**Appendix**

**Descriptive Analysis & Data Visualization Tools**

```{r}

df <- read\_excel("~/Desktop/tmdb\_5000\_movies.xlsx")

View(df)

str(df)

plot(df$revenue ~ df$budget, xlab="budget", ylab="revenue")

rev.bud <- lm(formula = revenue ~ budget, data=df)

summary.lm(rev.bud)

plot(df$vote\_average ~ df$budget, xlab="vote\_average", ylab="revenue")

rev.votavg <- lm(formula = revenue ~ vote\_average, data=df)

summary.lm(rev.votavg)

plot(df$revenue ~ df$popularity, xlab="popularity", ylab="revenue")

rev.pop <- lm(formula = revenue ~ popularity, data=df)

summary.lm(rev.pop)

```

```{r}

#remove columns

MovieData <- df

MovieData = subset(MovieData, select = -c(17:33) )

ggplot(MovieData, aes(x=popularity)) +

geom\_histogram(binwidth=20, color="white", fill="black") +

ylab("Popularity") + xlab("Revenue") + xlim(-10,150) + ggtitle("Revenue vs. Popularity")

```

```{r}

NewMovieData <- data.frame(MovieData$title,MovieData$popularity,MovieData$vote\_count,MovieData$revenue)

str(NewMovieData)

ggplot(NewMovieData,aes(x=MovieData.title, y=MovieData.revenue)) + geom\_point(aes(size=MovieData.popularity, color=MovieData.vote\_count)) + ggtitle("Popularity According to the Viewers")

**Linear Regression and Multicollinearity Test**

```{r}

tmdb\_5000\_movies <- read\_xlsx("C:/Users/stena/Desktop/tmdb\_5000\_movies.xlsx")

head(tmdb\_5000\_movies)

str(tmdb\_5000\_movies)

summary(tmdb\_5000\_movies)

str(tmdb\_5000\_movies)

## Step 2 remove columns 18 through 34

tmdb\_5000\_movies <- tmdb\_5000\_movies[,-18:-34]

str(tmdb\_5000\_movies)

summary(tmdb\_5000\_movies)

tmdb\_5000\_movies[,12]

## Step 3 clean up function (Tim)

LoadMovies <- function(){

  Movies <- tmdb\_5000\_movies

  Movies <- Movies[,1:17]

  Movies$release\_date <- as.Date(Movies$release\_date,origin="1899-12-30")

  Movies$release\_month <- format(Movies$release\_date,"%b")

  Movies$release\_year <- format(Movies$release\_date, "%Y")

  Movies <-Movies[!(Movies$budget==0 | Movies$popularity==0 | Movies$revenue==0 | Movies$runtime==0 | Movies$vote\_average==0 | Movies$vote\_count==0),]

  Movies <- Movies[complete.cases(Movies[,c(3,6:8,11,12)]),]

  Movies$budget <- ifelse(Movies$budget<200,Movies$budget\*1000000,Movies$budget)

  Movies$revenue <- ifelse(Movies$revenue<200,Movies$revenue\*1000000,Movies$revenue)

  Percent\_Budget <- (Movies$revenue/Movies$budget)\*100

  cbind(Movies,Percent\_Budget)

  return(Movies)

}

## Step 4 store as df

LoadMoviesdf <- LoadMovies()

str(LoadMoviesdf)

summary(LoadMoviesdf)

##Net Revenue

Net\_Revenue <- LoadMoviesdf$revenue - LoadMoviesdf$budget

Net\_Revenue

#Which movie generated a profit(yes or no)

Profit\_Loss <- ifelse(tmdb\_5000\_movies$revenue > tmdb\_5000\_movies$budget, "yes", "no")

Profit\_Loss

## Movie ROI (Revenue - cost)/cost

MovieROI <- (LoadMoviesdf$revenue - LoadMoviesdf$budget)/LoadMoviesdf$budget

MovieROI

##create linear model

model <- lm(formula = revenue ~ budget + vote\_average + vote\_count + runtime + popularity, data=LoadMoviesdf)

summary(model)

##re run model linear model removing independent variables that aren't statistically significant (vote\_average and runtime)

## these indepedent variables are highly statistically significant close to 100%, we

## reject the null hypothesis, which allows us to conclude that there is a relationship

## revenue and budget, vote\_count and popularity where these 3 independent variables are drives of revenue.

## However, only indicates an R squared of 5694 which infers that changes in budget, vote\_count and popularity can explain

## only 56.94 of the change in revenues, which suggests that there are other drivers not reflected in the data that would impact movie revenues.

model2 <- lm(formula = revenue ~ budget + vote\_count + popularity, data=LoadMoviesdf)

summary(model2)

##re run model linear model removing independent variables that aren't statistically significant (vote\_average and runtime)

model3 <- lm(formula = revenue ~ budget + popularity, data=LoadMoviesdf)

summary(model3)

##re run model linear model removing independent variables that aren't statistically significant (vote\_average and runtime)

model4 <- lm(formula = revenue ~ budget + vote\_count, data=LoadMoviesdf)

summary(model4)

