**Predicting Gross Revenues of Movies**

By,

Theron Spencer || Spencet5@wwu.edu

MATH342

Dr. Ramadha Piyadi Gamage

6/5/2018

Introduction

The film industry is massive and accrues billions of dollars each year. With an enormous amount of money at stake, it’s in the best interest of anyone in the industry to know what variables lead to financial success. Obtained from Kaggle Inc, is a data set consisting of ten variables pertaining to a sample of five hundred movies, including gross revenue, budget, the movie’s IMDb rating, total IMDb ratings count, movie length (minutes), MPAA rating (G, PG, PG-13, R), cast popularity and main actor’s popularity. (<https://www.kaggle.com/carolzhangdc/analyze-imdb-score-with-data-mining-algorithms/data>) In an attempt to predict a movie’s financial success (gross revenue), a multiple linear regression model will be fitted using the other nine variables. With high budgets come technological excellence and famous actors, factors that are expected to contribute to a movie’s gross revenue. Successful films are predicted to have high IMDb ratings and rating counts. Larger rating counts may imply a larger amount of people have seen it which in turn means more ticket sales. Cast popularity and the main actor’s popularity, measured by the amount of Facebook likes they have, may also contribute to a movie’s financial success. After all, many people are known to see movies on the criteria of cast alone. Longer movies provide a longer duration of entertainment, a possible contributor to gross. Movies of different MPAA ratings pander to different audiences. For example, G rated movies attract large audiences of all ages compared to R rated movies that may gross less because they attract an older and smaller demographic.

Regression Model

In the end it was found that a movie’s gross revenue was explained best by the number of votes, its MPAA rating (all 4 levels), and budget. Before model selection was even attempted, a strong positive correlation of 0.92 was discovered between the cast popularity and the main actor’s popularity. Due to the main actor’s popularity being a direct contributor to the overall cast popularity, the main actor’s popularity variable was removed from the variable candidates in order to avoid multicollinearity. Using the stepwise regression procedure with an in and out alpha level of 0.05, it was found that a movie’s gross revenue is best explained by the number of votes, its MPAA rating and the budget. In identifying the best model using the best subset selection procedure according to Cp criterion, the best subset was determined to be gross being explained best by votes, budget, cast popularity, and its MPAA rating (all 4 levels). Cast popularity was removed from the final model for two reasons. Firstly, after transforming the response variable to achieve normality and remedy heteroscedasticity, cast popularity became insignificant and secondly, its contribution to gross was meager in comparison to the other variables. The final model was found to have an adjusted R-squared value of 0.549, which indicates that a movie’s gross isn’t explained very well by the final selected variables. Vote count, budget, and the MPAA PG-13 dummy variable were found to be significant at a 0.001 alpha level. The MPAA PG dummy variable was found to be significant at a 0.01 alpha level and the MPAA G dummy variable was found to be significant at a 0.05 alpha level.

Regression Output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Call:  lm(Formula = Gross1/3 ~ Votes + Budget + MPAAG + MPAAPG13 + MPAAPG | | | | |
|  | | | | |
| Residuals: | | | | |
| Min | 1Q | Median | 3Q | Max |
| -302.20 | -70.55 | -3.55 | 68.14 | 379.00 |
|  | | | | |
| Coefficients: | | | | |
|  | Estimate | Std. Error | t-value | Pr(>|t|) |
| (Intercept) | 1.981e+02 | 7.737e+00 | 25.608 | < 2e-16 \*\*\* |
| Votes | 3.661e-04 | 3.084e-05 | 11.870 | < 2e-16 \*\*\* |
| MPAAPG | 6.087e+01 | 1.469e+01 | 4.144 | 4.01e-05 \*\*\* |
| MPAAPG13 | 2.823e+01 | 1.091e+01 | 2.587 | 0.00996 \*\* |
| Budget | 1.572e-06 | 1.226e-07 | 12.819 | < 2e-16 \*\*\* |
| MPAAG | 7.121e+01 | 3.096e+01 | 2.300 | 0.02188 \* |
| Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1 | | | | |
| Residual standard error: 103.7 on 493 degrees of freedom  Multiple R-squared: 0.5532, Adjusted R-squared: 0.5487  F-statistic: 122.1 on 5 and 493 DF, p-value: < 2.2e-16 | | | | |

