Regression

Team: Abed Ahmed (ASA190005) & Dylan Kapustka (DLK190000)

Date: 09/25/2022

HomeWork 3 - Regression

Linear Regression

Linear Regression is a Machine Learning algorithm based on the mathematics concept that allows us to predict one dependent target variable, based on one or more independent variables.

In very simple terms, if we look at the equation $\mathbf{y = mx + b}$, With target **y** and independant given variable **x**, Linear Regression will estimate values **m** and **b** allowing us to plug them in and estimate target **y** for any **x** .

In this notebook, we will look at a data set describing various attributes of Anime and Manga, and build some linear models with them.

Data Source: https://www.kaggle.com/datasets/hernan4444/anime-recommendation-database-2020

Pros of Linear Regression

- Easy To implement and interpret
- Easy to identify Use cases for by spotting potential correlations in data
- Has some techniques to avoid over fitting

Cons of Linear Regression

- Assumes relationship between independent and dependent variables. Assumes there is a straight line.
- Does not provide a complete description of relationships among variables.

The below chunk does the following

- Imports and cleans data
- Splits into 80/20 train/test
- Explores the data
- Plots two informative graphs

```r
data <- read.csv(file="Anime-Data.csv",header=TRUE)
data = dplyr::select(data, -c('Japanese.name','Score.1':'Score.10'))
data <- na.omit(data)

set.seed(2)
library(caTools)
```

```
split <- sample.split(data,SplitRatio=0.8)
train <- subset(data, split==TRUE)
test <- subset(data,split==FALSE)

names(train)
```

```
##  [1] "MAL_ID"        "Name"          "Score"         "Genres"
##  [5] "English.name"  "Type"          "Episodes"      "Aired"
##  [9] "Premiered"     "Producers"     "Licensors"     "Studios"
## [13] "Source"        "Duration"      "Rating"        "Ranked"
## [17] "Popularity"    "Members"       "Favorites"     "Watching"
## [21] "Completed"     "On.Hold"       "Dropped"       "Plan.to.Watch"
```

```
head(train)
```

```
##    MAL_ID                 Name Score
## 1       1         Cowboy Bebop  8.78
## 3       6               Trigun  8.24
## 4       7   Witch Hunter Robin  7.27
## 7      16 Hachimitsu to Clover  8.06
## 13     22  Tennis no Ouji-sama  7.90
## 15     24        School Rumble  7.94
##                                                  Genres         English.name
## 1      Action, Adventure, Comedy, Drama, Sci-Fi, Space         Cowboy Bebop
## 3    Action, Sci-Fi, Adventure, Comedy, Drama, Shounen               Trigun
## 4  Action, Mystery, Police, Supernatural, Drama, Magic   Witch Hunter Robin
## 7         Comedy, Drama, Josei, Romance, Slice of Life     Honey and Clover
## 13             Action, Comedy, Sports, School, Shounen The Prince of Tennis
## 15                    Comedy, Romance, School, Shounen        School Rumble
##    Type Episodes                        Aired   Premiered
## 1    TV       26  Apr 3, 1998 to Apr 24, 1999 Spring 1998
## 3    TV       26  Apr 1, 1998 to Sep 30, 1998 Spring 1998
## 4    TV       26  Jul 2, 2002 to Dec 24, 2002 Summer 2002
## 7    TV       24 Apr 15, 2005 to Sep 27, 2005 Spring 2005
## 13   TV      178 Oct 10, 2001 to Mar 23, 2005   Fall 2001
## 15   TV       26  Oct 5, 2004 to Mar 29, 2005   Fall 2004
##                                                                                    Producers
## 1                                                                              Bandai Visual
## 3                                                                       Victor Entertainment
## 4                                      TV Tokyo, Bandai Visual, Dentsu, Victor Entertainment
## 7                                                                   Genco, Fuji TV, Shueisha
## 13                                                          Production I.G, Nihon Ad Systems
## 15 TV Tokyo, Sotsu, Marvelous, Starchild Records, Media Factory, DAX Production, Studio Jack
##                               Licensors      Studios   Source        Duration
## 1      Funimation, Bandai Entertainment      Sunrise Original 24 min. per ep.
