Data Analysis Methods

Predicting ufo sightings based on air quality

Team Member 1: Karishma Bhatnagar (M12865125) 33.3%

Team Member 2: Tanvi Kini (M12934812) 33.3% (Evaluator)

Team Member 3: Ankeeta Majumder (M12825237) 33.3%

2019

Contents

[Predicting UFO sightings based on Air Quality 2](#_Toc2259770)

[Introduction 2](#_Toc2259771)

[Reading the data 2](#_Toc2259772)

[Exploratory Data analysis 2](#_Toc2259773)

[Missing value check 3](#_Toc2259774)

[Visualizing the data 3](#_Toc2259775)

[Dealing with categorical variables 5](#_Toc2259776)

[Split data into training and test data set 6](#_Toc2259777)

[Model Specification 7](#_Toc2259778)

[Multiple Linear Regression – Model 1 7](#_Toc2259779)

[Multiple Linear Regression – Model 2 8](#_Toc2259780)

[Parameter Estimation 8](#_Toc2259781)

[Parameter Estimation – Model 1 8](#_Toc2259782)

[Parameter Estimation – Model 2 10](#_Toc2259783)

[Model Adequacy Checking 10](#_Toc2259784)

[Residual Plots and Residual Analysis 10](#_Toc2259785)

[Transformation 11](#_Toc2259786)

[Multicollinearity Check 11](#_Toc2259787)

[Model Validity 14](#_Toc2259788)

[Logistic Regression 15](#_Toc2259789)

[Model Specification 15](#_Toc2259790)

[Parameter Estimation 16](#_Toc2259791)

[Model Adequacy Checking 17](#_Toc2259792)

[Plot Logistic regression 19](#_Toc2259793)

[Confusion matrix 21](#_Toc2259794)

[Model Use 21](#_Toc2259795)

[Confusion Matrix 23](#_Toc2259796)

[Conclusion 23](#_Toc2259797)

# Predicting UFO sightings based on Air Quality

## Introduction

We will be using a Kaggle data set (<https://www.kaggle.com/infof422henni/ufo-air-quality/version/3>), this data set explores how sightings of UFOs in the US are affected by the air quality at the time of sighting. The data set has dates, locations and climate information about 4 main gases – NO2, O3, SO2 and CO about the UFO sightings that occurred in the USA together with noise to include observations that have no UFO sighting.

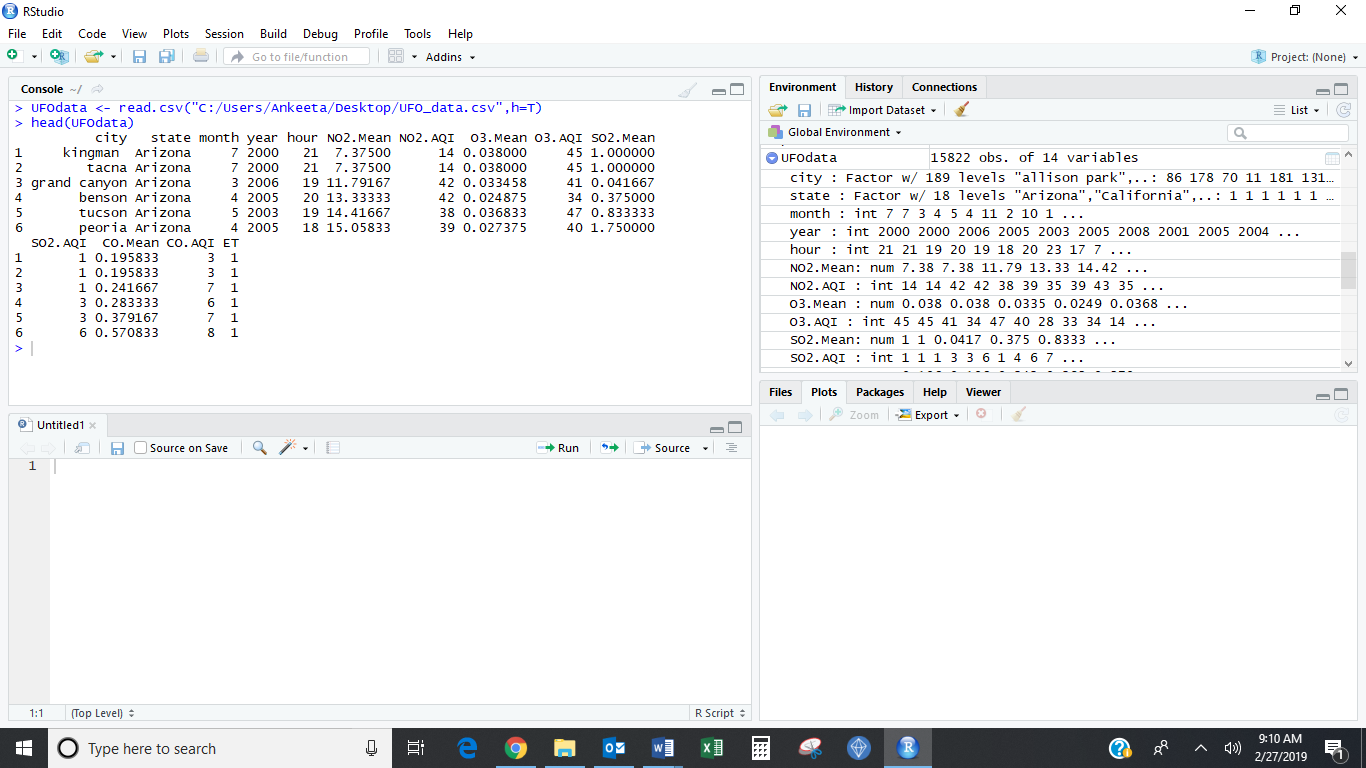
## Reading the data

First, we will read the csv file into our R Studio:

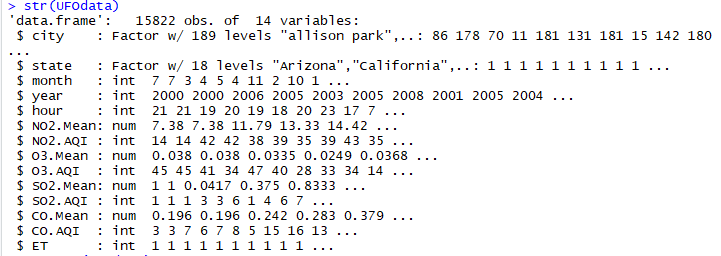
R Code:

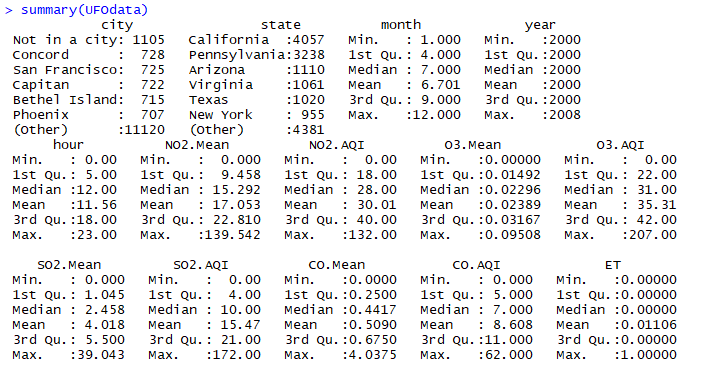
>UFOdata <- read.csv(“C:/Users/Ankeeta/Desktop/UFO\_Data.csv”,h=T)

R Output:



## Exploratory Data analysis





## Missing value check

We already removed the rows and columns with “Null” values during data cleansing process from our CSV file, but we will still want to check whether any missing values are present in our current dataset before proceeding further with our regression analysis:

R Code:

> sum(is.na(UFOdata)

R Output:



From the output above we see that there are no missing values in our current dataset.

## Visualizing the data

1. Plot Histogram: We will plot a histogram for all the numeric variables to see the distribution:

R Code:

plotHistogram <- function(y)

{

for (a in 1:ncol(y))

{

if(col.type[a,1]=="numeric"|col.type[a,1]=="integer")

{

hist(UFOdata[,a],

xlab = colnames(y[a]),

main = colnames(y[a]))

}

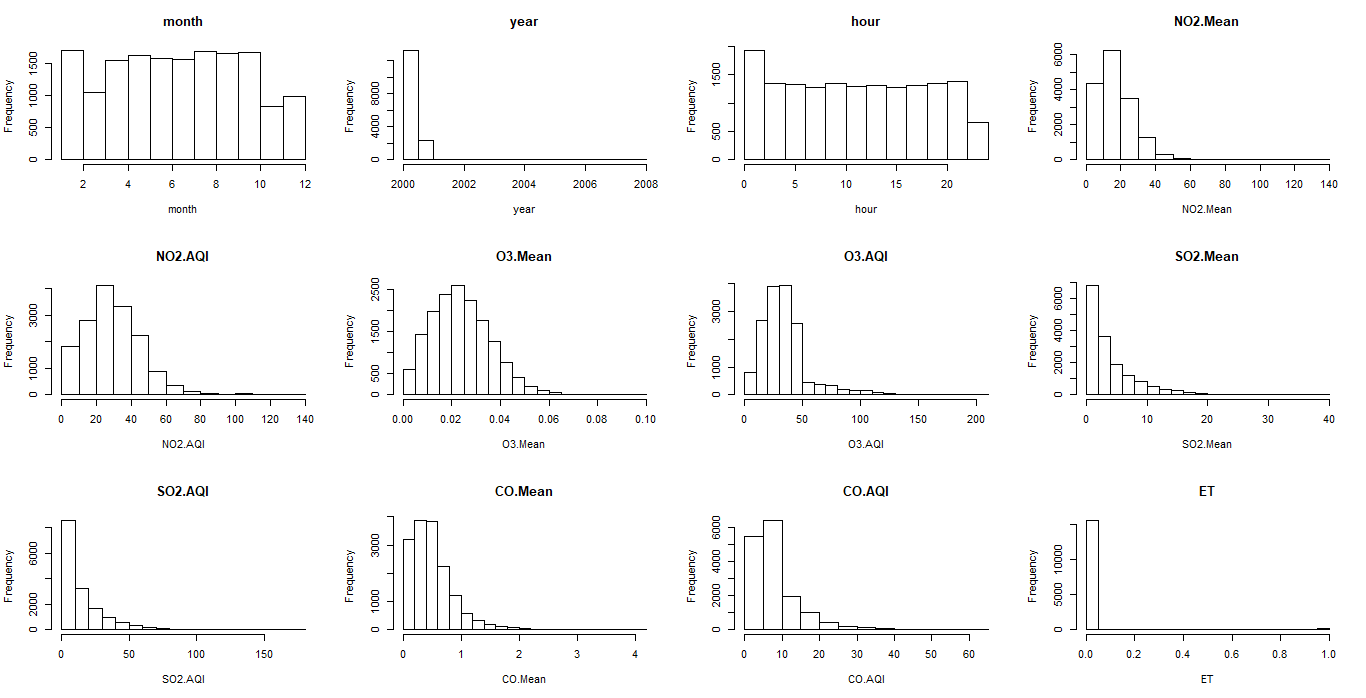
}

}

> par(mfrow=c(3,4))

> plotHistogram(UFOdata)

R Output:



1. Correlation: We will check the linear dependence between the response variable and the covariates with the help of Correlation.

