

```
In [1]: %matplotlib inline
```

```
In [2]: import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
In [33]: #Q.1. Describe Statistics of all column
```

```
In [4]: data = pd.read_csv('coviddistrict.csv')
```

```
In [5]: data.head(10)
```

```
Out[5]:
```

	district	active	confirmed	recovered	deceased
0	Ahmadnagar	17	42	23	2
1	Yavatmal	69	79	10	0
2	Washim	1	2	1	0
3	Solapur	93	99	0	6
4	Sindhudurg	1	2	1	0
5	Satara	21	32	9	2
6	Sangli	3	29	25	1
7	Ratnagiri	2	8	5	1
8	Raigarh	44	71	24	3
9	Parbhani	1	2	1	0

```
In [6]: data.tail(10)
```

```
Out[6]:
```

	district	active	confirmed	recovered	deceased
24	Buldana	3	21	17	1
25	Bid	0	1	1	0
26	Bhandara	1	1	0	0
27	Aurangabad	102	131	22	7
28	Amravati	17	28	4	7
29	Akola	30	39	8	1
30	Ahmadnagar	17	42	23	2
31	Mumbai	5679	7061	1092	290
32	Thane	755	943	172	16
33	Pune	912	1248	248	88

```
In [7]: data.describe
```

```
Out[7]: <bound method NDFrame.describe of
0 Ahmadnagar 17 42 23 2
1 Yavatmal 69 79 10 0
2 Washim 1 2 1 0
3 Solapur 93 99 0 6
4 Sindhudurg 1 2 1 0
5 Satara 21 32 9 2
6 Sangli 3 29 25 1
7 Ratnagiri 2 8 5 1
8 Raigarh 44 71 24 3
9 Parbhani 1 2 1 0
10 Palghar 119 169 46 4
11 Osmanabad 0 3 3 0
12 Nashik 179 197 6 12
13 Nandurbar 10 11 0 1
14 Nanded 3 3 0 0
15 Nagpur 100 139 37 2
16 Latur 3 12 8 1
17 Kolhapur 10 14 4 0
18 Buldana 3 21 17 1
19 Jalgaon 30 40 1 9
20 Hingoli 14 15 1 0
21 Gondiya 0 1 1 0
22 Dhule 22 25 0 3
23 Chandrapur 0 2 2 0
24 Buldana 3 21 17 1
25 Bid 0 1 1 0
26 Bhandara 1 1 0 0
27 Aurangabad 102 131 22 7
28 Amravati 17 28 4 7
29 Akola 30 39 8 1
30 Ahmadnagar 17 42 23 2
31 Mumbai 5679 7061 1092 290
32 Thane 755 943 172 16
33 Pune 912 1248 248 88>
```

```
In [8]: data.shape
```

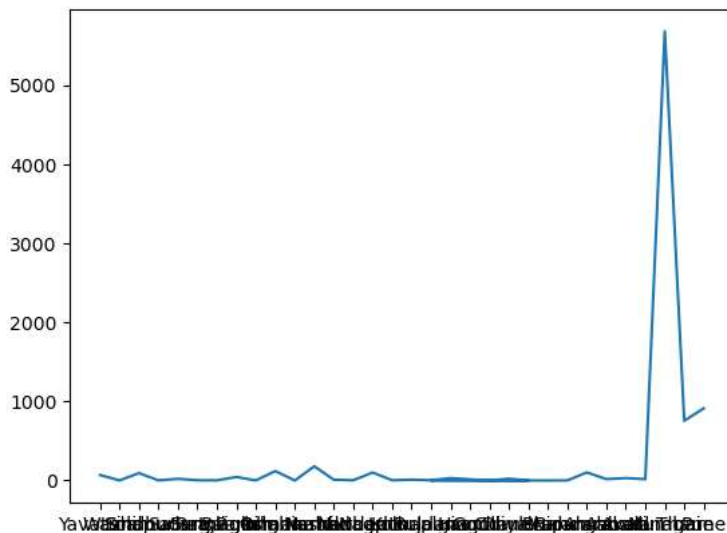
```
Out[8]: (34, 5)
```

```
In [ ]: # Q.2. plot line diagram of active,confirmed,recovered,deceased cases district wise.
```

```
In [ ]: Simple line plot
```

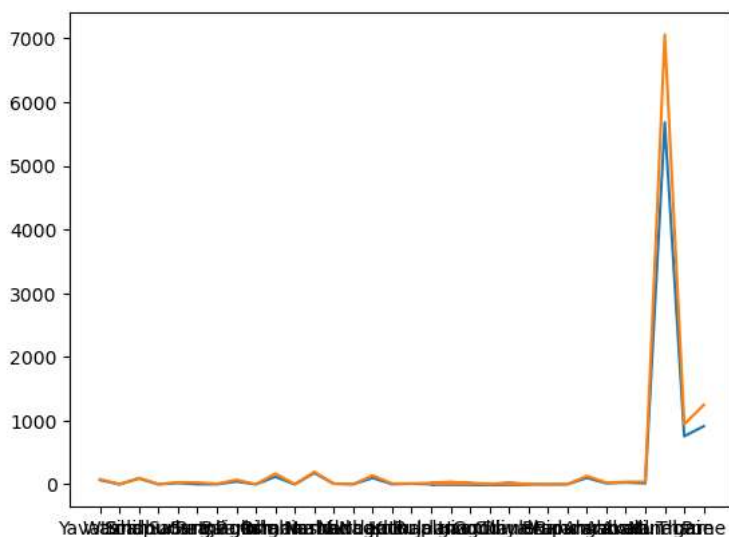
```
In [9]: #designating the data values to a alphabet.
Y = data.iloc[1:,1].values
R = data.iloc[1:,2].values
D = data.iloc[1:,3].values
W = data.iloc[1:,4].values
X = data.iloc[1:,0]
#Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
```

```
Out[9]: [<matplotlib.lines.Line2D at 0x22584caa040>]
```



