Exploring Spotify Data

Christy Hui

12/14/2021

Introduction

When I first started listening to music, I wondered what makes a song good? Why do I like this song more than another? Why do I like this song while another person dislikes it? What aspects of a song make it popular? Why do some songs seem to be popular in one decade, and then non-popular the next?

I found a Spotify data set online that used the Spotify Web API to describe songs. And although the data set does not rank each song in terms of popularity, it does rank the artist's popularity. What trends can I find?

The data set can be found here on Kaggle:

https://www.kaggle.com/subhaskumarray/spotify-tracks-data?select=tracks.csv

For the sake of this project, the data was downloaded in January 2022, with Version 1 (the initial release) being the latest version of the data set. July 25, 2021 seems to be the version of the data set we use in this project.

Further information about the columns can be found here:

https://developer.spotify.com/documentation/web-api/reference/#/operations/get-audio-features

https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-tracks

What does the data look like?

Pre-Processing Data and a Closer Look

```
tracks = read.csv("tracks.csv")
```

```
names(tracks)
```

```
##  [1] "id"               "name"             "popularity"       "duration_ms"
##  [5] "explicit"         "artists"          "id_artists"       "release_date"
##  [9] "danceability"     "energy"           "key"              "loudness"
## [13] "mode"             "speechiness"      "acousticness"     "instrumentalness"
## [17] "liveness"         "valence"          "tempo"            "time_signature"
```

```
dim(tracks)
```

```
## [1] 586672     20
```

```
sum(is.na(tracks))
```

```
## [1] 0
```

The column names are as listed above.

By looking at the dimensions, we can see there are 586672 observations. This means there is information on 586672 songs.

How many of the columns are numeric? How many are characters? We can do this by iterating the typeof() function on each column of the tracks data set. We initialize the column to be a character list of 20, just to help R speed up the process a bit.

```
column_types = character(20L)
for (i in 1:ncol(tracks)) {
  column_types[i] = typeof(tracks[, i])
}
column_types
```

```
##  [1] "character" "character" "integer"   "integer"   "integer"   "character"
##  [7] "character" "character" "double"    "double"    "integer"   "double"
## [13] "integer"   "double"    "double"    "double"    "double"    "double"
## [19] "double"    "integer"
```

The data set is seems to be easy to work with because there are only 5 character types. The rest, however, are integers. In fact, for our analysis, the "id" of a song is irrelevant to how well an artist does (and it is also undecipherable to our eyes). The id of the artist is also irrelevant and undecipherable. Thus, we should delete the id and id_artist variables.

```
tracks = tracks[, !names(tracks) %in% c("id", "id_artists")] # drop column names by their names
names(tracks)
```

```
##  [1] "name"             "popularity"       "duration_ms"      "explicit"
##  [5] "artists"          "release_date"     "danceability"     "energy"
##  [9] "key"              "loudness"         "mode"             "speechiness"
## [13] "acousticness"     "instrumentalness" "liveness"         "valence"
## [17] "tempo"            "time_signature"
```

One last thing: I'm interested in the year that each song is released (rather than the full release date). Since the date is so easily formatted in YYYY-MM-DD format, we can use the substr() function to extract the year.

```
tracks$release_year = as.numeric(substr(tracks$release_date, 1, 4))
```

For the sake of consistency, let's reiterate which columns are numeric and which are categorical.

```
column_types = character(19L)
for (i in 1:ncol(tracks)) {
  column_types[i] = typeof(tracks[, i])
}
column_types
```

```
##  [1] "character" "integer"   "integer"   "integer"   "character" "character"
##  [7] "double"    "double"    "integer"   "double"    "integer"   "double"
## [13] "double"    "double"    "double"    "double"    "double"    "integer"
## [19] "double"
```

Numeric Variable Analysis

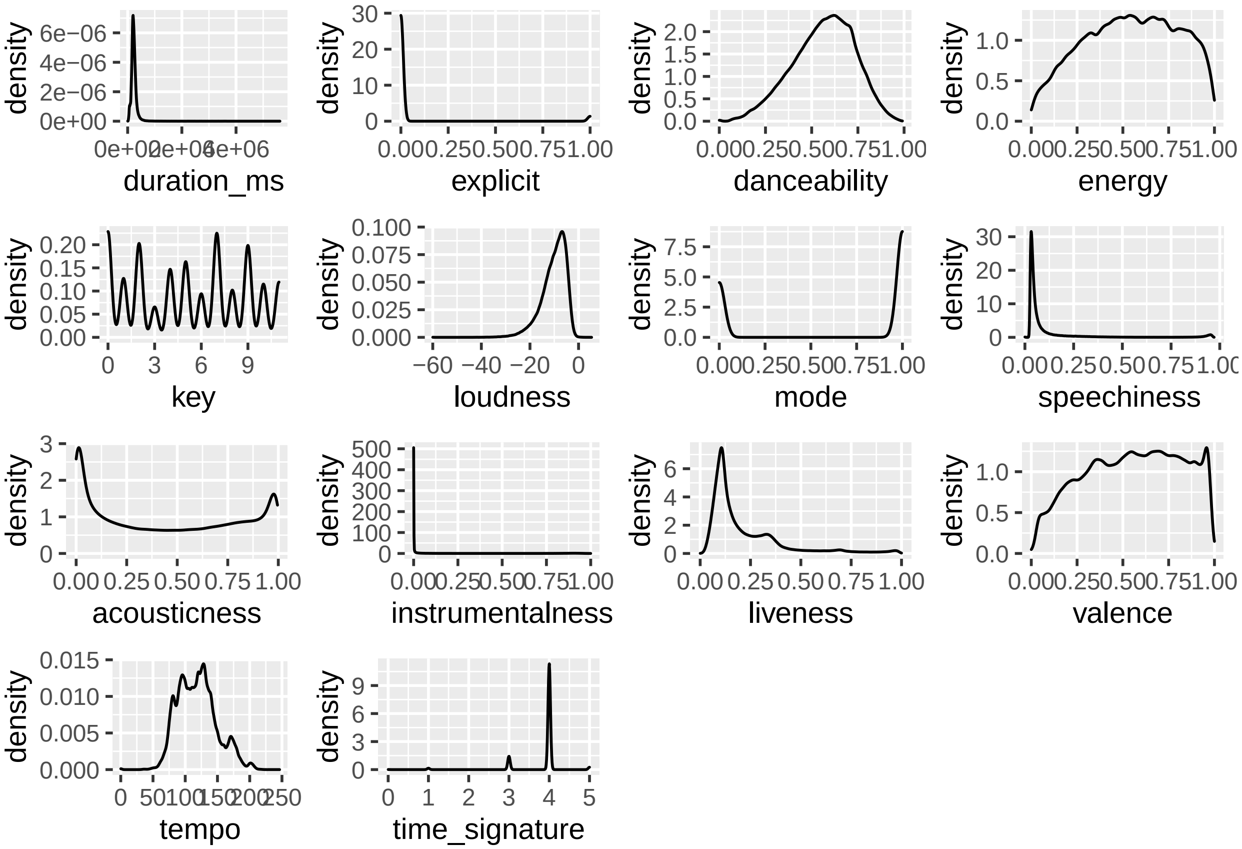
I am a bit curious about the variables themselves. It can be helpful to create a density plot for the numerical predictors. What are the numeric variables?

```
numeric_cols = names(tracks)[column_types != "character"]
numeric_cols
```

```
##  [1] "popularity"       "duration_ms"      "explicit"         "danceability"    
##  [5] "energy"           "key"              "loudness"         "mode"            
##  [9] "speechiness"      "acousticness"     "instrumentalness" "liveness"        
## [13] "valence"          "tempo"            "time_signature"   "release_year"
```

With the names, let's look at the density plots. What are the distributions of the numeric variables?

