Regression

Name: Gabriel Bentley

Date: 9/13/22

Dataset: Summary of Weather

Link to source

How does linear regression work?

Linear regression works by attempting to predict target quantitative values y with a set of perdictor values x of a data set. A linear line will be drawn through the data set values with a slope and intercept to show the relationship between the y and x values. Linear regression has the advantage of being simple and easy to use, but tends to have a high bias on its results.

Import the data field and seperate it into train and test sets

We will only use the MaxTemp, MinTemp, MeanTemp, year, month, and day columns of the data set.

```
df <- read.csv("Weather.csv")

keeps <- c("MaxTemp","MinTemp", "MeanTemp", "YR", "MO", "DA")
df <- df[keeps]

set.seed(1234)
i <- sample(1:nrow(df), nrow(df)*0.80, replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Explore the training data

```
str(train)
```

```
## 'data.frame':    95232 obs. of  6 variables:
##  $ MaxTemp : num  32.8 29.4 22.8 31.1 40 ...
##  $ MinTemp : num  24.4 23.9 16.1 22.2 22.8 ...
##  $ MeanTemp: num  28.9 26.7 19.4 26.7 31.1 ...
##  $ YR      : int  44 45 44 45 43 43 44 44 45 44 ...
##  $ MO      : int  12 3 4 12 5 11 4 2 11 8 ...
##  $ DA      : int  21 1 3 21 21 22 24 20 22 18 ...
```

```
summary(train)
```

```
##     MaxTemp          MinTemp          MeanTemp            YR       
##  Min.   :-32.78   Min.   :-37.78   Min.   :-35.00   Min.   :40.00  
##  1st Qu.: 25.56   1st Qu.: 15.00   1st Qu.: 20.56   1st Qu.:43.00  
##  Median : 29.44   Median : 21.11   Median : 25.56   Median :44.00  
##  Mean   : 27.05   Mean   : 17.79   Mean   : 22.42   Mean   :43.81  
##  3rd Qu.: 31.67   3rd Qu.: 23.33   3rd Qu.: 27.22   3rd Qu.:45.00  
##  Max.   : 50.00   Max.   : 33.89   Max.   : 40.00   Max.   :45.00  
##        MO               DA
##  Min.   : 1.000   Min.   : 1.0
##  1st Qu.: 4.000   1st Qu.: 8.0
##  Median : 7.000   Median :16.0
##  Mean   : 6.723   Mean   :15.8
##  3rd Qu.:10.000   3rd Qu.:23.0
##  Max.   :12.000   Max.   :31.0
```

```
head(train)
```

```
##         MaxTemp  MinTemp MeanTemp YR MO DA
## 106320 32.77778 24.44444 28.88889 44 12 21
## 106390 29.44444 23.88889 26.66667 45  3  1
## 41964  22.77778 16.11111 19.44444 44  4  3
## 15241  31.11111 22.22222 26.66667 45 12 21
## 33702  40.00000 22.77778 31.11111 43  5 21
## 101252 32.22222 22.77778 27.77778 43 11 22
```

```
names(train)
```

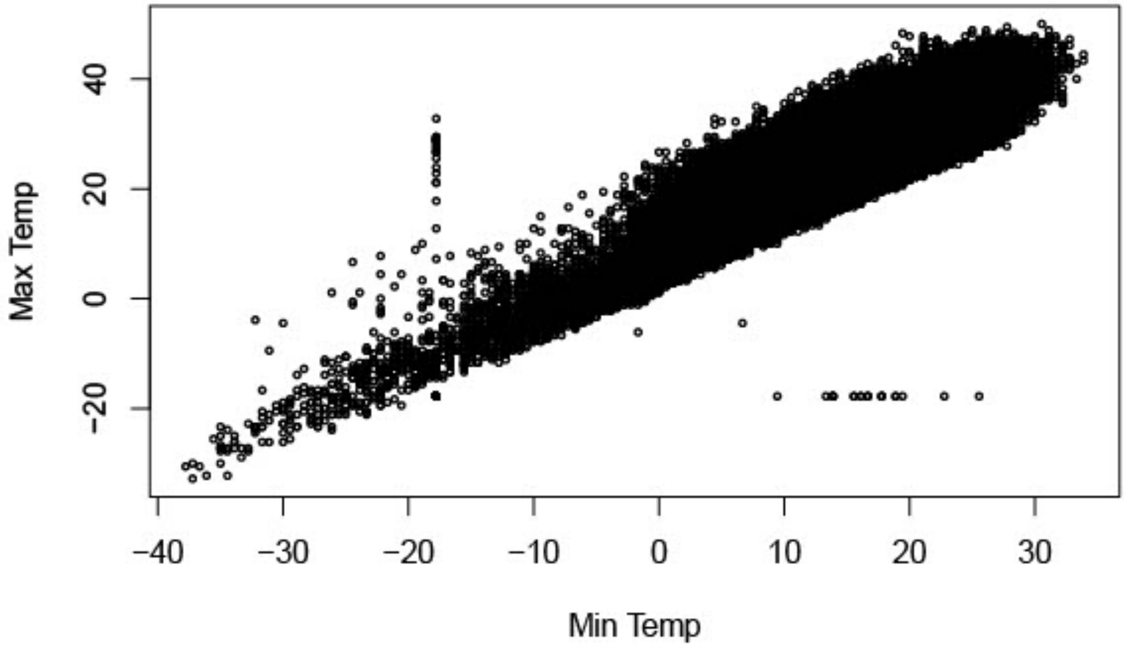
```
## [1] "MaxTemp"  "MinTemp"  "MeanTemp" "YR"       "MO"       "DA"
```

