Regression on Rainfall in Australia

Youssef Donia

3/27/2022

Dataset was found through this link: https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package?datasetId=6012&language=R

This is a dataset on movies the weather in Australia. I will be trying to predict how much rainfall occurs in Australia as a result of other variables in the dataset. This is an interesting project, as it would give me more insight to how weather forecasts work.

Reading from the Data Set

```
df <- read.csv("weatherAUS.csv")
nrow(df)
```

```
## [1] 145460
```

```
head(df)
```

```
##         Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir
## 1 2008-12-01   Albury    13.4    22.9      0.6          NA       NA           W
## 2 2008-12-02   Albury     7.4    25.1      0.0          NA       NA         WNW
## 3 2008-12-03   Albury    12.9    25.7      0.0          NA       NA         WSW
## 4 2008-12-04   Albury     9.2    28.0      0.0          NA       NA          NE
## 5 2008-12-05   Albury    17.5    32.3      1.0          NA       NA           W
## 6 2008-12-06   Albury    14.6    29.7      0.2          NA       NA         WNW
##   WindGustSpeed WindDir9am WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am
## 1            44          W        WNW           20           24          71
## 2            44        NNW        WSW            4           22          44
## 3            46          W        WSW           19           26          38
## 4            24         SE          E           11            9          45
## 5            41        ENE         NW            7           20          82
## 6            56          W          W           19           24          55
##   Humidity3pm Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm
## 1          22      1007.7      1007.1        8       NA    16.9    21.8
## 2          25      1010.6      1007.8       NA       NA    17.2    24.3
## 3          30      1007.6      1008.7       NA        2    21.0    23.2
## 4          16      1017.6      1012.8       NA       NA    18.1    26.5
## 5          33      1010.8      1006.0        7        8    17.8    29.7
## 6          23      1009.2      1005.4       NA       NA    20.6    28.9
##   RainToday RainTomorrow
## 1        No           No
```

```
## 2        No           No
## 3        No           No
## 4        No           No
## 5        No           No
## 6        No           No
```

```
str(df)
```

```
## 'data.frame':    145460 obs. of  23 variables:
##  $ Date         : chr  "2008-12-01" "2008-12-02" "2008-12-03" "2008-12-04" ...
##  $ Location     : chr  "Albury" "Albury" "Albury" "Albury" ...
##  $ MinTemp      : num  13.4 7.4 12.9 9.2 17.5 14.6 14.3 7.7 9.7 13.1 ...
##  $ MaxTemp      : num  22.9 25.1 25.7 28 32.3 29.7 25 26.7 31.9 30.1 ...
##  $ Rainfall     : num  0.6 0 0 0 1 0.2 0 0 0 1.4 ...
##  $ Evaporation  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Sunshine     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ WindGustDir  : chr  "W" "WNW" "WSW" "NE" ...
##  $ WindGustSpeed: int  44 44 46 24 41 56 50 35 80 28 ...
##  $ WindDir9am   : chr  "W" "NNW" "W" "SE" ...
##  $ WindDir3pm   : chr  "WNW" "WSW" "WSW" "E" ...
##  $ WindSpeed9am : int  20 4 19 11 7 19 20 6 7 15 ...
##  $ WindSpeed3pm : int  24 22 26 9 20 24 24 17 28 11 ...
##  $ Humidity9am  : int  71 44 38 45 82 55 49 48 42 58 ...
##  $ Humidity3pm  : int  22 25 30 16 33 23 19 19 9 27 ...
##  $ Pressure9am  : num  1008 1011 1008 1018 1011 ...
##  $ Pressure3pm  : num  1007 1008 1009 1013 1006 ...
##  $ Cloud9am     : int  8 NA NA NA 7 NA 1 NA NA NA ...
##  $ Cloud3pm     : int  NA NA 2 NA 8 NA NA NA NA NA ...
##  $ Temp9am      : num  16.9 17.2 21 18.1 17.8 20.6 18.1 16.3 18.3 20.1 ...
##  $ Temp3pm      : num  21.8 24.3 23.2 26.5 29.7 28.9 24.6 25.5 30.2 28.2 ...
##  $ RainToday    : chr  "No" "No" "No" "No" ...
##  $ RainTomorrow : chr  "No" "No" "No" "No" ...
```

Cleaning the Data

Let's first remove columns that will likely not be good predictors, convert columns that need to be converted to factors, and remove any NA values from the dataset.

