Bayesian Statistics Project

Setup

Load packages

```r
library(ggplot2)
library(dplyr)
library(statsr)
library(BAS)
library(broom)
```

Load data

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

Part 1: Data

According to the description of this data set, the data set consists of 651 randomly sampled movies produced and released before 2016. Besides, the sample size, 651 movies, is less than 10% of the population, and the sample size is greater than 30. With that being said, these samples are qualified to be independent. And since we aren't sure whether this data set is randomly assigned, we could conclude that this data set is generalized but not causal.

Part 2: Data manipulation

Firstly, we create a new variable based on `title_type`, called `feature_film` with level yes (feature films) and no.

```r
 movies<- movies %>%
  mutate(feature_film = ifelse(movies$title_type == "Feature Film", "yes","no"))
```

Secondly, we create a new variable based on `genre`, called `drama` with levels yes (movies that are dramas) and no.

```r
movies <- movies %>%
  mutate(drama = ifelse(movies$genre == "Drama", "yes","no"))
```

Thirdly, we create a new variable based on `mpaa_rating`, called `mpaa_rating_R` with levels yes (R rated movies) and no.

```
movies <-movies %>%
  mutate(mpaa_rating_R = ifelse(movies$mpaa_rating == "R", "yes", "no"))
```

Finally, we create two new variables based on `thtr_rel_month`, the one is called `oscar_season` with levels yes (if the movie is released in November, October, or December) and no; the other is called `summer_season` with levels yes (if the film is released in May, June, July, or August) and no.

```
movies <-movies%>%
  mutate(oscar_season = ifelse(movies$thtr_rel_month == 11&10&12, "yes","no"))
```

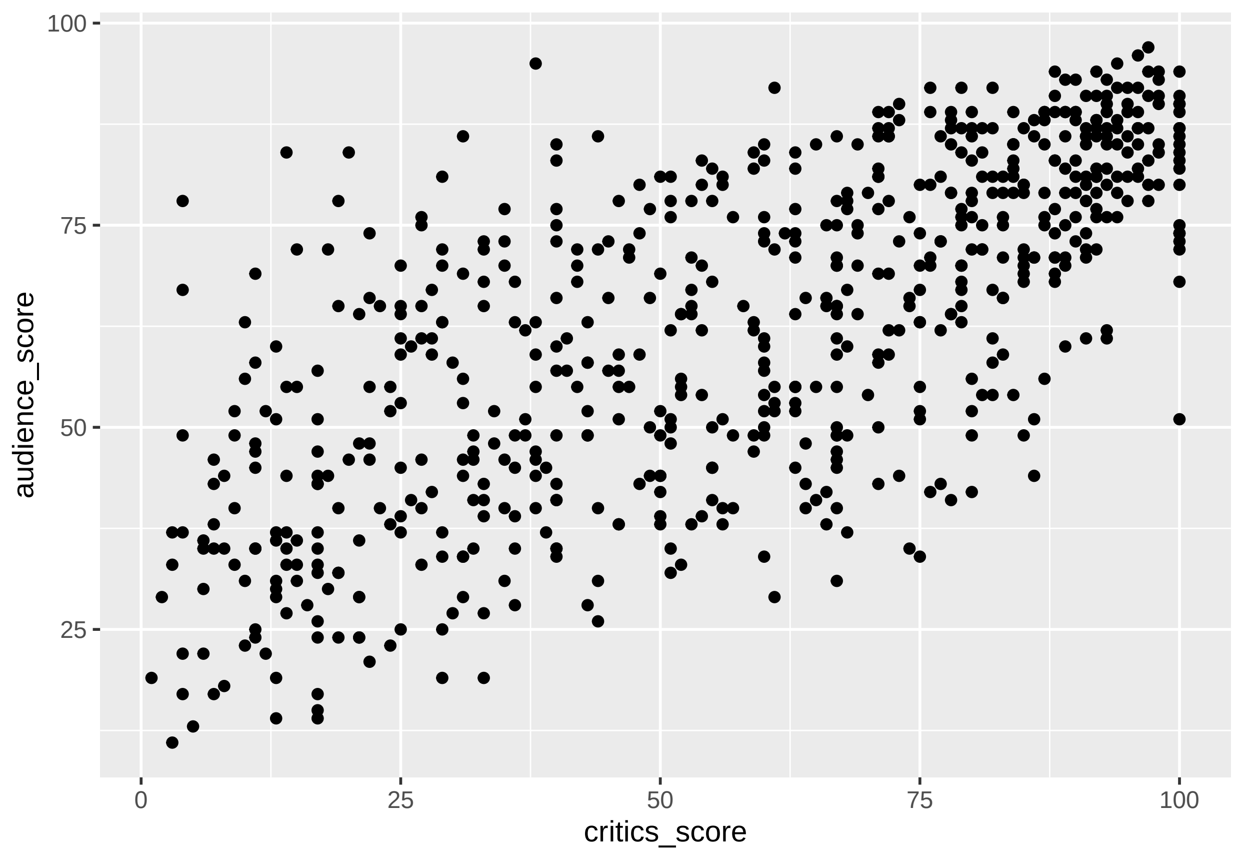
```
movies <-movies %>%
  mutate(summer_season = ifelse(movies$thtr_rel_month == 5&6&7&8, "yes","no"))
```

Part 3: Exploratory data analysis

Step 1:

First of all, we create a scatterplot to see the relationship between audience score and critics score since most of us may refer to the critics' score before seeing the movie or making comments.

```
ggplot(data = movies, aes(x = critics_score, y = audience_score)) +
  geom_point()
```

