Northeastern

MPS Analytics

ALY6010 : Probability Theory and Introductory Statistics

Module 6

Final project - Executive Summary Report

Topic: Spotify Dataset Analysis

Prepared By : Shyamala Venkatakrishnan

Date: 12/16/2022

Introduction

Spotify is one of the largest music platforms which has millions of songs from different languages and over 200 million monthly subscribers from all over the world. It provides high quality user experience and generates music content based on the user's interest and preferences. Spotify makes this possible by making use of AI and ML algorithms in order to create user specific song lists based on the user's favorite genre and albums. Spotify platform stores the features of each song along with its popularity score voted by the users. These song features are calculated either by ML models or using sophisticated tools and technologies. These features can be analyzed to understand different types of songs and gain some insights about the top popular songs. Spotify continues to rule the music streaming industry because of its ability to better understand its customers' preferences and recommend songs based on the user's history and listening patterns.

In this project, Spotify dataset from the Kaggle platform is to be analyzed to find the relationship between various song features, visualization and charts are to be generated to gain some useful insights from the dataset, hypothesis testing is to be conducted to validate a claim or hypothesis, linear regression is to be applied in order to predict the target variable based on various independent song attributes.

Dataset Definition

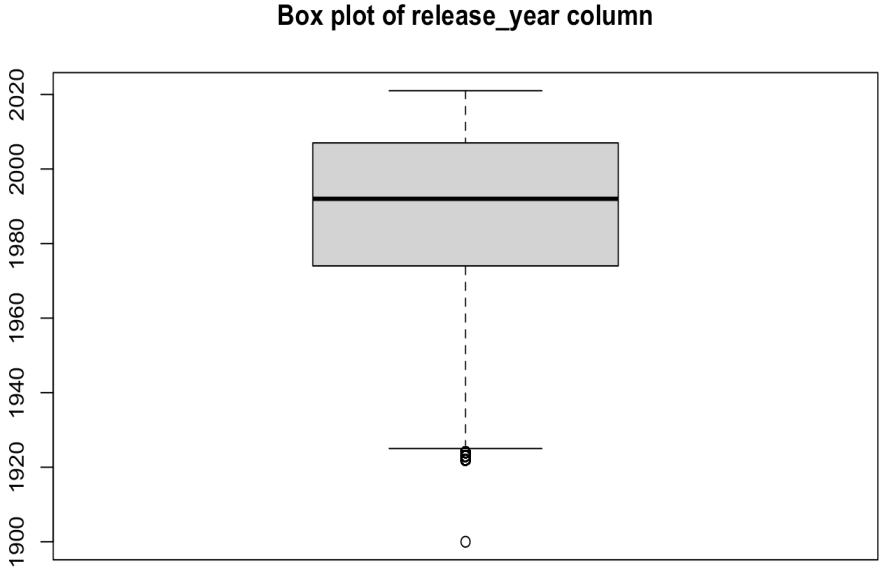
	id	name	popularity	duration_ms	explicit	artists	id_artists	release_date	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumental	liveness	valence	tempo	time_signature
2	35iwgR4jXetI318W	Carve	6	126903	0	['Uli']	['45tIt06XoI0	2/22/22	0.645	0.445	0	-13.338	1	0.451	0.674	0.744	0.151	0.127	104.851	3
3	021ht4sdgPcrDgSk	Capv#tulo 2.	0	98200	0	['Fernando P	['14jtPCOoNZ	6/1/22	0.695	0.263	0	-22.136	1	0.957	0.797	0	0.148	0.655	102.009	1
4	07A5yehtSnoedViJ	Vivo para Qu	0	181640	0	['Ignacio Cor	['5LiOoJbxVS	3/21/22	0.434	0.177	1	-21.18	1	0.0512	0.994	0.0218	0.212	0.457	130.418	5
5	08FmqUhxtyLTn6p	El Prisionero	0	176907	0	['Ignacio Cor	['5LiOoJbxVS	3/21/22	0.321	0.0946	7	-27.961	1	0.0504	0.995	0.918	0.104	0.397	169.98	3
6	08y9GfoqCWfOGsI	Lady of the E	0	163080	0	['Dick Hayme	['3BiJGZsyX9	1922	0.402	0.158	3	-16.9	0	0.039	0.989	0.13	0.311	0.196	103.22	4
7	0BRXJHRNGQ3W4	Ave Maria	0	178933	0	['Dick Hayme	['3BiJGZsyX9	1922	0.227	0.261	5	-12.343	1	0.0382	0.994	0.247	0.0977	0.0539	118.891	4
8	0Dd9ImXtAtGwsm	La Butte Rou	0	134467	0	['Francis Ma	['2nuMRGze	1922	0.51	0.355	4	-12.833	1	0.124	0.965	0	0.155	0.727	85.754	5
9	0IA0Hju8CAgYfV1h	La Java	0	161427	0	['Mistinguett	['4AxgXfD7IS	1922	0.563	0.184	4	-13.757	1	0.0512	0.993	1.55E-05	0.325	0.654	133.088	3
10	0Igl1UCz84pYeVet	Old Fashione	0	310073	0	['Greg Fieler	['5nWIsH5RI	1922	0.488	0.475	0	-16.222	0	0.0399	0.62	0.00645	0.107	0.544	139.952	4

```
     id                 name              popularity      duration_ms         explicit         artists
 Length:586672      Length:586672      Min.   :  0.00   Min.   :   3344   Min.   :0.00000   Length:586672
 Class :character   Class :character   1st Qu.: 13.00   1st Qu.: 175093   1st Qu.:0.00000   Class :character
 Mode  :character   Mode  :character   Median : 27.00   Median : 214893   Median :0.00000   Mode  :character
                                       Mean   : 27.57   Mean   : 230051   Mean   :0.04409
                                       3rd Qu.: 41.00   3rd Qu.: 263867   3rd Qu.:0.00000
                                       Max.   :100.00   Max.   :5621218   Max.   :1.00000
  id_artists         release_date        danceability        energy            key            loudness
 Length:586672      Length:586672      Min.   :0.0000   Min.   :0.000   Min.   : 0.000   Min.   :-60.000
 Class :character   Class :character   1st Qu.:0.4530   1st Qu.:0.343   1st Qu.: 2.000   1st Qu.:-12.891
 Mode  :character   Mode  :character   Median :0.5770   Median :0.549   Median : 5.000   Median : -9.243
                                       Mean   :0.5636   Mean   :0.542   Mean   : 5.222   Mean   :-10.206
                                       3rd Qu.:0.6860   3rd Qu.:0.748   3rd Qu.: 8.000   3rd Qu.: -6.482
                                       Max.   :0.9910   Max.   :1.000   Max.   :11.000   Max.   :  5.376
      mode          speechiness      acousticness    instrumentalness       liveness         valence
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0340   1st Qu.:0.0969   1st Qu.:0.0000000   1st Qu.:0.0983   1st Qu.:0.3460
 Median :1.0000   Median :0.0443   Median :0.4220   Median :0.0000245   Median :0.1390   Median :0.5640
 Mean   :0.6588   Mean   :0.1049   Mean   :0.4499   Mean   :0.1134508   Mean   :0.2139   Mean   :0.5523
 3rd Qu.:1.0000   3rd Qu.:0.0763   3rd Qu.:0.7850   3rd Qu.:0.0095500   3rd Qu.:0.2780   3rd Qu.:0.7690
 Max.   :1.0000   Max.   :0.9710   Max.   :0.9960   Max.   :1.0000000   Max.   :1.0000   Max.   :1.0000
     tempo        time_signature
 Min.   :  0.0   Min.   :0.000
 1st Qu.: 95.6   1st Qu.:4.000
 Median :117.4   Median :4.000
 Mean   :118.5   Mean   :3.873
 3rd Qu.:136.3   3rd Qu.:4.000
 Max.   :246.4   Max.   :5.000
```

- This dataset contains a total of 586672 songs and 20 variables/attributes which are the features of the song.
- There are 5 categorical variables namely id, name,artists,id_artists and release_date and the remaining are numerical variables.
- The meaning of these columns can be referred to in the Spotify developer platform where we can find the Web APIs to get information about various tracks,albums and other entities like audio books, genres, artists, etc.
- In addition to that,
    - The **name** column indicates the name of the song
    - The **popularity** indicates the popularity score of a song ,voted by millions of users of the Spotify Application,
    - The **explicit** column has a value of 0 or 1. A value of 1 indicates that a song contains any offensive,aggressive,violent content and needs parental guidance. A value of 0 indicates the absence of explicit content in that song.
    - The **artists** column indicates the name of the artists involved in a song.
    - The **release_date** column indicates the date on which the song was released.

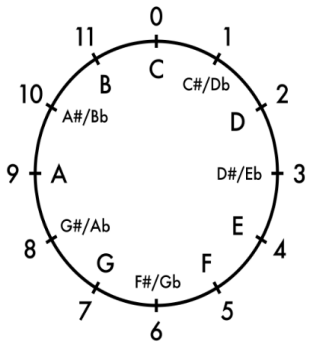
Data Cleaning

- Dropping the unwanted column : “id” and “id_artists”.
- **Creating a new column : release_year** to extract the year value from the release_date column.
- Removing the single quotes and brackets from the artists column to improve the readability of the values.
- **Creating a new column : duration_min** to hold the duration of the songs converted from milliseconds to minutes.
- Deleting the records of the songs released before the year 1930 since there are only a few hundreds of songs before 1930.

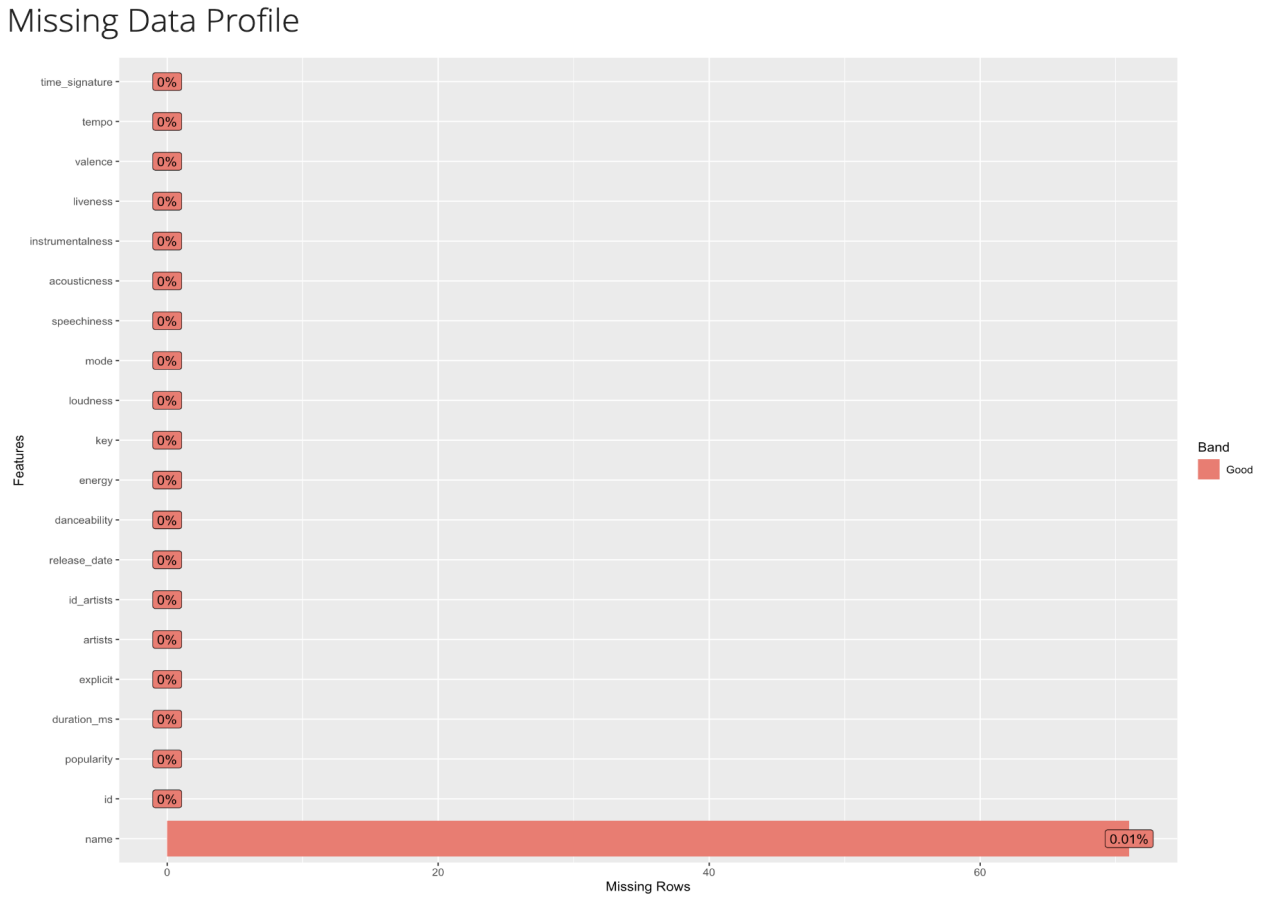


- **Creating a new column: Decade** to hold the value of the decade in which the song was released.
- **Creating a new column : key_class** to hold the mapping of the songs keys to its corresponding pitch class according to pitch_class notation.

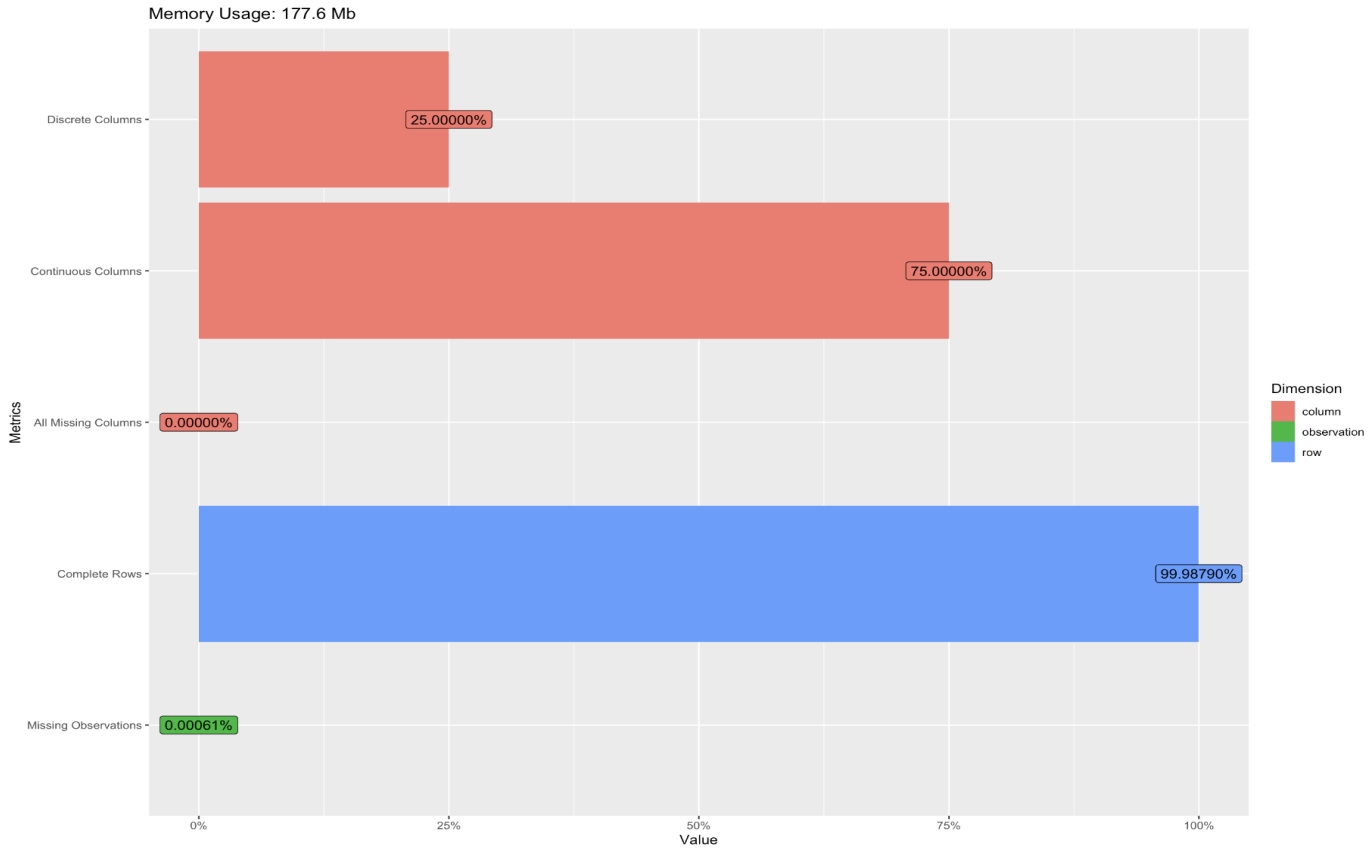
Mapping of songs keys with its corresponding pitch class



- There are only 70 records found with missing song names. There are no missing values found in other columns.

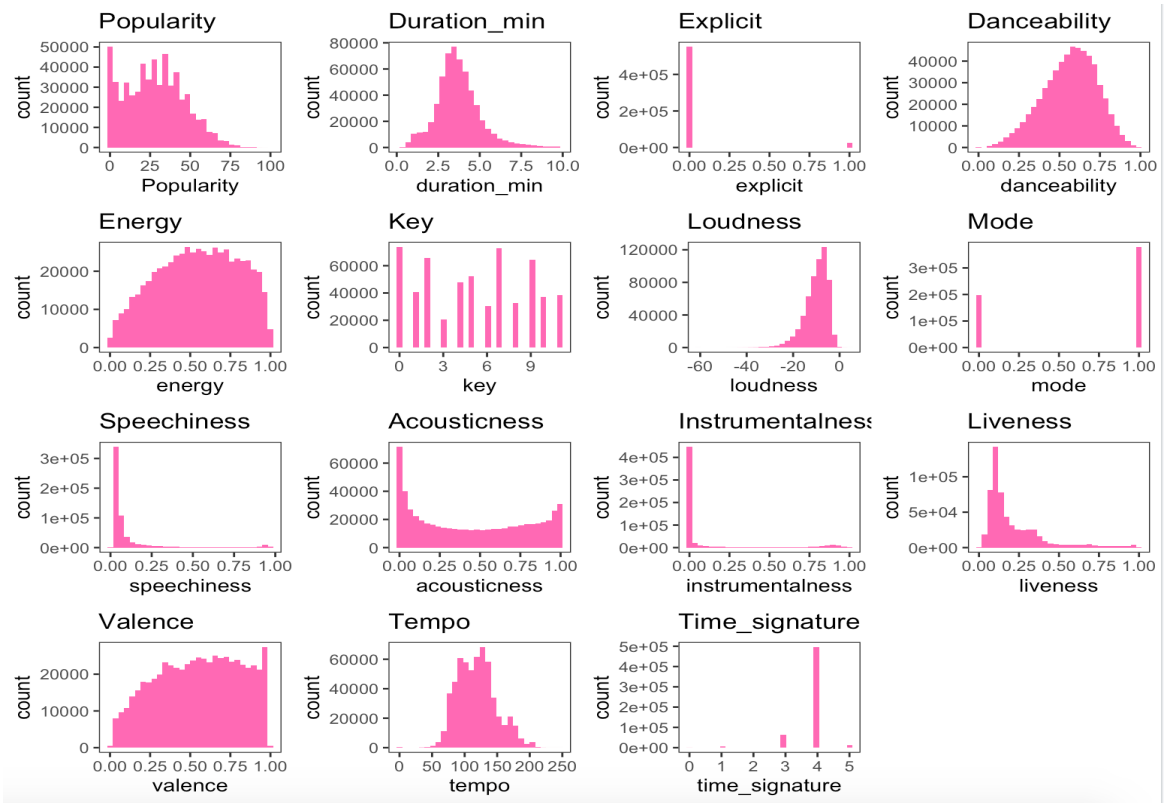


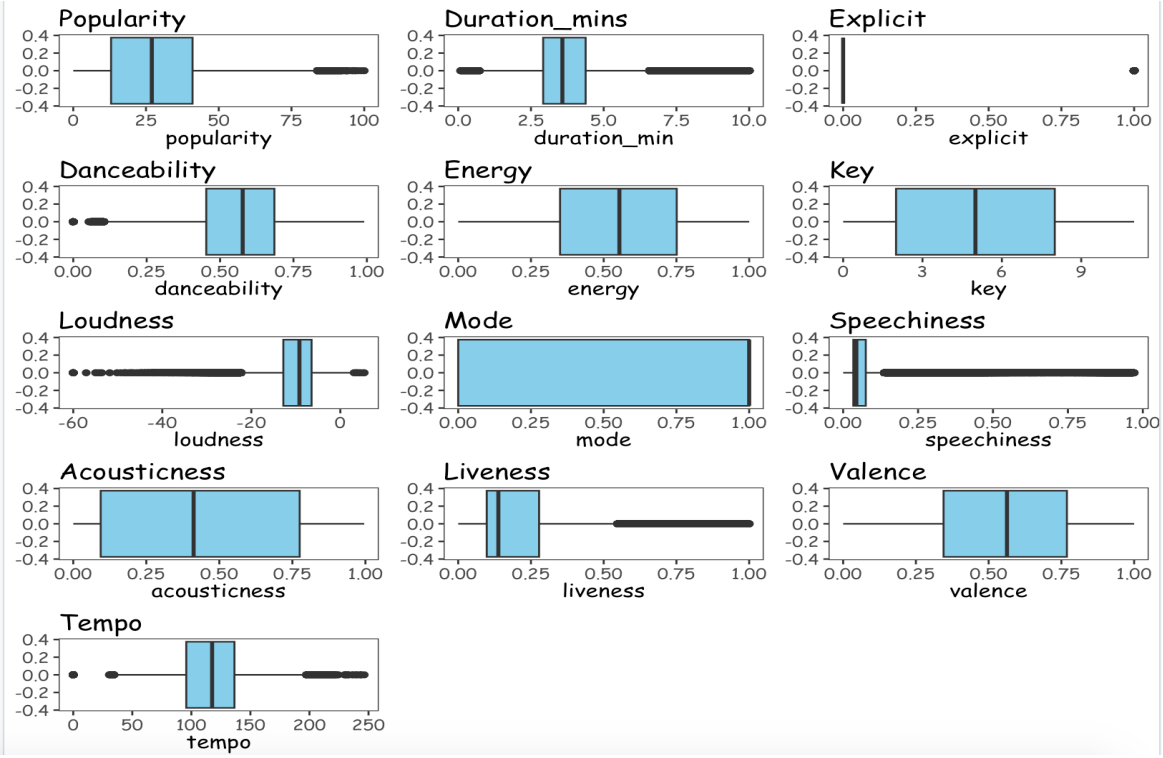
Summary information of the dataset:



Data Exploration

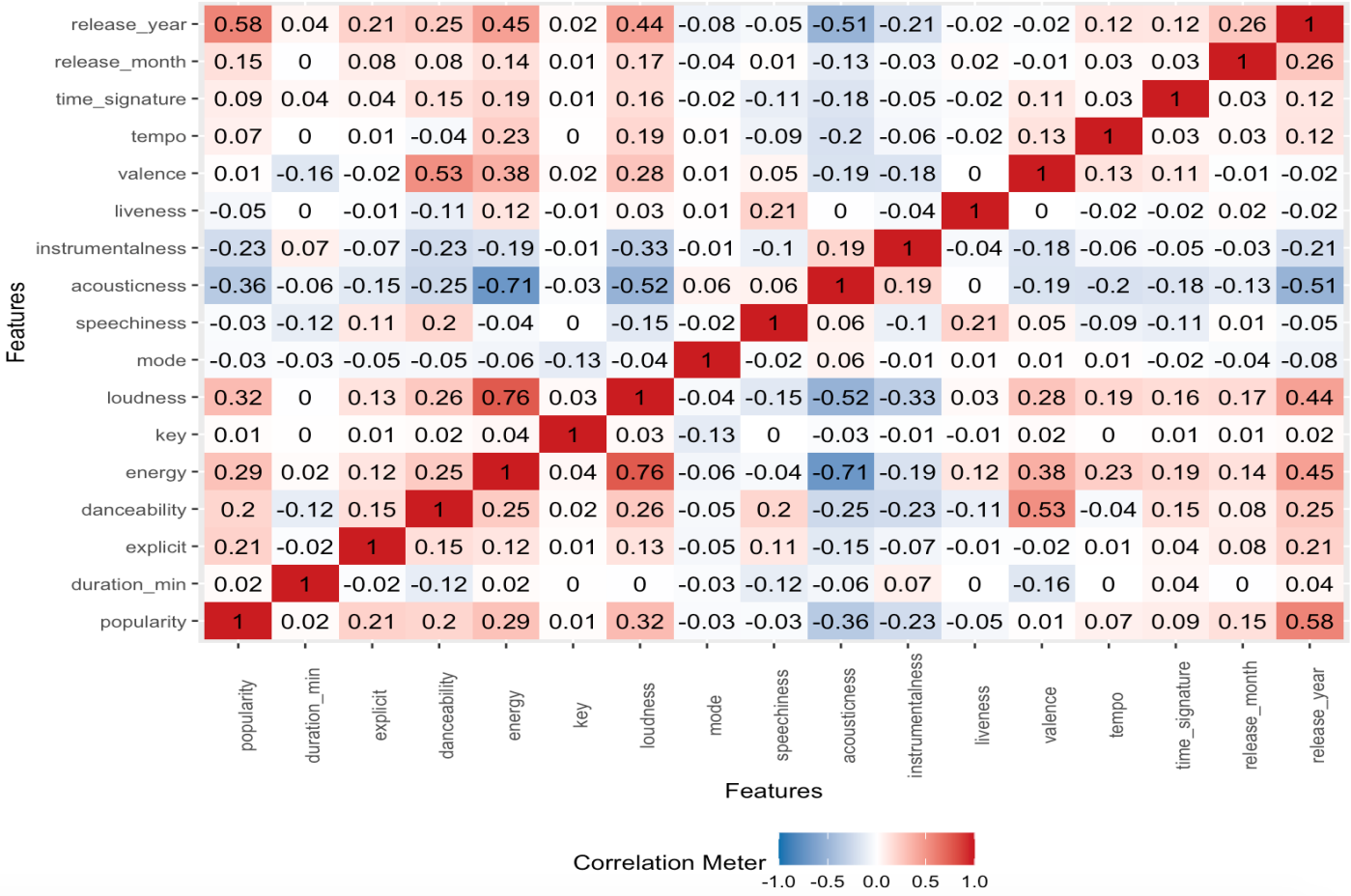
```
> describe(spotify_dataset)
                  vars      n     mean     sd      min      max   range   se
name                 1 576906      NaN     NA      Inf     -Inf    -Inf   NA
popularity           2 576906    27.92  18.25     0.00   100.00  100.00 0.02
duration_min         3 576906     3.84   2.12     0.08    93.69   93.61 0.00
explicit             4 576906     0.04   0.21     0.00     1.00    1.00 0.00
artists              5 576906      NaN     NA      Inf     -Inf    -Inf   NA
release_date         6 576906      NaN     NA      Inf     -Inf    -Inf   NA
danceability         7 576906     0.56   0.17     0.00     0.99    0.99 0.00
energy               8 576906     0.55   0.25     0.00     1.00    1.00 0.00
key                  9 576906     5.22   3.52     0.00    11.00   11.00 0.00
loudness            10 576906   -10.14   5.05   -60.00     5.38   65.38 0.01
mode                11 576906     0.66   0.47     0.00     1.00    1.00 0.00
speechiness         12 576906     0.10   0.17     0.00     0.97    0.97 0.00
acousticness        13 576906     0.44   0.35     0.00     1.00    1.00 0.00
instrumentalness    14 576906     0.11   0.26     0.00     1.00    1.00 0.00
liveness            15 576906     0.21   0.18     0.00     1.00    1.00 0.00
valence             16 576906     0.55   0.26     0.00     1.00    1.00 0.00
tempo               17 576906   118.54  29.72     0.00   246.38  246.38 0.04
time_signature      18 576906     3.87   0.47     0.00     5.00    5.00 0.00
release_month       19 576906     4.22   4.21     0.00    12.00   12.00 0.01
release_year        20 576906  1989.46  21.82  1930.00  2021.00   91.00 0.03
key_class           21 576906      NaN     NA      Inf     -Inf    -Inf   NA
Decade              22 576906      NaN     NA      Inf     -Inf    -Inf   NA
mode_class          23 576906      NaN     NA      Inf     -Inf    -Inf   NA
There were 13 warnings (use warnings() to see them)
```





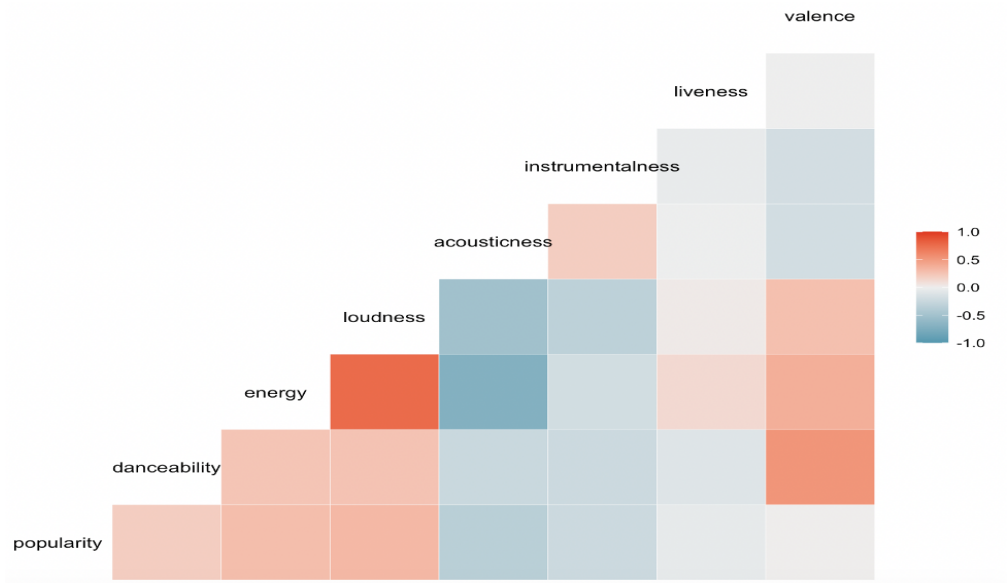
- The majority of the songs in this dataset have a popularity score of 0 to 50 which means that most of the songs in this dataset are not extremely popular.
- The duration of the songs is 3.5 to 4 mins on an average.
- The explicit content of the songs is present in only a few songs which means that most of the songs do not contain any offensive,violent,aggressive content and do not require any parental guidance.
- It can be observed that the variables Tempo,Energy and Danceability are having a normal distribution which means that most of the values are around the mean.
- The songs are composed in all the key ranges.. Song keys of 0 and 7 (C and G) are the most used and D# key is the least used.
- The majority of the songs are quite loud in nature since the average loudness score of the songs is from -20 to 0.
- Number of songs composed in major mode is more in number than with the minor mode.
- The presence of spoken words is quite less in most of the songs and the speechiness attribute denotes it.
- Most of the songs were recorded in a studio and not recorded with a live audience or on stage.
- The common time signature in all the songs is 4/4, which means 4 beats in one bar.
- Most of the songs have a mix of vocal content and instruments and the instrumentalness attribute denotes it.
- The outliers are present in some of the attributes in the dataset, the extreme outliers are removed by using the IQR criterion.

Correlation between the variables:

