Bayesian Statistics Project

Therese Smit

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
library(BAS)
library(knitr)
library(DT)
library(xtable)
library(polycor)
library(lattice)
```

Load data

```
load("movies.Rdata")
```

Part 1: Data

This is an observational study, where the data collected is a random sample of movies collected from the IMDB website. This is a case of simple random sampling.

There was no random assignment in this study, therefore no causality. This, and the random selection of the movies, makes the study generalisable.

Part 2: Data manipulation

Five new variables will now be created, based off the orginal dataset.

```
movies <- movies %>% mutate(feature_film = ifelse(title_type %in% "Feature Film", "Yes", "No"))
movies <- movies %>% mutate(drama = ifelse(genre %in% "Drama", "Yes", "No"))
movies <- movies %>% mutate(mpaa_rating_R = ifelse(mpaa_rating %in% "R", "Yes", "No"))
movies <- movies %>% mutate(oscar_season = ifelse(thtr_rel_month %in% c("10", "11", "12"), "Yes", "No"))
movies <- movies %>% mutate(summer_season = ifelse(thtr_rel_month %in% c("5", "6", "7", "8"), "Yes", "No"))
```

Part 3: Task

Develop a Bayesian regression model to predict audience_score from the following explanatory variables:

```
Explanatory_Variables <- c('feature_film', 'drama', 'runtime', 'mpaa_rating_R', 'thtr_rel_year', 'oscar
df <- data.frame(Explanatory_Variables)

kable(df)
```

Explanatory_Variables

feature_film
drama
runtime
mpaa_rating_R
thtr_rel_year
oscar_season
summer_season
imdb_rating
imdb_num_votes
critics_score
best_pic_nom
best_pic_win
best_actor_win
best_actress_win
best_dir_win
top200_box

Part 4: Exploratory data analysis

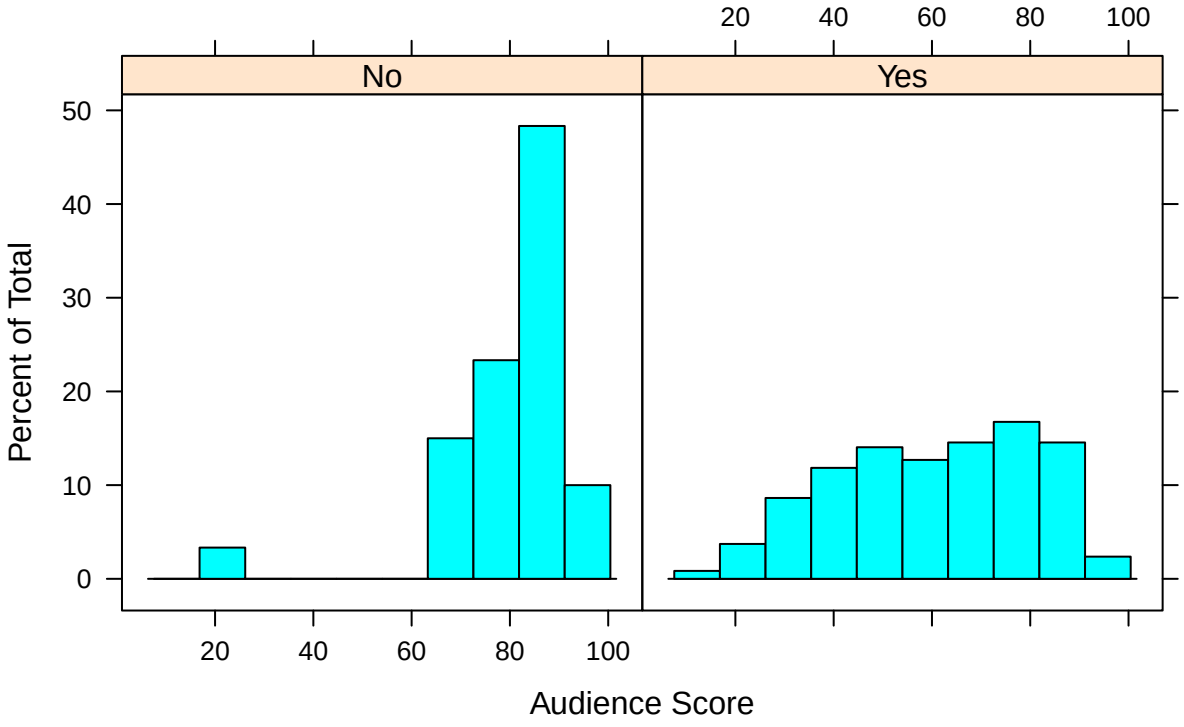
FEATURE FILM

In Fig. 1, one would assume that a high audience score is positively correlated with the title type not being a feature film. The mean calculated for each distribution strengthens this observation.
However, the sample size for feature_yes is 591 and the sample size for feature_no is 60. This is prior knowledge that can be used to rectify the observation above.

```
feature <- movies[ , c("audience_score", "feature_film")]

histogram(~ audience_score | feature_film, data = feature, xlab = "Audience Score",
          main = list(label="Figure 1: Side-by-Side Histograms of the audience score
          based on title type", cex = 1))
```

Figure 1: Side−by−Side Histograms of the audience score based on title type




```
mean(feature$audience_score[feature$feature_film == "Yes"])
```

```
## [1] 60.46531
```

```
mean(feature$audience_score[feature$feature_film == "No"])
```

```
## [1] 81.05
```

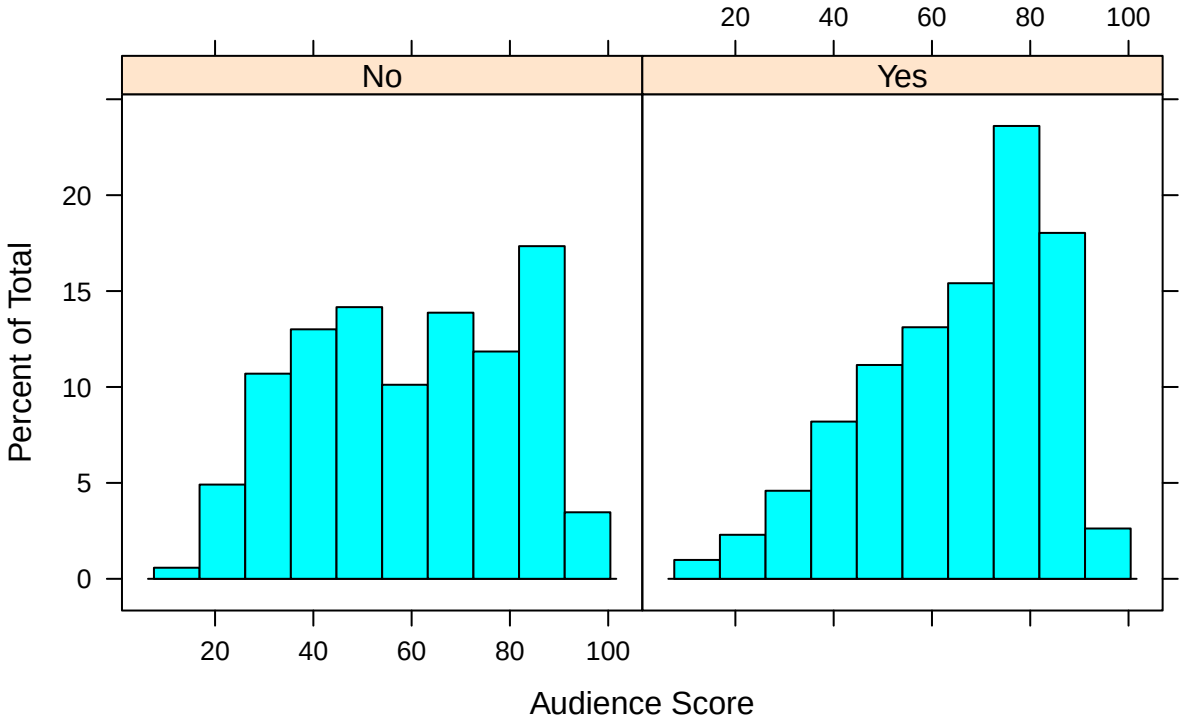
DRAMA

In Fig. 2, the plots of audience score based on whether the film is a drama or not show different distributions. However the means of each distribution are similar, with 'Drama' having a slightly higher mean. This implies that the audience score is more likely to be higher if the film is classified as a drama.

```
gen_dr <- movies[ , c("audience_score", "drama")]

histogram(~ audience_score | drama, data = gen_dr, xlab = "Audience Score",
          main = list(label="Figure 2: Side-by-Side Histograms of the audience score
          based on genre", cex = 1))
```

Figure 2: Side−by−Side Histograms of the audience score based on genre




```
mean(gen_dr$audience_score[gen_dr$drama == "Yes"])
```

```
## [1] 65.34754
```

```
mean(gen_dr$audience_score[gen_dr$drama == "No"])
```

```
## [1] 59.73121
```

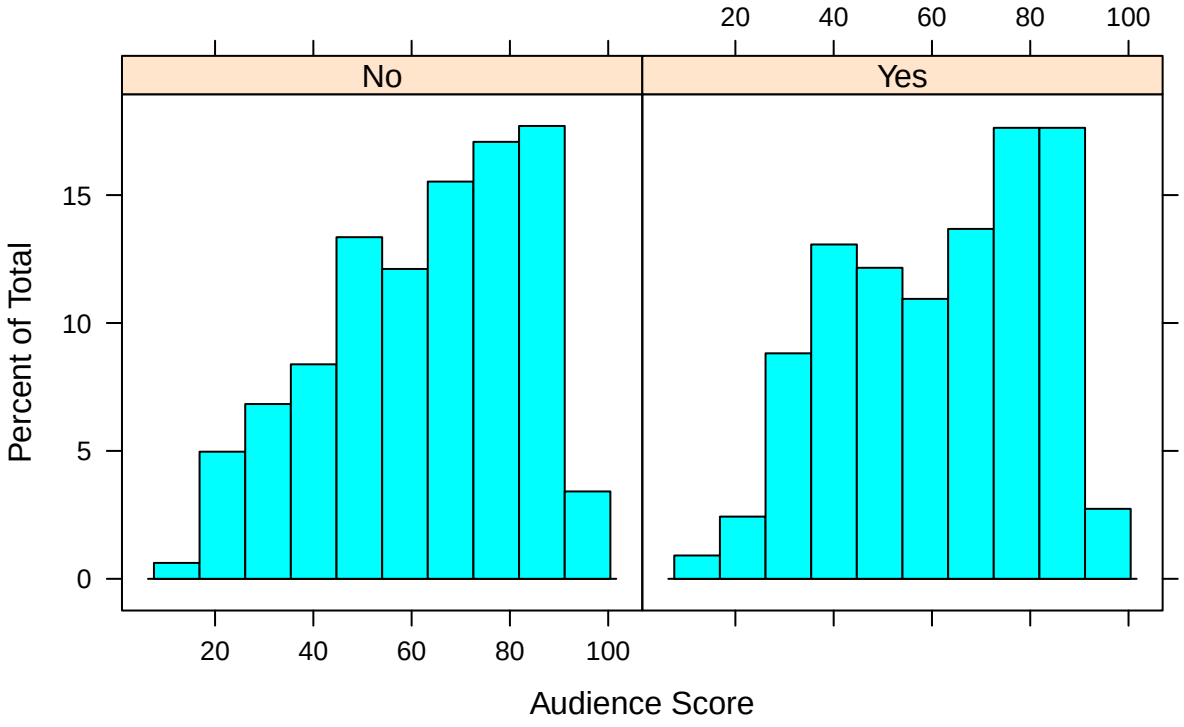
MPAA RATING R

In Fig. 3, the distributions shown are very similar. This implies that the audience score given is not necessarily affected by the MPAA rating of the film. The means of each distribution substantiates this observation.

