IMDB Movie Data

Kristin Lieber

February 3, 2018

```
# READ DATA
data <- read.csv("C:\\Users\\krist_000\\Documents\\All K docs\\Statistics\\2018 Spring\\Stat proj\\IMDB
print(c("Data dimensions are", "Rows:", nrow(data), "Columns:", ncol(data)))
```

```
## [1] "Data dimensions are" "Rows:"               "1000"
## [4] "Columns:"            "12"
```

```
colnames(data)[8] <- "Runtime"
colnames(data)[11] <- "Revenue"
data <- data[, c(1:10,12,11)]
data$Year <- as.numeric(data$Year)
data$Metascore <- as.numeric(data$Metascore)
data$Votes <- as.numeric(data$Votes)
data$Runtime <- as.numeric(data$Runtime)

data2 <- data[complete.cases(data),]   # Assign to data2 only complete records
print(c("Data2 dimensions are", "Rows:", nrow(data2), "Columns:", ncol(data2)))
```

```
## [1] "Data2 dimensions are" "Rows:"                "838"
## [4] "Columns:"             "12"
```

```
#head(data)
str(data)
```

```
## 'data.frame':    1000 obs. of  12 variables:
##  $ Rank       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Title      : Factor w/ 999 levels "(500) Days of Summer",..: 288 569 656 636 674 780 403 472 834 5
##  $ Genre      : Factor w/ 207 levels "Action","Action,Adventure",..: 12 86 196 93 8 8 117 109 3 75 .
##  $ Description: Factor w/ 1000 levels "\"21\" is the fact-based story about six MIT students who were
##  $ Director   : Factor w/ 644 levels "Aamir Khan","Abdellatif Kechiche",..: 267 519 392 106 137 641 1
##  $ Actors     : Factor w/ 996 levels "Aamir Khan, Anushka Sharma, Sanjay Dutt,Boman Irani",..: 185 73
##  $ Year       : num  2014 2012 2016 2016 2016 ...
##  $ Runtime    : num  121 124 117 108 123 103 128 89 141 116 ...
##  $ Rating     : num  8.1 7 7.3 7.2 6.2 6.1 8.3 6.4 7.1 7 ...
##  $ Votes      : num  757074 485820 157606 60545 393727 ...
##  $ Metascore  : num  76 65 62 59 40 42 93 71 78 41 ...
##  $ Revenue    : num  333 126 138 270 325 ...
```

```
# Count the number of missing values in each column
na_count <-sapply(data, function(y) sum(length(which(is.na(y)))))
na_count <- data.frame(na_count)
print(c("The number of missing values out of", nrow(data)," in each column is:"))
```

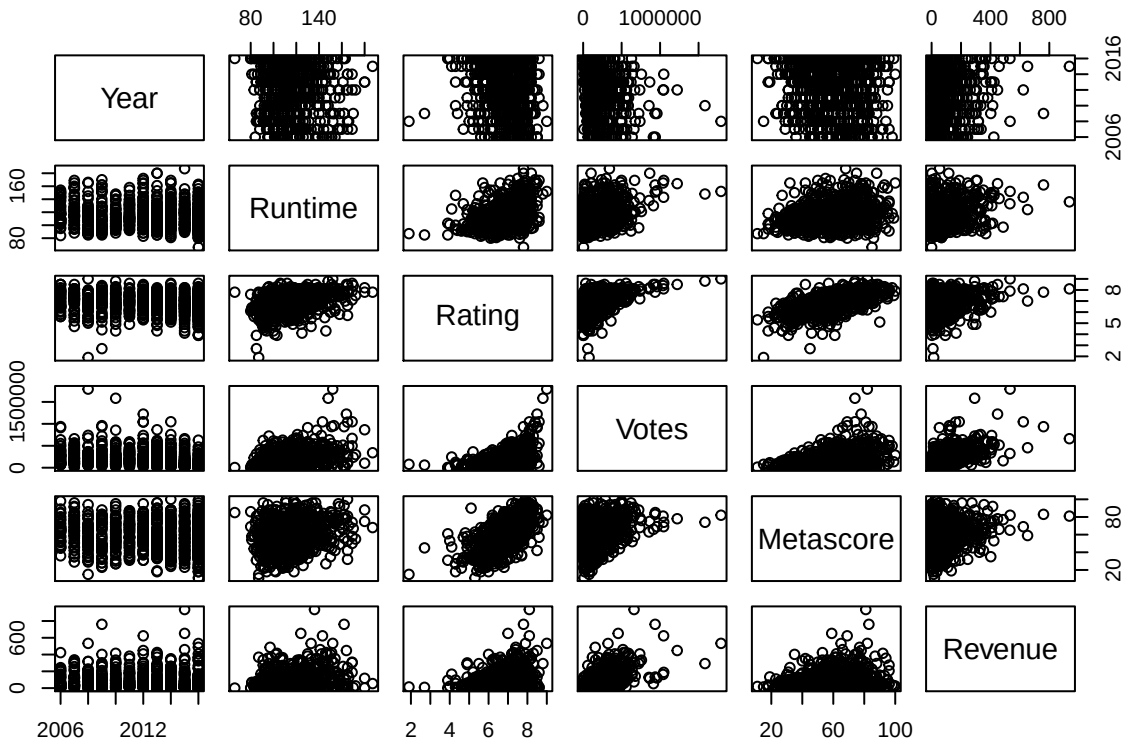
```
## [1] "The number of missing values out of"
## [2] "1000"
## [3] " in each column is:"
```

```
na_count
```

```
##             na_count
```

