Top_song_analysis.R

Apurva Sarode

2020-04-16

```
library(readr)
library(dplyr)

library(ggplot2)

library(ggridges)

library(highcharter)

library(plyr)

library(lubridate)

library(fmsb)
library(gridExtra)

library(cmna)

library(tidyselect)
library(factoextra)

library(psych)

library(gvlma)
library(MASS)

library(NbClust)
library(GGally)

library(car)

#--------------------Data Preparation------------------------------#


top10s <- read.csv("C:\\Users\\Apurva
Sarode\\Desktop\\Spotify_mva.csv",header = TRUE)
View(top10s)
Data <- top10s
```

```r
#----------------------Data Cleaning-------------------------------#
#finding missing data
dim(Data)
```

```
## [1] 603  15
```

```r
any(Data$bpm==0)
```

```
## [1] TRUE
```

```r
any(Data$pop==0)
```

```
## [1] TRUE
```

```r
Data = filter(Data, bpm != 0)
Data = filter(Data, pop != 0)
dim(Data)
```

```
## [1] 598  15
```

```r
#reordering columns
Data <- Data[,c(1,2,3,4,5,6,12,7,8,9,10,11,13,14,15)]
View(Data)

#speechiness also include podcast and speeches
#For songs speechiness will be low and inaccurate, hence removing spch
Data$spch <- NULL
#Liveness includes lives shows which are also inaccurate to test songs
#in a recording studio, Hence removing live
Data$live <- NULL

#Renaming columns to a more readable format
colnames(Data)[4] <- "Genre"
colnames(Data)[7] <- "Duration"
colnames(Data)[8] <- "Energy"
colnames(Data)[9] <- "Dancebility"
colnames(Data)[10] <- "Loudness"
colnames(Data)[11] <- "Valence"
colnames(Data)[12] <- "Acoustiveness"
colnames(Data)[13] <- "Popularity"


#Normalizing Loudness
x  = Data$Loudness
normalized = (x-min(x))/(max(x)-min(x))
loud = normalized * 100
rounded_loud = round(loud, digits=0)
Data$Loudness = rounded_loud
```

```r
#Creating the Dependannt Variable Rating based on the popularity given by Spotify
y = Data$Popularity
shapiro.test(y)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  y
## W = 0.94946, p-value = 1.984e-13
```

```r
qqnorm(y)
qqline(y, col=2)
```

```r
#we dont see normal distribution hence we cannot split the data in quartiles equally
#Instead we divide by average
mean(y)
```

```
## [1] 67.07692
```

