STAT515-MID PROJECT

Sai Bhaskara Ganesh Gandi- G01474817

2024-03-10

Loaded the dataset into the Rstudio.

```
spotify = read.csv("/Users/saibhaskarganeshgandi/Desktop/R Studio/Spotify-2000.csv")
View(spotify)
```

```
head(spotify)
```

```
##   Index                                 Title            Artist
## 1     1                               Sunrise       Norah Jones
## 2     2                           Black Night       Deep Purple
## 3     3                        Clint Eastwood          Gorillaz
## 4     4                         The Pretender      Foo Fighters
## 5     5                Waitin' On A Sunny Day Bruce Springsteen
## 6     6 The Road Ahead (Miles Of The Unknown)      City To City
##              Top.Genre Year Beats.Per.Minute..BPM. Energy Danceability
## 1      adult standards 2004                    157     30           53
## 2           album rock 2000                    135     79           50
## 3  alternative hip hop 2001                    168     69           66
## 4    alternative metal 2007                    173     96           43
## 5         classic rock 2002                    106     82           58
## 6 alternative pop rock 2004                     99     46           54
##   Loudness..dB. Liveness Valence Length..Duration. Acousticness Speechiness
## 1           -14       11      68               201           94           3
## 2           -11       17      81               207           17           7
## 3            -9        7      52               341            2          17
## 4            -4        3      37               269            0           4
## 5            -5       10      87               256            1           3
## 6            -9       14      14               247            0           2
##   Popularity
## 1         71
## 2         39
## 3         69
## 4         76
## 5         59
## 6         45
```

```
str(spotify)
```

```
## 'data.frame':    1994 obs. of  15 variables:
##  $ Index                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Title                 : chr  "Sunrise" "Black Night" "Clint Eastwood" "The Pretender" ...
##  $ Artist                : chr  "Norah Jones" "Deep Purple" "Gorillaz" "Foo Fighters" ...
##  $ Top.Genre             : chr  "adult standards" "album rock" "alternative hip hop" "alternative me
##  $ Year                  : int  2004 2000 2001 2007 2002 2004 2002 2006 2004 2002 ...
##  $ Beats.Per.Minute..BPM.: int  157 135 168 173 106 99 102 137 148 112 ...
##  $ Energy                : int  30 79 69 96 82 46 71 96 92 67 ...
##  $ Danceability          : int  53 50 66 43 58 54 71 37 36 91 ...
##  $ Loudness..dB.         : int  -14 -11 -9 -4 -5 -9 -6 -5 -4 -3 ...
##  $ Liveness              : int  11 17 7 3 10 14 13 12 10 24 ...
##  $ Valence               : int  68 81 52 37 87 14 54 21 23 66 ...
##  $ Length..Duration.     : chr  "201" "207" "341" "269" ...
##  $ Acousticness          : int  94 17 2 0 1 0 6 0 0 0 ...
##  $ Speechiness           : int  3 7 17 4 3 2 3 14 8 7 ...
##  $ Popularity            : int  71 39 69 76 59 45 74 69 77 82 ...
```

```
class(spotify$Length..Duration.)
```

```
## [1] "character"
```

