Data Analysis on Movie's Gross

Group Members
Ming Zhao (mezhao@ucdavis.edu)

Abstract

The analysis is based on the US movies over the last three decades. I am interest in exploring what increases movie gross and how the movie preferences change over time. By conducting data preprocessing, exploratory data analysis, preliminary fit, model selection, model validation, model diagnostics, and bootstrap, my final regression model shows that higher budget and better directors increase gross, unsurprisingly. Releasing movies in summer and winter may help to increase gross. Horror movies become the most popular genre since the 1990's, animation and adventure movies behave increasingly better than other genres recently, in terms of gross. In addition, increasing budget on animation brings significantly more gross while the rate of return for horror movies is the lowest among all the movies. PG and PG-13 are also more popular than the other ratings.

Introduction

Over the last several decades, the US movie industry has experienced significant changes as the constant improvement in streaming platforms create competition for market share in the movie industry. I believe that gross best describes success in the movie industry and provides crucial information regarding financial performance of the business. Therefore, I am interested in exploring the relationships between gross and movie features, such as budget, genre, rating, year, and others. Especially, I would like to know how the effects of genre and rating on gross change as budget increases and time goes by.

The analysis relies on the data of global movies from 1980 to 2020, which scraped from IMDb and posted on Kaggle by Daniel Grijalva. The original dataset contains 7668 observations and 15 variables combined with 6 quantitative and 9 categorical variables. The variables include gross, budget, runtime, rating, genre, score, country, director, and release date.

To explore what affects movie gross and how the effects change, I first preprocess the data and conduct exploratory data analysis. I then examine preliminary findings followed by model selection, model validation, and model diagnostics. Finally, I perform bootstrapping to see the random variation in the results.

Methods and Results

1. *Data Preprocessing*

First, I restrict my data to the movies released before 2020 and produced by US companies due to much fewer observations in 2020 and of other countries. Therefore, the analysis focuses on US movies throughout three decades from1980 to 2019. Second, I remove some variables from the dataset, such as name, writer, star, and company, as they have considerable variation. I also exclude votes from the analysis as it is an endogenous variable. However, to keep movie's information as much as possible, I generate a new variable, director tier; it has levels, A, B, and C, which grouped by the average score a director received for all his or her movies. I believe this categorical variable is a proxy for the capability or popularity of the director and further affects gross. In addition, based on the released date, I create a categorical variable, season, in which March, April, and May are categorized as spring, June, July, August as summer, September, October, November as fall, and the rest as winter. Third, I find the dataset has a great number of missing values due to the variable, budget. Since the missingness is not completely random, I decide to compute the missing values with the single imputation for the purpose of better practice of what I learn from this course. Finally, I have a full dataset with 5345 observations and 8 variables. In addition, I split the data into training and validation sets in the ratio 50:50 for the later model validation.

2. Exploratory Data Analysis

Among the 8 variables, one half is quantitative, and the other half is qualitative. Figure 1 displays the distributions of quantitative variables, gross, budget, runtime, and year. Except year, all the other variables are right-skewed. Looking at their scatter plots, shown in Figure 2, there is no obvious nonlinear relationship between gross and budget and they are highly correlated. Rating, genre, director tier, and season are the four qualitative variables, and their illustrations are pie charts in the Figure 3. As shown, NC-17/R and PG-13 are the two largest ratings; comedy, action, and drama are the three largest genres; for tier, the majority of directors belong to B while the minority to A; four seasons have similar proportions although winter's is slightly smaller. Looking at the side-by-side box plots in Figure 4, it shows that animation and tier A have noticeable higher gross than their counterparts.

3. Preliminary Fit

Based on the training data, I first regress budget, runtime, year, rating, genre, director tier, and released season onto gross. By performing model diagnostics, I find the residuals are not normally distributed. I then apply the Box-Cox procedure and find that lambda is 0, so I take a log transformation on gross, referring to log(gross) thereafter. Also, I take the log transformation on budget and runtime because they have right-skewed distributions as well. Repeating the procedures of model fitting, model diagnostics, and Box-Cox transformation with the new variables, the results show that residuals are still non-normal, and lambda is 2 this time. I therefore raise log(gross) to the second power and conduct the whole process with the updated variable; again, the violation of the normality assumption is still there, and lambda is close to 2. As a result, I raise log(gross) to the fourth power, and this solves the non-normality problem of the residuals. Therefore, the model assumption of normality generally holds for the last regression model. All the results about the transformation procedures are shown in Figure 5. For the last model, there exists an unobvious problem of unequal variance, but I believe the model assumptions hold reasonably.

Then, I make a scatter plot matrix on the newly constructed quantitative variables in Figure 6, and it shows that there is no obvious nonlinearity among the variables. Additionally, I plot residuals of the last fit against interaction terms in Figure 7. From the observation, it is possible to include interaction terms between quantitative variables as there might be some unclear linear patterns between log(budget) and log(runtime) as well as log(budget) and year.

4. Model Selection

In this step, I apply forward stepwise procedure on the training data to select a good model that balance bias and variance, with the consideration of both first-order models and models with interaction terms. The initial model has 0 predictor, while the full first-order model contains 7 predictors and the full model with interaction terms contains 29 predictors. In my case, the number of coefficients for the full first-order model is 20 and for the full model with interaction terms is 156. The criteria used for model selection are AIC and BIC, in which the smaller, the better. Consequently, the procedure returns 3 candidate models. First, AIC and BIC are identical in selecting the first-order model and they find the best model as below:

$$log(gross)_i^{\frac{1}{4}} = \beta_0 + \beta_1 log(budget)_i + \beta_2 log(runtime)_i + \beta_3 year_i + \beta_4 rating_i + \beta_5 genre_i + \beta_6 tier_i + \beta_7 season_i + \varepsilon_i$$

Second, for the models with interaction terms, AIC finds the following:

$$\begin{aligned} log(gross)_i^{\frac{1}{4}} = {} & \beta_0 + \beta_1 log(budget)_i + \beta_2 log(runtime)_i + \beta_3 year_i + \beta_4 rating_i + \beta_5 genre_i + \beta_6 tier_i + \beta_7 season_i \\ & + \beta_8 log(budget)_i * log(runtime)_i + \beta_9 log(budget)_i * year_i + \beta_{10} log(budget)_i * rating_i + \beta_{11} log(budget)_i * genre_i + \beta_{12} log(budget)_i * tier_i \\ & + \beta_{13} log(runtime)_i * year_i + \beta_{14} log(runtime)_i * genre_i + \beta_{15} log(runtime)_i * season_i \\ & + \beta_{16} year_i * rating_i + \beta_{17} year_i * genre_i + \beta_{18} year_i * season_i \\ & + \beta_{19} rating_i * tier_i + \varepsilon_i \end{aligned}$$

Third, in contrast to AIC, BIC gives the following as the best model with interaction terms:

$$\begin{aligned} log(gross)_i^{\frac{1}{4}} = {} & \beta_0 + \beta_1 log(budget)_i + \beta_2 log(runtime)_i + \beta_3 year_i + \beta_4 rating_i + \beta_5 genre_i + \beta_6 tier_i + \beta_7 season_i \\ & + \beta_8 log(budget)_i * log(runtime)_i + \beta_9 log(budget)_i * year_i + \beta_{10} log(runtime)_i * year_i + \varepsilon_i \end{aligned}$$

5. *Model Validation*

For this part, I perform both internal validation using the training data and external validation using the validation data. In terms of internal validation, I compute C_p and $Press_p$, and then compare C_p with p as well as $Press_p$ with SSE_p for each candidate model. Specifically,

Model 1: $Press_p = 1.80 \times 10^{12}, C_p = 348.06, SSE_p = 1.77 \times 10^{12}, p = 20, MSPE_v = 6.48 \times 10^8, MSE_p = 6.67 \times 10^8$

Model 2: $Press_p = 1.65 \times 10^{12}, C_p = 82.78 \ \ , SSE_p = 1.56 \times 10^{12}, p = 67, MSPE_v = 6.00 \times 10^8, MSE_p = 5.97 \times 10^8$

Model 3: $Press_p = 1.68 \times 10^{12}, C_p = 151.56, SSE_p = 1.65 \times 10^{12}, p = 23, MSPE_v = 6.03 \times 10^8, MSE_p = 6.22 \times 10^8$

If C_p is close to p, the model has little bias; among the three models, Model 2 has the least bias as its C_p is similar to p the most although they still have big difference. If $Press_p$ is not much larger than SSE_p, there is no severe over-fitting; since the model bias is relatively high for all the models, no over-fitting problem is detected.

In terms of external validation, I first fit same regression models on the validation data, and then compute mean square prediction error, $MSPE_v$, and compare it with SSE_p/n. If $MSPE_v$ is not much larger than SSE_p/n, there is no severe over-fitting; again, the three models do not have the over-fitting problem. Since Model 2 has the smallest $MSPE_v$, its predictive ability is the best compared to the other two. As a result, Model 2 is the best model I can have relying on the training and validation data.

6. *Model Diagnostics*

I conduct model diagnostics for the final model on the entire dataset. The diagnostic plots in Figure 8 shows that although the residuals are generally normal distributed, there is a little non-constancy in variance probably due to outliers. I therefore use Cook's distance to identify the influential cases. Before the influential cases, I use studentized deleted residuals and Bonferroni outlier test procedure to identify outlying Y and use leverage and the value of $2p/n$ to identify outlying X. The results show that there are 548 cased defined as outlying X observations while only 1 case for Y. According to Cook's distance. I find 3 influential cases. Removing the 3 cases, I refit the model, but there is no obvious change based on the diagnostic plots in Figure 9. The average absolute percent difference in the fitted values with and without the most influential cases is 1%, so they do not have very large influence on prediction, and I decide to retain all the cases.

7. *Bootstrap*

As there is slight non-constancy in residual variance, I perform bootstrap by resampling the original data to have better standard errors. The distribution of coefficients is normal. Figure 10 displays the distribution of coefficients for log(budget) and year, for example.

Conclusion and Discussion

Following the procedures: data preprocessing, exploratory data analysis, preliminary fit, model selection, model validation, model diagnostics, and bootstrap, my final regression model contains 19 X variables with 7 main effects and 12 second order interactions. In addition, the model assumption of residuals about normality and constant variance reasonably holds by examining the diagnostic plots of the final model. However, due to the cumulative transformations, log and fourth power, on gross and the log transformation on budget and runtime, along with the interaction terms, the results become complicated to interpret. Therefore, I use the interaction plots combined with the model parameters to answer my questions of interest in the first place.

Generally, budget, runtime, and year positively affect gross; compared with tier A, tier B and C tend to decrease gross; compared with fall, summer and winter tend to increase gross. In terms of movie rating and genre, my interpretation is based on the interaction plots. In Figure 11, it shows the effect of interaction between budget and rating on gross. As shown, the interaction effect is not significant because different ratings have similar effects on gross as budget increases. In Figure 12, as budget increases, all genres have positive effects on gross; in contrast, animation has the greatest effects while horror has the smallest effects compared with other genres. Figure 13 shows gross against the interaction between rating and year. Based on the observation, G movies are less preferred as time goes by, but such effect is not as significant as it shows since the number of G movies is limited in the dataset. PG and PG-13 become more and more popular over the last three decades; R and NC-17 movies also become more popular while their effect on gross increases slowly. In Figure 14, the overall effects of interaction between genre and year on gross are positive. Particularly, horror movies become the most popular than all the others since the 1990's in terms of gross. Interestingly, action movies which brought the most gross during the 1980's become less popular than most of the others today. Since movies of other genres have limited observation, its effects are not significant. In addition, adventure and animation movies behave better on gross from years to years.

One of the limitations of this study is the great number of missing values in budget. Although I use the single imputation to compute the missing values, the results possible remain biased somehow. Second limitation is that the dataset does not contain all the important features of the movies and therefore the model bias is relatively higher. The third one is about the sampling; I doubt the sample distribution of year is representative.

