```
import pandas as pd
import numpy as np
```

```
a = pd.read_csv("/content/airquality.csv")
a
```

	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	medium
2	3	12.0	149.0	12.6	74	5	3	low
3	4	18.0	313.0	11.5	62	5	4	medium
4	5	NaN	NaN	14.3	56	5	5	low
...	...	...	...	...	...	...	...	...
148	149	30.0	193.0	6.9	70	9	26	low
149	150	NaN	145.0	13.2	77	9	27	NaN
150	151	14.0	191.0	14.3	75	9	28	low
151	152	18.0	131.0	8.0	76	9	29	NaN
152	153	20.0	223.0	11.5	68	9	30	low

153 rows × 8 columns

Next steps: Generate code with a | View recommended plots | New interactive sheet

```
a.isnull().sum()
```

	0
Ozone	37
Solar.R	7
Wind	0
Temp	0
Month	0
Day	0
Humidity	72

dtype: int64

```
a['Ozone']=a['Ozone'].fillna(a['Ozone'].mean())
a['Solar.R']=a['Solar.R'].fillna(a['Solar.R'].mean())
a['Wind']=a['Wind'].fillna(a['Wind'].mean())
a['Humidity']=a['Humidity'].fillna(a['Humidity'].mode()[0])
a.isnull().sum()
```

	0
Ozone	0
Solar.R	0
Wind	0
Temp	0
Month	0
Day	0
Humidity	0

dtype: int64

```
from sklearn import preprocessing
a['Humidity']=preprocessing.LabelEncoder().fit_transform(a['Humidity'])
a
```

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

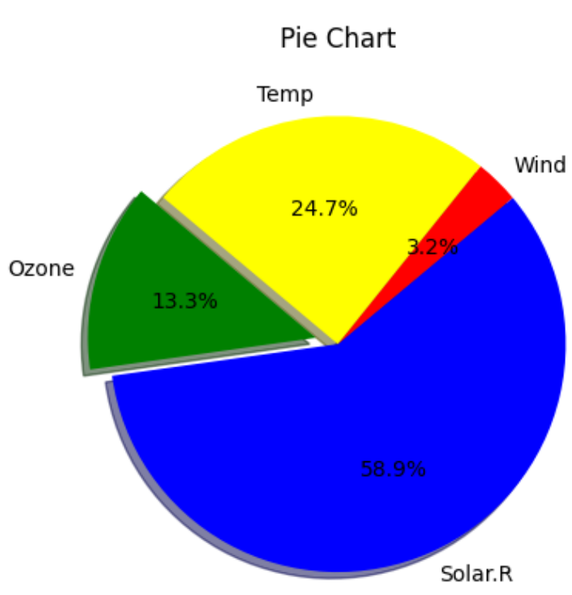
153 rows × 7 columns

Next steps: Generate code with a | View recommended plots | New interactive sheet

```
#1. PIE Chart
```

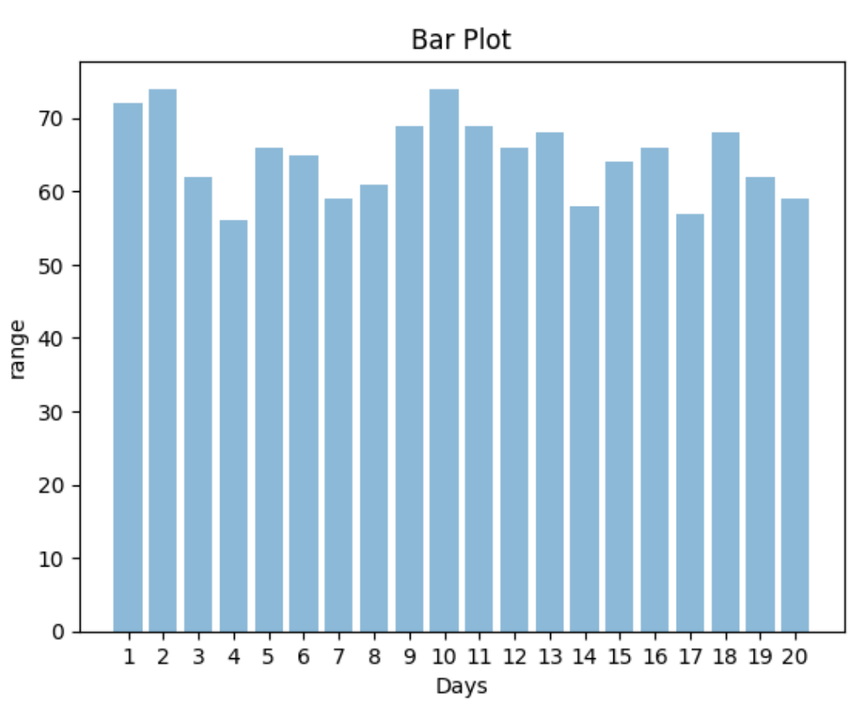
```
import matplotlib.pyplot as plt
labels = ['Ozone', 'Solar.R', 'Wind', 'Temp']
sizes = [a['Ozone'].mean(), a['Solar.R'].mean(), a['Wind'].mean(), a['Temp'].mean()]
colors = ['Green', 'Blue', 'Red', 'Yellow']
explode = (0.1, 0, 0, 0)
```

```
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=140)
plt.title('Pie Chart')
plt.show()
```



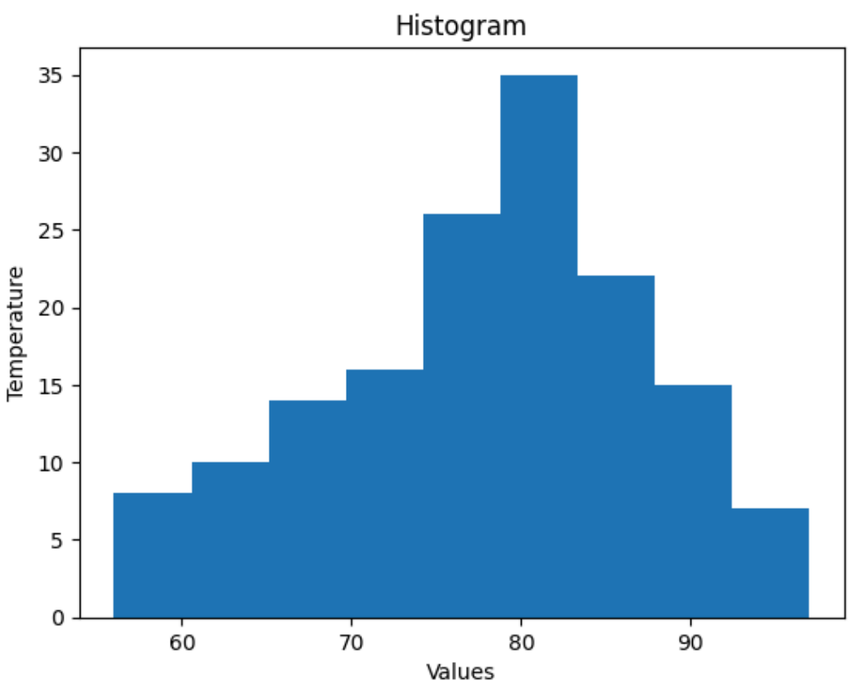

```
#2. BAR Plot
```

```
import numpy as np
h = a.iloc[1:21,3]
y_pos = np.arange(len(h))
v = range(1, 21)
plt.bar(y_pos, h, align = 'center', alpha =0.5)
plt.xticks(y_pos, v)
plt.ylabel('range')
plt.xlabel('Days')
plt.title('Bar Plot')
plt.show()
```



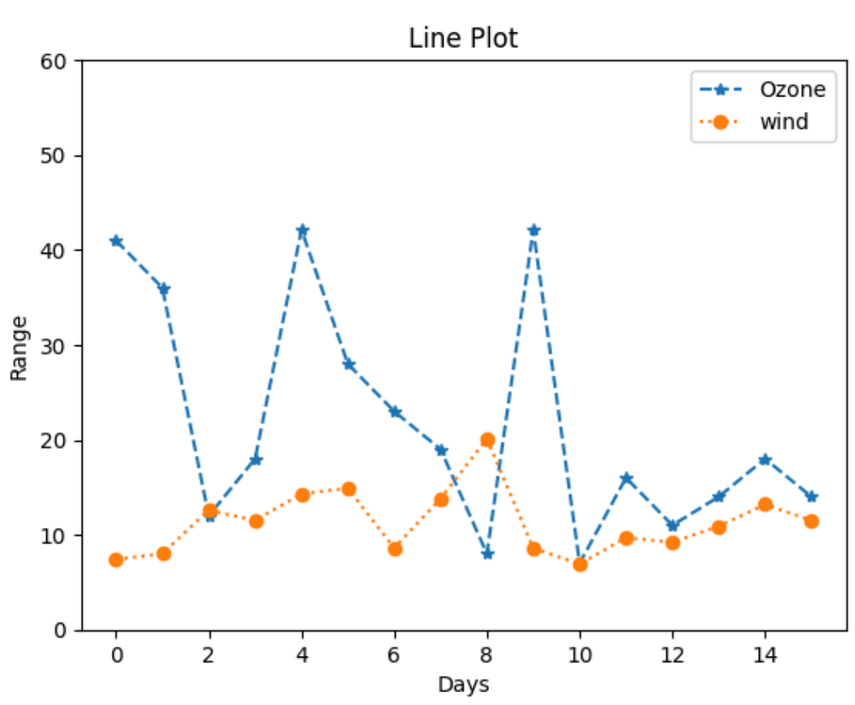

```
#3.HISTOGRAM
```

```
e=a.iloc[:,3]
plt.hist(e, bins='auto')
plt.title('Histogram')
plt.ylabel("Temperature")
plt.xlabel("Values")
plt.show()
```



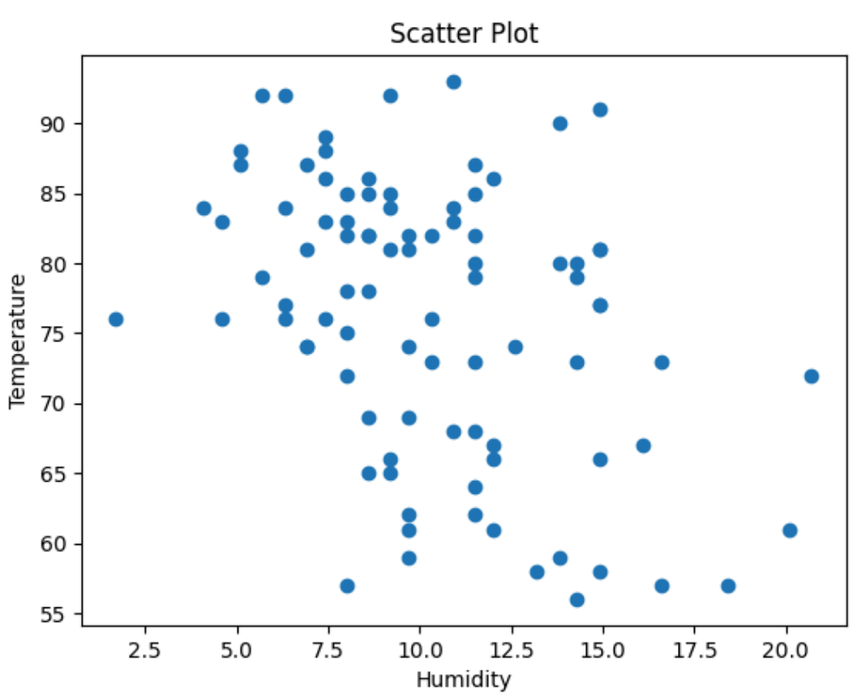

```
#4.LINE graph
```

```
l=a.iloc[:16, 0]
m=a.iloc[:16, 2]
plt.plot(l, label="Ozone", marker = '*', linestyle ='dashed')
plt.plot(m, label="wind", marker = 'o', linestyle ='dotted')
plt.ylim(0, 60)
plt.legend()
plt.title('Line Plot')
plt.ylabel("Range")
plt.xlabel("Days")
plt.savefig('plot.png')
plt.show()
```




```
#%. SCATTER Plot
```