## 3  Funimation, Geneon Entertainment USA     Madhouse    Manga 24 min. per ep.
## 4      Funimation, Bandai Entertainment      Sunrise Original 25 min. per ep.
## 7             VIZ Media, Discotek Media    J.C.Staff    Manga 23 min. per ep.
## 13                            VIZ Media   Trans Arts    Manga 22 min. per ep.
## 15                           Funimation Studio Comet    Manga 23 min. per ep.
##                            Rating Ranked Popularity Members Favorites Watching
## 1  R - 17+ (violence & profanity)     28         39 1251960     61971   105808
## 3       PG-13 - Teens 13 or older    266        201  558913     12944    29113
## 4       PG-13 - Teens 13 or older   2481       1467   94683       587     4300
## 7       PG-13 - Teens 13 or older    468        687  214499      4101    11909
## 13      PG-13 - Teens 13 or older    675       1039  141832      3124    11235
## 15      PG-13 - Teens 13 or older    625        514  275464      5137    12277
##    Completed On.Hold Dropped Plan.to.Watch
## 1     718161   71513   26678        329800
## 3     343492   25465   13925        146918
## 4      46165    5121    5378         33719
## 7      81145   11901   11026         98518
## 13     76881   12905   12516         28295
## 15    157789   12856   13491         79051
```

```
summary(train)
```

```
##      MAL_ID           Name               Score          Genres
##  Min.   :    1.0   Length:976         Min.   :4.720   Length:976
##  1st Qu.:  956.8   Class :character   1st Qu.:6.840   Class :character
##  Median : 6105.5   Mode  :character   Median :7.260   Mode  :character
##  Mean   :10356.0                      Mean   :7.276
##  3rd Qu.:18548.0                      3rd Qu.:7.702
##  Max.   :32214.0                      Max.   :9.190
##  English.name          Type             Episodes         Aired
##  Length:976         Length:976         Min.   :  3.00   Length:976
##  Class :character   Class :character   1st Qu.: 12.00   Class :character
##  Mode  :character   Mode  :character   Median : 13.00   Mode  :character
##                                        Mean   : 24.27
##                                        3rd Qu.: 25.00
##                                        Max.   :500.00
##   Premiered          Producers          Licensors           Studios
##  Length:976         Length:976         Length:976         Length:976
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     Source            Duration            Rating              Ranked
##  Length:976         Length:976         Length:976         Min.   :    1
##  Class :character   Class :character   Class :character   1st Qu.: 1012
##  Mode  :character   Mode  :character   Mode  :character   Median : 2520
##                                                           Mean   : 2907
##                                                           3rd Qu.: 4286
##                                                           Max.   :10803
##    Popularity         Members          Favorites          Watching
##  Min.   :    1.0   Min.   :    763   Min.   :     0.0   Min.   :    23
##  1st Qu.:  512.5   1st Qu.:  43240   1st Qu.:   140.5   1st Qu.:  1969
##  Median : 1258.5   Median : 114871   Median :   560.5   Median :  6126
##  Mean   : 1779.3   Mean   : 239974   Mean   :  4486.2   Mean   : 12994
##  3rd Qu.: 2438.8   3rd Qu.: 278572   3rd Qu.:  2186.5   3rd Qu.: 13484
##  Max.   :11065.0   Max.   :2589552   Max.   :183914.0   Max.   :362124
##    Completed         On.Hold          Dropped         Plan.to.Watch
##  Min.   :    300   Min.   :    27   Min.   :    94   Min.   :   146
##  1st Qu.:  22721   1st Qu.:  1697   1st Qu.:  1858   1st Qu.: 12417
##  Median :  62792   Median :  4255   Median :  5485   Median : 30107
##  Mean   : 160033   Mean   :  7827   Mean   :  8782   Mean   : 50338
##  3rd Qu.: 174432   3rd Qu.:  8623   3rd Qu.: 11333   3rd Qu.: 65370
##  Max.   :2182587   Max.   :109707   Max.   :148408   Max.   :425531
```

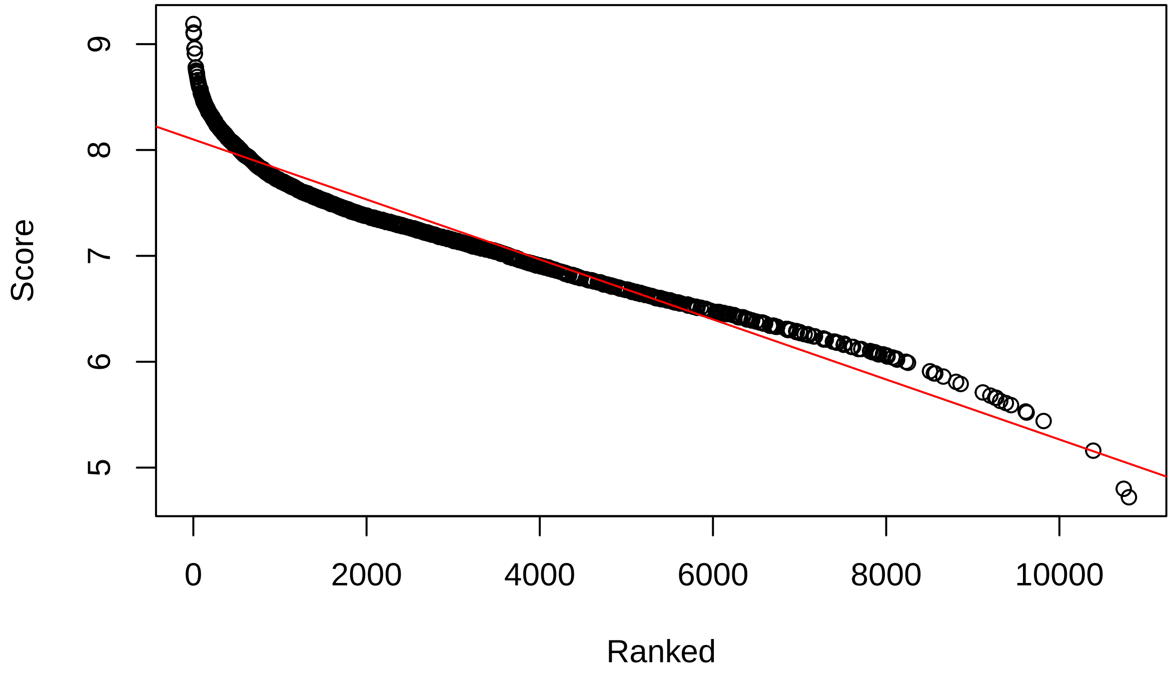
```
mean(train$Score)
```

```
## [1] 7.275809
```

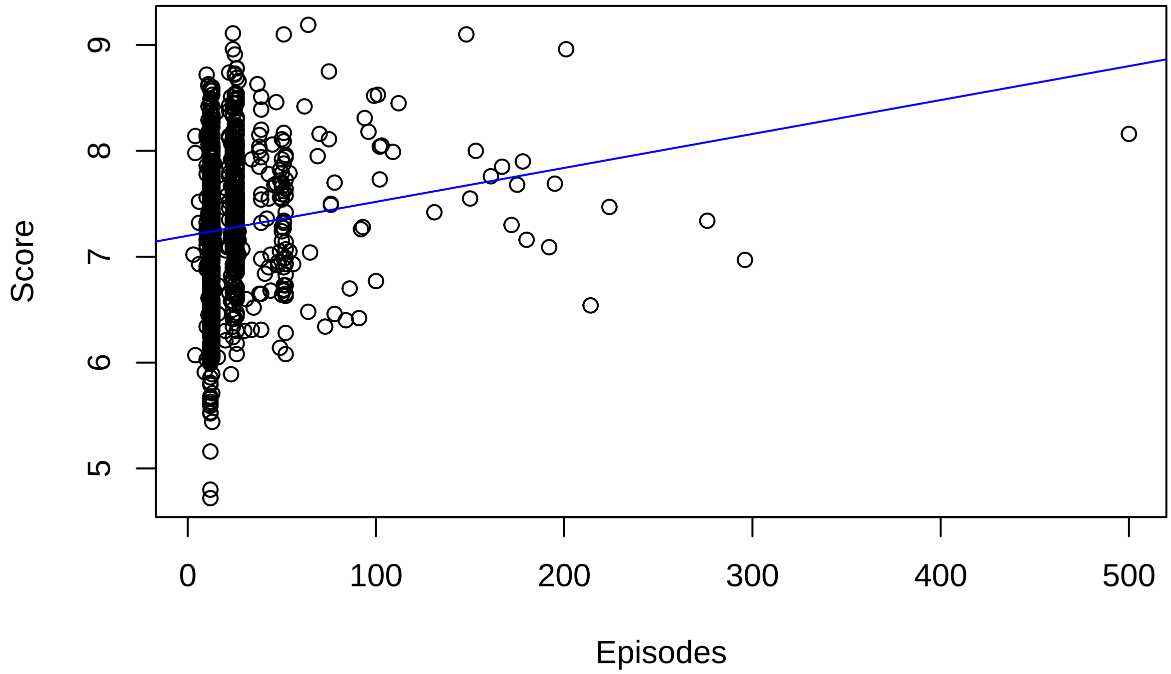
```
median(train$Score)
```

```
## [1] 7.26
```

```
plot(train$Score~train$Ranked,xlab="Ranked",ylab="Score")
abline(lm(train$Score~train$Ranked),col="red")
```



```
plot(train$Score~train$Episodes,xlab="Episodes",ylab="Score")
abline(lm(train$Score~train$Episodes),col="blue")
```



This chunk will

- Build a simple linear model of the data
- outputs the summary

```
lm1 <- lm(Score~Ranked,data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = Score ~ Ranked, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31804 -0.12011 -0.06364  0.07201  1.09077
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.100e+00  9.165e-03   883.7   <2e-16 ***
## Ranked      -2.834e-04  2.493e-06  -113.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1753 on 974 degrees of freedom
## Multiple R-squared:  0.9299, Adjusted R-squared:  0.9298
## F-statistic: 1.292e+04 on 1 and 974 DF,  p-value: < 2.2e-16
```

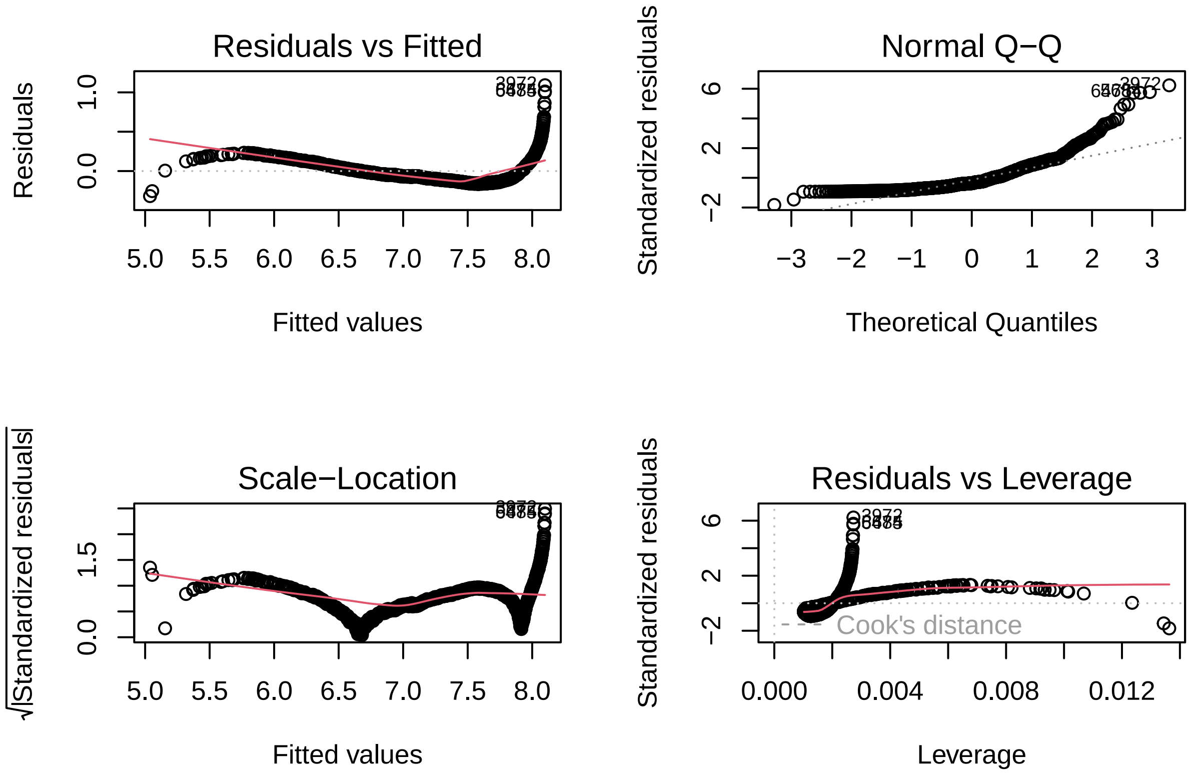
Summary of Simple Linear Model

The summary function in R has outputted a number of things.

- A **formula** that shows that we are modelling the score as a function of Rank
- **Residuals** that show the difference between what the model predicted and the actual value of **y**
- **Coefficients**
- The **Estimates** where the intercept tells us the value when all other features are 0. For the other features, the estimates give us the expected change in the response due to a unit change in the feature.
- **Standard Error** which allows us to construct marginal confidence intervals for the estimate of that particular feature.
- **t-value** which tells us about how far our estimated parameter is from the hypothesized 0 value.
- The **p-value** for the individual coefficient, which is the level of marginal significance within a statistical hypothesis test, representing the probability of the occurrence of a given event.
- The Residual Standard Error which gives the standard deviation of the residuals, and tells us about how large the prediction error is.
- **Multiple and Adjusted $\mathbf{R^2}$** which tell us what proportion of the variance is explained by out model
- **F-Stats** Which is the ratio of two variances

The next chunk will output various residuals plots

```
par(mfrow=c(2,2))
plot(lm1)
```



Explanation of Residual plot for the Simple Linear Model

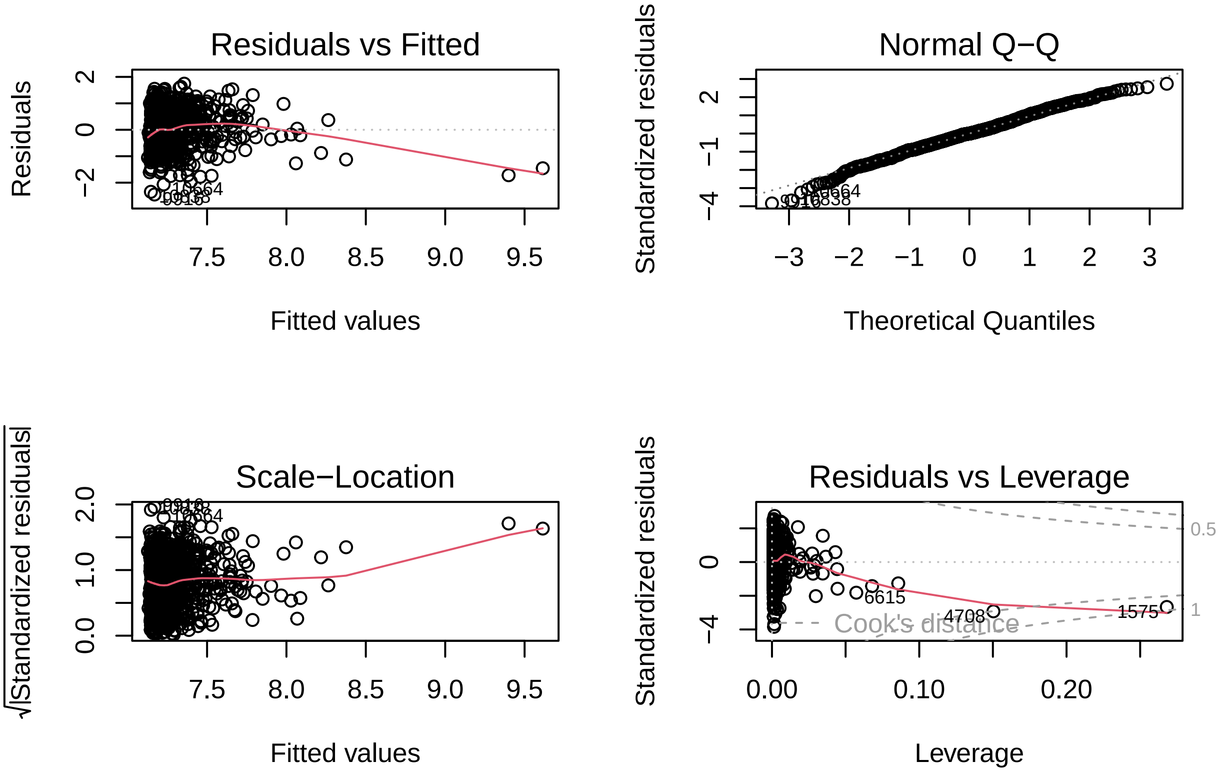
For starters, the residuals max is relatively low. This can be verified by the fact that the original plot of the linear model showed a very strong linear trend. Also, in a residual plot, we want there to be a spread of values above and below the line that are close to even. In this residual plot however, many fall exactly on that lne which shows an almost direct causation relationship between ranking and Score, which implies that Score is directly derived from the Rank.

