cleaning.R

Apurva Sarode

2020-02-27

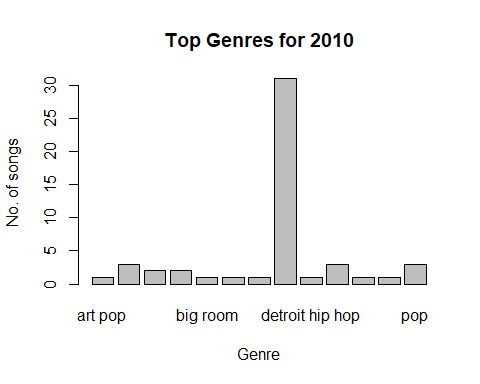
#Top Songs Analysis  
#importing dataset top10s and copying it to test data  
data = read.csv('C:\\Users\\Apurva Sarode\\Desktop\\Spotify\_mva.csv')  
View(data)  
#Data Cleaning  
#Adding column Rank which will denote rank of a song based on it popularity.  
# popularity from 90 - 100 is Rank 10 and so on  
for(x in 1:length(data$pop)){  
 if(data[x,15] <= 100 && data[x,15] >= 80){  
 data[x,16] = 5  
 }else if(data[x,15] < 80 && data[x,15] >= 60){  
 data[x,16] = 4  
 }else if(data[x,15] < 60 && data[x,15] >= 40){  
 data[x,16] = 3  
 }else if(data[x,15] < 40 && data[x,15] >= 20){  
 data[x,16] = 2  
 }else if(data[x,15] < 20 && data[x,15] >= 0){  
 data[x,16] = 1  
 }  
}  
data$pop <- NULL  
dim(data)

## [1] 603 15

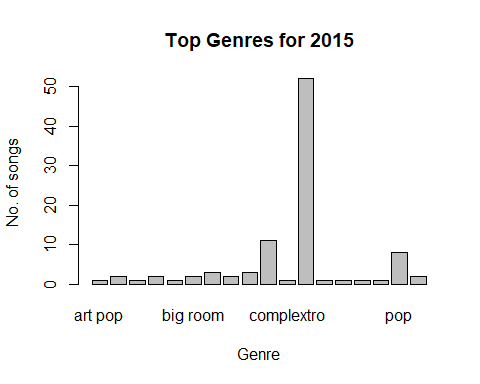
#removing values with 0 BPM and duration as 0 seconds  
data\_clean <- data[-c(433),]  
names(data\_clean)[15]<- "rating"  
View(data\_clean)  
#EDA  
#checking the ranges for all columns  
dim(data\_clean)

## [1] 602 15

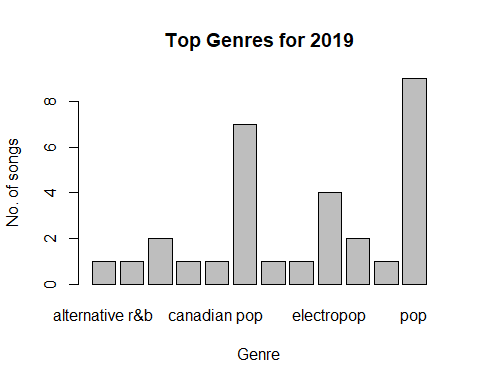
library(plyr)  
library(ggplot2)  
#Finding top genre for 3 years  
year1 = data\_clean[data\_clean$year == 2010,]  
gen1 = count(year1$top.genre)  
barplot(gen1$freq, names.arg = gen1$x,main = 'Top Genres for 2010',xlab = 'Genre',ylab = 'No. of songs')



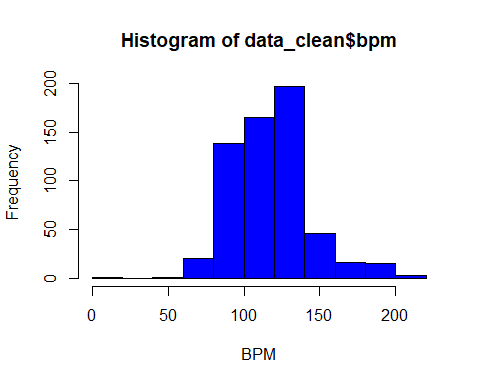
year2 = data\_clean[data\_clean$year == 2015,]  
gen2 = count(year2$top.genre)  
barplot(gen2$freq, names.arg = gen2$x,main = 'Top Genres for 2015',xlab = 'Genre',ylab = 'No. of songs')



