MATPLOTLIB

In [85]:
```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [86]:
```
data = pd.read_csv('district.csv')
```

In [87]:
```
data.head(15)
#top 10 data from dataset
```

Out[87]:

	district	active cases	confirmed cases	deceased	recovered
0	Ahmadnagar	17	42	2	23
1	Yavatmal	69	79	0	10
2	Washim	1	2	0	1
3	Solapur	93	99	6	0
4	Sindhudurg	1	2	0	1
5	Satara	21	32	2	9
6	Sangli	3	29	1	25
7	Ratnagiri	2	8	1	5
8	Raigarh	44	71	3	24
9	Parbhani	1	2	0	1
10	Palghar	119	169	4	46
11	Osmanabad	0	3	0	3
12	Nashik	179	197	12	6
13	Nandurbar	10	11	1	0
14	Nanded	3	3	0	0

In [88]:
```
data.tail(15)
#bottom 10 data from dataset
```

Out[88]:

	district	active cases	confirmed cases	deceased	recovered
19	Jalgaon	30	40	9	1
20	Hingoli	14	15	0	1
21	Gondiya	0	1	0	1
22	Dhule	22	25	3	0
23	Chandrapur	0	2	0	2
24	Buldana	3	21	1	17
25	Bid	0	1	0	1
26	Bhandara	1	1	0	0
27	Aurangabad	102	131	7	22
28	Amravati	17	28	7	4
29	Akola	30	39	1	8
30	Ahmadnagar	17	42	2	23
31	Mumbai	5679	7061	290	1092
32	Thane	755	943	16	172
33	Pune	912	1248	88	248

In [89]:
```
data.describe
#it 'describes' the data and shows you how the dataset looks
```

Out[89]:
```
<bound method NDFrame.describe of       district  active cases  confirmed c
ases  deceased  recovered
0   Ahmadnagar            17              42         2         23
1     Yavatmal            69              79         0         10
2       Washim             1               2         0          1
3      Solapur            93              99         6          0
4   Sindhudurg             1               2         0          1
5       Satara            21              32         2          9
6       Sangli             3              29         1         25
7    Ratnagiri             2               8         1          5
8      Raigarh            44              71         3         24
9     Parbhani             1               2         0          1
10     Palghar           119             169         4         46
11   Osmanabad             0               3         0          3
12      Nashik           179             197        12          6
13   Nandurbar            10              11         1          0
14      Nanded             3               3         0          0
15      Nagpur           100             139         2         37
16       Latur             3              12         1          8
17    Kolhapur            10              14         0          4
18     Buldana             3              21         1         17
19     Jalgaon            30              40         9          1
20     Hingoli            14              15         0          1
21     Gondiya             0               1         0          1
22       Dhule            22              25         3          0
23  Chandrapur             0               2         0          2
24     Buldana             3              21         1         17
25         Bid             0               1         0          1
26    Bhandara             1               1         0          0
27  Aurangabad           102             131         7         22
28    Amravati            17              28         7          4
29       Akola            30              39         1          8
30  Ahmadnagar            17              42         2         23
31      Mumbai          5679            7061       290       1092
32       Thane           755             943        16        172
33        Pune           912            1248        88        248>
```

In [90]:
```
print(data.shape[0])
#it will gives you No. of rows
```

```
34
```

In [91]:
```
print(data.shape[1])
#it will gives you No. of columns
```

```
5
```

In [92]:
```
data.index
#shows range
```

Out[92]:
```
RangeIndex(start=0, stop=34, step=1)
```

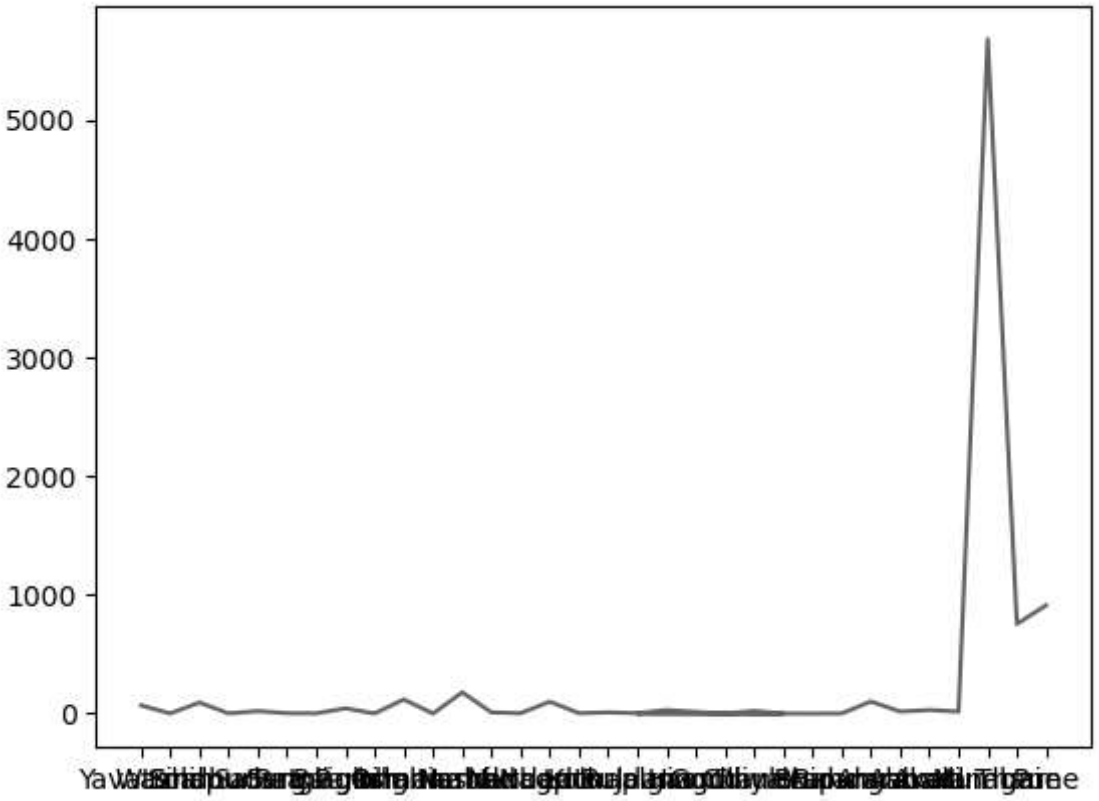
In [93]:
```
data.dtypes
#dtype is for data types, float is decimals
```

Out[93]:
```
district           object
active cases        int64
confirmed cases     int64
deceased            int64
recovered           int64
dtype: object
```

LINEGRAPH

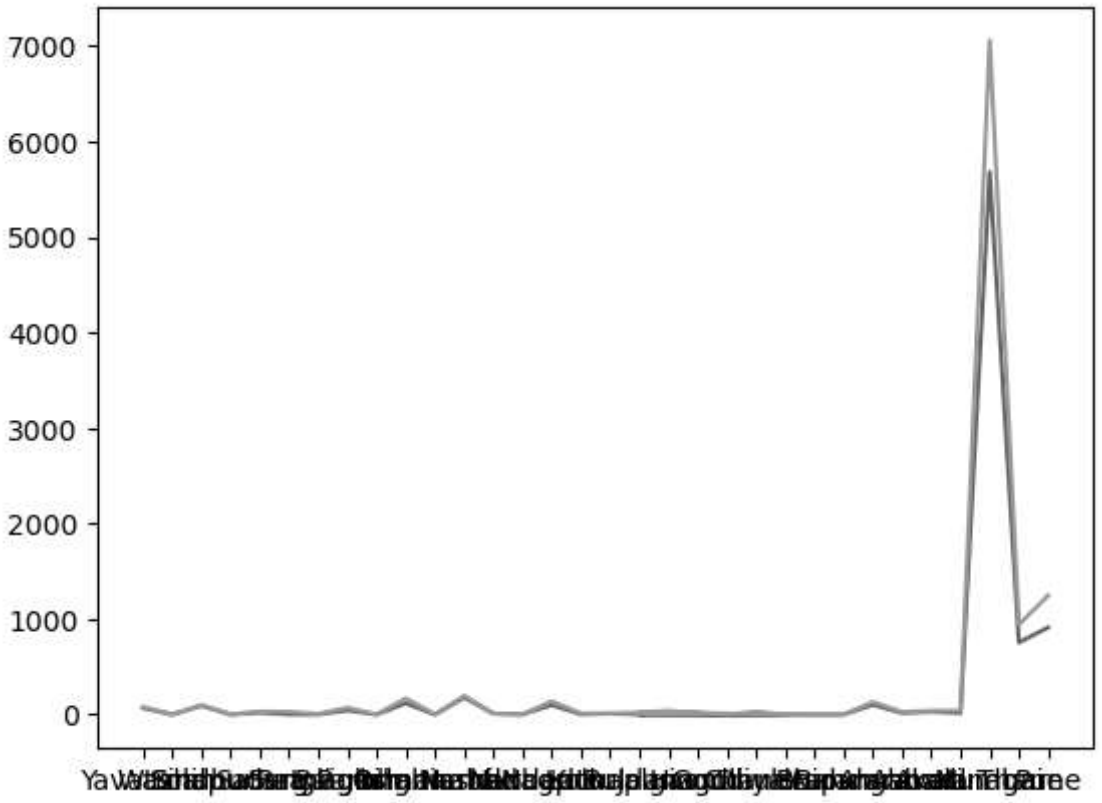
In [94]:
```
#designating the data values to a alphabet.
Y = data.iloc[1:,1].values
R = data.iloc[1:,2].values
D = data.iloc[1:,3].values
W = data.iloc[1:,4].values
X = data.iloc[1:,0]

#Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
```

Out[94]:
```
[<matplotlib.lines.Line2D at 0x23388f3a550>]
```



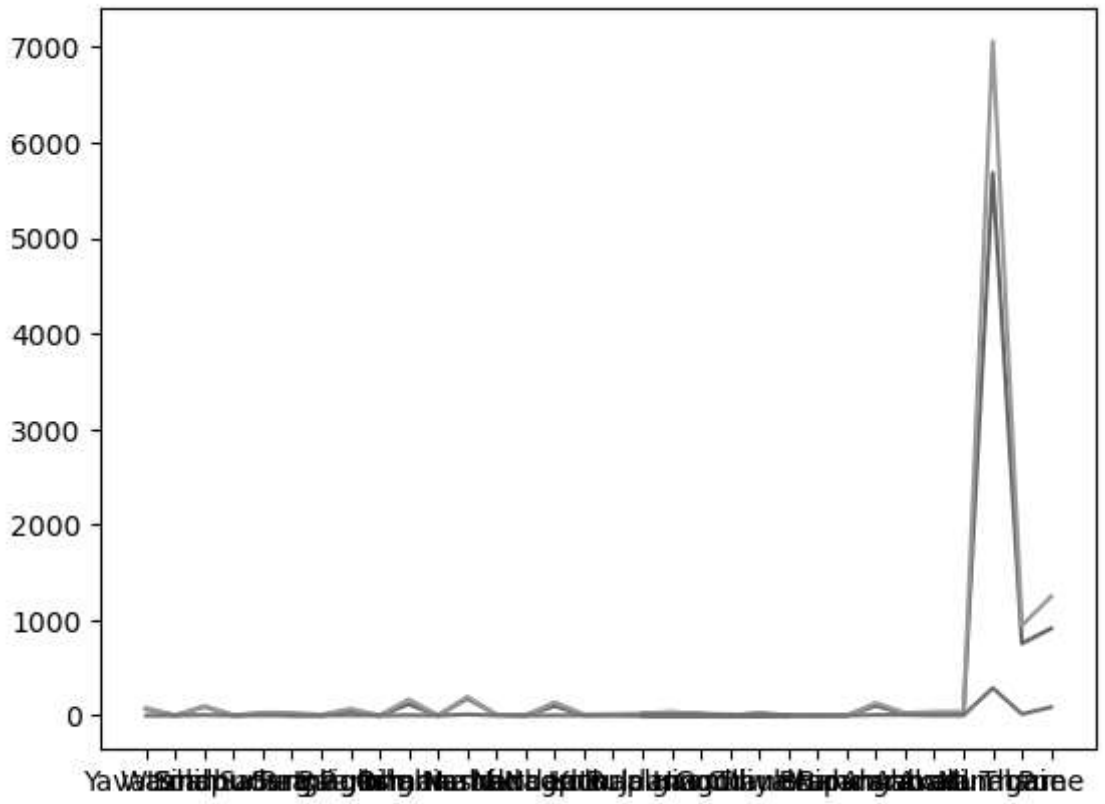
In [95]:
```
#Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
#Line plot between District(X) and Confirmed cases(R)
plt.plot(X, R)
```

Out[95]:
```
[<matplotlib.lines.Line2D at 0x23389cbca30>]
```



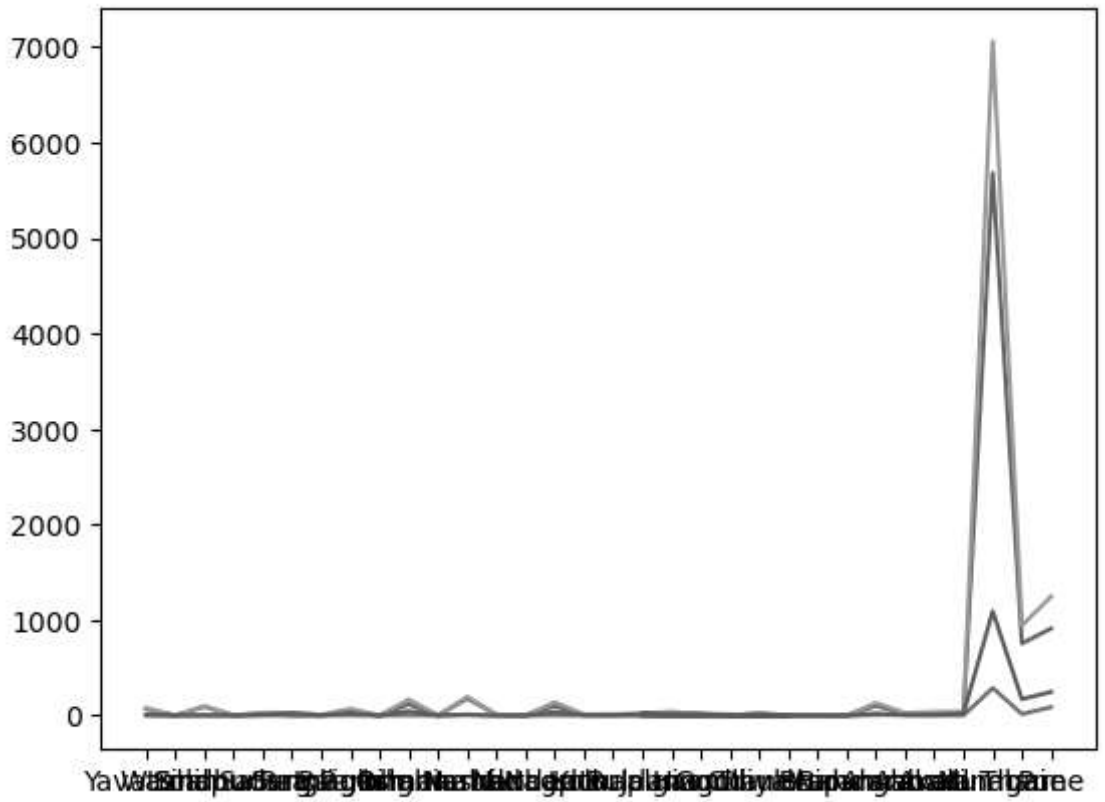
In [96]:
```
#Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
#Line plot between District(X) and Confirmed cases(R)
plt.plot(X, R)
#Line plot between District(X) and Recovered cases(D)
plt.plot(X, D)
```

Out[96]: [<matplotlib.lines.Line2D at 0x2338964ce80>]

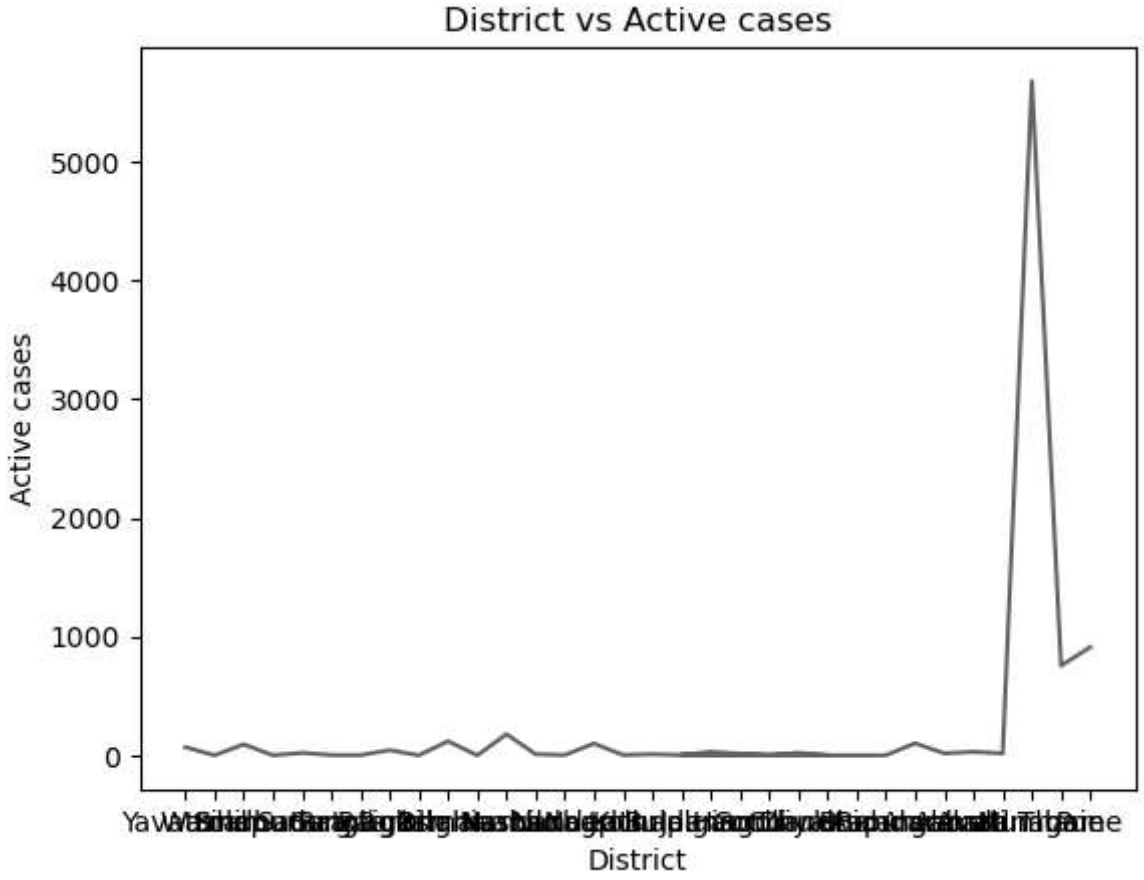


In [97]:
```
#Line plot between District(X) and Active cases(Y)
plt.plot(X, Y)
