TITLE: AIR QUALITY ANALYSIS IN TAMIL NADU

PHASE 3: DEVELOPMENT PART ONE

Importing Libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import Imputer
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['figure.figsize'] = (10, 7)
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_log_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.feature_selection import RFE
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
import statsmodels.formula.api as sm
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from statsmodels.regression.linear_model import OLS
from statsmodels.tools import add_constant
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn import metrics
from statsmodels.stats.outliers_influence import
variance_inflation_factor
import warnings; warnings.simplefilter('ignore')
```

*NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

* Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008

*Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

*Scikit-Learn, also known as sklearn is a python library to implement machine learning models and statistical modelling. Through scikit-learn, we can implement various machine learning models for regression, classification, clustering, and statistical tools for analyzing these models.

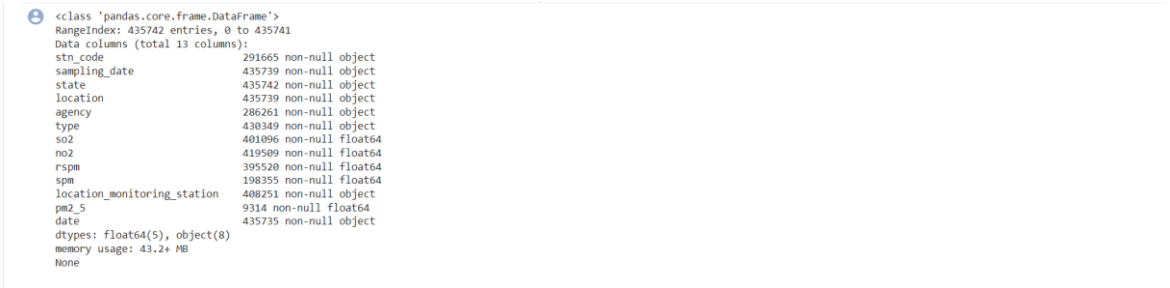
Dataset Summary

Statistcal analysis of given dataset

```
dataset=pd.read_csv('../input/data.csv',encoding="ISO-8859-1")
dataset.describe()
```

	so2	no2	rspm	spm	pm2_5
count	401096.000000	419509.000000	395520.000000	198355.000000	9314.000000
mean	10.829414	25.809623	108.832784	220.783480	40.791467
std	11.177187	18.503086	74.872430	151.395457	30.832525
min	0.000000	0.000000	0.000000	0.000000	3.000000
25%	5.000000	14.000000	56.000000	111.000000	24.000000
50%	8.000000	22.000000	90.000000	187.000000	32.000000
75%	13.700000	32.200000	142.000000	296.000000	46.000000
max	909.000000	876.000000	6307.033333	3380.000000	504.000000

Information about each column and about null values for each column



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 435742 entries, 0 to 435741
Data columns (total 13 columns):
stn_code                        291665 non-null object
sampling_date                   435739 non-null object
state                           435742 non-null object
location                        435739 non-null object
agency                          286261 non-null object
type                            430349 non-null object
so2                             401096 non-null float64
no2                             419509 non-null float64
rspm                            395520 non-null float64
spm                             198355 non-null float64
location_monitoring_station     408251 non-null object
pm2_5                           9314 non-null float64
date                            435735 non-null object
dtypes: float64(5), object(8)
memory usage: 43.2+ MB
None
```

The following table shows the first five rows of the given dataset, thereby giving us insight about what sort of dataset it is. And what are the attributes included in the dataset.

```
[ ] dataset.head()
```

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm	spm	location_monitoring_station	pm2_5	date
0	150	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	NaN	NaN	NaN	NaN	1990-02-01
1	151	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	NaN	NaN	NaN	NaN	1990-02-01
2	152	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	NaN	NaN	NaN	NaN	1990-02-01
3	150	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	NaN	NaN	NaN	NaN	1990-03-01
4	151	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	NaN	NaN	NaN	NaN	1990-03-01

Dataset Cleaning

Following tables gives information about new dataset after dropping of unneccessary columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 435742 entries, 0 to 435741
Data columns (total 9 columns):
state       435742 non-null object
location    435739 non-null object
type        430349 non-null object
so2         401096 non-null float64
no2         419509 non-null float64
rspm        395520 non-null float64
spm         198355 non-null float64
pm2_5       9314 non-null float64
date        435735 non-null object
dtypes: float64(5), object(4)
memory usage: 29.9+ MB
```

	state	location	type	so2	no2	rspm	spm	pm2_5	date
0	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	4.8	17.4	NaN	NaN	NaN	1990-02-01
1	Andhra Pradesh	Hyderabad	Industrial Area	3.1	7.0	NaN	NaN	NaN	1990-02-01
2	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.2	28.5	NaN	NaN	NaN	1990-02-01
3	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.3	14.7	NaN	NaN	NaN	1990-03-01
4	Andhra Pradesh	Hyderabad	Industrial Area	4.7	7.5	NaN	NaN	NaN	1990-03-01

Often a DataFrame will contain columns that are not useful to your analysis. Such columns should be dropped from the DataFrame to make it easier for you to focus on the remaining columns. The columns can be removed by specifying label names and corresponding axis, or by specifying index or column names directly.

*code:

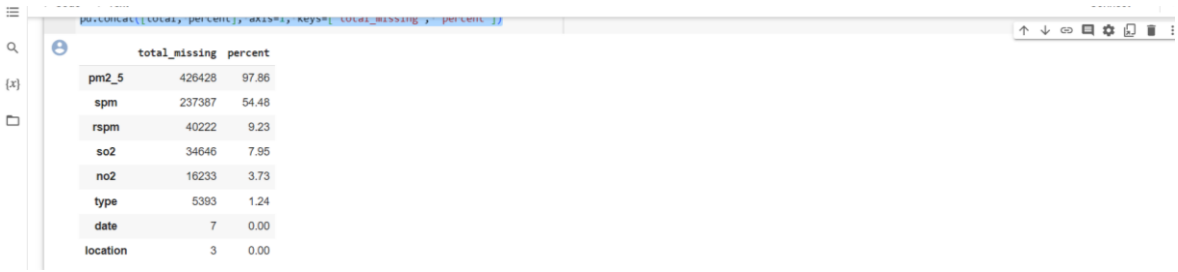
```
dataset.drop(['stn_code','agency','sampling_date','location_monitoring_
station'],axis=1,inplace=True)
dataset.info()
dataset.head()
```

FINDING THE MISSING VALUES IN THE DATASET:

If you are aiming for a job as a data scientist, you must know how to handle the problem of missing values, which is quite common in many real-life datasets. Incomplete data can bias the results of the machine learning models and/or reduce the accuracy of the model.

*CODE:

```
total = dataset.isnull().sum()[dataset.isnull().sum() !=
0].sort_values(ascending = False)
percent = pd.Series(round(total/len(dataset)*100,2))
pd.concat([total, percent], axis=1, keys=['total_missing', 'percent'])
```



	total_missing	percent
pm2_5	426428	97.86
spm	237387	54.48
rspm	40222	9.23
so2	34646	7.95
no2	16233	3.73
type	5393	1.24
date	7	0.00
location	3	0.00

REMOVING OUTLIERS:

Outliers can distort statistical analyses and skew results as they are extreme values that differ from the rest of the data. Removing outliers makes the results more robust and accurate by eliminating their influence.

CODE:

```
def remove_outlier(df_in, col_name):
    q1 = df_in[col_name].quantile(0.25)
    q3 = df_in[col_name].quantile(0.75)
```

```
    iqr = q3-q1 #Interquartile range
    fence_low  = q1-1.5*iqr
    fence_high = q3+1.5*iqr
    df_out = df_in.loc[(df_in[col_name] > fence_low) & (df_in[col_name]
< fence_high)]
    #return df_out

remove_outlier(dataset,'so2')
remove_outlier(dataset,'no2')
remove_outlier(dataset,'rspm')
remove_outlier(dataset,'spm')
```