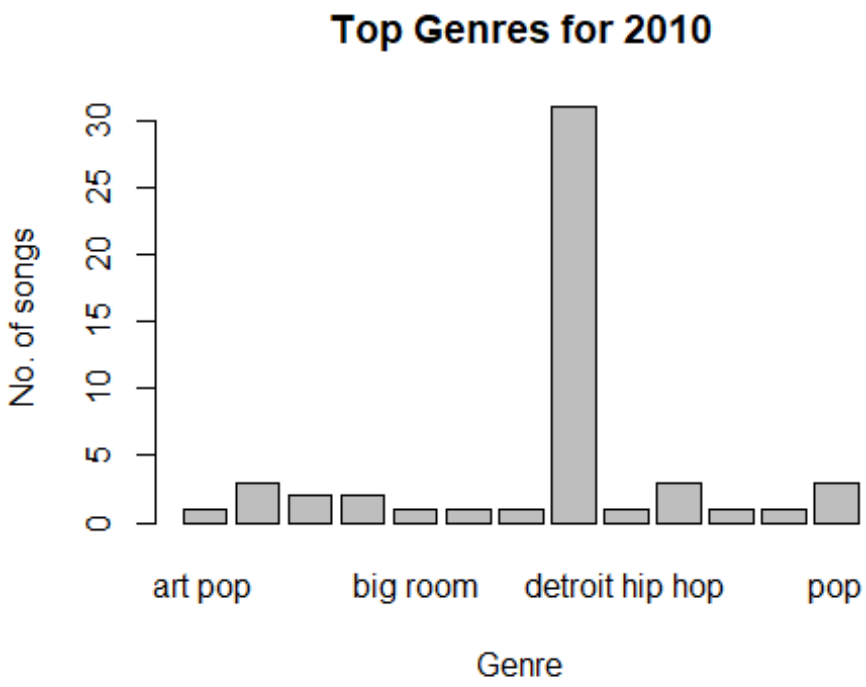
cleaning.R

Apurva Sarode

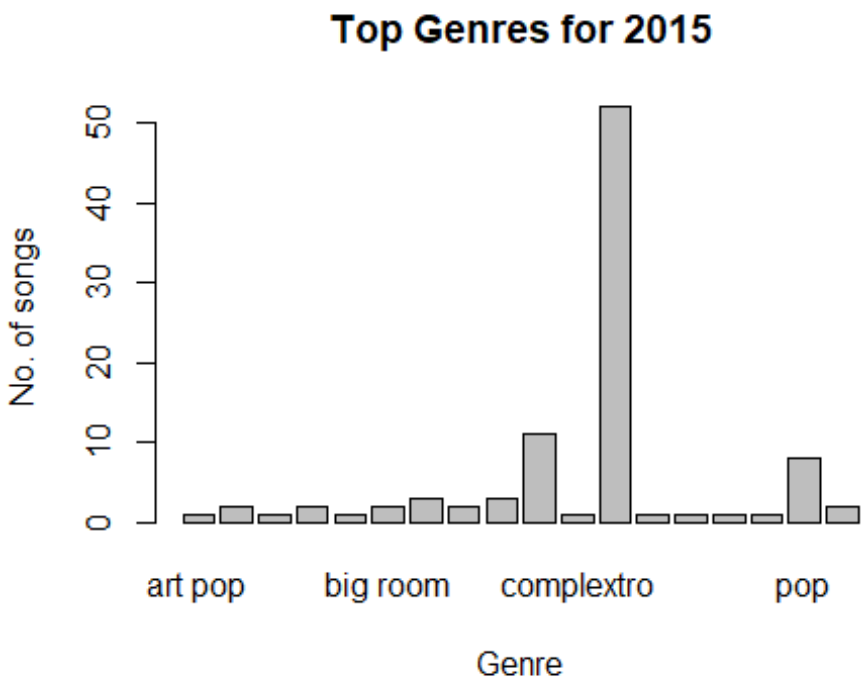
2020-02-20

```
#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)
```