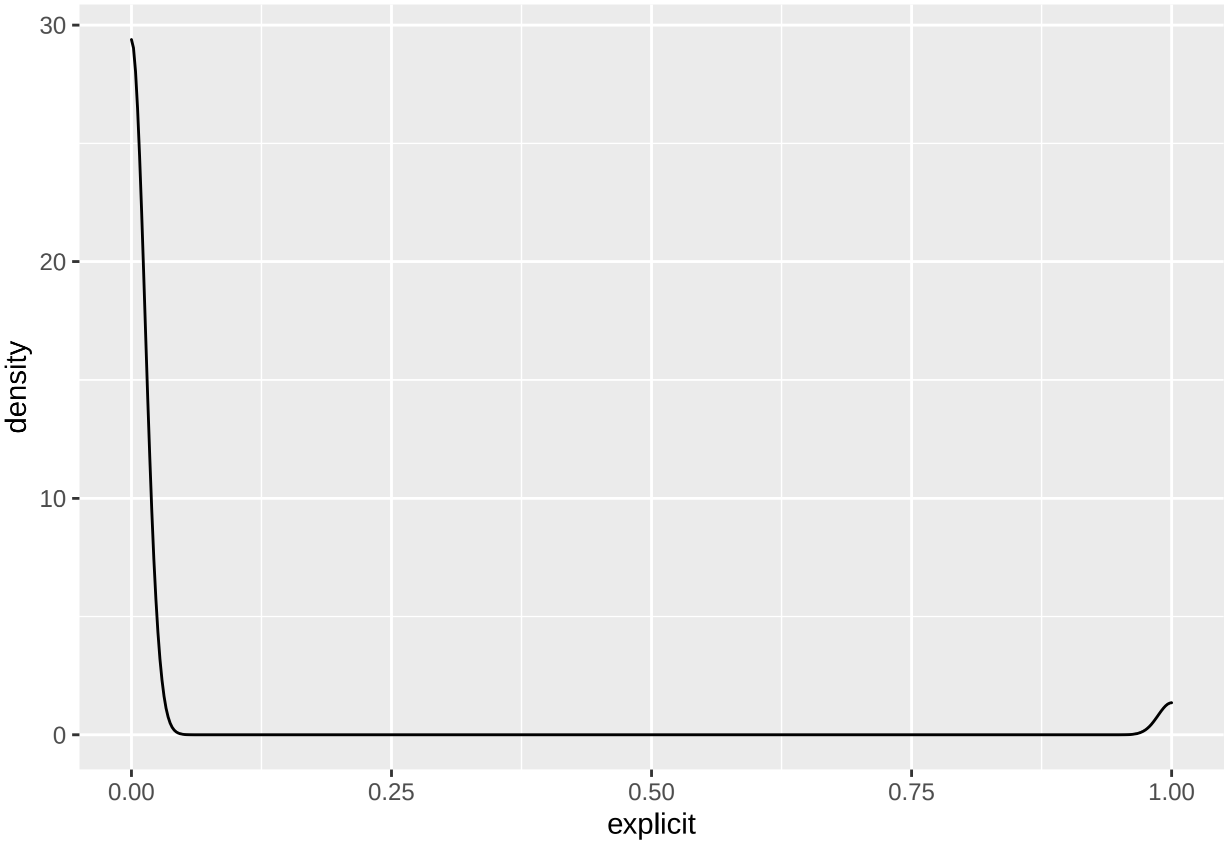
```
library(ggplot2)
library(gridExtra)
grid.arrange(
  ggplot(tracks, aes(x = duration_ms)) + geom_density(),
  ggplot(tracks, aes(x = explicit)) + geom_density(),
  ggplot(tracks, aes(x = danceability)) + geom_density(),
  ggplot(tracks, aes(x = energy)) + geom_density(),
  ggplot(tracks, aes(x = key)) + geom_density(),
  ggplot(tracks, aes(x = loudness)) + geom_density(),
  ggplot(tracks, aes(x = mode)) + geom_density(),
  ggplot(tracks, aes(x = speechiness)) + geom_density(),
  ggplot(tracks, aes(x = acousticness)) + geom_density(),
  ggplot(tracks, aes(x = instrumentalness)) + geom_density(),
  ggplot(tracks, aes(x = liveness)) + geom_density(),
  ggplot(tracks, aes(x = valence)) + geom_density(),
  ggplot(tracks, aes(x = tempo)) + geom_density(),
  ggplot(tracks, aes(x = time_signature)) + geom_density())
```



Looking at the density plots, it is clear that some of these variables aren't truly numeric (but actually categorical). Let's look at these weird ones first.

First is the "explicit" variable. The density plot seems to indicate that there are only 0s and 1s.

```
ggplot(tracks, aes(x = explicit)) + geom_density()
```



Upon closer inspection, this variable isn't numeric at all (but categorical).

```
unique(tracks$explicit)
```

```
## [1] 0 1
```

With only 0s and 1s, this column determines whether or not the song is explicit or not. 0s mean the song is not explicit and 1s mean the song is explicit.