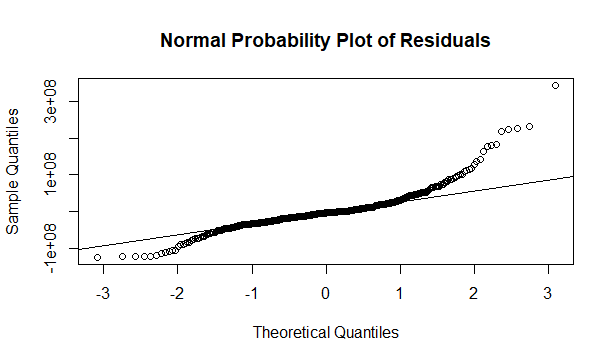
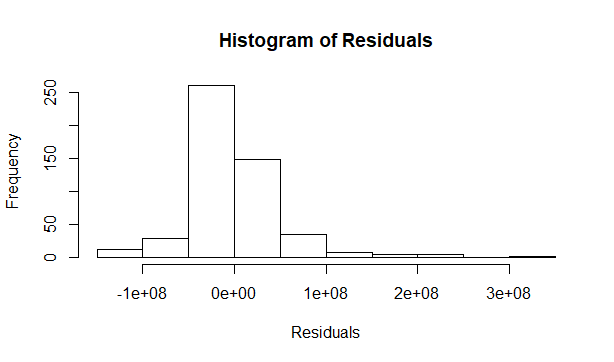
Analysis of Variance Table (Response: Gross1/3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Df | Sum Sq | Mean Sq | F- value | Pr(>F) |
| Votes | 1 | 3580383 | 3580383 | 333.1252 | < 2.2e-16 \*\*\* |
| MPAAPG | 1 | 570418 | 570418 | 53.0727 | 1.283e-12 \*\*\* |
| MPAAPG13 | 1 | 473364 | 473364 | 44.0427 | 8.485e-11 \*\*\* |
| Budget | 1 | 1880422 | 1880422 | 174.9578 | < 2.2e-16 \*\*\* |
| MPAAG | 1 | 56844 | 56844 | 5.2889 | 0.02188 \* |
| Residuals | 493 | 5298694 | 10748 |  |  |

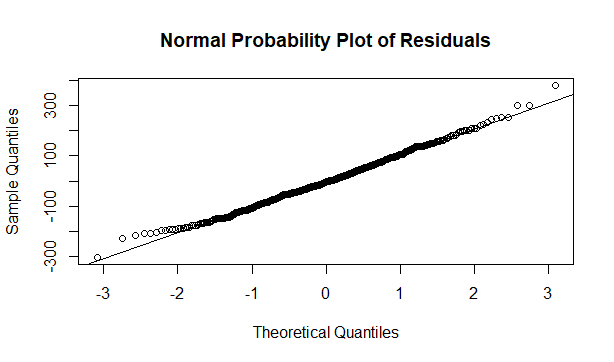
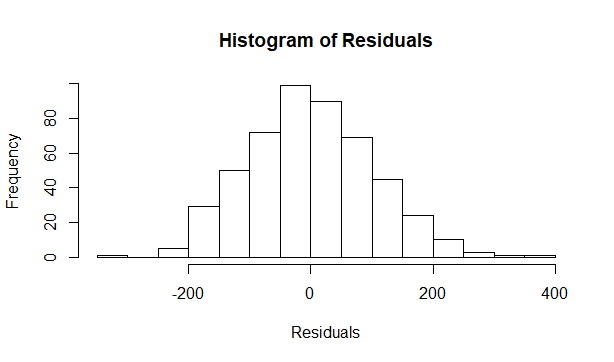
Regression Diagnostics

In order to achieve normality and remedy heteroscedasticity, a cubic root transformation on the dependent variable, gross, was necessary. Many combinations of transformations were tried on both the dependent and predictor variables but the cubic root of the response was most successful. The following diagrams show the histograms, normal probability plots, and studentized residuals plotted against fitted values before and after the transformation.

