Untitled

Mei Gilhousen

10/12/2022

Install necessary packages and functions. Attach to Spotify API to retrive data.

```
library(httr)
library(jsonlite)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggExtra)
```

```
library(spotifyr)
Sys.setenv(SPOTIFY_CLIENT_ID = '4b6653925c2c4642b3d434ae599e7ca0')
Sys.setenv(SPOTIFY_CLIENT_SECRET = 'ffcf9dcb2c704e7c8186da25022ab60c')

access_token <- get_spotify_access_token()
top100 <- get_playlist_tracks('6UeSakyzhiEt4NB3UAd6NQ')
songs <- get_track_audio_features(top100$track.id, authorization = get_spotify_access_token())
songs$name <- top100$track.name
songs
```

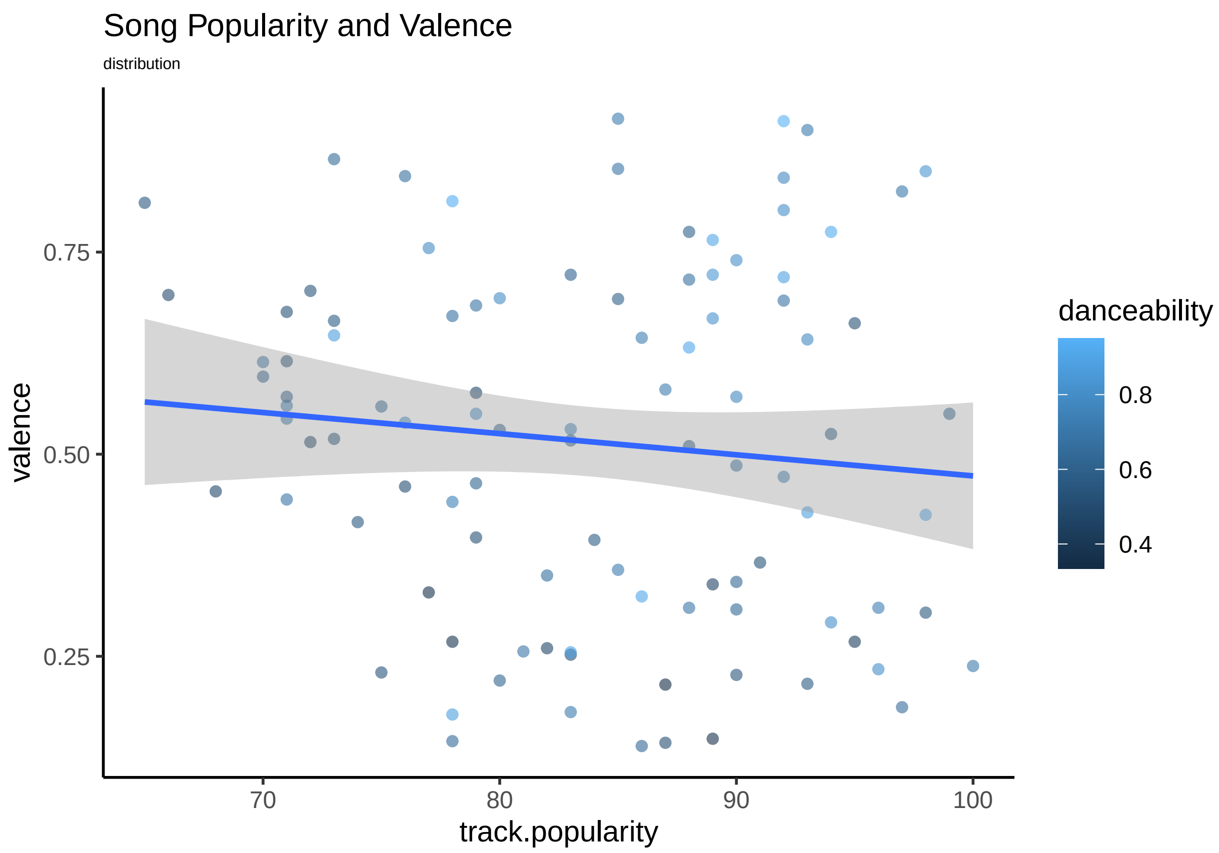
```
## # A tibble: 100 x 19
##    danceabi~1 energy   key loudn~2  mode speec~3 acous~4 instr~5 liven~6 valence
##         <dbl>  <dbl> <int>   <dbl> <int>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1      0.686  0.507     1   -7.10     1  0.0357  0.626  7.51e-5  0.383    0.69
##  2      0.714  0.472     2   -7.38     1  0.0864  0.013  4.51e-6  0.266    0.238
##  3      0.52   0.731     6   -5.34     0  0.0557  0.342  1.01e-3  0.311    0.662
##  4      0.733  0.67      5   -6.01     1  0.0751  0.121  0         0.121    0.472
##  5      0.768  0.714    10   -5.11     1  0.0401  0.352  0         0.15     0.842
##  6      0.704  0.797     0   -5.93     1  0.0475  0.0826 7.45e-4  0.0546   0.825
##  7      0.731  0.85      9   -4.91     1  0.0336  0.252  0         0.605    0.644
##  8      0.95   0.891     2   -2.65     1  0.241   0.0645 1.77e-5   0.309    0.912
##  9      0.608  0.745     1   -4.13     1  0.0277  0.0226 6.47e-6   0.0942   0.464
## 10      0.836  0.743    10   -6.30     0  0.0656  0.0995 0         0.335    0.722
## # ... with 90 more rows, 9 more variables: tempo <dbl>, type <chr>, id <chr>,
## #   uri <chr>, track_href <chr>, analysis_url <chr>, duration_ms <int>,
## #   time_signature <int>, name <chr>, and abbreviated variable names
## #   1: danceability, 2: loudness, 3: speechiness, 4: acousticness,
## #   5: instrumentalness, 6: liveness
```

Variables are displayed above. Top 100 songs at the time using the Spotify Top 100 Billboard is linked to the variables retrieved through API.

```
total <- cbind(songs, top100)
```

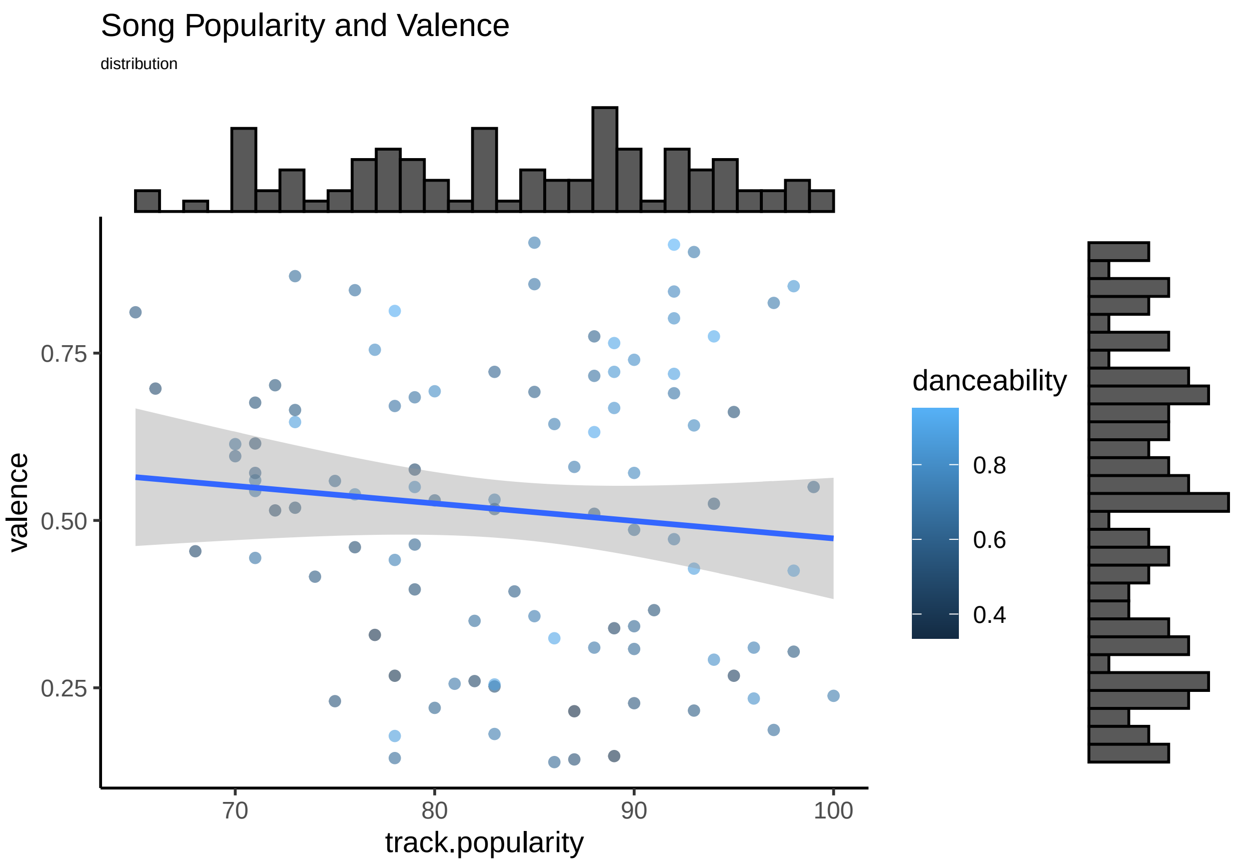
```
graph1<-ggplot(total, aes(track.popularity, valence ,color=danceability))+
   geom_point(alpha = .6)+
  geom_smooth(method="lm")+
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
    theme(panel.background = element_blank())+
   theme(axis.line = element_line(colour ="black"))+
   ggtitle("Song Popularity and Valence",
       subtitle="distribution")+
   theme(plot.title = element_text(hjust = 0.0))+
 theme(plot.title = element_text(size=12))+
 theme(plot.subtitle = element_text(size=6))


graph1
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



```
ggMarginal(graph1, type="histogram")
```

```
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
```



```r
mysample <- total[sample(1:nrow(total), 20,
   replace=FALSE),]
as.numeric(mysample$danceability)
```

```
##  [1] 0.524 0.520 0.723 0.542 0.801 0.720 0.667 0.479 0.505 0.714 0.582 0.665
## [13] 0.608 0.582 0.557 0.709 0.820 0.881 0.369 0.575
```

```r
as.numeric(mysample$energy)
```

```
##  [1] 0.643 0.751 0.772 0.820 0.806 0.715 0.750 0.768 0.657 0.472 0.568 0.666
## [13] 0.745 0.525 0.843 0.357 0.686 0.592 0.192 0.842
```

```r
as.numeric(mysample$valence)
```

```
##  [1] 0.397 0.519 0.580 0.702 0.802 0.357 0.486 0.576 0.252 0.238 0.394 0.844
## [13] 0.464 0.510 0.811 0.544 0.668 0.719 0.148 0.665
```

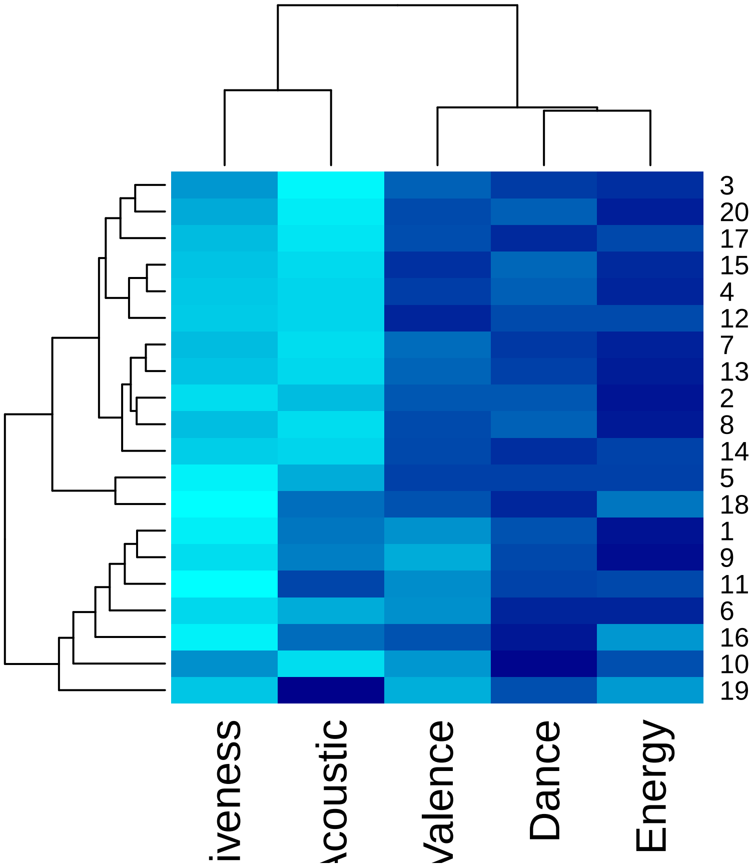
```r
as.numeric(mysample$liveness)
```

```
##  [1] 0.2210 0.0624 0.3670 0.0561 0.1140 0.1070 0.1970 0.1210 0.1260 0.2660
## [11] 0.1050 0.0841 0.0942 0.1340 0.1090 0.1120 0.1840 0.0901 0.0954 0.2790
```

```
as.numeric(mysample$acousticness)
```

```
##  [1] 4.51e-01 1.71e-01 5.24e-03 6.82e-04 3.82e-01 2.58e-01 7.46e-02 7.02e-05
##  [9] 3.73e-01 1.30e-02 5.77e-01 3.31e-02 2.26e-02 1.11e-01 1.32e-03 4.78e-01
## [17] 4.12e-03 6.19e-01 5.55e-01 6.83e-03
```

```
Dance<-mysample$danceability
Energy<-mysample$energy
Valence<-mysample$valence
Liveness<-mysample$liveness
Acoustic<-mysample$acousticness

newdata<-cbind(Dance, Energy, Valence, Liveness, Acoustic)
my_colors <- colorRampPalette(c("cyan", "darkblue"))

map<-heatmap(newdata,col=my_colors(100))
```



```
map
```

```
## $rowInd
##  [1] 19 10 16  6 11  9  1 18  5 14  8  2 13  7 12  4 15 17 20  3
##
## $colInd
## [1] 4 5 3 1 2
```

```
##
## $Rowv
## NULL
##
## $Colv
## NULL
```