R Code:

> UFOnumeric <- unlist(lapply(UFOdata, is.numeric))

> head(UFOnumeric)

> UFOnum <- UFOdata[ , UFOnumeric]

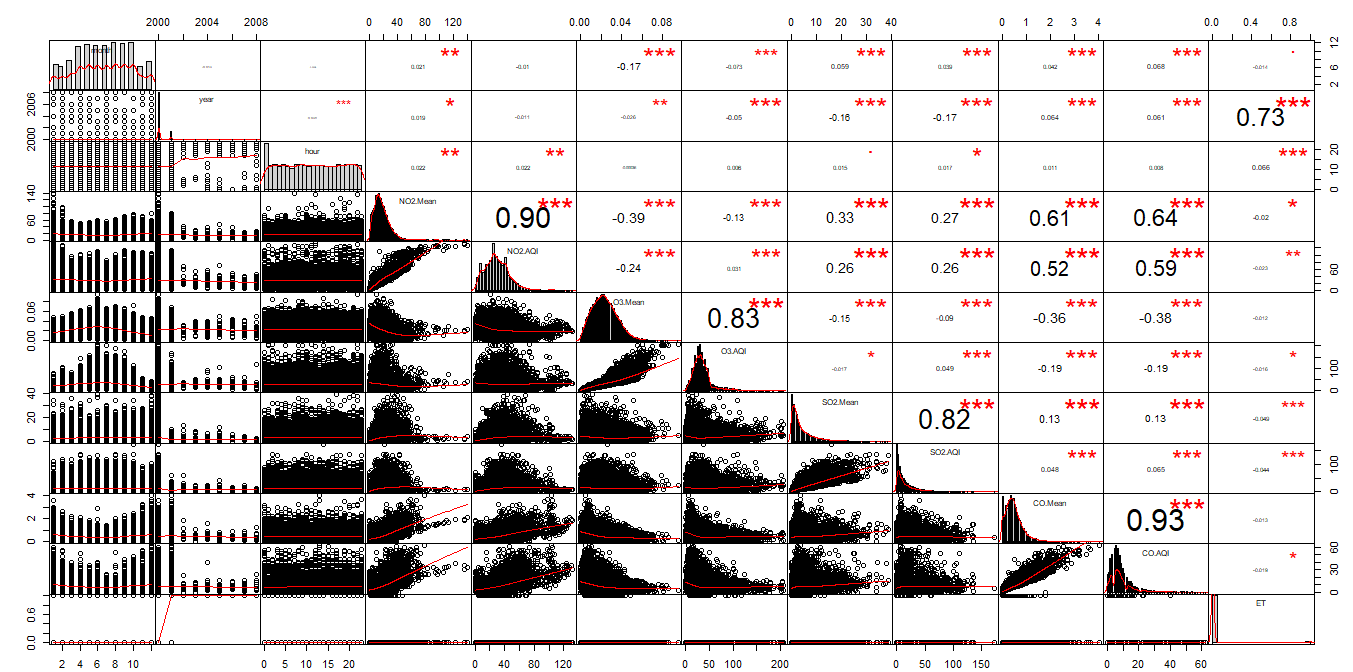
> head(UFOnum)

> install.packages("PerformanceAnalytics")

> library(PerformanceAnalytics)

> chart.Correlation(UFOnum,histogram = TRUE, pch=19)

R Output:



1. Barplot:

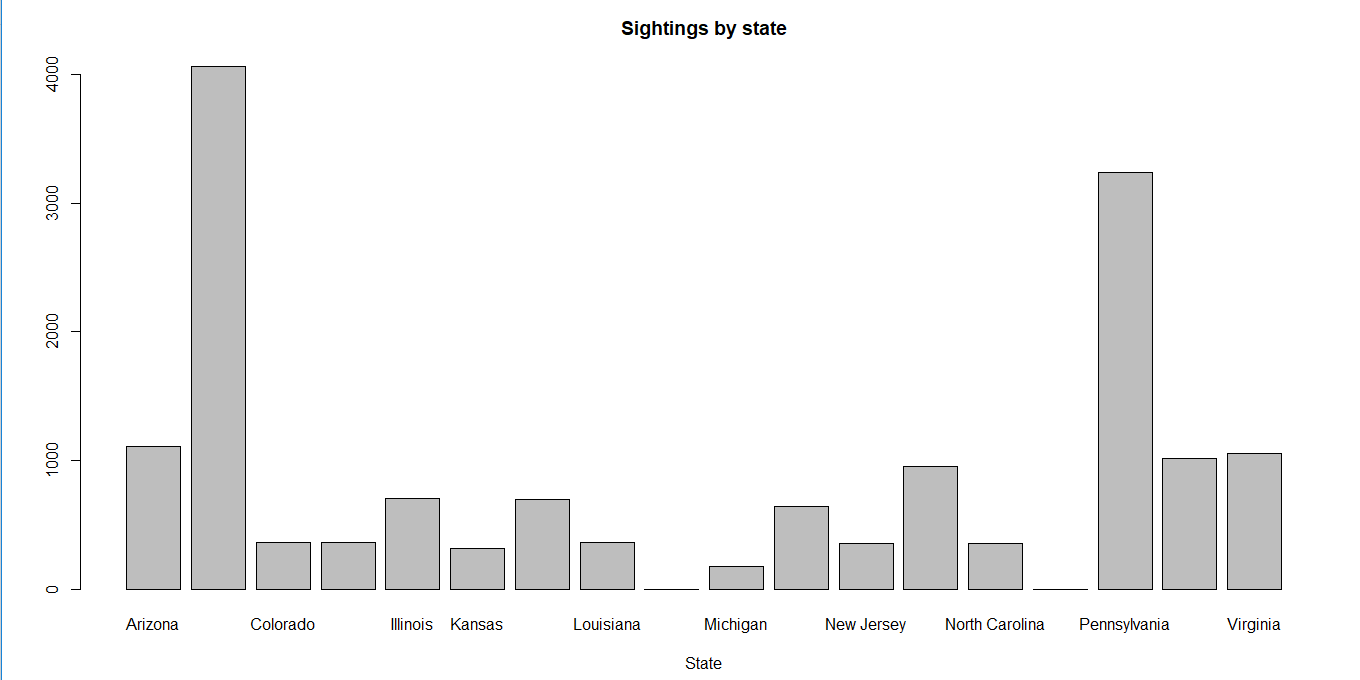
R Code:

> barCount <- table(UFOdata$state)

> par(mfrow=c(1,1))

> barplot(barCount, main = "Sightings by state", xlab="State")

R Output:



We can see the number of entries for UFO Sightings by state here.

## Dealing with categorical variables

Now we will create new variables to categorize our “State” column with 18 different levels. To deal with this, we created different columns for each state (level). By doing this we will include all the regions into our multiple linear regression model to test the significance of the covariates in model building and adequacy checking.

> UFOdata$Arizona =ifelse(UFOdata$state=="Arizona", 1,0)

> UFOdata$California =ifelse(UFOdata$state=="California", 1,0)

> UFOdata$Colorado =ifelse(UFOdata$state=="Colorado", 1,0)

> UFOdata$Florida =ifelse(UFOdata$state=="Florida", 1,0)

> UFOdata$Illinois =ifelse(UFOdata$state=="Illinois", 1,0)

> UFOdata$Kansas =ifelse(UFOdata$state=="Kansas", 1,0)

> UFOdata$Kentucky =ifelse(UFOdata$state=="Kentucky", 1,0)

> UFOdata$Louisiana =ifelse(UFOdata$state=="Louisiana", 1,0)

> UFOdata$Massachusetts =ifelse(UFOdata$state=="Massachusetts", 1,0)

> UFOdata$Michigan =ifelse(UFOdata$state=="Michigan", 1,0)

> UFOdata$Missouri =ifelse(UFOdata$state=="Missouri", 1,0)

> UFOdata$NewJersey =ifelse(UFOdata$state=="New Jersey", 1,0)

> UFOdata$NewYork =ifelse(UFOdata$state=="New York", 1,0)

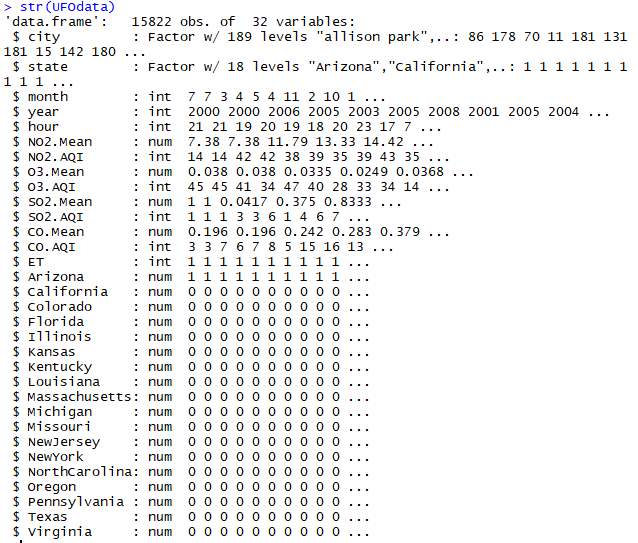
> UFOdata$NorthCarolina =ifelse(UFOdata$state=="North Carolina", 1,0)

> UFOdata$Oregon =ifelse(UFOdata$state=="Orlando", 1,0)

> UFOdata$Pennsylvania =ifelse(UFOdata$state=="Pennsylvania", 1,0)

> UFOdata$Texas =ifelse(UFOdata$state=="Texas", 1,0)

> UFOdata$Virginia =ifelse(UFOdata$state=="Virginia", 1,0)



Each region has been created successfully with an assigned value of 0 or 1.