#Line plot between District(X) and Confirmed cases(R)
plt.plot(X, R)
#Line plot between District(X) and Recovered cases(D)
plt.plot(X, D)
#Line plot between District(X) and Deceased cases(W)
plt.plot(X, W)
```

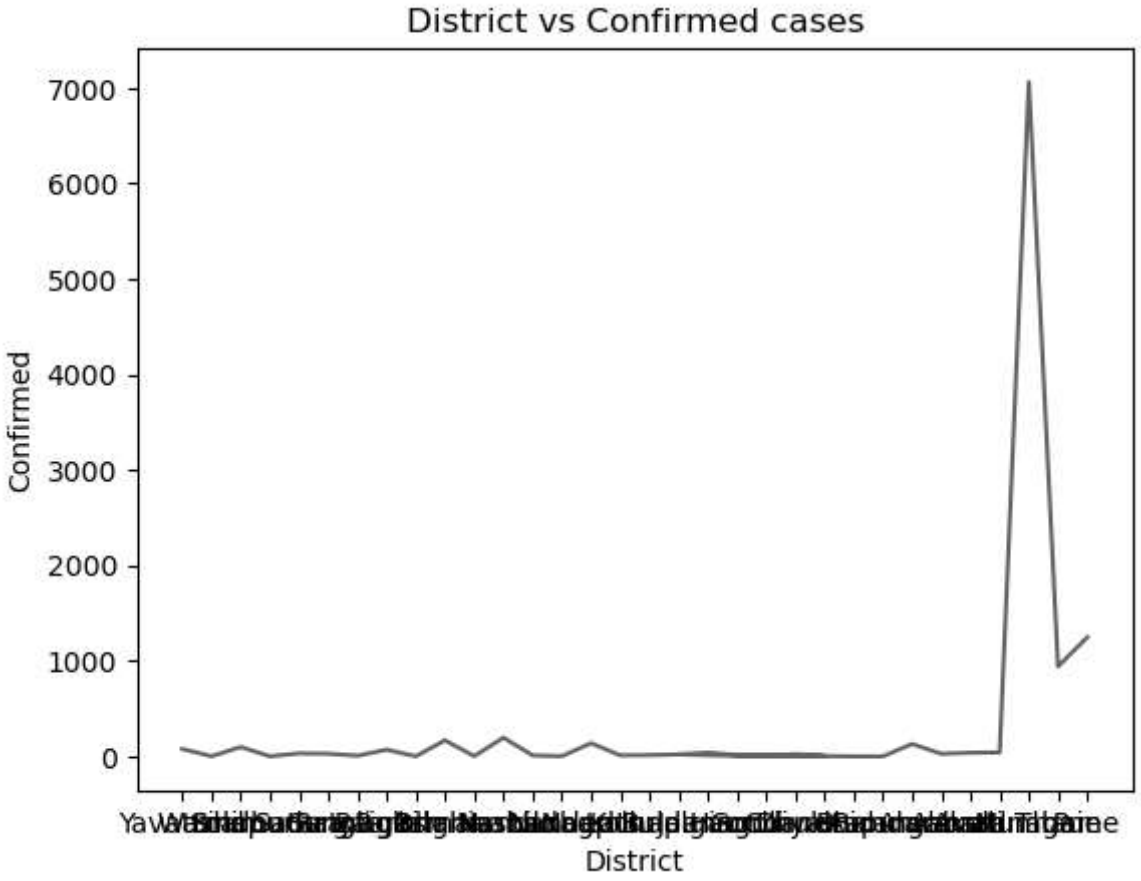
Out[97]:
```
[<matplotlib.lines.Line2D at 0x2338979c430>]
```



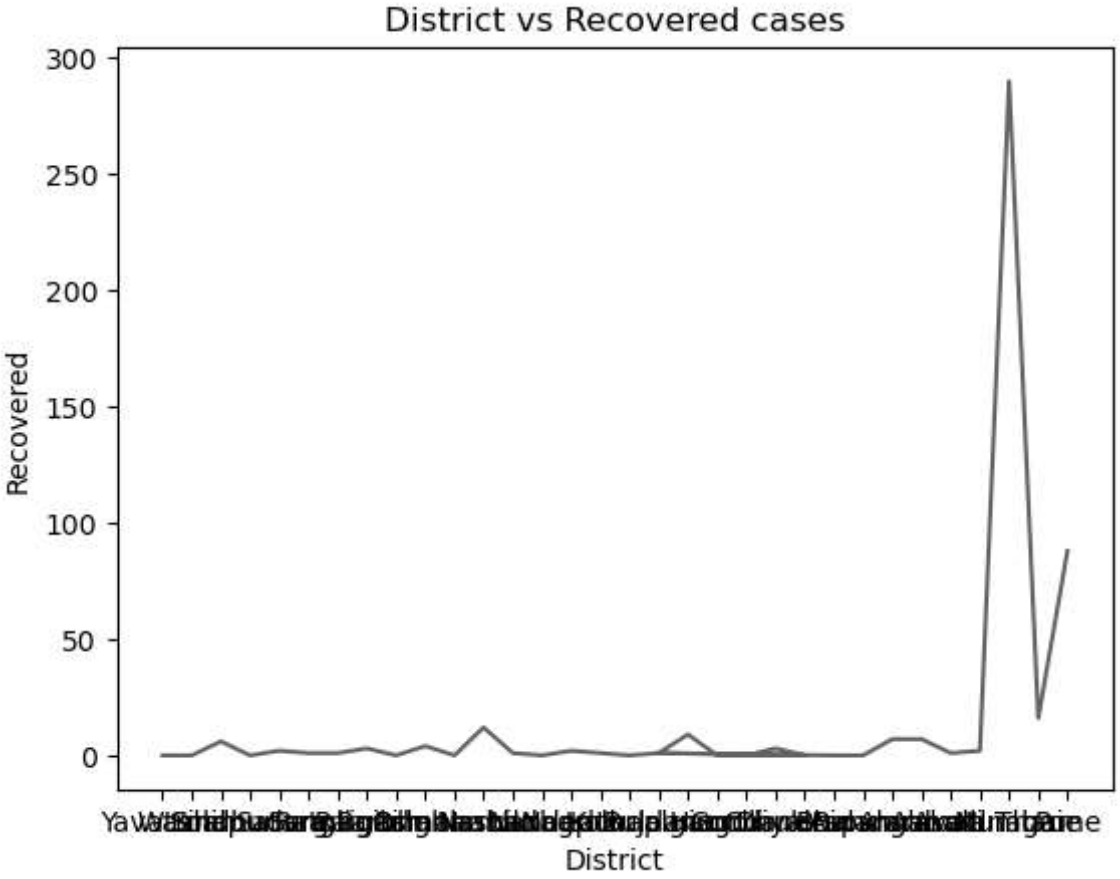
In [98]:
```
plt.plot(X, Y)
plt.xlabel('District')
plt.ylabel('Active cases')
plt.title('District vs Active cases')
plt.show()
```



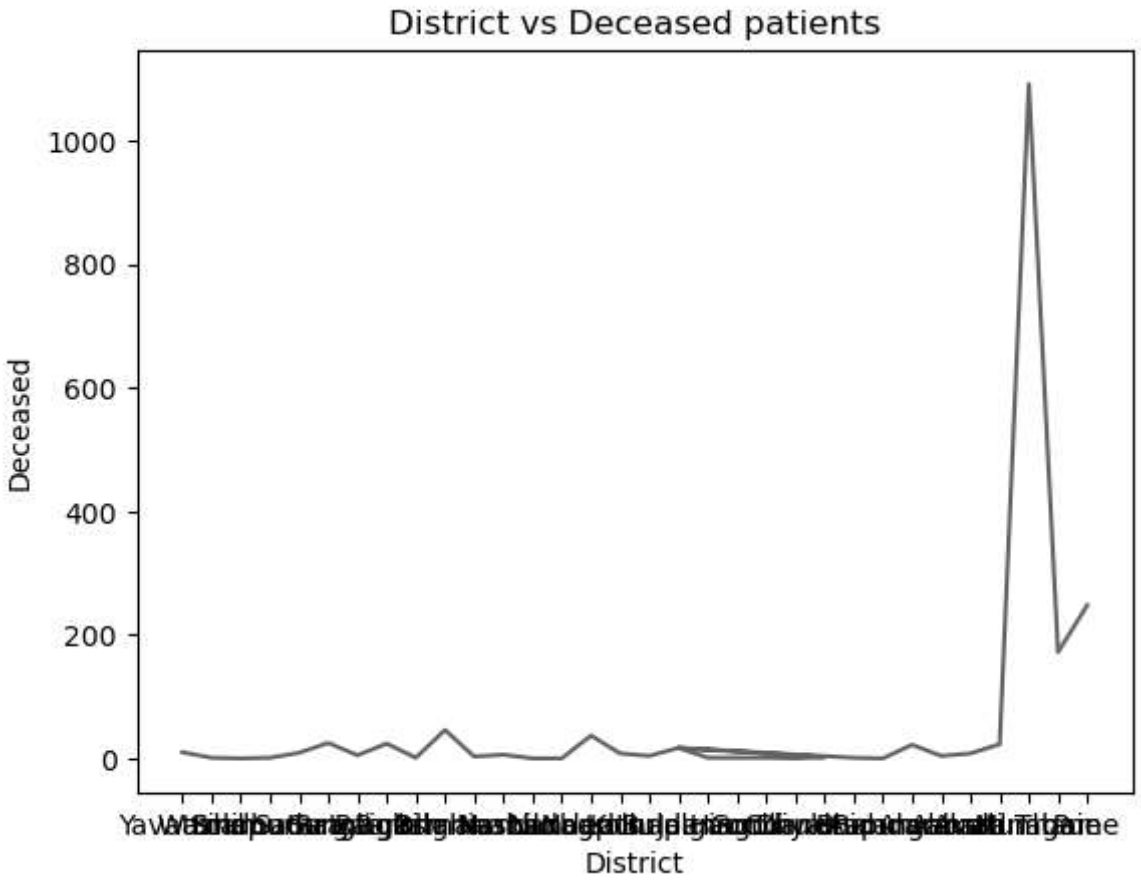
In [99]:
```
plt.plot(X, R)
plt.xlabel('District')
plt.ylabel('Confirmed')
plt.title('District vs Confirmed cases')
plt.show()
```



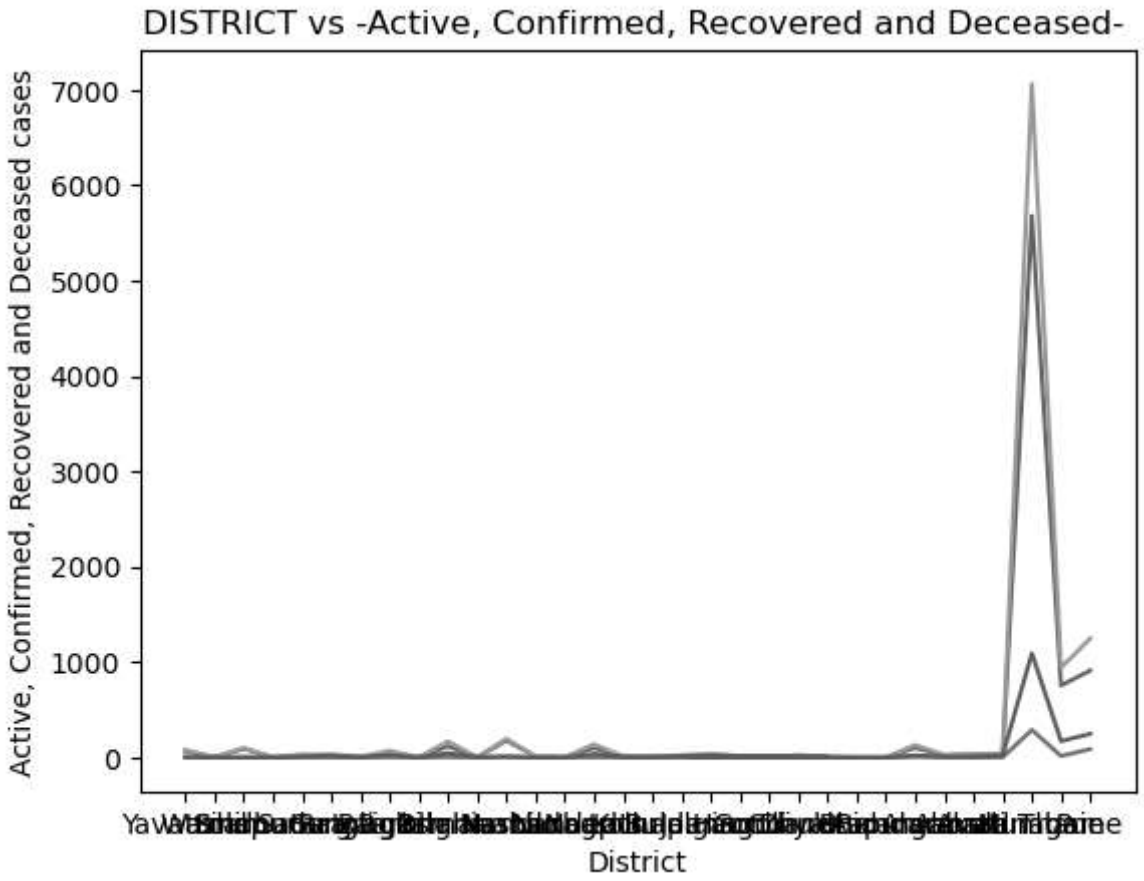
In [100]:
```
plt.plot(X, D)
plt.xlabel('District')
plt.ylabel('Recovered')
plt.title('District vs Recovered cases')
