# Linear Models: Regression

Fredrick Horn

---

Data from: https://www.kaggle.com/datasets/budincsevity/szeged-weather (Kaggle:%20Weather%20in%20Szeged%202006-2016)

Linear Regression attempts to find a linear relationship between a dependent variable and one or more independent variables and then use that relationship to predict the dependent variable based on the independent variable or variables. The general form of linear regression is y = a + bx where y is the variable we are trying to predict and x is the variable used for prediction. a and b are what we are trying to find to fit the linear model to our data set. Linear regression is great because its simple and allows you to quantify a the relationship between your predictors and your predicted variable. But linear regression also will always try to find a linear relationship in the data, even if its not there, which tends to underfit the data or produce a model that will not perform well. It also can be impacted quite heavily by outliers that can skew the model.

## Data Exploration

The data set I chose is from Kaggle and contains weather data in Szeged, Hungary from 2006 to 2016. First, read in the data and select the relavent columns from the data. This data set had a few problems, the main one being that there is supposed to be a column on what I assume should've been "Cloud Cover" but it is instad called "Loud Cover" and is only filled with 0. A few other columns also have missing data for some days and I have choosen to omit them as they were not as relavent. I will be trying to predict the apprent temperature on a given day based on the humidity, wind speed, wind bearing, and real temperature.

```
set.seed(1)
weather <- read.csv("weatherHistory.csv", header = TRUE) # read in csv
weather <- weather[, c(4, 5, 6, 7, 8)] # select relavent columns
i <- sample(1:nrow(weather), 0.8 * nrow(weather), replace = FALSE) # split data
train <- weather[i, ] # 80% train
test <- weather[-i, ] # 20% test
str(train) # structure of training data
```

```
## 'data.frame':    77162 obs. of  5 variables:
##  $ Temperature        : num  -2.006 3.711 22.039 17.15 -0.394 ...
##  $ ApparentTemperature: num  -2.01 1.06 22.04 17.15 -4.7 ...
##  $ Humidity           : num  0.92 1 0.69 0.72 0.87 0.89 0.86 0.87 0.47 0.89 ...
##  $ WindSpeed          : num  3.62 10.34 7.79 4.99 14.06 ...
##  $ WindBearing        : num  136 21 282 223 171 112 24 27 98 308 ...
```

```
head(train) # first few lines
```

```
##       Temperature ApparentTemperature Humidity WindSpeed WindBearing
## 24388  -2.0055556           -2.005556     0.92    3.6225         136
## 59521   3.7111111            1.061111     1.00   10.3362          21
## 43307  22.0388889           22.038889     0.69    7.7924         282
## 69586  17.1500000           17.150000     0.72    4.9910         223
## 11571  -0.3944444           -4.700000     0.87   14.0553         171
## 25173   7.8444444            7.844444     0.89    3.0751         112
```

```
tail(train) # last few lines
```

```
##       Temperature ApparentTemperature Humidity WindSpeed WindBearing
## 54791   -3.266667          -10.422222     0.95   27.5793           9
## 34810   16.111111           16.111111     0.56   20.9300          30
## 92256   26.700000           26.927778     0.46   25.2770         172
## 51207   17.383333           17.383333     0.74    9.7083         301
## 27741   22.727778           22.727778     0.72    2.8658         114
## 5169     2.111111           -2.605556     0.92   20.3665          21
```