So, "explicit" really should be categorical. Let's change it so that it does look categorical:

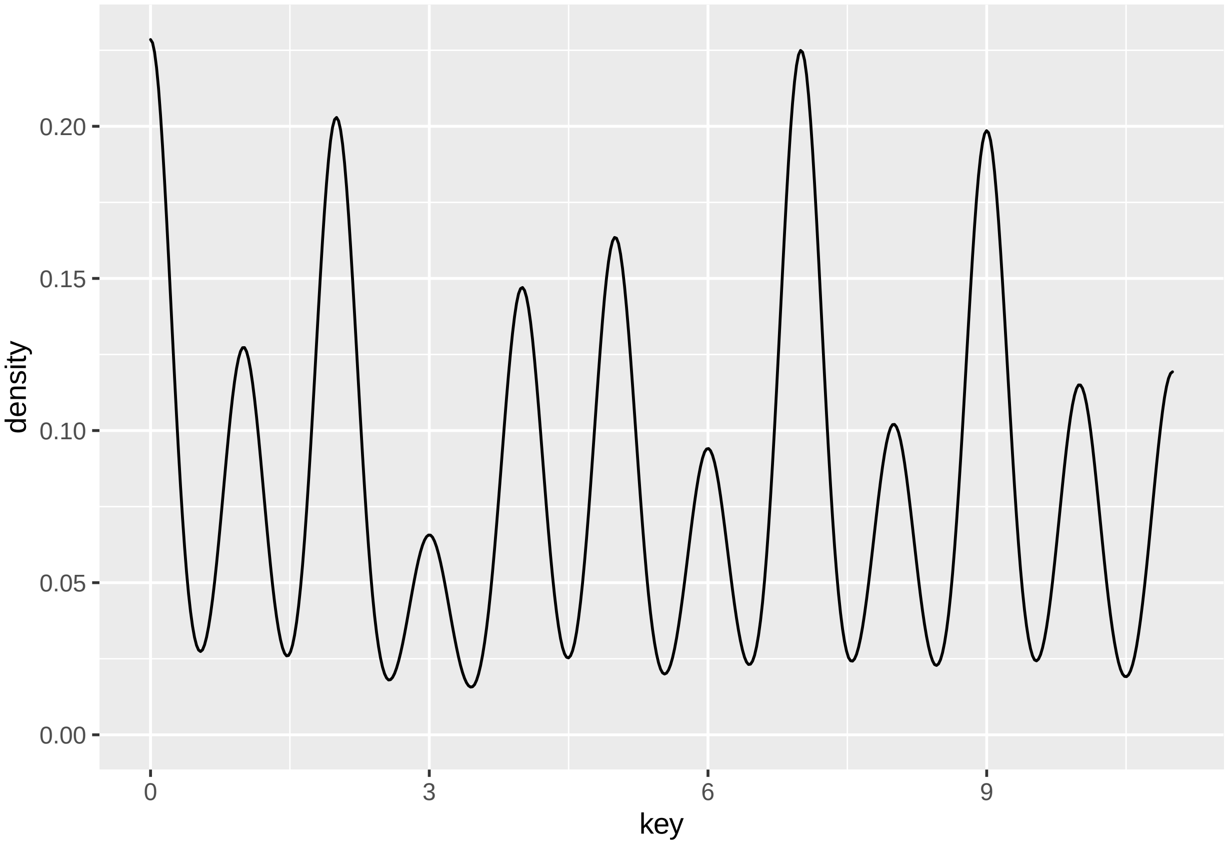
```
tracks$explicit = replace(tracks$explicit, tracks$explicit != 0, "Explicit")
tracks$explicit = replace(tracks$explicit, tracks$explicit == 0, "Not Explicit")
unique(tracks$explicit)
```

```
## [1] "Not Explicit" "Explicit"
```

```
tracks$explicit = as.factor(tracks$explicit) # factor so easier to work with later
```

The key variable's density plots also looks weird due to the vast amounts of oscillation.

```
ggplot(tracks, aes(x = key)) + geom_density()
```



Upon closer inspection, we also see that the “key” variable is categorical.

```
unique(tracks$key)
```

```
##  [1]  0  1  7  3  5  4  6 11  2  8 10  9
```

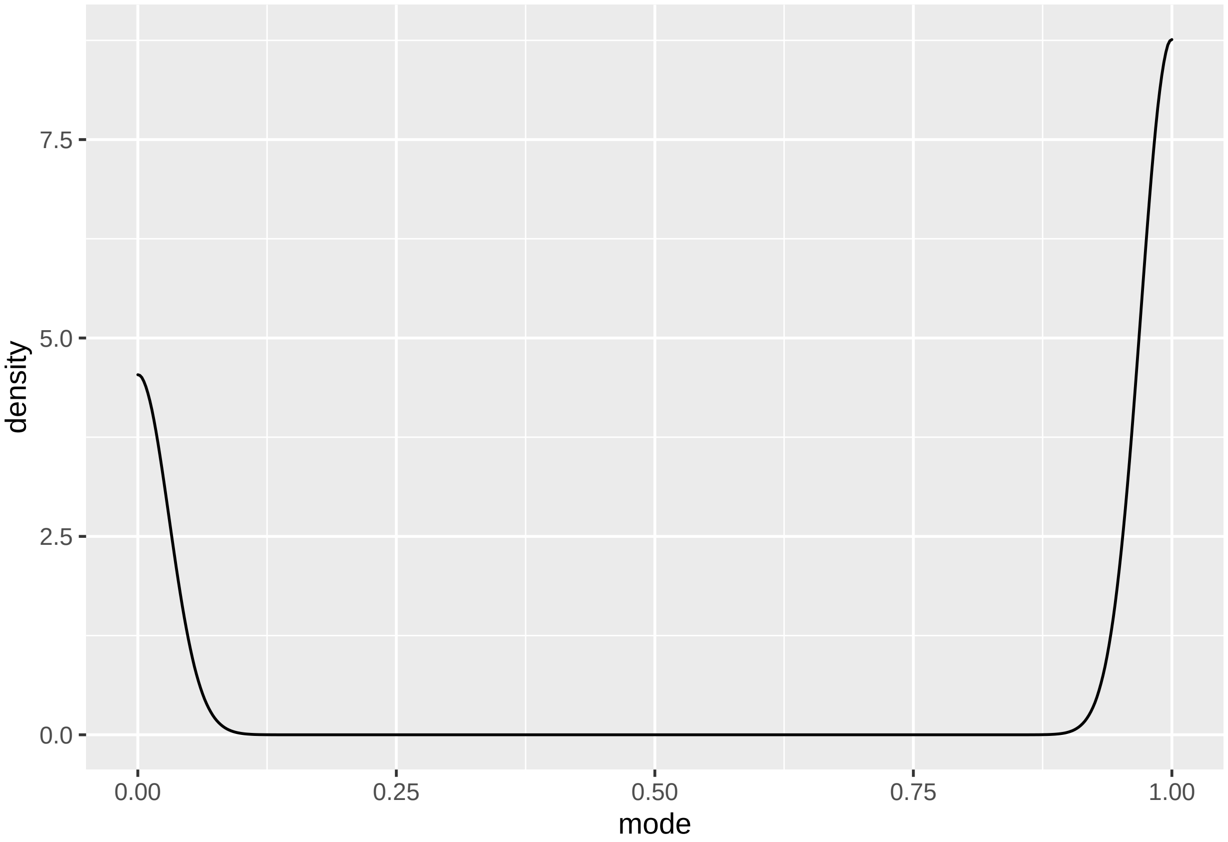
Per the Spotify web API reference webpage, the “key” represents the “key the track is in. Integers map to pitches using standard Pitch Class notation.” This is inherently categorical. We will keep the variables as is, but change the type to factors.

```
tracks$key = as.factor(tracks$key)
unique(tracks$key)
```

```
##  [1] 0  1  7  3  5  4  6  11 2  8  10 9
## Levels: 0 1 2 3 4 5 6 7 8 9 10 11
```

The density plot for the “mode” variable looks similar to the “explicit” variable. The density plot seems to indicate that only 0s and 1s are present.

```
ggplot(tracks, aes(x = mode)) + geom_density()
```



```
unique(tracks$mode)
```

```
## [1] 1 0
```

Just like the “explicit” variable, the “mode” variable is just a bunch of 0s and 1s. The reference site explains that mode “indicates the modality (major or minor) of a track... Major is represented by 1 and minor is 0.”

