# Stat 306

## Term Project: Multiple Regression Analysis of IMDB Movie Data Set

April 07, 2017

Team Members:

Sam McConnell

Kelly Ng

Luke Schwalfenberg

Richard Wang

Table of Contents

[Stat 306 1](#_Toc479242306)

[Term Project: Multiple Regression Analysis of IMDB Movie Data Set 1](#_Toc479242307)

[Abstract/Summary 3](#_Toc479242308)

[Description of Data 3](#_Toc479242309)

[Data Analysis and Results 4](#_Toc479242310)

[Initial Numerical Analysis 4](#_Toc479242311)

[Overall Analysis with Quantitative Variables 9](#_Toc479242312)

[Directors Analysis 10](#_Toc479242313)

[Description of Data (Director’s Analysis) 11](#_Toc479242314)

[Data Analysis and Results 12](#_Toc479242315)

[Residual Plots 12](#_Toc479242316)

[Brief Discussion (Director’s Analysis): 17](#_Toc479242317)

[Actor’s Analysis 18](#_Toc479242318)

[Abstract/Summary 18](#_Toc479242319)

[Description of Data 18](#_Toc479242320)

[Data Analysis and Results 19](#_Toc479242321)

[Sample Correlations 19](#_Toc479242322)

[Residual Plots 20](#_Toc479242323)

[Brief Discussion (Actor’s Analysis): 23](#_Toc479242324)

[Conclusion: 24](#_Toc479242325)

[Contributions 25](#_Toc479242326)

[Team Formation: 25](#_Toc479242327)

[Authorship: 25](#_Toc479242328)

[Teamwork and contributions: 25](#_Toc479242329)

[APPENDIX I - Director’s Analysis 26](#_Toc479242330)

[APPENDIX II - Sample Corr Matrix, Director’s Analysis 39](#_Toc479242331)

[APPENDIX III – Inflation Rate Table 40](#_Toc479242332)

[APPENDIX IV – Sample Correlation Matrix – Actor’s Analysis 41](#_Toc479242333)

# Abstract/Summary

The goal of this investigation is to form a predictive equation for Box Office Gross Earnings. The primary explanatory variables being considered for the model are: Director, Number of Critic Reviews, Duration, Lead Actor/Actress, Number of User Votes, Content Rating, Budget, IMDB Score and in some cases the number of Actor/Actress/Director's facebook likes and some other variables. After some initial analysis, it made sense to create several different subsets for this data set since there are 2399 unique director names, and several thousand different actresses and actors that star in the films examined. The subsets that were chosen for consideration included a model that does not include lead actor/actresses or directors and consists primarily of numerical data, a model which includes directors who have directed more than 3 movies and a model which includes actor/actresses appearing in more than 25 movies. To get a regression model or equation so that the homoscedasticity assumption approximately holds for the models considered, the response variable was transformed to log(BoxOfficeGrossEarnings).

# Description of Data

In this case study, data was retrieved from a data set made publically available on Kaggle by user chuansun76 who had used python scripts to mine this data from Imdb. This data set includes roughly 5000 individual entries with the earliest movie dating from 1916. Roughly 50% of the dataset is from movies released after 2004. The values of Gross Revenue and Budget were in USD. Sample sizes varied depending on the model being examined. Since a number of key values were missing for data points, no sample size was as large as that of the original data set. Some of the explanatory variables examined can be found in Table 1.

Table 1: Tables of Variables That Might Explain Movie Box Office Gross

|  |  |
| --- | --- |
| Variables | Explanation or Unit |
| Box Office Gross Earnings | Movie gross box office earnings in USD |
| Director | Categorical variable for directors |
| Number of Critic Reviews | Measurement of movie critical success |
| Lead Actor/Actress | Categorical variable for lead actor actresses |
| Number of User Votes | Measurement of movie popular success, the number of users who voted on Imdb |
| Content Rating | Content rating: G, PG, PG-13, R, X |
| Budget | Movie budget in USD |
| IMDB Score | Score on a scale of 0.0 - 10.0 |
| Actor | Lead Actor in film |

# Data Analysis and Results

## Initial Numerical Analysis

Summary statistic and plots of the main numerical variables can be found in Figures 1, 2 and 3 and in Tables 2 - 6. Figure 3 suggests that log(Gross Revenue) should be transformed so that the deviations from the prediction equation can be closer to homoscedastic.

Figure 1: Plot of Gross Revenue and log(Gross Revenue) versus Imdb Score

|  |
| --- |
| ImdbGross.png |

Figure 2: Box plot of log(Gross Revenue) vs. Content Rating. Far right box plot is for those films that did not have a content rating associated with them in the data set.

|  |
| --- |
| BoxPlotRatings.png |

Figure 3: Plots of Gross Revenue and log(Gross Revenue) before adjusting for inflation versus the numerical explanatory variables, and plots of log(Gross Revenue) vs. logged versions of the numerical explanatory variables

|  |
| --- |
| NumVotesGross.pngBudgetGross.pngCriticReviewsGross.pngNumVotesGross.png |

Table 2: Summary Statistics of Variables included in Data Sets

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | num\_critic\_for\_reviews | lnnum\_critic\_reviews | duration | director\_facebook\_likes | actor\_3\_facebook\_likes |
| Min. | 1 | 0 | 7 | 0 | 0 |
| 1st Qu. | 53 | 3.97 | 93 | 7 | 165.2 |
| Median | 113 | 4.727 | 103 | 52 | 399 |
| Mean | 143.5 | 4.508 | 106.8 | 726.4 | 684.1 |
| 3rd Qu. | 198 | 5.288 | 117 | 209 | 651 |
| Max. | 813 | 6.701 | 334 | 23000 | 23000 |

Table 3: Continuation of Summary Statistics of Variables that may be included in data sets

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | actor\_1\_facebook\_likes | gross | lngross | num\_voted\_users | lnnum\_voted\_users |
| Min. | 0 | 703 | 6.555 | 5 | 1.609 |
| 1st Qu. | 670 | 7164367 | 15.785 | 9808 | 9.191 |
| Median | 1000 | 28429808 | 17.163 | 37462 | 10.531 |
| Mean | 6951 | 51070645 | 16.555 | 87523 | 10.186 |
| 3rd Qu. | 11000 | 65526300 | 17.998 | 102110 | 11.534 |
| Max. | 640000 | 760505847 | 20.45 | 1689764 | 14.34 |

Table 4: Continuation of Summary Statisticcs of Variables that may be included in data sets

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | cast\_total\_facebook\_likes | facenumber\_in\_poster | num\_user\_for\_reviews | budget (USD) | title\_year |
| Min. | 0 | 0 | 1 | 218 | 1916 |
| 1st Qu. | 1622 | 0 | 73 | 6500000 | 1999 |
| Median | 3339 | 1 | 165 | 20000000 | 2005 |
| Mean | 10290 | 1.384 | 284.4 | 34484957 | 2002 |
| 3rd Qu. | 14578 | 2 | 340 | 45000000 | 2010 |
| Max. | 656730 | 43 | 5060 | 390000000 | 2016 |

Table 5: Continuation of Summary Statistic of Variables that may be included in data sets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | actor\_2\_facebook\_likes | imdb\_score | aspect\_ratio | movie\_facebook\_likes |
| Min. | 0 | 1.6 | 1.18 | 0 |
| 1st Qu. | 329 | 5.8 | 1.85 | 0 |
| Median | 631 | 6.5 | 2.35 | 175.5 |
| Mean | 1755 | 6.409 | 2.209 | 7797.7 |
| 3rd Qu. | 936 | 7.2 | 2.35 | 4000 |
| Max. | 137000 | 9.5 | 16 | 349000 |

Table 6: Sample Correlations of Select Explanatory Variables

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | num\_critic\_for\_reviews | duration | gross | num\_voted\_users | num\_user\_for\_reviews | budget | imdb\_score | lngross | lnnum\_critic\_reviews l | lnnum\_voted\_users |
| num\_critic\_for\_reviews | 1.00 | 0.26 | 0.48 | 0.60 | 0.57 | 0.48 | 0.35 | 0.41 | 0.86 | 0.68 |
| duration | 0.26 | 1.00 | 0.28 | 0.36 | 0.38 | 0.31 | 0.37 | 0.25 | 0.23 | 0.34 |
| gross | 0.48 | 0.28 | 1.00 | 0.64 | 0.55 | 0.65 | 0.24 | 0.62 | 0.41 | 0.56 |
| num\_voted\_users | 0.60 | 0.36 | 0.64 | 1.00 | 0.78 | 0.40 | 0.49 | 0.42 | 0.50 | 0.70 |
| num\_user\_for\_reviews | 0.57 | 0.38 | 0.55 | 0.78 | 1.00 | 0.42 | 0.33 | 0.39 | 0.50 | 0.61 |
| budget | 0.48 | 0.31 | 0.65 | 0.40 | 0.42 | 1.00 | 0.05 | 0.49 | 0.42 | 0.45 |
| imdb\_score | 0.35 | 0.37 | 0.24 | 0.49 | 0.33 | 0.05 | 1.00 | 0.13 | 0.32 | 0.43 |
| lngross | 0.41 | 0.25 | 0.62 | 0.42 | 0.39 | 0.49 | 0.13 | 1.00 | 0.50 | 0.69 |
| lnnum\_critic\_reviews | 0.86 | 0.23 | 0.41 | 0.50 | 0.50 | 0.42 | 0.32 | 0.50 | 1.00 | 0.80 |
| lnnum\_voted\_users | 0.68 | 0.34 | 0.56 | 0.70 | 0.61 | 0.45 | 0.43 | 0.69 | 0.80 | 1.00 |

## 

## Overall Analysis with Quantitative Variables

The goal of this analysis is to determine a regression equation based solely on quantitative explanatory variables. The best prediction equation, after residual plots, variable selection and using criteria of adjusted R2 and cross-validated root mean square prediction error is:

log(adjgross) = -14.342 - 0.438(num\_critic\_for\_reviews) - 0.045(director\_facebook\_likes) - 0.045(director\_facebook\_likes) - 0.410(actor\_1\_facebook\_likes) + 0.843(num\_voted\_users) + 0.344(cast\_total\_facebook\_likes) + 0.266(num\_user\_for\_reviews) + 0.472(budget) - 0.094(imdb\_score)

According to the regression on quantitative variables (Tables 7 and 8), almost all the variables are significant. num\_voted \_users has the highest positive coefficient of 0.843, while num\_critic\_for\_reviews appears to have the largest negative coefficient of -0.438. Therefore, in our next analyses, we will include categorical variables that incorporate directors and actors.