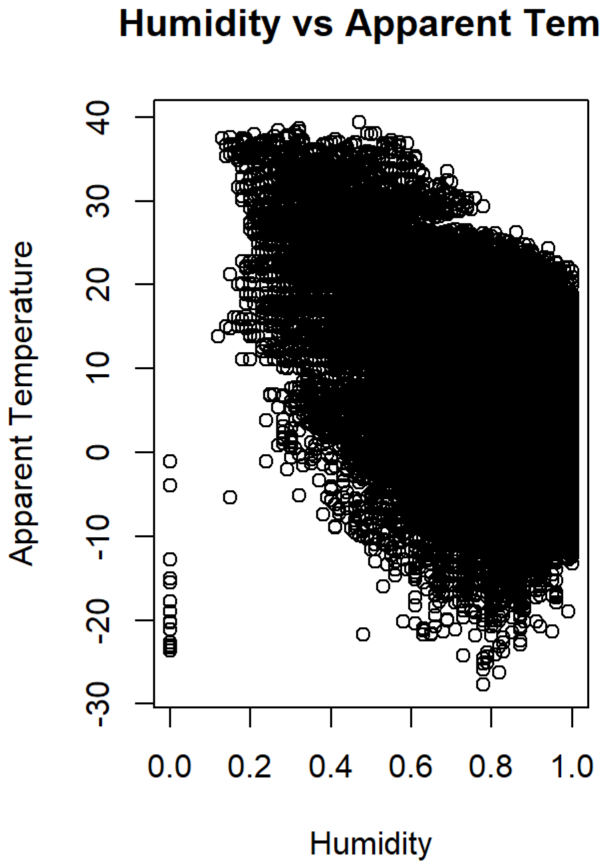
```
summary(train) # summary of data columns
```

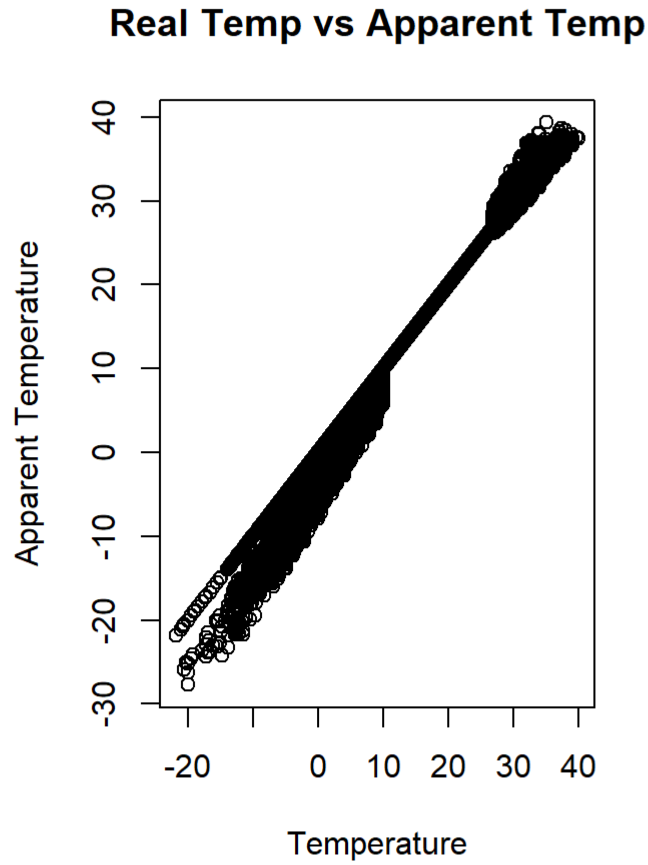
```
##   Temperature      ApparentTemperature    Humidity        WindSpeed
##  Min.   :-21.822   Min.   :-27.717     Min.   :0.0000   Min.   : 0.000
##  1st Qu.:  4.733   1st Qu.:  2.333     1st Qu.:0.6000   1st Qu.: 5.796
##  Median : 12.036   Median : 12.036     Median :0.7800   Median : 9.966
##  Mean   : 11.956   Mean   : 10.884     Mean   :0.7344   Mean   :10.802
##  3rd Qu.: 18.844   3rd Qu.: 18.844     3rd Qu.:0.8900   3rd Qu.:14.120
##  Max.   : 39.906   Max.   : 39.344     Max.   :1.0000   Max.   :63.853
##   WindBearing
##  Min.   :  0.0
##  1st Qu.:115.0
##  Median :180.0
##  Mean   :187.5
##  3rd Qu.:290.0
##  Max.   :359.0
```