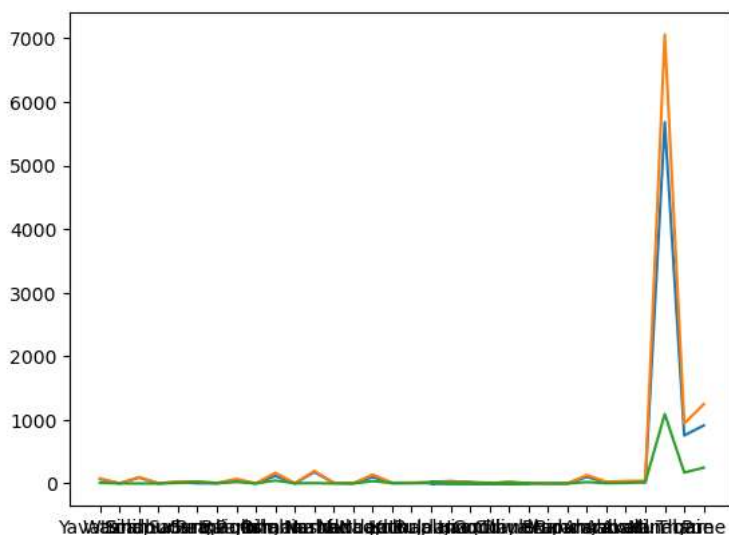
```
In [10]: #Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
#Line plot between District(X) and Confirmed cases(R)
plt.plot(X, R)
```

Out[10]: [<matplotlib.lines.Line2D at 0x22584f2d790>]



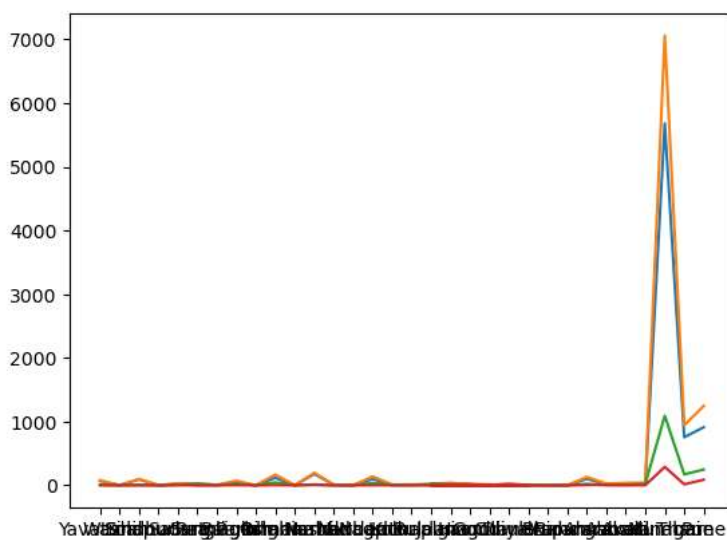
```
In [11]: #Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
#Line plot between District(X) and Confirmed cases(R)
plt.plot(X, R)
#Line plot between District(X) and Recovered cases(D)
plt.plot(X, D)
```

Out[11]: [<matplotlib.lines.Line2D at 0x22584d9ed00>]



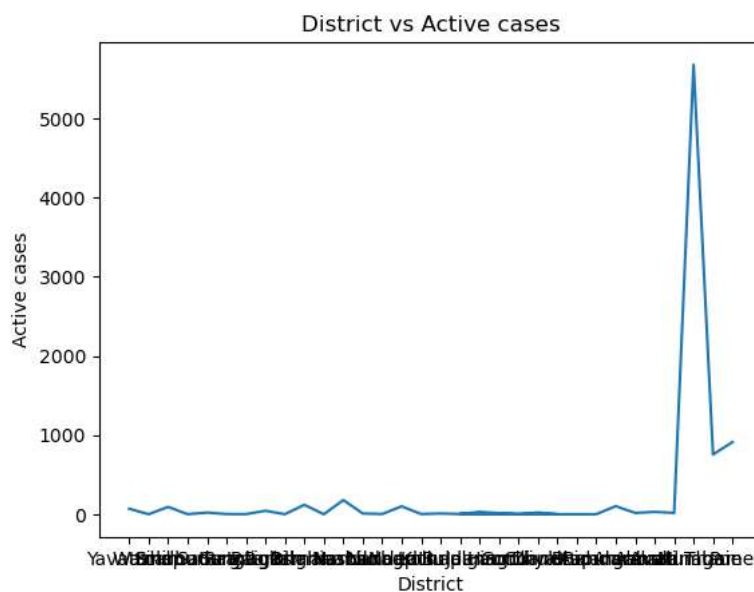
```
In [12]: #Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
#Line plot between District(X) and Confirmed cases(R)
plt.plot(X, R)
#Line plot between District(X) and Recovered cases(D)
plt.plot(X, D)
#Line plot between District(X) and Deceased cases(W)
plt.plot(X, W)
```

```
Out[12]: []
```

