

# **ANÁLISIS DE DATOS**

# **SPOTIFY**

## **ANEXO**

---

**PROYECTO II**

**Grado en Ciencia de Datos**

**2023 - 2024**

*Elena Navarro, Javier*

*Amores Giner, Pau*

*Ciobanu Borinschi, Luminita*

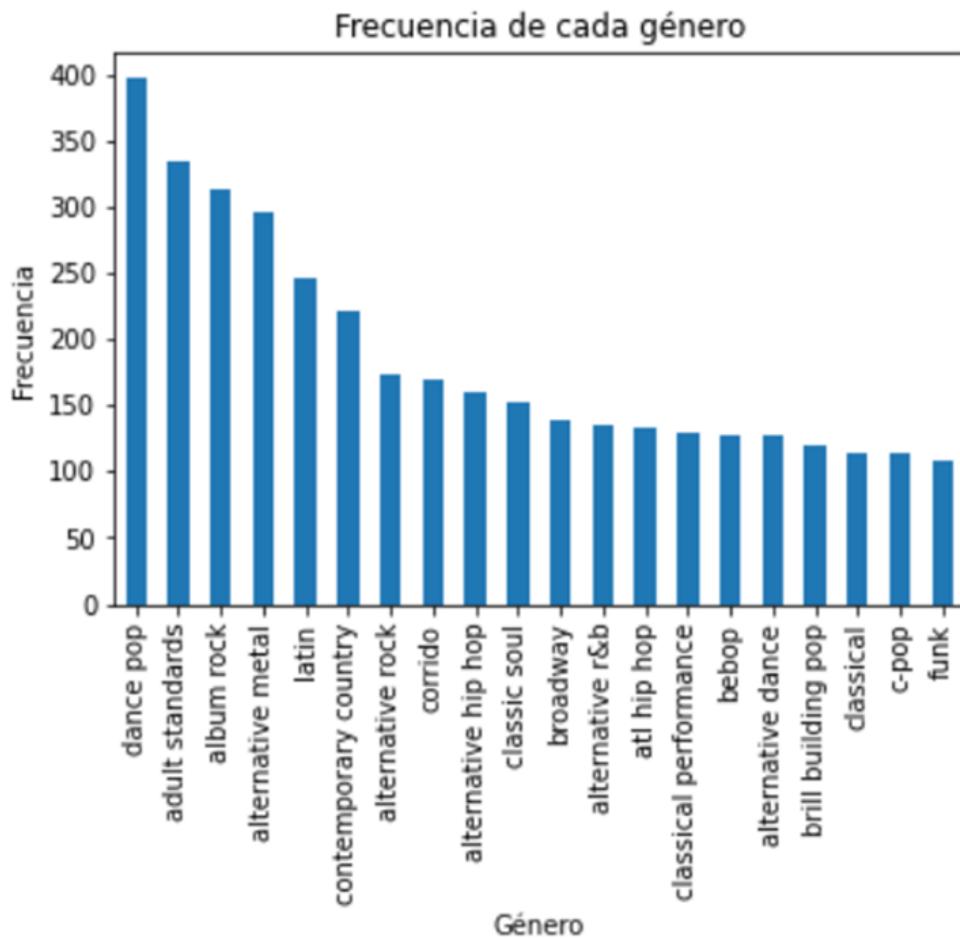
*Suleimanov Ismail Ogly, Rustam*

## **ÍNDICE**

- 1- DESCUBRIR LOS FACTORES CLAVE EN LAS RECOMENDACIONES DE CONTENIDO MUSICAL**
- 2- IDENTIFICACIÓN DE TENDENCIAS TEMPORALES**
- 3- SEGMENTACIÓN DE USUARIOS SEGÚN PREFERENCIAS REGIONALES**
- 4- PREDICCIÓN DEL ÉXITO DE LA CANCIÓN DE UN ARTISTA**
- 5- IDENTIFICACIÓN DE TENDENCIAS TEMPORALES**

## 1- DESCUBRIR LOS FACTORES CLAVE EN LAS RECOMENDACIONES DE CONTENIDO MUSICAL

Para realizar la regresión nos hemos basado en los géneros que más se repetían en las canciones del dataset 2020:



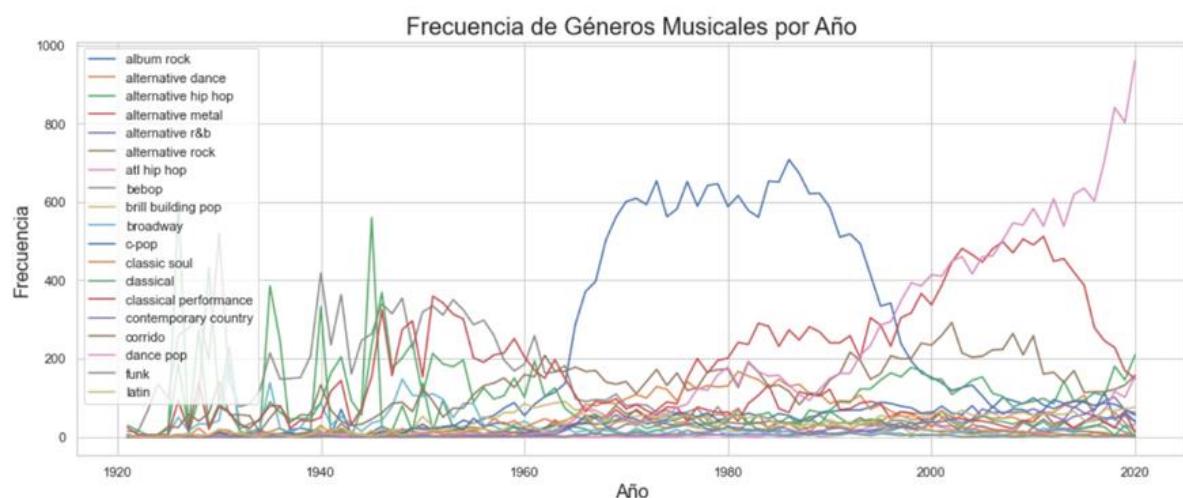
Los resultados de la predicción fueron los siguientes:

Accuracy (primera opción): 0.3822341857335128

	precision	recall	f1-score	support
adult standards	0.34	0.65	0.45	72
album rock	0.33	0.55	0.42	58
alternative dance	0.23	0.14	0.17	22
alternative hip hop	0.46	0.81	0.59	27
alternative metal	0.55	0.88	0.68	50
alternative r&b	0.70	0.22	0.33	32
alternative rock	0.46	0.18	0.26	34
atl hip hop	0.75	0.10	0.18	29
bebop	0.53	0.38	0.44	21
brill building pop	0.50	0.06	0.11	33
broadway	0.62	0.16	0.25	32
c-pop	0.33	0.11	0.16	19
classic soul	0.33	0.10	0.16	39
classical	0.26	0.26	0.26	19
classical performance	0.43	0.37	0.40	27
contemporary country	0.20	0.05	0.08	39
corrido	0.46	0.59	0.52	32
dance pop	0.32	0.71	0.44	82
funk	0.00	0.00	0.00	26
latin	0.25	0.10	0.14	50
accuracy			0.38	743
macro avg	0.40	0.32	0.30	743
weighted avg	0.40	0.38	0.33	743

(38% para el primer género y 59% contando los 2 más aproximados)

Para analizar la fiabilidad una vez hemos sacado la información por género para todos los datasets (todos comparten las variables utilizadas), hemos analizado la cantidad de canciones de cada género por año para el dataset de 2020 y 2023 (el otro dataset no incluye el año por lo que no se puede incluir):





Podemos observar concordancia entre los gráficos (el primero es para el dataset 2020 y el segundo para 2023), ya que a pesar de la independencia de los datos, vemos en el primer gráfico como el género de “dance pop” (barra rosa), es el que cobra mucha importancia a partir de los 2000 y en 2020 es el más frecuente con diferencia, y en el segundo gráfico se aprecia cómo en 2023 es el más frecuente también. Además, también son frecuentes hip hop y metal en 2020 según el primer gráfico, y son el segundo y tercero más frecuentes en 2023 en el segundo.

Por lo tanto, asumimos que la agrupación es, en cierta medida, fiable.

Los algoritmos que hemos desarrollado son los siguientes:

```
def sugerir(artista):

    if artista not in df11['artists'].values:
        return "Artista no encontrado"
    indice_artista = df11.loc[df11['artists'] == artista].index[0]
    artista_ref = stats3.iloc[indice_artista]

    listaerrores = []
    indices = []
    for i, (indice, valores_fila) in enumerate(stats3.iterrows()):
        if indice == indice_artista:
            continue
        error = 0
        for columna in stats3.columns:
            error += abs(artista_ref[columna] - valores_fila[columna])
        listaerrores.append((error, indice))

    listaerrores.sort()
    primeros_10_valores = listaerrores[:10]

    nombres_similares = [df11.loc[indice, 'artists'] for error, indice in primeros_10_valores]

    return nombres_similares

artistas_similares = sugerir("Plan B")
print(artistas_similares)

['Fundisha', 'Pitbull', 'Three 6 Mafia', 'Yandel', 'Sean Paul', 'J Balvin', 'Nicole Scherzinger', 'China Anne McClain', 'KEVVO', 'Jory Boy']
```

```

def sugerir2(cancion):
    if cancion not in dff['name'].values:
        return "canción no encontrada"
    indice_cancion = dff[dff['name'] == cancion].index[0]
    cancion_ref = stats4.iloc[indice_cancion]

    listaerrores = []
    indices = []
    for i, (indice, valores_fila) in enumerate(stats4.iterrows()):
        if indice == indice_cancion:
            continue
        error = 0
        for columna in stats4.columns:
            error += abs(cancion_ref[columna] - valores_fila[columna])
        listaerrores.append((error, indice))

    listaerrores.sort()
    primeros_10_valores = listaerrores[:10]

    nombres_similares = [dff.loc[indice, 'name'] for error, indice in primeros_10_valores]

    return nombres_similares

canciones_similares = sugerir2("GOOD BOY")
print(canciones_similares)

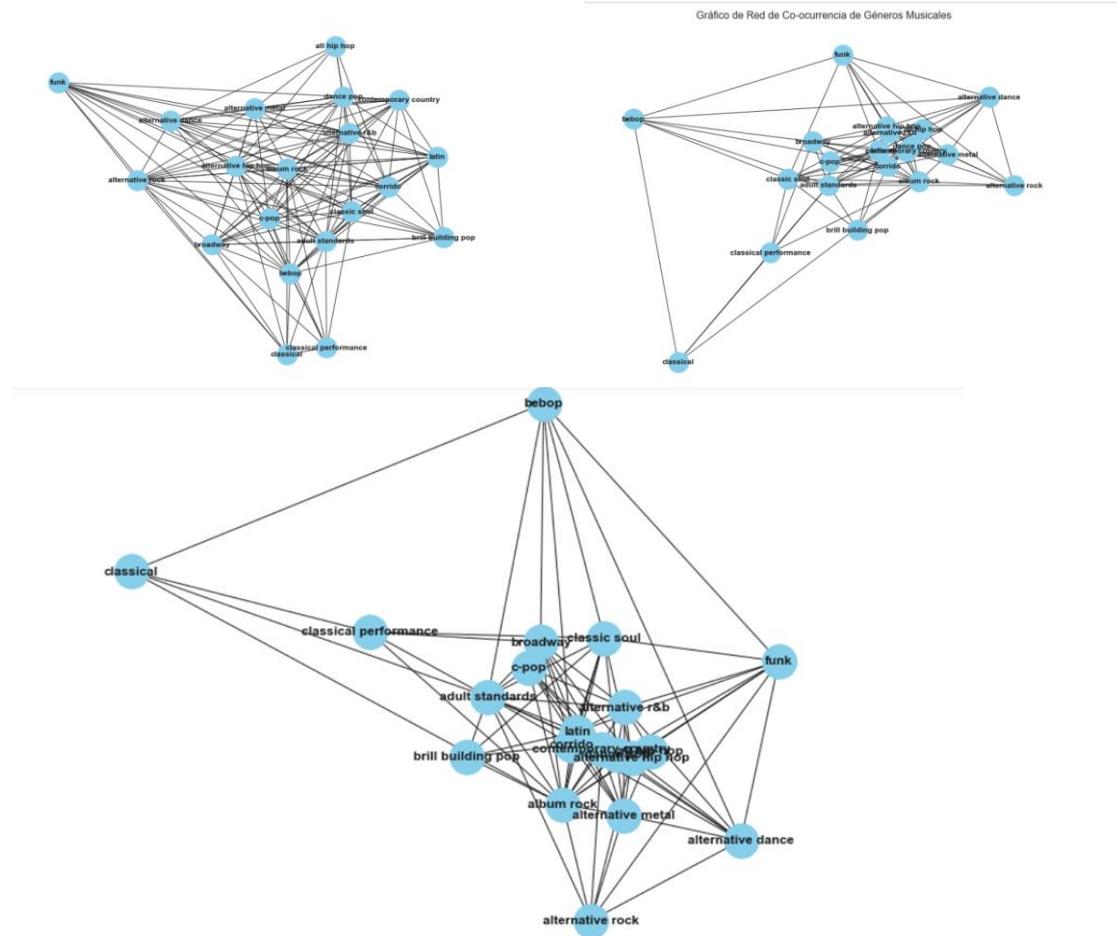
```

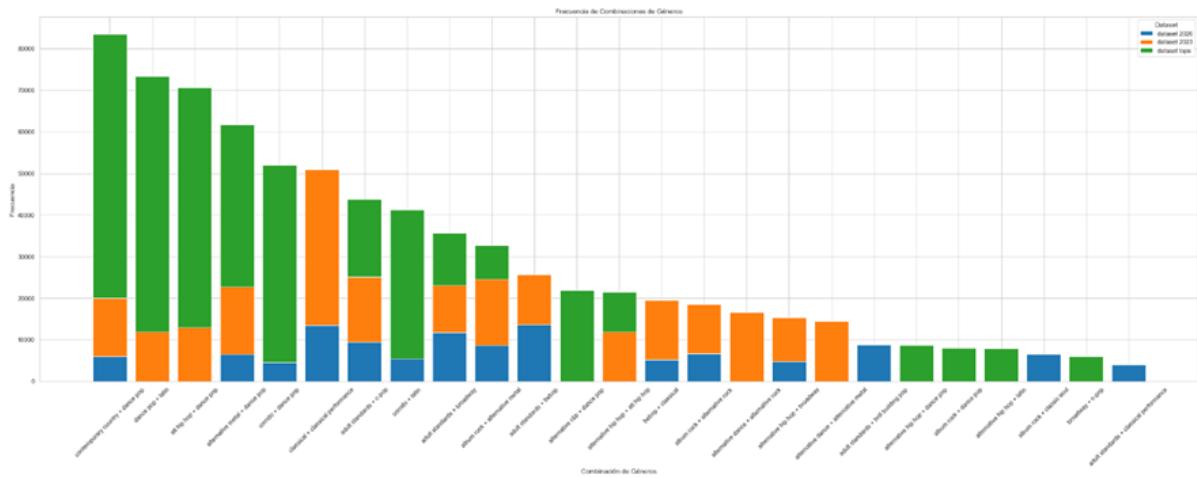
Activar Window  
Ve a Configuración

[ 'Sam And Delilah', 'Ogo Purabasi', 'A Change Is Gonna Come', 'Dry Your Eyes', 'Mentira, Mentira', 'Remember the Mountain Bed', 'Senza Mamma E Nnammurata', 'One Room Home', 'Werther, Acte III: Ciel! Ai-je compris?', 'Patanga Chala Hai Deepak Or' ]

Como se aprecia, comparan uno a uno cada artista y cada canción con la elegida en cuanto a las 8 variables del modelo de regresión y devuelven los artistas y canciones con menor diferencia (no era necesario escalar las variables, ya que iban todas de 0 a 1)

A partir de la predicción de 2 géneros para cada canción en los 3 datasets, sacamos los siguientes gráficos:

