Bayesian modeling and prediction for movies

Setup

Load packages

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

Load data

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

Part 1: Data

The data consists of 651 randomly sampled movies produced and released between 1972 and 2016. For each sample movie, there is information on 32 features, such as ‘runtime’ and ‘audience_score’, the audience score on Rotten Tomatoes.

Since the data comes from an observational study, we cannot answer questions of causality, but rather attempt to identify correlation between variables. Specifically, we are interested in investigating the relationship between the variable ‘audience_score’ and the other features.

We would like to apply Bayesian inference methods to build a model that would predict the value of ‘audience_score’ given a set of values for some other variables.

Part 2: Data manipulation

First, we remove the observations which contain missing entries.

```
#eliminate rows with NA entries
movies_no_na=na.omit(movies)
```

Many of the variables are categorical with multiple levels. We use some of these to create new, binary categorical features, which we are going to regard as potential predictors in our final model.

```
# create new variables

movies_no_na <- movies_no_na%>%

  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==11 | thtr_rel_month==10 | thtr_rel_month==12, "yes"
, "no")) %>%
  mutate(summer_season=ifelse(thtr_rel_month==5 | thtr_rel_month==6 | thtr_rel_month == 7 | thtr
_rel_month ==8, "yes", "no"))
```

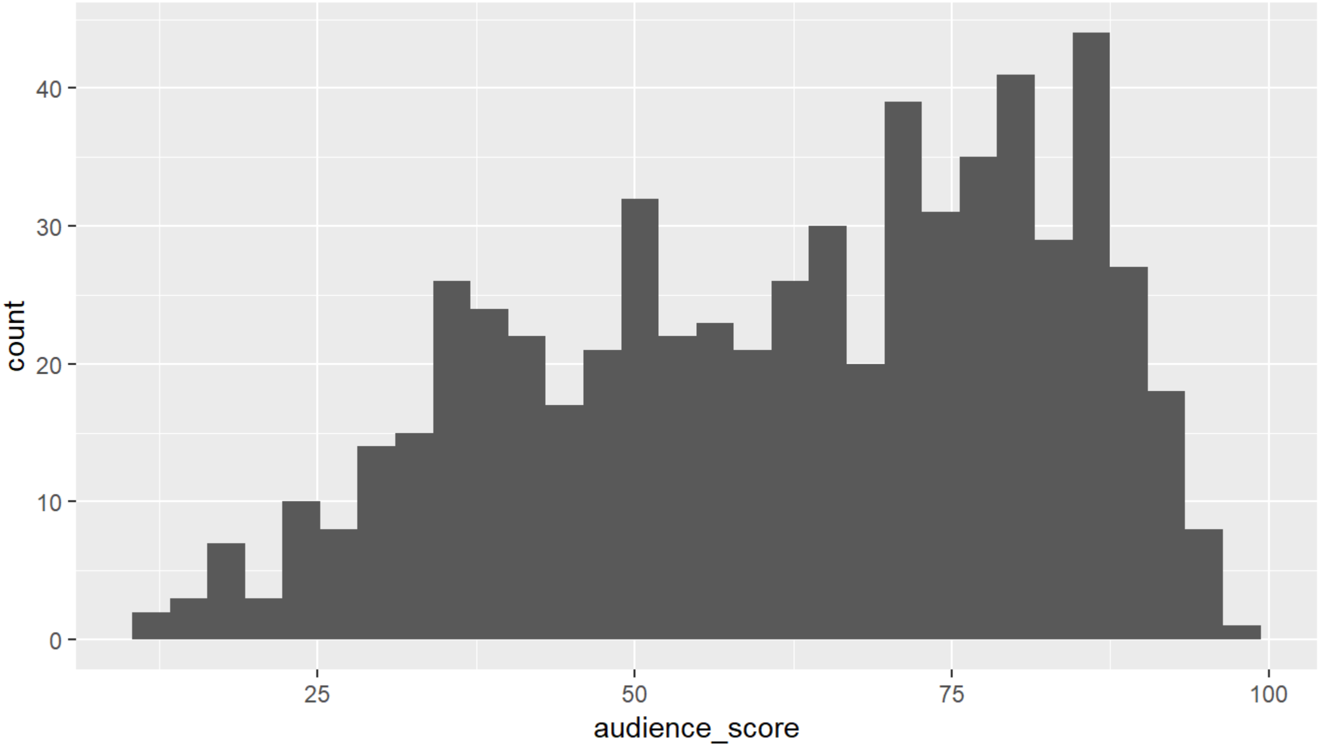
Part 3: Exploratory data analysis

Let's start with a visualization of the distribution of the variable 'audience_score'.

```
#distribution of audience_score

ggplot(movies_no_na, aes(x=audience_score)) +geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
summary(movies_no_na$audience_score)
```