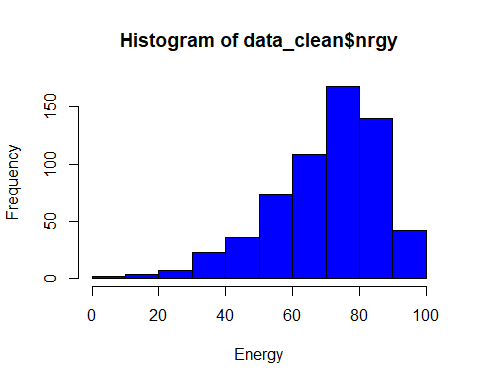
year3 = data\_clean[data\_clean$year == 2019,]  
gen3 = count(year3$top.genre)  
barplot(gen3$freq, names.arg = gen3$x,main = 'Top Genres for 2019',xlab = 'Genre',ylab = 'No. of songs')



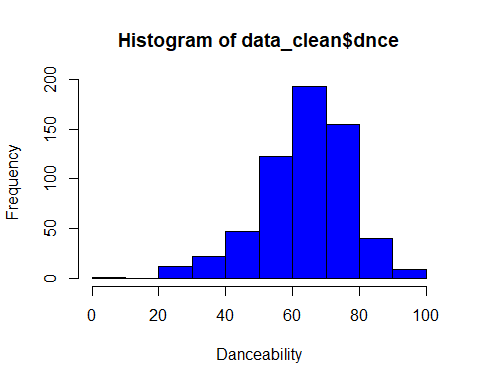
#Histogram view of audio properties  
hist(data\_clean$bpm, breaks=12,col="blue",xlab="BPM")



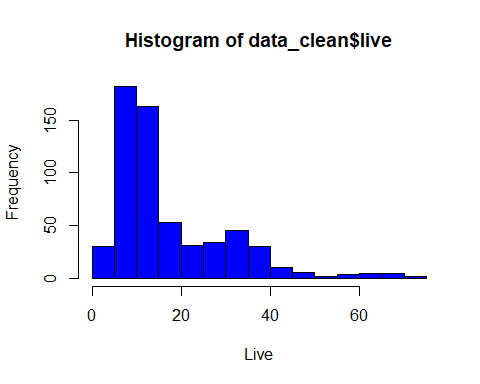
hist(data\_clean$nrgy, breaks=12,col="blue",xlab="Energy")



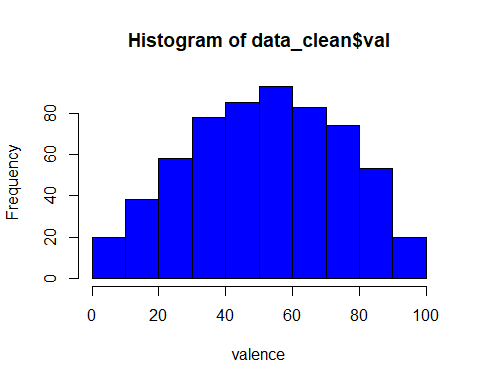
hist(data\_clean$dnce, breaks=12,col="blue",xlab="Danceability")



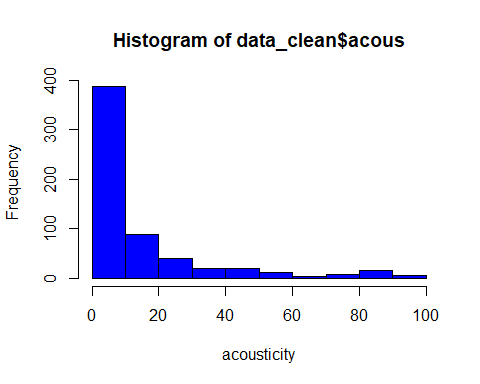
hist(data\_clean$live, breaks=12,col="blue",xlab="Live")