I originally created columns that would represent a change in the wind speed, pressure, humidity and temperature, but found little to no correlation between them, so I decided to comment them out. You can find the code below, but it is commented out.

```
# delete useless columns and change some columns to factors
df$Location <- NULL
df$Date <- NULL

# remove NAs: this will be different for every column
df <- na.omit(df) # this removes a lot of rows. maybe in future consider replacing NAs with mean or 0
nrow(df)
```

```
## [1] 56420
```

```
# create columns for changes in wind speed, pressure, humidity and temp
#df$WindChange[1:nrow(df)] <- df$WindSpeed3pm - df$WindSpeed9am
#df$PressureChange[1:nrow(df)] <- df$Pressure3pm - df$Pressure9am
#df$HumidityChange[1:nrow(df)] <- df$Humidity3pm - df$Humidity9am
#df$TempChange[1:nrow(df)] <- df$MaxTemp - df$MinTemp


# convert RainToday and RainTomorrow columns to numeric
df$RainToday[df$RainToday == "Yes"] <- TRUE
df$RainToday[df$RainToday == "No"] <- FALSE

df$RainTomorrow[df$RainTomorrow == "Yes"] <- TRUE
df$RainTomorrow[df$RainTomorrow == "No"] <- FALSE

# convert to factors
for (i in 1:ncol(df)){
    if(is.character(df[1,i])){
      df[,i] <- factor(df[,i])
    } # convert to numeric
    if(!is.numeric(df[1,i])) {
      df[,i] <- as.integer(df[,i])
    }
}

names(df)
```

```
##  [1] "MinTemp"       "MaxTemp"       "Rainfall"      "Evaporation"
##  [5] "Sunshine"      "WindGustDir"   "WindGustSpeed" "WindDir9am"
##  [9] "WindDir3pm"    "WindSpeed9am"  "WindSpeed3pm"  "Humidity9am"
## [13] "Humidity3pm"   "Pressure9am"   "Pressure3pm"   "Cloud9am"
## [17] "Cloud3pm"      "Temp9am"       "Temp3pm"       "RainToday"
## [21] "RainTomorrow"
```

```
str(df)
```

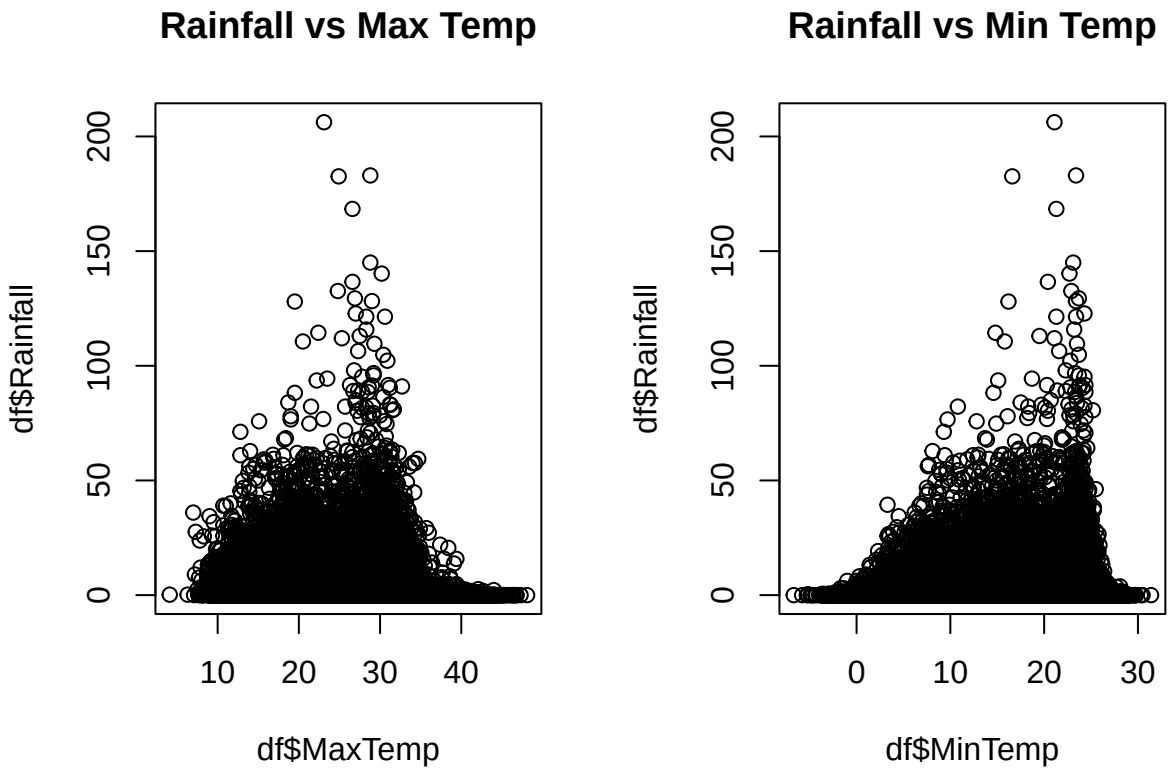
```
## 'data.frame':    56420 obs. of  21 variables:
##  $ MinTemp      : num  17.9 18.4 19.4 21.9 24.2 27.1 23.3 16.1 19 19.7 ...
##  $ MaxTemp      : num  35.2 28.9 37.6 38.4 41 36.1 34 34.2 35.5 35.5 ...
##  $ Rainfall     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Evaporation  : num  12 14.8 10.8 11.4 11.2 13 9.8 14.6 12 11 ...
##  $ Sunshine     : num  12.3 13 10.6 12.2 8.4 0 12.6 13.2 12.3 12.7 ...
##  $ WindGustDir  : int  12 9 6 15 15 4 12 10 2 5 ...
##  $ WindGustSpeed: int  48 37 46 31 35 43 41 37 48 41 ...
##  $ WindDir9am   : int  2 11 6 15 8 4 9 10 2 6 ...
##  $ WindDir3pm   : int  13 11 7 16 15 15 11 9 16 16 ...
##  $ WindSpeed9am : int  6 19 30 6 17 7 17 15 30 15 ...
##  $ WindSpeed3pm : int  20 19 15 6 13 20 19 6 9 17 ...
##  $ Humidity9am  : int  20 30 42 37 19 26 33 25 46 61 ...
##  $ Humidity3pm  : int  13 8 22 22 15 19 15 9 28 14 ...
##  $ Pressure9am  : num  1006 1013 1012 1013 1011 ...
##  $ Pressure3pm  : num  1004 1012 1009 1009 1007 ...
##  $ Cloud9am     : int  2 1 1 1 1 8 3 1 1 1 ...
##  $ Cloud3pm     : int  5 1 6 5 6 8 1 1 5 5 ...
```

```
##  $ Temp9am      : num  26.6 20.3 28.7 29.1 33.6 30.7 25 20.7 23.4 24 ...
##  $ Temp3pm      : num  33.4 27 34.9 35.6 37.6 34.3 31.5 32.8 33.3 33.6 ...
##  $ RainToday    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ RainTomorrow : int  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:89040] 1 2 3 4 5 6 7 8 9 10 ...
##   ..- attr(*, "names")= chr [1:89040] "1" "2" "3" "4" ...
```