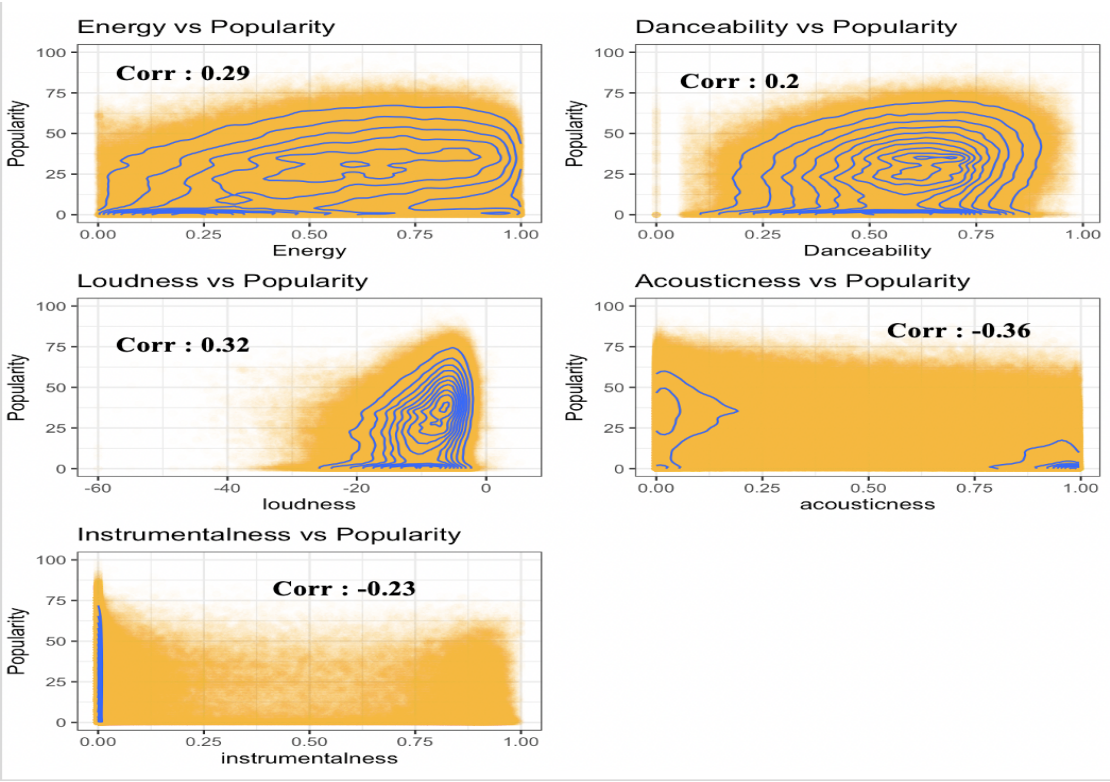


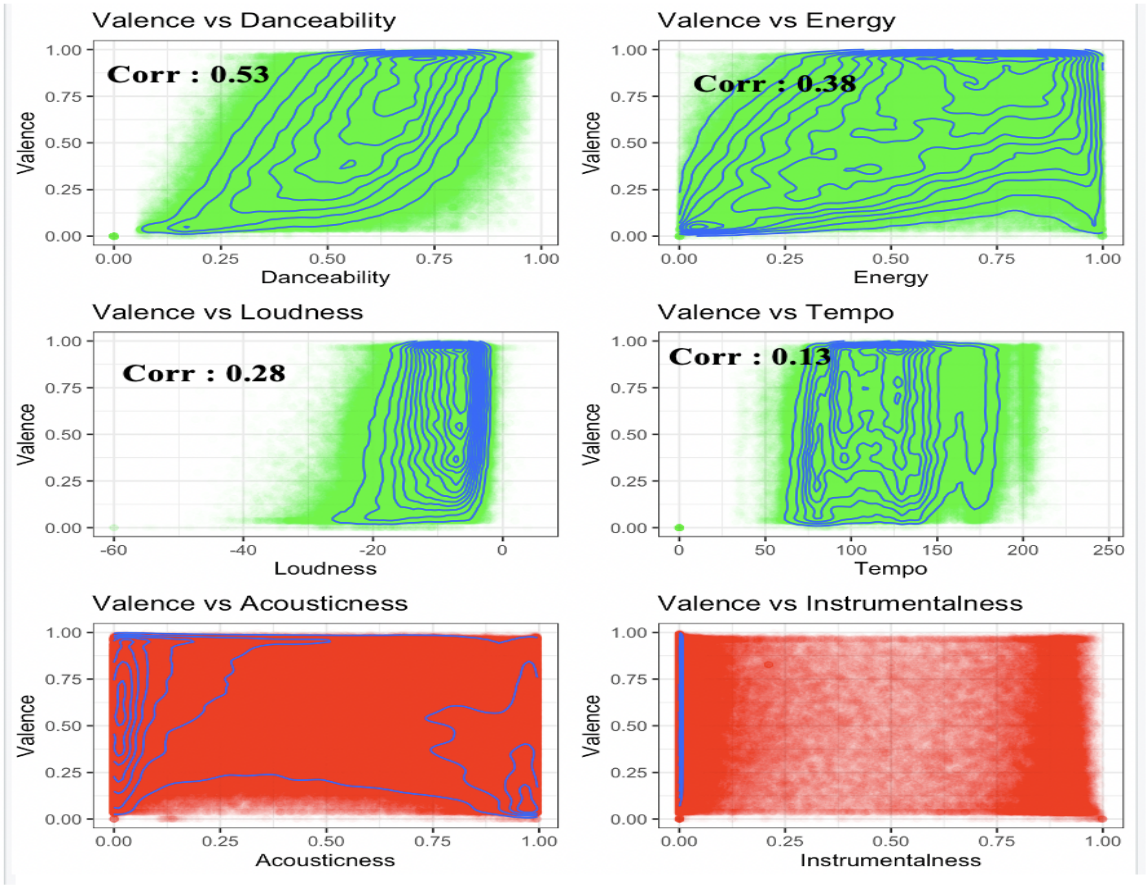
- From this correlation matrix, it can be observed that the song popularity and valence are having a positive correlation with song attributes like energy,loudness, danceability and a negative correlation with acousticness and instrumentalness.
- The attributes energy and loudness are highly positively correlated. As the energy of the song increases, the loudness also increases.

Correlation of the important song features:



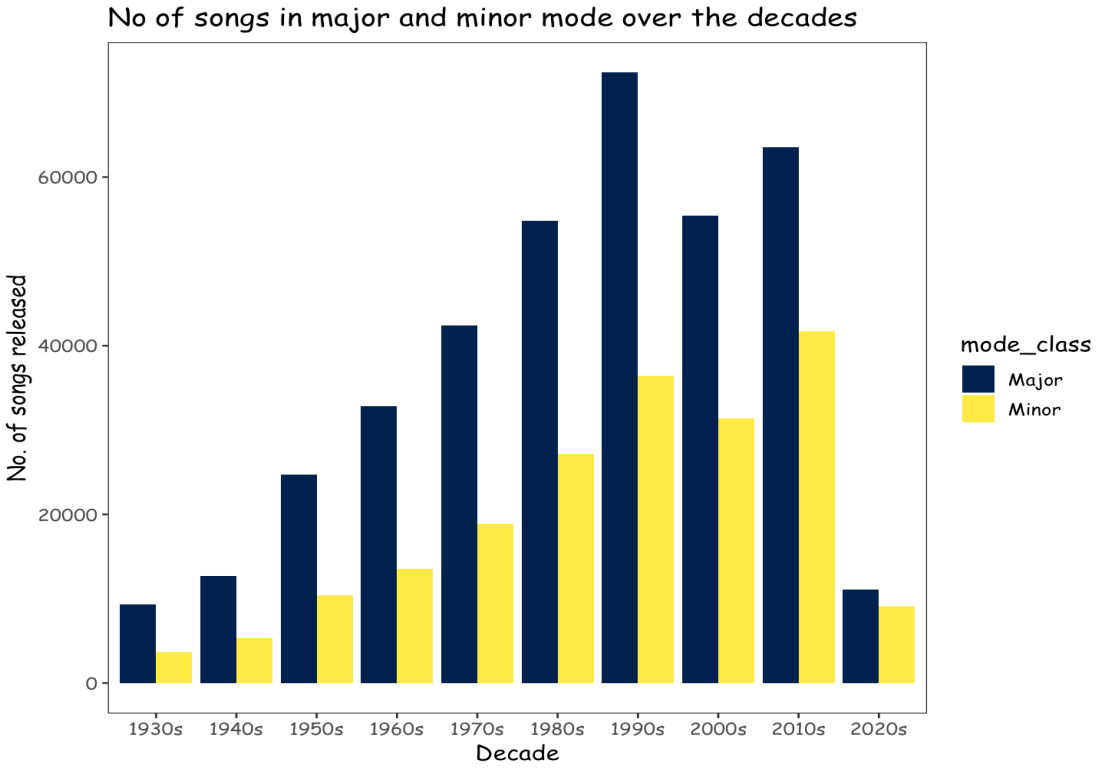
2-D Density plot of the song attributes valence and popularity with other related attributes.





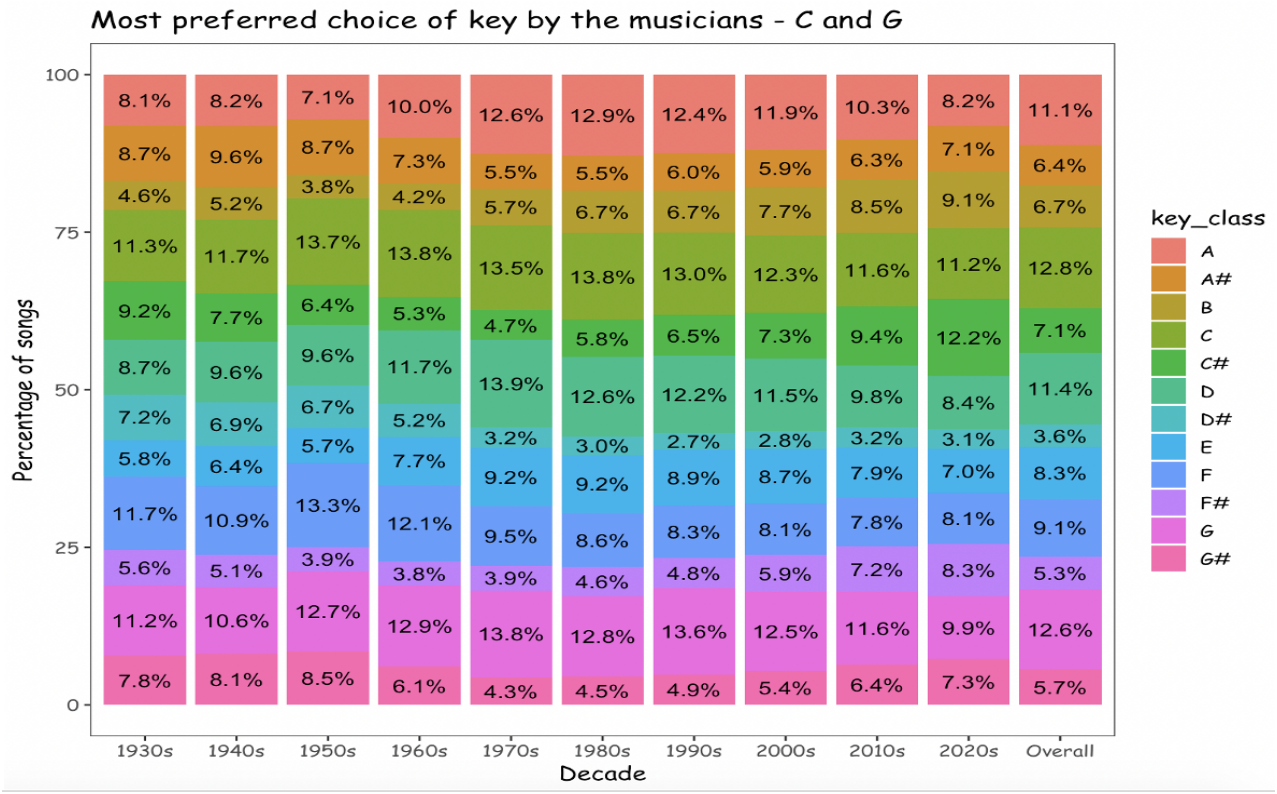
Data Analysis

1. What is the popular choice for the mode of the songs by the musicians in the decade 1930s - 2020s?



This grouped bar plot gives information about the number of songs composed in major and minor mode over the decades 1930s-2020s. It can be observed that the song composers highly preferred the major mode for composing the songs than the minor mode in the decades 1930s - 2020s.

2. What is the percentage of the songs in each key class released in the decades 1930s - 2020s?



This graph gives information about the percentage of songs composed in different keys in the decades 1930s - 2020s. It also provides the overall result of it. It can be observed that the keys C and G were used to compose nearly 25% of the overall songs and were highly preferred by the musicians in the 1930s - 2020s.

3. What are the top 5 popular songs in each decade with C and G keys and major mode?

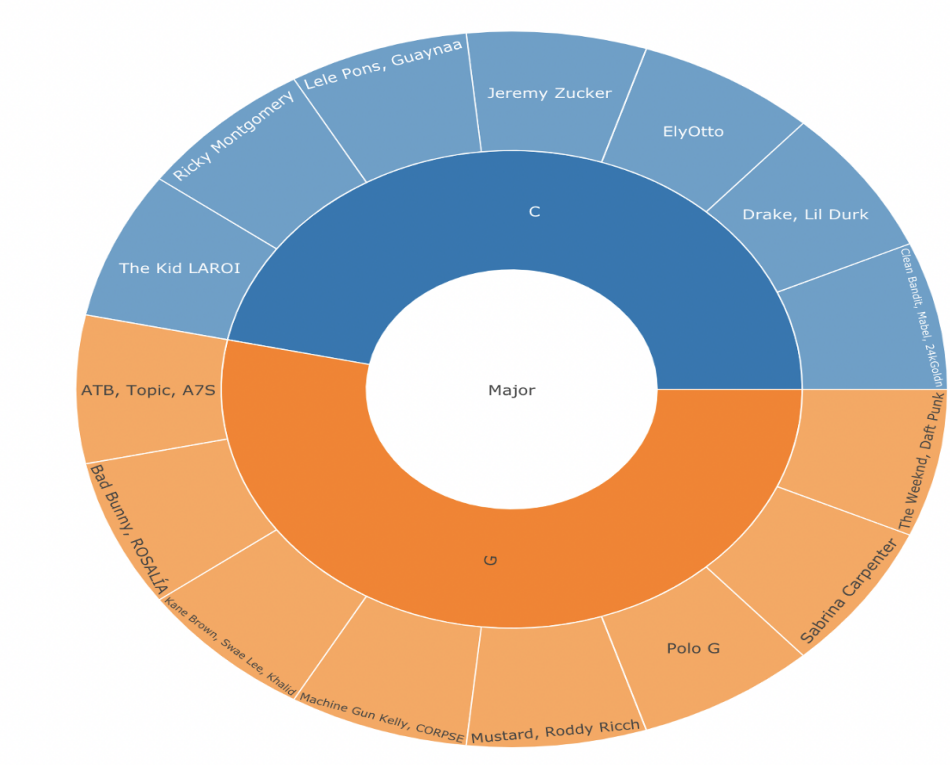
Top 5 popular songs in each decade, in C major key

Top 5 popular songs in G major key in the decade 1930s-2020s:

The above treemaps are used to represent the top 5 popular songs in C major and G major keys in the decades 1930s-2020s.

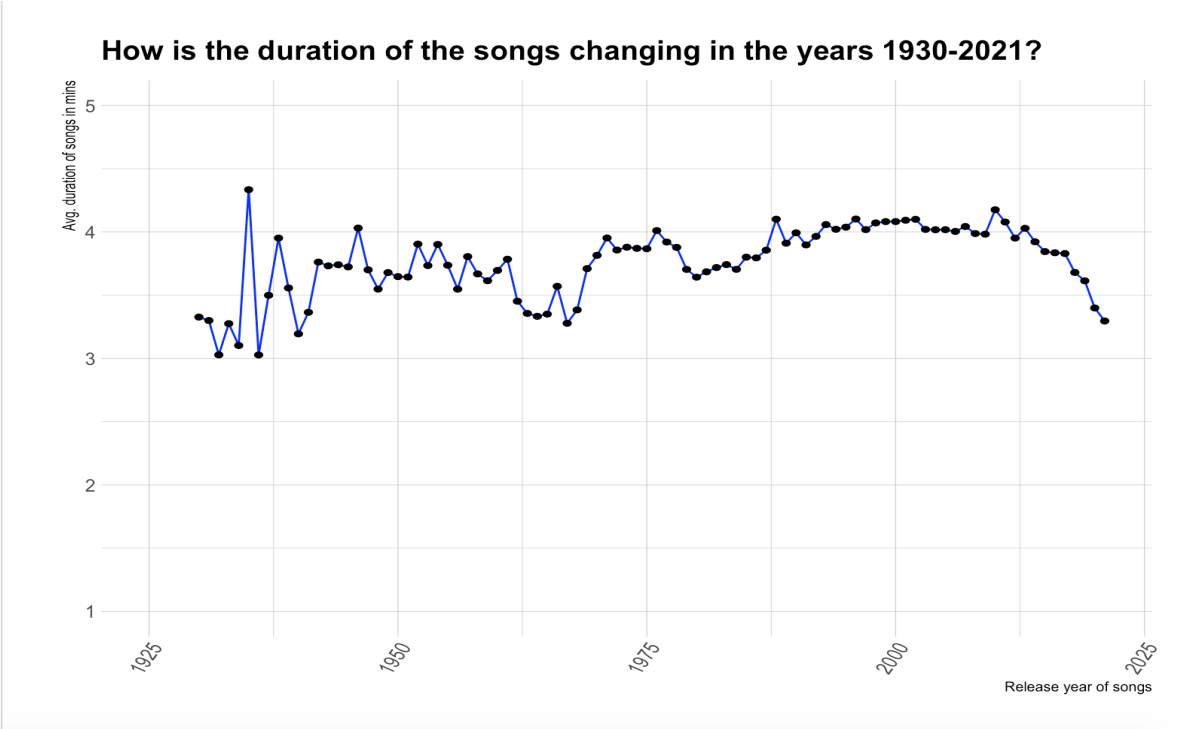
4. Who are the top 15 popular artists involved in the songs composed in C major and G major scale?

Top 15 popular artists of songs in C major and G major scale:



The above sunburst chart shows the top 15 popular artists of songs which were composed in C major and G major keys.

5. What is the average duration or the length of the songs in the decades 1930s-2020s?



From the above line graph we can see that the average duration of the songs was around 3.5 mins in the years 1930-1970. From the year 1970 - 2000, it was approaching the 4 min mark and we can see a downward trend in the average duration of the songs in the last 20 years and the number is approaching below 3.5 mins.

6. Is there a relation between song popularity and the duration of the songs?

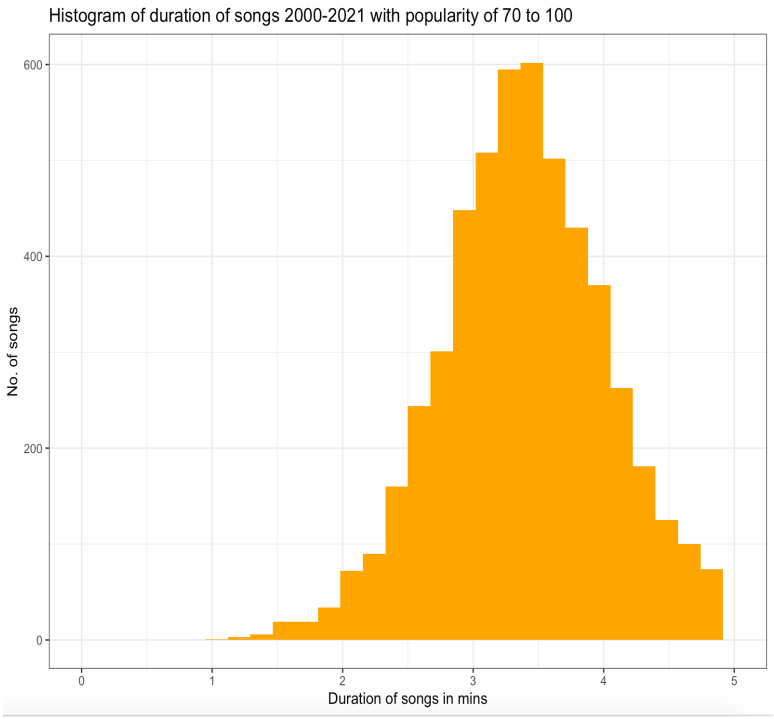
   For this study, we can consider the songs released in the years 2000-2021 and let us divide the songs into two classes:

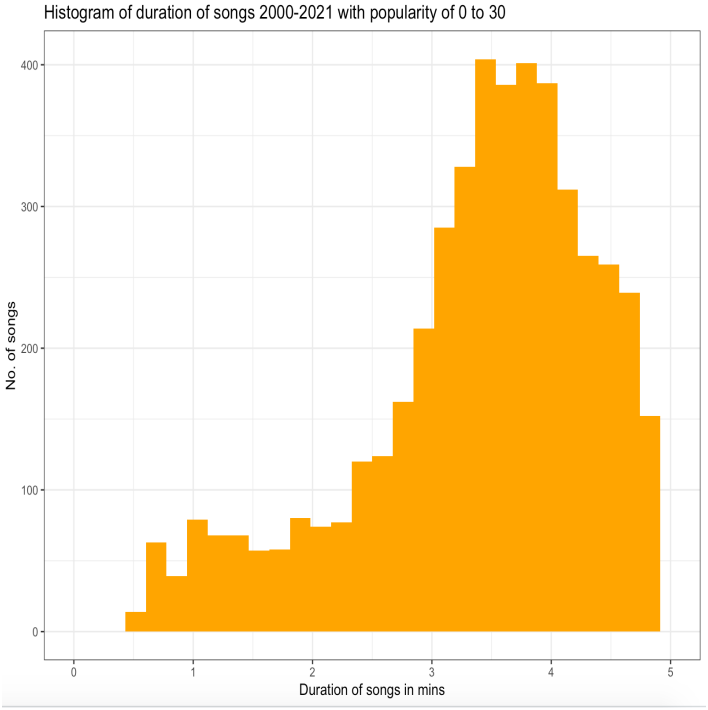
   - Most popular songs (Song popularity 70 - 100)
   - Least popular songs (Song popularity 0 - 30)

No of songs in the least popular song category is more in number than the most popular songs category. In order to keep the comparison fair, let us do undersampling for the second category of songs.

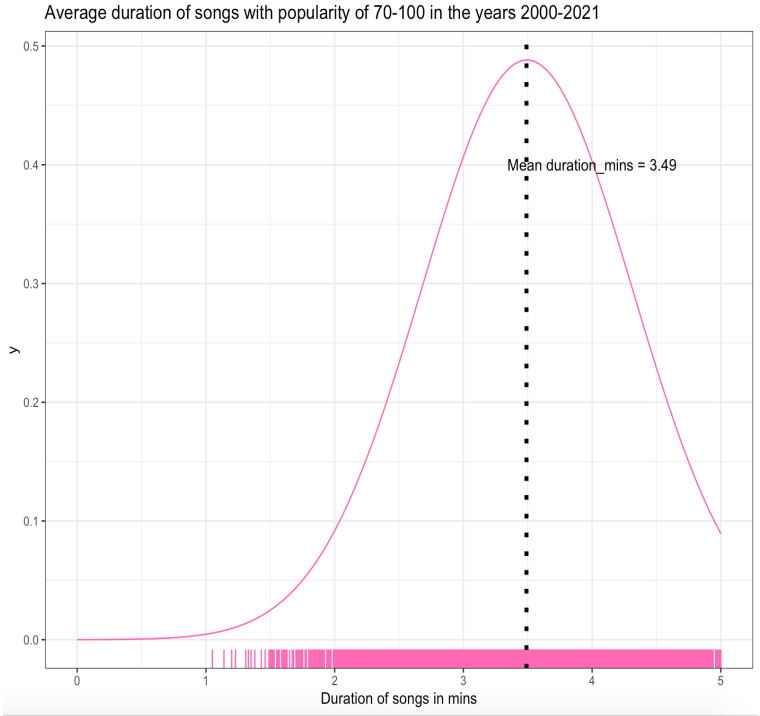
Resultant number of songs in both the classes after undersampling:

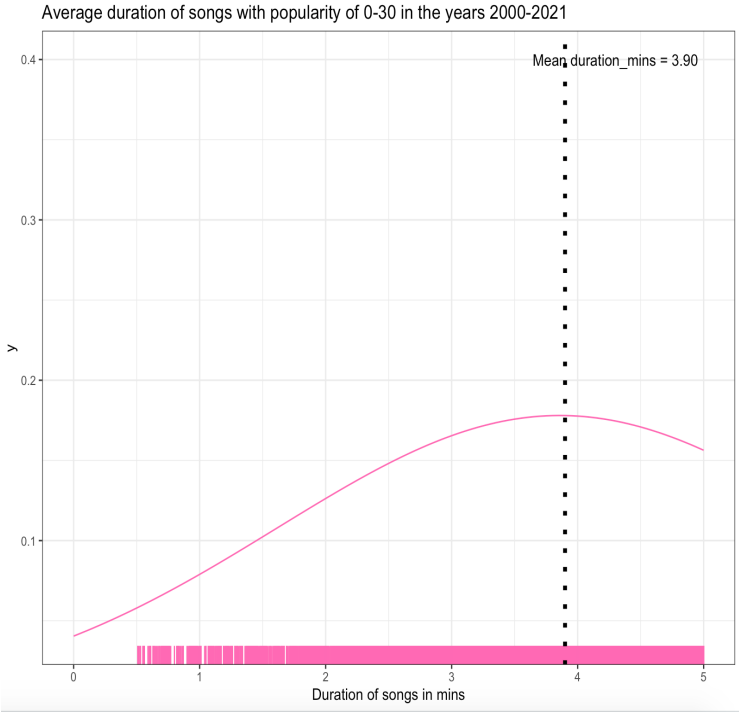
- No of songs in the most popular songs class (Song popularity 70 - 100) - 5402
- No of songs in the least popular songs class (Song popularity 0 - 30) - 5402





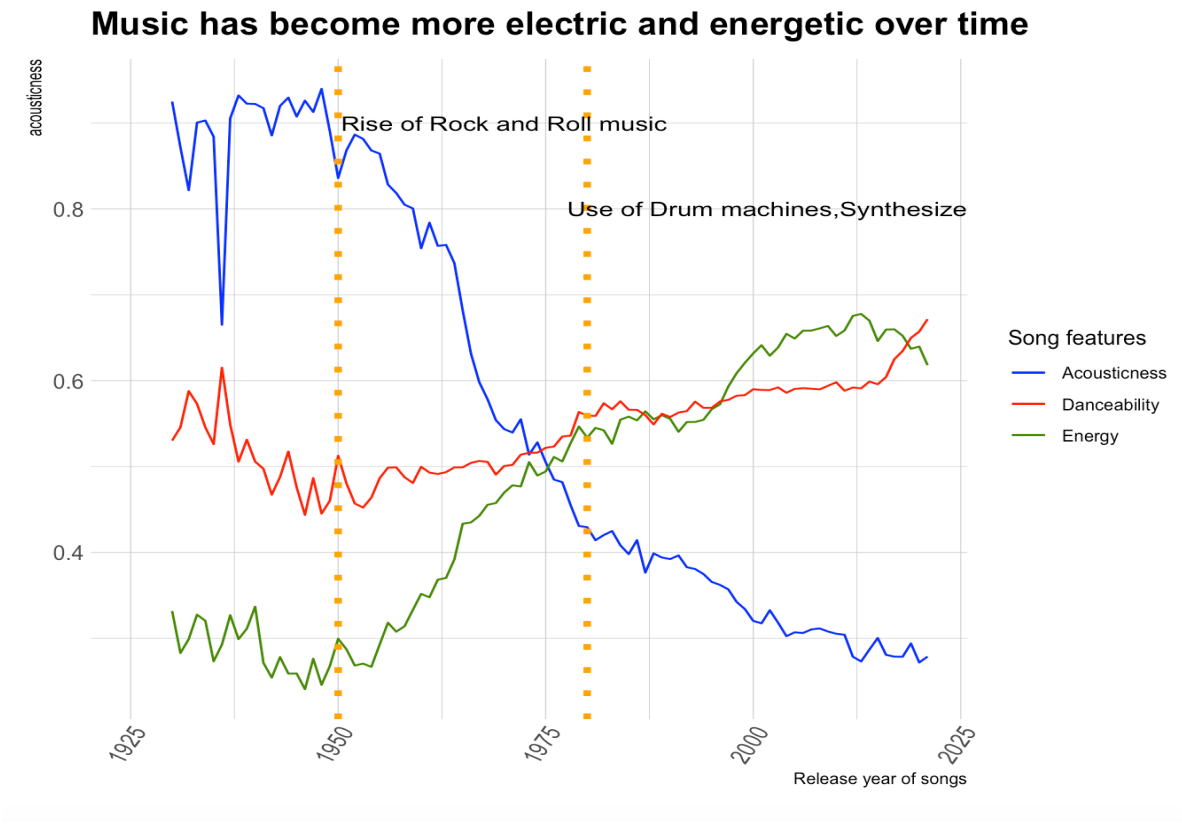
The above histogram shows the no of songs with duration in both the classes.





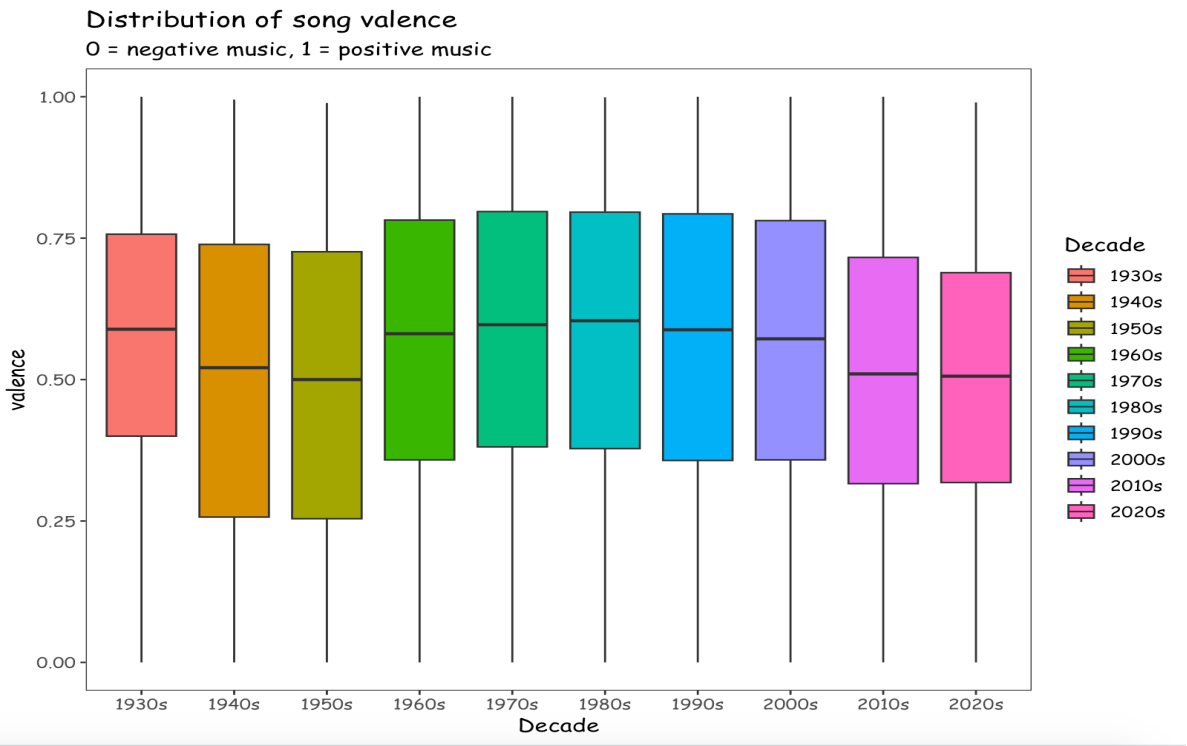
From the above graphs, it can be observed that as per the trend in the recent 20 years, most popular songs are shorter in duration than the least popular songs. This highlights a fact that the attention span of the average music consumer is falling and it suggests that shorter songs have a higher chance of gaining more attention and can reach the majority of the audience. The artists and musicians can make use of this fact to produce more songs with shorter duration to maximize the revenue and outreach with the audience.

7. What is the trend of Energy, Danceability and acousticness in the songs released in the years 1930-2021?



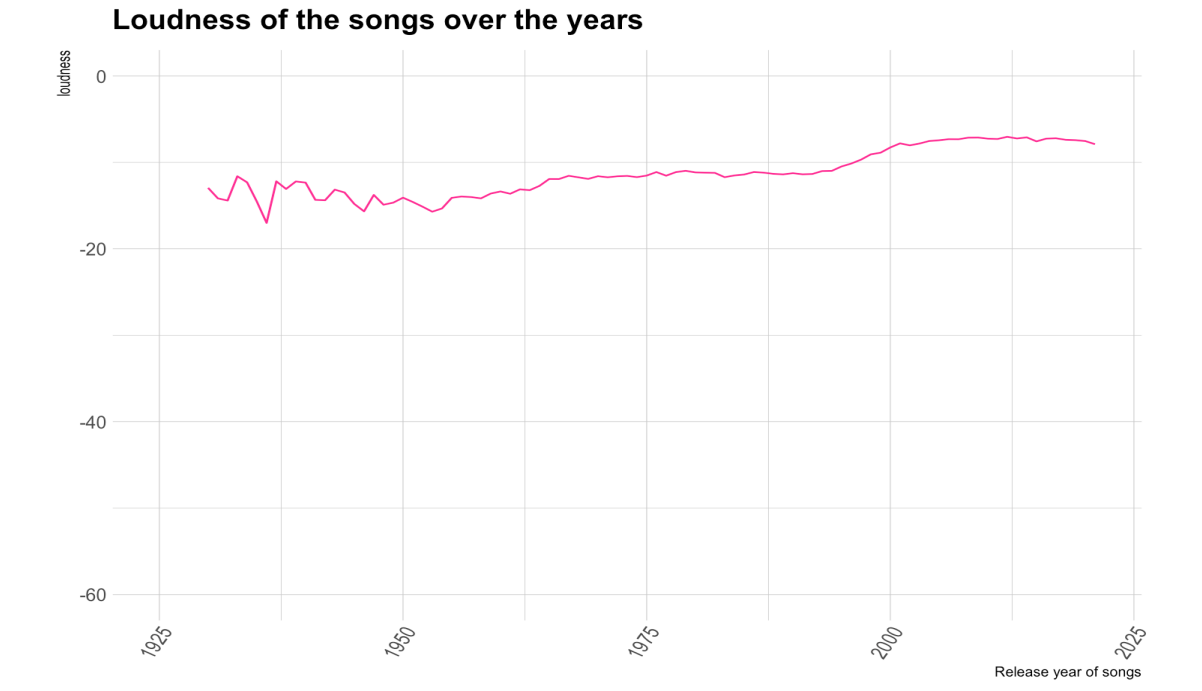
In the above line graph, the trends of three song features are depicted namely energy, danceability and acousticness over the years 1930-2021. It can be observed that the acousticness feature is gradually decreasing since 1950 and one of the major reasons for this decline is the heavy use of electronic instruments like Guitars, synthesizers,drum machines and use of sophisticated tools and technologies in the music industry. In the 1950s, songs in the Rock and Roll genre were rising with popularity and rock and roll songs were generally high in energy and there was no scope of using acoustic instruments in that genre. As many electronic instruments like Bass synth and Guitars were used, the songs were also produced with more energy and more danceable and foot tapping tunes were produced. Hence the energy and danceability of the songs increased over the years as the acousticness of the songs decreased.

8. How has the song positivity changed over the years 1930 - 2021 ?



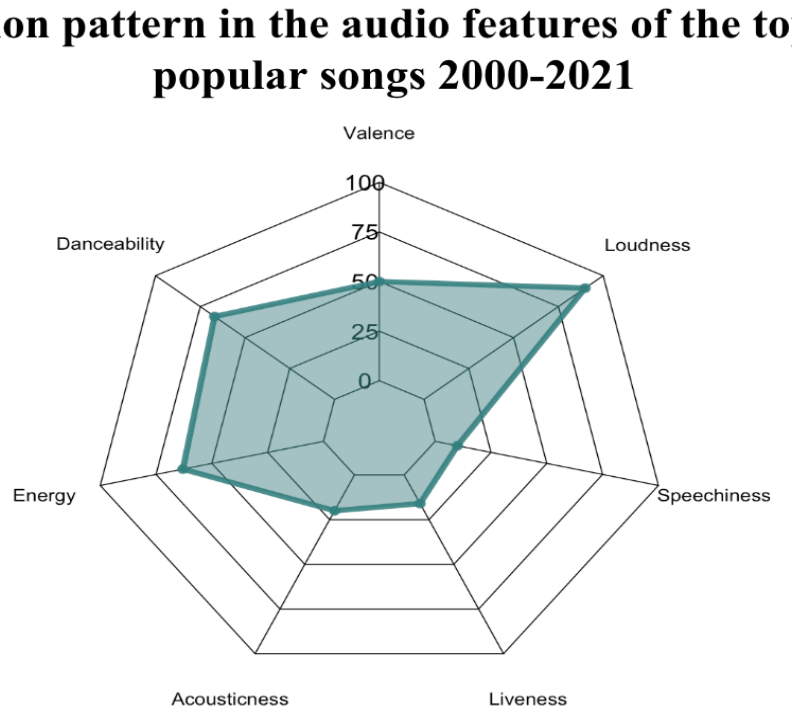
From the above boxplot of valence scores, it can be observed that the song positivity is reducing from the year 1980 to 2020s decade and it is approaching below 0.5. It can be understood that the music produced these days generally are not sounding very happy and a neutral mood is maintained in the songs.

9. What is the pattern of the loudness feature of the songs?



The loudness of the songs in this Spotify dataset is measured in the unit of dbFS (Decibels relative to Full scale) in the range of -60 to 0. -60 being the lowest possible sound value and 0 being the loudest value. The loudness of the songs have been increasing every year and was constantly in the range of -20 and 0. In recent times since the year 2000, the songs are produced with high loudness scores of -6 and -5. The song composers and artists of the songs can try to reduce the loudness in the songs as it can make the song sound very jarring and not very enjoyable.

10. What is the common pattern observed in the audio features of the most popular 1000 songs released in the recent years 2000-2021?



For this Spider plot, the above song features are scaled in the range of 0 - 100 in order to maintain uniformity for representing these values in a single graph.

It can be observed that the top 1000 popular songs released in the years 2000-2021 are having high energy,danceability and loudness values.

These songs are having a less acoustic value which means that these songs are mostly recorded using electronic instruments than the acoustic instruments.

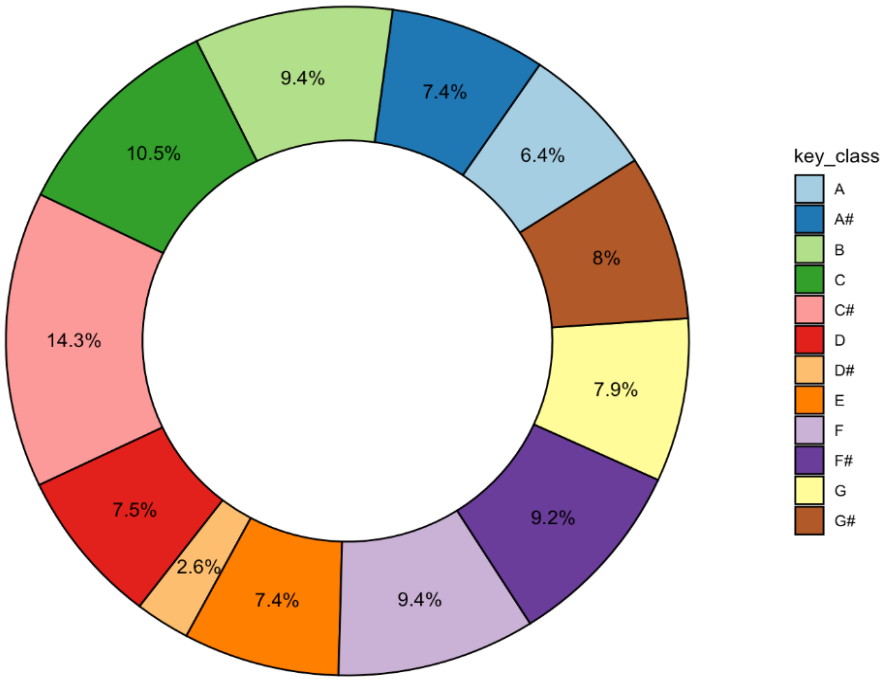
The speechiness value in these songs is very less which means that the presence of spoken words is very less in these songs.

The liveness attribute is quite less indicating that these songs were recorded in a studio and were not recorded in stage or with live audience.

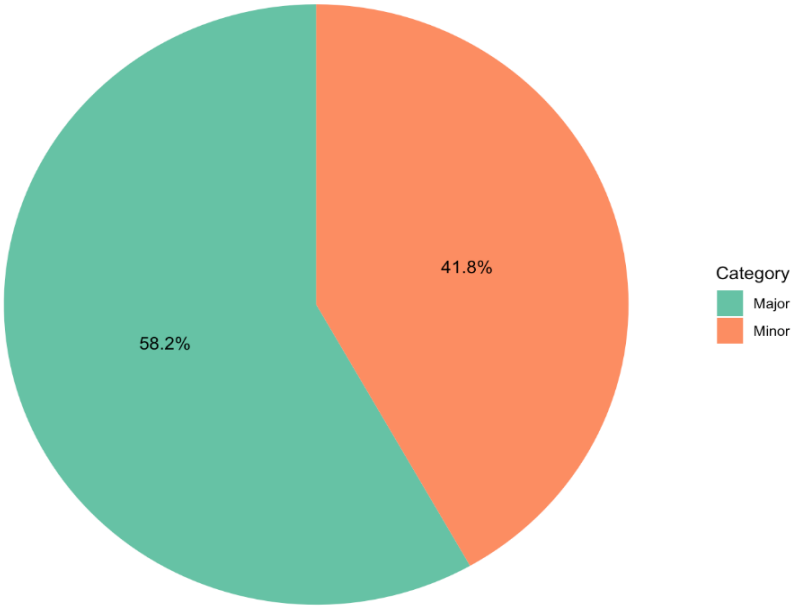
The valence score is neutral meaning that these top popular songs are not sounding too happy or too sad.

11. What is the percentage of keys and modes observed in the top 1000 popular songs 2000-2021?

Percentage of the top 1000 popular songs (2000-2021) in each key



Percentage of the top 1000 popular songs (2000-2021) in song modes



From the above donut chart, we can see that the C# key was the most used key in the top 1000 popular songs, followed by C key and the D# key was the least used. Around 58% of songs were composed in the major mode.

Hypothesis Testing

One sample and two sample t-tests are to be conducted to validate claims and hypothesis statements.

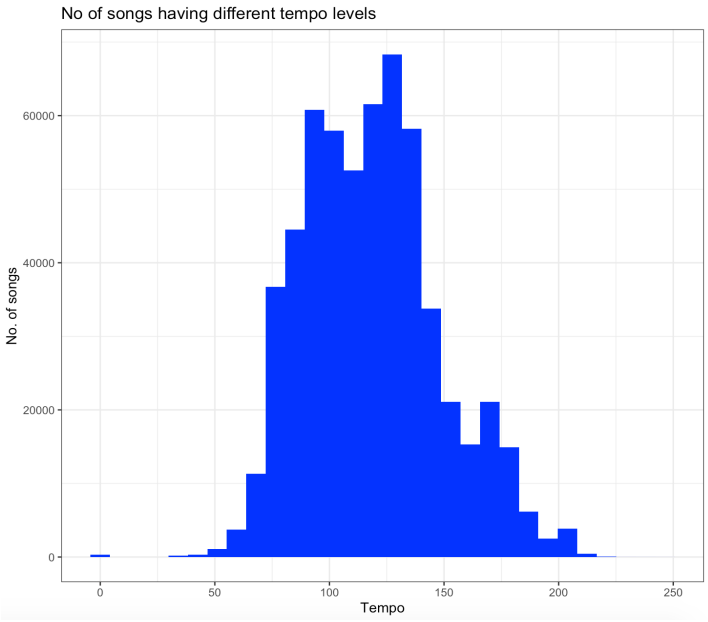
1. One sample T-test:

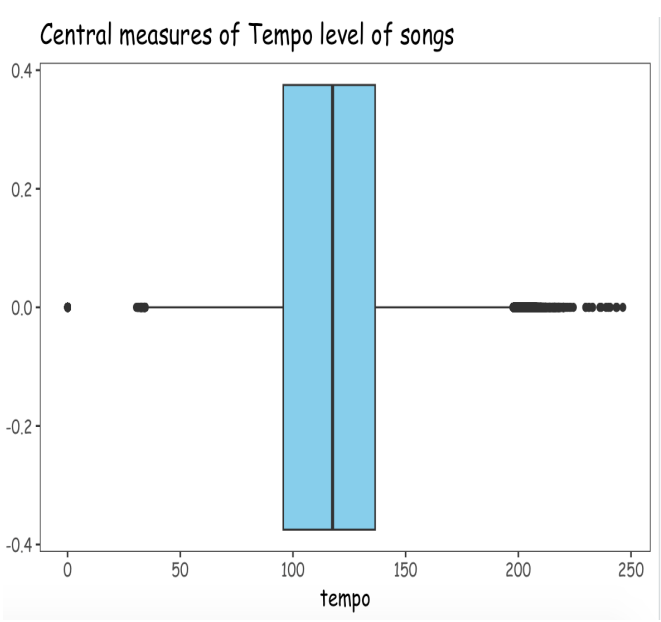
Claim to be tested:

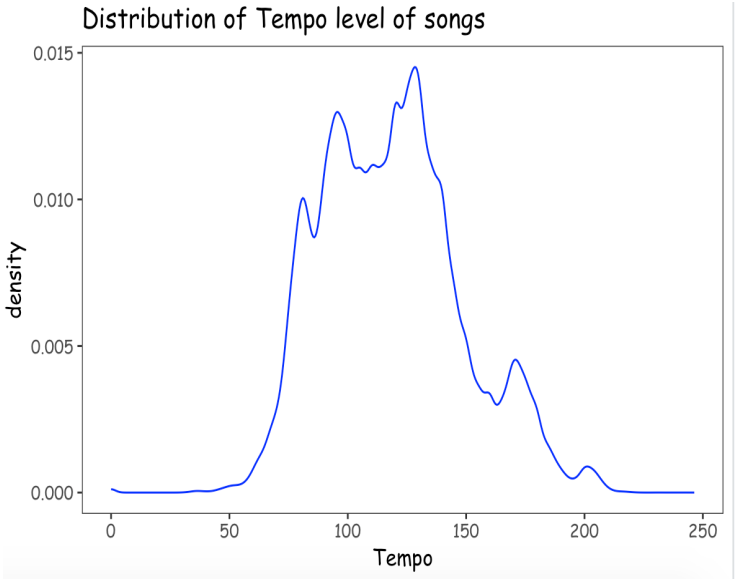
The mean tempo level of the songs is equal to 120 Beats Per Minute.

Data Visualization:

- Visualization of the tempo level of the song in the form of histogram,boxplot and density plot.







```
> describe(spotify_dataset$tempo)
   vars      n   mean    sd median trimmed   mad min    max  range skew kurtosis   se
X1    1 576906 118.54 29.72 117.57  116.92 30.48   0 246.38 246.38  0.4    -0.07 0.04
>
```

From the graphs, it can be observed that most of the songs have an average tempo level of 120 BPM. This claim can be checked using one sample t-test. From the summary statistics of the tempo level, the below points can be noted:

- The tempo attribute is having almost equal mean and median values.
- The skewness of the tempo variable is 0.4 and the distribution appears to be fairly symmetrical since the skewness values lie in the range of -0.5 and 0.5.
- We can assume the tempo variable to be normally distributed from the above graphs and summary table.

Null Hypothesis:

- The mean tempo level of the songs is equal to 120 Beat Per Minute.

Alternate Hypothesis:

- The mean tempo level of the songs is not equal to 120 Beat Per Minute.

- One sample t-test can be conducted using t.test() function in R

```
> t.test(spotify_dataset$tempo, mu = 120)

        One Sample t-test

data:  spotify_dataset$tempo
t = -37.293, df = 576905, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 120
95 percent confidence interval:
 118.4640 118.6174
sample estimates:
mean of x
 118.5407
```

The below points can be interpreted from the above values:

- This is nothing but a one sample two-tailed test since the mean true value can lie on any side of the hypothesized mean if the null hypothesis is to be rejected.
- The t-value is very low, there are high chances of rejecting the null hypothesis.
- The mean value of the tempo is 118.54
- The p-value is the probability or the chances in favor of the null hypothesis. P-value is very low with a value less than 2.2 e-16.
- The confident interval at 95% is in the range of 118.4640 and 118.6174.

Conclusion:

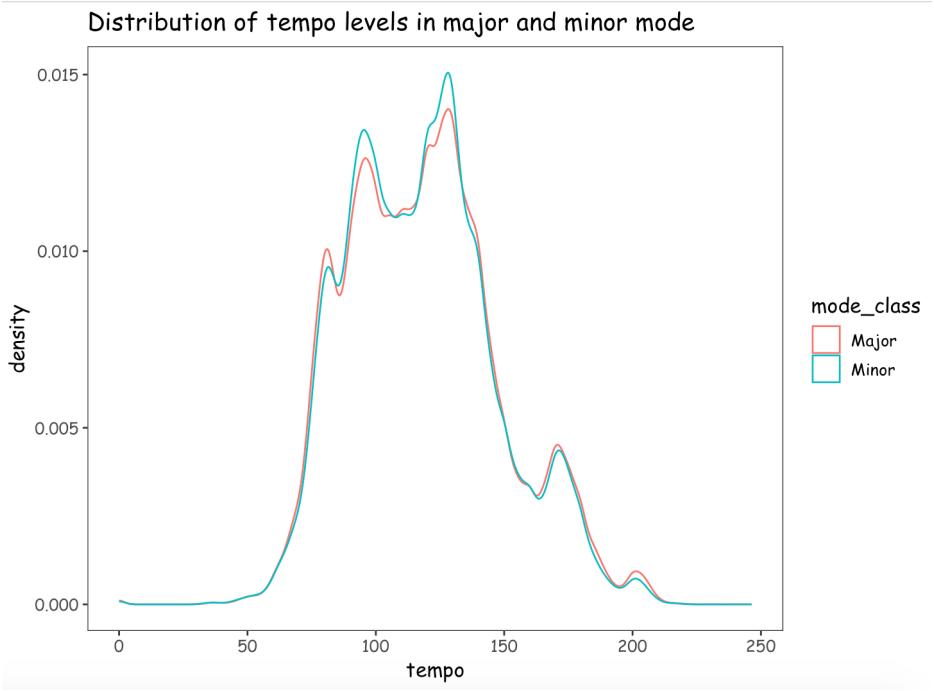
- The p-value is 2.2 e^-16 and it is very less than the significance level 0.05, hence we can reject the null hypothesis and conclude that the average tempo level of the songs is not equal to 120 Beats Per Minute and can be greater or less than this value.

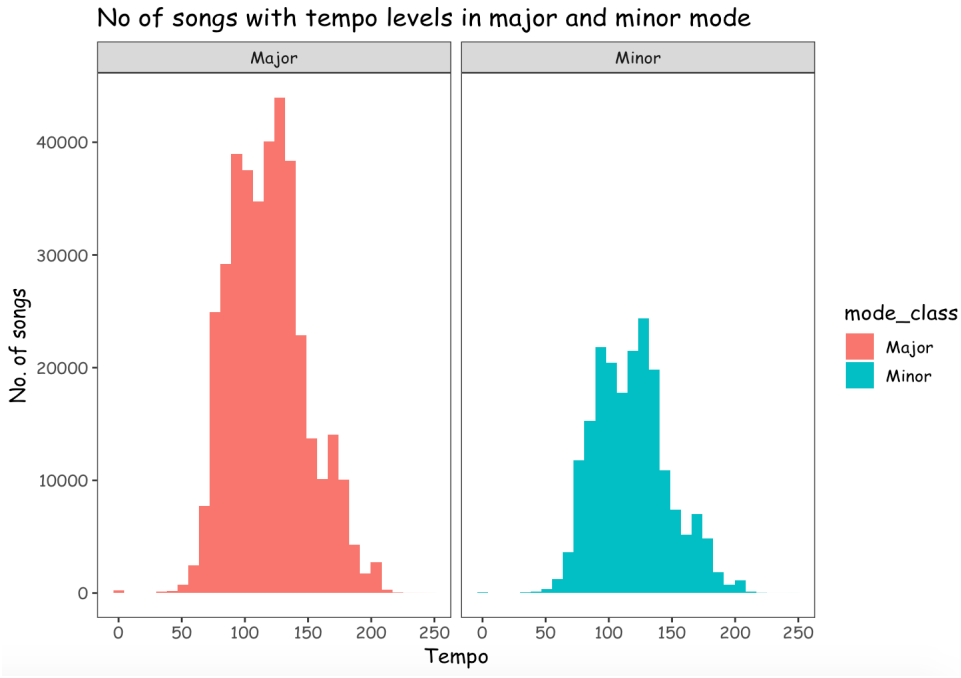
2. Two sample independent T-test:

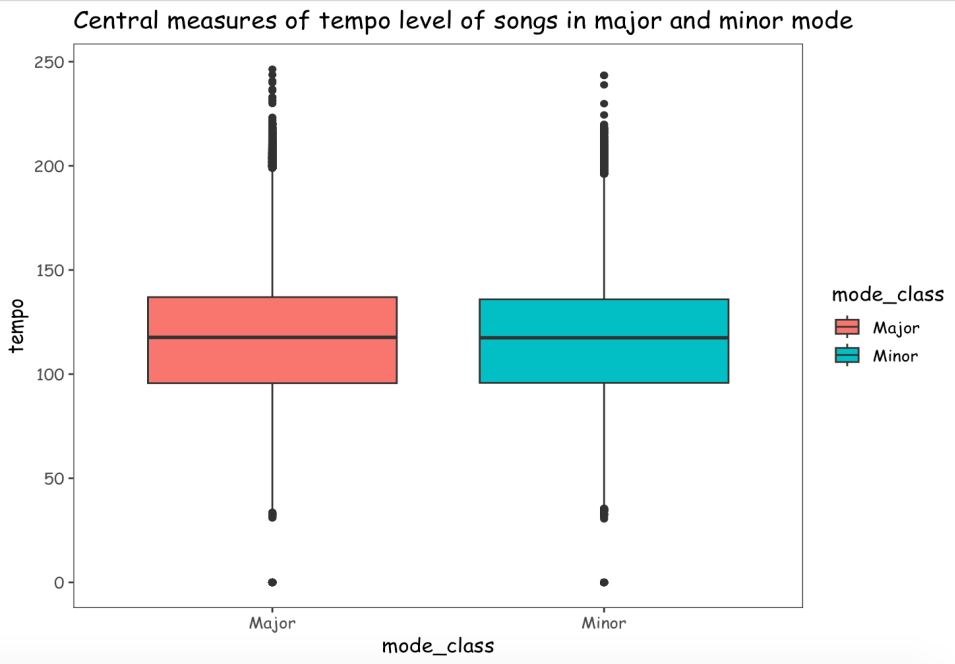
Claim to be tested:

The tempo level of the songs is different in songs in major mode and minor mode.

Data Visualization:








```
> describe(major_tempo_data)
   vars      n   mean    sd median trimmed   mad min    max  range skew kurtosis   se
X1    1 379313 118.71 30.02 117.64  117.07 30.72   0 246.38 246.38  0.4    -0.09 0.05
> describe(minor_tempo_data)
   vars      n   mean    sd median trimmed   mad min    max  range skew kurtosis   se
X1    1 197593 118.22 29.14 117.44  116.65 30.26   0 243.51 243.51  0.4    -0.06 0.07
```

```
> var(major_tempo_data)
[1] 901.1196
> var(minor_tempo_data)
[1] 849.2004
```

- Many songs were composed in major mode when compared with the minor mode. This shows that the music composers are more interested towards making songs in major mode.
- The groups in comparison are not required to have an equal number of samples for carrying out a T-test and therefore this hypothesis testing can be performed.
- The tempo attribute's distribution is quite similar in these modes and able to observe two peaks at 90-100 and 130-150 BPM.
- The average value of the tempo levels is around 120 BPM for both the modes.
- The variance of both the groups are unequal.
- The mean and median values are almost equal and the skewness value id 0.4 and hence tempo variable can be assumed to have a normal distribution.

Null Hypothesis:

- The average tempo levels in major and minor modes are equal.

Alternate Hypothesis:

- The average tempo levels in major and minor modes are not equal.

- One sample t-test can be conducted using t.test() function in R.

```
> t.test(major_tempo_data, minor_tempo_data, alt = "two.sided",
+        ,conf=0.95,
+        var.eq=F, paired=F)

        Welch Two Sample t-test

data:  major_tempo_data and minor_tempo_data
t = 5.9209, df = 410996, p-value = 3.205e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.3235675 0.6437905
sample estimates:
mean of x mean of y
 118.7064  118.2227
```

The below points can be interpreted from the above values:

- This is nothing but a two sample two-tailed test since the mean difference value can lie on any side of the hypothesized mean difference if the null hypothesis is to be rejected.
- The mean difference between major and minor mode is 0.48.
- The p-value is the probability or the chances in favor of the null hypothesis. P-value is very low with a value less than 3.205e^-09.
- The confident interval at 95% is in the range of 0.32 and 0.64.

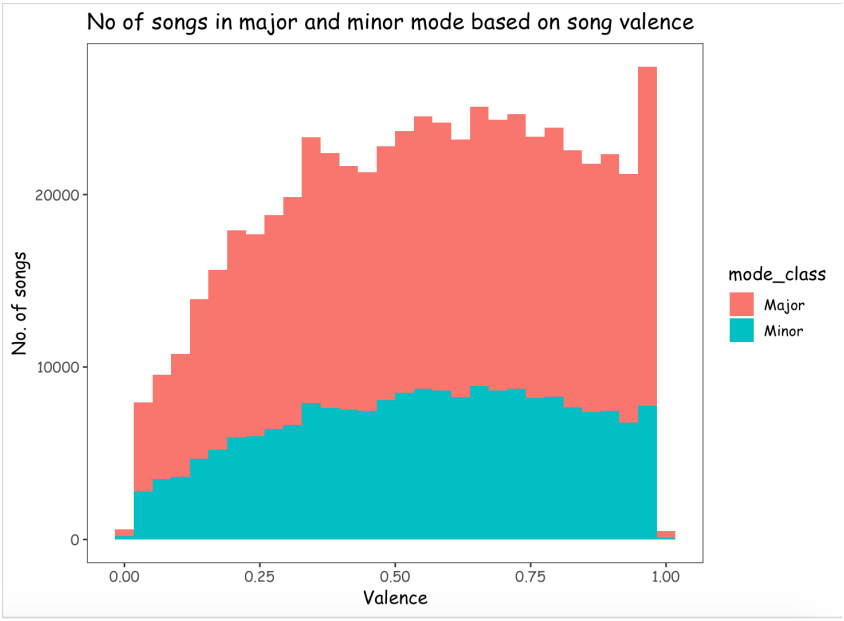
Conclusion:

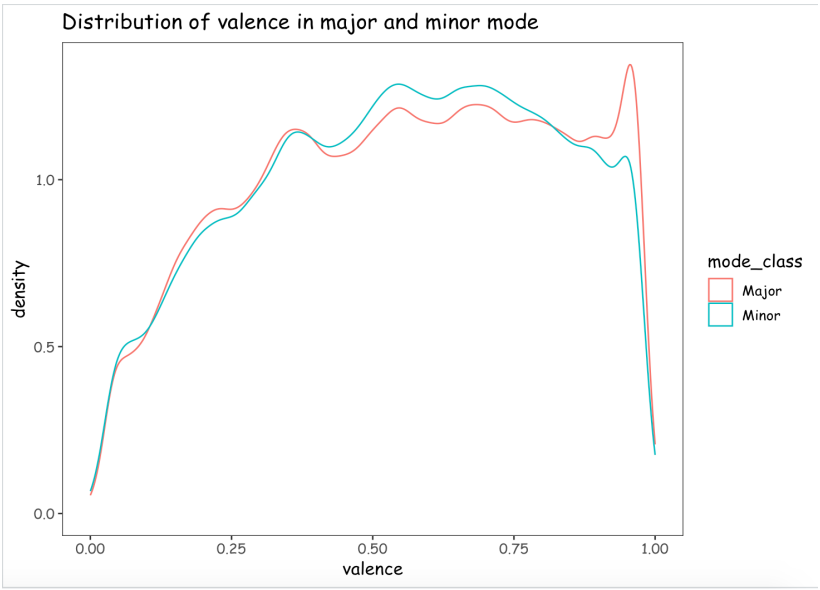
- The p-value is 3.20e^-09 and it is very less than the significance level 0.05, hence we can reject the null hypothesis and conclude that the average tempo level of the songs is different in major and minor mode.
- The tempo of the songs in major mode are slightly greater than the tempo of the songs in minor mode.
- A general fact is that the songs in the major mode sound much happier,merrier and hence there is a slight increase in the tempo level.
- Songs in minor mode often sound sad,gloomy and hence a lesser tempo is found.

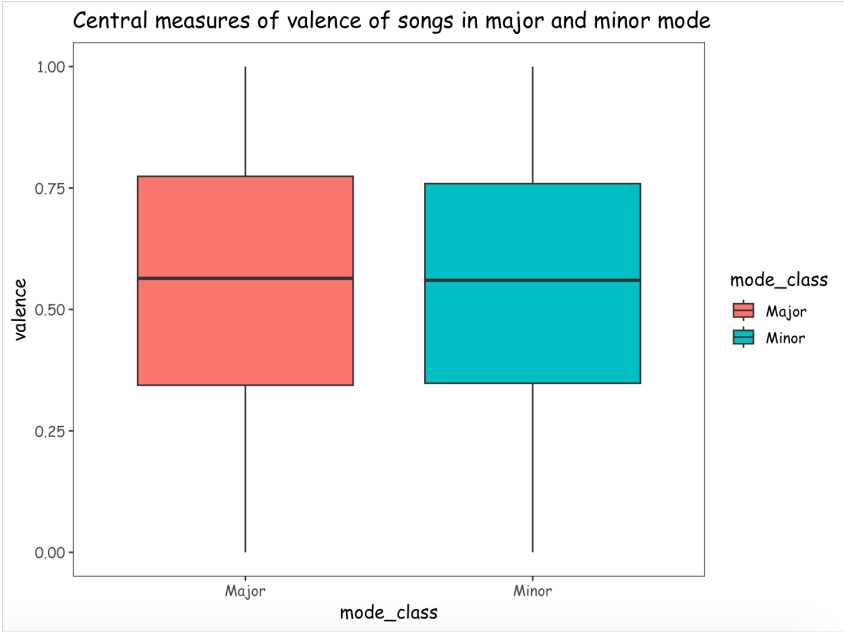
3. **Two sample independent T-test:**

Claim to be tested:

The songs composed in major mode sound much happier than the songs composed in minor mode.


No of songs in major and minor mode based on song valence
No. of songs
20000
10000
0
0.00
0.25
0.50
0.75
1.00
Valence
mode_class
Major
Minor


Distribution of valence in major and minor mode
density
1.0
0.5
0.0
0.00
0.25
0.50
0.75
1.00
valence
mode_class
Major
Minor


Central measures of valence of songs in major and minor mode
valence
1.00
0.75
0.50
0.25
0.00
Major
Minor
mode_class
mode_class
Major
Minor

```
> describe(major_valence_data)
   vars      n mean   sd median trimmed  mad min max range  skew kurtosis se
X1    1 379313 0.55 0.26   0.56    0.56 0.32   0   1     1 -0.14    -1.06  0
> describe(minor_valence_data)
   vars      n mean   sd median trimmed  mad min max range  skew kurtosis se
X1    1 197593 0.55 0.25   0.56    0.55 0.31   0   1     1 -0.16       -1  0
```

```
> var(major_valence_data)
[1] 0.06753145
> var(minor_valence_data)
[1] 0.06459943
```

- Many songs were composed in major mode when compared with the minor mode. This shows that the music composers are more interested towards making songs in major mode.
- The groups in comparison are not required to have an equal number of samples for carrying out a T-test and therefore this hypothesis testing can be performed.
- The valence attribute's distribution is quite similar in these modes and no of songs in major mode with a valence of 1 is greater than that of minor mode songs. No of songs in minor mode with a mean valence of 0.5 to 0.75 is greater than that of major mode songs.
- The average valence value is 0.55 for both the modes.
- The variance of the valence is quite similar in both the modes.
- The mean and median are almost similar and we can assume the data in both groups to be normally distributed.

Null Hypothesis:

- The average valence in major mode songs is greater than the average valence score in minor mode songs.

Alternate Hypothesis:

- The average valence in major mode songs is less than the average valence score in minor mode songs.

- One sample t-test can be conducted using t.test() function in R.

```
> t.test(major_valence_data, minor_valence_data, alt = "less",
+        ,conf=0.95,
+        var.eq=F, paired=F)

        Welch Two Sample t-test

data:  major_valence_data and minor_valence_data
t = 7.4952, df = 408316, p-value = 1
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
        -Inf 0.006495056
sample estimates:
mean of x mean of y
0.5536241 0.5482979
```

The below points can be interpreted from the above values:

- This is an example of a two sample one-tailed test.
- The mean difference between major and minor mode is 0.01.
- The p-value is the probability or the chances in favor of the null hypothesis. P-value is high with a value of 1.
- The confident interval at 95% is in the range of -Inf and 0.006.

Conclusion:

- The p-value of this test is 1 and it is significantly greater than the significance level 0.05, hence we fail to reject the null hypothesis and conclude that the average valence score in major mode songs is greater than the average valence score in minor mode songs.
- Songs composed in major mode are sounding happier than the songs in minor mode.

Linear regression

The song valence (song positivity) is the target variable to be predicted based on various song features.

Initial steps:

- The outlier records were eliminated from the dataset.
- The input attributes were scaled in the range of 0 and 1.

Selection of the best features to predict the song valence:

- The regsubsets() function of the leaps package can be used to select the best variables which can better predict the target variable. It will select the variables with size from 1 to number mentioned in the nvmax variable in the regsubsets() function.

```
Subset selection object
Call: regsubsets.formula(valence ~ ., data = spotify_numeric_df, nvmax = 5)
11 Variables  (and intercept)
                 Forced in Forced out
popularity           FALSE      FALSE
explicit             FALSE      FALSE
danceability         FALSE      FALSE
energy               FALSE      FALSE
loudness             FALSE      FALSE
speechiness          FALSE      FALSE
acousticness         FALSE      FALSE
instrumentalness     FALSE      FALSE
liveness             FALSE      FALSE
key                  FALSE      FALSE
mode                 FALSE      FALSE
1 subsets of each size up to 5
Selection Algorithm: exhaustive
         popularity explicit danceability energy loudness speechiness acousticness instrumentalness liveness key mode
1  ( 1 ) " "        " "      "*"          " "    " "      " "         " "          " "              " "      " " " "
2  ( 1 ) " "        " "      "*"          "*"    " "      " "         " "          " "              " "      " " " "
3  ( 1 ) "*"        " "      "*"          "*"    " "      " "         " "          " "              " "      " " " "
4  ( 1 ) "*"        " "      "*"          "*"    " "      " "         "*"          " "              " "      " " " "
5  ( 1 ) "*"        "*"      "*"          "*"    " "      " "         "*"          " "              " "      " " " "
```

1. **Selecting the danceability feature to predict the song valence.**

   Independent variables: danceability
   Dependent variable: valence

```
Call:
lm(formula = valence ~ danceability, data = spotify_numeric_df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.83897 -0.16226  0.00292  0.16836  0.74525

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.090329   0.001020   88.53   <2e-16 ***
danceability 0.819116   0.001737  471.51   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2191 on 576904 degrees of freedom
Multiple R-squared:  0.2782,    Adjusted R-squared:  0.2782
F-statistic: 2.223e+05 on 1 and 576904 DF,  p-value: < 2.2e-16
```

2. **Selecting the danceability and energy features to predict the song valence.**

   Independent variables: danceability,energy
   Dependent variable: valence

```
Call:
lm(formula = valence ~ danceability + energy, data = spotify_numeric_df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.83977 -0.14572  0.00348  0.15711  0.76493

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.173e-05  1.045e-03   0.011    0.991
danceability 7.185e-01  1.711e-03 419.959   <2e-16 ***
energy       2.694e-01  1.132e-03 238.018   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2091 on 576903 degrees of freedom
Multiple R-squared:  0.3427,    Adjusted R-squared:  0.3427
F-statistic: 1.504e+05 on 2 and 576903 DF,  p-value: < 2.2e-16
```

3. **Selecting the danceability ,energy,popularity features to predict the song valence.**

   Independent variables: danceability,energy,popularity
   Dependent variable: valence

```
Call:
lm(formula = valence ~ danceability + energy + popularity, data = spotify_numeric_df

Residuals:
     Min       1Q   Median       3Q      Max
-0.93578 -0.13994  0.00557  0.15114  0.78293

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.374e-02  1.033e-03   22.98   <2e-16 ***
danceability  7.547e-01  1.689e-03  446.77   <2e-16 ***
energy        3.160e-01  1.145e-03  276.01   <2e-16 ***
popularity   -2.493e-03  1.557e-05 -160.17   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2046 on 576902 degrees of freedom
Multiple R-squared:  0.3707,    Adjusted R-squared:  0.3707
F-statistic: 1.133e+05 on 3 and 576902 DF,  p-value: < 2.2e-16
```

4. Selecting the danceability ,energy,popularity, acousticness features to predict the song valence.

Independent variables: danceability,energy,popularity,acousticness
Dependent variable: valence

```
Call:
lm(formula = valence ~ danceability + energy + popularity + acousticness,
    data = spotify_numeric_df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.96950 -0.13780  0.00469  0.14760  0.88474

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1340035  0.0015924  -84.15   <2e-16 ***
danceability  0.7722132  0.0016709  462.15   <2e-16 ***
energy        0.4471019  0.0015194  294.26   <2e-16 ***
popularity   -0.0020643  0.0000157 -131.46   <2e-16 ***
acousticness  0.1449899  0.0011250  128.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2017 on 576901 degrees of freedom
Multiple R-squared:  0.3883,    Adjusted R-squared:  0.3883
F-statistic: 9.156e+04 on 4 and 576901 DF,  p-value: < 2.2e-16
```

5. Selecting the danceability ,energy,popularity, acousticness,explicit,liveness,instrumentalness,key,mode features to predict the song valence.

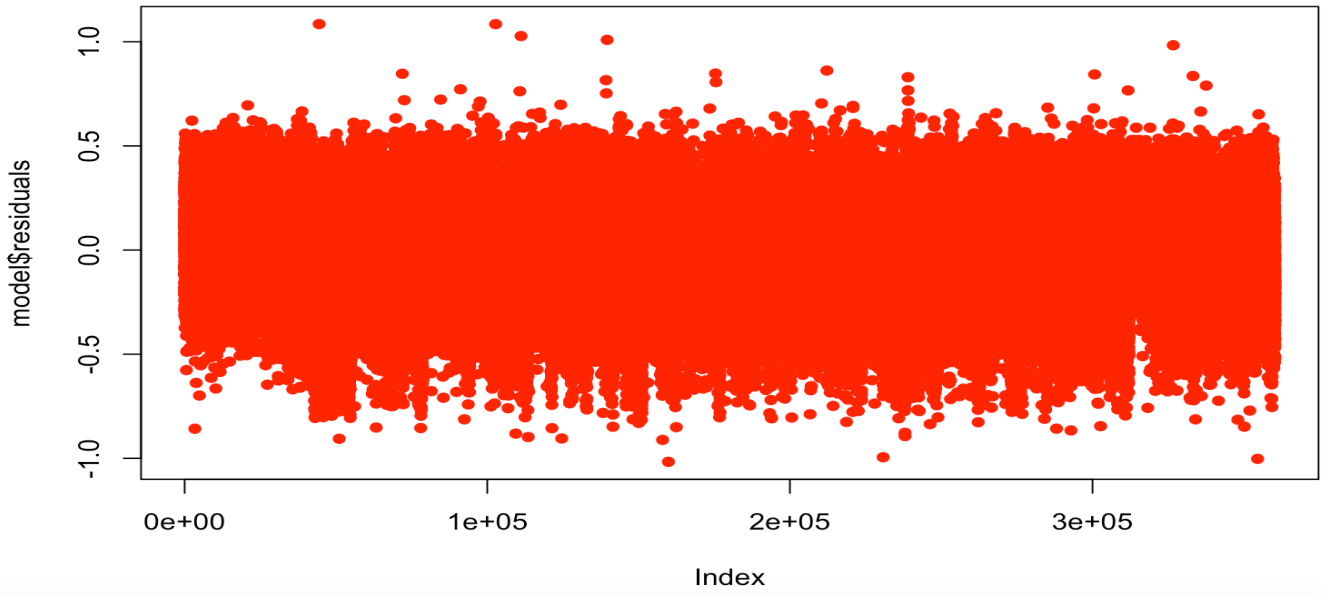
Independent variables:
danceability,energy,popularity,acousticness,explicit,liveness,instrumentalness,key,mode
Dependent variable: valence

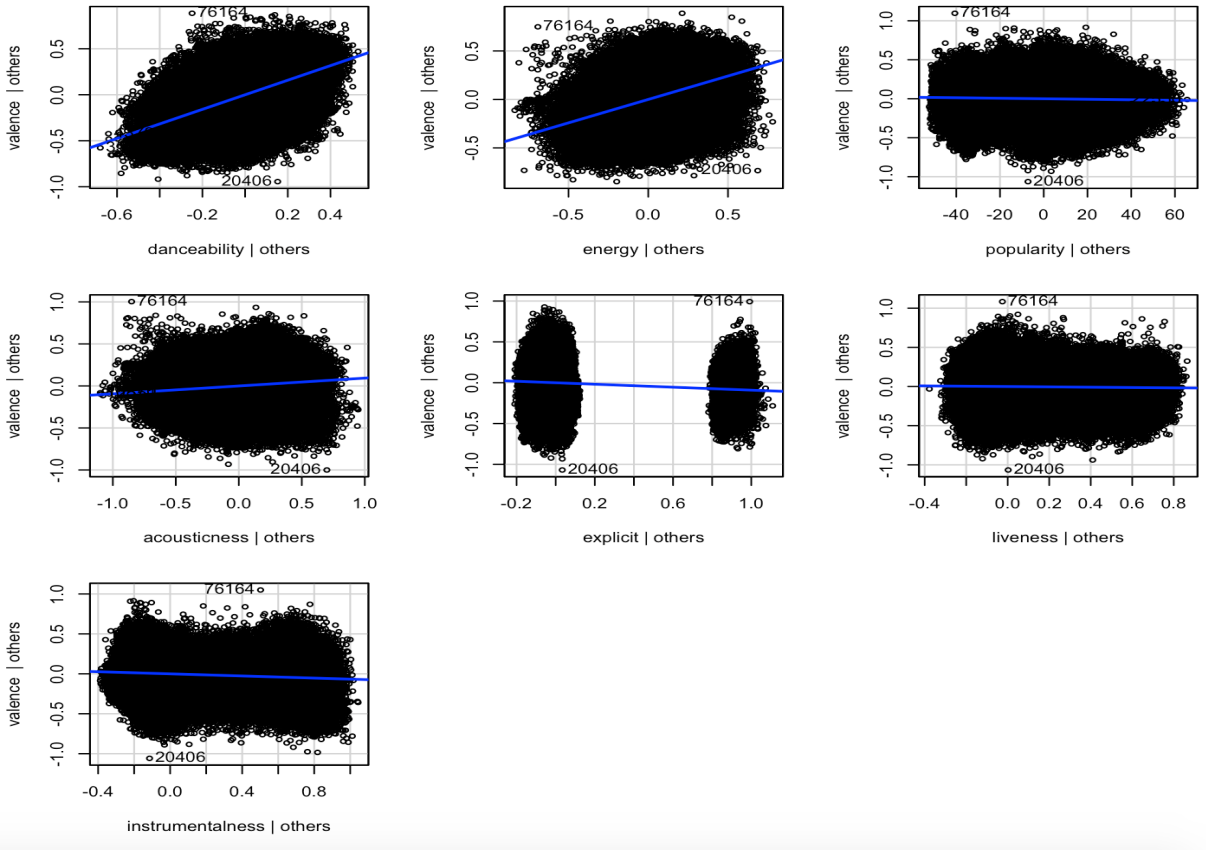
```
Call:
lm(formula = valence ~ danceability + energy + popularity + acousticness +
    explicit + liveness + instrumentalness + key + mode, data = spotify_dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-0.98354 -0.13749  0.00452  0.14500  0.97736

Coefficients:
                  Estimate Std. Error  t value Pr(>|t|)
(Intercept)     -1.378e-01  1.749e-03  -78.806   <2e-16 ***
danceability     7.698e-01  1.702e-03  452.246   <2e-16 ***
energy           4.480e-01  1.534e-03  292.027   <2e-16 ***
popularity      -1.994e-03  1.596e-05 -124.970   <2e-16 ***
acousticness     1.439e-01  1.120e-03  128.458   <2e-16 ***
explicit        -1.100e-01  1.312e-03  -83.845   <2e-16 ***
liveness        -1.615e-02  1.462e-03  -11.047   <2e-16 ***
instrumentalness -5.877e-02  1.051e-03  -55.922   <2e-16 ***
key              6.314e-04  7.534e-05    8.381   <2e-16 ***
mode             2.252e-02  5.606e-04   40.167   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1996 on 576896 degrees of freedom
Multiple R-squared:  0.4011,    Adjusted R-squared:  0.4011
F-statistic: 4.294e+04 on 9 and 576896 DF,  p-value: < 2.2e-16
```





Interpretation of R-squared,residuals and p-value:

- R-squared value is 0.40 which means that the model explains 40% of the total variance in the results.
- The residual standard error is 0.1 which means that most of the data points lie closely to the regression line and the difference between the predicted and the actual output value will be low.
- The p- value is very small for all the independent variables, meaning that these predictor variables are highly significant in predicting the target variable.

Summary:

The Spotify dataset was analyzed and the results and outputs were documented. Data cleaning, EDA, visualizations, hypothesis testing and linear regression was done on various songs and their features.

The below insights were generated after analyzing the Spotify dataset:

- The song writers prefer to compose the songs in major mode than the minor mode as the no. of songs in major mode is greater that of minor mode over the years 1930-2021.
- The keys C and G are the most used keys for composing the songs in the decades 1930s - 2020s.
- The average duration or length of the songs is expected to approach below 3.5 mins as there is a downward trend observed in the last 20 years 2000-2021.
- Most popular songs are slightly shorter in duration than the least popular songs. This highlights a fact that the attention span of the average music consumer is falling and it suggests that shorter songs have a higher chance of gaining more attention and can reach the majority of the audience.
- Music has become more energetic and loud over time.
- Electronic instruments are mostly used in the songs than the acoustic instruments.

- The song positivity (song valence) is decreasing from the decade 1980s.
- The speechiness attribute of the songs is increasing in the last 20 years and this indicates the rise in the number of audio podcasts in the Spotify platform.
- A common pattern observed in the top 1000 popular songs released in the years 2000-2021 : high energy, danceability,loudness , less speechiness, liveness, acousticness and neutral valence score.

Hypothesis testing was conducted and the below conclusion were drawn:

- **<u>One sample t-test conclusions:</u>**
  - The mean tempo level of the songs is not equal to 120 Beats per minute.
- **<u>Two sample t-test conclusions:</u>**
  - The mean tempo levels of the songs vary in major mode songs and minor mode songs. The tempo level of the major mode songs is found to be slightly greater than the tempo level of the minor mode songs.
  - Songs in major mode do sound much happier than the songs in minor mode.This can be attributed to the fact that the songs played in major mode are usually happier,merrier, cheerful tunes and hence a slight increase in valence level can be observed.

Linear regression was used to build a model to predict the song valence based on various song features:

- Multiple models were created to predict the song valence score with different combinations of input variables.
- The best set of input variables to the model was found using the regsubsets() function of the leaps package in R.
- The residuals were plotted for the model and the residual error was very less in predicting the target variable.
- The model was able to explain around 40% of the total variance in the results.

<u>References:</u>

- https://www.kaggle.com/. (2019). *Retrieved from*. https://www.kaggle.com/datasets/yamaerenay/spotify-dataset-19212020-600k-tracks?select=tracks.csv
- *Robert I. Kabacoff. (2015).* R in Action,second edition. Manning Publications Co. www.manning.com
- *Allan G.Bluman. (2018)*. 558010983-Elementary-Statistics-a-Step-by-Step-Approach-10th-Edition. McGraw-Hill Education
- https://wmich.edu/. (1983). Song keys and their nature. *Retrieved from*. https://wmich.edu/mus-theo/courses/keys.html
- https://web.archive.org/. (2013). Plotting music's Emotional valence. *Retrieved from*. https://web.archive.org/web/20170422195736/http://blog.echonest.com/post/66097438564/plotting-musics-emotional-valence-1950-2013
- https://www.digitaltrends.com/. (2015). Play it in G!. *Retrieved from*.

https://www.digitaltrends.com/music/whats-the-most-popular-music-key-spotify/

- https://developer.spotify.com/. Get Track's Audio Features. *Retrieved from*. https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features

Appendix:

```
#-------------------------------------------------------------------------------#
# Shyamala Venkatakrishnan                                        12/15/2022 #
#                                                                            #
#                          ALY6010: Module 6 Final project                   #
#-------------------------------------------------------------------------------#


install.packages(c("FSA","FSAdata","dplyr","plotrix","ggplot2","moments","stringr"
                   ,"data.table","sqldf","tidyverse","wordcloud","RColorBrewer",
                   "devtools","plotly","gmodels","formattable","tidyr"))

loadlibrary <- c("FSA","FSAdata","dplyr","plotrix","ggplot2","moments",
                 "stringr","data.table","sqldf","tidyverse","wordcloud","RColorBrewer",
                 "devtools","plotly","plotme","gmodels","formattable","tidyr")

lapply(loadlibrary,require,character.only=TRUE)

spotify_df <- read.csv("Spotify_tracks.csv",header=TRUE,stringsAsFactors = FALSE,
                       na.string = "")

#No. of rows and columns of a dataset
dim(spotify_df)
#Summary of the dataset
summary(spotify_df)
#First few rows from the dataset
head(spotify_df)
#Size of the dataset
file.info("Spotify_tracks.csv")$size
#Structure of the dataset
str(spotify_df)

install.packages('DataExplorer')
library(DataExplorer)

#dropping unwanted columns - id, id_artists
spotify_df = select(spotify_df, -id, -id_artists)
#Correcting the format for the "releae_date" column.
library(lubridate)

spotify_df <- spotify_df %>%
  mutate(release_date = as.character(release_date),
         release_month= if_else(grepl('-', release_date),
                        month(as.Date(release_date, origin = '1899-12-30')),
                        0),
         release_date = if_else(grepl('-', release_date),
                       as.Date(release_date, origin = '1899-12-30'),
```

```
                        as.Date(release_date, format = "%Y")),
         release_year = year(release_date))

remove_bracket_artists <- unlist(lapply(spotify_df$artists, function(x)
  gsub("\\[|\\]", "", x)))

spotify_df$artists <- remove_bracket_artists

remove_quotes_artists <- unlist(lapply(spotify_df$artists, function(x)
  gsub("\\'|\\'", "", x)))

spotify_df$artists <- remove_quotes_artists

#Num of Records with null/missing values:
na_count <-sapply(spotify_df, function(y) sum(length(which(is.na(y)))))
print(na_count)

plot_missing(spotify_df)

#Removing the records with missing values.
clean_spotify_df <- na.omit(spotify_df)

#Scaling the data - converting duration_ms to duration_mins:

clean_spotify_df$duration_ms <- round(clean_spotify_df$duration_ms/60000,2)
setnames(clean_spotify_df, "duration_ms", "duration_min")

boxplot(clean_spotify_df$release_year,
        main="Box plot of release_year column")

#Let us do the analysis over the songs released after the year 1930
#hence deleting the records with release year less than 1930

spotify_dataset <- subset(clean_spotify_df,
                          subset=(clean_spotify_df$release_year>=1930))

boxplot(spotify_dataset$release_year)

#Mapping key to its corresponding tone as per the pitch class notation:
spotify_dataset = spotify_dataset %>%
  mutate(
    key_class = case_when(
      key == 0 ~ "C",
      key == 1 ~ "C#",
      key == 2 ~ "D",
      key == 3 ~ "D#",
      key == 4 ~ "E",
      key == 5 ~ "F",
      key == 6 ~ "F#",
      key == 7 ~ "G",
      key == 8 ~ "G#",
      key == 9 ~ "A",
      key == 10 ~ "A#",
      key == 11 ~ "B"
    )
```

```r
  )

spotify_dataset = spotify_dataset %>%
  mutate(
    Decade = case_when(
      startsWith(as.character(release_year),"193") ~ "1930s",
      startsWith(as.character(release_year),"194") ~ "1940s",
      startsWith(as.character(release_year),"195") ~ "1950s",
      startsWith(as.character(release_year),"196") ~ "1960s",
      startsWith(as.character(release_year),"197") ~ "1970s",
      startsWith(as.character(release_year),"198") ~ "1980s",
      startsWith(as.character(release_year),"199") ~ "1990s",
      startsWith(as.character(release_year),"200") ~ "2000s",
      startsWith(as.character(release_year),"201") ~ "2010s",
      startsWith(as.character(release_year),"202") ~ "2020s",
    )
  )

spotify_dataset = spotify_dataset %>%
  mutate(
    mode_class = case_when(
      mode == 1 ~ "Major",
      mode == 0 ~ "Minor"
    )
  )

#Descriptive summary statistics:
install.packages("psych")
library(psych)

formattable(describe(spotify_dataset),
            caption = "Descriptive statistics summary of the Spotify dataset")
describe(spotify_dataset)


#Computing the correlation matrix:
plot_correlation(spotify_dataset, type = 'continuous')

#EDA:
install.packages("GGally")
library(GGally)

corr_df <- select(spotify_dataset, -release_year,
                  -release_month,-time_signature,-tempo,-key,-mode,
                  -explicit,-duration_min,-speechiness)

ggcorr(corr_df, method = c("everything", "pearson"))

corr_df <- sqldf("select * from corr_df where release_year >= 2010")

ggpairs(corr_df, title="correlogram with ggpairs()")

#1. Popularity vs energy, danceability,loudness,acousticness -
#heat maps with density
nrow(eliminated)
```

```
pop_energy = ggplot(spotify_dataset,aes(x=energy,y=popularity)) +
  ggtitle("Popularity vs Energy") +
  xlab("Energy") +
  ylab("Popularity")

pop_danceability = ggplot(spotify_dataset,aes(x=danceability,y=popularity)) +
  ggtitle("Popularity vs Danceability") +
  xlab("Danceability") +
  ylab("Popularity")

pop_loudness = ggplot(spotify_dataset,aes(x=loudness,y=popularity)) +
  ggtitle("Popularity vs Loudness") +
  xlab("Loudness") +
  ylab("Popularity")

pop_acoustic = ggplot(spotify_dataset,aes(x=acousticness,y=popularity)) +
  ggtitle("Popularity vs Acousticness") +
  xlab("Acousticness") +
  ylab("Popularity")

pop_instrument = ggplot(spotify_dataset,aes(x=instrumentalness,y=popularity)) +
  ggtitle("Popularity vs Instrumentalness") +
  xlab("Instrumentalness") +
  ylab("Popularity")

p1 = pop_energy +
  geom_point(alpha = 0.01, colour="orange") +
  geom_density2d() +
  theme_bw()

p2 = pop_danceability +
  geom_point(alpha = 0.01, colour="orange") +
  geom_density2d() +
  theme_bw()


p3 = pop_loudness +
  geom_point(alpha = 0.01, colour="orange") +
  geom_density2d() +
  theme_bw()

p4 = pop_acoustic +
  geom_point(alpha = 0.01, colour="orange") +
  geom_density2d() +
  theme_bw()


p5 = pop_instrument +
  geom_point(alpha = 0.01, colour="orange") +
  geom_density2d() +
  theme_bw()

grid.arrange(p1,p2,p3,p4,p5,ncol=2,nrow=3)
```

```r
#2. Energy-loudness - heat maps
energy_loudness = ggplot(spotify_dataset,aes(x=energy,y=loudness)) +
  ggtitle("Energy vs Loudness") +
  xlab("Energy") +
  ylab("Loudness")

p6 = energy_loudness +
  geom_point(alpha = 0.01, colour="pink") +
  geom_density2d() +
  theme_bw()

#3. No of songs in Major mode vs minor mode - decade wise - grouped bar plot
mode_decade_no_of_songs <- sqldf("select mode_class, Decade,count(name) as count
                                 from spotify_dataset
                          group by Decade,mode_class")
library(lattice)
library(viridis)

ggplot(mode_decade_no_of_songs,aes(x = Decade, y = count, fill = mode_class)) +
  geom_bar(stat = "identity", position = "dodge")+
  scale_fill_viridis(discrete = T, option='E')+
  theme_bw()+
  ggtitle("No of songs in major and minor mode over the decades")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))+
  xlab("Decade")+ylab("No. of songs released")

#4. Percentage of songs in each key over the decades - stacked bar plot
key_decade_count <- sqldf("select Decade,key_class,count(name) as count
                          from spotify_dataset
                          group by Decade,key_class")

overall_key_count <- sqldf("select key_class,count(name) as count
                          from spotify_dataset
                          group by key_class")

overall_key_count <- overall_key_count %>%
  mutate(Decade = 'Overall')
combined <- rbind(key_decade_count,overall_key_count)
combined %>%
  group_by(Decade) %>%
  mutate(pct= prop.table(count) * 100) %>%
  ggplot() + aes(Decade, pct, fill=key_class) +
  geom_bar(stat="identity") +
  ylab("Percentage of songs") +
  geom_text(aes(label=paste0(sprintf("%1.1f", pct),"%")),
            position=position_stack(vjust=0.5)) +
  ggtitle("Most preferred choice of key by the musicians - C and G") +
  theme_bw()+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))+
  xlab("Decade")+ylab("Percentage of songs")

#5. Top 5 popular songs in each decade,C major,G major. - tree map.
```

```
top_5_popular_songs <- sqldf("select * from
                             (select name,artists,release_year,Decade,
                             popularity,  mode_class,key_class,
                             row_number() over (partition by
                             Decade,key_class order by popularity desc) as n from
                             spotify_dataset where key_class in ('C','G') and
                             mode_class = 'Major'
                              order by Decade desc) as x
                               where n <= 5 ")

install.packages("treemap")
library(treemap)

##Coloring the boxes by a measure##
treemap(top_5_popular_songs,
        index = c("Decade","mode_class","key_class","name"),
        vSize ="n",vColor = "popularity",type="value",
        title="Top 5 popular songs in each decade, in C and G major keys")

top_5_popular_songs %>%
  count(Decade, mode_class, key_class, name,wt = popularity) %>%
  count_to_treemap()

#8. Top 10 happy,cheerful songs 2000-2021:
df_20s <- sqldf("select * from spotify_dataset where release_year >= 2000")

happy_songs_top_10 <- sqldf("select name,artists,release_year,
                            valence, danceability,energy,loudness,
                            popularity from df_20s where
                            valence between 0.75 and 1 and
                            danceability between 0.75 and 1 and
                            energy between 0.75 and 1 and
                            loudness between -20 and 0
                            order by popularity desc limit 10")

formattable(happy_songs_top_10, caption = "Top 10 popular,happy,energetic,
            foot tapping songs 2000-2021")

#11. How is the duration of the songs changing in the years 1930-2021?

duration_year_df <- sqldf("select release_year, avg(duration_min)
                          as Avg_duration_min from spotify_dataset
                         group by release_year order by Avg_duration_min desc")

ggplot(duration_year_df, aes(x=release_year, y=Avg_duration_min)) +
  geom_line( color="blue") +
  geom_point() +
  theme_ipsum() +
  ylim(1,5)+
  scale_x_continuous(limits=c(1925, 2021))+
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  xlab("Release year of songs") +ylab("Avg. duration of songs in mins")+
  ggtitle("How is the duration of the songs changing in the years 1930-2021?")
```

```r
df_20s_more_popular <- sqldf("select * from df_20s
                             where popularity > 70")

df_20s_least_popular <- sqldf("select * from df_20s
                             where popularity > 0 and popularity <= 30")

df_20s_least_popular <- df_20s_least_popular[sample(nrow(df_20s_least_popular),
                                                    5500,replace = F),]

ggplot(data = df_20s_more_popular, mapping = aes(x = duration_min)) +
  geom_histogram(fill="orange")+
  xlab("Duration of songs in mins")+ylab("No. of songs")+
  xlim(0,5)+
  ggtitle("Histogram of duration of songs 2000-2021 with popularity of 70 to 100")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))+
  theme_bw()

ggplot(data = df_20s_least_popular, mapping = aes(x = duration_min)) +
  geom_histogram(fill="orange")+
  xlab("Duration of songs in mins")+ylab("No. of songs")+
  xlim(0,5)+
  ggtitle("Histogram of duration of songs 2000-2021 with popularity of 0 to 30")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))+
  theme_bw()

more_popular<- ggplot(data = df_20s_more_popular, mapping = aes(x = duration_min)) +
  geom_rug(color = 'hotpink') +
  xlim(0,5)+
  xlab("Duration of songs in mins")+
  ggtitle("Average duration of songs with popularity of 70-100 in the years 2000-2021")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))+
  theme_bw()+
  stat_function(fun = ~dnorm(.x,
                             mean = mean(df_20s_more_popular$duration_min,na.rm=TRUE),
                             sd = sd(df_20s_more_popular$duration_min,na.rm=TRUE))
                ,col = "hotpink")+
  geom_vline(xintercept = 3.49, linetype="dotted",
             color = "black", linewidth=1.5)+
  annotate("text", x=4, y=0.4, label= "Mean duration_mins = 3.49")

least_popular<- ggplot(data = df_20s_least_popular, mapping = aes(x = duration_min)) +
  geom_rug(color = 'hotpink') +
  xlim(0,5)+
  xlab("Duration of songs in mins")+
  ggtitle("Average duration of songs with popularity of 0-30 in the years 2000-2021")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))+
  theme_bw()+
  stat_function(fun = ~dnorm(.x,
                             mean = mean(df_20s_least_popular$duration_min,na.rm=TRUE),
                             sd = sd(df_20s_least_popular$duration_min,na.rm=TRUE))
                ,col = "hotpink")+
```

```
  geom_vline(xintercept = 3.9, linetype="dotted",
             color = "black", linewidth=1.5)+
  annotate("text", x=4.3, y=0.4, label= "Mean duration_mins = 3.90")

#How is the song valence distributed every year?
mood_songs_df <- sqldf("select release_year, avg(valence) as Avg_valence
                        from spotify_dataset
                        group by release_year")

ggplot()+ geom_boxplot(data = spotify_dataset,
                       aes(x=Decade,y=valence, fill=Decade))+
  theme_bw()+
  ggtitle("Distribution of song valence")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))+
  labs(subtitle = "0 = negative music, 1 = positive music")

#Music has become more electric and energetic over time
trend_en_da_ac <- sqldf("select release_year, avg(acousticness) as acousticness
,avg(energy) as avg_energy, avg(danceability) as avg_danceability
                        from spotify_dataset
                        group by release_year")

legend_colors <- c("Acousticness" = "blue", "Energy" = "#458B00",
                   "Danceability" = "red")

ggplot()+
  geom_line(data = trend_en_da_ac,mapping = aes(x=release_year,
                                        y=acousticness,color="Acousticness"))+
  geom_line(data = trend_en_da_ac,mapping = aes(x=release_year,
                                        y=avg_energy,color="Energy"))+
  geom_line(data = trend_en_da_ac,mapping = aes(x=release_year,
                                    y=avg_danceability,color="Danceability"))+
  theme_ipsum() +scale_x_continuous(limits=c(1925, 2021))+theme(axis.text.x=element_text(angle=60,
hjust=1)) +labs(color = "Song features") + scale_color_manual(values = legend_colors) +
  xlab("Release year of songs") +ggtitle("Music has become more electric and energetic over time")+
  geom_vline(xintercept = 1950, linetype="dotted",color = "orange", linewidth=1.5)+
  geom_vline(xintercept = 1980, linetype="dotted",color = "orange", linewidth=1.5)+
  annotate("text", x=1970, y=0.9, label= "Rise of Rock and Roll music")+
  annotate("text", x=2003, y=0.8,label= "Use of Drum machines,Synthesizers")

#Loudness of tracks
loudness <- sqldf("select release_year, avg(loudness) as loudness
                        from spotify_dataset
                        group by release_year")

ggplot()+
  geom_line(data = loudness,mapping = aes(x=release_year,
                                          y=loudness),color="#FF1493")+
  theme_ipsum() +ylim(-60,0)+scale_x_continuous(limits=c(1925, 2021))+
  theme(axis.text.x=element_text(angle=60, hjust=1)) +xlab("Release year of songs") +
  ggtitle("Loudness of the songs over the years")

#Pattern of top 100 popular songs in the recent years 2000-2021
top_1000 <- sqldf("select * from spotify_dataset where release_year >= 2000
```

```
                  order by popularity desc limit 1000")

top_1000_pattern <- sqldf("select avg(valence),avg(danceability),avg(energy)
                ,avg(acousticness),avg(liveness),
                  avg(speechiness),avg(loudness),avg(tempo)
                       from top_1000")

install.packages("fmsb")
library(fmsb)

df <- data.frame(Valence=c(1, 0, 0.50),
                 Danceability=c(1, 0, 0.67),
                 Energy=c(1, 0, 0.63),
                 Acousticness=c(1, 0, 0.20),
                 Liveness=c(1, 0, 0.16),
                 Speechiness=c(1, 0, 0.10),
                 Loudness=c(0,-60,-6))
radarchart(df,
           axistype=1 ,
           pcol=rgb(0.2,0.5,0.5,0.9) , pfcol=rgb(0.2,0.5,0.5,0.5) , plwd=4 ,
           cglcol="black", cglty=1, axislabcol="black", caxislabels=seq(0,100,25), cglwd=0.8,
           vlcex=0.8 )

#Top 1000 popular songs - key %,mode % - donut, pie
key_table <- table(top_1000$key_class)

key_table_perc <- round((key_table / 1000) * 100, digits = 2)

key_table_df <- data.frame(key_table_perc)

colnames(key_table_df)[1] <- "key_class"
colnames(key_table_df)[2] <- "Percentage"

donut_key_top_1000 <- ggplot(data = key_table_df, aes(x=2,y=Percentage,
                                                      fill = key_class))+
  geom_col(color = "black")+
  coord_polar("y", start=1)+
  geom_text(aes(label = paste(Percentage,"%", sep = "")),
            position = position_stack(vjust = 0.5)) +
  theme_void() +
  scale_fill_brewer(palette = "Paired")+
  xlim(.2,2.5)

mode_table <- table(top_1000$mode_class)
mode_table_perc <- round((mode_table / 1000) * 100, digits = 2)
mode_table_df <- data.frame(mode_table_perc)

colnames(mode_table_df)[1] <- "Mode_class"
colnames(mode_table_df)[2] <- "Percentage"

ggplot(mode_table_df, aes(x = "", y = Percentage, fill = Mode_class)) +
  geom_col() + geom_text(aes(label = paste(Percentage,"%", sep = ""))
                         ,position = position_stack(vjust = 0.5)) +
  guides(fill = guide_legend(title = "Category")) +
  coord_polar(theta = "y") +
```

```
  theme_void() +
  scale_fill_brewer(palette = "Set2")+
  ggtitle("Percentage of mode class in top 1000 popular songs 2000 - 2021")


install.packages("caret",dep = TRUE)
library(caret)


install.packages("scales")
library("scales")

#Removing outliers from these variables:
#popularity,loudness, danceability,speechiness,liveness

#Popularity:
Q <- quantile(spotify_dataset$popularity, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(spotify_dataset$popularity)
up <-  Q[2]+1.5*iqr
low<- Q[1]-1.5*iqr
eliminated<- subset(spotify_dataset, spotify_dataset$popularity > (Q[1] - 1.5*iqr)
                    & spotify_dataset$popularity < (Q[2]+1.5*iqr))

#loudness:
Q <- quantile(eliminated$loudness, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(eliminated$loudness)
up <-  Q[2]+1.5*iqr
low<- Q[1]-1.5*iqr
eliminated<- subset(eliminated, eliminated$loudness > (Q[1] - 1.5*iqr)
                    & eliminated$loudness < (Q[2]+1.5*iqr))


#danceability:
Q <- quantile(eliminated$danceability, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(eliminated$danceability)
up <-  Q[2]+1.5*iqr
low<- Q[1]-1.5*iqr

eliminated<- subset(eliminated, eliminated$danceability > (Q[1] - 1.5*iqr)
                    & eliminated$danceability < (Q[2]+1.5*iqr))

#speechiness:
Q <- quantile(eliminated$speechiness, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(eliminated$speechiness)
up <-  Q[2]+1.5*iqr
low<- Q[1]-1.5*iqr
eliminated<- subset(eliminated, eliminated$speechiness > (Q[1] - 1.5*iqr)
                    & eliminated$speechiness < (Q[2]+1.5*iqr))

#liveness:
Q <- quantile(eliminated$liveness, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(eliminated$liveness)
up <-  Q[2]+1.5*iqr
low<- Q[1]-1.5*iqr
eliminated<- subset(eliminated, eliminated$liveness > (Q[1] - 1.5*iqr)
```

```
                    & eliminated$liveness < (Q[2]+1.5*iqr))

#Duration_mins
Q <- quantile(eliminated$duration_min, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(eliminated$duration_min)
up <-  Q[2]+1.5*iqr
low<- Q[1]-1.5*iqr
eliminated<- subset(eliminated, eliminated$duration_min > (Q[1] - 1.5*iqr)
                    & eliminated$duration_min < (Q[2]+1.5*iqr))

#Linear regression:
#Model 1: Predicting song popularity

devtools::install_github("kassambara/ggpubr")
library("ggpubr")
popularity <- ggdensity(eliminated$popularity,
                        xlab = "popularity")

popularity_cutoff <- 75
spotify_sample_1 <-
  spotify_dataset[spotify_dataset$popularity >= popularity_cutoff,]

spotify_sample_2 <-
  spotify_dataset[spotify_dataset$popularity < popularity_cutoff,]

spotify_sample_2_random <- spotify_sample_2[sample(nrow(spotify_sample_2),
                                                   3000,replace = F),]
spotify_sample_combined =
  rbind(spotify_sample_1,spotify_sample_2_random)

set.seed(1)
training.samples <- createDataPartition(spotify_sample_combined$popularity,
                                        p = 0.8, list = FALSE)

head(training.samples,10)
train.data  <- spotify_sample_combined[training.samples, ]
head(train.data,10)
test.data <- spotify_sample_combined[-training.samples, ]
head(test.data, 10)
train.data$loudness <- rescale(train.data$loudness,0,1)
train.data$key <- rescale(train.data$key,0,1)
test.data$loudness <- rescale(test.data$loudness,0,1)
test.data$key <- rescale(test.data$key,0,1)

model_2 <- lm(popularity ~ energy + danceability + loudness + valence
              + acousticness + explicit + liveness +
                speechiness + instrumentalness,
            data = train.data)

# Summarize the model
summary(model_2)
plot(model_2$residuals, pch = 16, col = "red")

library(car)
avPlots(model_2)
```

```r
# Make predictions
predictions <- model_2 %>% predict(test.data)

# Model performance
# (a) Prediction error, RMSE
RMSE(predictions, test.data$popularity)

# (b) R-square
R2(predictions, test.data$popularity)

newdata <- data.frame(
  energy = 0.8, danceability = 0.75
)

model_2 %>% predict(newdata)

#Model 2 :  Predicting song valence:
install.packages("leaps")
library(leaps)

spotify_numeric_df <- select(spotify_dataset,popularity, explicit,danceability,
                             energy,loudness,speechiness,acousticness,instrumentalness,
                             liveness,valence,key,mode)

model_1 <- regsubsets(valence~., data = spotify_numeric_df,
                      nvmax = 5)
summary(model_1)

model1 <- lm(valence ~ danceability, data = spotify_numeric_df)
model2 <- lm(valence ~ danceability+energy, data = spotify_numeric_df)
model3 <- lm(valence ~ danceability+energy+popularity, data = spotify_numeric_df)
model4 <- lm(valence ~ danceability+energy+popularity+acousticness
             , data = spotify_numeric_df)

model5 <- lm(valence ~ danceability+energy+popularity+acousticness+explicit+liveness+
               instrumentalness+key+mode
             , data = spotify_dataset)

summary(model1) #0.27
summary(model2) #0.34
summary(model3) #0.37
summary(model4) #0.38
summary(model5) #0.40

plot(model5$residuals, pch = 16, col = "red")

avPlots(model5)
```