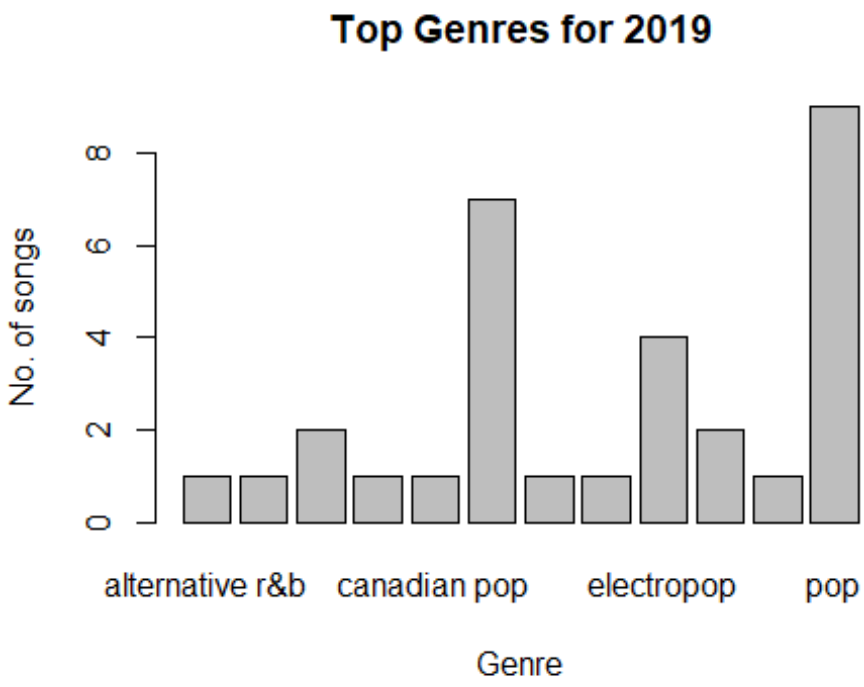
```r
#Finding top genre for 3 years
year1 = data_clean[data_clean$year == 2010,]
gen1 = count(year1$top.genre)
barplot(gen1$freq, names.arg = gen1$x,main = 'Top Genres for 2010',xlab =
'Genre',ylab = 'No. of songs')
```



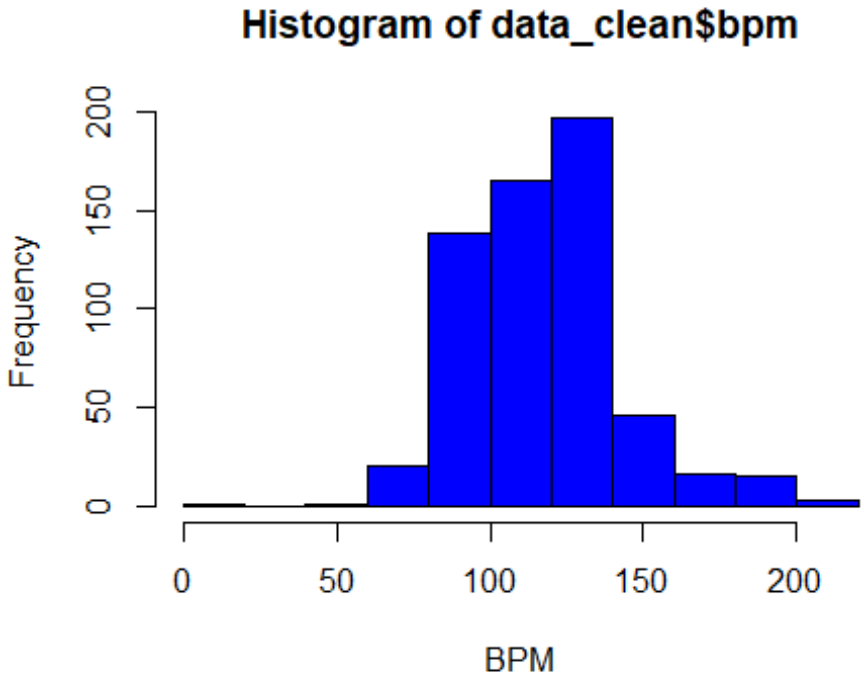

```r
year2 = data_clean[data_clean$year == 2015,]
gen2 = count(year2$top.genre)
barplot(gen2$freq, names.arg = gen2$x,main = 'Top Genres for 2015',xlab =
'Genre',ylab = 'No. of songs')
```



