EXAMPLE – US CITIES

Example – US cities

- The file *cities1.xlsx* has useful data from hundreds of US metropolitan cities
- It is of interest to group them into clusters and to characterize the cities in the clusters
- To characterize the clusters, identify what attributes the cities in each cluster have in common
- Use KMeans from sklearn to find the clusters
- Use a biplot and pivot tables to characterize them

Example – US cities

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

```
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

```
from scipy.spatial import ConvexHull
```

```
df0 = pd.read_csv("cities1.csv")
df0.head()
```

	Metro_Area	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property
0	Abilene, TX	96.32	36.54	17.28	49.29	55.52	49.58	27.20	45.04	2.83	123711	582	4396
1	Akron, OH	47.31	69.68	86.11	71.95	22.66	54.11	81.59	24.07	77.33	689538	518	4527
2	Albany, GA	86.12	28.02	32.01	26.62	75.63	15.59	33.15	20.11	6.79	120838	761	7036
3	Albany-Schenectady-Troy, NY	25.22	82.71	52.97	99.43	8.78	73.94	79.61	77.33	77.62	885782	365	3531
4	Albuquerque, NM	44.48	84.13	90.65	71.67	78.18	2.84	75.36	77.90	70.25	734255	1133	7261

Example – US cities

description

	A	J	K	L	M	N	O	P
1	Metropolitan_Area	Recreation	Population_2000	Total_Violent	Total_Property	Crime_Trend	Unemployment_Threat	Past_Job_Growth
2	Abilene, TX	2.83	123711	582	4396	Constant	Average	6.1
3	Akron, OH	77.33	689538	518	4527	Constant	Average	11.6
4	Albany, GA	6.79	120838	761	7036	Constant	Average	6.6
5	Albany-Schenectady-Troy, NY	77.62	885782	365	3531	Constant	Below average	5.2
6	Albuquerque, NM	70.25	734255	1133	7261	Constant	Below average	21.4
7	Alexandria, LA	22.66	127635	1100	5581	Constant	Below average	6.1
8	Allentown-Bethlehem-Easton, PA	47.87	616924	226	3153	Constant	Average	4.7
9	Altoona, PA	0	131388	162	2240	Constant	Average	9.2
10	Amarillo, TX	30.59	212395	588	6646	Constant	Average	16.6
11	Anchorage, AK	6.23	263727	727	5808	Decreasing	Below average	14.5
12	Ann Arbor, MI	83.28	554998	356	4078	Decreasing	Average	11.6
13	Anniston, AL	13.88	118556	882	4607	Constant	Average	6.6
14	Appleton-Oshkosh-Neenah, WI	86.4	352708	84	3247	Decreasing	Above average	18.1
15	Asheville, NC	38.81	218552	372	4063	Constant	Average	14.6
16	Athens, GA	32.01	144889	551	6207	Constant	Average	10.3
17	Atlanta, GA	76.2	3807451	834	6787	Constant	Below average	16.9

	Metro_Area	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property
0	Abilene, TX	96.32	36.54	17.28	49.29	55.52	49.58	27.20	45.04	2.83	123711	582	4396
1	Akron, OH	47.31	69.68	86.11	71.95	22.66	54.11	81.59	24.07	77.33	689538	518	4527
2	Albany, GA	86.12	28.02	32.01	26.62	75.63	15.59	33.15	20.11	6.79	120838	761	7036
3	Albany-Schenectady-Troy, NY	25.22	82.71	52.97	99.43	8.78	73.94	79.61	77.33	77.62	885782	365	3531
4	Albuquerque, NM	44.48	84.13	90.65	71.67	78.18	2.84	75.36	77.90	70.25	734255	1133	7261

Example – US cities

	A	J	K	L	M	N	O	P
1	Metropolitan_Area	Recreation	Population_2000	Total_Violent	Total_Property	Crime_Trend	Unemployment_Threat	Past_Job_Growth
2	Abilene, TX	2.83	123711	582	4396	Constant	Average	6.1
3	Akron, OH	77.33	689538	518	4527	Constant	Average	11.6
4	Albany, GA	6.79	120838	761	7036	Constant	Average	6.6
5	Albany-Schenectady-Troy, NY	77.62	885782	365	3531	Constant	Below average	5.2
6	Albuquerque, NM	70.25	734255	1133	7261	Constant	Below average	21.4
7	Alexandria, LA	22.66	127635	1100	5581	Constant	Below average	6.1
8	Allentown-Bethlehem-Easton, PA	47.87	616924	226	3153	Constant	Average	4.7
9	Altoona, PA	0	131388	162	2240	Constant	Average	9.2
10	Amarillo, TX	30.59	212395	588	6646	Constant	Average	16.6
11	Anchorage, AK	6.23	263727	727	5808	Decreasing	Below average	14.5
12	Ann Arbor, MI	83.28	554998	356	4078	Decreasing	Average	11.6
13	Anniston, AL	13.88	118556	882	4607	Constant	Average	6.6
14	Appleton-Oshkosh-Neenah, WI	86.4	352708	84	3247	Decreasing	Above average	18.1
15	Asheville, NC	38.81	218552	372	4063	Constant	Average	14.6
16	Athens, GA	32.01	144889	551	6207	Constant	Average	10.3
17	Atlanta, GA	76.2	3807451	834	6787	Constant	Below average	16.9

	Metro_Area	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property
0	Abilene, TX	96.32	36.54	17.28	49.29	55.52	49.58	27.20	45.04	2.83	123711	582	4396
1	Akron, OH	47.31	69.68	86.11	71.95	22.66	54.11	81.59	24.07	77.33	689538	518	4527
2	Albany, GA	86.12	28.02	32.01	26.62	75.63	15.59	33.15	20.11	6.79	120838	761	7036
3	Albany-Schenectady-Troy, NY	25.22	82.71	52.97	99.43	8.78	73.94	79.61	77.33	77.62	885782	365	3531
4	Albuquerque, NM	44.48	84.13	90.65	71.67	78.18	2.84	75.36	77.90	70.25	734255	1133	7261

Example – US cities

	A	J	K	L	M	N	O	P
1	Metropolitan_Area	Recreation	Population_2000	Total_Violent	Total_Property	Crime_Trend	Unemployment_Threat	Past_Job_Growth
2	Abilene, TX	2.83	123711	582	4396	Constant	Average	6.1
3	Akron, OH	77.33	689538	518	4527	Constant	Average	11.6
4	Albany, GA	6.79	120838	761	7036	Constant	Average	6.6
5	Albany-Schenectady-Troy, NY	77.62	885782	365	3531	Constant	Below average	5.2
6	Albuquerque, NM	70.25	734255	1133	7261	Constant	Below average	21.4
7	Alexandria, LA	22.66	127635	1100	5581	Constant	Below average	6.1
8	Allentown-Bethlehem-Easton, PA	47.87	616924	226	3153	Constant	Average	4.7
9	Altoona, PA	0	131388	162	2240	Constant	Average	9.2
10	Amarillo, TX	30.59	212395	588	6646	Constant	Average	16.6
11	Anchorage, AK	6.23	263727	727	5808	Decreasing	Below average	14.5
12	Ann Arbor, MI	83.28	554998	356	4078	Decreasing	Average	11.6
13	Anniston, AL	13.88	118556	882	4607	Constant	Average	6.6
14	Appleton-Oshkosh-Neenah, WI	86.4	352708	84	3247	Decreasing	Above average	18.1
15	Asheville, NC	38.81	218552	372	4063	Constant	Average	14.6
16	Athens, GA	32.01	144889	551	6207	Constant	Average	10.3
17	Atlanta, GA	76.2	3807451	834	6787	Constant	Below average	16.9

```
df0 = df0.drop(columns=['Crime_Trend', 'Unemployment_Threat'])
df0.head()
```

	Metro_Area	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property	Past_JobG
0	Abilene, TX	96.32	36.54	17.28	49.29	55.52	49.58	27.20	45.04	2.83	123711	582	4396	6.1
1	Akron, OH	47.31	69.68	86.11	71.95	22.66	54.11	81.59	24.07	77.33	689538	518	4527	11.6
2	Albany, GA	86.12	28.02	32.01	26.62	75.63	15.59	33.15	20.11	6.79	120838	761	7036	6.6
3	Albany-Schenectady-Troy, NY	25.22	82.71	52.97	99.43	8.78	73.94	79.61	77.33	77.62	885782	365	3531	5.2
4	Albuquerque, NM	44.48	84.13	90.65	71.67	78.18	2.84	75.36	77.90	70.25	734255	1133	7261	21.4

Example – US cities

.

