Bayesian modeling and prediction for movies

TNMayer

2016-12-25

Setup

Load packages

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

Load data

Make sure your data and R Markdown files are in the same directory. When loaded your data file will be called `movies` . Delete this note when before you submit your work.

```
load("movies.Rdata")
#base_path <- getwd()
#path <- file.path(base_path, 'course04', 'movies.Rdata')
#load(path)
```

Part 1: Data

Instruction: Describe how the observations in the sample are collected, and the implications of this data collection method on the scope of inference (generalizability / causality).

Study design: The data set is comprised of 651 randomly sampled movies produced and released before 2016.

Source: Rotten Tomatoes (http://www.rottentomatoes.com/) and IMDB (http://www.imdb.com/) APIs.

Implications: The described sampling methodology outlines a **random sampling but no random assignment**, hence it is an **Observational Study**. Each subject in the population is equally likely to be selected and the sample is likely representative for the aforementioned population. No random assignment means the study can provide **associations but no causation**. Random sampling means the findings of the study can be **generalized** to the population. The population is for this study: **all movies listed in the Rotten Tomatoes and IMDB databases**.

Generalizability is given.

Causality is not given.

Part 2: Data manipulation

NOTE: Insert code chunks as needed by clicking on the “Insert a new code chunk” button above. Make sure that your code is visible in the project you submit. Delete this note when before you submit your work.

Instructions:

- Create new variable based on title_type: New variable should be called feature_film with levels yes (movies that are feature films) and no (2 pt)
- Create new variable based on genre: New variable should be called drama with levels yes (movies that are dramas) and no (2 pt)
- Create new variable based on mpaa_rating: New variable should be called mpaa_rating_R with levels yes (movies that are R rated) and no (2 pt)
- Create two new variables based on thtr_rel_month:
- New variable called oscar_season with levels yes (if movie is released in November, October, or December) and no (2 pt)
- New variable called summer_season with levels yes (if movie is released in May, June, July, or August) and no (2 pt)

```r
# create new variable feature_film
movies <- movies %>%
  mutate(feature_film = ifelse(title_type == "Feature Film", 'yes', 'no'))

movies$feature_film <- as.factor(movies$feature_film)

# create new variable drama
movies <- movies %>%
  mutate(drama = ifelse(genre == "Drama", 'yes', 'no'))

movies$drama <- as.factor(movies$drama)

# create variable mpaa_rating_R
movies <- movies %>%
  mutate(mpaa_rating_R = ifelse(mpaa_rating == "R", 'yes', 'no'))

movies$mpaa_rating_R <- as.factor(movies$mpaa_rating_R)

# create new variable oscar_season
movies <- movies %>%
  mutate(oscar_season = ifelse((thtr_rel_month %in% c(10:12)), 'yes', 'no'))

movies$oscar_season <- as.factor(movies$oscar_season)

# create new variable summer_season
movies <- movies %>%
  mutate(summer_season = ifelse((thtr_rel_month %in% c(5:8)), 'yes', 'no'))

movies$summer_season <- as.factor(movies$summer_season)
```

Part 3: Exploratory data analysis

NOTE: Insert code chunks as needed by clicking on the “Insert a new code chunk” button above. Make sure that your code is visible in the project you submit. Delete this note when before you submit your work.

Instructions:

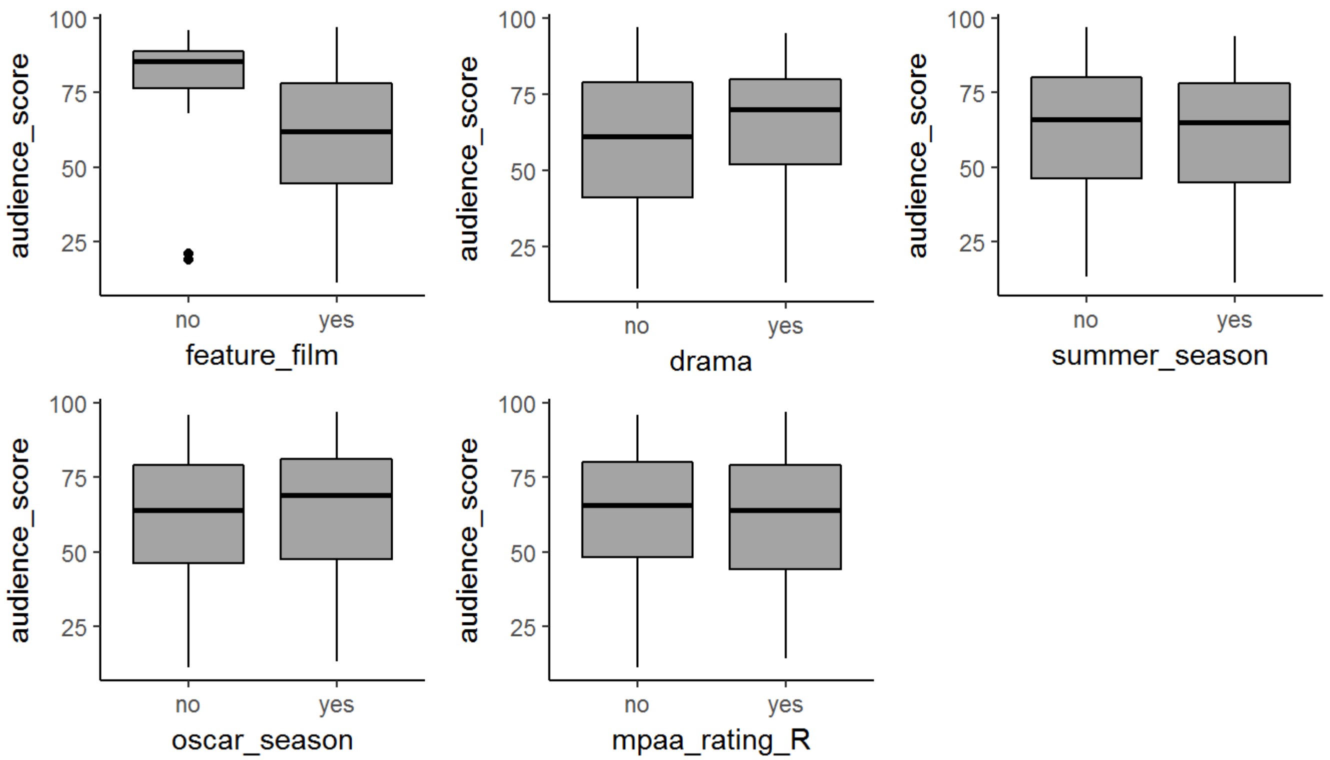
Conduct exploratory data analysis of the relationship between audience_score and the new variables constructed in the previous part.

- plots
- summary statistics
- narrative

```
# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                    ncol = cols, nrow = ceiling(numPlots/cols))
  }

 if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

Plots

```
# ggpairs(movies, columns = c(18, 33:37))
p1 <- ggplot(movies, aes(x=feature_film, y=audience_score)) +
        geom_boxplot(fill='#A4A4A4', color="black")+
        theme_classic()
p2 <- ggplot(movies, aes(x=oscar_season, y=audience_score)) +
        geom_boxplot(fill='#A4A4A4', color="black")+
        theme_classic()
p3 <- ggplot(movies, aes(x=drama, y=audience_score)) +
        geom_boxplot(fill='#A4A4A4', color="black")+
        theme_classic()
p4 <- ggplot(movies, aes(x=mpaa_rating_R, y=audience_score)) +
        geom_boxplot(fill='#A4A4A4', color="black")+
        theme_classic()
p5 <- ggplot(movies, aes(x=summer_season, y=audience_score)) +
        geom_boxplot(fill='#A4A4A4', color="black")+
        theme_classic()

multiplot(p1, p2, p3, p4, p5, cols=3)
```



Feature films tend to have a lower audience score than no feature films in general. Drama´s tend to have higher audience scores than no drama movies. There is no remarkable relationship in between summer season and non summer season movies as well as betwenn oscar season and non oscar season movies. There is also only a weak relationship identifyable between movies rated with “R” and movies not rated with “R” (MPAA rating). All categories tend to show a left skewed distribution on behalf of their audience scores.

Summary Statistics

```
movies %>%
  group_by(feature_film) %>%
  summarise(Mean = mean(audience_score), Median = median(audience_score), Min = min(aud
ience_score),
            Max = max(audience_score), IQR = IQR(audience_score))
```