To make things perhaps more readable, let’s change the column:

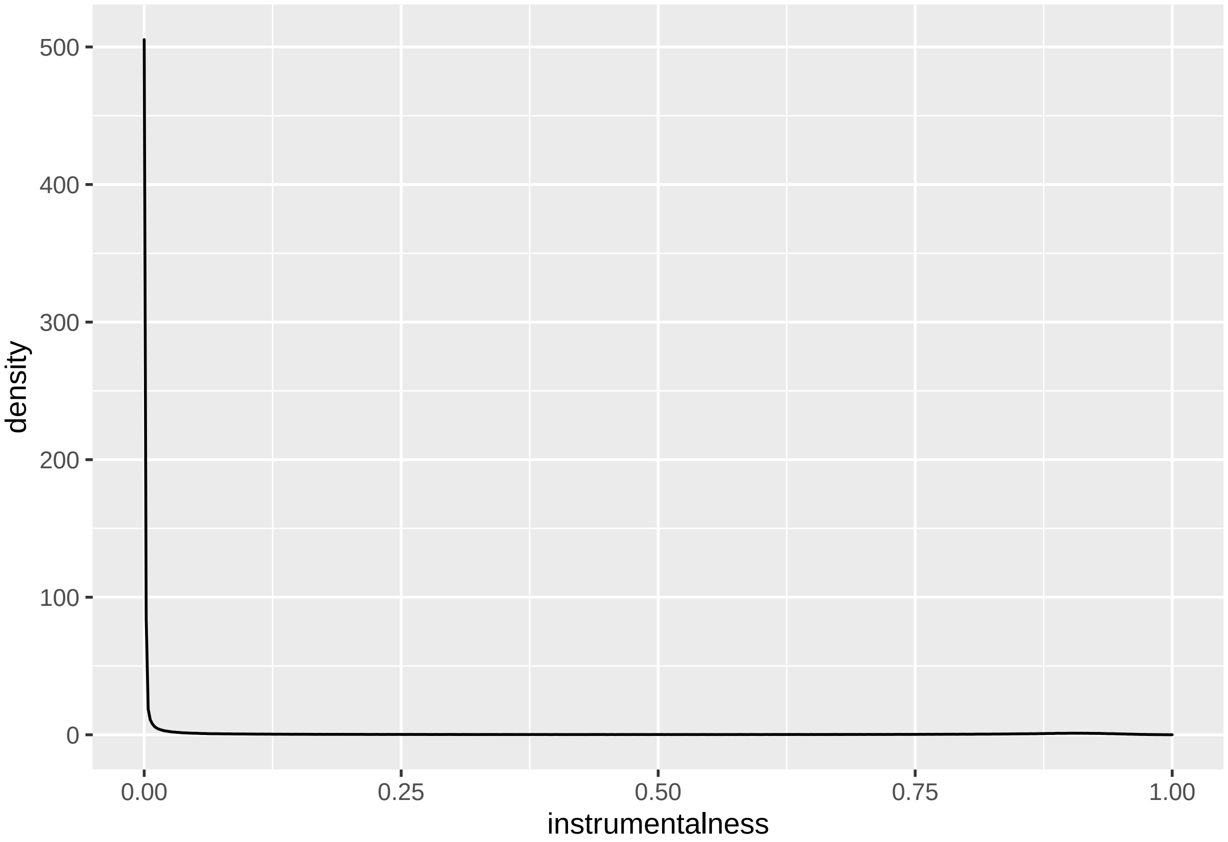
```
tracks$mode = replace(tracks$mode, tracks$mode != 0, "Major")
tracks$mode = replace(tracks$mode, tracks$mode == 0, "Minor")
unique(tracks$mode)
```

```
## [1] "Major" "Minor"
```

```
tracks$mode = as.factor(tracks$mode) # factor so easier to work with later
```

The “instrumentalness” value looks weird as well because it dips down to 0 so quickly.

```
ggplot(tracks, aes(x = instrumentalness)) + geom_density()
```



Upon closer review, the numbers for "instrumentalness" are between 0 and 1 (but still represent numbers):

```
head(tracks$instrumentalness)
```

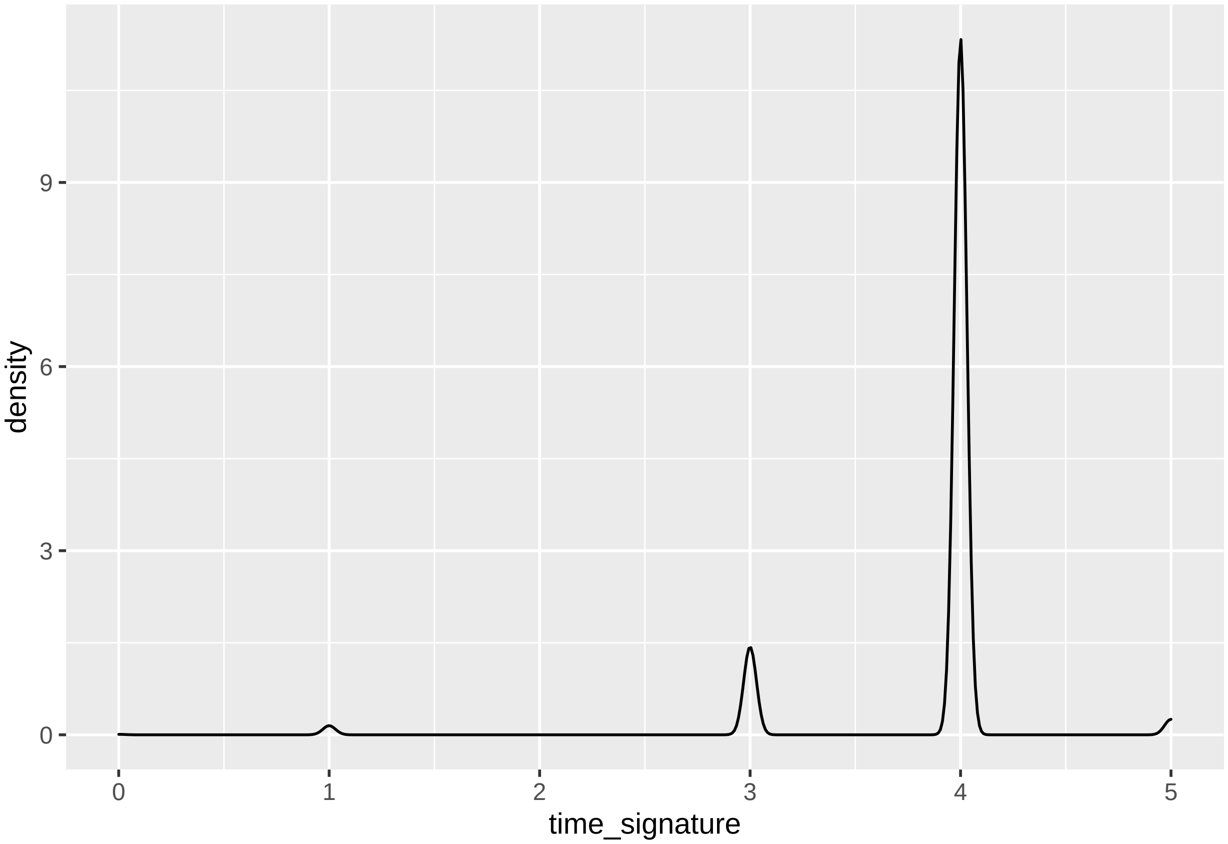
```
## [1] 0.7440 0.0000 0.0218 0.9180 0.1300 0.2470
```