# Graphs

The predictor used for the simple linear model is humidity. It does not have a strong linear relation with apparent tempurature, as shown in the first graph, but it was the best predictor of the relevant columns (excluding real temperature). The (obviously) best predictor for apprent temperature is real temperature as shown by the 2nd graph.

```
par(mfrow = c(1, 2)) # output graphs in 2x1
plot(train$Humidity, train$ApparentTemperature, xlab = "Humidity", ylab = "Apparent Temperatur
e", main = "Humidity vs Apparent Temp")
plot(train$Temperature, train$ApparentTemperature, xlab = "Temperature", ylab = "Apparent Temper
ature", main = "Real Temp vs Apparent Temp")
```




# Simple Linear Regression

This first model is only using the humidity to predict that apprent temperature. This does not perform very well as the data is not linearly correlated and does not fit the model very well as shown by the adjuested R-squared value being about 0.3633. The residuals of this model are also not great and have quite a large range in the min and max and has the largest residual standard error of the models. R reports that the humidity is at least a significant predictor of apprent temperature as is p-value is very low but its overall it is not doing a great job of predicting apparent temperature by itself.

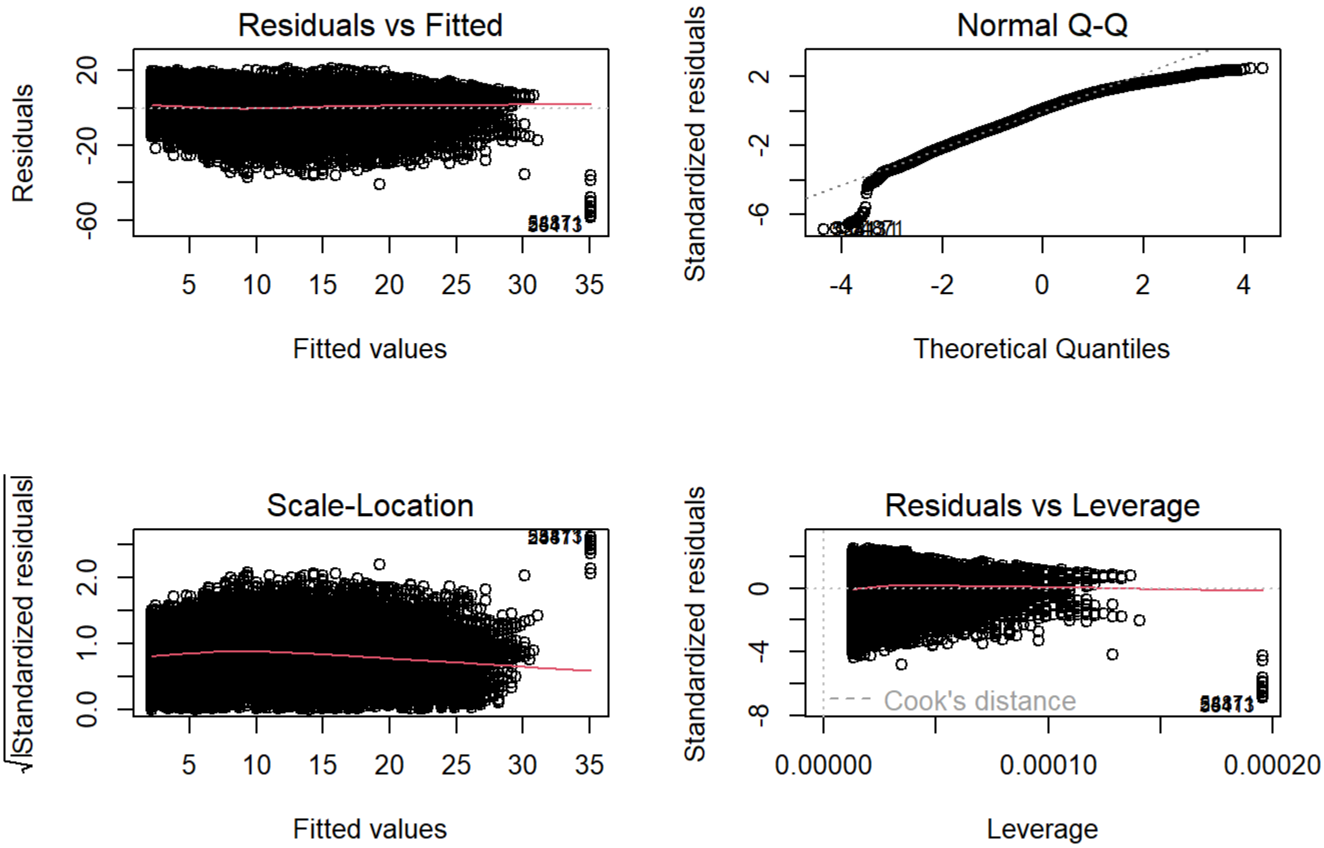
```
model <- lm(ApparentTemperature ~ Humidity, data = train) # create the first model
summary(model) # output summary
```

```
##
## Call:
## lm(formula = ApparentTemperature ~ Humidity, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.695  -5.938   0.844   6.544  21.214
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.0784     0.1193   294.0   <2e-16 ***
## Humidity    -32.9445     0.1570  -209.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.537 on 77160 degrees of freedom
## Multiple R-squared:  0.3633, Adjusted R-squared:  0.3633
## F-statistic: 4.404e+04 on 1 and 77160 DF,  p-value: < 2.2e-16
```