Table 7: Multiple Regression Summary of Quantitative Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | t value | Pr(>|t|) |  |
| (Intercept) | -14.34181 | 0.51136 | -28.047 | < 2e-16 | \*\*\* |
| num\_critic\_for\_reviews | -0.43824 | 0.07237 | -6.056 | 1.74e-09 | \*\*\* |
| director\_facebook\_likes | -0.04461 | 0.02305 | -1.935 | 0.053138 | . |
| actor\_1\_facebook\_likes | -0.41035 | 0.08652 | -4.743 | 2.30e-06 | \*\*\* |
| num\_voted\_users | 0.84273 | 0.06342 | 13.288 | < 2e-16 | \*\*\* |
| cast\_total\_facebook\_likes | 0.34444 | 0.10268 | 3.354 | 0.000814 | \*\*\* |
| num\_user\_for\_reviews | 0.26582 | 0.07839 | 3.391 | 0.000714 | \*\*\* |
| budget | 0.4724 | 0.03179 | 14.861 | < 2e-16 | \*\*\* |
| imdb\_score | -0.094 | 0.04179 | -2.249 | 0.024631 | \* |

Backward Elimination and Forward Selection algorithms were applied to determine the best regression model which is shown below:

Table 8: Cross Validation and Out of Sample Comparisons for Quantitative Analysis

|  |  |  |
| --- | --- | --- |
| statistic\model | Full Model | Best Model |
| adjusted R squared | 0.5638149 | 0.5641097 |
| residual SD | 1.462724 | 1.462229 |
| rmsepred(leave-one-out) | 1.470517 | 1.468138 |
| rmsepred(train/holdout) | 1.479248 | 1.473195 |

## Directors Analysis

This purpose of this separate analysis is to look at the dataset with a focus on directors who have more than 3 films in their filmography with respect to our dataset. The goal of this subset analysis is to find a prediction equation for predicting box office gross with a focus on directors.

Similar to the other models in this report, the response variable of adjusted gross was transformed to log(adjgross).

The best prediction equation, after residual plots, variable selection and using criteria of adjusted R2 and cross-validated root mean square prediction error is:

log(adjgross) = 7.402 – 0.134(lncast\_total\_facebook\_likes) – 0.00341(lnactor1\_facebook\_likes) - 0.004(lndirector\_facebook\_likes) + 1.08(lnnum\_voted\_users) – 0.306(lnnum\_users\_for\_review) + 0.106(lnbudget) – 0.192(imdb\_score) – 0.950(content\_ratingPG-13) – 1.06(content\_ratingR) + 0.007(duration) +0.382(BobbyFarrelly) + 1.06(BrettRatner) + 0.776(ClintEastwood) – 0.071(DavidFincher) + 1.043(JoelSchumacher) + 1.066(JohnMcTiernan) – 0.170(MartinScorsese) + 0.143(PaulWSAnderson) + 0.654(RennyHarlin) + 1.40(RichardDonner) + 1.33(RobCohen) + 0.547(RonHoward) + 0.52(ShawnLevy) + 0.616(StephenFrears) + 1.01(StevenSpielberg) + 0.590(TimBurton) + 0.643(TonyScott)

Note that director categorical variables are not grouped together. This is because directors act like a brand, and grouping them together would not provide any useful analysis as directors do not work together in real life.

Significant categorical variables include Steven Spielberg, Rob Cohen, Clint Eastwood and Brett Ratner. A review of their films show that their films are not only typically high grossing but very profitable.

Steven Spielberg has a high coefficient of 1.009 to the log(adjgross), Rob Cohen is 1.33, Clint Eastwood is 0.776 and Brett Ratner is 1.06.

### 

### Description of Data (Director’s Analysis)

Data are collected by scraping both imdb resources and Box Office Mojo websites. Assumptions were made regarding the adjusted gross (estimated data from Box Office Mojo) which are outlined in Appendix III. The variables used in this analysis can be found in Table 9.

Table 9: Tables of variables that might explain movie box office gross for director's analysis

|  |  |
| --- | --- |
| Variables | Explanation or unit |
| adjgross | Gross US domestic box office, adjusted for inflation |
| director\_facebook\_likes | Number of Facebook likes a director has |
| actor\_1\_facebook\_likes | Number of Facebook likes from the lead actor |
| duration | Duration of the film (in minutes) |
| num\_voted\_users | Number of users who have voted for film on imdb |
| cast\_total\_facebook\_likes | Number of Facebook likes received by the cast |
| Budget | Budget for the film |
| num\_users\_for\_review | Number of uses that have reviewed the film |
| movie\_facebook\_likes  content\_rating  director\_name | Number of Facebook likes received by the film  Rating for the film (e.g. PG, R, NC-17, etc.)  Film directors that have made more than 3 films |

This particular data set subsets the original data set such that only directors that have made more than 3 films are included. This reduces the dataset from over 5000 observations to approximately 400.

Directors are not combined into fewer categorical variables because in practical situations, directors do not work together and their contributions to the movie industry are considered singular.

### 

### 

### Data Analysis and Results

Summary statistics are provided in Tables 10, 11, 12 and 13. Plots that justify the use of transforms for explanatory variables are given in Figure 4 and remain unchanged through this subset analysis.

The sample correlation matrix for this analysis is in Appendix II. Note the lnadjgross is more highly correlated with lnnum\_user\_for\_reviews (0.657) and lnmovie\_facebook\_likes (0.594).

We fit a multiple regression model (Table 14) with the explanatory variables given above. The adj-R2 of this model is 0.7412 with a Residual Standard Error of 0.6683.

### Residual Plots

Sample residual plots are given in Figure 4. The residuals do not suggest any curvilinear form, therefore quadratic terms or interaction terms were not added to the model.

Figure 4: Residual plots for some explanatory variables included in the director's analysis

|  |
| --- |
|  |

Backward Elimination and Forward Selection techniques were used to determine other good models. The 2nd best model eliminated two explanatory variables: lntotal\_facebook\_likes and lndirector\_facebook\_likes. This model, as well as the full model, are compared via Cp statistic, adj-R2 and error from a random holdout set (Table 15).

Table 10: Summary statistics of variables included in directors' regression analysis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | lnnum\_voted\_users | lncast\_total\_facebook\_likes | lnbudget | lnnum\_users\_for\_review | lnmovie\_facebook\_likes |  |
| Min. | 7.243 | 6.252 | 14.08 | 3.258 | 4.883 |  |
| 1st Q | 10.587 | 8.12 | 16.81 | 5.13 | 7.905 |  |
| Median | 11.789 | 9.483 | 17.5 | 5.807 | 9.547 |  |
| Mean | 11.644 | 9.177 | 17.5 | 5.826 | 9.079 |  |
| 3rd Q | 12.615 | 10.096 | 18.06 | 6.539 | 10.127 |  |
| Max. | 14.114 | 11.3 | 19.11 | 7.996 | 11.891 |  |
| Table 11: Continuation of summary statistics of variables included in director’s regression analysis | | | | | |  |
|  | lnadjgross | lngross | lndirector\_facebook\_likes | lnactor\_1\_facebook\_likes | duration |  |
| Min. | 11.63 | 11.23 | 4.615 | 5.935 | 87 |  |
| 1st Q | 17.34 | 17.08 | 5.858 | 6.872 | 103 |  |
| Median | 18.05 | 17.69 | 9.306 | 9.306 | 116 |  |
| Mean | 17.92 | 17.64 | 7.858 | 8.575 | 119.8 |  |
| 3rd Q | 18.83 | 18.44 | 9.547 | 9.695 | 132 |  |
| Max. | 20.81 | 19.89 | 9.952 | 10.8 | 240 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 12: Frequency Table for Directors with >= 3 Movies | | | | | |  | |
| Barry Levinson | Bobby Farrelly | Brett Ratner | Clint Eastwood | David Fincher |  | |
| 5 | 4 | 4 | 13 | 8 |  | |
| Joel Schumacher | John McTiernan | Martin Scorsese | Paul WS Anderson | Renny Harlin |  | |
| 3 | 4 | 6 | 5 | 9 |  | |
| Rob Cohen | Ron Howard | Shawn Levy | Stephen Frears | Steven Spielberg |  | |
| 6 | 6 | 8 | 3 | 17 |  | |
| Tim Burton | Tony Scott | Woody Allen |  |  |  | |
| 9 | 4 | 12 |  |  |  | |
| Table 13: Frequency Table for Content Rating | | | | |  | |
| Approved | G | GP | M | NC-17 |  | |
| 303 | 55 | 112 | 6 | 5 |  | |
| PG-13 | R | TV-14 | TV-G | TV-MA |  | |
| 1461 | 2118 | 30 | 10 | 20 |  | |
|  |  |  |  |  |  | |
| Not | Rated | Passed | PG | X |  | |
| 7 | 116 | 9 | 701 | 13 |  | |
| TV-PG | TV-Y | TV-Y7 | Unrated |  |  | |
| 13 | 1 | 1 | 62 |  |  | |