The reference site states that the "instrumentalness" variable is a prediction number, predicting "whether a track contains no vocals… Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0."

Since this variable is numeric, it should not be changed. However, it should be noted that the density plot for the "instrumentalness" variable is not very helpful.

"Time_signature" is the final variable that looks abnormal.

```
ggplot(tracks, aes(time_signature)) + geom_density()
```



There only seem to be peaks at certain intervals, which may indicate that the time_signature variable is not numeric.

```
unique(tracks$time_signature)
```

```
## [1] 3 1 5 4 0
```

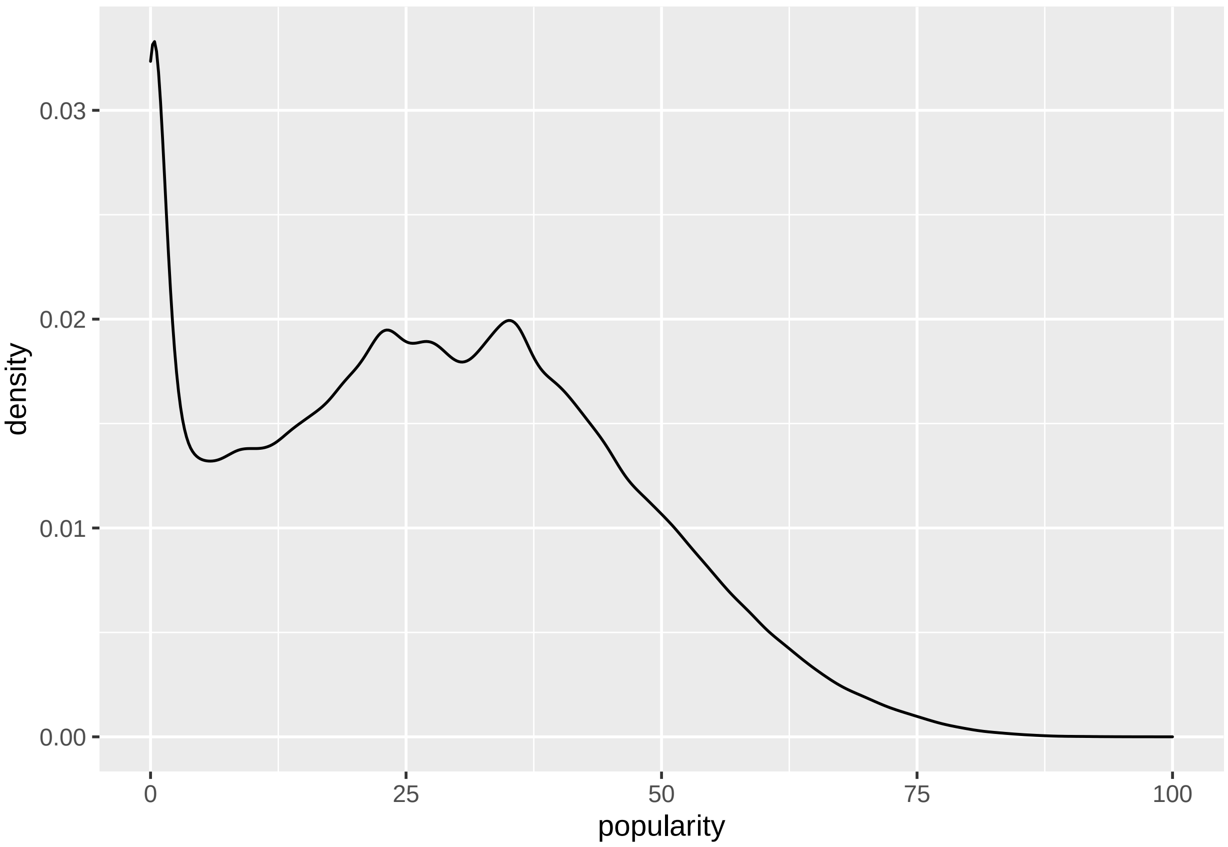
The reference site explains that the “time_signature” variable is “a notational convention to specify how many beats are in each bar (or measure).” So, this is numeric variable; however, because of the discrete nature of the variable, it looks categorical. Looking at the density plot, a large portion of the data has a time signature of 4. Interestingly, this seems to indicate that most songs have 4 beats per measure. Very few songs have 5 or 1 beats, and none of the songs have 2 beats. Some have 0 beats per measure (which seems fishy to me).

Looking closer, the reference site explains that the “time_signature” variable “ranges from 3 to 7 indicating time signatures of”3/4“, to”7/4“.” This is in direct opposition to the variables found (since we have numbers below 3 and do not have 6 or 7). This may indicate that the “time_signature” variable is inaccurate or the description is inaccurate. Nonetheless, both are inconsistent, meaning that we might have to refrain from looking at the variable.

To recap, we have learned that three variables (explicit, key, mode) were disguised at numeric; however, they should be viewed as categorical. Also, the “instrumentalness” variable is continuous and numeric, but because it is only between 0 and 1, it may look small compared to other variables. Finally, the “time_signature” variable is also numeric, but it is discrete, so it may look weird when compared to other numeric variables (of which are continuous). In fact, the numbers in “time_signature” is inconsistent with the Spotify reference web page, indicating that it may be good to not look at the variable when analyzing the data.