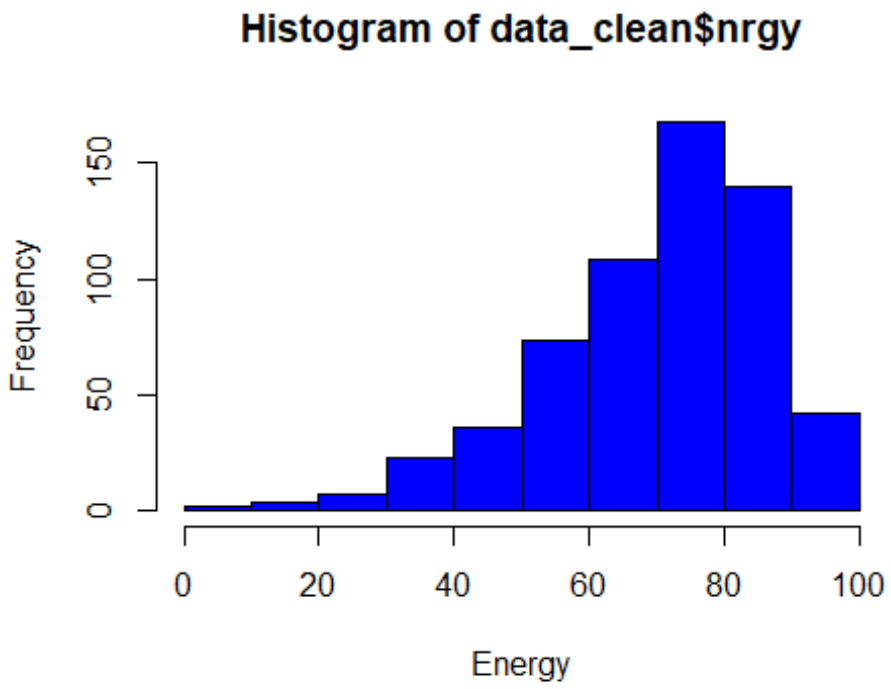

```
year3 = data_clean[data_clean$year == 2019,]
gen3 = count(year3$top.genre)
barplot(gen3$freq, names.arg = gen3$x,main = 'Top Genres for 2019',xlab =
'Genre',ylab = 'No. of songs')
```



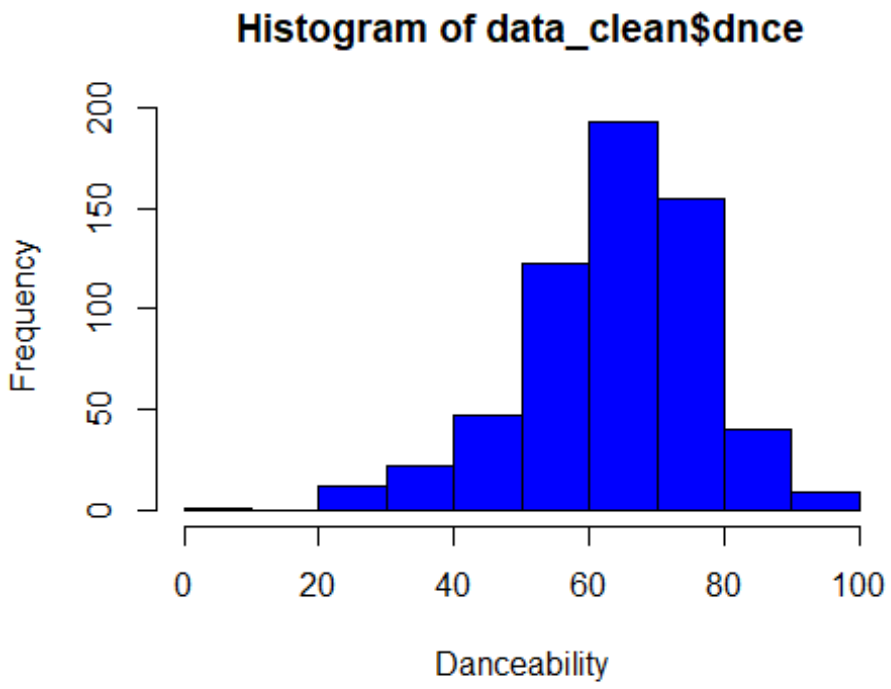

```r
#Histogram view of audio properties
hist(data_clean$bpm, breaks=12,col="blue",xlab="BPM")
```