```
df0 = df0.drop(columns=['Crime_Trend', 'Unemployment_Threat'])
df0.head()
```

	Metro_Area	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property	Past_JobG
0	Abilene, TX	96.32	36.54	17.28	49.29	55.52	49.58	27.20	45.04	2.83	123711	582	4396	6.1
1	Akron, OH	47.31	69.68	86.11	71.95	22.66	54.11	81.59	24.07	77.33	689538	518	4527	11.6
2	Albany, GA	86.12	28.02	32.01	26.62	75.63	15.59	33.15	20.11	6.79	120838	761	7036	6.6
3	Albany-Schenectady-Troy, NY	25.22	82.71	52.97	99.43	8.78	73.94	79.61	77.33	77.62	885782	365	3531	5.2
4	Albuquerque, NM	44.48	84.13	90.65	71.67	78.18	2.84	75.36	77.90	70.25	734255	1133	7261	21.4

```
df = df0.copy()
df.set_index('Metro_Area',inplace=True)
df.head()
```

← set index column

	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property	Past_JobG
Metro_Area													
Abilene, TX	96.32	36.54	17.28	49.29	55.52	49.58	27.20	45.04	2.83	123711	582	4396	6.1
Akron, OH	47.31	69.68	86.11	71.95	22.66	54.11	81.59	24.07	77.33	689538	518	4527	11.6
Albany, GA	86.12	28.02	32.01	26.62	75.63	15.59	33.15	20.11	6.79	120838	761	7036	6.6
Albany-Schenectady-Troy, NY	25.22	82.71	52.97	99.43	8.78	73.94	79.61	77.33	77.62	885782	365	3531	5.2
Albuquerque, NM	44.48	84.13	90.65	71.67	78.18	2.84	75.36	77.90	70.25	734255	1133	7261	21.4

Example – US cities

.

```
df.shape
```

```
(325, 13)
```

```
m = 325  cities
n = 13   variables
```

Scale all numeric columns

```
df_scaled = (df - df.min()) / (df.max() - df.min())
df_scaled.head().round(3)
```

← MinMaxScaler

Metro_Area	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property	Past_JobG
Abilene, TX	0.963	0.365	0.173	0.493	0.550	0.496	0.272	0.454	0.028	0.009	0.272	0.320	0.290
Akron, OH	0.473	0.697	0.861	0.720	0.218	0.541	0.816	0.243	0.773	0.071	0.239	0.335	0.389
Albany, GA	0.861	0.280	0.320	0.266	0.754	0.156	0.331	0.203	0.068	0.009	0.367	0.613	0.299
Albany-Schenectady-Troy, NY	0.252	0.827	0.530	0.994	0.077	0.739	0.796	0.780	0.776	0.092	0.158	0.225	0.274
Albuquerque, NM	0.445	0.841	0.907	0.717	0.779	0.028	0.754	0.786	0.702	0.075	0.564	0.638	0.566

Example – US cities

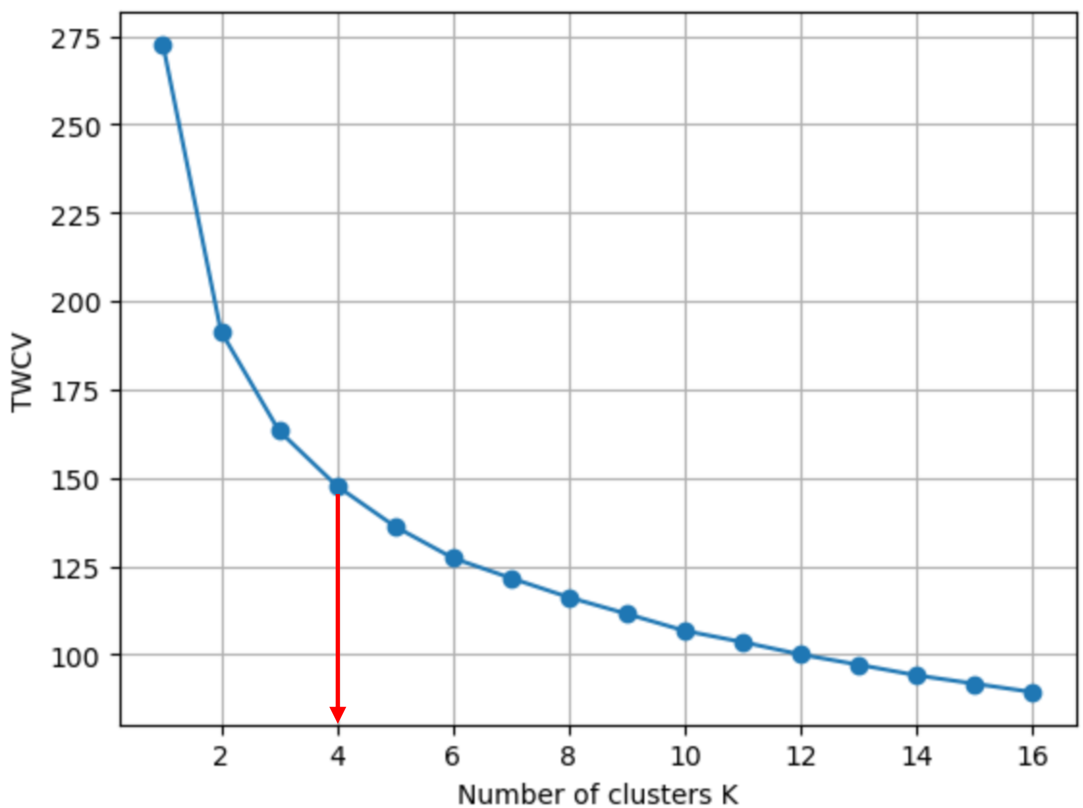
.

Find best number of clusters (try up to 17 clusters)

```
np.random.seed(123)
twcv_values = []
for k in range( 1, 17):
    kmeans1 = KMeans(n_clusters=k,
                     n_init=25,
                     random_state=123)
    kmeans1.fit(df_scaled)
    twcv_values.append(kmeans1.inertia_)
```

Elbow chart

```
plt.plot(np.arange(1 , 17) ,
         twcv_values, 'o-')
plt.ylabel('TWCV')
plt.xlabel('Number of clusters K')
```



Looks like the curve starts decreasing slowly at k=4 clusters

Example – US cities

K=4 Clusters

```
np.random.seed(123)
k4 = KMeans(n_clusters=4, n_init=25,random_state=123).fit(df_scaled)
```

PCA

```
pca = PCA()
pca_x = pca.fit_transform(df_scaled)
```

```
df_pca0 = pd.DataFrame(pca_x)
df_pca0.round(4)
```

← store transformed values in a DataFrame

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	-0.5651	0.0781	-0.3617	0.0769	-0.2303	-0.0327	-0.0632	0.2951	0.0608	0.0680	0.0217	-0.0271	-0.0038
1	0.4317	-0.2196	-0.0231	-0.1703	0.4887	-0.1259	0.0337	-0.0156	0.0800	0.0552	0.0991	-0.0697	0.0114
2	-0.5556	0.5714	-0.1304	0.1934	-0.0307	-0.1151	-0.0343	0.1905	0.1641	-0.0568	0.1143	0.0544	-0.0149
3	0.6355	-0.7293	-0.1477	0.0563	0.0491	-0.0263	-0.0264	-0.0027	-0.1280	-0.0043	0.0022	-0.0561	0.0026
4	0.8183	0.5237	-0.0791	0.0171	-0.0780	-0.1384	0.0220	-0.0597	-0.0009	-0.1062	-0.0889	-0.0815	0.0630

325 rows × 13 columns

Example – US cities

K=4 Clusters

