In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv('airquality.csv')
```

In [3]:
```
df.head()
```

Out[3]:

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

In [4]:
```
df.shape
```

Out[4]:
```
(153, 7)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 153 entries, 0 to 152
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  153 non-null    int64  
 1   Ozone       116 non-null    float64
 2   Solar.R     146 non-null    float64
 3   Wind        153 non-null    float64
 4   Temp        153 non-null    int64  
 5   Month       153 non-null    int64  
 6   Day         153 non-null    int64  
dtypes: float64(3), int64(4)
memory usage: 8.5 KB
```

In [7]:
```
df.isnull()
```

Out[7]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	True	True	False	False	False	False
...	...	...	...	...	...	...	...
148	False	False	False	False	False	False	False
149	False	True	False	False	False	False	False
150	False	False	False	False	False	False	False
151	False	False	False	False	False	False	False
152	False	False	False	False	False	False	False

153 rows × 7 columns

In [8]:
```
df.isnull().sum
```

Out[8]:
```
<bound method NDFrame._add_numeric_operations.<locals>.sum of      Unnamed: 0  Ozone
Solar.R   Wind   Temp  Month    Day
0         False  False    False  False  False  False  False
1         False  False    False  False  False  False  False
2         False  False    False  False  False  False  False
3         False  False    False  False  False  False  False
4         False   True     True  False  False  False  False
..          ...    ...      ...    ...    ...    ...    ...
148       False  False    False  False  False  False  False
149       False   True    False  False  False  False  False
150       False  False    False  False  False  False  False
151       False  False    False  False  False  False  False
152       False  False    False  False  False  False  False

[153 rows x 7 columns]>
```

In [9]:
```
df.count()
```

Out[9]:
```
Unnamed: 0    153
Ozone         116
Solar.R       146
Wind          153
Temp          153
Month         153
Day           153
dtype: int64
```

In [10]:
```
df.describe
```

Out[10]:
```
<bound method NDFrame.describe of      Unnamed: 0  Ozone  Solar.R  Wind  Temp  Month
Day
0             1   41.0    190.0   7.4    67      5    1
1             2   36.0    118.0   8.0    72      5    2
2             3   12.0    149.0  12.6    74      5    3
3             4   18.0    313.0  11.5    62      5    4
4             5    NaN      NaN  14.3    56      5    5
..          ...    ...      ...   ...   ...    ...  ...
148         149   30.0    193.0   6.9    70      9   26
149         150    NaN    145.0  13.2    77      9   27
150         151   14.0    191.0  14.3    75      9   28
151         152   18.0    131.0   8.0    76      9   29
152         153   20.0    223.0  11.5    68      9   30

[153 rows x 7 columns]>
```

In [13]:
```
a = df.dropna()
```

In [14]:
```
a.shape
```

Out[14]:
```
(111, 7)
```

In [15]:
```
a = df.fillna(0)
```

In [16]:
```
a.shape
```

Out[16]:
```
(153, 7)
```

In [17]:
```
a.head()
```

Out[17]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day
0	1	41.0	190.0	7.4	67	5	1
1	2	36.0	118.0	8.0	72	5	2
2	3	12.0	149.0	12.6	74	5	3
3	4	18.0	313.0	11.5	62	5	4
4	5	0.0	0.0	14.3	56	5	5

In [18]:
```
a = df.fillna(method='pad')
```

In [19]:
```
a.head()
```

Out[19]:

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

In [20]:
```
 a = df.fillna(method='backfill')
```

In [21]:
```
a.head()
```

Out[21]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day
0	1	41.0	190.0	7.4	67	5	1
1	2	36.0	118.0	8.0	72	5	2
2	3	12.0	149.0	12.6	74	5	3
3	4	18.0	313.0	11.5	62	5	4
4	5	28.0	299.0	14.3	56	5	5

In [22]:
```
import numpy as np
```

In [23]:
```
A = df['Ozone'].replace(np.NaN,df['Ozone'].mean())
```

In [24]:
```
A.head()
```

Out[24]:
```
0    41.00000
1    36.00000
2    12.00000
3    18.00000
4    42.12931
Name: Ozone, dtype: float64
```

In [25]:
```
A = df['Ozone'].replace(np.NaN,df['Ozone'].median())
```

In [26]:
```
A.head()
```

Out[26]:
```
0    41.0
1    36.0
2    12.0
3    18.0
4    31.5
Name: Ozone, dtype: float64
```

In [28]:
```
A = df['Ozone'].replace(np.NaN,df['Ozone'].mode())
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3808\536667357.py in <module>
----> 1 A = df['Ozone'].replace(np.NaN,df['Ozone'].mode())

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\series.py in replace(self, to_
replace, value, inplace, limit, regex, method)
   4958         method: str | lib.NoDefault = lib.no_default,
   4959     ):
-> 4960         return super().replace(
   4961             to_replace=to_replace,
   4962             value=value,

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in replace(self, to
_replace, value, inplace, limit, regex, method)
   6734                     # Operate column-wise
   6735                     if self.ndim == 1:
-> 6736                         raise ValueError(
   6737                             "Series.replace cannot use dict-value and "
   6738                             "non-None to_replace"

ValueError: Series.replace cannot use dict-value and non-None to_replace
```

In [29]:
```
from sklearn.impute import SimpleImputer
```

In [30]:
```
imp = SimpleImputer(missing_values=np.nan,strategy='mean')
```

In [31]:
```
A = imp.fit_transform(df)
```

In [32]:
```
A
```

Out[32]:
```
array([[  1.,  41., 190., ...,  67.,   5.,   1.],
       [  2.,  36., 118., ...,  72.,   5.,   2.],
       [  3.,  12., 149., ...,  74.,   5.,   3.],
       ...,
       [151.,  14., 191., ...,  75.,   9.,  28.],
       [152.,  18., 131., ...,  76.,   9.,  29.],
       [153.,  20., 223., ...,  68.,   9.,  30.]])
```

In [33]:
```
A = pd.DataFrame(A, columns=df.columns)
```

In [34]:
```
A.head()
```

Out[34]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day
0	1.0	41.00000	190.000000	7.4	67.0	5.0	1.0
1	2.0	36.00000	118.000000	8.0	72.0	5.0	2.0
2	3.0	12.00000	149.000000	12.6	74.0	5.0	3.0
3	4.0	18.00000	313.000000	11.5	62.0	5.0	4.0
4	5.0	42.12931	185.931507	14.3	56.0	5.0	5.0

In [35]:
```
from sklearn.model_selection import train_test_split
```

In [36]:
```
len(A)
```

Out[36]:
```
153
```

In [37]:
```
train, test = train_test_split(A)
```

In [38]:
```
len(train)
```

Out[38]:
```
114
```

In [39]:
```
len(test)
```

Out[39]:
```
39
```

In [40]:
```
train.head()
```

Out[40]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day
16	17.0	34.00000	307.000000	12.0	66.0	5.0	17.0
26	27.0	42.12931	185.931507	8.0	57.0	5.0	27.0
40	41.0	39.00000	323.000000	11.5	87.0	6.0	10.0
128	129.0	32.00000	92.000000	15.5	84.0	9.0	6.0
8	9.0	8.00000	19.000000	20.1	61.0	5.0	9.0

In [41]:
```
train, test = train_test_split(A, test_size = 0.20)
```

In [42]:
```
len(test)
```

Out[42]:
```
31
```

In [43]:
```
len(train)
```

Out[43]:
```
122
```

In [44]:
```
A.describe()
```

Out[44]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day
count	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000
mean	77.000000	42.129310	185.931507	9.957516	77.882353	6.993464	15.803922
std	44.311398	28.693372	87.960267	3.523001	9.465270	1.416522	8.864520
min	1.000000	1.000000	7.000000	1.700000	56.000000	5.000000	1.000000
25%	39.000000	21.000000	120.000000	7.400000	72.000000	6.000000	8.000000
50%	77.000000	42.129310	194.000000	9.700000	79.000000	7.000000	16.000000
75%	115.000000	46.000000	256.000000	11.500000	85.000000	8.000000	23.000000
max	153.000000	168.000000	334.000000	20.700000	97.000000	9.000000	31.000000

In [45]:
```
from sklearn.preprocessing import StandardScaler
```

In [46]:
```
scaler = StandardScaler()
```

In [47]:
```
B = scaler.fit_transform(A)
```

In [48]:
```
pd.DataFrame(B).describe()
```

Out[48]:

	0	1	2	3	4	5	
count	1.530000e+02	1.530000e+02	1.530000e+02	1.530000e+02	1.530000e+02	1.530000e+02	1.
mean	-1.596399e-17	-8.344814e-17	-6.313033e-17	6.385596e-17	8.069072e-16	6.675851e-17	-6
std	1.003284e+00	1.003284e+00	1.003284e+00	1.003284e+00	1.003284e+00	1.003284e+00	1.
min	-1.720767e+00	-1.438115e+00	-2.040912e+00	-2.351584e+00	-2.319450e+00	-1.411916e+00	-1.
25%	-8.603835e-01	-7.388013e-01	-7.520217e-01	-7.283322e-01	-6.235080e-01	-7.036434e-01	-8
50%	0.000000e+00	0.000000e+00	9.203008e-02	-7.333578e-02	1.184665e-01	4.629233e-03	2
75%	8.603835e-01	1.353414e-01	7.992086e-01	4.392701e-01	7.544446e-01	7.129018e-01	8
max	1.720767e+00	4.401158e+00	1.688885e+00	3.059256e+00	2.026401e+00	1.421174e+00	1.

In [49]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [50]:
```
scaler = MinMaxScaler()
```

In [51]:
```
B = scaler.fit_transform(A)
```

In [52]:
```
pd.DataFrame(B).describe()
```

Out[52]:

	0	1	2	3	4	5	6
count	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000
mean	0.500000	0.246283	0.547191	0.434606	0.533716	0.498366	0.493464
std	0.291522	0.171817	0.268992	0.185421	0.230860	0.354131	0.295484
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.250000	0.119760	0.345566	0.300000	0.390244	0.250000	0.233333
50%	0.500000	0.246283	0.571865	0.421053	0.560976	0.500000	0.500000
75%	0.750000	0.269461	0.761468	0.515789	0.707317	0.750000	0.733333
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

In [53]:
```
B = pd.DataFrame(B).describe()
```

In [54]:
```
from sklearn.preprocessing import Binarizer
```

In [55]:
```
bin = Binarizer(threshold=0.5)
```

In [56]:
```
B = bin.fit_transform(B)
```

In [57]:
```
pd.DataFrame(B)
```

Out[57]:

	0	1	2	3	4	5	6
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1	0.0	0.0	1.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	1.0	0.0	1.0	0.0	0.0
6	1.0	0.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0

In [58]:
```
pd.DataFrame(B)
```

Out[58]:

	0	1	2	3	4	5	6
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1	0.0	0.0	1.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	1.0	0.0	1.0	0.0	0.0
6	1.0	0.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0

In [59]:
```
data=pd.read_csv('student.csv')
```

In [60]:
```
from sklearn.preprocessing import LabelEncoder
```

In [61]:
```
le = LabelEncoder()
```

In [62]:
```
B = le.fit_transform(data['name'])
```

In [ ]:
```

```

In [63]:
```
B
```

Out[63]:
```
array([6, 2, 5, 0, 8, 9, 1, 3, 4, 7])
```

In [64]:
```
B = data[:]
```

In [65]:
```
B['name'] = le.fit_transform(B['name'])
```

In [66]:
```
B
```

Out[66]:

	roll	name	class	marks	age
0	1	6	TE	56.77	22
1	2	2	TE	59.77	21
2	3	5	BE	76.88	19
3	4	0	TE	69.66	20
4	5	8	TE	63.28	20
5	6	9	BE	49.55	20
6	7	1	BE	65.34	19
7	8	3	BE	68.33	23
8	9	4	TE	56.75	20
9	10	7	BE	78.66	21

In [67]:
```
A
```

Out[67]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day
0	1.0	41.00000	190.000000	7.4	67.0	5.0	1.0
1	2.0	36.00000	118.000000	8.0	72.0	5.0	2.0
2	3.0	12.00000	149.000000	12.6	74.0	5.0	3.0
3	4.0	18.00000	313.000000	11.5	62.0	5.0	4.0
4	5.0	42.12931	185.931507	14.3	56.0	5.0	5.0
...	...	...	...	...	...	...	...
148	149.0	30.00000	193.000000	6.9	70.0	9.0	26.0
149	150.0	42.12931	145.000000	13.2	77.0	9.0	27.0
150	151.0	14.00000	191.000000	14.3	75.0	9.0	28.0
151	152.0	18.00000	131.000000	8.0	76.0	9.0	29.0
152	153.0	20.00000	223.000000	11.5	68.0	9.0	30.0

153 rows × 7 columns

In [68]:
```
from sklearn.linear_model import LinearRegression
```

In [69]:
```
X=A['Ozone'].values
```

In [70]:
```
X=X.reshape(-1,1)
```

In [71]:
```
Y = A['Temp']
```

In [72]:
```
model = LinearRegression()
```

In [73]:
```
model.fit(X,Y)
```

Out[73]:
```
LinearRegression()
```

In [74]:
```
model.score(X,Y)*100
```

Out[74]:
```
37.056682983646425
```

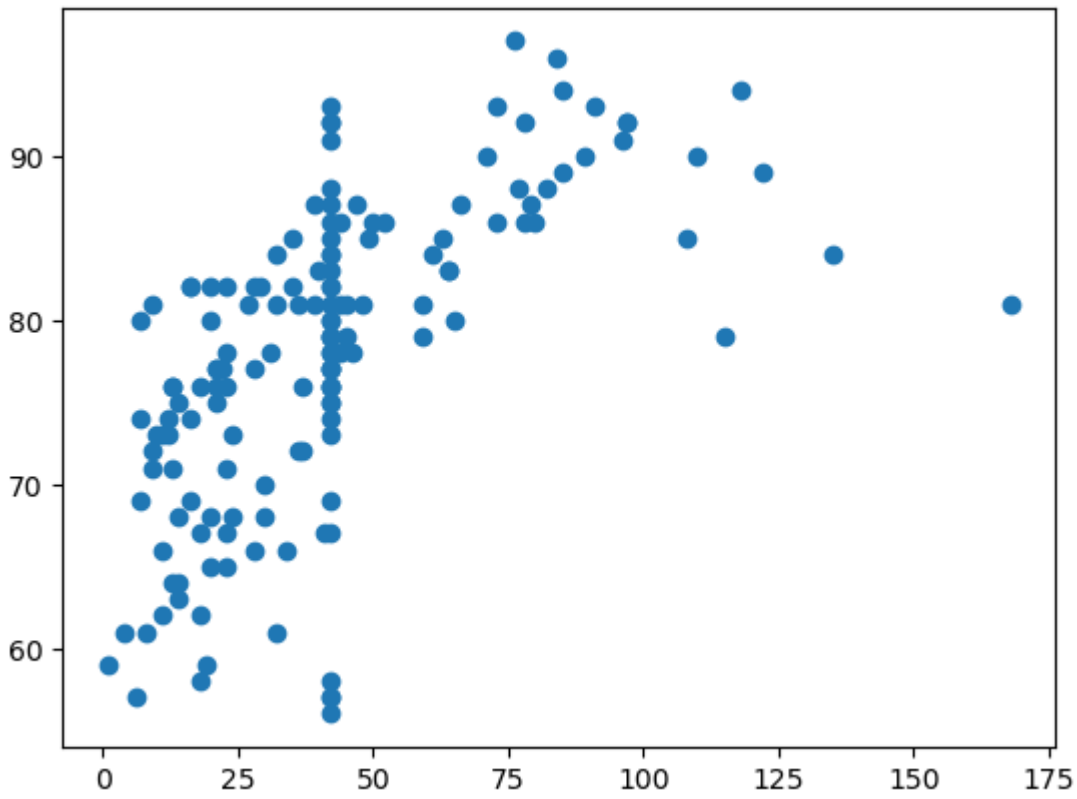
In [75]:
```
model.predict([[128]])
```

Out[75]:
```
array([95.12601986])
```

In [76]:
```
import matplotlib.pyplot as plt
```

In [77]:
```
plt.scatter(X,Y)
```

Out[77]:
```
<matplotlib.collections.PathCollection at 0xe9cc47f190>
```



In [ ]:
```

```