```r
max(y)
```

```
## [1] 99
```

```r
Rating <- cut(y, breaks = c(0,67,99),
              labels = c("Below Average", "Above Average"),
              right = FALSE, include.lowest = TRUE)
Data['Rating'] <- Rating
View(Data)

summary(Data)
```

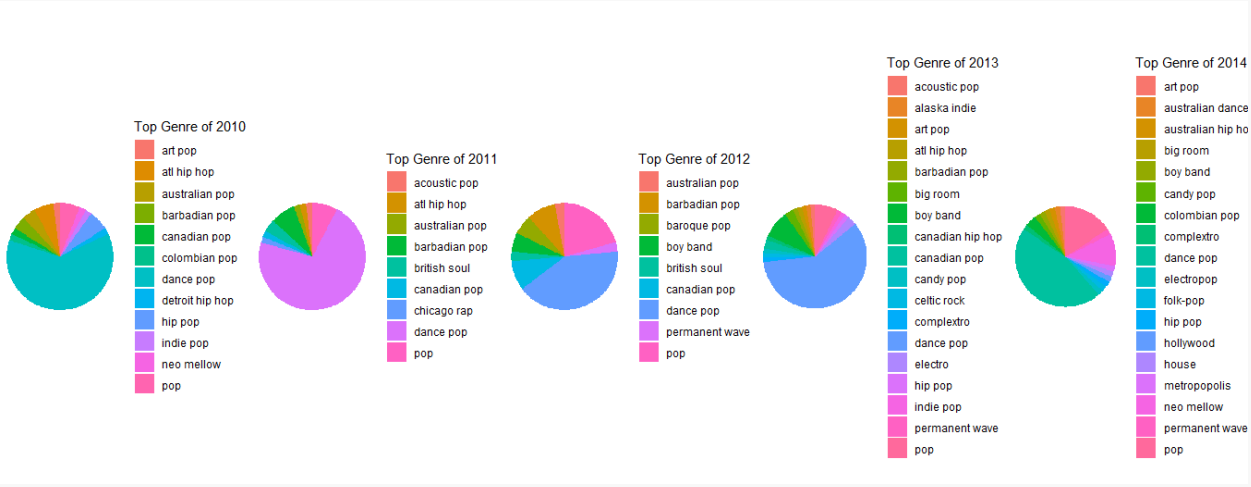
```
##        X                                                      title
##  Min.   :  1.0   A Little Party Never Killed Nobody (All We Got):
2
##  1st Qu.:152.2   All I Ask                                       :
2
##  Median :302.5   Castle Walls (feat. Christina Aguilera)        :
2
##  Mean   :302.4   Company                                         :
2
##  3rd Qu.:453.8   First Time                                      :
2
##  Max.   :603.0   Here                                            :
```

```
2
##                  (Other)
:586
##             artist              Genre          year           bpm
##  Katy Perry   : 17   dance pop    :324   Min.   :2010   Min.   :
43.0
##  Justin Bieber: 16   pop          : 60   1st Qu.:2013   1st
Qu.:100.0
##  Maroon 5     : 15   canadian pop : 34   Median :2015   Median
:120.0
##  Rihanna      : 15   barbadian pop: 15   Mean   :2015   Mean
:118.7
##  Lady Gaga    : 14   boy band     : 15   3rd Qu.:2017   3rd
Qu.:129.0
##  Bruno Mars   : 13   electropop   : 13   Max.   :2019   Max.
:206.0
##  (Other)      :508   (Other)      :137
##     Duration         Energy       Dancebility       Loudness
##  Min.   :134.0   Min.   : 4.00   Min.   :23.00   Min.   :  0.00
##  1st Qu.:202.0   1st Qu.:61.00   1st Qu.:57.00   1st Qu.: 69.00
##  Median :220.5   Median :74.00   Median :66.00   Median : 77.00
##  Mean   :224.7   Mean   :70.59   Mean   :64.53   Mean   : 73.23
##  3rd Qu.:239.0   3rd Qu.:82.00   3rd Qu.:73.75   3rd Qu.: 85.00
##  Max.   :424.0   Max.   :98.00   Max.   :97.00   Max.   :100.00
##
##     Valence      Acoustiveness    Popularity             Rating
##  Min.   : 4.00   Min.   : 0.0   Min.   : 7.00   Below Average:245
##  1st Qu.:35.00   1st Qu.: 2.0   1st Qu.:60.00   Above Average:353
##  Median :52.00   Median : 6.0   Median :69.00
##  Mean   :52.34   Mean   :14.4   Mean   :67.08
##  3rd Qu.:69.00   3rd Qu.:17.0   3rd Qu.:76.00
##  Max.   :98.00   Max.   :99.0   Max.   :99.00
##
```

