In [23]:
```
import pandas as pd
```

In [24]:
```
import seaborn as sns
```

In [25]:
```
tej = pd.read_csv("C:\\Users\\Shree\\Desktop\\dsbdl_lab\\airquality.csv")
```

In [26]:
```
tej
```

Out[26]:

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

153 rows × 8 columns

In [27]:
```
tej.shape
```

Out[27]:
```
(153, 8)
```

In [28]:
```
tej.head()
```

Out[28]:

	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	Low
2	3	12.0	149.0	12.6	74	5	3	High
3	4	18.0	313.0	11.5	62	5	4	Medium
4	5	NaN	NaN	14.3	56	5	5	High

In [29]:
```
tej.isnull().sum()
```

Out[29]:
```
Unnamed: 0     0
Ozone         37
Solar.R        7
Wind           0
Temp           0
Month          0
Day            0
Humidity       4
dtype: int64
```

Data Cleaning

In [30]:
```
tej = tej.drop('Unnamed: 0' , axis=1)
```

In [31]:
```
tej
```

Out[31]:

	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	Low
2	12.0	149.0	12.6	74	5	3	High
3	18.0	313.0	11.5	62	5	4	Medium
4	NaN	NaN	14.3	56	5	5	High
...	...	...	...	...	...	...	...
148	30.0	193.0	6.9	70	9	26	High
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	High

153 rows × 7 columns

Replacing Numerical Null Values

In [33]:
```
tej['Ozone'] = tej['Ozone'].fillna(tej['Ozone']. mean())
tej.isnull().sum()
```

Out[33]:
```
Ozone       0
Solar.R     7
Wind        0
Temp        0
Month       0
Day         0
Humidity    4
dtype: int64
```

In [34]:
```
tej['Solar.R'] = tej['Solar.R'].fillna(tej['Solar.R']. mean())
tej.isnull().sum()
```

Out[34]:
```
Ozone       0
Solar.R     0
Wind        0
Temp        0
Month       0
Day         0
Humidity    4
dtype: int64
```

In [35]:
```
tej['Humidity'] = tej['Humidity'].fillna(tej['Humidity']. mode()[0])
tej.isnull().sum()
```

Out[35]:
```
Ozone       0
Solar.R     0
Wind        0
Temp        0
Month       0
Day         0
Humidity    0
dtype: int64
```

In [37]:
```
tej.dtypes
```

Out[37]:
```
Ozone       float64
Solar.R     float64
Wind        float64
Temp          int64
Month         int64
Day           int64
Humidity     object
dtype: object
```

Data Transformation

In [42]:
```
from sklearn.preprocessing import LabelEncoder
```

In [44]:
```
label = LabelEncoder()
```

In [45]:
```
tej['Humidity'] = label.fit_transform(tej['Humidity'])
```

In [46]:
```
tej
```

Out[46]:

	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	1
2	12.00000	149.000000	12.6	74	5	3	0
3	18.00000	313.000000	11.5	62	5	4	2
4	42.12931	185.931507	14.3	56	5	5	0
...	...	...	...	...	...	...	...
148	30.00000	193.000000	6.9	70	9	26	0
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	0