Now that we have looked at data that seem inconsistent/strange, we should take a quick look at the other numeric variables that seem right. Firstly is popularity, which is of particular interest to me.

```r
ggplot(tracks, aes(x = popularity)) + geom_density()
```



Looking at the density plot, there seem to be a lot of artists with a popularity of 0. Generally, it seems that as popularity increases, the number of artists decreases. Interestingly, however, popularity and density of artists rise at around approximately 15-35 before strictly decreasing from there. Looking at the numbers directly, I find it interesting that only very few artists have a popularity of 90-100.

```r
table(tracks$popularity)
```

```
## 
##     0     1     2     3     4     5     6     7     8     9    10    11    12 
## 44690 12024  9639  8154  7733  7730  7659  7726  7988  8265  8019  8098  7985 
##    13    14    15    16    17    18    19    20    21    22    23    24    25 
##  8512  8747  8725  9269  9137  9466 10324 10098 10534 11206 12139 11148 10876 
##    26    27    28    29    30    31    32    33    34    35    36    37    38 
## 10937 11292 11146 10835 10171 10414 10773 11174 11328 12231 11879 10808 10100 
##    39    40    41    42    43    44    45    46    47    48    49    50    51 
## 10063  9949  9594  9106  8789  8472  8120  7318  7067  6836  6536  6240  6023 
##    52    53    54    55    56    57    58    59    60    61    62    63    64 
##  5588  5264  4964  4586  4290  3906  3642  3496  2976  2806  2625  2359  2086 
##    65    66    67    68    69    70    71    72    73    74    75    76    77 
##  1910  1715  1479  1335  1230  1110   968   846   732   673   571   485   391 
##    78    79    80    81    82    83    84    85    86    87    88    89    90 
##   314   279   218   158   136   116    91    60    53    38    19    16    12 
##    91    92    93    94    95    96    97    98    99   100 
##    11    10     2     6     1     2     2     1     1     1
```

Who are these artists? And what are the song names?

```
tracks$artists[tracks$popularity %in% 90:100]
```

```
##  [1] "[’The Neighbourhood’]"
##  [2] "[’Doja Cat’]"
##  [3] "[’Harry Styles’]"
##  [4] "[’Lewis Capaldi’]"
##  [5] "[’The Weeknd’]"
##  [6] "[’Kali Uchis’]"
##  [7] "[’The Weeknd’]"
##  [8] "[’Tiësto’]"
##  [9] "[’Giveon’]"
## [10] "[’The Kid LAROI’]"
## [11] "[’Myke Towers’, ’Juhn’]"
## [12] "[’Bad Bunny’, ’ROSALÍA’]"
## [13] "[’SZA’]"
## [14] "[’Bad Bunny’, ’Jhay Cortez’]"
## [15] "[’Ariana Grande’]"
## [16] "[’Boza’]"
## [17] "[’MEDUZA’, ’Dermot Kennedy’]"
## [18] "[’BTS’]"
## [19] "[’Tate McRae’]"
## [20] "[’KAROL G’]"
## [21] "[’Ariana Grande’]"
## [22] "[’DaBaby’, ’Roddy Ricch’]"
## [23] "[’Joel Corry’, ’MNEK’]"
## [24] "[’Pop Smoke’]"
## [25] "[’Dua Lipa’, ’DaBaby’]"
## [26] "[’Billie Eilish’]"
## [27] "[’Bad Bunny’, ’Jhay Cortez’]"
## [28] "[’Cardi B’, ’Megan Thee Stallion’]"
## [29] "[’Ava Max’]"
## [30] "[’Justin Bieber’, ’Daniel Caesar’, ’Giveon’]"
## [31] "[’Olivia Rodrigo’]"
## [32] "[’Masked Wolf’]"
## [33] "[’Bruno Mars’, ’Anderson .Paak’, ’Silk Sonic’]"
## [34] "[’Los Legendarios’, ’Wisin’, ’Jhay Cortez’]"
## [35] "[’Riton’, ’Nightcrawlers’, ’Mufasa & Hypeman’, ’Dopamine’]"
## [36] "[’Rochy RD’, ’Myke Towers’, ’Nicki Nicole’]"
## [37] "[’Cardi B’]"
## [38] "[’Travis Scott’, ’HVME’]"
## [39] "[’Justin Bieber’]"
## [40] "[’Nathan Evans’, ’220 KID’, ’Billen Ted’]"
## [41] "[’Sech’]"
## [42] "[’ATB’, ’Topic’, ’A7S’]"
## [43] "[’Drake’]"
## [44] "[’Dua Lipa’]"
## [45] "[’Selena Gomez’, ’Rauw Alejandro’]"
## [46] "[’Maroon 5’, ’Megan Thee Stallion’]"
## [47] "[’Olivia Rodrigo’]"
## [48] "[’Justin Bieber’]"
## [49] "[’Saweetie’, ’Doja Cat’]"
```

```
tracks$name[tracks$popularity %in% 90:100]
```

```
##  [1] "Sweater Weather"
##  [2] "Streets"
##  [3] "Watermelon Sugar"
##  [4] "Someone You Loved"
##  [5] "Save Your Tears"
##  [6] "telepatía"
##  [7] "Blinding Lights"
##  [8] "The Business"
##  [9] "Heartbreak Anniversary"
## [10] "WITHOUT YOU"
## [11] "Bandido"
## [12] "LA NOCHE DE ANOCHE"
## [13] "Good Days"
## [14] "DÁKITI"
## [15] "positions"
## [16] "Hecha Pa’ Mi"
## [17] "Paradise (feat. Dermot Kennedy)"
## [18] "Dynamite"
## [19] "you broke me first"
## [20] "BICHOTA"
## [21] "34+35"
## [22] "ROCKSTAR (feat. Roddy Ricch)"
## [23] "Head & Heart (feat. MNEK)"
## [24] "What You Know Bout Love"
## [25] "Levitating (feat. DaBaby)"
## [26] "Therefore I Am"
## [27] "DÁKITI"
## [28] "WAP (feat. Megan Thee Stallion)"
## [29] "My Head & My Heart"
## [30] "Peaches (feat. Daniel Caesar & Giveon)"
## [31] "drivers license"
## [32] "Astronaut In The Ocean"
## [33] "Leave The Door Open"
## [34] "Fiel"
## [35] "Friday (feat. Mufasa & Hypeman) - Dopamine Re-Edit"
## [36] "Ella No Es Tuya - Remix"
## [37] "Up"
## [38] "Goosebumps - Remix"
## [39] "Hold On"
## [40] "Wellerman - Sea Shanty / 220 KID x Billen Ted Remix"
## [41] "911"
## [42] "Your Love (9PM)"
## [43] "What’s Next"
## [44] "We’re Good"
## [45] "Baila Conmigo (with Rauw Alejandro)"
## [46] "Beautiful Mistakes (feat. Megan Thee Stallion)"
## [47] "deja vu"
## [48] "Anyone"
## [49] "Best Friend (feat. Doja Cat)"
```

In fact, maybe we can find out more. What about the other columns? Maybe we can find a trend that makes these songs more popular than the rest.

```
head(tracks[tracks$popularity %in% 90:100, ])
```

```
##                    name popularity duration_ms     explicit
## 86017   Sweater Weather         90      240400 Not Explicit
## 91867           Streets         94      226987     Explicit
## 91868  Watermelon Sugar         92      174000 Not Explicit
## 91870 Someone You Loved         90      182161 Not Explicit
## 92811   Save Your Tears         97      215627     Explicit
## 92812         telepatía         97      160191 Not Explicit
##                     artists release_date danceability energy key loudness  mode
## 86017 [’The Neighbourhood’]   2013-04-19        0.612  0.807  10   -2.810 Major
## 91867          [’Doja Cat’]   2019-11-07        0.749  0.463  11   -8.433 Major
## 91868      [’Harry Styles’]   2019-12-13        0.548  0.816   0   -4.209 Major
## 91870     [’Lewis Capaldi’]   2019-05-17        0.501  0.405   1   -5.679 Major
## 92811        [’The Weeknd’]   2020-03-20        0.680  0.826   0   -5.487 Major
## 92812        [’Kali Uchis’]   2020-12-04        0.653  0.524  11   -9.016 Minor
##       speechiness acousticness instrumentalness liveness valence   tempo
## 86017      0.0336       0.0495         1.77e-02    0.101   0.398 124.053
## 91867      0.0828       0.2080         3.71e-02    0.337   0.190  90.028
## 91868      0.0465       0.1220         0.00e+00    0.335   0.557  95.390
## 91870      0.0319       0.7510         0.00e+00    0.105   0.446 109.891
## 92811      0.0309       0.0212         1.24e-05    0.543   0.644 118.051
## 92812      0.0502       0.1120         0.00e+00    0.203   0.553  83.970
##       time_signature release_year
## 86017              4         2013
## 91867              4         2019
## 91868              4         2019
## 91870              4         2019
## 92811              4         2020
## 92812              4         2020
```

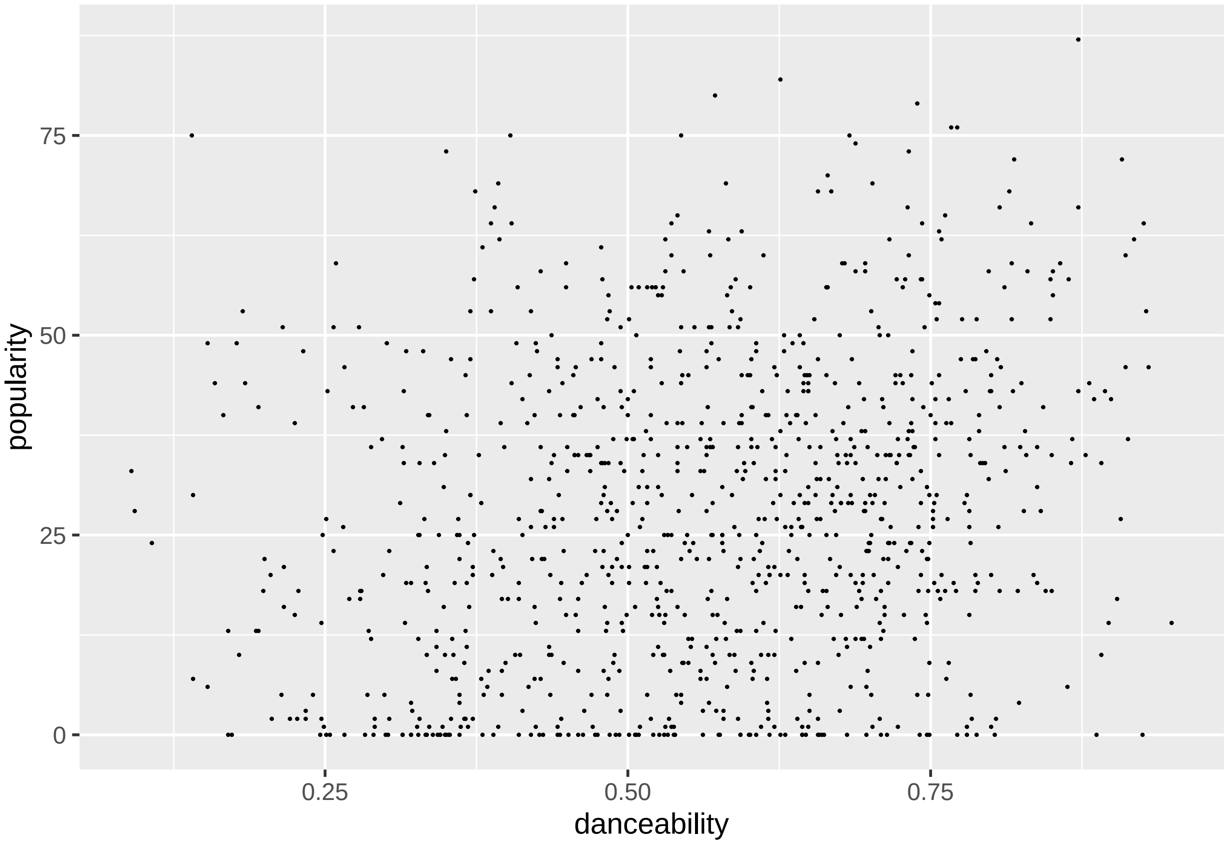
This isn’t helpful. Maybe plots can be more helpful. How is popularity related to the variables of, for example, danceability and energy?

Before we observe the plot, note that we have way too many observations for a simple plot to be useful. We wouldn’t be able to tell what’s happening with 500000+ data points. Let’s shrink the data by sampling 1000 random observations and then look at the plots.

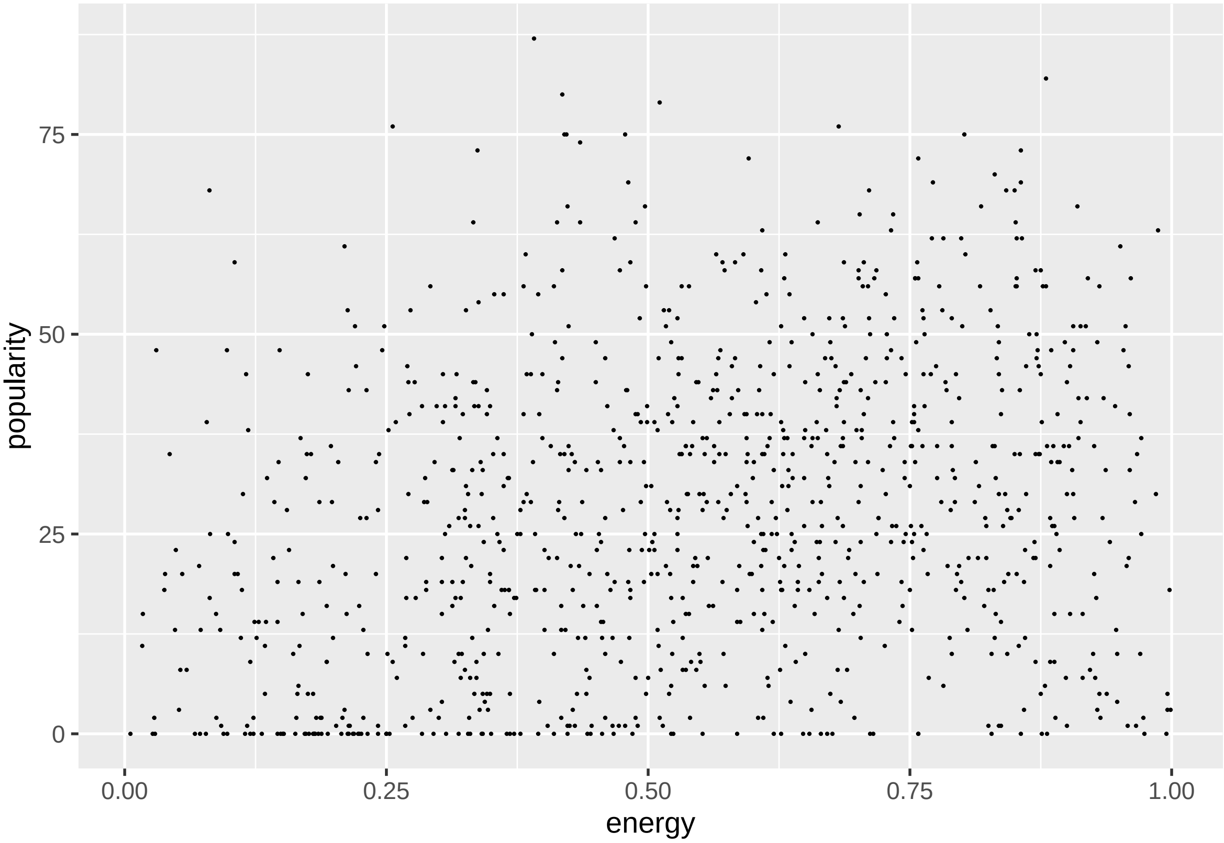
```
set.seed(213)
tracks_sampled = tracks[sample(dim(tracks)[1], 1000, replace = FALSE), ]
dim(tracks_sampled)
```

```
## [1] 1000   19
```

```
ggplot(tracks_sampled, aes(x = danceability, y = popularity)) + geom_point(size = 0.1)
```



```
ggplot(tracks_sampled, aes(x = energy, y = popularity)) + geom_point(size = 0.1)
```



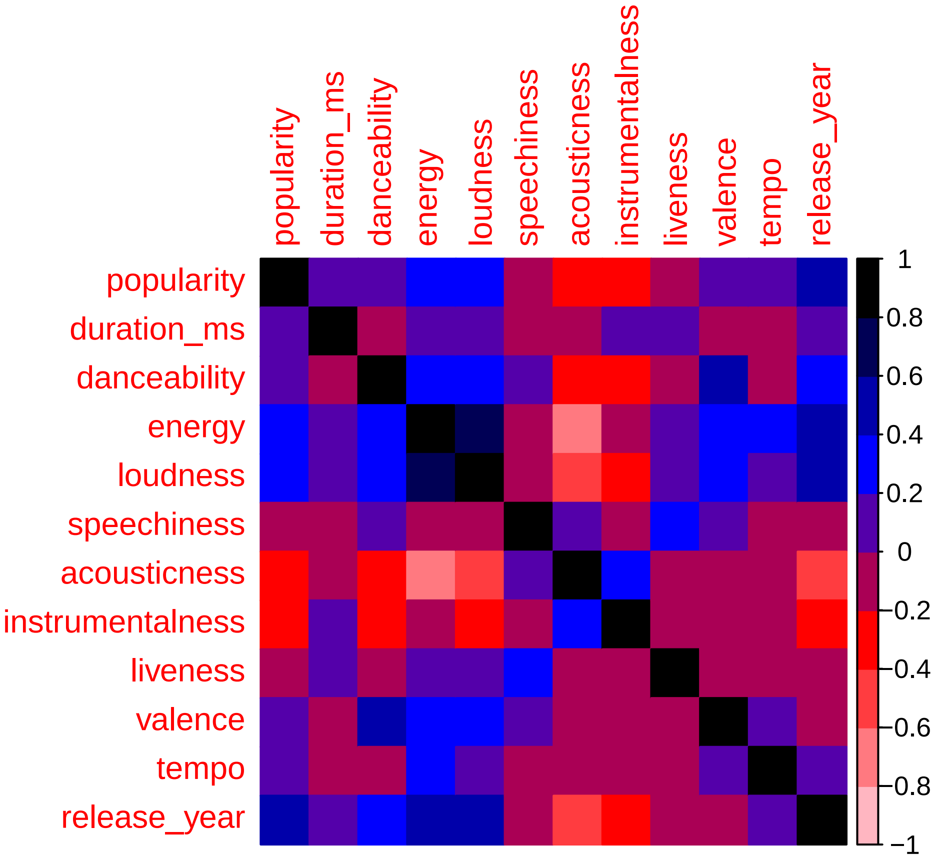
There doesn't really seem to be much of a correlation between danceability and popularity. Between popularity and energy, however, there seems to be a slight positive correlation.

In fact, it may be useful to look at the correlation between all variables. Let's create a heatmap of the relevant numeric variables.

```
tracks_numeric = tracks[, c("popularity", "duration_ms", "danceability", "energy", "loudness", "speechi
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor(tracks_numeric),
         method = "color",
         col= colorRampPalette(c("lightpink","red", "blue", "black"))(10))
```



Notice how the correlation both the correlation between danceability and popularity is close to 0. The correlaion between energy and popularity, however, is a little higher than 0 (indicating a positive correlation between the two).

Upon closer inspection, I see a lot more in this heat map worth noting. For example, there seems to be a very high correlation between loudness and energy. This makes sense, since the louder a song is, the more energy they seem to eminate. Acousticness and energy seem to have a very low correlation. This also makes sense, since the more acoustic a song is, the less energy the song seems to give.