# Micheal Bay Movie Survey

## Survey

In this survey I asked my participants to rate all of the movies that Micheal Bay directed. The participants responded on Microsoft Form which is a online survey tool Survey. The survey consists of putting entering their name and rating every Micheal Bay movie that he has created. When the participants finish entering I then will download the excel file

```
library(readxl)

if (!file.exists('Micheal Bay Movie Rating.xlsx')) {
  download.file('https://github.com/xvicxpx/Data607/blob/main/Homework%202/Micheal%20Bay%20Movie%20Ratin
}

data_set = read_excel('Micheal Bay Movie Rating.xlsx')
```

## SQLite and Importing Data

I will be loading the data into a local database using SQLite which is a light weight version of SQL. The only downside is that SQlite does not have a lot of functions compared to MySQL, but because we will be doing a lot of the data manipulation inside of R it should not be an issue

First I will be creating the database and loading the data into the database. But before I load the data into the database I will also clean up the data in R first

```
library(DBI)
conn = dbConnect(RSQLite::SQLite(), "database.db")
```

```
data_set
```

```
## # A tibble: 6 x 18
##      ID `Start time`        `Completion time`   Email Name  Name2 `Bad Boys`
##   <dbl> <dttm>              <dttm>              <chr> <lgl> <chr>      <dbl>
## 1     1 2021-02-04 16:45:37 2021-02-04 16:46:17 anon~ NA    Nate           2
## 2     2 2021-02-04 17:03:36 2021-02-04 17:04:14 anon~ NA    Step~          1
## 3     3 2021-02-04 18:05:20 2021-02-04 18:05:45 anon~ NA    Vic ~          2
## 4     4 2021-02-04 18:08:11 2021-02-04 18:08:49 anon~ NA    Kash~          4
## 5     5 2021-02-04 18:20:04 2021-02-04 18:21:49 anon~ NA    Gint~         NA
## 6     6 2021-02-04 18:50:48 2021-02-04 18:51:16 anon~ NA    Sapt~          4
## # ... with 11 more variables: `The Rock` <dbl>, Armageddon <dbl>, `Bad Boys
## #   II` <dbl>, `The Island` <dbl>, Transformers <dbl>, `Transformers: Revenge
## #   of the Fallen` <dbl>, `Transformers: Dark of the Moon` <dbl>, `Pain &
## #   Gain` <dbl>, `Transformers: Age of Extinction` <dbl>, `13 Hours: The Secret
## #   Soldiers of Benghazi` <dbl>, `6 Underground` <dbl>
```

Looking at the dataset we can see that there are many colmns that are most likely not going to be needed. We can also see that some of the movie ratings are NA which mean that person did not see that movie.

```
library(reshape2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data_set = data_set[c('Start time', 'Completion time', 'Name2', 'Bad Boys', 'The Rock', 'Armageddon', '

data_set = melt(data_set, id=c('Start time', 'Completion time', 'Name2'), variable.name = 'Movie', valu

data_set = data_set %>% rename(Participant = Name2)

head(data_set)
```

```
##            Start time     Completion time       Participant    Movie Rating
## 1 2021-02-04 16:45:37 2021-02-04 16:46:17             Nate Bad Boys      2
## 2 2021-02-04 17:03:36 2021-02-04 17:04:14       Stephen Ren Bad Boys      1
## 3 2021-02-04 18:05:20 2021-02-04 18:05:45          Vic Chan Bad Boys      2
## 4 2021-02-04 18:08:11 2021-02-04 18:08:49 Kashyap Gummaraju Bad Boys      4
## 5 2021-02-04 18:20:04 2021-02-04 18:21:49            Gintas Bad Boys     NA
## 6 2021-02-04 18:50:48 2021-02-04 18:51:16     Saptarsi Guha Bad Boys      4
```

Now that the data has been cleaned I am going to load the data into the database that was created with SQLite. The reason why I set the table to be overwritten every time I import is because I will not be appending and instead erasing the table and uploading the information again every time

```
dbWriteTable(conn, 'movie_rating', data_set, overwrite = TRUE)
```

Verifying that the data went into the database correctly

```
dbListTables(conn)
dbReadTable(conn, 'movie_rating')
```

Looking at the columns Start time and Completion time it seems like that they need to be converted back into time units

```
c = dbSendQuery(conn, "SELECT
                  datetime([Start time],'unixepoch', 'localtime') AS [Start Time],
                  datetime([Completion time], 'unixepoch', 'localtime') AS [Completion Time],
```

```
                  Participant,
                  Movie,
                  Rating
                FROM movie_rating")
dbFetch(c)
```

Seeing as I have fixed the datetime issue I will put it into a new table with the fix datetime

```
if (dbExistsTable(conn, 'movie_rating_fix')) {
  dbRemoveTable(conn, 'movie_rating_fix')
}
```

```
## Warning: Closing open result set, pending rows
```

```
dbExecute(conn, "CREATE TABLE movie_rating_fix
                AS
                  SELECT
                  datetime([Start time],'unixepoch', 'localtime') AS [Start Time],
                  datetime([Completion time], 'unixepoch', 'localtime') AS [Completion Time],
                  Participant,
                  Movie,
                  Rating
                  FROM movie_rating
                ")

dbReadTable(conn, 'movie_rating_fix')
```

If a participant did not see a movie then we can see that the rating will be NULL in SQL. We can see that the participant has not seen a lot of Micheal Bay movies which is a shame.

```
c = dbSendQuery(conn, "SELECT * FROM movie_rating_fix WHERE Rating IS NULL")

dbFetch(c)
```

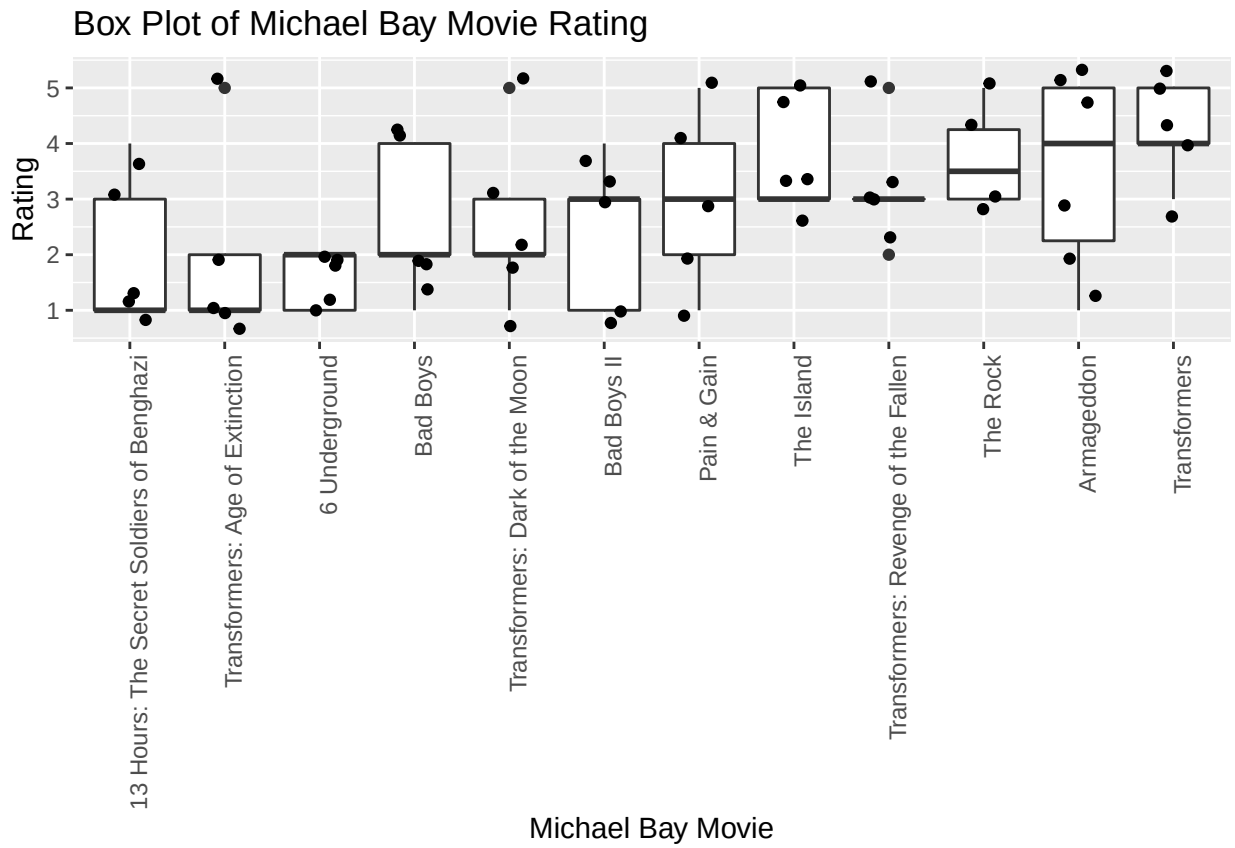
## Anaylsis of Michael Bay Movie

We are going to see that is the average rating of each rating in order to see which one to recommend. I am also going to make sure to not include any null values as that means the person has not seen the movie before. I personally would recommend Armageddon because why train astronauts to become drillers when you can train drillers to become astronauts. Makes total sense to me.

```
c = dbSendQuery(conn, "SELECT
                       movie, Rating, Participant
                       FROM movie_rating_fix
                       WHERE Rating IS NOT NULL")
```

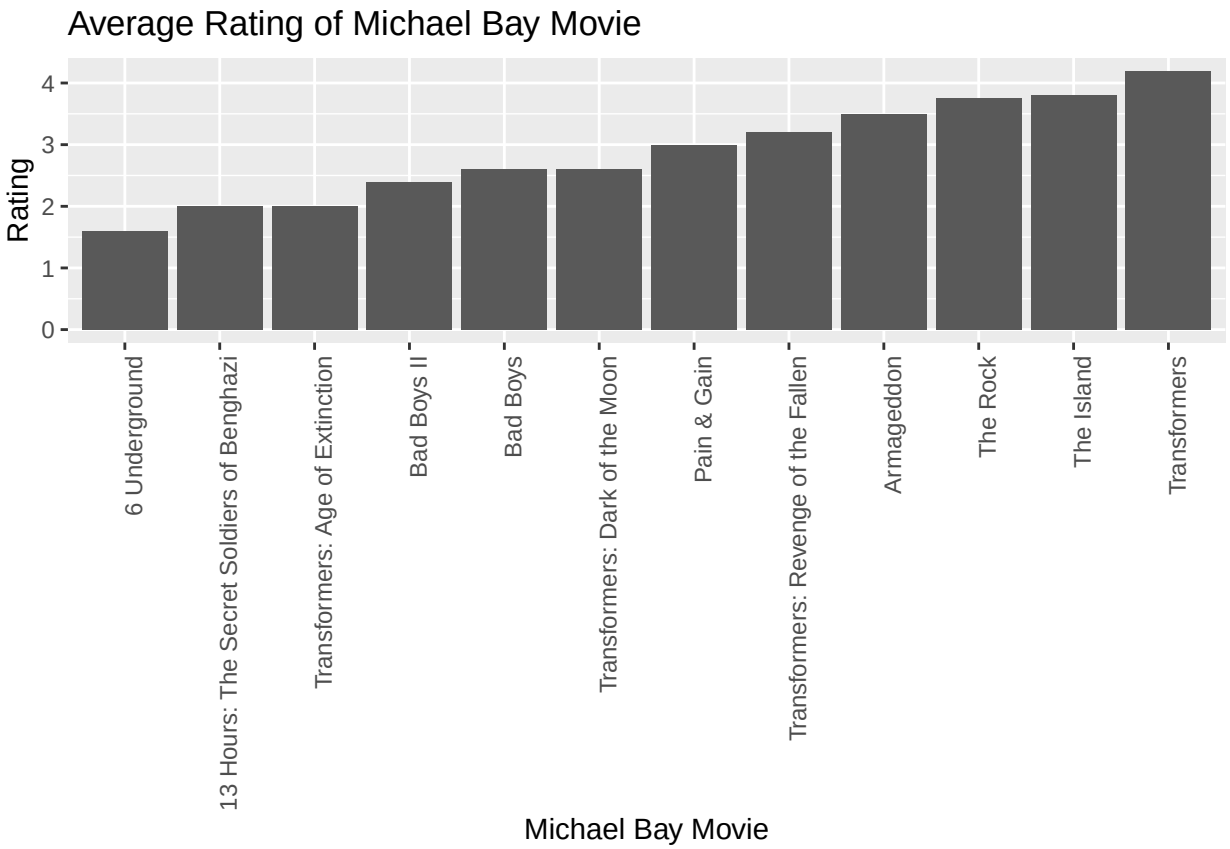
```
## Warning: Closing open result set, pending rows
```

```
rating = dbFetch(c)
dbClearResult(c)

library(ggplot2)

ggplot(data=rating, aes(x=reorder(Movie, Rating, FUN=median), y=Rating)) + geom_boxplot() + theme(axis.t
```




We can see that Transformers is clearly the highest rated Micheal Bay movie based on my survey of 5 people. But a boxplot may not have a clear answer shown by the movie Transformers: Age of Extinction. We can see that the spread of rating on this movie is quite large but because we are using a box plot and ordering it by the median the movie got a low rating. A box plot may be more appropriate when we have more data but until then using a average might be a more accurate representation.

```
ggplot(data=rating, aes(x=reorder(Movie, Rating, fun='mean'), y=Rating)) + geom_bar(fun='mean', stat='su
```

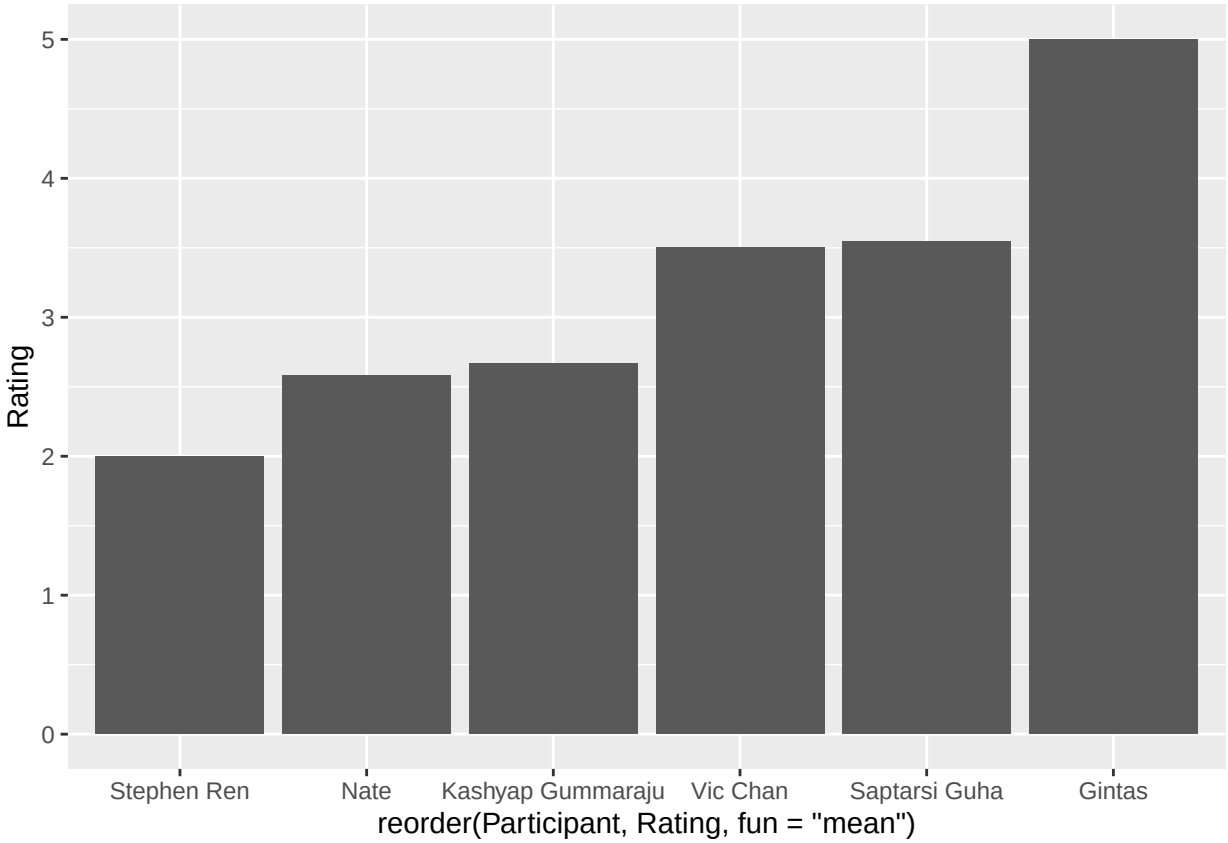



Using the average rating we can see that Transformers is still the highest rated movie therefore based on the data I would recommend one to watch transformers. (I would also still recommend watching Armageddon)

## Participant Rating Bias

Another thing to see is if a participant has any rating Bias. Everyone has a different way of rating and some people might say 3 stars is average while other people will say that 3 stars is terrible movie.

```
ggplot(data=rating, aes(x=reorder(Participant, Rating, fun='mean'), y=Rating)) + geom_bar(fun='mean', s
```




```
rating %>%
  filter(Participant != 'Gintas') %>%
  group_by(Participant) %>%
  summarise(avg_rating = mean(Rating))
```

```
## # A tibble: 5 x 2
##   Participant       avg_rating
## * <chr>                  <dbl>
## 1 Kashyap Gummaraju       2.67
## 2 Nate                    2.58
## 3 Saptarsi Guha           3.55
## 4 Stephen Ren             2
## 5 Vic Chan                3.5
```

I would recommend taking out Gintas as a participant since he has only watched one Michael Bay movie. We can also see 3/4 of the participant have ratings less than 3.

## Conclusion

If I were to do this survey again I would let my participants know that rating all of the movies is optional. The participants would rate a movie 1 star even if they have never seen it and did not know that skipping was a option. Another improvement that I would do to the survey is to create a more standardized rating system where 3 star would be average. In my survey I allowed the participants to create their own system to

rate out of 5 stars and that can make certain participants outliers. Based on the data I would recommend everyone to watch Transformers.