Linear Regression Appliance Energy Data

Yazan Abughazaleh

2/13/2023

Introduction

The purpose of this notebook is to demonstrate the basic process of performing linear regression on a data set in R. For this notebook, I have selected the Appliance Energy data set from https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction. Performing linear regression involves finding the $\hat{w}$ and $\hat{b}$ coefficients for the linear equation:

$$\hat{b} = \bar{y} - \hat{w}\bar{x}$$

The use of this equation allows a target y to be predicted from predictors x, however these predictions only work for quantitative values. The coefficients are the slope and the intercept of a linear equation respectively. Linear regression works best when the data follows a linear pattern and there is low variance in the data. The linear regression algorithm has a high bias because it assumes that the data will fit a linear pattern.

Data Exploration

The first step in performing linear regression is to perform data exploration to determine whether regression can be performed. The Appliance Energy data is first read into a data frame, which will then be split into a train and test set. We can then look at the structure of the data to identify possible targets and predictors. We are also able to view the dimension of the data frame.

```
df <- read.csv("energydata_complete.csv", header = TRUE)

set.seed(3)
i <- sample(1:nrow(df), nrow(df) * 0.80,replace=FALSE)
train <-df[i,]
test <- df[-i,]
str(train)
```

```
## 'data.frame':    15788 obs. of  29 variables:
##  $ date       : chr  "2016-03-28 04:20:00" "2016-02-27 04:10:00" "2016-02-28 05:10:00" "2016-03-08 1
##  $ Appliances : int  60 50 70 50 120 60 40 40 40 40 ...
##  $ lights     : int  0 0 0 0 0 30 0 0 0 0 ...
##  $ T1         : num  21.5 20.1 20.2 19.4 24.8 ...
##  $ RH_1       : num  38.2 36.6 35.2 37.6 41.6 ...
##  $ T2         : num  18.8 18.3 18.2 17.6 23.8 ...
##  $ RH_2       : num  41.6 37.3 36 40 39.3 ...
##  $ T3         : num  22.7 20.5 20.6 20.2 25.4 ...
##  $ RH_3       : num  38.2 37.3 36.6 35.8 38.1 ...
##  $ T4         : num  20 19.2 18.9 18.7 24.3 ...
##  $ RH_4       : num  39 35.1 33.5 35.8 39.8 ...
##  $ T5         : num  19.9 18.6 17.6 17.6 23.8 ...
```

```
##  $ RH_5       : num  47.5 56.3 50.6 45 44.3 ...
##  $ T6         : num  7.56 0.167 -0.55 2.56 15.033 ...
##  $ RH_6       : num  56.7 68.9 60.2 79.6 1 ...
##  $ T7         : num  21.4 18.8 19.5 17.9 23.5 ...
##  $ RH_7       : num  38.2 35.8 34.3 32.4 33.9 ...
##  $ T8         : num  23.1 20.2 21.2 19.5 24.7 ...
##  $ RH_8       : num  43 44.3 40.8 39.5 40 ...
##  $ T9         : num  20.3 17.7 18 17.4 22.7 ...
##  $ RH_9       : num  42.7 40.6 39.8 36.8 37.8 ...
##  $ T_out      : num  8.53 1.3 -0.1 2.13 13.6 ...
##  $ Press_mm_hg: num  744 750 755 757 758 ...
##  $ RH_out     : num  75.3 82.7 77.7 95.5 65.3 ...
##  $ Windspeed  : num  10 3 5.17 2.83 5.67 ...
##  $ Visibility : num  48.3 20.8 24.3 54.3 32.7 ...
##  $ Tdewpoint  : num  4.27 -1.35 -3.55 1.48 7.2 ...
##  $ rv1        : num  25.44 26.31 24.05 2.03 1.28 ...
##  $ rv2        : num  25.44 26.31 24.05 2.03 1.28 ...
```

```
dim(train)
```

```
## [1] 15788    29
```