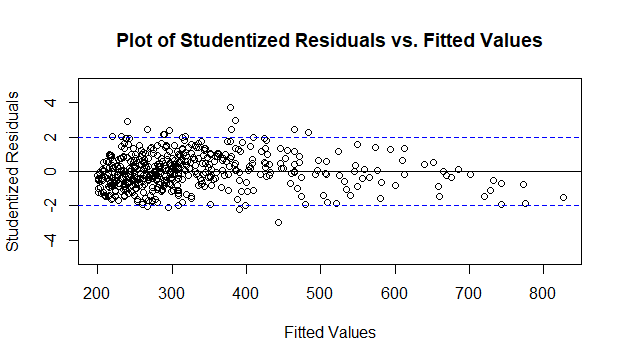
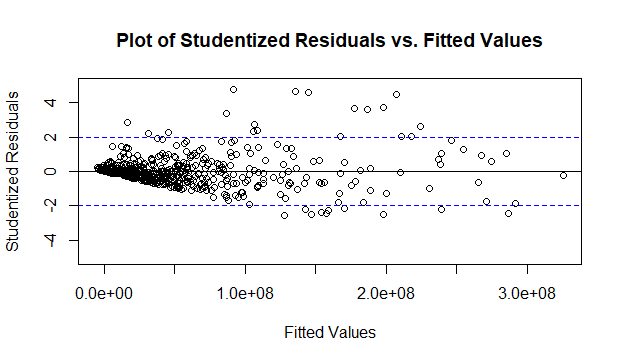
Before Transformation

After Transformation

Before Transformation After Transformation



Before, the Shapiro Wilks test yielded a p-value less than 2.2e-16 implying the residuals are not normally distributed. After the transformation, the Shapiro Wilks test yielded a p-value of 0.5744, in which case the null hypothesis stating the residuals are normally distributed didn’t get rejected. After the transformation was made, no distinct trends in the residual plots were identified. There were no high leverage observations and while there were 14 outliers, it was found that only one was influential enough to warrant action. The 454th data point had a cook’s distance of nearly 150, and when removed increased the adjusted r-squared of the model significantly from 0.397 to 0.549.

Using the Model

Now that the final model has been determined, an attempt at predicting the gross revenue of a particular movie will be made. The movie of choice will be “Black Panther,” a PG-13 rated film with 320,181 IMDb votes, and a budget of $200,000,000. Using the final model, the fitted value is determined to be 657.9237 with a 95% confidence interval of [622.063, 693.7843]. Note however that this is the cubic root of the gross and after cubing these quantities, a value of $284,791,218 is obtained with a 95% confidence interval of [$240,714,976, $333,943,814]. We can be 95% confident that the true gross of “Black Panther” is somewhere between $240,714,976 and $333,943,814. This however, is not the case. As has already been stated, the model only has an adjusted r-squared value of 0.549 so it’s unlikely the model will be sufficiently accurate. When comparing the predicted value to the actual value, we find “Black Panther” grossed $698,882,949, over double what the model predicts.

Conclusions

In order for the model assumptions to hold, a cubic root transformation on the response variable was made. However, this Transformation inevitably led to being unable to accurately interpret the parameter estimates. From the regression output, the only information that can be gleaned, just as hypothesized, is the fact that G rated movies gross the most, followed by PG rated movies, then PG-13 rated movies, with R rated movies being the baseline and grossing the least. This particular transformation makes it impossible to say by exactly how much. The impact of vote count and budget on gross is significant but again, it’s difficult to say to what extent. Surprisingly, cast popularity wasn’t influential enough to warrant its inclusion in the model. After inspection, it was found that the number of Facebook likes in the dataset failed to correspond with how many the cast and main actors actually had (by factors of 2+) suggesting whoever gathered this category of data failed to obtain the correct numbers. It may also be the case that Facebook likes is a poor metric in determining popularity, in which case a different metric may be more appropriate such as the cast and actor’s total gross revenue to date.