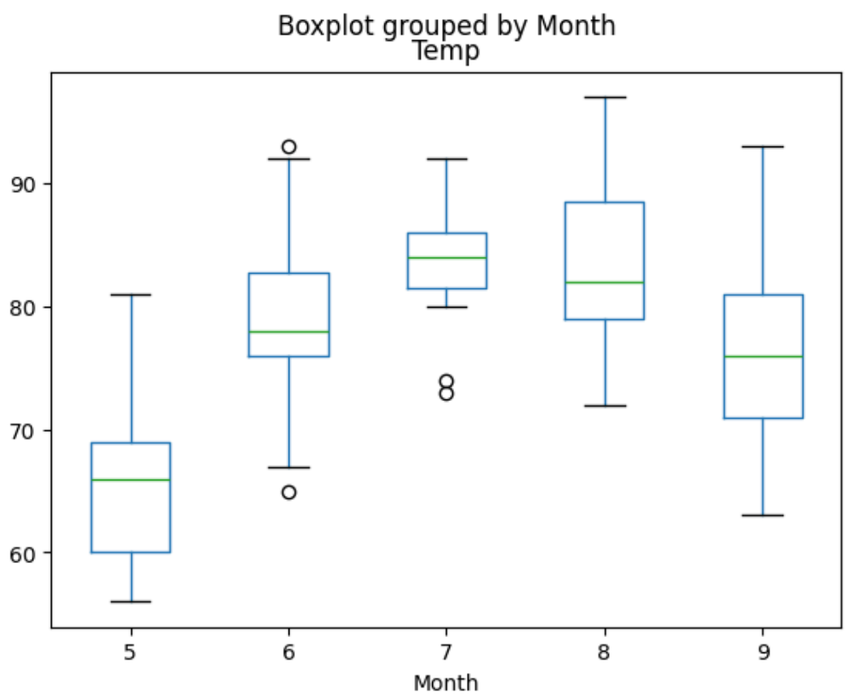
```
s=a.iloc[1:91,2]
t=a.iloc[1:91, 3]
plt.scatter(s, t)
plt.title('Scatter Plot')
plt.ylabel("Temperature")
plt.xlabel("Humidity")
plt.show()
```




```
#6. BOX Plot
```

```
a.boxplot(by='Month', column=['Temp'], grid=False)
```

```
<Axes: title={'center': 'Temp'}, xlabel='Month'>
```



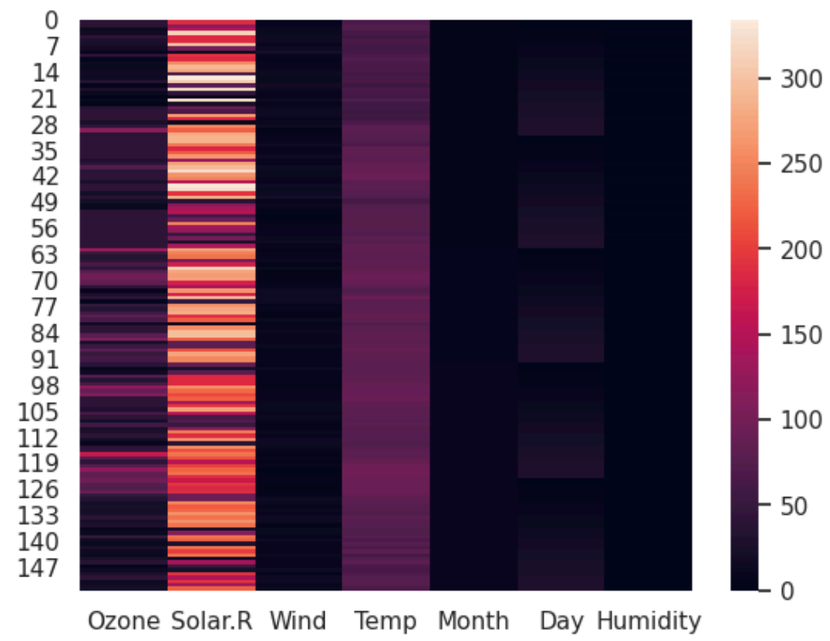

```
#7. PAIR Plot
```

```
import seaborn as sns
sns.set(style='dark')
sns.pairplot(a)
plt.show()
```


```
#7. HEATMAP
```

```
sns.heatmap(a)
```

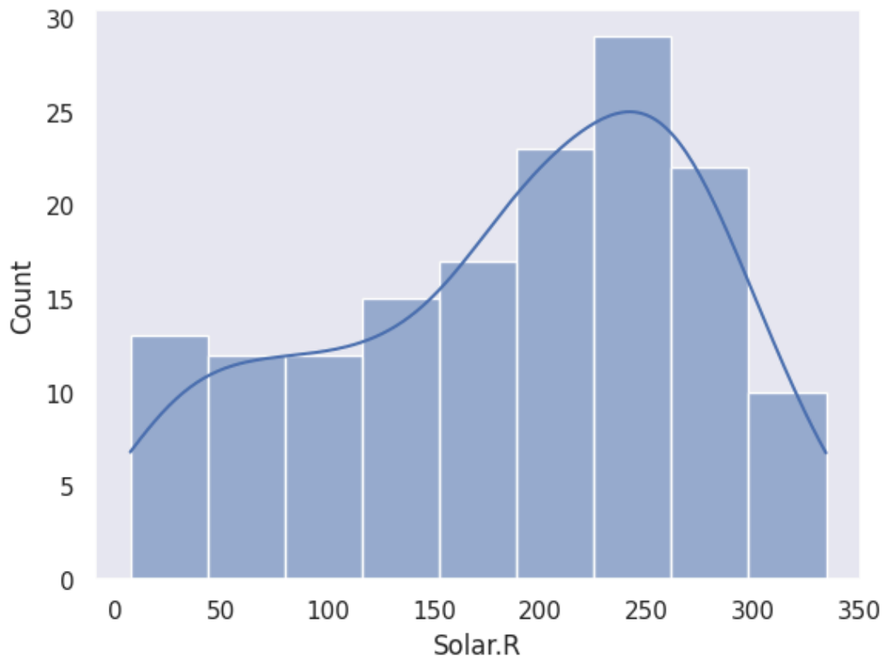
```
<Axes: >
```




```
#0. HISTOGRAM
```

```
sns.histplot(data=a, x="Solar.R", kde=True)
```

```
<Axes: xlabel='Solar.R', ylabel='Count'>
```




```
#10. WorldCloud
```

Generated code may be subject to a license | TheBridgeMachineLearningPythonLibrary/MachineLearningToolKit

```
!pip -q install WorldCloud
from wordcloud import WordCloud
with open(r"sample.txt", 'r', encoding='utf-8') as file:
  text = file.read()
wordcloud = WordCloud(width=800, height=800, background_color='white').generate(text)
plt.figure(figsize=(8, 8), facecolor=None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

```
ERROR: Could not find a version that satisfies the requirement WorldCloud (from versions: none)
ERROR: No matching distribution found for WorldCloud
```