Next, we can view a summary of the data, which gives us statistics for each attribute in the data set.

```
summary(train)
```

```
##      date              Appliances          lights             T1
##  Length:15788       Min.   :  10.00   Min.   : 0.000   Min.   :16.79
##  Class :character   1st Qu.:  50.00   1st Qu.: 0.000   1st Qu.:20.79
##  Mode  :character   Median :  60.00   Median : 0.000   Median :21.60
##                     Mean   :  97.45   Mean   : 3.847   Mean   :21.68
##                     3rd Qu.: 100.00   3rd Qu.: 0.000   3rd Qu.:22.60
##                     Max.   :1080.00   Max.   :50.000   Max.   :26.26
##       RH_1             T2             RH_2             T3
##  Min.   :27.02   Min.   :16.10   Min.   :20.46   Min.   :17.20
##  1st Qu.:37.40   1st Qu.:18.79   1st Qu.:37.93   1st Qu.:20.79
##  Median :39.66   Median :20.00   Median :40.50   Median :22.10
##  Mean   :40.28   Mean   :20.33   Mean   :40.45   Mean   :22.26
##  3rd Qu.:43.06   3rd Qu.:21.50   3rd Qu.:43.29   3rd Qu.:23.29
##  Max.   :63.36   Max.   :29.86   Max.   :56.03   Max.   :29.24
##       RH_3             T4             RH_4             T5
##  Min.   :28.77   Min.   :15.10   Min.   :28.14   Min.   :15.33
##  1st Qu.:36.90   1st Qu.:19.50   1st Qu.:35.56   1st Qu.:18.28
##  Median :38.56   Median :20.60   Median :38.43   Median :19.39
##  Mean   :39.26   Mean   :20.84   Mean   :39.05   Mean   :19.59
##  3rd Qu.:41.76   3rd Qu.:22.10   3rd Qu.:42.13   3rd Qu.:20.60
##  Max.   :50.16   Max.   :26.20   Max.   :51.09   Max.   :25.80
##       RH_5             T6              RH_6             T7
##  Min.   :29.82   Min.   :-6.065   Min.   : 1.00   Min.   :15.39
##  1st Qu.:45.50   1st Qu.: 3.592   1st Qu.:30.40   1st Qu.:18.70
##  Median :49.10   Median : 7.293   Median :55.47   Median :20.02
##  Mean   :51.03   Mean   : 7.882   Mean   :54.81   Mean   :20.26
##  3rd Qu.:53.72   3rd Qu.:11.204   3rd Qu.:83.30   3rd Qu.:21.60
```

```
##  Max.   :96.32   Max.   :28.236   Max.   :99.90   Max.   :26.00
##       RH_7             T8             RH_8             T9             RH_9
##  Min.   :23.2   Min.   :16.31   Min.   :29.60   Min.   :14.89   Min.   :29.17
##  1st Qu.:31.5   1st Qu.:20.79   1st Qu.:39.09   1st Qu.:18.00   1st Qu.:38.53
##  Median :34.9   Median :22.12   Median :42.40   Median :19.39   Median :40.90
##  Mean   :35.4   Mean   :22.03   Mean   :42.96   Mean   :19.48   Mean   :41.57
##  3rd Qu.:39.0   3rd Qu.:23.39   3rd Qu.:46.59   3rd Qu.:20.60   3rd Qu.:44.33
##  Max.   :51.4   Max.   :27.23   Max.   :58.78   Max.   :24.50   Max.   :53.33
##      T_out          Press_mm_hg        RH_out          Windspeed
##  Min.   :-5.000   Min.   :729.3   Min.   : 24.00   Min.   : 0.000
##  1st Qu.: 3.646   1st Qu.:750.9   1st Qu.: 70.33   1st Qu.: 2.000
##  Median : 6.900   Median :756.0   Median : 83.67   Median : 3.667
##  Mean   : 7.393   Mean   :755.5   Mean   : 79.80   Mean   : 4.053
##  3rd Qu.:10.400   3rd Qu.:760.9   3rd Qu.: 91.67   3rd Qu.: 5.500
##  Max.   :26.033   Max.   :772.3   Max.   :100.00   Max.   :14.000
##    Visibility      Tdewpoint           rv1                rv2
##  Min.   : 1.00   Min.   :-6.600   Min.   : 0.00603   Min.   : 0.00603
##  1st Qu.:29.00   1st Qu.: 0.900   1st Qu.:12.58015   1st Qu.:12.58015
##  Median :40.00   Median : 3.467   Median :25.00667   Median :25.00667
##  Mean   :38.34   Mean   : 3.755   Mean   :25.02091   Mean   :25.02091
##  3rd Qu.:40.00   3rd Qu.: 6.567   3rd Qu.:37.56472   3rd Qu.:37.56472
##  Max.   :66.00   Max.   :15.317   Max.   :49.99653   Max.   :49.99653
```

