STOR 455 Group Project (Due 5pm on November 24th)

Marvel Revenue Avengers

Liujie Zheng (730427441), Yuhan Zhou (730621041), Zhetao Zhang (730491003), Eliana Li (730521478), Haoyang Li (730610085)

The Prediction (Required)

Our prediction of the cumulative domestic box office of “The Marvels” by December 8, 2023 is **$87, 389, 387**

Summary of Justification (Required)

For this prediction task, we collected the **daily box office revenues** and the **Rotten Tomatoes scores** of all 33 Marvel movies. We found out that the cumulative domestic box office revenues of Marvel movies at their Day 29 since released are **highly correlated** with their cumulative **domestic box office at Day 13**, and are **moderately correlated** with their **Rotten Tomato scores**. We also know that for “The Marvels”, December 8, 2023 is its Day 29 since released. With these information, we fit a **simple linear regression model** to predict the cumulative domestic box office of “The Marvels” by December 8, 2023 from its cumulative **domestic box office at November 22 (Day 13)** and its **Rotten Tomato scores**. The model has a **train percentage error of 3.239%**. Our data and code can be found at this Github repository.

Data (Required)

For this prediction task, we collected the **daily box office revenues** and the **Rotten Tomatoes scores** of all 33 Marvel movies. The daily box office revenues are extracted from Box Office Mojo, into the **data_raw** folder (scrapper code in **scripts/data_scrapper.py**); the Rotten Tomatoes scores are collected from Wikipedia into the **metadata.csv**.

Metadata.csv

```r
metadata = read.csv("metadata.csv")
dim(metadata)
```

```
## [1] 33  4
```

metadata.csv contains 33 rows and 4 columns.

```r
str(metadata)
```

```
## 'data.frame':    33 obs. of  4 variables:
##  $ Movie        : chr  "Iron Man" "The Incredible Hulk" "Iron Man 2" "Thor" ...
##  $ Release.Date : chr  "2-May-08" "13-Jun-08" "7-May-10" "6-May-11" ...
##  $ Rotten.Tomato: chr  "94% (281 reviews)[288]" "67% (239 reviews)[291]" "72% (304 reviews)[294]" "77
##  $ Url          : chr  "https://www.boxofficemojo.com/release/rl1482327553/" "https://www.boxofficemo
```

Each row of **metadata.csv** represents a Marvel movie. Explanation of each column:

Movie: Name of the Marvel movie

Release.Date: Release date of the movie in day-month-year

Rotten.Tomato: Rotten Tomatoes score

Url: Link to the domestic daily box office revenue

Raw Data

Each "Movie Name.csv" file in **data_raw** folder gives the **domestic daily box office revenue** of that movie. Take the movie "Ant-Man and the Wasp" as an example:

```r
ant_man = read.csv("data_raw/Ant-Man and the Wasp.csv")
dim(ant_man)
```

```
## [1] 119  11
```

data_raw/Ant-Man and the Wasp.csv contains 119 rows and 11 columns

```r
str(ant_man)
```

```
## 'data.frame':    119 obs. of  11 variables:
##  $ Date     : chr  "Jul 6World Cup (Russia)" "Jul 7World Cup (Russia)" "Jul 8World Cup (Russia)" "Jul
##  $ DOW      : chr  "Friday" "Saturday" "Sunday" "Monday" ...
##  $ Rank     : int  1 1 1 1 1 1 1 3 2 2 ...
##  $ Daily    : chr  "$33,725,082" "$23,555,372" "$18,531,751" "$6,983,824" ...
##  $ X...YD   : chr  "-" "-30.2%" "-21.3%" "-62.3%" ...
##  $ X...LW   : chr  "-" "-" "-" "-" ...
##  $ Theaters : chr  "4,206" "4,206" "4,206" "4,206" ...
##  $ Avg      : chr  "$8,018" "$5,600" "$4,406" "$1,660" ...
##  $ To.Date  : chr  "$33,725,082" "$57,280,454" "$75,812,205" "$82,796,029" ...
##  $ Day      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Estimated: chr  "false" "false" "false" "false" ...
```

Each row represents a consecutive day. Explanation of each column:

Date: Screening day

DOW: Day of the week

Rank: The movie's rank in terms of box office revenue compared to other movies showing on the same day

Daily: The total box office revenue generated by the movie on that specific day

X. . . YD: Percent Change from Yesterday

X. . . LW: Percent Change from Last Week

Theaters:The number of theaters in which the movie was shown on that day

Avg: Average Revenue per Theater

To.Date: The total cumulative box office revenue for the movie up to and including that day