Table 14: Multiple Regression Summary for director's analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable Name | Estimate | Std. Error | t value | Pr(>|t|) |  |
| (Intercept) | 7.402044 | 2.343641 | 3.158 | 0.0021 | \*\* |
| lncast\_total\_facebook\_likes | -0.133733 | 0.227064 | -0.589 | 0.55721 |  |
| lnactor\_1\_facebook\_likes | -0.003454 | 0.180389 | -0.019 | 0.98476 |  |
| lndirector\_facebook\_likes | -0.003638 | 0.112191 | -0.032 | 0.97419 |  |
| lnnum\_voted\_users | 1.075662 | 0.177303 | 6.067 | 2.34E-08 | \*\*\* |
| lnnum\_users\_for\_review | -0.306319 | 0.222169 | -1.379 | 0.17104 |  |
| lnbudget | 0.105663 | 0.115545 | 0.914 | 0.36267 |  |
| imdb\_score | -0.191828 | 0.134387 | -1.427 | 0.15657 |  |
| content\_ratingPG-13 | -0.949634 | 0.232948 | -4.077 | 9.18E-05 | \*\*\* |
| content\_ratingR | -1.064077 | 0.26723 | -3.982 | 0.00013 | \*\*\* |
| duration | 0.006723 | 0.005051 | 1.331 | 0.18622 |  |
| director\_nameBobby Farrelly | 0.382473 | 0.591534 | 0.647 | 0.51939 |  |
| director\_nameBrett Ratner | 1.061384 | 0.509952 | 2.081 | 0.03996 | \* |
| director\_nameClint Eastwood | 0.775558 | 0.357935 | 2.167 | 0.03263 | \* |
| director\_nameDavid Fincher | -0.07068 | 0.45463 | -0.155 | 0.87677 |  |
| director\_nameJoel Schumacher | 1.043301 | 0.529577 | 1.97 | 0.0516 | . |
| director\_nameJohn McTiernan | 1.066037 | 0.511246 | 2.085 | 0.0396 | \* |
| director\_nameMartin Scorsese | -0.169529 | 0.515933 | -0.329 | 0.74315 |  |
| director\_namePaul W.S. Anderson | 0.143406 | 0.466768 | 0.307 | 0.75931 |  |
| director\_nameRenny Harlin | 0.653876 | 0.464068 | 1.409 | 0.16193 |  |
| director\_nameRichard Donner | 1.396116 | 0.624239 | 2.237 | 0.02754 | \* |
| director\_nameRob Cohen | 1.329123 | 0.485584 | 2.737 | 0.00734 | \*\* |
| director\_nameRon Howard | 0.546761 | 0.406754 | 1.344 | 0.18192 |  |
| director\_nameShawn Levy | 0.519174 | 0.506346 | 1.025 | 0.30768 |  |
| director\_nameStephen Frears | 0.616334 | 0.544897 | 1.131 | 0.26072 |  |
| director\_nameSteven Spielberg | 1.009328 | 0.365303 | 2.763 | 0.00682 | \*\* |
| director\_nameTim Burton | 0.589871 | 0.373339 | 1.58 | 0.11727 |  |
| director\_nameTony Scott | 0.64321 | 0.458547 | 1.403 | 0.1638 |  |
| Residual standard error: 0.7493 on 100 degrees of freedom | | |  |  |  |
| Multiple R-squared: 0.7569, | Adjusted R-squared: 0.6913 | | |  |  |
| F-statistic: 11.53 on 27 and 100 DF, p-value: < 2.2e-16 | | |  |  |  |

Table 15: Cross Validation and Out of Sample Comparisons for Quantitative Analysis

|  |  |  |
| --- | --- | --- |
| Statistic / Model | fit1 | fit2 |
| Adj-R2 | 0.6913 | 0.6908 |
| Res-SD | 0.75 | 0.75 |
| rmsepred(leave-one-out) | 0.84 | 0.832 |
| rmsepred(train / holdout) | 1.884 | 3.367 |

The RMSE for the training/holdout is quite high. It’s possible that there could be some issues with the data (the variables that were omitted for the second model reduced the RMSE by a large factor, therefore perhaps those variables should not be included in the analysis), or that simply it is very hard to predict the performance of films, and high variability is to be expected when doing an analysis within this industry.

## 

### Brief Discussion (Director’s Analysis):

In conclusion, we have found a good-fitting model when the model focuses on directors as categorical variables, and residual plots that show no sign of curvilinear form. There are some directors that are significantly correlated with box office gross (i.e. Steven Spielberg, Clint Eastwood).

The sample did not include foreign films – only domestic films made in the US were used for this subset. It is possible to include genre data or actor data, but the model suffered from overfitting if the categorical data is not grouped into fewer groups. Please see the Actor’s Analysis for a more comprehensive regression with consideration of actors.

## Actor’s Analysis

### Abstract/Summary

This purpose of this separate analysis is to look at the dataset with a focus on actors who have performed in more than 25 films in their filmography with respect to our dataset. The goal is to find a prediction equation for predicting box office gross with a focus on actors.

The response variable of adjusted gross was transformed to log(adjgross), otherwise the residual plots do not seem heteroscedastic .

The best prediction equation, after residual plots, variable selection and using criteria of adjusted R2 and cross-validated root mean square prediction error is:

log(adjgross) = -2.0176274- -0.0062244duration

+ 0.8828473(lognum\_voted\_users) - 0.0031701(num\_critic\_for\_reviews)

+ 0.5978969(logbudget) + 0.1475909(lognum\_user\_for\_reviews) + 0.0233070(imdb\_score) + 0.6570164(Denzel Washington) + 0.7746052(Harrison Ford) + 0.4390425(J.K. Simmons) - 0.5567939(Jason Statham) - 0.0978200(Johnny Depp) + 0.0513779(Liam Neeson) + 0.1420526(Matt Damon) - 0.0958409(Nicolas Cage) - 0.0127797(Robert De Niro) + 0.1114163(Robert Downey Jr.) + 0.3149969(Robin Williams) - 0.1972292(content\_ratingPG) - 0.4430192(content\_ratingPG-13) - 0.8910856(content\_ratingR) + 1.3003917 (content\_ratingX)

Actor categorical variables are not grouped together. Since an actor/actress acts as a brand, and this analysis focuses on the effect of a single main actor on the gross grouping actors/actresses them together would not provide any useful analysis. Significant categorical variables of actors include Denzel Washington, Harrison Ford and Jason Statham.

Denzel Washington has a high coefficent of 0.657, Harrison Ford is 0.775, and Jason Statham is 0.557.

### Description of Data

Data are collected by scraping both imdb resources and Box Office Mojo websites. Assumptions were made regarding the adjusted gross (estimated data from Box Office Mojo) which are outlined in APPENDIX III. Variables used in this analysis can be found in Table 16.

Table 16: Table of variables that might explain movie box office ross for actor's analysis

|  |  |
| --- | --- |
| Variables | Explanation or unit |
| adjgross | Gross US domestic box office, adjusted for inflation |
| actor\_1\_facebook\_likes | Number of Facebook likes from the lead actor |
| duration | Duration of the film (in minutes) |
| num\_voted\_users | Number of users who have voted for film on imdb |
| cast\_total\_facebook\_likes | Number of Facebook likes received by the cast |
| Budget | Budget for the film |
| num\_critic\_for\_review | Number of uses that have reviewed the film |
| movie\_facebook\_likes  content\_rating  actor\_name | Number of Facebook likes received by the film  Rating for the film (e.g. PG, R, NC-17, etc.)  Actors who has appeared in more than 25 films |

This particular data set subsets the original data set such that only actors that have appeared in more than 25 films are considered. This reduces the dataset from over 5000 observations to 356.

Actors are not combined into fewer categorical variables because we try to focus on the effect of a single actor so their contributions to the gross are considered singular.

### Data Analysis and Results

Summary statistics are provided are given in Tables 17, 18 and 19. Plots that justify the use of transforms for explanatory variables are given in Figure 3 and remain unchanged through this subset analysis.

### Sample Correlations

The highest correlation with the log(adjgross) appears to be the log(num\_voted\_users) with 0.609, log(budget) with 0.540 and log(num\_user\_for\_reviews) with 0.517. Please see Appendix IV for the full sample correlation matrix.

We fit a multiple regression model (Table 20) with the explanatory variables given above. The adj-R2 of this model is 0.7412 with a Residual Standard Error of 0.6683.

### Residual Plots

Sample residual plots are given in Figure 5. The residuals do not suggest any curvilinear form, therefore quadratic terms or interaction terms were not added to the model.

Figure 5: Residual plots for some explanatory variables included in the actor's analysis

|  |
| --- |
|  |

Backward Elimination and Forward Selection techniques were used to determine other good models. The best fit model suggested by both adj and cp value eliminated two explanatory variables: logactor\_1\_facebook\_likes and logcast\_total\_facebook\_likes. This model, as well as the full model, are compared via Cp statistic, adj-R2 Cross-validation root mean square error(leave-one-out) and Cross-validation root mean square error from a random holdout set (Table 21).