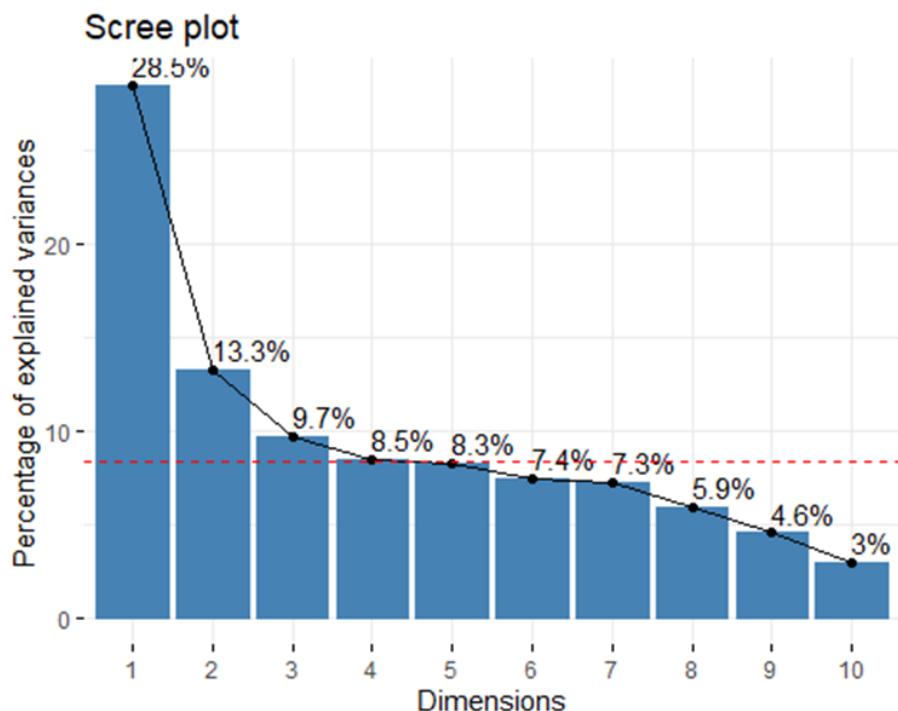




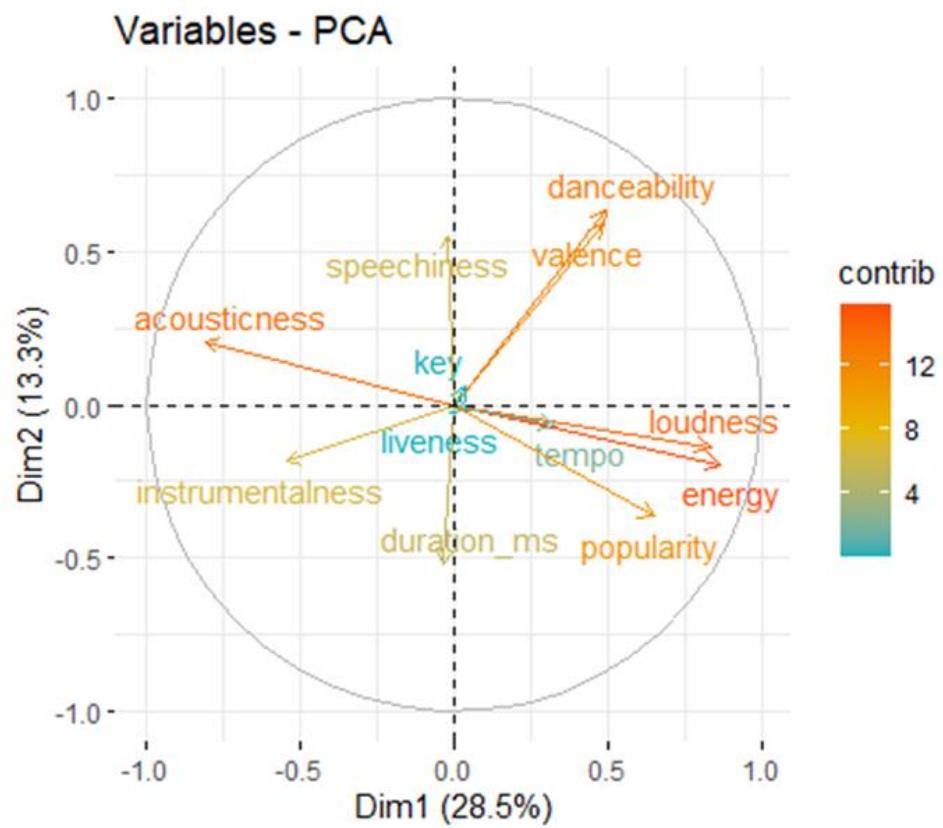
Hay 3 grafos, unos para cada dataset. Cada nodo es un género y arista significa que existe canción con los 2 géneros. Esto nos reporta una idea de los géneros que no tienen nada que ver entre sí, y por tanto no sería intuitivo recomendar ese tipo de música si no está relacionada con la del oyente. Como podemos ver en el grafo, la mayoría de géneros sí que están relacionados, pero algunos como música clásica, funk, bebop o rock están relacionados con menos géneros.

Además, en el gráfico de barras podemos saber cuales son las combinaciones de género más comunes, teniendo en cuenta las ocurrencias en las predicciones de los 3 datasets. Vemos que destacan las combinaciones country + pop, pop + latin, hip hop + dance pop, metal + pop, corrido + pop... Esto nos reporta información sobre los géneros que se podrían recomendar en función de los que el oyente escuche, ya que estarán más estrechamente relacionados.

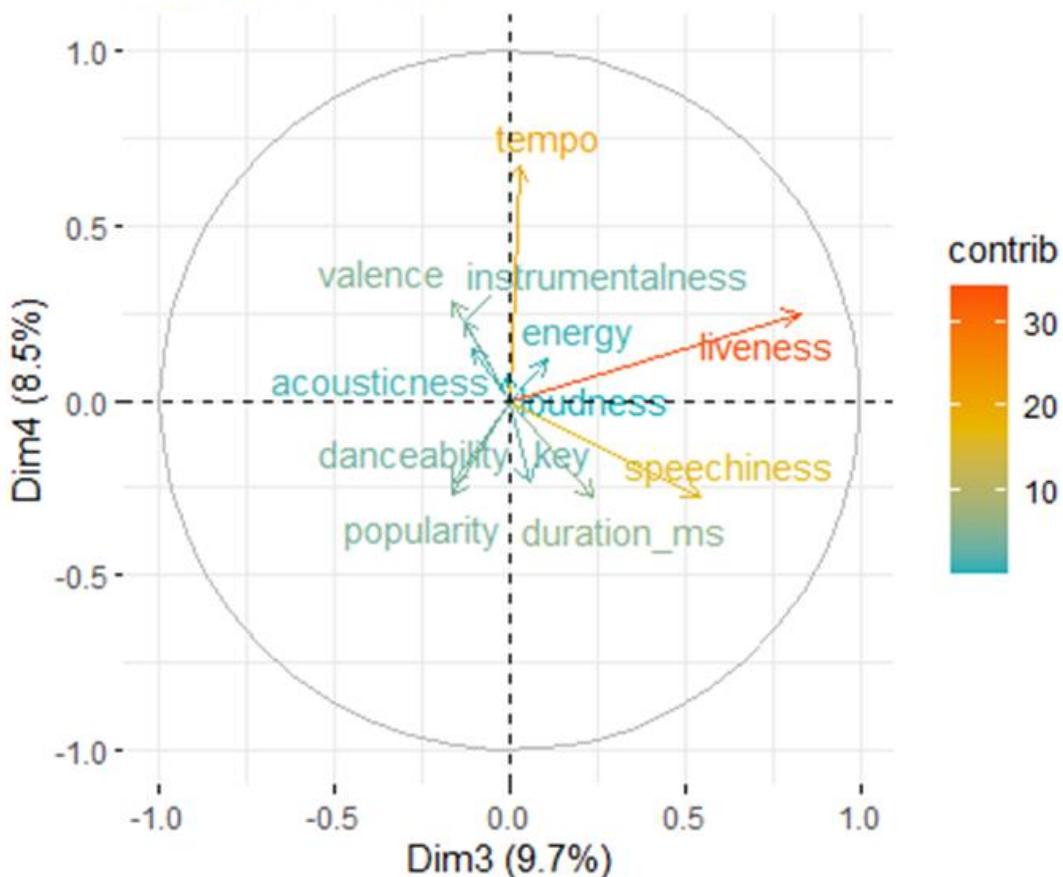
Por último, hemos realizado un análisis PCA sobre las variables de interés 'acousticness', 'danceability', 'energy', 'instrumentalness', 'liveness', 'loudness', 'speechiness' y 'valence', para tratar de entender las relaciones entre ellas.



Elegimos 4 componentes, ya que cumplen con el porcentaje de varianza explicada necesario.



## Variables - PCA



Si ambas variables tienen coeficientes de carga altos y del mismo signo en PC1, significa que las canciones con altos niveles de esa variables también tienden a tener un nivel alto de la otra variable.

Si una variable tiene una carga negativa en PC1 mientras que otra tiene una carga positiva, esto indica que tienen una relación negativa

Por lo tanto, de este análisis PCA obtenemos las siguientes relaciones:

En cuanto a las Relaciones en la Primera Componente Principal (PC1) (que es la que explica más porcentaje de la variabilidad, y , por tanto, más importante)

Positivamente Correlacionadas:

- Energy y Tempo

- Danceability y Energy

Negativamente Correlacionadas:

- Acousticness y Energy

-Instrumentalness y Speechiness

En cuanto a las Relaciones en el Segundo Componente Principal (PC2) (La segunda componente principal (PC2) explica la segunda mayor cantidad de varianza)

Positivamente Correlacionadas:

-Key y Mode

Negativamente Correlacionadas:

-Liveness y Acousticness

Estas relaciones serían otra forma de interpretar las diferencias entre canciones, para poder recomendar canciones acorde a los gustos.

## 2- IDENTIFICACIÓN DE TENDENCIAS TEMPORALES

## 3- SEGMENTACIÓN DE USUARIOS SEGÚN PREFERENCIAS REGIONALES

Para hacer el análisis de distribución por países pero de los artistas, tratamos de hacer web scraping para sacar la nacionalidad de cada artista, y no lo conseguimos, ya que habían demasiados artistas como para que pudiéramos hacerlo de forma eficiente.

primero nos guardamos los artistas en un archivo:

```
: import requests
from bs4 import BeautifulSoup
import pandas as pd

df = pd.read_csv("data_by_artist.csv")

b = df["artists"]
l = []
for a in b:
    l.append(a)

[""Cats" 1981 Original London Cast",
"Cats" 1983 Broadway Cast",
"Fiddler On The Roof" Motion Picture Chorus',
"Fiddler On The Roof" Motion Picture Orchestra',
"Joseph And The Amazing Technicolor Dreamcoat" 1991 London Cast',
"Joseph And The Amazing Technicolor Dreamcoat" 1992 Canadian Cast',
"Mama" Helen Teagarden',
"Test for Victor Young",
"Weird Al" Yankovic',
'$NOT',
'$atori Zoom',
'$pyda',
'$stupid Young',
'$uicideBoy$',
"In The Heights' Original Broadway Company",
"Legally Blonde' Ensemble",
"Legally Blonde' Greek Chorus",
"'Til Tuesday",
'((( O )))',
...]

with open("artistas.txt",'w',encoding='utf-8') as archivo:
    for artista in l:
        artista = str(artista)
        archivo.write(f'{artista}\n')
```

luego tratamos de hacer webscapping para sacar la nacionalidad de cada artista:

```
paises.py > [E] fichero
1 import requests
2 from bs4 import BeautifulSoup
3 from googlesearch import search
4
5 with open("artistas.txt", "r", encoding='utf-8') as fichero:
6     for linea in fichero:
7         busqueda1 = f'{linea.strip()} pais de origen' |
8         web = "site:wikkipedia.org"
9         try:
10             busqueda = search(busqueda1 + ' ' + web, sleep_interval = 1, timeout = 5, num_results = 1)
11             for e in busqueda:
12                 url = e
13                 print(url)
14                 break
15             time.sleep(2)
16             result = requests.get(url)
17             content = result.text
18             soup = BeautifulSoup(content, 'html.parser')
19         except:
20             pass
21
```

como hemos dicho, funcionaba pero al final paraba (intuimos que por demasiadas solicitudes), así que nos enfocamos en otros aspectos del proyecto

## → Introducción

Las preferencias musicales varían significativamente según la región geográfica. Este estudio tiene como objetivo investigar cómo los usuarios de diferentes regiones del mundo prefieren distintos géneros musicales y analizar el crecimiento de artistas en estos géneros. Mediante la segmentación de usuarios por región, se busca comprender mejor las dinámicas regionales y ofrecer insights valiosos sobre cómo las preferencias locales impactan en el consumo musical.

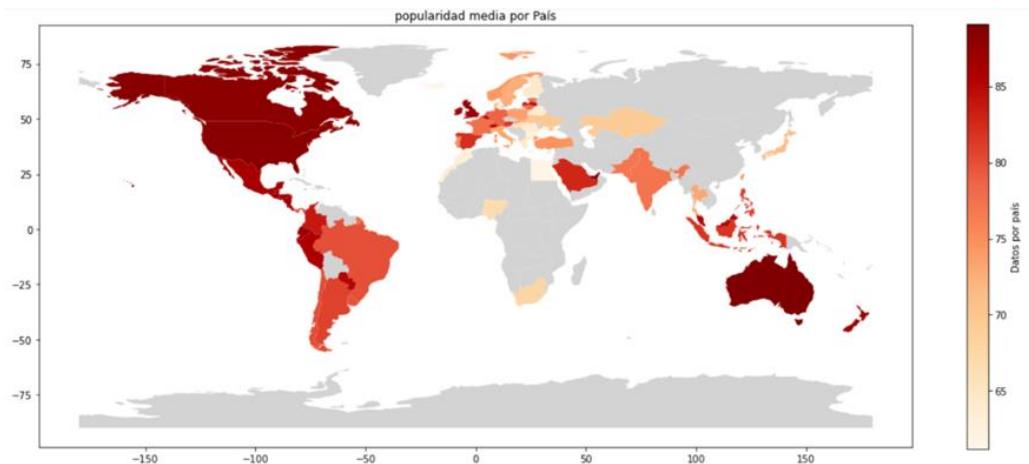
## → Metodología

Para realizar el análisis regional, se utilizó la librería `world` para visualizar la distribución de datos. La base de datos empleada contiene información sobre canciones que han aparecido en el top de Spotify en algún país. También se ha analizado el Top 50 diario por países usando R.

## → Resultados

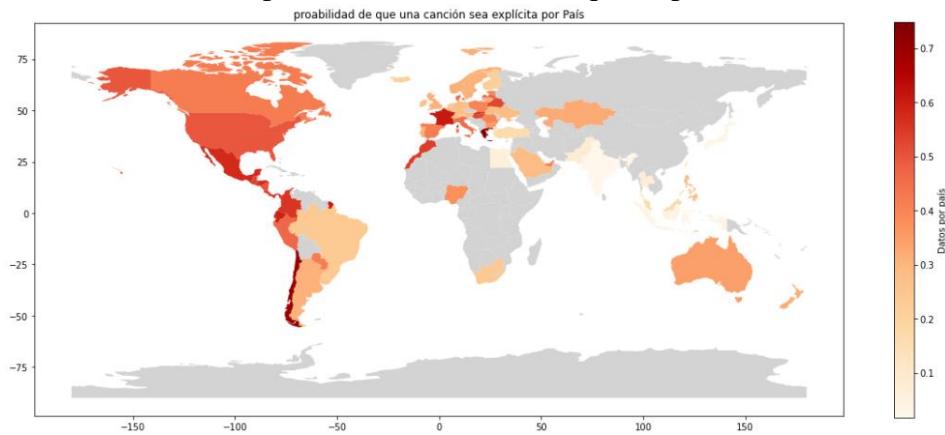
### 1. Popularidad Promedio de las Canciones por País:

- Se graficó la popularidad promedio de las canciones en cada país, lo que proporciona una idea de si la música escuchada en el país tiende a ser más conocida a nivel global.



## 2. Probabilidad de Canciones Explícitas por País:

- Se analizó la probabilidad de que una canción sea explícita en cada país, proporcionando una visión sobre el tipo de contenido musical que se prefiere en diferentes regiones.



