Stat 418 - Final project

Yichen Zhou

Libraray Packages

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 3.5.3
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## 
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
## 
##     format.pval, units
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.5.3
```

```
## Loading required package: carData
```

```
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 3.5.3
```

Read data

```
data <- read.csv("films.csv",na.strings = "null")
data <- data[,c(4:7,9,10)]
```

Genre - select the first one as the genre

```
for (i in 1:length(data[,'Genre'])) {
  a <- as.vector.factor(data[i,'Genre'])
  b <- unlist(strsplit(a,split=","))
  data[i,'Genre'] <- b[1]
}
data$Genre[data$Genre=='Thriller'] = "Horror"
data$Genre[data$Genre=='Sport'|data$Genre=='Crime'|data$Genre=='Western'] = "Action"
data$Genre[data$Genre=="Family"|data$Genre=="Musical"|data$Genre=='Music'|data$Genre=='History'] = "Dram
data$Genre[data$Genre=='Sci-Fi'|data$Genre=='Fantasy'|data$Genre=='War'|data$Genre=='Mystery'] = "Advent
data$Genre <- factor(data$Genre, levels = c("Action","Adventure","Animation","Biography","Comedy","Drama
```

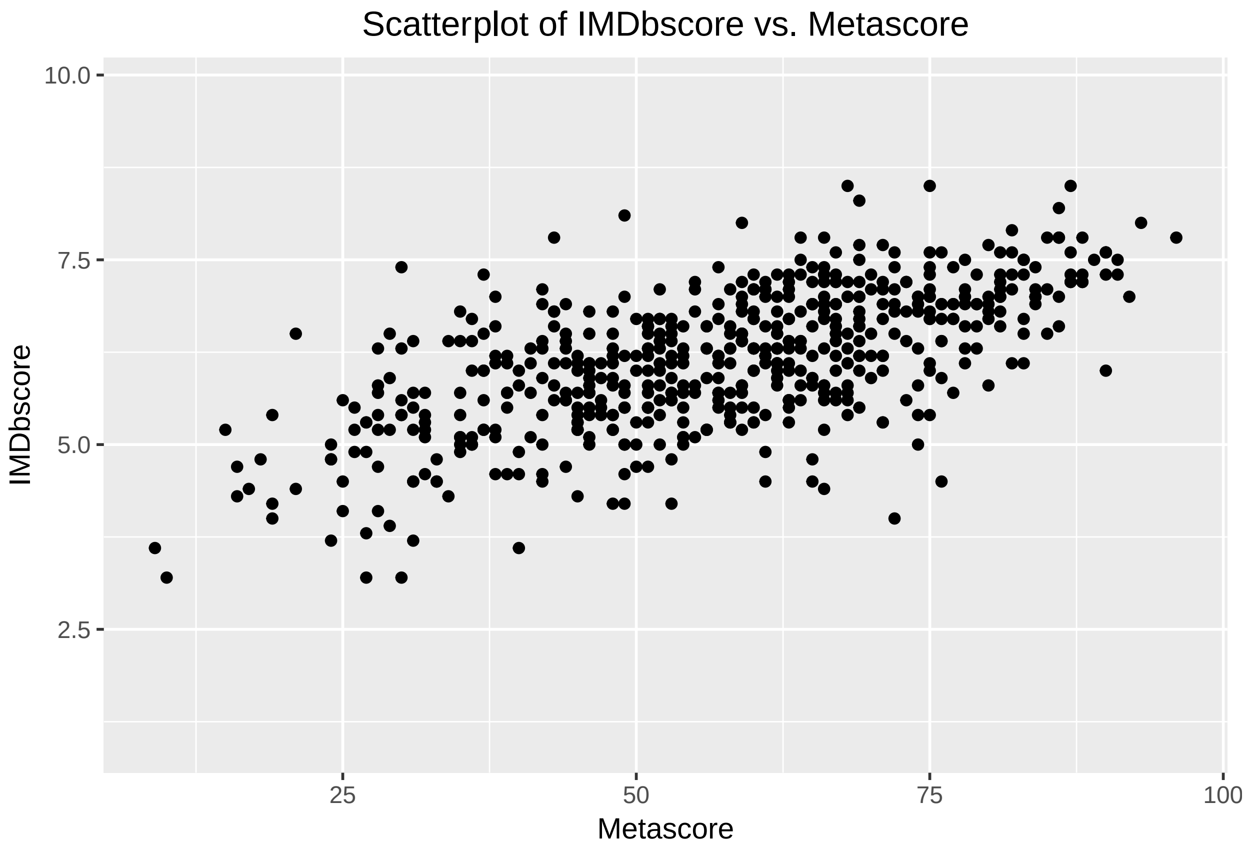
Run_time

```
Run_time <- data$Run_time
data$Run_time <- as.numeric(data$Run_time)
for (i in 1:length(data[,'Run_time'])) {
  a <- as.vector(Run_time[i])
  b <- unlist(strsplit(a,split=" "))
  data[i,'Run_time'] <- as.numeric(b[1])
}
```

Plot

```
ggplot(data,aes(Metascore,IMDbscore))+geom_point()+labs(title="Scatterplot of IMDbscore vs. Metascore")+
```

```
## Warning: Removed 4437 rows containing missing values (geom_point).
```



Fill in Missing value

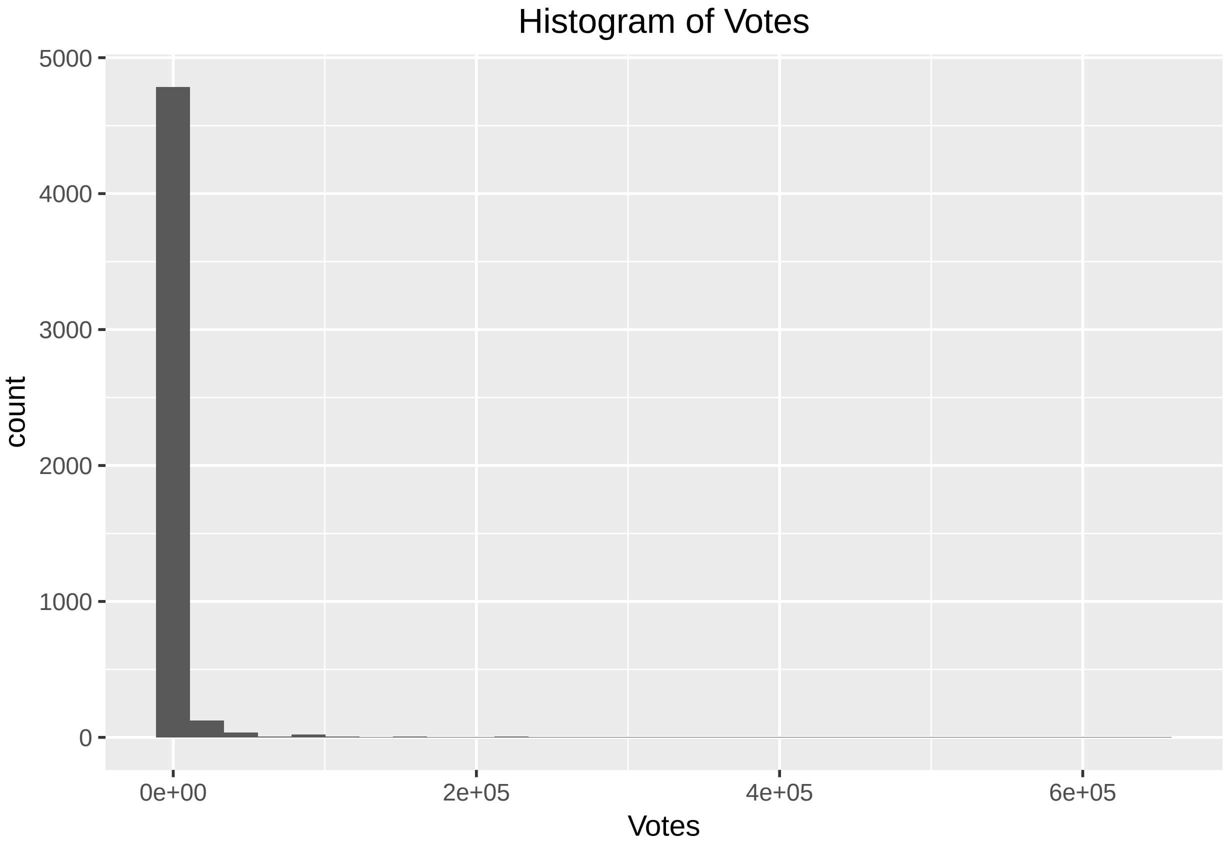
```
data$IMDbscore <- impute(data$IMDbscore,median)
data$Metascore <- impute(data$Metascore,median)
data$Votes <- impute(data$Votes,median)
data$Run_time <- impute(data$Run_time,median)
data$Genre <- impute(data$Genre, 'Drama')
```

