Movie Popularity Prediction

What makes a movie popular? The question is important for movie executives, online marketers, and social media companies, all who have a stake in promoting and partnering with the top movies of a season. In this project, I predict audience score based on a variety of variables about a random sample of movies from Rotten Tomatoes and IMDB.

Part 1: Data

The data is a random sample of 651 movies released before 2016. With all requirements of randomization being justified, we can assume that this is representative of the population of movies that have Rotten Tomatoes and IMBD pages. This means that we can better make arguments about generalizability (only to this specific population of movies released before 2016, not after, through), and to association, but not causality. If we were interested in causality, we might want to amp up the methods and pursue something like matching, which could control for a variety of covariates in the model.

Part 2: Data manipulation

First, I mutate, or create, new variables based on features that make more sense for the question at hand.

```
library(tidyverse)

movies <- movies %>%
  mutate(feature_film = ifelse(title_type == "Feature Film", "yes", "no"),
         drama = ifelse(genre== "Drama", "yes", "no"),
         mpaa_rating_R = ifelse(mpaa_rating == "R", "yes", "no"),
         oscar_season = ifelse(thtr_rel_month %in% c(10, 11,12), "yes", "no"),
         summer_season = ifelse(thtr_rel_month %in% c(5,6,7,8), "yes", "no"))
```

Part 3: Exploratory data analysis

Audience score is normally distributed, and this normality holds for feature films. For non feature films, however, the audience score is more bi-modal, indicating that they are usually either “good” or “bad”. The same observation is reflected in the five number summary, where feature films have a lower minimum and mean value than non-feature films. A simmilar trend holds for drama and non-drama films, where audience score for drama films is normally distributed, but for non-drama films, is bimodal, reflecting that the tendancy might be for people to rate them as either good or bad. On the other hand, R rated films are the ones that are bi-modal, where as not-R rated films are normal.

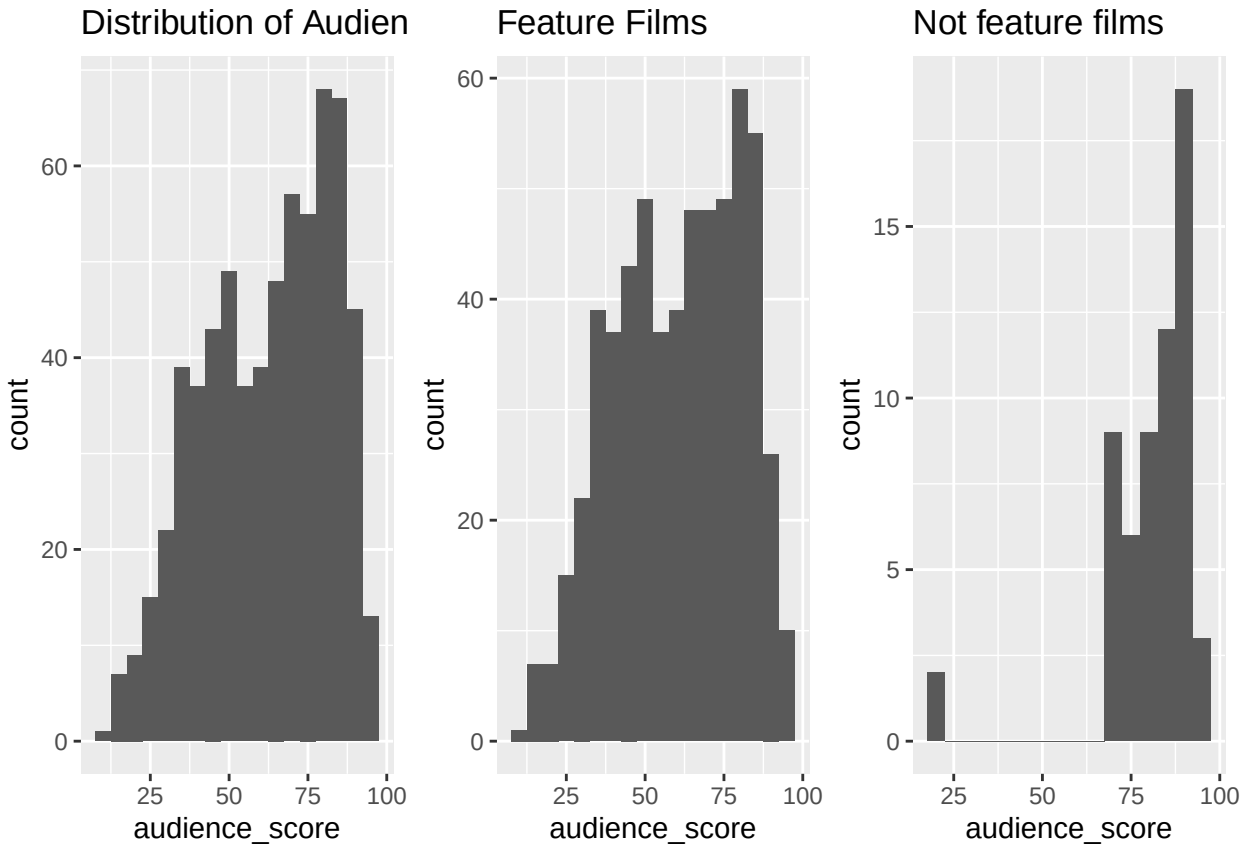
The numerical variable of critics score is expectadely linearly associated, though not too strong, with audience score. The correlation between critics score and audience score is 0.70, which is moderate to strong, so we can certainly use this as a predictor. As for the collinearlity among features in the dat set, imdb_score is highly correlated with critics score, at 0.76, so we might not want to keep both of these variables in the set.

```
# distribution of outcome

p1 <- ggplot(data = movies, aes(x = audience_score)) +
  geom_histogram(binwidth = 5) +
```

```
  ggtitle("Distribution of Audience Score for all Films")

#distribution of outcome for feature films
feature_films = filter(movies, feature_film == 'yes')
p2 <- ggplot(data = feature_films, aes(x = audience_score)) +
  geom_histogram(binwidth = 5) +
  ggtitle("Feature Films")
not_feature_films = filter(movies, feature_film == 'no')
p3 <- ggplot(data = not_feature_films, aes(x = audience_score)) +
  geom_histogram(binwidth = 5) +
  ggtitle("Not feature films")
grid.arrange(p1, p2, p3, nrow = 1)
```




```
summary(feature_films$audience_score)
```