plt.show()
```



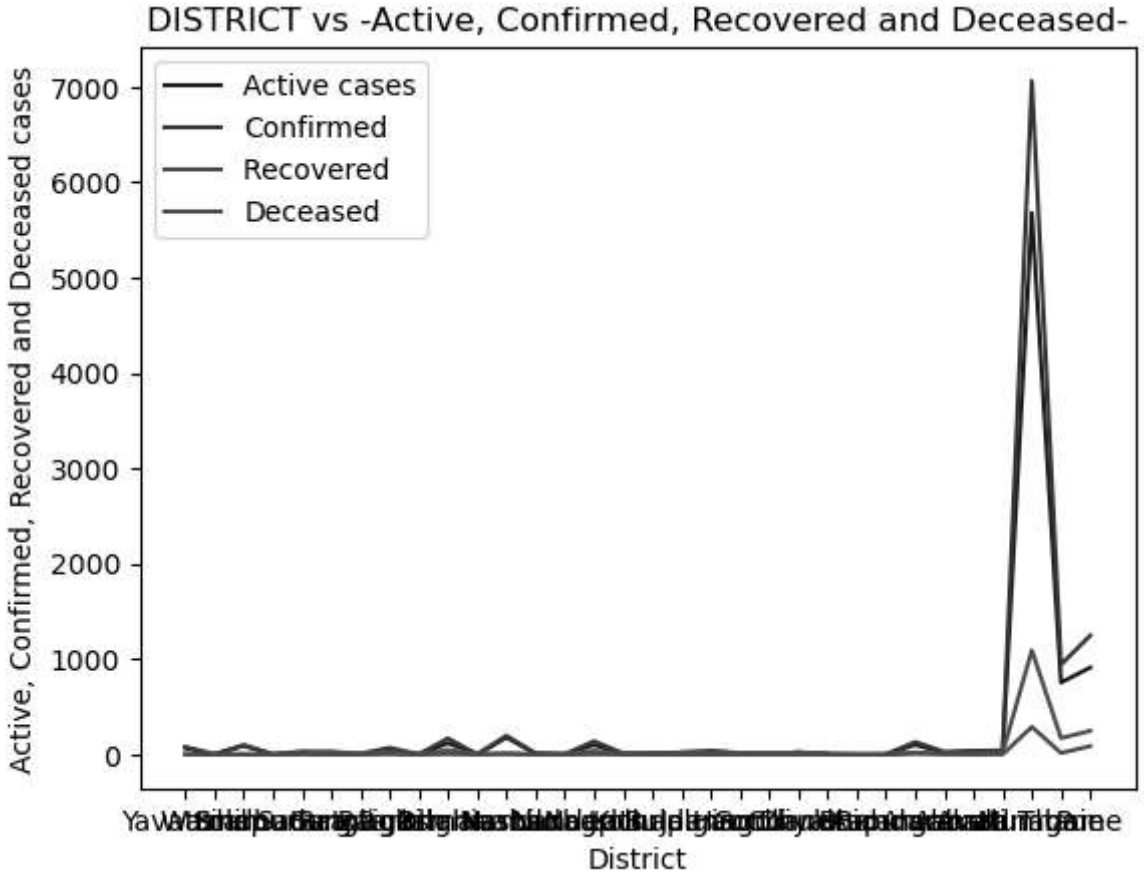
In [101]:
```
plt.plot(X, W)
plt.xlabel('District')
plt.ylabel('Deceased')
plt.title('District vs Deceased patients')
plt.show()
```



In [102]:
```
#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 cases')
plt.title('DISTRICT vs -Active, Confirmed, Recovered and Deceased- ')
plt.show()
```



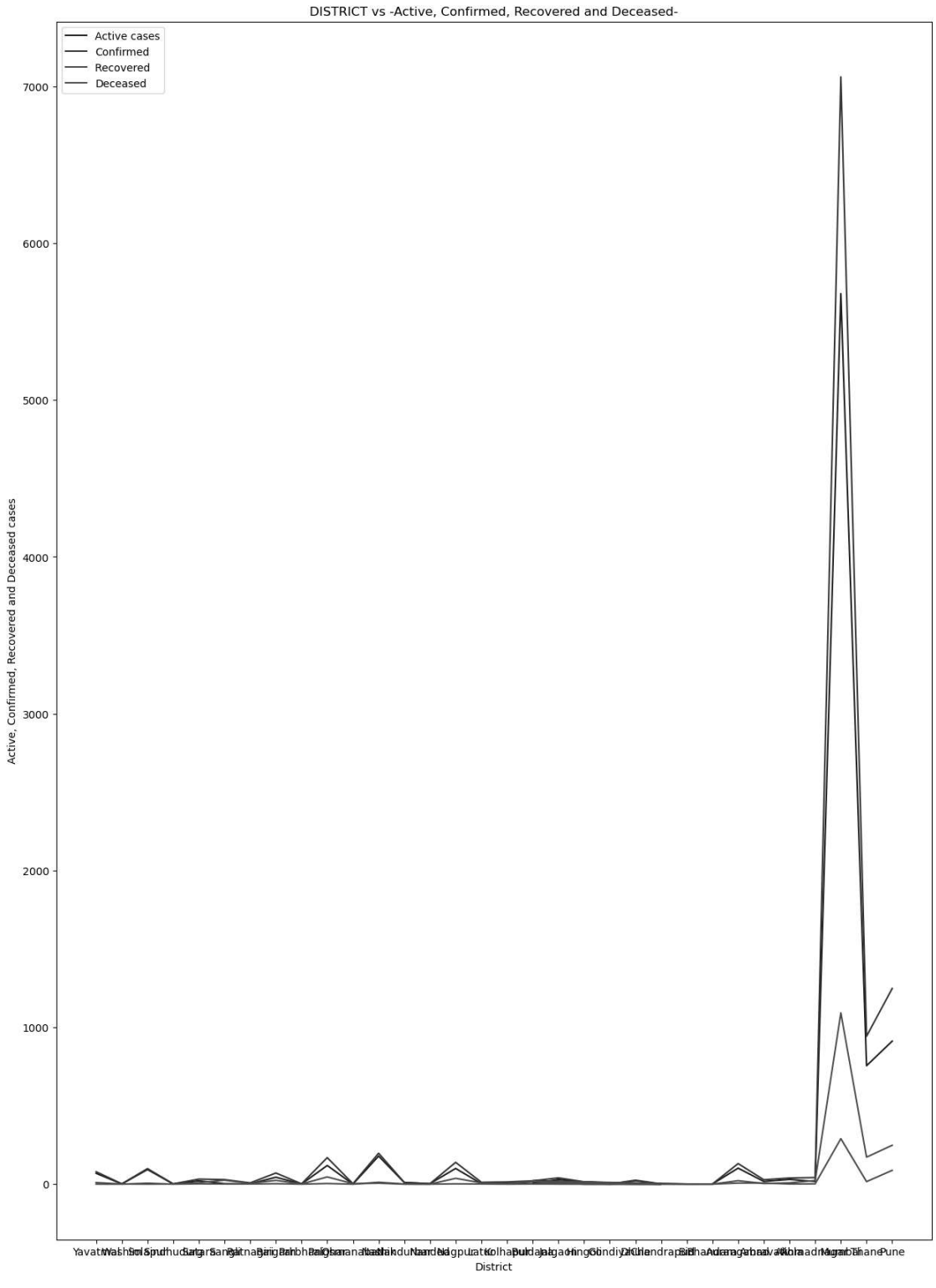
In [103]:
```
# District vs Active cases
plt.plot(X, Y, label="Active cases", color ="blue")
# District vs Confirmed cases
plt.plot(X, R, label="Confirmed", color = "purple")
# District vs Recovered cases
plt.plot(X, D, label="Recovered " , color ="Green")
# District vs Deceased patients
plt.plot(X, W, label="Deceased", color ="red")
plt.xlabel('District')
plt.ylabel('Active, Confirmed, Recovered and Deceased cases')
