DV2 Final Project: Spotify

Halmschlager Lisa (1902224)

Feb 2020

Introduction

This project is about describing the spotify_songs.csv dataset from the 4th week of #tidytuesday at https://github.com/rfordatascience/tidytuesday (https://github.com/rfordatascience/tidytuesday) and expoloring possible problem/questions.

I am going to demonstrate

- how to work with the data.table package
- how to create various plots using ggplot2
- how to do multidimensional scaling
- how to animate a plot with clustered data points
- how to tweak ggplot2 themes
- how to add a tooltip to an interactive plot

Import data and check dimensions

```
# Get the Data
spotify_songs <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/
2020-01-21/spotify_songs.csv')
dim(spotify_songs)
```

```
## [1] 32833    23
```

The dataset 32833 observations of 23 variables.

```
glimpse(spotify_songs)
```

```
## Observations: 32,833
## Variables: 23
## $ track_id                 <chr> "6f807x0ima9a1j3VPbc7VN", "0r7CVbZTWZgbTCYdf…
## $ track_name               <chr> "I Don't Care (with Justin Bieber) - Loud Lu…
## $ track_artist             <chr> "Ed Sheeran", "Maroon 5", "Zara Larsson", "T…
## $ track_popularity         <dbl> 66, 67, 70, 60, 69, 67, 62, 69, 68, 67, 58, …
## $ track_album_id           <chr> "2oCs0DGTsRO98Gh5ZSl2Cx", "63rPSO264uRjW1X5E…
## $ track_album_name         <chr> "I Don't Care (with Justin Bieber) [Loud Lux…
## $ track_album_release_date <chr> "2019-06-14", "2019-12-13", "2019-07-05", "2…
## $ playlist_name            <chr> "Pop Remix", "Pop Remix", "Pop Remix", "Pop …
## $ playlist_id              <chr> "37i9dQZF1DXcZDD7cfEKhW", "37i9dQZF1DXcZDD7c…
## $ playlist_genre           <chr> "pop", "pop", "pop", "pop", "pop", "pop", "p…
## $ playlist_subgenre        <chr> "dance pop", "dance pop", "dance pop", "danc…
## $ danceability             <dbl> 0.748, 0.726, 0.675, 0.718, 0.650, 0.675, 0.…
## $ energy                   <dbl> 0.916, 0.815, 0.931, 0.930, 0.833, 0.919, 0.…
## $ key                      <dbl> 6, 11, 1, 7, 1, 8, 5, 4, 8, 2, 6, 8, 1, 5, 5…
## $ loudness                 <dbl> -2.634, -4.969, -3.432, -3.778, -4.672, -5.3…
## $ mode                     <dbl> 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0,…
## $ speechiness              <dbl> 0.0583, 0.0373, 0.0742, 0.1020, 0.0359, 0.12…
## $ acousticness             <dbl> 0.10200, 0.07240, 0.07940, 0.02870, 0.08030,…
## $ instrumentalness         <dbl> 0.00e+00, 4.21e-03, 2.33e-05, 9.43e-06, 0.00…
## $ liveness                 <dbl> 0.0653, 0.3570, 0.1100, 0.2040, 0.0833, 0.14…
## $ valence                  <dbl> 0.518, 0.693, 0.613, 0.277, 0.725, 0.585, 0.…
## $ tempo                    <dbl> 122.036, 99.972, 124.008, 121.956, 123.976, …
## $ duration_ms              <dbl> 194754, 162600, 176616, 169093, 189052, 1630…
```

Data cleaning

As a first step I am going to clean the dataset and deal with missing and extreme values

Missing values

```
# find NAs
df_na <- sapply(spotify_songs, function(x) sum(is.na(x)))
data.frame(df_na[df_na >0])
```

```
##                  df_na.df_na...0.
## track_name                      5
## track_artist                    5
## track_album_name                5
```

```
# remove observations with missing data
`%notin%` <- Negate(`%in%`)
spotify_songs <- data.table(spotify_songs)
spotify_songs <- spotify_songs[track_id %notin% spotify_songs[is.na(track_name),track_id],]
```