## Split data into training and test data set

We will then split the dataset into 80:20 (train: test) data ratio. The training set will contain a known output and the model learns on this data in order to be generalized to other data later. We have the test dataset (or subset) in order to test our model’s prediction on this subset.

R Code:

> library(dyplr)

> df = subset(UFOdata, select = -c(city,state))

> trainData <- sample\_frac(df, 0.8)

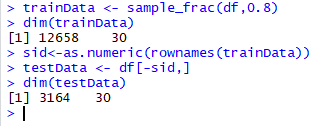
> dim(trainData)

> sid <- as.numeric(rownames(trainData))

> testData <- df[-sid,]

> dim(testData)

R Output:



## Model Specification

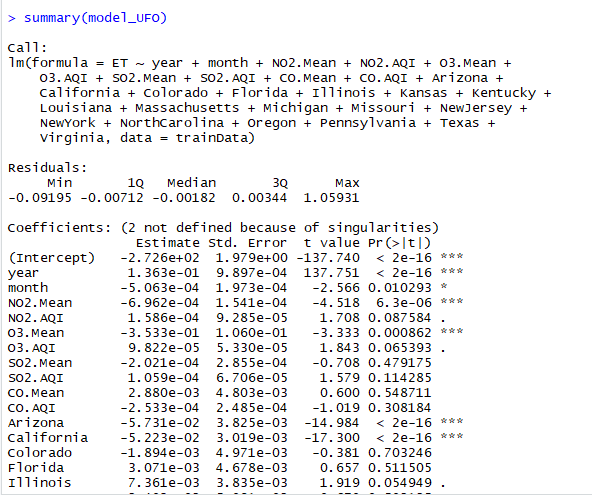
### Multiple Linear Regression – Model 1

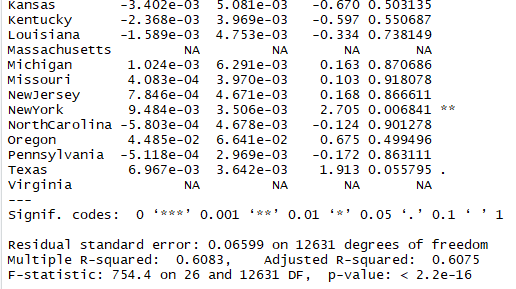
We will start with multiple linear regression since we have more than one covariate against our one response variable.

R Code:

> model\_UFO <- lm(ET ~ year+month+NO2.Mean+NO2.AQI+O3.Mean+O3.AQI+SO2.Mean+SO2.AQI+CO.Mean+CO.AQI+Arizona+California+Colorado+Florida+Illinois+Kansas+Kentucky+Louisiana+Massachusetts+Michigan+Missouri+NewJersey+NewYork+NorthCarolina+Oregon+Pennsylvania+Texas+Virginia, data = trainData)

> summary(model\_UFO)





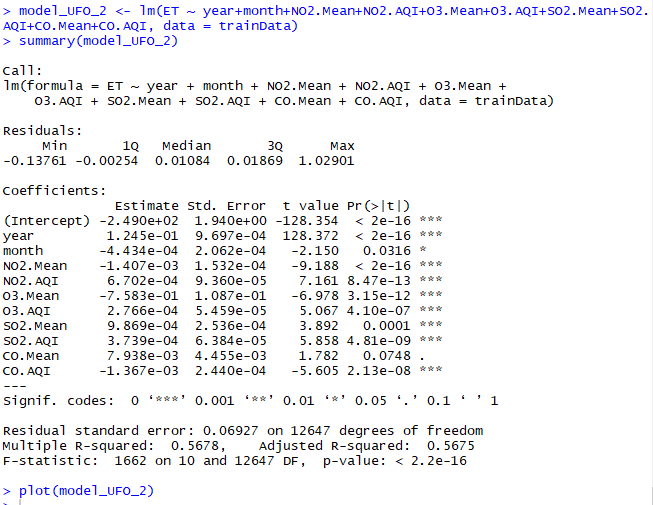
### Multiple Linear Regression – Model 2

Now, we are removing all the regions from our covariate set in order to check whether our response is dependent largely only upon the Air Quality:

R Code:

> model\_UFO <- lm(ET ~ year+month+NO2.Mean+NO2.AQI+O3.Mean+O3.AQI+SO2.Mean+SO2.AQI+CO.Mean+CO.AQI, data = trainData)

> summary(model\_UFO)



With these 2 multiple linear regression models, we will now conduct parameter estimation to get a parsimonious model.

## Parameter Estimation

### Parameter Estimation – Model 1

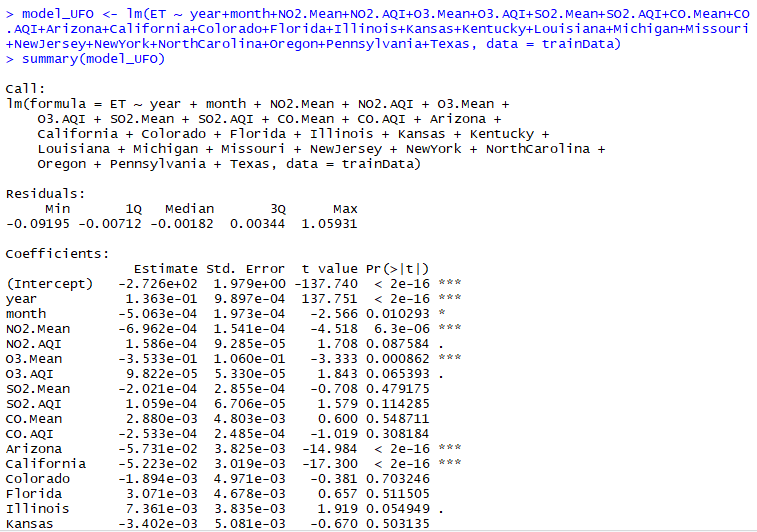
We will first remove regions Virginia and Massachusetts from our model since they have no significance.

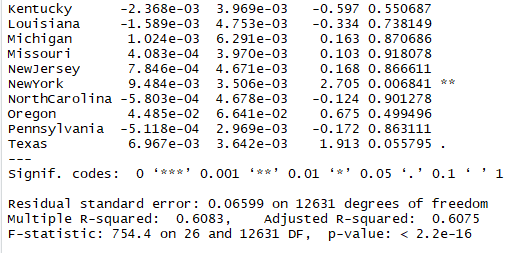
R Code:

> model\_UFO <- lm(ET ~ year+month+NO2.Mean+NO2.AQI+O3.Mean+O3.AQI+SO2.Mean+SO2.AQI+CO.Mean+CO.AQI+Arizona+California+Colorado+Florida+Illinois+Kansas+Kentucky+Louisiana+Michigan+Missouri+NewJersey+NewYork+NorthCarolina+Oregon+Pennsylvania+Texas, data = trainData)

> summary(model\_UFO)

R Output:





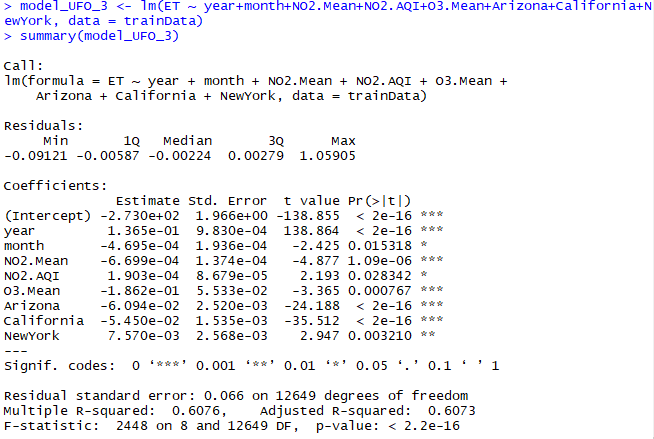
We keep iterating this process to remove covariates which has no significance in changing or affecting the p-value i.e. with p-value>0.05:

R Code:

> model\_UFO\_3<- lm(ET ~ year+month+NO2.Mean+NO2.AQI+O3.Mean+Arizona+California+NewYork, data = trainData)

> summary(model\_UFO\_3)

R Output:



But if we choose this model then we believe the model will be a biased one since the regions Arizona, California and New York have a larger percentage of observations than the other regions.