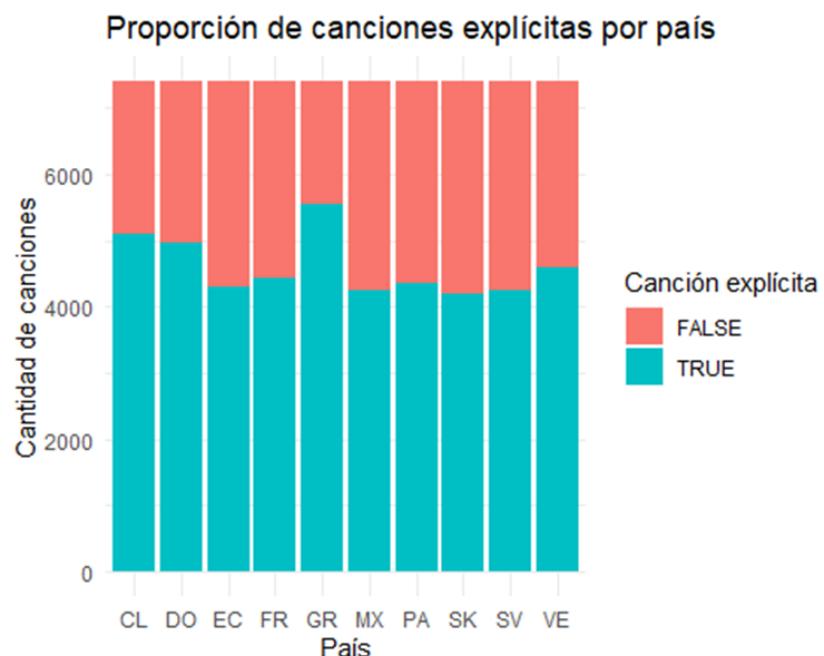
```

# Calcula la proporción de canciones explícitas por país y selecciona los
# 10 países con mayor proporción
top_countries <- datos %>%
  group_by(country) %>%
  summarize(prop_explicit = mean(is_explicit)) %>%
  top_n(10, prop_explicit)

# Filtra los datos originales para incluir solo los países en
# top_countries
filtered_data <- datos %>%
  filter(country %in% top_countries$country)

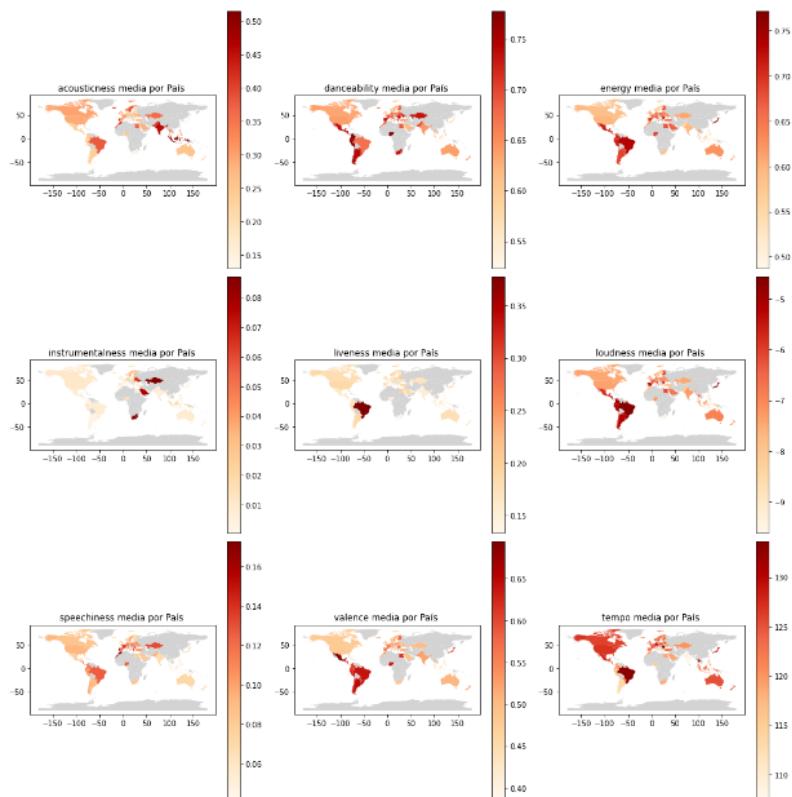
filtered_data %>%
  ggplot(aes(x = country, fill = is_explicit)) +
  geom_bar() +
  labs(title = "Proporción de canciones explícitas por país",
       x = "País",
       y = "Cantidad de canciones",
       fill = "Canción explícita") +
  theme_minimal()

```



### 3. Características de las Canciones por País:

- Se realizaron gráficos para analizar diversas características de las canciones (como tempo, energía, duración, etc.) por cada país, permitiendo entender mejor las preferencias musicales regionales.



Hay que tener en cuenta que la variable 'daily\_rank' tiene menor valor para una mejor posición.

```
correlation_matrix <- cor(datos$danceability, datos$daily_rank)
```

```
# Imprimir la matriz de correlación
print(correlation_matrix)
```

```
## [1] -0.04492597
```

La correlación de -0.0449 indica una relación lineal muy débil y negativa entre la bailabilidad de una canción (danceability) y su rango diario (daily\_rank). Esto significa que, en promedio, un ligero incremento en la bailabilidad de una canción se asocia con una muy ligera mejor posición en el ranking diario, pero esta relación es tan débil que prácticamente no tiene importancia estadística.

```

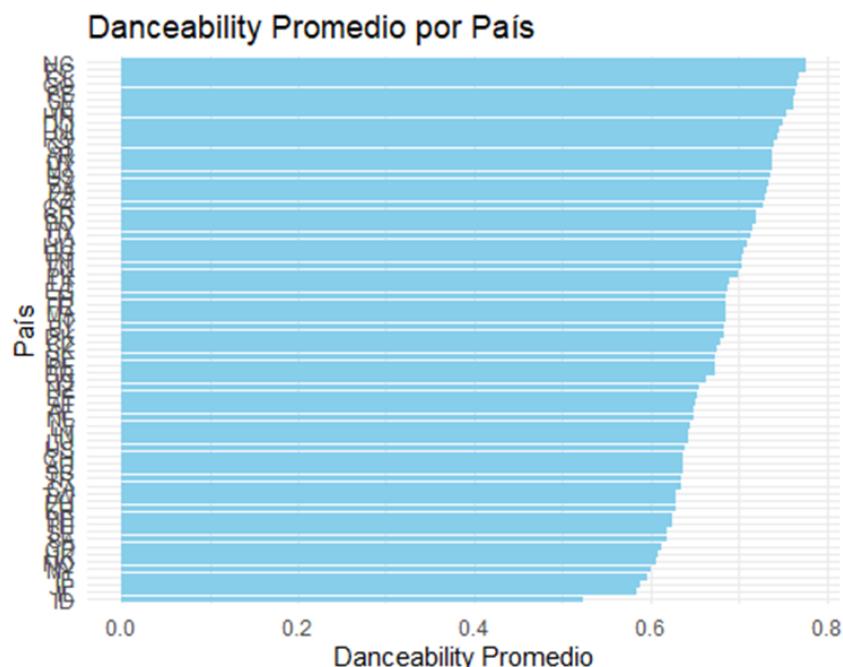
attributes_by_country <- datos %>%
  group_by(country) %>%
  summarise(across(c(danceability, energy, valence), mean, na.rm = TRUE))

## Warning: There was 1 warning in `summarise()` .
## i In argument: `across(c(danceability, energy, valence), mean, na.rm = TRUE)` .
## i In group 1: `country = "AE"` .
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `~.fns` through an anonymous function
## instead.
##
## # Previously
## across(a:b, mean, na.rm = TRUE)
##
## # Now
## across(a:b, \((x) mean(x, na.rm = TRUE))
```

**# Visualización de un atributo (por ejemplo, danceability)**

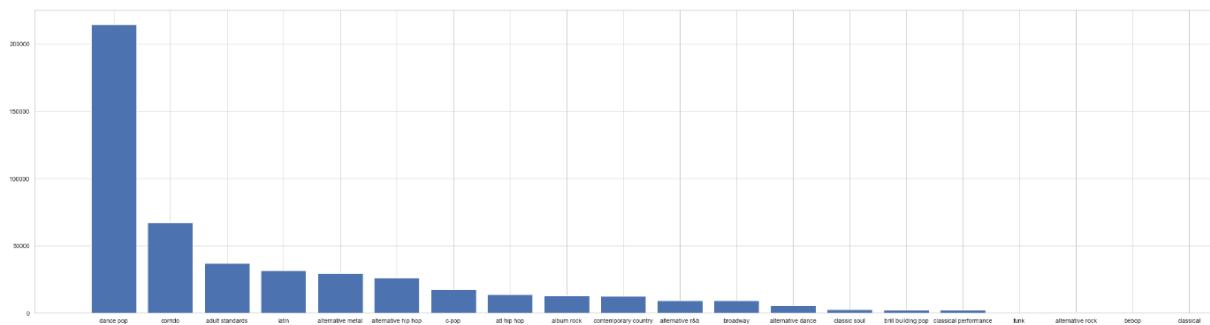
```

ggplot(attributes_by_country, aes(x = reorder(country, danceability), y =
danceability)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  theme_minimal() +
  labs(title = "Danceability Promedio por País",
       x = "País",
       y = "Danceability Promedio")
```



#### 4. Análisis del Género Musical por País:

- Se predijo y graficó la frecuencia de diferentes géneros musicales en la base de datos.
- Se creó una distribución específica de géneros por país.
- Para mejorar la comparación entre países, se escalaron las variables dividiéndolas entre su media. Valores superiores a 1 indican más escuchas de un género en comparación con la media global.



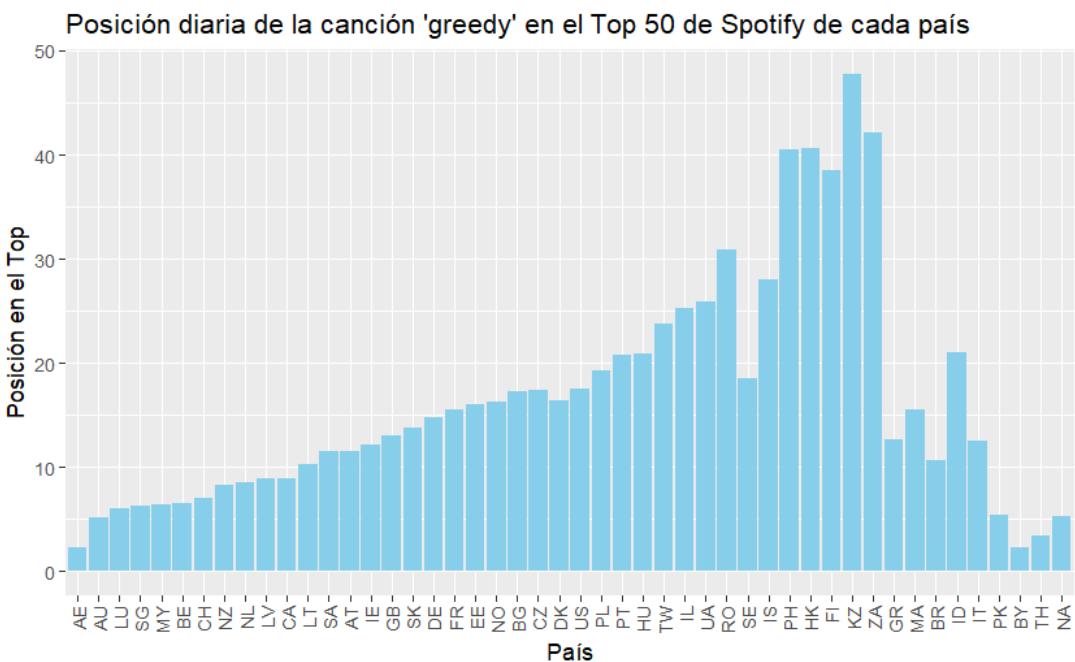
## 5. Análisis Top 50 Diario

A partir del data set Top Spotify Songs in 73 Countries con fecha de 13 de Marzo de 2024, después de preparar los datos y llevar a cabo un análisis exploratorio inicial, se ha buscado información como:

## Canción que más aparece en todos los rankings y su posición en cada ranking:

```
# Filtrar datos para la canción "greedy" con un rango diario no nulo y
# ordenar ascendente por ranking diario
greedy_data <- datos %>%
  filter(name == "greedy" & !is.na(daily_rank)) %>%
  mutate(daily_rank = daily_rank / 100) %>% # Dividir por 100 si es
# necesario
arrange(daily_rank)

# Crear el gráfico de barras ordenado ascendente
ggplot(greedy_data, aes(x = reorder(country, daily_rank), y =
daily_rank)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Posición diaria de la canción 'greedy' en el Top 50 de
Spotify de cada país",
    x = "País",
    y = "Posición en el Top") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



- Canción que más veces aparece en el Top 1:

```
# Filtrar datos para incluir solo las observaciones donde la canción
estuvo en la posición 1
top1_counts <- datos %>%
  filter(daily_rank == 1) %>%
  group_by(name) %>%
  summarise(num_times_top1 = n(), .groups = "drop") %>%
  arrange(desc(num_times_top1))

# Mostrar los resultados
print(top1_counts)

## # A tibble: 486 × 2
##   name           num_times_top1
##   <chr>          <int>
## 1 LUNA            565
## 2 PERRO NEGRO     501
## 3 La Diabla       417
## 4 Si No Estás      335
## 5 Seven (feat. Latto) (Explicit Ver.) 326
## 6 Last Christmas - Single Version    261
## 7 Según Quién       259
## 8 Stick Season      257
## 9 Lovin On Me        246
## 10 Beautiful Things 244
```

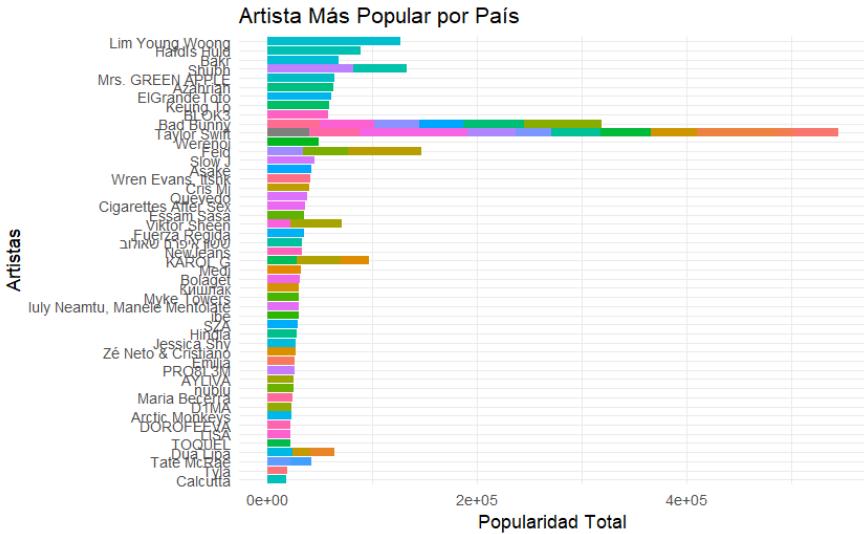
```
# Calcular la proporción de países que tienen "LUNA" en la posición 1
total_countries <- n_distinct(datos$country)
countries_with_luna_top1 <- n_distinct(luna_top1_by_country$country)
proportion_with_luna_top1 <- countries_with_luna_top1 / total_countries

# Mostrar la proporción
print(proportion_with_luna_top1)

## [1] 0.1111111
```

- Artista que más aparece en el top de cada país:

```
# Visualización de barras para el artista más popular por país
ggplot(top_artist_by_country, aes(x = reorder(artists, total_popularity),
y = total_popularity, fill = country)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_minimal() +
  labs(title = "Artista Más Popular por País",
x = "Artistas",
y = "Popularidad Total") +
  theme(legend.position = "none")
```



### → Observaciones Destacadas

- Japón: Predominancia del Bebop.
- Bielorrusia, Corea y Kazajistán: Preferencia por el Rock.
- Finlandia, Islandia y Marruecos: Alta presencia de Música Clásica.
- Sudáfrica: Destaca el Funk.

### → Conclusión

El estudio revela que las preferencias musicales varían notablemente según la región geográfica. Estas diferencias reflejan no solo gustos culturales diversos sino también la influencia de artistas locales y géneros musicales específicos en cada país. La segmentación por región proporciona una visión detallada de estas preferencias, permitiendo a la industria musical adaptar estrategias de mercado y promoción de manera más eficaz. Además, el análisis de características y la predicción de géneros musicales ofrecen una comprensión profunda de las dinámicas regionales, destacando la importancia de considerar factores locales en el consumo musical global.

## 4. PREDICCIÓN DEL ÉXITO DE UNA CANCIÓN DE ARTISTA

En este apartado se incluyen todos los pasos y códigos en R y explicaciones extra que no se han explicado con suficiente detalle del modelo predictivo del éxito de una canción de un artista.

Pasos:

1. Seleccionamos las variables numéricas relevantes que utilizaremos como variables explicativas y la variable explicada

En nuestro caso, deseamos identificar la fórmula que determine las combinaciones a emplear en una canción para alcanzar el éxito. Por consiguiente, utilizaremos las siguientes variables explicativas: danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo y duration\_sec, que son las que caracterizan una canción, y la variable explicada, track\_popularity, que buscamos incrementar al máximo mediante distintas combinaciones de las variables explicativas.

Para llevar a cabo este paso en R, dividimos estas variables en dos grupos: las variables explicativas y la variable explicada.