plt.title('DISTRICT vs -Active, Confirmed, Recovered and Deceased-')
plt.legend()
plt.show()
```



Adding legends to the graph

In [104]:

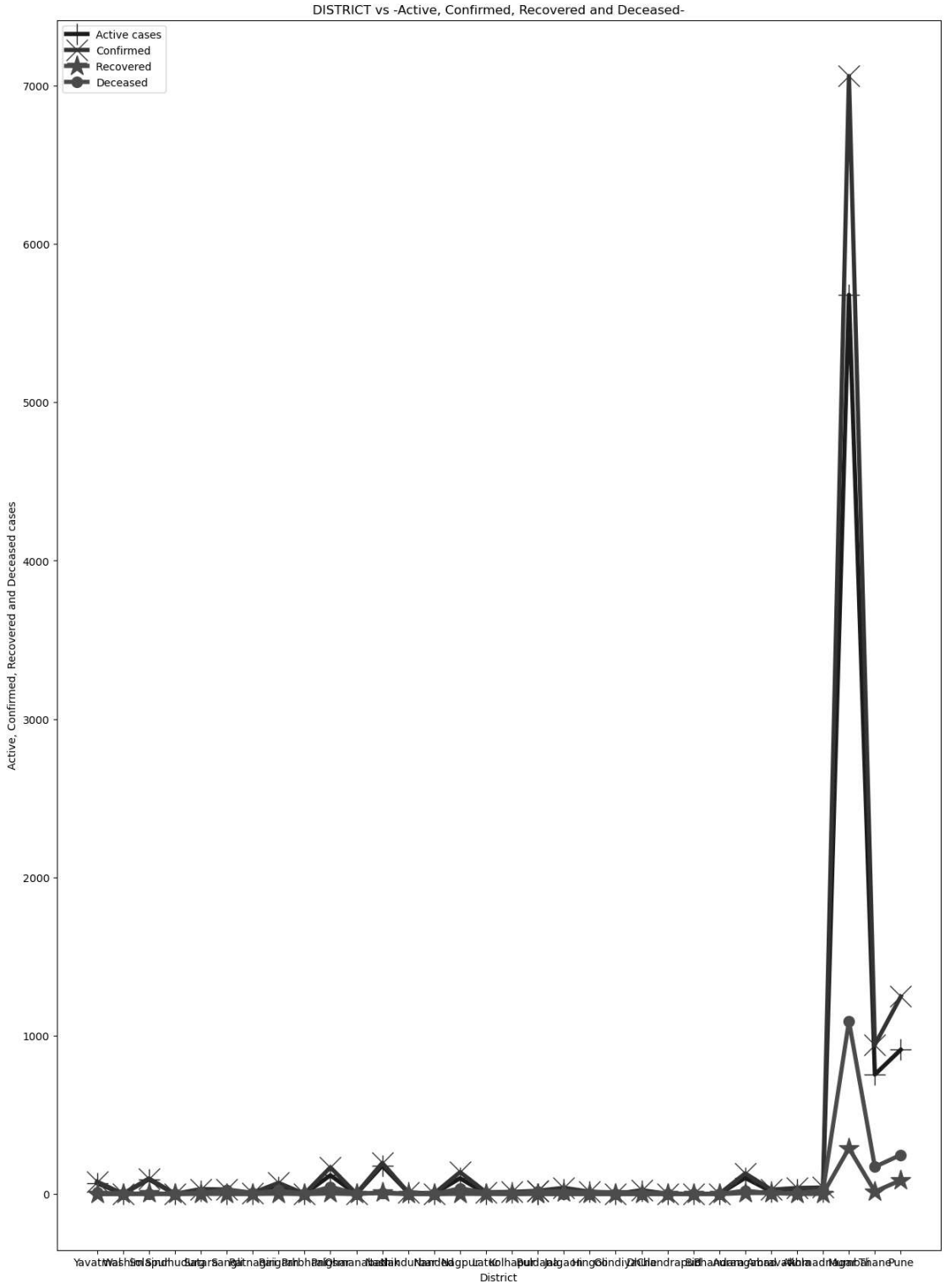
```
plt.figure(figsize=(15,21))
# District vs Active cases
plt.plot(X, Y, label="Active cases", color ="blue")
# District vs Confirmed cases
plt.plot(X, R, label="Confirmed", color = "purple")
# District vs Recovered cases
plt.plot(X, D, label="Recovered " , color ="Green")
# District vs Deceased patients
plt.plot(X, W, label="Deceased", color ="red")
plt.xlabel('District')
plt.ylabel('Active, Confirmed, Recovered and Deceased cases')
plt.title('DISTRICT vs -Active, Confirmed, Recovered and Deceased-')
plt.legend()
plt.show()
```



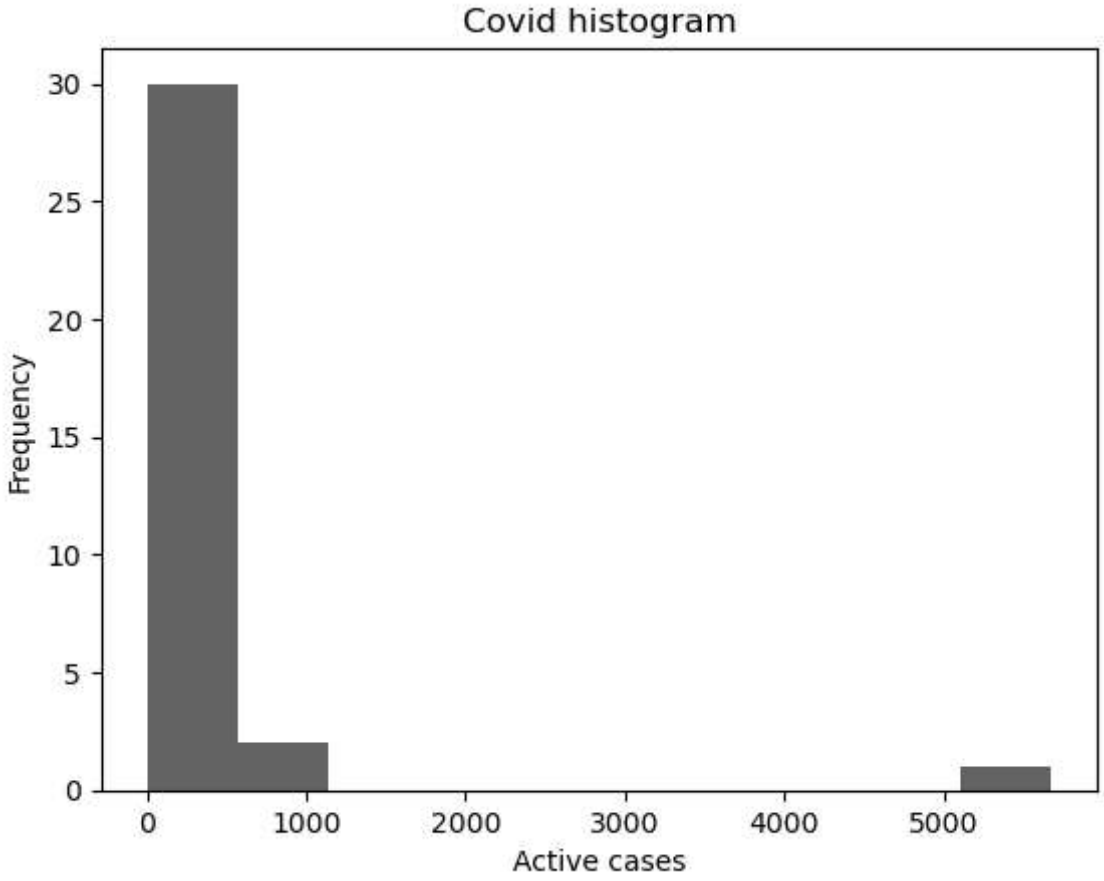
Customisation

In [105]:

```
plt.figure(figsize=(15,21))
# District vs Active cases
# By writing linewidth, we can increase or decrease the width of the line in
# By writing marker, we can designate the end point of the line graph as a ma
plt.plot(X, Y, label="Active cases", color ="blue", linewidth = 4, marker ='+
# District vs Confirmed cases
plt.plot(X, R, label="Confirmed", color = "purple",linewidth = 4, marker ='x'
# District vs Recovered cases
plt.plot(X, D, label="Recovered " , color ="Green", linewidth = 4, marker ='*
# District vs Deceased patients
plt.plot(X, W, label="Deceased", color ="red", linewidth = 4, marker ='.' , m
plt.xlabel('District')
plt.ylabel('Active, Confirmed, Recovered and Deceased cases')
plt.title('DISTRICT vs -Active, Confirmed, Recovered and Deceased-')
plt.legend()
plt.show()
```

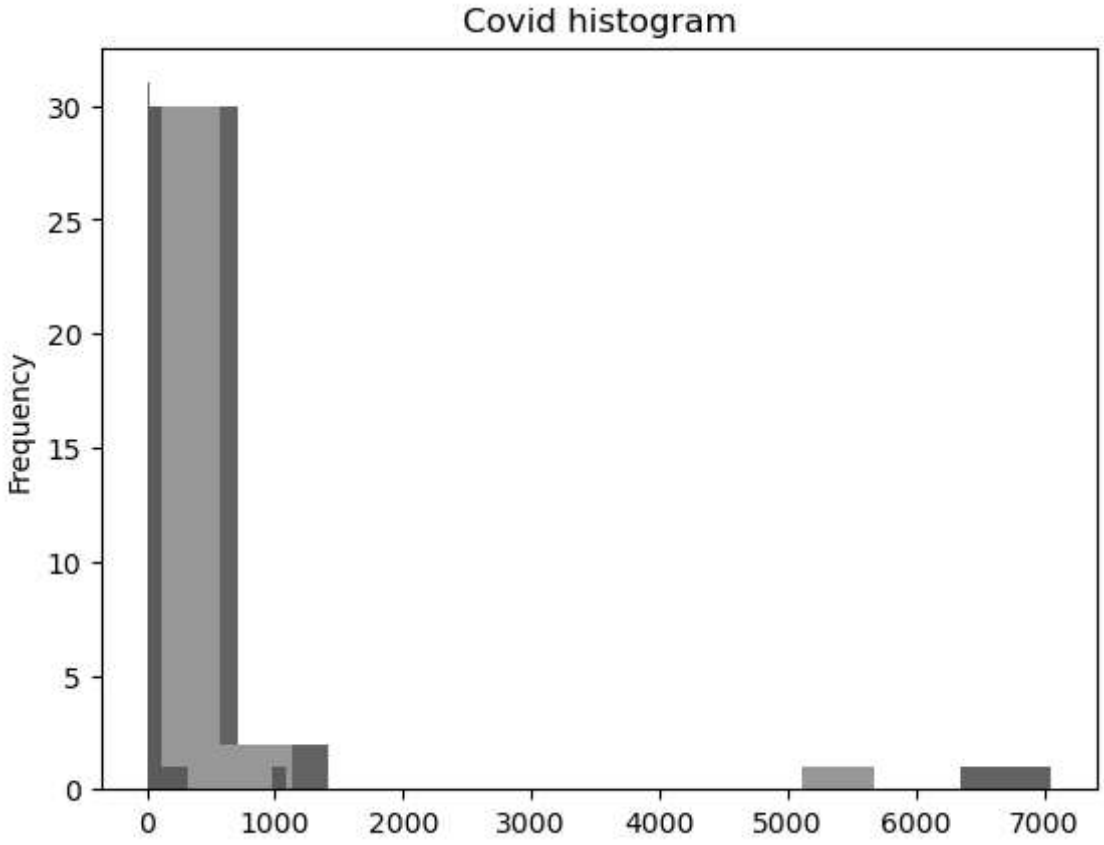

DISTRICT vs -Active, Confirmed, Recovered and Deceased-
Active cases
Confirmed
Recovered
Deceased
Active, Confirmed, Recovered and Deceased cases
0
1000
2000
3000
4000
5000
6000
7000
District
Pune

In [106]:
```
plt.hist(Y)
plt.xlabel("Active cases")
plt.ylabel("Frequency")
plt.title("Covid histogram")
plt.show()
```



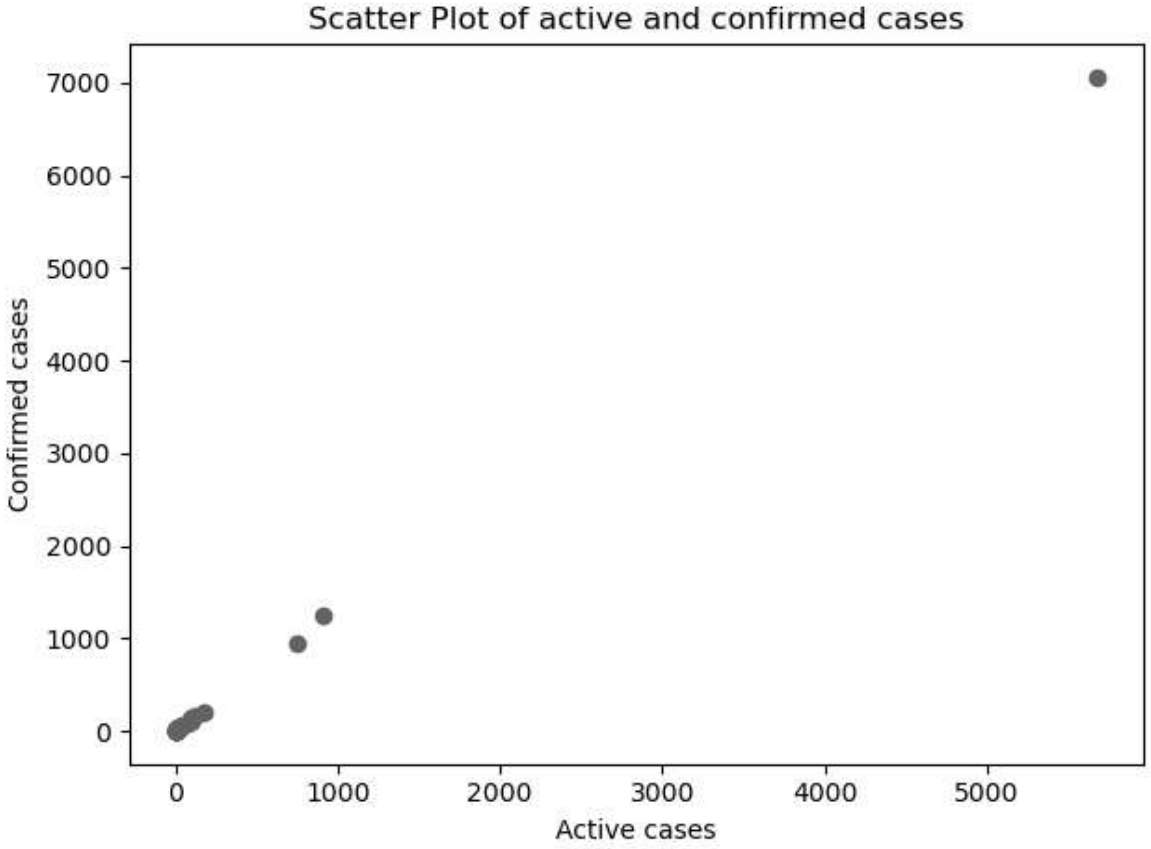
Histogram

In [107]:
```
plt.hist(R)
plt.hist(Y)
plt.hist(D)
plt.hist(W)

