# Regression

2023-02-16

## Linear Models

Linear Regression is an approach for modelling the relationship between two or more quantitative variables. Linear predictor functions are used to estimate parameters from the data. This was the first type of regression to be studied and used for practical applications. Some of the strengths that come with LR are simplicity, low variance, and effectiveness on data that has a linear pattern. On the other hand, LR retains a high bias due to the linear shape.

## Analyzing a Data Set

The following steps use this data from Kaggle.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
spot_data <- read_csv("top_spot.csv")
```

```
## Rows: 11084 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Artist Name, Song Name, Peak Position (xTimes)
## dbl (6): Position, Days, Top 10 (xTimes), Peak Position, Peak Streams, Total...
##
## 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.
```

### Divide into 80/20 train/test

```
set.seed(5)
idx <- sample(1:nrow(spot_data), nrow(spot_data)*0.8, replace=FALSE)
train <- spot_data[idx,]
test <- spot_data[-idx,]
```

### Data Exploration

```
str(train)
```

```
## tibble [8,867 x 9] (S3: tbl_df/tbl/data.frame)
```

```
##  $ Position              : num [1:8867] 10937 2255 6859 1833 3797 ...
##  $ Artist Name           : chr [1:8867] "Jhené Aiko" "The Game" "Ski Mask The Slump God" "Lil Uzi Ver
##  $ Song Name             : chr [1:8867] "The Worst" "100" "Save Me Pt 2" "Silly Watch" ...
##  $ Days                  : num [1:8867] 1 152 3 42 18 2 68 8 11 172 ...
##  $ Top 10 (xTimes)       : num [1:8867] 0 0 0 6 0 0 17 0 1 0 ...
##  $ Peak Position         : num [1:8867] 130 55 164 3 61 148 2 15 6 14 ...
##  $ Peak Position (xTimes): chr [1:8867] "0" "0" "0" "(x1)" ...
##  $ Peak Streams          : num [1:8867] 81775 191050 322555 2591124 448542 ...
##  $ Total Streams         : num [1:8867] 81775 16091392 827994 23219511 5020300 ...
```

```
head(train)
```

```
## # A tibble: 6 x 9
##   Position `Artist Name`   Song ~1  Days Top 1~2 Peak ~3 Peak ~4 Peak ~5 Total~6
##      <dbl> <chr>           <chr>   <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>
## 1    10937 Jhené Aiko      The Wo~     1       0     130 0         81775  8.18e4
## 2     2255 The Game        100       152       0      55 0        191050  1.61e7
## 3     6859 Ski Mask The S~ Save M~     3       0     164 0        322555  8.28e5
## 4     1833 Lil Uzi Vert    Silly ~    42       6       3 (x1)    2591124  2.32e7
## 5     3797 Tory Lanez      Broke ~    18       0      61 0        448542  5.02e6
## 6     7239 Dan + Shay      Offici~     2       0     148 0        359154  6.87e5
## # ... with abbreviated variable names 1: `Song Name`, 2: `Top 10 (xTimes)`,
## #   3: `Peak Position`, 4: `Peak Position (xTimes)`, 5: `Peak Streams`,
## #   6: `Total Streams`
```

```
tail(train)
```

```
## # A tibble: 6 x 9
##   Position `Artist Name` Song Na~1  Days Top 1~2 Peak ~3 Peak ~4 Peak ~5 Total~6
##      <dbl> <chr>         <chr>     <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>
## 1     7504 Thomas Rhett  Blessed       2       0      81 0        390422  6.10e5
## 2     6854 Isaiah Rashad Claymore      3       0      71 0        366541  8.30e5
## 3       76 Polo G        RAPSTAR     519      84       1 (x22)   2691084  2.99e8
## 4     9799 Rich Brian    100 Degr~     1       0     198 0        243937  2.44e5
## 5     3384 Migos         Supastars    18       1       6 0       1278764  6.85e6
## 6     6472 Polo G        Through ~     5       0     177 0        216573  1.05e6
## # ... with abbreviated variable names 1: `Song Name`, 2: `Top 10 (xTimes)`,
## #   3: `Peak Position`, 4: `Peak Position (xTimes)`, 5: `Peak Streams`,
## #   6: `Total Streams`
```

```
names(train)
```

```
## [1] "Position"               "Artist Name"            "Song Name"
## [4] "Days"                   "Top 10 (xTimes)"        "Peak Position"
## [7] "Peak Position (xTimes)" "Peak Streams"           "Total Streams"
```

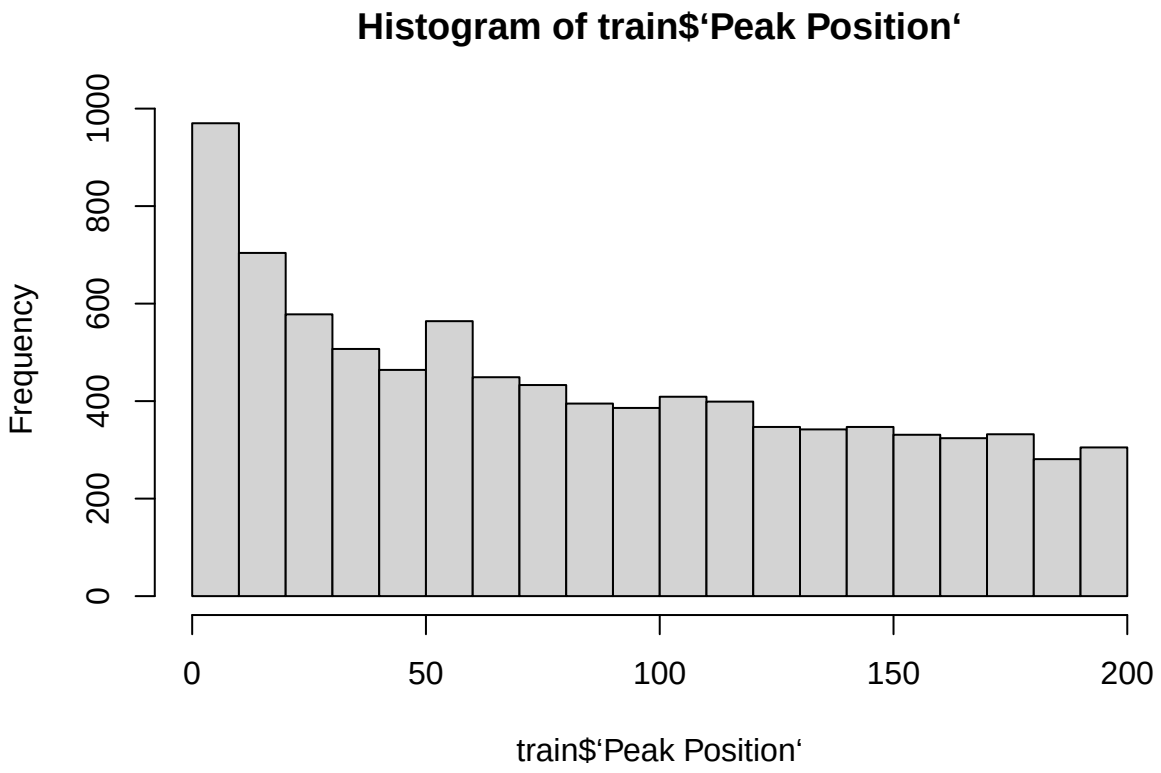
```
summary(train)
```

```
##     Position     Artist Name         Song Name             Days
##  Min.   :    1   Length:8867        Length:8867        Min.   :   1.00
##  1st Qu.: 2752   Class :character   Class :character   1st Qu.:   2.00
##  Median : 5558   Mode  :character   Mode  :character   Median :   7.00
##  Mean   : 5546                                         Mean   :  53.02
##  3rd Qu.: 8320                                         3rd Qu.:  40.00
##  Max.   :11084                                         Max.   :1995.00
##  Top 10 (xTimes)   Peak Position    Peak Position (xTimes)  Peak Streams
##  Min.   :  0.000   Min.   :  1.00   Length:8867            Min.   :  44323
```

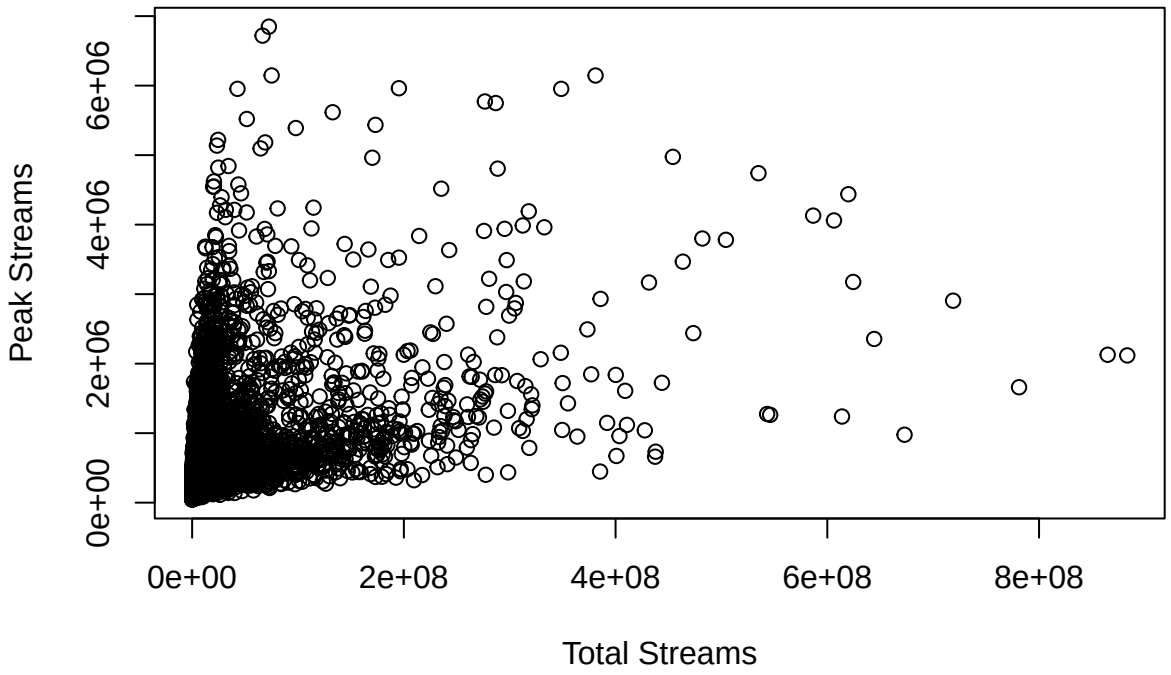
```
##  1st Qu.:  0.000   1st Qu.: 30.00   Class :character       1st Qu.: 242744
##  Median :  0.000   Median : 75.00   Mode  :character       Median : 349833
##  Mean   :  2.546   Mean   : 83.18                          Mean   : 548632
##  3rd Qu.:  0.000   3rd Qu.:132.00                          3rd Qu.: 590952
##  Max.   :302.000   Max.   :200.00                          Max.   :6847851
##  Total Streams
##  Min.   :    44323
##  1st Qu.:   380280
##  Median :  1691747
##  Mean   : 18061311
##  3rd Qu.: 10953018
##  Max.   :883369738
```

Creating Informative Graphs

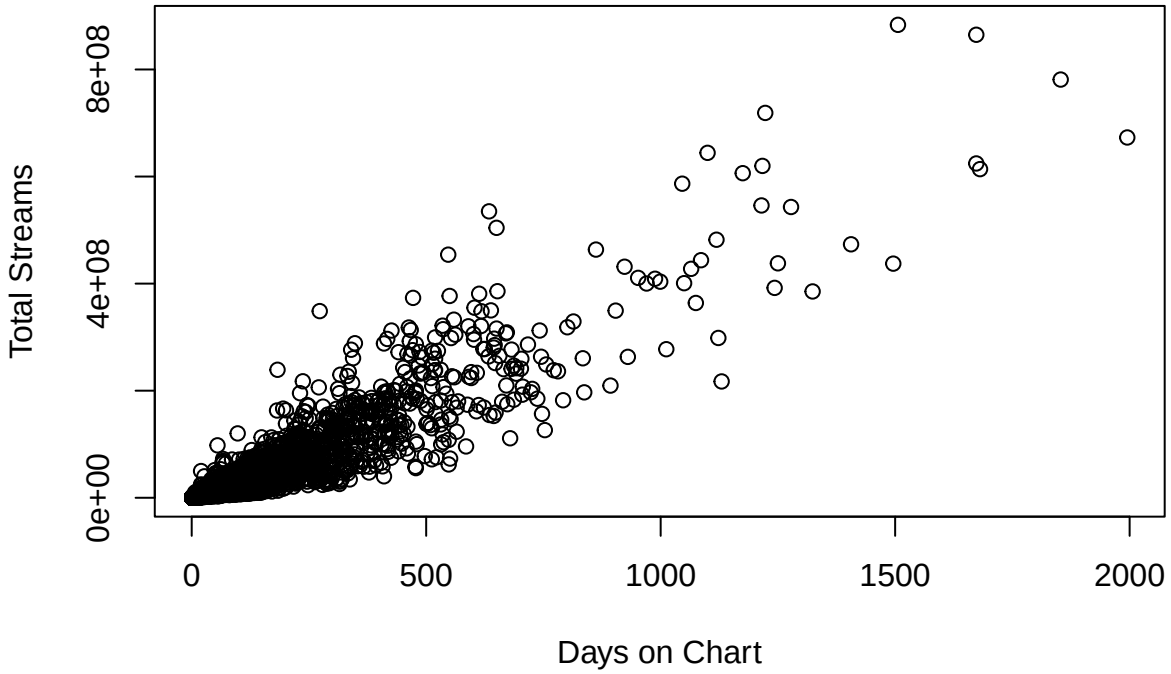
```
hist(train$`Peak Position`)
```




```
plot(train$`Total Streams`, train$`Peak Streams`, xlab="Total Streams", ylab="Peak Streams")
```




```
plot(train$Days, train$`Total Streams`, xlab="Days on Chart", ylab="Total Streams")
```




### Simple LR Model

```
lm1 <- lm(Days~`Total Streams`, data=train)
summary(lm1)
```

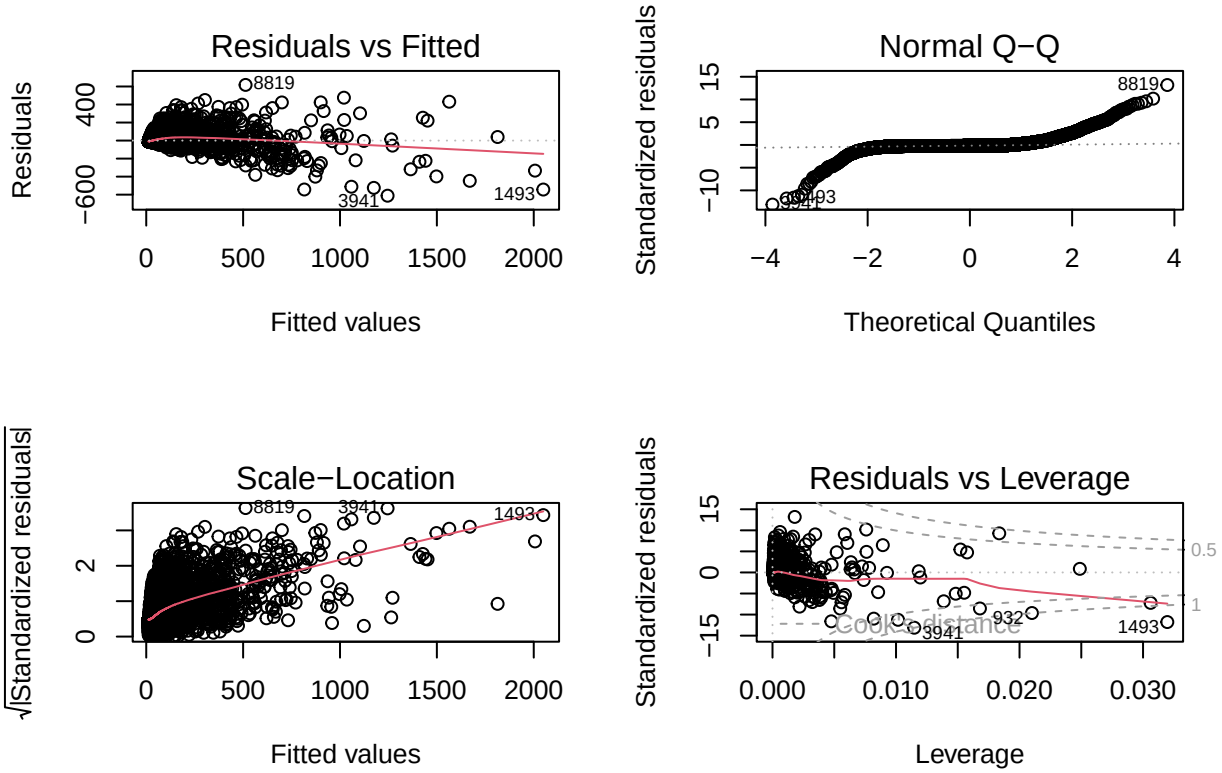
```
##
## Call:
## lm(formula = Days ~ `Total Streams`, data = train)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -611.55  -10.92  -10.04   -3.86  617.40
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.135e+01  5.277e-01   21.51   <2e-16 ***
## `Total Streams` 2.307e-06  9.670e-09  238.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.89 on 8865 degrees of freedom
## Multiple R-squared:  0.8652, Adjusted R-squared:  0.8652
## F-statistic: 5.691e+04 on 1 and 8865 DF,  p-value: < 2.2e-16
```

The summary shows components of the model that can determine how good of a fit it was. For example, The three asterisks at the end of the `Total Streams` row denotes this as a good predictor for our target. Furthermore, an R-squared value of 0.8652 measures the goodness of fit. This ranges from 0.0 to 1.0, indicating that the generated model is good, but can be improved for more accurate predictions.

**Plot residuals**

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




```
par(mfrow=c(1,1))
```

This residual plot shows four different views into our model performance. The first graph looks stable, indicating that the data fits the assumptions of regression well. Along with this, the Normal Q-Q graph describes how well this model fits the standard distribution. For most quantiles, the model is stable. However,

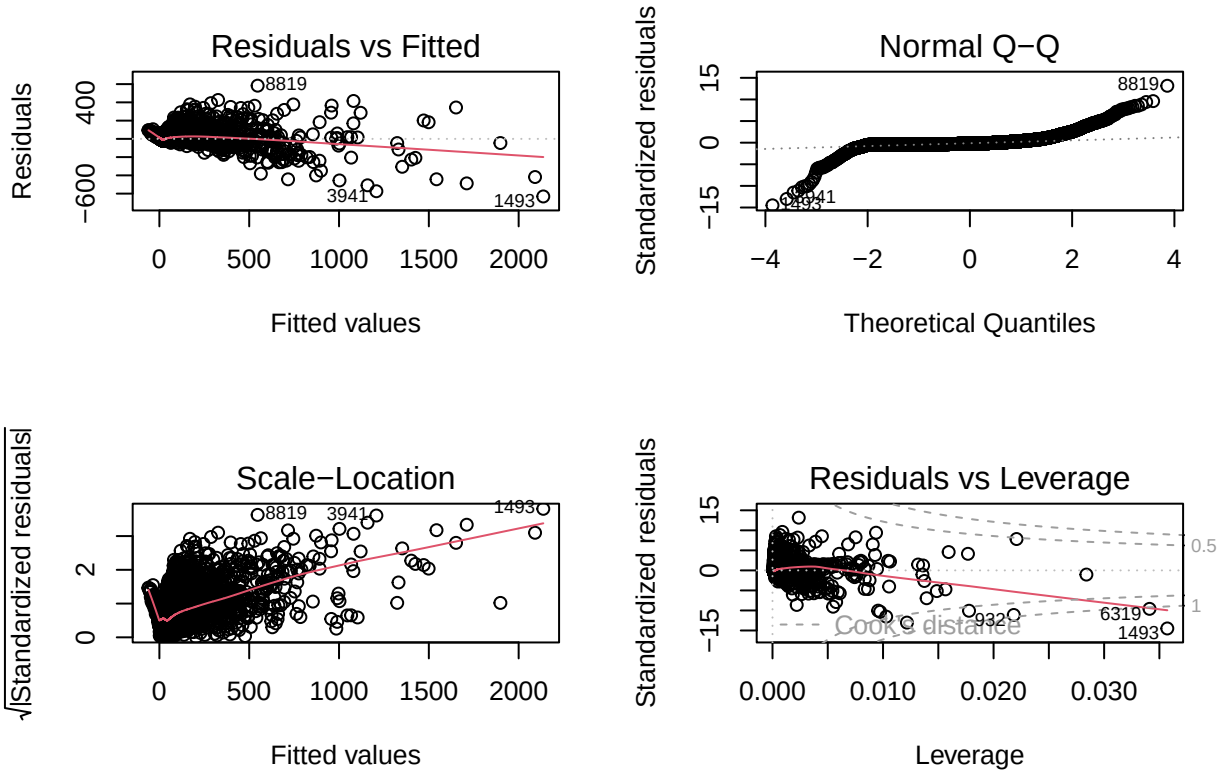
extremes on both sides have a large impact. The steep angle of the red line in the Scale-Location plot shows that residuals are not spread equally along the predictor range. This means that the assumption of equal variance is not met by the model. Finally, the Residuals vs Leverage graph is good for a data set of this size. The red line almost stays within bounds of the Cook's distance, however a few extreme outliers do have an impact on the entire model.

## Multiple LR Model

```
lm2 <- lm(Days~`Total Streams`+`Peak Streams`, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = Days ~ `Total Streams` + `Peak Streams`, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -631.14  -15.70  -10.57    4.07  582.98
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.379e+01  6.299e-01   37.77   <2e-16 ***
## `Total Streams`  2.459e-06  1.028e-08  239.22   <2e-16 ***
## `Peak Streams`  -2.767e-05  8.550e-07  -32.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.34 on 8864 degrees of freedom
## Multiple R-squared:  0.8795, Adjusted R-squared:  0.8794
## F-statistic: 3.234e+04 on 2 and 8864 DF,  p-value: < 2.2e-16
```

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



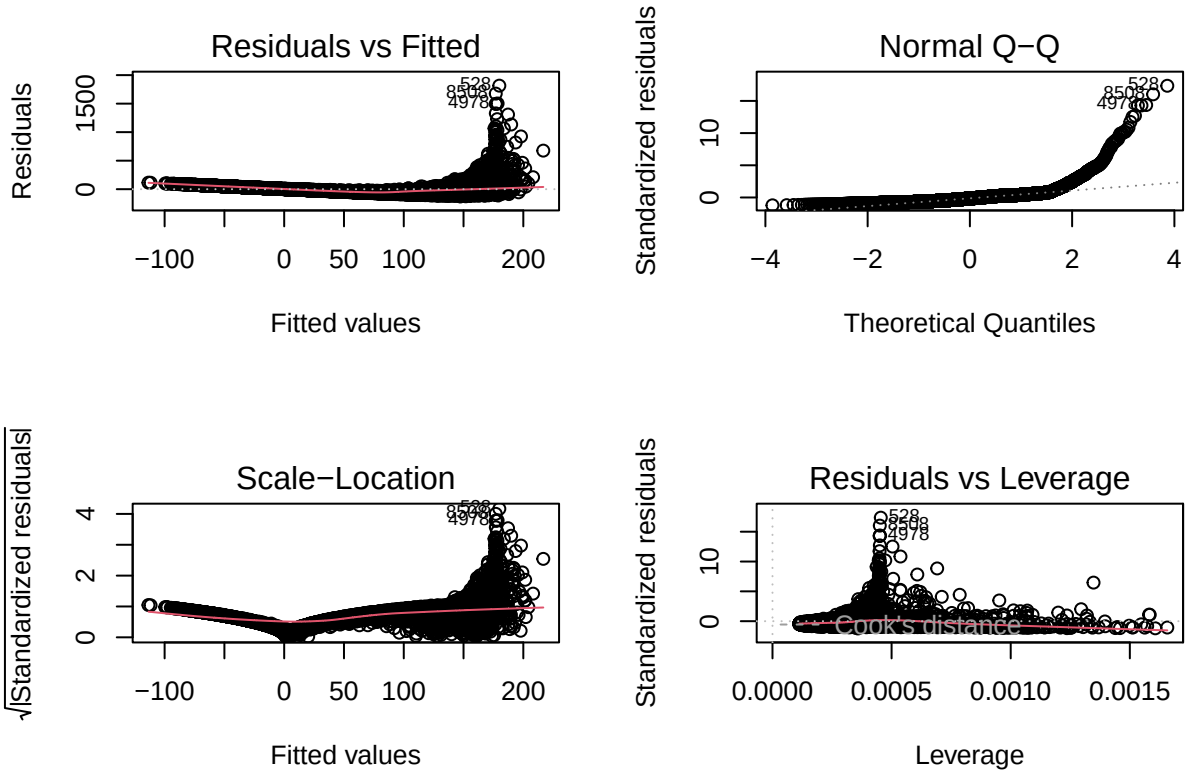

```
par(mfrow=c(1,1))
```

### Third LR Model

```
lm3 <- lm(Days~`Peak Position`+Position, data=train)
summary(lm3)
```

```
##
## Call:
## lm(formula = Days ~ `Peak Position` + Position, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -127.34  -56.40  -16.12   28.93 1815.07
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.766e+02  2.222e+00   79.48   <2e-16 ***
## `Peak Position`  4.383e-01  3.188e-02   13.75   <2e-16 ***
## Position        -2.885e-02  5.862e-04  -49.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 104.6 on 8864 degrees of freedom
## Multiple R-squared:  0.3289, Adjusted R-squared:  0.3287
## F-statistic:  2172 on 2 and 8864 DF,  p-value: < 2.2e-16
```

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




```
par(mfrow=c(1,1))
```

## Comparing Results

Out of all 3 models generated Model 2 is the best. This can be determined using the summary values and residual plots. All 3 R-squared values are as follows: 0.8652, 0.8794, 0.3287. Using multiple predictors improved the fit for model 2 but tanked it in model 3. This is also shown in the plotted residuals. The Residuals vs Leverage graph is clearly improved from model 1 to 2. On the other hand, model 3 has dramatically different plots due to it's poor predictors. The Normal Q-Q graph is more stable, however all other graphs have much worse fits.

## Predictions and Evaluations Using Metrics Correlation and MSE

### Predictions

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

### Evaluations

```
mse1 <- mean((pred1 - test$Days)^2)
cor1 <- cor(pred1, test$Days)
mse2 <- mean((pred2 - test$Days)^2)
cor2 <- cor(pred2, test$Days)
mse3 <- mean((pred3 - test$Days)^2)
cor3 <- cor(pred3, test$Days)
```

**Results**

```
print("MODEL 1:")
```

```
## [1] "MODEL 1:"
```

```
print(paste("MSE: ", mse1))
```

```
## [1] "MSE:  2908.25525917139"
```

```
print(paste("Correlation: ", cor1))
```

```
## [1] "Correlation:  0.920041556240522"
```

```
print("MODEL 2:")
```

```
## [1] "MODEL 2:"
```

```
print(paste("MSE: ", mse2))
```

```
## [1] "MSE:  2649.32971903947"
```

```
print(paste("Correlation: ", cor2))
```

```
## [1] "Correlation:  0.927445129618743"
```

```
print("MODEL 3:")
```

```
## [1] "MODEL 3:"
```

```
print(paste("MSE: ", mse3))
```

```
## [1] "MSE:  13021.1334132218"
```

```
print(paste("Correlation: ", cor3))
```

```
## [1] "Correlation:  0.559882145451058"
```

These results match what is expected of the models. Model 2 is still the best, with the lowest MSE and highest correlation. I think the predictors chosen for model 2 had the highest impact on the final results. This can be further improved by cleaning the extreme outliers in the data set.