MLB Salary Prediction with Multiple Linear Regression in R

Sean O'Malley

4/2/2017

ISLR Notes: Multiple Linear Regression in R

- Instead of fitting a separate simple linear regression model for each predictor, a better approach is to extend the simple linear regression model so that it can directly accomodate multiple predictors. We can do this by giving each predictor a separate slope coefficient in a single model
- When interpreting results, specifically coefficients, its helpful to interpret them within the business problem. For example a coefficient of 0.189 on radio advertising implies that for every 1000 dollars we spend on radio we increase sales by 189.
- However, there is a difference in coefficients when moved from simple to multiple linear regression. In SLR, the slope represents the average effect of a $1,000 increase in newspaper, ignoring TV and radio. In contrast, MLR, the coefficient for newspapers represents the average effect of increasing newspaper spending by $1,000 while holding TV and radio fixed.
- Interpreting Results
  - R^2 : Measures how close the data are to the fitted regression line. It is also known as the coefficient of determination.
  - **F-Statistic** : Probability that the null hypothesis for the full model is true, given that all of the regression coefficiants are zero. The larger the f-statistic, the more evidence rejecting the null.
  - **P-Value** : The probability of obtaining a result equal to or "more extreme" than what is actually observed, when the null hypothesis is true. In frequentist inference, the p-value is widely used in statistical hypothesis testing, specifically in null hypothesis significance testing.
  - **Coeficient** : The slope of the linear relationship between the criterion available and the part of a predictor variable that is independent of all other perdictor variables.

Questions:

What kind of questions can be answered by MLR? Give examples. * Multiple Linear Regression is best used to solve questions that need to find how to best predict a numeric value as influenced by other related numeric indicators. The build of a regression mathematical model is to determine how strongly a set of numeric variables helps determine the numeric value of a response variable. This model can then be used to predict future response variable inputs. * Some great examples of multiple linear regression model are: + In baseball, using slugging percentage, at bats, RBI's and games played to determine a players on base percentage + In medicine, using weeks of gestation, mother's weight, mother's height, household income, and baby's height to predict the birthweight of the baby.

Compare/contrast simple linear regression and multiple linear regression * Simple linear regression uses one numeric predictor variable to determine the value of the response variable. There is one coefficient involved in determining the value of incremental units given the output of the simple linear regression model. The model operates without understanding of the presence of any other varaibles that could be at play. Multiple linear regression uses multiple predictor variables to determine the linear prediction of the response variable. Essentially, MLR is used to explain the relationship between one continuous dependent variable and two or more independent variables. Both models are similar in the class of statistical outputs to determine model efficiency and each use a linear output across points to best predict the trajectory of the response varaible given the independent varaible(s).

What is (multi)collinearity? What are the consequences of collinearity in regression? * Multicollinearity is defined as a phenomenon in which two or more predictor variables in a multiple regression model are highly correlated, meaning that one can be linearly predicted from the others with a substantial degree of accuracy. The problem is multicollinearity is that the results are unstable parameter estimates which makes it very difficult to assess the effect of independent variables on dependent varaibles.

- I recently encountered an example of this in my last assignment. I attempted to predict income per capita using a Boston housing dataset, but ran into the issue of multi-collinearity with factors such as tax bracket, and lower class percentage. These factors were highly correlated, however were redundant and took weight away from the other, less trivial, independent variables.

How do you know if collinearity is present? What should you do about it? * One can find if multicollinearity is present via the following outputs: + A regression coefficient is not significant even though, theoretically, that variable should be highly correlated with Y + When you add or delete an X varaible, the regression coefficients change dramatically + You see a negative regression coefficient when your response varaible should increase along with X + You see a positive regression coefficient when your repsonse variable should decrease as X increases + Your X variables have high pairwise correlations (use the corrplot package in R to best visualize this)

The best way to deal with the problem of multicollinearity is to remove highly correlated predictors from the model, the other is to use partial least squares regression or principal components analysis that cut the number of predictors to a smaller sent of uncorrelated components.

Hitters

- I aim to perform multiple linear regression to predict player salary based on the descriptive statistics of their ouput on the field.

- Perform multiple linear regression using 95% confidence level.
- State your your hypothesis, test statistics, p-value, and conclusion. Plot graphs and interpret them.
- Which predictors will cause you to reject the null hypothesis? (Give the interpretation of each coefficient in the model).

- Now, try a different model (e.g. include more predictors or use less predictors). Which model fits the data better? What is your selected model? How did you select the model. Explain your answers. Address any other concerns you might have.

EDA :

```
data(Hitters)

glimpse(Hitters)
```

```
## Observations: 322
## Variables: 20
## $ AtBat     <int> 293, 315, 479, 496, 321, 594, 185, 298, 323, 401, 57...
## $ Hits      <int> 66, 81, 130, 141, 87, 169, 37, 73, 81, 92, 159, 53, ...
## $ HmRun     <int> 1, 7, 18, 20, 10, 4, 1, 0, 6, 17, 21, 4, 13, 0, 7, 3...
## $ Runs      <int> 30, 24, 66, 65, 39, 74, 23, 24, 26, 49, 107, 31, 48,...
## $ RBI       <int> 29, 38, 72, 78, 42, 51, 8, 24, 32, 66, 75, 26, 61, 1...
## $ Walks     <int> 14, 39, 76, 37, 30, 35, 21, 7, 8, 65, 59, 27, 47, 22...
## $ Years     <int> 1, 14, 3, 11, 2, 11, 2, 3, 2, 13, 10, 9, 4, 6, 13, 3...
## $ CAtBat    <int> 293, 3449, 1624, 5628, 396, 4408, 214, 509, 341, 520...
## $ CHits     <int> 66, 835, 457, 1575, 101, 1133, 42, 108, 86, 1332, 13...
## $ CHmRun    <int> 1, 69, 63, 225, 12, 19, 1, 0, 6, 253, 90, 15, 41, 4,...
## $ CRuns     <int> 30, 321, 224, 828, 48, 501, 30, 41, 32, 784, 702, 19...
## $ CRBI      <int> 29, 414, 266, 838, 46, 336, 9, 37, 34, 890, 504, 186...
## $ CWalks    <int> 14, 375, 263, 354, 33, 194, 24, 12, 8, 866, 488, 161...
## $ League    <fctr> A, N, A, N, N, A, N, A, N, A, A, N, N, A, N, A, N, ...
## $ Division  <fctr> E, W, W, E, E, W, E, W, W, E, E, W, E, E, E, W, W, ...
## $ PutOuts   <int> 446, 632, 880, 200, 805, 282, 76, 121, 143, 0, 238, ...
## $ Assists   <int> 33, 43, 82, 11, 40, 421, 127, 283, 290, 0, 445, 45, ...
## $ Errors    <int> 20, 10, 14, 3, 4, 25, 7, 9, 19, 0, 22, 11, 7, 6, 8, ...
## $ Salary    <dbl> NA, 475.000, 480.000, 500.000, 91.500, 750.000, 70.0...
## $ NewLeague <fctr> A, N, A, N, N, A, A, A, N, A, A, N, N, A, N, A, N, ...
```

I noticed some factor variables and NA’s in the dataset, of which I would like to omit

```
Hitters <- na.omit(Hitters)

Hitters <- Hitters %>%
              dplyr::select(AtBat:Salary) %>%
              dplyr::select(-(League:Division))

glimpse(Hitters)
```

