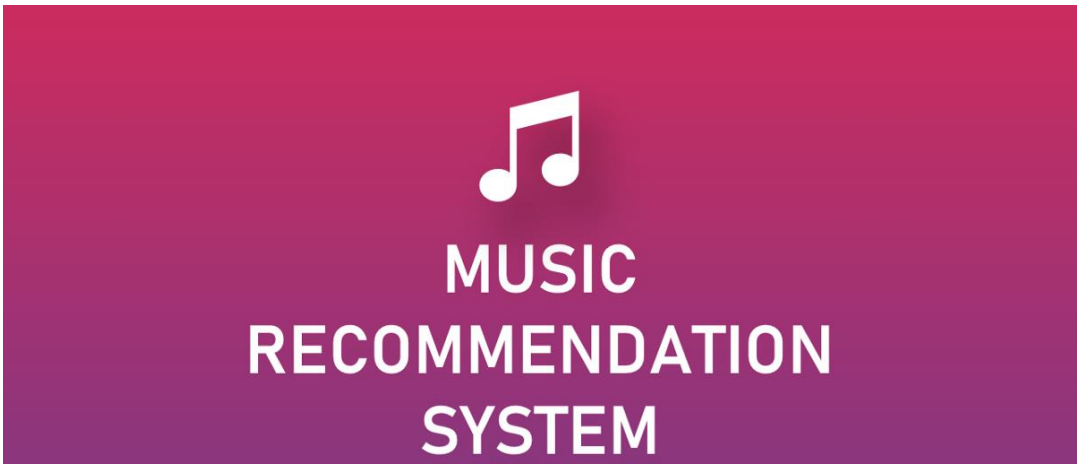
4th December 2022

Spotify Music Analytics and Recommendation



Team 7

Contents

1 Introduction

The Internet has changed not just the way we get our music, but what comes with it. As far as accessing music and info about your favourite songs, not much changed as phonographs gave way to vinyl records, and records, in turn, were replaced by CDs: you bought or borrowed a physical media with the music on it, and with the music, there was a tiny bit of information tucked away in the CD case or on the record's label. Usually, all it told you about the songs was limited to the song's name, who performed it, maybe the length of the song, and perhaps the artist's thoughts about the song, if the insert was detailed. If you listened to the radio, the DJ would say the song and the artist's name, maybe tell an amusing anecdote about the musician if there was time. But with the introduction of the internet and cloud storage, now music lives on servers.

The server hosts the songs and streams them out to thousands, or millions of people each day, with the goal that you continue listening. This is where you might have heard about the music streaming giant which has dominated the music market for so many years. Founded in April 2006, Spotify is a Swedish company that offers media streaming and other services. It is the biggest music streaming service in the world, with more than 381 million active users each month, including 172 million paying subscribers.

With this large active user base, the main goal of these companies is to have a greater user retention period. Eventually, the songs you already know you like run out, and that's when Spotify and other streaming services start suggesting things you might like. They analyze each song on their servers, gauging them on a slew of categories like the temp, is the song upbeat or slow, they even throw together some measurements of their own and how happy the song feels. With all this information, one can look for similarities in the songs that the user likes or from the user's playlist, and then Spotify will retrieve other similar songs that share those characteristics. They would then create a playlist or send a notification like, "*Recommended music as per your choices*" or *"Jump back in"*.

Today, data analysis is a major in many industries, including business, research, metrology, and many more. The data that is extracted from the databases is used to produce academic articles, forecast the weather, and many other things. The biggest audio streaming service in the world, Spotify, offers a variety of capabilities, including the ability to freely share music and view song lyrics as they are being played. With an ever-growing library and competitors striving to steal Spotify's crown as the number 1 streaming service. There's more music, and more data about music, out there than ever before and we as data analysts are trying to find interesting patterns with the music data.

With this in mind, a group of young and bubbling data scientists Rahul Chauhan, Nikhil Belavinakodige, Prasenjeet Gadhe, Annanya Jain, Marshall Pauley, Junaid Shaik, Siddharth Katti, Sanyukta Nair, Shubhankar Goje from the University of Colorado at Boulder are trying to find intriguing answers on how this shift to online and advancements in technology to find

how the music industry has involved over the past 20 years and how other aspects of music have evolved. To address some of these intriguing questions we will analyze the data, develop an algorithm, and present our findings.

2 Analyses

2.1 Data Discovery

Regardless of the subject of research, gathering data is the first and most crucial stage. Depending on the type of data needed, different disciplines of research require different approaches to data gathering.

There are around 25 attributes to find insights from our given dataset. Let's talk about what it individually means in brief:

1. Song ID: ID of the song on Spotify.
2. Song Name: Title of the song.
3. Album ID: ID of the album in which the song is present.
4. Artist: Name of the Artist.
5. Genre: Genre of the song.
6. Artist ID: ID of the Artist of the song.
7. Track Number: The order in which the song appeared on its album.
8. Disc Number: It indicates the number on the disc in which the album is present.
9. Explicit: Indicates whether the song has explicit content or not.
10. Danceability: Indicates whether the song is appropriate for dancing. The least danceable number is 0.0, and the most danceable number is 1.0.
11. Energy: Represents a perceptual measure of intensity and activity. Typically, energetic songs are loud, fast, and noisy. It has values between 0.0 and 1.0, with 1.0 being the most energetic.
12. Key: Estimated overall key of the song. Integers are mapped based on the standard pitch class notation. Eg: 0=C,2=D,etc.
13. Loudness: overall loudness of the track in decibels. Values range from -60 to 0 dB.
14. Mode: It indicates the modality of the track(whether major or minor), with 1 being a Major and 0 for a Minor.
15. Speechiness: Detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audiobook, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech. Values below 0.33 most likely represent music and other non-speechlike tracks.
16. Acousticness: Measure from 0.0 to 1.0 whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
17. Instrumentalness: Measure whether a track contains no vocals. 'Ooh' and 'aah' sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly 'vocal'.

The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content.

18. Liveness: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides a strong likelihood that the track is live.
19. Valence: Those with a high valence sound happier, cheerier, and more euphoric, whilst tracks with a low valence sound more depressing.
20. Tempo: The 'speed' of the song. Tempo is how fast the beats come after each other.
21. Duration in ms: Duration of the song in milliseconds.
22. Time signature: Time signature is a glorified fraction that guides musicians on how to read music written on a score. The top number tells you how many beats there are per measure, while the bottom number indicates which note is a full beat long. Typical time signatures include 4/4 time, with 4 beats per measure and a quarter note representing the beat. Spotify handles this strangely, presenting the time signature as a single integer, probably because the time signature is a tool for reading sheet music, which Spotify doesn't handle.
23. Year: The year in which the song is released.
24. Season of Release: The season in which the song is released e.g.: Fall, spring, etc.
25. Release Date: The date on which the song is released.
26. Rating: User-given rating of the song.

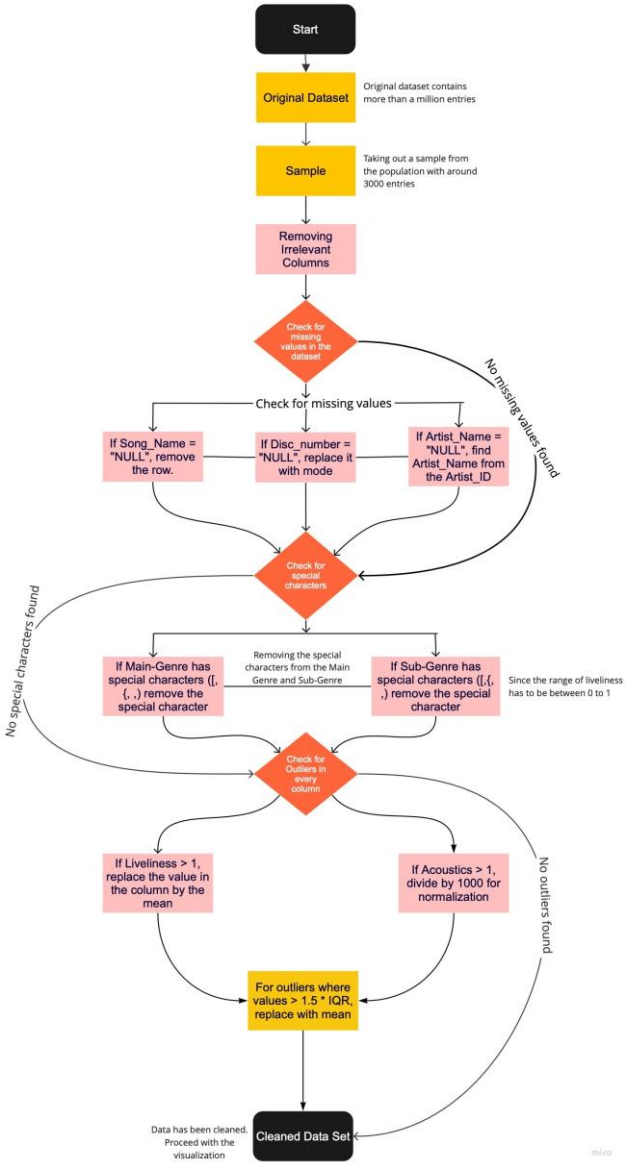
To find insights from our dataset, and to discover more about the characteristics of the songs, and how it plays an important role in answering our questions. These are the list of the questions we wish to answer through data discovery and the EDA process:

1. Will a song with explicit lyrics or curse words become more successful in terms of likability than songs without such content?
2. What is the likelihood that the song will be danceable if it's highly acoustic?
3. Depending on the time of day, which song will be appealing to me?
4. Can the genre of a song help us forecast how intense it will be?
5. Who will win the Eurovision winner's average?
6. If I publish a song to Spotify, what do I need to ensure that it's a hit or likeable by my audience?
7. How has the musical genre evolved over the last twenty years?
8. Which genre do people prefer?
9. What kind of song will lift your spirits?
10. Myth or Truth: If a chord of music is lower (P.S. Chord/Key ranges from 0-10), will the song be sadder?

2.2 Data Cleaning

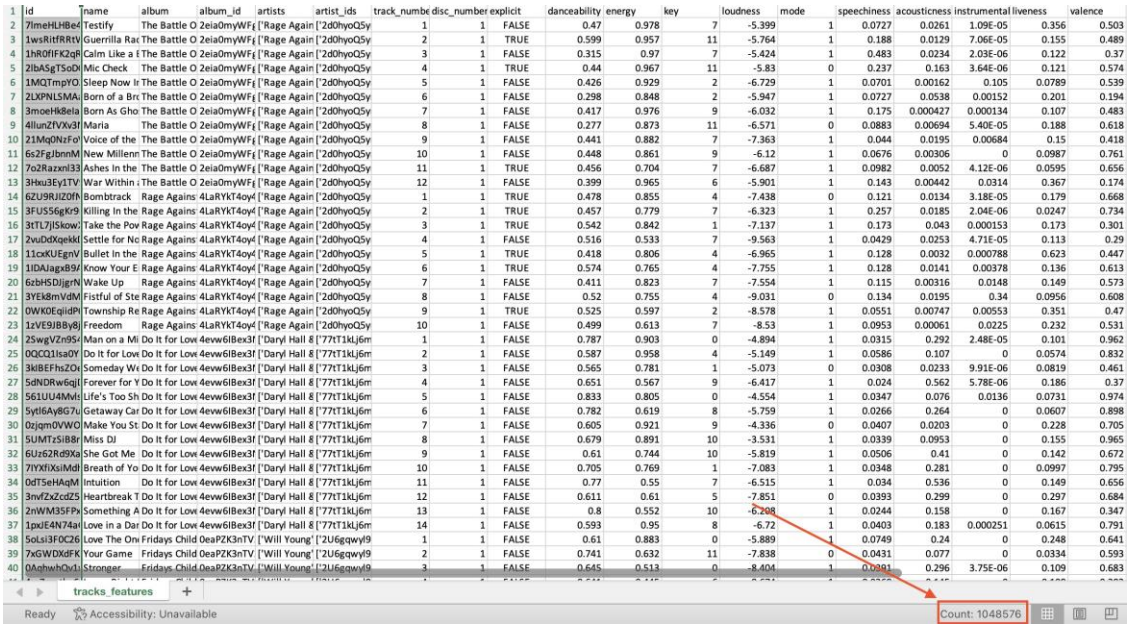
2.2.1 Algorithm

Algorithms are like the recipe you require for cooking. To proceed with data discovery, these are certain steps or procedures are followed in this project. By following the plan laid out below, one can find answers to our questions:

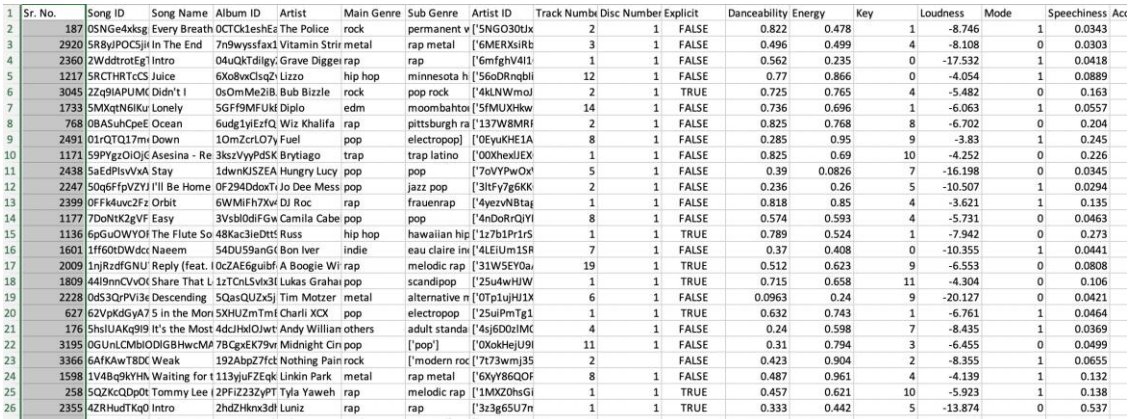


2.2.2 Sampling

The original dataset contains more than a million entries and 24 columns which is the population for this dataset and the sample contains 3200 entries. The below figure shows the data entries before the sampling.



To analyze, discover patterns and finding answers to the asked questions, a subset of data was taken to analyze. The subset comprises of 3200 entries and 27 columns in the dataset.



2.2.3 Data Cleaning

The title of this sample is not in the proper format.

```
## # A tibble: 3,402 x 28
##    `Sr. No.` `Song ID`    Song ~1 Album~2 Artist Main ~3 Sub G~4 Artis~5 Track~6
##        <dbl> <chr>        <chr>   <chr>   <chr>  <chr>   <chr>   <chr>     <dbl>
##  1       187 0SNGe4xksgp~ Every ~ 0CTCk1~ The P~ ['rock] perman~ ['5NGO~       2
##  2      2920 5R8yJPOC5ji~ In The~ 7n9wys~ Vitam~ ['meta~ rap me~ ['6MER~       3
##  3      2360 2WddtrotEgT~ Intro   04uQkT~ Grave~ ['rap]  rap     ['6mfg~       1
##  4      1217 5RCTHRTcCSv~ Juice   6Xo8vx~ Lizzo  ['hip ~ minnes~ ['56oD~      12
##  5      3045 2Zq9IAPUMOr~ Didn't~ 0sOmMe~ Bub B~ ['rock] pop ro~ ['4kLN~       2
##  6      1733 5MXqtN6IKuv~ Lonely  5GFf9M~ Diplo  ['edm]  moomba~ ['5fMU~      14
##  7       768 0BASuhCpeEy~ Ocean   6udg1y~ Wiz K~ ['rap]  pittsb~ ['137W~       2
##  8      2491 01rQTQ17meS~ Down    1OmZcr~ Fuel   ['pop]  electr~ ['0Eyu~       8
##  9      1171 59PYgzOiOjG~ Asesin~ 3kszVy~ Bryti~ ['trap] trap l~ ['00Xh~       1
## 10      2438 5aEdPIsvVxA~ Stay    1dwnKJ~ Hungr~ ['pop]  pop     ['7oVY~       5
## # ... with 3,392 more rows, 19 more variables: `Disc Number` <dbl>,
## #   Explicit <lgl>, 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>, Year <dbl>,
## #   `Season of Release` <chr>, `Release Date` <chr>, `Rating %` <dbl>, and
## #   abbreviated variable names 1: `Song Name`, 2: `Album ID`, ...
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Columns are renamed using function clean name from janitor package.

```
##  [1] "sr_no"              "song_id"            "song_name"
##  [4] "album_id"           "artist"             "main_genre"
##  [7] "sub_genre"          "artist_id"          "track_number"
## [10] "disc_number"        "explicit"           "danceability"
## [13] "energy"             "key"                "loudness"
## [16] "mode"               "speechiness"        "acousticness"
## [19] "instrumentalness"   "liveness"           "valence"
## [22] "tempo"              "duration_ms"        "time_signature"
## [25] "year"               "season_of_release" "release_date"
## [28] "rating_percent"
```

Obtaining a summary of the sample:

```
##    song_id           song_name           album_id            artist
##  Length:3402        Length:3402        Length:3402        Length:3402
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   main_genre         sub_genre          artist_id          track_number
##  Length:3402        Length:3402        Length:3402        Min.   : 1.000
##  Class :character   Class :character   Class :character   1st Qu.: 2.000
##  Mode  :character   Mode  :character   Mode  :character   Median : 5.000
##                                                           Mean   : 6.352
##                                                           3rd Qu.: 9.000
##                                                           Max.   :50.000
##                                                           NA's   :8
##   disc_number     explicit        danceability        energy
##  Min.   :1.000   Mode :logical   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.000   FALSE:2637      1st Qu.:0.5020   1st Qu.:0.4730
##  Median :1.000   TRUE :735       Median :0.6355   Median :0.6270
##  Mean   :1.013   NA's :30        Mean   :0.6166   Mean   :0.6071
##  3rd Qu.:1.000                   3rd Qu.:0.7470   3rd Qu.:0.7760
##  Max.   :5.000                   Max.   :0.9800   Max.   :0.9980
##  NA's   :65                      NA's   :30       NA's   :31
##       key            loudness            mode        speechiness
##  Min.   : 0.000   Min.   :-60.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.: 2.000   1st Qu.: -9.312   1st Qu.:0.000   1st Qu.:0.0370
##  Median : 5.000   Median : -6.851   Median :1.000   Median :0.0560
##  Mean   : 5.078   Mean   : -7.899   Mean   :0.628   Mean   :0.1117
##  3rd Qu.: 8.000   3rd Qu.: -5.252   3rd Qu.:1.000   3rd Qu.:0.1343
##  Max.   :11.000   Max.   :  1.908   Max.   :1.000   Max.   :0.9660
##  NA's   :31       NA's   :48        NA's   :55      NA's   :66
##   acousticness      instrumentalness     liveness          valence
##  Min.   :  0.0000   Min.   :0.00000   Min.   : 0.0000   Min.   :0.0000
##  1st Qu.:  0.0258   1st Qu.:0.00000   1st Qu.: 0.0983   1st Qu.:0.2850
##  Median :  0.1340   Median :0.00000   Median : 0.1290   Median :0.4635
##  Mean   :  2.5259   Mean   :0.08501   Mean   : 0.2419   Mean   :0.4739
##  3rd Qu.:  0.4470   3rd Qu.:0.00250   3rd Qu.: 0.2460   3rd Qu.:0.6520
##  Max.   :995.0000   Max.   :0.99700   Max.   :13.2000   Max.   :0.9780
##  NA's   :41         NA's   :41        NA's   :33        NA's   :34
##      tempo         duration_ms      time_signature       year
##  Min.   :  0.00   Min.   :   8400   Min.   :0.000   Min.   :1936
##  1st Qu.: 96.49   1st Qu.: 177391   1st Qu.:4.000   1st Qu.:2004
##  Median :118.57   Median : 208780   Median :4.000   Median :2017
##  Mean   :120.59   Mean   : 215725   Mean   :3.907   Mean   :2011
##  3rd Qu.:141.17   3rd Qu.: 243714   3rd Qu.:4.000   3rd Qu.:2019
##  Max.   :216.33   Max.   :3605720   Max.   :5.000   Max.   :2020
##  NA's   :34       NA's   :34        NA's   :23      NA's   :23
##  season_of_release  release_date       rating_percent
##  Length:3402        Length:3402        Min.   : 0.00
##  Class :character   Class :character   1st Qu.:20.00
```

2.2.4 Removing Irrelevant Columns

Song id will work as the Primary key, thus Sr. No. is irrelevant.

```
df = subset(df, select = -sr_no )
col_names <- colnames(df)
```

2.2.5 Missing values in the sample:

Checking for NULL values in the CSV File using the following syntax:

df% > %summarise(across(everything(); sum(is:na(:))))

```
## # A tibble: 1 x 27
##   song_id song_~1 album~2 artist main_~3 sub_g~4 artis~5 track~6 disc_~7 expli~8
##     <int>   <int>   <int>  <int>   <int>   <int>   <int>   <int>   <int>   <int>
## 1       0     172       0     28       0       0       0       8      65      30
## # ... with 17 more variables: danceability <int>, energy <int>, key <int>,
## #   loudness <int>, mode <int>, speechiness <int>, acousticness <int>,
## #   instrumentalness <int>, liveness <int>, valence <int>, tempo <int>,
## #   duration_ms <int>, time_signature <int>, year <int>,
## #   season_of_release <int>, release_date <int>, rating_percent <int>, and
## #   abbreviated variable names 1: song_name, 2: album_id, 3: main_genre,
## #   4: sub_genre, 5: artist_id, 6: track_number, 7: disc_number, ...
## # i Use `colnames()` to see all variable names
```

From the above table, one can infer that column like song name, track number, explicit, and danceability are not dependent on any other column. Moreover, replacing danceability with its mean might not give desired results hence it is safer to remove them. But the artist's name might be dependent on the artist_id. Also, one can substitute the disc number with the most frequent disc number.

Song Name:
One can see song names have null values. Those rows which don't have song name details cannot be replaced by any other qualitative data. Hence removing all rows that contain NA.

```
## # A tibble: 3,207 x 27
##    song_id        song_~1 album~2 artist main_~3 sub_g~4 artis~5 track~6 disc_~7
##    <chr>          <chr>   <chr>   <chr>  <chr>   <chr>   <chr>     <dbl> <chr>
##  1 0SNGe4xksgp7C~ Every ~ 0CTCk1~ The P~ ['rock] perman~ ['5NGO~       2 1
##  2 5R8yJPOC5jiOF~ In The~ 7n9wys~ Vitam~ ['meta~ rap me~ ['6MER~       3 1
##  3 2WddtrotEgTQm~ Intro   04uQkT~ Grave~ ['rap]  rap     ['6mfg~       1 1
```

```
##  4 5RCTHRTcCSvkb~ Juice   6Xo8vx~ Lizzo  ['hip ~ minnes~ ['56oD~      12 1
##  5 2Zq9IAPUMOrMq~ Didn't~ 0sOmMe~ Bub B~ ['rock] pop ro~ ['4kLN~       2 1
##  6 5MXqtN6IKuvnN~ Lonely  5GFf9M~ Diplo  ['edm]  moomba~ ['5fMU~      14 1
##  7 0BASuhCpeEyBE~ Ocean   6udg1y~ Wiz K~ ['rap]  pittsb~ ['137W~       2 1
##  8 01rQTQ17meSwW~ Down    1OmZcr~ Fuel   ['pop]  electr~ ['0Eyu~       8 1
##  9 59PYgzOiOjGDz~ Asesin~ 3kszVy~ Bryti~ ['trap] trap l~ ['00Xh~       1 1
## 10 5aEdPIsvVxAAL~ Stay    1dwnKJ~ Hungr~ ['pop]  pop     ['7oVY~       5 1
## # ... with 3,197 more rows, 18 more variables: explicit <lgl>,
## #   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>, year <dbl>, season_of_release <chr>,
## #   release_date <chr>, rating_percent <dbl>, and abbreviated variable names
## #   1: song_name, 2: album_id, 3: main_genre, 4: sub_genre, 5: artist_id, ...
## # i Use 'print(n = ...)' to see more rows, and 'colnames()' to see all variable names
```

2.2.6 Disk Number

Column disc Number column has NA. So, replacing NA with Mode in case of disc number.

```
df <-
df %>%
  mutate(disc_number = ifelse(is.na(disc_number),
                              mode(disc_number),
                              disc_number))
```

Checking for the sample again for any variations and it's clear that there are no NAs in the sample.

```
## # A tibble: 1 x 27
##   song_id song_~1 album~2 artist main_~3 sub_g~4 artis~5 track~6 disc_~7 expli~8
##     <int>   <int>   <int>  <int>   <int>   <int>   <int>   <int>   <int>   <int>
## 1       0       0       0      0       0       0       0       0       0       0
## # ... with 17 more variables: danceability <int>, energy <int>, key <int>,
## #   loudness <int>, mode <int>, speechiness <int>, acousticness <int>,
## #   instrumentalness <int>, liveness <int>, valence <int>, tempo <int>,
## #   duration_ms <int>, time_signature <int>, year <int>,
## #   season_of_release <int>, release_date <int>, rating_percent <int>, and
## #   abbreviated variable names 1: song_name, 2: album_id, 3: main_genre,
## #   4: sub_genre, 5: artist_id, 6: track_number, 7: disc_number, ...
## # i Use 'colnames()' to see all variable names
```

2.2.7 Check for Special Characters

Having a look at the column Subgenre and Main Genre one can notice that there are some unnecessary characters in genre names like [,',]. So, these are removed.

Main Genre:

```
# to remove "['" in genre column at the front
df$main_genre <- str_remove(df$main_genre, pattern = "^\\['")
# to remove "]" in genre column at the back
df$main_genre  <- gsub("]","",as.character(df$main_genre))
```

Sub-Genre:

```
# to remove "['" in genre column at the front
df$sub_genre <- str_remove(df$sub_genre, pattern = "^\\['")
# to remove "]" in genre column at the back
df$sub_genre  <- gsub("]","",as.character(df$sub_genre))
```

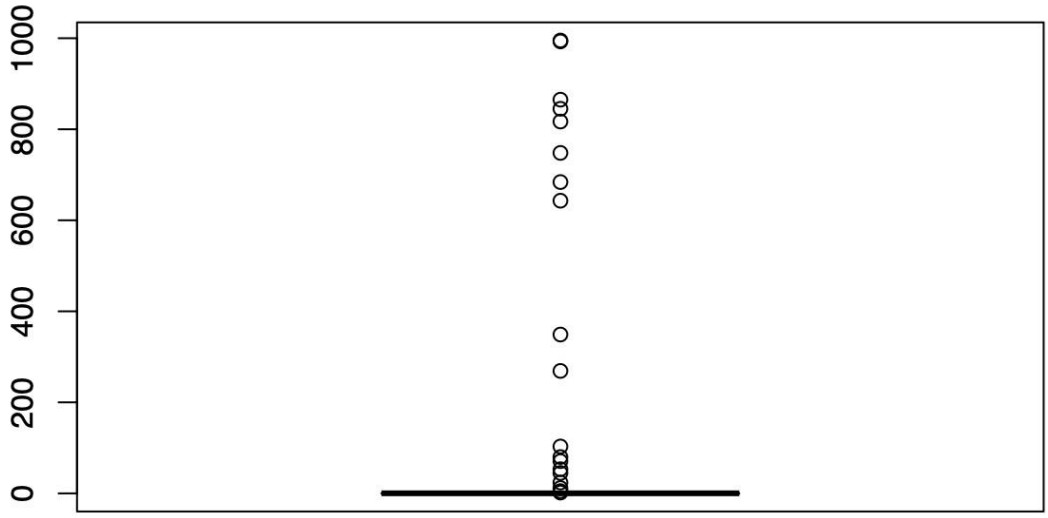
2.2.8 Check for Outliers

Acoustics: When plotting the summary of our data frame, it catches one's eye that the maximum value of acousticness is very high which is unusual. (Acoustics can range anywhere from 0 to 1).

```
 acousticness
Min.   :  0.0000
1st Qu.:  0.0269
Median :  0.1330
Mean   :  2.6328
3rd Qu.:  0.4420
Max.   :995.0000
```

Trying to understand this distribution better using a box plot:

Box Plot of Acousticness



Finding the rows where Acousticness is greater than 1:

```
## # A tibble: 19 x 27
##    acousticness song_id   song_~1 album~2 artist main_~3 sub_g~4 artis~5 track~6
##           <dbl> <chr>     <chr>   <chr>   <chr>  <chr>   <chr>   <chr>     <dbl>
##  1       865    06l7JCG7~ Only H~ 2jccmY~ The H~ pop     post-t~ ['0xm0~       2
##  2       643    7cqq5ebd~ Freedom 1cJccL~ Melis~ house   tropic~ ['6t9I~       1
##  3        70.8  0nGvVryR~ Angel   4NOioj~ Katha~ pop     pop'    ['5VXt~       7
##  4        11.7  6KM8gsHD~ The Ot~ 70L3Sn~ Galax~ others  hollyw~ ['6guT~      12
##  5       349    6RoModa3~ Smells~ 3iVI9k~ The B~ rock    grunge' ['5qOf~       3
##  6        23.9  5uLIEYV9~ Want Y~ 4NOioj~ Katha~ pop     metrop~ ['5VXt~       8
##  7       684    2CGz42nO~ Unforg~ 5ZzzZs~ Peppe~ rap     rap'    ['0lnN~       1
##  8       845    3FO0qjXm~ Touch   7e3ev6~ Tribe~ others  talent~ ['74TK~       5
##  9        53.1  1Aw4vFcR~ I Want~ 32Kacw~ Zombi~ rap     melodi~ ['4BxR~       4
```

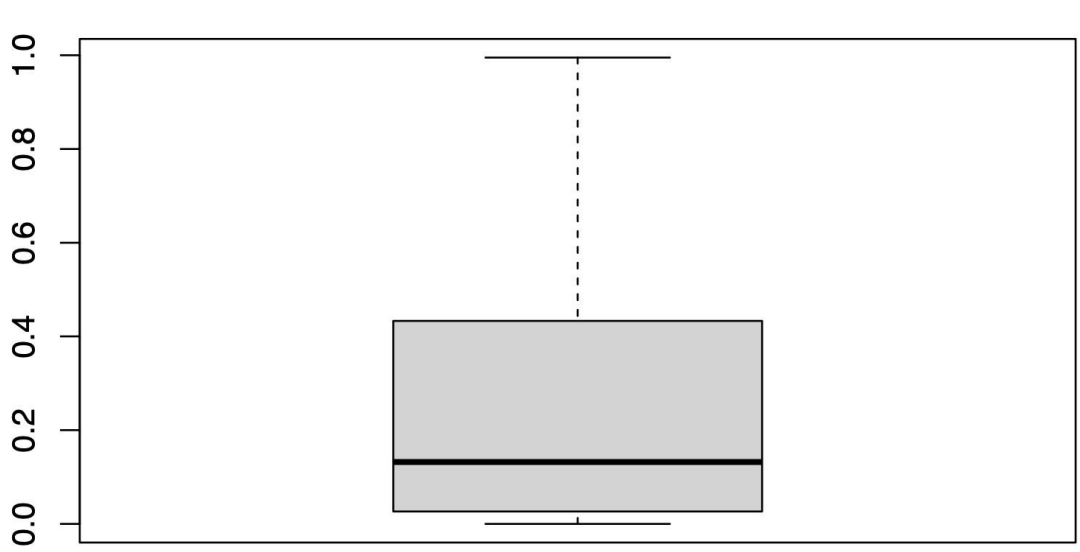
Since the acousticness can't be greater than 1, the values might be due to incorrect data. Dividing the acousticness by 1000 and get values between 0 and 1.

```
df <-
df %>%
  mutate(acousticness = ifelse(acousticness>1,
                               acousticness/1000,
                               acousticness))
```

After normalizing the acousticness attribute in our dataset, let's re-display the summary of our dataset – acousticness

```
  acousticness
Min.   :0.00000
1st Qu.:0.02645
Median :0.13200
Mean   :0.26482
3rd Qu.:0.43300
Max.   :0.99500
```

Box plot for Acousticness after normalization:



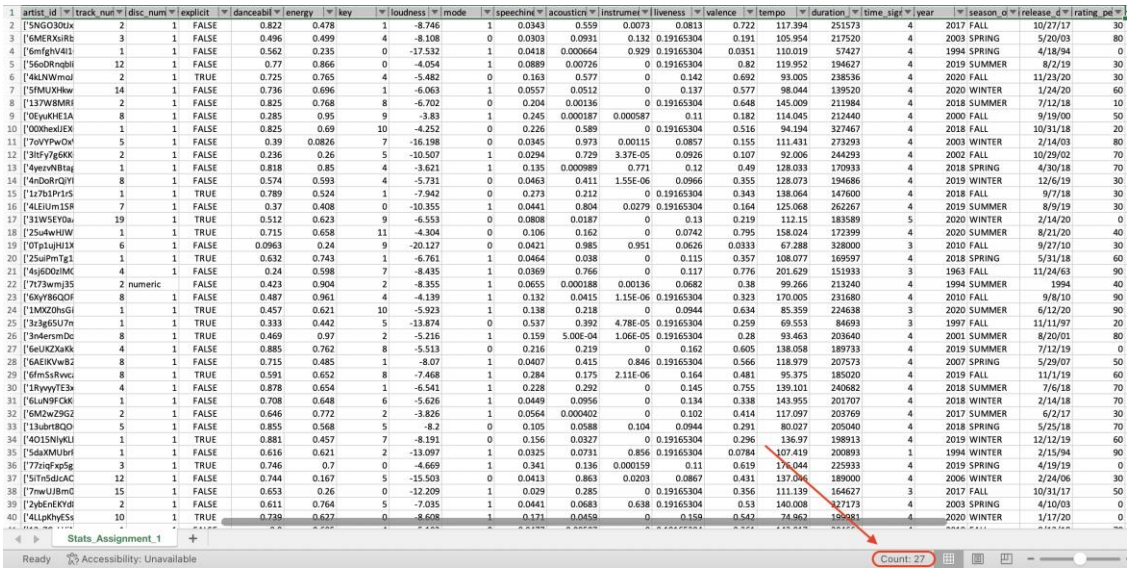
2.2.9 Cleaned Dataset

Checking the summary to check the validity of data for all columns.

```
##    song_id          song_name          album_id          artist
```

```
##  Length:3207        Length:3207        Length:3207        Length:3207
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   main_genre         sub_genre          artist_id          track_number
##  Length:3207        Length:3207        Length:3207        Min.   : 1.000
##  Class :character   Class :character   Class :character   1st Qu.: 2.000
##  Mode  :character   Mode  :character   Mode  :character   Median : 5.000
##                                                           Mean   : 6.325
##                                                           3rd Qu.: 9.000
##                                                           Max.   :50.000
##  disc_number         explicit        danceability        energy
##  Length:3207        Mode :logical   Min.   :0.0000   Min.   :0.0000
##  Class :character   FALSE:2492      1st Qu.:0.5080   1st Qu.:0.4760
##  Mode  :character   TRUE :715       Median :0.6390   Median :0.6260
##                                     Mean   :0.6213   Mean   :0.6072
##                                     3rd Qu.:0.7505   3rd Qu.:0.7740
##                                     Max.   :0.9800   Max.   :0.9980
##       key            loudness            mode          speechiness
##  Min.   : 0.000   Min.   :-60.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 2.000   1st Qu.: -9.159   1st Qu.:0.0000   1st Qu.:0.0372
##  Median : 5.000   Median : -6.799   Median :1.0000   Median :0.0566
##  Mean   : 5.097   Mean   : -7.833   Mean   :0.6261   Mean   :0.1121
##  3rd Qu.: 8.000   3rd Qu.: -5.216   3rd Qu.:1.0000   3rd Qu.:0.1370
##  Max.   :11.000   Max.   :  1.908   Max.   :1.0000   Max.   :0.9660
##   acousticness     instrumentalness       liveness          valence
##  Min.   :0.00000   Min.   :0.0000000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.02645   1st Qu.:0.0000000   1st Qu.:0.09775   1st Qu.:0.2865
##  Median :0.13200   Median :0.0000037   Median :0.12800   Median :0.4640
##  Mean   :0.26482   Mean   :0.0780404   Mean   :0.13624   Mean   :0.4751
##  3rd Qu.:0.43300   3rd Qu.:0.0019450   3rd Qu.:0.19165   3rd Qu.:0.6520
##  Max.   :0.99500   Max.   :0.9970000   Max.   :0.20600   Max.   :0.9780
##      tempo          duration_ms      time_signature       year
##  Min.   :  0.00   Min.   :   8400   Min.   :0.000   Min.   :1936
##  1st Qu.: 96.43   1st Qu.: 177212   1st Qu.:4.000   1st Qu.:2005
##  Median :118.45   Median : 207760   Median :4.000   Median :2017
##  Mean   :120.58   Mean   : 214812   Mean   :3.908   Mean   :2012
##  3rd Qu.:141.17   3rd Qu.: 242094   3rd Qu.:4.000   3rd Qu.:2019
##  Max.   :216.33   Max.   :3605720   Max.   :5.000   Max.   :2020
##  season_of_release  release_date       rating_percent
##  Length:3207        Length:3207        Min.   : 0.00
##  Class :character   Class :character   1st Qu.:20.00
##  Mode  :character   Mode  :character   Median :40.00
##                                        Mean   :44.74
##                                        3rd Qu.:70.00
##                                        Max.   :90.00
```

After cleaning the dataset, one can understand that there were about 200 entries that were cleaned using R and this cleaned dataset is then saved to a new sample file.



	artist_id	track_num	disc_num	explicit	danceabil	energy	key	loudness	mode	speechine	acousticn	instrumer	liveness	valence	tempo	duration_	time_sigr	year	season_d	release_d	rating_pe
2	['5NGO30Ux	2	1	FALSE	0.822	0.478	1	-8.746	1	0.0343	0.559	0.0073	0.0813	0.722	117.394	251573	4	2017	FALL	10/27/17	30
3	['6MERXsiRb	3	1	FALSE	0.496	0.499	4	-8.108	0	0.0303	0.0931	0.132	0.19165304	0.191	105.954	217520	4	2003	SPRING	5/20/03	80
4	['6mfgNV4I3	1	1	FALSE	0.562	0.235	0	-17.532	1	0.0418	0.000664	0.929	0.19165304	0.0351	110.019	57427	4	1994	SPRING	4/18/94	0
5	['56oDRnqbI	12	1	FALSE	0.77	0.866	0	-4.054	1	0.0889	0.00726	0	0.19165304	0.82	119.952	194627	4	2019	SUMMER	8/2/19	30
6	['4kLNWmoJ	2	1	TRUE	0.725	0.765	4	-5.482	0	0.163	0.577	0	0.142	0.692	93.005	238536	4	2020	FALL	11/23/20	30
7	['5fMUXHkw	14	1	FALSE	0.736	0.696	1	-6.063	1	0.0557	0.0512	0	0.137	0.577	98.044	139520	4	2020	WINTER	1/24/20	60
8	['137W8MRI	2	1	FALSE	0.825	0.768	8	-6.702	0	0.204	0.00136	0	0.19165304	0.648	145.009	211984	4	2018	SUMMER	7/12/18	10
9	['0EyuKHE1A	8	1	FALSE	0.285	0.95	9	-3.83	1	0.245	0.000187	0.000587	0.11	0.182	114.045	212440	4	2000	FALL	9/19/00	50
10	['00XhexJEX	1	1	FALSE	0.825	0.69	10	-4.252	0	0.226	0.589	0	0.19165304	0.516	94.194	327467	4	2018	FALL	10/31/18	20
11	['7oVYPwOx	5	1	FALSE	0.39	0.0826	7	-16.198	0	0.0345	0.973	0.00115	0.0857	0.155	111.431	273293	4	2003	WINTER	2/14/03	80
12	['3ltFy7g6KK	2	1	FALSE	0.236	0.26	5	-10.507	1	0.0294	0.729	3.37E-05	0.0926	0.107	92.006	244293	4	2002	FALL	10/29/02	70
13	['4yezvNBtaq	1	1	FALSE	0.818	0.85	4	-3.621	1	0.135	0.000989	0.771	0.12	0.49	128.033	170933	4	2018	SPRING	4/30/18	70
14	['4nDoRrQiYI	8	1	FALSE	0.574	0.593	4	-5.731	0	0.0463	0.411	1.55E-06	0.0966	0.355	128.073	194686	4	2019	WINTER	12/6/19	30
15	['1z7b1Pr1rS	1	1	TRUE	0.789	0.524	1	-7.942	0	0.273	0.212	0	0.19165304	0.343	138.064	147600	4	2018	FALL	9/7/18	30
16	['4LEiUm1SR	7	1	FALSE	0.37	0.408	0	-10.355	1	0.0441	0.804	0.0279	0.19165304	0.164	125.068	262267	4	2019	SUMMER	8/9/19	30
17	['31W5EY0a	19	1	TRUE	0.512	0.623	9	-6.553	0	0.0808	0.0187	0	0.13	0.219	112.15	183589	5	2020	WINTER	2/14/20	0
18	['25u4wHIW	1	1	TRUE	0.715	0.658	11	-4.304	0	0.106	0.162	0	0.0742	0.795	158.024	172399	4	2020	SUMMER	8/21/20	40
19	['0Tp1ujHJ1X	6	1	FALSE	0.0963	0.24	9	-20.127	0	0.0421	0.985	0.951	0.0626	0.0333	67.288	328000	3	2010	FALL	9/27/10	30
20	['25uiPmTg1	1	1	TRUE	0.632	0.743	1	-6.761	1	0.0464	0.038	0	0.115	0.357	108.077	169597	4	2018	SPRING	5/31/18	60
21	['4sj6D0zIMC	4	1	FALSE	0.24	0.598	7	-8.435	1	0.0369	0.766	0	0.117	0.776	201.629	151933	3	1963	FALL	11/24/63	90
22	['7t73wmj35	2	numeric	FALSE	0.423	0.904	2	-8.355	1	0.0655	0.000188	0.00136	0.0682	0.38	99.266	213240	4	1994	SUMMER	1994	40
23	['6XyY86QOf	8	1	FALSE	0.487	0.961	4	-4.139	1	0.132	0.0415	1.15E-06	0.19165304	0.323	170.005	231680	4	2010	FALL	9/8/10	90
24	['1MXZ0hsGi	1	1	TRUE	0.457	0.621	10	-5.923	1	0.138	0.218	0	0.0944	0.634	85.359	224638	3	2020	SUMMER	6/12/20	90
25	['3z3g65U7n	1	1	TRUE	0.333	0.442	5	-13.874	0	0.537	0.392	4.78E-05	0.19165304	0.259	69.553	84693	3	1997	FALL	11/11/97	20
26	['3n4ersmDc	8	1	TRUE	0.469	0.97	2	-5.216	1	0.159	5.00E-04	1.06E-05	0.19165304	0.28	93.463	203640	4	2001	SUMMER	8/20/01	80
27	['6eUKZXaKk	4	1	FALSE	0.885	0.762	8	-5.513	0	0.216	0.219	0	0.162	0.605	138.058	189733	4	2019	SUMMER	7/12/19	0
28	['6AEIKVwB2	8	1	FALSE	0.715	0.485	1	-8.07	1	0.0407	0.415	0.846	0.19165304	0.566	118.979	207573	4	2007	SPRING	5/29/07	50
29	['6fmSsRvvc	8	1	TRUE	0.591	0.652	8	-7.468	1	0.284	0.175	2.11E-06	0.164	0.481	95.375	185020	4	2019	FALL	11/1/19	60
30	['1RyvyyTE3x	4	1	FALSE	0.878	0.654	1	-6.541	1	0.228	0.292	0	0.145	0.755	139.101	240682	4	2018	SUMMER	7/6/18	70
31	['6LuN9FCiK	1	1	FALSE	0.708	0.648	6	-5.626	1	0.0449	0.0956	0	0.134	0.338	143.955	201707	4	2018	WINTER	2/14/18	70
32	['6M2wZ9GZ	2	1	FALSE	0.646	0.772	2	-3.826	1	0.0564	0.000402	0	0.102	0.414	117.097	203769	4	2017	SUMMER	6/2/17	30
33	['13ubrt8QO	5	1	FALSE	0.855	0.568	5	-8.2	0	0.105	0.0588	0.104	0.0944	0.291	80.027	205040	4	2018	SPRING	5/25/18	70
34	['4O15NlyKLI	1	1	TRUE	0.881	0.457	7	-8.191	0	0.156	0.0327	0	0.19165304	0.296	136.97	198913	4	2019	WINTER	12/12/19	60
35	['5daXMUbrI	1	1	FALSE	0.616	0.621	2	-13.097	1	0.0325	0.0731	0.856	0.19165304	0.0784	107.419	200893	1	1994	WINTER	2/15/94	90
36	['77ziqFxp5g	3	1	TRUE	0.746	0.7	0	-4.669	1	0.341	0.136	0.000159	0.11	0.619	176.044	225933	4	2019	SPRING	4/19/19	0
37	['5iTn5dJcAC	12	1	FALSE	0.744	0.167	5	-15.503	0	0.0413	0.863	0.0203	0.0867	0.431	137.046	189000	4	2006	WINTER	2/24/06	30
38	['7mwUJBmC	15	1	FALSE	0.653	0.26	0	-12.209	1	0.029	0.285	0	0.19165304	0.356	111.139	164627	3	2017	FALL	10/31/17	50
39	['2ybEnEKYdI	2	1	FALSE	0.611	0.764	5	-7.035	1	0.0441	0.0683	0.638	0.19165304	0.53	140.008	227173	4	2003	SPRING	4/10/03	0
40	['4LLpKhyESs	10	1	TRUE	0.739	0.627	0	-8.608	1	0.171	0.0459	0	0.159	0.542	74.962	189981	4	2020	WINTER	1/17/20	0

2.3 Data Analysis and Visualization:

Discovering underlying trends in our data, gaining meaningful insights and making data-driven decisions:

Using R to calculate the mean, median, mode, range, variance, stddev, quartiles, and IQR.
The quantitative Variables chosen for analysis are:

1. danceability
2. energy
3. loudness
4. speechiness
5. acousticness
6. instrumentalness
7. valence
8. duration_ms

Part 1: Description of each variable

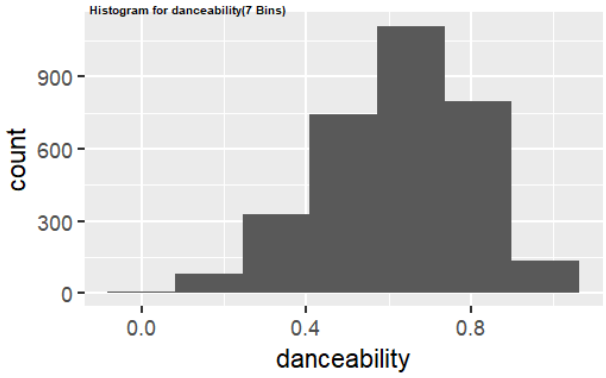
If the mean is much larger than the median, the data are generally skewed right; a few values are larger than the rest. If the mean is much smaller than the median, the data are generally skewed left; a few smaller values bring the mean down. If the mean and median are close, you know the data is balanced, or symmetric, on each side (but not necessarily bell-shaped).

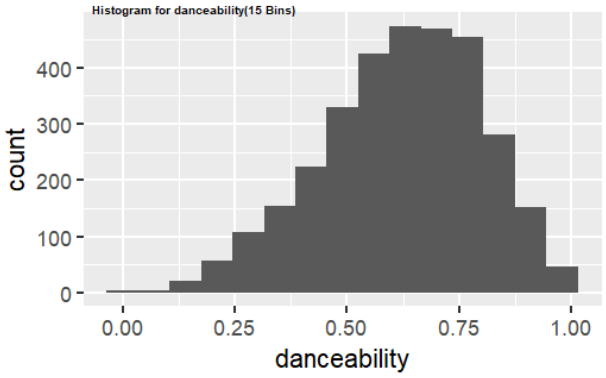
```
$track_number
          Stand dev                Mean                Mode              Median            Variance  Standard Deviation
           5.678357            6.324914            1.000000            5.000000           32.243743            5.678357
            Minimum             Maximun               Range Upper Quantile.100%    LowerQuartile.0%    1st Quartile.25%
           1.000000           50.000000           49.000000           50.000000            1.000000            2.000000
   2st Quartile.50%    3st Quartile.75%                 IQR
           5.000000            9.000000            7.000000

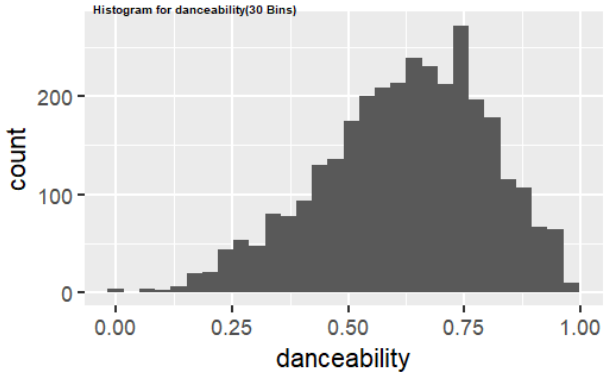
$danceability
          Stand dev                Mean               Mode1               Mode2               Mode3              Median
         0.17775897          0.62132710          0.66100000          0.65200000          0.82100000          0.63900000
           Variance  Standard Deviation             Minimum             Maximun               Range Upper Quantile.100%
         0.03159825          0.17775897          0.00000000          0.98000000          0.98000000          0.98000000
   LowerQuartile.0%    1st Quartile.25%    2st Quartile.50%    3st Quartile.75%                 IQR
         0.00000000          0.50800000          0.63900000          0.75050000          0.24250000
```

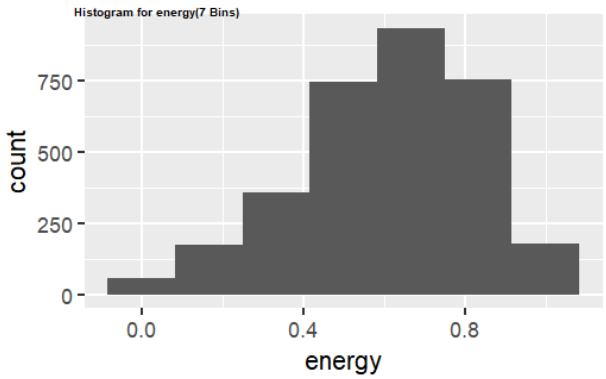
Part 2: Creating a set histogram for each variable using three or more bin sizes. Discussing how the histogram confirms the measures observed above.

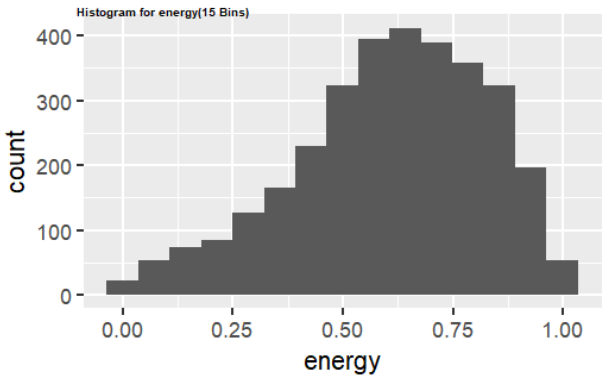
This can be seen in the plots below. For example:
1. Danceability is skewed towards the right a little so we can expect the mean to be a little greater than the median and from the data calculated in part 1, mean = 6.32 and median = 5. Which confirms the findings. Interestingly, there is a spike in the frequency of danceable songs around danceability = 0.75, which only becomes apparent when the number of bins is increased. At lower bin counts there appears to be a plateau between 0.6 and 0.75, and the spike is hidden.
2. Energy shows a skew at all 3 bin counts, which is not as readily apparent from the numeric analysis done previously.
3. Loudness skews strongly left which the numbers above suggest. But the lower bin count (7) and upper bin count (30) both show a more gradual tapering towards 0 than the middle bin count (15).
4. The numeric analysis of speechiness suggests a right-skewed distribution because the median is lower than the mean and mode by almost half. This skew is readily apparent in all 3 versions of the histogram.
5. Acousticness, like speechiness, has a median around half its mean & mode, so we expect the data to be right-skewed, but due to high variance more spread out when graphed. This is the case.
6. Instrumentelness is skewed towards the left a lot so we can expect the mean to be a lot greater than median and from the data calculated in part 1, mean = 0.0780404 and median = 0.00000374. Which confirms the findings.
7. Valence looks symmetrically distributed so we can expect mean to be similar median and from the data calculated in part 1, mean = 0.4751 and median = 0.464. Which confirms the findings.
8. From the huge variance in duration (8.455e+09) I would not have expected such a strongly right-skewed graph. Zoomed in there might be just a sliver of a bar around 3.6e06, but it's only one pixel tall, and there's nothing between that possibly single data point and the rest of the distribution.

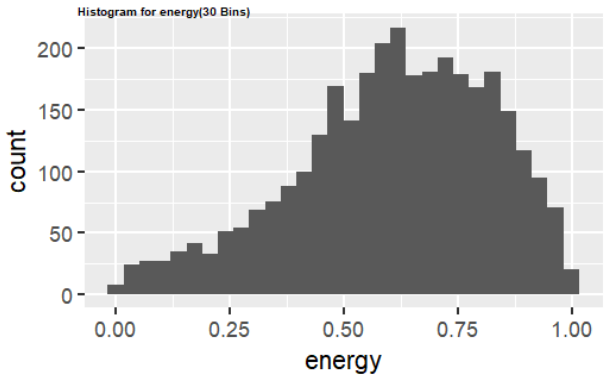

Histogram for danceability(7 Bins)
count
900
600
300
0
0.0
0.4
0.8
danceability

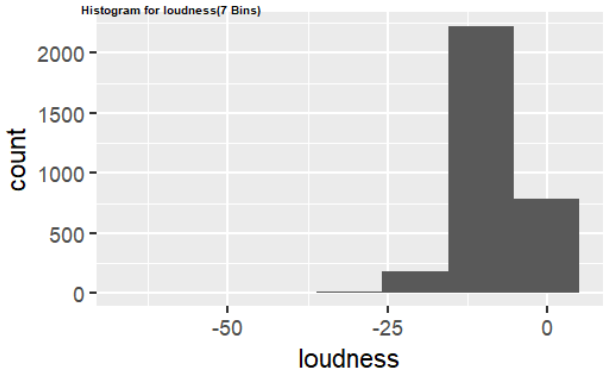

Histogram for danceability(15 Bins)
count
400
300
200
100
0
0.00
0.25
0.50
0.75
1.00
danceability

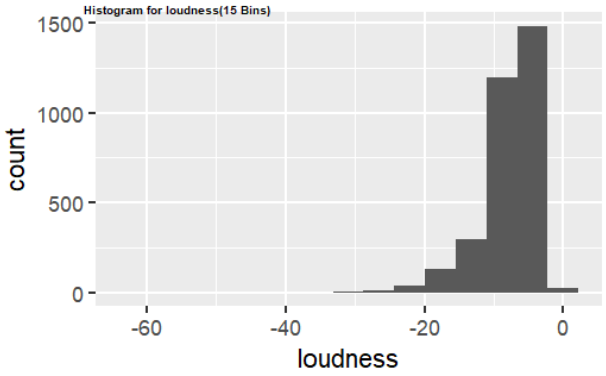

Histogram for danceability(30 Bins)
count
200
100
0
0.00
0.25
0.50
0.75
1.00
danceability

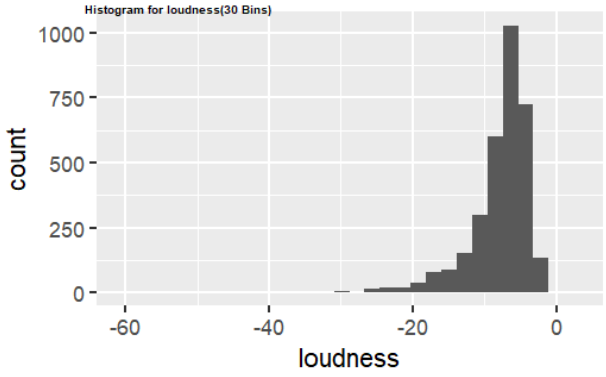

Histogram for energy(7 Bins)
count
750
500
250
0
0.0
0.4
0.8
energy

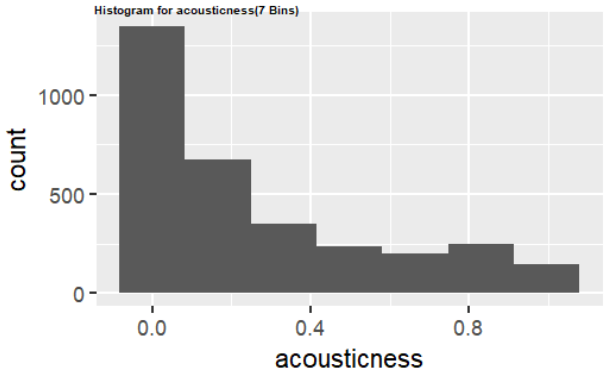

Histogram for energy(15 Bins)
count
400
300
200
100
0
0.00
0.25
0.50
0.75
1.00
energy

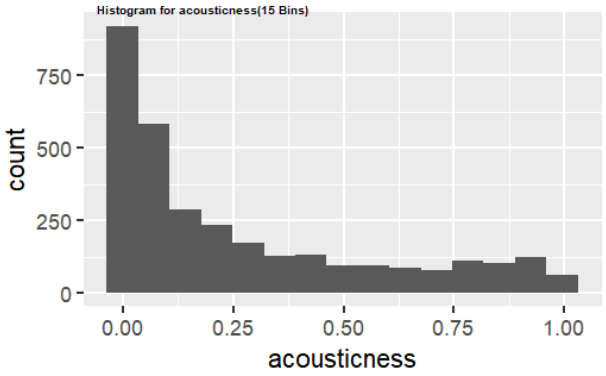

Histogram for energy(30 Bins)
count
200
150
100
50
0
0.00
0.25
0.50
0.75
1.00
energy

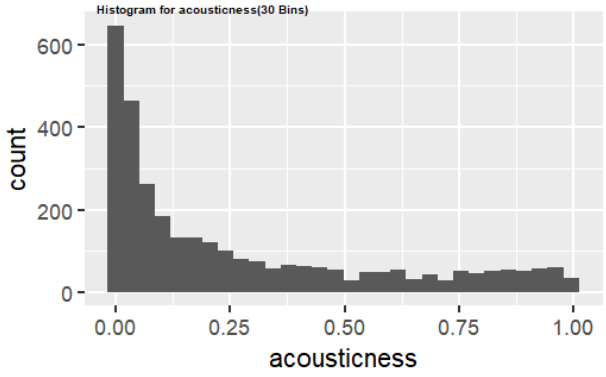

Histogram for loudness(7 Bins)
count
2000
1500
1000
500
0
-50
-25
0
loudness

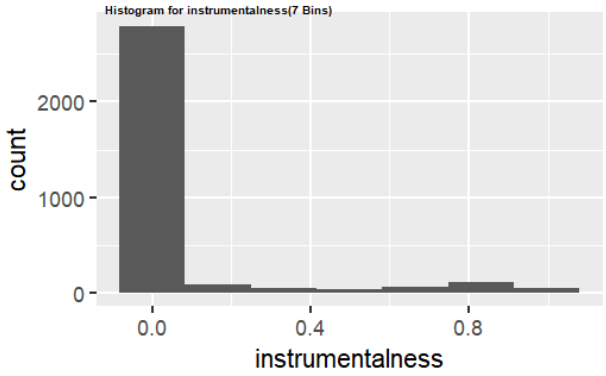

Histogram for loudness(15 Bins)
count
1500
1000
500
0
-60
-40
-20
0
loudness

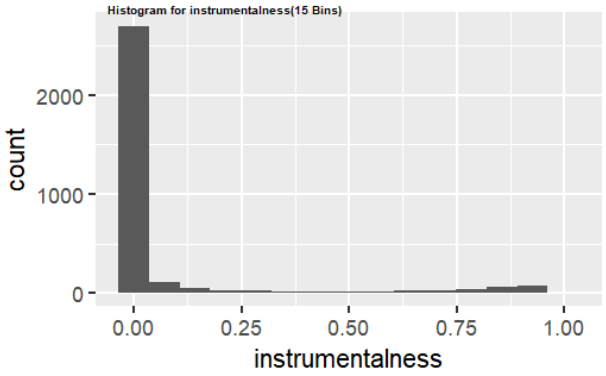

Histogram for loudness(30 Bins)
count
1000
750
500
250
0
-60
-40
-20
0
loudness

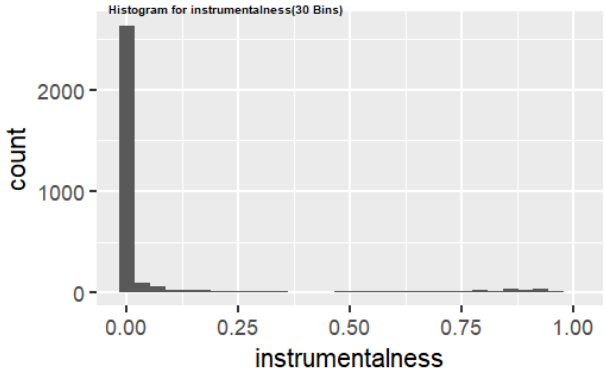

Histogram for acousticness(7 Bins)
count
1000
500
0
0.0
0.4
0.8
acousticness

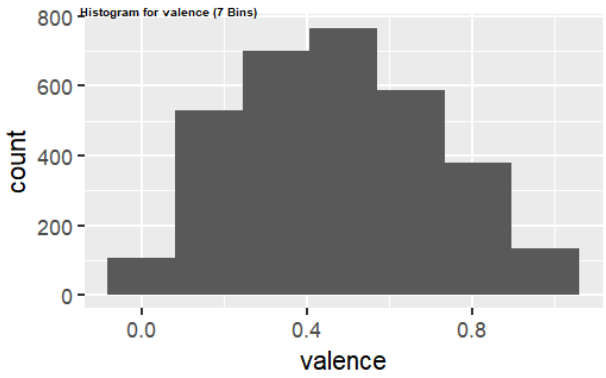

Histogram for acousticness(15 Bins)
count
750
500
250
0
0.00
0.25
0.50
0.75
1.00
acousticness

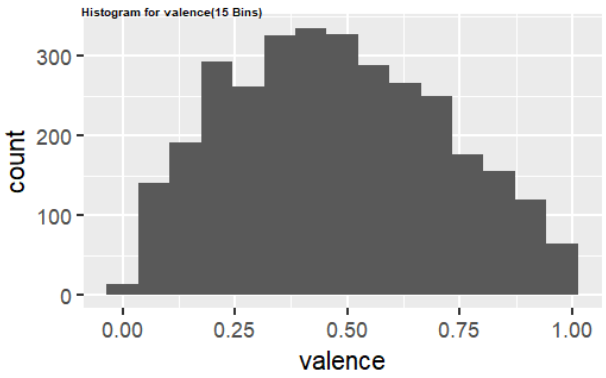

Histogram for acousticness(30 Bins)
count
600
400
200
0
0.00
0.25
0.50
0.75
1.00
acousticness

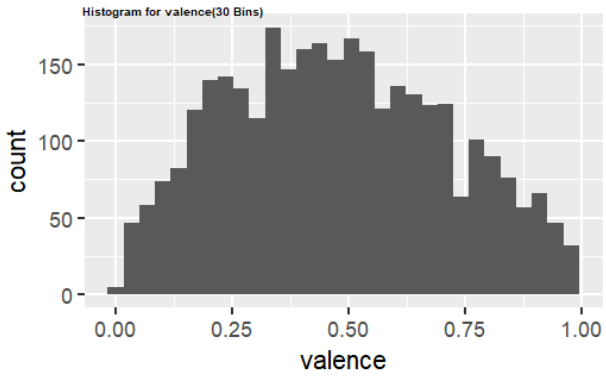

Histogram for instrumentalness(7 Bins)
count
2000
1000
0
0.0
0.4
0.8
instrumentalness

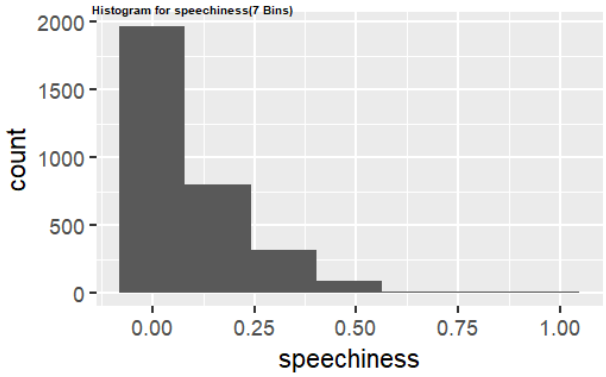

Histogram for instrumentalness(15 Bins)
count
2000
1000
0
0.00
0.25
0.50
0.75
1.00
instrumentalness

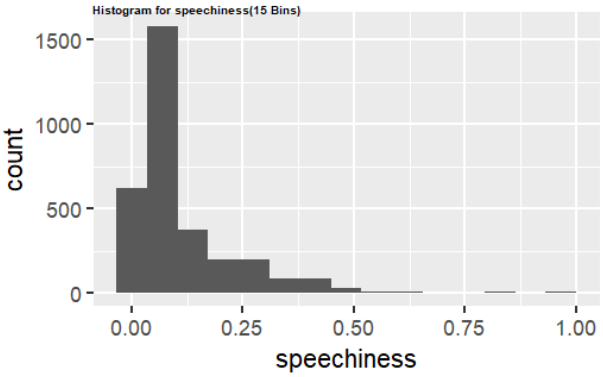

Histogram for instrumentalness(30 Bins)
count
2000
1000
0
0.00
0.25
0.50
0.75
1.00
instrumentalness

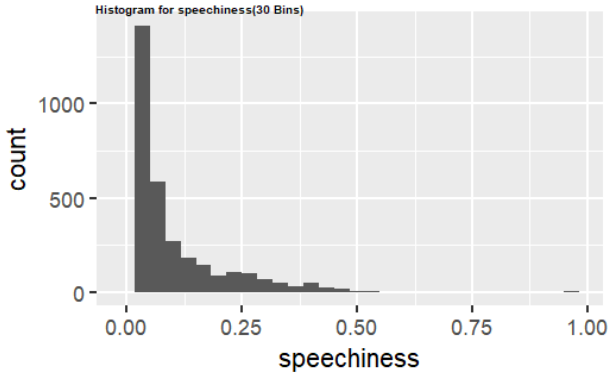

Histogram for valence (7 Bins)
count
800
600
400
200
0
0.0
0.4
0.8
valence

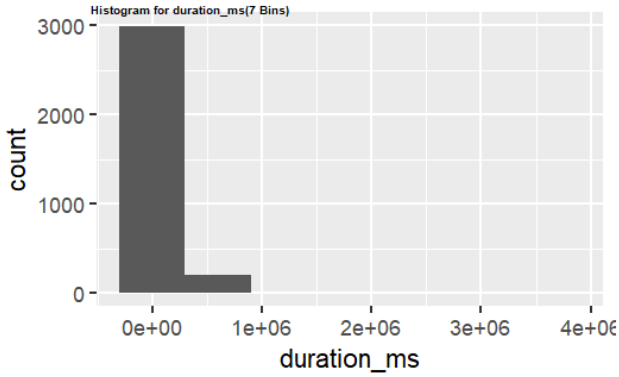

Histogram for valence(15 Bins)
count
300
200
100
0
0.00
0.25
0.50
0.75
1.00
valence

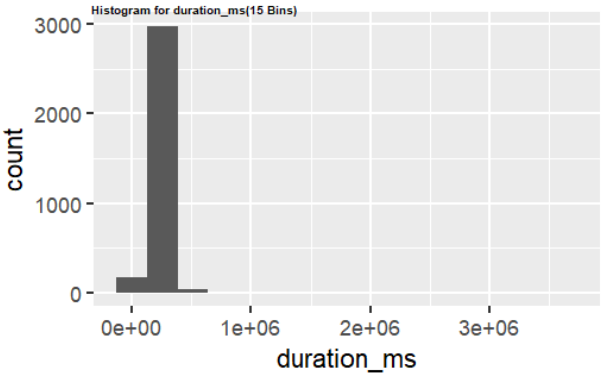

Histogram for valence(30 Bins)
count
150
100
50
0
0.00
0.25
0.50
0.75
1.00
valence

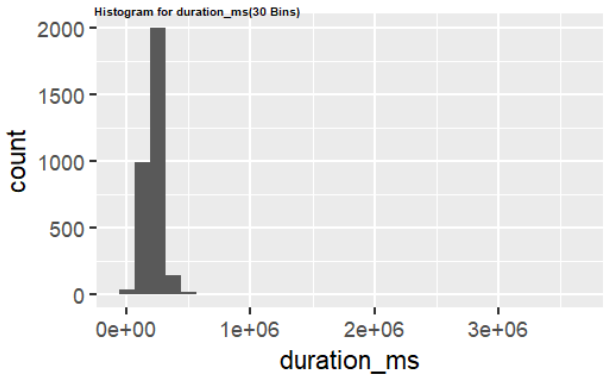






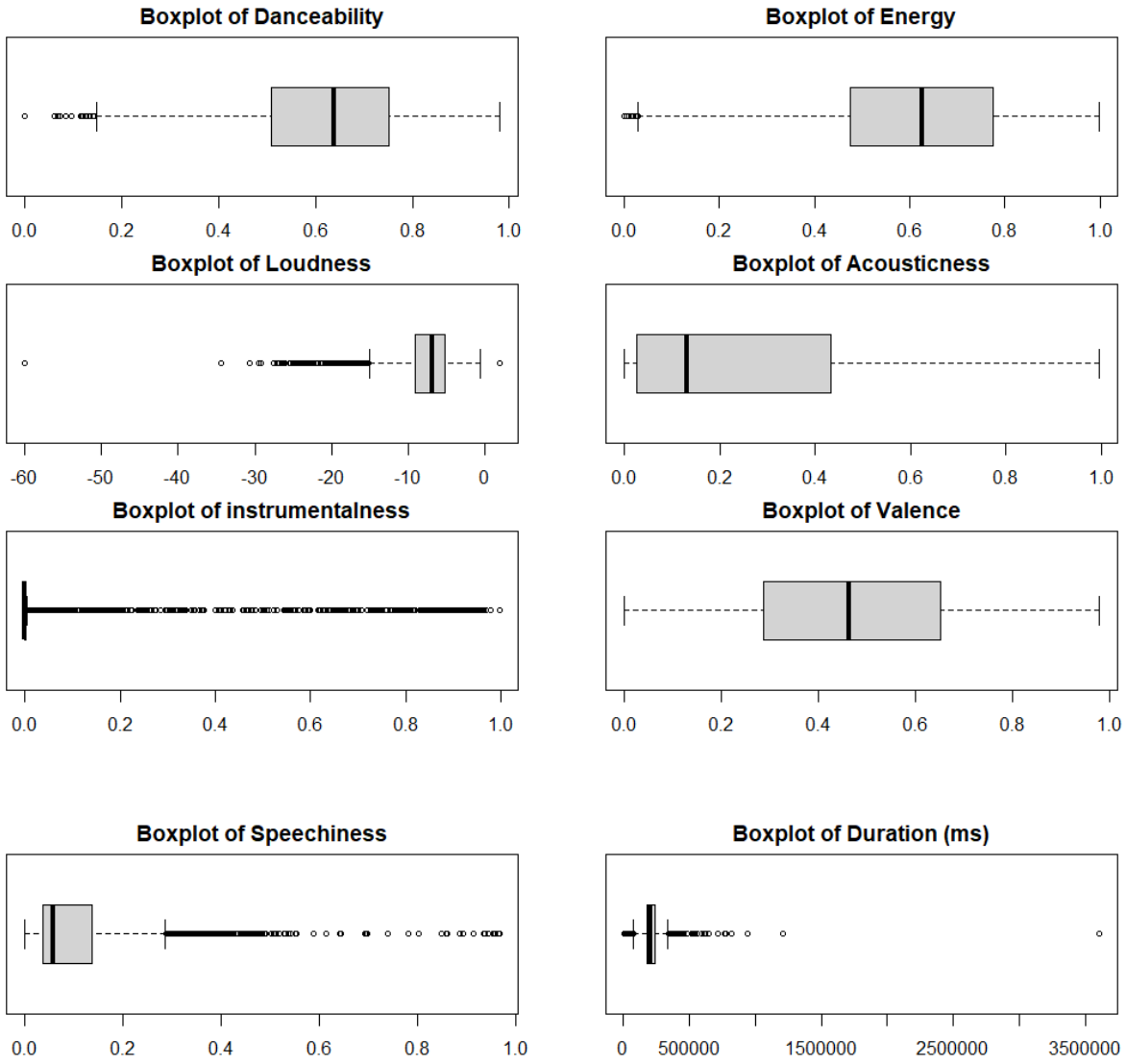






Part 3: Creating boxplots (or violin plots) to visualize each variable to check the min, max, median, Q1, and Q3. Box plot is used to determine if there are any outliers.

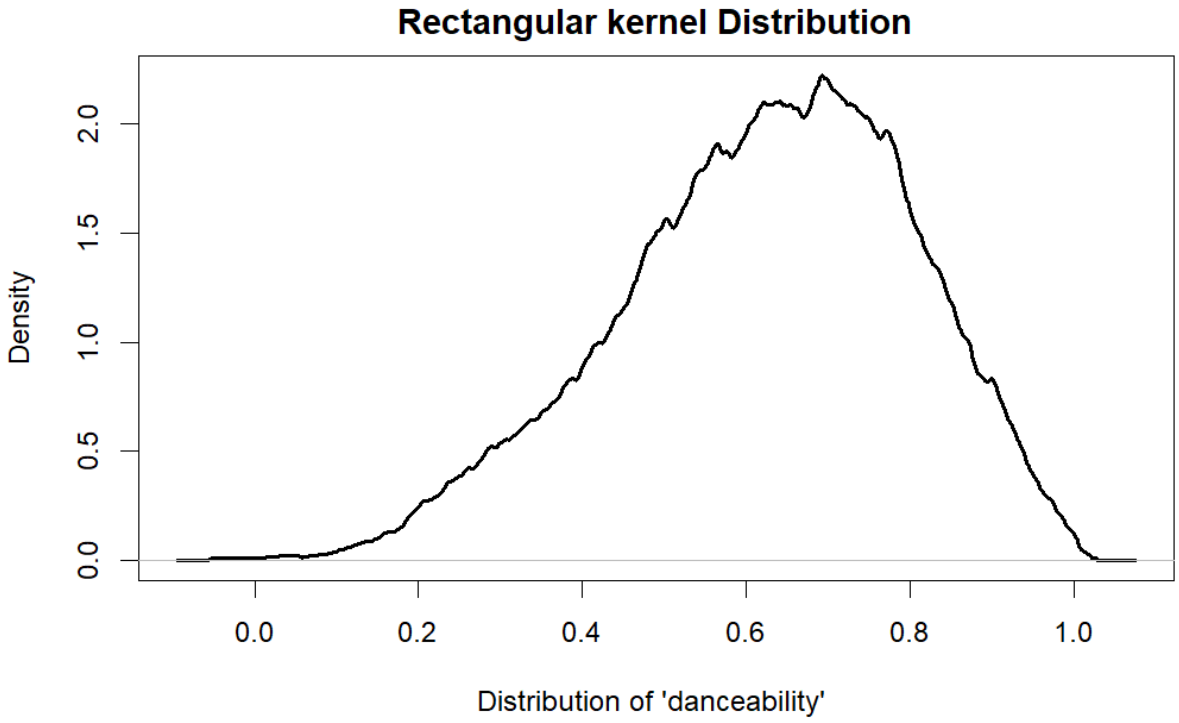
For the Danceability, there are outliers outside Q1-1.5 IQR range. The boxplot of energy follows similar pattern as well. Coming to Acousticness, It has no outliers meaning that all the points are withing the IQR range. Loudness has an outlier at -60, which is basically silent, which is strange for music to be--might be an error on Spotify's classifaction's part. Instrumentalness has many outliers. Valence attribute has no outliers as well. Speechiness has a lot of outliers outside the Q3+1.5 IQR range. Duration has some outliers, including the most incredible outlier that sits way, way beyond the IQR and more than double the next highest datapoint.

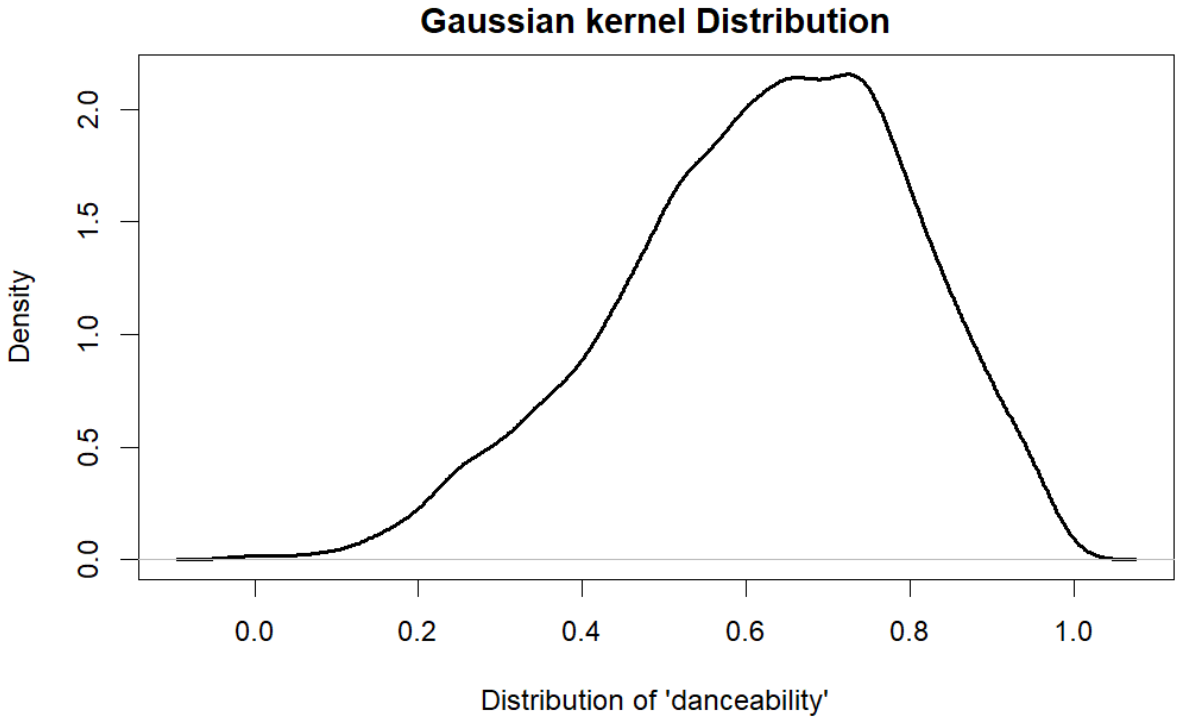


Part 4: Using kernel density functions to estimate the shape (distribution) of the data.

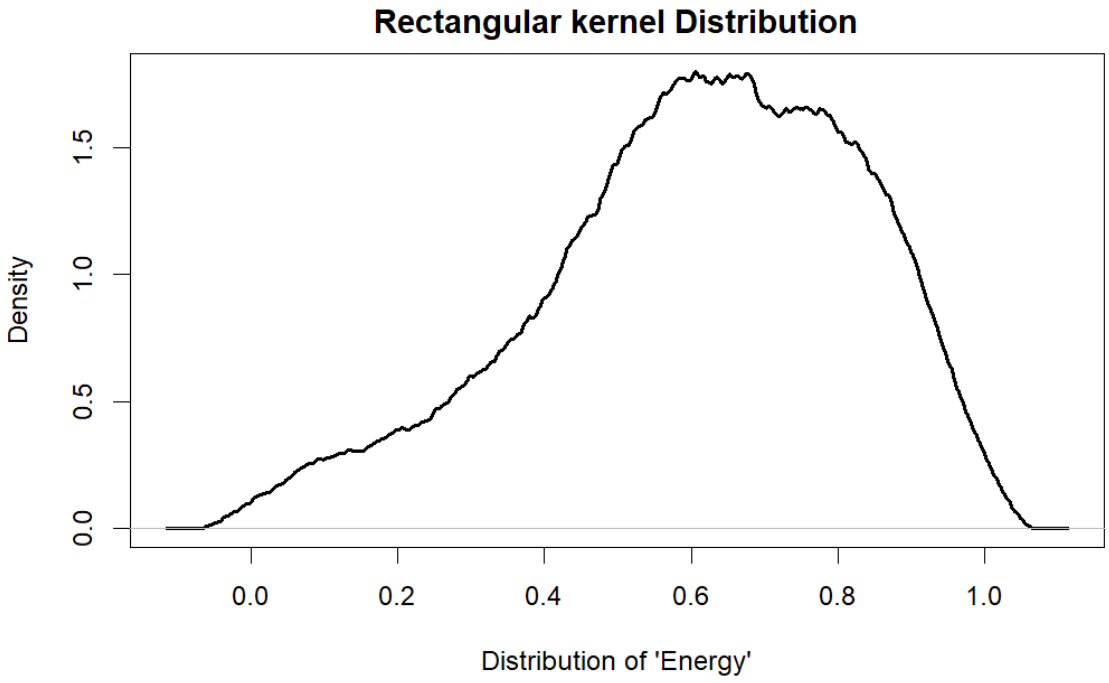
a. For the quantitative variable Danceability, it's clear from the gaussian and rectangular kernel distribution that it's a well-balanced distribution. Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability,

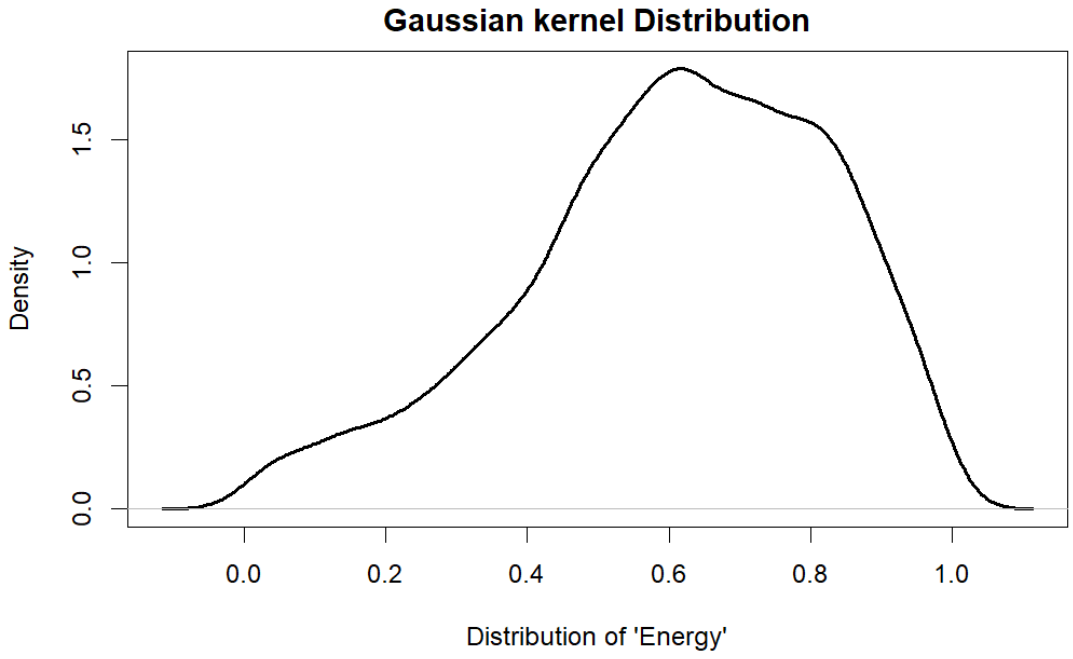
beat strength, and overall regularity. Since the distribution is more shifting towards 1, so the songs are more danceable.



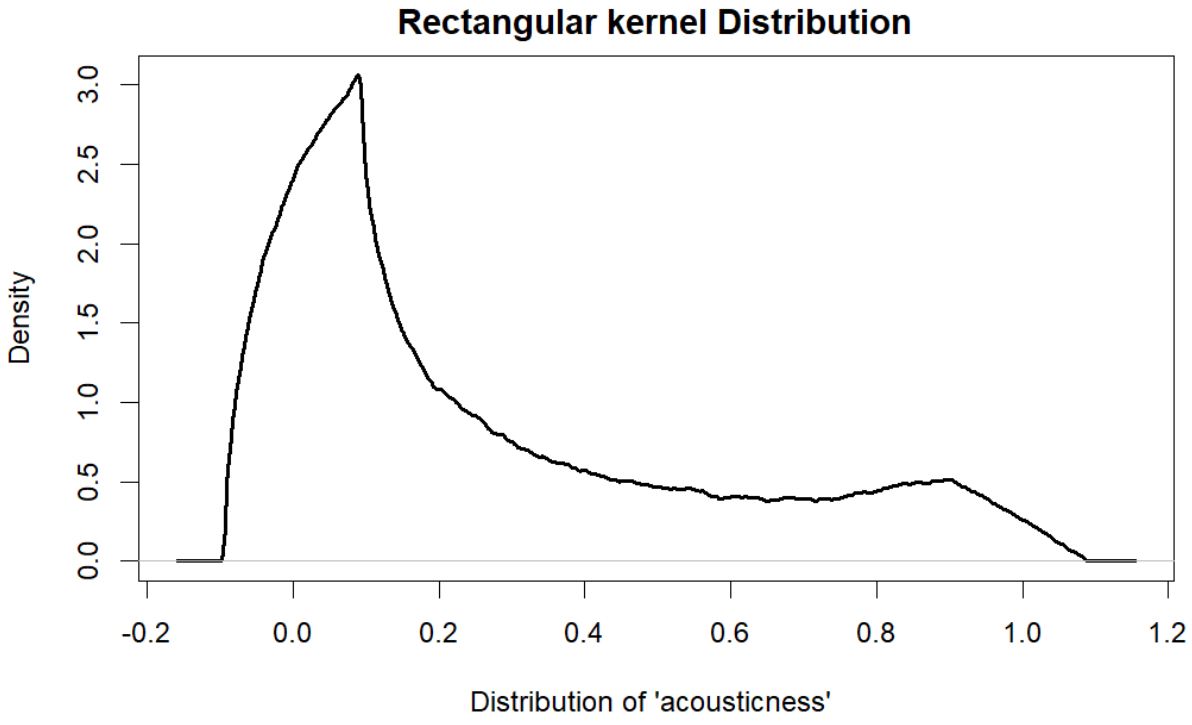


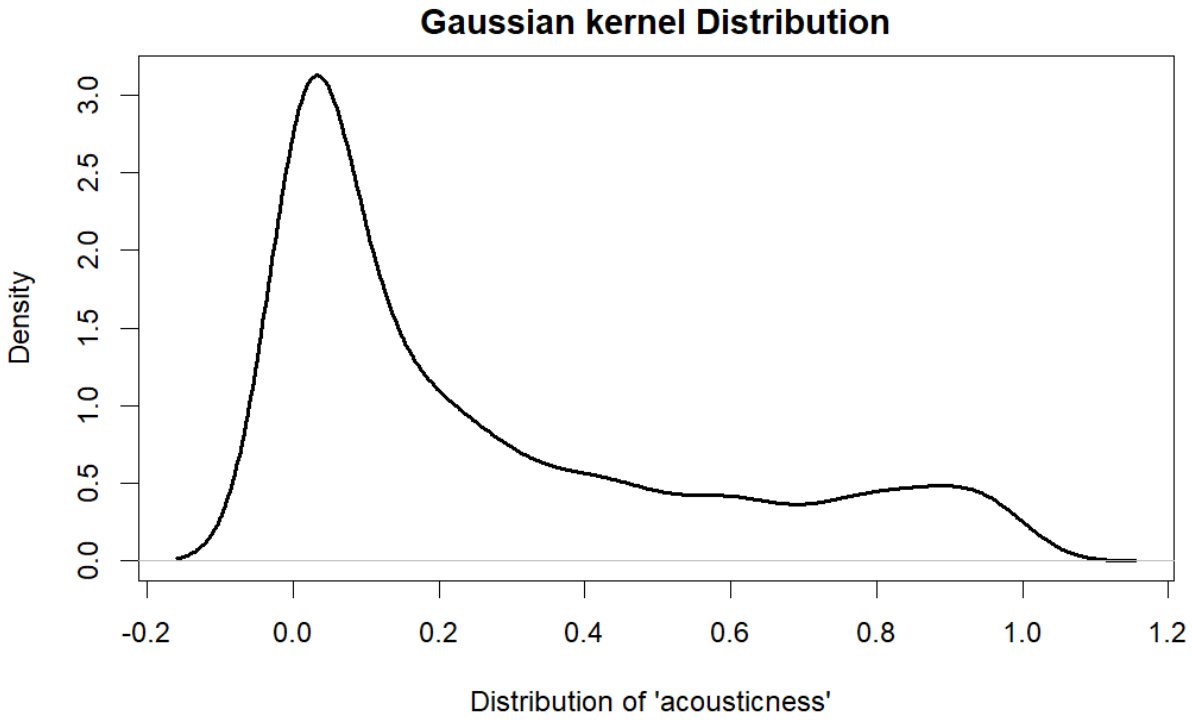
b. For the quantitative variable Energy, it's clear from the gaussian and rectangular kernel distribution that it's a well-balanced distribution. Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Since the distribution is more shifting towards 1, so the songs in our sample dataset have more energy.



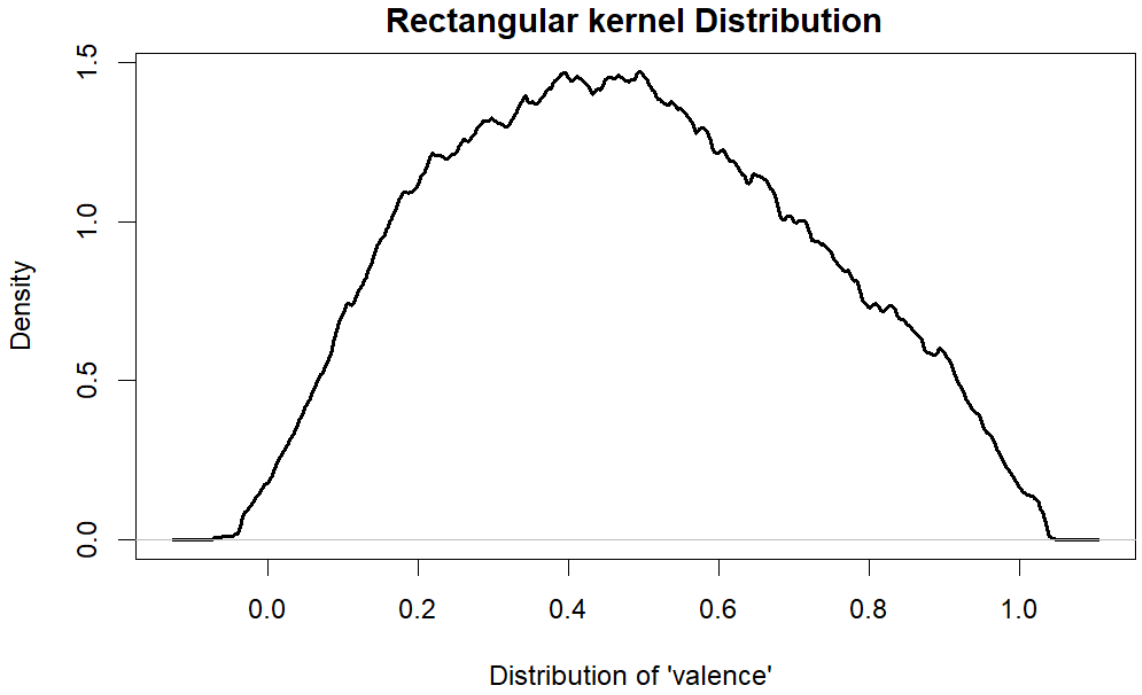


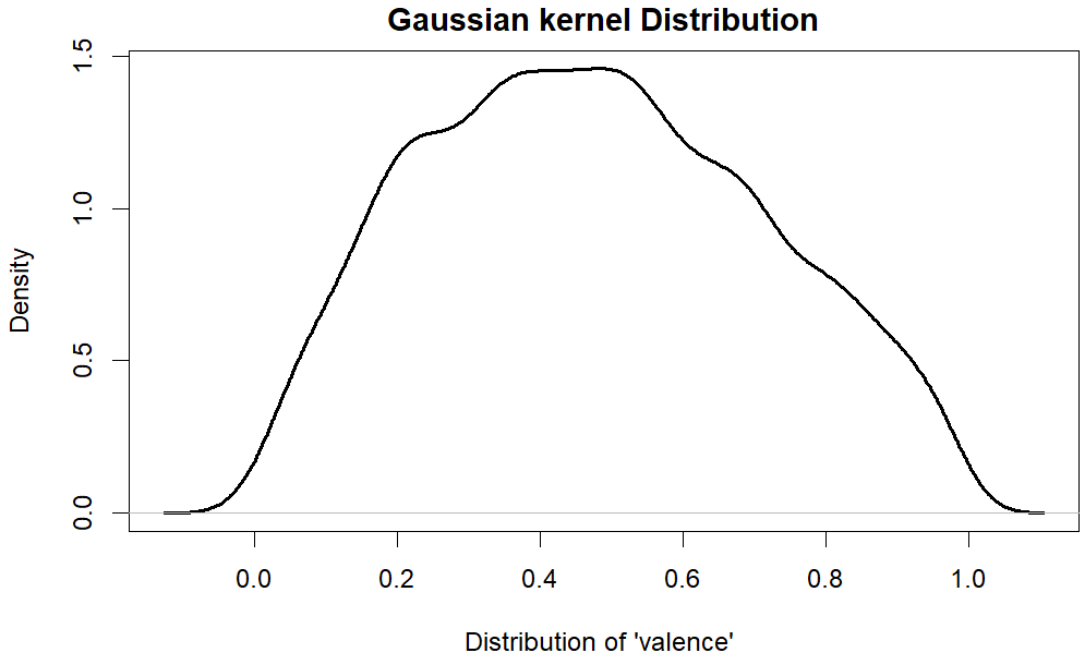
c. For the quantitative variable acousticness, it's clear from the gaussian and rectangular kernel distribution that it's a well-balanced distribution. A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic. Since the distribution is skewed towards 0, the songs in our sample dataset are less acoustic in nature.



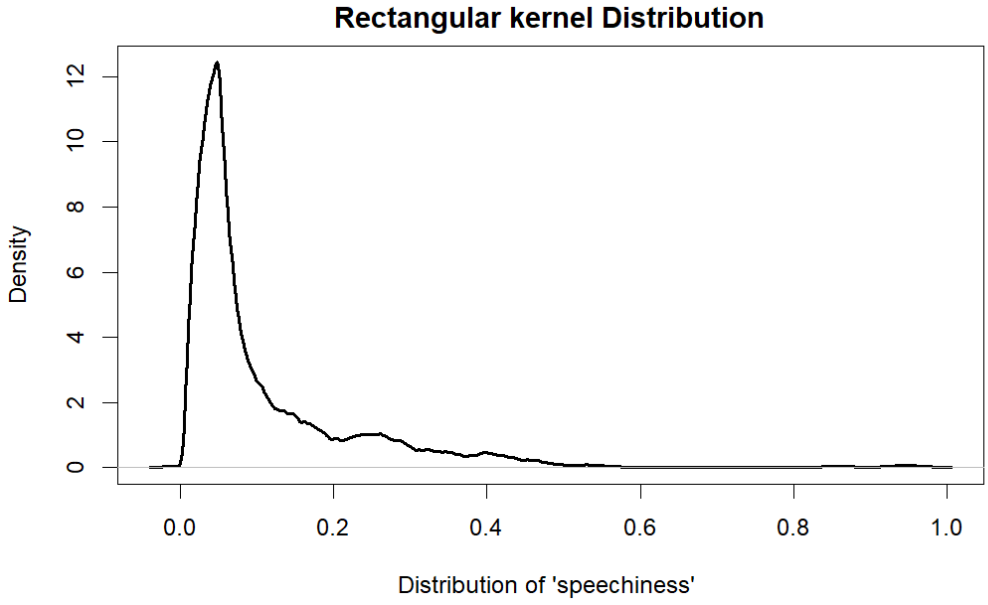


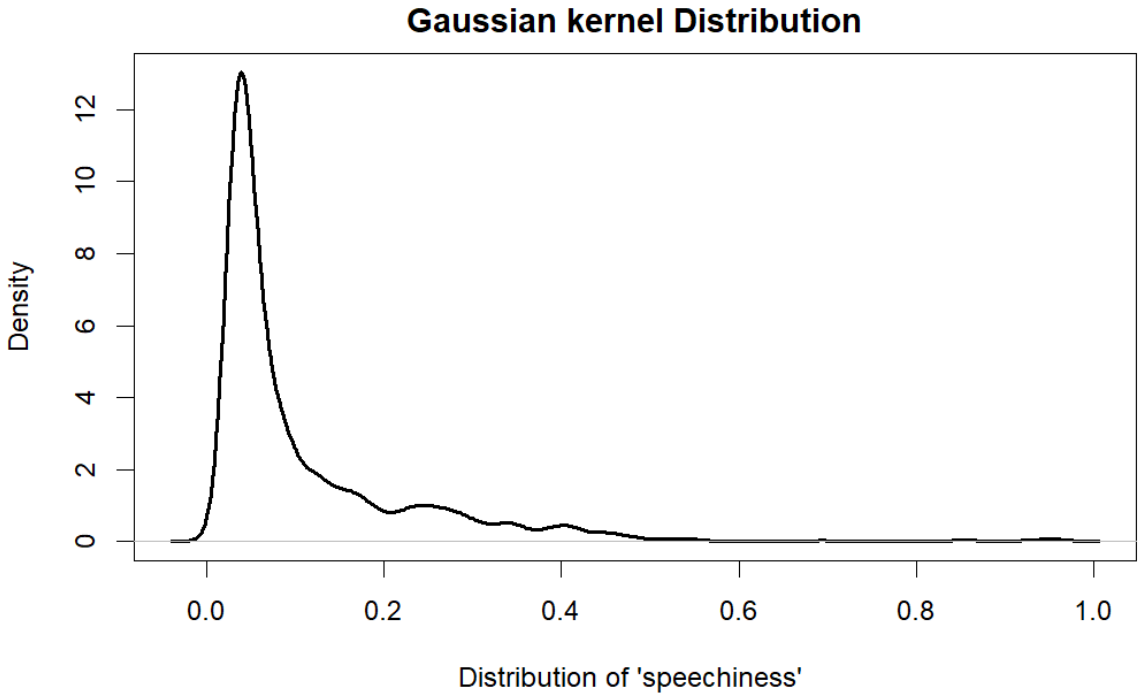
d. For the quantitative variable valence, it's clear from the gaussian and rectangular kernel distribution that it's a well-balanced distribution. A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry). From the distribution, we can infer that the sample dataset has a mix of both sad and happy songs.



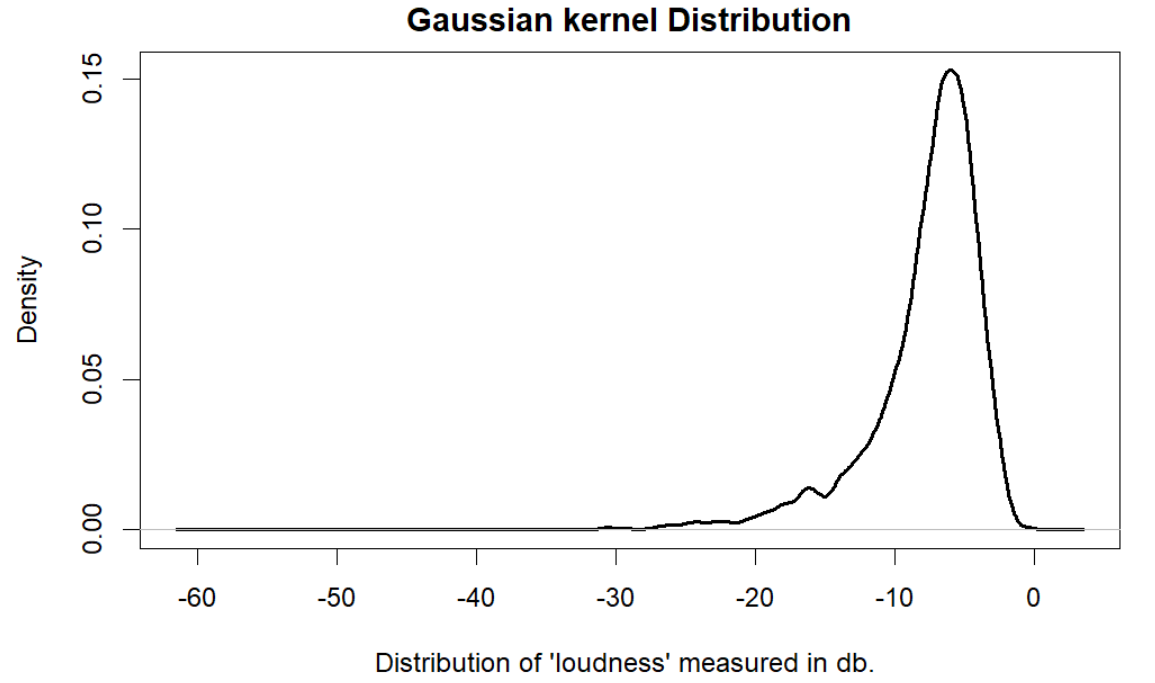


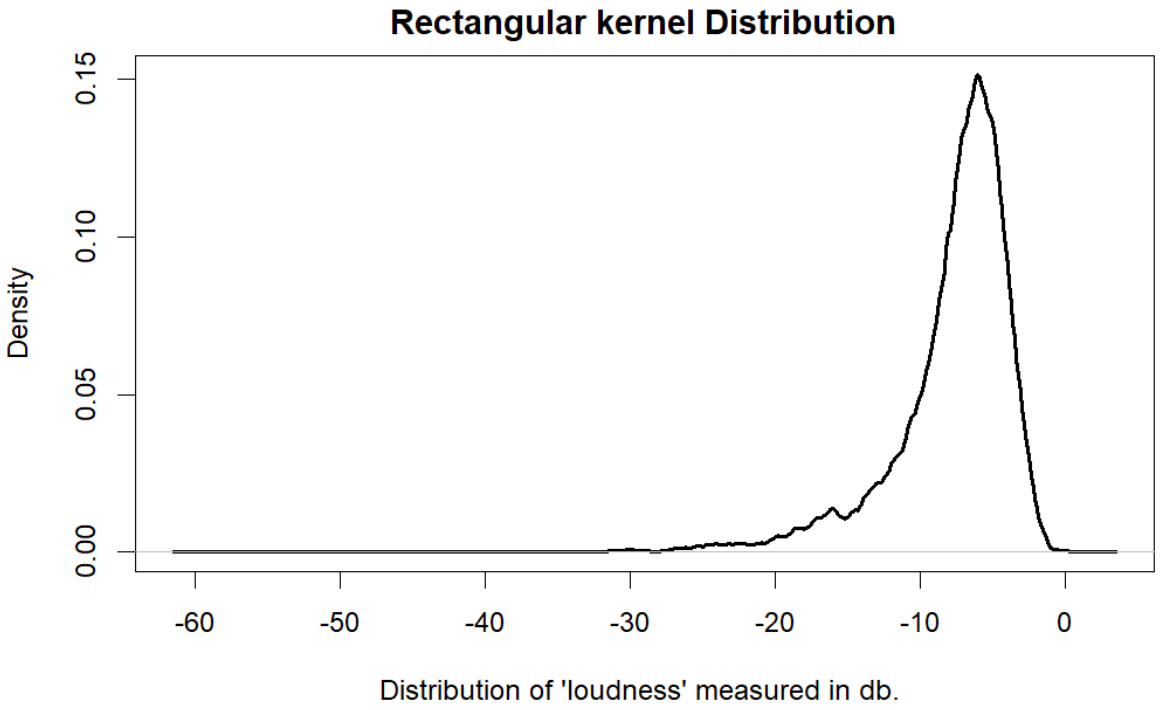
e. Below is the distribution for quantitative variable speechiness. Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks. From the distribution, we can infer that the sample dataset has more of instrumental music and has less speech like tracks. This is also confirmed with the box plot.



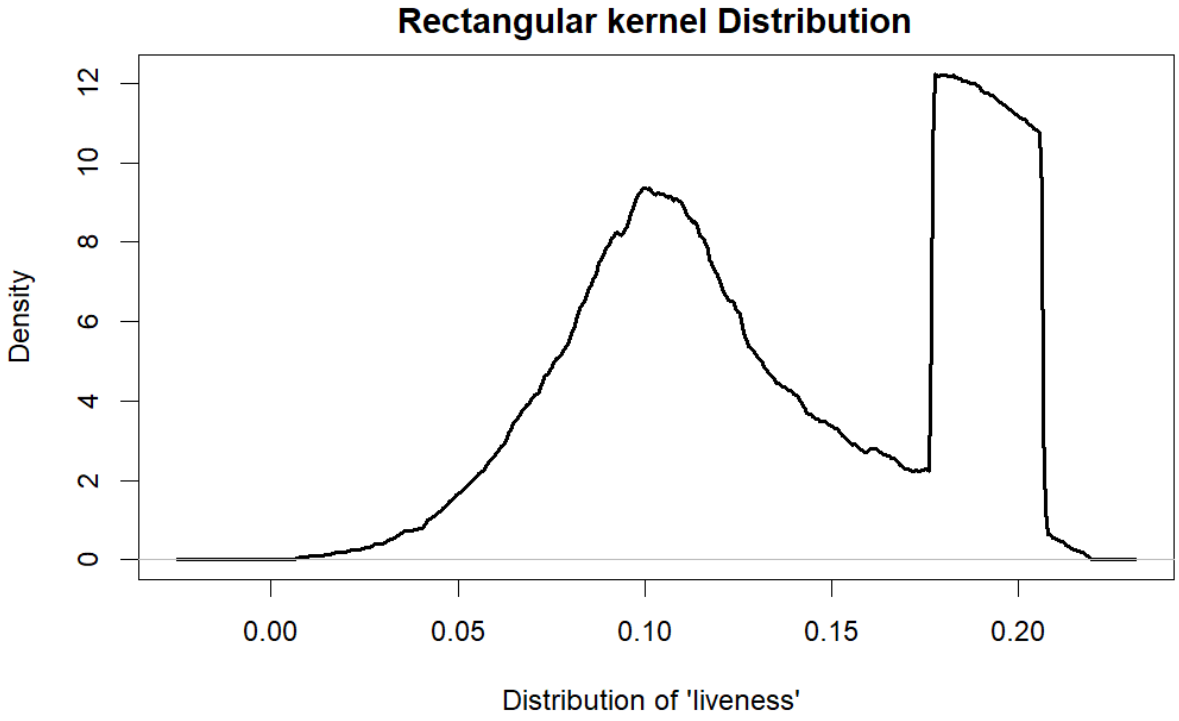


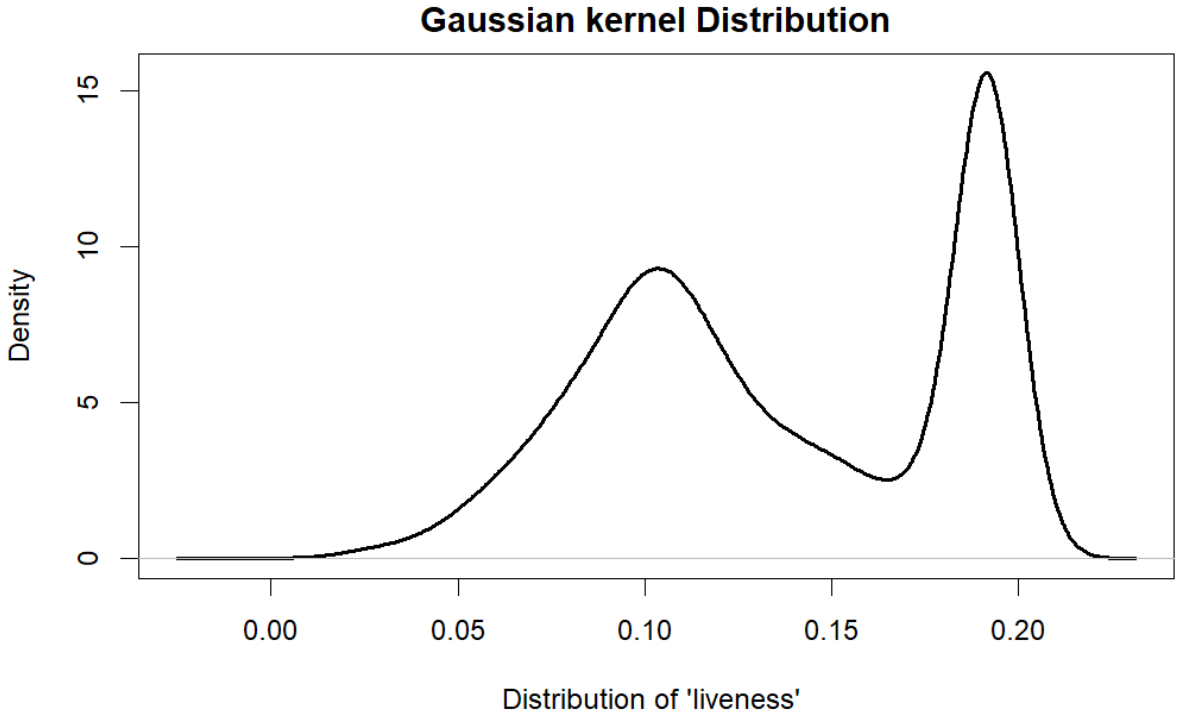
f. Below is the distribution for quantitative variable loudness. The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing the relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 dB. From the distribution, we can infer that the sample dataset has loudness ranging from -14db to 0db. This is because no extra distortion is introduced in the transcoding process. Also, Spotify usually compresses the loudness to the -20 to 0db range.





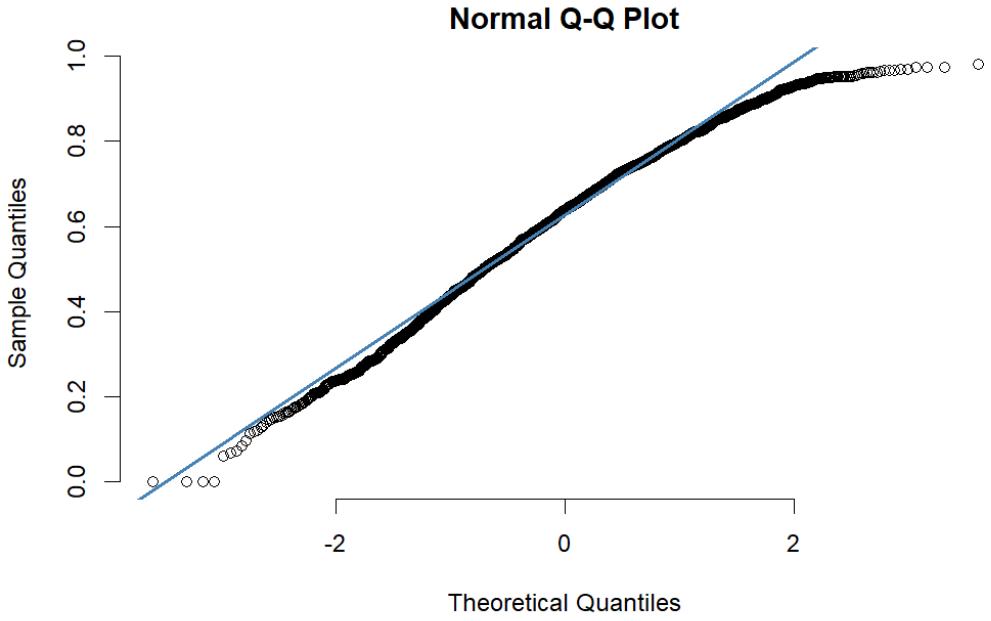
g. Below is the distribution for quantitative variable liveness. Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides a strong likelihood that the track is live. From the distribution, we can infer that the sample dataset has liveness with 2 maxims.



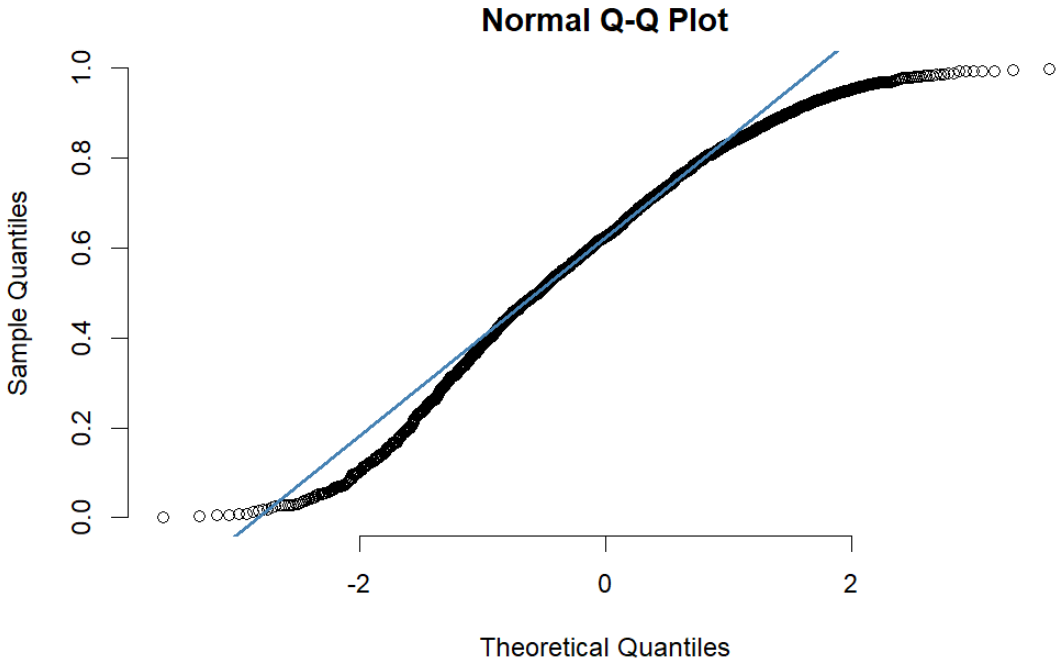


Plotting Q-Q plot to get a better understanding of the attributes of the sample.

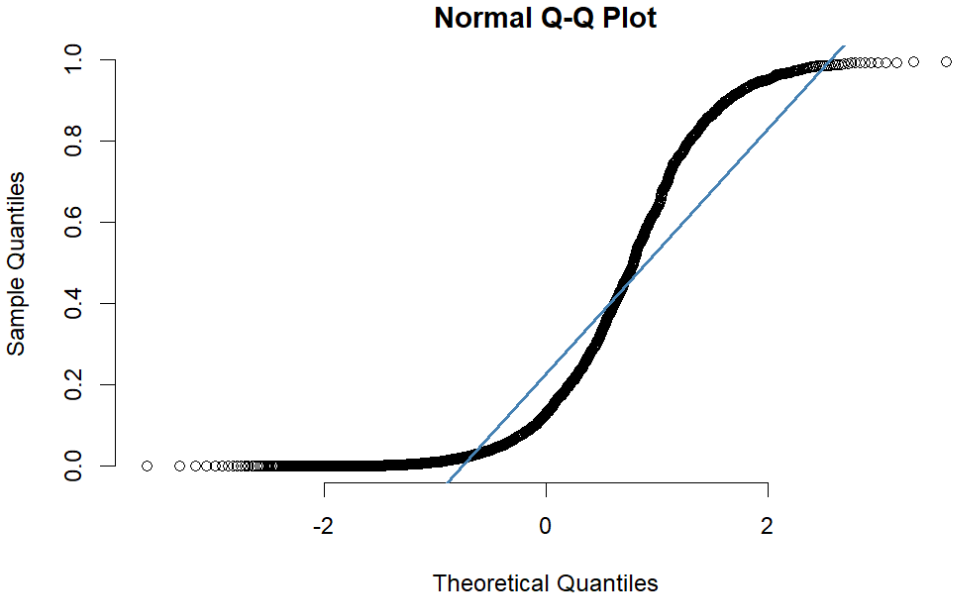
a. For the quantitative variable Danceability, from the Q-Q plot it can be observed that it is very close to the normal distribution which was earlier observed by histogram.



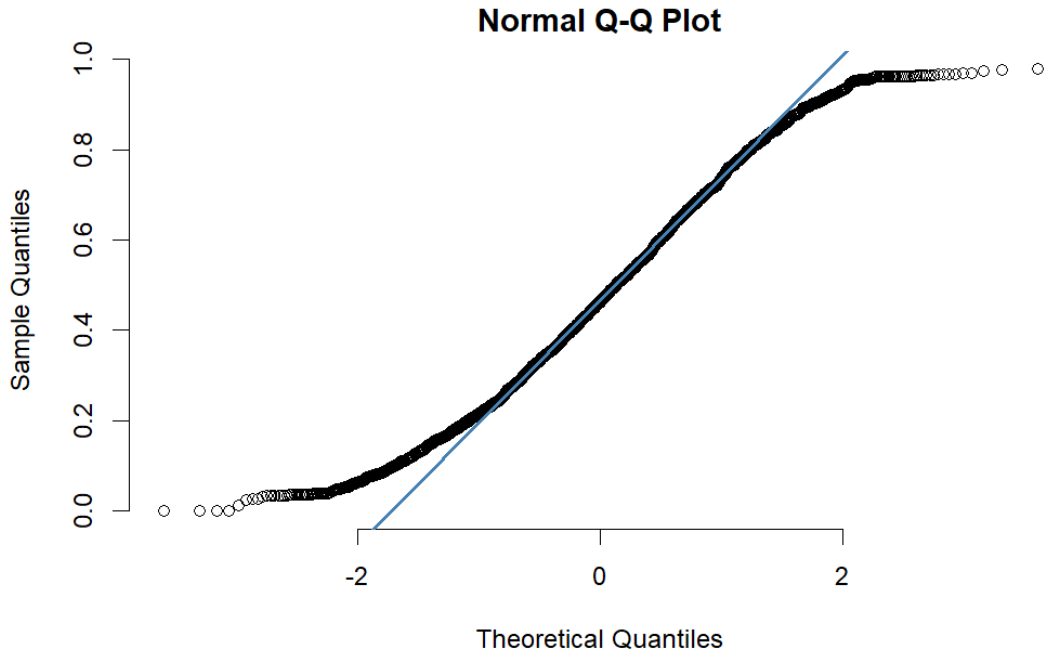
b. For the quantitative variable Energy, from the Q-Q plot, it can be observed that it is very close to normal distribution also as it exhibits growing departure from the fitted line above the line in the first few points and rising departure from the fitted line below the line in the last few points it is slightly short-tailed.



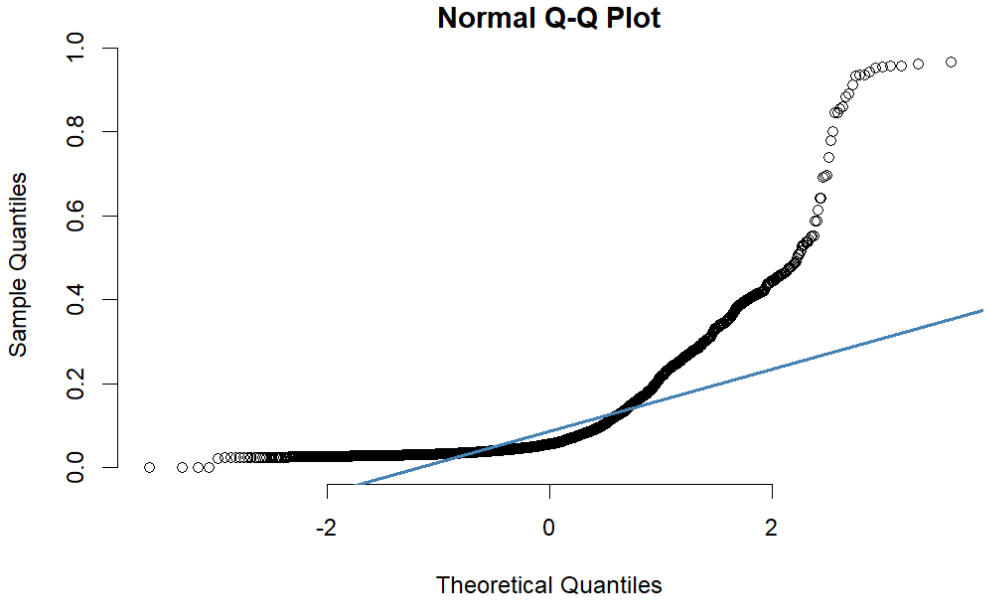
c. For the variable Acousticness, from the Q-Q plot it can be observed that it is right skewed also known as positive skew, in this distribution majority of values are concentrated in the left tail, but the right tail is longer.



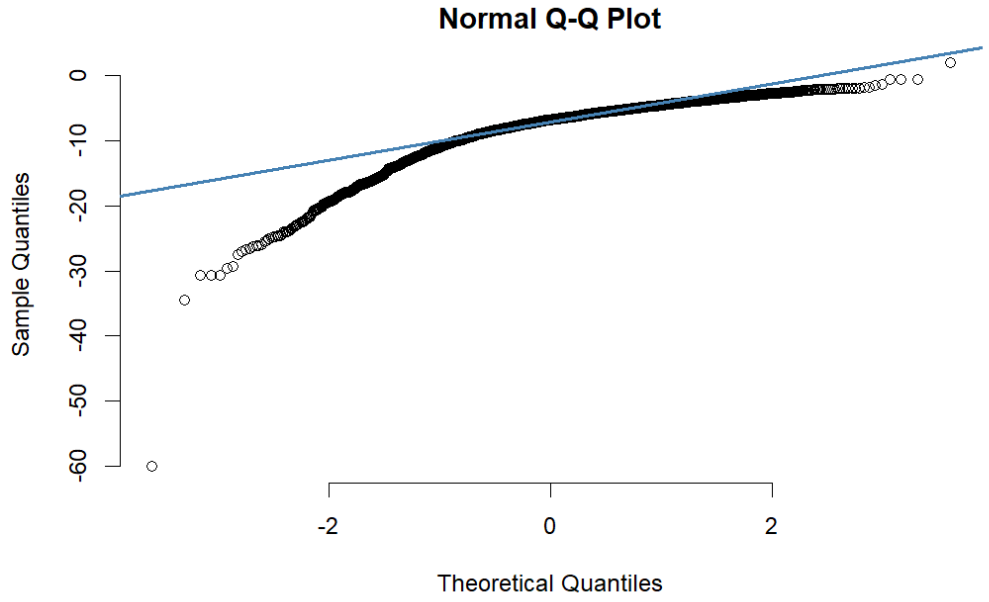
d. For the quantitative variable Valence, from the Q-Q plot, it can be observed that it is very close to normal distribution also as it exhibits growing departure from the fitted line above the line in the first few points and rising departure from the fitted line below the line in the last few points it is slightly short-tailed.



e. For the variable Speechiness, from the Q-Q plot it can be observed that it is right skewed also known as positive skew, in this distribution majority of values are concentrated in the left tail, but the right tail is longer.

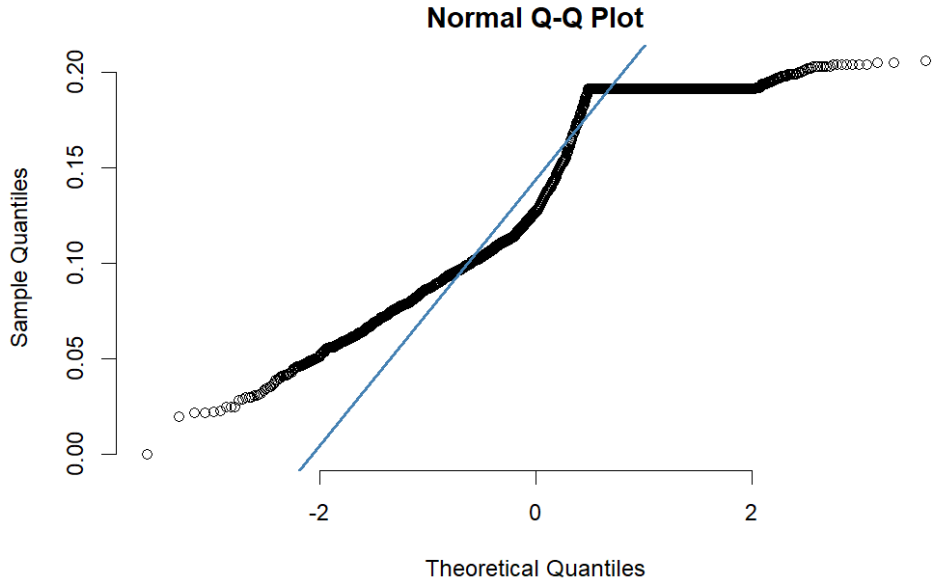


f. For the variable Loudness, from Q-Q plot it can be observed that it is left skewed also known as negative skew, in this distribution majority of values are concentrated in the right tail, but the left tail is longer.



g. For the variable Liveness from Q-Q plot it can be observed that the shape of the plot is consistent with a left-skew, possibly bimodal distribution (with a small mode on the right). Some of these plots show that some of the variables are normally or almost normally distributed: liveness, speechiness, acousticness, and energy all fall fairly close to the fit line. But the other variables (valence, but especially loudness and liveness)

have strange shapes that rarely come near the diagonal fit-line, which strongly suggests these song-features are not normally distributed at all.



3 Results

3.1 Hypothesis Testing

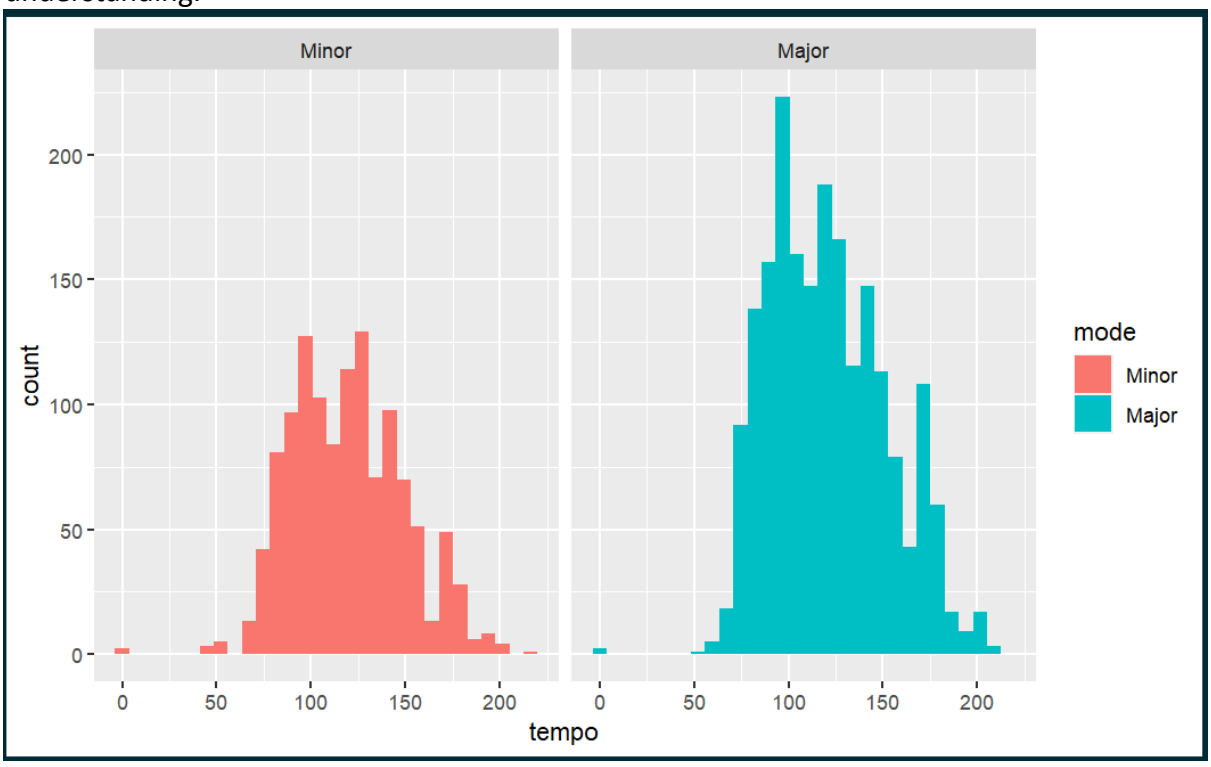
The hypothesis is a preconceived assumption about the population of our interest based on some evidence. When a researcher believes that there is no correlation between the variables in an observation, a null hypothesis is developed. The goal of the investigation in this instance is to confirm or disprove this assumption. Hypothesis testing enables one to assess the reliability of any assertion or assumption before applying it to the data collection. In this project using various hypothesis testing techniques, we tried to answer multiple questions that we were curious about, and which had the potential of guiding us in further analysis. We wanted to understand if the tempo is affected by chords or not, if the genre of music is louder than the others or if any genre is preferred by the listeners.

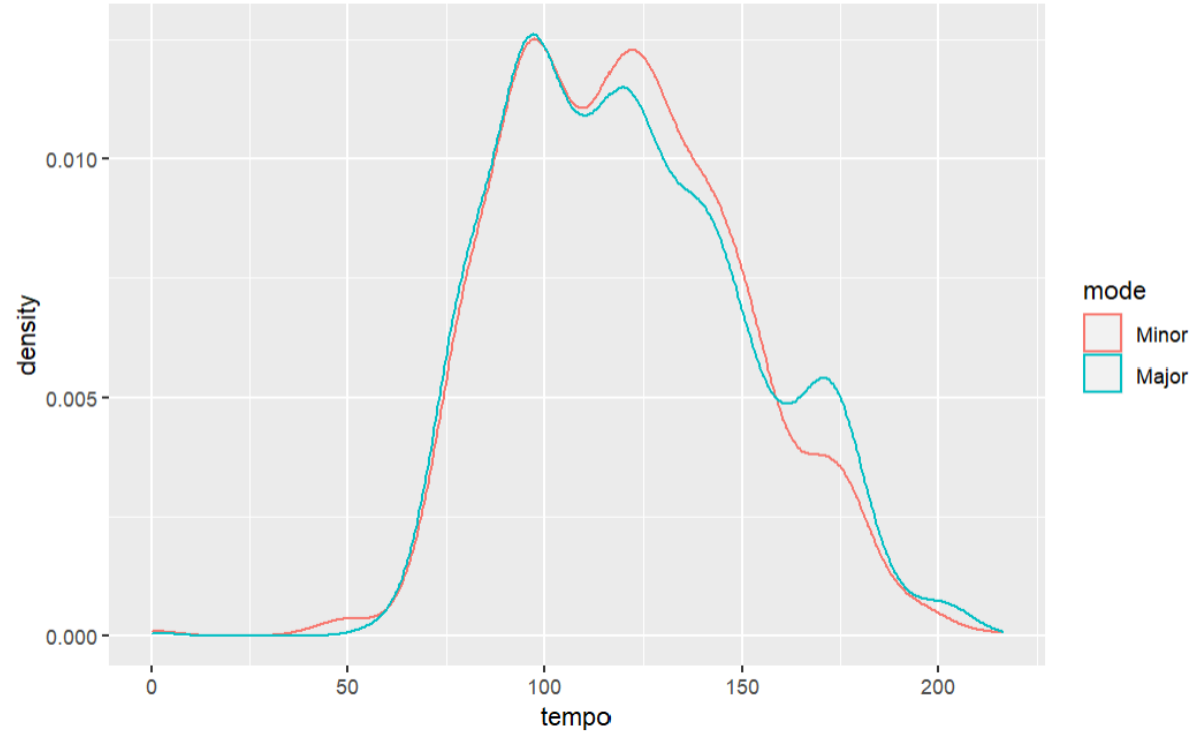
3.1.1 T Test:

In statistics, Welch's T-test is used which is a two-sample location test. It is used to test the hypothesis such that the two populations have equal means. Welch's test, which is an adaptation of the Student's T-test is much more robust than the latter. It is more reliable when the two samples have unequal variances and unequal sample sizes. Student's t-test assumes that the sample means being compared for two populations are normally distributed and that the populations have equal variances. Welch's t-test is designed for unequal population variances, but the assumption of normality is maintained.

1. Differences in tempo between songs in major and minor keys:

Test if there is a difference in the distribution of tempo between songs in a major key and songs in a minor key. Firstly, Let's look at the distribution of "Tempo" to get a better understanding.





Since the distribution looks very alike, the aim is to compute the mean for the modes through a t-test.

```
        Welch Two Sample t-test

data:  major_data and minor_data
t = 1.2683, df = 2612.1, p-value = 0.2048
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.7615593  3.5505965
sample estimates:
mean of x mean of y
 121.1018  119.7073
```

The p-value for this test is around 0.20, so the null hypothesis can't be rejected which will eventually pass the alternate hypothesis. To further analyze, the test was conducted to comprehend if the distribution of tempo for songs in a major key is significantly different from the distribution of tempo for songs in a minor key with the Kolmogorov-Smirnov test.

Simple STUDENTS T-TEST:

Ho = Acousticness decreased over the years.
Ha = Acousticness varied significantly over the years.

```
        One Sample t-test

data:  model_test$acousticness
t = 0.11197, df = 641, p-value = 0.9109
alternative hypothesis: true mean is not equal to 0.2648193
95 percent confidence interval:
 0.2427320 0.2895779
sample estimates:
mean of x
0.2661549
```

As we see from the results, the observed T-score is 0.11 and P-value is 0.91. Since the observed T-score is below the critical value and, the P-value is greater than 0.05 so we cannot reject the null hypothesis. So Acousticness did not decrease over the years.

3.1.2 Z – Test

What is a Z-Test?
When the variances are known and the sample size is large, a z-test is a statistical test that is used to assess whether two population means differ from one another. To execute a precise z-test, the test statistic is expected to have a normal distribution, and nuisance variables like standard deviation should be known.

1. Was the White Christmas song the most popular in the 1990s?

Null Hypothesis: 'White Christmas was the most popular song"

Alternate Hypothesis: "White Christmas was not the most popular song"

```
        One-sample z-Test

data:  data
z = 1.3054, p-value = 0.1917
alternative hypothesis: true mean is not equal to 44.74275
95 percent confidence interval:
 37.09305 82.90695
sample estimates:
mean of x
       60
```

As per the results, the observed Z-score is -6.5618 and P-value is 5.315e-11. Since the observed Z-score is above the critical value and, the P-value is very less than 0.05 so the null hypothesis can be rejected. Hence, it concludes that classical music is not very loud.

What's the relationship between rock music and loudness?
Ho: Rock music has more loudness
Ha: Rock music doesn't more loudness

```
        One-sample z-Test

data:  data2
z = -1.6394, p-value = 0.1011
alternative hypothesis: true mean is not equal to -7.832886
95 percent confidence interval:
 -8.924013 -7.735723
sample estimates:
mean of x
-8.329868
```

As per the results, the observed Z-score is -1.6394 and P-value is 0.1011. Since the observed Z-score is below the critical value and, the P-value is greater than 0.05 so the null hypothesis cannot be rejected. Hence it concludes that the rock genre is indeed the louder.

2. What's the relationship between Classical music and loudness?

Ho: Classical music has more loudness
Ha: Classical music doesn't have more loudness

```
        One-sample z-Test

data:  data3
z = -6.5618, p-value = 5.315e-11
alternative hypothesis: true mean is not equal to -7.832886
95 percent confidence interval:
 -16.73499 -12.64013
sample estimates:
mean of x
-14.68756
```

As per the results, the observed Z-score is -6.5618 and P-value is 5.315e-11. Since the observed Z-score is above the critical value and, the P-value is very less than 0.05 so the null hypothesis can be rejected. Hence it concludes that classical music is not very loud.

3.1.3 ANOVA Testing

Spotify has some of everything, from the century's old classical pieces to the latest experiments from a band in some garage in Seattle. Are some genres less well-liked than others?
Ho: The ratings of songs are the same regardless of the songs' genre on Spotify.
Ha: The ratings of songs will be high or low depending on whether they are from a popular genre.
alpha = 0.05

```
```{r genre ratings ANOVA}
ANOVA1 <- aov(rating_percent~main_genre, data = df)
summary(ANOVA1)
```
```

```
              Df  Sum Sq Mean Sq F value Pr(>F)
main_genre    14   18357  1311.2   1.604 0.0704 .
Residuals   3192 2609206   817.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since P-Value is greater than 0.05, Hence, it concludes that the null hypothesis can be rejected.

```r
```{r genre ratings TukeyHSD}
TukeyHSD(ANOVA1, conf.level = .95)
```
```

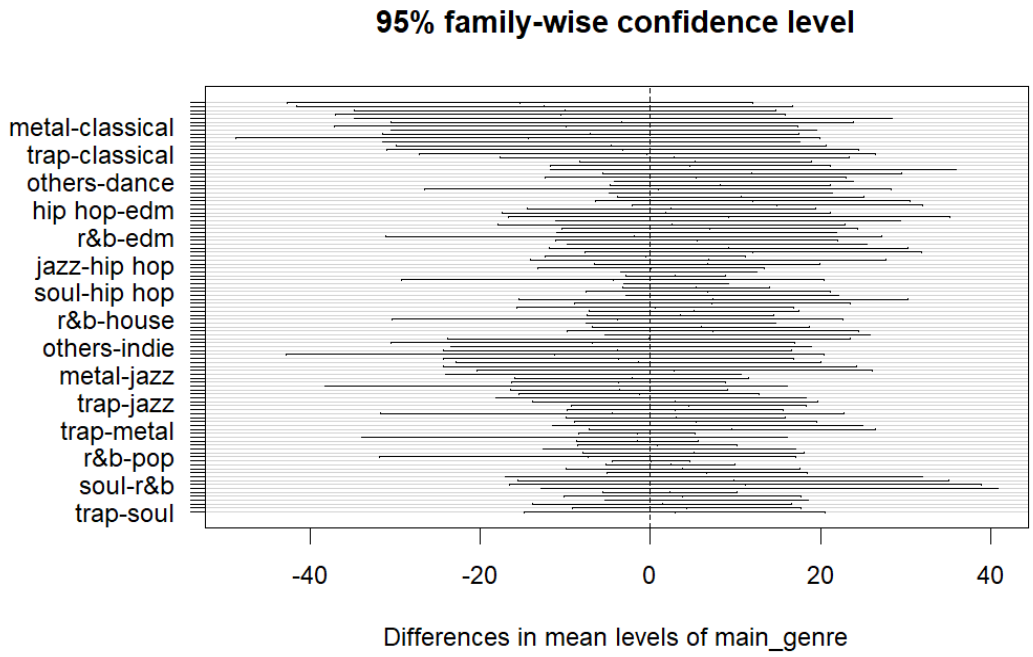
```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = rating_percent ~ main_genre, data = df)

$main_genre
                          diff        lwr       upr     p adj
dance-classical   -15.26483051 -42.615606 12.085945 0.8563892
edm-classical     -12.43055556 -41.585716 16.724605 0.9829248
hip hop-classical  -9.98975410 -34.772884 14.793376 0.9899187
house-classical   -10.56547619 -37.033735 15.902782 0.9908238
indie-classical    -3.17934783 -34.768168 28.409472 1.0000000
jazz-classical     -3.30357143 -30.468482 23.861339 1.0000000
metal-classical    -9.93951613 -37.148760 17.269728 0.9962213
others-classical   -5.45524691 -30.499712 19.589218 0.9999907
pop-classical      -7.01785714 -31.438955 17.403241 0.9997310
r&b-classical     -14.37500000 -48.681784 19.931784 0.9854560
rap-classical      -6.94982993 -31.471007 17.571347 0.9997710
rock-classical     -4.61184211 -29.871172 20.647488 0.9999990
soul-classical     -3.22115385 -30.961878 24.519571 1.0000000
```

A post-hoc test upholds the results of the ANOVA. Zero falls within the realm of the confidence interval for every comparison test, so the null hypothesis stands.

```r
```{r genre rating Tukey plot}
par(mar=c(5, 7.1, 3.1, 2.1))
plot(TukeyHSD(ANOVA1, conf.level = .95), las = 1)
par(mar=c(5, 4, 4, 2) + 0.1) # reset margins to default to not mess with following plots
```
```



3.2 Regression Analysis

a) What is a Regression Analysis?

Regression is a statistical technique used in the fields of statistics and other disciplines that aims to establish the nature and strength of the relationship between a single dependent variable (often represented by Y) and several independent variables (known as independent variables).

The most popular variation of this method is linear regression, which is also known as simple regression or ordinary least squares (OLS). Based on a line of best fit, linear regression determines the linear relationship between two variables. The slope of a straight line used to represent linear regression thus indicates how changing one variable effect changing another. In a linear regression connection, the value of one variable when the value of the other is zero is represented by the y-intercept. There are also non-linear regression models, although they are far more complicated. A potent method for identifying the relationships between variables in data is regression analysis

b) What is Multiple Linear Regression?

Multiple linear regression is used to estimate the relationship between two or more independent variables and one dependent variable. Multiple regression was used to predict popularity/ratings based on song features. In the given case, track popularity/rating is the dependent variable, and multiple linear regression is used to predict its value based on the song's qualities as the independent variable.

Firstly, an initial model was created, and a regression model was applied. Creating a multiple linear regression model with track popularity/rating value as the response variable and danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo and duration_ms as the covariates.

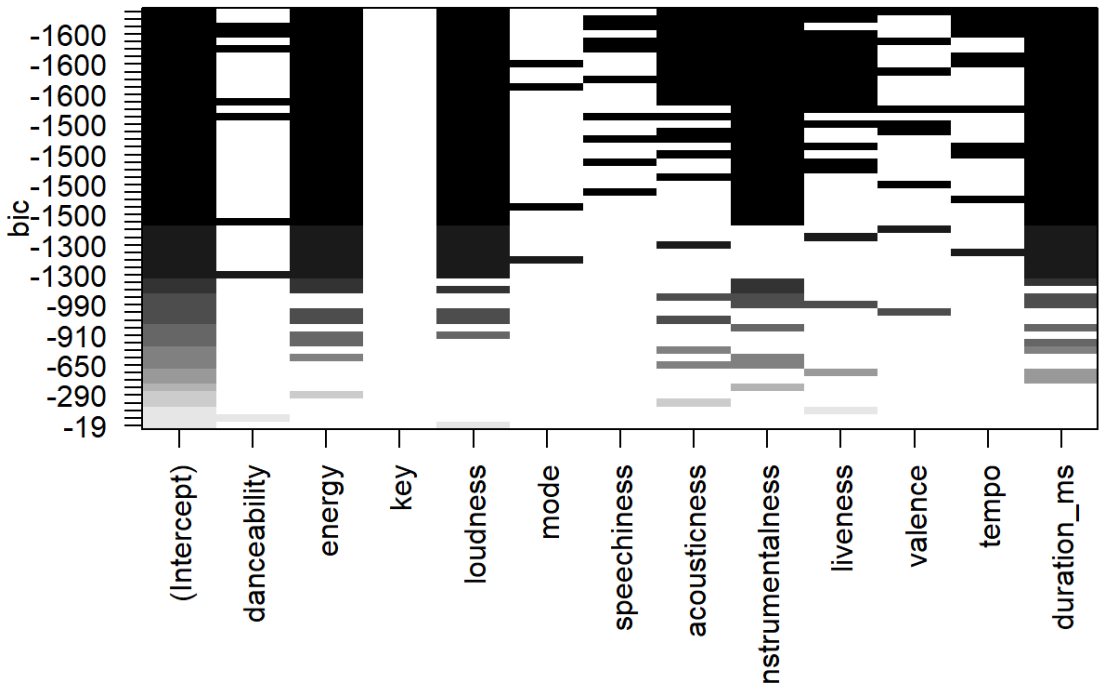
```{r}
model_1 <- lm(rating_percent ~ danceability + energy +key + loudness + mode + speechiness + acousticness + instrumentalness +
liveness + valence + tempo + duration_ms,
              data = df)

summary(model_1)
```

```
Call:
lm(formula = track_popularity ~ danceability + energy + key +
   loudness + mode + speechiness + acousticness + instrumentalness +
   liveness + valence + tempo + duration_ms, data = spotify_data_4)

Residuals:
  Min     1Q Median     3Q    Max
-54.62 -17.22   2.95  18.10  60.54

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       6.783e+01  1.704e+00  39.805  < 2e-16 ***
danceability      3.721e+00  1.072e+00   3.470 0.000522 ***
energy           -2.321e+01  1.220e+00 -19.028  < 2e-16 ***
key               3.190e-03  3.844e-02   0.083 0.933870
loudness          1.156e+00  6.527e-02  17.711  < 2e-16 ***
mode              8.616e-01  2.809e-01   3.067 0.002161 **
speechiness      -6.328e+00  1.380e+00  -4.587 4.52e-06 ***
acousticness      4.331e+00  7.466e-01   5.801 6.67e-09 ***
instrumentalness -9.292e+00  6.255e-01 -14.856  < 2e-16 ***
liveness         -4.280e+00  8.990e-01  -4.761 1.93e-06 ***
valence           1.788e+00  6.565e-01   2.724 0.006458 **
tempo             2.609e-02  5.239e-03   4.979 6.42e-07 ***
duration_ms      -4.342e-05  2.294e-06 -18.925  < 2e-16 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.01 on 28343 degrees of freedom
Multiple R-squared:  0.05808,    Adjusted R-squared:  0.05768
F-statistic: 145.6 on 12 and 28343 DF,  p-value: < 2.2e-16
```

It can be noticed that all the covariates in the model are significant except the key since the p-value for each of them is less than 0.05. Besides, the Adjusted R- squared value is *0.05768* which is moderate. a p-value of the model is **< 2.2e-16** suggesting all the results are significant. Performing a variable selection process to identify the significant covariates.

Selecting the required variables.

```{r}
model_3 = regsubsets(rating_percent ~ danceability + energy + key + loudness + mode + speechiness + acousticness +
instrumentalness + liveness + valence + tempo + duration_ms,
             data = df,
             nbest = 8)

plot(model_3, scale = "bic")
```



According to the best subset selection, the influence of 'Energy' > 'Loudness'. Upon comparing both these results one can conclude that all variables except 'key' are statistically significant in predicting the track popularity. Also, the p-value for the column 'key' is greater than 0.05.

```{r}
model_2 <- lm(rating_percent ~ danceability + energy + loudness + mode + speechiness + acousticness + instrumentalness + liveness
+ valence + tempo + duration_ms,
              data = df)

summary(model_2)
```

```
Call:
lm(formula = track_popularity ~ danceability + energy + loudness +
    mode + speechiness + acousticness + instrumentalness + liveness +
    valence + tempo + duration_ms, data = spotify_data_4)

Residuals:
    Min      1Q  Median      3Q     Max
-54.624 -17.226   2.949  18.099  60.533

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       6.785e+01  1.692e+00  40.113  < 2e-16 ***
danceability      3.720e+00  1.072e+00   3.469 0.000523 ***
energy           -2.321e+01  1.219e+00 -19.030  < 2e-16 ***
loudness          1.156e+00  6.527e-02  17.712  < 2e-16 ***
mode              8.576e-01  2.765e-01   3.101 0.001930 **
speechiness      -6.326e+00  1.379e+00  -4.586 4.54e-06 ***
acousticness      4.331e+00  7.465e-01   5.803 6.60e-09 ***
instrumentalness -9.292e+00  6.254e-01 -14.856  < 2e-16 ***
liveness         -4.280e+00  8.990e-01  -4.761 1.93e-06 ***
valence           1.789e+00  6.564e-01   2.726 0.006414 **
tempo             2.608e-02  5.239e-03   4.979 6.44e-07 ***
duration_ms      -4.342e-05  2.294e-06 -18.928  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.01 on 28344 degrees of freedom
Multiple R-squared:  0.05808,    Adjusted R-squared:  0.05771
F-statistic: 158.9 on 11 and 28344 DF,  p-value: < 2.2e-16
```

The adjusted R- squared value is *0.05717*. This implies that the model can predict the track popularity and can explain 5.71% of the variation in the data set. With this model, one can check the accuracy of the mode.

```{r}
par(mfrow = c(1,2))
# generate QQ plot
qqnorm(model_2$residuals,main = "Model")
qqline(model_2$residuals)

# generate Scatter Plot
plot(model_2$fitted.values,model_2$residuals,pch = 20)
abline(h = 0,col = "grey")
```

Observing the values one can see that popularity is less than the observed values. The variation exists because of skewness in the data. The model needs to be transformed to make accurate predictions about the popularity.

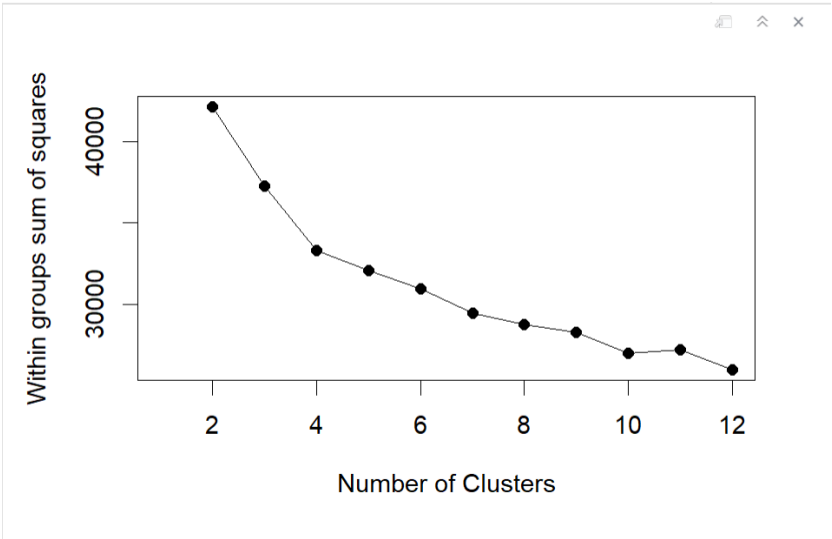
3.3 Data Modelling – K Squared Mean Clustering

For Spotify Music Dataset, an unsupervised machine learning model known as K-squared mean clustering was used to find the hidden clusters among the songs based on attributes like danceability, acusticness, etc which will be used to find the next song similar to the previously played song. Attributes like danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_ms, and genre were used to find more precise results.

```{r}
spotify_clu <- scale(df[,c(11,12,13,14,16,17,18,19,20,21,22)])
spotify_scaled <- cbind(spotify_clu, df$genr_new)

summary(spotify_scaled)
```

```
 danceability          energy              key              loudness      
Min.   :-3.49533   Min.   :-2.80328   Min.   :-1.42226   Min.   :-12.4846  
1st Qu.:-0.63753   1st Qu.:-0.60565   1st Qu.:-0.86418   1st Qu.: -0.3172  
Median : 0.09942   Median : 0.08688   Median :-0.02706   Median :  0.2474  
Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   :  0.0000  
3rd Qu.: 0.72667   3rd Qu.: 0.77018   3rd Qu.: 0.81006   3rd Qu.:  0.6263  
Max.   : 2.01775   Max.   : 1.80436   Max.   : 1.64718   Max.   :  2.3312  
 speechiness       acousticness     instrumentalness     liveness       
Min.   :-0.9007   Min.   :-0.8899   Min.   :-0.3513   Min.   :-2.8910  
1st Qu.:-0.6018   1st Qu.:-0.8011   1st Qu.:-0.3513   1st Qu.:-0.8168  
Median :-0.4459   Median :-0.4463   Median :-0.3513   Median :-0.1749  
Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  
3rd Qu.: 0.2001   3rd Qu.: 0.5652   3rd Qu.:-0.3425   3rd Qu.: 1.1758  
Max.   : 6.8607   Max.   : 2.4538   Max.   : 4.1366   Max.   : 1.4802  
    valence             tempo           duration_ms            V12        
Min.   :-2.01807   Min.   :-3.95574   Min.   :-2.24477   Min.   : 1.00  
1st Qu.:-0.80113   1st Qu.:-0.79214   1st Qu.:-0.40891   1st Qu.: 9.00  
Median :-0.04717   Median :-0.06989   Median :-0.07669   Median :10.00  
Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 9.53  
3rd Qu.: 0.75139   3rd Qu.: 0.67541   3rd Qu.: 0.29670   3rd Qu.:12.00  
Max.   : 2.13611   Max.   : 3.14128   Max.   :36.87675   Max.   :15.00  
```

The Elbow method was used to find the right number of clusters to form, as shown in the below diagram, K=4 was concluded, further, the dataset was dived into four clusters.

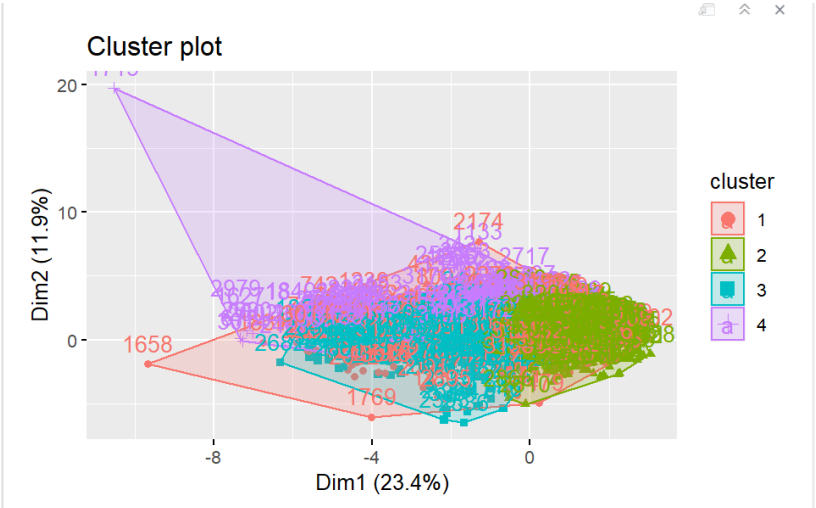


The four clusters' centroids are shown in the below table.

```{r}
spotify_kmeans$centers
```

	danceability	energy	key	loudness	speechiness	acousticness
1	0.08891358	-0.0282202	-0.09956631	-0.09875406	0.002856245	-0.0419617
2	0.25360871	0.4095639	0.02013392	0.40099083	0.128371570	-0.4298381
3	-0.63462188	-1.2087511	0.03364517	-0.87866958	-0.278965751	1.3452435
4	-0.87204584	-0.3493260	0.04058599	-0.95590488	-0.423001556	0.3844030

	instrumentalness	liveness	valence	tempo	duration_ms	
1	0.07136906	0.04008556	0.05472221	-0.01211485	-0.15518949	4.130709
2	-0.29464635	0.03518975	0.19608636	0.06574055	-0.02860048	11.041051
3	-0.25099801	-0.06623337	-0.51645172	-0.18780851	0.04204018	10.394881
4	3.18331032	-0.27028455	-0.55807720	-0.04890812	0.64546675	10.510101



As shown in the code, when the user will insert the song name and artist name for example here, he has entered “Stay” and “Hungry Lucy” then he will get 5 recommended songs based on his choice of songs.

Here, firstly the cluster of the given song is found and then on that basis algorithm finds the five closest songs to this song from the same cluster. Further, the number of songs to recommend is also customizable, which will give the user the flexibility to choose the number of songs to get recommended.

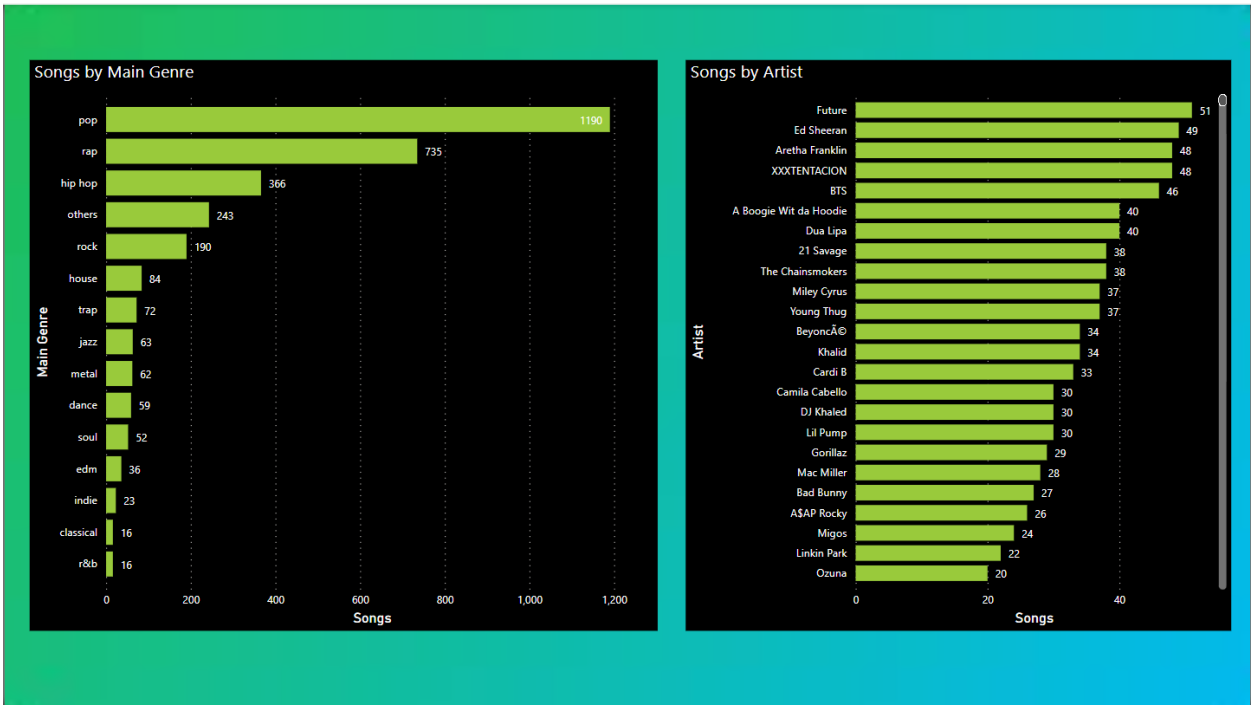
```
# Input Song Name and Artist Name
```{r}
df %>%
  filter(song_name == "Stay", artist == "Hungry Lucy")
clust <- df$cluster[df$song_name == "Stay" & df$artist =="Hungry Lucy"]
clust
```
```

Description: df [5 × 29]

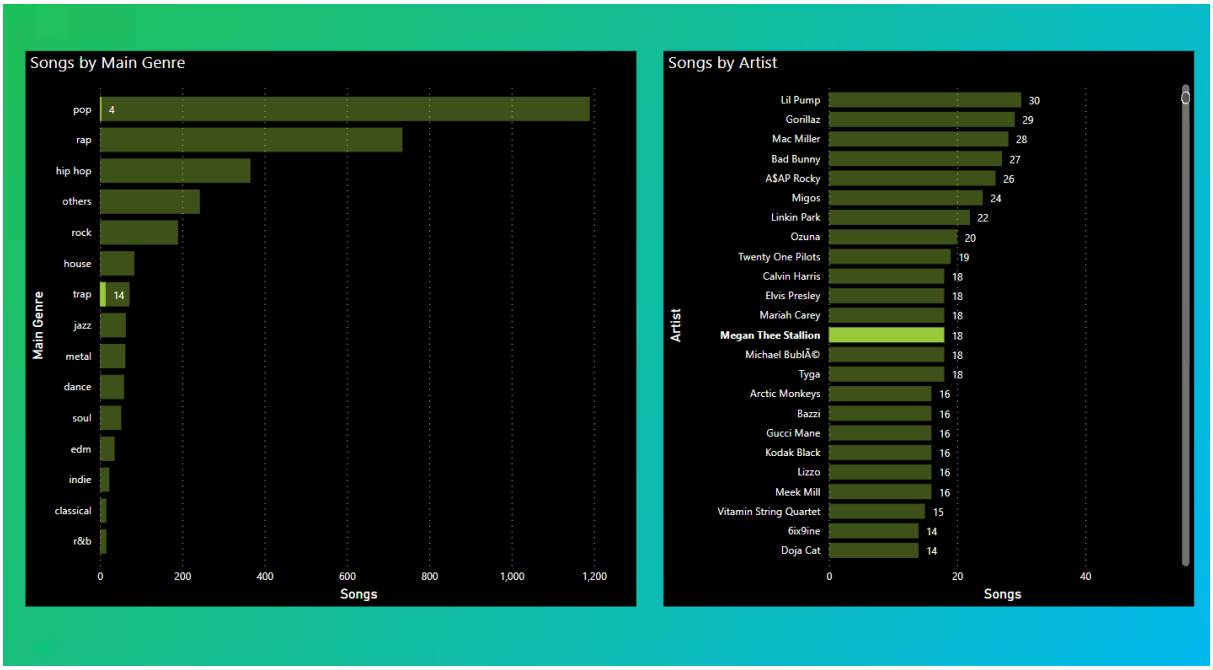
song_id <chr>	song_name <chr>
2xfy66cc1pwdNRCAb1J6L9	halo
103NIOjApz302KrrITpFiL	Winter Wonderland
4kJewF3bbkPTktz702H56J	Silver Bells
2nIkEazEHtNYs7vRE7DYmL	Christmas Time Is Here
12VqHHz4wvVcnEdSivjLeQ	Alive

5 rows | 1-2 of 29 columns

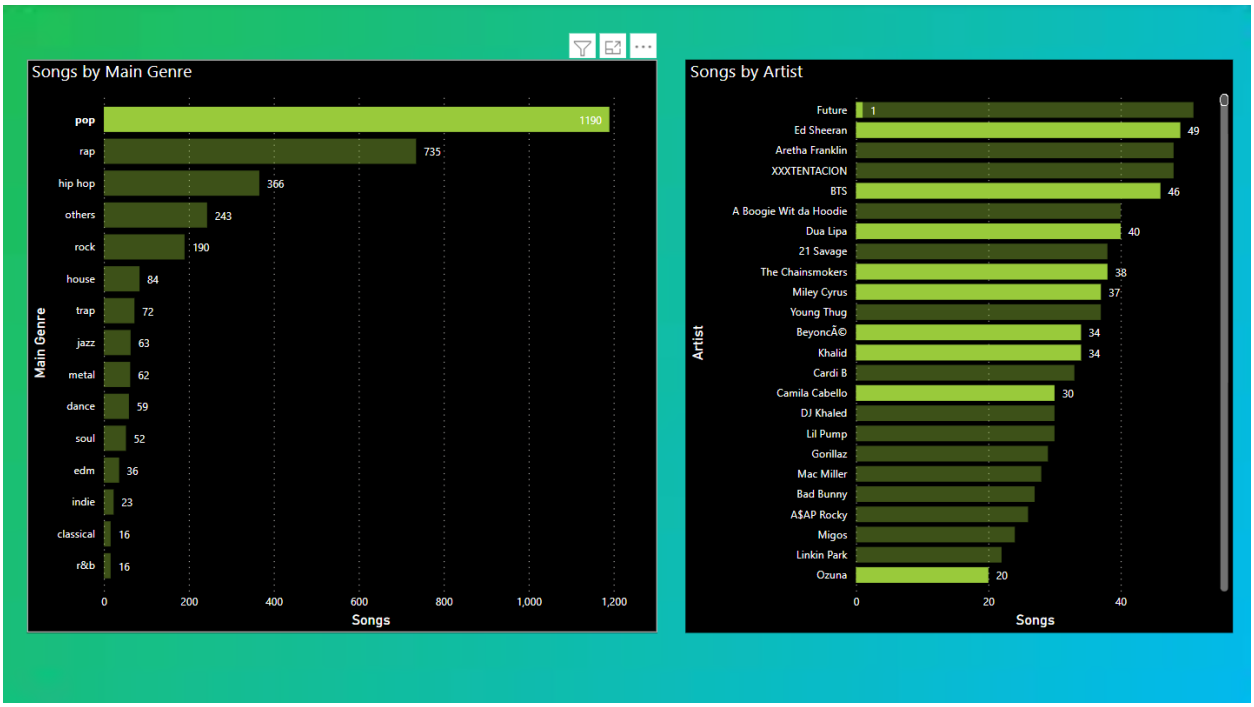
3.4 Interactive Dashboard



Page 1: The stats page is the first page that shows the breakdown of data based on genres or artists. In this, an artist can be selected on the right graph to show their contributions to eat genre. As shown below, Megan Thee Stallion released 4 pop and 14 trap genre songs.

