

Final project EDA

Harrison DiStefano

Final Project EDA

```
# Libraries
#install.packages("RMariaDB")
library(RMariaDB)

## Warning: package 'RMariaDB' was built under R version 4.0.5
library(DBI)
library(ggplot2)

# Connect to database
db_host <- Sys.getenv("DB_READ_ENDPOINT")
db_user <- Sys.getenv("DB_READ_USER")
db_pw   <- Sys.getenv("DB_READ_PASSWORD")
db_port <- Sys.getenv("DB_READ_PORT")
db_name <- Sys.getenv("DB_READ_DB")
db_drv  <- RMariaDB::MariaDB()

con <- dbConnect(db_drv, user=db_user, password=db_pw, dbname=db_name, host=db_host, port=db_port)

dbListTables(con)

## [1] "music"      "postal"     "postal_r"
# Take a look at the first few rows
query1 <- "
SELECT *
FROM music
LIMIT 10

;

"

result1 <- dbGetQuery(con, query1)
result1

##    id                  track        artist
## 1  1      The Continental Walk  The Rollers
## 2  2              Two Lovers  Mary Wells
## 3  3            If I Knew  Nat King Cole
## 4  4 "Lara's Theme from ""Dr. Zhivago"""
## 5  5      Say Wonderful Things  Patti Page
## 6  6      Till The End Of The Day  The Kinks
## 7  7          Hot Smoke & Sasafrass  The Bubble Puppy
## 8  8        I'm A Drifter  Bobby Goldsboro
```

```

## 9      Bust Out      The Busters
## 10     School Is Out Gary U.S. Bonds
##
##                                     uri danceability energy song_key loudness
## 1  spotify:track:00Bu7AiNb06604KMuYTQAI    0.603  0.732      0 -5.647
## 2  spotify:track:00CmjheeHvAVKvx3tcIizTy    0.678  0.405      2 -16.965
## 3  spotify:track:00Vwp9jQUss52J0nbbLaz5e    0.371  0.386      1 -9.238
## 4  spotify:track:00YhuN9o0mXUyLQiHjXPxt    0.361  0.280      7 -13.422
## 5  spotify:track:010BIyGminG03GMg8afVAq    0.490  0.440      3 -9.387
## 6  spotify:track:014N0unS25K1LbcM6D1Q5I    0.542  0.929      0 -7.066
## 7  spotify:track:01AxKIwrI7bCLOZ0nmw41I    0.558  0.738      0 -14.270
## 8  spotify:track:01cZbN980X7YkWdzSR1BGD    0.426  0.404      0 -17.804
## 9  spotify:track:01f0S7TvfazvHbg1fEbIug    0.445  0.787      6 -10.145
## 10 spotify:track:01GarP7Iim3fsxASclkEFW    0.464  0.778     10 -11.338
##
##   song_mode speechiness acousticness instrumentalness liveness valence tempo
## 1          1       0.0372      0.80700      0.00e+00  0.0993  0.802 105.425
## 2          1       0.0304      0.42600      0.00e+00  0.1090  0.960 105.902
## 3          1       0.0308      0.70800      4.67e-04  0.0787  0.169 80.207
## 4          1       0.0294      0.82100      4.35e-01  0.1440  0.213 82.298
## 5          1       0.0321      0.87400      0.00e+00  0.3370  0.426 109.329
## 6          1       0.0784      0.52600      5.97e-03  0.1250  0.793 140.800
## 7          1       0.0668      0.75000      4.79e-03  0.0876  0.841 82.556
## 8          1       0.0339      0.10600      1.43e-04  0.0351  0.654 198.205
## 9          1       0.0772      0.00812      8.17e-01  0.1740  0.857 121.472
## 10         1       0.1850      0.86800      3.68e-05  0.4340  0.812 149.199
##
##   duration_ms time_signature chorus_hit sections hit decade
## 1      144000                  3  31.93079      6  1  60s\r
## 2      167000                  4  29.18796      8  1  60s\r
## 3      168000                  4  57.12898      7  1  60s\r
## 4      160000                  3  38.22192      8  1  60s\r
## 5      140000                  3  21.83825      7  1  60s\r
## 6      138000                  4  88.39831      5  1  60s\r
## 7      156000                  4  27.82633      9  1  60s\r
## 8      208000                  4  32.27175     10  1  60s\r
## 9      152000                  4  29.06464      8  1  60s\r
## 10     150000                  4  26.98884      8  1  60s\r

query2 <- "
SELECT speechiness, instrumentalness, energy
FROM music

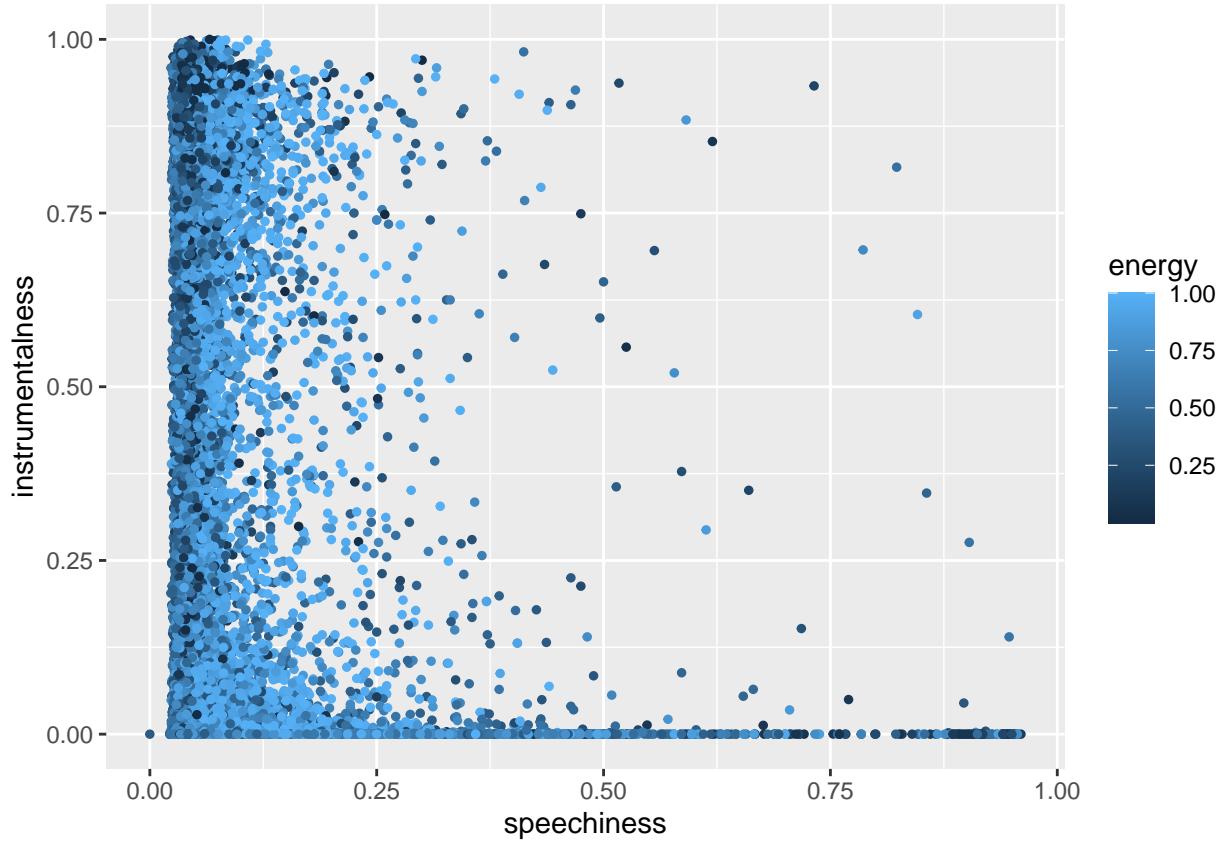
;

"

result2 <- dbGetQuery(con, query2)

ggplot(data = result2, mapping = aes(x = speechiness, y = instrumentalness, col = energy)) +
  geom_point(size = 1)

```



Most of the tracks displaying a high degree of speechiness have 0 instrumentalness, probably indicating they are audio books, podcasts, spoken word, etc.

There are a few outliers that have both high speechiness and instrumentalness (> 0.5 for both) which is interesting because from the variable descriptions these seem mutually exclusive.

More energetic tracks clustered more around the extremes of instrumental and speechiness?

Let's take a closer look at the outliers:

```
query3 <- "
SELECT artist, track, speechiness, instrumentalness
FROM music
WHERE speechiness > 0.5 AND instrumentalness > 0.5
ORDER BY speechiness DESC

;

"

result3 <- dbGetQuery(con, query3)
result3

##          artist                      track
## 1      Traditional                  Clowns
## 2 Natural Sounds    Divine Protection
## 3 Morton Subotnick   "Touch, Pt. 1"
## 4           Iasos            Lagoon Night
## 5 Daniel Johnston  I Am A Baby (In My Universe)
## 6 Karl-Heinz Schäfer  L'agresseur agressé
```

```

## 7           Black Asteroid          Turbine
## 8           Joe Mcphee            Improvisation 7
## 9           Karlheinz Stockhausen Klavierstück III
## 10          American Symphony Orchestra Hermit's Bell Overture Written by Maillart
##       speechiness instrumentalness

## 1           0.846             0.604
## 2           0.823             0.816
## 3           0.786             0.697
## 4           0.732             0.933
## 5           0.620             0.853
## 6           0.591             0.884
## 7           0.578             0.520
## 8           0.556             0.696
## 9           0.525             0.557
## 10          0.517             0.937

```

Listening to these tracks, most have very little actual vocalization but are generally noisy. It's possible Spotify's algorithm is mistaking some of the cacophonous sounds as voices.

```

query4 <- "
SELECT CASE WHEN song_key = 0 THEN 'C'
            WHEN song_key = 1 THEN 'C#/Db'
            WHEN song_key = 2 THEN 'D'
            WHEN song_key = 3 THEN 'D#/Eb'
            WHEN song_key = 4 THEN 'E'
            WHEN song_key = 5 THEN 'F'
            WHEN song_key = 5 THEN 'F#/Gb'
            WHEN song_key = 6 THEN 'G'
            WHEN song_key = 7 THEN 'G#/Ab'
            WHEN song_key = 8 THEN 'A'
            WHEN song_key = 9 THEN 'A#/Bb'
            WHEN song_key = 10 THEN 'B'
            ELSE 'Unknown Key'
        END,
        CASE WHEN song_mode = 0 THEN 'minor'
            WHEN song_mode = 1 THEN 'major'
            ELSE 'Unknown Mode'
        END
FROM music

;

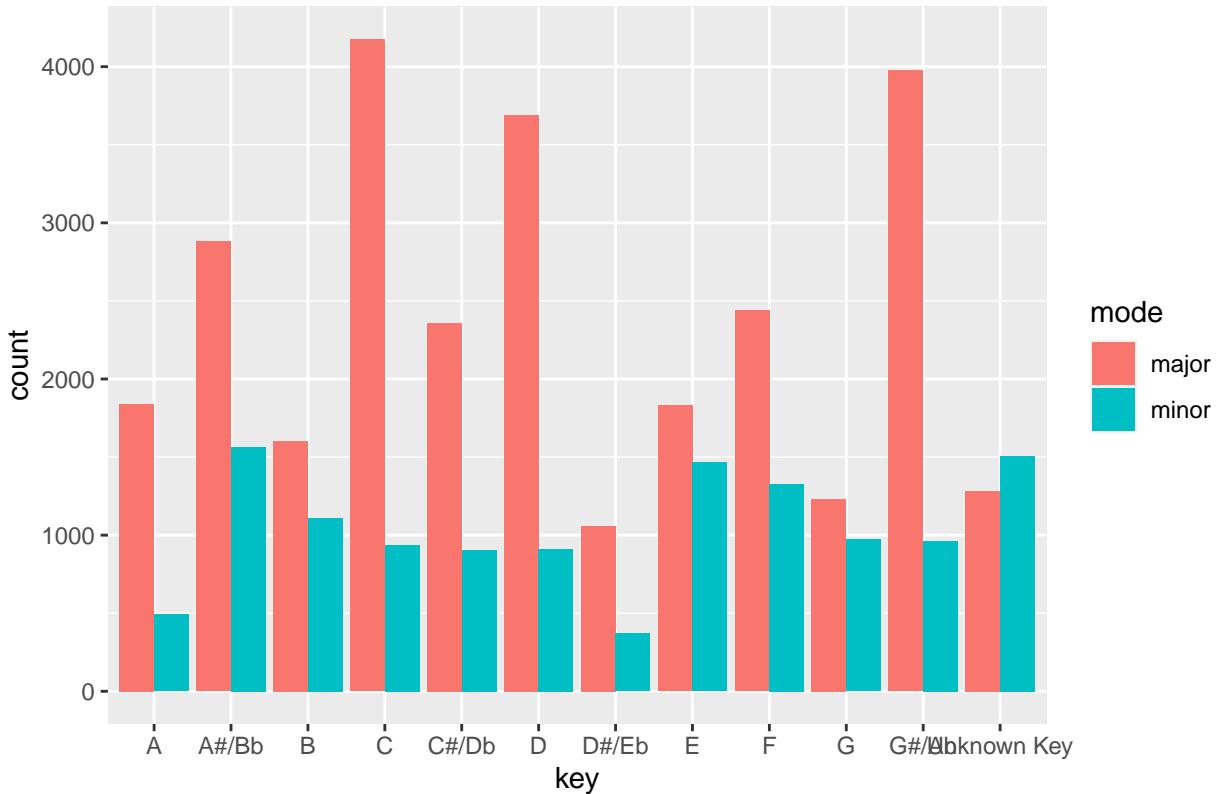
"

result4 <- dbGetQuery(con, query4)
names(result4) <- c("key", "mode")

ggplot(data = result4, aes(x = key, group = mode, fill = mode, stat = "count")) +
  geom_bar(position = "dodge") +
  labs(title = "Frequency of Keys Assigned by Spotify")

```

Frequency of Keys Assigned by Spotify



Spotify assigned most songs to a major key which makes sense because most songs are written in major keys. C, D, and G#/Ab (probably Ab, G# is awkward) major are the most commonly found keys. Generally keys with less accidentals (#/b) keys are more popular—two of the most popular keys, C major and D major, have 0 and 2 respectively.

Around 2800 tracks are of unknown key, or about 7% of the database.

```
query5 <- "
SELECT
    decade
    , danceability
    , liveness
    , duration_ms / 60000 AS duration -- Convert to minutes
FROM music

;

"

query6 <- "
SELECT
    decade
    , AVG(duration_ms / 60000) AS avg_duration -- Convert to minutes
    , STD(duration_ms / 60000) AS sd_duration
FROM music
GROUP BY decade

;"
```

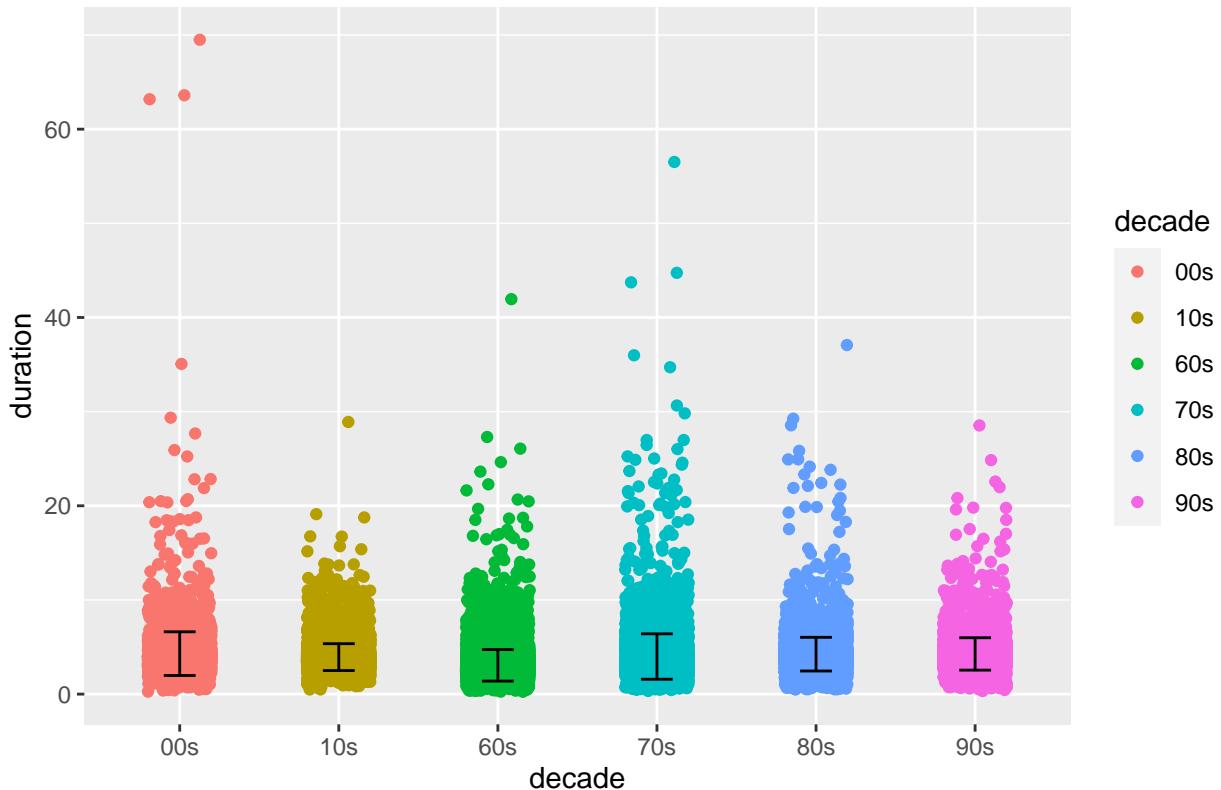
```

result5 <- dbGetQuery(con, query5)
result6 <- dbGetQuery(con, query6)

ggplot(data = result5, aes(x = decade, y = duration, color = decade)) +
  geom_point(position = position_jitter(w = 0.2)) +
  #geom_errorbar(aes(ymax = duration, ymin = duration)) +
  geom_errorbar(data = result6, mapping = aes(x = decade, y = avg_duration,
                                              ymin = avg_duration - sd_duration,
                                              ymax = avg_duration + sd_duration
                                              ),
                color = 'black', width = 0.2) +
  labs(title = "Track Duration by Decade")

```

Track Duration by Decade



```

# Disconnect from database to clean up connection
dbDisconnect(con)

```