```
## # A tibble: 2 × 6
##   feature_film     Mean Median   Min   Max   IQR
##         <fctr>    <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1           no 81.05000   85.5    19    96  12.5
## 2          yes 60.46531   62.0    11    97  33.5
```

```
movies %>%
  group_by(drama) %>%
  summarise(Mean = mean(audience_score), Median = median(audience_score), Min = min(aud
ience_score),
            Max = max(audience_score), IQR = IQR(audience_score))
```

```
## # A tibble: 2 × 6
##    drama     Mean Median   Min   Max   IQR
##   <fctr>    <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1     no 59.73121     61    11    97    38
## 2    yes 65.34754     70    13    95    28
```

```
movies %>%
  group_by(summer_season) %>%
  summarise(Mean = mean(audience_score), Median = median(audience_score), Min = min(aud
ience_score),
            Max = max(audience_score), IQR = IQR(audience_score))
```

```
## # A tibble: 2 × 6
##   summer_season     Mean Median   Min   Max   IQR
##          <fctr>    <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1            no 62.62302     66    13    97 34.00
## 2           yes 61.80769     65    11    94 33.25
```

```
movies %>%
  group_by(oscar_season) %>%
  summarise(Mean = mean(audience_score), Median = median(audience_score), Min = min(aud
ience_score),
            Max = max(audience_score), IQR = IQR(audience_score))
```

```
## # A tibble: 2 × 6
##   oscar_season     Mean Median   Min   Max   IQR
##         <fctr>    <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1           no 61.81304     64    11    96  33.0
## 2          yes 63.68586     69    13    97  33.5
```

```
movies %>%
  group_by(mpaa_rating_R) %>%
  summarise(Mean = mean(audience_score), Median = median(audience_score), Min = min(aud
ience_score),
            Max = max(audience_score), IQR = IQR(audience_score))
```

```
## # A tibble: 2 × 6
##   mpaa_rating_R     Mean Median   Min   Max   IQR
##          <fctr>    <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1            no 62.68944   65.5    11    96 31.75
## 2           yes 62.04255   64.0    14    97 35.00
```

The same patterns as described above are also observable by the summary statistics. There is a quite big difference between the feature_film groups according to their average audience scores. Their is also a difference of roughly 6 points on average between the drama groups. The differences in the mean audience scores between the oscar_season, summer_season as well as mpaa_rating_R groups are not worth to mention. Hence the feature_film and drama variables should be able to differentiate well between high and low audience scores. How strong the new variables will influence the linear regression model will be observed in the next chapter.

Part 4: Modeling

NOTE: Insert code chunks as needed by clicking on the “Insert a new code chunk” button above. Make sure that your code is visible in the project you submit. Delete this note when before you submit your work.

Instruction:

Develop a Bayesian regression model to predict audience_score from the following explanatory variables. Note that some of these variables are in the original dataset provided, and others are new variables you constructed earlier.

Complete Bayesian model selection and report the final model.

- Carrying out the model selection correctly (5 pts)
- Model diagnostics (5 pts)
- Interpretation of model coefficients (5 pts)

Create the full model:

```
# select the needed variables (dependent and independent)
movies_model <- dplyr::select(movies, audience_score, feature_film, drama, runtime, mpa
a_rating_R, thtr_rel_year, oscar_season, summer_season, imdb_rating, imdb_num_votes, cr
itics_score, best_pic_nom, best_pic_win, best_actor_win, best_actress_win, best_dir_wi
n, top200_box)

m_movies_full = lm(audience_score ~ ., data = movies_model)

# print out the model summary
summary(m_movies_full)
```

```
##
## Call:
## lm(formula = audience_score ~ ., data = movies_model)
##
## 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
```

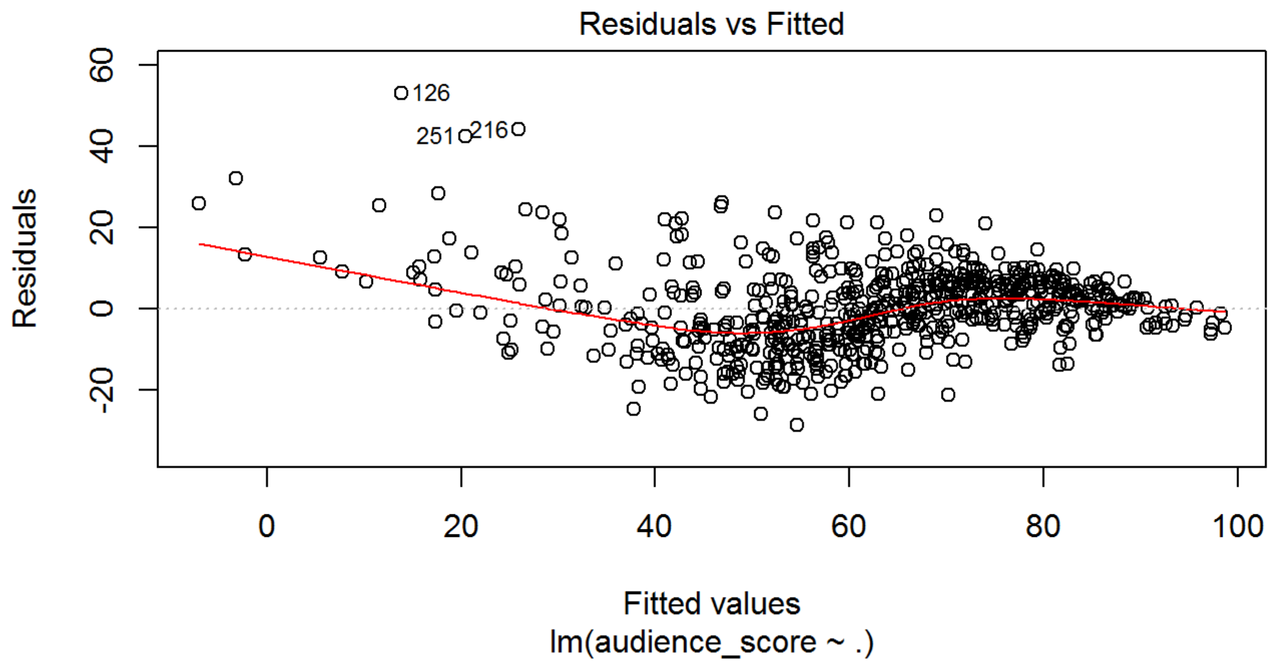
```
# get the BIC score of the full model
BIC(m_movies_full)
```

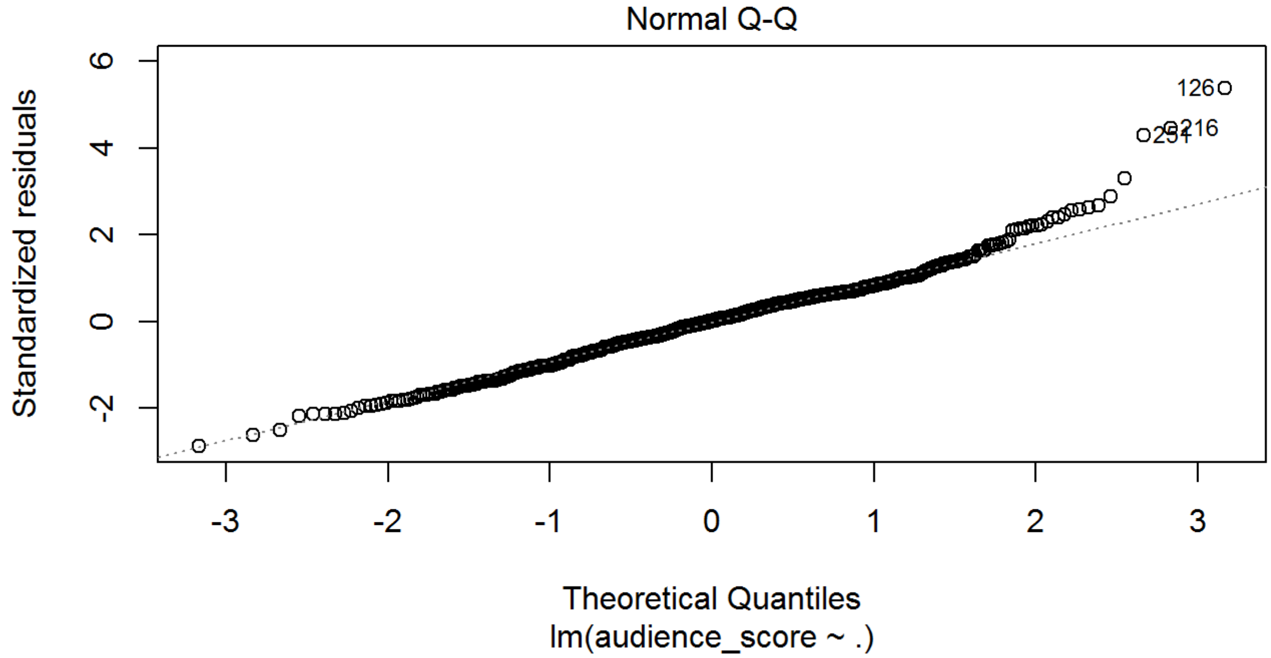
```
## [1] 4934.145
```

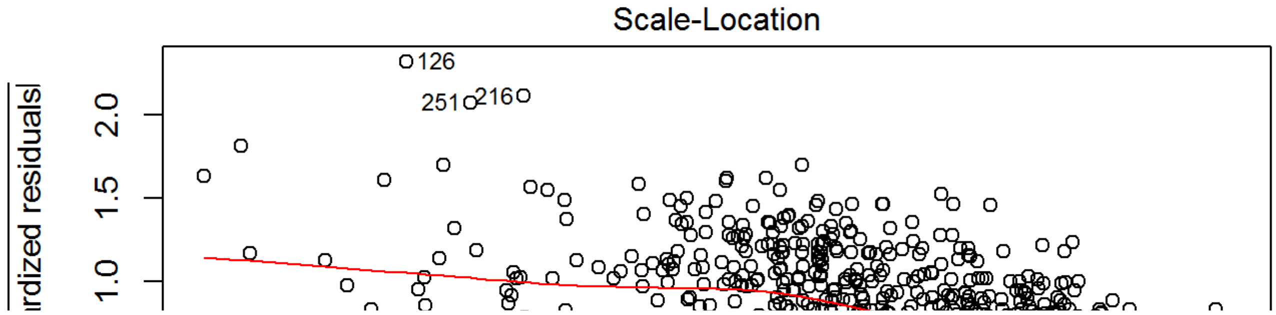
This shows that initially only seven out of 16 model weights differ significantly from 0. The Bayesian Information Criterion (BIC) of the full model is 4934.145. The goal of the following model selection process is to minimize the BIC. The adjusted R^2 value of the model is 0.757. A higher adjusted R^2 value indicates a better predictive value of the model and draws the number of independent variables into account.

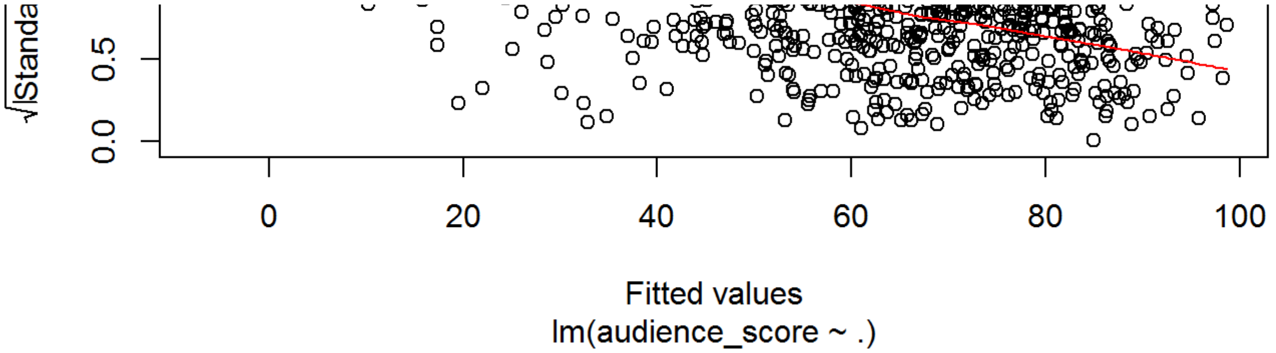
Before performing the model selection process let´s have a look at the model assumptions.

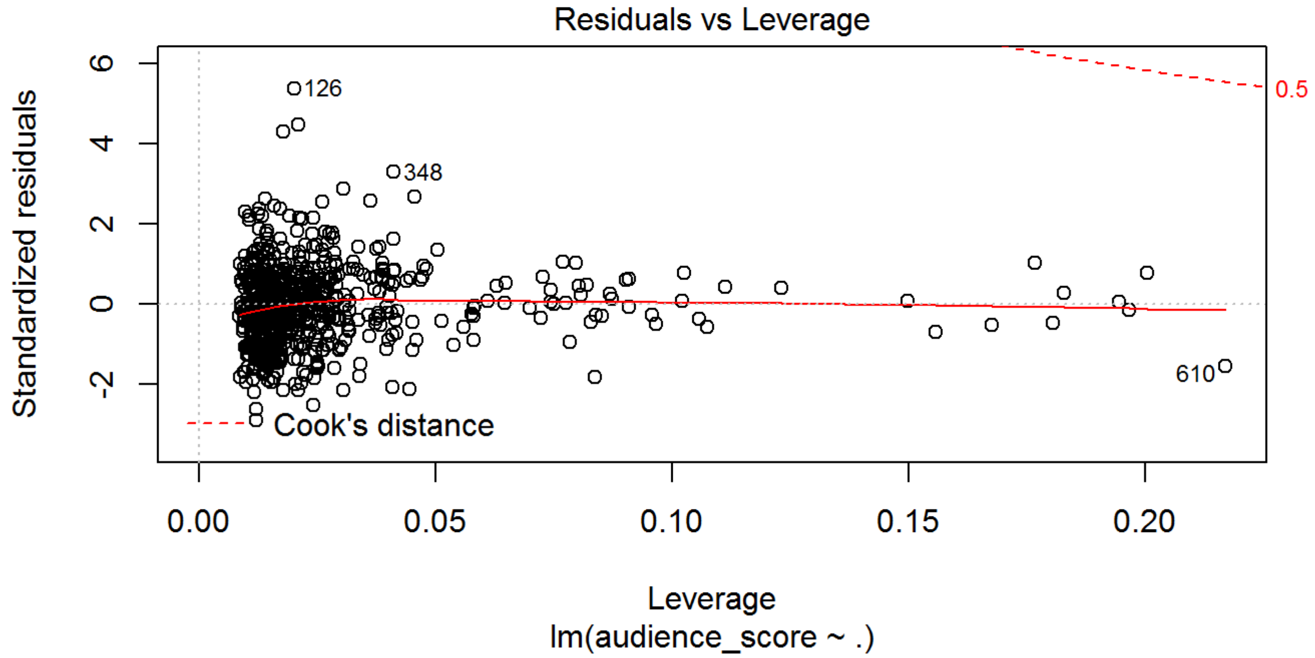
```
plot(m_movies_full)
```


Residuals vs Fitted
Residuals
60
40
20
0
-20
126
251
216
0
20
40
60
80
100
Fitted values
lm(audience_score ~ .)


Normal Q-Q
Standardized residuals
6
4
2
0
-2
126
251
216
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(audience_score ~ .)


Scale-Location
rdized residuals|
2.0
1.5
1.0
126
251
216





The distribution of the residuals seems to be right-skewed. One reason for that might be that the dependent variable of the model (audience score) is left skewed. Maybe a transformation of the variable can help to normalize the plot. One way to overcome the the skewness could be a flipped log transformation. But the Q-Q-Plot doesn´t look extremely bad, so the model selection process will be performed next.

Now the model selection process is performed using the stepAIC-function out of the MASS package. In order to find the model with the lowest BIC the penalty parameter k has to be adjusted as follows: $k = \log(n)$.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
m_movies_full = lm(audience_score ~ ., data = na.omit(movies_model))
m_movies.step <- stepAIC(m_movies_full, trace = T, k = log(nrow(movies_model)))
```

```
## Start:  AIC=3083.07
## 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
## 
##                    Df Sum of Sq    RSS    AIC
## - top200_box        1         9  62999 3076.7
## - oscar_season      1        28  63018 3076.9
## - best_pic_win      1        48  63038 3077.1
## - best_dir_win      1        51  63040 3077.1
## - summer_season     1        92  63081 3077.5
## - best_actor_win    1       171  63160 3078.4
## - feature_film      1       177  63166 3078.4
## - drama             1       216  63206 3078.8
## - imdb_num_votes    1       255  63244 3079.2
## - best_actress_win  1       283  63273 3079.5
## - mpaa_rating_R     1       314  63304 3079.8
## - thtr_rel_year     1       397  63386 3080.7
## - best_pic_nom      1       408  63398 3080.8
## - runtime           1       538  63527 3082.1
## <none>                           62990 3083.1
## - critics_score     1       669  63659 3083.5
## - imdb_rating       1     58556 121545 3503.9
## 
## Step:  AIC=3076.69
## 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 3070.5
## - best_pic_win      1        49  63047 3070.7
## - best_dir_win      1        52  63051 3070.8
## - summer_season     1        94  63093 3071.2
## - best_actor_win    1       169  63168 3072.0
## - feature_film      1       176  63175 3072.0
## - drama             1       214  63213 3072.4
## - best_actress_win  1       279  63278 3073.1
## - imdb_num_votes    1       302  63301 3073.3
## - mpaa_rating_R     1       330  63329 3073.6
## - best_pic_nom      1       404  63403 3074.4
## - thtr_rel_year     1       415  63414 3074.5
## - runtime           1       535  63534 3075.7
## <none>                           62999 3076.7
## - critics_score     1       681  63680 3077.2
## - imdb_rating       1     58606 121604 3497.7
## 
## Step:  AIC=3070.49
## 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 3064.5
## - best_dir_win      1        56  63081 3064.6
## - best_actor_win    1       174  63200 3065.8
## - summer_season     1       177  63202 3065.8
## - feature_film      1       182  63207 3065.9
## - drama             1       222  63247 3066.3
## - best_actress_win  1       281  63307 3066.9
## - imdb_num_votes    1       302  63328 3067.1
## - mpaa_rating_R     1       329  63354 3067.4
## - best_pic_nom      1       387  63412 3068.0
## - thtr_rel_year     1       410  63436 3068.2
## - runtime           1       587  63613 3070.0
## <none>                           63025 3070.5
## - critics_score     1       679  63704 3071.0
## - imdb_rating       1     58603 121628 3491.3
## 
## Step:  AIC=3064.48
## 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 3059.0
## - best_actor_win    1       163  63234 3059.7
## - feature_film      1       171  63242 3059.8
## - summer_season     1       174  63245 3059.8
## - drama             1       220  63291 3060.3
## - imdb_num_votes    1       271  63342 3060.8
## - best_actress_win  1       294  63365 3061.0
## - mpaa_rating_R     1       330  63401 3061.4
## - best_pic_nom      1       342  63414 3061.5
## - thtr_rel_year     1       397  63468 3062.1
## - runtime           1       586  63657 3064.0
## <none>                           63071 3064.5
## - critics_score     1       680  63751 3065.0
## - imdb_rating       1     58858 121929 3486.5
## 
## Step:  AIC=3058.97
## 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 3054.2
## - best_actor_win    1       171  63336 3054.2
## - feature_film      1       183  63348 3054.4
## - drama             1       228  63394 3054.8
## - imdb_num_votes    1       247  63412 3055.0
## - best_actress_win  1       299  63464 3055.6
## - best_pic_nom      1       326  63491 3055.8
## - mpaa_rating_R     1       345  63510 3056.0
## - thtr_rel_year     1       368  63533 3056.3
## <none>                           63165 3059.0
## - critics_score     1       651  63816 3059.2
## - runtime           1       673  63839 3059.4
## - imdb_rating       1     58895 122061 3480.7
## 
## Step:  AIC=3054.2
## 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 3049.3
## - best_actor_win    1       195  63527 3049.7
## - drama             1       204  63536 3049.8
## - imdb_num_votes    1       260  63592 3050.4
## - best_pic_nom      1       297  63629 3050.8
## - best_actress_win  1       297  63629 3050.8
## - mpaa_rating_R     1       356  63688 3051.4
## - thtr_rel_year     1       361  63693 3051.4
## <none>                           63332 3054.2
## - runtime           1       690  64022 3054.8
## - critics_score     1       732  64064 3055.2
## - imdb_rating       1     58763 122095 3474.4
## 
## Step:  AIC=3049.32
## 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 3044.1
## - imdb_num_votes    1       173  63661 3044.6
## - best_actor_win    1       219  63706 3045.1
## - thtr_rel_year     1       277  63765 3045.7
## - best_pic_nom      1       291  63778 3045.8
## - best_actress_win  1       306  63794 3046.0
## - mpaa_rating_R     1       453  63941 3047.5
## <none>                           63488 3049.3
## - runtime           1       715  64203 3050.1
## - critics_score     1       875  64363 3051.7
## - imdb_rating       1     63189 126677 3491.9
## 
## Step:  AIC=3044.09
## 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 3039.1
## - best_actor_win    1       209  63818 3039.7
## - thtr_rel_year     1       272  63881 3040.4
## - best_actress_win  1       274  63883 3040.4
## - best_pic_nom      1       307  63916 3040.7
## - mpaa_rating_R     1       391  64000 3041.6
## - runtime           1       631  64240 3044.0
## <none>                           63609 3044.1
## - critics_score     1       916  64525 3046.9
## - imdb_rating       1     63434 127043 3487.3
##
## Step:  AIC=3039.12
## 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
## - thtr_rel_year     1       201  63958 3034.7
## - best_actor_win    1       219  63976 3034.9
## - best_actress_win  1       266  64023 3035.3
## - mpaa_rating_R     1       367  64124 3036.4
## - best_pic_nom      1       442  64199 3037.1
## - runtime           1       519  64276 3037.9
## <none>                           63757 3039.1
## - critics_score     1       879  64635 3041.5
## - imdb_rating       1     67356 131113 3501.3
##
## Step:  AIC=3034.68
## audience_score ~ runtime + mpaa_rating_R + imdb_rating + critics_score +
##     best_pic_nom + best_actor_win + best_actress_win
##
##                     Df Sum of Sq    RSS    AIC
## - best_actor_win    1       207  64165 3030.3
## - best_actress_win  1       261  64219 3030.9
## - mpaa_rating_R     1       373  64331 3032.0
## - best_pic_nom      1       447  64405 3032.7
## - runtime           1       468  64425 3032.9
## <none>                           63958 3034.7
## - critics_score     1       968  64926 3038.0
## - imdb_rating       1     67172 131129 3494.9
##
## Step:  AIC=3030.3
## audience_score ~ runtime + mpaa_rating_R + imdb_rating + critics_score +
##     best_pic_nom + best_actress_win
##
##                     Df Sum of Sq    RSS    AIC
## - best_actress_win  1       296  64461 3026.8
## - mpaa_rating_R     1       366  64531 3027.5
## - best_pic_nom      1       396  64561 3027.8
## <none>                           64165 3030.3
## - runtime           1       643  64808 3030.3
## - critics_score     1       968  65133 3033.6
## - imdb_rating       1     67296 131461 3490.0
##
## Step:  AIC=3026.82
## audience_score ~ runtime + mpaa_rating_R + imdb_rating + critics_score +
##     best_pic_nom
##
##                  Df Sum of Sq    RSS    AIC
## - best_pic_nom   1       303  64765 3023.4
## - mpaa_rating_R  1       354  64815 3023.9
## <none>                        64461 3026.8
## - runtime        1       814  65275 3028.5
## - critics_score  1       957  65418 3029.9
## - imdb_rating    1     67424 131885 3485.7
## 
## Step:  AIC=3023.39
## audience_score ~ runtime + mpaa_rating_R + imdb_rating + critics_score
## 
##                 Df Sum of Sq    RSS    AIC
## - mpaa_rating_R  1       361  65126 3020.5
## - runtime        1       638  65403 3023.3
## <none>                        64765 3023.4
## - critics_score  1      1027  65792 3027.1
## - imdb_rating    1     68173 132937 3484.3
## 
## Step:  AIC=3020.53
## audience_score ~ runtime + imdb_rating + critics_score
## 
##                 Df Sum of Sq    RSS    AIC
## <none>                        65126 3020.5
## - runtime        1       653  65779 3020.5
## - critics_score  1      1073  66199 3024.7
## - imdb_rating    1     67874 133000 3478.2
```

The model selection process shows that the minimum BIC/AIC value can be achieved by a model with only three independent variables (runtime, imdb_rating and critics_score). Let´s have a look on the summary statistics and the BIC of the model.

```
summary(m_movies.step)
```

```
##
## Call:
## lm(formula = audience_score ~ runtime + imdb_rating + critics_score,
##     data = na.omit(movies_model))
##
## 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
## Multiple R-squared:  0.7549, Adjusted R-squared:  0.7538
## F-statistic: 663.3 on 3 and 646 DF,  p-value: < 2.2e-16
```

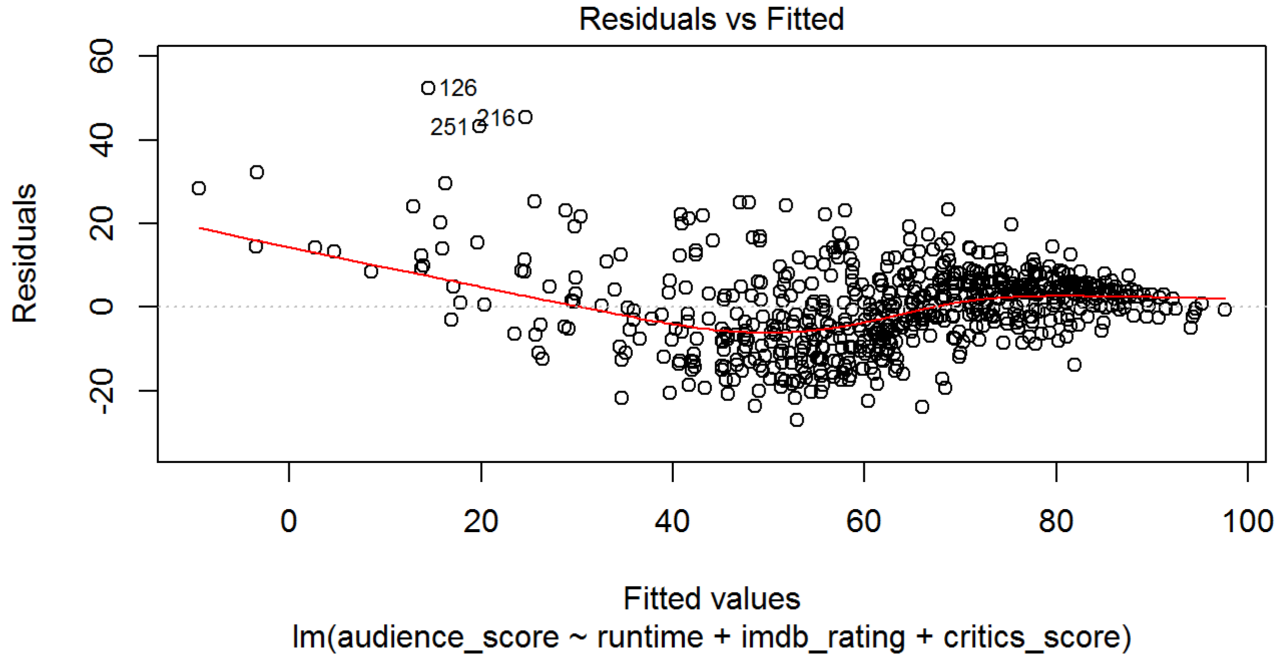
```
BIC(m_movies.step)
```

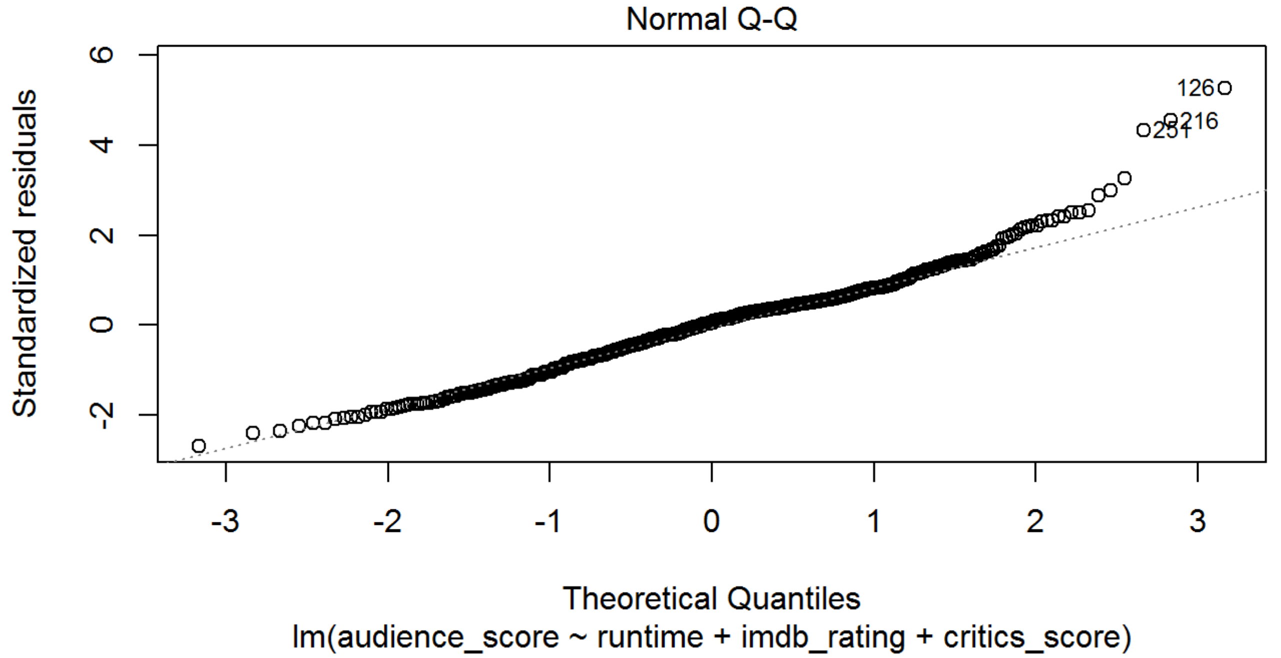
```
## [1] 4871.623
```

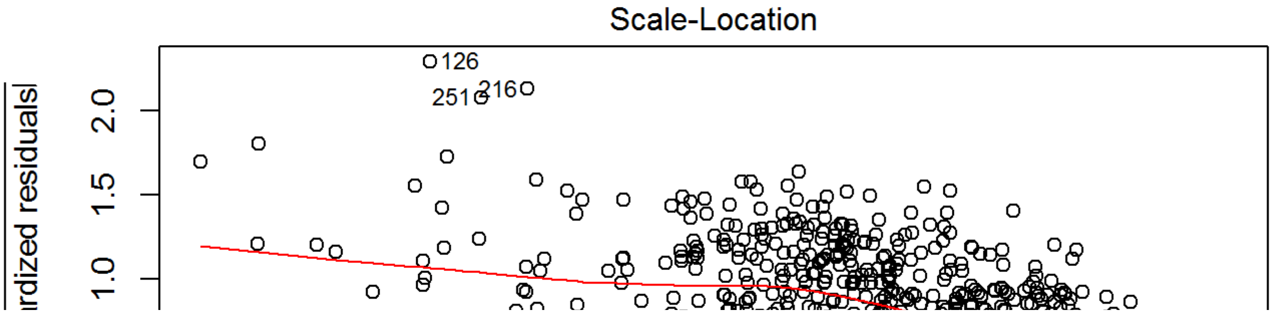
Let´s have a short look on the model coefficients. To give a short explanation I will focus on the coefficient for the imdb_rating variable with a coefficient value of 14.98. This means, holding all other variables in the model constant, a movie with one point more in the imdb_rating is expected to increase the audience score by 14.98 points. The range for imdb_rating scores is [1.9; 9.0].

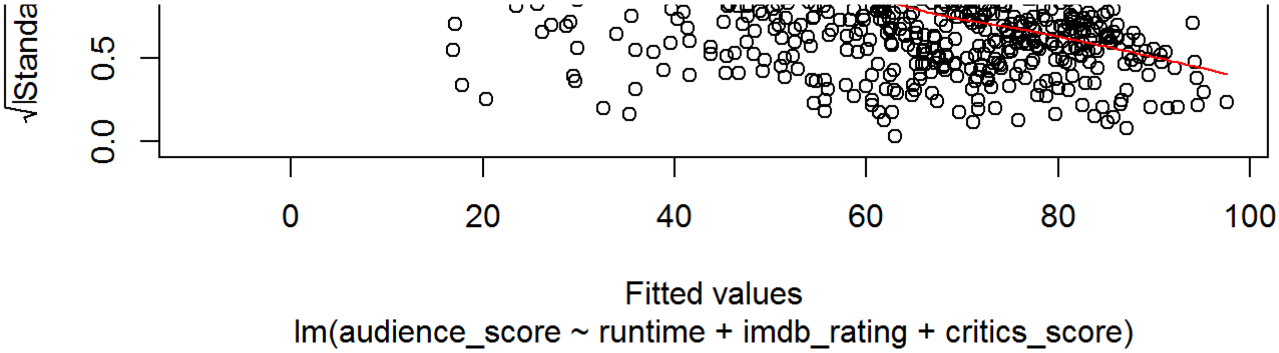
Now let´s have a look at the model diagnostics by having a look on the residual plots:

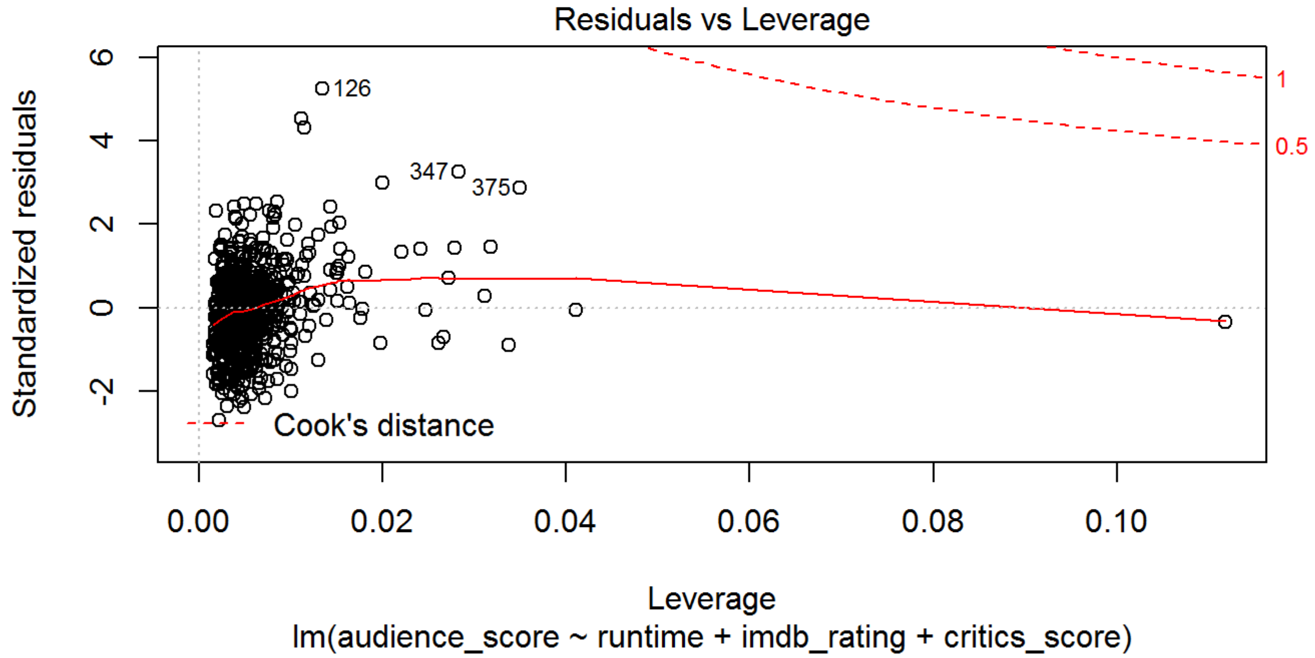
```
plot(m_movies.step)
```


Residuals vs Fitted
Residuals
60
40
20
0
-20
126
251
216
0
20
40
60
80
100
Fitted values
lm(audience_score ~ runtime + imdb_rating + critics_score)


Normal Q-Q
Standardized residuals
6
4
2
0
-2
126
251
216
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(audience_score ~ runtime + imdb_rating + critics_score)


Scale-Location
rdized residuals|
2.0
1.5
1.0
126
251
216

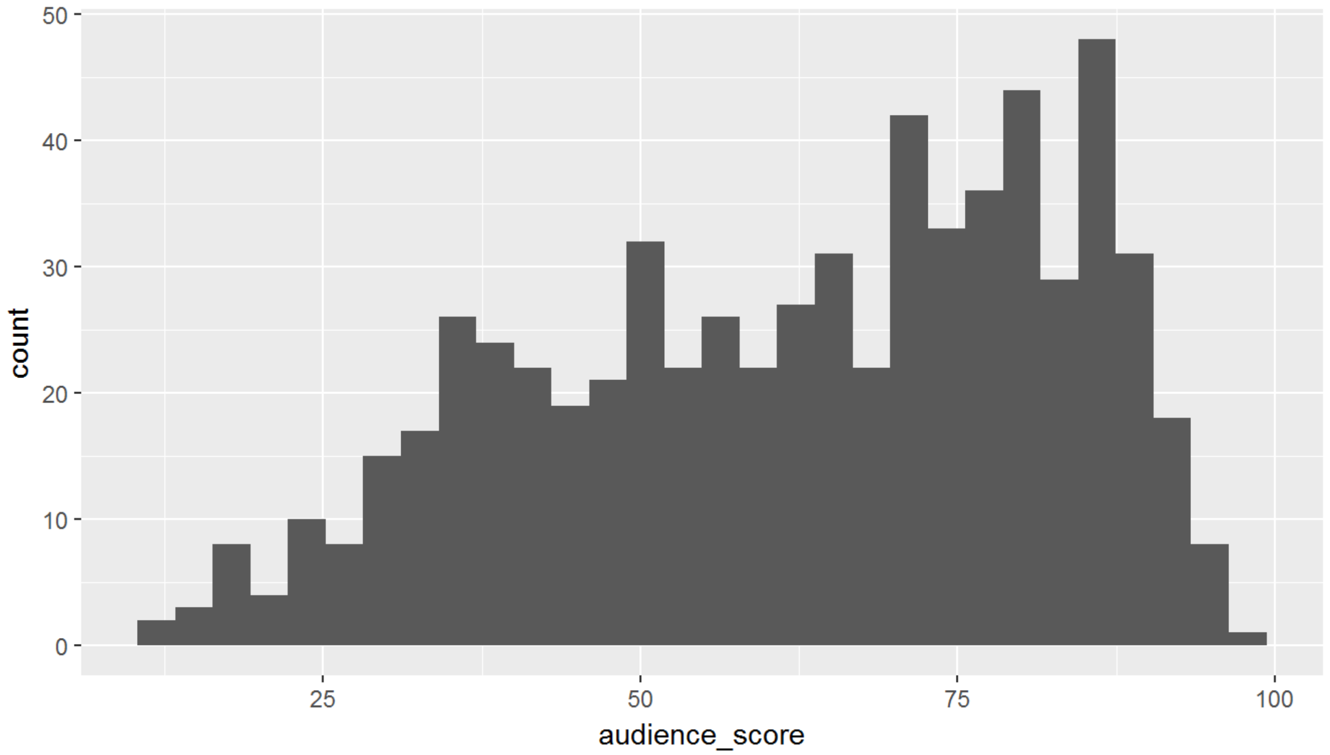




The residual plot still appears to be right skewed but it doesn´t look extremely bad. One way to overcome this kind of problem could be by normalizing the included variables by transformation. Let´s have a look on the variable distributions:

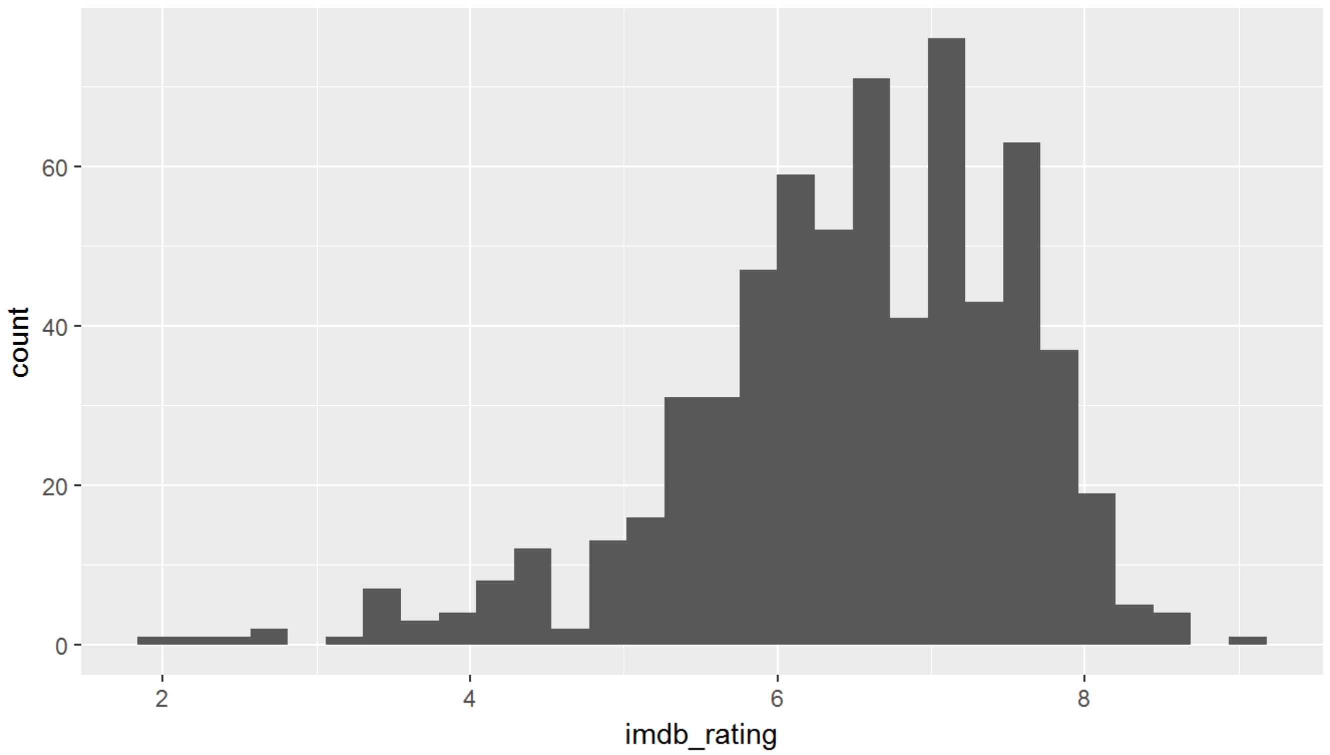
```
ggplot(data = movies_model, aes(x = audience_score)) + geom_histogram()
```

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



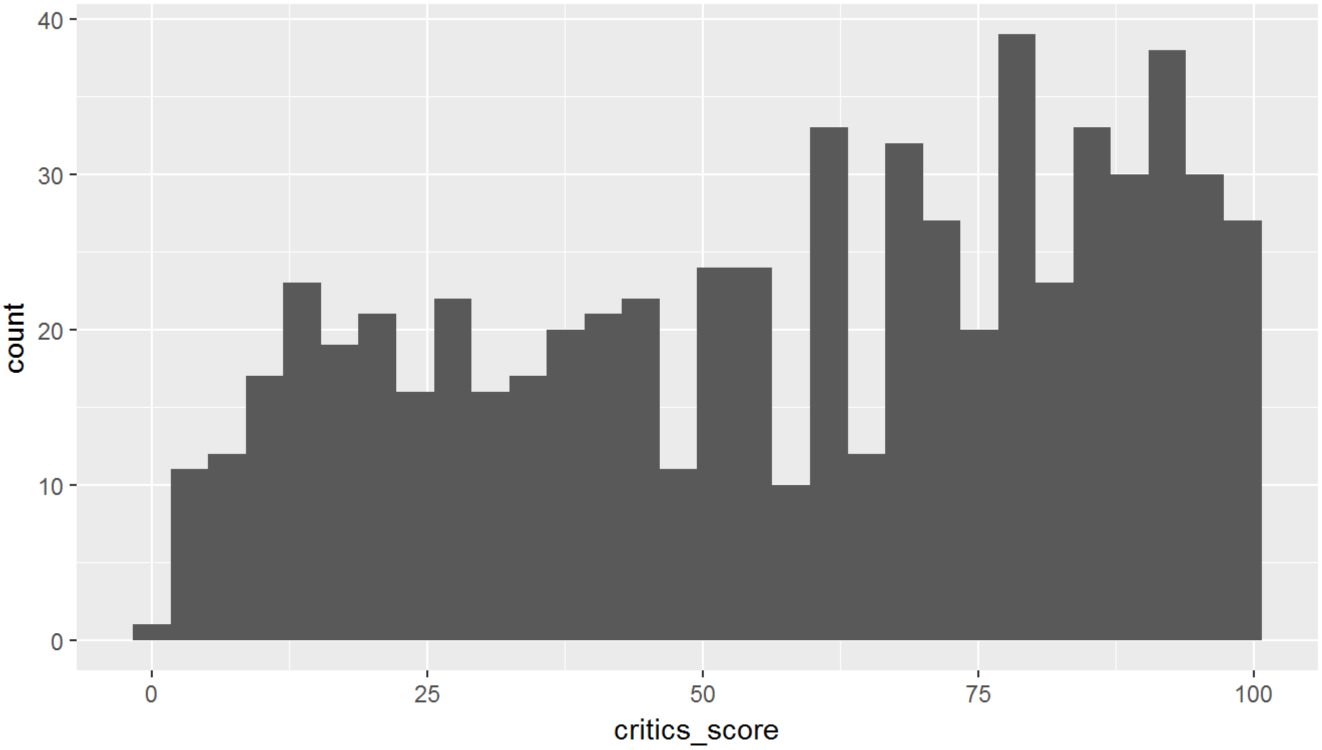
```r
ggplot(data = movies_model, aes(x = imdb_rating)) + geom_histogram()
```

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



```r
ggplot(data = movies_model, aes(x = critics_score)) + geom_histogram()
```

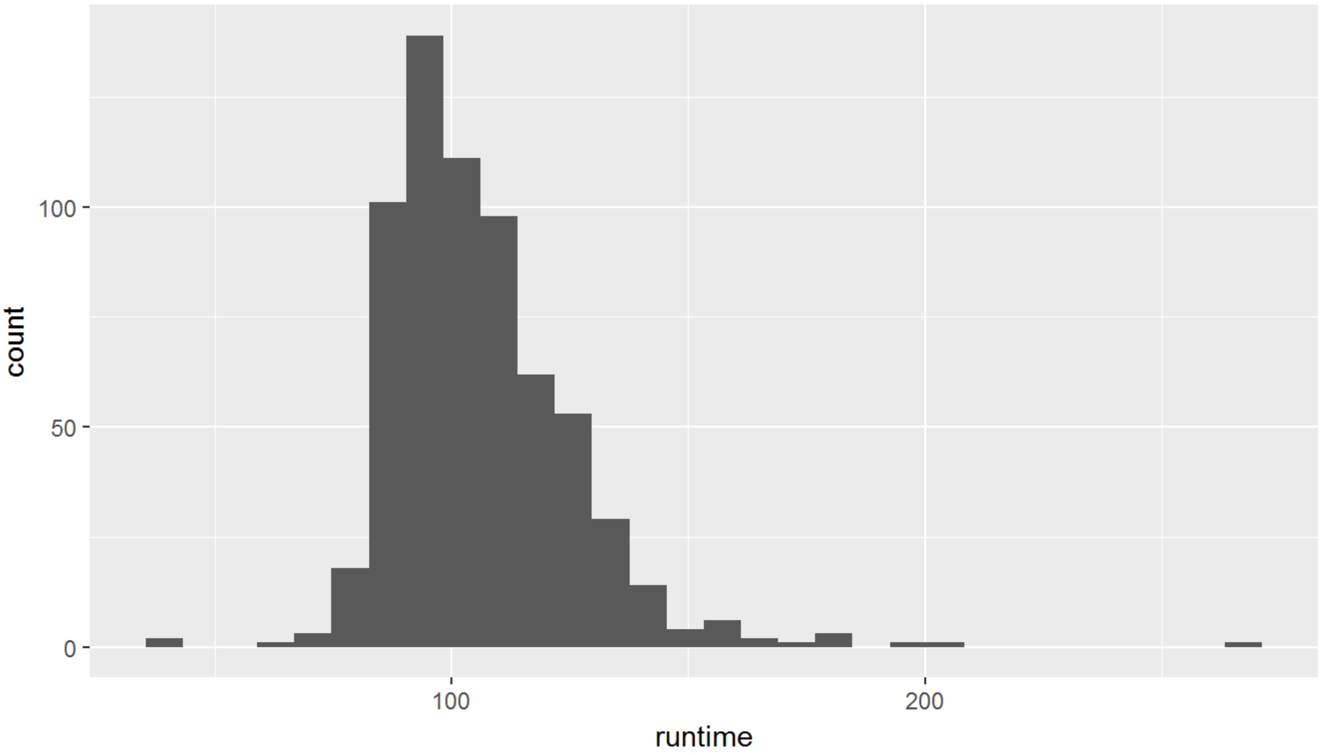
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(data = movies_model, aes(x = runtime)) + geom_histogram()
```

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

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



As already mentioned the audience_score is left skewed and also the runtime variable is left skewed. The critics_score has no bellshade shape and the runtime variable looks pretty normal distributed. The imdb_rating is left skewed. For the further research process no additional transformations will be applied to the data.

Bayesian Model Averaging (BMA)

Before predicting the audience score for new movies I want to apply a model selection process by Bayesian Model Averaging first. I don´t want to ignore the inherent uncertainty involved in choosing the variables to include in the model.

```
# exclude NA values from the data
movies_nona <- na.omit(movies_model)
bma_audience = bas.lm(audience_score ~ ., data = movies_nona,
                   prior = "BIC",
                   modelprior = uniform())
bma_audience
```

```
##
## Call:
## bas.lm(formula = audience_score ~ ., data = movies_nona, prior = "BIC",     modelpri
or = uniform())
##
##
##  Marginal Posterior Inclusion Probabilities:
##           Intercept      feature_filmyes             dramayes
##             1.00000              0.06537              0.04320
##             runtime     mpaa_rating_Ryes        thtr_rel_year
##             0.46971              0.19984              0.09069
##     oscar_seasonyes     summer_seasonyes          imdb_rating
##             0.07506              0.08042              1.00000
##      imdb_num_votes        critics_score      best_pic_nomyes
##             0.05774              0.88855              0.13119
##     best_pic_winyes    best_actor_winyes  best_actress_winyes
##             0.03985              0.14435              0.14128
##     best_dir_winyes        top200_boxyes
##             0.06694              0.04762
```

```
summary(bma_audience)
```

```
##      Intercept feature_filmyes dramayes runtime mpaa_rating_Ryes
## [1,]         1               0        0       1                0
## [2,]         1               0        0       0                0
## [3,]         1               0        0       0                0
## [4,]         1               0        0       0                1
## [5,]         1               0        0       1                1
##      thtr_rel_year oscar_seasonyes summer_seasonyes imdb_rating
## [1,]             0               0                0           1
## [2,]             0               0                0           1
## [3,]             0               0                0           1
## [4,]             0               0                0           1
## [5,]             0               0                0           1
##      imdb_num_votes critics_score best_pic_nomyes best_pic_winyes
## [1,]              0             1               0               0
## [2,]              0             1               0               0
## [3,]              0             1               0               0
## [4,]              0             1               0               0
## [5,]              0             1               0               0
##      best_actor_winyes best_actress_winyes best_dir_winyes top200_boxyes
## [1,]                 0                   0               0             0
## [2,]                 0                   0               0             0
## [3,]                 1                   0               0             0
## [4,]                 0                   0               0             0
## [5,]                 0                   0               0             0
##             BF PostProbs     R2 dim   logmarg
## [1,] 1.0000000    0.1297 0.7549   4 -3615.279
## [2,] 0.9968489    0.1293 0.7525   3 -3615.282
## [3,] 0.2543185    0.0330 0.7539   4 -3616.648
## [4,] 0.2521327    0.0327 0.7539   4 -3616.657
## [5,] 0.2391994    0.0310 0.7563   5 -3616.710
```

Printing the model object and the summary command gives both the posterior model inclusion probability for each variable and the most probable models. For example, the posterior probability that `runtime` is included in the model is 0.46971. Further, the most likely model, which has posterior probability of 0.1297, includes an intercept, runtime, imdb_rating and critics_score. While a posterior probability of 0.1297 sounds small, it is much larger than the uniform prior probability assigned to it, since there are 2^{17} (intercept + 16 explanatory variables) possible models.

Now let´s have a look on the predictor variables of the BMA best predictive model (BPM).

```
BPM_pred_audience <- predict(bma_audience, estimator = "BPM", se.fit = T)
bma_audience$namesx[BPM_pred_audience$bestmodel + 1]
```

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

Also the best predictive model by BMA yields the same predictor variables as the model selected by the stepAIC function from the MASS package that works backwards through the model space, removing variables until BIC can be no longer lowered.

Part 5: Prediction

NOTE: Insert code chunks as needed by clicking on the “Insert a new code chunk” button above. Make sure that your code is visible in the project you submit. Delete this note when before you submit your work.

Instructions:

Pick a movie from 2016 (a new movie that is not in the sample) and do a prediction for this movie using your the model you developed and the predict function in R.

- Correct prediction (4 pts)
- Reference(s) for where the data for this movie come from (1 pt)

In order to be able to predict the dependent/response variable for a movie from 2016 I first have to get the scores for the independent/explanatory variables. Those variables are:

- imdb_num_votes
- genre
- critics_score

I selected the movie The Accountant from 2016.

variable	score
`imdb_rating`	7.5
`runtime`	128
`critics_score`	52

The movie achieved an audience score of 81 % on Rotten Tomatoes.

Source: Rotten Tomatoes (http://www.rottentomatoes.com/) and IMDB (http://www.imdb.com/) APIs.

Now I want to observe how well the model will be able to predict the audience score of the Accountant.

First, I need to create a new data frame for this movie.

```r
newmovie <- data.frame(imdb_rating = 7.5, runtime = 128,
                       critics_score = 52)

# IOT make a prediction we have to fit a new prediction model with only the three best
# predictor variables included
# the already fitted m_movies.step model from above includes those variables and can be
 drawn into account
predict(m_movies.step, newdata = newmovie, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 75.86833 56.08264 95.65402
```

Hence, the model predicts, with 95% confidence, that a movie with an imdb_rating of 7.5, a runtime of 128 minutes and a critics_score of 52 % is expected to have an audience score on Rotten Tomatoes between 56.08264 and 95.65402. The current audience score for the movie The Accountant on Rotten Tomatoes is 81 % what is pretty close to the predicted value of 75.86833.

Part 6: Conclusion

Instructions:

A brief summary of your findings from the previous sections without repeating your statements from earlier as well as a discussion of what you have learned about the data and your research question. You should also discuss any shortcomings of your current study (either due to data collection or methodology) and include ideas for possible future research.

- Conclusion not repetitive of earlier statements (1 pt)
- Cohesive synthesis of findings that appropriate address the research question stated earlier (1 pt)
- Discussion of shortcomings (1 pt)

Altogether five new variables were derived from the movies dataset. The task was to learn about what attributes make a movie popular. In order to answer this question a Bayesian regression model was developed to predict audience_score from 16 explanatory variables included in the full movies dataset (old variables and additionally derived variables). The best predictive model derived from Bayesian Model Averaging and a stepAIC process kept three explanatory variables in the final model:

- imdb_rating
- critics_score
- runtime

The newly derived variables from the original data were not able to gain any additional predictive value after a BMA model selection process. The goal of every statistical modeling approach is a high generalizability to unknown data and final model should therefore be as sparse as possible according to Ockham´s Razor principle. Hence only three variables were kept in the final model.

The prediction results achieved by the model are promisingly and can be assessed as overall good. The random sample from the Rotten Tomatoes and IMDb API´s is large enough to achieve good research results. Nevertheless neither the dependent nor the independent variables were normal distributed and hence the provieded data slightly violated the model assumptions in terms of linear regression models (e.g. normally distributed errors). Further data transformation procedures weren´t very promisingly. Therefore other modeling approaches without those strict assumptions should be included in future research approaches (e.g. kNN-regression, SVM regression, generalized additive models, etc.).