Lab 2 EDA

Charles Martin S. Lim

11/13/2021

Load the Data

```
# Read the dataset
original_data <- read.csv(file = "boxoffice2017_2019.csv")
head(original_data)
```

```
##                                     title domestic_revenue  world_revenue
## 1 Star Wars: Episode VIII - The Last Jedi     $620,181,382 $1,332,539,889
## 2                 The Fate of the Furious     $226,008,385 $1,236,005,118
## 3                            Wonder Woman     $412,563,408   $821,847,012
## 4          Guardians of the Galaxy Vol. 2     $389,813,101   $863,756,051
## 5                    Beauty and the Beast     $504,014,165 $1,263,521,126
## 6                                      It     $327,481,748   $700,381,748
##                           distributor opening_revenue opening_theaters
## 1 Walt Disney Studios Motion Pictures    $220,009,584            4,232
## 2                  Universal Pictures     $98,786,705            4,310
## 3                        Warner Bros.    $103,251,471            4,165
## 4 Walt Disney Studios Motion Pictures    $146,510,104            4,347
## 5 Walt Disney Studios Motion Pictures    $174,750,616            4,210
## 6                        Warner Bros.    $123,403,419            4,103
##         budget  MPAA                              genres release_days
## 1 $317,000,000 PG-13     Action,Adventure,Fantasy,Sci-Fi          382
## 2 $250,000,000 PG-13           Action,Adventure,Thriller          262
## 3 $149,000,000 PG-13 Action,Adventure,Fantasy,Sci-Fi,War          217
## 4 $200,000,000 PG-13      Action,Adventure,Comedy,Sci-Fi          241
## 5 $160,000,000    PG      Family,Fantasy,Musical,Romance          290
## 6  $35,000,000     R                              Horror          119
```

Outcome variable:
`world_revenue` - Must be $10,000,000 or more

Explanatory Variables:
- `budget` - Might require a log-transform
- `MPAA` - Filter to indicator variables

Clean The Data

```
