Regression

Created by John Lawler - JML190001

CS4375.004 with Karen Mazidi

Created on 2/13/2023, last worked on 2/20/2023

==========

Introduction

Linear regression is a method used to model the relationship between the response variable (dependent variable) and the predictor variable(independent variable). Linear regression attempts to find a relationship between a these variables in terms of a straight line. Along with this, it produces how accurate it was in trying to create this line in terms of R-squared and other information included when summary(lm(data)) is used. Linear regression's strengths lie in the fact that they are easy to implement and interpret, and can identify the strength of a predictor variable. On that note, multiple independent variables can be used to determine which is the best predictor variable. The weakness of this is not everything is linearly related, and sometimes a curve or other types of relationships (logarithm, normal distributions, etc.) trump the strengths of a linear regression.

Getting the File

To run any of the chunks of R script in this file, please insert *weatherAUS.csv (https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package)* into the same folder as this file. You can also find this csv on my Github (https://github.com/Billy-Budd/mensch-maschine/blob/main/blue-monday/weatherAUS.csv) repository. Any other information that pertains to other documents can be found in the read me (https://github.com/Billy-Budd/mensch-maschine#readme). Unfortunately, the prediction function would not work with NA data, so I had to remove it which probably skews the overall results of this. Major sections have headers that are bolded and there are some intermediary sections that just have headers and are not bolded.

```
df <- read.csv("weatherAUS.csv", header = TRUE)
df <- na.omit(df) # remove missing data
```

Creating an 80/20 Split

a. Divide into 80/20 train/test

We set the seed so that our split does not change between runs of the sample() function. We then separate it into a list called train (the 80 split) and test (the 20 split). Output lengths of each split. Also outputs the date of data collection start and end to get a sense of the time period of this data.

```
set.seed(1234)
split <- sample(1:nrow(df), nrow(df)*.8, replace=FALSE) # split the data into 80/20 samples
train <- df[ split, ] # set train as the  80 split
test <- df[ -split, ] # set test as the 20 split

# I used MinTemp here to show observations, but any column header would be acceptable
tmp <- c("Number of observations in train:", length(train$MinTemp),
"Number of observations in test:", length(test$MinTemp),
"Date of data collection start: ", min(df$Date),
"Date of data collection end: ", max(df$Date))
cat(tmp, sep = '\n')
```

```
## Number of observations in train:
## 45136
## Number of observations in test:
## 11284
## Date of data collection start:
## 2007-11-01
## Date of data collection end:
## 2017-06-25
```

Summary Data

b. Use at least 5 R functions for data exploration, using the training data

This is some data that shows some of the data from Australian weather from all over the country for our training data.

```
head(train)
```

	Date <chr>	Location <chr>	MinTemp <dbl>	MaxTemp <dbl>	Rainfall <dbl>	Evaporation <dbl>	Sunshine <dbl>
104049	2013-05-04	Nuriootpa	10.9	16.2	0.0	2.6	5.2
104142	2013-08-05	Nuriootpa	9.6	18.6	0.8	0.8	9.9
106046	2010-04-29	Woomera	10.4	21.3	0.0	4.6	9.7
46691	2010-11-08	Canberra	9.4	24.2	4.4	5.4	7.5
88792	2013-07-08	Cairns	22.3	27.6	0.2	7.0	8.8
94459	2012-03-23	Townsville	24.0	31.1	0.4	7.0	5.9

6 rows | 1-8 of 24 columns

```
tail(train)
```

	Date <chr>	Location <chr>	MinTemp <dbl>	MaxTemp <dbl>	Rainfall <dbl>	Evaporation <dbl>	Sunshine <dbl>
65178	2011-10-14	MelbourneAirport	6.8	25.0	0.0	4.0	11.1
139250	2008-11-16	Darwin	25.6	35.0	0.0	7.4	8.7
120115	2016-01-19	PerthAirport	12.8	25.9	0.0	5.2	5.5
106938	2012-11-06	Woomera	14.5	27.1	9.4	11.0	5.4
78695	2010-12-07	Watsonia	19.9	28.4	0.0	6.0	1.0
75329	2009-12-15	Portland	6.8	31.7	0.0	5.4	13.2