```
np.random.seed(123)
k4 = KMeans(n_clusters=4, n_init=25,random_state=123).fit(df_scaled)
```

PCA

```
pca = PCA()
pca_x = pca.fit_transform(df_scaled)
```

```
df_pca0 = pd.DataFrame(pca_x)
df_pca0.columns = range(1,14)
df_pca0.round(4)
```

← rename columns

	1	2	3	4	5	6	7	8	9	10	11	12	13
0	-0.5651	0.0781	-0.3617	0.0769	-0.2303	-0.0327	-0.0632	0.2951	0.0608	0.0680	0.0217	-0.0271	-0.0038
1	0.4317	-0.2196	-0.0231	-0.1703	0.4887	-0.1259	0.0337	-0.0156	0.0800	0.0552	0.0991	-0.0697	0.0114
2	-0.5556	0.5714	-0.1304	0.1934	-0.0307	-0.1151	-0.0343	0.1905	0.1641	-0.0568	0.1143	0.0544	-0.0149
3	0.6355	-0.7293	-0.1477	0.0563	0.0491	-0.0263	-0.0264	-0.0027	-0.1280	-0.0043	0.0022	-0.0561	0.0026
4	0.8183	0.5237	-0.0791	0.0171	-0.0780	-0.1384	0.0220	-0.0597	-0.0009	-0.1062	-0.0889	-0.0815	0.0630

325 rows × 13 columns

Example – US cities

K=4 Clusters

```
np.random.seed(123)
k4 = KMeans(n_clusters=4, n_init=25,random_state=123).fit(df_scaled)
```

PCA

```
pca = PCA()
pca_x = pca.fit_transform(df_scaled)
```

```
df_pca0 = pd.DataFrame(pca_x)
df_pca0.columns = range(1,14)
df_pca0 = df_pca0.add_prefix('PC')          ← rename columns
df_pca0.round(4)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
0	-0.5651	0.0781	-0.3617	0.0769	-0.2303	-0.0327	-0.0632	0.2951	0.0608	0.0680	0.0217	-0.0271	-0.0038
1	0.4317	-0.2196	-0.0231	-0.1703	0.4887	-0.1259	0.0337	-0.0156	0.0800	0.0552	0.0991	-0.0697	0.0114
2	-0.5556	0.5714	-0.1304	0.1934	-0.0307	-0.1151	-0.0343	0.1905	0.1641	-0.0568	0.1143	0.0544	-0.0149
3	0.6355	-0.7293	-0.1477	0.0563	0.0491	-0.0263	-0.0264	-0.0027	-0.1280	-0.0043	0.0022	-0.0561	0.0026
4	0.8183	0.5237	-0.0791	0.0171	-0.0780	-0.1384	0.0220	-0.0597	-0.0009	-0.1062	-0.0889	-0.0815	0.0630

325 rows × 13 columns

Example – US cities

```
df_pca = df_pca0.iloc[:,:2]
df_pca.round(4)
```

	PC1	PC2
0	-0.5651	0.0781
1	0.4317	-0.2196
2	-0.5556	0.5714
3	0.6355	-0.7293
4	0.8183	0.5237
...	...	...
320	-0.3216	0.4353
321	-0.8279	-0.3056
322	-0.0945	-0.0703
323	-0.7848	0.5576
324	-0.6590	0.6347

325 rows × 2 columns

Add column **cluster** to assign cluster ID to each city

↓

```
df_pca4 = df_pca.copy()
df_pca4['cluster'] = k4.labels_
df_pca4
```

	PC1	PC2	cluster
0	-0.565077	0.078104	3
1	0.431726	-0.219634	0
2	-0.555608	0.571353	3
3	0.635549	-0.729267	0
4	0.818256	0.523719	2
...	...	...	...
320	-0.321592	0.435332	3
321	-0.827879	-0.305564	1
322	-0.094524	-0.070281	0
323	-0.784795	0.557625	3
324	-0.658983	0.634707	3

325 rows × 3 columns

Example – split US cities into clusters

```
df_pca4.cluster.value_counts()
```

```
3      98
0      80
1      74
2      73
```

cities
cities
cities
cities

Split df_pca4 into clusters

```
df0 = df_pca4.loc[k4.labels_ == 0]
df1 = df_pca4.loc[k4.labels_ == 1]
df2 = df_pca4.loc[k4.labels_ == 2]
df3 = df_pca4.loc[k4.labels_ == 3]
```

cluster 1

```
df0
```

	PC1	PC2	cluster
1	0.431726	-0.219634	0
3	0.635549	-0.729267	0
6	0.058806	-0.548673	0
10	0.472166	-0.497614	0
12	0.184926	-0.578597	0
...	...	...	...
302	0.030043	0.044177	0
313	0.320184	-0.001436	0
317	0.112015	-0.125230	0
318	0.228308	-0.550483	0
322	-0.094524	-0.070281	0

80 rows × 3 columns

cluster 2

```
df1
```

	PC1	PC2	cluster
7	-0.862086	-0.317707	1
28	-0.507079	-0.208128	1
30	-0.261518	-0.654082	1
32	-0.349208	-0.506870	1
33	-0.335003	-0.476352	1
...	...	...	...
309	-0.388266	-0.398723	1
310	-0.577212	-0.524846	1
312	-0.440582	-0.699956	1
315	-0.767706	-0.406099	1
321	-0.827879	-0.305564	1

74 rows × 3 columns

cluster 3

```
df2
```

	PC1	PC2	cluster
4	0.818256	0.523719	2
15	0.998636	0.326776	2
18	0.749694	0.154682	2
20	1.127669	0.213710	2
31	0.844966	0.216857	2
...	...	...	...
290	0.874282	0.654882	2
296	0.652035	0.565162	2
297	0.340690	0.132563	2
307	1.245694	-0.197515	2
311	0.850432	0.602948	2

73 rows × 3 columns

cluster 4

```
df3
```

	PC1	PC2	cluster
0	-0.565077	0.078104	3
2	-0.555608	0.571353	3
5	-0.363770	0.532921	3
8	-0.157821	0.185864	3
9	-0.068184	0.394189	3
...	...	...	...
316	0.199569	0.375270	3
319	-0.502952	0.542135	3
320	-0.321592	0.435332	3
323	-0.784795	0.557625	3
324	-0.658983	0.634707	3