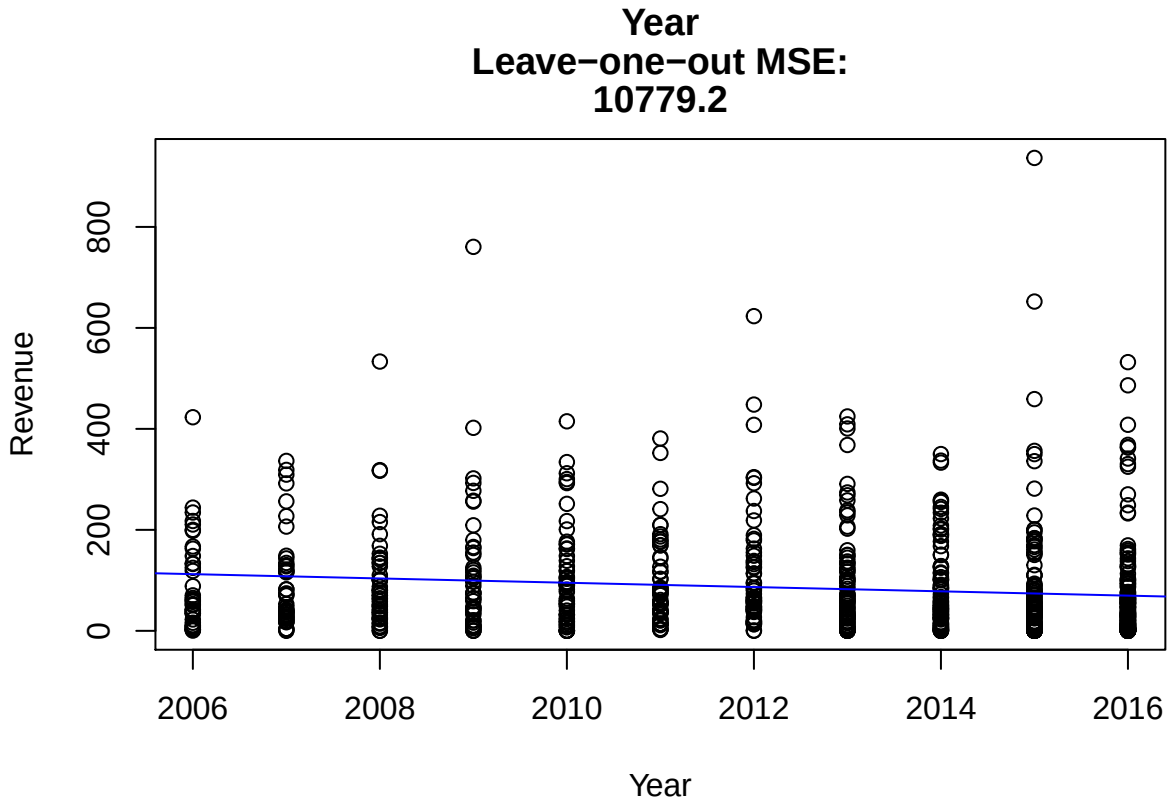
```
## Rank                0
## Title               0
## Genre               0
## Description         0
## Director            0
## Actors              0
## Year                0
## Runtime             0
## Rating              0
## Votes               0
## Metascore          64
## Revenue           128
```

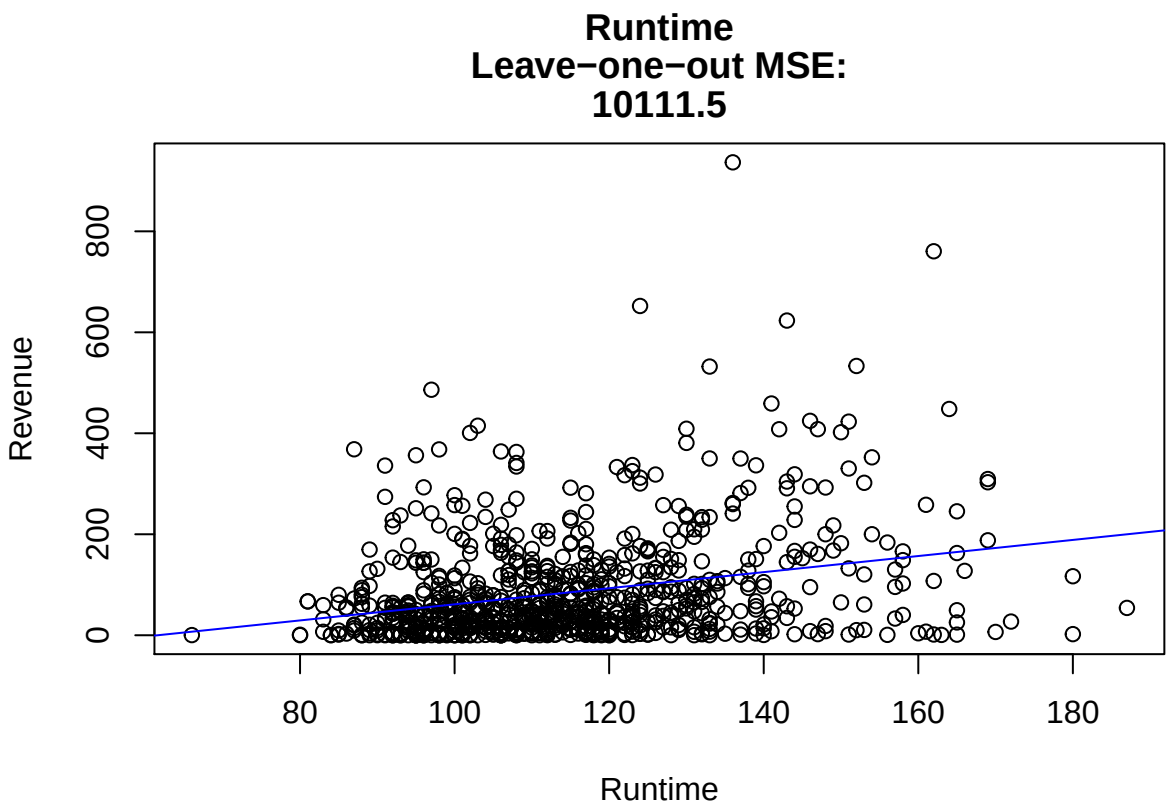
```r
# ANALYSIS

pairs(data2[7:12])  # matrix of scatterplots
```

