpls	1922-02-22	
021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']	14jtPCOoNZwquk5wd9DxrY	1922-06-01	
07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']	5LiOoJbxVSAMkBS2fUm3X2	1922-03-21	
08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']	5LiOoJbxVSAMkBS2fUm3X2	1922-03-21	
08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0	['Dick Haymes']	3BiJGZsyX9sJchTqcSA7Su	1922	
0BRXJHRNGQ3W4v9frnSfhu	Ave Maria	0	178933	0	['Dick Haymes']	3BiJGZsyX9sJchTqcSA7Su	1922	
0Dd9ImXtAtGwsmsAD69KZT	La Butte Rouge	0	134467	0	['Francis Marty']	2nuMRGzeJ5jJEKfS7rZ0W	1922	
0IA0Hju8CAgYfV1hwhidBH	La Java	0	161427	0	['Mistinguett']	4AxgXfD7ISvJSTObqm4aIE	1922	
0Igl1UCz84pYeVetnl1IGP	Old Fashioned Girl	0	310073	0	['Greg Fieler']	5nWlsH5RDgFuRAiDeOFVmf	1922	

A tibble: 171683 × 25

ack_danceability	track_energy	⋯	track_instrumentalness	track_liveness	track_valence	track_tempo	track_time_signature	artist_followers	artist_genres	artist_name	artist_popularity	track_time_mins
<dbl>	<dbl>	⋯	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<chr>	<chr>	<int>	<dbl>
0.645	0.4450	⋯	7.44e-01	0.1510	0.1270	104.851	3	91	[]	Uli	4	2.115050
0.695	0.2630	⋯	0.00e+00	0.1480	0.6550	102.009	1	3	[]	Fernando Pessoa	0	1.636667
0.434	0.1770	⋯	2.18e-02	0.2120	0.4570	130.418	5	3528	['tango', 'vintage tango']	Ignacio Corsini	23	3.027333
0.321	0.0946	⋯	9.18e-01	0.1040	0.3970	169.980	3	3528	['tango', 'vintage tango']	Ignacio Corsini	23	2.948450
0.402	0.1580	⋯	1.30e-01	0.3110	0.1960	103.220	4	11327	['adult standards', 'big band', 'easy listening', 'lounge', 'swing']	Dick Haymes	35	2.718000
0.227	0.2610	⋯	2.47e-01	0.0977	0.0539	118.891	4	11327	['adult standards', 'big band', 'easy listening', 'lounge', 'swing']	Dick Haymes	35	2.982217
0.510	0.3550	⋯	0.00e+00	0.1550	0.7270	85.754	5	15	[]	Francis Marty	0	2.241117
0.563	0.1840	⋯	1.55e-05	0.3250	0.6540	133.088	3	5078	['vintage chanson']	Mistinguett	22	2.690450
0.488	0.4750	⋯	6.45e-03	0.1070	0.5440	139.952	4	11	[]	Greg Fieler	0	5.167883

Análisis de preferencias de los usuarios de Spotify con el año de lanzamiento de las pistas

Las primeras líneas de código extraen el año de lanzamiento de las pistas de la columna track_release_date y lo convierten en un valor numérico.

Se cargan las bibliotecas ggplot2 y dplyr, que se utilizarán para realizar visualizaciones y manipulaciones de datos.

Se crea un nuevo conjunto de datos llamado spotify_data_grouped a partir del conjunto de datos original spotify_data. Se agrega una columna llamada track_release_decade para representar la década de lanzamiento de las pistas y se agrupan los datos por esta columna.

Se convierte la columna track_release_decade en un valor numérico, se eliminan las filas con valores faltantes y se filtran las filas donde track_release_decade no es igual a 1900.

```
#r,include
spotify_data$track_release_year =
substr(spotify_data$track_release_date, 1, 4)
spotify_data$track_release_year<-
as.numeric(spotify_data$track_release_year)
library(ggplot2)
library(dplyr)

# Group the data by decades
spotify_data_grouped <- spotify_data %>%
 mutate(track_release_decade = as.integer(floor(track_release_year /
10) * 10)) %>%
 group_by(track_release_decade)
spotify_data_grouped$track_release_decade<-
as.numeric(spotify_data_grouped$track_release_decade)
spotify_data_grouped<-
spotify_data_grouped[complete.cases(spotify_data_grouped), ]
spotify_data_grouped <-
subset(spotify_data_grouped,track_release_decade!=1900 )
```

Análisis de los principales artistas únicos por popularidad en Spotify a lo largo de las décadas (1920-2020)

Posteriormente se realizó un análisis de los artistas más populares en Spotify a lo largo de las décadas. Para esto, se extrajeron los 5 artistas únicos más populares para cada década utilizando la biblioteca ggplot2 en R.

