Regression

Charles Wallis

Linear Regression

Linear model is one of the simplest machine learning models that can be computed, and it is often used in regression to predict numerical values. Linear regression is intended to draw a line that best fits the test data, from the given data frame. Linear regression has advantage of being easy to understand and users can easily read the graph to make predictions of unknown data Some weaknesses can include that it is sensitive to some outlier instances, and that it is prone to over fitting.

Data

To perform regression, we will import an Asteroid Data set with 958,524 unique data

Basic Column Definitions

- name: Object IAU name
- Albedo: Geometric albedo
- e: Eccentricity
- a: Semi-major axis au Unit
- q: perihelion distance au Unit

Import File

```
asteroid <- read.csv("Asteroid.csv")
```

Data Cleanup (Take only first half of data without any NAs)

```
keep <- c("name", "H" , "albedo", "H", "e", "a" , "q")
df <- asteroid[keep]
dfc <- df[rowSums(is.na(df)) == 0, ]
i <- sample(1:nrow(dfc), nrow(dfc)*0.5, replace=FALSE)
dfh <- df[i,]
```

Divide into 80/20 train/test

```
set.seed(1234)
i <- sample(1:nrow(dfh), nrow(dfh)*0.8, replace=FALSE)
train <- dfh[i,]
test <- dfh[-i,]
```

Summary of Asteroids

```
summary(dfh)
```

```
##      name                H             albedo           H.1        
##  Length:65620       Min.   : 3.30   Min.   :0.001   Min.   : 3.30  
##  Class :character   1st Qu.:14.30   1st Qu.:0.070   1st Qu.:14.30  
##  Mode  :character   Median :15.00   Median :0.155   Median :15.00  
##                     Mean   :14.84   Mean   :0.178   Mean   :14.84  
##                     3rd Qu.:15.60   3rd Qu.:0.262   3rd Qu.:15.60  
##                     Max.   :22.70   Max.   :1.000   Max.   :22.70  
##                                     NA's   :31351                  
##        e                    a                 q          
##  Min.   :0.0009785   Min.   :  0.6423   Min.   : 0.1401  
##  1st Qu.:0.0885720   1st Qu.:  2.3669   1st Qu.: 2.0025  
##  Median :0.1331931   Median :  2.5976   Median : 2.2102  
##  Mean   :0.1378297   Mean   :  2.7118   Mean   : 2.3258  
##  3rd Qu.:0.1795459   3rd Qu.:  2.8977   3rd Qu.: 2.5650  
##  Max.   :0.9632867   Max.   :564.6993   Max.   :43.2263  
##
```

Dimensions of the Data Frame, There are 131142 rows, and 8 columns

```
dim(dfh)
```

```
## [1] 65620     7
```

Structure of Asteroids

```
str(dfh)
```

```
## 'data.frame':    65620 obs. of  7 variables:
##  $ name  : chr  "Lesire" "Shunda" "Krat" "" ...
##  $ H     : num  12.7 14.3 10.4 13.8 16.2 14.5 15.6 13.2 14.7 15.7 ...
##  $ albedo: num  0.184 0.473 0.102 0.056 NA NA NA 0.303 NA 0.071 ...
##  $ H.1   : num  12.7 14.3 10.4 13.8 16.2 14.5 15.6 13.2 14.7 15.7 ...
##  $ e     : num  0.1999 0.1915 0.1002 0.0656 0.1933 ...
##  $ a     : num  2.68 2.3 3.21 2.45 2.39 ...
##  $ q     : num  2.14 1.86 2.89 2.29 1.92 ...
```

First 5 rows of Asteroid Data

```
head(dfh, n=5)
```

```
##         name    H albedo  H.1          e        a        q
## 29311 Lesire 12.7  0.184 12.7 0.19993301 2.680711 2.144749
## 13906 Shunda 14.3  0.473 14.3 0.19147733 2.300672 1.860146
## 3036    Krat 10.4  0.102 10.4 0.10016515 3.210480 2.888901
## 19525        13.8  0.056 13.8 0.06560057 2.452024 2.291169
## 97904        16.2     NA 16.2 0.19329014 2.385105 1.924088
```

