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

```
df = pd.read_csv("/content/AirQuality.csv")
```

```
df
```

```
     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 df | View recommended plots | New interactive sheet

```
df=df.drop(columns='Unnamed: 0')
```

```
df
```

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

153 rows × 7 columns
```

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

```
df.isnull().sum()
```

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

dtype: int64
```

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

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

dtype: int64
```

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

```
        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 df View recommended plots New interactive sheet

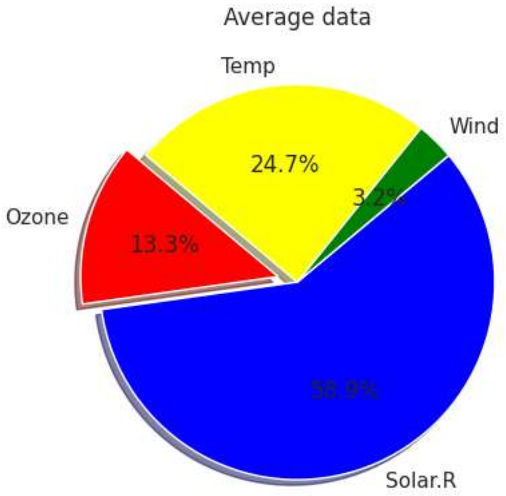
1. PIE Chart

```
import matplotlib.pyplot as plt

labels = ['Ozone', 'Solar.R', 'Wind', 'Temp']
sizes = [df['Ozone'].mean(), df['Solar.R'].mean(), df['Wind'].mean(), df['Temp'].mean()]
colors = ['Red', 'Blue', 'Green', 'Yellow']
explode = (0.1, 0, 0, 0)
```

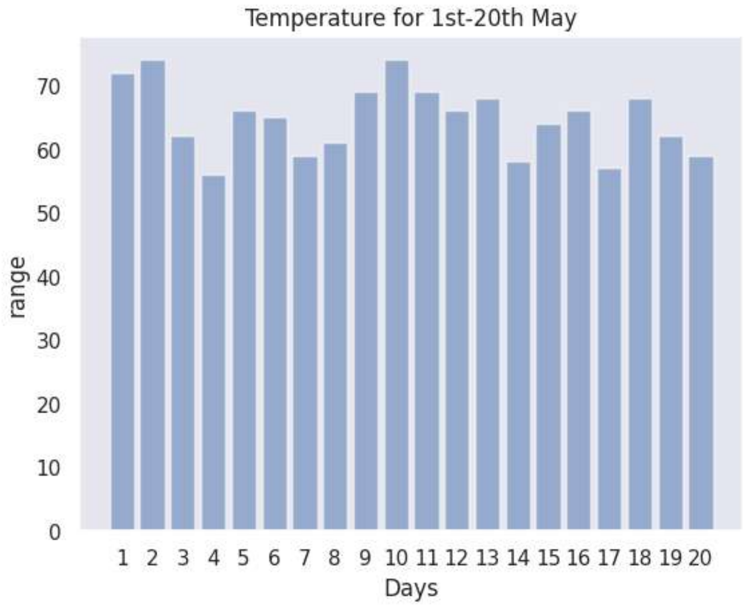
```
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=140)
plt.title('Average data')

