

 <b>VIT</b> Vidyalankar Institute of Technology ACCREDITED A+ BY NAAC	<b>Department of Information Technology</b>
--	---

Semester	B.E. Semester VIII – INFT
Subject	R programming
Laboratory Teacher:	Shruti Agrawal

Student Name	Harshala Ardekar
Roll Number	19101A2006
Grade and Subject Teacher's Signature	

<p align="center"><b>CODING TEST</b></p> <p><b>Q1. Write a R program to import a dataset and perform exploratory data analysis in it.</b>  <b>Code:</b></p>
---

```

library(tidyverse)
#install.packages("dplyr")
# import dataset of movie ratings into df variable
# This dataset is taken from kaggle
df<-read_csv("Movie Ratings.csv")
# To show the structure of data
head(df)

# Lowest Rotten Tomatoes Ratings given to movies
min(df$`Rotten Tomatoes Ratings %`)
# Highest Rotten Tomatoes Ratings given to movies
max(df$`Rotten Tomatoes Ratings %`)

# Lowest Audience Ratings given to movies
min(df$`Audience Ratings %`)# Highest Audience Tomatoes Ratings given to movies
max(df$`Audience Ratings %`)

#Range of budget
range(df$`Budget (million $)`))

# Average audience ratings
mean(df$`Audience Ratings %`)
# Average Rotten Tomatoes Rating
mean(df$`Rotten Tomatoes Ratings %`)
# Average of budget of movies as per dataset
mean(df$`Budget (million $)`))

# median of audience ratings
median(df$`Audience Ratings %`)
# median of rotten tomatoes ratings
median(df$`Rotten Tomatoes Ratings %`)

# standard deviation of budget
sd(df$`Budget (million $)`))
# variance of budget
var(df$`Budget (million $)`))

# Summary of entire data
summary(df)

# summary of the data as per the different types of movies
by(df,df$Genre,summary)

```

Output:

```

> head(df)
# A tibble: 6 x 6
  Film          Genre  `Rotten Tomatoes Ra-` `Audience Rating~` `Budget (million~` `Year of releas~
  <chr>        <chr>      <dbl>          <dbl>          <dbl>          <dbl>
1 (500) Days of Summer Comedy      87            81             8            2009
2 10,000 B.C.    Adventure     9            44            105           2008
3 12 Rounds      Action       30            52             20            2009
4 127 Hours      Adventure    93            84             18            2010
5 17 Again       Comedy      55            70             20            2009
6 2012           Action      39            63            200            2009
>
> # Lowest Rotten Tomatoes Ratings given to movies
> min(df$`Rotten Tomatoes Ratings %`)
[1] 0
> # Highest Rotten Tomatoes Ratings given to movies
> max(df$`Rotten Tomatoes Ratings %`)
[1] 97
>
> # Lowest Audience Ratings given to movies
> min(df$`Audience Ratings %`)
[1] 0
> # Highest Audience Tomatoes Ratings given to movies
> max(df$`Audience Ratings %`)
[1] 96
>
> #Range of budget
> range(df$`Budget (million $)` )
[1] 0 300
> # Average audience ratings
> mean(df$`Audience Ratings %`)
[1] 58.83096
> # Average Rotten Tomatoes Rating
> mean(df$`Rotten Tomatoes Ratings %`)
[1] 47.40391
> # Average of budget of movies as per dataset
> mean(df$`Budget (million $)` )
[1] 50.10498
>
>
> # median of audience ratings
> median(df$`Audience Ratings %`)
[1] 58
> # median of rotten tomatoes ratings
> median(df$`Rotten Tomatoes Ratings %`)
[1] 46
>
> # standard deviation of budget
> sd(df$`Budget (million $)` )
[1] 48.63848
> # variance of budhet
> var(df$`Budget (million $)` )
[1] 2365.702
> # Summary of entire data
> summary(df)
      Film          Genre  Rotten Tomatoes Ratings % Audience Ratings % Budget (million $)
Length:562      Length:562      Min.   : 0.0           Min.   : 0.00      Min.   : 0.0
Class :character Class :character 1st Qu.:25.0         1st Qu.:47.00     1st Qu.: 20.0
Mode  :character Mode  :character Median :46.0         Median :58.00     Median : 35.0
                        Mean :47.4         Mean  :58.83      Mean  : 50.1
                        3rd Qu.:70.0        3rd Qu.:72.00     3rd Qu.: 65.0
                        Max.   :97.0         Max.   :96.00     Max.   :300.0

Year of release
Min.   :2007
1st Qu.:2008
Median :2009
Mean   :2009
3rd Qu.:2010
Max.   :2011

> # summary of the data as per the different types of movies
> by(df,df$Genre,function(x){summary(x)})
df$Genre: Action
      Film          Genre  Rotten Tomatoes Ratings % Audience Ratings % Budget (million $)
Length:154      Length:154      Min.   : 4.0           Min.   :19.00     Min.   : 0.00
Class :character Class :character 1st Qu.:27.0         1st Qu.:47.00     1st Qu.: 32.25
Mode  :character Mode  :character Median :40.5         Median :57.50     Median : 69.00
                        Mean :44.4         Mean  :58.72      Mean  : 84.63
                        3rd Qu.:64.0        3rd Qu.:71.00     3rd Qu.:128.75
                        Max.   :94.0         Max.   :93.00     Max.   :300.00

Year of release
Min.   :2007
1st Qu.:2008
Median :2009
Mean   :2009
3rd Qu.:2010
Max.   :2011

-----
df$Genre: Adventure
      Film          Genre  Rotten Tomatoes Ratings % Audience Ratings % Budget (million $)
Length:29       Length:29       Min.   : 9.0           Min.   :31.00     Min.   : 12.00
Class :character Class :character 1st Qu.:30.0         1st Qu.:50.00     1st Qu.: 38.00
Mode  :character Mode  :character Median :52.0         Median :61.00     Median : 60.00
                        Mean :53.1         Mean  :62.72      Mean  : 81.48
                        3rd Qu.:76.0        3rd Qu.:78.00     3rd Qu.:125.00
                        Max.   :96.0         Max.   :87.00     Max.   :250.00

Year of release
Min.   :2007
1st Qu.:2008
Median :2009
Mean   :2009
3rd Qu.:2010
Max.   :2011

```

```

df$Genre: Comedy
  Film      Genre      Rotten Tomatoes Ratings % Audience Ratings % Budget (million $)
Length:172 Length:172 Min. : 1.00 Min. :28.00 Min. : 0.00
Class :character Class :character 1st Qu.:22.00 1st Qu.:46.00 1st Qu.: 18.75
Mode :character Mode :character Median :43.00 Median :56.00 Median : 30.00
Mean :44.92 Mean :56.41 Mean : 36.11
3rd Qu.:68.00 3rd Qu.:67.25 3rd Qu.: 50.50
Max. :97.00 Max. :93.00 Max. :175.00

Year of release
Min. :2007
1st Qu.:2008
Median :2009
Mean :2009
3rd Qu.:2010
Max. :2011

df$Genre: Drama
  Film      Genre      Rotten Tomatoes Ratings % Audience Ratings % Budget (million $)
Length:101 Length:101 Min. : 5.00 Min. :20.00 Min. : 0.00
Class :character Class :character 1st Qu.:34.00 1st Qu.:52.00 1st Qu.: 15.00
Mode :character Mode :character Median :56.00 Median :66.00 Median : 21.00
Mean :56.48 Mean :64.43 Mean : 27.85
3rd Qu.:79.00 3rd Qu.:79.00 3rd Qu.: 35.00
Max. :97.00 Max. :91.00 Max. :150.00

Year of release
Min. :2007
1st Qu.:2008
Median :2009
Mean :2009
3rd Qu.:2010
Max. :2011

df$Genre: Horror
  Film      Genre      Rotten Tomatoes Ratings % Audience Ratings % Budget (million $)
Length:49 Length:49 Min. : 0.00 Min. : 0.00 Min. : 1.00
Class :character Class :character 1st Qu.:11.00 1st Qu.:38.00 1st Qu.:10.00
Mode :character Mode :character Median :25.00 Median :48.00 Median :20.00
Mean :34.57 Mean :47.39 Mean :21.67
3rd Qu.:57.00 3rd Qu.:57.00 3rd Qu.:32.00
Max. :92.00 Max. :78.00 Max. :50.00

Year of release
Min. :2007
1st Qu.:2008
Median :2009
Mean :2009
3rd Qu.:2011
Max. :2011

df$Genre: Romance
  Film      Genre      Rotten Tomatoes Ratings % Audience Ratings % Budget (million $)
Length:21 Length:21 Min. : 0.00 Min. : 0.00 Min. : 0.00
Class :character Class :character 1st Qu.:21.00 1st Qu.:50.00 1st Qu.: 16.00
Mode :character Mode :character Median :49.00 Median :68.00 Median : 30.00
Mean :48.19 Mean :62.33 Mean : 33.76
3rd Qu.:71.00 3rd Qu.:77.00 3rd Qu.: 45.00
Max. :93.00 Max. :84.00 Max. :110.00

Year of release
Min. :2007
1st Qu.:2008
Median :2011
Mean :2009
3rd Qu.:2011
Max. :2011

df$Genre: Thriller
  Film      Genre      Rotten Tomatoes Ratings % Audience Ratings % Budget (million $)
Length:36 Length:36 Min. :11.00 Min. :24.00 Min. : 4.00
Class :character Class :character 1st Qu.:32.00 1st Qu.:58.50 1st Qu.: 20.75
Mode :character Mode :character Median :68.00 Median :68.50 Median : 37.50
Mean :59.08 Mean :65.58 Mean : 54.67
3rd Qu.:84.25 3rd Qu.:74.25 3rd Qu.: 77.50
Max. :94.00 Max. :96.00 Max. :185.00

Year of release
Min. :2007
1st Qu.:2007
Median :2008
Mean :2009
3rd Qu.:2011
Max. :2011

```

**Q2. Write a R program to make various plots.**

**Code:**

```

library(tidyverse)
#install.packages("dplyr")
# import dataset of movie ratings into df variable
# This dataset is taken from kaggle

df<-read_csv("Movie Ratings.csv")

# To show the structure of data
head(df)

# Variation of scatter Rotten tomatoes ratings and Audience Ratings
ggplot(data=df,aes(x=`Rotten Tomatoes Ratings %`,y=`Audience Ratings %`,color=`Audience Ratings %`)) +
  geom_point()

# Variation of Rotten tomatoes Ratings with the budget
rotten_tomatoes_ratings = c(df$`Rotten Tomatoes Ratings %`)
budget = c(df$`Budget (million $)`)
datas <- data.frame(rotten_tomatoes_ratings,budget)

ggplot(datas, aes(x = rotten_tomatoes_ratings, y = budget )) +
  geom_line(color = 4, # Color of the line
            lwd = 1, # width of the line
            linetype = 1) # Line type

# how budget varies with the different types of movies
ggplot(data=df,aes(x=`Budget (million $)`) +
  geom_density(color='black',stack=10,aes(fill=Genre))

# To find frequency of different types of movies in a dataset
datas <- data.frame(table(df$Genre))
ggplot(datas, aes(x=Var1, y=Freq))+
  geom_bar(width = 1, stat = "identity",fill="steelblue")

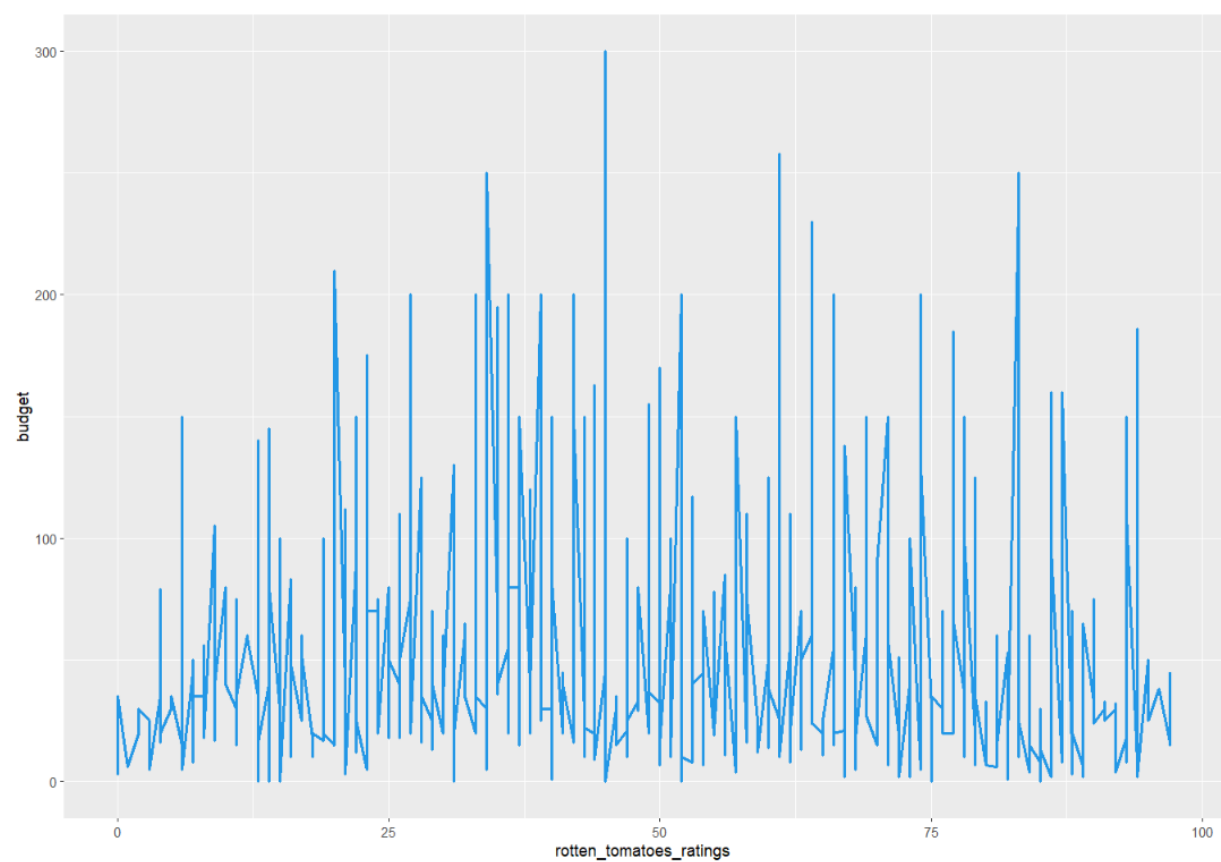
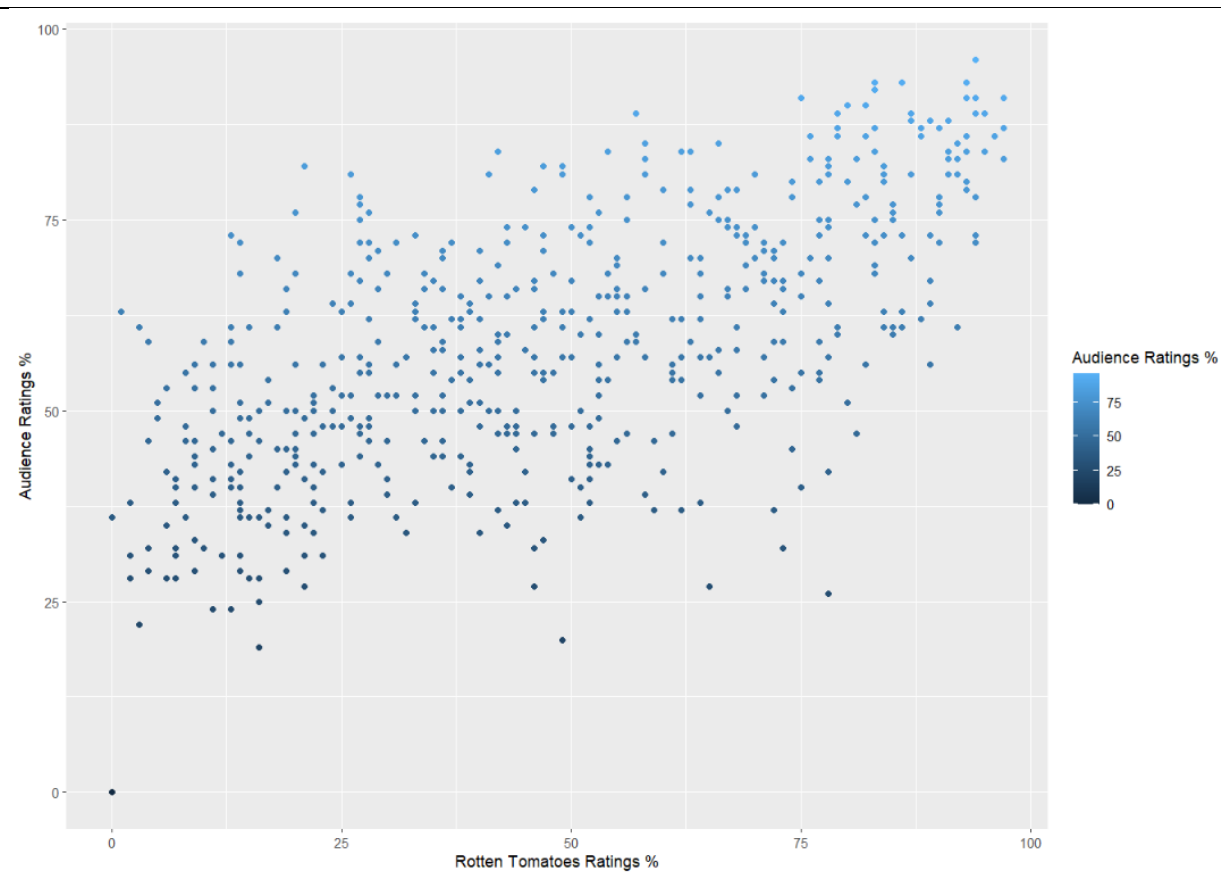
# Audience rating in pie chart representation
rat1 = 0
rat2 = 0
rat3 = 0
rat4 = 0

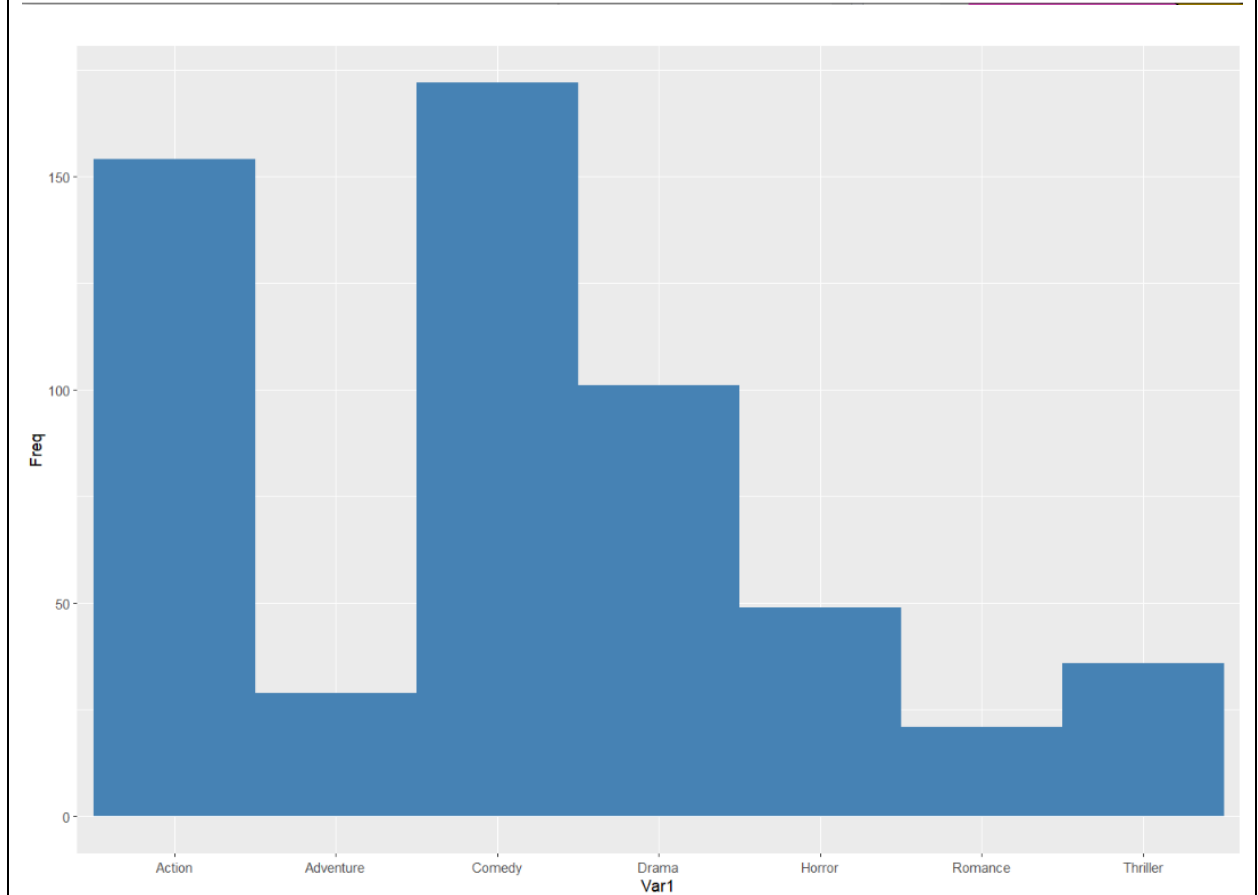
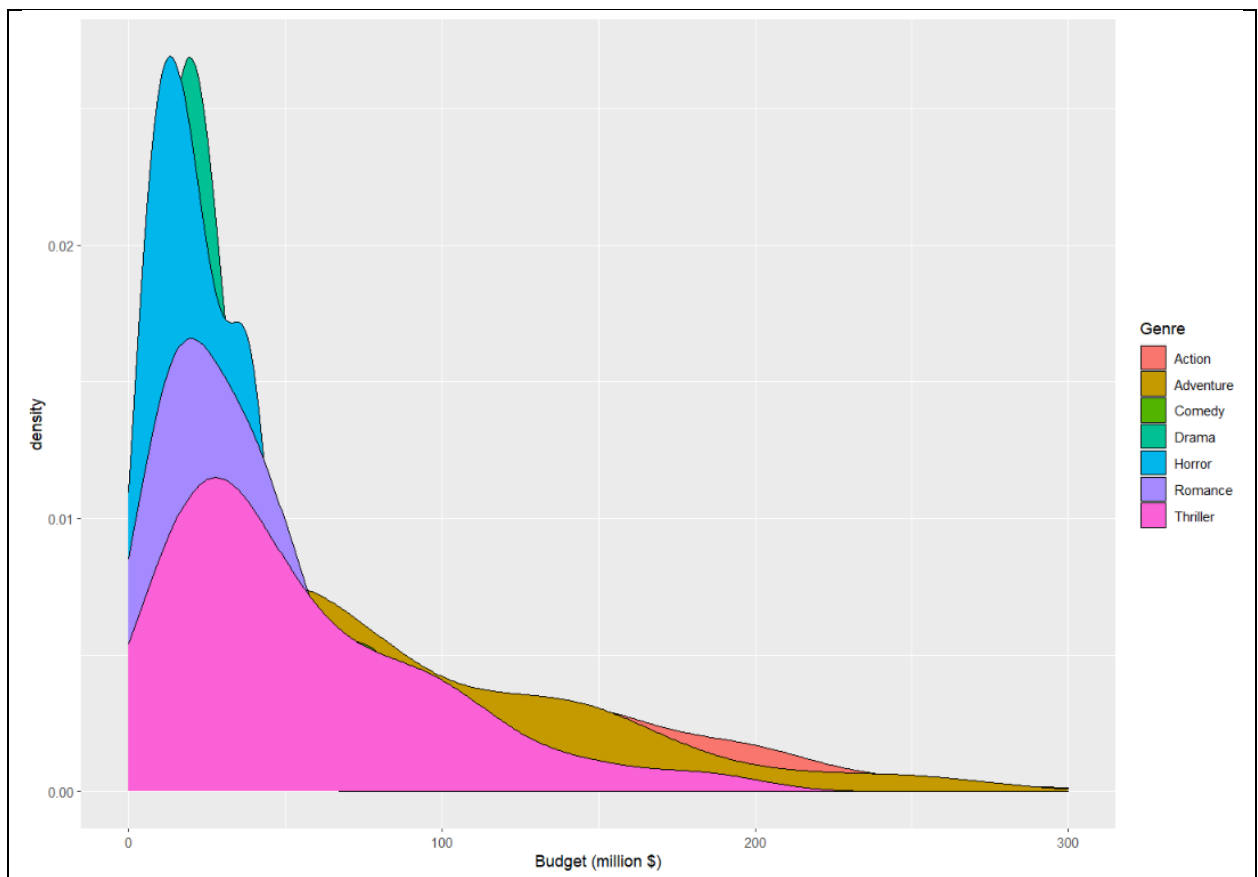
for (item in df$`Audience Ratings %`) {
  if (item>74){
    rat1 = rat1 +1
  }else if (item>49){
    rat2 = rat2 + 1
  }else if (item>24){
    rat3 = rat3 + 1
  }else{
    rat4 = rat4 + 1
  }
}
print(rat1)

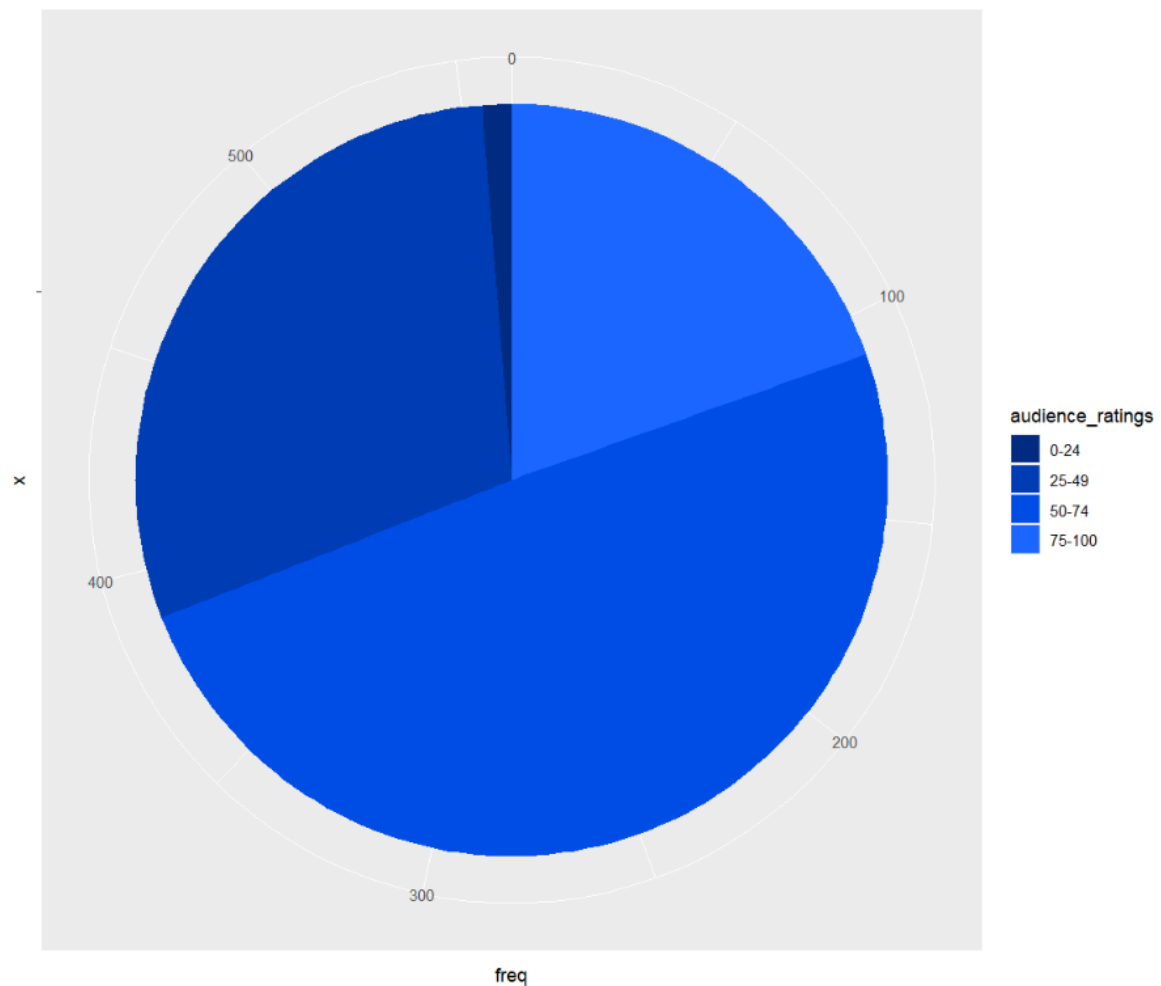
audience_ratings <- c('75-100','50-74','25-49','0-24')
freq <- c(rat1,rat2,rat3,rat4)
audience_freq_count <- data.frame(audience_ratings,freq)
ggplot(data = audience_freq_count, aes( y = freq, fill = audience_ratings),col=c("chartreuse", "blue4"))
  geom_bar(stat = "identity") +
  scale_fill_manual(values=c("#002b80", "#003cb3", "#004de6", "#1a66ff"))+
  coord_polar("y")

```

**Output:**







**Q3:** Write a R Program to find the sum of natural numbers using recursion

**Input:**

```
sum = function(n) {  
  if(n <= 1) {  
    return(1)  
  } else {  
    return(n + sum(n-1))  
  }  
}  
n = as.integer(readline("Enter the number:"))  
print("Sum is:")  
print(sum(n))
```

**Output:**

```
Enter the number:15  
[1] "Sum is:"  
[1] 120
```



**Q4:** Write a R Program to demonstrating the use of aggregate function

**Input:**

```
data = data.frame(subjects=c("java", "python", "java",  
                             "java", "php", "php"),  
                  id=c(1, 2, 3, 4, 5, 6),  
                  names=c("manoj", "sai", "mounika",  
                           "durga", "deepika", "roshan"),  
                  marks=c(89, 89, 76, 89, 90, 67))  
  
print(data)  
  
print(aggregate(data$marks, list(data$subjects), FUN=mean))
```

**Output:**

```
  subjects id  names marks  
1    java  1  manoj    89  
2  python  2    sai    89  
3    java  3 mounika    76  
4    java  4   durga    89  
5     php  5 deepika    90  
6     php  6   roshan    67  
Group.1      x  
1    java 84.66667  
2     php 78.50000  
3  python 89.00000
```