Lastly, we can use the cor() function to find the correlation between a target and its predictors to determine which predictors are best to use for a target. Since the model I want to create is trying to predict appliance energy use. my target is the appliance column in the data, and I am trying to find the best predictors to use. A correlation value which is closest to +1 or -1 is typically the best indicator of which attributes make good predictors. To find the best two predictors in this case, I am using the max and min functions to find the best correlations.

```
cor(train[4:27],train[2])
```

```
##                Appliances
## T1           0.0602138320
## RH_1         0.0840747361
## T2           0.1252324431
## RH_2        -0.0648261908
## T3           0.0881817050
## RH_3         0.0319282451
## T4           0.0442896092
## RH_4         0.0143318509
## T5           0.0223431110
## RH_5         0.0035574856
## T6           0.1207345793
## RH_6        -0.0855636048
## T7           0.0285093156
## RH_7        -0.0568853172
## T8           0.0435641120
## RH_8        -0.0968579921
## T9           0.0139360861
## RH_9        -0.0548021602
## T_out        0.1030316418
## Press_mm_hg -0.0407055966
```

```
## RH_out      -0.1558767118
## Windspeed    0.0820941196
## Visibility  -0.0005350116
## Tdewpoint    0.0167823153
```

```
correl <- cor(train[4:27],train[2])
max(correl)
```

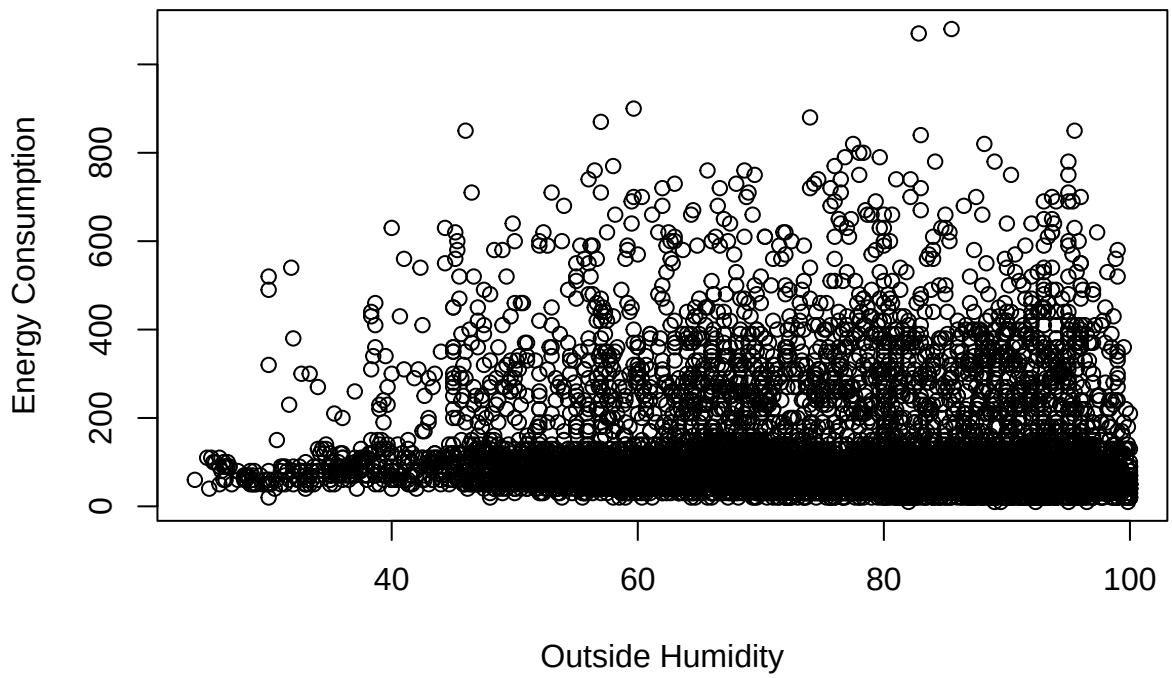
```
## [1] 0.1252324
```

```
min(correl)
```

