

# Music, Emotions

*Maria Ren*

*5/6/2018*

## Introduction

We often say, “I’m in the mood for a happy tune.” But what exactly makes a song happy? Or sad?

In the twenty-first century, music has become one of the most important parts of our lives. Whether it’s classical, jazz, funk, pop, or country, the demand for music has created a billion dollar music industry worldwide. Music industry revenue in turn became a significant component in the global economy, which also led to fierce competitions across record companies and other music platforms. These companies strive to satisfy the listener’s taste by creating music that fits individual tastes,

Among these music platforms, Spotify is one of the most popular digital music services that provides its users with access to millions of songs. It also gives users the ability to rate music according to their own personal preference and taste.

This project uses real time and historical data extracted from the Spotify API (Developer Portal), and analyzes the different elements of a soundtrack that changes human emotions. The musical data consists of musical features and ratings for the number one songs from “Billboard Hot 100 Chart” from 1960 to 2018.

## Audio Features:

The following audio features are extracted from the Spotify API, and used in the data analytics.

### Danceability:

Describes how a song track is suitable for dancing. Values range from 0 to 1, with 0 being the least danceable, and 1 being the most danceable. Danceability is calculated based on music tempo, rhythm stability, beat strength.

### Energy:

Describes level of music intensity and activity. For example, the most energetic music feels fast and loud (like death metal), while a quiet and peaceful classical music (like Chopin) would score lower on the energy scale. Calculated from loudness, onset rate, tempo, and dynamic ranges. Values range from 0 to 1, with 0 being least energetic, and 1 being most energetic.

### Loudness:

Describes the decibel (dB) level of a score. Values range from -60 to 0. With -60 being the loudest, and 0 being quietest.

### Speechiness:

Describes the amount of spoken words in a track. Values range from 0 to 1, with 0 being music having no words, and 1 being completely spoken song track. (For example, a talk show would have closer to 1, song with rap would have values above 0.66, values between 0.33 and 0.66 would be regular songs with lyrics, and values below 0.33 are most likely classical music that is completely instrumental).

**Acousticness:**

Describes the level of acoustic sound (as opposed to electronic sound) detected from the soundtrack. Values ranges from 0 to 1, with 0 being entirely electronic sound, and 1 being acoustic sound.

**Instrumentalness:**

Describes whether the track contains vocals. ( “Ooh” and “aah” sound are treated as instrumental). Values ranges from 0 to 1, with 0 being the soundtrack having no instrumentation, and 1 being instrumental music. (For example, a rap song would have score closer to 0, and classical symphonic music would have value closer to 1).

**Liveness:**

Describes whether the song track was recorded live or in a studio. Calculated from detection of audience sound in the sound track. Values ranges from 0 to 1, with values closer to 0 indicating music which were recorded in a studio setting, and with values above 0.8 indicating music recorded live.

**Valence:**

Describes the happiness level of a music soundtrack. Values ranges from 0 to 1, with 0 being music tracks that are negative (sad, depressed, and angry), while 1 being music that are positive (happy, exciting and cheerful). Calculated from audience reviews and ratings.

**Tempo:**

Describes the tempo of a track in beats per minute (BPM). Values ranges from 0 to around 200. With values around 50 being very slow, and around 200 being very fast.

**Duration(Duration\_ms):**

Describes the length of the soundtrack. Calculated in milliseconds.

## Data Cleaning

- Obtain Spotify access token
- Set Spotify client ID and client secret
- Obtain real time data from Spotify

```
Sys.setenv(SPOTIFY_CLIENT_ID = '0da3921447fa4e3099913296e99b43d6')
Sys.setenv(SPOTIFY_CLIENT_SECRET = '216f8762ba494bb7a626becb67d6a0ab')
access_token <- get_spotify_access_token()
```

- Obtain historical data set: Billboards Hot 100 list (containing top 100 most popular songs from 1960 to 2016. This dataset comes from the R library “billboard”, and the data was scraped from
- [https://en.wikipedia.org/wiki/Billboard\\_Year-End\\_Hot\\_100\\_singles\\_of\\_](https://en.wikipedia.org/wiki/Billboard_Year-End_Hot_100_singles_of_)
- [https://en.wikipedia.org/wiki/Billboard\\_Year-End\\_Hot\\_100\\_singles\\_of\\_1960](https://en.wikipedia.org/wiki/Billboard_Year-End_Hot_100_singles_of_1960)

no	track_name	artist	year
1	Theme from A Summer Place	Percy Faith	1960
2	He'll Have to Go	Jim Reeves	1960
3	Cathy's Clown	The Everly Brothers	1960
4	Running Bear	Johnny Preston	1960
5	Teen Angel	Mark Dinning	1960
6	I'm Sorry	Brenda Lee	1960
7	It's Now or Never	Elvis Presley	1960
8	Handy Man	Jimmy Jones	1960
9	Stuck on You	Elvis Presley	1960
10	The Twist	Chubby Checker	1960
11	Everybody's Somebody's Fool	Connie Francis	1960
12	Wild One	Bobby Rydell	1960
13	Greenfields	The Brothers Four	1960
14	What in the World's Come Over You	Jack Scott	1960
15	El Paso	Marty Robbins	1960

**Note:**

(The table shows the first 15 rows of the dataset)

- Filter out top song from each year from 1960-2016
- Form a complete list of top song tracks from 1960 to 2018

track_name	artist	year
Theme from A Summer Place	Percy Faith	1960
Tossin' and Turnin'	Bobby Lewis	1961
Stranger on the Shore	Acker Bilk	1962
Sugar Shack	The Fireballs	1963
I Want to Hold Your Hand	The Beatles	1964
Wooly Bully	Sam the Sham and the Pharaohs	1965
Ballad of the Green Berets	SSgt. Barry Sadler	1966
To Sir With Love	Lulu	1967
Hey Jude	The Beatles	1968
Sugar, Sugar	The Archies	1969
Bridge over Troubled Water	Simon & Garfunkel	1970
Joy to the World	Three Dog Night	1971
The First Time Ever I Saw Your Face	Roberta Flack	1972
Tie a Yellow Ribbon Round the Ole Oak Tree	Tony Orlando and Dawn	1973
The Way We Were	Barbra Streisand	1974

**Note:**

(The table shows the first 15 rows of the dataset)

- Clean up and tidying data
- Include track features into the table
- Still need data for the following song tracks

track_name
Sugar Shack
Wooly Bully
Ballad of the Green Berets
Sugar, Sugar
Tie a Yellow Ribbon Round the Ole Oak Tree
My Sharona
Careless Whisper
That's What Friends Are For
Gangsta's Paradise
Macarena (Bayside Boys Mix)
Yeah!
Low
Somebody That I Used to Know
Thrift Shop
Uptown Funk
Love Yourself
Shape of You
Perfect

- Pull data from Spotify API
- Write functions `get_track` and `get_features`
- `get_track` is a function used to obtain `track_uri` from Spotify
- `get_features` is a function used to obtain track features by matching `track_uri` with track needed for analysis

```
get_track <- function(artist) {
  artists <- get_artists(artist)
  albums <- get_albums(artists$artist_uri[1])
  return(get_album_tracks(albums))
}

get_features <- function(artist) {
  artists <- get_artists(artist)
  albums <- get_albums(artists$artist_uri[1])
  tracks <- get_album_tracks(albums)
  artists_audio_features <- get_track_audio_features(tracks)
  return(artists_audio_features)
}
```

- Obtaining audio features for the incomplete song tracks
- Warning: the `get_track` and `get_features` functions works, but can be a little slow. Also, sometimes they don't work on the first try- you might have to rerun the specific code again. (They will work for sure, the Spotify API access code can be a little slow and weird.) But I already saved the correct finalized tidy dataset named `Billboard_Top_1.csv`, and it is saved in the same directory as both the github and the shiny app folder. So you can skip to the section labeled Analysis of dataset of the document if runtime of the code takes too long.

```
# track name: Sugar Shack
# artist: The Fireballs
# year: 1963

song_list <- get_track("The Fireballs")
feature_list <- get_features("The Fireballs")
```

```

song_data <- as.tibble(song_list) %>% filter(track_name=="Sugar Shack")
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data)
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Sugar Shack"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

```

```

# track name: Wooly Bully
# artist: Sam the Sham and the Pharaohs
# year: 1965

```

```

song_list <- get_track("Sam the Sham and the Pharaohs")
feature_list <- get_features("Sam the Sham and the Pharaohs")
song_data <- as.tibble(song_list) %>% filter(track_name=="Wooly Bully")
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Wooly Bully"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

```

```

# track name: Ballad of the Green Berets
# artist: SSgt. Barry Sadler
# year: 1966

```

```

song_list <- get_track("Barry Sadler")
feature_list <- get_features("Barry Sadler")
song_data <- as.tibble(song_list) %>% filter(track_name=="The Ballad Of The Green Berets")
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data)
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Ballad of the Green Berets"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: Sugar, Sugar
# artist: The Archies
# year: 1969

song_list <- get_track("The Archies")
feature_list <- get_features("The Archies")
song_data <- as.tibble(song_list) %>% filter(track_name=="Sugar, Sugar")
song_data
# the song does not exist in the Spotify API with the band name "The Archies"
# But Spotify has features from Ron Dante (the lead singer)
song_list <- get_track("Ron Dante")
feature_list <- get_features("Ron Dante")
song_data <- as.tibble(song_list) %>% filter(track_name=="Sugar Sugar") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data)
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Sugar, Sugar"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

```

```

}

# track name: Tie a Yellow Ribbon Round the Ole Oak Tree
# artist: Tony Orlando and Dawn
# year: 1973

song_list <- get_track("Tony Orlando and Dawn")
feature_list <- get_features("Tony Orlando and Dawn")
song_data <- as.tibble(song_list) %>% filter(track_name=="Tie a Yellow Ribbon Round the Ole Oak Tree")
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data, feature_data)
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Tie a Yellow Ribbon Round the Ole Oak Tree"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: My Sharona
# artist: The Knack
# year: 1979

song_list <- get_track("The Knack")
feature_list <- get_features("The Knack")
song_data <- as.tibble(song_list) %>% filter(track_name=="My Sharona")
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data, feature_data[1,]) %>% slice(1)
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "My Sharona"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

```

```

# track name: Careless Whisper
# artist: Wham! featuring George Michael
# year: 1985

song_list <- get_track("Wham")
feature_list <- get_features("Wham")
song_data <- as.tibble(song_list) %>% filter(track_name=="Careless Whisper") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Careless Whisper"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: That's What Friends Are For
# artist: Dionne and Friends (Dionne Warwick, Gladys Knight, Elton John)
# year: 1986

song_list <- get_track("Dionne Warwick")
feature_list <- get_features("Dionne Warwick")
song_data <- as.tibble(song_list) %>% filter(track_name=="That's What Friends Are For") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data)
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "That's What Friends Are For"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

```



```

# track name: Gangsta's Paradise
# artist: Coolio featuring L.V.
# year: 1995

song_list <- get_track("Coolio")
feature_list <- get_features("Coolio")
song_data <- as.tibble(song_list) %>% filter(track_name=="Gangsta's Paradise") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Gangsta's Paradise"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

```

```

# track name: Macarena (Bayside Boys Mix)
# artist: Los del Río
# year: 1996

song_data <- as.tibble(spotify_track_data) %>%
  filter(track_name=="Macarena - Bayside Boys Remix") %>%
  slice(1)
get_feature_data <- song_data
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Macarena (Bayside Boys Mix)") {
    list$track_uri[i] <- str_sub(get_feature_data$track_id)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

```

```

# track name: Yeah!
# artist: Usher featuring Lil Jon and Ludacris
# year: 2004

song_list <- get_track("Usher")
feature_list <- get_features("Usher")
song_data <- as.tibble(song_list) %>% filter(track_name=="Yeah!") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Yeah!"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: Low
# artist: Flo Rida featuring T-Pain
# year: 2008

song_list <- get_track("Flo Rida")
feature_list <- get_features("Flo Rida")
song_data <- as.tibble(song_list) %>% filter(track_name=="Low (feat T-Pain)")
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri) %>% slice(1)
get_feature_data <- cbind(song_data,feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Low"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
  }
}

```

```

    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: Somebody That I Used to Know
# artist: Gotye featuring Kimbra
# year: 2012

song_list <- get_track("Gotye")
feature_list <- get_features("Gotye")
song_data <- as.tibble(song_list) %>% filter(track_name=="Somebody That I Used To Know") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data, feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Somebody That I Used to Know"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: Thrift Shop
# artist: Macklemore and Ryan Lewis featuring Wanz
# year: 2015

song_list <- get_track("Ryan Lewis")
feature_list <- get_features("Ryan Lewis")
song_data <- as.tibble(song_list) %>% filter(track_name=="Thrift Shop (feat. Wanz)") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data, feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Thrift Shop"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
  }
}