plt.ylabel("Frequency")
plt.title("Covid histogram")
plt.show()
```

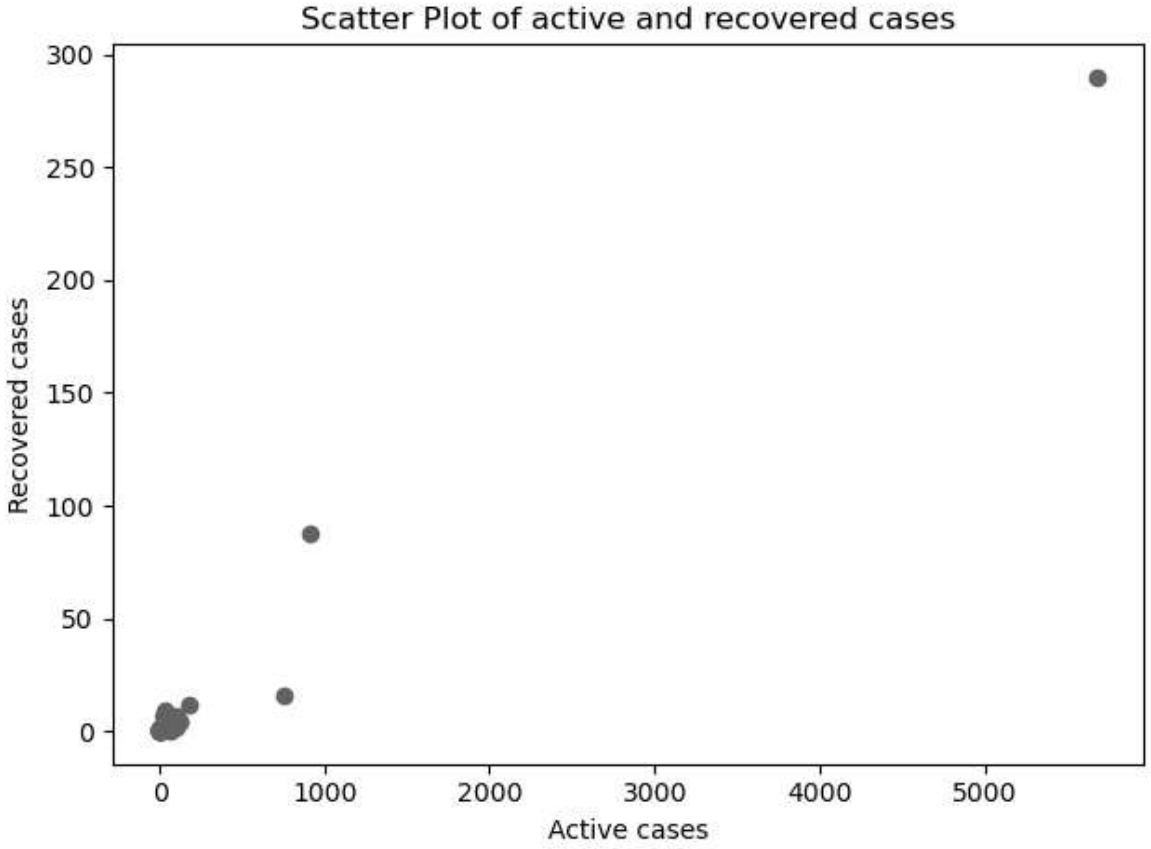


Scatter plot

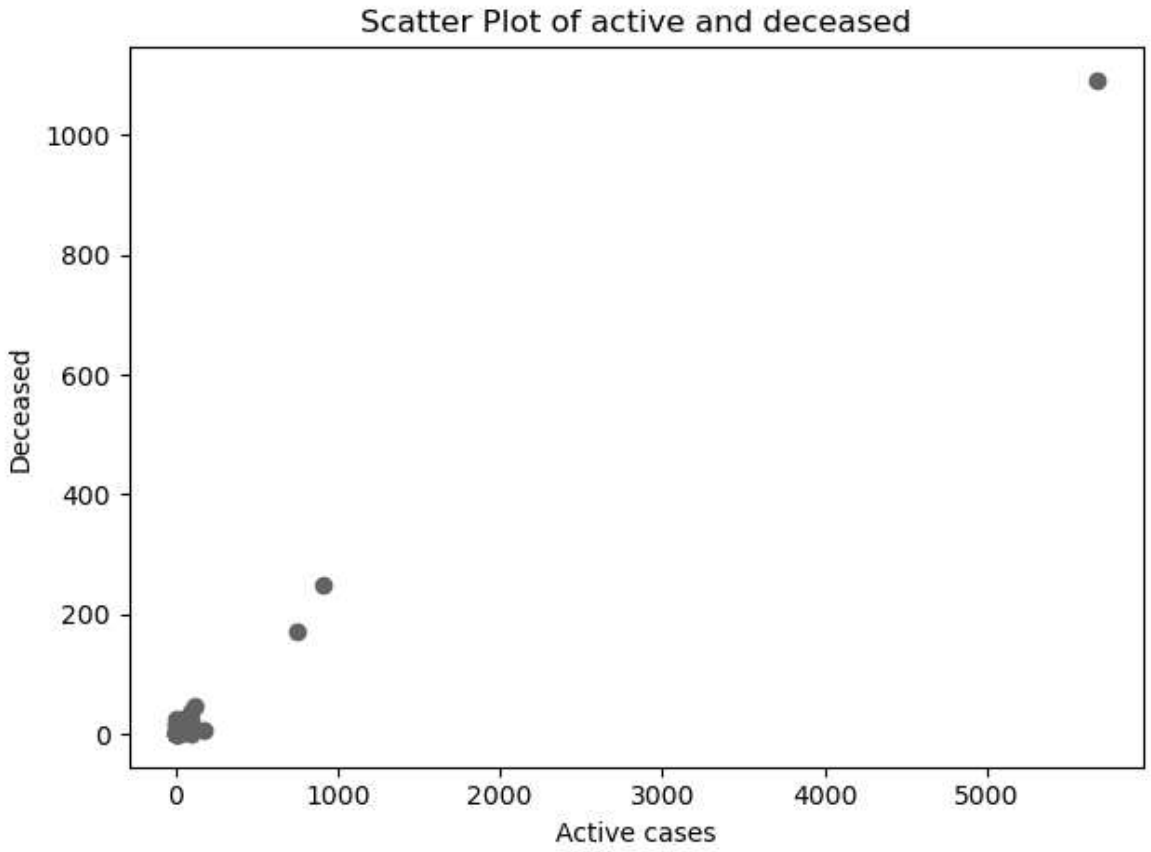
In [108]:
```
plt.scatter(Y, R)
plt.xlabel("Active cases")
plt.ylabel("Confirmed cases")
plt.title("Scatter Plot of active and confirmed cases")
plt.tight_layout()
plt.show()
```



In [109]:
```
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()
```

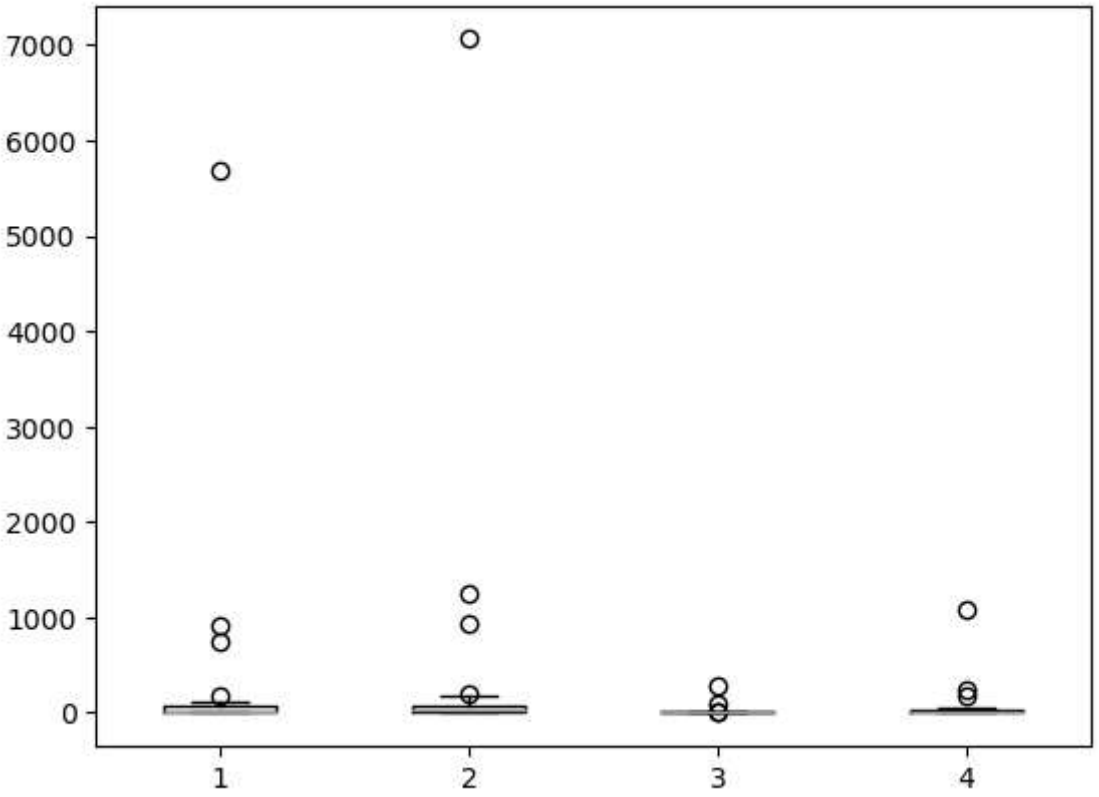


In [110]:
```
plt.scatter(Y, W)
plt.xlabel("Active cases")
plt.ylabel("Deceased")
plt.title("Scatter Plot of active and deceased")
plt.tight_layout()
plt.show()
```



BoxPlot

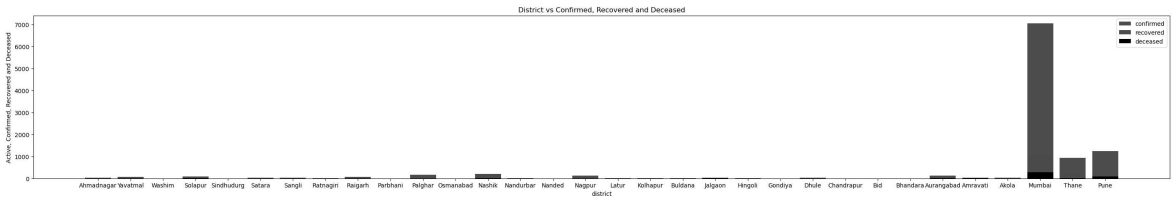
In [111]:
```
collections = [Y, R, D, W]
plt.boxplot(collections)
plt.show()
```



Bar graph

In [112]:
```
data = pd.read_csv('district.csv')
#datacalling