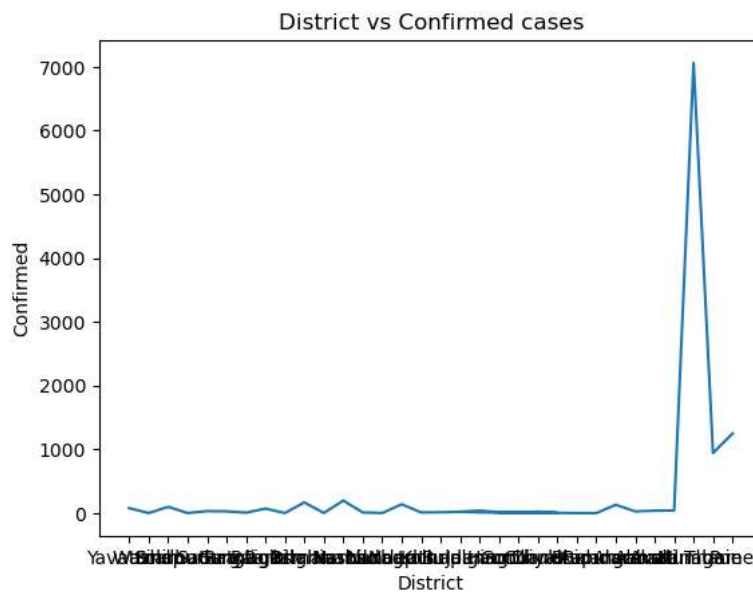


```
In [ ]: Title and Labels
```

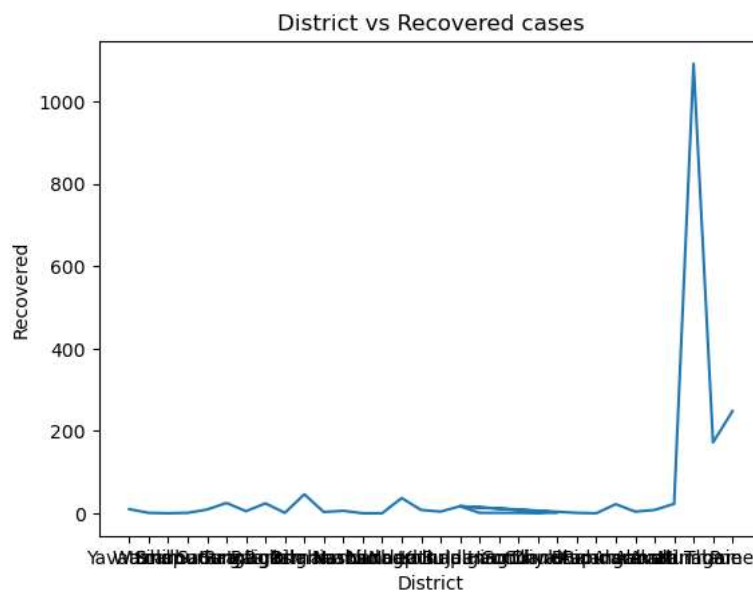
```
In [13]: plt.plot(X, Y)
plt.xlabel('District')
plt.ylabel('Active cases')
plt.title('District vs Active cases')
plt.show()
```



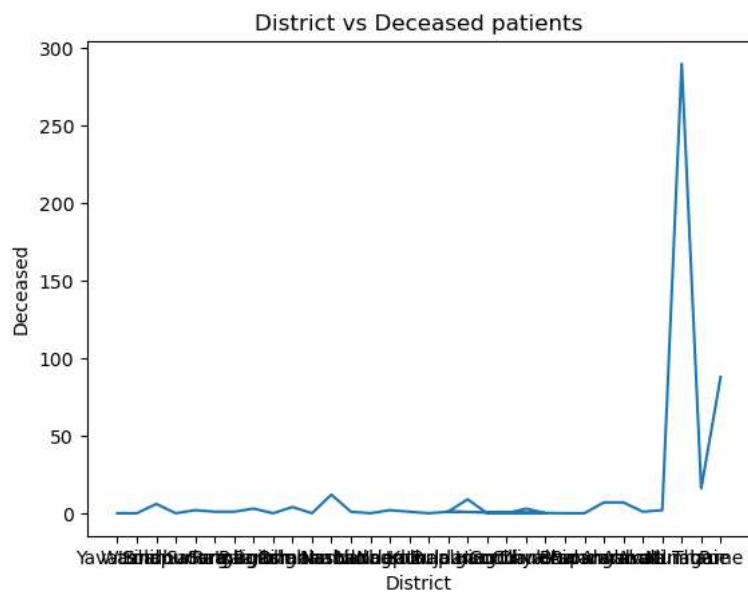
```
In [14]: plt.plot(X, R)
plt.xlabel('District')
plt.ylabel('Confirmed')
plt.title('District vs Confirmed cases')
plt.show()
```



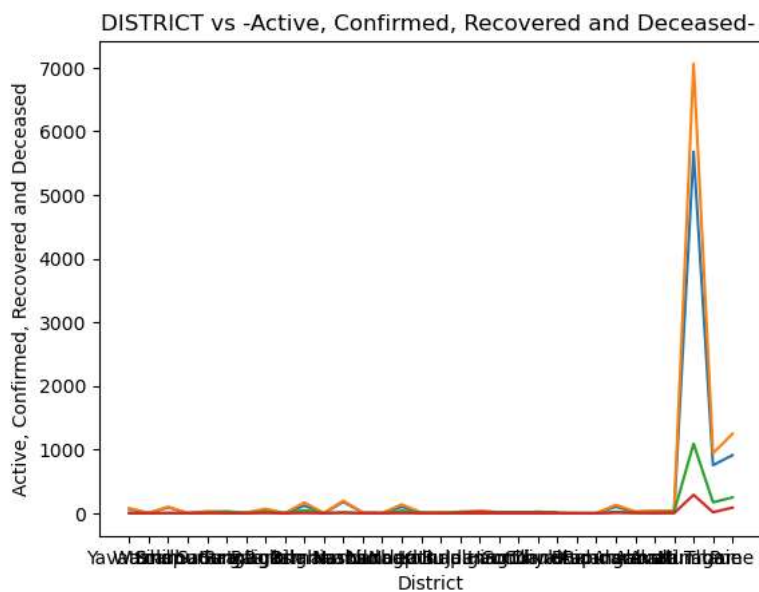
```
In [15]: plt.plot(X, D)
plt.xlabel('District')
plt.ylabel('Recovered')
plt.title('District vs Recovered cases')
plt.show()
```



```
In [16]: plt.plot(X, W)
plt.xlabel('District')
plt.ylabel('Deceased')
plt.title('District vs Deceased patients')
plt.show()
```