153 rows × 7 columns

In [48]:
```
tej['Humidity'].unique()
```

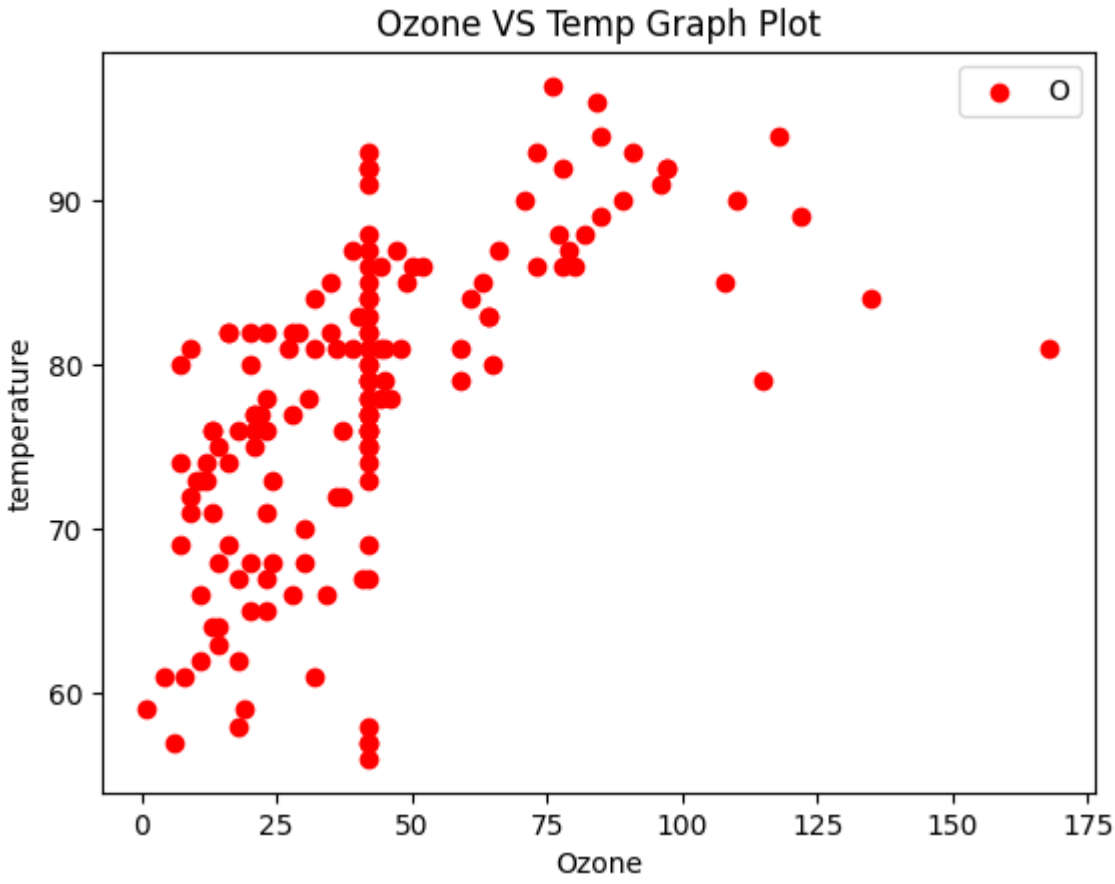
Out[48]:
```
array([0, 1, 2])
```

Visualizing the data

1)Scatter Plot

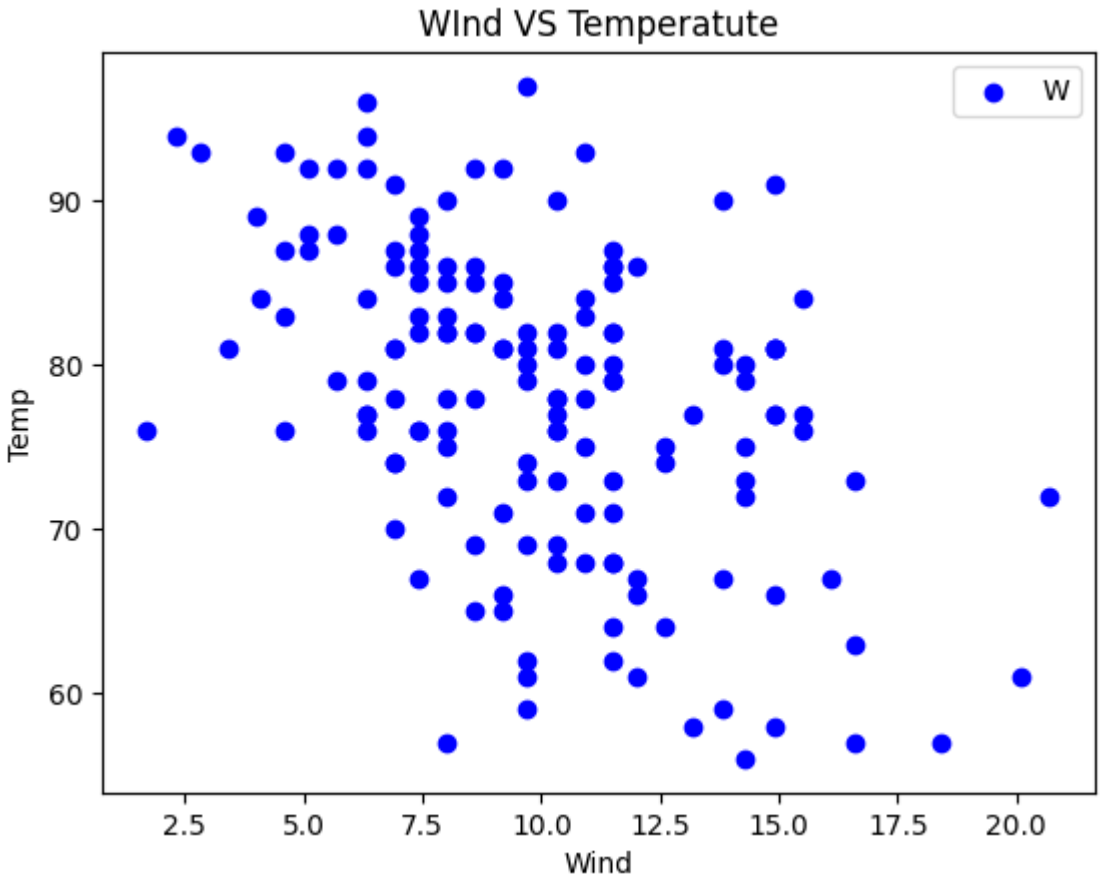
In [54]:
```
import matplotlib.pyplot as plt
```