Day: The number of days since the movie was first released

Estimated: Whether the box office numbers are actual (false) or estimated (true) figures. 'True' suggests that the figures are preliminary and may be adjusted later, while 'False' indicates that the figures are final

For "The Marvels" movie, we have the daily box office revenue of the **first 13 days**.

```
the_marvels = read.csv("data_raw/The Marvels.csv")
dim(the_marvels)
```

```
## [1] 13 11
```

```
str(the_marvels)
```

```
## 'data.frame':    13 obs. of  11 variables:
##  $ Date     : chr  "Nov 10" "Nov 11" "Nov 12" "Nov 13" ...
##  $ DOW      : chr  "Friday" "Saturday" "Sunday" "Monday" ...
##  $ Rank     : int  1 1 1 1 1 1 1 4 3 3 ...
##  $ Daily    : chr  "$21,603,104" "$15,260,052" "$9,247,703" "$2,372,375" ...
##  $ X...YD   : chr  "-" "-29.4%" "-39.4%" "-74.3%" ...
##  $ X...LW   : chr  "-" "-" "-" "-" ...
##  $ Theaters : chr  "4,030" "4,030" "4,030" "4,030" ...
##  $ Avg      : chr  "$5,360" "$3,786" "$2,294" "$588" ...
##  $ To.Date  : chr  "$21,603,104" "$36,863,156" "$46,110,859" "$48,483,234" ...
##  $ Day      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Estimated: chr  "false" "false" "false" "false" ...
```

Data Cleaning

To get the data ready for analysis, we:

1) separated **Date** and **Event**

2) turned **DOW** into 1 to 7

3) turned `-` mark into `NA`

4) turned columns like **Daily**, **X...YD** and others into numeric

The data cleaning code is in **scripts/data_cleaner.py**. Here is cleaned data of "Ant-Man and the Wasp" :

```
ant_man_cleaned = read.csv("data_cleaned/Ant-Man and the Wasp_cleaned.csv")
str(ant_man_cleaned)
```

```
## 'data.frame':    119 obs. of  12 variables:
##  $ Date     : chr  "Jul 6" "Jul 7" "Jul 8" "Jul 9" ...
##  $ Event    : chr  "World Cup (Russia)" "World Cup (Russia)" "World Cup (Russia)" "World Cup (Russia
##  $ DOW      : int  5 6 7 1 2 3 4 5 6 7 ...
##  $ Rank     : int  1 1 1 1 1 1 1 3 2 2 ...
##  $ Daily    : int  33725082 23555372 18531751 6983824 10042976 5852591 5293629 8431124 11756671 8910
##  $ YD.      : num  NA -30.2 -21.3 -62.3 43.8 -41.7 -9.6 59.3 39.4 -24.2 ...
##  $ LW.      : num  NA NA NA NA NA -82.6 -77.5 -75 -50.1 -51.9 ...
##  $ Theaters : int  4206 4206 4206 4206 4206 4206 4206 4206 4206 4206 ...
##  $ Avg      : int  8018 5600 4406 1660 2387 1391 1258 2004 2795 2118 ...
##  $ To.Date  : int  33725082 57280454 75812205 82796029 92839005 98691596 103985225 112416349 1241730
##  $ Day      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Estimated: chr  "False" "False" "False" "False" ...
```

And the cleaned data of "The Marvels":

```r
the_marvels_cleaned = read.csv("data_cleaned/The Marvels_cleaned.csv")
str(the_marvels_cleaned)
```

```
## 'data.frame':    13 obs. of  12 variables:
##  $ Date     : chr  "Nov 10" "Nov 11" "Nov 12" "Nov 13" ...
##  $ Event    : logi  NA NA NA NA NA NA ...
##  $ DOW      : int  5 6 7 1 2 3 4 5 6 7 ...
##  $ Rank     : int  1 1 1 1 1 1 1 4 3 3 ...
##  $ Daily    : int  21603104 15260052 9247703 2372375 3300946 1789239 1251387 2756659 4453682 2910248
##  $ YD.      : num  NA -29.4 -39.4 -74.3 39.1 ...
##  $ LW.      : num  NA NA NA NA NA NA NA -87.2 -70.8 -68.5 ...
##  $ Theaters : int  4030 4030 4030 4030 4030 4030 4030 4030 4030 4030 ...
##  $ Avg      : int  5360 3786 2294 588 819 443 310 684 1105 722 ...
##  $ To.Date  : int  21603104 36863156 46110859 48483234 51784180 53573419 54824806 57581465 62035147 
##  $ Day      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Estimated: chr  "False" "False" "False" "False" ...
```

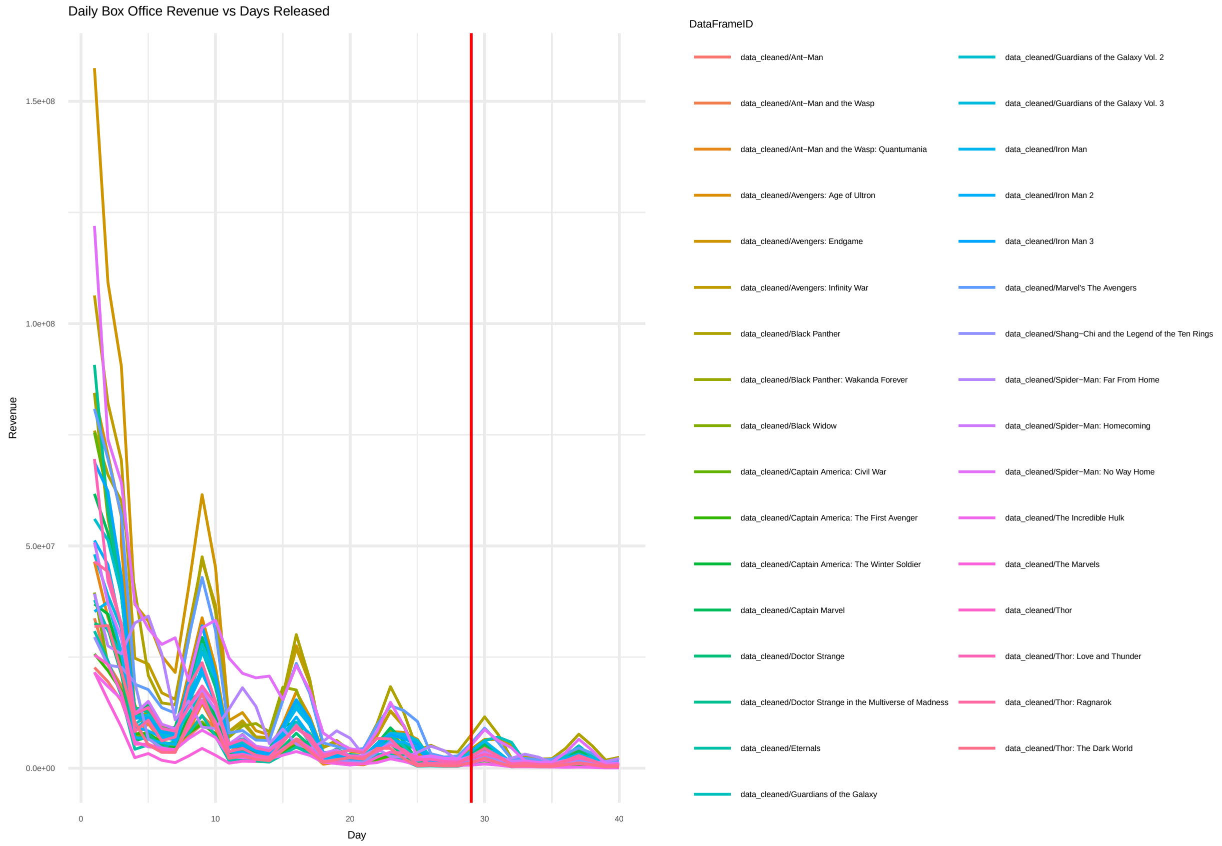
Analysis (Required)

Data Visualization

With our data cleaned, we first want to check out how the box office revenue varies with respect to the day since the movie was first released.

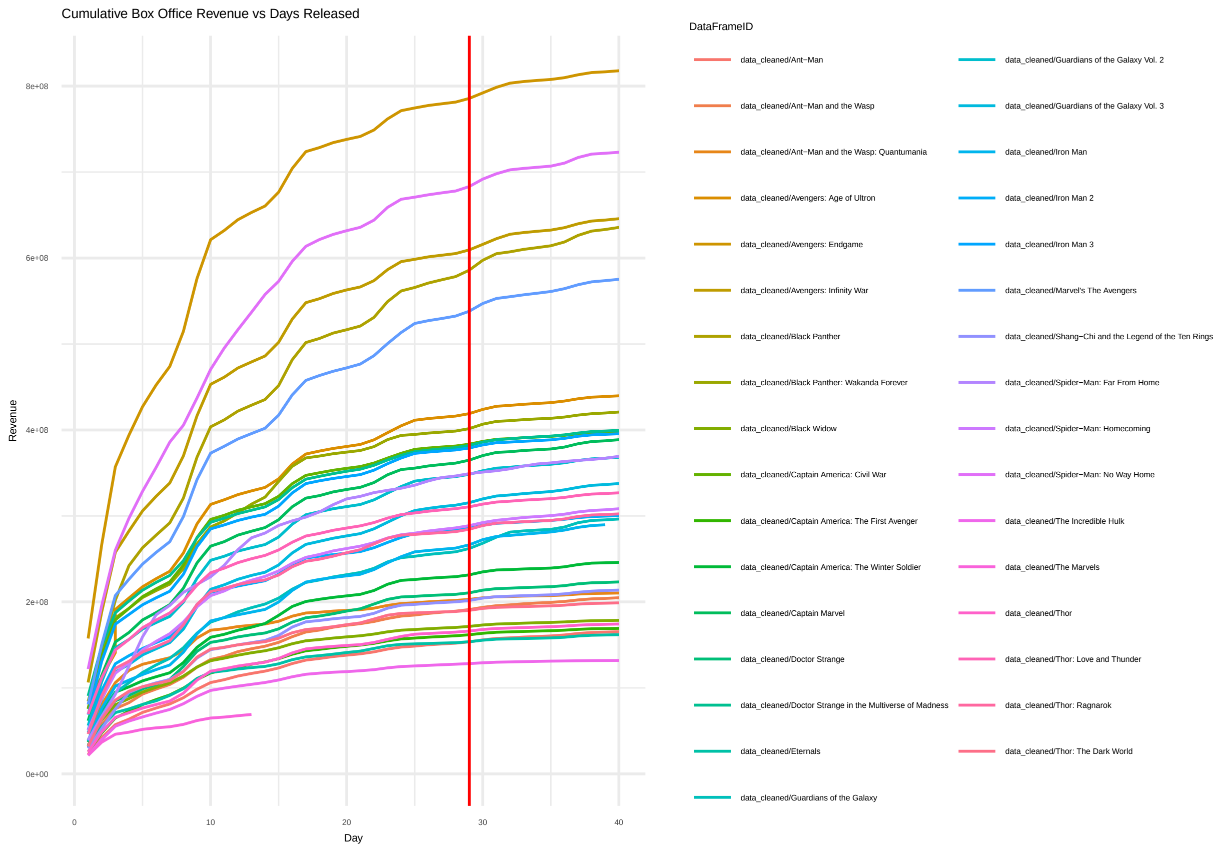
```r
list_of_movies_40 <- lapply(list_of_movies, function(df) {
  head(df, 40)
})
list_of_movies_40 <- lapply(list_of_movies_40, function(df) {
    df$Day <- as.numeric(df$Day)
    return(df)
})
```

```r
combined_df <- bind_rows(list_of_movies_40, .id = "DataFrameID")
ggplot(combined_df, aes(x = Day, y = `Daily`, group = DataFrameID, color = DataFrameID)) +
  geom_line() +
  geom_vline(xintercept = 29, color = "red", size = 0.5) +
  theme_minimal() +
  theme(text = element_text(size = 4)) +
  labs(title = "Daily Box Office Revenue vs Days Released", x = "Day", y = "Revenue")
```



As the **Daily Box Office Revenue vs Days Released** plot tells, the daily box office revenues of the movies share a similar trend: drop drastically in the initial several days and have periodically peaks. These peaks usually represents Friday, Saturday and Tuesday.

```
ggplot(combined_df, aes(x = Day, y = `To Date`, group = DataFrameID, color = DataFrameID)) +
  geom_line() +
  geom_vline(xintercept = 29, color = "red", size = 0.5) +
  theme_minimal() +
  theme(text = element_text(size = 4)) +
  labs(title = "Cumulative Box Office Revenue vs Days Released", x = "Day", y = "Revenue")
```



The lines in **Cumulative Box Office Revenue vs Days Released** plot barely intersects. This means that **if a movie performs well in the first several days, it will keep doing so**, and vice versa. We then suppose that from the cumulative box office revenue of "The Marvels" at Day 13, we are able to predict the box office revenue at Day 29 (December 8).

We would like to further verify that by calculating the correlation between these two values.

```
# Get all file names of individual marvel movies csv from the folder
file <- list.files(path = "data_cleaned", pattern = "\\.csv$", full.names = TRUE)
file_names <- basename(file)

# helper function to extract cumulative box office revenue at a certain day
extract_data <- function(file, day) {
   data <- read.csv(file)
   if (ncol(data) >= 10 && nrow(data) >= 10) {
    return(data[day, 10])
  }
}
day29 <- lapply(all_files, extract_data, 29)

movie_titles <- gsub("_cleaned\\.csv", "", file_names)
metadata <- arrange(metadata, Movie)
final_data <- data.frame(Movie = movie_titles, Days_29 = unlist(day29))
final_data <- arrange(final_data, Movie)
final_data$Movie = metadata$Movie

day13 <- lapply(file, extract_data, 13)
```

```
total <- data.frame(Movie = movie_titles, Days_13 = unlist(day13))
total <- arrange(total,Movie)
total$Movie <- metadata$Movie
final_data <- merge(final_data,total, by = "Movie", all = T)
final_movie <- final_data[-29,]
str(final_movie)
```

```
## ’data.frame':    32 obs. of  3 variables:
##  $ Movie  : chr  "Ant-Man" "Ant-Man and the Wasp" "Ant-Man and the Wasp: Quantumania" "Avengers: Age
##  $ Days_29: int  153588288 191113233 202943135 419033597 785610412 609429495 585858908 401571467 1715
##  $ Days_13: int  116814026 145506045 172867770 329133743 652935585 479116406 428792346 313723465 1408
```

```
cor(x=final_movie$Days_13, y=final_movie$Days_29, use='complete.obs')
```

```
## [1] 0.9957663
```

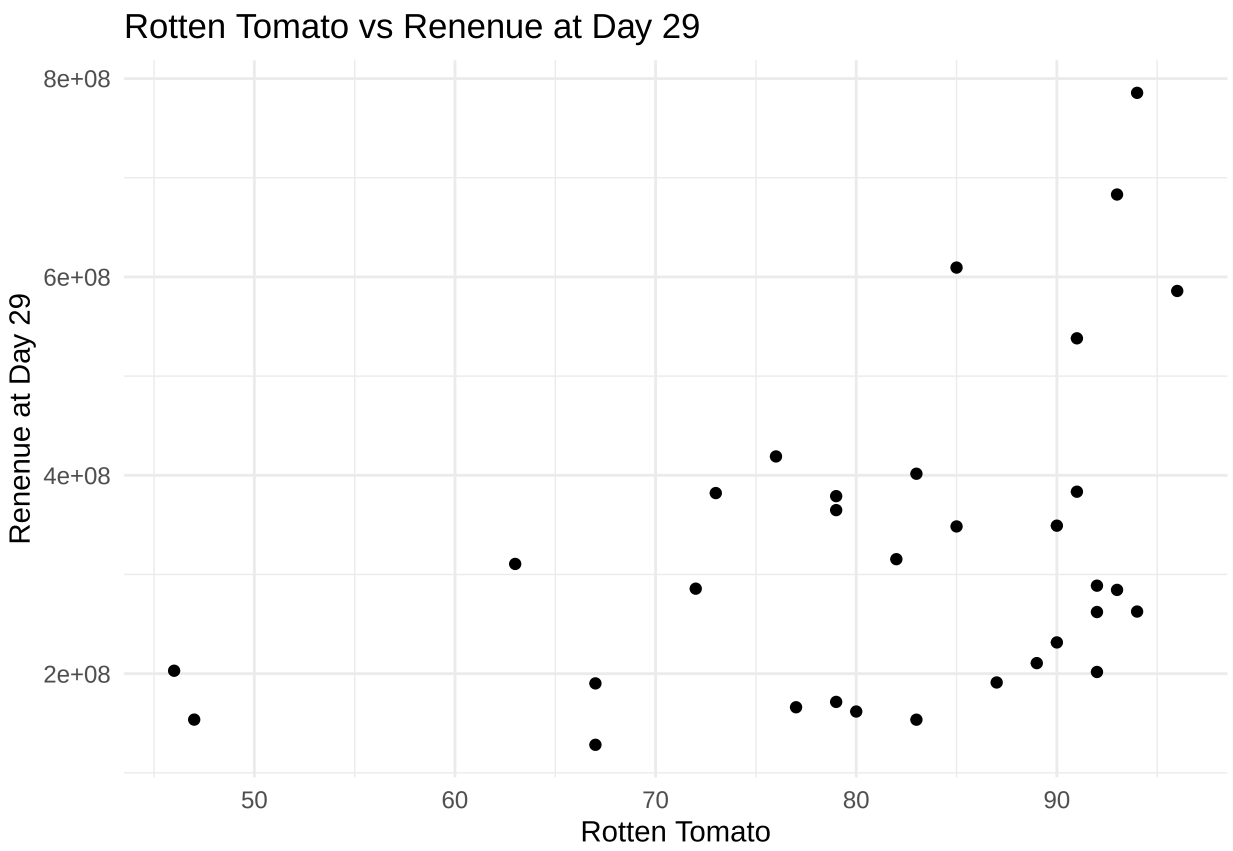
The correlation is >**0.99**, which is extremely high.

Next, we would like to see how the **Rotten Tomato scores** influences the box office Revenue. Since the Rotten Tomato scores are in **metadata.csv**, we need to combine them with cumulative box office revenue at Day 29.

```
# extact and convert the Rotten Tomato scores to numeric values
rotten_tomato <- read.csv("metadata.csv")[-c(2,4)]
rotten_tomato$Rotten.Tomato <- as.numeric(gsub("^(\\d+).*", "\\1", rotten_tomato$Rotten.Tomato))
final_movie_2 <- merge(rotten_tomato, final_movie, by = "Movie", all = TRUE)
str(final_movie_2)
```

```
## ’data.frame':    33 obs. of  4 variables:
##  $ Movie        : chr  "Ant-Man" "Ant-Man and the Wasp" "Ant-Man and the Wasp: Quantumania" "Avenger
##  $ Rotten.Tomato: num  83 87 46 76 94 85 96 83 79 91 ...
##  $ Days_29      : int  153588288 191113233 202943135 419033597 785610412 609429495 585858908 40157146
##  $ Days_13      : int  116814026 145506045 172867770 329133743 652935585 479116406 428792346 31372346
```

```
ggplot(final_movie_2, aes(x = Rotten.Tomato, y = Days_29)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Rotten Tomato vs Renenue at Day 29", x = "Rotten Tomato", y = "Renenue at Day 29")
```



```
cor(x=final_movie_2$Rotten.Tomato, y=final_movie_2$Days_29, use='complete.obs')
```

```
## [1] 0.4206142
```

From the plot and the correlation calculation. There exists a **intermediate positive correlation** between Rotten Tomato scores and the cumulative box office revenue at Day 29. That means we can **potentially** include it in our model.

Simple Linear Regression

Given all the information we have, we decided to try out two models:

- Day_29 = beta_0 + beta_1 * Day_13 + beta_2 * Rotten_tomato + e
- Day_29 = beta_0 + beta_1 * Day_13 + e

```
# 29 is "The Marvels", exclude it in model fitting
mod1 = lm(Days_29~Days_13+Rotten.Tomato,data=final_movie_2)
prediction1 <- predict(mod1,newdata=final_movie_2)[-29]

mae <- mean(abs(final_movie_2$Days_29[-29] - prediction1))
print(mae)
```

```
## [1] 9015337
```

```r
percentage_errors <- abs((final_movie_2$Days_29[-29] - prediction1) /
                           final_movie_2$Days_29[-29])
mape <- mean(percentage_errors)
print(mape)
```

```
## [1] 0.0284575
```

The first model gives a 2.846% error.

```r
mod2 = lm(Days_29~Days_13,data=final_movie_2)
prediction2 <- predict(mod2,newdata=final_movie_2)[-29]

mae <- mean(abs(final_movie_2$Days_29[-29] - prediction2))
print(mae)
```

```
## [1] 9934444
```

```r
percentage_errors <- abs((final_movie_2$Days_29[-29] - prediction2) /
                           final_movie_2$Days_29[-29])
mape <- mean(percentage_errors)
print(mape)
```

```
## [1] 0.03239431
```

The second model gives a 3.239% error. We choose to stick with the first model.

```r
final_movie_2[29,4] = 69153446
predict(mod1,newdata=final_movie_2[29,])
```

```
##       29
## 87389387
```

Therefore, our prediction of the cumulative domestic box office of “The Marvels” by December 8, 2023 is 87389387.