Assignment on Bayesian Inference

Paper Name: Theory of Estimation

Paper Code: STAT-421

Application Topic: Movies' Rating Prediction Using Frequentist and Bayesian Linear Regression Approach

Department of Statistics

Pondicherry University

Submitted To,

Dr. SUDESH PUNDIR

Associate Professor

Submitted By,

ARNAB MANNA

(Reg. No. 21375009)

AISHWARYA VELAYUDHAN M

(Reg. No. 21375004)

HOMAGNI ROY

(Reg. No. 21375026)

JUNE 23, 2022

CONTENTS **Page No.**

DATA SOURCE:

We take the dataset of movies rating from the Linear regression modeling course of coursera offered by Duke University, North Carolina. The link of the dataset is given below:

https://www.coursera.org/learn/linear-regression-model/supplement/UQWxR/project-instructions-data-files-and-checklist

DATA DESCRIPTION:

The data set is comprised of 651 randomly sampled movies from imdb and Rotten Tomatoes website produced and released before 2016. The description of the columns of this dataset is given below:

1. **title**: Title of movie
2. **title_type:** Type of movie (Documentary, Feature Film, TV Movie)
3. **genre:** Genre of movie (Action & Adventure, Comedy, Documentary, Drama, Horror, Mystery & Suspense, Other)
4. **runtime:** Runtime of movie (in minutes)
5. **mpaa_rating:** MPAA rating of the movie (G, PG, PG-13, R, Unrated)
6. **studio:** Studio that produced the movie
7. **thtr_rel_year:** Year the movie is released in theatres
8. **thtr_rel_month:** Month the movie is released in theatres
9. **thtr_rel_day:** Day of the month the movie is released in theatres
10. **dvd_rel_year:** Year the movie is released on DVD
11. **dvd_rel_month:** Month the movie is released on DVD
12. **dvd_rel_day:** Day of the month the movie is released on DVD
13. **imdb_rating:** Rating on IMDB
14. **imdb_num_votes:** Number of votes on IMDB
15. **critics_rating:** Categorical variable for critics rating on Rotten Tomatoes (Certified Fresh, Fresh, Rotten)
16. **critics_score:** Critics score on Rotten Tomatoes
17. **audience_rating:** Categorical variable for audience rating on Rotten Tomatoes (Spilled, Upright)
18. **audience_score:** Audience score on Rotten Tomatoes
19. **best_pic_nom:** Whether or not the movie was nominated for a best picture Oscar (no, yes)
20. **best_pic_win:** Whether or not the movie won a best picture Oscar (no, yes)

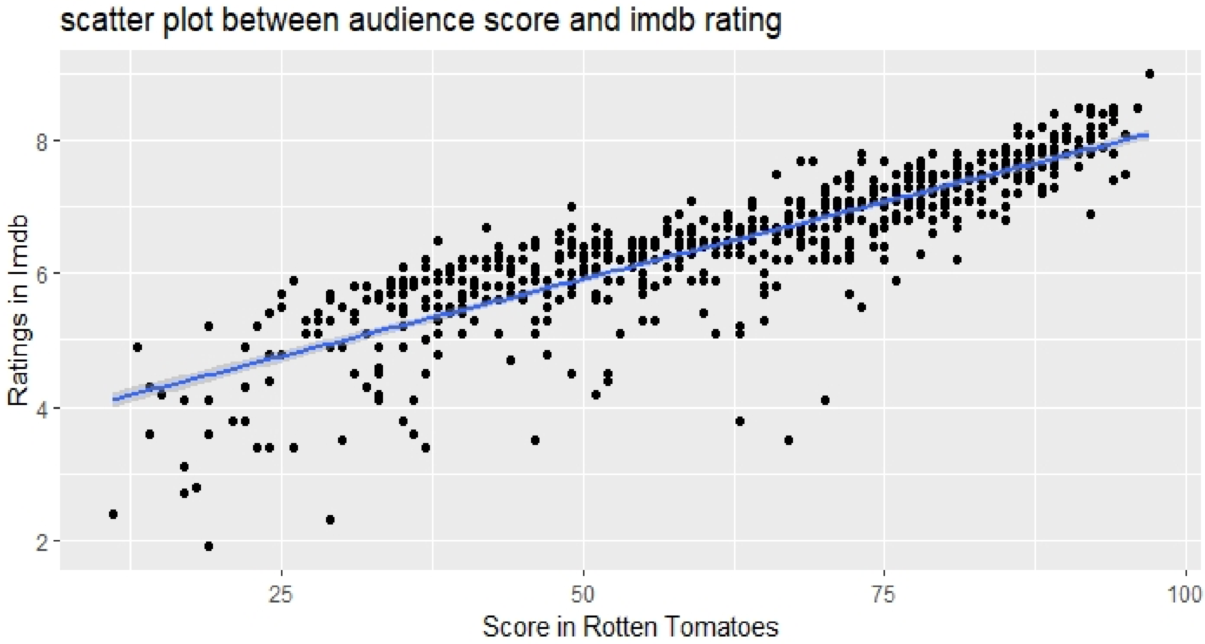
21. **best_actor_win:** Whether or not one of the main actors in the movie ever won an Oscar (no, yes) – note that this is not necessarily whether the actor won an Oscar for their role in the given movie
22. **best_actress win:** Whether or not one of the main actresses in the movie ever won an Oscar (no, yes) – not that this is not necessarily whether the actresses won an Oscar for their role in the given movie
23. **best_dir_win:** Whether or not the director of the movie ever won an Oscar (no, yes) – not that this is not necessarily whether the director won an Oscar for the given movie
24. **top200_box:** Whether or not the movie is in the Top 200 Box Office list on BoxOfficeMojo (no, yes)
25. **director:** Director of the movie
26. **actor1:** First main actor/actress in the abridged cast of the movie
27. **actor2:** Second main actor/actress in the abridged cast of the movie
28. **actor3:** Third main actor/actress in the abridged cast of the movie
29. **actor4:** Fourth main actor/actress in the abridged cast of the movie
30. **actor5:** Fifth main actor/actress in the abridged cast of the movie
31. **imdb_url:** Link to IMDB page for the movie
32. **rt_url:** Link to Rotten Tomatoes page for the movie

OBJECTIVE:

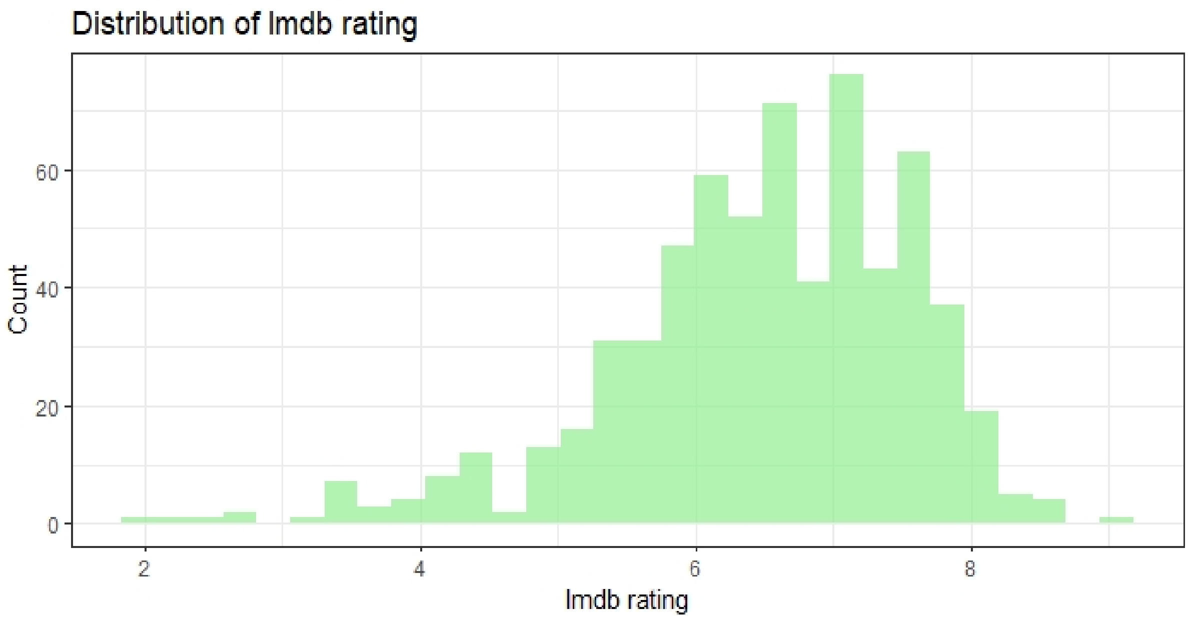
Our main objective of this project is to predict the imdb rating of the movies using multiple linear regression and Bayesian linear regression approach and compare both of these approaches.

EXPLORATORY DATA ANALYSIS:

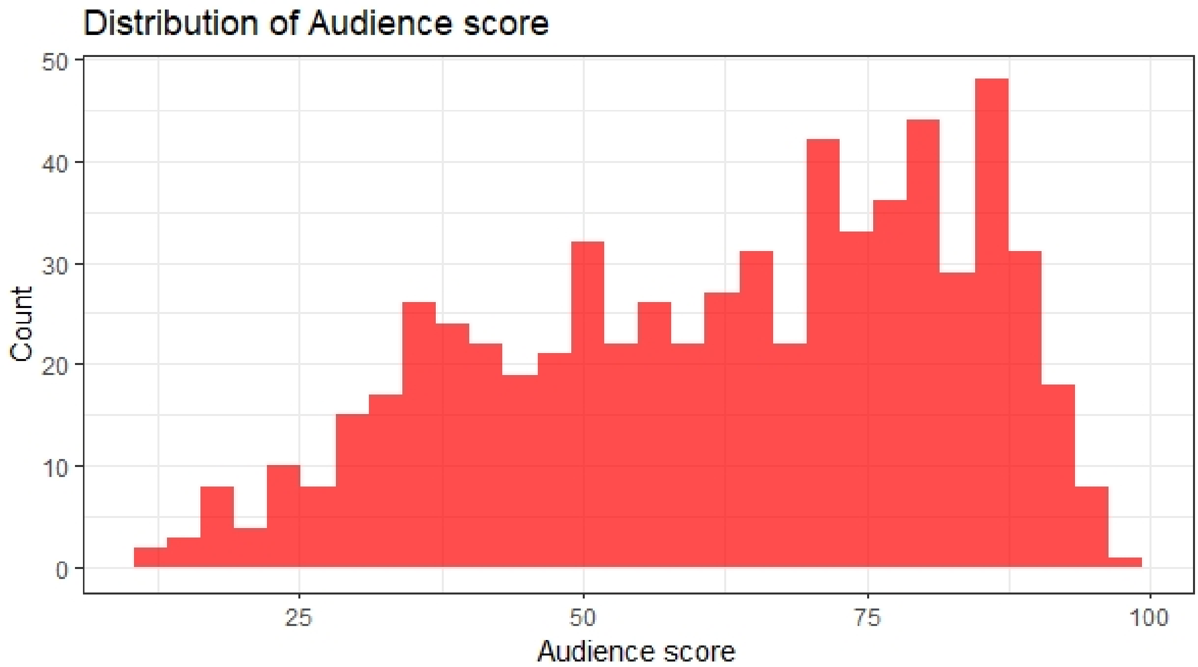
Our dataset is taken from two different source imdb and Rotten Tomatoes website. To know the dependent variable from this dataset first of all we make a scatter plot between imdb_rating and audience_score.



From this scatter plot we can see that there is a linear relationship between imdb_rating and audience_score with high correlation of 0.864. So, here we have to decide which variable we should take. To make a decision we make a histogram of these two variables.



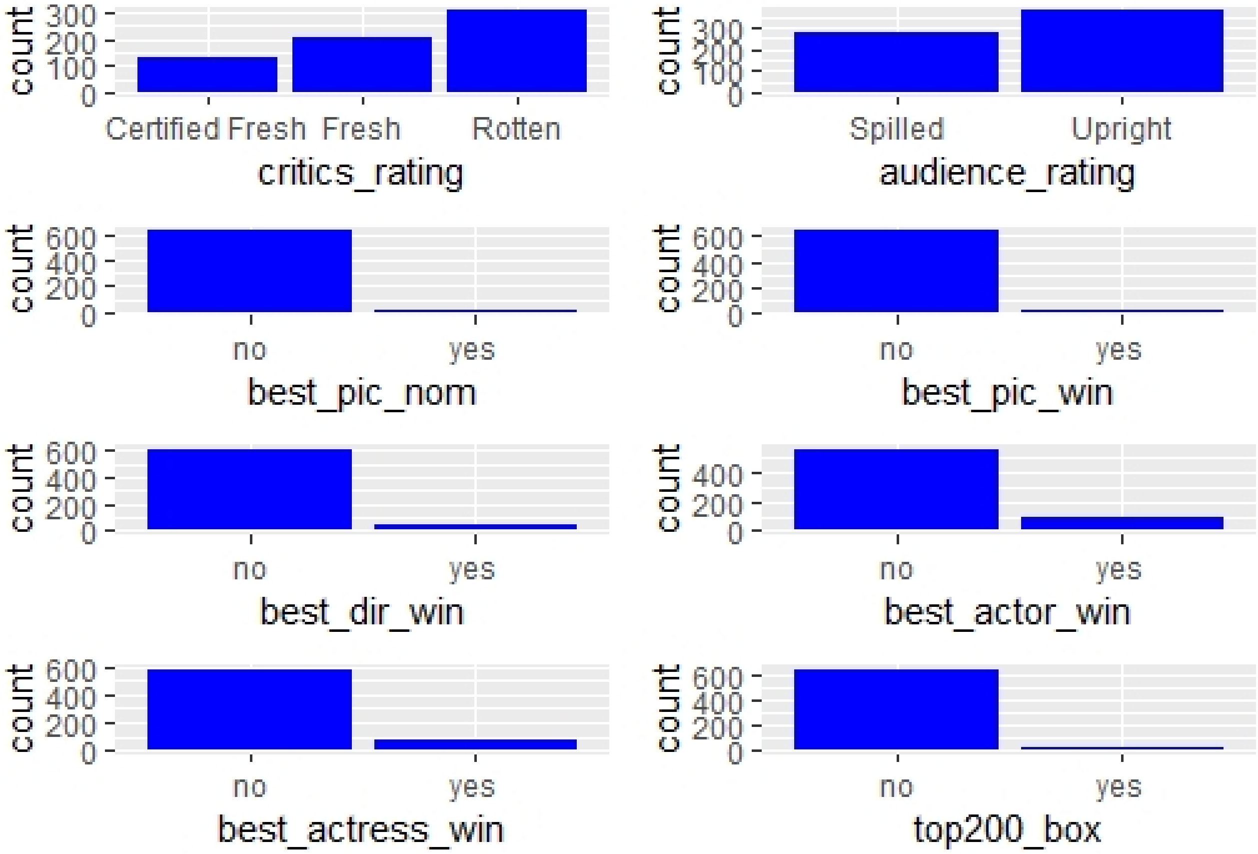
```
> summary(imdb_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.900   5.900   6.600   6.493   7.300   9.000
```



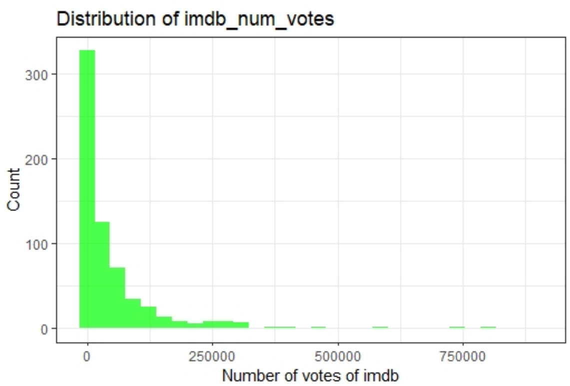

```
summary(audience_score)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  11.00   46.00   65.00   62.36   80.00   97.00
```

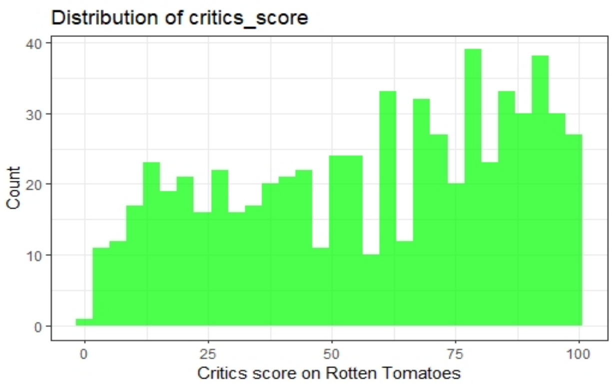
From this histogram we can tell imdb_rating is normally distributed (slightly negatively skwed) with mean 6.493 and audience_score shows uniform distribution with mean 62.36 . That's why we decided to take imdb_rating as our dependent variable.

Categorical Variable plots:



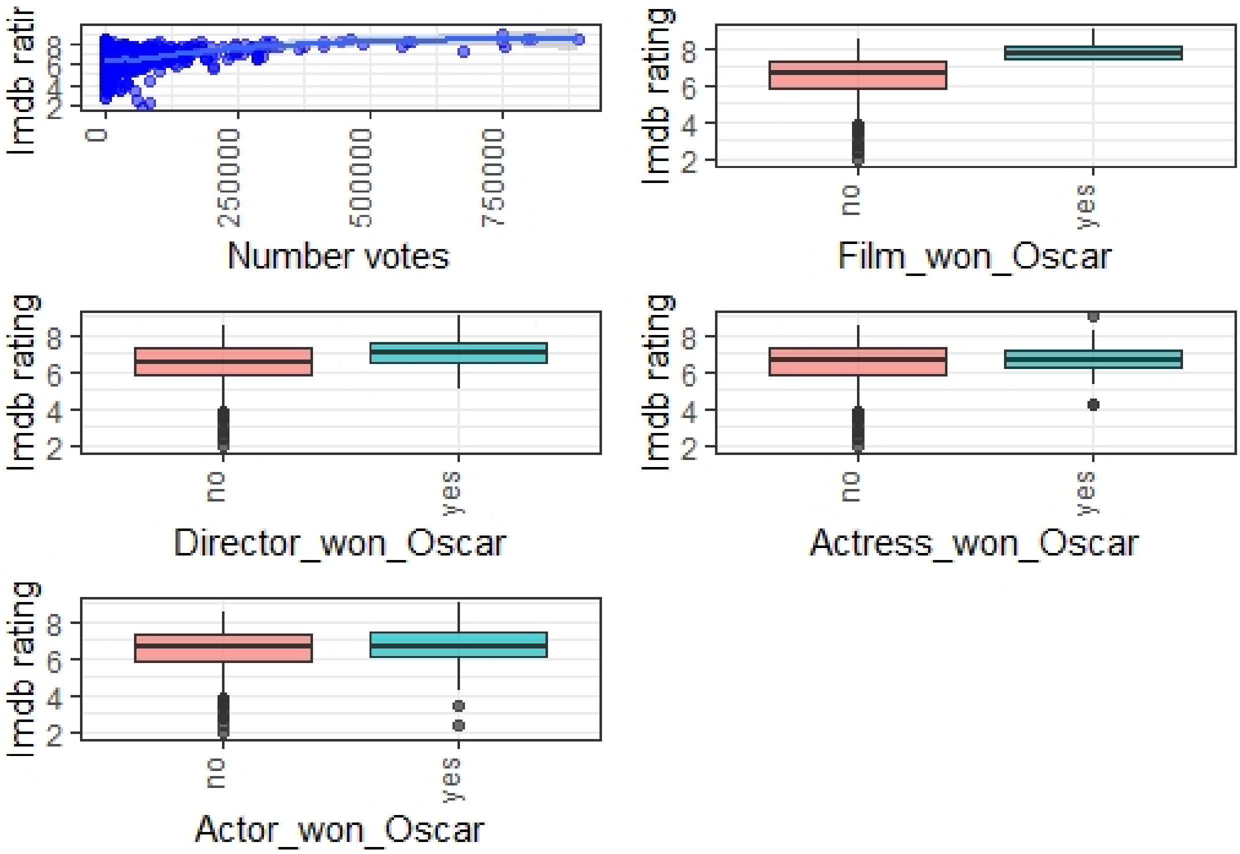
Continuous Variable plot:





From this continuous variable plot we can see that imdb_num_votes is right skewed. To make this variable's distribution as normal we take the log transformation.

Some more plots:



From the plots and the summary descriptive obtained, it can be seen that, in our dataset, those movies that won an *Oscar* or the director ever won an *Oscar* appear to have a slightly higher rating. Moreover, the number of votes given show a weak positive association with the IMDB rating. Last, the variables best_actor_win and best_actress_win appear to have the same distribution and a similar association with imdb_rating, so we will combine these two variables in a new one called main_oscar_win.

MODEL BUILDING (MULTIPLE LINEAR REGRESSION):

To build a multiple linear regression model we first include only six variables i.e. genre, best_pic_win, best_dir_win, main_oscar_win, log_votes and mpaa_ratin g

- **VARIABLE SELECTION PROCEDURE:**

  To select important variable which can predict Imdb_rating we perform backward elimination method and the results of this method is given below:

```
Call:
lm(formula = imdb_rating ~ genre + best_dir_win + log_votes +
    mpaa_rating, data = movies)

Coefficients:
                  (Intercept)                genreAnimation  genreArt House & International
                     3.504558                     -0.450241                        1.050237
                  genreComedy             genreDocumentary                      genreDrama
                     0.076515                      2.220150                        0.920872
                  genreHorror  genreMusical & Performing Arts        genreMystery & Suspense
                     0.001477                      1.691405                        0.586803
                   genreOther  genreScience Fiction & Fantasy                 best_dir_winyes
                     0.802540                     -0.166562                        0.330614
                    log_votes                mpaa_ratingNC-17                   mpaa_ratingPG
                     0.291937                     -0.203334                       -0.544136
               mpaa_ratingPG-13                  mpaa_ratingR             mpaa_ratingUnrated
                    -0.956184                     -0.595780                       -0.105652
```

  After performing this method, we get genre, best_dir_win, log_votes and mpaa_rating are the best independent variables which can predict imdb_rating.

- **REDUCED MODEL:**

  The summary statistics of this reduced model is given in below:

```
Coefficients:
                               Estimate Std. Error t value Pr(>|t|)    
(Intercept)                    3.504558   0.325767  10.758  < 2e-16 ***
genreAnimation                -0.450241   0.327185  -1.376 0.169276    
genreArt House & International 1.050237   0.255683   4.108 4.52e-05 ***
genreComedy                    0.076515   0.140085   0.546 0.585117    
genreDocumentary               2.220150   0.189429  11.720  < 2e-16 ***
genreDrama                     0.920872   0.118924   7.743 3.85e-14 ***
genreHorror                    0.001477   0.209140   0.007 0.994366    
genreMusical & Performing Arts 1.691405   0.268103   6.309 5.28e-10 ***
genreMystery & Suspense        0.586803   0.154910   3.788 0.000166 ***
genreOther                     0.802540   0.235773   3.404 0.000706 ***
genreScience Fiction & Fantasy -0.166562  0.299285  -0.557 0.578044    
best_dir_winyes                0.330614   0.135739   2.436 0.015140 *  
log_votes                      0.291937   0.022649  12.890  < 2e-16 ***
mpaa_ratingNC-17              -0.203334   0.636184  -0.320 0.749365    
mpaa_ratingPG                 -0.544136   0.229996  -2.366 0.018289 *  
mpaa_ratingPG-13              -0.956184   0.234244  -4.082 5.04e-05 ***
mpaa_ratingR                  -0.595780   0.227769  -2.616 0.009116 ** 
mpaa_ratingUnrated            -0.105652   0.260689  -0.405 0.685408    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8398 on 633 degrees of freedom
Multiple R-squared:  0.4163,	Adjusted R-squared:  0.4006 
F-statistic: 26.56 on 17 and 633 DF,  p-value: < 2.2e-16
```

  Here adjusted R-squared value is 0.4006 which is very less.

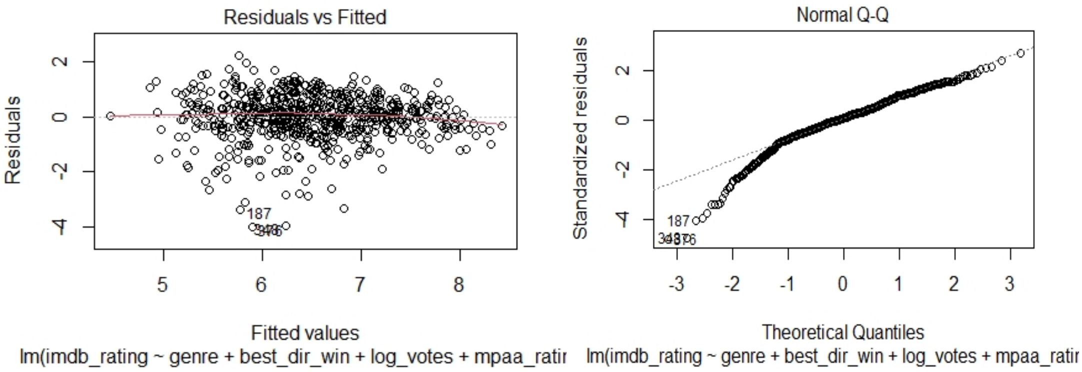
- **CHECKING FOR MULTICOLLINEARITY:**

  Using VIF values we check whether multicollinearity has present in our model or not.

```
> vif(reduced_model)
                GVIF Df GVIF^(1/(2*Df))
genre       2.546252 10        1.047840
best_dir_win 1.049144  1        1.024277
log_votes   1.325889  1        1.151473
mpaa_rating 2.416571  5        1.092245
```

  As vif values are very small there is no multicollinearity presents in our model.

- **<u>SOME PLOTS OF OUR MODEL:</u>**



Here residuals are randomly scattered in a band with a constant width around 0. The QQ plot shows that the dataset is close to normally distributed.

BRIEF CONCEPT OF BAYESIAN MODELLING:

Now it's time for building Bayesian linear regression model. Bayesian model is mainly based on two concepts conditional probability and Bayes theorem.

*Conditional probability i*s the probability that an event will happen given that another event took place. If the event B is known or assumed to have taken place, then the conditional probability of our event of interest A given B is written as $P(A|B)$.

According to Bayes theorem $P(A|B)$ can be written as,

$$p(A|B) = p(A)\ p(B|A)\ /\ p(B)$$

To put this on words: the probability of A given that B have occurred is calculated as the unconditioned probability of A occurring multiplied by the probability of B occurring if A happened, divided by the unconditioned probability of B. In Bayesian context these notations have following meanings,

- $p(A)$ is the probability of the hypothesis before we see the data, called the prior probability, or just **prior**.
- $p(A|B)$ is our goal, this is the probability of the hypothesis after we see the data, called the **posterior**.
- $p(B|A)$ is the probability of the data under the hypothesis, called the **likelihood**.
- $p(B)$ is the probability of the data under any hypothesis, called the **normalizing constant**.

MODEL BUILDING (BAYESIAN LINEAR REGRESSION):

To implement this Bayesian linear regression model, we used `BAS` package in R. parameters of this functions are as below,

Prior: **Zellner-Siow Cauchy** prior distribution.

Model prior: Uniform (assign equal probabilities to all models)

Method: Markov Chain Monte Carlo (**MCMC**) (improves the model search efficiency)

```r
library('BAS')
movies_bas <- bas.lm(imdb_rating ~ .,
                     data = movies_final,
                     method = "MCMC",
                     prior = "ZS-null",
                     modelprior = uniform())
```

- **Zellner-Siow Cauchy prior:**

  As there are no information or belief about this data, we take the prior as non-informative prior. Here Zellner-Siow Cauchy prior is a non-informative prior. This is a mixture of g-priors with an inverse Gamma prior, Inv-Gamma (g | 1/2, n/2), on g, namely,

$$\pi(\boldsymbol{\beta}_\gamma \mid \phi) \propto \int N(\boldsymbol{\beta}_\gamma \mid \mathbf{0}, \tfrac{g}{\phi}(\mathbf{X}_\gamma^T \mathbf{X}_\gamma)^{-1})\ \pi(g)\ dg$$

where

$$\pi(g) = \frac{(n/2)^{1/2}}{\Gamma(1/2)}\ g^{-3/2} e^{-n/(2g)}\ .$$

For detailed information about this prior check out the following links,

Zellner-Siow Cauchy Prior (duke.edu)

The Marginal posterior inclusion probability is given by,

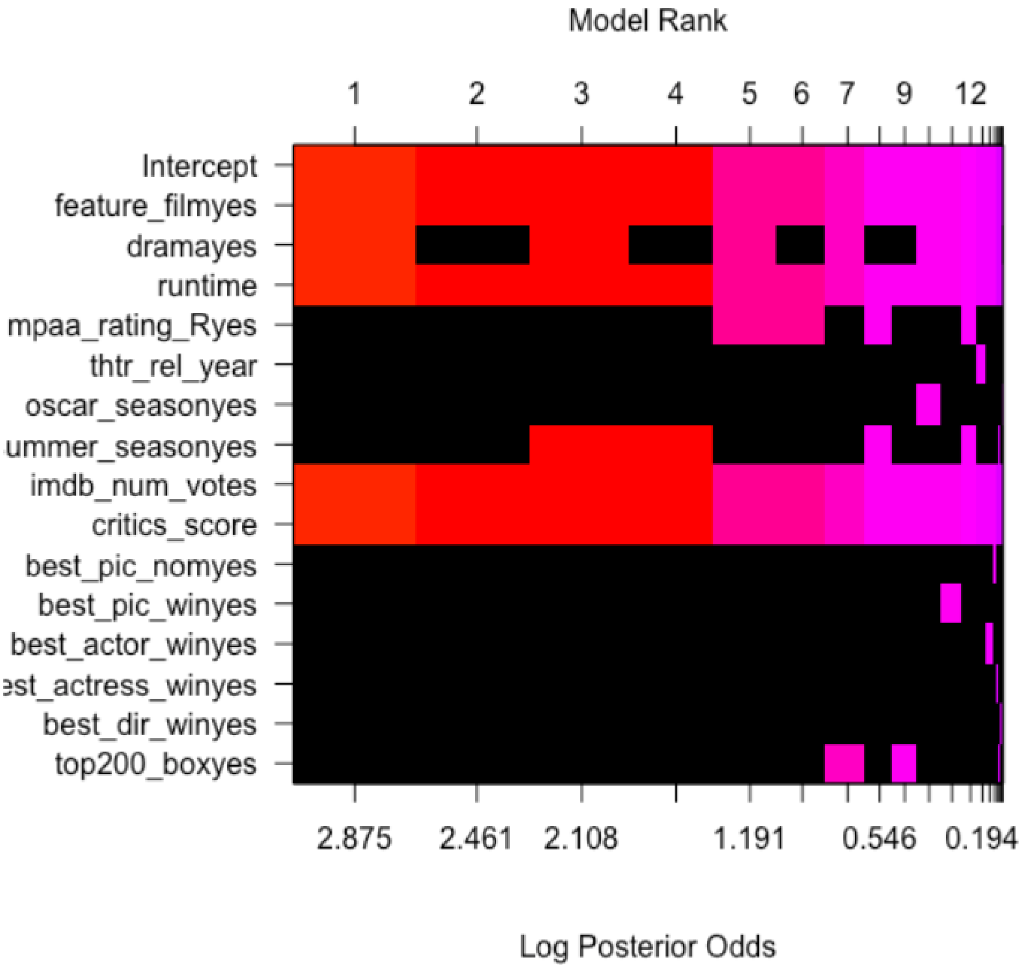
```
Call:
bas.lm(formula = imdb_rating ~ ., data = movies_final, prior = "ZS-null",
    modelprior = uniform(), method = "MCMC")


 Marginal Posterior Inclusion Probabilities:
         Intercept      feature_filmyes            dramayes             runtime      mpaa_rating_Ryes
           1.00000              1.00000             0.57670             0.98051               0.17350
     thtr_rel_year      oscar_seasonyes    summer_seasonyes      imdb_num_votes          critics_score
           0.06793              0.07858             0.34396             0.99998               0.99998
    best_pic_nomyes       best_pic_winyes   best_actor_winyes  best_actress_winyes       best_dir_winyes
           0.06111              0.08554             0.05861             0.05898               0.05727
      top200_boxyes
           0.11992
```

Summary of the model to see top 5 models,

```
                    P(B != 0 | Y)  model 1     model 2     model 3     model 4     model 5
Intercept              1.00000000   1.0000   1.0000000   1.0000000   1.0000000   1.0000000
feature_filmyes        0.99999847   1.0000   1.0000000   1.0000000   1.0000000   1.0000000
dramayes               0.57669830   1.0000   0.0000000   1.0000000   0.0000000   1.0000000
runtime                0.98051147   1.0000   1.0000000   1.0000000   1.0000000   1.0000000
mpaa_rating_Ryes       0.17350006   0.0000   0.0000000   0.0000000   0.0000000   1.0000000
thtr_rel_year          0.06792908   0.0000   0.0000000   0.0000000   0.0000000   0.0000000
oscar_seasonyes        0.07858276   0.0000   0.0000000   0.0000000   0.0000000   0.0000000
summer_seasonyes       0.34396057   0.0000   0.0000000   1.0000000   1.0000000   0.0000000
imdb_num_votes         0.99998322   1.0000   1.0000000   1.0000000   1.0000000   1.0000000
critics_score          0.99998474   1.0000   1.0000000   1.0000000   1.0000000   1.0000000
best_pic_nomyes        0.06111145   0.0000   0.0000000   0.0000000   0.0000000   0.0000000
best_pic_winyes        0.08553619   0.0000   0.0000000   0.0000000   0.0000000   0.0000000
best_actor_winyes      0.05860901   0.0000   0.0000000   0.0000000   0.0000000   0.0000000
best_actress_winyes    0.05898285   0.0000   0.0000000   0.0000000   0.0000000   0.0000000
best_dir_winyes        0.05727081   0.0000   0.0000000   0.0000000   0.0000000   0.0000000
top200_boxyes          0.11991730   0.0000   0.0000000   0.0000000   0.0000000   0.0000000
BF                             NA   1.0000   0.6524129   0.4482058   0.4023594   0.1831721
PostProbs                      NA   0.1750   0.1157000   0.0813000   0.0742000   0.0325000
R2                             NA   0.6408   0.6371000   0.6431000   0.6398000   0.6421000
dim                            NA   6.0000   5.0000000   7.0000000   6.0000000   7.0000000
logmarg                        NA 314.5823 314.1552445 313.7798194 313.6719125 312.8849932
```

Here for each 5 models, Bayes factor, posterior probabilities, R^2 and dimension of the model is given.

Visualization of Log Posterior Odds and Model Rank,



From this plot we can tell that,

feature_film has a marginal probability of 0.999, and appears in all five top models

critics_score has a marginal probability of 0.999 and also appears in all five top models

runtime has a marginal probability of 0.98 and appears in all five top models

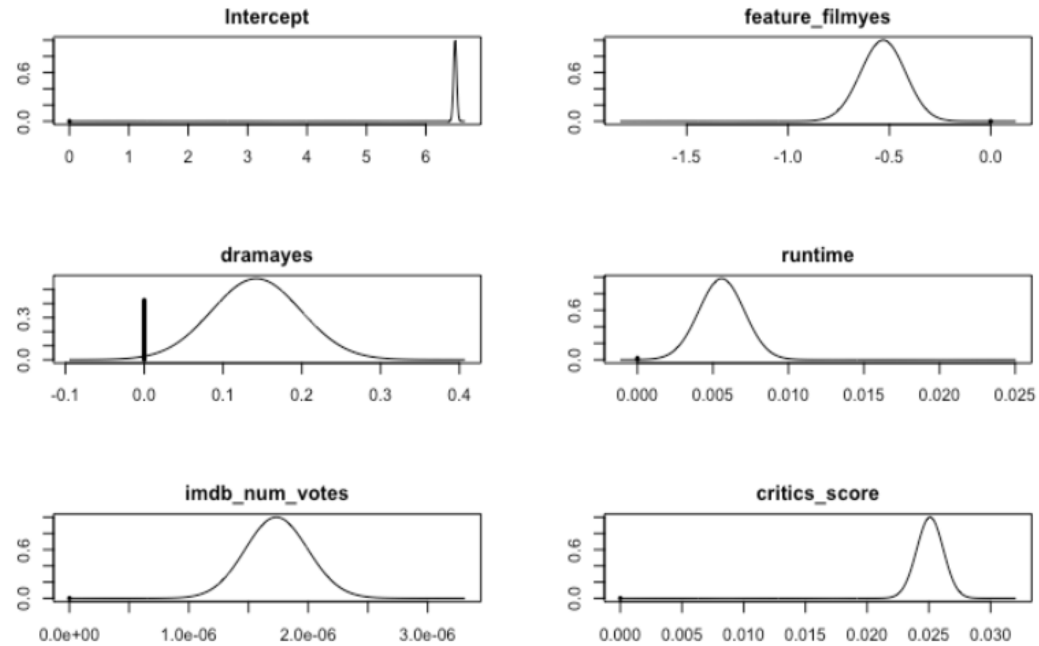
drama has a marginal probability of 0.57 and appears in three of the five top models imbd_num_votes has a marginal probability of 0.99 and appears in three of the five top models

the *intercept* also has a marginal probability of 1, and appears in all five top models

According to this, the best model includes the

intercept, feature_film, critics_score, drama, imbd_num_votes and runtime

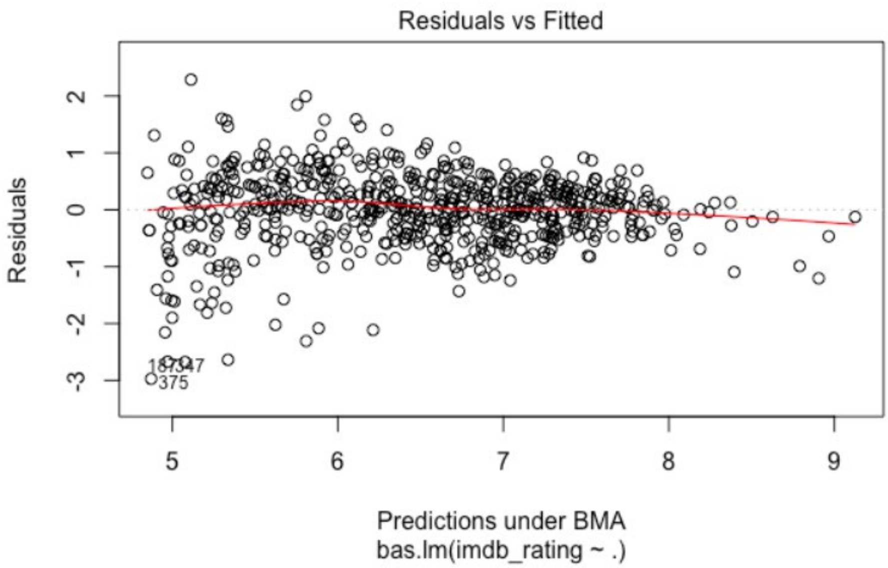
Probability Distribution of coefficients of Bayesian linear regression model:



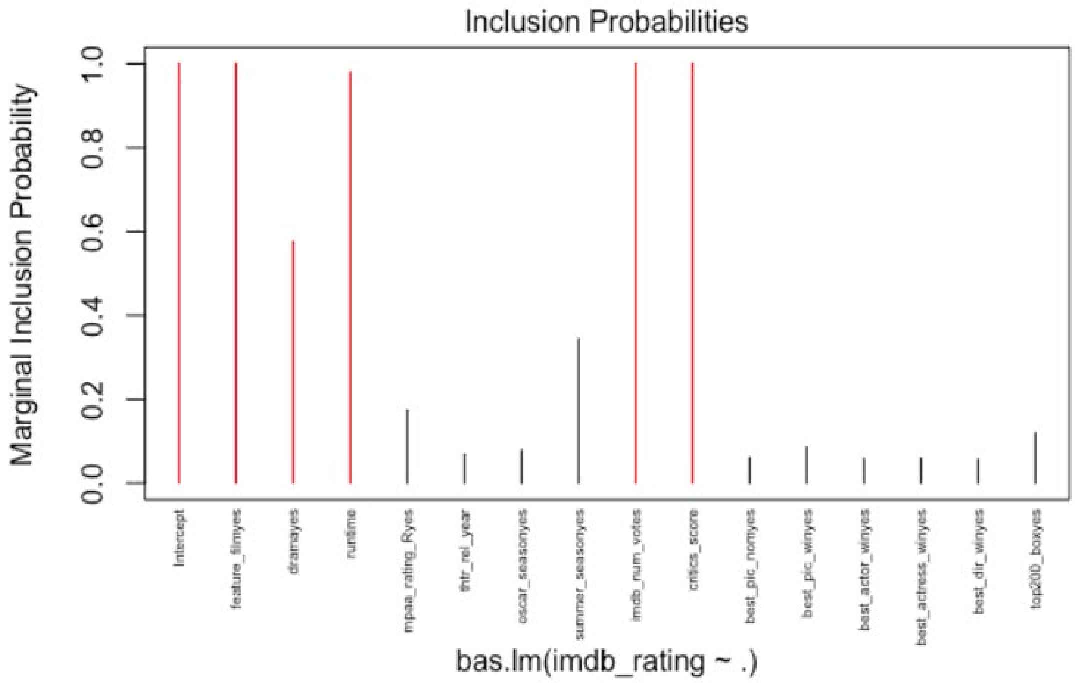
Now the 95% credible interval (The probability that the true mean is contained within a given interval is 0.95) for each of the significant variables are,

```
                           2.5%         97.5%          beta
Intercept            6.441269e+00  6.540970e+00  6.491538e+00
feature_filmyes     -7.506111e-01 -3.272220e-01 -5.323906e-01
dramayes             0.000000e+00  2.227271e-01  8.213439e-02
runtime              2.244793e-03  8.618580e-03  5.459517e-03
mpaa_rating_Ryes    -1.905161e-04  1.227913e-01  1.465901e-02
thtr_rel_year       -1.880919e-03  1.235919e-06 -1.018876e-04
oscar_seasonyes     -5.086382e-03  5.788970e-02  3.432267e-03
summer_seasonyes    -1.792779e-01  3.199856e-04 -3.995598e-02
imdb_num_votes       1.214750e-06  2.249993e-06  1.737390e-06
critics_score        2.314062e-02  2.728858e-02  2.508962e-02
best_pic_nomyes     -5.258736e-03  6.511025e-02  2.845958e-03
best_pic_winyes     -2.785934e-01  1.960888e-03 -2.103027e-02
best_actor_winyes   -1.831484e-02  3.156531e-04  5.131402e-04
best_actress_winyes -2.737283e-02  1.170712e-02 -1.143805e-03
best_dir_winyes     -1.646210e-02  2.212813e-04 -7.061217e-04
top200_boxyes       -3.066508e-01  0.000000e+00 -2.720183e-02
attr(,"Probability")
[1] 0.95
attr(,"class")
[1] "confint.bas"
```

Some graphical summaries of our model:



we can see that there is a constant spread over the prediction but there are two outliers presents in our final dataset.



In this case, we can observe the marginal posterior inclusion probabilities for each of the covariates, with marginal posterior inclusion probabilities that are greater than 0.5 shown in red (important variables for explaining the data and prediction). In the graph, we can see what it was show already before about which variables contribute to the final scores.

PREDICTION:

Now it's time to see the predictive power of two models. For prediction purpose we take a new data named as "*Zootropolis*" released in 2016. The corresponding information was obtained from the IMDB website and Rotten Tomatoes.

- **Prediction using Multiple Linear Regression:**

Movie <fctr>	Predicted rating <fctr>	95% CI <fctr>	IMDb rating <dbl>
Zootropolis	7.1	5.4–8.8	8

- **Prediction using Bayesian Linear Regression:**

Movie <fctr>	Estimated.IMDB.rating <dbl>	Real.IMDB.rating <dbl>
Zootropolis	7.913177	8

CONCLUSION:

It is clearly seen that for the movie 'Zootropolis' the actual IMDB rating is 8.0, using linear regression we get the predicted rating as 7.1. But using Bayesian linear regression model we get the predicted rating as 7.913 which is very much closer to the actual IMDB rating. For Frequentist approach R^2 value is around 40% and for Bayesian approach we seen that for top 5 model the R^2 value is around 64% which is pretty much good than the frequentist approach.

REFERENCES:

1. BAYESIAN INFERENCE IN STATISTICAL ANALYSIS by GEORGE E. P. BOX and GEORGE C. TJAO Department of Statistics, University of Wisconsin

2. MONTE CARLO STATISTICAL METHODS by Christian P. Robert George Casella

3. Exploratory Data Analysis with *R* (2016) by Peng Roger D.

APPENDIX:

```
library(gridExtra)

library(ggplot2)

library(car)

attach(movies)

View(movies)

str(movies)


##******DATA CLEANING*****


## scatter plot between audience score and imdb rating

ggplot(movies, aes(x = audience_score, y = imdb_rating)) +

  geom_point() + stat_smooth(method = "lm") +

labs(title = "scatter plot between audience score and imdb rating",

    x = "Score in Rotten Tomatoes",

    y = "Ratings in Imdb")


## correlation find

cor(audience_score,imdb_rating)
```

```
## Distribution of imdb rating

ggplot(movies, aes(x=imdb_rating)) +

  geom_histogram(fill="lightgreen", alpha = 0.7)+

  theme_bw()+

  labs(x = "Imdb rating", y= "Count", title = "Distribution of Imdb rating")

summary(imdb_rating)



## Distribution of audience score

ggplot(movies, aes(x=audience_score)) +

  geom_histogram(fill="red", alpha = 0.7)+

  theme_bw()+

  labs(x = "Audience score", y= "Count", title = "Distribution of Audience score")

summary(audience_score)



## Categorical Variables plot

f1 = ggplot(movies, aes(x=critics_rating)) +

  geom_bar(fill="blue")

f2 = ggplot(movies, aes(x=audience_rating)) +
```

```r
  geom_bar(fill="blue")

f3 = ggplot(movies, aes(x=best_pic_nom)) +

  geom_bar(fill="blue")

f4 = ggplot(movies, aes(x=best_pic_win)) +

  geom_bar(fill="blue")

f5 = ggplot(movies, aes(x=best_dir_win)) +

  geom_bar(fill="blue")

f6 = ggplot(movies, aes(x=best_actor_win)) +

  geom_bar(fill="blue")

f7 = ggplot(movies, aes(x=best_actress_win)) +

  geom_bar(fill="blue")

f8 = ggplot(movies, aes(x=top200_box)) +

  geom_bar(fill="blue")

grid.arrange(f1, f2, f3, f4, f5, f6, f7, f8, nrow = 4)


## Continuous variable plot

ggplot(movies, aes(x=imdb_num_votes)) +

  geom_histogram(fill="green", alpha = 0.7)+

  theme_bw()+
```

```
  labs(x = "Number of votes of imdb", y= "Count", title = "Distribution of imdb_num_votes")

summary(imdb_num_votes)

ggplot(movies, aes(x=critics_score)) +

  geom_histogram(fill="green", alpha = 0.7)+

  theme_bw()+

  labs(x = "Critics score on Rotten Tomatoes", y= "Count", title = "Distribution of critics_score")

summary(critics_score)


## some more plots

p1 <- ggplot(movies, aes(x=imdb_num_votes, y = imdb_rating))+

  geom_point(colour = "blue", alpha = 0.5)+

  theme_bw()+

  geom_smooth()+

  labs(x = "Number votes", y= "Imdb rating", fill = "won_oscar")+

  theme(axis.text.x=element_text(angle=90, hjust = 1, vjust = 0))+

  theme(legend.position="none")

p2 <- ggplot(movies, aes(x=best_pic_win, y = imdb_rating, fill = best_pic_win))+

  geom_boxplot(alpha = 0.7)+
```

```
  theme_bw()+
  labs(x = "Film_won_Oscar", y= "Imdb rating", fill = "best_pic_win")+
  theme(axis.text.x=element_text(angle=90, hjust = 1, vjust = 0))+
  theme(legend.position="none")
p3 <- ggplot(movies, aes(x=best_dir_win, y = imdb_rating, fill = best_dir_win))+
  geom_boxplot(alpha = 0.7)+
  theme_bw()+
  labs(x = "Director_won_Oscar", y= "Imdb rating", fill = "best_dir_win")+
  theme(axis.text.x=element_text(angle=90, hjust = 1, vjust = 0))+
  theme(legend.position="none")


p4 <- ggplot(movies, aes(x=best_actress_win, y = imdb_rating, fill =
best_actress_win))+
  geom_boxplot(alpha = 0.7)+
  theme_bw()+
  labs(x = "Actress_won_Oscar", y= "Imdb rating", fill = "best_actress_win")+
  theme(axis.text.x=element_text(angle=90, hjust = 1, vjust = 0))+
  theme(legend.position="none")
p5 <- ggplot(movies, aes(x=best_actor_win, y = imdb_rating, fill =
best_actor_win))+
  geom_boxplot(alpha = 0.7)+
```

```
  theme_bw()+
  labs(x = "Actor_won_Oscar", y= "Imdb rating", fill = "best_actor_win")+
  theme(axis.text.x=element_text(angle=90, hjust = 1, vjust = 0))+
  theme(legend.position="none")
grid.arrange(p1, p2, p3, p4, p5, nrow = 3)


best_actor_win
best_actress_win
movies['main_oscar_win']=paste(best_actor_win,best_actress_win)
movies['log_votes'] = log(imdb_num_votes)
View(movies)


## Performing backward elimination method
fullmodel <- lm(imdb_rating ~
genre+best_pic_win+best_dir_win+main_oscar_win+log_votes+mpaa_rating,
              data = movies)
step(fullmodel,data=movies,direction = 'backward')


## Performing multiple linear regression method
```

```
reduced_model = lm(imdb_rating ~ genre + best_dir_win + log_votes +
mpaa_rating, data = movies)

plot(reduced_model)


## cheaking for multicolinearity

vif(reduced_model)


## Performing Bayesian linear regression


movies_final<-
data.frame(title_type,genre,runtime,mpaa_rating,thtr_rel_year,imdb_rating,i
mdb_num_votes,

critics_score,best_pic_nom,best_pic_win,best_actor_win,best_actress_win,be
st_dir_win)


View(movies_final)

library('BAS')

movies_bas <- bas.lm(imdb_rating ~ .,

               data = movies_final,

               method = "MCMC",
```

```
                 prior = "ZS-null",

                 modelprior = uniform())


movies_bas

summary(movies_bas)


#visualization of Log Posterior odds and model rank

image(movies_bas, rotate=F)


#plot for coefficients

coef_movies <- coef(movies_bas)

par(mfrow=c(3,2))

plot(coef_movies, subset = c(1, 2, 3, 4, 9, 10), ask=F)


#Residual vs. fitted plot

plot(movies_bas, which = 1, ask=F)


#Marginal inclusion probabilities

plot(movies_bas, which = 4, ask=F)
```

```
# Prediction for Multiple linear regression

zoo <- data.frame(genre="Comedy", mpaa_rating="PG", best_dir_win="yes",

              log_votes = log(345340))

predict_1 <- predict(reduced_model, zoo, interval="predict")

imdb_rating_predictions <- c(8.0, 7.8)

predictions <- data.frame("Movie" = "Zootropolis",

                "Predicted rating" = sprintf("%2.1f", predict_1[1]),

                "95% CI" = sprintf("%2.1f-%2.1f", predict_1[2], predict_1[3]),

                "IMDb rating" = imdb_rating_predictions[1])

predictions


# Prediction for Bayesian linear regression

zootropolis <- data.frame(feature_film = "yes", drama="no",

                runtime=108, mpaa_rating_R = "no",

                thtr_rel_year = 2016, oscar_season = "no",

                summer_season = "no",

                imdb_num_votes = 345433, critics_score=98,

                best_pic_nom = "yes", best_pic_win = "yes",

                best_actor_win = "no", best_actress_win = "no",
```

```
                    best_dir_win = "yes", top200_box = "no")

predict_1 <- predict(movies_bas, zootropolis, estimator="BMA", interval =
"predict", se.fit=TRUE)

data.frame('Movie' = 'Zootropolis',

        'Estimated IMDB rating' = predict_1$Ybma,

        'Real IMDB rating' = 8.0)
```