plt.show()
```



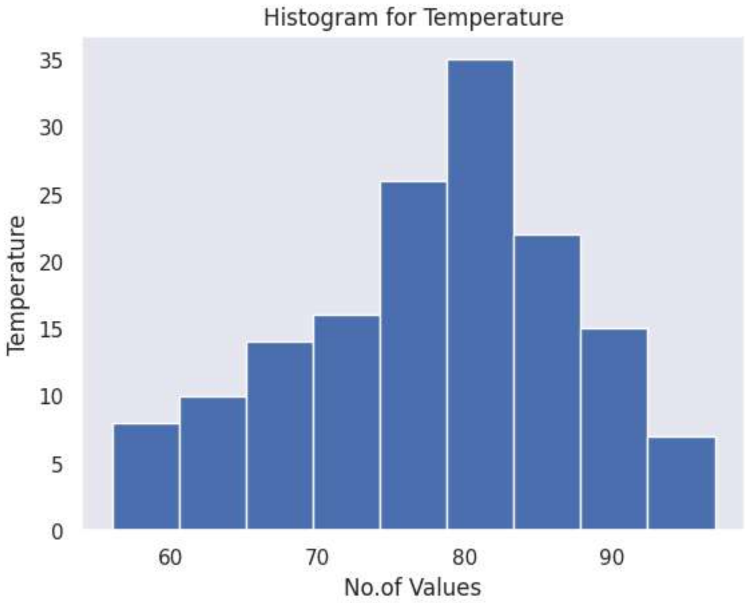
2. BAR Plot

```
h = df.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('Temperature for 1st-20th May')
plt.show()
```



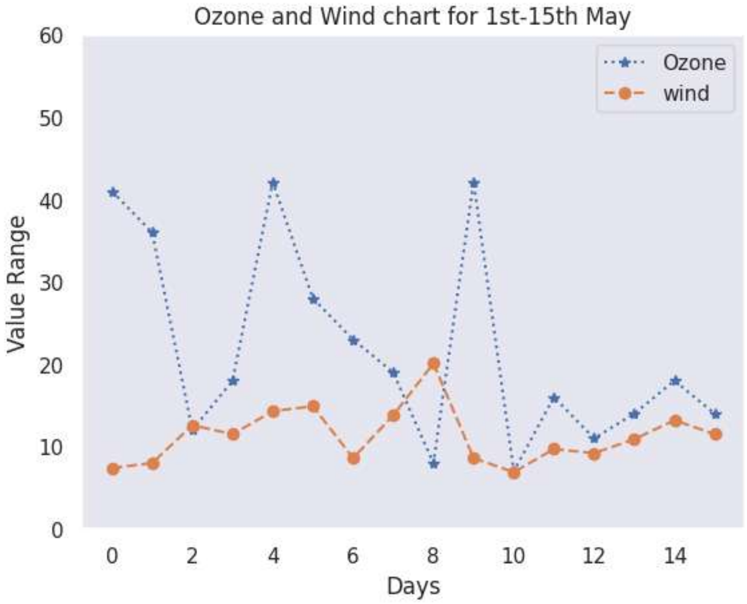
3. HISTOGRAM

```
e=df.iloc[:,3]
plt.hist(e,bins='auto')
plt.title("Histogram for Temperature")
plt.xlabel("No.of Values")
plt.ylabel("Temperature")
plt.show()
plt.show()
```



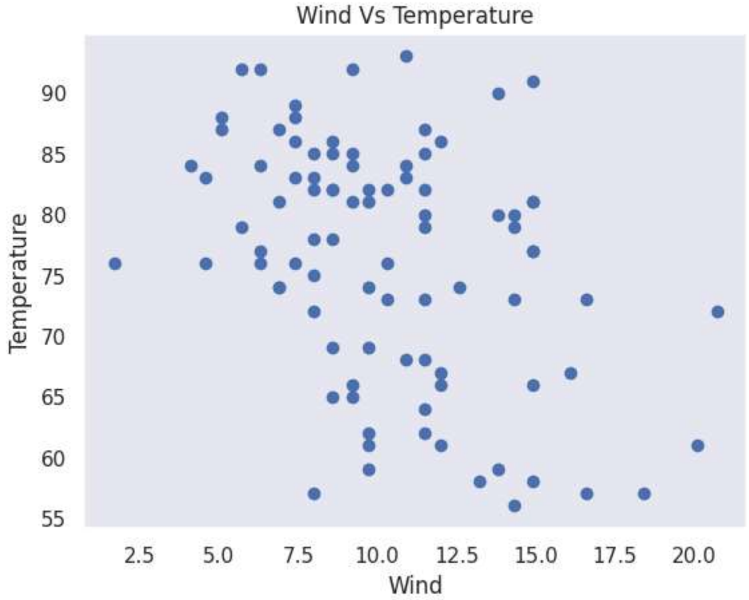
4. LINE Graph