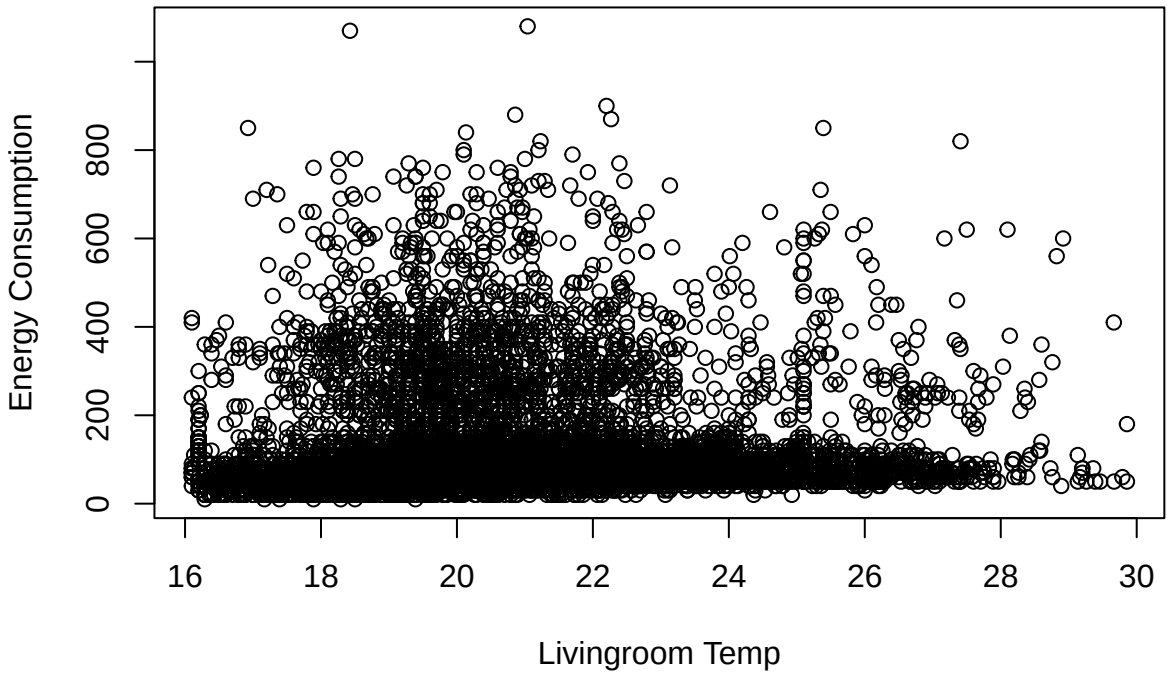
```
## [1] -0.1558767
```

Visualizing the data can be done using the plot function in R.

```
plot(train$RH_out,train$Appliances ,
     xlab="Outside Humidity", ylab="Energy Consumption")
```




```
plot(train$T2,train$Appliances , xlab = "Livingroom Temp", ylab = "Energy Consumption")
```



Above we can see two plots that relate the outside humidity to the appliance energy consumption and the living room temperature to the appliance energy consumption. While there isn't a clear linear correlation in the data, we can still attempt to perform linear regression and evaluate the results.

Creating a Model

Now, it is possible to create a simple regression model on the train set of the data. I am using the outside humidity as a predictor for the energy consumption.

```
lm1 <- lm(train$Appliances~RH_out,data=train)
summary(lm1)
```

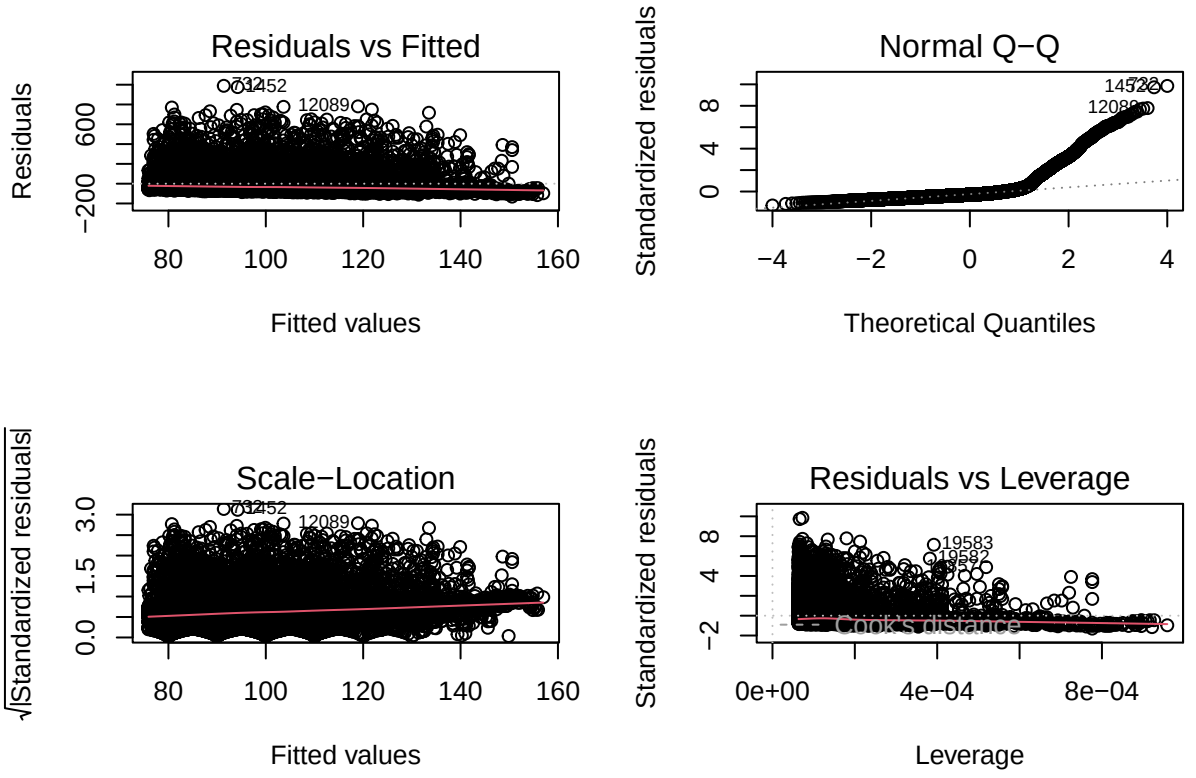
```
##
## Call:
## lm(formula = train$Appliances ~ RH_out, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -130.59  -45.68  -30.17   -2.93  988.63
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 182.59339    4.36771   41.80   <2e-16 ***
## RH_out       -1.06692    0.05381  -19.83   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 100.3 on 15786 degrees of freedom
## Multiple R-squared:  0.0243, Adjusted R-squared:  0.02424
## F-statistic: 393.1 on 1 and 15786 DF,  p-value: < 2.2e-16
```

Based off of the summary, we can see that the model does not account for the variance in the data as indicated by the very low R-squared value, which typically should be closer to 1. The F statistic however is good in this model, since it is greater that 1 and the associated p value is very low. This means that the predictor of outside humidity is significant in predicting the appliance energy use.

Next, we want to plot the residuals of our model.

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



The first plot, "Residuals vs Fitted" helps determine if the data fits a linear relationship or not. The data does not have a good spread around the line but seems to follow the line rather consistently until the end. This seems tho show that the data does not properly fit the linear model. In the "Normal Q-Q" plot, we are looking to see that the residuals are distributed normally. The residuals should closely align with the dashed line. In the plot above, we can see that the residuals line up with the dashed line but then drastically deviate, showing that our residuals are not normally distributed. The "Scale-Location" plot shows a relative concentration of data points on the left side of the graph, which thins out on the right side. This shows that there is a lack of equal variance in the residuals, indicating that the residuals are not evenly spread among predictors. The final plot helps indicate where there are influential data points. Shown in the plot, most data points have a lower leverage meaning removing them will not be very impactful to the model.

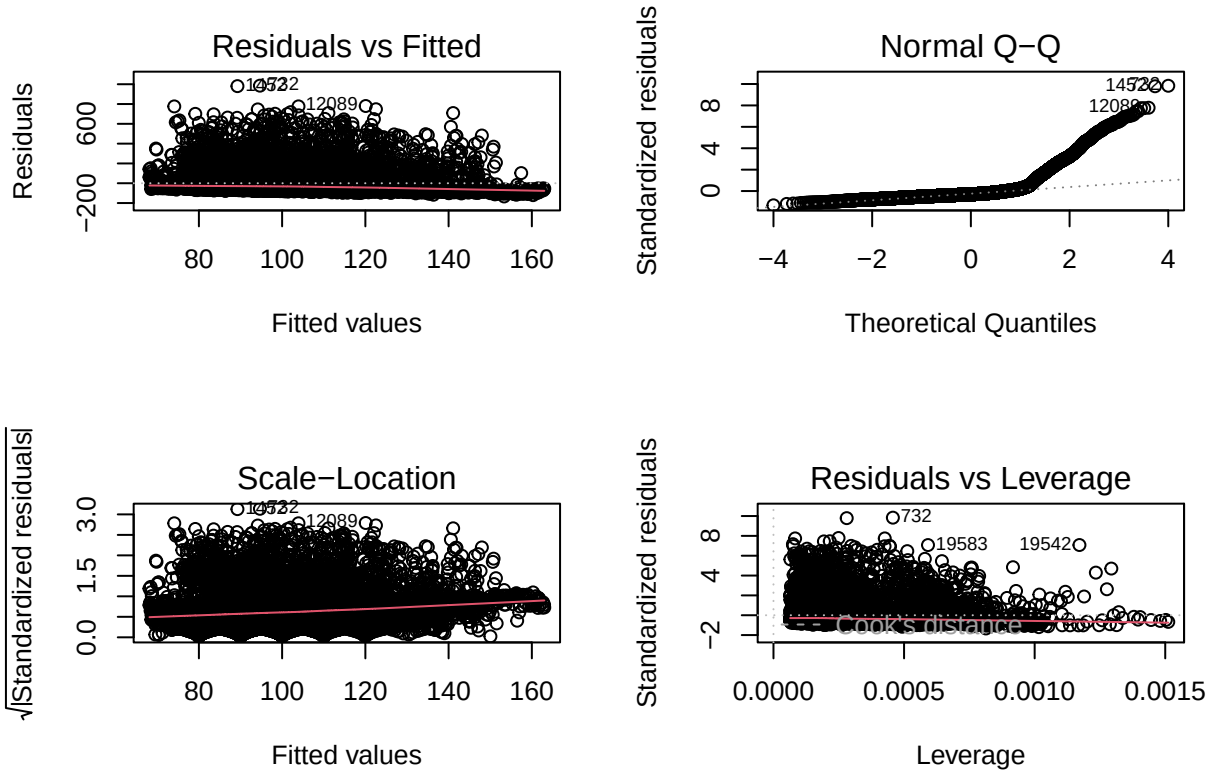
Multiple Linear Regression Model

Next, we are adding more predictors to our model to determine if it performs better.

```
lm2 <- lm(train$Appliances~RH_out+T2+T6, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = train$Appliances ~ RH_out + T2 + T6, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -133.34  -45.27  -30.41   -3.46  985.33
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 105.646158  13.136254   8.042 9.44e-16 ***
## RH_out       -0.850999   0.065527 -12.987  < 2e-16 ***
## T2            2.936694   0.612665   4.793 1.66e-06 ***
## T6            0.001172   0.230842   0.005    0.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 100.2 on 15784 degrees of freedom
## Multiple R-squared:  0.02726,    Adjusted R-squared:  0.02708
## F-statistic: 147.4 on 3 and 15784 DF,  p-value: < 2.2e-16
```

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

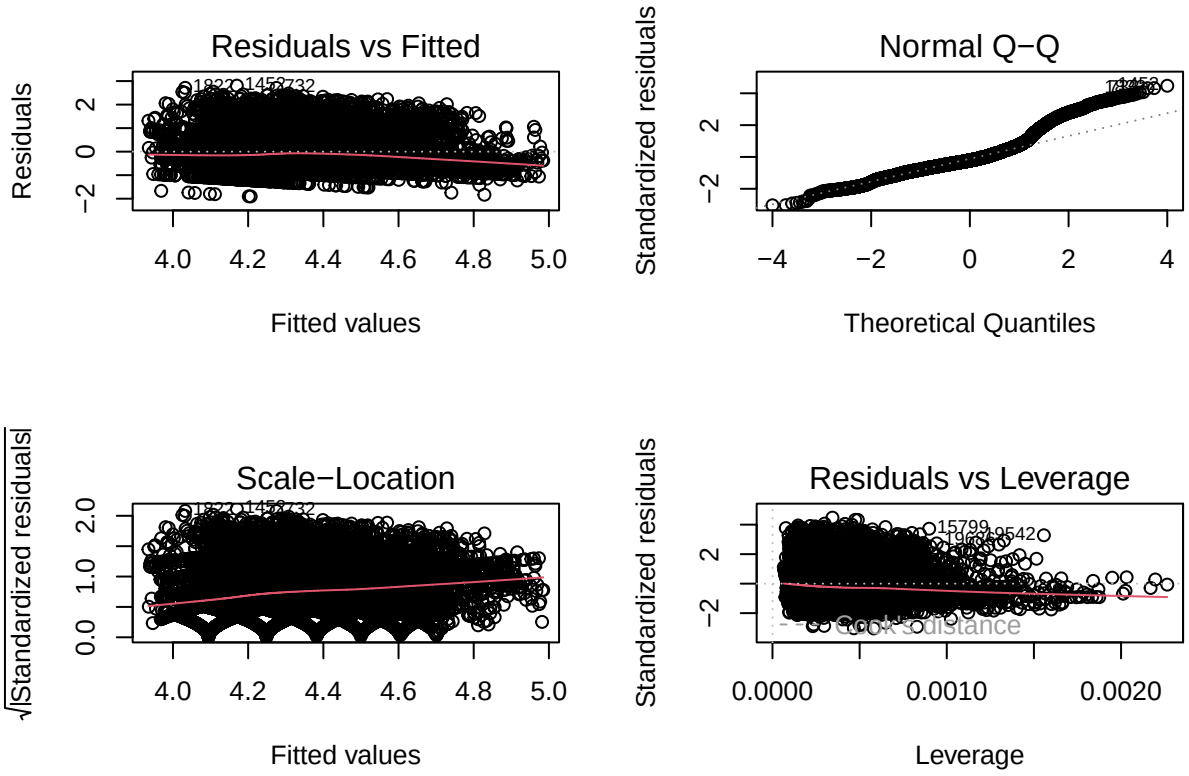



```
#lm3 <- lm(train$Appliances~train$RH_out+train$T2+train$T6+train$Windspeed+train$T_out, data=train)
lm3 <- lm(log(train$Appliances)~RH_out+T2+T6+Windspeed+T_out, data=train)
summary(lm3)
```

```
##
## Call:
## lm(formula = log(train$Appliances) ~ RH_out + T2 + T6 + Windspeed +
##     T_out, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9042 -0.3743 -0.1393  0.2313  2.8056
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.8218267  0.0850905  44.915   <2e-16 ***
## RH_out      -0.0071740  0.0004141 -17.326   <2e-16 ***
## T2           0.0518771  0.0038932  13.325   <2e-16 ***
## T6           0.0394227  0.0037873  10.409   <2e-16 ***
## Windspeed    0.0175917  0.0021099   8.338   <2e-16 ***
## T_out       -0.0516413  0.0042967 -12.019   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6271 on 15782 degrees of freedom
```

```
## Multiple R-squared:  0.07827,    Adjusted R-squared:  0.07798
## F-statistic:   268 on 5 and 15782 DF,  p-value: < 2.2e-16
```