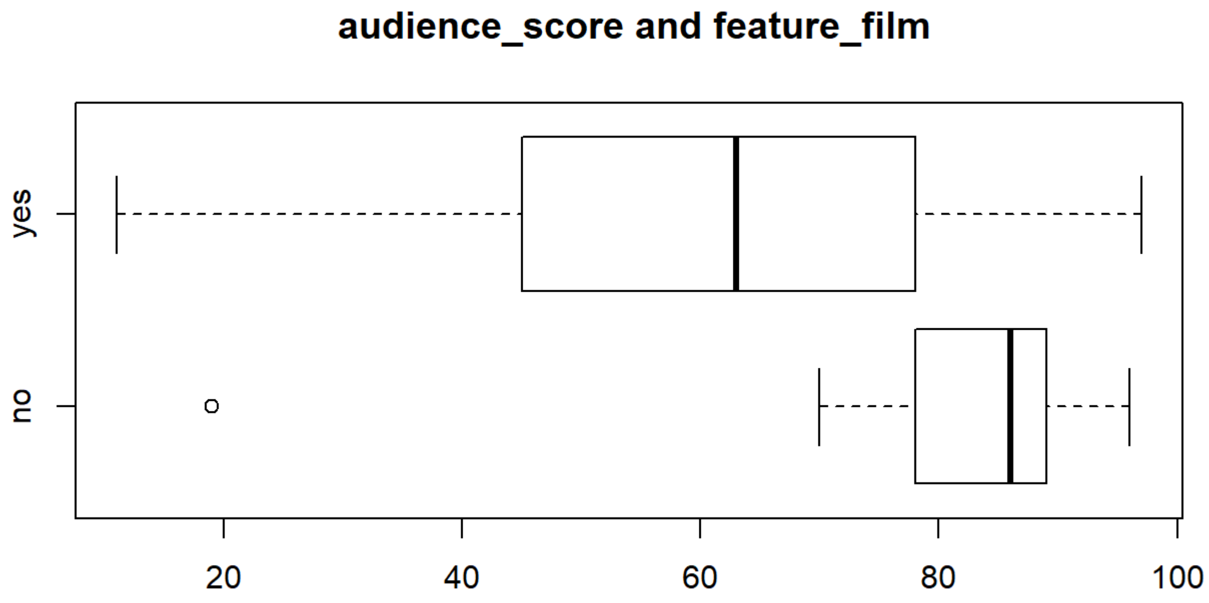
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   11.00   46.00   65.00   62.21   80.00   97.00
```

The histogram and summary statistics indicate that the distribution of our sample ‘audience_score’ data is left-skewed, ranging between 11 and 97, with sample mean 62.21.

Now, for each of the five categorical variables defined earlier, we create a side-by-side boxplot to compare its distribution to the ‘audience_score’ distribution.

The goal is to see whether each of these categorical variable has a clear effect on the audience score. So, below each boxplot, we include a bayesian inference analysis on comparing means via a Bayes factor.

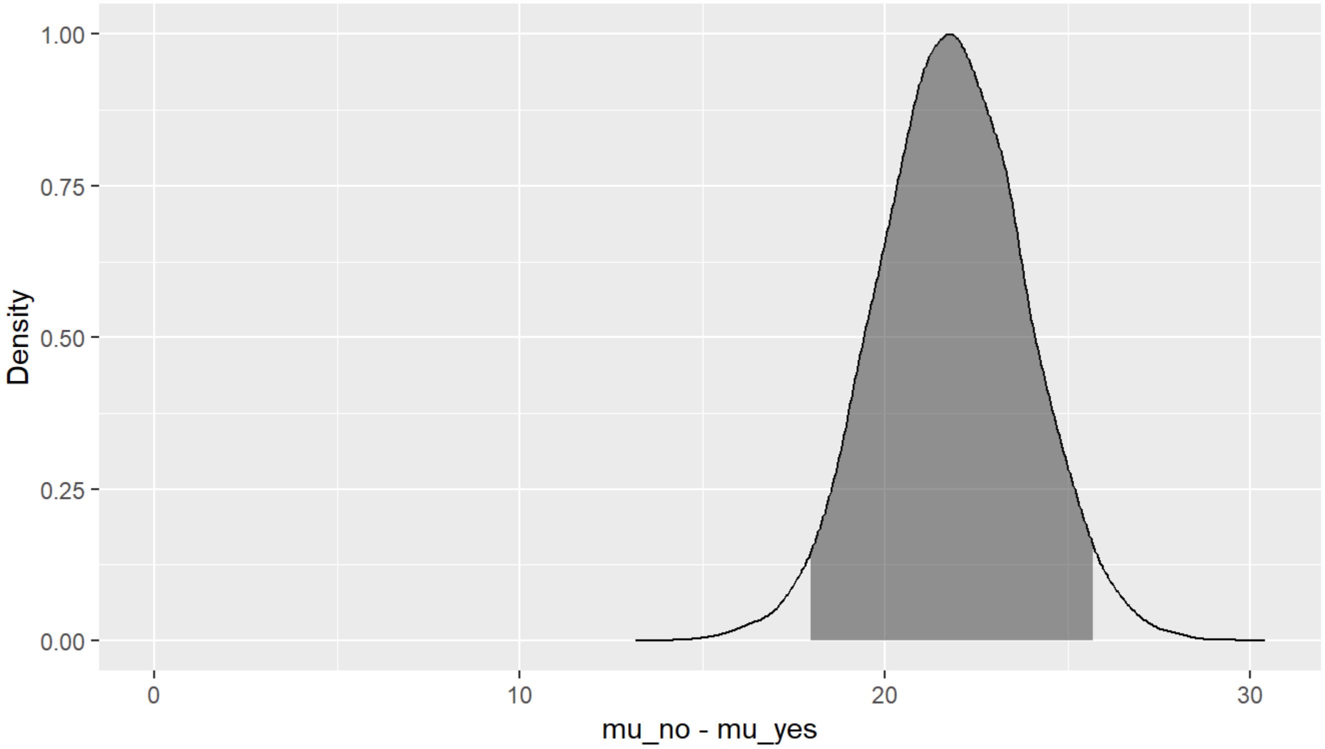
```
# boxplot of audience_score and feature_film
boxplot(movies_no_na$audience_score ~ movies_no_na$feature_film, main=" audience_score and featu
re_film", horizontal=TRUE)
```



There visible difference between the medians and overall distributions of the two groups. This suggests that there might be a relationship between ‘feature_film’ and ‘audience_score’. To investigate further, let’s consider two competing models, each with prior probability 1/2: the null hypothesis H_1 claiming that the true means of the two categories are equal and the alternative hypothesis H_2 claiming the two true means are different. We can use the bayes_inference function to obtain Bayes factors and posterior probabilities of our two models.

```
bayes_inference(y = audience_score, x =feature_film, data = movies_no_na, statistic = "mean", ty
pe = "ht", null = 0, alternative = "twosided")
```

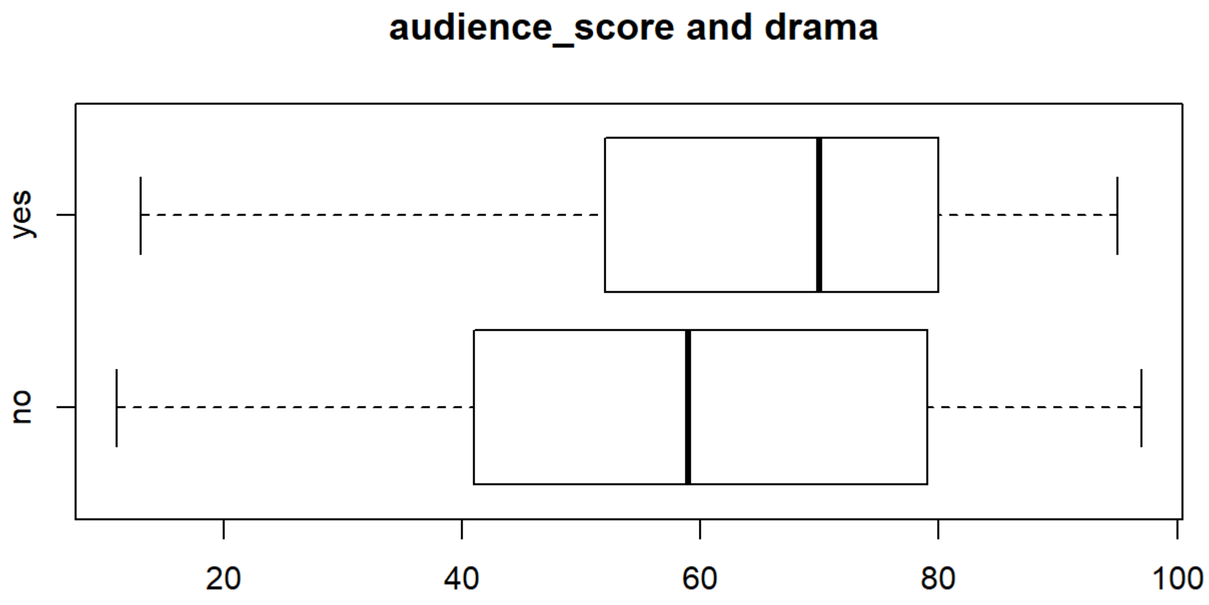
```
## Response variable: numerical, Explanatory variable: categorical (2 levels)
## n_no = 46, y_bar_no = 82.5435, s_no = 11.9177
## n_yes = 573, y_bar_yes = 60.5777, s_yes = 19.8187
## (Assuming intrinsic prior on parameters)
## Hypotheses:
## H1: mu_no  = mu_yes
## H2: mu_no != mu_yes
##
## Priors:
## P(H1) = 0.5
## P(H2) = 0.5
##
## Results:
## BF[H2:H1] = 1.212332e+13
## P(H1|data) = 0
## P(H2|data) = 1
```



From these results, $BF[H_2 : H_1] > 150$, so the data provide very strong evidence against H_1. We conclude that variable feature_film category strongly influences the audience score, on average.

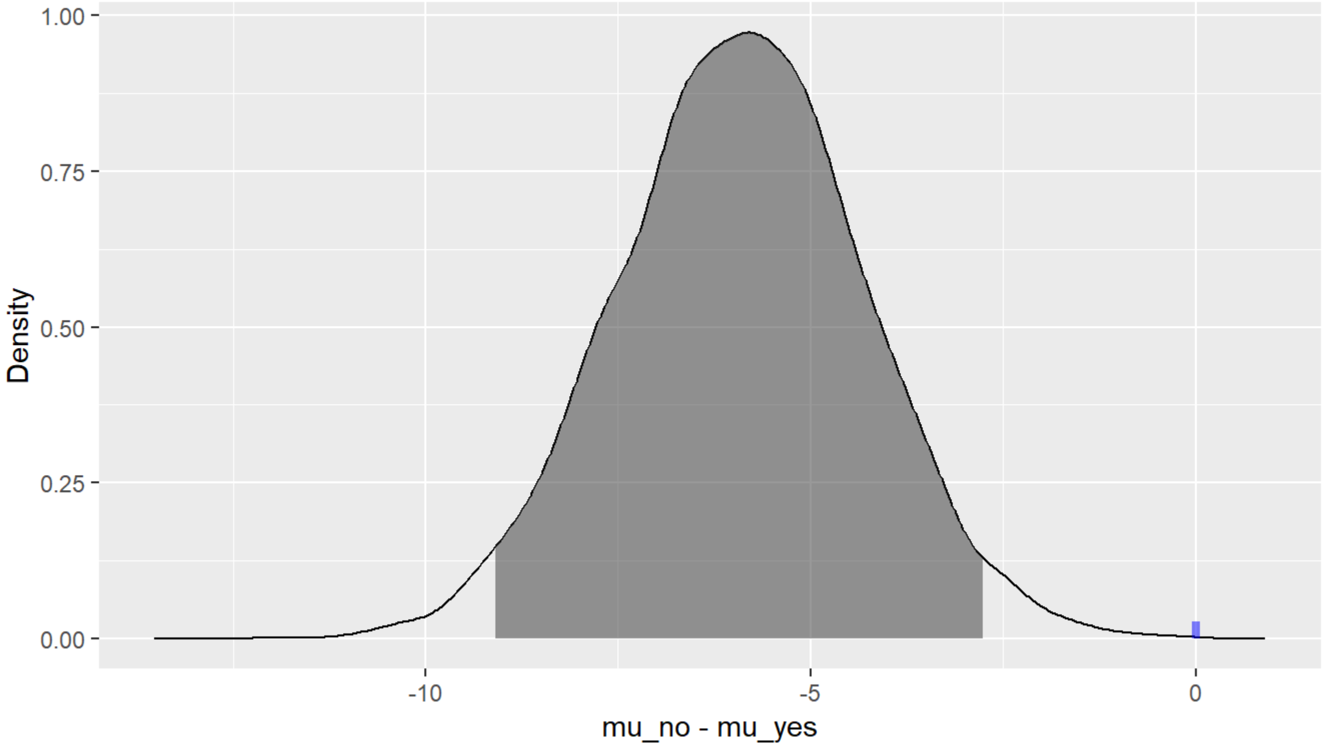
For the remaining 4 categorical variables, we perform similar Bayesian analysis on the null and alternative hypotheses about the true means of the groups .

```
# boxplot of audience_score and drama
boxplot(movies_no_na$audience_score ~ movies_no_na$drama, main=" audience_score and drama", hori
zontal=TRUE)
```



```
bayes_inference(y = audience_score, x =drama, data = movies_no_na, statistic = "mean", type = "h
t", null = 0, alternative = "twosided")
```

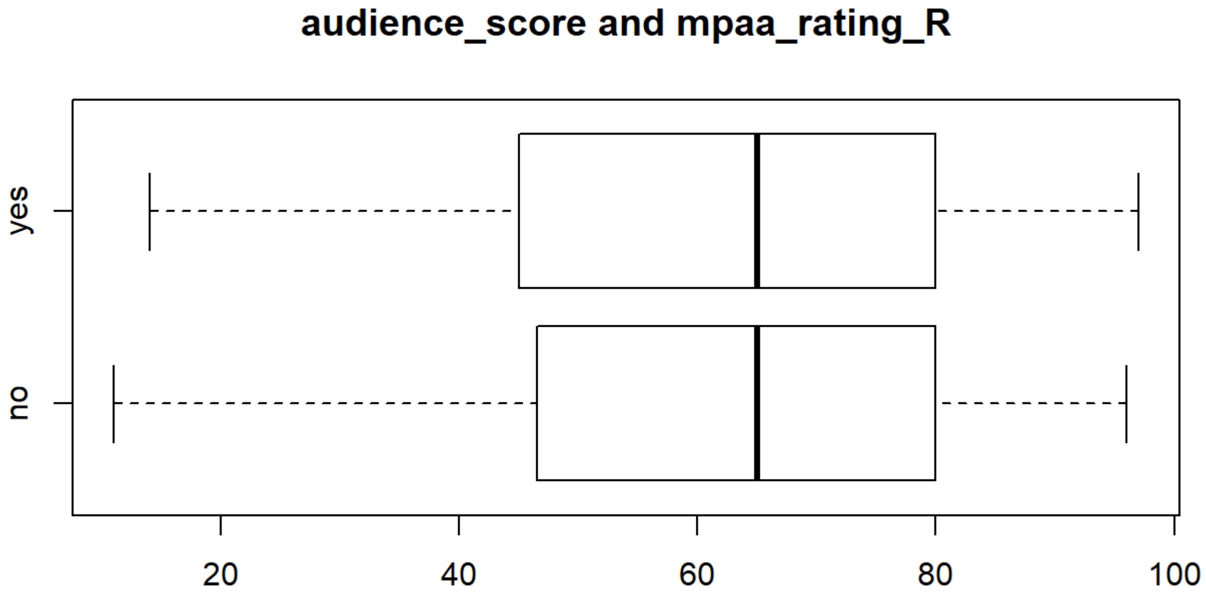
```
## Response variable: numerical, Explanatory variable: categorical (2 levels)
## n_no = 321, y_bar_no = 59.352, s_no = 21.1448
## n_yes = 298, y_bar_yes = 65.2886, s_yes = 18.6305
## (Assuming intrinsic prior on parameters)
## Hypotheses:
## H1: mu_no  = mu_yes
## H2: mu_no != mu_yes
##
## Priors:
## P(H1) = 0.5
## P(H2) = 0.5
##
## Results:
## BF[H2:H1] = 34.6357
## P(H1|data) = 0.0281
## P(H2|data) = 0.9719
```



In this case, we got $BF[H2:H1] = 34.6357$, which provides strong evidence against H_1. We conclude that the ‘drama’ variable has a strong effect on the ‘audience_score’.

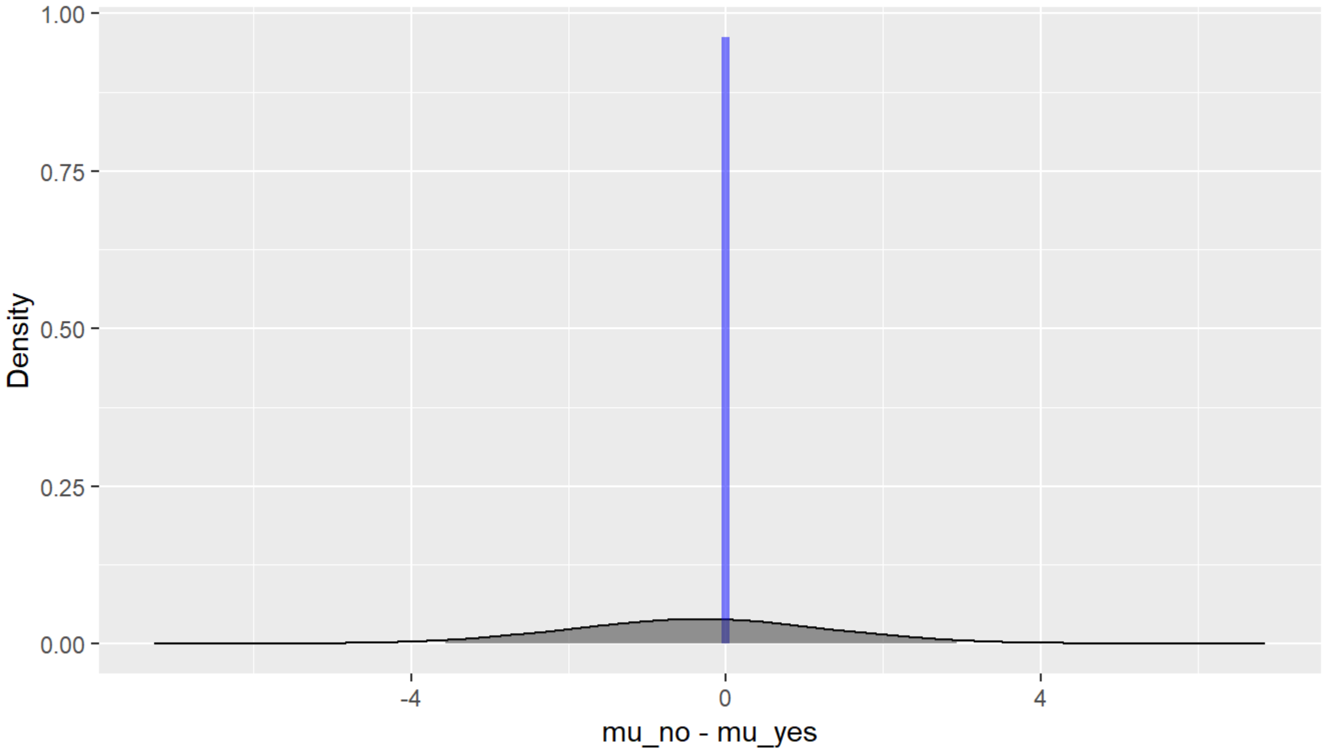
Three more categorical variables remain to be studied.

```
#boxplot of audience_score and mpaa_rating_R
boxplot(movies_no_na$audience_score ~ movies_no_na$mpaa_rating_R, main="audience_score and mpaa_
rating_R", horizontal=TRUE)
```



```
bayes_inference(y = audience_score, x =mpaa_rating_R, data = movies_no_na, statistic = "mean", t
ype = "ht", null = 0, alternative = "twosided")
```

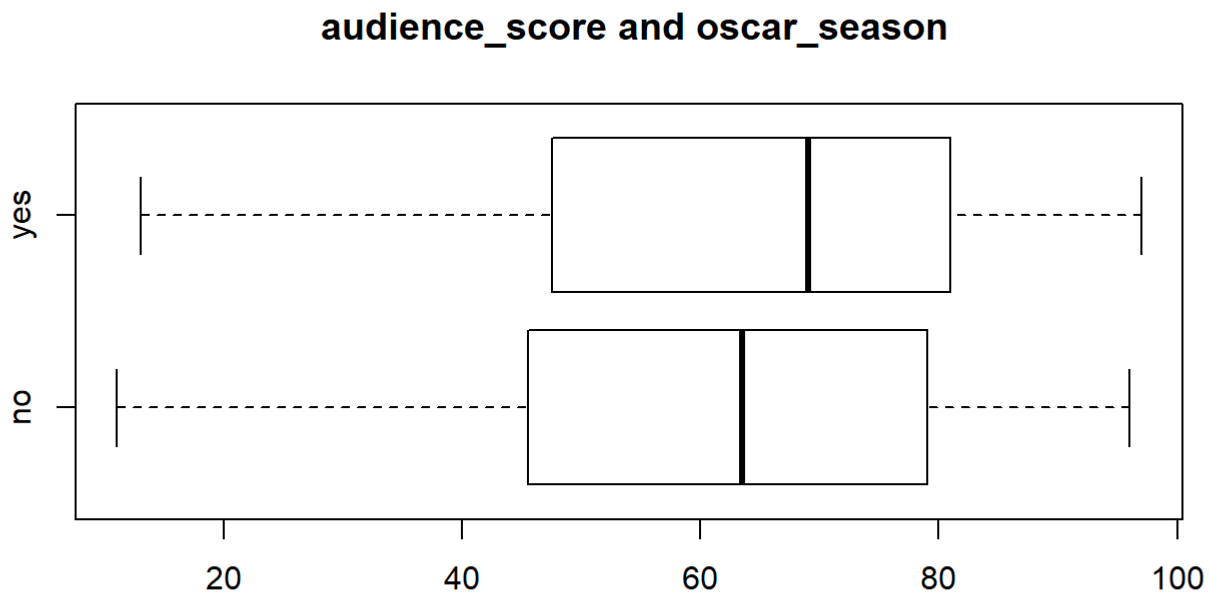
```
## Response variable: numerical, Explanatory variable: categorical (2 levels)
## n_no = 300, y_bar_no = 62.0367, s_no = 20.3187
## n_yes = 319, y_bar_yes = 62.373, s_yes = 20.0743
## (Assuming intrinsic prior on parameters)
## Hypotheses:
## H1: mu_no  = mu_yes
## H2: mu_no != mu_yes
##
## Priors:
## P(H1) = 0.5
## P(H2) = 0.5
##
## Results:
## BF[H1:H2] = 24.8392
## P(H1|data) = 0.9613
## P(H2|data) = 0.0387
```



The distrbutions of the two groups are very similar. The Bayes factor of 24.83 indicates positive evidence against H_2. Based on this, we assume no relationship between the two variables.

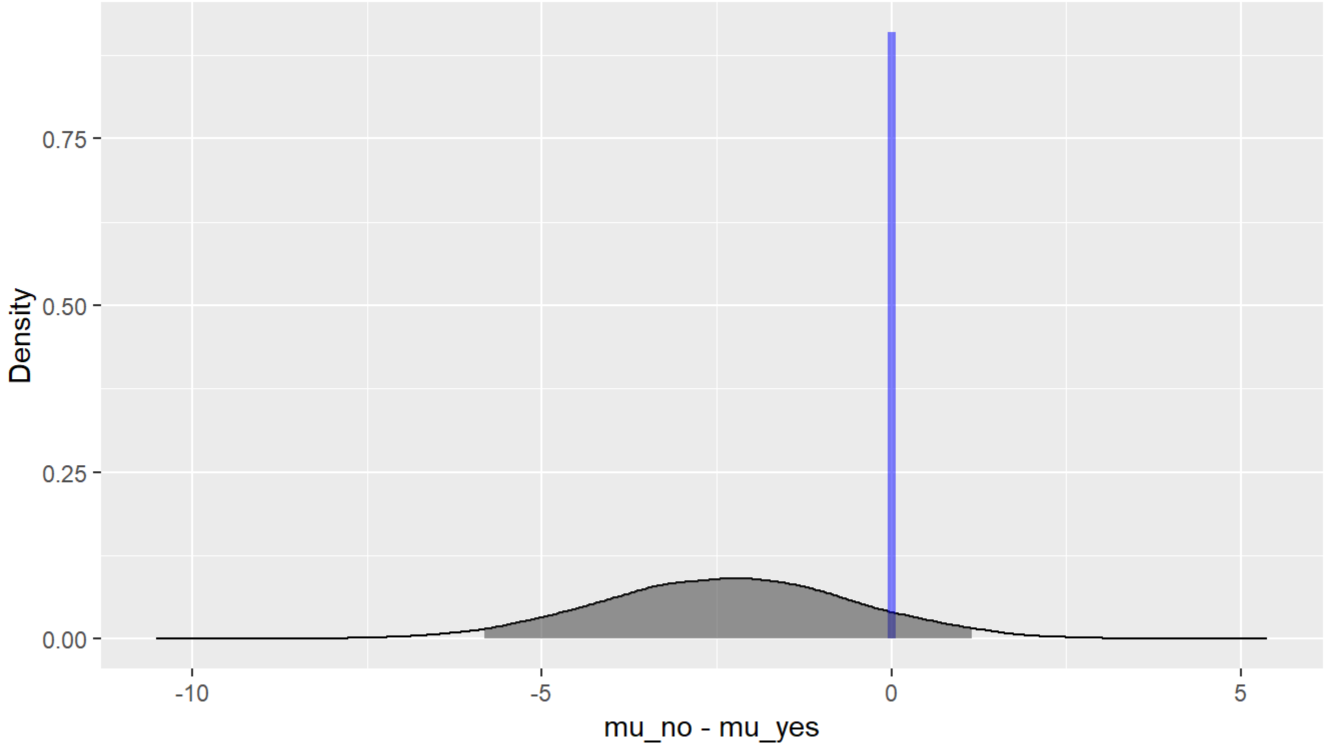
Next, we consider analysis for the ‘oscar_season’ variable.

```
#boxplot of audience_score and oscar_season
boxplot(movies_no_na$audience_score ~ movies_no_na$oscar_season, main=" audience_score and oscar
_season", horizontal=TRUE)
```



```
bayes_inference(y = audience_score, x =oscar_season, data = movies_no_na, statistic = "mean", ty
pe = "ht", null = 0, alternative = "twosided")
```

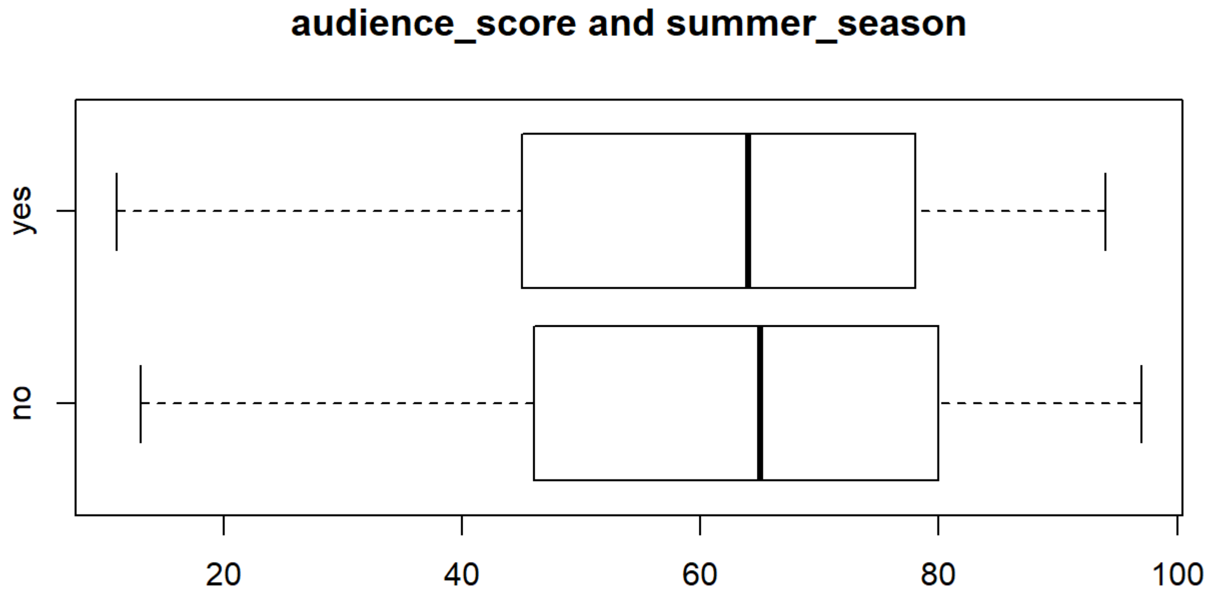
```
## Response variable: numerical, Explanatory variable: categorical (2 levels)
## n_no = 440, y_bar_no = 61.5386, s_no = 20.107
## n_yes = 179, y_bar_yes = 63.8603, s_yes = 20.3118
## (Assuming intrinsic prior on parameters)
## Hypotheses:
## H1: mu_no  = mu_yes
## H2: mu_no != mu_yes
##
## Priors:
## P(H1) = 0.5
## P(H2) = 0.5
##
## Results:
## BF[H1:H2] = 10.019
## P(H1|data) = 0.9092
## P(H2|data) = 0.0908
```



The Bayes factor and posterior probabilities suggest there is positive evidence against H_2. We conclude that ‘oscar_season’does not have an overall strong effect on ’audience score’.

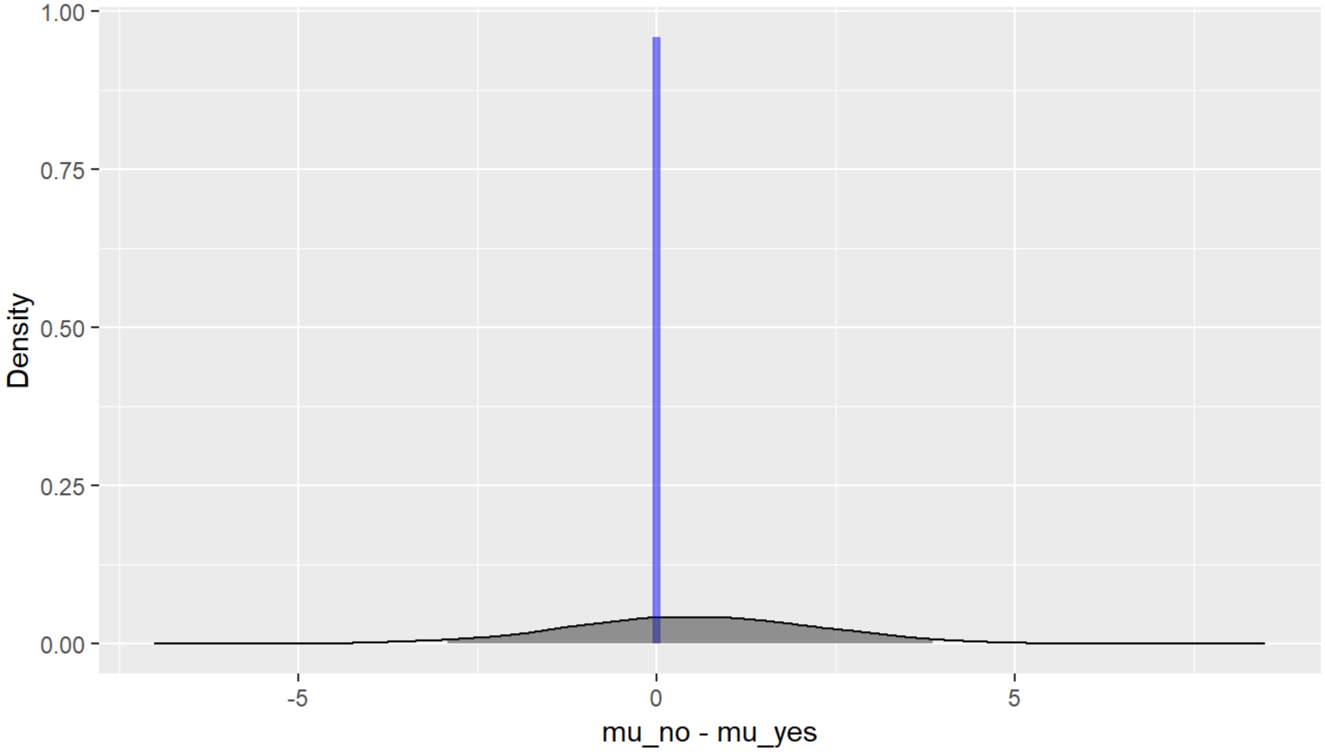
Finally, we look at the the ‘summer_season’ variable.

```
#boxplot of audience_score and summer_season
boxplot(movies_no_na$audience_score ~ movies_no_na$summer_season, main="audience_score and summe
r_season", horizontal=TRUE)
```



```
bayes_inference(y = audience_score, x =summer_season, data = movies_no_na, statistic = "mean", t
ype = "ht", null = 0, alternative = "twosided")
```

```
## Response variable: numerical, Explanatory variable: categorical (2 levels)
## n_no = 418, y_bar_no = 62.3828, s_no = 20.3266
## n_yes = 201, y_bar_yes = 61.8507, s_yes = 19.9092
## (Assuming intrinsic prior on parameters)
## Hypotheses:
## H1: mu_no  = mu_yes
## H2: mu_no != mu_yes
##
## Priors:
## P(H1) = 0.5
## P(H2) = 0.5
##
## Results:
## BF[H1:H2] = 22.7623
## P(H1|data) = 0.9579
## P(H2|data) = 0.0421
```



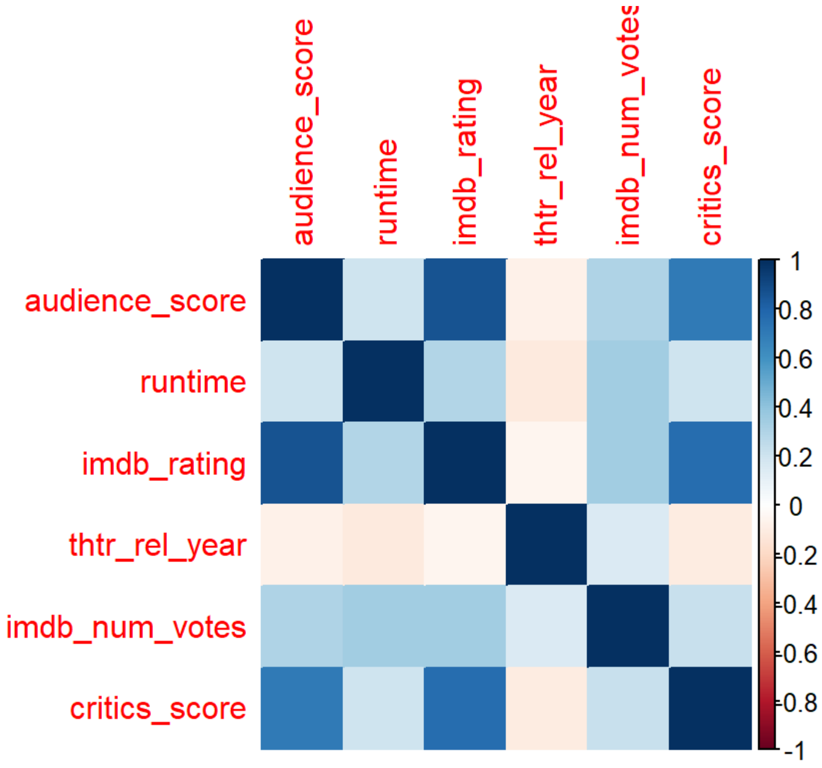
A Bayes factor of 22.76 suggests there is positive evidence against H_2, so we assume that 'summer_season' doesn't have an overall strong effect on "audience_score".

After this Bayesian analysis concerning the effect of a categorical variable on 'audience_score', we decidde to keep 'feature_film' and 'drama' among the predictors to be included in our final model, while disregarding 'oscar_season', 'summer_season', 'mpaa_rating_R'.

We now turn to some numerical variables of interest. These are 'audience_score', 'runtime', 'imdb_rating', 'thtr_rel_year', 'imdb_num_votes', 'critics_scores'. Let's use a correlation matrix to explore the relationship between them.

```
# correlation matrix of numerical variables
dat <- movies_no_na[, c( 'audience_score', 'runtime',  'imdb_rating', 'thtr_rel_year',
                        'imdb_num_votes', 'critics_score')]
corr <-cor(dat)


corrplot(corr, method='color')
```



The feature ‘thtr_rel_year’ seems to be weakly correlated to all the other variables. This also makes intuitive sense, since the year the movie was released in theatres shouldn’t affect the audience score of the movie.

We decide to disregard the variable ‘thtr_rel_year’ from our model, while keeping the variables ‘runtime’, ‘imdb_rating’, ‘imdb_num_votes’, ‘critics_score’ as potential predictors.

From the remaining unexplored variables, we select ‘best_pic_nom’, ‘best_pic_win’, ‘best_actor_win’, ‘best_actress_win’, ‘best_dir_win’, ‘top200_box’ to add to our list of predictors. Intuitively, we expect these variables to affect the popularity of a movie and in turn, the audience score.

Part 4: Modeling

After our exploratory analysis, we settle on 12 final potential predictors out of the initial 32 variales in the dataset. We apply Bayesian model averaging on the set of all possible linear models we could build through combinations of these 12 variables, where each model is assumed to be equally likely.

```
bma_audience_score = bas.lm(audience_score ~ feature_film+drama+runtime+imdb_rating+
                              imdb_num_votes+critics_score+best_pic_nom+best_pic_win+best_actor_
win+best_actress_win+best_dir_win+top200_box,
                               data =movies_no_na, prior = "BIC",
                              modelprior = uniform())
bma_audience_score
```

```
##
## Call:
## bas.lm(formula = audience_score ~ feature_film + drama + runtime +
##     imdb_rating + imdb_num_votes + critics_score + best_pic_nom +
##     best_pic_win + best_actor_win + best_actress_win + best_dir_win +
##     top200_box, data = movies_no_na, prior = "BIC", modelprior = uniform())
##
##
##  Marginal Posterior Inclusion Probabilities:
##           Intercept      feature_filmyes             dramayes
##             1.00000              0.05932              0.04365
##             runtime          imdb_rating       imdb_num_votes
##             0.51350              1.00000              0.05980
##       critics_score      best_pic_nomyes      best_pic_winyes
##             0.92874              0.12965              0.04072
##   best_actor_winyes  best_actress_winyes      best_dir_winyes
##             0.11678              0.14657              0.06736
##       top200_boxyes
##             0.04972
```

```
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        0.05932139     0.0000     0.0000000     0.0000000
## dramayes               0.04364996     0.0000     0.0000000     0.0000000
## runtime                0.51350496     1.0000     0.0000000     0.0000000
## imdb_rating            1.00000000     1.0000     1.0000000     1.0000000
## imdb_num_votes         0.05979605     0.0000     0.0000000     0.0000000
## critics_score          0.92874345     1.0000     1.0000000     1.0000000
## best_pic_nomyes        0.12964940     0.0000     0.0000000     0.0000000
## best_pic_winyes        0.04072311     0.0000     0.0000000     0.0000000
## best_actor_winyes      0.11677525     0.0000     0.0000000     0.0000000
## best_actress_winyes    0.14656683     0.0000     0.0000000     1.0000000
## best_dir_winyes        0.06735553     0.0000     0.0000000     0.0000000
## top200_boxyes          0.04972344     0.0000     0.0000000     0.0000000
## BF                             NA     1.0000     0.8715404     0.2048238
## PostProbs                      NA     0.2363     0.2060000     0.0484000
## R2                             NA     0.7483     0.7455000     0.7470000
## dim                            NA     4.0000     3.0000000     4.0000000
## logmarg                        NA -3434.7520 -3434.8894481 -3436.3375603
##                           model 4       model 5
## Intercept               1.0000000     1.0000000
## feature_filmyes         0.0000000     0.0000000
## dramayes                0.0000000     0.0000000
## runtime                 1.0000000     0.0000000
## imdb_rating             1.0000000     1.0000000
## imdb_num_votes          0.0000000     0.0000000
## critics_score           1.0000000     1.0000000
## best_pic_nomyes         1.0000000     0.0000000
## best_pic_winyes         0.0000000     0.0000000
## best_actor_winyes       0.0000000     1.0000000
## best_actress_winyes     0.0000000     0.0000000
## best_dir_winyes         0.0000000     0.0000000
## top200_boxyes           0.0000000     0.0000000
## BF                      0.1851908     0.1745817
## PostProbs               0.0438000     0.0413000
## R2                      0.7495000     0.7469000
## dim                     5.0000000     4.0000000
## logmarg             -3436.4383237 -3436.4973172
```

The summary table includes the most likely 5 models listed in decreasing order, according to their posterior probability. The first column gives the probability that the coefficient corresponding to a predictor is nonzero, given the data.

The first model listed, let's call it model 1, has the highest posterior probability, equal to 0.2363. It contains the variables 'runtime', 'imdb_rating' and 'critics_score'. Model 2 follows closely with posterior probability of 0.2060 and predictors 'imdb_rating' and 'critics_score'. The first two models differentiate themselves from the rest by a 0.20 difference.

Part 5: Prediction

Let’s use model 1 to make a prediction on the audience_score for a movie called ‘Captain America: Civil War (2016)’. From the IMDb website (https://www.imdb.com/title/tt3498820/?ref_=adv_li_tt (https://www.imdb.com/title/tt3498820/?ref_=adv_li_tt)) and Rotten Tomatoes website (https://www.rottentomatoes.com/m/captain_america_civil_war (https://www.rottentomatoes.com/m/captain_america_civil_war)), we learn that this movie has a runtime of 147 minutes, critics_score of 91, imdb_rating of 7.8 and audience_score of 89. We’d like to see how well our model predicts the audience_score.

First, we build the model:

```
#build the linear model with specified predictors
model1 <- lm(audience_score ~ runtime+ imdb_rating+critics_score, data=movies_no_na)
summary(model1)
```

```
##
## Call:
## lm(formula = audience_score ~ runtime + imdb_rating + critics_score,
##     data = movies_no_na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.996  -6.706   0.738   5.435  52.550
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -32.90825    3.35451  -9.810  < 2e-16 ***
## runtime        -0.05791    0.02238  -2.588 0.009891 **
## imdb_rating    14.95043    0.60249  24.814  < 2e-16 ***
## critics_score   0.07531    0.02227   3.381 0.000769 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.15 on 615 degrees of freedom
## Multiple R-squared:  0.7483, Adjusted R-squared:  0.747
## F-statistic: 609.4 on 3 and 615 DF,  p-value: < 2.2e-16
```

Next, we create a small dataframe for this movie that we will feed into model1 to obtain a prediction.

```
runtime=c(147)
imdb_rating=c(7.8)
critics_score=c(91)

captain_2016=data.frame(runtime, imdb_rating, critics_score)
audience_score_captain <- predict(model1, captain_2016)
audience_score_captain
```

```
##        1
## 82.04604
```

The predicted value for audience_score using model 1 is 82.04604 while the true score posted on Rotten Tomatoes is 89.

Part 6: Conclusion

Our variable selection process used a combination of bayesian inference, correlation matrix and intuition in order to identify 12 out of the 32 features that were most likely to affect the value of the audience score of a movie. We applied Bayesian model averaging to obtain the posterior model inclusion probability for each of the 12 variables and the most probable models. The model containing the variables ‘critics_score’, ‘runtime’, imdb_rating’ had the highest posterior probability, equal to 0.2363. This value might seem small, but it is much larger than the uniform prior probability assigned to it, since there are 2^{12} possible models. We tested the model on a randomly chosen movie and the score prediction did not turn out too far from the true score.