6 rows | 1-8 of 24 columns

```
summary(train)
```

```
##      Date              Location            MinTemp          MaxTemp      
##  Length:45136       Length:45136       Min.   :-6.70   Min.   : 6.30  
##  Class :character   Class :character   1st Qu.: 8.60   1st Qu.:18.70  
##  Mode  :character   Mode  :character   Median :13.20   Median :23.90  
##                                        Mean   :13.46   Mean   :24.22  
##                                        3rd Qu.:18.40   3rd Qu.:29.70  
##                                        Max.   :31.40   Max.   :48.10  
##     Rainfall        Evaporation        Sunshine     WindGustDir       
##  Min.   :  0.000   Min.   : 0.000   Min.   : 0.00   Length:45136      
##  1st Qu.:  0.000   1st Qu.: 2.800   1st Qu.: 5.10   Class :character  
##  Median :  0.000   Median : 5.000   Median : 8.60   Mode  :character  
##  Mean   :  2.148   Mean   : 5.511   Mean   : 7.73                     
##  3rd Qu.:  0.600   3rd Qu.: 7.400   3rd Qu.:10.70                     
##  Max.   :206.200   Max.   :72.200   Max.   :14.50                     
##  WindGustSpeed     WindDir9am         WindDir3pm         WindSpeed9am    
##  Min.   :  9.00   Length:45136       Length:45136       Min.   : 2.00   
##  1st Qu.: 31.00   Class :character   Class :character   1st Qu.: 9.00   
##  Median : 39.00   Mode  :character   Mode  :character   Median :15.00   
##  Mean   : 40.87                                        Mean   :15.65   
##  3rd Qu.: 48.00                                        3rd Qu.:20.00   
##  Max.   :124.00                                        Max.   :67.00   
##   WindSpeed3pm    Humidity9am      Humidity3pm      Pressure9am    
##  Min.   : 2.00   Min.   :  0.00   Min.   :  0.00   Min.   : 980.5  
##  1st Qu.:13.00   1st Qu.: 55.00   1st Qu.: 35.00   1st Qu.:1012.7  
##  Median :19.00   Median : 67.00   Median : 51.00   Median :1017.2  
##  Mean   :19.77   Mean   : 65.91   Mean   : 49.61   Mean   :1017.2  
##  3rd Qu.:26.00   3rd Qu.: 79.00   3rd Qu.: 63.00   3rd Qu.:1021.8  
##  Max.   :76.00   Max.   :100.00   Max.   :100.00   Max.   :1040.4  
##   Pressure3pm        Cloud9am        Cloud3pm        Temp9am      
##  Min.   : 977.1   Min.   :0.000   Min.   :0.000   Min.   :-0.70  
##  1st Qu.:1010.1   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:13.10  
##  Median :1014.7   Median :5.000   Median :5.000   Median :17.80  
##  Mean   :1014.8   Mean   :4.243   Mean   :4.326   Mean   :18.20  
##  3rd Qu.:1019.4   3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:23.23  
##  Max.   :1038.9   Max.   :8.000   Max.   :9.000   Max.   :39.40  
##     Temp3pm       RainToday         RainTomorrow      
##  Min.   : 4.60   Length:45136       Length:45136      
##  1st Qu.:17.40   Class :character   Class :character  
##  Median :22.40   Mode  :character   Mode  :character  
##  Mean   :22.71                                        
##  3rd Qu.:27.90                                        
##  Max.   :46.10
```

```
tmp <- c("", "Correlation between Maximum Temperature and Rainfall:", cor(train$MaxTemp, train$Rainfall), "",
        "Correlation between Humidity at 9AM and Rainfall:", cor(train$Humidity9am, train$Rainfall), "",
        "Covariance of Maximum Temperature and Rainfall:", cov(train$MaxTemp, train$Rainfall), "",
        "Covariance of Humidity at 9AM and Rainfall:", cov(train$Humidity9am, train$Rainfall), "",
        "Variance of Rainfall:", var(train$Rainfall), "",
        "Average Rainfall:", mean(train$Rainfall), "",
        "Range of Rainfall:", range(train$Rainfall))
cat(tmp, sep = '\n')
```

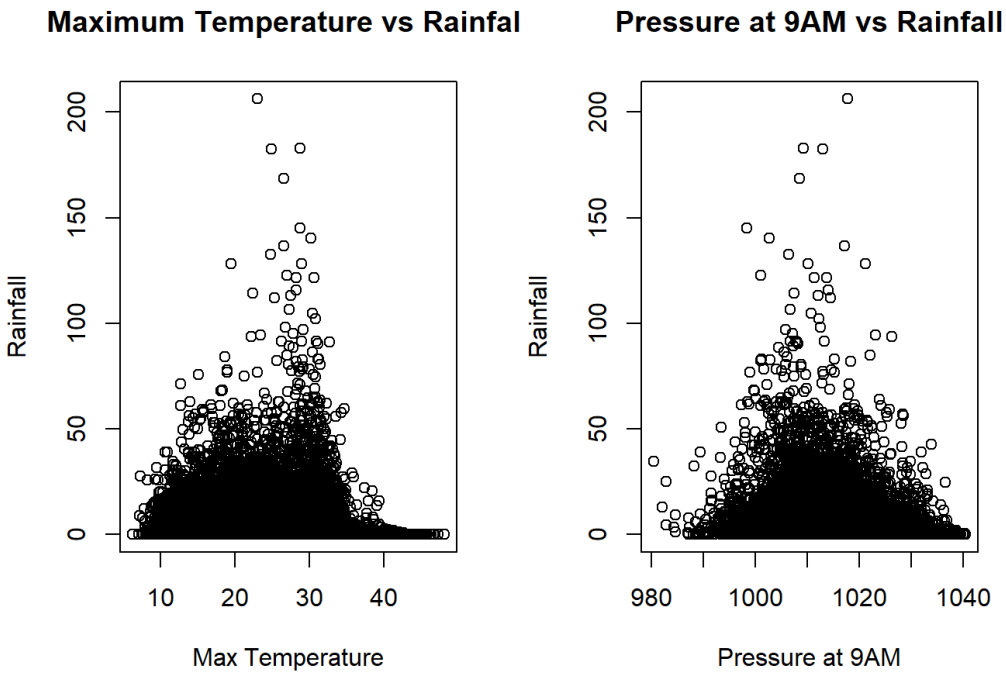
```
##
## Correlation between Maximum Temperature and Rainfall:
## -0.0694395978384363
##
## Correlation between Humidity at 9AM and Rainfall:
## 0.26291622571638
##
## Covariance of Maximum Temperature and Rainfall:
## -3.42564163918501
##
## Covariance of Humidity at 9AM and Rainfall:
## 34.5359918687162
##
## Variance of Rainfall:
## 50.3459042512268
##
## Average Rainfall:
## 2.14793956043956
##
## Range of Rainfall:
## 0
## 206.2
```

Simple Graphs

c. Create at least 2 informative graphs, using the training data

These are some simple graphs to just get an idea of what the data looks like. By the looks of these graphs, Pressure at 9AM seems to be a better predictor than maximum temperature at predicting the amount of rainfall in Australia.

```
par(mfrow=c(1,2)) # output side by side
plot(train$MaxTemp, train$Rainfall, xlab = "Max Temperature", ylab = "Rainfall", main = "Maximum Temperature vs Rainfall")
# plot 1
plot(train$Pressure9am, train$Rainfall, xlab = "Pressure at 9AM", ylab = "Rainfall", main = "Pressure at 9AM vs Rainfall")
# plot 2
```



Creating a Simple Linear Model

d. Build a simple linear regression model (one predictor) and output the summary. Write a thorough explanation of the information in the model summary.