```
# Seleccionar solo las variables numéricas relevantes
numeric_vars <- datos3 %>% select(danceability, energy, loudness, speechiness,
                                      acousticness, instrumentalness, liveness,
                                      valence, tempo, duration_sec)

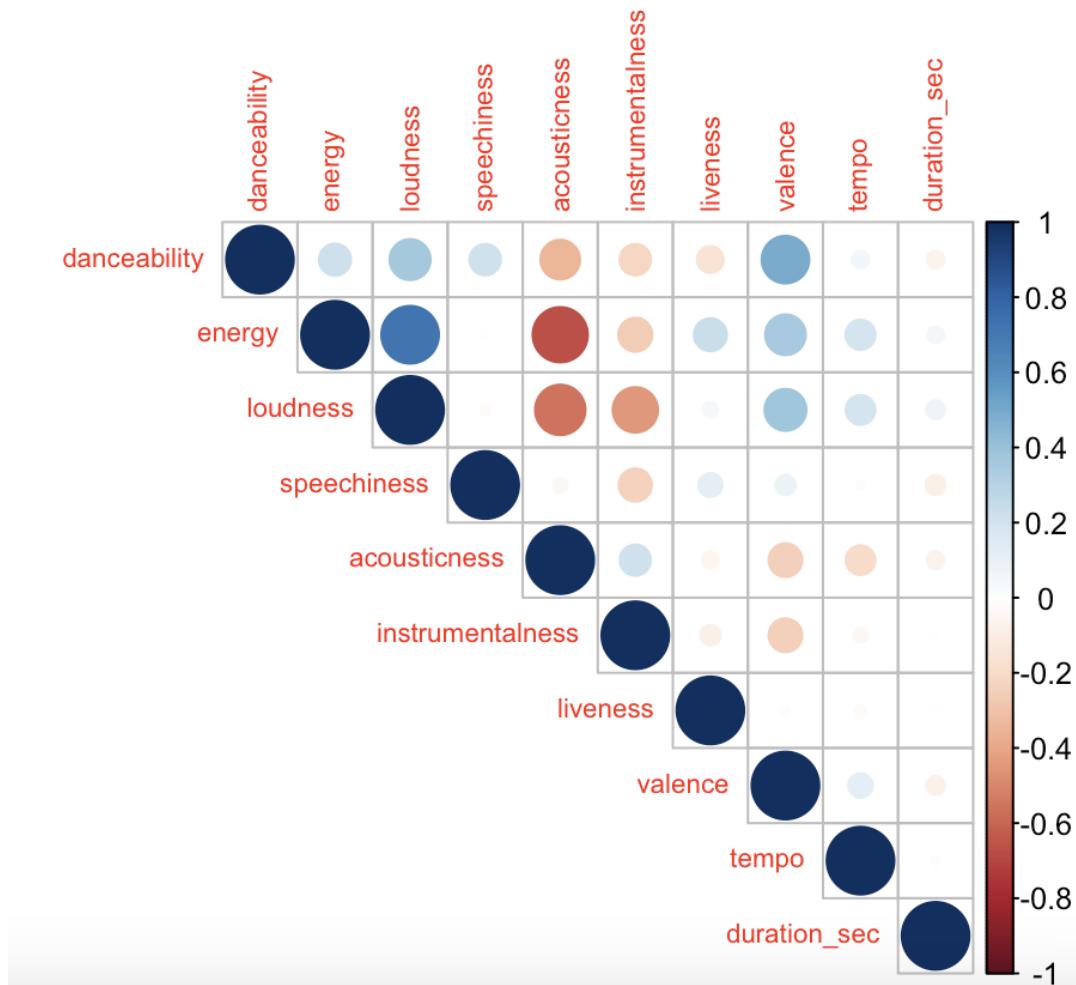
# Añadir la variable de respuesta 'track_popularity' al dataframe original
datos4 <- datos3 %>% select(-track_popularity)
```

**2. Para continuar con nuestro modelo de regresión, es necesario analizar los posibles problemas de multicolinealidad, los cuales, de existir, nos obligarían a emplear otro tipo de modelo de regresión.**

Para ello, examinaremos dos aspectos: la matriz de correlaciones y los coeficientes VIF, que nos permiten confirmar la presencia de dicho problema.

```
# Calcular la matriz de correlación
corr_matrix <- cor(numeric_vars)

# Graficar la matriz de correlación
corrplot(corr_matrix, method = "circle", type = "upper", tl.cex = 0.7)
```



A simple vista, puede parecer que existen variables, como energy y loudness, entre otras, que complican el análisis. Sin embargo, para poder confirmar esta afirmación, calcularemos el coeficiente VIF. Para ello, utilizaremos la librería "car" de R y ejecutaremos los siguientes comandos:

```
# Ajustar un modelo de regresión lineal con todas las variables explicativas
model <- lm(track_popularity ~ ., data = datos4)
|
# Calcular el VIF para cada variable
library(car)
vif_values <- vif(model)
print(vif_values)
```

danceability	energy	loudness	speechiness	acousticness	instrumentalness
1.678289	3.353681	2.915416	1.171231	2.072487	1.362604
liveness	valence	tempo			
1.173849	1.523980	1.065574			

Una vez obtenidos estos valores, los interpretamos de la siguiente manera:

1. VIF < 5: No hay correlación significativa entre la variable explicativa y cualquier otro predictor.
  2. 5 < VIF < 10: Correlación moderada. Puede ser aceptable, pero debe ser monitorizada.
  3. VIF > 10: Alta correlación. Existe una preocupación significativa por la multicolinealidad.

Como se puede observar, hemos evitado el problema más grave que puede afectar a una regresión lineal, que es la multicolinealidad, la cual no está presente en este modelo propuesto. Una vez hecho esto, procedemos al siguiente paso, que es proponer un modelo de regresión lineal adecuado.

**3. Proponemos un modelo de regresión lineal inicial y, en función de su variabilidad y de la contribución de cada variable explicativa, determinamos si es necesario realizar algún ajuste**

El modelo que se propondrá es el siguiente:

$$\hat{Y} = b_0 + b_1 * \text{danceability} + b_2 * \text{energy} + b_3 * \text{loudness} + b_4 * \text{speechiness} + b_5 * \text{acousticness} + b_6 * \text{instrumentalness} + b_7 * \text{liveness} + b_8 * \text{valence} + b_9 * \text{tempo}$$

Y los resultados son los siguientes:

```

> summary(final_model0)

Call:
lm(formula = track_popularity ~ danceability + energy + loudness +
    speechiness + acousticness + instrumentalness + liveness +
    valence + tempo, data = datos4)

Residuals:
    Min      1Q  Median      3Q     Max 
-18.902 -10.339 - 5.453   4.843  86.823 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 16.617939  0.218917 75.910 < 2e-16 ***
danceability 0.879458  0.167993  5.235 1.65e-07 ***
energy        2.468706  0.171538 14.392 < 2e-16 ***
loudness      0.374580  0.006452 58.054 < 2e-16 ***
speechiness   -3.868253  0.164405 -23.529 < 2e-16 ***
acousticness   0.768378  0.099210  7.745 9.58e-15 ***
instrumentalness -5.774207  0.076164 -75.813 < 2e-16 ***
liveness       0.090690  0.153164  0.592  0.55378  
valence        -4.272198  0.114688 -37.251 < 2e-16 ***
tempo          -0.002656  0.000837 -3.173  0.00151 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

Residual standard error: 15.14 on 374322 degrees of freedom  
 Multiple R-squared: 0.06282, Adjusted R-squared: 0.0628  
 F-statistic: 2788 on 9 and 374322 DF, p-value: < 2.2e-16

Como se puede observar, tenemos una variable explicativa, liveness, que no es significativa en nuestro modelo. Por ende, será necesario eliminarla para mejorar el ajuste de nuestro modelo. Sin embargo, no procederemos a hacerlo en esta etapa, dado que la variabilidad explicada por este modelo es muy baja, lo que hace que la predicción sea poco fiable. En consecuencia, no vale la pena analizar la heterocedasticidad, la normalidad de las variables ni los problemas de autocorrelación, ya que faltan muchas variables. Para mejorar el modelo, añadiremos nuevas variables que nos ayudarán a realizar una regresión más precisa del modelo de popularidad de la canción de un artista.

#### **4. Proponemos un nuevo modelo que incluye más variables, las cuales contribuyen a la regresión lineal de la popularidad de una canción de un artista.**

El modelo que propondremos ahora es el siguiente:

$$\hat{Y} = b_0 + b_1 * \text{danceability} + b_2 * \text{energy} + b_3 * \text{loudness} + b_4 * \text{speechiness} + b_5 * \text{acousticness} + b_6 * \text{instrumentalness} + b_7 * \text{liveness} + b_8 * \text{valence} + b_9 * \text{tempo} + b_{10} * \text{duration\_sec} + b_{11} * \text{followers} + b_{12} * \text{album\_popularity} + b_{13} * \text{release\_year} + b_{14} * \text{artist\_popularity}$$

Para incluir la variable ‘release\_year’, hemos decidido tratarla como una variable numérica con el objetivo de evitar la creación de numerosas variables ficticias que podrían complicar la compilación e interpretación del modelo. Las operaciones necesarias en R para realizar este paso son las siguientes:

```

Call:
lm(formula = track_popularity ~ danceability + energy + loudness +
    speechiness + acousticness + instrumentalness + liveness +
    valence + tempo + followers + album_popularity + artist_popularity +
    release_year, data = datos4)

Residuals:
    Min      1Q  Median      3Q     Max 
-55.405 -2.965  0.024  2.206 80.539 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.055e+02  2.199e+00 -93.415 < 2e-16 ***
danceability 3.213e+00  8.275e-02  38.826 < 2e-16 ***
energy        9.617e-01  8.369e-02  11.491 < 2e-16 ***
loudness      6.120e-02  3.164e-03  19.345 < 2e-16 ***
speechiness   -3.196e+00  8.056e-02 -39.669 < 2e-16 ***
acousticness  -5.957e-01  4.842e-02 -12.304 < 2e-16 ***
instrumentalness -7.554e-01  3.759e-02 -20.097 < 2e-16 ***
liveness      -1.255e+00  7.475e-02 -16.783 < 2e-16 ***
valence       -1.960e+00  5.702e-02 -34.379 < 2e-16 ***
tempo          2.228e-03  4.077e-04  5.464 4.66e-08 ***
followers     1.099e-07  1.463e-09  75.077 < 2e-16 ***
album_popularity 6.339e-01  9.246e-04  685.622 < 2e-16 ***
artist_popularity 1.684e-02  7.463e-04  22.567 < 2e-16 ***
release_year   1.014e-01  1.092e-03  92.845 < 2e-16 ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.371 on 374318 degrees of freedom
Multiple R-squared:  0.7779,    Adjusted R-squared:  0.7779 
F-statistic: 1.008e+05 on 13 and 374318 DF,  p-value: < 2.2e-16

```

En este modelo observamos una mejoría significativa en la variabilidad explicada de la popularidad de la canción de un artista. Sin embargo, esta mejoría podría ser engañosa debido a posibles problemas de heterocedasticidad, normalidad, multicolinealidad y autocorrelación que pueden estar presentes en el modelo. Analizaremos estos aspectos uno por uno y, en caso de detectar problemas, tomaremos las medidas necesarias para corregirlos.

1) **Autocorrelación** → para ver la existencia de la autocorrelación en este modelo, utilizaremos la prueba de Durbin-Watson. Antes de ver los resultados en R, veamos las posibles interpretaciones que tiene:

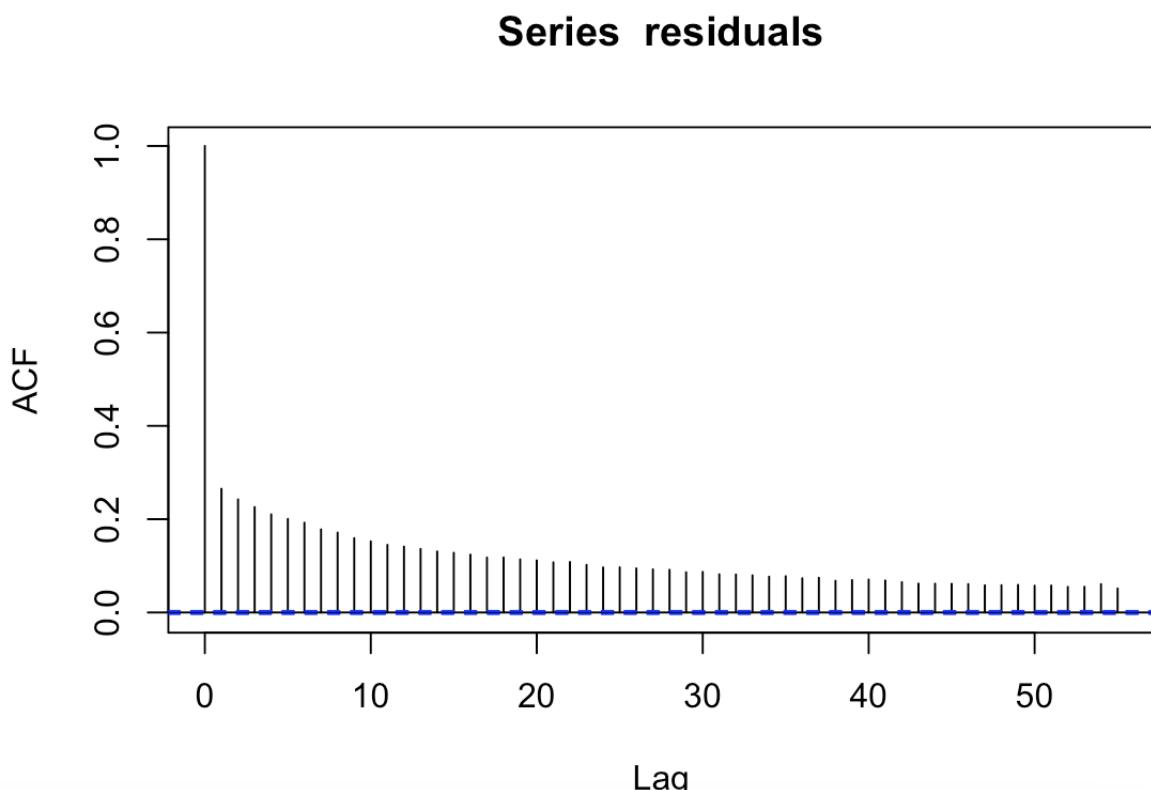
- Si el estadístico de Durbin-Watson se acerca a 2, sugiere que no hay autocorrelación de primer orden en los residuos del modelo.
- Valores cercanos a 0 indican autocorrelación positiva.
- Valores cercanos a 4 indican autocorrelación negativa.
- El valor-p asociado proporciona la significancia estadística de la prueba.

Una vez sepamos esto, miramos en R si existe o no.

```
# Realizar la prueba de Durbin-Watson en el modelo con datos muestreados
durbinWatsonTest(final_model1)

> durbinWatsonTest(final_model1)
lag Autocorrelation D-W Statistic p-value
 1   -0.0006148822    2.001016   0.952
Alternative hypothesis: rho != 0

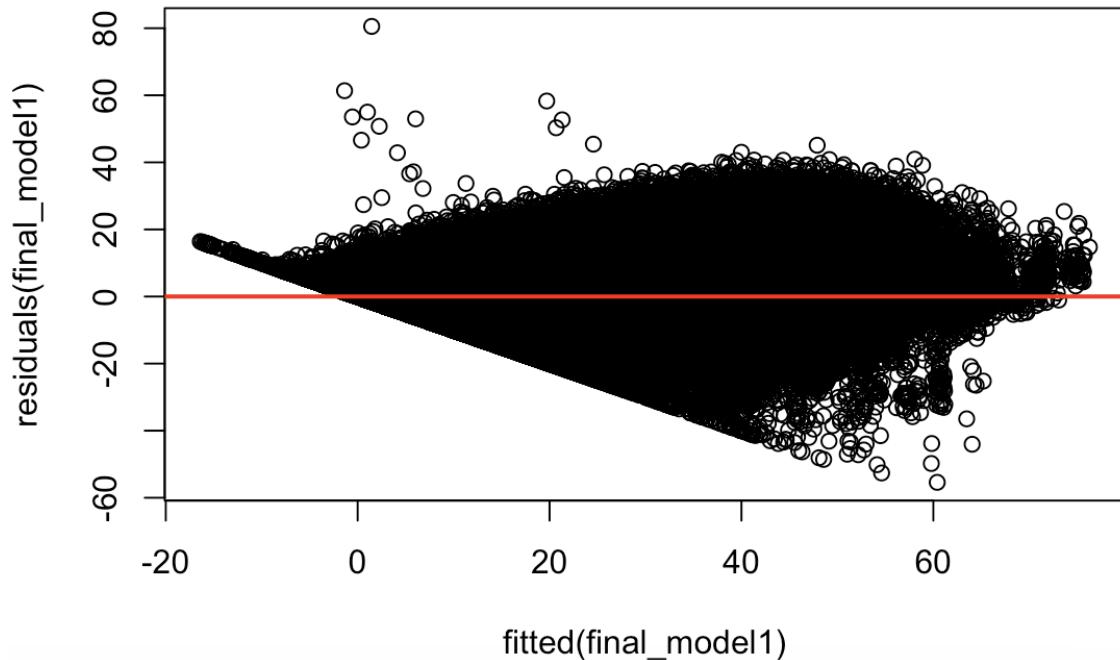
residuals <- residuals(final_model1)
acf(residuals)
```



A través del gráfico y la prueba de Estadística de Durbin-Watson (D-W), podemos concluir que no hay evidencia de autocorrelación en nuestro modelo. Este hallazgo es un aspecto positivo para nuestra regresión.