```r
data$Popularity_rank<-as.numeric(data$Popularity_rank)
data$IMDbscore<-as.numeric(data$IMDbscore)
data$Metascore<-as.numeric(data$Metascore)
data$Votes<-as.numeric(data$Votes)
data$Run_time<-as.numeric(data$Run_time)
data$Genre<- as.factor(data$Genre)
```

Transformation for Vote

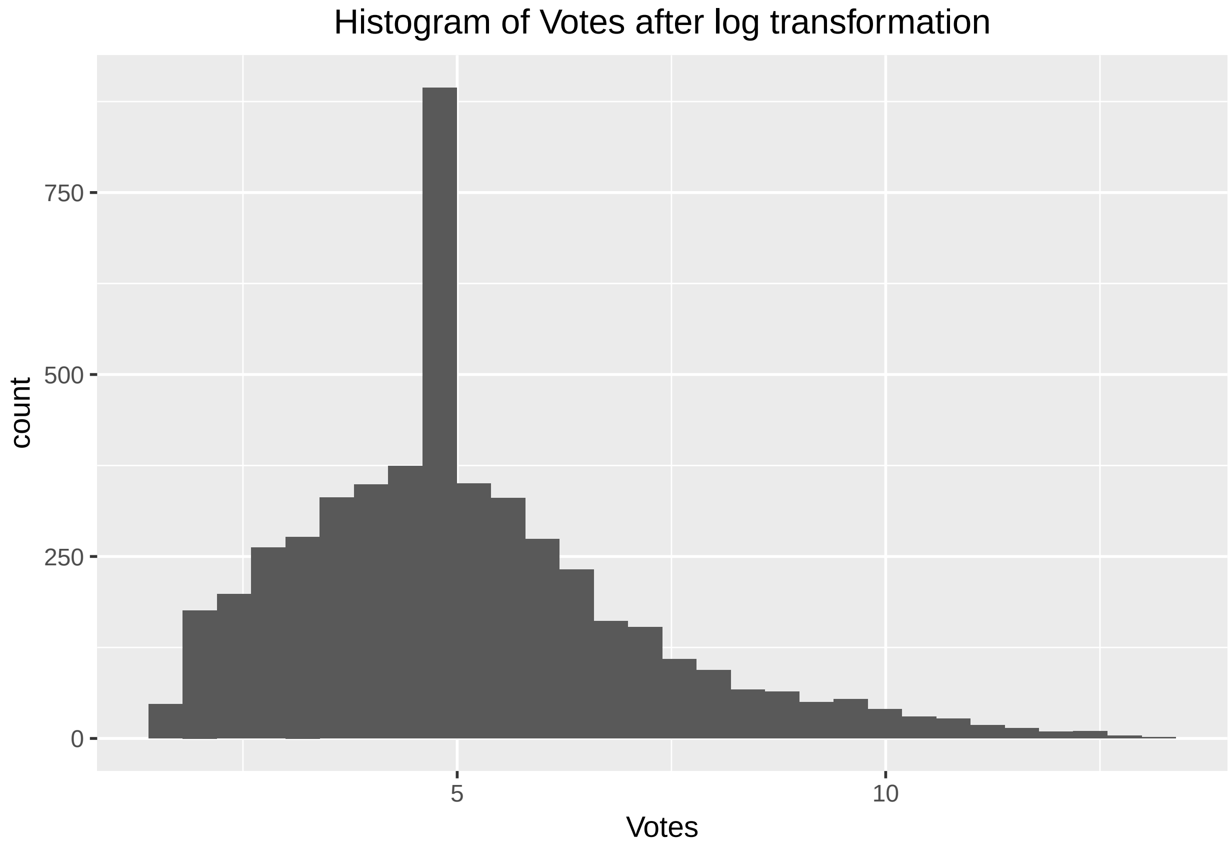
```r
ggplot(data,aes(Votes))+geom_histogram()+ labs(title="Histogram of Votes")+ theme(plot.title = element_
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
d <- data
d$Votes = log(d$Votes+1)
ggplot(d,aes(Votes))+geom_histogram() + labs(title="Histogram of Votes after log transformation")+ them
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Split data

```
#set.seed(2019)
#index <- sample(1:5000,3500,replace = FALSE)
#data <- data[index,]
#test_data<- data[-index,]
```

