Regression

Zachary Canoot & Gray Simpson

Regression

Using the Hungary Dataset Weather in Szeged 2006-2016 Found on Kaggle.

Our goal is to see if we can see how other weather factors, such as Wind Speed and Humidity, relate to the difference between Apparent Temperature and actual Temperature. Though we identify apparent temperature as a very good predictor of the difference, we do not use this in this assignment as we are interested in exploring more the other factors that influence the disparity.

Linear Regression is one of many supervised models of Machine Learning that functions by finding a trend in given data, using one or more input parameters to find a line of best fit, though it is not always a straight line. As shown by the name, linear regression models assume that the relationship between relevant attributes is linear. The model will predict coefficients for the effect of each predictor. It has low variance due to its linear nature, but with such an assumption, it will also be very high bias.

Data Exploration

First, we read the data in, then divide our data up into training and testing. We have to add a column for the data we are interested in learning about, however, it is simply the difference between two other columns.

```
df <- read.csv("weatherHistory.csv")
#Here we'll add the data that we are interested in: difference in Apparent Temp and Temp.
df$Temperature.Diff <- df$Temperature..C. - df$Apparent.Temperature..C.
#We'll also convert some data to factors for ease.
df$Precip.Type <- as.factor(df$Precip.Type)
df$Summary <- as.factor(df$Summary)
str(df)
```

```
## 'data.frame':    96453 obs. of  13 variables:
##  $ Formatted.Date          : chr  "2006-04-01 00:00:00.000 +0200" "2006-04-01 01:00:00.000 +0200" "20
##  $ Summary                 : Factor w/ 27 levels "Breezy","Breezy and Dry",..: 20 20 18 20 18 20 20 1
##  $ Precip.Type             : Factor w/ 3 levels "null","rain",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Temperature..C.         : num  9.47 9.36 9.38 8.29 8.76 ...
##  $ Apparent.Temperature..C.: num  7.39 7.23 9.38 5.94 6.98 ...
##  $ Humidity                : num  0.89 0.86 0.89 0.83 0.83 0.85 0.95 0.89 0.82 0.72 ...
##  $ Wind.Speed..km.h.       : num  14.12 14.26 3.93 14.1 11.04 ...
##  $ Wind.Bearing..degrees.  : num  251 259 204 269 259 258 259 260 259 279 ...
##  $ Visibility..km.         : num  15.8 15.8 15 15.8 15.8 ...
##  $ Loud.Cover              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pressure..millibars.    : num  1015 1016 1016 1016 1017 ...
##  $ Daily.Summary           : chr  "Partly cloudy throughout the day." "Partly cloudy throughout the d
##  $ Temperature.Diff        : num  2.08 2.13 0 2.34 1.78 ...
```

```
#Now we'll divide into train and test.
set.seed(8)
trainindex <- sample(1:nrow(df),nrow(df)*.8,replace=FALSE)
train <- df[trainindex,]
test <- df[-trainindex,]
```

Next, we want to explore our training data.

```
names(df)
```

```
##  [1] "Formatted.Date"           "Summary"
##  [3] "Precip.Type"              "Temperature..C."
##  [5] "Apparent.Temperature..C." "Humidity"
##  [7] "Wind.Speed..km.h."        "Wind.Bearing..degrees."
##  [9] "Visibility..km."          "Loud.Cover"
## [11] "Pressure..millibars."     "Daily.Summary"
## [13] "Temperature.Diff"
```

```
dim(df)
```

```
## [1] 96453    13
```

```
head(df)
```

```
##                  Formatted.Date       Summary Precip.Type Temperature..C.
## 1 2006-04-01 00:00:00.000 +0200 Partly Cloudy        rain        9.472222
## 2 2006-04-01 01:00:00.000 +0200 Partly Cloudy        rain        9.355556
## 3 2006-04-01 02:00:00.000 +0200 Mostly Cloudy        rain        9.377778
## 4 2006-04-01 03:00:00.000 +0200 Partly Cloudy        rain        8.288889
## 5 2006-04-01 04:00:00.000 +0200 Mostly Cloudy        rain        8.755556
## 6 2006-04-01 05:00:00.000 +0200 Partly Cloudy        rain        9.222222
##   Apparent.Temperature..C. Humidity Wind.Speed..km.h. Wind.Bearing..degrees.
## 1                 7.388889     0.89           14.1197                    251
## 2                 7.227778     0.86           14.2646                    259
## 3                 9.377778     0.89            3.9284                    204
## 4                 5.944444     0.83           14.1036                    269
## 5                 6.977778     0.83           11.0446                    259
## 6                 7.111111     0.85           13.9587                    258
##   Visibility..km. Loud.Cover Pressure..millibars.
## 1         15.8263          0              1015.13
## 2         15.8263          0              1015.63
## 3         14.9569          0              1015.94
## 4         15.8263          0              1016.41
## 5         15.8263          0              1016.51
## 6         14.9569          0              1016.66
##                       Daily.Summary Temperature.Diff
## 1 Partly cloudy throughout the day.         2.083333
## 2 Partly cloudy throughout the day.         2.127778
## 3 Partly cloudy throughout the day.         0.000000
## 4 Partly cloudy throughout the day.         2.344444
## 5 Partly cloudy throughout the day.         1.777778
## 6 Partly cloudy throughout the day.         2.111111
```

```
colMeans(df[4:11])
```

```
##          Temperature..C. Apparent.Temperature..C.                 Humidity
##                11.932678                10.855029                 0.734899
##        Wind.Speed..km.h.   Wind.Bearing..degrees.          Visibility..km.
##                10.810640               187.509232                10.347325
##               Loud.Cover     Pressure..millibars.
##                 0.000000              1003.235956
```

```
#Noticing the mean of 0 of df$Loud.Cover, lets check its sum in specific.
sum(df$Loud.Cover)
```

```
## [1] 0
```

```
#Let's see if we have any NAs more generally, now.
colSums(is.na(df))
```

```
##           Formatted.Date                  Summary              Precip.Type
##                        0                        0                        0
##          Temperature..C. Apparent.Temperature..C.                 Humidity
##                        0                        0                        0
##        Wind.Speed..km.h.   Wind.Bearing..degrees.          Visibility..km.
##                        0                        0                        0
##               Loud.Cover     Pressure..millibars.            Daily.Summary
##                        0                        0                        0
##         Temperature.Diff
##                        0
```

```
#Okay, so we don't have any NAs.

#Now, lets see how R would summarize this data.
summary(df)
```

```
##  Formatted.Date                    Summary      Precip.Type  Temperature..C.
##  Length:96453       Partly Cloudy      :31733   null:  517   Min.   :-21.822
##  Class :character   Mostly Cloudy      :28094   rain:85224   1st Qu.:  4.689
##  Mode  :character   Overcast           :16597   snow:10712   Median : 12.000
##                     Clear              :10890                Mean   : 11.933
##                     Foggy              : 7148                3rd Qu.: 18.839
##                     Breezy and Overcast:  528                Max.   : 39.906
##                     (Other)            : 1463
##  Apparent.Temperature..C.    Humidity      Wind.Speed..km.h.
##  Min.   :-27.717          Min.   :0.0000   Min.   : 0.000
##  1st Qu.:  2.311          1st Qu.:0.6000   1st Qu.: 5.828
##  Median : 12.000          Median :0.7800   Median : 9.966
##  Mean   : 10.855          Mean   :0.7349   Mean   :10.811
##  3rd Qu.: 18.839          3rd Qu.:0.8900   3rd Qu.:14.136
##  Max.   : 39.344          Max.   :1.0000   Max.   :63.853
##
##  Wind.Bearing..degrees. Visibility..km.   Loud.Cover Pressure..millibars.
##  Min.   :  0.0          Min.   : 0.00   Min.   :0    Min.   :   0
```

```
##  1st Qu.:116.0          1st Qu.: 8.34   1st Qu.:0    1st Qu.:1012
##  Median :180.0          Median :10.05   Median :0    Median :1016
##  Mean   :187.5          Mean   :10.35   Mean   :0    Mean   :1003
##  3rd Qu.:290.0          3rd Qu.:14.81   3rd Qu.:0    3rd Qu.:1021
##  Max.   :359.0          Max.   :16.10   Max.   :0    Max.   :1046
##
##  Daily.Summary      Temperature.Diff
##  Length:96453       Min.   :-4.811
##  Class :character   1st Qu.: 0.000
##  Mode  :character   Median : 0.000
##                     Mean   : 1.078
##                     3rd Qu.: 2.217
##                     Max.   :10.183
##
```

```
#We would also like to look at this particular aspect to see how the different values pan out.
summary(df$Summary)
```

```
##                              Breezy                       Breezy and Dry
##                                  54                                    1
##                     Breezy and Foggy             Breezy and Mostly Cloudy
##                                  35                                  516
##                   Breezy and Overcast             Breezy and Partly Cloudy
##                                 528                                  386
##                               Clear Dangerously Windy and Partly Cloudy
##                               10890                                    1
##                             Drizzle                                  Dry
##                                  39                                   34
##                 Dry and Mostly Cloudy                Dry and Partly Cloudy
##                                  14                                   86
##                               Foggy              Humid and Mostly Cloudy
##                                7148                                   40
##                   Humid and Overcast              Humid and Partly Cloudy
##                                   7                                   17
##                            Light Rain                         Mostly Cloudy
##                                  63                                28094
##                              Overcast                         Partly Cloudy
##                                 16597                                31733
##                                 Rain                                 Windy
##                                  10                                    8
##                          Windy and Dry                     Windy and Foggy
##                                    1                                    4
##               Windy and Mostly Cloudy                   Windy and Overcast
##                                  35                                   45
##               Windy and Partly Cloudy
##                                  67
```

One thing we notice is that there is an attribute labeled 'Loud Cover' that all values are 0 in. Therefore, this will be an aspect that we will ignore.

However, in the summary, we can notice that there is a minimum value of 0 on Pressure, which has an average and max similar to each other. We can assume a 0 is a NA value here.

The other values that are 0 we can't make assumptions on validity.

If Wind Speed is 0, so will Wind Bearing, we can realize from looking through the data in passing. Since there is no place in earth without wind, we can also state that these values aren't accurate. After we look a the data some more, we'll decide how we want to clean them up.