This creates a simple model of how pressure relates to rainfall in Australia. This shows us the values of the residuals (the observed value - the

predicted value) in terms of a normal distribution to show how much deviance there is in the line of best fit. We also see information about our line in terms of the estimated slope, the standard error of that slope, the t-statistics, and the p-value. All of this data tells us how well the slope fits to our data. In this case, we have a low standard error and a low p-value, which provides us evidence that these two values are related in some way; however, the R-squared value tells us that the variance between the data is high. This is probably due to the fact that the data here is not linearly related and would be better fit for a non-linear approach. The f-statistic shows that there is some statistic significance, but we will need to do more to analyze this fact. There is some other information here, such as degrees of freedom. This is dependent on how much data we have in our set, and in this case we have many degrees of freedom because this is a larger data set. In summary, all of this information is to tell us how significant the resulting data is, how related these two values are, and how much variance/residuals there is in the data set.

```
lm1 <- lm(Pressure9am~Rainfall, data=train) # create linear model
summary(lm1) # summary
```

```
##
## Call:
## lm(formula = Pressure9am ~ Rainfall, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.910  -4.518  -0.118   4.482  36.334
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  1.018e+03  3.337e-02 30499.08   <2e-16 ***
## Rainfall    -1.758e-01  4.501e-03   -39.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.785 on 45134 degrees of freedom
## Multiple R-squared:  0.0327, Adjusted R-squared:  0.03268
## F-statistic:  1526 on 1 and 45134 DF,  p-value: < 2.2e-16
```

Residual Plots for Simple Linear Model

e. Plot the residuals and write a thorough explanation of what the residual plot tells you.

This shows us our residuals for the linear model.

The first plot, residuals vs fitted, the pattern shows us that a linear fit is probably not the best solution to capture the relevant data.

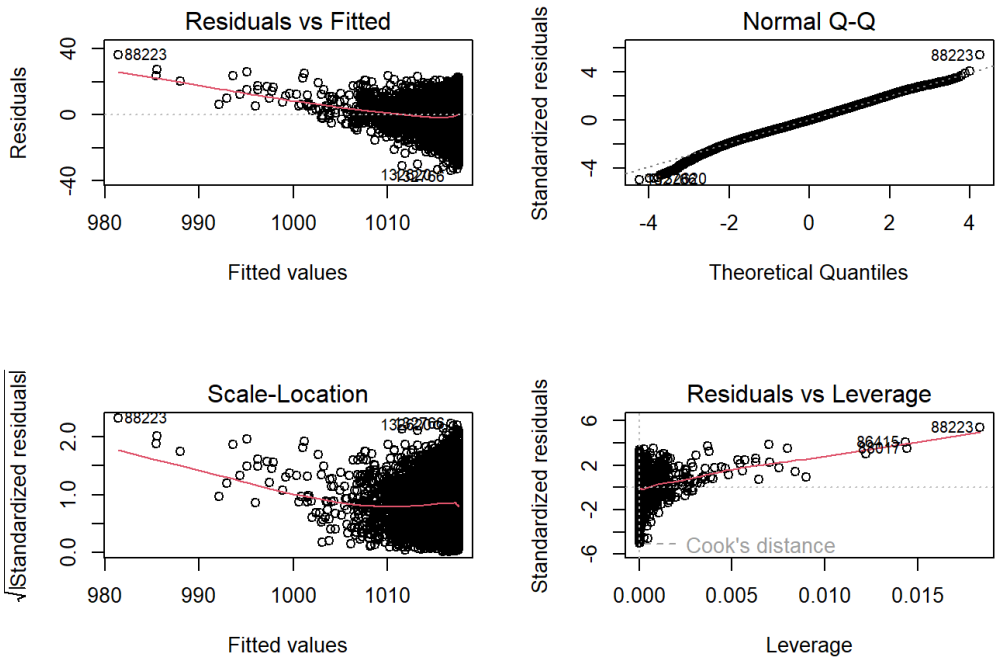
The second plot, normal q-q, shows us that the data falls very close to a normal distribution as most of the dots fall along a positive straight line.

