SDA_Project (3)

November 17, 2022

[ ]:
```
pip install geopandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: geopandas in /usr/local/lib/python3.7/dist-
packages (0.10.2)
Requirement already satisfied: fiona>=1.8 in /usr/local/lib/python3.7/dist-
packages (from geopandas) (1.8.22)
Requirement already satisfied: pyproj>=2.2.0 in /usr/local/lib/python3.7/dist-
packages (from geopandas) (3.2.1)
Requirement already satisfied: pandas>=0.25.0 in /usr/local/lib/python3.7/dist-
packages (from geopandas) (1.3.5)
Requirement already satisfied: shapely>=1.6 in /usr/local/lib/python3.7/dist-
packages (from geopandas) (1.8.5.post1)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.7/dist-
packages (from fiona>=1.8->geopandas) (1.15.0)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.7/dist-
packages (from fiona>=1.8->geopandas) (7.1.2)
Requirement already satisfied: click-plugins>=1.0 in
/usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (1.1.1)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.7/dist-
packages (from fiona>=1.8->geopandas) (22.1.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages
(from fiona>=1.8->geopandas) (2022.9.24)
Requirement already satisfied: munch in /usr/local/lib/python3.7/dist-packages
(from fiona>=1.8->geopandas) (2.5.0)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.7/dist-
packages (from fiona>=1.8->geopandas) (0.7.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-
packages (from fiona>=1.8->geopandas) (57.4.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-
packages (from pandas>=0.25.0->geopandas) (2022.6)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.7/dist-packages (from pandas>=0.25.0->geopandas) (2.8.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-
packages (from pandas>=0.25.0->geopandas) (1.21.6)
```

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import geopandas as gp
from scipy import stats
from scipy.stats import pearsonr
from sklearn.datasets import make_circles
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

[ ]:
```
Rain=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/SDA/
 ↪NWH-CRU_25km_precipitation_1901-2020-monthly.csv')
Temp=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/SDA/
 ↪NWH-CRU_25km_temp-1901-2020-monthly.csv')
```

[ ]:
```
Rain.head()
```

[ ]:

	Unnamed: 0	0	1	2	3	4	5	6	7	8	\
0	0	79.00	29.0	55.75	25.30	19.65	1.15	30.40	48.75	292.30	
1	1	79.25	29.0	59.65	26.55	21.85	1.25	36.00	53.80	305.55	
2	2	79.50	29.0	63.50	29.95	25.80	1.50	44.60	58.90	304.10	
3	3	79.75	29.0	66.55	31.15	26.90	1.60	49.65	64.55	325.45	
4	4	80.00	29.0	66.70	33.10	26.70	1.85	57.10	68.10	347.20	

	…	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441
0	…	48.70	22.25	38.40	121.9	284.35	287.90	20.05	0.25	0.55	2.25
1	…	54.35	26.90	45.55	132.0	294.40	292.65	19.95	0.15	0.60	2.35
2	…	66.90	37.85	56.95	142.6	299.85	292.45	20.40	0.15	0.65	2.70
3	…	70.50	40.65	63.75	155.2	318.60	301.00	20.95	0.15	0.60	2.75
4	…	70.40	44.50	74.05	164.5	330.85	306.80	21.75	0.30	0.60	3.00

[5 rows x 1443 columns]

[ ]:
```
Temp.head()
```

[ ]:

	Unnamed: 0	0	1	2	3	4	5	6	7	8	\
0	0	79.00	29.0	12.80	15.35	22.35	27.95	31.65	33.75	30.65	
1	1	79.25	29.0	12.40	14.90	21.80	27.30	30.95	33.05	30.00	
2	2	79.50	29.0	11.45	13.75	20.40	25.75	29.25	31.40	28.40	
3	3	79.75	29.0	11.50	13.85	20.40	25.65	29.05	31.30	28.30	
4	4	80.00	29.0	11.50	13.85	20.35	25.55	28.95	31.20	28.30	

	…	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441
0	…	21.70	28.40	31.60	31.55	30.35	30.00	29.80	26.80	19.95	16.05
1	…	21.20	27.75	30.90	30.80	29.65	29.30	29.20	26.20	19.50	15.70
2	…	19.85	26.25	29.25	29.10	28.10	27.75	27.65	24.75	18.35	14.85

```
3  …  19.85  26.15  29.15  29.00  28.05  27.70  27.60  24.70  18.35  14.85
4  …  19.90  26.15  29.05  28.90  28.05  27.70  27.55  24.70  18.40  14.90

[5 rows x 1443 columns]
```

[ ]:
```
Temp=Temp.iloc[:,1:]
Rain=Rain.iloc[:,1:]
```

[ ]:
```
Temp.head()
```

[ ]:
```
       0     1      2      3      4      5      6      7      8     9  …  \
0  79.00  29.0  12.80  15.35  22.35  27.95  31.65  33.75  30.65  29.5  …
1  79.25  29.0  12.40  14.90  21.80  27.30  30.95  33.05  30.00  28.8  …
2  79.50  29.0  11.45  13.75  20.40  25.75  29.25  31.40  28.40  27.2  …
3  79.75  29.0  11.50  13.85  20.40  25.65  29.05  31.30  28.30  27.1  …
4  80.00  29.0  11.50  13.85  20.35  25.55  28.95  31.20  28.30  27.1  …

    1432   1433   1434   1435   1436   1437   1438   1439   1440   1441
0  21.70  28.40  31.60  31.55  30.35  30.00  29.80  26.80  19.95  16.05
1  21.20  27.75  30.90  30.80  29.65  29.30  29.20  26.20  19.50  15.70
2  19.85  26.25  29.25  29.10  28.10  27.75  27.65  24.75  18.35  14.85
3  19.85  26.15  29.15  29.00  28.05  27.70  27.60  24.70  18.35  14.85
4  19.90  26.15  29.05  28.90  28.05  27.70  27.55  24.70  18.40  14.90

[5 rows x 1442 columns]
```

[ ]:
```
Rain.head()
```

[ ]:
```
       0     1      2      3      4     5      6      7       8       9  …  \
0  79.00  29.0  55.75  25.30  19.65  1.15  30.40  48.75  292.30  425.80  …
1  79.25  29.0  59.65  26.55  21.85  1.25  36.00  53.80  305.55  443.80  …
2  79.50  29.0  63.50  29.95  25.80  1.50  44.60  58.90  304.10  452.20  …
3  79.75  29.0  66.55  31.15  26.90  1.60  49.65  64.55  325.45  473.60  …
4  80.00  29.0  66.70  33.10  26.70  1.85  57.10  68.10  347.20  493.55  …

    1432   1433   1434   1435    1436    1437   1438  1439  1440  1441
0  48.70  22.25  38.40  121.9  284.35  287.90  20.05  0.25  0.55  2.25
1  54.35  26.90  45.55  132.0  294.40  292.65  19.95  0.15  0.60  2.35
2  66.90  37.85  56.95  142.6  299.85  292.45  20.40  0.15  0.65  2.70
3  70.50  40.65  63.75  155.2  318.60  301.00  20.95  0.15  0.60  2.75
4  70.40  44.50  74.05  164.5  330.85  306.80  21.75  0.30  0.60  3.00

[5 rows x 1442 columns]
```

[ ]:
```
print(Temp.shape,Rain.shape)
```

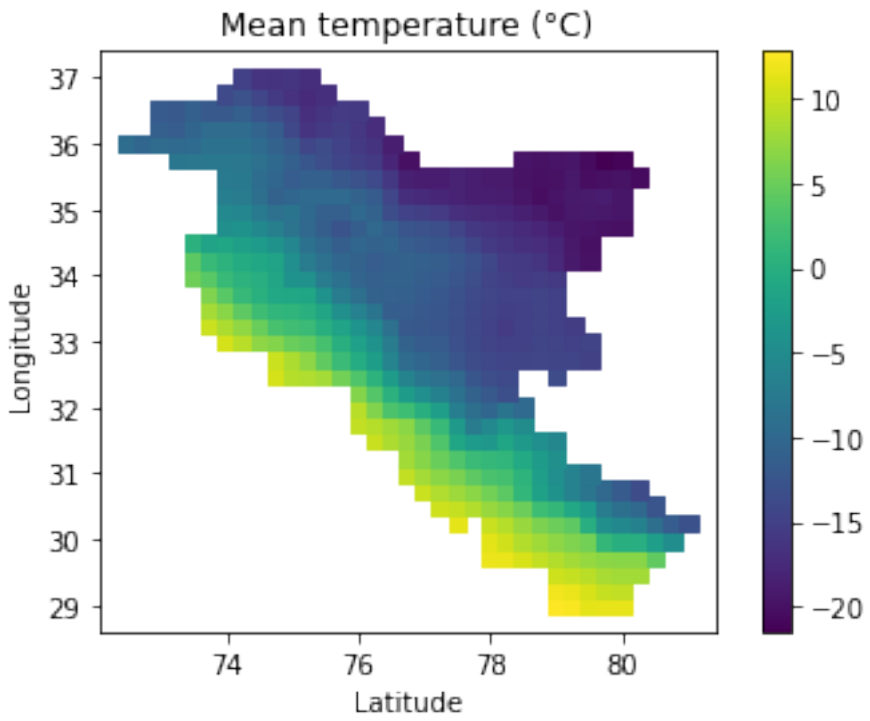
```
(525, 1442) (525, 1442)
```

[ ]:
```
gdf = gp.GeoDataFrame(Temp, geometry=gp.points_from_xy(Temp['0'], Temp['1']))
```

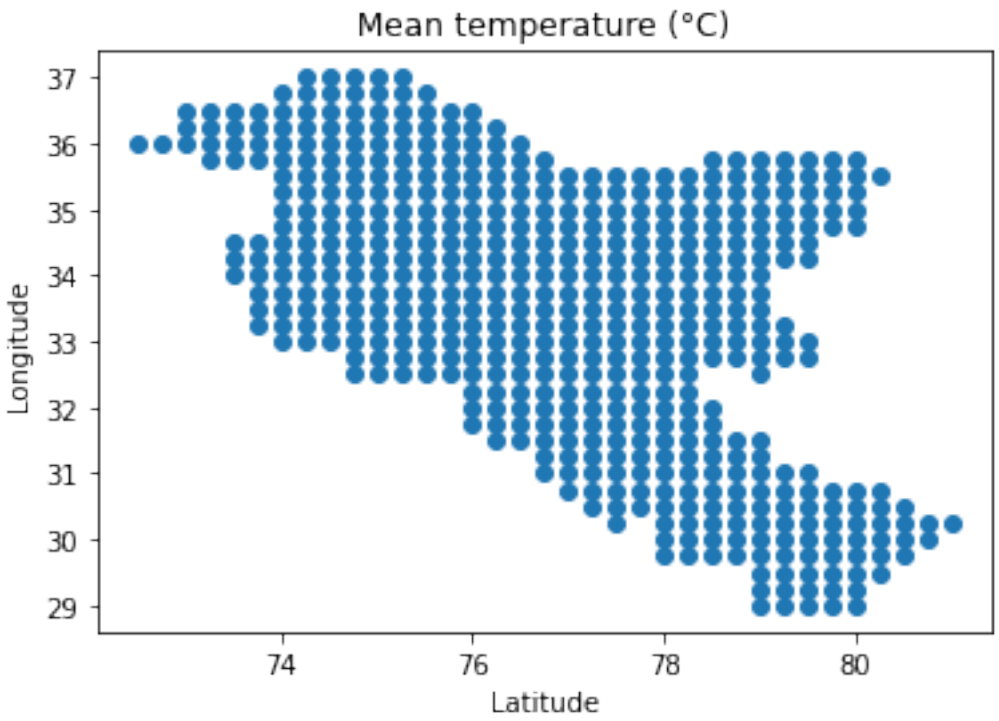
[ ]:
```
gdf.head()
```

[ ]:
```
        0     1      2      3      4      5      6      7      8     9  …  \
0  79.00  29.0  12.80  15.35  22.35  27.95  31.65  33.75  30.65  29.5  …
1  79.25  29.0  12.40  14.90  21.80  27.30  30.95  33.05  30.00  28.8  …
2  79.50  29.0  11.45  13.75  20.40  25.75  29.25  31.40  28.40  27.2  …
3  79.75  29.0  11.50  13.85  20.40  25.65  29.05  31.30  28.30  27.1  …
4  80.00  29.0  11.50  13.85  20.35  25.55  28.95  31.20  28.30  27.1  …

     1433   1434   1435   1436   1437   1438   1439   1440   1441  \
0  28.40  31.60  31.55  30.35  30.00  29.80  26.80  19.95  16.05
1  27.75  30.90  30.80  29.65  29.30  29.20  26.20  19.50  15.70
2  26.25  29.25  29.10  28.10  27.75  27.65  24.75  18.35  14.85
3  26.15  29.15  29.00  28.05  27.70  27.60  24.70  18.35  14.85
4  26.15  29.05  28.90  28.05  27.70  27.55  24.70  18.40  14.90

                     geometry
0  POINT (79.00000 29.00000)
1  POINT (79.25000 29.00000)
2  POINT (79.50000 29.00000)
3  POINT (79.75000 29.00000)
4  POINT (80.00000 29.00000)

[5 rows x 1443 columns]
```

[ ]:
```
ax=gdf.plot(Temp['2'],legend=True,marker='s',markersize=45)
plt.title("Mean temperature (°C)")
plt.xlabel("Latitude")
plt.ylabel("Longitude")
plt.show()
```



[ ]:
```
plt.scatter(Temp['0'],Temp['1'])
plt.title("Mean temperature (°C)")
plt.xlabel("Latitude")
plt.ylabel("Longitude")
plt.show()
```



[ ]:
```
Temp
```

[ ]:
```
          0     1      2      3      4      5      6      7      8      9  …  \
0     79.00  29.0  12.80  15.35  22.35  27.95  31.65  33.75  30.65  29.50  …
1     79.25  29.0  12.40  14.90  21.80  27.30  30.95  33.05  30.00  28.80  …
2     79.50  29.0  11.45  13.75  20.40  25.75  29.25  31.40  28.40  27.20  …
3     79.75  29.0  11.50  13.85  20.40  25.65  29.05  31.30  28.30  27.10  …
4     80.00  29.0  11.50  13.85  20.35  25.55  28.95  31.20  28.30  27.10  …
..      …    …     …      …      …      …      …      …      …      …  …
520   74.25  37.0 -14.95 -14.05  -5.50  -2.15   1.10   3.90   9.40   9.75  …
521   74.50  37.0 -16.35 -15.65  -7.10  -3.75  -0.60   2.40   7.75   8.35  …
522   74.75  37.0 -16.50 -15.70  -7.05  -3.70  -0.55   2.60   7.80   8.45  …
523   75.00  37.0 -16.15 -15.45  -6.45  -3.20  -0.10   3.10   8.25   8.90  …
524   75.25  37.0 -17.00 -16.05  -7.05  -3.80  -0.75   2.25   7.75   8.20  …

      1433   1434   1435   1436   1437   1438   1439   1440   1441  \
0    28.40  31.60  31.55  30.35  30.00  29.80  26.80  19.95  16.05
1    27.75  30.90  30.80  29.65  29.30  29.20  26.20  19.50  15.70
2    26.25  29.25  29.10  28.10  27.75  27.65  24.75  18.35  14.85
3    26.15  29.15  29.00  28.05  27.70  27.60  24.70  18.35  14.85
4    26.15  29.05  28.90  28.05  27.70  27.55  24.70  18.40  14.90
..     …      …      …      …      …      …      …      …      …
520  -1.00   2.95   6.30  10.20  10.85   5.75  -0.60  -7.60 -12.70
```

```
521  -2.65   1.20   4.60   8.60   9.35   4.25  -2.20  -9.15 -14.15
522  -2.60   1.25   4.65   8.55   9.40   4.30  -2.20  -9.20 -14.20
523  -2.00   1.85   5.20   9.10   9.90   4.80  -1.80  -8.85 -14.00
524  -2.60   1.00   4.55   8.60   9.30   4.35  -2.20  -9.45 -14.50

                      geometry
0    POINT (79.00000 29.00000)
1    POINT (79.25000 29.00000)
2    POINT (79.50000 29.00000)
3    POINT (79.75000 29.00000)
4    POINT (80.00000 29.00000)
..                         …
520  POINT (74.25000 37.00000)
521  POINT (74.50000 37.00000)
522  POINT (74.75000 37.00000)
523  POINT (75.00000 37.00000)
524  POINT (75.25000 37.00000)

[525 rows x 1443 columns]
```

[ ]:
```
Avg_all=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/SDA/Data/
↪drive-download-20221110T140448Z-001.zip (Unzipped Files)/
↪skewness_kurtosis_rainfall_temperature_average_year_1901_2020.csv')
```

[ ]:
```
Avg_all.head()
```

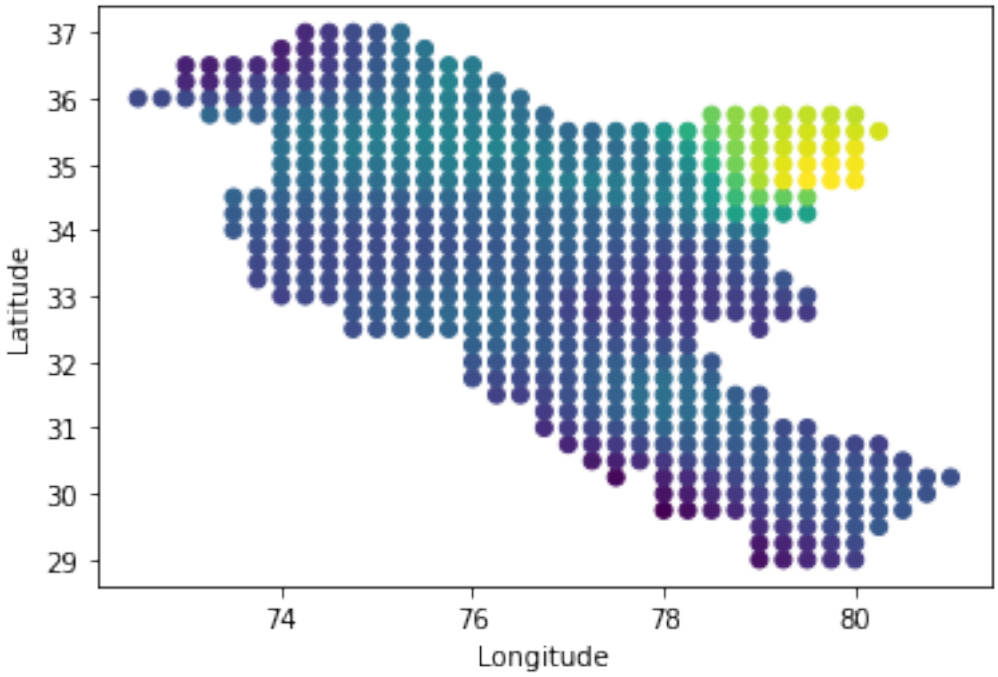
[ ]:
```
   Longitude  Latitude  rain_skewness  temperature_skewness  rain_kurtosis  \
0      79.00      29.0       0.105740              0.380739       0.019206
1      79.25      29.0       0.166450              0.379858       0.148737
2      79.50      29.0       0.220982              0.378947       0.414965
3      79.75      29.0       0.257823              0.383174       0.454959
4      80.00      29.0       0.322586              0.392282       0.521756

   temperature_kurtosis
0              0.327284
1              0.297456
2              0.274039
3              0.234783
4              0.230557
```

1 For Rain only

Note:- Giving expected outcome

[ ]:
```
plt.scatter(Avg_all['Longitude'],Avg_all['Latitude'],c=Avg_all['rain_skewness'])
plt.ylabel("Latitude")
plt.xlabel("Longitude")
```

```
plt.show()
```

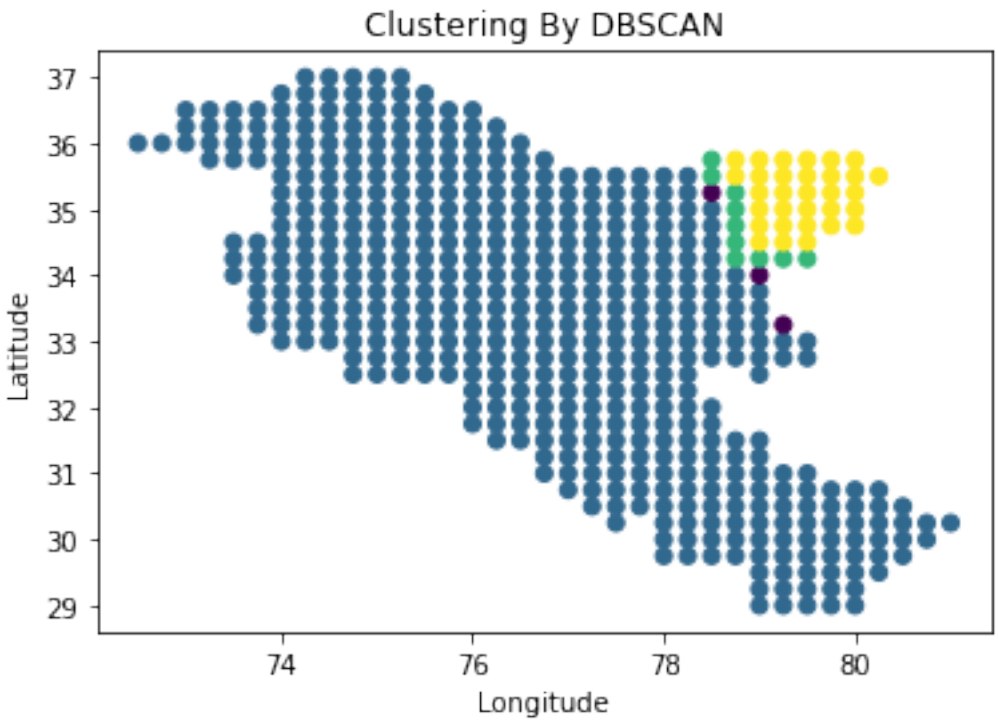


[ ]:
```
X=list(zip(Avg_all['rain_skewness'],Avg_all['rain_kurtosis']))
X=np.array(X)
```

1.0.1 By Using DBSCAN

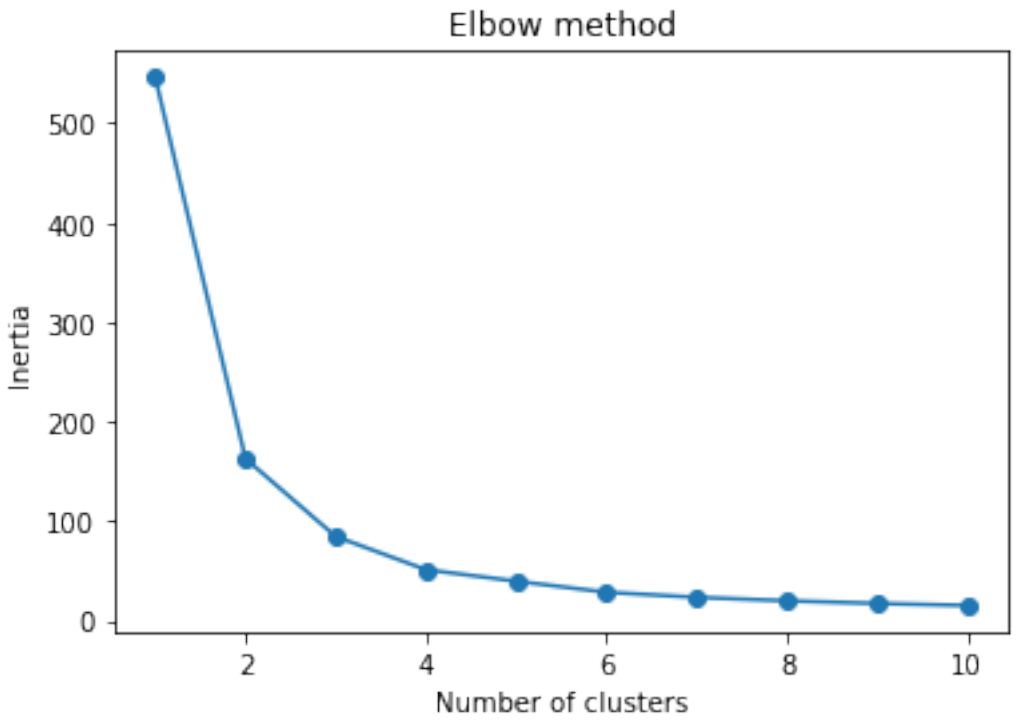
[ ]:
```
db = DBSCAN(eps=0.3, min_samples=10).fit(X)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
```

[ ]:
```
plt.scatter(Avg_all['Longitude'],Avg_all['Latitude'],c=labels)
plt.title("Clustering By DBSCAN")
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```

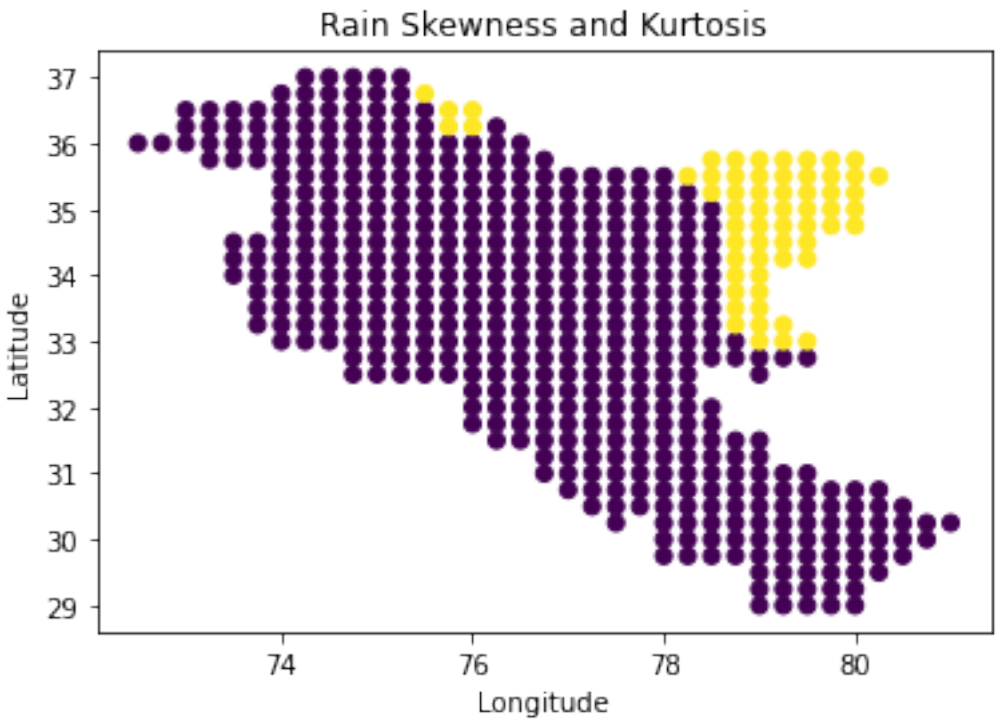


1.0.2 By Using K-Mean

[ ]:
```
data = X
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



[ ]:
```
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)

plt.scatter(Avg_all['Longitude'],Avg_all['Latitude'],c=kmeans.labels_)
plt.title('Rain Skewness and Kurtosis')
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```



2 For Temperature

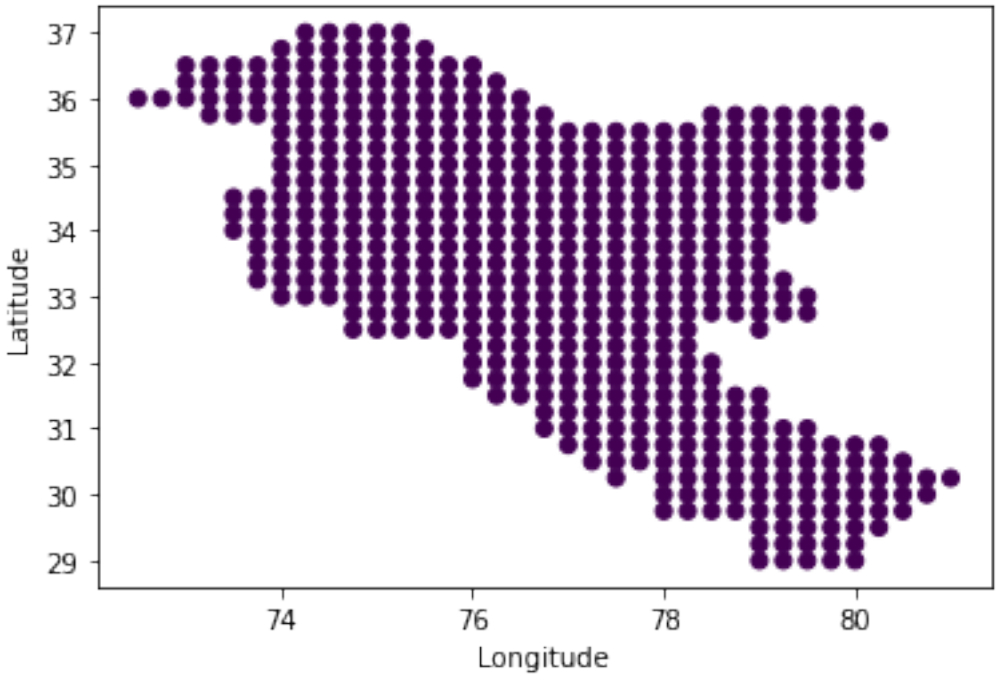
Note:- Not Giving expected outcome

[ ]:
```
Y=list(zip(Avg_all['temperature_skewness'],Avg_all['temperature_kurtosis']))
Y=np.array(Y)
#Y= StandardScaler().fit_transform(Y)
```

2.0.1 By Using DBSCAN

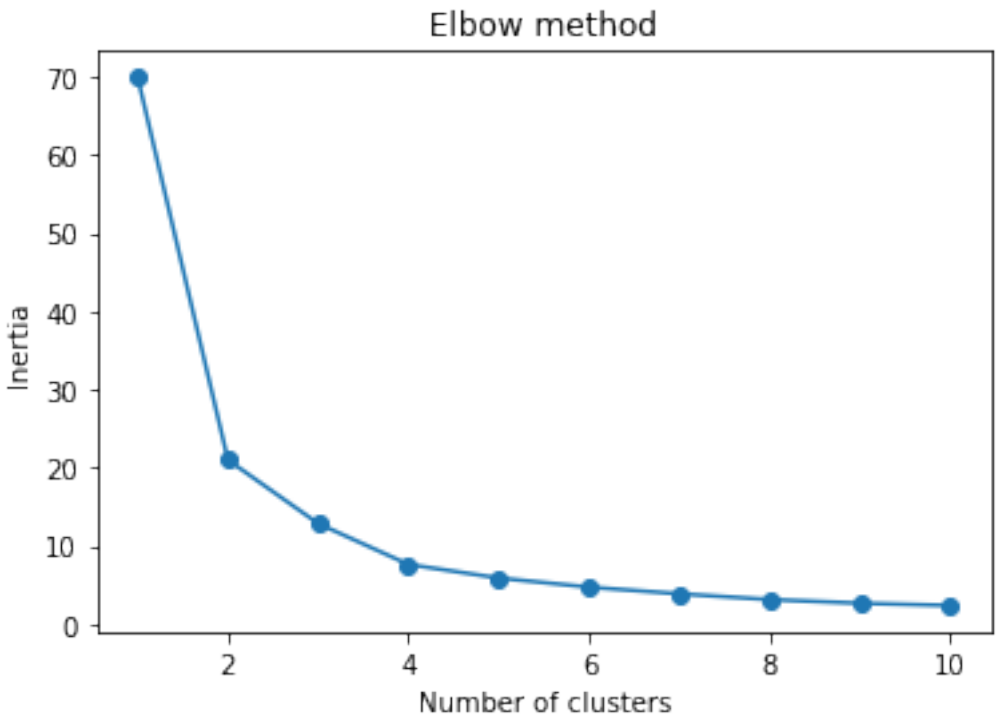
[ ]:
```
db = DBSCAN(eps=0.8, min_samples=20).fit(Y)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
```

[ ]:
```
plt.scatter(Avg_all['Longitude'],Avg_all['Latitude'],c=labels)
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```

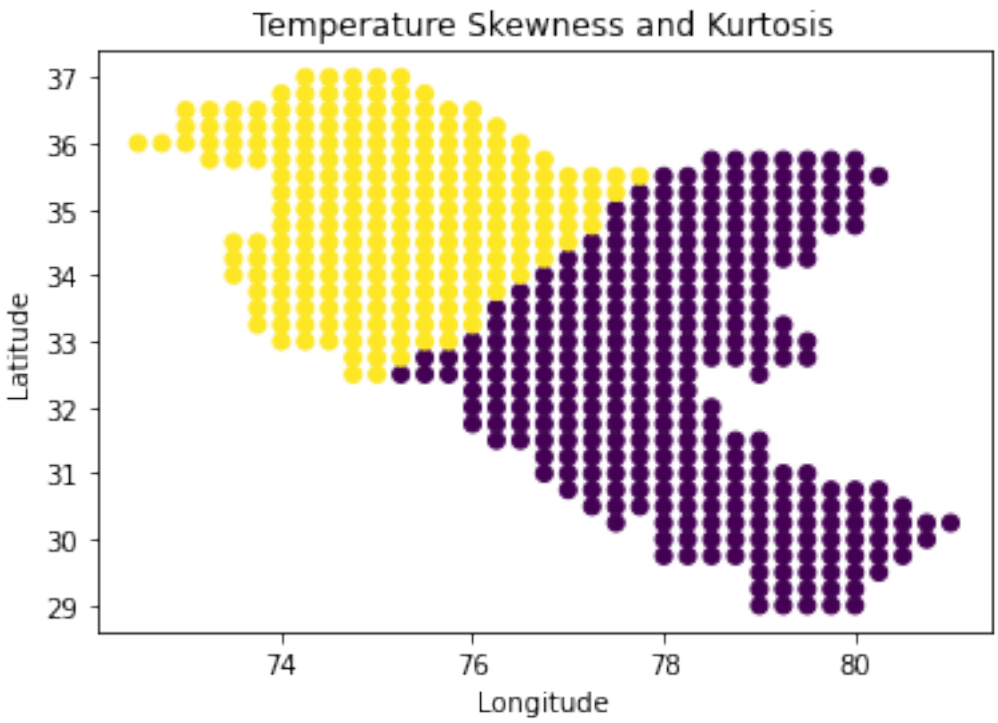


2.0.2 By Using K_Mean

[ ]:
```
data = Y
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



[ ]:
```
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)

plt.scatter(Avg_all['Longitude'],Avg_all['Latitude'],c=kmeans.labels_)
plt.title('Temperature Skewness and Kurtosis')
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```



3 Consedering both Temperature and Rain

Note:- Giving Expected outcome

[ ]:
```
TR=list(zip(Avg_all['temperature_skewness'],Avg_all['temperature_kurtosis'],Avg_all['rain_skew
TR=np.array(TR)
#Y= StandardScaler().fit_transform(Y)
```

[ ]:
```
TR
```

[ ]:
```
array([[ 0.3807386 ,  0.32728449,  0.10573978,  0.01920637],
       [ 0.37985762,  0.297456  ,  0.16644982,  0.14873687],
       [ 0.37894728,  0.27403948,  0.22098168,  0.41496544],
       …,
       [ 0.17100296, -0.67556919,  0.40772777,  0.50929839],
       [ 0.18338106, -0.66487954,  0.44773353,  0.66342212],
       [ 0.19559583, -0.6429112 ,  0.60713284,  1.38031993]])
```

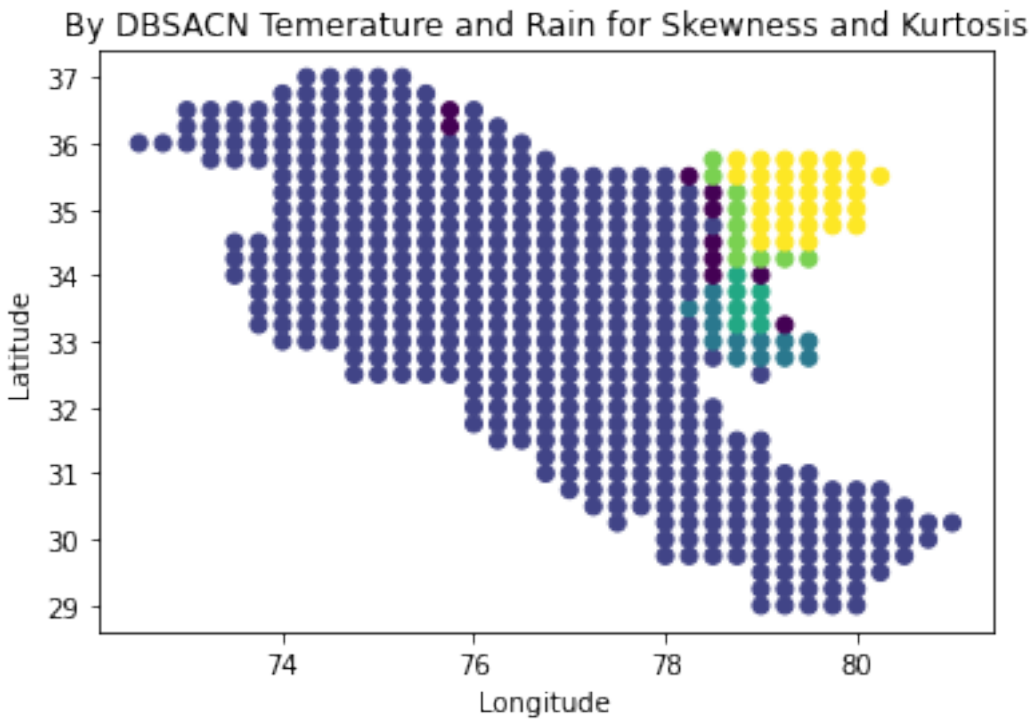
3.0.1 By Using DBSCAN

[ ]:
```
db = DBSCAN(eps=0.3, min_samples=10).fit(TR)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
```

[ ]:
```
labels
```

[ ]:
```
array([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  1,  1,  1,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  1,  1,
        1,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  1,  2,  2, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  1,  2,  2,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  2,
        2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0, -1,  2, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1,  3,  3,  3,  3,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0, -1,  3,  4,  4,  4,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  3,  4,  4,  4,  4,  4,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1,  3,
        4,  4,  4,  4,  4,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0, -1,  3,  4,  4,  4,  4,  4,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1,  3,  4,  4,
        4,  4,  4,  4,  4,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  3,  4,  4,  4,  4,  4,  4,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0])
```

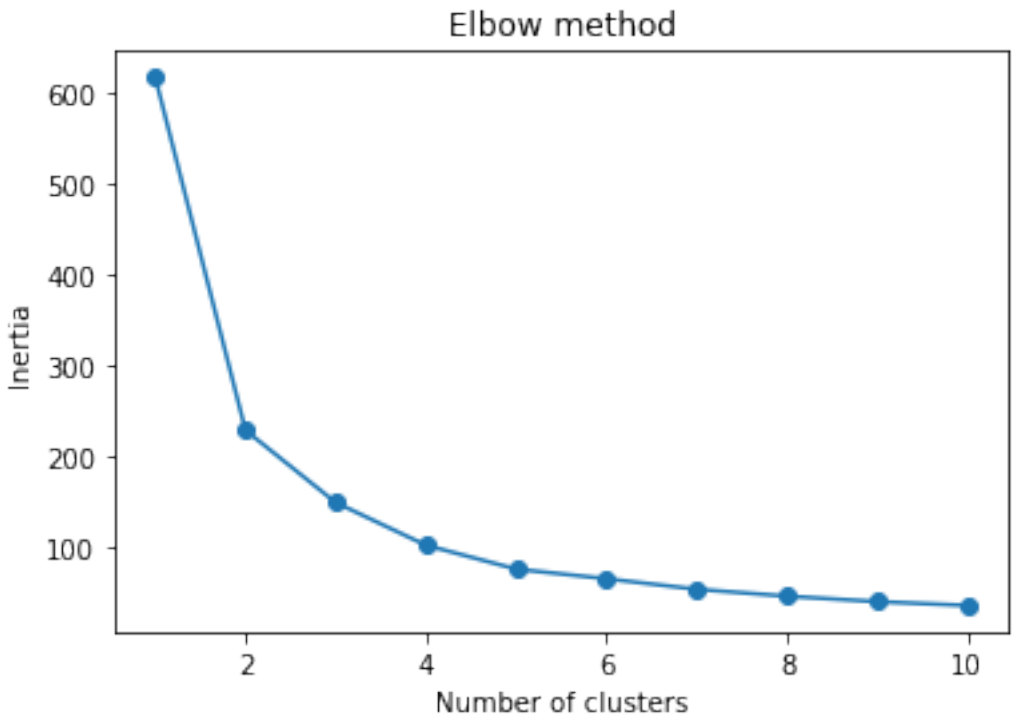
[ ]:
```
plt.scatter(Avg_all['Longitude'],Avg_all['Latitude'],c=labels)
plt.title('By DBSACN Temerature and Rain for Skewness and Kurtosis')
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```



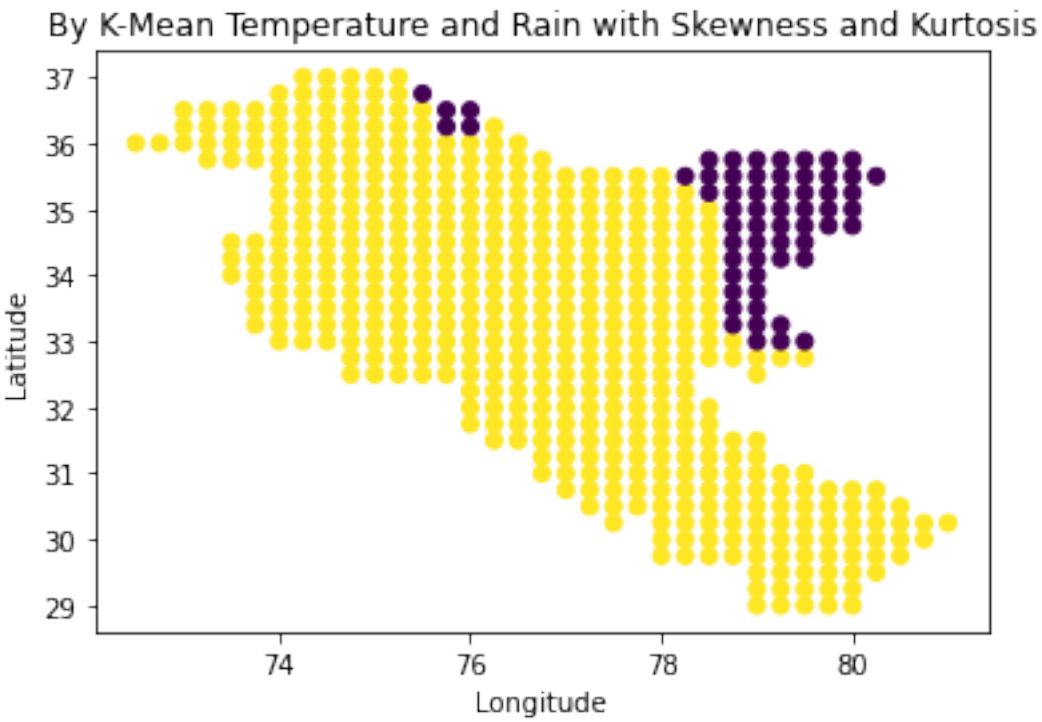
3.0.2 By Using K-Mean

[ ]:

[ ]:
```
data = TR
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



[ ]:
```
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)

plt.scatter(Avg_all['Longitude'],Avg_all['Latitude'],c=kmeans.labels_)
plt.title('By K-Mean Temperature and Rain with Skewness and Kurtosis')
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```



4 By considering both Temperature and Rainfall Correlation

[ ]:
```
RTC=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/SDA/Data/
↪drive-download-20221110T140448Z-001.zip (Unzipped Files)/
↪correlation_rainfall_temperature_average_year_1901_2020.csv')
```

[ ]:
```
RTC
```

[ ]:

	Longitude	Latitude	Correlation_rainfall_temperature
0	79.00	29.0	-0.270496
1	79.25	29.0	-0.277403
2	79.50	29.0	-0.281576
3	79.75	29.0	-0.278581
4	80.00	29.0	-0.299870
..	…	…	…
520	74.25	37.0	0.002959
521	74.50	37.0	-0.019783
522	74.75	37.0	-0.013005
523	75.00	37.0	-0.023426
524	75.25	37.0	-0.022828

[525 rows x 3 columns]

[ ]:
```
RTC['Extra Variable']=list(np.ones(525))
```

[ ]:
```
XC=list(zip(RTC['Correlation_rainfall_temperature'],RTC['Extra Variable']))
```

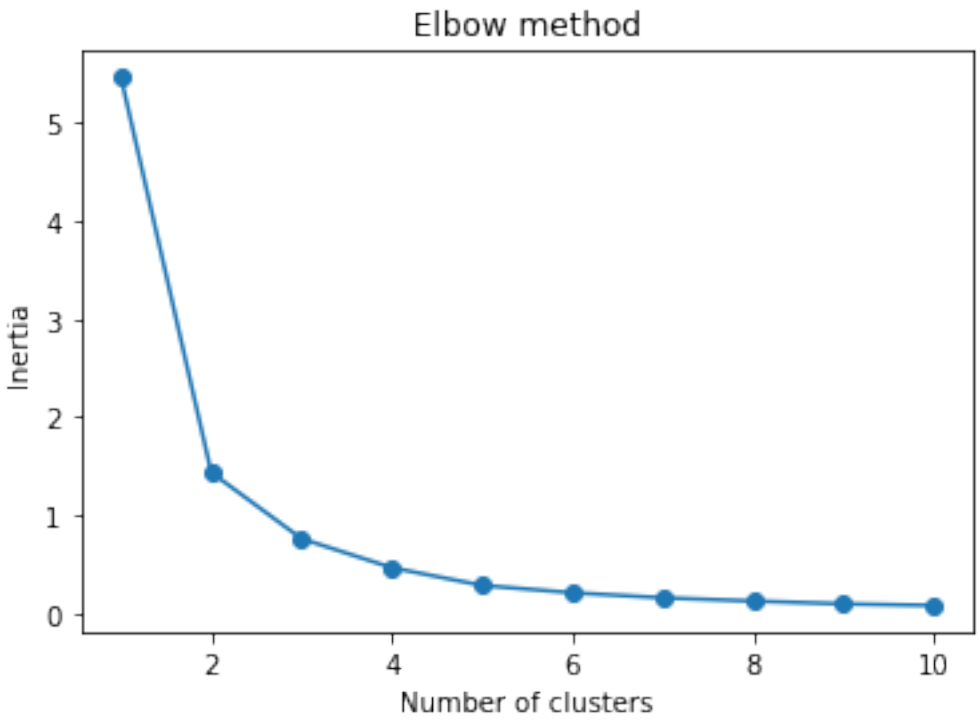
4.0.1 By Using DBSCAN

[ ]:
```
# db_cluster = DBSCAN(eps=5, min_samples=2, algorithm='ball_tree',␣
 ↪metric='minkowski', leaf_size=50, p=4)
# arr = db_cluster.fit_predict(XC)
# print ("Clusters assigned are:", set(db_cluster.labels_))
```

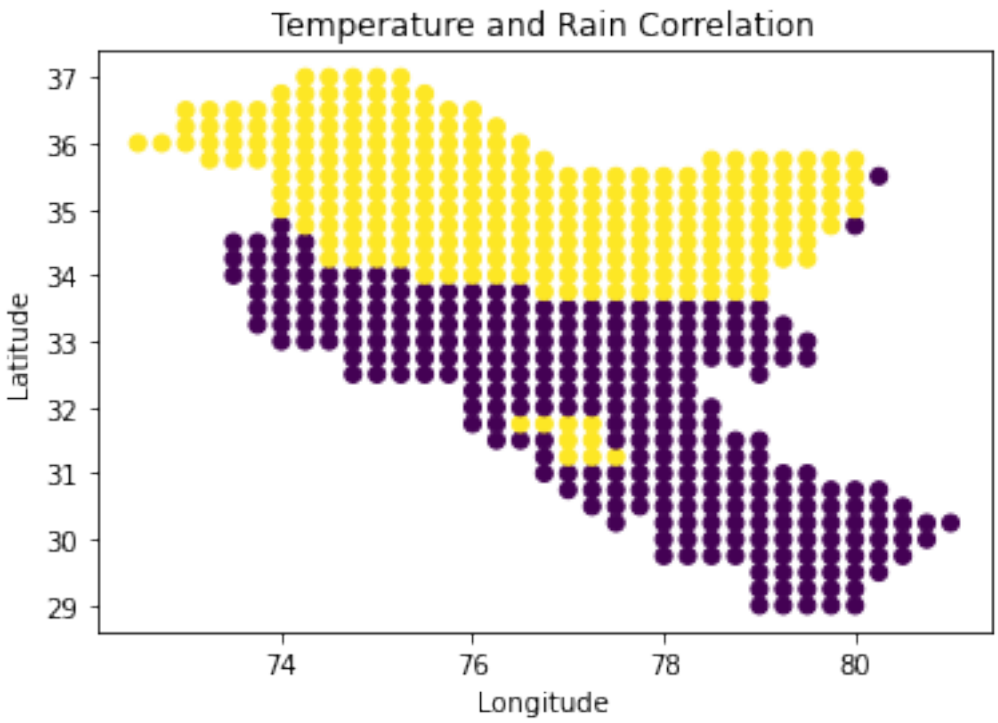
[ ]:
```
# plt.scatter(RTC['Longitude'],RTC['Latitude'],c=labels)
# plt.ylabel("Latitude")
# plt.xlabel("Longitude")
# plt.show()
```

4.0.2 By Using K-Mean

[ ]:
```
data = XC
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



[ ]:
```
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)

plt.scatter(RTC['Longitude'],RTC['Latitude'],c=kmeans.labels_)
plt.title('Temperature and Rain Correlation')
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```



5 By taking mean of Temparature And Rainfall

Note:- Not Giving Expected outcome for DBscan but giving for k-Mean

[ ]:
```
insrt=[]
insrt.append('Longitude')
insrt.append('Latitute')
for i in range(1901,2021):
    for j in range(0,12):
        insrt.append(i)
```

[ ]:
```
Rain=Rain.T
```

[ ]:
```
Rain.insert(0,'Title',insrt)
```

[ ]:
```
Rain
```

[ ]:

	Title	0	1	2	3	4	5	6	\
0	Longitude	79.00	79.25	79.50	79.75	80.00	79.00	79.25	
1	Latitute	29.00	29.00	29.00	29.00	29.00	29.25	29.25	
2	1901	55.75	59.65	63.50	66.55	66.70	65.80	69.70	
3	1901	25.30	26.55	29.95	31.15	33.10	28.95	30.20	
4	1901	19.65	21.85	25.80	26.90	26.70	24.20	26.40	

```
…          …     …      …      …      …      …      …      …
1437       2020  287.90  292.65  292.45  301.00  306.80  282.75  287.50
1438       2020   20.05   19.95   20.40   20.95   21.75   19.20   19.10
1439       2020    0.25    0.15    0.15    0.15    0.30    0.10    0.00
1440       2020    0.55    0.60    0.65    0.60    0.60    0.65    0.70
1441       2020    2.25    2.35    2.70    2.75    3.00    2.40    2.50

            7       8  …    515    516    517    518    519    520    521  \
0       79.50   79.75  …  74.50  74.75  75.00  75.25  75.50  74.25  74.50
1       29.25   29.25  …  36.75  36.75  36.75  36.75  36.75  37.00  37.00
2       73.55   77.40  …  14.15  13.70  13.05  12.40  10.50  15.85  15.40
3       33.60   37.00  …   7.40   7.80   7.20   6.60   5.75   5.10   5.50
4       30.35   34.30  …  11.55  11.40  10.15   8.90   7.55  10.05   9.90
…       …      …  …    …    …      …      …      …      …      …
1437  287.30  287.10  …   8.15   5.70   3.90   2.10   1.05   6.90   4.45
1438   19.55   20.00  …   5.15   2.80   1.40   0.00   0.00   5.05   2.70
1439    0.00    0.00  …  14.50  14.70  13.30  11.90   9.90  16.65  16.85
1440    0.75    0.80  …  20.50  21.50  21.95  22.40  24.70  24.25  25.25
1441    2.85    3.20  …  28.60  31.60  32.00  32.40  28.30  32.55  35.55

         522    523    524
0      74.75  75.00  75.25
1      37.00  37.00  37.00
2      13.50  11.50  10.85
3       5.45   5.40   4.80
4       8.75   8.35   7.10
…      …      …      …
1437    3.75   2.85   1.05
1438    1.85   1.40   0.00
1439   14.95  12.45  11.05
1440   21.75  20.05  20.50
1441   34.50  33.40  33.80

[1442 rows x 526 columns]
```

[ ]:
```
Rain_new=Rain.groupby('Title').mean()
```

[ ]:
```
Rain_new=Rain_new.T
```

[ ]:
```
Rain_new=Rain_new.drop(['Latitute','Longitude'],axis=1)
```

[ ]:
```
Rain_new
```

```
Title        1901        1902       1903        1904       1905        1906  \
0       82.558333   82.079167  63.091667  102.983333  57.308333   94.141667
1       87.404167   86.962500  65.787500  109.537500  60.908333  100.904167
2       90.654167   90.841667  66.837500  115.166667  63.195833  107.112500
```

```
3       96.558333   97.345833  70.600000  123.033333  67.391667  114.737500
4      102.633333  103.354167  75.004167  129.550000  71.637500  120.316667
..            …           …          …           …          …           …
520     10.816667   21.737500  19.654167   21.470833  18.412500   21.179167
521      9.929167   22.212500  18.212500   21.258333  17.554167   20.795833
522      8.625000   20.270833  16.083333   19.783333  15.683333   19.600000
523      7.700000   18.412500  14.041667   18.212500  13.887500   17.983333
524      6.612500   17.229167  12.375000   17.345833  12.650000   16.975000

Title       1907        1908        1909        1910  …        2011  \
0      64.537500   95.687500  112.975000  101.862500  …  101.470833
1      67.866667   99.237500  120.362500  107.541667  …  108.012500
2      72.950000  102.595833  126.812500  109.333333  …  114.045833
3      77.241667  107.383333  136.200000  116.195833  …  122.645833
4      81.970833  110.183333  145.345833  122.800000  …  131.875000
..           …           …           …           …  …           …
520    15.012500   27.537500   20.879167   21.875000  …   21.004167
521    14.408333   26.883333   20.212500   21.208333  …   20.729167
522    13.054167   24.883333   18.641667   19.454167  …   19.695833
523    11.875000   22.695833   17.008333   17.904167  …   18.354167
524    10.995833   21.200000   15.850000   16.725000  …   17.850000

Title        2012        2013       2014        2015        2016        2017  \
0       93.816667   98.658333  65.345833   80.300000   78.429167  101.283333
1       97.633333  104.491667  70.962500   86.629167   83.237500  108.508333
2      100.808333  109.920833  77.750000   93.616667   87.829167  115.808333
3      106.162500  116.229167  83.933333  100.312500   94.641667  124.195833
4      110.650000  120.187500  89.870833  106.525000  101.429167  131.737500
..            …           …          …           …           …           …
520     26.987500   22.095833  19.508333   23.783333   26.187500   30.025000
521     27.670833   21.616667  19.829167   22.862500   26.008333   30.466667
522     25.741667   19.254167  18.245833   20.750000   23.679167   28.816667
523     23.383333   17.204167  16.416667   18.650000   21.345833   27.029167
524     22.337500   15.533333  15.370833   17.145833   19.787500   26.416667

Title        2018        2019       2020
0       93.204167   83.658333  75.070833
1       98.370833   89.879167  79.191667
2      102.675000   96.216667  84.279167
3      109.041667  102.850000  89.200000
4      114.654167  109.370833  92.883333
..            …           …          …
520     19.500000   22.583333  24.416667
521     20.033333   22.329167  24.108333
522     18.775000   20.150000  21.812500
523     17.329167   18.004167  19.912500
524     16.570833   16.629167  18.791667
```

```
[525 rows x 120 columns]
```

[ ]:
```
Rain_mean=Rain_new.T.mean()
```

[ ]:
```
Rain_mean.head()
```

[ ]:
```
0     89.031111
1     94.317326
2     99.205104
3    105.685104
4    111.143889
dtype: float64
```

[ ]:
```
Temp=Temp.T
```

[ ]:
```
Temp
```

[ ]:
```
         0      1      2      3      4      5      6      7      8      9    \
0     79.00  79.25  79.50  79.75  80.00  79.00  79.25  79.50  79.75  80.00
1     29.00  29.00  29.00  29.00  29.00  29.25  29.25  29.25  29.25  29.25
2     12.80  12.40  11.45  11.50  11.50  12.10  11.70  10.75   9.80   9.55
3     15.35  14.90  13.75  13.85  13.85  14.55  14.10  12.95  11.80  11.55
4     22.35  21.80  20.40  20.40  20.35  21.45  20.90  19.50  18.10  17.85
…     …      …      …      …      …      …      …      …      …      …
1437  30.00  29.30  27.75  27.70  27.70  29.20  28.50  26.95  25.40  25.15
1438  29.80  29.20  27.65  27.60  27.55  28.90  28.30  26.75  25.20  25.00
1439  26.80  26.20  24.75  24.70  24.70  25.90  25.30  23.85  22.40  22.15
1440  19.95  19.50  18.35  18.35  18.40  19.15  18.70  17.55  16.40  16.15
1441  16.05  15.70  14.85  14.85  14.90  15.35  15.00  14.15  13.30  13.00

      …    515    516    517    518    519    520    521    522    523  \
0     …  74.50  74.75  75.00  75.25  75.50  74.25  74.50  74.75  75.00
1     …  36.75  36.75  36.75  36.75  36.75  37.00  37.00  37.00  37.00
2     … -14.30 -15.70 -16.55 -17.40 -16.75 -14.95 -16.35 -16.50 -16.15
3     … -13.70 -15.30 -15.90 -16.50 -15.85 -14.05 -15.65 -15.70 -15.45
4     …  -5.20  -6.80  -7.40  -8.00  -6.95  -5.50  -7.10  -7.05  -6.45
…     …    …      …      …      …      …      …      …      …      …
1437  …  10.90   9.40   8.80   8.20   9.30  10.85   9.35   9.40   9.90
1438  …   5.70   4.20   3.75   3.30   4.65   5.75   4.25   4.30   4.80
1439  …  -0.50  -2.10  -2.50  -2.90  -1.85  -0.60  -2.20  -2.20  -1.80
1440  …  -7.25  -8.80  -9.40 -10.00  -9.15  -7.60  -9.15  -9.20  -8.85
1441  … -12.45 -13.90 -14.40 -14.90 -14.15 -12.70 -14.15 -14.20 -14.00

        524
0     75.25
1     37.00
```

```
2    -17.00
3    -16.05
4     -7.05
…      …
1437   9.30
1438   4.35
1439  -2.20
1440  -9.45
1441 -14.50

[1442 rows x 525 columns]
```

[ ]:
```
Temp.insert(0,'Title',insrt)
```

[ ]:
```
Temp_new=Temp.groupby('Title').mean()
```

[ ]:
```
Temp_new=Temp_new.T
```

[ ]:
```
Temp_new=Temp_new.drop(['Latitute','Longitude'],axis=1)
```

[ ]:
```
Temp_mean=Temp_new.T.mean()
```

[ ]:
```
Temp_mean.head()
```

[ ]:
```
0    24.437674
1    23.865347
2    22.485729
3    22.439861
4    22.427083
dtype: float64
```

[ ]:
```
arr=list(zip(Temp_mean,Rain_mean))
arr=np.array(arr)
#Y= StandardScaler().fit_transform(Y)
```

5.0.1 By Using DBSCAN

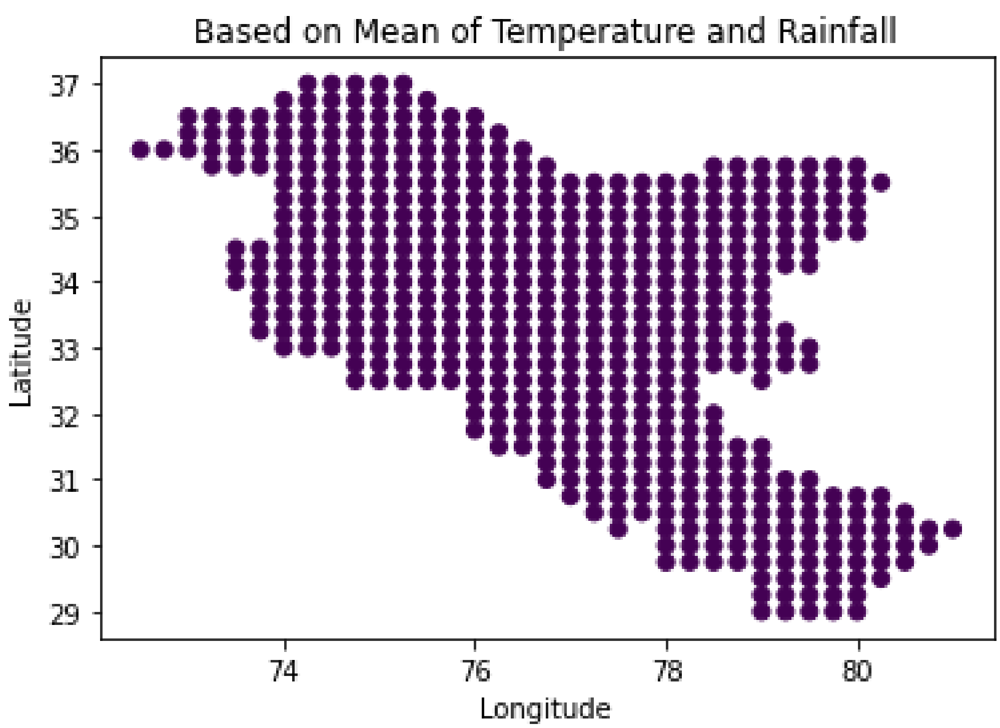
[ ]:
```
db = DBSCAN(eps=20, min_samples=3).fit(arr)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
```

[ ]:
```
labels
```

[ ]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

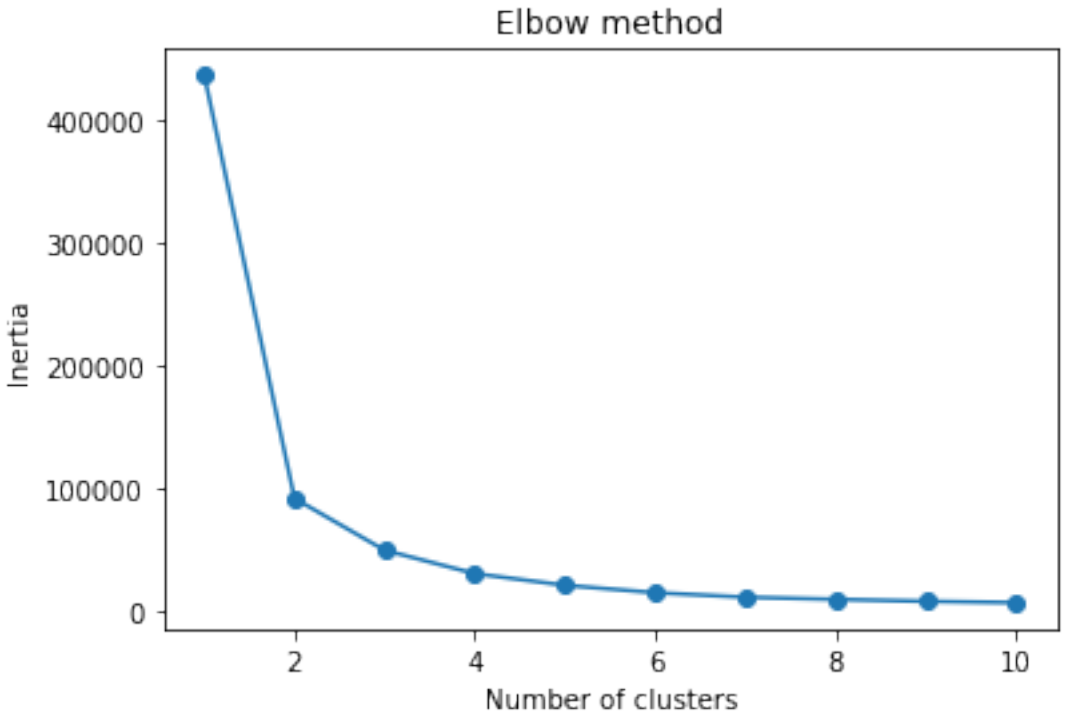
[ ]:
```
plt.scatter(RTC['Longitude'],RTC['Latitude'],c=labels)
plt.title('Based on Mean of Temperature and Rainfall')
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```



[ ]:

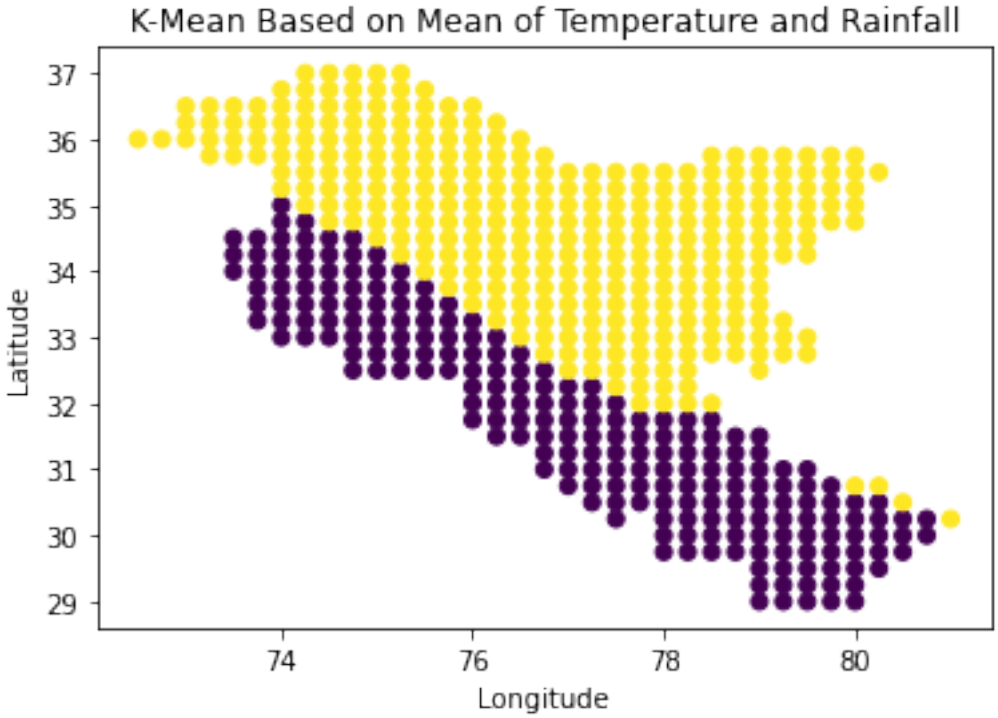
5.0.2 Byusing KMean

[ ]:
```
data = arr
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



[ ]:
```
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)
```

```
plt.scatter(RTC['Longitude'],RTC['Latitude'],c=kmeans.labels_)
plt.title('K-Mean Based on Mean of Temperature and Rainfall')
plt.ylabel("Latitude")
plt.xlabel("Longitude")
plt.show()
```



6 Clustering Based on skewness and kurtosis for data from every month for 1901 to 2020.

[ ]:
```
RTE=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/SDA/Data/
↪drive-download-20221110T140448Z-001.zip (Unzipped Files)/
↪skewness_kurtosis_rainfall_temperature_every_month_1901_2020.csv')
```

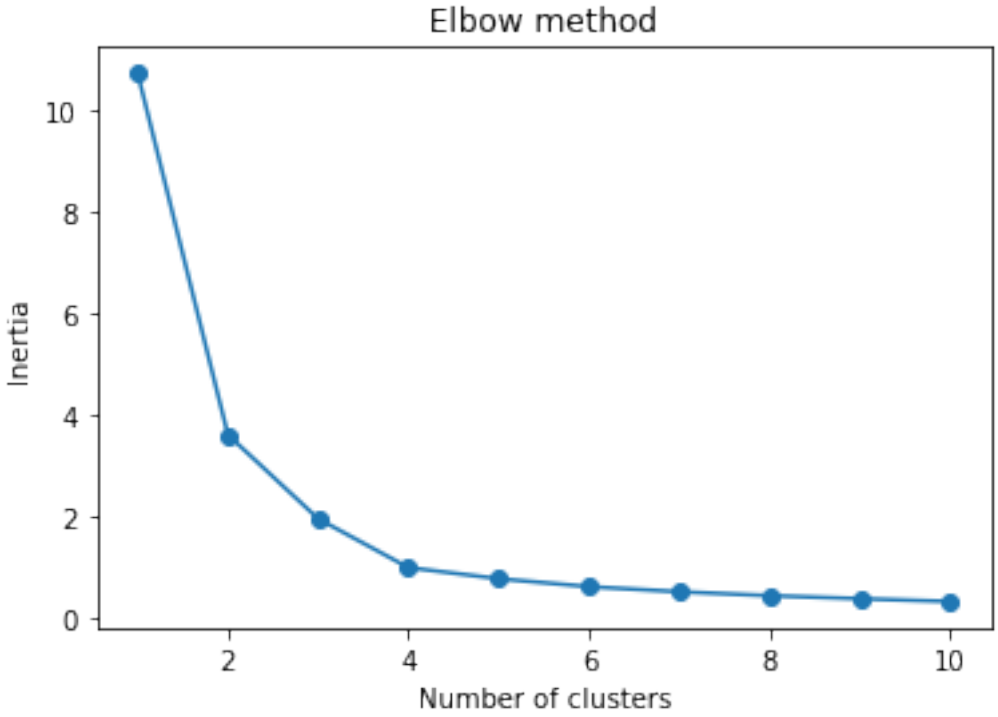
[ ]:
```
RTE.columns
```

[ ]:
```
Index(['Longitude', 'Latitude', 'rain_skewness', 'temperature_skewness',
       'rain_kurtosis', 'temperature_kurtosis'],
      dtype='object')
```

6.0.1 For Temperature

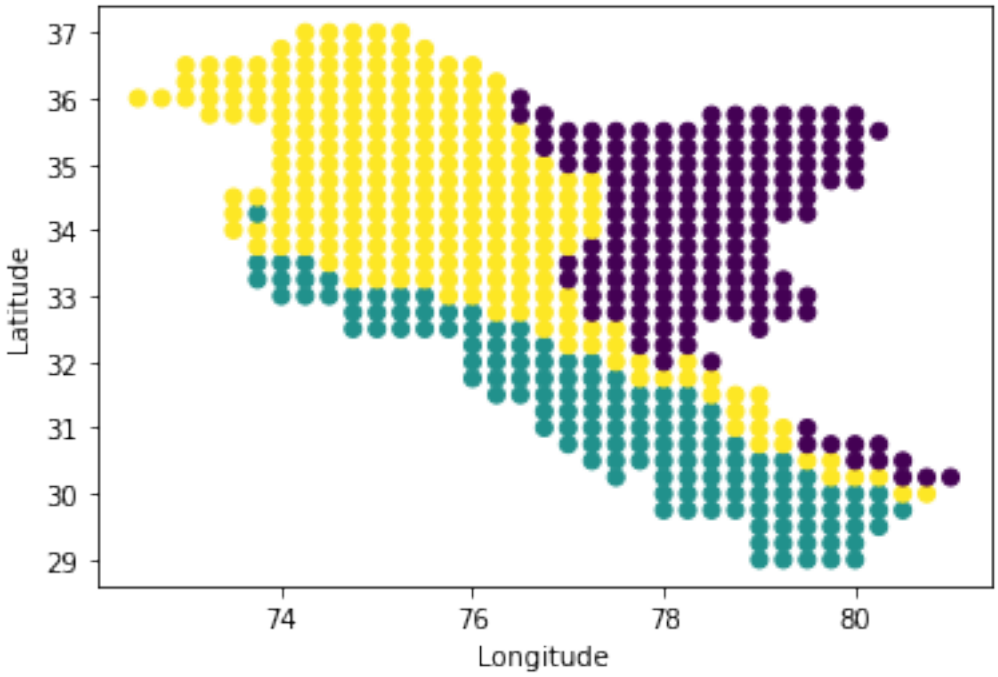
[ ]:
```
Y=list(zip(RTE['temperature_skewness'],RTE['temperature_kurtosis']))
```

[ ]:
```
data = Y
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```



[ ]:
```
kmeans = KMeans(n_clusters=3)
kmeans.fit(data)

plt.scatter(RTE['Longitude'],RTE['Latitude'],c=kmeans.labels_)
plt.ylabel("Latitude")
plt.xlabel("Longitude")
```

```
plt.show()
```




```
[ ]:
```