convert_to_numeric <- function(df, index){
  return_val <- gsub("\\$", "", df[index])      # Remove $ symbol
  return_val <- gsub(",", "", return_val)       # Remove , symbol
  return_val <- as.numeric(return_val)          # Convert to numeric
```

```r
  return(return_val)
}

# Retrieve only the needed columns

# Outcome variables
world_revenue <-c(original_data$world_revenue)             # World revenue

opening_revenue <-c(original_data$opening_revenue)    # Opening revenue

domestic_revenue <-c(original_data$domestic_revenue)  # Domestic revenue


# Explanatory variables
budget <-c(original_data$budget)                      # Budget
MPAA <-c(original_data$MPAA)                          # Content Rating

# Add title to remove duplicates
title <-c(original_data$title)

df_raw <- data.frame(world_revenue, opening_revenue,
                     domestic_revenue, budget,
                     MPAA, title)

# Convert number columns to numeric
df_raw$world_revenue = apply(df_raw, 1, convert_to_numeric, 1)
df_raw$opening_revenue = apply(df_raw, 1, convert_to_numeric, 2)
df_raw$domestic_revenue = apply(df_raw, 1, convert_to_numeric, 3)
df_raw$budget = apply(df_raw, 1, convert_to_numeric, 4)

head(df_raw)
```

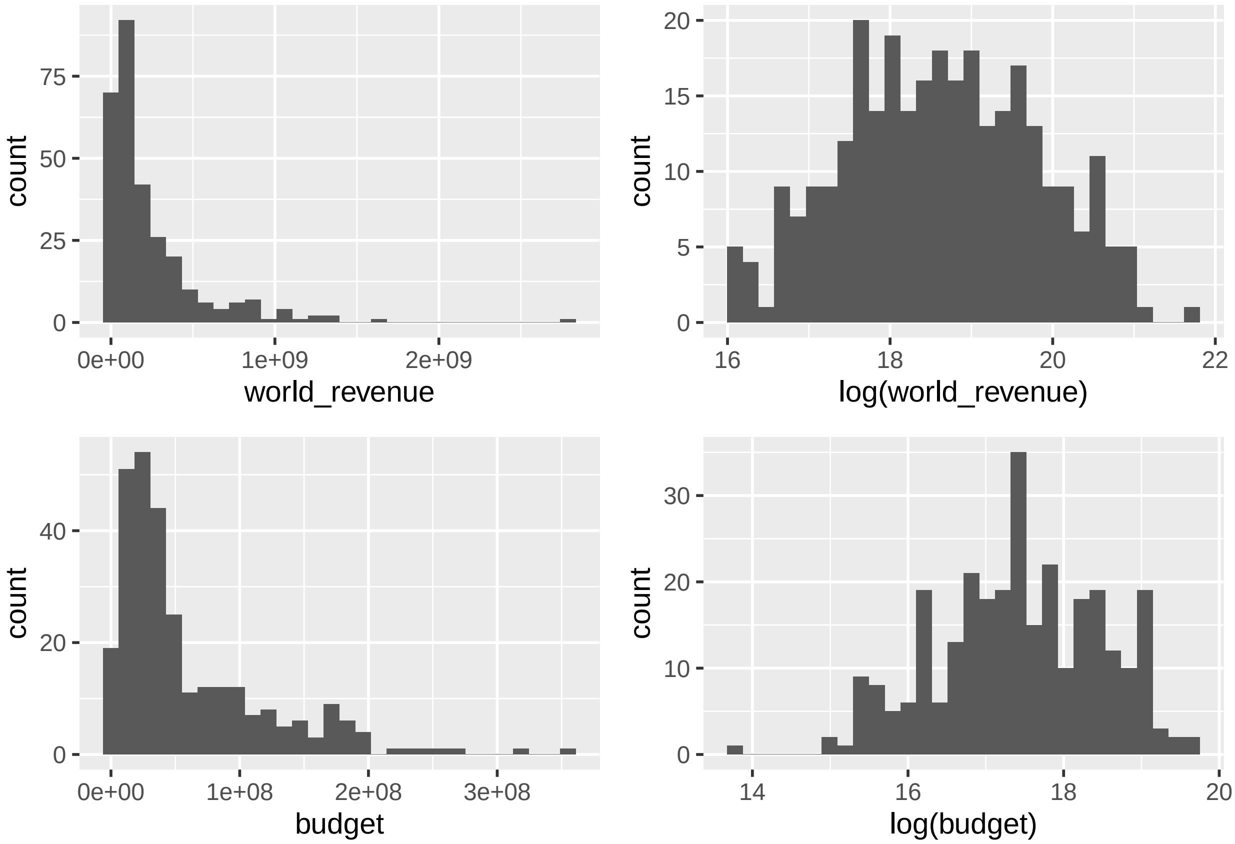
```
##   world_revenue opening_revenue domestic_revenue   budget  MPAA
## 1    1332539889       220009584        620181382 3.17e+08 PG-13
## 2    1236005118        98786705        226008385 2.50e+08 PG-13
## 3     821847012       103251471        412563408 1.49e+08 PG-13
## 4     863756051       146510104        389813101 2.00e+08 PG-13
## 5    1263521126       174750616        504014165 1.60e+08    PG
## 6     700381748       123403419        327481748 3.50e+07     R
##                                     title
## 1 Star Wars: Episode VIII - The Last Jedi
## 2                 The Fate of the Furious
## 3                            Wonder Woman
## 4          Guardians of the Galaxy Vol. 2
## 5                    Beauty and the Beast
## 6                                      It
```

```r
# Remove world revenue under MIN_REVENUE
df_raw <- subset(df_raw, df_raw$world_revenue >= MIN_REVENUE & !is.na(df_raw$world_revenue))

# Remove 0 budget under MIN_BUDGET
df_raw <- subset(df_raw, df_raw$budget >= MIN_BUDGET & !is.na(df_raw$budget))

# Remove duplicate N/A ratings if desired
```

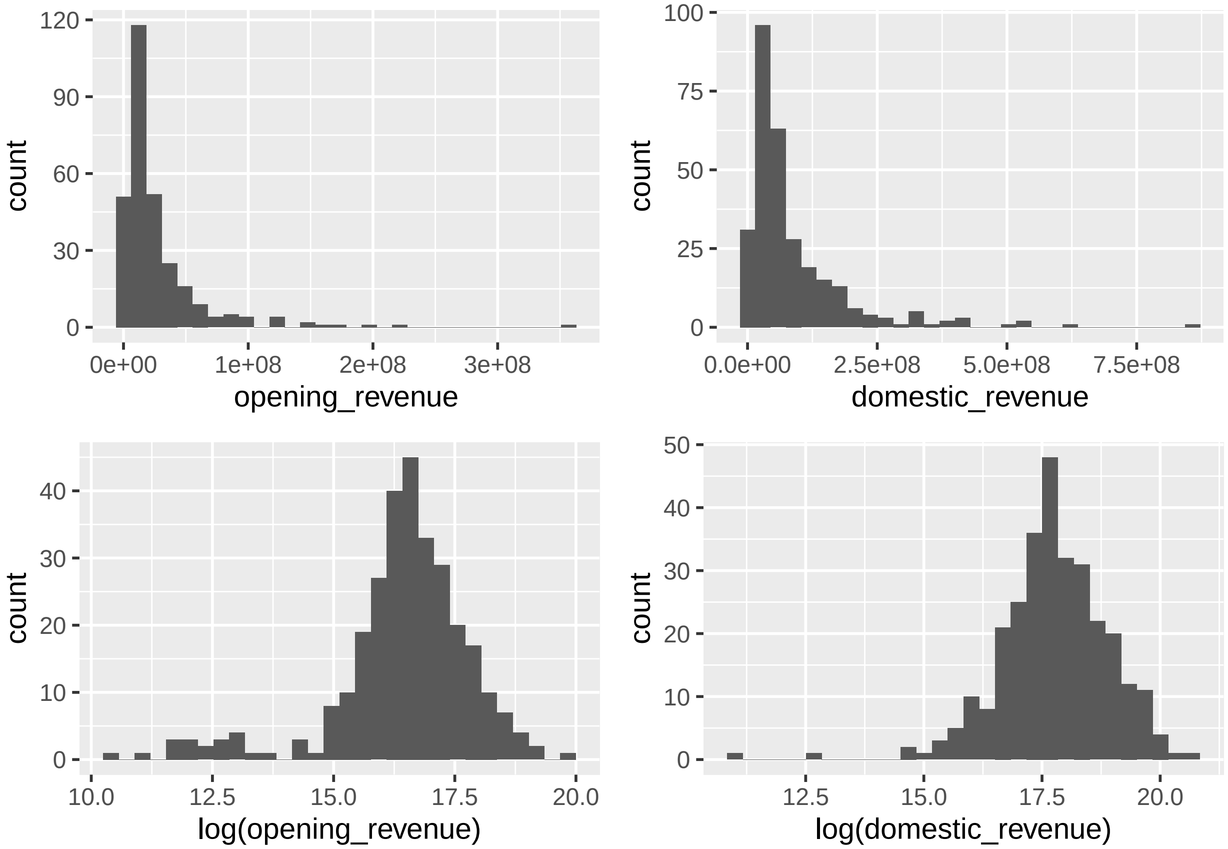
```
if (REMOVE_NA_RATING) {
  df_raw <- subset(df_raw, df_raw$MPAA != "N/A")
}

# Hash for title : budget
h <- hash()

# Clean dataframe
df = data.frame()

for(i in 1:nrow(df_raw)) {       # for-loop over rows
  title_key = df_raw[i,6]

  if (has.key( title_key, h )) {
    # Title Is already recorded

    # Search for existing row in clean dataframe with the same title
    for (k in 1:nrow(df)) {

      if (tolower(title_key) == tolower(df[k,6])) {
        # Replace row if the budget of the new value is higher than that of the
        # budget of the recorded title
        if (df_raw[i, 4] > df[k, 4]) {

          # Delete found row in cleaned dataframe
          df = df[!k,]

          # Bind raw dataframe row to clean dataframe
          df <- rbind(df, df_raw[i,])

          # Revise title_key and budget to hash
          h[[title_key]] = df_raw[i,4]
        }

        break
      }

    }

  } else {
    # Add title_key and budget to hash
    h[[title_key]] = df_raw[i,4]

    # Bind raw dataframe row to clean dataframe
    df <- rbind(df, df_raw[i,])
  }
}

# Print number of rows with unique titles
length(df[["title"]])
```

```
## [1] 295
```

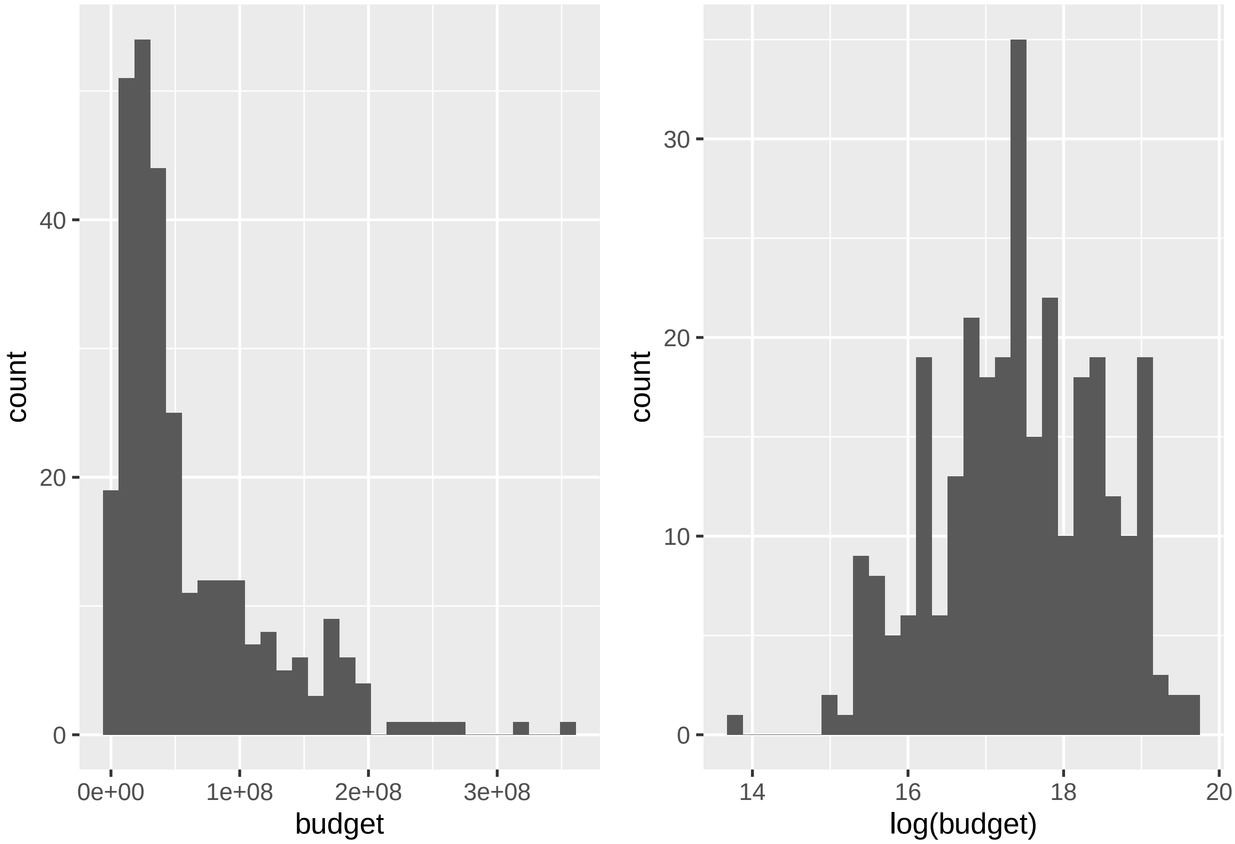
```
# CHECK MAIN NUMERIC VARIABLES
world_revenue_histogram <- df %>%
  ggplot(aes(world_revenue)) +
  geom_histogram(bins=30)

log_world_revenue_histogram <- df %>%
  ggplot(aes(log(world_revenue))) +
  geom_histogram(bins=30)

budget_histogram <- df %>%
  ggplot(aes(budget)) +
  geom_histogram(bins=30)

log_budget_histogram <- df %>%
  ggplot(aes(log(budget))) +
  geom_histogram(bins=30)

grid.arrange(world_revenue_histogram, log_world_revenue_histogram,
             budget_histogram, log_budget_histogram,
             nrow = 2, ncol = 2)
```



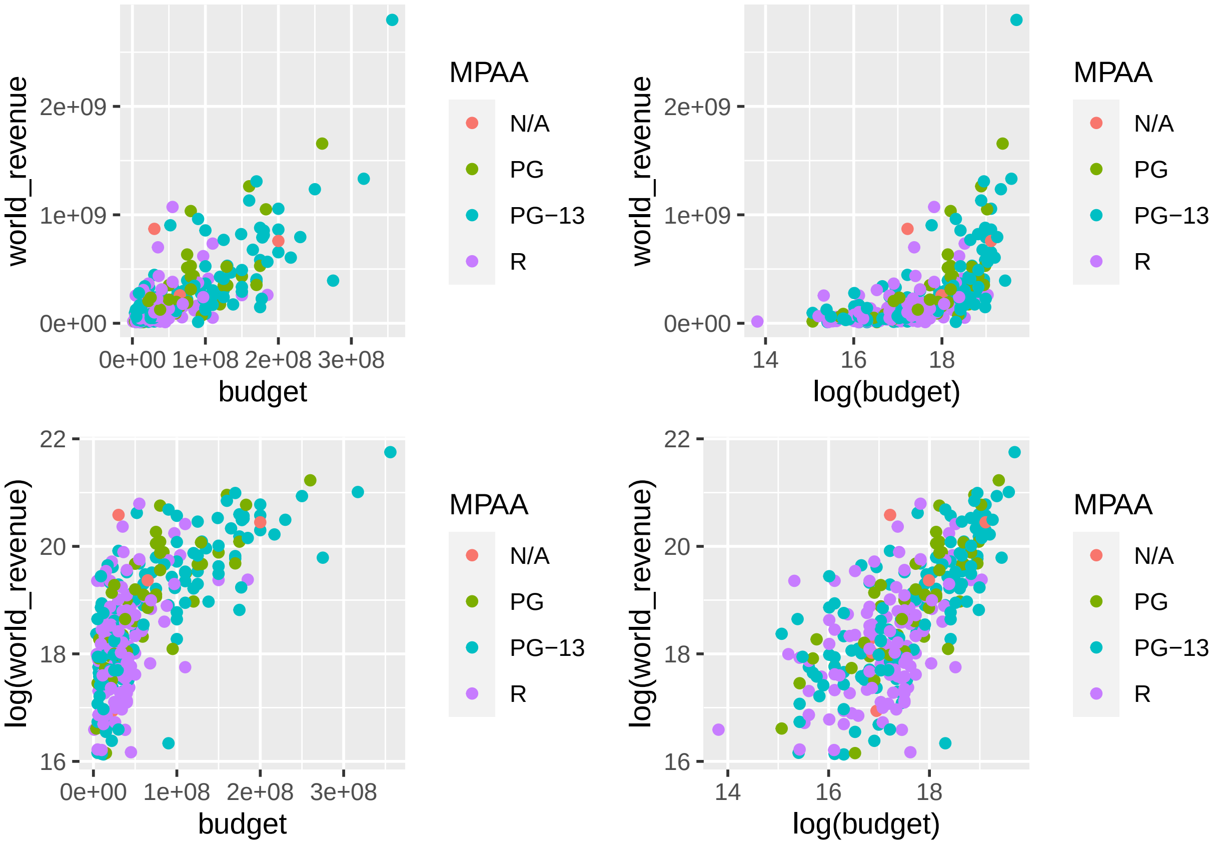
```
# CHECK BONUS OUTCOME VARIABLES
opening_revenue_histogram <- df %>%
  ggplot(aes(opening_revenue,)) +
  geom_histogram(bins=30)

domestic_revenue_histogram <- df %>%
  ggplot(aes(domestic_revenue,)) +
  geom_histogram(bins=30)

log_opening_revenue_histogram <- df %>%
  ggplot(aes(log(opening_revenue))) +
  geom_histogram(bins=30)

log_domestic_revenue_histogram <- df %>%
  ggplot(aes(log(domestic_revenue))) +
  geom_histogram(bins=30)

grid.arrange(opening_revenue_histogram, domestic_revenue_histogram,
             log_opening_revenue_histogram, log_domestic_revenue_histogram,
             nrow = 2, ncol = 2)
```



```
# CHECK NUMERIC EXPLANATORY VARIABLE
budget_histogram <- df %>%
  ggplot(aes(budget)) +
  geom_histogram(bins=30)

log_budget_histogram <- df %>%
  ggplot(aes(log(budget))) +
  geom_histogram(bins=30)


grid.arrange(budget_histogram, log_budget_histogram,
             nrow = 1, ncol = 2)
```



```
# COMPARE EXPLANATORY VARIABLES
level_level_plot <- df %>%
  ggplot(aes(x=budget, y=world_revenue, color=MPAA)) +
  geom_point()

level_log_plot <- df %>%
  ggplot(aes(x=log(budget), y=world_revenue, color=MPAA)) +
  geom_point()

log_level_plot <- df %>%
  ggplot(aes(x=budget, y=log(world_revenue), color=MPAA)) +
  geom_point()

log_log_plot <- df %>%
  ggplot(aes(x=log(budget), y=log(world_revenue), color=MPAA)) +
  geom_point()

grid.arrange(level_level_plot, level_log_plot,
             log_level_plot, log_log_plot,
             nrow = 2, ncol = 2)
```



```
# CHECK PG-RATED FILMS (it appears that G and PG are in the same "PG" bucket)
df_pg <- subset(df, df$MPAA != "PG-13" & df$MPAA != "R" & df$MPAA != "N/A")
df_pg
```

```
##      world_revenue opening_revenue domestic_revenue   budget MPAA
## 5       1263521126       174750616        504014165 1.60e+08   PG
## 10      1034799409        72434025        264624300 8.00e+07   PG
## 16       634151679        35258145        270395425 7.50e+07   PG
## 726      131457147        26608020         68549695 4.20e+07   PG
## 1265      62812974         9812674         40852824 2.00e+07   PG
## 1454      10367161          304933          8874389 1.50e+07   PG
## 1573     158970776         1638895          4412170 2.50e+07   PG
## 1736     173961069        20352491         54858851 1.20e+08   PG
## 1749      16376066         4245630         16376066 3.50e+06   PG
## 1793      52090236        10411189         29790236 1.00e+07   PG
## 1830     296069199        13401586         84410380 1.11e+08   PG
## 1910      90497778        10604774         43242871 5.90e+07   PG
## 1916      93320380        15802225         46700633 3.50e+07   PG
## 1988     197744377        24585139         99215042 7.50e+07   PG
## 1989      86026201        17108914         83482352 7.00e+06   PG
## 1995     349537494        23523121        171958438 1.30e+08   PG
## 1997     351266433        25010928        115253424 5.00e+07   PG
## 2003     434993183         8805843        174340174 8.40e+07   PG
## 2033     528583774        44076225        167510016 8.00e+07   PG
## 2035     529323962        56237634        201091711 1.75e+08   PG
## 2049     183388953        21311407         72679278 7.00e+07   PG
## 2074     511595957        67572855        270620950 7.50e+07   PG
## 2410      71954915         6619870         27154915 9.50e+07   PG
## 2423     346864462        46581142        153707064 1.25e+08   PG
## 2481      38026103         8146533         34746945 5.00e+06   PG
## 2487     178027844        16755310        102961255 4.00e+07   PG
## 2490      50444358        11282333         40713082 1.40e+07   PG
## 2495     154656687        10354073         41667116 6.50e+07   PG
## 2496      80708134        11251263         42004346 1.80e+07   PG
## 2521      64391669        12723781         44451847 2.99e+07   PG
## 2534     120729461        17431588         60477943 4.90e+07   PG
## 2537      65797820        13251238         61335815 2.50e+07   PG
## 2539     189772088        20612100         60716390 7.50e+07   PG
## 2566      40140972         7126084         20738724 2.20e+07   PG
## 2588     353284621        45990748        114766307 1.70e+08   PG
## 2590      60330833         8885899         21885107 6.50e+06   PG
## 2601      68743485         8342311         28370522 4.00e+07   PG
## 2621     433005346        54365242        144105346 1.50e+08   PG
## 2622     430051293        46652680        158874395 8.00e+07   PG
## 2623     521799505        55022245        160799505 1.29e+08   PG
## 2626    1050693953        91500929        355559216 1.83e+08   PG
## 2638    1656943394       191770759        543638043 2.60e+08   PG
## 2649     197183546        13210449         45020282 6.00e+07   PG
## 2669     205035819        18222810         64508620 2.20e+07   PG
## 2670     125427681        23851539         73921000 3.80e+07   PG
## 2672     217776646        24531923         86089513 5.00e+07   PG
## 2685     235956898          515499        169607287 2.50e+07   PG
## 2691     311950384        53003468        175750384 8.00e+07   PG
##                                           title
## 5                          Beauty and the Beast
## 10                              Despicable Me 3
## 16                                         Sing
## 726         The House with a Clock in Its Walls
## 1265                                   The Star
## 1454                             Queen of Katwe
## 1573               Johnny English Strikes Again
## 1736         The Nutcracker and the Four Realms
## 1749                            Forever My Girl
## 1793              Teen Titans GO! to the Movies
## 1830                                  Ferdinand
## 1910                            Sherlock Gnomes
## 1916            Goosebumps 2: Haunted Halloween
## 1988                          Christopher Robin
## 1989                         I Can Only Imagine
## 1995                       Mary Poppins Returns
## 1997                               Peter Rabbit
## 2003                       The Greatest Showman
## 2033      Hotel Transylvania 3: Summer Vacation
## 2035                  Ralph Breaks the Internet
## 2049                                     Storks
## 2074                      Dr. Seuss’ The Grinch
## 2410                                       Cats
## 2423                                     Trolls
## 2481                                  Overcomer
## 2487                               Little Women
## 2490                               Breakthrough
## 2495                   The Angry Birds Movie 2
## 2496                          A Dog’s Way Home
## 2521                         Playing with Fire
## 2534            Dora and the Lost City of Gold
## 2537       A Beautiful Day in the Neighborhood
## 2539                                Abominable
## 2566       Diary of a Wimpy Kid: The Long Haul
## 2588                                     Dumbo
## 2590                  My Little Pony: The Movie
## 2601             The Nut Job 2: Nutty by Nature
## 2621              Pok\xe9mon Detective Pikachu
## 2622                  The Secret Life of Pets 2
## 2623 How to Train Your Dragon: The Hidden World
## 2626                                   Aladdin
## 2638                             The Lion King
## 2649                   Smurfs: The Lost Village
## 2669                            A Dog’s Purpose
## 2670   Captain Underpants: The First Epic Movie
## 2672                            The Emoji Movie
## 2685                             Hidden Figures
## 2691                      The Lego Batman Movie
```