# Plots For Simple Linear Model

The Residuals vs Fitted plot here shows that the model did not have any non-linear trends as all the points are clustered around the main horizantal. The Normal Q-Q plot shows that the residuals are normally distributed and they do follow a straight line quite well. Its only in the top and bottom ends that they deviate from the straight line. The Scale-Location plot shows that most of the residuals are spread evenly across all the predictors, as all the points are spread along the mostly horizantal line. Finally, The Residuals vs Leverage shows that there seems to be a few case that could be outliers, but nothing is outside of cook’s distance as the dashed lines denoting it are not even in the frame of teh graph.

```
par(mfrow = c(2, 2)) # make graphs nicer by drawing them in a 2x2 grid
plot(model) # draw 4 graphs for the model
```




# Multiple Linear Regression

Thie model now uses humidity, wind speed, and wind bearing in predicting apparent temperature. It performs a little better than the simple linear model, but still only has a slight increase in the adjusted R-squared value of 0.4041. Interestingly the residuals for this model have a wider spread in the min and max but have a slightly lower residual standard error when compared to the first model. Otherwise not much changes. R reports that all 3 predictors are at least significant but looking at the estimates, its clear that wind bearing and to a lesser degree wind speed are not affecting the strength of the model as significatly as the humidity is.

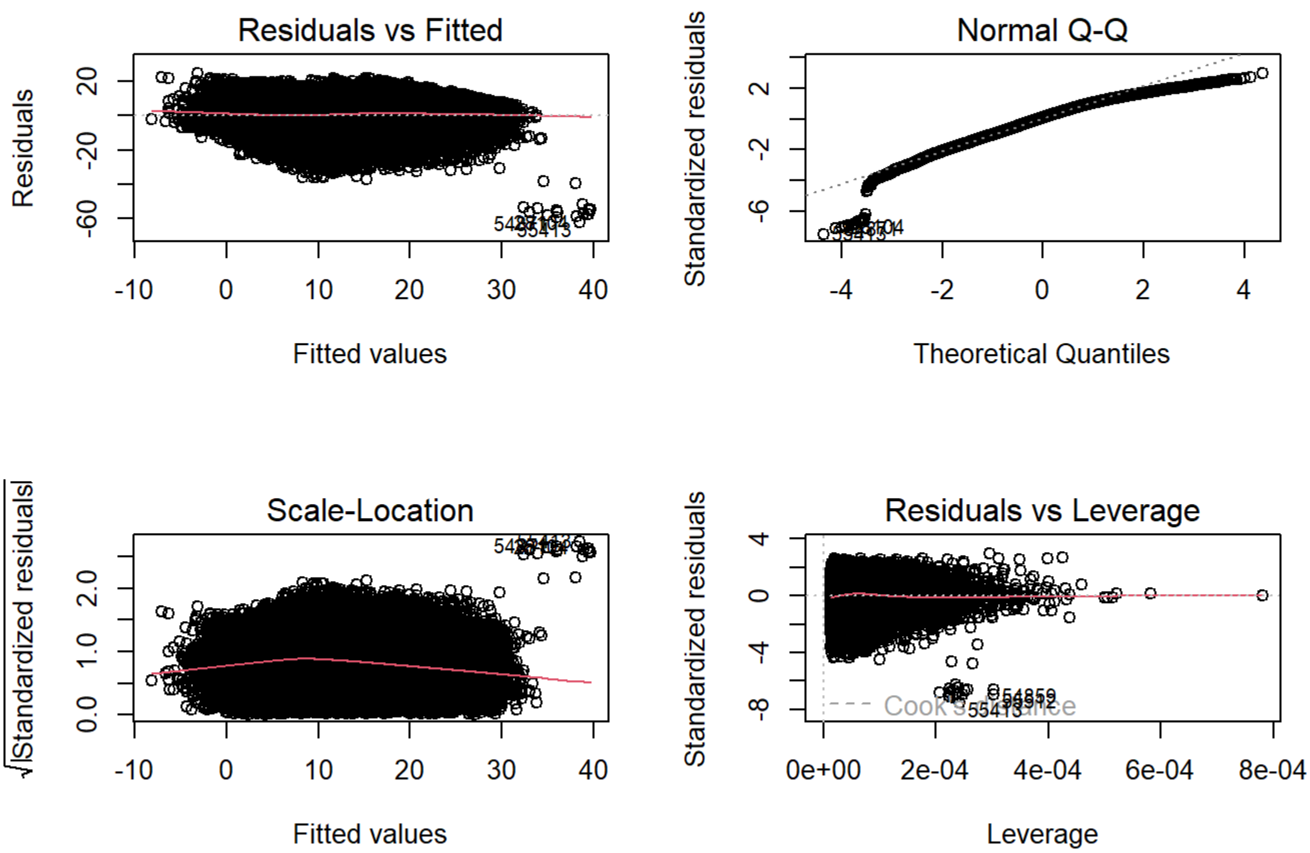
```
model2 <- lm(ApparentTemperature ~ Humidity + WindSpeed + WindBearing, data = train) # create th