Having such a low adjusted R-squared value implies the final model fails to accurately predict a movie’s gross revenue. Some remedies to improving model accuracy include using other variables that may influence a movie’s gross revenue. These additional variables may include the movie’s director, genre, technical specifications, production company, language, country of creation, etc. It may also be the case that multiple linear regression isn’t sufficient in predicting a movie’s gross revenue, in which case, a different statistical model may be more suitable.

Appendix

#read in data

> data <- read.csv("500movies.csv")

> data <- data.frame(data)

> attach(data)

#name variables

> length = data$duration

> a1pop = data$actor\_1\_facebook\_likes

> castpop = data$cast\_total\_facebook\_likes

> rating = data$imdb\_score

> budget = data$budget

> gross = data$gross

> votes = data$num\_voted\_users

#create MPAA dummy variables

> MPAA.dummies=dummy(content\_rating)

> MPAA.dummies

> MPAAG=MPAA.dummies[,"content\_ratingG"]

> MPAAPG=MPAA.dummies[,"content\_ratingPG"]

> MPAAPG13=MPAA.dummies[,"content\_ratingPG-13"]

> MPAAR=MPAA.dummies[,"content\_ratingR"]

>

> imdb = lm(gross~length+castpop+rating+budget+votes+MPAAPG+MPAAPG13+MPAAR)

> summary(imdb)

#begin Stepwise Regression

> alpha.in = 0.05

> alpha.out = 0.05

> n = 500

>

> model0 = lm(gross~1)

>

> model11 = lm(gross~length)

> model12 = lm(gross~castpop)

> model13 = lm(gross~rating)

> model14 = lm(gross~budget)

> model15 = lm(gross~votes)

> model16 = lm(gross~MPAAPG)

> model17 = lm(gross~MPAAPG13)

> model18 = lm(gross~MPAAG)

>

> f.stat11 = anova(model0,model11)[2,"F"]

> f.stat12 = anova(model0,model12)[2,"F"]

> f.stat13 = anova(model0,model13)[2,"F"]

> f.stat14 = anova(model0,model14)[2,"F"]

> f.stat15 = anova(model0,model15)[2,"F"]

> f.stat16 = anova(model0,model16)[2,"F"]

> f.stat17 = anova(model0,model17)[2,"F"]

> f.stat18 = anova(model0,model18)[2,"F"]

>

> f.stats1 = c(f.stat11,f.stat12,f.stat13,f.stat14,f.stat15,f.stat16,f.stat17,f.stat18)

> f.stats1

> F.in1 = qf(alpha.in, 1, n-2, lower.tail=FALSE)

> F.in1

#f.stat15 = 250.36 > F.in1 = 3.86 ==> add votes

> model01 = lm(gross~votes)

>

> model21 = lm(gross~votes+length)

> model22 = lm(gross~votes+castpop)

> model23 = lm(gross~votes+rating)

> model24 = lm(gross~votes+budget)

> model25 = lm(gross~votes+MPAAPG)

> model26 = lm(gross~votes+MPAAPG13)

> model27 = lm(gross~votes+MPAAG)

>

> f.stat21 = anova(model01,model21)[2,"F"]

> f.stat22 = anova(model01,model22)[2,"F"]

> f.stat23 = anova(model01,model23)[2,"F"]

> f.stat24 = anova(model01,model24)[2,"F"]

> f.stat25 = anova(model01,model25)[2,"F"]

> f.stat26 = anova(model01,model26)[2,"F"]

> f.stat27 = anova(model01,model27)[2,"F"]

