Statpadders

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

2025-03-17

```r
library(tidyverse)
library(tidymodels)
library(dplyr)
library(corrplot)

imdb_top_1000 <- read_csv("data/imdb_top_1000.csv") |>
  drop_na()
```

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.

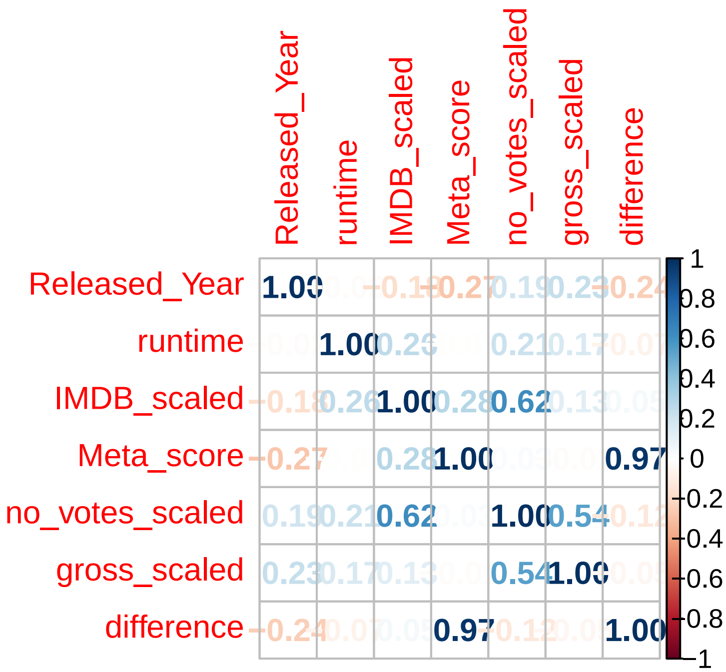
```r
imdb_top_1000 <- imdb_top_1000 |>
  mutate(no_votes_scaled = No_of_Votes / 10^6,
         gross_scaled = Gross / 10^6,
         IMDB_scaled = IMDB_Rating *10,
         Released_Year = if_else(Series_Title == "Apollo 13", "1995",
                                 Released_Year),
         difference = Meta_score - IMDB_scaled,
         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",))
```

Further, we turned our year 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 productive. Very few 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 for better interpretability, we changed this variable into a categorical variable `decade`, where all of the years released are divided into decades (i.e. 1950s, 1960s, etc.) This gives better interpretability for that variable.

Additionally, the variable `Runtime` gave the runtime of each movie in the following template: "(number) mins". As such, it was a categorical variable. We changed this to remove the "mins" label, and make the runtime into a numerical variable, which only shows the number of minutes in that movie's runtime.

```
#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")
```

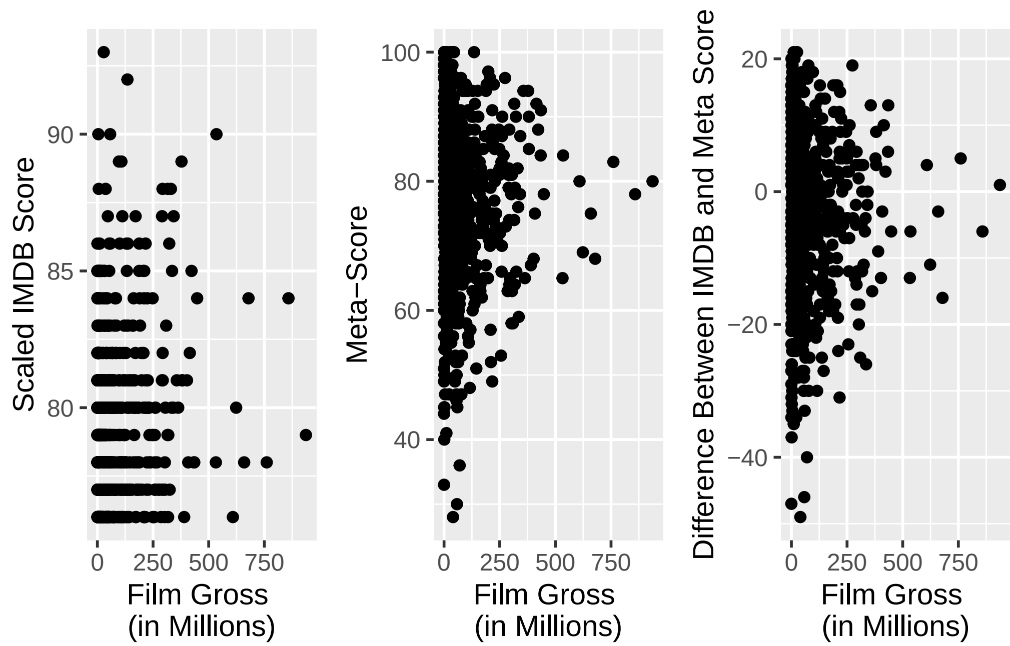


Here, we created a correlation matrix to see which of our numerical predictors may be highly correlated, thus indicating potential multicollinearity. There is high correlation (0.97 correlation coefficient) between the variable `difference` and the variables `IMDB_scaled` and `meta_score`, but that is to be expected - the `difference` variable is mutated from both of those variables, to show the difference between them.`IMDB_scaled` and `no_votes_scaled` are also highly correlated with a coefficient of 0.62, but that is also to be expected - of course the votes on IMDB are correlated with an IMDB score. We would not fit a model with both of those variables.

Univariate EDA

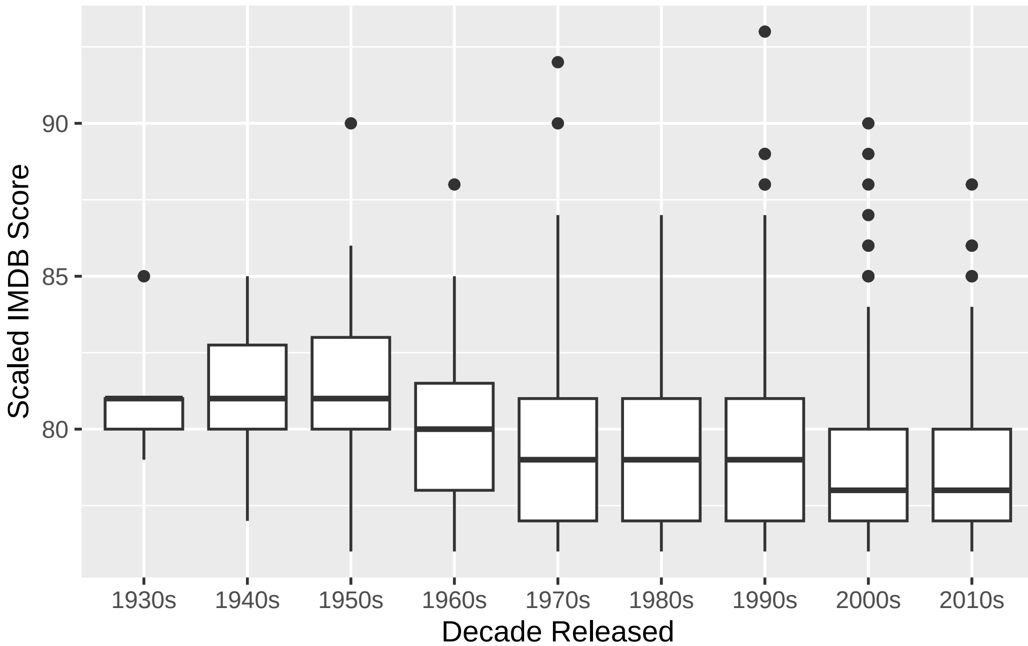
```
library(patchwork)
p1 <- imdb_top_1000 %>%
  ggplot(aes(x = gross_scaled, y = IMDB_scaled )) +
  geom_point() + labs(
    x = "Film Gross \n(in Millions)",
    y = "Scaled IMDB Score"
  )
p2 <- imdb_top_1000 %>%
  ggplot(aes(x = gross_scaled, y = Meta_score )) +
  geom_point() +
   labs(
    x =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

```
  )
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
```



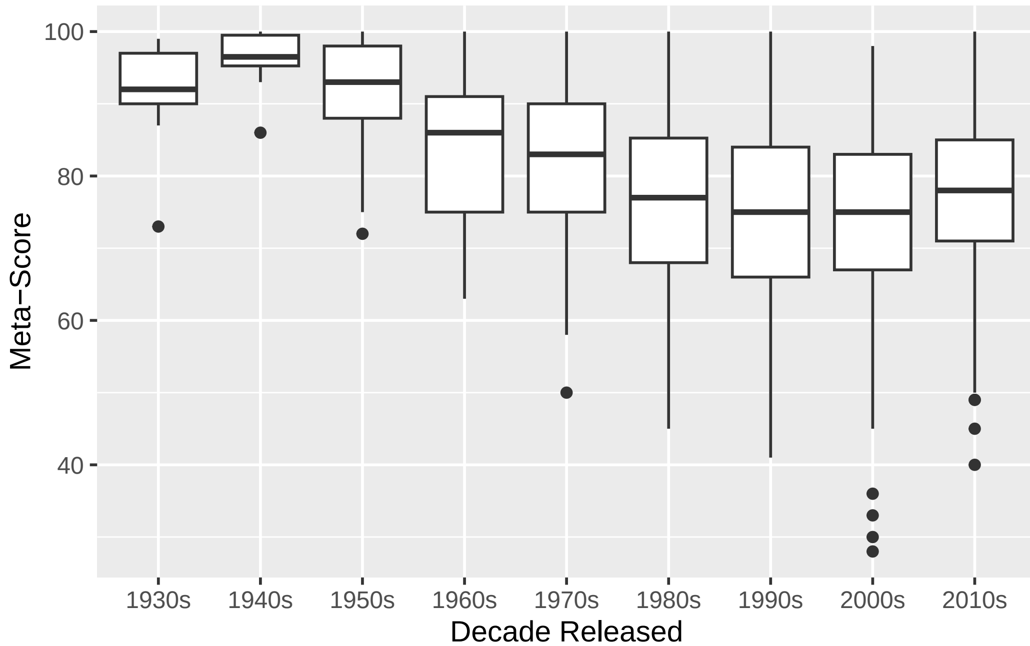
Add narrative here

```
library(patchwork)

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



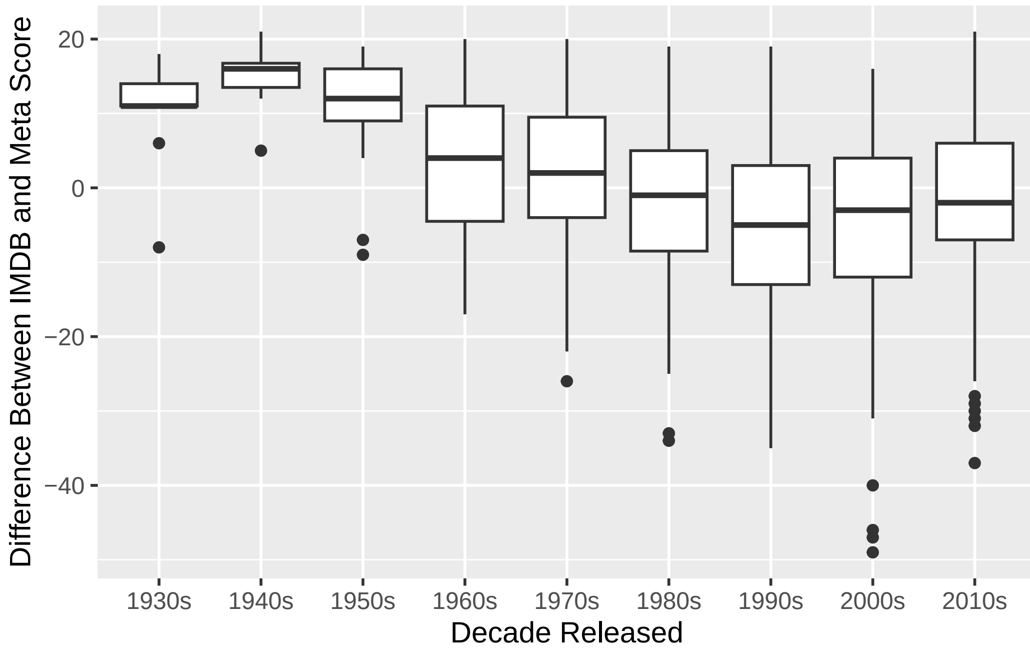
Judging from this intial univariate EDA of decade released vs the scaled IMDB score, there seems to be a negative correlation between date and IMDB score; as movies are newer (coming out in more recent decades), the median scaled IMDB score tends to be lower.

```
imdb_top_1000 %>%
  ggplot(aes(x = decade, y = Meta_score )) +
  geom_boxplot() +
   labs(
    x = "Decade Released",
    y = "Meta-Score"
  )
```



Similarly to IMDB score, the critics' median meta-scores 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.

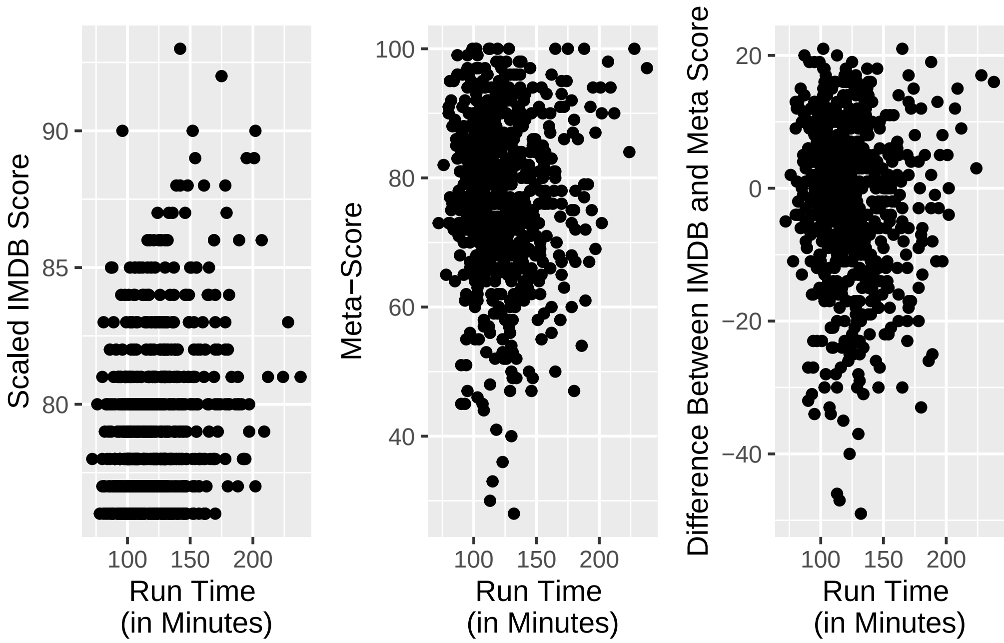
```
imdb_top_1000 %>%
  ggplot(aes(x = decade, y = difference )) +
  geom_boxplot() +
   labs(
    x = "Decade Released",
    y = "Difference Between IMDB and Meta Score"
  )
```



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, meta-scores tend to be lower than IMDB scores in more recent decades.

```r
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 = "Meta-Score"
  )
p3 <- imdb_top_1000 %>%
  ggplot(aes(x = runtime, y = difference )) +
  geom_point() +
   labs(
    x = "Run Time \n(in Minutes)",
```

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

p1 + p2 + p3
```



Add narrative here

> **❗ 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.