Movie Analysis

Cavan Donohoe

1/15/2020

cavandonohoe@gmail.com

Movies

It’s Oscar Season so let’s take a little look at some historical movie data. I just learned how to Web Scrape so here is a little sample of easy web scraping to save yourself from mindless copying and pasting.

Libraries

Here are the libraries needed to be run for this run.

```
library(tidyverse)
library(readxl)
library(rvest)
library(scales)
```

Box Office

Web Scraping

First we have to grab the base url, then the for loop can get every single year after that. So the first year available is 1977 and the url would then become https://www.boxofficemojo.com/year/1977. We want to go all the way up to 2020. Unfortunately, this function needs to be updated every year to be looped for the latest year available. Please note that this is only US domestic box office.

```
url = "https://www.boxofficemojo.com/year/"

Box_Office.List = list()
for (year in c(1977:2020)) {
  url.year = paste(url, year, sep = "")
  xml.version.year = read_html(url.year)
  Box_Office.List[year-1976] = html_table(xml.version.year, header = TRUE)
}

Box_Office.Table = bind_rows(Box_Office.List, .id = "column_label")
```

Here’s what our table looks like:

```
as_tibble(Box_Office.Table)
```

```
## # A tibble: 19,450 x 12
##    column_label  Rank Release Genre Budget `Running Time` Gross Theaters
##    <chr>        <int> <chr>   <chr> <chr>  <chr>          <chr> <chr>
##   1 1                1 Star W~ -     -      -              $195~ 1,750
##   2 1                2 The De~ -     -      -              $47,~ 731
##   3 1                3 The Sp~ -     -      -              $45,~ 200
##   4 1                4 Oh, Go~ -     -      -              $41,~ 198
##   5 1                5 Exorci~ -     -      -              $30,~ 703
##   6 1                6 The Tu~ -     -      -              $25,~ 5
##   7 1                7 Lookin~ -     -      -              $22,~ 110
##   8 1                8 Saturd~ -     -      -              $18,~ 726
##   9 1                9 Close ~ -     -      -              $16,~ 650
##  10 2                1 Grease  -     -      -              $159~ 862
## # ... with 19,440 more rows, and 4 more variables: `Total Gross` <chr>,
## #   `Release Date` <chr>, Distributor <chr>, Estimated <chr>
```

```
as_tibble(Box_Office.Table) %>% select(Release, `Total Gross`, Distributor)
```

```
## # A tibble: 19,450 x 3
##    Release                            `Total Gross` Distributor
##    <chr>                              <chr>         <chr>
##   1 Star Wars: Episode IV - A New Hope $307,263,857  Twentieth Century Fox
##   2 The Deep                           $47,346,365   Columbia Pictures
##   3 The Spy Who Loved Me               $46,838,673   United Artists
##   4 Oh, God!                           $41,687,243   Warner Bros.
##   5 Exorcist II: The Heretic           $30,749,142   Warner Bros.
##   6 The Turning Point                  $25,933,445   Twentieth Century Fox
##   7 Looking for Mr. Goodbar            $22,512,655   Paramount Pictures
##   8 Saturday Night Fever               $94,213,184   Paramount Pictures
##   9 Close Encounters of the Third Kind $116,395,460  Columbia Pictures
##  10 Grease                             $159,978,870  Paramount Pictures
## # ... with 19,440 more rows
```

Data Cleaning

I want to know what year each movie came out, so let's cheat a little bit. The function I used earlier

```
Box_Office.Table = bind_rows(Box_Office.List, .id = "column_label")
```

made a column called "column_label" and that is essentially a year indicator, so let's adjust it.

```
Box_Office.Table$Year = as.numeric(Box_Office.Table$column_label) + 1976
as_tibble(Box_Office.Table) %>% select(Year, Release, `Total Gross`,
Distributor)
```

```
## # A tibble: 19,450 x 4
##     Year Release                            `Total Gross` Distributor
##    <dbl> <chr>                              <chr>         <chr>
##   1  1977 Star Wars: Episode IV - A New Hope $307,263,857  Twentieth
```

```
Century Fox
##  2  1977 The Deep                          $47,346,365   Columbia
Pictures
##  3  1977 The Spy Who Loved Me              $46,838,673   United Artists
##  4  1977 Oh, God!                          $41,687,243   Warner Bros.
##  5  1977 Exorcist II: The Heretic          $30,749,142   Warner Bros.
##  6  1977 The Turning Point                 $25,933,445   Twentieth
Century Fox
##  7  1977 Looking for Mr. Goodbar           $22,512,655   Paramount
Pictures
##  8  1977 Saturday Night Fever              $94,213,184   Paramount
Pictures
##  9  1977 Close Encounters of the Third Kind $116,395,460  Columbia
Pictures
## 10  1978 Grease                            $159,978,870  Paramount
Pictures
## # ... with 19,440 more rows
```