In [65]:
```
plt.scatter(x = tej['Ozone'] , y = tej['Temp'] , c='red')
plt.legend('Ozone')
plt.xlabel("Ozone")
plt.ylabel("temperature")
plt.title("Ozone VS Temp Graph Plot")
plt.show()
```



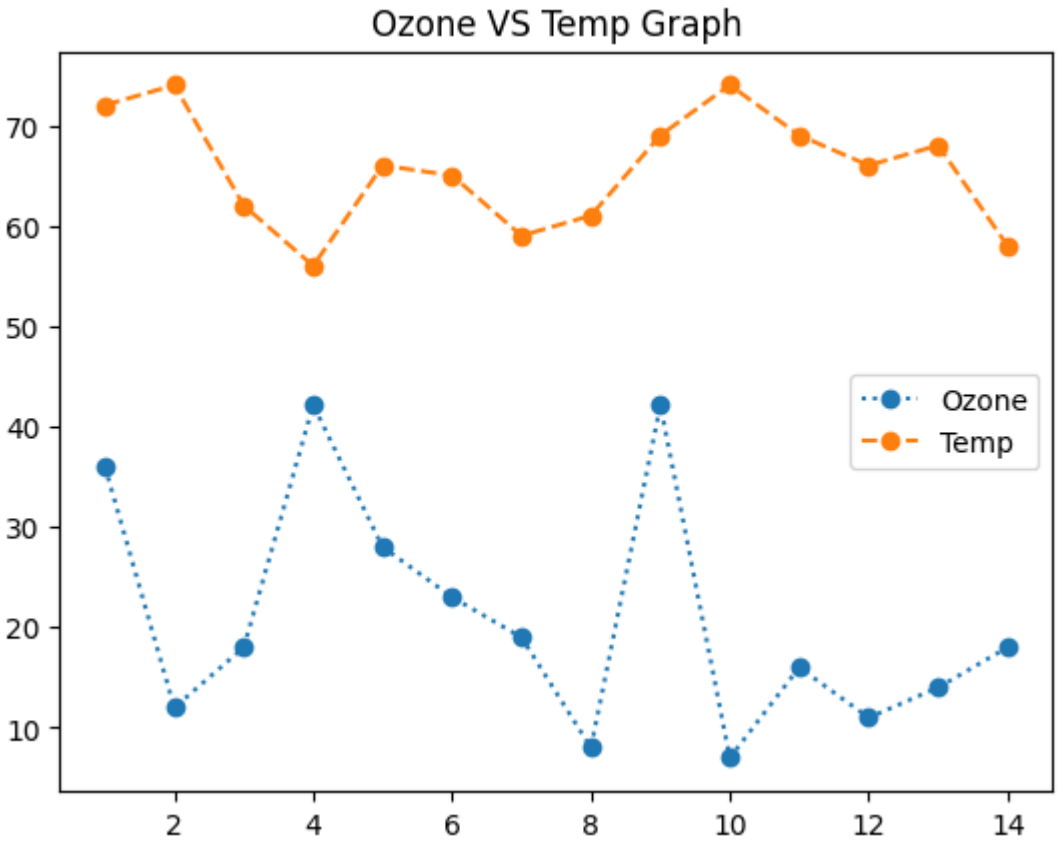
In [66]:
```
import matplotlib.pyplot as plt
```

In [71]:
```
plt.scatter(x = tej['Wind'] , y = tej['Temp'] , c='blue' )
plt.legend("Wind")
plt.xlabel("Wind")
plt.ylabel("Temp")
plt.title("Wind VS Temperatute")
plt.show()
```