The third plot, scale-location, attempts to show heteroscedasticity, which would mean that there would be an even distribution along the graph. This graph is heavily weighted to one side, indicating no heteroscedasticity.

The fourth plot, residuals vs leverage, shows us that for the most part, there are not many outliers. The density is mostly clustered together, showing that there are only a handful of ‘influential points.’

Judging from these plots, a **normal distribution** is the best way to go in terms of determining what would be best for data.

```
par(mfrow=c(2,2)) # plot in a 2x2 grid
plot(lm1) # create plots
```



Creating a More Complex Linear Models

f. Build a multiple linear regression model (multiple predictors), output the summary and residual plots.

Here, our R-squared increases showing that we can better predict pressure using the rainfall and windspeed. While the R-squared is still low, using the new information of windspeed allows us to better calculate pressure.

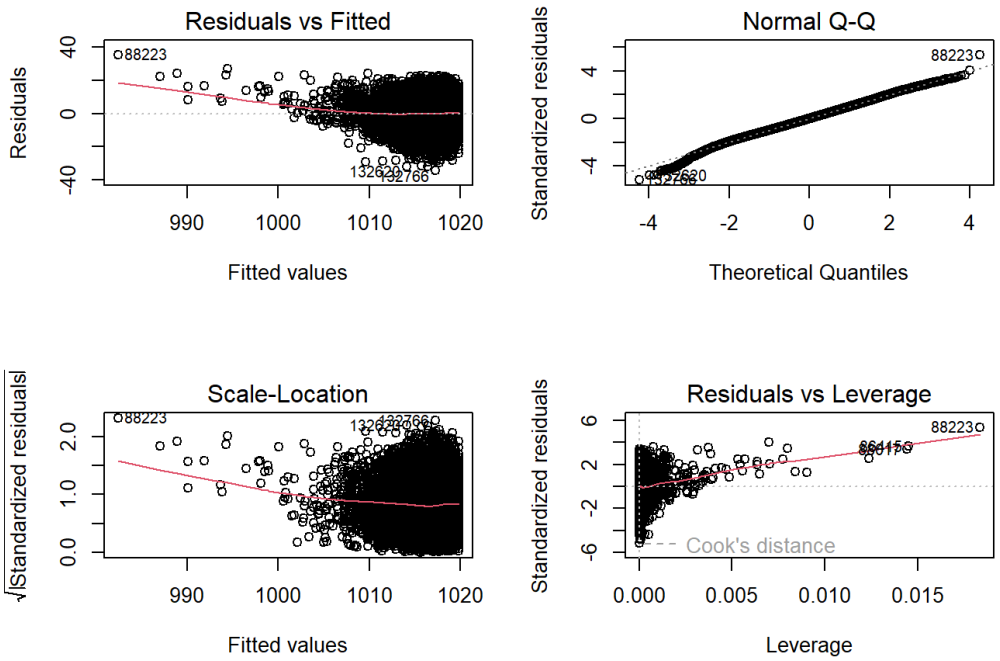
```
lm2 <- lm(Pressure9am~Rainfall + WindSpeed9am, data=train) # create linear model
summary(lm2) # summary
```

```
##
## Call:
## lm(formula = Pressure9am ~ Rainfall + WindSpeed9am, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.364  -4.515  -0.098   4.408  35.270
##
## Coefficients:
##                Estimate Std. Error  t value Pr(>|t|)
## (Intercept)   1.020e+03  6.702e-02 15222.05   <2e-16 ***
## Rainfall     -1.656e-01  4.418e-03   -37.47   <2e-16 ***
## WindSpeed9am -1.617e-01  3.770e-03   -42.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.65 on 45133 degrees of freedom
## Multiple R-squared:  0.07059,    Adjusted R-squared:  0.07055
## F-statistic:  1714 on 2 and 45133 DF,  p-value: < 2.2e-16
```

Residual Plots for Complex Linear Model

This shows us our residuals for the linear model. The data here is mostly the same as in our first model, but we can see that the normal q-q graph follows the line a bit more closely than it did the first time, again showing that a normal distribution is the best way to go.

```
par(mfrow=c(2,2)) # plot in a 2x2 grid
plot(lm2) # create plots
```



An Even More Complex Model

g. Build a third linear regression model using a different combination of predictors, interaction effects, polynomial regression, or any combination to try to improve the results. Output the summary and residual plots.

Here, our R-squared increases showing that we can better predict pressure using the rainfall and windspeed. Using even more values, we can better see a much better R-squared using these predictors.

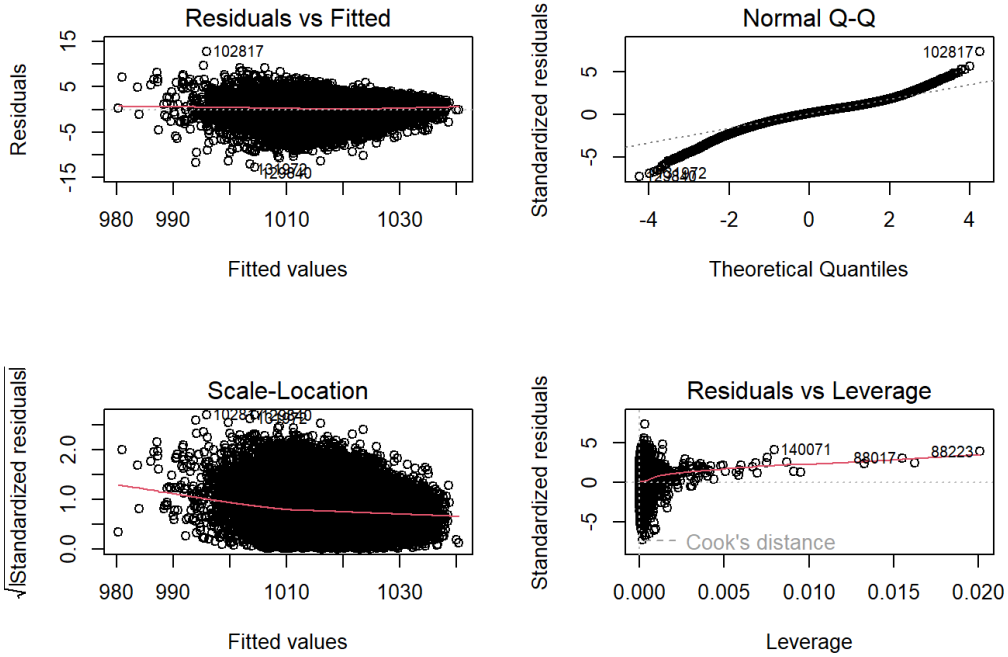
```
lm3 <- lm(Pressure9am~Rainfall + WindSpeed9am + Pressure3pm + Humidity9am + Humidity3pm, data=train) # create linear model
summary(lm3) # summary
```

```
##
## Call:
## lm(formula = Pressure9am ~ Rainfall + WindSpeed9am + Pressure3pm +
##     Humidity9am + Humidity3pm, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.7755  -0.9184   0.1868   1.0898  12.7974
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  49.1045612  1.2697349  38.673   <2e-16 ***
## Rainfall     -0.0226439  0.0012393 -18.272   <2e-16 ***
## WindSpeed9am -0.0472268  0.0010421 -45.319   <2e-16 ***
## Pressure3pm   0.9561898  0.0012546 762.136   <2e-16 ***
## Humidity9am  -0.0001524  0.0006552  -0.233    0.816
## Humidity3pm  -0.0282864  0.0005739 -49.292   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.75 on 45130 degrees of freedom
## Multiple R-squared:  0.9356, Adjusted R-squared:  0.9356
## F-statistic: 1.312e+05 on 5 and 45130 DF,  p-value: < 2.2e-16
```

Residual Plots for the Even More Complex Linear Model

Here is the first time we see something quite different than our simple model. Here, we see that the normal q-q graph deviates from it's line more so than we previously graphed. Residuals vs Fitted begins to show less of a density on one side, and being more random around the middle (zero) line. This indicates that this model works pretty good in comparison to the others, and that a more linear model would work for showing this data.

```
par(mfrow=c(2,2)) # plot in a 2x2 grid
plot(lm3) # create plots
```



Comparing the Linear Models

h. Write a paragraph or more comparing the results. Indicate which model is better and why you think that is the case.

As far as linear models go, Model 3 is the best one. This is because it has the most “random” Residuals vs Fitted graph. The other two have graphs that lump on the right side indicating that there is less linearity. Additionally, it has the best R-squared meaning that it fits a line much better than the other two models. The F-statistic is high and the P-value is low, also indicating that there is a higher significance and relationship in this model rather than the previous two. Alternatively, I would like to see how a normal distribution would fair for models 1 and 2, as they have a strong line on their Normal Q-Q graphs. The third model has the best linear relationship because of its additional predictors. As I will state later, weather needs many predictors to accurately predict how it will act, and model three gives it the most.

Correlation, MSE, and RMSE

i. Using your 3 models, predict and evaluate on the test data using metrics correlation and mse. Compare the results and indicate why you think these results happened

Here, we can directly compare the correlations and the errors of the three models. Model 3 has the a great increase in correlation with a small amount of additional MSE (and RMSE). Again, this is showing that model 3 has the best performance of these three models, but all of these models have high error. This is due to the unpredictable nature of the weather, but I think using more predictors had a positive result overall because there are a multitude of factors that impact the weather. Using more predictors is probably the only way to predict the weather, and using a more encompassing dataset with less NA values and more data points would allow a data scientist (or meteorologist) to better predict weather patterns.

```
pred1 <- predict(lm1, newdata = test) # prediction 1
cor1 <- cor(pred1, test$Humidity9am) # correlation 1
mse1 <- mean((pred1-test$Humidity9am)^2) # mse1
rmse1 <- sqrt(mse1) # rmse1

pred2 <- predict(lm2, newdata = test) # prediction 2
cor2 <- cor(pred2, test$Humidity9am) # correlation 2
mse2 <- mean((pred2-test$Humidity9am)^2) # mse2
rmse2 <- sqrt(mse2) # rmse2

pred3 <- predict(lm3, newdata = test) # prediction 3
cor3 <- cor(pred3, test$Humidity9am) # correlation 3
mse3 <- mean((pred3-test$Humidity9am)^2) # mse3
rmse3 <- sqrt(mse3) # rmse3

# output data collected in a nice format
tmp <- c("Simple Linear Model", "Correlation:", cor1, "", "MSE:", mse1, "", "RMSE:", rmse1,
         "","",
         "Complex Linear Model", "Correlation:", cor2, "", "MSE:", mse2, "", "RMSE:", rmse2,
         "","",
         "Even More Complex Linear Model", "Correlation:", "", cor3, "MSE:", mse3, "", "RMSE:", rmse3)

cat(tmp, sep = '\n')
```

```
## Simple Linear Model
## Correlation:
## -0.266953024713973
##
## MSE:
## 905741.438741224
##
## RMSE:
## 951.704491289824
##
##
## Complex Linear Model
## Correlation:
## 0.00623556768268792
##
## MSE:
## 905705.709578604
##
## RMSE:
## 951.685719961482
##
##
## Even More Complex Linear Model
## Correlation:
##
## 0.130382052339765
## MSE:
## 905708.851846003
##
## RMSE:
## 951.687370855578
```