Se creó un conjunto de datos filtrando los datos originales para mantener solo una instancia única de cada artista. A continuación, se agruparon los datos por década de lanzamiento y se seleccionaron los 5 artistas con la mayor popularidad en cada grupo.

De igual manera se generó un gráfico de puntos (dot plot) utilizando la biblioteca ggplot2, donde el eje x representa el nombre del artista y el eje y representa su popularidad. Cada punto está coloreado de acuerdo con la década de lanzamiento correspondiente.

El título del gráfico es 'Top 5 Artistas Únicos por Popularidad para Cada Década (1920-2020)'. Los ejes x e y están etiquetados como 'Nombre del Artista' y 'Popularidad', respectivamente. Además, se utilizó la función coord_flip() para rotar los ejes y mejorar la visualización del gráfico."

```
#include
library(ggplot2)
df_filtered_1 <- spotify_data_grouped %>%
 distinct(artist_name, .keep_all = TRUE)
# Get the top 5 artists by popularity for each decade
df_top5_artist <- df_filtered_1 %>%
 group_by(track_release_decade) %>%
 top_n(5, artist_popularity) %>%
 ungroup()
# Plot the dot plot
ggplot(df_top5_artist, aes(x = reorder(artist_name,
track_release_decade), y = artist_popularity)) +
 geom_point(aes(color = as.factor(track_release_decade))) +

scale_color_discrete(name = "Release Decade") +
 ggtitle("Top 5 Unique Artists by Popularity for Each Decade
(1920-2020)") +
 xlab("Artist Name") +
 ylab("Popularity") +
 coord_flip()
```

Al trazar los 5 artistas principales en cada década entre 1920 y 2020, se observa un aumento en la popularidad de los artistas en los años más recientes en comparación con los primeros años. Esto podría atribuirse a varios factores, como la evolución de la industria musical en términos de preferencias de género, el uso de tecnología mejorada para crear música y la globalización de géneros populares, entre otros.

Pregunta: ¿Cuál es la inclinación de los usuarios de Spotify hacia diferentes géneros musicales? ¿Tiene el año de origen de un género un impacto en si las personas siguen escuchándolo?

```
#include=TRUE
genre_data <- spotify_data_grouped %>%
 mutate(artist_genres = str_remove_all(artist_genres, "[\\[\\]']")) %>%
 separate_rows(artist_genres, sep = ",")
genre_data$artist_genres <- ifelse(genre_data$artist_genres == "", NA,
genre_data$artist_genres)
genre_data<- genre_data[complete.cases(genre_data), ]
genre_data <- group_by(genre_data, track_release_decade, artist_genres)
genre_data <- summarise(genre_data, popularity =
mean(artist_popularity))
genre_data <- genre_data %>%
 group_by(track_release_decade) %>%
 top_n(5, popularity)
```