Model - Linear Regression

```
reg <- lm(IMDbscore ~ Popularity_rank + Metascore + log(Votes) + Run_time + Genre, data = data)
summary(reg)
```

```
##
## Call:
## lm(formula = IMDbscore ~ Popularity_rank + Metascore + log(Votes) +
##     Run_time + Genre, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8620 -0.6727  0.0934  0.7895  4.0798
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.366e+00  2.230e-01  15.092  < 2e-16 ***
## Popularity_rank  3.750e-05  1.662e-05   2.256 0.024116 *
## Metascore        2.762e-02  3.212e-03   8.601  < 2e-16 ***
## log(Votes)      -5.127e-02  1.248e-02  -4.108 4.06e-05 ***
## Run_time         1.067e-02  8.888e-04  12.000  < 2e-16 ***
## GenreAdventure   2.290e-01  9.411e-02   2.433 0.014997 *
## GenreAnimation   3.995e-01  1.131e-01   3.531 0.000418 ***
## GenreBiography   9.251e-01  1.271e-01   7.277 3.94e-13 ***
## GenreComedy      1.959e-01  6.008e-02   3.261 0.001116 **
## GenreDrama       5.958e-01  5.696e-02  10.459  < 2e-16 ***
## GenreHorror     -4.002e-01  7.173e-02  -5.580 2.54e-08 ***
## GenreRomance     4.255e-01  1.390e-01   3.061 0.002220 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.303 on 4988 degrees of freedom
## Multiple R-squared:  0.1209, Adjusted R-squared:  0.119
## F-statistic: 62.36 on 11 and 4988 DF,  p-value: < 2.2e-16
```

```
sprintf("The RMSE of the model is %f",sqrt(sum(residuals(reg)^2)/reg$df.residual))
```

```
## [1] "The RMSE of the model is 1.303423"
```

Model - Xgboost

```
data1<-data
data1$Run_time<- as.numeric(data1$Run_time)
data1$Genre <- as.numeric(data1$Genre)
x<-as.matrix(data1[,c(1,3:6)])
y<-as.matrix(data1$IMDbscore)
xgb<-xgboost(data = x, label = y, max.depth = 6,eta = 0.3, nthread = 2, verbose=2, nround = 10)
```

```
## [1]  train-rmse:4.182815
## [2]  train-rmse:3.061179
## [3]  train-rmse:2.313911
## [4]  train-rmse:1.829632
## [5]  train-rmse:1.530007
## [6]  train-rmse:1.348144
## [7]  train-rmse:1.247435
## [8]  train-rmse:1.187237
## [9]  train-rmse:1.154672
## [10] train-rmse:1.136153
```

```
summary(xgb)
```

```
##                 Length Class              Mode
## handle              1  xgb.Booster.handle externalptr
## raw             32406  -none-             raw
## niter               1  -none-             numeric
## evaluation_log      2  data.table         list
## call               16  -none-             call
## params              4  -none-             list
## callbacks           2  -none-             list
## feature_names       5  -none-             character
## nfeatures           1  -none-             numeric
```