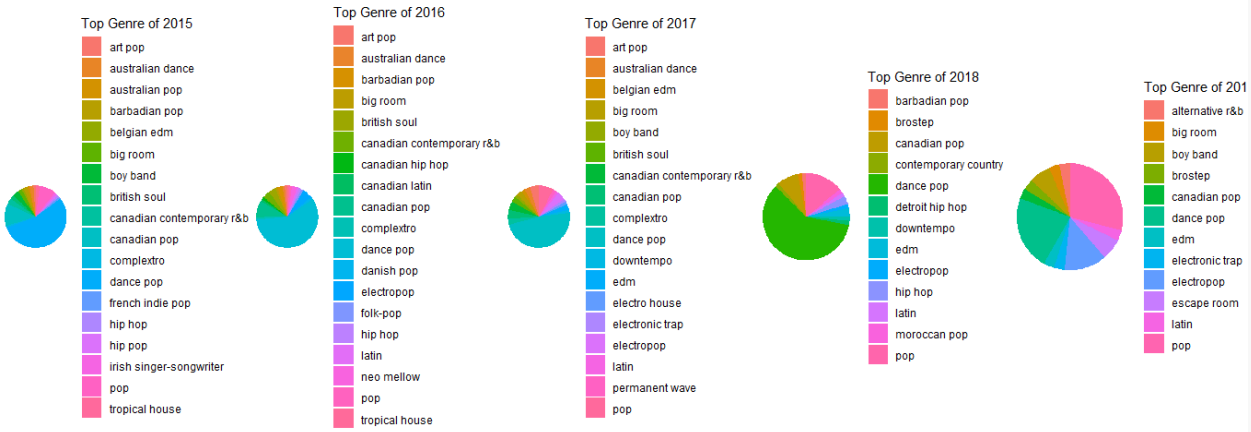
```r
#----------------------Exploratory Data Analysis---------------------#


#Exploring Genres
gen = count(Data$Genre)
gen_dsc = gen[order(-gen$freq),]
gen10 = gen_dsc[1:10,]
barplot(gen10$freq, names.arg = gen10$x,main = 'Top 10 Genres',xlab =
'Genre',ylab = 'No. of songs')


years1 = Data[Data$year == c(2010),4:5]
gen1 = count(years1$Genre)
gen1 = gen1[order(-gen1$freq),]

years2 = Data[Data$year == c(2011),4:5]
gen2 = count(years2$Genre)
gen2 = gen2[order(-gen2$freq),]

years3 = Data[Data$year == c(2012),4:5]
gen3 = count(years3$Genre)
gen3 = gen3[order(-gen3$freq),]

years4 = Data[Data$year == c(2013),4:5]
gen4 = count(years4$Genre)
gen4 = gen4[order(-gen4$freq),]

years5 = Data[Data$year == c(2014),4:5]
gen5 = count(years5$Genre)
gen5 = gen5[order(-gen5$freq),]

years6 = Data[Data$year == c(2015),4:5]
gen6 = count(years6$Genre)
gen6 = gen6[order(-gen6$freq),]

years7 = Data[Data$year == c(2016),4:5]
gen7 = count(years7$Genre)
gen7 = gen7[order(-gen7$freq),]

years8 = Data[Data$year == c(2017),4:5]
gen8 = count(years8$Genre)
gen8 = gen8[order(-gen8$freq),]

years9 = Data[Data$year == c(2018),4:5]
```

```r
gen9 = count(years9$Genre)
gen9 = gen9[order(-gen9$freq),]

years10 = Data[Data$year == c(2019),4:5]
gen10 = count(years10$Genre)
gen10 = gen10[order(-gen10$freq),]

plot1 <- ggplot(gen1, aes(x="", y=gen1$freq, fill=gen1$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2010")

plot2 <- ggplot(gen2, aes(x="", y=gen2$freq, fill=gen2$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2011")

plot3 <- ggplot(gen3, aes(x="", y=gen3$freq, fill=gen3$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2012")

plot4 <- ggplot(gen4, aes(x="", y=gen4$freq, fill=gen4$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2013")

plot5 <- ggplot(gen5, aes(x="", y=gen5$freq, fill=gen5$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2014")

plot6 <- ggplot(gen6, aes(x="", y=gen6$freq, fill=gen6$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2015")

