Introduction

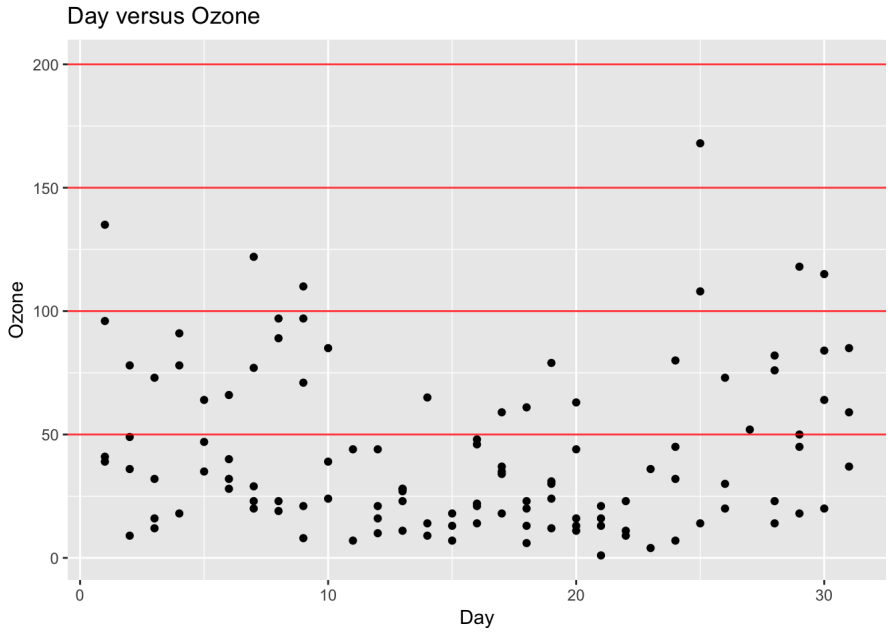
I chose to write my post on airquality in New York for two reasons. First and foremost, I want to write about something that is impactful and will have some sort of influence on people lives. Writing about airquality allows me to do that: airquality has a strong correlation with the overall health of people. Second, I chose the airquality dataset from R because it allows my data to be reproducible.

The main determinant of airquality is the level of Ozone in parts per million (PPS). The higher the ozone is, the less healthy the air quality becomes. It is defined that if the Ozone is below 50, that level is considered a safe. If it were in the range of 50 to 100, it would be considered acceptible, but sensitive peoples such as those with asthma are encouraged to be wary and exercise less. If the level were to be within 150 to 200, that is considered unhealthy and people are encouraged not to exercise. If the level exceeds 200, everyone is encouraged to avoid all exericise and stay indoors.

If we refer to the chart below, from the airquality dataset (of New York Metropolitan) we can see that on most days, the Ozone levels are relatively norma (below 100). However, there are 7 days where Ozone levels are particularly unhealthy.

```
library(ggplot2)
ggplot(airquality, aes(x = Day, y = Ozone)) + geom_point() + geom_hline(yintercept = seq(50, 200, 50), color = 're
d', alpha = 0.8) + ggtitle("Day versus Ozone")
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```



These types of “unhealthy” days are the data points I am concerned about. I think that it will be especially beneficial if people were given a forecast of airquality, especially when it reaches unhealthy levels so that they can prepare accordingly. I hope to resolve this issue by finding a variable (from the dataframe below) that is strongly correlated with Ozone and can successfully predict what Ozone levels will be in the future.

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

In order to find the best predictor for airquality (Ozone levels), I first constructed a correlation matrix, converted it to a dataframe using the “reshape” library, and then got rid of the NA vaues. With the new variable “melted_corr”, I created a correlation table, color coded respectively. Since our goal is to find the variable that is most strongly linked to Ozone, we will only be looking at the bottom panels of the correlations table. As you can see, temperature (in orange), is the most strongly correlated with airquality at 0.70 and therefore, it will be our variable of interest.

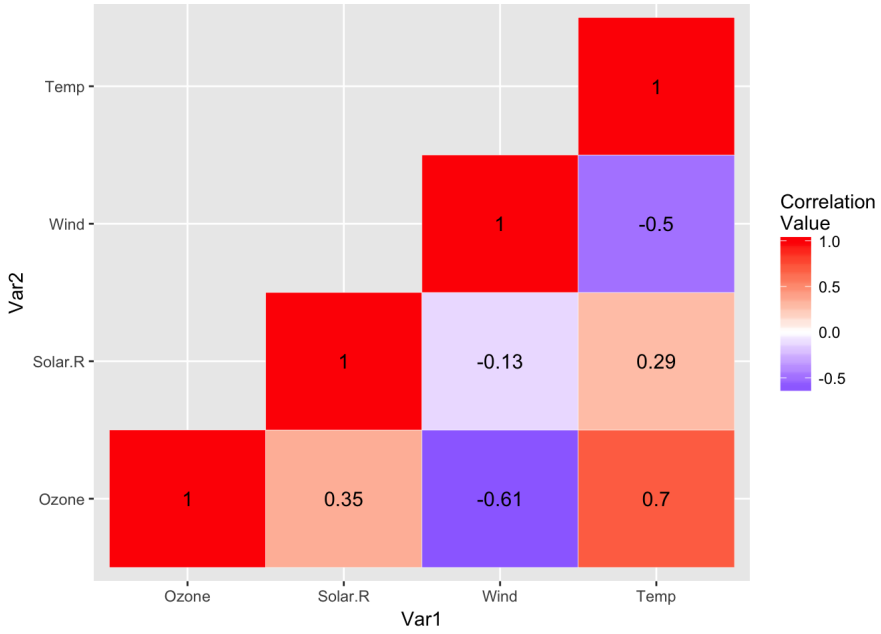
```
# Create the correlation matrix and convert upper triangle and lower triangles into NA values, repectively.
no_nas = airquality[!is.na(airquality$Solar.R) & !is.na(airquality$Ozone), ]
airqual <- no_nas[ , c(1,2,3,4)]
correlations = round(cor(airqual), 2)
corr1 = correlations
correlations[upper.tri(correlations)] = NA
corr1[lower.tri(corr1)] = NA
correlations
```

```
##         Ozone Solar.R Wind Temp
## Ozone    1.00      NA   NA   NA
## Solar.R  0.35    1.00   NA   NA
## Wind    -0.61   -0.13  1.0   NA
## Temp     0.70    0.29 -0.5    1
```

```
# Convert the correlation matrix into dataframe form.
library(reshape2)
melted_corr = melt(correlations, na.rm = TRUE)
melted_corr1 = melt(corr1, na.rm = TRUE)
melted_corr
```

```
##       Var1    Var2 value
## 1    Ozone   Ozone  1.00
## 2  Solar.R   Ozone  0.35
## 3     Wind   Ozone -0.61
## 4     Temp   Ozone  0.70
## 6  Solar.R Solar.R  1.00
## 7     Wind Solar.R -0.13
## 8     Temp Solar.R  0.29
## 11    Wind    Wind  1.00
## 12    Temp    Wind -0.50
## 16    Temp    Temp  1.00
```

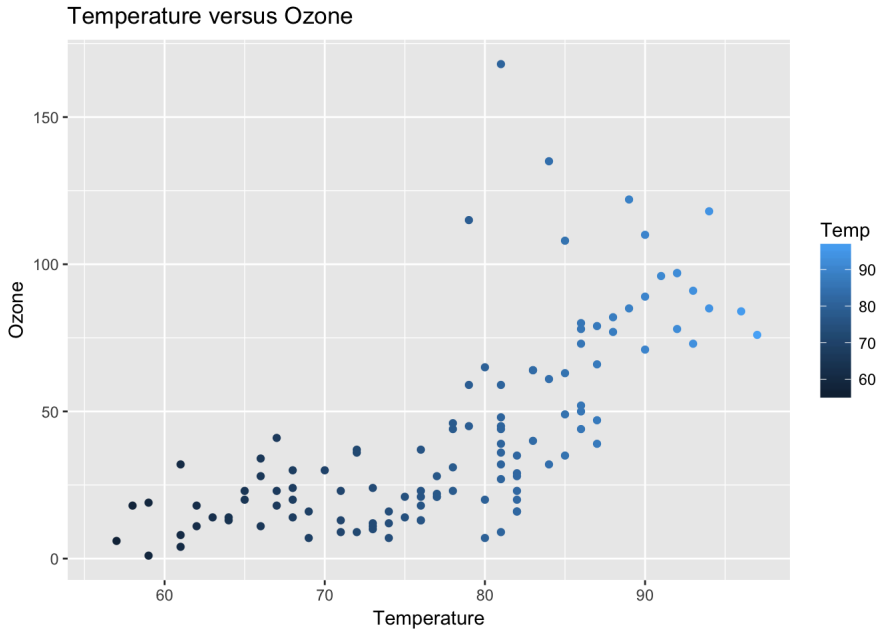
```
# Create the heatmap
library(ggplot2)
ggplot(data = melted_corr, aes(Var1, Var2, fill = value, label = value)) + geom_tile(color = 'white') +
  scale_fill_gradient2(low = 'blue', high = 'red', mid = 'white', name="Correlation\nValue") +
  geom_text(data = melted_corr1 ,aes(Var2, Var1, label = value), color = "black", size = 4)
```



Indeed, if we were to plot temperature against Ozone, we can see a positive correlation (seen below). Higher temperatures generally are linked to higher Ozone levels and vise versa.

```
ggplot(airquality, aes(x = Temp, y = Ozone, color = Temp)) + geom_point() + xlab("Temperature") + ggtitle("Tempera
ture versus Ozone")
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```



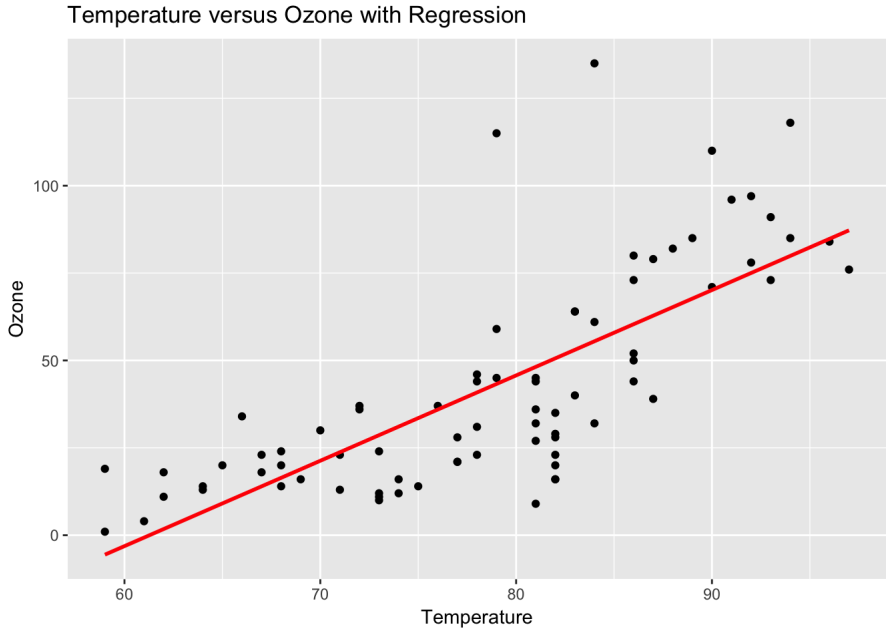
To quantify this linkage between temperature and ozone, we need to use a linear regression model. In order to do so, we must split our data into training and test sets. The training set is the set used to obtain the regression line (these will be all of our predicted values). The test set is used to test the accuracy of our regression line. Below, I use the library caTools to split my training set into a dataframe with 58 rows and my test set into a data of 35 rows.

```
library(caTools)
ind = sample.split(airqual$Ozone,SplitRatio = 0.7)
train = airqual[ind, ]
test = airqual[!ind, ]
c(nrow(train), nrow(test))
```

```
## [1] 77 34
```

Now that we have our test and training sets, we need to obtain the regression line which will serve as our predictor. To do so, I call the “lm” function on the “Ozone” and “Temp” vectors of the train set. Since the output of this is a list, I convert it to a vector by calling “unlist.” Because the first and second elements of the vector are the intercept and slope respectively, I index accordingly to obtain their values. Using this slope and intercept, I use the “Temp” vector from the test set to calculate the predicted values, which outputs a vector of length 35. Finally, I calculate the root mean squared error between the actual values for air quality and the predicted values of the test set which gives a root mean squared error of 24.1.

```
# Plotting the linear regression for the training set.
ggplot(train, aes(x = Temp, y = Ozone)) + geom_point() + xlab("Temperature") +
ggtitle("Temperature versus Ozone with Regression") + geom_smooth(method = 'lm', se = FALSE, color = 'red')
```

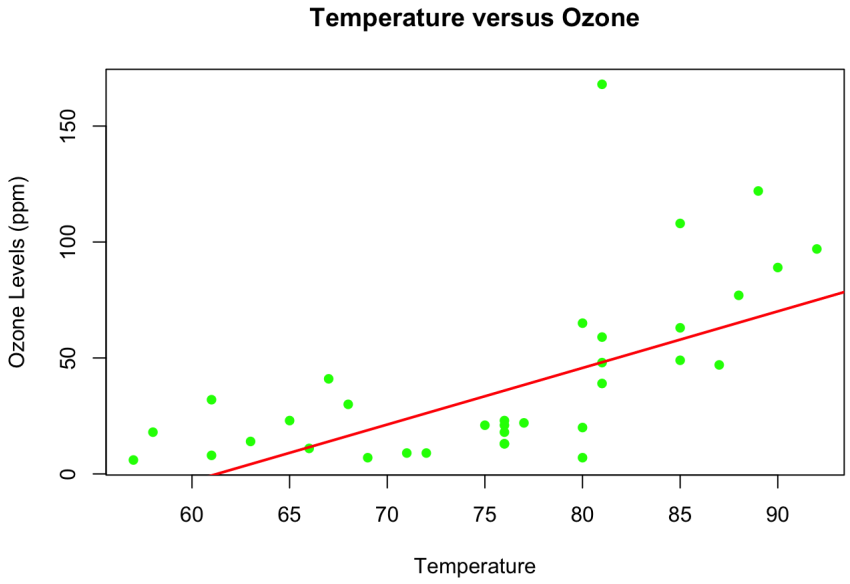


```
# Obtaining the formula for the regression line on the training set.
intercept = as.numeric(unlist(lm(train$Ozone ~ train$Temp))[1])
slope = as.numeric(unlist(lm(train$Ozone~ train$Temp))[2])

# Using the slope and intercept to calculate predicted values for the air quality.
predictions = test$Temp * slope + intercept
predictions
```

```
##  [1]  13.9717568   9.0879061  -0.6797952  11.5298314  -8.0055713
##  [6] -10.4474966  16.4136821  -0.6797952  35.9490848  57.9264128
## [11]  65.2521888  75.0198902  45.7167862  48.1587115  57.9264128
## [16]  57.9264128  48.1587115  48.1587115  67.6941142  70.1360395
## [21]  45.7167862  38.3910101  35.9490848  26.1813835  48.1587115
## [26]  62.8102635  45.7167862  33.5071595  35.9490848  23.7394581
## [31]  35.9490848  18.8556074   4.2040554  35.9490848
```

```
# Drawing the regression line on the test set.
plot(test$Temp, test$Ozone, ylab = 'Ozone Levels (ppm)', xlab = 'Temperature', main = 'Temperature versus Ozone',
     col = 'green', pch = 16)
abline(lm(train$Ozone ~ train$Temp), lwd = 2, col = 'red')
```




```
# Calculate the root mean squared error.
RMSE = sqrt(mean((test$Ozone - predictions)**2))
print(RMSE)
```

```
## [1] 29.7164
```

Conclusion

A mean squared Error of 24.1 is reasonable but somewhat large. I think that the RMSE may have been this large for a number of reasons. First, the training set only had 58 data points to learn from. Therefore, small variations in the data could ultimately lead to large variations in the predictions. Less data points also means that the given data was not a good representative of the entire population from which it was drawn. Another reason for the error can be attributed to a certain number of outliers. There were approximately 7 points above a measurement of 100 for Ozone (ppm), which is drastically higher than the sample average. This could have dragged the slope of regression line up and ultimately increased the amount of overall error.

One way in which the RMSE can be reduced is by incorporating more variables into the prediction, such as using Wind and Solar.R to perform a multivariate regression. The problem with this strategy is that since both Wind and Solar.R are also correlated with Temp in the same way that they correlate with Ozone, the overall effect on RMSE would not be substantial.

All in all, I think that my findings serve as a good starting point for more research and hope that society at large will also place more importance on air quality and its overall effect on people's lives.

References

1. http://www.stat.wisc.edu/~larget/stat302/chap2.pdf
2. https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/airquality.html
3. https://airnow.gov/index.cfm?action=pubs.aqiguideozone
4. http://www.sthda.com/english/wiki/correlation-matrix-a-quick-start-guide-to-analyze-format-and-visualize-a-correlation-matrix-using-r-software
5. http://blogs.ei.columbia.edu/2016/06/06/air-quality-pollution-new-york-city/
6. http://www.upout.com/blog/new-york-city/air-quality-map-of-nyc-rip-lungs

7. https://www1.nyc.gov/assets/doh/downloads/pdf/eode/eode-air-quality-impact.pdf

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.