Harvard PH125.9x Data Science Capstone project

Introduction:

Since the inception of Spotify in 2006 in Sweden, it took 15 years for the company to enter one of the largest markets in the world - Indian music streaming market. The underlying reasons for this are the unreliability of internet connection and smartphone penetration among the population But Spotify took a risk and entered the Indian market in 2019.

Indian music business is complex considering the number of official languages recognized in are 21 and there are many other that are spoken. In order to cater to this very diverse audience Spotify needs to understand the culture of the country and each individual language.

The reasons that make Spotify a success are its recommendation algorithm and the curated playlists for occasions such as heartbreak, happiness, road trip and dance. Tracks in these playlists share common features such as Danceability, Loudness, Liveness, Speechiness and many more. An example of implementation of these features is that a track can be classified into a Rap playlist by the level of Speechiness present in the song.

As someone who has grown up listening to Indian music from Telugu, Tamil, Malayalam and Hindi languages, I can attest to the fact that songs popular in Telugu do not sound the same in Hindi because each of these languages have words and pronunciation that are completely different from each other and are unique.

As I was listening to a Telugu song while I was walking across Charles River in MA I was inspired to do this project.

Dataset:
The tracks for this dataset have been taken from the Spotify's Top Telugu Songs for the Year 2019 playlist. I have complied this dataset with audio features data by utilizing Spotify's Web API – Get Audio Features for a Track available on their Spotify for Developers Website.

Through this API I was able to extract the complete set of audio features of each individual track and have complied them into a dataset in Excel.

Code:

```
#Libraries needed
if(!require(tidyverse)) install.packages("tidyverse", repos =
"http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos =
"http://cran.us.r-project.org")
if(!require(readr)) install.packages("readr", repos =
"https://cran.us.r-project.org")
if(!require(plyr)) install.packages("plyr", repos =
"http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos =
"http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos =
"https://cran.us.r-project.org")
if(!require(formattable)) install.packages("formattable", repos
= "http://cran.us.r-project.org")
if(!require(RWeka)) install.packages("RWeka", repos =
"http://cran.us.r-project.org")
if(!require(qdap)) install.packages("qdap", repos =
"https://cran.us.r-project.org")
if(!require(tm)) install.packages("tm", repos =
"http://cran.us.r-project.org")
if(!require(readxl)) install.packages("readxl", repos =
"http://cran.us.r-project.org")
if(!require(tibble)) install.packages("tibble", repos =
"https://cran.us.r-project.org")
if(!require(corrplot)) install.packages("corrplot", repos =
"https://cran.us.r-project.org")
if(!require(rpart)) install.packages("rpart", repos =
"http://cran.us.r-project.org")
if(!require(rpart.plot)) install.packages("rpart.plot", repos =
"http://cran.us.r-project.org")

#Loading the dataset
Spotify_Telugu <- read_excel("Documents/Right now/Spotify
Telugu.xlsx")
```

Literature review:

Audio Features:

Duration: The duration of the track in Milliseconds

Key: The estimated overall key of the track.

Mode: Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
Acousticness: A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

Danceability: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

Energy: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.

Instrumentalness: Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the Instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

Liveness: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

Loudness: The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.

Speechiness: Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks

Valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Tempo: The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

After loading the dataset, I have performed summary statistics.

```
#Summary statistics

head(Spotify_Telugu)
```

```
Name         Album         Artists      Danceability Energy   Key
Loudness  Mode Speechiness Acousticness Instrumentalness
Liveness Valence Tempo Duration
  <chr>        <chr>         <chr>                    <dbl>  <dbl>
<dbl>    <dbl> <dbl>        <dbl>        <dbl>           <dbl>
<dbl>   <dbl> <dbl>    <dbl>
1 Samajavara… Ala Vaikunth… Sid Sriram          0.568  0.663
8    -6.18     1       0.0345        0.91       0.0000482
0.124    0.808 165.    219818
2 He's Soo C… Sarileru Nee… Madhu Priya         0.82   0.801
7    -4.90     0       0.185         0.412      0.000193
0.054    0.963 154.    209649
3 Hoyna Hoyna Gang Leader   Anirudh Rav…        0.713  0.727
6    -7.17     0       0.0388        0.183      0.00000122
0.0756   0.515  97.0   271938
4 Ramuloo Ra… Ala Vaikunth… Anurag Kulk…        0.663  0.913
5    -4.68     0       0.152         0.415      0.000496
0.158    0.805 188.    245760
5 Prema Venn… Chitralahari  Sudharshan …        0.673  0.63
2    -8.75     0       0.0338        0.534      0.0000205
0.0454   0.744  80.0   217875
6 Mind Block  Sarileru Nee… Ranina Reddy        0.922  0.921
7    -5.03     1       0.216         0.645      0.00536
0.0972   0.73  102.    264824
```

```
summary(Spotify_Telugu)
```

```
    Name                 Album                Artists
Danceability        Energy              Key             Loudness
Mode
```

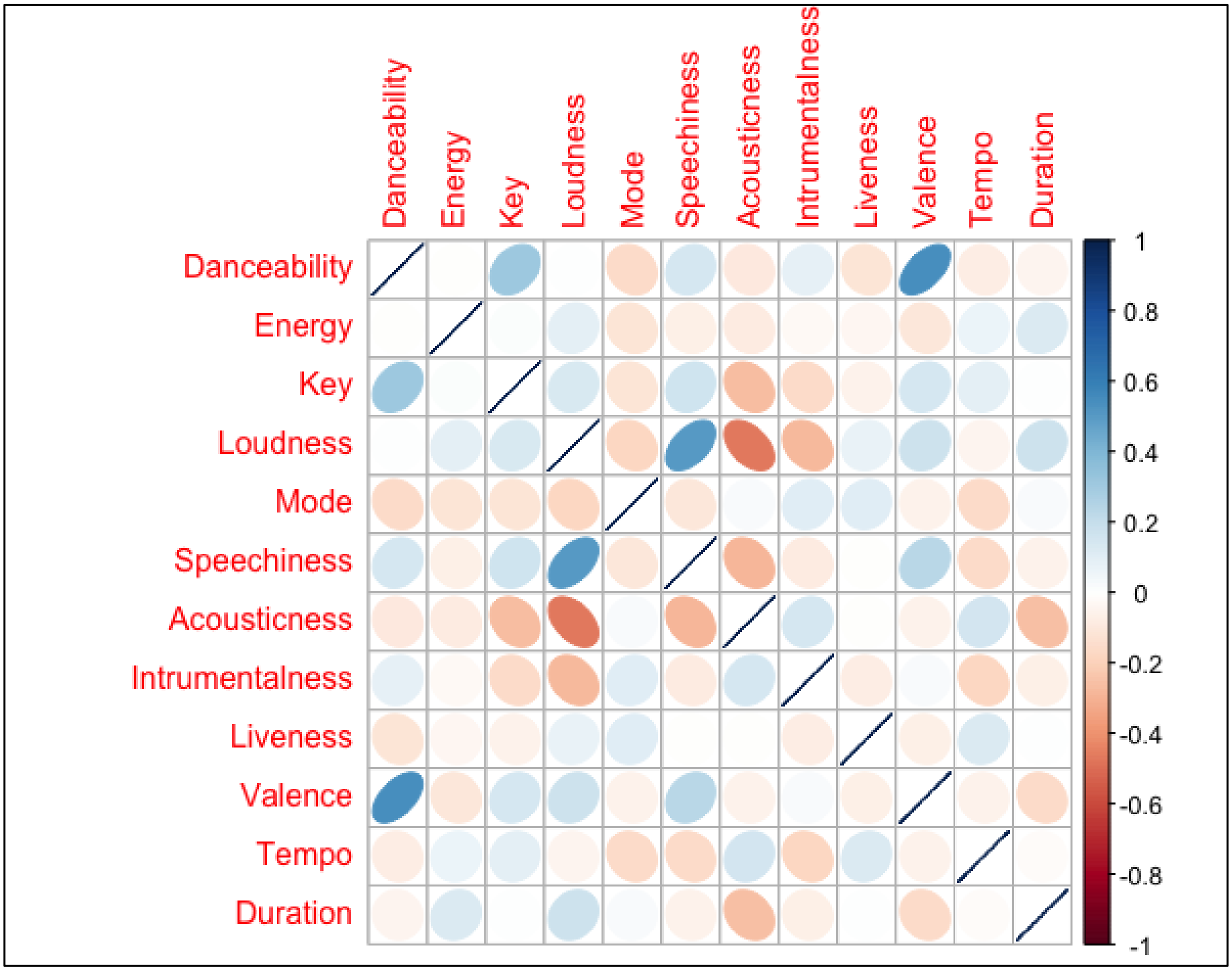
```
 Length:77          Length:77          Length:77          Min.
:0.2670   Min.   :  0.219   Min.   : 0.000   Min.   :-13.793
Min.   :0.0000
 Class :character   Class :character   Class :character   1st
Qu.:0.6030   1st Qu.:  0.535   1st Qu.: 1.000   1st Qu.: -8.255
1st Qu.:0.0000
 Mode  :character   Mode  :character   Mode  :character   Median
:0.6860   Median :  0.668   Median : 4.000   Median : -6.605
Median :1.0000
                                                           Mean
:0.6811   Mean   : 10.107   Mean   : 4.584   Mean   : -6.870
Mean   :0.5065
                                                           3rd
Qu.:0.7730   3rd Qu.:  0.806   3rd Qu.: 8.000   3rd Qu.: -5.219
3rd Qu.:1.0000
                                                           Max.
:0.9220   Max.   :729.000   Max.   :11.000   Max.   : -0.985
Max.   :1.0000
  Speechiness      Acousticness    Intrumentalness
Liveness         Valence          Tempo           Duration
 Min.   :0.0257   Min.   :0.0229   Min.   :0.0000000   Min.
:0.0090   Min.   :0.2200   Min.   : 66.05   Min.   : 24564
 1st Qu.:0.0376   1st Qu.:0.2440   1st Qu.:0.0000000   1st
Qu.:0.0884   1st Qu.:0.4380   1st Qu.: 98.07   1st Qu.:196123
 Median :0.0502   Median :0.4150   Median :0.0000135   Median
:0.1170   Median :0.6440   Median :119.98   Median :242174
 Mean   :0.0984   Mean   :0.4460   Mean   :0.0049750   Mean
:0.1505   Mean   :0.6071   Mean   :122.25   Mean   :234958
 3rd Qu.:0.1490   3rd Qu.:0.6440   3rd Qu.:0.0003510   3rd
Qu.:0.1790   3rd Qu.:0.8000   3rd Qu.:139.51   3rd Qu.:267949
 Max.   :0.3730   Max.   :0.9190   Max.   :0.2160000   Max.
:0.6900   Max.   :0.9630   Max.   :189.99   Max.   :364110
```

To understand the strength of a relationship between these variables will help us in understanding why certain tracks are more popular than the other.

```
#Correlation between variables
Spotify_Telugu_num <- Spotify_Telugu[,-(1:3)]
MCor <- cor(Spotify_Telugu_num)

#Plotting the Correlation
```

```
corrplot(MCor, method = "ellipse")
```

We plot the correlation to better visualize the correlation.

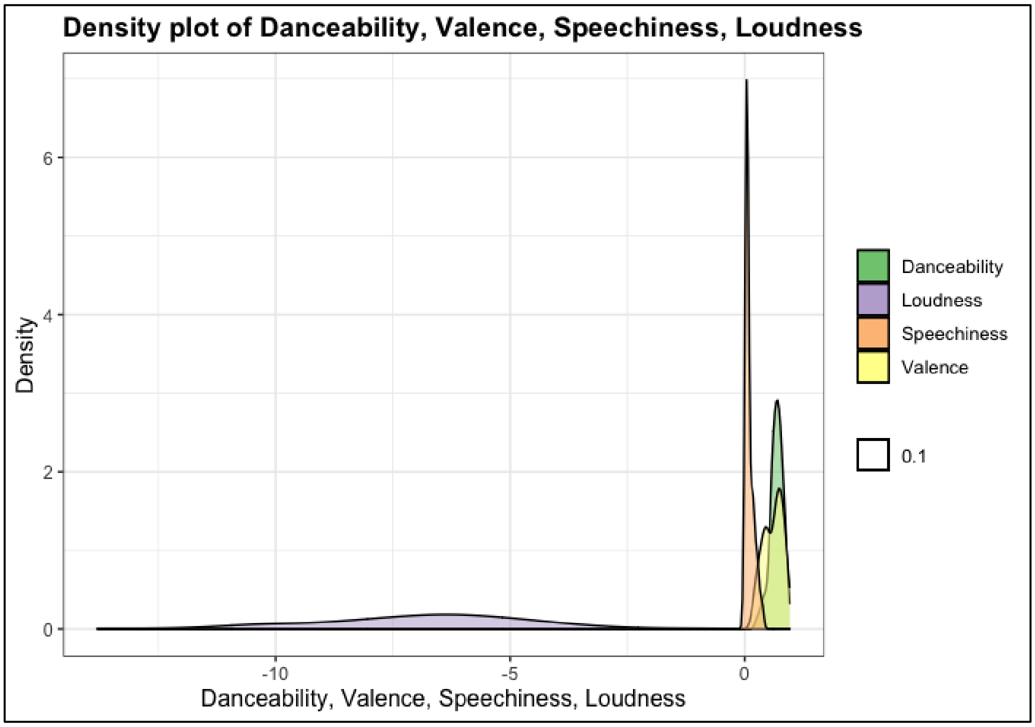


- We observe that Danceability, Loudness, Speechiness, and Valence are positively correlated.
- We see that Danceability and Valence are highly correlated, which suggests that they are Happy songs which make people Dance, considering that Valence measures the positivity of a sound track and Danceability describes the how suitable the sound track is for dancing.
- We also see that Speechiness and Loudness are positively correlated too.

Density of the correlated variables:

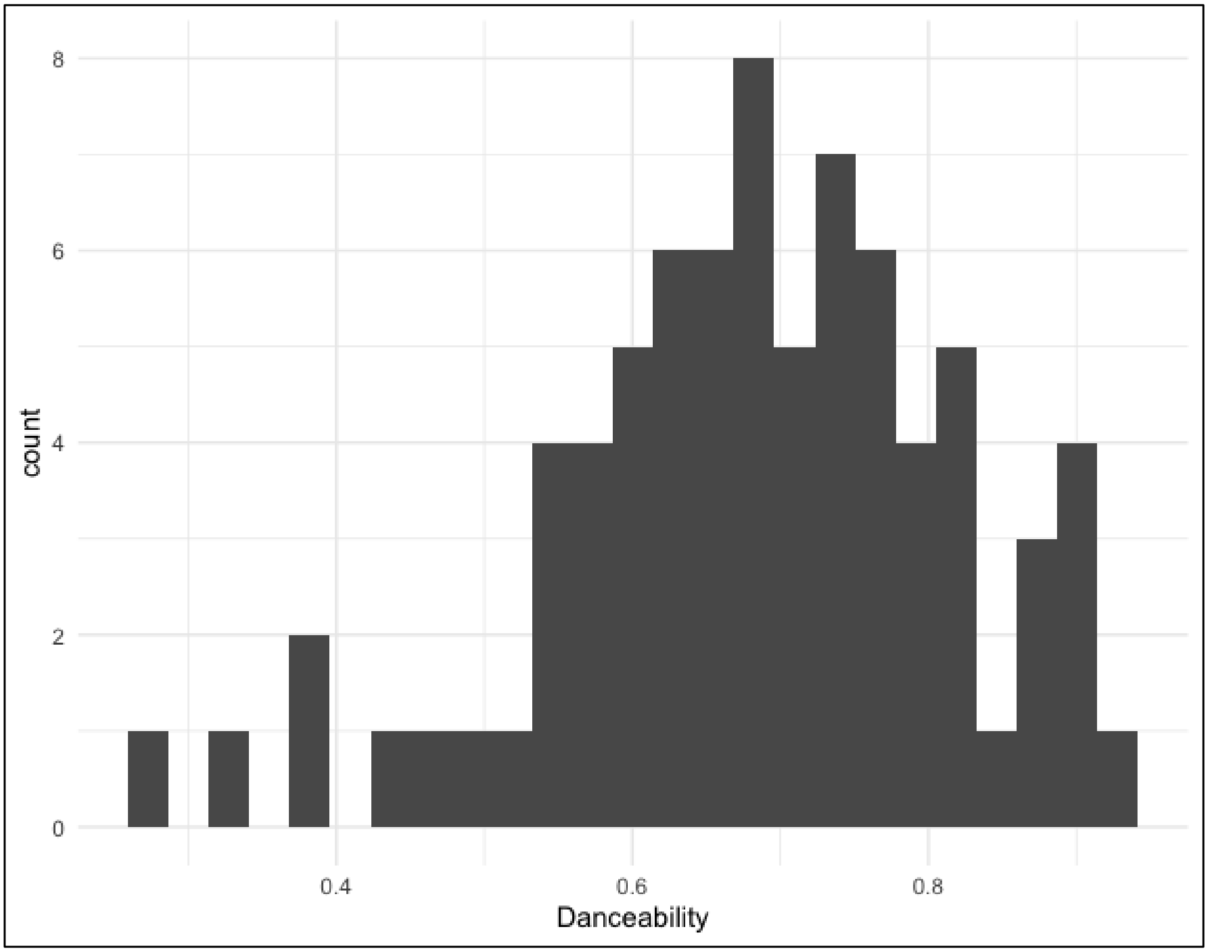
This will allow us to see how these variables are distributed over the songs in the playlist.

```
correlated_density <- ggplot(Spotify_Telugu) +
  geom_density(aes(Danceability, fill = "Danceability", alpha =
0.1)) +
  geom_density(aes(Valence, fill = "Valence", alpha = 0.1)) +
  geom_density(aes(Loudness, fill = "Loudness", alpha = 0.1)) +
  geom_density(aes(Speechiness, fill = "Speechiness", alpha =
0.1)) +
  scale_x_continuous(name = "Danceability, Valence, Speechiness,
Loudness") +
  scale_y_continuous(name = "Density") +
  ggtitle("Density plot of Danceability, Valence, Speechiness,
Loudness") +
  theme_bw() +
  theme(plot.title = element_text(size = 14, face = "bold"),
        text = element_text(size = 12)) +
  theme(legend.title = element_blank()) +
  scale_fill_brewer(palette="Accent")
correlated_density
```



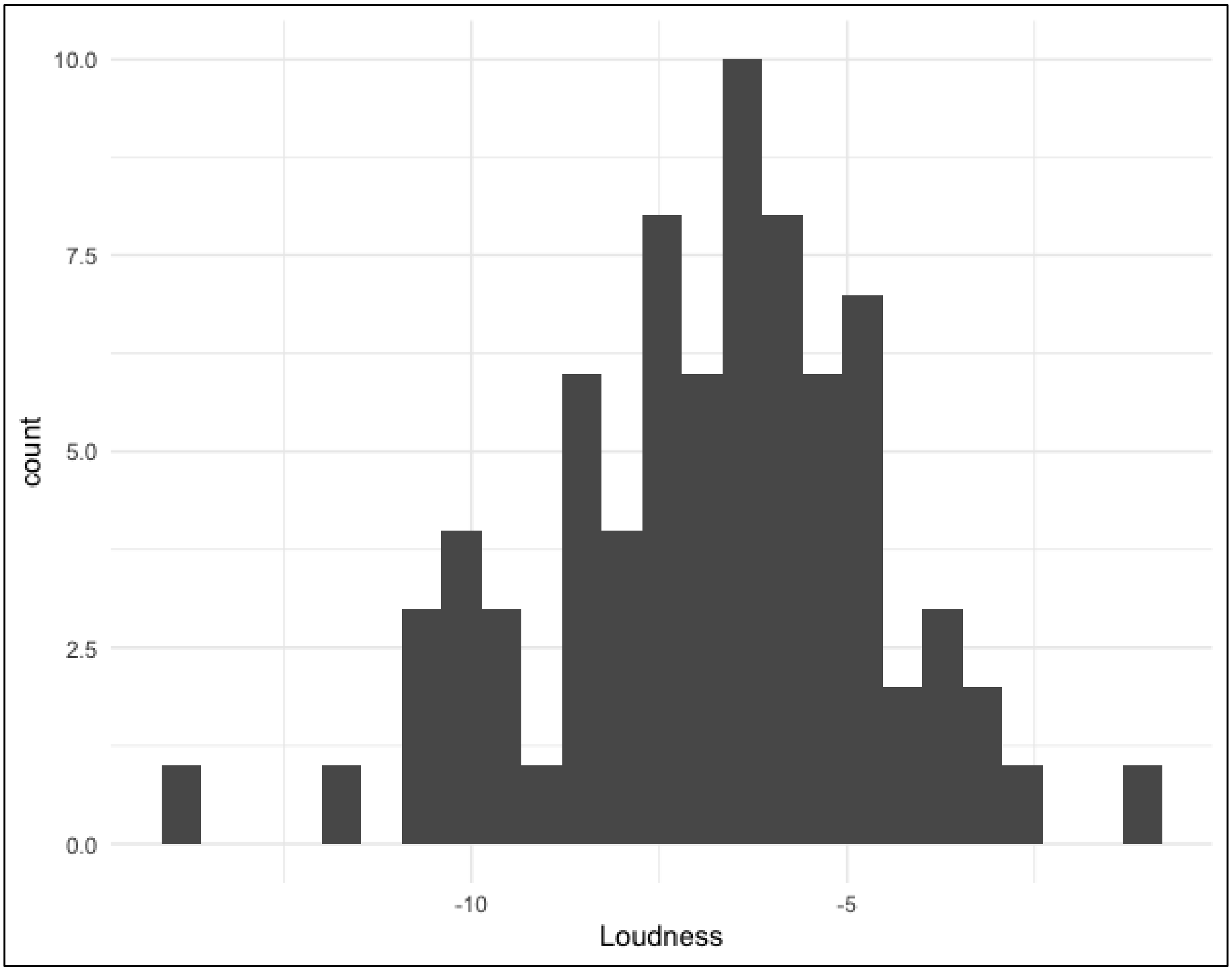
We visualize the positively correlated variables of Danceability, Loudness, Speechiness and Valence below.

```
#Data Visualization

ggplot(Spotify_Telugu, aes(x = Danceability)) +
geom_histogram(bins = 25) + theme_minimal()
```



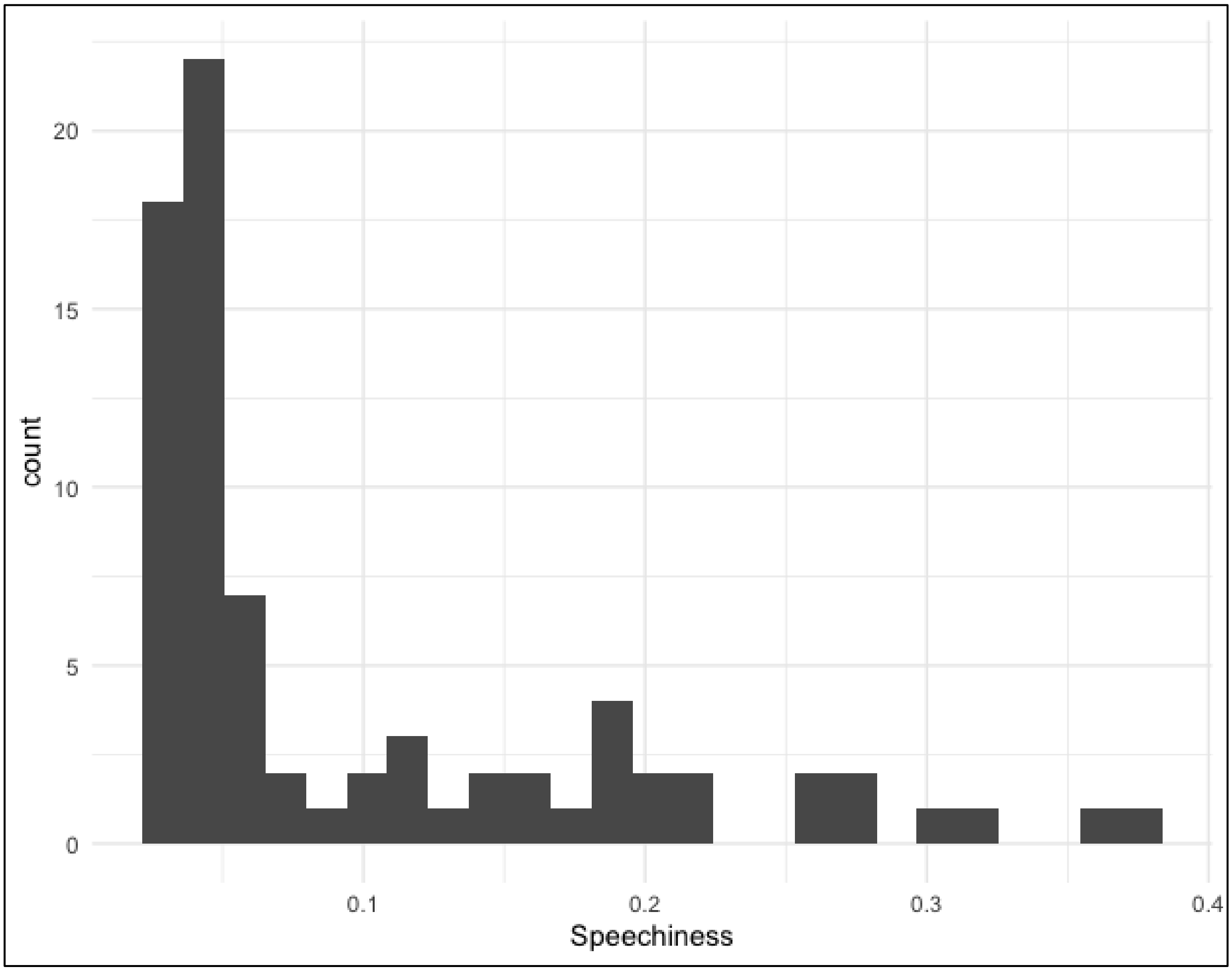
We see that there are many tracks in the playlist that have high Danceability, which might suggest that Telugu population likes tracks with Danceability in the year 2019.

```
#Loudness

ggplot(Spotify_Telugu, aes(x = Loudness)) + geom_histogram(bins
= 25) + theme_minimal()
```



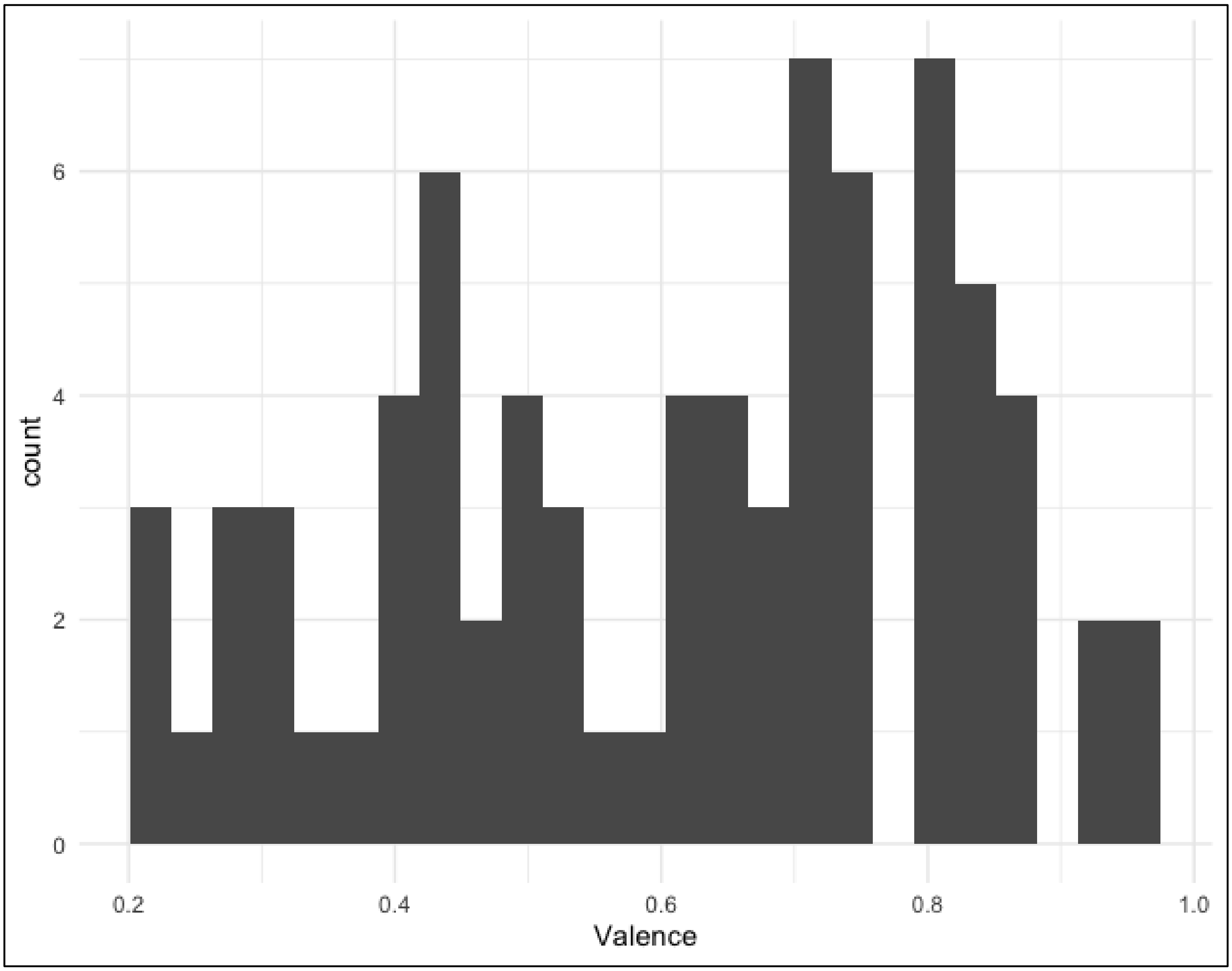
We see that many tracks on this playlist have high Loudness, which in turn might suggest that the Telugu population like tracks with high Loudness in the year 2019.

```
#Speechiness
ggplot(Spotify_Telugu, aes(x = Speechiness)) +
geom_histogram(bins = 25) + theme_minimal()
```



We see that a high number of songs have less Speechiness in their tracks, which might suggest that Telugu population do not like songs in the Rap genre in the year 2019.

```
#Valence
ggplot(Spotify_Telugu, aes(x = Valence)) + geom_histogram(bins =
25) + theme_minimal()
```



We see that Telugu population like happy songs more than sad songs in the year 2019.

Popular Artists of Telugu in the year 2019:

We derive this by calculating the number of times Artist appears in this playlist.

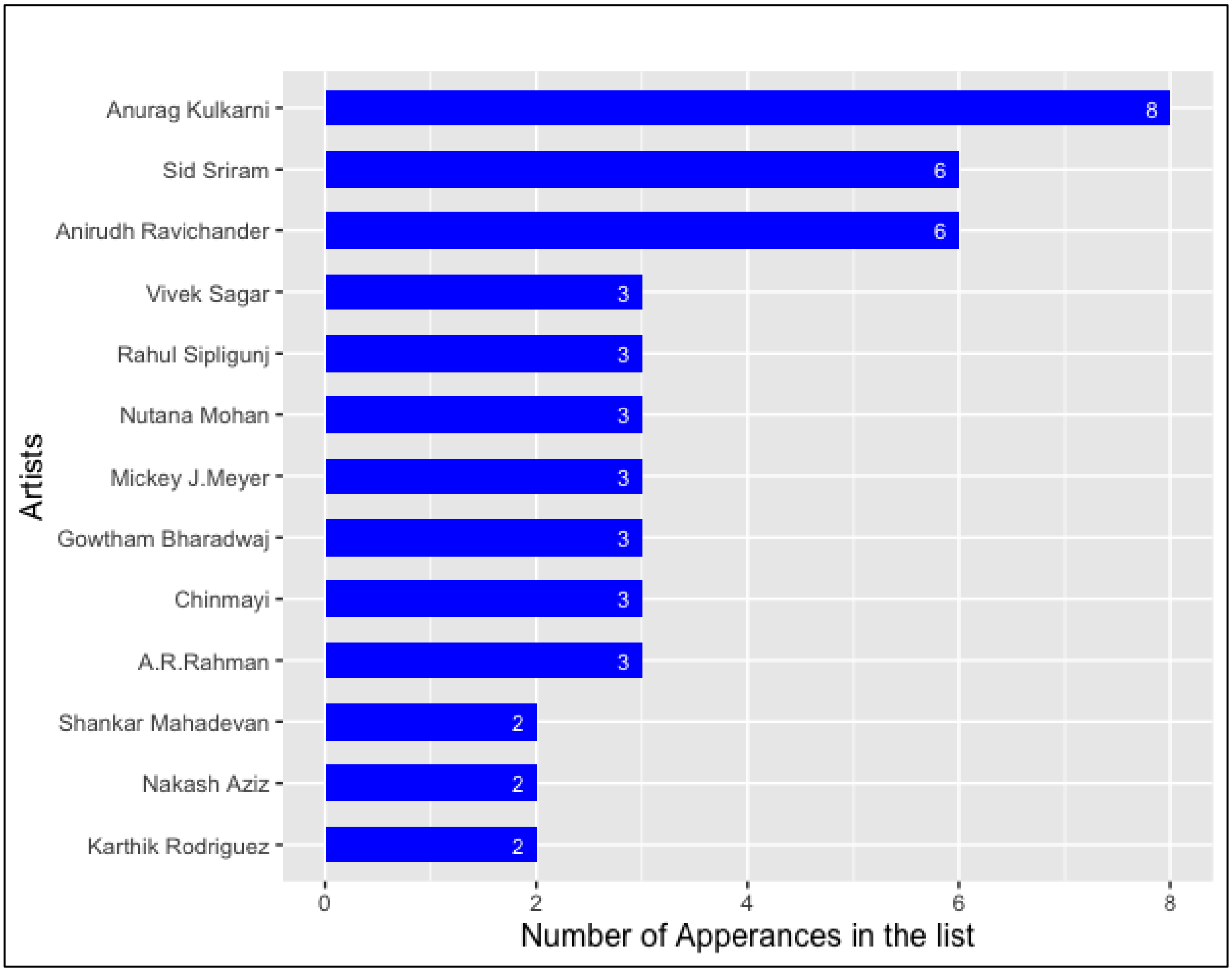
Code:

```
Top_Artists <- Spotify_Telugu %>%
  group_by(Artists) %>%
  summarise(n_apperance = n()) %>%
  filter(n_apperance > 1) %>%
  arrange(desc(n_apperance))
Top_Artists$Artists <- factor(Top_Artists$Artists, levels =
Top_Artists$Artists [order(Top_Artists$n_apperance)])
head(Top_Artists,10)
```

```
Artists             n_apperance
   <fct>                    <int>
 1 Anurag Kulkarni              8
 2 Anirudh Ravichander          6
 3 Sid Sriram                   6
 4 A.R.Rahman                   3
 5 Chinmayi                     3
 6 Gowtham Bharadwaj            3
 7 Mickey J.Meyer               3
 8 Nutana Mohan                 3
 9 Rahul Sipligunj              3
10 Vivek Sagar                  3
```

Plotting the popular artists

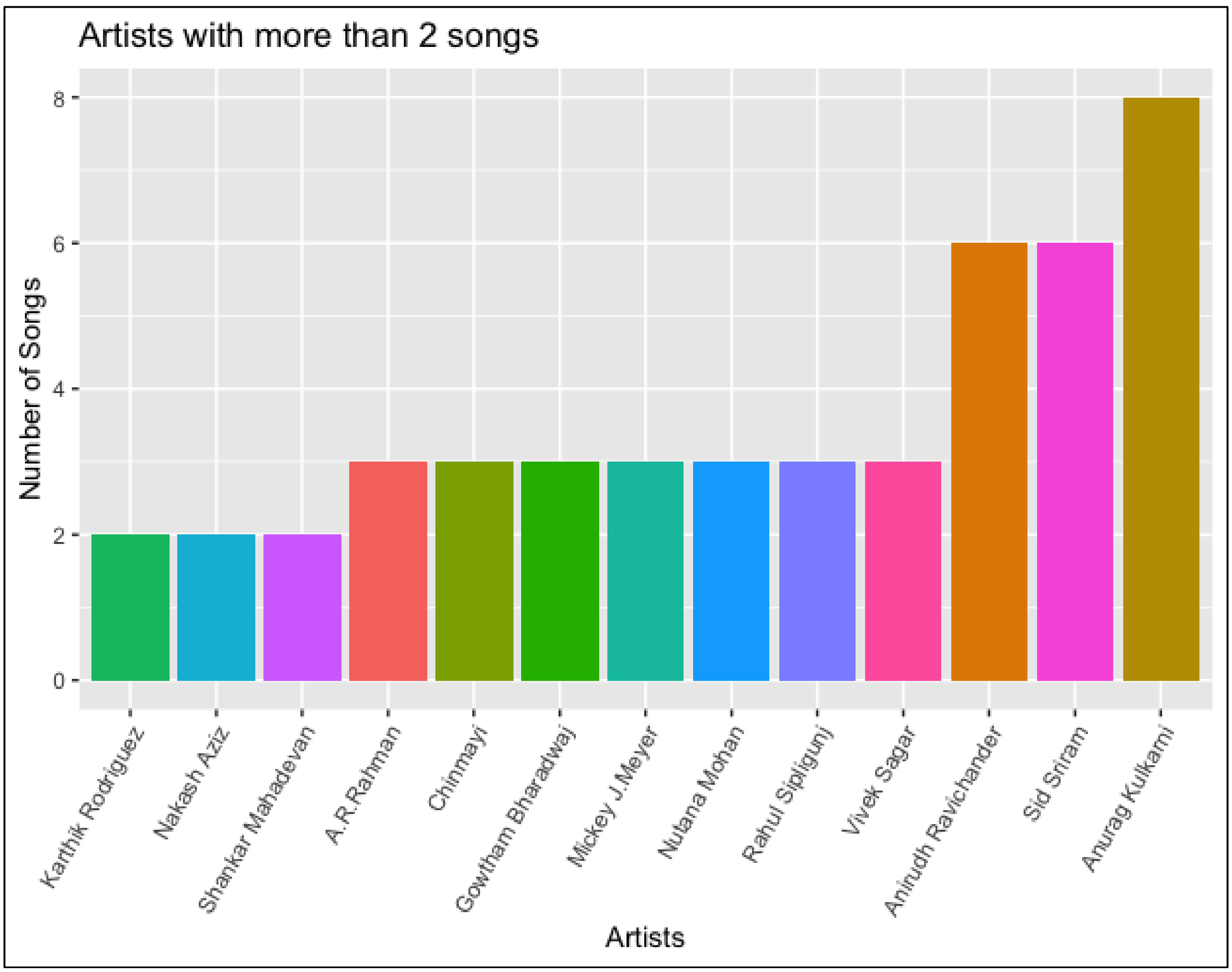
```
ggplot(Top_Artists, aes(x = Artists, y = n_apperance)) +
  geom_bar(stat = "identity", fill = "blue", width = 0.6) +
  labs(title = "Popular Telugu Artists of 2019", x = "Artists",
y = "Number of Apperances in the list") +
  theme(plot.title = element_text(size=15, hjust=-3, face =
"bold"), axis.title = element_text(size=12)) +
  geom_text(aes(label=n_apperance), hjust=2, size = 3, color =
'white') +
  coord_flip()
```



- We see that Anurag Kulkarni, Sid Sriram and Anirudh Ravichander are the popular artists for the year 2019 in Telugu.

- Interestingly Sid Sriram and Anirudh Ravichander both do not speak Telugu as their first language and are from the neighboring state of Tamil Nadu where Tamil is the predominantly the most spoken language yet they are really popular with the Telugu audience.

```
#Artists with more than two songs in the playlist

A1 <- group_by(Spotify_Telugu, Artists)
A2 <- dplyr::summarise(A1, count=n())
A2 <- arrange(A2, desc(count))
A3 <- filter(A2, count>1)
AP1 <- ggplot(A3, aes(x = reorder(Artists, count), y = count)) +
  geom_bar(aes(y = count, fill = Artists), stat = "identity") +
  labs(x = "Artists", y = "Number of Songs",
       title = "Artists with more than 2 songs") +
theme(legend.position = "none", axis.text.x = element_text(angle
= 60, hjust = 1))
AP1
```



Popular Albums in Telugu for the year 2019:

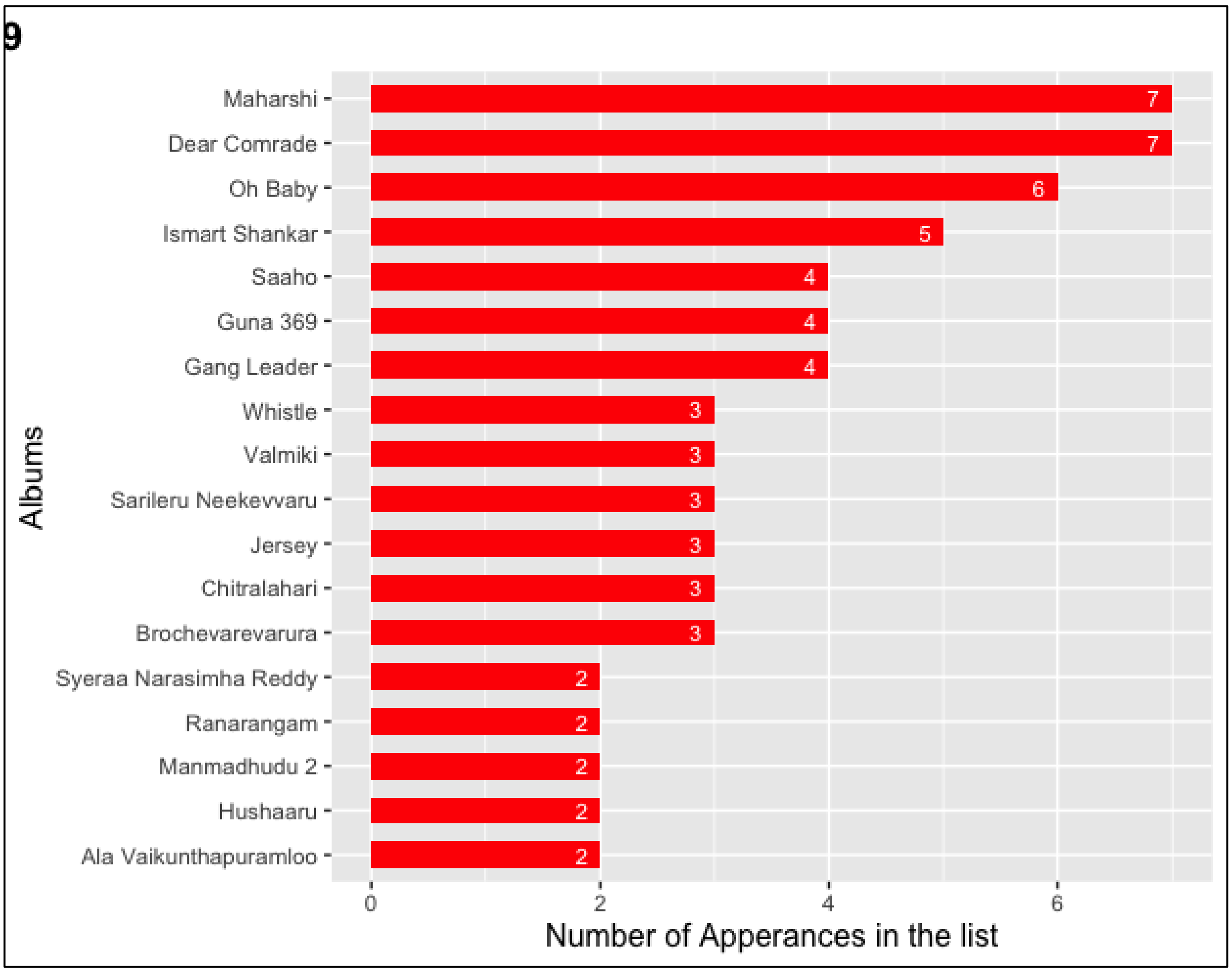
```
Top_Albums <- Spotify_Telugu %>%
  group_by(Album) %>%
  summarise(n_apperance = n()) %>%
  filter(n_apperance > 1) %>%
  arrange(desc(n_apperance))
Top_Albums$Album <- factor(Top_Albums$Album, levels =
Top_Albums$Album [order(Top_Albums$n_apperance)])
head(Top_Albums, 10)
```

```
Album            n_apperance
   <fct>                  <int>
 1 Dear Comrade               7
 2 Maharshi                   7
 3 Oh Baby                    6
 4 Ismart Shankar             5
 5 Gang Leader                4
 6 Guna 369                   4
 7 Saaho                      4
 8 Brochevarevarura           3
 9 Chitralahari               3
10 Jersey                     3
```

- We see that the popular Albums are Maharshi, Dear Comrade, and Oh Baby.
- It is interesting to understand that Indian Music Industry and Movie industry are extremely inter dependent on each other. Considering All the Albums in the playlists are sound tracks of movies they are part of.
- Maharshi, Dear Comrade and Oh Baby are all Telugu Movies in 2019.

Plotting the Popular Albums of Telugu in 2019

```
ggplot(Top_Albums, aes(x = Album, y = n_apperance)) +
  geom_bar(stat = "identity" , fill = "red", width = 0.6) +
  labs(title = "Popular Telugu Albums of 2019", x = "Albums", y
= "Number of Apperances in the list") +
  theme(plot.title = element_text(size=15, hjust=-3, face =
"bold"), axis.title = element_text(size=12)) +
  geom_text(aes(label=n_apperance), hjust=2, size = 3, color =
'white') +
  coord_flip()
```



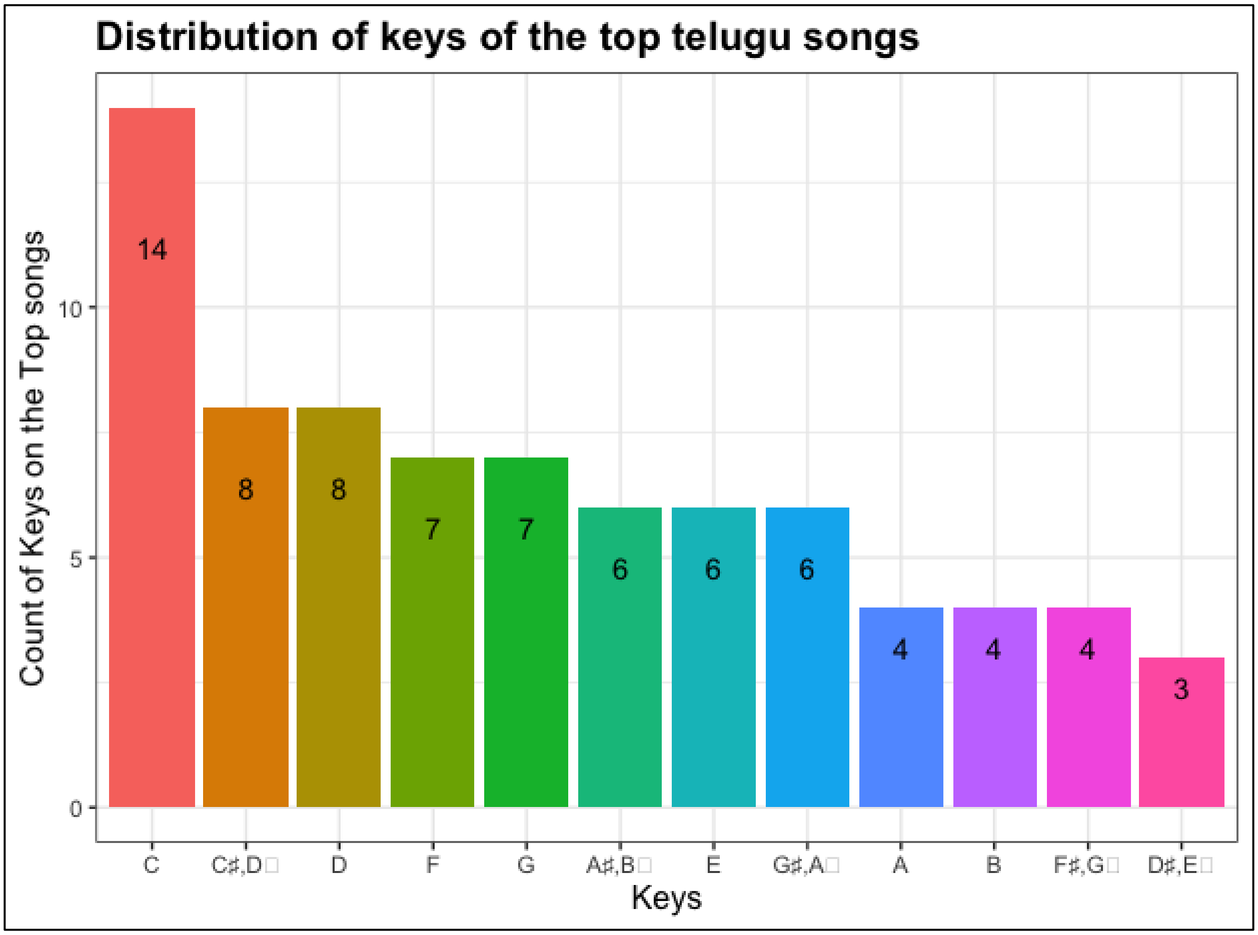
We consolidate all the numerical data of the Spotify Telugu dataset into a separate data frame for easier analysis.

```
#Consolidating all the numerical values of features
Spotify_Telugu_num_norm <- sapply(Spotify_Telugu_num, scale)
summary(Spotify_Telugu_num_norm)
```

Common keys among the tracks on the playlist:

In order to derive this, we need to convert the integer value of Key originally provided by the Spotify API into their assigned keys.

```
#Common keys among the songs
Spotify_Telugu$Key <- as.character(Spotify_Telugu$Key)
Spotify_Telugu$Key <- revalue(Spotify_Telugu$Key, c("0" =
"C","1" = "C♯,D♭", "2" = "D", "3" = "D♯,E♭","4" = "E", "5" = "F",
"6" = "F♯,G♭", "7" = "G", "8" = "G♯,A♭", "9" = "A", "10" =
"A♯,B♭", "11" = "B"))
song_keys <- Spotify_Telugu %>%
  group_by(Key) %>%
  summarise(n_key = n()) %>%
  arrange(desc(n_key))
song_keys$Key <- factor(song_keys$Key, levels =
song_keys$Key[order(song_keys$n_key)])

#Plot the keys
ggplot(song_keys, aes(x = reorder(Key, -n_key), y = n_key, fill
= reorder(Key, -n_key))) +
  geom_bar(stat = "identity") +
  labs(title = "Distribution of keys of the top telugu songs", x
= "Keys", y = "Count of Keys on the Top songs") +
  geom_text(aes(label=n_key), position =
position_stack(vjust=0.8)) +
  theme_bw() +
  theme(plot.title = element_text(size = 15, face = "bold"),
axis.title = element_text(size=12)) +
  theme(legend.position = "none")
```



We see that the most common keys are C, C#D and D.

Logistic Regression

We build a logistic regression model that predicts the artists given the features of songs, using multiple independent variables such as Danceability, Energy, Valance and such.

In order to run a logistic regression in R we use ‘glm’ – generalized linear model function whose syntax is of the function similar to a linear regression. But with specifying ‘binomial’ for the family argument we will be able to treat glm function as a dependent variable as binary.

We make sure that the outcome (Artist) is a factor.

```
Spotify_Telugu$Artists <- as.factor(Spotify_Telugu$Artists)
Spotify_Telugu <- Spotify_Telugu %>%
  select(Artists, Danceability, Valence, Speechiness, Loudness)

#Splitting the data
mp_siz = floor(0.80*nrow(Spotify_Telugu))
set.seed(123)
train_data = sample(seq_len(nrow(Spotify_Telugu)), size =
smp_siz)
train = Spotify_Telugu[train_data,]
test = Spotify_Telugu[-train_data,]
test$Artists <- as.factor(test$Artists)
train$Artists <- as.factor(train$Artists)
```

Now we train a logistic regression model on the training data and analyze the output.

```
#training logistic regression model
Logit_Spotify <- glm(Artists ~ Danceability + Valence , data =
train, family = "binomial")

#inspect
summary(logit_Spotify)
```

```
Call:
glm(formula = Artists ~ Danceability + Valence, family =
"binomial",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6902   0.1518   0.2161   0.2958   0.5645

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)     3.274      3.956   0.828    0.408
Danceability   -3.877      7.069  -0.548    0.583
Valence         5.122      4.503   1.137    0.255

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 17.605  on 60  degrees of freedom
Residual deviance: 16.208  on 58  degrees of freedom
AIC: 22.208

Number of Fisher Scoring iterations: 7
```

The coefficients of the model are used to inspect the strength of association between variables.

We use logistic regression to predict the Artist on testing data.

```
attach(test)
pdata <- predict(logit_Spotify, newdata = test, type =
"response")
pdata = ifelse(pdata > .5, 1, 0)
table(pdata, test$Artists)
```

```
pdata A.R.Rahman Anirudh Ravichander Anurag Kulkarni Benny Dayal
Chaitan Bharadwaj
    1          1                   1               2           0
0

pdata Chinmayi Darshan Raval David Simon Devi Sri Prasad Dhibu
Ninan Thomas
```

```
    1        0             0           0               0
1

pdata Gowtham Bharadwaj Guru Randhawa Haricharan Kala Bhairava
Karthik Rodriguez
    1                 1             0          1             0
0

pdata Keerthana Sharma Madhu Priya Mickey J.Meyer Mohana
Bhogaraju Nakash Aziz
    1                0           1              0
0           0

pdata Neeti Mohan Nikhita Gandhi Nutana Mohan Rahul Sipligunj
Ranina Reddy
    1           0              0            1               0
0

pdata S.P.Balasubrahmanyam Sarath Santosh Sathyaprakash Shankar
Mahadevan
    1                    0              1             1
0

pdata Shashaa Tirupati Shweta Mohan Sid Sriram Sreerama Chandra
Sudharshan Ashok
    1                0            0          2                0
0

pdata Sunidhi Chauhan Suresh Bobbili Tushar Joshi Vedala
Hemachandra Vijay Devarakonda
    1               0              0            1
1                 0

pdata Vijay Prakash Vijay Yesudas Vivek Sagar Yazin Nizar
    1             0             0           1           0
```

We have built a logistic regression model that evaluates how the predictors of Danceability and Valence contribute to the probability of a song being from each artist in the playlist. We then used this same model to predict the Artist for songs in the testing set.

RMSE Analysis

Basic Mean Model

This model uses the mean of each variable to predict their respective Danceability, Loudness, Speechiness, and Valence for all sound tracks. The model assumes that all differences are due to a random error.

```
#Average model Danceability
mu <- mean(train$Danceability)
mu

0.6730328

#Danceability
basic_rmse_dancebility <- RMSE(test$Danceability, mu)
basic_rmse_dancebility

0.09887948

#Loudness
basic_rmse_loudness <- RMSE(test$Loudness, mu)
basic_rmse_loudness

7.832492

#Valence
basic_rmse_valence <- RMSE(test$Valence, mu)
basic_rmse_valence

0.2104865

#Speechiness
basic_rmse_speechiness <- RMSE(test$Speechiness, mu)
basic_rmse_speechiness

0.5998211
```

RMSE Results:

Danceability	0.09887948
Loudness	7.832492
Valence	0.2104865
Speechiness	0.5998211

Conclusion:

In this project we have successfully implemented learning concepts from all the previous courses in the Data Science Professional Certificate Program. We have created visualizations and analyzed the data for any insights. We have also built a logistic regression model and also developed a Machine Learning algorithm to predict the artists based on variables of audio features of Danceability and Valence.