```
l=df.iloc[:16,0]
m=df.iloc[:16,2]
plt.plot(l,label="Ozone",marker = '*',linestyle = "dotted")
plt.plot(m,label = 'wind',marker = 'o',linestyle = "dashed")
plt.ylim(0,60)
plt.legend()
plt.title("Ozone and Wind chart for 1st-15th May")
plt.ylabel("Value Range")
plt.xlabel("Days")
plt.savefig("plot5.png")
```



SCATTER Plot

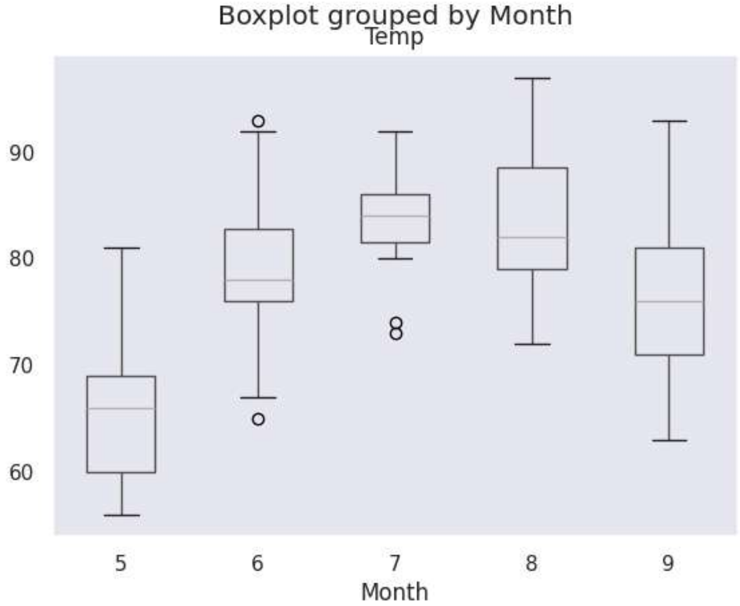
```
s=df.iloc[1:91,2]
t=df.iloc[1:91,3]
plt.scatter(s,t)
plt.title("Wind Vs Temperature")
plt.xlabel("Wind")
plt.ylabel("Temperature")
plt.show()
```



6. BOX Plot