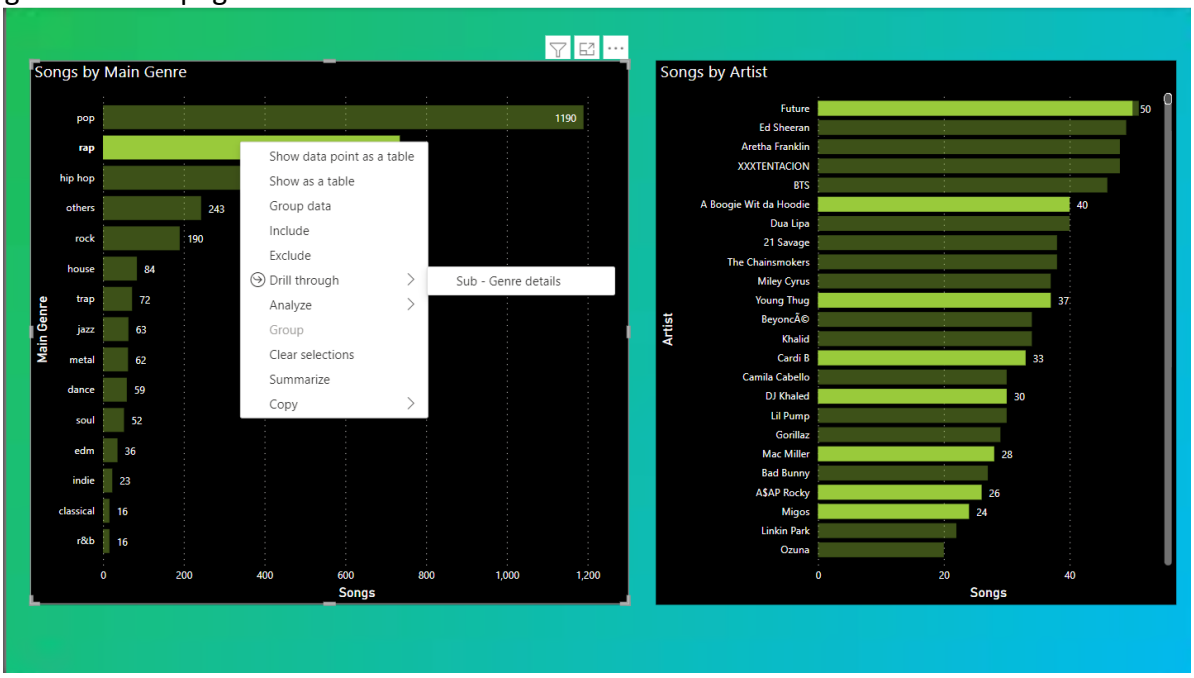


Similarly, a particular genre can be selected, for instance, pop has 1 song released by Future, 49 songs by Ed Shereen, 46 songs by BTS, 40 songs by Dua Lipa etc.



Page 2: Subgenre

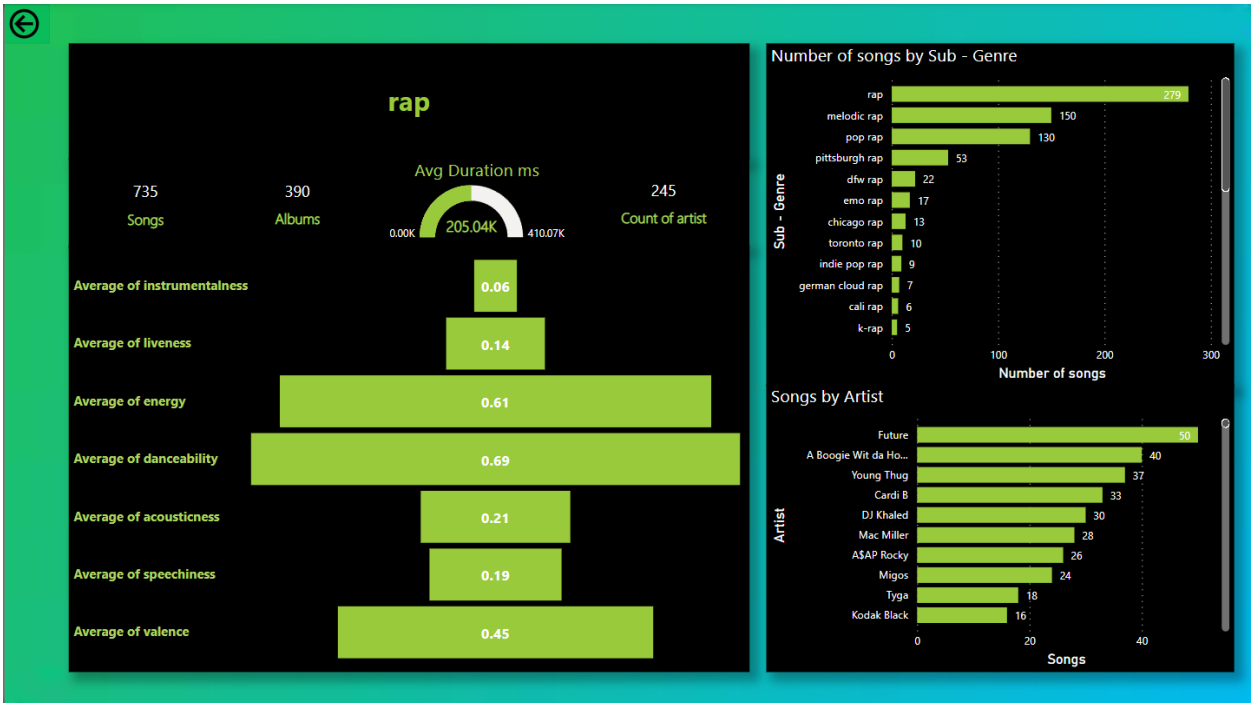
To explore more into the subgenre more, the drill-through option is used to go to the sub-genre details page.



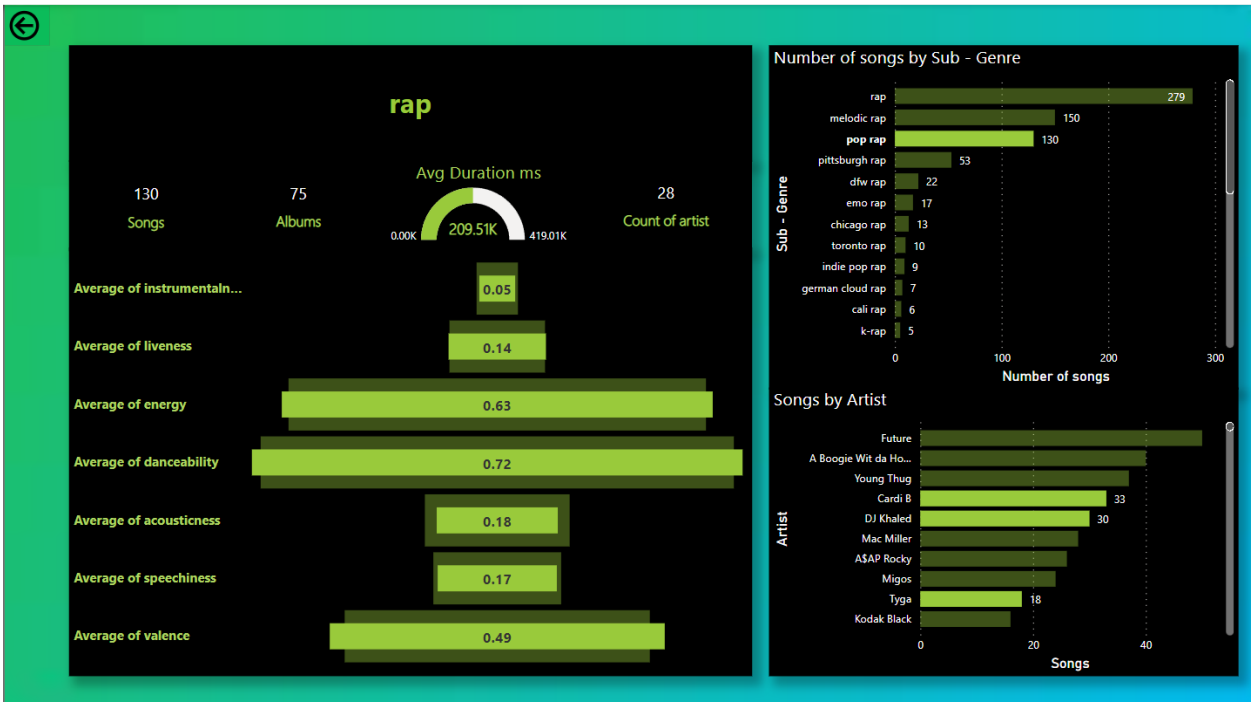
This takes to the next page which describes the aggregate matrices of each main genre and its subgenres for the selected rap main genre.

There are 735 total songs, and 390 total albums, the average duration in milliseconds of a rap song is 205,000 milliseconds, and 245 total artists are contributing to raps of the genre this page also contains statistics like average instrumentalness, average liveliness, average energy, average danceability, average acousticness, average speechiness and average valance.

The right-hand side graphs described the number of songs in each subgenre of the selected main genre, for instance, rap main genre contains subgenre rap with 279 songs melodic rap with 150 songs pop rap with 130 songs, etc. It also describes artists who have contributed to rap subgenres like future has contributed 50 songs Cardi B with 33 songs and Mac Miller with 28 songs.

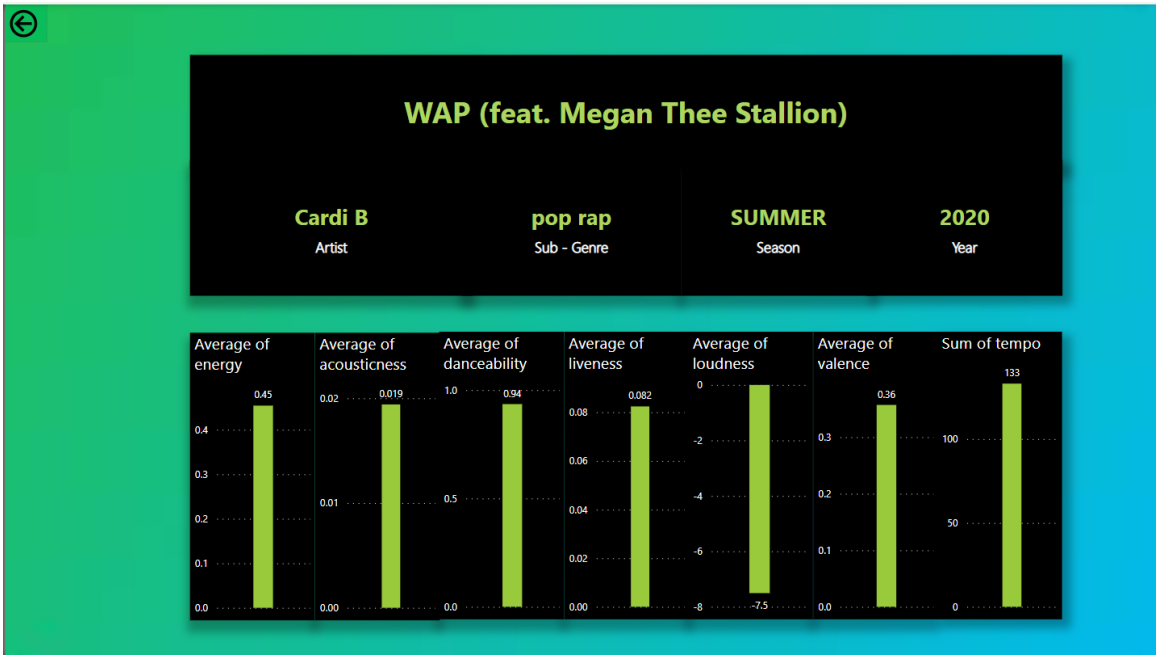


Each artist can be selected to show which subgenre of rap they contributed to like Cardi B contributed 33 songs to the pop-rap subgenre and the statistics change according to the artist selected and/or the sub-genre selected.

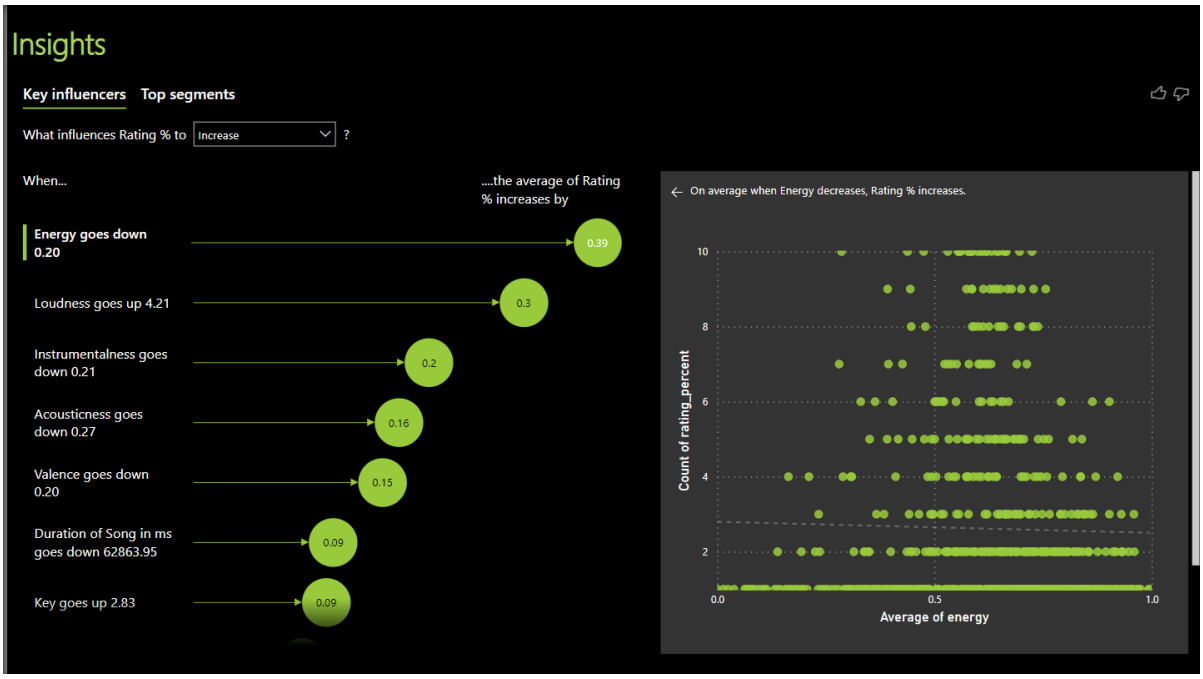


Furthermore, we can drill through each artist from this main genre details page or the previous Page 1 to give us details of the artist. The following Chuck shows aggregate statistics of the selected artist, for instance, Cardi B, released 33 songs in total across nine albums with an average duration of 213,000 milliseconds in the subgenre pop rap the chart also describes features like average instrumentalness, etc. The right-hand side describes her contribution to each subgenre and the rating percentage of her songs.

A particular song from this artist can be selected to explore more about that song. This gives us information like what sub-genre the song belongs to, the season was it released, the year of release of the song, and various matrices of the song.

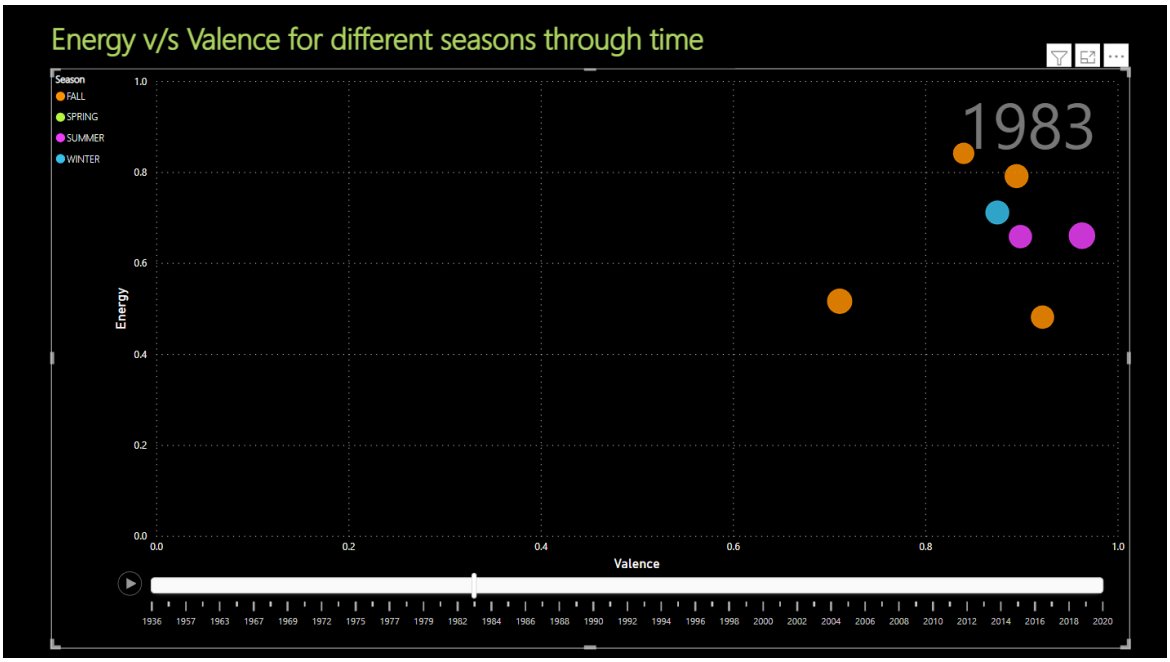


The next slide titled insights describes how each metric of a song affects its rating percentage and how to maximize the rating percentage based on these matrices. As stated below energy has the maximum impact on the rating percentage of a song such that if the energy goes down by 0.2 from the average energy the rating percentage increases by 0.39. the graph on the right side describes the regression line that was used to infer the previous statement.



The next slide titled maximum rating percentage helps an artist choose various art attributes to create a song that would maximize rating percentage. For instance, if an artist chooses to release an explicit song with zero instrumentals their success would be maximized in the rap genre with a key one.

The next graph describes energy versus valence for different seasons through time. The data set has 7 songs released in 1983, four of them released in the fall season depicted in an orange colour called mom two of them released in summer depicted in purple and one of them in winter depicted in blue. the slider can be moved to change the ear and each bubble can be drilled through to get the song details.



4 Conclusion

Today, businesses hire data analysts to analyze their collected data and use the extracted information to know more about their consumers. The analysis of the data to draw useful insights can be easily done with various libraries and functions.

Spotify makes it super easy to analyze Spotify data for playlists, artists, songs and users and can be used in a variety of ways, be it for classification/recommendation systems, or just for fun if one is a data enthusiast. This project aimed to learn to analyze music data, create interesting visualizations, find correlations and extract useful insights using the Spotify dataset.

Spotify music analytics and Recommendation Projects can be used primarily for two types of people: first, Spotify users who want recommendations for their next songs, and second, music composers who can use visualizations and data to understand what the market wants and what types of songs are most likely to be successful. This project also helps emerging artists understand current trends and shifts in trends, which can assist with questions like how much funding should be allocated to specific projects.

One of the well-known issues this recommender system has is that it can lead to a unification of people's tastes; consequently, the most well-liked song or song by the most well-known artist has a much higher likelihood of being recommended than the song by a new artist. The objective of this project is to find a solution and improve the fairness of the Spotify platform going forward. Having said that, the majority of people's frustration with choosing songs from a vast dataset of songs will be alleviated by this current system.

5 References

[1] Mar a Dolores Ugarte, Ana F. Militino, Alan T. Arnholt Probability and Statistics with R, CRC Press
[2] https://www.statsforspotify.com
[3] Who will win Eurovision Music Award?
[4] To build a music recommendation system with Spotify
[5] Spotify Dataset on Kaggle
[6] What Is The Secret of Spotify's Recommendation System Success?
[7] https://towardsdatascience.com/create-music-recommendation-system-using-python-ce5401317159
[8] https://rstudio-pubs-static.s3.amazonaws.com/605392_421bd6cab4674f6e87cea7ebf593d0e1.html