Estimating the variables that influence a films sucess in IMDB

Group 09

```
library(ggplot2)
library(tidyverse)
library(skimr)
library(moderndive)
library(gapminder)
library(sjPlot)
library(stats)
library(jtools)
library(kableExtra)
library(GGally)
library(caret)
library(pROC)
library(janitor)
```

0.1 Introduction

The following analysis aims to understand the relationship between a set of descriptive variables about a film and its success measured by its respective IMDB rating.

The central question around this analysis will be the following: **Which properties of films influence whether they are rated by IMDB as greater than 7 or not?**

From this question it is established that the target variable will be binary and hence a Logistic Regression model seems reasonable for this scenario. It is also established that missing variables (in case they are found) will be inputted with a summary statistic like mean or median if the distribution of this subset is similar to that of the complete data set, otherwise they will be deleted if they do not represent a large portion of the data set.

Throughout this analysis a full model will be fitted taking into account all numerical and categorical variables in the data set. Then the best performing model will be selected and it will only include those variables which are found to be significant.

Finally, a short summary of the model and answers to the analysis question will be found in the conclusion section.

0.2 Data Cleaning

The film data set obtained from IMDB contains the following variables:

- film.id - The unique identifier for the film
- year - Year of release of the film in cinemas
- length - Duration (in minutes)
- budget - Budget for the films production (in $1000000s)
- votes - Number of positive votes received by viewers
- genre - Genre of the film
- rating - IMDB rating from 0-10

```r
#Read data set
film <- read.csv("dataset09.csv") %>%
  mutate(target = ifelse(rating>7, 1, 0)) %>%  #Define target variable
  mutate(Rating = ifelse(rating>7, ">7", "<=7"))  #Define Rating variable help us get better da

#Create summary
film %>%
  skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	3001
Number of columns	9
Column type frequency:	
character	2
numeric	7
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
genre	0	1	5	11	0	7	0
Rating	0	1	2	3	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
film_id	0	1.00	29709.49	17071.72	16.0	14874.00	29673.0	44660.0	58753.0	
year	0	1.00	1975.88	24.13	1895.0	1957.00	1983.0	1997.0	2005.0	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
length	127	0.96	81.57	39.54	1.0	71.25	90.0	100.0	555.0	
budget	0	1.00	11.98	2.97	1.2	10.10	12.1	14.0	23.4	
votes	0	1.00	655.83	3780.10	5.0	11.00	30.0	118.0	92437.0	
rating	0	1.00	5.40	2.07	0.8	3.70	4.7	7.8	9.2	
target	0	1.00	0.35	0.48	0.0	0.00	0.0	1.0	1.0	

It is now established that film_id will not be used as an explanatory variable since it is only an identifier for the film, rather than an informative feature about it. Genre is the only categorical variable contained in the data set. Year, length, budget, and votes are the numerical explanatory variables to be tested in this analysis.

When it comes to the data set, there seems to be an issue with the length variable as there are 127 rows where this information is missing.

```
#Group by genre and select the variables 'genre' and 'length'
film %>%
  group_by(genre) %>%
  select(genre, length) %>%
  skim()
```

Table 4: Data summary

Name	Piped data
Number of rows	3001
Number of columns	2
Column type frequency:	
numeric	1
Group variables	genre

Variable type: numeric

skim_variable	genre	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
length	Action	34	0.96	94.67	29.67	2	84.00	91	100.0	480	
length	Animation	4	0.98	13.89	20.50	1	7.00	7	8.0	97	
length	Comedy	39	0.95	82.50	31.15	1	78.00	90	100.0	181	
length	Documentary	5	0.97	70.85	43.21	1	43.50	75	90.0	278	
length	Drama	40	0.95	94.59	33.27	4	85.25	96	107.0	555	
length	Romance	3	0.90	90.21	44.47	12	76.75	97	108.5	189	
length	Short	2	0.98	15.54	9.12	2	10.00	13	20.0	44	

It is evident from the summary table above that the length distribution is not equal among different film genres and therefore the missing film lengths will be handled by adding the median film length by genre to its corresponding missing columns (the mean is not used to avoid outlier influence). The different behaviour between genre and film length was expected, especially because one category is called "Short".

```
#Median length of each genre
film.median <- film %>%
  group_by(genre) %>%
  select(genre, length) %>%
  summarise(median.length = median(length, na.rm=TRUE))
film.median
```

```
# A tibble: 7 x 2
  genre       median.length
  <chr>               <dbl>
1 Action                 91
2 Animation               7
3 Comedy                 90
4 Documentary            75
5 Drama                  96
6 Romance                97
7 Short                  13
```

```
#Input corresponding genre median for length missing values
film <- film %>%
  inner_join(film.median,by=join_by(genre)) %>%
  mutate(had_NAS=ifelse(is.na(length),TRUE,FALSE),length=ifelse(is.na(length),median.length,len
  select(-median.length)
```

0.3 Exploratory Analysis

The last step before fitting the Logistic Regression model is analysing the data set to identify possible patterns.

```
film$Rating <- as.factor(film$Rating)
ggpairs(film[,c(2,3,4,5,9)], aes(colour = Rating, alpha = 0.4), title="Pair plots")
```

Figure 1: Graphical and numerical summaries of the relationships between pairs of variables

In the plot above we can check the correlation between the different covariates. They all maintain a low correlation coefficient and their scatter plots do not seem to show any linear relationship between them. This means these variables can be included in a logistic regression model without suspecting multicolinearity.

```r
# To show original counts
film %>%
  tabyl(genre, Rating) %>%
  adorn_percentages() %>%
  adorn_pct_formatting() %>%
  adorn_ns()
```

```
       genre          <=7            >7
      Action 87.2% (755) 12.8                                         (111)
   Animation 29.3%  (55) 70.7                                         (133)
      Comedy 37.8% (276) 62.2                                         (455)
 Documentary 12.0%  (22) 88.0                                         (162)
       Drama 93.8% (820)  6.2%  (54)
     Romance 87.1%  (27) 12.9%   (4)
       Short  0.8%   (1) 99.2                                         (126)
```

```
#Proportion of films with rating >7 by genre
ggplot(film, aes(x= Rating,  y = after_stat(prop), group=genre, fill=genre)) +
    geom_bar(position="dodge", stat="count") +
    labs(y = "Proportion")
```

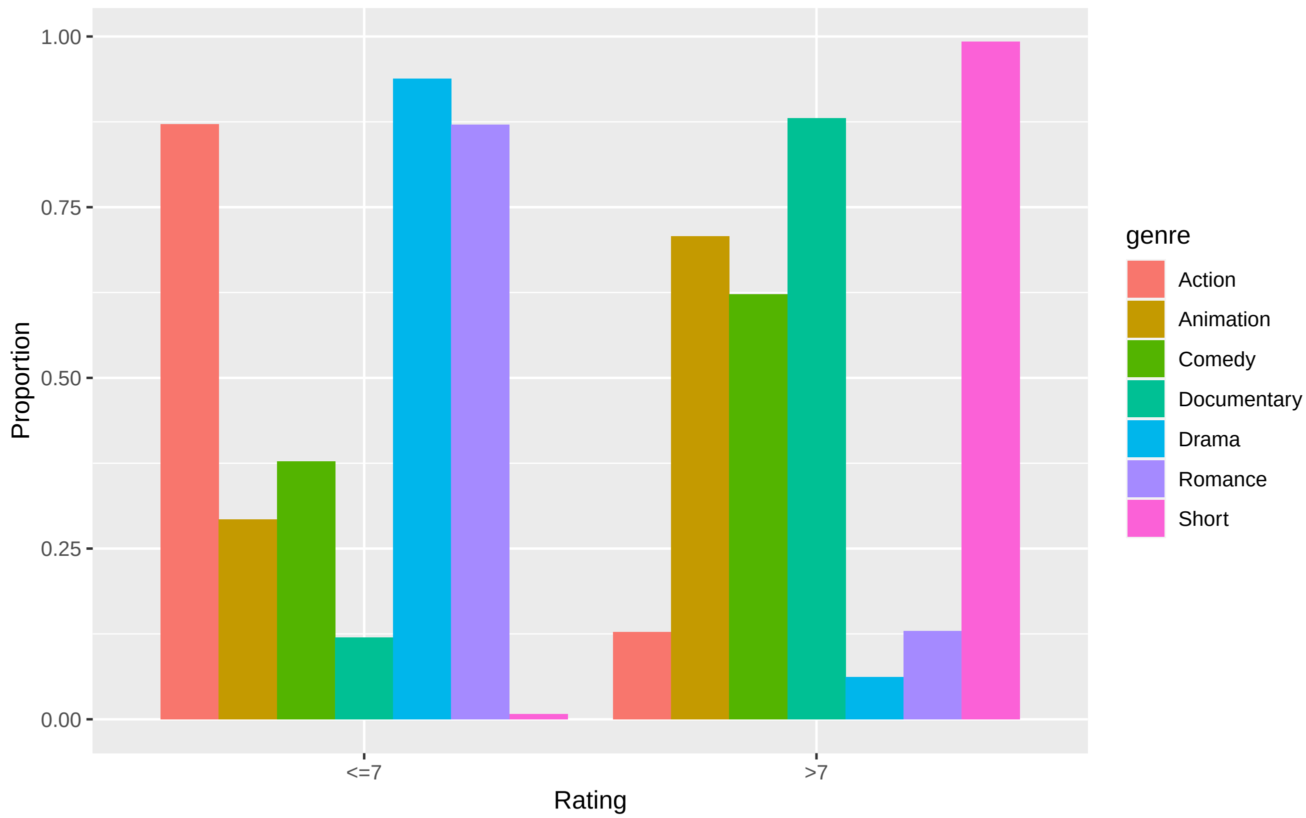


Figure 2: Proportion of Movie Ratings by Genre

```
#Plot target variable against year covariate
film %>% ggplot(aes(x=Rating, y=year, colour=Rating)) +
  geom_boxplot() +
  theme(legend.position="none") +
  labs(x="Rating", y="Year")
```

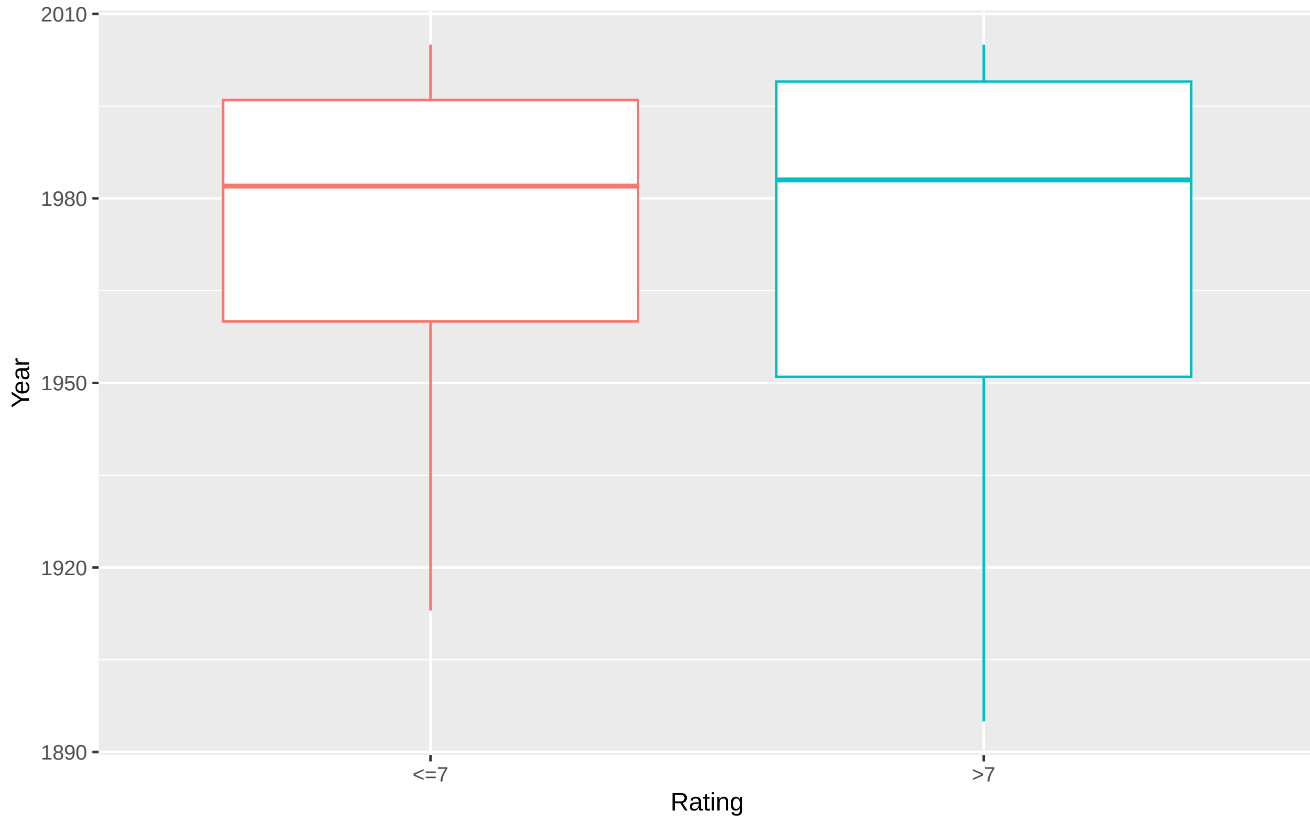


Figure 3: Boxplot of Year by Rating

```
#Plot target variable against length covariate
film %>% ggplot(aes(x=Rating, y=length, colour=Rating)) +
  geom_boxplot() +
  theme(legend.position="none") +
  labs(x="Rating", y="Length")
```

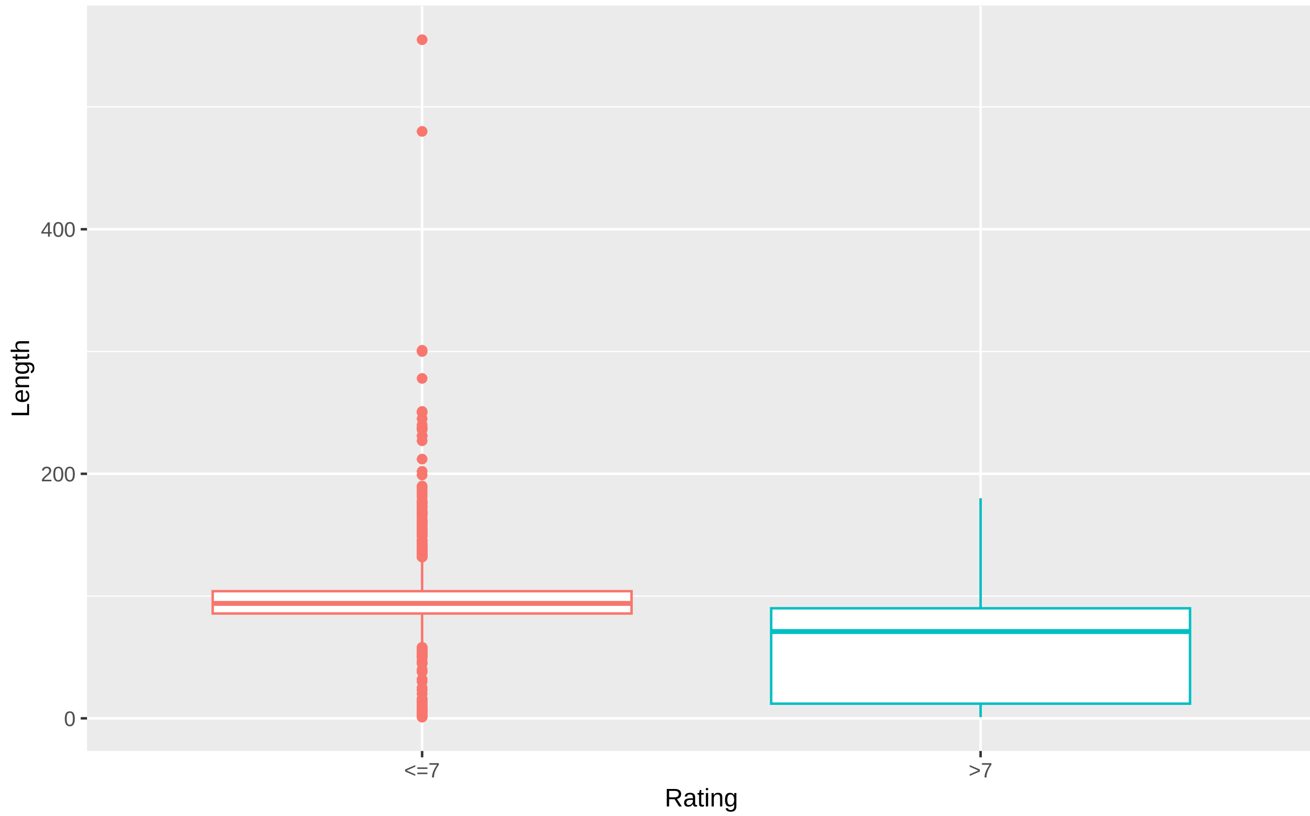


Figure 4: Boxplot of Length by Rating

```
#Plot target variable against budget covariate
film %>% ggplot(aes(x=Rating, y=budget, colour=Rating)) +
  geom_boxplot() +
  theme(legend.position="none") +
  labs(x="Rating", y="Budget")
```

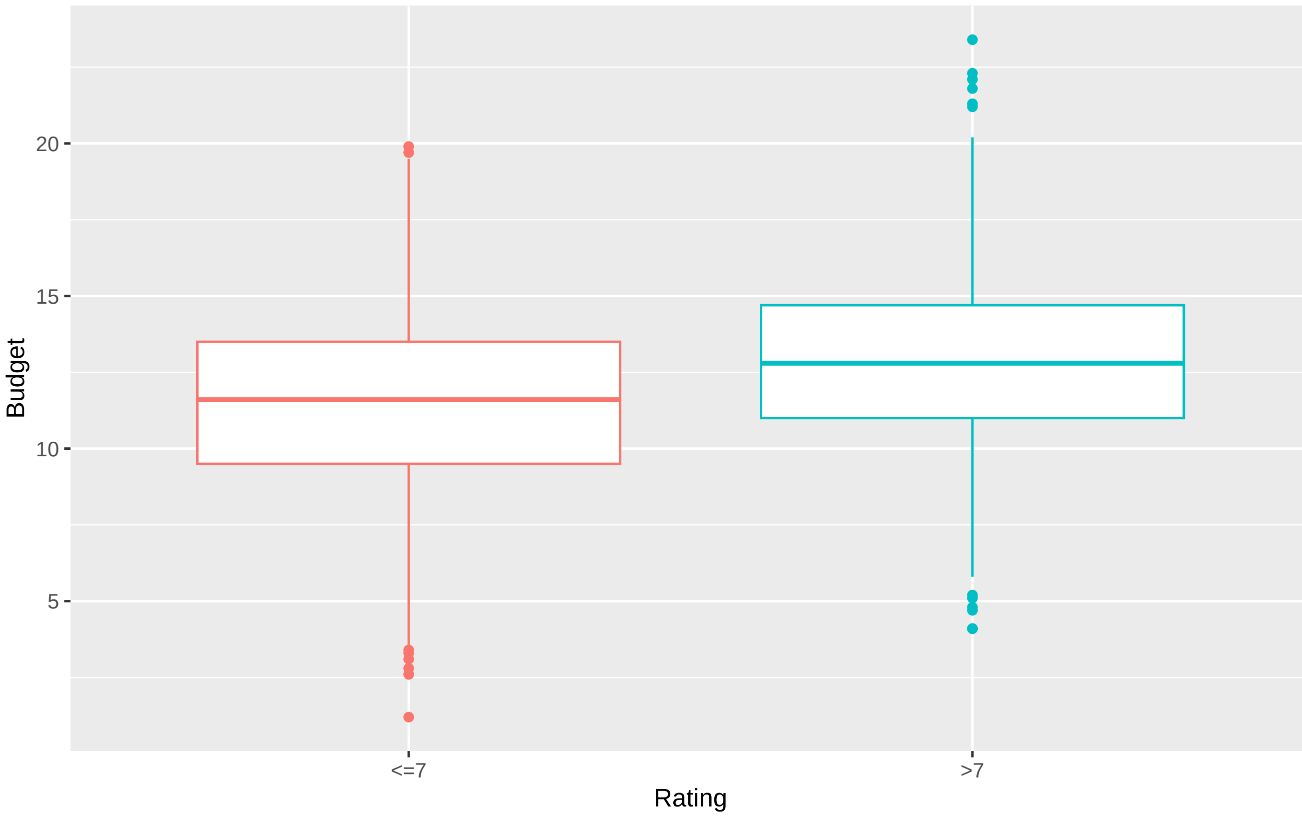


Figure 5: Boxplot of Budget by Rating

```
#Plot target variable against votes covariate
film %>% ggplot(aes(x=Rating, y=votes, colour=Rating)) +
  geom_boxplot() +
  theme(legend.position="none") +
  labs(x="Rating", y="Votes")
```

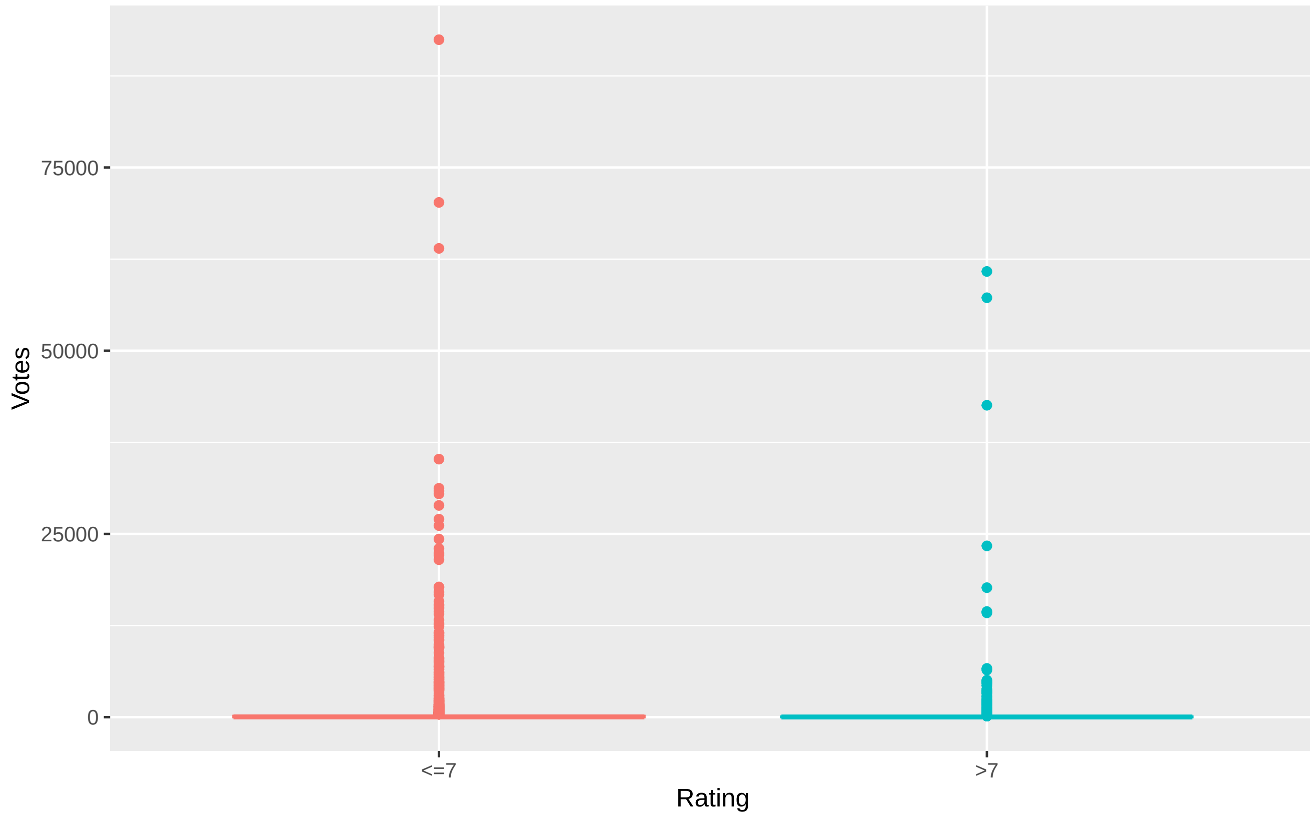


Figure 6: Boxplot of Votes by Rating

```
#Count by genre
film %>%
  ggplot(aes(x=genre, colour=genre)) +
  geom_bar() +
  theme(legend.position="none") +
  labs(y="Count", x="Film genre")
```

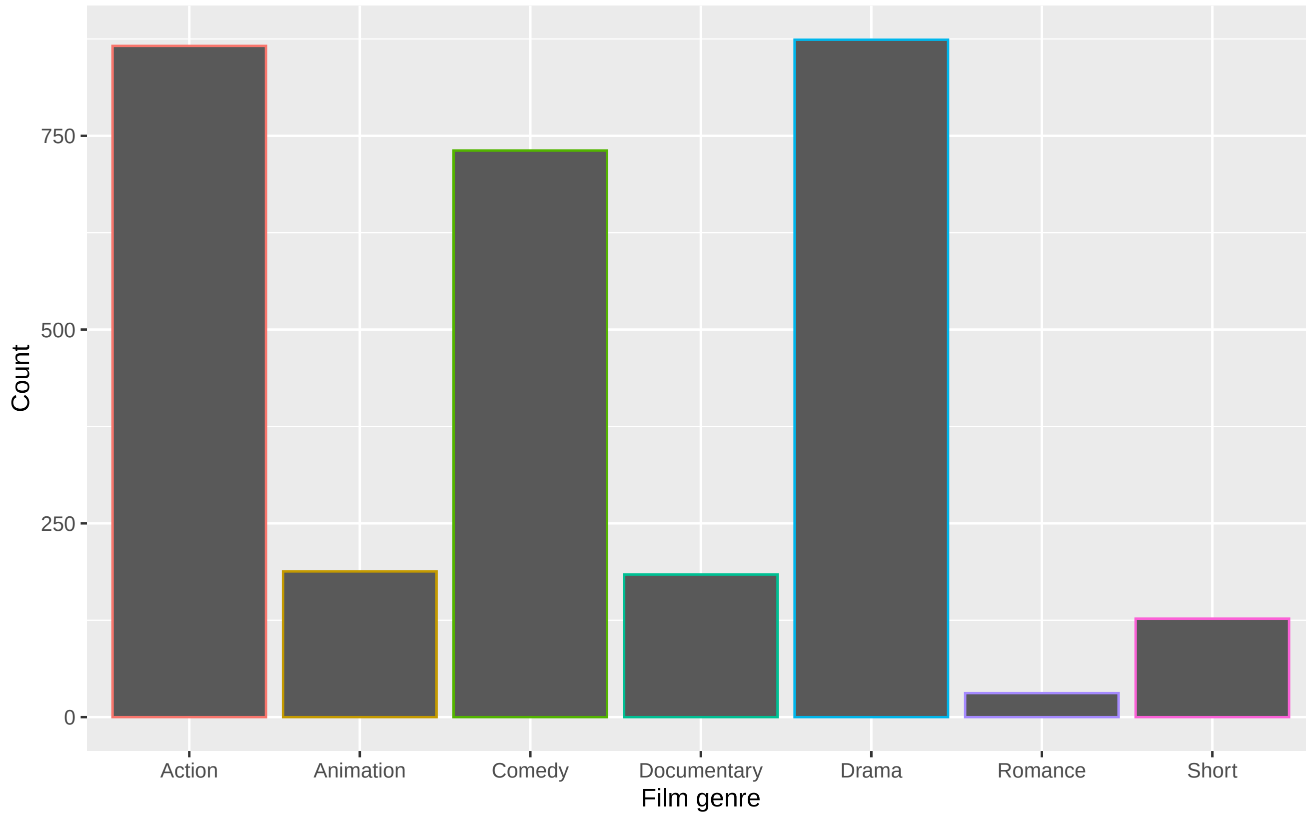


Figure 7: Barplot of count by genre

```
#Proportion of films with rating >7 by genre
film %>% group_by(genre) %>%
  summarise(prop = mean(target)) %>%
  arrange() %>%
  ggplot(aes(x=genre, y=prop, colour=genre)) +
  geom_col() +
  theme(legend.position="none") +
  labs(y="Proportion with rating > 7", x="Film genre")
```

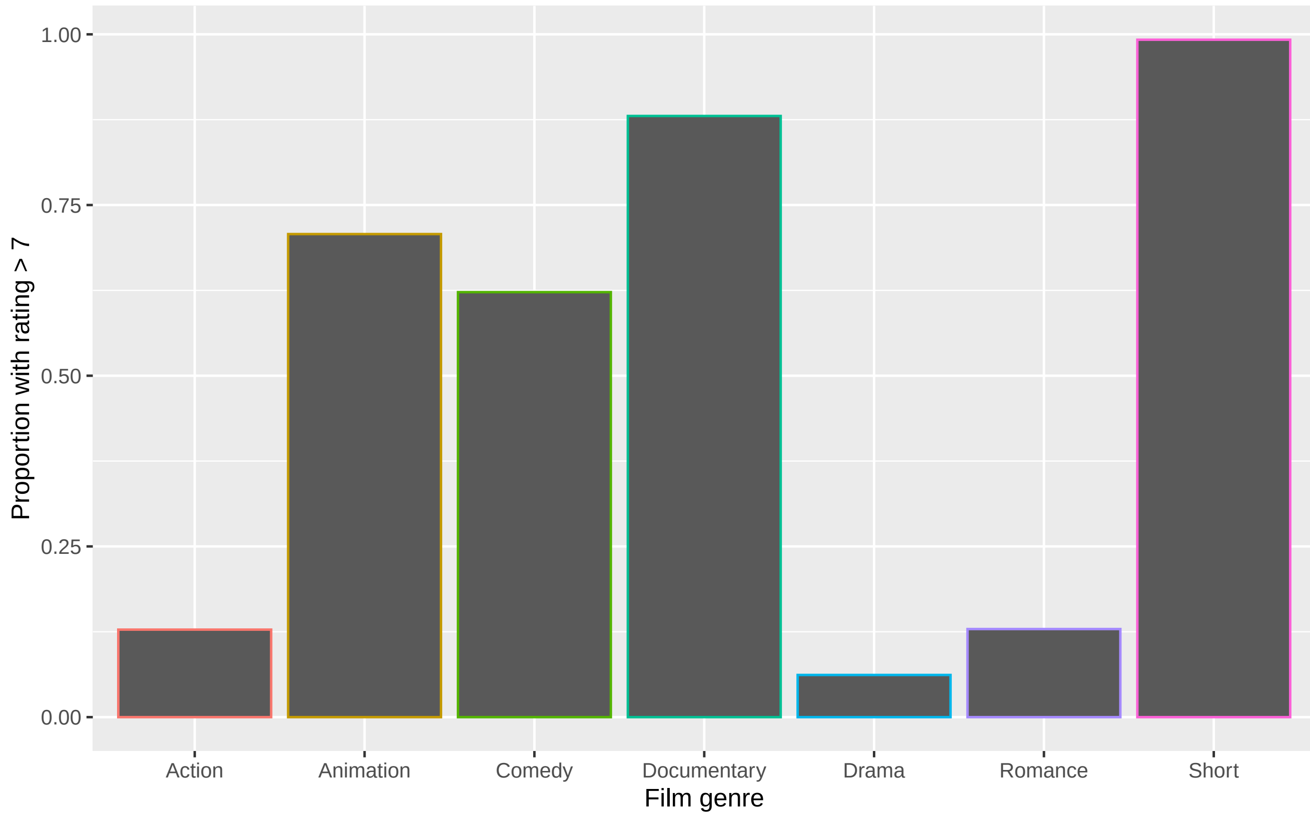


Figure 8: Proportion of films with rating >7 by genre

0.4 Model Fitting

```
#Fit a full model with all possible covariates
model1_1 <- glm(target ~ year + length + budget + votes + genre , data = film,
             family = binomial(link = "logit"))
#The length in 1_1 the NA value in length is replaced by median
model1_1 %>%
  summary()
```

```
Call:
glm(formula = target ~ year + length + budget + votes + genre,
    family = binomial(link = "logit"), data = film)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.325e+01  5.580e+00  -2.375  0.01755 *
year              4.958e-03  2.844e-03   1.743  0.08131 .
length           -6.176e-02  3.674e-03 -16.812  < 2e-16 ***
budget            5.180e-01  2.897e-02  17.882  < 2e-16 ***
votes             4.417e-05  1.526e-05   2.895  0.00379 **
genreAnimation   -5.376e-01  3.331e-01  -1.614  0.10654
genreComedy       3.343e+00  1.759e-01  19.006  < 2e-16 ***
```

```
genreDocumentary  5.311e+00  3.824e-01  13.889  < 2e-16 ***
genreDrama       -1.485e+00  2.281e-01  -6.510 7.53e-11 ***
genreRomance     -7.748e-01  8.552e-01  -0.906  0.36491
genreShort        4.280e+00  1.051e+00   4.072 4.65e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3879.3  on 3000  degrees of freedom
Residual deviance: 1543.9  on 2990  degrees of freedom
AIC: 1565.9

Number of Fisher Scoring iterations: 7
```

```r
#set a dataset without NA
film_without <- read.csv("dataset09.csv") %>%
  mutate(target = ifelse(rating>7, 1, 0)) %>%  #Define target variable
  mutate(Rating = ifelse(rating>7, ">7", "<=7"))  #Define Rating variable help us get better da
film_without<- na.omit(film_without)

#The length in 1_2 the NA value in length is removed
model1_2 <- glm(target ~ year + length + budget + votes + genre , data = film_without,
             family = binomial(link = "logit"))
model1_2 %>%
  summary()
```

```
Call:
glm(formula = target ~ year + length + budget + votes + genre,
    family = binomial(link = "logit"), data = film_without)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.645e+01  5.733e+00  -2.869  0.00412 **
year              6.564e-03  2.921e-03   2.247  0.02463 *
length           -6.208e-02  3.715e-03 -16.711  < 2e-16 ***
budget            5.190e-01  2.976e-02  17.440  < 2e-16 ***
votes             4.423e-05  1.528e-05   2.894  0.00380 **
genreAnimation   -4.419e-01  3.357e-01  -1.316  0.18808
genreComedy       3.370e+00  1.813e-01  18.591  < 2e-16 ***
genreDocumentary  5.320e+00  3.871e-01  13.743  < 2e-16 ***
genreDrama       -1.415e+00  2.309e-01  -6.129 8.84e-10 ***
genreRomance     -6.235e-01  8.967e-01  -0.695  0.48683
genreShort        4.293e+00  1.052e+00   4.081 4.49e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 3722.9  on 2873  degrees of freedom
Residual deviance: 1468.9  on 2863  degrees of freedom
AIC: 1490.9

Number of Fisher Scoring iterations: 7
```

```
summ(model1_1)
```

Observations	3001
Dependent variable	target
Type	Generalized linear model
Family	binomial
Link	logit

$\chi^2(10)$	2335.40
Pseudo-R^2 (Cragg-Uhler)	0.75
Pseudo-R^2 (McFadden)	0.60
AIC	1565.91
BIC	1631.98

	Est.	S.E.	z val.	p
(Intercept)	-13.25	5.58	-2.37	0.02
year	0.00	0.00	1.74	0.08
length	-0.06	0.00	-16.81	0.00
budget	0.52	0.03	17.88	0.00
votes	0.00	0.00	2.90	0.00
genreAnimation	-0.54	0.33	-1.61	0.11
genreComedy	3.34	0.18	19.01	0.00
genreDocumentary	5.31	0.38	13.89	0.00
genreDrama	-1.49	0.23	-6.51	0.00
genreRomance	-0.77	0.86	-0.91	0.36
genreShort	4.28	1.05	4.07	0.00

Standard errors: MLE

```
summ(model1_2)
```

Observations	2874
Dependent variable	target
Type	Generalized linear model
Family	binomial
Link	logit

```
mod1_1coefs <- round(coef(model1_1), 3)
mod1_1coefs
```

$\chi^2(10)$	2254.00
Pseudo-R^2 (Cragg-Uhler)	0.75
Pseudo-R^2 (McFadden)	0.61
AIC	1490.89
BIC	1556.48

	Est.	S.E.	z val.	p
(Intercept)	-16.45	5.73	-2.87	0.00
year	0.01	0.00	2.25	0.02
length	-0.06	0.00	-16.71	0.00
budget	0.52	0.03	17.44	0.00
votes	0.00	0.00	2.89	0.00
genreAnimation	-0.44	0.34	-1.32	0.19
genreComedy	3.37	0.18	18.59	0.00
genreDocumentary	5.32	0.39	13.74	0.00
genreDrama	-1.42	0.23	-6.13	0.00
genreRomance	-0.62	0.90	-0.70	0.49
genreShort	4.29	1.05	4.08	0.00

Standard errors: MLE

```
     (Intercept)             year           length           budget
         -13.251            0.005           -0.062            0.518
           votes   genreAnimation      genreComedy genreDocumentary
           0.000           -0.538            3.343            5.311
      genreDrama     genreRomance       genreShort
          -1.485           -0.775            4.280
```

```
confint(model1_1) %>%
  kable()
```

	2.5 %	97.5 %
(Intercept)	-24.2445156	-2.3580785
year	-0.0005982	0.0105574
length	-0.0691657	-0.0547570
budget	0.4624918	0.5761037
votes	0.0000100	0.0000732
genreAnimation	-1.1955373	0.1108750
genreComedy	3.0053315	3.6953339
genreDocumentary	4.5929006	6.0950856
genreDrama	-1.9437200	-1.0478436
genreRomance	-2.6159021	0.7340490
genreShort	2.6371413	7.1944033

```
mod1_2coefs <- round(coef(model1_2), 3)
mod1_2coefs
```

```
  (Intercept)              year            length            budget 
      -16.449             0.007            -0.062             0.519 
        votes    genreAnimation       genreComedy genreDocumentary 
        0.000            -0.442             3.370             5.320 
   genreDrama      genreRomance        genreShort 
       -1.415            -0.624             4.293 
```

```
confint(model1_2) %>%
  kable()
```

	2.5 %	97.5 %
(Intercept)	-27.7549315	-5.2663362
year	0.0008627	0.0123205
length	-0.0695698	-0.0549994
budget	0.4620653	0.5787970
votes	0.0000099	0.0000733
genreAnimation	-1.1049878	0.2118953
genreComedy	3.0215569	3.7325236
genreDocumentary	4.5923810	6.1129144
genreDrama	-1.8788807	-0.9722200
genreRomance	-2.5305831	0.9642310
genreShort	2.6467563	7.2073797

The two treatments of length have slightly different impacts on the film.

```
#Fit without categorical variable
model2 <- glm(target ~ year +length + budget + votes , data = film, family = binomial(link = "l
model2 %>%
  summary()
```

```
Call:
glm(formula = target ~ year + length + budget + votes, family = binomial(link = "logit"), 
    data = film)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -1.887e+01  4.091e+00  -4.614 3.96e-06 ***
year         9.162e-03  2.082e-03   4.400 1.08e-05 ***
length      -4.549e-02  1.885e-03 -24.128  < 2e-16 ***
budget       3.001e-01  1.886e-02  15.915  < 2e-16 ***
votes        2.375e-05  1.255e-05   1.892   0.0584 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3879.3  on 3000  degrees of freedom
```

```
Residual deviance: 2694.9  on 2996  degrees of freedom
AIC: 2704.9

Number of Fisher Scoring iterations: 5
```

```r
summ(model2)
```

Observations	3001
Dependent variable	target
Type	Generalized linear model
Family	binomial
Link	logit

$\chi^2(4)$	1184.45
Pseudo-R^2 (Cragg-Uhler)	0.45
Pseudo-R^2 (McFadden)	0.31
AIC	2704.86
BIC	2734.89

	Est.	S.E.	z val.	p
(Intercept)	-18.87	4.09	-4.61	0.00
year	0.01	0.00	4.40	0.00
length	-0.05	0.00	-24.13	0.00
budget	0.30	0.02	15.91	0.00
votes	0.00	0.00	1.89	0.06

Standard errors: MLE

```r
mod2coefs <- round(coef(model2), 3)
mod2coefs
```

```
(Intercept)        year      length      budget       votes
    -18.875       0.009      -0.045       0.300       0.000
```

```r
confint(model2) %>%
  kable()
```

	2.5 %	97.5 %
(Intercept)	-26.9369571	-10.8939970
year	0.0051006	0.0132657
length	-0.0492689	-0.0418743
budget	0.2636287	0.3375720
votes	-0.0000039	0.0000472

0.5 Models Comparison

```r
#AIC
aic_values <- c(AIC(model1_1), AIC(model1_2), AIC(model2))
models_aic <- data.frame(Model = c("model1_1", "model1_2", "model2"),
                         AIC = aic_values)

print(models_aic)
```

```
     Model      AIC
1 model1_1 1565.906
2 model1_2 1490.885
3   model2 2704.857
```

```r
pred_prob_model1_1 <- predict(model1_1, newdata = film, type = "response")
y1_1_true <- film$target

pred_prob_model1_2 <- predict(model1_2, newdata = film_without, type = "response")
y1_2_true <- film_without$target

pred_prob_model2 <- predict(model2, newdata = film, type = "response")
y2_true <- film$target
```

```r
#ROC plots
roc_curve <- roc(y1_1_true, pred_prob_model1_1)
plot(roc_curve, main = "ROC Curve for Model1_1", col = "blue")
abline(a = 0, b = 1, lty = 2, col = "red")
auc_value <- round(auc(roc_curve), 2)
legend("bottomright", legend = paste("AUC =", auc_value), col = "blue", lty = 1, bty = "n")
```

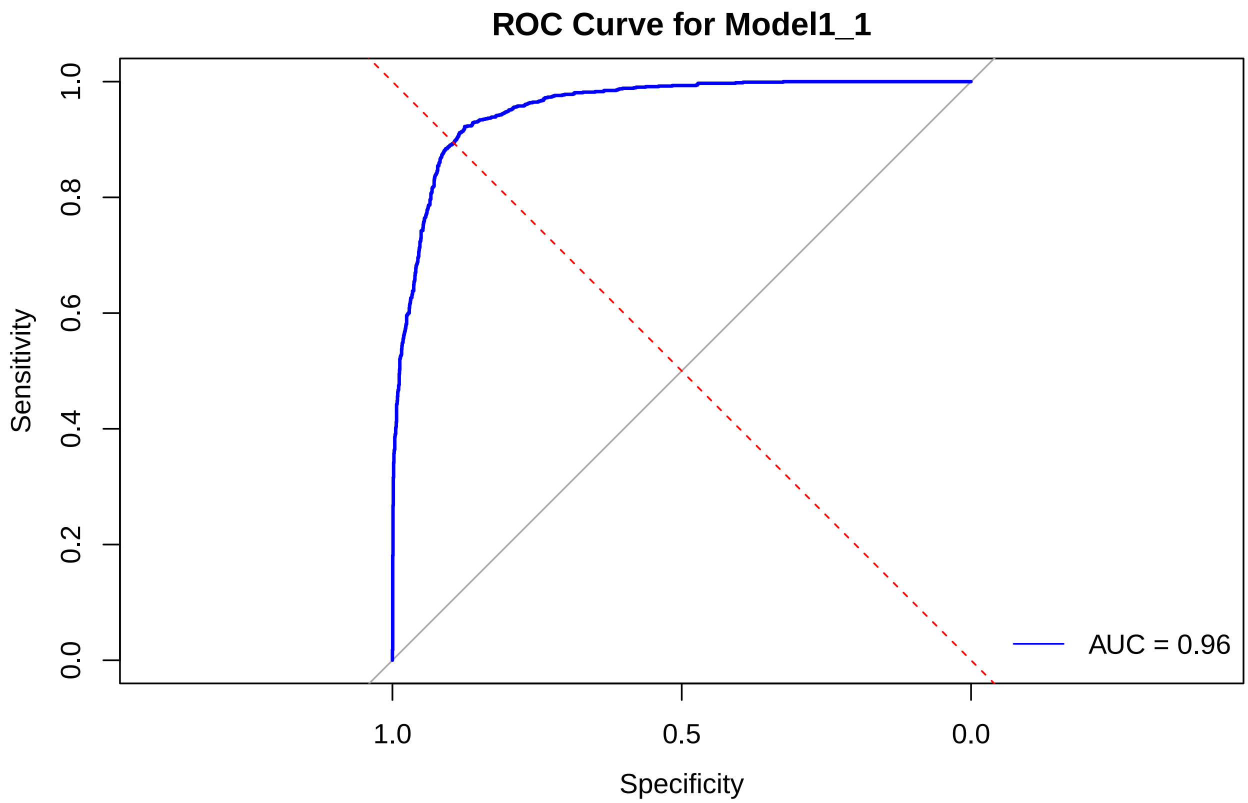


Figure 9: ROC for Model 1.1

```
roc_curve <- roc(y1_2_true, pred_prob_model1_2)
plot(roc_curve, main = "ROC Curve for Model1_2", col = "blue")
abline(a = 0, b = 1, lty = 2, col = "red")
auc_value <- round(auc(roc_curve), 2)
legend("bottomright", legend = paste("AUC =", auc_value), col = "blue", lty = 1, bty = "n")
```

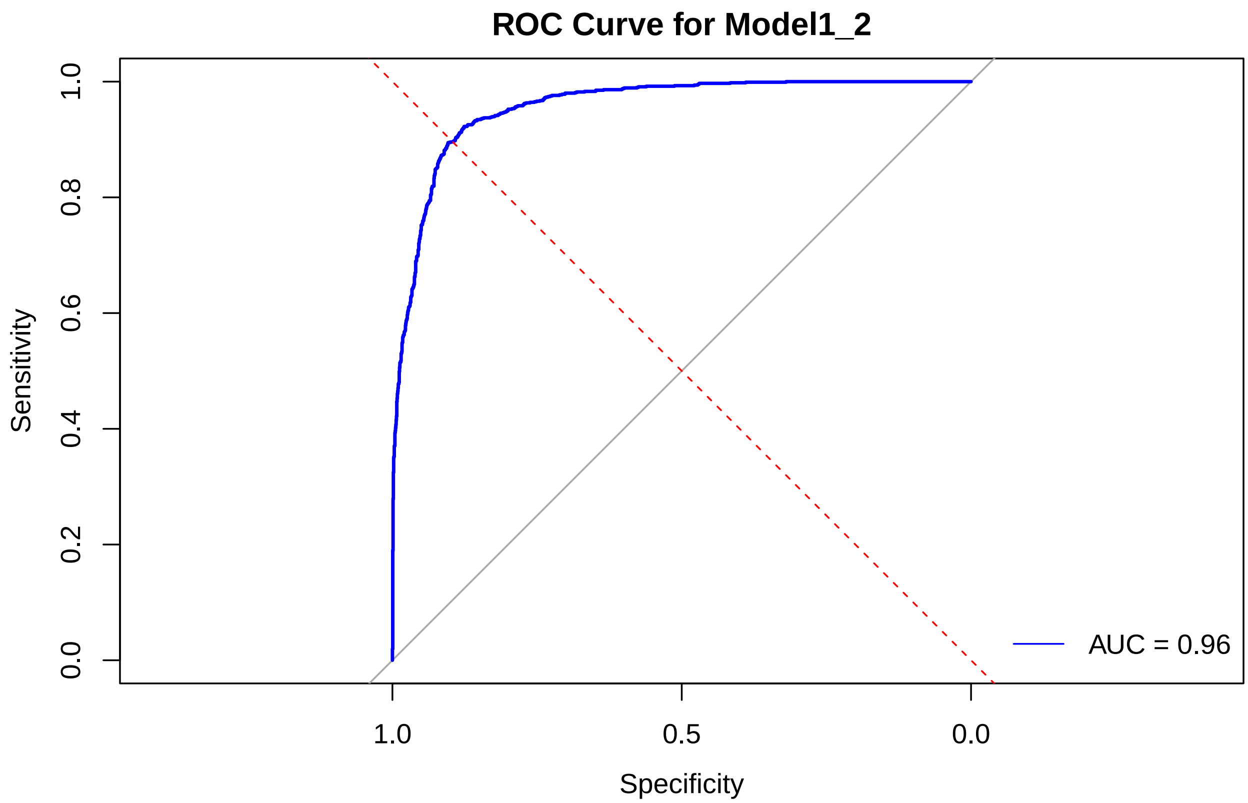
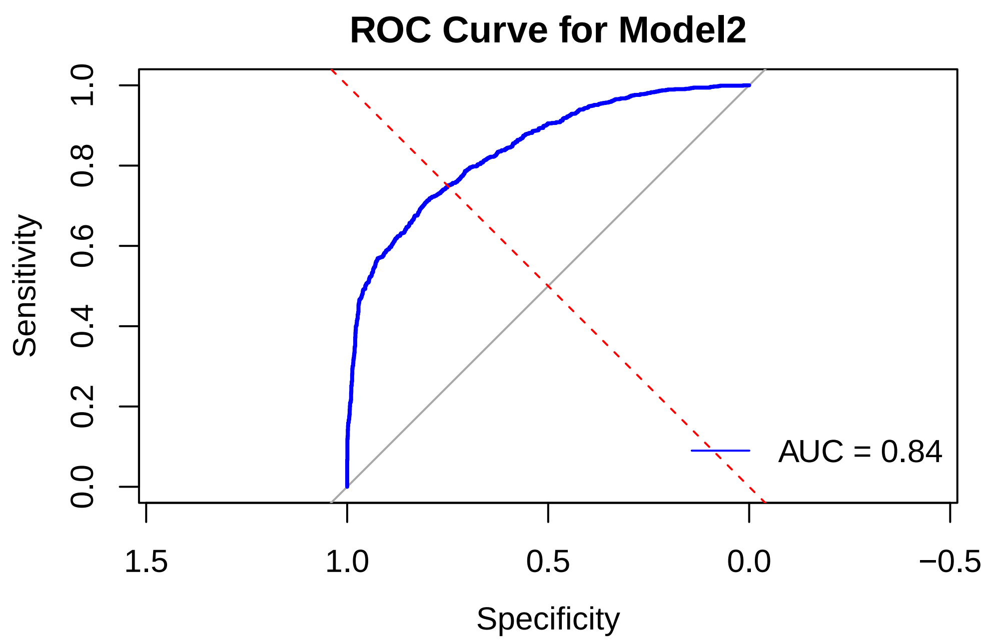


Figure 10: ROC for Model 1.2

```
#| echo: true
#| label: fig_11
#| fig-cap: ROC for Model 2
#| fig-width: 8
#| fig-height: 5
#| fig-align: center


roc_curve <- roc(y2_true, pred_prob_model2)
plot(roc_curve, main = "ROC Curve for Model2", col = "blue")
abline(a = 0, b = 1, lty = 2, col = "red")
auc_value <- round(auc(roc_curve), 2)
legend("bottomright", legend = paste("AUC =", auc_value), col = "blue", lty = 1, bty = "n")
```




```r
# Compute confusion matrix for each model
y_pred_model1_1 <- factor(ifelse(pred_prob_model1_1 > 0.5, 1, 0), levels = c(0, 1))
y1_1_true <- factor(y1_1_true, levels = c(0, 1))
conf_matrix_model1_1 <- confusionMatrix(y_pred_model1_1, y1_1_true)
print(conf_matrix_model1_1)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1815  178
         1  141  867

               Accuracy : 0.8937
                 95% CI : (0.8821, 0.9045)
    No Information Rate : 0.6518
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.7639

 Mcnemar's Test P-Value : 0.04384

            Sensitivity : 0.9279
            Specificity : 0.8297
         Pos Pred Value : 0.9107
         Neg Pred Value : 0.8601
             Prevalence : 0.6518
         Detection Rate : 0.6048
   Detection Prevalence : 0.6641
```

```
      Balanced Accuracy : 0.8788

       'Positive' Class : 0
```

```r
y_pred_model1_2 <- factor(ifelse(pred_prob_model1_2 > 0.5, 1, 0), levels = c(0, 1))
y1_2_true <- factor(y1_2_true, levels = c(0, 1))
conf_matrix_model1_2 <- confusionMatrix(y_pred_model1_2, y1_2_true)
print(conf_matrix_model1_2)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1733  168
         1  134  839

               Accuracy : 0.8949
                 95% CI : (0.8831, 0.9059)
    No Information Rate : 0.6496
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.7674

 Mcnemar's Test P-Value : 0.05757

            Sensitivity : 0.9282
            Specificity : 0.8332
         Pos Pred Value : 0.9116
         Neg Pred Value : 0.8623
             Prevalence : 0.6496
         Detection Rate : 0.6030
   Detection Prevalence : 0.6614
      Balanced Accuracy : 0.8807

       'Positive' Class : 0
```

```r
y_pred_model2 <- factor(ifelse(pred_prob_model2 > 0.5, 1, 0), levels = c(0, 1))
y2_true <- factor(y2_true, levels = c(0, 1))
conf_matrix_model2 <- confusionMatrix(y_pred_model2, y2_true)
print(conf_matrix_model2)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
```

```
         0 1795  448
         1  161  597
                                          
               Accuracy : 0.7971          
                 95% CI : (0.7822, 0.8113)
    No Information Rate : 0.6518          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.5224          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.9177          
            Specificity : 0.5713          
         Pos Pred Value : 0.8003          
         Neg Pred Value : 0.7876          
             Prevalence : 0.6518          
         Detection Rate : 0.5981          
   Detection Prevalence : 0.7474          
      Balanced Accuracy : 0.7445          
                                          
       'Positive' Class : 0               
```

```r
# Compute precision
precision_model1_1 <- conf_matrix_model1_1$byClass["Precision"]
precision_model1_2 <- conf_matrix_model1_2$byClass["Precision"]
precision_model2 <- conf_matrix_model2$byClass["Precision"]

# Compute recall
recall_model1_1 <- conf_matrix_model1_1$byClass["Recall"]
recall_model1_2 <- conf_matrix_model1_2$byClass["Recall"]
recall_model2 <- conf_matrix_model2$byClass["Recall"]

# Compute accuracy
accuracy_model1_1 <- conf_matrix_model1_1$overall["Accuracy"]
accuracy_model1_2 <- conf_matrix_model1_2$overall["Accuracy"]
accuracy_model2 <- conf_matrix_model2$overall["Accuracy"]

# Create a data frame to store the metrics
metrics <- data.frame(Model = c("Model 1_1", "Model 1_2", "Model 2"),
                      Precision = c(precision_model1_1, precision_model1_2, precision_model2),
                      Recall = c(recall_model1_1, recall_model1_2, recall_model2),
                      Accuracy = c(accuracy_model1_1, accuracy_model1_2, accuracy_model2))

# Print the metrics
print(metrics)
```

```
      Model Precision    Recall  Accuracy
```

```
1 Model 1_1 0.9106874 0.9279141 0.8937021
2 Model 1_2 0.9116255 0.9282271 0.8949200
3   Model 2 0.8002675 0.9176892 0.7970676
```

0.6 Conclusion

From the different models that were fit we can see that model 1.2 had the best perfomance when it comes to AIC and also classification metrics (precision, recall and accuracy) so this model will be chosen as the model that describes the influence each covariate has on film rating.

```r
summary(model1_2)
```

```

Call:
glm(formula = target ~ year + length + budget + votes + genre,
    family = binomial(link = "logit"), data = film_without)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.645e+01  5.733e+00  -2.869  0.00412 **
year              6.564e-03  2.921e-03   2.247  0.02463 *
length           -6.208e-02  3.715e-03 -16.711  < 2e-16 ***
budget            5.190e-01  2.976e-02  17.440  < 2e-16 ***
votes             4.423e-05  1.528e-05   2.894  0.00380 **
genreAnimation   -4.419e-01  3.357e-01  -1.316  0.18808
genreComedy       3.370e+00  1.813e-01  18.591  < 2e-16 ***
genreDocumentary  5.320e+00  3.871e-01  13.743  < 2e-16 ***
genreDrama       -1.415e+00  2.309e-01  -6.129 8.84e-10 ***
genreRomance     -6.235e-01  8.967e-01  -0.695  0.48683
genreShort        4.293e+00  1.052e+00   4.081 4.49e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3722.9  on 2873  degrees of freedom
Residual deviance: 1468.9  on 2863  degrees of freedom
AIC: 1490.9

Number of Fisher Scoring iterations: 7
```

$$\log(\frac{p}{1-p}) = \hat{\beta}_0+\hat{\beta}_1 Year+\hat{\beta}_2 Length+\hat{\beta}_3 Budget+\hat{\beta}_4 Votes+\hat{\beta}_5 \mathbb{1}_{Animation}+\hat{\beta}_6 \mathbb{1}_{Comedy}+\hat{\beta}_7 \mathbb{1}_{Documentary}+\hat{\beta}_8 \mathbb{1}_{Drama}+\hat{\beta}_9 \mathbb{1}$$

All covariates were significant except for the animation and short genres, meaning that both of these genres do not have an intercept term which is statistically different to that of the action genre.

```r
coef <- as.data.frame(model1_2$coefficients)
coef <- cbind(Variable = rownames(coef), coef)
rownames(coef) <- 1:nrow(coef)
colnames(coef) <- c('variable', 'estimate')
coef <- coef %>% mutate(estimate = estimate)

coef %>% kable()
```

variable	estimate
(Intercept)	-16.4486717
year	0.0065636
length	-0.0620766
budget	0.5190078
votes	0.0000442
genreAnimation	-0.4419237
genreComedy	3.3697170
genreDocumentary	5.3198865
genreDrama	-1.4150833
genreRomance	-0.6235100
genreShort	4.2925991