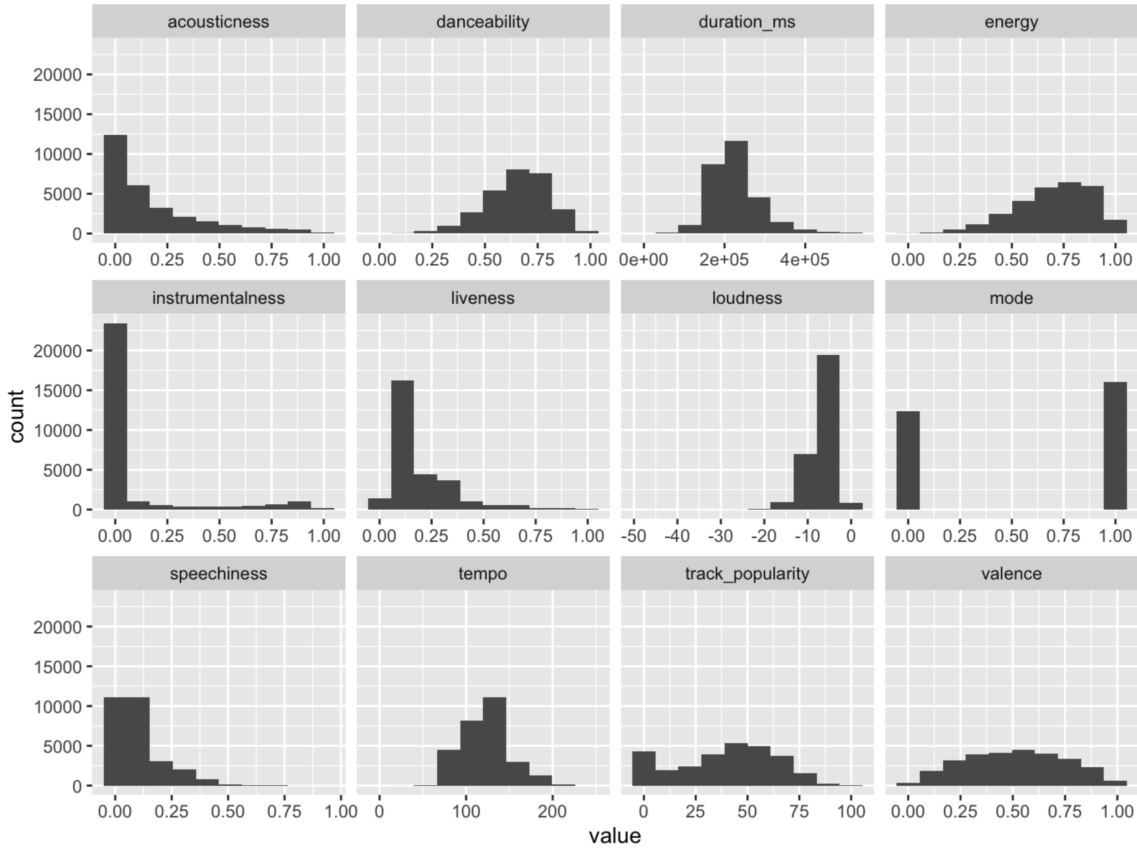
There were 5 missing values for track_name, track_artist and track_album_name, which I removed.

