Project 2

Importing the data and saving it

```r
AirQ <- read.csv(file = '/Users/trishalvarma/Desktop/Changping.csv')
```

# Data Clearning

Website: http://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data
From: "UCI Machine Learning Repositor" — Used any(is.na(data)) to see if there are any data missing from the columns. Then used na.omit, because data can be filled with NA and not show up as a blank data. Eliminates missing values. Provided the link from where the data is downloaded.

Steps taken are listed above, to remove any blank/na values. #Datal Cleanup

```r
AirQ$No <- NULL
AirQ$PRES <- NULL
AirQ$DEWP <- NULL
AirQ$station <- NULL
AirQ$RAIN <- NULL
AirQ$TEMP <- NULL
AirQ$wd <- NULL
AirQ$O3 <- NULL

any(is.na(AirQ))
```

```
## [1] TRUE
```

```r
na.rm = TRUE
```

```r
df <- AirQ

df$year <- as.factor(df$year)
df$month <- as.factor(df$month)
df$day <- as.factor(df$day)
df$PM2.5 <- as.factor(df$PM2.5)
df$PM10 <- as.factor(df$PM10)
df$SO2 <- as.factor(df$SO2)
df$CO <- as.factor(df$CO)
df$NO2 <- as.factor(df$NO2)
```

Data Exploration

Using more than 5 functions to look at the Air Quality data, then providing a informative R graphs using a historgram, and a plot chart to show the just the sheer amount of data that was collected.

```
class(AirQ)

## [1] "data.frame"

head(AirQ, 5)

##   year month day hour PM2.5 PM10 SO2 NO2  CO WSPM
## 1 2013     3   1    0     3    6  13   7 300  0.5
## 2 2013     3   1    1     3    3   6   6 300  0.7
## 3 2013     3   1    2     3    3  22  13 400  0.2
## 4 2013     3   1    3     3    6  12   8 300  1.0
## 5 2013     3   1    4     3    3  14   8 300  2.1

tail(AirQ, 2)

##       year month day hour PM2.5 PM10 SO2 NO2  CO WSPM
## 35063 2017     2  28   22    11   20    3  15 500  1.4
## 35064 2017     2  28   23    20   25    6  28 900  1.9

dim(AirQ)

## [1] 35064    10

summary(AirQ)

##       year          month             day             hour      
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   : 0.00  
##  1st Qu.:2014   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 5.75  
##  Median :2015   Median : 7.000   Median :16.00   Median :11.50  
##  Mean   :2015   Mean   : 6.523   Mean   :15.73   Mean   :11.50  
##  3rd Qu.:2016   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:17.25  
##  Max.   :2017   Max.   :12.000   Max.   :31.00   Max.   :23.00  
##                                                                 
##      PM2.5            PM10             SO2                NO2         
##  Min.   :  2.0   Min.   :  2.00   Min.   :  0.2856   Min.   :  1.848  
##  1st Qu.: 18.0   1st Qu.: 34.00   1st Qu.:  2.0000   1st Qu.: 22.000  
##  Median : 46.0   Median : 72.00   Median :  7.0000   Median : 36.000  
##  Mean   : 71.1   Mean   : 94.66   Mean   : 14.9589   Mean   : 44.182  
##  3rd Qu.:100.0   3rd Qu.:131.00   3rd Qu.: 18.0000   3rd Qu.: 60.358  
##  Max.   :882.0   Max.   :999.00   Max.   :310.0000   Max.   :226.000  
##  NA's   :774     NA's   :582      NA's   :628        NA's   :667      
##        CO             WSPM      
##  Min.   :  100   Min.   : 0.000  
##  1st Qu.:  500   1st Qu.: 1.000  
##  Median :  800   Median : 1.500  
##  Mean   : 1152   Mean   : 1.854  
##  3rd Qu.: 1400   3rd Qu.: 2.300
##  Max.   :10000   Max.   :10.000
##  NA's   :1521    NA's   :43
```

```r
names(AirQ)
```

```
##  [1] "year"  "month" "day"   "hour"  "PM2.5" "PM10"  "SO2"   "NO2"   "CO"
## [10] "WSPM"
```

```r
# Visual Data Exploration
hist(AirQ$NO2,
     main = "Histogram of NO2 in Beijing",
     xlab = "NO2",
     xlim=c(0,220),
     col = "chartreuse4")
```

```r
plot(AirQ$NO2, col = rgb(0,0,0, alpha =.3))
```

#Running Algorithms on the data set.

Linear regression run on NO2 + PM10 + PM2.5 in comparison to hour + day to find strong predictors for later tests to be comepleted for this data. I will be paired with other data to corroborate the increase in polution in comparision to either hour, day, or year.

PM10 and PM2.5 refer to the diameter of the particulates pollution in the air which is a good comparision with hour and time with rising levels of harmful gasses.

We use hour and day so we can see the trend throughout the day of the levels of harmful

```r
##### Linear Regression #####
set.seed(1234)
sample_i <- sample(1:nrow(AirQ), .75*nrow(AirQ), replace=FALSE)
train_no2 <- AirQ[sample_i,]
test_no2 <- AirQ[-sample_i,]

lm1 <- lm(NO2 + PM10 + PM2.5 ~ hour + day, data=train_no2)
summary(lm1)
```

```
##
## Call:
## lm(formula = NO2 + PM10 + PM2.5 ~ hour + day, data = train_no2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -211.37 -128.21  -48.09   78.05 1151.46
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 189.5290     2.8223  67.154   <2e-16 ***
## hour          1.4003     0.1544   9.071   <2e-16 ***
## day           0.2883     0.1215   2.373   0.0177 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 170.8 on 25520 degrees of freedom
##   (775 observations deleted due to missingness)
## Multiple R-squared:  0.003436,   Adjusted R-squared:  0.003358
## F-statistic:    44 on 2 and 25520 DF,  p-value: < 2.2e-16

plot(lm1)
```

```
# Day is not a strong prediction, so we will run other algorithms to look at
the data in the next chunk of code.

pd1 <- predict(lm1, newdata = test_no2)

cor_lm <- cor(pd1,test_no2$NO2)
mse_lm <- mean(pd1 - test_no2$NO2)
```

We will use kNN function, which can be used for both classification and regression problems. Higher range varibales can have bias, and it does get expensive to run this algorithm. However, it might help us remove some noise from the data we have built.

```
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(lattice)
library(ggplot2)

no2_knn <- knnreg(NO2~year, data = AirQ, k = 5)
pred1 <- predict(lm1, test_no2[,1:10])

cor_knn1 <- cor(pred1 - as.matrix(test_no2$year))

print(paste("Correlation: ", cor_knn1))

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

We get a correlation of 1, which means that both variables move in the same direction together. So there is a strong connection to the the NO2 levels moving in the same direction as year goes on. A negative correlation would have shown us that as the years go on, levels of NO2 in the atmosphere in Beijing is decreasing, which would have been a good sign.

However that is not the case for us.

```
test_no2$NO2 <- as.factor(test_no2$NO2)
test_no2$year <- as.factor(test_no2$year)

library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(1234)

fit_rf <- randomForest(formula = NO2~.,
                       data=test_no2,
                       importance = TRUE,
                       pr0ximity = TRUE,
                       mtry =1,
                       na.action = na.roughfix)

fit_rf
```

```
##
## Call:
##  randomForest(formula = NO2 ~ ., data = test_no2, importance = TRUE,
pr0ximity = TRUE, mtry = 1, na.action = na.roughfix)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 93.61%
## Confusion matrix:
##          1.8477 2 3 4 5 5.5431 6 7 8 8.0067 9 9.6491 9.8544 10 11 11.7021
## 1.8477        0 0 0 0 0      0 0 0 0      0 0      0      0  0  0       0
## 2             0 3 0 3 2      0 2 3 1      0 0      0      0  4  0       0
## 3             0 0 2 0 1      0 1 1 0      0 1      0      0  1  0       0
## 4             0 1 1 7 3      0 1 3 0      0 4      0      0  3  2       0
## 5             0 3 0 4 2      0 1 1 3      0 1      0      0  4  2       0
## 5.5431        0 0 0 0 0      0 0 0 0      0 0      0      0  0  0       0
## 6             0 0 0 3 1      0 8 5 2      0 6      0      0  3  4       0
## 7             0 1 0 3 2      0 3 2 4      0 5      0      0  7  7       0
## 8             0 1 0 2 1      0 2 7 0      0 8      0      0  7  5       0
## 8.0067        0 0 0 0 0      0 0 0 0      0 0      0      0  0  0       0
## 9             0 0 0 4 0      0 1 4 4      0 7      0      0 16  4       0
## 9.6491        0 0 0 0 0      0 0 0 0      0 0      0      0  0  0       0
```

Results Analysis

Ranking of algorithms

1) random Forest
2) Linear Regression
3) kNN

The performance of random forest was next to none as it made over 500 trees and the reported on the error rate. Linear regression comes right after that since it is a regression model on such a large data, it was easier to use it than the kNN function.

kNN function was the worst case scenario for the data.

script was able to learn that target variables were weak linear relatinship however, still able to complete the test.