```
colSums(is.na(train))
```

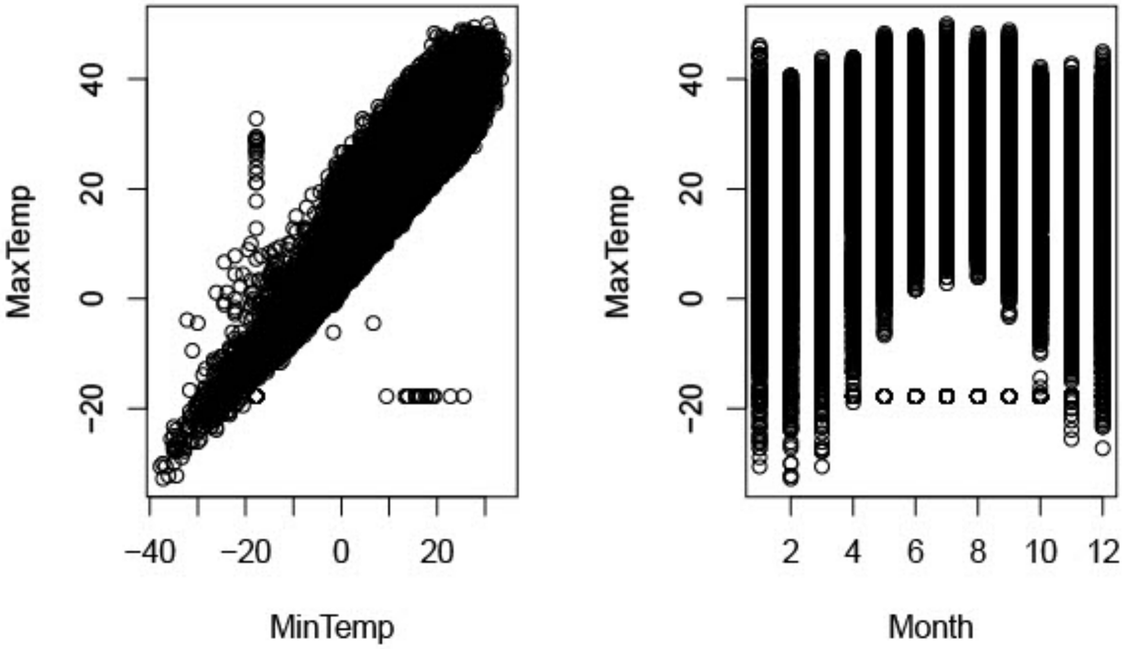
```
##  MaxTemp  MinTemp MeanTemp       YR       MO       DA
##        0        0        0        0        0        0
```

Create graphs

```
plot(train$MaxTemp~train$MinTemp, xlab="Min Temp", ylab="Max Temp", cex=0.5)
```