##re run model linear model removing independent variables that aren't statistically significant (vote\_average and runtime)

model5 <- lm(formula = revenue ~ budget, data=LoadMoviesdf)

summary(model5)

## to review if multicolinearity exists .  No evidence or issue with multicolinearity because variance inflation factor is lower than 10, so very low.

library(faraway)

vif(model)

## Conclude that model2 is the best model

model2 <- lm(formula = revenue ~ budget + vote\_count + popularity, data=LoadMoviesdf)

summary(model2)

##Plot revenue versus budget

plot(LoadMoviesdf$budget, LoadMoviesdf$revenue)

model5 <- lm(formula = revenue ~ budget, data=LoadMoviesdf)

summary(model5)

#Fit

abline(model5)

library(ggplot2)

g <- ggplot(LoadMoviesdf, aes(x = budget, y = revenue)) + geom\_point() + stat\_smooth(method = "lm", col = "red") + ggtitle("Revenue versus Budget")

##Plot lm - ggplot

ggplot(LoadMoviesdf, aes(x = budget, y = revenue)) + geom\_point() + stat\_smooth(method = "lm", col = "red") + ggtitle("Revnue vs. Budget")

##Plot revenue versus vote\_count.  Vote count has the strongest relationship with

## revenue.

plot(LoadMoviesdf$vote\_count, LoadMoviesdf$revenue)

model6 <- lm(formula = revenue ~ vote\_count, data=LoadMoviesdf)

summary(model6)

#Fit

abline(model6)

g2 <- ggplot(LoadMoviesdf, aes(x = vote\_count, y = revenue)) + geom\_point() + stat\_smooth(method = "lm", col = "red") + ggtitle("Revenue versus Vote Count")

##Plot revenue versus popularity

plot(LoadMoviesdf$popularity, LoadMoviesdf$revenue)

model7 <- lm(formula = revenue ~ popularity, data=LoadMoviesdf)

summary(model7)

#Fit

abline(model7)

g3 <- ggplot(LoadMoviesdf, aes(x = popularity, y = revenue)) + geom\_point() + stat\_smooth(method = "lm", col = "red") + ggtitle("Revenue versus Popularity")

library(gridExtra)

grid.arrange(g, g2, g3, nrow=2, ncol=2)

str(LoadMoviesdf)

## Added columns Net\_Revenue and ROI

LoadMoviesdfNew <- cbind(LoadMoviesdf, Net\_Revenue, MovieROI)

LoadMoviesdfNew

str(LoadMoviesdfNew)

summary(LoadMoviesdfNew)

##Movie with the highest ROI - Paranomal Activity

LoadMoviesdfNew[which.max(LoadMoviesdfNew$MovieROI), 1]

max(LoadMoviesdfNew$MovieROI)

##Movie with the Lowest ROI - "The Adventurer: The Curse of the Midas Box"

LoadMoviesdfNew[which.min(LoadMoviesdfNew$MovieROI), 1]

min(LoadMoviesdfNew$MovieROI)

##Movie with the highest Revenue - Avatar - $2,787,965,087

LoadMoviesdfNew[which.max(LoadMoviesdfNew$revenue), 1]

max(LoadMoviesdfNew$revenue)

##Movie with the highest Revenue - Avatar

LoadMoviesdfNew[which.max(LoadMoviesdfNew$revenue-LoadMoviesdf$budget), 1]

#Movie with the highest  Net\_Revenue - Avatar, $ 2,550,965,087

LoadMoviesdfNew[which.max(LoadMoviesdfNew$Net\_Revenue), 1]

max(LoadMoviesdfNew$revenue-LoadMoviesdfNew$budget)

max(LoadMoviesdfNew$Net\_Revenue)

##Movie with the lowest  Net\_Revenue - "The Lone Ranger", $ -165,710,090

LoadMoviesdfNew[which.min(LoadMoviesdfNew$Net\_Revenue), 1]

min(LoadMoviesdfNew$revenue-LoadMoviesdfNew$budget)

min(LoadMoviesdfNew$Net\_Revenue)

##Movie with the Lowest Revenue - "I Married a Strange Person", $203

LoadMoviesdfNew[which.min(LoadMoviesdfNew$revenue), 1]

min(LoadMoviesdfNew$revenue)

#Movie with the highest  Net\_Revenue - Avatar, $ 2,550,965,087

LoadMoviesdfNew[which.max(LoadMoviesdfNew$Net\_Revenue), 1]

max(LoadMoviesdfNew$Net\_Revenue)

##Movie with the Lowest Net\_Revenue - "The Loan Ranger", Net loss of -$165,70,090

LoadMoviesdfNew[which.min(LoadMoviesdfNew$Net\_Revenue), 1]

min(LoadMoviesdfNew$Net\_Revenue)

##Movie with the highest budget - "Pirates of the Caribbean: On Stranger Tides", $3.8e+08

LoadMoviesdfNew[which.max(LoadMoviesdfNew$budget), 1]

max(LoadMoviesdfNew$budget)