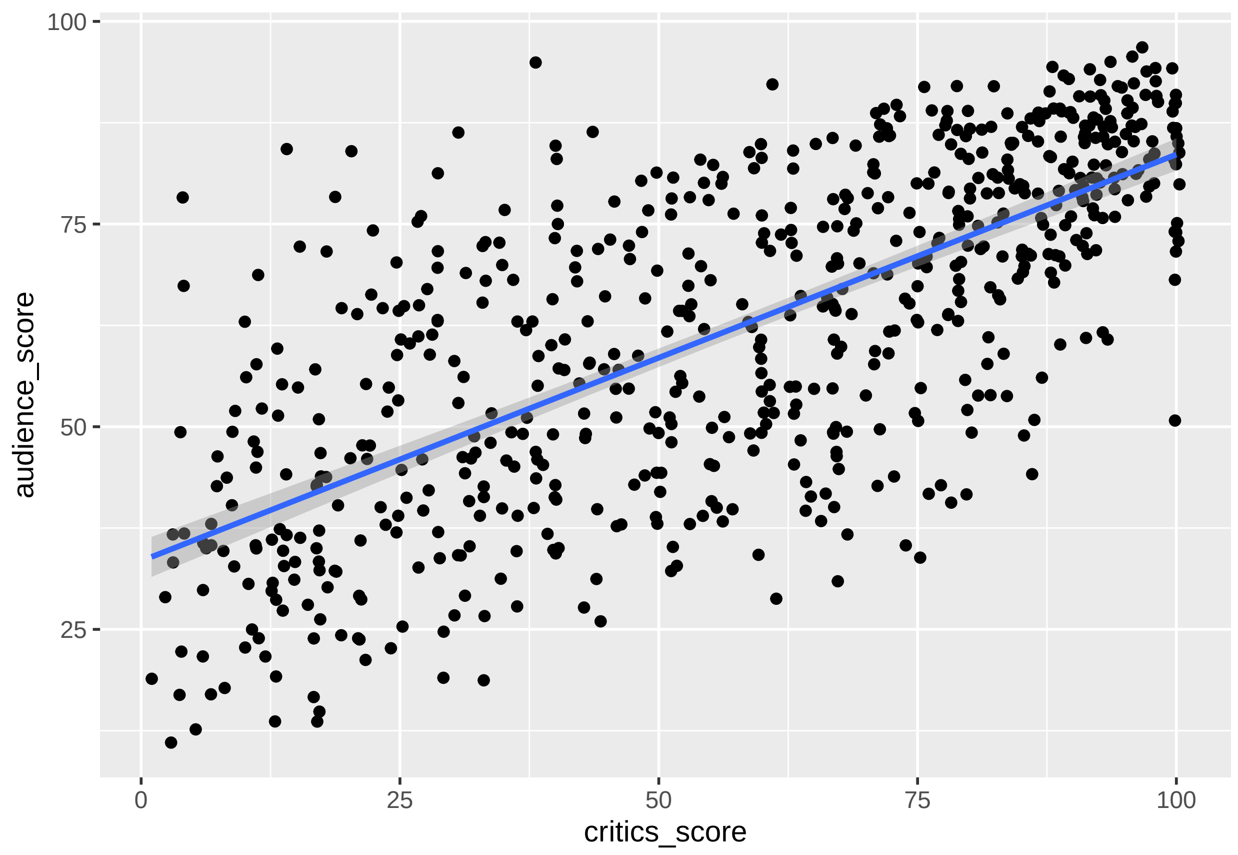


Here we can tell there's an apparent correlation between these two variables, and the direction is positive. And the distribution is more concentrated on the group when the critics' score is between 75 and 100.

Now we want to see if the trend in this plot is something more than natural variation:

```
ggplot(data = movies, aes(x = critics_score, y = audience_score)) +
  geom_jitter() +
  geom_smooth(method = "lm")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

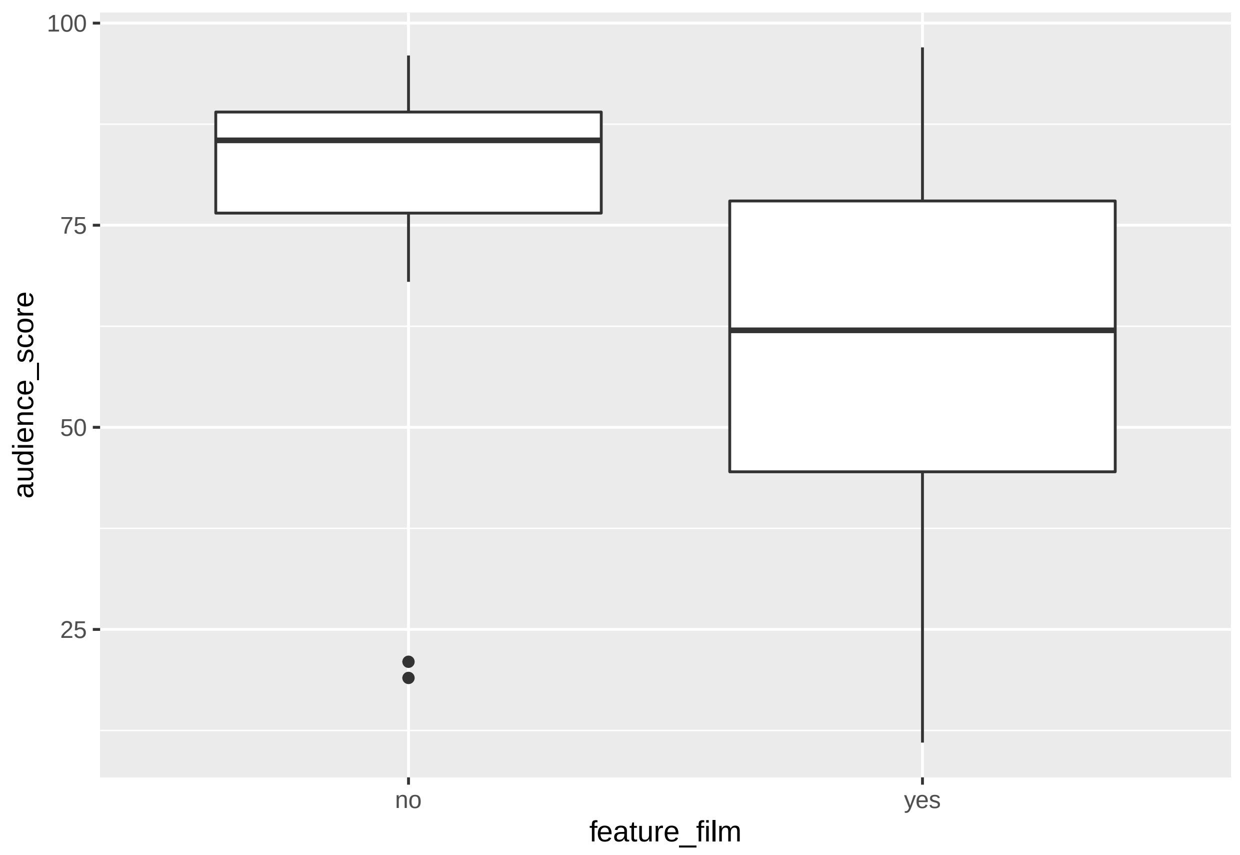


The shaded gray area around the line above demonstrates the variability we expect in our estimation, and now we know the trend in this plot is within a natural variation.

Step 2:

We want to get to know more about the relationship between one of the categorical variables in this data set (feature_film) and our response variable (audience_score)