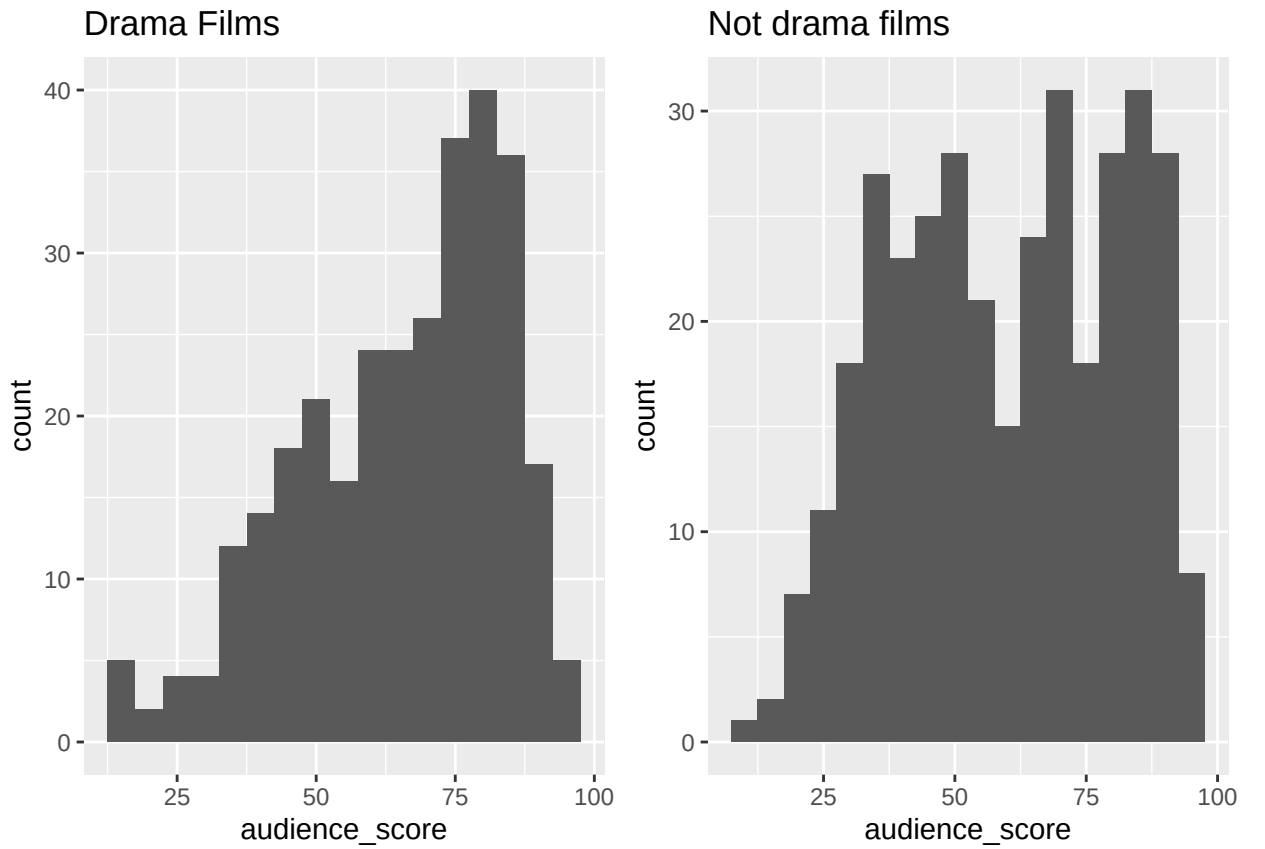
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   44.50   62.00   60.47   78.00   97.00
```

```
summary(not_feature_films$audience_score)
```

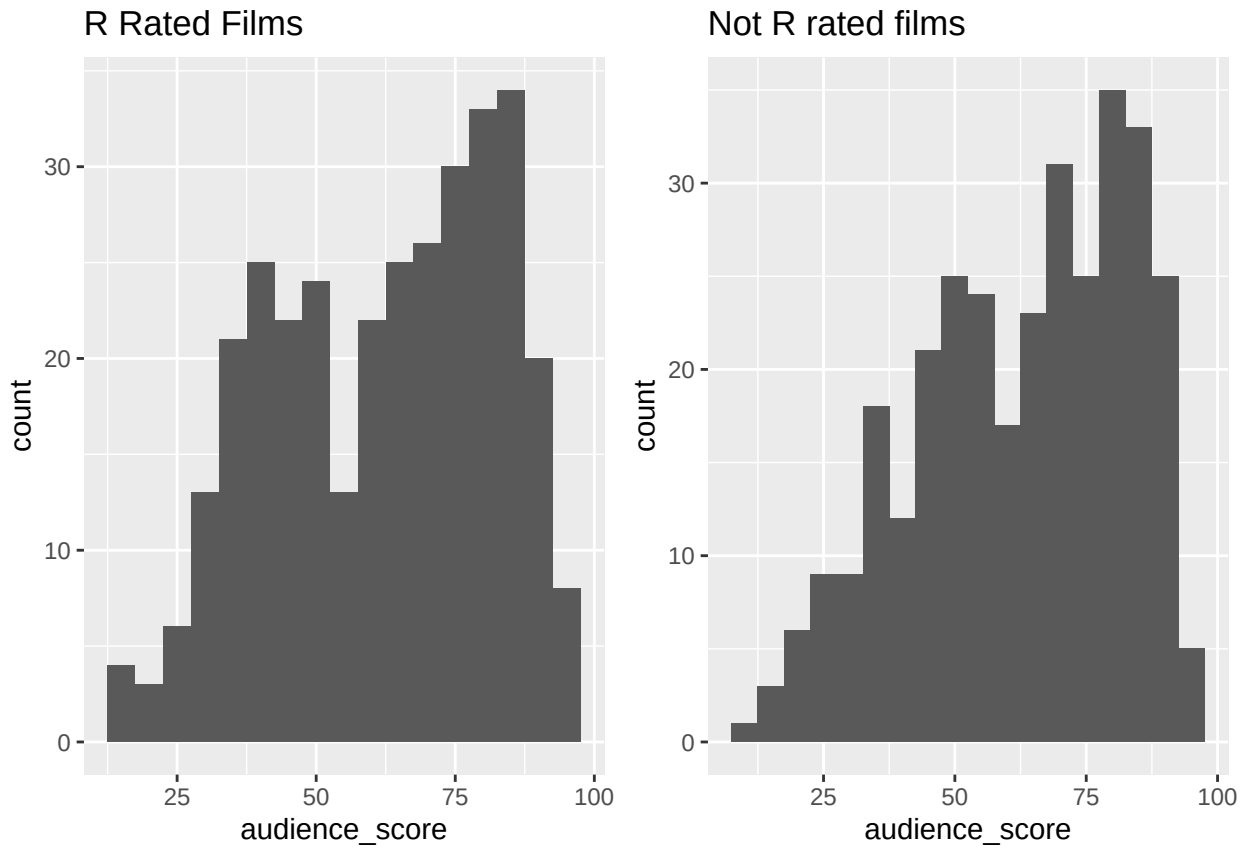
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   76.50   85.50   81.05   89.00   96.00
```

```
#distribution of outcome for dramas
drama_films = filter(movies, drama == 'yes')
p1 <- ggplot(data = drama_films, aes(x = audience_score)) +
  geom_histogram(binwidth = 5) +
  ggtitle("Drama Films")
```