Data Visualization

Let's plot the relationship between the Rainfall variable (which will be our response) with the other possible predictor variables

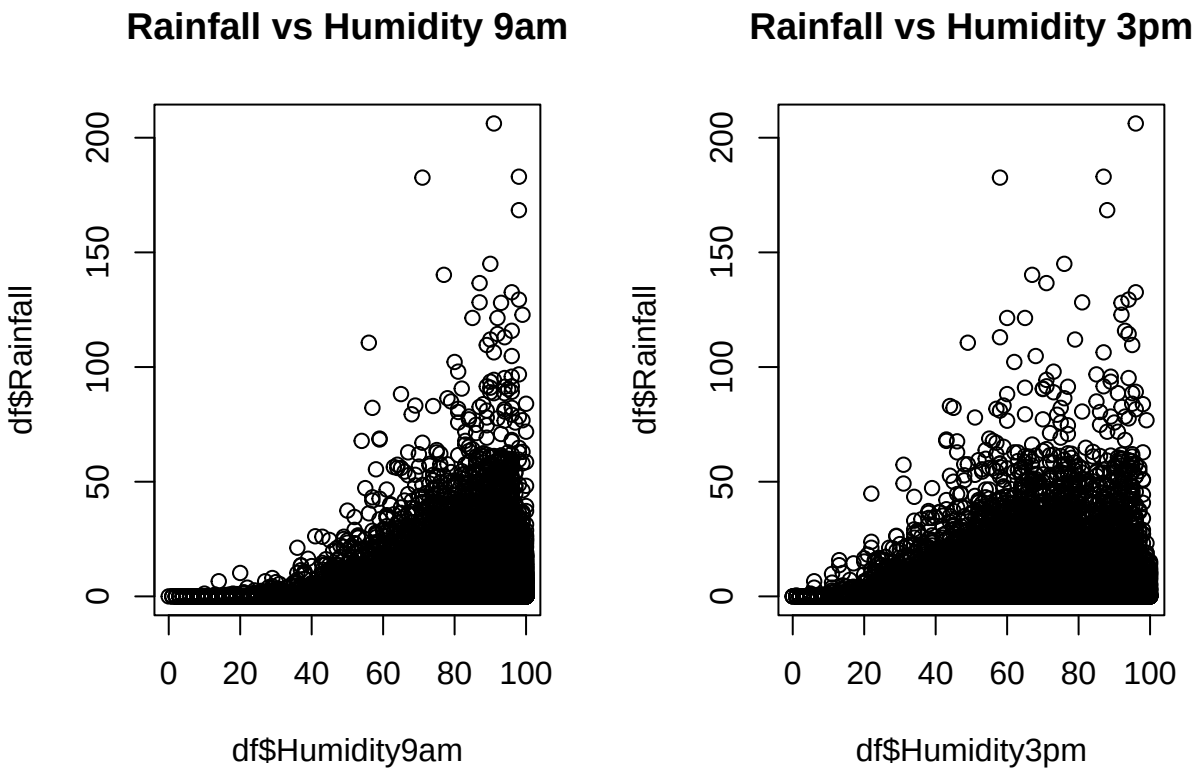
```
par(mfrow=c(1,2))
plot(df$Rainfall~df$MaxTemp, main="Rainfall vs Max Temp")
plot(df$Rainfall~df$MinTemp, main="Rainfall vs Min Temp")
```