Next, we create a multiple linear model using Number of Episodes, Rank, and target Score

```
lm2 <- lm(Score~Dropped+Episodes,data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = Score ~ Dropped + Episodes, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44889 -0.40969  0.00062  0.40777  1.74373
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.118e+00  2.809e-02 253.346  < 2e-16 ***
## Dropped     1.342e-05  1.849e-06   7.256 8.13e-13 ***
## Episodes    1.659e-03  6.854e-04   2.421   0.0157 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6378 on 973 degrees of freedom
## Multiple R-squared:  0.0732, Adjusted R-squared:  0.0713
## F-statistic: 38.43 on 2 and 973 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm2)
```



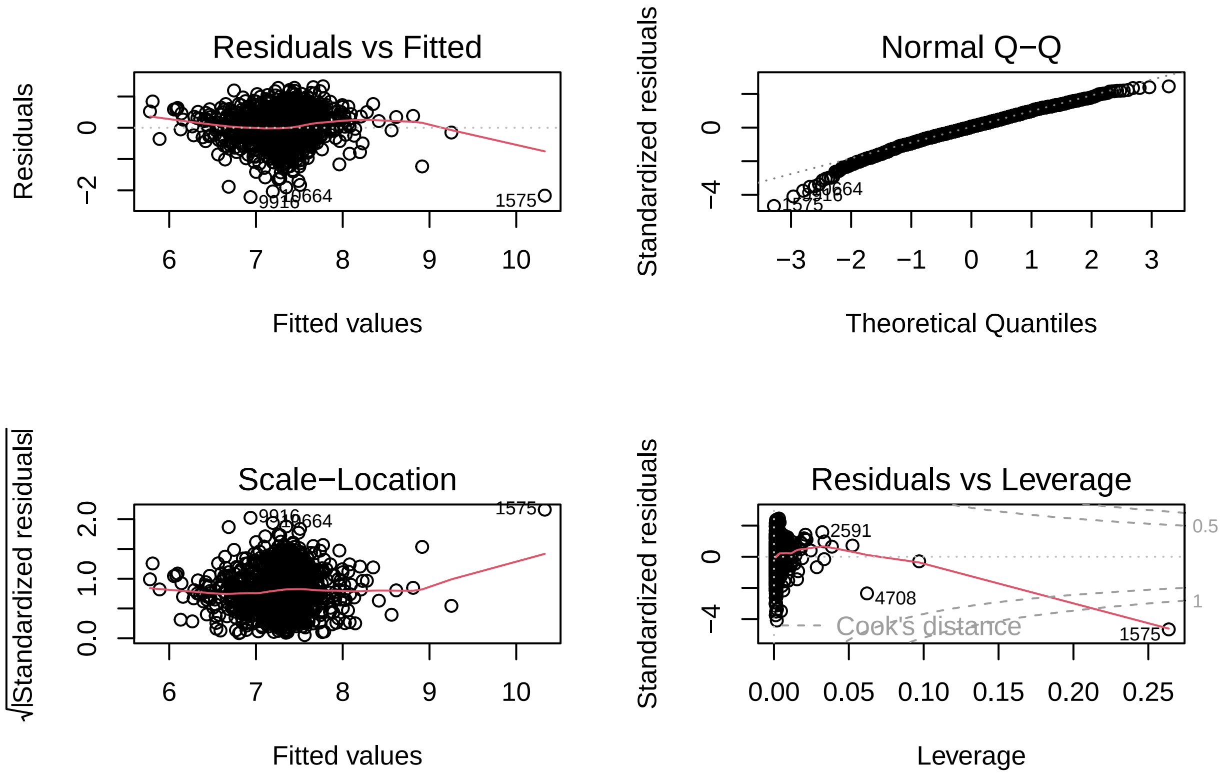
Next, we create a second multiple linear model using Popularity, Rank, and target Score

```
lm3 <- lm(Score~Watching+Popularity,data=train)
summary(lm3)
```

```
##
## Call:
## lm(formula = Score ~ Watching + Popularity, data = train)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.21746 -0.31713  0.02945  0.36631  1.32679
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.439e+00  3.118e-02 238.621   <2e-16 ***
## Watching     7.987e-06  8.173e-07   9.773   <2e-16 ***
## Popularity  -1.501e-04  1.099e-05 -13.661   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5419 on 973 degrees of freedom
## Multiple R-squared:  0.3311, Adjusted R-squared:  0.3297
## F-statistic: 240.8 on 2 and 973 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm3)
```



Next, let us use our models and make perdictions

```
pred1 <- predict(lm1,newdata=test)
pred2 <- predict(lm2,newdata=test)
pred3 <- predict(lm3,newdata=test)
```

```r
cor1 <- cor(pred1,test$Score)
mse1 <- mean((pred1 - test$Score)^2)
rmse1 <- sqrt(mse1)
print(paste("Correlation of the first model: ", cor1))
```

```
## [1] "Correlation of the first model:  0.971267531662069"
```

```r
print(paste("mse of the first model: ", mse1))
```

```
## [1] "mse of the first model:  0.0217860311525616"
```

```r
print(paste("rmse of the first model: ", rmse1))
```

```
## [1] "rmse of the first model:  0.14760091853563"
```

```r
cor2 <- cor(pred2,test$Score)
mse2 <- mean((pred2 - test$Score)^2)
rmse2 <- sqrt(mse2)
print(paste("Correlation of the second model: ", cor2))
```

```
## [1] "Correlation of the second model:  0.23700889131067"
```

```r
print(paste("mse of the second model: ", mse2))
```

```
## [1] "mse of the second model:  0.356885250337707"
```

```r
print(paste("rmse of the second model: ", rmse2))
```

```
## [1] "rmse of the second model:  0.597398736471468"
```

```r
cor3 <- cor(pred3,test$Score)
mse3 <- mean((pred3 - test$Score)^2)
rmse3 <- sqrt(mse3)
print(paste("Correlation of the third model: ", cor3))
```

```
## [1] "Correlation of the third model:  0.567895691688818"
```

```r
print(paste("mse of the third model: ", mse3))
```

```
## [1] "mse of the third model:  0.249705718964579"
```

```r
print(paste("rmse of the third model: ", rmse3))
```

```
## [1] "rmse of the third model:  0.499705632312244"
```

Analysis of models

The first model showed Score as a function of Rank. The second model showed Score as a function of number of existing episodes + how many people dropped the show. The third model showed Score as a function of user rated popularity + individuals currently watching the show.

Ultimately, the simple linear model performed best out of the 3. Despite the latter two providing a multivaried analysis using multiple attributes, the single best predictor was proven to be the ranking of each anime, as the plots and correlation showed than rank directly influences the score it received.

The first simple model had the greatest correlation sitting at 0.97, as compared to the second with 0.23 and third, 0.56. We want this number to be as close to -1/1 as possible.

The rmse of the first was also the least, which shows we are less off using the first model than the other two. The first model was off by an average of 0.14, while the other two were off by 0.59 and 0.49 respectively.

Ultimately, I believe the reason for this was simply the chose attributes. It was interesting though how the third model performed, which showed a semi-decent correlation with popularity, people currently watching, and Score.