Table 17: Summary statistics of variables in director's regression analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | num\_critic\_for\_reviews | duration | imdb\_score | logadjgross | logactor\_1\_facebook\_likes |
| Min: | 16.0 | 77.0 | 3.100 | 9.709 | 9.306 |
| 1st Qu: | 112.0 | 103.0 | 6.200 | 17.184 | 9.473 |
| Median: | 167.0 | 113.5 | 6.700 | 17.956 | 9.952 |
| Mean: | 196.6 | 117.3 | 6.722 | 17.686 | 9.903 |
| 3rd Qu: | 266.2 | 126.0 | 7.300 | 18.594 | 10.106 |
| Max: | 608.0 | 289.0 | 9.000 | 21.047 | 10.800 |

Table 18: Summary statistics of variables included in director's regression analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | lognum\_user\_for\_reviews  s | logcast\_total\_facebook\_likes | lognum\_voted\_users | logbudget |
| Min: | 3.401 | 9.384 | 7.979 | 9.547 |
| 1st Qu: | 5.112 | 9.726 | 10.770 | 16.860 |
| Median: | 5.768 | 10.136 | 11.562 | 17.504 |
| Mean: | 5.710 | 10.188 | 11.503 | 17.390 |
| 3rd Qu: | 6.260 | 10.596 | 12.299 | 18.081 |
| Max: | 7.694 | 11.300 | 14.097 | 19.519 |

Table 19: Frequency Table for actors involved >= 25 movies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bruce Willis | Denzel Washington | Harrison Ford | J.K. Simmons | Jason Statham |
| 29 | 30 | 25 | 31 | 25 |
| Johnny Depp | Liam Neeson | Matt Damon | Nicolas Cage | Robert De Niro |
| 39 | 26 | 28 | 30 | 42 |
| Robert Downey Jr. | Robin Williams |  |  |  |
| 26 | 25 |  |  |  |
|  |  |  |  |  |

Table 20: Multiple regression summary for actor's analysis

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| variable | Estimate | Std. Error | t value | Pr(>|t|) |  |  |  |
| (Intercept) | -2.01763 | 1.453236 | -1.388 | 0.16595 |  |  |  |
| duration | -0.00622 | 0.002684 | -2.319 | 0.02097 | \* |  |  |
| lognum\_voted\_users | 0.882847 | 0.123172 | 7.168 | 4.93E-12 | \*\*\* |  |  |
| num\_critic\_for\_reviews | -0.00317 | 0.000682 | -4.648 | 4.84E-06 | \*\*\* |  |  |
| logbudget | 0.597897 | 0.062404 | 9.581 | < 2e-16 | \*\*\* |  |  |
| lognum\_user\_for\_reviews | 0.147591 | 0.134108 | 1.101 | 0.27189 |  |  |  |
| imdb\_score | 0.023307 | 0.096751 | 0.241 | 0.80978 |  |  |  |
| actor\_1\_nameDenzel Washington | 0.657016 | 0.267544 | 2.456 | 0.01457 | \* |  |  |
| actor\_1\_nameHarrison Ford | 0.774605 | 0.280838 | 2.758 | 0.00613 | \*\* |  |  |
| actor\_1\_nameJ.K. Simmons | 0.439043 | 0.265523 | 1.654 | 0.09917 | . |  |  |
| actor\_1\_nameJason Statham | -0.55679 | 0.281165 | -1.98 | 0.04849 | \* |  |  |
| actor\_1\_nameJohnny Depp | -0.09782 | 0.254302 | -0.385 | 0.70073 |  |  |  |
| actor\_1\_nameLiam Neeson | 0.051378 | 0.276059 | 0.186 | 0.85247 |  |  |  |
| actor\_1\_nameMatt Damon | 0.142053 | 0.279337 | 0.509 | 0.61141 |  |  |  |
| actor\_1\_nameNicolas Cage | -0.09584 | 0.263541 | -0.364 | 0.71634 |  |  |  |
| actor\_1\_nameRobert De Niro | -0.01278 | 0.24486 | -0.052 | 0.95841 |  |  |  |
| actor\_1\_nameRobert Downey Jr. | 0.111416 | 0.282508 | 0.394 | 0.69355 |  |  |  |
| actor\_1\_nameRobin Williams | 0.314997 | 0.283612 | 1.111 | 0.26751 |  |  |  |
| content\_ratingPG | -0.19723 | 0.739032 | -0.267 | 0.78973 |  |  |  |
| content\_ratingPG-13 | -0.44302 | 0.736631 | -0.601 | 0.54797 |  |  |  |
| content\_ratingR | -0.89109 | 0.737744 | -1.208 | 0.22796 |  |  |  |
| content\_ratingX | 1.300392 | 1.053237 | 1.235 | 0.21782 |  |  |  |
|  |  |  |  |  |  |  |  |
| Residual standard error: 0.9982 on 334 degrees of freedom | | | | | | | |
| Multiple R-squared: 0.644, Adjusted R-squared: 0.6216 | | | | | | | |
| F-statistic: 28.77 on 21 and 334 DF, p-value: < 2.2e-16 | | | | | | | |

Table 21: Cross Validation and Out of Sample Comparisons for Actor's Analysis

|  |  |  |
| --- | --- | --- |
| Statistic \ Model | Full Model | Best Model |
| Adj-R2 | 0.556 | 0.558 |
| Res-SD | 1.082 | 1.079 |
| rmsepred(leave-one-out) | 1.103 | 1.097 |
| rmsepred(train / holdout 1) | 1.089 | 1.051 |
| rmsepred(train / holdout 2) | 1.082 | 1.214 |
| rmsepred(train / holdout 3) | 1.073 | 1.055 |

### Brief Discussion (Actor’s Analysis):

The model we found for actors is fine, there is no sign of curvilinear form. The adjusted R2 for the best model we choose is about 0.56, which is not very high, however the RMSE for train/holdout is really high, indicating that it may be hard to predict the gross of films using the data given and high variability is to be expected when doing an analysis within the film industry. One thing to be noticed, when fitting the best model selected by regsubsets, most of the variables we kept are not related to actors and only 3 of the actors shows statistical significance for the response variable.

## Conclusion:

Predicting movie box office success by gross revenue is extremely difficult since the monetary success of a movie depends on a great deal of factors that cannot be easily predicted upon. In this investigation models were found could fit the data reasonably well, but were not completely accurate because movie performance varies widely.

# Contributions

## Team Formation:

Initially, our team formed when Kelly, Sam and Luke decided to work on a project together about trying to predict some property related to a data set on movies. Then, after the deadline for project proposals had passed, Richard had submitted a proposal for the same data set that was selected by Kelly, Sam and Luke. Dr. Joe suggested that Richard join our team and thus our team of four formed.

## Authorship:

The names of the authors are listed in alphabetical order by surname.

## Teamwork and contributions:

Our team equally contributed to the work done for this project. Together our team decided on the topic: that we wanted to study the effect of variables in our data set on predicting the gross revenue of movies. Initially, all of our team members helped to sort out the data and make it useful since some of the columns of the data set were difficulty to work with. Once our data was in a form that could be more easily worked with, Kelly performed the regression analysis including the directors, Richard performed the regression analysis including the actors/actresses, Sam performed the regression analysis for only the numerical explanatory of the data set and Luke created the charts and tables for the summary statistics. All team members contributed to writing the report and creating it in its final form. As well, all team members contributed to the R code used in this project. Luke and Kelly were co-team leaders that ensured the group stayed on track during the course of the project. As well, during the course of the project Luke helped bring to the attention of the other team members that we needed to consider adjusting the currencies used in this analysis for inflation. All group members were involved with planning group meetings and booking rooms to meet in to work on this project.

# APPENDIX I - Director’s Analysis

**R-Script:**

#setwd("./Movie\_Project")

install.packages("leaps")

library(leaps)

dat = read.csv("data/moviedata\_sortedby\_year\_split\_genre.csv", stringsAsFactors = FALSE)

# omit bad data with na.omit

options(na.action=na.omit)

options(max.print=99999)

# Adjust for inflation

year <- dat$title\_year

Len  <- length(year)

for (i in 1:Len) {

  if ( dat$title\_year[i] >= 1915 & dat$title\_year[i] <= 1925) {

    dat$adjgross[i] <- dat$gross[i] \* 40

  } else if (dat$title\_year[i] > 1925 & dat$title\_year[i] <= 1930) {

    dat$adjgross[i] <- dat$gross[i] \* 35

  } else if (dat$title\_year[i] > 1930 & dat$title\_year[i] <= 1935) {

    dat$adjgross[i] <- dat$gross[i] \* 30

  } else if (dat$title\_year[i] > 1935 & dat$title\_year[i] <= 1940) {

    dat$adjgross[i] <- dat$gross[i] \* 25

  } else if (dat$title\_year[i] > 1940 & dat$title\_year[i] <= 1945) {

    dat$adjgross[i] <- dat$gross[i] \* 21.6

  } else if (dat$title\_year[i] > 1945 & dat$title\_year[i] <= 1950) {

    dat$adjgross[i] <- dat$gross[i] \* 14.5

  } else if (dat$title\_year[i] > 1950 & dat$title\_year[i] <= 1955) {

    dat$adjgross[i] <- dat$gross[i] \* 11.5

  } else if (dat$title\_year[i] > 1955 & dat$title\_year[i] <= 1960) {

    dat$adjgross[i] <- dat$gross[i] \* 10

  } else if (dat$title\_year[i] > 1960 & dat$title\_year[i] <= 1965) {

    dat$adjgross[i] <- dat$gross[i] \* 7.5

  } else if (dat$title\_year[i] > 1965 & dat$title\_year[i] <= 1970) {

    dat$adjgross[i] <- dat$gross[i] \* 5

  } else if (dat$title\_year[i] > 1970 & dat$title\_year[i] <= 1975) {

    dat$adjgross[i] <- dat$gross[i] \* 4

  } else if (dat$title\_year[i] > 1975 & dat$title\_year[i] <= 1980) {

    dat$adjgross[i] <- dat$gross[i] \* 3

  } else if (dat$title\_year[i] > 1980 & dat$title\_year[i] <= 1985) {

    dat$adjgross[i] <- dat$gross[i] \* 2.5

  } else if (dat$title\_year[i] > 1985 & dat$title\_year[i] <= 1990) {

    dat$adjgross[i] <- dat$gross[i] \* 2

  } else if (dat$title\_year[i] > 1990 & dat$title\_year[i] <= 1995) {

    dat$adjgross[i] <- dat$gross[i] \* 1.8

  } else if (dat$title\_year[i] > 1995 & dat$title\_year[i] <= 2000) {

    dat$adjgross[i] <- dat$gross[i] \* 1.5

  } else if (dat$title\_year[i] > 2000 & dat$title\_year[i] <= 2005) {

    dat$adjgross[i] <- dat$gross[i] \* 1.3

  } else if (dat$title\_year[i] > 2005 & dat$title\_year[i] <= 2010) {

    dat$adjgross[i] <- dat$gross[i] \* 1.1

  } else {

    dat$adjgross[i] <- dat$gross[i]

  }

}

# Use these lines to filter out by country and language

dat<-dat[(dat$country=="USA"| dat$country=="UK" | dat$country=="Canada" |

            dat$country=="New Zealand" | dat$country=="Australia" | dat$country=="Ireland" |

            (dat$country=="Germany" & dat$language=="English") | (dat$country=="France" & dat$language=="English") |

            (dat$country=="Italy" & dat$language=="English") | (dat$country=="Spain" & dat$language=="English")),]

# get rid of categorical variables and gross for this analysis

drops <- c("gross", "color", "director\_name", "actor\_2\_name", "genres", "primary\_genre", "actor\_1\_name", "movie\_title", "actor\_3\_name", "plot\_keywords", "movie\_imdb\_link", "language", "country", "content\_rating", "facenumber\_in\_poster", "title\_year", "aspect\_ratio")

dat <- dat[ , !(names(dat) %in% drops)]

# creating regression

dat$adjgross <- log(dat$adjgross/10000000)                          # scale adjusted gross by 10 million dollars and log

dat$num\_critic\_for\_reviews <- log(dat$num\_critic\_for\_reviews)       # log number of critic reviews

dat$director\_facebook\_likes <- log(dat$director\_facebook\_likes)     # log nunber of director likes

dat$actor\_3\_facebook\_likes <- log(dat$actor\_3\_facebook\_likes)       # log number of 3rd top billing actor likes

dat$actor\_1\_facebook\_likes <- log(dat$actor\_1\_facebook\_likes)       # log number of 1st top billing actor likes

dat$num\_voted\_users <- log(dat$num\_voted\_users)                     # log number of user votes

dat$cast\_total\_facebook\_likes <- log(dat$cast\_total\_facebook\_likes) # log number of total cast likes

dat$num\_user\_for\_reviews <- log(dat$num\_user\_for\_reviews)           # log number of user reviews

dat$budget <- log(dat$budget)                                       # log budget

dat$actor\_2\_facebook\_likes <- log(dat$actor\_2\_facebook\_likes)       # log number of 2nd top billing actor likes

dat$movie\_facebook\_likes <- log(dat$movie\_facebook\_likes)           # log number of facebook likes

dat <- do.call(data.frame,lapply(dat, function(x) replace(x, is.infinite(x),NA)))

dat <- na.omit(dat)

reg\_quant <- lm(adjgross ~., data = dat, na.action = na.omit)

s\_quant <- summary(reg\_quant)

# check which model works best with forward selection  (best fit is 8|10 variables)

s.fwd <- regsubsets(adjgross ~., data=dat, method="forward", nvmax = 20)

ss.fwd <- summary(s.fwd)

which.min(ss.fwd$cp)     # 8 variables for best model

which.max(ss.fwd$adjr2)  # 10 variables for best model

# check which model works best with backward selection   (best fit is 8|10 variables)

s.bwd <- regsubsets(adjgross ~., data=dat, method="backward", nvmax = 20)

ss.bwd <- summary(s.bwd)

which.min(ss.bwd$cp)     # 8 variables for best model

which.max(ss.bwd$adjr2)  # 10 variables for best model

# the 12 explanatory variables to make the best model as identified by forward/backward selection

keep <- c("adjgross", "num\_critic\_for\_reviews", "director\_facebook\_likes", "actor\_1\_facebook\_likes", "num\_voted\_users", "cast\_total\_facebook\_likes", "num\_user\_for\_reviews", "budget", "imdb\_score")

# generate a new regression for the best model

datBest <- dat[keep]

regBest <- lm(adjgross ~., data = datBest)

sBest <- summary(regBest)

ls.cvrmse <- function(ls.out)

  # Compute the leave-one-out cross-validated root mean squared error of prediction.

  # Handles missing values.

  # ls.out is a fitted regression model from lsfit or lm.

  # (c) Copyright William J. Welch 1997

{

  res.cv <- ls.out$residuals / (1.0 - ls.diag(ls.out)$hat)

  # Identify NA's and remove them.

  is.na.res <- is.na(res.cv)

  res.cv <- res.cv[!is.na.res]

  cvrmse <- sqrt(sum(res.cv^2) / length(res.cv))

  return(cvrmse)

}

n <- nrow(datBest)

set.seed(1)

# here select half of the data randomly.

index.subset1 <- sort(sample(1:n, round(n/2), replace = FALSE))

# Initially subset1 is the training set and subset2 is the hold-out set

quant.subset1 <- dat[index.subset1,]

quant.subset2 <- dat[-index.subset1,]

quantFull.subset1 <- lm(adjgross~., data = quant.subset1)

# Make predictions at the hold-out data set.

quantFull.pred1 <- predict(quantFull.subset1, quant.subset2)

quantFull.err1 <- sqrt(sum((quant.subset2$adjgross - quantFull.pred1)^2)/length(quantFull.pred1))

# Compare to the selected model.

quantBest.subset1 <- lm(adjgross~., data = datBest[index.subset1,])

quantBest.pred1 <- predict(quantBest.subset1, datBest[-index.subset1,])

quantBest.err1 <- sqrt(sum((datBest[-index.subset1,]$adjgross - quantBest.pred1)^2)/length(quantBest.pred1))

# ~~~~~~~~~~~~~~~~~~~~~ SUMMARY OF ANALYSIS ~~~~~~~~~~~~~~~~~~~~~ #

sBest                 # Summary statistics of the best model

sBest$adj.r.squared   # adjusted r squared of the best model

s\_quant$adj.r.squared # adjusted r squared of the full model

sBest$sigma           # residual SD of the best model

s\_quant$sigma         # residual SD of the full model

ls.cvrmse(regBest)    # leave-one-out cross validation of the best model

ls.cvrmse(reg\_quant)  # leave-one-out cross validation of the full model

quantBest.err1        # rmse prediction of the training/holdout sets of the best model

quantFull.err1        # rmse prediction of the training/holdout sets of the full model

# ~~~~~~~~~~~~~~~~~~~~~ DIRECTOR ANALYSIS ~~~~~~~~~~~~~~~~~~~~~ #

data = read.csv("moviedata\_sortedby\_year.csv", colClasses=c("genres"="character"))

# add adjgross to data

for (i in 1:Len) {

    if ( data$title\_year[i] >= 1915 & data$title\_year[i] <= 1925) {

        data$adjgross[i] <- data$gross[i] \* 40

    } else if (data$title\_year[i] > 1925 & data$title\_year[i] <= 1930) {

        data$adjgross[i] <- data$gross[i] \* 35

    } else if (data$title\_year[i] > 1930 & data$title\_year[i] <= 1935) {

        data$adjgross[i] <- data$gross[i] \* 30

    } else if (data$title\_year[i] > 1935 & data$title\_year[i] <= 1940) {

        data$adjgross[i] <- data$gross[i] \* 25

    } else if (data$title\_year[i] > 1940 & data$title\_year[i] <= 1945) {

        data$adjgross[i] <- data$gross[i] \* 21.6

    } else if (data$title\_year[i] > 1945 & data$title\_year[i] <= 1950) {

        data$adjgross[i] <- data$gross[i] \* 14.5

    } else if (data$title\_year[i] > 1950 & data$title\_year[i] <= 1955) {

        data$adjgross[i] <- data$gross[i] \* 11.5

    } else if (data$title\_year[i] > 1955 & data$title\_year[i] <= 1960) {

        data$adjgross[i] <- data$gross[i] \* 10

    } else if (data$title\_year[i] > 1960 & data$title\_year[i] <= 1965) {

        data$adjgross[i] <- data$gross[i] \* 7.5

    } else if (data$title\_year[i] > 1965 & data$title\_year[i] <= 1970) {

        data$adjgross[i] <- data$gross[i] \* 5

    } else if (data$title\_year[i] > 1970 & data$title\_year[i] <= 1975) {

        data$adjgross[i] <- data$gross[i] \* 4

    } else if (data$title\_year[i] > 1975 & data$title\_year[i] <= 1980) {

        data$adjgross[i] <- data$gross[i] \* 3

    } else if (data$title\_year[i] > 1980 & data$title\_year[i] <= 1985) {

        data$adjgross[i] <- data$gross[i] \* 2.5

    } else if (data$title\_year[i] > 1985 & data$title\_year[i] <= 1990) {

        data$adjgross[i] <- data$gross[i] \* 2

    } else if (data$title\_year[i] > 1990 & data$title\_year[i] <= 1995) {

        data$adjgross[i] <- data$gross[i] \* 1.8

    } else if (data$title\_year[i] > 1995 & data$title\_year[i] <= 2000) {

        data$adjgross[i] <- data$gross[i] \* 1.5

    } else if (data$title\_year[i] > 2000 & data$title\_year[i] <= 2005) {

        data$adjgross[i] <- data$gross[i] \* 1.3

    } else if (data$title\_year[i] > 2005 & data$title\_year[i] <= 2010) {

        data$adjgross[i] <- data$gross[i] \* 1.1

    } else {

        data$adjgross[i] <- data$gross[i]

    }

}

directors <- subset(data, table(data$director\_name)[data$director\_name] >= 10)

directors$lngross <- log(directors$gross)

directors$lnadjgross <- log(directors$adjgross)

directors$lndirector\_facebook\_likes <- log(directors$director\_facebook\_likes)

directors$lnactor\_1\_facebook\_likes <- log(directors$actor\_1\_facebook\_likes)

directors$lnnum\_voted\_users <- log(directors$num\_voted\_users)

directors$lncast\_total\_facebook\_likes <- log(directors$cast\_total\_facebook\_likes)

directors$lnbudget <- log(directors$budget)

directors$lnnum\_users\_for\_review <- log(directors$num\_user\_for\_reviews)

directors$lnmovie\_facebook\_likes <- log(directors$movie\_facebook\_likes)

# cleanup bad data

directors$lnbudget[is.infinite(directors$lnbudget)] <- NA

directors$lnadjgross[is.infinite(directors$adjgross)] <- NA

directors$lnmovie\_facebook\_likes[is.infinite(directors$lnmovie\_facebook\_likes)] <- NA

directors$lngross[is.infinite(directors$lngross)] <- NA

directors$lnnum\_voted\_users[is.infinite(directors$lnnum\_voted\_users)] <- NA

directors$lncast\_total\_facebook\_likes[is.infinite(directors$lncast\_total\_facebook\_likes)] <- NA

directors$lndirector\_facebook\_likes[is.infinite(directors$lndirector\_facebook\_likes)] <- NA

directors$lncast\_total\_facebook\_likes[is.infinite(directors$lncast\_total\_facebook\_likes)] <- NA

directors$lnactor\_1\_facebook\_likes[is.infinite(directors$lnactor\_1\_facebook\_likes)] <- NA

directors$lnbudget[is.nan(directors$lnbudget)] <- NA

directors$lnadjgross[is.nan(directors$adjgross)] <- NA

directors$lnmovie\_facebook\_likes[is.nan(directors$lnmovie\_facebook\_likes)] <- NA

directors$lngross[is.nan(directors$lngross)] <- NA

directors$lnnum\_voted\_users[is.nan(directors$lnnum\_voted\_users)] <- NA

directors$lncast\_total\_facebook\_likes[is.nan(directors$lncast\_total\_facebook\_likes)] <- NA

directors$lndirector\_facebook\_likes[is.nan(directors$lndirector\_facebook\_likes)] <- NA

directors$lncast\_total\_facebook\_likes[is.nan(directors$lncast\_total\_facebook\_likes)] <- NA

directors$lnactor\_1\_facebook\_likes[is.nan(directors$lnactor\_1\_facebook\_likes)] <- NA

# create regression model

directors <- na.omit(directors)

fit <- lm(lnadjgross ~ lncast\_total\_facebook\_likes + lnactor\_1\_facebook\_likes + lndirector\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score + content\_rating + duration + director\_name, data = directors)

# install leaps package and run backward/forward selection

install.packages("leaps")

library(leaps)

# remove categorical variables as not compatible with variable selection algorithm

fitbackward <- regsubsets(lnadjgross ~ lncast\_total\_facebook\_likes + lnactor\_1\_facebook\_likes + lndirector\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score + duration, data = directors, method = "backward")

fitforward <- regsubsets(lnadjgross ~ lncast\_total\_facebook\_likes + lnactor\_1\_facebook\_likes + lndirector\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score + duration, data = directors, method = "backward")

# this process can be automated below to get the appropriate statistics for comparison

fitbackward$summary <- summary(fitbackward) # to find cp and adjr2

fitforward$summary <- summary(fitforward) # to find cp and adjr2

maxadjr2fitbackward <- which.max(fitbackward$summary$adjr2) #model with max adjr2 value

minfitforwardcp <- which.min(fitforward$summary$cp) #model with min cp value

# calculate leave-one-out CV error

ls.cvrmse <- function(ls.out)

# Compute the leave-one-out cross-validated root mean squared error of prediction.

# Handles missing values.

# ls.out is a fitted regression model from lsfit or lm.

# (c) Copyright William J. Welch 1997

{

    res.cv <- ls.out$residuals / (1.0 - ls.diag(ls.out)$hat)

    # Identify NA's and remove them.

    is.na.res <- is.na(res.cv)

    res.cv <- res.cv[!is.na.res]

    cvrmse <- sqrt(sum(res.cv^2) / length(res.cv))

    return(cvrmse)

}

# Compare the full model and best model found by regsubsets, categoricals removed

fit <- lm(lnadjgross ~ lncast\_total\_facebook\_likes + lnactor\_1\_facebook\_likes + lndirector\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score + content\_rating + duration + director\_name, data = directors)

#full model

# remove director\_facebook\_likes and duration due to variable selection process, categoricals removed

fit2 <- lm(lnadjgross ~ lnactor\_1\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score + content\_rating + director\_name, data = directors)# removed num\_users\_for\_review, but usually you use your best model against your full model# find best model using forward, backward or exhaustive methods

summary(fit2)

# Compare the full model and best model found by regsubsets, categoricals removed

fitA <- lm(lnadjgross ~ lncast\_total\_facebook\_likes + lnactor\_1\_facebook\_likes + lndirector\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score + duration, data = directors)

#full model

# remove director\_facebook\_likes and duration due to variable selection process, categoricals removed

fit2A <- lm(lnadjgross ~ lnactor\_1\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score, data = directors)# removed num\_users\_for\_review, but usually you use your best model against your full model# find best model using forward, backward or exhaustive methods

summary(fit2)

# Calculate the leave-one-out CV RMSE for the full model, categoricals removed

fitA.cvrmse <- ls.cvrmse(fitA)

# Calculate the leave-one-out CV RMSE for the best model via regsubsets, categoricals removed

fit2A.cvrmse <- ls.cvrmse(fit2A)

print(c(fitA.cvrmse, fit2A.cvrmse))

# The best model has smaller cvrmse

# fit2 has a smaller cvrmse which is surprising as it has a smaller adj-R2.  But maybe it's due to the NaNs produced

# Two-fold CV, using the Lab8 Technique

n <- nrow(directors)

set.seed(1)

id.subset1 <- sort(sample(1:n, round(n/2), replace = FALSE))

# randomly select holdout data

directors.subset1 <- directors[id.subset1,]

directors.subset2 <- directors[-id.subset1,]

fit.subset1 <- lm(lnadjgross ~ lncast\_total\_facebook\_likes + lnactor\_1\_facebook\_likes + lndirector\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score + content\_rating + duration + director\_name, data = directors.subset1)

# calculate a prediction based on the holdout set, using the full model

fit.pred1 <- predict(fit.subset1, data = directors.subset2)

# calculate the error

fit.err1 <- sqrt(sum((directors.subset2$lnadjgross - fit.pred1)^2)/length(fit.pred1))

# calculate a prediction based on the 2nd fit model

fit2.subset1 <- lm(lnadjgross ~ lnactor\_1\_facebook\_likes + lnnum\_voted\_users + lnnum\_users\_for\_review + lnbudget + imdb\_score + content\_rating + director\_name, data = directors.subset1)

fit2.pred1 <- predict(fit2.subset1, directors.subset2)

fit2.err1 <- sqrt(sum((directors.subset2$lnadjgross - fit2.pred1)^2)/length(fit2.pred1))

fit2.err1

# ~~~~~~~~~~~~~~~~~~~~~ SUMMARY OF ANALYSIS ~~~~~~~~~~~~~~~~~~~~~ #

fullModel <- summary(fit)

secondBest <- summary(fit2)

print(c(fitA.cvrmse, fit2A.cvrmse))

# calculate the error

fit.err1

fit2.err1

# ~~~~~~~~~~~~~~~~~~~~~ ACTOR'S ANALYSIS ~~~~~~~~~~~~~~~~~~~~~ #

#first clean up the data

library(readr)

dat <- read\_csv("~/R/movie\_metadata.csv/moviedata\_sortedby\_year.csv")

data<-dat[(dat$country=="USA"| dat$country=="UK" | dat$country=="Canada" |

             dat$country=="New Zealand" | dat$country=="Australia" | dat$country=="Ireland" |

             (dat$country=="Germany" & dat$language=="English") | (dat$country=="France" & dat$language=="English") |

             (dat$country=="Italy" & dat$language=="English") | (dat$country=="Spain" & dat$language=="English")),]

setwd(".")    # <--- trying to avoid direct path as we all have different setups

#install.packages('plyr')

#install.packages("stringr", repos='<http://cran.us.r-project.org')>

library(stringr)

library(plyr)

#read data from csv file sorted by year, with primary genres selected

options(na.action=na.omit)

options(max.print=99999)

#produces frewuency table output for content rating as seen on page 110:

count(data,"content\_rating")

#create a vector that includes genre in the form "Action|Comedy|etc." for use in a function used to count frequency of genres

genrevector <- data$genres

countAction = 0; countAdventure = 0; countAnimation = 0; countBiography = 0; countComedy = 0;

countCrime = 0; countDocumentary = 0; countDrama = 0; countFamily = 0;

countFantasy = 0; countFilmNoir = 0; countGameShow = 0; countWar=0;

countHistory = 0; countHorror = 0; countMusic = 0; countMusical = 0; countMystery = 0;

countRomance = 0; countSciFi = 0; countThriller = 0; countWestern = 0; countSport = 0; countNews = 0; countRealityTV = 0;

#code to count frequency of genres

for (i in genrevector ) {

  if (grepl(i, "Action"))

    countAction = countAction + 1;

  if (grepl(i, "Adventure"))

    countAdventure = countAdventure + 1;

  if (grepl(i, "Animation"))

    countAnimation = countAnimation + 1;

  if (grepl(i, "Biography"))

    countBiography = countBiography + 1;

  if (grepl(i, "Comedy"))

    countComedy = countComedy + 1;

  if (grepl(i, "Crime"))

    countCrime = countCrime + 1;

  if (grepl(i, "Documentary"))

    countDocumentary = countDocumentary + 1;

  if (grepl(i, "Drama"))

    countDrama = countDrama + 1;

  if (grepl(i, "Family"))

    countFamily = countFamily + 1;

  if (grepl(i, "Fantasy"))

    countFantasy = countFantasy + 1;

  if (grepl(i, "Film-Noir"))

    countFilmNoir = countFilmNoir + 1;

  if (grepl(i, "Game-Show"))

    countGameShow = countGameShow + 1;

  if (grepl(i, "History"))

    countHistory = countHistory + 1;

  if (grepl(i, "Horror"))

    countHorror = countHorror + 1;

  if (grepl(i, "Music"))

    countMusic = countMusic + 1;

  if (grepl(i, "Musical"))

    countMusical = countMusical + 1;

  if (grepl(i, "Mystery"))

    countMystery = countMystery + 1;

  if (grepl(i, "Romance"))

    countRomance = countRomance + 1;

  if (grepl(i, "Sci-Fi"))

    countSciFi = countSciFi + 1;

  if (grepl(i, "Thriller"))

    countThriller = countThriller + 1;

  if (grepl(i, "Western"))

    countWestern = countWestern + 1;

  if (grepl(i, "War"))

    countWar = countWar + 1;

  if (grepl(i, "Sport"))

    countSport = countSport + 1;

  if (grepl(i, "News"))

    countNews = countNews + 1;

  if (grepl(i, "Reality-TV"))

    countRealityTV = countRealityTV + 1;

}

countAction; countAdventure; countAnimation; countBiography; countComedy ;

countCrime ; countDocumentary ; countDrama ; countFamily ;

countFantasy ; countFilmNoir ; countGameShow ;

countHistory ; countHorror ; countMusic ; countMusical ; countMystery ;

countRomance ; countSciFi ; countThriller ; countWestern ; countWar; countSport; countNews; countRealityTV;

# Create binary vector for each genre

# Make vectors for:

#   - Action, Adventure, Animation, Biography, Comedy, Crime, Drama, Family, Fantasy, History, Horror, Music, Musical, Mystery, Romance, Sci-fi, Thriller, War

# Given a genre and a the genres of a movie, check if a movie contains it, if it does return 1, 0 otherwise

hasGenre <- function(genre, movieGenres) {

  regExp <- paste("(^|\\|)", genre, "(\\||$)", sep = "")

  if(grepl(regExp, movieGenres)){

    return(1)

  } else {

    return(0)

  }

}

# Given a genre and a vector of the genres of various movies, produce a vector of TRUE/FALSE for movies that contain/don't contain the given genre

moviesHaveGenre <- function(genre\_, genreVector) {

  return(sapply(genreVector, hasGenre, genre = genre\_, USE.NAMES = FALSE))

}

commonGenres <- c("Action", "Adventure", "Animation", "Biography", "Comedy", "Crime", "Drama", "Family", "Fantasy", "History", "Horror", "Music", "Musical", "Mystery", "Romance", "Sci-Fi", "Thriller", "War")

# Given a genre, create vector of movies that have or do not have the genre (1|0), and cbind to the dataframe

for (genre in commonGenres) {

  gVector <- moviesHaveGenre(genre, data$genres)

  data[genre] <- gVector

}

dat<-data.frame(data)

dat<-na.omit(dat)

year <- dat$title\_year

Len  <- length(year)

for (i in 1:Len) {

  if ( dat$title\_year[i] >= 1915 & dat$title\_year[i] <= 1925) {

    dat$adjgross[i] <- dat$gross[i] \* 40

  } else if (dat$title\_year[i] > 1925 & dat$title\_year[i] <= 1930) {

    dat$adjgross[i] <- dat$gross[i] \* 35

  } else if (dat$title\_year[i] > 1930 & dat$title\_year[i] <= 1935) {

    dat$adjgross[i] <- dat$gross[i] \* 30

  } else if (dat$title\_year[i] > 1935 & dat$title\_year[i] <= 1940) {

    dat$adjgross[i] <- dat$gross[i] \* 25

  } else if (dat$title\_year[i] > 1940 & dat$title\_year[i] <= 1945) {

    dat$adjgross[i] <- dat$gross[i] \* 21.6

  } else if (dat$title\_year[i] > 1945 & dat$title\_year[i] <= 1950) {

    dat$adjgross[i] <- dat$gross[i] \* 14.5

  } else if (dat$title\_year[i] > 1950 & dat$title\_year[i] <= 1955) {

    dat$adjgross[i] <- dat$gross[i] \* 11.5

  } else if (dat$title\_year[i] > 1955 & dat$title\_year[i] <= 1960) {

    dat$adjgross[i] <- dat$gross[i] \* 10

  } else if (dat$title\_year[i] > 1960 & dat$title\_year[i] <= 1965) {

    dat$adjgross[i] <- dat$gross[i] \* 7.5

  } else if (dat$title\_year[i] > 1965 & dat$title\_year[i] <= 1970) {

    dat$adjgross[i] <- dat$gross[i] \* 5

  } else if (dat$title\_year[i] > 1970 & dat$title\_year[i] <= 1975) {

    dat$adjgross[i] <- dat$gross[i] \* 4

  } else if (dat$title\_year[i] > 1975 & dat$title\_year[i] <= 1980) {

    dat$adjgross[i] <- dat$gross[i] \* 3

  } else if (dat$title\_year[i] > 1980 & dat$title\_year[i] <= 1985) {

    dat$adjgross[i] <- dat$gross[i] \* 2.5

  } else if (dat$title\_year[i] > 1985 & dat$title\_year[i] <= 1990) {

    dat$adjgross[i] <- dat$gross[i] \* 2

  } else if (dat$title\_year[i] > 1990 & dat$title\_year[i] <= 1995) {

    dat$adjgross[i] <- dat$gross[i] \* 1.8

  } else if (dat$title\_year[i] > 1995 & dat$title\_year[i] <= 2000) {

    dat$adjgross[i] <- dat$gross[i] \* 1.5

  } else if (dat$title\_year[i] > 2000 & dat$title\_year[i] <= 2005) {

    dat$adjgross[i] <- dat$gross[i] \* 1.3

  } else if (dat$title\_year[i] > 2005 & dat$title\_year[i] <= 2010) {

    dat$adjgross[i] <- dat$gross[i] \* 1.1

  } else {

    dat$adjgross[i] <- dat$gross[i]

  }

}

# data cleaned

movie\_actor<-na.omit(dat)

#only picks movies data with main actor who shows up in more than 20 movies

ac1<-subset(movie\_actor, table(actor\_1\_name)[actor\_1\_name] >= 25)

ac1<-na.omit(ac1)

#take log for big data

ac1$logadjgross<-log(ac1$adjgross)

ac1$logactor\_1\_facebook\_likes<-log(ac1$actor\_1\_facebook\_likes)

ac1$logcast\_total\_facebook\_likes<-log(ac1$cast\_total\_facebook\_likes)

ac1$lognum\_voted\_users<-log(ac1$num\_voted\_users)

ac1$logbudget<-log(ac1$budget)

ac1$lognum\_user\_for\_reviews<-log(ac1$num\_user\_for\_reviews)

ac1$logdirector\_facebook\_likes<-log(ac1$director\_facebook\_likes)

# fit full model with actor\_name as catagory variables

fit <- lm(logadjgross ~logactor\_1\_facebook\_likes+logcast\_total\_facebook\_likes

          +lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score+actor\_1\_name,data=ac1)

s1<-summary(fit)

par(mfrow=c(2,4))

plot(ac1$logbudget,s1$residuals,xlab="logbudget",ylab="residual")

plot(ac1$imdb\_score,s1$residuals,xlab="imdb\_score",ylab="residual")

plot(ac1$num\_critic\_for\_reviews,s1$residuals,xlab="num\_critic\_for\_reviews",ylab="residual")

plot(ac1$lognum\_voted\_users,s1$residuals,xlab="lognum\_voted\_users",ylab="residual")

plot(ac1$lognum\_user\_for\_reviews,s1$residuals,xlab="lognum\_user\_for\_reviews",ylab="residual")

plot(ac1$logactor\_1\_facebook\_likes,s1$residuals,xlab="logactor\_1\_facebook\_likes",ylab="residual")

plot(ac1$logcast\_total\_facebook\_likes,s1$residuals,xlab="logcast\_total\_facebook\_likes",ylab="residual")

#box plot

par(mfrow=c(1,1))

boxplot(ac1$logadjgross ~ ac1$actor\_1\_name, main="actor\_1\_name", xlab = "actor\_1\_name", ylab = "ln(Gross Revenue, USD)")

# with movie type, but doesn't make much sense

fit2 <- lm(logadjgross ~num\_critic\_for\_reviews+logcast\_total\_facebook\_likes + logactor\_1\_facebook\_likes

           + lognum\_voted\_users + lognum\_user\_for\_reviews + logbudget +

             imdb\_score + content\_rating + Action + Adventure + Animation +

             Biography + Comedy + Crime + Drama + Family + Fantasy + History + Horror + Music +

             Musical + Mystery + Romance + Thriller + War, data = ac1)

s2<-summary(fit2)

#install.packages("leaps")

library(leaps)

#forward and backward selection for full model without catagory variable

fitbackward <- regsubsets(logadjgross ~logactor\_1\_facebook\_likes+logcast\_total\_facebook\_likes+duration

+lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score, data = ac1, method = "backward")

fitforward <- regsubsets(logadjgross ~logactor\_1\_facebook\_likes+logcast\_total\_facebook\_likes+duration

+lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score, data = ac1, method = "forward")

fitbackward$summary <- summary(fitbackward)

fitforward$summary <- summary(fitforward)

#check adj and cp to choose the best model, adj suggests that full model is the best, but cp suggests model with four variable is better

#fit both models suggested by adj and cp value

which.max(fitbackward$summary$adjr2)

which.min(fitbackward$summary$cp)

which.max(fitforward$summary$adjr2)

which.min(fitforward$summary$cp)

fitbest<-lm(logadjgross ~duration +lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score, data = ac1)

fitfull <- lm(logadjgross ~logactor\_1\_facebook\_likes+logcast\_total\_facebook\_likes+duration

+lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score, data = ac1)

#summary

summaryfull<-summary(fitfull)

summarybest<-summary(fitbest)

ls.cvrmse <- function(ls.out)

{

    res.cv <- ls.out$residuals / (1.0 - ls.diag(ls.out)$hat)

    # Identify NA's and remove them.

    is.na.res <- is.na(res.cv)

    res.cv <- res.cv[!is.na.res]

    cvrmse <- sqrt(sum(res.cv^2) / length(res.cv))

    return(cvrmse)

}

cat("\nCross-validation root mean square error using ls.cvrmse function\n")

cvrmsebest<-ls.cvrmse(fitbest); cvrmsefull<-ls.cvrmse(fitfull)

cat(cvrmsebest,cvrmsefull,"\n")

#For each of the 2 models, we do 3 random holdout prediction to get the

#Cross-validation root mean square error with holdout,here we add

#back the catagory variables (actor\_1\_name)

# for the full model

n <- nrow(ac1)

set.seed(1)

id.subset1 <- sort(sample(1:n, round(n/2), replace = FALSE))

# randomly select holdout data

ac1.subset1 <- ac1[id.subset1,]

ac1.subset2 <- ac1[-id.subset1,]

#here we add back the category variables

fit.subseta <- lm(logadjgross ~logactor\_1\_facebook\_likes+logcast\_total\_facebook\_likes+duration

+lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score+actor\_1\_name, data = ac1.subset1)

# calculate a prediction based on the holdout set, using the full model

fit.preda <- predict(fit.subseta, ac1.subset2)

# calculate the error

fit.erra <- sqrt(sum((ac1.subset2$logadjgross - fit.preda)^2)/length(fit.preda))

fit.erra

#repeat for 3 times

id.subset3 <- sort(sample(1:n, round(n/2), replace = FALSE))

ac1.subset3 <- ac1[id.subset3,]

ac1.subset4 <- ac1[-id.subset3,]

fit.subsetb <- lm(logadjgross ~logactor\_1\_facebook\_likes+logcast\_total\_facebook\_likes+duration

+lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score+actor\_1\_name, data = ac1.subset3)

fit.predb <- predict(fit.subsetb, ac1.subset4)

# calculate the error

fit.errb <- sqrt(sum((ac1.subset4$logadjgross - fit.predb)^2)/length(fit.predb))

fit.errb

#third time

id.subset5 <- sort(sample(1:n, round(n/2), replace = FALSE))

ac1.subset5 <- ac1[id.subset5,]

ac1.subset6 <- ac1[-id.subset5,]

fit.subsetc <- lm(logadjgross ~logactor\_1\_facebook\_likes+logcast\_total\_facebook\_likes+duration

+lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score+actor\_1\_name, data = ac1.subset5)

fit.predc <- predict(fit.subsetc, ac1.subset6)

fit.errc <- sqrt(sum((ac1.subset6$logadjgross - fit.predc)^2)/length(fit.predc))

fit.errc

# do this again for 6 variable model and the category variables

id.subset1 <- sort(sample(1:n, round(n/2), replace = FALSE))

# randomly select holdout data

ac1.subset1 <- ac1[id.subset1,]

ac1.subset2 <- ac1[-id.subset1,]

#here we add back the category variables

fit.subseta <- lm(logadjgross ~duration +lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score+actor\_1\_name, data = ac1.subset1)

# calculate a prediction based on the holdout set, using the full model

fit.preda <- predict(fit.subseta, ac1.subset2)

# calculate the error

fit.errd <- sqrt(sum((ac1.subset2$logadjgross - fit.preda)^2)/length(fit.preda))

fit.errd

#repeat for 3 times

id.subset3 <- sort(sample(1:n, round(n/2), replace = FALSE))

ac1.subset3 <- ac1[id.subset3,]

ac1.subset4 <- ac1[-id.subset3,]

fit.subsetb <- lm(logadjgross ~duration +lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score+actor\_1\_name, data = ac1.subset3)

fit.predb <- predict(fit.subsetb, ac1.subset4)

# calculate the error

fit.erre <- sqrt(sum((ac1.subset4$logadjgross - fit.predb)^2)/length(fit.predb))

fit.erre

#third time

id.subset5 <- sort(sample(1:n, round(n/2), replace = FALSE))

ac1.subset5 <- ac1[id.subset5,]

ac1.subset6 <- ac1[-id.subset5,]

fit.subsetc <- lm(logadjgross ~duration +lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score+actor\_1\_name, data = ac1.subset5)

fit.predc <- predict(fit.subsetc, ac1.subset6)

fit.errf <- sqrt(sum((ac1.subset6$logadjgross - fit.predc)^2)/length(fit.predc))

fit.errf

#get the residual SD and adjusted R2

SD1<-summaryfull$sigma

SD2<-summarybest$sigma

adj1<-summaryfull$adj.r.squared

adj2<-summarybest$adj.r.squared

# build table

trial <- matrix(c(adj1,SD1,cvrmsefull,fit.erra,fit.errb,fit.errc,adj2,SD2,cvrmsebest,fit.errd,fit.erre,fit.errf), ncol=2)

colnames(trial) <- c("full model", "fit by cp")

rownames(trial) <- c("adjusted R2","residual SD","rmsepred(leave-one-out)","rmsepred(train/holdout) 1",

"rmsepred(train/holdout) 2","rmsepred(train/holdout) 3")

trial.table <- as.table(trial)

trial.table

# Model with 4 explanatory variables is better based on leave-one-out

fitbest2<-lm(logadjgross ~duration +lognum\_voted\_users+num\_critic\_for\_reviews+logbudget+lognum\_user\_for\_reviews+imdb\_score+actor\_1\_name+content\_rating, data = ac1)

summary(fitbest2)

ac2<-ac1[,c(3,4,26,48,49,50,51,52,53)]

summary(ac2)

# ~~~~~~~~~~~~~~~~~~~~~ SUMMARY OF ANALYSIS ~~~~~~~~~~~~~~~~~~~~~ #

summaryfull #summary(fitfull)

summarybest #summary(fitbest)

SD1 #summaryfull$sigma

SD2 #summarybest$sigma

adj1 #summaryfull$adj.r.squared

adj2 #summarybest$adj.r.squared

cvrmsebest

# calculate the error, training holdout

fit.erra

fit.errb

fit.errc

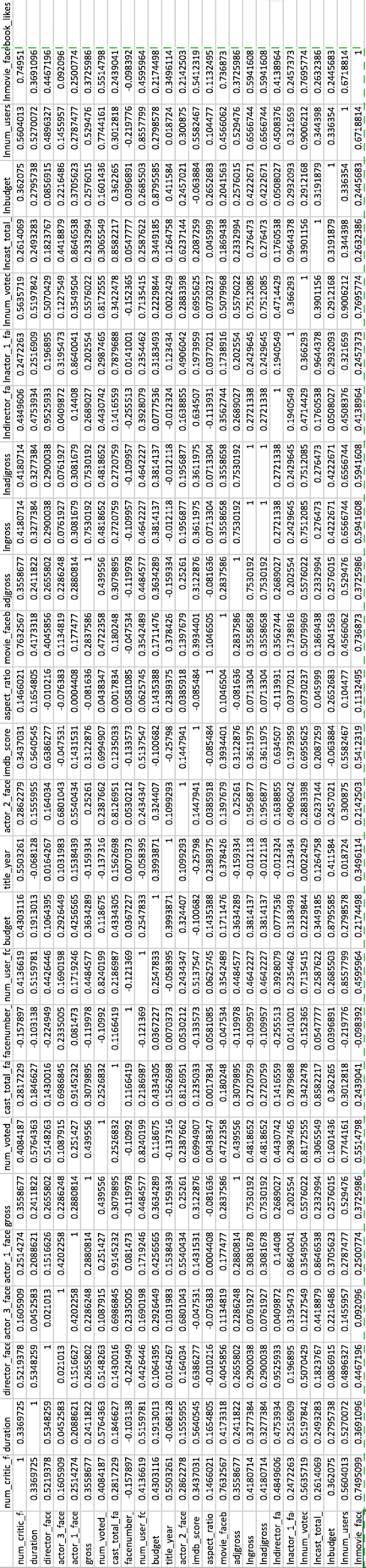
fit.errd

fit.erre

fit.errf

## 

# APPENDIX II - Sample Corr Matrix, Director’s Analysis



## 

# APPENDIX III – Inflation Rate Table

Inflation Rate (estimated, Box Office Mojo):

1940 - 1945: 21.6

1945 - 1950: 14.5

1950 - 1955: 11.5

1955 - 1960: 10

1960 - 1965: 7.5

1965 - 1970: 5

1970 - 1975: 4

1975 - 1980: 3

1980 - 1985: 2.5

1985 - 1990: 2

1990 - 1995: 1.8

1995 - 2000: 1.5

2000 - 2005: 1.3

2005 - 2010: 1.1

# APPENDIX IV – Sample Correlation Matrix – Actor’s Analysis