Data exploration

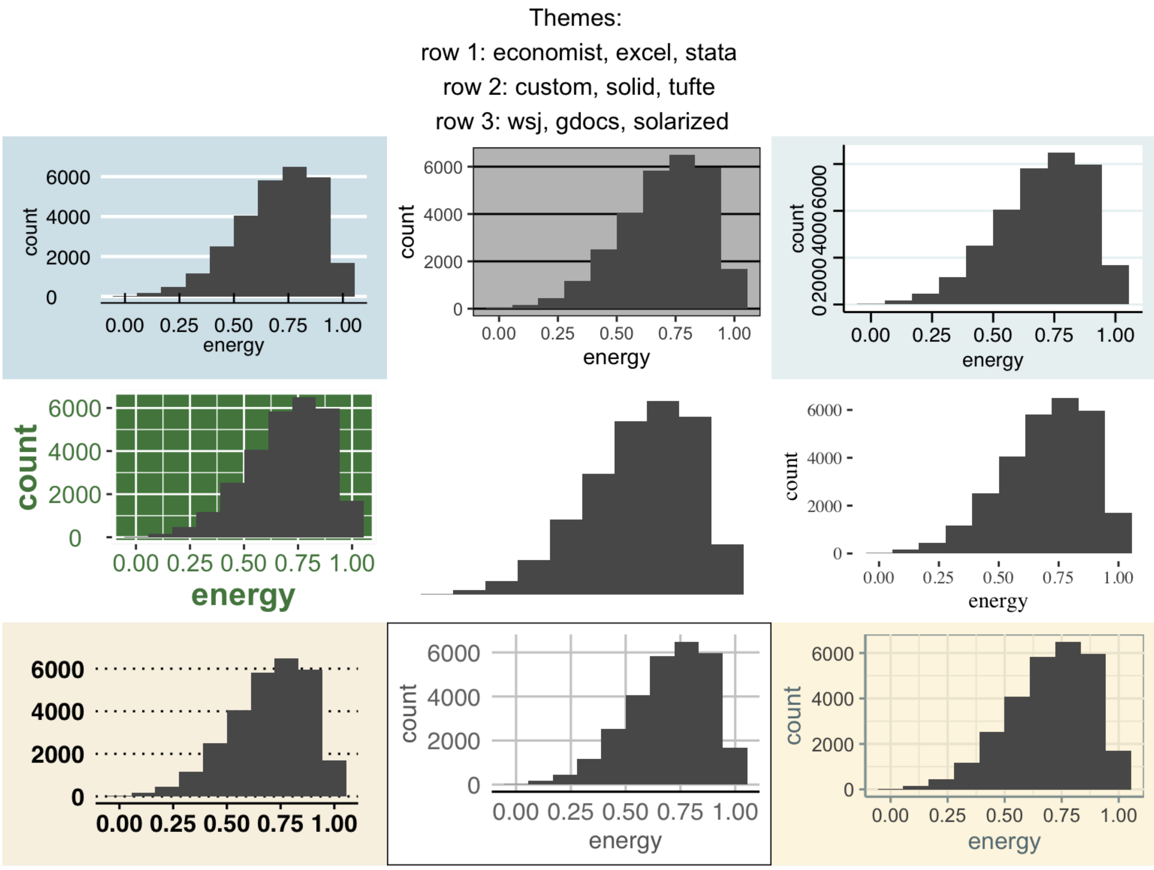
Next I am doing some data exploration to get more familiar with the datset:

```
# Create a new dataset with unique tracks only
ids <- unique(spotify_songs$track_id) #28352
names(ids) <- ids

# add artists, track and track popularity
tracks <- spotify_songs[match(names(ids), spotify_songs$track_id),]
```

```
# plot numeric variables

numcols <- which(sapply(tracks, is.numeric))

ggplot(gather(tracks[,..numcols]), aes(value)) +
     geom_histogram(bins = 10) +
     facet_wrap(~key, scales = 'free_x')
```



```
# try out different themes for one plot

p <- ggplot(tracks, aes(x = energy)) +
     geom_histogram(bins = 10)

p1 <- p + theme_economist() + scale_fill_economist()
p2 <- p + theme_stata() + scale_fill_stata()
p3 <- p + theme_excel() + scale_fill_excel()
p4 <- p + theme_wsj() + scale_fill_wsj('colors6', '')
p5 <- p + theme_gdocs() + scale_fill_gdocs()

theme_custom <- function() {
    theme(
        axis.text = element_text(
            family = 'Arial',
            color  = "#52854C",
            size   = 12),
        axis.title = element_text(
            family = 'Arial',
            color  = "#52854C",
            size   = 16,
            face   = "bold"),
        axis.text.y = element_text(hjust = 0.5),
        panel.background = element_rect(
            fill = "#52854C",
            color = "white",
            size = 2)
    )
}

p6 <- p + theme_custom()
p7 <- p + theme_tufte()
p8 <- p + theme_solid()
p9 <- p + theme_solarized()

#library(ggthemr)
#?ggthemr

grid.arrange(p1,p3,p2,p6,p8,p7, p4,p5,p9, top = paste("Themes:","\n","row 1: economist, excel, stata","\n","row
 2: custom, solid, tufte","\n","row 3: wsj, gdocs, solarized"))
```



```
# inspect factor variables about genre
tracks[, .(count =.N, avg_tempo = mean(tempo), avg_energy = mean(energy)), by = .(playlist_genre, playlist_subgen
re)]
```

```
##     playlist_genre         playlist_subgenre count avg_tempo avg_energy
##  1:            pop                 dance pop  1298  120.1066  0.7421888
##  2:            pop             post-teen pop  1036  124.3547  0.7184825
##  3:            pop               electropop  1251  122.6898  0.7228383
##  4:            pop          indie poptimism  1547  118.0129  0.6371750
##  5:            rap                   hip hop  1296  118.0693  0.5647272
##  6:            rap           southern hip hop  1582  118.9214  0.6810569
##  7:            rap              gangster rap  1314  116.6103  0.6884680
##  8:            rap                      trap  1206  129.8008  0.6579403
##  9:           rock                album rock  1039  122.5159  0.6625255
## 10:           rock              classic rock  1100  123.5544  0.6975100
## 11:           rock            permanent wave   964  124.7375  0.7092049
## 12:           rock                 hard rock  1202  128.8220  0.8457180
## 13:          latin                  tropical  1158  116.9400  0.6735464
## 14:          latin                 latin pop  1097  120.1612  0.6922179
## 15:          latin                 reggaeton   687  117.6784  0.7543552
## 16:          latin             latin hip hop  1194  119.0557  0.7376173
## 17:            r&b       urban contemporary  1187  117.7820  0.5673791
## 18:            r&b                   hip pop   803  116.3243  0.6224746
## 19:            r&b             new jack swing  1036  113.0174  0.6561952
## 20:            r&b                  neo soul  1478  110.1352  0.5408695
## 21:            edm             electro house  1416  125.1971  0.8024859
## 22:            edm                  big room  1034  129.2729  0.8690493
## 23:            edm                   pop edm   967  124.8919  0.7554705
## 24:            edm progressive electro house  1460  126.2906  0.8102616
##     playlist_genre         playlist_subgenre count avg_tempo avg_energy
```

Questions to analyze

Which genre ist the most popular?