```

```

    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: Uptown Funk
# artist: Mark Ronson featuring Bruno Mars
# year: 2015

song_list <- get_track("Mark Ronson")
feature_list <- get_features("Mark Ronson")
song_data <- as.tibble(song_list) %>% filter(track_name=="Uptown Funk") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Uptown Funk"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: Love Yourself
# artist: Justin Bieber
# year: 2016

song_list <- get_track("Justin Bieber")
feature_list <- get_features("Justin Bieber")
song_data <- as.tibble(song_list) %>% filter(track_name=="Love Yourself") %>% slice(1)
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data,feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Love Yourself"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
  }
}

```

```

    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: Shape of You
# artist: Ed Sheeran
# year: 2017

song_list <- get_track("Ed Sheeran")
feature_list <- get_features("Ed Sheeran")
song_data <- as.tibble(song_list) %>% filter(track_name=="Shape of You")
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data, feature_data[1,])
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Shape of You"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
    list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
    list$liveness[i] <- str_sub(get_feature_data$liveness)
    list$valence[i] <- str_sub(get_feature_data$valence)
    list$tempo[i] <- str_sub(get_feature_data$tempo)
    list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
  }
}

# track name: Perfect
# artist: Ed Sheeran
# year: 2018

song_list <- get_track("Ed Sheeran")
feature_list <- get_features("Ed Sheeran")
song_data <- as.tibble(song_list) %>% filter(track_name=="Perfect")
feature_data <- as.tibble(feature_list) %>% filter(track_uri==song_data$track_uri)
get_feature_data <- cbind(song_data, feature_data)
for (i in 1:nrow(list)) {
  if (list$track_name[i] == "Perfect"){
    list$track_uri[i] <- str_sub(get_feature_data$track_uri)
    list$danceability[i] <- str_sub(get_feature_data$danceability)
    list$energy[i] <- str_sub(get_feature_data$energy)
    list$loudness[i] <- str_sub(get_feature_data$loudness)
    list$speechiness[i] <- str_sub(get_feature_data$speechiness)
    list$acousticness[i] <- str_sub(get_feature_data$acousticness)
  }
}

```

```

list$instrumentalness[i] <- str_sub(get_feature_data$instrumentalness)
list$liveness[i] <- str_sub(get_feature_data$liveness)
list$valence[i] <- str_sub(get_feature_data$valence)
list$tempo[i] <- str_sub(get_feature_data$tempo)
list$duration_ms[i] <- str_sub(get_feature_data$duration_ms)
}
}

```

## Final complete dataset

- Final dataset Billboard\_Top\_1.csv including all songs and features from top 1 song of the Billboard Hot 100 song list from 1960 to 2018.

```
List <- write.csv(list,"Billboard_Top_1.csv")
```

## Data Analysis

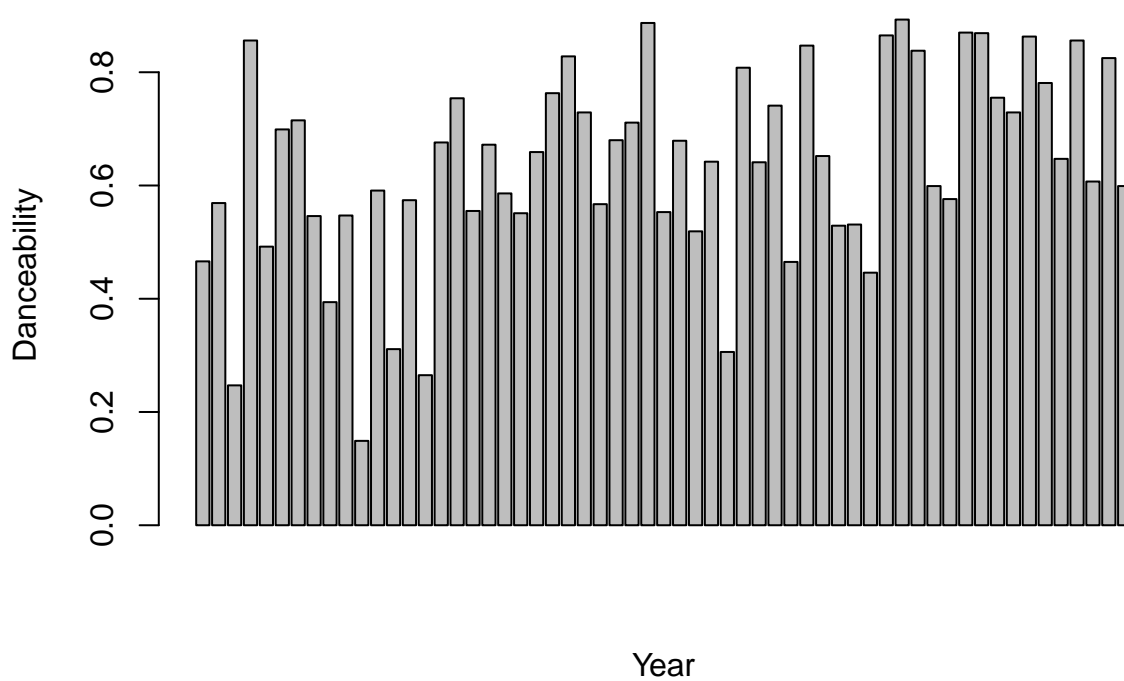
- The final dataset include 10 specific audio features (danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration)

```
## [1] "X"           "track_name"    "artist"
## [4] "year"        "track_uri"     "danceability"
## [7] "energy"      "loudness"      "speechiness"
## [10] "acousticness" "instrumentalness" "liveness"
## [13] "valence"     "tempo"         "duration_ms"

```

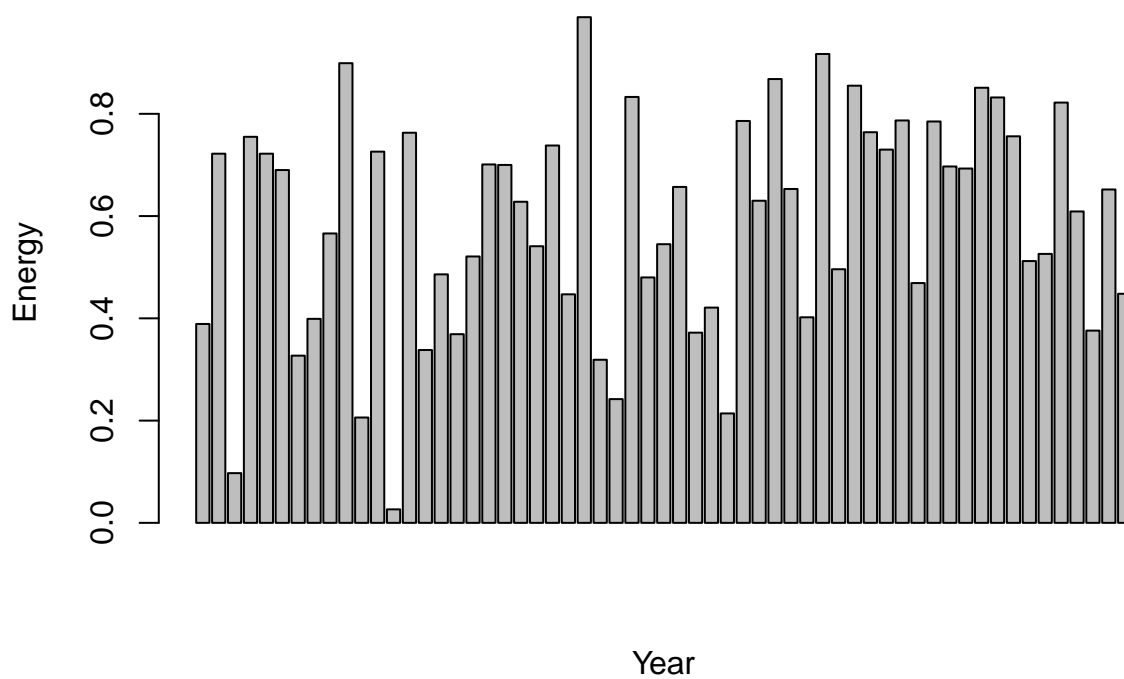
## Barplot and basic statistics of different audio features

### Danceability vs. Year



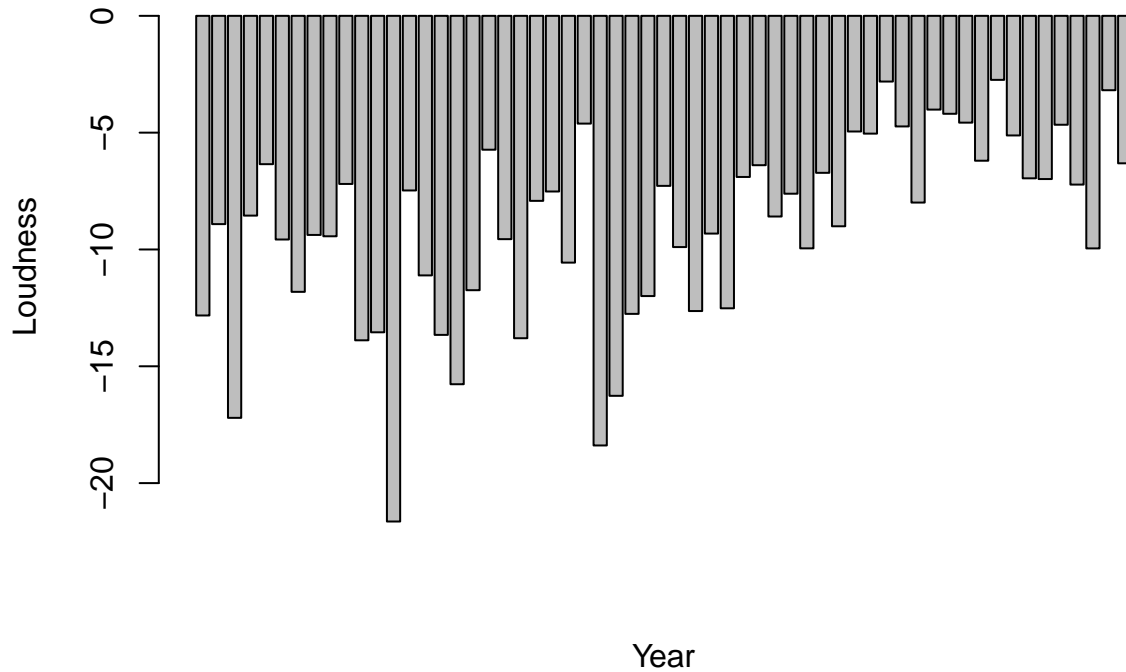
##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	0.1490	0.5490	0.6470	0.6368	0.7590	0.8930

### Energy vs. Year



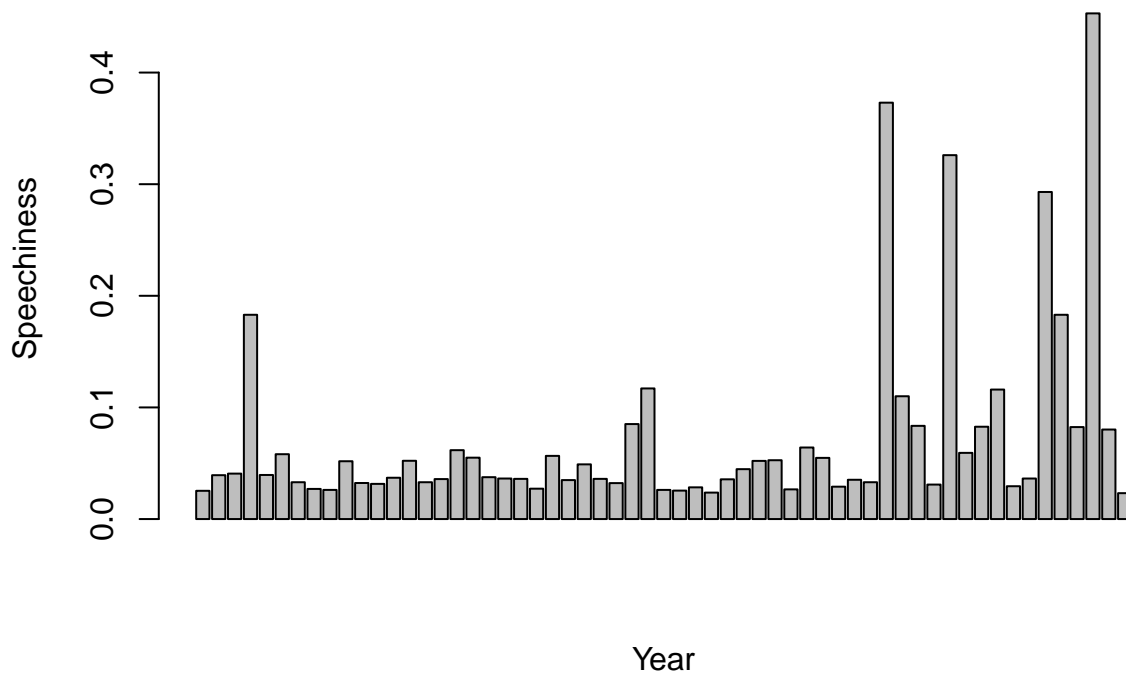
##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	0.0264	0.4340	0.6300	0.5889	0.7555	0.9890

**Loudness vs. Year**



##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	-21.644	-11.778	-8.550	-9.045	-6.330	-2.741

**Speechiness vs. Year**

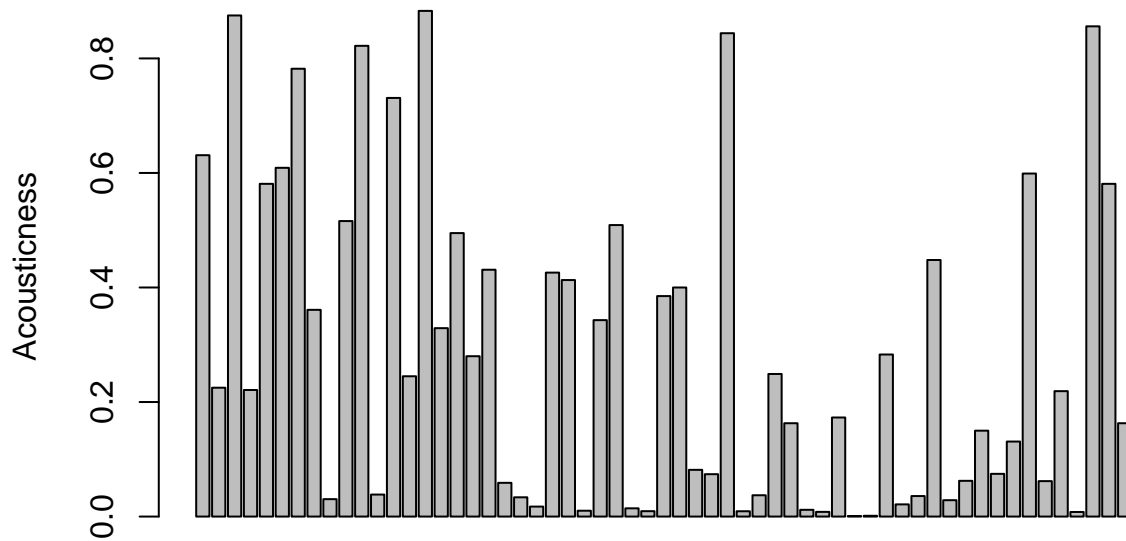


##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
----	------	---------	--------	------	---------	------



```
## 0.02320 0.03225 0.03930 0.07294 0.06290 0.45300
```

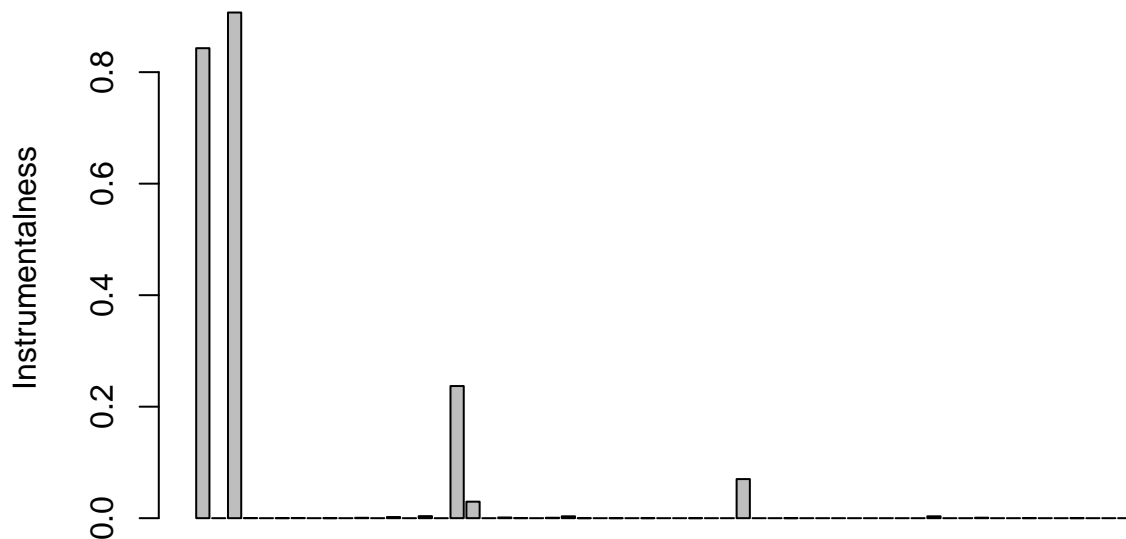
### Acousticness vs. Year



### Year

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## 0.000983 0.036500 0.221000 0.289521 0.471500 0.883000
```

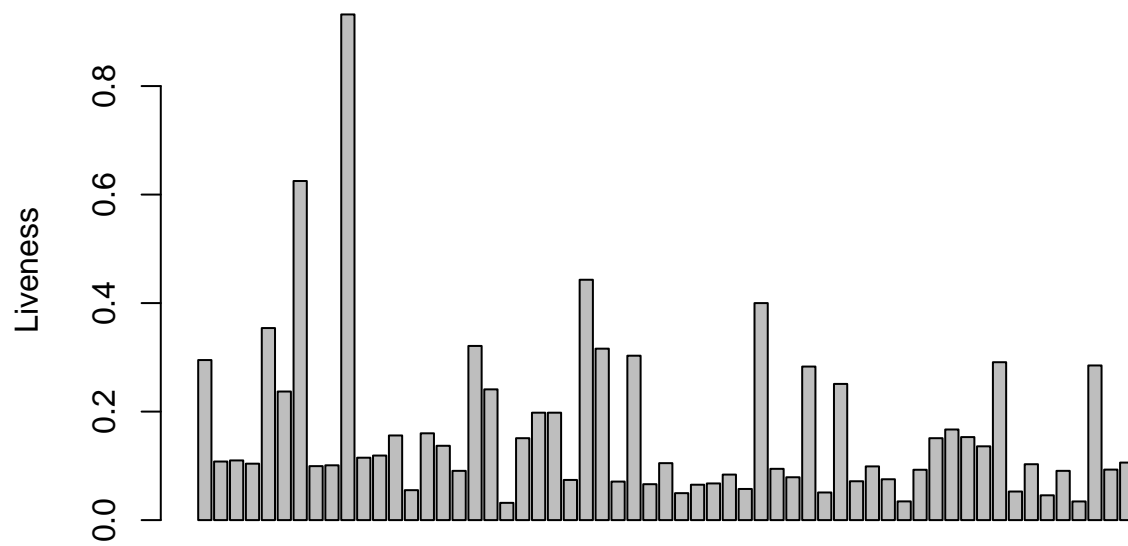
### Instrumentalness vs. Year



### Year

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## 0.0000000 0.0000000 0.0000032 0.0356621 0.0002365 0.9070000
```

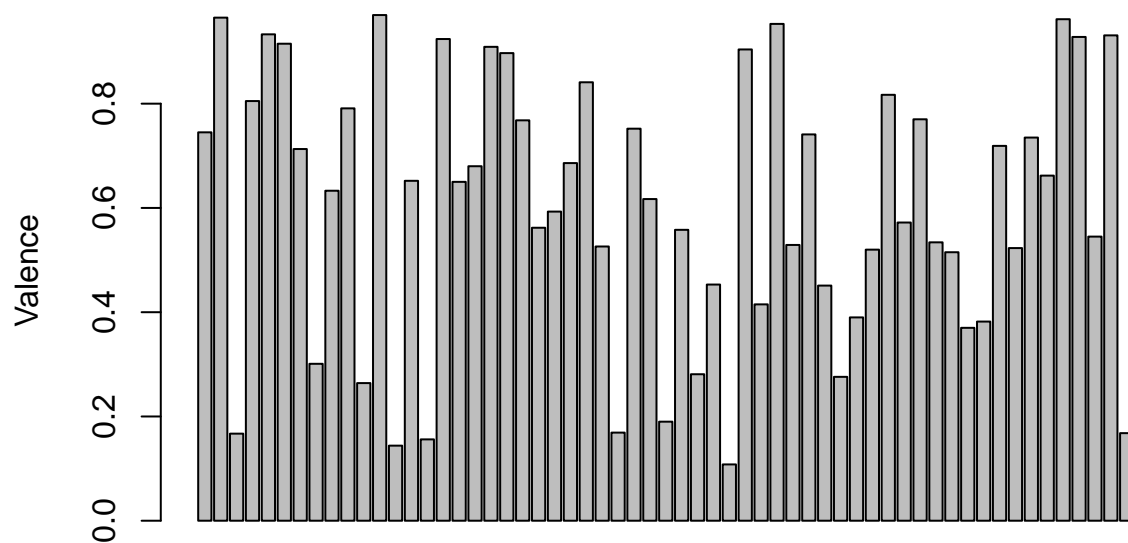
### Liveness vs. Year



Year

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	0.03180	0.07465	0.10600	0.16748	0.21750	0.93200

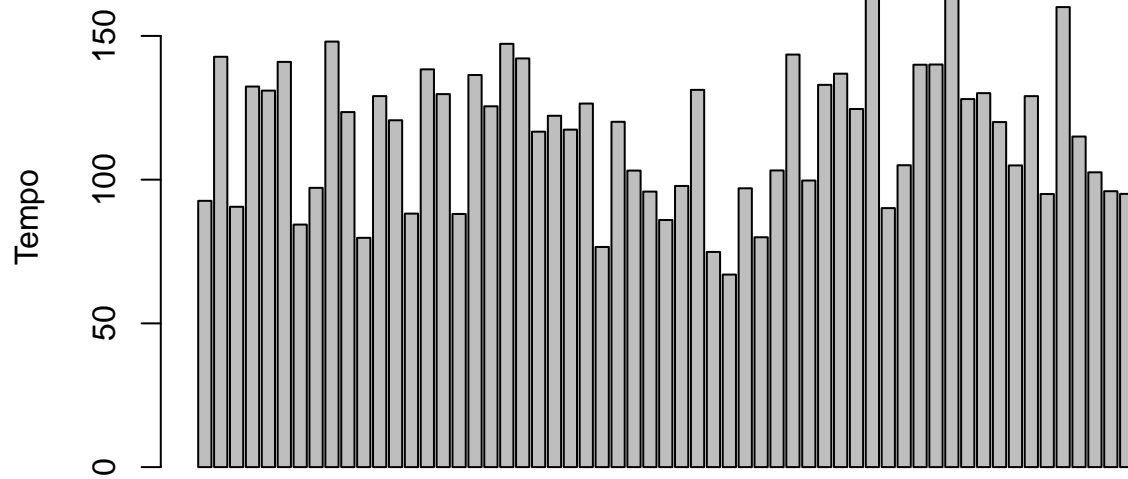
### Valence vs. Year



Year

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	0.1080	0.4330	0.6330	0.6039	0.7980	0.9700

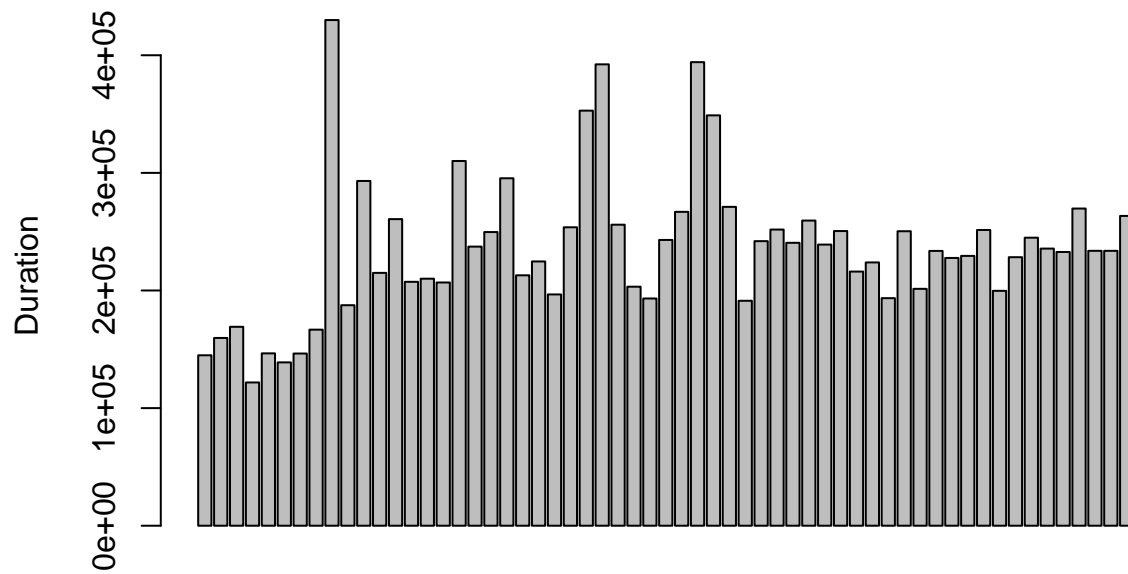
### Tempo vs. Year



### Year

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	66.98	95.91	120.13	116.28	132.68	175.91

### Duration vs. Year



### Year

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	121800	202314	233713	236436	254833	429933

## Description:

Initial data analysis include plotting individual histograms of individual features and looking at basic statistical analysis (min, median, mean, max).

## Analysis:

Danceability has average around 0.64, which is pretty high on the scale. This proves that on average, music that are rhythmic, upbeat- suitable for dancing - tends to be more popular among people. Similarly, energy has average around 0.59, which shows that music lovers prefer songs that consist faster tempo. The average value for tempo is 116, which in music is Allegretto - moderately fast. This also points to the same result as people tend to like more energetic/faster songs. Loudness has average -8.5, this is considerably low on the decibel level, proving that popular music tends to be not too loud. Speechiness and acoustiness both has average value that are low on the scale, showing that people prefer songs that include less electronic sounds and more vocal lines. Instrumentalness and liveness both have pretty low values. This makes sense as most of the songs on the list are recorded in a studio, and are songs instead of symphonic works. The mean and median duration of songs are pretty close together- both have values around 230000 milliseconds (around 3.83 min).

The most interesting analytics comes with the feature “valence”. This is the only feature of the entire dataset that involves user input data. The other features are calculated from the tempo, sound quality of the songs themselves, while valence is calculated through user ratings (labeling whether the song feels happy or sad), user ratings do depends on all the previous audio features ( i.e. danceability, loudness, energy etc). On the histogram for valence, the max value is 0.97, and the min value is 0.1, with average value 0.60. Valence values varies a lot through the 59 songs, and the max and min each stands at the opposite ends of the spectrum.

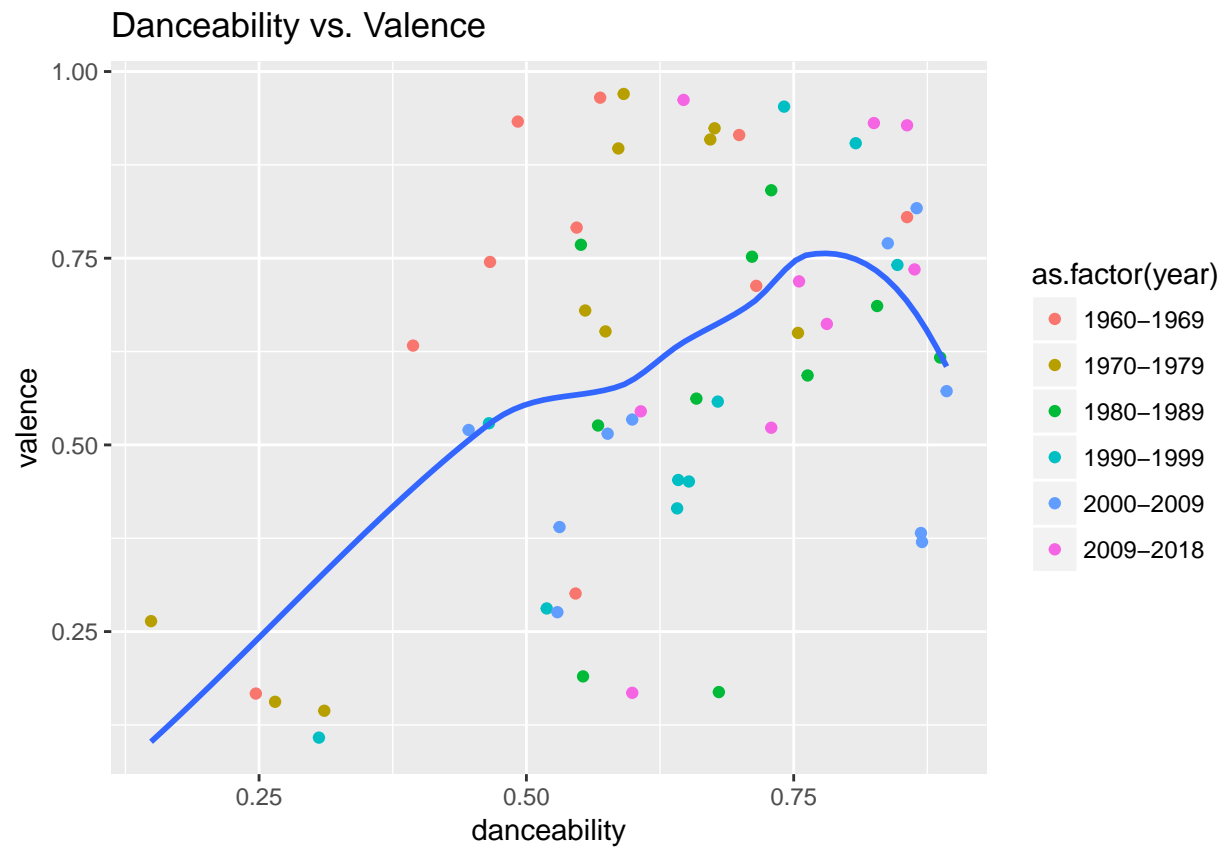
The rest of the project focus on deeper analysis of valence values.

## Changes to dataset values (grouping variable year)

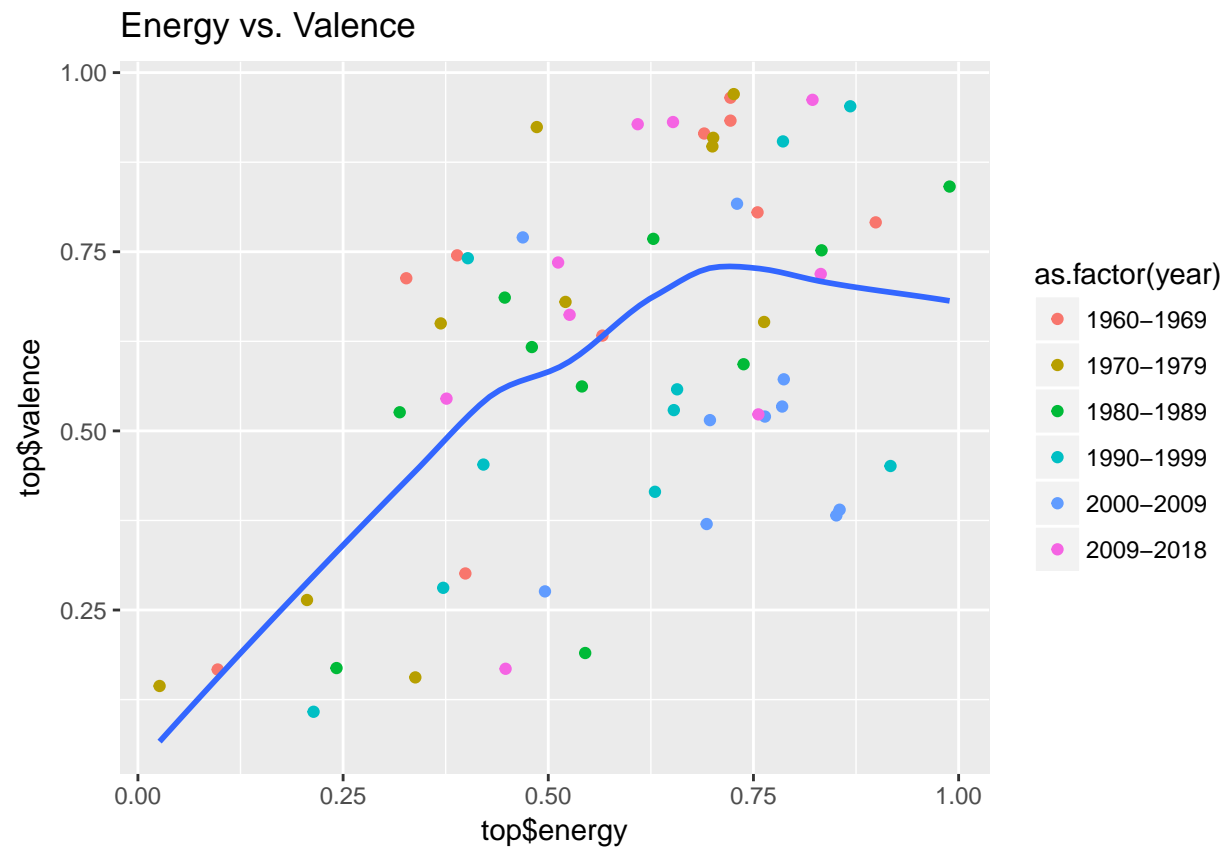
- Change year index so that year would be displayed as by decade instead of just one single year. (this is specifically created for the ggplot analysis later, since I don’t want continuous scale for the legend.)

## GGplot analysis for each audio feature vs. valence

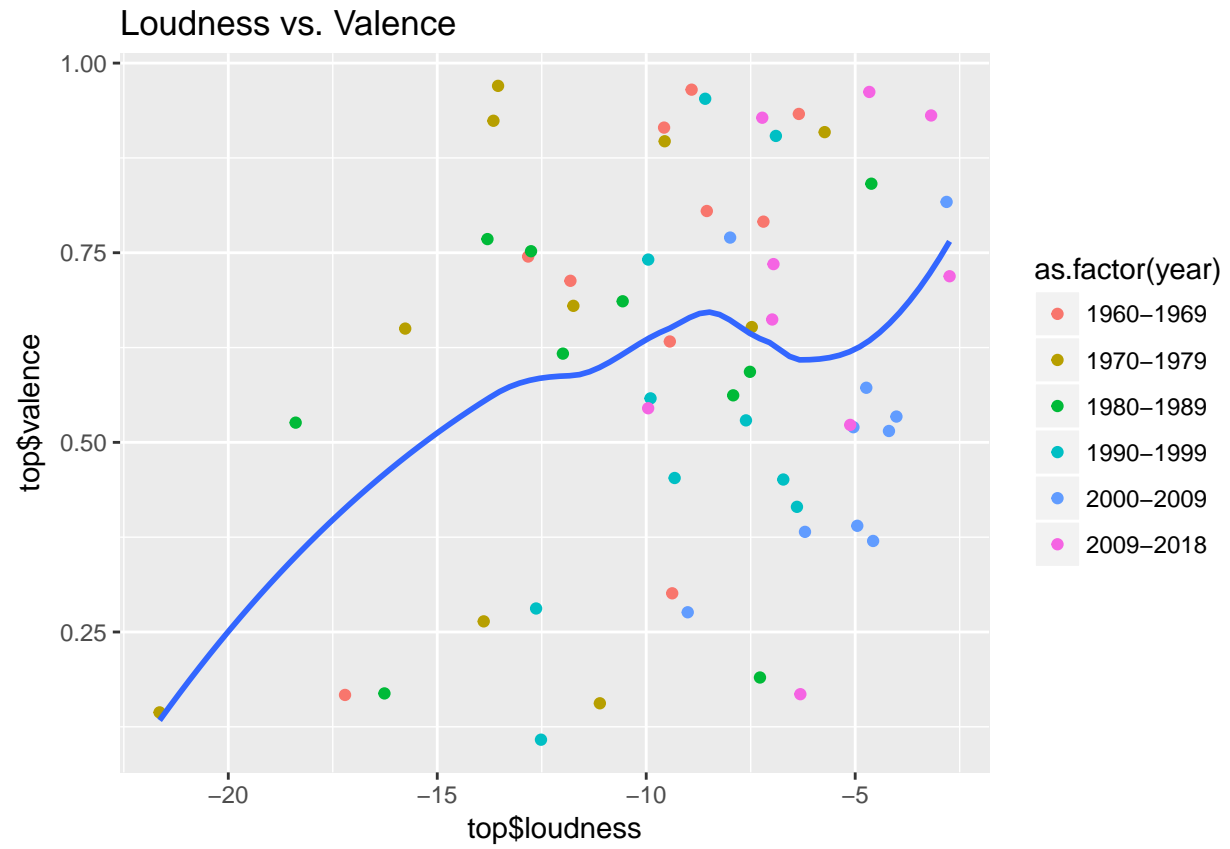
```
## `geom_smooth()` using method = 'loess'
```



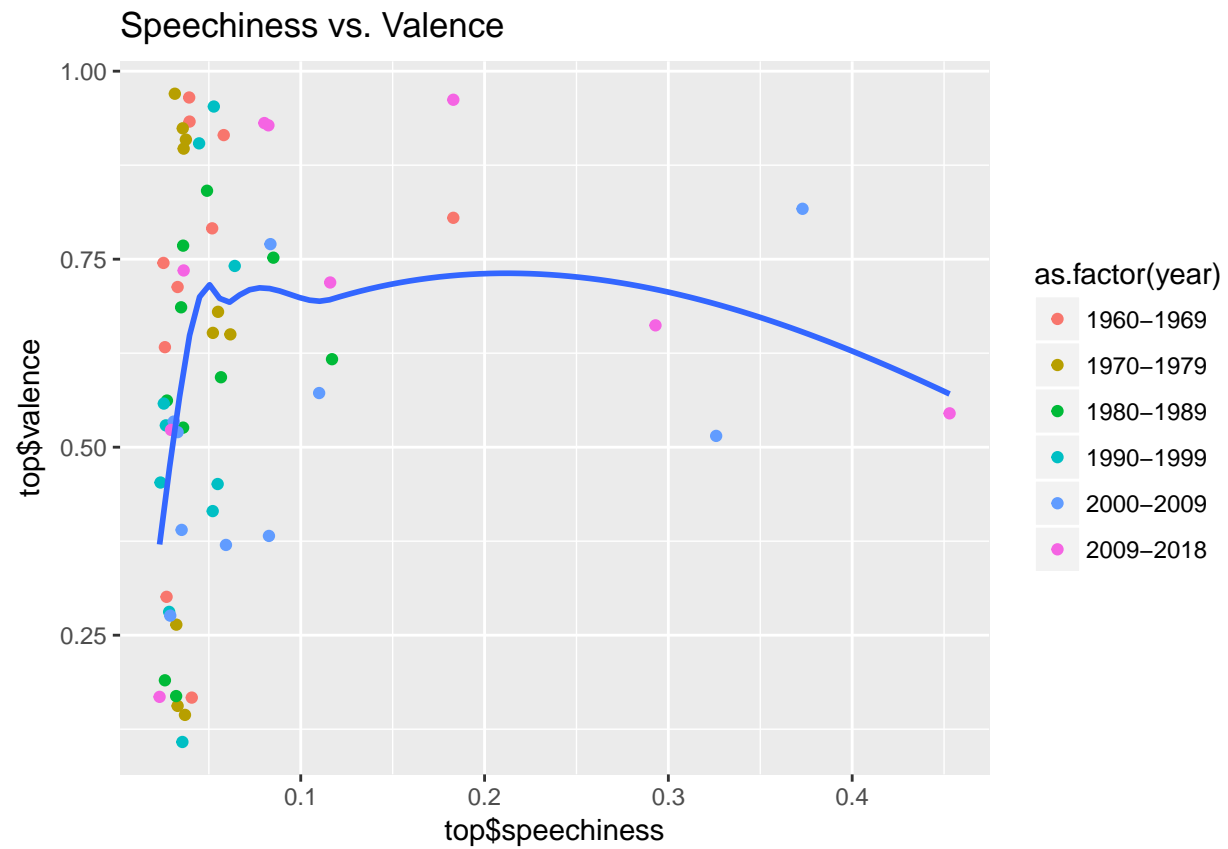
```
## `geom_smooth()` using method = 'loess'
```



```
## `geom_smooth()` using method = 'loess'
```

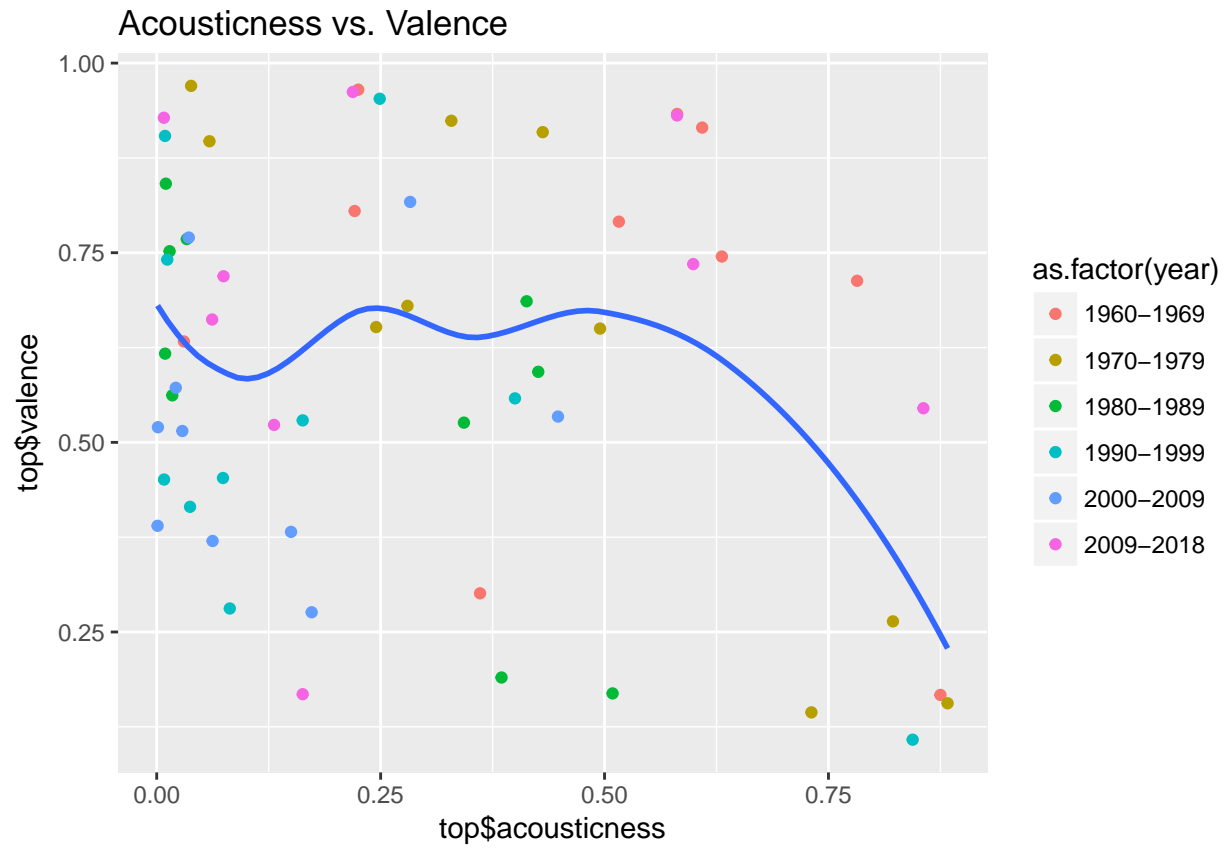


```
## `geom_smooth()` using method = 'loess'
```

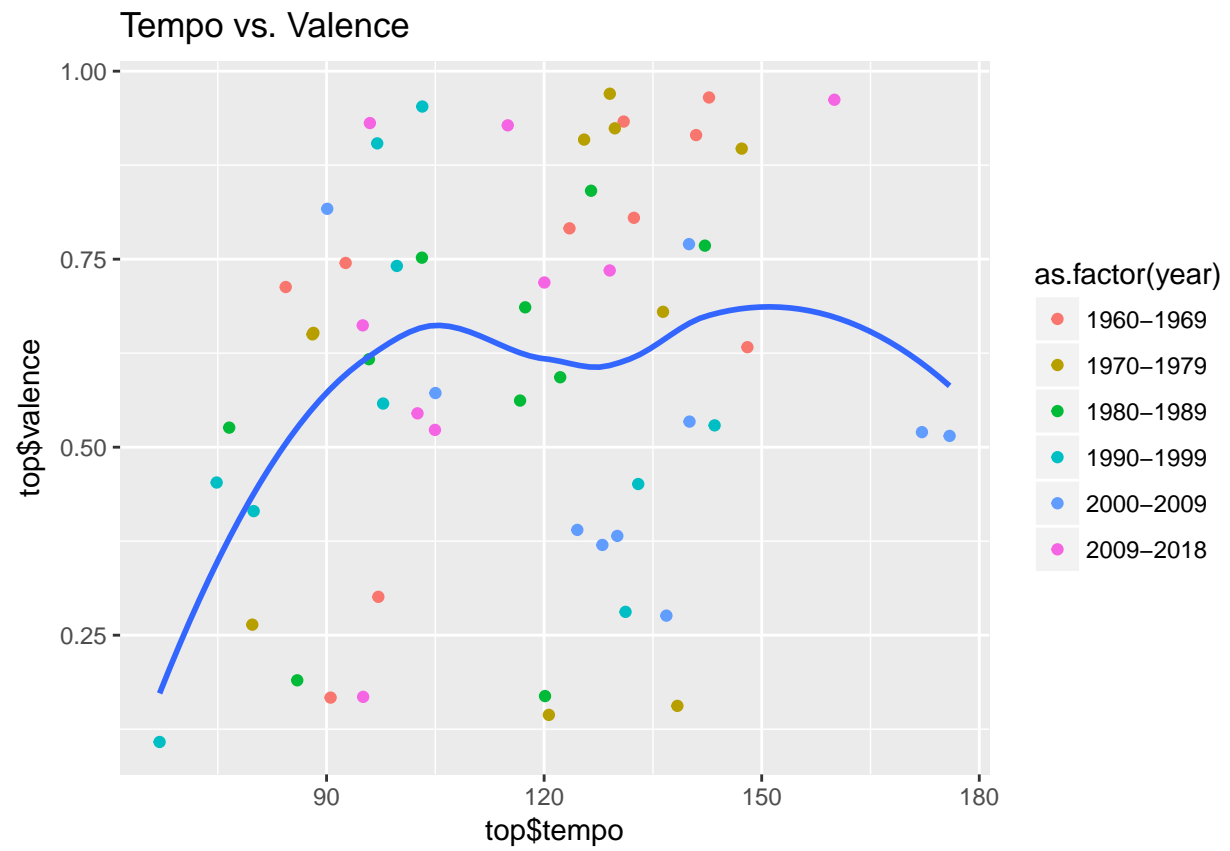


```
## `geom_smooth()` using method = 'loess'
```

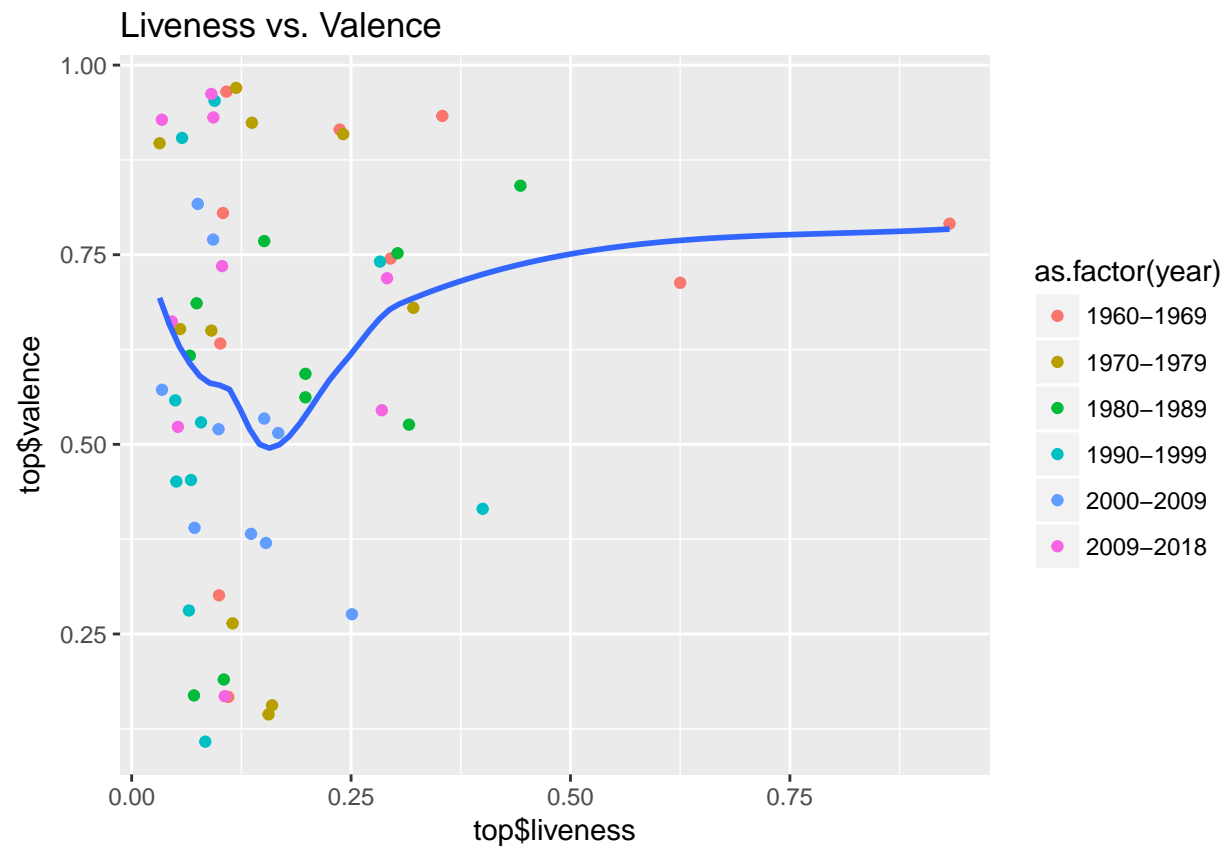




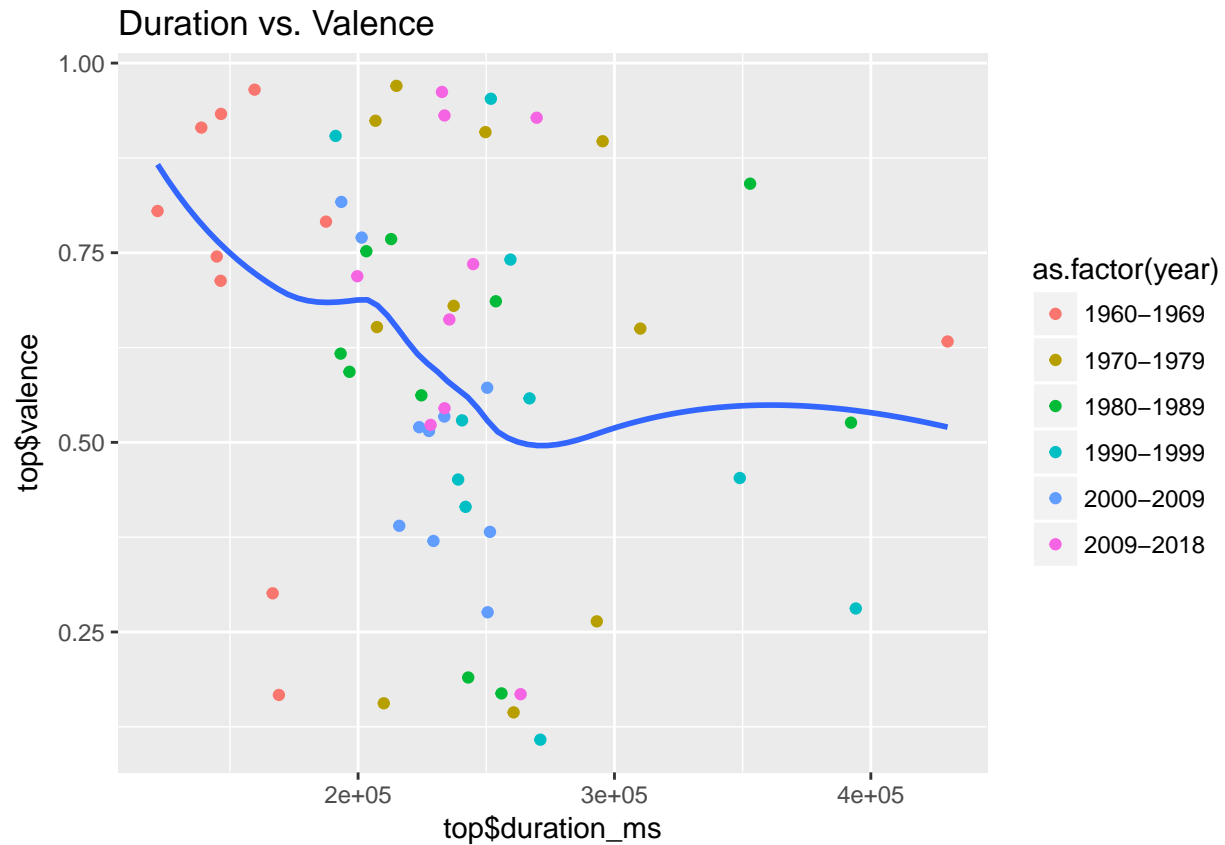
```
## `geom_smooth()` using method = 'loess'
```



```
## `geom_smooth()` using method = 'loess'
```



```
## `geom_smooth()` using method = 'loess'
```



## Description:

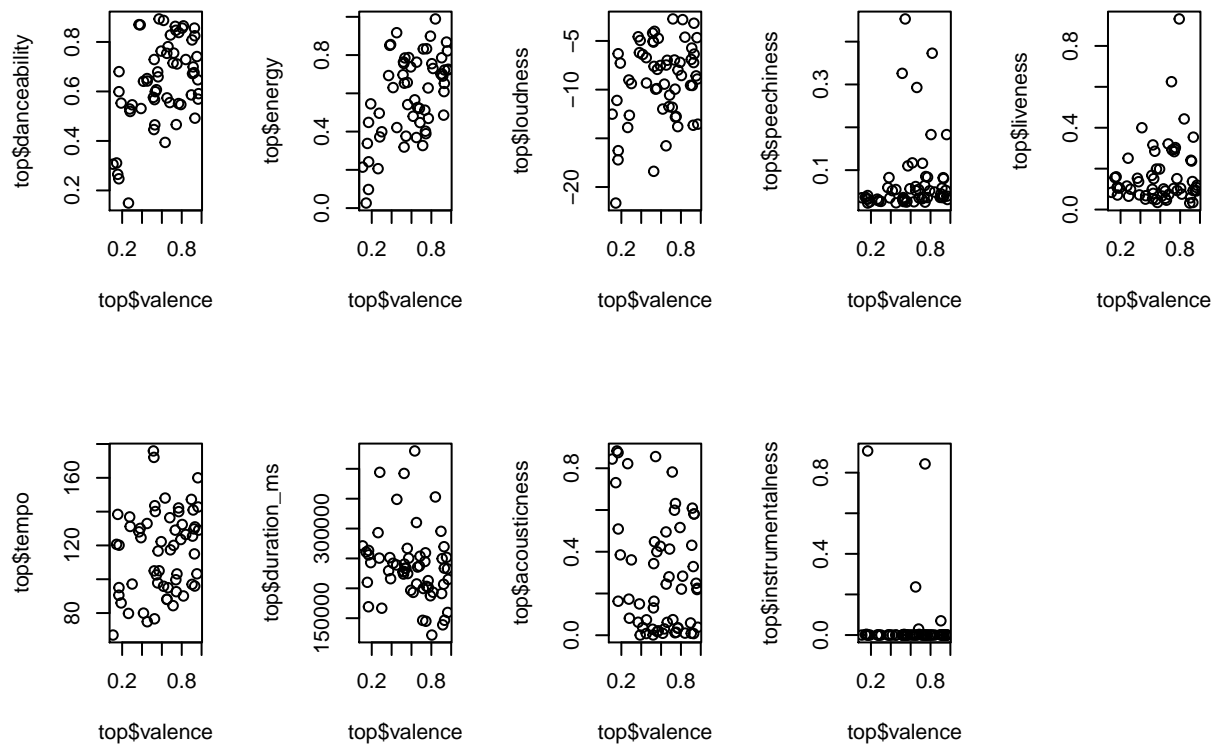
Focusing on analysis of valence level. Plots consist of each individual audio feature graphed on the x-axis, and valence graphed on the y-axis.

## Analysis:

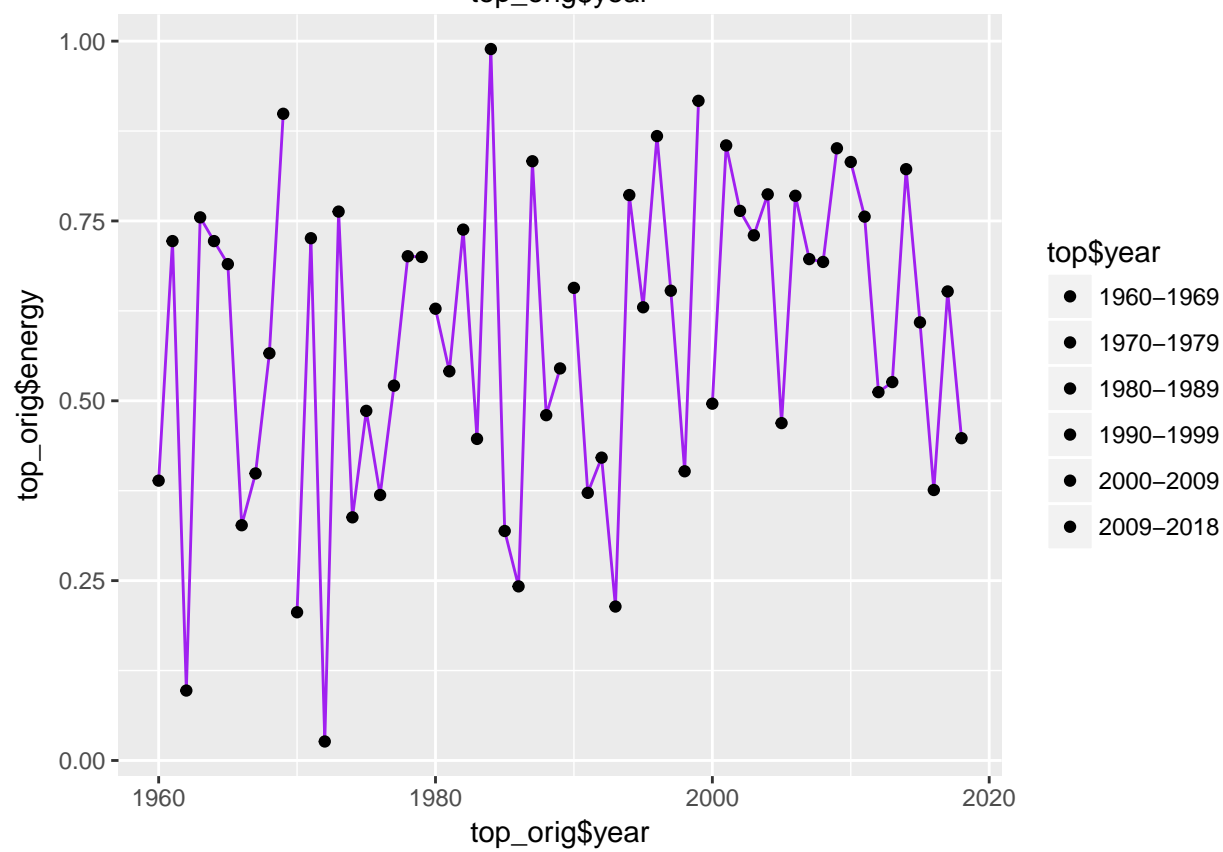
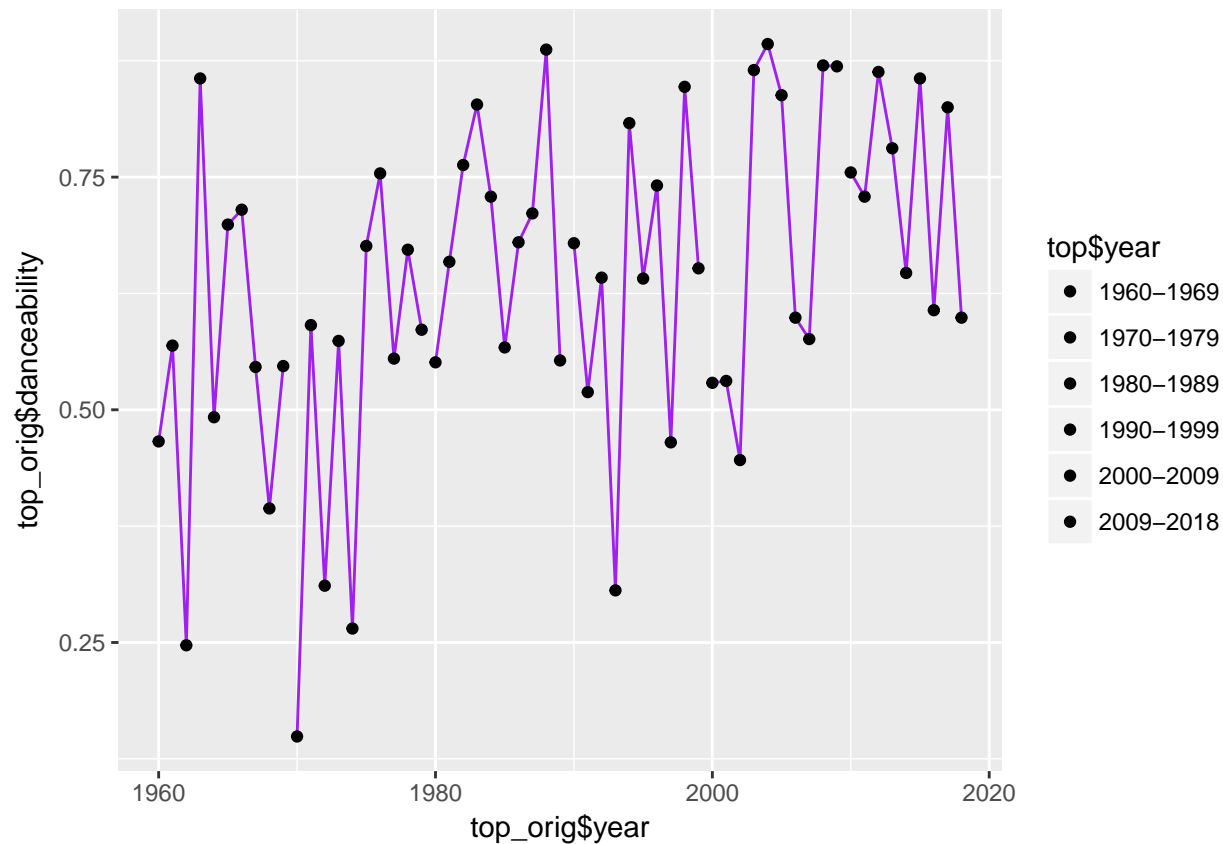
When danceability, energy, tempo increases, valence increases. While for loudness, speechiness, acousticness, and duration of songs, the higher they are on the scale, the lower the valence score. According to the graphs above, songs that are determined as “happy” are generally songs that are fast, upbeat, and not so loud. While songs that fits the “sad” or more negative mood are songs that are slower, longer, and includes more spoken words.

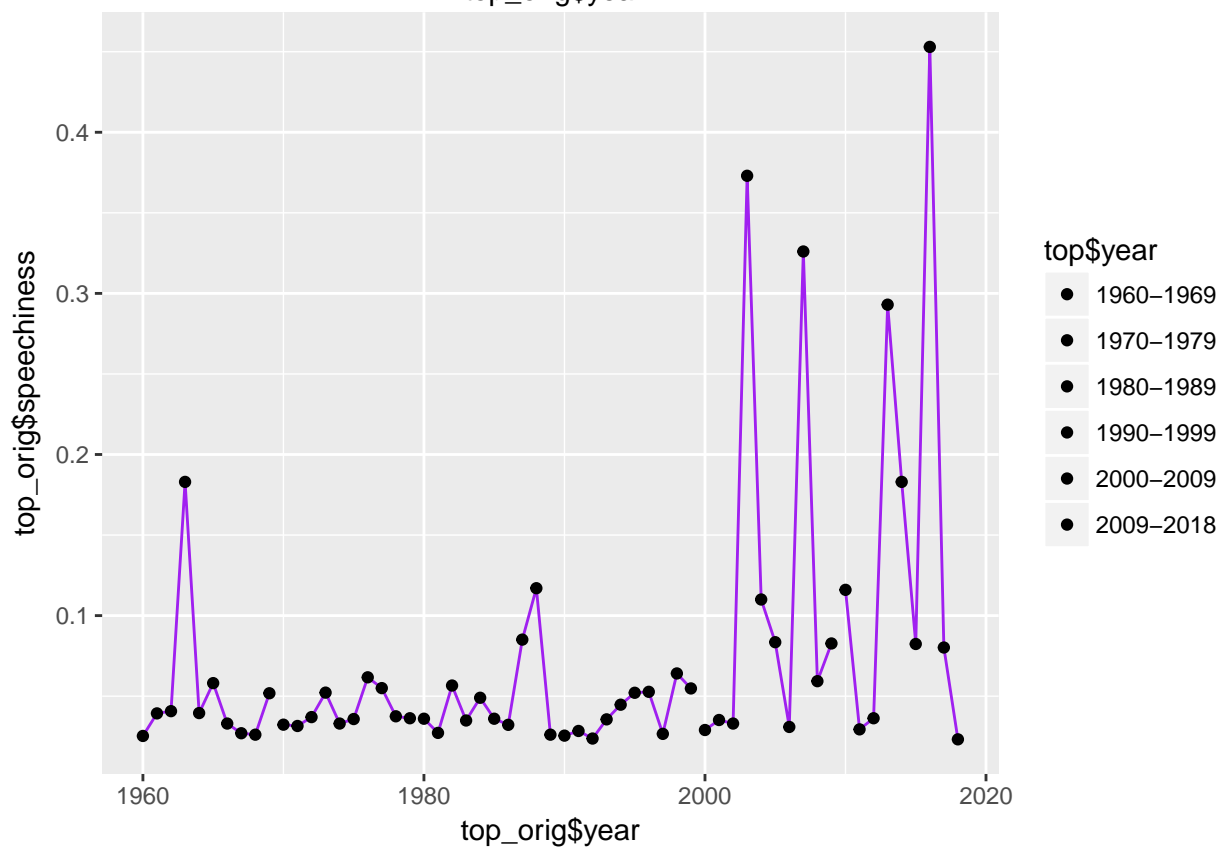
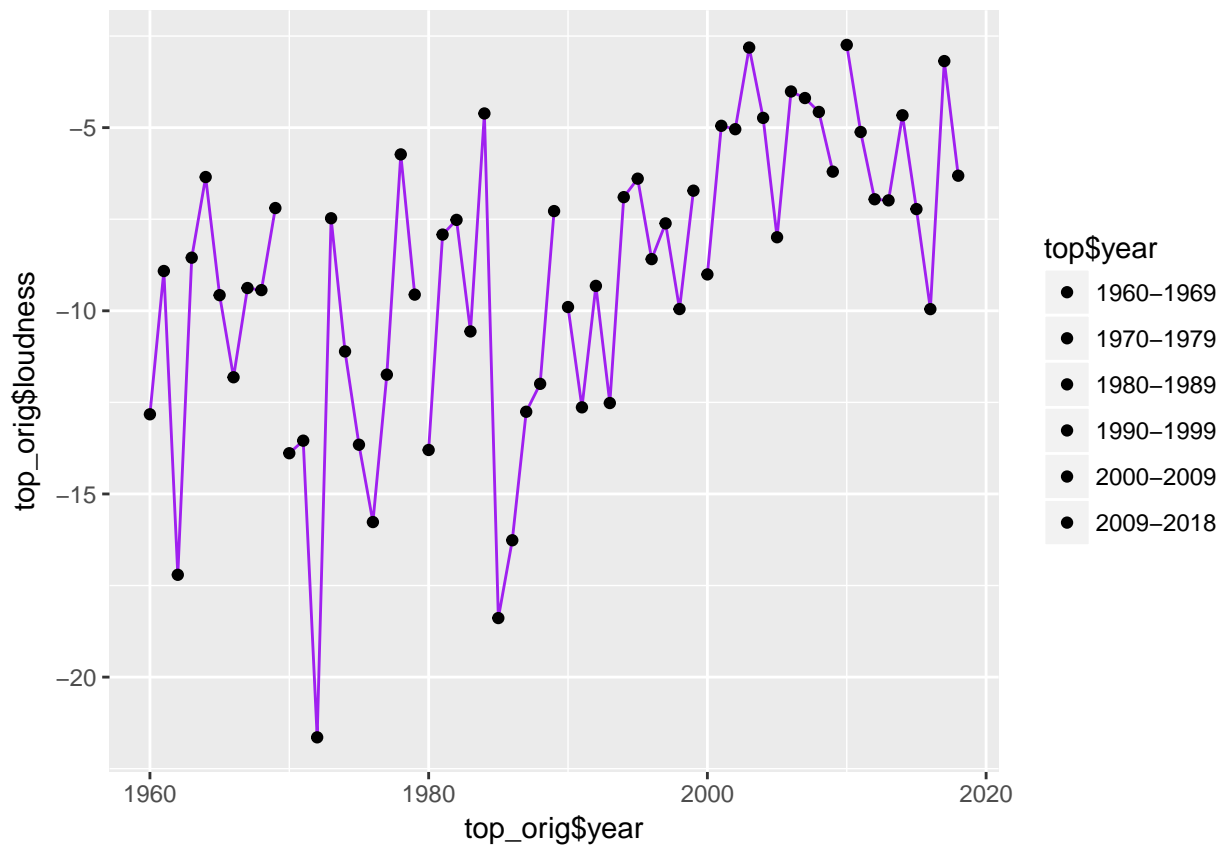
## Scatterplot of different audio features

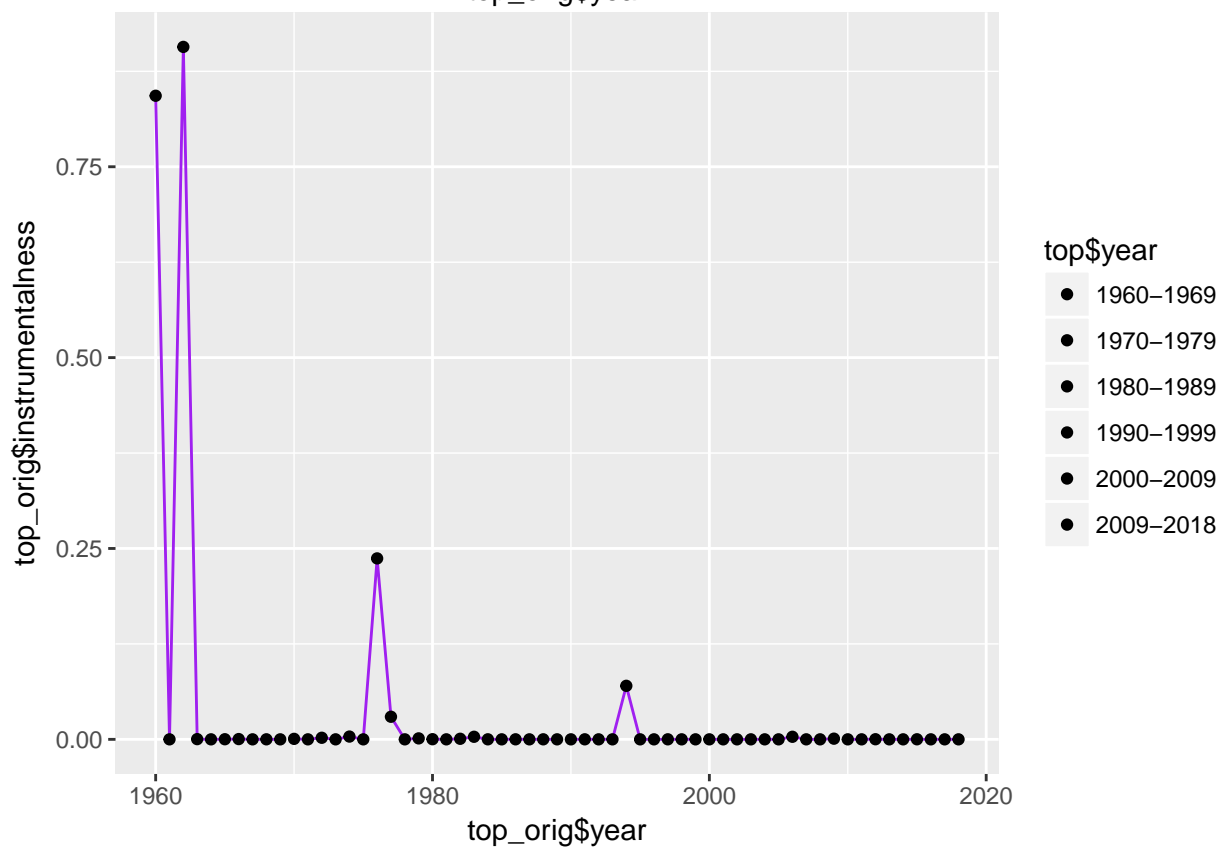
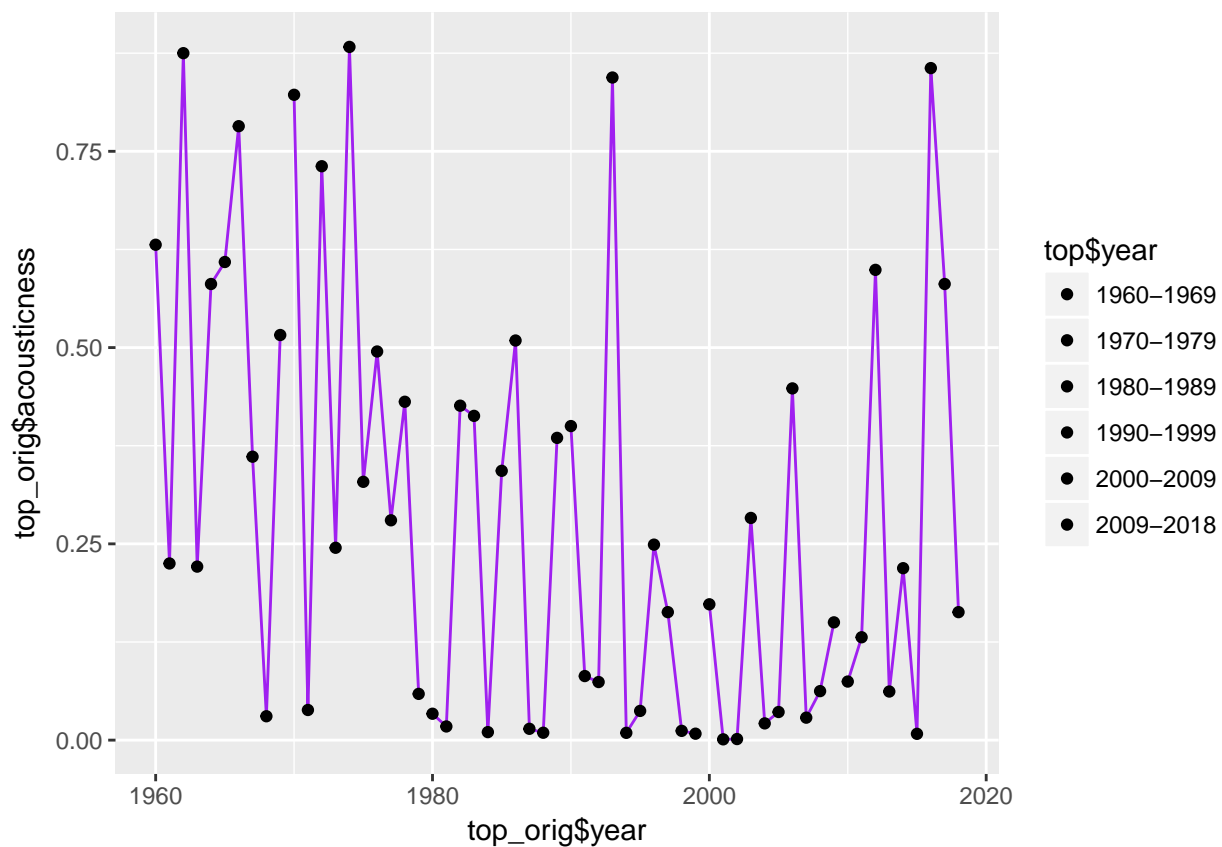
- The scatter plots are used to visualize dataset after filtering out certain valence values (see shiny app for more analysis)



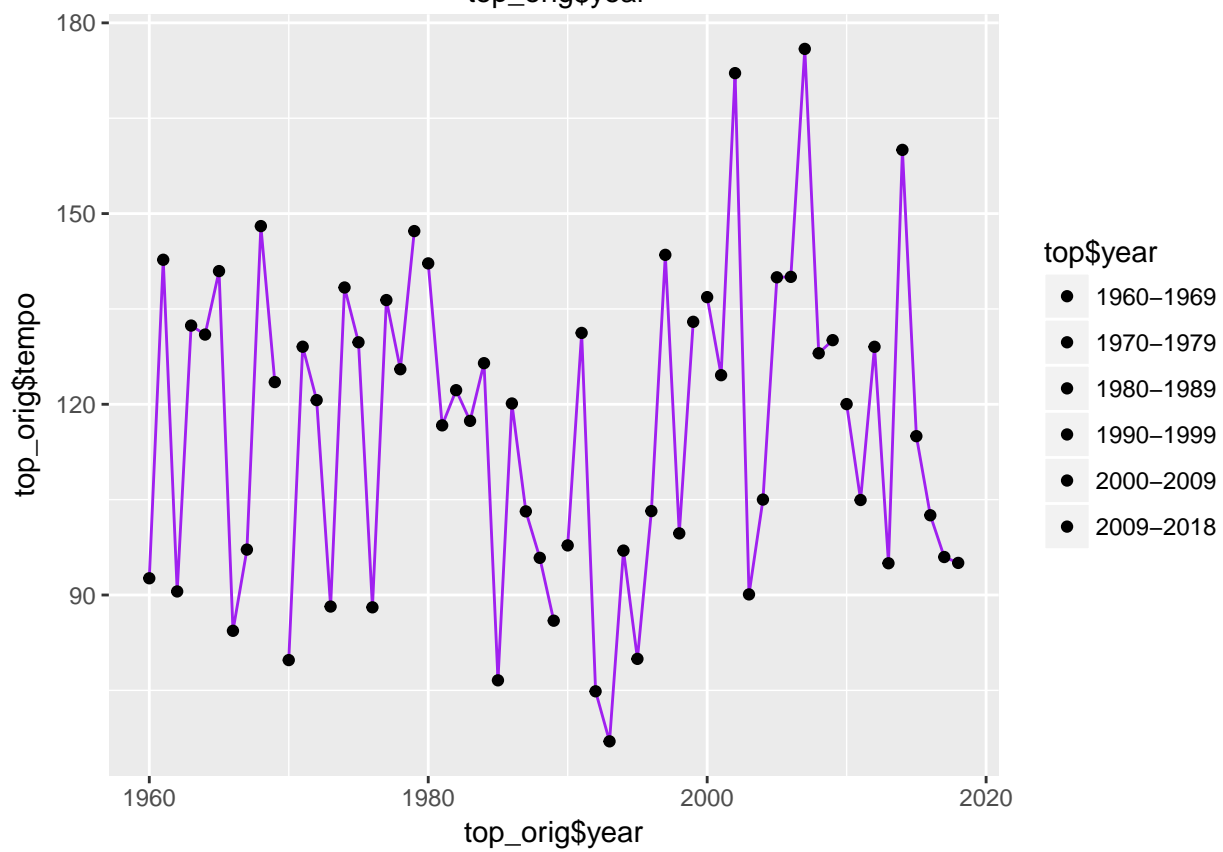
## Individual audio feature time series plot

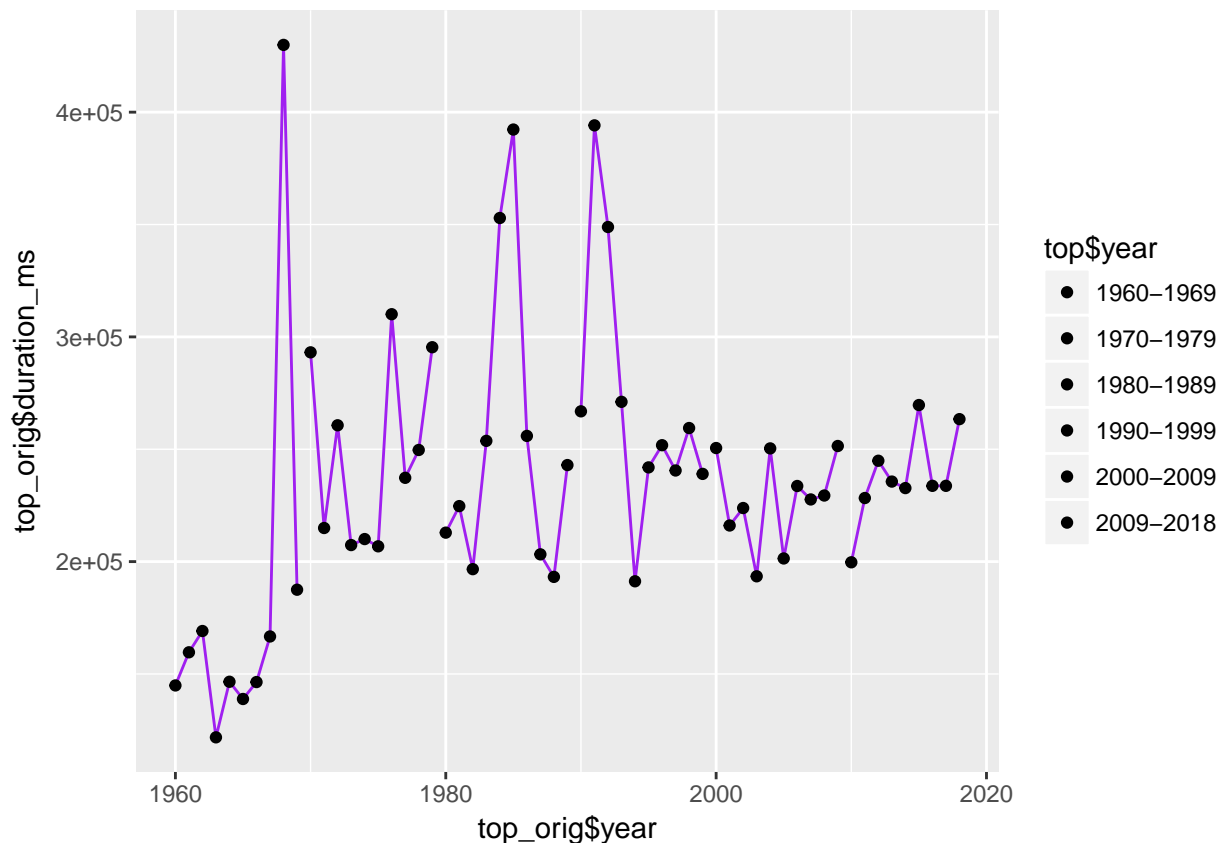












## Description:

Focusing on analysis of individual audio features. Time series plot shows the change of values across time.

## Analysis:

The features that seems to follow an increasing trend is loudness. This shows that as time goes on, popular music tends to be quieter (as the y-axis goes from -60 to 0, the closer loudness to 0, the quieter the songs are).

Similarly with speechiness. The songs between 2000 and 2018 seems to have many spikes in speechiness graphs. One reason could be the increasing popularity in rap music.

Duration is another feature that slightly increased with time. As time went on, music got a little longer (although not by a lot).

Instrumentalness seems to decrease over time. Between 1960 to 1965, music tends to have more instrumental preludes and interludes, while after 1965, popular music tends to have more lyrics and vocal lines.

All the other features seems to follow a pretty constant trend over time.

## Detailed analysis of the valence time series plot



### Description:

Focusing on analysis of valence level. Time series plot shows the change of valence values across time.

### Analysis:

While all the songs are listed as number one sound tracks from each year's hot billboard song ratings. Valence scores varies a lot between each year for each song track.

The happiest song (highest valence score) belongs to the song track "Joy to the World" by Three Dog Night in 1971. The score has a valence score of 0.97, with danceability 0.591, energy 0.726, tempo around 130. So overall, the song is fast, energetic, and suitable for dancing.

The saddest song (lowest valence score) belongs to the song track "I Will Always Love You" by Whitney Houston in 1993. The score has a valence score of 0.108, with danceability 0.306, energy 0.214, tempo 67. A significant drop in feature values from "Joy to the World". It is a slower song, with less energy ratings. Since it's valence rating is almost 0, this song expresses a sad emotion.

## Conclusion:

```
happy <- subset(top_orig,top_orig$valence > 0.95) %>% select("track_name","artist","year","danceability",
  "instrumentalness","liveness","valence","tempo","duration_ms")

sad <- subset(top_orig,top_orig$valence<0.15)%>% select("track_name","artist","year","danceability",
  "instrumentalness","liveness","valence","tempo","duration_ms")

happy
```

##		track_name	artist	year	danceability	energy	
## 2		Tossin' and Turnin'	Bobby Lewis	1961	0.569	0.722	
## 12		Joy to the World	Three Dog Night	1971	0.591	0.726	
## 37		Macarena (Bayside Boys Mix)	Los del Río	1996	0.741	0.868	
## 55		Happy	Pharrell Williams	2014	0.647	0.822	
##		loudness	speechiness	acousticness	instrumentalness	liveness	valence
## 2		-8.914	0.0393	0.2250	0.00e+00	0.1080	0.965
## 12		-13.545	0.0315	0.0384	2.49e-06	0.1190	0.970
## 37		-8.589	0.0527	0.2490	2.19e-06	0.0945	0.953
## 55		-4.662	0.1830	0.2190	0.00e+00	0.0908	0.962
##		tempo	duration_ms				
## 2		142.736	159640				
## 12		129.057	214960				
## 37		103.201	251797				
## 55		160.019	232720				

sad

##		track_name	artist	year	danceability		
## 13		The First Time Ever I Saw Your Face	Roberta Flack	1972	0.311		
## 34		I Will Always Love You	Whitney Houston	1993	0.306		
##		energy	loudness	speechiness	acousticness	instrumentalness	liveness
## 13		0.0264	-21.644	0.0370	0.731	2.05e-03	0.1560
## 34		0.2140	-12.518	0.0356	0.844	5.30e-06	0.0839
##		valence	tempo	duration_ms			
## 13		0.144	120.653	260666			
## 34		0.108	66.980	271093			

After filtering out and analyzing songs with valence score above 0.95 (happy) and valence score below 0.15 (sad), the project concludes with the following analysis:

A happy song generally has danceability rating above 0.6, energy above 0.7, loudness around -10, speechiness around 0.06, acousticness around 0.23, and tempo around 130. So if a song is fast, not too loud, have a good balance between instrumentation and vocal lyrics, it can usually stimulate happiness within human emotions, and would make people want to dance to the tune.

A sad song generally has danceability rating below 0.3, energy below 0.2, loudness around -15, speechiness around 0.03, acousticness around 0.80, and tempo around 60. So when we hear a song that generally involves slower tempo, filled with sounds that are more acoustic (as oppose to electronic), and with less spoken words/lyrics, it usually feels like a sad song.

Interestingly, sad songs seems to score higher in loudness than happy songs. Intuitively, sad songs should be quieter as they generally expresses melancholy and sorrow. Yet, some songs (like “I will always love you” Whitney Houston, which scored the lowest on valence level) gets loud during its climatic section, expressing heartfelt passions and emotions.

So next time, when you need to select a song that fits a certain mood, be sure to check out its audio features and valence scores!