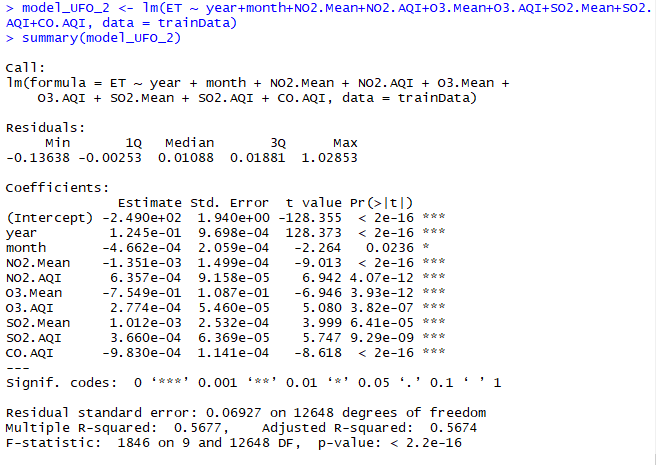
### Parameter Estimation – Model 2

We will create a new model with all the covariates, minus the regions. We will then first remove the covariate CO.Mean, since it has p value>0.05:

R code:

> model\_UFO\_2<- lm(ET ~ year+month+NO2.Mean+NO2.AQI+O3.Mean+O3.AQI+SO2.Mean+SO2.AQI+CO.AQI, data = trainData)

> summary(model\_UFO\_2)



Till now, model 2 is the best multiple linear regression model for the training data. We are choosing model 2 which is without regions since our main objective is to find UFO sightings based on air quality. Also, the model with regions Arizona, California and New York – are biased since they have a larger percentage of observations than the other regions. Hence, we will move forward and conduct model adequacy checks on model 2.

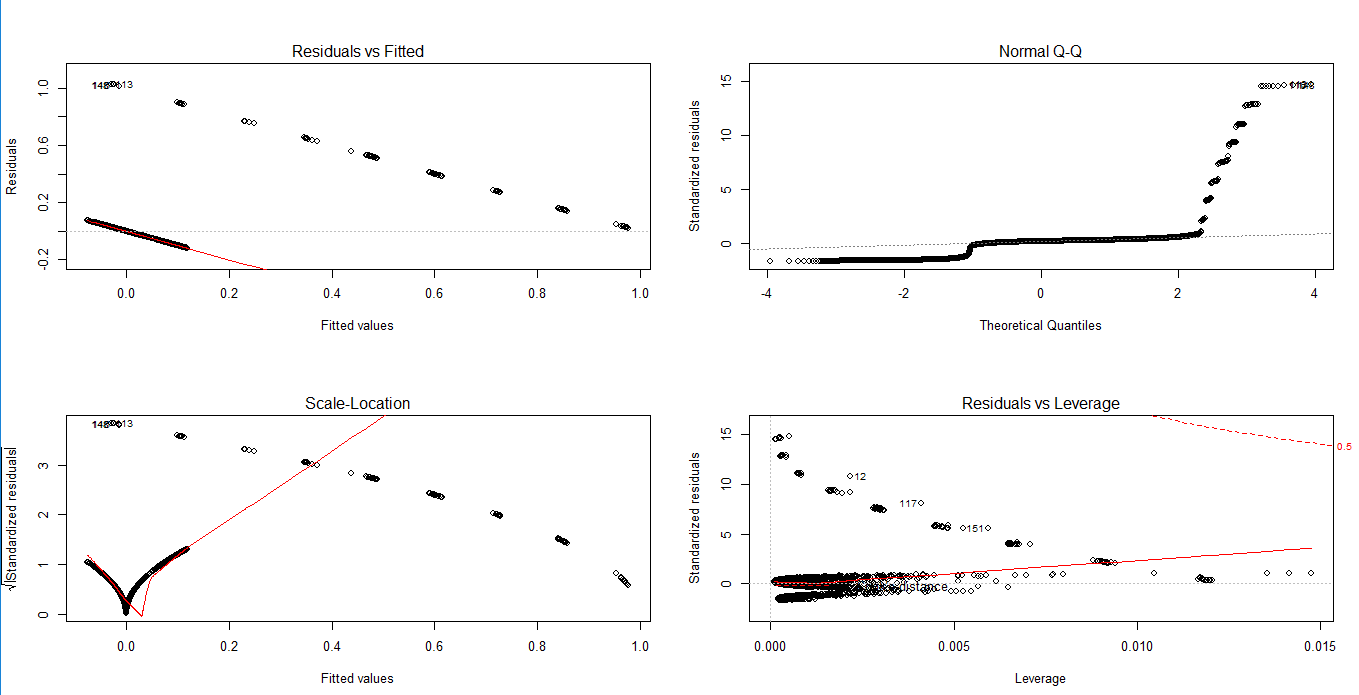
## Model Adequacy Checking

Model adequacy checking is essentially to make sure that the **LINE** assumptions are satisfied, therefore, model adequacy checking will check to see if the model fits the data well.

To do so, we did residual plots and residual analysis.

### Residual Plots and Residual Analysis

We plotted the scatter plots of residuals against fitted values to check for Equal variance and Linearity. We also plotted the QQ-plot of residuals to check for the normality of errors.



Analyzing the scatter plot of residual against the fitted values, we can see the points are not evenly distributed around zero with constant variance across the x-axis.

Analyzing QQ-plot above, we can see it is a heavy-tailed positively skewed distribution, which means that the residuals are not normally distributed. So, we should go for transformation on y or both x and y to resolve this issue.

### Transformation

We tried to conduct a Box-Cox transformation however, due to response variable being negative in the model, we could not apply and conduct the transformation. So, we decided to go ahead with the multicollinearity check.

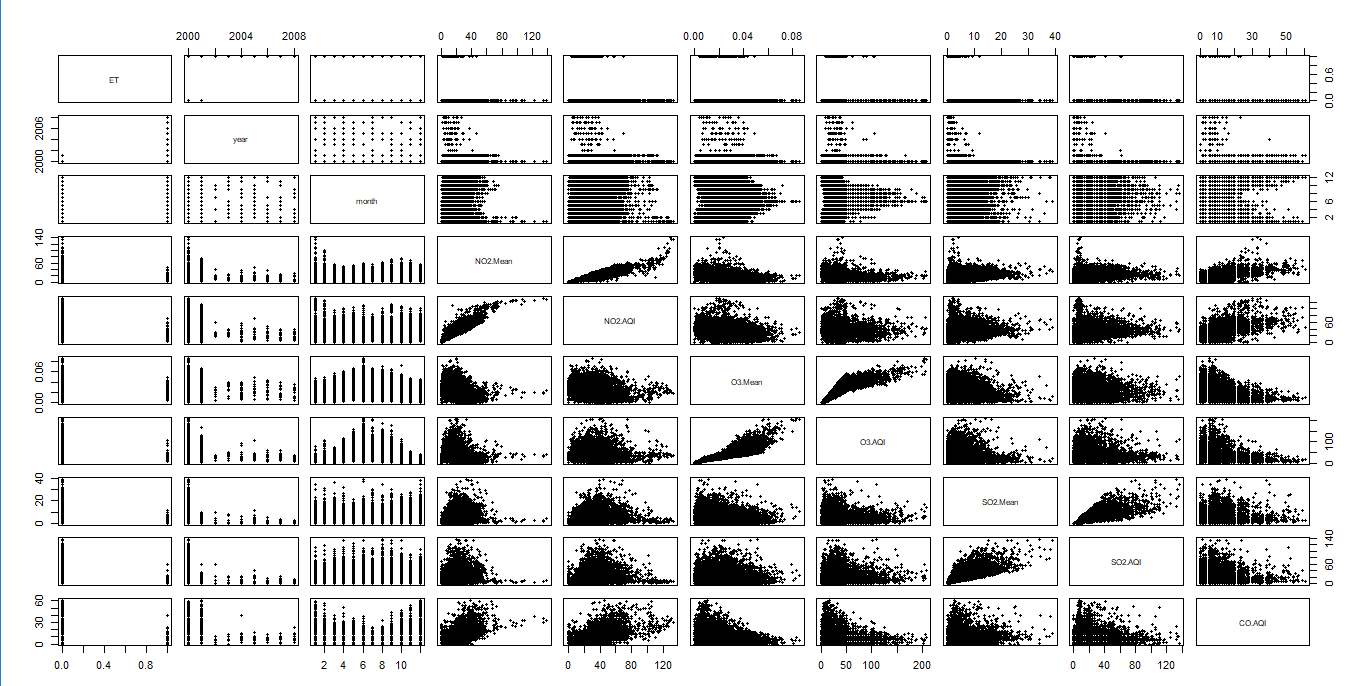
### Multicollinearity Check

In model adequacy checking, we will check for multicollinearity in order to understand the linear dependence amongst the covariates. We will plot each covariate with response variable and create a correlation plot as well.

R Code:

> plot(ET~year+month+NO2.Mean+NO2.AQI+O3.Mean+O3.AQI+SO2.Mean+SO2.AQI+CO.AQI, data=trainData, pch=20)

R output:



Correlation Plot

R Code:

> ufo <- trainData[c(1:9,11)]

> head(ufo)

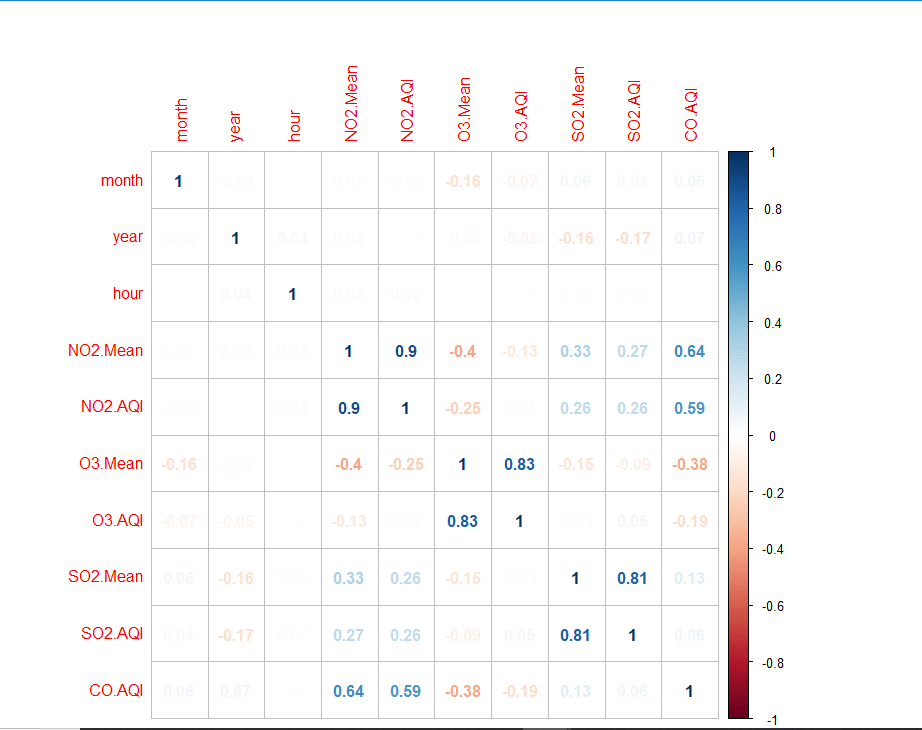
> UFOcor <- cor(ufo)

