# IMDB Ratings

Kyle Flashman

2023-08-10

## Introduction

The 100 most grossing feature films will be analysed here looking at a number of variables in conjunction with their gross earnings. The data will be scraped from the IMDB website.

## Pre-Processing

Some ideas to explore are what variables are correlated with the movies' gross earnings (as well as with each other). Does their IMDB ratings have some indication of their earnings? What about their metascore rating? Or their genre? And so on.

```r
selector_title <- ".lister-item-header a"
titles1 <- wp_content %>%
  html_nodes(selector_title) %>%
  html_text()
titles2 <- wp_content2 %>%
  html_nodes(selector_title) %>%
  html_text()
titles <- c(titles1, titles2)

selector_year <- ".lister-item-year"
years1 <- wp_content %>%
  html_nodes(selector_year) %>%
  html_text()
years1 <- readr::parse_number(years1)
years2 <- wp_content2 %>%
  html_nodes(selector_year) %>%
  html_text()
years2 <- readr::parse_number(years2)
years <- c(years1, years2)

selector_film_rating <- ".certificate"
film_ratings1 <- wp_content %>%
  html_nodes(selector_film_rating) %>%
  html_text()
film_ratings2 <- wp_content2 %>%
  html_nodes(selector_film_rating) %>%
  html_text()
film_ratings <- c(film_ratings1, film_ratings2)
```

As shown in the code above, the selectors for the variables ‘title’, ‘year’, and ‘film rating’ are used to extract from each feature film of both the top 50 movies page and the 51-100 movies page.

```r
selector_genre <- ".genre"
genres1 <- wp_content %>%
  html_nodes(selector_genre) %>%
  html_text()
genres1 <- strsplit(trimws(genres1), ",")
genres1 <- data.frame(do.call(rbind, genres1))

genres2 <- wp_content2 %>%
  html_nodes(selector_genre) %>%
  html_text()
genres2 <- strsplit(trimws(genres2), ",")
genres2 <- data.frame(do.call(rbind, genres2))
genres <- rbind(genres1, genres2)

dummies <- dummy_cols(genres, select_columns = c("X1", "X2", "X3"))

act <- as.logical(dummies$X1_Action)
adv <- as.logical(dummies$X1_Adventure) | as.logical(dummies$`X2_ Adventure`) | as.logical(dummies$X3_Ac
ani <- as.logical(dummies$X1_Animation)
cri <- as.logical(dummies$X1_Crime) | as.logical(dummies$`X2_ Crime`)
dra <- as.logical(dummies$X1_Drama) | as.logical(dummies$`X2_ Drama`) | as.logical(dummies$`X3_ Drama`)
hor <- as.logical(dummies$X1_Horror)
bio <- as.logical(dummies$`X2_ Biography`)
com <- as.logical(dummies$`X2_ Comedy`) | as.logical(dummies$`X3_ Comedy`)
fam <- as.logical(dummies$`X2_ Family`) | as.logical(dummies$`X3_ Family`)
fan <- as.logical(dummies$`X2_ Fantasy`) | as.logical(dummies$`X3_ Fantasy`)
rom <- as.logical(dummies$`X2_ Romance`)
sci <- as.logical(dummies$`X2_ Sci-Fi`) | as.logical(dummies$`X3_ Sci-Fi`)
thr <- as.logical(dummies$`X3_ Thriller`)

genre <- cbind(act, adv, ani, cri, dra, hor, bio, com, fam, fan, rom, sci, thr)

selector_ir <- ".ratings-imdb-rating"
imdb_ratings1 <- wp_content %>%
  html_nodes(selector_ir) %>%
  html_text()
imdb_ratings1 <- readr::parse_number(imdb_ratings1)
imdb_ratings2 <- wp_content2 %>%
  html_nodes(selector_ir) %>%
  html_text()
imdb_ratings2 <- readr::parse_number(imdb_ratings2)
imdb_ratings <- c(imdb_ratings1, imdb_ratings2)

selector_ms <- ".ratings-metascore"
metascore_ratings1 <- wp_content %>%
  html_nodes(selector_ms) %>%
  html_text()
metascore_ratings1 <- readr::parse_number(metascore_ratings1)
metascore_ratings2 <- wp_content2 %>%
  html_nodes(selector_ms) %>%
  html_text()
```

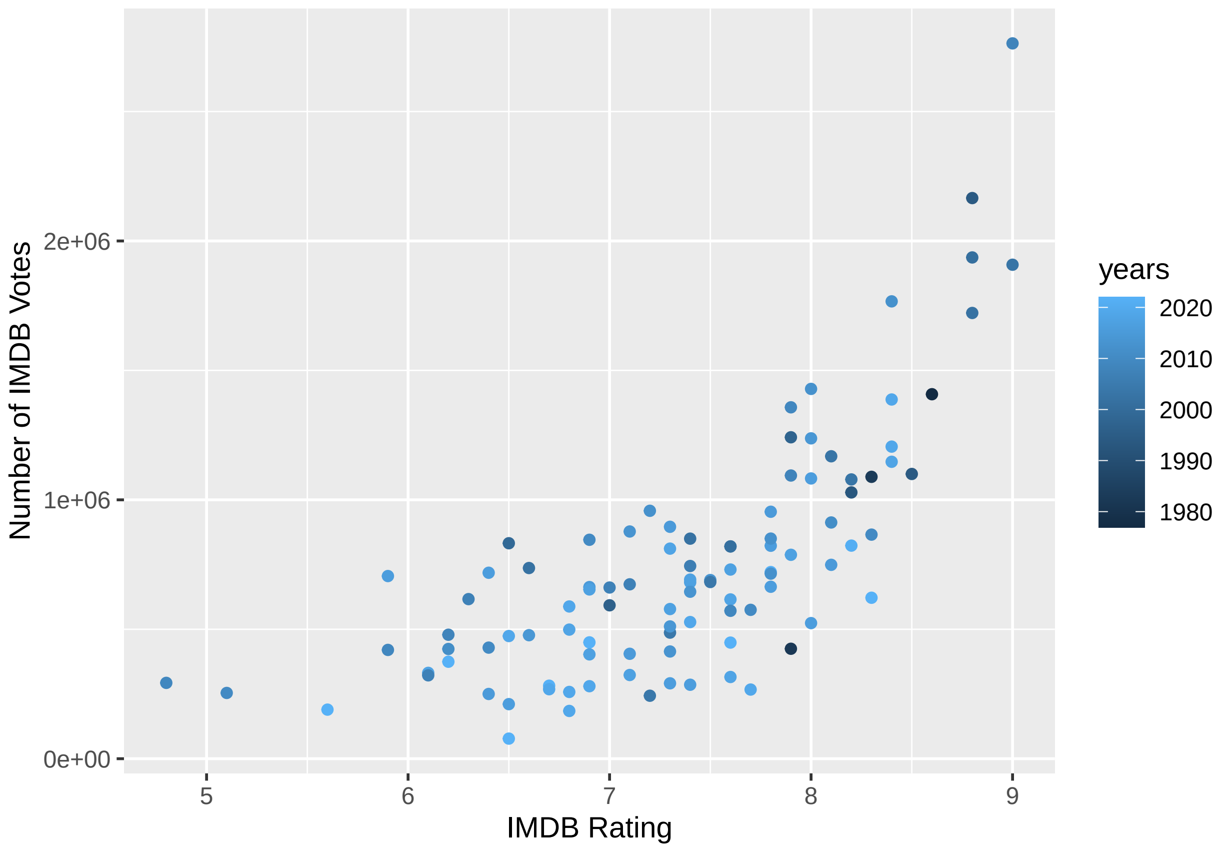
```
metascore_ratings2 <- readr::parse_number(metascore_ratings2)
metascore_ratings <- c(metascore_ratings1, metascore_ratings2)

selector_votes <- ".sort-num_votes-visible"
votes1 <- wp_content %>%
  html_nodes(selector_votes) %>%
  html_text()
votes1 <- readr::parse_number(votes1)
votes2 <- wp_content2 %>%
  html_nodes(selector_votes) %>%
  html_text()
votes2 <- readr::parse_number(votes2)
votes <- c(votes1, votes2)

selector_bo <- ".sort-num_votes-visible"
gross1 <- wp_content %>%
  html_elements(selector_bo) %>%
  html_text2()
gross1 <- sub(".*Gross: ", "", gross1)
gross1 <- readr::parse_number(gross1)
gross2 <- wp_content2 %>%
  html_elements(selector_bo) %>%
  html_text2()
gross2 <- sub(".*Gross: ", "", gross2)
gross2 <- readr::parse_number(gross2)
gross <- c(gross1, gross2)
```

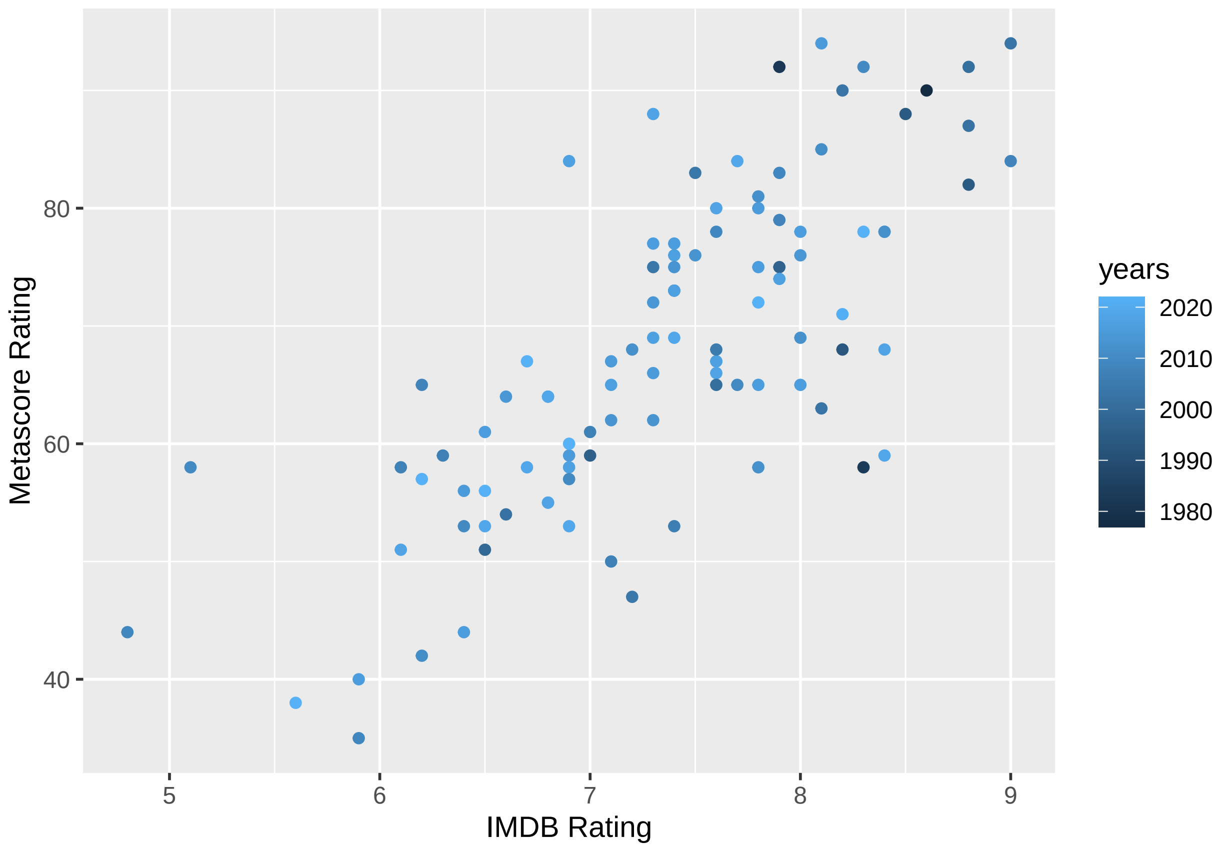
Then the variables 'Genre', 'IMDB rating', 'Metascore Rating', 'Number of IMDB Votes', and 'Gross Earnings' were also scraped with their respective selector. Since there are at most three genres used to classify each movie, Genre was split into three variables and then converted to dummy variables (one variable for each genre).

```
movies <- data.frame(titles, years, film_ratings, "Genre" = genre, imdb_ratings, metascore_ratings, gros
ggplot(movies,
       aes(x = imdb_ratings, y = votes, col = years)) +
  geom_point() +
  labs(x = "IMDB Rating", y = "Number of IMDB Votes")
```



There seems to be an exponential relationship between the number of IMDB votes and IMDB rating. Interestingly, the years seem to be spread out.

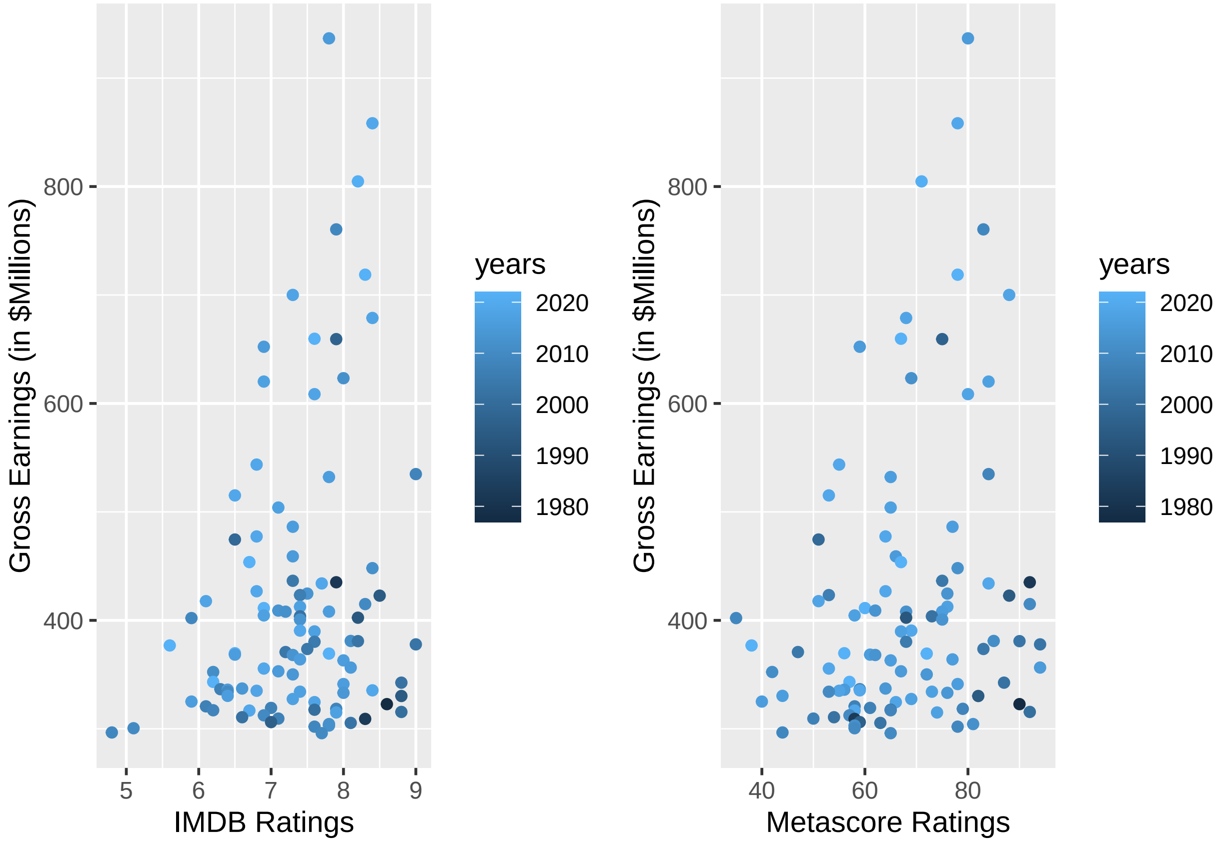
```
ggplot(movies,
       aes(x = imdb_ratings, y = metascore_ratings, col = years)) +
  geom_point() +
  labs(x = "IMDB Rating", y = "Metascore Rating")
```



Looks like there is a linear relationship between the Metascore and IMDB ratings. So we will likely have to take one of the two out of the linear regression model at the end.

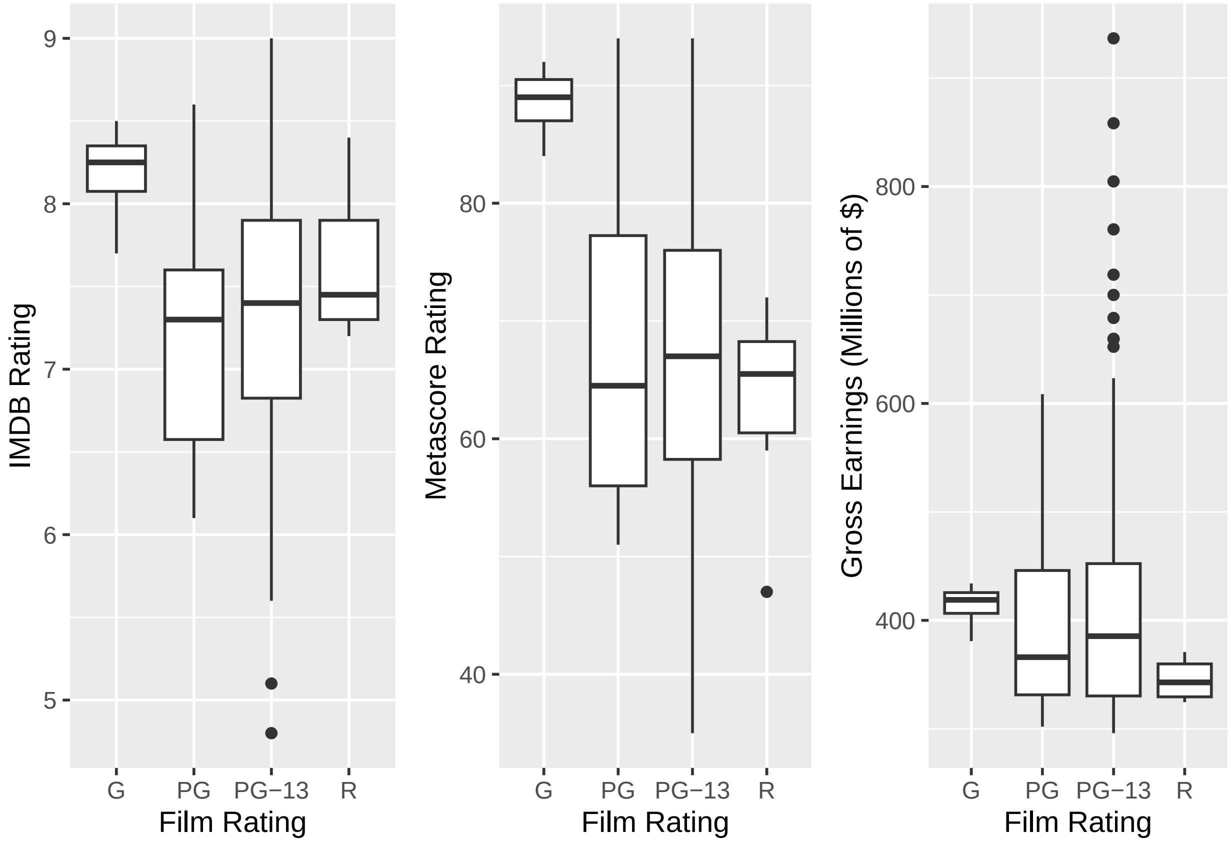
```
gp1 <- ggplot(movies,
       aes(x = imdb_ratings, y = gross, col = years)) +
  geom_point() +
  labs(x = "IMDB Ratings", y = "Gross Earnings (in $Millions)")

gp2 <- ggplot(movies,
              aes(x = metascore_ratings, y = gross, col = years)) +
  geom_point() +
  labs(x = "Metascore Ratings", y = "Gross Earnings (in $Millions)")
grid.arrange(gp1, gp2, nrow = 1)
```



These two plots are very similar, thus confirming our previous correlation declaration.

```
fr1 <- ggplot(movies, aes(x = film_ratings, y = imdb_ratings)) +
  geom_boxplot() +
  labs(x = "Film Rating", y = "IMDB Rating")
fr2 <- ggplot(movies, aes(x = film_ratings, y = metascore_ratings)) +
  geom_boxplot() +
  labs(x = "Film Rating", y = "Metascore Rating")
fr3 <- ggplot(movies, aes(x = film_ratings, y = gross)) +
  geom_boxplot() +
  labs(x = "Film Rating", y = "Gross Earnings (Millions of $)")
grid.arrange(fr1, fr2, fr3, nrow = 1)
```



Again, the IMDB and Metascore ratings plots with respect to the Film Rating are similar. And in each of these three boxplots, ‘PG-13’ has the biggest spread with 9 outliers on the Gross Earnings plot. However, there doesn’t seem to be any significance between the Film Rating and Gross Earnings.

```
movies %>% lm(gross ~ . - titles, .) %>% summary()
```

```
##
## Call:
## lm(formula = gross ~ . - titles, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -196.15  -69.37   -8.20   55.79  390.53
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -18415.884   3501.169  -5.260 1.18e-06 ***
## years                  9.153      1.736   5.274 1.11e-06 ***
## film_ratingsPG        35.144     68.262   0.515   0.6081
## film_ratingsPG-13    -73.799     91.786  -0.804   0.4238
## film_ratingsR        -80.332    121.220  -0.663   0.5094
## Genre.actTRUE         19.527     73.729   0.265   0.7918
## Genre.advTRUE        -30.960     48.629  -0.637   0.5262
## Genre.aniTRUE        -11.685    113.148  -0.103   0.9180
## Genre.criTRUE        -73.043     75.397  -0.969   0.3356
## Genre.draTRUE         47.734     65.251   0.732   0.4666
## Genre.horTRUE       -101.546    160.644  -0.632   0.5291
## Genre.bioTRUE       -126.817    139.098  -0.912   0.3647
## Genre.comTRUE        -73.691     67.042  -1.099   0.2750
## Genre.famTRUE       -108.886     91.867  -1.185   0.2394
## Genre.fanTRUE         84.766     69.396   1.221   0.2255
## Genre.romTRUE        142.840    119.256   1.198   0.2345
## Genre.sciTRUE         91.614     66.954   1.368   0.1750
## Genre.thrTRUE        -74.797     75.617  -0.989   0.3256
## imdb_ratings          37.847     24.236   1.562   0.1223
## metascore_ratings      2.707      1.523   1.777   0.0794 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 113 on 80 degrees of freedom
## Multiple R-squared:  0.4134, Adjusted R-squared:  0.2741
## F-statistic: 2.968 on 19 and 80 DF,  p-value: 0.0003663
```

```
movies %>% lm(gross ~ . - titles - imdb_ratings, .) %>% summary()
```

```
##
## Call:
## lm(formula = gross ~ . - titles - imdb_ratings, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -195.89  -62.13  -11.07   48.12  392.60
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -17165.613   3438.527  -4.992 3.36e-06 ***
## years                  8.607      1.715   5.018 3.03e-06 ***
## film_ratingsPG        32.374     68.842   0.470    0.639
## film_ratingsPG-13    -71.171     92.581  -0.769    0.444
## film_ratingsR        -52.725    120.984  -0.436    0.664
## Genre.actTRUE         29.688     74.090   0.401    0.690
## Genre.advTRUE        -46.671     47.998  -0.972    0.334
## Genre.aniTRUE         -5.928    114.088  -0.052    0.959
## Genre.criTRUE        -64.532     75.864  -0.851    0.397
## Genre.draTRUE         55.331     65.645   0.843    0.402
## Genre.horTRUE       -122.337    161.507  -0.757    0.451
## Genre.bioTRUE       -172.248    137.224  -1.255    0.213
## Genre.comTRUE        -62.789     67.267  -0.933    0.353
## Genre.famTRUE       -100.484     92.520  -1.086    0.281
## Genre.fanTRUE        100.887     69.231   1.457    0.149
## Genre.romTRUE        150.828    120.199   1.255    0.213
## Genre.sciTRUE         97.889     67.424   1.452    0.150
## Genre.thrTRUE        -70.245     76.229  -0.922    0.360
## metascore_ratings      4.454      1.042   4.273 5.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 114 on 81 degrees of freedom
## Multiple R-squared:  0.3956, Adjusted R-squared:  0.2612
## F-statistic: 2.945 on 18 and 81 DF,  p-value: 0.0004789
```

```
movies %>% lm(gross ~ . - titles - imdb_ratings - film_ratings - Genre.act - Genre.dra - Genre.cri - Ge
```

```
##
## Call:
## lm(formula = gross ~ . - titles - imdb_ratings - film_ratings -
##     Genre.act - Genre.dra - Genre.cri - Genre.adv - Genre.bio -
##     Genre.hor - Genre.thr - Genre.fam - Genre.ani - Genre.com,
##     data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -186.65  -77.09  -24.69   56.78  399.71
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.357e+04  2.951e+03  -4.600 1.32e-05 ***
## years              6.788e+00  1.455e+00   4.665 1.02e-05 ***
## Genre.fanTRUE      9.192e+01  3.165e+01   2.904  0.00459 **
## Genre.romTRUE      1.903e+02  8.748e+01   2.175  0.03214 *
## Genre.sciTRUE      9.370e+01  2.912e+01   3.218  0.00177 **
## metascore_ratings  4.240e+00  9.513e-01   4.457 2.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115.9 on 94 degrees of freedom
## Multiple R-squared:  0.2757, Adjusted R-squared:  0.2372
## F-statistic: 7.156 on 5 and 94 DF,  p-value: 1.032e-05
```

Since the IMDB and Metascore ratings were correlated as previously suggested I took out the variable with the biggest p-value which was the IMDB rating. And in the end, the variables that remain significant are the year the movie was released, metascore ratings, and the Fantasy, Romance, and Sci-Fi genres each with a positive linear relationship with their Gross Earnings.