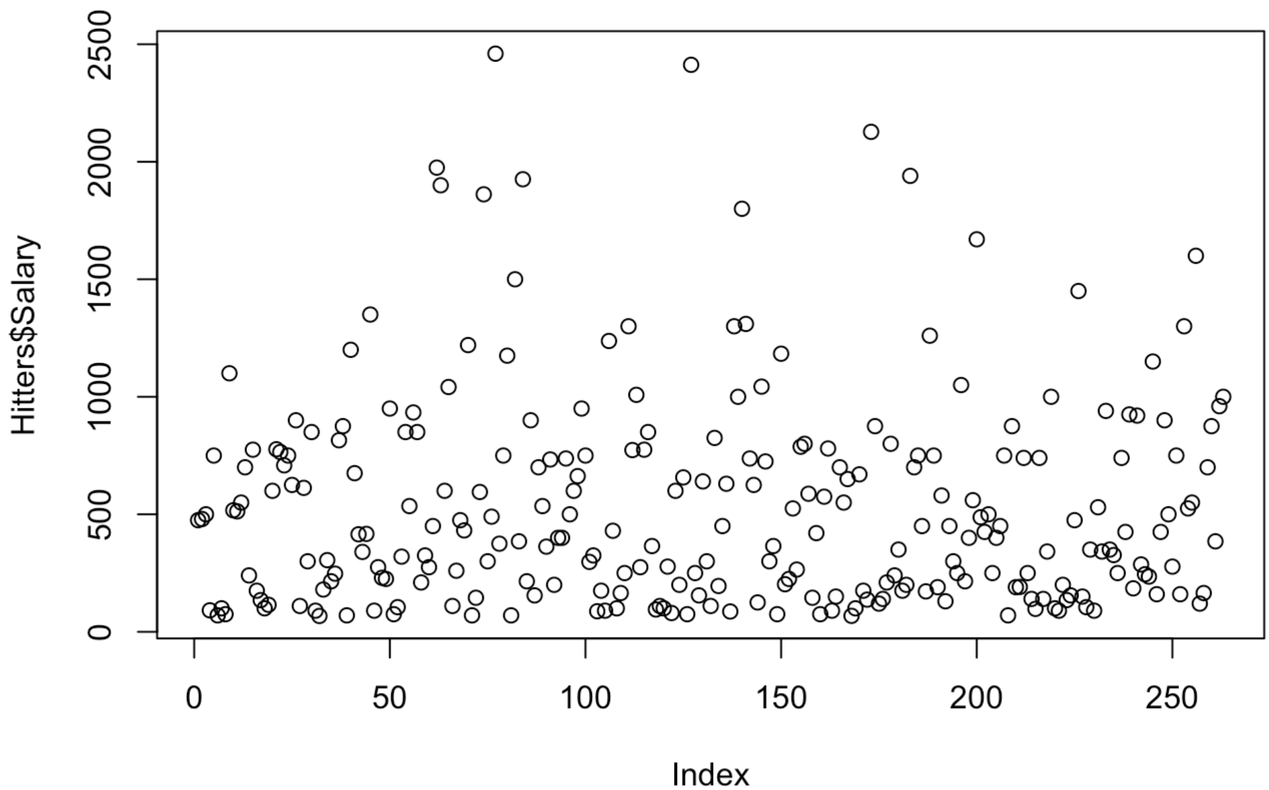
```
## Observations: 263
## Variables: 17
## $ AtBat   <int> 315, 479, 496, 321, 594, 185, 298, 323, 401, 574, 202,...
## $ Hits    <int> 81, 130, 141, 87, 169, 37, 73, 81, 92, 159, 53, 113, 6...
## $ HmRun   <int> 7, 18, 20, 10, 4, 1, 0, 6, 17, 21, 4, 13, 0, 7, 20, 2,...
## $ Runs    <int> 24, 66, 65, 39, 74, 23, 24, 26, 49, 107, 31, 48, 30, 2...
## $ RBI     <int> 38, 72, 78, 42, 51, 8, 24, 32, 66, 75, 26, 61, 11, 27,...
## $ Walks   <int> 39, 76, 37, 30, 35, 21, 7, 8, 65, 59, 27, 47, 22, 30, ...
## $ Years   <int> 14, 3, 11, 2, 11, 2, 3, 2, 13, 10, 9, 4, 6, 13, 15, 5,...
## $ CAtBat  <int> 3449, 1624, 5628, 396, 4408, 214, 509, 341, 5206, 4631...
## $ CHits   <int> 835, 457, 1575, 101, 1133, 42, 108, 86, 1332, 1300, 46...
## $ CHmRun  <int> 69, 63, 225, 12, 19, 1, 0, 6, 253, 90, 15, 41, 4, 36, ...
## $ CRuns   <int> 321, 224, 828, 48, 501, 30, 41, 32, 784, 702, 192, 205...
## $ CRBI    <int> 414, 266, 838, 46, 336, 9, 37, 34, 890, 504, 186, 204,...
## $ CWalks  <int> 375, 263, 354, 33, 194, 24, 12, 8, 866, 488, 161, 203,...
## $ PutOuts <int> 632, 880, 200, 805, 282, 76, 121, 143, 0, 238, 304, 21...
## $ Assists <int> 43, 82, 11, 40, 421, 127, 283, 290, 0, 445, 45, 11, 15...
## $ Errors  <int> 10, 14, 3, 4, 25, 7, 9, 19, 0, 22, 11, 7, 6, 8, 10, 16...
## $ Salary  <dbl> 475.000, 480.000, 500.000, 91.500, 750.000, 70.000, 10...
```

Now lets do some more exploratory data analysis, specifically keeping in mind multicollinearity and the general structure of our dependent varaible.

```
summary(Hitters)
```

