WHAT FACTORS INFLUENCE NET REVENUE AND RETURN ON INVESTMENT OF MOVIES AT THE BOX OFFICE?

Frederick McCollum
March 16th, 2019
BANA 6380 – Advanced SAS Programming
University of Dallas

TABLE OF CONTENTS

INTRODUCTION

As an avid movie-goer, I have a never-ending interest in all available data surrounding movies. Therefore, when given the opportunity to perform an exploratory analysis of a topic of my choosing for my Advanced SAS Programming course at the *University of Dallas*, I chose to concentrate my analysis on movies. More specifically, I would like to gain a better understanding of the following driving question: *What factors influence net revenue and return on investment of movies at the box office?*

This information is not only interesting to movie-buffs such as myself, but it is also very valuable to cinema-focused investors. Knowing what factors influence a movie's financial performance allows investors to make data-driven decisions when determining where their money is best invested. Net revenue and return on investment are two of the most common indicators of success for a financial investment. Therefore, this analysis will focus on the performance of these two measurables. I have included a definition of how these financial metrics relate to movies below, for reference:

$$Net\ Return = Revenue - Budget \qquad Return\ on\ Investment = \frac{Net\ Return}{Budget}$$

It should be noted that the dataset I will be analyzing is not fully up-to date. Therefore, I have outlined some requirements to filter out any inconsistent data. This analysis will mainly focus only on movies which meet the following criteria:

- Produced in the United States
- Released after the year 1999 *
- Budget greater than $1,000
- Revenue greater than $1,000

** Some reports will show a timeline for years prior to 1999. However, other criteria will still be met. A report will note if one of the above criteria are not fulfilled in the report.*

NOTE: The most recent movie recorded in this filtered dataset was released in 2017.

These criteria are set in order to ensure that this analysis is 1) using recent movie financial information, 2) focused only on movies which have valid financial data reported, and 3) focused on movies produced in the target country (United States of America). Also, all SAS scripts and source code used in this analysis may be found in the Appendix of this report.

DATA SOURCES

Acknowledgements

The data used in this analysis comes from *Kaggle.com* user *Rounak Banik*. The data set is named "The Movies Dataset," and is available as downloadable .csv files on the *Kaggle.com* website under the *Creative Commons Public Domain* license (*CC0: Public Domain*). This data was collected and compiled from the following sources:

- *The Movie DB* website public API
- *GroupLens* MovieLens Dataset

While this analysis makes use of data which was captured from *Kaggle.com*, *The Movie DB*, and *GroupLens* websites, in no way is this report endorsed or sponsored by *Kaggle*.com, *The Movie DB* (*TMDb*), or *GroupLens*.

Data Description

Prior to importing this data into SAS, some changes were made to the initial dataset .csv files which were downloaded from *Kaggle.*com. These changes were made in order to better suit this analysis. Some examples of the changes made are as follows:

- Duplicate records were removed from tables
- Tables were separated into more specific tables with less variables (columns)
- New columns were created to show calculated values

Once the above changes were made, the data was imported into SAS OnDemand for Academics (please reference the Appendix of this report for a full script reference). The imported data used in this analysis is structured as follows:

Table Name	Description
movies_details	Basic details of all movies included in the dataset.
movies_descriptions	Genres of all movies included in the dataset.
movies_production	Production details of all movies included in the dataset.
movies_financials	Financial details of all movies included in the dataset.
movies_production_us	Production details of movies in the data set which were produced in the US.
financials_filtered	Financial details of movies in the dataset which have budget and revenue > $1,000.

A more detailed representation of the data used in this analysis may be found in the Appendix of this report.

METHODOLOGY AND EXPLORATORY DATA ANALYSIS

Methodology Overview

Please find a brief overview of the operations and processes which were performed as part of this analysis below. A more in-depth discussion will be provided in the next section, “Data Analysis.” Additionally, all SAS queries and source code used in this analysis may be found in the Appendix.

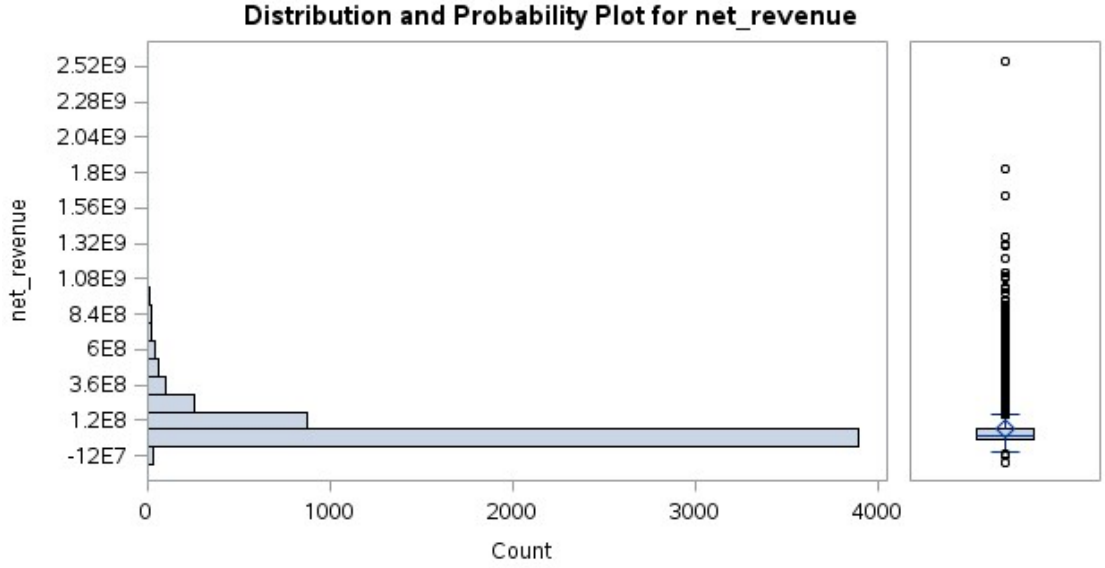
1. Remove duplicates, and clean data in raw .csv (comma separated values) file format.
2. Import data files into SAS.
3. Review top-level details data (proc contents, describe table, etc.).
4. Identify factors to further investigate how they influence target variables.
5. Create reports and analyze impact of chosen factors on target variables.

Please note that while the above is an overview of the main methodology used in this analysis, there were some additional operations performed in order to meet minimum course requirements for this project. These additional operations may not have been fully necessary for the analysis, or there may have been a simpler way to accomplish the end goal. However, the

scripts were written to demonstrate certain skills, and this will be noted with the script in the Appendix to ensure transparency.

Exploratory Data Analysis

After importing the dataset into SAS OnDemand for Academics, the first step is to review top-level descriptive statistics for the data. Since we will be focusing on how certain factors affect the financial performance of movies, reviewing the distribution of the financial performance is very important. Below, we can see some distribution plots for the net revenue variable in the "financials_filtered" table:



We can see that the distribution of this variable appears to be skewed, and the box plot seems to show many statistical outliers in this data set. This causes the mean to be much larger than the median. Due to this skewness and large number of outliers, I will focus on the median performance of the net revenue variable, rather than the mean. This will help our analysis produce a more accurate, central representation of the data. This will also help prevent any outliers from significantly impacting the results of our analysis in a negative way.

The next step is to identify what independent variables should be investigated further. Based on the dataset available, I chose to focus my exploratory analysis on the following variables:

- Budget
- Runtime
- Genre
- Production company

Therefore, moving forward I will be exploring the above factors, and how they affect median financial return (US dollars) and median return on investment (percentage of US dollars). Each variable and the respective effect on financial performance will be analyzed separately.