Here length.. Duration in Character, but it has to be in integer. So here i am changing the datatype.

```
spotify$Length..Duration. = as.integer(spotify$Length..Duration)
```

```
## Warning: NAs introduced by coercion
```

```
str(spotify)
```

```
## 'data.frame':    1994 obs. of  15 variables:
##  $ Index                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Title                 : chr  "Sunrise" "Black Night" "Clint Eastwood" "The Pretender" ...
##  $ Artist                : chr  "Norah Jones" "Deep Purple" "Gorillaz" "Foo Fighters" ...
##  $ Top.Genre             : chr  "adult standards" "album rock" "alternative hip hop" "alternative me
##  $ Year                  : int  2004 2000 2001 2007 2002 2004 2002 2006 2004 2002 ...
##  $ Beats.Per.Minute..BPM.: int  157 135 168 173 106 99 102 137 148 112 ...
##  $ Energy                : int  30 79 69 96 82 46 71 96 92 67 ...
##  $ Danceability          : int  53 50 66 43 58 54 71 37 36 91 ...
##  $ Loudness..dB.         : int  -14 -11 -9 -4 -5 -9 -6 -5 -4 -3 ...
##  $ Liveness              : int  11 17 7 3 10 14 13 12 10 24 ...
##  $ Valence               : int  68 81 52 37 87 14 54 21 23 66 ...
##  $ Length..Duration.     : int  201 207 341 269 256 247 257 366 223 290 ...
##  $ Acousticness          : int  94 17 2 0 1 0 6 0 0 0 ...
##  $ Speechiness           : int  3 7 17 4 3 2 3 14 8 7 ...
##  $ Popularity            : int  71 39 69 76 59 45 74 69 77 82 ...
```

The redesigned graph in between Popularity and Frequency(No. of Observations)

```
library(ggplot2)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

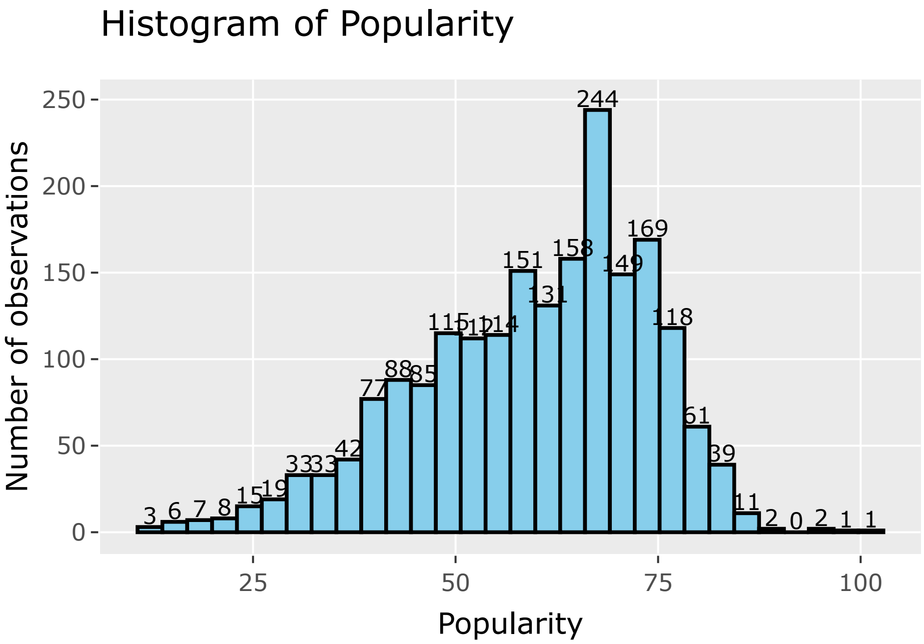
```r
popularity_plot = ggplot(spotify, aes(x = Popularity)) +
  geom_histogram(fill = "skyblue", color = "black", bins = 30) +
  labs(x = "Popularity", y = "Number of observations", title = "Histogram of Popularity") +
  stat_bin(geom = "text", aes(label = ..count.., y = ..count.. + 5), vjust = -0.5, color = "black", siz

popularity_plot = ggplotly(popularity_plot)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## i The deprecated feature was likely used in the base package.
##   Please report the issue to the authors.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

