Statpadders

Toma Shigaki-Than, CJ Frederickson, Camden Reeves, Sam Kakarla

2025-03-17

```
library(tidyverse)
library(tidymodels)
library(dplyr)
library(patchwork)
library(corrplot)
library(openintro)
library(knitr)
library(kableExtra)  # for table embellishments
library(Stat2Data)
library(broom)
library(yardstick)
imdb_top_1000 <- read_csv("data/imdb_top_1000.csv") |>
  drop_na()
```

Introduction

Online review platforms have transformed how audiences evaluate and choose movies. Between professional critics and everyday viewers, these platforms host a broad spectrum of opinions that increasingly shape the trajectory of films, from box office performance to streaming recommendations. However, these two groups often evaluate films through very different lenses: critics tend to emphasize artistic merit, narrative structure, and technical execution, while audiences may prioritize entertainment, emotional resonance, and accessibility. As a result, it is commonplace for a film to receive mixed signals across platforms. A film may be highly rated by audiences but poorly reviewed by critics, or vice versa.

This divergence can be confusing for consumers, who must navigate these conflicting assessments to make viewing decisions. For example, a film might thrive on word-of-mouth and accumulate high audience ratings on IMDb but perform poorly with critics or fail to receive awards recognition.

Motivation and Importance

Understanding these differences is essential not just for filmgoers but also for industry stakeholders. Movie studios, marketers, and streaming platforms rely on both critical acclaim and mass appeal to determine promotional strategies, content investments, and recommendation algorithms. Prior research has shown that user-driven ratings have powerful word-of-mouth effects, often extending a film's relevance and commercial success (Moon, Bergey, Iacobucci, 2010). Further, the typical consumer does not evaluate films with the same lens or criteria as professional critics.

By analyzing these patterns, we aim to identify the film characteristics, such as decade of release, runtime, gross earnings, and censorship rating, that are most predictive of audiences liking a film more than critics. These insights have direct implications for consumer behavior research, content recommendation systems, and media marketing strategies in an increasingly data-driven entertainment landscape.

Given the observed divergence between critic and audience evaluations, we pose the following research question:

What factors contribute to audiences liking a movie more than critics, and how can we use these factors to predict the likelihood of a film performing better with audiences than critics?

In this study, we focus on modeling the odds that audience scores exceed critic scores. By doing so, we shift attention from understanding average film quality to identifying conditions that foster fan-favorite films and resonate with general audiences even when they fail to win over the critics.

Data

This data is taken from the top 1000 movies on IMDB, obtained through data scraping. In cleaning our data, we first had to deal with the NA values in our data. Some observations had NA values in their Gross Revenues. After examining these observations, there were no discernible patterns or connections between the NA values; they were random. As such, we were able to drop these values without compromising our data set or losing important observations. We also created the variable `difference`, whose value indicates the difference between IMDB Score and MetaScore, scaled so that they can be compared. A negative value indicates that the audience score is lower than that of critics, and positive is vice versa. Because our dataset includes older and international films with various outdated or uncommon censorship ratings, we consolidated certificates into modern, widely recognizable categories: G, PG, PG-13, R, and Other, based on their closest equivalents in the current U.S. rating system.

Predictors:

- Runtime (numerical)

- Gross Revenue (numerical)
- Censorship Certificate (categorical)
- Decade Released (categorical)
- Number of Votes (numerical)

Response Variable:

- Difference: *The quantity that the IMDB score (score given by audience/fans on IMDB, scaled to match MetaScore) differs from the MetaScore (aggregate score of critics' ratings)*

We additionally created a variable `difference_binary` based on the difference between IMDb audience scores and critic MetaScores. To determine meaningful divergence, we classified films into two categories:

1 (divergent): if the difference was less than -13 or greater than 9, corresponding approximately to films with audience-critic score differences exceeding ±1 standard deviation from the mean.

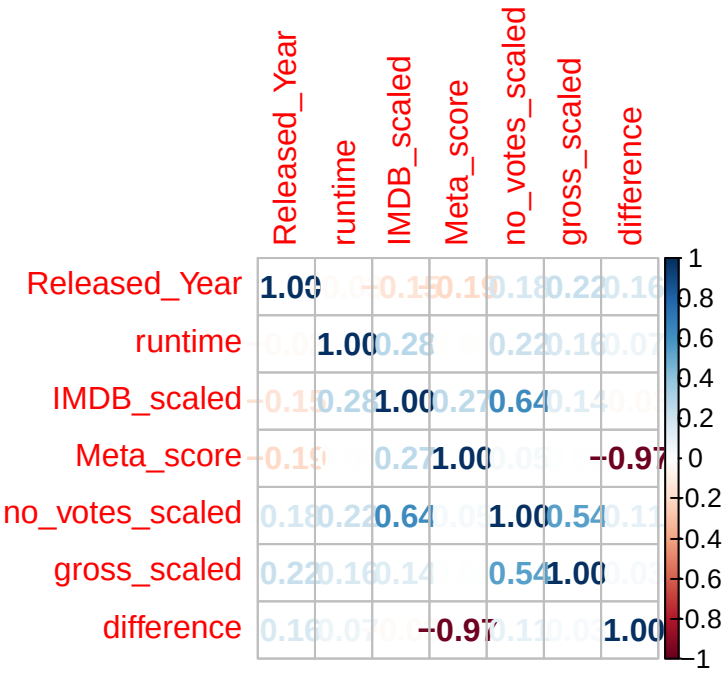
0 (non-divergent): otherwise.

This threshold approach is grounded in the properties of the original difference distribution, which was approximately normal with a slight left skew. With a mean of approximately -2.2 and a standard deviation of about 11.9, setting thresholds at roughly one standard deviation above and below the mean allowed us to identify films with statistically meaningful deviations rather than minor fluctuations. This method balances capturing important fan-favorite or controversial films while maintaining a reasonable sample size for modeling.

```
imdb_top_1000 <- imdb_top_1000 |>
  mutate(no_votes_scaled = No_of_Votes / 10^6,
         gross_scaled = Gross / 10^6,
         gross_cent = scale(gross_scaled, center = TRUE, scale = FALSE),
         IMDB_scaled = IMDB_Rating *10,
         Released_Year = if_else(Series_Title == "Apollo 13", "1995",
                                 Released_Year),
         difference = IMDB_scaled - Meta_score,
         difference_binary = factor(if_else(difference < -13 | difference > 9 , 1, 0)),
         runtime = as.numeric(str_remove(Runtime, " min")),
         Released_Year = as.numeric(Released_Year),
         decade = case_when(Released_Year < 1940 ~ "1930s",
                            Released_Year >= 1940 & Released_Year < 1950 ~ "1940s",
                            Released_Year >= 1950 & Released_Year < 1960 ~ "1950s",
                            Released_Year >= 1950 & Released_Year < 1960 ~ "1950s",
                            Released_Year >= 1960 & Released_Year < 1970 ~ "1960s",
```

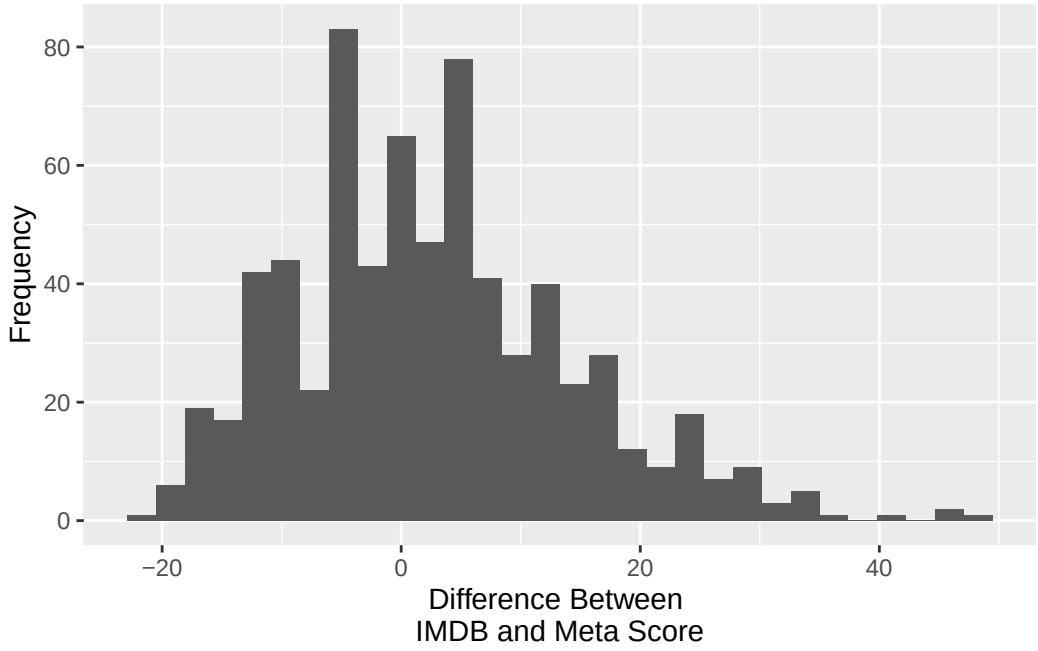
```
                          Released_Year >= 1970 & Released_Year < 1980 ~ "1970s",
                          Released_Year >= 1980 & Released_Year < 1990 ~ "1980s",
                          Released_Year >= 1990 & Released_Year < 2000 ~ "1990s",
                          Released_Year >= 2000 & Released_Year < 2010 ~ "2000s",
                          Released_Year >= 2010 ~ "2010s",),
        certificate = case_when(
      Certificate %in% c("G", "U", "APPROVED", "PASSED") ~ "G",
      Certificate %in% c("PG", "TV-PG", "GP", "UA", "U/A") ~ "PG",
      Certificate %in% c("PG-13") ~ "PG-13",
      Certificate %in% c("R", "A") ~ "R",
      TRUE ~ "Other"),
      runtime_cent = scale(runtime, center = TRUE, scale = FALSE),
      votes_cent = scale(no_votes_scaled, center = TRUE, scale = FALSE),

  )

imdb_top_1000 <- imdb_top_1000 %>%
  filter(!(decade %in% c("1930s", "1940s")))
```

```
#geeksforgeeks.org/correlation-matrix-in-r-programming

matrix <- imdb_top_1000 %>%
  select(Released_Year, runtime, IMDB_scaled, Meta_score, no_votes_scaled,
         gross_scaled, difference)
c <- cor(matrix)
corrplot(c, method = "number")
```



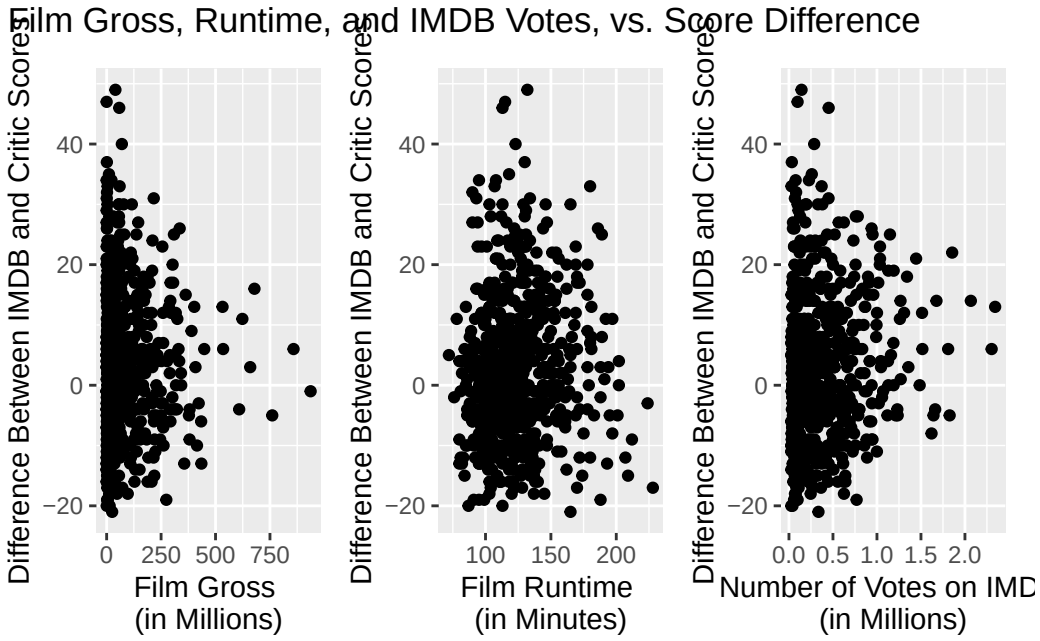
We used a correlation matrix to check for multicollinearity. As expected, difference is highly correlated with `IMDB_scaled` and `meta_score`, since it's derived from them. IMDB_scaled and no_votes_scaled also show strong correlation (0.62), so we avoid including both in the same model.

```
imdb_top_1000 %>%
  ggplot(aes(x = difference )) +
  geom_histogram() +
   labs(
    x = "Difference Between \nIMDB and Meta Score",
    y = "Frequency"
  )
```




```
library(patchwork)
p1 <- imdb_top_1000 %>%
  ggplot(aes(x = gross_scaled, y = difference )) +
  geom_point() + labs(
    x = "Film Gross \n(in Millions)",
    y = "Difference Between IMDB and Critic Scores"
  )
p2 <- imdb_top_1000 %>%
  ggplot(aes(x = runtime, y = difference)) +
  geom_point() +
   labs(
    x = "Film Runtime \n(in Minutes)",
    y = "Difference Between IMDB and Critic Scores"
   )

p3 <- imdb_top_1000 %>%

  ggplot(aes(x = no_votes_scaled, y = difference )) +
  geom_point() + labs(
    x = "Number of Votes on IMDB \n(in Millions)",
    y = "Difference Between IMDB and Critic Scores"
  )
p1 + p2 + p3 +
```

```
  plot_annotation('Film Gross, Runtime, and IMDB Votes, vs. Score Difference')
```



In terms of the `difference` between the two scores, it seems that the values are almost normally distributed, with a slight left skew. This suggests that it is nearly equally common for a IMDB Score to be either higher or lower than the MetaScore score, though slightly more often lower.

Additionally, when plotting our numerical predictors, there appear to be no linear relationships between each predictor and our response.

Methodology

```
imdb_linear <- lm(difference ~ gross_cent + runtime_cent + decade + certificate
                  + no_votes_scaled + gross_cent*certificate, data =
                    imdb_top_1000)

imdb_full_log <- glm(difference_binary ~ gross_cent + runtime_cent + decade +
certificate + votes_cent + gross_cent * certificate,
data = imdb_top_1000, family = "binomial")

#imdb_linear <- lm(difference ~ gross_cent + runtime_cent + decade + certificate
```

```
    #             + no_votes_scaled + gross_cent*certificate, data =
     #              imdb_top_1000)

#imdb_linear_no_cert <- lm(difference ~ gross_cent + runtime_cent + decade, data =
      #              imdb_top_1000)

tidy(imdb_linear)
```

```
# A tibble: 18 x 5
   term                        estimate std.error statistic       p.value
   <chr>                          <dbl>     <dbl>     <dbl>         <dbl>
 1 (Intercept)                -12.4      2.74       -4.51  0.00000753
 2 gross_cent                  -0.0127   0.00725    -1.75  0.0799
 3 runtime_cent                 0.0271   0.0176      1.54  0.124
 4 decade1960s                  7.61     3.21        2.37  0.0179
 5 decade1970s                  9.56     3.07        3.12  0.00191
 6 decade1980s                 14.3      2.95        4.85  0.00000151
 7 decade1990s                 16.7      2.84        5.90  0.00000000562
 8 decade2000s                 16.2      2.78        5.83  0.00000000863
 9 decade2010s                 12.2      2.82        4.34  0.0000167
10 certificateOther            -4.37    31.9        -0.137 0.891
11 certificatePG                1.55     1.31        1.18  0.237
12 certificatePG-13             1.53     2.52        0.608 0.543
13 certificateR                 1.40     1.16        1.21  0.229
14 no_votes_scaled              1.54     1.52        1.01  0.312
15 gross_cent:certificateOther -0.141    0.454      -0.311 0.756
16 gross_cent:certificatePG     0.0122   0.00878     1.39  0.166
17 gross_cent:certificatePG-13  0.0275   0.0363      0.758 0.449
18 gross_cent:certificateR      0.0290   0.0126      2.31  0.0214
```

```
tidy(imdb_full_log)
```

```
# A tibble: 18 x 5
   term                          estimate   std.error statistic p.value
   <chr>                            <dbl>       <dbl>     <dbl>   <dbl>
 1 (Intercept)                   -0.152        0.511     -0.297  0.767
 2 gross_cent                    -0.00214      0.00165   -1.30   0.194
 3 runtime_cent                   0.00948      0.00336    2.82   0.00481
 4 decade1960s                   -0.771        0.631     -1.22   0.222
 5 decade1970s                   -1.27         0.606     -2.10   0.0353
 6 decade1980s                   -0.725        0.559     -1.30   0.195
```

```
 7 decade1990s                  -0.453        0.533     -0.851  0.395
 8 decade2000s                  -0.569        0.523     -1.09   0.277
 9 decade2010s                  -0.775        0.533     -1.45   0.146
10 certificateOther           -538.        17550.        -0.0306 0.976
11 certificatePG                 0.240        0.252      0.953  0.341
12 certificatePG-13             -0.0467       0.490     -0.0952 0.924
13 certificateR                  0.0128       0.226      0.0569 0.955
14 votes_cent                    0.377        0.287      1.31   0.189
15 gross_cent:certificateOther  -7.03         231.        -0.0305 0.976
16 gross_cent:certificatePG      0.000311     0.00191     0.163  0.870
17 gross_cent:certificatePG-13   0.00299      0.00704     0.424  0.672
18 gross_cent:certificateR       0.00318      0.00252     1.26   0.207
```

```
glance(imdb_linear) %>%
  select(adj.r.squared)
```

```
# A tibble: 1 x 1
  adj.r.squared
          <dbl>
1        0.0881
```

```
#glance(imdb_linear_no_cert) %>%
 # select(adj.r.squared)
```

We initially explored a linear regression approach to model the difference between IMDb audience scores and critic MetaScores. However, diagnostics (including the above 8.81% adjusted r squared value) revealed no clear linear relationship between the response variable and the numerical predictors. Given this, and the bounded, binary nature of our eventual outcome (whether a film's audience score significantly diverges from critic scores), we transitioned to a logistic regression framework.

In constructing the logistic model, we first fit a full logistic regression model including: Gross revenue, centered for interpretability: `gross_cent`

Runtime in minutes, centered for interpretability: `runtime_cent`

Decade of release: `decade`

Number of votes, centered for interpretability: `votes_cent`

Censorship certificate: `certificate`

An interaction term between gross revenue and certificate `gross_cent` * `certificate`

```r
#just getting rid of certificate interaction
imdb_reduced_log <- glm(difference_binary ~ gross_cent + runtime_cent + decade + certificate
votes_cent, data = imdb_top_1000,
family = "binomial")

#getting rid of certificate altogether
imdb_log_noCert <- glm(difference_binary ~ gross_cent + runtime_cent + decade +
votes_cent, data = imdb_top_1000,
family = "binomial")

tidy(imdb_reduced_log)
```

```
# A tibble: 14 x 5
   term             estimate std.error statistic p.value
   <chr>               <dbl>     <dbl>     <dbl>   <dbl>
 1 (Intercept)      -0.118    0.501       -0.236 0.814
 2 gross_cent       -0.00158  0.000965    -1.64  0.102
 3 runtime_cent      0.00985  0.00332      2.97  0.00298
 4 decade1960s      -0.840    0.614       -1.37  0.171
 5 decade1970s      -1.28     0.596       -2.15  0.0317
 6 decade1980s      -0.754    0.550       -1.37  0.170
 7 decade1990s      -0.491    0.523       -0.940 0.347
 8 decade2000s      -0.614    0.513       -1.20  0.231
 9 decade2010s      -0.794    0.522       -1.52  0.128
10 certificateOther -1.29     1.18        -1.09  0.274
11 certificatePG     0.215    0.247        0.870 0.384
12 certificatePG-13 -0.123    0.419       -0.294 0.769
13 certificateR     -0.0648   0.219       -0.296 0.768
14 votes_cent        0.440    0.282        1.56  0.119
```

```r
tidy(imdb_log_noCert)
```

```
# A tibble: 10 x 5
   term         estimate std.error statistic p.value
   <chr>           <dbl>     <dbl>     <dbl>   <dbl>
 1 (Intercept)  -0.308    0.450       -0.684 0.494
 2 gross_cent   -0.00130  0.000911    -1.43  0.153
 3 runtime_cent  0.00978  0.00329      2.97  0.00298
 4 decade1960s  -0.713    0.588       -1.21  0.226
 5 decade1970s  -1.09     0.563       -1.93  0.0537
 6 decade1980s  -0.550    0.516       -1.07  0.287
```

```
 7 decade1990s  -0.299    0.487       -0.614 0.539
 8 decade2000s  -0.411    0.475       -0.866 0.386
 9 decade2010s  -0.565    0.483       -1.17  0.242
10 votes_cent    0.452    0.276        1.64  0.101
```

```
anova(imdb_reduced_log, imdb_full_log, test = "Chisq") %>%
  tidy() %>%
  kable(digits = 3)
```

term	df.residual	residual.deviance	df	deviance	p.value
difference_binary ~ gross_cent + runtime_cent + decade + certificate + votes_cent	681	840.351	NA	NA	NA
difference_binary ~ gross_cent + runtime_cent + decade + certificate + votes_cent + gross_cent * certificate	677	833.676	4	6.675	0.154

```
#with certificate and interaction vs. no certificate
anova(imdb_log_noCert, imdb_full_log, test = "Chisq") %>%
  tidy() %>%
  kable(digits = 3)
```

term	df.residual	residual.deviance	df	deviance	p.value
difference_binary ~ gross_cent + runtime_cent + decade + votes_cent	685	843.572	NA	NA	NA
difference_binary ~ gross_cent + runtime_cent + decade + certificate + votes_cent + gross_cent * certificate	677	833.676	8	9.896	0.272

```
glance(imdb_full_log)
```

```
# A tibble: 1 x 8
  null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
1          865.     694  -417.  870.  951.     834.         677   695
```

```
glance(imdb_log_noCert)
```

```
# A tibble: 1 x 8
  null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
1          865.     694  -422.  864.  909.     844.         685   695
```

```
#glance(imdb_reduced_log)
```

Upon evaluating the model, the main effects of certificate were not statistically significant (high p-values). However, we hypothesized from our EDA that gross revenue may interact with certificate rating. This is to assert, for instance, that commercial performance may matter differently for PG-13 versus R-rated films.

As such, we performed a drop-in-deviance test comparing the full model (with certificate and interaction) against a reduced model excluding the interaction terms. The results showed that including the gross × certificate interaction improved model fit, though modestly. Specifically, the deviance decreased with the interaction included, suggesting that there may be some moderating effect of the certificate on gross revenue's impact. Given this, we then narrowed our model comparison to two candidates: a model with certificate and the gross × certificate interaction terms included, and a model with no certificate variables at all (predictors: gross_cent + runtime_cent + decade + votes_cent).

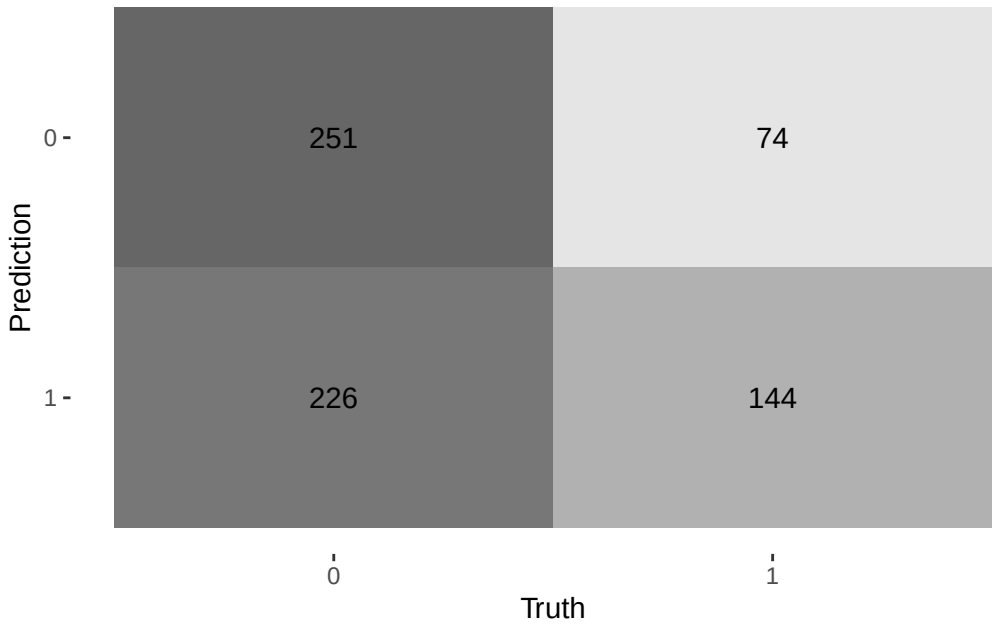
To formally select between them, we compared their AIC and BIC values. Both AIC and BIC were lower for the model without certificate variables. Since BIC penalizes model complexity more heavily than AIC, and favors simpler models, it is important to note that we would expect BIC to favor the reduced model. However, the fact that both AIC and BIC favored the simpler model gave strong evidence that removing `certificate` and its interaction was justified. Thus, despite the modest improvement in deviance with the interaction term, the penalized model selection criteria (AIC/BIC) led us to eliminate the censorship certificate variable and its interaction terms from the final model.

The final logistic regression model, used to predict whether a movie's IMDb audience score diverged significantly from its critic MetaScore, includes centered gross revenue, centered runtime, decade of release, and centered number of votes as predictors.

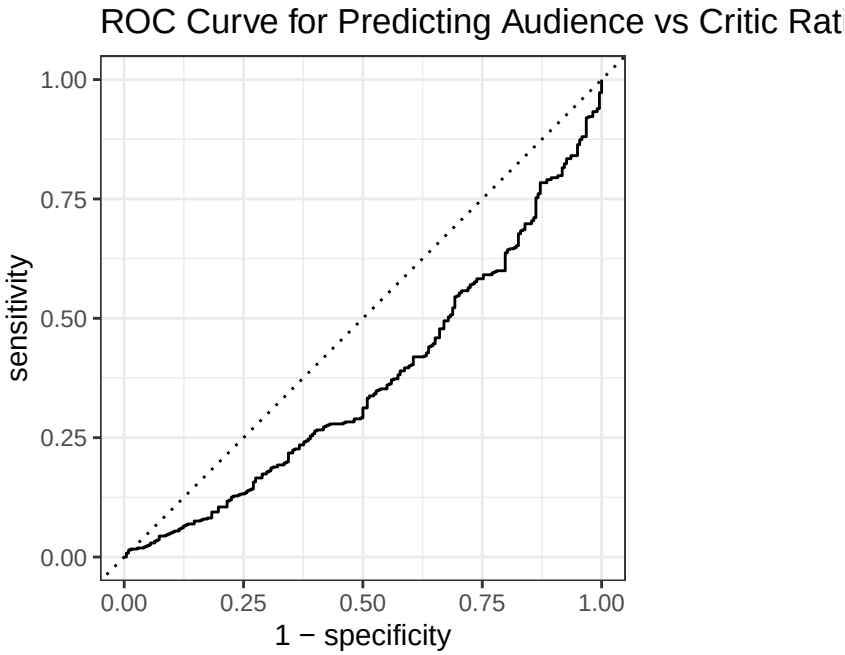
Results

After fitting the final logistic regression model using `gross_cent`, `runtime_cent`, `decade`, and `votes_cent` as predictors, we initially evaluated its performance using a default classification threshold of 0.5. However, under this threshold, the model overwhelmingly predicted the majority class (0), making it largely ineffective at identifying films where audience ratings diverged significantly from critic ratings. Very few films were classified as having a major divergence, despite a meaningful portion of the dataset meeting that condition. Recognizing that our binary outcome was based on ± 1 standard deviation from the mean difference (difference $<$ -13 or $>$ 9 being classified as 1), and thus rare by construction, we addressed this imbalance by lowering the classification threshold to 0.3. Under the adjusted 0.3 threshold, the model produced the following confusion matrix:

```
library(yardstick)
library(dplyr)
library(ggplot2)

predictions <- imdb_top_1000 %>%
  mutate(
    .pred_1 = predict(imdb_full_log, newdata = imdb_top_1000, type = "response"),  # NAME TH
    pred_class = factor(if_else(.pred_1 > 0.3, "1", "0")),
    difference_binary = factor(difference_binary)
  )

# Step 2: Confusion matrix (optional)
conf_mat(predictions, truth = difference_binary, estimate = pred_class) |>
  autoplot(type = "heatmap")
```




```
# Step 3: ROC Curve Data
roc_curve_data <- roc_curve(predictions, truth = difference_binary, .pred_1)

# Step 4: Plot ROC Curve
autoplot(roc_curve_data) +
  ggtitle("ROC Curve for Predicting Audience vs Critic Rating Divergence")
```




```
# Step 5: Calculate AUC
roc_auc(predictions, truth = difference_binary, .pred_1)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.372
```

```
# If you want even more detail:
classification_metrics <- predictions %>%
  metrics(truth = difference_binary, estimate = pred_class) %>%
  bind_rows(
    recall(predictions, truth = difference_binary, estimate = pred_class),
    precision(predictions, truth = difference_binary, estimate = pred_class),
    specificity(predictions, truth = difference_binary, estimate = pred_class)
  )

classification_metrics
```

```
# A tibble: 5 x 3
  .metric     .estimator .estimate
  <chr>       <chr>          <dbl>
```

```
1 accuracy    binary         0.568
2 kap         binary         0.157
3 recall      binary         0.526
4 precision   binary         0.772
5 specificity binary         0.661
```

In evaluating model performance, we considered whether to prioritize sensitivity or specificity. Sensitivity (52.6%) measures the model's ability to correctly identify films where audience scores diverge significantly from critic scores — the rare but important "fan-favorite" or "controversial" cases. Specificity (66.1%) measures the ability to correctly classify typical films with no major divergence. Given that our research question focuses on understanding what factors drive major rating divergence, prioritizing sensitivity is more appropriate. Missing a fan-favorite or controversial film (a false negative) would be more detrimental to the purpose of our study than incorrectly flagging a typical film (a false positive).

Our goal is to capture as many truly divergent films as possible, even if it means accepting a slightly higher false positive rate.Lowering the threshold to 0.3 significantly improved the model's sensitivity, allowing it to capture a much higher proportion of the divergent films. However, this came at the expected cost of increasing false positives, slightly reducing specificity and overall precision.

To assess the model's overall discriminative ability independently of any fixed threshold, we generated a ROC curve. The Area Under the Curve (AUC) was calculated to be approximately 0.37. An AUC of 0.5 would suggest random guessing, so a score of 0.37 indicates the model performs worse than random at ranking films according to their likelihood of audience-critic divergence. This suggests that the predictors included in the model are not strongly informative in distinguishing between divergent and non-divergent films across a wide range of thresholds.

Our final logistic regression model demonstrates that while adjusting the classification threshold to 0.3 improved practical classification performance, the model still struggles overall to differentiate between divergent and non-divergent movies. This limitation is understandable given the nature of the data. Factors like box office gross and runtime are broad structural characteristics that do not capture the nuanced reasons why audiences and critics might disagree. Additionally, over the decades, the relationship between commercial success and critical reception has shifted dramatically — blockbuster franchises, streaming releases, and genre preferences have evolved substantially, reducing the consistency of patterns. Since genre, storytelling trends, cultural resonance, and marketing dynamics are not directly captured in the predictors available, it can be understood that the model struggles to predict audience-critic divergence based on basic film attributes alone. Future research could benefit from incorporating content-based features such as genre classification, award nominations, review sentiment analysis, or even social media metrics to build more powerful predictive models.

Appendix

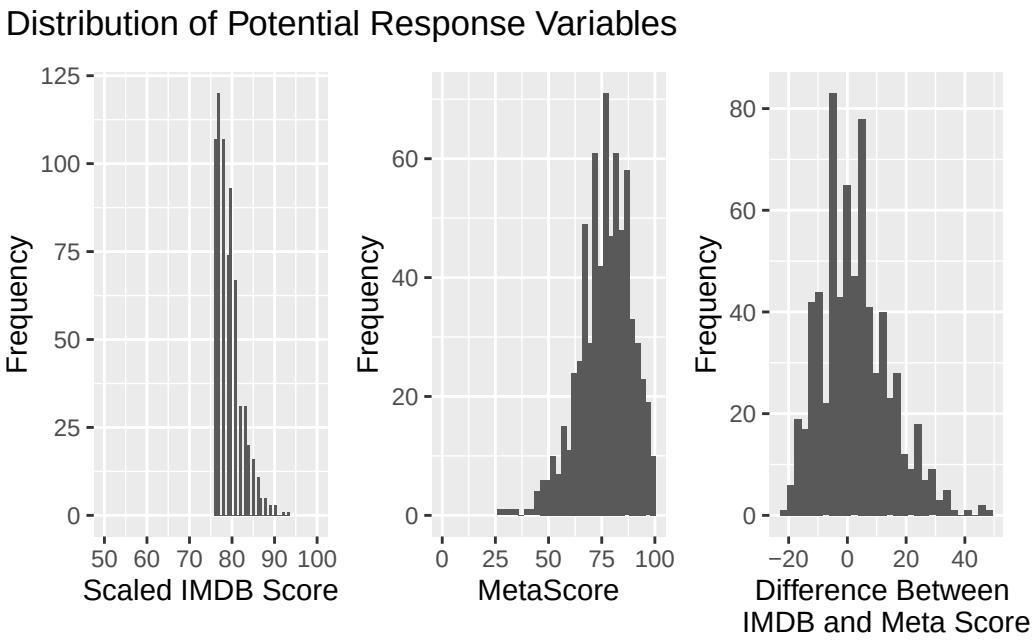
To begin our EDA, we first had to deal with the NA values in our data. Some observations had NA values in their Gross Revenues. After examining these observations, there were no discernible patterns or connections between the NA values; they were random. As such, we were able to drop these values without compromising our data set or losing important observations. We also created the variable `difference`, whose value indicates the difference between MetaScore and IMDB Score, scaled so that they can be compared. A negative value indicates that the MetaScore is lower than IMDB Score, and a positive value indicates that it is higher. Further, we turned our year predictor into a categorical variable by creating a new variable: `decade`. Since there is a very wide range of values in `Released_Year` for the movies selected, that variable itself is not particularly useful for our analysis. Not many observations even had the same released year, and the differences between one unit in that variable were arbitrary for some movies (for example a movie released in 1966 vs 1967 does not give much insight). For data cleaning and to improve clarity and interpretability, we changed this variable into a categorical variable `decade`, where all of the years released are grouped into decades (i.e. 1950s, 1960s, etc.). This categorical approach gives better interpretability; grouping movies into decades creates a better identifier than simply using individual years.

Additionally, the variable `Runtime` listed the runtime of each observation as a string with the number of minutes followed by the word "mins." For example, a movie 90 minutes long would be listed as the string "90 mins" instead of the number 90. As such, this made `Runtime` a categorical variable. We changed this by removing the "mins" label and refactoring it as numeric, thus making the `Runtime` into a numerical variable.

Univariate EDA

```
p1 <- imdb_top_1000 %>%
  ggplot(aes(x = IMDB_scaled )) +
  geom_histogram() +
  coord_cartesian(xlim = c(50, 100)) +
  labs(
    x = "Scaled IMDB Score",
    y = "Frequency"
  )
p2 <- imdb_top_1000 %>%
  ggplot(aes(x = Meta_score )) +
  geom_histogram() +
  coord_cartesian(xlim = c(0, 100)) +
   labs(
```

```
    x = "MetaScore",
    y = "Frequency"
  )
p3 <- imdb_top_1000 %>%
  ggplot(aes(x = difference )) +
  geom_histogram() +
   labs(
    x = "Difference Between \nIMDB and Meta Score",
    y = "Frequency"
  )

p1 + p2 + p3 + plot_annotation('Distribution of Potential Response Variables')
```




```
imdb_top_1000 %>%
  summarise(
    mean_IMDB_Score = mean(IMDB_scaled),
    med_IMDB_Score = median(IMDB_scaled),
    sd_IMDB_Score = sd(IMDB_scaled),
    IQR_IMDB_Score = IQR(IMDB_scaled),
    min_IMDB_Score = min(IMDB_scaled),
    max_IMDB_Score = max(IMDB_scaled),
    mean_MetaScore = mean(Meta_score),
```

```
    med_MetaScore = median(Meta_score),
    sd_MetaScore = sd(Meta_score),
    IQR_MetaScore = IQR(Meta_score),
    min_MetaScore = min(Meta_score),
    max_MetaScore = max(Meta_score),
    mean_Difference = mean(difference),
    med_Difference = median(difference),
    sd_Difference = sd(difference),
    IQR_Difference = IQR(difference),
    min_Difference = min(difference),
    max_Difference = max(difference)
  ) %>%
  pivot_longer(cols = everything(),
               names_to = "Statistic",
               values_to = "Value") %>%
  separate(Statistic, into = c("Measure", "Variable"), sep = "_", extra = "merge") %>%
  pivot_wider(names_from = Measure, values_from = Value)
```

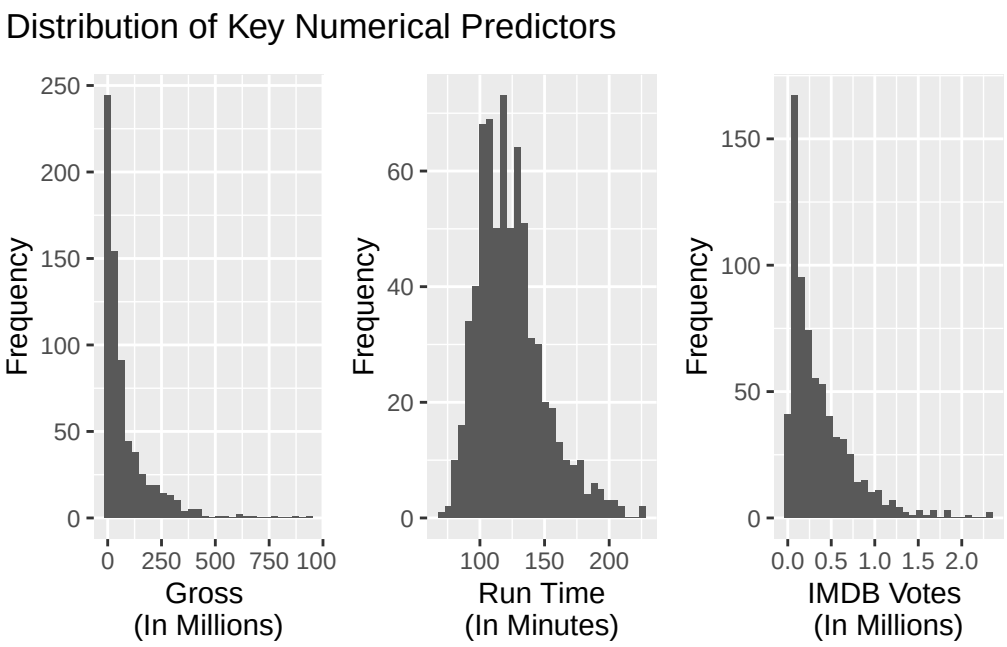
```
# A tibble: 3 x 7
  Variable    mean   med    sd   IQR   min   max
  <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 IMDB_Score 79.3     79  2.94     4    76    93
2 MetaScore  76.7     77 12.2     16    28   100
3 Difference  2.62     2 11.8     16   -21    49
```

As we can see from our response variables, scaled IMDB Score seems to be skewed right for these films, and scores tend to trend between 76 and 93. Both the mean score and median score are about 79, standard deviation of about 3, IQR of 4, and a range of 17.

The distribution for MetaScore seems to be skewed right, with a mean of about 77, a median about 78, a standard deviation of about 12, an IQR of 16 and a range of 72.

Furthermore in terms of the `difference` between the two scores, it seems that the values are almost normally distributed, with a slight left skew. This suggests that it is nearly equally common for a MetaScore to be either higher or lower than the IMDB score, though slightly more often lower. There is an outlier when MetaScore is about 49 points lower than IMDB score (-49). The mean `difference` is when about MetaScore is about two points lower than IMDB score (-2) and median at 1 point lower (-1). There is standard deviation of 12 points, IQR of about 15 points, and a range of 70 points.

```r
p1 <- imdb_top_1000 %>%
  ggplot(aes(x = gross_scaled)) +
  geom_histogram() + labs(
    x = "Gross \n(In Millions)",
    y = "Frequency"
  )
p2 <- imdb_top_1000 %>%
  ggplot(aes(x = runtime )) +
  geom_histogram() +
   labs(
    x = "Run Time\n(In Minutes)",
    y = "Frequency"
  )
p3 <- imdb_top_1000 %>%
  ggplot(aes(x = no_votes_scaled)) +
  geom_histogram() +
   labs(
    x = "IMDB Votes \n(In Millions)",
    y = "Frequency"
  )

p1 + p2 + p3 + plot_annotation('Distribution of Key Numerical Predictors')
```



```r
imdb_top_1000 %>%
  summarise(
    mean_gross = mean(gross_scaled),
    med_gross = median(gross_scaled),
    sd_gross = sd(gross_scaled),
    IQR_gross = IQR(gross_scaled),
    min_gross = min(gross_scaled),
    max_gross = max(gross_scaled),
    mean_runtime = mean(runtime),
    med_runtime = median(runtime),
    sd_runtime = sd(runtime),
    IQR_runtime = IQR(runtime),
    min_runtime = min(runtime),
    max_runtime = max(runtime),
    mean_votes = mean(no_votes_scaled),
    med_votes = median(no_votes_scaled),
    sd_votes = sd(no_votes_scaled),
    IQR_votes = IQR(no_votes_scaled),
    min_votes = min(no_votes_scaled),
    max_votes = max(no_votes_scaled)
  ) %>%
  pivot_longer(cols = everything(),
               names_to = "Statistic",
               values_to = "Value") %>%
  separate(Statistic, into = c("Measure", "Variable"), sep = "_", extra = "merge") %>%
  pivot_wider(names_from = Measure, values_from = Value)
```

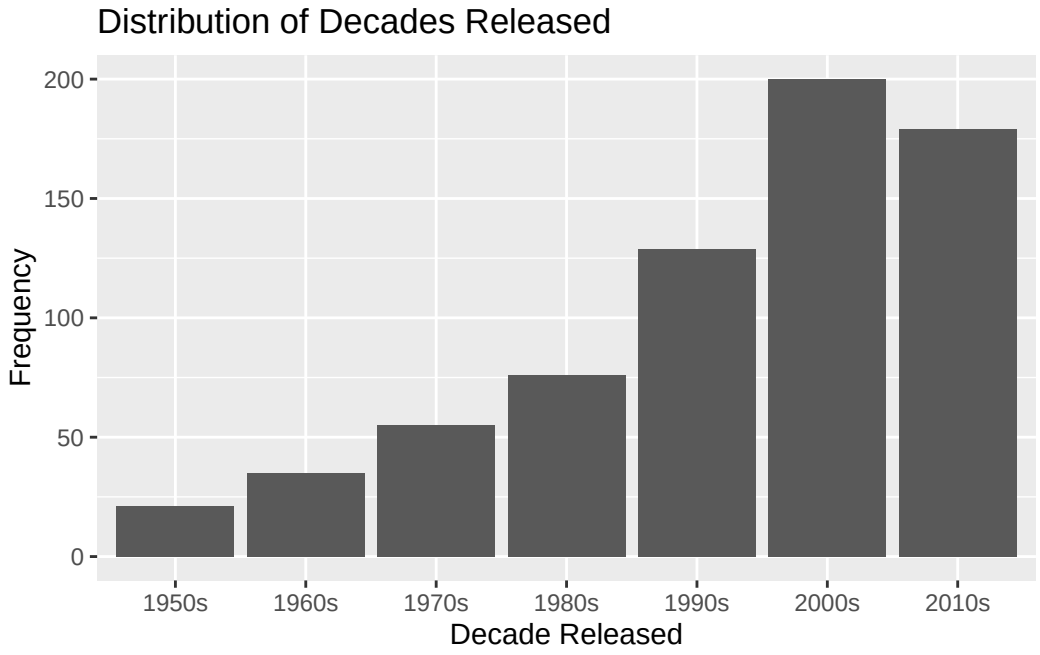
```
# A tibble: 3 x 7
  Variable    mean     med      sd    IQR      min    max
  <chr>      <dbl>   <dbl>   <dbl>  <dbl>    <dbl>  <dbl>
1 gross     80.1    35.9   116.    99.9    0.00130 937.
2 runtime  124.    120      25.6   32     72       228
3 votes      0.361   0.241   0.357  0.418  0.0252    2.34
```

Exploring our numerical predictors, it seems that the distribution of gross revenue in millions of dollars seems to have a right skew, with most values being below $500 million. It has a mean of $78.514 million, median of $34.850 million, standard deviation of about $115 million, IQR of $96.310 million, and a range of about $937 million.

Run time seems fairly normal with a slight right skew. There is a potential outlier around 238 minutes. It has a mean of about 124 minutes, median of about 120 minutes, standard deviation of about 26 minutes, IQR of about 32 minutes, and a range of about 166 minutes.

Finally, number of votes has a right skew. With a potential outlier at about 2.34 million votes, it has a mean of about 356,000 votes, median of about 267,000 votes, standard deviation of about 354,000 votes, IQR of about 412,000 votes, and a range of about 2.32 million votes.

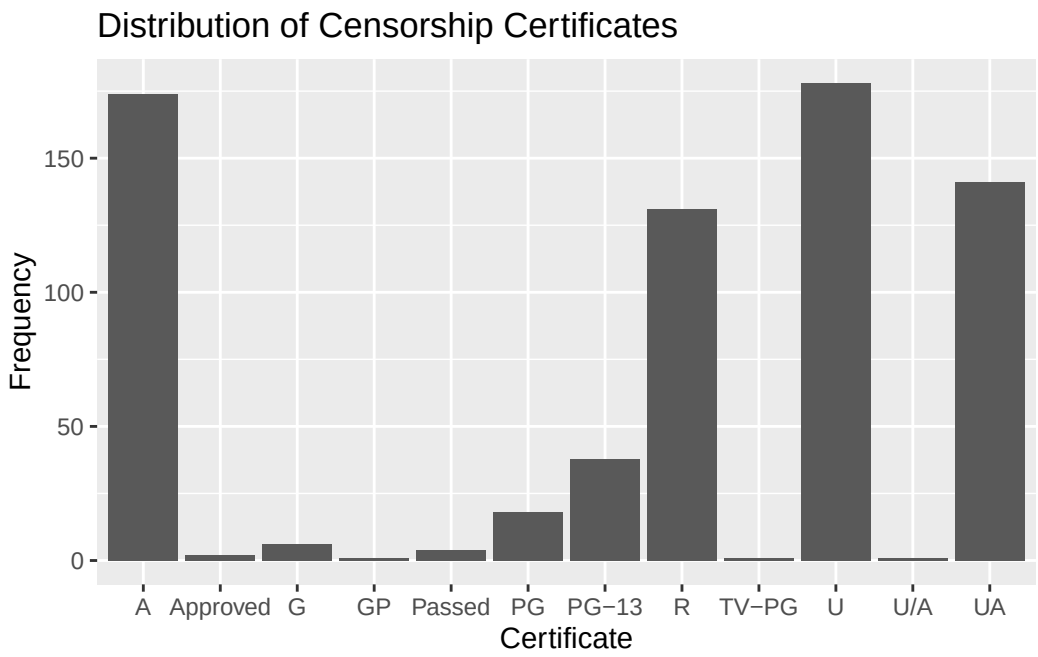
```
imdb_top_1000 %>%
  ggplot(aes(x = decade)) +
  geom_bar() + labs(
    title = "Distribution of Decades Released",
    x = "Decade Released",
    y = "Frequency"
  )
```



The distribution of `decade` seems to be skewed left. This is expected, as newer films are more likely to have been added to the internet in real time after release whereas older films are added retroactively.

```
imdb_top_1000 %>%
  ggplot(aes(x = Certificate)) +
  geom_bar() +
   labs(
     title = "Distribution of Censorship Certificates",
    x = "Certificate",
```

```
    y = "Frequency",
    fill = "Decade Released")
```

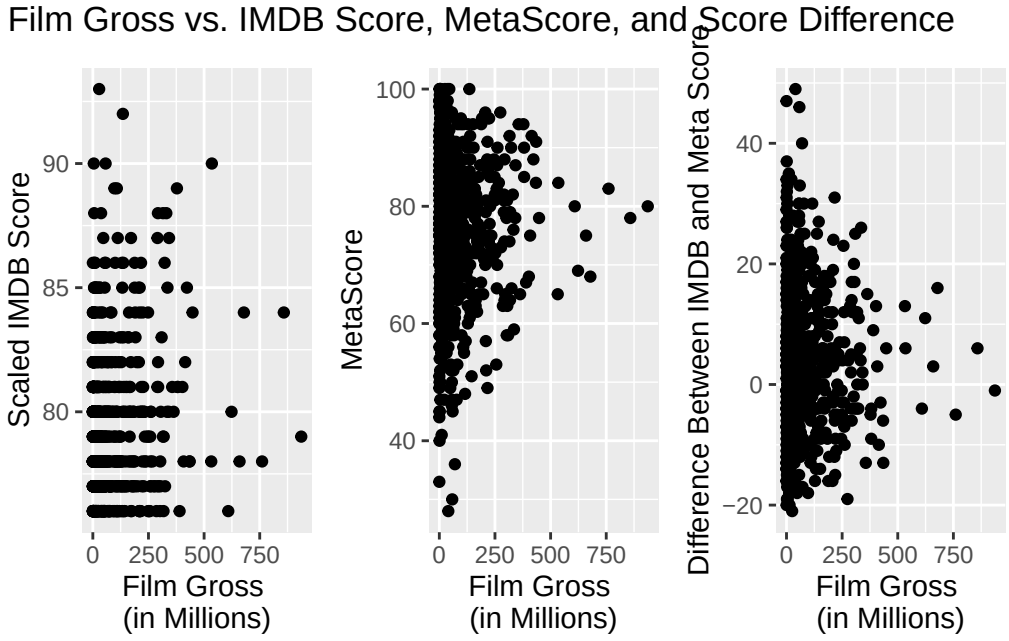


The distribution of `certificates` does not exhibit much of a normal shape, but notably the highest distribution is of "U" movies - those with unrestricted audiences.

Bivariate EDA

```
library(patchwork)
p1 <- imdb_top_1000 %>%
  ggplot(aes(x = gross_scaled, y = IMDB_scaled )) +
  geom_point() + labs(
    x = "Film Gross \n(in Millions)",
    y = "Scaled IMDB Score"
  )
p2 <- imdb_top_1000 %>%
  ggplot(aes(x = gross_scaled, y = Meta_score )) +
  geom_point() +
   labs(
    x = "Film Gross \n(in Millions)",
```

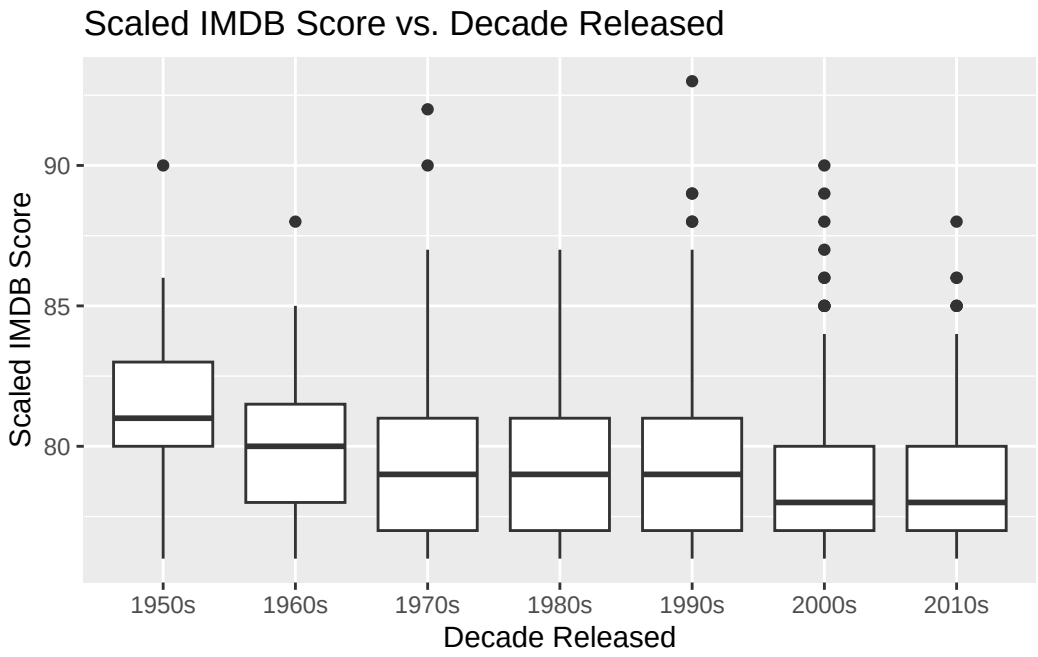
```
    y = "MetaScore"
  )
p3 <- imdb_top_1000 %>%
  ggplot(aes(x = gross_scaled, y = difference )) +
  geom_point() +
   labs(
    x = "Film Gross \n(in Millions)",
    y = "Difference Between IMDB and Meta Score"
  )

p1 + p2 + p3 +
  plot_annotation('Film Gross vs. IMDB Score, MetaScore, and Score Difference')
```



Upon initial Bivariate EDA, a clear linear relationship does not seem to appear between film's gross in millions and our three potential predictor variables. Perhaps later, to fit a model, we will need to find a variable transformation that gives us a promising model.

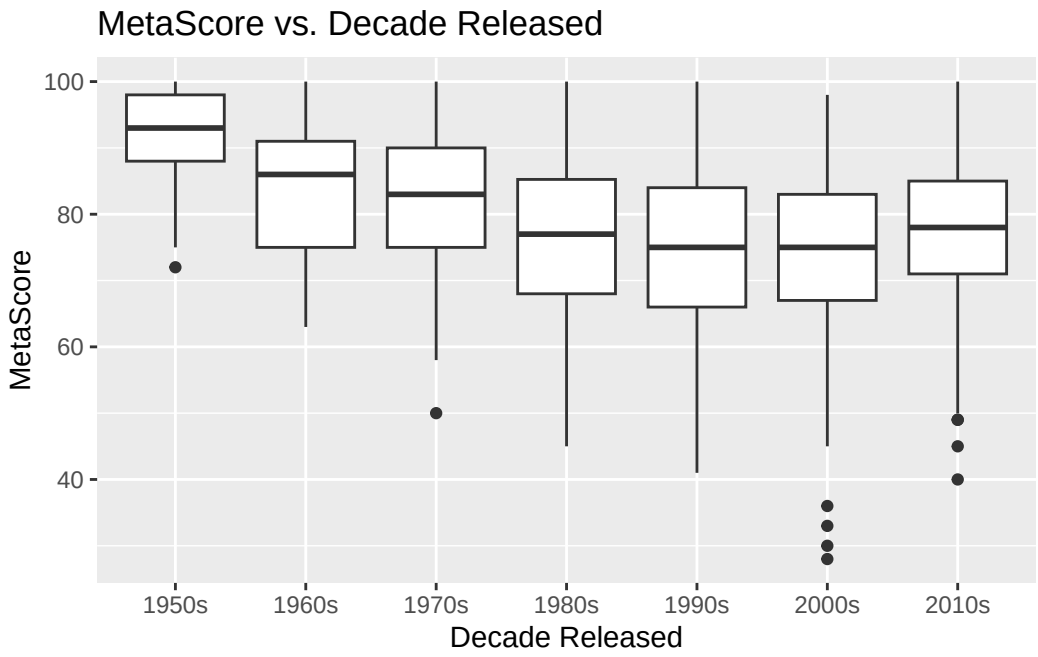
```
library(patchwork)

imdb_top_1000 %>%
  ggplot(aes(x = decade, y = IMDB_scaled )) +
  geom_boxplot() + labs(
    title = "Scaled IMDB Score vs. Decade Released",
```

```
    x = "Decade Released",
    y = "Scaled IMDB Score"
  )
```



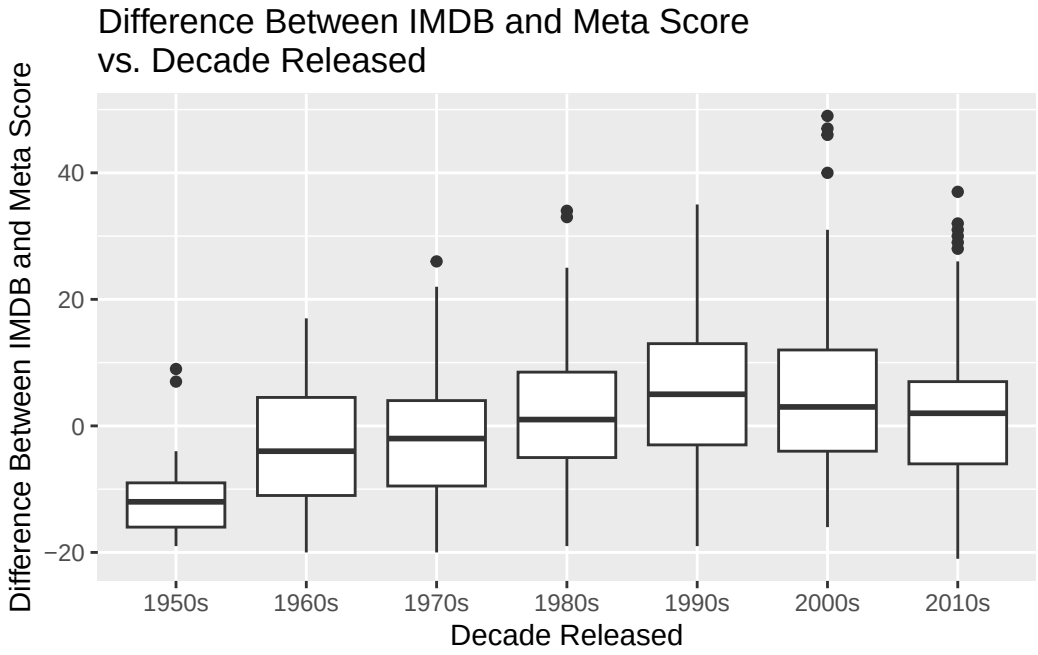
Judging from this initial bivariate EDA of decade released vs the scaled IMDB score, there seems to be a negative correlation between date and IMDB score; as movies are newer (coming out in more recent decades), the median scaled IMDB score tends to be lower.

```
imdb_top_1000 %>%
  ggplot(aes(x = decade, y = Meta_score )) +
  geom_boxplot() +
   labs(
    title = "MetaScore vs. Decade Released",
    x = "Decade Released",
    y = "MetaScore"
  )
```



Similarly to IMDB score, the critics' median MetaScores also seem to be lower as movies are newer. In other words, the overall aggregated critic scores for films tend to be lower for movies in more recent decades.

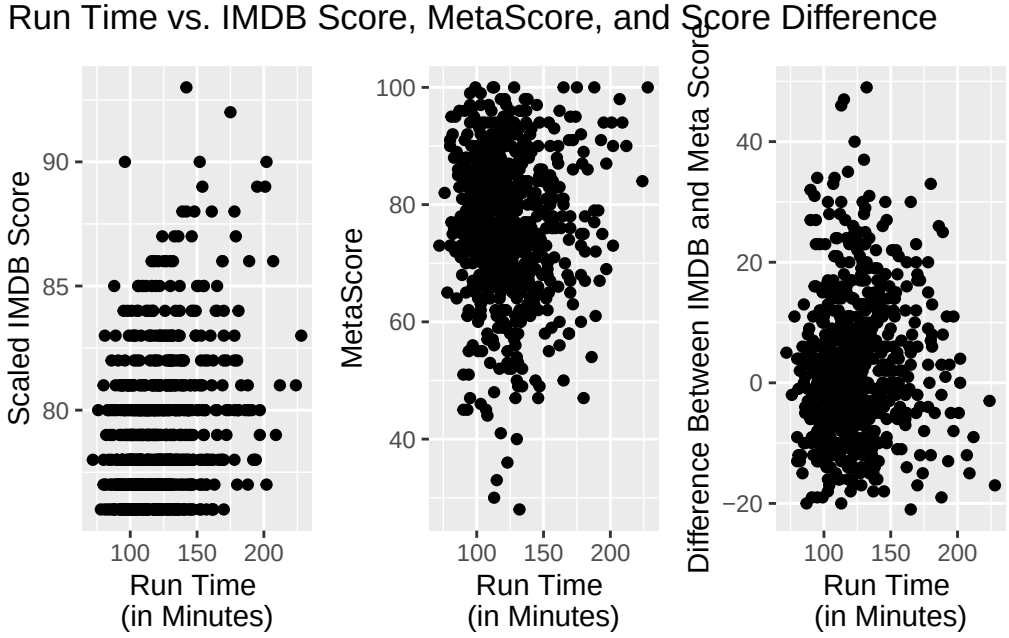
```
imdb_top_1000 %>%
  ggplot(aes(x = decade, y = difference )) +
  geom_boxplot() +
   labs(
    title = "Difference Between IMDB and Meta Score \nvs. Decade Released",
    x = "Decade Released",
    y = "Difference Between IMDB and Meta Score"
  )
```



Judging from this EDA, the median difference between `Meta_score` and `IMDB_scaled` also tends to be lower as movies are newer. In other words, MetaScores tend to be lower than IMDB scores in more recent decades.

```
library(patchwork)
p1 <- imdb_top_1000 %>%
  ggplot(aes(x = runtime, y = IMDB_scaled )) +
  geom_point() + labs(
    x = "Run Time \n(in Minutes)",
    y = "Scaled IMDB Score"
  )
p2 <- imdb_top_1000 %>%
  ggplot(aes(x = runtime, y = Meta_score )) +
  geom_point() +
   labs(
    x = "Run Time \n(in Minutes)",
    y = "MetaScore"
  )
p3 <- imdb_top_1000 %>%
  ggplot(aes(x = runtime, y = difference )) +
  geom_point() +
   labs(
    x = "Run Time \n(in Minutes)",
```

```
    y = "Difference Between IMDB and Meta Score"
  )

p1 + p2 + p3 +
  plot_annotation('Run Time vs. IMDB Score, MetaScore, and Score Difference')
```

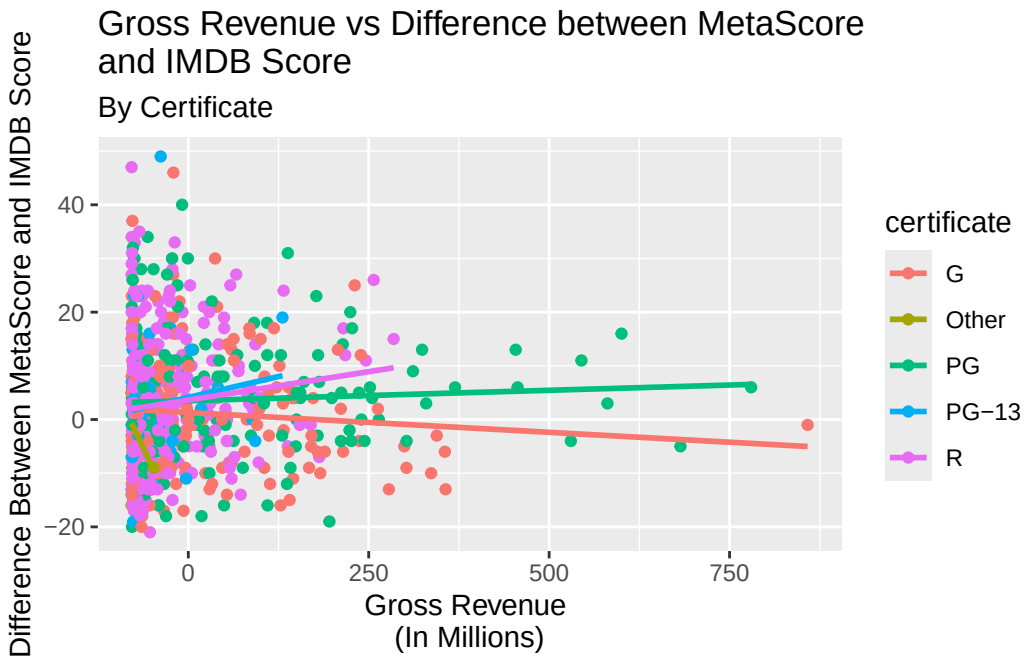


Similar to our gross predictor, a clear linear relationship does not seem to appear between film's run time in minutes and our three potential predictor variables. Perhaps later, to fit a model, we will need to find a variable transformation here as well that gives us a promising model.

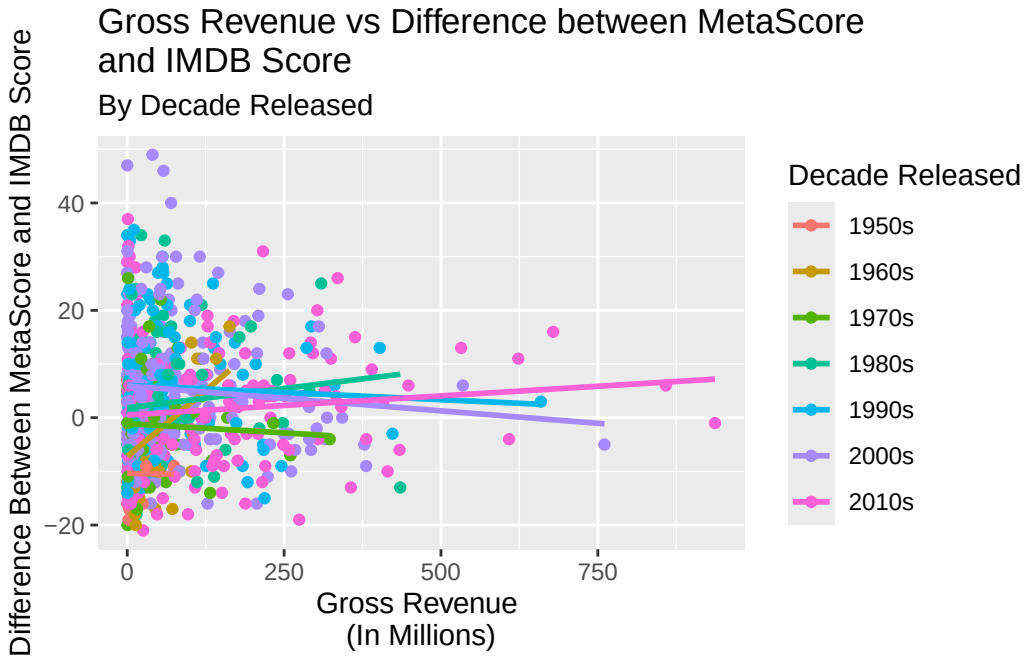
Interaction Effects

```
imdb_top_1000 %>%
  ggplot(aes(x = gross_cent, y = difference, color = certificate)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Gross Revenue vs Difference between MetaScore \nand IMDB Score",
       subtitle = "By Certificate",
```

```
       x = "Gross Revenue \n(In Millions)",
       y = "Difference Between MetaScore and IMDB Score")
```




```
imdb_top_1000 %>%
  ggplot(aes(x = gross_scaled, y = difference, color = decade)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Gross Revenue vs Difference between MetaScore \nand IMDB Score",
       subtitle = "By Decade Released",
       x = "Gross Revenue \n(In Millions)",
       y = "Difference Between MetaScore and IMDB Score",
       color = "Decade Released")
```



Here, we plotted gross revenue against the difference in scores (MetaScore minus scaled IMDB score) to examine how these variables interact against our categorical variables. The visualizations reveal clear interaction effects in two key relationships:

Gross Revenue and Certificate

Gross Revenue and Decade Released

> **!** Important
>
> Before you submit, make sure your code chunks are turned off with `echo: false` and there are no warnings or messages with `warning: false` and `message: false` in the YAML.

Reference: Moon, S., Bergey, P. K., & Iacobucci, D. (2010). Dynamic Effects among Movie Ratings, Movie Revenues, and Viewer Satisfaction. Journal of Marketing, 74(1), 108-121. https://doi.org/10.1509/jmkg.74.1.108