Assignment 2

Problem Statement-

Perform the following operations using Python on the Air quality data sets

a. Data cleaning b. Data transformation c. Data integration d. Error correcting e. Data model building

Importing python libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Loading a CSV file into a dataframe

In [2]:

```
A = pd.read_csv(r"C:\Users\HP\Downloads\airquality_dataset.csv")
A.head()
```

Out[2]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
0	1	41.0	190.0	7.4	67	5	1	High
1	2	36.0	118.0	8.0	72	5	2	High
2	3	12.0	149.0	12.6	74	5	3	Low
3	4	18.0	313.0	11.5	62	5	4	NaN
4	5	NaN	NaN	14.3	56	5	5	High

In [3]:

```
A.shape
```

Out[3]:

```
(153, 8)
```

Checking for null values in each column

In [4]:

```
A.isnull().sum()
```

Out[4]:

```
Unnamed: 0     0
Ozone         37
Solar.R        7
Wind           2
Temp           0
Month          0
Day            0
Humidity       8
dtype: int64
```

A] Data Cleaning

Removing unwanted column from dataset:

In [5]:

```
df=A.drop("Unnamed: 0", axis=1)
df
```

Out[5]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
0	41.0	190.0	7.4	67	5	1	High
1	36.0	118.0	8.0	72	5	2	High
2	12.0	149.0	12.6	74	5	3	Low
3	18.0	313.0	11.5	62	5	4	NaN
4	NaN	NaN	14.3	56	5	5	High
...	...	...	...	...	...	...	...
148	30.0	193.0	6.9	70	9	26	Low
149	NaN	145.0	13.2	77	9	27	Low
150	14.0	191.0	14.3	75	9	28	High
151	18.0	131.0	8.0	76	9	29	Medium
152	20.0	223.0	11.5	68	9	30	Low

153 rows × 7 columns

Replacing numerical null values

In [6]:

```
df["Ozone"] = df["Ozone"].fillna(df["Ozone"].mean())

df["Solar.R"] = df["Solar.R"].fillna(df["Solar.R"].mean())

df["Wind"] = df["Wind"].fillna(df["Wind"].mean())
```

Replacing categorical null values

In [7]:

```
df["Humidity"] = df["Humidity"].fillna(df["Humidity"].mode()[0])
df
```

Out[7]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
0	41.00000	190.000000	7.4	67	5	1	High
1	36.00000	118.000000	8.0	72	5	2	High
2	12.00000	149.000000	12.6	74	5	3	Low
3	18.00000	313.000000	11.5	62	5	4	High
4	42.12931	185.931507	14.3	56	5	5	High
...	...	...	...	...	...	...	...
148	30.00000	193.000000	6.9	70	9	26	Low
149	42.12931	145.000000	13.2	77	9	27	Low
150	14.00000	191.000000	14.3	75	9	28	High
151	18.00000	131.000000	8.0	76	9	29	Medium
152	20.00000	223.000000	11.5	68	9	30	Low

153 rows × 7 columns

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
Ozone       0
Solar.R     0
Wind        0
Temp        0
Month       0
Day         0
Humidity    0
dtype: int64
```

In [9]:

```
df.dtypes
```

Out[9]:

```
Ozone       float64
Solar.R     float64
Wind        float64
Temp          int64
Month         int64
Day           int64
Humidity     object
dtype: object
```

B] Data Transformation

Using Label Encoding for "Humidity" column

In [10]:

```
from sklearn.preprocessing import LabelEncoder

label_en = LabelEncoder()

df["Humidity"] = label_en.fit_transform(df["Humidity"])

df["Humidity"].unique()
```

Out[10]:

```
array([0, 1, 2])
```

In [11]:

```
df
```

Out[11]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
0	41.00000	190.000000	7.4	67	5	1	0
1	36.00000	118.000000	8.0	72	5	2	0
2	12.00000	149.000000	12.6	74	5	3	1
3	18.00000	313.000000	11.5	62	5	4	0
4	42.12931	185.931507	14.3	56	5	5	0
...	...	...	...	...	...	...	...
148	30.00000	193.000000	6.9	70	9	26	1
149	42.12931	145.000000	13.2	77	9	27	1
150	14.00000	191.000000	14.3	75	9	28	0
151	18.00000	131.000000	8.0	76	9	29	2
152	20.00000	223.000000	11.5	68	9	30	1

153 rows × 7 columns

In [12]:

```
df.dtypes
```

Out[12]:

```
Ozone       float64
Solar.R     float64
Wind        float64
Temp          int64
Month         int64
Day           int64
Humidity      int32
dtype: object
```

C] Data Integration

Row wise subset:

In [13]:

```
#subset1
subset1=df.iloc[[3,5,6,7,23,43,12],:]
subset1
```

Out[13]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
3	18.0	313.000000	11.5	62	5	4	0
5	28.0	185.931507	14.9	66	5	6	0
6	23.0	299.000000	8.6	65	5	7	0
7	19.0	99.000000	13.8	59	5	8	1
23	32.0	92.000000	12.0	61	5	24	0
43	23.0	148.000000	8.0	82	6	13	2
12	11.0	290.000000	9.2	66	5	13	1

In [14]:

```
subset1.shape
```

Out[14]:

```
(7, 7)
```

In [15]:

```
#subset2
subset2=df.iloc[[45,21,56,87,55,99,78,97,32],:]
subset2
```

Out[15]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
45	42.12931	322.000000	11.5	79	6	15	0
21	11.00000	320.000000	16.6	73	5	22	1
56	42.12931	127.000000	8.0	78	6	26	0
87	52.00000	82.000000	12.0	86	7	27	1
55	42.12931	135.000000	8.0	75	6	25	0
99	89.00000	229.000000	10.3	90	8	8	0
78	61.00000	285.000000	6.3	84	7	18	2
97	66.00000	185.931507	4.6	87	8	6	2
32	42.12931	287.000000	9.7	74	6	2	0

In [16]:

```
subset2.shape
```

Out[16]:

```
(9, 7)
```

Merging subsets

In [17]:

```
merge1=pd.concat([subset1,subset2])
merge1
```

Out[17]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
3	18.00000	313.000000	11.5	62	5	4	0
5	28.00000	185.931507	14.9	66	5	6	0
6	23.00000	299.000000	8.6	65	5	7	0
7	19.00000	99.000000	13.8	59	5	8	1
23	32.00000	92.000000	12.0	61	5	24	0
43	23.00000	148.000000	8.0	82	6	13	2
12	11.00000	290.000000	9.2	66	5	13	1
45	42.12931	322.000000	11.5	79	6	15	0
21	11.00000	320.000000	16.6	73	5	22	1
56	42.12931	127.000000	8.0	78	6	26	0
87	52.00000	82.000000	12.0	86	7	27	1
55	42.12931	135.000000	8.0	75	6	25	0
99	89.00000	229.000000	10.3	90	8	8	0
78	61.00000	285.000000	6.3	84	7	18	2
97	66.00000	185.931507	4.6	87	8	6	2
32	42.12931	287.000000	9.7	74	6	2	0

In [18]:

```
merge1.shape
```

Out[18]:

```
(16, 7)
```

Deriving correlation between Columns

In [19]:

```
correlation=df.corr()
correlation
```

Out[19]:

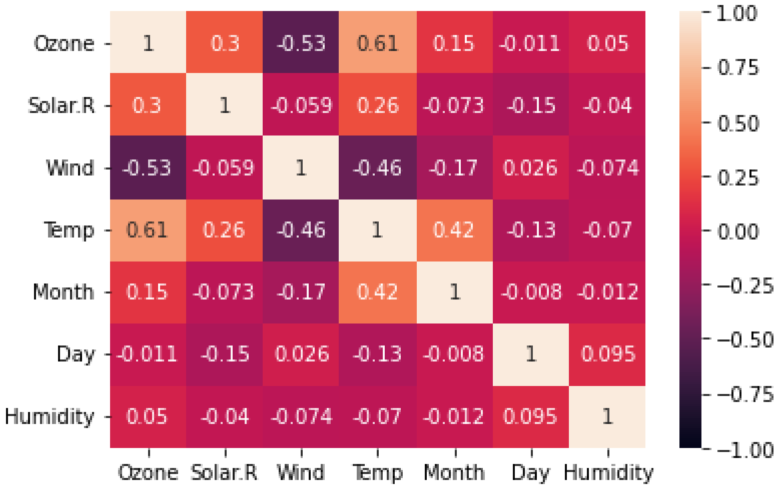
	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
Ozone	1.000000	0.302970	-0.529389	0.608742	0.149081	-0.011355	0.049965
Solar.R	0.302970	1.000000	-0.059408	0.262569	-0.072904	-0.145621	-0.039790
Wind	-0.529389	-0.059408	1.000000	-0.455128	-0.173857	0.025837	-0.073615
Temp	0.608742	0.262569	-0.455128	1.000000	0.420947	-0.130593	-0.070224
Month	0.149081	-0.072904	-0.173857	0.420947	1.000000	-0.007962	-0.011713
Day	-0.011355	-0.145621	0.025837	-0.130593	-0.007962	1.000000	0.094662
Humidity	0.049965	-0.039790	-0.073615	-0.070224	-0.011713	0.094662	1.000000

In [20]:

```
import seaborn as sns
sns.heatmap(correlation, vmin = -1, vmax = 1, annot=True)
```

Out[20]:

```
<AxesSubplot:>
```



E] Building Data Model

Using linear regression model

In [21]:

```
x=df[["Ozone"]]
y=df[["Temp"]]
```

In [22]:

```
from sklearn.model_selection import train_test_split

xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.3)

from sklearn.linear_model import LinearRegression
```

In [23]:

```
linear_reg=LinearRegression()

model=linear_reg.fit(xtrain, ytrain)
```

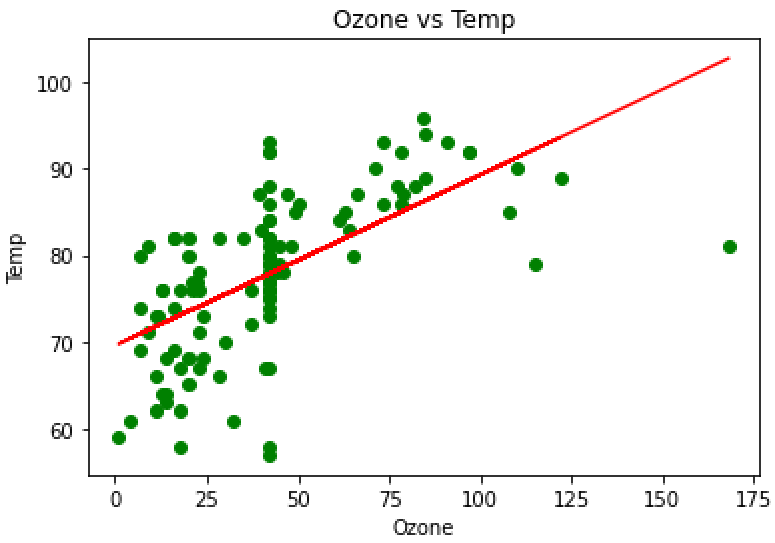
In [24]:

```
y_predict = model.predict(xtest)
```

Plotting graph

In [25]:

```
plt.scatter(xtrain, ytrain, color="green")
plt.plot(xtrain, linear_reg.predict(xtrain), color="red")
plt.xlabel("Ozone")
plt.ylabel("Temp")
plt.title("Ozone vs Temp")
plt.show()
```



Calculating metrics

In [26]:

```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

MSE = mean_squared_error(ytest,y_predict)
RMSE = np.sqrt(MSE)
MAE = mean_absolute_error(ytest,y_predict)
r2_score = r2_score(ytest,y_predict)
```

In [27]:

```
print("MSE- ",MSE)
print("RMSE- ",RMSE)
print("MAE- ",MAE)
print("r2_score- ",r2_score)
```

```
MSE-  63.06113067789956
RMSE-  7.941103870237409
MAE-  6.023553405377359
r2_score-  0.3362547565152908
```