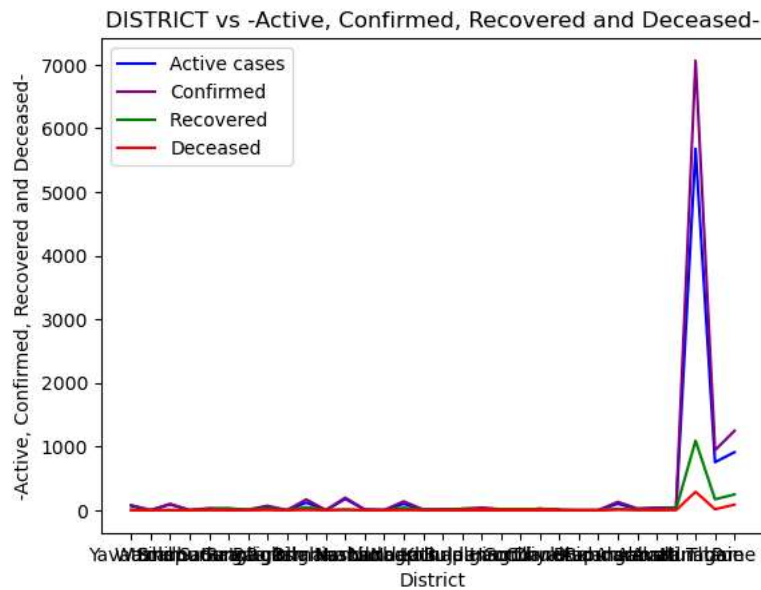


```
In [17]: #Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
#Line plot between District(X) and Confirmed cases(R)
plt.plot(X, R)
#Line plot between District(X) and Recovered cases(D)
plt.plot(X, D)
#Line plot between District(X) and Deceased cases(W)
plt.plot(X, W)
plt.xlabel('District')
plt.ylabel('Active, Confirmed, Recovered and Deceased')
plt.title('DISTRICT vs -Active, Confirmed, Recovered and Deceased- ')
plt.show()
```



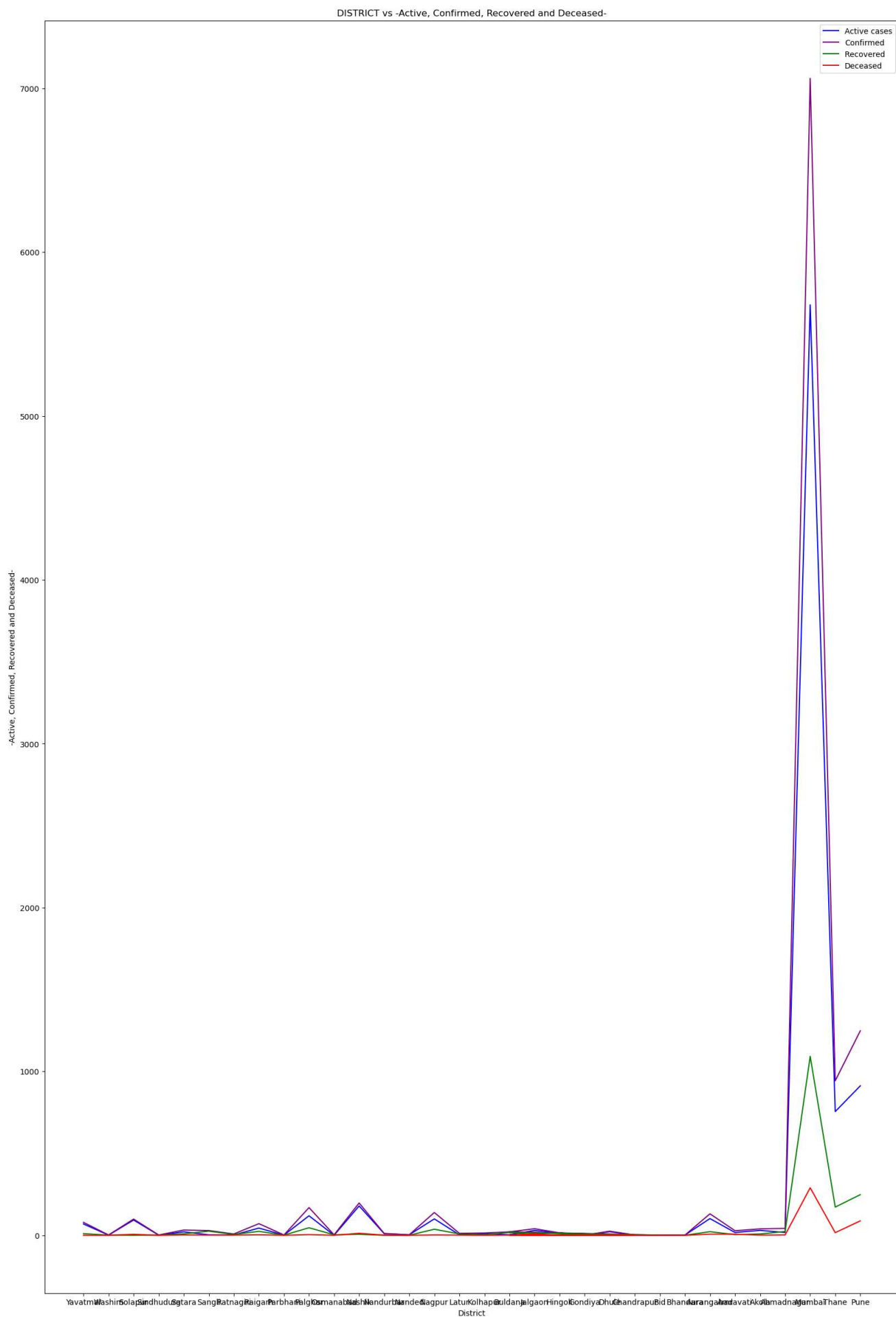
In []: Graph Legends

```
In [18]: # District vs Active cases
plt.plot(X, Y, label="Active cases", color ="blue")
# District vs Confirmed cases
plt.plot(X, R, label="Confirmed", color = "purple")
# District vs Recovered cases
plt.plot(X, D, label="Recovered " , color ="Green")
# District vs Deceased patients
plt.plot(X, W, label="Deceased", color ="red")
plt.xlabel('District')
plt.ylabel('-Active, Confirmed, Recovered and Deceased-')
plt.title('DISTRICT vs -Active, Confirmed, Recovered and Deceased-')
plt.legend()
plt.show()
```