There doesn't seem to be that much of a positive linear relationship between the temperatures and the amount of rainfall, but the graph still informs us that it seems like the most amount of rain seems to occur between 15 and 35 degrees. We also need to keep in mind that while one factor alone may not be a good predictor for rainfall, a combination of two or more factors could be more accurate. For example, a certain temp alone may not make it likely for rainfall, but a certain temp and a high humidity and an increased number of clouds in the sky could make it very likely that alot of rainfall occurs.

Let's look at the relationship between Rainfall and Humidity

```
par(mfrow=c(1,2))
plot(df$Rainfall~df$Humidity9am, main="Rainfall vs Humidity 9am")
plot(df$Rainfall~df$Humidity3pm, main="Rainfall vs Humidity 3pm")
```



As we can see above, Humidity seems to have a positive linear relationship with the amount of Rainfall that occurs. We will learn more about these variables when we try to create a model based off the data.

Models

Let's create some models using three different algorithms to predict our response variable, which is Rainfall. I will be using the Linear Regression, KNN, and Decision Tree algorithms to perform regression on this data.

Train and Test

Let's divide our data frame into our train and test data.

```
set.seed(1234)

i <- sample(1:nrow(df), nrow(df)*0.75, replace=FALSE)
train <- df[i,]
test <- df[-i,]
nrow(train) # size of train data
```

```
## [1] 42315
```

```
nrow(test) # size of test data
```

```
## [1] 14105
```

Linear Regression

The first algorithm I will use is Linear Regression. Since I did not see a particularly good correlation between any individual variabls and the amount of rainfall in Australia, I'm going to hope that mixing them together gives me better results as I explained above. Let's make a model using all of the variables (except for the response) as predictors to see the type of results we can get.

```
lm1 <- lm(Rainfall~., data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = Rainfall ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.572  -1.190  -0.081   0.797 194.369
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   26.632175   5.800447   4.591 4.42e-06 ***
## MinTemp       -0.101538   0.014647  -6.932 4.20e-12 ***
## MaxTemp       -0.187834   0.028051  -6.696 2.17e-11 ***
## Evaporation    0.110580   0.011320   9.769  < 2e-16 ***
## Sunshine      -0.171373   0.013930 -12.302  < 2e-16 ***
## WindGustDir   -0.012526   0.008099  -1.547 0.121951
## WindGustSpeed  0.026664   0.003680   7.245 4.41e-13 ***
## WindDir9am     0.026879   0.007153   3.758 0.000172 ***
## WindDir3pm     0.010524   0.007964   1.321 0.186365
## WindSpeed9am   0.016450   0.004675   3.518 0.000435 ***
## WindSpeed3pm  -0.037163   0.004904  -7.577 3.60e-14 ***
## Humidity9am    0.038038   0.003235  11.760  < 2e-16 ***
## Humidity3pm    0.026173   0.003748   6.983 2.94e-12 ***
## Pressure9am   -0.055461   0.019846  -2.795 0.005200 **
## Pressure3pm    0.014549   0.019639   0.741 0.458794
## Cloud9am      -0.007378   0.015117  -0.488 0.625512
## Cloud3pm      -0.114341   0.016117  -7.094 1.32e-12 ***
## Temp9am        0.070722   0.021326   3.316 0.000913 ***
## Temp3pm        0.321666   0.031222  10.302  < 2e-16 ***
## RainToday      8.288779   0.080860 102.508  < 2e-16 ***
## RainTomorrow   0.807733   0.085551   9.442  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.822 on 42294 degrees of freedom
## Multiple R-squared:  0.328,  Adjusted R-squared:  0.3277
## F-statistic:  1032 on 20 and 42294 DF,  p-value: < 2.2e-16
```

```r
mse <- mean(lm1$residuals^2)
print(mse)
```

```
## [1] 33.88048
```

```r
rmse <- sqrt(mse)
print(rmse)
```

```
## [1] 5.820694
```