plot7 <- ggplot(gen7, aes(x="", y=gen7$freq, fill=gen7$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2016")

plot8 <- ggplot(gen8, aes(x="", y=gen8$freq, fill=gen8$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2017")
```

```
plot9 <- ggplot(gen9, aes(x="", y=gen9$freq, fill=gen9$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2018")

plot10 <- ggplot(gen10, aes(x="", y=gen10$freq, fill=gen10$x)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() + scale_fill_discrete(name = "Top Genre of 2019")

grid.arrange(plot1, plot2,plot3,plot4,plot5 ,ncol=5)
```




```
grid.arrange(plot6,plot7,plot8,plot9,plot10 ,ncol=5)
```




```
#Checking if there is a optimal duration for a song
dur_data = Data[,c(1,2,7,13,14)]
durmin = round(dur_data$Duration/60, digits=1)
dur_data$Duration<- durmin #minute(period)
dur_data
```

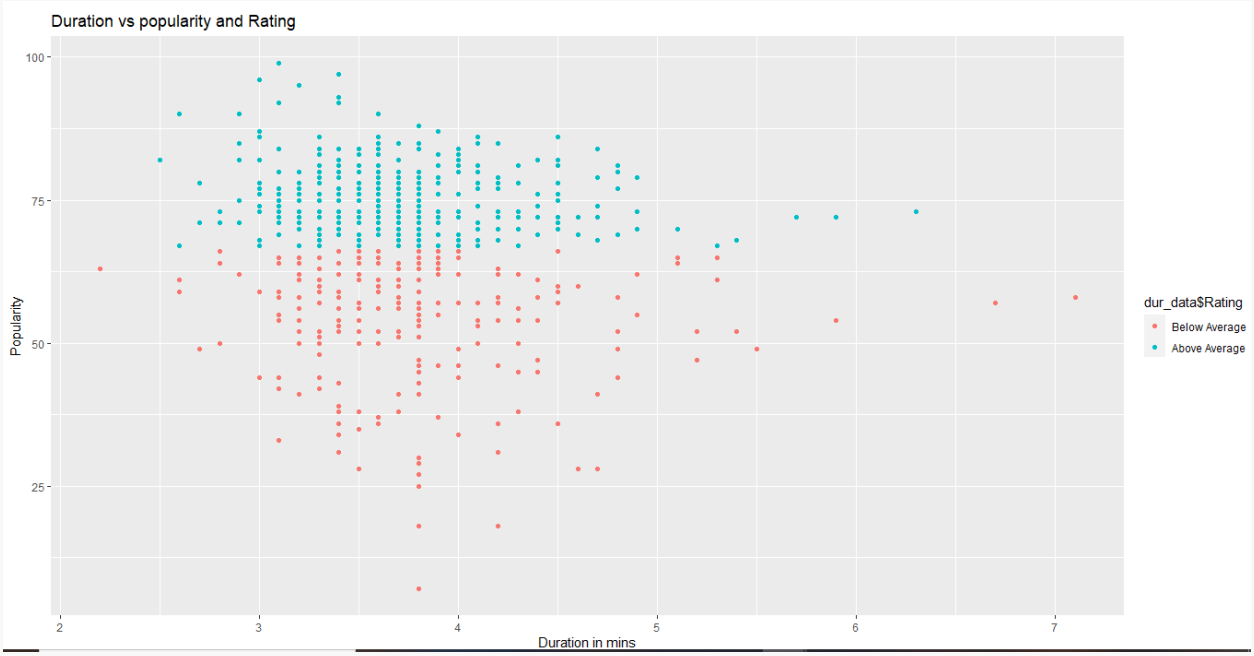
```
##        X
## 1      1
## 2      2
## 3      3
## 4      4
## 5      5
## 6      6
## 7      7
## 8      8
## 9      9
## 10    10
## 11    11
## 12    12
## 13    13
## 14    14
## 15    15
## 16    16
## 17    17
## 18    18
## 19    19
## 20    20
## 21    21
## 22    22
## 23    23
```

```
title
## 1
Hey, Soul Sister
## 2
Love The Way You Lie
## 3
TiK ToK
## 4
Bad Romance
## 5
Just the Way You Are
## 6
Baby
## 7
Dynamite
## 8
Secrets
## 9
Empire State of Mind (Part II) Broken Down
## 10
Only Girl (In The World)
## 11                                                                  Club
Can't Handle Me (feat. David Guetta)
## 12
Marry You
## 13
Cooler Than Me - Single Mix
## 14
Telephone
## 15
Like A G6
## 16
OMG (feat. will.i.am)
## 17
Eenie Meenie
## 18
The Time (Dirty Bit)
## 19
Alejandro
## 20
Your Love Is My Drug
## 21
Meet Me Halfway
## 22
Whataya Want from Me
## 23
```

```
Kills You Slowly
##     Duration Popularity        Rating
## 1        3.6         83 Above Average
## 2        4.4         82 Above Average
## 3        3.3         80 Above Average
## 4        4.9         79 Above Average
## 5        3.7         78 Above Average
## 6        3.6         77 Above Average
## 7        3.4         77 Above Average
## 8        3.8         77 Above Average
## 9        3.6         76 Above Average
## 10       3.9         73 Above Average
## 11       3.9         73 Above Average
## 12       3.8         73 Above Average
## 13       3.5         73 Above Average
## 14       3.7         73 Above Average
## 15       3.6         72 Above Average
## 16       4.5         72 Above Average
## 17       3.4         71 Above Average
## 18       5.1         70 Above Average
## 19       4.6         69 Above Average
## 20       3.1         69 Above Average
## 21       4.7         68 Above Average
## 22       3.8         66 Below Average
## 23       3.6         66 Below Average
## 24       3.6         65 Below Average
## 25       4.0         65 Below Average
## 26       3.5         65 Below Average
## 27       3.4         64 Below Average
## 28       3.8         63 Below Average
## 29       3.8         63 Below Average
## 30       3.9         62 Below Average
## 31       3.5         62 Below Average
## 32       3.2         62 Below Average
## 33       2.9         62 Below Average
## 34       4.3         62 Below Average
## 35       4.2         62 Below Average
## 36       3.3         61 Below Average
## 37       4.4         61 Below Average
## 38       3.8         59 Below Average
## 39       4.5         59 Below Average
## 40       4.2         58 Below Average
## 41       3.1         58 Below Average
## 42       4.5         57 Below Average
## 43       4.2         57 Below Average
```

```
## 44       3.2          56 Below Average
```

```
ggplot(dur_data, aes(x=dur_data$Duration,
y=dur_data$Popularity,color=dur_data$Rating)) +
  geom_point()+ labs(y = 'Popularity', x = "Duration in mins", title =
"Duration vs popularity and Rating")
```

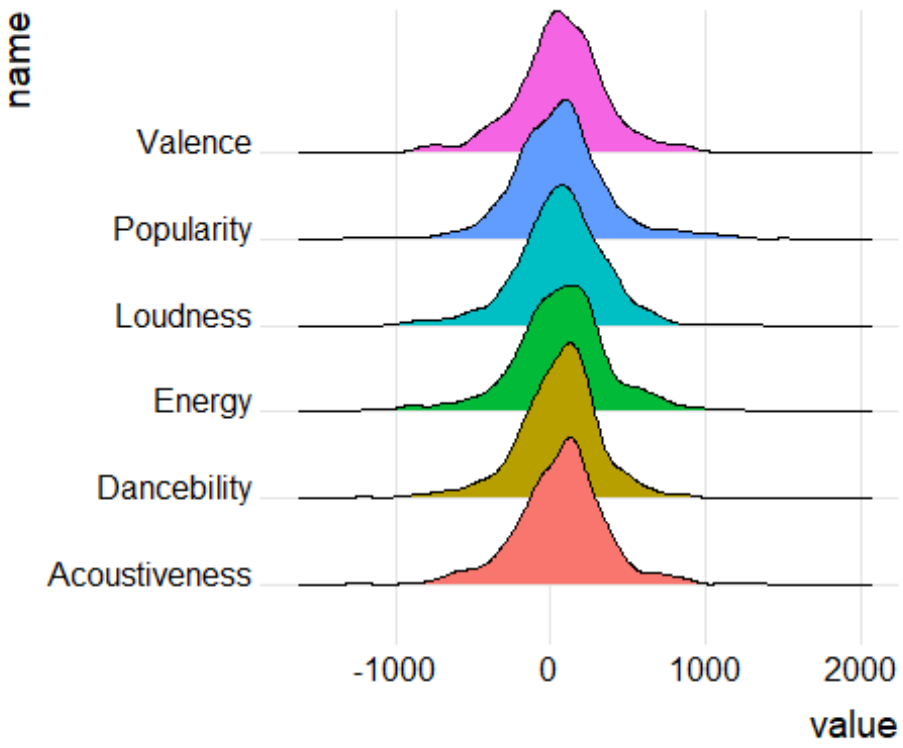



```
props = Data[,c(8:13)]

nrow(props)

## [1] 598

colMeans(props)

##        Energy   Dancebility      Loudness       Valence
Acoustiveness
##      70.58863      64.52676      73.23244      52.33946
14.39632
##    Popularity
##      67.07692

var(props)

##                    Energy Dancebility   Loudness     Valence
Acoustiveness
## Energy         256.21910    28.37283 138.353736  143.687619    -
```