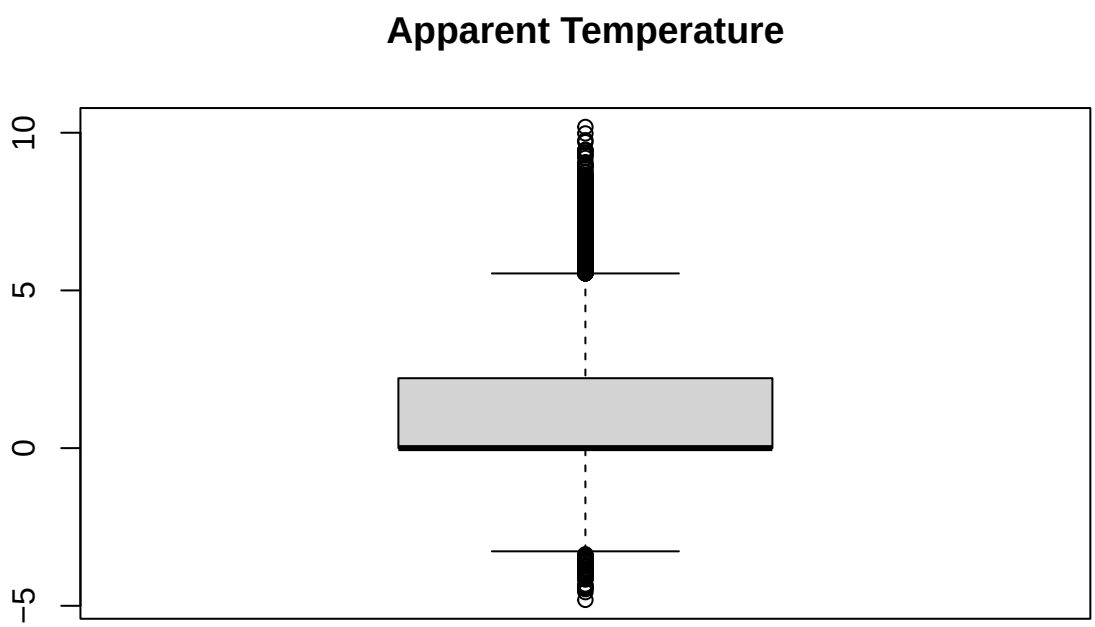
We cannot come to a clear resolution on other attributes.

We'll pull up some graphs to get a better idea of what we have to do, now. Yellow dots are null precipitation days, green is rain, and blue is snow.

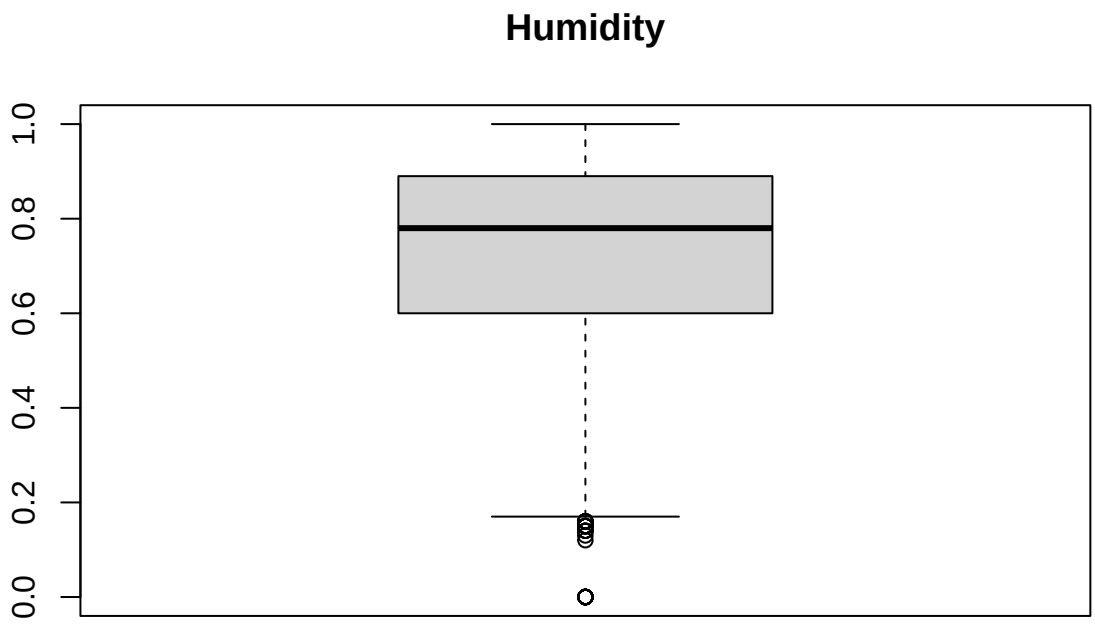
```
cor(df[4:7])
```

```
##                           Temperature..C. Apparent.Temperature..C.   Humidity
## Temperature..C.                1.000000000                0.9926286 -0.6322547
## Apparent.Temperature..C.       0.992628564                1.0000000 -0.6025710
## Humidity                      -0.632254675               -0.6025710  1.0000000
## Wind.Speed..km.h.              0.008956968               -0.0566497 -0.2249515
##                           Wind.Speed..km.h.
## Temperature..C.                0.008956968
## Apparent.Temperature..C.      -0.056649698
## Humidity                      -0.224951456
## Wind.Speed..km.h.              1.000000000
```

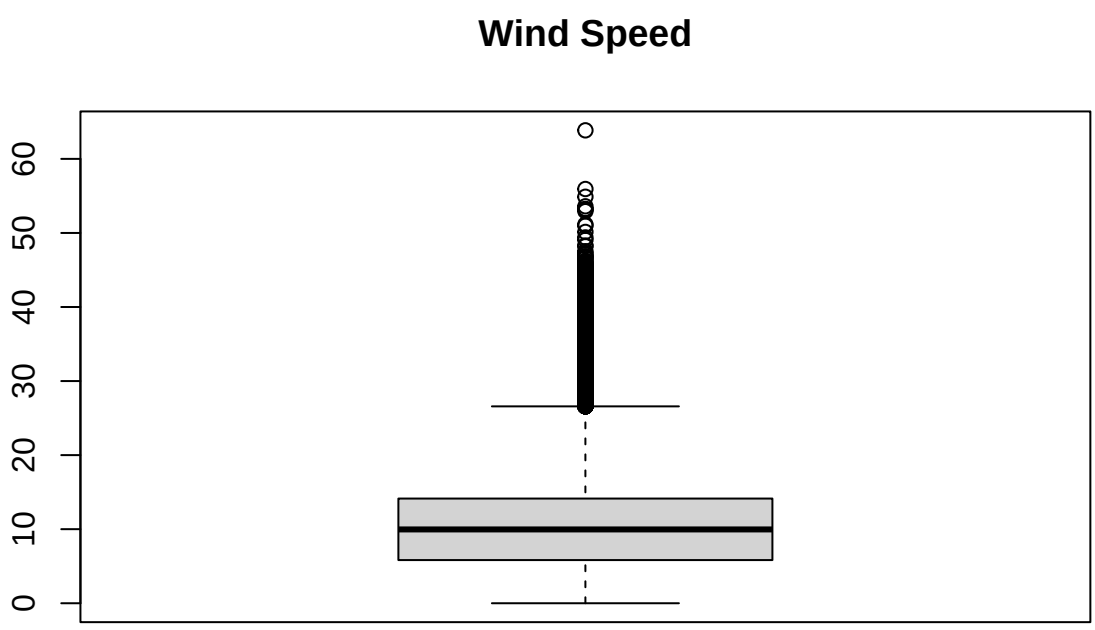
```
boxplot(df$Temperature.Diff,main="Apparent Temperature")
```



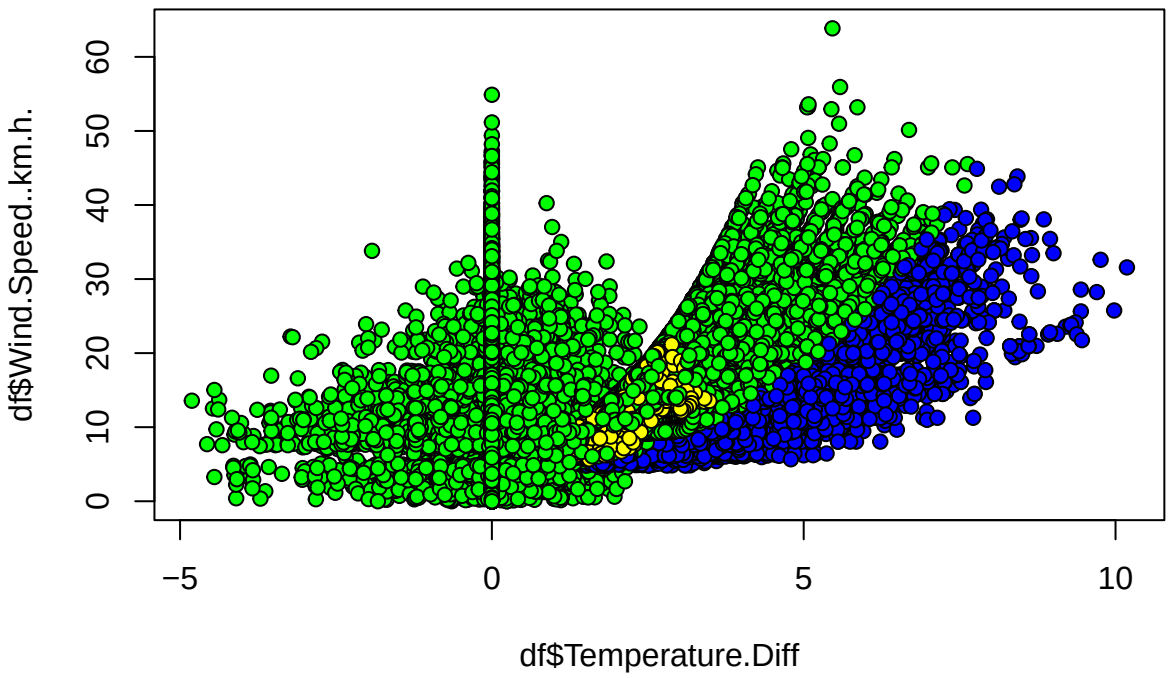
```
boxplot(df$Humidity,main="Humidity")
```



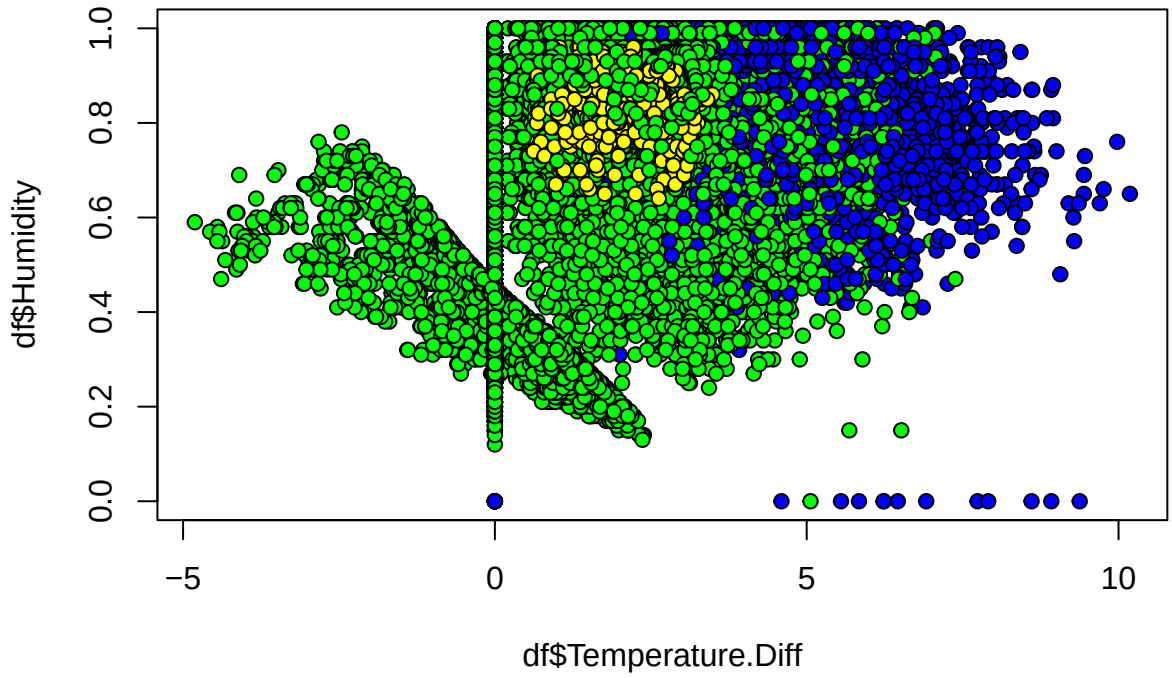

```
boxplot(df$Wind.Speed..km.h.,main="Wind Speed")
```



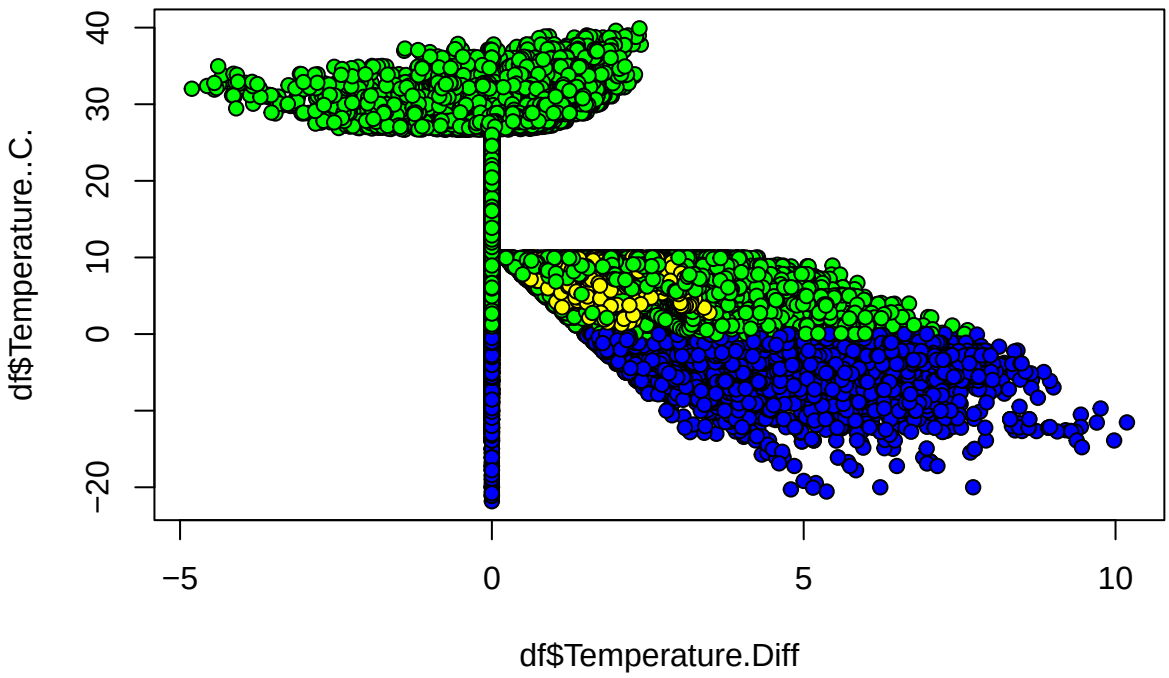

```
#pairs(df[4:7],main="Temperature, Humidity, and Wind Correlations")
plot(df$Temperature.Diff,df$Wind.Speed..km.h.,pch=21,bg=c("yellow","green","blue")[as.integer(df$Precip
```




```
plot(df$Temperature.Diff,df$Humidity,pch=21,bg=c("yellow","green","blue")[as.integer(df$Precip.Type)])
```




```
plot(df$Temperature.Diff, df$Temperature..C.,pch=21,bg=c("yellow","green","blue")[as.integer(df$Precip.T
```



We can notice that there are some outliers in humidity at the 0 line we'll want to sort out, as well.

We also notice a sort of "knife" shape in the data when being compared, at around 10 degrees Celsius. By and large, we have such a large amount of data, it's difficult to notice quick correlations, aside from wind speed and temperature when it is snowing/below freezing.

That's why we have Machine Learning, we suppose, even if it implies that linear regression may not be the best fit for this data set.

While we would like to predict the results without the base temperature, we can see that is is very clearly related and helpful.

Now, we'll clean up the data according to what we found. We'll clean up only what is referenced, but we will delete what we are uncertain about, since we have such a large amount of data.

```
df[,6:7][df[,6:7]==0] <- NA
df[,13:13][df[,13:13]==0] <- NA
df <- na.omit(df)
summary(df)
```

```
##  Formatted.Date                   Summary      Precip.Type  Temperature..C.
##  Length:40660       Partly Cloudy      :11421   null:  237   Min.   :-20.556
##  Class :character   Mostly Cloudy      :10907   rain:32750   1st Qu.:  1.139
##  Mode  :character   Overcast           : 9062   snow: 7673   Median :  5.122
##                     Clear              : 4167                Mean   :  7.924
##                     Foggy              : 3969                3rd Qu.:  8.867
##                     Breezy and Overcast:  375                Max.   : 39.906
##                     (Other)            :  759
```

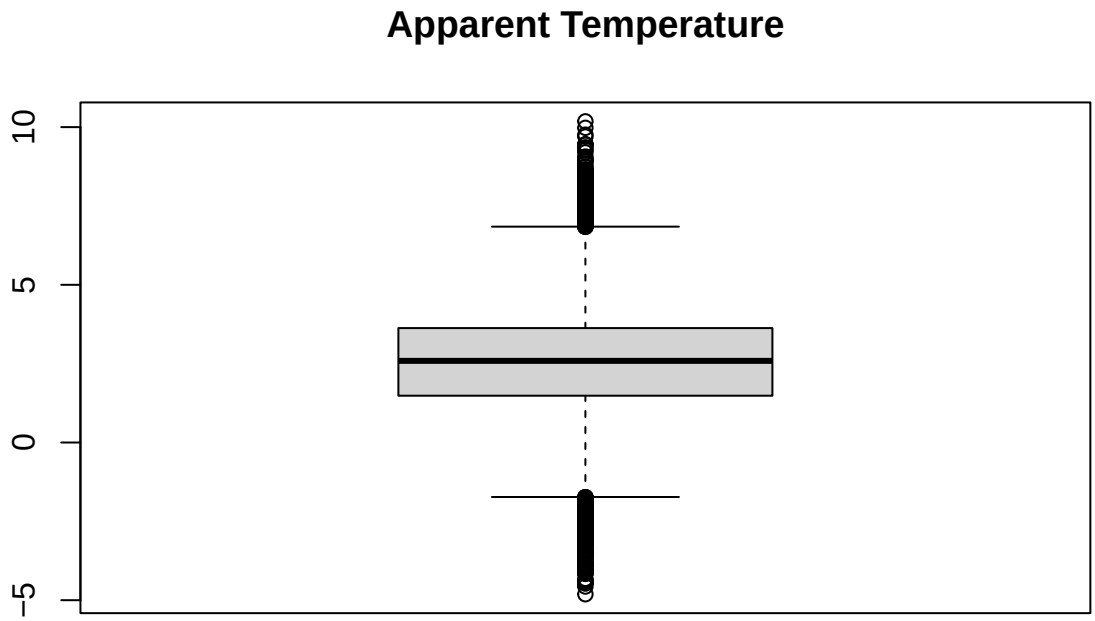
```
##  Apparent.Temperature..C.    Humidity      Wind.Speed..km.h.
##  Min.   :-27.717          Min.   :0.1300   Min.   : 0.1288
##  1st Qu.: -2.267          1st Qu.:0.6500   1st Qu.: 8.1788
##  Median :  2.544          Median :0.8200   Median :11.2217
##  Mean   :  5.370          Mean   :0.7524   Mean   :12.8271
##  3rd Qu.:  6.839          3rd Qu.:0.9100   3rd Qu.:15.4721
##  Max.   : 39.344          Max.   :1.0000   Max.   :63.8526
##
##  Wind.Bearing..degrees. Visibility..km.    Loud.Cover Pressure..millibars.
##  Min.   :  0.0          Min.   : 0.000   Min.   :0    Min.   :   0
##  1st Qu.:129.0          1st Qu.: 6.311   1st Qu.:0    1st Qu.:1012
##  Median :175.0          Median : 9.982   Median :0    Median :1017
##  Mean   :186.1          Mean   : 9.471   Mean   :0    Mean   :1001
##  3rd Qu.:280.0          3rd Qu.:11.270   3rd Qu.:0    3rd Qu.:1022
##  Max.   :359.0          Max.   :16.100   Max.   :0    Max.   :1046
##
##  Daily.Summary      Temperature.Diff
##  Length:40660       Min.   :-4.811
##  Class :character   1st Qu.: 1.483
##  Mode  :character   Median : 2.589
##                     Mean   : 2.554
##                     3rd Qu.: 3.628
##                     Max.   :10.183
##
```

Now we'll do some more graphs.

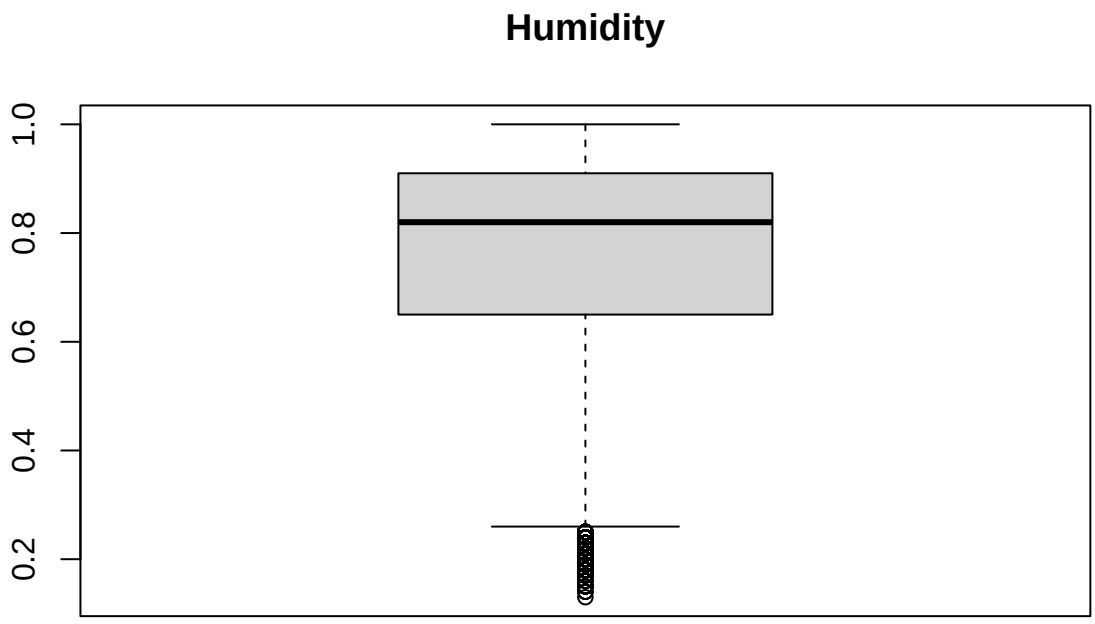
```
cor(df[4:7])
```

```
##                           Temperature..C. Apparent.Temperature..C.    Humidity
## Temperature..C.                1.00000000                0.9957386 -0.78282238
## Apparent.Temperature..C.       0.99573857                1.0000000 -0.75587228
## Humidity                      -0.78282238               -0.7558723  1.00000000
## Wind.Speed..km.h.             -0.08571425               -0.1545778 -0.06160538
##                           Wind.Speed..km.h.
## Temperature..C.                -0.08571425
## Apparent.Temperature..C.       -0.15457779
## Humidity                       -0.06160538
## Wind.Speed..km.h.               1.00000000
```

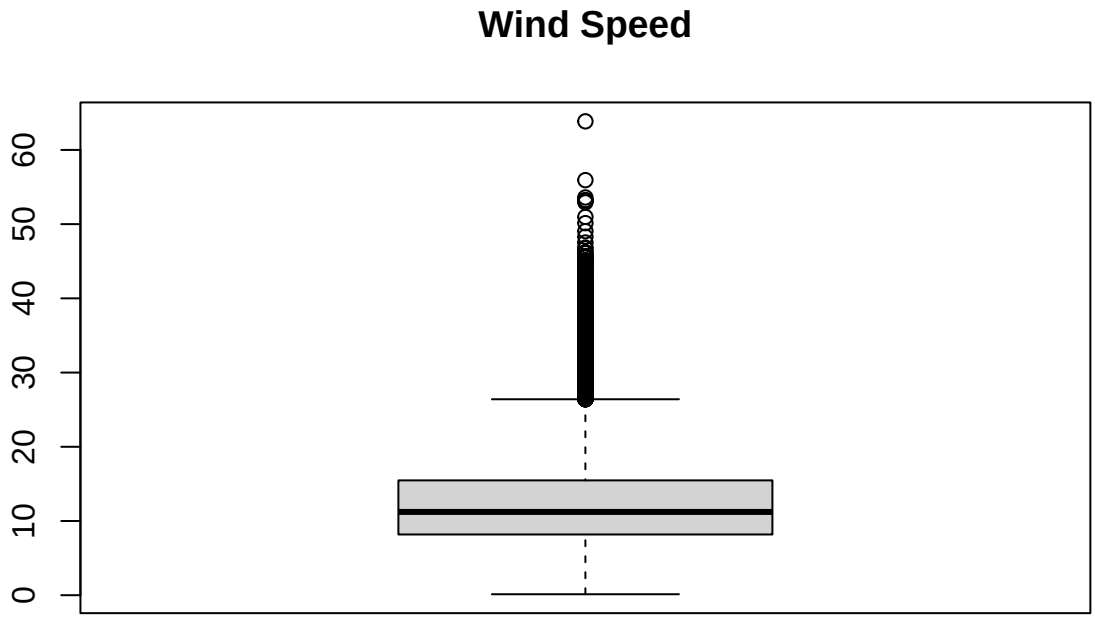
```
boxplot(df$Temperature.Diff,main="Apparent Temperature")
```



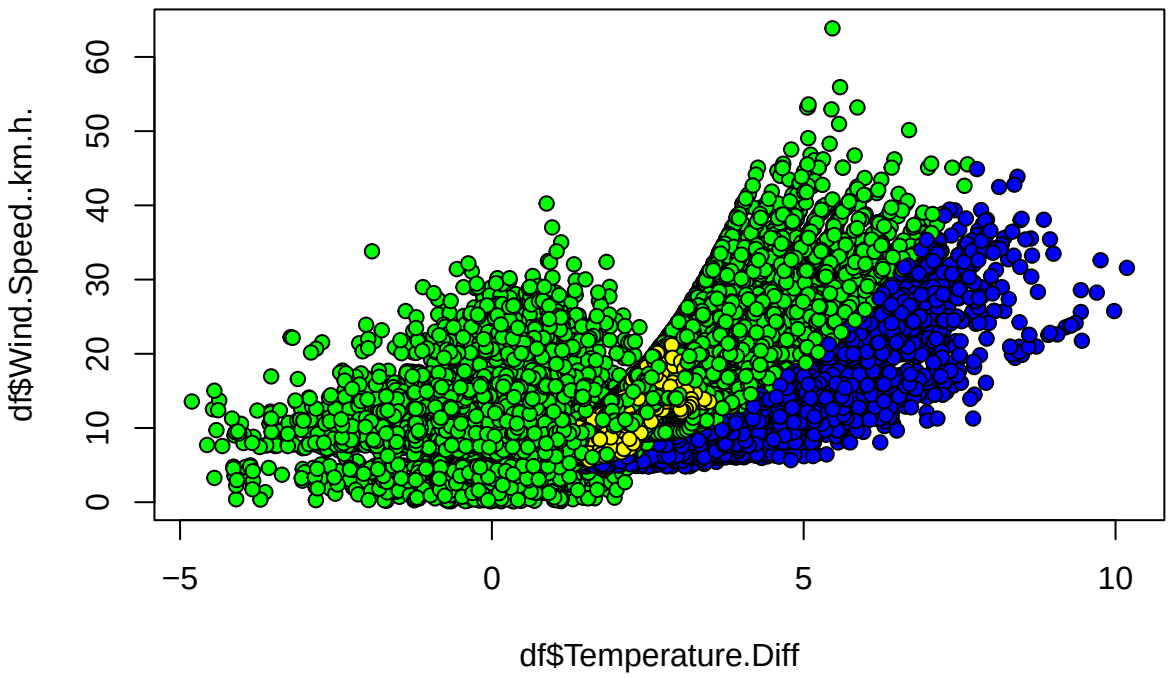

```
boxplot(df$Humidity,main="Humidity")
```



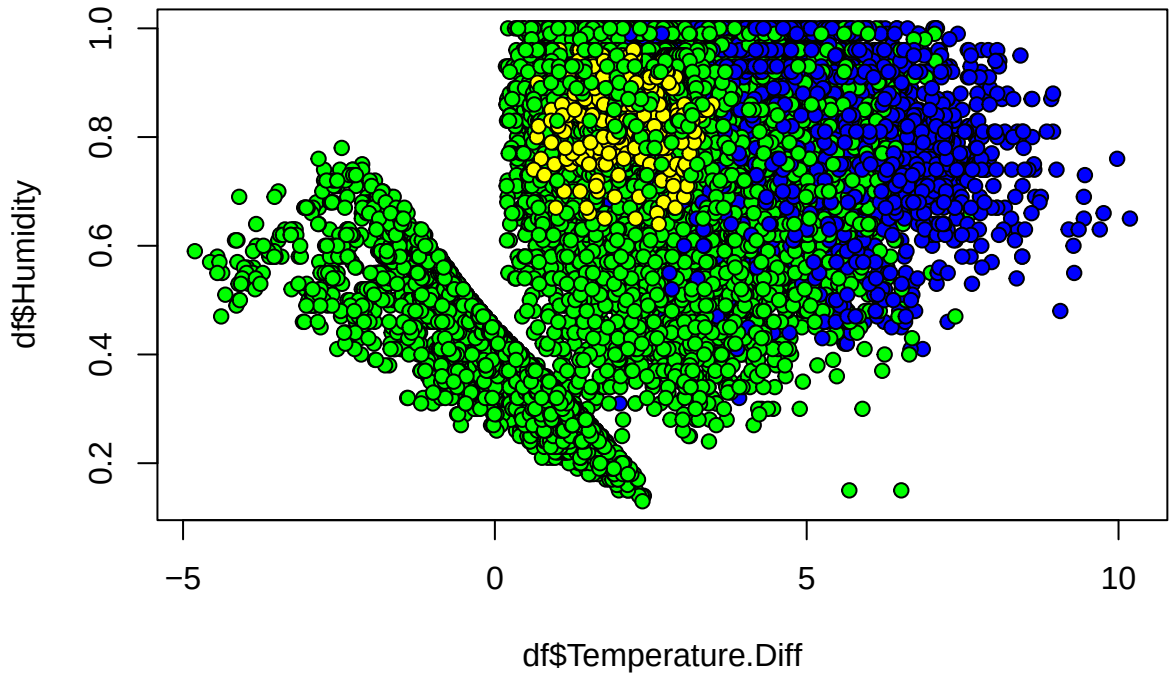

```
boxplot(df$Wind.Speed..km.h.,main="Wind Speed")
```



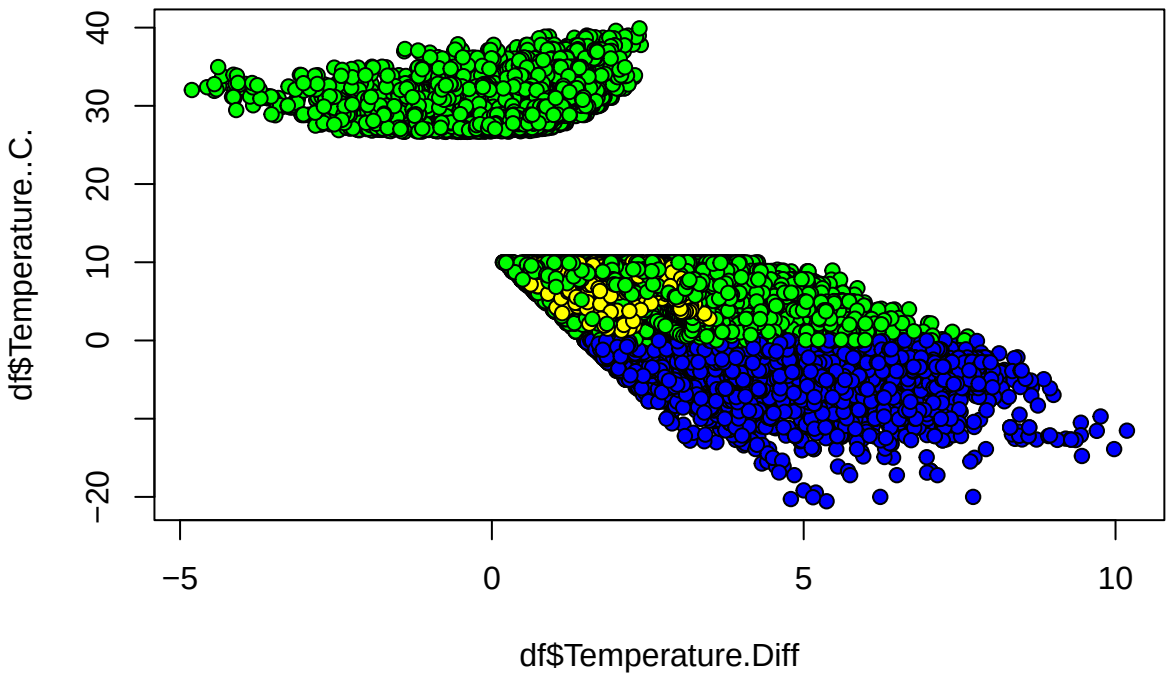

```
#pairs(df[4:7],main="Temperature, Humidity, and Wind Correlations")
plot(df$Temperature.Diff,df$Wind.Speed..km.h.,pch=21,bg=c("yellow","green","blue")[as.integer(df$Precip
```




```
plot(df$Temperature.Diff,df$Humidity,pch=21,bg=c("yellow","green","blue")[as.integer(df$Precip.Type)])
```




```
plot(df$Temperature.Diff, df$Temperature..C.,pch=21,bg=c("yellow","green","blue")[as.integer(df$Precip.T
```

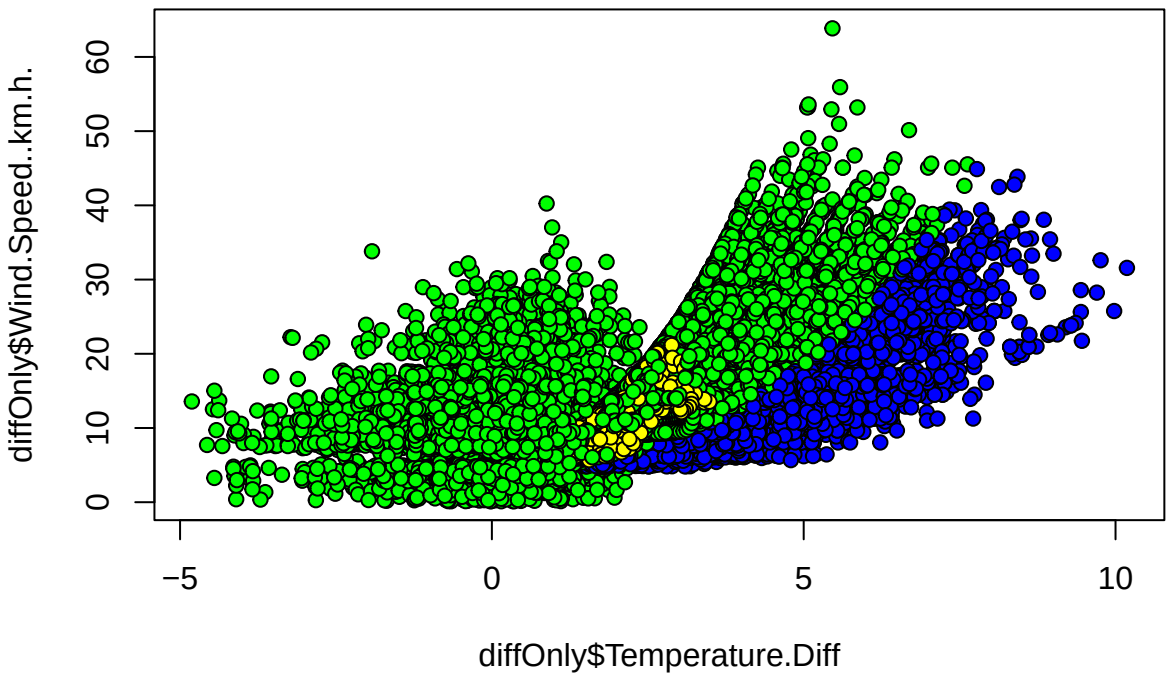


Before we move on to linear regression, we have one more question before we investigate disparities in the data.

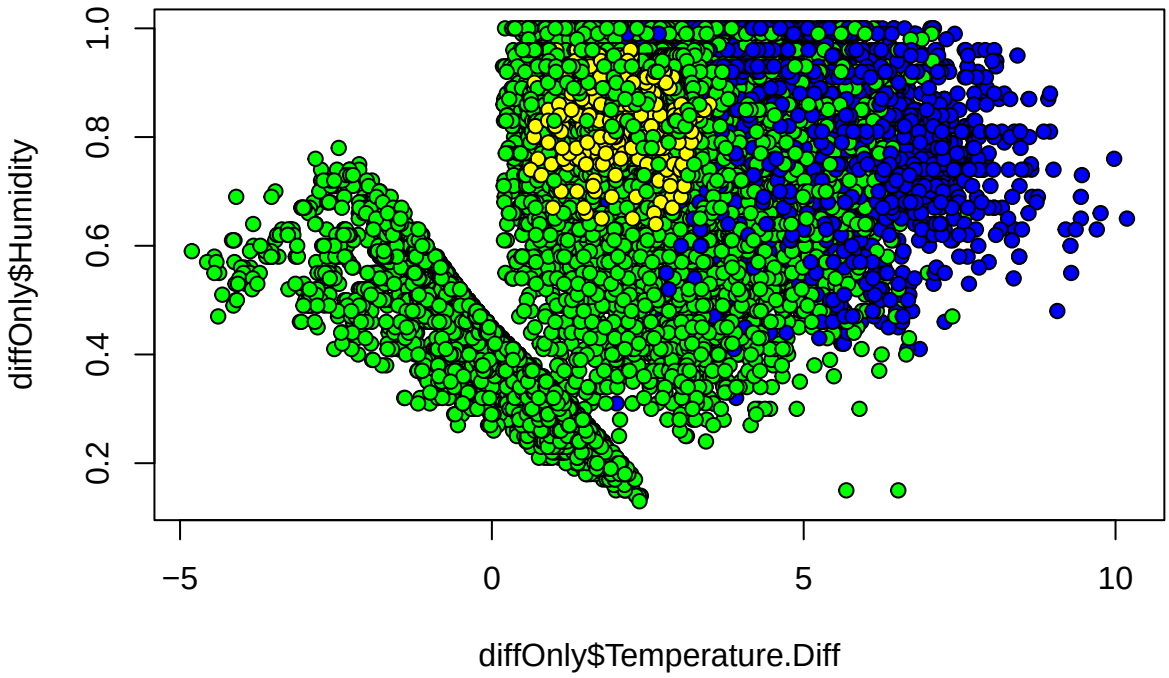
Since this project has multiple contributors, perhaps there are even more hidden NA's.

Let's see what the data looks like if we remove data where there is no difference.

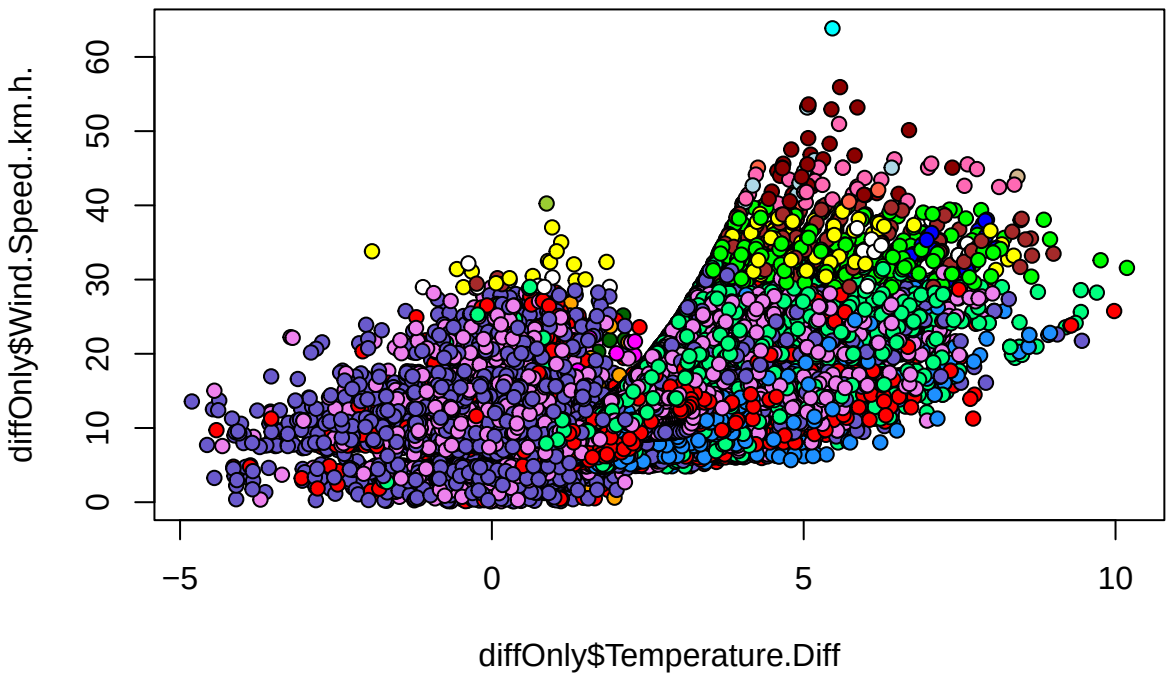
```
diffOnly <- df
diffOnly[,3:4][diffOnly[,3:3]==diffOnly[,4:4]] <- NA
diffOnly <- na.omit(diffOnly)
plot(diffOnly$Temperature.Diff,diffOnly$Wind.Speed..km.h.,pch=21,bg=c("yellow","green","blue")[as.integ
```



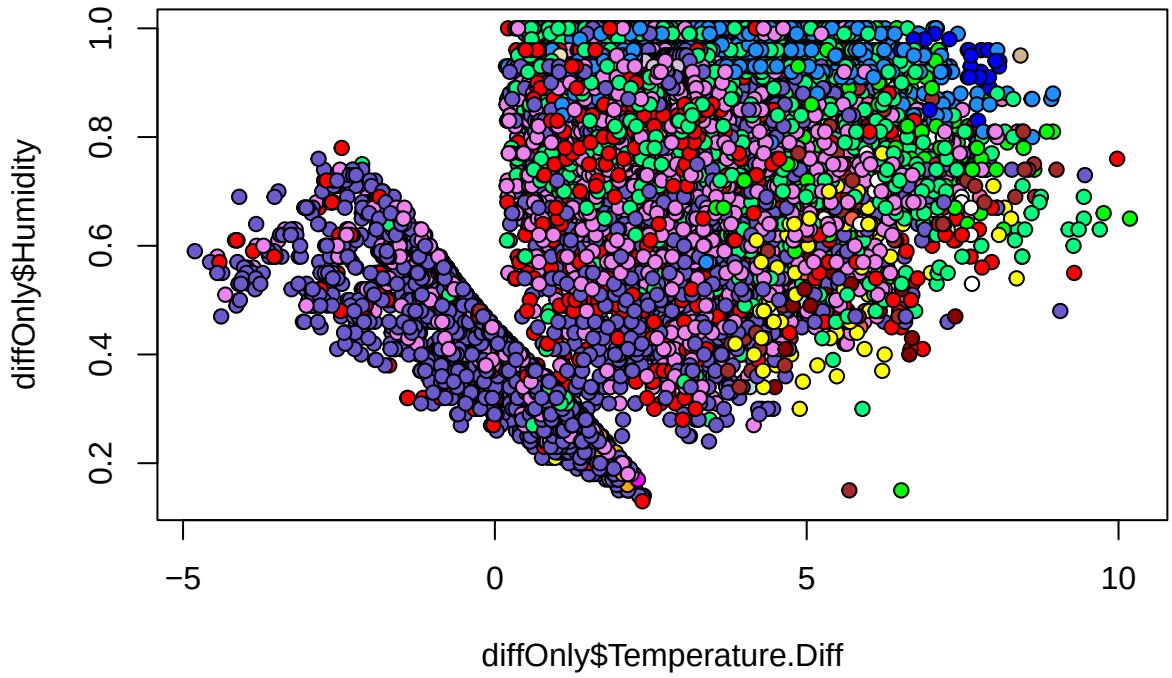

```
plot(diffOnly$Temperature.Diff,diffOnly$Humidity,pch=21,bg=c("yellow","green","blue")[as.integer(df$Pre
```



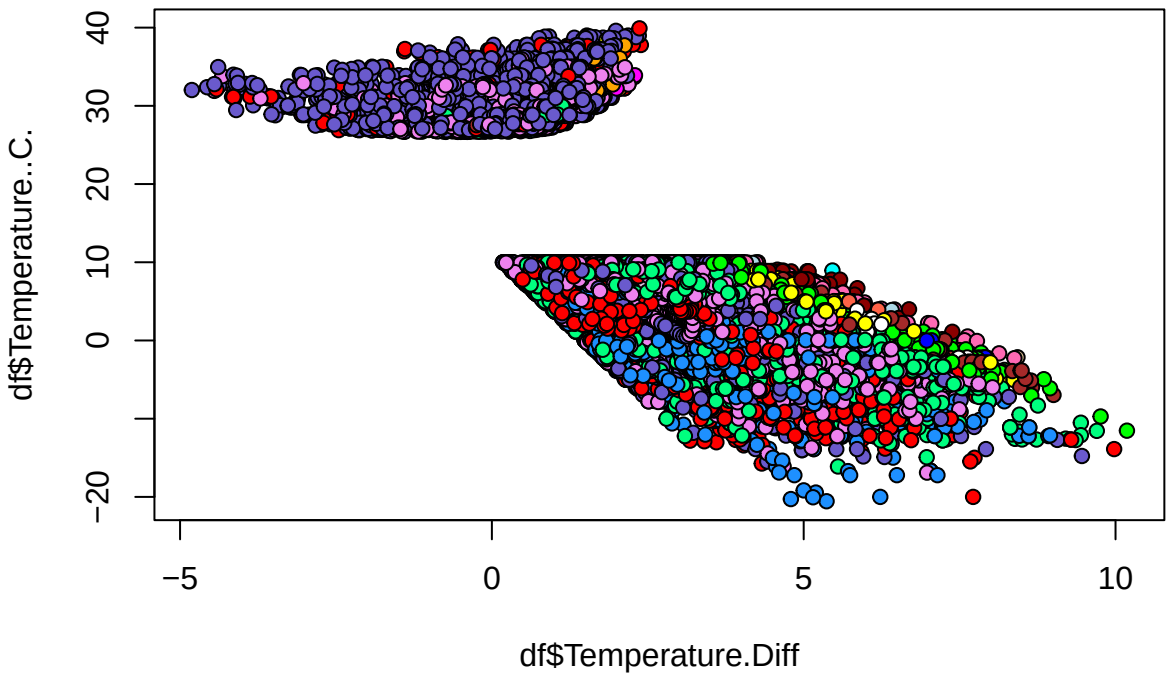

```
#These graphs show the overall descriptor of the weather. There are 27 options.
plot(diffOnly$Temperature.Diff,diffOnly$Wind.Speed..km.h.,pch=21,bg=c("white","aquamarine","blue","brow
```




```
plot(diffOnly$Temperature.Diff,diffOnly$Humidity,pch=21,bg=c("white","aquamarine","blue","brown","green
```




```
plot(df$Temperature.Diff, df$Temperature..C.,pch=21,bg=c("white","aquamarine","blue","brown","green","y
```



This does not seem to have effected data trends very much.

Looking at the data based on Summary does not help us much, but we can notice that the cloud of data that does not seem to have much trend is in the violet(18) and slateblue (20), or rather Mostly Cloudy and Partly Cloudy. There isn't much helpful we can do with that information at this time, however.

Since that seemed to cause no changes, but may have helped clean it up a small amount, let's write it to df. We'll also clean up the train and test data again.

```
df <- diffOnly
trainindex <- sample(1:nrow(df),nrow(df)*.8,replace=FALSE)
train <- df[trainindex,]
test <- df[-trainindex,]
```

Now, we'll move on to the regression.

Linear Regression: Simple

Let's start with a linear regression model with one predictor, wind speed, and summarize it.

```
simplelinreg <- lm(Temperature.Diff~Wind.Speed..km.h.,data=train)
summary(simplelinreg)
```

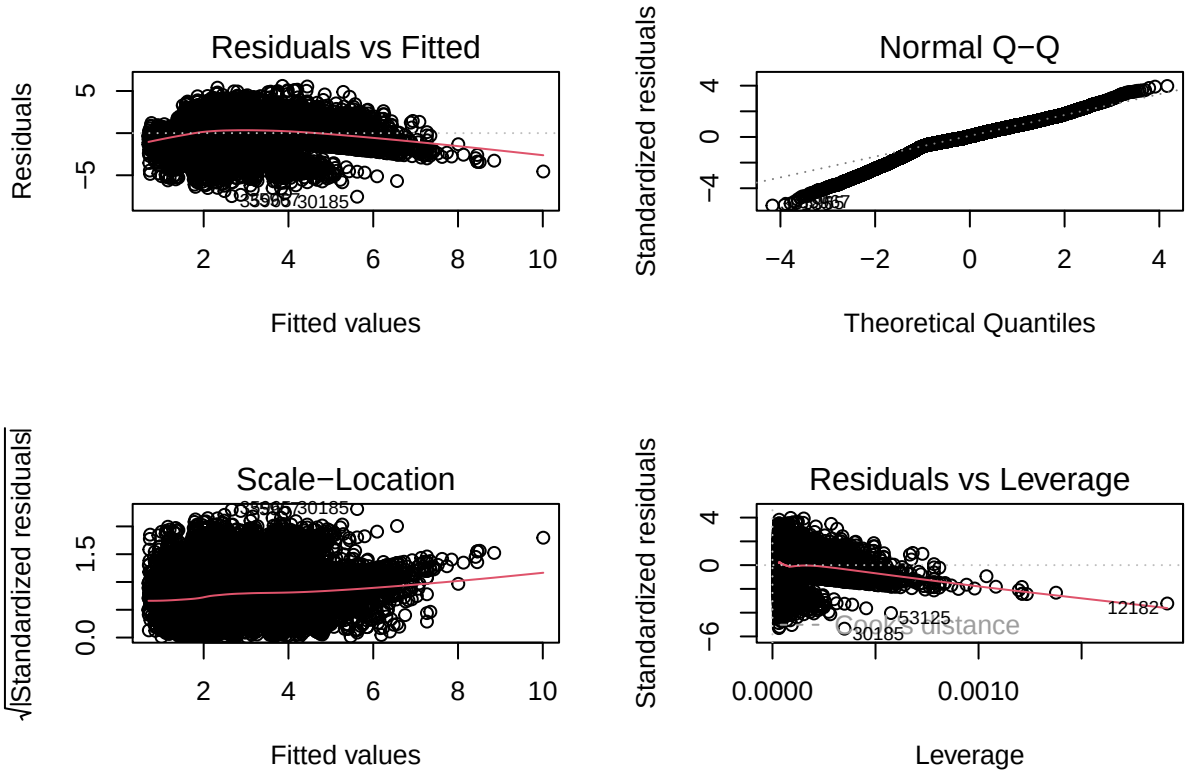
```
##
## Call:
## lm(formula = Temperature.Diff ~ Wind.Speed..km.h., data = train)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.5423 -0.6431  0.0712  0.9005  5.6035
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.67866    0.01727   39.29   <2e-16 ***
## Wind.Speed..km.h.  0.14615    0.00120  121.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.411 on 32526 degrees of freedom
## Multiple R-squared:  0.3131, Adjusted R-squared:  0.3131
## F-statistic: 1.483e+04 on 1 and 32526 DF,  p-value: < 2.2e-16
```

So, it's better than nothing it seems. The R^2 isn't great, but we can see that there's enough of a correlation to count. We could get a better reading by using the actual temperature, since those are very closely related, but one goal of this is learning to understand how the change in temperature works based on other factors.

Lets plot the residual errors, and evaluate.

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



We can see that the trends are fairly close to the given lines. They are in no way perfect, but they seem to get the gist. The most concerning piece seems to be Residuals vs Leverage. The given line implies that

we do have outlier (y-axis) leverage (x-axis) values that may influence our trend line. It may be something such as an issue during a case of severe weather, or a broken device used in data collection at that time.

As well, this is only the data from our simple regression. We will be able to see how other models compare at a later time.

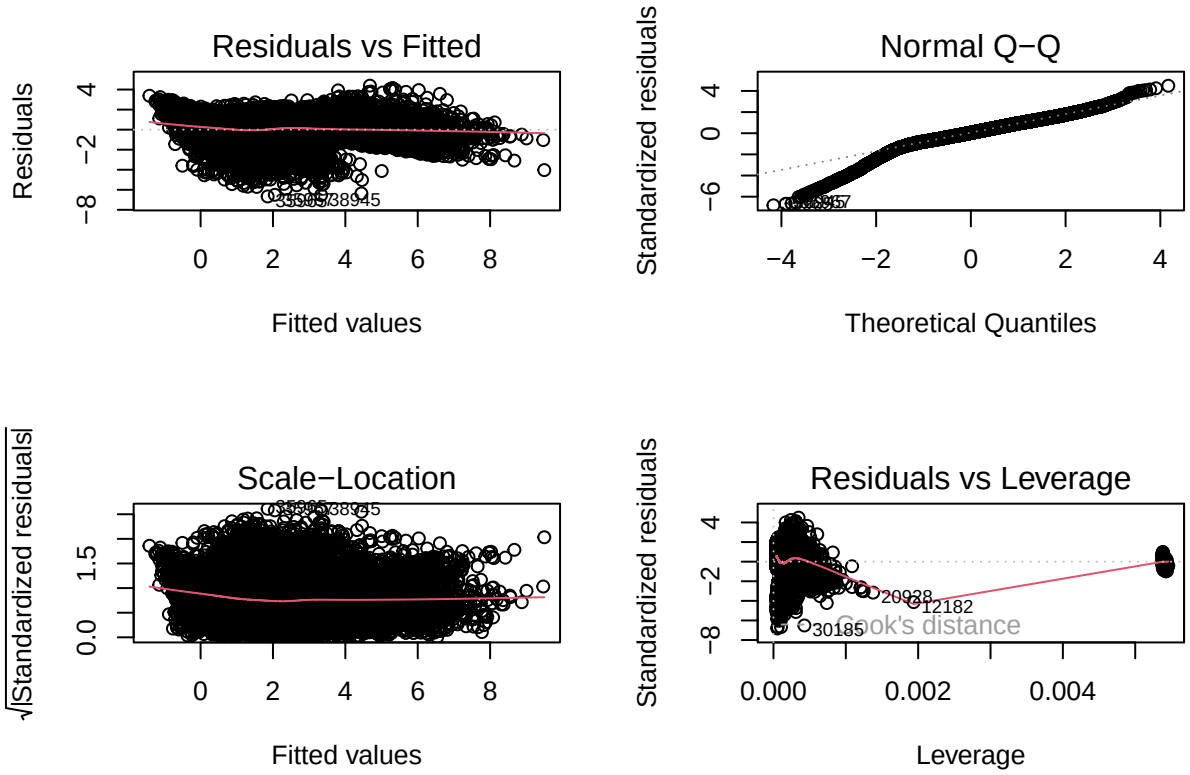
Linear Regression: Multiple

Let's up the complexity, now. We'll build a multiple linear regression model, and see if we can improve the accuracy.

```
multlinreg <- lm(Temperature.Diff~Humidity+Wind.Speed..km.h.+Precip.Type,data=train)
summary(multlinreg)
```

```
##
## Call:
## lm(formula = Temperature.Diff ~ Humidity + Wind.Speed..km.h. +
##     Precip.Type, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6642 -0.5270  0.0398  0.6317  4.3882
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.2316360  0.0759163 -29.396   <2e-16 ***
## Humidity           3.0433957  0.0275781 110.355   <2e-16 ***
## Wind.Speed..km.h.  0.1580733  0.0008347 189.380   <2e-16 ***
## Precip.Typerain    0.1436847  0.0720163   1.995    0.046 *
## Precip.Typesnow    1.8760791  0.0727876  25.775   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9778 on 32523 degrees of freedom
## Multiple R-squared:   0.67,  Adjusted R-squared:  0.6699
## F-statistic: 1.651e+04 on 4 and 32523 DF,  p-value: < 2.2e-16
```

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



We understand from our data exploration that Humidity, Wind Speed, and Precipitation Type all relate to the data in different ways. We can find different trends depending on what we're looking at, so we can ask the model to reference all of that data when its processing now. When the precipitation type was rain, it didn't add much to figuring things out, but knowing that it was in the snow range was very helpful.

It's doing better than our simple model, getting the R^2 up much more and a lower RSE. The Residuals vs Leverage chart looks like it has encountered some issues, however the two entirely separate sections does match up with some inconsistent trends that we noticed when we were graphing the attributes we planned on working with. The Residuals vs Fitted and Scale-Location graphs look comparatively stellar. Normal Q-Q is about the same.

Linear Regression: Combinations

Now let's go a step even farther. We'll use a combination of predictors, interaction effects, and polynomial regression to see if we can get even more accurate.

```
combolinreg <- lm(Temperature.Diff~poly(Humidity*Wind.Speed..km.h.)+Precip.Type+Summary,data=train)
summary(combolinreg)
```

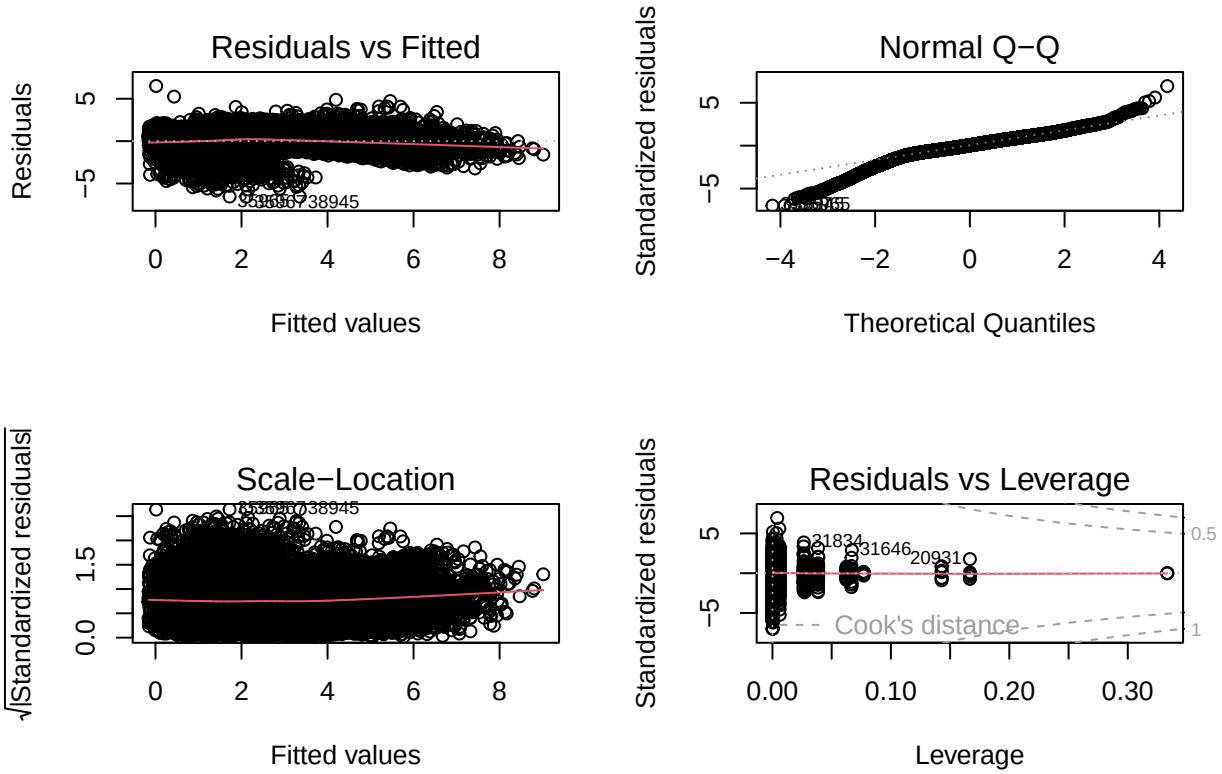
```
##
## Call:
## lm(formula = Temperature.Diff ~ poly(Humidity * Wind.Speed..km.h.) +
##     Precip.Type + Summary, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -6.5629 -0.4885  0.0405  0.6044  6.5009
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                               1.83657    0.16975  10.820
## poly(Humidity * Wind.Speed..km.h.)        223.15195    1.13970 195.799
## Precip.Typerain                           0.11849    0.06925   1.711
## Precip.Typesnow                           1.91715    0.07000  27.387
## SummaryBreezy and Foggy                  -1.35087    0.23819  -5.671
## SummaryBreezy and Mostly Cloudy          -0.34490    0.16878  -2.043
## SummaryBreezy and Overcast               -0.89024    0.16369  -5.438
## SummaryBreezy and Partly Cloudy           0.16717    0.17086   0.978
## SummaryClear                              0.41696    0.15612   2.671
## SummaryDangerously Windy and Partly Cloudy  -1.64055    0.95182  -1.724
## SummaryDrizzle                           -0.28248    0.30304  -0.932
## SummaryDry                                0.92062    0.28184   3.266
## SummaryDry and Mostly Cloudy              1.04067    0.38764   2.685
## SummaryDry and Partly Cloudy              1.05392    0.22780   4.626
## SummaryFoggy                              0.25100    0.15595   1.610
## SummaryLight Rain                        -0.72484    0.24044  -3.015
## SummaryMostly Cloudy                      0.36121    0.15535   2.325
## SummaryOvercast                           0.32325    0.15522   2.083
## SummaryPartly Cloudy                      0.13868    0.15563   0.891
## SummaryRain                              -0.37107    0.56394  -0.658
## SummaryWindy                              0.11952    0.41334   0.289
## SummaryWindy and Dry                     -1.09844    0.95185  -1.154
## SummaryWindy and Foggy                   -2.93145    0.95211  -3.079
## SummaryWindy and Mostly Cloudy           -1.46781    0.28760  -5.104
## SummaryWindy and Overcast                -2.29974    0.24091  -9.546
## SummaryWindy and Partly Cloudy           -0.43923    0.21698  -2.024
##                                          Pr(>|t|)
## (Intercept)                               < 2e-16 ***
## poly(Humidity * Wind.Speed..km.h.)        < 2e-16 ***
## Precip.Typerain                           0.08706 .
## Precip.Typesnow                           < 2e-16 ***
## SummaryBreezy and Foggy                  1.43e-08 ***
## SummaryBreezy and Mostly Cloudy           0.04101 *
## SummaryBreezy and Overcast               5.41e-08 ***
## SummaryBreezy and Partly Cloudy           0.32787
## SummaryClear                              0.00757 **
## SummaryDangerously Windy and Partly Cloudy  0.08479 .
## SummaryDrizzle                            0.35126
## SummaryDry                                0.00109 **
## SummaryDry and Mostly Cloudy              0.00727 **
## SummaryDry and Partly Cloudy             3.73e-06 ***
## SummaryFoggy                              0.10752
## SummaryLight Rain                         0.00257 **
## SummaryMostly Cloudy                      0.02007 *
## SummaryOvercast                           0.03730 *
## SummaryPartly Cloudy                      0.37290
## SummaryRain                               0.51054
## SummaryWindy                              0.77247
## SummaryWindy and Dry                      0.24850
## SummaryWindy and Foggy                    0.00208 **
```

```
## SummaryWindy and Mostly Cloudy              3.35e-07 ***
## SummaryWindy and Overcast                    < 2e-16 ***
## SummaryWindy and Partly Cloudy               0.04295 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9391 on 32502 degrees of freedom
## Multiple R-squared:  0.6957, Adjusted R-squared:  0.6955
## F-statistic:  2973 on 25 and 32502 DF,  p-value: < 2.2e-16
```

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

```
## Warning: not plotting observations with leverage one:
##   20297, 24839, 28039
```



Here, we added Summary as well as an interaction effect with precipitation. We made this decision based on the cloud of Partly Cloudy values that didn't seem to follow other data, and we can see that some specific Summary values were quite helpful in the result, and some were not.

Overall, though, R^2 is up a bit more, and RSE is down. It's not a huge change, but it does help. Humidity and Wind Speed seemed to have some similar trends and attributes when we graphed them, and the type of weather is related to the type of precipitation, which is why we had those certain attributes marked as an interaction effect.

The residuals are now very different from the other two models' results. It seems like the values are much more as intended, horizontal where they should be to indicate a good fit, though Q-Q seems to be the same. The outlying x and y observations also seem to be different than the ones the other models denoted.

Evaluation

With each new type, using more aspects of different Machine Learning model results, we were able to increase the model's ability to find lines within the data, which should help us when we predict our results. In general, the more ways we let the data interact, the better the resulting model seemed to be, so long as we did not do it blindly. Depending on how related certain attributes are, they need to be treated differently, since some attributes are a result of other attributes that may be included in the data.

So, in the end, the combination of interaction effects and multiple regression provided the best trends. Simple regression did not seem to fit the data well at all in comparison to the other models. The combination data may have only been a little about +.02 better on R^2 than multiple regression, however it is a significant enough change to be useful in data prediction.

Predictions

Using the three models, we will predict and evaluate using the metric correlation and MSE.

```
simplepred <- predict(simplelinreg,newdata=test)
simplecor <- cor(simplepred,test$Temperature.Diff)
simplemse <- mean((simplepred-test$Temperature.Diff)^2)
simplermse <- sqrt(simplemse)
multpred <- predict(multlinreg,newdata=test)
multcor <- cor(multpred,test$Temperature.Diff)
multmse <- mean((multpred-test$Temperature.Diff)^2)
multrmse <- sqrt(multmse)
combopred <- predict(combolinreg,newdata=test)
combocor <- cor(combopred,test$Temperature.Diff)
combomse <- mean((combopred-test$Temperature.Diff)^2)
combormse <- sqrt(combomse)
#Output results
print("-------Simple Model-------")
```

```
## [1] "-------Simple Model-------"
```

```
print(paste("Correlation: ", simplecor))
```

```
## [1] "Correlation:  0.553766132795759"
```

```
print(paste("MSE: ", simplemse))
```

```
## [1] "MSE:  2.01985524996796"
```

```
print(paste("RMSE: ", simplermse))
```

```
## [1] "RMSE:  1.42121611655932"
```

```
print("-------Multiple Model-------")
```

```
## [1] "-------Multiple Model-------"
```

```
print(paste("Correlation: ", multcor))
```

```
## [1] "Correlation:  0.820149093568624"
```

```
print(paste("MSE: ", multmse))
```

```
## [1] "MSE:  0.953857207069302"
```

```
print(paste("RMSE: ", multrmse))
```

```
## [1] "RMSE:  0.976656135530465"
```

```
print("-------Combo Model-------")
```

```
## [1] "-------Combo Model-------"
```

```
print(paste("Correlation: ", combocor))
```

```
## [1] "Correlation:  0.835127623256834"
```

```
print(paste("MSE: ", combomse))
```

```
## [1] "MSE:  0.881556455434504"
```

```
print(paste("RMSE: ", combormse))
```

```
## [1] "RMSE:  0.938912378997372"
```

```
anova(simplelinreg,multlinreg,combolinreg)
```

```
## Analysis of Variance Table
##
## Model 1: Temperature.Diff ~ Wind.Speed..km.h.
## Model 2: Temperature.Diff ~ Humidity + Wind.Speed..km.h. + Precip.Type
## Model 3: Temperature.Diff ~ poly(Humidity * Wind.Speed..km.h.) + Precip.Type +
##     Summary
##   Res.Df   RSS Df Sum of Sq        F    Pr(>F)
## 1  32526 64711
## 2  32523 31092  3     33619 12705.88 < 2.2e-16 ***
## 3  32502 28666 21      2427   131.02 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Judging by these values, we can verify our evaluation that the combination linear regression model was the best model for our data. The low MSE (mean squared error) in comparison to the simple and multiple models means that the mistakes made were smaller than others. The RMSE (root MSE) says that we were off, on average, .939 degrees Celsius. While this isn't entirely accurate, the range of the value was around

-5 degrees to +10 degrees, and if someone was predicting the weather the average person would likely be tolerant of a one degree difference. In that sense, the multiple regression would also be considered accurate enough to be helpful. The simple model is not terrible either, however the low correlation and high MSE do support the fact that there is much more room for improvement.

The difference in temperature is not extremely related to the wind speed, as we attempted in that first model. While it is a factor, the apparent temperature is a multifaceted issue better represented by numerous other effects, such as Humidity, precipitation, etc.

Our results were also very good considering we were purposely avoiding using one aspect of given data, and that there were disparities showing what may have been differences due to how different contributors to the data set initially reading data in different ways.

In summary, we can extract a surprising amount of data about the disparity in temperature based on wind, humidity, precipitation, and even descriptors of the sky. The more we combine usage of different attributes, acknowledging how they interact and work together, the better a result we can get.