As we can see, the model is not that great. We only got an R squared statistic of 0.3311, which is very low. Ideally, we would want an R squared statistic of at least 0.8. This could be due to too many predictors, so let's try to remove some that the model claims are not useful.

```r
lm2 <- lm(Rainfall~. - Pressure3pm-Pressure9am-WindDir3pm-WindDir9am-WindGustDir, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = Rainfall ~ . - Pressure3pm - Pressure9am - WindDir3pm -
##     WindDir9am - WindGustDir, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.828  -1.147  -0.074   0.767 194.199
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.553e+01  3.436e-01 -45.203  < 2e-16 ***
## MinTemp       -8.656e-02  1.440e-02  -6.010 1.87e-09 ***
## MaxTemp       -1.686e-01  2.784e-02  -6.055 1.42e-09 ***
## Evaporation    1.152e-01  1.123e-02  10.257  < 2e-16 ***
## Sunshine      -1.640e-01  1.385e-02 -11.845  < 2e-16 ***
## WindGustSpeed  3.398e-02  3.534e-03   9.615  < 2e-16 ***
## WindSpeed9am   1.275e-02  4.625e-03   2.757  0.00583 **
## WindSpeed3pm  -3.604e-02  4.888e-03  -7.374 1.69e-13 ***
## Humidity9am    3.923e-02  3.223e-03  12.173  < 2e-16 ***
## Humidity3pm    2.389e-02  3.706e-03   6.448 1.15e-10 ***
## Cloud9am       1.194e-04  1.509e-02   0.008  0.99369
## Cloud3pm      -1.071e-01  1.608e-02  -6.658 2.80e-11 ***
## Temp9am        8.568e-02  2.117e-02   4.047 5.20e-05 ***
## Temp3pm        2.856e-01  3.007e-02   9.496  < 2e-16 ***
## RainToday      8.408e+00  7.906e-02 106.344  < 2e-16 ***
## RainTomorrow   8.662e-01  8.432e-02  10.273  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.827 on 42299 degrees of freedom
## Multiple R-squared:  0.3267, Adjusted R-squared:  0.3265
## F-statistic:  1369 on 15 and 42299 DF,  p-value: < 2.2e-16
```

```
mse2 <- mean(lm2$residuals^2)
print(mse2)
```

```
## [1] 33.94654
```

```
rmse2 <- sqrt(mse2)
print(rmse2)
```

```
## [1] 5.826366
```

This model did not seem much better than the first. The R squared is a little lower than for the first model and the standard error was higher. The F statistic was much higher however. To figure out which model is slightly better, we can use the anova function.

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

```
## Analysis of Variance Table
##
## Model 1: Rainfall ~ MinTemp + MaxTemp + Evaporation + Sunshine + WindGustDir +
##     WindGustSpeed + WindDir9am + WindDir3pm + WindSpeed9am +
##     WindSpeed3pm + Humidity9am + Humidity3pm + Pressure9am +
##     Pressure3pm + Cloud9am + Cloud3pm + Temp9am + Temp3pm + RainToday +
##     RainTomorrow
## Model 2: Rainfall ~ (MinTemp + MaxTemp + Evaporation + Sunshine + WindGustDir +
##     WindGustSpeed + WindDir9am + WindDir3pm + WindSpeed9am +
##     WindSpeed3pm + Humidity9am + Humidity3pm + Pressure9am +
##     Pressure3pm + Cloud9am + Cloud3pm + Temp9am + Temp3pm + RainToday +
##     RainTomorrow) - Pressure3pm - Pressure9am - WindDir3pm -
##     WindDir9am - WindGustDir
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1  42294 1433652
## 2  42299 1436448 -5   -2795.5 16.494 2.65e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The second model is shown to have a small p value, which implies that the model is more significant. There is not much difference between the RSS, so we will go with model 2. This may or may not be a good choice, but the difference between these two models should not be significant and we should get similar results either way.

Now, let's try to accurately predict the amount of rainfall using the test data. I do not have high hopes for this, as the model did not seem promising, but let's see if there is an improvement here.

```
lm.pred <- predict(lm2, newdata = test)
lm.cor <- cor(lm.pred, test$Rainfall)
print(lm.cor)
```

```
## [1] 0.5829296
```

```
lm.mse <- mean((lm.pred - test$Rainfall)^2)
print(lm.mse)
```

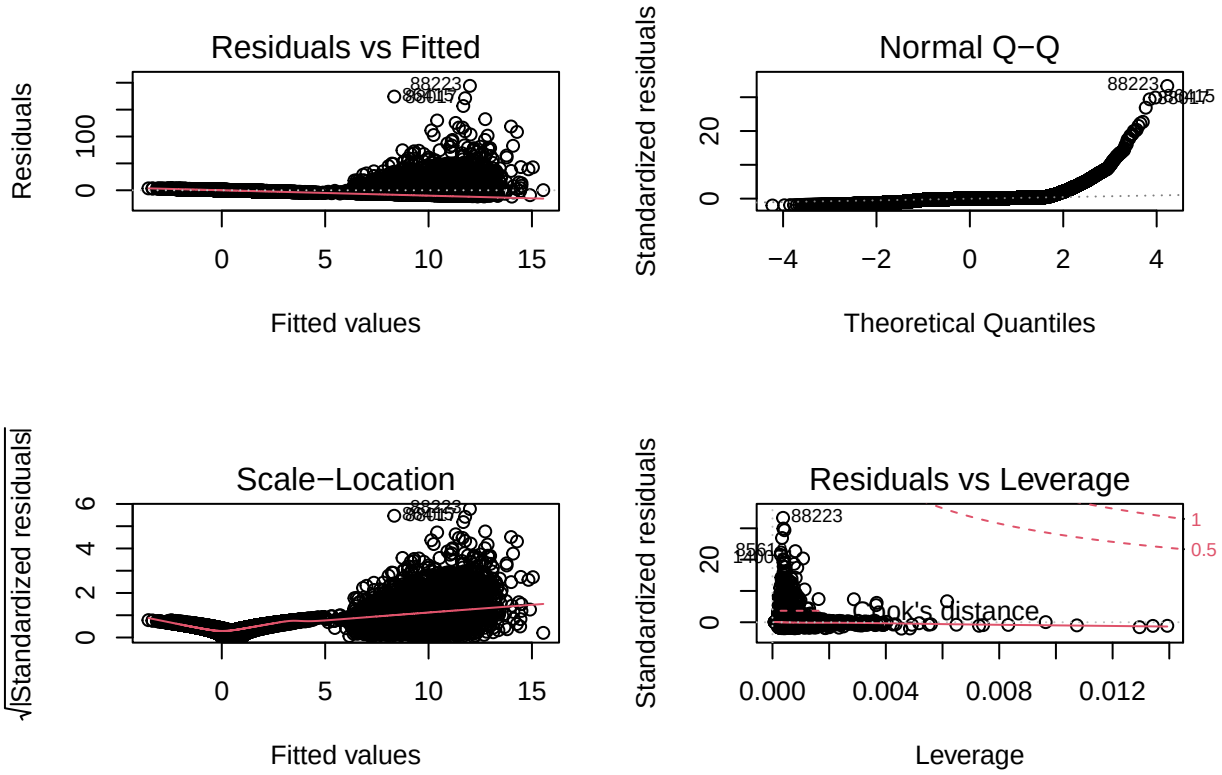
```
## [1] 30.08396
```

The predictions are more accurate than I thought. The correlation is almost 0.6, and the mean squared error is less than it was for the model. The moderately high correlation and the decrease in mse suggest that the model was able to generalize decently well to the test data.

Plot Residuals

I will plot residuals to see how well the model fit with the data

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



These residuals are not very good, indicating that the model did not fit as well to the data as it could. Overall, the model was moderately accurate, but on the weak side. A higher accuracy would be preferable and necessary to accurately predict the amount of rainfall.

KNN

Let's try to use the KNN algorithm and see if we can make a better model. The bar isn't very high, so hopefully we can get a better result.

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
predictors <- c("MinTemp", "MaxTemp", "Evaporation", "Sunshine", "WindGustSpeed", "WindSpeed9am", "Wind

knn1 <- knnreg(train[,predictors], train$Rainfall, k=3)
knn.pred <- predict(knn1, test[,predictors])
cor.knn1 <- cor(knn.pred, test$Rainfall)
mse.knn1 <- mean((knn.pred - test$Rainfall)^2)
print(paste("cor: ", cor.knn1))
```

```
## [1] "cor:  0.38567228031531"
```

```
print(paste("mse: ", mse.knn1))
```

```
## [1] "mse:  44.1038599984245"
```

The performance of the KNN algorithm was much worse than the Linear Regression. The correlation was lower and the mse was higher. This could be because the KNN needed weight for each characteristic. It also executed much more slowly, making the Linear Regression algorithm faster and more accurate. The data also could be scaled to possibly improve the performance of the model, which is something I did not do in this implementation of the algorithm.