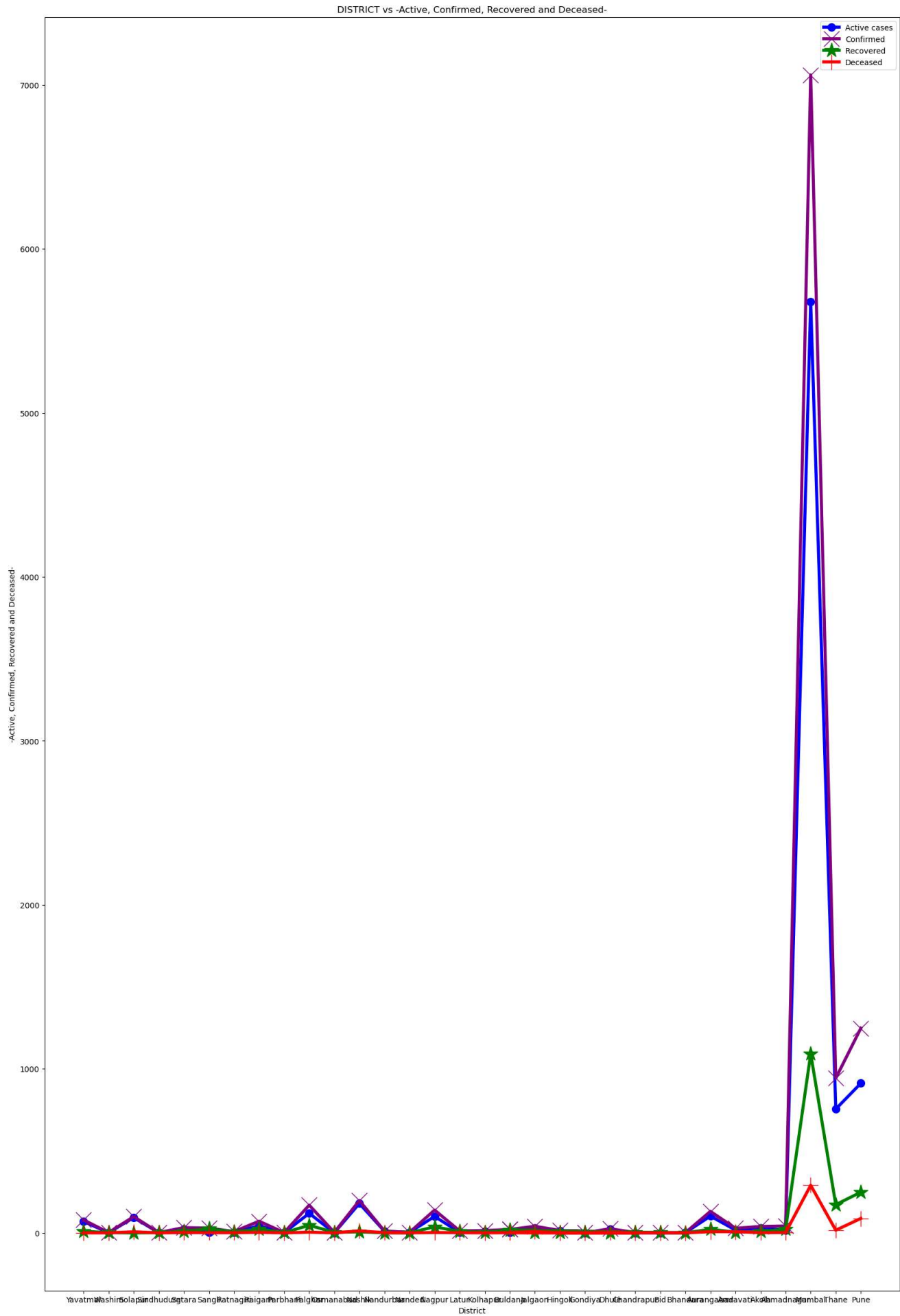


In []: Customization

```
In [19]: plt.figure(figsize=(20,30))
# District vs Active cases
plt.plot(X, Y, label="Active cases", color ="blue")
# District vs Confirmed cases
plt.plot(X, R, label="Confirmed", color = "purple")
# District vs Recovered cases
plt.plot(X, D, label="Recovered " , color ="Green")
# District vs Deceased patients
plt.plot(X, W, label="Deceased", color ="red")
plt.xlabel('District')
plt.ylabel('-Active, Confirmed, Recovered and Deceased-')
plt.title('DISTRICT vs -Active, Confirmed, Recovered and Deceased-')
plt.legend()
plt.show()
```

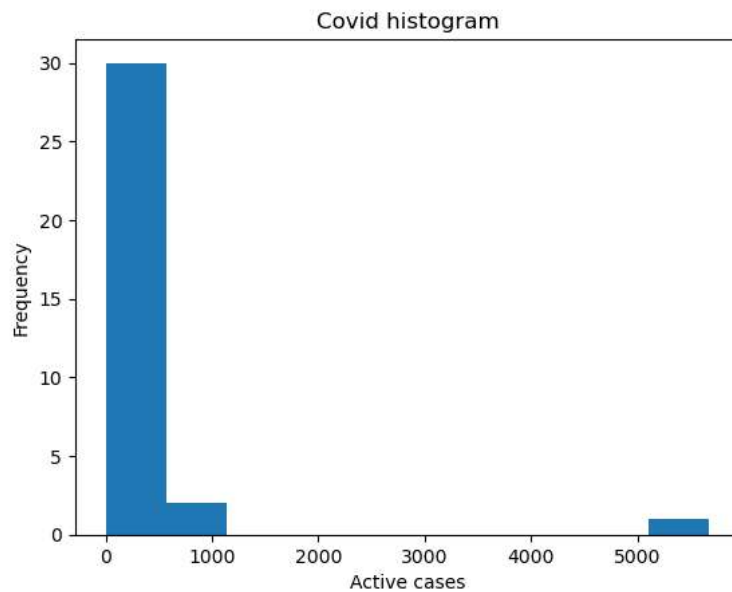



```
In [20]: plt.figure(figsize=(20,30))
# District vs Active cases
# By writing linewidth, we can increase or decrease the width of the line in line graph.
# By writing marker, we can designate the end point of the line graph as a marker.
plt.plot(X, Y, label="Active cases", color="blue", linewidth=4, marker='.', markersize='20')
# District vs Confirmed cases
plt.plot(X, R, label="Confirmed", color="purple", linewidth=4, marker='x', markersize='20')
# District vs Recovered cases
plt.plot(X, D, label="Recovered", color="green", linewidth=4, marker='*', markersize='20')
# District vs Deceased patients
plt.plot(X, W, label="Deceased", color="red", linewidth=4, marker='+', markersize='20')
plt.xlabel('District')
plt.ylabel('-Active, Confirmed, Recovered and Deceased-')
plt.title('DISTRICT vs -Active, Confirmed, Recovered and Deceased-')
plt.legend()
plt.show()
```

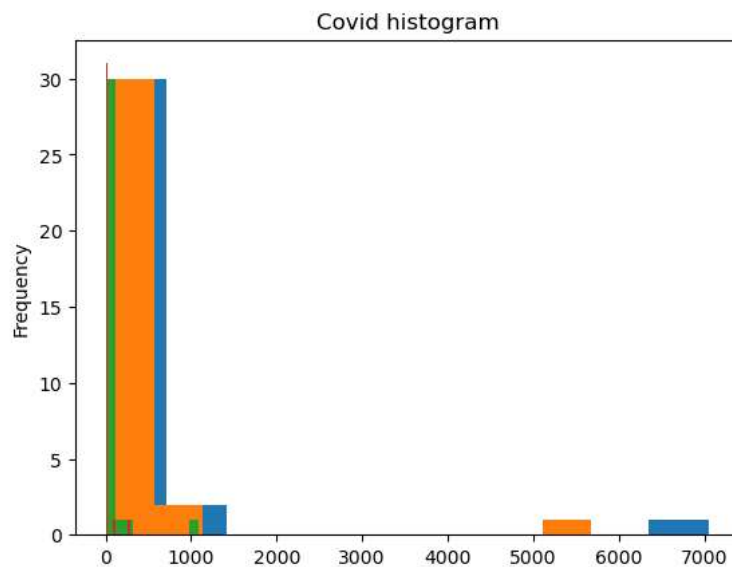


In []: Histogram

```
In [21]: plt.hist(Y)
plt.xlabel("Active cases")
plt.ylabel("Frequency")
plt.title("Covid histogram")
plt.show()
```

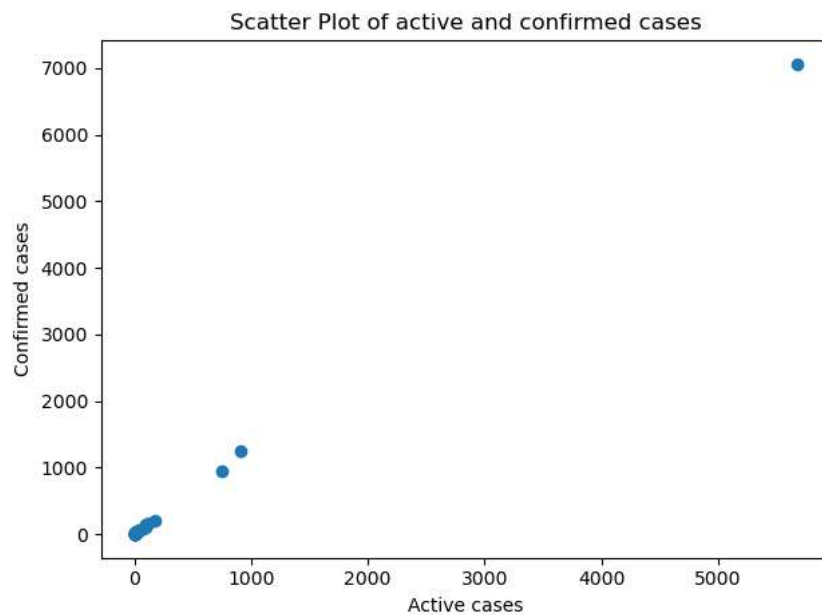


```
In [22]: plt.hist(R)
plt.hist(Y)
plt.hist(D)
plt.hist(W)
plt.ylabel("Frequency")
plt.title("Covid histogram")
plt.show()
```

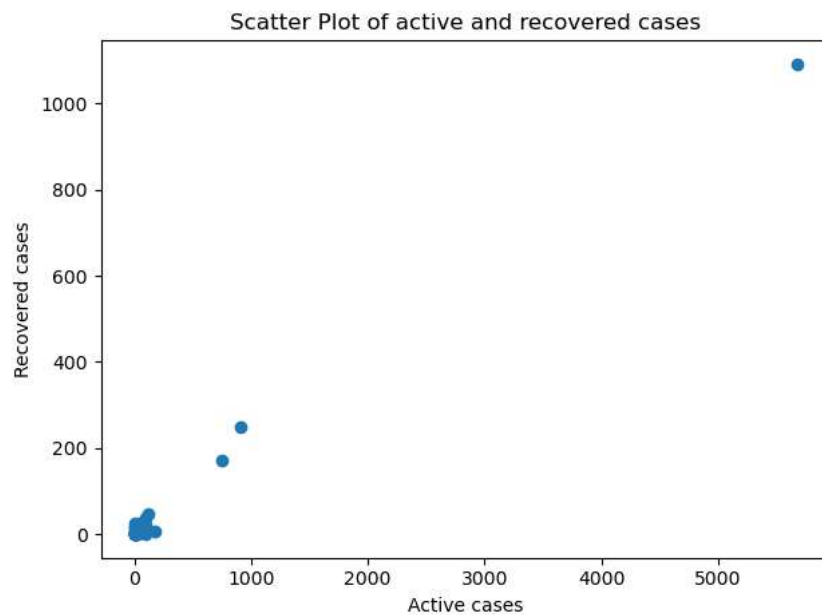


In []: Scatter Plot

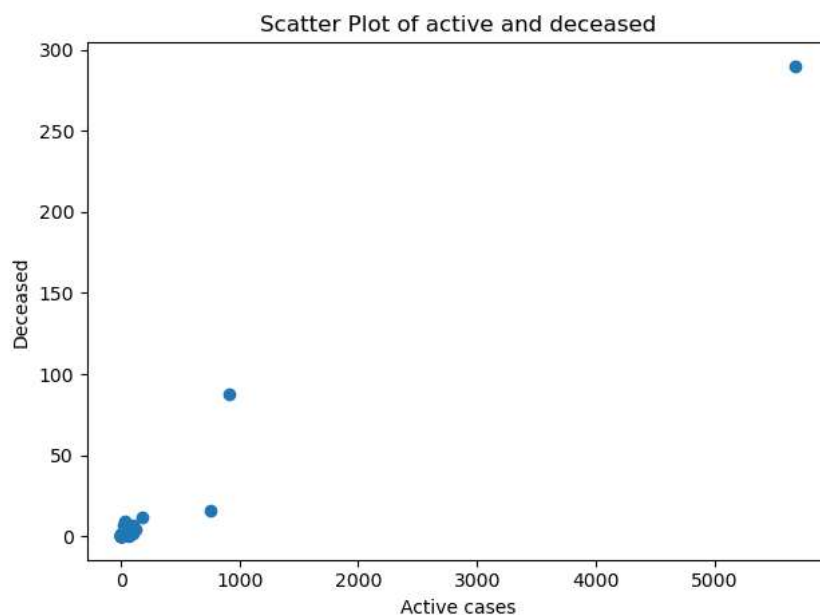
```
In [23]: plt.scatter(Y, R)
plt.xlabel("Active cases")
plt.ylabel("Confirmed cases")
plt.title("Scatter Plot of active and confirmed cases")
plt.tight_layout()
plt.show()
```



```
In [24]: plt.scatter(Y, D)
plt.xlabel("Active cases")
plt.ylabel("Recovered cases")
plt.title("Scatter Plot of active and recovered cases")
plt.tight_layout()
plt.show()
```