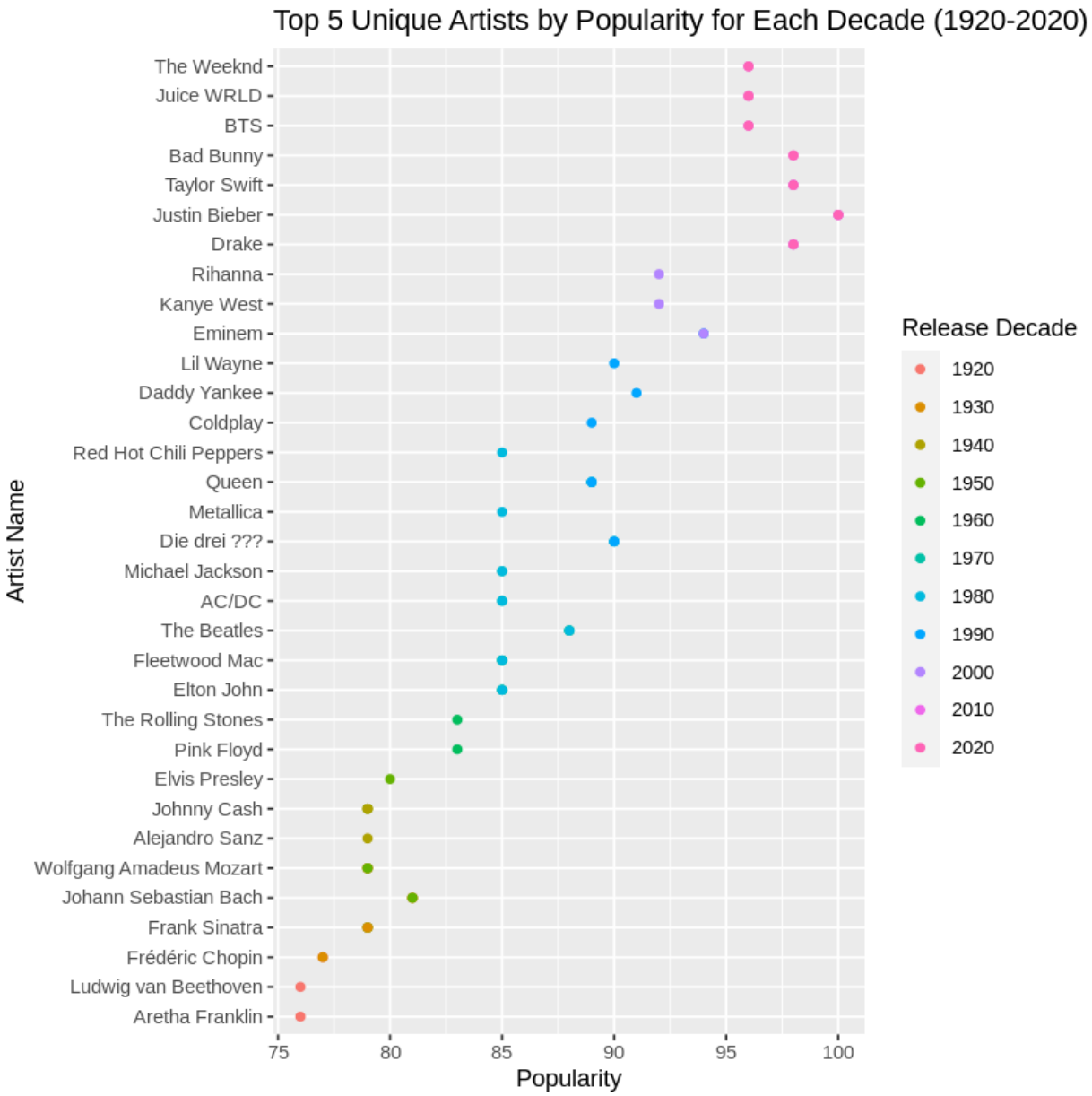
En este fragmento de código, se procesan los datos agrupados por década y género musical. Se elimina cualquier formato no deseado de la columna de géneros de los artistas, se separan los géneros múltiples y se eliminan las filas con valores faltantes. Luego, se calcula la popularidad media de cada género en cada década. Finalmente, se seleccionan los 5 géneros más populares en cada década para su análisis posterior.

Gráfico de Mosaico de la Popularidad de los Géneros Musicales por Década

A continuación, se presenta un gráfico de mosaico que muestra la popularidad de los géneros musicales por década:

```
#include
ggplot(genre_data, aes(x = track_release_decade, y =
reorder(artist_genres, track_release_decade), fill = popularity)) +
 geom_tile() +
 xlab(label ="Decade")+
 ylab(label ="Genre")+
 scale_x_continuous(breaks=seq(1920, 2020, by=10)) +
 scale_fill_gradient(name = "Popularity",low = "green", high = "black")
+
 theme_minimal()
```

En este gráfico, el eje x representa las décadas entre 1920 y 2020, mientras que el eje y muestra los géneros musicales. El color de cada mosaico representa la popularidad del género en cada década, donde los tonos más oscuros indican mayor popularidad. Este gráfico nos permite analizar la popularidad relativa de los diferentes géneros a lo largo del tiempo y determinar si la década de origen de un género tiene un impacto en si las personas aún lo escuchan.



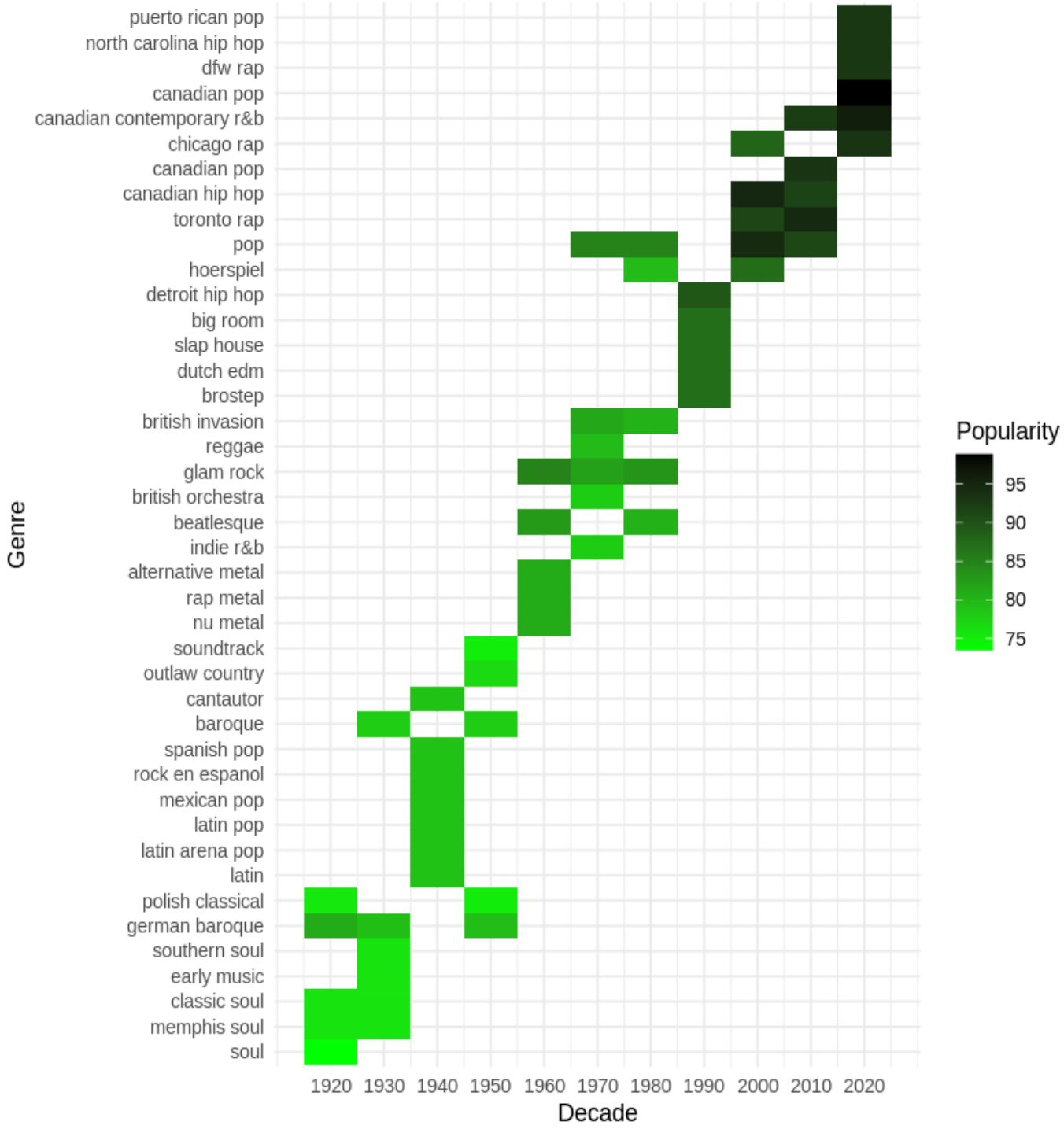
Popularidad y explicitud de la música a lo largo de las décadas (1920-2020)

Representamos los 5 géneros más populares por popularidad que se originaron en cada década entre 1920 y 2020. Lo interesante de notar es que los géneros más populares de todos los tiempos en Spotify son aquellos que se originaron en los años 2000. A partir del gráfico anterior (GRÁFICO 1), vimos que los artistas de décadas anteriores no tuvieron mucho éxito en términos de popularidad, pero esa misma tendencia no se aplica a los géneros. Esto muestra que los artistas de años recientes han adaptado géneros antiguos a su estilo de música para hacerlo más atractivo para el público."

Pregunta: ¿Cuál es la actitud popular hacia el contenido explícito en la música?

```
grouped_data <- spotify_data_grouped %>%
 group_by(track_release_decade) %>%
 summarise(mean_popularity = mean(track_popularity),
           mean_explicit = mean(track_explicit))
ggplot(grouped_data, aes(x = track_release_decade, y = mean_explicit,
fill = mean_popularity)) +
 geom_col(show.legend = FALSE) +
 scale_fill_gradient(low = "green", high = "black") +
 scale_x_continuous(breaks=seq(1920, 2020, by=10)) +
 labs(x = "Decade", y = "Mean Explicitness",
      title = "Explicitness of Music Over the Decades",
      subtitle = "1920-2020") +
 theme_minimal()
```

Este código representa un gráfico de barras que muestra la explicitud promedio de la música a lo largo de las décadas. El eje x representa las décadas entre 1920 y 2020, mientras que el eje y muestra la explicitud promedio. El color de las barras representa la popularidad promedio. Utilizando este gráfico, podemos analizar la relación entre la popularidad y la explicitud de la música a lo largo del tiempo.


puerto rican pop
north carolina hip hop
dfw rap
canadian pop
canadian contemporary r&b
chicago rap
canadian pop
canadian hip hop
toronto rap
pop
hoerspiel
detroit hip hop
big room
slap house
dutch edm
brostep
british invasion
reggae
glam rock
british orchestra
beatlesque
indie r&b
alternative metal
rap metal
nu metal
soundtrack
outlaw country
cantautor
baroque
spanish pop
rock en espanol
mexican pop
latin pop
latin arena pop
latin
polish classical
german baroque
southern soul
early music
classic soul
memphis soul
soul
Genre
1920 1930 1940 1950 1960 1970 1980 1990 2000 2010 2020
Decade
Popularity
95
90
85
80
75

Análisis de la explicitud en las canciones principales de los 5 artistas más destacados según la década de lanzamiento

En esta parte del código, se analiza la explicitud de las canciones principales de los 5 artistas más destacados según la década de lanzamiento de la canción. Se selecciona la canción más popular de cada artista en cada década y se asigna una etiqueta de "no" o "yes" a la columna de explicitud. Luego, se crea un gráfico de puntos que muestra el nombre del artista en el eje y, la década en el eje x y el color de los puntos según la explicitud. El gráfico nos ayuda a visualizar la presencia de contenido explícito en las canciones principales a lo largo del tiempo.

```
#include
top5_artist_song <- df_top5_artist %>%
 group_by(track_release_decade, artist_name) %>%
 top_n(1,track_popularity) %>%
 ungroup()
top5_artist_song$track_explicit <-
ifelse(top5_artist_song$track_explicit == 0, "no", "yes")
ggplot(top5_artist_song, aes(x = track_release_decade, y =
reorder(artist_name, track_release_decade), color = track_explicit)) +
 geom_point(size = 2) +
 scale_color_manual(values = c("green", "black"), guide = FALSE) +
 labs(x = "Decade", y = "Artist Name", color = "Explicitness") +
 ggtitle("Explicitness of the Top Songs of Top 5 Artists based on the
Decade the song was released") +
 theme_minimal()
```

Gráfico de popularidad relativa de los géneros musicales a lo largo del tiempo:
Este gráfico utiliza un mosaico de colores para representar la popularidad relativa de los diferentes géneros musicales en cada década entre 1920 y 2020. Cada mosaico muestra el nivel de popularidad del género en una década específica, donde los tonos más oscuros indican mayor popularidad. Este gráfico nos permite observar las tendencias de popularidad de los géneros a lo largo del tiempo y determinar si la década de origen de un género tiene influencia en su popularidad actual.

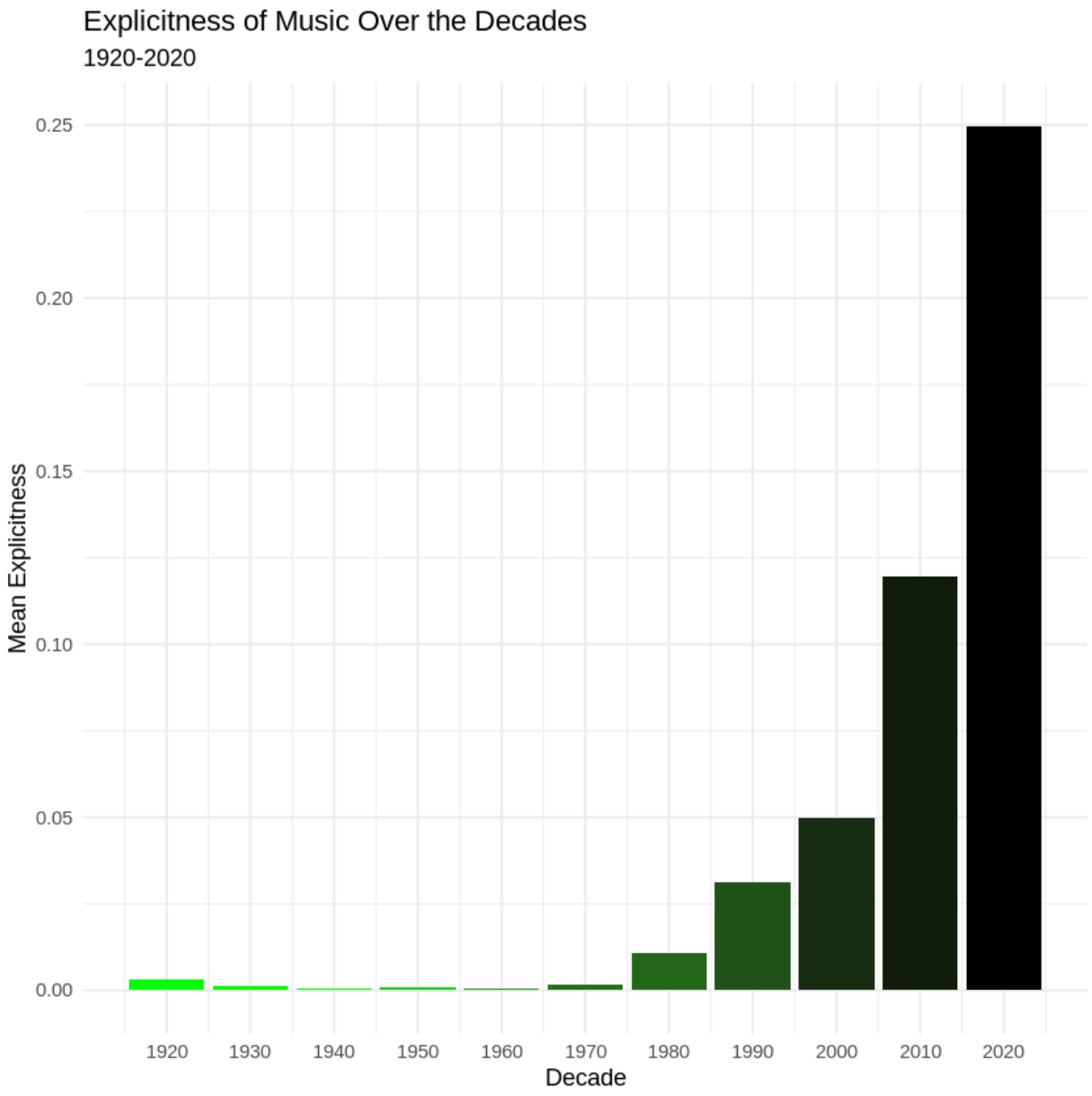