Decision Tree

Both of the previous algorithms did not perform very well, especially the KNN algorithm. Let's try creating a more accurate model using the Decision Tree algorithm and hope we get better results.

```
library(rpart)
tree1 <- rpart(Rainfall~.- Pressure3pm - Pressure9am - WindDir3pm -
    WindDir9am - WindGustDir, method="anova", data=train)
summary(tree1)
```

```
## Call:
## rpart(formula = Rainfall ~ . - Pressure3pm - Pressure9am - WindDir3pm -
##     WindDir9am - WindGustDir, data = train, method = "anova")
##   n= 42315
##
##           CP nsplit rel error    xerror       xstd
## 1 0.29966679      0 1.0000000 1.0000596 0.05001334
## 2 0.02901306      1 0.7003332 0.7005095 0.04339851
## 3 0.01514590      3 0.6423071 0.6505161 0.03960252
## 4 0.01157453      4 0.6271612 0.6438011 0.03894513
## 5 0.01000000      5 0.6155867 0.6459726 0.03916799
##
## Variable importance
```

```
##   RainToday Humidity9am     Temp9am     MinTemp     Temp3pm     MaxTemp
##          53          11           7           7           5           5
## Humidity3pm    Sunshine Evaporation    Cloud9am    Cloud3pm
##           4           2           2           2           1
##
## Node number 1: 42315 observations,    complexity param=0.2996668
##   mean=2.153194, MSE=50.421
##   left son=2 (32899 obs) right son=3 (9416 obs)
##   Primary splits:
##       RainToday    < 1.5   to the left,  improve=0.29966680, (0 missing)
##       Humidity9am  < 80.5  to the left,  improve=0.07578129, (0 missing)
##       Humidity3pm  < 66.5  to the left,  improve=0.06589245, (0 missing)
##       RainTomorrow < 1.5   to the left,  improve=0.06470570, (0 missing)
##       Cloud9am     < 7.5   to the left,  improve=0.04549992, (0 missing)
##   Surrogate splits:
##       Humidity9am < 89.5  to the left,  agree=0.794, adj=0.073, (0 split)
##       Humidity3pm < 79.5  to the left,  agree=0.790, adj=0.057, (0 split)
##       Sunshine    < 0.05  to the right, agree=0.781, adj=0.016, (0 split)
##       Temp3pm     < 10.25 to the right, agree=0.779, adj=0.007, (0 split)
##       Evaporation < 0.55  to the right, agree=0.779, adj=0.006, (0 split)
##
## Node number 2: 32899 observations
##   mean=0.07365573, MSE=0.03972453
##
## Node number 3: 9416 observations,    complexity param=0.02901306
##   mean=9.418989, MSE=158.5492
##   left son=6 (5247 obs) right son=7 (4169 obs)
##   Primary splits:
##       MinTemp      < 14.55 to the left,  improve=0.04025314, (0 missing)
##       Sunshine     < 1.65  to the right, improve=0.03214740, (0 missing)
##       Temp9am      < 16.35 to the left,  improve=0.03184233, (0 missing)
##       RainTomorrow < 1.5   to the left,  improve=0.03075139, (0 missing)
##       Cloud9am     < 7.5   to the left,  improve=0.02820295, (0 missing)
##   Surrogate splits:
##       Temp9am     < 16.85 to the left,  agree=0.935, adj=0.853, (0 split)
##       MaxTemp     < 21.75 to the left,  agree=0.882, adj=0.734, (0 split)
##       Temp3pm     < 19.95 to the left,  agree=0.876, adj=0.719, (0 split)
##       Evaporation < 4.15  to the left,  agree=0.703, adj=0.330, (0 split)
##       Sunshine    < 0.45  to the right, agree=0.578, adj=0.046, (0 split)
##
## Node number 6: 5247 observations
##   mean=7.167124, MSE=64.25064
##
## Node number 7: 4169 observations,    complexity param=0.02901306
##   mean=12.25313, MSE=262.8165
##   left son=14 (2869 obs) right son=15 (1300 obs)
##   Primary splits:
##       Humidity9am < 86.5  to the left,  improve=0.05814513, (0 missing)
##       Cloud9am    < 7.5   to the left,  improve=0.04897760, (0 missing)
##       Sunshine    < 1.65  to the right, improve=0.04084239, (0 missing)
##       Humidity3pm < 74.5  to the left,  improve=0.03286529, (0 missing)
##       Cloud3pm    < 7.5   to the left,  improve=0.02936295, (0 missing)
##   Surrogate splits:
##       Cloud9am    < 7.5   to the left,  agree=0.796, adj=0.347, (0 split)
```

```
##       Sunshine    < 0.45  to the right, agree=0.757, adj=0.222, (0 split)
##       Humidity3pm < 80.5  to the left,  agree=0.754, adj=0.210, (0 split)
##       Cloud3pm    < 7.5   to the left,  agree=0.731, adj=0.136, (0 split)
##       Temp9am     < 17.35 to the right, agree=0.711, adj=0.075, (0 split)
##
## Node number 14: 2869 observations
##   mean=9.621715, MSE=155.798
##
## Node number 15: 1300 observations,    complexity param=0.0151459
##   mean=18.06046, MSE=449.9915
##   left son=30 (975 obs) right son=31 (325 obs)
##   Primary splits:
##       Temp9am     < 23.25 to the left,  improve=0.05523995, (0 missing)
##       MinTemp     < 22.65 to the left,  improve=0.05498642, (0 missing)
##       MaxTemp     < 26.45 to the left,  improve=0.02809090, (0 missing)
##       Evaporation < 6.1   to the left,  improve=0.02657963, (0 missing)
##       Temp3pm     < 25.25 to the left,  improve=0.02474295, (0 missing)
##   Surrogate splits:
##       MinTemp       < 22.05 to the left,  agree=0.962, adj=0.849, (0 split)
##       MaxTemp       < 27.65 to the left,  agree=0.885, adj=0.542, (0 split)
##       Temp3pm       < 26.75 to the left,  agree=0.867, adj=0.468, (0 split)
##       WindGustSpeed < 96    to the left,  agree=0.751, adj=0.003, (0 split)
##
## Node number 30: 975 observations
##   mean=15.18195, MSE=304.4514
##
## Node number 31: 325 observations,    complexity param=0.01157453
##   mean=26.696, MSE=787.1819
##   left son=62 (312 obs) right son=63 (13 obs)
##   Primary splits:
##       Humidity9am  < 97.5  to the left,  improve=0.09652743, (0 missing)
##       Sunshine     < 1.95  to the right, improve=0.09635148, (0 missing)
##       Cloud3pm     < 7.5   to the left,  improve=0.09471097, (0 missing)
##       Cloud9am     < 7.5   to the left,  improve=0.09156106, (0 missing)
##       WindSpeed9am < 35    to the left,  improve=0.07124524, (0 missing)
##
## Node number 62: 312 observations
##   mean=24.91667, MSE=613.249
##
## Node number 63: 13 observations
##   mean=69.4, MSE=3061.957
```

Let's output the correlation and the mse to compare the tree with the other models.

```
pred.tree <- predict(tree1, newdata=test)
cor.tree <- cor(pred.tree, test$Rainfall)
print(paste("cor: ", cor.tree))
```

```
## [1] "cor:  0.624195776548486"
```

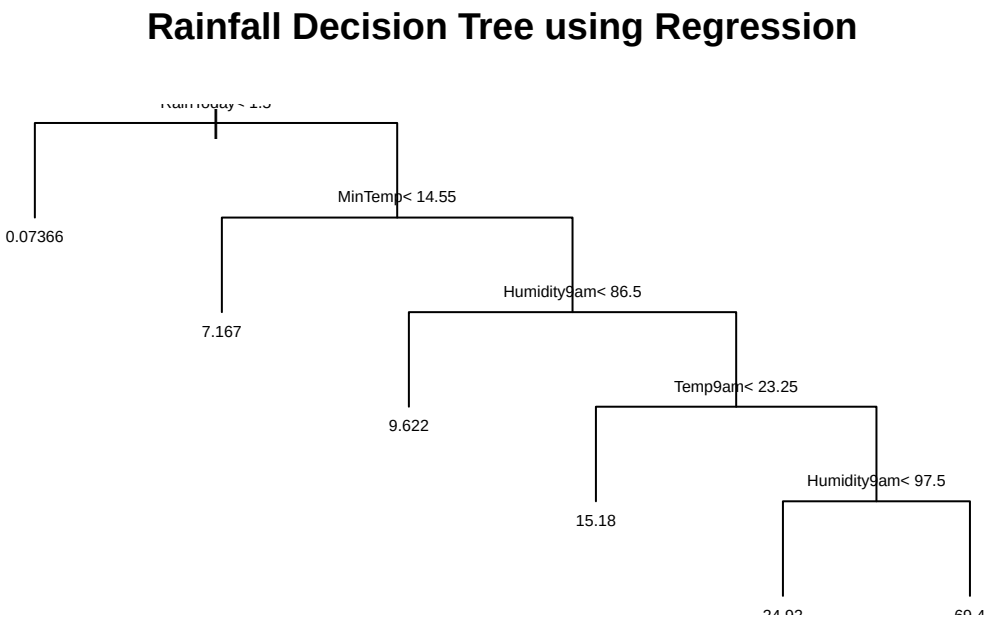
```
mse.tree <- mean((pred.tree - test$Rainfall))
print(paste("mse: ", mse.tree))
```

```
## [1] "mse:  0.0417918101243365"
```

The Decision Tree actually seemed to do better than both the KNN and the Linear Regression models. It has an correlation of about 0.62 and a 0.04 mse, which is much lower than the mse for the other models.

Let's plot this tree to get a better understanding of how it works. Granted, the RainToday seems like it would be a good predictor, but due to maybe some issues with the data, even with it the models do not reach a very high accuracy, so I have justified including it as a predictor.

```
plot(tree1, uniform=TRUE, main="Rainfall Decision Tree using Regression")
text(tree1, cex=0.5, pretty=0)
```



Results Analysis

The Decision Tree performed best out of the three algorithms I used. Based on what I know, it seems that the data was complex, leading to the decision tree outperforming the linear model. The KNN model was the least accurate out of the three. This could be because the characteristics did not have a weight to them. KNN also seems to perform better with purely quantitative data, and a prominent predictor I used was a factor. This could also be because the k value I inputted was not helping the model. Perhaps an optimal k would help the model be more accurate. Something I could also do in the future is scale the data for the KNN and see if that helps it perform better, but I doubt that it would have performed better to a point where it was usable.

What was learned from the data?

I learned that finding correlations between data without a significant amount of data cleaning can be very hard. If there are any inconsistencies in the data, it can be very hard to make an accurate model. NA values can make a big difference in the correlation of my data. To make my models more accurate, I would have had to perform a lot more data cleaning, and visualize the data more to get a better idea of how to model it. If I spent more time on this project, I believe I would eventually be able to make more accurate predictions for the amount of rainfall. My script did, however, learn from the data that temperature and humidity are big factors in predicting the amount of rainfall that could possibly occur. Even though this information is already known, a more detailed script could possibly find relationships between rainfall and other data that was not considered before.