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

```
data = pd.read_csv("airquality.csv")
```

```
data.head()
```

	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

```
data.isnull().sum()
```

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

```
data.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
```

```
ozone_mean = data['Ozone'].mean()
ozone_mean
```

```
42.12931034482759
```

```
data = data.fillna(ozone_mean)
data.head()
```

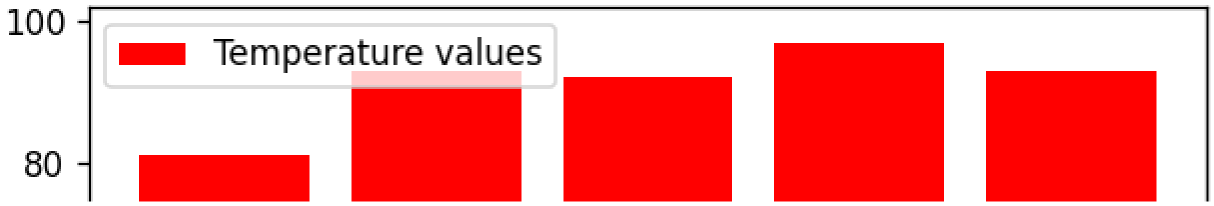
	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day
0	1	41.00000	190.00000	7.4	67	5	1
1	2	36.00000	118.00000	8.0	72	5	2
2	3	12.00000	149.00000	12.6	74	5	3
3	4	18.00000	313.00000	11.5	62	5	4
4	5	42.12931	42.12931	14.3	56	5	5

```
data.isnull().sum()
```

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

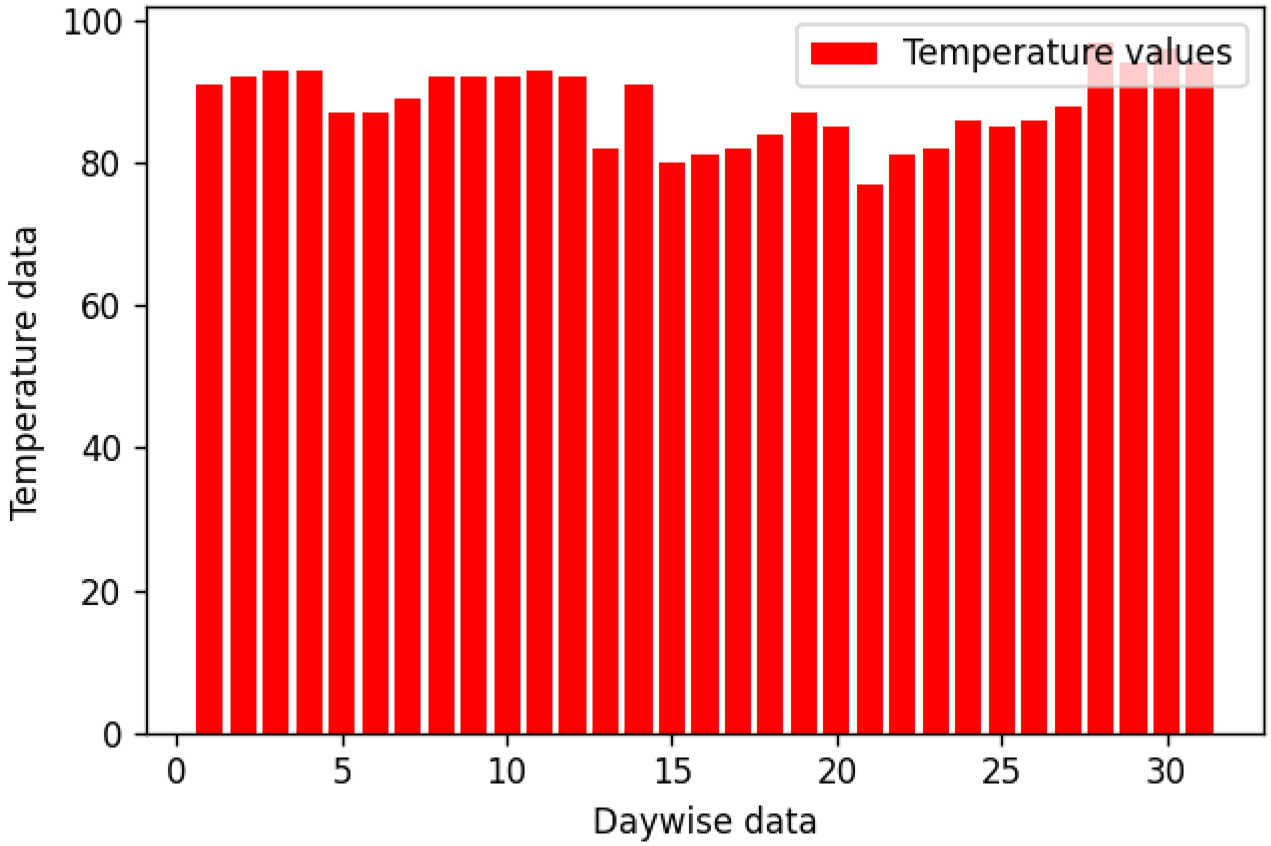
```
x = data['Month']
y = data['Temp']
plt.figure(dpi = 120)
plt.bar(x,y,color = 'red',label='Temperature values')
plt.xlabel('Monthly data')
plt.ylabel('Temperature data')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f357d244fd0>
```



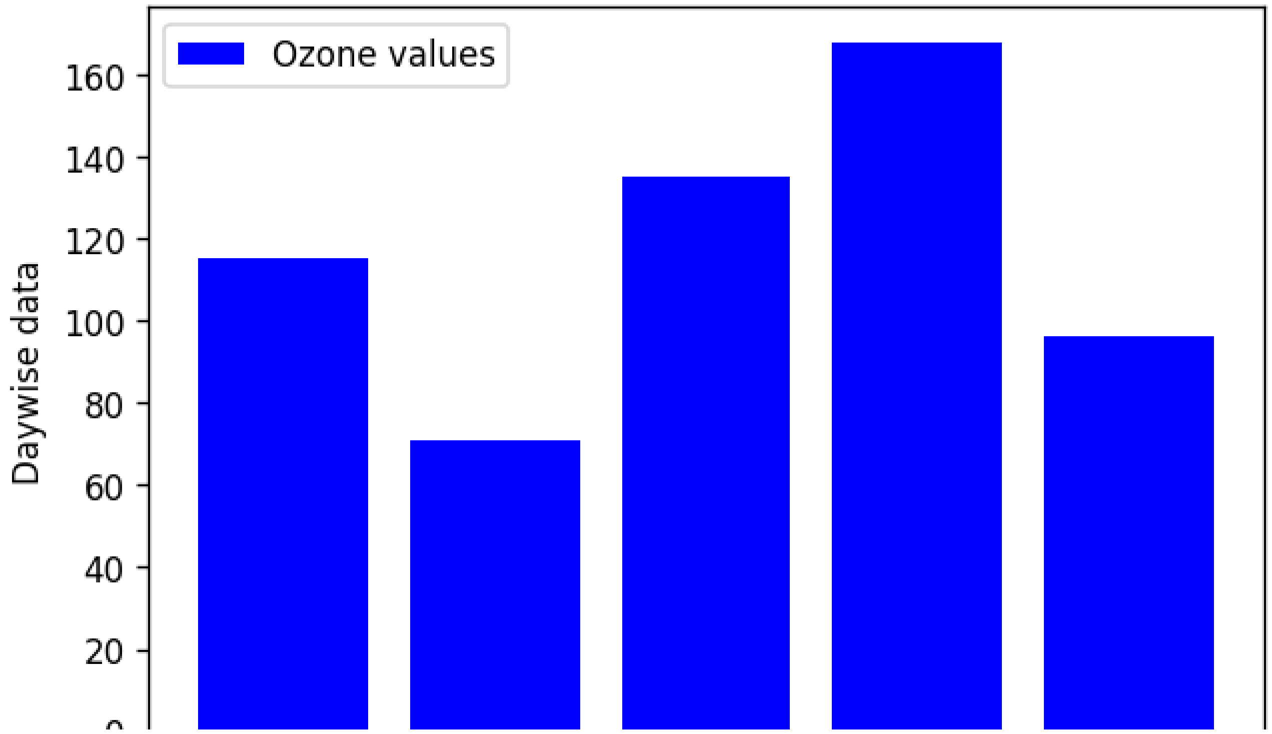
```
x = data['Day']
y = data['Temp']
plt.figure(dpi = 120)
plt.bar(x,y,color = 'red',label='Temperature values')
plt.xlabel('Daywise data')
plt.ylabel('Temperature data')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f357cd02f10>
```



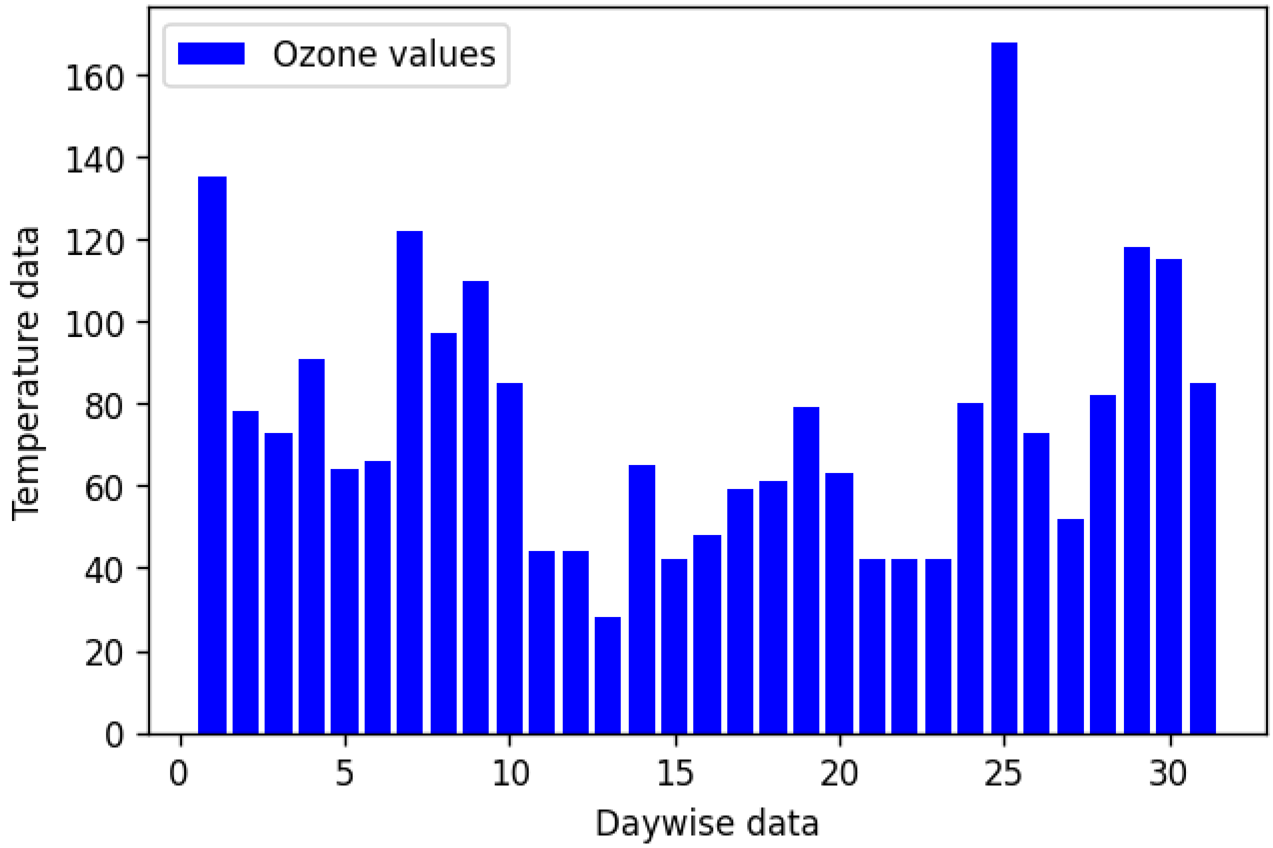
```
x = data['Month']
y = data['Ozone']
plt.figure(dpi = 120)
plt.bar(x,y,color = 'blue',label='Ozone values')
plt.xlabel('Monthly data')
plt.ylabel('Daywise data')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f357c612510>
```



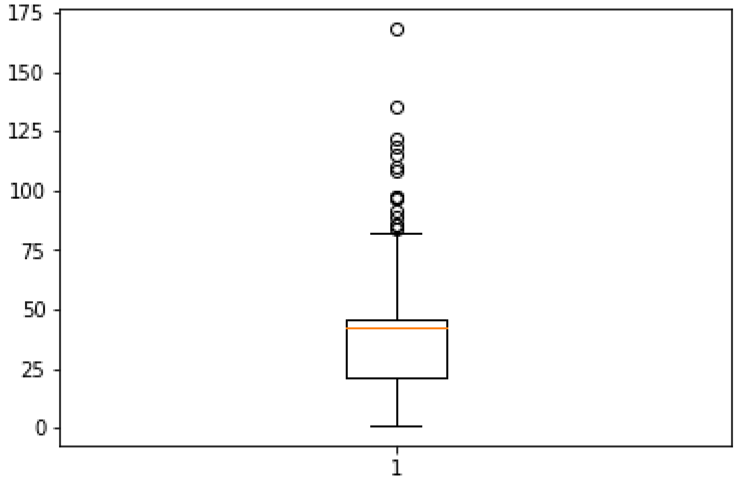
```
x = data['Day']
y = data['Ozone']
plt.figure(dpi = 120)
plt.bar(x,y,color = 'blue',label='Ozone values')
plt.xlabel('Daywise data')
plt.ylabel('Temperature data')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f357d050790>
```



