Bayesian Modeling and Prediction

Author: Jorge Álvarez de la Fuente

Setup

Load packages

```
library(ggplot2)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unable to identify current timezone 'H':
## please set environment variable 'TZ'
```

```
library(dplyr)
library(statsr)
library(BAS)
library(tidyr)
library(MASS)
```

Load data

Make sure your data and R Markdown files are in the same directory. When loaded your data file will be called `movies`. Delete this note when before you submit your work.

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

Part 1: Data

The data set includes information from Rotten Tomatoes and IMDB and is comprised of 651 **randomly sampled** movies produced and released before 2016, as shown in the code below:

```
# This code shows the number of movies (651 obs.) and variables (32 variables)
dim(movies)
```

```
## [1] 651  32
```

Scope of Inference

This is an **observational study** and the data was collected using a **random sampling** method instead of a randomly assigned method. The data set is a representative sample of the United States movies released between 1974 and 2016. Since a random sampling method was used to collect data, the results can be generalized to the population of interest, United States population and movies released between 1974 and 2016.

Causation cannot be established because a **random assignment** method was not used to collect the data.

Potential sources of bias

A potential source of bias can be attributed to nonvoting and non rating given, resulting in a Non-response type of sampling bias. Also, is noteworthy that Rotten Tomatoes audience score has been created voluntarily so this study may present voluntary response bias which may not be representative of the United States population.

Part 2: Data manipulation

New variables from the movies data source have been created as shown in the code below, in order to complement the original variables. These new variables are:

feature_film, drama, mpaa_rating_R, oscar_season, summer_season.

`feature_film`: "Yes", if title_type is Feature Film, "no" otherwise

```
# Code for creation of the new variable feature_film
movies <- mutate(movies, feature_film = ifelse(title_type == "Feature Film", "Yes", "No"))
movies$feature_film <- as.factor(movies$feature_film)
summary(movies$feature_film)
```

```
##  No Yes
##  60 591
```

`drama`: "Yes", if genre is drama, "no" otherwise

```
# Code for creation of the new variable drama
movies <- mutate(movies, drama = ifelse(genre == "Drama", "Yes", "No"))
movies$drama <- as.factor(movies$drama)
summary(movies$drama)
```

```
##  No Yes
## 346 305
```

`mpaa_rating_R`: "Yes", if mpaa_rating is R, "no" otherwise

```
# Code for creation of the new variable mpaa_rating_R
movies <- mutate(movies, mpaa_rating_R = ifelse(mpaa_rating == "R", "Yes", "No"))
movies$mpaa_rating_R <- as.factor(movies$mpaa_rating_R)
summary(movies$mpaa_rating_R)
```

```
##  No Yes
## 322 329
```

`oscar_season`: "Yes", if movie is released in October, November or December (according to variable thtr_rel_month), "no" otherwise

```r
# Code for creation of the new variable oscar_season
movies <- mutate(movies, oscar_season = ifelse(thtr_rel_month %in% c(10,11,12), "Yes", "No"))
movies$oscar_season <- as.factor(movies$oscar_season)
summary(movies$oscar_season)
```

```
##  No Yes
## 460 191
```

summer_season: "Yes", if movie is released in May, June, July or August (according to variable thtr_rel_month), "no" otherwise

```r
# Code for creation of the new variable summer_season
movies <- mutate(movies, summer_season = ifelse(thtr_rel_month %in% c(5,6,7,8), "Yes", "No"))
movies$summer_season <- as.factor(movies$summer_season)
summary(movies$summer_season)
```

```
##  No Yes
## 443 208
```

The code below summarizes the results of the new variables previously created. In addition, the existing variable audience_score has been added as well in the new dataframe to conduct the Exploratory data analysis afterwards.

```r
# This code summarizes the results of all new variables created through the data frame "df"
df <- movies[c("feature_film","drama","mpaa_rating_R","oscar_season","summer_season","audience_score")]

summary(df)
```

```
##  feature_film drama     mpaa_rating_R oscar_season summer_season
##  No : 60      No :346   No :322       No :460      No :443
##  Yes:591      Yes:305   Yes:329       Yes:191      Yes:208
##
##
##
##
##  audience_score
##  Min.   :11.00
##  1st Qu.:46.00
##  Median :65.00
##  Mean   :62.36
##  3rd Qu.:80.00
##  Max.   :97.00
```

The following code shows the names of all variables created:

```r
# This code checks that all variables have been successfully created
tail(names(df),6)
```

```
## [1] "feature_film"   "drama"          "mpaa_rating_R"  "oscar_season"
## [5] "summer_season"  "audience_score"
```

As can be observed in the code above, the new variables have been successfully created.

Part 3: Exploratory data analysis

A exploratory data analysis involving the relationships between `audience_score` and the new variables created in the previous section: `feature_film` , `drama` , `mpaa_rating_R` , `oscar_season` , `summer_season` is shown below.

To study the distribution of the variable `audience_score` a histogram is shown in Figure 1:

```
# This code shows the distribution of the variable 'audience_score'
ggplot(df, aes(x = audience_score, y = ..density..)) +
  geom_histogram(bins = 30, fill = 'deepskyblue2', colour = 'black') +
  geom_density(size = 1.1, colour = 'burlywood4')
```

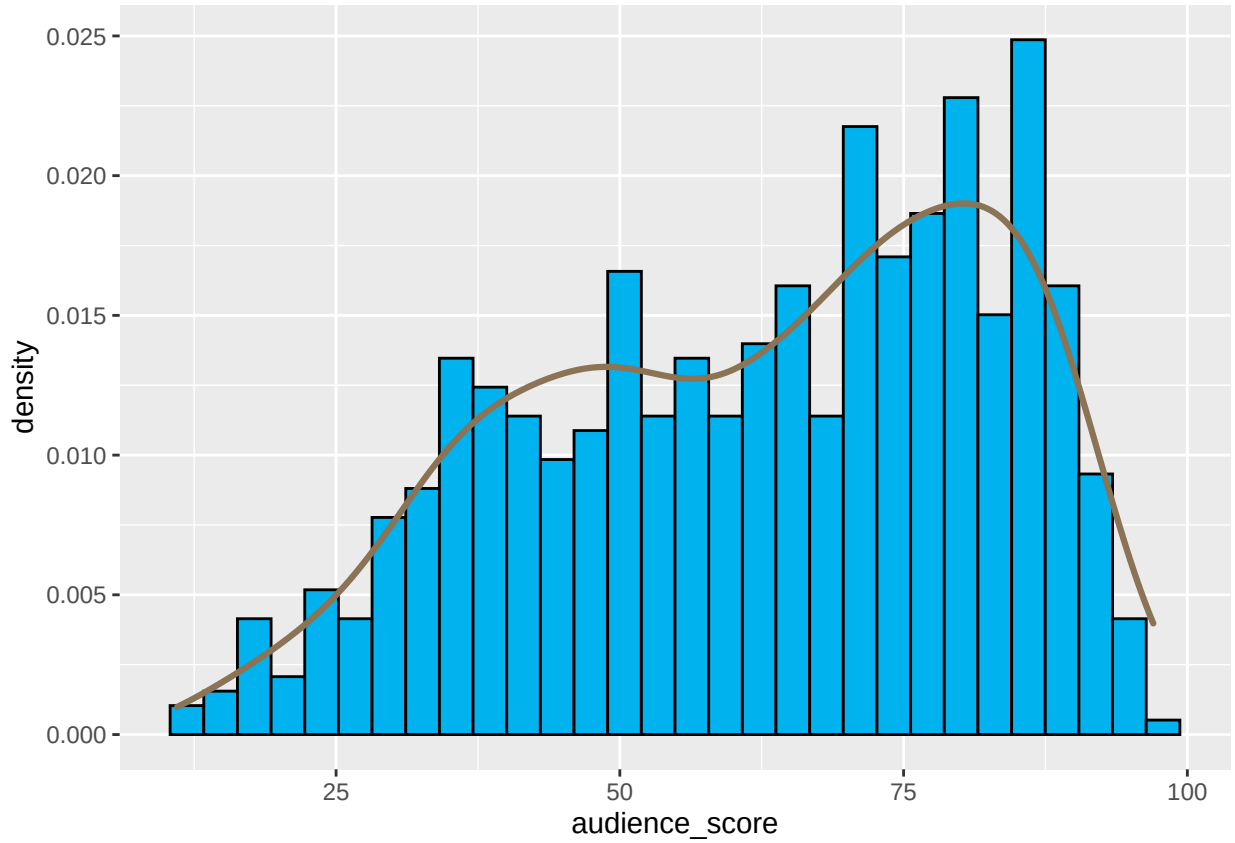


Figure 1: Histogram plot for Audience Score

As observed in Figure 1, audience Score distribution is left skewed.

Box plots visualizations are displayed in the Figures 2-7 comparing the new variables created with `audience_score`:

```
# This code shows box plot visualizations for all new variables created
movies_allnewvariables <- gather(movies, 'variable', 'result', 33:37)

ggplot(movies_allnewvariables, aes(x = variable, y = audience_score, fill = result)) +
  scale_fill_manual(values=c("coral","deepskyblue3")) +
  xlab("Variable Created") + ylab("Audience Score") + ggtitle("Audience Score vs Variable Created") +
  geom_boxplot()
```

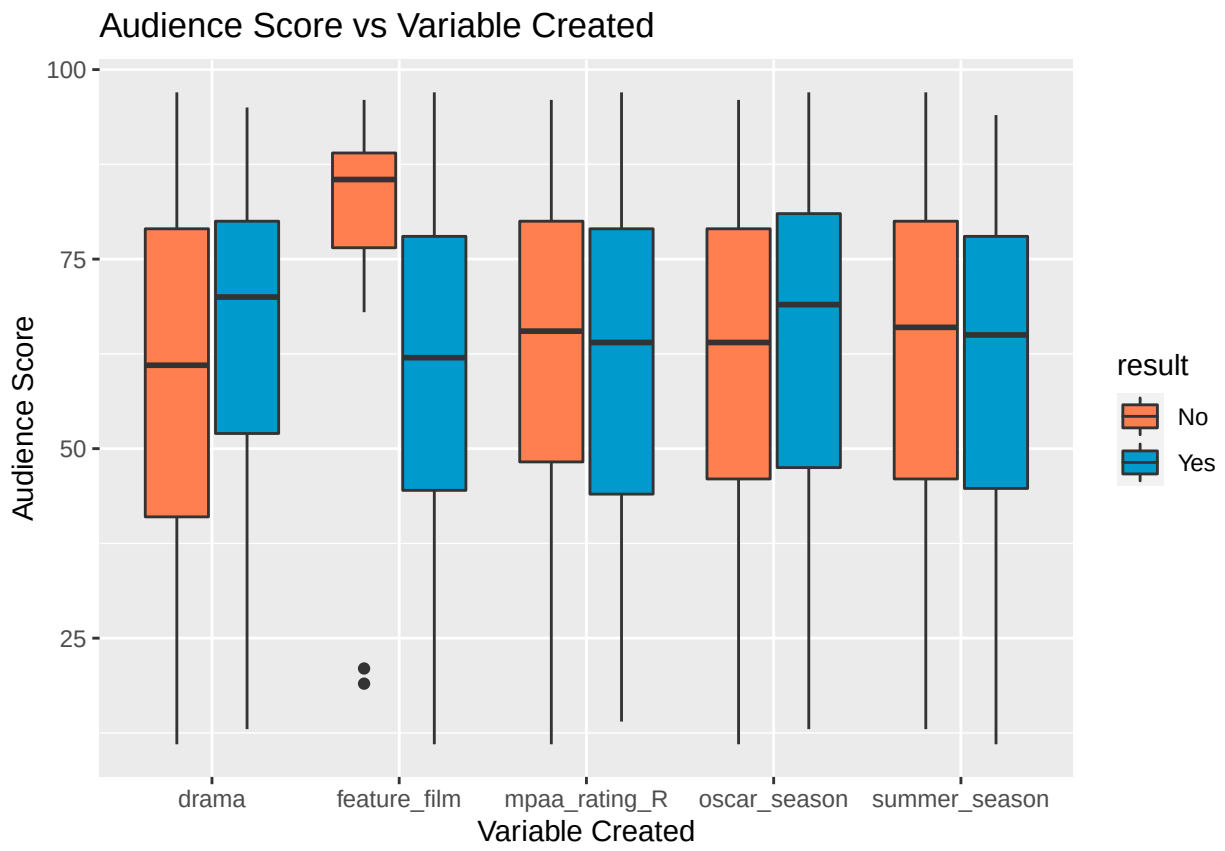


Figure 2: Box plot for Audience Score and new variables created

As observed in Figure 2 the new variables created do not show much variability when compared to Audience Score for results "Yes" , "No". `feature_film` variable shows 2 potential outliers.

```
# This code shows a box plot visualization for feature_film variable
ggplot(df, aes(y=audience_score, x=feature_film, fill = feature_film)) +
  geom_boxplot() +
  xlab("Feature Film") + ylab("Audience Score") + ggtitle("Audience Score vs Feature Film") +
  scale_fill_manual(values=c("coral","deepskyblue3"), name = "Feature Film")
```

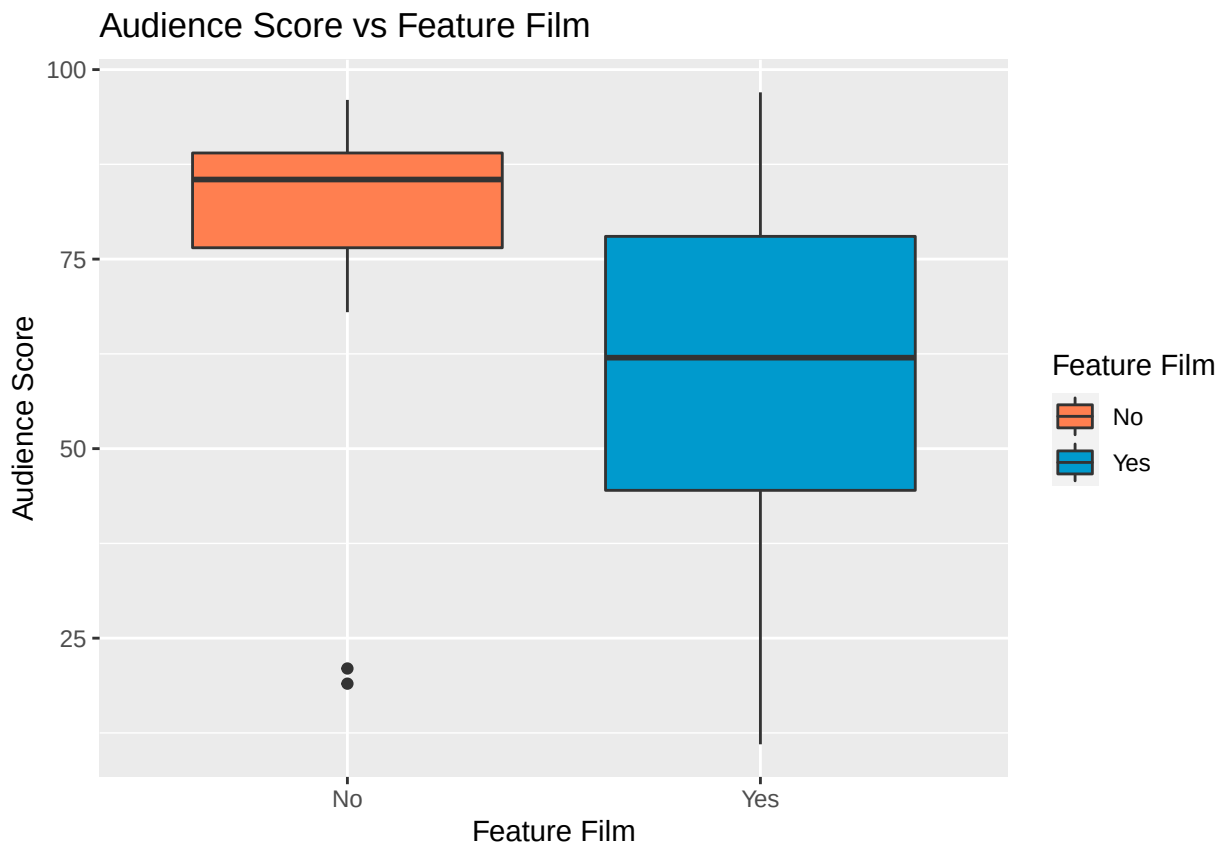


Figure 3: Box plot for Audience Score and Feature Film

From Figure 3, the distribution is more uniform for "Yes" while median is lower compared to "No". "No" result shows 2 potential outliers, as can be confirmed in this boxplot. The IQR is higher for "Yes" compared to "No".

```
# This code shows a box plot visualization for drama variable
ggplot(df, aes(y=audience_score, x=drama, fill = drama)) +
  geom_boxplot() +
  xlab("Drama") + ylab("Audience Score") + ggtitle("Audience Score vs Drama") +
  scale_fill_manual(values=c("coral","deepskyblue3"), name = "Drama")
```

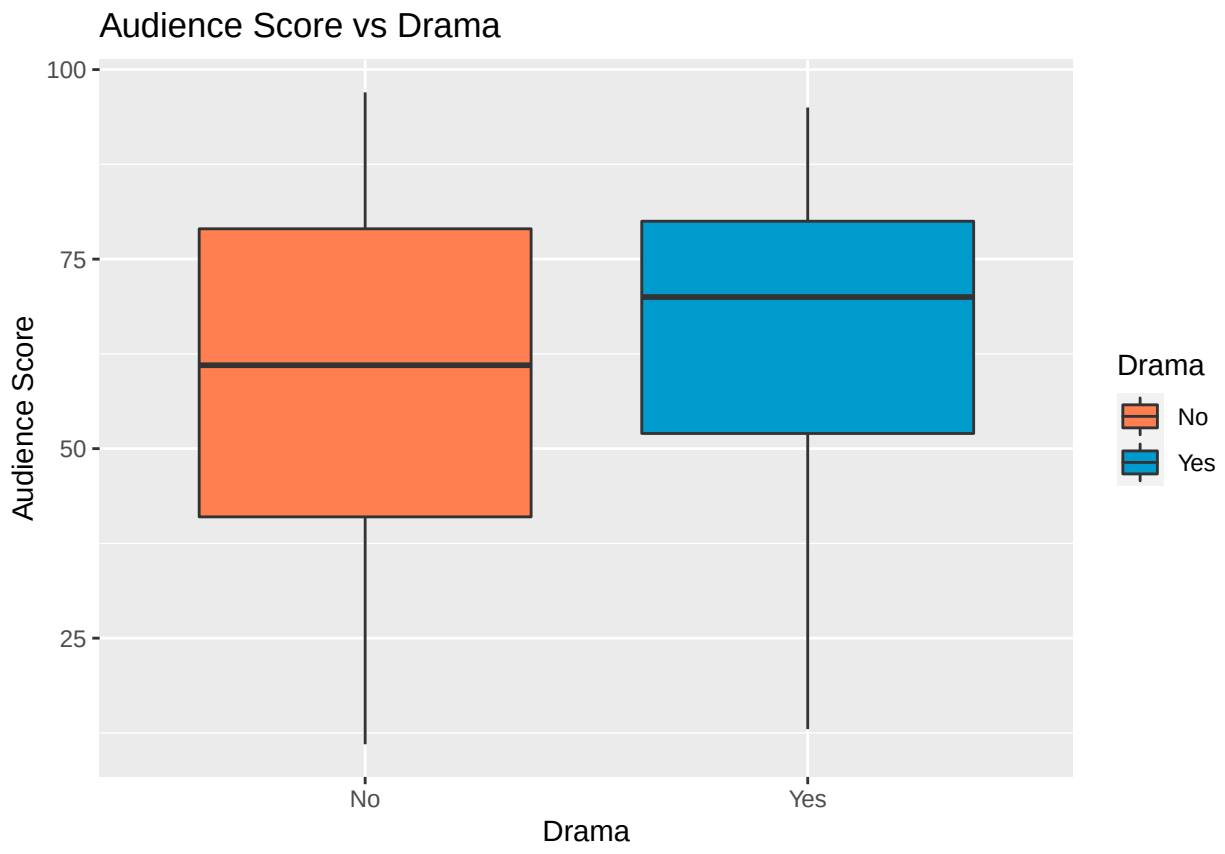


Figure 4: Box plot for Audience Score and Drama

From Figure 4, both results for `drama` variable do not display potential outliers. The median for "Yes" is higher than for "No". Distributions are uniform. The IQR is higher for "No" compared to "Yes".

```
# This code shows a box plot visualization for mpaa_rating_R variable
ggplot(df, aes(y=audience_score, x=mpaa_rating_R, fill = mpaa_rating_R)) +
  geom_boxplot() +
  xlab("mpaa rating R") + ylab("Audience Score") + ggtitle("Audience Score vs mpaa rating R") +
  scale_fill_manual(values=c("coral","deepskyblue3"), name = "mpaa rating R")
```

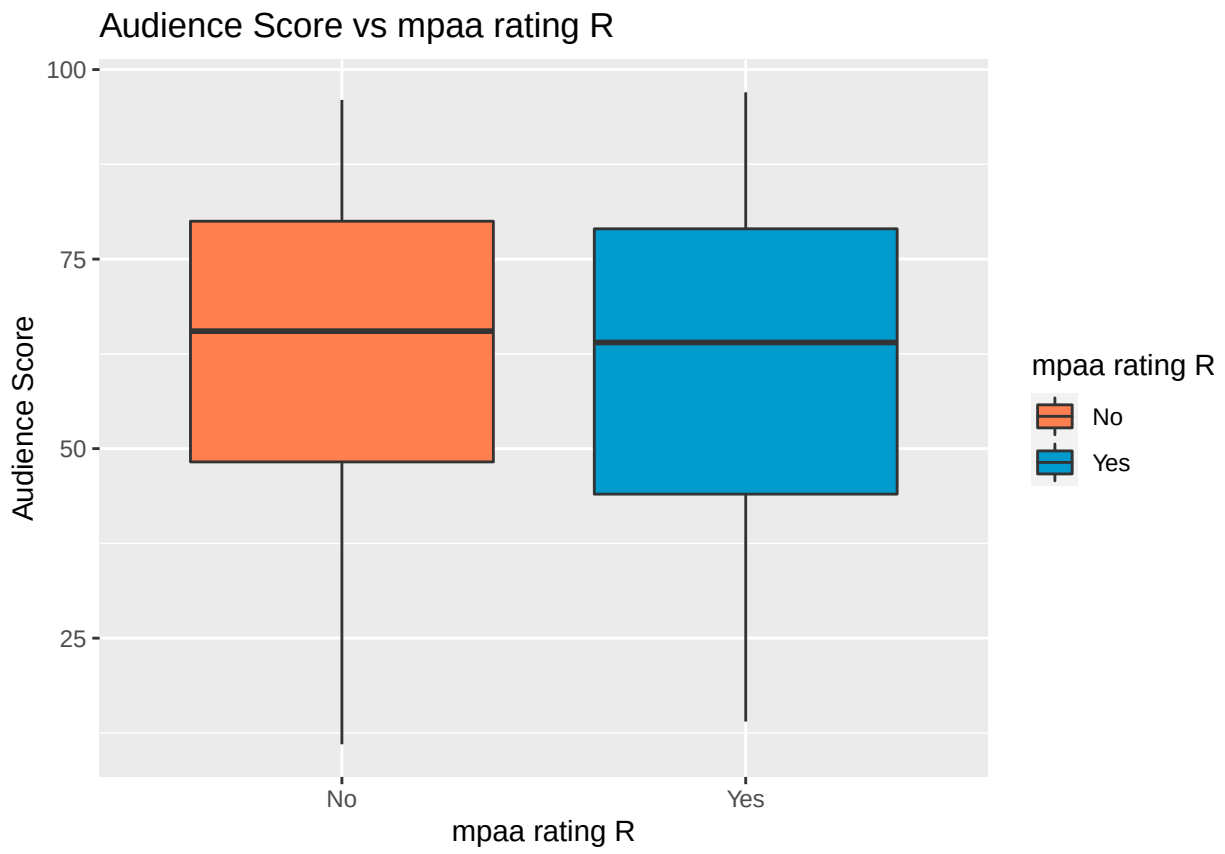


Figure 5: Box plot for Audience Score and mpaa rating R

From Figure 5, both results for `mpaa_rating_R` variable do not display potential outliers. The median for "No" is slightly higher than for "Yes". Distributions are uniform. The IQR is higher for "Yes" compared to "No".

```
# This code shows a box plot visualization for oscar_season variable
ggplot(df, aes(y=audience_score, x=oscar_season, fill = oscar_season)) +
  geom_boxplot() +
  xlab("Oscar Season") + ylab("Audience Score") + ggtitle("Audience Score vs Oscar Season") +
  scale_fill_manual(values=c("coral","deepskyblue3"), name = "Oscar Season")
```

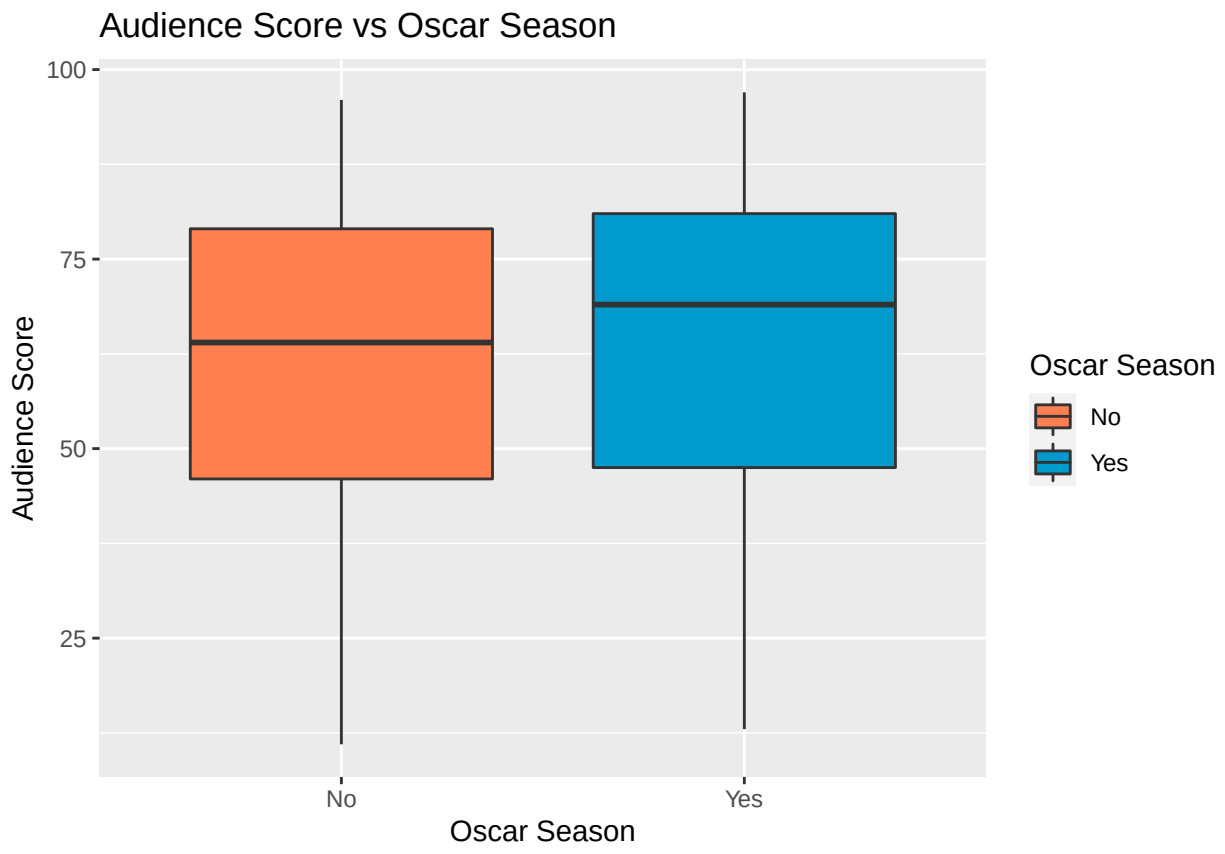


Figure 6: Box plot for Audience Score and Oscar Season

From Figure 6, both results for `oscar_season` variable do not display potential outliers. The median for "Yes" is higher than for "No". Distributions are uniform. The IQR is higher for "Yes" compared to "No".

```
# This code shows a box plot visualization for summer_season variable
ggplot(df, aes(y=audience_score, x=summer_season, fill = summer_season)) +
  geom_boxplot() +
  xlab("Summer Season") + ylab("Audience Score") + ggtitle("Audience Score vs Summer Season") +
  scale_fill_manual(values=c("coral","deepskyblue3"), name = "Summer Season")
```

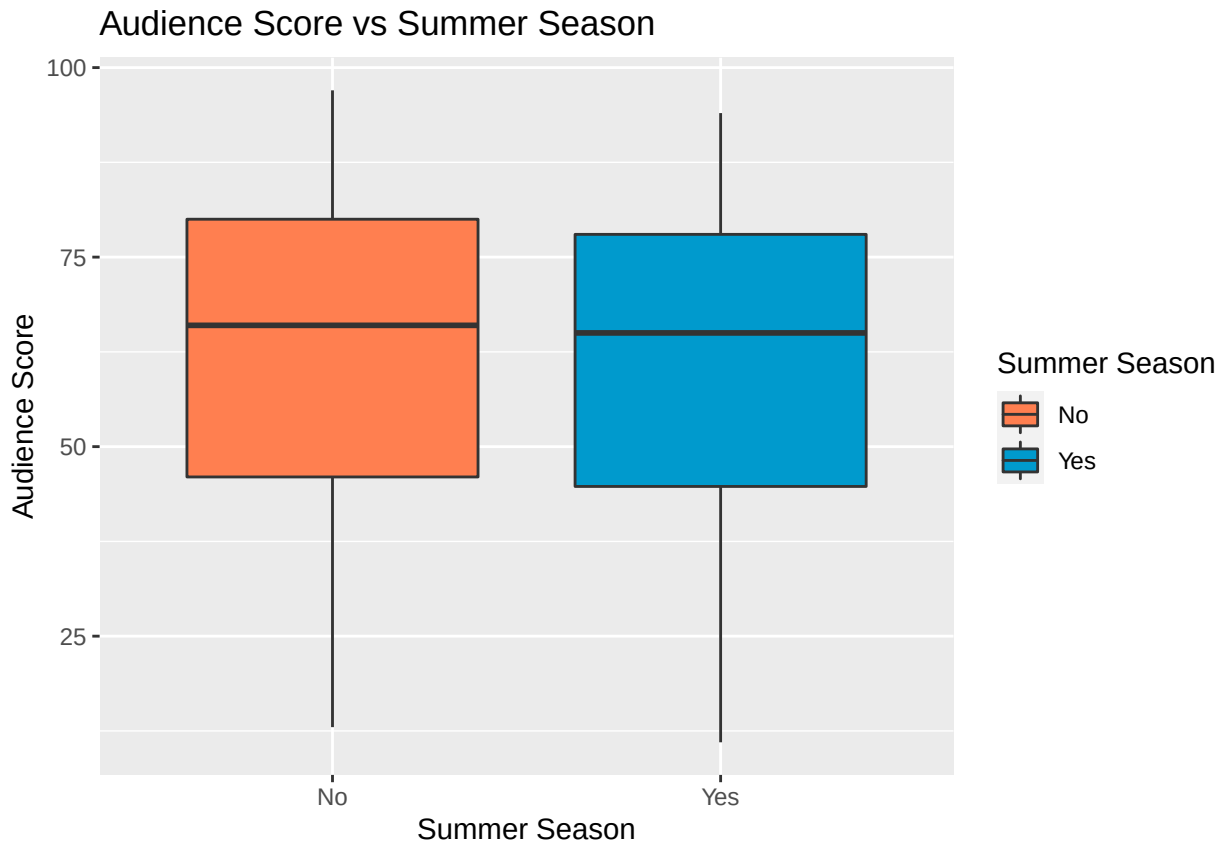


Figure 7: Box plot for Audience Score and Oscar Season

From Figure 7, both results for `summer_season` variable do not display potential outliers. The median for "No" is barely higher than for "Yes". Distributions are uniform.

In conclusion, as observed in Figures 2-7, of all new variables created, only the variable `feature_film` shows a significant variability. The remaining variables `drama` , `mpaa_rating_R` , `oscar_season` , `summer_season` shown an uniform distribution with very little variability. Therefore, the most valuable variable to predict the Audience Score is `feature_film`.

Part 4: Modeling

A stepwise backwards elimination process will be conducted to build the final model. This process begins with the full model, eliminating variables until the lowest BIC (Bayesian Information Criterion) is reached.

The following code builds the full model:

```
# This code builds the full model and conducts the Stepwise backwards elimination process
full_model <- movies[c("feature_film","drama","runtime","mpaa_rating_R","thtr_rel_year","oscar_season",
lm1 <- lm(audience_score ~ ., data = full_model)
score_step <- stepAIC(lm1, trace = FALSE)
score_step$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## audience_score ~ 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
##
## Final Model:
## audience_score ~ runtime + mpaa_rating_R + thtr_rel_year + imdb_rating +
##     critics_score + best_pic_nom + best_actor_win + best_actress_win
##
##
##                Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                                       633   62989.66 3006.940
## 2     - top200_box  1   9.240822       634   62998.90 3005.035
## 3   - oscar_season  1  26.461061       635   63025.36 3003.308
## 4   - best_pic_win  1  45.744109       636   63071.11 3001.780
## 5   - best_dir_win  1  93.998979       637   63165.11 3000.748
## 6  - summer_season  1 166.872376       638   63331.98 3000.463
## 7   - feature_film  1 155.730073       639   63487.71 3000.059
## 8          - drama  1 121.355602       640   63609.06 2999.300
## 9 - imdb_num_votes  1 147.829088       641   63756.89 2998.809
```

BMA (Bayesian Model Averaging) will be used to find the best model.

```
# Exclude observations with missing values in the data set
full_model_no_na <- na.omit(full_model)
# This code uses Bayesian Model Average (BMA) and find summary statistics int he full model
bma_audience_score <- bas.lm(audience_score ~ ., data = full_model_no_na, prior = "BIC", modelprior = u
# Print out the marginal posterior inclusion probabilities for each variable
bma_audience_score
```

```
##
## Call:
## bas.lm(formula = audience_score ~ ., data = full_model_no_na,
##     prior = "BIC", modelprior = uniform())
##
##
##  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
```

```
# Top 5 most probably models
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
```

Printing the model object and the summary command gives the posterior model inclusion probability for each variable and the most probable models. For example, the posterior probability that `critics_score` is included in the model is 0.889. Further, the most likely model, which has posterior probability of 0.1297 , includes the intercept, runtime, imdb_rating, critics_score. There are 2^{16} possible models.

The image function confirms and provides information about which variables have the highest posterior odds to build the final model (Figure 8).

```
# This code shows the image for the data bma_audience_score
image(bma_audience_score)
```

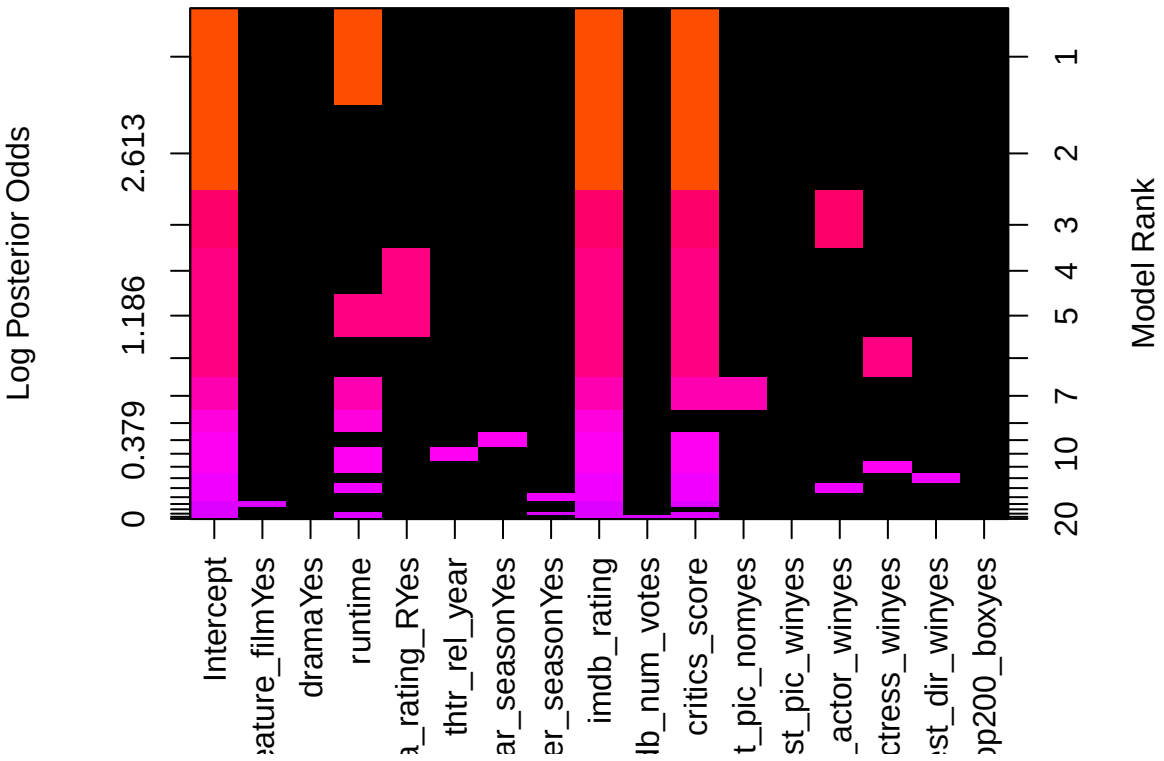
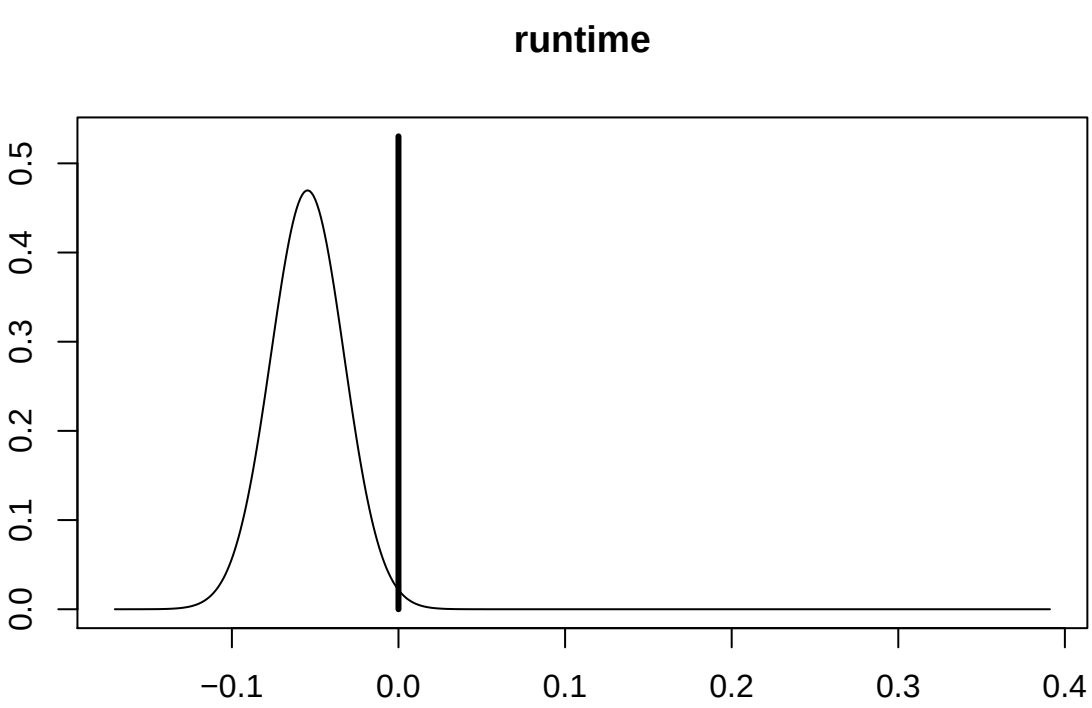


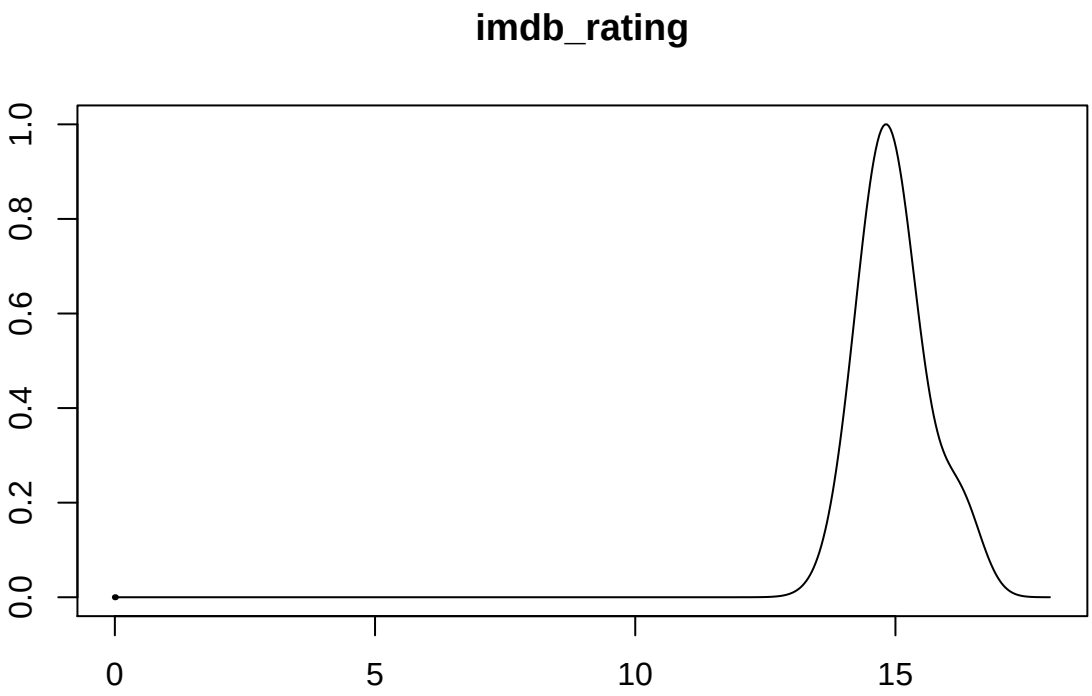
Figure 8: Image for bma_audience_score data

As observed in Figure 8 the variables with the highest posterior odds are `runtime` , `imdb_rating` , `critics_score`. In addition, the following code allows to visualize the posterior distribution of the coefficients under the model averaging approach (Figure 9). The posterior distribution of the coefficients of `runtime` , `imdb_rating` , `critics_score` are shown below.

```
# This code obtains the coefficients from the model bma_audience_score
coef_audience_score <- coefficients(bma_audience_score)

# 'runtime' is the 4th variable, while 'imdb_rating' is the 9th variable and 'critics_score' is the 11t
plot(coef_audience_score, subset = c(4,9,11), ask = FALSE)
```


runtime
0.0
0.1
0.2
0.3
0.4
0.5
−0.1
0.0
0.1
0.2
0.3
0.4


imdb_rating
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15

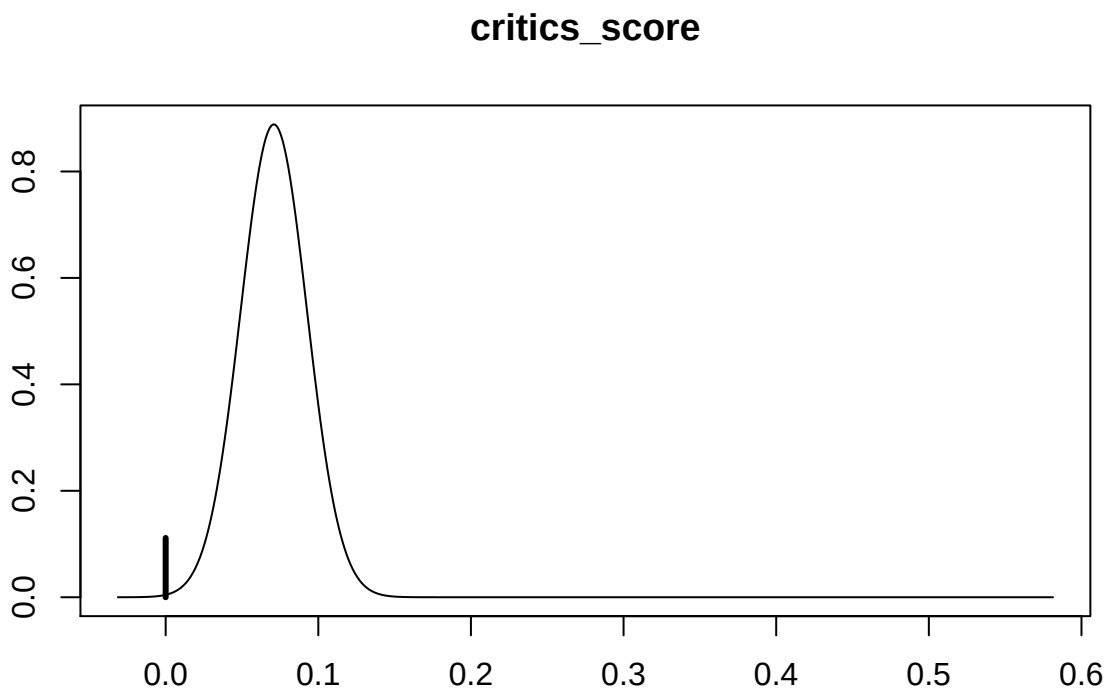


Figure 9: Posterior distribution of the coefficients of `runtime` , `imdb_rating` , `critics_score`

Furthermore, 95% credible intervals for these coefficients are provided below:

```
# 95% credible intervals for the coefficients in audience_score
confint(coef_audience_score)
```

```
##                              2.5%        97.5%          beta
## Intercept            6.157266e+01 6.310800e+01  6.234769e+01
## feature_filmYes     -1.346569e+00 1.295983e-04 -1.046908e-01
## dramaYes             0.000000e+00 0.000000e+00  1.604413e-02
## runtime             -8.263596e-02 0.000000e+00 -2.567772e-02
## mpaa_rating_RYes    -2.084809e+00 0.000000e+00 -3.036174e-01
## thtr_rel_year       -5.163459e-02 0.000000e+00 -4.532635e-03
## oscar_seasonYes     -9.748882e-01 7.591486e-03 -8.034940e-02
## summer_seasonYes     0.000000e+00 1.087388e+00  8.704545e-02
## imdb_rating          1.366024e+01 1.651127e+01  1.498203e+01
## imdb_num_votes      -4.983989e-09 1.191250e-06  2.080713e-07
## critics_score        0.000000e+00 1.048088e-01  6.296648e-02
## best_pic_nomyes      0.000000e+00 4.956666e+00  5.068035e-01
## best_pic_winyes      0.000000e+00 0.000000e+00 -8.502836e-03
## best_actor_winyes   -2.650457e+00 0.000000e+00 -2.876695e-01
## best_actress_winyes -2.777487e+00 0.000000e+00 -3.088382e-01
## best_dir_winyes     -1.254392e+00 0.000000e+00 -1.195011e-01
## top200_boxyes        0.000000e+00 0.000000e+00  8.648185e-02
## attr(,"Probability")
```

```
## [1] 0.95
## attr(,"class")
## [1] "confint.bas"
```

The reduced model is provided below as follows:

```
# This code displays BMA with the Zellner-Siow prior on the regression coefficients
reduced_model <- full_model[c("runtime","imdb_rating","critics_score","audience_score")]

bma_reduced_model <- bas.lm(audience_score ~., data = reduced_model, prior = "ZS-null", method = "MCMC"
```

```
## Warning in bas.lm(audience_score ~ ., data = reduced_model, prior = "ZS-null", :
## dropping 1 rows due to missing data
```

```
summary(bma_reduced_model)
```

```
##               P(B != 0 | Y)  model 1     model 2     model 3     model 4
## Intercept           1.00000   1.0000   1.0000000   1.0000000   1.0000000
## runtime             0.45000   0.0000   1.0000000   1.0000000   0.0000000
## imdb_rating         0.99375   1.0000   1.0000000   1.0000000   1.0000000
## critics_score       0.91250   1.0000   1.0000000   0.0000000   0.0000000
## BF                       NA   1.0000   0.8702806   0.1275819   0.1142849
## PostProbs                NA   0.5093   0.3913000   0.0559000   0.0311000
## R2                       NA   0.7525   0.7549000   0.7509000   0.7481000
## dim                      NA   3.0000   4.0000000   3.0000000   2.0000000
## logmarg                  NA 443.9495 443.8105657 441.8905087 441.7804447
##                     model 5
## Intercept      1.000000e+00
## runtime        0.000000e+00
## imdb_rating    0.000000e+00
## critics_score  0.000000e+00
## BF            1.567399e-193
## PostProbs      6.200000e-03
## R2             0.000000e+00
## dim            1.000000e+00
## logmarg        0.000000e+00
```

Based on this reduced data set, according to Bayesian model averaging, the variable with the lowest marginal posterior inclusion probability is `runtime` with a value of 0.44. As can be observed from the summary of BMA with the Zellner-Siow prior on the regression coefficients in the reduced model, the most likely model with the highest posterior probability is the model 1 with a value of 0.49 and contains the intercept, `imdb_rating`, `critics_score`. `imdb_rating` is the predictor with the highest posterior probability with a value of 0.99.

Part 5: Prediction

This section aims to predict the audience score for a new movie that is not in the sample. The movie "Sully" (released on 2016) will be used to make this prediction. The code below checks if the movie "Sully" is in movies data set:

```
# This code checks if the movie "Sully" exists in the movies data set
any(grepl("Sully", movies))
```

```
## [1] FALSE
```

As can be observed, the movie "Sully" is not in the movies data set. The following references for where the data for this movie come from were used to run the prediction through the model:

[1] Rottentomatoes (Sully, 2016) https://www.rottentomatoes.com/m/sully

[2] IMDB (Sully, 2016) https://www.imdb.com/title/tt3263904/?ref_=fn_al_tt_1

The prediction of the Audience Score is shown below:

```
# This code stores the data in a new variable prediction_movie
prediction_movie <- data.frame(feature_film="yes",
                               drama="yes",
                               runtime=96,
                               mpaa_rating_R="no",
                               thtr_rel_year=2016,
                               oscar_season="yes",
                               summer_season="no",
                               imdb_rating=7.4,
                               imdb_num_votes=233633,
                               critics_score=85,
                               best_pic_nom="no",
                               best_pic_win="no",
                               best_actress_win="no",
                               best_dir_win="yes",
                               top200_box="no",
                               audience_score=84)
# This code predicts the audience score
ascore_prediction = predict(bma_reduced_model, newdata = prediction_movie,estimator = "BMA", se.fit = TR
ascore_prediction$Ybma
```

```
##          [,1]
## [1,] 77.82263
```

The prediction gives a result of 77.807 of Audience Score, which is lower than the actual Audience Score with a value of 84.

The credible interval for the Audience Score is also provided below:

```
# This code obtains the credible interval for a confident level of 95% for the Audience Score variable
ci_bma_sully = confint(ascore_prediction, estimator="BMA")
ci_bma_sully
```

```
##          2.5%    97.5%     pred
## [1,] 58.15596 97.51534 77.82263
## attr(,"Probability")
## [1] 0.95
## attr(,"class")
## [1] "confint.bas"
```

The credible interval for the Audience Score from Rotten Tomatoes for the movie Sully is 58 to 97, with a predicted value of roughly 78 as obtained previously.

Part 6: Conclusion

The movies data set has been used to find out associations between `audience_score` and the other new variables created. The exploratory analysis and modeling allow to pick up the best significant predictors and select the best Bayes model with the highest posterior probabilities. Furthermore, the true Audience Score with a value of 84 is within the estimated credible interval (58 to 97).

Although the model performs reasonably well when using the new proposed variables, additional testing would be required to establish whether this model is a good fit or otherwise performed well with just one prediction.

The biggest shortcoming of the model consists in the use of some parameters as inputs into the predictor such as `imdb_num_votes` and `imdb_rating` which may lead to an extrapolation in the result. In addition, further testing with other movies would be desirable to get a more accurate model.