Modeling and prediction for movies

Setup

Load packages

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

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")
```

Part 1: Data

The data was randomly sampled from movies produced and released before 2016, therefore the analysis result can be generalized to all movies produced and released before 2016.

No random assignment was used, so we can only conclude correlation between variables and no causality.

Part 2: Research question

I'm interested in how the ratings on IMDB are associated with various factors of movies, including year of release in theater, scores on Rotten Tomatoes (both from critics and audience), whether or not the movie was nominated for a best picture Oscar, and its genre (more specifically, whether the movie is a documentary or not).

I'm a huge fan of the movies, but I don't want to waste my time watching low-quality ones. Therefore, having a model that help predict a movie's ratings on IMDB can be quite useful when I am choosing which movie to watch. I will specifically pay attention to the documentary genre because I personally don't watch documentaries very often, so I'm wondering whether I have been missing out some great movies in that genre.

Part 3: Exploratory data analysis

I. Plots

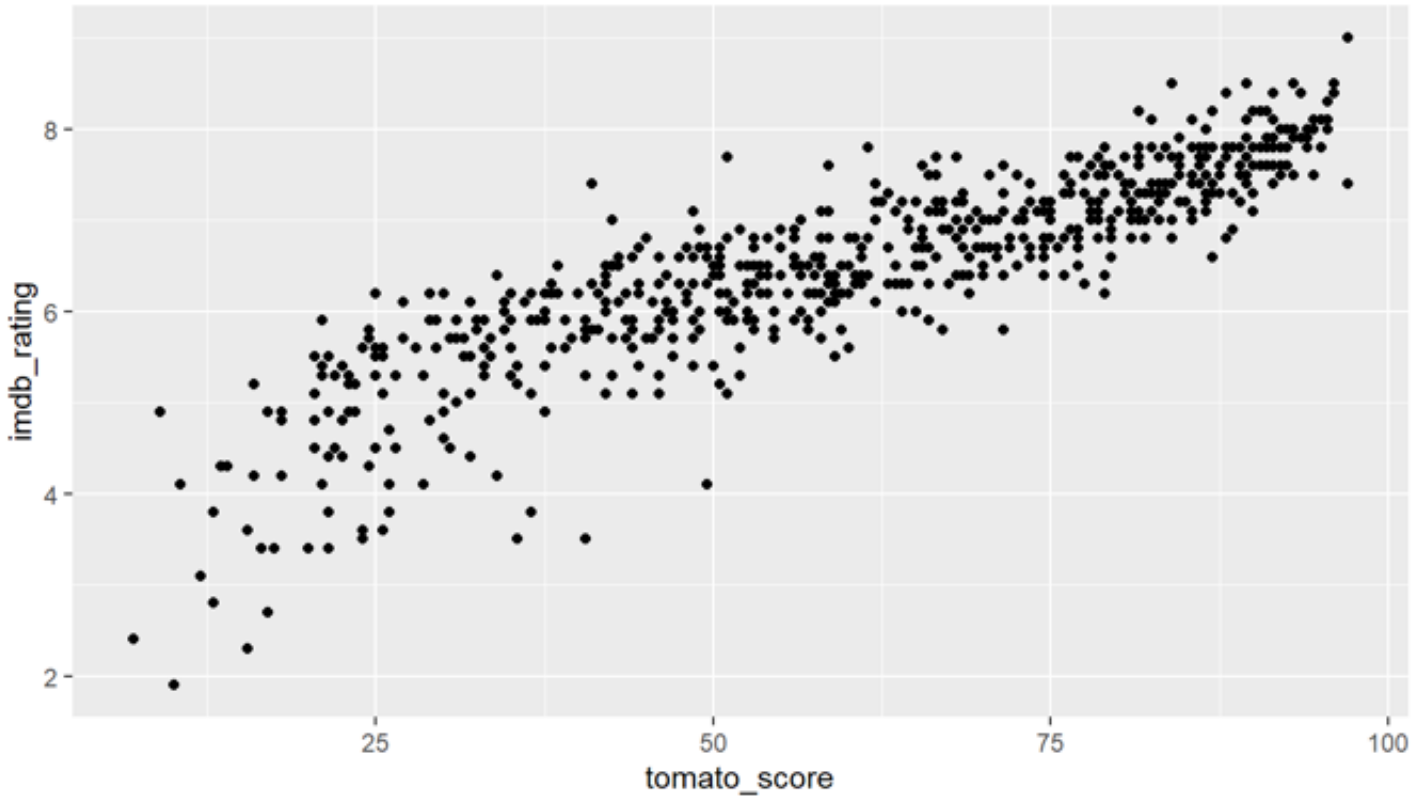
In this section, I'm going to explore whether the scores on Rotten Tomatoes are associated with the ratings on IMDB.

First, I need to combine the scores rated by the critics and that by the audience. I'll simply do so by creating a new variable 'tomato_score' that averages the two scores.

```
movies <- movies %>%
  mutate(tomato_score = (critics_score + audience_score)/2)
```

Then, plot a scatter plot between the variables 'tomato_score' and ratings on IMDB.

```
ggplot(data = movies, aes(x =tomato_score, y = imdb_rating)) +   geom_point()
```



The scatter plot exhibits a positive, moderately strong association between the two variables. Therefore, ratings on IMDB are positively correlated with scores on Rotten Tomatoes.

It is reasonable to expect that a movie with a higher score on Rotten Tomatoes tends to yield a higher rating on IMDB.

II. Summary Statistics

In this section, I'm going to explore whether movies that were nominated for a best picture Oscar tend to have higher ratings on IMDB.

```
movies %>%
  group_by(best_pic_nom) %>%
  summarise(IMDB_ratings_mean = mean(imdb_rating))
```

```
## # A tibble: 2 x 2
##   best_pic_nom IMDB_ratings_mean
## * <fct>                   <dbl>
## 1 no                       6.45
## 2 yes                      7.75
```

Movies that were nominated for a best picture Oscar on average have higher rating on IMDB compared to those that were not nominated. As a result, a movie that has been nominated to could tend to have a higher rating on IMDB compared to one that has not, but we still don't know whether the mean difference here is significant or due to sampling variability.

Part 4: Modeling

I. Variables Included

Variables included the full model are: year of release in theater, whether or not the movie was nominated for a best picture Oscar, scores on Rotten Tomatoes (both from critics and audience), and whether the movie is a documentary or not.

The latter two variables are not included in the original data and thus need to be created before we move on to build the model. The variable scores on Rotten Tomatoes has been created in the exploratory data analysis - plots section, so now I'm going to create the "whether documentary or not" variable.

```
movies <- movies %>%
  mutate(genre_documentary=ifelse(genre == "Documentary", "documentary", "not documen
         tary"))
```

II. Model Selection

Method

I will use adopt the "backwards elimination - p-value" method because it is relatively efficient and straight-forward.

Variable Selection

First, start with the full model. The variables include 1)year of release in theater, 2) scores on Rotten Tomatoes, 3) whether or not the movie was nominated for a best picture Oscar, and 4) its genre, or whether the movie is a documentary.

```
full_model <- lm(imdb_rating ~ thtr_rel_year + tomato_score + best_pic_nom +
        genre_documentary, data = movies)
summary(full_model)
```

```
##
## Call:
## lm(formula = imdb_rating ~ thtr_rel_year + tomato_score + best_pic_nom +
##     genre_documentary, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.54326 -0.27994  0.02486  0.34330  1.69109
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -1.920966   3.876812  -0.496   0.6204
## thtr_rel_year                    0.003037   0.001929   1.574   0.1159
## tomato_score                     0.041219   0.001015  40.594   <2e-16 ***
## best_pic_nomyes                  0.197073   0.118903   1.657   0.0979 .
## genre_documentarynot documentary -0.146536   0.082904  -1.768   0.0776 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5299 on 646 degrees of freedom
## Multiple R-squared:  0.7629, Adjusted R-squared:  0.7614
## F-statistic: 519.5 on 4 and 646 DF,  p-value: < 2.2e-16
```

As we can see, the variable with the greatest p-value is year of release in theater, so we delete that variable in our next step of model selection.

```
m1 <- lm(imdb_rating ~ tomato_score + best_pic_nom + genre_documentary, data =
        movies)
summary(m1)
```

```
##
## Call:
## lm(formula = imdb_rating ~ tomato_score + best_pic_nom + genre_documentary,
##     data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52078 -0.27955  0.02571  0.34469  1.70751
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.179669   0.112637  37.107   <2e-16 ***
## tomato_score                     0.041023   0.001009  40.663   <2e-16 ***
## best_pic_nomyes                  0.195626   0.119035   1.643    0.101
## genre_documentarynot documentary -0.169115   0.081747  -2.069    0.039 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5305 on 647 degrees of freedom
## Multiple R-squared:  0.7619, Adjusted R-squared:  0.7608
## F-statistic: 690.3 on 3 and 647 DF,  p-value: < 2.2e-16
```

Whether the movie has been nominated for the best picture Oscar is not a significant predictor in the revised model, so we get rid of that variable too.

```
m2 <- lm(imdb_rating ~ tomato_score + genre_documentary, data = movies)
summary(m2)
```

```
##
## Call:
## lm(formula = imdb_rating ~ tomato_score + genre_documentary,
##     data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.50806 -0.28624  0.02806  0.34758  1.70734
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.1445312  0.1107347   37.43   <2e-16 ***
## tomato_score                    0.0414385  0.0009779   42.37   <2e-16 ***
## genre_documentarynot documentary -0.1508539  0.0810942   -1.86   0.0633 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5312 on 648 degrees of freedom
## Multiple R-squared:  0.761,  Adjusted R-squared:  0.7602
## F-statistic:  1031 on 2 and 648 DF,  p-value: < 2.2e-16
```

Still, the genre variable is not significant, so we can't include that variable in our model either. As result, our final model would be a single linear regression model with scores on Rotten Tomatoes as the only explanatory variable.

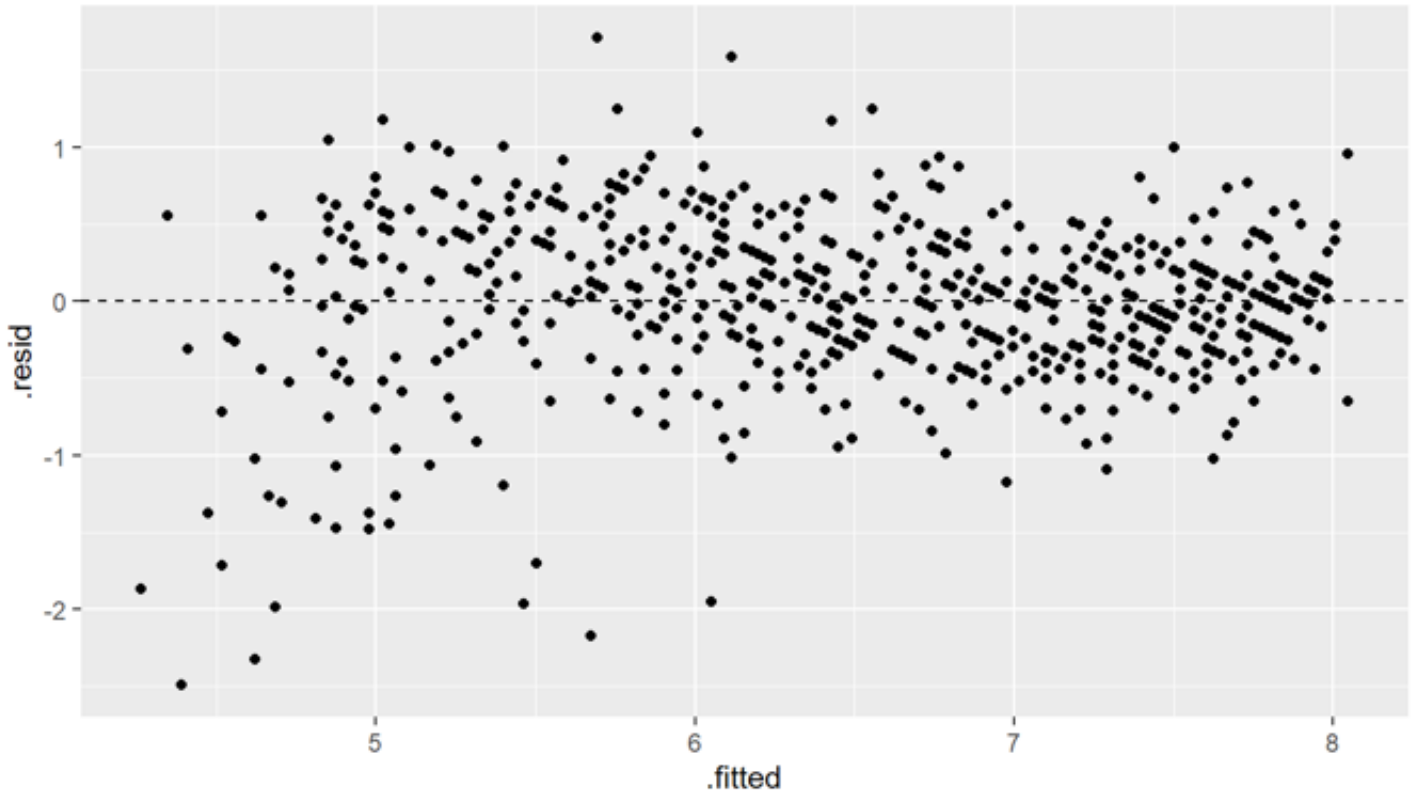
```
final_model <- lm(imdb_rating ~ tomato_score , data = movies)
summary(final_model)
```

```
##
## Call:
## lm(formula = imdb_rating ~ tomato_score, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.49086 -0.28871  0.02627  0.35078  1.70642
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.9706323  0.0594689   66.77   <2e-16 ***
## tomato_score 0.0420232  0.0009278   45.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5322 on 649 degrees of freedom
## Multiple R-squared:  0.7597, Adjusted R-squared:  0.7593
## F-statistic:  2052 on 1 and 649 DF,  p-value: < 2.2e-16
```

Model Diagnostics

1. linear relationship between explanatory and response variables.

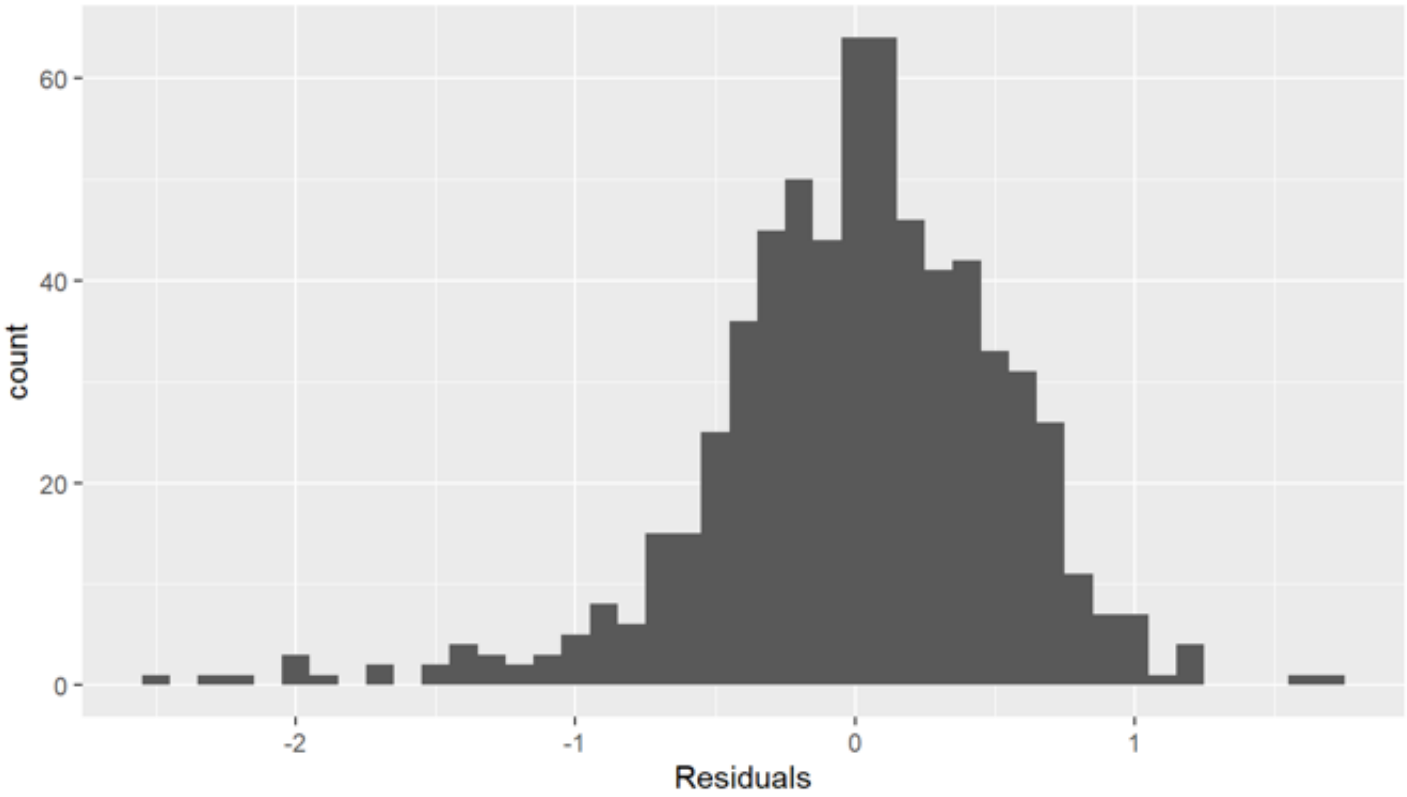
```
ggplot(data = final_model, aes(x = .fitted, y = .resid)) +
geom_point() +
geom_hline(yintercept = 0, linetype = "dashed")
```



There seems to exist a random scatter around 0 in the plot. However, the dots have a tendency to gather on the right side of the plot.

2. nearly normal residuals

```
ggplot(data = final_model, aes(x = .resid)) +
geom_histogram(binwidth = 0.1) +
xlab("Residuals")
```



The residual histogram displays a nearly normal distribution centered at 0.

3. constant variability As we can refer back to the residual plot, the variability seems to be larger on the left side of the plot than to the right side. However, taken into consideration the size of the sample size, it could be regarded as constant variability in general.
4. independent residuals The data were collected through random selection, thus we can expect this condition to be met.

Interpretation of model coefficients

1. slope = 0.0420232

All else held constant, for each 1 point increase in scores on Rotten Tomatoes, the model predicts the movie's rating on IMDB to increase on average by 0.042 point.

2. R-squared = 0.7597 The linear regression model with scores on Rotten Tomatoes as the only explanatory variable can explain 75.97% variability in ratings on IMDB.

Part 5: Prediction

The movie I will use for prediction is one of my favorites: How to Train Your Dragon. This movie has a 95 scores on Rotten Tomatoes (critics 99, audience 91). The reference for the data is as follows: https://www.rottentomatoes.com/m/how_to_train_your_dragon

(https://www.rottentomatoes.com/m/how_to_train_your_dragon)

Now, to generate a 95% confidence interval for the rating on IMDB of this movie.

```
prediction_dataframe <- data.frame(tomato_score = 95)
predict(final_model, prediction_dataframe)
```

```
##        1
## 7.962833
```

```
predict(final_model, prediction_dataframe, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 7.962833 6.915083 9.010584
```

We can be 95% confident that the model predicts that the rating on IMDB of How to Train Your Dragon are within the range 6.9 points to 9.0 points, which captures the movie's true rating on IMDB by the way.

Part 6: Conclusion

In conclusion, ratings on IMDB are significantly associated with only one variable: the average score of the movie's critics score and audience score on Rotten tomatoes, which is not surprising actually.

However, one shortcoming of this model is that many other variables that could possibly predict the ratings on IMDB are not included in this set of data, such as time of production or money invested to the money, which are all reasonable correlations with the quality of a movie. Further research can look at and try out models including those variables.