STA141A - Project

Effects on Movies' Popularity - Group 20

December 6, 2021

Name	Contribution	E-mail
Yuhan Dai	Model Construction, Model Selection	dyhdai@ucdavis.edu
Carys Jian	Data Processing, Model Transformation	cjian@ucdavis.edu
Yuqing Wang	Data collection, Data Visualization	yqqwang@ucdavis.edu

Emanuela Furfaro Instructor

STA 141A - Fundamentals of Statistical Data Science

University of California, Davis

1. Introduction

The following paper analyzes the factors that affect the popularity of movies. Since the last century, the movies industry has grown rapidly. More and more directors showed their distinguished talents through any successful self-made movie. However, the audience also tends to become stricter after the director becomes famous. It is easier for the audience to give poor ratings if the movie does not meet their expectations. At the same time, the more famous directors also have more sufficient financial support. A more sufficient budget might also provide the movie with more experienced auxiliary teams. Thus, our goal is to use the models and tests to learn more about the factors affecting the popularity of each movie.

2. Data background and Questions of Interest

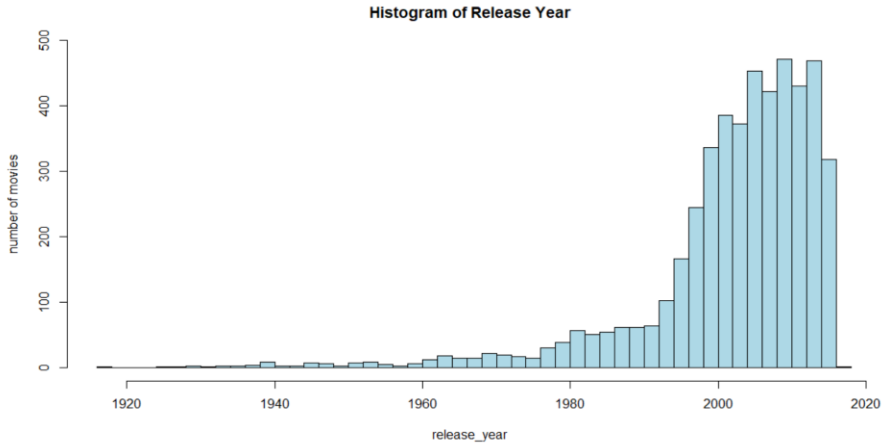
We collected the dataset from Kaggle. The dataset we decide to use contains 20 variables for 4803 movies, spanning across 100 years in 66 countries. It includes different information about the movies, including the budget, production companies, revenue, number of votes, runtime, genre, language, and so on. We would like to find the significant factors, which affect the popularity of the movies in the end. At the same time, building models to predict what specific kinds of movies would become more successful.

To achieve our goal, we decided to explore the relationship between the regressors and build linear regression models, Lasso models, Ridge models to analyze the potential factors affecting the response variable: movie popularity. Here are the questions we aimed to answer through the report:

1) What is the trend of the popularity of movies over time?
2) What factors affect the popularity of movies?
3) What is the most significant numeric factor determining the popularity of movies?

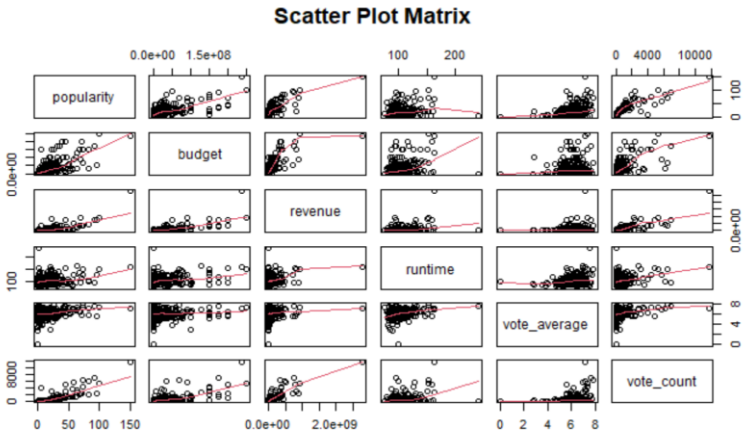
3. Data Visualization

3.1 Distribution of Release Year



We used the histogram above to find the distribution of all movies' release years. Then, we found that there are too many observations and the time for each movie released might also affect their popularity, such as the technology difference and economic situations between decades. Therefore, we decided to extract the movies released during 2009 and ignore the rest of the data. This would not affect the following research since the sample size (247) is large enough.

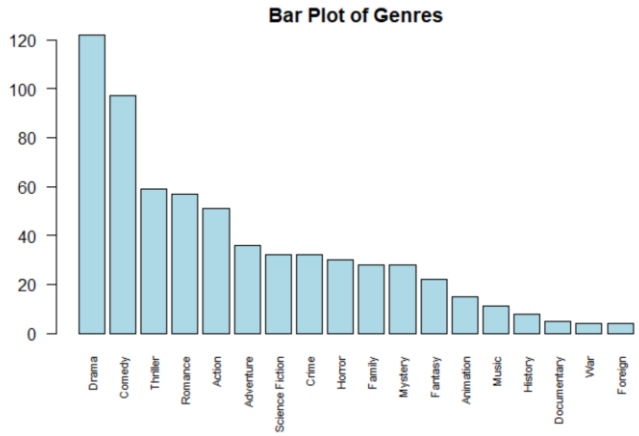
3.2 Correlation



From the Scatter Plot Matrix and the Correlation Table on the left, we found that the correlation between *budget and revenue* (0.790), *vote_count and revenue* (0.800), and *vote_count and budget* (0.675) are relatively strongly correlated.

	budget	revenue	runtime	vote_average	vote_count
budget	1.0000000	0.7898347	0.3148670	0.1693275	0.6751576
revenue	0.7898347	1.0000000	0.2713029	0.2106511	0.8004007
runtime	0.3148670	0.2713029	1.0000000	0.3597286	0.3258714
vote_average	0.1693275	0.2106511	0.3597286	1.0000000	0.3160553
vote_count	0.6751576	0.8004007	0.3258714	0.3160553	1.0000000

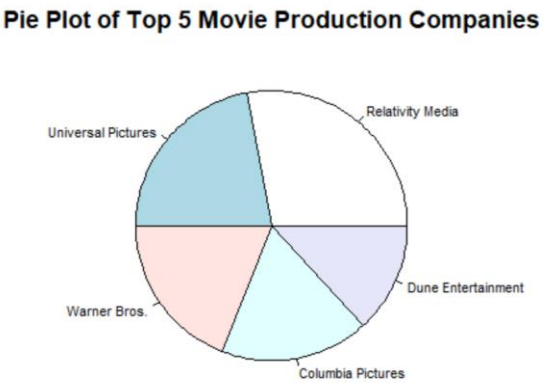
3.3 Distribution of Genres



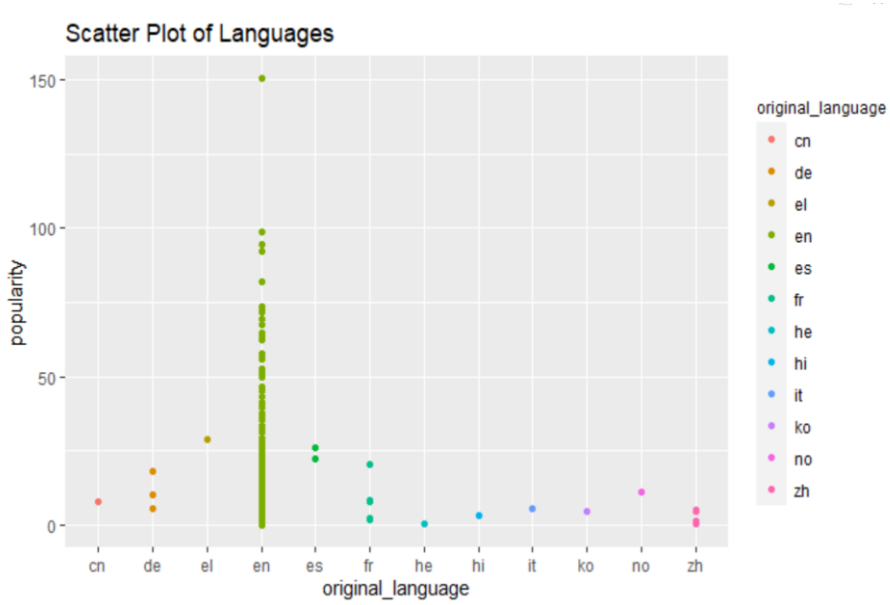
The distribution of genres for movies during 2009 on the left shows that drama, comedy, and thriller are the three genres with the most observations. At the same time, documentary, war, and foreign movies are the three genres with the least observations.

3.4 Distribution of Top 5 Movie Production Companies

We used the pie plot on the left to show the top 5 movie production companies' market share during 2009. It includes Warner Brothers, Universal Pictures, Dune Entertainment, Columbia Pictures, and Relativity Media. Additionally, the pie plot also shows that Relativity Media produced most movies during 2009.



3.5 Distribution of Languages



We used a scatter plot to show the distribution of movie language during 2009. From the diagram on the left, it is very clear that English movies are the movie category with the most observations compared to any other language. The rest movies also seem to share a similarly small share of the entire market.

4. Model Construction

4.1 Linear Regression

4.1.1 Full Model

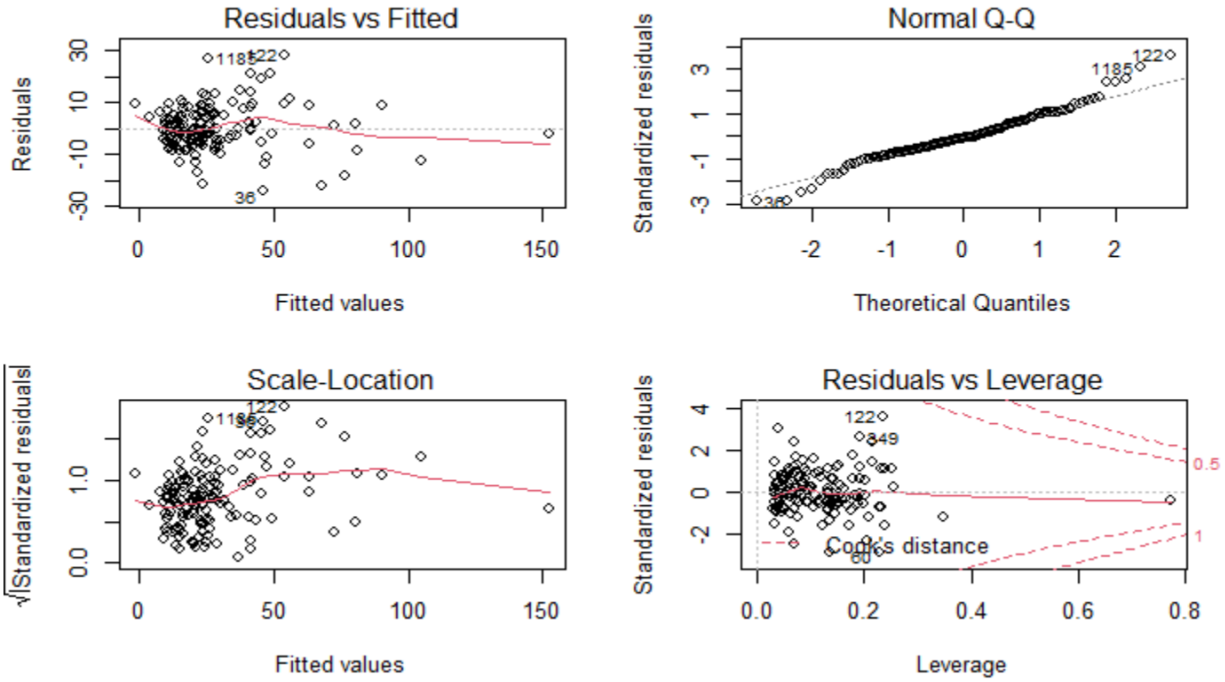
Full Model: popularity ~ budge + revenue + runtime + vote_average + vote_count + dDrama + dComedy + dThriller + dAction + dAdventure + dEnglish + dRelativity + dUniversal + dWarner + dColumbia + dDune

We constructed dummy variables for the 5 most common production companies, the 5 most common genres, and one dummy variable of whether the movie is in English. After regressing the categorical and numerical variables against the popularity, the full model has $R^2 = 0.8489$, which means about 84.89% of the variability in movies' popularity (Y) can be explained by its regression. In addition, the model has the adjusted $R^2 = 0.8305$ and residual standard error 8.99 of with 131 degrees of freedom. The F-test for a multiple regression model with an empty model obtained a small p-value (< 2.2e-16), which means we can reject the null hypothesis that all the coefficients are zeros.

4.1.2 Detection of Outliers

After checking the data away from 3 standard deviations, we found that there are two movies, *X-Men Origins: Wolverine* and *The Twilight Saga: New Moon*, which are the two hottest movies released in 2009. Thus, we decided to remove these two outliers.

4.1.3 Model Assumptions



I. According to the Residuals vs Fitted Plot, residuals roughly form a horizontal band around the 0-line. However, the data points do not bounce randomly around the 0-line, so the linearity assumptions are violated.

II. According to the Normal Q-Q Plot, most of the points are following the straight dash line, so the assumption of normality holds. However, since the sample size is larger than 30, this is not a big issue.

III. According to the Scale-Location Plot, the line is not that flat and the points are not equally spread, so we have a heteroscedasticity problem, and the equal variance assumptions are violated.

IV. According to the Residuals vs Leverage plot, there is no influence point outside the Cook's distance line.

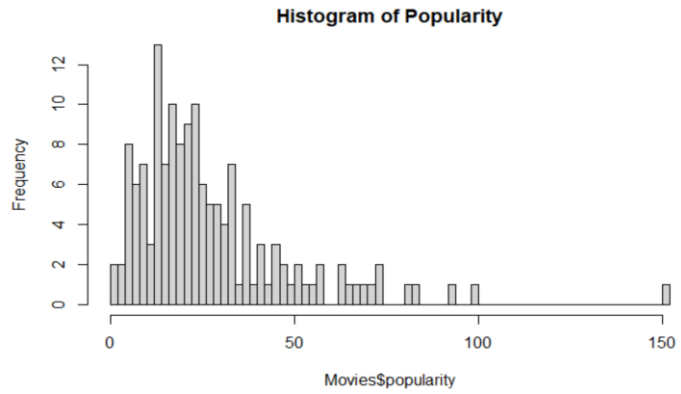
4.1.4 Reduced Model Decision

Reduced Model: popularity ~ vote_count + budget + runtime + vote_average + dThriller + dDune + dEnglish + dDrama + dColumbia

After the stepwise selections (backward, forward, both), 9 predictors are selected in the subset. According to the t-test for each coefficient, they all obtained a small p-value (< 2e-16 - 0.056643), which means all the variables are statistically significant.

4.2 Transformation of Linear Regression

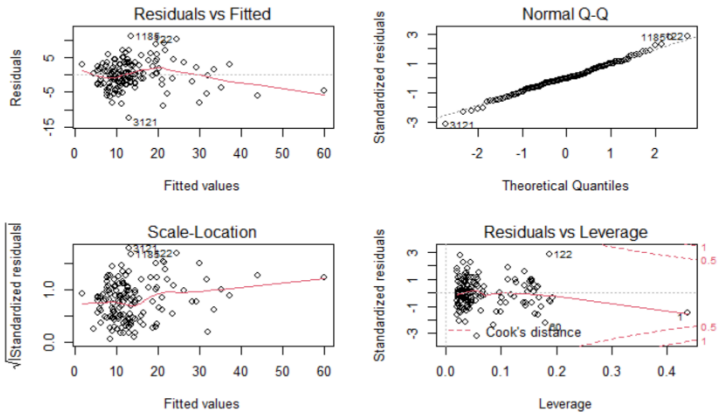
4.2.1 Distribution of Response Variable



The distribution of the response variable is right-skewed, and thus log and square root transformations can be used; the range of the response variable is 1.729 - 150.438.

4.2.2 Box-Cox Transformation

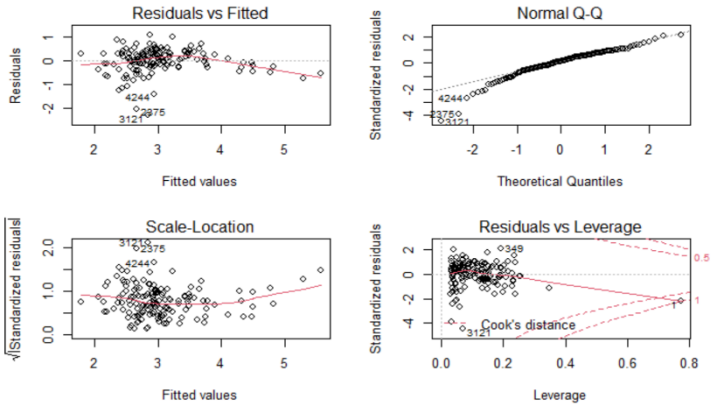
Reduced Model: popularity ~ vote_count + budget + runtime + vote_average + dThriller + dEnglish + dDrama + dDune + dColumbia



Since the model lacks linearity, we used Box-Cox transformation to fix the problem. Plotting the reduced model after the Box-Cox transformation, we can see that in the Normal Q-Q plot, the data points follow the straight dash line more carefully.

4.2.3 Log Transformation

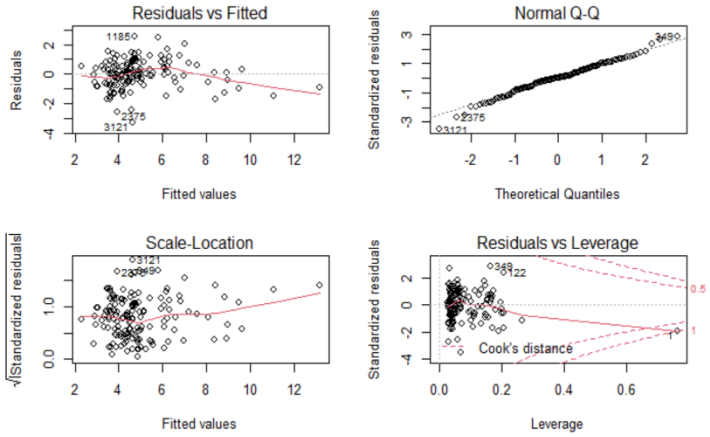
Reduced Model: popularity ~ vote_count + dEnglish + budget + revenue + dThriller + runtime + vote_average + dDrama + dColumbia + dComedy



Due to unequal variance, we tried to transform the response variable using log transformation. The diagnostic plot of the reduced model after the log transformation is shown below. Based on the Scale-Location plot, the log transformation improves the line by making it flatter and improves the data selection by more randomly and equally scattering around the line.

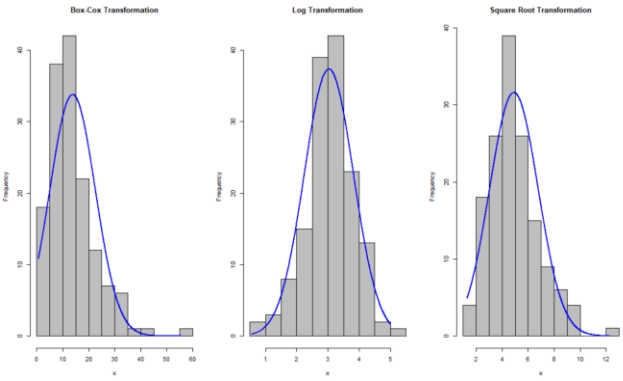
4.2.4 Square Root Transformation

Reduced Model: popularity ~ vote_count + budget + revenue + runtime + dComedy + vote_average + dEnglish + dDrama + dColumbia + dDune + dThriller



Because of the nonconstant variance, we also tried to transform the response variable using square root. The diagnostic plot of reduced models after the square root transformation shows that the horizontal level of the line and the distribution of points are improved in the Scale-Location plot. However, the improvement here is not as good as the log transformation.

4.2.5 Transformation of Linear Regression Decision



	linear.adjr2 <dbl>	boxcox.adjr2 <dbl>	log.adjr2 <dbl>	sqrt.adjr2 <dbl>
Full Model	0.8304857	0.7878760	0.5402299	0.7263857
Reduced Model	0.8343485	0.7904977	0.5427563	0.7298565

	linear.AIC <dbl>	boxcox.AIC <dbl>	log.AIC <dbl>	sqrt.AIC <dbl>
Full Model	668.0061	430.1792	-166.5313	11.029062
Reduced Model	658.2973	421.9893	-172.7177	4.686398

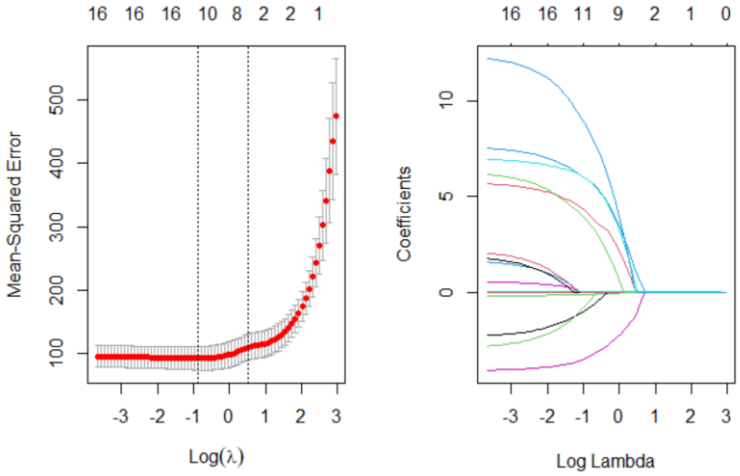
	linear.BIC <dbl>	boxcox.BIC <dbl>	log.BIC <dbl>	sqrt.BIC <dbl>
Full Model	721.9559	484.1291	-112.5814	64.97888
Reduced Model	691.2666	454.9586	-136.7511	43.65016

By looking at the distribution of the response variable with transformations above, log transformation can best improve the normality. Based on the criteria of adjusted r-squared, the preferred model is the reduced model of the original linear regression. However, the criteria of AIC and BIC both choose the reduced model with log transformation with the smallest value.

4.3 LASSO Regression

Based on the plot of the test MSE by lambda value, the optimal lambda minimized the test MSE equals 81.588. Then, the lasso regression model using the optimal lambda identified by the k-fold cross validation with k=10 removed 5 predictors from the model by approaching zero to the coefficients. The r-squared of the lasso regression model is 0.844025, which means about 84.4% of the variability in movies' popularity (Y) can be explained by the lasso regression.

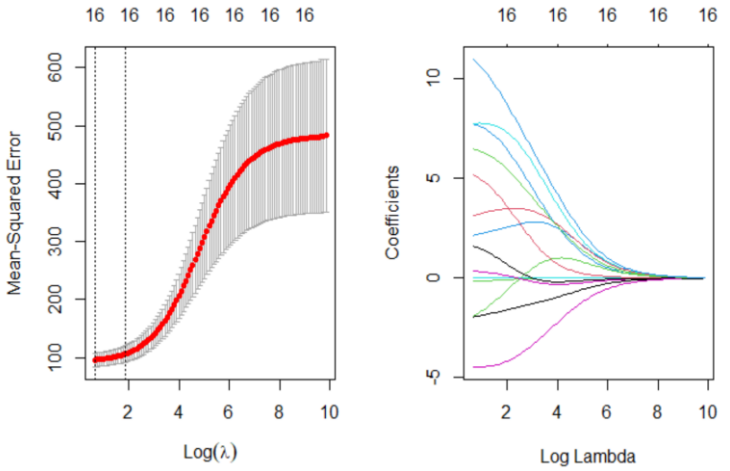
	s0
(Intercept)	-1.370579e+01
budget	9.698321e-08
revenue	.
runtime	-1.348051e-01
vote_average	5.861829e+00
vote_count	9.015883e-03
dDrama	-3.415660e+00
dComedy	-8.608679e-01
dThriller	4.117505e+00
dAction	-5.847792e-01
dAdventure	.
dEnglish	5.874774e+00
dRelativity	.
dUniversal	.
dWarner	.
dColumbia	3.686485e+00
dDune	8.543012e+00



4.4 Ridge Regression

We used a Trace plot to visualize the estimates of the coefficient according to the increase of the lambda. Based on the plot of the test MSE by lambda value, the optimal lambda minimized the test MSE equals 83.121. The r-squared of the ridge regression model is 0.8434663, which means about 84.35% of the variability in movies' popularity (Y) can be explained by the ridge regression.

	s0
(Intercept)	-2.364980e+01
budget	8.551089e-08
revenue	9.695175e-09
runtime	-1.577741e-01
vote_average	7.751482e+00
vote_count	6.780931e-03
dDrama	-4.446246e+00
dComedy	-1.960773e+00
dThriller	5.204713e+00
dAction	-1.935527e+00
dAdventure	2.142989e+00
dEnglish	7.753608e+00
dRelativity	3.620956e-01
dUniversal	1.625984e+00
dWarner	3.121302e+00
dColumbia	6.489623e+00
dDune	1.100652e+01



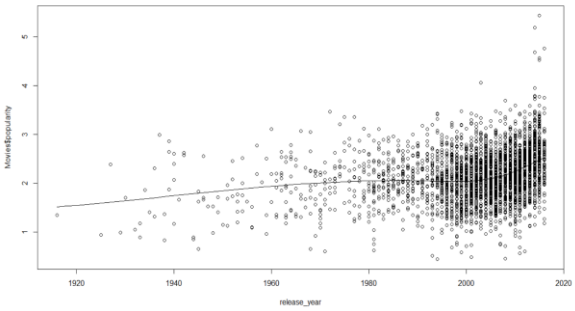
5. Model Evaluation and Selection

	MSE <dbl>	SSE <dbl>	R_squared <dbl>
Lasso.result	81.5877744	10932.76177	0.8440250
Ridge.result	83.1206344	10971.92374	0.8434663
Log.result	0.2859472	39.17477	0.5738613

Based on the table with R-squared, we concluded that Lasso regression explained more variability of movies' popularity with the regressors. Lasso regression is helpful because of the multicollinearity of the regressors in our data set.

6. Interpretation

1) What is the trend of the popularity of movies over time?



From the scatter plot release year against popularity on the left, we noticed that the number of movies being created got larger as time passed. There is a trend of increasing popularity of movies over time.

2) What factors affect the popularity of movies?

Based on the model selection's result, we found that the reduced linear regression model without transformation is the best model. Thus, the numeric factors revenue, vote count, and vote average affect the popularity of movies. In addition to the numeric factors, we found that the genres animation, thriller, and adventure had a significant impact on the popularity of movies. Other than genres, there are several production companies that also affect the popularity of movies, they are Jerry Bruckheimer Films, Lightstorm Entertainment, and Ingenious Film Partners.

3) What is the most significant numeric factor determining the popularity of movies?

According to the coefficient of the lasso regression model, the most significant numeric factor determining the popularity of movies is the vote count.

7. Conclusion

Based on the analysis of our data set, we reached the conclusion that budget, vote average, vote count, production companies, and language have a positive effect on a movie's popularity. Hence, if a producer wants to make a popular movie, an adequate budget, smart advertising and marketing, and a famous film company sponsor are beneficial. Moreover, considering the global audience, English is the most receptive.

Even though the Lasso regression gave us a good view of what factors affect the popularity of a movie, the regressors we got might not be a full image of the factors affecting the popularity. For example, the directors and the main actors of the movies sometimes have an impact on popularity. Hence, further research including more details of movies can reveal a better picture of factors affecting popularity.

8. Appendix Code

Part0: Overview

```
library(readr)
Movies.original <- read_csv("tmdb_5000_movies.csv",show_col_types = FALSE)
# head(Movies.original)

Movies.related <- subset(Movies.original, select = -c(homepage, id, overview, tagline, original_title, productio
n_countries, keywords, spoken_languages, status)) #Delete irrelevant columns

DuplicatedIndex <- duplicated(Movies.related) #Check for repeated information
num_Duplicate <- which(DuplicatedIndex) #No duplicated movie

Movies.organized <- data.frame(na.omit(Movies.related)) #Omit rows with NAs
```

Part1: Data Visualization

1.1 Categorical Features

1.1.1 Function of Counting Categories

```
library("stringr")
Count.Category <- function(category){
  count_name = c()
  count_nums = c()
  for (i in 1:nrow(Movies.organized)){
    category_split = unlist(str_split(category[i], "\"")) # variable that stores the category names
    for (k in 1:length(category_split)){
      if (category_split[k]==": "){
        category_name = category_split[k+1]
        if (category_name %in% count_name){# if the category name already exists, add one
          count_nums[which(count_name == category_name)] = count_nums[which(count_name == category_name)]+1
        }
        else{ # else, add the category name to the list of category
          count_name = c(count_name,category_name)
          count_nums[which(count_name == category_name)] = 1
        }
      }
    }
  }
  names(count_nums) <- count_name # assign corresponding category names to the numbers
  return(count_nums)
}
```

1.1.2 Distribution of Release Year

```
release_year <- as.numeric(substring(Movies.organized$release_date, 1,4))
release_year.sort <- sort(table(release_year))
release_year.result <- summary(Movies.organized$release_date)
hist(release_year, col = "lightblue", xlim = range(1916, 2017), ylim =range(0, 500), breaks = 40, ylab = 'number
 of movies', main ='Histogram of Release Year')
```

```
# consider movies from 2000 to the most current data
Movies.organized$release_year <- release_year
Movies.organized <- Movies.organized[which(Movies.organized$release_year==2009),]
```

1.1.3 Distribution of Split Genres

```
genre_count_nums <- Count.Category(Movies.organized$genres)
barplot(sort(genre_count_nums,decreasing=TRUE),las = 2,col="lightblue", main="Bar Plot of Genres",cex.names=0.7)
```

1.1.4 Distribution of Top 5 Movie Production Companies

```
company_count_nums <- Count.Category(Movies.organized$production_companies)
company_5 <- head(sort(company_count_nums,decreasing = T),5) # choose the top 5 companies that produce the large
st number of movies
pie(company_5,main="Pie Plot of Top 5 Movie Production Companies",cex=0.7)
```

1.1.5 Distribution of Languages

```
library(ggplot2)
ggplot(Movies.organized,aes(x=original_language, y=popularity, color=original_language))+
  ggtitle("Scatter Plot of Languages")+
  geom_point()+
  theme(legend.title=element_text(size = 9),legend.text=element_text(size = 9))
```

1.2 Correlation

1.2.1 Scatter Plot Matrix

```
pairs(~popularity+budget+revenue+runtime+vote_average+vote_count, data = Movies.organized, main = "Scatter Plot
 Matrix", panel = panel.smooth)
```

1.2.2 Correlation Table

```
res <- cor(data.frame(subset(Movies.organized, select = c(budget, revenue, runtime, vote_average, vote_count))))
```

Part2: Data Processing

2.1 Processing Missing Values

2.1.1 Numeric Variables

```
check_zeros <- function(check_col){
  num_zeros <- length(which(check_col == 0))
  percentage_zeros <- num_zeros/nrow(check_col)
  return(c(num_zeros,percentage_zeros))
}
```

```
num_cols <- which(unlist(lapply(Movies.organized, is.numeric)))
zeros_nums <- list()
zeros_ratio <- list()
for(i in 1:length(num_cols)){
  check_col <- Movies.organized[,num_cols[i]]
  zeros_nums[i] <- check_zeros(check_col)[1]
  zeros_ratio[i] <- check_zeros(check_col)[2]
}
names(zeros_nums) = names(num_cols)
zeroResult <- cbind(zeros_nums,zeros_ratio)
```

```
Movies.organized <- Movies.organized[which(!(Movies.organized$budget==0)),]
Movies.organized <- Movies.organized[which(!(Movies.organized$revenue==0)),]
Movies.organized <- Movies.organized[which(!(Movies.organized$runtime==0)),]
Movies.organized <- Movies.organized[which(!(Movies.organized$vote_average==0)),]
Movies.organized <- Movies.organized[which(!(Movies.organized$vote_count==0)),]
Movies.organized.result <- summary(Movies.organized)
```

2.1.2 Categorical Variables

```
Movies.organized <- Movies.organized[which(!(Movies.organized$genres=='[]')),]
Movies.organized <- Movies.organized[which(!(Movies.organized$production_companies=='[]')),]
```

2.2 Dummy Variables

```
split.category = list()
spe_category <- function(category){
  for(i in 1:nrow(Movies.organized)){
    category_split = unlist(str_split(category[i], "\""))
    spe_list = c()
    for(k in 1:length(category_split)){
      if(category_split[k] == ": "){
        spe_list = c(spe_list,category_split[k+1])
      }
    }
    split.category[[i]] = spe_list
  }
  return(split.category)
}
```

2.2.1 Genre

```
Movies <- data.frame(subset(Movies.organized, select = c(popularity,budget, revenue, runtime,vote_average, vote_
count))) # create new data set with only numeric variables
```

```
genre_count_nums <- Count.Category(Movies.organized$genres)
genre_count.sort <- sort(genre_count_nums,decreasing=TRUE)
genreNames.factor <- as.factor(names(genre_count_nums))

split.genres <- spe_category(Movies.organized$genres)
Movies.organized$genres <- split.genres # replace genres with a more readable form
```

```
Movies$dDrama <- 0 #(Dummy Drama)
for (i in 1:length(Movies.organized$genres)){
  if("Drama" %in% unlist(Movies.organized$genres[i])){
    Movies$dDrama[i] <- 1
  }
}

Movies$dComedy <- 0 #(Dummy Comedy)
for (i in 1:length(Movies.organized$genres)){
  if("Comedy" %in% unlist(Movies.organized$genres[i])){
    Movies$dComedy[i] <- 1
  }
}

Movies$dThriller <- 0 #(Dummy Thriller)
for (i in 1:length(Movies.organized$genres)){
  if("Thriller" %in% unlist(Movies.organized$genres[i])){
    Movies$dThriller[i] <- 1
  }
}

Movies$dAction <- 0 #(Dummy Action)
for (i in 1:length(Movies.organized$genres)){
  if("Action" %in% unlist(Movies.organized$genres[i])){
    Movies$dAction[i] <- 1
  }
}

Movies$dAdventure <- 0 #(Dummy Adventure)
for (i in 1:length(Movies.organized$genres)){
  if("Adventure" %in% unlist(Movies.organized$genres[i])){
    Movies$dAdventure[i] <- 1
  }
}
```

2.2.2 Language

```
Movies$dEnglish <- 0 #(Dummy en)
Movies$dEnglish <- ifelse(Movies.organized$original_language == 'en', 1,0)
```

2.2.3 Company

```
company_count_nums <- Count.Category(Movies.organized$production_companies)
companyNames.factor <- as.factor(head(names(sort(company_count_nums,decreasing=TRUE)),5))

split.company <- spe_category(Movies.organized$production_companies)
Movies.organized$production_companies <- split.company # replace production company with a readable form
```

```
Movies$dRelativity <- 0 #(Dummy Relativity Media)
for (i in 1:length(Movies.organized$production_companies)){
  if("Relativity Media" %in% unlist(Movies.organized$production_companies[i])){
    Movies$dRelativity[i] <- 1
  }
}

Movies$dUniversal <- 0 #(Dummy Universal Pictures)
for (i in 1:length(Movies.organized$production_companies)){
  if("Universal Pictures" %in% unlist(Movies.organized$production_companies[i])){
    Movies$dUniversal[i] <- 1
  }
}
Movies$dWarner <- 0 #(Dummy Warner Bros.)
for (i in 1:length(Movies.organized$production_companies)){
  if("Warner Bros." %in% unlist(Movies.organized$production_companies[i])){
    Movies$dWarner[i] <- 1
  }
}

Movies$dColumbia <- 0 #(Dummy Columbia Pictures)
for (i in 1:length(Movies.organized$production_companies)){
  if("Columbia Pictures" %in% unlist(Movies.organized$production_companies[i])){
    Movies$dColumbia[i] <- 1
  }
}

Movies$dDune <- 0 #(Dummy Dune Entertainment)
for (i in 1:length(Movies.organized$production_companies)){
  if("Dune Entertainment" %in% unlist(Movies.organized$production_companies[i])){
    Movies$dDune[i] <- 1
  }
}
```

Part3: Regressions

3.1 Linear Regression

3.1.1 Full Model & Assumptions Checking

```
Full <- lm(popularity~.,data=Movies)
Full.result <- summary(Full)
par(mfrow=c(2,2))
plot(Full)
```

```
p_outlier <- rstandard(Full)[rstandard(Full) < -3 | rstandard(Full) > 3] # check for outliers that are three sta
ndard deviations away
outlier_index <- names(p_outlier)[2:length(p_outlier)]
potential_outlier <- data.frame(Movies.organized[outlier_index,])


Movies.t <- cbind(Movies.organized$title,Movies)
for(i in 1:nrow(Movies.t)){ # remove outliers
  if(Movies.t$`Movies.organized$title`[i] %in% potential_outlier$title){
    Movies.t <- Movies.t[-c(i),]
  }
}

Movies <- Movies.t[,2:ncol(Movies.t)]
Full <- lm(popularity~.,data=Movies) # full model after removing outliers
Full.result <- summary(Full)
par(mfrow=c(2,2))
plot(Full)
```

```
full.adjr <- Full.result$adj.r.squared
Full.anova <- anova(Full)
Full.SSE <- Full.anova[nrow(Full.anova),2]
n = nrow(Movies)
k = 16+2
Full.AIC = n*log(Full.SSE/n)+2*k
Full.BIC = n*log(Full.SSE/n)+k*log(n)
```

3.1.2 Reduced Model by T-test

```
# Backward Stepwise Selection
backward <- step(Full,direction='backward',scope=formula(Full),trace=0)
back.anova <- backward$anova
# Forward Stepwise Selection
Empty <- lm(popularity~1,data=Movies)
forward <- step(Empty,direction='forward',scope=formula(Full),trace=0)
forward.anova <- forward$anova
# Both Stepwise Selection
both <- step(Empty,direction='both',scope=formula(Full),trace=0)
bi.anova <- both$anova
# Reduced Models
Reduced <- lm(formula = popularity ~ vote_count + budget + runtime + vote_average + dThriller + dDune + dEnglish
 + dDrama + dColumbia, data = Movies)
par(mfrow=c(2,2))
Reduced.result <- summary(Reduced)
Reduced.adjr <- Reduced.result$adj.r.squared

Reduced.anova <- anova(Reduced)
Reduced.SSE = Reduced.anova[nrow(Reduced.anova),2]
n = nrow(Movies)
k = 9+2
Reduced.AIC = n*log(Reduced.SSE/n)+2*k
Reduced.BIC = n*log(Reduced.SSE/n)+k*log(n)
```

3.1.3 Distribution of Y & Transformation Decision

```
hist(Movies$popularity,breaks = 100,main="Histogram of Popularity")
```

```
# The distribution of Y is right-skewed. (square root, cube root, and log)
popularity.result <- summary(Movies$popularity)
# The range is 1.729 - 150.438.
```

3.1.4 Box-Cox Transformation

```
library(MASS)
bc <- boxcox(Full)
```

```
lambda <- bc$x[which.max(bc$y)] # find lambda

Movies.boxcox <- Movies # create new data set for box cox transformation
Movies.boxcox$popularity <- (Movies$popularity^lambda-1)/lambda
popularity_boxcox <- (Movies$popularity^lambda-1)/lambda

boxcox.Full <- lm(popularity~.,data=Movies.boxcox) # regression on full model

Full.boxcox.result <- summary(boxcox.Full)
Full.boxcox.adjr <- Full.boxcox.result$adj.r.squared
par(mfrow=c(2,2))
plot(boxcox.Full)
```

```
boxcox.full.anova <- anova(boxcox.Full)
Full.SSE.boxcox = boxcox.full.anova[nrow(boxcox.full.anova),2]
n = nrow(Movies)
k = 16+2
Full.AIC.boxcox = n*log(Full.SSE.boxcox/n)+2*k
Full.BIC.boxcox = n*log(Full.SSE.boxcox/n)+k*log(n)

Empty.boxcox <- lm(popularity~1,data=Movies.boxcox)
forward.boxcox <- step(Empty.boxcox,direction='forward',
                   scope=formula(boxcox.Full),trace=0)
forward.boxcox.anova <- forward.boxcox$anova

boxcox.Reduced <- lm(formula = popularity ~ vote_count + budget + runtime + vote_average + dThriller + dEnglish
 + dDrama + dDune + dColumbia, data = Movies.boxcox) #regression

boxcox.Reduced.result <- summary(boxcox.Reduced)
Reduced.boxcox.adjr <- boxcox.Reduced.result$adj.r.squared
par(mfrow=c(2,2))
plot(boxcox.Reduced)
```

```
boxcox.reduce.anova <- anova(boxcox.Reduced)
Reduced.SSE.boxcox = boxcox.reduce.anova[nrow(boxcox.reduce.anova),2]
n = nrow(Movies)
k = 9+2
Reduced.AIC.boxcox = n*log(Reduced.SSE.boxcox/n)+2*k
Reduced.BIC.boxcox = n*log(Reduced.SSE.boxcox/n)+k*log(n)
```

3.1.5 Log Transformation

```
# due to unequal variance, transformation of y is used.
Movies.log <- Movies
Movies.log$popularity <- log(Movies$popularity)
# full model with log
Full.log <- lm(popularity~.,data=Movies.log)
Full.log.result <- summary(Full.log)
Full.log.adjr <- Full.log.result$adj.r.squared
par(mfrow=c(2,2))
plot(Full.log)
```

```
log.full.anova <- anova(Full.log)
Full.SSE.log = log.full.anova[nrow(log.full.anova),2]
n = nrow(Movies)
k = 16+2
Full.AIC.log = n*log(Full.SSE.log/n)+2*k
Full.BIC.log = n*log(Full.SSE.log/n)+k*log(n)

# reduced model with log
Empty.log <- lm(popularity~1,data=Movies.log)
forward.log <- step(Empty.log,direction='forward',scope=formula(Full.log),trace=0)
forward.log.anova <- forward.log$anova

Reduced.log <- lm(formula = popularity ~ vote_count + dEnglish + budget + revenue + dThriller + runtime + vote_a
verage + dDrama + dColumbia + dComedy, data = Movies.log) # regression

Reduced.log.result <- summary(Reduced.log)
Reduced.log.adjr <- Reduced.log.result$adj.r.squared

log.reduce.anova <- anova(Reduced.log)
Reduced.SSE.log = log.reduce.anova[nrow(log.reduce.anova),2]
n = nrow(Movies)
k = 10+2
Reduced.AIC.log = n*log(Reduced.SSE.log/n)+2*k
Reduced.BIC.log = n*log(Reduced.SSE.log/n)+k*log(n)
```

3.1.6 Square Root Transformation

```
Movies.sqrt <- Movies
Movies.sqrt$popularity <- sqrt(Movies.sqrt$popularity)
# full model with square root
Full.sqrt <- lm(popularity~.,data=Movies.sqrt) # regression
Full.sqrt.result <- summary(Full.sqrt)
Full.sqrt.adjr <- Full.sqrt.result$adj.r.squared
par(mfrow=c(2,2))
plot(Full.sqrt)
```

```
sqrt.full.anova <- anova(Full.sqrt)
Full.SSE.sqrt = sqrt.full.anova[nrow(sqrt.full.anova),2]
n = nrow(Movies)
k = 16+2
Full.AIC.sqrt = n*log(Full.SSE.sqrt/n)+2*k
Full.BIC.sqrt = n*log(Full.SSE.sqrt/n)+k*log(n)

# reduced model with square root
Empty.sqrt <- lm(popularity~1,data=Movies.sqrt)
forward.sqrt <- step(Empty.sqrt,direction='forward',scope=formula(Full.sqrt),trace=0)
forward.sqrt.anova <- forward.sqrt$anova

Reduced.sqrt <- lm(formula = popularity ~ vote_count + budget + revenue + runtime + dComedy + vote_average + dEn
glish + dDrama + dColumbia + dDune + dThriller, data = Movies.sqrt) #regression

Reduced.sqrt.result <- summary(Reduced.sqrt)
Reduced.sqrt.adjr <- Reduced.sqrt.result$adj.r.squared
par(mfrow=c(2,2))
plot(Reduced.sqrt)
```

```
sqrt.reduce.anova <- anova(Reduced.sqrt)
Reduced.SSE.sqrt = sqrt.reduce.anova[nrow(sqrt.reduce.anova),2]
n = nrow(Movies)
k = 11+2
Reduced.AIC.sqrt = n*log(Reduced.SSE.sqrt/n)+2*k
Reduced.BIC.sqrt = n*log(Reduced.SSE.sqrt/n)+k*log(n)
```

3.1.7 Decision of Transformation (combined with 3.1.3)

```
library(rcompanion)
par(mfrow = c(1,3))
plotNormalHistogram(popularity_boxcox,main='Box-Cox Transformation')
plotNormalHistogram(Movies.log$popularity,main='Log Transformation')
plotNormalHistogram(Movies.sqrt$popularity,main='Square Root Transformation')
```

3.1.8 Decision of Linear Regression by AIC and BIC

```
linear.adjr2 = c(full.adjr,Reduced.adjr)
boxcox.adjr2 = c(Full.boxcox.adjr,Reduced.boxcox.adjr)
log.adjr2 = c(Full.log.adjr,Reduced.log.adjr)
sqrt.adjr2 = c(Full.sqrt.adjr,Reduced.sqrt.adjr)
r2 = data.frame(cbind(linear.adjr2,boxcox.adjr2,log.adjr2,sqrt.adjr2))
rownames(r2) = c('Full Model','Reduced Model')

linear.AIC = c(Full.AIC,Reduced.AIC)
boxcox.AIC = c(Full.AIC.boxcox,Reduced.AIC.boxcox)
log.AIC = c(Full.AIC.log,Reduced.AIC.log)
sqrt.AIC = c(Full.AIC.sqrt,Reduced.AIC.sqrt)
AIC = data.frame(cbind(linear.AIC,boxcox.AIC,log.AIC,sqrt.AIC))
rownames(AIC) = c('Full Model','Reduced Model')

linear.BIC = c(Full.BIC,Reduced.BIC)
boxcox.BIC = c(Full.BIC.boxcox,Reduced.BIC.boxcox)
log.BIC = c(Full.BIC.log,Reduced.BIC.log)
sqrt.BIC = c(Full.BIC.sqrt,Reduced.BIC.sqrt)
BIC = data.frame(cbind(linear.BIC,boxcox.BIC,log.BIC,sqrt.BIC))
rownames(BIC) = c('Full Model','Reduced Model')
```

3.2 LASSO Regression

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-3
```

```
Movies.Lasso <- subset(Movies, select = -c(popularity))
y <- Movies$popularity
x <- data.matrix(Movies.Lasso)

Movies.lasso <- cv.glmnet(x,y,alpha=1,standardize=TRUE)
lambda.best.l <- Movies.lasso$lambda.min
lasso.model.best <- glmnet(x,y,alpha=1,lambda=lambda.best.l,standardize=TRUE)
lasso.models <- glmnet(x,y,alpha=1)

par(mfrow = c(1,2))
plot(Movies.lasso)
plot(lasso.models,xvar="lambda")
```

```
par(mfrow = c(1,1))
```

3.3 Ridge Regression

```
Movies.ridge <- cv.glmnet(x,y,alpha=0)
lambda.best.r <- Movies.ridge$lambda.min
ridge.model.best <- glmnet(x,y,alpha=0,lambda=lambda.best.r)
ridge.models <- glmnet(x,y,alpha=0)

par(mfrow = c(1,2))
plot(Movies.ridge)
plot(ridge.models,xvar="lambda")
```

```
par(mfrow = c(1,1))
```

Part4: Model Selection

4.1 R-squared, MSE, AIC, and BIC

```
#Lasso
predicted.l <- predict(lasso.model.best,s=lambda.best.l,newx=x)
k <- lasso.model.best$df
n <- lasso.model.best$nobs
SSE <- sum((predicted.l - y)^2)
MSE <- SSE/(n-k)
SSTO <- sum((y - mean(y))^2)
R_squared <- 1-SSE/SSTO
Lasso.result <- data.frame(MSE=MSE,SSE=SSE,R_squared=R_squared)
#Ridge
predicted.r <- predict(ridge.model.best,s=lambda.best.r,newx=x)
k <- ridge.model.best$df
n <- ridge.model.best$nobs
SSE <- sum((predicted.r - y)^2)
MSE <- SSE/(n-k)
SSTO <- sum((y - mean(y))^2)
R_squared <- 1-SSE/SSTO
Ridge.result <- data.frame(MSE=MSE,SSE=SSE,R_squared=R_squared)
#Reduced Log Regression
Reduced.SSE.log = log.reduce.anova[nrow(log.reduce.anova),2]
Reduced.MSE.log = log.reduce.anova[nrow(log.reduce.anova),3]
# summary(Reduced)
Reduced.R2.log <- Reduced.log.result$r.squared
k <- 10+2
n <- 148
Log.result <- data.frame(MSE=Reduced.MSE.log,SSE=Reduced.SSE.log,
                         R_squared=R_squared)
```

4.2 Model Selection Conclusion

```
selection_table <- rbind(Lasso.result=Lasso.result,
      Ridge.result=Ridge.result,
      Log.result=Log.result)
```

Part5: Movie Recommendation Function

```
Movies_recommendation <- function(){
  print(c('Here is the list of genres:',names(genre_count_nums)))
  Genre = readline(prompt="Enter Preferred Genre: ")
  recommend.index = c()
  for (i in 1:nrow(Movies.organized)){
    if (Genre %in% unlist(Movies.organized$genres[i])){
      recommend.index = c(recommend.index,i)
    }
  }
  Movies.recommend = data.frame(Movies.organized[recommend.index,])
  Movies.recommend = Movies.recommend[order(-Movies.recommend$popularity),]
  Movie = head(Movies.recommend$title,3)
  print(c("The most popular three movies are:",Movie))
}
```

The function is used to make a recommendation of a given genre. The function will first show a list of all genres, and then ask the user to type in one preferred genre. The function will finally return the three most popular movie's titles under this genre.