

Project - Datavisualisation - Spotify

Tristan Radulescu

2024-07-14

Datavisualisation - Spotify

Introduction

A-travers cette analyse statistique des données de Spotify, nous allons essayer de comprendre les tendances des chansons les plus populaires de 2010 à 2023. Nous allons essayer de répondre aux questions suivantes : Quelles musiques sont similaire dans ce dataset ? Comment evolue la musique au cours de ces dernieres annees ? Et a l'aide de la reponse a la question precedente nous en deduirons les criteres qui font qu'une musique est populaire.

Dataset

```
library(here)
```

```
## here() starts at /home/mihai/Cours/R/do3-dataviz/project
```

```
library(readr)
```

```
top50 <- read_csv(here("data", "playlist_2010to2023.csv"))
```

```
## Rows: 2399 Columns: 23
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr (7): playlist_url, track_id, track_name, album, artist_id, artist_name,...
```

```
## dbl (16): year, track_popularity, artist_popularity, danceability, energy, k...
```

```
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(top50)
```

```
## # A tibble: 6 x 23
```

```
##   playlist_url      year track_id track_name track_popularity album artist_id
```

```
##   <chr>           <dbl> <chr>   <chr>           <dbl> <chr> <chr>
```

```
## 1 https://open.spoti~ 2000 6naxalm~ Oops!...I~      81 Oops~ 26dSoYcl~
```

```
## 2 https://open.spoti~ 2000 2m1hi0n~ All The S~      83 Enem~ 6FBDaR13~
```

```
## 3 https://open.spoti~ 2000 3y4LxiY~ Breathe      66 Brea~ 25NQNrIV~
```

```
## 4 https://open.spoti~ 2000 0v1XpBH~ It's My L~      81 Crush 581V9VcR~
```

```
## 5 https://open.spoti~ 2000 62b0mKY~ Bye Bye B~      75 No S~ 6Ff53Kvc~
```

```
## 6 https://open.spoti~ 2000 5Mmk2ii~ Thong Song    71 Unle~ 6x9QLdzo~
```

```
## # i 16 more variables: artist_name <chr>, artist_genres <chr>,
```

```
## #   artist_popularity <dbl>, danceability <dbl>, energy <dbl>, key <dbl>,
```

```
## #   loudness <dbl>, mode <dbl>, speechiness <dbl>, acousticness <dbl>,
```

```
## #   instrumentalness <dbl>, liveness <dbl>, valence <dbl>, tempo <dbl>,
```

```
## # duration_ms <dbl>, time_signature <dbl>
```

Le dataset sur lequel nous travaillons a été construit à partir de la plateforme Spotify. Il contient les 50 chansons les plus populaires de chaque année de 2010 à 2023. Les variables présentes dans le dataset sont les suivantes : - **year** : l'année de sortie de la chanson, variable qualitative - **track_name** : le titre de la chanson, variable qualitative - **track_popularity** : la popularité de la chanson, variable quantitative continue - **album** : le nom de l'album, variable qualitative - **artist_name** : le nom de l'artiste, variable qualitative - **artist_popularity** : la popularité de l'artiste, variable quantitative continue - **danceability** : la dansabilité de la chanson, variable quantitative continue - **energy** : l'énergie de la chanson, variable quantitative continue - **key** : la tonalité de la chanson, variable qualitative - **loudness** : le volume de la chanson, variable quantitative continue - **mode** : le mode de la chanson, variable qualitative - **speechiness** : la présence de parole dans la chanson, variable quantitative continue - **acousticness** : le niveau d'acousticité de la chanson, variable quantitative continue - **instrumentalness** : le niveau d'instrumentalité de la chanson, variable quantitative continue - **liveness** : le niveau de présence de public dans la chanson, variable quantitative continue - **valence** : la positivité de la chanson, variable quantitative continue - **tempo** : le tempo de la chanson, variable quantitative continue - **duration_ms** : la durée de la chanson en millisecondes, variable quantitative continue - **time_signature** : la signature temporelle de la chanson, variable qualitative

Similarité entre les chansons

Pour répondre à la question de la similarité entre les chansons, nous allons utiliser TSNE pour visualiser les chansons en 2 dimensions. Nous allons utiliser les variables **key**, **loudness**, **tempo** et **duration** car ces variables n'ont pas été calculées par Spotify et sont donc indépendantes.

TSNE est une méthode de réduction de dimension qui permet de visualiser des données en 2 dimensions et de voir les similarités entre les données. Plus les points sont proches, plus les données sont similaires.

Dans un premier temps, nous allons standardiser les variables pour qu'elles aient une moyenne de 0 et un écart-type de 1.

```
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

dataset_track_name <- data.frame(
  track_name = paste0(top50$artist_name, " - ", top50$track_name),
  key = normalize(top50$key),
  loudness = normalize(top50$loudness),
  tempo = normalize(top50$tempo),
  duration_ms = normalize(top50$duration_ms)
)
```

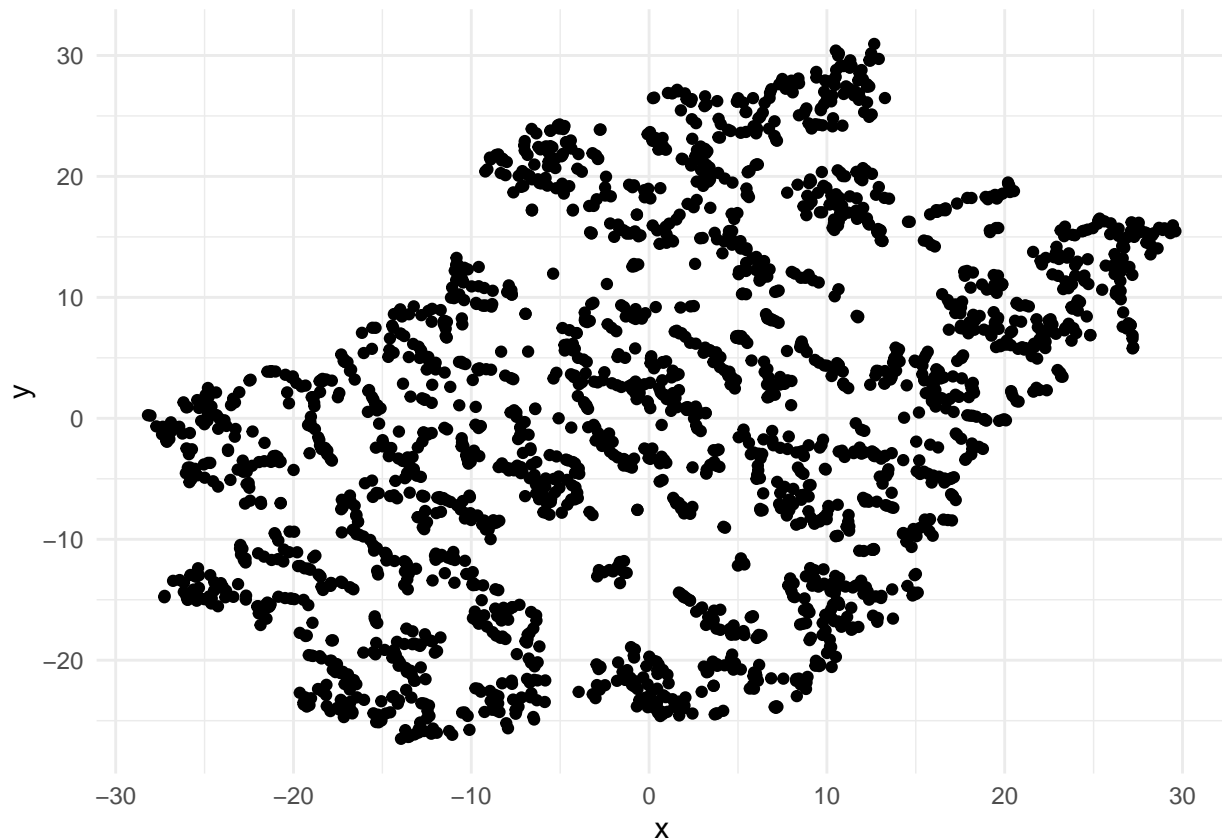
Ensuite, nous allons utiliser la fonction **Rtsne** du package **Rtsne** pour calculer les coordonnées des chansons en 2 dimensions.

```
library(Rtsne)
dataset_track_name <- unique(dataset_track_name)
tsne <- Rtsne(as.matrix(dataset_track_name[, -1]), dims = 2, perplexity = 30, max_iter = 500, check_duplicates = FALSE)
tsne_df <- data.frame(
  track_name = dataset_track_name$track_name,
  x = tsne$Y[, 1],
  y = tsne$Y[, 2]
)
```

Enfin, nous allons visualiser les chansons en 2 dimensions.

```
library(ggplot2)
ggplot(tsne_df, aes(x = x, y = y, label = track_name)) +
```

```
geom_point() +
theme_minimal()
```



Pour voir les noms des chansons, rendez-vous sur le projet ShinyR.