##Movie with the lowest budget - "I Married A Stranger", $250

LoadMoviesdfNew[which.min(LoadMoviesdfNew$budget), 1]

min(LoadMoviesdfNew$budget)

##Order by Revnue

OrderedLoadMoviesdfNew2 <- LoadMoviesdfNew[order(LoadMoviesdfNew$revenue),]

head(OrderedLoadMoviesdfNew2)

tail(OrderedLoadMoviesdfNew2)

##Order by ROI

OrderedLoadMoviesdfNew <- LoadMoviesdfNew[order(LoadMoviesdfNew$MovieROI),]

head(OrderedLoadMoviesdfNew)

tail(OrderedLoadMoviesdfNew)

##Order by Budget

OrderedLoadMoviesdfNew3 <- LoadMoviesdfNew[order(LoadMoviesdfNew$MovieROI),]

head(OrderedLoadMoviesdfNew3)

tail(OrderedLoadMoviesdfNew3)

##Order by Net\_Revenue

OrderedLoadMoviesdfNew4 <- LoadMoviesdfNew[order(LoadMoviesdfNew$Net\_Revenue),]

head(OrderedLoadMoviesdfNew4)

tail(OrderedLoadMoviesdfNew4)

##This model is not a good model.  Though low p-value indicating statistical signifcance showing that changes in revenue and budget

# are strong predictors of a change in ROI, however only explains .2% of the change in ROI. This suggest that there are

#other external factors exist which are not reflected in the data.

model8 <- lm(formula = MovieROI ~ revenue + budget, data=LoadMoviesdfNew)

summary(model8)

summary(LoadMoviesdfNew)

str(LoadMoviesdfNew)

##Plot Revenue versus ROI

g4 <- ggplot(LoadMoviesdfNew, aes(x = revenue, y = MovieROI)) + geom\_line() + ggtitle("Revenue versus ROI")

library(ggplot2)

g4

```

**KSVM Machine Learning Algorithm**

library(jsonlite)

library(reshape2)

library(moments)

library(scales)

library(tidyverse)

library(sqldf)

library(readxl)

library(kernlab)

library(e1071)

library(ggplot2)

credits <- read\_csv("Downloads/tmdb\_5000\_credits.csv",col\_names=TRUE,na="NA")

crew <- credits %>%

  filter(nchar(crew)>2) %>%

  mutate(

    js  = lapply(crew, fromJSON)

  )  %>%

  unnest(js)

crew <- crew[,c(2,9,10)]

crew <- sqldf("select title, name from crew where job='Director'group by title")

colnames(crew) <- c("title","director")

cast <- credits %>%

  filter(nchar(cast)>2) %>%

  mutate(js = lapply(cast, fromJSON)) %>%

  unnest(js) %>%

  select(-cast, -crew, -credit\_id) %>%

  rename(actor=name, movie\_cast\_id=cast\_id, actor\_id=id) %>%

  mutate\_if(is.character, factor)

cast <- cast %>% filter(order %in% c(0, 1, 2)) %>% select(movie\_id, title, order, actor)

cast$order[1] <- 0

for (i in 1:(nrow(cast)-1)){

  if(cast$movie\_id[i+1]!=cast$movie\_id[i]){

    cast$order[i+1] <- 0

  } else {cast$order[i+1] <- cast$order[i]+1}

}

cast <- cast %>% filter(order %in% c(0, 1, 2)) %>%

  spread(key=order, value=actor)

cast <- cast[,c(2:5)]

colnames(cast) <- c("title","actor\_1","actor\_2","actor\_3")

Cast\_Crew <- merge(cast,crew,by="title",all=T)

tmdb\_5000\_movies <- read\_excel("Documents/tmdb\_5000\_movies.xls")

Movies <- tmdb\_5000\_movies

Movies <- Movies[,1:17]

Movies$release\_date <- as.Date(Movies$release\_date,origin="1899-12-30")

Movies$release\_month <- months(Movies$release\_date)

Movies$release\_year <- format(Movies$release\_date, "%Y")

Movies <-Movies[!(Movies$budget==0 | Movies$popularity==0 | Movies$revenue==0 | Movies$runtime==0 | Movies$vote\_average==0 | Movies$vote\_count==0),]

Movies <- Movies[complete.cases(Movies[,c(3,6:8,11,12)]),]

Movies$budget <- ifelse(Movies$budget<200,Movies$budget\*1000000,Movies$budget)

Movies$revenue <- ifelse(Movies$revenue<200,Movies$revenue\*1000000,Movies$revenue)

Movies$Percent\_Budget <- (Movies$revenue/Movies$budget)\*100

Movies$Difference <- Movies$revenue-Movies$budget

Movies <- merge(Movies,Cast\_Crew,by="title",all=T)

Movies <- Movies[complete.cases(Movies[,2]),]

Summary\_Stats <- function(column){

Mean <- mean(column)

Median <- median(column)

SD <- sd(column)

Max <- max(column)

Min <- min(column)

Skewness <- skewness(column)

Vector <- c(Mean,Median,SD,Max,Min,Skewness)

return(Vector)

}

Vote\_Average <- Summary\_Stats(Movies$vote\_average)

Budget <- Summary\_Stats(Movies$budget)

Revenue <- Summary\_Stats(Movies$revenue)

Runtime <- Summary\_Stats(Movies$runtime)

Popularity <- Summary\_Stats(Movies$popularity)

Percent\_Budget <- Summary\_Stats(Movies$Percent\_Budget)

Difference <- Summary\_Stats(Movies$Difference)

Labels <- c("Mean","Median","SD","Max","Min","Skew")

Summary\_Statistics <- data.frame(Labels,Vote\_Average,Budget,Revenue,Runtime,Popularity,Percent\_Budget,Difference)

GenreMeans <- function(column){

  Genre1Avg <- aggregate(x=column,list(Movies$genre1),mean)

  Genre2Avg <- aggregate(x=column,list(Movies$genre2),mean)

  Genre3Avg <- aggregate(x=column,list(Movies$genre3),mean)

  GenreAvg <- merge(Genre1Avg,Genre2Avg, by="Group.1",all =TRUE, no.dups = T)

  GenreAvg <- merge(GenreAvg,Genre3Avg, by="Group.1",all =TRUE, no.dups = T)

  Average <- rowMeans(cbind(GenreAvg$x.x,GenreAvg$x.y,GenreAvg$x),na.rm=TRUE)

  GenreAvg <- cbind(GenreAvg,Average)

  GenreAvg <- GenreAvg[,c(1,5)]

  colnames(GenreAvg)[colnames(GenreAvg)=="Group.1"] <- "Genre"

  return(GenreAvg)

}

Genre\_VA\_Mean <-GenreMeans(Movies$vote\_average)

Genre\_Budget\_Mean <- GenreMeans(Movies$budget)

Genre\_Revenue\_Mean <- GenreMeans(Movies$revenue)

Genre\_Pop\_Mean <- GenreMeans(Movies$popularity)

Genre\_Run\_Mean <- GenreMeans(Movies$runtime)

Genre\_PB\_Mean <- GenreMeans(Movies$Percent\_Budget)

Genre\_Dif\_Mean <- GenreMeans(Movies$Difference)

GenreMedians <- function(column){

  Genre1Med <- aggregate(x=column,list(Movies$genre1),median)

  Genre2Med <- aggregate(x=column,list(Movies$genre2),median)

  Genre3Med <- aggregate(x=column,list(Movies$genre3),median)

  GenreMed <- merge(Genre1Med,Genre2Med, by="Group.1",all =TRUE, no.dups = T)

  GenreMed <- merge(GenreMed,Genre3Med, by="Group.1",all =TRUE, no.dups = T)

  Median <- rowMeans(cbind(GenreMed$x.x,GenreMed$x.y,GenreMed$x),na.rm=TRUE)

  GenreMed <- cbind(GenreMed,Median)

  GenreMed <- GenreMed[,c(1,5)]

  colnames(GenreMed)[colnames(GenreMed)=="Group.1"] <- "Genre"

  return(GenreMed)

}

Genre\_VA\_Median <-GenreMedians(Movies$vote\_average)

Genre\_Budget\_Median <- GenreMedians(Movies$budget)

Genre\_Revenue\_Median <- GenreMedians(Movies$revenue)

Genre\_Pop\_Median <- GenreMedians(Movies$popularity)

Genre\_Run\_Median <- GenreMedians(Movies$runtime)

Genre\_PB\_Median <- GenreMedians(Movies$Percent\_Budget)

Genre\_Dif\_Median <- GenreMedians(Movies$Difference)

MergeGenre <- function(df){

  G1 <- merge(Genre\_VA\_Mean,Genre\_VA\_Median,by="Genre")

  G2 <- merge(G1,Genre\_Budget\_Mean, by ="Genre")

  G3 <- merge(G2,Genre\_Budget\_Median, by ="Genre")

  G4 <- merge(G3,Genre\_Revenue\_Mean, by ="Genre")

  G5 <- merge(G4,Genre\_Revenue\_Median, by ="Genre")

  G6 <- merge(G5,Genre\_Pop\_Mean, by ="Genre")

  G7 <- merge(G6,Genre\_Pop\_Median, by ="Genre")

  G8 <- merge(G7,Genre\_Run\_Mean, by ="Genre")

  G9 <- merge(G8,Genre\_Run\_Median, by ="Genre")

  G10 <- merge(G9,Genre\_PB\_Mean, by ="Genre")

  G11 <- merge(G10,Genre\_PB\_Median, by ="Genre")

  G12 <- merge(G11,Genre\_Dif\_Mean, by ="Genre")

  Genre\_Summary <- merge(G12,df, by ="Genre")

  colnames(Genre\_Summary) <- c("Genre","VA\_mean","VA\_med","Budget\_mean","Budget\_med","Revenue\_mean","Revenue\_med","Pop\_mean","Pop\_med","Run\_mean","Run\_med","PB\_mean","PB\_med","Dif\_mean","Dif\_med")

  return(Genre\_Summary)

}

Genre\_Summary <- MergeGenre(Genre\_Dif\_Median)

Genre\_Summary$Rev\_Score <- scale(Genre\_Summary$Revenue\_mean)

Genre\_Summary$VA\_Score <- scale(Genre\_Summary$VA\_mean)

Genre\_Summary$Pop\_Score <- scale(Genre\_Summary$Pop\_mean)

Genre\_Summary$ROI\_Score <- scale(Genre\_Summary$PB\_med)

Genre\_Summary$Total\_Score <- rowSums(Genre\_Summary[,c(16:19)])

GenreScoreplot <- ggplot(Genre\_Summary,aes(x=Rev\_Score,y=VA\_Score))+geom\_point(aes(color=ROI\_Score,size=Pop\_Score))+geom\_text(aes(label=Genre),size=2,color="red")+ggtitle("Genre Scores")

MonthMeans <- function(column){

  MonthAvg <- aggregate(x=column,list(Movies$release\_month),mean)

  colnames(MonthAvg)[colnames(MonthAvg)=="Group.1"] <- "Month"

  colnames(MonthAvg)[colnames(MonthAvg)=="x"] <- "Average"

  return(MonthAvg)

}

Month\_VA\_Mean <-MonthMeans(Movies$vote\_average)

Month\_Budget\_Mean <- MonthMeans(Movies$budget)

Month\_Revenue\_Mean <- MonthMeans(Movies$revenue)

Month\_Pop\_Mean <- MonthMeans(Movies$popularity)

Month\_Run\_Mean <- MonthMeans(Movies$runtime)

Month\_PB\_Mean <- MonthMeans(Movies$Percent\_Budget)

Month\_Dif\_Mean <- MonthMeans(Movies$Difference)

MonthMedians <- function(column){

  MonthMed <- aggregate(x=column,list(Movies$release\_month),median)

  colnames(MonthMed)[colnames(MonthMed)=="Group.1"] <- "Month"

  colnames(MonthMed)[colnames(MonthMed)=="x"] <- "Median"

  return(MonthMed)

}

Month\_VA\_Median <-MonthMedians(Movies$vote\_average)

Month\_Budget\_Median <- MonthMedians(Movies$budget)

Month\_Revenue\_Median <- MonthMedians(Movies$revenue)

Month\_Pop\_Median <- MonthMedians(Movies$popularity)

Month\_Run\_Median <- MonthMedians(Movies$runtime)

Month\_PB\_Median <- MonthMedians(Movies$Percent\_Budget)

Month\_Dif\_Median <- MonthMedians(Movies$Difference)

MergeMonth <- function(df){

  G1 <- merge(Month\_VA\_Mean,Month\_VA\_Median,by="Month")

  G2 <- merge(G1,Month\_Budget\_Mean, by ="Month")

  G3 <- merge(G2,Month\_Budget\_Median, by ="Month")

  G4 <- merge(G3,Month\_Revenue\_Mean, by ="Month")

  G5 <- merge(G4,Month\_Revenue\_Median, by ="Month")

  G6 <- merge(G5,Month\_Pop\_Mean, by ="Month")

  G7 <- merge(G6,Month\_Pop\_Median, by ="Month")

  G8 <- merge(G7,Month\_Run\_Mean, by ="Month")

  G9 <- merge(G8,Month\_Run\_Median, by ="Month")

  G10 <- merge(G9,Month\_PB\_Mean, by ="Month")

  G11 <- merge(G10,Month\_PB\_Median, by ="Month")

  G12 <- merge(G11,Month\_Dif\_Mean, by ="Month")

  Month\_Summary <- merge(G12,df, by ="Month")

  colnames(Month\_Summary) <- c("Month","VA\_mean","VA\_med","Budget\_mean","Budget\_med","Revenue\_mean","Revenue\_med","Pop\_mean","Pop\_med","Run\_mean","Run\_med","PB\_mean","PB\_med","Dif\_mean","Dif\_med")

  return(Month\_Summary)

}

Month\_Summary <- MergeMonth(Month\_Dif\_Median)

Month\_Summary$Rev\_Score <- scale(Month\_Summary$Revenue\_mean)

Month\_Summary$VA\_Score <- scale(Month\_Summary$VA\_mean)

Month\_Summary$Pop\_Score <- scale(Month\_Summary$Pop\_mean)

Month\_Summary$ROI\_Score <- scale(Month\_Summary$PB\_med)

Month\_Summary$Total\_Score <- rowSums(Month\_Summary[,c(16:19)])

MonthScoreplot <- ggplot(Month\_Summary,aes(x=Rev\_Score,y=VA\_Score))+geom\_point(aes(color=ROI\_Score,size=Pop\_Score))+geom\_text(aes(label=Month),size=2,color="red")+ggtitle("Month Scores")

YearMeans <- function(column){

  YearAvg <- aggregate(x=column,list(Movies$release\_year),mean)

  colnames(YearAvg)[colnames(YearAvg)=="Group.1"] <- "Year"

  colnames(YearAvg)[colnames(YearAvg)=="x"] <- "Average"

  return(YearAvg)

}

Year\_VA\_Mean <- YearMeans(Movies$vote\_average)

Year\_Budget\_Mean <- YearMeans(Movies$budget)

Year\_Revenue\_Mean <- YearMeans(Movies$revenue)

Year\_Pop\_Mean <- YearMeans(Movies$popularity)

Year\_Run\_Mean <- YearMeans(Movies$runtime)

Year\_PB\_Mean <- YearMeans(Movies$Percent\_Budget)

Year\_Dif\_Mean <- YearMeans(Movies$Difference)

YearMedians <- function(column){

  YearMed <- aggregate(x=column,list(Movies$release\_year),median)

  colnames(YearMed)[colnames(YearMed)=="Group.1"] <- "Year"

  colnames(YearMed)[colnames(YearMed)=="x"] <- "Median"

  return(YearMed)

}

Year\_VA\_Median <-YearMedians(Movies$vote\_average)

Year\_Budget\_Median <- YearMedians(Movies$budget)

Year\_Revenue\_Median <- YearMedians(Movies$revenue)

Year\_Pop\_Median <- YearMedians(Movies$popularity)

Year\_Run\_Median <- YearMedians(Movies$runtime)

Year\_PB\_Median <- YearMedians(Movies$Percent\_Budget)

Year\_Dif\_Median <- YearMedians(Movies$Difference)

ProdCompMeans <- function(column){

  PC1Avg <- aggregate(x=column,list(Movies$production\_company),mean)

  PC2Avg <- aggregate(x=column,list(Movies$production\_company2),mean)

  PC3Avg <- aggregate(x=column,list(Movies$production\_company3),mean)

  PCAvg <- merge(PC1Avg,PC2Avg, by="Group.1",all =TRUE, no.dups = T)

  PCAvg <- merge(PCAvg,PC3Avg, by="Group.1",all =TRUE, no.dups = T)

  Average <- rowMeans(cbind(PCAvg$x.x,PCAvg$x.y,PCAvg$x),na.rm=TRUE)

  PCAvg <- cbind(PCAvg,Average)

  PCAvg <- PCAvg[,c(1,5)]

  colnames(PCAvg)[colnames(PCAvg)=="Group.1"] <- "Production\_Company"

  return(PCAvg)

}

Company\_VA\_Mean <- ProdCompMeans(Movies$vote\_average)

Company\_Budget\_Mean <- ProdCompMeans(Movies$budget)

Company\_Revenue\_Mean <- ProdCompMeans(Movies$revenue)

Company\_Pop\_Mean <- ProdCompMeans(Movies$popularity)

Company\_Run\_Mean <- ProdCompMeans(Movies$runtime)

Company\_PB\_Mean <- ProdCompMeans(Movies$Percent\_Budget)

Company\_Dif\_Mean <- ProdCompMeans(Movies$Difference)

ProdCompMedians <- function(column){

  PC1Med <- aggregate(x=column,list(Movies$production\_company),median)

  PC2Med <- aggregate(x=column,list(Movies$production\_company2),median)

  PC3Med <- aggregate(x=column,list(Movies$production\_company3),median)

  PCMed <- merge(PC1Med,PC2Med, by="Group.1",all =TRUE, no.dups = T)

  PCMed <- merge(PCMed,PC3Med, by="Group.1",all =TRUE, no.dups = T)

  Median <- rowMeans(cbind(PCMed$x.x,PCMed$x.y,PCMed$x),na.rm=TRUE)

  PCMed <- cbind(PCMed,Median)

  PCMed <- PCMed[,c(1,5)]

  colnames(PCMed)[colnames(PCMed)=="Group.1"] <- "Production\_Company"

  return(PCMed)

}

Company\_VA\_Median <- ProdCompMedians(Movies$vote\_average)

Company\_Budget\_Median <- ProdCompMedians(Movies$budget)

Company\_Revenue\_Median <- ProdCompMedians(Movies$revenue)

Company\_Pop\_Median <- ProdCompMedians(Movies$popularity)

Company\_Run\_Median <- ProdCompMedians(Movies$runtime)

Company\_PB\_Median <- ProdCompMedians(Movies$Percent\_Budget)

Company\_Dif\_Median <- ProdCompMedians(Movies$Difference)

CountryMeans <- function(column){

  CountryAvg <- aggregate(x=column,list(Movies$production\_country),mean)

  colnames(CountryAvg)[colnames(CountryAvg)=="Group.1"] <- "Country"

  colnames(CountryAvg)[colnames(CountryAvg)=="x"] <- "Average"

  return(CountryAvg)

}

Country\_VA\_Mean <- CountryMeans(Movies$vote\_average)

Country\_Budget\_Mean <- CountryMeans(Movies$budget)

Country\_Revenue\_Mean <- CountryMeans(Movies$revenue)

Country\_Pop\_Mean <- CountryMeans(Movies$popularity)

Country\_Run\_Mean <- CountryMeans(Movies$runtime)

Country\_PB\_Mean <- CountryMeans(Movies$Percent\_Budget)

Country\_Dif\_Mean <- CountryMeans(Movies$Difference)

CountryMedians <- function(column){

  CountryMed <- aggregate(x=column,list(Movies$production\_country),median)

  colnames(CountryMed)[colnames(CountryMed)=="Group.1"] <- "Country"

  colnames(CountryMed)[colnames(CountryMed)=="x"] <- "Median"

  return(CountryMed)

}

Country\_VA\_Median <- CountryMedians(Movies$vote\_average)

Country\_Budget\_Median <- CountryMedians(Movies$budget)

Country\_Revenue\_Median <- CountryMedians(Movies$revenue)

Country\_Pop\_Median <- CountryMedians(Movies$popularity)

Country\_Run\_Median <- CountryMedians(Movies$runtime)

Country\_PB\_Median <- CountryMedians(Movies$Percent\_Budget)

Country\_Dif\_Median <- CountryMedians(Movies$Difference)

Movies$boxoffice <- ifelse(Movies$Difference<=0,"Bomb","Not Bomb")

length(which(Movies$boxoffice == "Bomb"))

length(which(Movies$boxoffice == "Bomb"))/length(Movies$title)

ActorSum <- function(column){

  A1Sum <- aggregate(x=column,list(Movies$actor\_1),sum)

  A2Sum <- aggregate(x=column,list(Movies$actor\_2),sum)

  A3Sum <- aggregate(x=column,list(Movies$actor\_3),sum)

  ASum <- merge(A1Sum,A2Sum, by="Group.1",all =TRUE, no.dups = T)

  ASum <- merge(ASum,A3Sum, by="Group.1",all =TRUE, no.dups = T)

  Sum <- rowSums(cbind(ASum$x.x,ASum$x.y,ASum$x),na.rm=TRUE)

  ASum <- cbind(ASum,Sum)

  ASum <- ASum[,c(1,5)]

  colnames(ASum)[colnames(ASum)=="Group.1"] <- "Actor"

  colnames(ASum)[colnames(ASum)=="Sum"] <- "Total\_Revenue"

  return(ASum)

}

ActorRevenue <- ActorSum(Movies$revenue)

ActorRevenue <- ActorRevenue[order(-ActorRevenue$Total\_Revenue),]

DirectorSum <- function(column){

  DSum <- aggregate(x=column,list(Movies$director),sum)

  colnames(DSum)[colnames(DSum)=="Group.1"] <- "Director"

  colnames(DSum)[colnames(DSum)=="x"] <- "Total\_Revenue"

  return(DSum)

}

DirectorRevenue <- DirectorSum(Movies$revenue)

DirectorRevenue <- DirectorRevenue[order(-DirectorRevenue$Total\_Revenue),]

randIndex <- sample(1:dim(Movies)[1])

cut\_point <- floor(2\*dim(Movies)[1]/3)

Moviestrain <- Movies[randIndex[1:cut\_point],]

Moviestest <- Movies[randIndex[(cut\_point+1):dim(Movies)[1]],]

KSVMmod <- ksvm(boxoffice~budget+revenue+release\_month+popularity+vote\_average+vote\_count+runtime,data=Moviestrain,C=25)

KSVM\_Predict <- predict(KSVMmod, Moviestest)

compTable <- data.frame(Moviestest$boxoffice,KSVM\_Predict)

T <- table(compTable)

(T[1,1]+T[2,2])/nrow(Moviestest)\*100

Moviestrain$boxnum <- as.factor(ifelse(Moviestrain$boxoffice=="Bomb",0,1))

Moviestest$boxnum <- as.factor(ifelse(Moviestest$boxoffice=="Bomb",0,1))

NBmod <- naiveBayes(boxnum~budget+release\_month+popularity+vote\_average+vote\_count+runtime,data=Moviestrain)

NB\_Predict <- predict(NBmod, Moviestest)

compTable2 <- data.frame(Moviestest$boxnum,NB\_Predict)

T2 <- table(compTable2)

(T2[1,1]+T2[2,2])/nrow(Moviestest)\*100

Moviestest$predict <- compTable[,2]

KSVMplot <- ggplot(Moviestest, aes(x=Difference,y=Percent\_Budget)) + geom\_point(aes(shape=boxoffice,color=predict))

KSVMplot <- KSVMplot + xlim(-2000000,20000000)+ylim(0,500)

KSVMplot <- KSVMplot + xlab("Profit") +ylab("ROI")+ggtitle("KSVM Prediction Plot")

Act <- as.data.frame(table(Movies$actor\_1))

Act <- Act[order(-Act$Freq),]

Act2 <- as.data.frame(table(Movies$actor\_2))

Act2 <- Act2[order(-Act2$Freq),]

Act3 <- as.data.frame(table(Movies$actor\_3))

Act3 <- Act3[order(-Act3$Freq),]

Actor\_Freq <- merge(Act,Act2,by = "Var1")

Actor\_Freq <- merge(Actor\_Freq,Act3,by = "Var1")

Actor\_Freq$Frequency <- Actor\_Freq$Freq.x+Actor\_Freq$Freq.y+Actor\_Freq$Freq

Actor\_Freq <- Actor\_Freq[,c(1,5)]

Actor\_Freq <- Actor\_Freq[which(Actor\_Freq$Frequency>=10),]

colnames(Actor\_Freq)<- c("actor","number\_movies")

Movies\_Actors <- filter(Movies,actor\_1 | actor\_2 | actor\_3 %in% Actor\_Freq$actor)

ActorScore <- function(col){

  A <- aggregate(col,list(Movies\_Actors$actor\_1),mean)

  B <- aggregate(col,list(Movies\_Actors$actor\_2),mean)

  C <- aggregate(col,list(Movies\_Actors$actor\_3),mean)

  D <- merge(A,B, by = "Group.1",all=TRUE)

  E <- merge(D,C, by="Group.1",all=TRUE)

  E$average <- rowMeans(E[2:4],na.rm =TRUE)

  E <- E[,c(1,5)]

  colnames(E) <- c("actor","average")

  E <- filter(E,actor%in%Actor\_Freq$actor)

  return(E)

}

ARev <- ActorScore(Movies\_Actors$revenue)

AVA <-ActorScore(Movies\_Actors$vote\_average)

APop <- ActorScore(Movies\_Actors$popularity)

AROI <- ActorScore(Movies\_Actors$Percent\_Budget)

Actor\_Freq <- Actor\_Freq[-125,]

TopActors <- data.frame(Actor\_Freq$actor,ARev$average,AVA$average,APop$average,AROI$average)

colnames(TopActors) <- c("Actor","Avg\_Rev","Avg\_VA","Avg\_Pop","Avg\_ROI")

TopActors$Rev\_Score <- scale(TopActors$Avg\_Rev)

TopActors$VA\_Score <- scale(TopActors$Avg\_VA)

TopActors$Pop\_Score <- scale(TopActors$Avg\_Pop)

TopActors$ROI\_Score <- scale(TopActors$Avg\_ROI)

TopActors$Actor\_Score <- rowSums(TopActors[,c(6:9)])

TopActors <- TopActors[order(-TopActors$Actor\_Score),]

Dir <- as.data.frame(table(Movies$director))

Dir <- Dir[order(-Dir$Freq),]

Dir <- Dir[which(Dir$Freq>=6),]

colnames(Dir)<- c("director","number\_movies")

Movies\_Director <- filter(Movies,director %in% Dir$director)

DirectorScore <- function(col){

A <- aggregate(col,list(Movies\_Director$director),mean)

colnames(A) <- c("director","average")

A <- filter(A,director%in%Dir$director)

return(A)

}

DRev <- DirectorScore(Movies\_Director$revenue)

DVA <- DirectorScore(Movies\_Director$vote\_average)

DPop <- DirectorScore(Movies\_Director$popularity)

DROI <- DirectorScore(Movies\_Director$Percent\_Budget)

TopDirectors <- data.frame(Dir$director,DRev$average,DVA$average,DPop$average,DROI$average)

colnames(TopDirectors) <- c("Director","Avg\_Rev","Avg\_VA","Avg\_Pop","Avg\_ROI")

TopDirectors$Rev\_Score <- scale(TopDirectors$Avg\_Rev)

TopDirectors$VA\_Score <- scale(TopDirectors$Avg\_VA)

TopDirectors$Pop\_Score <- scale(TopDirectors$Avg\_Pop)

TopDirectors$ROI\_Score <- scale(TopDirectors$Avg\_ROI)

TopDirectors$Director\_Score <- rowSums(TopDirectors[,c(6:9)])

TopDirectors <- TopDirectors[order(-TopDirectors$Director\_Score),]

TopActors1 <- head(TopActors[order(-TopActors$Avg\_Rev),],10)

Actplot <- ggplot(TopActors1, aes(Actor,Avg\_Rev))

ActRevplot <- Actplot + geom\_col() + theme(axis.text.x = element\_text(angle=70,hjust =1),axis.text =element\_text(size=7),axis.title =element\_text(size = 8))

fx <- function(x){

  x/1000000

}

ActorScoreplot <- ggplot(TopActors,aes(x=Rev\_Score,y=VA\_Score))+geom\_point(aes(color=ROI\_Score,size=Pop\_Score))+geom\_text(aes(label=ifelse(Rev\_Score>4|VA\_Score<(-4)|ROI\_Score>4|Pop\_Score>5,as.character(Actor),"")))

TopActors20 <- head(TopActors,20)

TopDirectors20 <-head(TopDirectors, 20)

ActorScoreplot20 <- ggplot(TopActors20,aes(x=Rev\_Score,y=VA\_Score))+geom\_point(aes(color=ROI\_Score,size=Pop\_Score))+geom\_text(aes(label=Actor),size=2,color="red")+ggtitle("Top 20 Actor Scores")

DirectorScoreplot20 <- ggplot(TopDirectors20,aes(x=Rev\_Score,y=VA\_Score))+geom\_point(aes(color=ROI\_Score,size=Pop\_Score))+geom\_text(aes(label=Director),size=2,color="red")+ggtitle("Top 20 Director Scores")

```