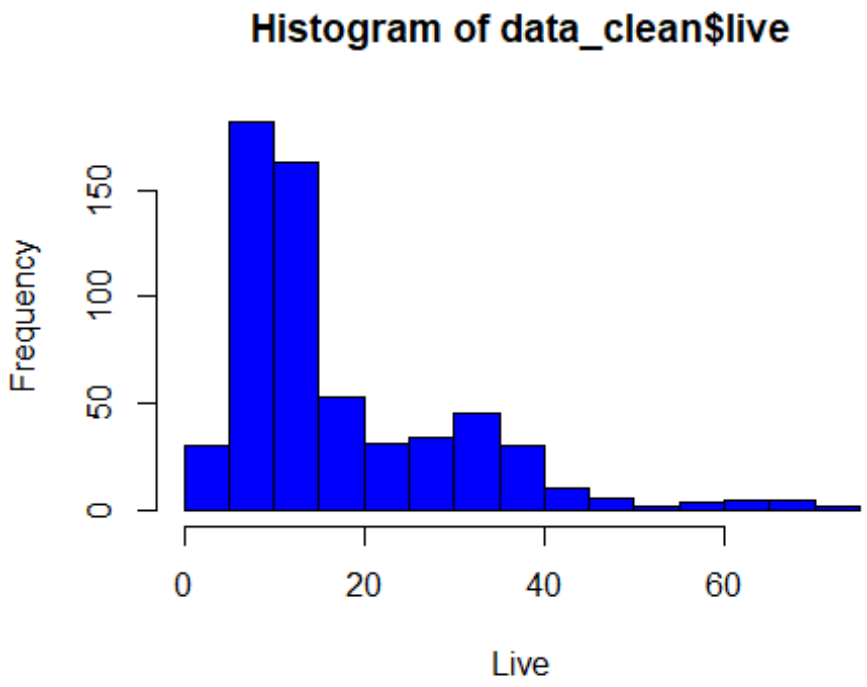
```r
hist(data_clean$nrgy, breaks=12,col="blue",xlab="Energy")
```



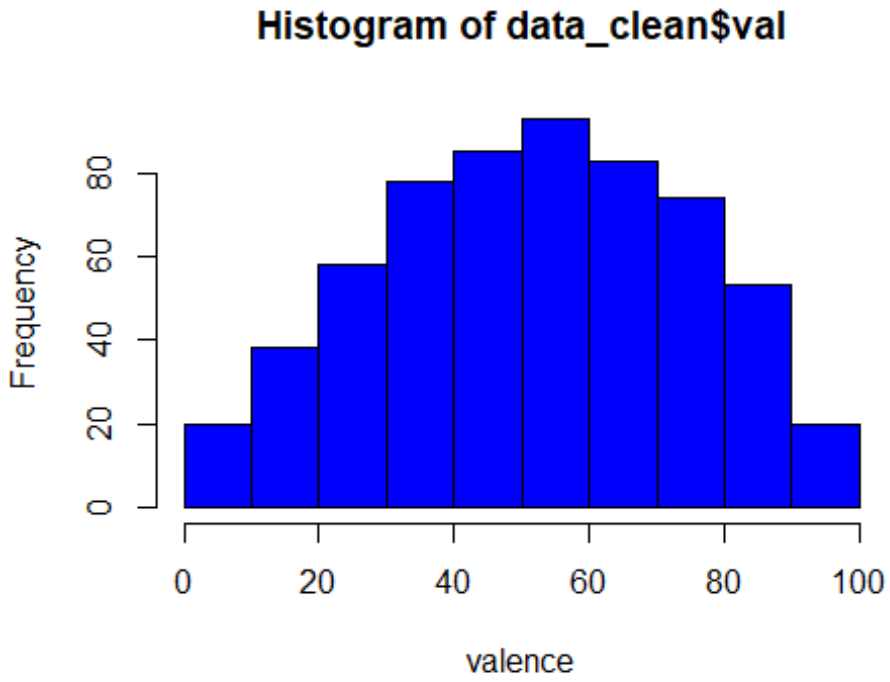

```r
hist(data_clean$dnce, breaks=12,col="blue",xlab="Danceability")
```



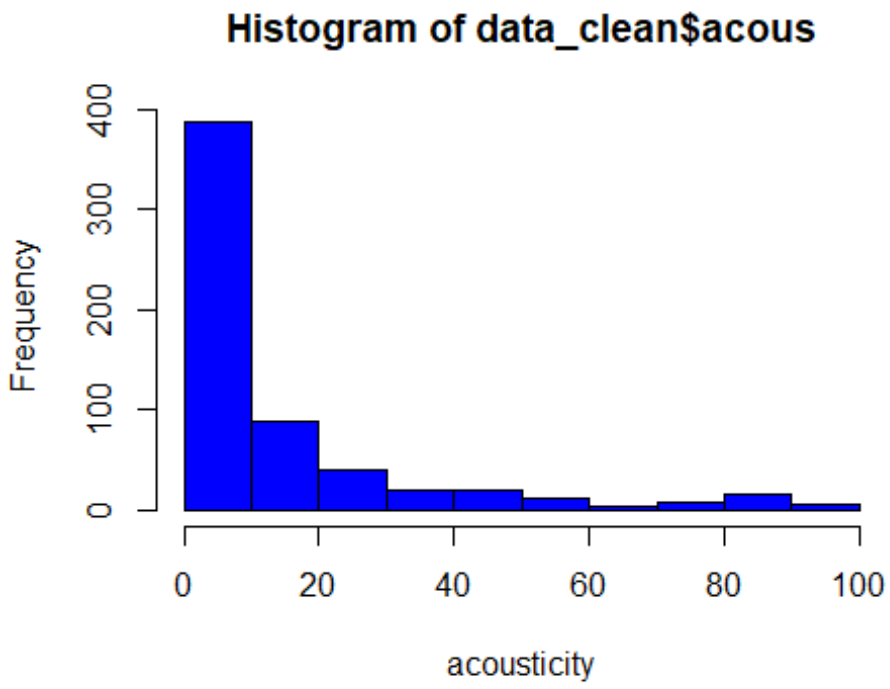
```r
hist(data_clean$live, breaks=12,col="blue",xlab="Live")
```



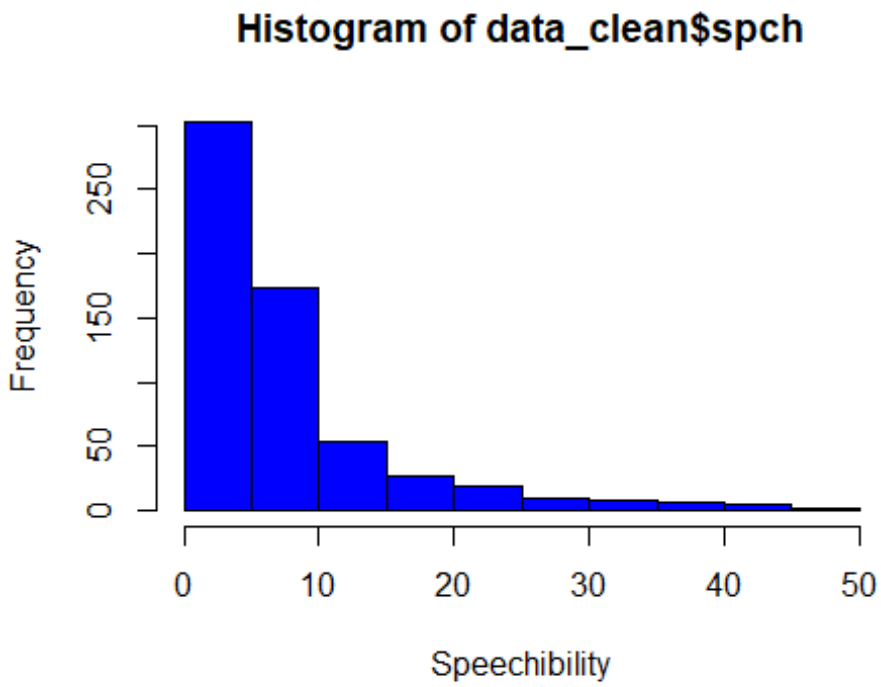

```r
hist(data_clean$val, breaks=12,col="blue",xlab="valence")
```



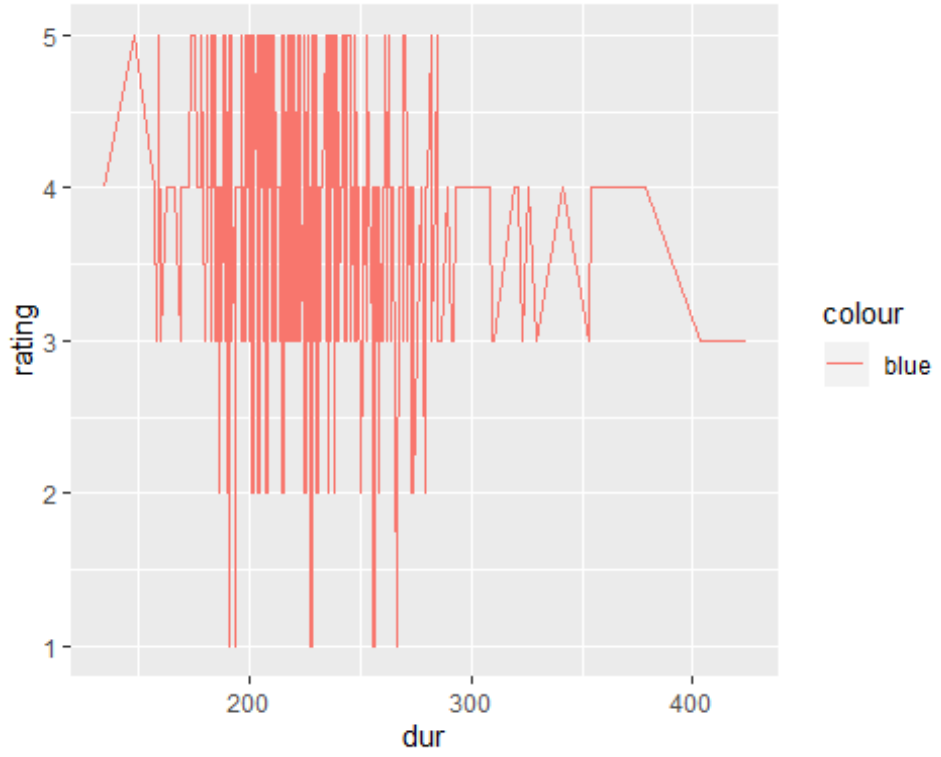
```r
hist(data_clean$acous, breaks=12,col="blue",xlab="acousticity")
```




```r
hist(data_clean$spch, breaks=12,col="blue",xlab="Speechibility")
```



```r
#Line chart for popularity and Duration
ggplot(data_clean) +geom_line(aes(x = dur, y = rating, color = "blue"))
```




```r
#tests

# 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
```