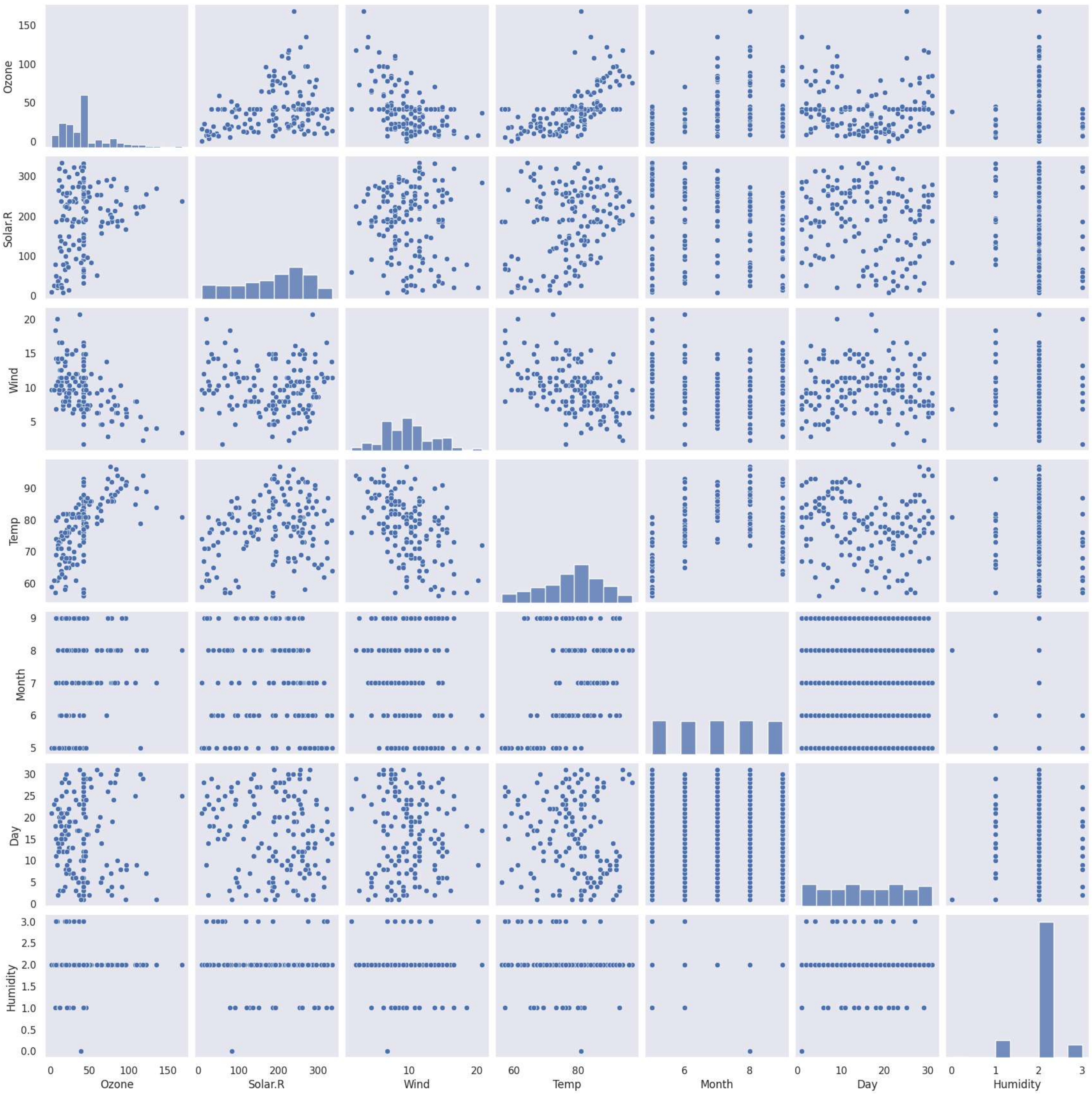
```
df.boxplot(by='Month',column=['Temp'], grid=False)
```

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



7. PAIR Plot

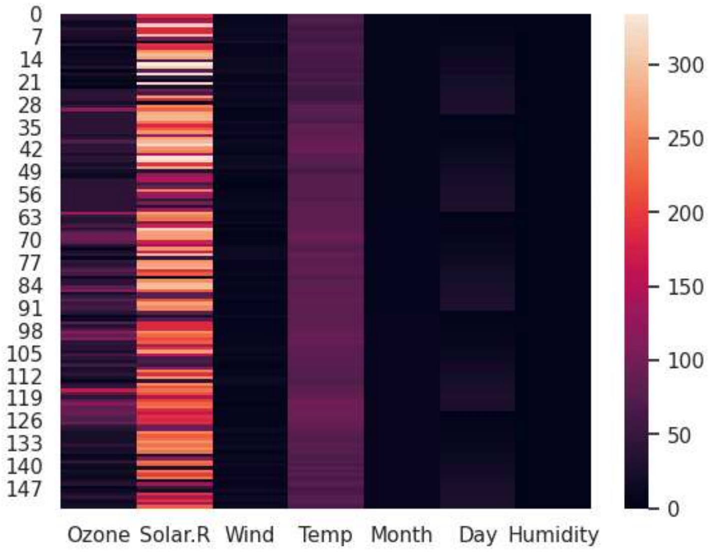
```
sns.set(style = "dark")
sns.pairplot(df)
plt.show()
```


Ozone
0
50
100
150
Solar.R
0
100
200
300
Wind
5
10
15
20
Temp
60
70
80
90
Month
5
6
7
8
9
Day
0
5
10
15
20
25
30
Humidity
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
50
100
150
Ozone
0
100
200
300
Solar.R
10
20
Wind
60
80
Temp
6
8
Month
0
10
20
30
Day
0
1
2
3
Humidity

8. HEATMAP

```
sns.heatmap(df)
```

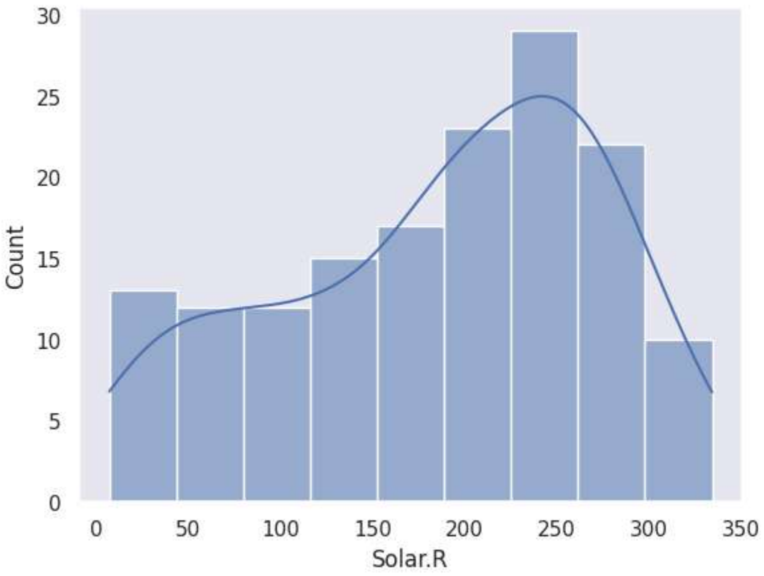
```
<Axes: >
```



9. HISTOGRAM

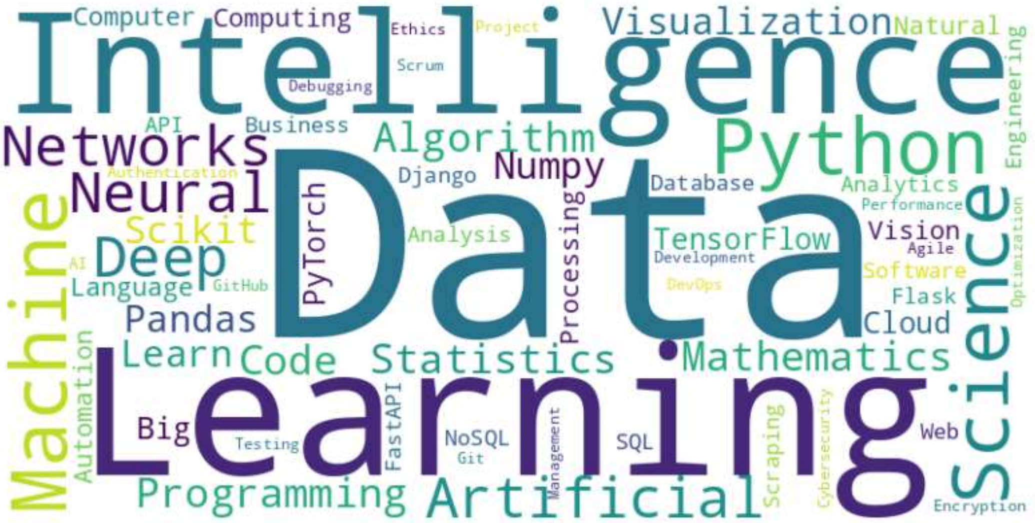
```
sns.histplot(data=df, x="Solar.R", kde=True)
```

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



10. WorldCloud

```
!pip -q install wordcloud
from wordcloud import WordCloud
import matplotlib.pyplot as plt
with open(r"sampletext.txt" ,'r', encoding='utf-8') as file:
  text = file.read()
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```


Intelligence
Computer
Computing
Ethics
Project
Visualization
Natural
Engineering
Scrum
Debugging
API
Business
Networks
Algorithm
Python
Authentication
Django
Numpy
Neural
Database
Analytics
Performance
Machine
Scikit
PyTorch
Data
Analysis
TensorFlow
Vision
Deep
Development
Agile
AI
GitHub
DevOps
Software
Language
Processing
Flask
Optimization
Pandas
Cloud
Science
Learn
Code
Statistics
Mathematics
Automation
Learning
Big
Testing
FastAPI
NoSQL
Git
Management
SQL
Scraping
Cybersecurity
Web
Programming
Artificial
Encryption