```
par(mfrow=c(1,2))
plot(train$MinTemp, train$MaxTemp,
     xlab="MinTemp", ylab="MaxTemp")
plot(train$MO, train$MaxTemp,
     xlab="Month", ylab="MaxTemp")
```



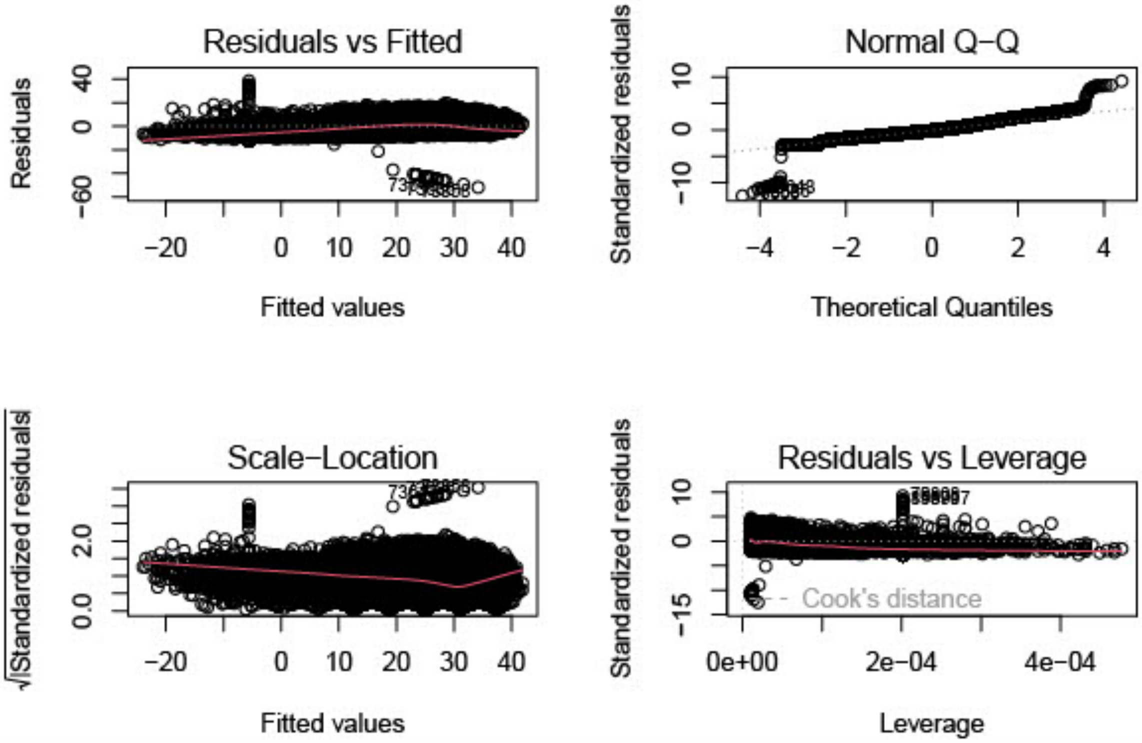
```
par(mfrow=c(1,1))
```

Build a one predicter linear regression model

```
lm1 <- lm(MaxTemp~MinTemp, data = train)
summary(lm1)
```

```
##
## Call:
## lm(formula = MaxTemp ~ MinTemp, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.954  -2.769  -0.515   2.178  38.371
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.722426   0.031734   337.9   <2e-16 ***
## MinTemp      0.917738   0.001615   568.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.157 on 95230 degrees of freedom
## Multiple R-squared:  0.7723, Adjusted R-squared:  0.7723
## F-statistic: 3.229e+05 on 1 and 95230 DF,  p-value: < 2.2e-16
```

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



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

What the summary tells us is that MinTemp is a good predictor for MaxTemp due to the 3 asterisks next to it and its low p-value and that the model is well fitted because the R-squared value is relatively close to 1. 0.77 is an ok R-squared value, but it would be better for it to be closer to 1. The low p-value and higher R-value prove that the model is good. The residual standard error was 4.157, which means that the estimated value would only be off by about 4 degrees. The F-statistic is huge, indicating that MinTemp and MaxTemp are very closely related to each other.

Plot 1: Since the red line is pretty horizontal and is closely following the dashed line, the plot shows that there is little variation not captured by the model

Plot 2: The residuals of the data are normally distributed due to them following the straight line diagonally

Plot 3: The red line is fairly horizontal with only a slight turn in it near the end, and the data points are spread out around the line equally except for a few outliers. This means that the model is mostly homoscedastic.

Plot 4: This plot shows that there are no leverage points affecting the model due to the spread out x values for the data points, but there are a few outliers in the data set affecting the model, as shown by the unusual y values at the beginning and middle of the plot.

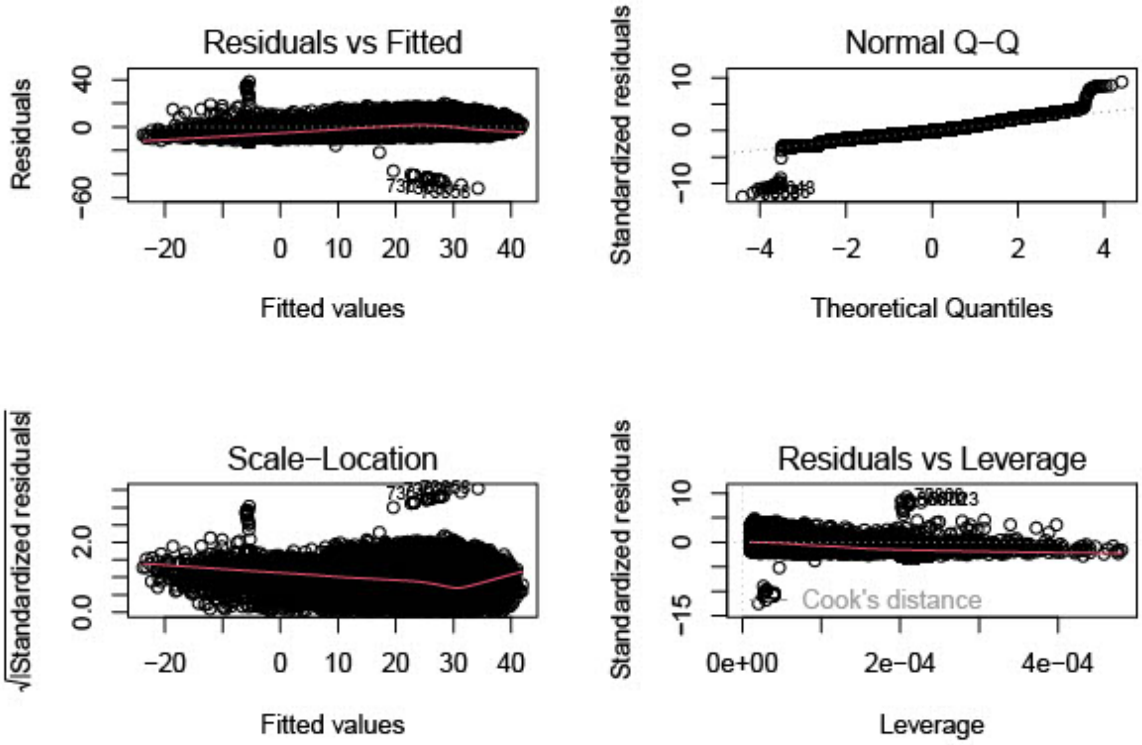
Build a multiple linear regression model

```r
lm2 <- lm(MaxTemp ~ MinTemp+MO, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = MaxTemp ~ MinTemp + MO, data = train)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.022  -2.813  -0.513   2.211  38.172
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.177568   0.040031   279.2   <2e-16 ***
## MinTemp      0.919801   0.001616   569.2   <2e-16 ***
## MO          -0.073156   0.003934   -18.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.15 on 95229 degrees of freedom
## Multiple R-squared:  0.7731, Adjusted R-squared:  0.7731
## F-statistic: 1.622e+05 on 2 and 95229 DF,  p-value: < 2.2e-16
```

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



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

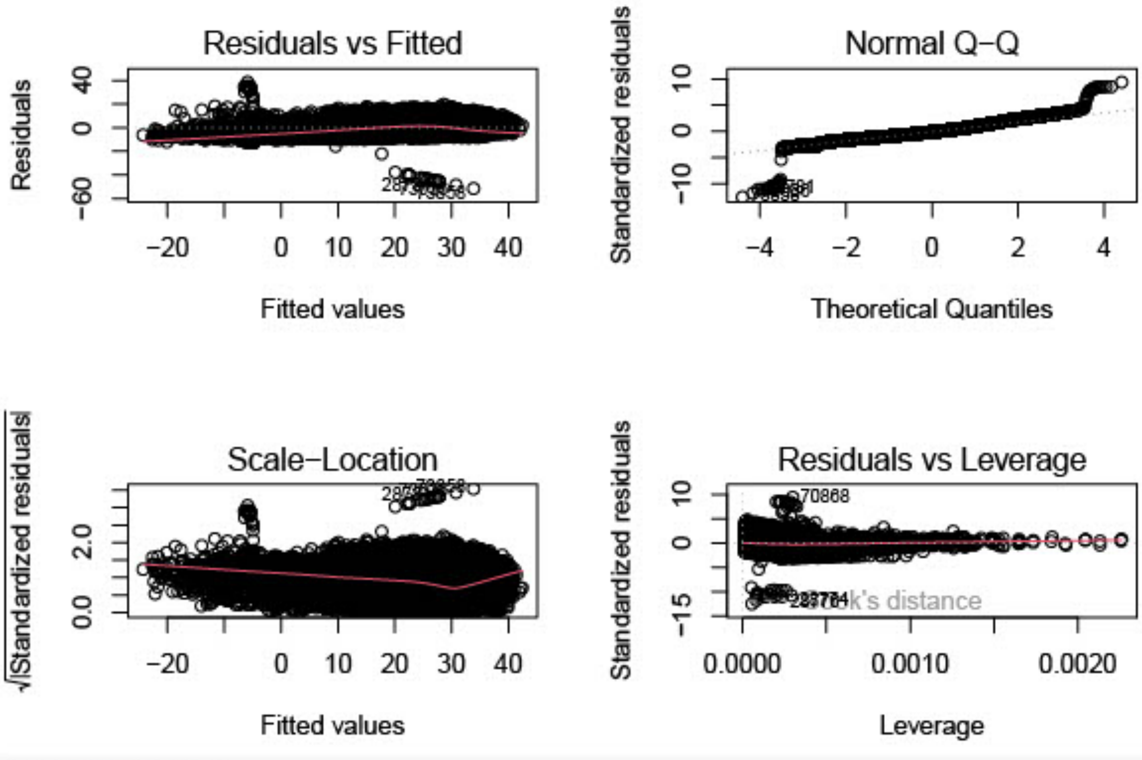
Build a third and fourth linear regression model

```
lm3 <- lm(MaxTemp ~ MinTemp + MO*YR*DA, data = train)
summary(lm3)
```

```
##
## Call:
## lm(formula = MaxTemp ~ MinTemp + MO * YR * DA, data = train)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.630  -2.799  -0.488   2.215  38.665
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.6381908  2.4158539  -3.162  0.00157 **
## MinTemp      0.9203598  0.0016065 572.897  < 2e-16 ***
## MO          -0.0371981  0.3113859  -0.119  0.90491
## YR           0.4234804  0.0549445   7.707  1.3e-14 ***
## DA          -0.2930378  0.1346807  -2.176  0.02957 *
## MO:YR        0.0001793  0.0071039   0.025  0.97986
## MO:DA        0.0454763  0.0172981   2.629  0.00857 **
## YR:DA        0.0068545  0.0030643   2.237  0.02530 *
## MO:YR:DA    -0.0010732  0.0003947  -2.719  0.00655 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.122 on 95223 degrees of freedom
## Multiple R-squared:  0.7761, Adjusted R-squared:  0.7761
## F-statistic: 4.125e+04 on 8 and 95223 DF,  p-value: < 2.2e-16
```

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

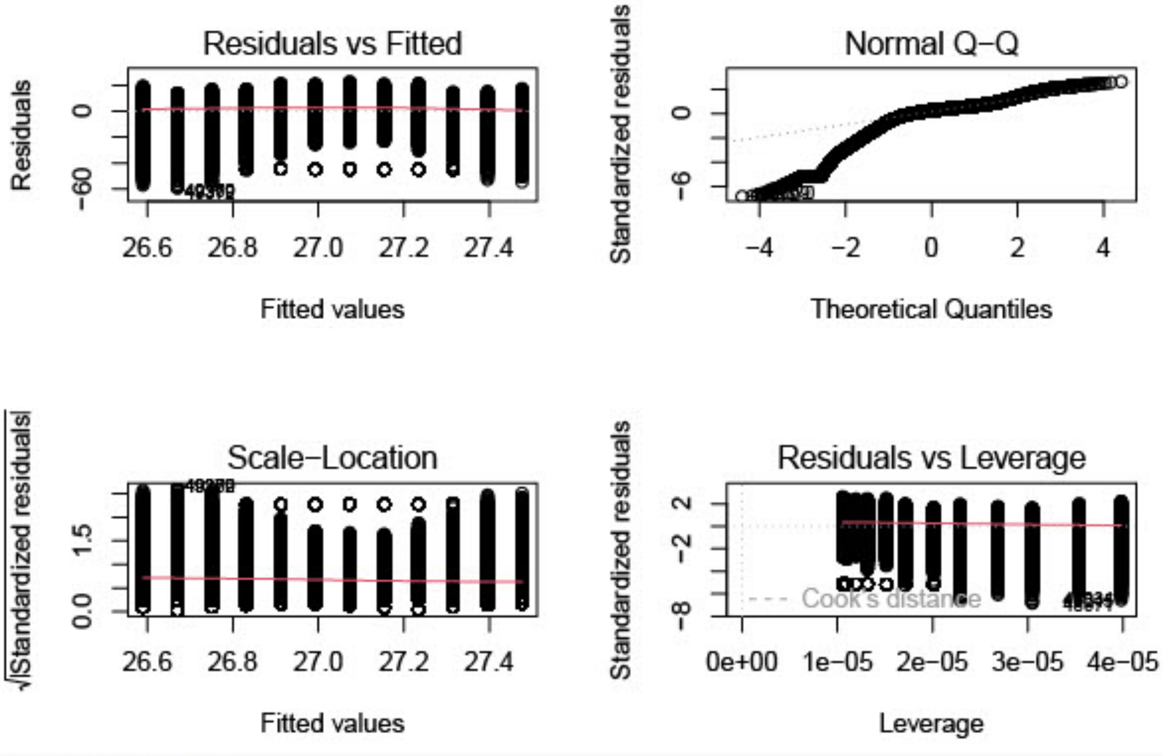


```
par(mfrow=c(1,1))

lm4 <- lm(MaxTemp ~ MO, data = train)
summary(lm4)
```

```
##
## Call:
## lm(formula = MaxTemp ~ MO, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -59.448  -1.670   2.291   4.513  22.927
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.509160   0.062141 426.598   <2e-16 ***
## MO           0.080597   0.008235   9.787   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.707 on 95230 degrees of freedom
## Multiple R-squared:  0.001005,   Adjusted R-squared:  0.0009944
## F-statistic: 95.79 on 1 and 95230 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm4)
```



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

Predict and evaluate on the test data using metrics correlation and mse. Compare the results and indicate why you think these results happened.

```
pred1 <- predict(lm1, newdata = test)
correlation1 <- cor(pred1, test$MaxTemp)
```

```r
mse1 <- mean((pred1 - test$MaxTemp)^2)
rmse1 <- sqrt(mse1)
print(paste("lm1 Correlation: ", correlation1, "lm1 Mean Square Error: ", mse1, "lm1 Root Mean Square E
```

```
## [1] "lm1 Correlation:  0.87678362849221 lm1 Mean Square Error:  17.683423641391 lm1 Root Mean Square
```

```r
pred2 <- predict(lm2, newdata = test)
correlation2 <- cor(pred2, test$MaxTemp)
mse2 <- mean((pred2 - test$MaxTemp)^2)
rmse2 <- sqrt(mse2)
print(paste("lm2 Correlation: ", correlation2, "lm2 Mean Square Error: ", mse2, "lm2 Root Mean Square E
```

```
## [1] "lm2 Correlation:  0.877367279621739 lm2 Mean Square Error:  17.6052508851512 lm2 Root Mean Squa
```

```r
pred3 <- predict(lm3, newdata = test)
correlation3 <- cor(pred3, test$MaxTemp)
mse3 <- mean((pred3 - test$MaxTemp)^2)
rmse3 <- sqrt(mse3)
print(paste("lm3 Correlation: ", correlation3, "lm3 Mean Square Error: ", mse3, "lm3 Root Mean Square E
```

```
## [1] "lm3 Correlation:  0.879211528790896 lm3 Mean Square Error:  17.3579950154381 lm3 Root Mean Squa
```

```r
pred4 <- predict(lm4, newdata = test)
correlation4 <- cor(pred4, test$MaxTemp)
mse4 <- mean((pred4 - test$MaxTemp)^2)
rmse4 <- sqrt(mse4)
print(paste("lm4 Correlation: ", correlation4, "lm4 Mean Square Error: ", mse4, "lm4 Root Mean Square E
```

```
## [1] "lm4 Correlation:  0.0299477601384579 lm4 Mean Square Error:  76.3918946565763 lm4 Root Mean Squ
```

Comparing the results of the correlations we can see that the first three models have roughly the same correlation of around .87, while the last model lm4 has a very low correlation. Additionally lm4 has a much higher mean square error than the other three models. The most likely reason for this is the use of MinTemp as a parameter in the first for models and the absence of it the the last one. From this we can conclude that the most important parameter when it comes to predicting MaxTemp from the data set is the MinTemp.