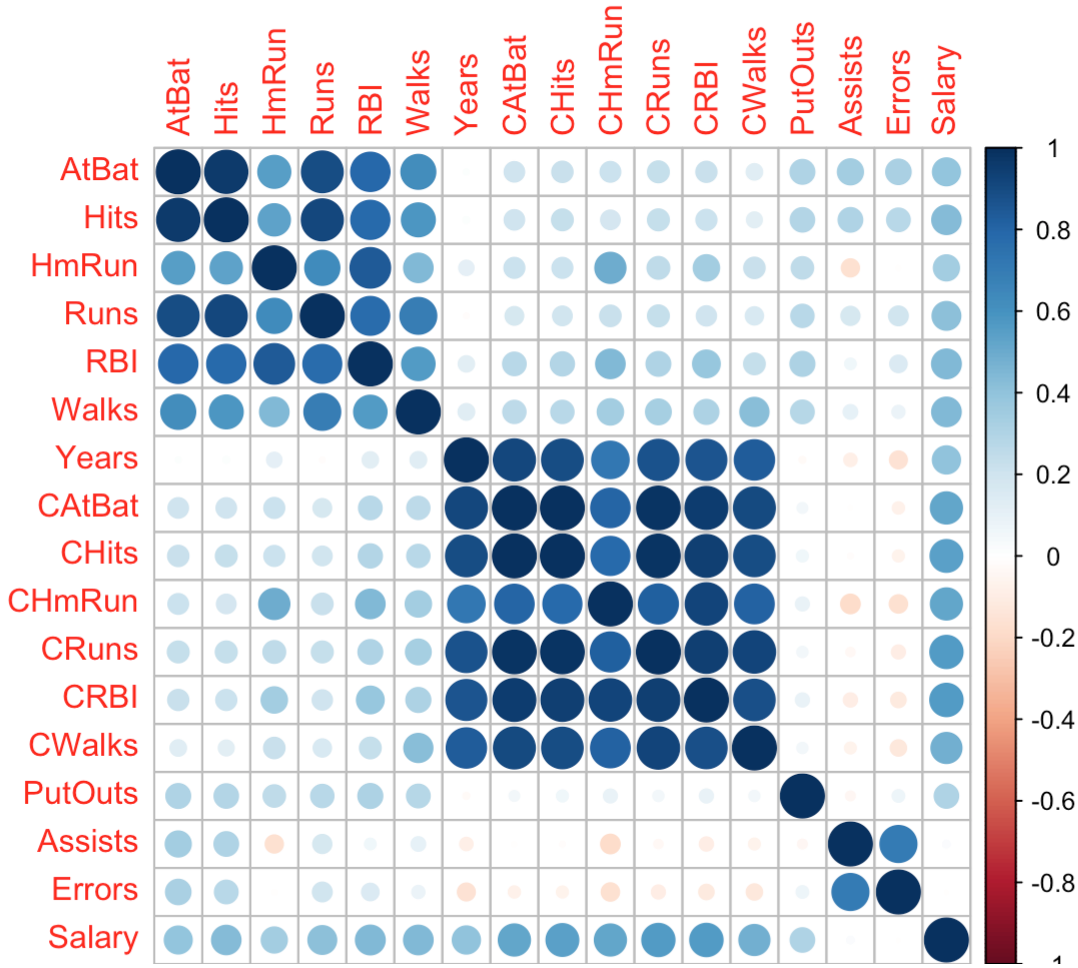
```
##      AtBat            Hits            HmRun            Runs
##  Min.   : 19.0   Min.   :  1.0   Min.   : 0.00   Min.   :  0.00
##  1st Qu.:282.5   1st Qu.: 71.5   1st Qu.: 5.00   1st Qu.: 33.50
##  Median :413.0   Median :103.0   Median : 9.00   Median : 52.00
##  Mean   :403.6   Mean   :107.8   Mean   :11.62   Mean   : 54.75
##  3rd Qu.:526.0   3rd Qu.:141.5   3rd Qu.:18.00   3rd Qu.: 73.00
##  Max.   :687.0   Max.   :238.0   Max.   :40.00   Max.   :130.00
##       RBI             Walks            Years            CAtBat
##  Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0
##  1st Qu.: 30.00   1st Qu.: 23.00   1st Qu.: 4.000   1st Qu.:  842.5
##  Median : 47.00   Median : 37.00   Median : 6.000   Median : 1931.0
##  Mean   : 51.49   Mean   : 41.11   Mean   : 7.312   Mean   : 2657.5
##  3rd Qu.: 71.00   3rd Qu.: 57.00   3rd Qu.:10.000   3rd Qu.: 3890.5
##  Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0
##      CHits            CHmRun           CRuns             CRBI
##  Min.   :   4.0   Min.   :  0.00   Min.   :   2.0   Min.   :   3.0
##  1st Qu.: 212.0   1st Qu.: 15.00   1st Qu.: 105.5   1st Qu.:  95.0
##  Median : 516.0   Median : 40.00   Median : 250.0   Median : 230.0
##  Mean   : 722.2   Mean   : 69.24   Mean   : 361.2   Mean   : 330.4
##  3rd Qu.:1054.0   3rd Qu.: 92.50   3rd Qu.: 497.5   3rd Qu.: 424.5
##  Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.0
##      CWalks           PutOuts           Assists          Errors
##  Min.   :   1.0   Min.   :   0.0   Min.   :  0.0   Min.   : 0.000
##  1st Qu.:  71.0   1st Qu.: 113.5   1st Qu.:  8.0   1st Qu.: 3.000
##  Median : 174.0   Median : 224.0   Median : 45.0   Median : 7.000
##  Mean   : 260.3   Mean   : 290.7   Mean   :118.8   Mean   : 8.593
##  3rd Qu.: 328.5   3rd Qu.: 322.5   3rd Qu.:192.0   3rd Qu.:13.000
##  Max.   :1566.0   Max.   :1377.0   Max.   :492.0   Max.   :32.000
##      Salary
##  Min.   :  67.5
##  1st Qu.: 190.0
##  Median : 425.0
##  Mean   : 535.9
##  3rd Qu.: 750.0
##  Max.   :2460.0
```

```
plot(Hitters$Salary)
```




```
Hitters_cor <- cor(Hitters)

corrplot(Hitters_cor, method = "circle")
```



Wow, the results of the data summary and distribution of salary seem fairly normal, nothing particular stands out; however, the correlation matrix is incredibly interesting. There appear to be very little correlation between fielding variables and batting variables, but salary appears to account for all factors evenly outisde of assists and errors.

As I look at the data, it affirms my thoughts on having salary being the most statistically solid thing to predict, so I will build a couple models to see how to best produce a lean, reproducible and accurate predicive model using multiple linear regression.

Build Model 1: All Variables

```
model1 <- lm(Salary~., data = Hitters)
```

Evaluate Model 1:

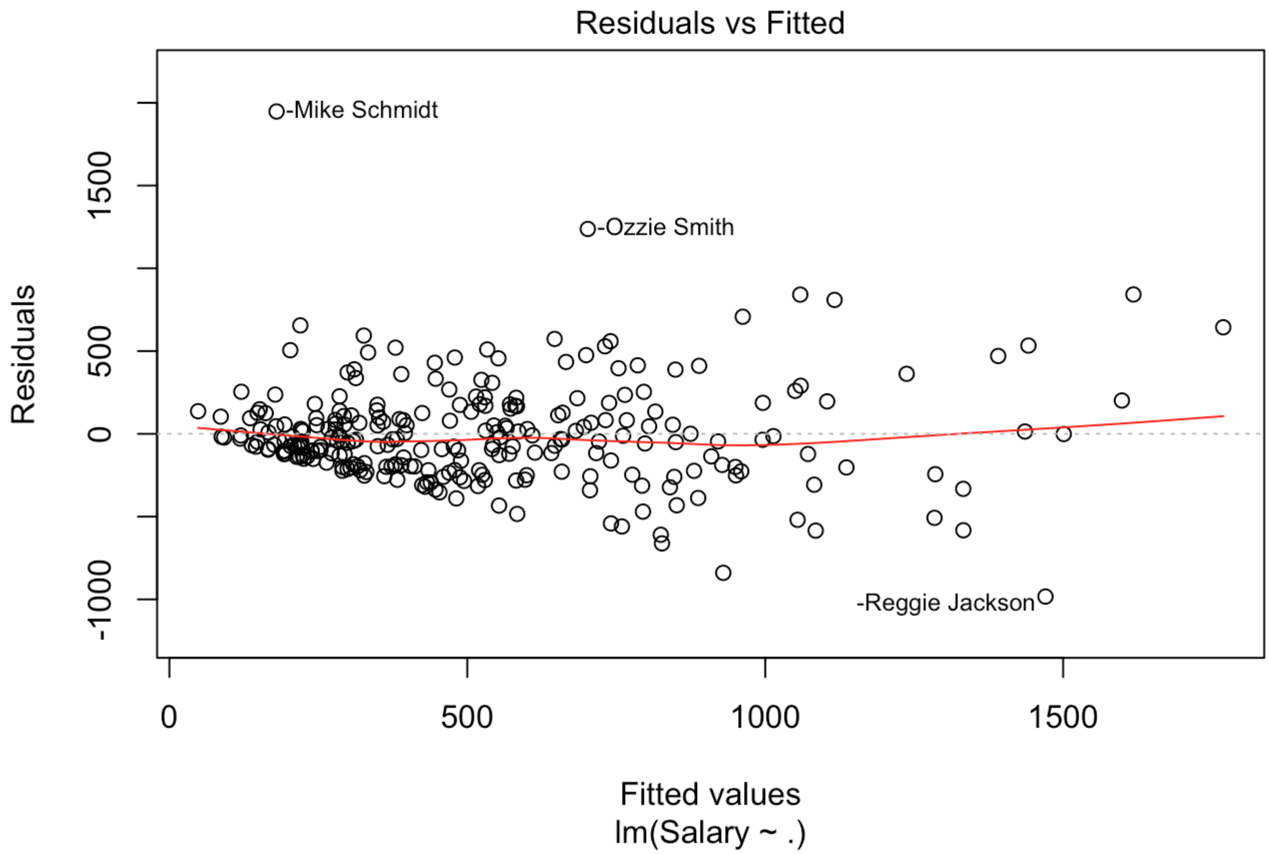
```
summary(model1)
```

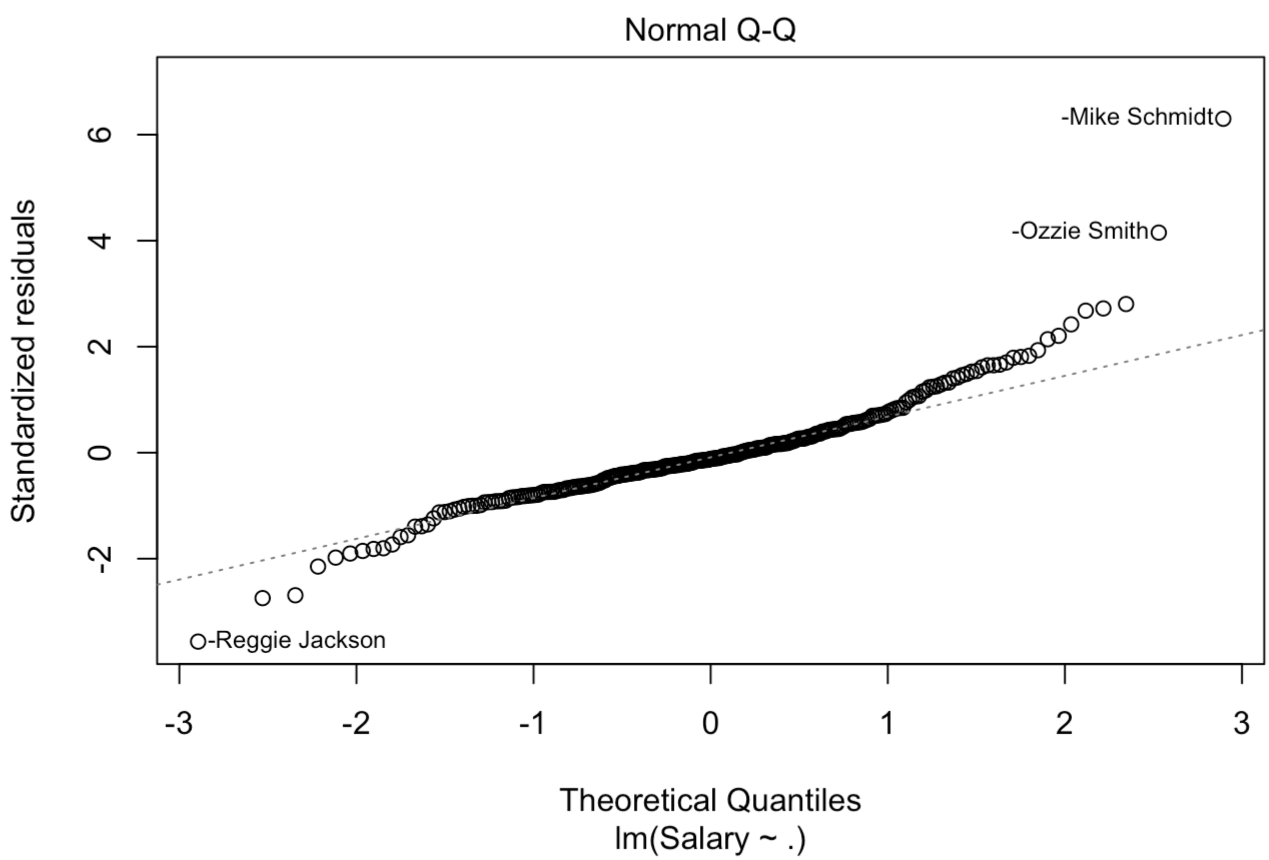
```
##
## Call:
## lm(formula = Salary ~ ., data = Hitters)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -982.81 -187.84  -35.66  130.61 1947.43
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 126.10553   83.62448   1.508 0.132838
## AtBat        -2.20302    0.63605  -3.464 0.000629 ***
## Hits          7.82776    2.40198   3.259 0.001276 **
## HmRun         2.16355    6.23618   0.347 0.728937
## Runs         -2.09957    3.00849  -0.698 0.485911
## RBI          -0.02292    2.61033  -0.009 0.993003
## Walks         6.15106    1.84028   3.342 0.000960 ***
## Years        -2.59237   12.45401  -0.208 0.835280
## CAtBat       -0.17628    0.13667  -1.290 0.198325
## CHits         0.06976    0.67874   0.103 0.918221
## CHmRun       -0.23309    1.63561  -0.143 0.886795
## CRuns         1.61005    0.75162   2.142 0.033168 *
## CRBI          0.80143    0.70000   1.145 0.253367
## CWalks       -0.79394    0.33243  -2.388 0.017681 *
## PutOuts       0.29457    0.07830   3.762 0.000211 ***
## Assists       0.38400    0.22383   1.716 0.087499 .
## Errors       -2.87871    4.42077  -0.651 0.515539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 319.9 on 246 degrees of freedom
## Multiple R-squared:  0.5279, Adjusted R-squared:  0.4972
## F-statistic: 17.19 on 16 and 246 DF,  p-value: < 2.2e-16
```

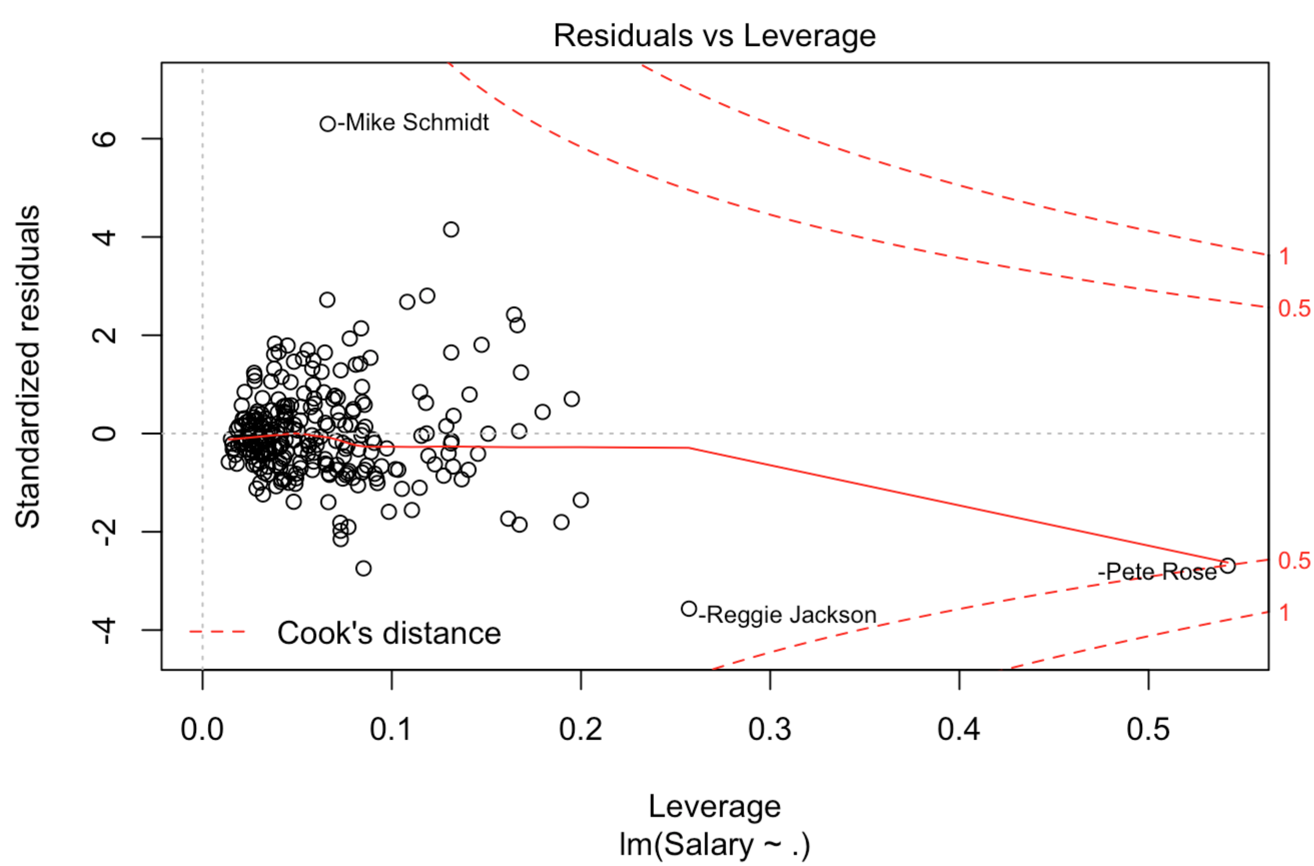
```
confint(model1, level=0.95)
```

```
##                    2.5 %       97.5 %
## (Intercept) -38.60578611 290.81683904
## AtBat        -3.45582878  -0.95021507
## Hits          3.09668456  12.55883854
## HmRun       -10.11956525  14.44665724
## Runs         -8.02524787   3.82611357
## RBI          -5.16436672   5.11853490
## Walks         2.52633873   9.77578274
## Years       -27.12245428  21.93771756
## CAtBat       -0.44546379   0.09291163
## CHits        -1.26712857   1.40665177
## CHmRun       -3.45468115   2.98850491
## CRuns         0.12961135   3.09048874
## CRBI         -0.57733731   2.18019263
## CWalks       -1.44870850  -0.13917426
## PutOuts       0.14033988   0.44880699
## Assists      -0.05687017   0.82486077
## Errors      -11.58610959   5.82868100
```

Plot Model 1:


Residuals vs Fitted
Residuals
Fitted values
lm(Salary ~ .)
-Mike Schmidt
-Ozzie Smith
-Reggie Jackson
1500
500
0
-1000
0
500
1000
1500


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Salary ~ .)
-Mike Schmidt
-Ozzie Smith
-Reggie Jackson
6
4
2
0
-2
-3
-2
-1
0
1
2
3



Residuals Model 1:

```
augment(model1)
```

```
##              .rownames   Salary AtBat Hits HmRun Runs RBI Walks Years
## 1          -Alan Ashby  475.000   315   81     7   24  38    39    14
## 2         -Alvin Davis  480.000   479  130    18   66  72    76     3
## 3        -Andre Dawson  500.000   496  141    20   65  78    37    11
## 4    -Andres Galarraga   91.500   321   87    10   39  42    30     2
##     CAtBat CHits CHmRun CRuns CRBI CWalks PutOuts Assists Errors
## 1     3449   835     69   321  414    375     632      43     10
## 2     1624   457     63   224  266    263     880      82     14
## 3     5628  1575    225   828  838    354     200      11      3
## 4      396   101     12    48   46     33     805      40      4
##        .fitted   .se.fit       .resid       .hat   .sigma      .cooksd
## 1    392.66074  85.79236   82.3392578 0.07193558 320.4780 3.255360e-04
## 2    793.25176  68.15163 -313.2517550 0.04539406 319.8693 2.810188e-03
## 3   1084.71881  88.78994 -584.7188122 0.07705025 318.1571 1.777910e-02
## 4    481.45582  57.13408 -389.9558179 0.03190341 319.5228 2.975967e-03
##      .std.resid
## 1   0.267202898
## 2  -1.002316138
## 3  -1.902748180
## 4  -1.239022903
##  [ reached getOption("max.print") -- omitted 259 rows ]
```

Model 1 Conclusion:

Looking towards the results we see nothing that suprises us terribly. Hits, walks and home runs have the most positive coefficients on the dependent variable of salary, and we see that years in the leauge and at bats actually have negative coefficients. The Multiple R-Squared is 0.529, which indicates that we can explain 52% of the variance, while the f statistic is 17.19, which indicates the strength of our ability to reject the null hypothesis. Lastly we see the low p-value, indicating a low probability of to reject the null hypothesis.

Things really seem to be a mixed bag here in the output statistics, but graphically our linear output appears to track the path of the data very well. Also graphically we see that there are some outliers that are having a significant effect on the output of our model. This is viewed in leverage output of the data. We see players like Pete Rose, Ozzie Smith and Mike Schmidt are significantly effecting the model and more than likely having a negative effect on our ability to properly linearly regress our model. Secondly, we see that some variables like PutOuts, Career Hits, RBI’s and Assists have very little effect on the model, so lets omit some of these factors and see if it improves our model.

```
# remove additional columns / player extreme stats
Hitters2 <- Hitters %>%
              dplyr::filter(HmRun <= 30, Runs < 100, Years < 20, CRuns < 1800, CRBI < 12
00) %>%
              dplyr::select(-(RBI)) %>%
              dplyr::select(-(CHits)) %>%
              dplyr::select(-(PutOuts)) %>%
              dplyr::select(-(Assists))

glimpse(Hitters2)
```

```
## Observations: 239
## Variables: 13
## $ AtBat  <int> 315, 479, 496, 321, 594, 185, 298, 323, 401, 202, 418, ...
## $ Hits   <int> 81, 130, 141, 87, 169, 37, 73, 81, 92, 53, 113, 60, 43,...
## $ HmRun  <int> 7, 18, 20, 10, 4, 1, 0, 6, 17, 4, 13, 0, 7, 20, 2, 8, 1...
## $ Runs   <int> 24, 66, 65, 39, 74, 23, 24, 26, 49, 31, 48, 30, 29, 89,...
## $ Walks  <int> 39, 76, 37, 30, 35, 21, 7, 8, 65, 27, 47, 22, 30, 73, 1...
## $ Years  <int> 14, 3, 11, 2, 11, 2, 3, 2, 13, 9, 4, 6, 13, 15, 5, 8, 1...
## $ CAtBat <int> 3449, 1624, 5628, 396, 4408, 214, 509, 341, 5206, 1876,...
## $ CHmRun <int> 69, 63, 225, 12, 19, 1, 0, 6, 253, 15, 41, 4, 36, 177, ...
## $ CRuns  <int> 321, 224, 828, 48, 501, 30, 41, 32, 784, 192, 205, 309,...
## $ CRBI   <int> 414, 266, 838, 46, 336, 9, 37, 34, 890, 186, 204, 103, ...
## $ CWalks <int> 375, 263, 354, 33, 194, 24, 12, 8, 866, 161, 203, 207, ...
## $ Errors <int> 10, 14, 3, 4, 25, 7, 9, 19, 0, 11, 7, 6, 8, 10, 16, 2, ...
## $ Salary <dbl> 475.000, 480.000, 500.000, 91.500, 750.000, 70.000, 100...
```

```
summary(Hitters2)
```

```
##      AtBat            Hits         HmRun            Runs
##  Min.   : 19.0   Min.   :  1   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:279.0   1st Qu.: 70   1st Qu.: 4.00   1st Qu.:32.50
##  Median :394.0   Median :101   Median : 8.00   Median :50.00
##  Mean   :389.1   Mean   :103   Mean   :10.42   Mean   :51.39
##  3rd Qu.:508.5   3rd Qu.:136   3rd Qu.:16.00   3rd Qu.:68.50
##  Max.   :687.0   Max.   :213   Max.   :30.00   Max.   :98.00
##      Walks           Years            CAtBat        CHmRun
##  Min.   : 0.00   Min.   : 1.000   Min.   :  19   Min.   :  0.00
##  1st Qu.:22.00   1st Qu.: 4.000   1st Qu.: 799   1st Qu.: 12.50
##  Median :35.00   Median : 6.000   Median :1789   Median : 36.00
##  Mean   :39.46   Mean   : 7.075   Mean   :2460   Mean   : 60.18
##  3rd Qu.:54.00   3rd Qu.:10.000   3rd Qu.:3612   3rd Qu.: 82.00
##  Max.   :97.00   Max.   :18.000   Max.   :8424   Max.   :347.00
##      CRuns             CRBI             CWalks           Errors
##  Min.   :   2.0   Min.   :   3.0   Min.   :   1.0   Min.   : 0.000
##  1st Qu.:  99.0   1st Qu.:  82.5   1st Qu.:  65.5   1st Qu.: 3.000
##  Median : 238.0   Median : 204.0   Median : 168.0   Median : 7.000
##  Mean   : 328.3   Mean   : 296.9   Mean   : 239.5   Mean   : 8.582
##  3rd Qu.: 456.0   3rd Qu.: 416.5   3rd Qu.: 311.0   3rd Qu.:13.000
##  Max.   :1175.0   Max.   :1152.0   Max.   :1380.0   Max.   :32.000
##      Salary
##  Min.   :  67.5
##  1st Qu.: 175.0
##  Median : 400.0
##  Mean   : 496.9
##  3rd Qu.: 737.5
##  Max.   :2460.0
```

Now that I have removed some outliers and normalized the valuable variables, lest see if my model performance will improve.

Build Model 2: All Variables

```
model2 <- lm(Salary~., data = Hitters2)
```

Evaluate Model 2:

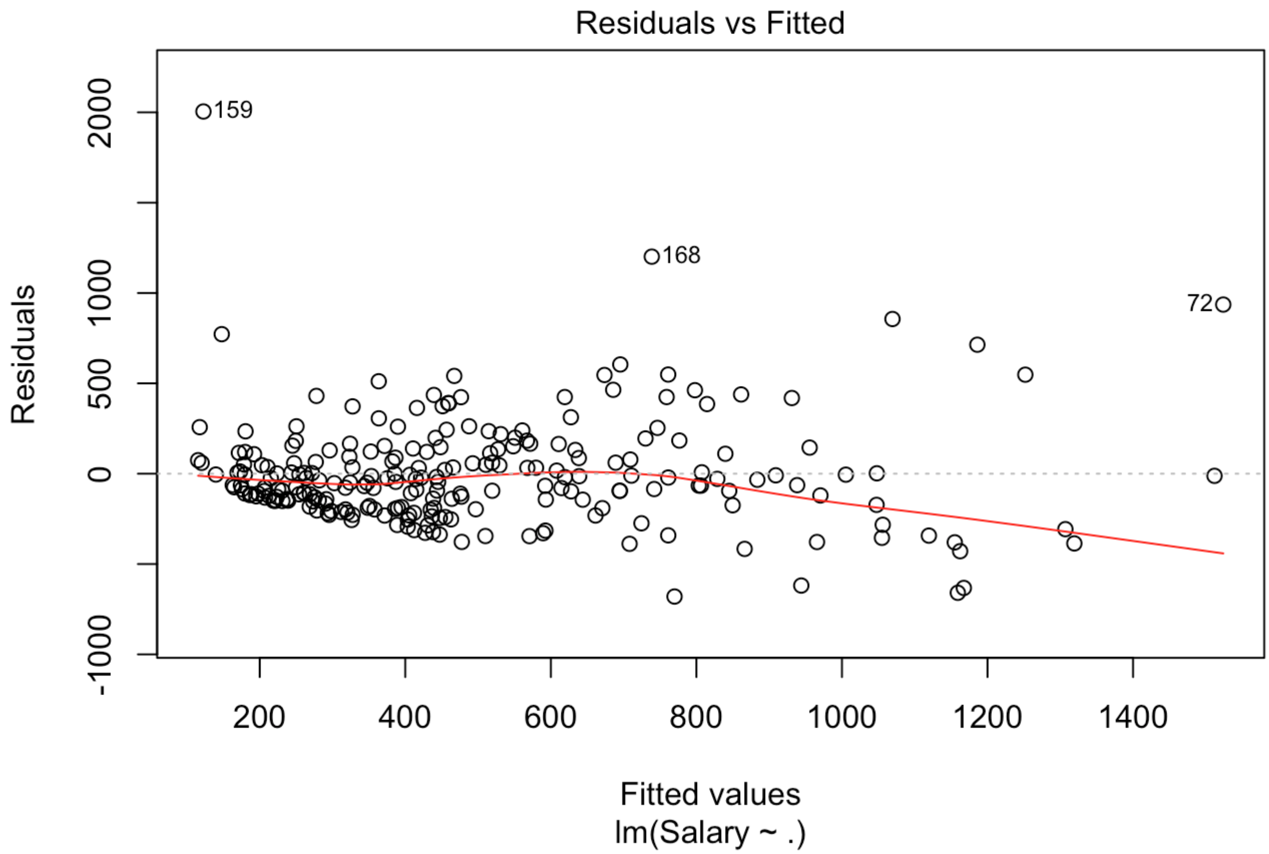
```
summary(model2)
```

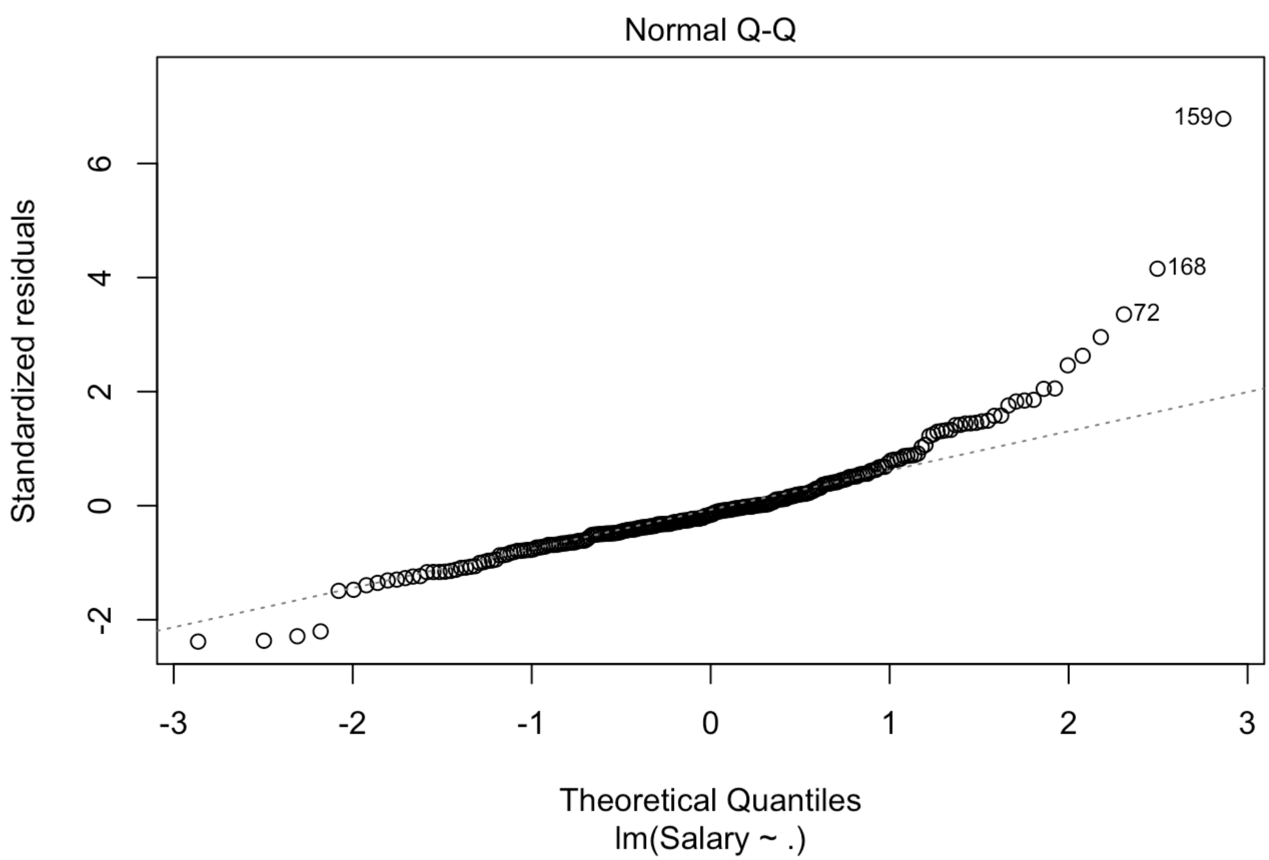
```
##
## Call:
## lm(formula = Salary ~ ., data = Hitters2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -679.95 -159.48  -45.18  117.69 2004.69
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 145.74768   79.58586   1.831 0.068368 .
## AtBat        -1.00150    0.63944  -1.566 0.118699
## Hits          5.15060    2.23739   2.302 0.022242 *
## HmRun        -1.68969    4.34866  -0.389 0.697971
## Runs         -4.27868    2.76073  -1.550 0.122580
## Walks         6.17828    1.77985   3.471 0.000621 ***
## Years       -13.44435   11.98601  -1.122 0.263193
## CAtBat       -0.08380    0.09009  -0.930 0.353284
## CHmRun        0.82587    1.34062   0.616 0.538491
## CRuns         1.28490    0.50575   2.541 0.011739 *
## CRBI          0.51450    0.54251   0.948 0.343950
## CWalks       -0.56439    0.28338  -1.992 0.047612 *
## Errors        1.89977    3.41283   0.557 0.578313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 302.5 on 226 degrees of freedom
## Multiple R-squared:  0.4771, Adjusted R-squared:  0.4493
## F-statistic: 17.18 on 12 and 226 DF,  p-value: < 2.2e-16
```

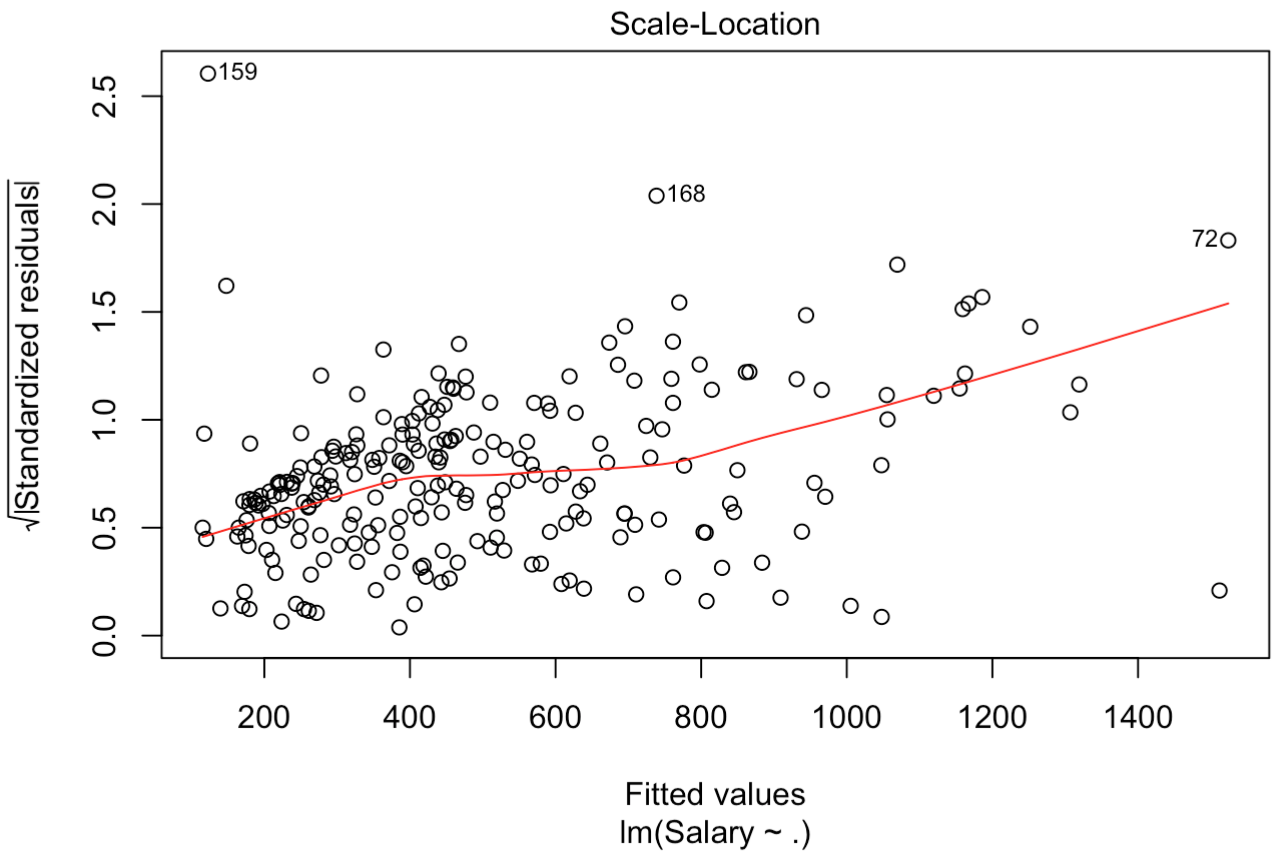
```
confint(model2, level=0.95)
```

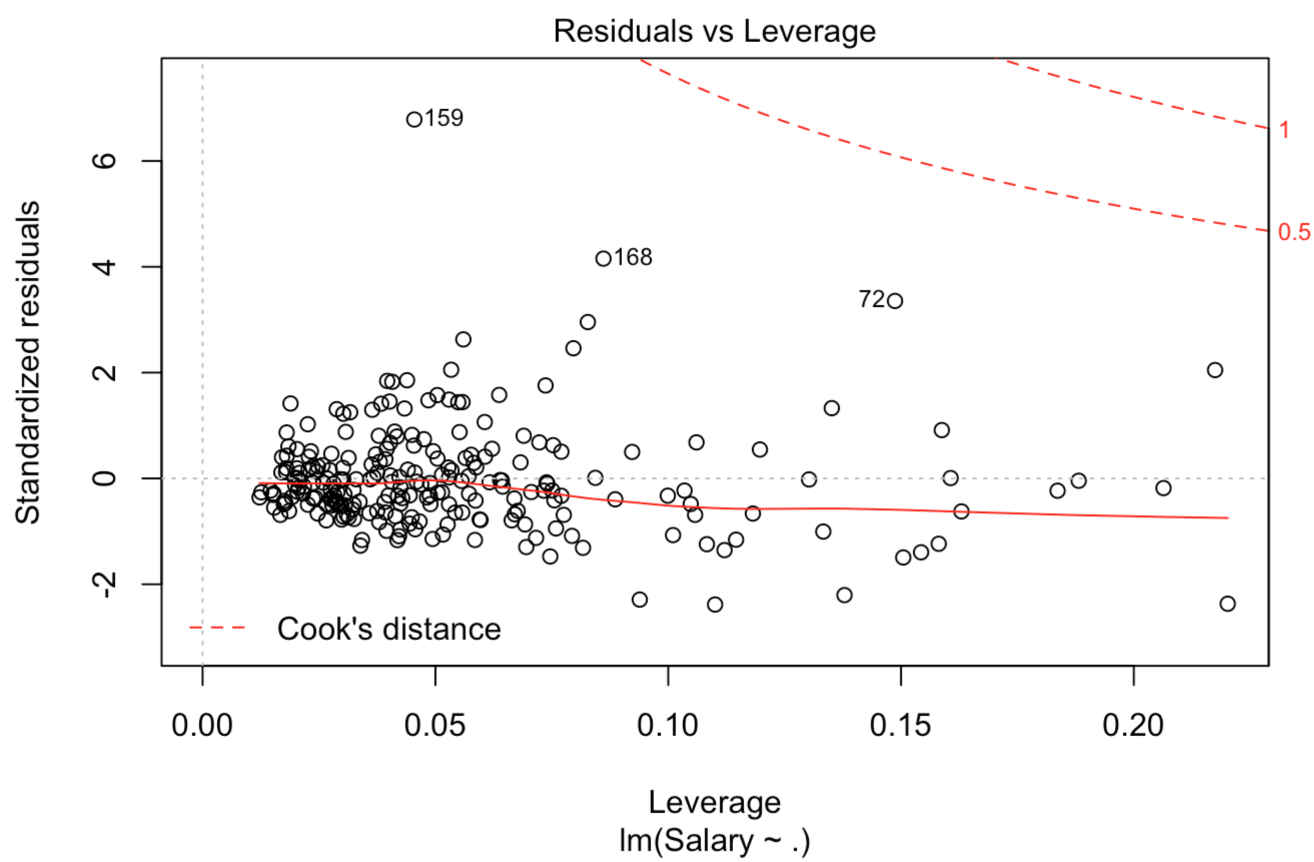
```
##                  2.5 %       97.5 %
## (Intercept) -11.0775385 302.57290545
## AtBat        -2.2615375   0.25853274
## Hits          0.7417885   9.55941732
## HmRun       -10.2587985   6.87941163
## Runs         -9.7187543   1.16138585
## Walks         2.6710587   9.68550778
## Years       -37.0629811  10.17428802
## CAtBat       -0.2613255   0.09372781
## CHmRun       -1.8158425   3.46758193
## CRuns         0.2882987   2.28149504
## CRBI         -0.5545163   1.58352010
## CWalks       -1.1227908  -0.00599234
## Errors       -4.8252588   8.62479954
```

Plot Model 2:


Residuals vs Fitted
Residuals
2000
1000
500
0
-1000
200
400
600
800
1000
1200
1400
Fitted values
lm(Salary ~ .)
159
168
72


Normal Q-Q
Standardized residuals
6
4
2
0
-2
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Salary ~ .)
159
168
72





Residuals Model 2:

```
augment(model2)
```

```
##       Salary AtBat Hits HmRun Runs Walks Years CAtBat CHmRun CRuns CRBI
## 1    475.000   315   81     7   24    39    14   3449     69   321  414
## 2    480.000   479  130    18   66    76     3   1624     63   224  266
## 3    500.000   496  141    20   65    37    11   5628    225   828  838
## 4     91.500   321   87    10   39    30     2    396     12    48   46
## 5    750.000   594  169     4   74    35    11   4408     19   501  336
##     CWalks Errors   .fitted   .se.fit       .resid       .hat   .sigma
## 1      375     10  386.4584  79.02475   88.5415520 0.06826753 303.0614
## 2      263     14  670.7919  59.95446 -190.7918953 0.03929442 302.8452
## 3      354      3 1159.1904  92.67257 -659.1903853 0.09388372 299.5868
## 4       33      4  298.1044  39.07322 -206.6044326 0.01668961 302.8047
## 5      194     25  567.1922  82.98798  182.8077704 0.07528671 302.8580
##          .cooksd   .std.resid
## 1   5.184066e-04  0.303281183
## 2   1.303211e-03 -0.643589153
## 3   4.178195e-02 -2.289617035
## 4   6.195678e-04 -0.688871587
## 5   2.474218e-03  0.628543085
##  [ reached getOption("max.print") -- omitted 234 rows ]
```

Model 2 Conclusion:

As I look at model 2, I was unable to improve the model performance overall, however I did accomplished what I wanted in reducing leverage on outliers. My model performance appears to rely less heavily on any single varaible, with Hits, Walks and Years being strongest coefficients effecting the model. Looking towards p-value, things look much the same, but the multiple R squared has not improved. The fstatistic is also the same ast the previous model.

This confirms some of my additional thoughts on valuing the attainment of as many factors as possible to predict an output with MLR (assuming we are avoiding multicollinearity).We see improved leverage but reduced model accuracy, nevertheless gaining knowledge on the baseball statistics most responsible for effects on salary. Specifically Hits, Walks and years in the league.