Appendix 1

Figure 1: Histograms of Quantitative Variables

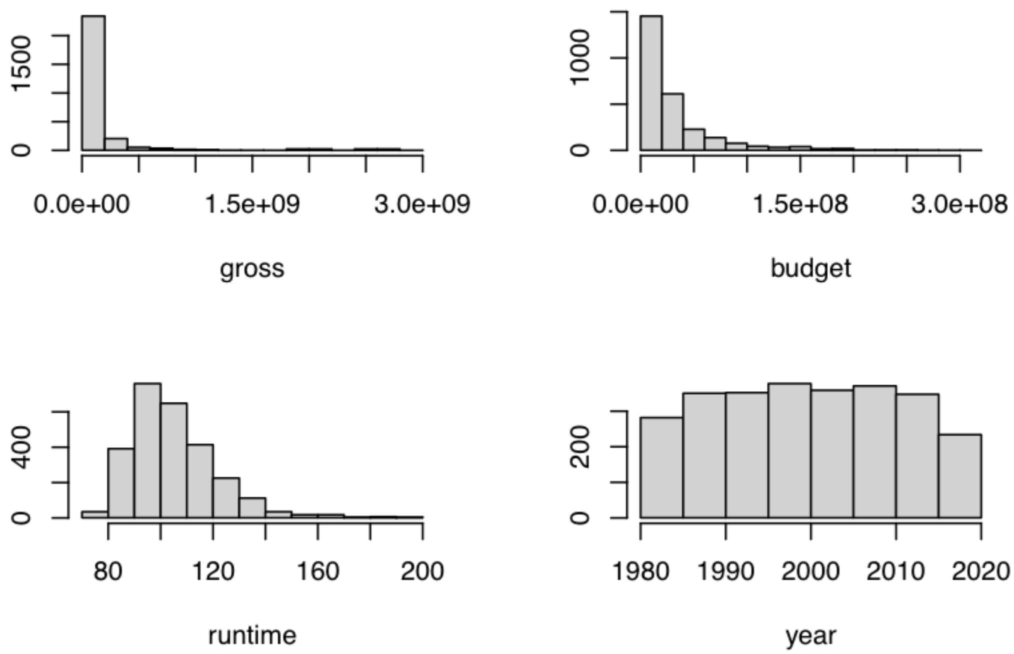


Figure 2: Scatter Plot Matrix of Quantitative Variables

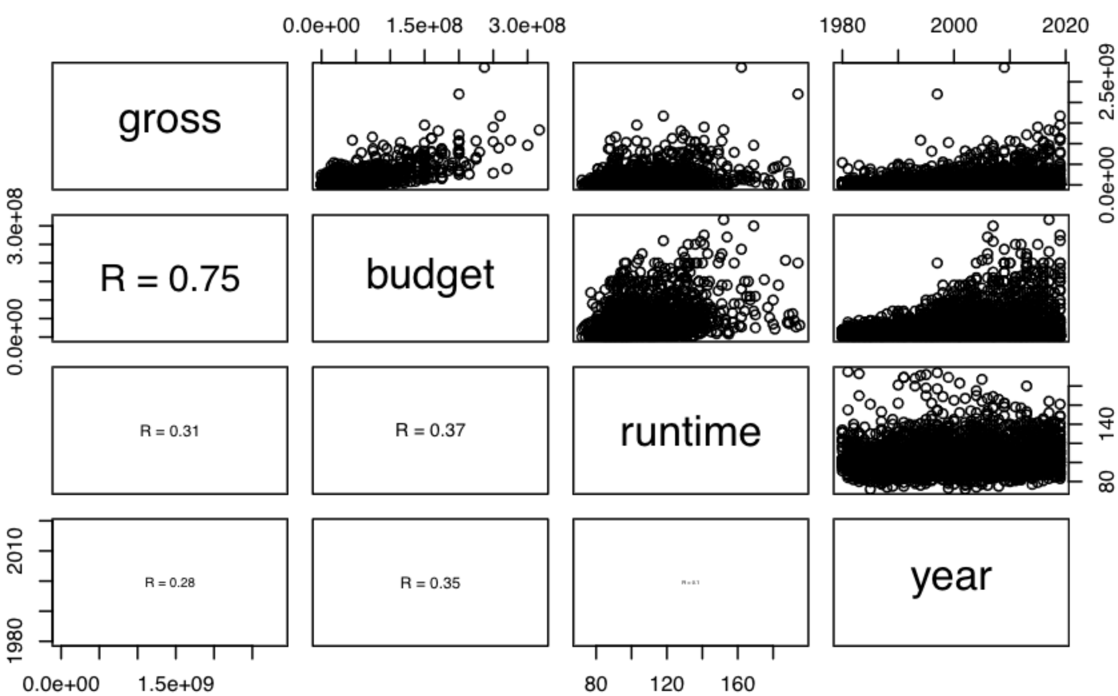


Figure 3a: Pie Charts of Qualitative Variables

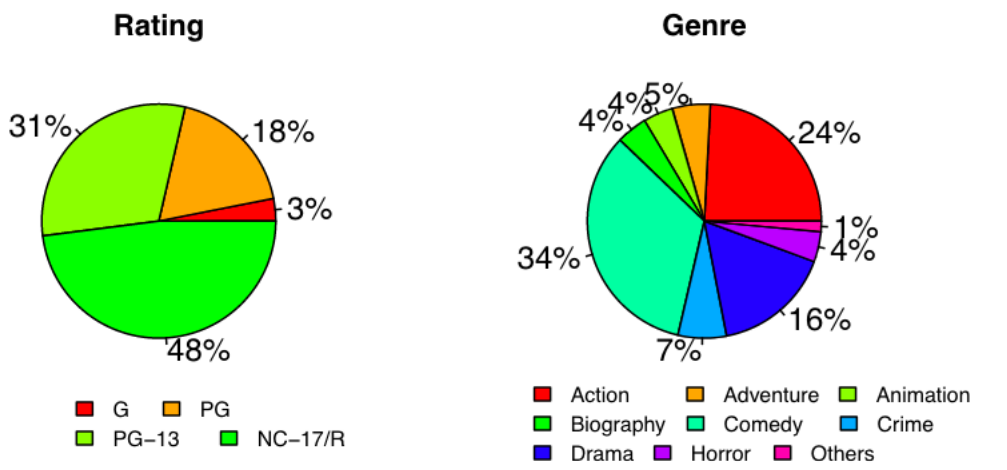


Figure 3b: Pie Charts of Qualitative Variables

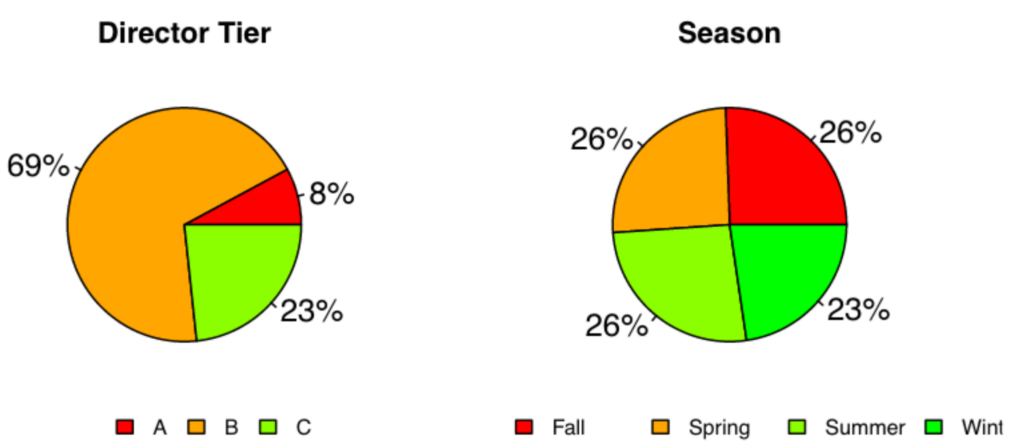


Figure 4a: Side–by–Side Box Plots

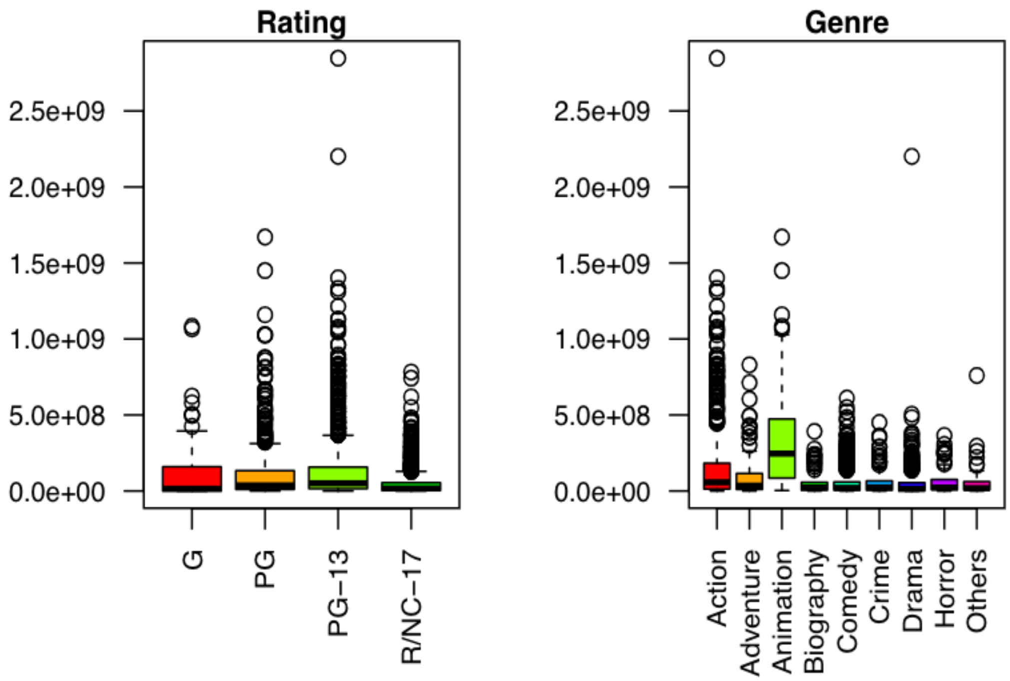


Figure 4b: Side–by–Side Box Plots

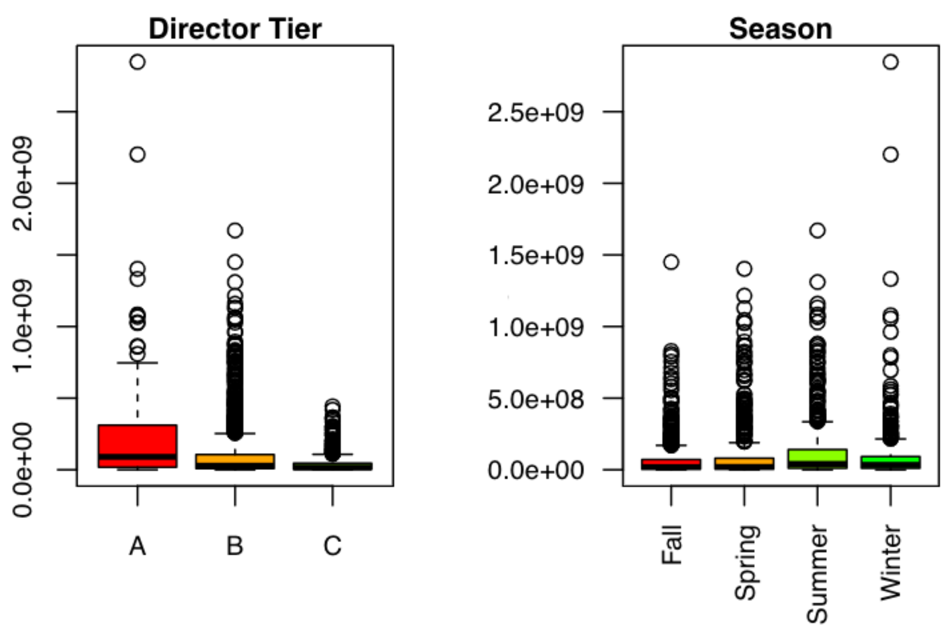


Figure 5a: Preliminary Regression of gross

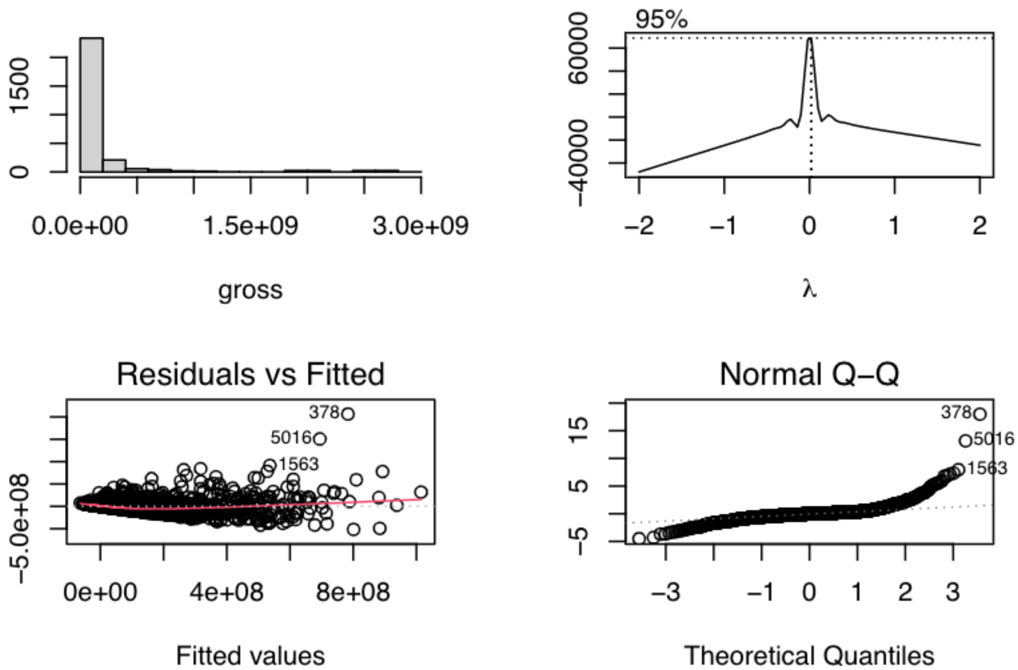


Figure 5b: Preliminary Regression of log(gross)

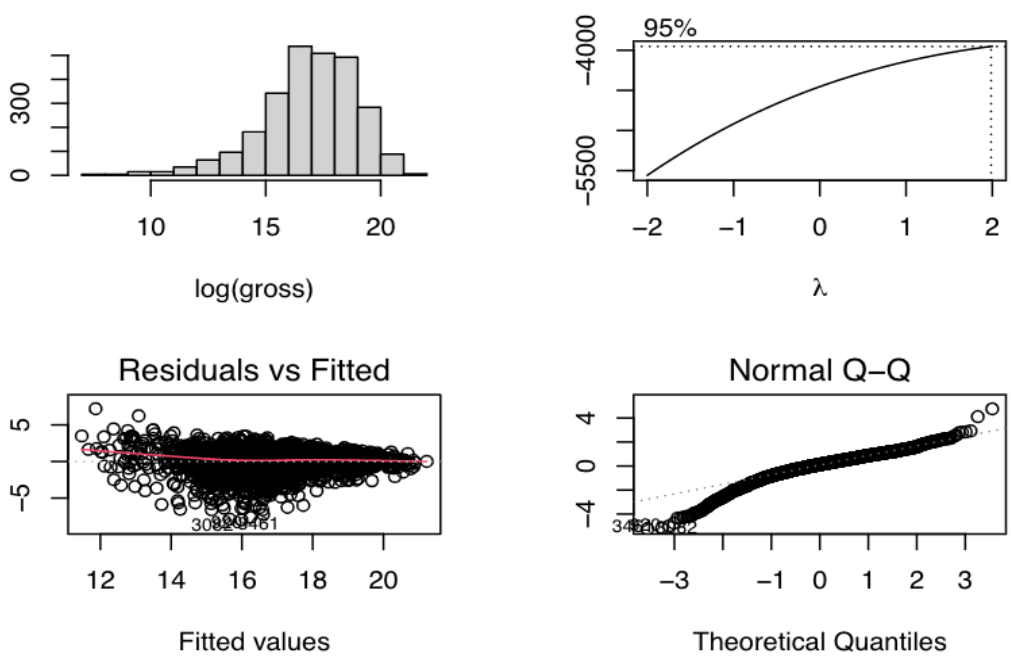


Figure 5c: Preliminary Regression of $\log(\text{gross})^2$

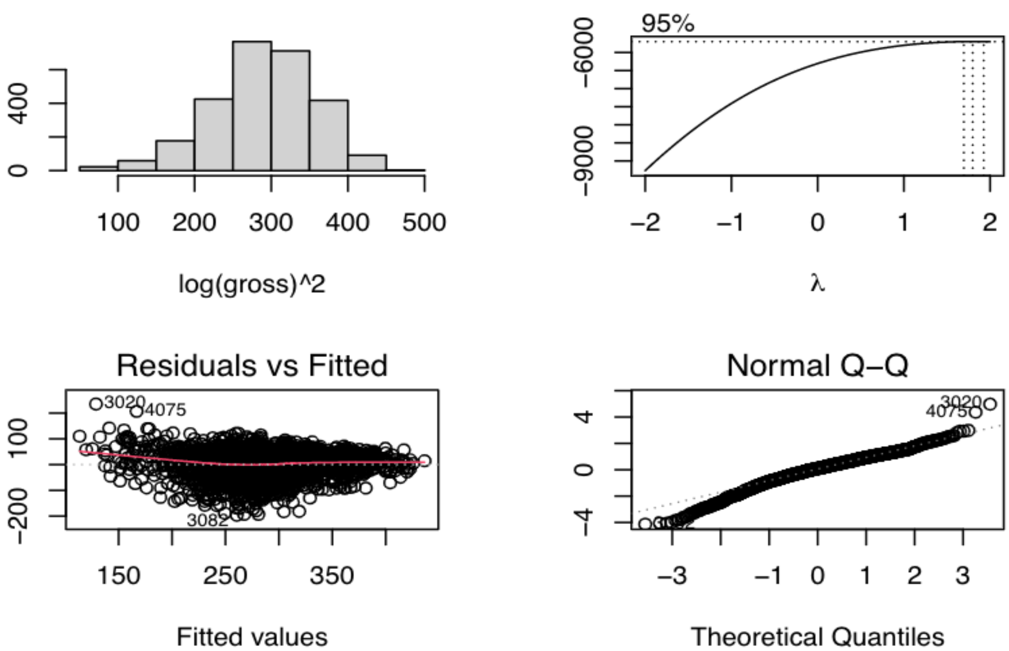


Figure 5d: Preliminary Regression of $\log(\text{gross})^4$

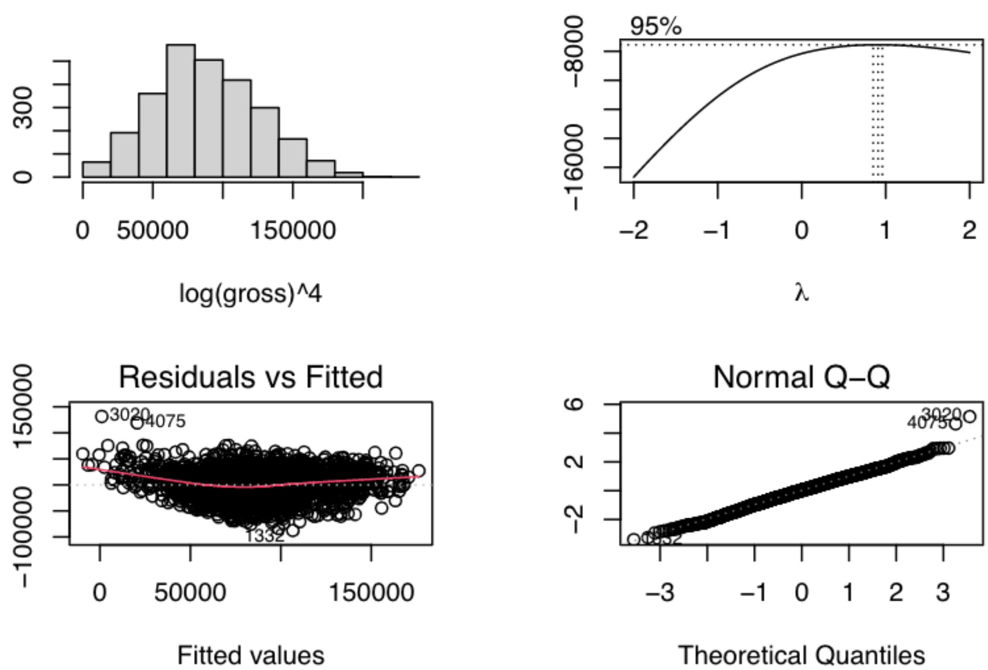


Figure 6: Scatter Plot Matrix of $\log(\text{gross})^4$ and Transformed X Variables