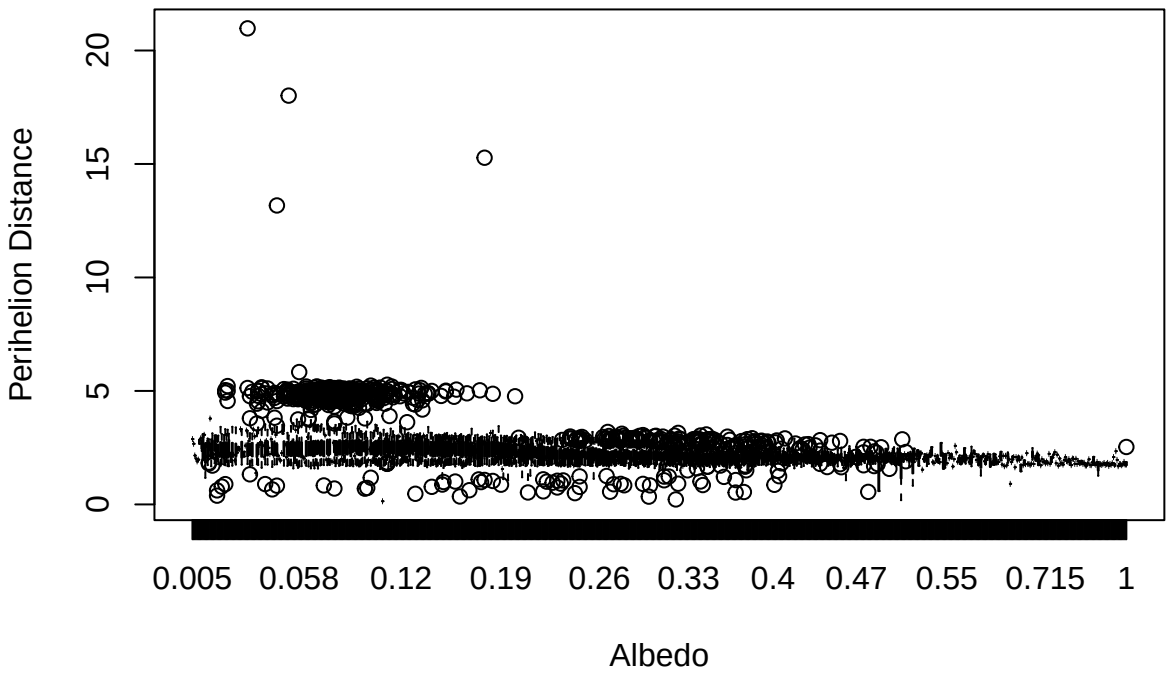
Last 5 rows of Asteroid Data

```
tail(dfh, n=5)
```

```
##          name    H albedo  H.1         e        a        q
## 21517    Dobi 14.7  0.259 14.7 0.1815820 2.357051 1.929053
## 107385       15.6     NA 15.6 0.2170938 2.247732 1.759764
## 10956  Vosges 15.7     NA 15.7 0.1193044 2.351678 2.071112
## 4334       Foo 13.1     NA 13.1 0.1954312 3.138359 2.525026
## 98273        16.3     NA 16.3 0.0674381 2.278171 2.124535
```

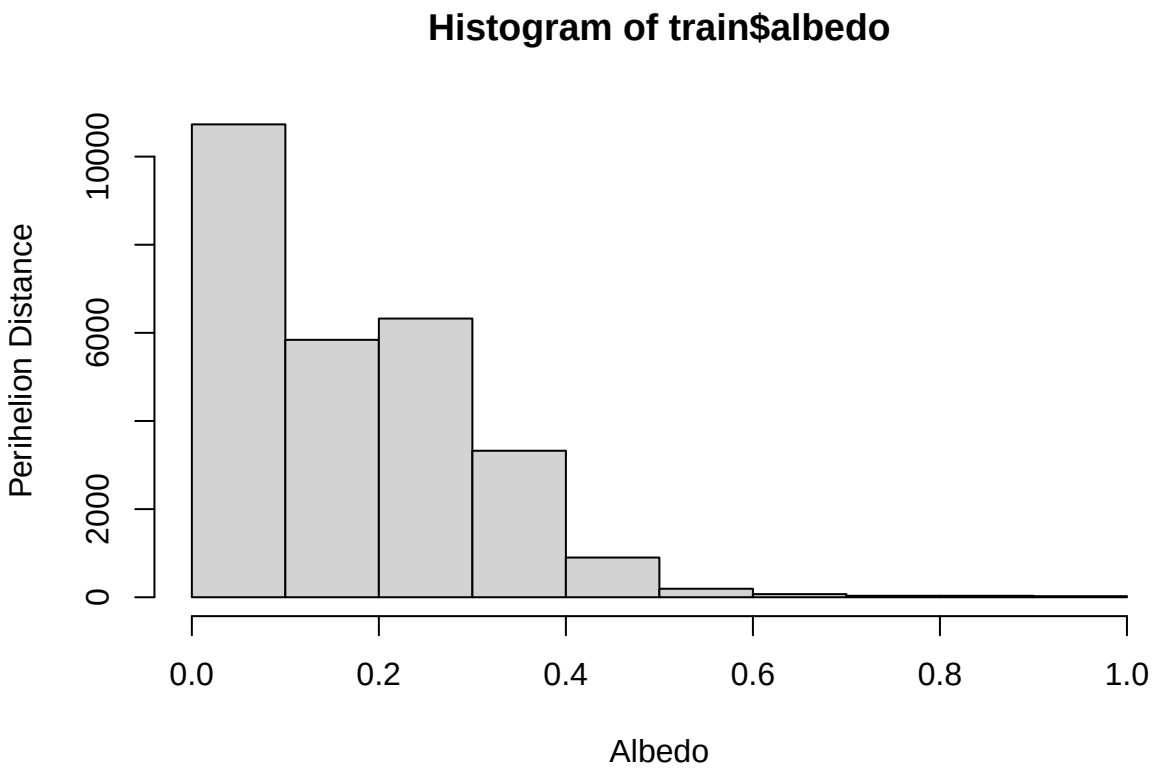
Plotting the data points for Albedo to Perihelion Distance (Asteroid's closest point in orbit to the star)

```
plot(train$q ~ (as.factor(train$albedo)), xlab = "Albedo", ylab = "Perihelion Distance")
```



Histogram of Albedo to Perihelion Distance

```
hist(train$albedo, breaks = c(0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1), xlab = "Albedo", yla
```



Build a simple linear regression model and output the summary.

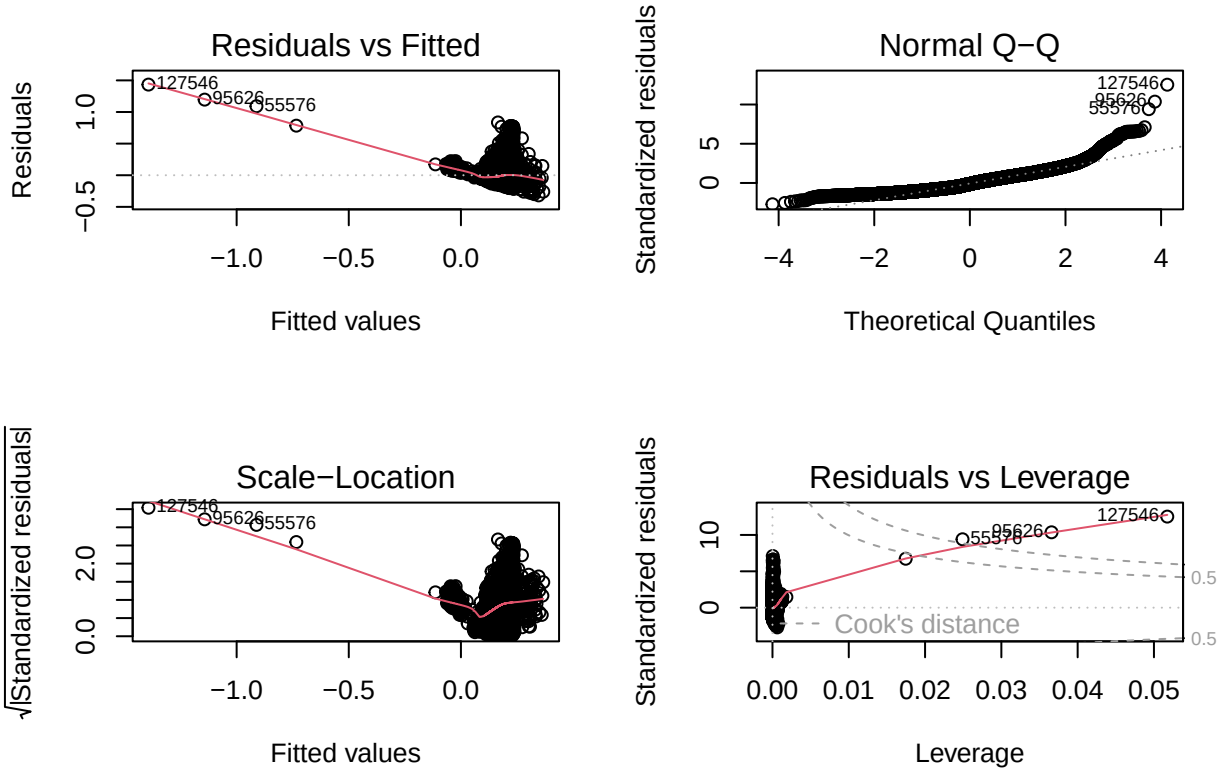
```
lm1 <- lm(albedo~q, data = train)
summary(lm1)
```

```
##
## Call:
## lm(formula = albedo ~ q, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31842 -0.09223 -0.01484  0.07646  1.43317
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.379034   0.003482  108.85   <2e-16 ***
## q           -0.084488   0.001437  -58.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1175 on 27477 degrees of freedom
##   (25017 observations deleted due to missingness)
## Multiple R-squared:  0.1118, Adjusted R-squared:  0.1118
## F-statistic:  3459 on 1 and 27477 DF,  p-value: < 2.2e-16
```