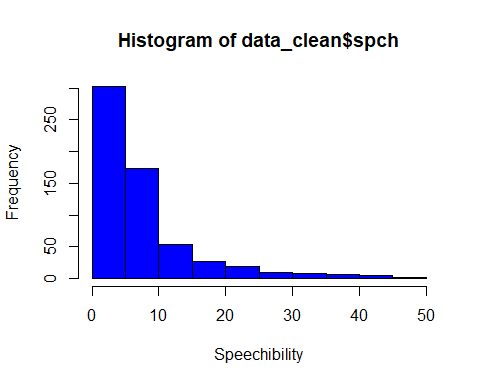
hist(data\_clean$val, breaks=12,col="blue",xlab="valence")



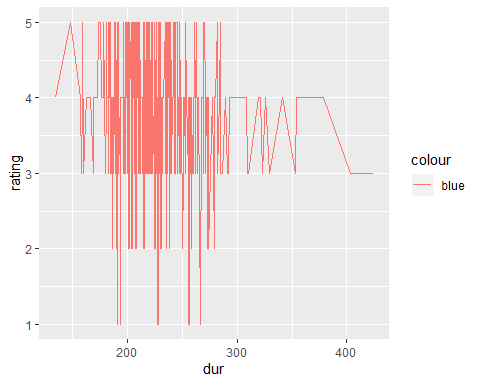
hist(data\_clean$acous, breaks=12,col="blue",xlab="acousticity")



hist(data\_clean$spch, breaks=12,col="blue",xlab="Speechibility")



#Line chart for popularity and Duration  
ggplot(data\_clean) +geom\_line(aes(x = dur, y = rating, color = "blue"))



# T-Test on dataset columns Duration and rating  
t.test(data\_clean$dur,data\_clean$rating, var.equal = TRUE, paired=FALSE)

##   
## Two Sample t-test  
##   
## data: data\_clean$dur and data\_clean$rating  
## t = 158.71, df = 1202, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 218.0532 223.5116  
## sample estimates:  
## mean of x mean of y   
## 224.611296 3.828904

#Comparing relation between two top genre from 2010 to 2019.  
star5 = data\_clean[which(data\_clean$rating==5),]  
with(star5,t.test(dnce[top.genre=="dance pop"],dnce[top.genre=="pop"],var.equal=TRUE))

##   
## Two Sample t-test  
##   
## data: dnce[top.genre == "dance pop"] and dnce[top.genre == "pop"]  
## t = -1.0029, df = 40, p-value = 0.3219  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -13.676389 4.604961  
## sample estimates:  
## mean of x mean of y   
## 67.03571 71.57143

with(star5,t.test(nrgy[top.genre=="dance pop"],nrgy[top.genre=="pop"],var.equal=TRUE))

##   
## Two Sample t-test  
##   
## data: nrgy[top.genre == "dance pop"] and nrgy[top.genre == "pop"]  
## t = 1.7587, df = 40, p-value = 0.08629  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -1.433565 20.647851  
## sample estimates:  
## mean of x mean of y   
## 66.67857 57.07143

with(star5,t.test(bpm[top.genre=="dance pop"],bpm[top.genre=="pop"],var.equal=TRUE))

##   
## Two Sample t-test  
##   
## data: bpm[top.genre == "dance pop"] and bpm[top.genre == "pop"]  
## t = 2.1881, df = 40, p-value = 0.03456  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 1.147886 28.923542  
## sample estimates:  
## mean of x mean of y   
## 119.3929 104.3571

with(star5,t.test(val[top.genre=="dance pop"],val[top.genre=="pop"],var.equal=TRUE))

##   
## Two Sample t-test  
##   
## data: val[top.genre == "dance pop"] and val[top.genre == "pop"]  
## t = -1.4541, df = 40, p-value = 0.1537  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -27.825938 4.540224  
## sample estimates:  
## mean of x mean of y   
## 48.78571 60.42857