98 rows × 3 columns

Example – US cities

```
x0 = df0.PC1
y0 = df0.PC2
```
cluster 1

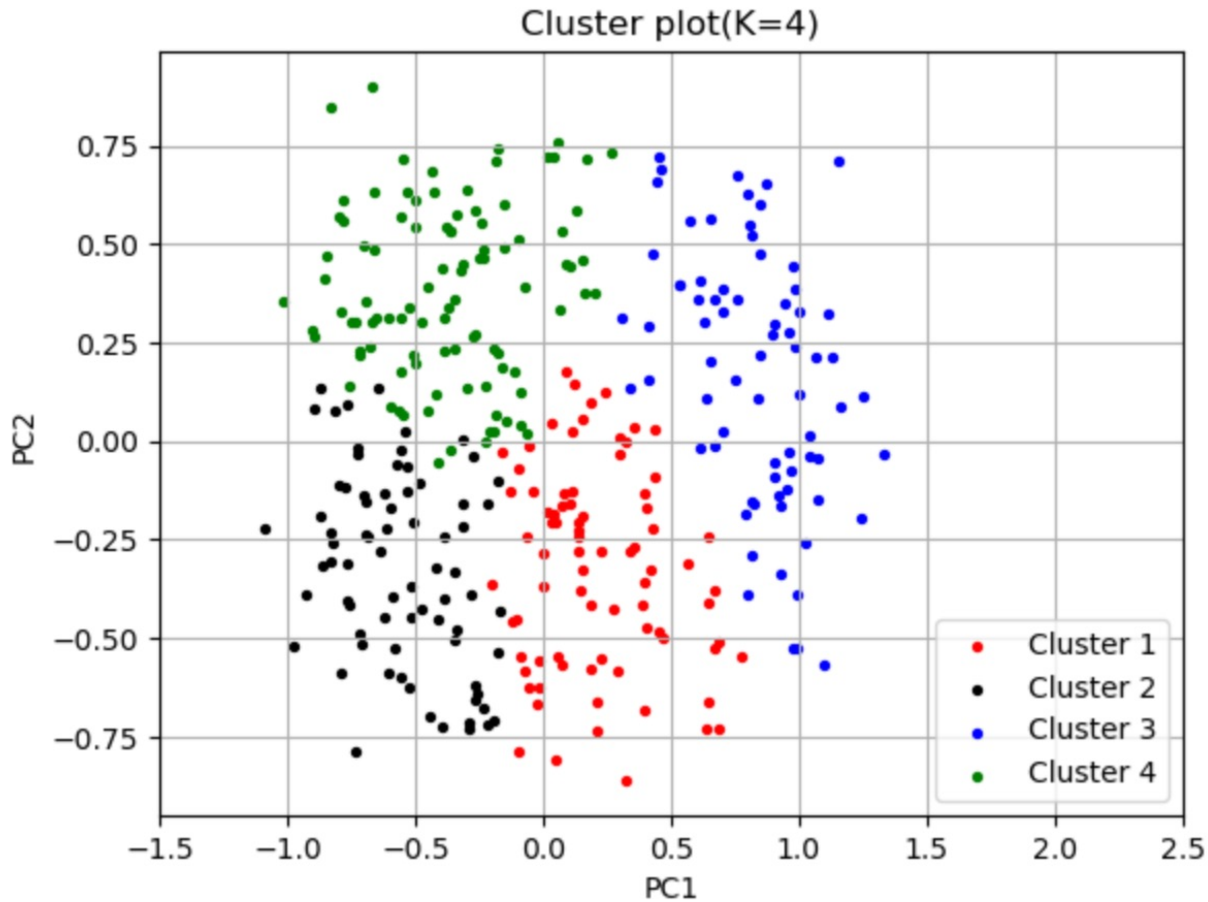
```
x1 = df1.PC1
y1 = df1.PC2
```
cluster 2

```
x2 = df2.PC1
y2 = df2.PC2
```
cluster 3

```
x3 = df3.PC1
y3 = df3.PC2
```
cluster 4

```
plt.scatter(x0, y0,s=8, c = 'r',
            label='Cluster 1')
plt.scatter(x1, y1,s=8, c = 'k',
            label='Cluster 2')
plt.scatter(x2, y2,s=8, c = 'b',
            label='Cluster 3')
plt.scatter(x3, y3,s=8, c = 'g',
            label='Cluster 4')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.xlim(-1.5,2.5)
plt.legend(loc=4)
plt.title('Cluster plot(K=4)')
```

How much do clusters overlap?

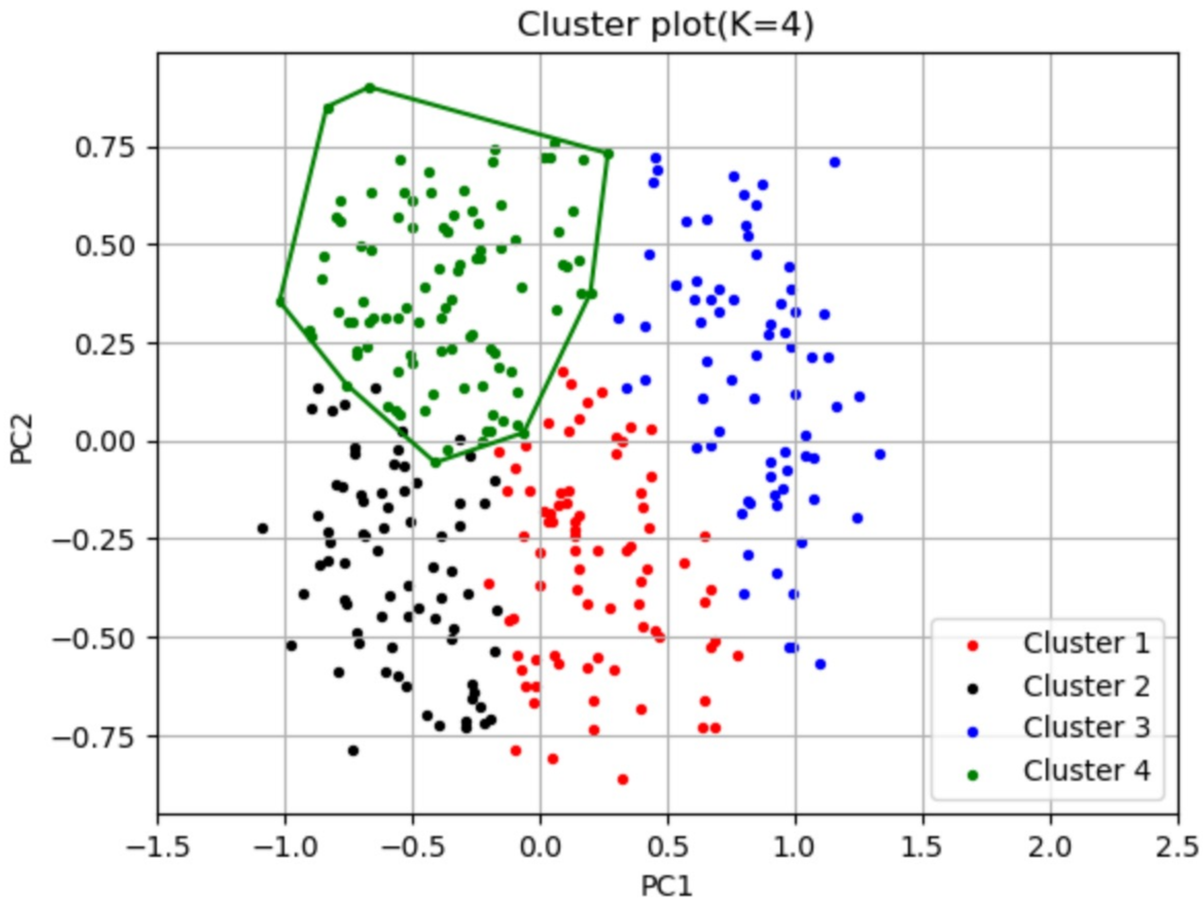


Example – US cities

```
x0 = df0.PC1
y0 = df0.PC2
x1 = df1.PC1
y1 = df1.PC2
x2 = df2.PC1
y2 = df2.PC2
x3 = df3.PC1
y3 = df3.PC2
```

```
plt.scatter(x0, y0,s=8, c = 'r',
            label='Cluster 1')
plt.scatter(x1, y1,s=8, c = 'k',
            label='Cluster 2')
plt.scatter(x2, y2,s=8, c = 'b',
            label='Cluster 3')
plt.scatter(x3, y3,s=8, c = 'g',
            label='Cluster 4')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.xlim(-1.5,2.5)
plt.legend(loc=4)
plt.title('Cluster plot(K=4)')
```

```
points = df3.iloc[:,:2].values
hull = ConvexHull(points)
for i in hull.simplices:
    plt.plot(points[i,0],
             points[i,1], 'g-')
```

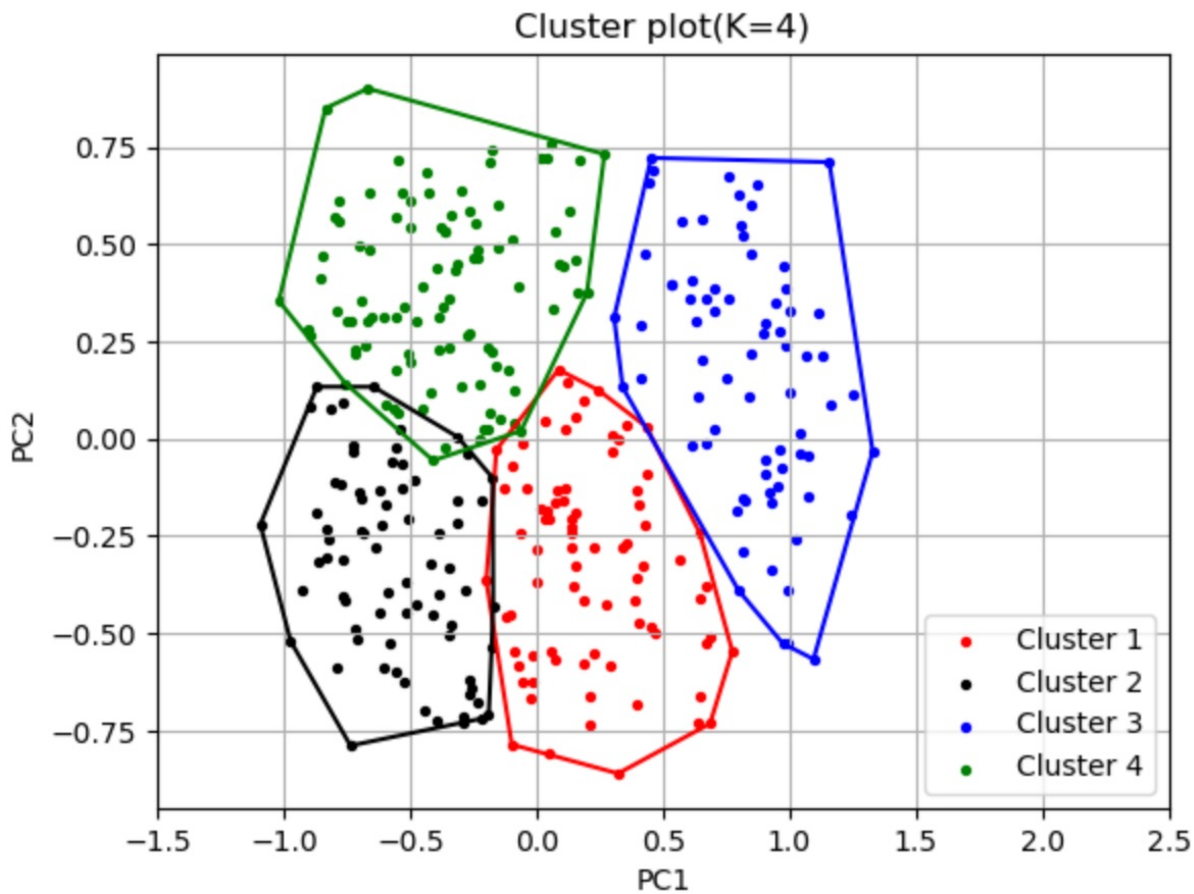


Example – US cities

```
x0 = df0.PC1
y0 = df0.PC2
x1 = df1.PC1
y1 = df1.PC2
x2 = df2.PC1
y2 = df2.PC2
x3 = df3.PC1
y3 = df3.PC2
```

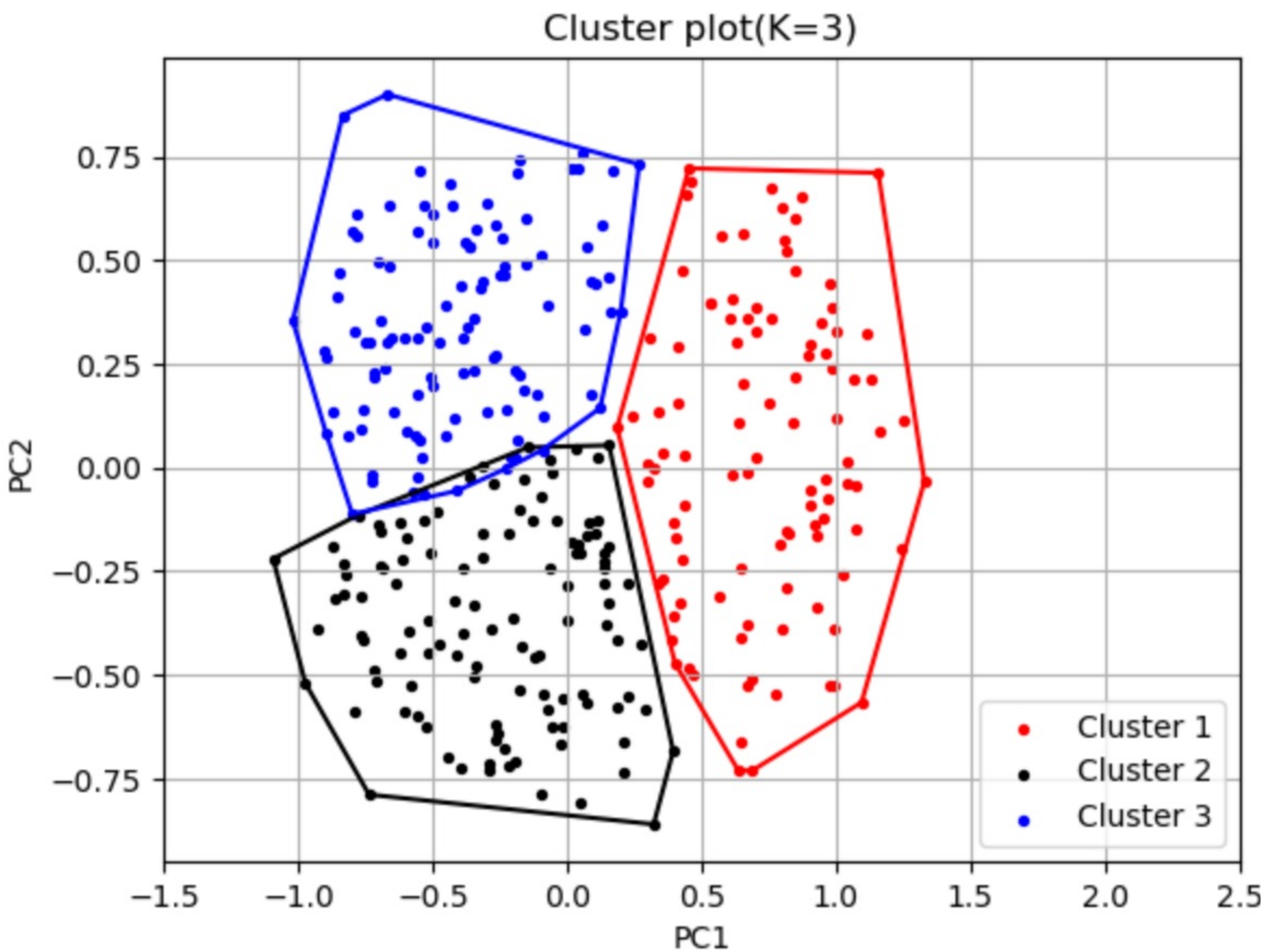
```
plt.scatter(x0, y0,s=8, c = 'r',
            label='Cluster 1')
plt.scatter(x1, y1,s=8, c = 'k',
            label='Cluster 2')
plt.scatter(x2, y2,s=8, c = 'b',
            label='Cluster 3')
plt.scatter(x3, y3,s=8, c = 'g',
            label='Cluster 4')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.xlim(-1.5,2.5)
plt.legend(loc=4)
plt.title('Cluster plot(K=4)')
```

```
points = df3.iloc[:,:2].values
hull = ConvexHull(points)
for i in hull.simplices:
    plt.plot(points[i,0],
             points[i,1], 'g-')
```



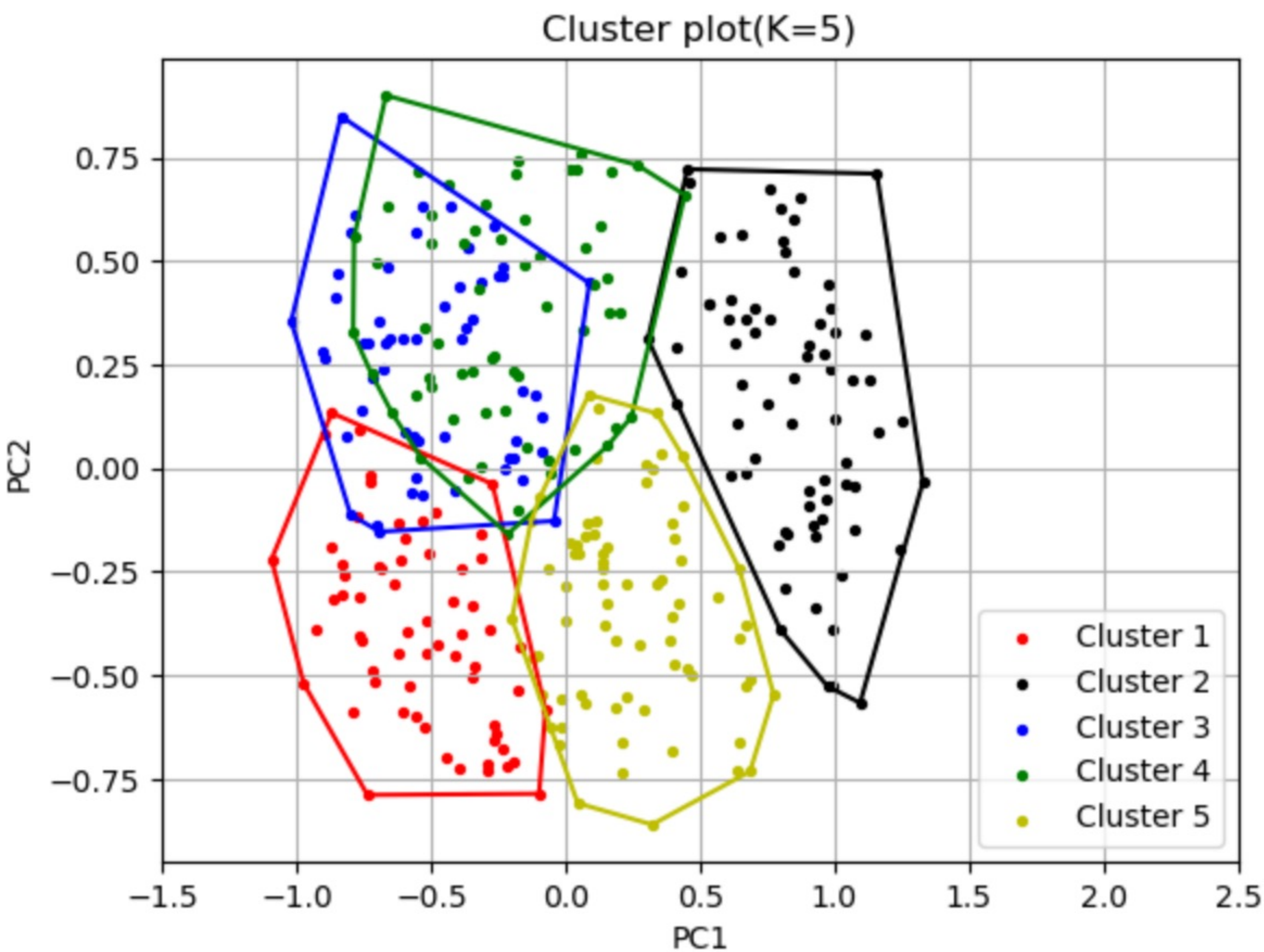
US cities – 3 clusters

.



US cities – 5 clusters

.



US cities – Rotation matrix R

.

```
vectors = pca.components_
R = vectors.T
R.shape
```

```
(13, 13)
```

```
df_R = pd.DataFrame(R)
df_R.index = df.columns
df_R.columns = df_pca0.columns
df_R.round(4)
```

↓ eigen vector1 ↓ eigen vector2 ↓ eigen vector3 …

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
Life_cost	-0.1997	0.2410	-0.7022	-0.3358	0.0426	0.3101	0.0732	0.2708	0.2989	0.1361	0.1038	0.0298	0.0042
Transport	0.4099	-0.0036	-0.2260	-0.1016	0.0847	-0.0297	-0.8504	0.0301	-0.0967	-0.0892	-0.0520	-0.1353	0.0106
Jobs	0.3516	0.2521	0.1098	-0.5690	0.1202	-0.3358	0.1887	-0.2368	-0.1190	0.3146	0.3757	-0.0515	0.0331
Educ	0.3932	-0.2427	-0.1807	0.0557	0.0179	-0.2367	0.3112	0.6963	-0.3251	-0.0410	-0.0712	0.0071	0.0087
Climate	0.0723	0.4924	0.4988	-0.1898	-0.4202	0.2062	-0.1098	0.4559	0.1312	-0.0893	-0.0557	-0.0150	-0.0017
Crime	-0.1677	-0.5888	0.1688	-0.4701	-0.1317	0.1109	-0.1087	0.0801	0.0540	0.0040	0.0253	0.1647	0.5450
Arts	0.4124	-0.1945	0.0630	0.1233	0.0560	-0.1442	0.0978	0.0107	0.8519	-0.0811	0.0529	-0.0643	-0.0025
H_Care	0.3312	-0.1038	-0.2893	0.0388	-0.7875	0.1559	0.1462	-0.3423	-0.0999	0.0006	-0.0229	0.0157	-0.0235
Rec	0.4083	-0.0392	0.1362	-0.0777	0.3864	0.7439	0.2086	-0.1317	-0.1329	-0.1339	-0.0856	0.0226	-0.0117
Pop_2000	0.1340	0.0173	0.0499	0.0969	0.0287	0.0108	-0.1237	0.0218	0.0621	0.5781	-0.2841	0.7226	-0.1108
Violent	0.0989	0.2932	-0.0653	0.3211	0.0510	0.0055	0.0623	-0.0375	-0.0072	0.3058	-0.2120	-0.2648	0.7618
Property	0.0786	0.2850	-0.1224	0.0939	0.0556	-0.1428	0.0256	-0.0810	-0.0317	-0.5842	0.2356	0.5941	0.3275
Past_JobG	-0.0093	0.1126	-0.0764	-0.3858	0.0573	-0.2454	0.1444	-0.1656	0.0544	-0.2673	-0.8029	-0.0244	-0.0322

US cities - loading vectors LD

```
LD = df_R * np.sqrt(pca.explained_variance_)
LD.columns = range(1,14)
LD = LD.add_prefix('L')
LD.round(4)
```

loading vector1 loading vector2 loading vector3 ...

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13
Life_cost	-0.1169	0.0987	-0.2143	-0.0835	0.0094	0.0540	0.0120	0.0422	0.0434	0.0138	0.0089	0.0022	0.0002
Transport	0.2400	-0.0015	-0.0690	-0.0253	0.0187	-0.0052	-0.1390	0.0047	-0.0140	-0.0091	-0.0044	-0.0100	0.0004
Jobs	0.2059	0.1032	0.0335	-0.1415	0.0266	-0.0585	0.0308	-0.0369	-0.0173	0.0320	0.0321	-0.0038	0.0013
Educ	0.2302	-0.0994	-0.0552	0.0139	0.0040	-0.0413	0.0509	0.1086	-0.0472	-0.0042	-0.0061	0.0005	0.0003
Climate	0.0424	0.2016	0.1522	-0.0472	-0.0929	0.0359	-0.0179	0.0711	0.0190	-0.0091	-0.0048	-0.0011	-0.0001
Crime	-0.0982	-0.2411	0.0515	-0.1169	-0.0291	0.0193	-0.0178	0.0125	0.0078	0.0004	0.0022	0.0122	0.0208
Arts	0.2415	-0.0797	0.0192	0.0307	0.0124	-0.0251	0.0160	0.0017	0.1236	-0.0082	0.0045	-0.0048	-0.0001
H_Care	0.1939	-0.0425	-0.0883	0.0096	-0.1741	0.0272	0.0239	-0.0534	-0.0145	0.0001	-0.0020	0.0012	-0.0009
Rec	0.2391	-0.0161	0.0416	-0.0193	0.0854	0.1297	0.0341	-0.0205	-0.0193	-0.0136	-0.0073	0.0017	-0.0004
Pop_2000	0.0785	0.0071	0.0152	0.0241	0.0064	0.0019	-0.0202	0.0034	0.0090	0.0587	-0.0243	0.0535	-0.0042
Violent	0.0579	0.1200	-0.0199	0.0799	0.0113	0.0010	0.0102	-0.0058	-0.0010	0.0311	-0.0181	-0.0196	0.0291
Property	0.0460	0.1167	-0.0374	0.0234	0.0123	-0.0249	0.0042	-0.0126	-0.0046	-0.0593	0.0201	0.0440	0.0125
Past_JobG	-0.0054	0.0461	-0.0233	-0.0960	0.0127	-0.0428	0.0236	-0.0258	0.0079	-0.0271	-0.0686	-0.0018	-0.0012

US cities - loading vectors L1, L2

.

```
L1 = LD['L1'].values
L1
```

```
array([-0.1169479 ,  0.24002067,  0.20589853,  0.23024314,  0.04235397,
       -0.09819206,  0.24148301,  0.19392203,  0.23909114,  0.0784726 ,
        0.0579252 ,  0.04602043, -0.00544999])
```

```
L2 = LD['L2'].values
L2
```

```
array([ 0.09867279, -0.00148099,  0.1032179 , -0.09939961,  0.20162807,
       -0.24111569, -0.07965386, -0.04250867, -0.01605206,  0.00707939,
        0.1200489 ,  0.11672158,  0.04611587])
```

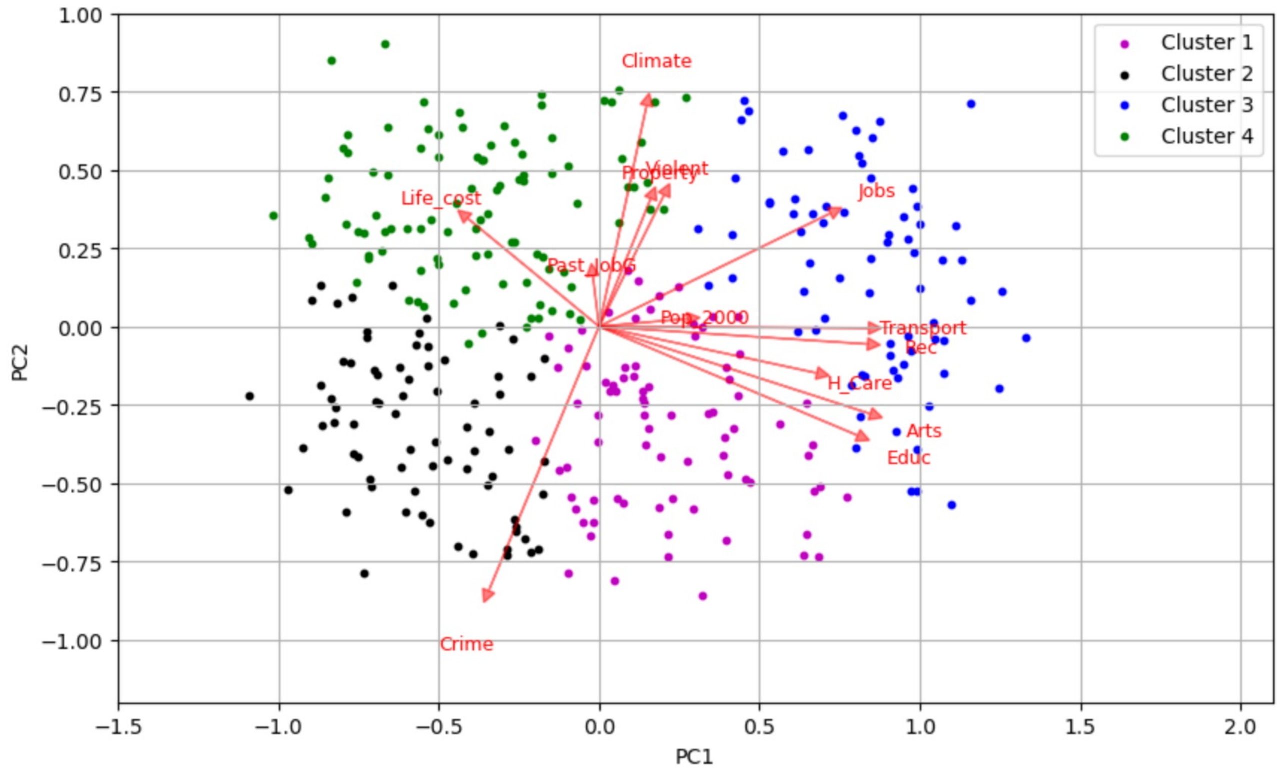
```
# adjust size of vectors for scatterplot
L1 = 3.5 * L1
L2 = 3.5 * L2
```

US cities - biplot

```
# plot cities
plt.scatter(x0, y0,s=10,c = 'm',
            label='Cluster 1')
plt.scatter(x1, y1,s=10,c = 'k',
            label='Cluster 2')
plt.scatter(x2, y2,s=10,c = 'b',
            label='Cluster 3')
plt.scatter(x3, y3,s=10,c = 'g',
            label='Cluster 4')
```

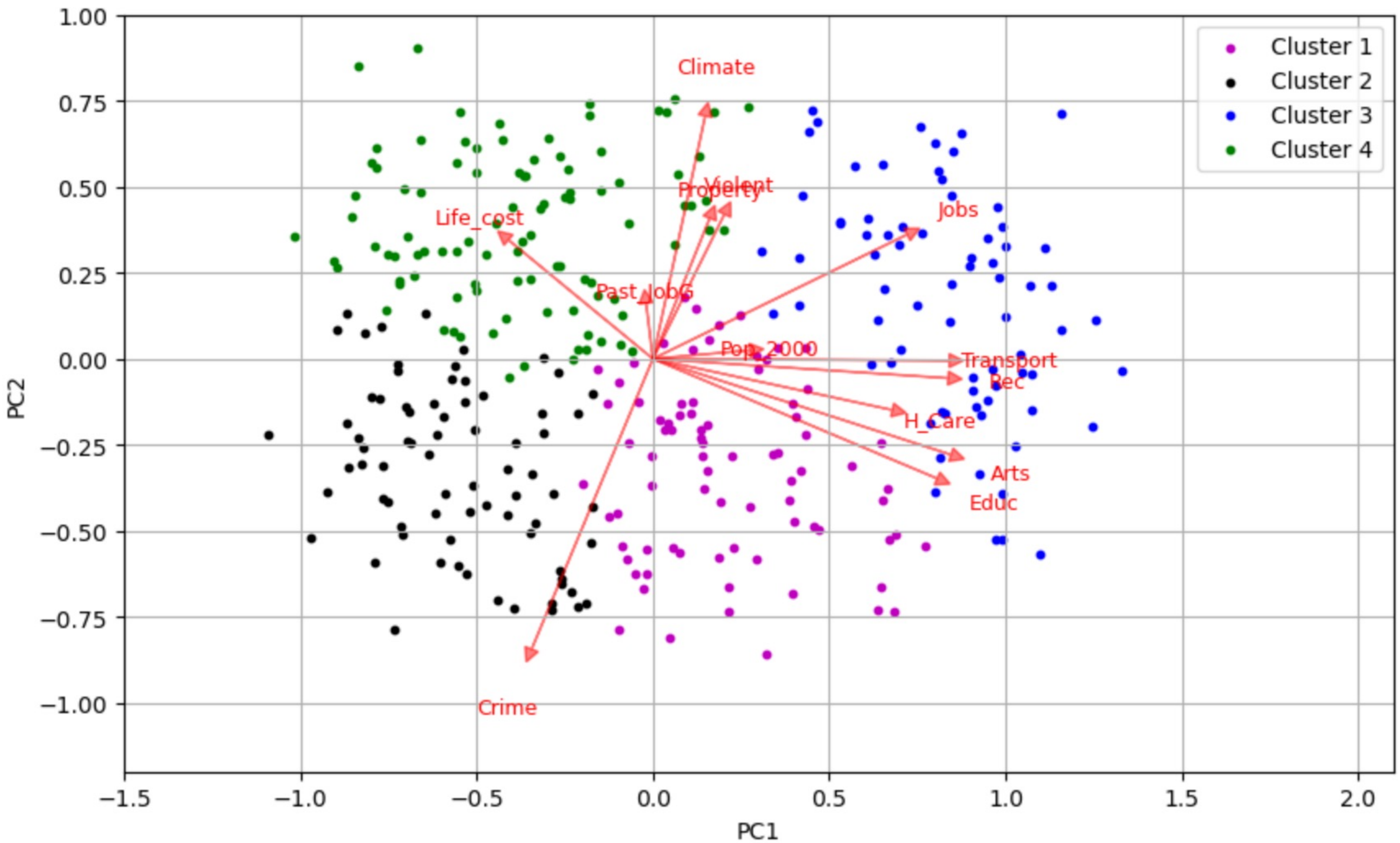
```
# plot loading vectors
for i in range(13):
    plt.arrow(0,0,L1[i],L2[i],
              color = 'r',
              alpha=0.5,
              head_width=0.04,
              head_length=0.04)
    plt.text(1.2*L1[i],1.2*L2[i],
             LD.index[i],
             color ='r',size=9,
             ha = 'center',
             va = 'center')

plt.legend()
```



US cities - biplot

.



Cluster 4: largest Cost-living, Climate; smallest Education, Arts

Cluster 3: largest Population, Jobs, Transportation, Health Care, Arts, Education; smallest Crime

Cluster 2: largest Crime; smallest Transportation, Climate, Health Care

Cluster 1: median Population; low Cost-living, Climate; high Jobs, Education, Crime, Arts

US cities

add column with cluster IDs to original dataframe ¶

```
df_4 = df.copy()
df_4.index = range(325)
df_4[:5]
```

	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property	Past_JobG
0	96.32	36.54	17.28	49.29	55.52	49.58	27.20	45.04	2.83	123711	582	4396	6.1
1	47.31	69.68	86.11	71.95	22.66	54.11	81.59	24.07	77.33	689538	518	4527	11.6
2	86.12	28.02	32.01	26.62	75.63	15.59	33.15	20.11	6.79	120838	761	7036	6.6
3	25.22	82.71	52.97	99.43	8.78	73.94	79.61	77.33	77.62	885782	365	3531	5.2
4	44.48	84.13	90.65	71.67	78.18	2.84	75.36	77.90	70.25	734255	1133	7261	21.4

325 rows × 13 columns

```
df_4['cluster'] = df_pca4['cluster']+1
df_4
```

	Life_cost	Transport	Jobs	Educ	Climate	Crime	Arts	H_Care	Rec	Pop_2000	Violent	Property	Past_JobG	cluster
0	96.32	36.54	17.28	49.29	55.52	49.58	27.20	45.04	2.83	123711	582	4396	6.1	4
1	47.31	69.68	86.11	71.95	22.66	54.11	81.59	24.07	77.33	689538	518	4527	11.6	1
2	86.12	28.02	32.01	26.62	75.63	15.59	33.15	20.11	6.79	120838	761	7036	6.6	4
3	25.22	82.71	52.97	99.43	8.78	73.94	79.61	77.33	77.62	885782	365	3531	5.2	1
4	44.48	84.13	90.65	71.67	78.18	2.84	75.36	77.90	70.25	734255	1133	7261	21.4	3

US cities – pivot tables

```
ptable1 = df_4.pivot_table(index = 'cluster')
ptable1.index.names = ['Cluster']
ptable1.round(2)
```

Cluster	Arts	Climate	Crime	Educ	H_Care	Jobs	Life_cost	Past_JobG	Pop_2000	Property	Rec	Transport	Violent
1	63.26	39.65	66.49	69.87	62.61	52.54	44.81	9.81	488830.62	4229.99	58.66	56.52	391.71
2	35.44	32.90	79.16	32.53	26.56	25.54	52.03	9.23	176179.93	3695.08	31.78	25.63	288.92
3	81.91	64.83	24.44	77.15	72.36	83.44	40.22	11.03	1942311.67	5886.15	82.97	84.45	835.32
4	24.93	67.08	30.05	26.51	33.66	44.88	66.32	12.25	255634.93	5692.92	32.87	33.68	717.22

```
ptable2 = df_4.pivot_table(index = 'cluster',aggfunc = np.median)
ptable2.index.names = ['Cluster']
ptable2.round(2)
```

Cluster	Arts	Climate	Crime	Educ	H_Care	Jobs	Life_cost	Past_JobG	Pop_2000	Property	Rec	Transport	Violent
1	63.60	36.68	69.55	69.54	66.56	55.67	46.89	10.15	377325.0	4233.5	59.76	61.18	355.5
2	33.58	31.72	83.00	33.85	22.66	21.38	57.52	8.45	144558.5	3645.0	30.73	19.12	253.5
3	86.12	69.40	20.40	82.71	73.93	86.40	39.38	11.50	1502584.0	5809.0	84.13	86.68	791.0
4	23.38	71.66	27.76	24.50	30.44	44.61	75.78	10.95	199160.5	5638.0	24.22	32.15	710.5

US cities – largest average values

```
ptable1 = df_4.pivot_table(index = 'cluster')
ptable1.index.names = ['Cluster']
ptable1.round(2)
```

Cluster	Arts	Climate	Crime	Educ	H_Care	Jobs	Life_cost	Past_JobG	Pop_2000	Property	Rec	Transport	Violent
1	63.26	39.65	66.49	69.87	62.61	52.54	44.81	9.81	488830.62	4229.99	58.66	56.52	391.71
2	35.44	32.90	79.16	32.53	26.56	25.54	52.03	9.23	176179.93	3695.08	31.78	25.63	288.92
3	81.91	64.83	24.44	77.15	72.36	83.44	40.22	11.03	1942311.67	5886.15	82.97	84.45	835.32
4	24.93	67.08	30.05	26.51	33.66	44.88	66.32	12.25	255634.93	5692.92	32.87	33.68	717.22

Cluster 4: largest Cost-living, Climate;

Cluster 3: largest Population, Jobs, Transportation, Health Care, Arts, Education, Violent

Cluster 2: largest Crime;

Cluster 1: high Jobs, Education, Crime, Arts

US cities – smallest average values

```
ptable1 = df_4.pivot_table(index = 'cluster')
ptable1.index.names = ['Cluster']
ptable1.round(2)
```

Cluster	Arts	Climate	Crime	Educ	H_Care	Jobs	Life_cost	Past_JobG	Pop_2000	Property	Rec	Transport	Violent
1	63.26	39.65	66.49	69.87	62.61	52.54	44.81	9.81	488830.62	4229.99	58.66	56.52	391.71
2	35.44	32.90	79.16	32.53	26.56	25.54	52.03	9.23	176179.93	3695.08	31.78	25.63	288.92
3	81.91	64.83	24.44	77.15	72.36	83.44	40.22	11.03	1942311.67	5886.15	82.97	84.45	835.32
4	24.93	67.08	30.05	26.51	33.66	44.88	66.32	12.25	255634.93	5692.92	32.87	33.68	717.22

Cluster 4: largest Cost-living, Climate; smallest Education, Arts

Cluster 3: largest Population, Jobs, Transportation, Health Care, Arts, Education; smallest Life_cost, Crime

Cluster 2: largest Crime; smallest Transportation, Climate, Health Care

Cluster 1: median Population; low Cost-living, Climate; high Jobs, Education, Crime, Arts