Bayesian modeling and prediction for movies

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
library(BAS)
library(gridExtra) # in order to arrange plots
```

Load data

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

Part 1: Data

The data set is comprised of 651 randomly sampled movies produced and released before 2016. The information contained in the variables of the data set comes from 2 databases : Rotten Tomatoes and IMDB, assumed to be the population of several thousands movies here.

Since each movie had an equal chance of being selected in the random process, the information from the randomly sampled data set can be used to build observational studies to show potential associations between the variables generalizable to the entire population of films made and released before 2016, and in a non-causal way as causality can only come from random assigned cases for experiments (which is not the case here).

** Research Question **

In this assignment we are asked to create the 5 following new categorical variables : feature_film, drama, mpaa_rating_R, oscar_season, and summer_season. Part 2 shows how they are built up. Is any of these new variable a good predictor of the audience_score value of a movie ?

Part 2: Data manipulation

```
# Create 5 new factor variables with values yes or no :
#       feature_film   == "yes" when title_type == "Feature Film", "no" otherwise
#       drama          == "yes" when genre == "Drama", "no" otherwise
#       mpaa_rating_R  == "yes" when mpaa_rating =="R", "no" otherwise
#       oscar_season   == "yes" when movie thtr_rel_month == 10 or 11 or 12,
#                                "no" otherwise
#       summer_season  == "yes" when movie thtr_rel_month == 5 or 6 or 7 or 8,
#                                "no" otherwise

movies <- movies %>%
    mutate(feature_film = as.factor(ifelse(title_type=="Feature Film", "yes", "no")),
           drama = as.factor(ifelse(genre=="Drama", "yes","no")),
           mpaa_rating_R = as.factor(ifelse(mpaa_rating=="R", "yes","no")),
           oscar_season = as.factor(ifelse(thtr_rel_month > 9, "yes","no")),
           summer_season = as.factor(ifelse(thtr_rel_month > 4 &
                                            thtr_rel_month < 9, "yes", "no")))
str(movies)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    651 obs. of  37 variables:
##  $ title           : chr  "Filly Brown" "The Dish" "Waiting for Guffman" "The Age of
 Innocence" ...
##  $ title_type      : Factor w/ 3 levels "Documentary",..: 2 2 2 2 2 1 2 2 1 2 ...
##  $ genre           : Factor w/ 11 levels "Action & Adventure",..: 6 6 4 6 7 5 6 6 5
 6 ...
##  $ runtime         : num  80 101 84 139 90 78 142 93 88 119 ...
##  $ mpaa_rating     : Factor w/ 6 levels "G","NC-17","PG",..: 5 4 5 3 5 6 4 5 6 6 ...
##  $ studio          : Factor w/ 211 levels "20th Century Fox",..: 91 202 167 34 13 16
3 147 118 88 84 ...
##  $ thtr_rel_year   : num  2013 2001 1996 1993 2004 ...
##  $ thtr_rel_month  : num  4 3 8 10 9 1 1 11 9 3 ...
##  $ thtr_rel_day    : num  19 14 21 1 10 15 1 8 7 2 ...
##  $ dvd_rel_year    : num  2013 2001 2001 2001 2005 ...
##  $ dvd_rel_month   : num  7 8 8 11 4 4 2 3 1 8 ...
##  $ dvd_rel_day     : num  30 28 21 6 19 20 18 2 21 14 ...
##  $ imdb_rating     : num  5.5 7.3 7.6 7.2 5.1 7.8 7.2 5.5 7.5 6.6 ...
##  $ imdb_num_votes  : int  899 12285 22381 35096 2386 333 5016 2272 880 12496 ...
##  $ critics_rating  : Factor w/ 3 levels "Certified Fresh",..: 3 1 1 1 3 2 3 3 2 1
 ...
##  $ critics_score   : num  45 96 91 80 33 91 57 17 90 83 ...
##  $ audience_rating : Factor w/ 2 levels "Spilled","Upright": 2 2 2 2 1 2 2 1 2 2 ...
##  $ audience_score  : num  73 81 91 76 27 86 76 47 89 66 ...
##  $ best_pic_nom    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_pic_win    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_actor_win  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...
##  $ best_actress_win: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_dir_win    : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ top200_box      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ director        : chr  "Michael D. Olmos" "Rob Sitch" "Christopher Guest" "Martin
 Scorsese" ...
##  $ actor1          : chr  "Gina Rodriguez" "Sam Neill" "Christopher Guest" "Daniel D
ay-Lewis" ...
##  $ actor2          : chr  "Jenni Rivera" "Kevin Harrington" "Catherine O'Hara" "Mich
elle Pfeiffer" ...
##  $ actor3          : chr  "Lou Diamond Phillips" "Patrick Warburton" "Parker Posey"
 "Winona Ryder" ...
##  $ actor4          : chr  "Emilio Rivera" "Tom Long" "Eugene Levy" "Richard E. Gran
t" ...
##  $ actor5          : chr  "Joseph Julian Soria" "Genevieve Mooy" "Bob Balaban" "Alec
 McCowen" ...
##  $ imdb_url        : chr  "http://www.imdb.com/title/tt1869425/" "http://www.imdb.co
m/title/tt0205873/" "http://www.imdb.com/title/tt0118111/" "http://www.imdb.com/title/t
t0106226/" ...
##  $ rt_url          : chr  "//www.rottentomatoes.com/m/filly_brown_2012/" "//www.rott
entomatoes.com/m/dish/" "//www.rottentomatoes.com/m/waiting_for_guffman/" "//www.rotten
tomatoes.com/m/age_of_innocence/" ...
##  $ feature_film    : Factor w/ 2 levels "no","yes": 2 2 2 2 2 1 2 2 1 2 ...
##  $ drama           : Factor w/ 2 levels "no","yes": 2 2 1 2 1 1 2 2 1 2 ...
##  $ mpaa_rating_R   : Factor w/ 2 levels "no","yes": 2 1 2 1 2 1 1 2 1 1 ...
##  $ oscar_season    : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...
##  $ summer_season   : Factor w/ 2 levels "no","yes": 1 1 2 1 1 1 1 1 1 1 ...
```

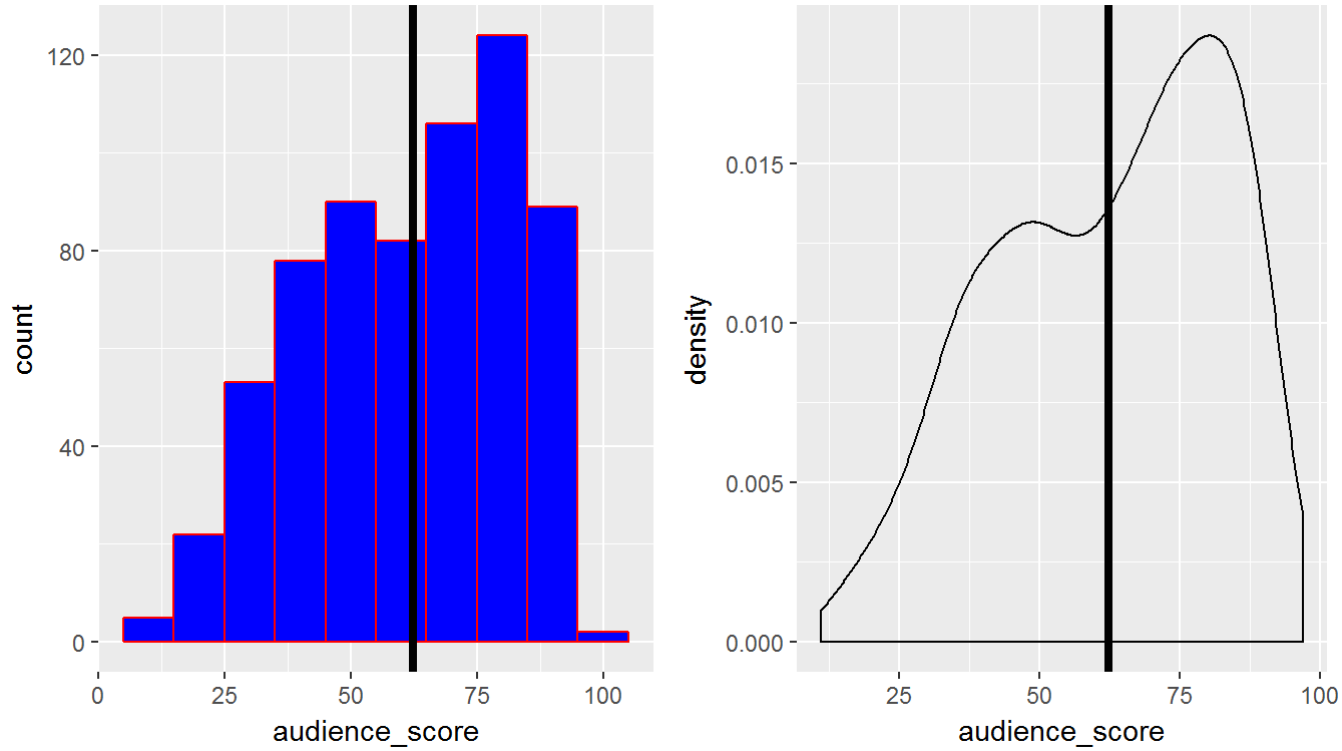
Part 3: Exploratory data analysis

Summary Statistics of audience_score and its distribution

```
summary(movies$audience_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   46.00   65.00   62.36   80.00   97.00
```

```
x_mean <- mean(movies$audience_score)

pl1 <- ggplot(movies, aes(x=audience_score))+
        geom_histogram(binwidth=10, colour="red",
                       fill="blue")+geom_vline(xintercept=x_mean, size=1.5)
pl2 <- ggplot(movies,aes(x=audience_score))+geom_density()+
        geom_vline(xintercept=x_mean, size=1.5)
grid.arrange(pl1,pl2, ncol=2)
```



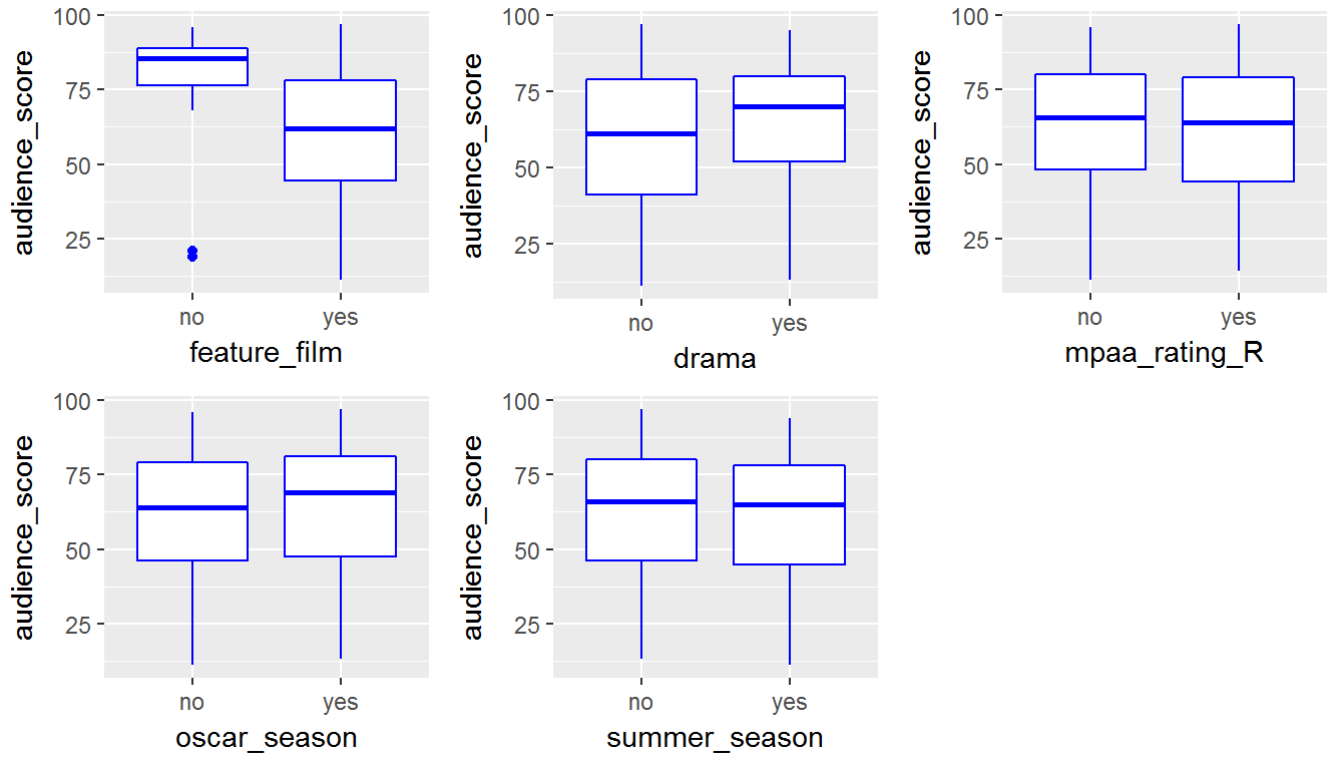
The distribution of **audience_score** is slightly left skewed with the median score at 65 and the mean score at 62.36. The mean is shown as the vertical black line on the X axis in the 2 plots above.

———

Correlations between audience_score and the 5 new created variables

feature_film, drama, mpaa_rating_R, oscar_season, and **summer_season**

```
pl1 <- ggplot(movies, aes(x=feature_film, y=audience_score))+
            geom_boxplot(colour="blue")
pl2 <- ggplot(movies, aes(x=drama, y=audience_score))+
            geom_boxplot(colour="blue")
pl3 <- ggplot(movies, aes(x=mpaa_rating_R,y=audience_score))+
            geom_boxplot(colour="blue")
pl4 <- ggplot(movies, aes(x=oscar_season, y=audience_score))+
            geom_boxplot(colour="blue")
pl5 <- ggplot(movies, aes(x=summer_season,y=audience_score))+
            geom_boxplot(colour="blue")
grid.arrange(pl1,pl2,pl3,pl4,pl5, ncol=3)
```



Correlations :

audience_score and feature_film are negatively correlated (higher scores for non-feature-films)

audience_score and drama are very slightly positively correlated, weak association

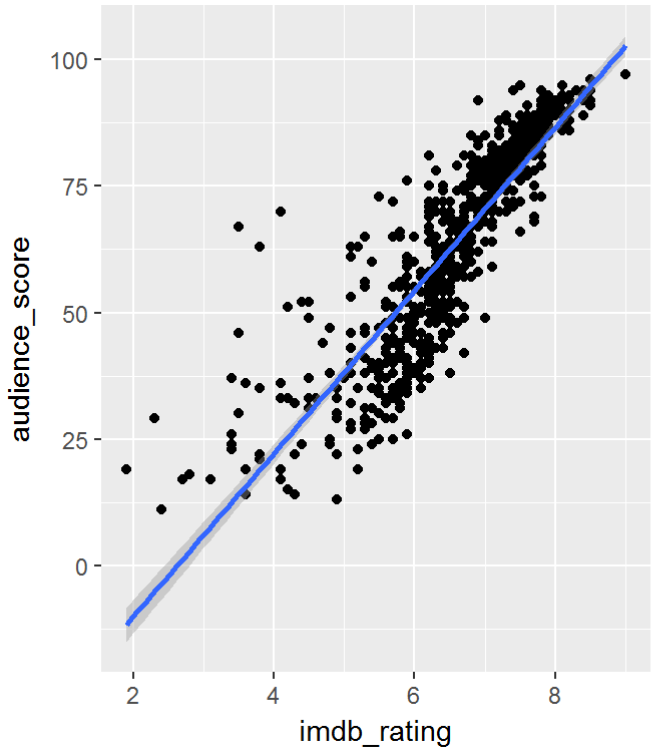
audience_score and mpaa_rating_R have are slightly correlated

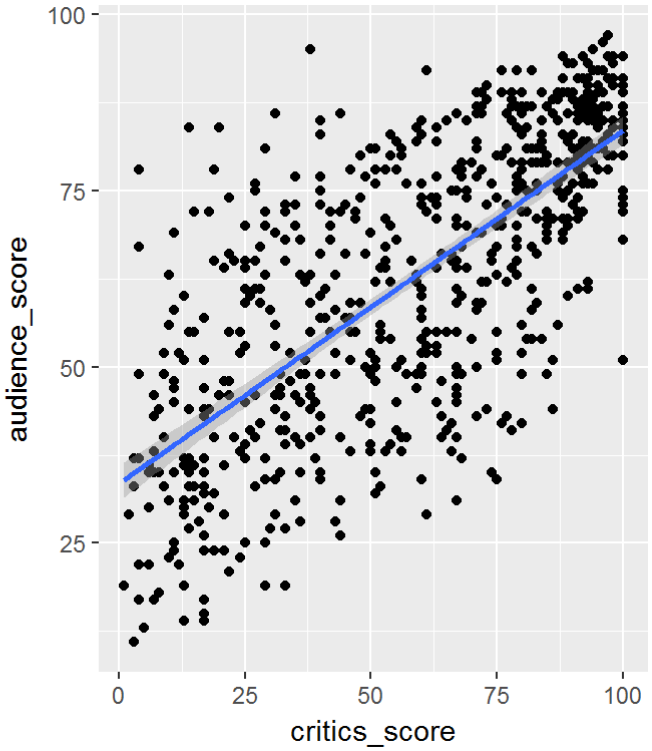
audience_score and oscar_season are very slightly positively correlated, although a pretty weak association

audience_score and summer_season are also weakly correlated

CORRELATIONS between audience_score and imdb_rating and critics_score

```
pl1 <- ggplot(movies, aes(x=imdb_rating, y=audience_score))+ geom_point()+
              geom_smooth(method=lm)
pl2 <- ggplot(movies, aes(x=critics_score, y=audience_score))+ geom_point()+
              geom_smooth(method=lm)

grid.arrange(pl1,pl2, ncol=2)
```






```
print(paste("corr imdb_rating and audience_score :", round(cor(movies$imdb_rating,
            movies$audience_score),2), "  ", "corr critics_score and audience_score: ",
            round(cor(movies$critics_score, movies$audience_score),2) ))
```

```
## [1] "corr imdb_rating and audience_score : 0.86    corr critics_score and audience_s
core:  0.7"
```

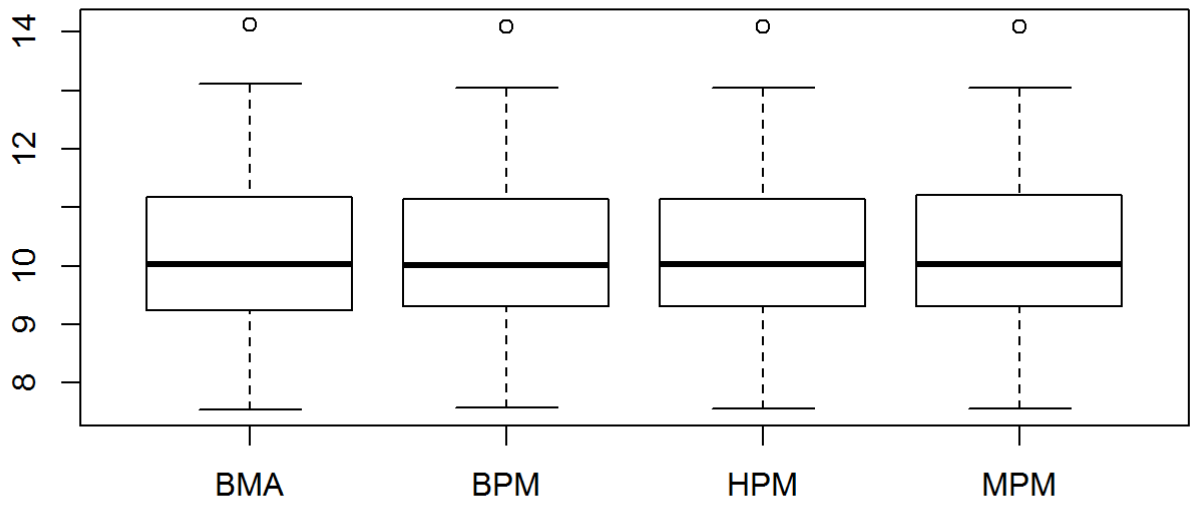
Very strong positive correlations between imdb_rating and audience_score, strong positive correlation between critics_score and audience_score.

Part 4: Modeling

With the BAS package I will use two runs of 50 random training and test sets from the movie sample to find the estimator that has the lowest prediction error. Estimators are Bayasian Model Averaging (BMA) , Best Predictor Model (BPM), Highest Probability Model (HPM), and Median Probability Model (MPM). A 90%-10% split will be used for training and test sets. First run will use “BIC” as the prior distribution for regression coefficients while the second run will use “ZS-null”. Both runs will use the uniform model as the model prior. Will choose the model with the lowest prediction error on the test sets.

```r
# Will use the reduced data set "mov"
mov <- movies %>%
    select( -title, -title_type, -genre, -mpaa_rating, -studio, -thtr_rel_month,
           -thtr_rel_day, -dvd_rel_year, -dvd_rel_month, -dvd_rel_day,
           -critics_rating, -audience_rating, -director, -actor1, -actor2,
           -actor3, -actor4, - actor5, -imdb_url, -rt_url)



mov_no_na = na.omit(mov) # Omit NAs
```

```r
set.seed(25)
n = nrow(mov_no_na)
n_cv = 50
ape = matrix(NA, ncol=4, nrow=n_cv)
colnames(ape) = c("BMA", "BPM", "HPM", "MPM")

for (i in 1:n_cv) {
  train = sample(1:n, size=round(.90*n), replace=FALSE)
  score_train = mov_no_na[train,]
  score_test = mov_no_na[-train,]

  bma_train_score = bas.lm(audience_score ~ ., data=score_train,
                           prior="BIC", modelprior=uniform(), initprobs="eplogp")
  yhat_bma = predict(bma_train_score, score_test, estimator="BMA")$fit
  yhat_hpm = predict(bma_train_score, score_test, estimator="HPM")$fit
  yhat_mpm = predict(bma_train_score, score_test, estimator="MPM")$fit
  yhat_bpm = predict(bma_train_score, score_test, estimator="BPM")$fit
  ape[i, "BMA"] = cv.summary.bas(yhat_bma, score_test$audience_score)
  ape[i, "BPM"] = cv.summary.bas(yhat_bpm, score_test$audience_score)
  ape[i, "HPM"] = cv.summary.bas(yhat_hpm, score_test$audience_score)
  ape[i, "MPM"] = cv.summary.bas(yhat_mpm, score_test$audience_score)
}

# Show the 4 estimators with their test error

boxplot(ape)
```

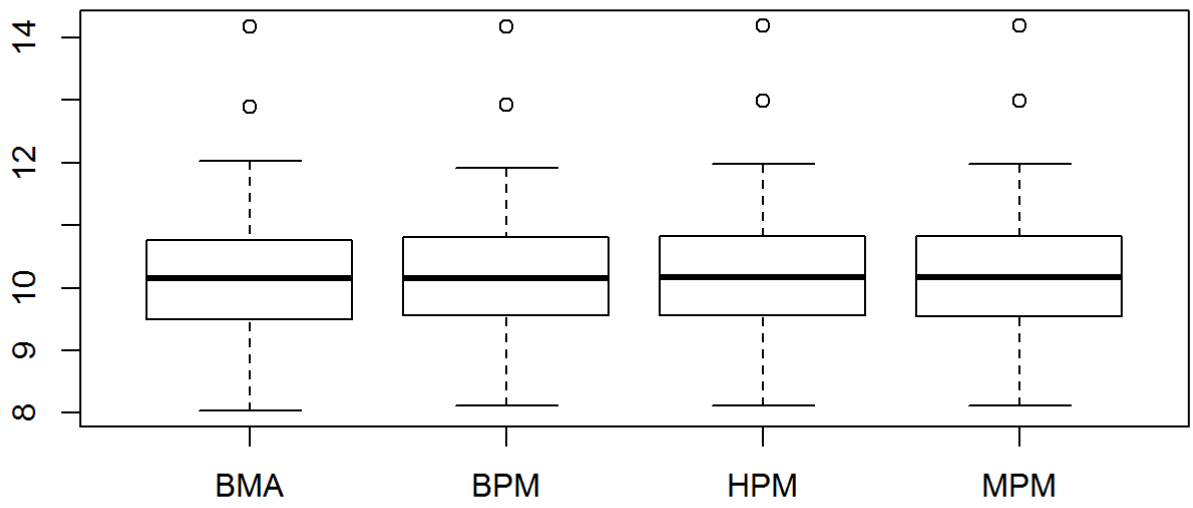



```
apply(ape, 2, mean)
```

```
##       BMA       BPM       HPM       MPM
## 10.23754 10.24519 10.25996 10.26574
```

THe BMA estimator shows the lowest test error

```
# Same as above but With prior ditribution for regression coefficients set to ZS-null i
nstead of BIC

for (i in 1:n_cv) {
  train = sample(1:n, size=round(.90*n), replace=FALSE)
  score_train = mov_no_na[train,]
  score_test = mov_no_na[-train,]

  bma_train_score = bas.lm(audience_score ~ ., data=score_train,
                           prior="ZS-null", modelprior=uniform(), initprobs="eplogp")
  yhat_bma = predict(bma_train_score, score_test, estimator="BMA")$fit
  yhat_hpm = predict(bma_train_score, score_test, estimator="HPM")$fit
  yhat_mpm = predict(bma_train_score, score_test, estimator="MPM")$fit
  yhat_bpm = predict(bma_train_score, score_test, estimator="BPM")$fit
  ape[i, "BMA"] = cv.summary.bas(yhat_bma, score_test$audience_score)
  ape[i, "BPM"] = cv.summary.bas(yhat_bpm, score_test$audience_score)
  ape[i, "HPM"] = cv.summary.bas(yhat_hpm, score_test$audience_score)
  ape[i, "MPM"] = cv.summary.bas(yhat_mpm, score_test$audience_score)
}

# Show the 4 estimators with their test error

boxplot(ape)
```




```
apply(ape, 2, mean)
```

```
##      BMA      BPM      HPM      MPM
## 10.20722 10.23360 10.24381 10.23627
```

Here again the BMA estimator shows the lowest test error. The BMA test error is lower than the BMA test error with the BIC prior.

Hence I will use the model with the ZS-null prior to make predictions.

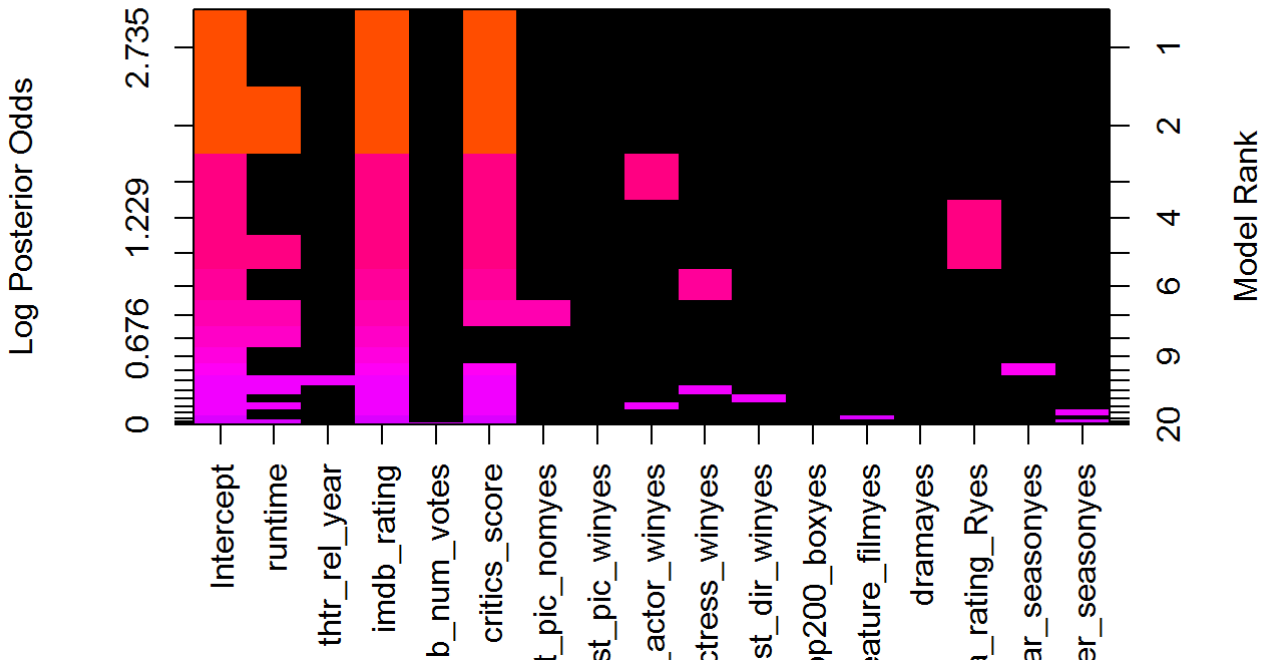
```
# with the full sample set as the training set
model_p <- bas.lm(audience_score ~ ., data=mov_no_na,
                          prior="ZS-null", modelprior=uniform(), initprobs="eplogp")

summary(model_p)
```

```
##      Intercept runtime thtr_rel_year imdb_rating imdb_num_votes
## [1,]         1       0             0           1              0
## [2,]         1       1             0           1              0
## [3,]         1       0             0           1              0
## [4,]         1       0             0           1              0
## [5,]         1       1             0           1              0
##      critics_score best_pic_nomyes best_pic_winyes best_actor_winyes
## [1,]             1               0               0                 0
## [2,]             1               0               0                 0
## [3,]             1               0               0                 1
## [4,]             1               0               0                 0
## [5,]             1               0               0                 0
##      best_actress_winyes best_dir_winyes top200_boxyes feature_filmyes
## [1,]                   0               0             0               0
## [2,]                   0               0             0               0
## [3,]                   0               0             0               0
## [4,]                   0               0             0               0
## [5,]                   0               0             0               0
##      dramayes mpaa_rating_Ryes oscar_seasonyes summer_seasonyes        BF
## [1,]        0                0               0                 0 1.0000000
## [2,]        0                0               0                 0 0.8702806
## [3,]        0                0               0                 0 0.2236679
## [4,]        0                1               0                 0 0.2217602
## [5,]        0                1               0                 0 0.2055844
##      PostProbs     R2 dim  logmarg
## [1,]    0.1388 0.7525   3 443.9495
## [2,]    0.1208 0.7549   4 443.8106
## [3,]    0.0311 0.7539   4 442.4519
## [4,]    0.0308 0.7539   4 442.4433
## [5,]    0.0285 0.7563   5 442.3676
```

There were 65536 (2^16) models analyzed in BAS as we have 16 predictors , the model with the highest posterior probability 0.1388 has 2 predictors only : **imdb_rating** and **critics_score** and the second best at 0.1208 has a 3rd predictor **runtime**

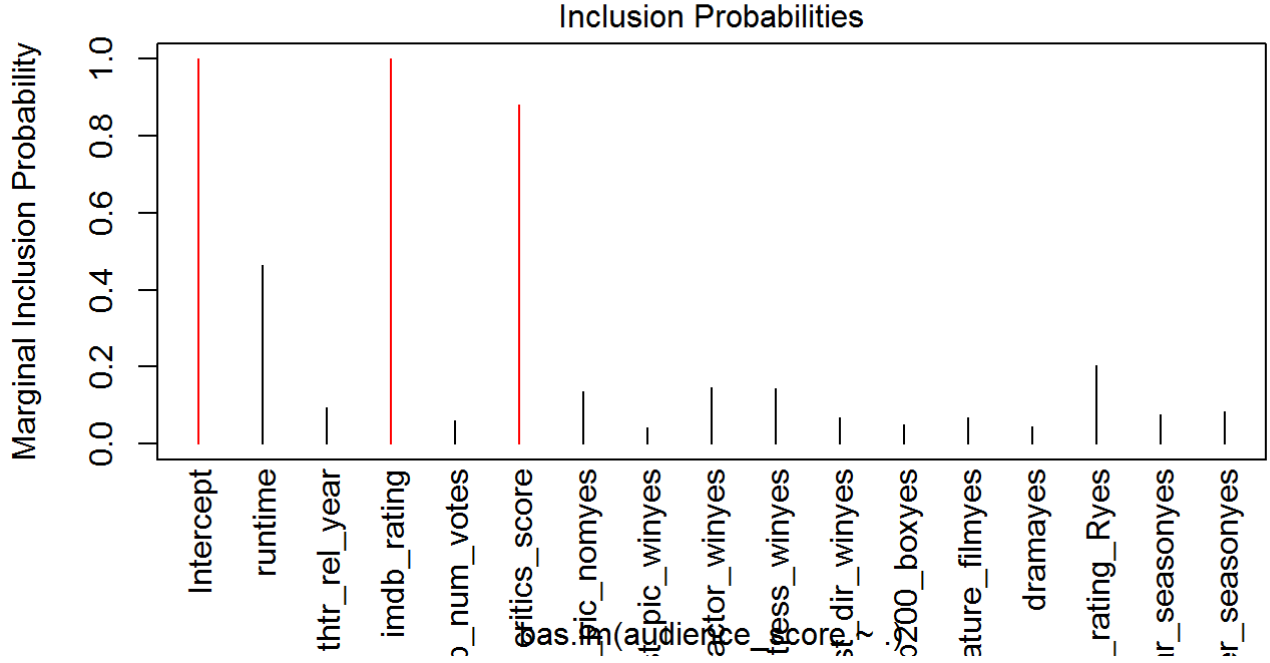
Best Models :

```
image(model_p)
```



Marginal Inclusion Probabilities of each predictor

```
plot(x=model_p, which=4)
```

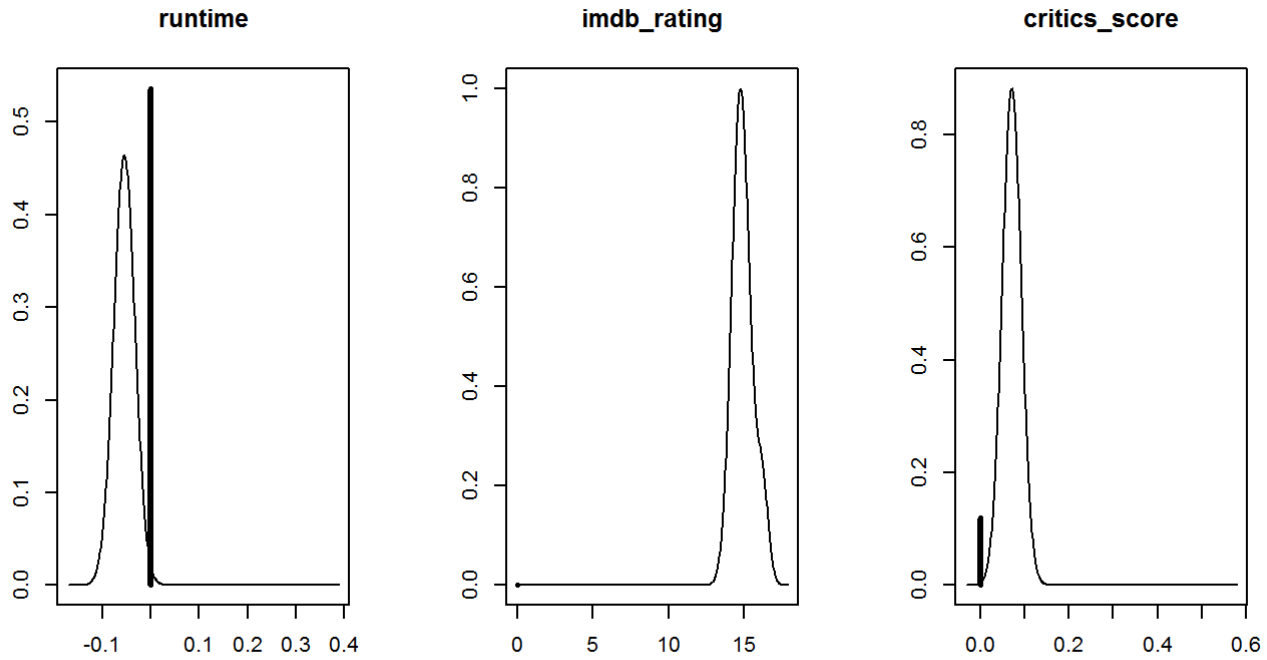


There are 3 predictors that have marginal inclusion probabilities higher than 40%, let's look at their coefficient distributions

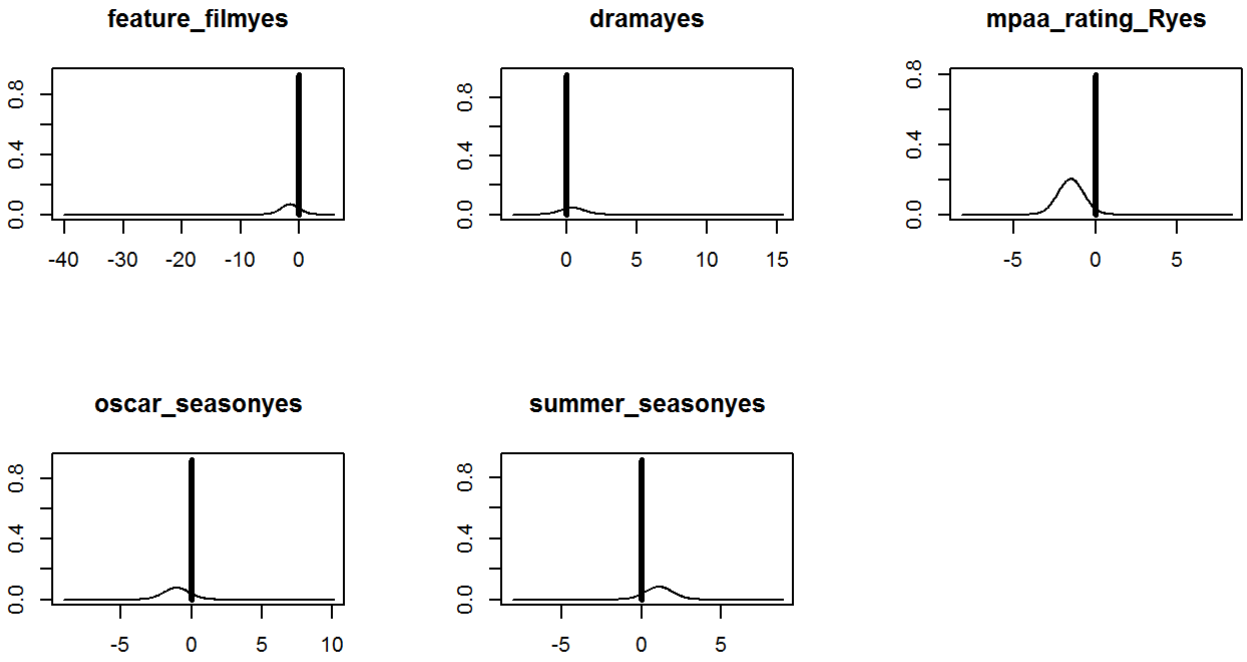
```
par(mfrow = c(1,3))
coef_model_p = coefficients(model_p)
coef_model_p
```

```
##
##  Marginal Posterior Summaries of Coefficients:
##                       post mean   post SD     post p(B != 0)
## Intercept             6.235e+01   3.946e-01   1.000e+00
## runtime              -2.535e-02   3.111e-02   4.642e-01
## thtr_rel_year        -4.772e-03   1.865e-02   9.500e-02
## imdb_rating           1.496e+01   7.370e-01   1.000e+00
## imdb_num_votes        2.242e-07   1.362e-06   6.115e-02
## critics_score         6.227e-02   3.067e-02   8.808e-01
## best_pic_nomyes       5.324e-01   1.606e+00   1.368e-01
## best_pic_winyes      -1.091e-02   8.781e-01   4.216e-02
## best_actor_winyes    -2.898e-01   8.331e-01   1.464e-01
## best_actress_winyes  -3.146e-01   9.122e-01   1.444e-01
## best_dir_winyes      -1.227e-01   6.305e-01   6.936e-02
## top200_boxyes         9.040e-02   7.208e-01   4.999e-02
## feature_filmyes      -1.081e-01   5.739e-01   6.797e-02
## dramayes              1.791e-02   2.017e-01   4.592e-02
## mpaa_rating_Ryes     -3.073e-01   7.060e-01   2.027e-01
## oscar_seasonyes      -8.221e-02   3.814e-01   7.750e-02
## summer_seasonyes      8.984e-02   3.887e-01   8.336e-02
```

```
plot(coef_model_p, subset=c(2,4,6), ask=FALSE)
```




```
par(mfrow = c(2,3))
plot(coef_model_p, subset=c(13,14,15,16,17), ask=FALSE)
```



imdb_rating has 0% probability of having a zero coefficient, critics_score has a 12% probability of having a zero coefficient while runtime has a 53.6% probability of having a zero coefficient. The 3 have larger posterior mean.

The 5 created categorical variables have very prob. of having zero coefficients, as shown in the plots. Of these 5 variables, mpaa_rating_R is the most significant predictor but far behind **imdb_rating** the most significant predictor, **critics_score** and **runtime** a distant third, after learning from the sample data.

Part 5: Prediction

Selected variables from the highest posterior probabilities for the model are **imdb_rating** and **critics_score**

The movie I picked is **SNOWDEN (2016)**

from the IMDB and Rotten Tomatoes web sites we get :

True audience_score is 73%, imdb_rating is 7.4, critics_score is 53%, runtime 134 minutes, Rated R, Drama

```
Snowden <- mov_no_na[1,]   # initialize Snowden row to the values of the
#                            first data set row and overwrite the following variables
#                            with the correct values

# From Rotten Tomatoes and IMDB we get
Snowden$imdb_rating <- 7.4  # insert imdb_rating value for Snowden movie
Snowden$critics_score <- 53 # value for critics_score
Snowden$runtime <- 134      # runtime in minutes
Snowden$thtr_rel_year <- 2016  # release year

print(paste("True Score of Snowden on Rotten Tomatoes is 73"))
```

```
## [1] "True Score of Snowden on Rotten Tomatoes is 73"
```

```
yhat_bma = predict(model_p, Snowden, estimator="BMA")$fit
print(paste("Predicted Audience Score BMA for Snowden is ", round(yhat_bma,1)))
```

```
## [1] "Predicted Audience Score BMA for Snowden is  74.7"
```

Let's have a look of the same prediction with estimator HPM and MPM (for fun!)

```
yhat_hpm = predict(model_p, Snowden, estimator="HPM")$fit
print(paste("Predicted Audience Score  HPM Snowden is ", round(yhat_hpm,1)))
```

```
## [1] "Predicted Audience Score  HPM Snowden is  75.3"
```

```
yhat_mpm = predict(model_p, Snowden, estimator="MPM")$fit
print(paste("Predicted Audience Score MPM for Snowden is ", round(yhat_mpm,1)))
```

```
## [1] "Predicted Audience Score MPM for Snowden is  75.3"
```

Part 6: Conclusion

In this assignment we were asked to construct 5 categorical variables from existing variables in the sample dataset and explore their relationships with the response variable. (see part 2 and 3). After conducting a Bayesian Model Averaging analysis on 16 selected predictors, their marginal probability inclusions were much lower than the probability model inclusions of 2 strongly associated continuous variables with the response variable audience_score, namely imdb_rating and critics_score. The best bayesian models excluded the 5 created categorical variables, therefore the response to the research question of part 1 (Data) is a clear no.