Problem Statement:

Visualize the data using Python libraries matplotlib,seaborn by plotting the graphs for assignment no.2 and 3.

Import The libraries ¶

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
A=pd.read_csv("airquality3.csv",delimiter=',')
```

In [3]:
```
A
```

Out[3]:

	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	high
3	4	18.0	313.0	11.5	62	5	4	high
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	high
150	151	14.0	191.0	14.3	75	9	28	high
151	152	18.0	131.0	8.0	76	9	29	high
152	153	20.0	223.0	11.5	68	9	30	high

153 rows × 8 columns

In [4]:
```
A.head(10)
```

Out[4]:

	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	high
3	4	18.0	313.0	11.5	62	5	4	high
4	5	NaN	NaN	14.3	56	5	5	high
5	6	28.0	NaN	14.9	66	5	6	high
6	7	23.0	299.0	8.6	65	5	7	low
7	8	19.0	99.0	13.8	59	5	8	low
8	9	8.0	19.0	20.1	61	5	9	low
9	10	NaN	194.0	8.6	69	5	10	low

In [5]:
```
A.tail(10)
```

Out[5]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day	humidity
143	144	13.0	238.0	12.6	64	9	21	high
144	145	23.0	14.0	9.2	71	9	22	high
145	146	36.0	139.0	10.3	81	9	23	high
146	147	7.0	49.0	10.3	69	9	24	high
147	148	14.0	20.0	16.6	63	9	25	NaN
148	149	30.0	193.0	6.9	70	9	26	high
149	150	NaN	145.0	13.2	77	9	27	high
150	151	14.0	191.0	14.3	75	9	28	high
151	152	18.0	131.0	8.0	76	9	29	high
152	153	20.0	223.0	11.5	68	9	30	high

In [6]:
```
A.shape
```

Out[6]:
```
(153, 8)
```

In [7]:
```
A.isnull().sum()
```

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

Replacing the null values

In [8]:
```
