SPOTIFY TREND ANALYSIS - YEAR 2017

TEFFY ANNIE GEORGE

APR 10, 2022

Data Sources:

1) Audio Features of Top 100 Spotify Songs of 2017

The data represents the audio features of the top 100 songs of year 2017 from Spotify. The data set has 17 columns where majority column represents the features such as loudness, danceability, instrumentalness, tempo etc

2) Spotify's Worldwide Daily Song Ranking

This data represents the daily ranking of the most listened songs from 2017 to 2018 by Spotify users. This data represents the top 200 that is most streamed daily in multiple countries.

Packages:

dplyr : To enable dataframe manipulation in an easy way

ggplot2: To create 'declaratively' graphics from data in a data frame

corrplot: To provide a visual view on correlation matrix

ggthemes: To provide ggplot with extra themes, geoms and scales

rpart: To build classification or regression trees

rpart.control: To control the parameter of the rpart fit

Tool Used:

R- Studio

Import the Data to R

Code in R:

```
# STEP 1: IMPORT DATA TO R
# Get the current working directory
getwd()
# Set the working directory to file path location
setwd("C:\\Data Files")

spotify_df <- read.csv("featuresdf.csv")
daily_df <- read.csv("data.csv")

# Taking the summary of spotify data to check if there are any missing values
summary(spotify_df)
# Taking the summary of daily data to check if there are any missing values
summary(daily_df)
# Data has no NA's .This is good data that requires no cleaning for analysis purpose
```

Output:

Summary of Spotify Data

```
> # Taking the summary of spotify data to check if there are any missing values
> summary(spotify_df)
      id                name              artists           danceability
 Length:100         Length:100         Length:100         Min.   :0.2580
 Class :character   Class :character   Class :character   1st Qu.:0.6350
 Mode  :character   Mode  :character   Mode  :character   Median :0.7140
                                                          Mean   :0.6968
                                                          3rd Qu.:0.7702
                                                          Max.   :0.9270
     energy            key           loudness            mode       speechiness
 Min.   :0.3460   Min.   : 0.00   Min.   :-11.462   Min.   :0.00   Min.   :0.02320
 1st Qu.:0.5565   1st Qu.: 2.00   1st Qu.: -6.595   1st Qu.:0.00   1st Qu.:0.04312
 Median :0.6675   Median : 6.00   Median : -5.437   Median :1.00   Median :0.06265
 Mean   :0.6607   Mean   : 5.57   Mean   : -5.653   Mean   :0.58   Mean   :0.10397
 3rd Qu.:0.7875   3rd Qu.: 9.00   3rd Qu.: -4.327   3rd Qu.:1.00   3rd Qu.:0.12300
 Max.   :0.9320   Max.   :11.00   Max.   : -2.396   Max.   :1.00   Max.   :0.43100
  acousticness      instrumentalness        liveness          valence
 Min.   :0.000259   Min.   :0.00000000   Min.   :0.04240   Min.   :0.0862
 1st Qu.:0.039100   1st Qu.:0.00000000   1st Qu.:0.09828   1st Qu.:0.3755
 Median :0.106500   Median :0.00000000   Median :0.12500   Median :0.5025
 Mean   :0.166306   Mean   :0.00479614   Mean   :0.15061   Mean   :0.5170
 3rd Qu.:0.231250   3rd Qu.:0.00001335   3rd Qu.:0.17925   3rd Qu.:0.6790
 Max.   :0.695000   Max.   :0.21000000   Max.   :0.44000   Max.   :0.9660
     tempo         duration_ms     time_signature
 Min.   : 75.02   Min.   :165387   Min.   :3.00
 1st Qu.: 99.91   1st Qu.:198491   1st Qu.:4.00
 Median :112.47   Median :214106   Median :4.00
 Mean   :119.20   Mean   :218387   Mean   :3.99
 3rd Qu.:137.17   3rd Qu.:230543   3rd Qu.:4.00
 Max.   :199.86   Max.   :343150   Max.   :4.00
```

Summary of Daily Data

```
> # Taking the summary of daily data to check if there are any missing values
> summary(daily_df)
    Position       Track.Name          Artist            Streams
 Min.   :  1.00   Length:3441197     Length:3441197     Min.   :    1001
 1st Qu.: 45.00   Class :character   Class :character   1st Qu.:    3322
 Median : 92.00   Mode  :character   Mode  :character   Median :    9227
 Mean   : 94.64                                        Mean   :   51892
 3rd Qu.:143.00                                        3rd Qu.:   29658
 Max.   :200.00                                        Max.   :11381520
     URL                Date              Region
 Length:3441197     Length:3441197     Length:3441197
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
```

I can see that the dataset provided has no missing details and is a good fit to proceed for analysis.

Data Analysis

- Artists who have the most no of songs in the top 100 list is listed below.

Code in R

```
# Find the top artist from the top 100 list
top_artists <- spotify_df %>%
            count(artists)
top_artists <- top_artists %>%
            arrange(desc(n))
head(top_artists)

#Getting the class types of each variables
str(top_artists)

#setting the field 'n' as factor as the data can be in seen in descending order
top_artists$artists <-factor(top_artists$artists,
                        levels=top_artists$artists[order(top_artists$n)])

#fetching the top 10 artists
top_artists_highest_songs_top10 <- top_artists[1:10,]

#Plotting the top most artists from the top 100 list
ggplot(top_artists_highest_songs_top10, aes(x = artists, y = n)) +
  geom_bar(stat='identity',fill = "coral3",alpha = 0.8,width = 0.9) +
  geom_text(aes(label=n), color = 'white', hjust = 2, size = 4) +
  ggtitle("Top Artists of 2017") +
  labs(x = "Artist",y = "Total Songs Per Artist Appeared in Top 100") +
  theme(axis.title.x = element_text(colour = "DarkGreen",size= 8),
        axis.title.y = element_text(colour = "DarkGreen",size= 8),
        axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8),
        plot.title = element_text(color = "Maroon",
                                  size = 10,
                                  family = "Arial",
                                  hjust = 0.5))+
```

coord_flip()

#Chain Smokers and Ed Sheeran topped the list

Output:

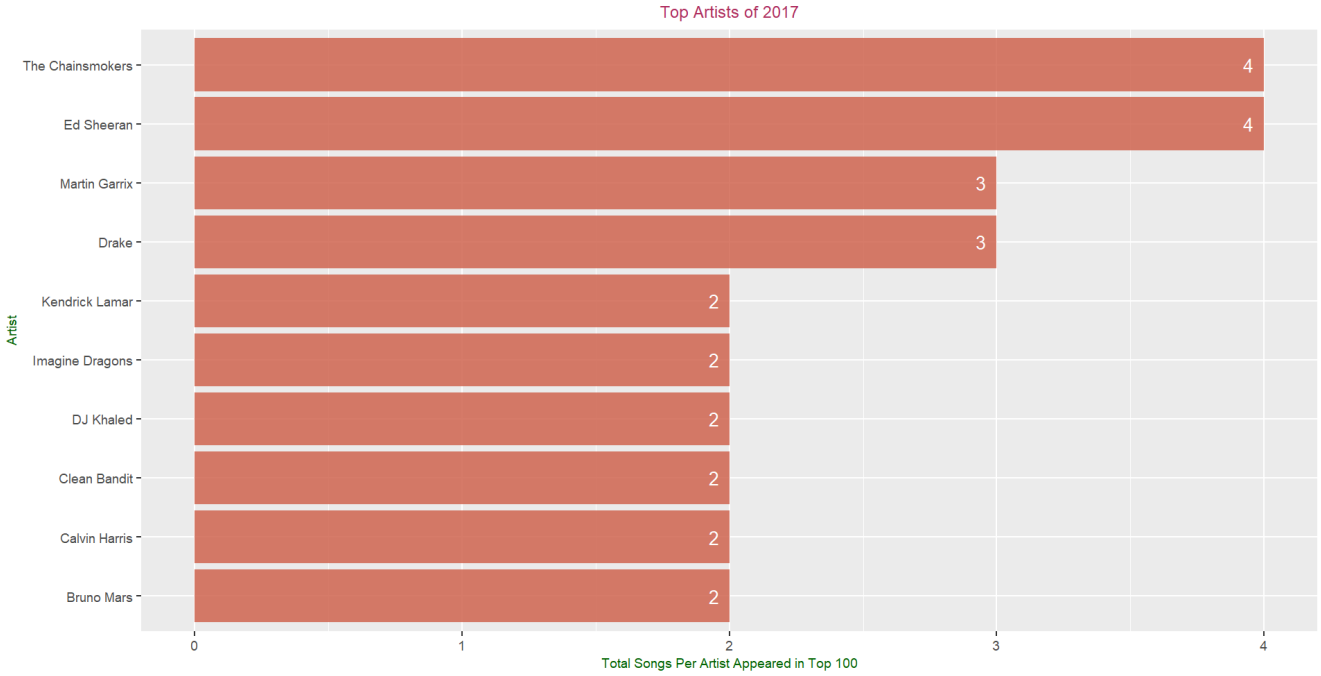
The below set shows the top 5 artists that have the highest no of songs in the list. The graphical representation of the top 10 artists having the highest number of songs in shown

```
> # STEP 2: DATA ANALYSIS
>
> # Find the top artist from the top 100 list
> top_artists <- spotify_df %>%
+                count(artists)
> top_artists <- top_artists %>%
+                arrange(desc(n))
> head(top_artists)
           artists n
1       Ed Sheeran 4
2 The Chainsmokers 4
3            Drake 3
4    Martin Garrix 3
5       Bruno Mars 2
6    Calvin Harris 2
>
```

Setting the *artists* variable as Factor as the graphical visualization can be shown in descending order

```
> #Getting the class types of each variables
> str(top_artists)
'data.frame':   78 obs. of  2 variables:
 $ artists: chr  "Ed Sheeran" "The Chainsmokers" "Drake" "Martin Garrix" ...
 $ n      : int  4 4 3 3 2 2 2 2 2 2 ...
>
> #setting the field 'n' as factor as the data can be in seen in descending order
> top_artists$artists <-factor(top_artists$artists,
+                       levels=top_artists$artists[order(top_artists$n)])
> #Getting the class types of each variables
> str(top_artists)
'data.frame':   78 obs. of  2 variables:
 $ artists: Factor w/ 78 levels "21 Savage","AJR",..: 77 78 75 76 63 64 65 66 67 68 ...
 $ n      : int  4 4 3 3 2 2 2 2 2 2 ...
>
```

Graphical Representation of Top 10 Artists having the greatest number of songs



From the graph, 'The Chainsmokers' and 'Ed Sheeran' tops the list with 4 songs. Following are Martin Garrix and Drake with 3 songs each. The rest show the artists that have 2 songs in the top list

- Artists who have the most streamed songs.

Code in R

```
# Based on streamed data in Spotify,
# listing the top streamed artist and top streamed song
# Find the top streamed artist from the top 100 list

top_streamed_artists <- daily_df %>%
                        group_by(Artist)  %>%
                        summarise(Total_Streamed_Time = sum(Streams)) %>%
                        arrange(desc(Total_Streamed_Time))

head(top_streamed_artists)

# Plotting the top 10 of Top Streamed Artists
options(scipen = 99999999)
top_streamed_artists_top10 <- top_streamed_artists[1:10,]

# setting the field 'Total_Streamed_Time' as factor as the data can be in seen in
descending order
top_streamed_artists_top10$Artist <-
factor(top_streamed_artists_top10$Artist,levels=top_streamed_artists_top10$Artist[ord
er(top_streamed_artists_top10$Total_Streamed_Time)])

#Plotting the artists with top streamed songs
ggplot(top_streamed_artists_top10, aes(x = Artist, y = Total_Streamed_Time)) +
  geom_bar(stat='identity',fill = c("turquoise4"),alpha = 0.8,width = 0.8) +
  geom_text(aes(label=Total_Streamed_Time), color = 'white', hjust = 2, size = 4) +
  ggtitle("Top 10 Artists with Highest Streamed Songs") +
  labs(x = "Artist",y = "Total Streamed Time") +
  theme(axis.title.x = element_text(colour = "DarkGreen",size= 8),
        axis.title.y = element_text(colour = "DarkGreen",size= 8),
        axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8),
        plot.title = element_text(color = "Maroon",
                                  size = 10,
```

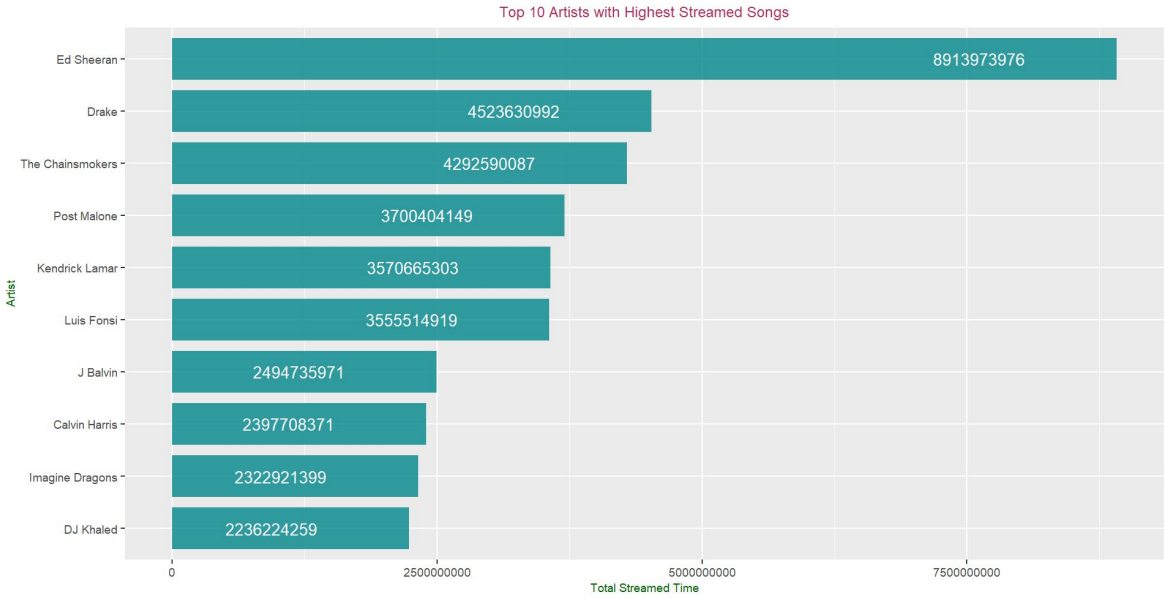
```
                    family = "Arial",
                    hjust = 0.5))+
coord_flip()
```

Output:

The below set shows the top 5 artists that have the maximum streamed songs in the list. The graphical representation of the top 10 artists having the highest streamed time of their songs is shown

```
> head(top_streamed_artists)
# A tibble: 6 x 2
  Artist           Total_Streamed_Time
  <chr>                          <dbl>
1 Ed Sheeran                8913973976
2 Drake                     4523630992
3 The Chainsmokers          4292590087
4 Post Malone               3700404149
5 Kendrick Lamar            3570665303
6 Luis Fonsi                3555514919
```

Graphical Representation of Top 10 Artists having the highest streamed songs



Ed Sheeran and Drake are the top artists with the top streamed time

- Top Streamed Songs

Code in R

```
# Find the top streamed song from the top 100 list
top_streamed_songs <- daily_df %>%
                      group_by(Track.Name)  %>%
                      summarise(Total_Streamed_Time = sum(Streams)) %>%
                      arrange(desc(Total_Streamed_Time))

head(top_streamed_songs)

# Plotting the top 10 of Top Streamed Songs
options(scipen = 99999999)
top_streamed_songs_top10 <- top_streamed_songs[1:10,]

#setting the field 'Total_Streamed_Time' as factor as the data can be in seen in
descending order
top_streamed_songs_top10$Track.Name <-
factor(top_streamed_songs_top10$Track.Name,levels=top_streamed_songs_top10$Trac
k.Name[order(top_streamed_songs_top10$Total_Streamed_Time)])

str(top_streamed_songs_top10)

#Plotting the top streamed songs
ggplot(top_streamed_songs_top10, aes(x = Track.Name, y = Total_Streamed_Time)) +
  geom_bar(stat='identity',fill = c("slateblue"),alpha = 0.8,width = 0.9) +
  geom_text(aes(label=Total_Streamed_Time), color = 'white', hjust = 2, size = 4) +
  ggtitle("Top 10 Streamed Songs") +
  labs(x = "Songs",y = "Total Streamed Time") +
  theme(axis.title.x = element_text(colour = "DarkGreen",size= 8),
        axis.title.y = element_text(colour = "DarkGreen",size= 8),
        axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8),
        plot.title = element_text(color = "Maroon",
                                  size = 10,
                                  family = "Arial",
```

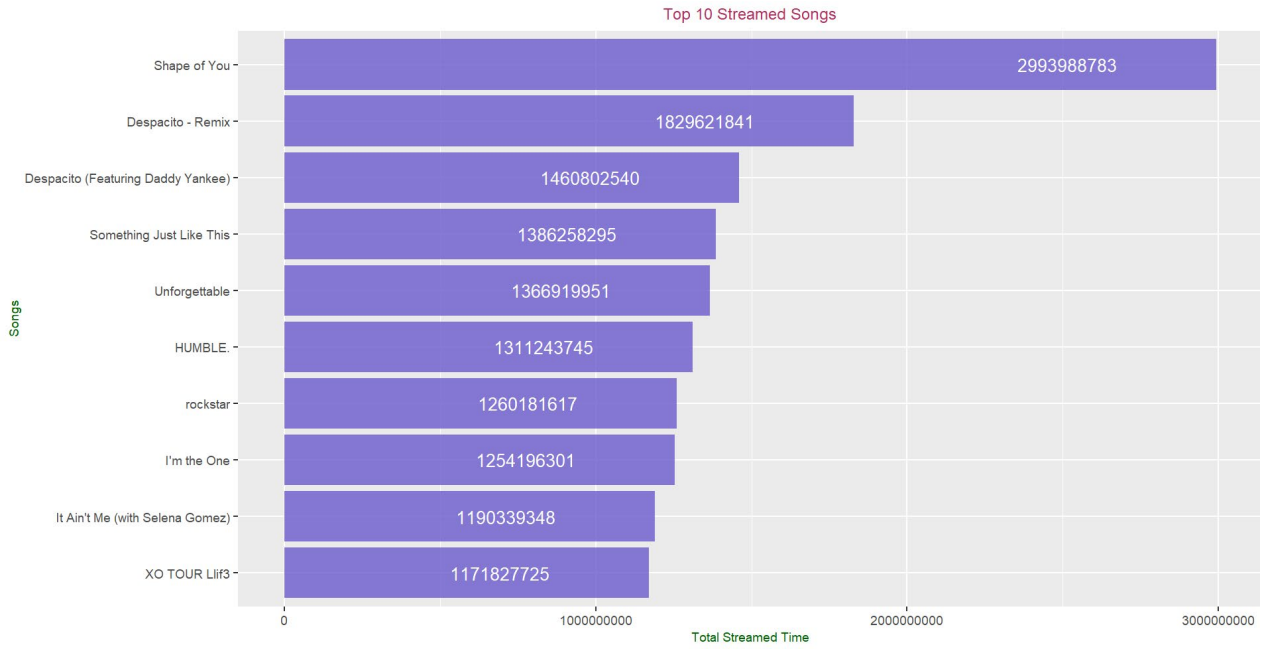
```
                    hjust = 0.5))+
coord_flip()
```

Output:

The below set shows the top 5 streamed songs in the list. The graphical representation of the top 10 songs with highest streamed time of their songs is shown

```
> head(top_streamed_songs)
# A tibble: 6 x 2
  Track.Name                          Total_Streamed_Time
  <chr>                                             <dbl>
1 Shape of You                                 2993988783
2 Despacito - Remix                            1829621841
3 Despacito (Featuring Daddy Yankee)           1460802540
4 Something Just Like This                     1386258295
5 Unforgettable                                1366919951
6 HUMBLE.                                      1311243745
>
```

Graphical Representation of Top 10 highest streamed songs



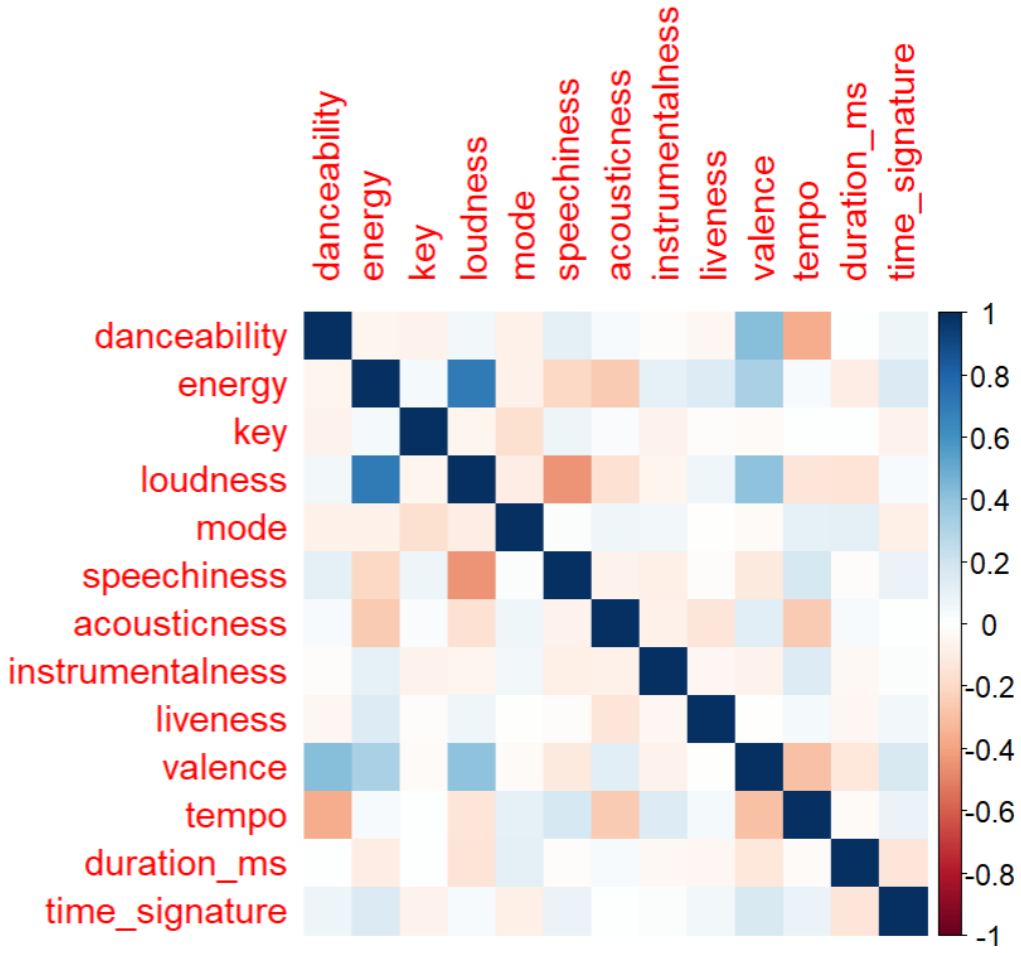
Shape of Drake and Despacito are the top songs with the highest streamed time

- Dependency of Audio Features

Let's determine the correlation of audio features for the top 100 songs

Code in R

```
features_df <- spotify_df[,-c(1:3)]
cor_features <- cor(features_df,method = "pearson", use="na.or.complete")
corrplot(cor_features,use="na.or.complete",method = "color")
```

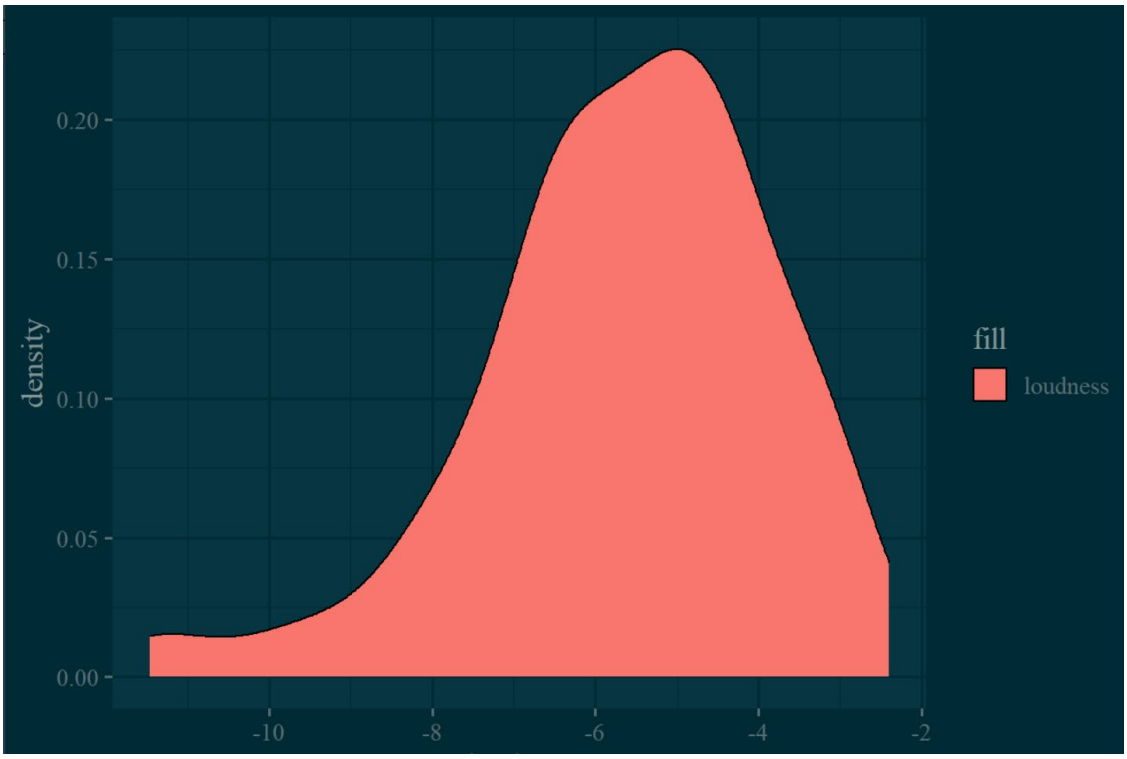


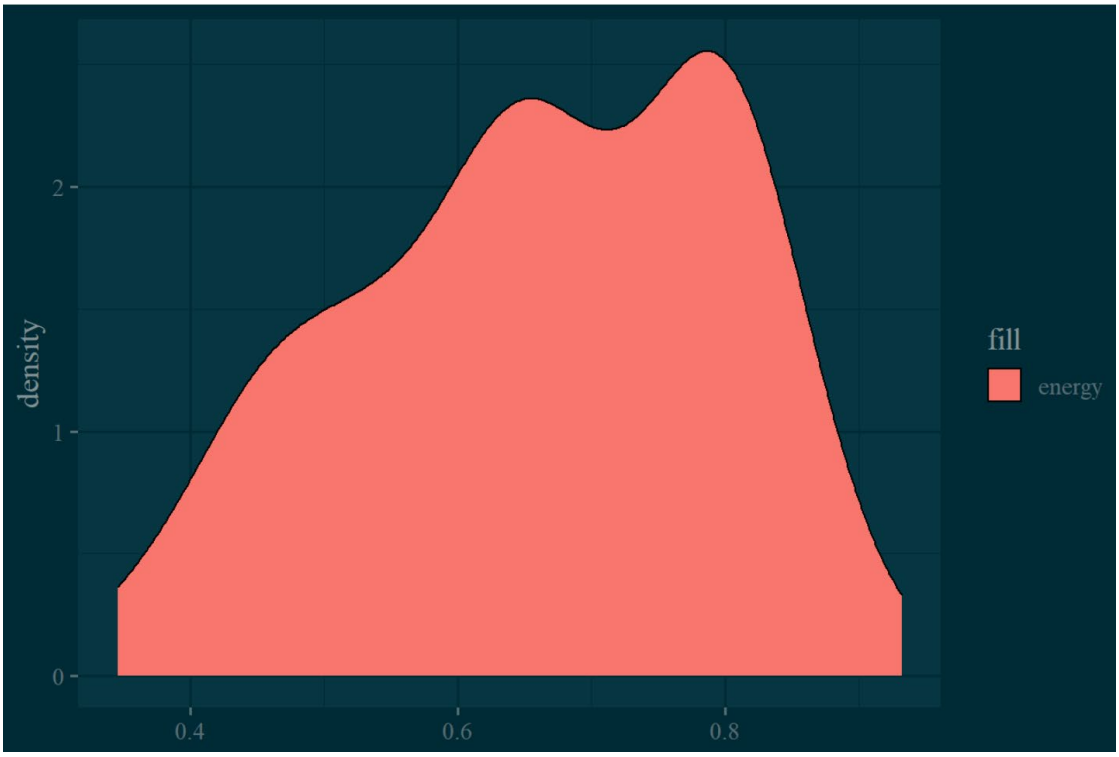
Here, we can see that the maximum correlation is shown for loudness and energy.
Valence is positively correlated with loudness and danceability.
Speechness and loudness show a negative correlation between them.

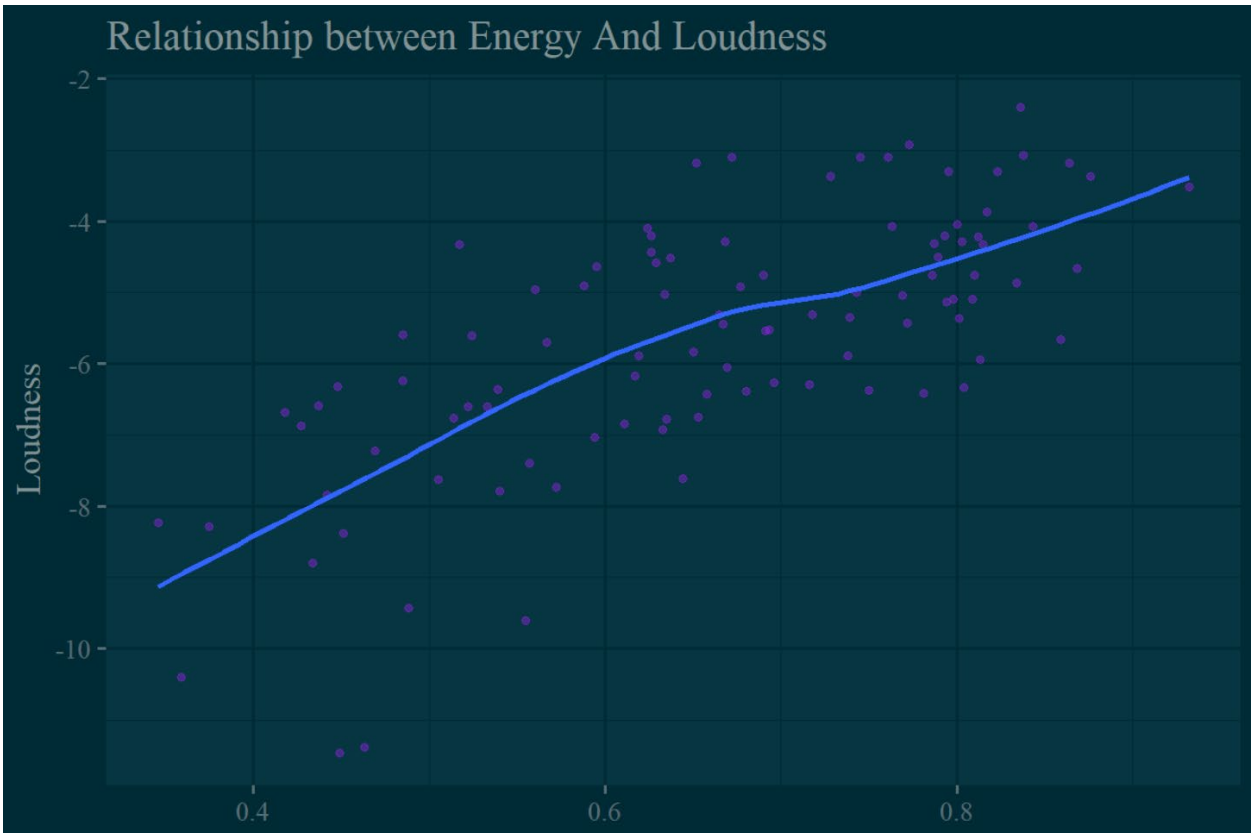
Since, energy and loudness are positively correlated, plotting the density of the 2 variables and the relation between the 2 variables.

Code in R

```
# Energy and Loudness most positively correlated,
# therefore we will plot the density of these 2 variables
# with the top 100 songs
ggplot(spotify_df) +
  geom_density(aes(x = loudness,fill = "loudness") ) +
  theme_solarized_2(light = FALSE,base_size = 15, base_family = "serif")

ggplot(spotify_df) +
  geom_density(aes(x = energy,fill = "energy") ) +
  theme_solarized_2(light = FALSE,base_size = 15, base_family = "serif")

#Relation between Energy and Loudness in the Top 10 Songs
ggplot(spotify_df) + aes(x= energy,y=loudness) +
  geom_point(color = "Purple",alpha = 0.4) + theme_light() +
  labs( x= "Energy",
        y = "Loudness",
        title = "Relationship between Energy And Loudness")+
  theme(axis.title.x = element_text(colour = "DarkGreen",size= 10),
        axis.title.y = element_text(colour = "DarkBlue",size= 10),
        axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 10),
        plot.title = element_text(color = "Brown",
                                  size = 15,
                                  family = "Arial",
                                  hjust = 0.5))+
  stat_smooth(se = FALSE) +
  theme_solarized_2(light = FALSE,base_size = 15, base_family = "serif")
```


0.20
0.15
0.10
0.05
0.00
density
-10
-8
-6
-4
-2
fill
loudness


2
1
0
density
0.4
0.6
0.8
fill
energy


Relationship between Energy And Loudness
Loudness
-2
-4
-6
-8
-10
0.4
0.6
0.8

Data Intepretation

Let's determine how the Key attribute affects the top 100 songs

Code in R

```
# CHECKING ON KEY VALUES
# Adding column based on other column:
spotify_df <- spotify_df %>%
  mutate(original_key = case_when(
    (key == "0") ~ "C",
    (key == "1") ~ "C#,Db",
    (key == "2") ~ "D",
    (key == "3") ~ "D#,Eb",
    (key == "4") ~ "E",
    (key == "5") ~ "F",
    (key == "6") ~ "F#,Gb",
    (key == "7") ~ "G",
    (key == "8") ~ "G#,A",
    (key == "9") ~ "A",
    (key == "10") ~ "A#,Bb",
    (key == "11") ~ "B"
  ))

# Checking which keys the top 100 songs use
# Find the top artist from the top 100 list
top_keys <- spotify_df %>%
  count( original_key)
top_keys <- top_keys %>%
  arrange(desc(n))
head(top_keys)

str(top_keys)

#setting the field 'n' as factor as the data can be in seen in descending order
top_keys$original_key <-
factor(top_keys$original_key,levels=top_keys$original_key[order(top_artists$n)])
```

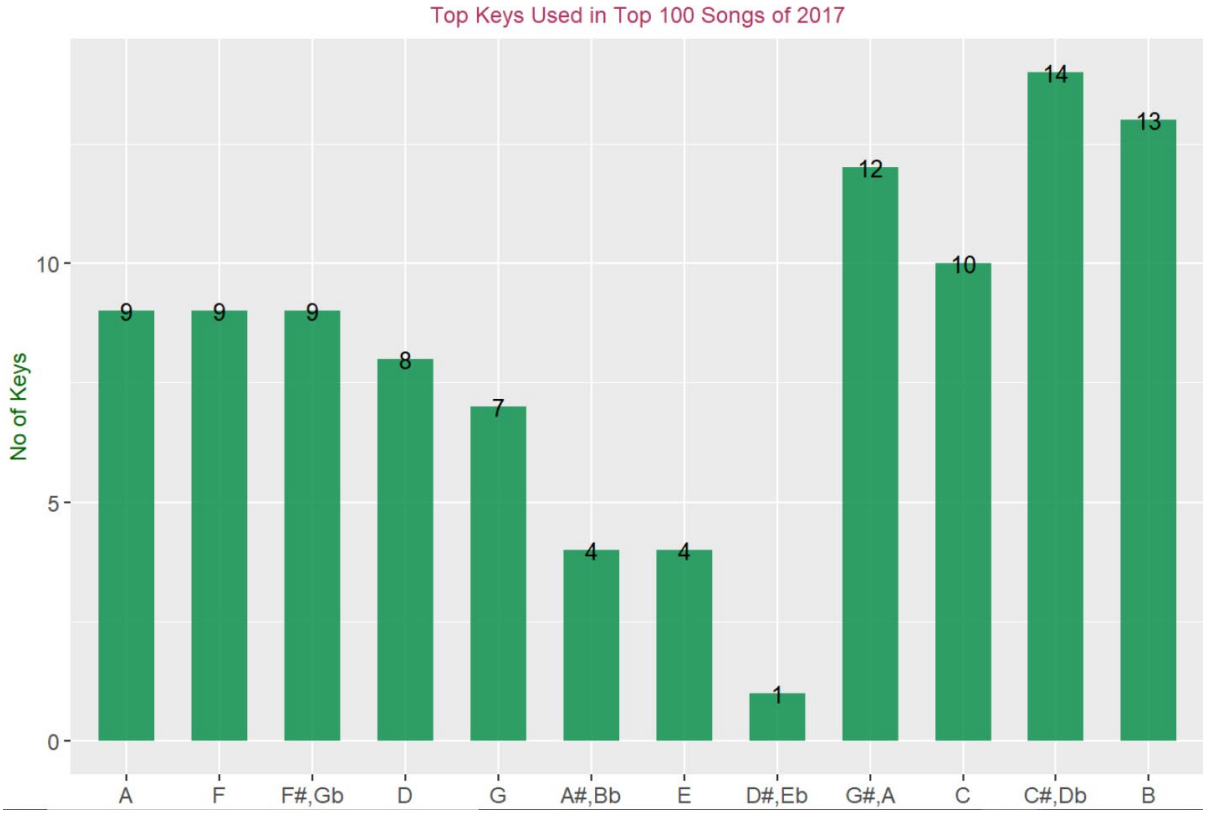
```
#Plotting the keys used in the top songs set
ggplot(top_keys, aes(x = original_key, y = n)) +
  geom_bar(stat='identity',fill = c("springgreen4"),alpha = 0.8,width = 0.6) +
  geom_text(aes(label=n), color = 'black') +
  ggtitle("Top Keys Used in Top 100 Songs of 2017") +
  labs(y = "No of Keys",x = "Keys") +
  theme(axis.title.x = element_text(colour = "DarkGreen",size= 10),
        axis.title.y = element_text(colour = "DarkGreen",size= 10),
        axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 10),
        plot.title = element_text(color = "Maroon",
                                  size = 10,
                                  family = "Arial",
                                  hjust = 0.5))
```

Output:

The below set shows the top 5 keys used by the Top 100 songs in the list. The graphical representation of the top 10 keys is shown

```
> # Checking which keys the top 100 songs use
> # Find the top artist from the top 100 list
> top_keys <- spotify_df %>%
+   count( original_key)
> top_keys <- top_keys %>%
+   arrange(desc(n))
> head(top_keys)
  original_key  n
1        C#,Db 14
2            B 13
3         G#,A 12
4            C 10
5            A  9
6            F  9
>
```

