EC455

Penguins

2023-11-23

The Penguins

Jabbir Ahmed(7304730), Hamza Khan(730401064), Jordan King (730465662), Maddie Peronto (PID)

```r
library(readr)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.2
```

```
## corrplot 0.92 loaded
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
df <- read_csv("MovieData455.csv")
```

```
## Rows: 51 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): Cumulative Gross, %± LY, #1 Release, % of Total, Month, Genre, rating
## dbl  (5): Year, Releases, Average, Gross, Budget
## time (1): runtime
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(df)
```

```
## # A tibble: 6 x 13
##    Year ‘Cumulative Gross‘ ‘%± LY‘ Releases Average ‘#1 Release‘           Gross
##   <dbl> <chr>              <chr>      <dbl>   <dbl> <chr>                  <dbl>
## 1  2023 $584,765,684.00    50%           89 6570400 Avatar: The Way of W~ 2.23e8
## 2  2022 $389,782,780.00    498%          66 5905799 Spider-Man: No Way H~ 1.64e8
## 3  2021 $65,138,132.00     -93%          55 1184329 Wonder Woman 1984     1.62e7
## 4  2020 $897,742,569.00    10%          146 6148921 Bad Boys for Life     1.36e8
## 5  2019 $812,849,718.00    -15%         161 5048756 Aquaman               1.20e8
## 6  2018 $958,572,920.00    -            154 6224499 Jumanji: Welcome to ~ 1.72e8
## # i 6 more variables: ‘% of Total‘ <chr>, Month <chr>, Budget <dbl>,
## #   Genre <chr>, rating <chr>, runtime <time>
```

Data cleaning, keeping relevant variables, creating a numeric-only data frame

```
df1 <- df[,c(1, 6, 7, 9, 10, 11, 12)]
colnames(df1)[2] = "Name"
colnames(df1)[7] = "Rating"
df$Genre<-as.factor(df1$Genre)
df1$Rating <- as.factor(df1$Rating)
df1$Month <- as.factor(df1$Month)

head(df1)
```

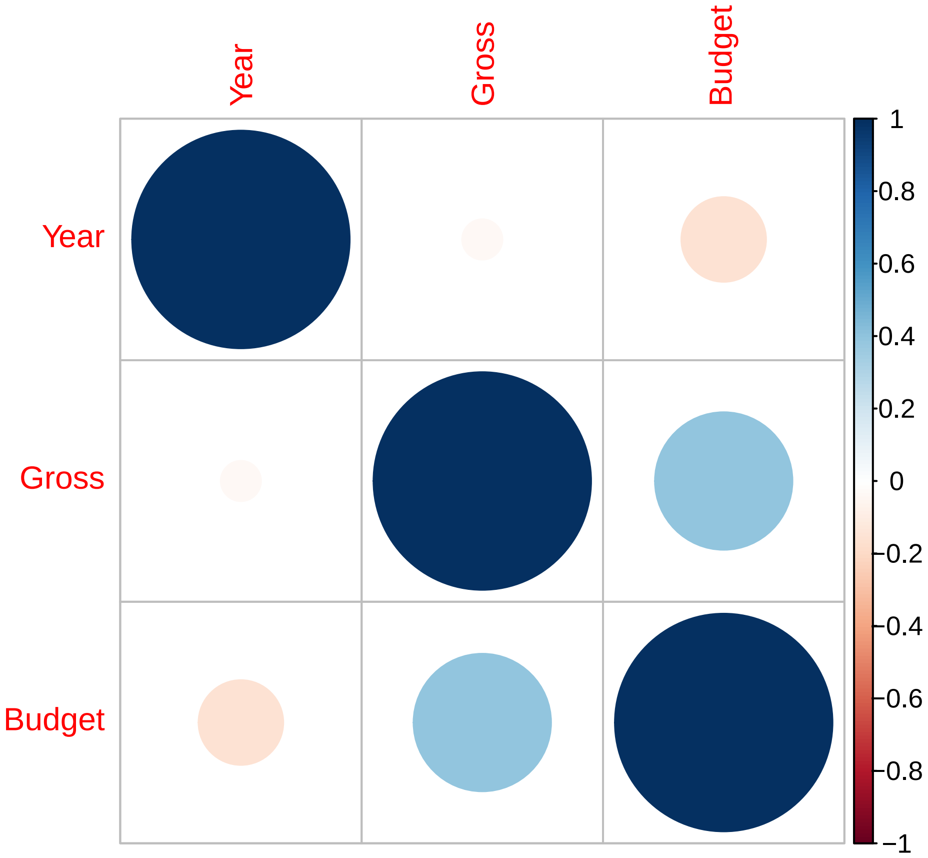
```
## # A tibble: 6 x 7
##    Year Name                               Gross Month      Budget Genre  Rating
##   <dbl> <chr>                              <dbl> <fct>       <dbl> <chr>  <fct>
## 1  2023 Avatar: The Way of Water       222528552 January 250000000 anima~ PG-13
## 2  2022 Spider-Man: No Way Home        163815488 January 200000000 super~ PG-13
## 3  2021 Wonder Woman 1984               16190880 January 200000000 super~ PG-13
## 4  2020 Bad Boys for Life              135561888 January  90000000 crime  R
## 5  2019 Aquaman                        119682416 January 205000000 super~ PG-13
## 6  2018 Jumanji: Welcome to the Jungle 171792998 January  90000000 Sci-Fi PG-13
```

```
df1.num <- df1[,c(1,3,5)]
head(df1.num)
```

```
## # A tibble: 6 x 3
##    Year     Gross    Budget
##   <dbl>     <dbl>     <dbl>
## 1  2023 222528552 250000000
## 2  2022 163815488 200000000
## 3  2021  16190880 200000000
## 4  2020 135561888  90000000
## 5  2019 119682416 205000000
## 6  2018 171792998  90000000
```

Correlation Matrix: I found very low correlations

```
df1.cor <- cor(df1.num)
corrplot(df1.cor)
```



Creating linear models with different variables and interactions

```
#Year*Month + Budget*Genre + Genre*Rating + Budget*Gross +
mod1 <- lm(Gross ~ Year + Month +  Gross + Budget + Genre + Rating, data = df1)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on
## the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 3 in
## model.matrix: no columns are assigned
```

```
summary(mod1)
```

```
##
## Call:
## lm(formula = Gross ~ Year + Month + Gross + Budget + Genre +
##     Rating, data = df1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -180462316  -51950845    -251188   70706124  259554482
```

```
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.503e+08  2.550e+10   0.029    0.977
## Year              -3.178e+05  1.262e+07  -0.025    0.980
## MonthAugust       -1.214e+08  1.029e+08  -1.180    0.248
## MonthFebruary     -9.688e+07  8.667e+07  -1.118    0.274
## MonthJanuary      -1.325e+08  9.327e+07  -1.420    0.167
## MonthJuly          2.474e+07  8.971e+07   0.276    0.785
## MonthJune          7.915e+07  1.036e+08   0.764    0.452
## MonthMarch        -7.182e+07  9.448e+07  -0.760    0.454
## MonthMay          -2.172e+07  1.051e+08  -0.207    0.838
## MonthOctober      -9.495e+07  1.099e+08  -0.864    0.395
## MonthSeptember    -1.261e+08  1.080e+08  -1.168    0.253
## Budget             2.547e-01  3.909e-01   0.652    0.520
## Genrecrime         6.132e+05  1.836e+08   0.003    0.997
## Genredocumentary  4.260e+07  1.748e+08   0.244    0.809
## Genrehistory      -4.492e+07  1.782e+08  -0.252    0.803
## GenreHorror        2.002e+07  1.703e+08   0.118    0.907
## GenreSci-Fi       -2.739e+07  8.468e+07  -0.323    0.749
## Genresports       -1.636e+06  1.714e+08  -0.010    0.992
## Genresuperhero     4.742e+07  8.284e+07   0.572    0.572
## Genrethriller     -7.002e+07  1.152e+08  -0.608    0.548
## RatingPG           9.850e+07  1.581e+08   0.623    0.539
## RatingPG-13        9.209e+07  1.824e+08   0.505    0.618
## RatingR            1.362e+08  2.172e+08   0.627    0.536
##
## Residual standard error: 130500000 on 27 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4056, Adjusted R-squared:  -0.07871
## F-statistic: 0.8375 on 22 and 27 DF,  p-value: 0.6616
```

The data from above proved to not be helpful, will be attempting with a different dataset.

```
df2 <- read_csv("df2na.csv")
```

```
## Rows: 50 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Company, Film, Director, Genre, Theme
## dbl (5): Release, Domestic, Budget, Runtime, Day1
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(df2)
```

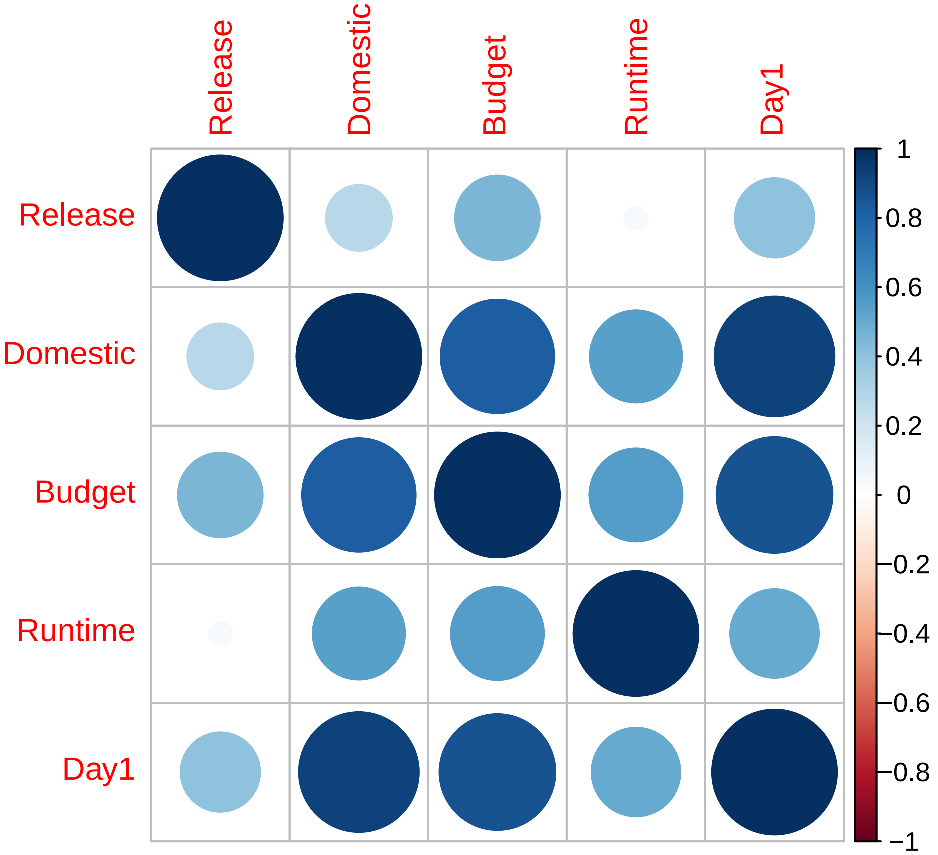
```
## # A tibble: 6 x 10
##   Company Film        Release Domestic Director Budget Genre Runtime Theme  Day1
##   <chr>   <chr>         <dbl>    <dbl> <chr>     <dbl> <chr>   <dbl> <chr> <dbl>
## 1 Marvel  Fantastic ~    2005     155. Tim Sto~    100 Acti~     106 Spac~  25.6
## 2 DC      Batman Ret~    1992     163. Tim Bur~     80 Acti~     126 Dark~  45.6
## 3 DC      Batman Beg~    2005     207. Christo~    150 Acti~     140 Orig~  48.7
## 4 DC       Superman R~    2006      200. Bryan S~    204 Acti~     154 Retu~   52.5
## 5 DC       Batman For~    1995      184. Joel Sc~    100 Acti~     121 Two--~   52.7
## 6 Marvel   X-Men          2000      157. Bryan S~     75 Acti~     104 Muta~   54.4
```

Creating a numeric-only data frame

```
df2.num <- df2[,c(3,4,6,8,10)]
df2.num <- na.omit(df2.num)
```

Created a correlation matrix and found the following correlations: - Budget and Runtime - Budget and Domestic - Domestic and Runtime The last two correlations are not useful as we want to predict Domestic.

```
df2.cor <- cor(df2.num)
corrplot(df2.cor)
```



Utilizing Budget, Release and Runtime to create a linear model. Budget and Release have low p-values.

```
mod2 <- lm(Domestic ~ Budget + Release + Runtime+ Day1, data = df2)
summary(mod2)
```

```
##
## Call:
## lm(formula = Domestic ~ Budget + Release + Runtime + Day1, data = df2)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -69.115 -15.626  -6.581   7.507 130.214
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3053.4963  1621.3384   1.883   0.0661 .
## Budget         0.2200     0.1616   1.361   0.1802
## Release       -1.5389     0.8045  -1.913   0.0621 .
## Runtime        0.3201     0.3987   0.803   0.4263
## Day1           2.1475     0.2839   7.563  1.5e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 36.66 on 45 degrees of freedom
## Multiple R-squared:  0.8685, Adjusted R-squared:  0.8568
## F-statistic: 74.29 on 4 and 45 DF,  p-value: < 2.2e-16
```

More linear models. Budget again has a low p-value.

```
mod3 <- lm(Release ~ as.factor(Company) + Release + Budget + Runtime + Day1, data = df2)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on
## the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 2 in
## model.matrix: no columns are assigned
```

```
summary(mod3)
```

```
##
## Call:
## lm(formula = Release ~ as.factor(Company) + Release + Budget +
##     Runtime + Day1, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.368  -2.772   0.333   3.873  10.162
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2009.67471    8.01805 250.644   <2e-16 ***
## as.factor(Company)Marvel    3.49526    2.18225   1.602   0.1162
## Budget                      0.06943    0.02782   2.496   0.0163 *
## Runtime                    -0.11658    0.07137  -1.633   0.1093
## Day1                       -0.01768    0.05473  -0.323   0.7482
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 6.607 on 45 degrees of freedom
## Multiple R-squared:  0.3208, Adjusted R-squared:  0.2605
## F-statistic: 5.314 on 4 and 45 DF,  p-value: 0.001363
```

More linear modeling with very high p-values this time.

```
mod4 <- lm(Domestic ~ as.factor(Company) + Release + Budget + Runtime + Budget*Runtime + Day1, data = d
summary(mod4)
```

```
##
## Call:
## lm(formula = Domestic ~ as.factor(Company) + Release + Budget +
##     Runtime + Budget * Runtime + Day1, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -68.616 -17.045  -5.845  10.319 127.982
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.100e+03  1.781e+03   1.741   0.0889 .
## as.factor(Company)Marvel  7.033e+00  1.274e+01   0.552   0.5836
## Release                   -1.578e+00  8.752e-01  -1.803   0.0783 .
## Budget                     4.526e-01  6.989e-01   0.648   0.5207
## Runtime                    5.735e-01  7.663e-01   0.748   0.4583
## Day1                       2.062e+00  3.164e-01   6.515  6.5e-08 ***
## Budget:Runtime            -1.695e-03  5.382e-03  -0.315   0.7543
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.31 on 43 degrees of freedom
## Multiple R-squared:  0.8698, Adjusted R-squared:  0.8517
## F-statistic: 47.89 on 6 and 43 DF,  p-value: < 2.2e-16
```

Created two Full linear models, the second one is terrible. The first one is pretty good and I will use it throughout.

```
Full <- lm(Domestic ~ Budget + Runtime + as.factor(Company) + Release+ Day1, data = df2)
summary(Full)
```

```
##
## Call:
## lm(formula = Domestic ~ Budget + Runtime + as.factor(Company) +
##     Release + Day1, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -72.550 -15.936  -5.683   9.507 127.922
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3274.0474  1674.8555   1.955   0.0570 .
## Budget                       0.2389     0.1659   1.440   0.1568
## Runtime                      0.3706     0.4105   0.903   0.3716
## as.factor(Company)Marvel     7.4436    12.5385   0.594   0.5558
## Release                     -1.6538     0.8331  -1.985   0.0534 .
## Day1                         2.0825     0.3062   6.800 2.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.92 on 44 degrees of freedom
## Multiple R-squared:  0.8695, Adjusted R-squared:  0.8547
## F-statistic: 58.64 on 5 and 44 DF,  p-value: < 2.2e-16
```

```
Full2 <- lm(Domestic ~
              Budget +
              Runtime +
              as.factor(Company) +
              Release +
              Budget*Runtime +
              as.factor(Company)*Runtime +
              as.factor(Company)*Budget +
              as.factor(Company)*Release, data = df2)
# summary(Full2)
```

Attempt 2

```
df2na <- na.omit(df2)
df2na <- df2na[order(-df2na$Domestic),]
df2na <- df2na[-c(1:4),]
Full <- lm(Domestic ~ Budget + Runtime + as.factor(Company) + Release+ Day1, data = df2)
summary(Full)
```

```
##
## Call:
## lm(formula = Domestic ~ Budget + Runtime + as.factor(Company) +
##     Release + Day1, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -72.550 -15.936  -5.683   9.507 127.922
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3274.0474  1674.8555   1.955   0.0570 .
## Budget                     0.2389     0.1659   1.440   0.1568
## Runtime                    0.3706     0.4105   0.903   0.3716
## as.factor(Company)Marvel    7.4436    12.5385   0.594   0.5558
## Release                   -1.6538     0.8331  -1.985   0.0534 .
## Day1                       2.0825     0.3062   6.800 2.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.92 on 44 degrees of freedom
## Multiple R-squared:  0.8695, Adjusted R-squared:  0.8547
## F-statistic: 58.64 on 5 and 44 DF,  p-value: < 2.2e-16
```

Going to perform forward/backwards/stepwise selections to find the most important variables.

```
# Backward
MSE=(summary(Full)$sigma)^2
step(Full, scale=MSE)
```

```
## Start:  AIC=6
## Domestic ~ Budget + Runtime + as.factor(Company) + Release +
##     Day1
##
##                      Df Sum of Sq    RSS      Cp
## - as.factor(Company)  1       480  60463  4.3524
## - Runtime             1      1111  61093  4.8149
## <none>                             59982  6.0000
## - Budget              1      2829  62811  6.0750
## - Release             1      5372  65354  7.9405
## - Day1                1     63039 123021 50.2419
##
## Step:  AIC=4.35
## Domestic ~ Budget + Runtime + Release + Day1
##
##           Df Sum of Sq    RSS      Cp
## - Runtime  1       866  61329  2.9877
## - Budget   1      2490  62953  4.1791
## <none>                  60463  4.3524
## - Release  1      4917  65379  5.9591
## - Day1     1     76856 137319 58.7305
##
## Step:  AIC=2.99
## Domestic ~ Budget + Release + Day1
##
##           Df Sum of Sq    RSS      Cp
## <none>                  61329  2.9877
## - Budget   1      4092  65420  3.9892
## - Release  1      6842  68170  6.0063
## - Day1     1     78403 139732 58.5003

##
## Call:
## lm(formula = Domestic ~ Budget + Release + Day1, data = df2)
##
## Coefficients:
## (Intercept)       Budget      Release         Day1
##   3473.1280       0.2647      -1.7321       2.1636
```

```
195.460 - 1.185*log(273.8) - 1.774*log(105)
```

```
## [1] 180.5532
```

```
#Forward
none=lm(Domestic ~ 1, data = df2na)
step(none,scope=list(upper=Full), scale=MSE,direction="forward")
```

```
## Start:  AIC=166.66
## Domestic ~ 1
##
##                        Df Sum of Sq    RSS      Cp
## + Day1                  1    243941  43232 -10.287
## + Budget                1    200219  86955  21.786
## + Runtime               1     90814 196359 102.040
## + Release               1     25779 261395 149.747
## + as.factor(Company)    1      8731 278443 162.252
## <none>                               287174 166.657
##
## Step:  AIC=-10.29
## Domestic ~ Day1
##
##                        Df Sum of Sq   RSS       Cp
## + Runtime               1    4348.9 38884 -11.4770
## + Release               1    3786.9 39446 -11.0647
## <none>                              43232 -10.2868
## + Budget                1    2130.9 41102  -9.8499
## + as.factor(Company)    1     998.5 42234  -9.0193
##
## Step:  AIC=-11.48
## Domestic ~ Day1 + Runtime
##
##                        Df Sum of Sq   RSS       Cp
## <none>                              38884 -11.4770
## + Release               1   2187.35 36696 -11.0815
## + Budget                1    969.24 37914 -10.1879
## + as.factor(Company)    1    126.99 38757  -9.5701
```

```
##
## Call:
## lm(formula = Domestic ~ Day1 + Runtime, data = df2na)
##
## Coefficients:
## (Intercept)         Day1      Runtime
##    -55.0559       2.2101       0.6677
```

```
# Stepwise
step(none,scope=list(upper=Full),scale=MSE)
```

```
## Start:  AIC=166.66
## Domestic ~ 1
##
##                        Df Sum of Sq    RSS      Cp
## + Day1                  1    243941  43232 -10.287
## + Budget                1    200219  86955  21.786
## + Runtime               1     90814 196359 102.040
## + Release               1     25779 261395 149.747
## + as.factor(Company)    1      8731 278443 162.252
## <none>                               287174 166.657
##
## Step:  AIC=-10.29
## Domestic ~ Day1
```

```
## 
##                        Df Sum of Sq    RSS       Cp
## + Runtime               1      4349  38884 -11.4770
## + Release               1      3787  39446 -11.0647
## <none>                               43232 -10.2868
## + Budget                1      2131  41102  -9.8499
## + as.factor(Company)  1       999  42234  -9.0193
## - Day1                  1    243941 287174 166.6566
## 
## Step:  AIC=-11.48
## Domestic ~ Day1 + Runtime
## 
##                        Df Sum of Sq    RSS       Cp
## <none>                               38884 -11.4770
## + Release               1      2187  36696 -11.0815
## - Runtime               1      4349  43232 -10.2868
## + Budget                1       969  37914 -10.1879
## + as.factor(Company)  1       127  38757  -9.5701
## - Day1                  1    157476 196359 102.0396
```

```
## 
## Call:
## lm(formula = Domestic ~ Day1 + Runtime, data = df2na)
## 
## Coefficients:
## (Intercept)         Day1      Runtime  
##    -55.0559       2.2101       0.6677
```

```
mod5 = step(none, scope=list(upper=Full), scale=MSE, trace = FALSE)
summary(mod5)
```

```
## 
## Call:
## lm(formula = Domestic ~ Day1 + Runtime, data = df2na)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -67.603 -13.536  -6.950   4.423  89.250 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -55.0559    33.3083  -1.653   0.1056    
## Day1          2.2101     0.1675  13.196   <2e-16 ***
## Runtime       0.6677     0.3045   2.193   0.0338 *  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 30.07 on 43 degrees of freedom
## Multiple R-squared:  0.8646, Adjusted R-squared:  0.8583 
## F-statistic: 137.3 on 2 and 43 DF,  p-value: < 2.2e-16
```

```
coef(mod5)
```

```
## (Intercept)        Day1     Runtime
## -55.0559147   2.2101290   0.6677124
```

```
-55.0559147+ 2.2101290*(46.1)+ 0.6677124*(105)
```

```
## [1] 116.9408
```

Through lowest AIC we found: Budget and Runtime are the best variables. Going to make linear model and add interactions between the important variables as well as squaring them.

```
slm1 <- lm(Domestic ~ Runtime, data = df2na)
summary(slm1)
```

```
##
## Call:
## lm(formula = Domestic ~ Runtime, data = df2na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -128.72  -51.34  -10.52   55.67  152.00
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -193.5279    70.2274  -2.756  0.00849 **
## Runtime        2.6528     0.5881   4.511 4.76e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 66.8 on 44 degrees of freedom
## Multiple R-squared:  0.3162, Adjusted R-squared:  0.3007
## F-statistic: 20.35 on 1 and 44 DF,  p-value: 4.755e-05
```

```
slm2 <- lm(Domestic ~ Budget, data = df2na)
summary(slm2)
```

```
##
## Call:
## lm(formula = Domestic ~ Budget, data = df2na)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -112.085  -26.443   -7.599   24.175   93.446
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.96944   12.58787   0.951    0.347
## Budget       0.99577    0.09893  10.065 5.46e-13 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Residual standard error: 44.45 on 44 degrees of freedom
## Multiple R-squared:  0.6972, Adjusted R-squared:  0.6903
## F-statistic: 101.3 on 1 and 44 DF,  p-value: 5.462e-13
```

```
slm3 <- lm(Domestic ~ Runtime + Budget, data = df2na)
summary(slm3)
```

```
##
## Call:
## lm(formula = Domestic ~ Runtime + Budget, data = df2na)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.283  -27.887   -4.226   29.075   81.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -68.6115    48.4536  -1.416   0.1640
## Runtime       0.7776     0.4522   1.720   0.0927 .
## Budget        0.8912     0.1143   7.796 9.24e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 43.5 on 43 degrees of freedom
## Multiple R-squared:  0.7167, Adjusted R-squared:  0.7035
## F-statistic: 54.39 on 2 and 43 DF,  p-value: 1.674e-12
```

```
slm4 <- lm(Domestic ~ Runtime + Budget + Budgetˆ2, data = df2na)
summary(slm4)
```

```
##
## Call:
## lm(formula = Domestic ~ Runtime + Budget + Budgetˆ2, data = df2na)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.283  -27.887   -4.226   29.075   81.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -68.6115    48.4536  -1.416   0.1640
## Runtime       0.7776     0.4522   1.720   0.0927 .
## Budget        0.8912     0.1143   7.796 9.24e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 43.5 on 43 degrees of freedom
## Multiple R-squared:  0.7167, Adjusted R-squared:  0.7035
## F-statistic: 54.39 on 2 and 43 DF,  p-value: 1.674e-12
```

```
slm5 <- lm(Domestic ~ Runtime + Budget + Runtime^2, data = df2na)
summary(slm5)
```

```
##
## Call:
## lm(formula = Domestic ~ Runtime + Budget + Runtime^2, data = df2na)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.283  -27.887   -4.226   29.075   81.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -68.6115    48.4536  -1.416   0.1640
## Runtime       0.7776     0.4522   1.720   0.0927 .
## Budget        0.8912     0.1143   7.796 9.24e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 43.5 on 43 degrees of freedom
## Multiple R-squared:  0.7167, Adjusted R-squared:  0.7035
## F-statistic: 54.39 on 2 and 43 DF,  p-value: 1.674e-12
```

```
slm6 <- lm(Domestic ~ Runtime + Budget + Runtime*Budget, data = df2na)
summary(slm6)
```

```
##
## Call:
## lm(formula = Domestic ~ Runtime + Budget + Runtime * Budget,
##     data = df2na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -98.129 -28.637   0.294  22.371  79.497
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.311e+02  9.057e+01  -2.552  0.01444 *
## Runtime         2.165e+00  7.928e-01   2.730  0.00921 **
## Budget          2.386e+00  7.223e-01   3.303  0.00196 **
## Runtime:Budget -1.220e-02  5.826e-03  -2.094  0.04237 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 41.88 on 42 degrees of freedom
## Multiple R-squared:  0.7435, Adjusted R-squared:  0.7251
## F-statistic: 40.57 on 3 and 42 DF,  p-value: 1.788e-12
```

More transformations - log is the best. Highest Rˆ2, lowest p-values, good t-values, low standard error.

```
slm7 <- lm(log(Domestic) ~ log(Runtime) + log(Budget)+ log(Day1), data = df2na)
summary(slm7)
```

```
##
## Call:
## lm(formula = log(Domestic) ~ log(Runtime) + log(Budget) + log(Day1),
##     data = df2na)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55227 -0.14806 -0.05531  0.12668  0.93145
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.2659     1.6439  -1.378   0.1754
## log(Runtime)   0.7272     0.3640   1.998   0.0523 .
## log(Budget)   -0.0112     0.1256  -0.089   0.9294
## log(Day1)      0.9601     0.0886  10.837 9.66e-14 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.2914 on 42 degrees of freedom
## Multiple R-squared:  0.9312, Adjusted R-squared:  0.9263
## F-statistic: 189.5 on 3 and 42 DF,  p-value: < 2.2e-16
```

```
slm8 <- lm(Domestic^2 ~ Runtime^2 + Budget^2+ Day1^2, data = df2na)
summary(slm8)
```

```
##
## Call:
## lm(formula = Domestic^2 ~ Runtime^2 + Budget^2 + Day1^2, data = df2na)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -20457  -5805   -348   4399  32248
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18170.28   11302.55  -1.608    0.115
## Runtime         98.72     105.56   0.935    0.355
## Budget          32.75      45.16   0.725    0.472
## Day1           542.31      95.71   5.666  1.2e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 10120 on 42 degrees of freedom
## Multiple R-squared:  0.7994, Adjusted R-squared:  0.7851
## F-statistic:  55.8 on 3 and 42 DF,  p-value: 1.054e-14
```

```
slm9 <- lm(Domestic ~ log(Runtime) + log(Budget)+log(Day1), data = df2na)
summary(slm9)
```

```
##
## Call:
## lm(formula = Domestic ~ log(Runtime) + log(Budget) + log(Day1),
##     data = df2na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -72.762 -27.547  -1.049  30.026  93.779
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -510.12     216.08  -2.361  0.02296 *
## log(Runtime)    65.56      47.85   1.370  0.17791
## log(Budget)     43.96      16.51   2.662  0.01095 *
## log(Day1)       35.62      11.65   3.059  0.00386 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.3 on 42 degrees of freedom
## Multiple R-squared:  0.7854, Adjusted R-squared:  0.7701
## F-statistic: 51.25 on 3 and 42 DF,  p-value: 4.309e-14
```

```
slm10 <- lm(Domestic ~ Runtime^2 + Budget^2+Day1^2, data = df2na)
summary(slm10)
```

```
##
## Call:
## lm(formula = Domestic ~ Runtime^2 + Budget^2 + Day1^2, data = df2na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.359 -13.399  -5.378   6.082  78.359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -50.4927    33.5699  -1.504   0.1400
## Runtime       0.5891     0.3135   1.879   0.0672 .
## Budget        0.1390     0.1341   1.036   0.3060
## Day1          1.9720     0.2843   6.937 1.79e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.05 on 42 degrees of freedom
## Multiple R-squared:  0.868,  Adjusted R-squared:  0.8585
## F-statistic: 92.04 on 3 and 42 DF,  p-value: < 2.2e-16
```

Utilizing slm7 to make our prediction for a movie with runtime of 1 hour 45 mins (105 mins) and budget of $273.8.

```
coef(slm7)
```

```
##  (Intercept) log(Runtime)  log(Budget)    log(Day1)
##  -2.26592140   0.72715682  -0.01120394   0.96007506
```

```
-2.26592140+0.72715682*log(105)-0.01120394*log(273.8)+0.96007506*log(46.1)
```

```
## [1] 4.733225
```

```
exp(4.733225)
```

```
## [1] 113.6615
```

Our estimate is $113.6615 (in millions), which seems reasonable considering the performance movie and including the factors of the day1 box office. Moreover, our orginal conclusion was $116.9408 (in Millions) but we wanted a more concrete answer utilizing the transformation which allowed us to have a better adjusted r-squared with a good p-value with a high f-statistic making our model more reliable.