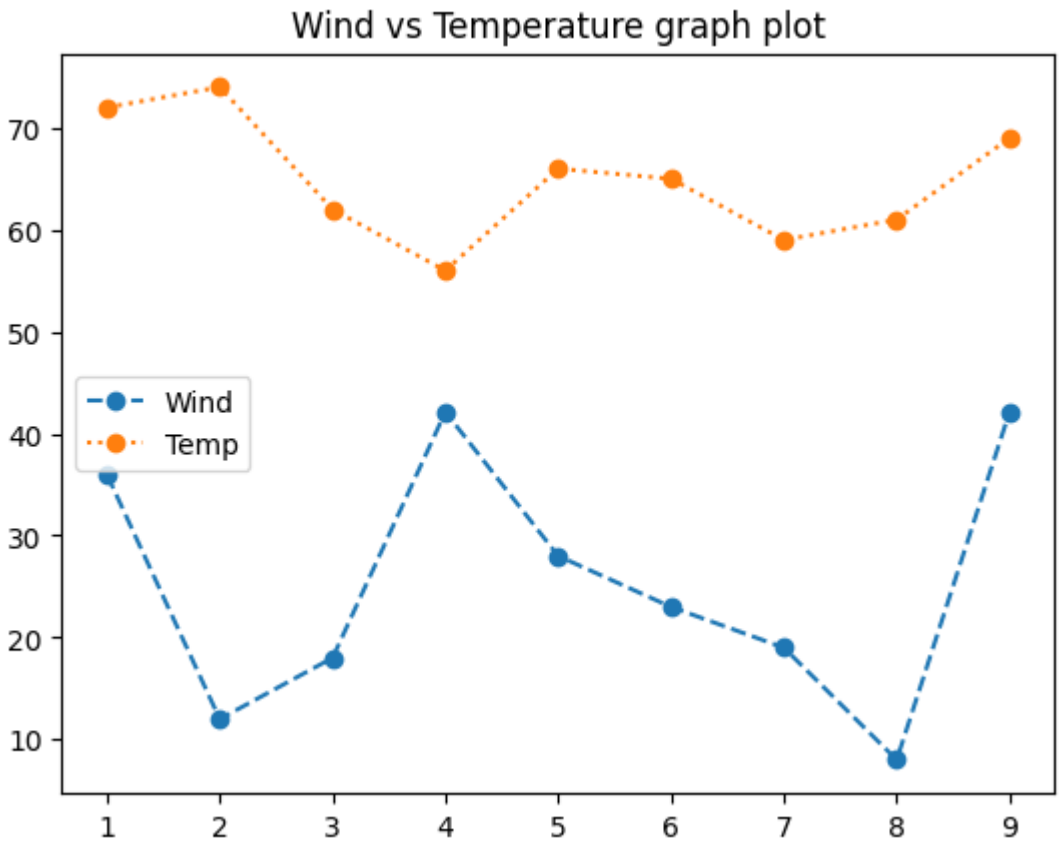


2) Line Graph

In [76]:
```
h = tej.iloc[1:15 , 0]
v = tej.iloc[1:15 , 3]
plt.plot(h , label="Ozone" ,  marker="o" , linestyle="dotted" )
plt.plot(v , label = "Temp" , marker="o" , linestyle="dashed")
plt.title("Ozone VS Temp Graph")
plt.legend()
plt.show()
```



In [91]:
```
m = tej.iloc[1:10 , 0]
n = tej.iloc[1:10 , 3]
plt.plot(m , label='Wind' , marker='o' , linestyle='dashed')
plt.plot(n , label='Temp' , marker='o' , linestyle='dotted')
plt.title("Wind vs Temperature graph plot")
plt.legend()
plt.show()
```



3) Pie Chart

In [92]:
```
tej
```

Out[92]:

	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	1
2	12.00000	149.000000	12.6	74	5	3	0
3	18.00000	313.000000	11.5	62	5	4	2
4	42.12931	185.931507	14.3	56	5	5	0
...	...	...	...	...	...	...	...
148	30.00000	193.000000	6.9	70	9	26	0
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	0