A["Ozone"]=A["Ozone"].fillna(A["Ozone"].mean())
A["Solar.R"]=A["Solar.R"].fillna(A["Solar.R"].mean())
```

In [9]:
```
A
```

Out[9]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day	humidity
0	1	41.00000	190.000000	7.4	67	5	1	high
1	2	36.00000	118.000000	8.0	72	5	2	high
2	3	12.00000	149.000000	12.6	74	5	3	high
3	4	18.00000	313.000000	11.5	62	5	4	high
4	5	42.12931	185.931507	14.3	56	5	5	high
...	...	...	...	...	...	...	...	...
148	149	30.00000	193.000000	6.9	70	9	26	high
149	150	42.12931	145.000000	13.2	77	9	27	high
150	151	14.00000	191.000000	14.3	75	9	28	high
151	152	18.00000	131.000000	8.0	76	9	29	high
152	153	20.00000	223.000000	11.5	68	9	30	high

153 rows × 8 columns

In [10]:
```
A.drop('Unnamed: 0',axis=1,inplace=True)
```

In [11]:
```
A
```

Out[11]:

	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	high
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	high
149	42.12931	145.000000	13.2	77	9	27	high
150	14.00000	191.000000	14.3	75	9	28	high
151	18.00000	131.000000	8.0	76	9	29	high
152	20.00000	223.000000	11.5	68	9	30	high

153 rows × 7 columns

1.Pie chart

In [12]:
```
plt.pie
```

Out[12]:
```
<function matplotlib.pyplot.pie(x, explode=None, labels=None, colors=None, autopct=None, pctdistance=0.6, shadow=False, labeldistance=1.1, startangle=0, radius=1, counterclock=True, wedgeprops=None, textprops=None, center=(0, 0), frame=False, rotatelabels=False, *, normalize=True, data=None)>
```

In [13]:
```
labels=['Ozone','Solar.R','Wind','Temp']
```

In [14]:
```
A.columns
```

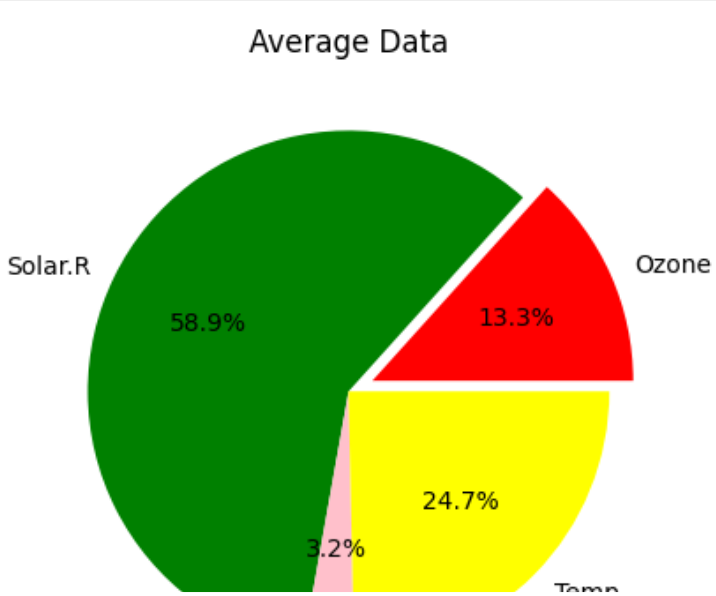
Out[14]:
```
Index(['Ozone', 'Solar.R', 'Wind', 'Temp', 'Month', 'Day', 'humidity'], dtype='object')
```

In [15]:
```
sizes=[A['Ozone'].mean(),A['Solar.R'].mean(),A['Wind'].mean(),A['Temp'].mean()]
```

In [16]:
```
colors=['red','green','pink','yellow']
```

In [17]:
```
explode=(0.1,0,0,0)
```

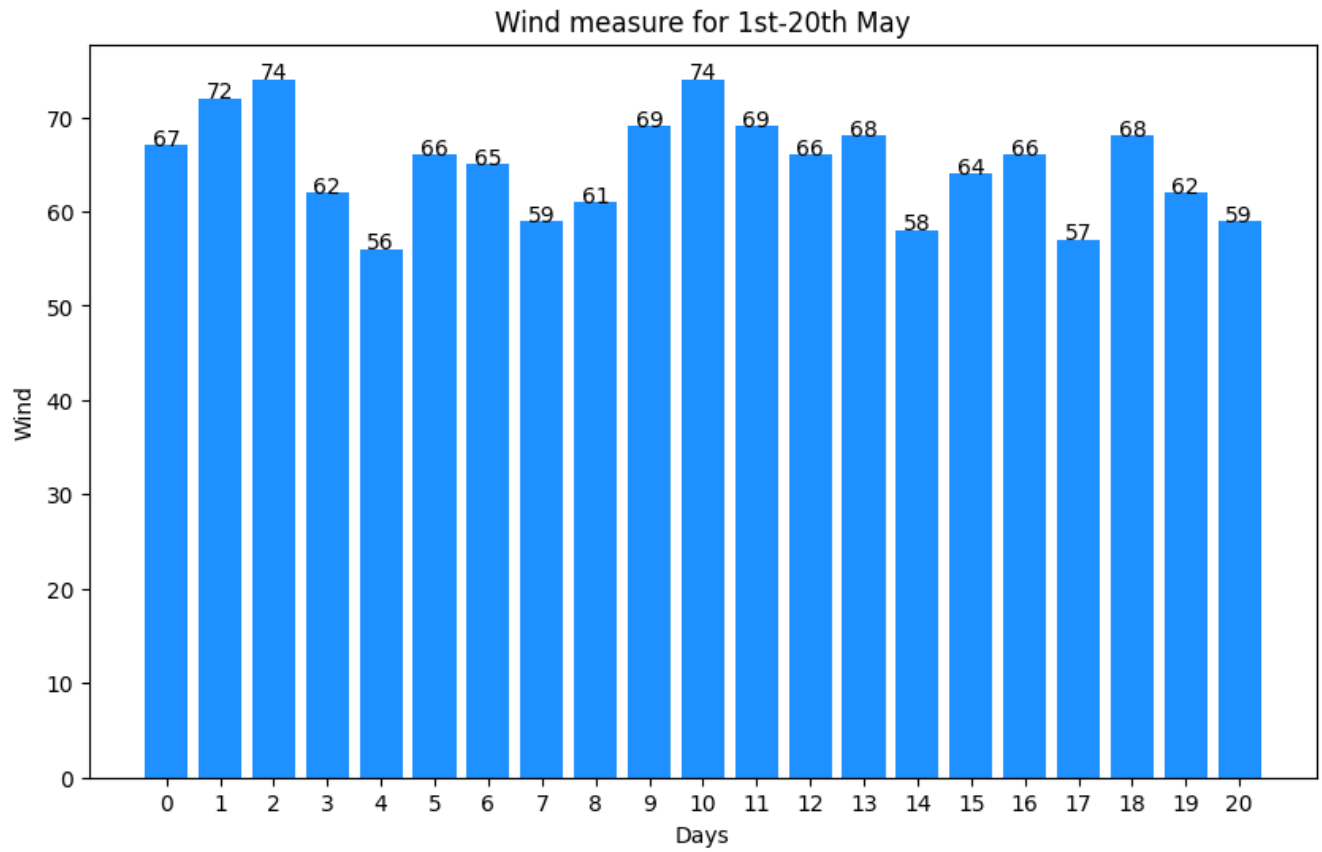
In [18]:
```
plt.pie(sizes,explode=explode,labeldistance=1.1,labels=labels,
colors=colors,autopct='%0.1f%%')
plt.title('Average Data')
plt.show()
```



2.Bar plot

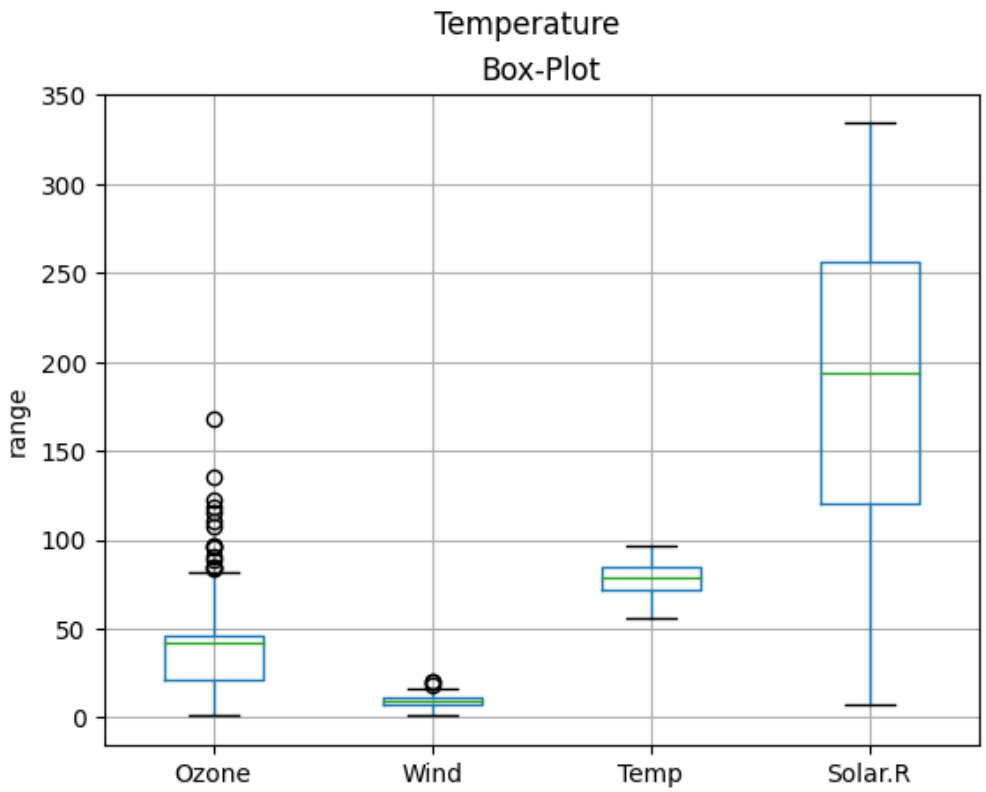
In [19]:
```
def addlabels():
    for i in range(len(A.iloc[0:21,3])):
        plt.text(i,A.iloc[0:21,3][i],A.iloc[0:21,3][i],ha='center')

y=np.arange(len(A.iloc[0:21,3]))
plt.figure(figsize=(10,6))
addlabels()
plt.bar(x=y,height=A.iloc[0:21,3],tick_label=y,color="dodgerblue")
plt.title('Wind measure for 1st-20th May')
plt.xlabel('Days')
plt.ylabel('Wind')
plt.show()
```



3.Box_plot

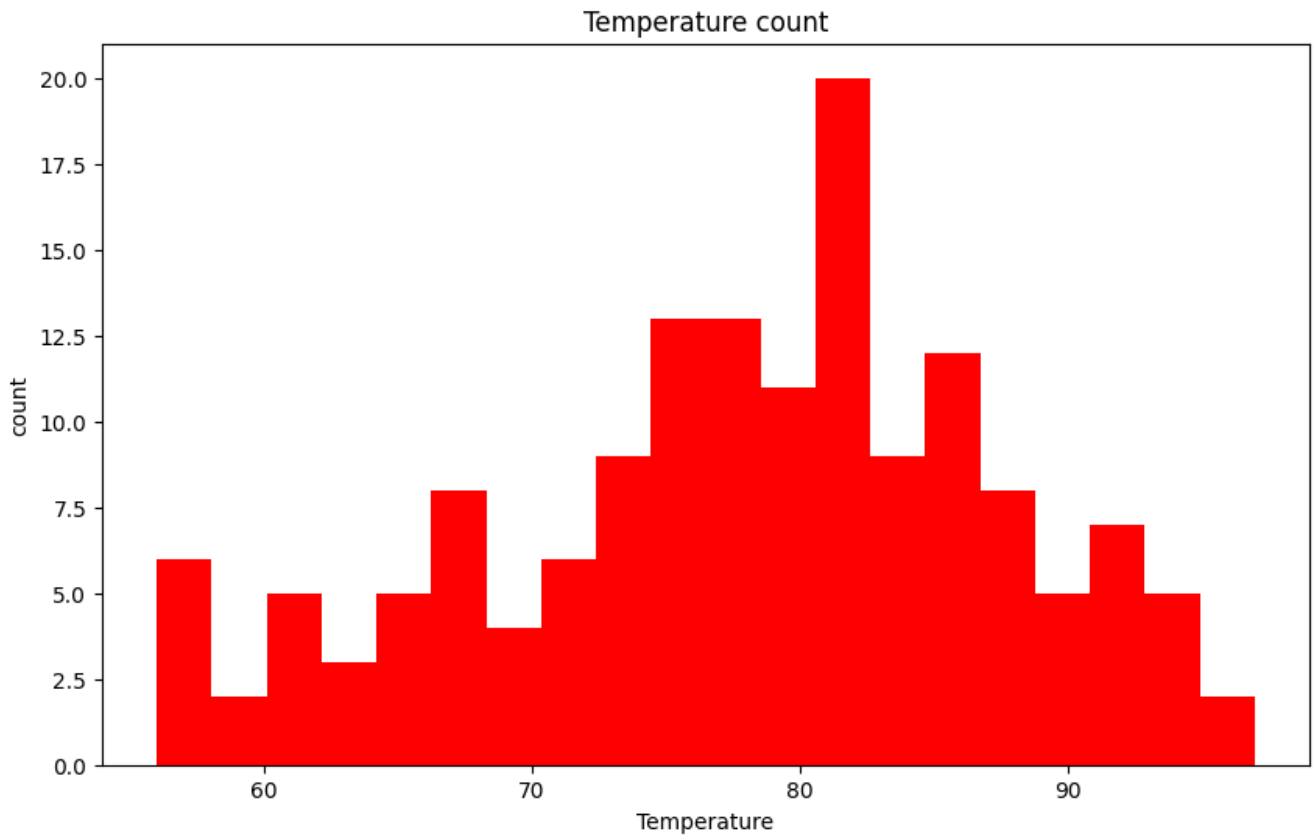
In [28]:
```
A.boxplot(column=['Ozone','Wind','Temp','Solar.R'],grid=True)
plt.title('Box-Plot')
plt.suptitle('Temperature')
plt.ylabel('range')
plt.savefig('plot3.png')
```



4.Histogram

In [21]:
```
plt.figure(figsize=(10,6))
plt.hist(x=A['Temp'],bins=20,color="Red",label="Temp")
plt.title("Temperature count")
plt.xlabel("Temperature")
plt.ylabel("count")
plt.plot()
```

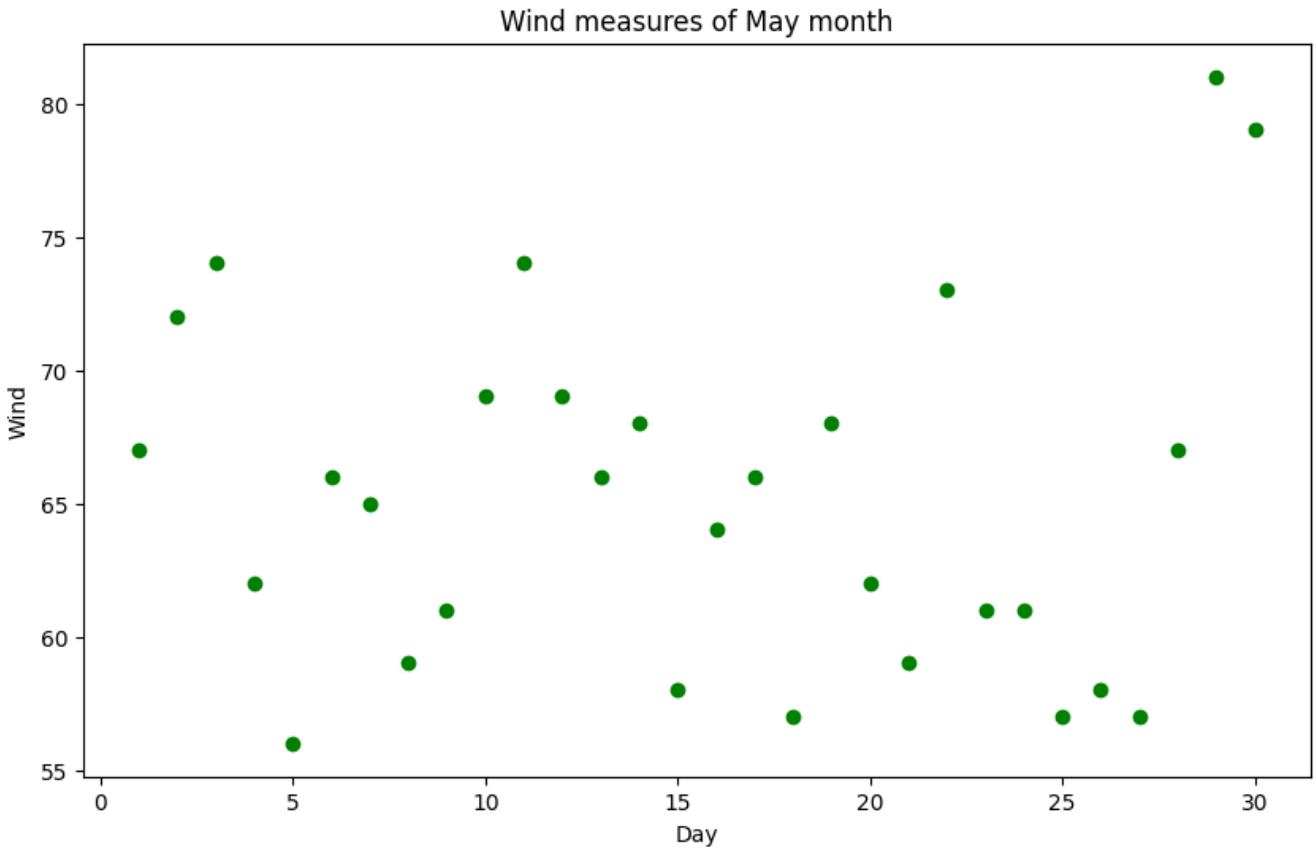
Out[21]: []



5.Scatter Plot

In [22]:
```
plt.figure(figsize=(10,6))
plt.scatter(A.iloc[0:30,5],A.iloc[0:30,3],color="green")
plt.title("Wind measures of May month")
plt.xlabel("Day")
plt.ylabel("Wind")
```

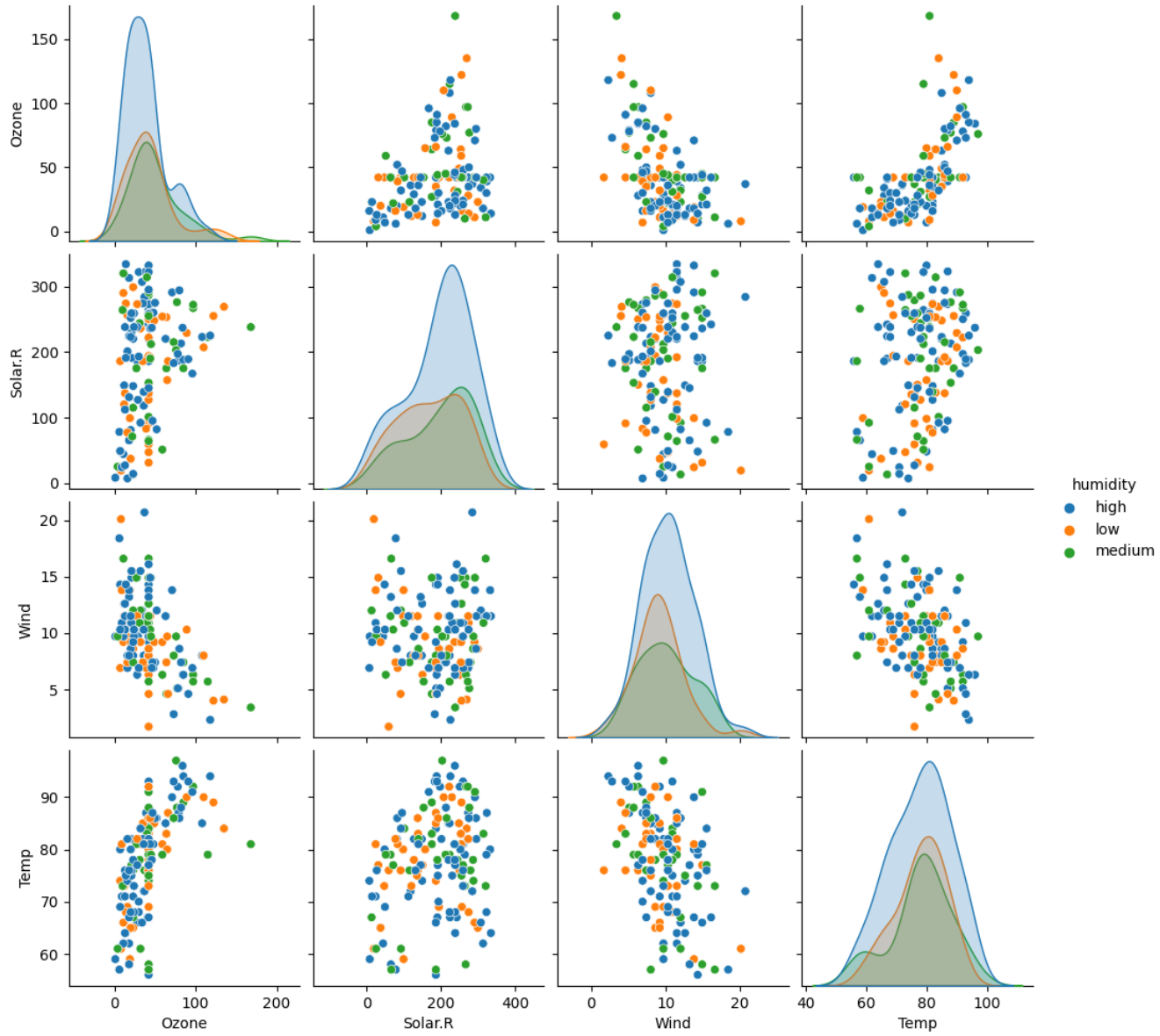
Out[22]: Text(0, 0.5, 'Wind')



6.Pair Plot

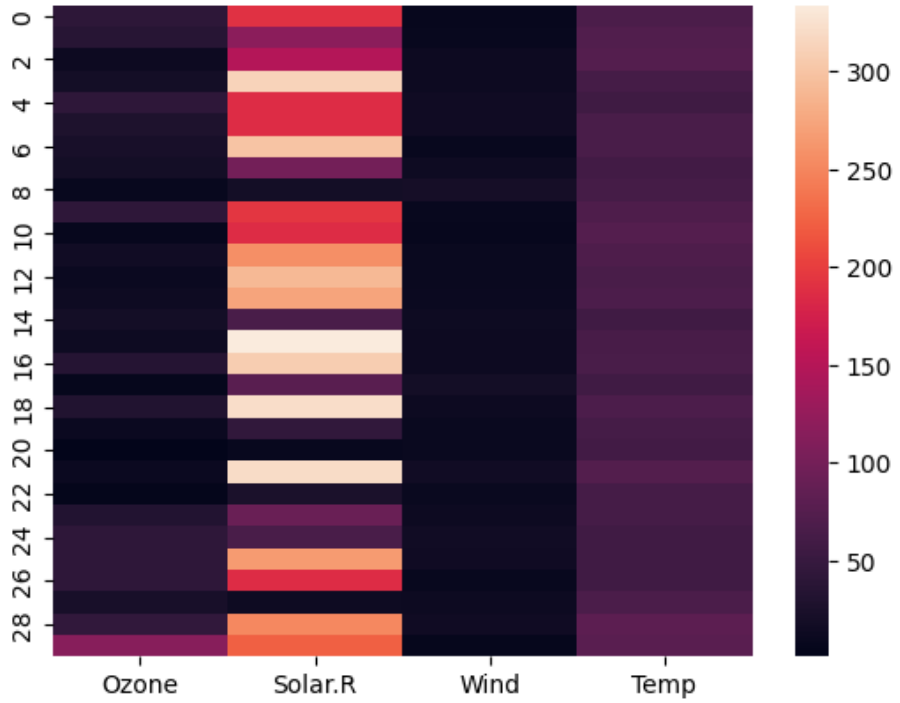
In [23]:
```
import seaborn

seaborn.pairplot(A.iloc[:,[0,1,2,3,6]],hue='humidity')
plt.show()
```



7.Heatmap

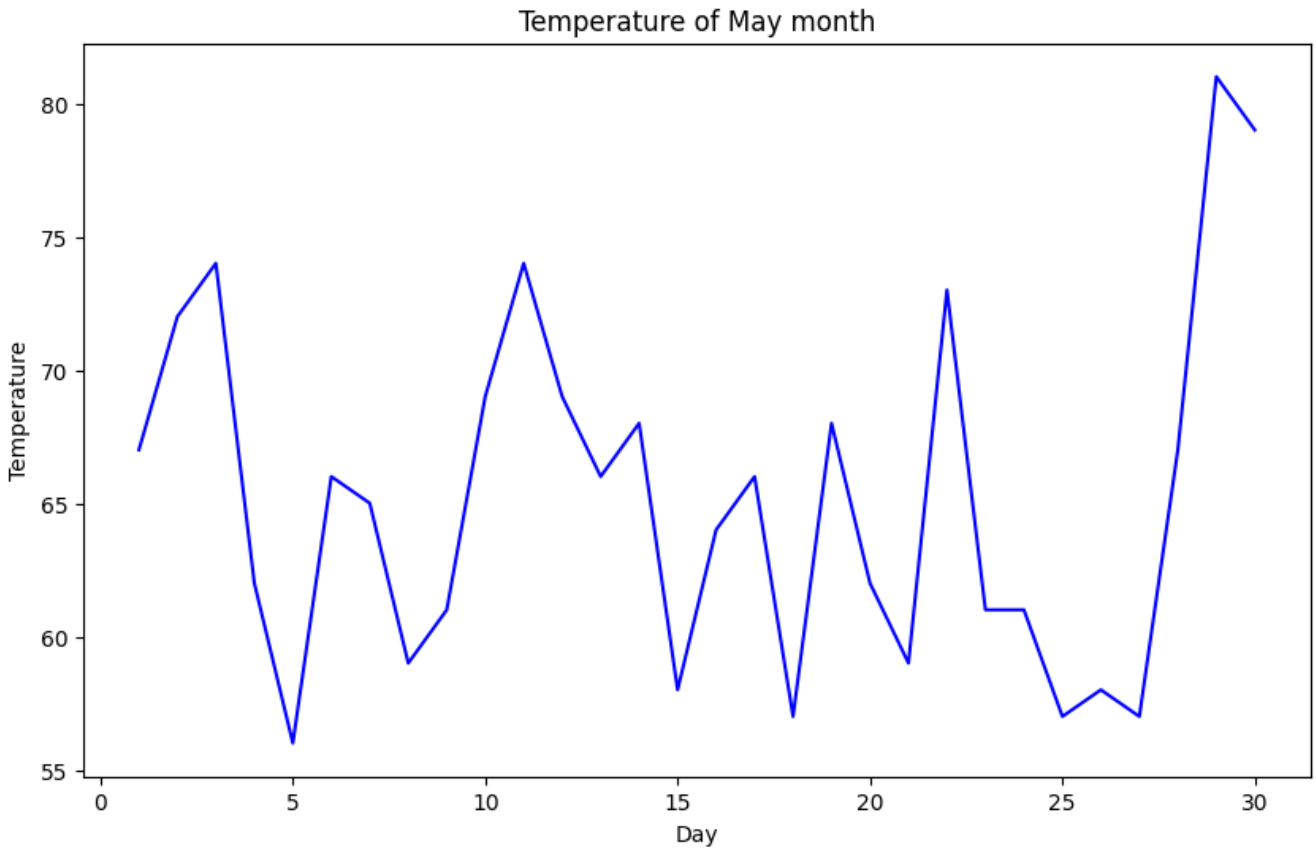
In [24]:
```
seaborn.heatmap(data=A.iloc[0:30,[0,1,2,3]])
plt.show()
```



8.Line graph

In [25]:
```
plt.figure(figsize=(10,6))
plt.plot(A.iloc[0:30,5],A.iloc[0:30,3],color="blue")
plt.title('Temperature of May month')
plt.xlabel("Day")
plt.ylabel('Temperature')
```

Out[25]: Text(0, 0.5, 'Temperature')

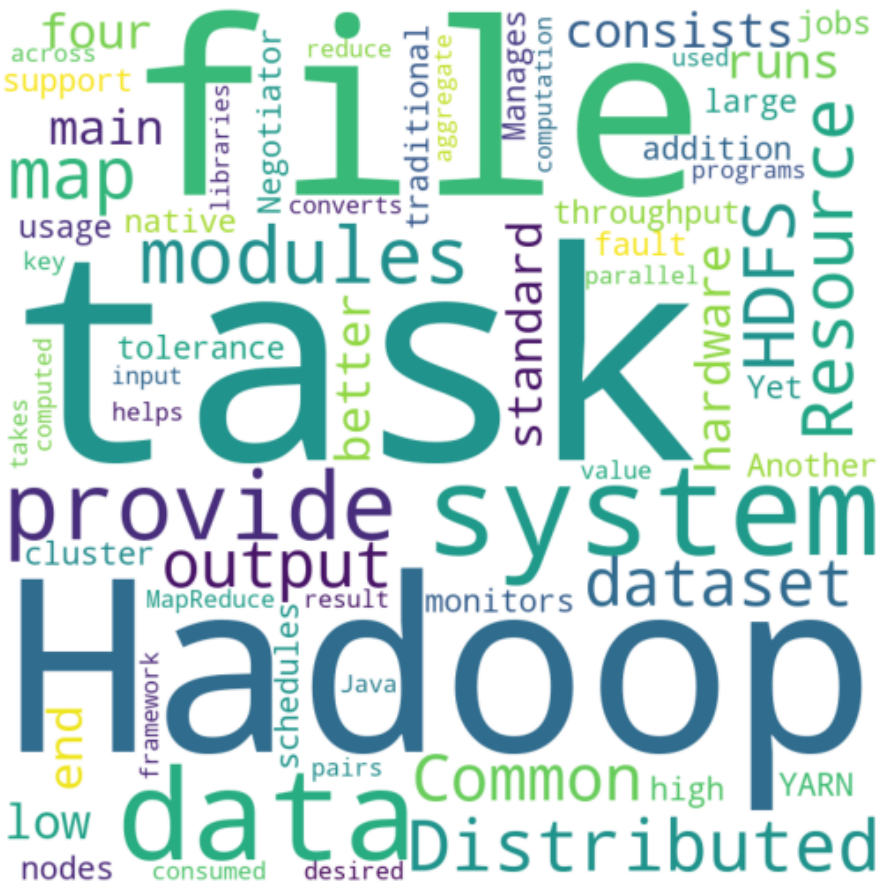


9.Word Cloud

In [29]:
```
from wordcloud import WordCloud,STOPWORDS
stopwords=set(STOPWORDS)
text="""Hadoop consists of four main modules:

Hadoop Distributed File System (HDFS) – A distributed file system that runs on standard or low-end
Yet Another Resource Negotiator (YARN) – Manages and monitors cluster nodes and resource usage. It
MapReduce – A framework that helps programs do the parallel computation on data. The map task takes
Hadoop Common – Provides common Java libraries that can be used across all modules.
"""

wordcloud=WordCloud(width=800,height=800,
                    background_color='white',
                    stopwords=stopwords,
                    min_font_size=10).generate(text)
plt.figure(figsize=(5,5))
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
plt.show()
```



In [ ]:
```
```