> corrplot(UFOcor, method = "number")

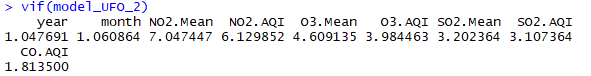
> par(mfrow=c(1,1))

> corrplot(UFOcor, method = "number")

R Output:



As we can see from these plots, there is slight linear relationship between NO2.Mean and NO2.AQI, O3.Mean and O3.AQI, SO2.Mean and SO2.AQI. Hence, we will use VIF to calculate multicollinearity for each covariate:



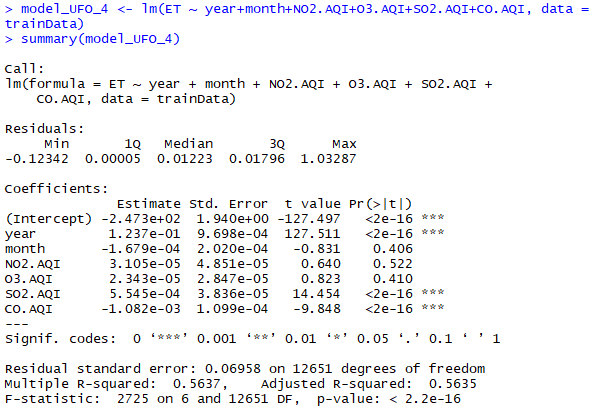
Based on multicollinearity result, removing one covariate linked with each other, removing means of NO2, O3 and SO2:

R Code:

> model\_UFO\_4 <- lm(ET ~ year+month+NO2.AQI+O3.AQI+SO2.AQI+CO.AQI, data=trainData)

> summary(model\_UFO\_4)

R Output:



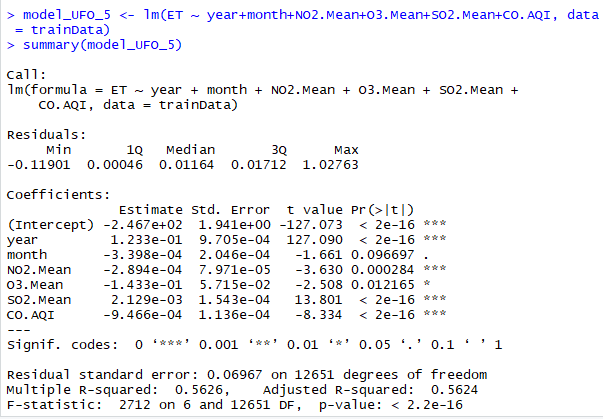
Let us now check what happens if we remove the AQIs instead of the Means:

R Code:

> model\_UFO\_5 <- lm(ET ~ year+month+NO2.Mean+O3.Mean+SO2.Mean+CO.AQI, data=trainData)

> summary(model\_UFO\_5)

R Output:



This is a better model, but we still need to remodel it by removing month with p-value>0.05 and the following is the final model:

> model\_UFO\_6 <- lm(ET ~ year+NO2.Mean+O3.Mean+SO2.Mean+CO.AQI, data=trainData)

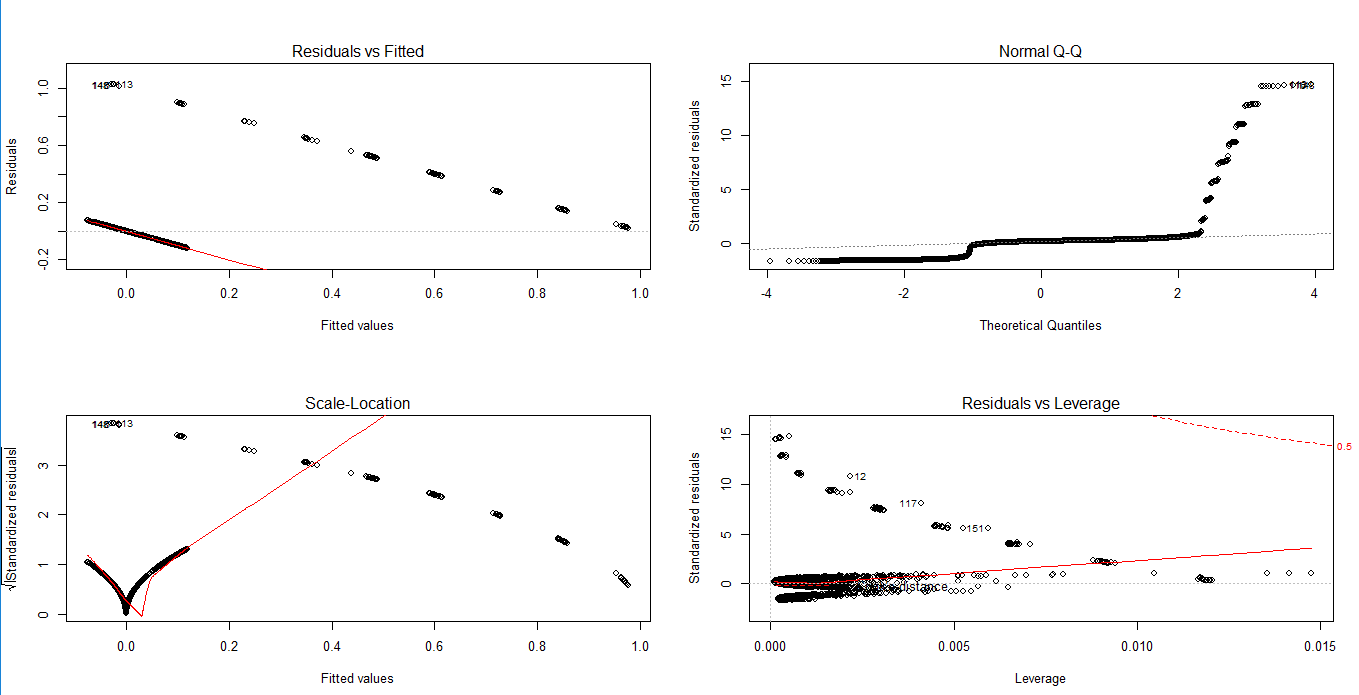
Ideally, there should be no multicollinearity between these covariates now so we will check multicollinearity between these covariates:



Our VIF values for the covariates are close to 1 hence there exists no multicollinearity between these covariates and this is our best model now. We will plot the graph for this model:

R Code:

> plot(model\_UFO\_6)



We are not getting a linear regression model after removing all possible covariates or adding the best covariates and fitting the model with the covariates with p value smaller than 0.05. So, we will check the validity of our model on our train data.

## Model Validity

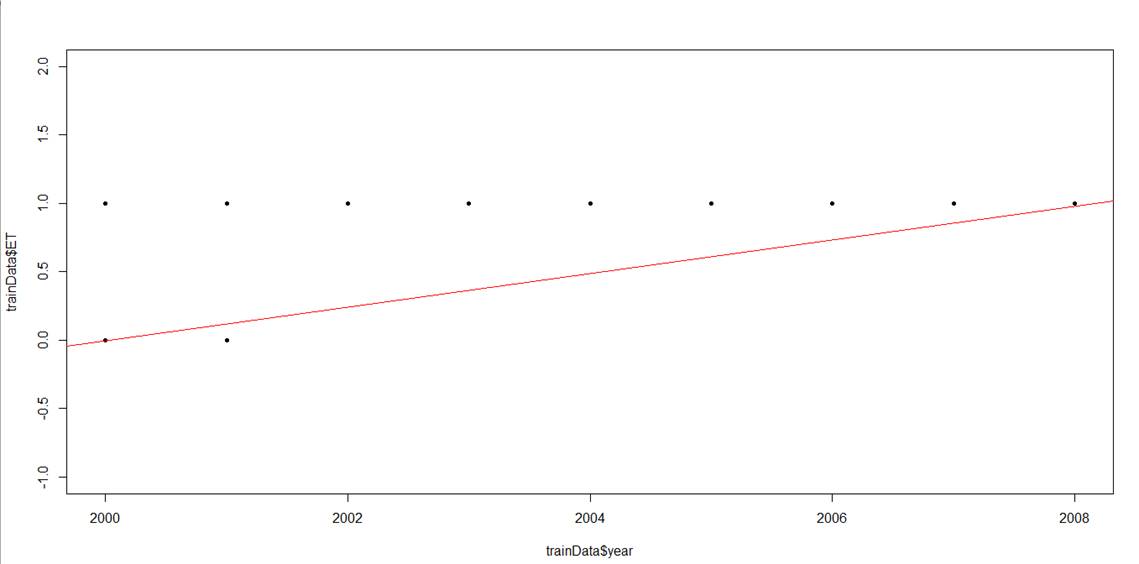
We fit the multiple linear regression to the training data set.

R Code:

> plot(trainData$year, trainData$ET, pch=20, ylim = c(-1,2))

> abline(model\_UFO\_6, col="red")

R output:



Obviously, linear regression does not make sense in this situation because some predictions are below 0 or above 1. Hence, this multiple linear regression does not make sense for our data set.

**Our research led us to find that if we have a binary response variable along with multiple covariates then we need to switch to Logistic Linear Regression to find our best fitted model.**

## Logistic Regression

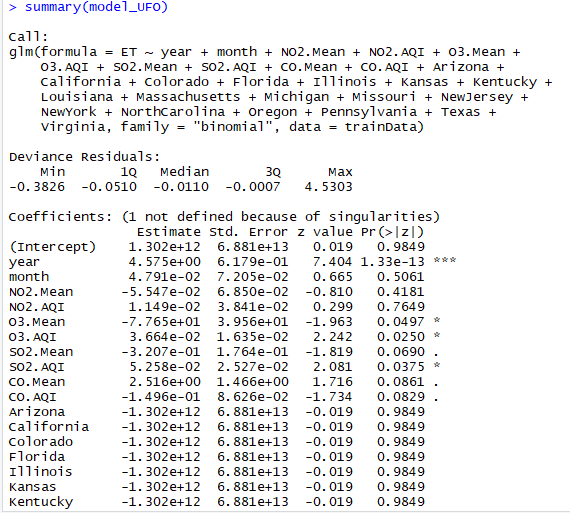
### Model Specification

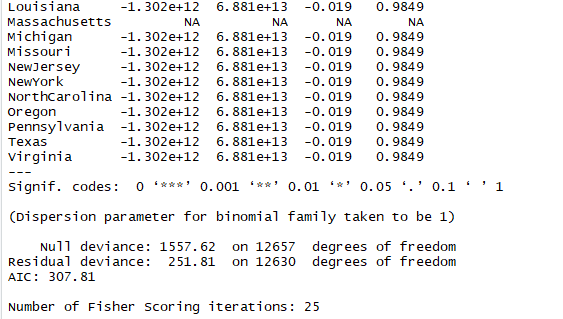
In [statistics](https://en.wikipedia.org/wiki/Statistics), the logistic model is a widely used [statistical model](https://en.wikipedia.org/wiki/Statistical_model) which uses a [logistic function](https://en.wikipedia.org/wiki/Logistic_function) to model a [binary](https://en.wikipedia.org/wiki/Binary_variable) [dependent (response) variable](https://en.wikipedia.org/wiki/Dependent_variable) against one or multiple covariates.

R Code:

> model\_UFO <- glm(ET ~ year+month+NO2.Mean+NO2.AQI+O3.Mean+O3.AQI+SO2.Mean+SO2.AQI+CO.Mean+CO.AQI+Arizona+California+Colorado+Florida+Illinois+Kansas+Kentucky+Louisiana+Massachusetts+Michigan+Missouri+NewJersey+NewYork+NorthCarolina+Oregon+Pennsylvania+Texas+Virginia, data = trainData, family = "binomial")

> summary(model\_UFO)





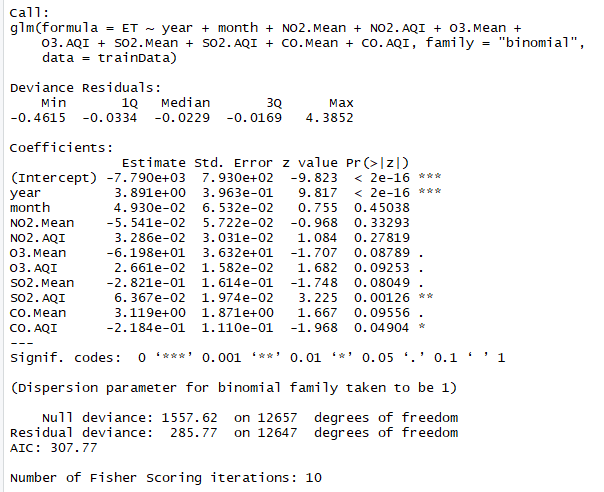
### Parameter Estimation

Like our model building approach for multiple linear regression, we will remove all the regions from the covariates since p-values for all of them is greater than 0.05:

R Code:

> model\_UFO\_2 <- glm(ET ~ year+month+NO2.Mean+NO2.AQI+O3.Mean+O3.AQI+SO2.Mean+SO2.AQI+CO.Mean+CO.AQI, data = trainData, family = "binomial")

> summary(model\_UFO\_2)



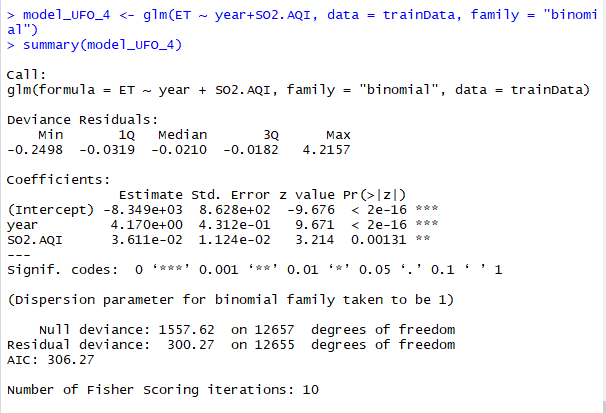
We kept on removing the covariates one after the other with p values greater than 0.05. Below is the best model which we could get after fitting in the logistic regression:

R Code:

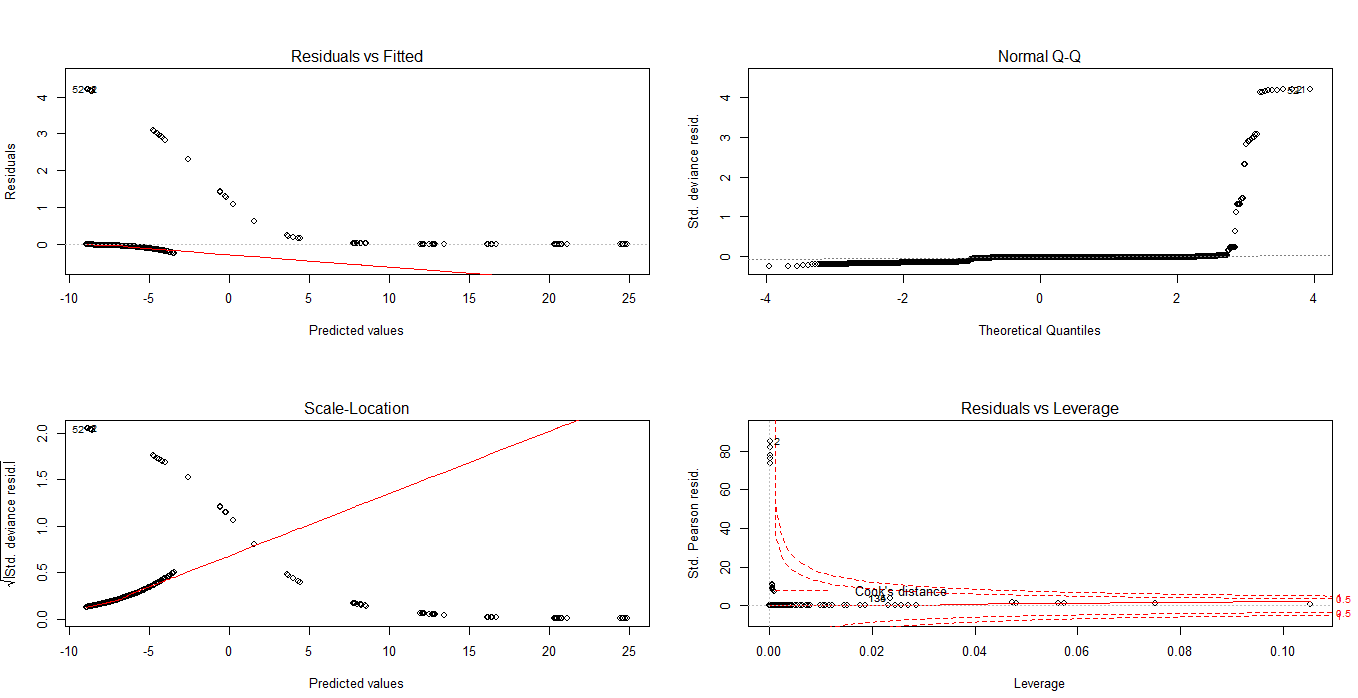
> model\_UFO\_4 <- glm(ET ~ year+SO2.AQI, data = trainData, family = "binomial")

> summary(model\_UFO\_4)

R output:



> plot(model\_UFO\_4)



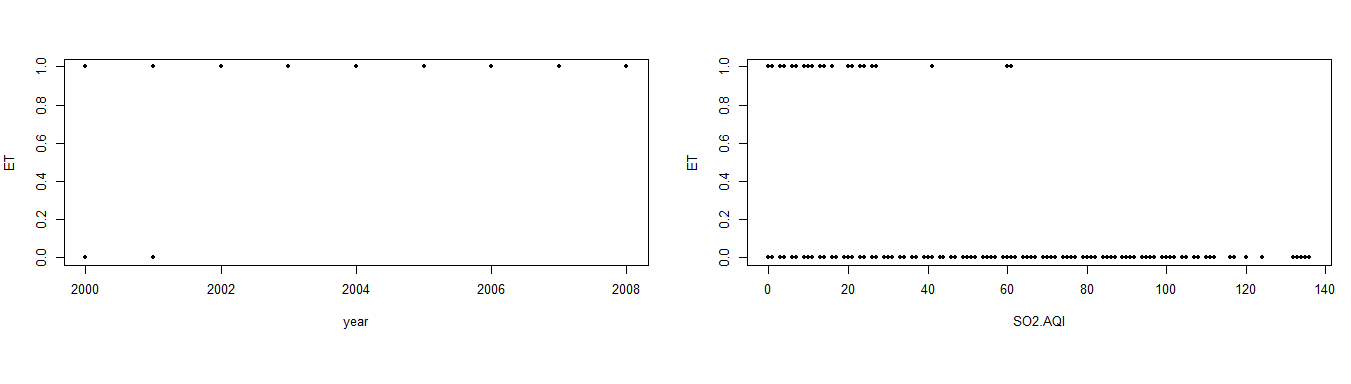
## Model Adequacy Checking

We will check for multicollinearity between the 2 covariates. We will plot each covariate with response variable and create a correlation plot as well.

R code:

plot(ET~year+ SO2.Mean, data=trainData, pch=20)

R Output:



Correlation Plot:

R Code:

> ufo <- trainData[c(2,9)]

> head(ufo)

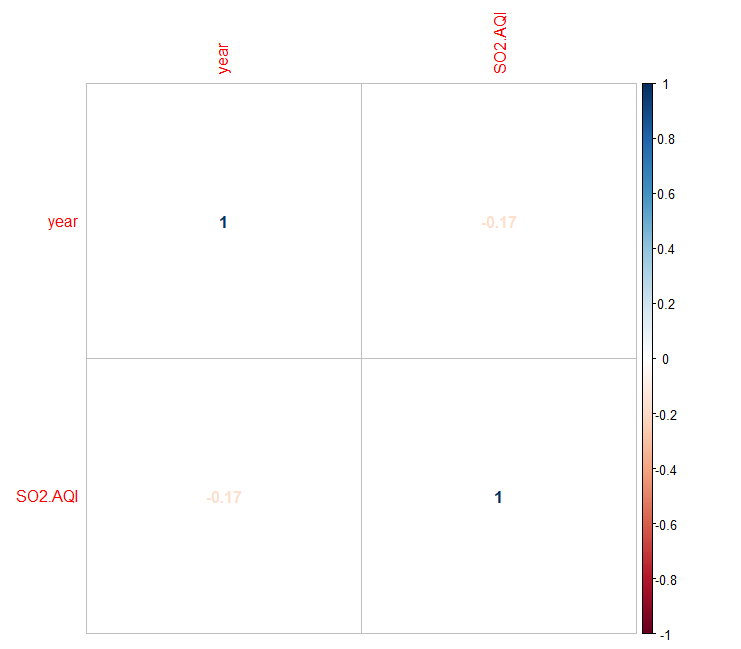
> UFOcor <- cor(ufo)

> corrplot(UFOcor, method = "number")

> par(mfrow=c(1,1))

> corrplot(UFOcor, method = "number")

R Output:



As we can see, there is no linear relationship between the 2 covariates. We will calculate the VIF values for confirmation as well.

Using VIF:



Since the VIF values are close to 1, we can confirm there is no linear relationship between the covariates

## Plot Logistic regression

We will now plot the logistic regression for each covariate against one response variable individually:

For Year:

R Code:

> plot(trainData$year, trainData$ET, ylim = c(-1,2), pch=20, type = "n")

> points(trainData$year, trainData$ET, pch=20)

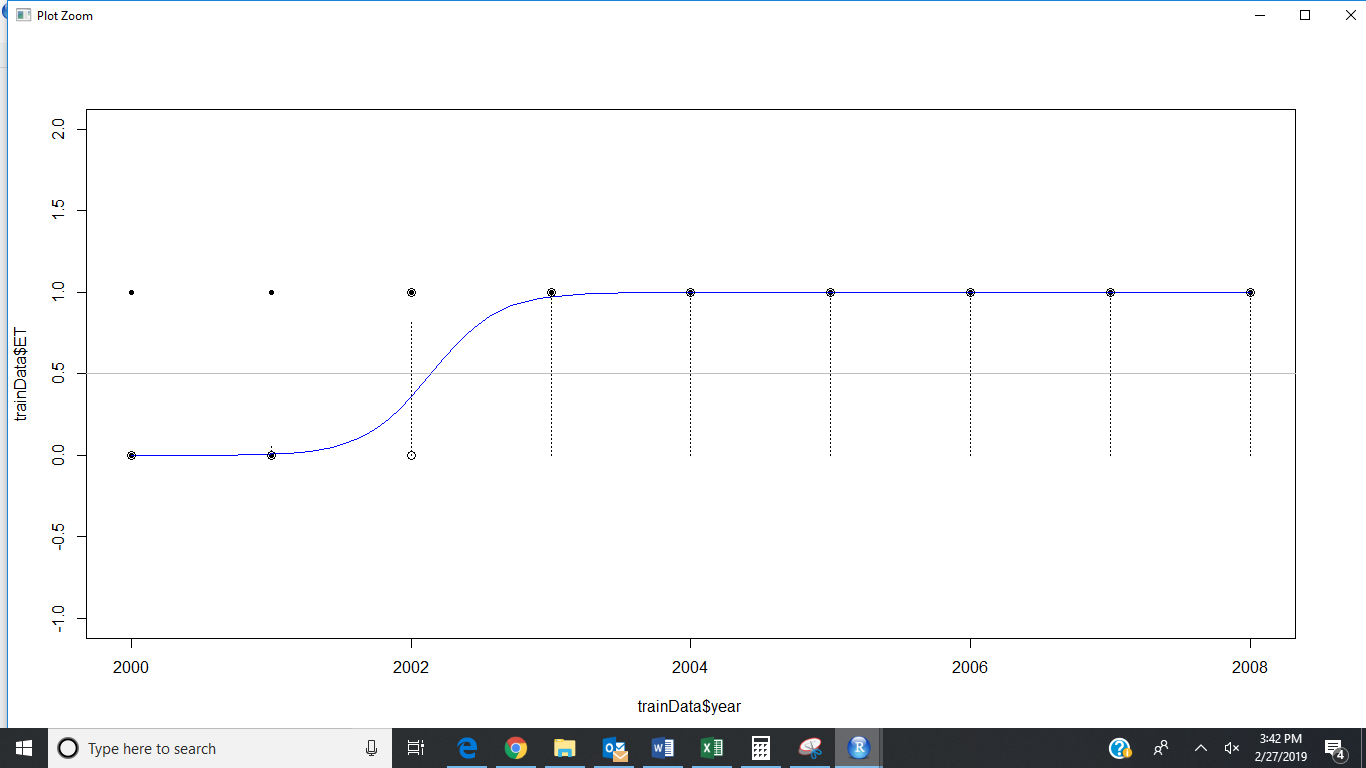
> points(trainData$year, pred\_prob, type = "h", lty=3)

> curve(1/(1+exp(-(model\_UFO\_4$coefficients[1]+model\_UFO\_4$coefficients[2]\*x))), add = TRUE, col="blue")

> abline(h=0.5, col="grey")

> points(trainData$year, pred\_value,pch=1,cex=1.3)

R Output:



From the above graph we can understand that there were more UFO sightings from 2003-2008 and that there were no UFO sightings for 2001 and 2002.

For SO2.AQI:

R Code:

> plot(trainData$SO2.AQI, trainData$ET, ylim = c(-1,2), pch=20, type = "n")

> points(trainData$SO2.AQI, trainData$ET, pch=20)

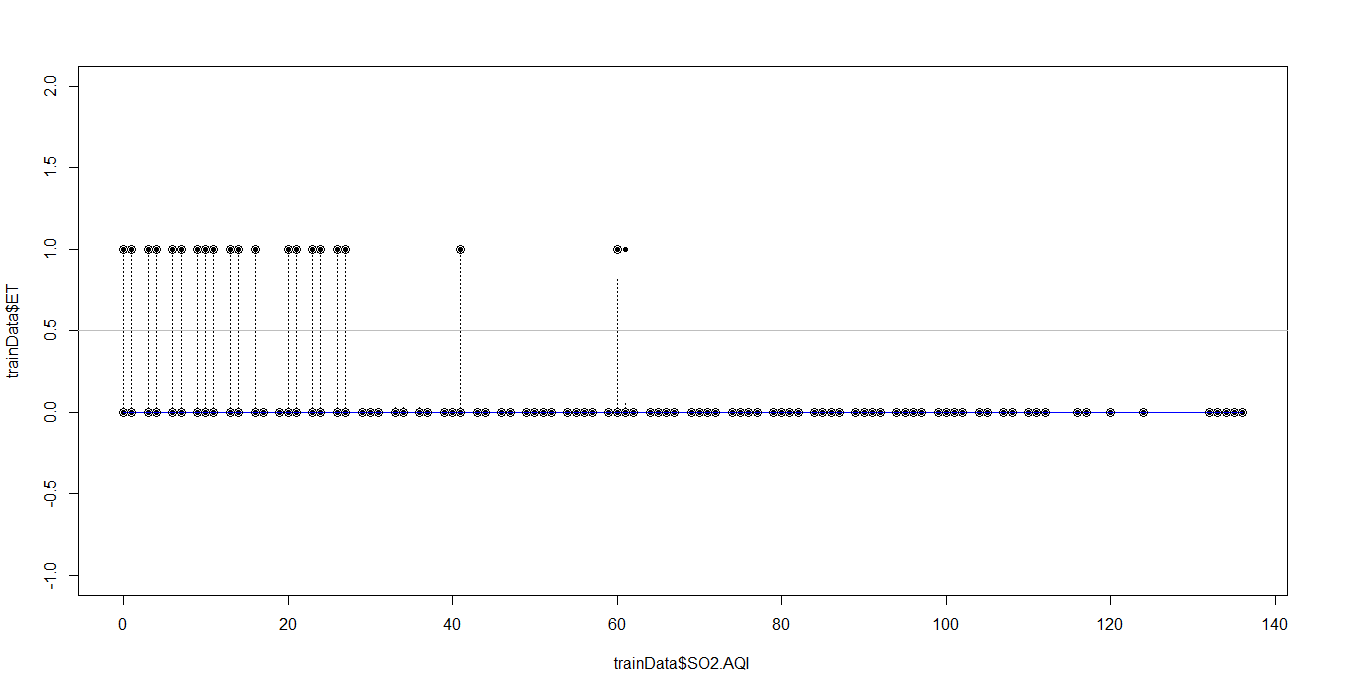
> points(trainData$SO2.AQI, pred\_prob, type = "h", lty=3)

> curve(1/(1+exp(-(model\_UFO\_4$coefficients[1]+model\_UFO\_4$coefficients[2]\*x))), add = TRUE, col="blue")

> abline(h=0.5, col="grey")

> points(trainData$SO2.AQI, pred\_value,pch=1,cex=1.3)

R Output:



As the concentration of SO2.AQI is increasing the UFO Sightings are decreasing. It is also obvious from the graph that after hitting a maximum SO2.AQI value, there are no more UFO sightings recorded. So, we decided to calculate that maximum SO2.AQI value for when a UFO sighting was observed.

R code:

> max(trainData[trainData$ET=="1",]$SO2.AQI)

R output:



The answer from the code above concurs with the graph plotted.

### Confusion matrix

We will use a confusion matrix to evaluate the prediction performance.

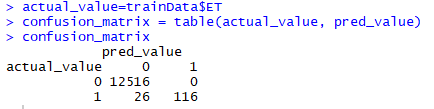
R code:

> actual\_value=trainData$ET

> confusion\_matrix = table(actual\_value, pred\_value)

> confusion\_matrix

R Output:



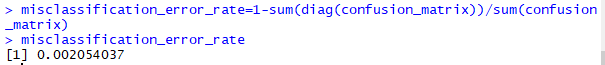
We can calculate the misclassification error rate using the confusion matrix.

R code:

> misclassification\_error\_rate=1-sum(diag(confusion\_matrix))/sum(confusion\_matrix)

> misclassification\_error\_rate

R Output:



As we can see, the misclassification error rate is low, which means our model is a good model to use for predictions on this data set.

## Model Use

We will now use the model we have created to predict UFO sightings based on the test data and then check it against the actual values.

Since we can only use 1 covariate to plot in logistic regression at a time, we will plot the test data and the predicted values against both Year and SO2.AQI.

Year:

R code:

> plot(testData$year, testData$ET, pch=20, type="n")

> points(testData$year, testData$ET, pch=20)

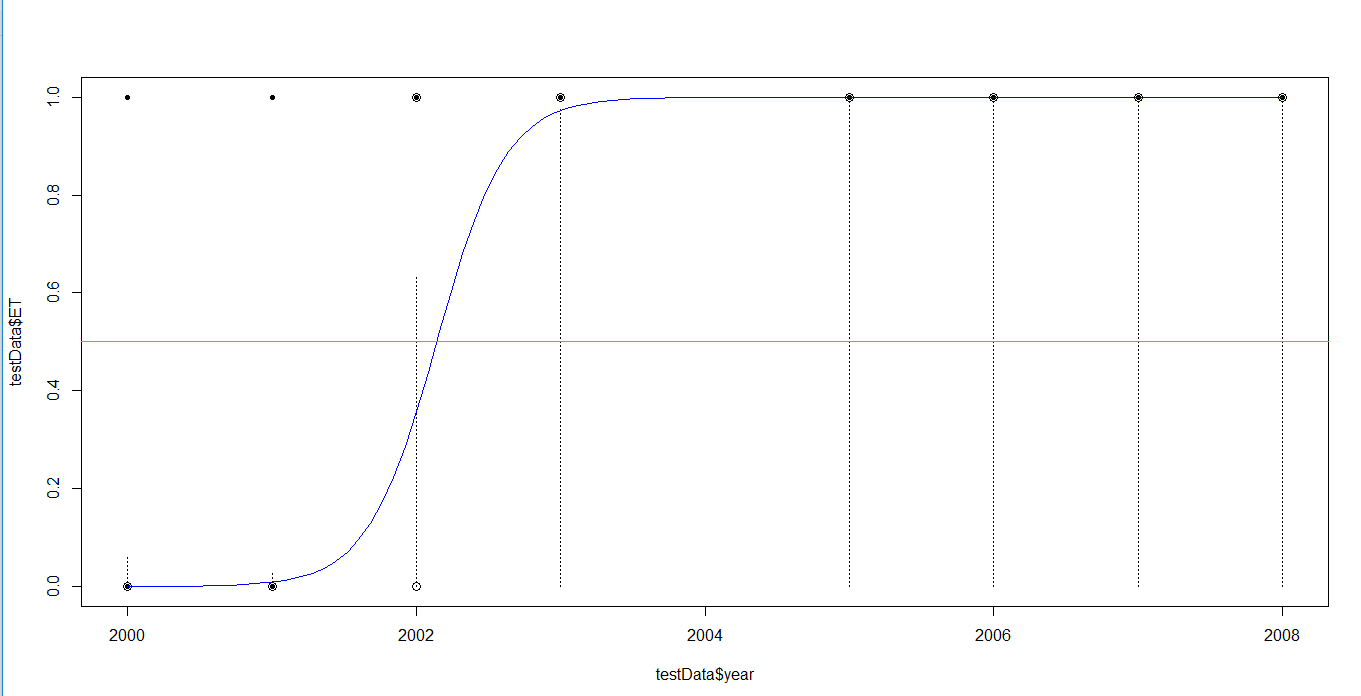
> points(testData$year, pred\_prob1, type = "h", lty=3)

> curve(1/(1+exp(-(model\_UFO\_4$coefficients[1]+model\_UFO\_4$coefficients[2]\*x))), add = TRUE, col="blue")

> abline(h=0.5, col="green")

> points(testData$year, pred\_value1,pch=1,cex=1.3)

R Output:



For SO2.AQI:

R Code:

> pred\_prob1 = predict(model\_UFO\_4, testData, type = "response")

> plot(testData$SO2.AQI, testData$ET, pch=20, type="n")

> points(testData$SO2.AQI, testData$ET, pch=20)

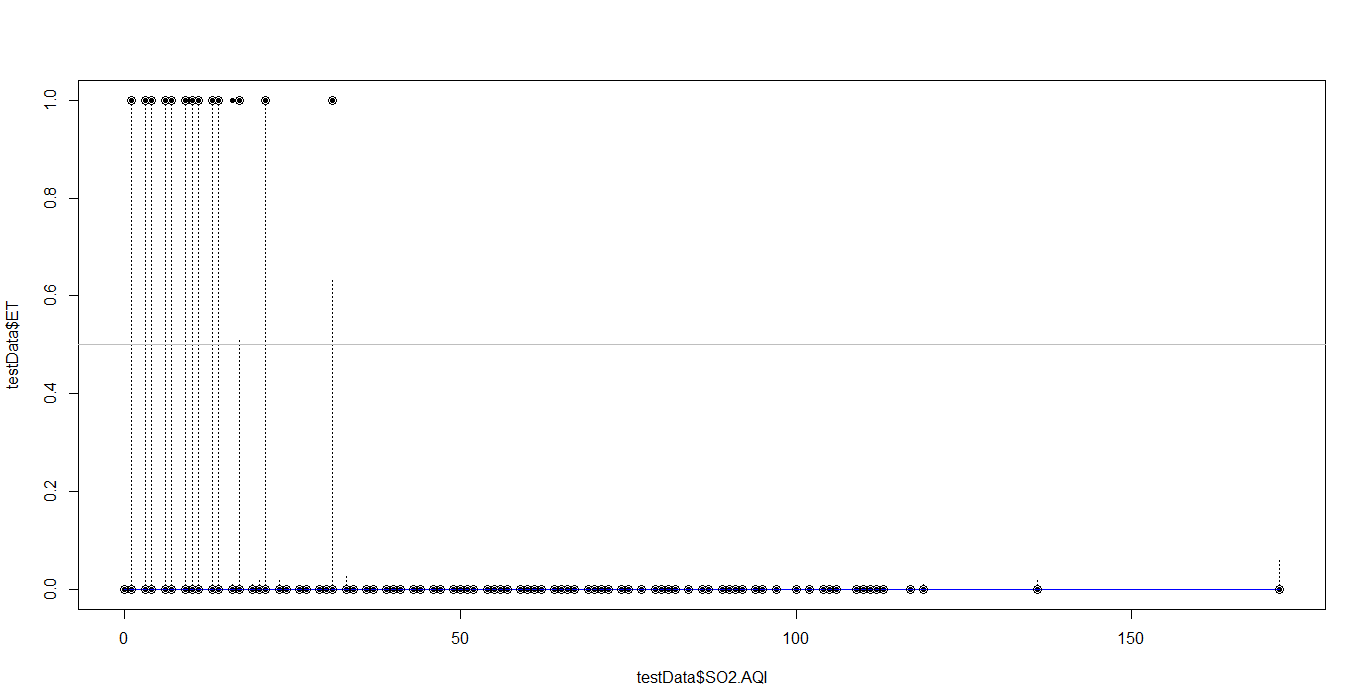
> points(testData$SO2.AQI, pred\_prob1, type = "h", lty=3)

> curve(1/(1+exp(-(model\_UFO\_4$coefficients[1]+model\_UFO\_4$coefficients[2]\*x))), add = TRUE, col="blue")

> abline(h=0.5, col="grey")

> points(testData$SO2.AQI, pred\_value1,pch=1,cex=1.3)

R Output:



### Confusion Matrix

We will use a confusion matrix to evaluate the prediction performance of our model on the test data.

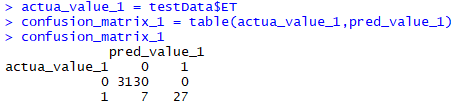
R Code:

> actua\_value\_1 = testData$ET

> confusion\_matrix\_1 = table(actua\_value\_1,pred\_value\_1)

> confusion\_matrix\_1

R Output:



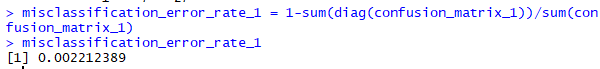
We can calculate the misclassification error rate using the confusion matrix.

R Code:

> misclassification\_error\_rate\_1 = 1-sum(diag(confusion\_matrix\_1))/sum(confusion\_matrix\_1)

> misclassification\_error\_rate\_1

R Output:



As we can see, the misclassification error rate is low, which means our model is a good model to use for predictions on this data set.

## Conclusion

Our findings with a never-before analyzed data-set taught us that a Logistic Regression is the best option when your response variable is binary.

From among the 24 covariates that we began our analysis with, only the SO2 Air Quality Index and Year were found to influence the number of UFO Sightings.