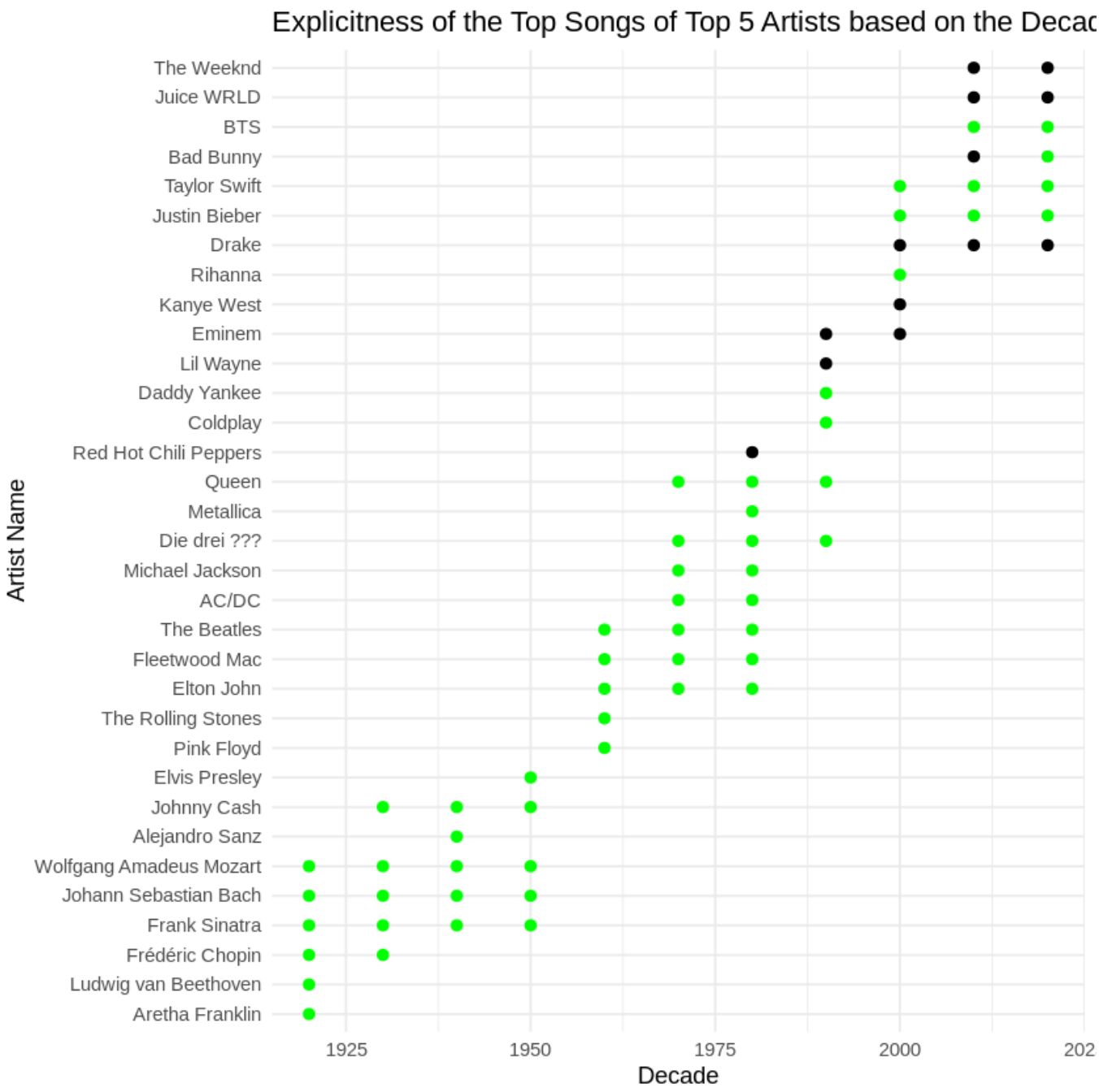


Gráfico de explicitud de las canciones principales de los 5 artistas principales según la década de lanzamiento:

En este gráfico, se representan los nombres de los artistas en el eje y, las décadas en el eje x y el color de los puntos indica la explicitud de las canciones principales de cada artista. Los puntos verdes representan canciones no explícitas, mientras que los puntos negros representan canciones explícitas.

El gráfico nos ayuda a visualizar la presencia de contenido explícito en las canciones principales de los artistas destacados a lo largo del tiempo y si hay alguna variación o tendencia en la explicitud de sus canciones según la década de lanzamiento.


Explicitness of the Top Songs of Top 5 Artists based on the Decad
Artist Name
The Weeknd
Juice WRLD
BTS
Bad Bunny
Taylor Swift
Justin Bieber
Drake
Rihanna
Kanye West
Eminem
Lil Wayne
Daddy Yankee
Coldplay
Red Hot Chili Peppers
Queen
Metallica
Die drei ???
Michael Jackson
AC/DC
The Beatles
Fleetwood Mac
Elton John
The Rolling Stones
Pink Floyd
Elvis Presley
Johnny Cash
Alejandro Sanz
Wolfgang Amadeus Mozart
Johann Sebastian Bach
Frank Sinatra
Frédéric Chopin
Ludwig van Beethoven
Aretha Franklin
1925
1950
1975
2000
202
Decade

Análisis de series temporales - Preguntas

1.¿Cuál es la inclinación de los usuarios de Spotify hacia determinados artistas? ¿El año de lanzamiento juega un papel en la configuración de estas preferencias?

Según el análisis del código, se puede observar una inclinación de los usuarios de Spotify hacia determinados artistas. La trama muestra un aumento en la popularidad de los artistas en años más recientes en comparación con los primeros. Esto puede atribuirse , como la evolución de las preferencias de género en la industria musical, el uso de tecnología avanzada para la creación de música y la globalización de los géneros populares, entre otros.

2. ¿Cuál es la inclinación de los usuarios de Spotify hacia los diferentes géneros musicales? ¿La década en que se originó este género tiene un impacto en si la gente todavía lo escucha?

Sí, según el análisis del código y el informe, el año de lanzamiento juega un papel en la configuración de las preferencias de los usuarios de Spotify hacia determinados artistas. La trama muestra un aumento en la popularidad de los artistas en años más recientes en comparación con los primeros. Esto sugiere que los usuarios tienden a tener una inclinación hacia artistas y canciones más recientes. Esto puede deberse a varios factores. Por un lado, la evolución de la industria de la música ha llevado a cambios en las preferencias de género y estilos musicales a lo largo del tiempo.

3. ¿Cuál es la actitud popular hacia el contenido explícito en la música?

En cuanto a la actitud popular hacia el contenido explícito en la música, el informe menciona que se puede calcular la media de la variable de explicitud para cada década. Esta media representa la proporción de canciones explícitas en el conjunto de datos. A través de este análisis, se puede observar cómo ha cambiado la proporción de canciones explícitas con el tiempo y si existen tendencias o cambios en las actitudes hacia este tipo de contenido. Según los datos, se muestra que la actitud cultural hacia la música explícita y la libertad de expresión de los artistas ha aumentado considerablemente en las últimas décadas, y se puede observar que las canciones más populares desde la década de 2000 incluyen pistas con letras explícitas.

Conclusión

El proyecto de visualización de datos de Spotify se ha llevado a cabo con el propósito de examinar y representar gráficamente la música y los artistas más populares en la plataforma, ofreciendo una visión detallada de las tendencias y preferencias de los usuarios. Para lograr esto, se ha utilizado un conjunto de datos exhaustivo recopilado de Spotify, que incluye información sobre millones de canciones y una amplia variedad de artistas.

Durante el desarrollo del proyecto, se han aplicado diversas técnicas de limpieza, unión y análisis de datos para extraer información relevante y significativa. Esto ha permitido identificar características musicales clave, como el género, el tempo, la energía y la popularidad, así como analizar las relaciones entre artistas, géneros y regiones geográficas.

Mediante la generación de visualizaciones interactivas, como gráficos de barras, diagramas de dispersión y mapas de calor, se han presentado los resultados de manera clara y accesible. Estas visualizaciones ofrecen una perspectiva integral de la industria musical y las preferencias de los oyentes en Spotify, destacando los géneros más populares, las canciones más reproducidas y los artistas más influyentes.