#### PCA  
for(y in 1:length(data\_clean$rating)){  
 if(data\_clean[y,15] >= 1 & data\_clean[y,15] <= 3){  
 data\_clean[y,16] = 1  
 }else{  
 data\_clean[y,16] = 5  
 }  
}  
View(data\_clean)  
#rating column values are split in 2 groups 1 and 5 for t and var test(columnname is V16)  
cor(data\_clean[c(7,8,11,13,14)])

## nrgy dnce val acous spch  
## nrgy 1.0000000 0.16685024 0.4102908 -0.5625564 0.10711812  
## dnce 0.1668502 1.00000000 0.5049296 -0.2413363 -0.02922118  
## val 0.4102908 0.50492963 1.0000000 -0.2486811 0.12284677  
## acous -0.5625564 -0.24133632 -0.2486811 1.0000000 0.00246410  
## spch 0.1071181 -0.02922118 0.1228468 0.0024641 1.00000000

data\_pca = prcomp(data\_clean[c(7,8,11,13,14)],scale. = TRUE)  
data\_pca

## Standard deviations (1, .., p=5):  
## [1] 1.4439153 1.0176814 1.0011165 0.7365874 0.5784789  
##   
## Rotation (n x k) = (5 x 5):  
## PC1 PC2 PC3 PC4 PC5  
## nrgy -0.53106816 0.3018103 -0.3408606 -0.3818033 -0.60408400  
## dnce -0.43372652 -0.5131816 0.3929811 0.4823965 -0.40172805  
## val -0.52681796 -0.1571937 0.3907000 -0.5388521 0.50472255  
## acous 0.49239464 -0.1382874 0.5100046 -0.5094188 -0.46777338  
## spch -0.09928882 0.7757074 0.5626977 0.2676767 -0.01184546

summary(data\_pca)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5  
## Standard deviation 1.444 1.0177 1.0011 0.7366 0.57848  
## Proportion of Variance 0.417 0.2071 0.2004 0.1085 0.06693  
## Cumulative Proportion 0.417 0.6241 0.8246 0.9331 1.00000

data\_pca$x

## PC1 PC2 PC3 PC4 PC5  
## 1 -1.168320396 -0.435362099 -0.039151263 -1.271456683 -0.2412041809  
## 2 -1.317131044 1.378217388 1.385378953 -0.136793273 -1.1308962645  
## 3 -1.432924832 0.285364744 0.704262305 -0.036255619 -0.3415971627  
## 4 -1.602879141 -0.305790987 -0.636065986 -0.552231502 -0.2164379874  
## 5 -0.445434523 -0.041214120 -1.082152129 0.039428584 -0.4127259666

data\_pca1 = cbind(data.frame(data\_clean$V16),data\_pca$x)  
data\_pca1

## data\_clean.V16 PC1 PC2 PC3 PC4  
## 1 5 -1.168320396 -0.435362099 -0.039151263 -1.271456683  
## 2 5 -1.317131044 1.378217388 1.385378953 -0.136793273  
## 3 5 -1.432924832 0.285364744 0.704262305 -0.036255619  
## 4 5 -1.602879141 -0.305790987 -0.636065986 -0.552231502  
## 5 5 -0.445434523 -0.041214120 -1.082152129 0.039428584  
## PC5  
## 1 -0.2412041809  
## 2 -1.1308962645  
## 3 -0.3415971627  
## 4 -0.2164379874  
## 5 -0.4127259666

var.test(PC3~data\_clean$V16,data=data\_pca1)

##   
## F test to compare two variances  
##   
## data: PC3 by data\_clean$V16  
## F = 1.022, num df = 146, denom df = 454, p-value = 0.8534  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.7915999 1.3436023  
## sample estimates:  
## ratio of variances   
## 1.021978

#t.test(PC1~data\_clean$V16,data=data\_pca)  
#t.test(PC2~data\_clean$V16,data=data\_pca)  
t.test(PC3~data\_clean$V16,data=data\_pca1)

##   
## Welch Two Sample t-test  
##   
## data: PC3 by data\_clean$V16  
## t = -0.065215, df = 245.03, p-value = 0.9481  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.1945103 0.1820429  
## sample estimates:  
## mean in group 1 mean in group 5   
## -0.004711502 0.001522178