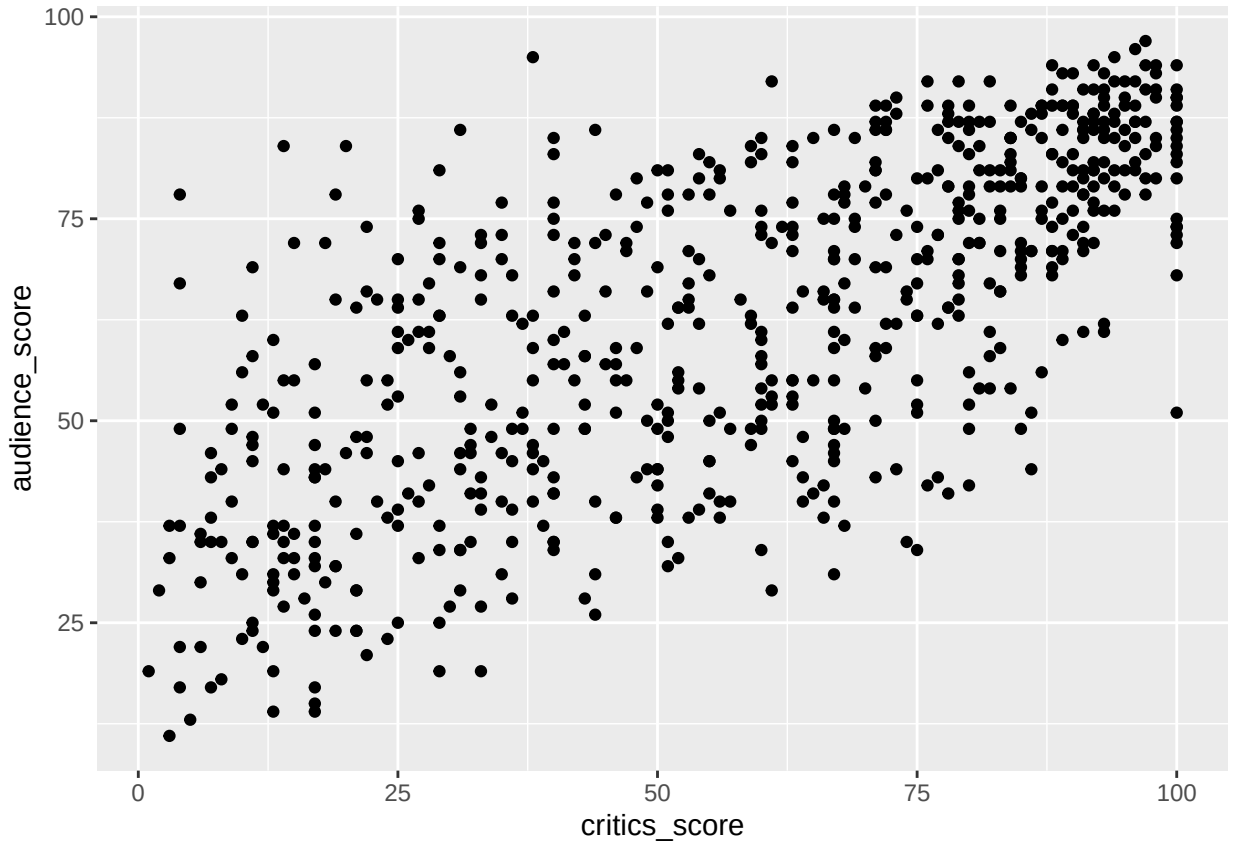
```
not_drama_films = filter(movies, drama == 'no')
p2 <- ggplot(data = not_drama_films, aes(x = audience_score)) +
  geom_histogram(binwidth = 5) +
  ggtitle("Not drama films")
grid.arrange(p1, p2, nrow = 1)
```




```
#distribution of outcome for R rated films
R_rated_films = filter(movies, mpaa_rating_R== 'yes')
p1 <- ggplot(data = R_rated_films, aes(x = audience_score)) +
  geom_histogram(binwidth = 5) +
  ggtitle("R Rated Films")
not_R_rated_films = filter(movies, mpaa_rating_R == 'no')
p2 <- ggplot(data = not_R_rated_films, aes(x = audience_score)) +
  geom_histogram(binwidth = 5) +
  ggtitle("Not R rated films")
grid.arrange(p1, p2, nrow = 1)
```




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




```
cor(movies$audience_score, movies$critics_score)
```

```
## [1] 0.7042762
```

```
numericals <- movies %>%
  dplyr::select(audience_score, runtime, thtr_rel_year,
         imdb_rating, imdb_num_votes, critics_score) %>%
  na.omit()
numericals$runtime <- as.numeric(numericals$runtime)
cor(numericals)
```

```
##                audience_score    runtime thtr_rel_year imdb_rating
## audience_score     1.00000000  0.1809629   -0.05479688  0.86490913
## runtime            0.18096290  1.0000000   -0.10437672  0.26824048
## thtr_rel_year     -0.05479688 -0.1043767    1.00000000 -0.03135507
## imdb_rating        0.86490913  0.2682405   -0.03135507  1.00000000
## imdb_num_votes     0.29029291  0.3472149    0.15675364  0.33216314
## critics_score      0.70415734  0.1724989   -0.08198620  0.76478324
##                imdb_num_votes critics_score
## audience_score      0.2902929     0.7041573
## runtime             0.3472149     0.1724989
## thtr_rel_year       0.1567536    -0.0819862
## imdb_rating         0.3321631     0.7647832
## imdb_num_votes      1.0000000     0.2100053
## critics_score       0.2100053     1.0000000
```

Part 4: Modeling

From the full model, we want to take out oscar season and top 200 box_yes. This has the effect of reducing the BIC from 4934 to 4922. We can continue removing predictor variables until we find the lowest BIC, which in this case would be a BIC of 4872, in the most parsmionious model that has only two predictors, which are imbd_rating and critics_score. Then, using the step AIC, I evaluate the model with the lowest AIC value, which has a BIC of 4890, and includes the variables of critics_score and imbd_rating, but additionally has the variables of R rating and best_pic_no.

Using this model, I evaluate the marginal posterior inclusion probabilities for each variable. The probability of runtime is high at .48, and the others might not seem high, but are higher than the probability of such predictors without a model.As an additional note, model assumptions, such as normality, are met through the analysis.

```
## develop a Bayesian regression model to predict audience score from the explanatory variables
## create a small data set
movies_small <- movies %>%
  dplyr::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)
##full model
mod_full <- lm(audience_score ~ . - audience_score, data = movies_small)
summary(mod_full)
```

```
##
## Call:
## lm(formula = audience_score ~ . - audience_score, data = movies_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.594  -6.156   0.157   5.909  53.125
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.244e+02  7.749e+01   1.606  0.10886
## feature_filmyes      -2.248e+00  1.687e+00  -1.332  0.18323
## dramayes              1.292e+00  8.766e-01   1.474  0.14087
## runtime              -5.614e-02  2.415e-02  -2.324  0.02042 *
## mpaa_rating_Ryes     -1.444e+00  8.127e-01  -1.777  0.07598 .
## thtr_rel_year        -7.657e-02  3.835e-02  -1.997  0.04628 *
## oscar_seasonyes      -5.333e-01  9.967e-01  -0.535  0.59280
## summer_seasonyes      9.106e-01  9.493e-01   0.959  0.33778
## imdb_rating           1.472e+01  6.067e-01  24.258  < 2e-16 ***
## imdb_num_votes        7.234e-06  4.523e-06   1.600  0.11019
## critics_score         5.748e-02  2.217e-02   2.593  0.00973 **
## best_pic_nomyes       5.321e+00  2.628e+00   2.025  0.04330 *
## best_pic_winyes      -3.212e+00  4.610e+00  -0.697  0.48624
## best_actor_winyes    -1.544e+00  1.179e+00  -1.310  0.19068
## best_actress_winyes -2.198e+00  1.304e+00  -1.686  0.09229 .
## best_dir_winyes      -1.231e+00  1.728e+00  -0.713  0.47630
## top200_boxyes         8.478e-01  2.782e+00   0.305  0.76067
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 9.975 on 633 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.763,  Adjusted R-squared:  0.757
## F-statistic: 127.3 on 16 and 633 DF,  p-value: < 2.2e-16
```

```
## complete Bayesian model selection
BIC(mod_full)
```

```
## [1] 4934.145
```

```
#reduced model 1
mod_reduced <- lm(audience_score ~ . - oscar_season - top200_box, data = movies_small)
summary(mod_reduced)
```

```
##
## Call:
## lm(formula = audience_score ~ . - oscar_season - top200_box,
##     data = movies_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.629  -6.023   0.186   5.919  53.334
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.261e+02  7.690e+01   1.640  0.10159
## feature_filmyes     -2.277e+00  1.684e+00  -1.352  0.17673
## dramayes             1.307e+00  8.739e-01   1.496  0.13515
## runtime             -5.796e-02  2.382e-02  -2.433  0.01527 *
## mpaa_rating_Ryes    -1.469e+00  8.068e-01  -1.820  0.06919 .
## thtr_rel_year       -7.739e-02  3.805e-02  -2.034  0.04241 *
## summer_seasonyes     1.137e+00  8.523e-01   1.334  0.18261
## imdb_rating          1.471e+01  6.053e-01  24.299  < 2e-16 ***
## imdb_num_votes       7.601e-06  4.355e-06   1.746  0.08137 .
## critics_score        5.780e-02  2.210e-02   2.615  0.00912 **
## best_pic_nomyes      5.148e+00  2.608e+00   1.974  0.04879 *
## best_pic_winyes     -3.123e+00  4.600e+00  -0.679  0.49746
## best_actor_winyes   -1.559e+00  1.176e+00  -1.325  0.18551
## best_actress_winyes -2.191e+00  1.301e+00  -1.684  0.09272 .
## best_dir_winyes     -1.290e+00  1.723e+00  -0.749  0.45417
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.963 on 635 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7628, Adjusted R-squared:  0.7576
## F-statistic: 145.9 on 14 and 635 DF,  p-value: < 2.2e-16
```

```
BIC(mod_reduced)
```

```
## [1] 4921.56
```

```
#reduced model 2
mod_reduced2 <- lm(audience_score ~ . - oscar_season - top200_box
                   - best_dir_win - summer_season - imdb_num_votes, data = movies_small)
summary(mod_reduced2)
```

```
##
## Call:
## lm(formula = audience_score ~ . - oscar_season - top200_box -
##     best_dir_win - summer_season - imdb_num_votes, data = movies_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.089  -6.420   0.210   5.561  53.111
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         81.64181   73.46837   1.111   0.2669
## feature_filmyes     -1.32595    1.60780  -0.825   0.4099
## dramayes             0.99183    0.86058   1.153   0.2495
## runtime             -0.05172    0.02298  -2.251   0.0247 *
## mpaa_rating_Ryes    -1.49611    0.80746  -1.853   0.0644 .
## thtr_rel_year       -0.05619    0.03655  -1.537   0.1247
## imdb_rating         14.91642    0.58469  25.512   <2e-16 ***
## critics_score        0.05977    0.02197   2.720   0.0067 **
## best_pic_nomyes      5.63083    2.57887   2.183   0.0294 *
## best_pic_winyes     -2.45019    4.34798  -0.564   0.5733
## best_actor_winyes   -1.76505    1.17413  -1.503   0.1333
## best_actress_winyes -2.15604    1.30342  -1.654   0.0986 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.981 on 638 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7608, Adjusted R-squared:  0.7567
## F-statistic: 184.5 on 11 and 638 DF,  p-value: < 2.2e-16
```

```
BIC(mod_reduced2)
```

```
## [1] 4907.624
```

```
#reduced model 3
mod_reduced3 <- lm(audience_score ~ . - oscar_season - top200_box
                   - best_dir_win - summer_season - imdb_num_votes
                   - best_actor_win - best_actress_win, data = movies_small)
summary(mod_reduced3)
```

```
##
## Call:
## lm(formula = audience_score ~ . - oscar_season - top200_box -
##     best_dir_win - summer_season - imdb_num_votes - best_actor_win -
##     best_actress_win, data = movies_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.613  -6.370   0.194   5.597  53.300
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      79.64397   73.65072   1.081  0.27994
## feature_filmyes  -1.54381    1.60892  -0.960  0.33765
```

```
## dramayes          0.85113    0.85993   0.990  0.32266
## runtime          -0.06388    0.02237  -2.856  0.00443 **
## mpaa_rating_Ryes -1.41069    0.80873  -1.744  0.08158 .
## thtr_rel_year    -0.05469    0.03664  -1.493  0.13597
## imdb_rating      14.92839    0.58620  25.466  < 2e-16 ***
## critics_score     0.05925    0.02203   2.690  0.00734 **
## best_pic_nomyes   4.70535    2.55427   1.842  0.06592 .
## best_pic_winyes  -2.42024    4.34305  -0.557  0.57754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.01 on 640 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7588, Adjusted R-squared:  0.7554
## F-statistic: 223.7 on 9 and 640 DF,  p-value: < 2.2e-16
```

```
BIC(mod_reduced3)
```

```
## [1] 4900.109
```

```
#reduced model 4
mod_reduced4 <- lm(audience_score ~ . - oscar_season - top200_box
                   - best_dir_win - summer_season - imdb_num_votes
                   - best_actor_win - best_actress_win - feature_film
                   - drama - runtime - best_pic_nom - best_pic_win - thtr_rel_year
                   -mpaa_rating_R
                   , data = movies_small)
summary(mod_reduced4)
```

```
##
## Call:
## lm(formula = audience_score ~ . - oscar_season - top200_box -
##     best_dir_win - summer_season - imdb_num_votes - best_actor_win -
##     best_actress_win - feature_film - drama - runtime - best_pic_nom -
##     best_pic_win - thtr_rel_year - mpaa_rating_R, data = movies_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.673  -6.770   0.719   5.499  52.454
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -37.07462    2.86593 -12.936  < 2e-16 ***
## imdb_rating    14.66518    0.56626  25.898  < 2e-16 ***
## critics_score   0.07324    0.02162   3.387 0.000748 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.08 on 647 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7525, Adjusted R-squared:  0.7517
## F-statistic: 983.3 on 2 and 647 DF,  p-value: < 2.2e-16
```

```
BIC(mod_reduced4)
```

```
## [1] 4871.629
```

```r
## now through stepAIC
stepAIC(mod_full)
```

```
## Start:  AIC=3006.94
## 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) -
##     audience_score
##
##                     Df Sum of Sq    RSS    AIC
## - top200_box         1         9  62999 3005.0
## - oscar_season       1        28  63018 3005.2
## - best_pic_win       1        48  63038 3005.4
## - best_dir_win       1        51  63040 3005.5
## - summer_season      1        92  63081 3005.9
## - best_actor_win     1       171  63160 3006.7
## - feature_film       1       177  63166 3006.8
## <none>                           62990 3006.9
## - drama              1       216  63206 3007.2
## - imdb_num_votes     1       255  63244 3007.6
## - best_actress_win   1       283  63273 3007.9
## - mpaa_rating_R      1       314  63304 3008.2
## - thtr_rel_year      1       397  63386 3009.0
## - best_pic_nom       1       408  63398 3009.1
## - runtime            1       538  63527 3010.5
## - critics_score      1       669  63659 3011.8
## - imdb_rating        1     58556 121545 3432.2
##
## Step:  AIC=3005.04
## 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
##
##                     Df Sum of Sq    RSS    AIC
## - oscar_season       1        26  63025 3003.3
## - best_pic_win       1        49  63047 3003.5
## - best_dir_win       1        52  63051 3003.6
## - summer_season      1        94  63093 3004.0
## - best_actor_win     1       169  63168 3004.8
## - feature_film       1       176  63175 3004.8
## <none>                           62999 3005.0
## - drama              1       214  63213 3005.2
## - best_actress_win   1       279  63278 3005.9
## - imdb_num_votes     1       302  63301 3006.1
## - mpaa_rating_R      1       330  63329 3006.4
## - best_pic_nom       1       404  63403 3007.2
## - thtr_rel_year      1       415  63414 3007.3
## - runtime            1       535  63534 3008.5
## - critics_score      1       681  63680 3010.0
## - imdb_rating        1     58606 121604 3430.5
##
## Step:  AIC=3003.31
```

```
## audience_score ~ feature_film + drama + runtime + mpaa_rating_R +
##     thtr_rel_year + summer_season + imdb_rating + imdb_num_votes +
##     critics_score + best_pic_nom + best_pic_win + best_actor_win +
##     best_actress_win + best_dir_win
##
##                     Df Sum of Sq    RSS    AIC
## - best_pic_win       1        46  63071 3001.8
## - best_dir_win       1        56  63081 3001.9
## - best_actor_win     1       174  63200 3003.1
## - summer_season      1       177  63202 3003.1
## - feature_film       1       182  63207 3003.2
## <none>                            63025 3003.3
## - drama              1       222  63247 3003.6
## - best_actress_win   1       281  63307 3004.2
## - imdb_num_votes     1       302  63328 3004.4
## - mpaa_rating_R      1       329  63354 3004.7
## - best_pic_nom       1       387  63412 3005.3
## - thtr_rel_year      1       410  63436 3005.5
## - runtime            1       587  63613 3007.3
## - critics_score      1       679  63704 3008.3
## - imdb_rating        1     58603 121628 3428.6
##
## Step:  AIC=3001.78
## audience_score ~ feature_film + drama + runtime + mpaa_rating_R +
##     thtr_rel_year + summer_season + imdb_rating + imdb_num_votes +
##     critics_score + best_pic_nom + best_actor_win + best_actress_win +
##     best_dir_win
##
##                     Df Sum of Sq    RSS    AIC
## - best_dir_win       1        94  63165 3000.7
## - best_actor_win     1       163  63234 3001.5
## - feature_film       1       171  63242 3001.5
## - summer_season      1       174  63245 3001.6
## <none>                            63071 3001.8
## - drama              1       220  63291 3002.0
## - imdb_num_votes     1       271  63342 3002.6
## - best_actress_win   1       294  63365 3002.8
## - mpaa_rating_R      1       330  63401 3003.2
## - best_pic_nom       1       342  63414 3003.3
## - thtr_rel_year      1       397  63468 3003.9
## - runtime            1       586  63657 3005.8
## - critics_score      1       680  63751 3006.8
## - imdb_rating        1     58858 121929 3428.2
##
## Step:  AIC=3000.75
## audience_score ~ feature_film + drama + runtime + mpaa_rating_R +
##     thtr_rel_year + summer_season + imdb_rating + imdb_num_votes +
##     critics_score + best_pic_nom + best_actor_win + best_actress_win
##
##                     Df Sum of Sq    RSS    AIC
## - summer_season      1       167  63332 3000.5
## - best_actor_win     1       171  63336 3000.5
## - feature_film       1       183  63348 3000.6
## <none>                            63165 3000.7
```

```
## - drama             1       228  63394 3001.1
## - imdb_num_votes    1       247  63412 3001.3
## - best_actress_win  1       299  63464 3001.8
## - best_pic_nom      1       326  63491 3002.1
## - mpaa_rating_R     1       345  63510 3002.3
## - thtr_rel_year     1       368  63533 3002.5
## - critics_score     1       651  63816 3005.4
## - runtime           1       673  63839 3005.6
## - imdb_rating       1     58895 122061 3426.9
##
## Step:  AIC=3000.46
## audience_score ~ feature_film + drama + runtime + mpaa_rating_R +
##     thtr_rel_year + imdb_rating + imdb_num_votes + critics_score +
##     best_pic_nom + best_actor_win + best_actress_win
##
##                    Df Sum of Sq    RSS    AIC
## - feature_film      1       156  63488 3000.1
## <none>                          63332 3000.5
## - best_actor_win    1       195  63527 3000.5
## - drama             1       204  63536 3000.6
## - imdb_num_votes    1       260  63592 3001.1
## - best_pic_nom      1       297  63629 3001.5
## - best_actress_win  1       297  63629 3001.5
## - mpaa_rating_R     1       356  63688 3002.1
## - thtr_rel_year     1       361  63693 3002.2
## - runtime           1       690  64022 3005.5
## - critics_score     1       732  64064 3005.9
## - imdb_rating       1     58763 122095 3425.1
##
## Step:  AIC=3000.06
## audience_score ~ drama + runtime + mpaa_rating_R + thtr_rel_year +
##     imdb_rating + imdb_num_votes + critics_score + best_pic_nom +
##     best_actor_win + best_actress_win
##
##                    Df Sum of Sq    RSS    AIC
## - drama             1       121  63609 2999.3
## - imdb_num_votes    1       173  63661 2999.8
## <none>                          63488 3000.1
## - best_actor_win    1       219  63706 3000.3
## - thtr_rel_year     1       277  63765 3000.9
## - best_pic_nom      1       291  63778 3001.0
## - best_actress_win  1       306  63794 3001.2
## - mpaa_rating_R     1       453  63941 3002.7
## - runtime           1       715  64203 3005.3
## - critics_score     1       875  64363 3007.0
## - imdb_rating       1     63189 126677 3447.1
##
## Step:  AIC=2999.3
## audience_score ~ runtime + mpaa_rating_R + thtr_rel_year + imdb_rating +
##     imdb_num_votes + critics_score + best_pic_nom + best_actor_win +
##     best_actress_win
##
##                    Df Sum of Sq    RSS    AIC
## - imdb_num_votes    1       148  63757 2998.8
```

```
## <none>                           63609 2999.3
## - best_actor_win    1       209  63818 2999.4
## - thtr_rel_year     1       272  63881 3000.1
## - best_actress_win  1       274  63883 3000.1
## - best_pic_nom      1       307  63916 3000.4
## - mpaa_rating_R     1       391  64000 3001.3
## - runtime           1       631  64240 3003.7
## - critics_score     1       916  64525 3006.6
## - imdb_rating       1     63434 127043 3447.0
##
## Step:  AIC=2998.81
## audience_score ~ runtime + mpaa_rating_R + thtr_rel_year + imdb_rating +
##     critics_score + best_pic_nom + best_actor_win + best_actress_win
##
##                    Df Sum of Sq    RSS    AIC
## <none>                           63757 2998.8
## - thtr_rel_year     1       201  63958 2998.9
## - best_actor_win    1       219  63976 2999.0
## - best_actress_win  1       266  64023 2999.5
## - mpaa_rating_R     1       367  64124 3000.5
## - best_pic_nom      1       442  64199 3001.3
## - runtime           1       519  64276 3002.1
## - critics_score     1       879  64635 3005.7
## - imdb_rating       1     67356 131113 3465.4
##
## Call:
## lm(formula = audience_score ~ runtime + mpaa_rating_R + thtr_rel_year +
##     imdb_rating + critics_score + best_pic_nom + best_actor_win +
##     best_actress_win, data = movies_small)
##
## Coefficients:
##         (Intercept)              runtime     mpaa_rating_Ryes
##            70.10675             -0.05116             -1.50528
##       thtr_rel_year          imdb_rating        critics_score
##            -0.05123             15.00149              0.06410
##     best_pic_nomyes    best_actor_winyes  best_actress_winyes
##             4.88277             -1.73482             -2.11568
```

```
## with the lowest AIC fit
aic_fit_model <- lm(audience_score ~ runtime + mpaa_rating_R + thtr_rel_year + imdb_rating +
    critics_score + best_pic_nom + best_actor_win + best_actress_win, data = movies_small)
summary(aic_fit_model)
```

```
##
## Call:
## lm(formula = audience_score ~ runtime + mpaa_rating_R + thtr_rel_year +
##     imdb_rating + critics_score + best_pic_nom + best_actor_win +
##     best_actress_win, data = movies_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.613  -6.343   0.178   5.356  52.999
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         70.10675   72.17547   0.971  0.33175
## runtime             -0.05116    0.02239  -2.285  0.02265 *
## mpaa_rating_Ryes    -1.50528    0.78367  -1.921  0.05520 .
## thtr_rel_year       -0.05123    0.03605  -1.421  0.15587
## imdb_rating         15.00149    0.57647  26.023  < 2e-16 ***
## critics_score        0.06410    0.02157   2.972  0.00307 **
## best_pic_nomyes      4.88277    2.31590   2.108  0.03539 *
## best_actor_winyes   -1.73482    1.16824  -1.485  0.13804
## best_actress_winyes -2.11568    1.29452  -1.634  0.10268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.973 on 641 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7601, Adjusted R-squared:  0.7571
## F-statistic: 253.8 on 8 and 641 DF,  p-value: < 2.2e-16
```

```
BIC(aic_fit_model)
```

```
## [1] 4890.199
```

```
##Bayesian model averaging: multiple models are averaged to obtain posteriors of coefficients and predi
# Fit the model using Bayesian linear regression, `bas.lm` function in the `BAS` package
bma_lwage <- bas.lm(audience_score ~ runtime + mpaa_rating_R + thtr_rel_year + imdb_rating +
    critics_score + best_pic_nom + best_actor_win + best_actress_win, data = movies_small,
                    prior = "BIC",
                    modelprior = uniform())
```

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

```
# Print out the marginal posterior inclusion probabilities for each variable
## so, the posterior probability of runtime is high, at 0.47, and crics_score, and best_pic_no
bma_lwage
```

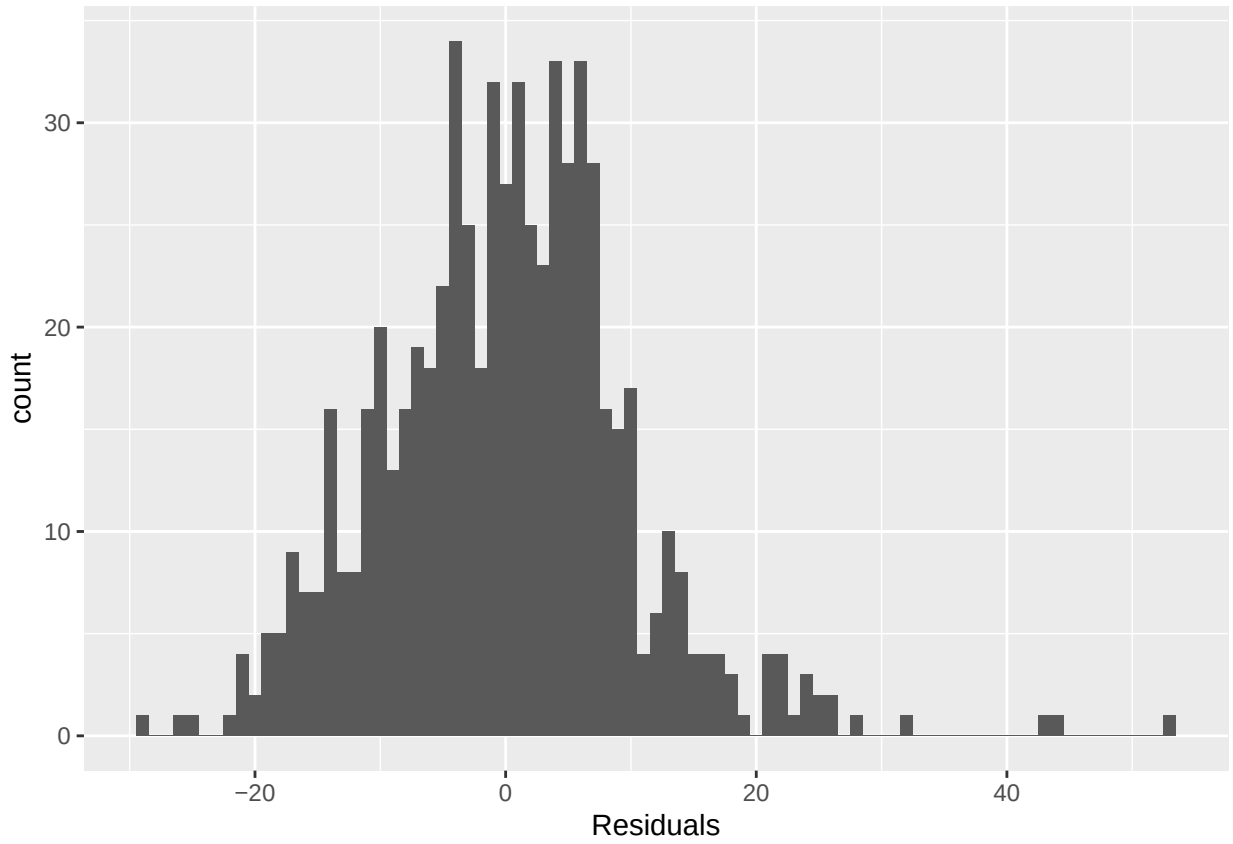
```
##
## Call:
## bas.lm(formula = audience_score ~ runtime + mpaa_rating_R + thtr_rel_year +
##     imdb_rating + critics_score + best_pic_nom + best_actor_win +
##     best_actress_win, data = movies_small, prior = "BIC", modelprior = uniform())
##
##
##  Marginal Posterior Inclusion Probabilities:
##           Intercept              runtime     mpaa_rating_Ryes
##             1.00000              0.48270              0.20316
##       thtr_rel_year          imdb_rating        critics_score
##             0.08617              1.00000              0.89551
##     best_pic_nomyes    best_actor_winyes  best_actress_winyes
##             0.12759              0.14874              0.14191
```

```
# Top 5 most probably models
summary(bma_lwage)
```

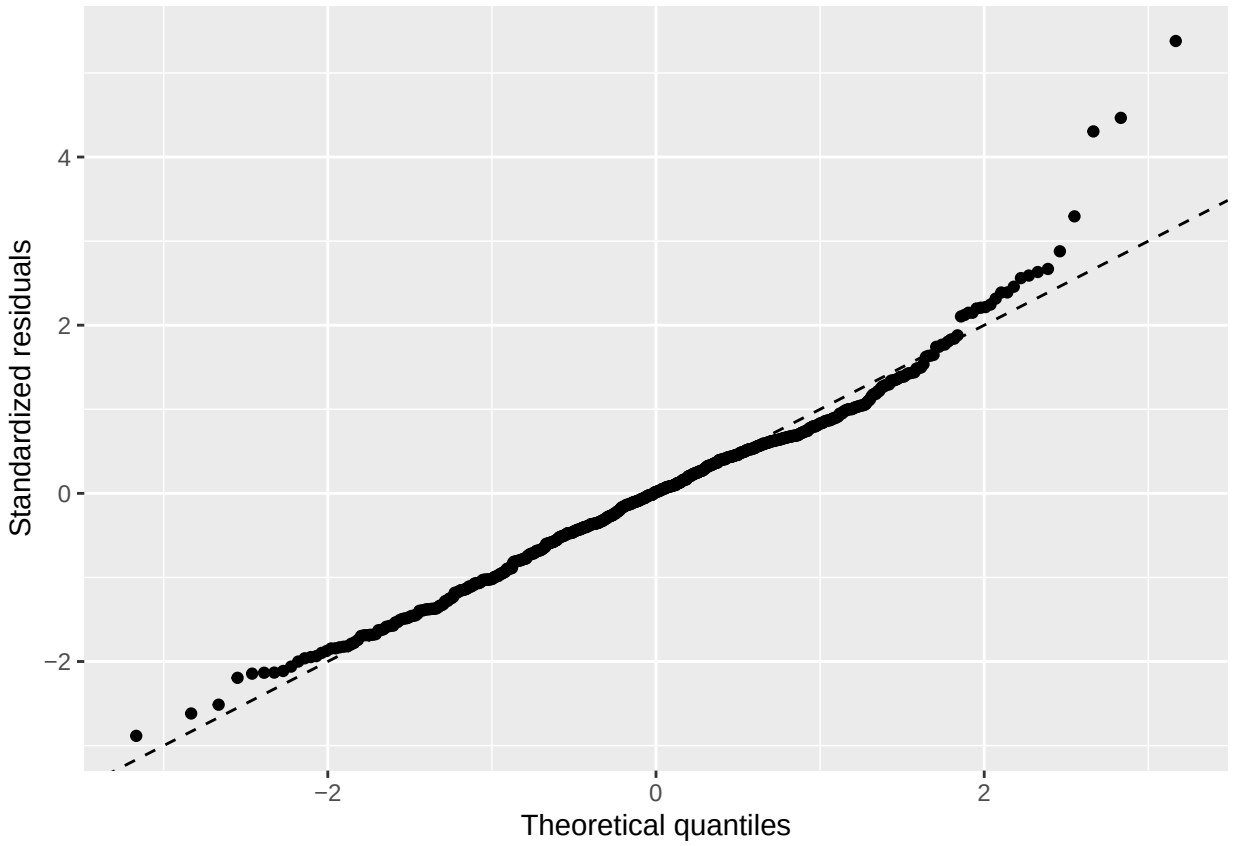
```
##                     P(B != 0 | Y)    model 1       model 2       model 3
## Intercept               1.0000000     1.0000     1.0000000     1.0000000
## runtime                 0.4827044     1.0000     0.0000000     0.0000000
```

```
## mpaa_rating_Ryes        0.2031587     0.0000     0.0000000     0.0000000
## thtr_rel_year           0.0861735     0.0000     0.0000000     0.0000000
## imdb_rating             1.0000000     1.0000     1.0000000     1.0000000
## critics_score           0.8955084     1.0000     1.0000000     1.0000000
## best_pic_nomyes         0.1275921     0.0000     0.0000000     0.0000000
## best_actor_winyes       0.1487434     0.0000     0.0000000     1.0000000
## best_actress_winyes     0.1419059     0.0000     0.0000000     0.0000000
## BF                             NA     1.0000     0.9968489     0.2543185
## PostProbs                      NA     0.2122     0.2116000     0.0540000
## 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
## runtime                0.0000000     1.0000000
## mpaa_rating_Ryes       1.0000000     1.0000000
## thtr_rel_year          0.0000000     0.0000000
## imdb_rating            1.0000000     1.0000000
## critics_score          1.0000000     1.0000000
## best_pic_nomyes        0.0000000     0.0000000
## best_actor_winyes      0.0000000     0.0000000
## best_actress_winyes    0.0000000     0.0000000
## BF                     0.2521327     0.2391994
## PostProbs              0.0535000     0.0508000
## R2                     0.7539000     0.7563000
## dim                    4.0000000     5.0000000
## logmarg            -3616.6568544 -3616.7095127
```

```r
##model diagnostics
library(MASS)
library(tidyverse)
library(statsr)
library(BAS)
library(broom)
mod_full_aug <- augment(mod_full)
ggplot(data = mod_full_aug, aes(x = .resid)) +
  geom_histogram(binwidth = 1) +
  xlab("Residuals")
```




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



Part 4.2: Interpretation of Final Model

In the final model, then a one point increase in imdb_rating is associated with a 15b point increase in audience score. Runtime is negatively associated, where longer movies have less audience score.

```
## with the lowest AIC fit
aic_fit_model <- lm(audience_score ~ runtime + mpaa_rating_R + thtr_rel_year + imdb_rating +
    critics_score + best_pic_nom + best_actor_win + best_actress_win, data = movies_small)
summary(aic_fit_model)
```

```
##
## Call:
## lm(formula = audience_score ~ runtime + mpaa_rating_R + thtr_rel_year +
##     imdb_rating + critics_score + best_pic_nom + best_actor_win +
##     best_actress_win, data = movies_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.613  -6.343   0.178   5.356  52.999
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         70.10675   72.17547   0.971  0.33175
## runtime             -0.05116    0.02239  -2.285  0.02265 *
## mpaa_rating_Ryes    -1.50528    0.78367  -1.921  0.05520 .
## thtr_rel_year       -0.05123    0.03605  -1.421  0.15587
```

```
## imdb_rating         15.00149    0.57647  26.023  < 2e-16 ***
## critics_score        0.06410    0.02157   2.972  0.00307 **
## best_pic_nomyes      4.88277    2.31590   2.108  0.03539 *
## best_actor_winyes   -1.73482    1.16824  -1.485  0.13804
## best_actress_winyes -2.11568    1.29452  -1.634  0.10268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.973 on 641 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7601, Adjusted R-squared:  0.7571
## F-statistic: 253.8 on 8 and 641 DF,  p-value: < 2.2e-16
```

Part 5: Prediction

Finally, I predict the score of movies in the dataset and evluate the difference in the predictions. Although I do not have access to other movies not in the dataset, what I would have done to answer this is scrape data from IMBD for a movie that is past the 2016 release date, and then predict it's audience score using the variables of runtime, imbd_rating, etc., using the above regression model.

```
movies.BMA <- predict(mod_full, estimator = "BPM", se.fit = TRUE)
fitted = movies.BMA$fit
as.vector(fitted)
```

```
##   [1] 46.926652 77.615889 81.327661 70.559631 39.196019 86.235051 72.696599
##   [8] 43.821373 80.975252 64.715333 69.280211 56.560477 75.421050 51.409330
##  [15] 18.815900 69.594948 70.022356 69.012750 28.442202 19.534057 67.236430
##  [22] 83.759684 56.730879 56.314614 82.127017 44.587274 67.009016 47.165561
##  [29] 62.327380 86.182176 72.061501 70.638983 82.396312 72.636025 60.457386
##  [36] 62.950116 65.760314 54.251123 93.274794 80.490643 63.383015 82.080499
##  [43] 60.364003 44.999870 86.655629 62.361576 66.695195 63.342282  7.759869
##  [50] 56.807903 78.953362 78.772089 61.294827 59.601729 73.614644 63.836319
##  [57] 60.539169 47.860488 44.469736 59.642590 60.505963 50.220492 95.803242
##  [64] 86.658779 61.200144 28.368784 92.384086 59.976530 62.633553 70.887120
##  [71] 53.114487 41.316084 48.527066 70.992163 65.147054 53.922200 58.086218
##  [78] 52.711739 29.570030 80.584612 76.744999 68.989651 48.854919 46.328918
##  [85] 89.291863 54.128623 57.730795 17.328616 69.435355 47.656364 76.295479
##  [92] 84.783336 52.375098 57.591843 54.713576 57.919712 50.958024 70.480322
##  [99] 90.413108 86.765573 71.459335 53.217335 51.597694 83.960682 24.171599
## [106] 56.648957 76.229012 52.336705 45.755571 21.070711 97.259098 59.188725
## [113] 68.857580 39.452669 48.794185 56.273830 82.472720 62.536594 48.974439
## [120] 83.486676 56.593151 66.890055 17.721440 50.200771 74.224808 13.875340
## [127] 59.920715 54.622310 67.095033 76.874378 82.733216 77.064313 11.628928
## [134] 79.673688 73.797960 62.956490 59.412896 45.153524 50.263961 86.440376
## [141] 51.998083 51.160514 73.476151 75.987308 61.867189 68.242080 60.818339
## [148] 63.029284 84.795406 85.384565 17.332587 68.831192 63.973122 65.845555
## [155] 24.695376 59.582940 52.593380 30.372906 77.221598 59.577177 67.097951
## [162] 77.882645 60.822302 78.691869 42.778005 77.868929 62.546366 60.662967
## [169] 67.213975 52.615240 39.795879 72.441195 91.620940 71.628596 78.028245
## [176] 80.817835 48.638670 66.505628 81.951070 63.707450 73.155073 78.388553
## [183] 48.448346 54.169501 24.412988 54.050590 -2.273160 59.114936 83.370000
## [190] 60.297985 85.501468 60.366337 51.857387 66.605583 73.867774 79.723063
```

```
## [197] 76.766763 78.889377 48.555140 55.313152 89.554984 67.908197 25.644659
## [204] 79.016117 61.043379 55.617949 50.681413 89.136478 66.459215 54.732950
## [211] 73.565751 70.138328 67.265435 61.060060 53.674820 25.924026 56.928016
## [218] 44.401604 85.658364 76.311647 56.457974 66.210966 21.987776 62.306652
## [225] 61.459135 86.393249 47.120643 83.000314 63.664517 61.601323 52.379730
## [232] 61.962418 77.220412 38.087799 81.710523 71.783929 78.847139 35.199712
## [239] 62.084562 70.000327 49.421676 57.618170 56.751481 87.675474 66.163335
## [246] 47.074270 69.122673 49.986745 32.325065 51.266571 20.442126 77.505096
## [253] 73.429397 64.330257 73.830496 51.961062 63.686971 28.971703 67.236557
## [260] 80.229662 32.473044 55.013969 79.390401 65.167605 85.678192 71.719299
## [267] 66.698159 86.596191 57.689294 61.497529 82.974240 68.896777 84.054958
## [274] 77.861904 77.193212 71.954946 75.253952 77.875635 51.816031 66.095073
## [281] 50.105877 66.559096 47.122699 54.617299 40.991358 73.380168 78.079509
## [288] 74.859590 57.581421 77.911576 79.604475 77.167346 78.681565 67.340331
## [295] 43.587950 68.761093 44.706980 62.371825 76.439196 60.030684 52.789085
## [302] 41.600678 58.372120 56.608922 48.473012 60.741223 57.798843 62.593782
## [309] 63.292054 55.282011 63.726981 40.679426 40.757213 82.111332 67.629678
## [316] 54.743765 54.275126 81.742469 65.944875 57.774731 72.691456 56.420137
## [323] 62.834211 43.143852 58.815723 68.638567 52.470367 51.645184 71.734843
## [330] 70.390542 51.153061 46.550077 73.897697 66.637568 81.265304 55.507411
## [337] 63.383300 59.653657 58.130478 89.829120 82.993756 75.777873 51.809513
## [344] 65.523072 72.768851 87.675474 -3.182413 55.995250 48.422982 73.438794
## [351] 98.289988 79.764816 62.826327 47.533813 81.714109 62.896926 86.334446
## [358] 53.513047 66.302885 55.909692 75.118954 52.079275 44.700498 56.265354
## [365] 41.015267 85.542711 71.741531 53.999764 40.923075 48.802770 70.187998
## [372] 34.760682 74.061001 32.860187 -7.008793 10.242129 88.453982 77.522270
## [379] 54.011062 74.773864 77.084141 81.391229 97.461311 61.699465 87.150509
## [386] 62.803022 82.230476 73.428622 52.653618 70.236727 50.516882 68.064031
## [393] 93.286246 72.800128 71.907003 68.308832 82.791209 92.596842 88.902901
## [400] 52.815589 62.917581 60.613163 66.143725 86.991105 39.698194 37.770311
## [407] 17.206981 62.806643 51.094064 61.361709 85.055960 41.684420 46.851471
## [414] 60.105443 69.737957 94.662292 38.641652 49.600141 28.742507 78.749073
## [421] 30.158925 86.105484 53.209353 62.977746 71.471215 68.091861 56.533741
## [428] 44.069812 77.625612 60.785210 58.746513 67.965517 52.292170 72.847660
## [435] 59.566348 57.139571 70.124596 42.253624 81.417436 67.938240 38.240198
## [442] 40.225339 74.146995 57.984118 73.651152 70.197584 66.113933 85.290439
## [449] 79.331170 90.943539 52.953697 47.548513 67.273234 49.965969 52.598767
## [456] 37.126601 83.123150 15.062178 61.591341 35.922349 76.237219 46.796409
## [463] 56.886602 59.623723 69.286205 41.372300 63.586975 98.687102 24.899404
## [470] 60.659836 35.385033 56.040701 68.963799 41.803250 47.098755 97.260484
## [477] 52.322928 75.386338 60.082897 63.372649 60.690632 15.767446 60.186339
## [484] 76.662444 53.443944 55.063505 43.854977 41.968469 64.952473 90.600782
## [491] 42.119625 91.534132 76.111192 52.793685 56.137005 82.703151 58.005061
## [498] 65.803826 44.750006 77.357089 85.001009 15.721211 42.692301 53.792670
## [505] 56.281401 85.167165 44.691358 81.197306 26.041758 71.396648 81.736901
## [512] 75.973717 53.157412 60.211081 58.000409 47.567296 72.010646 53.258941
## [519] 74.681507 64.604012 55.797001 69.605441 71.698292 64.814380 64.497089
## [526] 38.354438 92.624838 71.275222 71.217909 44.224687 72.642923 80.338875
## [533] 49.796509 61.867987 58.106941 64.401420 81.583328 42.694036 70.331095
## [540] 75.505141 67.076072 90.762726 47.146151 49.323731 30.302941 62.475770
## [547] 52.179625 58.942896 74.798962 25.193282 85.387600 59.109257 81.628093
## [554] 78.452429 65.630267 80.706286 54.360674 59.892245 82.308054 79.634437
## [561] 33.663113 52.478103 74.014625 72.050975 48.022610 31.411575 71.537815
## [568] 66.554568 71.393584 44.505995 71.978702 53.086486 76.734605 83.469881
```

```
## [575] 72.644460 79.465244 69.834706 69.192149 69.208673 73.310015 53.274159
## [582] 59.802847 71.888786 87.740882 58.241691 83.592763 38.259135 61.911955
## [589] 73.137658 73.114083 79.368985 61.006239 79.816122 77.574778 86.308873
## [596] 80.456491 68.519068 83.508391 56.696538 71.066606 94.606755 69.020591
## [603] 52.397395 56.728096 66.675827 65.747097 55.757505 80.281809 82.506695
## [610] 84.867895 52.683705 67.867976 70.678039 88.192531 81.200752 30.122404
## [617] 81.651300 53.294720 60.967403 51.689831 36.985652 74.962611 76.177173
## [624] 70.009010 88.898350 43.050656 63.445746 44.045780  5.491565 46.096861
## [631] 25.059519 77.232738 37.514313 65.907983 88.350101 55.252938 50.030431
## [638] 74.381529 58.801251 42.739810 82.981924 80.649725 70.997491 78.360425
## [645] 42.876324 54.593501 63.743461 53.589258 51.362459 26.678384
```

Part 6: Conclusion

What makes a good movie? Well, apparently it depends on what you define "good" to be. If we, as a company, are trying to optimize audience score, then different variables might be used in the prediction than if we were trying to optimize critics score, or probaiblity of recieving an oscar.

In this analysis, we focused on the former. That is, our intention is to build a prediction model that predicts audience score from a host of variables. While the most parsimonious model (with the lowest BIC), only included the two predictors of imbd_rating and critics_score, this might not be the best model, since it is leaving out variable information we have in other predictors. Additionally, imbd_rating and ceritics_score were highly correlated, as we saw from the initial descriptive statistics, and multi-collinearity might be resulting in a biased model.

Therefore, I choose to go with a model that might be a little bit more complex, and sacrifice some BIC, but on the other hand, will take into account other predictors that are still signifiantly associated with the outcome of audience score. By doing so, I actually increase the R squared value, meaning that these additional predictors explain more variation in the outcome of audience score. The lesson here is that parsimony is not always best.

Besides those statistical limitations of the analysis, shortcomings include a limited set of variables and the lack of a testing dataset to run predictions on. If I were to continue this research in the future, I'd again think about comparing the model to a different dependent, or outcome variable. What makes a movie "good" might be based on the box office monetary value, or the number of DVDs sold, rather than audience rating, which might be more subjective.