e second model
summary(model2) # output summary
```

```
##
## Call:
## lm(formula = ApparentTemperature ~ Humidity + WindSpeed + WindBearing,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.132  -5.639   0.678   6.258  24.402
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.945e+01  1.438e-01  274.36   <2e-16 ***
## Humidity    -3.551e+01  1.560e-01 -227.61   <2e-16 ***
## WindSpeed   -3.192e-01  4.443e-03  -71.84   <2e-16 ***
## WindBearing  5.096e-03  2.783e-04   18.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.259 on 77158 degrees of freedom
## Multiple R-squared:  0.4041, Adjusted R-squared:  0.4041
## F-statistic: 1.744e+04 on 3 and 77158 DF,  p-value: < 2.2e-16
```

# Plots For Multiple Linear Model

In this model, we see very similar graphs to the simple linear regression. This is very likely due to the fact that the added wind speed and wind bearing, while significant statistically, in reality are not affecting the model as much as humidity does.

```
par(mfrow = c(2, 2)) # make graphs nicer by drawing them in a 2x2 grid
plot(model2) # draw 4 graphs for the model
```




# Significantly Better Multiple Linear Regression

This final model is the best performing by a significant margin. The adjusted R-squared value is 0.9898 which is most definatly due to the addition of real temperature as a predictor. Obviously real temperature is the most significant predictor of apprent temperature. It shows in the residuals have a muich smaller range in the min and max as well as having a significatly smaller residual standard error. Clearly, compared to temperature and humidity, wind speed and wind bearing are not great predictors of the apprent temperature.

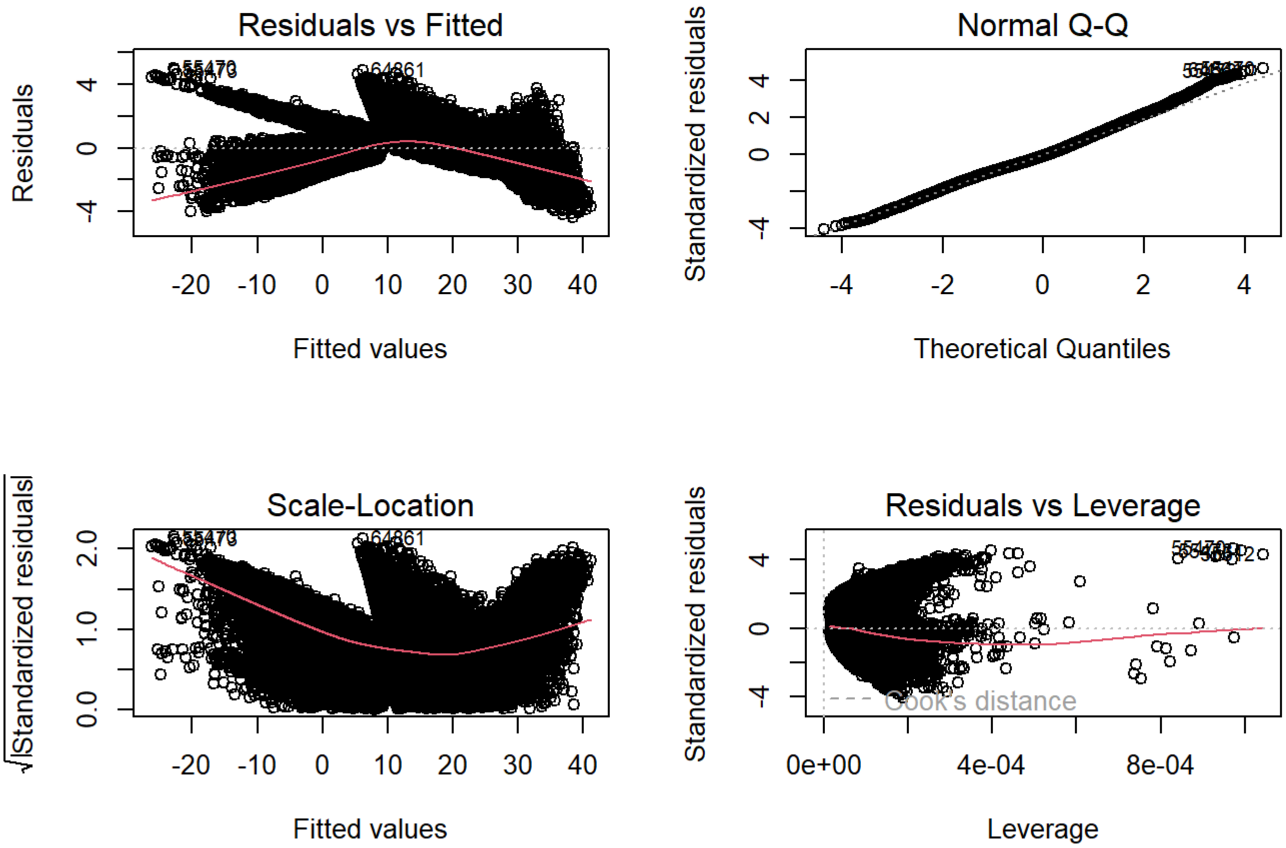
```
model3 <- lm(ApparentTemperature ~ Humidity + WindSpeed + WindBearing + Temperature, data = trai
n) # create the third model
summary(model3) # output summary
```

```
##
## Call:
## lm(formula = ApparentTemperature ~ Humidity + WindSpeed + WindBearing +
##     Temperature, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3694 -0.7152 -0.1054  0.6859  4.9969
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.399e+00  2.731e-02  -87.86   <2e-16 ***
## Humidity     1.039e+00  2.673e-02   38.88   <2e-16 ***
## WindSpeed   -9.547e-02  5.895e-04 -161.96   <2e-16 ***
## WindBearing  4.902e-04  3.639e-05   13.47   <2e-16 ***
## Temperature  1.126e+00  5.335e-04 2110.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.078 on 77157 degrees of freedom
## Multiple R-squared:  0.9898, Adjusted R-squared:  0.9898
## F-statistic: 1.881e+06 on 4 and 77157 DF,  p-value: < 2.2e-16
```