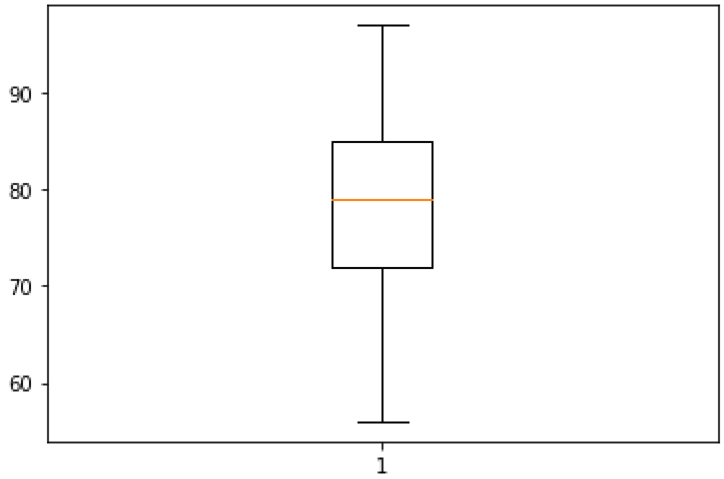
```
x = data['Ozone']
plt.boxplot(x)
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f357c192c10>,
  <matplotlib.lines.Line2D at 0x7f357c1a1190>],
 'caps': [<matplotlib.lines.Line2D at 0x7f357c1a16d0>,
  <matplotlib.lines.Line2D at 0x7f357c1a1c10>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f357c192690>],
 'medians': [<matplotlib.lines.Line2D at 0x7f357c1a71d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f357c1a7710>],
 'means': []}
```



```
x = data['Temp']
plt.boxplot(x)
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f357cf02690>,
  <matplotlib.lines.Line2D at 0x7f357ce663d0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f357ce7e610>,
  <matplotlib.lines.Line2D at 0x7f357ced8fd0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f357cd699d0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f357ce309d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f357ce24f10>],
 'means': []}
```



```
from sklearn.model_selection import train_test_split
```

```
X = data['Temp']
Y = data['Ozone']
X_train,x_test,y_train,y_test = train_test_split(X, Y, test_size=0.20, rando

```

```
X_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
((122,), (122,), (31,), (31,))
```

```
X = np.array(X).reshape(-1,1)
Y = np.array(Y).reshape(-1,1)
from sklearn.linear_model import LinearRegression
model = LinearRegression().fit(X,Y)
```

```
z = np.array([100,76,45])
z = z.reshape(-1,1)
model.predict(z)
```

```
array([[ 82.94440343],
       [ 38.6556854 ],
       [-18.55057539]])
```

```
#training_pred5
```

```
X_train = np.array(X_train).reshape(-1,1)
y_train = np.array(y_train).reshape(-1,1)
training_model = LinearRegression().fit(X_train,y_train)
```

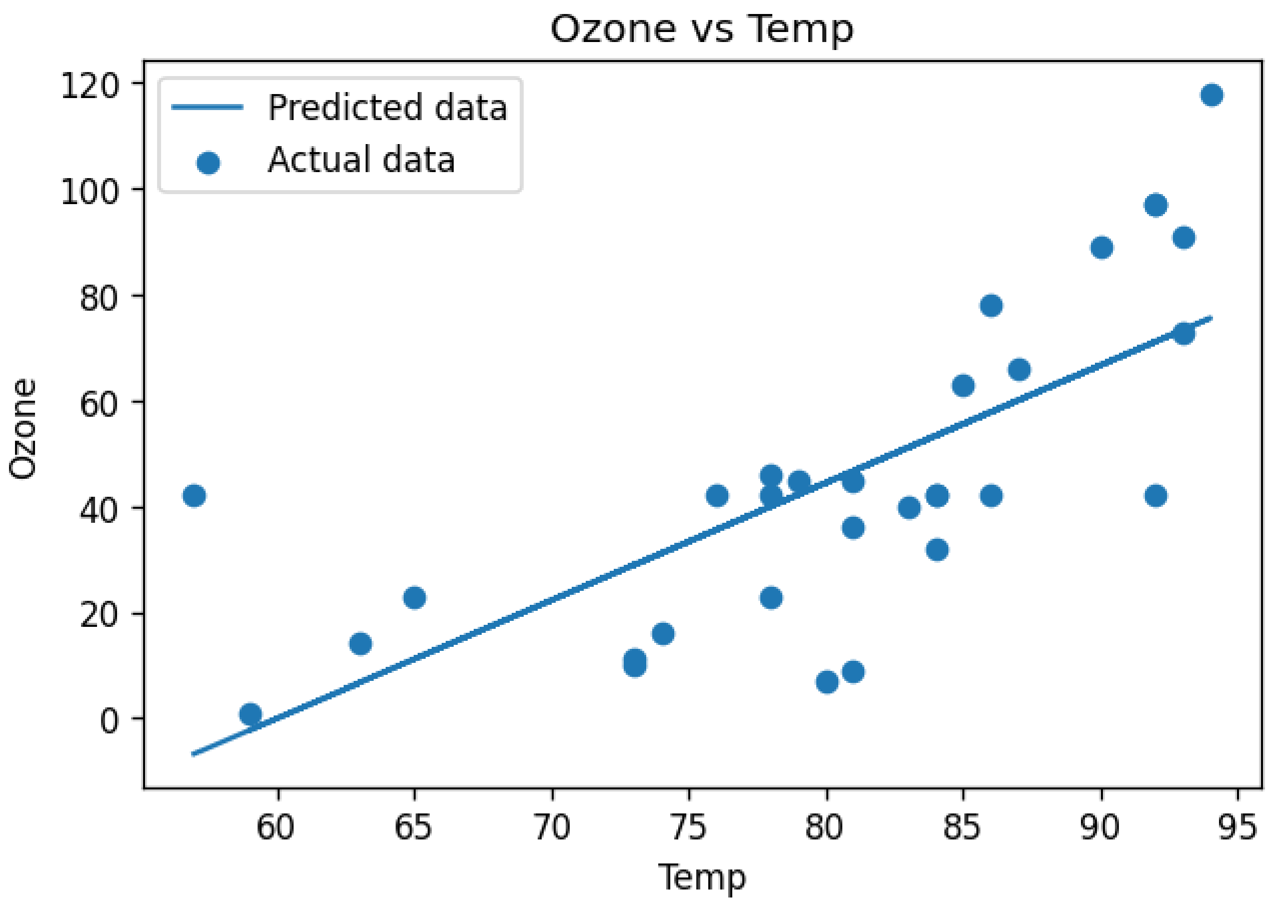
```
x_train_predicted =  training_model.predict(X_train)

```

```
x_test = np.array(x_test).reshape(-1,1)
y_test = np.array(y_test).reshape(-1,1)
testing_model = LinearRegression().fit(x_test,y_test)
```

```
y_test_predicted = testing_model.predict(x_test)
```

```
plt.figure(dpi = 120)
plt.scatter(x_test,y_test,label="Actual data")
plt.plot(x_test,y_test_predicted,label="Predicted data")
plt.title("Ozone vs Temp")
plt.xlabel("Temp")
plt.ylabel("Ozone")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f357b357650>
```



```
from sklearn.metrics import mean_squared_error

```

```
training_mse = mean_squared_error(y_train,x_train_predicted)
training_mse
```

```
532.9230042762385
```

```
training_rmse = training_mse**0.5
training_rmse
```

```
23.085125173501627
```

```
from sklearn.metrics import r2_score
model_mean = np.mean(y_train)
model_mean
r_squared  = 1 - (training_mse/model_mean)
r_squared
r2_score(y_train,x_train_predicted)
```

```
0.32981095616195255
```

```
testing_mse = mean_squared_error(y_test,y_test_predicted)
testing_mse
```

```
425.0761498073731
```

```
testing_rmse = testing_mse**0.5
testing_rmse
```

```
20.617374949478247
```

```
model_mean = np.mean(y_test)
model_mean
r_squared = 1 - (testing_mse/model_mean)
r_squared
r2_score(y_test,y_test_predicted)
```

```
0.521846272297713
```

✓ 0s completed at 8:21 AM ● ✕