Budget

The following report divides movies into different categories depending on their median budget. The report then displays financial details for each of these different budget categories in US Dollars:

Figure 1: Financial Performance by Budget

Budget	NumMovies	MedianBudget	MedianRev	MedianNetRev	MedianRoi
1: Less than 25 million	1100	$10,500,000.00	$15,004,080.50	$3,626,444.00	67.03%
2: Between 25 and 50 million	645	$33,000,000.00	$63,782,078.00	$31,255,921.00	95.29%
3: Between 50 and 75 million	331	$60,000,000.00	$104,478,416.00	$45,178,561.00	78.57%
4: Between 75 and 100 million	225	$85,000,000.00	$186,072,214.00	$101,001,478.00	119.0%
5: Greater than 100 million	305	$150,000,000.00	$407,602,906.00	$256,372,926.00	177.8%

Runtime

The following report divides movies into different categories depending on their runtime in minutes. The report then displays financial details for each of these different runtime categories in US Dollars:

Figure 2: Financial Performance by Runtime

Runtime	NumMovies	MedianBudget	MedianRev	MedianNetRev	MedianRoi
1: Less than 90 minutes	296	$18,000,000.00	$34,676,714.00	$11,649,535.00	99.99%
2: Between 90 and 120 Minutes	1775	$25,000,000.00	$47,801,389.00	$18,596,852.00	86.61%
3: Between 120 and 150 Minutes	520	$50,000,000.00	$114,408,507.00	$60,249,965.00	135.6%
4: Between 150 and 180 Minutes	56	$108,500,000.00	$362,506,637.50	$197,249,383.50	224.2%
5: Greater than 180 Minutes	5	$100,000,000.00	$449,220,945.00	$309,220,945.00	220.9%

Genre

For the purpose of this analysis, I chose to focus on the ten most common genres in the dataset. The following list includes the ten genres which were examined in this analysis.

- Comedy
- Action
- Drama
- Music
- Horror
- Science Fiction
- Animation
- Adventure
- Thriller
- Romance

Please note, while this analysis is focused on the performance of the above ten genres, the overall median was still calculated using all genres. Additionally, movies in the dataset can have more than one genre associated with them. Therefore, the genre variable is not mutually exclusive for each movie.

The following report compares each genre's median financial results with the overall median financial results:

Figure 3: Movie Genres' Financial Statistics Compared to Median

Action, Adventure, Animation, Comedy, Drama, Horror, Music, Romance, Science Fiction, Thriller, MEDIAN

Genre	NumMovies	MedianBudget	MedianRevenue	MedianNetRevenue	MedianRoiPercent
Action	702	$60,000,000.00	$100,795,298.00	$48,038,798.00	96.89%
Adventure	496	$85,000,000.00	$182,698,900.00	$91,944,083.50	125.8%
Animation	171	$80,000,000.00	$242,988,466.00	$140,875,730.00	177.8%
Comedy	933	$30,000,000.00	$59,827,328.00	$25,696,770.00	106.7%
Drama	1171	$21,000,000.00	$34,077,920.00	$7,618,727.00	56.68%
Horror	282	$13,000,000.00	$42,230,738.00	$25,042,188.00	137.6%
Music	87	$20,000,000.00	$47,126,295.00	$21,935,319.00	153.1%
Romance	440	$25,000,000.00	$43,347,102.50	$17,587,688.50	101.2%
Science Fiction	309	$65,000,000.00	$109,906,372.00	$54,478,416.00	118.0%
Thriller	776	$30,000,000.00	$56,334,873.00	$20,080,660.00	72.91%
MEDIAN	.	$30,000,000.00	$58,231,840.00	$24,461,547.00	100.2%

Next, I have taken the top five genres for both median net revenue and median return on investment:

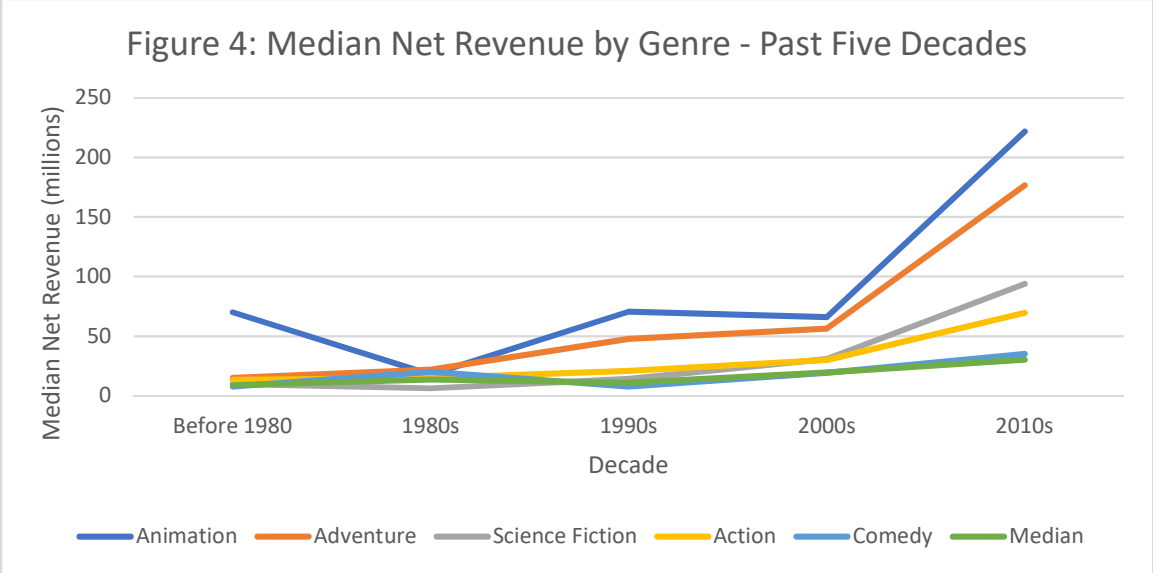
Top Median Net Revenue:

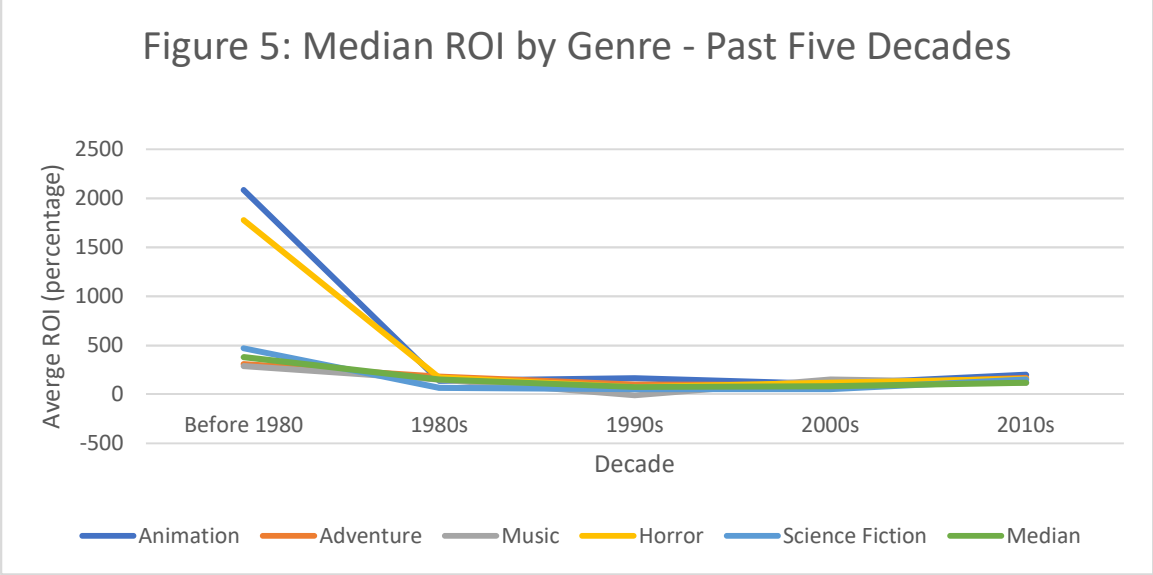
1. Animation
2. Adventure
3. Science Fiction
4. Action
5. Comedy

Top Median Return on Investment:

1. Animation
2. Music
3. Horror
4. Adventure
5. Science Fiction

Using the above two lists, I have compared them in a timeline with the overall median over the past five decades. Please find an associated timeline graph for median net revenue and median return on investment below:





Production Company

As with genres, I chose to focus only on a subset of the total production companies. For the purpose of this analysis, I chose to examine the following five popular movie production companies:

- Universal Pictures
- Paramount Pictures
- Walt Disney
- Twentieth Century Fox
- Warner Bros.

Please note, while this analysis is focused on the performance of the above five production companies, the overall median was still calculated using all production companies. Additionally, movies in the dataset can have more than one production company associated with them. Therefore, the production company variable is not mutually exclusive for each movie.

The following report compares each production company's median financial results with the overall median:

Figure 6: Movie Production Companys' Financial Statistics Compared to Median

Paramount Pictures, Twentieth Century Fox, Universal Pictures, Walt Disney, Warner Bros, MEDIAN

ProdCompany	NumMovies	MedianBudget	MedianRevenue	MedianNetRevenue	MedianRoiPercent
Paramount Pictures	157	$60,000,000.00	$114,501,299.00	$61,175,291.00	132.1%
Twentieth Century Fox	154	$58,000,000.00	$167,763,888.00	$99,636,238.50	164.9%
Universal Pictures	202	$50,000,000.00	$119,470,612.50	$65,023,671.00	164.1%
Walt Disney	108	$102500000.00	$197,604,023.50	$122,889,571.00	176.5%
Warner Bros	219	$60,000,000.00	$106,371,651.00	$46,234,523.00	98.28%
MEDIAN	.	$30,000,000.00	$58,231,840.00	$24,461,547.00	100.2%

Next, I have sorted these production companies for both median net revenue and median return on investment:

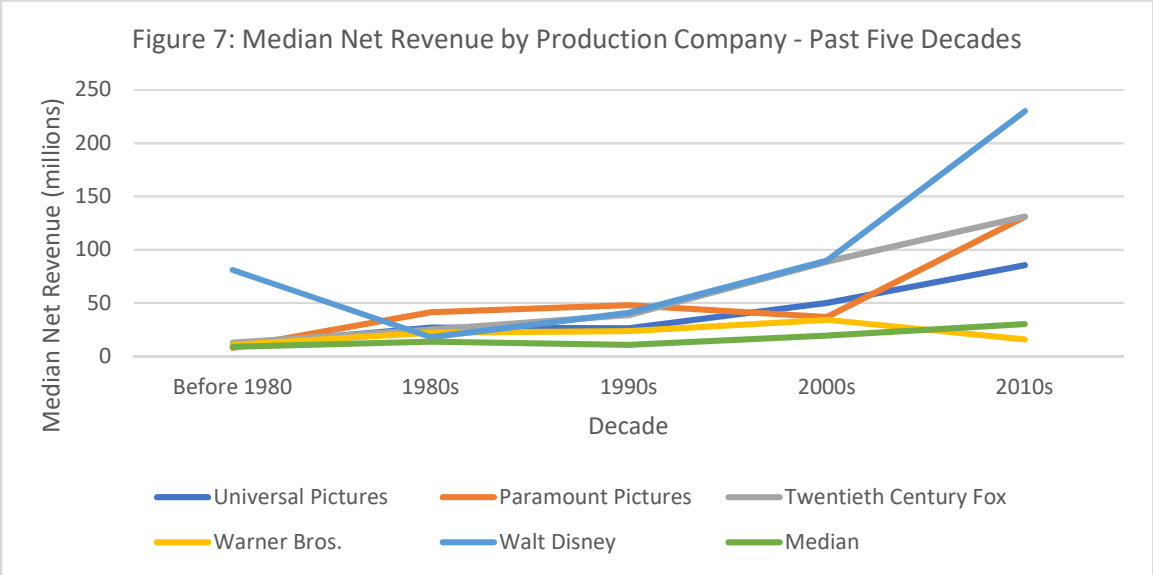
Top Median Net Revenue:

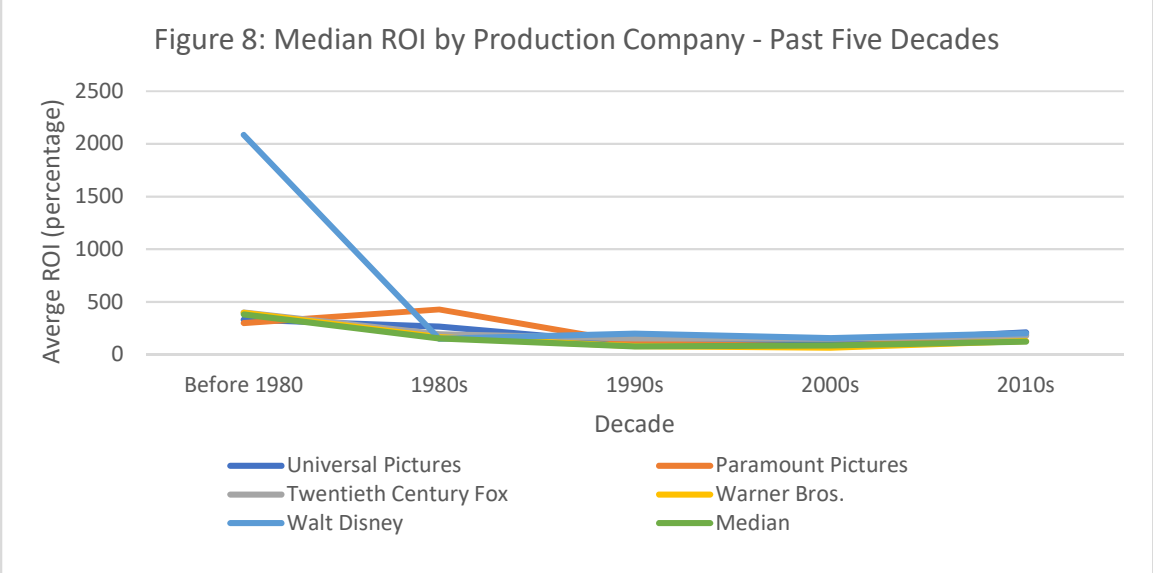
1. Walt Disney
2. Twentieth Century Fox
3. Universal Pictures
4. Paramount Pictures
5. Warner Bros.

Top Median Return on Investment:

1. Walt Disney
2. Twentieth Century Fox
3. Universal Pictures
4. Paramount Pictures
5. Warner Bros.

Using the above two lists, I have compared them in a timeline with the overall median over the past five decades. Please find an associated timeline graph for median net revenue and median return on investment below:





CONCLUSION

Results Analysis

In conclusion, this dataset has provided some valuable insight into how certain factors affect the financial performance of movies. Specifically, one can see how median net revenue and median return on investment vary (or do not vary) depending on target dependent variables. Having performed the above exploratory data analysis, there are some data-driven conclusions which I am able to draw regarding each of these four investigated factors. I will break down these findings individually by variable below, including recommendations for further analysis where possible.

Budget

Based on the "Financial Performance by Budget" report, it seems as though a higher movie budget is positively correlated with a higher net revenue. This does not come off as very surprising because higher budgeted movies tend to have larger marketing campaigns, better special effects, etc., and thus generate more potential viewers. What is surprising is the rate at which the net revenue increases. Also, it is interesting to see that a higher movie budget also appears to have a positive correlation with a higher return on investment. When budget increases, it means that a movie must make an exponentially greater amount in revenue in order to retain or increase its return on investment.

There is not much opportunity to analyze this variable further. We could use a statistical test, such as a linear regression analysis, to see if the correlation between budget and net revenue is significant. However, even if this proven to be significant, it is likely not considered new information. Additionally, with higher risk comes higher reward. Therefore, it is also somewhat expected for higher budgeted movies to experience a higher return on investment.

Runtime

Similar to budget levels, it appears as though a higher runtime has a positive correlation with both net revenue and return on investment. Although it should be noted that there are only five movies recorded in this dataset with a runtime greater than three hours. Therefore, this may not be enough data to draw a conclusion on movies of that runtime level. However, the other levels have much higher record counts, and they all show a positive increase in financial results when runtime increases.

Most notably, there is over a one hundred percent increase in return on investment when comparing movies that are less than two hours to movies that are over two and a half hours. In the same comparison, we also see an increase of almost two hundred million dollars in net revenue. This is an area which could warrant further investigation in order to determine if these results are significant. I would be interested to know if a statistical test, such as a linear regression analysis, indicates that there is a significant correlation between the runtime of a movie and its financial performance as measured in this analysis. This would be valuable to know as an investor, as longer movies would seem to be a better investment based on these results.

Genre

There are six genres which have a median net revenue above the overall median value. However, there is a large drop off in median net revenue after the top two genres: "Animation" and "Adventure." Interestingly, the "Music" genre is below the overall median in net revenue, yet it is second to "Animation" in return on investment. "Animation" is the clear leader in both net revenue and return on investment, and it seems to be the best choice for investment based on genre. Based on this outcome, it seems as though the "Animation" genre is the best choice for investment if you are searching solely on genre for a successful financial return. This is likely due to the lower cost to produce animated movies. Additionally, as technology improves, those animated movies can be made even more efficiently. "Adventure" has a strong argument for second, along with "Horror" and "Music" as well.

Regarding the two timeline analyses of the genre variable, "Animation" and "Adventure" movies appear to be experiencing a more significant positive trend in median net revenue over the past couple decades. As mentioned, "Animation" is likely experiencing the highest increase due to the improvements in technology recently. As technology improves, those animated movies can be made even more efficiently. The "Adventure" and "Science Fiction" genres have a slightly lesser upward trend for median net revenue (possibly also affected by improvements in technology), while "Comedy" movies have remained close to the overall median. However, this is not reciprocated when analyzing the graph for median return on investment. All genres appear to be following the same general trend for median return on investment, and it is very close to the overall median.

There is much more room for further analysis with this factor. For example, a single-sample t-test would be very valuable to understand how many of these results are statistically significant in comparison to the overall median financial results. While the findings in this analysis appear to be definitive, a statistical test for significance would be needed for increased confidence in the results.

Production Company

The report "Movie Production Companies' Financial Statistics Compared to Median" report shows that *Walt Disney* and *Twentieth Century Fox* are the top two production companies in both median net revenue and median return on investment. *Walt Disney* is far and away the highest when it comes to median net revenue, and it is the only production company with a median net revenue above one hundred million. It appears that this does come at a cost, as *Walt Disney* is also far and away the highest when it comes to median budget. Once again, *Walt Disney* is the only production company with a median budget over one hundred million. However, this does not seem to affect the company's return on investment, as *Walt Disney* is still a clear leader in median return on investment. Although it is interesting to see *Universal Pictures* make a big jump in median return on investment, considering their lower median net revenue.

This data suggests that *Walt Disney* is the best choice for movie investments based on production company. *Twentieth Century Fox* is closer to *Universal Pictures* in third place than either of them is to the top spot. We can see that *Walt Disney* has the most significant increase in median net revenue over time, as well. This is an indicator that the company has made smart investments over time, and thus suggests this company is a good choice for external investors.

As with the genre variable, I would like to understand how these results hold up to a statistical significance test. A single sample t-test to assess how these results compare to the overall median would help improve confidence in this analysis.

Additional Notes

While the above exploratory analysis is detailed, it should be reiterated that some of the above variables are not mutually exclusive for movie records. For example, one movie record can belong to multiple genres, as well as belong to multiple production companies. I can make suggestions or recommendations stating that a statistical test for significance would help improve the confidence level of these results. Yet, it will likely be more difficult to perform such a test for statistical significance considering these observations are not fully independent of each other for all variables.

However, having the possibility of multiple genres per movie record is not all bad. This does present the possibility of performing an analysis using combinations of genres. The same can be performed using combinations of other variables in the dataset, as well. With such a large number of combinations, there are near limitless ways to analyze this data further. The analysis performed in this report is detailed, but there is always room for further, deeper investigation.

APPENDIX

Data Description

A more detailed representation of the data used in this analysis may be found below:

Table Name	Description	# Rows	Columns
movies_details	Basic details of all movies included in the dataset.	45433	id (pk) – unique id for movie record title – movie title original_language – original movie language original_title – original movie title runtime – movie runtime in minutes spoken_languages – languages spoken in movie (json formatted list)
movies_descriptions	Genres of all movies included in the dataset.	45433	id (pk) – unique id for movie record adult – true/false indicator of an “adult” movie genres – json formatted list of genres for the movie
movies_production	Production details of all movies included in the dataset.	45433	id (pk) – unique id for movie record production_companies – json formatted list of companies that produced the movie production_countries – json formatted list of countries the movie is produced in release_date – date the movie was released status – released or not
movies_financials	Financial details of all movies included in the dataset.	45433	id (pk) – unique id for movie record budget – movie budget in us dollars revenue – movie revenue in us dollars net revenue – net revenue in us dollars
movies_production_us	Production details of movies which were produced in the US.	9798	id (pk) – unique id for movie record production_companies – json formatted list of companies that produced the movie production_countries – json formatted list of countries the movie is produced in release_date – date the movie was released status – released or not
financials_filtered	Financial details of movies which have budget and revenue > $1,000.	5307	id (pk) – unique id for movie record budget – movie budget in us dollars revenue – movie revenue in us dollars net revenue – net revenue in us dollars

SAS Source Code

All SAS source code used during this analysis may be found in the following sections. All SAS scripts were run using SAS OnDemand for Academics. Source code has been broken up into separate scripts, with accompanying descriptions. Each script has been separated into their respective methodology sections depending on the general order in which the scripts were run. Please find comments indicated in green font located between two "/*" brackets. Additionally, comments with numbers in them indicate specific requirements that needed to be met for the purpose of this course project.

Remove duplicates, clean data in raw .csv (comma separated values) file format.
These SQL scripts were run external to SAS using an SQLite database.

Script: Remove duplicate values.
This is the SQL script that was used to remove duplicate values from the "movies_details" table. This script was also edited to remove duplicate values from other tables in a similar way. Please note, this is the only script featured which did not utilize SAS.

```
--Remove duplicate records
delete from movies_details
where id in (
    select id
    from movies_details
    where id in (
        select id
        from movies_details
        group by id
        having count(id) > 1)
    group by id)
    AND popularity in (
        select min(popularity)
        from movies_details
        where id in (
            select id
            from movies_details
            group by id
            having count(id) > 1)
        group by id);
```

Import data files into SAS.
The following scripts correspond to the actions performed in the "Import data files into SAS" section of the "Methodology Overview."

Snippet: Set project library name.
This snippet sets the project library name, allowing all future scripts to reference "proj" as the libname in SAS.

```
/*Set libname*/
%let path=/home/fmccollum1/MacroSqlProject;
libname proj "&path";
```

Script: Import data.

The following script was used to import all .csv raw data files into SAS OnDemand for Academics as SAS datasets.

```
/*Import movie_details.csv*/
data proj.movies_details;
      infile
'/home/fmccollum1/MacroSqlProject/RawCsvFiles/movie_details.csv'
            dlm=',' dsd missover firstobs=2;
      input id title :$120. original_language :$5. original_title
:$120. runtime
          spoken_languages :$800.;
run;

/*Import movie_descriptions.csv*/
data proj.movies_descriptions;
      infile
'/home/fmccollum1/MacroSqlProject/RawCsvFiles/movie_descriptions.
csv'
            dlm=',' dsd missover firstobs=2;
      input id adult :$8. genres :$275.;
run;

/*Import movie_financials.csv*/
data proj.movies_financials;
      infile
'/home/fmccollum1/MacroSqlProject/RawCsvFiles/movie_financials.cs
v'
            dlm=',' dsd missover firstobs=2;
      input id budget :10. revenue :10. net_revenue :10.;
run;

/*Import movie_production.csv*/
data proj.movies_production;
      infile
'/home/fmccollum1/MacroSqlProject/RawCsvFiles/movie_production.cs
v'
            dsd missover firstobs=2;
      input  id  production_companies  :$1300.  production_countries
:$1250.
            release_date :YYMMDD10. status :$15.;
run;

/*Import movie_ratings.csv*/
data proj.movies_ratings;
```

```
      infile
'/home/fmccollum1/MacroSqlProject/RawCsvFiles/movie_ratings.csv'
            dlm=',' dsd missover firstobs=2;
      input id popularity :10. vote_average vote_count;
run;
```

Script: Drop movies_ratings table.

This SAS script was used to drop the "movies_ratings" table, as it was not necessary for the analysis of this project. Please note, I understand that it would not be necessary to import the table if it were not used. However, this was still performed in order to demonstrate how to drop a table for the purposes of this project.

```
/*Skills used: #20*/
/*Drop movies_ratings table*/
proc sql;
      drop table proj.movies_ratings;
quit;
```

Script: Create index on id column.

The following SAS script was used to create an index on the id column in all data sets. This helps to improve query performance and efficiency when querying tables.

```
/*Skills used: #19*/
/*Create index on id column*/
proc sql;
      create index id
      on proj.movies_details(id);

      create index id
      on proj.movies_financials(id);

      create index id
      on proj.movies_production(id);

      create index id
      on proj.movies_descriptions(id);
quit;
```

Script: Confirm there are no duplicate records.

The following SAS script was used to confirm that there are no duplicate records in the "movies_details" table.

```
/*Skills used: #1, #3, #9, #10, #11*/
/*Validate no duplicate records*/
proc sql;
      select count(id) as Count
      from proj.movies_details
```

```
      group by id
      having count(id) > 1;
quit;
```

Script: Modify release_date column format.

This script modifies the "release_date" column in the "movies_production" table to format the date values as a SAS DATE9 format.

```
/*Skills used: #18*/
/*Modify release_date format*/
proc sql;
   alter table proj.movies_production
      modify release_date format=date9.;
   select * from proj.movies_production;
quit;
```

Script: Create United States movies table.

This script creates the "movies_production_us" table, which only includes production details for those movies that were produced in the United States after the year 1999.

```
/*Skills used: #1, #5*/
/*Create US movies table*/
proc sql;
      create table proj.movies_production_us as
      select *
      from proj.movies_production
      where production_countries contains "US"
            and year(release_date) > 1999 and status = "Released";
quit;
```

Script: Create financials_filtered table.

This script creates the "financials_filtered" table, which includes only those movies that have over $1,000 in budget, and over $1,000 in revenue.

```
/*Skills used: #1, #5, #17*/
/*Create financials_filtered table*/
proc sql;
      create table proj.financials_filtered as
      select *
      from proj.movies_financials;

      delete from proj.financials_filtered
      where budget < 1000 or revenue < 1000;
quit;
```

Script: Create financials_filtered table (alternate).

This script is an alternate way to create the "financials_filtered" table. This script uses a "data" step, rather than using the "proc sql" procedure, as the previous script did. This script was used to fulfill a requirement of this project.

```
/*Skills used: #17*/
/*Create financials_filtered table using data step*/
data proj.financials_filtered:
      set proj.movies_financials:
      if budget < 1000 or revenue < 1000 then delete;
run;
```

Review top-level details of data (proc contents, describe table, etc.).
The following scripts correspond to the actions performed in the "Review top-level details of data" section of the "Methodology Overview."

Script: Review net revenue distribution.
This script produces graphs that show distribution information for the "net_revenue" variable in the "financials_filtered" table.

```
/*Check distribution of net revenue*/
ods graphics on;
proc univariate data=proj.financials_filtered plot;
      var net_revenue budget revenue;
run;
```

Script: Print table details.
This script outputs top level table details such as column and row counts, as well as variable types and lengths. Any table name can be substituted in the "proc contents" procedure to view details for that table. The "RowColNum" macro takes a dataset and writes to the log the number of columns and rows, as well as shows the DESCRIBE TABLE details for that table. Please note, this is information which can be found using simpler methods. However, this macro was written to demonstrate writing a macro that takes parameters for this project.

```
/*Check proc contents*/
title "Proc Contents for movies_details Table";
proc contents data=proj.movies_details;
run;
title;

/*Skills used: #22, #23, #24*/
/*Set global libname variable*/
%global lib;
%let lib = PROJ;
```

```
/*Skills used: #1, #2*/
/*Select libname table details from dictionary tables.*/
title "&lib Library Table Details";
proc sql;
      select  memname "Table",
            nobs "Rows",
            nvar "Columns",
            filesize "File Size",
            maxvar 'Widest Column',
            maxlabel 'Widest Label'
      from dictionary.tables
      where libname = "&lib";
quit;
title;

/*Skills used: #21, #33, #34, #37, #39, #40, #41, #42, #43*/
/*Create macro to check row and column counts in a dataset
This script outputs details to the log*/
options mlogic mprint symbolgen mcompilenote=all;
%macro RowColNum(dS, checkDescribe);
      %local NumCol;
      %local NumRow;

      %let dSiD=%sysfunc(open(&dS));
            %let NumCol = %sysfunc(attrn(&dSiD,nvars));
            %let NumRow = %sysfunc(attrn(&dSiD,nobs)) ;
            %put NOTE: There are &NumCol columns, and &NumRow rows
in this dataset.;
            %put NOTE: More than 5000 rows? %eval(&NumRow gt 5000);
      %let dSiD=%sysfunc(close(&dSiD));

      %if &checkDescribe=Y %then %do;
            proc sql;
                  describe table &dS;
            quit;
      %end;
      %else %do;
            %put NOTE: You chose not to view describe output for
this table.;
      %end;
%mend RowColNum;

%RowColNum(proj.movies_details, Y);
```

Identify factors to further investigate how they influence target variables.
No scripts were used in this section of the methodology for this analysis.

Create reports and analyze impact of chosen factors on target variables.

The following scripts correspond to the actions performed in the "Create reports and analyze impact of chosen factors on target variables" section of the "Methodology Overview."

Script: Check median financial values for all target movies.

This script produces a report showing median financial values for all target movies. This means all movies that meet the pre-defined criteria of this analysis.

```
/*Skills used: #3*/
/*Median financial values for all target movies*/
title "Median Financial Values for All Target Movies";
proc sql;
      select count(d.id) as NumMovies, median(f.budget) as
MedianBudget format=dollar14.2,
            median(f.revenue) as MedianRevenue format=dollar14.2,
            median(f.net_revenue) as MedianNetRevenue
format=dollar14.2,
            median(f.net_revenue/f.budget) as MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id;
quit;
title;
```

Script: Figure 1: Financial performance by budget.

This script creates a report detailing movies' financial performance by budget.

```
/*Skills used: #7, #8, #9*/
/*Financial Performance by budget*/
title "Figure 1: Financial Performance by Budget";
proc sql;
      select "1: Less than 25 million" as Budget, count(f.id) as
NumMovies,
            median(f.budget) as MedianBudget format=dollar16.2,
            median(f.revenue) as MedianRev format=dollar16.2,
            median(f.net_revenue) as MedianNetRev
format=dollar16.2,
            median(f.net_revenue/f.budget) as MedianRoi
format=percent8.2
      from proj.financials_filtered f
      inner join proj.movies_details d on f.id = d.id
      inner join proj.movies_production_us p on f.id = p.id
      where year(p.release_date) > 1999 and f.budget < 25000000
      UNION
      select "2: Between 25 and 50 million" as Budget, count(f.id)
as NumMovies,
            median(f.budget) as MedianBudget format=dollar16.2,
            median(f.revenue) as MedianRev format=dollar16.2,
```

```
          median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
          median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
     from proj.financials_filtered f
     inner join proj.movies_details d on f.id = d.id
     inner join proj.movies_production_us p on f.id = p.id
     where f.budget >= 25000000 and f.budget < 50000000
     UNION
     select "3: Between 50 and 75 million" as Budget, count(f.id)
as NumMovies,
          median(f.budget) as MedianBudget format=dollar16.2,
          median(f.revenue) as MedianRev format=dollar16.2,
          median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
          median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
     from proj.financials_filtered f
     inner join proj.movies_details d on f.id = d.id
     inner join proj.movies_production_us p on f.id = p.id
     where f.budget >= 50000000 and f.budget < 75000000
     UNION
     select "4: Between 75 and 100 million" as Budget, count(f.id)
as NumMovies,
          median(f.budget) as MedianBudget format=dollar16.2,
          median(f.revenue) as MedianRev format=dollar16.2,
          median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
          median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
     from proj.financials_filtered f
     inner join proj.movies_details d on f.id = d.id
     inner join proj.movies_production_us p on f.id = p.id
     where f.budget >= 75000000 and f.budget <= 100000000
     UNION
     select "5: Greater than 100 million" as Budget, count(f.id)
as NumMovies,
          median(f.budget) as MedianBudget format=dollar16.2,
          median(f.revenue) as MedianRev format=dollar16.2,
          median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
          median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
     from proj.financials_filtered f
     inner join proj.movies_details d on f.id = d.id
     inner join proj.movies_production_us p on f.id = p.id
     where f.budget > 100000000;
quit;
title;
```

Script: Figure 2: Financial performance by runtime.

This script creates a report detailing movies' financial performance by runtime.

```
/*Skills used: #7, #8, #9*/
/*Financials by Runtime*/
title "Figure 2: Financial Performance by Runtime";
proc sql;
      select "1: Less than 90 minutes" as Runtime, count(f.id) as
NumMovies,
            median(f.budget) as MedianBudget format=dollar16.2,
            median(f.revenue) as MedianRev format=dollar16.2,
            median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
            median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
      from proj.financials_filtered f
      inner join proj.movies_details d on f.id = d.id
      inner join proj.movies_production_us p on f.id = p.id
      where d.runtime < 90
      UNION
      select "2: Between 90 and 120 Minutes" as Runtime, count(f.id)
as NumMovies,
            median(f.budget) as MedianBudget format=dollar16.2,
            median(f.revenue) as MedianRev format=dollar16.2,
            median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
            median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
      from proj.financials_filtered f
      inner join proj.movies_details d on f.id = d.id
      inner join proj.movies_production_us p on f.id = p.id
      where d.runtime >= 90 and d.runtime < 121
      UNION
      select  "3:  Between  120  and  150  Minutes"  as  Runtime,
count(f.id) as NumMovies,
            median(f.budget) as MedianBudget format=dollar16.2,
            median(f.revenue) as MedianRev format=dollar16.2,
            median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
            median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
      from proj.financials_filtered f
      inner join proj.movies_details d on f.id = d.id
      inner join proj.movies_production_us p on f.id = p.id
      where d.runtime >= 120 and d.runtime < 151
      UNION
      select  "4:  Between  150  and  180  Minutes"  as  Runtime,
count(f.id) as NumMovies,
            median(f.budget) as MedianBudget format=dollar16.2,
            median(f.revenue) as MedianRev format=dollar16.2,
```

```
          median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
          median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
     from proj.financials_filtered f
     inner join proj.movies_details d on f.id = d.id
     inner join proj.movies_production_us p on f.id = p.id
     where d.runtime >= 150 and d.runtime < 181
     UNION
     select "5: Greater than 180 Minutes" as Runtime, count(f.id)
as NumMovies,
          median(f.budget) as MedianBudget format=dollar16.2,
          median(f.revenue) as MedianRev format=dollar16.2,
          median(f.net_revenue)          as          MedianNetRev
format=dollar16.2,
          median(f.net_revenue/f.budget)       as       MedianRoi
format=percent8.2
     from proj.financials_filtered f
     inner join proj.movies_details d on f.id = d.id
     inner join proj.movies_production_us p on f.id = p.id
     where d.runtime >= 180;
quit;
title;
```

Script: Figure 3: Create genre comparison table.

This script creates a report comparing financial performance of each movie genre. The overall median is inserted into the table as well using SAS macros.

```
/*Skills used: #4, #5*/
/*Create table genremedianroi*/
proc sql;
create table proj.GenreMedianRoi as
     select  "Comedy"  as  Genre,  count(d.id)  as  NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
          median(f.revenue) as MedianRevenue format=dollar16.2,
          median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
          median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
     from proj.movies_descriptions d
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_production_us p on d.id = p.id
     where d.genres contains "Comedy"
     UNION
     select  "Action"  as  Genre,  count(d.id)  as  NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
          median(f.revenue) as MedianRevenue format=dollar16.2,
          median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
```

```
          median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
     from proj.movies_descriptions d
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_production_us p on d.id = p.id
     where d.genres contains "Action"
     UNION
     select   "Drama"   as   Genre,   count(d.id)   as   NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
          median(f.revenue) as MedianRevenue format=dollar16.2,
          median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
          median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
     from proj.movies_descriptions d
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_production_us p on d.id = p.id
     where d.genres contains "Drama"
     UNION
     select   "Music"   as   Genre,   count(d.id)   as   NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
          median(f.revenue) as MedianRevenue format=dollar16.2,
          median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
          median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
     from proj.movies_descriptions d
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_production_us p on d.id = p.id
     where d.genres contains "Music"
     UNION
     select "Science Fiction" as Genre, count(d.id) as NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
          median(f.revenue) as MedianRevenue format=dollar16.2,
          median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
          median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
     from proj.movies_descriptions d
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_production_us p on d.id = p.id
     where d.genres contains "Science Fiction"
     UNION
     select    "Animation",    count(d.id)    as    NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
          median(f.revenue) as MedianRevenue format=dollar16.2,
          median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
          median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
```

```
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where d.genres contains "Animation"
      UNION
      select "Adventure" as Genre, count(d.id) as NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
           median(f.revenue) as MedianRevenue format=dollar16.2,
           median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
           median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where d.genres contains "Adventure"
      UNION
      select "Thriller" as Genre, count(d.id) as NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
           median(f.revenue) as MedianRevenue format=dollar16.2,
           median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
           median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where d.genres contains "Thriller"
      UNION
      select "Romance" as Genre, count(d.id) as NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
           median(f.revenue) as MedianRevenue format=dollar16.2,
           median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
           median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where d.genres contains "Romance"
      UNION
      select "Horror" as Genre, count(d.id) as NumMovies,
median(f.budget) as MedianBudget format=dollar14.2,
           median(f.revenue) as MedianRevenue format=dollar16.2,
           median(f.net_revenue)       as       MedianNetRevenue
format=dollar16.2,
           median(f.net_revenue/f.budget)   as   MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
```

```
      inner join proj.movies_production_us p on d.id = p.id
      where d.genres contains "Horror";
quit;

/*Skills used: #15, #26*/
/*Use select into :macro in order to insert data into table*/
proc sql noprint;
      select median(f.budget), median(f.revenue),
            median(f.net_revenue), median(f.net_revenue/f.budget)
      into :Budget, :Revenue, :NetRevenue, :Roi
      from proj.financials_filtered f
      inner join proj.movies_production_us p on f.id = p.id;

      insert into proj.genreMedianroi
      values ("Median", null, &Budget, &Revenue, &NetRevenue,
&Roi);
quit;

/*Skills used: #16*/
/*Update to caps Median*/
proc sql noprint;
      update proj.genreMedianroi
      set Genre = %upcase("Median")
      where Genre = "Median";
quit;

/*Skills used: #35, #36, #44*/
/*View the table*/
proc sql noprint;
      select Genre
      into :Genres
      separated by ', '
      from proj.genreMedianroi;
quit;

%let title = %str(Figure 3: Movie Genres%' Financial Statistics
Compared to Median);
title "&title";
title3 "&Genres";
proc sql;
      select *
      from proj.genreMedianroi
      order by MedianNetRevenue desc;
quit;
title;
title3;
```

Script: Figure 6: Create production company comparison table.

This script creates a report comparing financial performance of each movie production company. The overall median is inserted into the table as well using SAS macros.

```
/*Skills used: #4, #5*/
/*Create table prodMedianroi*/
proc sql;
create table proj.ProdMedianRoi as
      select "Universal Pictures" as ProdCompany, count(d.id) as
NumMovies,
            median(f.budget) as MedianBudget format=dollar14.2,
            median(f.revenue) as MedianRevenue format=dollar16.2,
            median(f.net_revenue)        as        MedianNetRevenue
format=dollar16.2,
            median(f.net_revenue/f.budget)    as    MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where p.production_companies contains "Universal Pictures"
      UNION
      select "Paramount Pictures" as ProdCompany, count(d.id) as
NumMovies,
            median(f.budget) as MedianBudget format=dollar16.2,
            median(f.revenue) as MedianRevenue format=dollar16.2,
            median(f.net_revenue)        as        MedianNetRevenue
format=dollar16.2,
            median(f.net_revenue/f.budget)    as    MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where p.production_companies contains "Paramount Pictures"
      UNION
      select   "Walt   Disney"   as   ProdCompany,   count(d.id)   as
NumMovies,
            median(f.budget) as MedianBudget format=dollar14.2,
            median(f.revenue) as MedianRevenue format=dollar16.2,
            median(f.net_revenue)        as        MedianNetRevenue
format=dollar16.2,
            median(f.net_revenue/f.budget)    as    MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where p.production_companies contains "Walt Disney"
      UNION
      select "Twentieth Century Fox" as ProdCompany, count(d.id) as
NumMovies,
            median(f.budget) as MedianBudget format=dollar14.2,
            median(f.revenue) as MedianRevenue format=dollar16.2,
            median(f.net_revenue)        as        MedianNetRevenue
format=dollar16.2,
```

```
            median(f.net_revenue/f.budget)    as    MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where p.production_companies contains "Twentieth Century Fox"
      UNION
      select   "Warner   Bros"   as   ProdCompany,   count(d.id)   as
NumMovies,
            median(f.budget) as MedianBudget format=dollar14.2,
            median(f.revenue) as MedianRevenue format=dollar16.2,
            median(f.net_revenue)        as        MedianNetRevenue
format=dollar16.2,
            median(f.net_revenue/f.budget)    as    MedianRoiPercent
format=percent8.2
      from proj.movies_descriptions d
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_production_us p on d.id = p.id
      where p.production_companies contains "Warner Bros";
quit;

/*Skills used: #15, #26*/
/*Use select into :macro in order to insert data into table*/
proc sql noprint;
      select median(f.budget), median(f.revenue),
            median(f.net_revenue), median(f.net_revenue/f.budget)
      into :Budget, :Revenue, :NetRevenue, :Roi
      from proj.financials_filtered f
      inner join proj.movies_production_us p on f.id = p.id;

      insert into proj.prodMedianroi
      values  ("Median",  null,  &Budget,  &Revenue,  &NetRevenue,
&Roi);
quit;

/*Skills used: #16*/
/*Update to caps Median*/
proc sql noprint;
      update proj.prodMedianroi
      set ProdCompany = %upcase("Median")
      where ProdCompany = "Median";
quit;

/*Skills used: #35, #36, #44*/
/*View the table*/
proc sql noprint;
      select ProdCompany
      into :ProdCompanies
      separated by ', '
      from proj.prodMedianroi;
```

```
quit;

%let title = %str(Figure 6: Movie Production Companys%' Financial
Statistics Compared to Median);
title "&title";
title3 "&ProdCompanies";
proc sql;
      select *
      from proj.prodMedianroi;
quit;
title;
title3;
```

Script: Figures 4 & 5: Analyze genre financials over time.

This script is used to produce a report showing financial performance over time for one specific genre. This script was used for each genre to produce a line graph representation of financial performance over time for each genre compared to the median.

Please note, this script does not meet the requirement of being released after the year 1999.

```
/*Check financial performance of genre over time.
Set genre below. Options include:
      Comedy, Action, Drama, Music, Horror,
      Science Fiction, Animation, Adventure, Thriller, Romance*/
%let Genre = Science Fiction;

title "Median Return for &Genre Movies by Decade";
proc sql;
      select "1: Before 1980s" as Decade, count(d.id) as Count,
median(f.budget) as MedianBudget format = dollar16.2,
            median(f.net_revenue) as MedianNetRevenue format =
dollar16.2,
            median(f.net_revenue/f.budget) as MedianRoi format =
percent8.2
      from proj.movies_details d
      inner join proj.movies_production p on d.id = p.id
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_descriptions c on d.id = c.id
            and year(p.release_date) < 1980
            and      c.genres      contains      "&Genre"      and
p.production_countries contains "US"
      UNION
      select "2: 1980s" as Decade, count(d.id) as Count,
median(f.budget) as MedianBudget format = dollar16.2,
            median(f.net_revenue) as MedianNetRevenue format =
dollar16.2,
            median(f.net_revenue/f.budget) as MedianRoi format =
percent8.2
      from proj.movies_details d
```

```
      inner join proj.movies_production p on d.id = p.id
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_descriptions c on d.id = c.id
           and      year(p.release_date)      >      1979      and
year(p.release_date) < 1990
           and      c.genres      contains      "&Genre"      and
p.production_countries contains "US"
      UNION
      select   "3:   1990s"   as   Decade,   count(d.id)   as   Count,
median(f.budget) as MedianBudget format = dollar16.2,
           median(f.net_revenue)   as   MedianNetRevenue   format   =
dollar16.2,
           median(f.net_revenue/f.budget)   as   MedianRoi   format   =
percent8.2
      from proj.movies_details d
      inner join proj.movies_production p on d.id = p.id
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_descriptions c on d.id = c.id
           and      year(p.release_date)      >      1989      and
year(p.release_date) < 2000
           and      c.genres      contains      "&Genre"      and
p.production_countries contains "US"
      UNION
      select   "4:   2000s"   as   Decade,   count(d.id)   as   Count,
median(f.budget) as MedianBudget format = dollar16.2,
           median(f.net_revenue)   as   MedianNetRevenue   format   =
dollar16.2,
           median(f.net_revenue/f.budget)   as   MedianRoi   format   =
percent8.2
      from proj.movies_details d
      inner join proj.movies_production p on d.id = p.id
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_descriptions c on d.id = c.id
           and      year(p.release_date)      >      1999      and
year(p.release_date) < 2010
           and      c.genres      contains      "&Genre"      and
p.production_countries contains "US"
      UNION
      select   "5:   2010s"   as   Decade,   count(d.id)   as   Count,
median(f.budget) as MedianBudget format = dollar16.2,
           median(f.net_revenue)   as   MedianNetRevenue   format   =
dollar16.2,
           median(f.net_revenue/f.budget)   as   MedianRoi   format   =
percent8.2
      from proj.movies_details d
      inner join proj.movies_production p on d.id = p.id
      inner join proj.financials_filtered f on d.id = f.id
      inner join proj.movies_descriptions c on d.id = c.id
           and      year(p.release_date)      >      2009      and
year(p.release_date) < 2020
```

```
          and      c.genres      contains      "&Genre"      and
p.production_countries contains "US";
quit;
title;
```

Script: Figures 7 & 8: Analyze production company financials over time.

This script is used to produce a report showing financial performance over time for one specific production company. This script was used for each production company to produce a line graph representation of financial performance over time for each production company compared to the median.

Please note, this script does not meet the requirement of being released after the year 1999.

```
/*Skills used: #28, #29, #38*/
/*Check financial performance of Production Company over time.
Set Production Company below. Options include:
     Universal  Pictures,  Paramount  Pictures,  Walt  Disney,
Twentieth Century Fox, Warner Bros*/
data _null_;
     call symputx('ProdCompany', 'Walt Disney');
     test=symget('ProdCompany');
     PUT "NOTE: The variable test is equal to " test;
run;

title "Median Return for &ProdCompany Movies by Decade";
proc sql;
     select "1: Before 1980s" as Decade, count(d.id) as Count,
median(f.budget) as MedianBudget format = dollar16.2,
          median(f.net_revenue)  as  MedianRevenue  format  =
dollar16.2,
          median(f.net_revenue/f.budget) as MedianRoi format =
percent8.2
     from proj.movies_details d
     inner join proj.movies_production p on d.id = p.id
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_descriptions c on d.id = c.id
          and year(p.release_date) < 1980
          and p.production_companies contains "&ProdCompany"
          and p.production_countries contains "US"
     UNION
     select  "2:  1980s"  as  Decade,  count(d.id)  as  Count,
median(f.budget) as MedianBudget format = dollar16.2,
          median(f.net_revenue)  as  MedianRevenue  format  =
dollar16.2,
          median(f.net_revenue/f.budget) as MedianRoi format =
percent8.2
     from proj.movies_details d
     inner join proj.movies_production p on d.id = p.id
     inner join proj.financials_filtered f on d.id = f.id
```

```
     inner join proj.movies_descriptions c on d.id = c.id
          and      year(p.release_date)      >      1979      and
year(p.release_date) < 1990
          and p.production_companies contains "&ProdCompany"
          and p.production_countries contains "US"
     UNION
     select  "3:  1990s"  as  Decade,  count(d.id)  as  Count,
median(f.budget) as MedianBudget format = dollar16.2,
          median(f.net_revenue)   as   MedianRevenue   format   =
dollar16.2,
          median(f.net_revenue/f.budget)  as  MedianRoi  format  =
percent8.2
     from proj.movies_details d
     inner join proj.movies_production p on d.id = p.id
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_descriptions c on d.id = c.id
          and      year(p.release_date)      >      1989      and
year(p.release_date) < 2000
          and p.production_companies contains "&ProdCompany"
          and p.production_countries contains "US"
     UNION
     select  "4:  2000s"  as  Decade,  count(d.id)  as  Count,
median(f.budget) as MedianBudget format = dollar16.2,
          median(f.net_revenue)   as   MedianRevenue   format   =
dollar16.2,
          median(f.net_revenue/f.budget)  as  MedianRoi  format  =
percent8.2
     from proj.movies_details d
     inner join proj.movies_production p on d.id = p.id
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_descriptions c on d.id = c.id
          and      year(p.release_date)      >      1999      and
year(p.release_date) < 2010
          and p.production_companies contains "&ProdCompany"
          and p.production_countries contains "US"
     UNION
     select  "5:  2010s"  as  Decade,  count(d.id)  as  Count,
median(f.budget) as MedianBudget format = dollar16.2,
          median(f.net_revenue)   as   MedianRevenue   format   =
dollar16.2,
          median(f.net_revenue/f.budget)  as  MedianRoi  format  =
percent8.2
     from proj.movies_details d
     inner join proj.movies_production p on d.id = p.id
     inner join proj.financials_filtered f on d.id = f.id
     inner join proj.movies_descriptions c on d.id = c.id
          and      year(p.release_date)      >      2009      and
year(p.release_date) < 2020
          and p.production_companies contains "&ProdCompany"
          and p.production_countries contains "US";
```

```
quit;
title;
```

Additional analysis performed.
The scripts included in this section were not used in this final project analysis. These scripts were used to demonstrate certain concepts as requirements for this course project.

Script: Macro to determine what genres are above median.
This script iterates over each genre, and outputs a note to the log informing whether the genre is greater or less than the median return on investment. This script was written to demonstrate the use of a "Do" loop in a user-defined macro.

```
/*Skills used: #27, #30, #31, #32, #45, #46*/
/*Macro iteration over genre*/
options mlogic mprint symbolgen mcompilenote=all;
%macro CheckIfGenreAboveMedian;
      %let var1 = Action;
      %let var2 = Adventure;
      %let var3 = Animation;
      %let var4 = Comedy;
      %let var5 = Drama;
      %let var6 = Horror;
      %let var7 = Music;
      %let var8 = Romance;
      %let var9 = Science Fiction;
      %let var10 = Thriller;
      %local GenreRoi;
      %local MedianRoi;
      %local i;

      %do i=1 %to 10;
            proc sql noprint;
                  select MedianRoiPercent
                  into :GenreRoi
                  from proj.genreMedianroi
                  where Genre = "&&var&i";

                  select MedianRoiPercent
                  into :MedianRoi
                  from proj.genreMedianroi
                  where Genre = "MEDIAN";
            quit;

            %if &GenreRoi > &MedianRoi %then %do;
                  %put NOTE: &&var&i is greater than the MEDIAN ROI.;
            %end;
            %else %do;
                  %put NOTE: &&var&i is less than the MEDIAN ROI.;
            %end;
```

```
	%end;
%mend;

%CheckIfGenreAboveMedian;
```

Script: Identify top 10 movies by return on investment.

This script returns the top 10 movies by the greatest return on investment. The "validate" key word is used to validate the query. However, once this is removed the full report will be returned. This script was written to demonstrate use of the "validate" key word, as well as querying an in-line view.

```
/*Skills used: #4, #5, #6, #7, #14*/
/*Top ROI numbers*/
title "Top 10 Movies by ROI Percentage";
proc sql outobs=10;
	validate
	select d.id, d.title, c.genres, d.runtime, f.budget
format=Dollar12.,
		f.net_revenue format=Dollar12.,
		(f.net_revenue / f.budget) as ROI format=percent12.2
	from proj.movies_details d
	inner join proj.movies_financials f on d.id = f.id
	inner join proj.movies_descriptions c on d.id = c.id
	inner join
		(select *
		from proj.movies_production
		where production_countries contains "US" and
year(release_date) > 1999
			and status = "Released")
		p on d.id = p.id
	where f.budget > 1000 and f.revenue > 1000
	order by ROI desc;
quit;
```

Script: Report showing statistics by original_language.

This script was written to demonstrate the use of non-correlated and correlated sub-queries. The first report returns the original languages which have a net ROI above the overall median. The second report shows a list of financial data for all English movies.

```
/*Skills used: #12*/
/*View which original languages have net roi above the median*/
/*Non-Correlated Subquery*/
title "Which Original Languages have an Median Net ROI Above the
Overall median?";
proc sql;
	select count(f.id) as NumMovies, d.original_language,
median(f.revenue) as medianRev format=dollar16.2,
		median(f.net_revenue/f.budget) as medianNetRoi
format=percent8.2
```

```
	from proj.financials_filtered f
	inner join proj.movies_production_us p on p.id=f.id
	inner join proj.movies_details d on f.id=d.id

	group by d.original_language
	having (median(f.net_revenue/f.budget)) >
		(select median(net_revenue) / median(budget)
		from proj.financials_filtered g
		inner join proj.movies_production_us u on g.id=u.id);
quit;
title;

/*Skills used: #13*/
/*Correlated Subquery*/
title "List of All English Movies' Financials";
proc sql;
	select	count(f.id),	f.budget	format=dollar16.2,
f.net_revenue format=dollar16.2,
		f.net_revenue / f.budget as Roi format=percent8.2
	from proj.financials_filtered f
	inner join proj.movies_production_us p on p.id=f.id
	where 'en' =
		(select d.original_language
		from proj.movies_details d
		where d.id = f.id);
quit;
title;
```

Works Cited

Banik, R. (2017, November 10). The Movies Dataset. Retrieved March 2, 2019, from https://www.kaggle.com/rounakbanik/the-movies-dataset. *Creative Commons Public Domain* license (*CC0: Public Domain: https://creativecommons.org/publicdomain/zero/1.0/)*. Data set used in report and analysis.

GroupLens. (2018, September 27). MovieLens Latest Datasets. Retrieved March 2, 2019, from https://grouplens.org/datasets/movielens/latest/. Dataset used by Kaggle user to compile the full dataset used in this report.

The Movie DB. API Overview. Retrieved March 2, 2019, from https://www.themoviedb.org/documentation/api. API used to retrieve movie data from The Movie DB website.