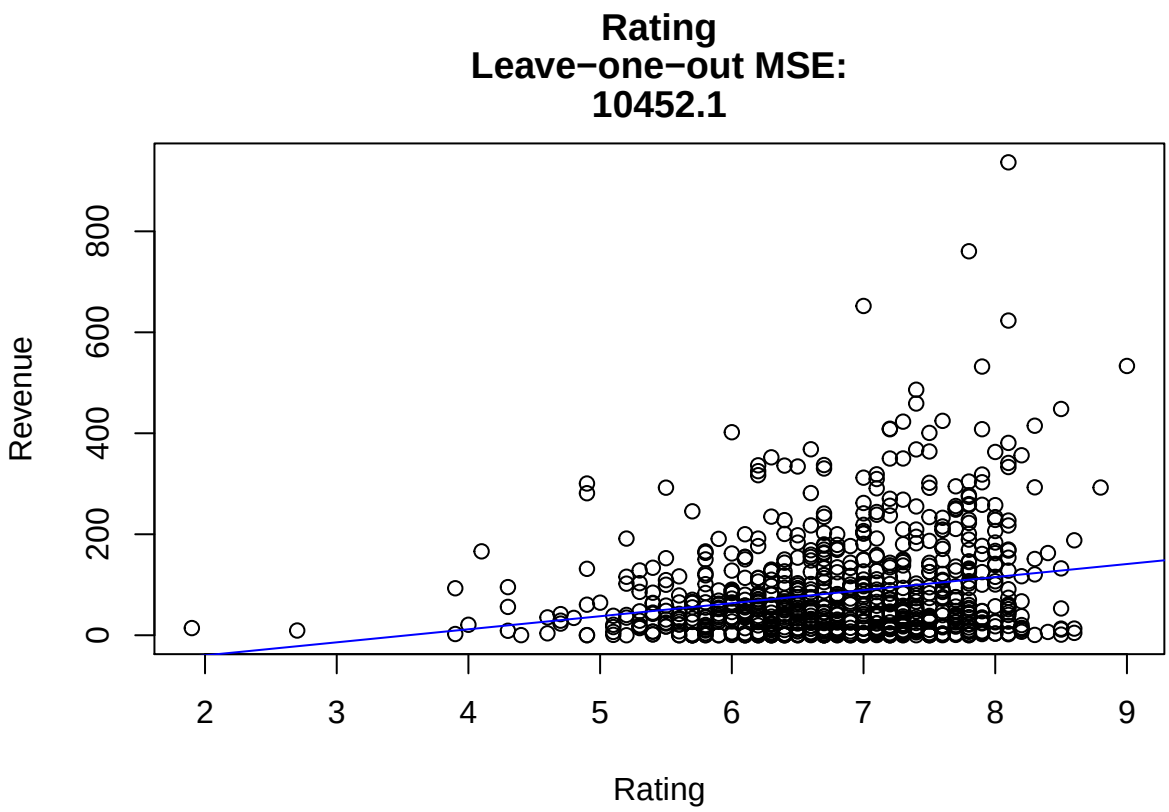


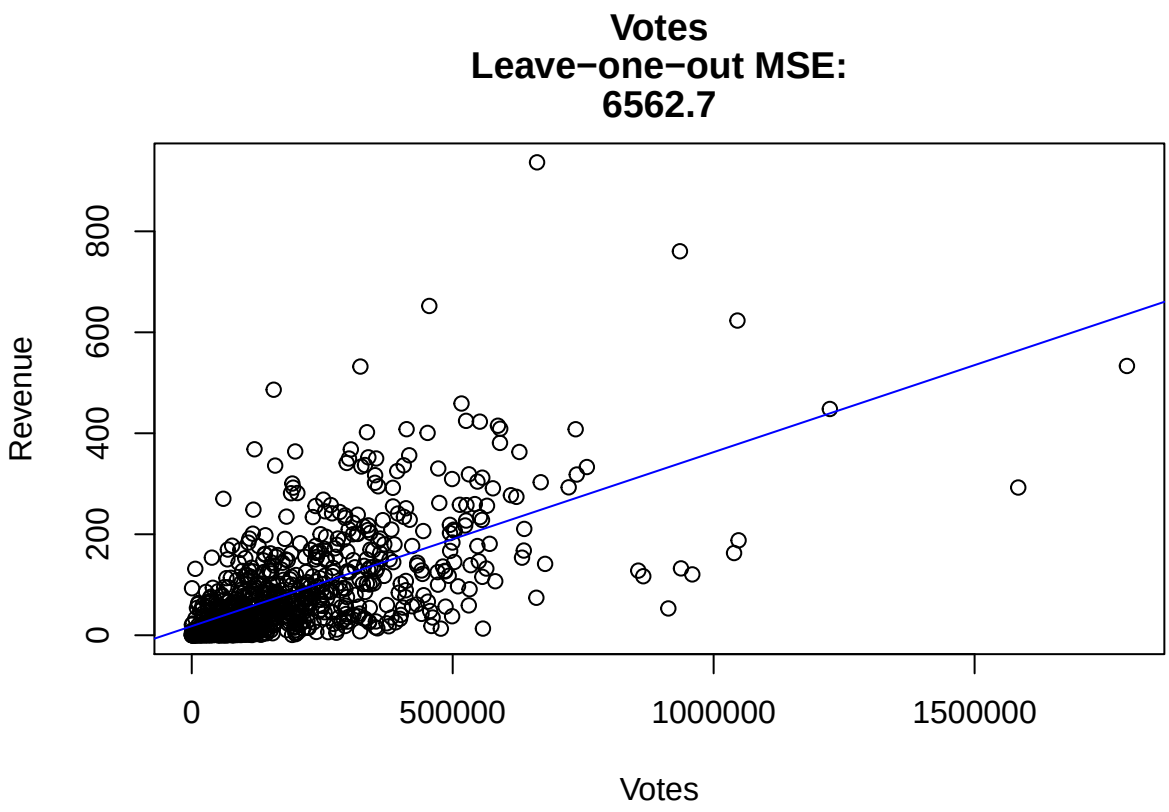

```r
# simple linear regression using one variable for
# numeric variables year, runtime, rating, votes, and metascore.

attach(data2)
numVars <- colnames(data2)[7:11]  # names of numeric variables

for(i in seq_along(numVars)){     #for each variable in numVars
    glm.fit <- glm(reformulate(numVars[i], "Revenue"))  # generalized linear model
    cv.err=cv.glm(data2,glm.fit)    #leave-one-out CV
    plot(reformulate(numVars[i], "Revenue"),
```

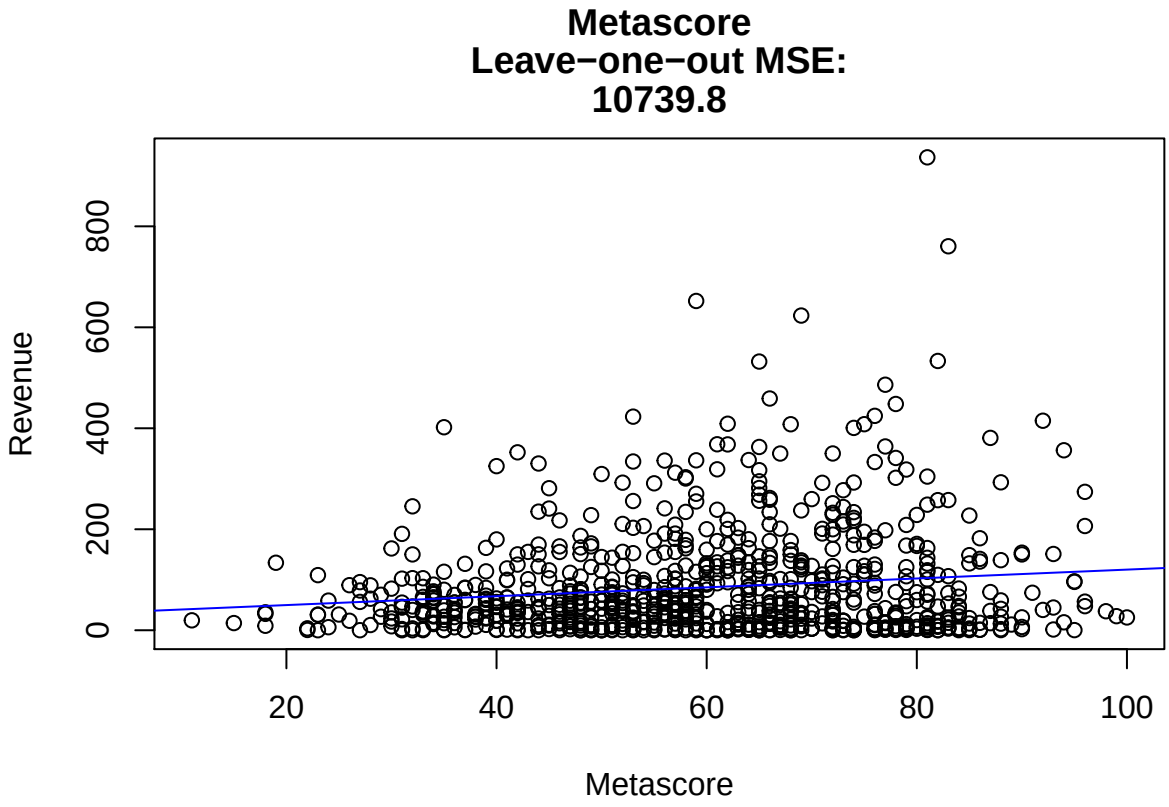
```
        main = c(numVars[i],"Leave-one-out MSE:",round(cv.err$delta[1],1)))
    abline(glm.fit, col = "blue")
}
```




Runtime
Leave−one−out MSE:
10111.5
Revenue
0
200
400
600
800
80
100
120
140
160
180
Runtime


Rating
Leave−one−out MSE:
10452.1
Revenue
0
200
400
600
800
2
3
4
5
6
7
8
9
Rating


Votes
Leave−one−out MSE:
6562.7
Revenue
0
200
400
600
800
0
500000
1000000
1500000
Votes




```
# multivariate regression with no interaction terms
glm.fit = glm(Revenue ~ Year + Runtime + Rating + Votes + Metascore)
cv.err=cv.glm(data2,glm.fit)
print (c("Year+Runtime+Rating+Votes+Metascore LOOCV:",round(cv.err$delta[1],1)))
```

```
## [1] "Year+Runtime+Rating+Votes+Metascore LOOCV:"
## [2] "6260.5"
```