Sometimes if a movie is released in December, it will still be in theaters in January. That will make it have two years in their Year column. So let's just look at the earliest year and assume that is the actual release year.

```
Box_Office.Table_v2 = Box_Office.Table %>% group_by(Release, `Total Gross`,
Distributor) %>% summarise(`Release Year` = min(Year, na.rm = TRUE)) %>%
  rename(Title = Release) %>% mutate(`Title Type` = "movie")
Box_Office.Table_v2

## # A tibble: 16,255 x 5
## # Groups:   Title, Total Gross [16,255]
##    Title              `Total Gross` Distributor      `Release Year` `Title
Type`
##    <chr>              <chr>         <chr>                     <dbl> <chr>
##  1 '71                $1,270,847    Roadside Attrac~           2015 movie
##  2 '85: The Greatest~ $124,573      Fathom Events              2018 movie
##  3 'night, Mother     $441,863      Universal Pictu~           1986 movie
##  4 'R Xmas            $850          -                          2002 movie
##  5 'Round Midnight    $3,272,593    Warner Bros.               1986 movie
##  6 'Til There Was You $3,525,125    Paramount Pictu~           1997 movie
##  7 'Tis Autumn: The ~ $1,476        Outsider Films             2007 movie
##  8 !Women Art Revolu~ $52,681       Zeitgeist Films            2011 movie
##  9 $9.99              $52,384       Regent Releasing           2008 movie
## 10 (Untitled)         $230,600      The Samuel Gold~           2009 movie
## # ... with 16,245 more rows
```

Rotten Tomatoes

Web Scraping

Let's take a look at the Rotten Tomatoes' "TomatoMeter":

```
url_rt = "https://www.rottentomatoes.com/top/bestofrt/?year="

Rotten_Tomatoes.List = list()
for (year_rt in c(1950:2020)) {
  url.year_rt = paste(url_rt, year_rt, sep = "")
  xml.version.year_rt = read_html(url.year_rt)
  Rotten_Tomatoes.List[[year_rt-1949]] = html_table(xml.version.year_rt,
header = TRUE)[[3]]
}

Rotten_Tomatoes.Table = bind_rows(Rotten_Tomatoes.List, .id = "column_label")
```

There is a slight difference in the “html_table” function this time though. Rotten Tomatoes has a few tables on every single page they have, so I want the third table in each sheet for each year.

Data Cleaning

Rotten Tomatoes has years attached to their title, so let’s extract that and change the title to not have that year anymore. Also, let’s convert that percent to an actual number for the TomatoMeter.

```
Rotten_Tomatoes.Table_v2 = as_tibble(Rotten_Tomatoes.Table) %>% mutate(Year =
as.numeric(str_sub(Title, end=-2,start=-5)),
                                                            TomatoMeter =
as.numeric(sub("%","", RatingTomatometer))) %>%
  mutate(Title = substr(Title, start = 1, stop=nchar(Title)-7))

Rotten_Tomatoes.Table_v2 %>% select(Year, Title, RatingTomatometer,
TomatoMeter, `No. of Reviews`)

## # A tibble: 3,071 x 5
##      Year Title                    RatingTomatomet~ TomatoMeter `No. of
Reviews`
##     <dbl> <chr>                    <chr>                  <dbl>
<int>
##   1  1950 All About Eve            100%                     100
69
##   2  1950 Sunset Boulevard         98%                       98
63
##   3  1950 In a Lonely Place        98%                       98
40
##   4  1951 A Streetcar Named Desire  98%                       98
57
##   5  1951 Rashômon                 98%                       98
55
##   6  1951 Strangers on a Train     98%                       98
48
##   7  1951 An American in Paris     95%                       95
62
```

```
##   8  1951 The African Queen         98%                       98
44
##   9  1951 The Day the Earth Stood ~ 95%                       95
55
##  10  1952 Singin' in the Rain       100%                     100
56
## # ... with 3,061 more rows
```

Joining Two Tables

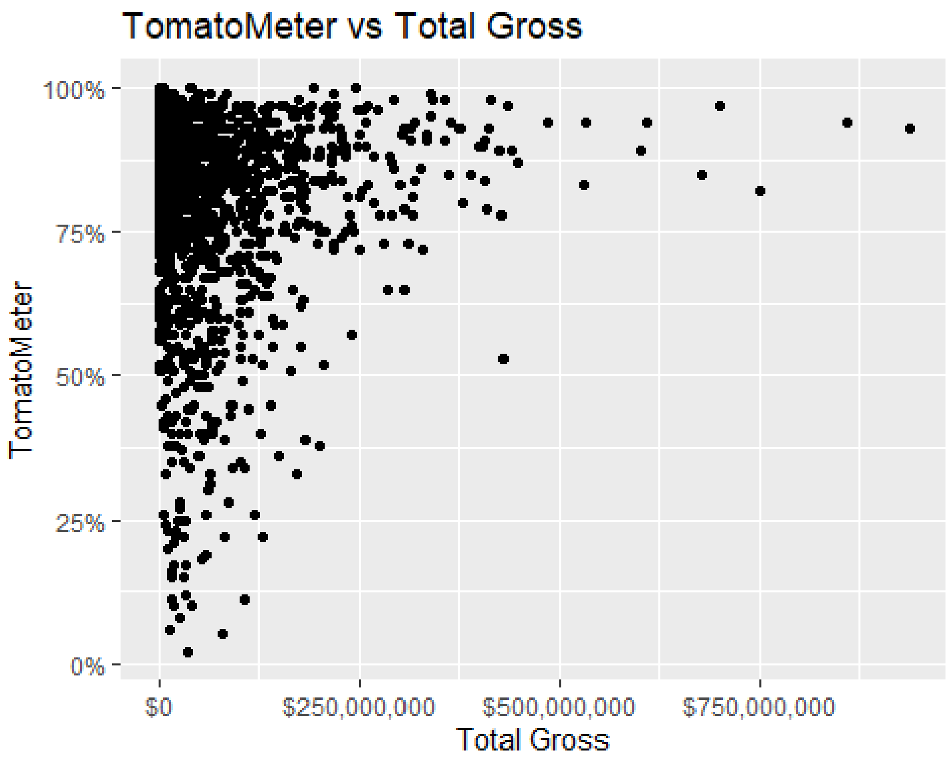
I am going to join both the Box office and Rotten Tomato tables. Rememeber how we made the Rotten Tomatoes percentage into an actual number? I'm going to do the same thing with the Box Office Gross by getting rid of the dollar sign ($) and commas (,).

```
Rotten_Tomatoes.Box_Office = Rotten_Tomatoes.Table_v2 %>%
inner_join(Box_Office.Table_v2 %>% ungroup() %>% mutate(Year = `Release
Year`)) %>%
  select(-column_label, - Rank, -`Release Year`) %>% mutate(`Total Gross
Number` = as.numeric(gsub('[$,]', "", `Total Gross`)))

## Joining, by = c("Title", "Year")

Rotten_Tomatoes.Box_Office

## # A tibble: 2,102 x 9
##    RatingTomatomet~ Title `No. of Reviews`   Year TomatoMeter `Total Gross`
##    <chr>            <chr>            <int> <dbl>       <dbl> <chr>
##  1 93%              Star~              124  1977          93 $307,263,857
##  2 95%              Clos~               60  1977          95 $116,395,460
##  3 83%              Satu~               48  1977          83 $94,213,184
##  4 79%              The ~               52  1977          79 $46,838,673
##  5 94%              Supe~               67  1978          94 $134,218,018
##  6 93%              Inva~               57  1978          93 $24,946,533
##  7 90%              Nati~               49  1978          90 $120,091,123
##  8 75%              Grea~               71  1978          75 $159,978,870
##  9 52%              The ~               42  1978          52 $30,471,420
## 10 97%              Alien              118  1979          97 $78,944,891
## # ... with 2,092 more rows, and 3 more variables: Distributor <chr>,
`Title
## #   Type` <chr>, `Total Gross Number` <dbl>
```

Data Visualization

Let's take a look at TomatoMeter vs Total Gross

```
Rotten_Tomatoes.Box_Office %>% ggplot(aes(x=`Total Gross Number`,
y=TomatoMeter)) + geom_point() + ggtitle("TomatoMeter vs Total Gross") +
xlab("Total Gross") + scale_y_continuous(labels = function(TomatoMeter)
paste0(TomatoMeter,"%")) + scale_x_continuous(labels = dollar)
```



Grouping by Distributor

I want to take a look at each distributor to see how much of a powerhouse Mickey is.

```
RT.BO_Distributor = Rotten_Tomatoes.Box_Office %>% group_by(Distributor) %>%
  summarise(`Total Gross` = sum(`Total Gross Number`, na.rm = TRUE), `Average
TomatoMeter` = mean(TomatoMeter, na.rm = TRUE),
            `Average Gross` = mean(`Total Gross Number`, na.rm = TRUE),
`Median TomatoMeter` = median(TomatoMeter, na.rm = TRUE)) %>%
  arrange(desc(`Total Gross`))
RT.BO_Distributor

## # A tibble: 159 x 5
##    Distributor   `Total Gross` `Average Tomato~ `Average Gross` `Median
TomatoM~
##    <chr>                 <dbl>            <dbl>           <dbl>
<dbl>
##  1 Walt Disney ~   22123395760             77.9      152575143.
84
##  2 Warner Bros.    15063721233             76.7       98455694.
81
##  3 Paramount Pi~   12257534152             74.4       91474135.
80
##  4 Twentieth Ce~   11219592843             79.2      106853265.
84
##  5 Universal Pi~    9008066788             76.6       69829975.
81
```

```
##  6 Sony Picture~    7033075504             72.6       75624468.
77
##  7 New Line Cin~    2969173695             74.7       78136150.
80
##  8 DreamWorks D~    2369220756             81.5      107691853.
83
##  9 Miramax          2282561515             81.5       28893184.
85
## 10 Lionsgate        2241250117             85.9       46692711.
86
## # ... with 149 more rows
```

It’s interesting if we arrange by median TomatoMeter.

```
RT.BO_Distributor %>% arrange(desc(`Median TomatoMeter`))
```

```
## # A tibble: 159 x 5
##    Distributor   `Total Gross` `Average Tomato~ `Average Gross` `Median
TomatoM~
##    <chr>                 <dbl>            <dbl>           <dbl>
<dbl>
##  1 CJ Entertain~        541719            100           541719
100
##  2 Arthouse Fil~        187716            100           187716
100
##  3 Utopia                57188            100            57188
100
##  4 1091 Media            25363            100            25363
100
##  5 Janus Films         3940579             96.8         788116.
99
##  6 Icarus Films         181750             99            90875
99
##  7 Big World Pi~        362328             96           120776
98
##  8 BritBox              330500             98           330500
98
##  9 MUBI                 117460             98           117460
98
## 10 United Artis~      22680962             97         22680962
97
## # ... with 149 more rows
```

Data Visualization and Analysis

Even though Disney has made the most money overall (perhaps because of the mass amount of movies they have produced), they don’t have the highest TomatoMeter rating. Let’s see the Average and Median Gross and maybe that will have a stronger relation with the Median TomatoMeter.

```
RT.BO_Distributor = Rotten_Tomatoes.Box_Office %>% group_by(Distributor) %>%
  summarise(`Total Gross` = sum(`Total Gross Number`, na.rm = TRUE), `Average
TomatoMeter` = mean(TomatoMeter, na.rm = TRUE),
            `Average Gross` = mean(`Total Gross Number`, na.rm = TRUE),
`Median TomatoMeter` = median(TomatoMeter, na.rm = TRUE),
            `Median Gross` = median(`Total Gross Number`, na.rm = TRUE))
RT.BO_Distributor
```

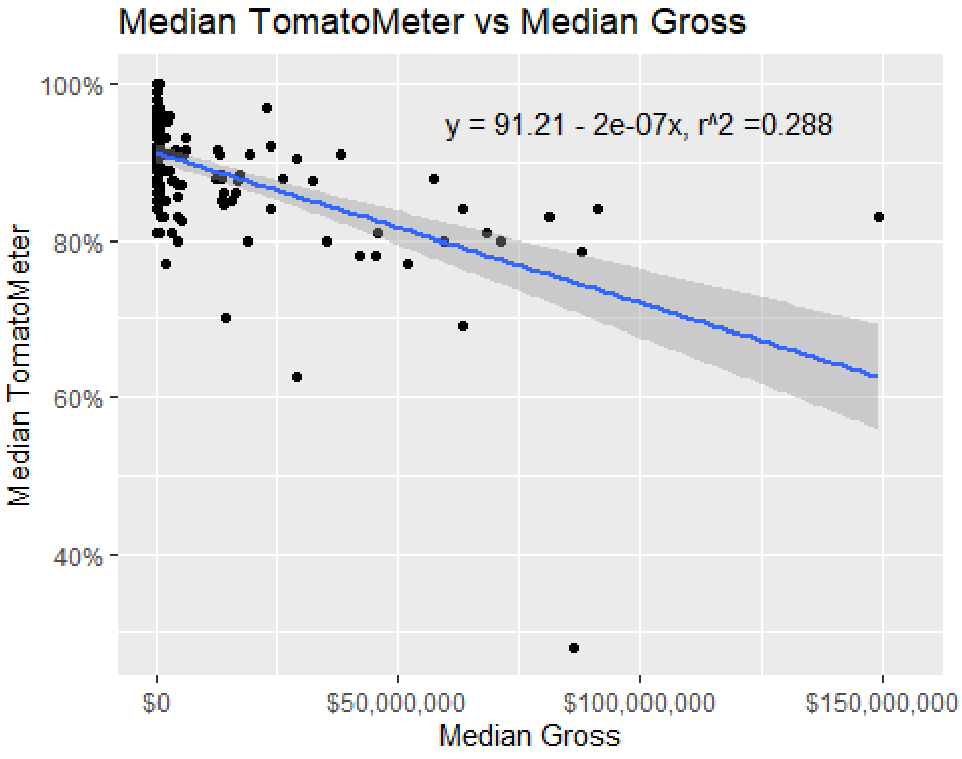
```
## # A tibble: 159 x 6
##    Distributor `Total Gross` `Average Tomato~ `Average Gross` `Median
TomatoM~
##    <chr>               <dbl>            <dbl>           <dbl>
<dbl>
##  1 -               233718873             88.9        7082390.
91
##  2 1091 Media          25363            100             25363
100
##  3 4th Row Fi~        117470             95            117470
95
##  4 A24             388256578             92.2       12133018.
91
##  5 ABKCO Films        293680             94            293680
94
##  6 Abramorama        6445793             93.5          805724.
95
##  7 Access Ent~        634566             91            634566
91
##  8 Adopt Films       1601469             93            400367.
93
##  9 Alluvial F~         77556             92             77556
92
## 10 Amazon Stu~      13017948             90.8         2603590.
89
## # ... with 149 more rows, and 1 more variable: `Median Gross` <dbl>
```

```
# let's see the simple linear regression of this sample
lm_1 = summary(lm(data=RT.BO_Distributor, `Median TomatoMeter` ~ `Median
Gross`))
lm_1
```

```
##
## Call:
## lm(formula = `Median TomatoMeter` ~ `Median Gross`, data =
RT.BO_Distributor)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.641  -3.169   0.801   4.103  20.406
##
## Coefficients:
```

```
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.121e+01  5.951e-01 153.279  < 2e-16 ***
## `Median Gross` -1.918e-07  2.407e-08  -7.967  3.1e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.789 on 157 degrees of freedom
## Multiple R-squared:  0.2879, Adjusted R-squared:  0.2834
## F-statistic: 63.48 on 1 and 157 DF,  p-value: 3.098e-13
```

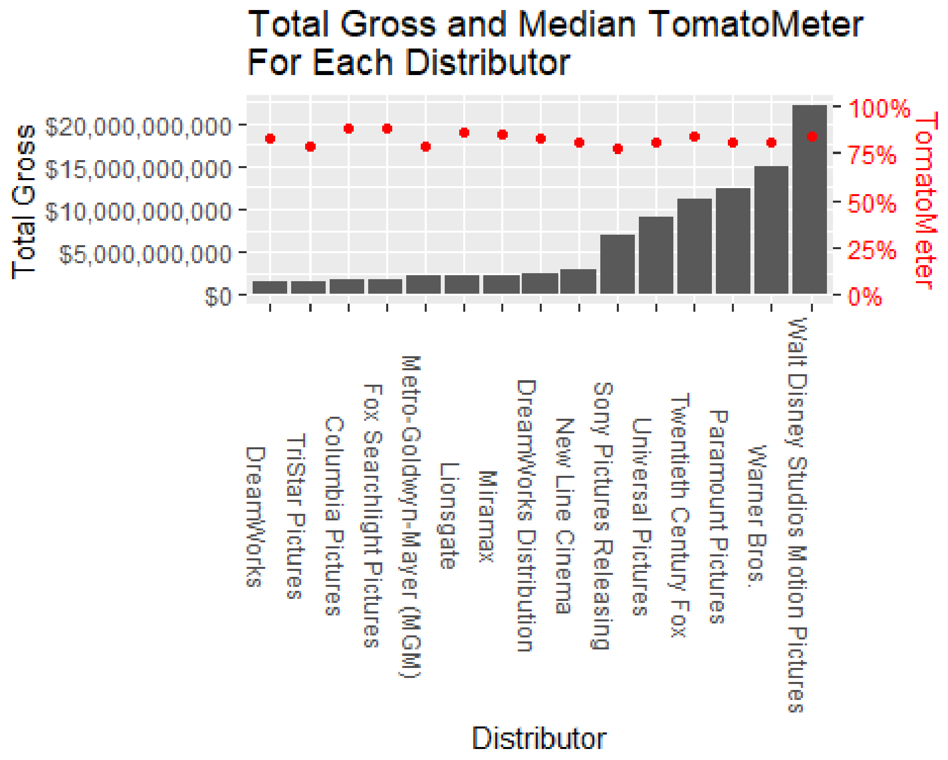
```
RT.BO_Distributor %>% ggplot(aes(x=`Median Gross`, y=`Median TomatoMeter`)) +
geom_point() + geom_smooth(method = "lm") +
  annotate("text",x=10^8, y =95, hjust=.5,vjust=.5,
           label=paste("y =
",round(lm_1$coefficients[[1]],2),ifelse(lm_1$coefficients[[2]] < 0, " - ","
+ "),

abs(round(lm_1$coefficients[[2]],7)),"x, ",
                                                                 "r^2
=",round(lm_1$r.squared,3),sep="")) + ggtitle("Median TomatoMeter vs Median
Gross") +
  scale_y_continuous(labels = function(TomatoMeter) paste0(TomatoMeter,"%"))
+ scale_x_continuous(labels = dollar, limits = c(0, 155000000))
```



So for the most part we cannot conclude the Median TomatoMeter has a linear relationship with Median Gross from a Distributor. So any of the top grossing distributors can either have highly rated movies or awful movies.

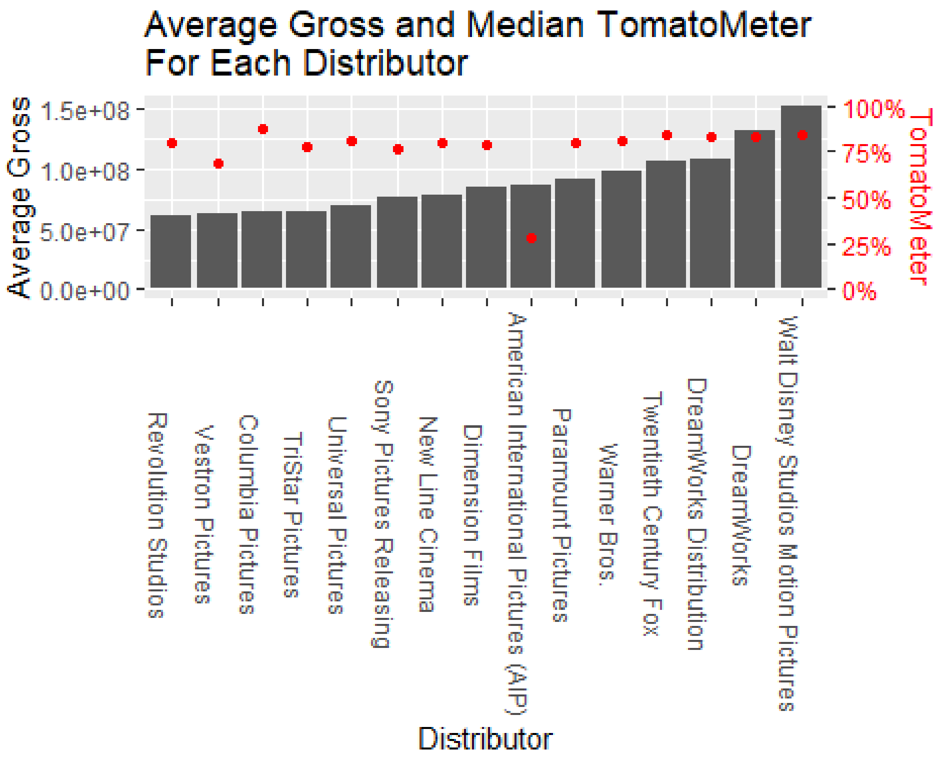
Let’s take a look at the top 15 Total Gross Distributors and see what their TomatoMeters look like:

```
# Total Gross and TomatoMeter for each Distributor
RT.BO_Distributor %>% arrange(desc(`Total Gross`)) %>% slice(1:15) %>%
ggplot() +
  geom_bar(aes(x=reorder(Distributor, `Total Gross`), y=`Total Gross`), stat
= "identity") +
  geom_point(aes(x=Distributor, y=`Median
TomatoMeter`*max(RT.BO_Distributor$`Total Gross`,na.rm = TRUE)/100),
color="red") +
  scale_y_continuous(sec.axis = sec_axis((~./max(RT.BO_Distributor$`Total
Gross`)),name="TomatoMeter",
                     labels= function(b) {
paste(round(b*100,0),"%",sep="")}), labels=dollar) +
  theme(axis.text.x = element_text(angle=-90),axis.title.y.right =
element_text(color="red"), axis.text.y.right = element_text(color="red")) +
  ggtitle("Total Gross and Median TomatoMeter\nFor Each Distributor") +
xlab("Distributor")
```



The TomatoMeters for each distributor again does not seem to have any relation to the Total Gross of these distributors.

```
# Average Gross and TomatoMeter for each Distributor
RT.BO_Distributor %>% arrange(desc(`Average Gross`)) %>% slice(1:15) %>%
ggplot() +
  geom_bar(aes(x=reorder(Distributor, `Average Gross`), y=`Average Gross`),
stat = "identity") +
  geom_point(aes(x=Distributor, y=`Median
TomatoMeter`*max(RT.BO_Distributor$`Average Gross`,na.rm = TRUE)/100),
color="red") +
  scale_y_continuous(sec.axis = sec_axis((~./max(RT.BO_Distributor$`Average
Gross`)),name="TomatoMeter",
                                         labels= function(b) {
paste(round(b*100,0),"%",sep="")})) +
  theme(axis.text.x = element_text(angle=-90),axis.title.y.right =
element_text(color="red"), axis.text.y.right = element_text(color="red")) +
  ggtitle("Average Gross and Median TomatoMeter\nFor Each Distributor") +
xlab("Distributor")
```



Thank you for going through my extracurricular stats project. Movies don't necessarily have to be great on Rotten Tomatoes to make a lot of money. More in-depth analysis can be made for these movies and their distributors and we can get a more clear picture of their movie making process and the risk they might be facing whenever they accept a script and want their movie to be successful.

Anyways, I hope you enjoyed some data cleaning and data analysis that I've learned by Googling for the past year at my company and applying this work as a non-traditional actuarial analyst. If you or anyone you know is hiring quantitative analysts or data analysts, please let me know.