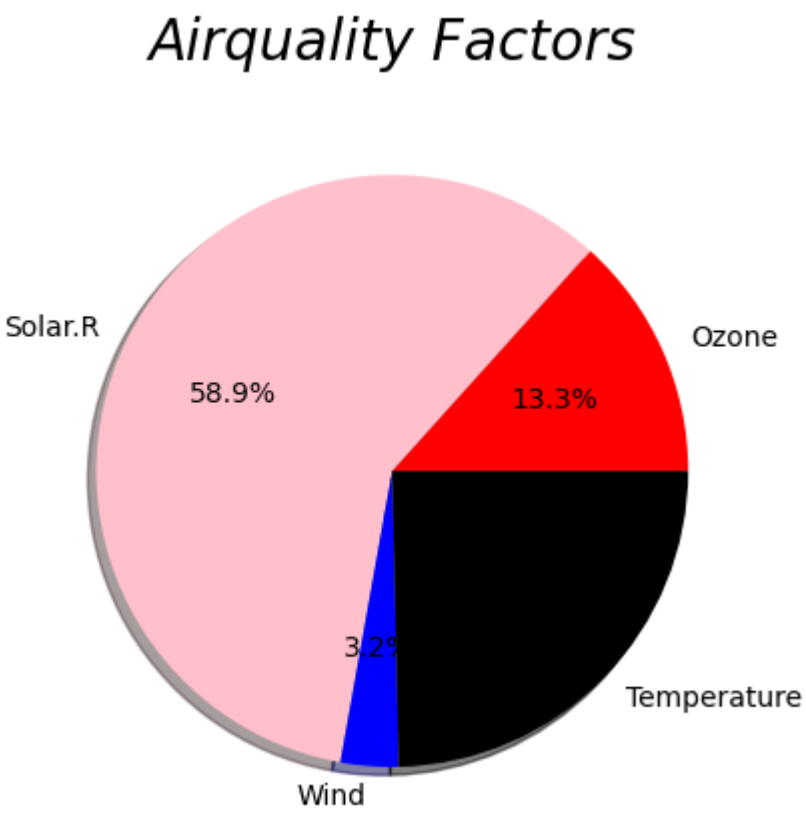
153 rows × 7 columns

In [110…

```
labels = ['Ozone' , 'Solar.R' , 'Wind' , 'Temperature']
sizes = [tej['Ozone'].mean() , tej['Solar.R'].mean() , tej['Wind'].mean() , tej['Te
colors = ['red' , 'pink' , 'blue' , 'black']
textprops = {'fontsize' : 15}
plt.pie(sizes , labels=labels , colors=colors , autopct='%1.1f%%' , shadow=True)
plt.title("Airquality Factors" , fontsize=20 , style='italic' , pad=15)
```

Out[110]:

```
Text(0.5, 1.0, 'Airquality Factors')
```

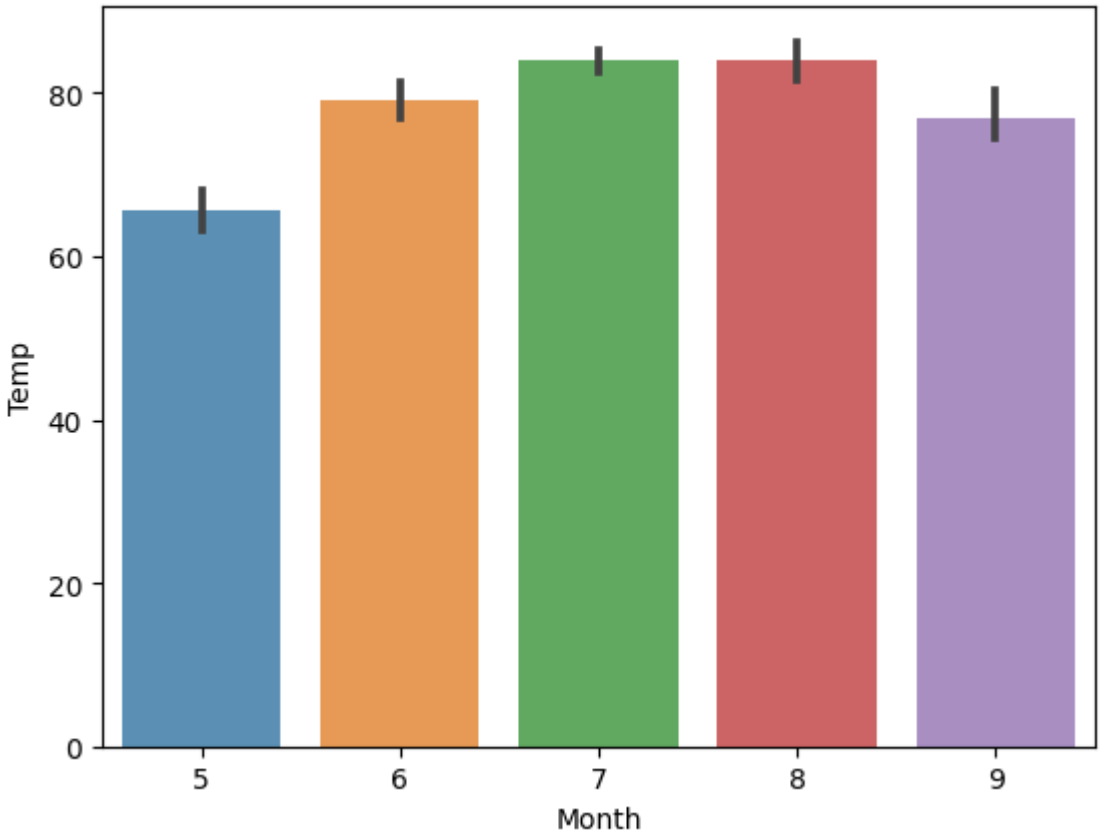


4) Barplot

In [116…
```
import seaborn as sns
```

In [125…
```
sns.barplot(x = tej['Month'] , y = tej['Temp'], alpha=0.8)
```

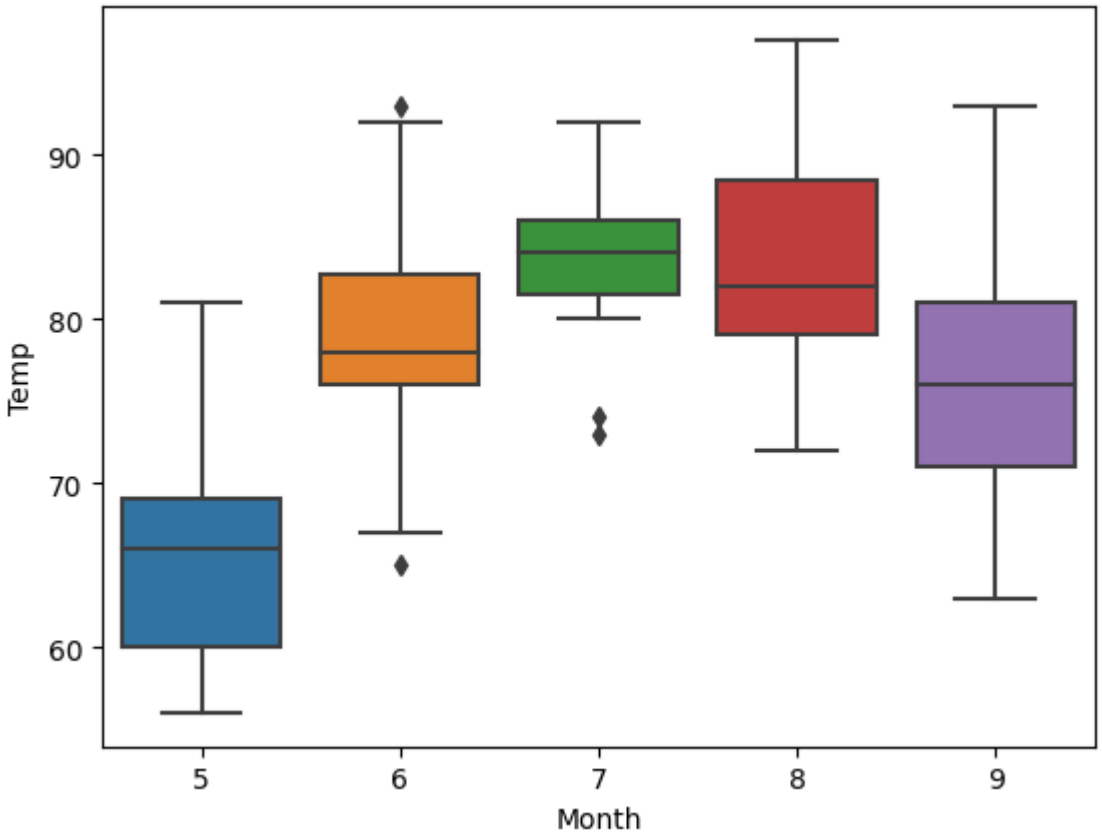
Out[125]:
```
<Axes: xlabel='Month', ylabel='Temp'>
```



5) Box Plot

In [131…
```
sns.boxplot(x=tej['Month'] , y=tej['Temp'])
```

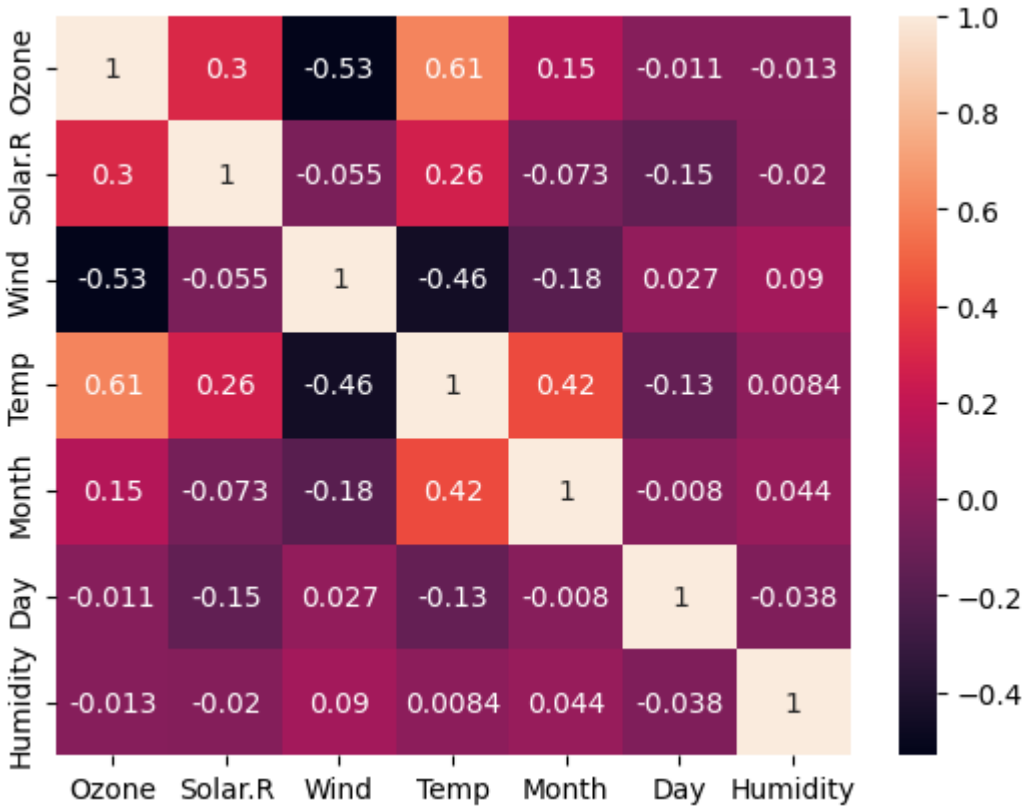
Out[131]: <Axes: xlabel='Month', ylabel='Temp'>



6)HeatMap

In [126…
```
corr = tej.corr()
sns.heatmap(corr , annot=True)
```

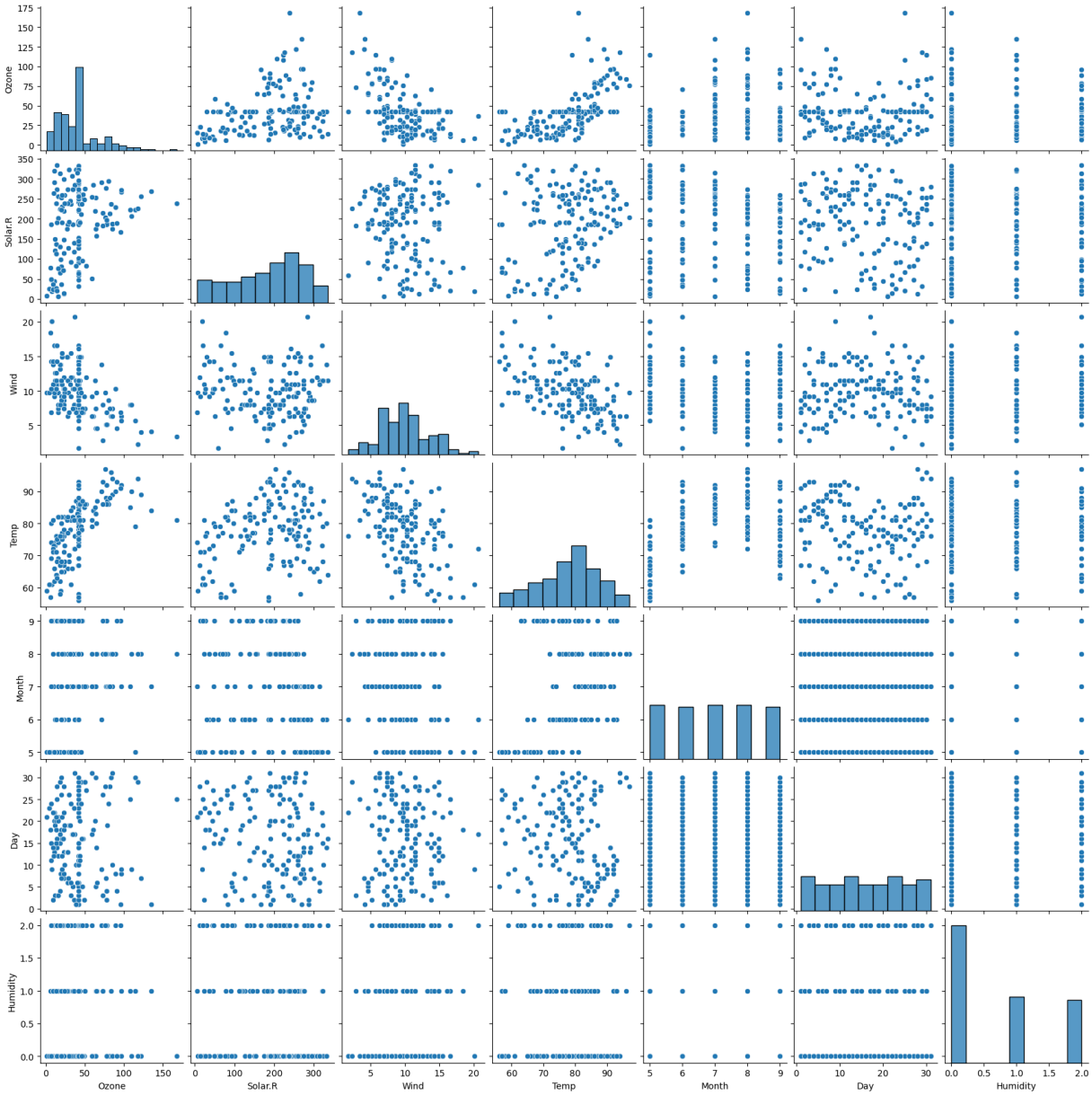
Out[126]: <Axes: >



7) Pair Plot

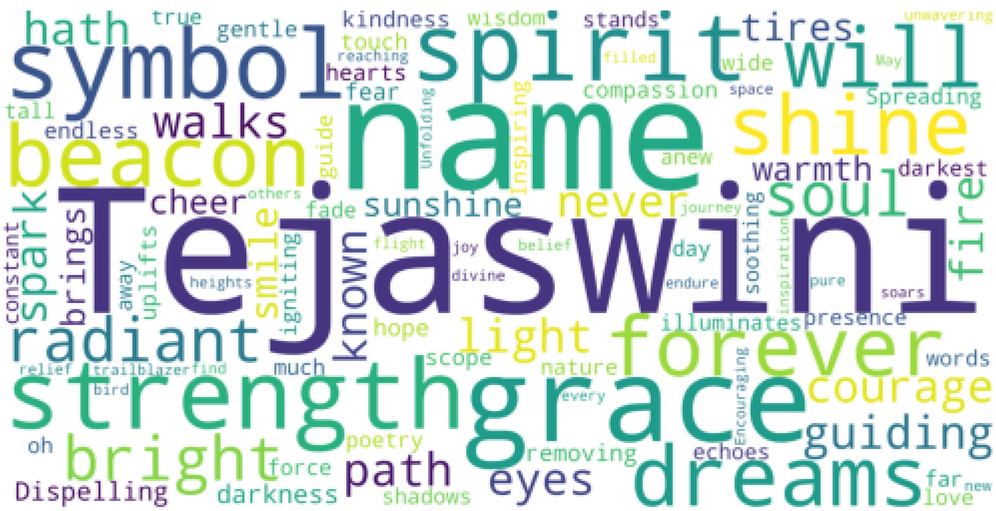
In [128…
```
sns.pairplot(tej)
```

Out[128]: <seaborn.axisgrid.PairGrid at 0x25621577ad0>



8) Word Cloud

In [12]:
```
import numpy as np
import matplotlib.pyplot as plt
from wordcloud import WordCloud, ImageColorGenerator, STOPWORDS
text = open("C:\\Users\\Shree\\Desktop\\tej_wordcloud.txt").read()
wrd_cld = WordCloud(background_color='white' , height=2225 , width=4450)
wrd_cld.generate_from_text(text)
plt.imshow(wrd_cld)
plt.axis('off')
plt.show()
```


Tejaswini
name
strength
grace
symbol
spirit
will
beacon
shine
dreams
forever
radiant
soul
bright
guiding
courage
light
path
eyes
walks
hath
tires
never
sunshine
cheer
warmth
known
smile
spark
brings
fire
Dispelling
darkness
shadows
poetry
force
removing
echoes
hope
scope
nature
much
illuminates
presence
words
compassion
Spreading
kindness
wisdom
stands
gentle
true
touch
hearts
fear
tall
endless
darkest
anew
wide
fade
day
constant
uplifts
away
igniting
guide
Inspiring
soothing
oh
far
love
new
unwavering

In [ ]: