In [1]:

```
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
```

In [ ]:

```
# Bar plot
# Box plot
# Histogram

# --- SEABORN ---
# displot
# heat map
# line plot
```

In [2]:

```
#DATASET_PATH = ''

data = pd.read_csv(r'D:\COLLEGE\DSBDA\Assignments\DSBD\Datasets\Iris.csv');
```

In [3]:

```
data
```

Out[3]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica

In [4]:

```
data.shape
```

Out[4]:

```
(150, 6)
```

In [5]:

```
data.columns
```

Out[5]:

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidth
Cm',
       'Species'],
      dtype='object')
```

In [6]:

```
data.dtypes
```

Out[6]:

```
Id                 int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

In [7]:

```
data.head()
```

Out[7]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [8]:

```
data.describe()
```

Out[8]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [9]:

```
data['Species'].value_counts()
```

Out[9]:

```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

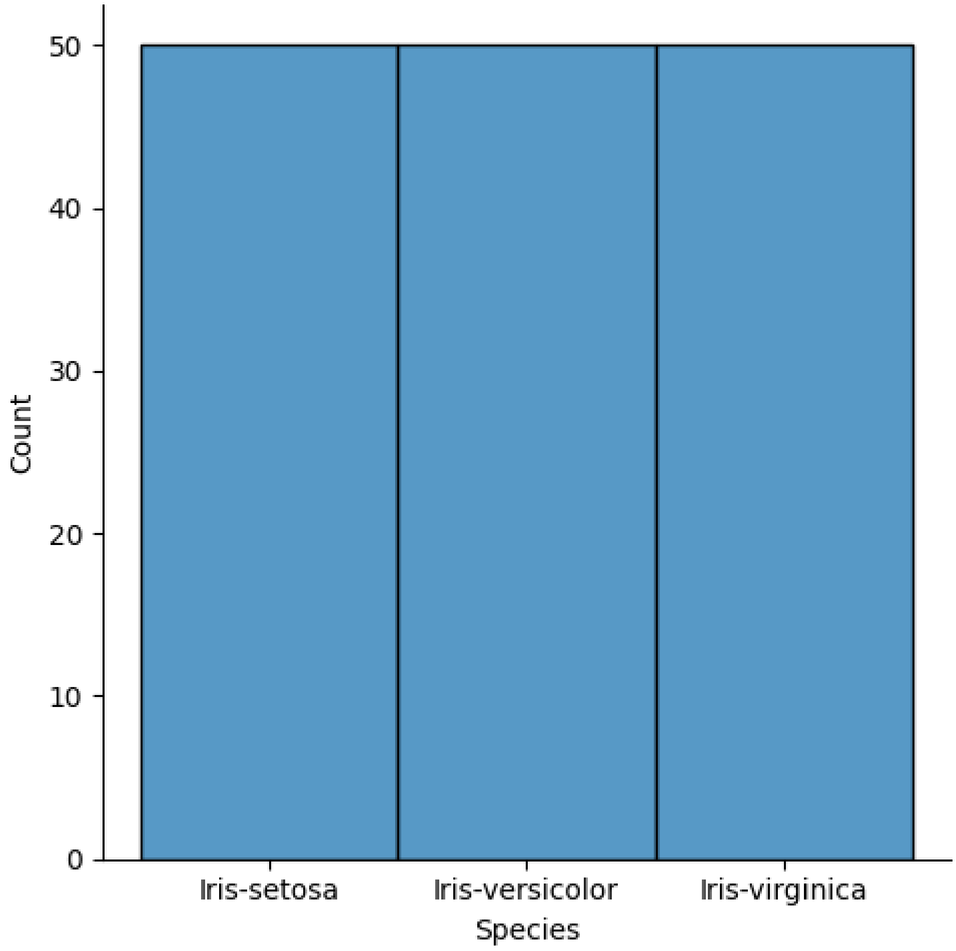
In [10]:

```
data['SepalLengthCm'].value_counts()
```

Out[10]:

```
SepalLengthCm
5.0    10
5.1     9
6.3     9
5.7     8
6.7     8
5.8     7
5.5     7
6.4     7
4.9     6
5.4     6
6.1     6
6.0     6
5.6     6
4.8     5
6.5     5
6.2     4
7.7     4
6.9     4
4.6     4
5.2     4
5.9     3
4.4     3
7.2     3
6.8     3
6.6     2
4.7     2
7.6     1
7.4     1
7.3     1
7.0     1
7.1     1
5.3     1
4.3     1
4.5     1
7.9     1
Name: count, dtype: int64
```

```
In [11]:

# Count Plot

sn.displot(data=data, x = 'Species')

Out[11]:

<seaborn.axisgrid.FacetGrid at 0x2129a2f7850>
```

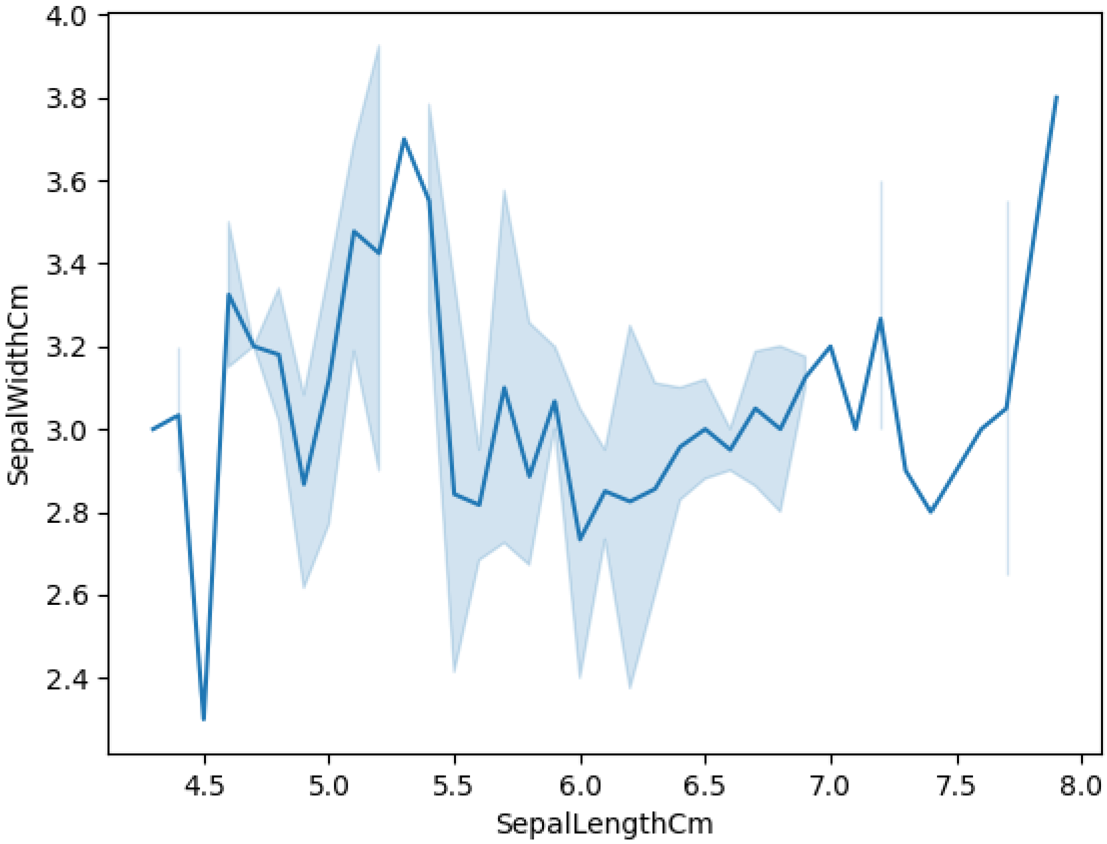


In [12]:

```
# Line Plot

sn.lineplot(data=data, x="SepalLengthCm", y="SepalWidthCm")
```

Out[12]:

```
<Axes: xlabel='SepalLengthCm', ylabel='SepalWidthCm'>
```



In [13]:

```
data = pd.read_csv(r'D:\COLLEGE\DSBDA\Assignments\DSBD\Datasets\forestfires.csv')
```

In [14]:

```
data
```

Out[14]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
512	4	3	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	2	4	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	7	4	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	1	4	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	6	3	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

517 rows × 13 columns

In [15]:

```
data.shape
```

Out[15]:

```
(517, 13)
```

In [16]:

```
data.dtypes
```

Out[16]:

```
X           int64
Y           int64
month      object
day        object
FFMC      float64
DMC       float64
DC        float64
ISI       float64
temp      float64
RH          int64
wind      float64
rain      float64
area      float64
dtype: object
```

In [17]:

```
monthly_data = data['month'].value_counts()
monthly_data
```

Out[17]:

```
month
aug    184
sep    172
mar     54
jul     32
feb     20
jun     17
oct     15
apr      9
dec      9
jan      2
may      2
nov      1
Name: count, dtype: int64
```

Type *Markdown* and LaTeX: α^2

In [18]:

```
monthly_data = data.groupby('month').size().sort_values(ascending=False)
monthly_data
```

Out[18]:

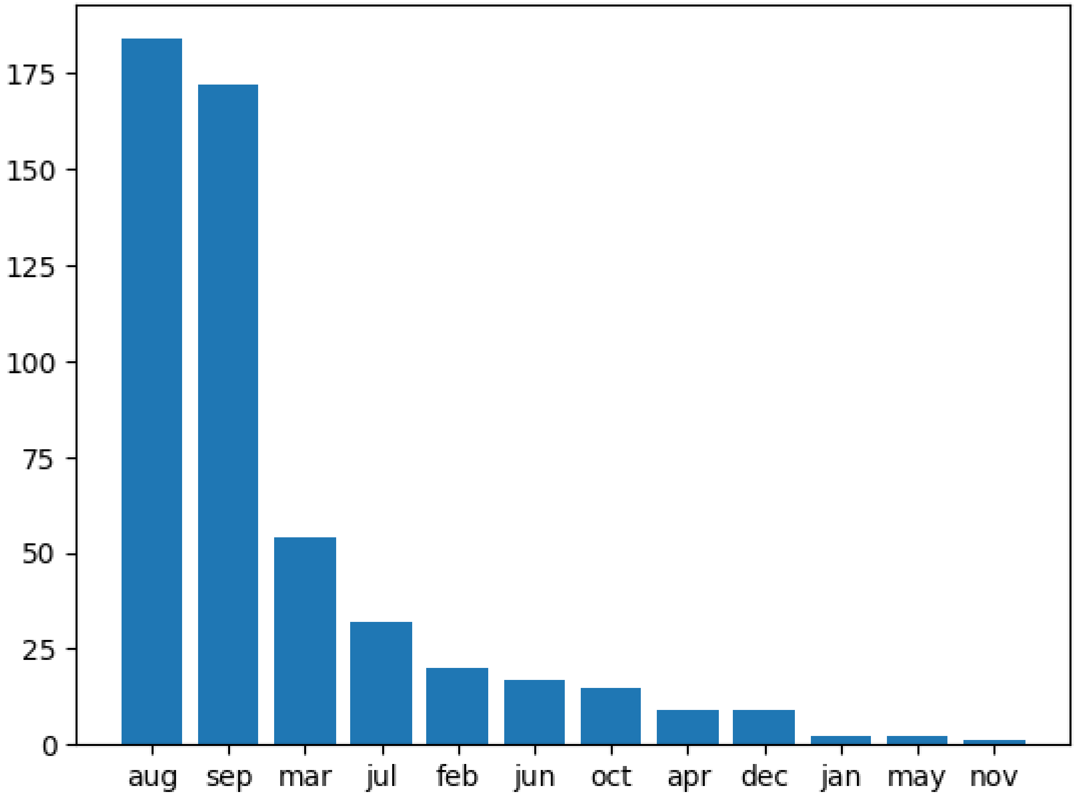
```
month
aug    184
sep    172
mar     54
jul     32
feb     20
jun     17
oct     15
apr      9
dec      9
jan      2
may      2
nov      1
dtype: int64
```

In [19]:

```
# Bar plot

plt.bar(monthly_data.index, monthly_data.values)
```

Out[19]:

```
<BarContainer object of 12 artists>
```



In [20]:

```
day_data = data['day'].value_counts()
day_data
```

Out[20]:

```
day
sun    95
fri    85
sat    84
mon    74
tue    64
thu    61
wed    54
Name: count, dtype: int64
```

In [21]:

```
day_data.index
```

Out[21]:

```
Index(['sun', 'fri', 'sat', 'mon', 'tue', 'thu', 'wed'], dtype='object', n
ame='day')
```

In [22]:

```
day_data.values
```

Out[22]:

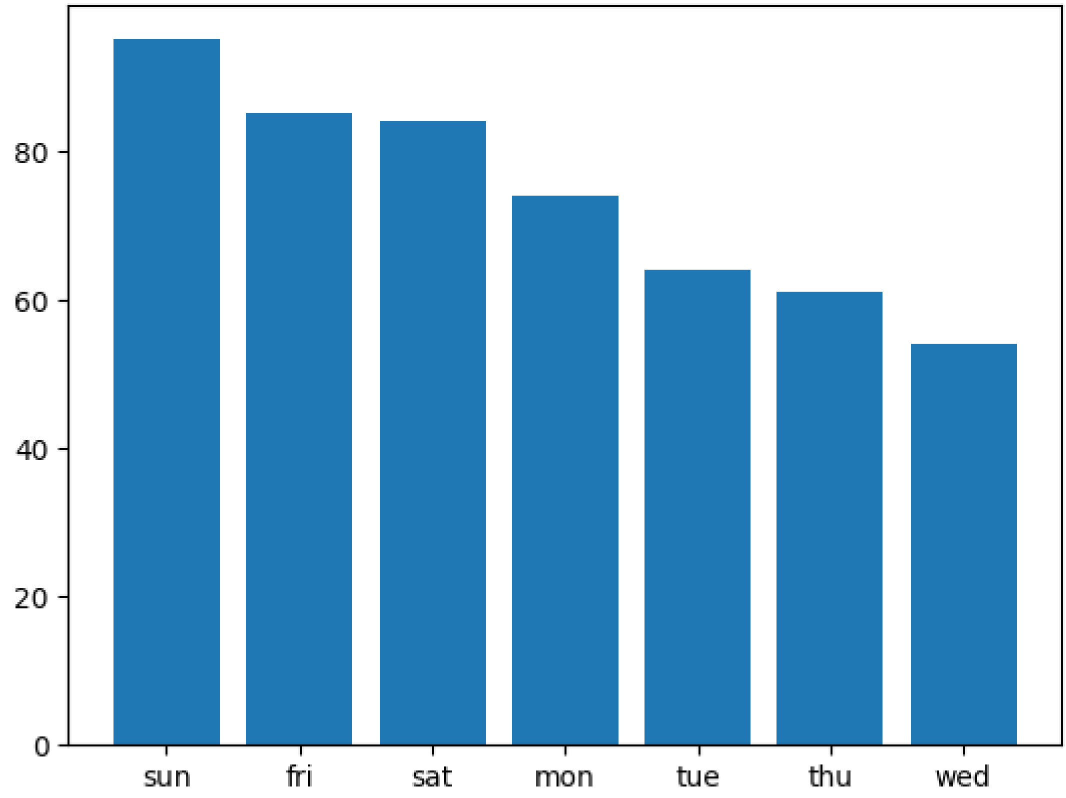
```
array([95, 85, 84, 74, 64, 61, 54], dtype=int64)
```

In [23]:

```
#day wise data

plt.bar(day_data.index, day_data.values)
```

Out[23]:

```
<BarContainer object of 7 artists>
```



In [24]:

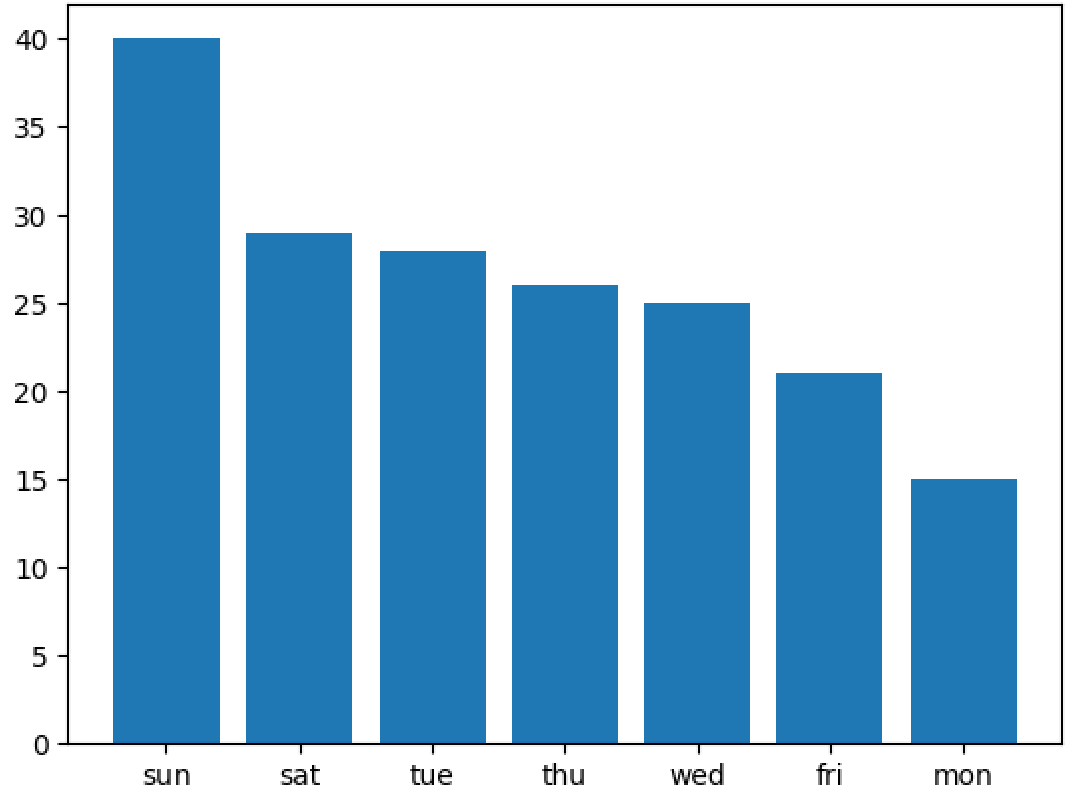
```
# month and day wise data
dm_data = data[data['month'] == 'aug']['day'].value_counts()
```

In [25]:

```
plt.bar(dm_data.index, dm_data.values)
```

Out[25]:

