Disney+ Data Analysis

Maxwell Guevarra, Mia Kobayashi, Eunice Tu

2022-12-15

Pre-processing/Data Clean-UP

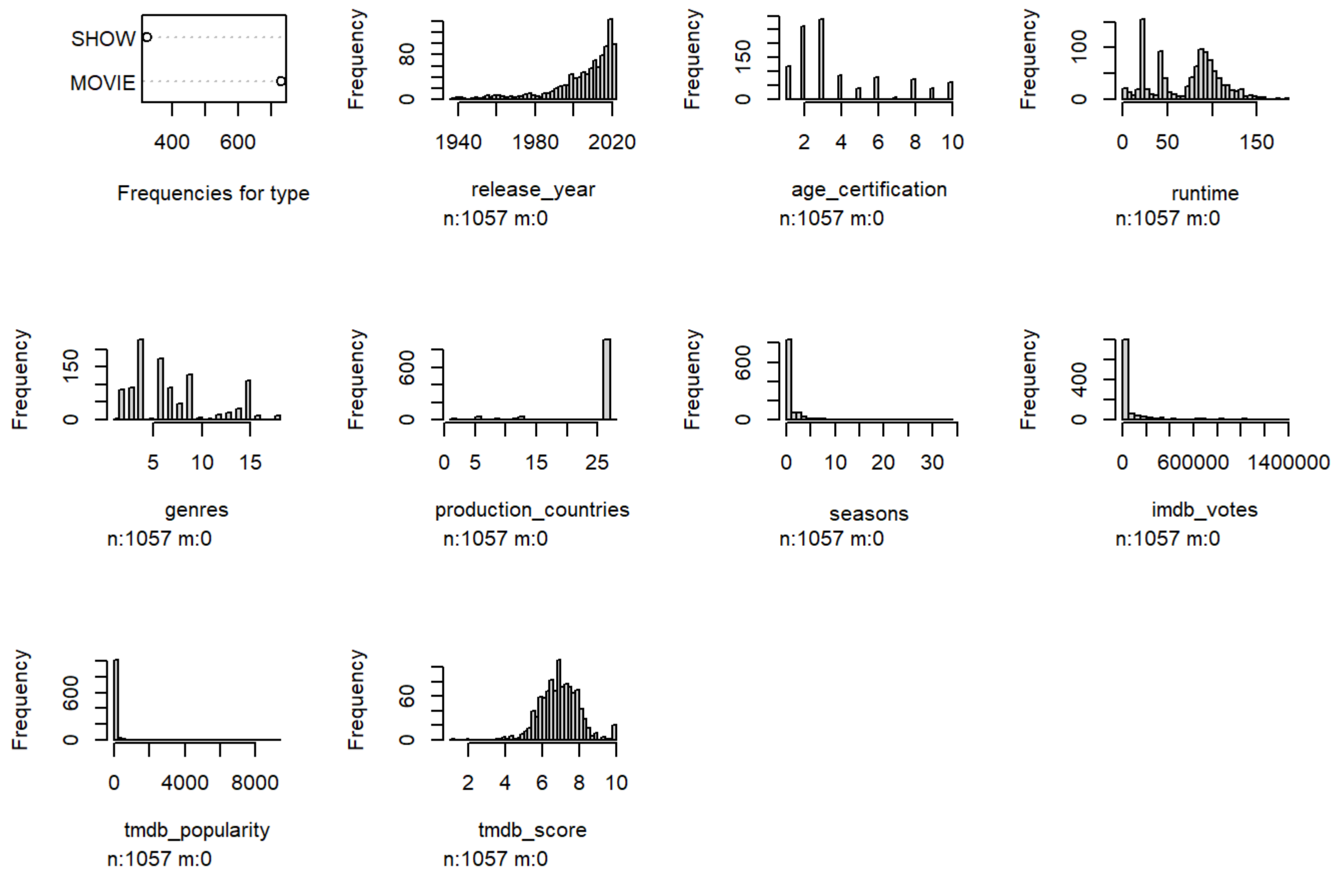
```
#initial data load
disney.data <- read.csv(file = "titles.csv")

#clean copy of data
df.c <- disney.data
df.c$seasons[is.na(df.c$seasons)] <- 0
df.c <- na.omit(df.c)
response <- df.c$imdb_score
df.c <- df.c[, -c(1, 2, 4, 11, 12)]

#Factorize "type"
df.c$type <- as.factor(df.c$type)

#Factorize "age_cert"
df.c$age_certification <- as.numeric(as.factor(df.c$age_certification))

#Factorize "genres"
df.c$genres <- sub("\\[", "", df.c$genres)
df.c$genres <- sub("\\]", "", df.c$genres)
df.c$genres <- str_split_fixed(df.c$genres, pattern = ",", n = 2)
df.c$genres <- df.c$genres[, -2]
df.c$genres <- as.numeric(as.factor(df.c$genres))

#Factorize "prod_country"
df.c$production_countries <- sub("\\[", "", df.c$production_countries)
df.c$production_countries <- sub("\\]", "", df.c$production_countries)
df.c$production_countries <- str_split_fixed(df.c$production_countries, pattern = ",", n = 2)
df.c$production_countries <- df.c$production_countries[, -2]
df.c$production_countries <- as.numeric(as.factor(df.c$production_countries))
```

Inital Variable Correlations and Graphs

```
cor(df.c[, unlist(lapply(df.c, is.numeric))])
```

```
##                     release_year age_certification     runtime       genres
## release_year         1.000000000        0.23836952 -0.22632056 -0.022035605
## age_certification    0.238369516        1.00000000 -0.58420825 -0.028922191
## runtime             -0.226320555       -0.58420825  1.00000000  0.146300689
## genres              -0.022035605       -0.02892219  0.14630069  1.000000000
## production_countries -0.097588939        0.08835386  0.02885275  0.025573989
## seasons              0.075786874        0.48625188 -0.46001410  0.002284336
## imdb_votes           0.004653356       -0.07877011  0.42728896  0.138165232
## tmdb_popularity      0.071617051        0.01329548  0.05028342 -0.010578456
## tmdb_score           0.232451997        0.29021623 -0.29939150  0.027181472
##                     production_countries      seasons   imdb_votes
## release_year                 -0.09758894  0.075786874  0.004653356
## age_certification             0.08835386  0.486251882 -0.078770112
## runtime                       0.02885275 -0.460014100  0.427288963
## genres                        0.02557399  0.002284336  0.138165232
## production_countries          1.00000000  0.032079020  0.093359067
## seasons                       0.03207902  1.000000000 -0.105954661
## imdb_votes                    0.09335907 -0.105954661  1.000000000
## tmdb_popularity               0.01106457  0.006234180  0.147706729
## tmdb_score                   -0.01389984  0.227082837  0.124162950
##                     tmdb_popularity  tmdb_score
## release_year             0.07161705  0.23245200
## age_certification        0.01329548  0.29021623
## runtime                  0.05028342 -0.29939150
## genres                  -0.01057846  0.02718147
## production_countries     0.01106457 -0.01389984
## seasons                  0.00623418  0.22708284
## imdb_votes               0.14770673  0.12416295
## tmdb_popularity          1.00000000  0.08273120
## tmdb_score               0.08273120  1.00000000
```

```
hist.data.frame(df.c)
```



Some Notes About the Data

- `tmdb_score` seems to be normally distributed
- As the years increase, the number of movies seem to be produced exponentially
- There appear to be groupings in the distribution for `runtime`
- There doesn’t appear to be any strong correlations between the variables

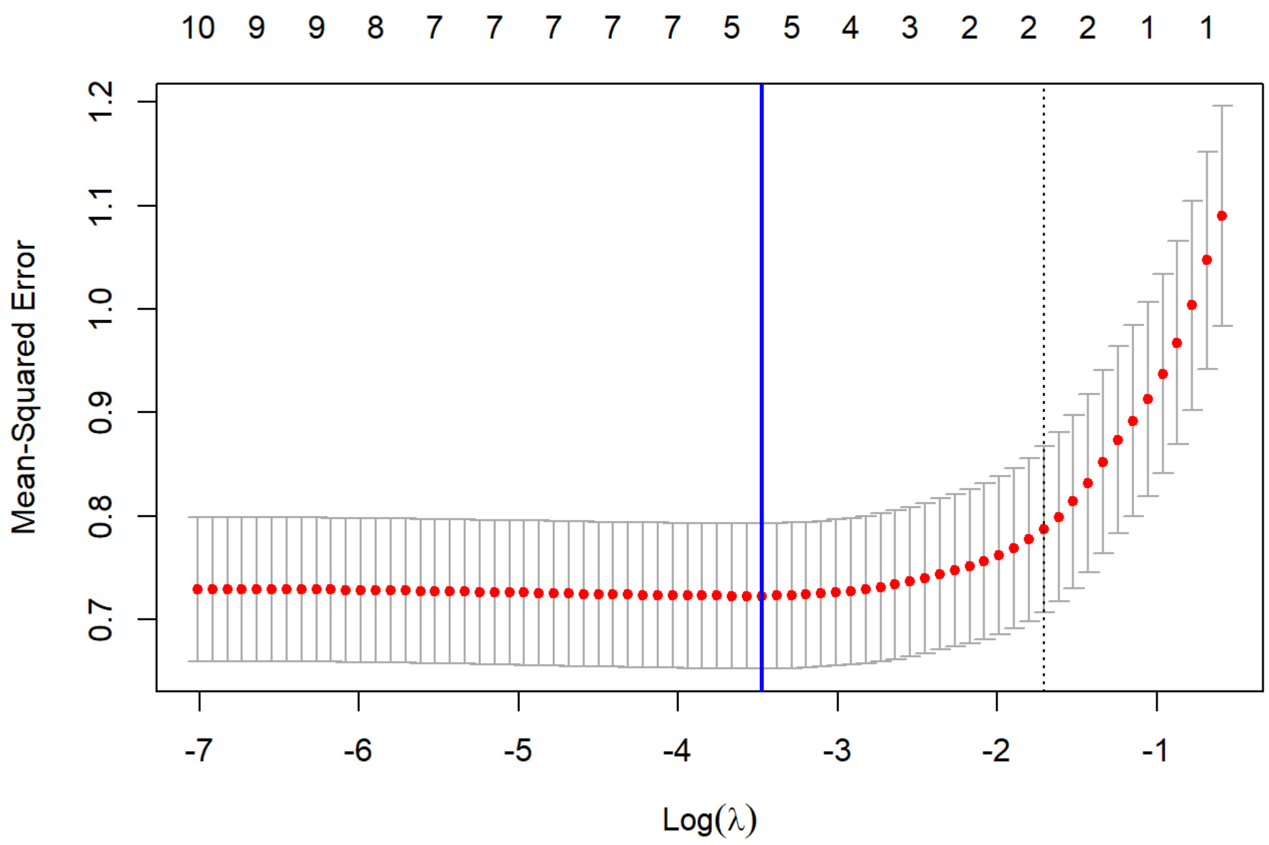
---

Prediction Analysis

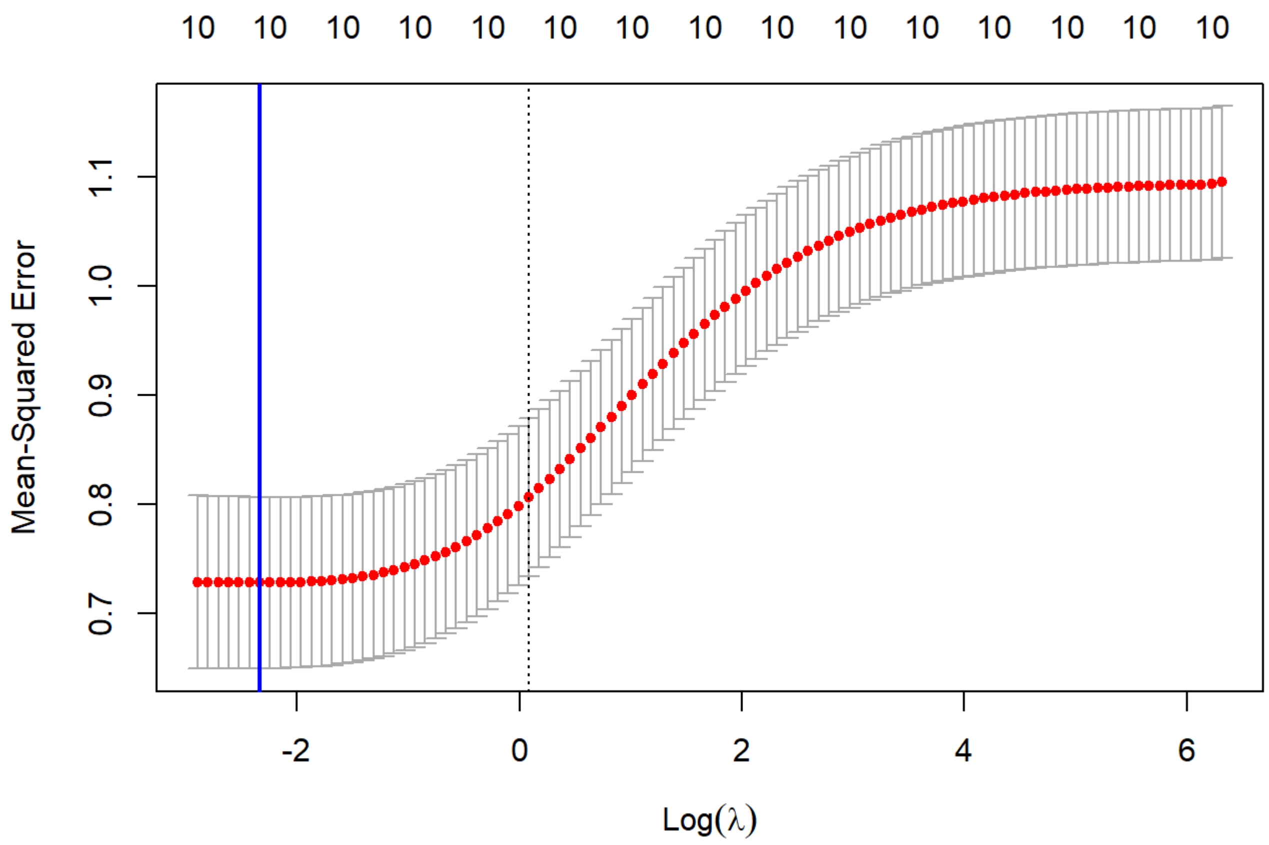
Initial Regression Models

```
xy.l <- inital.predict(df.c, response)
```

```
## [1] "Full OLS Model"
## 
## Call:
## lm(formula = response ~ ., data = dataset)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.4501 -0.4169  0.0550  0.4776  3.3924 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           1.725e+01  3.187e+00   5.412 7.73e-08 ***
## typeSHOW              3.121e-01  1.338e-01   2.333  0.01985 *  
## release_year         -7.407e-03  1.590e-03  -4.657 3.61e-06 ***
## age_certification    -1.465e-02  1.693e-02  -0.866  0.38687    
## runtime               2.783e-03  1.204e-03   2.312  0.02095 *  
## genres                2.330e-03  6.156e-03   0.378  0.70515    
## production_countries  6.324e-03  4.342e-03   1.457  0.14553    
## seasons               3.800e-02  1.400e-02   2.715  0.00674 ** 
## imdb_votes            1.282e-06  1.575e-07   8.137 1.15e-15 ***
## tmdb_popularity      -1.826e-05  8.191e-05  -0.223  0.82369    
## tmdb_score            5.227e-01  2.779e-02  18.808  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.8223 on 1046 degrees of freedom
## Multiple R-squared:  0.3989, Adjusted R-squared:  0.3931 
## F-statistic: 69.41 on 10 and 1046 DF,  p-value: < 2.2e-16
## 
## [1] "LASSO Regression:"
```



```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                               s0
## (Intercept)           1.339184e+01
## typeSHOW              1.119769e-02
## release_year         -5.236432e-03
## age_certification     .
## runtime               .
## genres                .
## production_countries  2.311802e-03
## seasons               3.223012e-02
## imdb_votes            1.228234e-06
## tmdb_popularity       .
## tmdb_score            5.035050e-01
## [1] "Ridge Regression:"
```


10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
Mean-Squared Error
0.7
0.8
0.9
1.0
1.1
-2
0
2
4
6
Log(λ)

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                s0
## (Intercept)           1.547607e+01
## typeSHOW              2.281683e-01
## release_year         -6.335750e-03
## age_certification    -5.116358e-03
## runtime               1.972299e-03
## genres                3.736192e-03
## production_countries  6.003543e-03
## seasons               3.711864e-02
## imdb_votes            1.224932e-06
## tmdb_popularity       1.908973e-06
## tmdb_score            4.759150e-01
## [1] "Comparing MSPE Values:"
##       LASSO     Ridge
## 1 0.6785527 0.6725153
## [1] "Comparing R-Squared Values:"
##         OLS     LASSO     Ridge
## 1 0.3988785 0.3904821 0.3959053
## [1] "Comparing Coefficient Values:"
##                               OLS         LASSO         Ridge
## (Intercept)           1.724867e+01  1.339184e+01  1.547607e+01
## typeSHOW              3.120501e-01  1.119769e-02  2.281683e-01
## release_year         -7.406963e-03 -5.236432e-03 -6.335750e-03
## age_certification    -1.465459e-02  0.000000e+00 -5.116358e-03
## runtime               2.783102e-03  0.000000e+00  1.972299e-03
## genres                2.330015e-03  0.000000e+00  3.736192e-03
## production_countries  6.324472e-03  2.311802e-03  6.003543e-03
## seasons               3.799731e-02  3.223012e-02  3.711864e-02
## imdb_votes            1.281701e-06  1.228234e-06  1.224932e-06
## tmdb_popularity      -1.825500e-05  0.000000e+00  1.908973e-06
## tmdb_score            5.226641e-01  5.035050e-01  4.759150e-01
```

```
#Saving the Training and Test Sets
x <- xy.l[[1]]
y <- xy.l[[2]]
x.train <- xy.l[[3]]
y.train <- xy.l[[4]]
x.test <- xy.l[[5]]
y.test <- xy.l[[6]]
```

Notes About the Models

- When comparing the LASSO and Ridge models, both sets of metrics are relatively the same but the Ridge model is technically better with a smaller MSPE and larger R^2
- The LASSO model does label some variables as "unimportant" with a coefficient of 0
- The Full OLS model has the best R^2 value between the three models
- The coefficients between the OLS and Ridge models are fairly close outside of a couple of variables; LASSO has the most coefficient shrinkage

- Based on the above observations, we might want to choose the Full OLS model; however we can use the variable selection from LASSO to fit another OLS model and compare metrics

OLS Model Variants

```
#Full OLS model
ols.full <- lm(response ~ ., data = df.c)

#OLS with selected variables
ols.select <- lm(response ~ release_year + seasons + imdb_votes + tmdb_score, data = df.c)

#OLS with LASSO selected variables
ols.las <- ols.LAS <- lm(response ~ type + release_year + production_countries + seasons + imdb_
votes + tmdb_score, data = df.c)
```

- The “Selected” OLS model consists of variables that had a >2-star significance level based on the full model summary (p = 4)
- The “LASSO” OLS model consists of variables selected by the LASSO regression process from above (p = 6)

Comparing Metrics Across OLS Models

```
#OLS Full
olstr.full <- lm(y.train ~ ., data.frame(x.train))
metrics.full <- metrics(olstr.full, data.frame(x.test), y.test)
ols.full.rsq <- metrics.full[[1]]
ols.full.rmse <- metrics.full[[2]]
ols.full.aic <- metrics.full[[3]]
ols.full.bic <- metrics.full[[4]]
met.list.full <- c(ols.full.rsq, ols.full.rmse, ols.full.aic, ols.full.bic)


#OLS Selected
olstr.select <- lm(y.train ~ release_year + seasons + imdb_votes + tmdb_score, data.frame(x.trai
n))
metrics.select <- metrics(olstr.select, data.frame(x.test), y.test)
ols.select.rsq <- metrics.select[[1]]
ols.select.rmse <- metrics.select[[2]]
ols.select.aic <- metrics.select[[3]]
ols.select.bic <- metrics.select[[4]]
met.list.sel <- c(R2 = ols.select.rsq, ols.select.rmse, ols.select.aic, ols.select.bic)

#OLS LASSO
olstr.las <- lm(y.train ~ typeSHOW + release_year + production_countries + seasons + imdb_votes
+ tmdb_score, data.frame(x.train))
metrics.las <- metrics(olstr.las, data.frame(x.test), y.test)
ols.las.rsq <- metrics.las[[1]]
ols.las.rmse <- metrics.las[[2]]
ols.las.aic <- metrics.las[[3]]
ols.las.bic <- metrics.las[[4]]
met.list.las <- c(R2 = ols.las.rsq, ols.las.rmse, ols.las.aic, ols.las.bic)

met.list.all <- data.frame(Full = met.list.full, Select = met.list.sel, LASSO = met.list.las)
row.names(met.list.all) <- c("R2", "RMSE", "AIC", "BIC")
met.list.all
```

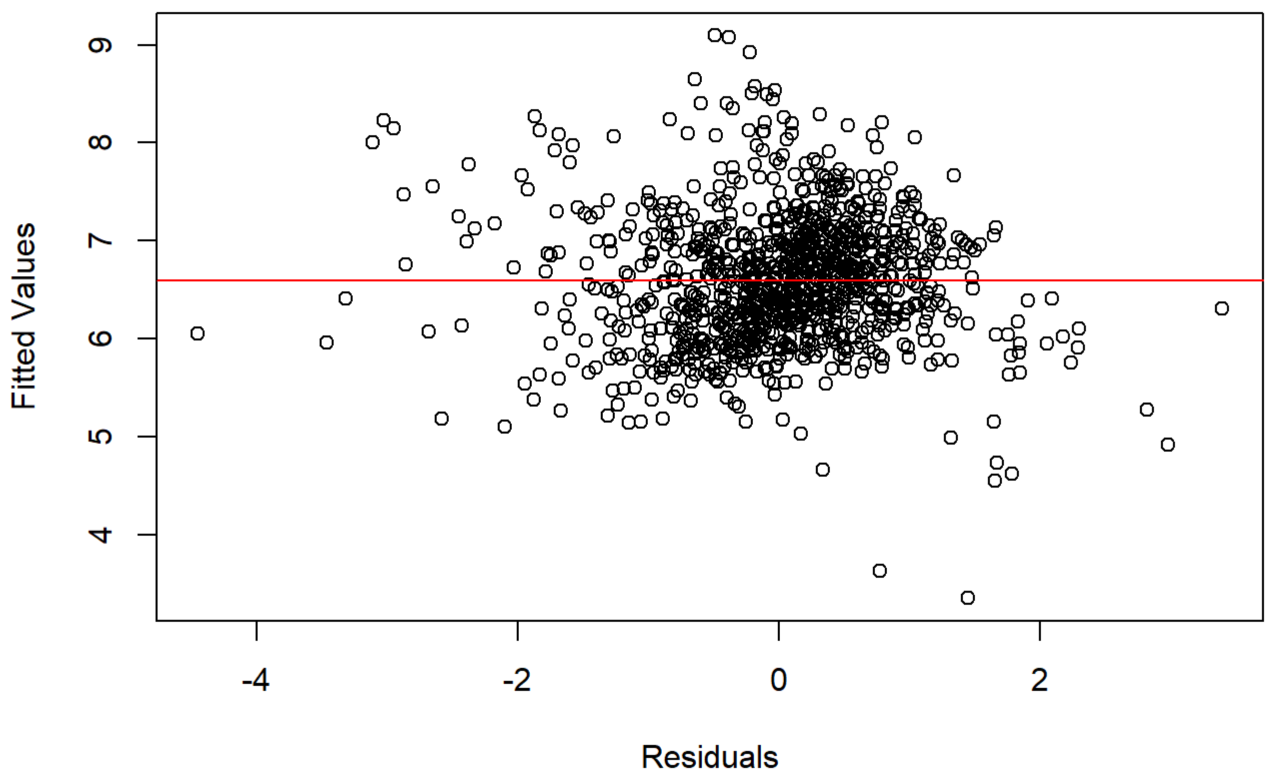
```
##              Full       Select        LASSO
## R2      0.4228548    0.4247226    0.4235757
## RMSE    0.8104045    0.8094893    0.8100578
## AIC  1333.6110434 1323.5664783 1326.5569402
## BIC  1384.8401988 1349.1810560 1360.7097105
```

- The “Selected” OLS model had the best metrics across all three models
  - Has the greatest R^2 value – more variability is explained by this model compared to the others
  - Has the best RMSE, AIC, and BIC values (lowest values)
    - The standard deviations of the residuals are low, and the quality of the model is the best out of the three based on the AIC/BIC scores
- Based on these comparisons, the “Selected” OLS model should be the final predictive model

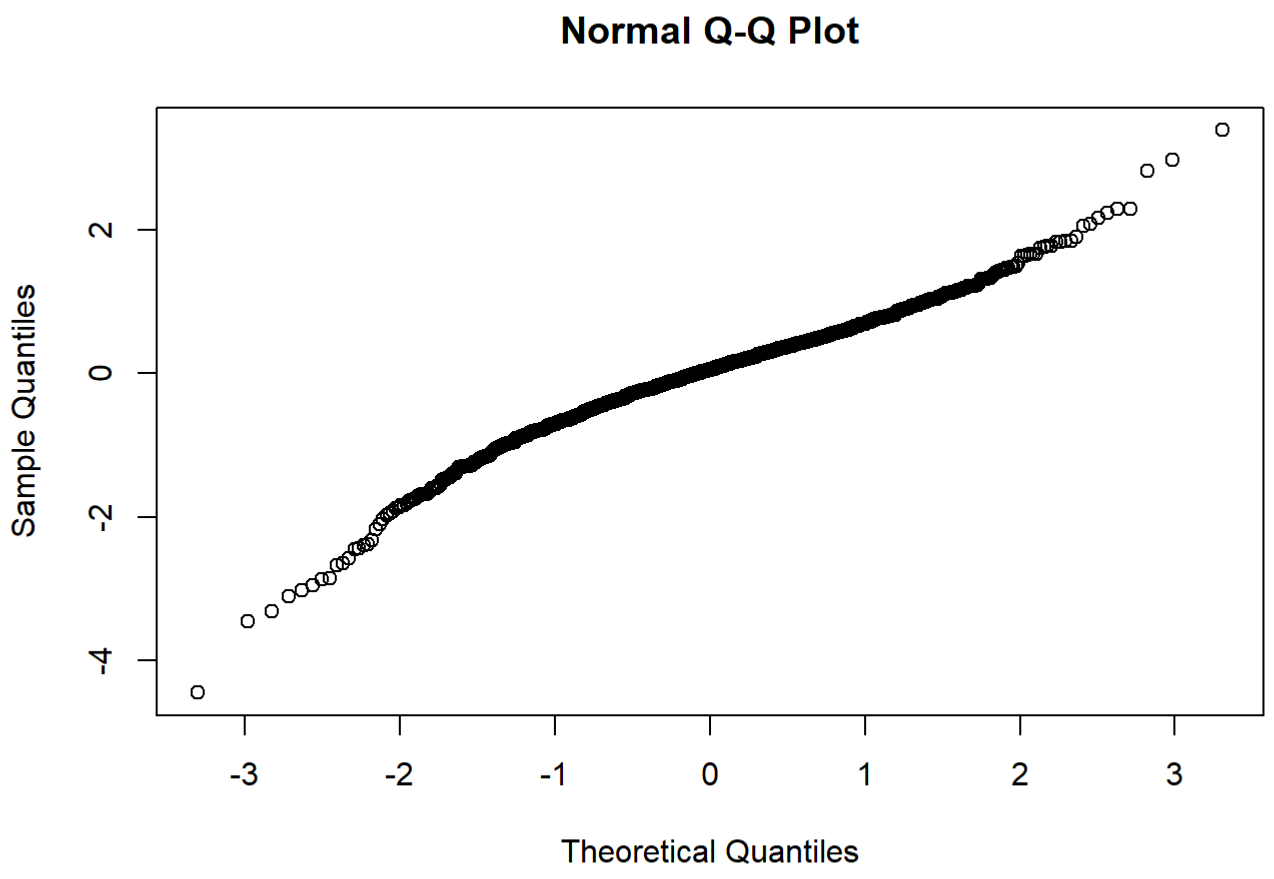
Explanatory Analysis

Diagnostics

```
diagnostics(ols.full)
```



```
## [1] "Correlation:"
## [1] 1.388303e-16
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 85.488, df = 10, p-value = 4.183e-14
```



```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  residuals and normal.sample
## D = 0.076472, p-value = 0.004908
## alternative hypothesis: two-sided
```

Linearity Assumption:

- The data seems to be dispersed randomly around the mean residual value of 0
- The correlation value is practically 0
- Can assume linearity in the data

Homoskedasticity Assumption:

- The spreads of the data are not exactly even
- Furthermore, the Breush-Pagan test indicates that heteroskedasticity is present
- Cannot assume homoskedasticity in the data

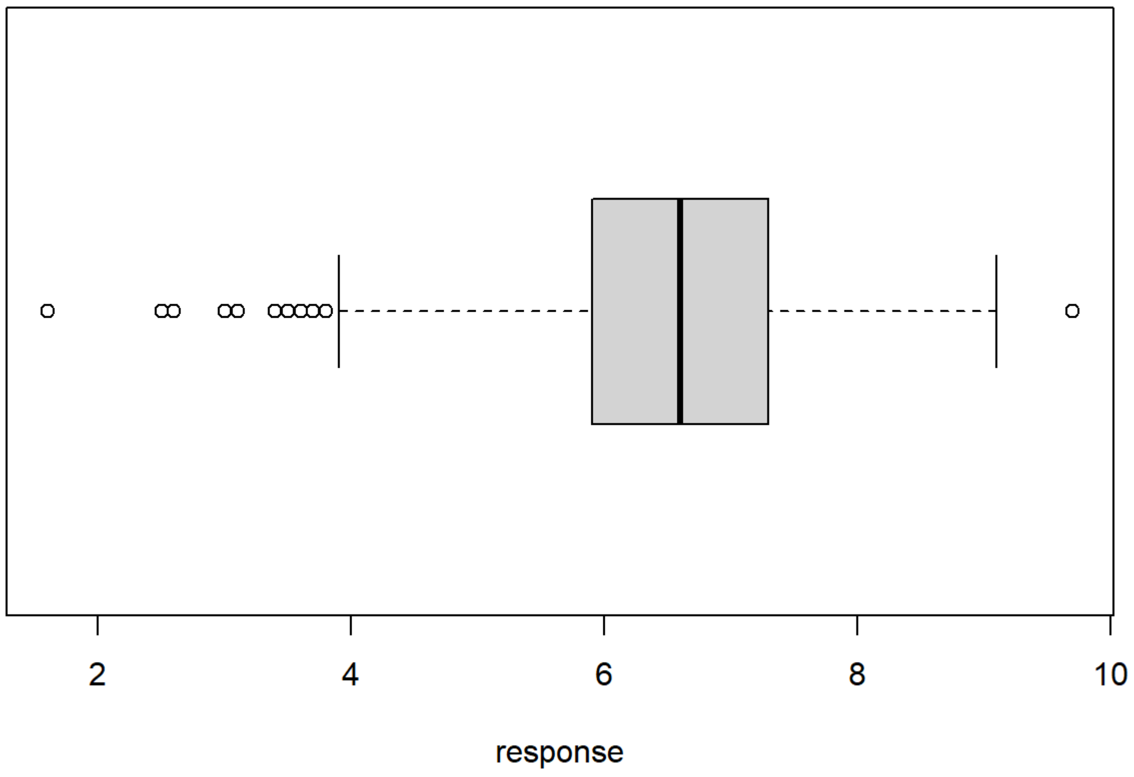
Normality Assumption:

- Hard to tell from the QQ-Plot if the trend is exactly linear
- KS-test results in a small p-value, thus we cannot assume normality for the data

Outlier Identification

Since the data did not initially pass some of the assumptions above, let’s try and identify any outliers and/or points of leverage

```
visual.out(response)
```



With a simple visualization, can see that there are about 12 outliers in the dataset.

We can use Cook’s Distance to remove any points of leverage from the dataset

```
list.co <- cooks.dist(ols.full, df.c)
```

```
## Joining, by = c("type", "release_year", "age_certification", "runtime",
## "genres", "production_countries", "seasons", "imdb_votes", "tmdb_popularity",
## "tmdb_score")
```

```
df.co <- list.co[[1]]
response.co <- list.co[[2]]
length(response) - length(response.co)
```

```
## [1] 50
```

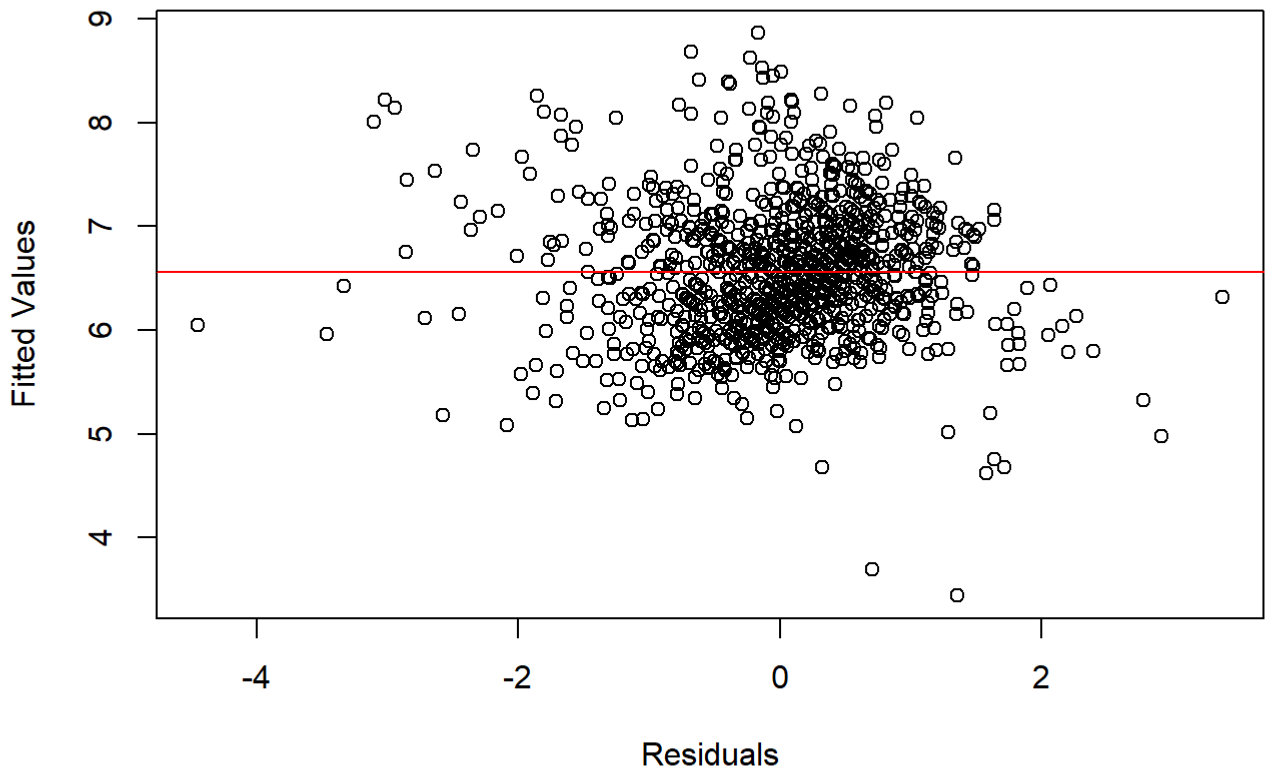
50 points of leverage were removed. Will now see how the modified dataset operates under diagnostics.

Diagnostics on New Dataset (Cook's)

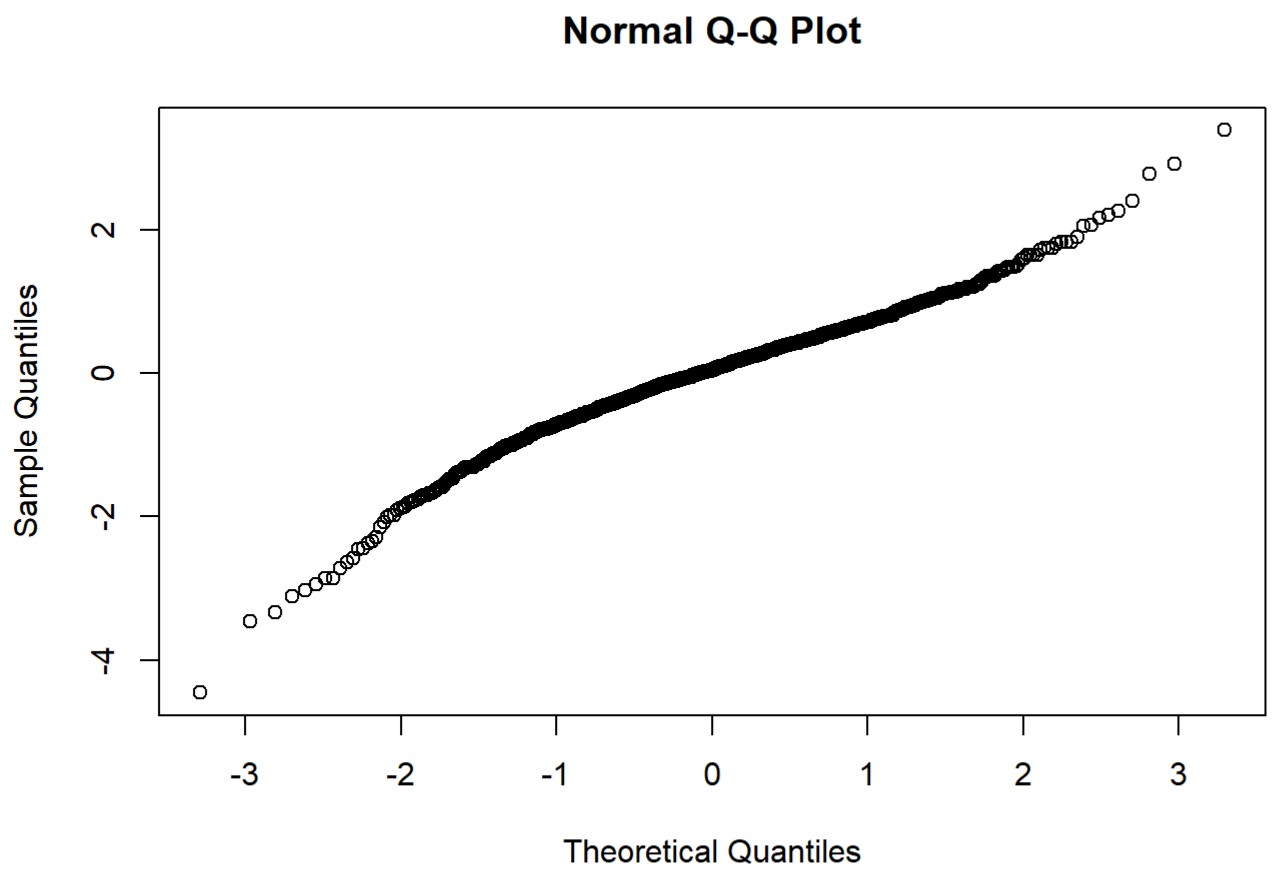
```
ols.co <- lm(response.co ~ ., df.co)
summary(ols.co)
```

```
##
## Call:
## lm(formula = response.co ~ ., data = df.co)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4492 -0.4498  0.0444  0.5001  3.3819
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.107e+01  4.151e+00   2.667  0.00777 **
## typeSHOW              2.927e-01  1.378e-01   2.125  0.03384 *
## release_year         -4.311e-03  2.074e-03  -2.078  0.03793 *
## age_certification    -1.211e-02  1.740e-02  -0.696  0.48670
## runtime               2.813e-03  1.274e-03   2.209  0.02742 *
## genres                3.087e-03  6.451e-03   0.479  0.63235
## production_countries  6.561e-03  4.447e-03   1.476  0.14037
## seasons               3.881e-02  1.432e-02   2.709  0.00686 **
## imdb_votes            1.303e-06  1.687e-07   7.723 2.76e-14 ***
## tmdb_popularity      -2.594e-05  8.353e-05  -0.311  0.75616
## tmdb_score            5.128e-01  2.875e-02  17.835  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8372 on 996 degrees of freedom
## Multiple R-squared:  0.3885, Adjusted R-squared:  0.3823
## F-statistic: 63.27 on 10 and 996 DF,  p-value: < 2.2e-16
```

```
diagnostics(ols.co)
```



```
## [1] "Correlation:"
## [1] -8.357314e-16
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 74.415, df = 10, p-value = 6.184e-12
```

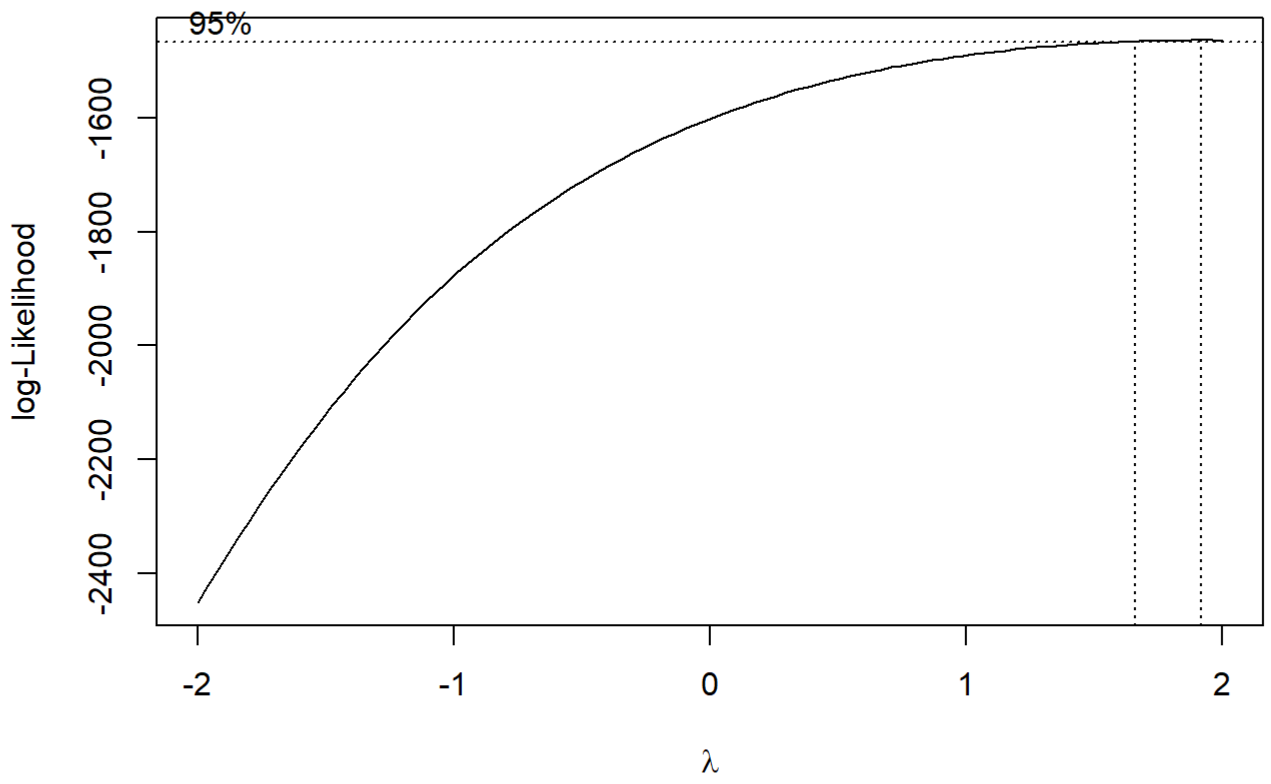


```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  residuals and normal.sample
## D = 0.042356, p-value = 0.329
## alternative hypothesis: two-sided
```

Looking at the spread of residuals across the fitted values, the spread isn't still entirely even – there seems to be a more dense cluster around zero. There is probably a difference in variation.
Modified dataset now passes normality assumption.

Boxcox Transformation

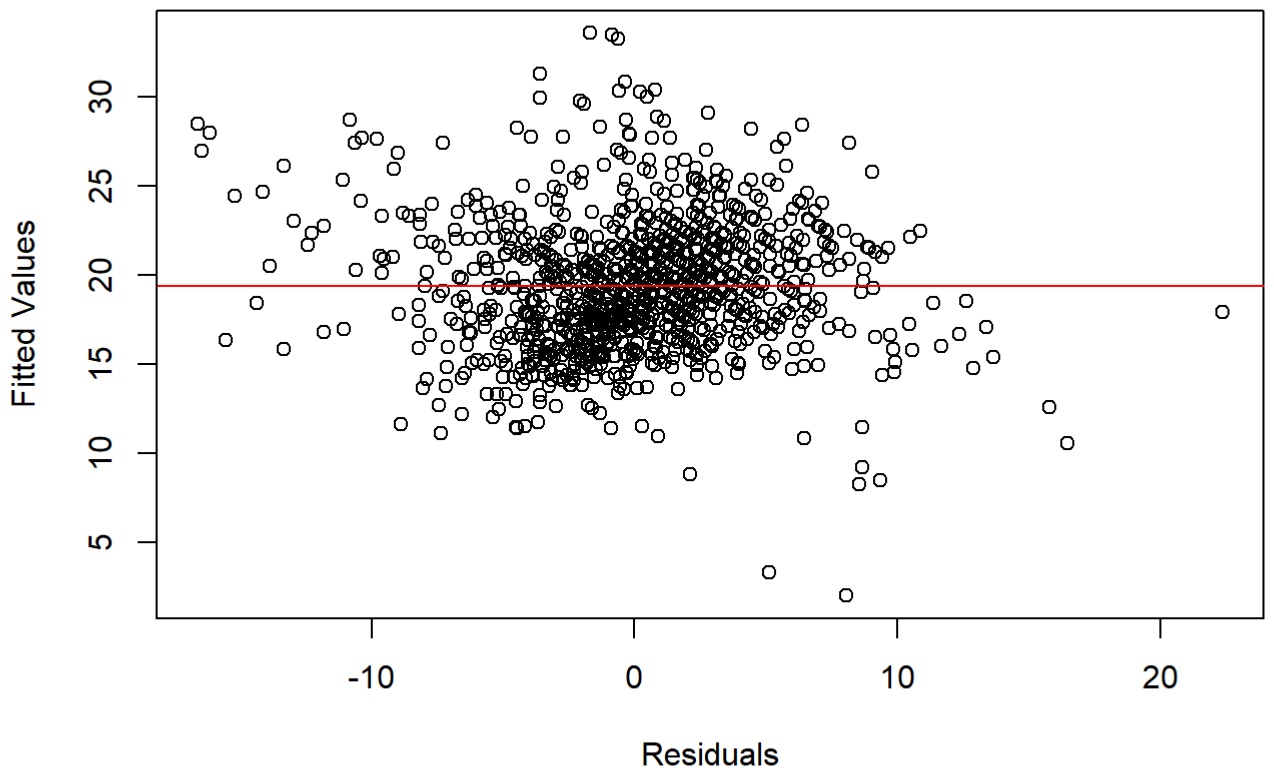
```
new.response <- bc.t(ols.full, response)
```



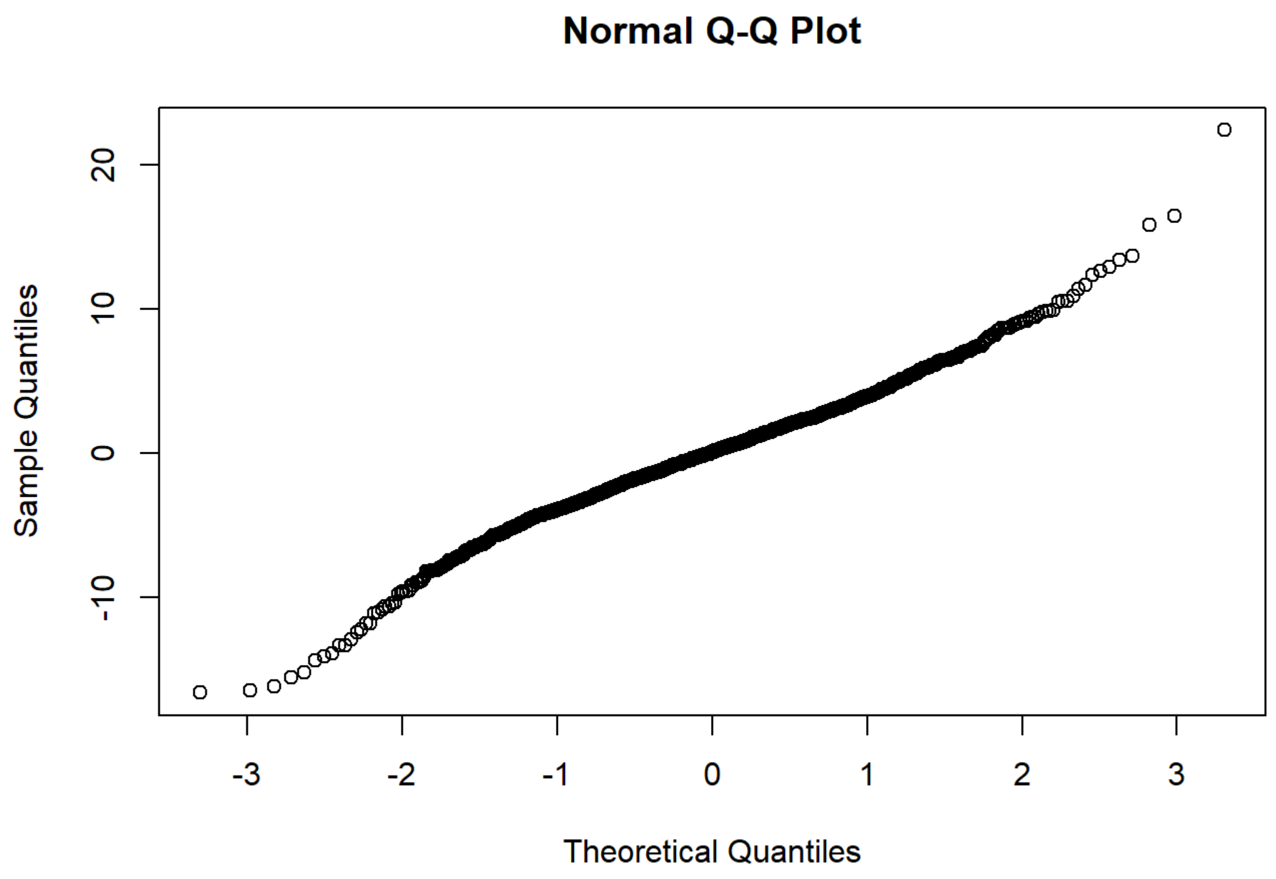
```
ols.bc <- lm(new.response ~ ., data = df.c)
```

Diagnostics with Boxcox transformation

```
diagnostics(ols.bc)
```



```
## [1] "Correlation:"
## [1] -1.683131e-16
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 123.39, df = 10, p-value < 2.2e-16
```



```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  residuals and normal.sample
## D = 0.040574, p-value = 0.366
## alternative hypothesis: two-sided
```

QQ-Plot:
- has some smaller residuals bottom right - still some larger residuals top right - some sway from central tendency, overall okay

Correlation is close to 0, passes normality assumption.
Looking at the spread of residuals across the fitted values, the spread now more even. Left outliers are obviously removed, maybe potentially still a far right outlier.

Due to similar diagnostic results, we would rather keep the points taken out by Cook’s Distance for our final explanatory model.

Choose Between Selected and LASSO

```
f.tests(ols.select, ols.las)
```

```
## Analysis of Variance Table
##
## Model 1: response ~ release_year + seasons + imdb_votes + tmdb_score
## Model 2: response ~ type + release_year + production_countries + seasons +
##     imdb_votes + tmdb_score
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1052 714.03
## 2   1050 711.76  2     2.277 1.6796  0.187
```

H_0: smaller model (OLS selected) is sufficient to explain the data
H_1: larger model (LASSO) is required to explain the variability in y

p-value from the F-test is > 0.05, the alpha value, and the F-statistic being close to zero indicates that we should fail to reject the null and keep the smaller OLS selected model to represent the Disney+ dataset. However these models are fitted on the full dataset which did not pass the normality assumption above. Therefore this test is invalid.

Selected vs LASSO OLS Models with Boxcox Data

```
ols.select.bc <- lm(new.response ~ release_year + seasons + imdb_votes + tmdb_score, df.c)
ols.las.bc <- lm(new.response ~ type + release_year + production_countries + seasons + imdb_vote
s + tmdb_score, df.c)

f.tests(ols.select.bc, ols.las.bc)
```

```
## Analysis of Variance Table
##
## Model 1: new.response ~ release_year + seasons + imdb_votes + tmdb_score
## Model 2: new.response ~ type + release_year + production_countries + seasons +
##     imdb_votes + tmdb_score
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1   1052 21265
## 2   1050 21174  2    90.725 2.2495  0.106
```

H_0: smaller model (OLS selected) is sufficient to explain the data
H_1: larger model (LASSO) is required to explain the variability in y

p-value from the F-test is also > 0.05, the alpha value, and the F-statistic being close to zero indicates that we should fail to reject the null and keep the smaller OLS selected model to represent the Disney+ dataset.

These models were made with the transformed data that did pass the normality assumption, thus this “selected” OLS model will be the final explanatory model.

```
summary(ols.select.bc)
```

```
##
## Call:
## lm(formula = new.response ~ release_year + seasons + imdb_votes +
##     tmdb_score, data = df.c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.5096  -2.5374   0.0608   2.5824  21.2951
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.378e+01  1.666e+01   4.429 1.05e-05 ***
## release_year -3.768e-02  8.401e-03  -4.486 8.07e-06 ***
## seasons       3.007e-01  6.314e-02   4.763 2.18e-06 ***
## imdb_votes    8.151e-06  7.330e-07  11.120  < 2e-16 ***
## tmdb_score    2.899e+00  1.397e-01  20.754  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.496 on 1052 degrees of freedom
## Multiple R-squared:  0.4035, Adjusted R-squared:  0.4012
## F-statistic: 177.9 on 4 and 1052 DF,  p-value: < 2.2e-16
```

```
#MSE
mean(summary(ols.select.bc)$residuals^2)
```

```
## [1] 20.11817
```

Notes about the model metrics

- Adjusted R^2: 0.4012 – Based on both the Adj. R^2 and regular R^2 values, about 40% of the varibaility in the data is explained by this model
- Each predictor is considered significant at the highest alpha level of 0.01