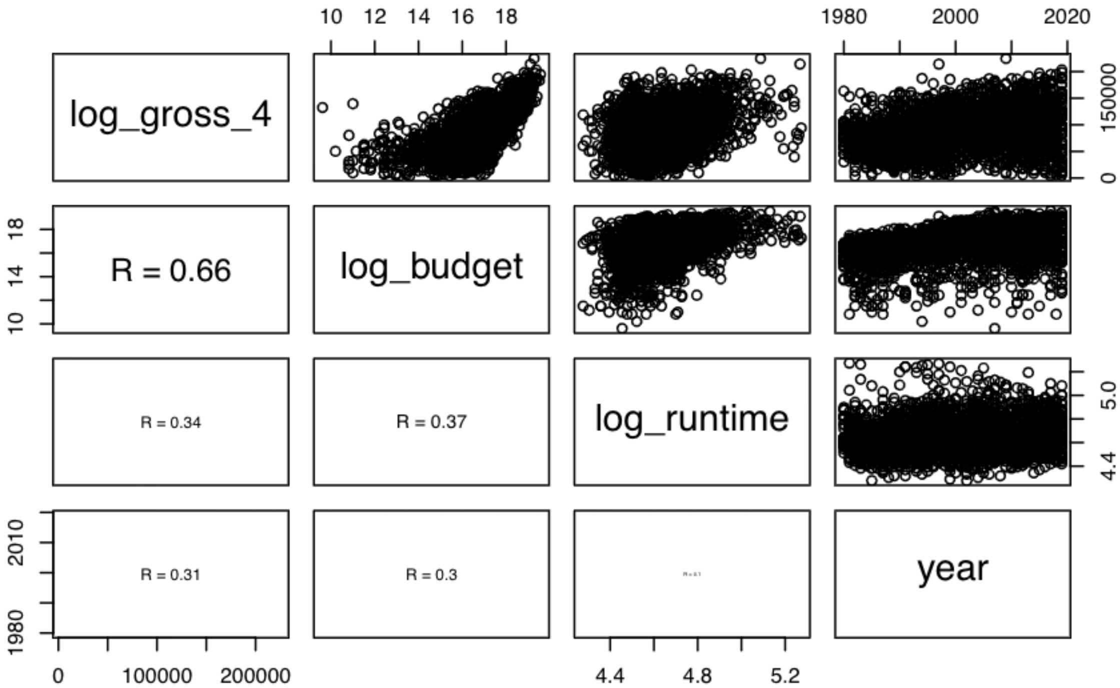


Figure 7: Model on $\log(\text{gross})^4$: Residuals vs. Interaction Terms

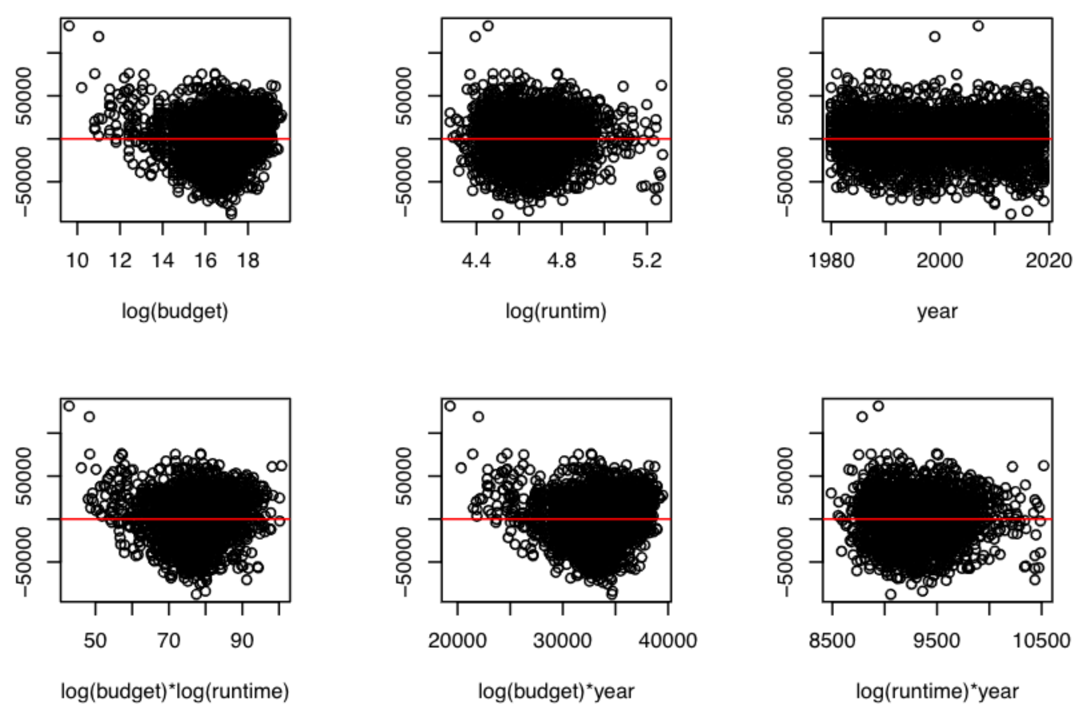


Figure 8: Diagnostic Plots for Final Model

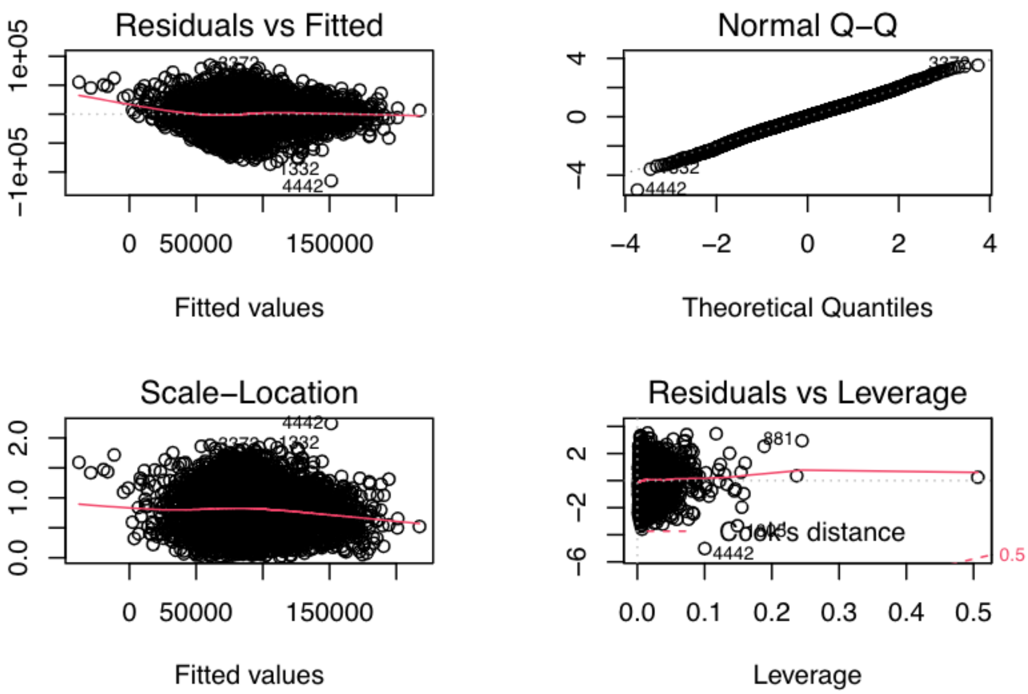


Figure 9: Cook's Distance

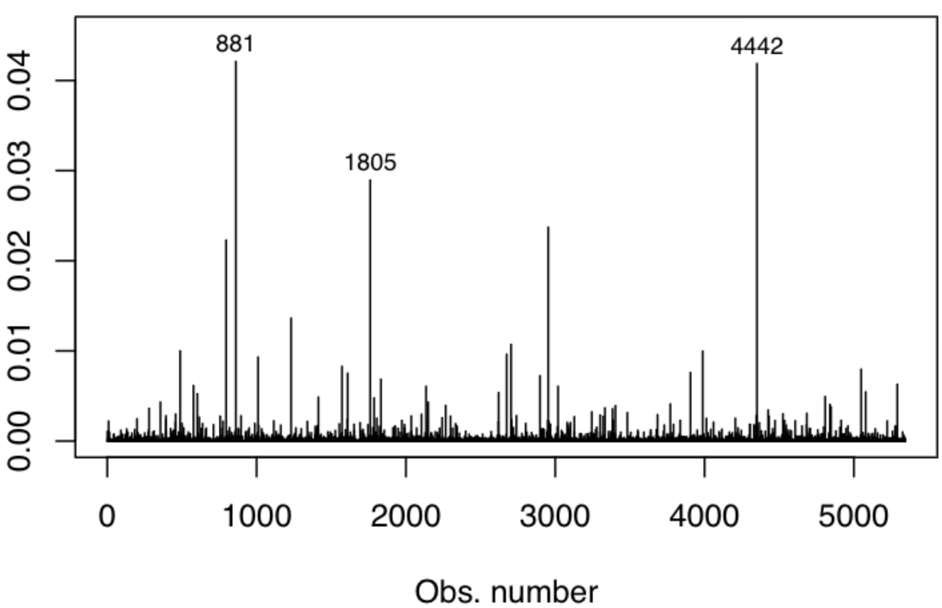
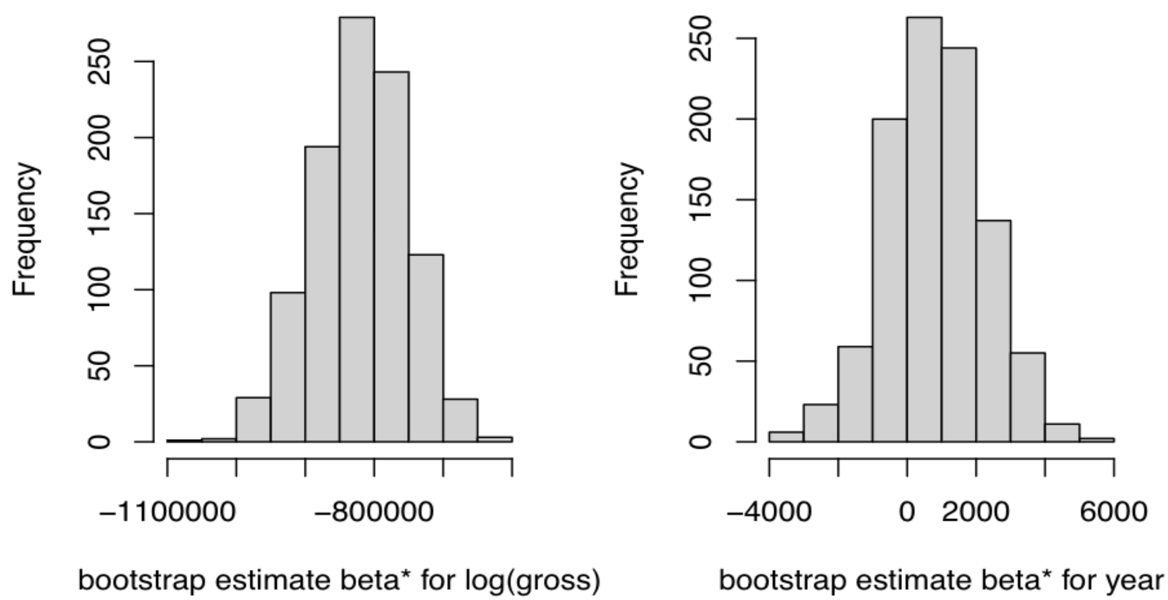


Figure 10: Bootstrap Estimate Coefficients



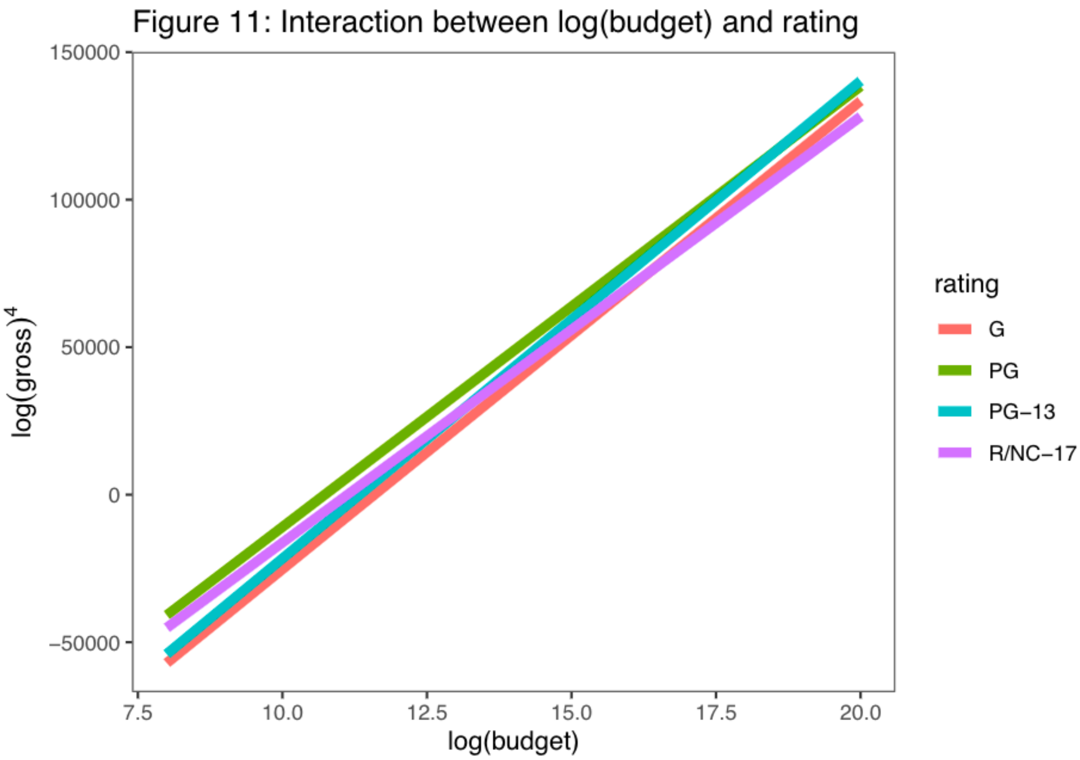


Figure 11: Interaction between log(budget) and rating
150000
100000
50000
0
−50000
log(gross)^4
7.5
10.0
12.5
15.0
17.5
20.0
log(budget)
rating
G
PG
PG−13
R/NC−17

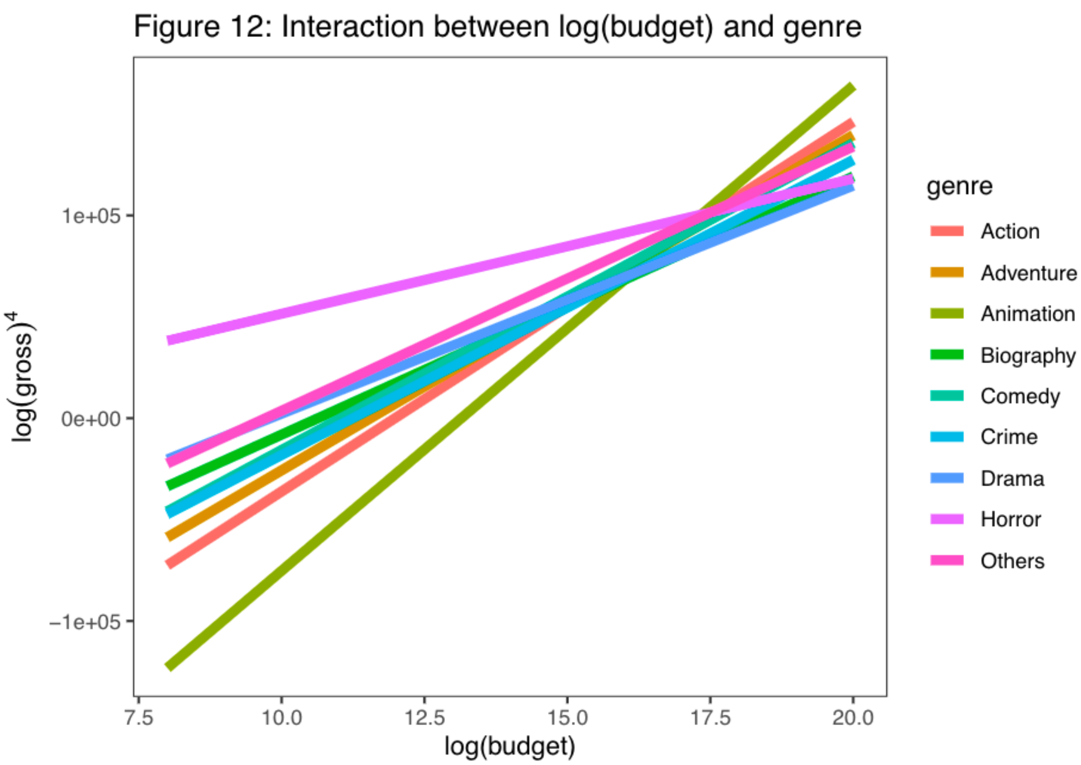


Figure 12: Interaction between log(budget) and genre
1e+05
0e+00
−1e+05
log(gross)^4
7.5
10.0
12.5
15.0
17.5
20.0
log(budget)
genre
Action
Adventure
Animation
Biography
Comedy
Crime
Drama
Horror
Others

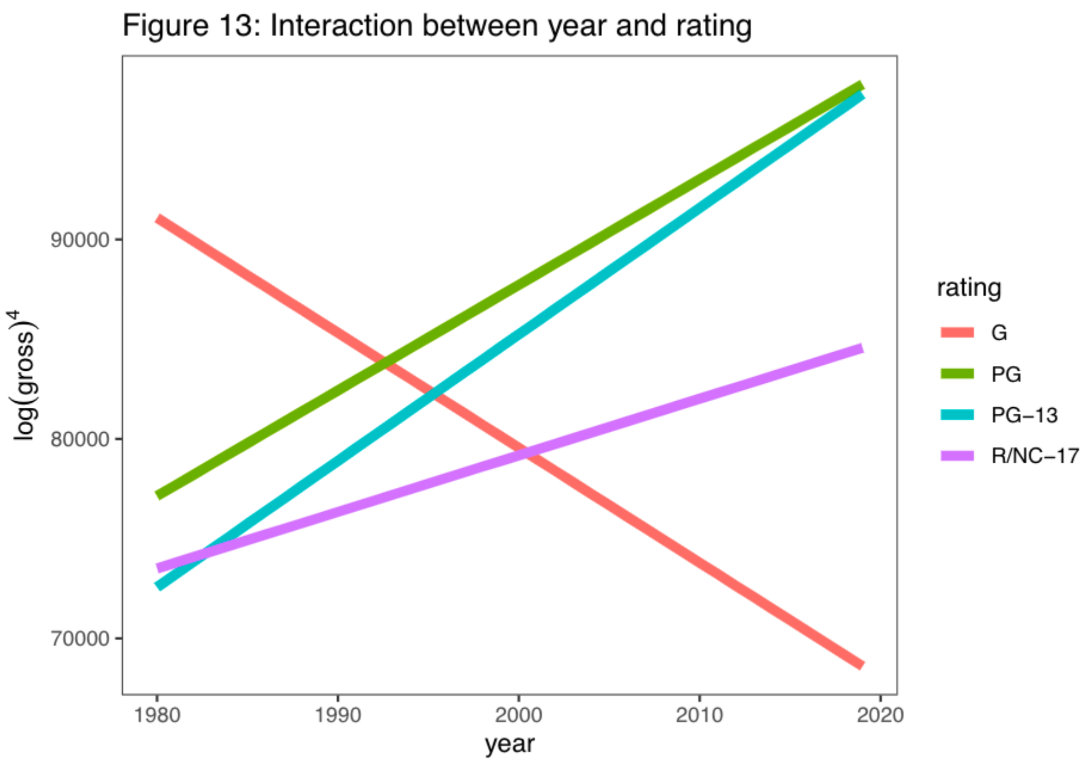


Figure 13: Interaction between year and rating
90000
80000
70000
log(gross)^4
1980
1990
2000
2010
2020
year
rating
G
PG
PG−13
R/NC−17

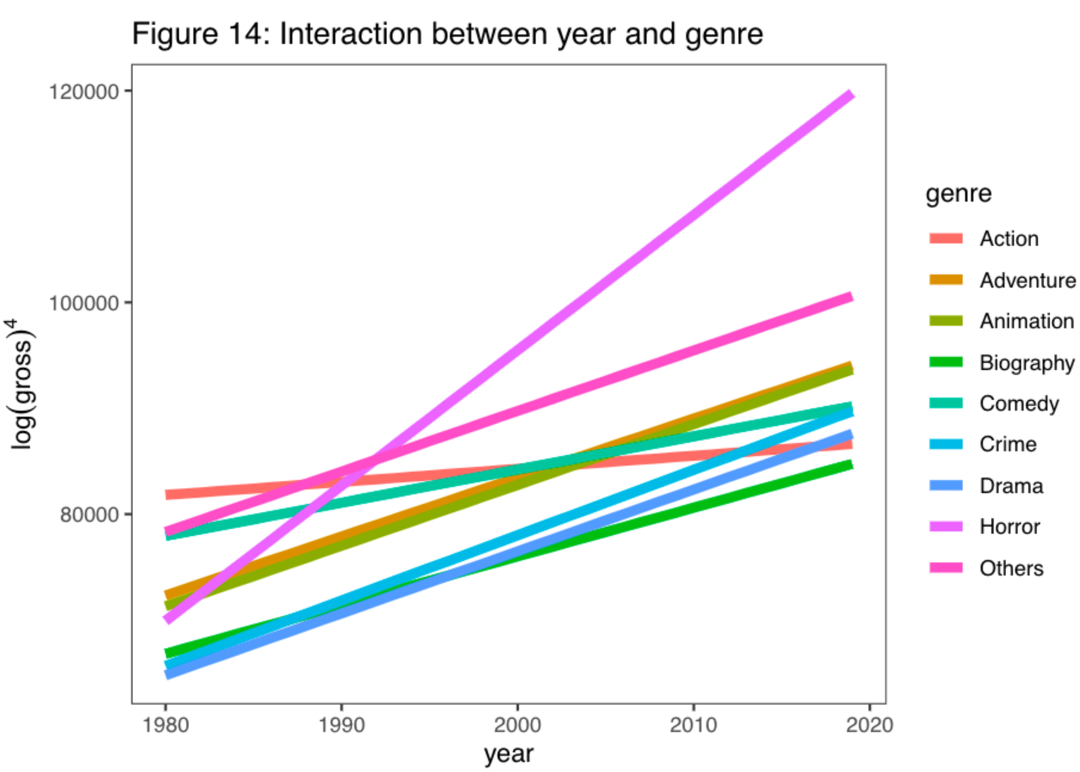


Figure 14: Interaction between year and genre
120000
100000
80000
log(gross)^4
1980
1990
2000
2010
2020
year
genre
Action
Adventure
Animation
Biography
Comedy
Crime
Drama
Horror
Others

Appendix 2

Data Preprocessing

```
setwd("/Users/mingzhao/Desktop")

movies <- read.csv("movies.csv")

dim(movies)
```

```
## [1] 7668   15
```

```
sapply(movies, class)
```

```
##        name      rating       genre        year    released       score
## "character" "character" "character"   "integer" "character"   "numeric"
##       votes    director      writer        star     country      budget
##   "numeric" "character" "character" "character" "character"   "numeric"
##       gross     company     runtime
##   "numeric" "character"   "numeric"
```

```
sum(is.na(movies))
```

```
## [1] 2370
```

```
sapply(movies, function(x) sum(is.na(x)))
```

```
##     name   rating    genre     year released    score    votes director
##        0        0        0        0        0        3        3        0
##   writer     star  country   budget    gross  company  runtime
##        0        0        0     2171      189        0        4
```

```
##################################################################

# Y
hist(movies$gross)
```

```
#0.USA
length(unique(movies$country))
```

```
## [1] 60
```

```
movies$USA <- ifelse(movies$country=="United States", 1, 0)
table(movies$USA)
```

```
##
##    0    1
## 2193 5475
```

```
#1.year
table(movies$year)
```

```
##
## 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995
```

```
##   92  113  126  144  168  200  200  200  200  200  200  200  200  200  200  200
## 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
##  200  200  200  200  200  200  200  200  200  200  200  200  200  200  200  200
## 2012 2013 2014 2015 2016 2017 2018 2019 2020
##  200  200  200  200  200  200  200  200   25
```

```
is.na(movies$year) <- which(movies$year==2020)
#movies$period[movies$year>1989 & movies$year<2000] <- "1990-1999"
#movies$period[movies$year>1999 & movies$year<2010] <- "2000-2009"
#movies$period[movies$year>2009 & movies$year<2020] <- "2010-2019"
table(movies$year)
```

```
##
## 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995
##   92  113  126  144  168  200  200  200  200  200  200  200  200  200  200  200
## 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
##  200  200  200  200  200  200  200  200  200  200  200  200  200  200  200  200
## 2012 2013 2014 2015 2016 2017 2018 2019
##  200  200  200  200  200  200  200  200
```

```
#2.budget
hist(movies$budget)
```

```
#3.genre
table(movies$genre)
```

```
##
##    Action Adventure Animation Biography    Comedy     Crime     Drama    Family
##      1705       427       338       443      2245       551      1518        11
##   Fantasy   History    Horror     Music   Musical   Mystery   Romance    Sci-Fi
##        44         1       322         1         2        20        10        10
##     Sport  Thriller   Western
##         1        16         3
```

```
movies$genre[movies$genre=="Family"   |
             movies$genre=="Fantasy"  |
             movies$genre=="History"  |
             movies$genre=="Music"    |
             movies$genre=="Musical"  |
             movies$genre=="Mystery"  |
             movies$genre=="Romance"  |
             movies$genre=="Sci-Fi"   |
             movies$genre=="Sport"    |
             movies$genre=="Thriller" |
             movies$genre=="Western"] <- "Others"
movies$genre <- as.factor(movies$genre)
table(movies$genre)
```

```
##
##    Action Adventure Animation Biography    Comedy     Crime     Drama    Horror
##      1705       427       338       443      2245       551      1518       322
##    Others
##       119
```

```
#4.rating
table(movies$rating)
```

```
##
##           Approved         G     NC-17 Not Rated        PG     PG-13         R
##        77         1       153        23       283      1252      2112      3697
##     TV-14     TV-MA     TV-PG   Unrated         X
##         1         9         5        52         3
```

```
movies$rating[movies$rating==""         |
              movies$rating=="Approved"  |
              movies$rating=="Not Rated" |
              movies$rating=="Unrated"] <- "G"
movies$rating[movies$rating=="X"        |
              movies$rating=="TV-MA"     |
              movies$rating=="NC-17"     |
              movies$rating=="R"] <- "R/NC-17"
movies$rating[movies$rating=="TV-PG"] <- "PG"
movies$rating[movies$rating=="TV-14"] <- "PG-13"
movies$rating <- as.factor(movies$rating)
table(movies$rating)
```

```
##
##       G      PG   PG-13 R/NC-17
##     566    1257    2113    3732
```

```
#5.runtime
summary(movies$runtime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    55.0    95.0   104.0   107.3   116.0   366.0       4
```

```
#6.season/month
movies$released <- as.character(movies$released)
movies$released <- gsub("\\(..*", " ", movies$released)
movies$released <- as.Date(movies$released, "%B %d, %Y")
movies$month=as.integer(lubridate::month(movies$released))
movies$season=ifelse(movies$month %in% c(12,1,2),"Winter",
              ifelse(movies$month %in% c(3,4,5),"Spring",
              ifelse(movies$month %in% c(6,7,8),"Summer", "Fall")))
movies$season <- as.factor(movies$season)
table(movies$season)
```

```
##
##   Fall Spring Summer Winter
##   2114   1893   1874   1787
```

```
#7.director tier
summary(movies$score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.90    5.80    6.50    6.39    7.10    9.30       3
```

```
movies$mscore <- ave(movies$score, movies$director, FUN = mean)
movies$smscore <- scales::rescale(movies$mscore, to=c(0,1))
summary(movies$smscore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.6162  0.6913  0.6803  0.7559  1.0000      30
```

```r
movies$tier[movies$smscore>=0.8] <- "A"
movies$tier[movies$smscore>=0.6 & movies$smscore<0.8] <- "B"
movies$tier[movies$smscore<0.6] <- "C"
movies$tier <- as.factor(movies$tier)
table(movies$tier)
```

```
##
##    A    B    C
## 1028 5076 1534
```

```r
##################################################################

# single imputation

mi_data <- movies[movies$USA==1, ]
mi_data <- mi_data[!is.na(mi_data$year),]

mi_data <- subset(mi_data,select=-c(released, director, writer, star, company,
                              gross, month, season, tier, country, USA, mscore, smscore))

sapply(mi_data, function(x) sum(is.na(x)))
```

```
##    name  rating   genre    year   score   votes  budget runtime
##       0       0       0       0       0       0    1092       1
```

```r
dim(mi_data)
```

```
## [1] 5456    8
```

```r
sum(is.na(mi_data))
```

```
## [1] 1093
```

```r
sapply(mi_data, class)
```

```
##        name      rating       genre        year       score       votes
## "character"    "factor"    "factor"   "integer"   "numeric"   "numeric"
##      budget     runtime
##   "numeric"   "numeric"
```

```r
mi_data$budget[mi_data$budget>50000000 & mi_data$budget<=356000000] <- 0
sapply(mi_data, function(x) sum(is.na(x)))
```

```
##    name  rating   genre    year   score   votes  budget runtime
##       0       0       0       0       0       0    1092       1
```

```r
mi_data_screen <- mi_data[(mi_data$budget!=0 | is.na(mi_data$budget)), ]

mi_data_screen <-mi_data_screen[complete.cases(mi_data_screen$runtime),]

sapply(mi_data_screen, function(x) sum(is.na(x)))
```

```
##    name  rating   genre    year   score   votes  budget runtime
##       0       0       0       0       0       0    1092       0
```

```r
mi <- lm(budget~rating+genre+year+score+votes+runtime, data=mi_data_screen)
library(MASS)
par(mfrow=c(1,1))
boxcox(mi)
```

```r
par(mfrow=c(2,2))
plot(mi)
```

```r
mi_data_screen$sqrt_budget <- sqrt(mi_data_screen$budget)
mi_data_screen$log_votes <- log(mi_data_screen$votes)
mi_data_screen$log_runtime <- log(mi_data_screen$runtime)

mi2 <- lm(sqrt_budget~rating+genre+year+score+log_votes+log_runtime, data=mi_data_screen)
par(mfrow=c(1,1))
boxcox(mi2)
```

```r
par(mfrow=c(2,2))
plot(mi2)

library(mice)
```

```r
mice_data <- subset(mi_data_screen, select=c(name,sqrt_budget,rating,genre,year,score,log_votes,log_run
sapply(mice_data, function(x) sum(is.na(x)))
```

```
##        name sqrt_budget      rating       genre        year       score
##           0        1092           0           0           0           0
##   log_votes log_runtime
##           0           0
```

```r
imputed_data <- mice(mice_data[-1], m = 5, seed=2021)
```

```
##
##  iter imp variable
##   1   1  sqrt_budget
##   1   2  sqrt_budget
##   1   3  sqrt_budget
##   1   4  sqrt_budget
##   1   5  sqrt_budget
##   2   1  sqrt_budget
##   2   2  sqrt_budget
##   2   3  sqrt_budget
##   2   4  sqrt_budget
##   2   5  sqrt_budget
##   3   1  sqrt_budget
##   3   2  sqrt_budget
##   3   3  sqrt_budget
##   3   4  sqrt_budget
##   3   5  sqrt_budget
##   4   1  sqrt_budget
##   4   2  sqrt_budget
##   4   3  sqrt_budget
##   4   4  sqrt_budget
##   4   5  sqrt_budget
##   5   1  sqrt_budget
##   5   2  sqrt_budget
##   5   3  sqrt_budget
##   5   4  sqrt_budget
##   5   5  sqrt_budget
```

```r
#imputed_data$imp$sqrt_budget
#imp_tot <- complete(imputed_data, "broad", inc = TRUE)
#imp_tot <- subset(imp_tot, select=c(sqrt_budget.0, sqrt_budget.1, sqrt_budget.2, sqrt_budget.3, sqrt_b
#imp_tot$sqrt_budget <-  apply(imp_tot[-1], 1, mean)
imp_tot <- complete(imputed_data, 2)
mi_data_screen$imp_budget <- imp_tot$sqrt_budget^2

imp_movies <- subset(mi_data_screen, select = c(name,imp_budget,year))


samples <- movies[movies$USA==1, ]
samples <- samples[!is.na(samples$year),]

data <- merge(samples, imp_movies, by = c("name","year"), all.x = TRUE)

data$budget[is.na(data$budget)] <- data$imp_budget[is.na(data$budget)]

data <- subset(data,select=c(gross,budget,runtime,year,rating,genre,tier,season))

dt.split <- data[complete.cases(data),]

dim(dt.split)
```

```
## [1] 5345    8
```

```r
#################################################################

# data splitting

set.seed(2021)

n <- nrow(dt.split)
ids = sample(1:n, size=n/2, replace=FALSE)

train = dt.split[ids,]
valid = dt.split[-ids,]

dim(data)
```

```
## [1] 5456    8
```

```r
dim(valid)
```

```
## [1] 2673    8
```

Exploratory Data Analysis

```r
par(mfrow = c(2, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
hist(train$gross, main=NULL, xlab="gross")
hist(train$budget, main=NULL, xlab="budget")
mtext("Figure 1: Histograms of Quantitative Variables", side = 3, font=2, line=-1, outer=TRUE)
hist(train$runtime, main=NULL, xlab="runtime")
hist(train$year, main=NULL, xlab="year")
```

```r
panel.cor <- function(x, y) {
    # usr <- par('usr') on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- round(cor(x, y, use = "complete.obs"), 2)
    txt <- paste0("R = ", r)
    cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}
pairs(~ + gross + budget + runtime + year, data = train, lower.panel = panel.cor, main="Figure 2: Scatte
```

```r
par(mfrow = c(1, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
pct <- round(100*prop.table(table(train$rating)))
lab <- paste(pct)
lab <- paste(lab,'%',sep='')
pie(table(train$rating), labels=lab, col=rainbow(9))
title("Rating", line=-0.9, cex.main=0.9)
legend(-0.7, -1.1, c('G','PG'), cex = 0.7, fill = rainbow(9)[1:2], horiz = TRUE, inset = c(0, -0.1), xpd
legend(-0.7, -1.3, c('PG-13','NC-17/R'), cex = 0.7, fill = rainbow(9)[3:4], horiz = TRUE, inset = c(0, -

pct <- round(100*prop.table(table(train$genre)))
lab <- paste(pct)
lab <- paste(lab,'%',sep='')
pie(table(train$genre), labels=lab, col=rainbow(9))
title("Genre", line=-0.9, cex.main=0.9)
legend(-1.3, -1,   c('Action','Adventure','Animation'), cex = 0.7,
       fill = rainbow(9)[1:3], horiz = TRUE, inset = c(0, -0.1), xpd = TRUE, bty = "n")
legend(-1.3, -1.2, c('Biography','Comedy   ','Crime'), cex = 0.7,
       fill = rainbow(9)[4:6], horiz = TRUE, inset = c(0, -0.1), xpd = TRUE, bty = "n")
legend(-1.3, -1.4, c('Drama','Horror','Others'), cex = 0.7,
       fill = rainbow(9)[7:9], horiz = TRUE, inset = c(0, -0.1), xpd = TRUE, bty = "n")
mtext("Figure 3a: Pie Charts of Qualitative Variables", side = 3, font=2, line=-1, outer=TRUE)
```

```r
par(mfrow = c(1, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
pct <- round(100*prop.table(table(train$tier)))
lab <- paste(pct)
lab <- paste(lab,'%',sep='')
pie(table(train$tier), labels=lab, col=rainbow(9))
title("Director Tier", line=-1, cex.main=0.9)
legend(-0.6, -1.2, c('A','B','C'), cex = 0.7, fill = rainbow(9), horiz = TRUE, inset = c(0, -0.1), xpd =

pct <- round(100*prop.table(table(train$season)))
lab <- paste(pct)
lab <- paste(lab,'%',sep='')
pie(table(train$season), labels=lab, col=rainbow(9))
title("Season", line=-1, cex.main=0.9)
legend(-1.6, -1.2, c('Fall','Spring','Summer','Winter'), cex = 0.7, fill = rainbow(9), horiz = TRUE, ins
mtext("Figure 3b: Pie Charts of Qualitative Variables", side = 3, font=2, line=-1, outer=TRUE)
```

```r
par(mfrow = c(1, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
boxplot(train$gross~train$rating, xlab=NULL, ylab='gross',col=rainbow(9), las = 2, cex.axis=0.8)
title("Rating", line=0.2, cex.main=0.9)
boxplot(train$gross~train$genre, xlab=NULL, ylab='gross',col=rainbow(9), las = 2, cex.axis=0.8)
title("Genre", line=0.2, cex.main=0.9, xpd = FALSE)
mtext("Figure 4a: Side-by-Side Box Plots", side = 3, font=2, line=-1, outer=TRUE)
```

```r
boxplot(train$gross~train$tier, xlab=NULL,ylab='gross',col=rainbow(9), las = 0, cex.axis=0.8)
title("Director Tier", line=0.2, cex.main=0.9)
boxplot(train$gross~train$season, xlab=NULL,ylab='gross',col=rainbow(9), las = 2, cex.axis=0.8)
title("Season", line=0.2, cex.main=0.9)
mtext("Figure 4b: Side-by-Side Box Plots", side = 3, font=2, line=-1, outer=TRUE)
```

Preliminary Fit

```r
# model 1
model1 <- lm(gross~.,data=train)
summary(model1) #R-squared: 0.5833
```

```
##
## Call:
## lm(formula = gross ~ ., data = train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -517019649  -40497234   -2553287   25724957 2063903139
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -6.131e+08  4.618e+08  -1.328 0.184373
## budget           2.967e+00  7.828e-02  37.906  < 2e-16 ***
## runtime          8.324e+05  1.695e+05   4.909 9.68e-07 ***
## year             2.773e+05  2.300e+05   1.206 0.227983
## ratingPG         3.469e+05  1.440e+07   0.024 0.980778
## ratingPG-13     -6.180e+06  1.479e+07  -0.418 0.676153
## ratingR/NC-17   -1.216e+07  1.443e+07  -0.843 0.399311
## genreAdventure -4.125e+06  1.134e+07  -0.364 0.716140
## genreAnimation  9.925e+07  1.419e+07   6.993 3.39e-12 ***
## genreBiography -2.290e+07  1.237e+07  -1.851 0.064270 .
## genreComedy      4.901e+06  6.480e+06   0.756 0.449520
## genreCrime      -1.437e+07  1.025e+07  -1.402 0.160892
## genreDrama      -2.684e+06  7.797e+06  -0.344 0.730687
## genreHorror      4.433e+07  1.230e+07   3.604 0.000319 ***
## genreOthers      2.272e+07  1.923e+07   1.181 0.237636
## tierB           -4.113e+07  8.837e+06  -4.654 3.42e-06 ***
## tierC           -5.641e+07  1.019e+07  -5.538 3.37e-08 ***
## seasonSpring     1.077e+07  6.317e+06   1.705 0.088351 .
## seasonSummer     1.774e+07  6.298e+06   2.817 0.004887 **
## seasonWinter     7.387e+06  6.516e+06   1.134 0.257027
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115600000 on 2652 degrees of freedom
## Multiple R-squared:  0.5833, Adjusted R-squared:  0.5803
## F-statistic: 195.3 on 19 and 2652 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(model1, sub.caption = "")
```

```r
library(MASS)
par(mfrow=c(1,1))
```

```r
boxcox(model1) #Lambda is around 0, so we need to take log transformation on Y.
```

```r
# model 2
train$log_gross <- log(train$gross)
train$log_budget <- log(train$budget)
train$log_runtime <- log(train$runtime)

hist(train$log_gross)
```

```r
model2 <- lm(log_gross~log_budget+log_runtime+year+rating+genre+tier+season,data=train)
summary(model2) #R-squared: 0.4683
```

```
##
## Call:
## lm(formula = log_gross ~ log_budget + log_runtime + year + rating +
##     genre + tier + season, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9900 -0.7032  0.2026  0.9423  7.2120
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -25.313860   6.089790  -4.157 3.33e-05 ***
## log_budget       0.745561   0.028452  26.205  < 2e-16 ***
## log_runtime      2.144239   0.254175   8.436  < 2e-16 ***
## year             0.009958   0.003007   3.311 0.000940 ***
## ratingPG         0.458742   0.192866   2.379 0.017451 *
## ratingPG-13      0.446976   0.198442   2.252 0.024376 *
## ratingR/NC-17    0.052990   0.191939   0.276 0.782509
## genreAdventure  -0.036945   0.149471  -0.247 0.804792
## genreAnimation   1.224642   0.188213   6.507 9.15e-11 ***
## genreBiography  -0.615193   0.160649  -3.829 0.000131 ***
## genreComedy     -0.103712   0.083740  -1.238 0.215642
## genreCrime      -0.467234   0.133530  -3.499 0.000475 ***
## genreDrama      -0.605070   0.101178  -5.980 2.53e-09 ***
## genreHorror      0.741838   0.163114   4.548 5.66e-06 ***
## genreOthers      0.026839   0.254104   0.106 0.915891
## tierB           -0.319131   0.116720  -2.734 0.006296 **
## tierC           -0.692771   0.134768  -5.140 2.94e-07 ***
## seasonSpring     0.071238   0.083584   0.852 0.394129
## seasonSummer     0.390091   0.083341   4.681 3.00e-06 ***
## seasonWinter     0.258869   0.086297   3.000 0.002727 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.531 on 2652 degrees of freedom
## Multiple R-squared:  0.4686, Adjusted R-squared:  0.4648
## F-statistic: 123.1 on 19 and 2652 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(model2, sub.caption = "")
```

```r
par(mfrow=c(1,1))
boxcox(model2)
```

```r
# model 3
train$log_gross_2 <- train$log_gross^2

hist(train$log_gross_2)
```

```r
model3 <- lm(log_gross_2~log_budget+log_runtime+year+rating+genre+tier+season,data=train)
summary(model3) #R-squared: 0.499
```

```
##
## Call:
## lm(formula = log_gross_2 ~ log_budget + log_runtime + year +
##     rating + genre + tier + season, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -196.697  -25.487    4.764   30.732  235.215
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.354e+03  1.900e+02  -7.124 1.35e-12 ***
## log_budget      2.382e+01  8.877e-01  26.831  < 2e-16 ***
## log_runtime     7.475e+01  7.930e+00   9.426  < 2e-16 ***
## year            4.537e-01  9.382e-02   4.836 1.40e-06 ***
## ratingPG        1.202e+01  6.017e+00   1.997  0.04593 *
## ratingPG-13     1.129e+01  6.191e+00   1.824  0.06821 .
## ratingR/NC-17  -2.175e+00  5.988e+00  -0.363  0.71648
## genreAdventure -2.885e+00  4.663e+00  -0.619  0.53625
## genreAnimation  4.220e+01  5.872e+00   7.187 8.56e-13 ***
## genreBiography -2.333e+01  5.012e+00  -4.655 3.40e-06 ***
## genreComedy    -4.900e+00  2.613e+00  -1.875  0.06084 .
## genreCrime     -1.648e+01  4.166e+00  -3.955 7.85e-05 ***
## genreDrama     -2.121e+01  3.157e+00  -6.719 2.23e-11 ***
## genreHorror     2.353e+01  5.089e+00   4.624 3.94e-06 ***
## genreOthers     3.830e-01  7.928e+00   0.048  0.96147
## tierB          -1.164e+01  3.642e+00  -3.197  0.00141 **
## tierC          -2.385e+01  4.205e+00  -5.672 1.56e-08 ***
## seasonSpring    3.286e+00  2.608e+00   1.260  0.20777
## seasonSummer    1.308e+01  2.600e+00   5.029 5.25e-07 ***
## seasonWinter    8.034e+00  2.692e+00   2.984  0.00287 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.78 on 2652 degrees of freedom
## Multiple R-squared:  0.499,  Adjusted R-squared:  0.4954
## F-statistic:   139 on 19 and 2652 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(model3, sub.caption = "")
```

```r
par(mfrow=c(1,1))
boxcox(model3)
```

```r
#model 4
train$log_gross_4 <- train$log_gross^4
```

```r
hist(train$log_gross_4)
```

```r
model4 <- lm(log_gross_4~log_budget+log_runtime+year+rating+genre+tier+season,data=train)
summary(model4) #R-squared: 0.499
```

```
##
## Call:
## lm(formula = log_gross_4 ~ log_budget + log_runtime + year +
##     rating + genre + tier + season, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87655 -16668    536  17606 131510
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.072e+06  1.027e+05 -10.432  < 2e-16 ***
## log_budget      1.280e+04  4.799e+02  26.675  < 2e-16 ***
## log_runtime     4.692e+04  4.287e+03  10.944  < 2e-16 ***
## year            3.693e+02  5.072e+01   7.282 4.32e-13 ***
## ratingPG        4.307e+03  3.253e+03   1.324  0.18565
## ratingPG-13     3.622e+03  3.347e+03   1.082  0.27921
## ratingR/NC-17  -4.584e+03  3.237e+03  -1.416  0.15686
## genreAdventure -3.399e+03  2.521e+03  -1.348  0.17771
## genreAnimation  2.639e+04  3.174e+03   8.312  < 2e-16 ***
## genreBiography -1.647e+04  2.710e+03  -6.078 1.39e-09 ***
## genreComedy    -4.533e+03  1.412e+03  -3.209  0.00135 **
## genreCrime     -1.083e+04  2.252e+03  -4.810 1.59e-06 ***
## genreDrama     -1.340e+04  1.706e+03  -7.852 5.90e-15 ***
## genreHorror     1.180e+04  2.751e+03   4.291 1.84e-05 ***
## genreOthers    -5.878e+02  4.286e+03  -0.137  0.89092
## tierB          -8.003e+03  1.969e+03  -4.065 4.94e-05 ***
## tierC          -1.502e+04  2.273e+03  -6.608 4.71e-11 ***
## seasonSpring    2.599e+03  1.410e+03   1.843  0.06539 .
## seasonSummer    7.595e+03  1.406e+03   5.403 7.13e-08 ***
## seasonWinter    3.927e+03  1.456e+03   2.698  0.00702 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25830 on 2652 degrees of freedom
## Multiple R-squared:  0.5331, Adjusted R-squared:  0.5297
## F-statistic: 159.4 on 19 and 2652 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(model4, sub.caption = "")
```

```r
par(mfrow=c(1,1))
boxcox(model4)
```

```r
#summary
par(mfrow = c(2, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
hist(train$gross, main=NULL, xlab = "gross")
boxcox(model1)
plot(model1,1, sub.caption = expression(paste("Figure 5a: Preliminary Regression of ", gross)))
plot(model1,2, sub.caption = "")
```

```r
par(mfrow = c(2, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
hist(train$log_gross, main=NULL, xlab = "log(gross)")
boxcox(model2)
plot(model2,1, sub.caption = expression(paste("Figure 5b: Preliminary Regression of ", log(gross))))
plot(model2,2, sub.caption = "")
```

```r
par(mfrow = c(2, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
hist(train$log_gross_2, main=NULL, xlab = "log(gross)^2")
boxcox(model3)
plot(model3,1, sub.caption = expression(paste("Figure 5c: Preliminary Regression of ", log(gross)^2)))
plot(model3,2, sub.caption = "")
```

```r
par(mfrow = c(2, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
hist(train$log_gross_4, main=NULL, xlab = "log(gross)^4")
boxcox(model4)
plot(model4,1, sub.caption = expression(paste("Figure 5d: Preliminary Regression of ", log(gross)^4)))
plot(model4,2, sub.caption = "")
```

```r
#pairs(~ + log_gross + log_budget + log_runtime + year, data = train, lower.panel = panel.cor, main=exp

#pairs(~ + log_gross_2 + log_budget + log_runtime + year, data = train, lower.panel = panel.cor, main=e

pairs(~ + log_gross_4 + log_budget + log_runtime + year, data = train, lower.panel = panel.cor, main=exp
```

```r
#interaction discussion
par(mfrow = c(2, 3),oma=c(2,2,2,2),mar=c(4,3,3,3))
plot(train$log_budget,model4$residuals, xlab="log(budget)")
abline(h=0, col='red')
plot(train$log_runtime,model4$residuals, xlab="log(runtim)")
abline(h=0, col='red')
plot(train$year,model4$residuals, xlab="year")
abline(h=0, col='red')
plot(train$log_budget*train$log_runtime,model4$residuals, xlab="log(budget)*log(runtime)")
abline(h=0, col='red')
plot(train$log_budget*train$year,model4$residuals, xlab="log(budget)*year")
abline(h=0, col='red')
plot(train$log_runtime*train$year,model4$residuals, xlab="log(runtime)*year")
abline(h=0, col='red')
mtext(expression(paste("Figure 7: Model on ", log(gross)^4,": Residuals vs. Interaction Terms")), side =
```

Model Selection

```r
train <- subset(train,select=c(log_gross_4, log_budget, log_runtime, year, rating, genre, tier, season)

model_0 = lm(log_gross_4~1, data=train) #only intercept
model_F = lm(log_gross_4~., data=train) #first-order models
model_F2 = lm(log_gross_4~.^2, data=train) #interaction models

#forwrd stepwise procedure
length(model_F$coefficients)
```

```
## [1] 20
```

```r
length(model_F2$coefficients)
```

```
## [1] 156
```

```r
library(MASS)

sel1 = stepAIC(model_0, scope=list(lower=model_0, upper=model_F), direction="both", k=2, trace=0) #AIC
sel2 = stepAIC(model_0, scope=list(lower=model_0, upper=model_F), direction="both", k=log(n), trace=0)

sel3 = stepAIC(model_0, scope=list(lower=model_0, upper=model_F2), direction="both", k=2, trace=0) #AIC
sel4 = stepAIC(model_0, scope=list(lower=model_0, upper=model_F2), direction="both", k=log(n), trace=0)

sel1$call; sel2$call; sel3$call; sel4$call
```

```
## lm(formula = log_gross_4 ~ log_budget + genre + log_runtime +
##     year + rating + tier + season, data = train)

## lm(formula = log_gross_4 ~ log_budget + tier + genre + log_runtime +
##     year + rating + season, data = train)

## lm(formula = log_gross_4 ~ log_budget + genre + log_runtime +
##     tier + rating + year + season + log_budget:genre + log_budget:log_runtime +
##     log_budget:year + genre:year + log_budget:rating + log_runtime:year +
##     rating:year + genre:log_runtime + log_runtime:season + year:season +
##     log_budget:tier + tier:rating, data = train)

## lm(formula = log_gross_4 ~ log_budget + tier + genre + log_runtime +
##     year + rating + season + log_budget:log_runtime + log_budget:year +
##     log_runtime:year, data = train)
```

```r
# sel1 and sel2 are identical

sel1$anova; sel3$anova; sel4$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## log_gross_4 ~ 1
##
## Final Model:
## log_gross_4 ~ log_budget + genre + log_runtime + year + rating +
##     tier + season
##
##
##            Step Df     Deviance Resid. Df   Resid. Dev      AIC
## 1                                    2671 3.788908e+12 56307.76
## 2  + log_budget  1 1.653074e+12      2670 2.135834e+12 54778.12
## 3       + genre  8 1.101726e+11      2662 2.025662e+12 54652.61
## 4 + log_runtime  1 1.185188e+11      2661 1.907143e+12 54493.51
## 5        + year  1 4.589669e+10      2660 1.861246e+12 54430.42
## 6      + rating  3 3.645073e+10      2657 1.824795e+12 54383.57
## 7        + tier  2 3.531243e+10      2655 1.789483e+12 54335.36
## 8      + season  3 2.033636e+10      2652 1.769147e+12 54310.82

## Stepwise Model Path
## Analysis of Deviance Table
```

```
##
## Initial Model:
## log_gross_4 ~ 1
##
## Final Model:
## log_gross_4 ~ log_budget + genre + log_runtime + tier + rating +
##     year + season + log_budget:genre + log_budget:log_runtime +
##     log_budget:year + genre:year + log_budget:rating + log_runtime:year +
##     rating:year + genre:log_runtime + log_runtime:season + year:season +
##     log_budget:tier + tier:rating
##
##
##                         Step Df     Deviance Resid. Df   Resid. Dev      AIC
## 1                                                 2671 3.788908e+12 56307.76
## 2               + log_budget  1 1.653074e+12      2670 2.135834e+12 54778.12
## 3                    + genre  8 1.101726e+11      2662 2.025662e+12 54652.61
## 4              + log_runtime  1 1.185188e+11      2661 1.907143e+12 54493.51
## 5         + log_budget:genre  8 8.006003e+10      2653 1.827083e+12 54394.92
## 6   + log_budget:log_runtime  1 4.024945e+10      2652 1.786833e+12 54337.40
## 7                     + tier  2 2.894769e+10      2650 1.757886e+12 54297.76
## 8                   + rating  3 2.635329e+10      2647 1.731532e+12 54263.40
## 9                     + year  1 1.985428e+10      2646 1.711678e+12 54234.58
## 10         + log_budget:year  1 5.672015e+10      2645 1.654958e+12 54146.54
## 11                  + season  3 2.140430e+10      2642 1.633554e+12 54117.76
## 12             + year:genre  8 1.787773e+10      2634 1.615676e+12 54104.35
## 13       + log_budget:rating  3 1.103331e+10      2631 1.604642e+12 54092.04
## 14        + log_runtime:year  1 6.647675e+09      2630 1.597995e+12 54082.95
## 15            + rating:genre 22 2.964114e+10      2608 1.568354e+12 54076.92
## 16             + year:rating  3 5.025142e+09      2605 1.563329e+12 54074.35
## 17       + log_runtime:genre  8 1.070051e+10      2597 1.552628e+12 54072.00
## 18      + log_runtime:season  3 6.428350e+09      2594 1.546200e+12 54066.91
## 19             + year:season  3 4.369502e+09      2591 1.541830e+12 54065.35
## 20         + log_budget:tier  2 3.248526e+09      2589 1.538582e+12 54063.71
## 21            - genre:rating 22 2.515573e+10      2611 1.563737e+12 54063.05
## 22             + rating:tier  6 7.809289e+09      2605 1.555928e+12 54061.67
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## log_gross_4 ~ 1
##
## Final Model:
## log_gross_4 ~ log_budget + tier + genre + log_runtime + year +
##     rating + season + log_budget:log_runtime + log_budget:year +
##     log_runtime:year
##
##
##                         Step Df     Deviance Resid. Df   Resid. Dev      AIC
## 1                                                 2671 3.788908e+12 56314.35
## 2               + log_budget  1 1.653074e+12      2670 2.135834e+12 54791.29
## 3                     + tier  2 8.384097e+10      2668 2.051993e+12 54701.45
## 4                    + genre  8 1.234406e+11      2660 1.928553e+12 54604.35
## 5              + log_runtime  1 6.569999e+10      2659 1.862853e+12 54520.32
```

```
## 6  + log_budget:log_runtime  1 5.891365e+10      2658 1.803939e+12 54443.03
## 7                    + year  1 2.505423e+10      2657 1.778885e+12 54414.25
## 8         + log_budget:year  1 7.011358e+10      2656 1.708771e+12 54315.38
## 9                  + rating  3 2.806517e+10      2653 1.680706e+12 54296.89
## 10                 + season  3 2.308749e+10      2650 1.657618e+12 54285.68
## 11       + log_runtime:year  1 8.646139e+09      2649 1.648972e+12 54280.29
```

```
step.f = sel1
step.f2 = sel3
step.f3 = sel4

# Therefore, there is 3 candidate models.
```

```
#best subset selection procesure

#library(leaps)

#sub_set <- regsubsets(log_gross_4~.^2,data=train,nbest=1,nvmax=15,method="exhaustive", really.big=T)

#sum_sub <- summary(sub_set)

#n <- nrow(train)
#p.m <- as.integer(as.numeric(rownames(sum_sub$which))+1)

#sse=sum_sub$rss
#aic=n*log(sse/n)+2*p.m
#bic=n*log(sse/n)+log(n)*p.m

#res_sub <- cbind((sum_sub$which+0), sse, sum_sub$rsq, sum_sub$adjr2, sum_sub$cp, bic, aic)

#sse0 <- sum(model_0$residuals^2)
#p0 <- 1
#c0 <- sse0/(summary(model_F2)$sigma^2)-(n-2*p0)
#aic0=n*log(sse0/n)+2*p0
#bic0=n*log(sse0/n)+log(n)*p0

#none=c(1, rep(0,20), sse0, 0, 0, c0, bic0, aic0)

#res_sub <- rbind(none, res_sub)
#colnames(res_sub) <- c(colnames(sum_sub$which), "sse", "R^2", "R^2_a", "Cp", "bic", "aic")
#round(res_sub,5)
```

Model Validation

Internal Validation

```
fit3 <- lm(log_gross_4~.^2, data=train)

mse3<-anova(fit3)["Residuals",3]
mse3 #593462127
```

```
## [1] 593462127
```

```
# Candidate Model 1
sse.fs1<-anova(step.f)["Residuals",2]
sse.fs1 #1.769147e+12
```

```
## [1] 1.769147e+12
```

```r
mse.fs1<-anova(step.f)["Residuals",3]
mse.fs1 #667098982
```

```
## [1] 667098982
```

```r
p.fs1<-length(step.f$coefficients)
p.fs1 #20
```

```
## [1] 20
```

```r
##C_p
cp.fs1<-sse.fs1/mse3-(n-2*p.fs1)
cp.fs1 #348.0605
```

```
## [1] -2323.94
```

```r
##Press_p
press.fs1<-sum(step.f$residuals^2/(1-influence(step.f)$hat)^2)
press.fs1 #1.797707e+12
```

```
## [1] 1.797707e+12
```

```r
## Candidate Model 2
sse.fs2<-anova(step.f2)["Residuals",2]
sse.fs2 #1.555928e+12
```

```
## [1] 1.555928e+12
```

```r
mse.fs2<-anova(step.f2)["Residuals",3]
mse.fs2 #597285255
```

```
## [1] 597285255
```

```r
p.fs2<-length(step.f2$coefficients)
p.fs2 #67
```

```
## [1] 67
```

```r
##C_p
cp.fs2<-sse.fs2/mse3-(n-2*p.fs2)
cp.fs2 #82.78
```

```
## [1] -2589.218
```

```r
##Press_p
press.fs2<-sum(step.f2$residuals^2/(1-influence(step.f2)$hat)^2)
press.fs2 #1.653771e+12
```

```
## [1] 1.653771e+12
```

```r
# Candidate Model 3
sse.fs3<-anova(step.f3)["Residuals",2]
sse.fs3 #1.648972e+12
```

```
## [1] 1.648972e+12
```

```r
mse.fs3<-anova(step.f3)["Residuals",3]
mse.fs3 #622488616
```

```
## [1] 622488616
```

```r
p.fs3<-length(step.f3$coefficients)
p.fs3 #23
```

```
## [1] 23
```

```r
##C_p
cp.fs3<-sse.fs3/mse3-(n-2*p.fs3)
cp.fs3 #151.5637
```

```
## [1] -2520.436
```

```r
##Press_p
press.fs3<-sum(step.f3$residuals^2/(1-influence(step.f3)$hat)^2)
press.fs3 #1.680874e+12
```

```
## [1] 1.680874e+12
```

External Validation

```r
valid$log_gross_4 <- log(valid$gross)^4
valid$log_budget <- log(valid$budget)
valid$log_runtime <- log(valid$runtime)

valid <- subset(valid,select=c(log_gross_4, log_budget, log_runtime, year, rating, genre, tier, season))

n <- nrow(valid)

# Candidate Model 1
fit.fs1.v<-lm(step.f, data=valid)
summary(step.f)
```

```
##
## Call:
## lm(formula = log_gross_4 ~ log_budget + genre + log_runtime +
##     year + rating + tier + season, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87655 -16668    536  17606 131510
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.072e+06  1.027e+05 -10.432  < 2e-16 ***
## log_budget      1.280e+04  4.799e+02  26.675  < 2e-16 ***
## genreAdventure -3.399e+03  2.521e+03  -1.348  0.17771
## genreAnimation  2.639e+04  3.174e+03   8.312  < 2e-16 ***
## genreBiography -1.647e+04  2.710e+03  -6.078 1.39e-09 ***
## genreComedy    -4.533e+03  1.412e+03  -3.209  0.00135 **
## genreCrime     -1.083e+04  2.252e+03  -4.810 1.59e-06 ***
## genreDrama     -1.340e+04  1.706e+03  -7.852 5.90e-15 ***
## genreHorror     1.180e+04  2.751e+03   4.291 1.84e-05 ***
## genreOthers    -5.878e+02  4.286e+03  -0.137  0.89092
## log_runtime     4.692e+04  4.287e+03  10.944  < 2e-16 ***
## year            3.693e+02  5.072e+01   7.282 4.32e-13 ***
## ratingPG        4.307e+03  3.253e+03   1.324  0.18565
## ratingPG-13     3.622e+03  3.347e+03   1.082  0.27921
```

```
## ratingR/NC-17  -4.584e+03  3.237e+03  -1.416  0.15686    
## tierB          -8.003e+03  1.969e+03  -4.065 4.94e-05 ***
## tierC          -1.502e+04  2.273e+03  -6.608 4.71e-11 ***
## seasonSpring    2.599e+03  1.410e+03   1.843  0.06539 .  
## seasonSummer    7.595e+03  1.406e+03   5.403 7.13e-08 ***
## seasonWinter    3.927e+03  1.456e+03   2.698  0.00702 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 25830 on 2652 degrees of freedom
## Multiple R-squared:  0.5331, Adjusted R-squared:  0.5297 
## F-statistic: 159.4 on 19 and 2652 DF,  p-value: < 2.2e-16
```

```r
summary(fit.fs1.v)
```

```
## 
## Call:
## lm(formula = step.f, data = valid)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -106348  -16177     983   16416   88096 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    -1.096e+06  9.910e+04 -11.060  < 2e-16 ***
## log_budget      1.258e+04  4.778e+02  26.333  < 2e-16 ***
## genreAdventure -6.430e+03  2.470e+03  -2.603  0.00929 ** 
## genreAnimation  1.850e+04  3.087e+03   5.994 2.33e-09 ***
## genreBiography -1.896e+04  2.537e+03  -7.474 1.05e-13 ***
## genreComedy    -7.461e+03  1.414e+03  -5.276 1.43e-07 ***
## genreCrime     -1.357e+04  2.165e+03  -6.267 4.28e-10 ***
## genreDrama     -1.374e+04  1.686e+03  -8.153 5.42e-16 ***
## genreHorror     1.151e+04  2.655e+03   4.335 1.51e-05 ***
## genreOthers     1.459e+03  4.320e+03   0.338  0.73555    
## log_runtime     4.042e+04  4.396e+03   9.194  < 2e-16 ***
## year            3.956e+02  4.904e+01   8.067 1.08e-15 ***
## ratingPG        1.512e+04  3.113e+03   4.857 1.26e-06 ***
## ratingPG-13     1.537e+04  3.148e+03   4.883 1.11e-06 ***
## ratingR/NC-17   5.218e+03  3.052e+03   1.710  0.08740 .  
## tierB          -1.382e+04  1.896e+03  -7.287 4.17e-13 ***
## tierC          -1.844e+04  2.219e+03  -8.310  < 2e-16 ***
## seasonSpring    4.393e+03  1.383e+03   3.178  0.00150 ** 
## seasonSummer    8.696e+03  1.375e+03   6.326 2.94e-10 ***
## seasonWinter    6.354e+03  1.394e+03   4.557 5.42e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 25310 on 2653 degrees of freedom
## Multiple R-squared:  0.5462, Adjusted R-squared:  0.5429 
## F-statistic:   168 on 19 and 2653 DF,  p-value: < 2.2e-16
```

```r
##percent change in parameter estimation
round(abs(coef(step.f)-coef(fit.fs1.v))/abs(coef(step.f))*100,3)
```

```
##    (Intercept)     log_budget genreAdventure genreAnimation genreBiography
##          2.288          1.708         89.194         29.879         15.133
##    genreComedy     genreCrime     genreDrama    genreHorror    genreOthers
##         64.613         25.263          2.560          2.520        348.258
##    log_runtime           year       ratingPG    ratingPG-13  ratingR/NC-17
##         13.847          7.107        251.142        324.373        213.828
##          tierB          tierC   seasonSpring   seasonSummer   seasonWinter
##         72.679         22.807         69.062         14.494         61.797
```

```
##percent change in standard errors
sd.fs1<- summary(step.f)$coefficients[,"Std. Error"]
sd.fs1.v<- summary(fit.fs1.v)$coefficients[,"Std. Error"]
round(abs(sd.fs1-sd.fs1.v)/sd.fs1*100,3)
```

```
##    (Intercept)     log_budget genreAdventure genreAnimation genreBiography
##          3.514          0.432          2.018          2.758          6.368
##    genreComedy     genreCrime     genreDrama    genreHorror    genreOthers
##          0.127          3.856          1.226          3.508          0.801
##    log_runtime           year       ratingPG    ratingPG-13  ratingR/NC-17
##          2.552          3.318          4.294          5.936          5.731
##          tierB          tierC   seasonSpring   seasonSummer   seasonWinter
##          3.669          2.357          1.933          2.210          4.209
```

```
##mean squared prediction error
pred.fs1<-predict.lm(step.f,valid[,-1])  #valid[,-1]=dataset without log_gross_4
mspe.fs1<-mean((pred.fs1-valid[,1])^2) #valid[,1]=log_gross_4
mspe.fs1 #648278691
```

```
## [1] 648278691
```

```
press.fs1/n #672542932
```

```
## [1] 672542932
```

```
mse.fs1 #667098982
```

```
## [1] 667098982
```

```
# Candidate Model 2
fit.fs2.v<-lm(step.f2,data=valid)
summary(step.f2)
```

```
##
## Call:
## lm(formula = log_gross_4 ~ log_budget + genre + log_runtime +
##     tier + rating + year + season + log_budget:genre + log_budget:log_runtime +
##     log_budget:year + genre:year + log_budget:rating + log_runtime:year +
##     rating:year + genre:log_runtime + log_runtime:season + year:season +
##     log_budget:tier + tier:rating, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -85243 -15357    296  16160  85341
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.415e+06  3.437e+06   0.412 0.680561
## log_budget                -8.623e+05  8.882e+04  -9.708  < 2e-16 ***
```

```
## genreAdventure            -7.095e+05  4.875e+05  -1.455 0.145664
## genreAnimation            -9.907e+05  7.642e+05  -1.296 0.194956
## genreBiography            -9.277e+05  4.978e+05  -1.863 0.062509 .
## genreComedy               -4.395e+05  2.709e+05  -1.622 0.104844
## genreCrime                -6.533e+05  4.810e+05  -1.358 0.174523
## genreDrama                -9.045e+05  3.119e+05  -2.900 0.003759 **
## genreHorror               -2.505e+06  4.779e+05  -5.241 1.73e-07 ***
## genreOthers               -1.389e+06  8.171e+05  -1.700 0.089178 .
## log_runtime                2.635e+06  8.004e+05   3.293 0.001006 **
## tierB                     -5.225e+04  2.440e+04  -2.141 0.032376 *
## tierC                     -2.274e+04  2.887e+04  -0.788 0.430982
## ratingPG                  -1.097e+06  6.512e+05  -1.685 0.092078 .
## ratingPG-13               -1.475e+06  6.531e+05  -2.258 0.023998 *
## ratingR/NC-17             -7.166e+05  6.240e+05  -1.148 0.250924
## year                      -1.492e+02  1.742e+03  -0.086 0.931740
## seasonSpring              -3.255e+04  2.557e+05  -0.127 0.898690
## seasonSummer               4.557e+05  2.496e+05   1.826 0.067959 .
## seasonWinter              -1.012e+05  2.655e+05  -0.381 0.703170
## log_budget:genreAdventure  -4.951e+03  2.284e+03  -2.168 0.030258 *
## log_budget:genreAnimation  -4.264e+02  2.853e+03  -0.149 0.881234
## log_budget:genreBiography  -9.077e+03  2.683e+03  -3.383 0.000728 ***
## log_budget:genreComedy     -4.493e+03  1.312e+03  -3.426 0.000623 ***
## log_budget:genreCrime      -1.634e+03  2.398e+03  -0.681 0.495704
## log_budget:genreDrama      -7.987e+03  1.529e+03  -5.222 1.91e-07 ***
## log_budget:genreHorror     -1.347e+04  2.281e+03  -5.907 3.95e-09 ***
## log_budget:genreOthers     -8.297e+02  4.330e+03  -0.192 0.848067
## log_budget:log_runtime      1.765e+04  3.205e+03   5.508 3.98e-08 ***
## log_budget:year             3.993e+02  4.358e+01   9.163  < 2e-16 ***
## genreAdventure:year         4.104e+02  2.443e+02   1.680 0.093101 .
## genreAnimation:year         6.358e+02  4.043e+02   1.573 0.115891
## genreBiography:year         4.169e+02  2.438e+02   1.710 0.087427 .
## genreComedy:year            2.431e+02  1.370e+02   1.774 0.076112 .
## genreCrime:year             2.871e+02  2.415e+02   1.189 0.234732
## genreDrama:year             4.291e+02  1.569e+02   2.735 0.006272 **
## genreHorror:year            1.181e+03  2.342e+02   5.044 4.88e-07 ***
## genreOthers:year            7.343e+02  4.016e+02   1.829 0.067567 .
## log_budget:ratingPG         1.334e+03  1.829e+03   0.729 0.465784
## log_budget:ratingPG-13      1.715e+03  1.840e+03   0.932 0.351384
## log_budget:ratingR/NC-17   -1.338e+03  1.638e+03  -0.817 0.414227
## log_runtime:year           -1.458e+03  4.053e+02  -3.598 0.000326 ***
## ratingPG:year               5.264e+02  3.279e+02   1.606 0.108489
## ratingPG-13:year            7.117e+02  3.270e+02   2.177 0.029594 *
## ratingR/NC-17:year          3.553e+02  3.123e+02   1.138 0.255321
## genreAdventure:log_runtime -5.971e+03  1.600e+04  -0.373 0.709076
## genreAnimation:log_runtime -5.761e+04  2.941e+04  -1.959 0.050209 .
## genreBiography:log_runtime  5.009e+04  2.042e+04   2.453 0.014246 *
## genreComedy:log_runtime     6.354e+03  1.126e+04   0.564 0.572596
## genreCrime:log_runtime      2.191e+04  1.503e+04   1.458 0.144926
## genreDrama:log_runtime      3.689e+04  1.256e+04   2.937 0.003343 **
## genreHorror:log_runtime     8.131e+04  2.600e+04   3.127 0.001784 **
## genreOthers:log_runtime    -1.236e+04  3.522e+04  -0.351 0.725706
## log_runtime:seasonSpring    1.724e+04  9.854e+03   1.750 0.080247 .
## log_runtime:seasonSummer    2.019e+04  9.653e+03   2.091 0.036604 *
## log_runtime:seasonWinter   -7.874e+03  9.035e+03  -0.871 0.383588
```

```
## year:seasonSpring          -2.288e+01  1.293e+02  -0.177 0.859539
## year:seasonSummer          -2.711e+02  1.245e+02  -2.178 0.029516 *
## year:seasonWinter           7.065e+01  1.324e+02   0.533 0.593825
## log_budget:tierB            1.110e+03  1.365e+03   0.813 0.416157
## log_budget:tierC           -9.634e+02  1.651e+03  -0.584 0.559525
## tierB:ratingPG              2.983e+04  8.894e+03   3.354 0.000808 ***
## tierC:ratingPG              2.831e+04  1.023e+04   2.767 0.005701 **
## tierB:ratingPG-13           2.705e+04  8.461e+03   3.198 0.001402 **
## tierC:ratingPG-13           2.473e+04  9.955e+03   2.485 0.013031 *
## tierB:ratingR/NC-17         2.764e+04  8.246e+03   3.352 0.000814 ***
## tierC:ratingR/NC-17         2.560e+04  9.705e+03   2.638 0.008394 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24440 on 2605 degrees of freedom
## Multiple R-squared:  0.5893, Adjusted R-squared:  0.5789
## F-statistic: 56.64 on 66 and 2605 DF,  p-value: < 2.2e-16
```

```
summary(fit.fs2.v)
```

```
##
## Call:
## lm(formula = step.f2, data = valid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -105388  -14829     477   15230   86838
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -1.521e+06  3.303e+06  -0.460 0.645245
## log_budget                  -8.022e+05  8.462e+04  -9.480  < 2e-16 ***
## genreAdventure              -7.914e+05  4.539e+05  -1.743 0.081393 .
## genreAnimation              -2.018e+05  7.087e+05  -0.285 0.775842
## genreBiography              -3.314e+05  4.724e+05  -0.701 0.483097
## genreComedy                 -2.410e+05  2.640e+05  -0.913 0.361226
## genreCrime                  -1.284e+06  4.025e+05  -3.189 0.001445 **
## genreDrama                  -9.420e+05  3.099e+05  -3.040 0.002392 **
## genreHorror                 -2.416e+06  4.463e+05  -5.413 6.78e-08 ***
## genreOthers                 -4.428e+05  7.559e+05  -0.586 0.558099
## log_runtime                  3.285e+06  7.666e+05   4.286 1.89e-05 ***
## tierB                       -9.234e+03  2.087e+04  -0.442 0.658225
## tierC                       -1.441e+04  2.561e+04  -0.563 0.573639
## ratingPG                    -3.071e+06  5.713e+05  -5.376 8.30e-08 ***
## ratingPG-13                 -3.069e+06  5.579e+05  -5.501 4.15e-08 ***
## ratingR/NC-17               -2.428e+06  5.320e+05  -4.564 5.26e-06 ***
## year                         1.536e+03  1.675e+03   0.917 0.359095
## seasonSpring                 3.364e+05  2.400e+05   1.402 0.161139
## seasonSummer                 2.757e+05  2.368e+05   1.164 0.244368
## seasonWinter                 1.305e+05  2.459e+05   0.531 0.595774
## log_budget:genreAdventure    3.592e+03  2.690e+03   1.335 0.182019
## log_budget:genreAnimation    1.405e+04  3.060e+03   4.593 4.59e-06 ***
## log_budget:genreBiography   -1.989e+03  2.678e+03  -0.743 0.457763
## log_budget:genreComedy      -1.481e+03  1.349e+03  -1.098 0.272412
## log_budget:genreCrime       -5.045e+03  2.109e+03  -2.392 0.016819 *
```

```
## log_budget:genreDrama      -5.853e+03  1.450e+03  -4.037 5.58e-05 ***
## log_budget:genreHorror     -1.066e+04  2.058e+03  -5.182 2.37e-07 ***
## log_budget:genreOthers     -5.054e+03  3.068e+03  -1.647 0.099616 .
## log_budget:log_runtime      2.422e+04  2.946e+03   8.221 3.15e-16 ***
## log_budget:year             3.539e+02  4.151e+01   8.526  < 2e-16 ***
## genreAdventure:year         4.599e+02  2.305e+02   1.995 0.046178 *
## genreAnimation:year         1.489e+02  3.764e+02   0.396 0.692466
## genreBiography:year         2.780e+02  2.325e+02   1.196 0.231951
## genreComedy:year            1.306e+02  1.333e+02   0.980 0.327247
## genreCrime:year             6.244e+02  1.988e+02   3.141 0.001704 **
## genreDrama:year             5.223e+02  1.558e+02   3.353 0.000812 ***
## genreHorror:year            1.227e+03  2.285e+02   5.369 8.60e-08 ***
## genreOthers:year            7.052e+01  3.764e+02   0.187 0.851403
## log_budget:ratingPG        -3.058e+03  1.697e+03  -1.802 0.071689 .
## log_budget:ratingPG-13     -4.265e+02  1.579e+03  -0.270 0.787120
## log_budget:ratingR/NC-17   -1.071e+03  1.427e+03  -0.751 0.453016
## log_runtime:year           -1.833e+03  3.883e+02  -4.721 2.47e-06 ***
## ratingPG:year               1.563e+03  2.889e+02   5.412 6.82e-08 ***
## ratingPG-13:year            1.545e+03  2.797e+02   5.526 3.60e-08 ***
## ratingR/NC-17:year          1.226e+03  2.667e+02   4.598 4.47e-06 ***
## genreAdventure:log_runtime -4.130e+04  1.658e+04  -2.490 0.012831 *
## genreAnimation:log_runtime -7.310e+04  2.773e+04  -2.636 0.008435 **
## genreBiography:log_runtime -4.256e+04  1.780e+04  -2.391 0.016887 *
## genreComedy:log_runtime     5.844e+02  1.180e+04   0.050 0.960493
## genreCrime:log_runtime      2.355e+04  1.804e+04   1.305 0.191880
## genreDrama:log_runtime     -2.886e+03  1.220e+04  -0.236 0.813095
## genreHorror:log_runtime     3.182e+04  2.727e+04   1.167 0.243334
## genreOthers:log_runtime     8.445e+04  4.137e+04   2.041 0.041328 *
## log_runtime:seasonSpring    9.274e+03  9.359e+03   0.991 0.321829
## log_runtime:seasonSummer    8.802e+03  9.103e+03   0.967 0.333688
## log_runtime:seasonWinter    8.599e+03  8.827e+03   0.974 0.330065
## year:seasonSpring          -1.879e+02  1.219e+02  -1.542 0.123300
## year:seasonSummer          -1.540e+02  1.192e+02  -1.292 0.196324
## year:seasonWinter          -8.193e+01  1.241e+02  -0.660 0.509064
## log_budget:tierB            1.964e+01  1.232e+03   0.016 0.987283
## log_budget:tierC           -1.370e+02  1.515e+03  -0.090 0.927966
## tierB:ratingPG              6.838e+03  8.190e+03   0.835 0.403895
## tierC:ratingPG              1.030e+04  9.576e+03   1.076 0.282090
## tierB:ratingPG-13          -6.597e+03  8.010e+03  -0.824 0.410253
## tierC:ratingPG-13          -7.463e+02  9.358e+03  -0.080 0.936441
## tierB:ratingR/NC-17        -4.716e+03  7.517e+03  -0.627 0.530421
## tierC:ratingR/NC-17        -3.263e+03  8.872e+03  -0.368 0.713077
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23800 on 2606 degrees of freedom
## Multiple R-squared:  0.6057, Adjusted R-squared:  0.5957
## F-statistic: 60.65 on 66 and 2606 DF,  p-value: < 2.2e-16
```

```r
##percent change in parameter estimation
round(abs(coef(step.f2)-coef(fit.fs2.v))/abs(coef(step.f2))*100,3)
```

```
##                (Intercept)                 log_budget
##                    207.479                      6.975
##             genreAdventure             genreAnimation
```

```
##                       11.538                      79.627
##               genreBiography                  genreComedy
##                       64.282                      45.155
##                   genreCrime                   genreDrama
##                       96.494                       4.137
##                  genreHorror                  genreOthers
##                        3.543                      68.131
##                  log_runtime                        tierB
##                       24.667                      82.326
##                        tierC                     ratingPG
##                       36.618                     179.864
##                  ratingPG-13                ratingR/NC-17
##                      108.053                     238.772
##                         year                 seasonSpring
##                     1129.627                    1133.358
##                 seasonSummer                 seasonWinter
##                       39.496                     228.964
##    log_budget:genreAdventure    log_budget:genreAnimation
##                      172.547                    3396.298
##    log_budget:genreBiography       log_budget:genreComedy
##                       78.088                      67.046
##        log_budget:genreCrime        log_budget:genreDrama
##                      208.715                      26.721
##       log_budget:genreHorror       log_budget:genreOthers
##                       20.841                     509.159
##       log_budget:log_runtime              log_budget:year
##                       37.205                      11.365
##           genreAdventure:year          genreAnimation:year
##                       12.044                      76.583
##           genreBiography:year             genreComedy:year
##                       33.314                      46.276
##               genreCrime:year              genreDrama:year
##                      117.533                      21.720
##              genreHorror:year             genreOthers:year
##                        3.874                      90.397
##          log_budget:ratingPG       log_budget:ratingPG-13
##                      329.254                     124.876
##     log_budget:ratingR/NC-17             log_runtime:year
##                       19.947                      25.690
##                ratingPG:year             ratingPG-13:year
##                      196.988                     117.157
##           ratingR/NC-17:year genreAdventure:log_runtime
##                      245.090                     591.566
## genreAnimation:log_runtime genreBiography:log_runtime
##                       26.884                     184.984
##       genreComedy:log_runtime       genreCrime:log_runtime
##                       90.803                       7.469
##        genreDrama:log_runtime      genreHorror:log_runtime
##                      107.823                      60.869
##       genreOthers:log_runtime     log_runtime:seasonSpring
##                      783.363                      46.221
##     log_runtime:seasonSummer     log_runtime:seasonWinter
##                       56.397                     209.209
##            year:seasonSpring            year:seasonSummer
```

```
##                      721.365                        43.177
##           year:seasonWinter               log_budget:tierB
##                      215.974                        98.231
##            log_budget:tierC                tierB:ratingPG
##                       85.783                        77.079
##              tierC:ratingPG             tierB:ratingPG-13
##                       63.615                       124.383
##           tierC:ratingPG-13           tierB:ratingR/NC-17
##                      103.017                       117.064
##         tierC:ratingR/NC-17
##                      112.746
```

```r
##percent change in standard errors
sd.fs2<- summary(step.f2)$coefficients[,"Std. Error"]
sd.fs2.v<- summary(fit.fs2.v)$coefficients[,"Std. Error"]
round(abs(sd.fs2-sd.fs2.v)/sd.fs2*100,3)
```

```
##                  (Intercept)                    log_budget
##                        3.886                         4.733
##               genreAdventure                genreAnimation
##                        6.881                         7.254
##               genreBiography                   genreComedy
##                        5.110                         2.563
##                   genreCrime                    genreDrama
##                       16.313                         0.637
##                  genreHorror                   genreOthers
##                        6.605                         7.487
##                  log_runtime                         tierB
##                        4.220                        14.474
##                        tierC                      ratingPG
##                       11.292                        12.270
##                  ratingPG-13                  ratingR/NC-17
##                       14.581                        14.754
##                         year                  seasonSpring
##                        3.841                         6.127
##                 seasonSummer                  seasonWinter
##                        5.117                         7.367
##    log_budget:genreAdventure     log_budget:genreAnimation
##                       17.815                         7.244
##    log_budget:genreBiography        log_budget:genreComedy
##                        0.184                         2.835
##        log_budget:genreCrime         log_budget:genreDrama
##                       12.069                         5.205
##       log_budget:genreHorror        log_budget:genreOthers
##                        9.769                        29.146
##       log_budget:log_runtime               log_budget:year
##                        8.070                         4.747
##          genreAdventure:year           genreAnimation:year
##                        5.640                         6.890
##          genreBiography:year              genreComedy:year
##                        4.637                         2.711
##              genreCrime:year               genreDrama:year
##                       17.680                         0.688
##             genreHorror:year              genreOthers:year
##                        2.425                         6.268
```

```
##        log_budget:ratingPG     log_budget:ratingPG-13
##                       7.187                     14.149
##    log_budget:ratingR/NC-17           log_runtime:year
##                      12.896                      4.199
##               ratingPG:year           ratingPG-13:year
##                      11.885                     14.462
##          ratingR/NC-17:year genreAdventure:log_runtime
##                      14.605                      3.628
## genreAnimation:log_runtime genreBiography:log_runtime
##                       5.704                     12.814
##     genreComedy:log_runtime      genreCrime:log_runtime
##                       4.765                     20.045
##      genreDrama:log_runtime     genreHorror:log_runtime
##                       2.835                      4.865
##     genreOthers:log_runtime   log_runtime:seasonSpring
##                      17.467                      5.026
##    log_runtime:seasonSummer   log_runtime:seasonWinter
##                       5.693                      2.306
##           year:seasonSpring          year:seasonSummer
##                       5.704                      4.254
##           year:seasonWinter           log_budget:tierB
##                       6.333                      9.731
##            log_budget:tierC             tierB:ratingPG
##                       8.228                      7.913
##              tierC:ratingPG          tierB:ratingPG-13
##                       6.428                      5.329
##           tierC:ratingPG-13        tierB:ratingR/NC-17
##                       6.001                      8.845
##         tierC:ratingR/NC-17
##                       8.581
```

```
##mean squared prediction error
pred.fs2<-predict.lm(step.f2, valid[,-1])
mspe.fs2<-mean((pred.fs2-valid[,1])^2)
mspe.fs2 #600040753, smaller than mspe.fs1
```

```
## [1] 600040753
```

```
press.fs2/n #618694544
```

```
## [1] 618694544
```

```
mse.fs2 #597285255
```

```
## [1] 597285255
```

```
# Candidate Model 3
fit.fs3.v<-lm(step.f3,data=valid)
summary(step.f3)
```

```
##
## Call:
## lm(formula = log_gross_4 ~ log_budget + tier + genre + log_runtime +
##     year + rating + season + log_budget:log_runtime + log_budget:year +
##     log_runtime:year, data = train)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -89706 -16178    295  16336 133745
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1025309.03 3080081.64   0.333 0.739248
## log_budget             -880826.48   73499.76 -11.984  < 2e-16 ***
## tierB                    -7652.38    1923.53  -3.978 7.13e-05 ***
## tierC                   -14359.16    2210.61  -6.496 9.85e-11 ***
## genreAdventure           -1857.89    2443.20  -0.760 0.447064
## genreAnimation           23323.02    3148.79   7.407 1.73e-13 ***
## genreBiography          -11033.09    2653.16  -4.158 3.31e-05 ***
## genreComedy              -1525.11    1384.03  -1.102 0.270592
## genreCrime               -7244.27    2197.75  -3.296 0.000993 ***
## genreDrama               -8693.33    1690.65  -5.142 2.92e-07 ***
## genreHorror              13185.22    2662.60   4.952 7.81e-07 ***
## genreOthers               2400.94    4146.70   0.579 0.562639
## log_runtime            2446079.38  720690.65   3.394 0.000699 ***
## year                        60.20    1566.83   0.038 0.969357
## ratingPG                  3850.48    3201.67   1.203 0.229220
## ratingPG-13               1881.94    3279.41   0.574 0.566107
## ratingR/NC-17            -4301.56    3182.33  -1.352 0.176586
## seasonSpring              2194.91    1364.91   1.608 0.107933
## seasonSummer              7891.26    1360.13   5.802 7.34e-09 ***
## seasonWinter              3355.90    1408.00   2.383 0.017221 *
## log_budget:log_runtime   18628.26    2585.58   7.205 7.55e-13 ***
## log_budget:year            404.23      36.63  11.036  < 2e-16 ***
## log_runtime:year         -1361.71     365.37  -3.727 0.000198 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24950 on 2649 degrees of freedom
## Multiple R-squared:  0.5648, Adjusted R-squared:  0.5612
## F-statistic: 156.3 on 22 and 2649 DF,  p-value: < 2.2e-16
```

```r
summary(fit.fs3.v)
```

```
##
## Call:
## lm(formula = step.f3, data = valid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -127218  -15186     673   15615   80126
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -3.060e+06  2.959e+06  -1.034 0.301147
## log_budget             -8.476e+05  7.091e+04 -11.952  < 2e-16 ***
## tierB                  -1.201e+04  1.853e+03  -6.482 1.07e-10 ***
## tierC                  -1.726e+04  2.154e+03  -8.012 1.68e-15 ***
## genreAdventure         -3.952e+03  2.390e+03  -1.654 0.098299 .
## genreAnimation          1.552e+04  3.044e+03   5.097 3.69e-07 ***
## genreBiography         -1.420e+04  2.471e+03  -5.747 1.01e-08 ***
## genreComedy            -4.779e+03  1.378e+03  -3.469 0.000531 ***
```

```
## genreCrime            -9.624e+03  2.111e+03  -4.558 5.40e-06 ***
## genreDrama            -9.412e+03  1.659e+03  -5.674 1.55e-08 ***
## genreHorror            1.232e+04  2.563e+03   4.807 1.62e-06 ***
## genreOthers            1.680e+03  4.167e+03   0.403 0.686784
## log_runtime            3.146e+06  6.889e+05   4.567 5.16e-06 ***
## year                   2.245e+03  1.509e+03   1.487 0.137006
## ratingPG               1.312e+04  3.018e+03   4.346 1.44e-05 ***
## ratingPG-13            1.260e+04  3.048e+03   4.133 3.70e-05 ***
## ratingR/NC-17          3.793e+03  2.953e+03   1.284 0.199141
## seasonSpring           3.962e+03  1.334e+03   2.971 0.002999 **
## seasonSummer           8.899e+03  1.331e+03   6.687 2.77e-11 ***
## seasonWinter           6.620e+03  1.346e+03   4.917 9.35e-07 ***
## log_budget:log_runtime  2.198e+04  2.594e+03   8.471  < 2e-16 ***
## log_budget:year        3.798e+02  3.528e+01  10.767  < 2e-16 ***
## log_runtime:year      -1.743e+03  3.502e+02  -4.978 6.84e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24410 on 2650 degrees of freedom
## Multiple R-squared:  0.5783, Adjusted R-squared:  0.5748
## F-statistic: 165.2 on 22 and 2650 DF,  p-value: < 2.2e-16
```

```r
##percent change in parameter estimation
round(abs(coef(step.f3)-coef(fit.fs3.v))/abs(coef(step.f3))*100,3)
```

```
##            (Intercept)             log_budget                  tierB
##                398.433                  3.778                 56.956
##                  tierC         genreAdventure         genreAnimation
##                 20.203                112.742                 33.462
##         genreBiography            genreComedy             genreCrime
##                 28.684                213.362                 32.854
##             genreDrama            genreHorror            genreOthers
##                  8.271                  6.557                 30.012
##            log_runtime                   year               ratingPG
##                 28.634               3630.021                240.683
##            ratingPG-13          ratingR/NC-17           seasonSpring
##                569.352                188.181                 80.524
##           seasonSummer           seasonWinter log_budget:log_runtime
##                 12.767                 97.263                 17.981
##        log_budget:year       log_runtime:year
##                  6.038                 28.018
```

```r
##percent change in standard errors
sd.fs3<- summary(step.f3)$coefficients[,"Std. Error"]
sd.fs3.v<- summary(fit.fs3.v)$coefficients[,"Std. Error"]
round(abs(sd.fs3-sd.fs3.v)/sd.fs3*100,3)
```

```
##            (Intercept)             log_budget                  tierB
##                  3.940                  3.519                  3.673
##                  tierC         genreAdventure         genreAnimation
##                  2.547                  2.176                  3.313
##         genreBiography            genreComedy             genreCrime
##                  6.881                  0.458                  3.925
##             genreDrama            genreHorror            genreOthers
##                  1.878                  3.743                  0.487
##            log_runtime                   year               ratingPG
```

```
##                  4.413                  3.661                  5.721
##            ratingPG-13          ratingR/NC-17           seasonSpring
##                  7.052                  7.193                  2.276
##           seasonSummer           seasonWinter log_budget:log_runtime
##                  2.159                  4.370                  0.341
##        log_budget:year       log_runtime:year
##                  3.689                  4.155
```

```
##mean squared prediction error
pred.fs3<-predict.lm(step.f3, valid[,-1])
mspe.fs3<-mean((pred.fs3-valid[,1])^2)
mspe.fs3 #602561604 smaller than mspe.fs2
```

```
## [1] 602561604
```

```
press.fs3/n #628834176
```

```
## [1] 628834176
```

```
mse.fs3 #622488616
```

```
## [1] 622488616
```

```
# Candidate Model 2 is the final model
```

Model Diagnostics

```
# fit Candidate Model 2 on whole data
dt.split$log_gross_4 <- log(dt.split$gross)^4
dt.split$log_budget <- log(dt.split$budget)
dt.split$log_runtime <- log(dt.split$runtime)

dt.split <- subset(dt.split,select=c(log_gross_4, log_budget, log_runtime, year, rating, genre, tier, s

fit.fs2.final<-lm(step.f2, data=dt.split)
summary(fit.fs2.final)
```

```
##
## Call:
## lm(formula = step.f2, data = dt.split)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -115175  -15625     401   15933   84745
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -3.344e+05  2.369e+06  -0.141 0.887753
## log_budget                 -8.125e+05  6.058e+04 -13.412  < 2e-16 ***
## genreAdventure             -7.485e+05  3.314e+05  -2.259 0.023952 *
## genreAnimation             -6.893e+05  5.143e+05  -1.340 0.180189
## genreBiography             -5.790e+05  3.415e+05  -1.695 0.090087 .
## genreComedy                -3.487e+05  1.891e+05  -1.844 0.065302 .
## genreCrime                 -1.019e+06  3.064e+05  -3.326 0.000887 ***
## genreDrama                 -8.902e+05  2.197e+05  -4.052 5.15e-05 ***
## genreHorror                -2.370e+06  3.254e+05  -7.282 3.76e-13 ***
## genreOthers                -9.360e+05  5.487e+05  -1.706 0.088071 .
```

```
## log_runtime                2.967e+06  5.503e+05   5.392 7.26e-08 ***
## tierB                     -2.333e+04  1.558e+04  -1.497 0.134406    
## tierC                     -1.194e+04  1.881e+04  -0.635 0.525493    
## ratingPG                  -2.203e+06  4.230e+05  -5.207 1.99e-07 ***
## ratingPG-13               -2.428e+06  4.183e+05  -5.806 6.79e-09 ***
## ratingR/NC-17             -1.705e+06  3.983e+05  -4.282 1.89e-05 ***
## year                       8.482e+02  1.200e+03   0.707 0.479757    
## seasonSpring               1.832e+05  1.744e+05   1.050 0.293732    
## seasonSummer               3.966e+05  1.712e+05   2.317 0.020545 *  
## seasonWinter               6.143e+04  1.798e+05   0.342 0.732651    
## log_budget:genreAdventure -1.698e+03  1.706e+03  -0.995 0.319748    
## log_budget:genreAnimation  5.699e+03  2.059e+03   2.768 0.005656 ** 
## log_budget:genreBiography -5.490e+03  1.889e+03  -2.907 0.003665 ** 
## log_budget:genreComedy    -3.090e+03  9.384e+02  -3.292 0.001000 ***
## log_budget:genreCrime     -3.674e+03  1.545e+03  -2.379 0.017407 *  
## log_budget:genreDrama     -6.983e+03  1.048e+03  -6.664 2.93e-11 ***
## log_budget:genreHorror    -1.156e+04  1.523e+03  -7.592 3.70e-14 ***
## log_budget:genreOthers    -5.198e+03  2.417e+03  -2.151 0.031527 *  
## log_budget:log_runtime     2.134e+04  2.135e+03   9.997  < 2e-16 ***
## log_budget:year            3.661e+02  2.974e+01  12.312  < 2e-16 ***
## genreAdventure:year        4.332e+02  1.671e+02   2.593 0.009534 ** 
## genreAnimation:year        4.503e+02  2.726e+02   1.652 0.098568 .  
## genreBiography:year        3.365e+02  1.678e+02   2.005 0.044977 *  
## genreComedy:year           1.899e+02  9.555e+01   1.987 0.046935 *  
## genreCrime:year            4.932e+02  1.526e+02   3.231 0.001240 ** 
## genreDrama:year            4.612e+02  1.104e+02   4.176 3.01e-05 ***
## genreHorror:year           1.155e+03  1.625e+02   7.106 1.35e-12 ***
## genreOthers:year           4.480e+02  2.695e+02   1.662 0.096489 .  
## log_budget:ratingPG       -9.322e+02  1.222e+03  -0.763 0.445504    
## log_budget:ratingPG-13     3.142e+02  1.184e+03   0.265 0.790770    
## log_budget:ratingR/NC-17  -1.430e+03  1.059e+03  -1.350 0.177068    
## log_runtime:year          -1.654e+03  2.786e+02  -5.935 3.12e-09 ***
## ratingPG:year              1.105e+03  2.136e+02   5.176 2.35e-07 ***
## ratingPG-13:year           1.210e+03  2.096e+02   5.773 8.22e-09 ***
## ratingR/NC-17:year         8.601e+02  1.996e+02   4.309 1.67e-05 ***
## genreAdventure:log_runtime -1.949e+04  1.138e+04  -1.713 0.086850 .  
## genreAnimation:log_runtime -6.608e+04  2.001e+04  -3.303 0.000962 ***
## genreBiography:log_runtime -2.398e+03  1.330e+04  -0.180 0.856880    
## genreComedy:log_runtime     4.331e+03  8.109e+03   0.534 0.593322    
## genreCrime:log_runtime      1.882e+04  1.148e+04   1.640 0.101146    
## genreDrama:log_runtime      1.634e+04  8.688e+03   1.881 0.060055 .  
## genreHorror:log_runtime     5.654e+04  1.860e+04   3.040 0.002379 ** 
## genreOthers:log_runtime     2.834e+04  2.576e+04   1.100 0.271297    
## log_runtime:seasonSpring    1.270e+04  6.756e+03   1.879 0.060263 .  
## log_runtime:seasonSummer    1.605e+04  6.594e+03   2.433 0.014991 *  
## log_runtime:seasonWinter    1.710e+03  6.258e+03   0.273 0.784684    
## year:seasonSpring          -1.198e+02  8.839e+01  -1.355 0.175513    
## year:seasonSummer          -2.316e+02  8.575e+01  -2.701 0.006929 ** 
## year:seasonWinter          -3.217e+01  9.018e+01  -0.357 0.721322    
## log_budget:tierB            2.256e+02  9.013e+02   0.250 0.802344    
## log_budget:tierC           -8.522e+02  1.101e+03  -0.774 0.438934    
## tierB:ratingPG              1.677e+04  5.949e+03   2.819 0.004831 ** 
## tierC:ratingPG              1.810e+04  6.915e+03   2.618 0.008867 ** 
## tierB:ratingPG-13           9.106e+03  5.758e+03   1.582 0.113809    
```

```
## tierC:ratingPG-13          1.064e+04  6.767e+03   1.572 0.115908
## tierB:ratingR/NC-17        9.599e+03  5.469e+03   1.755 0.079274 .
## tierC:ratingR/NC-17        9.182e+03  6.476e+03   1.418 0.156334
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24200 on 5278 degrees of freedom
## Multiple R-squared:  0.5896, Adjusted R-squared:  0.5844
## F-statistic: 114.9 on 66 and 5278 DF,  p-value: < 2.2e-16
```

```
anova(fit.fs2.final)
```

```
## Analysis of Variance Table
##
## Response: log_gross_4
##                          Df     Sum Sq    Mean Sq   F value    Pr(>F)
## log_budget                1 3.3261e+12 3.3261e+12 5677.2940 < 2.2e-16 ***
## genre                     8 2.0890e+11 2.6112e+10   44.5699 < 2.2e-16 ***
## log_runtime               1 1.9874e+11 1.9874e+11  339.2237 < 2.2e-16 ***
## tier                      2 9.2860e+10 4.6430e+10   79.2500 < 2.2e-16 ***
## rating                    3 9.8391e+10 3.2797e+10   55.9802 < 2.2e-16 ***
## year                      1 7.7087e+10 7.7087e+10  131.5771 < 2.2e-16 ***
## season                    3 4.6414e+10 1.5471e+10   26.4079 < 2.2e-16 ***
## log_budget:genre          8 8.9146e+10 1.1143e+10   19.0201 < 2.2e-16 ***
## log_budget:log_runtime    1 7.2210e+10 7.2210e+10  123.2524 < 2.2e-16 ***
## log_budget:year           1 1.0557e+11 1.0557e+11  180.2024 < 2.2e-16 ***
## genre:year                8 3.8380e+10 4.7975e+09    8.1887 4.521e-11 ***
## log_budget:rating         3 7.6902e+09 2.5634e+09    4.3754  0.004402 **
## log_runtime:year          1 2.0587e+10 2.0587e+10   35.1391 3.264e-09 ***
## rating:year               3 2.3477e+10 7.8255e+09   13.3572 1.106e-08 ***
## genre:log_runtime         8 1.9671e+10 2.4589e+09    4.1970 5.026e-05 ***
## log_runtime:season        3 3.8233e+09 1.2744e+09    2.1753  0.088781 .
## year:season               3 5.0512e+09 1.6837e+09    2.8739  0.034864 *
## log_budget:tier           2 1.0887e+09 5.4434e+08    0.9291  0.394969
## tier:rating               6 6.3615e+09 1.0603e+09    1.8097  0.093084 .
## Residuals              5278 3.0922e+12 5.8587e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow = c(2, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
plot(fit.fs2.final, sub.caption = "Figure 8: Diagnostic Plots for Final Model")
```

```
# outlying and influential cases
n.s <- nrow(dt.split)
res <- residuals(fit.fs2.final)
p <- length(fit.fs2.final$coefficients)
h1 <- influence(fit.fs2.final)$hat
d.res.std <- studres(fit.fs2.final) #studentized deleted residuals
qt(1-0.05/(2*n.s),n.s-p) # bonferronis thresh hold
```

```
## [1] 4.435915
```

```
idx.Y <- as.vector(which(abs(d.res.std)>=qt(1-0.1/(2*n.s),n.s-p)))
#idx.Y ## outliers in Y
length(idx.Y)
```

```
## [1] 1
```

```
idx.X <- as.vector(which(h1>(2*p/n.s)))
#idx.X ## outliers in X
length(idx.X)
```

```
## [1] 548
```

```
#plot(h1,res,xlab="leverage",ylab="residuals")
par(mfrow=c(1,1))
plot(fit.fs2.final, which=4, main = "Figure 9: Cook's Distance", caption = "" )
```

```
##cooksd <- cooks.distance(fit.fs2.final)
##n <- nrow(dt.split)
##influential <- as.numeric(names(cooksd)[(cooksd > (4/n))])
##df_screen <- dt.split[-influential, ]

#Case 881, 1805, 4442 is an influential case according to Cook's distance

influential <- c(881, 1805, 4442)
fit.fs2.final2<-lm(fit.fs2.final, data=dt.split[-influential,])
par(mfrow = c(2, 2),oma=c(2,2,2,2),mar=c(4,3,3,3))
plot(fit.fs2.final2, sub.caption = "Figure 9: Diagnostic Plots for Final Model without Influential Case
```

```
f1<-fitted(fit.fs2.final)
f2<-fitted(fit.fs2.final2)
SUM<-sum(abs((f1[-influential]-f2)/f1[-influential]))
SUM<-SUM+abs((f1[influential]-predict(fit.fs2.final,newdata = dt.split[influential,]))/f1[influential])
per.average<-SUM/n.s
per.average
```

```
##        902       1849       4533
## 0.00100251 0.00100251 0.00100251
```

```
# No case is removed
```

Bootstrap

```
library(boot)
set.seed(2021)

model_coef <- function(data, i){
  d <- data[i,]
  fit <- lm(step.f2, data=d)
  return(coef(fit))
}

coeff <- boot(data=dt.split, statistic= model_coef, R=1000)

coeff
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = dt.split, statistic = model_coef, R = 1000)
```

```
##
##
## Bootstrap Statistics :
##           original        bias     std. error
## t1*  -3.343765e+05  4.621735e+04 2.803274e+06
## t2*  -8.125443e+05 -7.915358e+03 6.741213e+04
## t3*  -7.484556e+05  1.450402e+04 3.643172e+05
## t4*  -6.892917e+05  4.768616e+04 5.897861e+05
## t5*  -5.789723e+05  1.104027e+04 4.167428e+05
## t6*  -3.486978e+05 -5.918612e+03 1.866280e+05
## t7*  -1.019197e+06 -3.127004e+03 3.185119e+05
## t8*  -8.902362e+05 -1.014567e+04 2.429246e+05
## t9*  -2.369750e+06 -5.017108e+03 3.625512e+05
## t10* -9.360414e+05  1.139331e+04 5.722025e+05
## t11*  2.967412e+06  2.761634e+04 6.529946e+05
## t12* -2.333148e+04 -1.406276e+03 1.686459e+04
## t13* -1.194303e+04 -1.828338e+03 2.028627e+04
## t14* -2.203020e+06 -2.762387e+04 5.046326e+05
## t15* -2.428496e+06 -1.915614e+04 4.877757e+05
## t16* -1.705497e+06 -2.784591e+04 4.719416e+05
## t17*  8.482446e+02 -3.062632e+01 1.432337e+03
## t18*  1.831675e+05  6.224597e+03 1.774253e+05
## t19*  3.965688e+05 -5.973453e+03 1.731317e+05
## t20*  6.142952e+04  2.834644e+03 1.809121e+05
## t21* -1.697825e+03  9.886877e+01 1.912147e+03
## t22*  5.699328e+03  3.867338e+02 2.870052e+03
## t23* -5.490348e+03  3.130200e+02 2.276641e+03
## t24* -3.089699e+03 -1.601011e+01 9.257017e+02
## t25* -3.674323e+03  1.025558e+02 1.627404e+03
## t26* -6.982589e+03  1.586158e+01 1.120382e+03
## t27* -1.156282e+04  5.267382e+01 2.007890e+03
## t28* -5.198044e+03  1.305190e+02 2.501277e+03
## t29*  2.133809e+04 -1.454977e+02 2.419109e+03
## t30*  3.661414e+02  4.307837e+00 3.284898e+01
## t31*  4.332512e+02 -9.229379e+00 1.921018e+02
## t32*  4.502932e+02 -2.548130e+01 3.167035e+02
## t33*  3.365157e+02 -7.500283e+00 1.942870e+02
## t34*  1.898960e+02  5.108729e+00 9.649625e+01
## t35*  4.931986e+02  1.615105e+00 1.619396e+02
## t36*  4.611857e+02  5.518914e+00 1.231000e+02
## t37*  1.154959e+03  2.126031e+00 1.834657e+02
## t38*  4.479895e+02 -3.495454e+00 2.983190e+02
## t39* -9.322277e+02 -8.906468e+01 1.427945e+03
## t40*  3.141672e+02 -9.172539e+01 1.381856e+03
## t41* -1.430123e+03 -6.893691e+01 1.212684e+03
## t42* -1.653618e+03 -1.220781e+01 3.327947e+02
## t43*  1.105470e+03  1.441973e+01 2.552888e+02
## t44*  1.210121e+03  1.020408e+01 2.442548e+02
## t45*  8.600685e+02  1.435983e+01 2.360160e+02
## t46* -1.949035e+04  4.849136e+02 1.582374e+04
## t47* -6.608411e+04 -8.135773e+02 2.016255e+04
## t48* -2.397976e+03 -2.550657e+02 1.993837e+04
## t49*  4.330642e+03 -8.626076e+02 8.692965e+03
## t50*  1.881677e+04 -3.709832e+02 1.247407e+04
```

```
## t51*  1.633959e+04 -2.404797e+02 1.013241e+04
## t52*  5.654036e+04  2.253911e+00 1.979156e+04
## t53*  2.833944e+04 -1.405757e+03 2.797132e+04
## t54*  1.269655e+04 -1.571968e+02 8.355639e+03
## t55*  1.604631e+04 -7.318880e+02 7.248504e+03
## t56*  1.709884e+03 -6.089079e+02 7.268200e+03
## t57* -1.197586e+02 -2.747174e+00 9.269095e+01
## t58* -2.316420e+02  4.717497e+00 8.806689e+01
## t59* -3.216991e+01 -3.578886e-03 9.085082e+01
## t60*  2.256218e+02  8.562872e+01 9.149091e+02
## t61* -8.522166e+02  7.090670e+01 1.143515e+03
## t62*  1.677316e+04  3.098690e+01 7.153011e+03
## t63*  1.810305e+04  5.670091e+02 7.976870e+03
## t64*  9.106102e+03 -4.143491e+01 6.586210e+03
## t65*  1.064100e+04  6.320769e+02 7.493780e+03
## t66*  9.599393e+03 -3.877014e+01 6.535085e+03
## t67*  9.181576e+03  6.666586e+02 7.327605e+03
```

```
par(mfrow=c(1,2))
hist(coeff$t[,2], xlab="bootstrap estimate beta* for log(gross)", main=NULL)
hist(coeff$t[,17], xlab="bootstrap estimate beta* for year", main=NULL)
mtext("Figure 10: Bootstrap Estimate Coefficients", side = 3, font=2, line=-1, outer=TRUE)
```

Discussion

```
library(carData)
library(effects)
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
#interaction

budget.genre <- effect('log_budget*genre', fit.fs2.final,
                                          se=TRUE, confidence.level=.95, typical=mean)

budget.rating <- effect('log_budget*rating', fit.fs2.final,
                                          se=TRUE, confidence.level=.95, typical=mean)

inter.budget1 <- as.data.frame(budget.genre)
inter.budget2 <- as.data.frame(budget.rating)

summary(inter.budget1$fit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -123118   -3556   44777   45161   92203  163977
```

```
summary(inter.budget2$fit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -57034   -6467   42313   42905   92127  140187
```

```
library(ggplot2)
plot.inter.budget1<-ggplot(data=inter.budget1, aes(x=log_budget, y=fit, group=genre))+
      coord_cartesian()+
      geom_line(size=2, aes(color=genre))+
```

```r
      ylab(expression(paste(log(gross)^4)))+
      xlab("log(budget)")+
      ggtitle("Figure 12: Interaction between log(budget) and genre")+
      theme_bw()+
        theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
      scale_fill_grey()


plot.inter.budget2<-ggplot(data=inter.budget2, aes(x=log_budget, y=fit, group=rating))+
      coord_cartesian()+
      geom_line(size=2, aes(color=rating))+
      ylab(expression(paste(log(gross)^4)))+
      xlab("log(budget)")+
      ggtitle("Figure 11: Interaction between log(budget) and rating")+
      theme_bw()+
        theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
      scale_fill_grey()

plot.inter.budget2
```

```r
plot.inter.budget1
```

```r
####################

year.genre <- effect('genre*year', fit.fs2.final,
                                          xlevels=list(year = c(1980:2019)),
                                          se=TRUE, confidence.level=.95, typical=mean)

year.rating <- effect('rating*year', fit.fs2.final,
                                          xlevels=list(year = c(1980:2019)),
                                          se=TRUE, confidence.level=.95, typical=mean)

inter.genre <- as.data.frame(year.genre)
inter.rating <- as.data.frame(year.rating)

summary(inter.genre$fit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   64783   77262   82904   83099   87282  119778
```

```r
summary(inter.rating$fit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   68606   77597   82044   82816   87883   97774
```

```r
library(ggplot2)
plot.inter.genre<-ggplot(data=inter.genre, aes(x=year, y=fit, group=genre))+
      coord_cartesian()+
      geom_line(size=2, aes(color=genre))+
      ylab(expression(paste(log(gross)^4)))+
      xlab("year")+
      ggtitle("Figure 14: Interaction between year and genre")+
      theme_bw()+
```

```
        theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
      scale_fill_grey()


plot.inter.rating<-ggplot(data=inter.rating, aes(x=year, y=fit, group=rating))+
      coord_cartesian()+
      geom_line(size=2, aes(color=rating))+
      ylab(expression(paste(log(gross)^4)))+
      xlab("year")+
      ggtitle("Figure 13: Interaction between year and rating")+
      theme_bw()+
        theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
      scale_fill_grey()

plot.inter.rating
```

```
plot.inter.genre
```

References

https://www.kaggle.com/danielgrijalvas/movies