>

> f.stats2 = c(f.stat21,f.stat22,f.stat23,f.stat24,f.stat25,f.stat26,f.stat27)

> f.stats2

>

> F.in2 = qf(alpha.in, 1, n-3, lower.tail=FALSE)

> F.in2

#f.stat25 = 46.09 > F.in2 = 3.86 ==> add MPAAPG

> modelaa = lm(gross~MPAAPG)

> modelab = lm(gross~votes+MPAAPG)

>

> f.stata = anova(modelaa,modelab)[2,"F"]

> f.stata

> F.out1 = qf(alpha.out, 1, n-3, lower.tail=FALSE)

> F.out1

#f.stata = 291.60 > F.out1 = 3.86 ==> dont drop any variables

> model02 = lm(gross~votes+MPAAPG)

>

> model31 = lm(gross~votes+MPAAPG+length)

> model32 = lm(gross~votes+MPAAPG+castpop)

> model33 = lm(gross~votes+MPAAPG+rating)

> model34 = lm(gross~votes+MPAAPG+budget)

> model35 = lm(gross~votes+MPAAPG+MPAAG)

> model36 = lm(gross~votes+MPAAPG+MPAAPG13)

>

> f.stat31 = anova(model02,model31)[2,"F"]

> f.stat32 = anova(model02,model32)[2,"F"]

> f.stat33 = anova(model02,model33)[2,"F"]

> f.stat34 = anova(model02,model34)[2,"F"]

> f.stat35 = anova(model02,model35)[2,"F"]

> f.stat36 = anova(model02,model36)[2,"F"]

>

> f.stats3 = c(f.stat31,f.stat32,f.stat33,f.stat34,f.stat35,f.stat36)

> f.stats3

>

> F.in3 = qf(alpha.in, 1, n-4, lower.tail=FALSE)

> F.in3

#f.stat36 = 27.04 > F.in3 = 3.86 ==> add MPAAPG13

> modela = lm(gross~votes+MPAAPG13)

> modelb = lm(gross~MPAAPG+MPAAPG13)

> modelc = lm(gross~votes+MPAAPG+MPAAPG13)

>

> f.stata = anova(modela,modelc)[2,"F"]

> f.statb = anova(modelb,modelc)[2,"F"]

>

> f.stats = c(f.stata,f.statb)

> f.stats

>

> F.out2 = qf(alpha.out, 1, n-4, lower.tail=FALSE)

> F.out2

#f.stata = 68.24 > F.out2 and f.statb = 301.52 > F.out2 ==> dont drop any variables

> model03 = lm(gross~votes+MPAAPG+MPAAPG13)

>

> model41 = lm(gross~votes+MPAAPG+MPAAPG13+length)

> model42 = lm(gross~votes+MPAAPG+MPAAPG13+castpop)

> model43 = lm(gross~votes+MPAAPG+MPAAPG13+rating)

> model44 = lm(gross~votes+MPAAPG+MPAAPG13+budget)

> model45 = lm(gross~votes+MPAAPG+MPAAPG13+MPAAG)

>

> f.stat41 = anova(model03,model41)[2,"F"]

> f.stat42 = anova(model03,model42)[2,"F"]

> f.stat43 = anova(model03,model43)[2,"F"]

> f.stat44 = anova(model03,model44)[2,"F"]

> f.stat45 = anova(model03,model45)[2,"F"]

>

> f.stats4 = c(f.stat41,f.stat42,f.stat43,f.stat44,f.stat45)

> f.stats4

>

> F.in4 = qf(alpha.in, 1, n-5, lower.tail=FALSE)

> F.in4

#f.stat44 = 6.25 > F.in4 = 3.86 ==> add budget

> modela = lm(gross~votes+MPAAPG+budget)

> modelb = lm(gross~votes+MPAAPG13+budget)

> modelc = lm(gross~MPAAPG+MPAAPG13+budget)

> model44 = lm(gross~votes+MPAAPG+MPAAPG13+budget)

>

> f.stata = anova(modela,model44)[2,"F"]

> f.statb = anova(modelb,model44)[2,"F"]

> f.statc = anova(modelc,model44)[2,"F"]

>

> f.stats = c(f.stata,f.statb,f.statc)

> f.stats

>

> F.out3 = qf(alpha.out, 1, n-5, lower.tail=FALSE)

> F.out3

#f.stata = 22.6 < F.out3 = 3.86 and f.statb = 62.6 < F.out3 and f.statc = 28#1.14 < F.out3 ==> dont drop any variables

>

> model04 = lm(gross~votes+MPAAPG+MPAAPG13+budget)

>

> model51 = lm(gross~votes+MPAAPG+MPAAPG13+budget+length)

> model52 = lm(gross~votes+MPAAPG+MPAAPG13+budget+rating)

> model53 = lm(gross~votes+MPAAPG+MPAAPG13+budget+castpop)

> model54 = lm(gross~votes+MPAAPG+MPAAPG13+budget+MPAAG)

>

> f.stat51 = anova(model04,model51)[2,"F"]

> f.stat52 = anova(model04,model52)[2,"F"]

> f.stat53 = anova(model04,model53)[2,"F"]

> f.stat54 = anova(model04,model54)[2,"F"]

>

> f.stats5 = c(f.stat51,f.stat52,f.stat53,f.stat54)

> f.stats5

>

> F.in5 = qf(alpha.in, 1, n-6, lower.tail=FALSE)

> F.in5

#f.stat54 = 12.77 > F.in4 = 3.86 ==> add MPAAG

> modela = lm(gross~votes+MPAAPG+budget+MPAAG)

> modelb = lm(gross~votes+MPAAPG13+budget+MPAAG)

> modelc = lm(gross~MPAAPG+MPAAPG13+budget+MPAAG)

> modeld = lm(gross~votes+MPAAPG+MPAAPG13+budget)

> model52 = lm(gross~votes+MPAAPG+MPAAPG13+budget+MPAAG)

>

> f.stata = anova(modela,model52)[2,"F"]

> f.statb = anova(modelb,model52)[2,"F"]

> f.statc = anova(modelc,model52)[2,"F"]

> f.statd = anova(modeld,model52)[2,"F"]

>

> f.stats = c(f.stata,f.statb,f.statc,f.statd)

> f.stats

>

> F.out4 = qf(alpha.out, 1, n-6, lower.tail=FALSE)

> F.out4

#f.stata = 28.11 > F.out4 = 3.86, f.statb = 70.0 > F.out4, f.statc = 287.5 > #F.out4, f.statd = 12.77 > F.out4 ==> dont drop any variables

> model05 = lm(gross~votes+MPAAPG+MPAAPG13+budget+MPAAG)

>

> model61 = lm(gross~votes+MPAAPG+MPAAPG13+budget+MPAAG+length)

> model62 = lm(gross~votes+MPAAPG+MPAAPG13+budget+MPAAG+rating)

> model63 = lm(gross~votes+MPAAPG+MPAAPG13+budget+MPAAG+castpop)

>

> f.stat61 = anova(model05,model61)[2,"F"]

> f.stat62 = anova(model05,model62)[2,"F"]

> f.stat63 = anova(model05,model63)[2,"F"]

>

> f.stats6 = c(f.stat61,f.stat62,f.stat63)

> f.stats6

>

> F.in6 = qf(alpha.in, 1, n-7, lower.tail=FALSE)

> F.in6

#f.stat61 = 0.136 < F.in6 = 3.86, f.stat62 = 0.0029 < F.in6, f.stat63 =0.0014 #< F.in6 ==> dont add anymore variables. Stepwise regression complete

#begin best subset procedure

> best.subset.cp=leaps(x=cbind(length,castpop,rating,budget,votes,MPAAPG,MPAAPG13,MPAAG), y=gross, method="Cp")

> min.cp.value=min(best.subset.cp$Cp)

>

> min.cp.location=which.min(best.subset.cp$Cp)

>

> best.subset.cp$which[min.cp.location,]

1 2 3 4 5 6 7 8

FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE

#according to best subset procedure the best model is: gross~castpop+budget+votes+MPAAPG13+MPAAR

> imdbfinal1 = lm(gross~votes+MPAAPG+MPAAPG13+budget+MPAAG)

> imdbfinal2 = lm(gross~votes+castpop+MPAAPG+MPAAPG13+budget+MPAAG)

> anova(imdbfinal1,imdbfinal2)

> imdbfinal = lm(gross~votes+MPAAPG+MPAAPG13+budget+MPAAG)

#identify any influential observations

> cutoff <- 0.5

> plot(imdbfinal, which=4, cook.levels=cutoff)

#observation 454 has a cooks distance of nearly 150, remove it.

> rating.new = rating[-454]

> gross.new = gross[-454]

> castpop.new = castpop[-454]

> budget.new = budget[-454]

> votes.new = votes[-454]

> MPAAG.new = MPAAG[-454]

> MPAAPG.new = MPAAPG[-454]

> MPAAPG13.new = MPAAPG13[-454]

> MPAAR.new = MPAAR[-454]

> imdbfinal.new = lm(gross.new~votes.new+MPAAPG.new+MPAAPG13.new+budge.> new+MPAAG.new)

> summary(imdbfinal.new)

#adjust r-squared increases from .397 to .549 after dropping obs. #454

#check for high influence measures

> h.model2 = influence.measures(imdbfinal.new)$infmat[,"hat"]

> plot(h.model2, main="Plot of h\_i", ylab="Leverage", xlab="Observation Number", xaxt = "n")

> axis(1, at=1:499)

#no high leverage observations over 0.10

#Begin checking for constant variance and homoscedasticity

> Resid = imdbfinal.new$residuals

> y.hat = predict(imdbfinal.new)

> plot(resid~y.hat, main="Plot of Residuals vs. Fitted Values", xlab="Fitted > Values", ylab="Residuals")

> abline(h=0)

> plot(resid~budget.new, main="Plot of Residuals vs. budget", xlab="budget", > ylab="Residuals")

> abline(h=0)

> plot(resid~votes, main="Plot of Residuals vs. number of IMDb votes", xlab="> IMDb vote count", ylab="Residuals")

> abline(h=0)

> hist(resid, main="Histogram of Residuals", xlab="Residuals")

> qqnorm(resid, main="Normal Probability Plot of Residuals")

> qqline(resid)

> shapiro.test(resid)

# p-value < 2.2e-16 ==> e isn’t normally distributed

> e.star = studres(imdbfinal.new)

> plot(e.star~y.hat, ylim=c(-5,5), ylab="Studentized Residuals",

xlab="Fitted Values", main="Plot of Studentized Residuals vs. Fitted Val> ues")

> abline(h=2, col="blue", lty=2)

> abline(h=-2, col="blue", lty=2)

> abline(h=0)

> plot(gross~votes.new, main="Plot of Residuals vs. Fitted Values", xlab="Fit> ted Values", ylab="Residuals")

> abline(h=0)

> plot(gross~budget.new, main="Plot of Residuals vs. Fitted Values", xlab="Fi> tted Values", ylab="Residuals")

> abline(h=0)

#apparent patterns are observed, need to transform response variable

> imdbfinaltrans.new = lm(gross.new^(1/3)~votes.new+MPAAPG.new+MPAAPG13.new+b> udget.new+MPAAG.new)

#used plot codes written above to recheck model assumptions, everything checks out for the most part

> summary(imdbfinaltrans.new)

> anova(imdbfinaltrans.new)

#find prediction interval for ‘Black Panther’ movie

> newdata = data.frame(votes.new = 320181, MPAAG.new = 0, MPAAPG13.new = 1, M> PAAPG.new = 0, budget.new = 200000000)

> pred.conf = predict(imdbfinal, newdata, interval="prediction", level=0.95)

> pred.conf