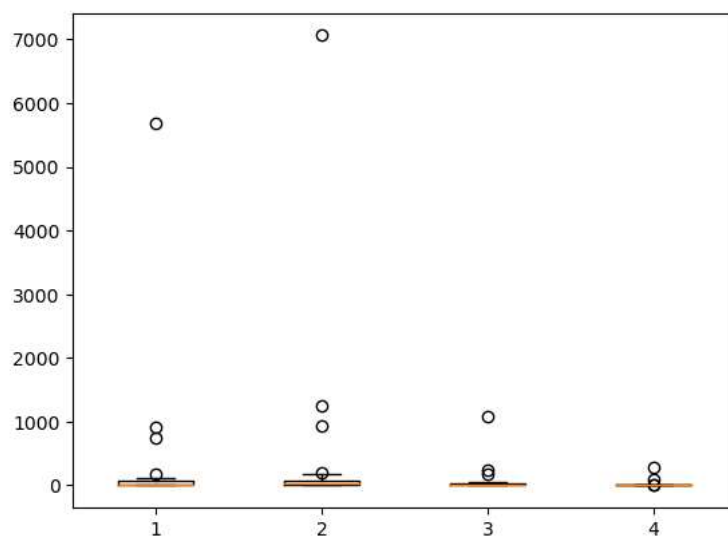


```
In [25]: plt.scatter(Y, W)
plt.xlabel("Active cases")
plt.ylabel("Deceased")
plt.title("Scatter Plot of active and deceased")
plt.tight_layout()
plt.show()
```



```
In [ ]: Box Plot
```

```
In [26]: collections = [Y, R, D, W]
plt.boxplot(collections)
plt.show()
```



```
In [37]: #Q.3. Plot a bar diagram showing confirmed, recovered and deceased cases district wise.
```

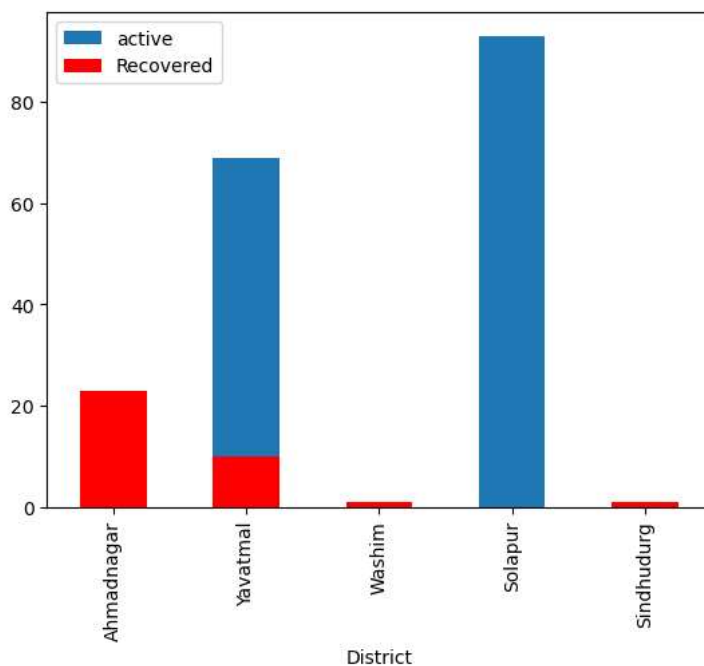
```
In [ ]: Bar Graph
```

```
In [27]: import matplotlib.pyplot as plt
import pandas as pd
```

```
In [28]: data = pd.read_csv('coviddistrict.csv')
```

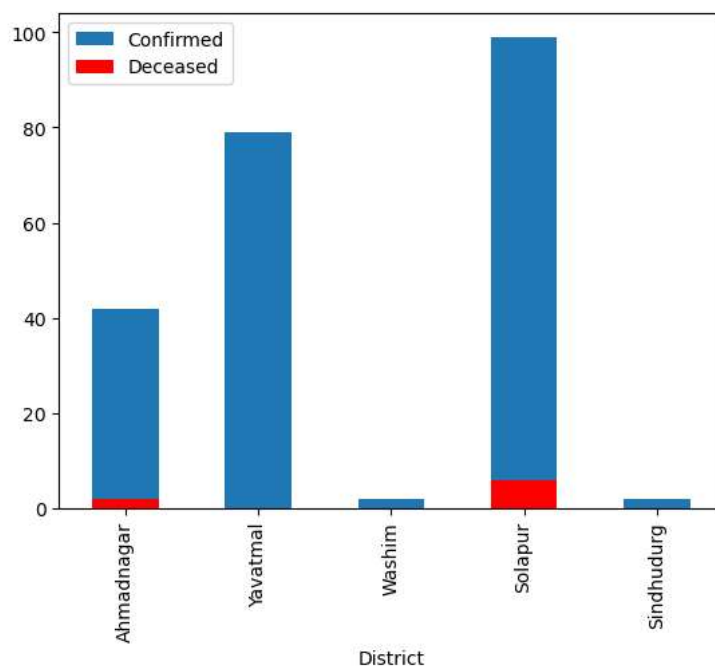
```
In [29]: df = pd.DataFrame({
'District': ['Ahmadnagar', 'Yavatmal', 'Washim', 'Solapur', 'Sindhudurg'],
'active': [17, 69, 1, 93, 1],
'Confirmed': [42, 79, 2, 99, 2],
'Recovered': [23, 10, 1, 0, 1],
'Deceased': [2, 0, 0, 6, 0]
})
ax = df.plot(x="District", y="active", kind="bar")
df.plot(x="District", y="Recovered", kind="bar", ax=ax, color="red")
```

Out[29]: <AxesSubplot:xlabel='District'>



```
In [30]: df = pd.DataFrame({
'District': ['Ahmadnagar', 'Yavatmal', 'Washim', 'Solapur', 'Sindhudurg'],
'active': [17, 69, 1, 93, 1],
'Confirmed': [42, 79, 2, 99, 2],
'Recovered': [23, 10, 1, 0, 1],
'Deceased': [2, 0, 0, 6, 0]
})
ax = df.plot(x="District", y="Confirmed", kind="bar")
df.plot(x="District", y="Deceased", kind="bar", ax=ax, color="red")
```

Out[30]: <AxesSubplot:xlabel='District'>



In []: