ProblemSet3.R

DavidAarhus

2020-02-28

```r
rm(list = ls()) #removing all variables

#Question 1a
movies <- read.csv("/Users/DavidAarhus/Documents/310 R/Datasets/movie_metadata.csv") #loads dataset

#Question 1b
names(movies) #prints the names of all the columns
```

```
##  [1] "color"                     "director_name"
##  [3] "num_critic_for_reviews"    "duration"
##  [5] "director_facebook_likes"   "actor_3_facebook_likes"
##  [7] "actor_2_name"              "actor_1_facebook_likes"
##  [9] "gross"                     "genres"
## [11] "actor_1_name"              "movie_title"
## [13] "num_voted_users"           "cast_total_facebook_likes"
## [15] "actor_3_name"              "facenumber_in_poster"
## [17] "plot_keywords"             "movie_imdb_link"
## [19] "num_user_for_reviews"      "language"
## [21] "country"                   "content_rating"
## [23] "budget"                    "title_year"
## [25] "actor_2_facebook_likes"    "imdb_score"
## [27] "aspect_ratio"              "movie_facebook_likes"
```

```r
#Question 1c
missingvalues <-  sum(is.na(movies$budget)) #counts missing values
missingvalues
```

```
## [1] 492
```

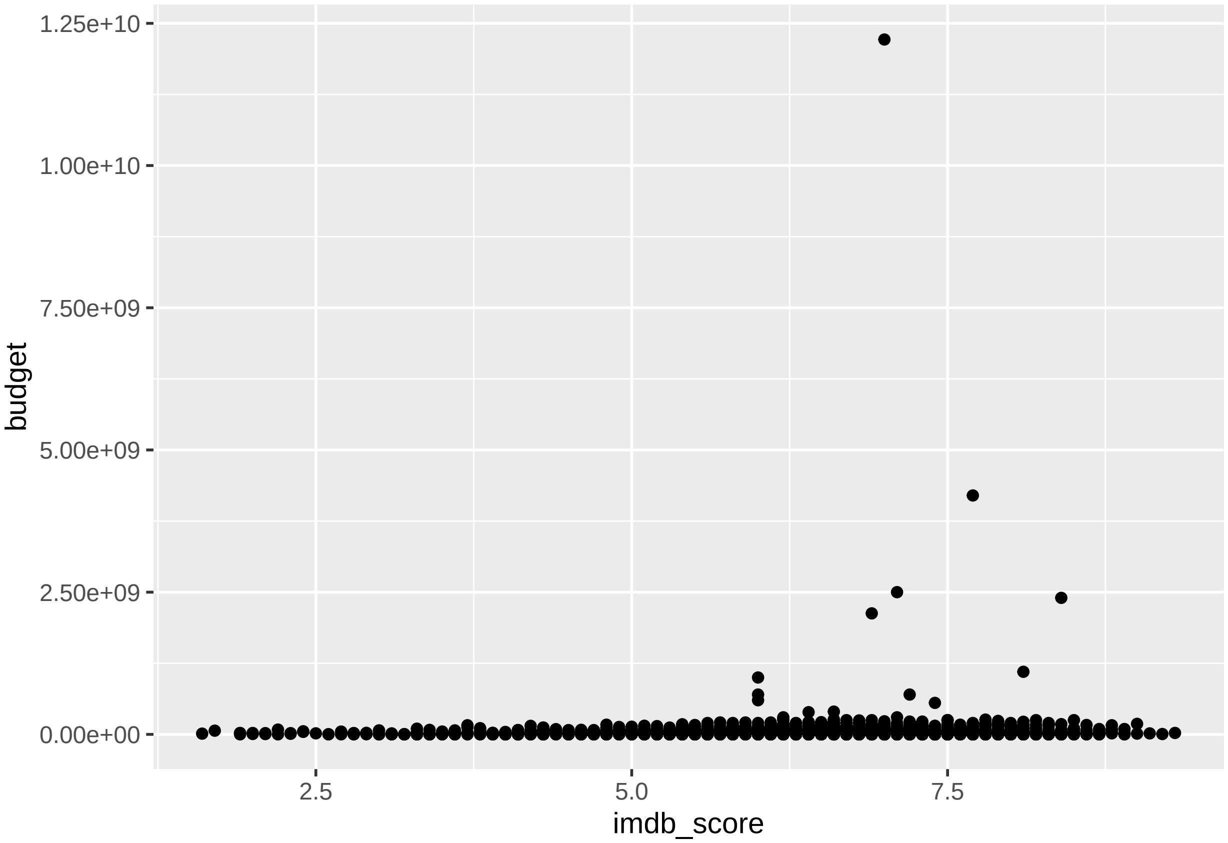
```r
movies <- movies[!is.na(movies$budget),] #removes missing balues in budget
dim(movies) #lists the dimensions of the new movies dataset
```

```
## [1] 4551   28
```

```r
#Question 1d
length(unique(movies$director_name, incomparables = FALSE)) #counts the amount of unique directors in t
```

```
## [1] 2175
```

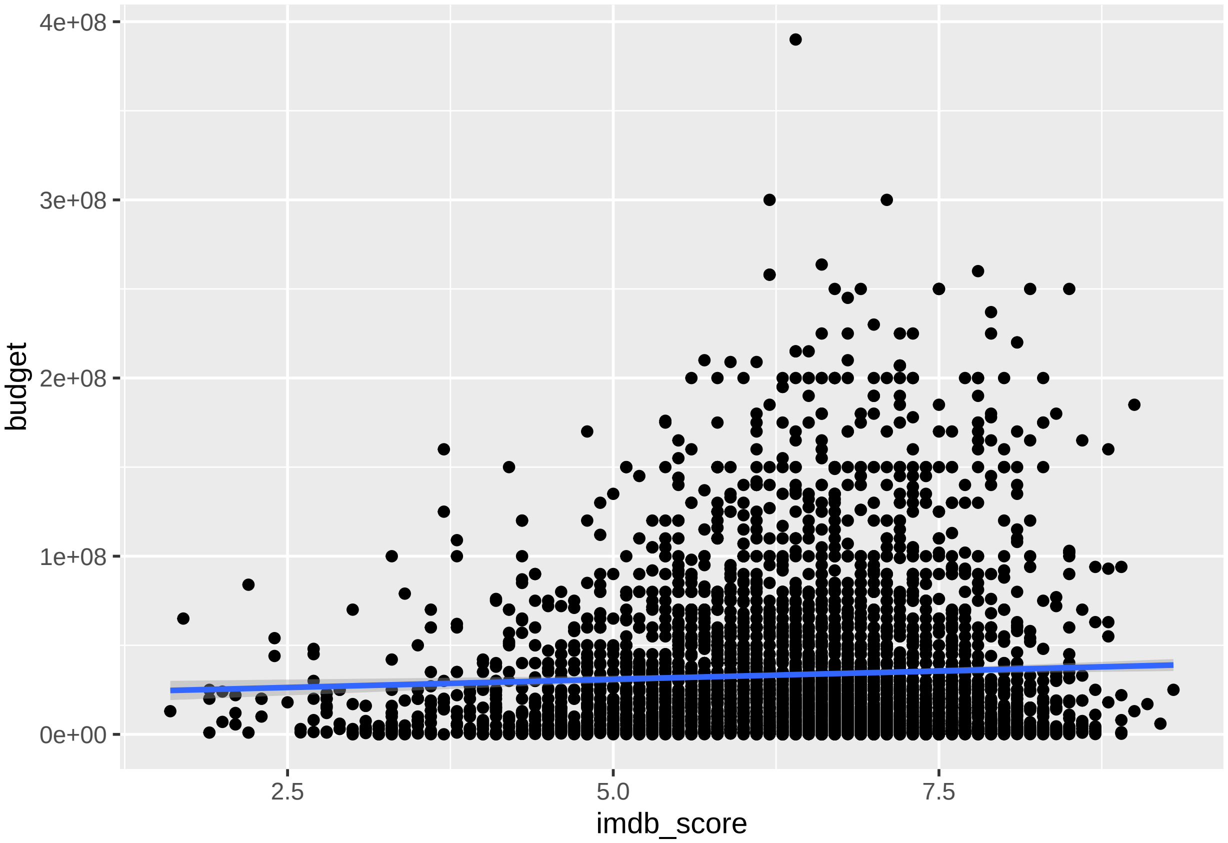
```r
#Question 1e
library("ggplot2") #loads ggplot library
ggplot(movies, aes(imdb_score, budget)) + geom_point() #prints off scatterplot
```



```
#Question 1f
movies <- movies[movies$budget<400000000,] #removes rows with movie budgets over 400m
nrow(movies) #4539 movies in data set
```

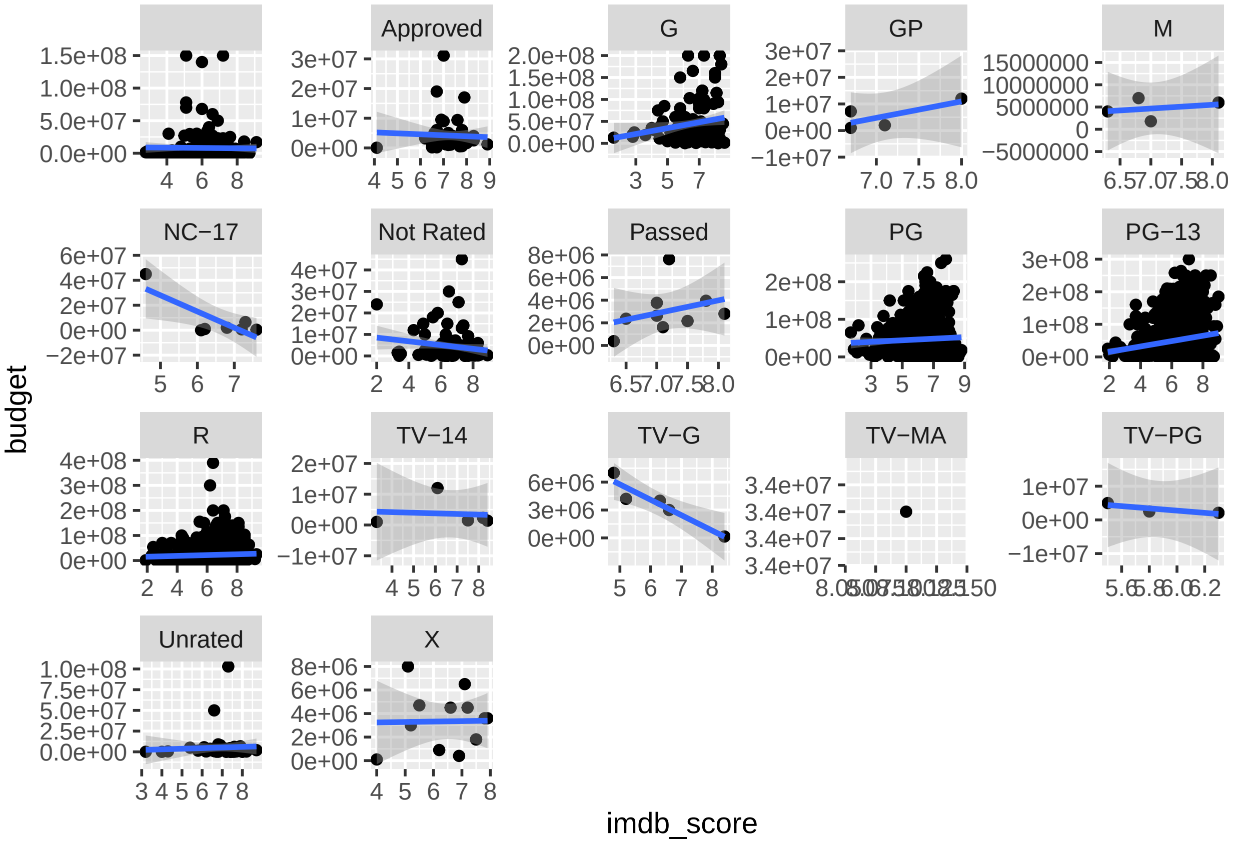
```
## [1] 4539
```

```
#Question 1g
ggplot(movies, aes(imdb_score, budget)) +
  geom_point() +
  geom_smooth(method = 'lm') #creates linear trendline for imdb and budget
```



```
#there is no definitive explanation for a relationship between the two variables.
#Only a slight positive slope in the trendline.

#Question 1h
ggplot(movies, aes(imdb_score, budget)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  facet_wrap(~content_rating, scales = "free")
```



```
#if we are looking soley at relationship strength,
#TV-G and NC-17 have a strong negative relationship.
#However they do not have alot of data points.
#If the amount of data points matter, PG-13 has the strongest (positive) relationship


#Question 2
# to create budget and gross columns in millions
movies$grossM <- movies$gross/1e+6
movies$budgetM <- movies$budget/1e+6 # note how we created new columns
# to create a column for main genre
movies$genre_main <- do.call('rbind',strsplit(as.character(movies$genres), '|', fixed=TRUE))[,1]
```

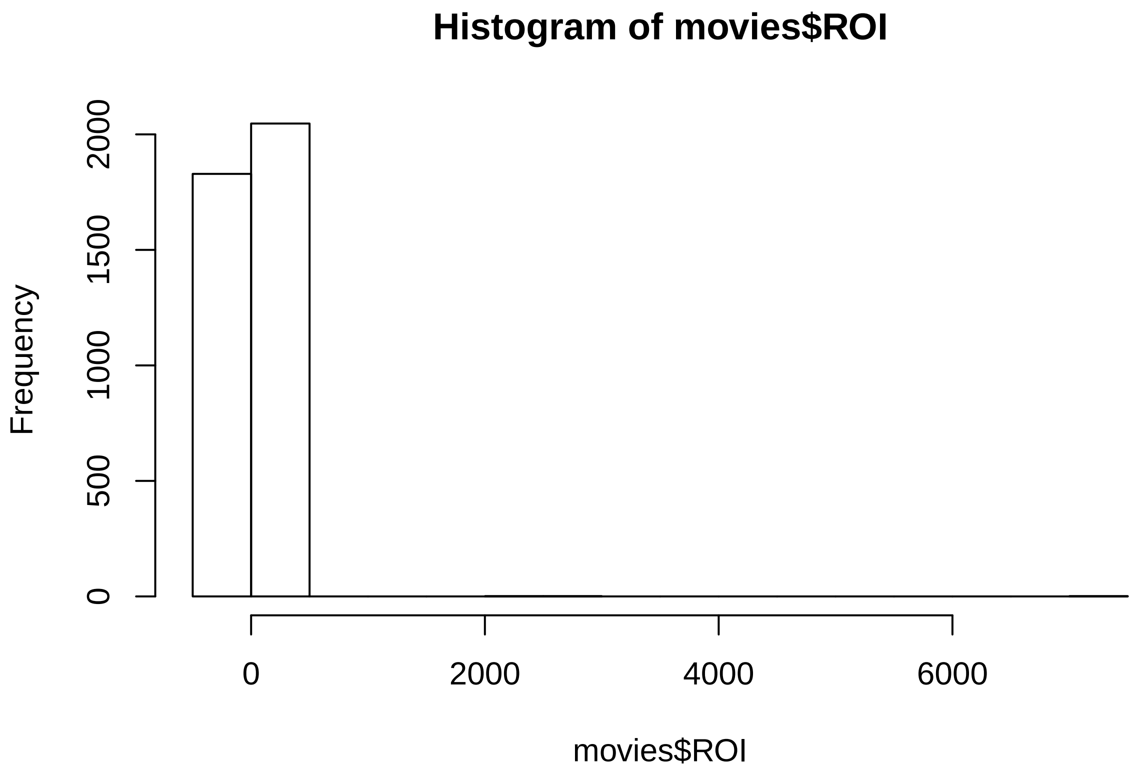
```
## Warning in rbind(c("Action", "Adventure", "Fantasy", "Sci-Fi"), c("Action", :
## number of columns of result is not a multiple of vector length (arg 2)
```

```
#Question 2a
movies$profitM <- movies$grossM - movies$budgetM #creates profit margin
movies$ROI <- movies$profitM/movies$budgetM #creates ROI margin

#Question 2b
mean(movies$ROI, na.rm= TRUE) #average ROI
```

```
## [1] 5.273088
```

```
#Question 2c
hist(movies$ROI) #creates histogram for ROI in movie dataset
```



```
#Question 2d
sum(movies$ROI > 10, na.rm = TRUE) #counts movies with ROI greater than 10
```

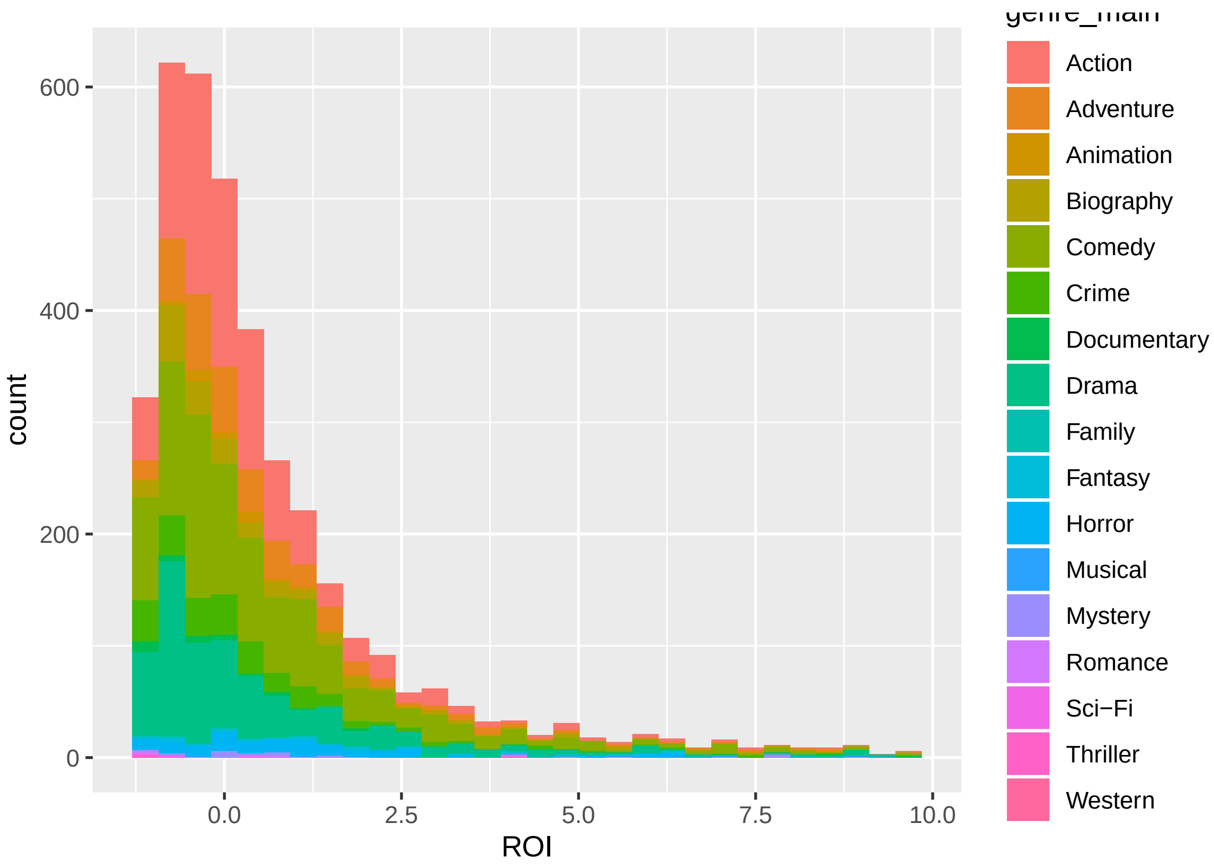
```
## [1] 145
```

```
movies <- movies[movies$ROI<10,] #removes Movies with ROI greater than 10

#Question 2e
ggplot(movies, aes(ROI, fill = genre_main)) +
  geom_histogram() #new histogram
```

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

```
## Warning: Removed 660 rows containing non-finite values (stat_bin).
```



```
#Question 2f
library("doBy")
summaryBy(ROI ~ genre_main, movies, FUN = mean) #creates a summary that gives the mean ROI for each fil
```

```
##     genre_main   ROI.mean
## 1       Action  0.3146972
## 2    Adventure  0.6117778
## 3    Animation  0.4749139
## 4    Biography  0.6730581
## 5       Comedy  0.7502510
## 6        Crime  0.4230916
## 7  Documentary  0.2681136
## 8        Drama  0.5484959
## 9       Family -0.5971447
## 10     Fantasy  2.0929081
## 11      Horror  1.3994674
## 12     Musical  6.4089710
## 13     Mystery  1.3665859
## 14     Romance  1.1126902
## 15      Sci-Fi  0.3892234
## 16    Thriller  2.3503454
## 17     Western  5.4029778
## 18        <NA>         NA
```

```
genre_mean <- summaryBy(ROI ~ genre_main, movies, FUN = mean) #assigns mean genre list to an object
max(genre_mean[,2], na.rm= TRUE ) #gives highest ROI
```

```
## [1] 6.408971
```

```
which(genre_mean$ROI.mean == max(genre_mean[,2], na.rm = TRUE )) #identifies the row of which genre has
```

```
## [1] 12
```

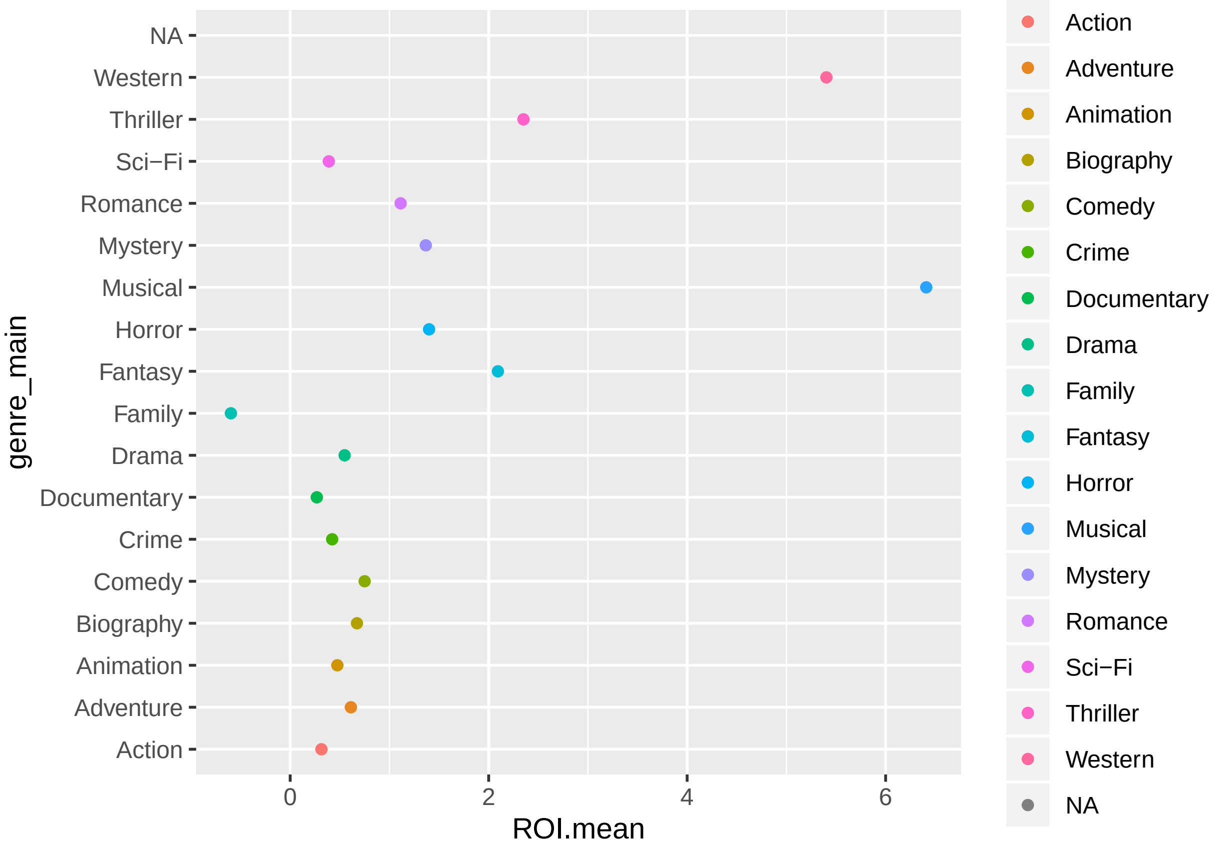
```
genre_mean[which(genre_mean$ROI.mean == max(genre_mean[,2], na.rm= TRUE )),] #identifies the genre name
```

```
##    genre_main ROI.mean
## 12    Musical 6.408971
```

```
#Musical genres have the highest ROI

#Question 2g
ggplot(genre_mean, aes(ROI.mean, genre_main, color = genre_main)) +
  geom_point() #creates scatterplot that shows the variety in mean ROI amongst genres
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



```
#Question 3a
set.seed(310)
train_idx <- sample(1:nrow(movies), size = 0.80*nrow(movies), replace = FALSE)
movies.training <- movies[train_idx, ]
movies.test <- movies[-train_idx, ]

#Question 3b
dim(movies) #checks the dimensions
```

```
## [1] 4394   33
```

```
dim(movies.training)
```

```
## [1] 3515   33
```

```
dim(movies.test)
```

```
## [1] 879  33
```

```
#Question 3c
mod1 <- lm(profitM ~ imdb_score, movies.training) #estimating our model using the training dataset
summary(mod1)
```

```
##
## Call:
## lm(formula = profitM ~ imdb_score, data = movies.training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -386.36  -24.47   -9.02   14.59  495.25
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -65.292      5.812  -11.23   <2e-16 ***
## imdb_score    11.842      0.890   13.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.79 on 2989 degrees of freedom
##   (524 observations deleted due to missingness)
## Multiple R-squared:  0.05592,    Adjusted R-squared:  0.0556
## F-statistic:   177 on 1 and 2989 DF,  p-value: < 2.2e-16
```

```
#Question 3d
coef(mod1)
```

```
## (Intercept)  imdb_score
##   -65.29241    11.84167
```

```
#These coefficents show that as the imdb score for a film increases,
#the profit of the Movie will also increase
```