Bayesian modeling and prediction for movies

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
library(BAS)
library(tidyr)
```

Load data

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

Part 1: Data

The dataset under analysis includes 651 movies reviewee on two platforms, Rotten Tomatoes and Internet Movies Database (IMBD). The following analysis is aimed at finding attributes that make a movie popular. As a data scientist of Paramount picture, I am conducting a study that will be useful in order to make project regarding production of new movies. The dataset contains information about movies released from 1970 to 2014. This will be an observational study due to the nature of the data (i.e. no random assignment). The study can be generalized to movies issued between 1970 and 2014. Because the dataset is based only on reviews via rotten tomatoes and Internet Movie Database (IMDB), it might be biased, because we are considering audience rating from only two sources.

Part 2: Data manipulation

I will create new variables from the dataset, specifically feature_film: "yes" if title_type is Feature Film, "no" otherwise drama: "yes" if genre is Drama, "no" otherwise runtime mpaa_rating_R: "yes" if mpaa_rating is R, "no" otherwise thtr_rel_year oscar_season: "yes" if movie is released in November, October, or December (based on thtr_rel_month), "no" otherwise summer_season: "yes" if movie is released in May, June, July, or August (based on thtr_rel_month), "no" otherwise

```
# feature_film: "yes" if title_type is Feature Film, "no" otherwise
movies <- movies %>%
mutate(feature_film = as.factor(ifelse(title_type == 'Feature Film',"yes","no")))%>%
# drama: "yes" if genre is Drama, "no" otherwise
mutate(drama = as.factor(ifelse(genre == 'Drama',"yes","no")))%>%

# mpaa_rating_R: "yes" if mpaa_rating is R, "no" otherwise
mutate(mpaa_rating_R = as.factor(ifelse(mpaa_rating == 'R',"yes","no")))%>%

# oscar_season: "yes" if movie is released in November, October, or December (based o
n thtr_rel_month), "no" otherwise
mutate(oscar_season =as.factor(ifelse(thtr_rel_month == 10, "yes",
                       ifelse(thtr_rel_month == 11, "yes",
                       ifelse(thtr_rel_month == 12, "yes", "no")))))

# summer_season: "yes" if movie is released in May, June, July, or August (based on t
htr_rel_month), "no" otherwise
movies <- movies %>% mutate(summer_season = as.factor(ifelse(movies$thtr_rel_month ==
5, "yes",
                                                     ifelse(movies$thtr_rel_month ==
6, "yes",
                                                     ifelse(movies$thtr_rel_month ==
7, "yes",
                                                     ifelse(movies$thtr_rel_month ==
8, "yes", "no"))))))
```

Part 3: Exploratory data analysis

I create a subset of the data including only the variables that I will use in the analysis.

```
movies<-movies%>% select(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,audience_score)

summary(movies)
```

```
##  feature_film drama        runtime      mpaa_rating_R thtr_rel_year
##  no : 60      no :346   Min.   : 39.0   no :322       Min.   :1970
##  yes:591      yes:305   1st Qu.: 92.0   yes:329       1st Qu.:1990
##                         Median :103.0                 Median :2000
##                         Mean   :105.8                 Mean   :1998
##                         3rd Qu.:115.8                 3rd Qu.:2007
##                         Max.   :267.0                 Max.   :2014
##                         NA's   :1
##  oscar_season summer_season  imdb_rating    imdb_num_votes
##  no :460      no :443       Min.   :1.900   Min.   :   180
##  yes:191      yes:208       1st Qu.:5.900   1st Qu.:  4546
##                             Median :6.600   Median : 15116
##                             Mean   :6.493   Mean   : 57533
##                             3rd Qu.:7.300   3rd Qu.: 58301
##                             Max.   :9.000   Max.   :893008
##
##  critics_score    best_pic_nom best_pic_win best_actor_win
##  Min.   :  1.00   no :629      no :644      no :558
##  1st Qu.: 33.00   yes: 22      yes:  7      yes: 93
##  Median : 61.00
##  Mean   : 57.69
##  3rd Qu.: 83.00
##  Max.   :100.00
##
##  best_actress_win best_dir_win top200_box audience_score
##  no :579          no :608      no :636    Min.   :11.00
##  yes: 72          yes: 43      yes: 15    1st Qu.:46.00
##                                           Median :65.00
##                                           Mean   :62.36
##                                           3rd Qu.:80.00
##                                           Max.   :97.00
##
```

Runtime has 1 NA value, and we will remove it.

```
movies<-filter(movies, !is.na(runtime))
```

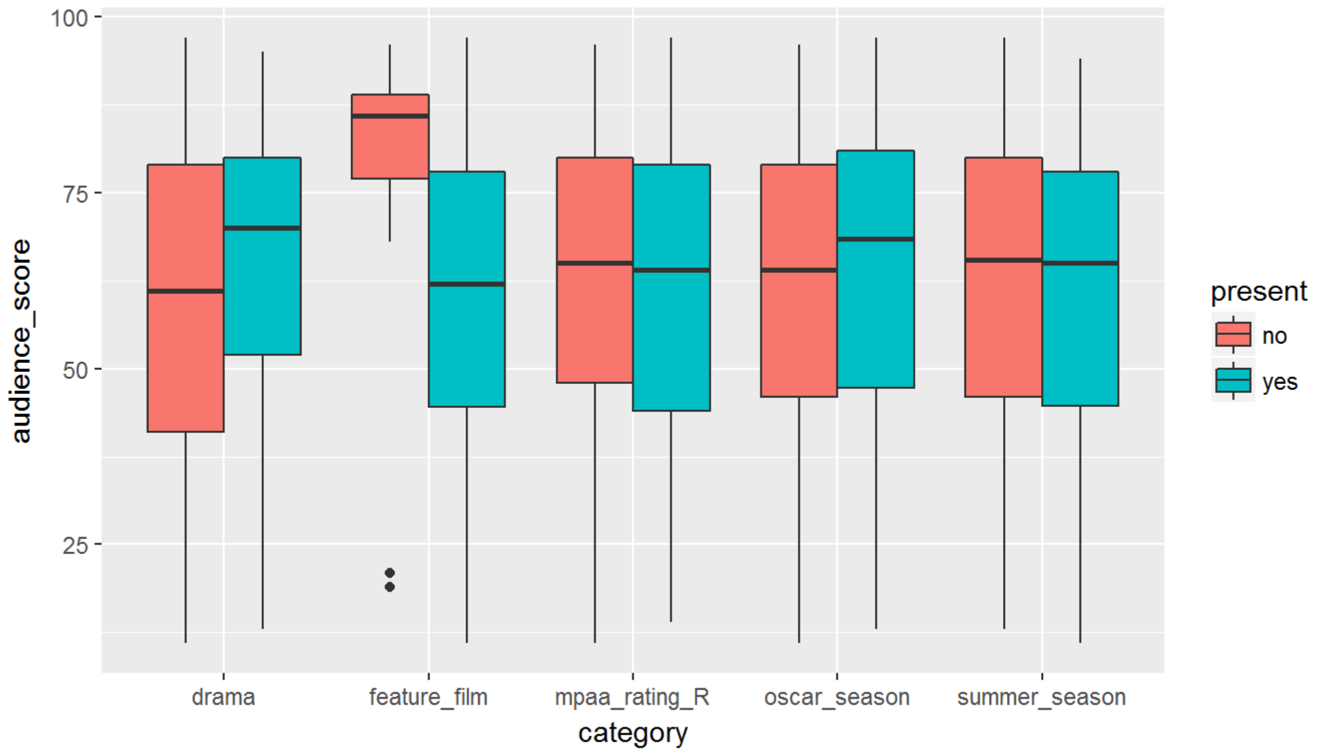
From the summary statistics we can see that:

1.Most of the movies are feature films and half of the movies are drama.

2. The mean runtime is 105.8 minutes and the median is approximately the same.
3. About a quarter of the movies were released in the three months of the Oscar season and 25% released in the 3 summer months.
4. Only 22 movies were nominated for the best picture and only 7 won the award. Also only 93 movies star a best actor Oscar winner and 72 a best actress Oscar winner. Only 43 movies were directed by a best director Oscar winner. This low percentage is predictable, as this award is very prestigious and we expect only few to win it.

Below a side by side boxplot, where we can visualize the same results as above for the categorical variables created.

```
moviesplot <- movies %>% gather('category','present',feature_film, drama,mpaa_rating_
R, oscar_season,summer_season)
ggplot(data=moviesplot,aes(x=category, y= audience_score,fill=present))+geom_boxplot
()
```



The audience score does not seem to vary much according t whether the movie is in one of the above categories or not. There is only one exception which is movies which are not feature film. In this dataset non feature films are documentaries and TV movies. These categories have higher audience scores in this dataset. This might be due to the fact that they are not so popular as feature films, and therefore are mainly watched by passionate of the genre and which are therefore prone to give positive score because of their preference of the genre.

Part 4: Modeling

Model selection I use a Bayesian regression model to predict audience_score from the 16 potential explanatory variables I previously subset. I use the Zernell-Siow Cauchy prior (ZS-null) and equal prior probability to all models using the uniform function, and I use MCMC, Monte Carlo, sampling rather than enumeration because this analysis is small enough to enumerate quickly.

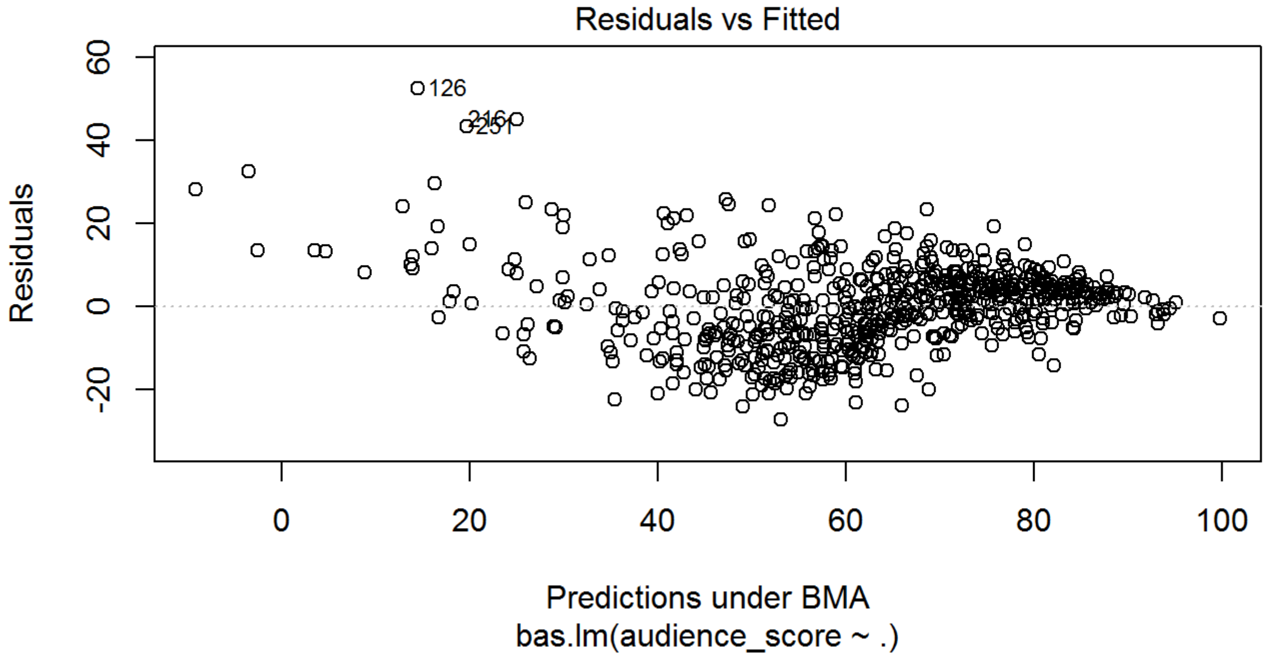
```
model<-bas.lm(audience_score~.,data=movies, prior="ZS-null", modelprior = uniform(),
method = "MCMC")
```

```
## Warning in bas.lm(audience_score ~ ., data = movies, prior = "ZS-null", :
## We recommend using the implementation using the Jeffreys-Zellner-Siow prior
## (prior='JZS') which uses numerical integration rahter than the Laplace
## approximation
```

MODEL DIAGNOSTICS

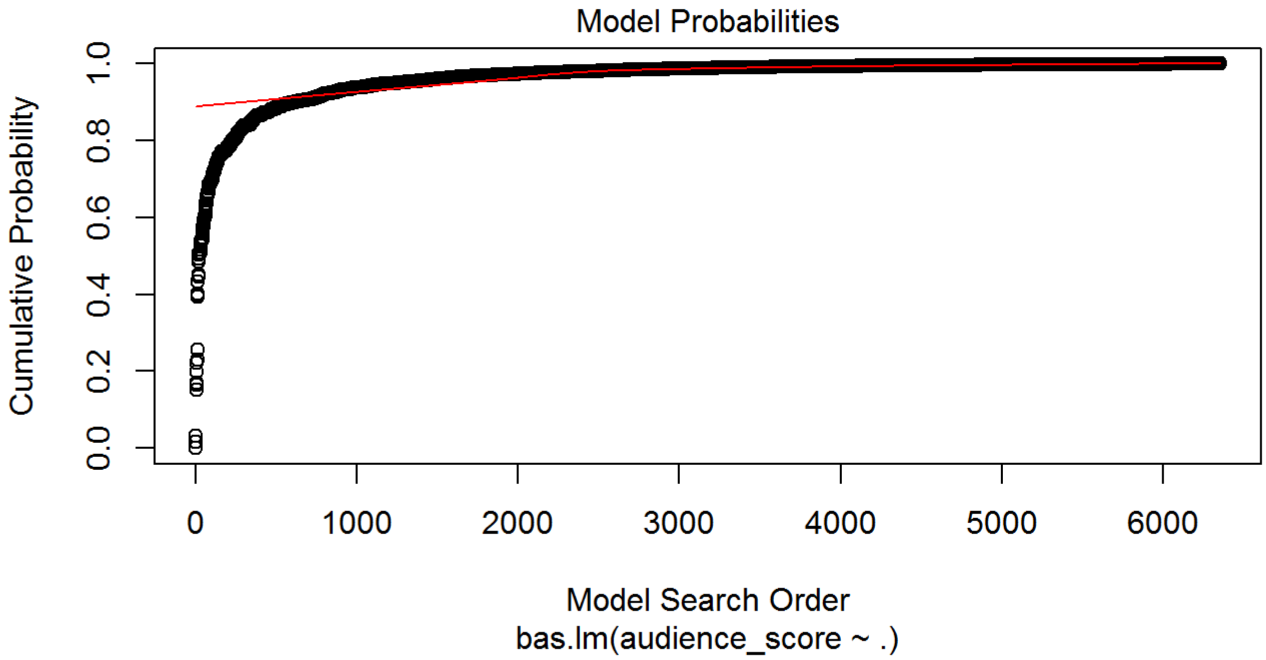
```r
#1.residuals vs fitted

plot(model, which=1, add.smooth=F)
```

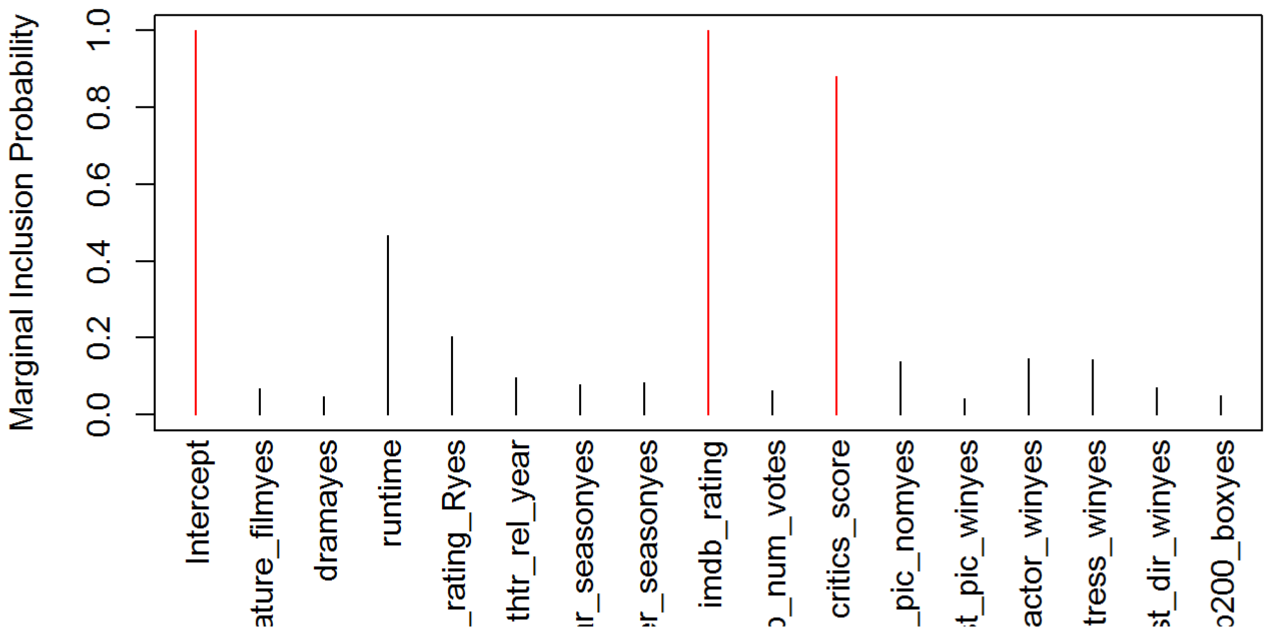



```r
#2. cumulative probability
plot(model, which=2)

#3. model dimension plot
plot(model, which=2)
```

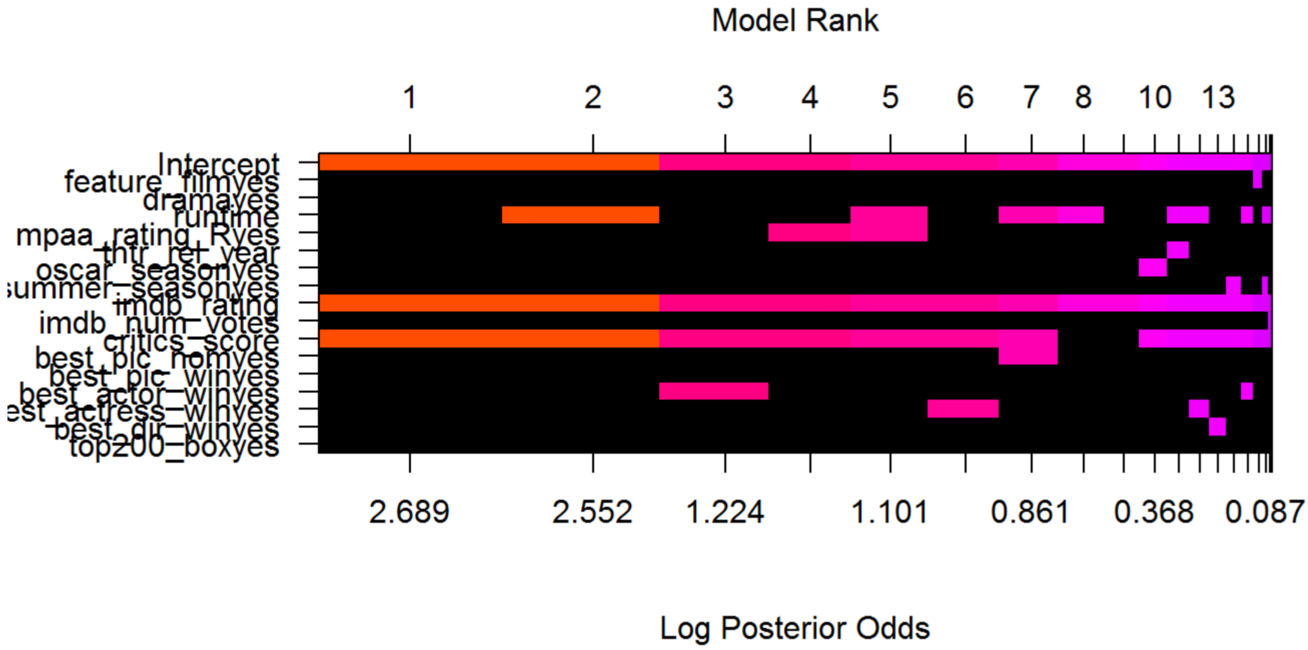


```
#4.PIP
plot(model, which = 4, ask=FALSE, caption="", sub.caption="")
```



1. From the plot of the residuals vs fitted values, we see that the spread is not constant over the whole range of the fitted values; specifically in this case low prediction scores (below 30) might not be accurately predicted because in this range we see residuals with larger values.
2. The cumulative probability plot, which adds up model probabilities each time a model is sampled, we discovered about 6000 unique models with our sampling method, MCMC, after this number the probabilities level off to 1, meaning that additional models will not add additional probability.
3. The model dimension plot shows the model size versus the log of the marginal likelihood, comparing each model to the null model with the Bayes factor. The models with highest Bayes factor have around 3 or 4 predictors, even if the difference is minimal from 3 to 11 predictors.
4. The marginal inclusion probability plot shows the importance of each predictor, the lines in red correspond to the variables with marginal posterior inclusion probability greater than 0.5, therefore these variables are important for prediction audience_score. This means that imdb_rating and critics_score are important predictors in my model.

```
#model rank
image(model, rotate = F)
```



The image of the model space show that, includes imdb_rating and critics_score are actually present in almost all the top 20 highest probability model. The plot also does not show any strong correlation between the dependent variables.

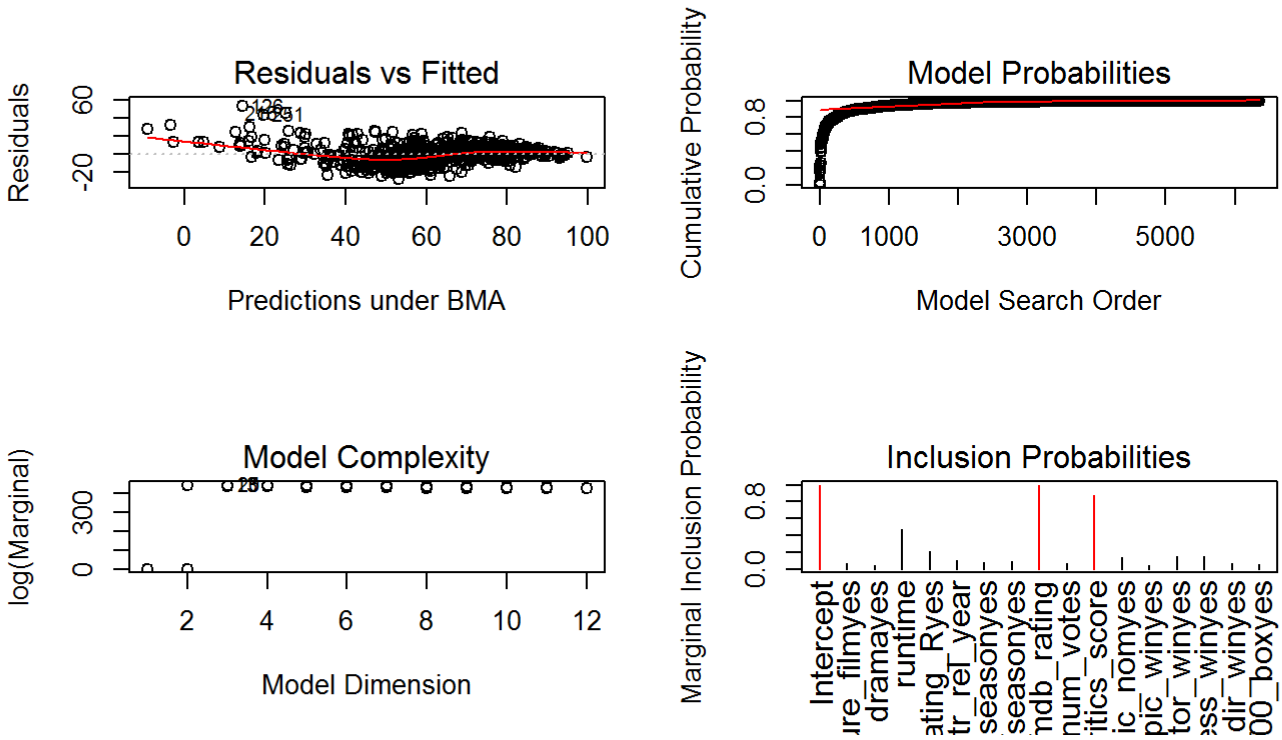
```
summary(model)
```

```
##                     P(B != 0 | Y)  model 1     model 2     model 3
## Intercept              1.00000000   1.0000   1.0000000   1.0000000
## feature_filmyes        0.06810379   0.0000   0.0000000   0.0000000
## dramayes               0.04756851   0.0000   0.0000000   0.0000000
## runtime                0.46603088   0.0000   1.0000000   0.0000000
## mpaa_rating_Ryes       0.20412979   0.0000   0.0000000   0.0000000
## thtr_rel_year          0.09790192   0.0000   0.0000000   0.0000000
## oscar_seasonyes        0.07863770   0.0000   0.0000000   0.0000000
## summer_seasonyes       0.08441620   0.0000   0.0000000   0.0000000
## imdb_rating            0.99999924   1.0000   1.0000000   1.0000000
## imdb_num_votes         0.06274872   0.0000   0.0000000   0.0000000
## critics_score          0.88111191   1.0000   1.0000000   1.0000000
## best_pic_nomyes        0.13757324   0.0000   0.0000000   0.0000000
## best_pic_winyes        0.04189987   0.0000   0.0000000   0.0000000
## best_actor_winyes      0.14744873   0.0000   0.0000000   1.0000000
## best_actress_winyes    0.14450684   0.0000   0.0000000   0.0000000
## best_dir_winyes        0.07081909   0.0000   0.0000000   0.0000000
## top200_boxyes          0.05024261   0.0000   0.0000000   0.0000000
## BF                             NA   1.0000   0.8702806   0.2236679
## PostProbs                      NA   0.1367   0.1192000   0.0316000
## R2                             NA   0.7525   0.7549000   0.7539000
## dim                            NA   3.0000   4.0000000   4.0000000
## logmarg                        NA 443.9495 443.8105657 442.4519125
##                         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                    0.2217602   0.2055844
## PostProbs             0.0305000   0.0279000
## R2                    0.7539000   0.7563000
## dim                   4.0000000   5.0000000
## logmarg             442.4433468 442.3676066
```

The model summary shows that the imdb_rating and critics_score have the highest posterior probability. Model 1 includes intercept that these two predictors and has Bayes factor 1 and posterior probability of 0.1367, therefore there are 13.76% chances that this is the true model. Critics_score and imdb_rating have also the highest probability that the coefficients are not zero, respectively 88% and almost 100%.

INTERPRETATION OF MODEL COEFFICIENTS

```
coef_model=coefficients(model)
par(mfrow=c(2, 2))
plot(model)
```




```
par(mfrow=c(1,1))
```

Below the representation of the plausible values for the coefficients. The spike indicated the posterior probability that the coefficient is zero. The bell shape curve represents the shape of al the possible model where the coefficient is non-zero. From the graphical representation again we see that critics_score and imdb_rating have the lowest zero coefficient probability.

Part 5: Prediction

We test the validity of the model by predicting the score of a movie released in 2016 and not present on the dataset, Arrival. Data from this movie are taken from https://www.rottentomatoes.com/m/arrival_2016 (https://www.rottentomatoes.com/m/arrival_2016) and http://www.imdb.com/title/tt2543164/?ref_=nv_sr_1 (http://www.imdb.com/title/tt2543164/?ref_=nv_sr_1). I will use the model containing the two relevant variables, critics_score and imbd_rating.

```
arrival<-data.frame(feature_film="yes",drama="yes",runtime=116,mpaa_rating_R="no",
                     thtr_rel_year=2016,oscar_season="yes",summer_season="no",imdb_ra
ting=8,imdb_num_votes=409543,critics_score=94,best_pic_nom="yes",best_pic_win="no",be
st_actor_win="yes",best_actress_win="no", best_dir_win="no",top200_box="yes")

pred = predict(model, newdata = arrival, estimator = "BPM",  se.fit=T)
pred$fit  # fitted values
```

```
## [1] 87.38605
## attr(,"model")
## [1]  0  4  5  7  8 10 16
## attr(,"best")
## [1] 939
```

This code predicts an audience _score of 87% while the true audience score on Rotten Tomatoes is 82%. This means that the model has slightly over predicted the value, however it is still very close.

Part 6: Conclusion

We have analysed a dataset predicting audience scores from IMDB and Rotten Tomatoes with a Bayesian model using Markov Chain Monte Carlo sampling, which samples models based on their posterior probabilities. From this analysis we have found that the most important explanatory variables to predict audiensce score are critics_score and imdb_rating. We have also tested the model by predicting the audience score of a movie and the result was very close to the actual score on Rotten Tomatoes, even if or model overestimate the score. This dataset has a some limits, and therefore the analysis too, in fact it is biased toward movies that have been rated on Rotten Tomatoes and IMDB websites. Further research could include a different sampling method such as interviews.