Tracks of which genre are using a lot of text and which are compile of more acoustic parts?

In which genre can we find the most live tracks?

```
# compute the average popluarity and liveness of tracks per genre,
# sorted by popularity in descending order
tracks[,.(avg_popularity = round(mean(track_popularity),2),
                 avg_speechiness = round(mean(speechiness),4),
                 avg_acousticness = round(mean(acousticness),4),
                 avg_liveness = round(mean(liveness),4)),
              by = playlist_genre][order(-avg_popularity)]
```

```
##    playlist_genre avg_popularity avg_speechiness avg_acousticness avg_liveness
## 1:            pop          45.91          0.0742           0.1721       0.1773
## 2:            rap          41.85          0.1974           0.1966       0.1911
## 3:          latin          41.45          0.1005           0.2127       0.1817
## 4:           rock          39.69          0.0579           0.1475       0.2048
## 5:            r&b          35.93          0.1155           0.2641       0.1763
## 6:            edm          30.68          0.0879           0.0769       0.2143
```

The most popular genre, on average, is pop, followed by rap and latin.

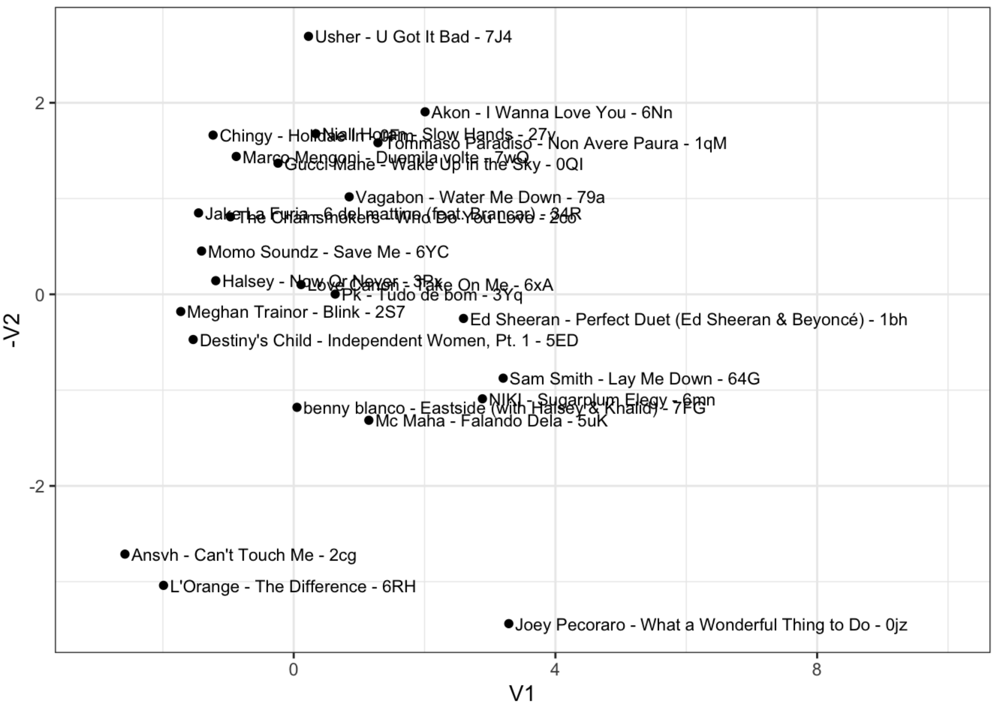
Looking at speechiness, tracks that are made entirely of spoken words are close to 1, while values below 0.33 most likely represent music and other non-speech-like tracks. Within the music category rap is obviously the genre with the highest presence of spoken words in a track, while rock music seems to be rather sparing with words.

Acousticness is a confidence measure from 0.0 to 1.0 of whether the track is acoustic. Among all genres r&b shows the highest average acoustic score in this dataset.

Liveness detects the presence of an audience in the recording. edm tracks were moste likely performed live on average in this dataset.

How do different hip hop tracks compare to each other?

```
# MDS multi dimensial scaling
# for 25 randomly selected tracks in subgenre hip hop

set.seed(456)
df <- tracks[playlist_subgenre=="hip pop",]
df <- df[sample(1:nrow(df), 25, replace=FALSE),]
rn <- paste(df$track_artist, df$track_name , substr(df$track_id, start = 1, stop = 3), sep = " - ")
df <- df[,..numcols]
df <- data.frame(sapply(df, function(x) scale(x)))
rownames(df) <- rn

df_mds <- data.table(cmdscale(dist(df)),keep.rownames = TRUE)
df_mds$song <- rn

ggplot(df_mds, aes(V1, -V2, label = song)) +
  geom_point() +
  geom_text(hjust = 0, nudge_x = 0.1, size = 3) +
  xlim(-3, 10) +
  theme_bw()
```



How does energy influece track popularity?

```
# ANIMATION
# energy vs track_popularity for 1000 randomly choosen tracks.

set.seed(456)

df2 <- tracks[sample(1:nrow(tracks), 1000, replace=FALSE),]
rn <- paste(df2$track_artist, df2$track_name , substr(df2$track_id, start = 1, stop = 3), sep = " - ")
df2num <- df2[,..numcols]
df2num <- data.frame(sapply(df2num, function(x) scale(x)))
rownames(df2num) <- rn


dm <- dist(df2num)
hc <- hclust(dm)
clusters <- dendextend::cutree(hc, 3)

df2$cluster <- factor(clusters)

ggplot(df2,aes(energy, track_popularity, color = factor(clusters))) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se= FALSE) +
  transition_states(playlist_genre)+
  labs(colour = "Cluster",
       title = paste("{closest_state}"),
       subtitle = "Number of tracks: {nrow(subset(df2, playlist_genre == closest_state))}")+
  theme_economist() + scale_fill_economist()
```

What are the 10 most favoured artists in terms of their track popularity?

```
tracks[,.(avg_popularity = round(mean(track_popularity),2),
          tracks = .N),by=track_artist][order(-avg_popularity)][1:10]
```

```
##       track_artist avg_popularity tracks
##  1:  Trevor Daniel          97.00      1
##  2:            Y2K          91.00      1
##  3:    Don Toliver          87.50      2
##  4:           Kina          85.50      2
##  5:       JACKBOYS          84.33      3
##  6:   Dadá Boladão          84.00      1
##  7:         DaBaby          83.67      6
##  8:    Roddy Ricch          83.43      7
##  9:      Baby Keem          83.00      1
## 10: Internet Money          83.00      1
```

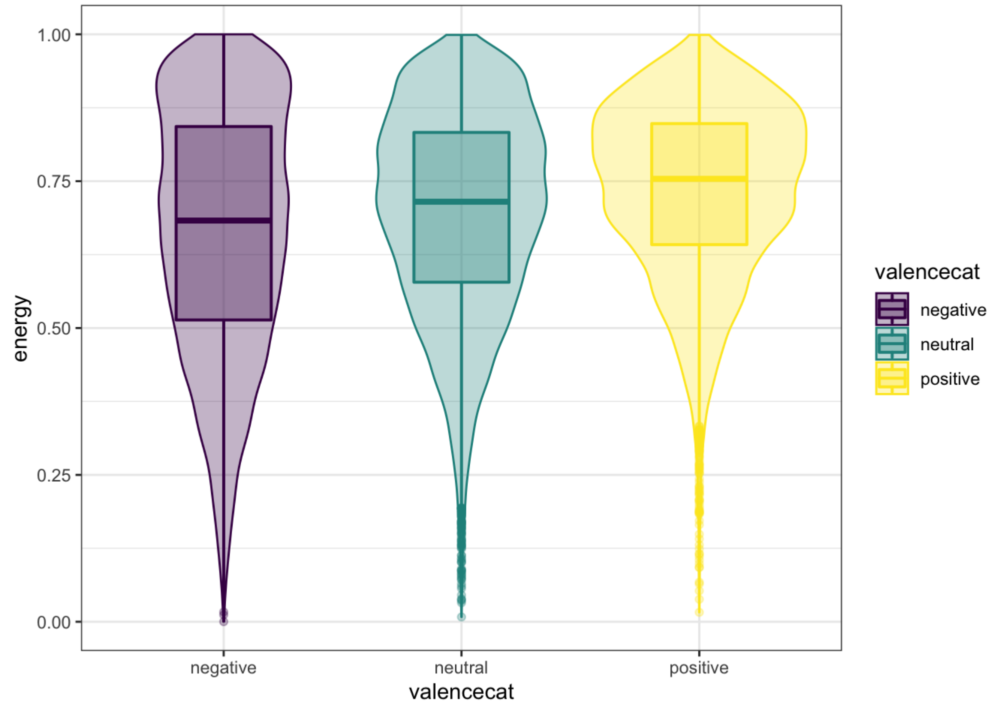
Who is Tevor Daniel?

```
tracks[track_artist == "Trevor Daniel", c(1,2,4,6,10)][order(track_album_name)]
```

```
##                  track_id track_name track_popularity track_album_name
## 1: 4TnjEaWOeW0eKTKIEvJyCa    Falling               97          Falling
##    playlist_genre
## 1:            pop
```

Is there a different energy in tracks that are conveying positiveness or negativeness?

```
# group valence into three groups
a <- spotify_songs[, valencecat := cut(valence, 3, labels = c("negative", "neutral", "positive"), ordered_result
 = TRUE )]

# boxplot + violinplot
ggplot(a, aes(valencecat, energy, color = valencecat, fill = valencecat)) +
  geom_violin(alpha = 0.3) +
  geom_boxplot(size = 0.7, width = 0.4, alpha = 0.3) +
  theme_bw()
```



Valence is 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).

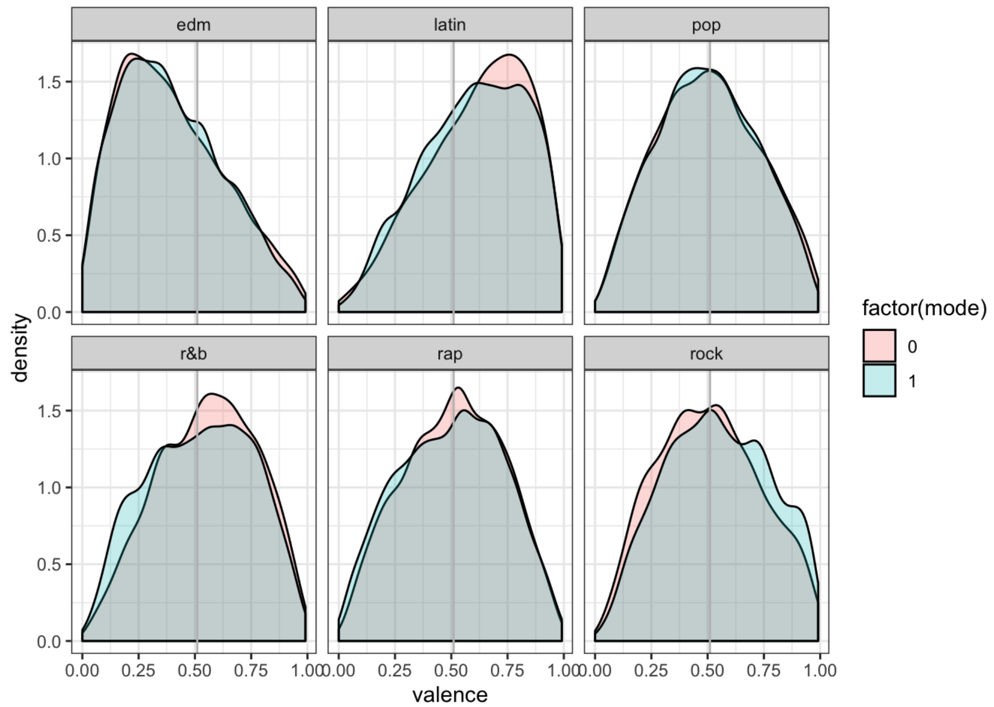
Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

I grouped the trackes into three groups based on their valence and visualised their energy distribution.

Positive tracks have on average more engergy than negative songs.

Is mode an indicator for postiveness or negativeness (valence)?

```
# Density chart:
ggplot(tracks, aes(valence, fill = factor(mode))) +
  geom_density(alpha = 0.25) +
  theme(legend.position = 'top') +
  geom_vline(aes(xintercept = mean(valence)), color="grey") +
  facet_wrap(~playlist_genre) +
  theme_bw()
```

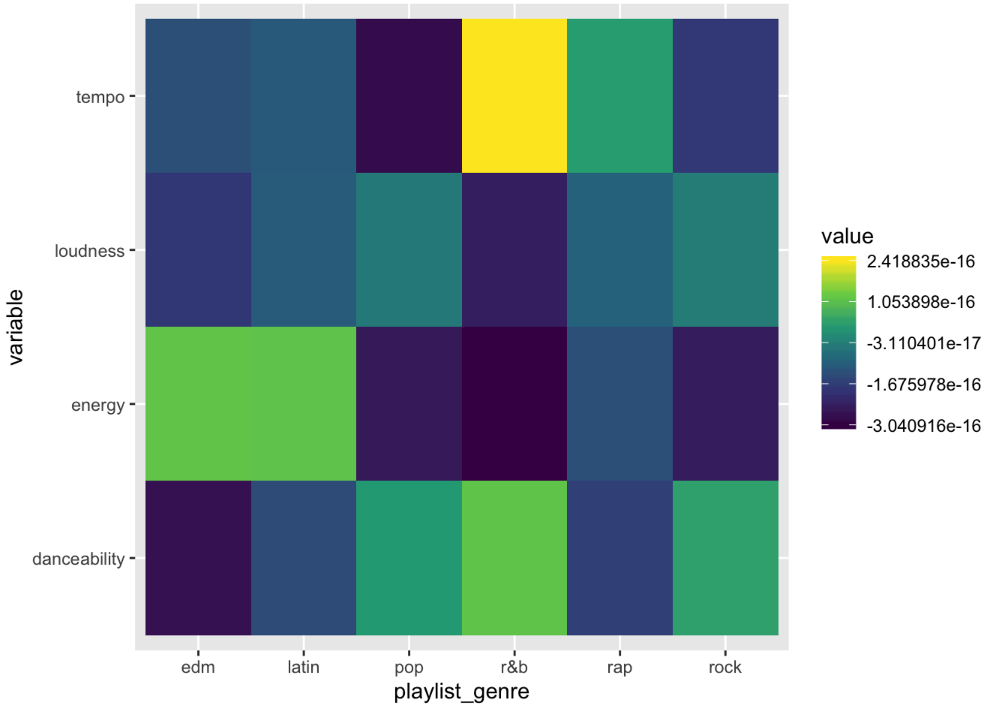


Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.

We can only observe small differences for example in latin music tracks tend to be positive (high valence) and the minor mode is used more frequently in those high valence tracks. The opposite is true for rock, where tracks with high valence (positiveness) use major more often and minor in tracks with low valence (negativeness).

Which genres are most energetic, loud, and good to dance?

```
# Heatmap
ggplot(melt(tracks[, .(danceability = mean(scale(danceability)),
                       energy = mean(scale(energy)),
                       loudness = mean(scale(loudness)),
                       tempo = mean(scale(tempo))),
                   by = playlist_genre], id = 'playlist_genre'),
       aes(playlist_genre, variable, fill = value)) + geom_tile() +
  scale_fill_viridis_c()
```



r&b tracks have the highest average beat duration, loudness and energy. edm is the genre with the highest danceability on average.

What are the most danceable songs per genre?

```
tracks[,c(2:3,10,12)][order(-danceability)][, head(.SD, 1), by=playlist_genre]
```

```
##    playlist_genre                                       track_name
## 1:            edm If Only I Could (feat. Steve Lucas) - Liem Remix
## 2:            pop                                     Ice Ice Baby
## 3:          latin                   Enseñame a Soñar - Original Mix
## 4:            r&b                                        Slow Down
## 5:            rap                                     Funky Friday
## 6:           rock                                         Hunnybee
##              track_artist danceability
## 1:  Fusion Groove Orchestra        0.983
## 2:              Vanilla Ice        0.979
## 3:                 DJ Goozo        0.979
## 4:               India.Arie        0.977
## 5:                     Dave        0.975
## 6: Unknown Mortal Orchestra        0.956
```

What are the top ten songs that are most often part of a playlists?

```
# list of tracks that were most often part of playlists
head(spotify_songs[,.(count = length(unique(playlist_id))),by = .(track_name, track_artist)][order(-count)],10)
```

```
##                                       track_name   track_artist count
##  1:                                    One Dance          Drake    12
##  2:                                     Señorita   Shawn Mendes    11
##  3:                            Livin' On A Prayer       Bon Jovi    11
##  4:            I Took A Pill In Ibiza - Seeb Remix    Mike Posner    10
##  5:                            Sweet Home Alabama Lynyrd Skynyrd    10
##  6:                           Sweet Child O' Mine  Guns N' Roses    10
##  7:                                 Cheap Thrills            Sia     9
##  8:                                       ROXANNE Arizona Zervas     9
##  9:              I Don't Care (with Justin Bieber)     Ed Sheeran     9
## 10: Sunflower - Spider-Man: Into the Spider-Verse    Post Malone     9
```

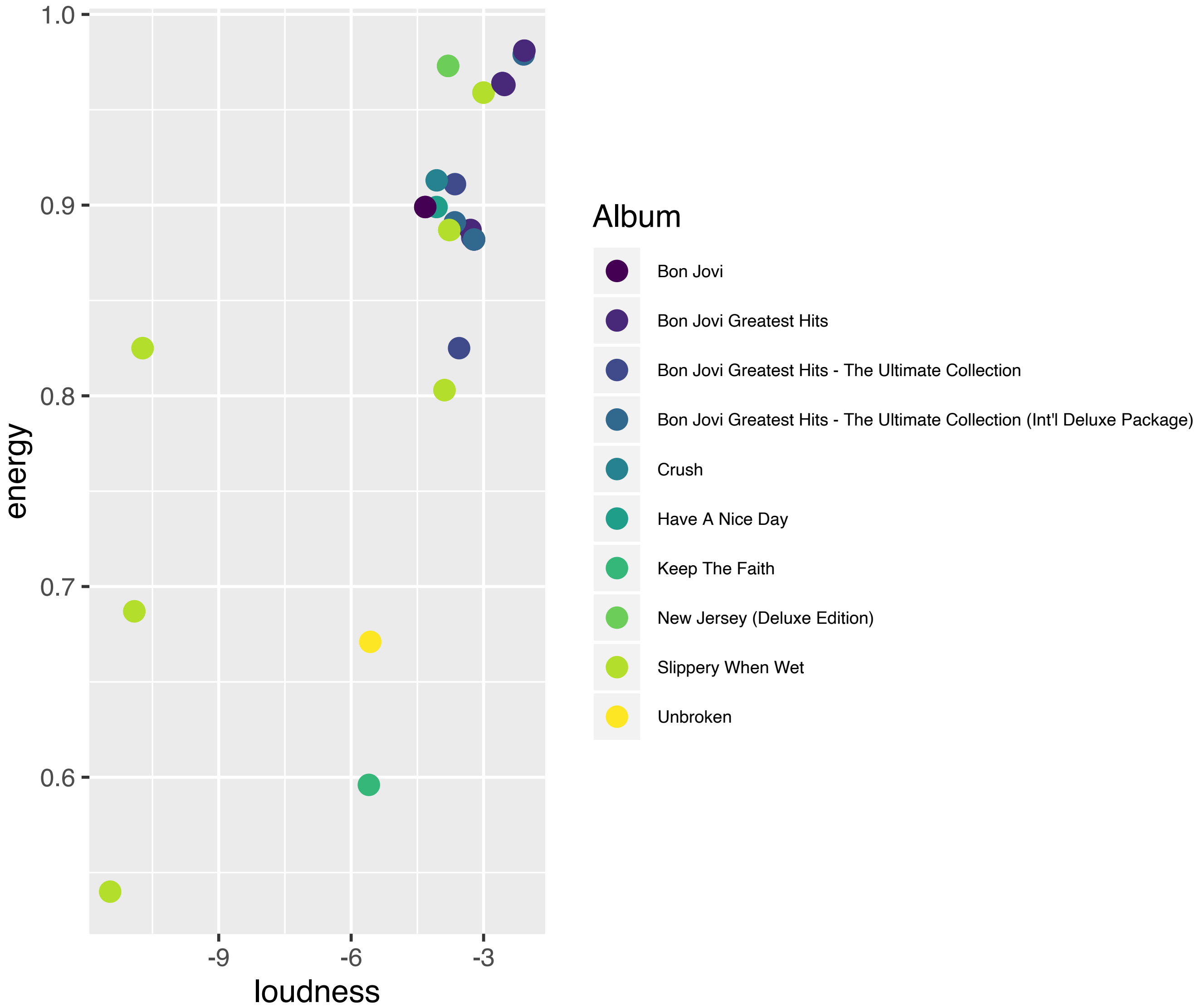
What genre is “One Dance” by “Drake”?

```
spotify_songs[track_name == "One Dance" & track_artist == "Drake",.(count = .N), by = .(playlist_genre, playlist_
subgenre) ][order (playlist_genre)]
```

```
##    playlist_genre  playlist_subgenre count
## 1:            edm            pop edm     2
## 2:          latin      latin hip hop     2
## 3:            pop         electropop     4
## 4:            pop    indie poptimism     1
## 5:            r&b urban contemporary     1
## 6:            r&b            hip pop     1
## 7:            rap   southern hip hop     1
```

Show all Bon Jovi tracks by their album name

```
# TOOLTIP
p <- ggplot(spotify_songs[track_artist == "Bon Jovi",],
            aes(loudness, energy,
                colour = factor(track_album_name),
                tooltip = paste('track:', track_name))) +
  geom_point_interactive(size = 3) +
  labs(colour = "Album", x = "loudness", y = "energy") +
  theme(legend.text = element_text(size = 6)) +
  scale_color_viridis_d()

girafe(ggobj = p, options = list(
  opts_hover(css = "fill:black;"),
  opts_zoom(max = 2)))
```



Summary

In this project I have demonstrated various data visualisation techniques that allowed me to gain the following insights in the spotify_dataset:

- Pop is the most popular genre followed by rap and latin.
- After rap, rock music is the genre with the most spoken words per track, on average, while r&b has the most acoustic parts per track.
- Tevor Daniel, a pop artist, produced the most popular track that could be found in this dataset: Falling (https://open.spotify.com/album/1Czfd5tEby3DbdYNdqzrCa (https://open.spotify.com/album/1Czfd5tEby3DbdYNdqzrCa))
- Tracks conveying positiveness tend to be more energetic, while tracks that are not very energetic are more negative on average.
- The mode of a song however does not necessarily indicate whether a song is more positive or negative
- r&b tracks have the highest average beat duration, loudness and energy. edm is the genre with the highest danceability on average.
- If Only I Could (feat. Steve Lucas) - Liem Remix is the most danceable track in the spotify dataset (https://open.spotify.com/album/0QOi08F2SPc3GznHjwUWLr (https://open.spotify.com/album/0QOi08F2SPc3GznHjwUWLr))
- Tracks that were included most often in different playlists included “One Dance” by Drake, “Senorita” by Shawn Mendes and “Living on a Prayer” by John Bon Jovi. One Dance is apparently not easy to categorize, as it was listed in four different genres (edm latin pop and r&b).
- Investigating the John Bon Jovi albums we can see that most of them are very energetic and loud, only tracks from the album Slippery When Wet were more calm.

Surprising to me was that edm (electronic dance music) had on average a low score for danceability and that I haven’t heard of any of the most popular artists (which could be explained by my age or insufficient exposure to current music hits).