x=data['district']
y=data['confirmed cases']
h=data['recovered']
j=data['deceased']
#assigning each parameter to an alphabate

plt.figure(figsize=(35,5))
#assigning graph size

plt.bar(x,y,color=['red'],label="confirmed")
#confirmed cases districtwise indicating with red color

plt.bar(x,h,color=['green'],label="recovered")
#recovered cases districtwise indicating with green color

plt.bar(x,j,color=['black'],label="deceased")
#deceased cases districtwise indicating with black color

plt.xlabel("district")
#labeling x-axis

plt.ylabel("Active, Confirmed, Recovered and Deceased")
#labeling y-axis

plt.title('District vs Confirmed, Recovered and Deceased')
#giving title to graph

plt.legend()
#visulization of legend
```

Out[112]: <matplotlib.legend.Legend at 0x233899398b0>



In [113]:
```
data = pd.read_csv('district.csv')
data.groupby(by="district")["active cases"].nlargest(5)
#sorting of districtwise active cases, picking highest 5 active cases
```

Out[113]:
```
district
Ahmadnagar  0       17
            30      17
Akola       29      30
Amravati    28      17
Aurangabad  27     102
Bhandara    26       1
Bid         25       0
Buldana     18       3
            24       3
Chandrapur  23       0
Dhule       22      22
Gondiya     21       0
Hingoli     20      14
Jalgaon     19      30
Kolhapur    17      10
Latur       16       3
Mumbai      31    5679
Nagpur      15     100
Nanded      14       3
Nandurbar   13      10
Nashik      12     179
Osmanabad   11       0
Palghar     10     119
Parbhani    9        1
Pune        33     912
Raigarh     8       44
Ratnagiri   7        2
Sangli      6        3
Satara      5       21
Sindhudurg  4        1
Solapur     3       93
Thane       32     755
Washim      2        1
Yavatmal    1       69
Name: active cases, dtype: int64
```

In [114]:
```
data.groupby(by="active cases")["recovered"].nlargest(5)
#sorting of recovered cases from active cases
```

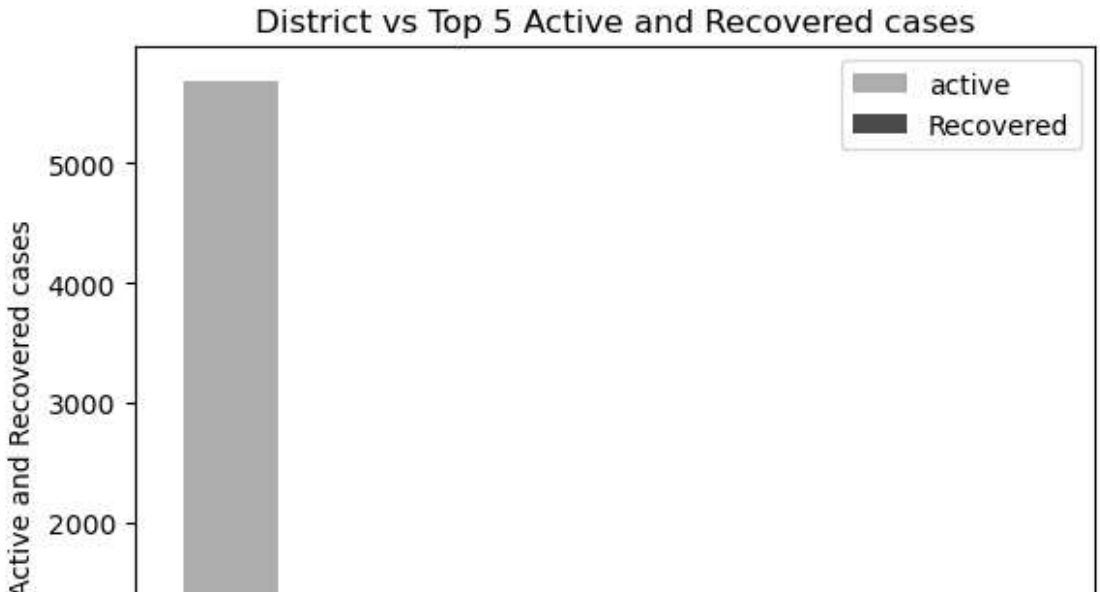
Out[114]:

active cases		
0	11	3
	23	2
	21	1
	25	1
1	2	1
	4	1
	9	1
	26	0
2	7	5
3	6	25
	18	17
	24	17
	16	8
	14	0
10	17	4
	13	0
14	20	1
17	0	23
	30	23
	28	4
21	5	9
22	22	0
30	29	8
	19	1
44	8	24
69	1	10
93	3	0
100	15	37
102	27	22
119	10	46
179	12	6
755	32	172
912	33	248
5679	31	1092

Name: recovered, dtype: int64

In [115]:
```
data = pd.DataFrame({'District': ['Mumbai', 'Pune', 'Thane', 'Nashik', 'Palgh
                     'Recovered':[1092,248,172,6,46]})
#assigning values of top 5 districts with active cases

ax = data.plot(x="District",y="active",kind="bar", color="orange")
#ploting active cases destrictwise

data.plot(x="District", y="Recovered", kind="bar", ax=ax, color="green")
#ploting recovered cases destrictwise

plt.ylabel("Active and Recovered cases")
#labeling y-axis

plt.title('District vs Top 5 Active and Recovered cases')
#giving title to graph
```

Out[115]: Text(0.5, 1.0, 'District vs Top 5 Active and Recovered cases')



In [ ]: