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 repondre 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 62bOmKY~ 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 ete construit a partir de la plateforme Spotify. Il contient les 50 chansons les plus populaires de chaque annee de 2010 a 2023. Les variables presentes dans le dataset sont les suivantes : - `year` : l'annee de sortie de la chanson, variable qualitative - `track_name` : le titre de la chanson, variable qualitative - `track_popularity` : la popularite 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 popularite de l'artiste, variable quantitative continue - `danceability` : la dansabilite de la chanson, variable quantitative continue - `energy` : l'energie de la chanson, variable quantitative continue - `key` : la tonalite de la chanson, variable qualitative - `loudness` : le volume de la chanson, variable quantitative continue - `mode` : le mode de la chanson, variable qualitative - `speechiness` : la presence de parole dans la chanson, variable quantitative continue - `acousticness` : le niveau d'acousticite de la chanson, variable quantitative continue - `instrumentalness` : le niveau d'instrumentalite de la chanson, variable quantitative continue - `liveness` : le niveau de presence de public dans la chanson, variable quantitative continue - `valence` : la positivite de la chanson, variable quantitative continue - `tempo` : le tempo de la chanson, variable quantitative continue - `duration_ms` : la duree de la chanson en millisecondes, variable quantitative continue - `time_signature` : la signature temporelle de la chanson, variable qualitative

Similarite entre les chansons

Pour repondre a la question de la similarite 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 ete calculees par Spotify et sont donc independantes.

TSNE est une methode de reduction de dimension qui permet de visualiser des donnees en 2 dimensions et de voir les similarites entre les donnees. Plus les points sont proches, plus les donnees sont similaires.

Dans un premier temps, nous allons standardiser les variables pour qu'elles aient une moyenne de 0 et un ecart-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)
)
```

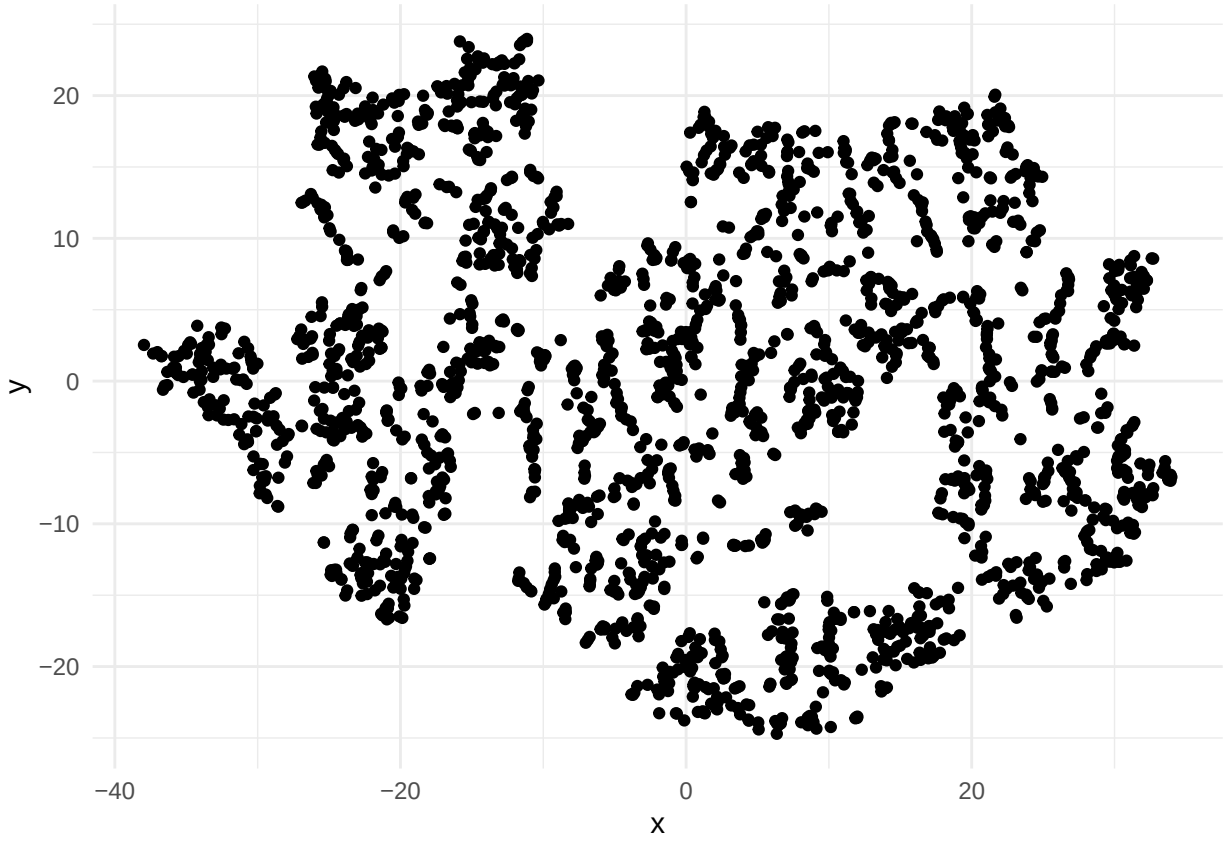
Ensuite, nous allons utiliser la fonction `Rtsne` du package `Rtsne` pour calculer les coordonnees des chansons en 2 dimensions.

```
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_dupl
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()
```

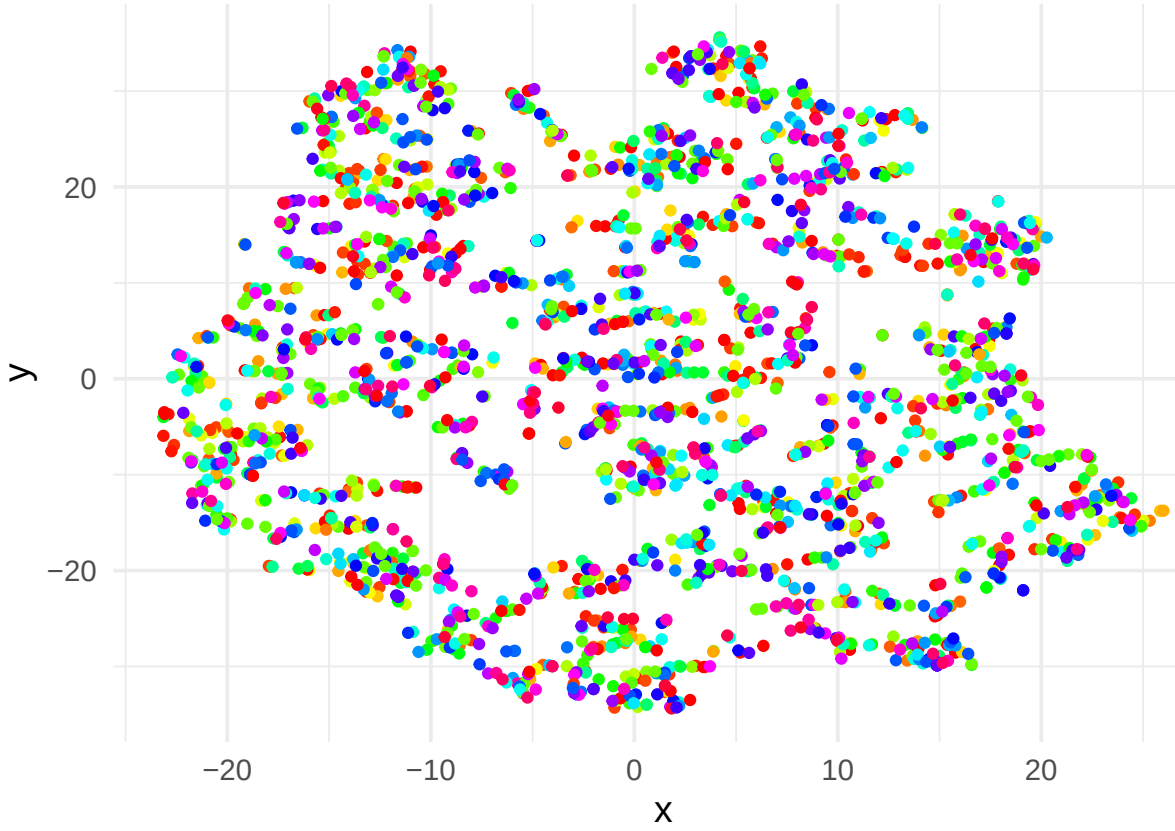


Pour voir les noms des chansons, rendez-vous sur le projet ShinyR.

Nous pouvons egalement definir des clusters de chansons en fonction de leur genre musicale a l'aide de la modalite `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() +
    scale_color_manual(values = colors)+
    theme_minimal(base_size = 14) +
    theme(legend.position = "none")
```



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 regroupees en fonction de leur genre musical, ainsi un genre musical n'est pas defini par les variables `key`, `loudness`, `tempo` et `duration`.

Pour ces deux diagrammes, j'ai du cacher la legende car elle etait trop grande pour etre affichee. Pour voir les noms des chansons et des genres, rendez-vous sur le projet ShinyR.

Evolution de la musique

Pour repondre a la question de l'evolution de la musique, nous allons d'abord visualiser l'evolution de la popularite des chansons au cours des annees.

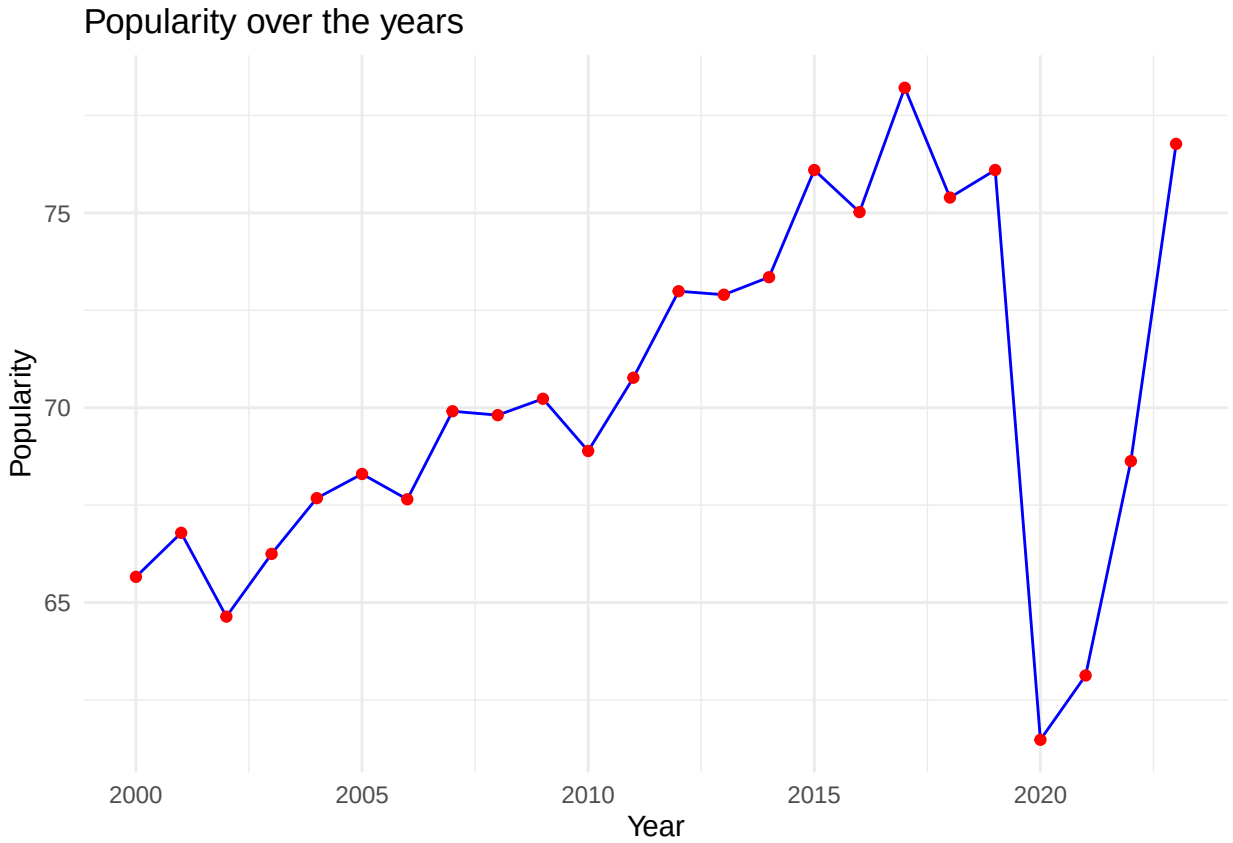
```
library(dplyr)
```

```
##
## 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
```

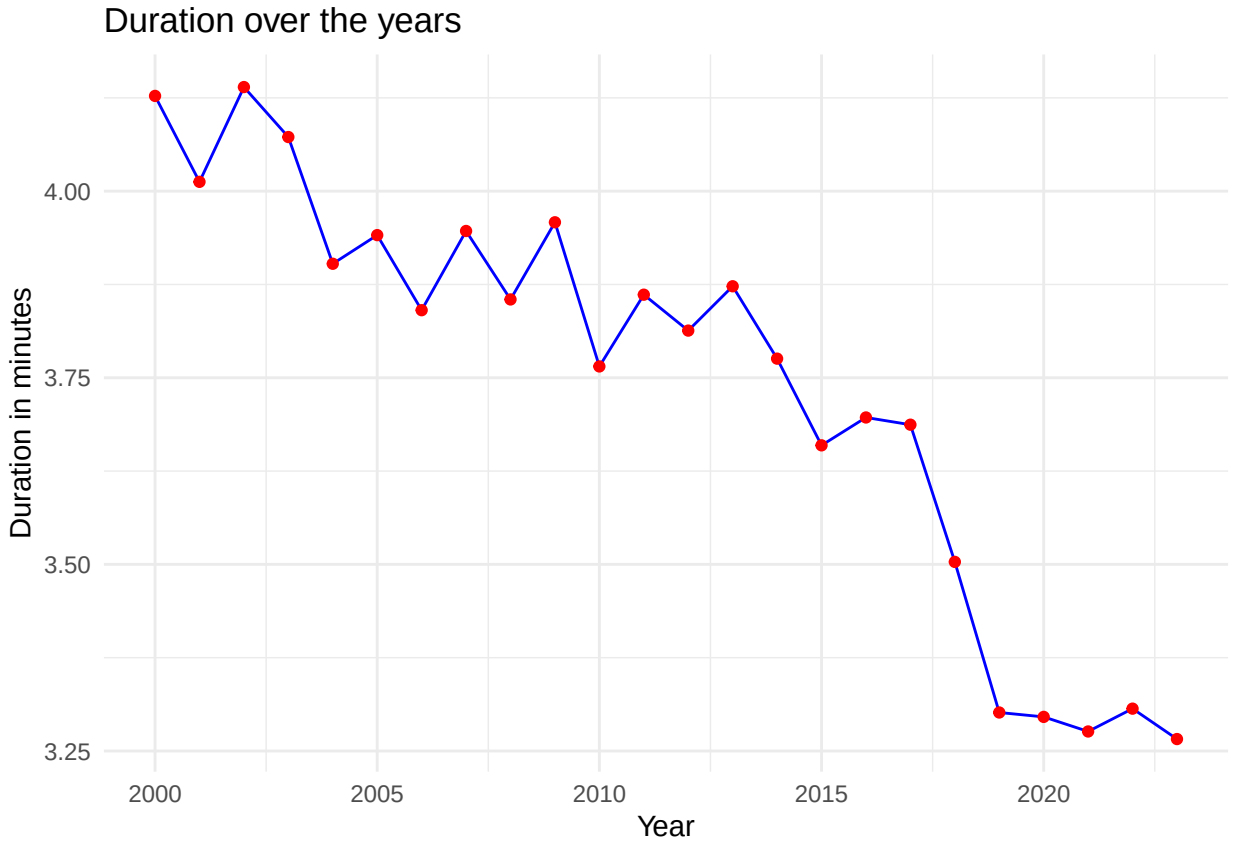
```r
dataset <- data.frame(
    year = top50$year,
    popularity = top50$track_popularity
)

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 popularite des chansons a tendance a augmenter au cours des annees. Cela peut etre du a l'augmentation du nombre d'utilisateurs de Spotify au cours des annees. Neanmoins en 2019, on observe une baisse de la popularite des chansons. Cela peut etre du a la pandemie de Covid-19 et donc a une baisse de l'ecoute de musique. Ceci reste une hypothese que nous ne chercheons pas a verifier dans cette analyse.

Ensuite, nous allons visualiser l'evolution de la duree des chansons au cours des annees.

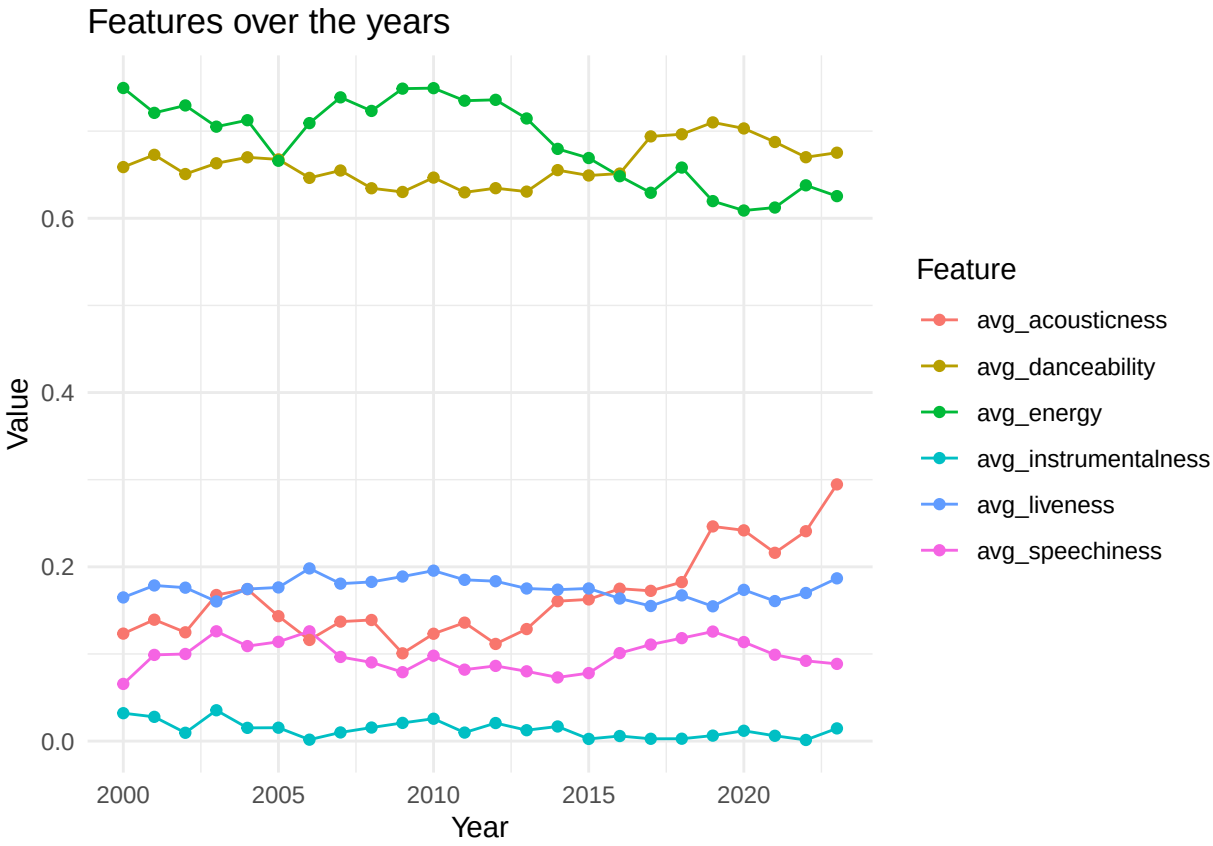
```r
ms_to_min <- function(ms) {
    return(ms / 60000)
}

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 duree des chansons a tendance a diminuer au cours des annees. Cela peut etre du a l'evolution des gouts musicaux des auditeurs. En effet, les chansons plus courtes sont plus faciles a ecouter et a retenir. En tout cas il s'agit de la reflexion des maisons de disques qui produisent les chansons.

Enfin nous allons visualiser le reste des modalites pour voir l'evolution de la musique au cours des annees.

```
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_speechi



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()
```



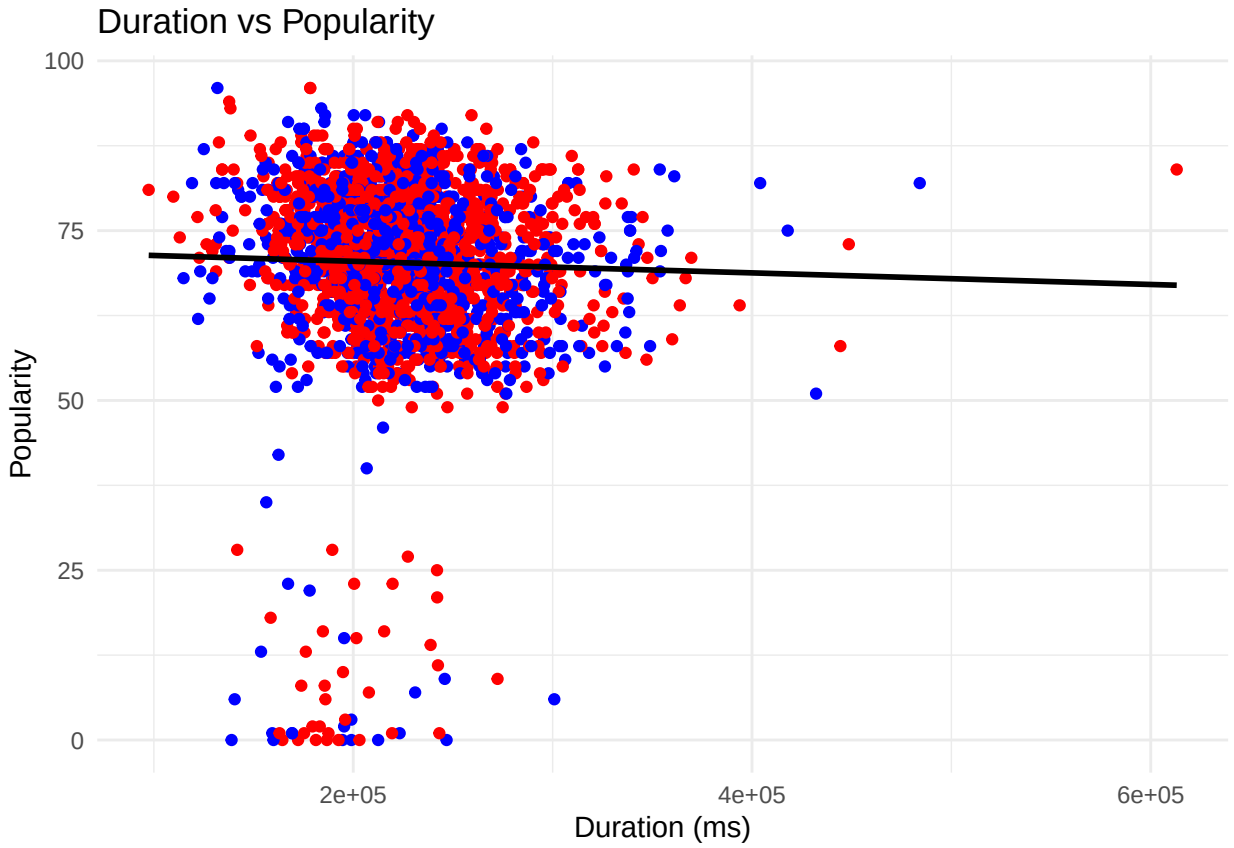
Nous voyons que toutes ces modalites n'evoluent pas.

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'
```

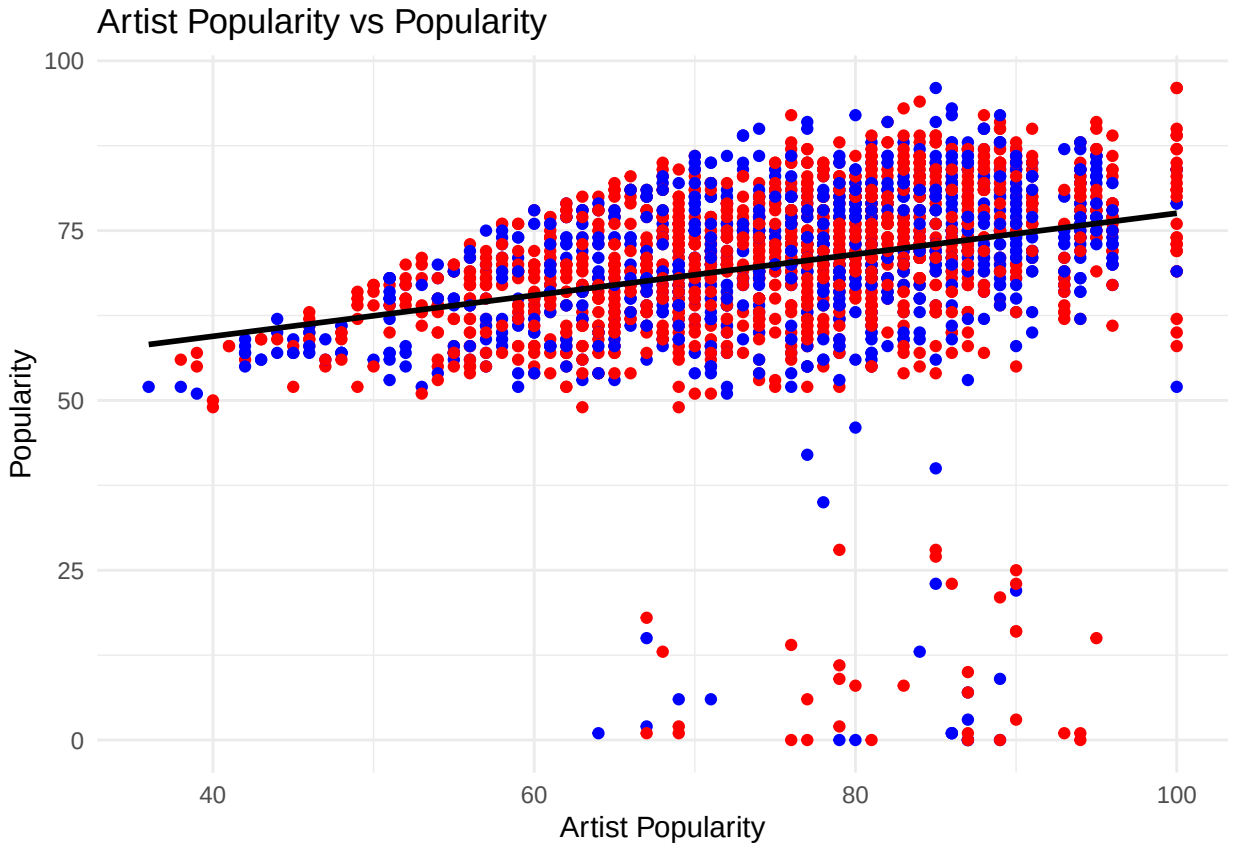


On voit qu'il n'y a pas de correlation entre la duree des chansons et leur popularite. Cela signifie que la duree des chansons n'est pas un critere determinant pour la popularite des chansons.

Pour verifier si toute fois il existe un critere determinant pour la popularite des chansons, nous allons tracer une regression lineaire entre la popularite des chansons et la popularite 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 correlation entre la popularite des artistes et la popularite de leurs chansons. Cela signifie que la popularite des artistes est un critere determinant pour la popularite des chansons.

Conclusion

Nous pouvons conclure de cette analyse que la musique est quelque chose de tres subjectif. En effet, il n'y a pas de critere determinant pour la popularite des chansons. Cependant, nous avons vu que la popularite des artistes est un critere determinant pour la popularite des chansons. Cela signifie que les auditeurs de Spotify ont tendance a ecouter les chansons des artistes les plus populaires. Il est donc essentiel pour un artiste de chercher a construire une communaute autour de sa musique pour augmenter sa popularite au lieu d'essayer de construire sa musique autour de criteres generaux. Il n'existe pas de recette miracle pour creer une chanson populaire, il faut simplement creer de la musique qui nous plait et qui plait a notre communaute.