Based on the summary output, for this model we know that the slope is -0.086912 with an intercept of 0.384249. the increase in q by 1 would lower the albedo by 0.086912, which can be used to predict albedo value based on q the summary also provides *** as the significance code to show that the value has a high correlation, hence a good predictor The R-Squared value is on the lower end, meaning that there is a high variance in this model High F-statistic value and low p-value also helps me understand that the predictor is significant, and null hypothesis is ignored

Plot the residuals

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



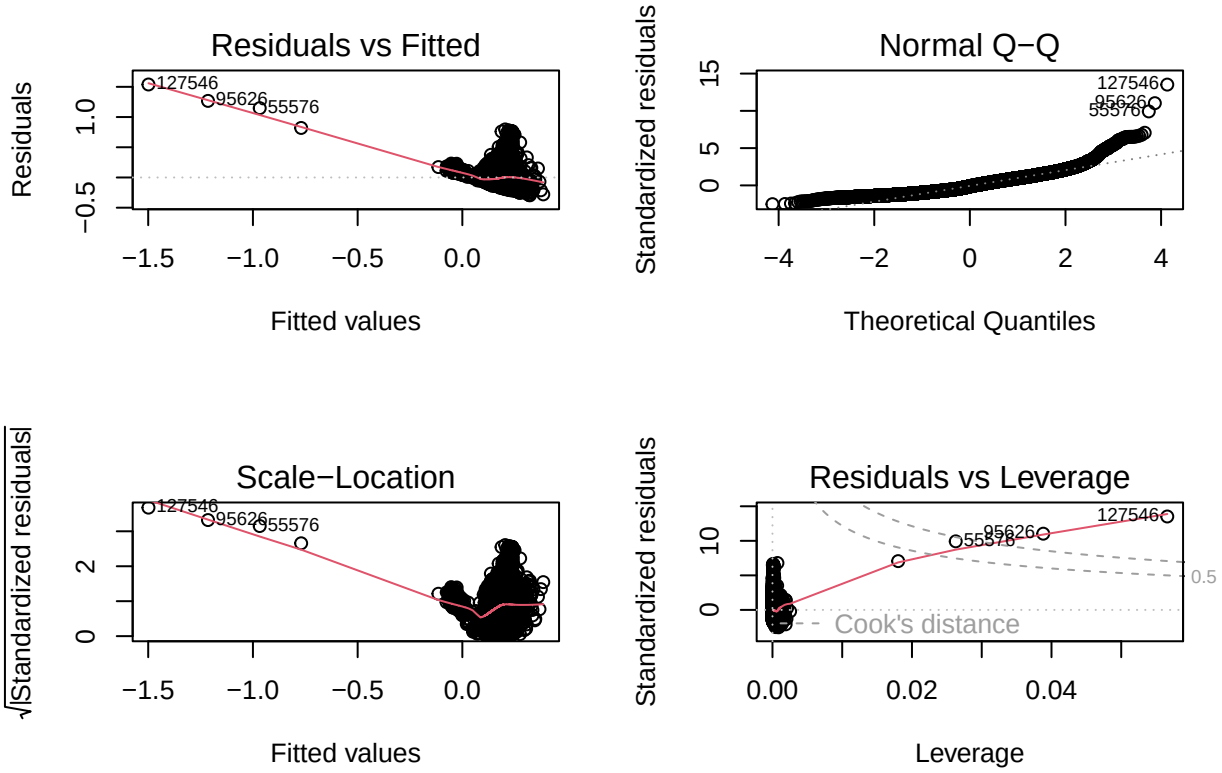
Residuals vs Fitted: this tells us that the relationship is not really linear, since the residuals are clumped in one area Normal Q-Q: this shows that the residuals are normally distributed, since it's plotted near the dashed line Scale-Location: this again, shows that the residuals are not spread equally along the range of predictors (similar to residuals vs fitted) Residuals vs Leverage: many points are not on the cook's distance, which means that there are a lot of instances influencing the whole model

Build a multiple linear regression model

```
lm2 <- lm(albedo ~ q + H, data = train)
summary(lm2)
```

```
##
## Call:
## lm(formula = albedo ~ q + H, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29440 -0.09126 -0.01495  0.07583  1.53882
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.514700   0.011042   46.61   <2e-16 ***
## q           -0.092916   0.001573  -59.06   <2e-16 ***
## H           -0.008063   0.000623  -12.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1172 on 27476 degrees of freedom
##   (25017 observations deleted due to missingness)
## Multiple R-squared:  0.1172, Adjusted R-squared:  0.1171
## F-statistic:  1824 on 2 and 27476 DF,  p-value: < 2.2e-16
```

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

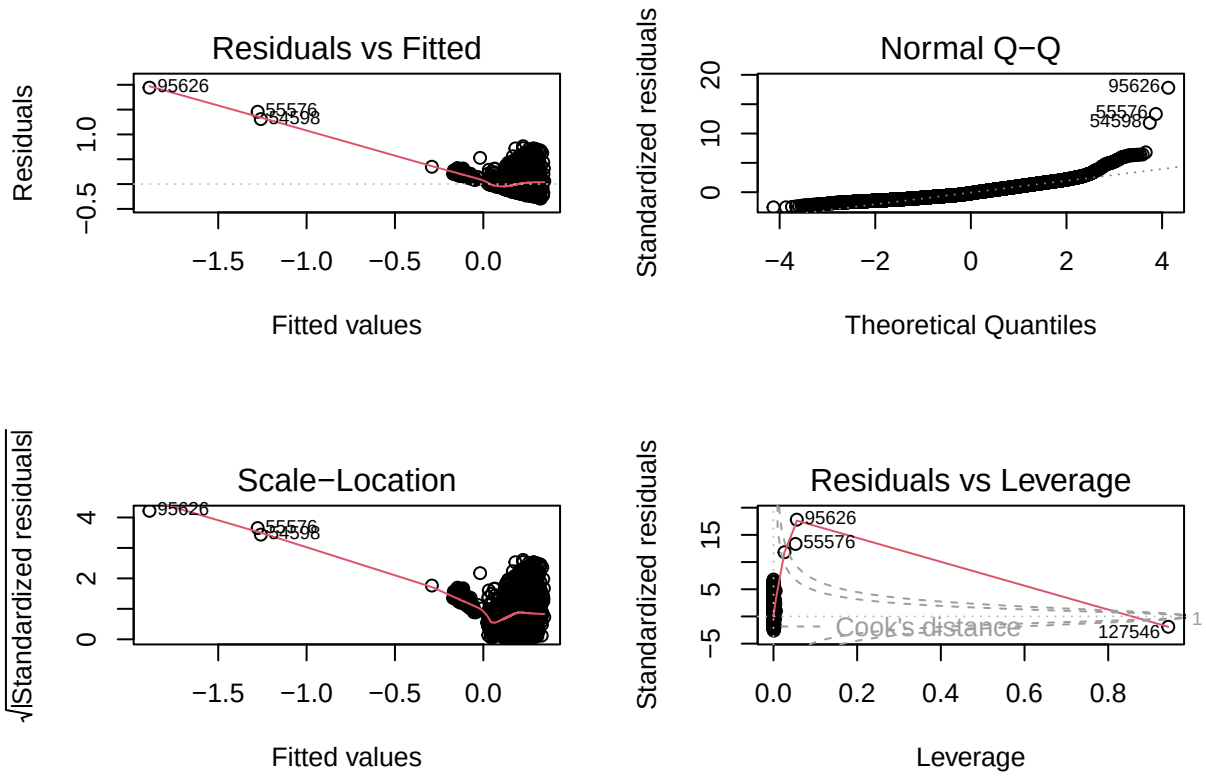


Build a third linear regression model

```
lm3 <- lm(albedo ~ q + H + a + e, data = train)
summary(lm3)
```

```
##
## Call:
## lm(formula = albedo ~ q + H + a + e, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28890 -0.08236 -0.01891  0.06949  1.94197
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.826527   0.012348   66.94   <2e-16 ***
## q           -0.226296   0.004221  -53.61   <2e-16 ***
## H           -0.013695   0.000608  -22.53   <2e-16 ***
## a            0.070159   0.002838   24.72   <2e-16 ***
## e           -0.781876   0.016327  -47.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1123 on 27474 degrees of freedom
##   (25017 observations deleted due to missingness)
## Multiple R-squared:  0.1891, Adjusted R-squared:  0.189
## F-statistic:  1602 on 4 and 27474 DF,  p-value: < 2.2e-16
```

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



Model Comparison:

After seeing the three models, in terms of Residuals vs Fitted graph, we can see that none of them are really linear but the second model is closest to it since the first and third model has a triangle shape near the end that reaches far away from the red line. the Normal Q-Q graph looks best in the third model, which included "a" and "e" variables into account. the dotted line is almost all the way covered, compared to the first two models that curved up near the end. similarly with the Scale-Location graph, we can see that the third model had a more cluster and a smaller variance. But the third model has a strange cook's distance line shown in the Residuals vs Leverage graph due to a instance, which is on top of the line. most of the plot points are on the cook's distance so this model proves that there are not many instances that influence the entire model.

The third model also showed highest adjusted R-squared value of 0.1758, and all variable data shows significance as predictors. Hence I think the third model, a curve between albedo and q with other variables such as a,e, and H provides the best linear model results.

Using 3 models, Predict and Evaluate on the test data using metrics correlation and MSE

Model 1

```
pred1 <- predict(lm1, newdata=test)
cor1 <- cor(pred1, test$q)
```

```r
mse1 <- mean((pred1-test$q)^2)
print(paste('correlation:', cor1))
```

```
## [1] "correlation: -1"
```

```r
print(paste('Mean Squared Error:', mse1))
```

```
## [1] "Mean Squared Error: 5.80683513215284"
```

Model 2

```r
pred2 <- predict(lm2, newdata=test)
cor2 <- cor(pred2, test$q)
mse2 <- mean((pred2-test$q)^2)
print(paste('correlation:', cor2))
```

```
## [1] "correlation: -0.995065675911355"
```

```r
print(paste('Mean Squared Error:', mse2))
```

```
## [1] "Mean Squared Error: 5.83318280474776"
```

Model 3

```r
pred3 <- predict(lm3, newdata=test)
cor3 <- cor(pred3, test$q)
mse3 <- mean((pred3-test$q)^2)
print(paste('correlation:', cor3))
```

```
## [1] "correlation: -0.881018522148435"
```

```r
print(paste('Mean Squared Error:', mse3))
```

```
## [1] "Mean Squared Error: 5.88902735330519"
```

Here we can see the correlation has gone from -1 to -0.99 to 0.94 for each new model, and the mean squared error increased So my prediction that model 3 was the best might not be completely correct.