Nous pouvons également définir des clusters de chansons en fonction de leur genre musicale à l'aide de la modalité `artist_genres`.

```
dataset_genres <- data.frame(
  artist_genres = top50$artist_genres,
  key = normalize(top50$key),
  loudness = normalize(top50$loudness),
  tempo = normalize(top50$tempo),
  duration_ms = normalize(top50$duration_ms)
)

tsne_genres <- Rtsne(as.matrix(dataset_genres[, -1]), dims = 2, perplexity = 30, max_iter = 500, check_
tsne_genres_df <- data.frame(
  artist_genres = dataset_genres$artist_genres,
  x = tsne_genres$Y[, 1],
  y = tsne_genres$Y[, 2]
)

colors <- rainbow(length(unique(dataset_genres$artist_genres)))
names(colors) <- unique(dataset_genres$artist_genres)

ggplot(tsne_genres_df, aes(x = x, y = y, color = artist_genres)) +
```

```
geom_point() +
theme_minimal() +
scale_color_manual(values = colors)
```

```
'hip hop', 'hip pop', 'pop rap', 'queens hip hop', 'rap', 'urban contemporary']
op', 'pop rap', 'r&b', 'rap', 'southern hip hop', 'trap queen', 'urban contemporary']
op', 'pop rap', 'rap']
, 'pop dance', 'progressive house', 'uk dance']
l', 'pop', 'pop dance']
```

```
lipop', 'swedish electropop', 'swedish pop']
```

```
rap', 'post-teen pop']
```

```
'ock']
```

```
&b', 'rap', 'southern hip hop', 'trap', 'urban contemporary']
ap', 'st louis rap', 'urban contemporary']
ntemporary']
mporav']
```

```
• ['dance pop', 'girl group', 'post-
• ['dance pop', 'hip hop', 'hip pop
• ['dance pop', 'hip pop', 'neo sou
• ['dance pop', 'hip pop', 'neo sou
• ['dance pop', 'hip pop', 'pop rap
• ['dance pop', 'hip pop', 'pop', 'r&
• ['dance pop', 'hip pop', 'r&b', 'ra
• ['dance pop', 'hip pop', 'r&b', 'ur
• ['dance pop', 'idol', 'pop', 'post-
• ['dance pop', 'latin pop', 'mexic:
• ['dance pop', 'metropolis', 'pc
• ['dance pop', 'miami hip hop', 'f
• ['dance pop', 'miami hip hop', 'f
• ['dance pop', 'neo mellow', 'pop
• ['dance pop', 'neo mellow', 'pop
• ['dance pop', 'neo soul', 'pop ra
• ['dance pop', 'pop dance', 'uk d
• ['dance pop', 'pop rap', 'post-te
• ['dance pop', 'pop rap', 'recoae
```

On voit qu'il n'y a pas de groupes de points clairs se formant dans le graphique. Cela signifie que les chansons ne sont pas regroupées en fonction de leur genre musical, ainsi un genre musical n'est pas défini par les variables `key`, `loudness`, `tempo` et `duration`.