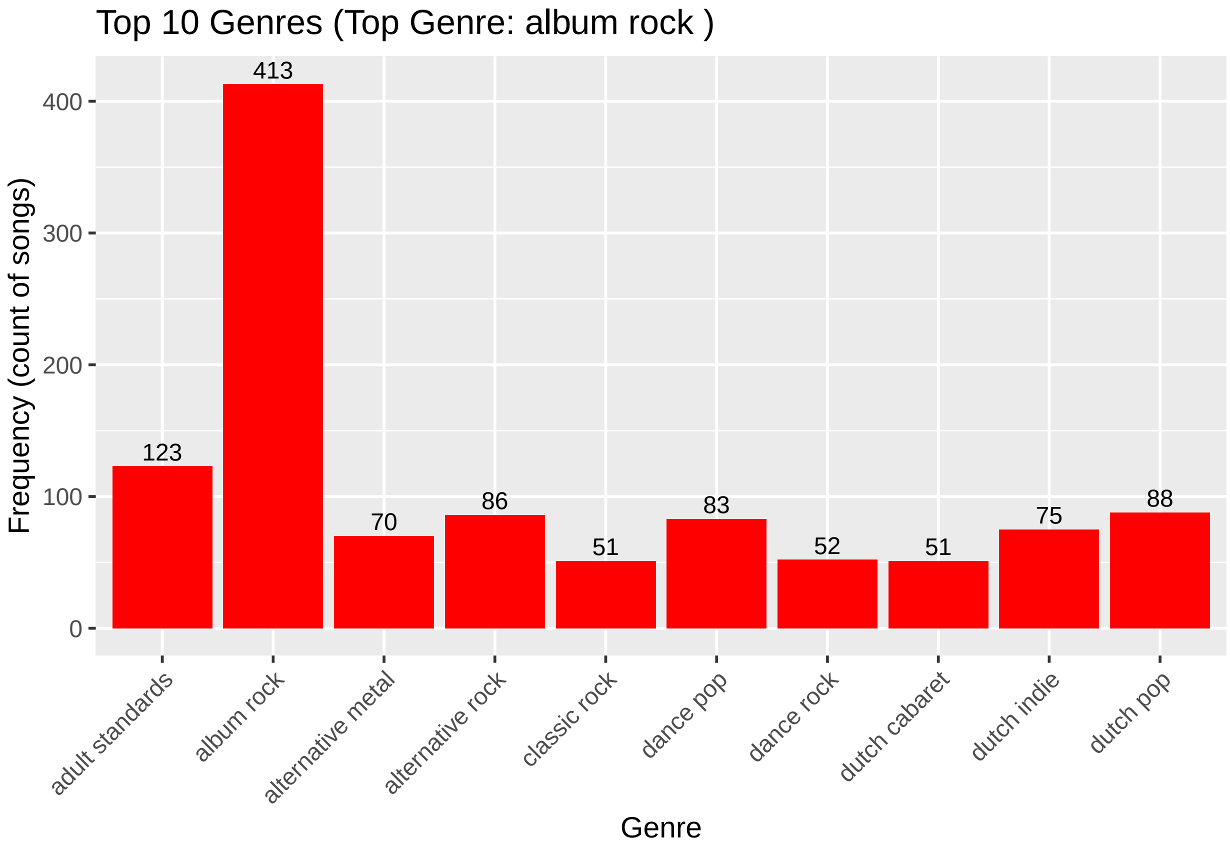
```r
popularity_plot
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, pleas
```



Here i designed with the help of plotly. which can understandable.

```r
library(ggplot2)

# Assuming spotify$Top.Genre contains the genre information

top_genres <- head(sort(table(spotify$Top.Genre), decreasing = TRUE), 10)

# Extract the name of the top first genre
top_first_genre <- names(top_genres)[1]

# Create a data frame for plotting
plot_data <- data.frame(
  genre = names(top_genres),
  count = as.numeric(top_genres)
)

# Create the plot
ggplot(plot_data, aes(x = factor(genre), y = count)) +
  geom_bar(stat = "identity", fill = "red") +
  geom_text(aes(label = count, y = count + 0.5), vjust = -0.3, color = "black", size = 3) +  # Add labe
  labs(title = paste("Top 10 Genres (Top Genre:", top_first_genre, ")"),
       x = "Genre",
       y = "Frequency (count of songs)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels
```



“‘