Graphical Representation of Top 10 Keys Used



Here, the maximum keys used by the top songs are C#, Db keys

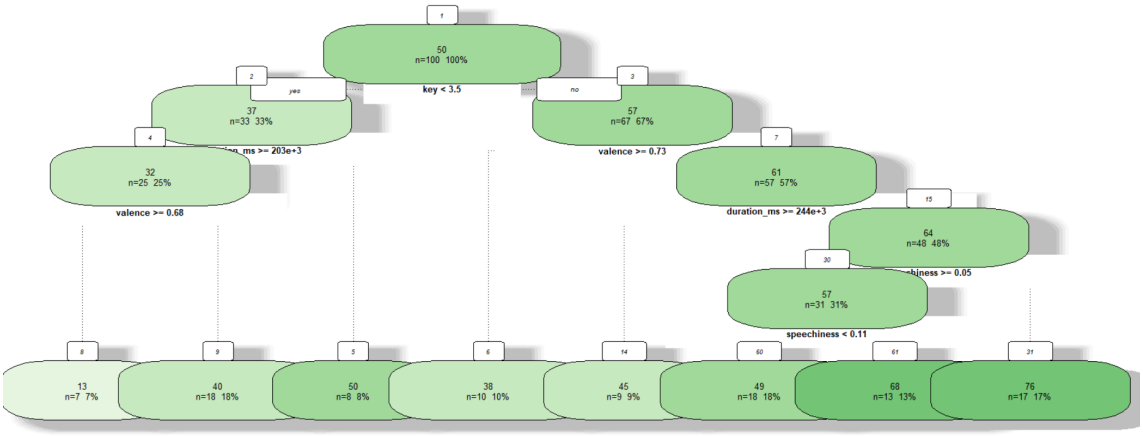
- What makes the Top 1 songs differ by the rest??

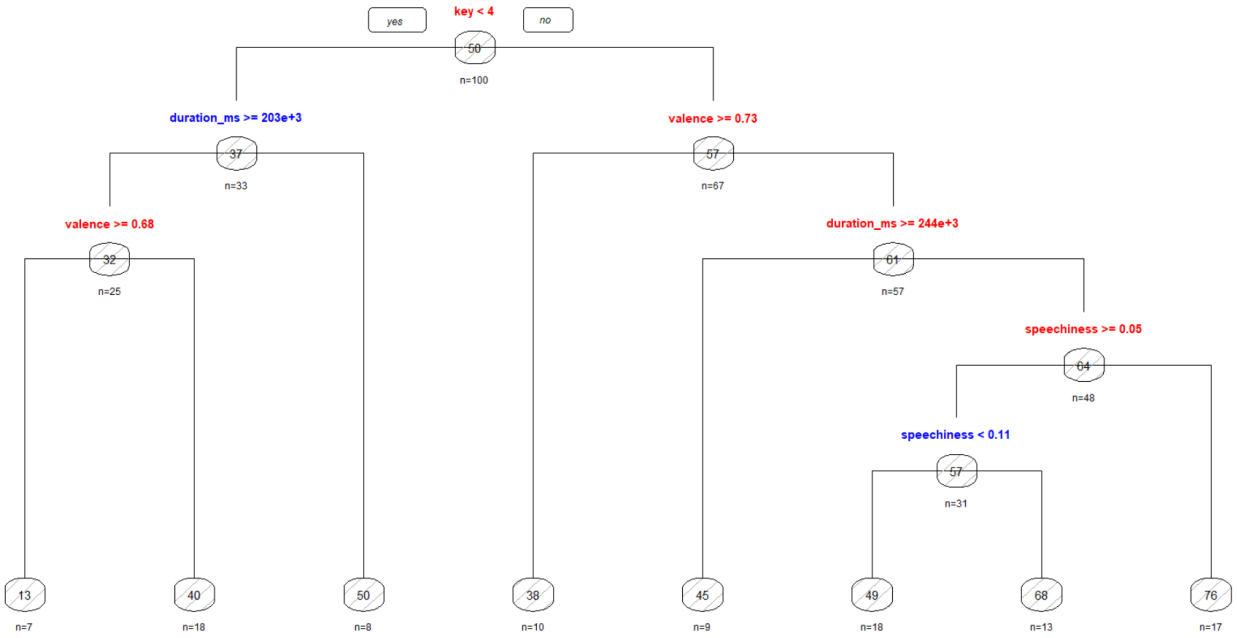
I have ranked the top songs based on their listing , so I will consider add this as a new field, rank and consider this field as the dependent variable for the analysis

Code in R

```
# Setting a new field, rank to set as the dependent variable
features_df$rank <- c(1:100)

# Using the dependent field, to check the decision tree model (classification model)
decision_tree <- rpart(rank ~ ., data = features_df)

#plotting the decision tree model
prp(decision_tree,type = 1,extra = 1,varlen = 10, under = T,fallen.leaves = F,
split.col = c("red","blue"), box.col = c("green","yellow")[decision_tree$frame$yval])

fancyRpartPlot(decision_tree,cex= 0.45)
```

Plotting the rpart graph





This show that for songs which have **keys** less than 4 (‘C’,’C#,Db’,’D’,’D#,Eb’) and **duration_ms** is more than 204 milliseconds and **valence** more than 0.68 have higher chances (18%) to be in the top 10

Inference

- ‘The Chainsmokers’ and ‘Ed Sheeran’ tops the list with 4 songs. Following are ‘Martin Garrix’ and ‘Drake’ with 3 songs each. The rest show the artists that have 2 songs in the top list

- ‘Ed Sheeran’ and ‘'Drake’ are the top artists with the top streamed time

- ‘Shape of You’ and ‘Despacito’ are the top songs with the highest streamed time. Both versions of Despacito songs can be seen in the top list

- Audio Features:
  - Maximum correlation can be seen for features – Loudness and Energy
  - Valence is positively correlated with loudness and danceability
  - Speechness and Loudness show a negative correlation between them

- Top Keys used by the Top Songs are C#,Db. The least key used is D#,Eb

- For songs, which have keys less than 4 (‘C’,’C#,Db’,’D’,’D#,Eb’) and duration_ms is more than 204 milliseconds and valence more than 0.68 have higher chances (18%) to be in the top 10.