Coursework Project-Fundamentals of Information Visualisation(COMP3021)

Implementing Visualizations on a Dataset in R

Basic Description about the DataSet

The Dataset that has been chosen by me for exploratory analysis and to implement Various Visualisation Techniques is called "SPOTIFY SONGS" which is a CSV file that contains a lot of music data along with its corresponding audio features data that can be used to classify, explore and visualize data extensively on the basis of exploratory data analysis and different kinds of visualization techniques.I will be making majority of my visualizations on pop music from this dataset.Spotify is a Swedish-based audio streaming and media services provider, which launched in October 2008. It is now one of the biggest digital music, podcast, and video streaming service in the world that gives access to millions of songs from artists all over the world.

Basic Description about the data dictionary

Here are all the variable columns that are involved with a short description about them

track_id: The unique identification alphanumeric assigned to every song

track_name: Name of the track

track_artist: Name of the artist

track_popularity: A metric that shows how popular that track is .(100: Most Popular,0:Least Popular)

track_album_id: The unique identification alphanumeric assigned to every album

track_album_name: Name of the album

track_album_release_date: Release date of the album

playlist_name: Name of the playlist assigned to that song

playlist_id:The unique identification alphanumeric assigned to every playlist

playlist_genre: The genre of the playlist

playlist_subgenre:The sub-genre of the playlist

Danceability:Describes how suitable a track is for dancing by keeping in mind the rhythm and tempo of that song.(100: Most Danceable,0:Least Danceable)

Valence: Describes how positive the track is from 1 to 0. Happier songs will have a valence closer to 1 and sadder songs will have a valence closer to 0.

Acousticness: A measure to determine whether the track is acoustic or not.

Key: Estimated overall key of the track. If no key was detected then its assigned -1, otherwise 0 = C, 1 = C/D, 2 = D and so on.

Energy: How energetic the track is, measures the intensity and adrenaline of the track.Range of the values assigned will be from 0.0 to 1.0

Loudness: The overall loudness of the track in decibels. The values range from -60 to 0.

Speechiness: Describes the intensity of the spoken words in a track. Values range from 0 to 1 where values ranging from 0.66 to 1 describe that the song is entirely made u of spoken words. 0.33 to 0.66 describe a mixture of both music and spoken words.Less than 0.33 predominantly describes tracks that are purely musical and have very little spoken words

Mode: The modality of a track. Major-1, Minor-0

Instrumentalness: A measure to determine the intensity of the instrumentalness of a track.

Liveness:Measure to determine the presence of a live studio audience in a track

Tempo: The speed or the pace of the track. The higher the speed of the track, the higher is the tempo in BPM

Song_Duration(in ms): Duration of the songs(in milliseconds)

Setting up and loading the data for preliminary data exploratory analysis

Initially it is important to load the data into the rStudio and get a summary of the data we are dealing with extensively

```
##    track_id          track_name         track_artist       track_popularity
##  Length:32833       Length:32833       Length:32833       Min.   :  0.00  
##  Class :character   Class :character   Class :character   1st Qu.: 24.00  
##  Mode  :character   Mode  :character   Mode  :character   Median : 45.00  
##                                                           Mean   : 42.48  
##                                                           3rd Qu.: 62.00  
##                                                           Max.   :100.00  
##  track_album_id     track_album_name   track_album_release_date
##  Length:32833       Length:32833       Length:32833            
##  Class :character   Class :character   Class :character        
##  Mode  :character   Mode  :character   Mode  :character        
##                                                                
##                                                                
##                                                                
##  playlist_name      playlist_id        playlist_genre     playlist_subgenre 
##  Length:32833       Length:32833       Length:32833       Length:32833      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##   danceability        energy              key            loudness      
##  Min.   :0.0000   Min.   :0.000175   Min.   : 0.000   Min.   :-46.448  
##  1st Qu.:0.5630   1st Qu.:0.581000   1st Qu.: 2.000   1st Qu.: -8.171  
##  Median :0.6720   Median :0.721000   Median : 6.000   Median : -6.166  
##  Mean   :0.6548   Mean   :0.698619   Mean   : 5.374   Mean   : -6.720  
##  3rd Qu.:0.7610   3rd Qu.:0.840000   3rd Qu.: 9.000   3rd Qu.: -4.645  
##  Max.   :0.9830   Max.   :1.000000   Max.   :11.000   Max.   :  1.275  
##       mode         speechiness      acousticness    instrumentalness   
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000000  
##  1st Qu.:0.0000   1st Qu.:0.0410   1st Qu.:0.0151   1st Qu.:0.0000000  
```

```
## Median :1.0000   Median :0.0625   Median :0.0804   Median :0.0000161
## Mean   :0.5657   Mean   :0.1071   Mean   :0.1753   Mean   :0.0847472
## 3rd Qu.:1.0000   3rd Qu.:0.1320   3rd Qu.:0.2550   3rd Qu.:0.0048300
## Max.   :1.0000   Max.   :0.9180   Max.   :0.9940   Max.   :0.9940000
##    liveness         valence            tempo         duration_ms
## Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   :  4000
## 1st Qu.:0.0927   1st Qu.:0.3310   1st Qu.: 99.96   1st Qu.:187819
## Median :0.1270   Median :0.5120   Median :121.98   Median :216000
## Mean   :0.1902   Mean   :0.5106   Mean   :120.88   Mean   :225800
## 3rd Qu.:0.2480   3rd Qu.:0.6930   3rd Qu.:133.92   3rd Qu.:253585
## Max.   :0.9960   Max.   :0.9910   Max.   :239.44   Max.   :517810
```

The data is successfully loaded up with all the column attributes showing their basic metrics such as minimum,maximum, mean etc..

Exploratory data analysis

Checking for missing values and how to deal with them successfully

To check for the number of missing values(if any) in the dataset we will be using is.na() function and will use its result in the sum function as shown below.The is.na() is expected to return a dataset of values consisiting of boolean values of False and True and if a value is not available it will return "TRUE". We can also infer from this small value of mean that the percentage of missing values in our data is extremely low which is a desirable thing to have

```
## [1] 15
```

```
## [1] 1.986337e-05
```

As we can clearly see that there are 15 missing values in our dataset and we can deal with these missing values with the help of the omit function which will return the dataset with the incomplete values removed as follows:

na.omit(spotify_Songs)

```
## # A tibble: 32,828 x 23
##    track_id               track_name track_artist track_popularity track_album_id
##    <chr>                  <chr>      <chr>                   <dbl> <chr>
##  1 6f807x0ima9a1j3VPbc7~ I Don't C~ Ed Sheeran                 66 2oCs0DGTsRO98~
##  2 0r7CVbZTWZgbTCYdfa2P~ Memories ~ Maroon 5                   67 63rPSO264uRjW~
##  3 1z1Hg7Vb0AhHDiEmnDE7~ All the T~ Zara Larsson               70 1HoSmj2eLcsrR~
##  4 75FpbthrwQmzHlBJLuGd~ Call You ~ The Chainsm~               60 1nqYsOef1yKKu~
##  5 1e8PAfcKUYoKkxPhrHqw~ Someone Y~ Lewis Capal~               69 7m7vv9wlQ4i0L~
##  6 7fvUMiyapMsRRxr07cU8~ Beautiful~ Ed Sheeran                 67 2yiy9cd2QktrN~
##  7 2OAylPUDDfwRGfe0lYql~ Never Rea~ Katy Perry                 62 7INHYSeusaFly~
##  8 6b1RNvAcJjQH73eZO4BL~ Post Malo~ Sam Feldt                  69 6703SRPsLkS4b~
##  9 7bF6tCO3gFb8INrEDcjN~ Tough Lov~ Avicii                     68 7CvAfGvq4RlIw~
## 10 1IXGILkPm0tOCNeq00kC~ If I Can'~ Shawn Mendes               67 4QxzbfSsVryEQ~
## # ... with 32,818 more rows, and 18 more variables: track_album_name <chr>,
## #   track_album_release_date <chr>, playlist_name <chr>, playlist_id <chr>,
## #   playlist_genre <chr>, playlist_subgenre <chr>, 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>
```

Now if we take a look at the song duration of the dataset it is given in milliseconds which is not a favourable way of measuring time so we can convert it into minutes and seconds which can be done as follows

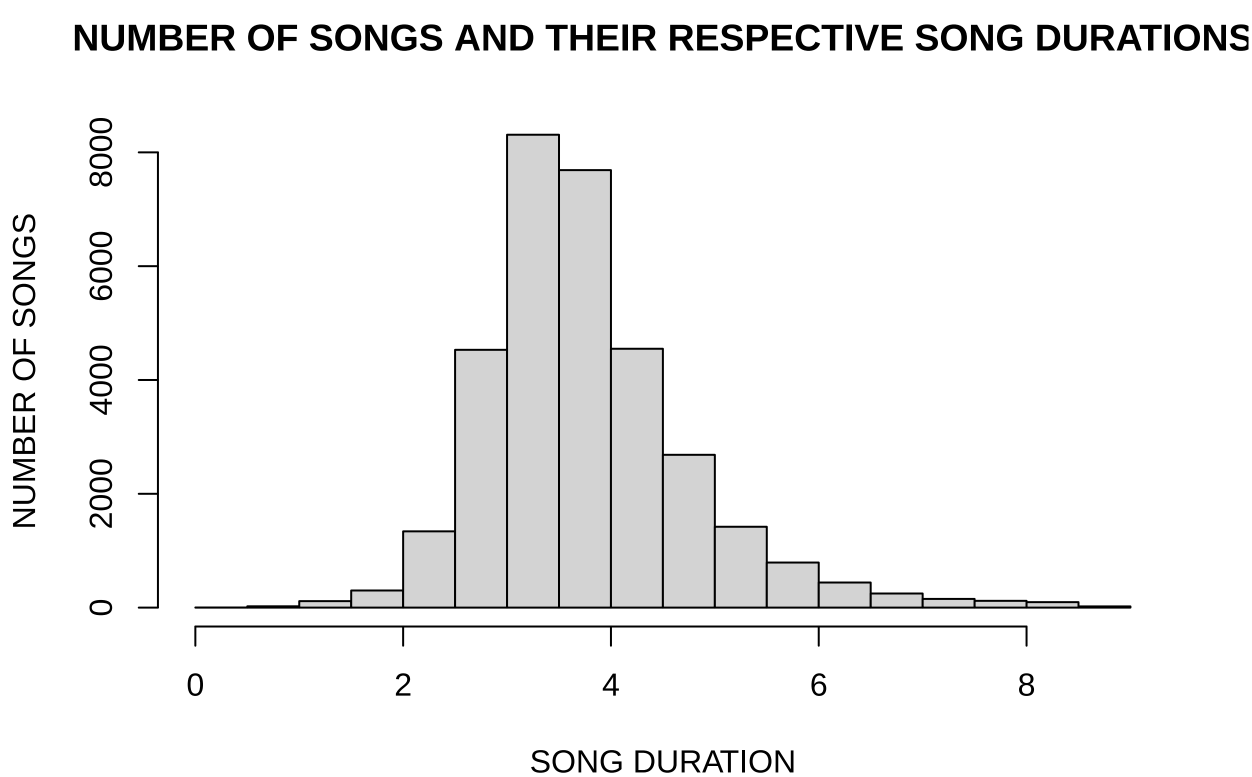
valence	tempo	duration_ms
0.518	122.036	194754
0.693	99.972	162600
0.613	124.008	176616
0.277	121.956	169093

Figure 1: THE SONG DURATION IN MILLISECONDS

valence	tempo	duration_ms
0.518	122.036	3.245900
0.693	99.972	2.710000
0.613	124.008	2.943600
0.277	121.956	2.818217
0.725	123.976	3.150867

Figure 2: THE SONG DURATION IS HENCE CONVERTED TO MINS AND SECONDS FROM MS

After the conversion we can plot a histogram and see what we can infer from it

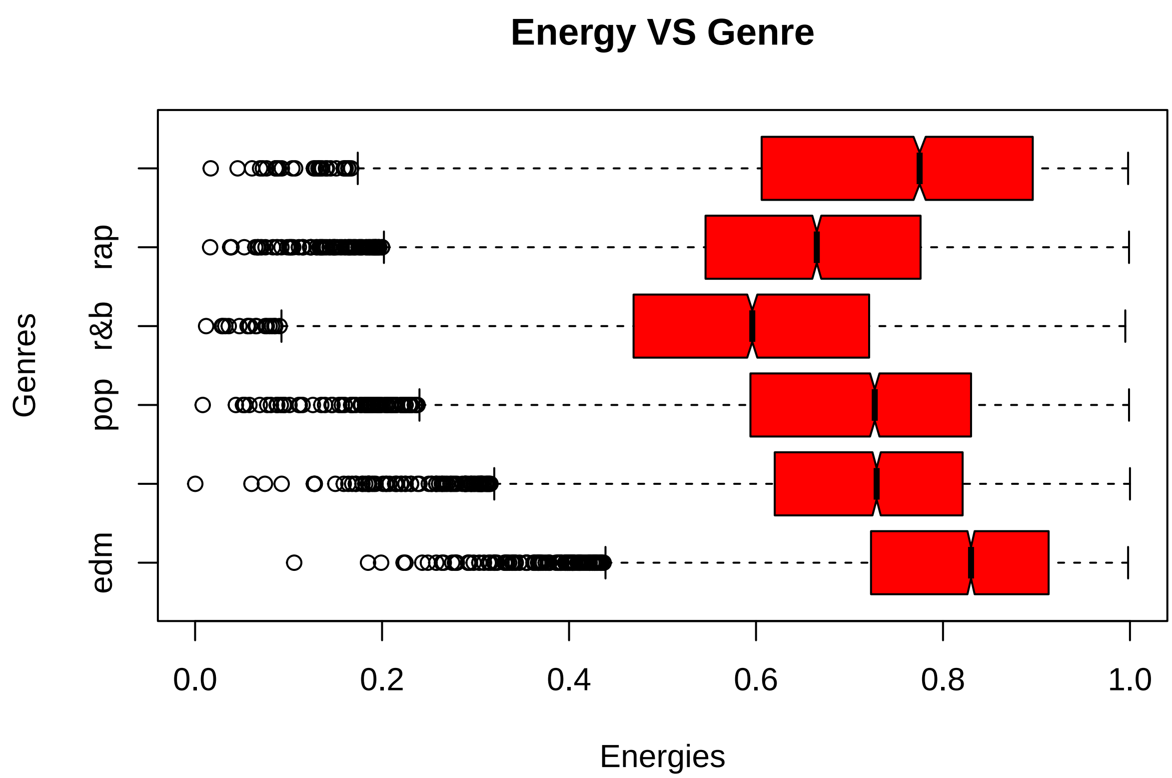


After plotting the histogram, we see it is slightly skewed at the left hand side. For the same histogram we can also derive the inference that most of the songs are 3-4 mins long as they are the ones with the most peaks. We can ask questions from this graph that why aren't songs generally more long.

Next we can start to give an initial analysis towards having a fair idea about the metrics in a plotted format

From these histograms we can observe the following: 1. Maximum songs have the instrumentalness of zero 2. Most of the songs have a tempo ranging from upwards of 100 to 120 3. Majority of the songs have energy values of more than atleast 0.5

We will be making a boxplot that involves all genres plotted against a metric energy to see what kind of data we are actually dealing with.



As we can see Electronic Dance Music is the genre with the maximum energy

Removing data from other genres other than pop and other redundant data

After all this analysis we will remove duplicate data to prevent overlapping between redundant data-points(in case of plotting scatter plots) (eg:same song names that occur more than once linked to different playlists)

```
##    track_id          track_name        track_artist       track_popularity
##  Length:99          Length:99          Length:99          Min.   : 0.0
##  Class :character   Class :character   Class :character   1st Qu.:24.5
##  Mode  :character   Mode  :character   Mode  :character   Median :49.0
##                                                           Mean   :49.0
##                                                           3rd Qu.:73.5
##                                                           Max.   :98.0
##  track_album_id     track_album_name   track_album_release_date
##  Length:99          Length:99          Length:99
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  playlist_name      playlist_id        playlist_genre     playlist_subgenre
##  Length:99          Length:99          Length:99          Length:99
```

```
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   danceability        energy            key            loudness
##  Min.   :0.4490   Min.   :0.2250   Min.   : 0.000   Min.   :-14.454
##  1st Qu.:0.6245   1st Qu.:0.6900   1st Qu.: 2.000   1st Qu.: -7.056
##  Median :0.6730   Median :0.8010   Median : 6.000   Median : -5.219
##  Mean   :0.6673   Mean   :0.7612   Mean   : 5.737   Mean   : -5.686
##  3rd Qu.:0.7230   3rd Qu.:0.8565   3rd Qu.: 8.000   3rd Qu.: -4.231
##  Max.   :0.8800   Max.   :0.9920   Max.   :11.000   Max.   : -2.634
##       mode         speechiness       acousticness      instrumentalness
##  Min.   :0.0000   Min.   :0.02690   Min.   :0.000609   Min.   :0.0000000
##  1st Qu.:0.0000   1st Qu.:0.03940   1st Qu.:0.026800   1st Qu.:0.0000000
##  Median :1.0000   Median :0.05530   Median :0.079400   Median :0.0000078
##  Mean   :0.5657   Mean   :0.07917   Mean   :0.114294   Mean   :0.0352945
##  3rd Qu.:1.0000   3rd Qu.:0.09440   3rd Qu.:0.169000   3rd Qu.:0.0011950
##  Max.   :1.0000   Max.   :0.37500   Max.   :0.902000   Max.   :0.7970000
##     liveness          valence           tempo         duration_ms
##  Min.   :0.0185   Min.   :0.0358   Min.   : 92.98   Min.   :2.204
##  1st Qu.:0.0891   1st Qu.:0.3955   1st Qu.:110.02   1st Qu.:3.089
##  Median :0.1190   Median :0.5090   Median :122.04   Median :3.351
##  Mean   :0.1764   Mean   :0.5096   Mean   :120.01   Mean   :3.457
##  3rd Qu.:0.2205   3rd Qu.:0.6140   3rd Qu.:126.08   3rd Qu.:3.720
##  Max.   :0.7040   Max.   :0.9690   Max.   :180.05   Max.   :7.634
```

As we can see the data has significantly reduced to only 99 songs and their details. We now not only have songs from only one genre(POP), but we also have songs with different popularity indexes that will prevent points from overlapping and will make better visualisations.

We will also be removing columns such as Track_id, Playlist_id and Album_id because they have no use in our dataset

```
##   track_name        track_artist       track_popularity track_album_name
##  Length:99          Length:99          Min.   : 0.0     Length:99
##  Class :character   Class :character   1st Qu.:24.5     Class :character
##  Mode  :character   Mode  :character   Median :49.0     Mode  :character
##                                        Mean   :49.0
##                                        3rd Qu.:73.5
##                                        Max.   :98.0
##  track_album_release_date playlist_name      playlist_genre
##  Length:99                Length:99          Length:99
##  Class :character         Class :character   Class :character
##  Mode  :character         Mode  :character   Mode  :character
##
##
##
##  playlist_subgenre   danceability        energy            key
##  Length:99          Min.   :0.4490   Min.   :0.2250   Min.   : 0.000
##  Class :character   1st Qu.:0.6245   1st Qu.:0.6900   1st Qu.: 2.000
##  Mode  :character   Median :0.6730   Median :0.8010   Median : 6.000
##                     Mean   :0.6673   Mean   :0.7612   Mean   : 5.737
##                     3rd Qu.:0.7230   3rd Qu.:0.8565   3rd Qu.: 8.000
```

```
##                     Max.   :0.8800   Max.   :0.9920   Max.   :11.000
##     loudness           mode          speechiness       acousticness
##  Min.   :-14.454   Min.   :0.0000   Min.   :0.02690   Min.   :0.000609
##  1st Qu.: -7.056   1st Qu.:0.0000   1st Qu.:0.03940   1st Qu.:0.026800
##  Median : -5.219   Median :1.0000   Median :0.05530   Median :0.079400
##  Mean   : -5.686   Mean   :0.5657   Mean   :0.07917   Mean   :0.114294
##  3rd Qu.: -4.231   3rd Qu.:1.0000   3rd Qu.:0.09440   3rd Qu.:0.169000
##  Max.   : -2.634   Max.   :1.0000   Max.   :0.37500   Max.   :0.902000
##  instrumentalness      liveness         valence           tempo
##  Min.   :0.0000000   Min.   :0.0185   Min.   :0.0358   Min.   : 92.98
##  1st Qu.:0.0000000   1st Qu.:0.0891   1st Qu.:0.3955   1st Qu.:110.02
##  Median :0.0000078   Median :0.1190   Median :0.5090   Median :122.04
##  Mean   :0.0352945   Mean   :0.1764   Mean   :0.5096   Mean   :120.01
##  3rd Qu.:0.0011950   3rd Qu.:0.2205   3rd Qu.:0.6140   3rd Qu.:126.08
##  Max.   :0.7970000   Max.   :0.7040   Max.   :0.9690   Max.   :180.05
##   duration_ms
##  Min.   :2.204
##  1st Qu.:3.089
##  Median :3.351
##  Mean   :3.457
##  3rd Qu.:3.720
##  Max.   :7.634
```

Description of the initial questions

The following are some of the questions that can be formed about the dataset initially

Task 1

Q1) Examine the dataset. According to the dataset how does danceability and Track_Popularity compare to each other? Would it be safe to assume that the most danceable song is also the most popular?

As we can see the tracks in the rectangle are not only some of the most popular but also the most danceable. We can analyse this data and infer from it that to make the most popular songs we need to put some emphasis on the danceability factor.

We can also make our visualisations interactive using plotly library and put our cursor at any data point which will in turn tell us the danceability and the track_popularity of that datapoint which can be shown below

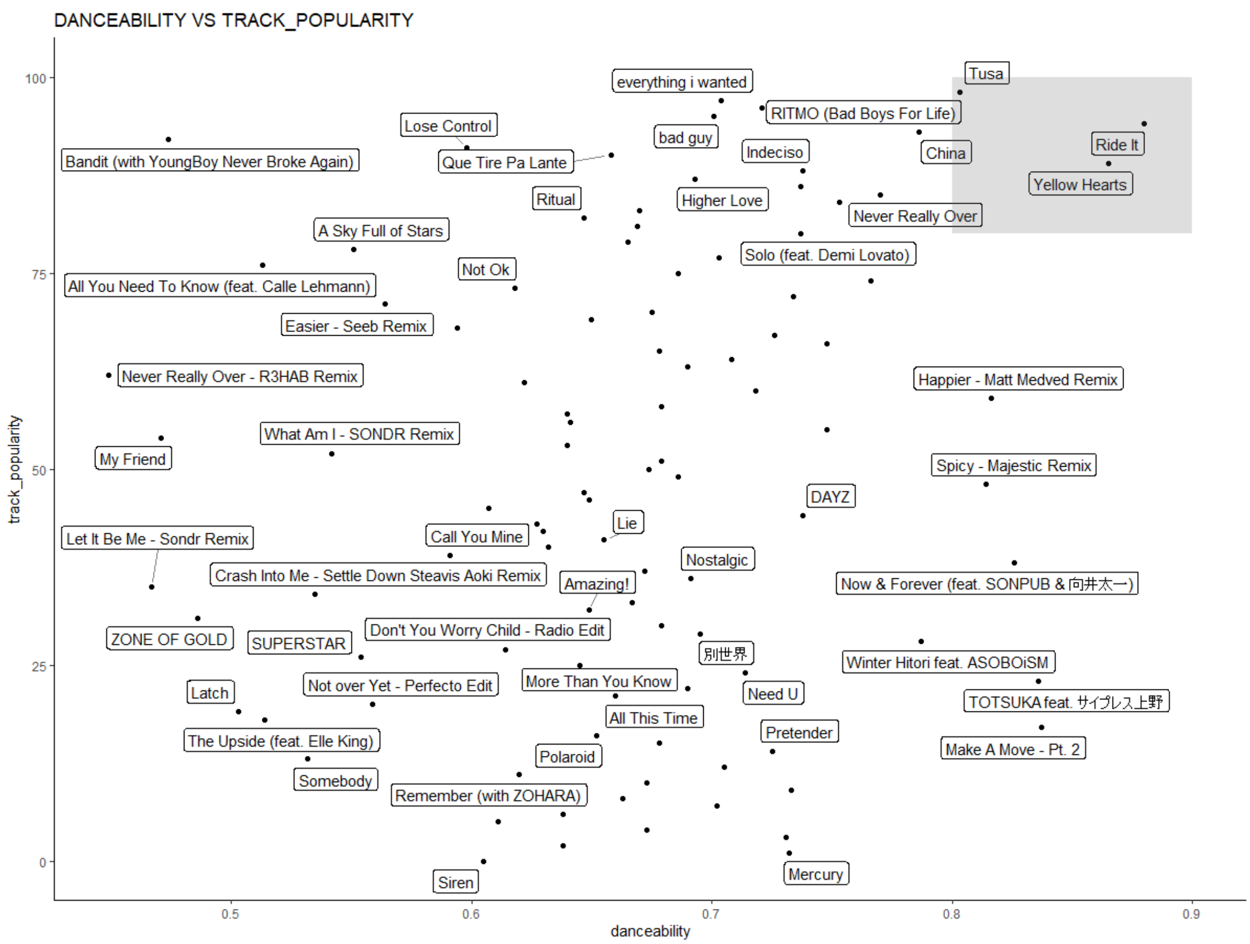
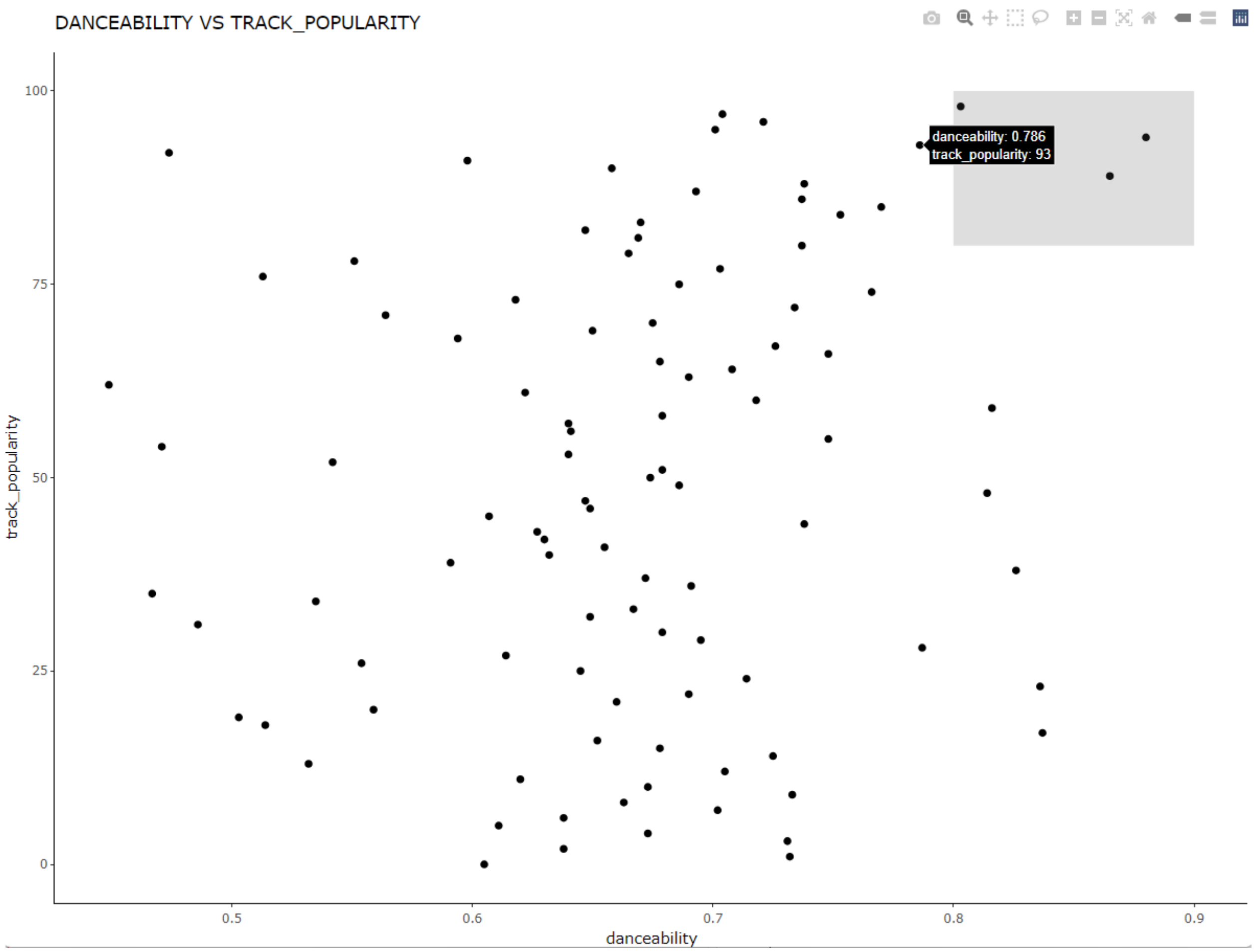
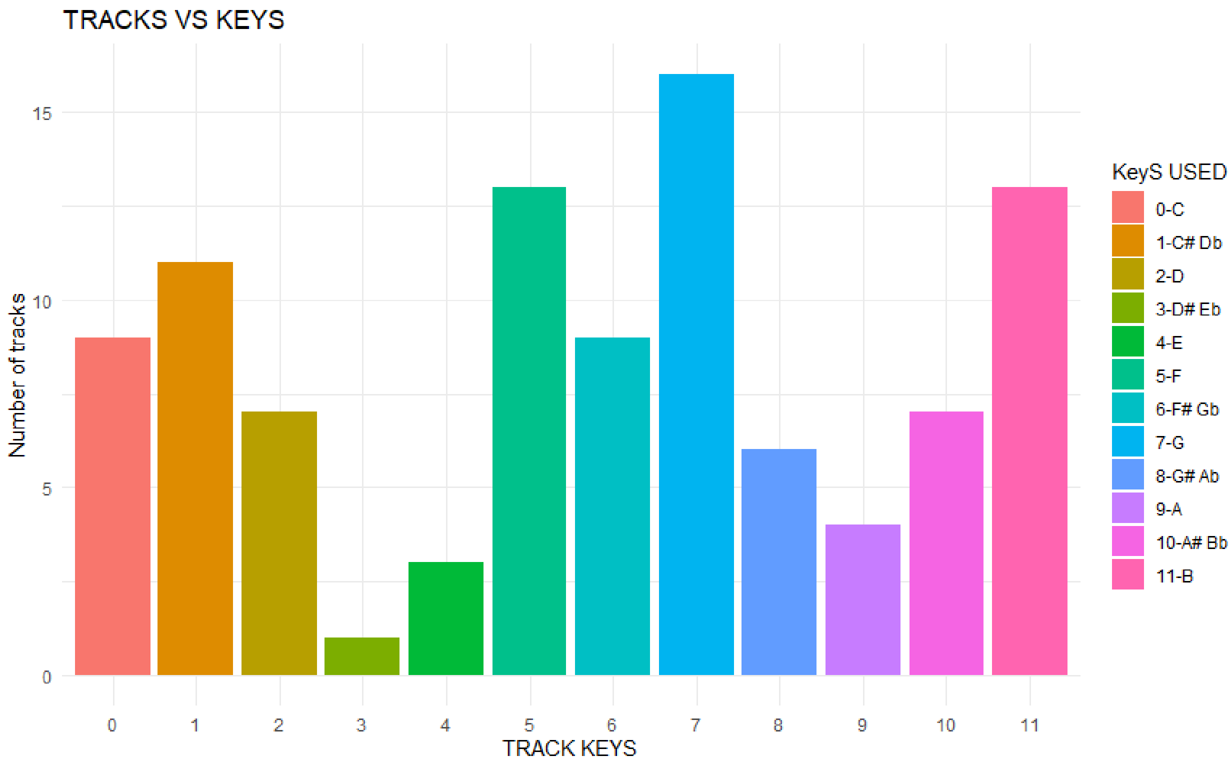


Figure 3: DANCEABILITY VS TRACK_POPULARITY


DANCEABILITY VS TRACK_POPULARITY
danceability: 0.786
track_popularity: 93
track_popularity
danceability
100
75
50
25
0
0.5
0.6
0.7
0.8
0.9

Task 2

Q2)Group the tracks according to the keys. Which key is the favourite for artists to make their pop songs?



As we can see in this bar_graph it is therefore the #7 key which is the G key which is used up the most by the artists to make the songs. The bar graph here gives us the detailed analysis on how many songs use which key and by seeing the count we can make an affirmative inference on which is the most used key.

Task 3

Q3)Define an arbitrary music metric called OPTIMAL FIGURE which can be defined as the following expression:

OPTIMALFIG=DANCEABILITY+ENERGY+SPEEECHINESS+VALENCE+TEMPO-LOUDNESS-LIVENESS. How does the track_popularity vary with the optimal figure.Are there any artists that are popular and have a complete song(high optimal figure.)"

A:First we need to make sure that we define an optimal figure column and put it in our dataset using the mutate function of the dplyr library.

```
library(dplyr)
optimal_figure<-spotify_songs$danceability+spotify_so
ify_songs$acousticness-spotify_songs$liveness+spotify
spotify_songs<-spotify_songs%>%mutate(optimal_figure)
spotify_songs
view(spotify_songs)
```

We see a new column in the dataset defined as the optimal figure for every track

optimal_figure
126.7430
106.7829
129.5438
127.5283

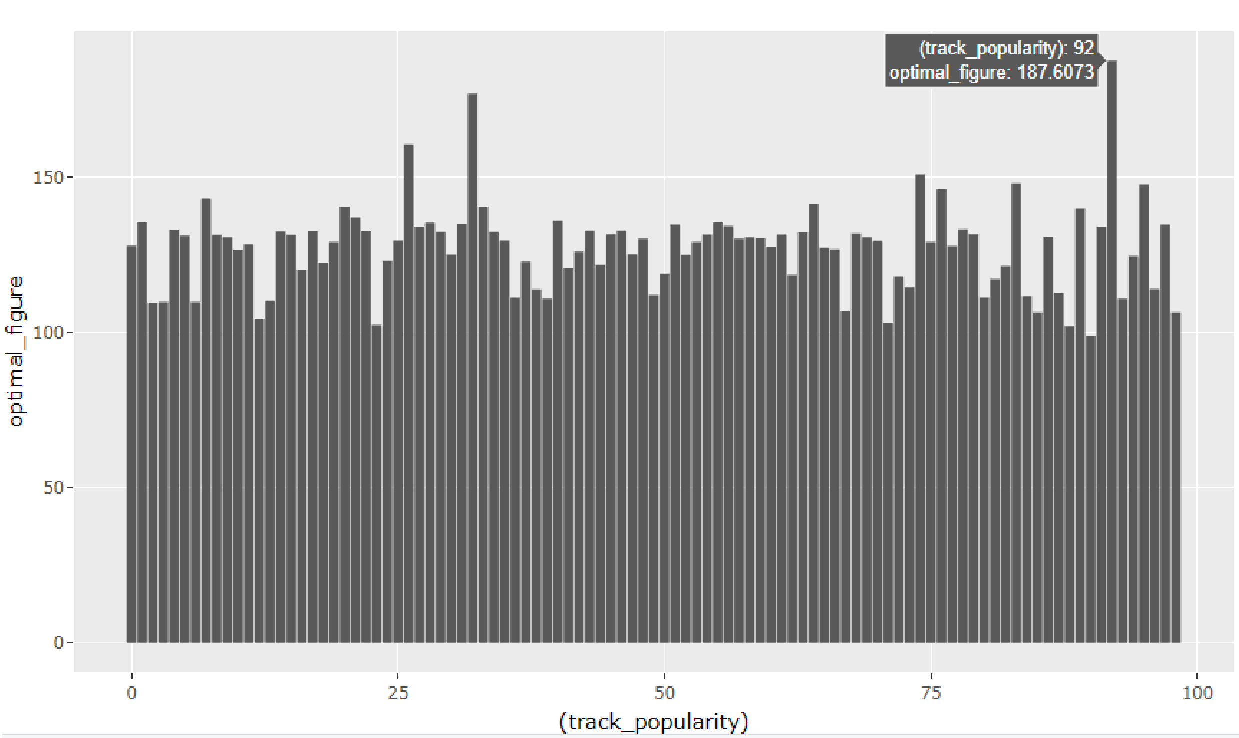
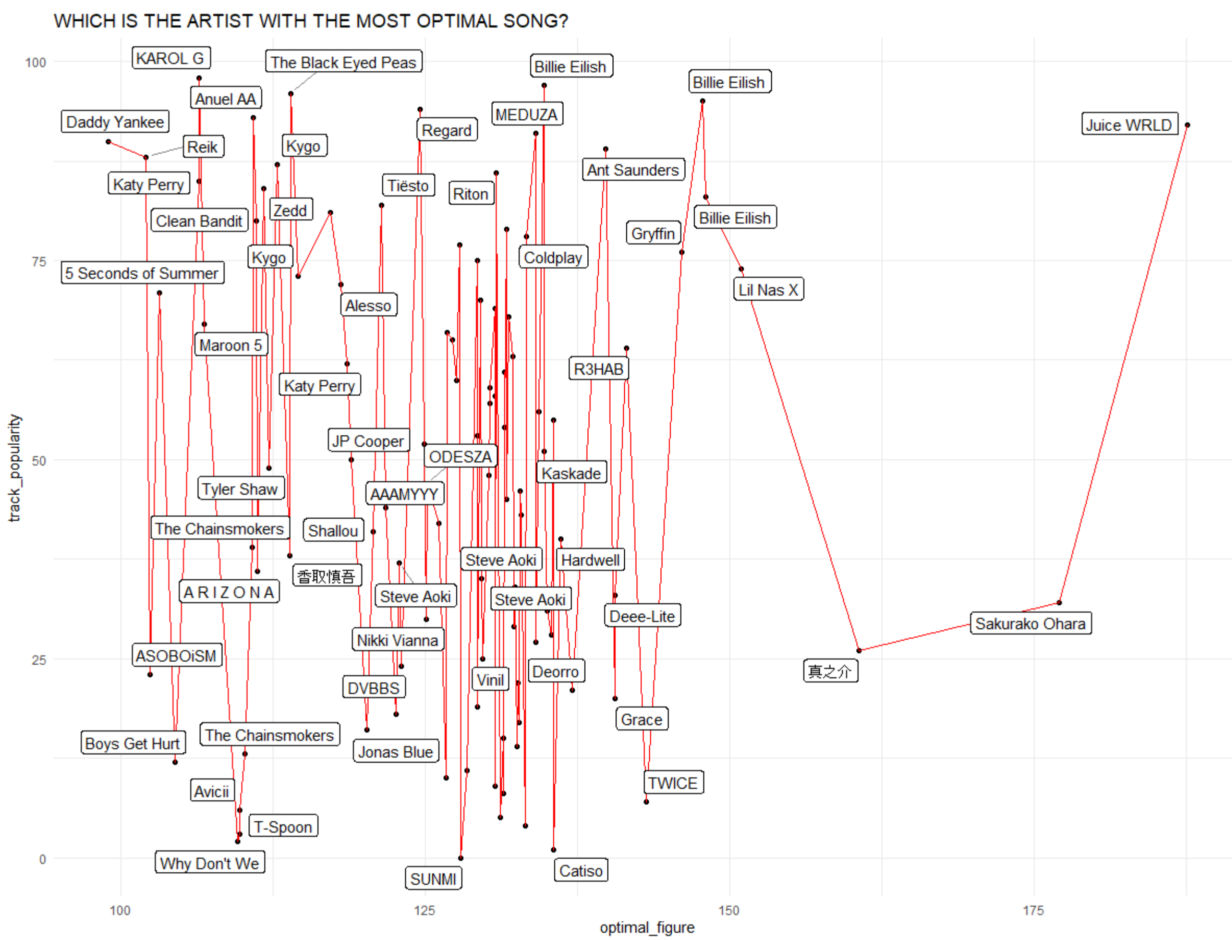


Figure 4: BAR GRAPH SHOWING track_pop and OF



As we can see Juice WRLD with the Track_popularity of 92 and Optimal_Figure of 180.6 makes the cut and he has the complete song.

Task 4

Q: What is the energy distribution of the songs?

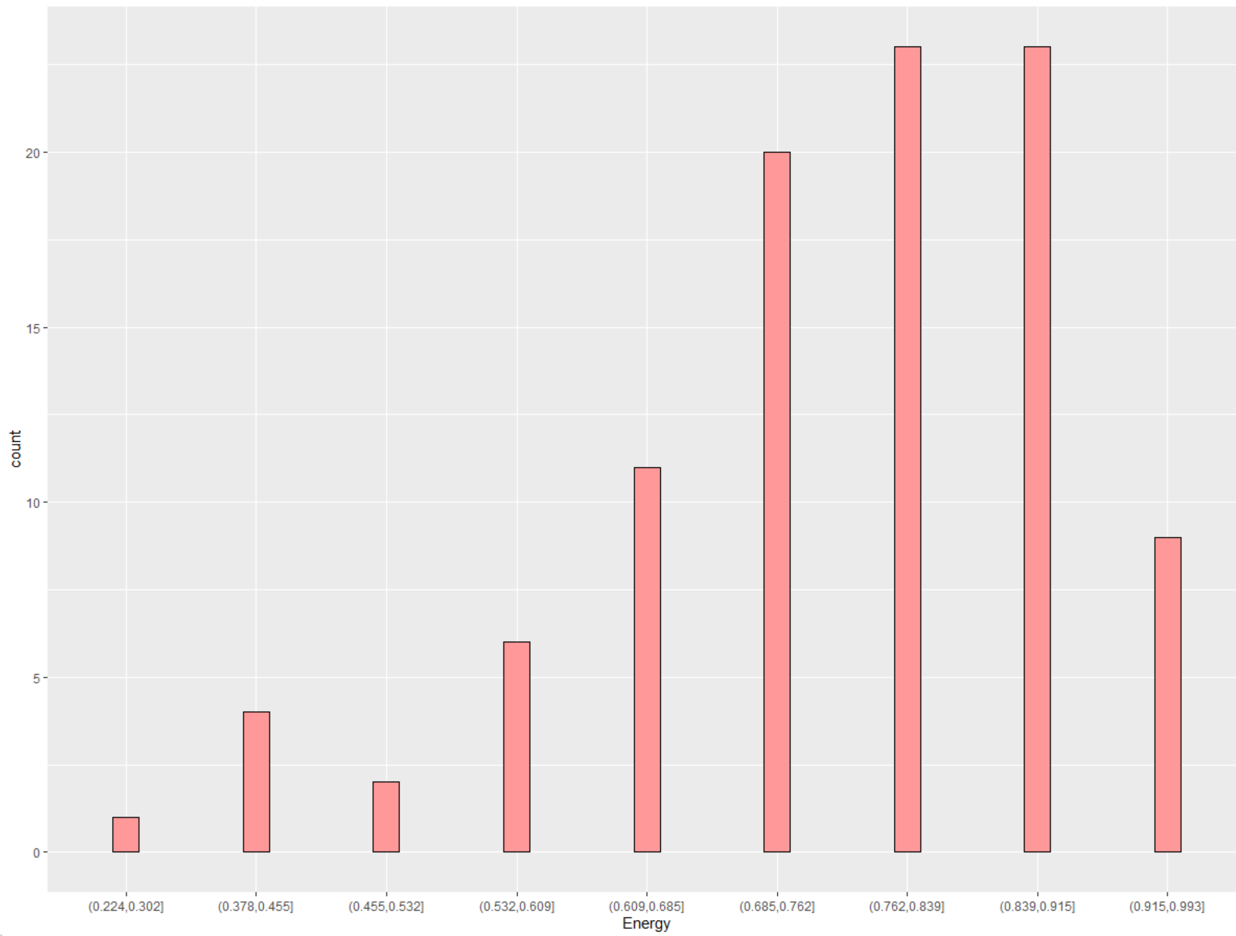


Figure 5: ENERGY DISTRIBUTION

From this it would be safe to infer that Spotify users like to listen to energetic tracks rather than those that are laid back and chilled.

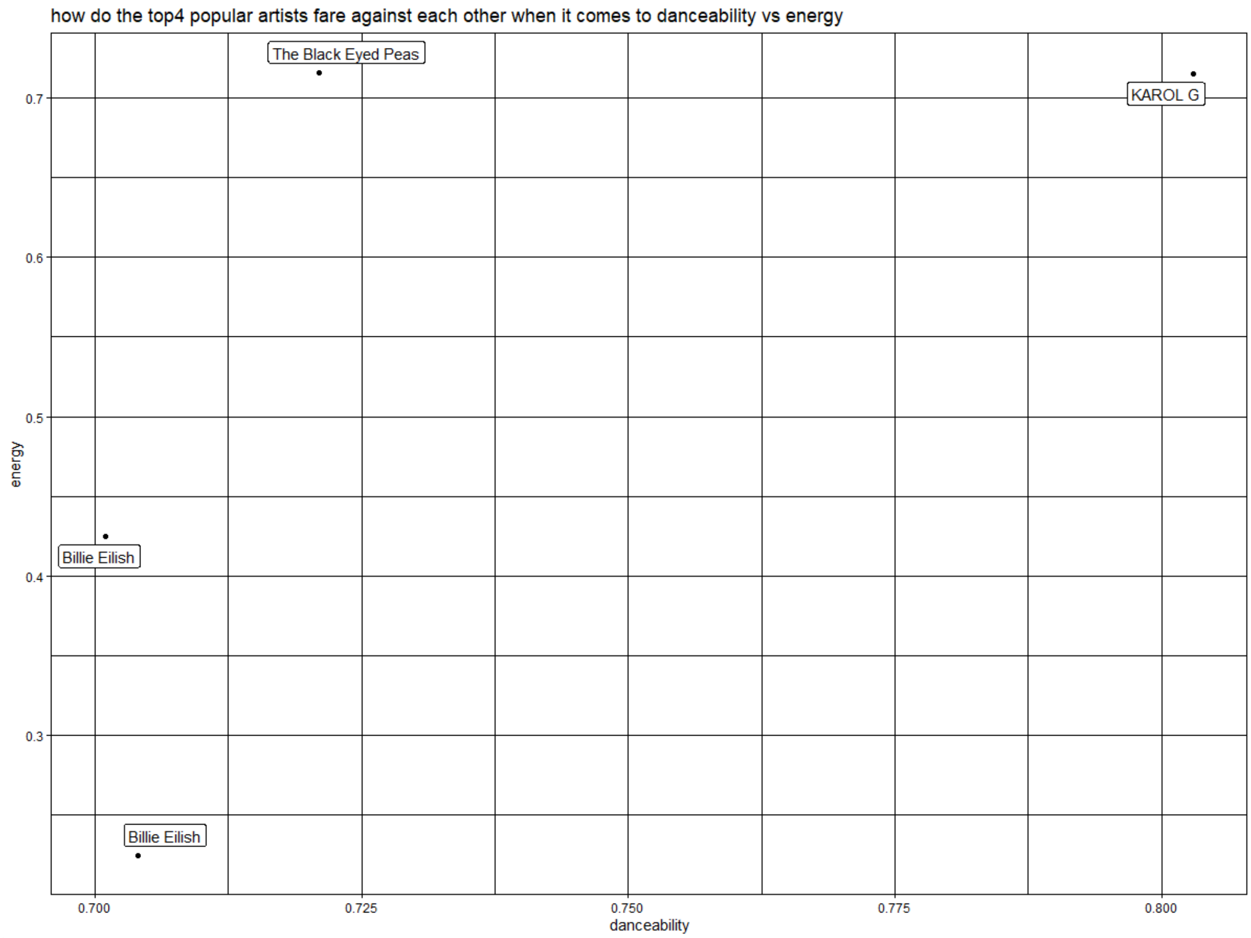
Refining the data and posing some questions

After answering some basic questions on our dataset we can refine a subset of the data and pose some questions on the same

Q.Suppose we have to refine the data of only the top4 artists in the data based on their popularity index and talk about their energy and danceability values and decide which is the perfect artist based on that

A: We will put the dataset into a dataframe and subset only those artists that have a track_popularity that is greater than 94.After that we will use the select() to subset the danceability, energy and artist_name columns from our dataset which will act as an input for the data for our ggplot.

```
```{r echo=FALSE,message=FALSE,warning=FALSE}
data.frame(spotify_songs)
c<-df[df$track_popularity>94,]
t<-select(c,danceability,energy,track_artist)
ggplot(t, aes(x = danceability, y = energy, supp = track_artist)) +geom_point()+geom_label_repel(aes(label =
track_artist), box.padding   = 0.35, point.padding = 0.5, segment.color = 'grey50',max.overlaps = 5) +
theme_linedraw()+ggtitle("how do the top4 popular artists fare against each other when it comes to danceability vs
energy")
```
```



After plotting in ggplot here is what we get. We see that "KAROLG" is the artist with the most linear behaviour but we can also see that "billie Eilish" has two songs in the top4 which could make her the most popular artist out of them all purely based on numbers.

Q:Are any data columns dependent on the track_popularity? Is there muticollinearity?

After taking a look at the correleation matrix plot below it is right to admit that there is no major dependenices between track_popularity. However we can see certain dependencies between columns that are closely related to each other such as Loudness and Energy

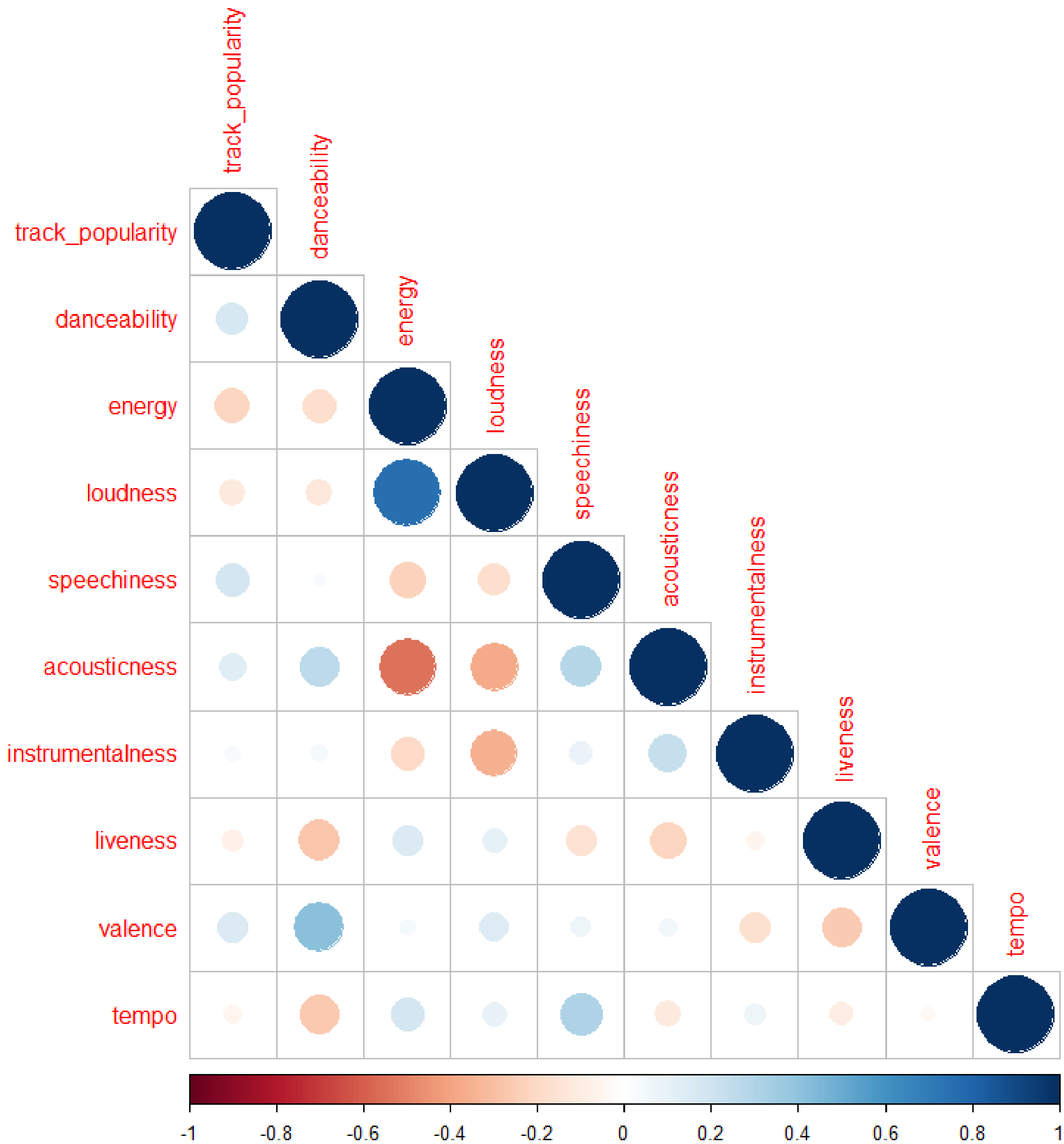


Figure 6: CORRPLOT SHOWING DEPENDENCE