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



Comparing Results.

```
anova(lm1,lm2)
```

```
## Analysis of Variance Table
##
## Model 1: train$Appliances ~ RH_out
## Model 2: train$Appliances ~ RH_out + T2 + T6
##   Res.Df       RSS Df Sum of Sq      F    Pr(>F)
## 1  15786 158962216
## 2  15784 158479446  2    482770 24.041 3.758e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using the analysis of variance function, we can find that model 2 outperforms model 1 with lower RSS and P value. However we know that model 3 is the best performing by looking at the residual plots. The use of the log() in model 3 greatly improved the residuals and the R-squared value.

Predictions

Here, we are testing the model's prediction capabilities on our test data set. The metrics we are looking at are both correlation and mean squared error, which measures the average of the square of errors in the model.

```
predict1 <- predict(lm1, newdata = test)
predict2 <- predict(lm2, newdata = test)
predict3 <- predict(lm3, newdata = test)
cor1 <- cor(predict1,test$Appliances)
mse1 <-mean((predict1 - test$Appliances)^2)
rmse1 <- sqrt(mse1)
print(paste('correlation1:', cor1))
```

```
## [1] "correlation1: 0.138709809724308"
```

```
print(paste('mse1:', mse1))
```

```
## [1] "mse1: 11061.776237192"
```

```
print(paste('rmse1:', rmse1))
```

```
## [1] "rmse1: 105.17497914044"
```

```
cor2 <- cor(predict2,test$Appliances)
mse2 <-mean((predict2 - test$Appliances)^2)
rmse2 <- sqrt(mse2)
print(paste('correlation2:', cor2))
```

```
## [1] "correlation2: 0.142075851729742"
```

```
print(paste('mse2:', mse2))
```

```
## [1] "mse2: 11053.8749169467"
```

```
print(paste('rmse2:', rmse2))
```

```
## [1] "rmse2: 105.137409692966"
```

```
predict3 <- exp(predict3)
cor3 <- cor(predict3,test$Appliances)
mse3 <-mean((predict3 - test$Appliances)^2)
rmse3 <- sqrt(mse3)
print(paste('correlation3:', cor3))
```

```
## [1] "correlation3: 0.166508405120223"
```

```
print(paste('mse3:', mse3))
```

```
## [1] "mse3: 11493.226272906"
```

```
print(paste('rmse3:', rmse3))
```

```
## [1] "rmse3: 107.206465630138"
```

In the three models, we see that there is a noticeable improvement in the correlation of the data as more predictors are added. We want to e a correlation be as close to +1 or -1 as possible. Adding more predictors allows the model to account for more variations that are not accounted for by a lower number of predictors. While the third model did have the best correlation, it suffered in a higher MSE value than the other models indicating a higher presence of errors. Model three however is still better able to fit the data and thus produce better results due to its correlation. In general however, these three models are ineffective at explaining the high variance in the data and produce inaccurate results, indicating that linear regression may not be the best approach for this data set.