**2) Heterocedasticidad** → para comprobar la existencia de la heterocedasticidad utilizaremos dos herramientas: gráfica de residuos y la prueba de Breusch-Pagan.

```
heterocedasticidad <- plot(fitted(final_model1), residuals(final_model1))
abline(h = 0, col = "red", lwd = 2)
```



Este gráfico proporciona una clara evidencia de la existencia de heterocedasticidad severa. Sin embargo, como medida adicional, vamos a confirmarlo mediante la prueba de Breusch-Pagan.

```
# Instalar y cargar el paquete lmtest
library(lmtest)

# Realizar la prueba de Breusch-Pagan
breusch_pagan_test <- bptest(final_model1, studentize = FALSE)
# Imprimir los resultados
print(breusch_pagan_test)
```

```
Breusch-Pagan test

data: final_model1
BP = 245176, df = 13, p-value < 2.2e-16
```

Con este resultado, podemos apreciar la gravedad del problema de heterocedasticidad en nuestro modelo, lo cual indica claramente la necesidad de realizar ajustes para mejorar su desempeño.

3) **Multicolinealidad** → tal y como se hizo anteriormente, analizaremos el problema de multicolinealidad mediante coeficiente VIF

```
vif_values <- vif(final_model1)
print(vif_values)

print(vif_values)
danceability      energy       loudness      speechiness    acousticness instrumentalness
1.717865          3.368179     2.956965     1.186591      2.082700      1.400059
liveness          valence      tempo         followers      album_popularity artist_popularity
1.179711          1.589082     1.066614     1.372478      2.443675      2.644500
release_year
1.147879
```

Como se observa en este resultado, no se evidencia el problema de multicolinealidad en este modelo. Esta ausencia, aunque dentro de un contexto complicado, es un indicador positivo, ya que de haber multicolinealidad, se requeriría un cambio completo en el modelo.

4) **Normalidad/asimetría de la variable explicada** → Analizaremos en este paso la simetría de nuestra variable explicada en el modelo. Esta evaluación se realiza para determinar la eficacia del modelo, ya que una alta asimetría en la variable explicada puede resultar en predicciones con errores más significativos. En esta ocasión, utilizaremos el coeficiente de asimetría de la distribución, cuya interpretación es la siguiente:

- Skewness  $\approx 0$ : La distribución es aproximadamente simétrica. Esto significa que los datos están distribuidos de manera bastante uniforme alrededor de la media.
- Skewness  $> 0$ : La distribución tiene una asimetría positiva (sesgo a la derecha). Esto significa que hay una cola más larga o más pesada en el lado derecho de la distribución. En otras palabras, los valores a la derecha de la media son más dispersos.
- Skewness  $< 0$ : La distribución tiene una asimetría negativa (sesgo a la izquierda). Esto significa que hay una cola más larga o más pesada en el lado izquierdo de la distribución. En otras palabras, los valores a la izquierda de la media son más dispersos.

```

# Calcular el coeficiente de asimetría de la variable "track_popularity"
skewness_track_popularity <- skewness(datos4$track_popularity)

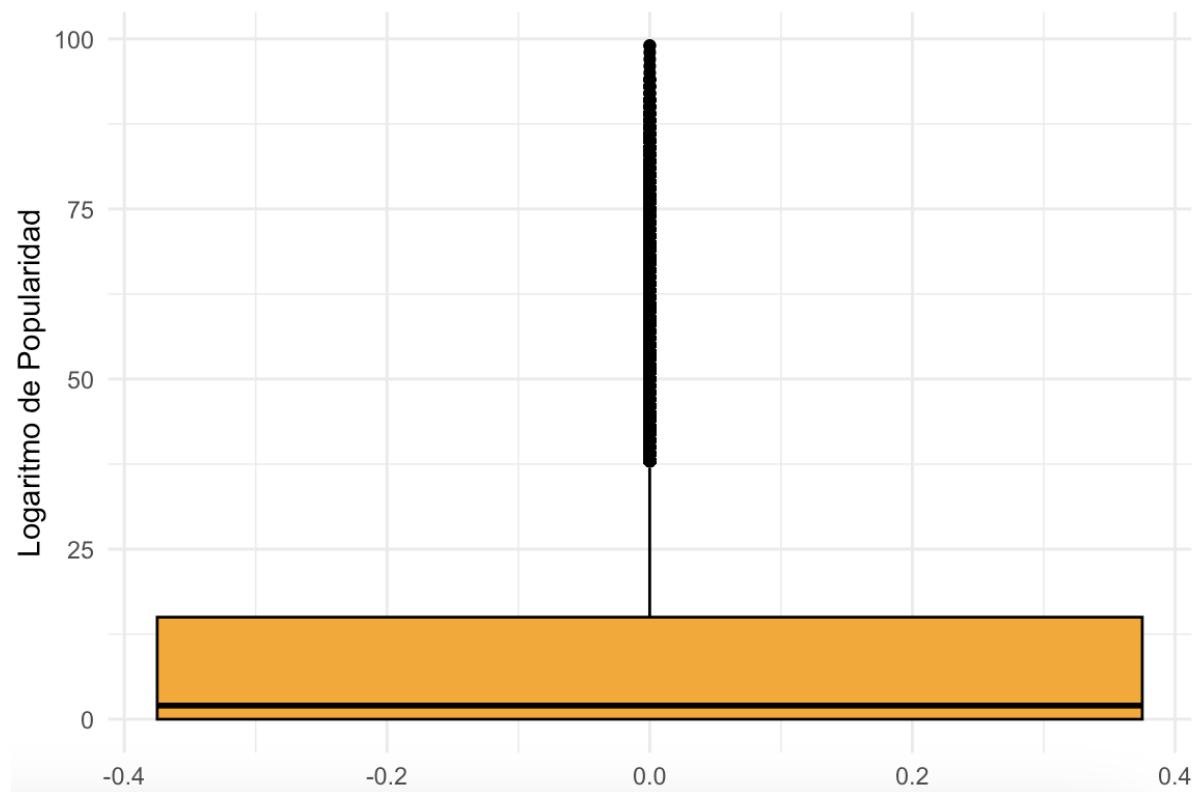
# Imprimir el coeficiente de asimetría
print(skewness_track_popularity)

> print(skewness_track_popularity)
[1] 1.895574

```

Como se puede observar, nuestra variable explicada exhibe una alta asimetría, lo cual constituye un indicador desfavorable para las predicciones de nuestro modelo. Una evidencia adicional que respalda esta observación es el gráfico de Box and Whisker.

Diagrama de Caja y Bigotes de Popularidad de Pistas



Como conclusión, este modelo propuesto presenta problemas de heterocedasticidad y asimetría. Por lo tanto, para mejorarlo, ajustaremos al máximo nuestro nuevo modelo con el fin de evitar estos inconvenientes o, al menos, intentar reducirlos en la medida de lo posible. Para ello, procedemos al siguiente paso:

**5. Proponemos el modelo definitivo que aborda los errores del modelo anterior, buscando corregir la asimetría en la variable explicada y reducir la heterocedasticidad en la medida de lo posible.**

Al tener una variable dependiente, es decir, una variable explicada muy asimétrica y difícil de transformar, aplicamos una transformación doble o incluso triple para ajustar lo mejor posible la distribución de nuestra variable (la fórmula está dentro de la variable `log_track_popularity`) y mejorar la predicción del modelo. También ajustamos una variable explicativa de 'followers', que presenta numerosos valores atípicos. El modelo definitivo que proponemos es el siguiente:

$$\hat{Y} = b_0 + b_1 * \text{danceability} + b_2 * \text{energy} + b_3 * \text{loudness} + b_4 * \text{speechiness} + b_5 * \text{acousticness} + b_6 * \text{instrumentalness} + b_7 * \text{liveness} + b_8 * \text{valence} + b_9 * \text{tempo} + b_{10} * \text{log(followers + 1)} + b_{11} * \text{album\_popularity} + b_{12} * \text{artist\_popularity} + b_{13} * \text{release\_year} + b_{14} * \text{duration\_sec}$$

Una vez propuesto este modelo, lo metemos en R para sacar sus características.

```
datos4$log_track_popularity <- sqrt(log(datos4$track_popularity+1)+10)
final_model <- lm(log_track_popularity ~ danceability + energy + loudness + speechiness +
    acousticness + instrumentalness + liveness + valence +
    tempo + duration_sec+ log(followers+1) +
    album_popularity + artist_popularity + release_year,
    data = datos4)

summary(final_model)
```

```

> summary(final_model)

Call:
lm(formula = log_track_popularity ~ danceability + energy + loudness +
    speechiness + acousticness + instrumentalness + liveness +
    valence + tempo + duration_sec + log(followers + 1) + album_popularity +
    artist_popularity + release_year, data = datos4)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.69942 -0.07788 -0.03261  0.07707  0.54827 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 8.482e-01 3.390e-02 25.022 <2e-16 ***
danceability 2.056e-02 1.250e-03 16.443 <2e-16 ***
energy       1.728e-02 1.264e-03 13.673 <2e-16 ***
loudness     1.730e-03 4.829e-05 35.830 <2e-16 ***
speechiness -2.450e-02 1.219e-03 -20.092 <2e-16 ***
acousticness 1.329e-02 7.320e-04 18.150 <2e-16 ***
instrumentalness -1.149e-02 5.702e-04 -20.155 <2e-16 ***
liveness     -1.836e-02 1.129e-03 -16.264 <2e-16 ***
valence      -1.224e-02 8.641e-04 -14.160 <2e-16 ***
tempo        6.033e-05 6.155e-06 9.801 <2e-16 ***
duration_sec 2.629e-05 1.602e-06 16.415 <2e-16 ***
log(followers + 1) 1.962e-03 1.237e-04 15.860 <2e-16 ***
album_popularity 7.577e-03 1.391e-05 544.702 <2e-16 ***
artist_popularity 7.472e-04 2.101e-05 35.568 <2e-16 ***
release_year   1.170e-03 1.675e-05 69.862 <2e-16 ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1113 on 374317 degrees of freedom
Multiple R-squared:  0.7176,    Adjusted R-squared:  0.7176 
F-statistic: 6.796e+04 on 14 and 374317 DF,  p-value: < 2.2e-16

```

Como se puede observar, la variabilidad explicada de este nuevo modelo sigue siendo alta, alcanzando un 71,76%, mientras que el intervalo de residuos ha disminuido significativamente. Ahora procederemos a verificar si persisten los problemas identificados anteriormente y si ha habido alguna mejora en ellos.

1) **Autocorrelación** → para ver la existencia de la autocorrelación en este modelo, utilizaremos la prueba de Durbin-Watson. Antes de ver los resultados en R, veamos las posibles interpretaciones que tiene:

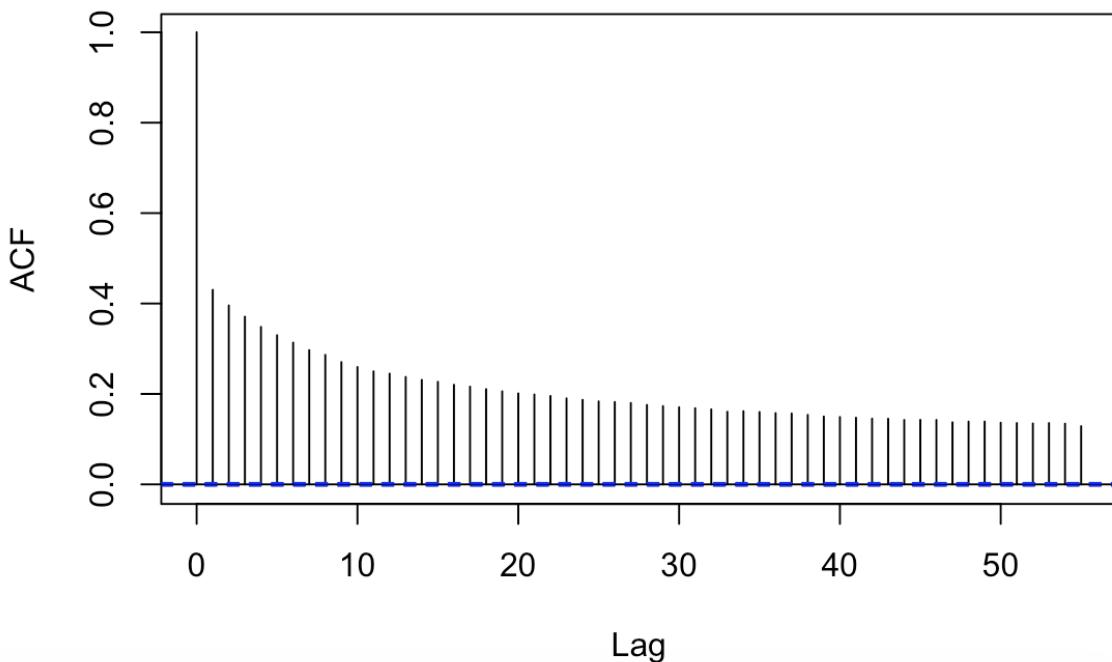
- Si la estadística de Durbin-Watson se acerca a 2, sugiere que no hay autocorrelación de primer orden en los residuos del modelo.
- Valores cercanos a 0 indican autocorrelación positiva.
- Valores cercanos a 4 indican autocorrelación negativa.
- El valor-p asociado proporciona la significancia estadística de la prueba.

Una vez sepamos esto, miramos en R si existe o no.

```
# Realizar la prueba de Durbin-Watson en el modelo con datos muestreados
durbinWatsonTest(final_model)

residuals <- residuals(final_model)
acf(residuals)
```

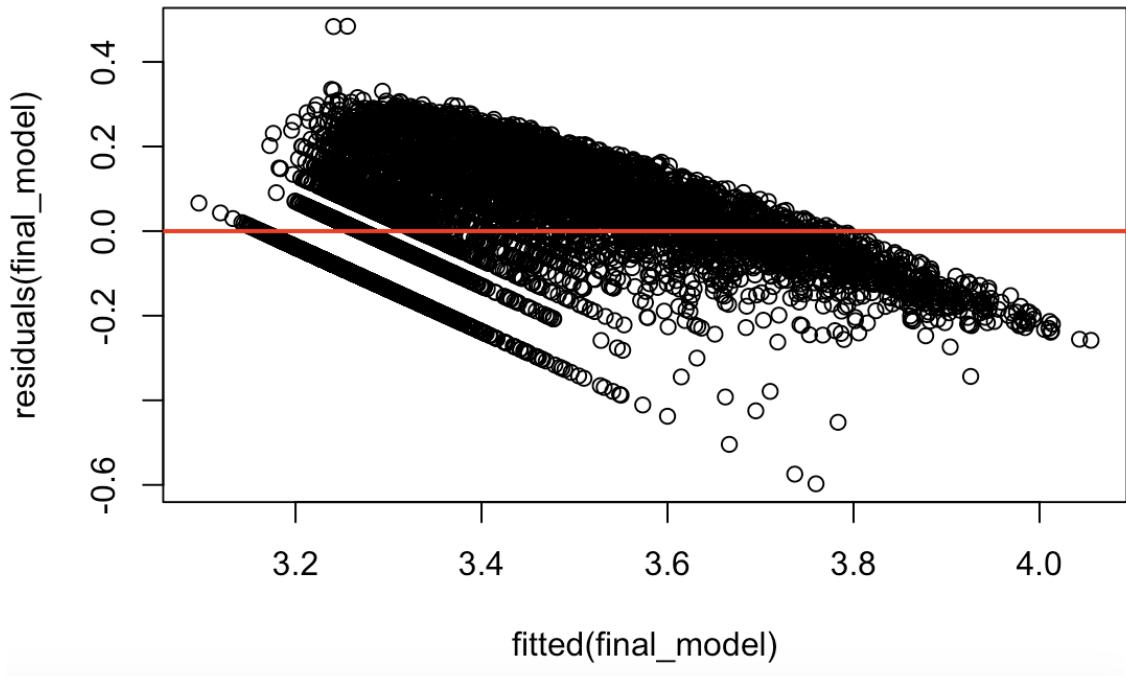
## Series residuals



Mediante el gráfico y la prueba de la Estadística de Durbin-Watson (D-W), podemos confirmar que no hay evidencia de autocorrelación en nuestro modelo definitivo. Este resultado es un punto positivo para la robustez de nuestro modelo.

- 2) **Heterocedasticidad** → para comprobar la existencia de la heterocedasticidad utilizaremos dos herramientas: gráfica de residuos y la prueba de Breusch-Pagan.

```
heterocedasticidad <- plot(fitted(final_model), residuals(final_model))
abline(h = 0, col = "red", lwd = 2)
```



Con este gráfico, podemos observar una clara evidencia de la existencia de heterocedasticidad en nuestro modelo. Sin embargo, es importante destacar que es mucho menos grave que en el modelo anterior, lo cual indica una mejora en este nuevo modelo. No obstante, como medida adicional, procederemos a verificarlo mediante la prueba de Breusch-Pagan.

```
# Instalar y cargar el paquete lmtest
library(lmtest)

# Realizar la prueba de Breusch-Pagan
breusch_pagan_test <- bptest(final_model, studentize = FALSE)
# Imprimir los resultados
print(breusch_pagan_test)
```

Breusch-Pagan test

```
data: final_model
BP = 5909.1, df = 14, p-value < 2.2e-16
```

Con este resultado, podemos observar una mejora significativa en el problema de heterocedasticidad que existía en nuestro modelo anterior. Esta mejora es un indicador claro de que el ajuste ha mejorado la eficiencia de la predicción del modelo, ya que hemos pasado de un valor de BP = 245176 a BP = 5909, lo cual representa una mejora significativa. Sin

embargo, aún persiste el problema de heterocedasticidad, el cual será abordado de manera más efectiva en futuras iteraciones.

- 3) **Multicolinealidad** → tal y como se hizo anteriormente, analizaremos el problema de multicolinealidad mediante coeficiente VIF.

```
vif_values <- vif(final_model)
print(vif_values)

> print(vif_values)
  danceability      energy      loudness      speechiness      acousticness
  1.721162        3.368291    3.023008      1.192433        2.088514
instrumentalness   liveness      valence       tempo
  1.413406        1.180191    1.601135      1.066633        1.054981
log(followers + 1) album_popularity artist_popularity release_year
  8.332018        2.426511     9.192382      1.184695
```

Tal como se observa en este resultado, no se evidencia el problema de multicolinealidad en este modelo, lo cual sigue siendo un indicador positivo. Sin embargo, es importante destacar que los coeficientes se han incrementado y han estado cerca de rozar el límite(\* → para ver). Esto requiere una revisión adicional para interpretar correctamente los resultados.

\*VIF < 5: No hay correlación entre la variable explicativa y cualquier otro predictor.  
\*5 < VIF < 10: Correlación moderada. Puede ser aceptable, pero debe ser monitorizada.  
\*VIF > 10: Alta correlación. Existe una preocupación significativa por la multicolinealidad.

- 4) **Normalidad/asimetría de la variable explicada** → Analizaremos en este paso la simetría de nuestra variable explicada en el modelo. Esta evaluación se realiza para determinar la eficacia del modelo, ya que en caso de una asimetría pronunciada en la variable explicada, las predicciones pueden contener errores más significativos. Las interpretaciones son las mismas que se mencionaron anteriormente (cuanto más cercano al valor 0, mejor).

```
# Calcular el coeficiente de asimetría de la variable "track_popularity"
skewness_track_popularity <- skewness(datos4$log_track_popularity)

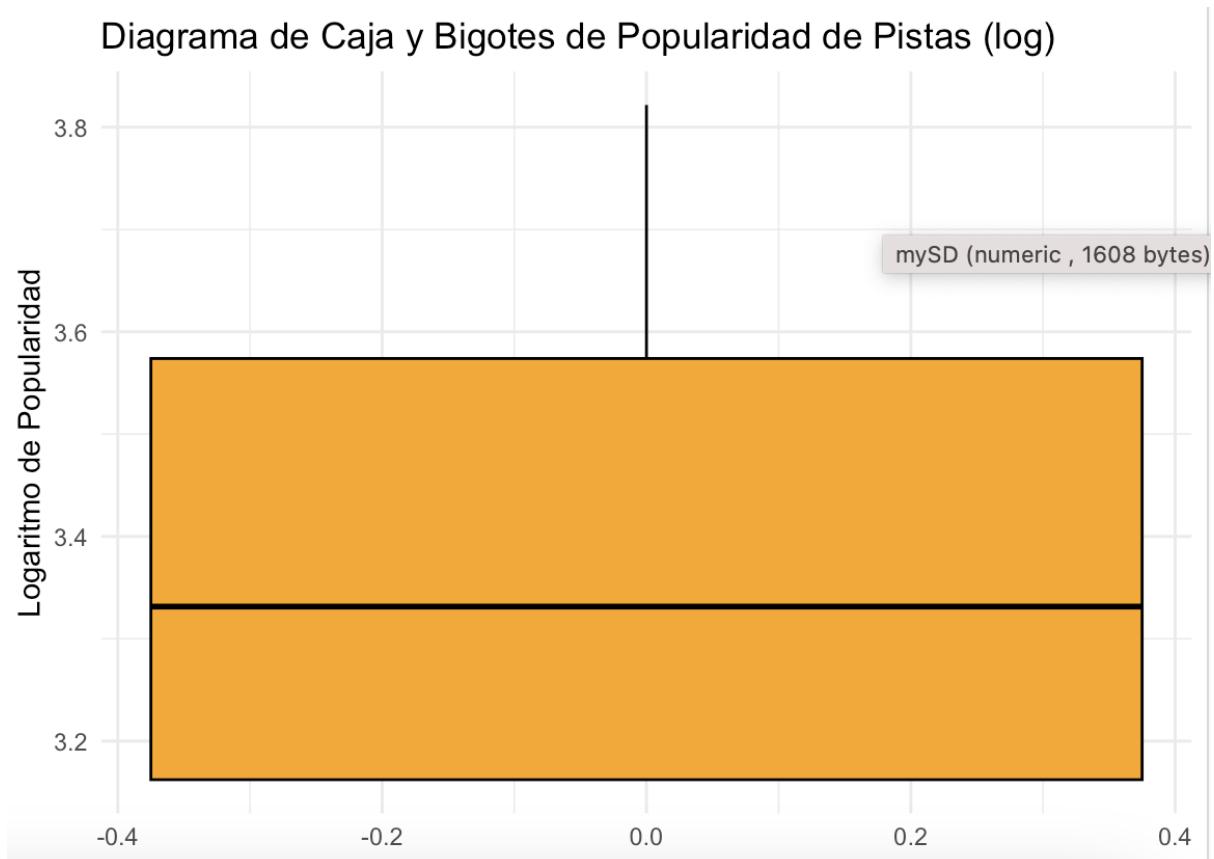
# Imprimir el coeficiente de asimetría
print(skewness_track_popularity)

> print(skewness_track_popularity)
[1] 0.349777
```

Como se puede observar, nuestra variable explicada presenta simetría, lo cual es un indicador positivo para las predicciones de nuestro modelo definitivo y sugiere que hemos logrado

resolver uno de los problemas del modelo anterior. Esta observación se respalda además con el análisis del gráfico de Box and Whisker.

Diagrama de Caja y Bigotes de Popularidad de Pistas (log)



Como conclusión, este modelo propuesto aún presenta problemas de heterocedasticidad, aunque con una mejora significativa en comparación con el modelo anterior. Sin embargo, hemos logrado resolver el problema de la asimetría mediante las transformaciones adecuadas. Tras completar todos estos pasos, procedemos al paso definitivo: extraer conclusiones que nos ayudarán a maximizar la popularidad de una canción.

## 6. Conclusiones

```

> summary(final_model)

Call:
lm(formula = log_track_popularity ~ danceability + energy + loudness +
    speechiness + acousticness + instrumentalness + liveness +
    valence + tempo + duration_sec + log(followers + 1) + album_popularity +
    artist_popularity + release_year, data = datos4)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.69942 -0.07788 -0.03261  0.07707  0.54827 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 8.482e-01 3.390e-02 25.022 <2e-16 ***
danceability 2.056e-02 1.250e-03 16.443 <2e-16 ***
energy       1.728e-02 1.264e-03 13.673 <2e-16 ***
loudness     1.730e-03 4.829e-05 35.830 <2e-16 ***
speechiness -2.450e-02 1.219e-03 -20.092 <2e-16 ***
acousticness 1.329e-02 7.320e-04 18.150 <2e-16 ***
instrumentalness -1.149e-02 5.702e-04 -20.155 <2e-16 ***
liveness     -1.836e-02 1.129e-03 -16.264 <2e-16 ***
valence      -1.224e-02 8.641e-04 -14.160 <2e-16 ***
tempo        6.033e-05 6.155e-06 9.801 <2e-16 ***
duration_sec 2.629e-05 1.602e-06 16.415 <2e-16 ***
log(followers + 1) 1.962e-03 1.237e-04 15.860 <2e-16 ***
album_popularity 7.577e-03 1.391e-05 544.702 <2e-16 ***
artist_popularity 7.472e-04 2.101e-05 35.568 <2e-16 ***
release_year   1.170e-03 1.675e-05 69.862 <2e-16 ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1113 on 374317 degrees of freedom
Multiple R-squared:  0.7176,    Adjusted R-squared:  0.7176 
F-statistic: 6.796e+04 on 14 and 374317 DF,  p-value: < 2.2e-16

```

## Variables con Efecto Positivo Significativo

### 1) Intercepto:

Coeficiente: 0.8482

Interpretación: Este es el valor predicho de log\_track\_popularity cuando todas las variables independientes son cero. Sin embargo, en el contexto de este modelo, la interpretación del intercepto puede no ser significativa por sí misma.

### 2) danceability

Coeficiente: 0.02056

Interpretación: Un aumento de una unidad en la danceability se asocia con un aumento de 0.02056 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

### 3) energy

Coeficiente: 0.01728

Interpretación: Un aumento de una unidad en energy se asocia con un aumento de 0.01728 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

#### **4) loudness**

Coeficiente: 0.00173

Interpretación: Un aumento de una unidad en loudness se asocia con un aumento de 0.00173 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

#### **5) acousticness**

Coeficiente: 0.01329

Interpretación: Un aumento de una unidad en acousticness se asocia con un aumento de 0.01329 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

#### **6) tempo**

Coeficiente: 0.00006033

Interpretación: Un aumento de una unidad en tempo se asocia con un aumento de 0.00006033 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

#### **7) duration\_sec**

Coeficiente: 0.00002629

Interpretación: Un aumento de una unidad en duration\_sec se asocia con un aumento de 0.00002629 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

#### **8) log(followers + 1)**

Coeficiente: 0.001962

Interpretación: Un aumento de una unidad en log(followers+1) se asocia con un aumento de 0.001962 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

#### **9) album\_popularity**

Coeficiente: 0.007577

Interpretación: Un aumento de una unidad en album\_popularity se asocia con un aumento de 0.007577 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

#### **10) artist\_popularity**

Coeficiente: 0.0007472

Interpretación: Un aumento de una unidad en artist\_popularity se asocia con un aumento de 0.0007472 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

### **11) release\_year**

Coeficiente: 0.001170

Interpretación: Un aumento de una unidad en release\_year se asocia con un aumento de 0.001170 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

## **Variables con Efecto Negativo Significativo**

### **1) speechiness**

Coeficiente: -0.02450

Interpretación: Un aumento de una unidad en speechiness se asocia con una disminución de 0.02450 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

### **2) instrumentalness**

Coeficiente: -0.01149

Interpretación: Un aumento de una unidad en instrumentalness se asocia con una disminución de 0.01149 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

### **3) liveness**

Coeficiente: -0.01836

Interpretación: Un aumento de una unidad en liveness se asocia con una disminución de 0.01836 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

### **4) valence**

Coeficiente: -0.01224

Interpretación: Un aumento de una unidad en valence se asocia con una disminución de 0.01224 unidades en log\_track\_popularity, manteniendo las demás variables constantes.

El 71.76% de la variabilidad en log\_track\_popularity se explica por las variables independientes incluidas en el modelo. Este es un buen nivel de ajuste.

Adjusted R-squared: 0.7176

Indica que el modelo en su conjunto es estadísticamente significativo, lo que significa que al menos una de las variables independientes tiene un efecto significativo sobre log\_track\_popularity.

## Resumen

- El modelo explica aproximadamente el 71.76% de la variabilidad en log\_track\_popularity, lo que indica un buen ajuste.
- Las variables danceability, energy, loudness, acousticness, tempo, duration\_sec, log(followers + 1), album\_popularity, artist\_popularity y release\_year tienen un efecto positivo sobre log\_track\_popularity. Esto significa que cuanto más alto sea el valor de estas características, es más probable que la popularidad de la canción sea mayor.
- Por otro lado, las variables speechiness, instrumentalness, liveness y valence tienen un efecto negativo sobre log\_track\_popularity. Esto implica que cuanto más alto sea el valor de estas características, es más probable que la popularidad de la canción sea menor.

## ANÁLISIS PLS-ALTERNATIVA/SOLUCIÓN AL MODELO DE REGRESIÓN LINEAL:

Aquí está el código R para obtener las soluciones del modelo de regresión PLS, el cual nos ayuda a resolver problemas de regresión lineal como heterocedasticidad. Las conclusiones y los resultados relevantes obtenidos a partir de este código están interpretados en el documento de proyecto.

```
pls_model <- plsr(track_popularity ~ danceability + energy + loudness + speechiness +
                     acousticness + instrumentalness + liveness + valence + tempo + followers +
                     album_popularity + artist_popularity + release_year,
                     data = datos4, scale = TRUE, validation = "CV")

summary(pls_model)
coef(pls_model)

validationplot(pls_model, val.type = "MSEP")
plot(pls_model)

loadingplot(pls_model, comps = 1:2)
biplot(pls_model, comps = 1:2)
scoreplot(pls_model, comps = 1:2)
plot(pls_model, ncomp = 13, which = "validation", val.type = "R2")
RMSEP(pls_model)
```

## Librerías utilizadas:

- library(corrplot) → utilizada para matrices de correlación
- library(dplyr) → utilizada para regresión
- library(ggplot2) → utilizada para gráficos como box and whisker y otros

- library(car) → utilizada para prueba de multicolinealidad
- library(lmtest) → utilizada para prueba de autocorrelación y heterocedasticidad
- library(e1071) → utilizada para la prueba de asimetría

## 5- IDENTIFICACIÓN DE TENDENCIAS TEMPORALES

La música es un reflejo de las tendencias culturales y sociales de cada época. Este objetivo se centra en analizar la evolución de la popularidad de los géneros musicales a lo largo del tiempo, utilizando las características de las canciones más populares. A través de este estudio, se pretende observar y comprender las tendencias temporales en la industria musical, revelando cómo han cambiado las preferencias del público y qué factores han influido en estos cambios.

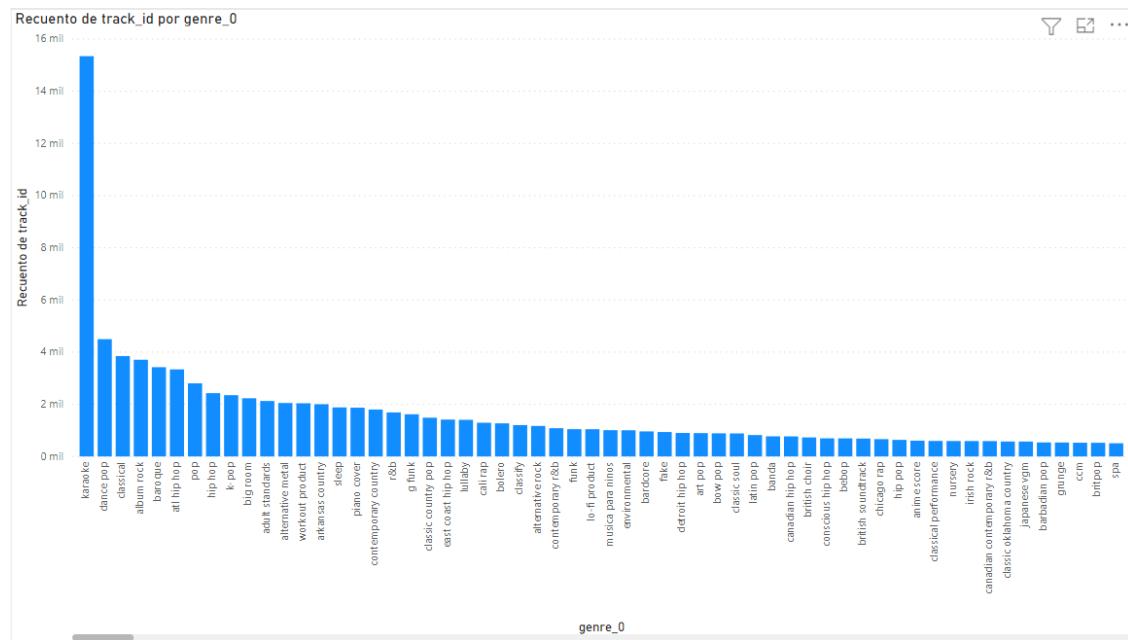
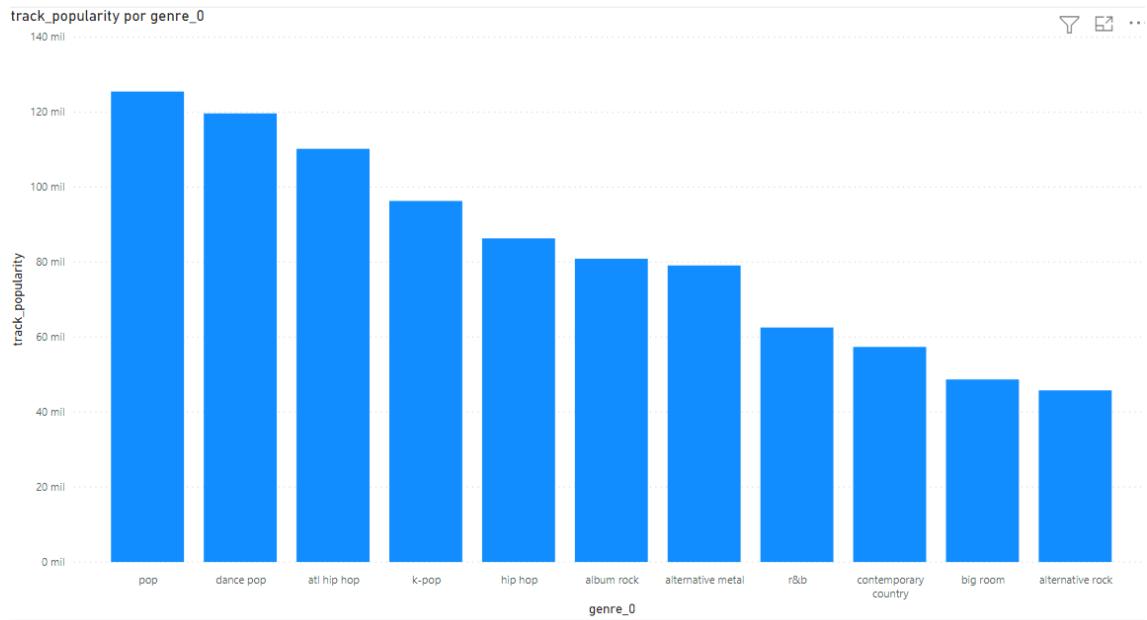
Para lograr este objetivo en nuestro proyecto, hemos decidido emplear la herramienta Power BI para obtener los resultados de nuestro análisis. Esta parte del trabajo estará dividida en tres partes: estudiar la evolución de las características de los géneros más relevantes, comprobar si la época del año influye en el consumo de determinado tipo de música y, por último, analizar la popularidad de los géneros musicales a lo largo del tiempo.

Antes de centrarnos en estos objetivos, es necesario construir previamente en Power Query el esquema que vamos a emplear para realizar los informes y adaptar los datos originales a este esquema para que los resultados sean correctos. El modelo de datos resultante estará compuesto por una tabla de hechos que representa las canciones, y 3 dimensiones alrededor de esta tabla que representan los álbumes, los artistas y las fechas. Este modelo nos

proporcionará la estructura necesaria para llevar a cabo nuestro análisis de manera efectiva.

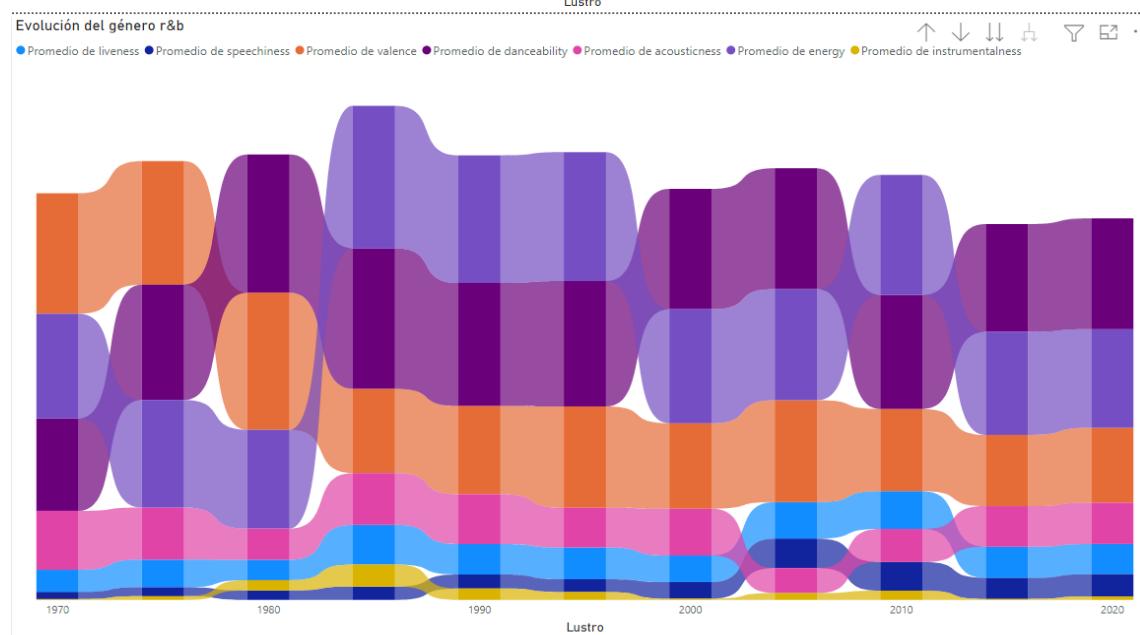
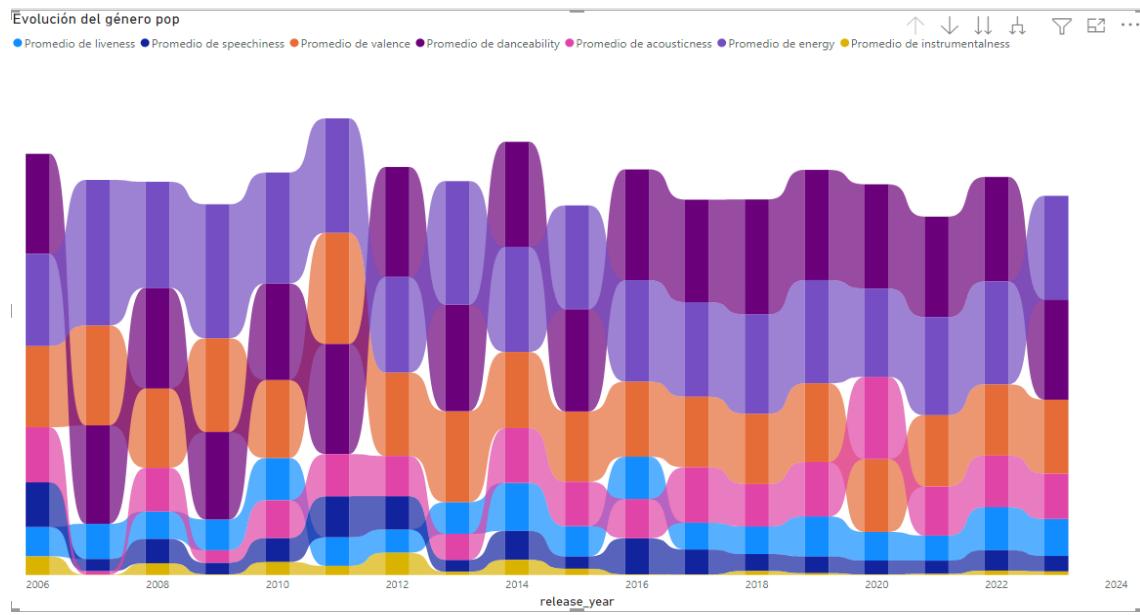


Una vez explicada la construcción del modelo vamos a determinar los géneros que vamos a estudiar. Para ello, nos centraremos en estos dos gráficos.



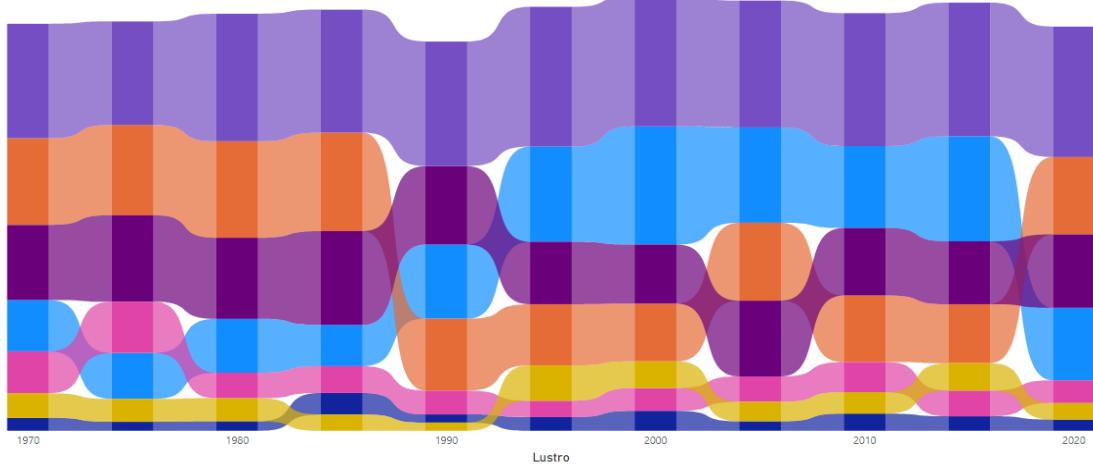
Estos gráficos nos proporcionan información acerca del número de canciones que tiene cada género en la base de datos y la media de la popularidad de todas sus canciones. En base a esto he decidido escoger los siguientes géneros: pop, hip hop, big room, r&b y rock.

El tipo de gráfico que consideramos que más se ajustaba a nuestro objetivo es el gráfico de barra de herramientas. Los siguientes gráficos que vamos a mostrar representan el promedio de cada una de las características de una canción a lo largo de los años.



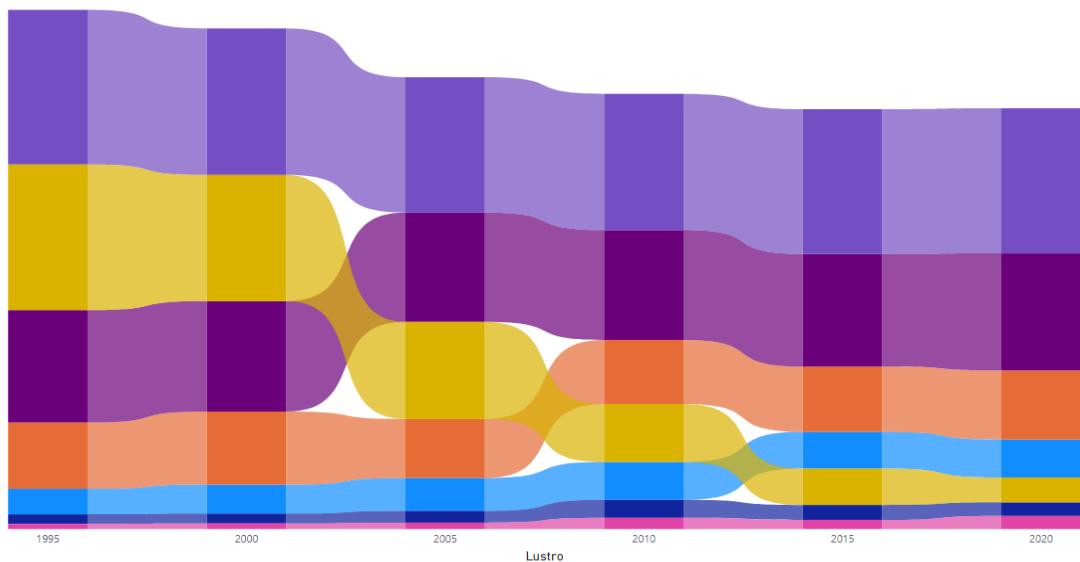
### Evolución del género rock

● Promedio de liveness ● Promedio de speechiness ● Promedio de valence ● Promedio de danceability ● Promedio de acousticness ● Promedio de energy ● Promedio de instrumentalness



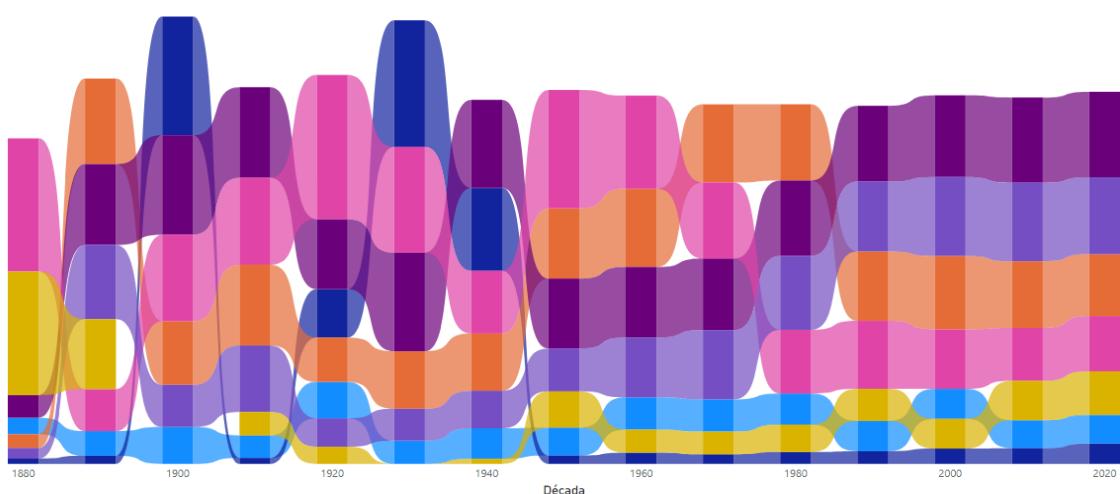
### La evolución del género big room

● Promedio de liveness ● Promedio de speechiness ● Promedio de valence ● Promedio de danceability ● Promedio de acousticness ● Promedio de energy ● Promedio de instrumentalness



### Evolución de las canciones por década

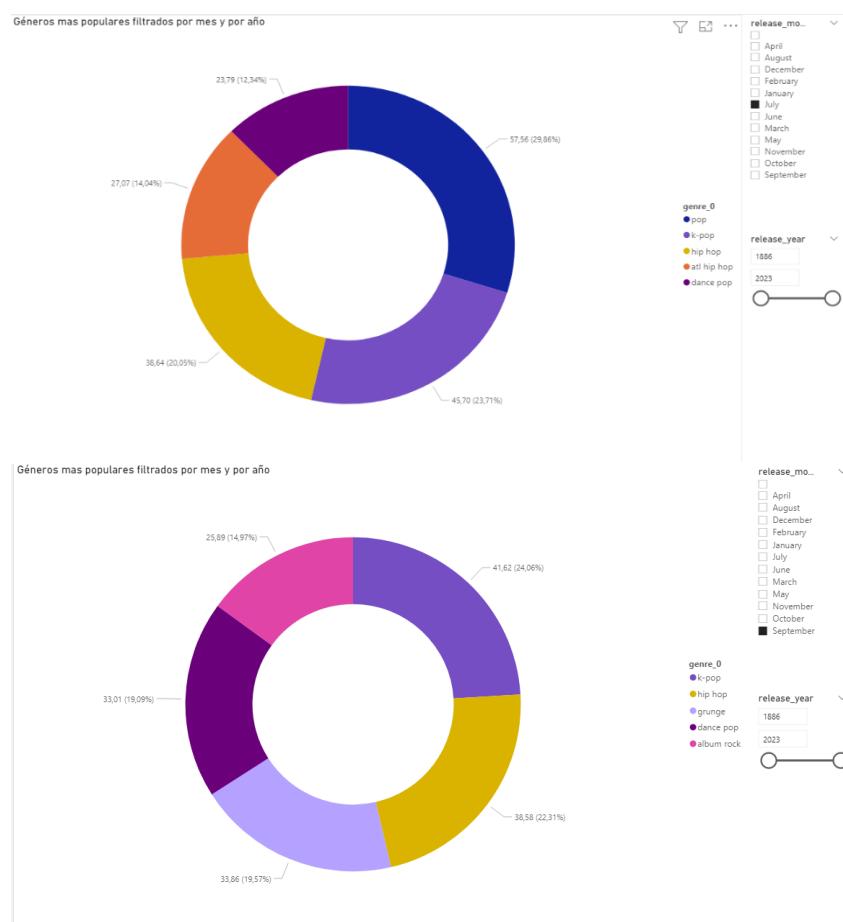
● Promedio de liveness ● Promedio de speechiness ● Promedio de valence ● Promedio de danceability ● Promedio de acousticness ● Promedio de energy ● Promedio de instrumentalness



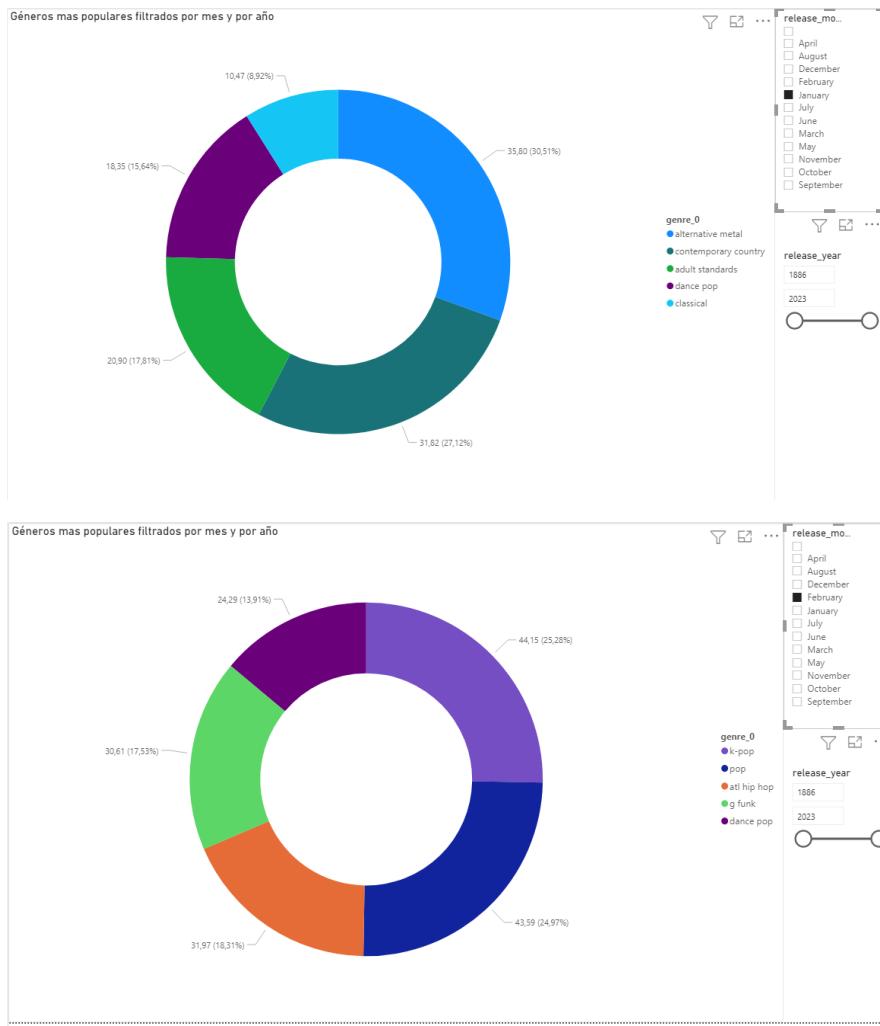
Como conclusiones de estos gráficos podemos extraer que el género pop, hip hop y R&B con el paso de los años tienen más canciones con un estilo "bailable" y menos canciones con un estilo "alegre". Por otro lado, el género rock muestra un aumento en la creación de canciones "en vivo" a medida que pasan los años desde su creación. También es importante destacar que el género big room experimenta un descenso considerable en la variable instrumentalness, lo que sugiere que con el tiempo este género tiene menos canciones puramente instrumentales.

En la última gráfica, representamos la evolución de todas las canciones en conjunto. Aquí podemos observar dos picos de un estilo con una alta presencia de palabras en la canción en las décadas de 1900 y 1930. También notamos un descenso considerable en las canciones acústicas a partir de la década de 1960 y un aumento en las canciones energéticas. A partir de la década de 1990, las canciones parecen estabilizarse siendo la mayor parte canciones "bailables".

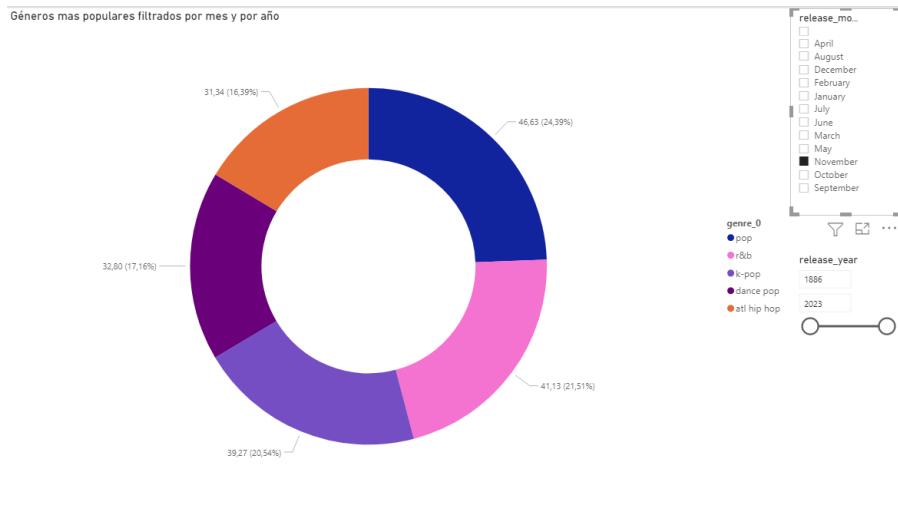
En la segunda parte de este objetivo, analizaremos si la época del año influye en la popularidad de los géneros. Para ello, hemos creado un gráfico de anillos que muestra los géneros más populares con dos segmentadores de tiempo, que segmentan por año de salida y por mes. Verano



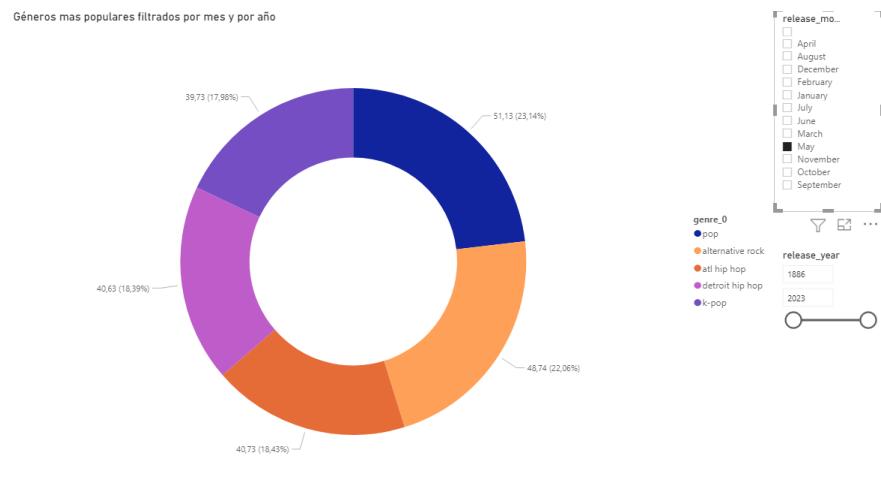
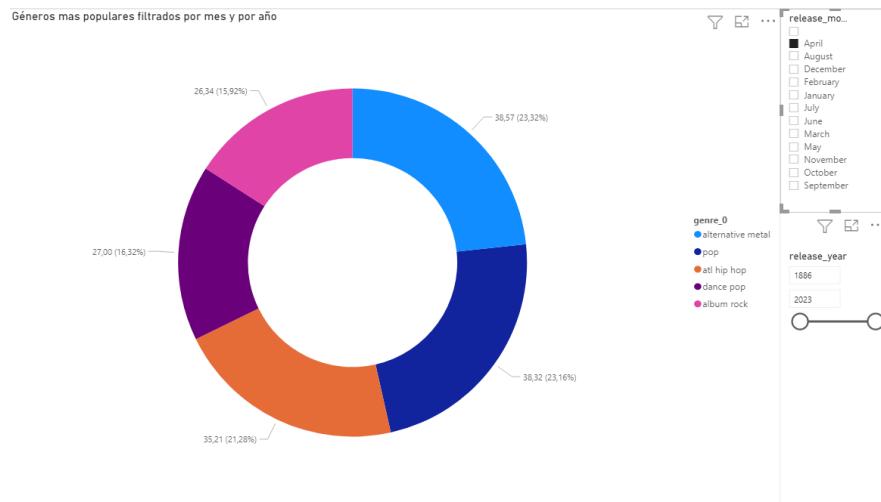
Invierno



## Otoño



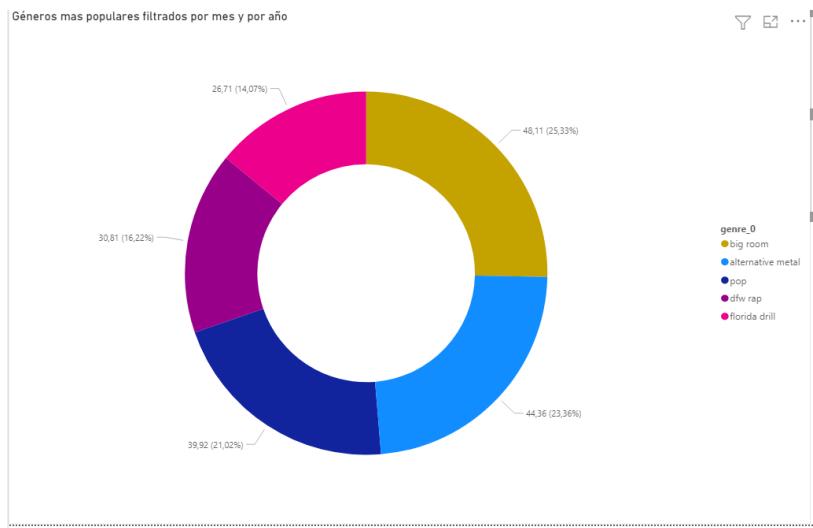
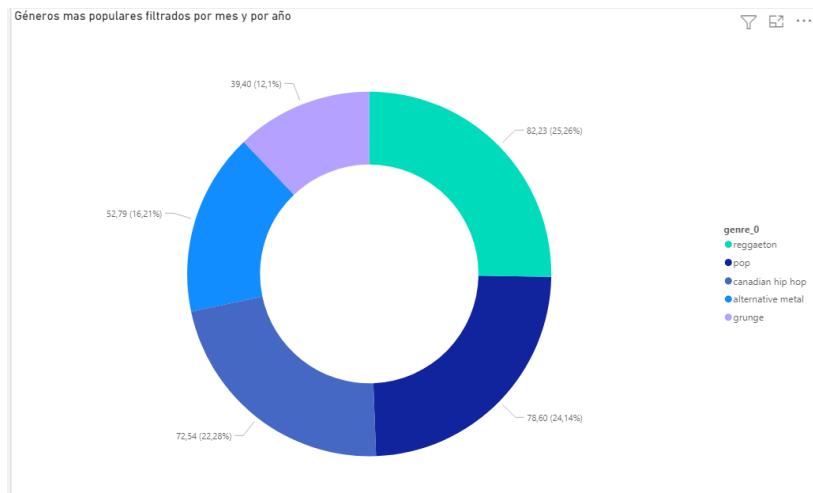
## Primavera



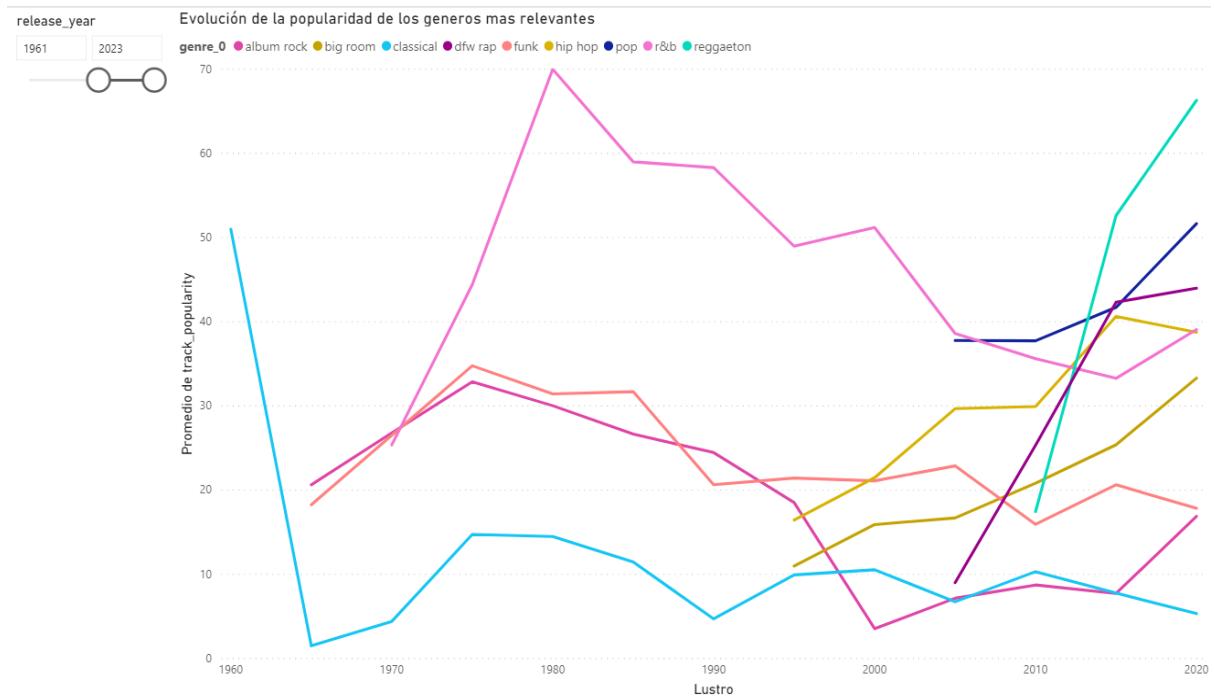
Gracias a estos segmentadores hemos podido comprobar que en general en otoño es más popular el r&b, mientras que en primavera lo es más el metal y el rock. Además, en invierno es más popular el funk y el country, y en verano lo es el k-pop. Es importante destacar que el género pop y hip hop suele ser popular durante todo el año.

Si nos centramos únicamente en los últimos años, la situación cambia. Aparecen en el top géneros como el drill, el rap y el reggaeton y el big room. Es importante destacar que el pop y el hip hop siguen manteniéndose como unos de los géneros más populares.

A continuación, se muestran dos gráficos con ejemplos de estas explicaciones previas.



Por último, tenemos como último propósito ver la evolución de la popularidad de los géneros estudiados. Además de estos géneros estudiados vamos a implementar algunos géneros más recientes para comparar la popularidad entre estos géneros más nuevos con los más antiguos.



En este gráfico de líneas podemos observar el promedio de popularidad de cada género. En el eje x, cada valor está separado por un tiempo de un lustro. De aquí podemos extraer que la música clásica a partir de la década de los 60 e incluso de los 50 deja de ser el máximo exponente en cuanto a la popularidad de las canciones, siendo en la actualidad el género menos popular entre los 9 estudiados.

También se puede apreciar que el r&b alcanza un pico de popularidad promedio de 70 en la década de 1980, siendo el pico más alto del gráfico. A partir de esta década ha ido disminuyendo progresivamente hasta la actualidad. Asimismo, también podemos ver que los géneros más antiguos en este informe, que es el caso del funk y el rock, también decaen con el tiempo, siendo un poco más populares que el género de música clásica.

Por otro lado, los géneros más recientes son los que más popularidad han ganado con los últimos años, teniendo un crecimiento drástico a partir de la década de los 2000. Este es el caso del género el big room, hip hop, pop, reggaeton y el rap. Es importante destacar que el género que más popularidad ha ganado estos últimos años es el reggaeton, teniendo una popularidad media de 68 con sus canciones y, que el pop desde sus inicios, ya tenía una popularidad alta.