```r
movies%>%
  ggplot(movies, mapping = aes(x = feature_film, y = audience_score)) +
  geom_boxplot()
```



From the box plot above, it's easy to tell that the variation of audience scores towards feature films is more extensive than those not categorized as a feature film. And the median of those non-feature films is also higher than feature films.

Step 3:

Here we use the numerical summary to get closer to its distribution since it's a numerical variable.

```r
movies%>%
  filter(!is.na(audience_score)) %>%
  summarise(asmean = mean(audience_score), asmedian = median(audience_score), asd = sd(audience_score),
```

```
## # A tibble: 1 x 5
##   asmean asmedian   asd asmin asmax
##    <dbl>    <dbl> <dbl> <dbl> <dbl>
## 1   62.4       65  20.2    11    97
```

From above, we get to know that the mean audience score is around 62.4 and the median is 65, indicating that this distribution is a bit left-skewed. Also, combined with our exploration in step 2, it's clear that half of the audience score falls into the group focus on the feature film, which to some extent suggests the popularity of feature films among audiences in this data set.

Part 4: Modeling

Step 1: Multiple Linear Regression

According to this project's requirement, we know that we need to set up a multiple linear regression model among several predictors.

Firstly, let's take a look at multiple regression before removing any cases.

```
m_movies_full <-lm(audience_score ~ feature_film + drama + runtime + mpaa_rating_R + thtr_rel_year +osca
```

And here we use the function summary() to see the output here:

```
summary(m_movies_full)
```

```
##
## Call:
## lm(formula = 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, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.674  -6.210   0.198   5.815  53.151
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.219e+02  7.742e+01   1.575   0.1158
## feature_filmyes     -2.252e+00  1.682e+00  -1.339   0.1810
## dramayes             1.328e+00  8.747e-01   1.519   0.1293
## runtime             -5.975e-02  2.394e-02  -2.496   0.0128 *
## mpaa_rating_Ryes    -1.437e+00  8.123e-01  -1.769   0.0773 .
## thtr_rel_year       -7.520e-02  3.831e-02  -1.963   0.0501 .
## oscar_seasonyes      4.460e-01  1.483e+00   0.301   0.7637
## summer_seasonyes     2.715e+00  1.584e+00   1.714   0.0869 .
## imdb_rating          1.473e+01  6.064e-01  24.298   <2e-16 ***
## imdb_num_votes       7.438e-06  4.519e-06   1.646   0.1003
## critics_score        5.666e-02  2.217e-02   2.556   0.0108 *
## best_pic_nomyes      5.192e+00  2.612e+00   1.988   0.0473 *
## best_pic_winyes     -3.288e+00  4.622e+00  -0.711   0.4771
## best_actor_winyes   -1.784e+00  1.178e+00  -1.515   0.1304
## best_actress_winyes -2.142e+00  1.303e+00  -1.643   0.1008
## best_dir_winyes     -9.814e-01  1.731e+00  -0.567   0.5709
## top200_boxyes        6.093e-01  2.780e+00   0.219   0.8266
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.968 on 633 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7633, Adjusted R-squared:  0.7573
## F-statistic: 127.6 on 16 and 633 DF,  p-value: < 2.2e-16
```

As we can see from the quick summary of the full linear model, many independent variables' coefficients are not statistically significant. Here, we want to use BIC (Bayesian Information Criterion) used for model selection. This method is based on model fit while simultaneously penalizing the number of parameters in proportion to the sample size.

Firstly, we get to know the BIC of the full model we just set up.

```
BIC(m_movies_full)
```

```
## [1] 4933.208
```

Now we can compare the BIC of the full model with that of a reduced model. To begin with, let's try to remove best__actor__win, best__actress__win, and best__dir__win.

```
m_movie_nobest3win <- lm(audience_score ~ feature_film + drama + runtime + mpaa_rating_R + thtr_rel_yea
BIC(m_movie_nobest3win)
```

```
## [1] 4919.696
```

As we can see, removing the three best-oriented variables from the regression reduces BIC, which we seek to minimize by model selection.

Next,we try to remove best__pic__nom and best__pic__win to explore the changes of BIC.

```
m_movie_no2bestpic <- lm(audience_score ~ feature_film + drama + runtime + mpaa_rating_R + thtr_rel_yea
BIC(m_movie_no2bestpic)
```

```
## [1] 4909.684
```

Still, there's an apparent reduction in BIC after removing the two variables related to whether the film is nominated or awarded with Oscar.

We may want to know BIC's change when we remove the variables related to the film's release month. e.g., oscar season, summer season.

```
m_movie_no3season<- lm(audience_score ~ feature_film + drama + runtime + mpaa_rating_R  + imdb_rating +
BIC(m_movie_no3season)
```

```
## [1] 4896.792
```

From the result, BIC declined after removing the three variables.

Next, let's remove the variable top200__box to see whether or not the movie is in the Top 200 Box Office list has something to do with audience scores.

```
m_movie_nobox<- lm(audience_score ~ feature_film + drama + runtime + mpaa_rating_R  + imdb_rating + imd
BIC(m_movie_nobox)
```

```
## [1] 4890.474
```

As a result, BIC declined after removing this variable.

Then we choose to remove the variable drama to see whether or not the genre of drama films makes a difference in the audience score.

```
m_movie_nodrama<- lm(audience_score ~ feature_film + runtime + mpaa_rating_R  + imdb_rating + imdb_num_
BIC(m_movie_nodrama)
```

```
## [1] 4885.431
```

As expected, BIC still went down after removing this variable.

What about the feature film? Let's remove it to see what will happen.

```
m_movie_noffilm<- lm(audience_score ~ runtime + mpaa_rating_R  + imdb_rating + imdb_num_votes + critics_
BIC(m_movie_noffilm)
```

```
## [1] 4879.32
```

BIC is still declining. This time we want to remove the runtime of movies.

```
m_movie_noruntime<- lm(audience_score ~ mpaa_rating_R  + imdb_rating + imdb_num_votes + critics_score, 
BIC(m_movie_noruntime)
```

```
## [1] 4887.589
```

Finally, there's an increase in BIC, indicating that we should keep runtime this time and try to remove another variable, imdb_num_votes.

```
m_movie_noimdbnv<- lm(audience_score ~ mpaa_rating_R  + runtime+imdb_rating  + critics_score, data = mo
BIC(m_movie_noimdbnv)
```

```
## [1] 4874.484
```

This indicates that we should continue our next steps based on this model selection and keep run time in the model. Now we want to remove the variable mpaa_rating_R to see if there's a difference.

```
m_movie_nompaa<- lm(audience_score ~ runtime + imdb_rating  + critics_score, data = movies)
BIC(m_movie_nompaa)
```

```
## [1] 4871.623
```

The BIC did reduce again, denoting that this removal step is sensible. Now we try to remove the imdb_rating.

```
m_movie_noimdbr<- lm(audience_score ~ runtime + critics_score, data = movies)
BIC(m_movie_noimdbr)
```

```
## [1] 5329.265
```

This time there is a BIC spike after removing the imdb_rating variable, so clearly, this variable makes a significant difference in our response variable audience score. We need to keep this variable in our model and keep removing the other variables critics_score and run time to see the difference.

```
m_movie_nocriscore<- lm(audience_score ~ runtime + imdb_rating, data = movies)
BIC(m_movie_nocriscore)
```

```
## [1] 4875.773
```

```
m_movie_noruntime2<- lm(audience_score ~ imdb_rating + critics_score, data = movies)
BIC(m_movie_noruntime2)
```

```
## [1] 4878.542
```

By comparing the results above, now we can conclude that the best model of good fitness in the BIC method is audience_score ~ runtime + imdb_rating + critics_score.

Step 2: Bayesian Model Averaging

Generally, several models are equally plausible, and choosing only one of them would ignore the inherent uncertainty involved in choosing different variables in the model. Therefore, here we implement BMA (Bayesian Model Averaging), where multiple models are averaged to obtain posterior of different coefficients and predictions from new data.

We start it by applying BMA to the data using all potential predictors according to the project rubric requirement. Here we also use function summary() to see the top 5 most possible models.

```
bma_audiencescore <- bas.lm(audience_score ~ feature_film + drama + runtime + mpaa_rating_R + thtr_rel_
                            prior = "BIC",
                            modelprior = uniform())
```

```
## Warning in bas.lm(audience_score ~ feature_film + drama + runtime +
## mpaa_rating_R + : dropping 1 rows due to missing data
```

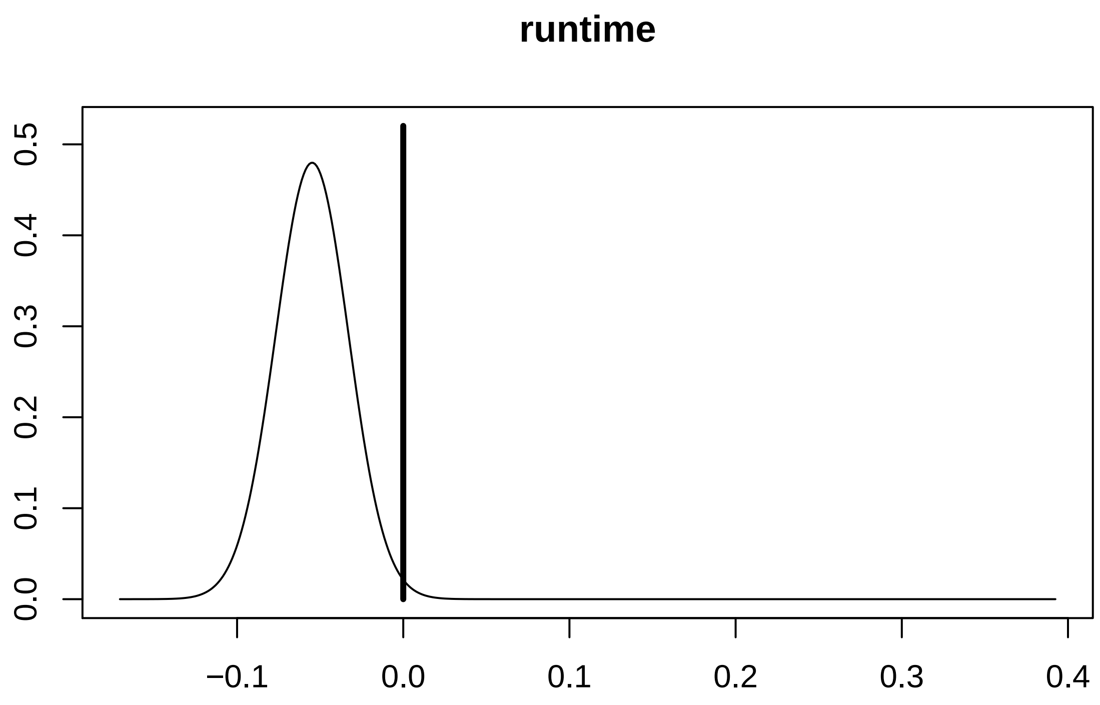
```
summary(bma_audiencescore)
```

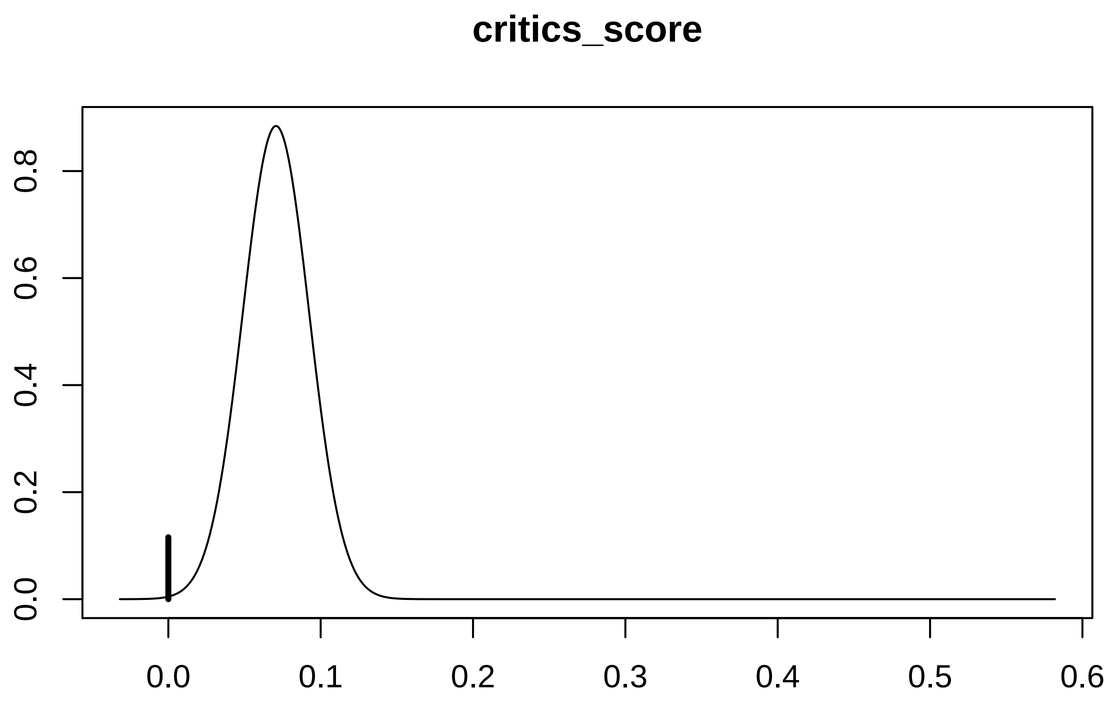
```
##                      P(B != 0 | Y)    model 1       model 2       model 3
## Intercept               1.00000000     1.0000     1.0000000     1.0000000
## feature_filmyes         0.06539662     0.0000     0.0000000     0.0000000
## dramayes                0.04345796     0.0000     0.0000000     0.0000000
## runtime                 0.47975438     1.0000     0.0000000     0.0000000
## mpaa_rating_Ryes        0.19881279     0.0000     0.0000000     0.0000000
## thtr_rel_year           0.09042234     0.0000     0.0000000     0.0000000
## oscar_seasonyes         0.03831324     0.0000     0.0000000     0.0000000
## summer_seasonyes        0.13758919     0.0000     0.0000000     0.0000000
## imdb_rating             1.00000000     1.0000     1.0000000     1.0000000
## imdb_num_votes          0.05804388     0.0000     0.0000000     0.0000000
## critics_score           0.88428345     1.0000     1.0000000     1.0000000
## best_pic_nomyes         0.13078323     0.0000     0.0000000     0.0000000
## best_pic_winyes         0.03982753     0.0000     0.0000000     0.0000000
## best_actor_winyes       0.14895993     0.0000     0.0000000     1.0000000
## best_actress_winyes     0.14000643     0.0000     0.0000000     0.0000000
## best_dir_winyes         0.06580905     0.0000     0.0000000     0.0000000
## top200_boxyes           0.04737610     0.0000     0.0000000     0.0000000
## BF                               NA     1.0000     0.9968489     0.2543185
## PostProbs                        NA     0.1267     0.1263000     0.0322000
## 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               0.0320000     0.0303000
## R2                      0.7539000     0.7563000
## dim                     4.0000000     5.0000000
## logmarg             -3616.6568544 -3616.7095127
```

Now we also want to visualize the posterior distribution of the coefficients under the model averaging approach. Based on our new model found with the BIC method above, we want to graph the posterior distribution of the coefficients of critics score and run time.

```
# We obtain the coefficients from the model 'bma_audience'
coef_bmad <- coefficients(bma_audiencescore)

#'critics_score is the 11th variable, while 'runtime' is t he 4th variable in the data set.
plot(coef_bmad, subset = c(4, 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



Focus on the first plot for the runtime of movies: the vertical bar represents the posterior probability that the coefficient is 0, around 49%. And the bell-shaped curve represents the density of plausible values from the models where the coefficient is not 0, and it's centered slightly far from 0.

As for the critics' score on Rotten Tomatoes of movies: the probability that the coefficient is non-zero is relatively small, so the vertical bar is not evident, only around 12%. The range of plausible values is also slightly far from 0, reflecting our beliefs after seeing the data that this variable is essential.

Step 3: Model Diagnostics

The Bayesian model assumes that the errors are normally distributed with constant variance and that the mean expected audience score is linear in our three predictors: runtime, imdb_rating, and critics_score.

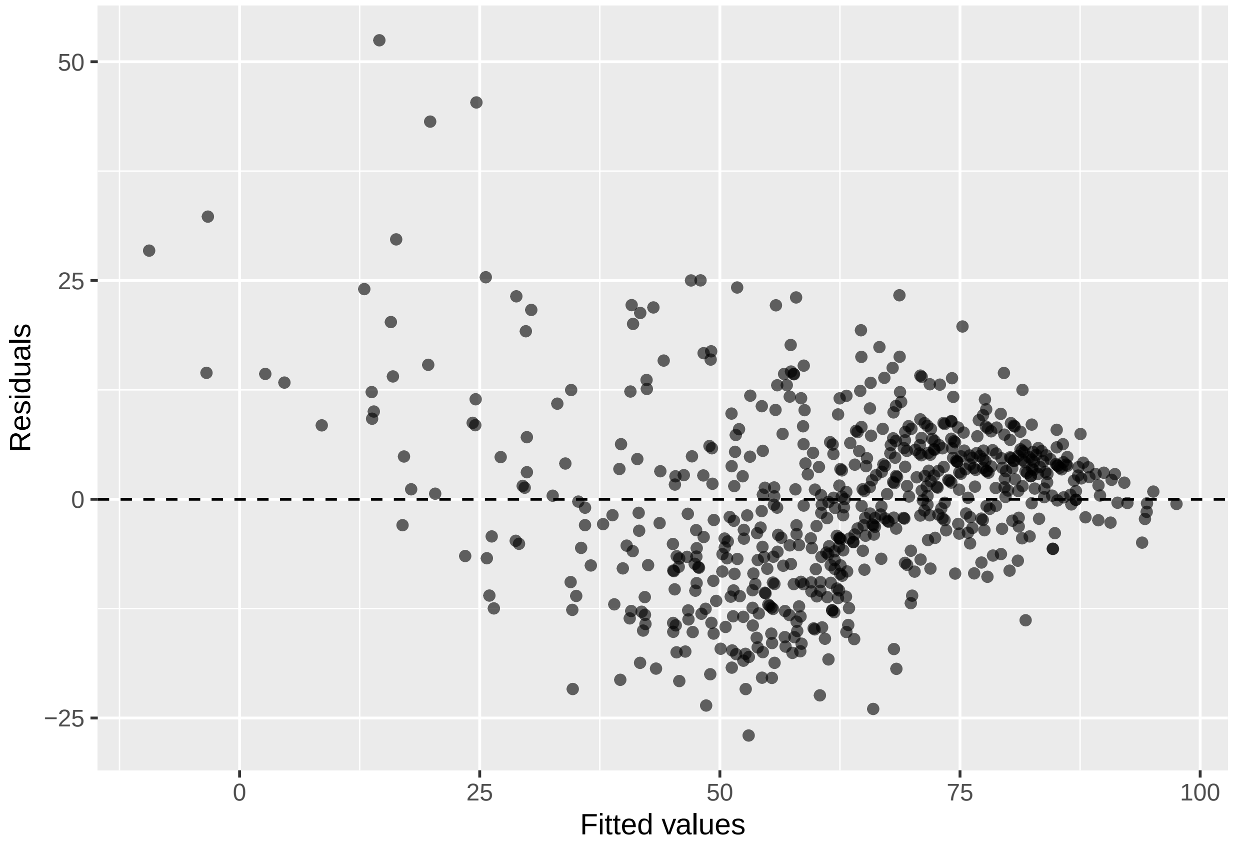
Here we use the augment function in the broom package to be handy here since it takes in a model object(the output of an lm) and returns a data frame with columns corresponding to variables in the model as well as predictive values(.fitted), residuals(.resid), and standardized residuals(.std.resid), along with a few others.

```
m_movies_aug <- augment(m_movie_nompaa)
```

Part 1: Linearity and Constant Variance Since we've already checked the relationship between audience score and the three predictors before, we need to verify this condition with a plot of residuals vs. fitted (predicted) values.

```
# we set the alpha level of our points to a value lower than 1 (0.6 precisely) to add the plot with som
```

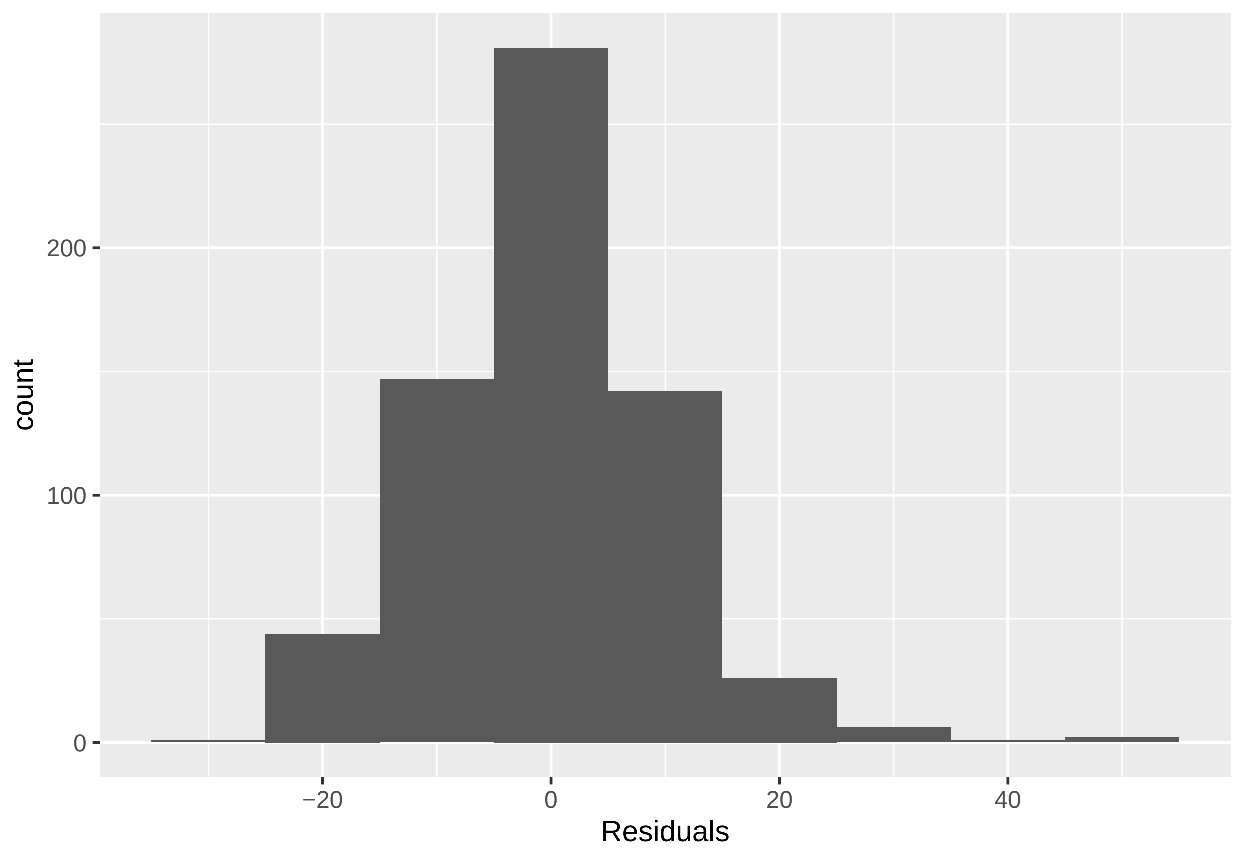
```r
ggplot(data = m_movies_aug, aes( x = .fitted, y = .resid)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Fitted values", y = "Residuals")
```



From the above plot, we can tell that the random scatter is around 0, indicating that this condition fits this model.

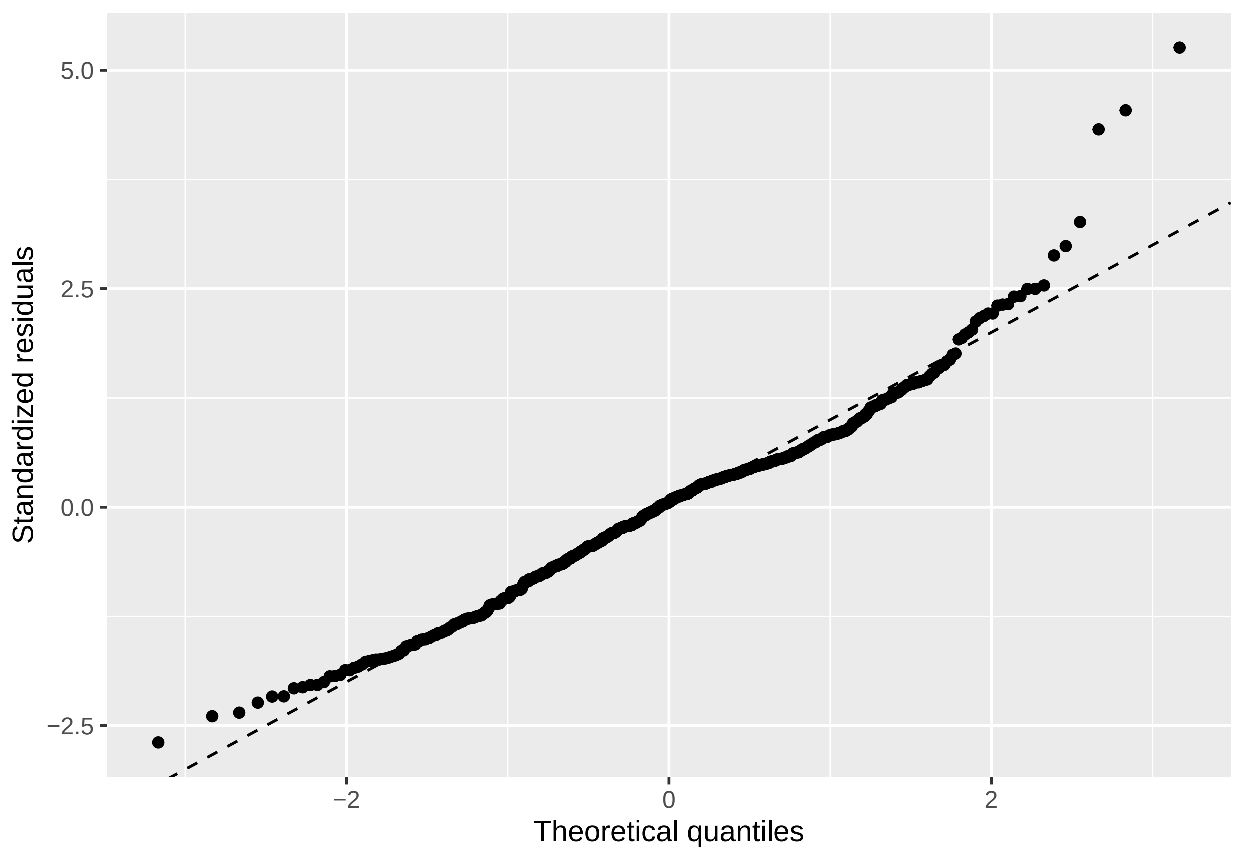
Part 2: Normality To check this condition, we can look at the histogram of residuals.

```r
ggplot(data = m_movies_aug, aes(x = .resid)) +
  geom_histogram(binwidth = 10
                 ) +
  xlab("Residuals")
```



Here the the residuals are nearly normal with mean 0, indicating this condition is met in our model as well. Also we can use a normal probability plot of residuals to verify this conclusion.

```r
ggplot(m_movies_aug)+
  geom_qq(aes(sample = .std.resid)) +
  geom_abline(slope = 1, intercept = 0, linetype ="dashed") +
  labs(x = "Theoretical quantiles", y = "Standardized residuals")
```



According to the plot above, we can tell the normal probability of residuals is also satisfied with the linear condition, denoting that our model is met with this requirement.

Part 5: Prediction

Step 1 Model Choice

Now we will be using Bayesian Predictive distribution for predictions and interpretation of prediction. First, let's find the predictive values under the Best Predictive Model(BPM), which has predictions closest to BMA and corresponding to posterior standard deviations.

```
BPM_pred_movies <- predict(bma_audiencescore, estimator = "BPM", se.fit = TRUE)
variable.names((BPM_pred_movies))
```

```
## [1] "Intercept"     "runtime"       "imdb_rating"   "critics_score"
```

Here we use the function "variable.names" to extract all the predictions in the Best Predictive Model. As expected, this result is the same as the estimation we used in the BIC method before.

Next, we want to identify the variables in the Highest Probability Model (HPM).

```r
HPM_pre_movies <- predict(bma_audiencescore, estimator = "HPM")
variable.names(HPM_pre_movies)
```

```
## [1] "Intercept"     "runtime"       "imdb_rating"   "critics_score"
```

Still the variables in the Highest Probability Model is the same as the ones in Best Predictive Model.

What about the Median Probability Model(MPM)?

```r
MPM_pre_movies <-predict(bma_audiencescore, estimator = "MPM")
```

```
## Warning in bas.lm(eval(object$call$formula), data = eval(object$call$data, :
## dropping 1 rows due to missing data
```

```r
variable.names(MPM_pre_movies)
```

```
## [1] "Intercept"     "imdb_rating"   "critics_score"
```

Here we can see in this Median Probability Model it drooped runtime which distinguished it form HPM and BPM.

Now let us to examine what characteristics lead to the highest audience score in the BPM model.

```r
opt <- which.max(BPM_pred_movies$fit)
movies %>%
  slice(opt) %>%
  glimpse()
```

```
## Rows: 1
## Columns: 37
## $ title            <chr> "The Saddest Music in the World"
## $ title_type       <fct> Feature Film
## $ genre            <fct> Drama
## $ runtime          <dbl> 100
## $ mpaa_rating      <fct> R
## $ studio           <fct> IFC Films
## $ thtr_rel_year    <dbl> 2004
## $ thtr_rel_month   <dbl> 4
## $ thtr_rel_day     <dbl> 30
## $ dvd_rel_year     <dbl> 2004
## $ dvd_rel_month    <dbl> 11
## $ dvd_rel_day      <dbl> 16
## $ imdb_rating      <dbl> 7.2
## $ imdb_num_votes   <int> 5002
## $ critics_rating   <fct> Certified Fresh
## $ critics_score    <dbl> 78
## $ audience_rating  <fct> Upright
## $ audience_score   <dbl> 79
## $ best_pic_nom     <fct> no
## $ best_pic_win     <fct> no
## $ best_actor_win   <fct> no
## $ best_actress_win <fct> no
## $ best_dir_win     <fct> no
## $ top200_box       <fct> no
## $ director         <chr> "Guy Maddin"
## $ actor1           <chr> "Isabella Rossellini"
## $ actor2           <chr> "Mark McKinney"
## $ actor3           <chr> "Maria de Medeiros"
## $ actor4           <chr> "Ross McMillan"
## $ actor5           <chr> "David Fox"
## $ imdb_url         <chr> "http://www.imdb.com/title/tt0366996/"
## $ rt_url           <chr> "//www.rottentomatoes.com/m/saddest_music_in_the_w...
## $ feature_film     <chr> "yes"
## $ drama            <chr> "yes"
## $ mpaa_rating_R    <chr> "yes"
## $ oscar_season     <chr> "no"
## $ summer_season    <chr> "no"
```

From the output above, we now know which specific characteristics lead to the highest audience score. We also might want to explore a 95% credible interval for predicting the audience score using the BPM.

```
ci_movies <- confint(BPM_pred_movies, parm = "pred")
ci_movies[opt,]
```

```
##      2.5%     97.5%      pred
##  77.42102 117.65500  97.53801
```

This output tells us that this BPM yields a 95% credible interval of audience score between 77.42 to 117.66 for the proportion of all audience scores on Rotten Tomatoes about movies produced and released before 2016.

Step 2: Coefficient Interpretation

Now that we've decided on our BPM, it's time to get down to its coefficients and make a closer interpretation of their true meaning in the context of data.

```
linear_bpm = lm(audience_score ~ runtime + imdb_rating + critics_score, data = movies)
summary(linear_bpm)
```

```
##
## Call:
## lm(formula = audience_score ~ runtime + imdb_rating + critics_score,
##     data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.998  -6.565   0.557   5.475  52.448
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -33.28321    3.21939 -10.338  < 2e-16 ***
## runtime        -0.05362    0.02107  -2.545  0.01117 *
## imdb_rating    14.98076    0.57735  25.947  < 2e-16 ***
## critics_score   0.07036    0.02156   3.263  0.00116 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.04 on 646 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7549, Adjusted R-squared:  0.7538
## F-statistic: 663.3 on 3 and 646 DF,  p-value: < 2.2e-16
```

So the BPM of audience score is: audience score = -33.28 - 0.05* runtime + 14.98* imdb_rating + 0.07*critics_score. This result indicates that with all being equal, one additional minute in the movie would lead the audience score to decrease by 0.05, and one increased rating score on IMDB and critics score on Rotten Tomatoes would cause an additional 14.98 increase as well as a 0.07 increase on the audience score on Rotten Tomatoes.

Part 6: Conclusion

In this project, we conduct a Best Predictive Model through some standard methods used in Bayesian inferences, such as BIC and BMA, then make diagnostics of the model combined with the knowledge we have learned from the last course of this specialization: Linear Regression and modeling. The result quite fits our common sense - before going to the theater or rating a movie we have watched, we often refer to others' rating scores and opinions from different channels, and some long-runtime films seem a bit annoying in our eyes.

However, there's a shortcoming in this analysis process: we haven't carried out a detailed analysis of the outliers in this data set, which might influence our final results. All in all, this model and the whole project analysis process are of great reference value for people to find out how much audiences and critics like the movies and the relationship between different variables and the audience scores of films.