Evolution de la musique

Pour répondre à la question de l'évolution de la musique, nous allons d'abord visualiser l'évolution de la popularité des chansons au cours des années.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
dataset <- data.frame(
  year = top50$year,
  popularity = top50$track_popularity
```

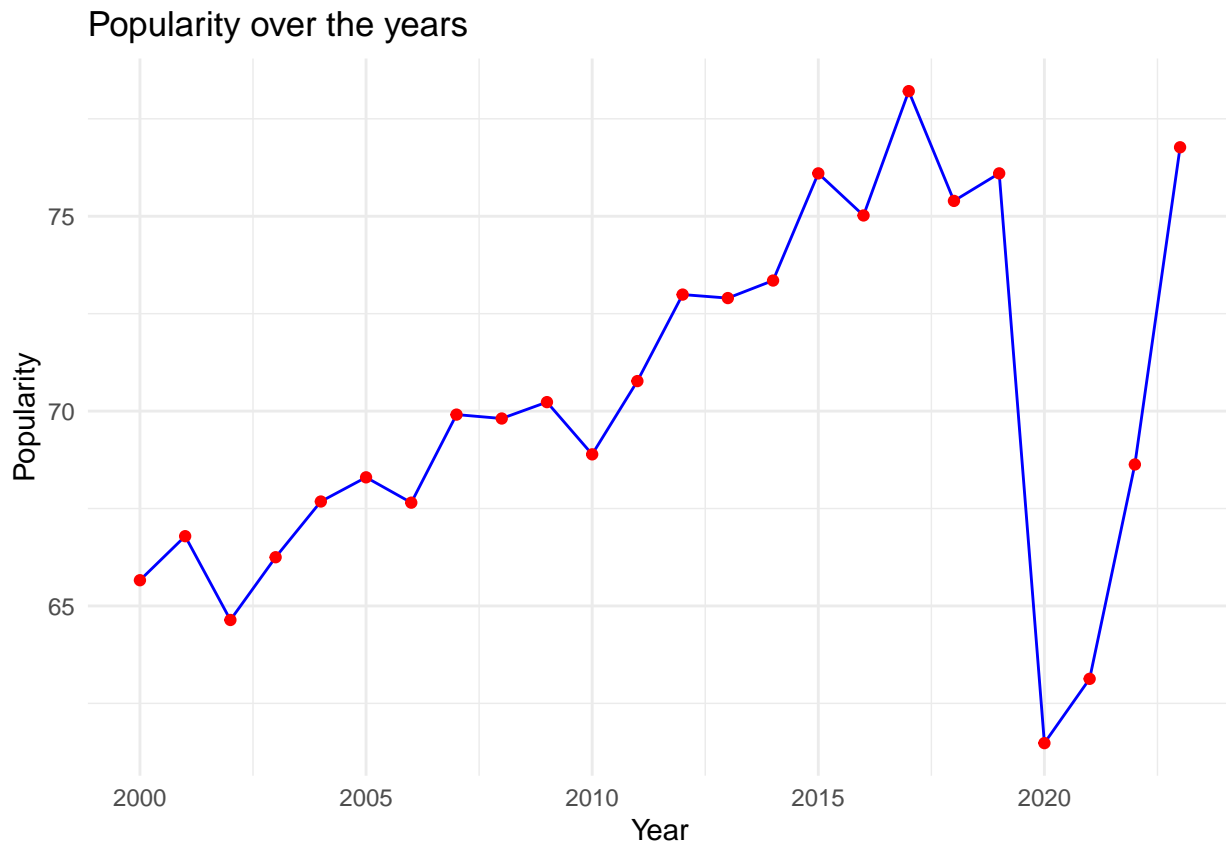
```

)

aggregated_data <- dataset %>%
  group_by(year) %>%
  summarise(avg_popularity = mean(popularity)) %>%
  ungroup()

ggplot(aggregated_data, aes(x = year, y = avg_popularity)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  labs(
    title = "Popularity over the years",
    x = "Year",
    y = "Popularity"
  ) +
  theme_minimal()

```



On voit que la popularité des chansons a tendance à augmenter au cours des années. Cela peut être dû à l'augmentation du nombre d'utilisateurs de Spotify au cours des années. Néanmoins en 2019, on observe une baisse de la popularité des chansons. Cela peut être dû à la pandémie de Covid-19 et donc à une baisse de l'écoute de musique. Ceci reste une hypothèse que nous ne cherchons pas à vérifier dans cette analyse.

Ensuite, nous allons visualiser l'évolution de la durée des chansons au cours des années.

```

ms_to_min <- function(ms) {
  return(ms / 60000)
}

```

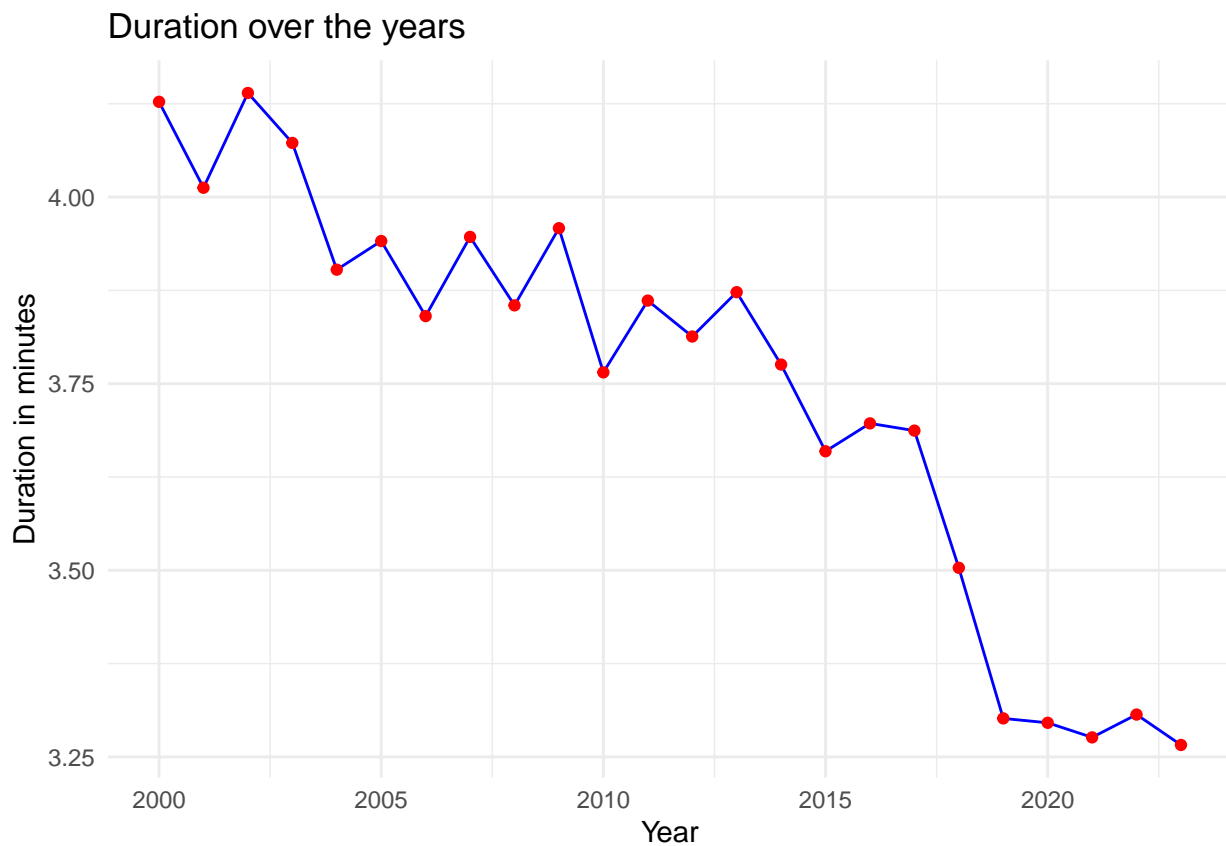
```

dataset <- data.frame(
  year = top50$year,
  duration = ms_to_min(top50$duration_ms)
)

aggregated_data <- dataset %>%
  group_by(year) %>%
  summarise(avg_duration = mean(duration)) %>%
  ungroup()

ggplot(aggregated_data, aes(x = year, y = avg_duration)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  labs(
    title = "Duration over the years",
    x = "Year",
    y = "Duration in minutes"
  ) +
  theme_minimal()

```



On voit que la durée des chansons a tendance à diminuer au cours des années. Cela peut être dû à l'évolution des goûts musicaux des auditeurs. En effet, les chansons plus courtes sont plus faciles à écouter et à retenir. En tout cas il s'agit de la réflexion des maisons de disques qui produisent les chansons.

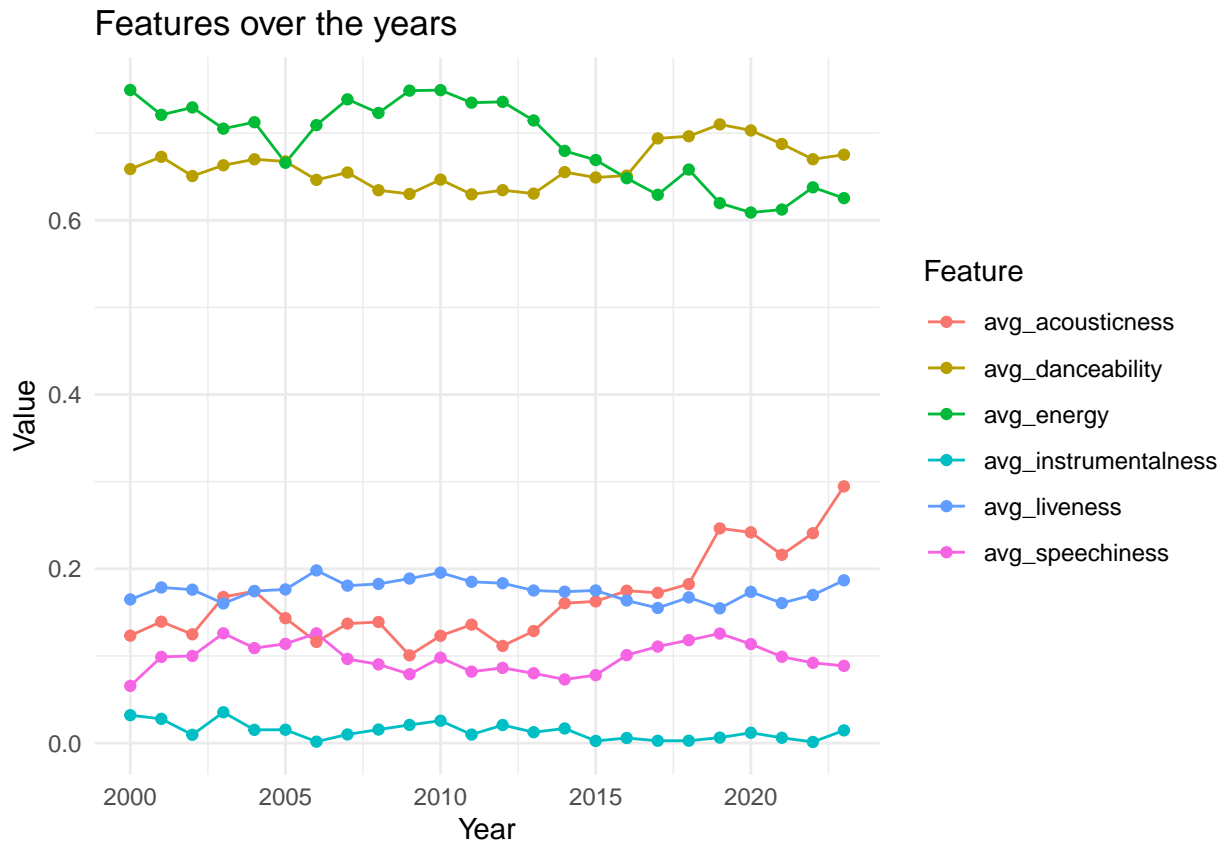
Enfin nous allons visualiser le reste des modalités pour voir l'évolution de la musique au cours des années.

```
library(tidyr)
dataset <- data.frame(
  year = top50$year,
  danceability = top50$danceability,
  energy = top50$energy,
  speechiness = top50$speechiness,
  acousticness = top50$acousticness,
  instrumentalness = top50$instrumentalness,
  liveness = top50$liveness
)

aggregated_data <- dataset %>%
  group_by(year) %>%
  summarise(
    avg_danceability = mean(danceability),
    avg_energy = mean(energy),
    avg_speechiness = mean(speechiness),
    avg_acousticness = mean(acousticness),
    avg_instrumentalness = mean(instrumentalness),
    avg_liveness = mean(liveness)
  ) %>%
  ungroup()

aggregated_data <- pivot_longer(aggregated_data, cols = c("avg_danceability", "avg_energy", "avg_speechiness", "avg_acousticness", "avg_instrumentalness", "avg_liveness"))

ggplot(aggregated_data, aes(x = year, y = value, color = feature)) +
  geom_line() +
  geom_point() +
  labs(
    title = "Features over the years",
    x = "Year",
    y = "Value",
    color = "Feature"
  ) +
  theme_minimal()
```



Nous voyons que toutes ces modalites n'évoluent pas.

Qu'est ce qui rend une musique populaire ?

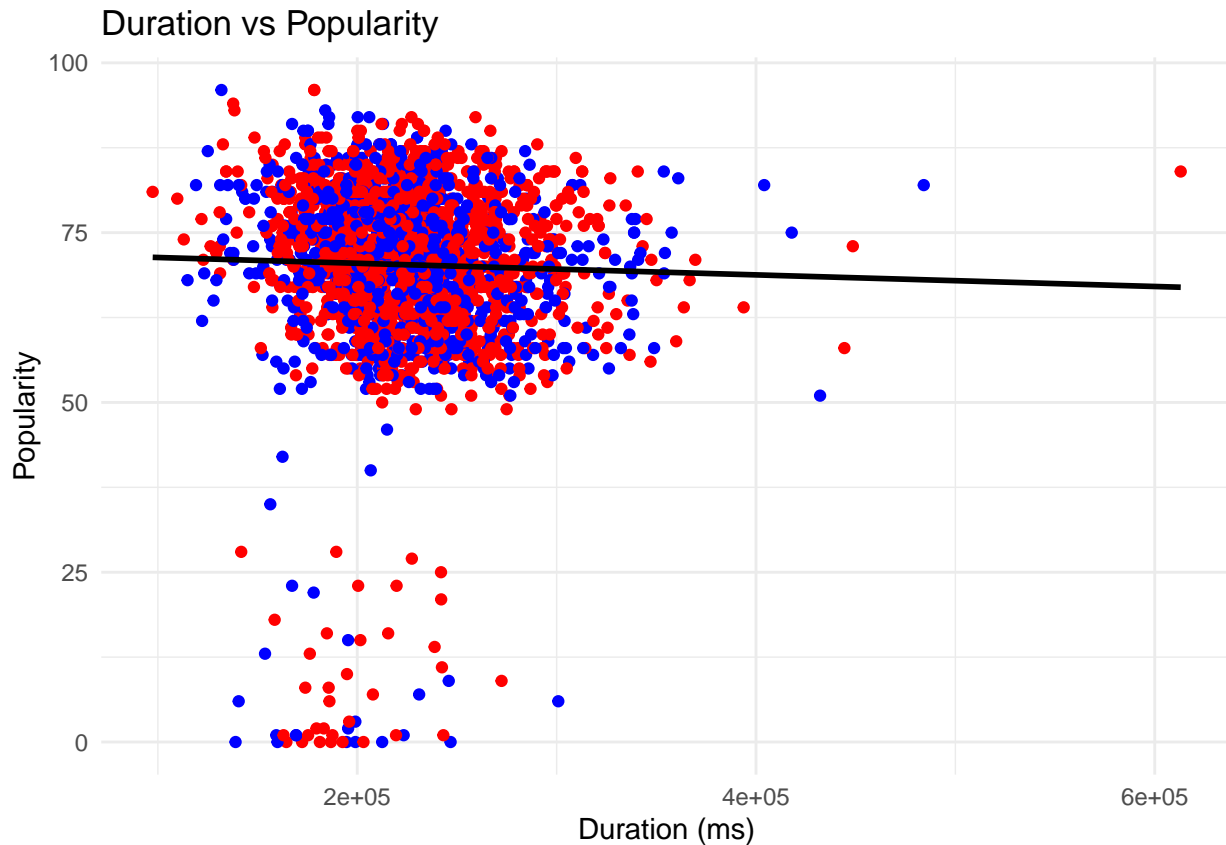
On deduit de la question precedente que le seul critere qui a evolue au cours des annees est la duree des chansons. Nous allons donc verifier si la duree des chansons est correlee a la popularite des chansons.

```
dataset <- data.frame(
  title = top50$track_name,
  duration = top50$duration_ms,
  popularity = top50$track_popularity,
  modes = top50$mode
)

dataset$color <- ifelse(dataset$mode == 1, "red", "blue")

ggplot(dataset, aes(x = duration, y = popularity, color = color)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  scale_color_identity() +
  labs(
    title = "Duration vs Popularity",
    x = "Duration (ms)",
    y = "Popularity",
    color = "Mode"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

On voit qu'il n'y a pas de corrélation entre la durée des chansons et leur popularité. Cela signifie que la durée des chansons n'est pas un critère déterminant pour la popularité des chansons.

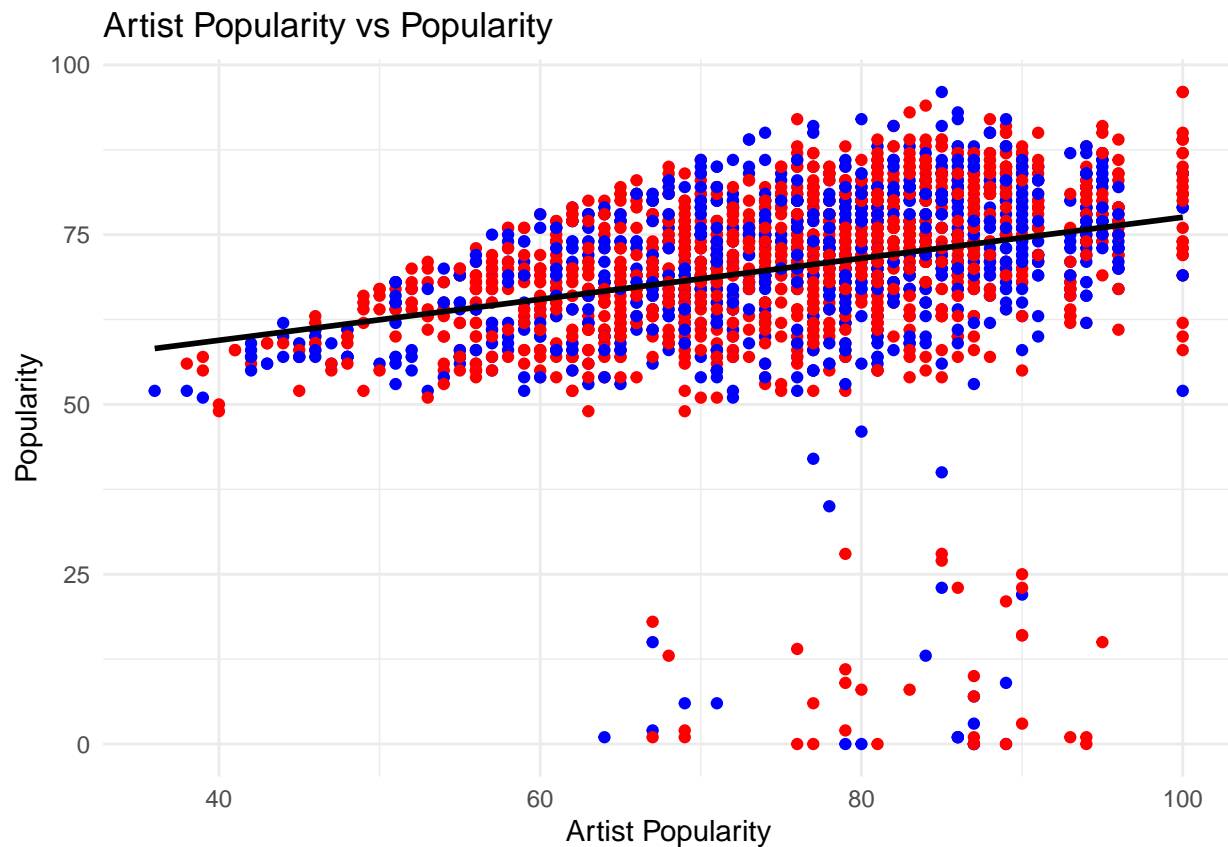
Pour vérifier si toute fois il existe un critère déterminant pour la popularité des chansons, nous allons tracer une régression linéaire entre la popularité des chansons et la popularité des artistes.

```
dataset <- data.frame(
  title = top50$track_name,
  artist_popularity = top50$artist_popularity,
  popularity = top50$track_popularity,
  modes = top50$mode
)

dataset$color <- ifelse(dataset$mode == 1, "red", "blue")

ggplot(dataset, aes(x = artist_popularity, y = popularity, color = color)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  scale_color_identity() +
  labs(
    title = "Artist Popularity vs Popularity",
    x = "Artist Popularity",
    y = "Popularity",
    color = "Mode"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



On voit qu'il y a une corrélation entre la popularité des artistes et la popularité de leurs chansons. Cela signifie que la popularité des artistes est un critère déterminant pour la popularité des chansons.

Conclusion

Nous pouvons conclure de cette analyse que la musique est quelque chose de très subjectif. En effet, il n'y a pas de critère déterminant pour la popularité des chansons. Cependant, nous avons vu que la popularité des artistes est un critère déterminant pour la popularité des chansons. Cela signifie que les auditeurs de Spotify ont tendance à écouter les chansons des artistes les plus populaires. Il est donc essentiel pour un artiste de chercher à construire une communauté autour de sa musique pour augmenter sa popularité au lieu d'essayer de construire sa musique autour de critères généraux. Il n'existe pas de recette miracle pour créer une chanson populaire, il faut simplement créer de la musique qui nous plaît et qui plaît à notre communauté.