```
<BarContainer object of 7 artists>
```

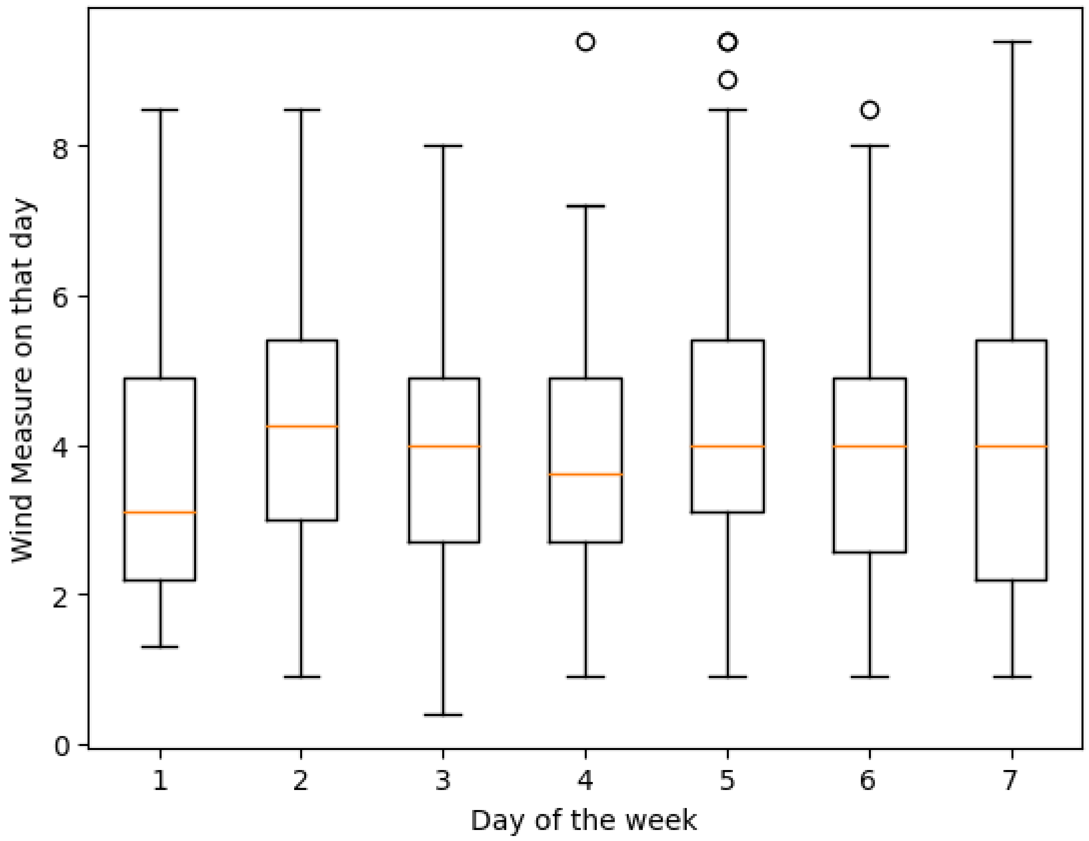


In [26]:

```
box_plot_data = []
box_plot_data.append(data[data['day'] == 'mon']['wind']);
box_plot_data.append(data[data['day'] == 'tue']['wind']);
box_plot_data.append(data[data['day'] == 'wed']['wind']);
box_plot_data.append(data[data['day'] == 'thu']['wind']);
box_plot_data.append(data[data['day'] == 'fri']['wind']);
box_plot_data.append(data[data['day'] == 'sat']['wind']);
box_plot_data.append(data[data['day'] == 'sun']['wind']);
```

In [27]:

```
plt.boxplot(box_plot_data)

# plt.title('Forest Fires by Month')
plt.xlabel('Day of the week')
plt.ylabel('Wind Measure on that day')
plt.show()
```

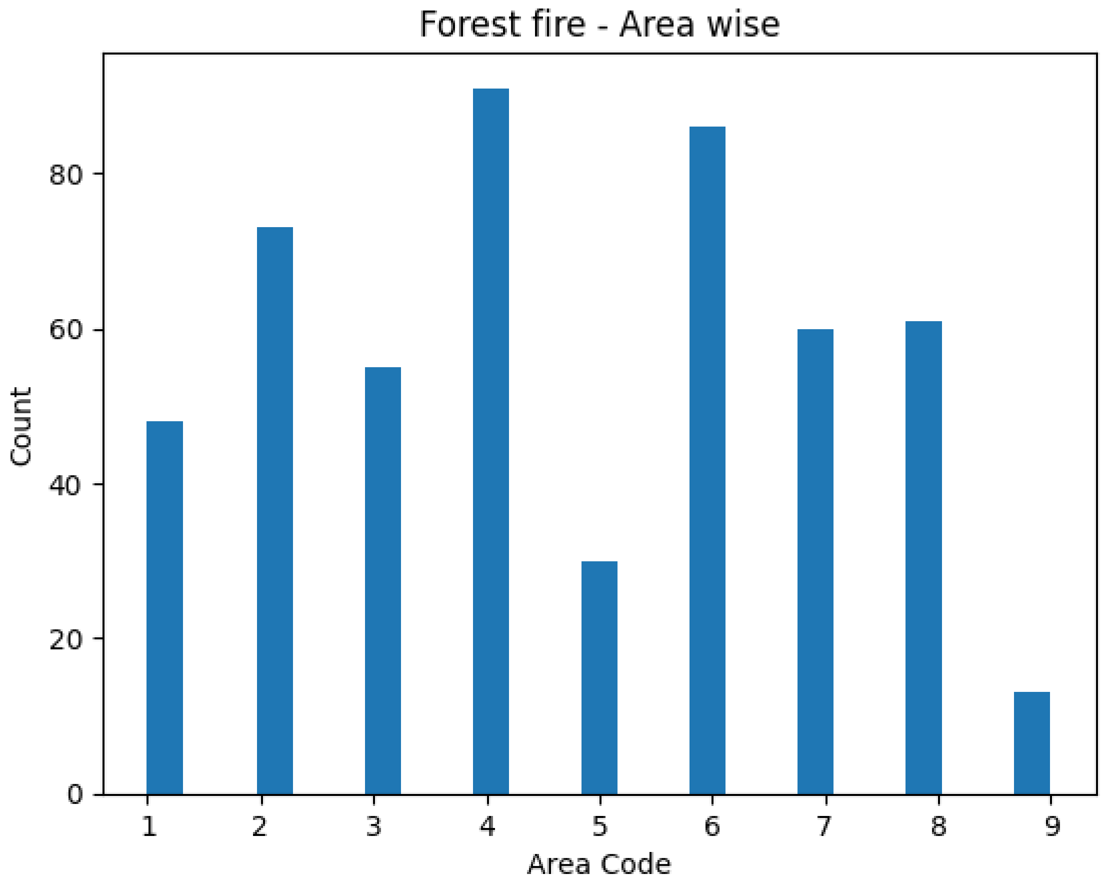


In [33]:

```
# Histogram

plt.hist(data['X'], bins=25)
plt.title('Forest fire - Area wise')
plt.xlabel('Area Code')
plt.ylabel('Count')
```

Out[33]:

```
Text(0, 0.5, 'Count')
```



In [ ]:

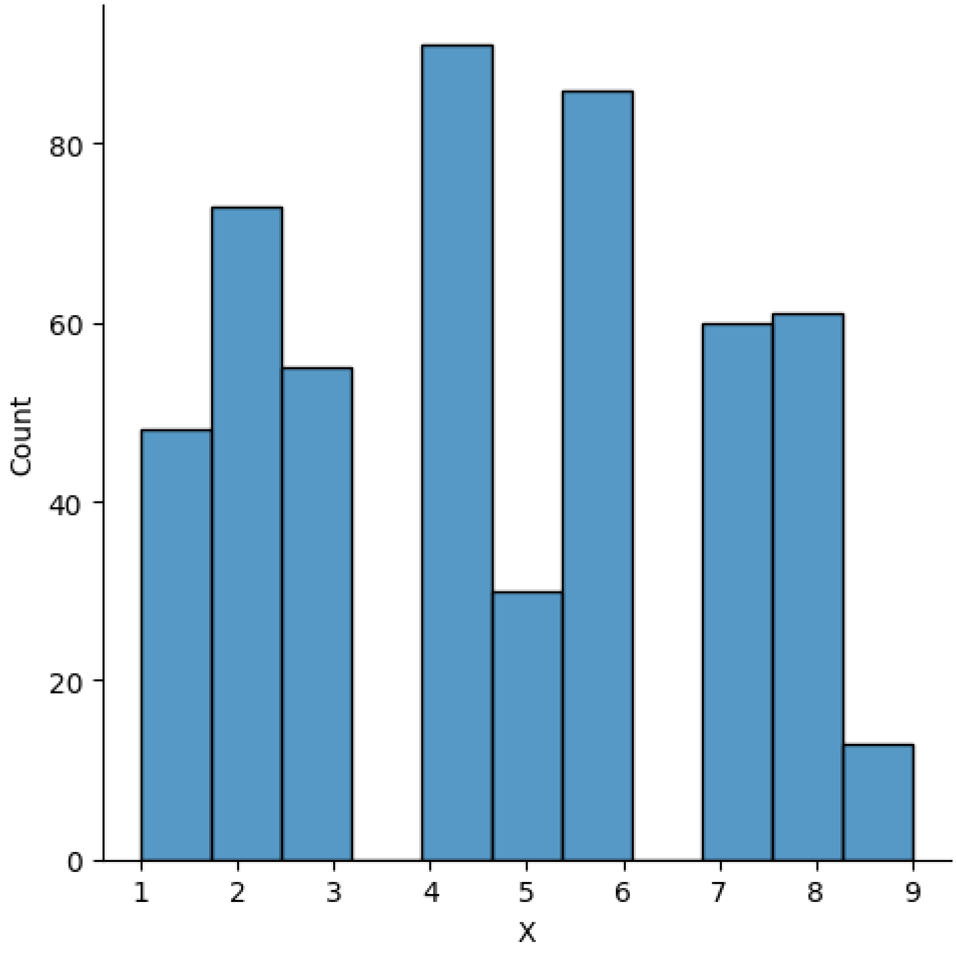
```
## Seaborn
```

In [34]:

```
sn.displot(data=data, x='X')
```

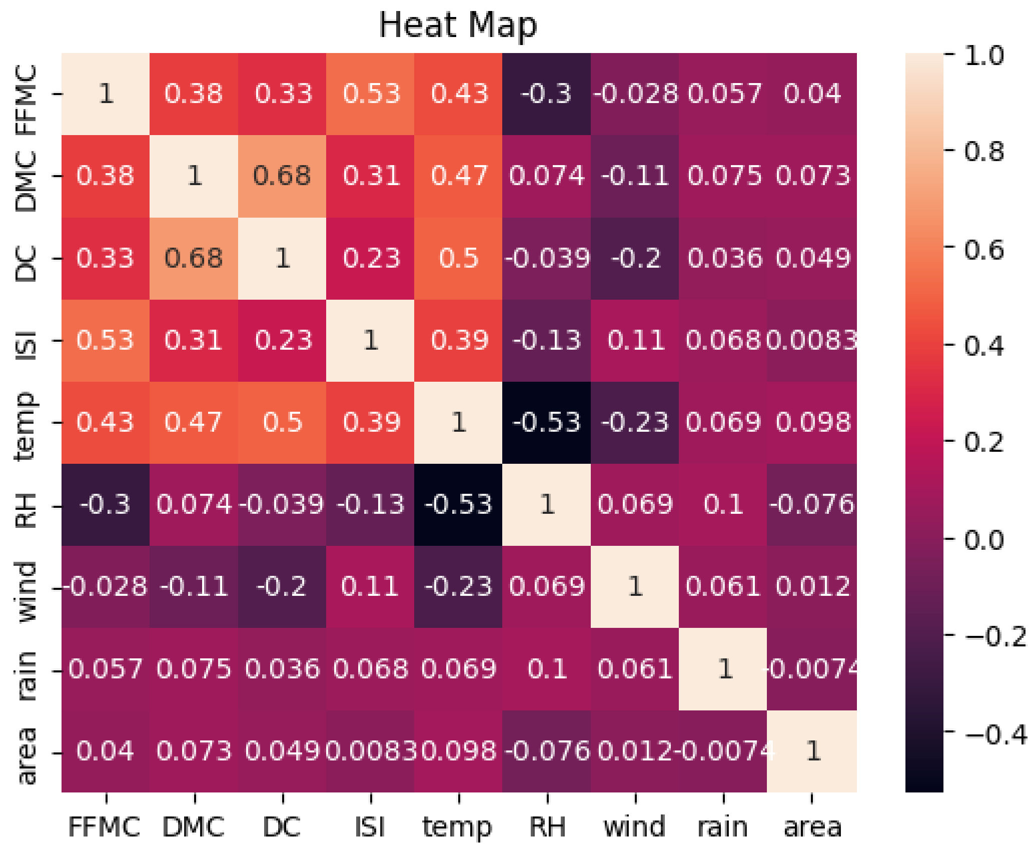
Out[34]:

```
<seaborn.axisgrid.FacetGrid at 0x2129d27cc50>
```



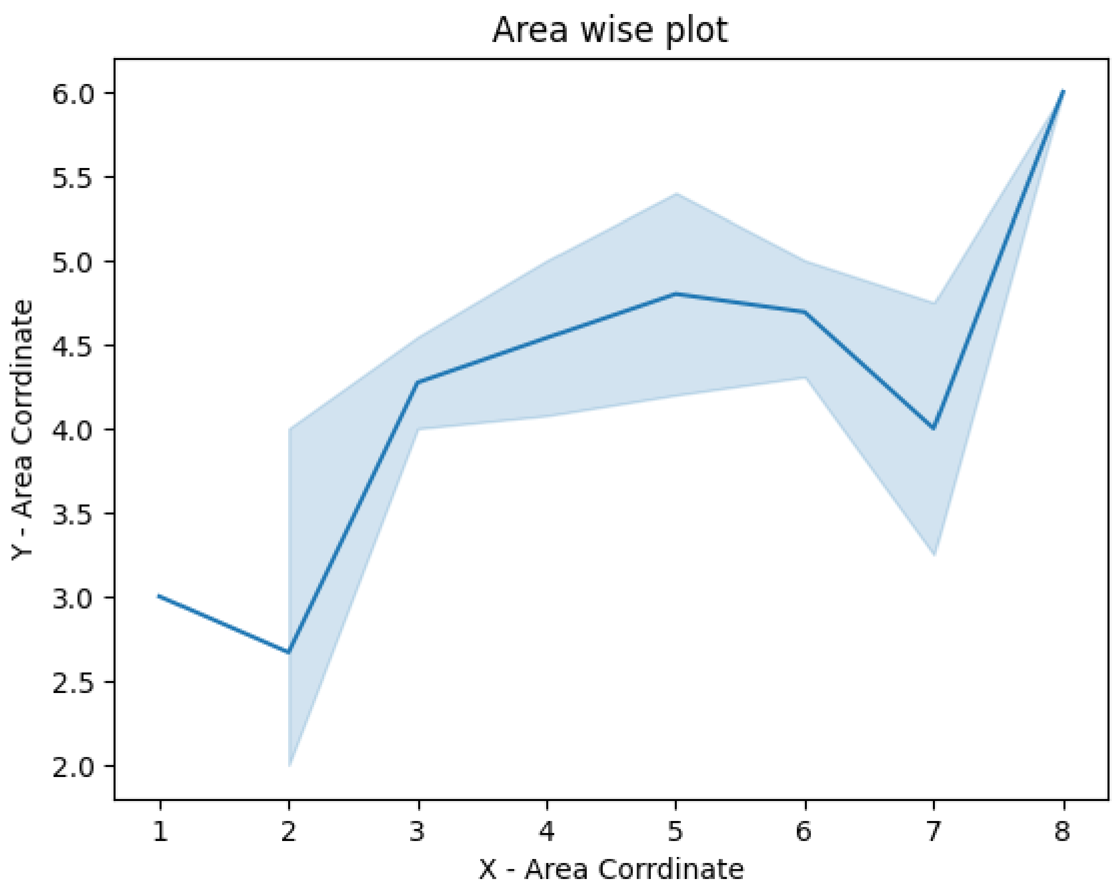
In [40]:

```
# Heat Map

correlation_matrix = data.iloc[:, 4:].corr()
sn.heatmap(correlation_matrix, annot=True)
plt.title('Heat Map')
plt.show()
```



In [49]:

```
# Line plot

sn.lineplot(data = data[data['month'] == 'mar'], x = 'X', y = 'Y')
plt.title('Area wise plot')
plt.xlabel('X - Area Corrdinate')
plt.ylabel('Y - Area Corrdinate')
plt.show()
```



In [47]:

```
data[data['month'] == 'mar']['Y'].unique()
```

Out[47]:

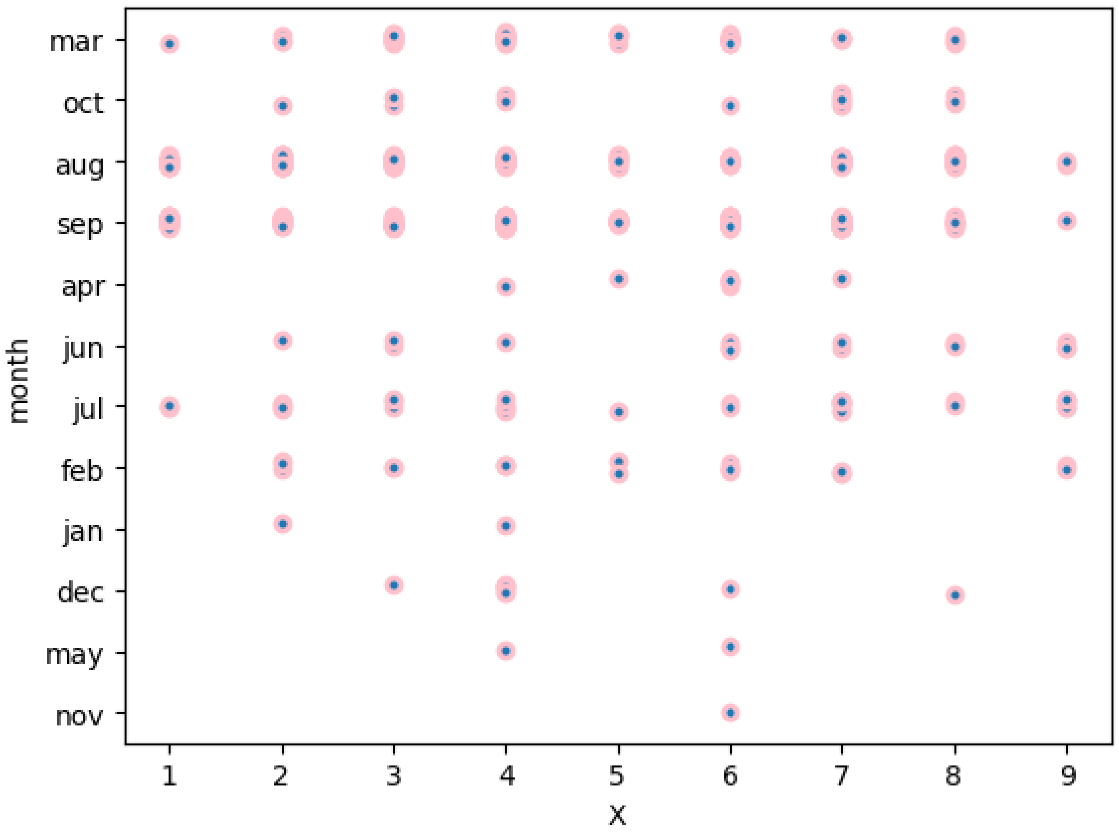
```
array([5, 6, 4, 2, 3], dtype=int64)
```

In [54]:

```
sn.stripplot(data=data, x = 'X', y = 'month', jitter = True, linewidth=2, edgecolor='pin
```

Out[54]:

```
<Axes: xlabel='X', ylabel='month'>
```



In [ ]: