Bayesian Model Selection and Prediction

## Setup

The following analysis will take a data set of movies released in the United states and fit a Bayesian regression model to determine model popularity as measured by the audience score from the the Rotten Tomatoes website. All of the code for this project can be found here:

<https://github.com/lecroc/Bayesian-Statistics/tree/master/Project>

### Load packages

library(dplyr)  
library(ggplot2)  
library(gridExtra)  
library(GGally)  
library(statsr)  
library(BAS)  
library(knitr)

### Load data

load("movies.Rdata")  
  
d1<-as.data.frame(movies)  
  
dim(d1)

## [1] 651 32

## Part 1: Data

The data set is a randomly selected group of 651 movies produced and released between 1970 and 2014. Since this data is observational and not related to an expirement with random assignment, we can not look for causal relationships. However, since the movies were randomly selected, we can look for associations and apply them to the population of interest, movies produced and released for consumption in the US market.

There is little information on how the movies were randomly selected. One might want to check to see if the composition of the the factor varialbes in the sample is representative of the total population of movies released within the date range. There is also a potential for bias in the audience scores if we find that the populaton providing their opinions to IMDb and Rotten Tomatoes is not a good representation of the population viewing movies in the US market.

## Part 2: Data manipulation

First, I’ll select the variables I’ll need from the original data set, and use the mutate() function to create the new factors required for the model.

d2<-d1 %>%   
 select(title\_type, genre, runtime, mpaa\_rating, thtr\_rel\_year,  
 thtr\_rel\_month, 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) %>%  
 mutate(feature\_film=ifelse(title\_type=="Feature Film", "yes", "no")) %>%  
 mutate(drama=ifelse(genre=="Drama", "yes", "no")) %>%  
 mutate(mpaa\_rating\_R=ifelse(mpaa\_rating=="R", "yes", "no")) %>%  
 mutate(oscar\_season=ifelse(thtr\_rel\_month>9, "yes", "no")) %>%  
 mutate(summer\_season=ifelse(thtr\_rel\_month>4 & thtr\_rel\_month<9, "yes", "no"))

Next, I’ll select only the varialbes specified for EDA and Modelling, putting the dependant variable audience\_score first.

movdat<-d2 %>%  
 select(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)

Finally, I will check for any observations with missing data and remove any that I find. Then I’ll look at the dimensions of the final data set.

cc<-complete.cases(movdat)  
movdat<-movdat[cc,]  
na<-sum(is.na(movdat))  
na

## [1] 0

dim(movdat)

## [1] 650 17

We see that only one record is removed and that the final data set has 650 observations of 17 variables.

## Part 3: Exploratory data analysis

Now that we have our data set, let’s take a look relationships between the dependant variable (audience\_score) and the numeric predictors.

First, we’ll look at some summary statistics:

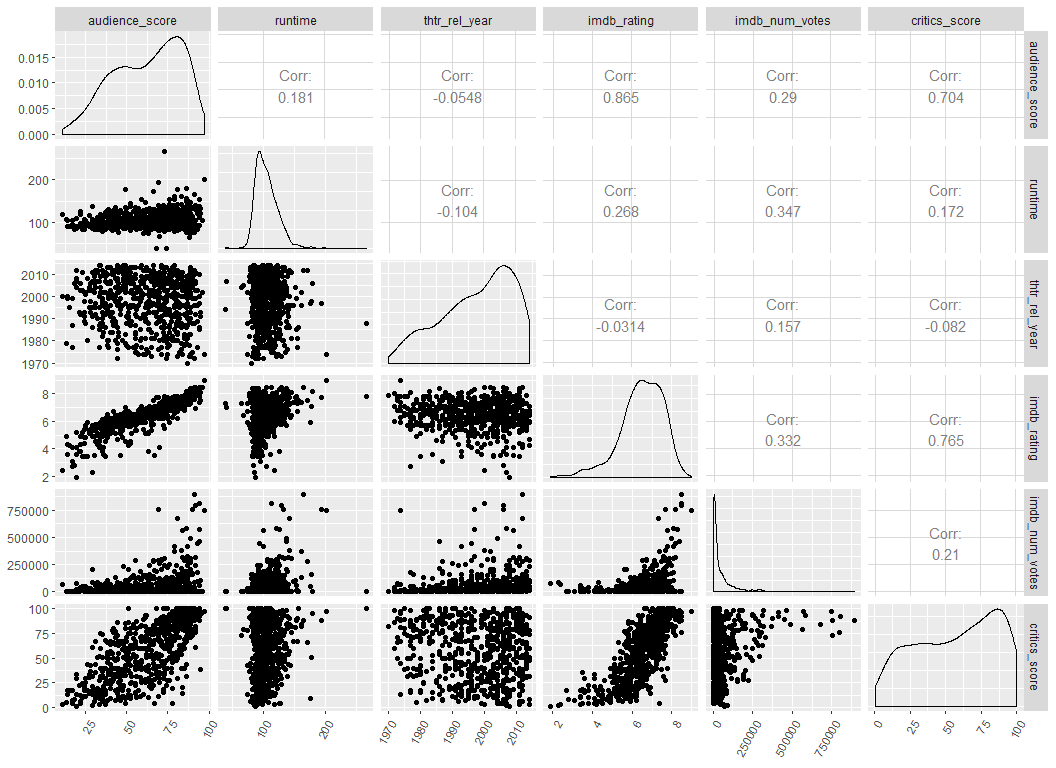
edan<-select\_if(movdat, is.numeric)  
  
s1<-summary(edan$audience\_score)  
s2<-summary(edan$runtime)  
s3<-summary(edan$thtr\_rel\_year)  
s4<-summary(edan$imdb\_rating)  
s5<-summary(edan$imdb\_num\_votes)  
s6<-summary(edan$critics\_score)  
  
stats<-names(edan)  
  
sumtable<-as.data.frame(rbind(s1,s2,s3,s4,s5,s6))  
sumtable<-as.data.frame(cbind(stats, sumtable))  
  
sumtable<-sumtable %>%  
 mutate(Skew=ifelse(Mean>Median, "Right", "Left"))  
  
kable(sumtable)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| stats | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. | Skew |
| audience\_score | 11.0 | 46.00 | 65.0 | 62.347692 | 80.00 | 97 | Left |
| runtime | 39.0 | 92.00 | 103.0 | 105.821539 | 115.75 | 267 | Right |
| thtr\_rel\_year | 1970.0 | 1990.00 | 2000.0 | 1997.926154 | 2007.00 | 2014 | Left |
| imdb\_rating | 1.9 | 5.90 | 6.6 | 6.491538 | 7.30 | 9 | Left |
| imdb\_num\_votes | 180.0 | 4584.25 | 15203.5 | 57620.358461 | 58484.25 | 893008 | Right |
| critics\_score | 1.0 | 33.00 | 61.0 | 57.653846 | 83.00 | 100 | Left |

We see from the summary statistics that the IMDb number of votes is exrremely right skewed. Most of the films are between and hour and a half and two hours, with a few much longer. The IMDb ratings and Rotten Tomatoes scores are left skewed with the ratings/scores bunching up a bit at the higher end of the range.

The following code will plot the variables from the table above in a correlation matrix. This will allow us to see how these variables correlate with the audience\_score variable, and help us look for colinearity between independent variables. It also provides a density curve which will help us look at the distributions.

p1<-ggpairs(edan)+  
 theme(legend.position = "none", axis.text.x = element\_text(angle = 60,hjust = 1))  
  
p1



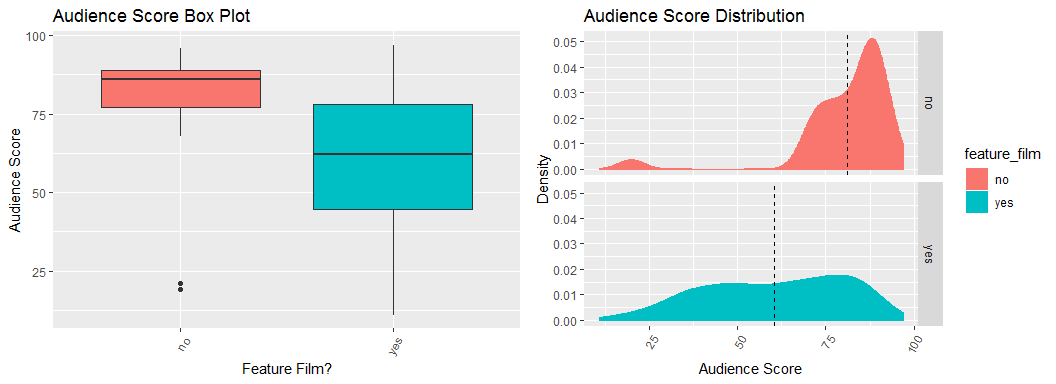
From these plots we see that the IMDb rating and critics score are both highly correlated with our dependant variable, audience score. They are also highly correlated with each other. It will be interesting to see how the bayesian model seletion process handles the colinearity between the IMDb rating and the critics score.

Now we’ll evaluate the new factor varialbes we created. We’ll use summary tables, box plots and density plots. First up is the feature\_film variable.

#### Feature Film

eda<-movdat %>%  
 select(audience\_score, feature\_film, drama,  
 mpaa\_rating\_R, oscar\_season, summer\_season)  
  
eda1<-eda %>%  
 group\_by(feature\_film) %>%  
 summarise(n(), min(audience\_score), quantile(audience\_score, .25),   
 median(audience\_score), mean(audience\_score),  
 quantile(audience\_score, .75), max(audience\_score))  
  
names(eda1)<-c("FeatureFilm", "Count", "Min", "1st Qu.", "Median", "Mean",  
 "3rd Qu.", "Max")  
eda1<-mutate(eda1, Skew=ifelse(Mean>Median, "Right", "Left"))  
  
p2<-ggplot(eda, aes(x=factor(feature\_film), y=audience\_score, fill=factor(feature\_film)))+  
 geom\_boxplot()+  
 theme(legend.position = "none", axis.text.x = element\_text(angle = 60,hjust = 1))+  
 ggtitle("Audience Score Box Plot")+  
 labs(x="Feature Film?", y="Audience Score")  
  
mu2<-eda %>%  
 group\_by(feature\_film) %>%  
 summarise(Mean=mean(audience\_score))  
  
p2a<-ggplot(eda, aes(x=audience\_score, color=feature\_film, fill=feature\_film))+  
 geom\_density()+facet\_grid(feature\_film~.)+  
 geom\_vline(data=mu2, aes(xintercept=Mean),  
 linetype="dashed")+  
 theme(legend.position = "right", axis.text.x = element\_text(angle = 60,hjust = 1))+  
 ggtitle("Audience Score Distribution")+  
 labs(x="Audience Score", y="Density")

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FeatureFilm | Count | Min | 1st Qu. | Median | Mean | 3rd Qu. | Max | Skew |
| no | 59 | 19 | 77.0 | 86 | 81.20339 | 89 | 96 | Left |
| yes | 591 | 11 | 44.5 | 62 | 60.46531 | 78 | 97 | Left |

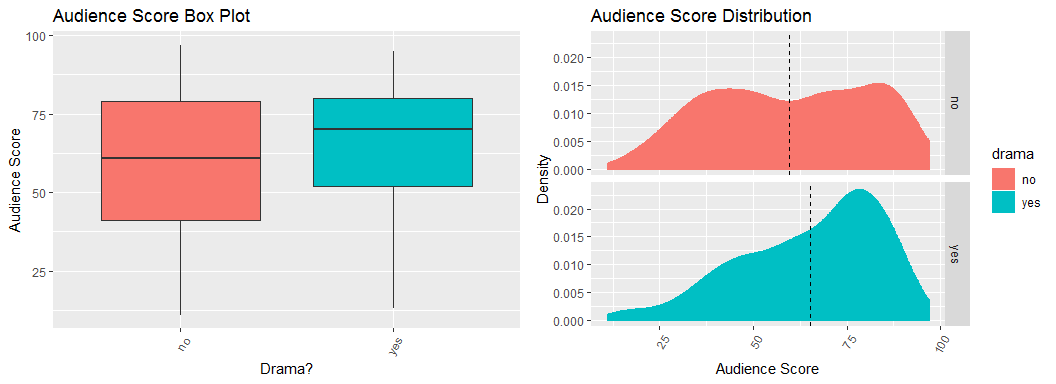


We see that there is a significant difference in the mean audience scores between feature films and non-feature films. There are far fewer non-feature films, but they tend to have higher audience scores.

We’ll take a similar look at the other factor variables we created.

#### Drama

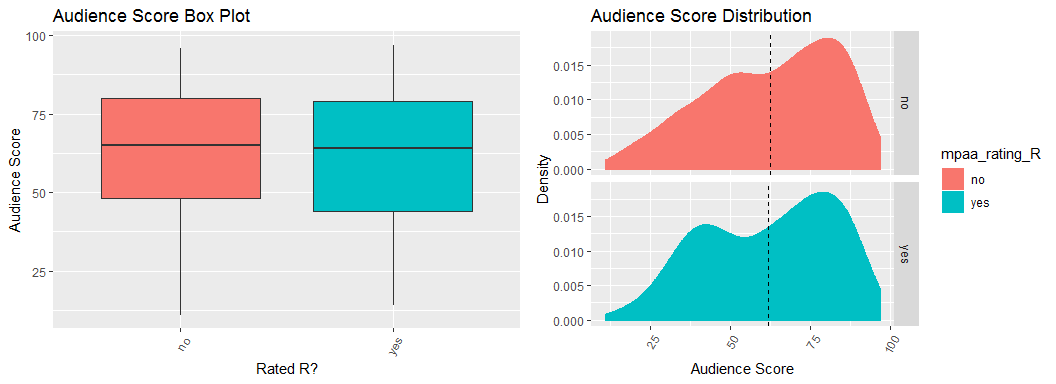
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Drama | Count | Min | 1st Qu. | Median | Mean | 3rd Qu. | Max | Skew |
| no | 345 | 11 | 41 | 61 | 59.69565 | 79 | 97 | Left |
| yes | 305 | 13 | 52 | 70 | 65.34754 | 80 | 95 | Left |



Movies classified as Dramas have a slightly higher mean audience score, however, it doesn’t appear significant from the plot.

#### R Rating

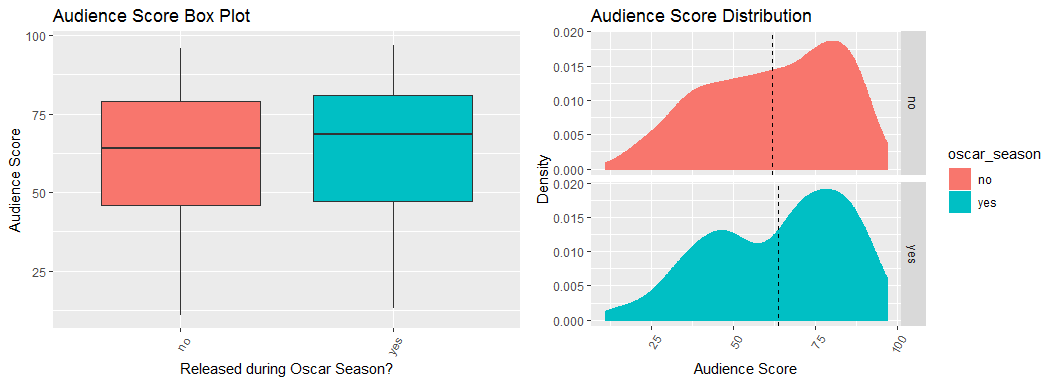
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R Rating | Count | Min | 1st Qu. | Median | Mean | 3rd Qu. | Max | Skew |
| no | 321 | 11 | 48 | 65 | 62.66044 | 80 | 96 | Left |
| yes | 329 | 14 | 44 | 64 | 62.04255 | 79 | 97 | Left |



There is no difference in the mean audience score for movies that are rated R and those that are not.

#### Oscar Season

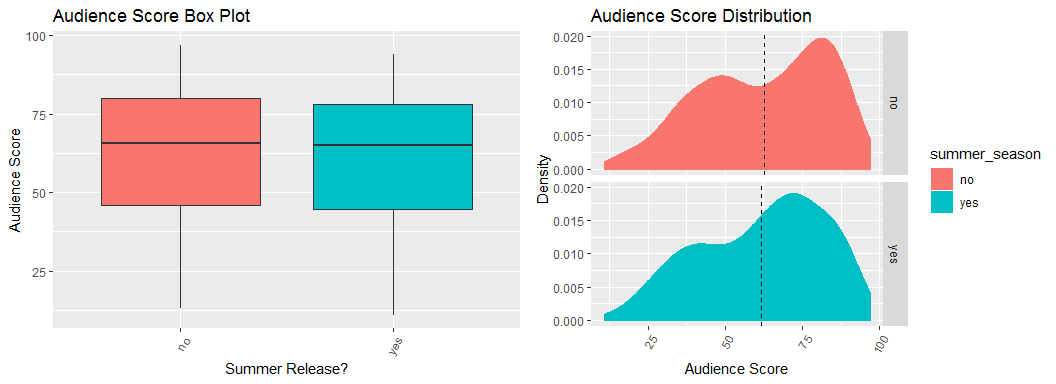
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Oscar Season | Count | Min | 1st Qu. | Median | Mean | 3rd Qu. | Max | Skew |
| no | 460 | 11 | 46.00 | 64.0 | 61.81304 | 79 | 96 | Left |
| yes | 190 | 13 | 47.25 | 68.5 | 63.64211 | 81 | 97 | Left |



Movies released during Oscar season have a slightly higher average audience score. I don’t expect to find much explanatory value from this variable.

#### Summer Season

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Summer Release | Count | Min | 1st Qu. | Median | Mean | 3rd Qu. | Max | Skew |
| no | 442 | 13 | 46.00 | 65.5 | 62.60181 | 80 | 97 | Left |
| yes | 208 | 11 | 44.75 | 65.0 | 61.80769 | 78 | 94 | Left |



Like the R rating, there is almost not difference in mean audience score between movies released during the summer and those that are not.

## Part 4: Modeling

Now that we’ve evaluated the data set, we’ll fit a bayesian linear model. I’ll use the bas.lm function from the BAS package with a BIC prior and uniform model prior. We have 16 predictor variables, so we will be generating 65,536 (2^p) models.

movbas<-bas.lm(audience\_score~., prior = "BIC", modelprior = uniform(), data=movdat)

#### Model Diagnostics

First, I’ll use the summary() function to have a look at top five models and the posterior inclusion probabilities for each predictor.

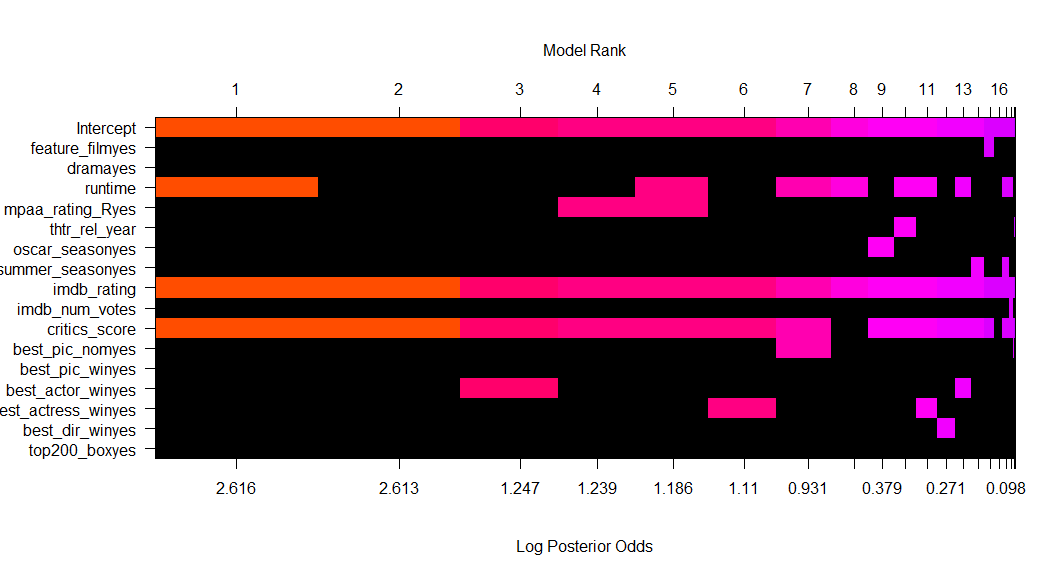
sumtable<-as.data.frame(round(summary(movbas), 3))  
  
names(sumtable)<-c("Probs", "Model1", "Model2", "Model3", "Model4", "Model5")  
  
sumtable$Stat<-rownames(sumtable)  
  
sumtable<-sumtable[,c(7,1,2,3,4,5,6)]  
  
sumtable<-arrange(sumtable, desc(Probs))  
  
kable(sumtable)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stat | Probs | Model1 | Model2 | Model3 | Model4 | Model5 |
| Intercept | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| imdb\_rating | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| critics\_score | 0.889 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| runtime | 0.470 | 1.000 | 0.000 | 0.000 | 0.000 | 1.000 |
| mpaa\_rating\_Ryes | 0.200 | 0.000 | 0.000 | 0.000 | 1.000 | 1.000 |
| best\_actor\_winyes | 0.144 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 |
| best\_actress\_winyes | 0.141 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| best\_pic\_nomyes | 0.131 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| thtr\_rel\_year | 0.091 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| summer\_seasonyes | 0.080 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| oscar\_seasonyes | 0.075 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| best\_dir\_winyes | 0.067 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| feature\_filmyes | 0.065 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| imdb\_num\_votes | 0.058 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| top200\_boxyes | 0.048 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| dramayes | 0.043 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| best\_pic\_winyes | 0.040 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| BF | NA | 1.000 | 0.997 | 0.254 | 0.252 | 0.239 |
| PostProbs | NA | 0.130 | 0.129 | 0.033 | 0.033 | 0.031 |
| R2 | NA | 0.755 | 0.752 | 0.754 | 0.754 | 0.756 |
| dim | NA | 4.000 | 3.000 | 4.000 | 4.000 | 5.000 |
| logmarg | NA | -3615.279 | -3615.282 | -3616.648 | -3616.657 | -3616.710 |

The imdb\_rating and critics\_score are the only two variables with a posterior inclusion probability of greater than .5. Of the top five models, the top two have a combined posterior probability of .26 of being the true model. The third best model’s posterior probability drops to .033, so after the first two models, the remaining posterior probability is very displersed. This creates a lot of uncertainty with a single model, so I will use bayesien model averaging (BMA) for my predictions.

#### Model Image

image(movbas, rotate=F)



From the model image we see that the top model includes imdb\_rating, critics\_score, and runtime. The second model, which is really just as likely as the first, drops runtime to include only imdb\_rating and critics\_score.

#### Confidence Intervals

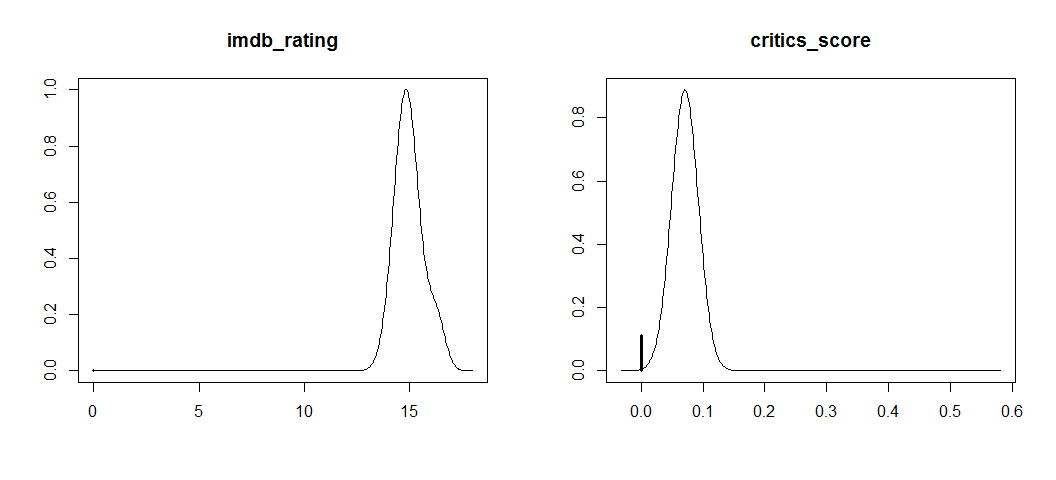
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stat | post mean | post SD | postprobincl | 2.5 | 97.5 |
| Intercept | 62.3476923 | 0.3946018 | 1.0000000 | 61.6076344 | 63.1450710 |
| imdb\_rating | 14.9820325 | 0.7310236 | 1.0000000 | 13.7144289 | 16.5912879 |
| critics\_score | 0.0629665 | 0.0302791 | 0.8885506 | 0.0000000 | 0.1050296 |
| runtime | -0.0256777 | 0.0311847 | 0.4697148 | -0.0823804 | 0.0000000 |
| mpaa\_rating\_Ryes | -0.3036174 | 0.7032090 | 0.1998402 | -2.1624550 | 0.0032228 |
| best\_actor\_winyes | -0.2876695 | 0.8318203 | 0.1443490 | -2.5856551 | 0.0000000 |
| best\_actress\_winyes | -0.3088382 | 0.9057061 | 0.1412809 | -2.9664879 | 0.0000000 |
| best\_pic\_nomyes | 0.5068035 | 1.5679029 | 0.1311914 | 0.0000000 | 4.8508224 |
| thtr\_rel\_year | -0.0045326 | 0.0181895 | 0.0906897 | -0.0552504 | 0.0002052 |
| summer\_seasonyes | 0.0870455 | 0.3831385 | 0.0804202 | 0.0000000 | 1.0882505 |
| oscar\_seasonyes | -0.0803494 | 0.3773258 | 0.0750568 | -1.0524814 | 0.0025367 |
| best\_dir\_winyes | -0.1195011 | 0.6227133 | 0.0669390 | -1.5744715 | 0.0045279 |
| feature\_filmyes | -0.1046908 | 0.5642631 | 0.0653695 | -1.1809170 | 0.0014825 |
| imdb\_num\_votes | 0.0000002 | 0.0000013 | 0.0577350 | 0.0000000 | 0.0000017 |
| top200\_boxyes | 0.0864818 | 0.7049625 | 0.0476223 | 0.0000000 | 0.0000000 |
| dramayes | 0.0160441 | 0.1939207 | 0.0431983 | 0.0000000 | 0.0000000 |
| best\_pic\_winyes | -0.0085028 | 0.8479272 | 0.0398477 | 0.0000000 | 0.0000000 |

For the imdb\_rating, we have a 95% credible interval that a unit increase in imdb\_rating will result in an increase in audience\_score of between 13.64 and 16.51, with the most likely increase of 14.98. For a unit increase in critics\_score we have a 95% credible interval of between a very small number close to zero and .1 increase in audience score. The most likely increase in this case is .06. All of the other predictor variables’ confidence intervals include 0.

#### Colinear variables

Even thoght imdb\_rating and critics\_score are correlated with each other, they are both included in the most likely models. We see from the plots below that the posterior probability (vertical bar) for critics\_score equalling zero is about .12, and that the same probability for imdb\_rating is very close to zero since there is no vertical bar in the plot.

par(mfrow=c(1,2))  
  
plot(coef, subset=c(9))  
plot(coef, subset=c(11))

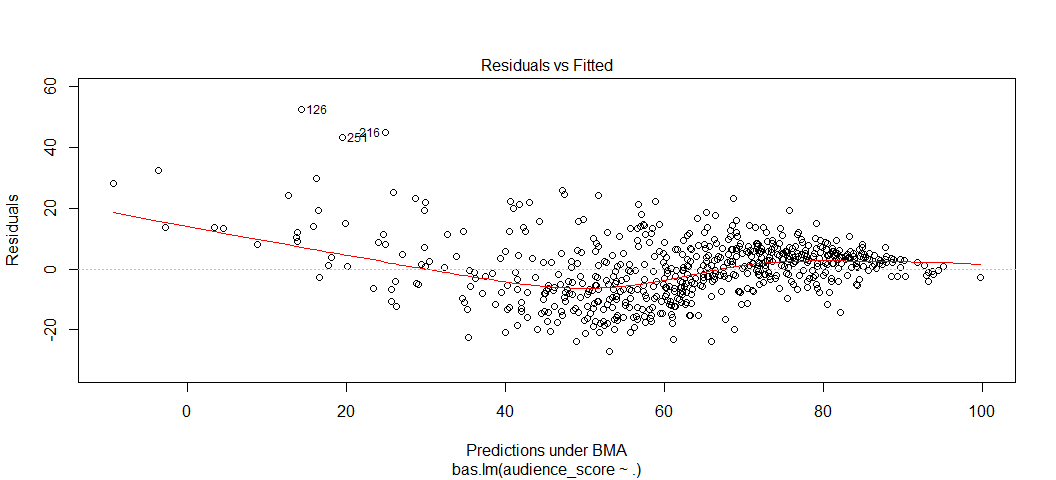


dev.off()

## null device   
## 1

#### Residuals vs. Fitted

plot(movbas, which=1)



We see from this plot that the residuals appear randomly distributed around zeor for the most part. For films that have a low audience score, the model tends to underestimate audience\_score which is shown by the few residuals all above zero on the left hand side of the plot.

The plot also highlights potential outliers at observations 126, 216, and 251. If I were to continue to develop this model, I would look for ways to address the few observations with low audience scores and evalutate the potential outliers further.

## Part 5: Prediction

Now I’ll predict the audience score from two movies not included in the data set. Data points for “The Revenant” and “La La Land” were pulled from the following web sites.

<https://www.boxofficemojo.com>

<https://www.rottentomatoes.com>

<https://www.imdb.com>

I’ll read in the new data and use the Predict() function to predict the audience\_score for the two new films.

predmovdat<-read.csv("predmovdat.csv")  
  
preds <- predict(movbas, predmovdat, estimator = "BMA", se.fit=TRUE)  
ci\_audience <- confint(preds, parm="pred")  
  
preds1<-as.data.frame(ci\_audience[,1:3])  
actual<-predmovdat[,1]  
preds1$actual<-actual  
preds1<-round(preds1, digits = 0)  
preds1$Movie<-c("The Revenant", "La La Land")  
preds1<-preds1[,c(5,4,3,1,2)]  
names(preds1)<-c("Movie", "Actual", "Predicted", "Lower CI", "Upper CI")  
  
kable(preds1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Movie | Actual | Predicted | Lower CI | Upper CI |
| The Revenant | 84 | 85 | 64 | 105 |
| La La Land | 81 | 87 | 66 | 106 |

We see that the model did a reasonably good job predicting the audience\_score for the two new movies. The model esimates fall well within the 95% credible interval, and the model estimates are fairly close to the actual audience scores.

## Part 6: Conclusion

In conclusion, I’ve found that the data provided include a few informative variables and many uninformative ones. I think that some further research might turn up some additional variables that would help improve model fit and prediction accuracy.

The model above should be considered a decent first draft. With more time, I would look at methods for improving model performance for films with a low audience score. I would also look to get a larger, more representative data set; perhaps by scraping all of the movie data from the IMDb and Rotten Tomatoes websites.