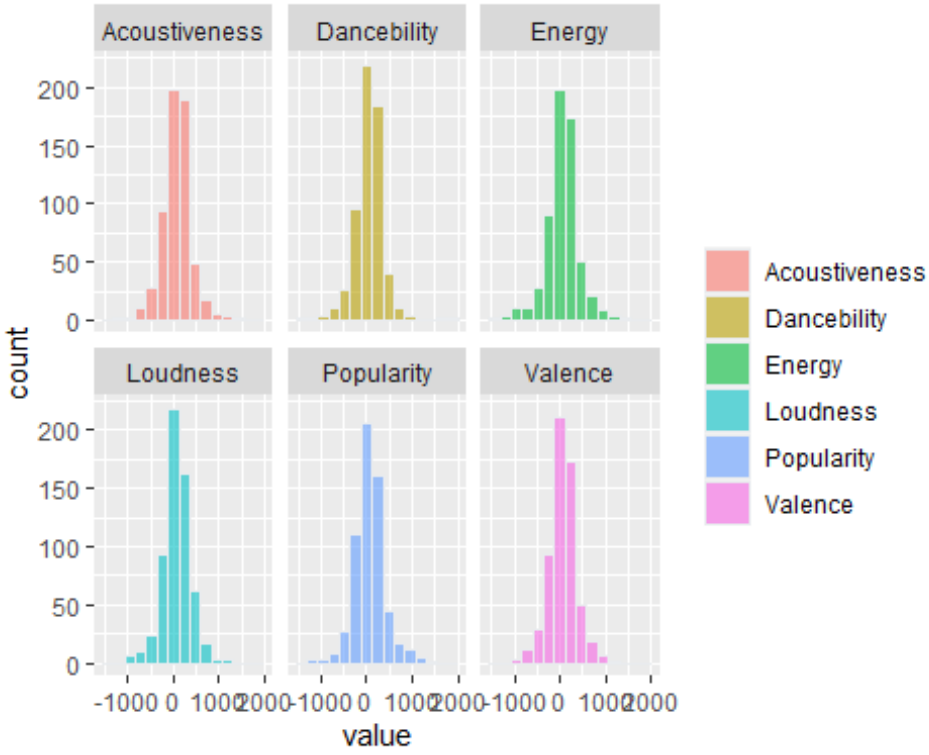
```
193.319104
## Dancebility     28.37283   172.98337  22.193938  146.175994    -
69.080136
## Loudness       138.35374    22.19394 171.455093  100.537378    -
95.835994
## Valence        143.68762   146.17599 100.537378  504.412209   -
118.667426
## Acoustiveness -193.31910   -69.08014 -95.835994 -118.667426
433.331779
## Popularity     -19.44066    12.85388   4.353949    5.183224
3.440149
##                Popularity
## Energy         -19.440665
## Dancebility     12.853885
## Loudness         4.353949
## Valence          5.183224
## Acoustiveness    3.440149
## Popularity     175.159902
```

```
density_data <- data.frame(
  name=c("Energy","Dancebility","Loudness",
         "Valence","Acoustiveness","Popularity"),
  value=c( rnorm(598, 70, 256), rnorm(598, 64, 172), rnorm(598, 73,
171),
           rnorm(598, 52, 504), rnorm(598, 14, 433),
rnorm(598,67,175))
)

ggplot(density_data, aes(x = value, y = name, fill = name)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none")
```

```
## Picking joint bandwidth of 60.8
```



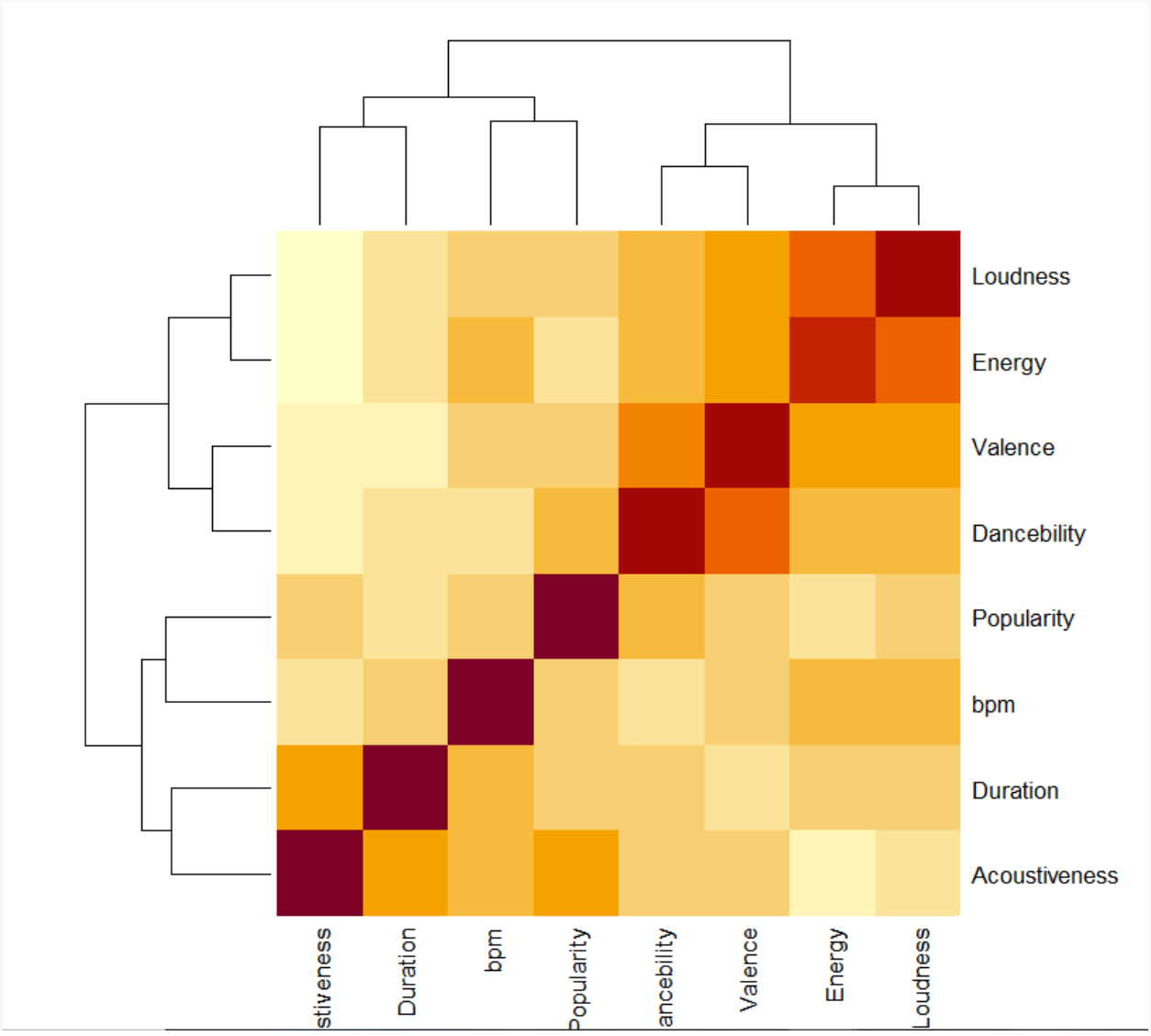

```
ggplot(density_data, aes(x=value, fill=name)) +
  geom_histogram( color="#e9ecef", alpha=0.6, position =
'identity',bins=15) +
  labs(fill="") + facet_wrap(~name)
```




```
#------------------Statistical Testing-------------------------------#

Data_num = Data[,6:13]
Corr_mat = cor(Data_num)
Corr_mat

##                       bpm    Duration      Energy Dancebility
Loudness
## bpm            1.000000000 -0.02078528  0.10794891 -0.17258993
0.05654377
## Duration      -0.020785282  1.00000000 -0.14955130 -0.18096389 -
0.17058486
## Energy         0.107948905 -0.14955130  1.00000000  0.13477048
0.66010032
## Dancebility   -0.172589927 -0.18096389  0.13477048  1.00000000
0.12887153
## Loudness       0.056543770 -0.17058486  0.66010032  0.12887153
1.00000000
## Valence        0.003892528 -0.26790249  0.39968772  0.49485856
0.34186887
## Acoustiveness -0.117685930  0.09020716 -0.58017466 -0.25231364 -
0.35159540
## Popularity    -0.000492410 -0.11296345 -0.09176731  0.07384394
0.02512412
##                    Valence Acoustiveness  Popularity
```

```
## bpm             0.003892528   -0.11768593 -0.00049241
## Duration       -0.267902487    0.09020716 -0.11296345
## Energy          0.399687717   -0.58017466 -0.09176731
## Dancebility     0.494858557   -0.25231364  0.07384394
## Loudness        0.341868875   -0.35159540  0.02512412
## Valence         1.000000000   -0.25382153  0.01743772
## Acoustiveness -0.253821529    1.00000000  0.01248676
## Popularity      0.017437725    0.01248676  1.00000000
```

```
heatmap(Corr_mat)
```



```r
# T-Test on dataset columns Duration and Popularity
t.test(Data$Duration,Data$Popularity, var.equal = TRUE, paired=FALSE)
```

```
##
##  Two Sample t-test
##
## data:  Data$Duration and Data$Popularity
## t = 105.2, df = 1194, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  154.6432 160.5207
## sample estimates:
## mean of x mean of y
## 224.65886  67.07692
```

```r
# p-value is <2.2e-16 which is very less and hence we reject the null hypothesis
```

```r
with(Data,t.test(Energy,Valence))
```

```
##
##  Welch Two Sample t-test
##
## data:  Energy and Valence
## t = 16.181, df = 1079.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  16.03621 20.46212
## sample estimates:
## mean of x mean of y
##  70.58863  52.33946
```

```r
with(Data,t.test(Valence,Dancebility))
```

```
##
##  Welch Two Sample t-test
##
## data:  Valence and Dancebility
## t = -11.451, df = 963.38, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -14.27594 -10.09865
## sample estimates:
## mean of x mean of y
##  52.33946  64.52676
```

```r
with(Data,t.test(Energy,Acoustiveness))
```

```
##
##  Welch Two Sample t-test
##
## data:  Energy and Acoustiveness
## t = 52.329, df = 1120.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  54.08538 58.29924
## sample estimates:
## mean of x mean of y
##  70.58863  14.39632
```

```r
# (Energy, Valence), (Valence,Dancebility) and Energy,Acoustiveness
have very low p-value
# as seen from heat map earlier it has significant correlation and
hence we reject the null
# hypothesis for these audio properties.

with(Data,t.test(Popularity,Duration))
```

```
##
##  Welch Two Sample t-test
##
## data:  Popularity and Duration
## t = -105.2, df = 772.33, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -160.5223 -154.6416
## sample estimates:
## mean of x mean of y
##  67.07692 224.65886
```

```r
with(Data,t.test(Popularity,Energy))
```

```
##
##  Welch Two Sample t-test
##
## data:  Popularity and Energy
## t = -4.1347, df = 1153.3, p-value = 3.812e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.178120 -1.845291
## sample estimates:
## mean of x mean of y
##  67.07692  70.58863
```

```r
with(Data,t.test(Popularity,Dancebility))
```

```
##
##  Welch Two Sample t-test
##
## data:  Popularity and Dancebility
## t = 3.3423, df = 1194, p-value = 0.0008567
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.053184 4.047151
## sample estimates:
## mean of x mean of y
##  67.07692  64.52676
```

```r
with(Data,t.test(Popularity,Loudness))
```

```
##
##  Welch Two Sample t-test
##
## data:  Popularity and Loudness
## t = -8.0852, df = 1193.9, p-value = 1.511e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.649213 -4.661824
## sample estimates:
## mean of x mean of y
##  67.07692  73.23244
```

```r
with(Data,t.test(Popularity,Valence))
```

```
##
##  Welch Two Sample t-test
##
## data:  Popularity and Valence
## t = 13.825, df = 967.01, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  12.64547 16.82945
## sample estimates:
## mean of x mean of y
##  67.07692  52.33946
```

```r
with(Data,t.test(Popularity,Acoustiveness))
```

```
##
##  Welch Two Sample t-test
##
## data:  Popularity and Acoustiveness
```

```
## t = 52.224, df = 1011.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  50.70115 54.66005
## sample estimates:
## mean of x mean of y
##  67.07692  14.39632

# Checking the relation between dependent variable Popularity and
different audio properties as independent variables
# we found out that the p-value is very low for all the t-test
conducted between Popularity
# and independent variable and hence we reject the null hypothesis
stating there is significant
# relationship between dependent and independent variables.
```