# Plots For Significantly Better Multiple Linear Model

Now these graphs look quite different to the other 2 models. First, we can see there is a parabala in the Residuals vs Fitted plot, which implies there is a non linear relationship in this data that the linear model did not catch. The Normal Q-Q plot shows that the data is very strongly normally distributed. The Scale-Location plot again shows that parabala curve and the points are definatly not distributed evenly across it. Finally the Residuals vs Leverage plot shows some potential outliers, but we still dont see cook’s distance in the frame of the graph so these are probably fine.

```
par(mfrow = c(2, 2)) # make graphs nicer by drawing them in a 2x2 grid
plot(model3) # draw 4 graphs for the model
```




# Evaluate Models On Test Set

Evaluating the models, its clear that model 3 out performs models 1 and 2 by a significant margin. This is definatly due to the temperature predictor. Model 1 and 2 both have a similar correlation and rmse that show it performs okay. But model 3 is very strongly correlated being close to 1, and has a much smaller rmse.

```
pred1 <- predict(model, newdata = test) # run prediction with model 1 on test set
cor1 <- cor(pred1, test$ApparentTemperature) # calculate correlation
mse1 <- mean((pred1 - test$ApparentTemperature)^2) # calculate mse
rmse1 <- sqrt(mse1) # calculate rmse

pred2 <- predict(model2, newdata = test) # run prediction with model 2 on test set
cor2 <- cor(pred2, test$ApparentTemperature) # calculate correlation
mse2 <- mean((pred2 - test$ApparentTemperature)^2) # calculate mse
rmse2 <- sqrt(mse2) # calculate rmse

pred3 <- predict(model3, newdata = test) # run prediction with model 3 on test set
cor3 <- cor(pred3, test$ApparentTemperature) # calculate correlation
mse3 <- mean((pred3 - test$ApparentTemperature)^2) # calculate mse
rmse3 <- sqrt(mse3) # calculate rmse

correlation <- c(cor1, cor2, cor3)
mse <- c(mse1, mse2, mse3)
rmse <- c(rmse1, rmse2, rmse3)
table <- data.frame(correlation, mse, rmse, row.names = c("Model 1", "Model 2", "Model 3"))
table # display formated output
```

```
##         correlation       mse     rmse
## Model 1   0.6016791 72.889510 8.537535
## Model 2   0.6375438 67.810363 8.234705
## Model 3   0.9948891  1.164917 1.079313
```

```
anova(model, model2, model3)
```

```
## Analysis of Variance Table
##
## Model 1: ApparentTemperature ~ Humidity
## Model 2: ApparentTemperature ~ Humidity + WindSpeed + WindBearing
## Model 3: ApparentTemperature ~ Humidity + WindSpeed + WindBearing + Temperature
##   Res.Df     RSS Df Sum of Sq       F    Pr(>F)
## 1  77160 5623006
## 2  77158 5263097  2    359909  154873 < 2.2e-16 ***
## 3  77157   89652  1   5173445 4452390 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```