```
mpaa <- movies[ , c("audience_score", "mpaa_rating_R")]

histogram(~ audience_score | mpaa_rating_R, data = mpaa, xlab = "Audience Score",
          main = list(label="Figure 3: Side-by-Side Histograms of the audience score
          based on MPAA rating", cex = 1))
```

Figure 3: Side−by−Side Histograms of the audience score based on MPAA rating




```
mean(mpaa$audience_score[mpaa$mpaa_rating_R == "Yes"])
```

```
## [1] 62.04255
```

```
mean(mpaa$audience_score[mpaa$mpaa_rating_R == "No"])
```

```
## [1] 62.68944
```

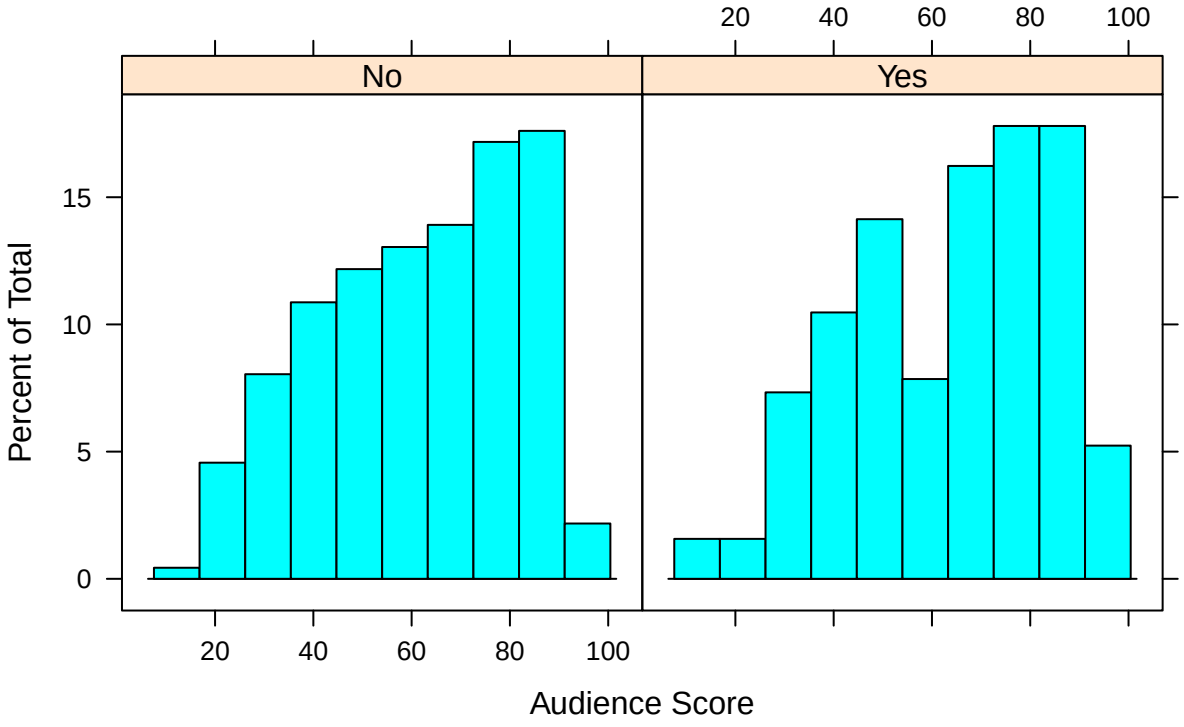
OSCAR SEASON

In Fig. 4, the distributions shown are very similar. This implies that the audience score given is not necessarily affected by whether the film was released in Oscar season. However, the mean of the audience score when the film was released in Oscar season is slightly higher than the other, which could imply that this has some affect on the audience score.

```
oscar <- movies[ , c("audience_score", "oscar_season")]

histogram(~ audience_score | oscar_season, data = oscar, xlab = "Audience Score",
          main = list(label="Figure 4: Side-by-Side Histograms of the audience score
          based on Oscar season", cex = 1))
```

Figure 4: Side−by−Side Histograms of the audience score based on Oscar season




```
mean(oscar$audience_score[oscar$oscar_season == "Yes"])
```

```
## [1] 63.68586
```

```
mean(oscar$audience_score[oscar$oscar_season == "No"])
```

```
## [1] 61.81304
```

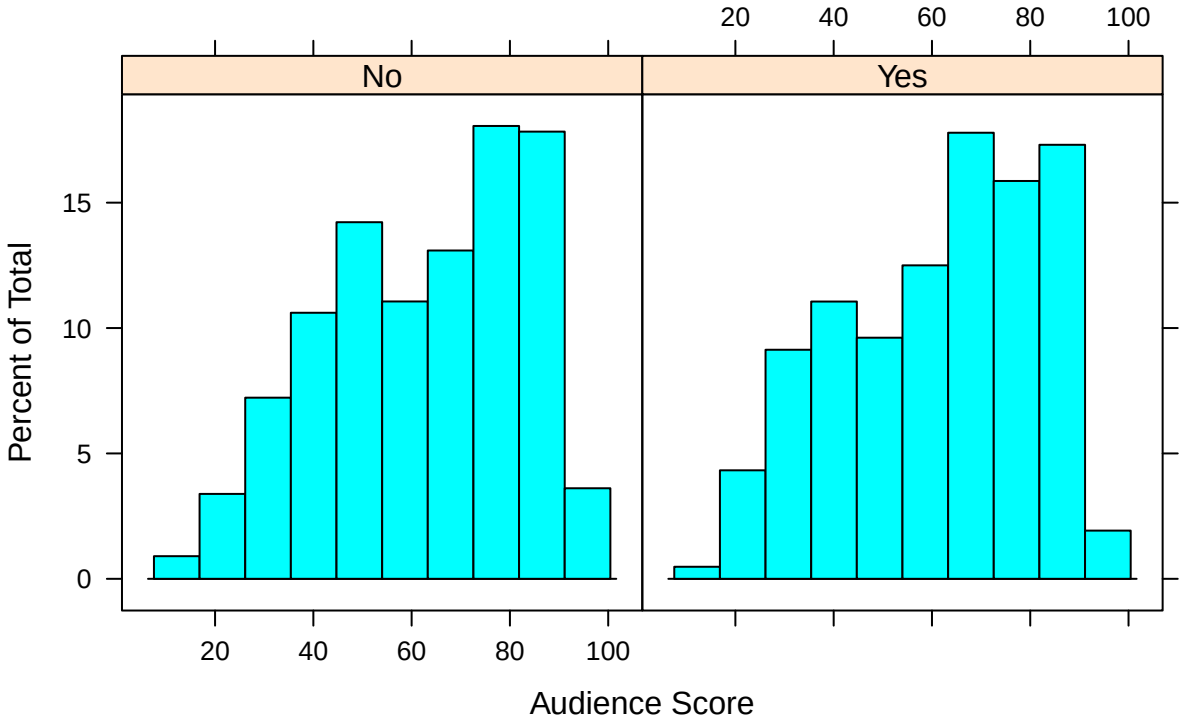
SUMMER SEASON

In Fig. 5, the distributions shown are very similar. This implies that the audience score given is not necessarily affected by whether the film was released in the summer season. The means of each distribution substantiates this observation.

```
summer <- movies[ , c("audience_score", "summer_season")]

histogram(~ audience_score | summer_season, data = summer, xlab = "Audience Score",
          main = list(label="Figure 5: Side-by-Side Histograms of the audience score
          based on Summer season", cex = 1))
```

Figure 5: Side−by−Side Histograms of the audience score based on Summer season




```
mean(summer$audience_score[summer$summer_season == "Yes"])
```

```
## [1] 61.80769
```

```
mean(summer$audience_score[summer$summer_season == "No"])
```

```
## [1] 62.62302
```

Part 5: Modeling

The audience score can be explained by many predictors. The initial multiple linear regression model includes all the explanatory variables stated above.

```
audience_score_df <- movies[, c('audience_score', 'feature_film', 'drama', 'runtime', 'mpaa_rating_R',
audience_score_full = lm(audience_score ~ . -audience_score, data = audience_score_df)
audience_score_full
```

```
##
## Call:
## lm(formula = audience_score ~ . - audience_score, data = audience_score_df)
##
## Coefficients:
##         (Intercept)      feature_filmYes             dramaYes
##           1.244e+02           -2.248e+00            1.292e+00
##             runtime     mpaa_rating_RYes        thtr_rel_year
##          -5.614e-02           -1.444e+00           -7.657e-02
```

```
##     oscar_seasonYes     summer_seasonYes          imdb_rating
##          -5.333e-01            9.106e-01            1.472e+01
##      imdb_num_votes        critics_score      best_pic_nomyes
##           7.234e-06            5.748e-02            5.321e+00
##     best_pic_winyes    best_actor_winyes  best_actress_winyes
##          -3.212e+00           -1.544e+00           -2.198e+00
##     best_dir_winyes        top200_boxyes
##          -1.231e+00            8.478e-01
```

The summary of the full linear model above gives the coefficients of the independent variables. Bayesian Model Averaging (BMA) will be implemented to perform model selection.

```
bma_audience_score = bas.lm(audience_score ~ . -audience_score, data = audience_score_df, prior = "BIC"
```

```
## Warning in bas.lm(audience_score ~ . - audience_score, data =
## audience_score_df, : dropping 1 rows due to missing data
```

```
bma_audience_score
```

```
##
## Call:
## bas.lm(formula = audience_score ~ . - audience_score, data = audience_score_df,     prior = "BIC", m
##
##
##  Marginal Posterior Inclusion Probabilities:
##           Intercept      feature_filmYes             dramaYes
##             1.00000              0.06537              0.04320
##             runtime     mpaa_rating_RYes        thtr_rel_year
##             0.46971              0.19984              0.09069
##     oscar_seasonYes     summer_seasonYes          imdb_rating
##             0.07506              0.08042              1.00000
##      imdb_num_votes        critics_score      best_pic_nomyes
##             0.05774              0.88855              0.13119
##     best_pic_winyes    best_actor_winyes  best_actress_winyes
##             0.03985              0.14435              0.14128
##     best_dir_winyes        top200_boxyes
##             0.06694              0.04762
```

```
summary(bma_audience_score)
```

```
##                     P(B != 0 | Y)    model 1       model 2       model 3
## Intercept              1.00000000     1.0000     1.0000000     1.0000000
## feature_filmYes        0.06536947     0.0000     0.0000000     0.0000000
## dramaYes               0.04319833     0.0000     0.0000000     0.0000000
## runtime                0.46971477     1.0000     0.0000000     0.0000000
## mpaa_rating_RYes       0.19984016     0.0000     0.0000000     0.0000000
## thtr_rel_year          0.09068970     0.0000     0.0000000     0.0000000
## oscar_seasonYes        0.07505684     0.0000     0.0000000     0.0000000
## summer_seasonYes       0.08042023     0.0000     0.0000000     0.0000000
## imdb_rating            1.00000000     1.0000     1.0000000     1.0000000
## imdb_num_votes         0.05773502     0.0000     0.0000000     0.0000000
## critics_score          0.88855056     1.0000     1.0000000     1.0000000
## best_pic_nomyes        0.13119140     0.0000     0.0000000     0.0000000
## best_pic_winyes        0.03984766     0.0000     0.0000000     0.0000000
## best_actor_winyes      0.14434896     0.0000     0.0000000     1.0000000
## best_actress_winyes    0.14128087     0.0000     0.0000000     0.0000000
## best_dir_winyes        0.06693898     0.0000     0.0000000     0.0000000
```

```
## top200_boxyes          0.04762234     0.0000     0.0000000     0.0000000
## BF                             NA     1.0000     0.9968489     0.2543185
## PostProbs                      NA     0.1297     0.1293000     0.0330000
## R2                             NA     0.7549     0.7525000     0.7539000
## dim                            NA     4.0000     3.0000000     4.0000000
## logmarg                        NA -3615.2791 -3615.2822108 -3616.6482224
##                           model 4       model 5
## Intercept               1.0000000     1.0000000
## feature_filmYes         0.0000000     0.0000000
## dramaYes                0.0000000     0.0000000
## runtime                 0.0000000     1.0000000
## mpaa_rating_RYes        1.0000000     1.0000000
## thtr_rel_year           0.0000000     0.0000000
## oscar_seasonYes         0.0000000     0.0000000
## summer_seasonYes        0.0000000     0.0000000
## imdb_rating             1.0000000     1.0000000
## imdb_num_votes          0.0000000     0.0000000
## critics_score           1.0000000     1.0000000
## best_pic_nomyes         0.0000000     0.0000000
## best_pic_winyes         0.0000000     0.0000000
## best_actor_winyes       0.0000000     0.0000000
## best_actress_winyes     0.0000000     0.0000000
## best_dir_winyes         0.0000000     0.0000000
## top200_boxyes           0.0000000     0.0000000
## BF                      0.2521327     0.2391994
## PostProbs               0.0327000     0.0310000
## R2                      0.7539000     0.7563000
## dim                     4.0000000     5.0000000
## logmarg             -3616.6568544 -3616.7095127
```

The most likely model shown in the results table above has posterior probability of 0.1297. This is Model 1, which includes an intercept, the runtime, the IMDB rating and the critics score.
This is then the final model.

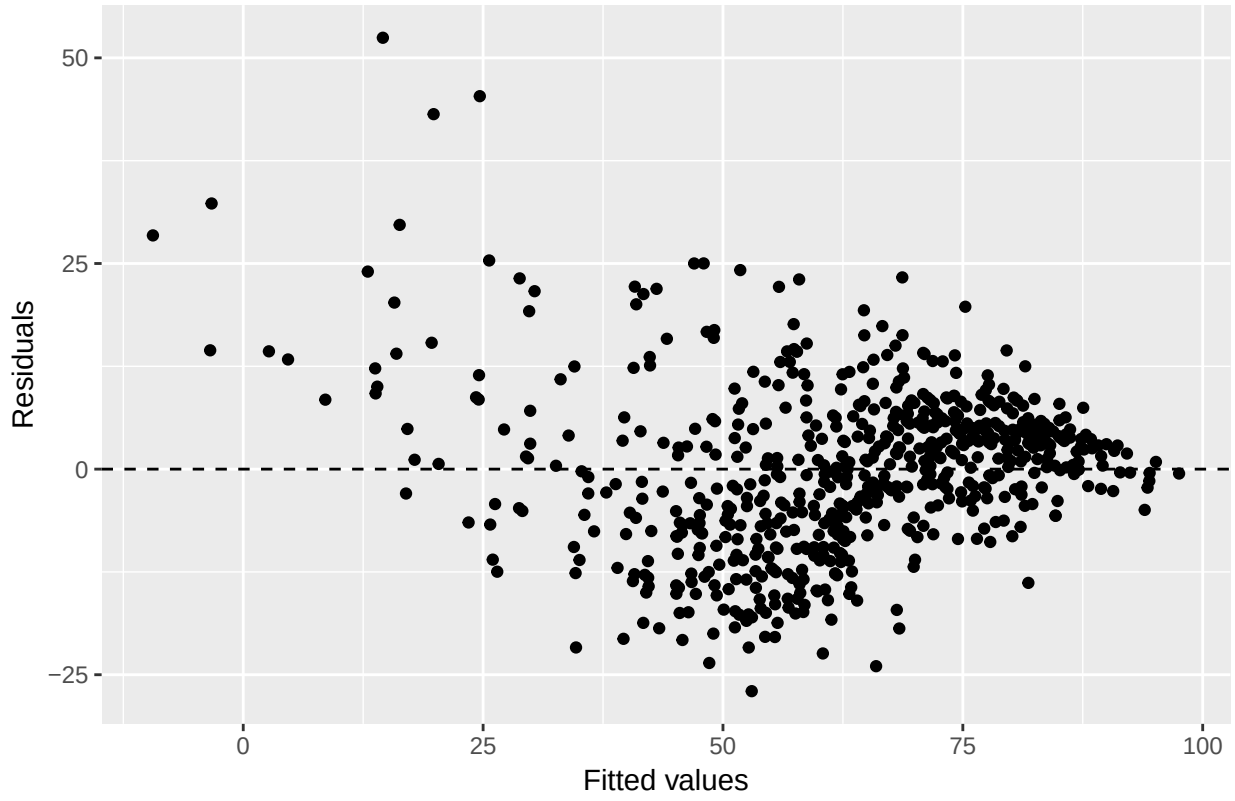
```
final_model = lm(audience_score ~ runtime + imdb_rating + critics_score, data = audience_score_df)
final_model
```

```
##
## Call:
## lm(formula = audience_score ~ runtime + imdb_rating + critics_score,
##     data = audience_score_df)
##
## Coefficients:
##   (Intercept)        runtime    imdb_rating  critics_score
##     -33.28321       -0.05362       14.98076        0.07036
```

Model diagnostics will now be performed on the final model.
In Fig. 6 below it can be seen that there is a random scatter around 0. This confirms a linear relationship between the x and y variables.

```
ggplot(data = final_model, aes(x = .fitted, y = .resid)) + geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") + xlab("Fitted values") +
  ylab("Residuals") + ggtitle("Figure 6: Residual plot of the final model")
```

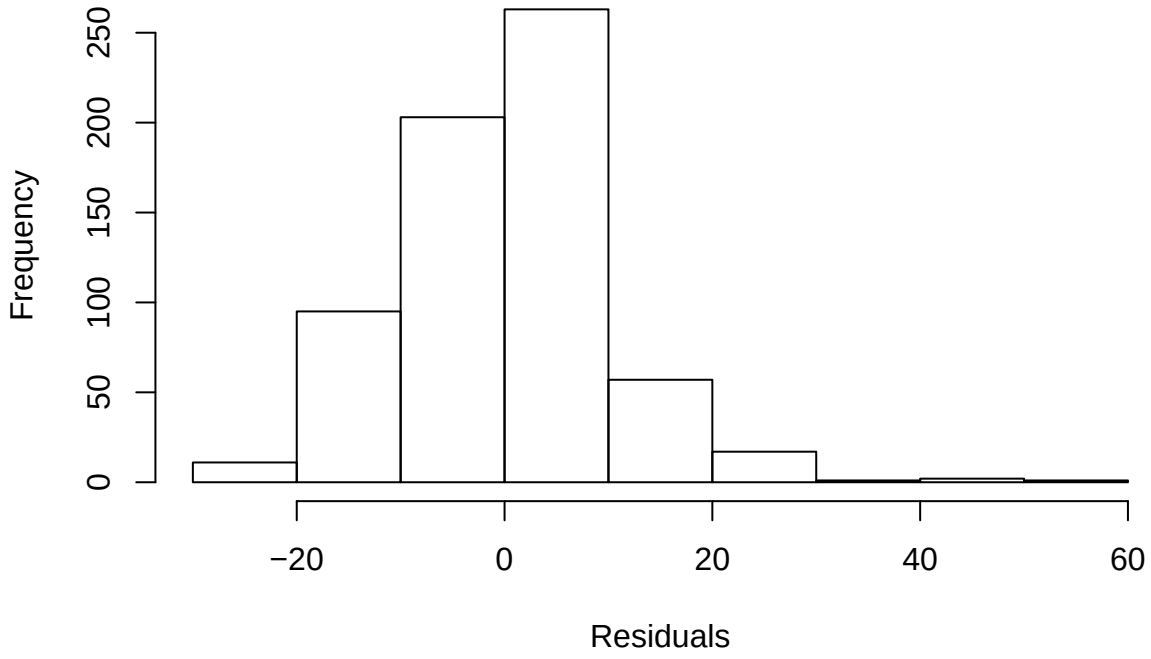
Figure 6: Residual plot of the final model



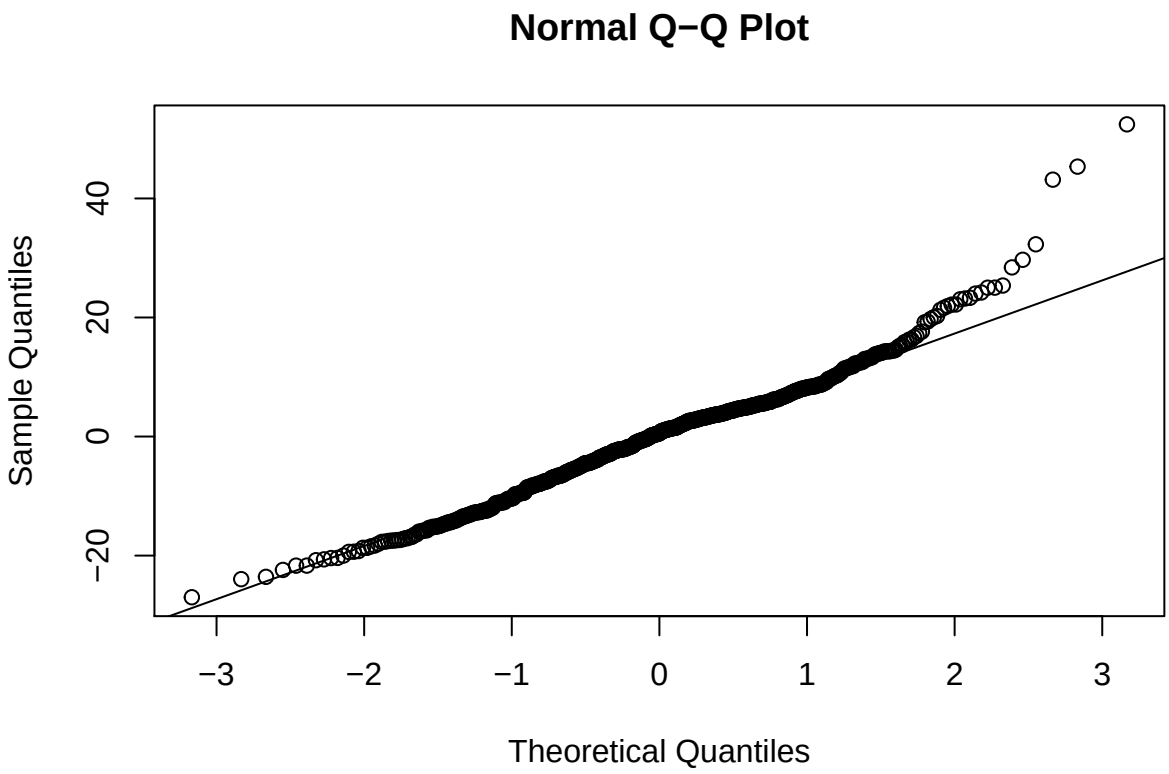
In Fig. 7 below, and the normal Q-Q plot, it can be seen that the residuals are centered around 0. Therefore the residuals are nearly normal with mean 0.

```
hist(final_model$residuals, main = "Figure 7: Histogram on the final models residuals", xlab = "Residua
```

Figure 7: Histogram on the final models residuals



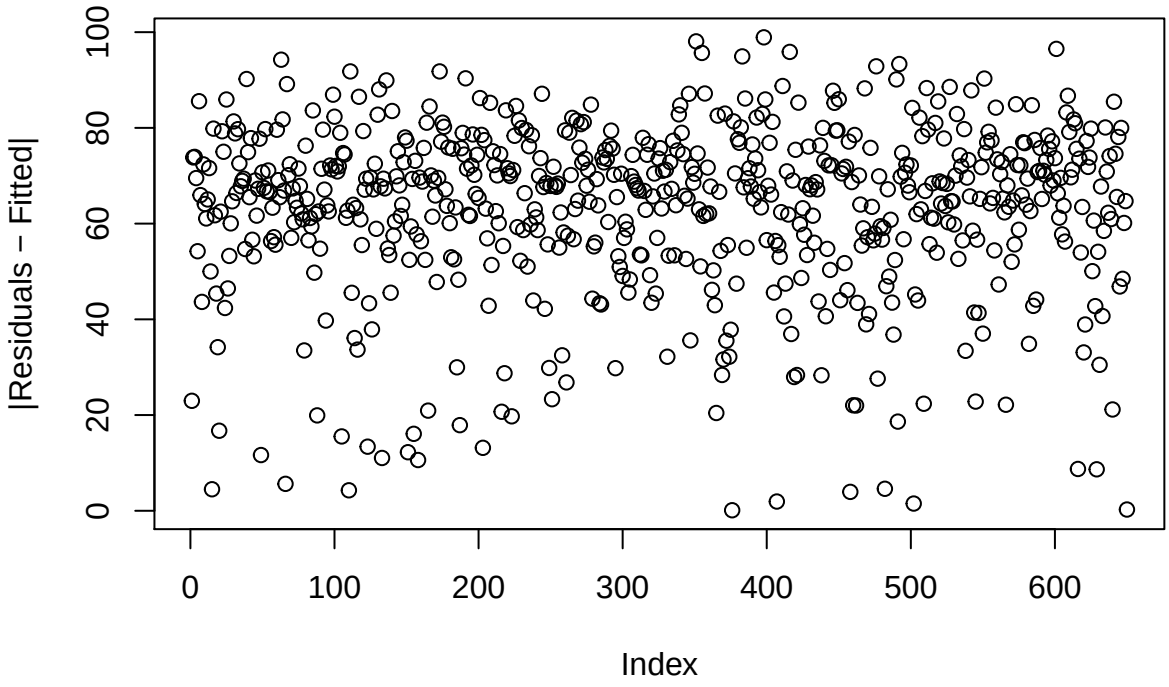

```
qqnorm(final_model$residuals)
qqline(final_model$residuals)
```



In the residual plot below, where the residuals are plotted against the predicted values, it can be seen that the residuals are not randomly scattered in a band with a constant width around 0. Therefore there is not a constant variability of the residuals.

```
plot(abs(final_model$residuals - final_model$fitted), main = "Figure 8: Residual plot of residuals vs pr
```

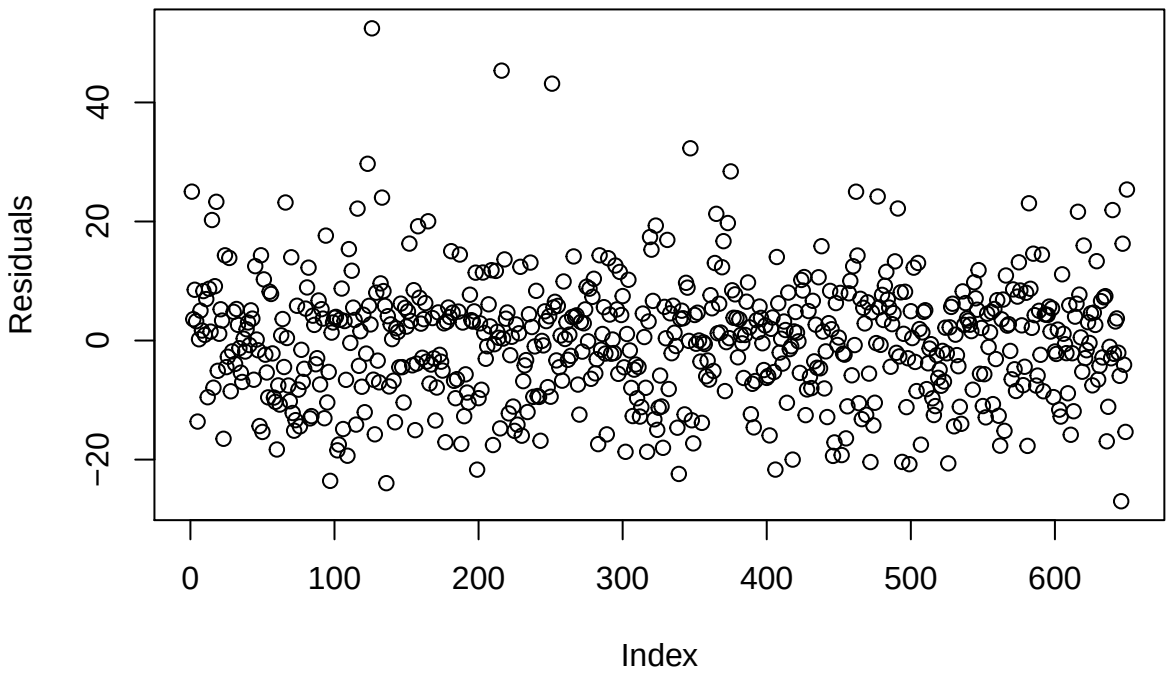
Figure 8: Residual plot of residuals vs predicted values



As mentioned previously, in the residual plot below it can be seen that there is random scatter around 0. Therefore the residuals are independent.

```
plot(final_model$residuals, main = "Figure 9: Residual plot of the residuals", ylab = "Residuals")
```

Figure 9: Residual plot of the residuals



Part 6: Prediction

The multiple linear regression model above will now be used to predict a movie from 2016. The chosen movie is La La Land, which has an audience score of 81%. This movie is not in the dataset provided for this project. The parameters we will be including in the prediction is the runtime (128 minutes), the IMDB rating (8.2/10) and the critics score (92%). This information was found on the IMDB website and the Rotten Tomatoes website.

```
new_df <- data.frame(title = "La La Land", runtime = 128, imdb_rating = 8.2, critics_score = 92)
predict(final_model, newdata = new_df, interval = 'prediction')
```

```
##        fit      lwr      upr
## 1 89.16913 69.39172 108.9465
```

The prediction given for the audience score of La La Land is 89%, whereas the real score is 81%.
The prediction interval calculated is (69.39, 108.95), which is impossible since the highest audience score that can be given is 100%. Therefore, the realistic prediction interval calculated is (69.39, 100). This tells us that 95% of movies with a runtime of 128 minutes, an IMDB rating of 8.2/10 and a critics score of 92% will have an audience score somewhere between 69.39% and 100%.
Since 81% is within that interval, the multiple linear regression model created is a good prediction tool for movies.

Part 7: Conclusion

Out of the explanatory variables stated above, it can be said that the runtime, the IMDB rating and the critics score had a stronger association with a movies popularity than the others did, based on the findings above.
The prediction done on the multiple linear regression model created accurately placed the real audience score in the prediction interval, however the direct prediction on 81% based on the model was not accurate enough. Improvements can be made on the model, such as implementing BMA instead just using the posterior probabilities to select the best model.