Dummy variables:

Collection:

numCollection = (Collection = “Yes”)

when numCollection =1 then the increase is the amount that you have more than if it was no collection at all.

Original\_language:

1 means “English”

0 means “not English”

Production Countries:

1 means USA

0 means not USA

Status

numStatus1 = (Status = ‘Released’)

numStatus2 = (Status = ‘Rumored’)

So when numstatus1 = 1 then it is the amount increased over “Post=production”

Production Companies:

“Other” is the base

numProd1 = (Production Companies = ‘Universal Studios’)

numProd2 = (Production Companies = ‘New Line Cinema’)

numProd3 = (Production Companies = ‘Walt Disney Pictures’)

numProd4 = (Production Companies = ‘Warner Bros.’)

numProd5 = (Production Companies = ‘Metro-Goldwyn-Mayer (MGM)’)

numProd6 = (Production Companies = ‘United Artists’)

numProd7 = (Production Companies = ‘Columbia Pictures Corporation’)

numProd8 = (Production Companies = ‘Miramax Films’)

numProd9 = (Production Companies = ‘Paramount Pictures’)

numProd10 = (Production Companies = ‘Twentieth Century Fox Film Corporation’)

numProd11 = (Production Companies = ‘Columbia Pictures’)

Genre:

\*Genre Action is the base genre

numGenre1 = (Genre = 'Adventure');

numGenre2 = (Genre = 'Animation');

numGenre3 = (Genre = 'Comedy');

numGenre4 = (Genre = 'Crime');

numGenre5 = (Genre = 'Documentary');

numGenre6 = (Genre = 'Drama');

numGenre7 = (Genre = 'Family');

numGenre8 = (Genre = 'Fantasy');

numGenre9 = (Genre = 'Foreign');

numGenre10 = (Genre = 'History');

numGenre11 = (Genre = 'Horror');

numGenre12 = (Genre = 'Music');

numGenre13 = (Genre = 'Other');

numGenre14 = (Genre = 'Romance');

numGenre15 = (Genre = 'Science Fiction');

numGenre16 = (Genre = 'Thriller');

numGenre17 = (Genre = 'TV Movie');

numGenre18 = (Genre = 'War');

numGenre19 = (Genre = 'Western');

Release\_Year:

Base year is 1900-1950

numYear1 = (Release\_Year = '1951-2000');

numYear2 = (Release\_Year = '2001-2010');

numYear3 = (Release\_Year = '2011-2018')

# Methodology

I obtained the data from Kaggle. First I went through the data to see which variables could be cleaned up or consolidated. I started by re-coding the Collection variable to be 0 or 1, 1 meaning the movie is part of a collection, 0 meaning the movies is not part of a collection. I then added an original\_lang variable which also consists of 0 and 1, 1 being English, 0 being not-English. I changed the production countries variable to be 1 (US) and 0(not-US), and I added a Release\_Year variable which split the release years into 3 categories. Lastly, I cleaned up the Production Companies variable to include the top 11 production companies and the rest are labeled as “other”.

My model approach was to first fit the model with all variables using popularity as my y-variable and created numerous dummy variables for the text variables. I started by performing an exploratory analysis on the variables with a scatterplot matrix and histogram of the popularity variable, similar to the analysis performed by Jeff Spoelstra on movie rating(1). I then transformed the y-variable by taking log(popularity) and sqrt(popularity) to get a better model. I used two selection models, stepwise and backward, to come up with the best model using the transformed y-variable. I took the final model and validated it by checking the model assumptions, goodness of fit variables, and outliers. Lastly, I split the data into test and train sets and performed validation techniques on the model.

# Exploratory Analysis

A scatterplot matrix and pearson correlation coefficients were created to see relationships between variables, shown in Figure E.1 and Figure E.21. The pearson correlation coefficient matrix shows that vote\_count and revenue have a slight correlation. The histograms of popularity, ln(popularity) and sqrt(popularity) are shown in Figure E.2, E.3, and E.4 respectively. You can see by the histogram that the popularity variable is not normal, but skewed left. Since the histogram is very skewed, I chose to transform the data in two ways and see their distribution. The histogram of ln(popularity) is slightly more normal but skewed left while the sqrt(popularity) histogram is normal skewed right. The skew might be due to some outliers which will be investigated after modeling.

# Interaction Variables

I created an interaction variable which combined original language and vote average. I thought these two variables make sense to combine since I would think the original language of the movie could have an effect on the vote average. However, when I ran the regression analysis, the interaction variable was not significant, based on a p-value of 0.8603. Since the new interaction variable was insignificant I decided to exclude it from my regression analysis. The results of this analysis are shown in Figure E.22

# Regression Analysis without transformation

The first regression analysis I used popularity as the y-variable. This produced the output included in figure E.5. The R-Square value for this model was 0.2989 and the Adj R-Sq was 0.2956, meaning that about 29% of the variability in the data could be explained by the x-variables. The model had a fairly high f-value meaning the model was a good fit. However, there were a few variables there were insignificant which could be removed and the model re-run. Before running a model selection, I transformed the data to see if I could get a better model.

# Transform Data

I transformed the data using log(popularity) and sqrt(popularity). I ran full regression models for both transformed values. The log(popularity) transformation, seen in Figure E.6, resulted in a R-Square value of 0.4625 and Adj-R2 value of 0.4237. This means that the ln(popularity) model was slightly better than the un-transformed y-variable, and that 46% of the variability of the data can be explained by the x-variables. The goodness of fit value, F-value, was 156.55. This is also higher than the un-transformed y-variable, concluding that this model is a better fit.

The results for the sqrt(popularity) transformation are shown in Figure E.5. This model produced an R-square value of 0.4907 and Adj-R2 value of 0.4883. This means that 49% of the variability in the model can be explained by the x-variables. The goodness of fit variable, f-value, was 202. This f-value was the highest of the three models, meaning that it has the best goodness of fit.

# Further analysis of Sqrt(popularity) transformation

Since the sqrt(popularity) produced the best model, I decided to explore it further. I decided to use both the stepwise and backward model selection to produce a model which removes in-significant x-variables. The results of the stepwise and backward model selections are shown in figures E.8 and E.9. The stepwise model included 28 variables while the backward model included 27 variables.

I next ran the regression model with the variables given in the backward model with vif option to check for multi-collinearity. I noticed that there were still some insignificant variables using alpha = 0.05, so I removed these and ran the model again. This model, Figure E.10, produced a R-Square value of 0.4891 and Adj-R2 value of 0.4878. The model also showed that numYear1 and numYear2 were multi-collinear, since their variance inflation values were both over 9.0. I removed these variable and ran the model again, however the R-square value did not change significantly and I decided to keep both variables in the final model.

I ran the model using the stb option to determine which variables were the strongest predictors of the model. The variables, vote\_count, vote\_average, and numYear3 have the strongest influence on the popularity of the movie. These variables have the highest standardized estimate values, which are shown in Figure E.10.

# Model Assumptions

I looked at the residual plots, (Student \* predicted) and (npp\*predicted), to check the model assumptions. The plots shown in Figure E.11 show that the model violates the constant variance and independent assumptions while the normal plot, shown in Figure E.12, shows that the model is normal and linear.

# Outliers and Influential points

I ran the model using the r and influence options in order to check for both outliers and influential points. Based on the Cook’s D metric being greater than 1, there were no outliers or influential points.

# Final Model and Predictions

The final model I used to calculate two predictions was:

Sqrt(popularity) = -0.64 +6.23budget +0.53original\_lang+0.17Production\_Countries +0.0023runtime + 0.24vote\_average + 0.00041vote\_count +0.31Collection -0.165 numProd2 + 0.19numProd4 + 0.20numProd5 – 0.13numGenre2 – 0.12numGenre3 – 1.07numGenre5 – 0.18numGenre6 – 1.02numGenre9 – 0.46numGenre10 – 0.37numGenre12 – 0.25numGenre13 – 0.22numGenre14 -0.33numGenre18 + 0.49numYear1 + 0.51numYear2 + 0.66numYear3

Where numProd2 = 1 > New Line Cinema

numProd4 = 1 > Warner Bros

numProd5 = 1 > MGM

numGenre2 = 1 > Animation

numGenre3 = 1 > Comedy

numGenre5 = 1 > Documentary

numGenre6 = 1 > Drama

numGenre9 = 1 > Foreign

numGenre10 = 1 > History

numGenre12 = 1 > Music

numGenre13 = 1 > Other

numGenre14 = 1 > Romance

numYear1 = 1 > Year is between 1951-2000

numYear2 = 1 > Year is between 2001-2010

numYear3 =1 > Year is between 2011-2017

Prediction 1:

First, I predicted the popularity of a movie with a budget of $500,000, language is English, Production Country is not in the USA, runtime of 120 mintutes, vote\_count of 100, vote\_average of 7.5, the movie is part of a collection, the production company is Walt Disney, the genre is Adventure, and it was released in 2000.

The prediction resulted in a value of 2.84 with Confidence Interval (2.769, 2.911), but since this was the sqrt(popularity) I needed to transform the value. The resulting value for the predicted popularity was 8.066 with CI (7.668, 8.475). The results are shown in Figure E.13.

Prediction 2:

The second movie has a budget of $800,000, language is English, Production Country is in the USA, runtime of 75 mintutes, vote\_count of 200, vote\_average of 8.5, the movie is part of a collection, the production company is New Line Cinema, the genre is Comedy, and it was released in 1995.

The prediction resulted in a value of 2.91 with Confidence Interval (2.7522, 3.0584) but since this was the sqrt(popularity) I needed to transform the value. The resulting value for the predicted popularity was 8.47 with CI (7.57, 9.35). The results are shown in Figure E.14

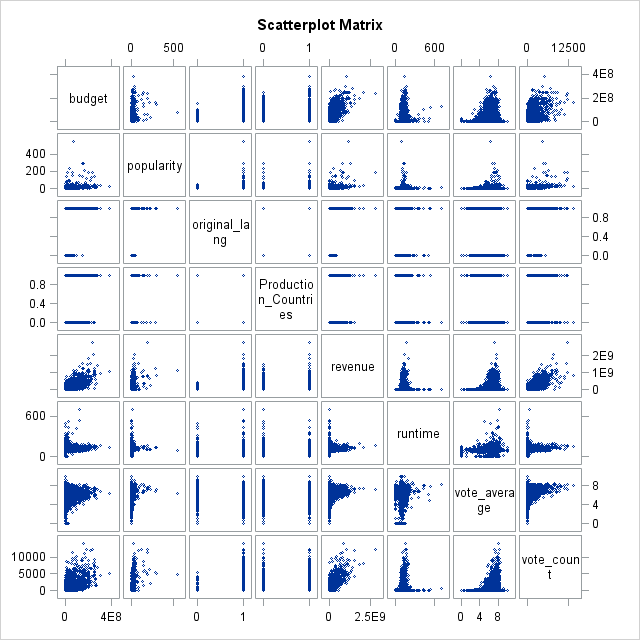
# Test and Train Model Test

I split the data into a test and train set in order to test the performance of the model. First, I used the train set on two different models, using a seed value of 56789 and samprate of 0.7. The first model, M1, resulted in a R-square value of 0.4858 and Adj-R2 of 0.4839. The second model, M2, resulted in a R-square value of 0.4774 and Adj-R2 of 0.4757. Based on the R-square and Adj-R2 values, M1 was a better model because it had higher R-square and Adj-R2 values. However, M2 had a higher F-value making it a better fit than M1. The results for these models can be seen in Figure E.15 and E.16.

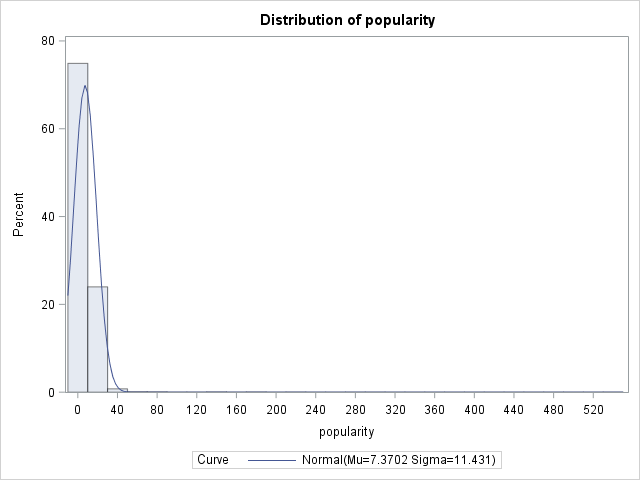
Next I ran the test set validation stats for both M1 and M2. The RMSE and MAE values of M1 were 0.8875 and 0.6643 respectively while the RMSE and MAE values for M2 were 0.8912 and 0.6486. M1 is a better model based on these two values since the goal is to reduce both the RMSE and MAE values. Usually we want to lean towards the model with better test performance, which in this case is M1.

# Appendix:

### Figure E.1: Scatterplot Matrix



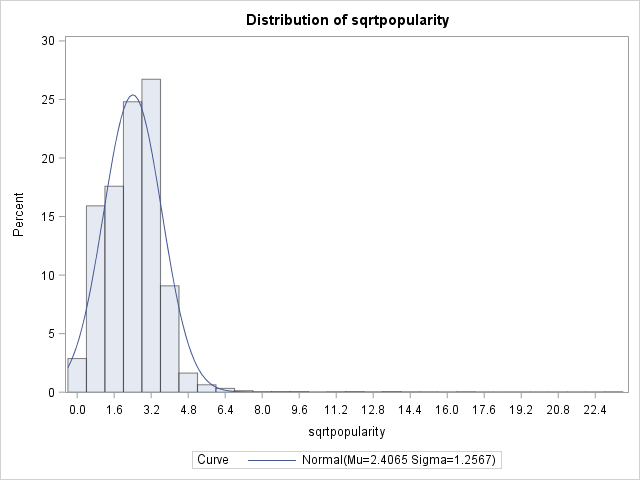
### Figure E.2 Popularity Histogram



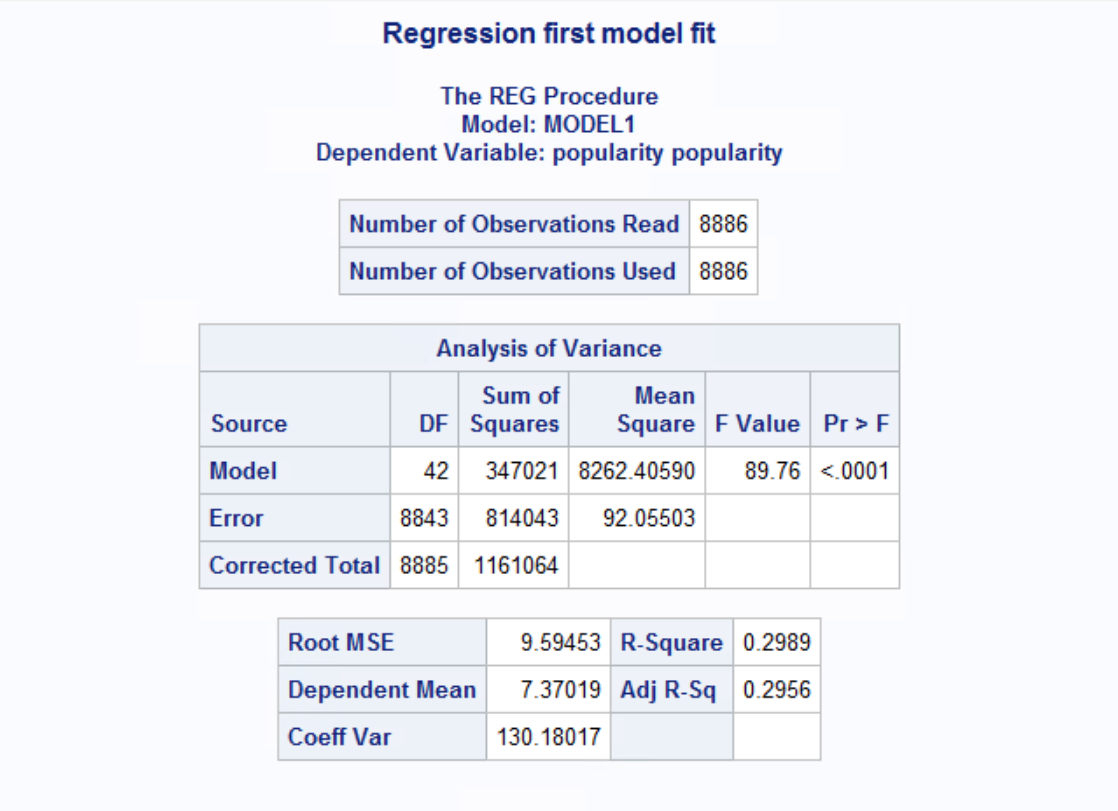
### Figure E.3 Ln(Popularity) Histogram



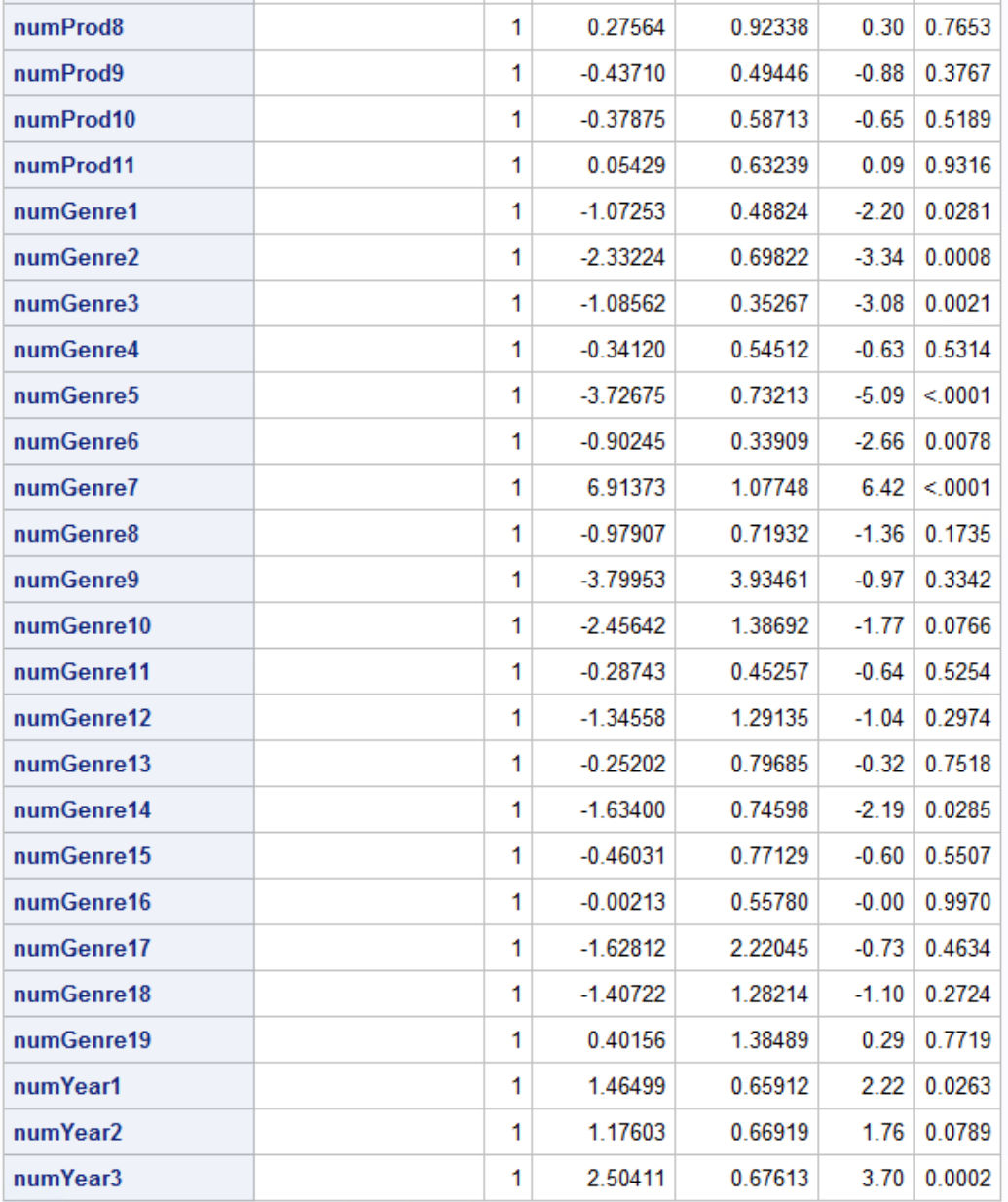
### Figure E.4 Sqrt(popularity) histogram

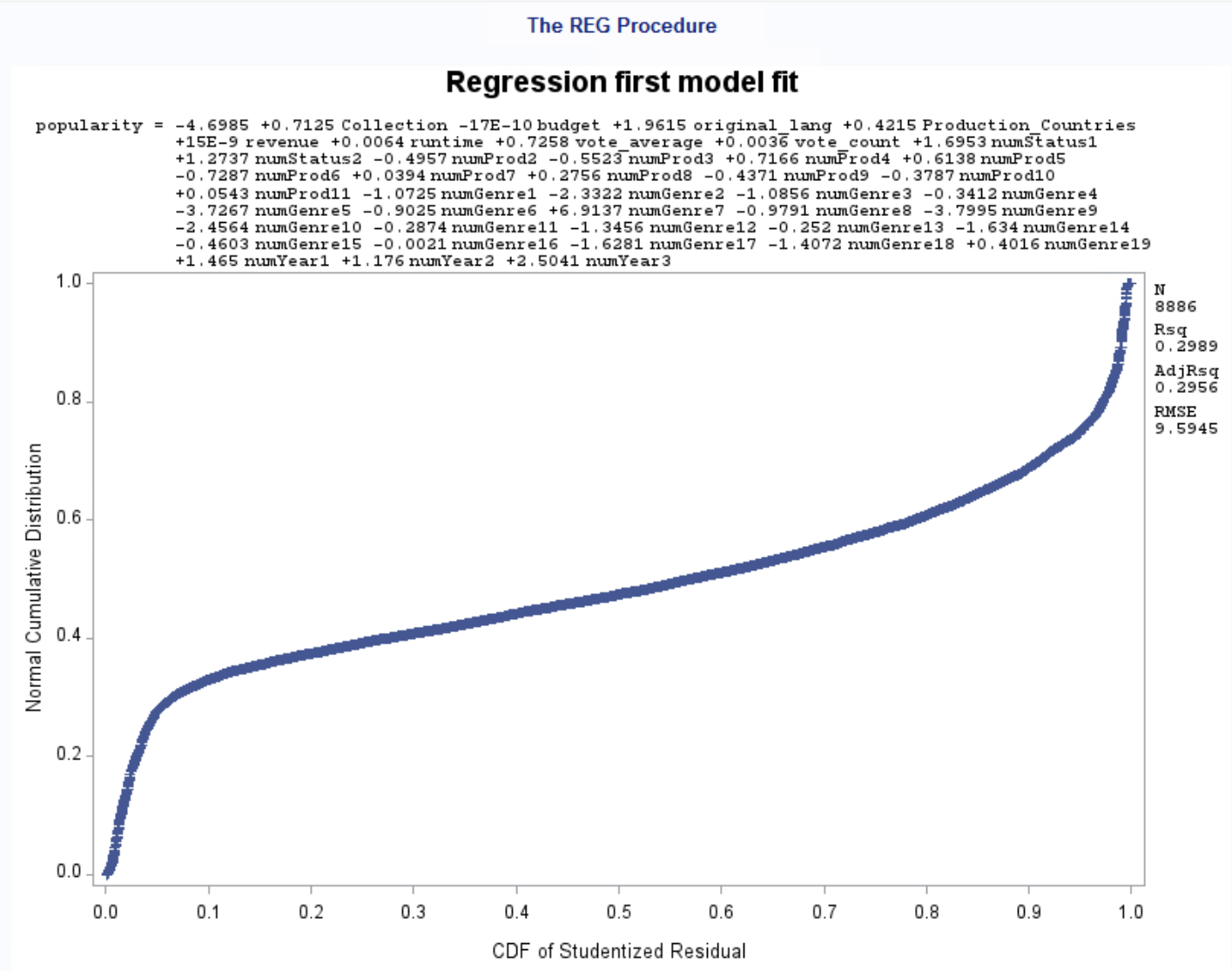


### Figure E.5 Popularity Regression Model

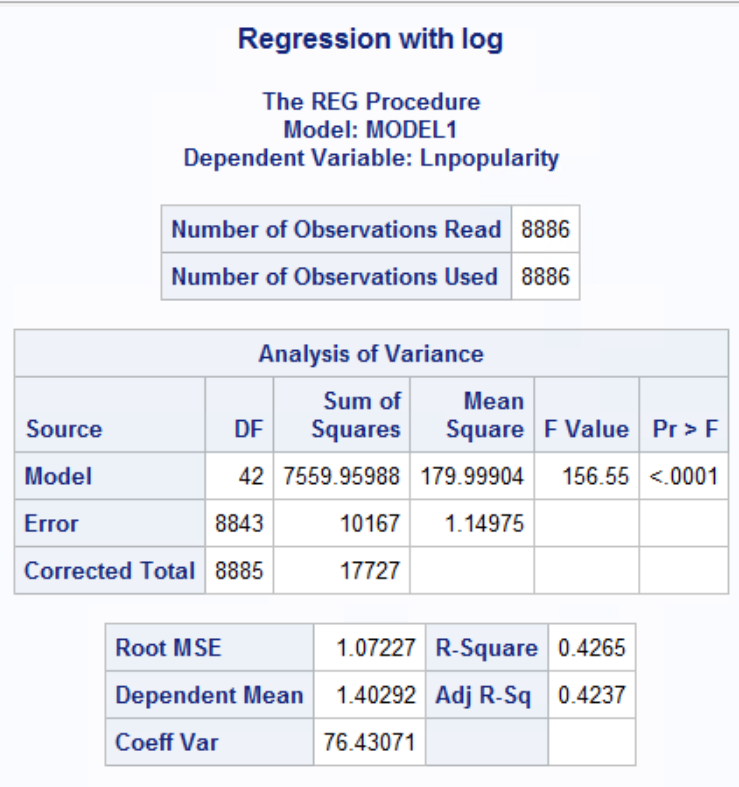


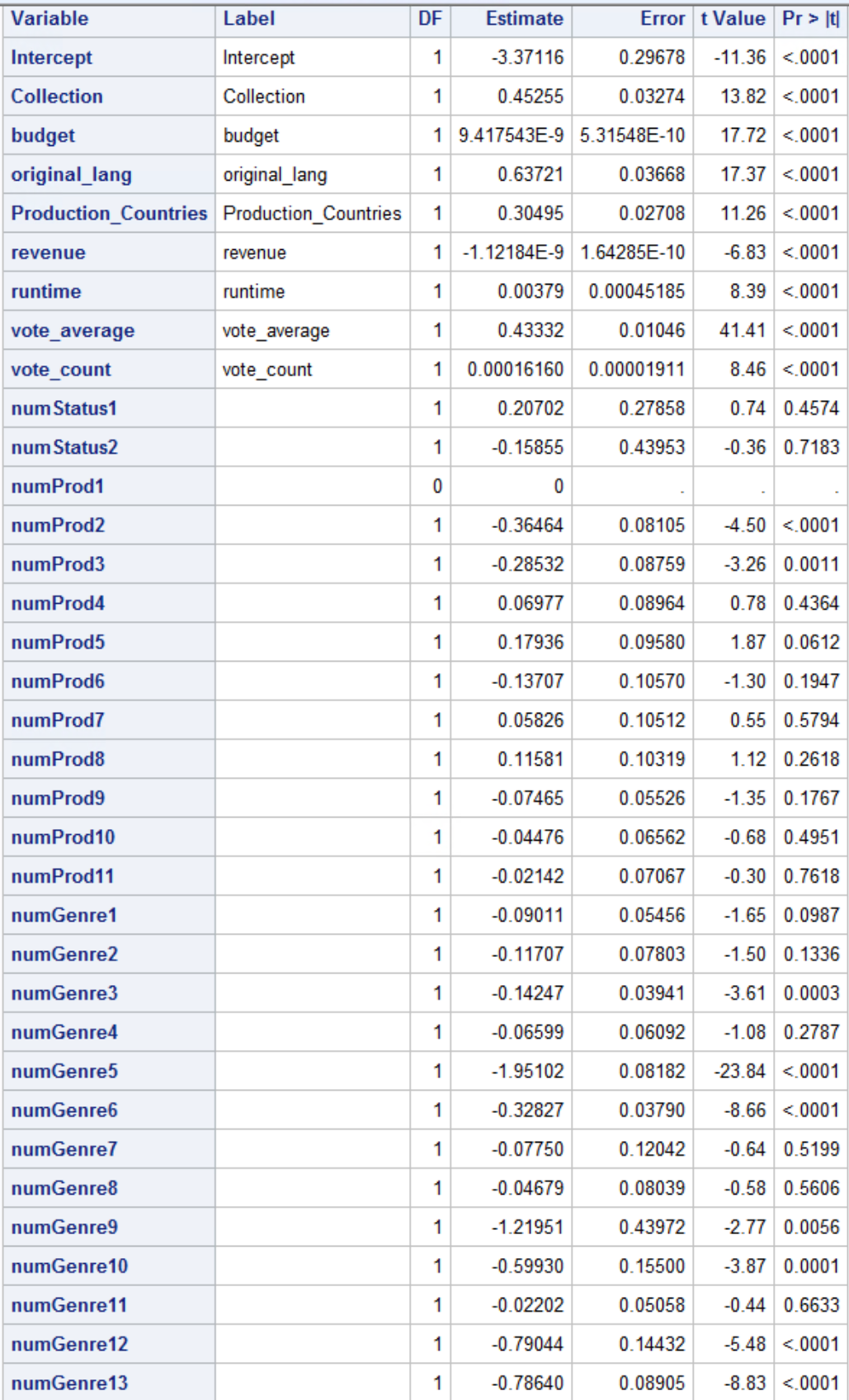






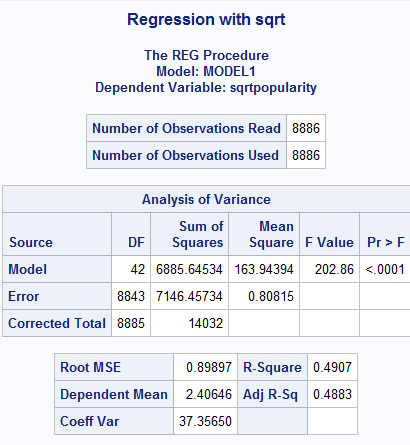
### Figure E.6 lnpopularity Regression Model

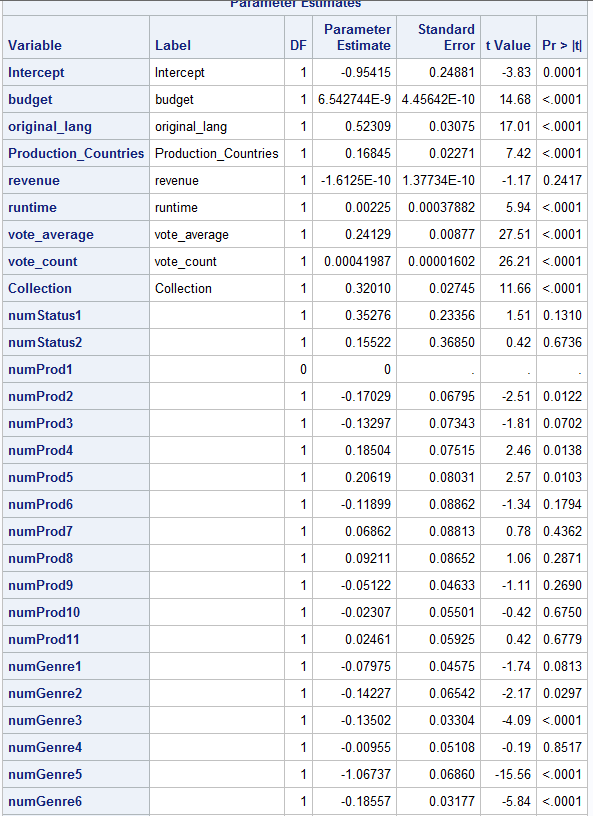


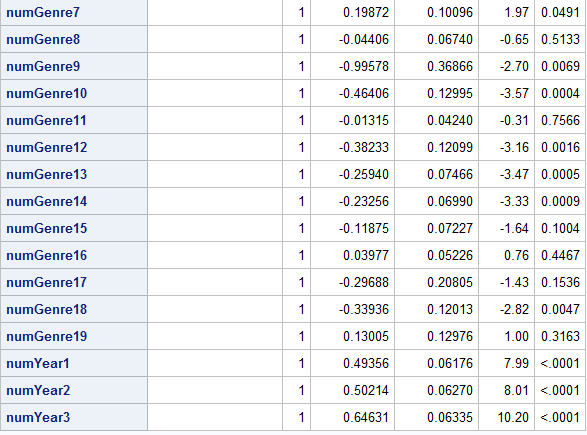




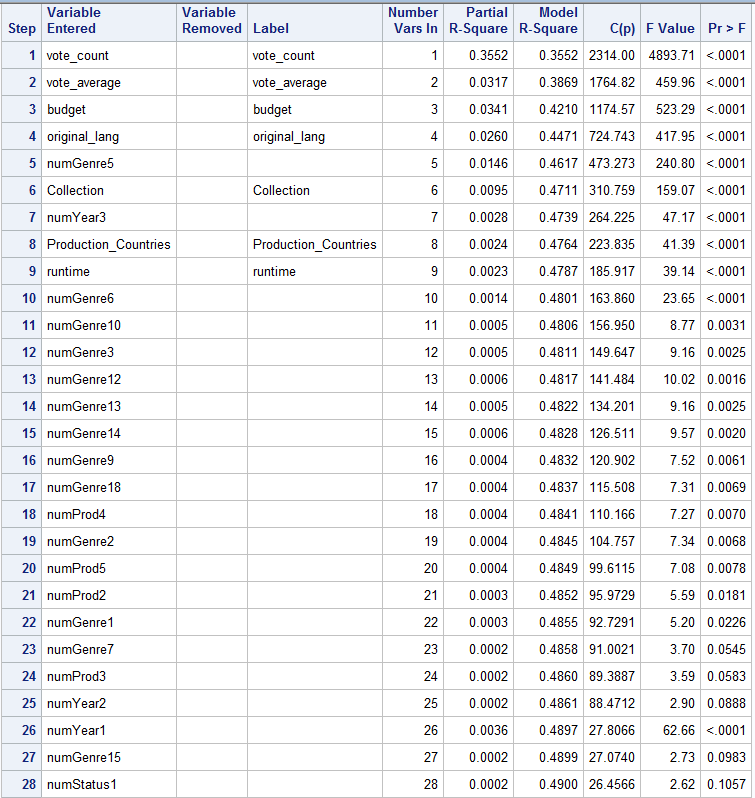
### Figure E.7 Sqrt(Popularity) Model



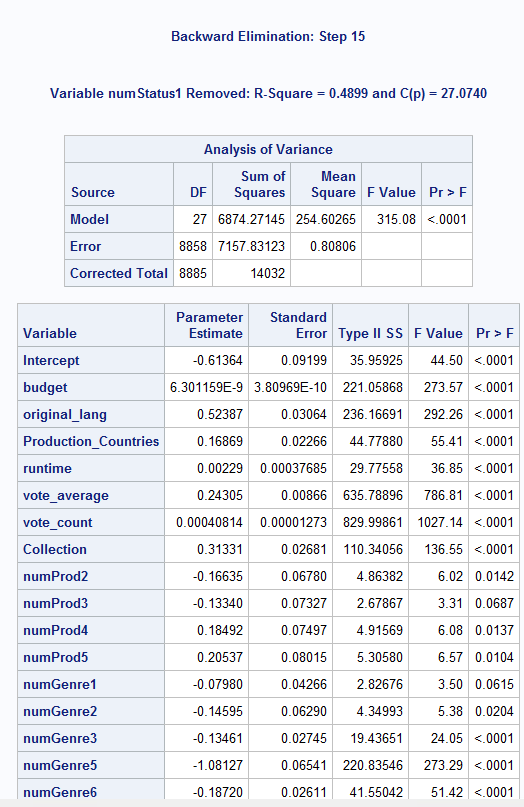


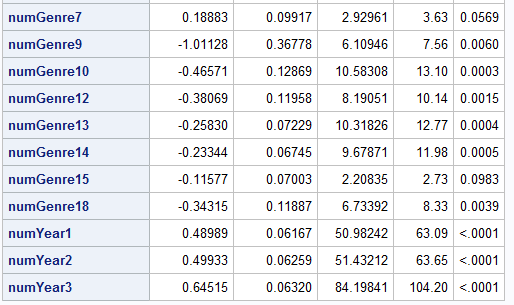


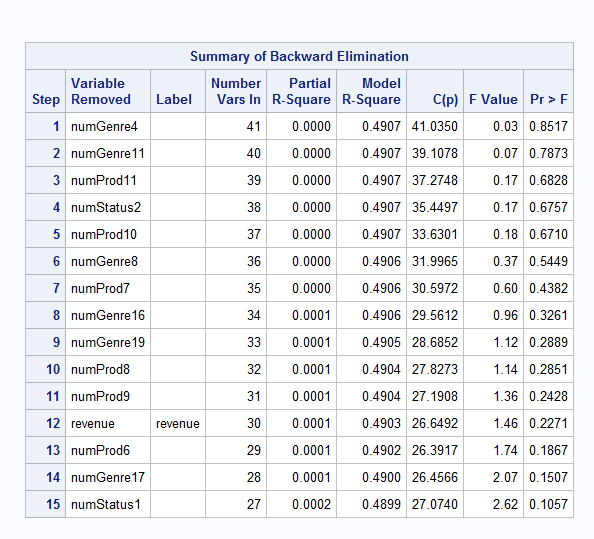
### Figure E.8 Stepwise model Selection with y=sqrt(popularity)



### Figure E.9 Backward model Selection with y=sqrt(popularity)



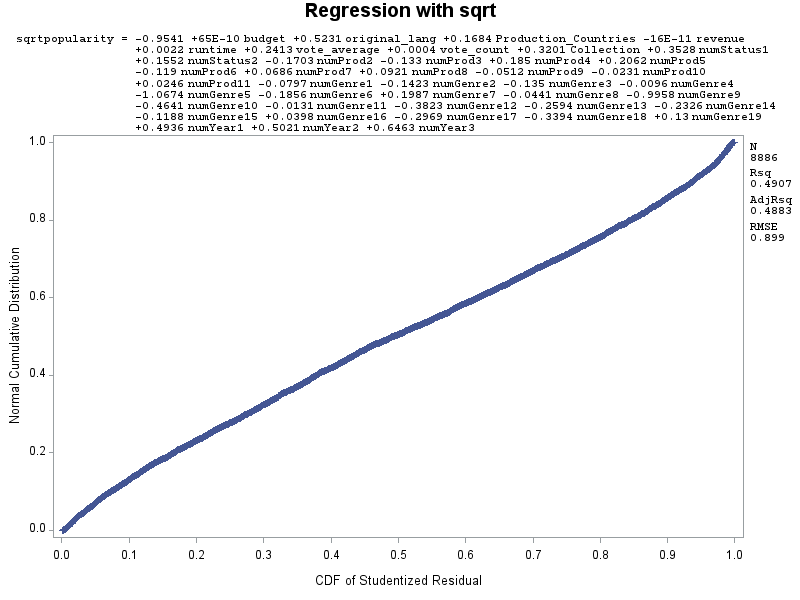


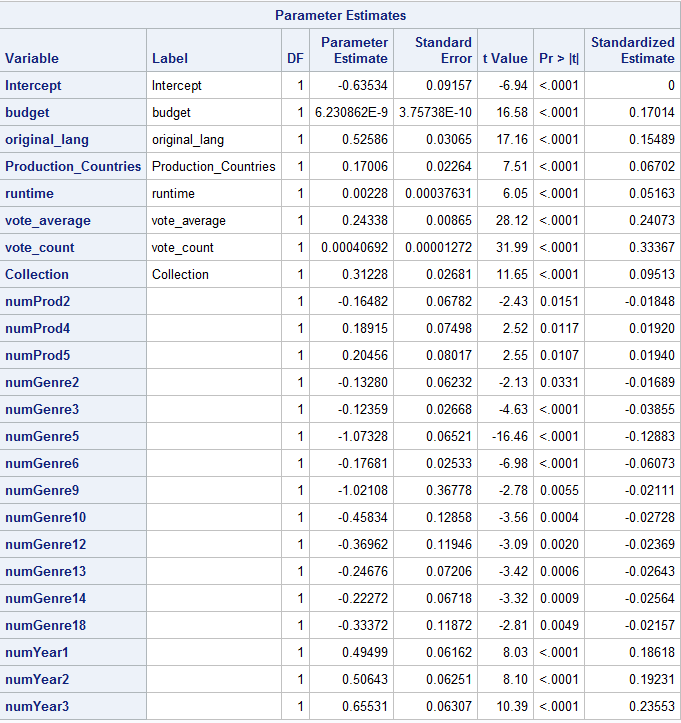


### Figure E.10 Sqrt(popularity) model with insignificant variables removed

### Figure E.11 Residual Plot (Student \* predicted)C:\Users\unkle_erica\AppData\Local\Microsoft\Windows\INetCache\Content.Word\reg1.png

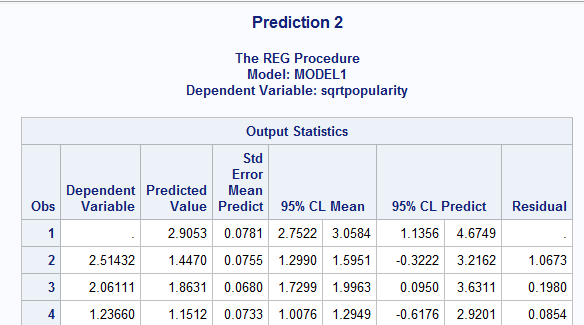
### Figure E.12 Residual Plot (Npp\*predicted)





### Figure E.13 Prediction 1

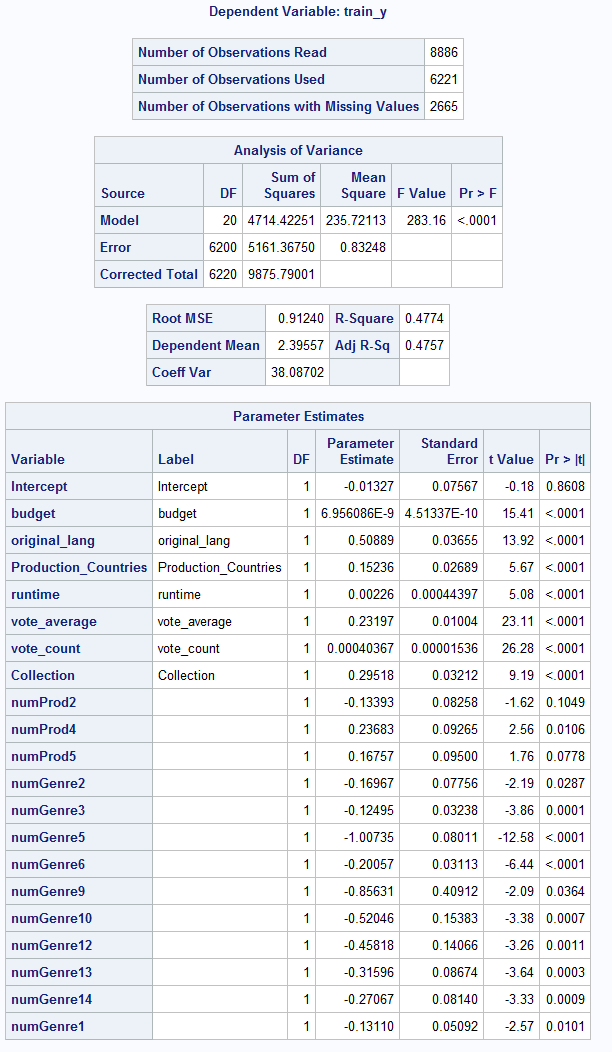
### Figure E.14 Prediction 2



### Figure E.15 Train set Model 1



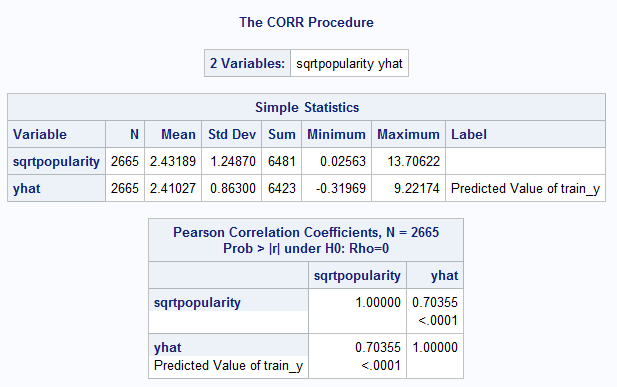
### Figure E.16 Train set Model 2



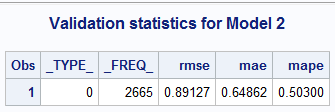
### Figure E.17 Validation stats for Model M1



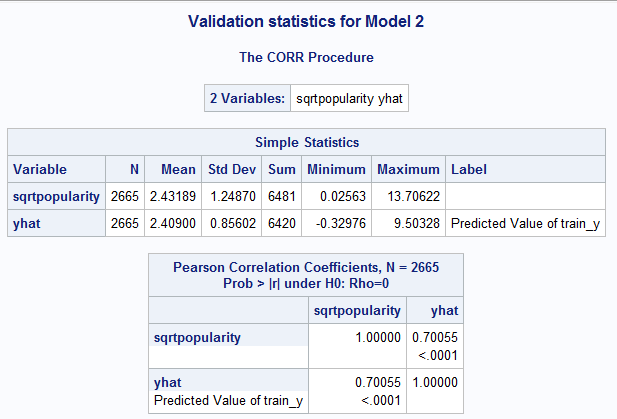
### Figure E.18 Correlation of observed and predicted values in test set for model M1



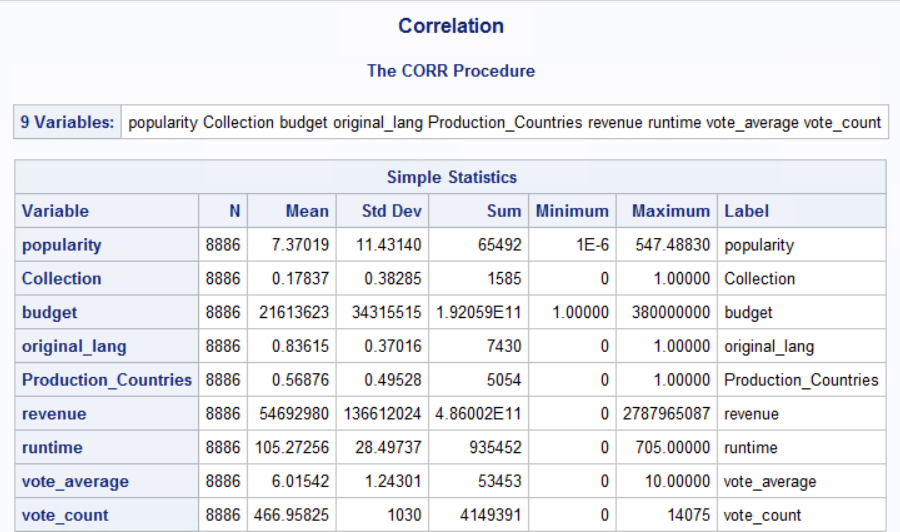
### Figure E.19 Validation stats for Model M2

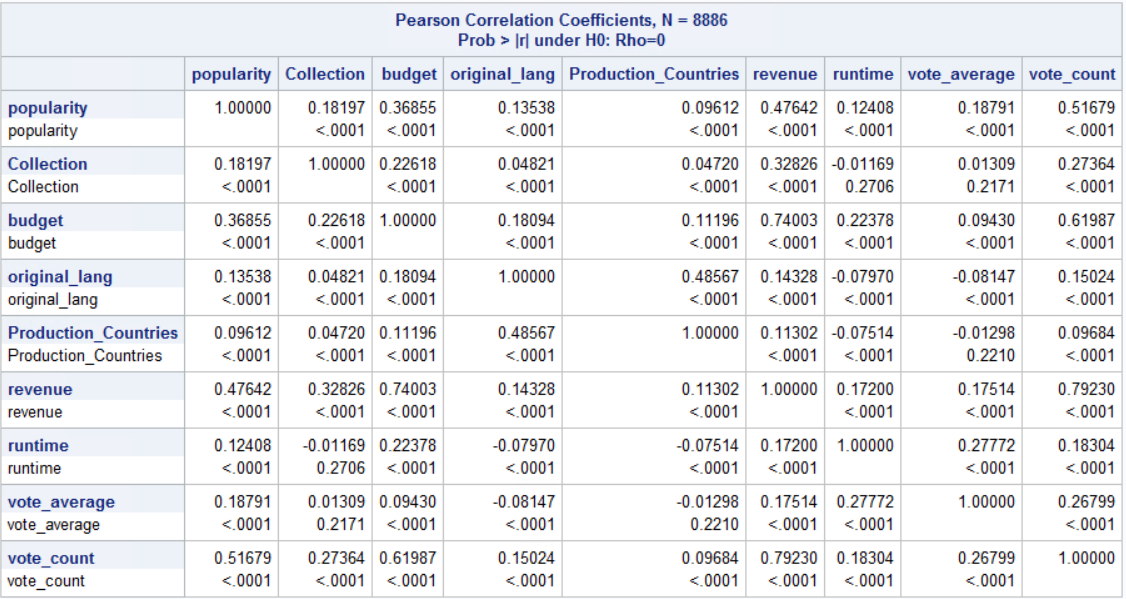


### Figure E.20 Correlation of observed and predicted values in test set for model M2

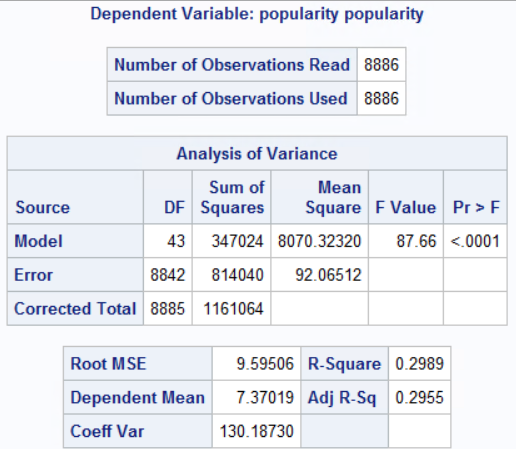


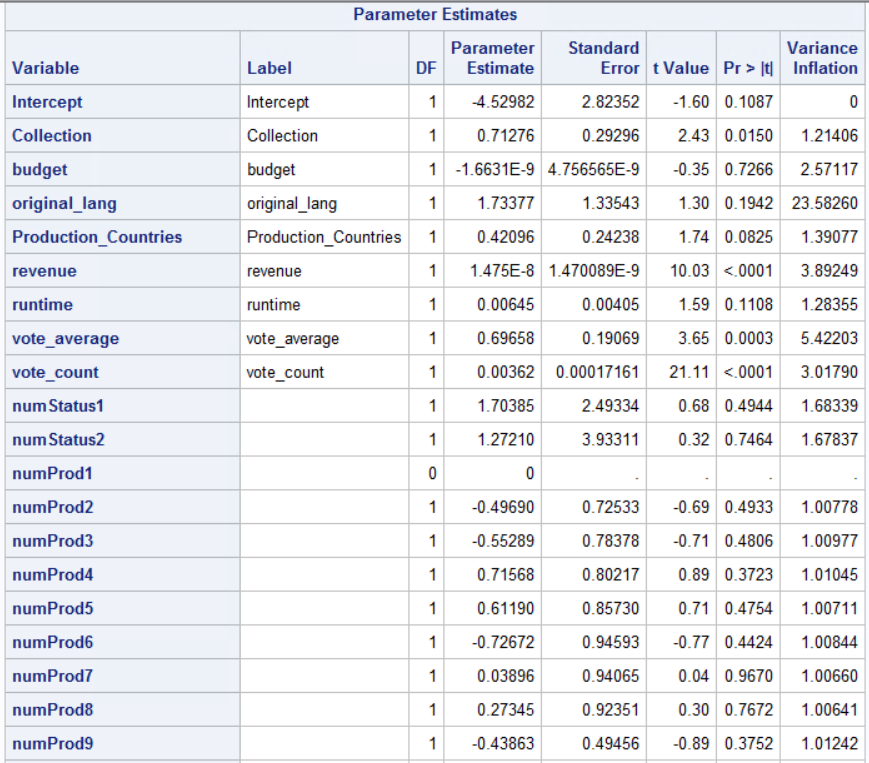
### Figure E.21 Pearson Correlation Coefficients



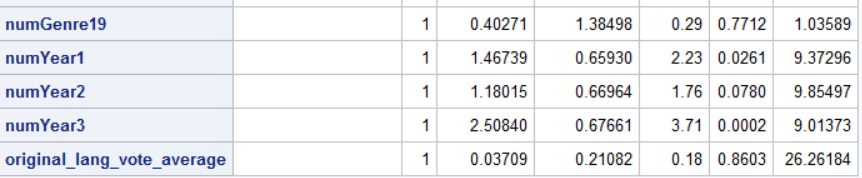


### Figure E.22 Reg Analysis with Interaction Variable









# References:

1. Spoelstra, Jeff. 2016. “And, the Oscar goes to…”: Modeling and predicting movie popularity. <https://rpubs.com/jeffspoelstra/movie-pop>. Accessed March 12, 2018.

# CODE:

title 'Import movie data';

**proc** **import** datafile='S:\Homeworks\movies\_project\_data.xls' out = movies dbms = xls replace;

getnames = yes;

**run**;

**proc** **print**;

title 'Add dummy variables';

**data** movies;

set movies;

numStatus1 = (Status = 'Released');

numStatus2 = (Status = 'Rumored');

\*Production company other is the base ;

numProd1 = (Production\_Companies = 'Universal Studios');

numProd2 = (Production\_Companies = 'New Line Cinema');

numProd3 = (Production\_Companies = **'Walt Disney Picture**s');

numProd4 = (Production\_Companies = 'Warner Bros.');

numProd5 = (Production\_Companies = 'Metro-Goldwyn-Mayer (MGM)');

numProd6 = (Production\_Companies = 'United Artists');

numProd7 = (Production\_Companies = 'Columbia Pictures Corporation');

numProd8 = (Production\_Companies = 'Miramax Films');

numProd9 = (Production\_Companies = 'Paramount Pictures');

numProd10 = (Production\_Companies = 'Twentieth Century Fox Film Corporation');

numProd11 = (Production\_Companies = 'Columbia Pictures');

\*Genre Action is the base genre;

numGenre1 = (Genre = 'Adventure');

numGenre2 = (Genre = 'Animation');

numGenre3 = (Genre = 'Comedy');

numGenre4 = (Genre = 'Crime');

numGenre5 = (Genre = 'Documentary');

numGenre6 = (Genre = 'Drama');

numGenre7 = (Genre = 'Family');

numGenre8 = (Genre = 'Fantasy');

numGenre9 = (Genre = 'Foreign');

numGenre10 = (Genre = 'History');

numGenre11 = (Genre = 'Horror');

numGenre12 = (Genre = 'Music');

numGenre13 = (Genre = 'Other');

numGenre14 = (Genre = 'Romance');

numGenre15 = (Genre = 'Science Fiction');

numGenre16 = (Genre = 'Thriller');

numGenre17 = (Genre = 'TV Movie');

numGenre18 = (Genre = 'War');

numGenre19 = (Genre = 'Western');

\*Base year is 1900-1950;

numYear1 = (Release\_Year = '1951-2000');

numYear2 = (Release\_Year = '2001-2010');

numYear3 = (Release\_Year = '2011-2018');

original\_lang\_vote\_average = original\_lang\*vote\_Average;

**proc** **print**;

**run**;

title "Scatterplot Matrix";

**proc** **sgscatter** data=movies;

matrix budget popularity original\_lang production\_Countries revenue runtime vote\_Average vote\_count;

**run**;

title "Histogram for (popularity)";

**proc** **univariate**;

var popularity;

histogram/normal;

**run**;

title "Regression first model fit";

**proc** **reg** data=movies;

model popularity = collection budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count numStatus1 numStatus2 numProd1 numProd2 numProd3 numProd4 numProd5 numProd6 numProd7 numProd8 numProd9 numProd10 numProd11 numGenre1 numGenre2 numGenre3 numGenre4 numGenre5 numGenre6 numGenre7 numGenre8 numGenre9 numGenre10 numGenre11 numGenre12 numGenre13 numGenre14 numGenre15 numGenre16 numGenre17 numGenre18 numGenre19 numYear1 numYear2 numYear3;

plot student.\*predicted.;

plot npp.\*student.;

plot student.\*(popularity collection budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count);

**run**;

title "Regression model with interaction variable”;

**proc** **reg** data=movies;

model popularity = collection budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count numStatus1 numStatus2 numProd1 numProd2 numProd3 numProd4 numProd5 numProd6 numProd7 numProd8 numProd9 numProd10 numProd11 numGenre1 numGenre2 numGenre3 numGenre4 numGenre5 numGenre6 numGenre7 numGenre8 numGenre9 numGenre10 numGenre11 numGenre12 numGenre13 numGenre14 numGenre15 numGenre16 numGenre17 numGenre18 numGenre19 numYear1 numYear2 numYear3 budget\_vote\_average;

plot student.\*predicted.;

plot npp.\*student.;

**run**;

title 'Correlation';

**proc** **corr** data = movies;

var popularity collection budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count;

**run**;

title 'Log transformation';

**data** movies;

set movies;

Lnpopularity = log(popularity);

**proc** **print**;

title "Histogram for ln(popularity)";

**proc** **univariate**;

var popularity;

histogram/normal;

**run**;

title 'Sqrt Transformation';

**data** movies;

set movies;

sqrtpopularity = sqrt(popularity);

**proc** **print**;

title "Histogram for sqrt(popularity)";

**proc** **univariate**;

var sqrtpopularity;

histogram/normal;

**run**;

title 'Regression with log';

**proc** **reg** data = movies;

model lnpopularity= collection budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count numStatus1 numStatus2 numProd1 numProd2 numProd3 numProd4 numProd5 numProd6 numProd7 numProd8 numProd9 numProd10 numProd11 numGenre1 numGenre2 numGenre3 numGenre4 numGenre5 numGenre6 numGenre7 numGenre8 numGenre9 numGenre10 numGenre11 numGenre12 numGenre13 numGenre14 numGenre15 numGenre16 numGenre17 numGenre18 numGenre19 numYear1 numYear2 numYear3;

\*plot student.\*predicted.;

\*plot npp.\*student.;

**run**;

title 'Regression with sqrt';

**proc** **reg** data = movies;

model sqrtpopularity= budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count Collection numStatus1 numStatus2 numProd1 numProd2 numProd3 numProd4 numProd5 numProd6 numProd7 numProd8 numProd9 numProd10 numProd11 numGenre1 numGenre2 numGenre3 numGenre4 numGenre5 numGenre6 numGenre7 numGenre8 numGenre9 numGenre10 numGenre11 numGenre12 numGenre13 numGenre14 numGenre15 numGenre16 numGenre17 numGenre18 numGenre19 numYear1 numYear2 numYear3;

plot student.\*predicted.;

plot npp.\*student.;

plot student.\*(popularity collection budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count);

**run**;

title 'Regression2 w/ sqrt using stepwise model selection and check for collinearity';

**proc** **reg** data = movies;

model sqrtpopularity= budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count Collection numStatus1 numStatus2 numProd1 numProd2 numProd3 numProd4 numProd5 numProd6 numProd7 numProd8 numProd9 numProd10 numProd11 numGenre1 numGenre2 numGenre3 numGenre4 numGenre5 numGenre6 numGenre7 numGenre8 numGenre9 numGenre10 numGenre11 numGenre12 numGenre13 numGenre14 numGenre15 numGenre16 numGenre17 numGenre18 numGenre19 numYear1 numYear2 numYear3/selection = stepwise;

**run**;

title 'Regression3 w/ sqrt using backward model selection';

**proc** **reg** data = movies;

model sqrtpopularity= budget original\_lang production\_Countries revenue runtime vote\_Average vote\_count Collection numStatus1 numStatus2 numProd1 numProd2 numProd3 numProd4 numProd5 numProd6 numProd7 numProd8 numProd9 numProd10 numProd11 numGenre1 numGenre2 numGenre3 numGenre4 numGenre5 numGenre6 numGenre7 numGenre8 numGenre9 numGenre10 numGenre11 numGenre12 numGenre13 numGenre14 numGenre15 numGenre16 numGenre17 numGenre18 numGenre19 numYear1 numYear2 numYear3/selection = backward;

**run**;

title 'Regression4 w/ sqrt using backward model variables ';

**proc** **reg** data = movies;

model sqrtpopularity= budget original\_lang production\_Countries runtime vote\_Average vote\_count Collection numProd2 numProd3 numProd4 numProd5 numGenre1 numGenre2 numGenre3 numGenre5 numGenre6 numGenre7 numGenre9 numGenre10 numGenre12 numGenre13 numGenre14 numGenre15 numGenre18 numYear1 numYear2 numYear3/vif;

**run**;

title 'Regression5 w/ sqrt removing insignificant variables ';

**proc** **reg** data = movies;

model sqrtpopularity= budget original\_lang production\_Countries runtime vote\_Average vote\_count Collection numProd2 numProd4 numProd5 numGenre2 numGenre3 numGenre5 numGenre6 numGenre9 numGenre10 numGenre12 numGenre13 numGenre14 numGenre18 numYear1 numYear2 numYear3/r influence stb;

**run**;

title 'Compute prediction for movie 1';

**data** new1;

input budget original\_lang production\_Countries runtime vote\_count vote\_average collection numProd3 numYear1 numGenre1 revenue numStatus1 numStatus2 numProd1 numProd2 numProd4 numProd5 numProd6 numProd7 numProd8 numProd9 numProd10 numProd11 numGenre2 numGenre3 numGenre4 numGenre5 numGenre6 numGenre7 numGenre8 numGenre9 numGenre10 numGenre11 numGenre12 numGenre13 numGenre14 numGenre15 numGenre16 numGenre17 numGenre18 numGenre19 numYear2 numYear3;

;

datalines;

500000 1 0 120 100 7.5 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

;

**proc** **print**;

**run**;

**data** pred;

set new1 movies;

**run**;

**proc** **print**;

**run**;

\*\*use cli for prediction intervals, and clm for intervals for averages CL mean is the CI;

title 'Prediction 1';

**proc** **reg** data=pred;

model sqrtpopularity= budget original\_lang production\_Countries runtime vote\_Average vote\_count Collection numProd2 numProd4 numProd5 numGenre2 numGenre3 numGenre5 numGenre6 numGenre9 numGenre10 numGenre12 numGenre13 numGenre14 numGenre18 numYear1 numYear2 numYear3/p CLI CLM;

**run**;

\*prediction 2;

title 'Compute prediction for movie 1';

**data** new1;

input budget original\_lang production\_Countries runtime vote\_count vote\_average collection numProd2 numYear1 numGenre3 revenue numStatus1 numStatus2 numProd1 numProd3 numProd4 numProd5 numProd6 numProd7 numProd8 numProd9 numProd10 numProd11 numGenre2 numGenre1 numGenre4 numGenre5 numGenre6 numGenre7 numGenre8 numGenre9 numGenre10 numGenre11 numGenre12 numGenre13 numGenre14 numGenre15 numGenre16 numGenre17 numGenre18 numGenre19 numYear2 numYear3;

;

datalines;

800000 1 1 75 200 8.5 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

;

**proc** **print**;

**run**;

**data** pred;

set new1 movies;

**run**;

\*\*use cli for prediction intervals, and clm for intervals for averages CL mean is the CI;

title 'Prediction 2';

**proc** **reg** data=pred;

model sqrtpopularity= budget original\_lang production\_Countries runtime vote\_Average vote\_count Collection numProd2 numProd4 numProd5 numGenre2 numGenre3 numGenre5 numGenre6 numGenre9 numGenre10 numGenre12 numGenre13 numGenre14 numGenre18 numYear1 numYear2 numYear3/p CLI CLM;

**run**;

\*Generate the test samples: training set used to fit the model;

**proc** **surveyselect** data=movies out=train\_all seed=**56789** samprate=**0.70** outall;

**run**;

\*proc print data=train\_all;

\*run;

\*create new variable train\_y = sqrtpopularity for training set, and = NA for testing set;

**data** train\_all;

set train\_all;

if selected then train\_y=sqrtpopularity;

**run**;

**proc** **print** data=train\_all;

**run**;

/\* Fit models on training data\*/

\* MODEL 1;

title 'Model 1';

**Proc** **reg** data=train\_all;

model train\_y= budget original\_lang production\_Countries runtime vote\_Average vote\_count Collection numProd2 numProd4 numProd5 numGenre2 numGenre3 numGenre5 numGenre6 numGenre9 numGenre10 numGenre12 numGenre13 numGenre14 numGenre18 numYear1 numYear2 numYear3;

output out=outm1(where=(train\_y=**.**)) p=yhat;

**run**;

\* MODEL 2 – excludes year variables;

title 'Model 2';

**Proc** **reg** data=train\_all;

model train\_y= budget original\_lang production\_Countries runtime vote\_Average vote\_count Collection numProd2 numProd4 numProd5 numGenre2 numGenre3 numGenre5 numGenre6 numGenre9 numGenre10 numGenre12 numGenre13 numGenre14 numGenre1;

output out=outm2(where=(train\_y=**.**)) p=yhat;

**run**;

\* Analysis of predictions on testing set for model M1;

title 'Analysis for M1';

**data** outm1\_test;

set outm1;

d= sqrtpopularity -yhat; \*d is the difference between observed and predicted values in training set;

absd=abs(d);

pe=abs(d/ sqrtpopularity);

**run**;

/\* Computes predictive statistics: root mean square error (rmse),

mean absolute error (mae) and mean absolute percentage error (MAPE) for model M1\*/

**proc** **summary** data=outm1\_test;

var d absd;

output out=outm1\_stats std(d)=rmse mean(absd)=mae mean(pe)=mape;

**run**;

title 'Validation statistics for Model 1';

**proc** **print** data=outm1\_stats;

**run**;

\*computes correlation of observed and predicted values in test set for model M1;

**proc** **corr** data=outm1;

var sqrtpopularity yhat;

**run**;

/\* Analysis of predictions on testing set for model M2\*/

**data** outm2\_test;

set outm2;

d= sqrtpopularity -yhat;

absd=abs(d);

pe=abs(d/ sqrtpopularity);

**run**;

/\* computes predictive statistics: root mean square error (rmse)

and mean absolute error (mae) and saves the output in new outm2\_stats dataset\*/

**proc** **summary** data=outm2\_test;

var d absd;

output out=outm2\_stats std(d)=rmse mean(absd)=mae mean(pe)=mape;

**run**;

title 'Validation statistics for Model 2';

**proc** **print** data=outm2\_stats;

**run**;

\*computes correlation of observed and predicted values in test set;

**proc** **corr** data=outm2;

var sqrtpopularity yhat;

**run**;