PCA_19

February 9, 2021

[62]:
```
import sklearn as sk
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[9]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

[109]:
```
import matplotlib.patches as patches
```

[96]:
```
df1 = pd.read_csv('dcom_19.csv')
df1
```

[96]:

	region	dam_gir	inv	shr_danax	dasaqm	daqir	shual_moxm	saq_momsax	prod_gam	brunva
0	Tbilisi	22077.4	912232258	7292.7	463871	453626	14390.1	45768.9	29768.8	81837.0
1	Adjara	4377.1	198094567	843.5	74804	67818	2044.0	4687.0	4426.1	7399.4
2	Guria	723.6	1652961	62.0	9674	7615	231.3	409.6	388.7	558.7
3	Imereti	3654.7	44466877	454.8	58015	47852	1454.4	3219.0	2701.6	4285.1
4	Kakheti	2187.6	199725	182.9	24619	19750	788.6	1551.7	1269.7	1871.8
5	Mtskheta_Mtianeti	1040.7	7860163	132.4	12344	10232	673.2	911.8	1034.4	1204.9
6	Racha_Lechkhumi_Qvemo_Svaneti	261.9	1744620	15.0	2603	1958	53.8	87.7	90.9	110.7
7	Samegrelo_Zemo_Svaneti	2466.1	44462707	288.6	31889	25957	1142.4	2372.7	2079.7	3200.3
8	Samtskhe_Javakheti	1422.5	50734222	101.2	14006	10419	502.0	945.0	998.3	1420.0
9	Qvemo_Qartli	3227.9	48082629	440.9	43751	39607	2329.1	3807.1	3638.2	5130.3
10	Shida_Qartli	1698.3	1243012	143.0	19991					

```
10   15837       578.7      1560.0    1008.2   1916.0
```

[97]:
```
df2 = pd.read_csv('dper_19.csv')
df2
```

[97]:

	region	dam_gir	inv	shr_danax
0	Tbilisi	18.851849	778953.341300	6.227222
1	Adjara	12.541834	567606.209200	2.416905
2	Guria	6.614260	15109.332720	0.566728
3	Imereti	7.347608	89398.626860	0.914355
4	Kakheti	7.000320	639.120000	0.585280
5	Mtskheta_Mtianeti	11.118590	83976.100430	1.414530
6	Racha_Lechkhumi_Qvemo_Svaneti	8.818182	58741.414140	0.505051
7	Samegrelo_Zemo_Svaneti	7.799178	140615.771700	0.912713
8	Samtskhe_Javakheti	9.231019	329229.214800	0.656716
9	Qvemo_Qartli	7.451293	110994.065100	1.017775
10	Shida_Qartli	6.600466	4830.983288	0.555771

	dasaqm	daqir	shual_moxm	saq_momsax	prod_gam	brunva
0	396.098540	387.350354	12.287678	39.081974	25.419520	69.880454
1	214.338109	194.320917	5.856734	13.429799	12.682235	21.201719
2	88.427788	69.606947	2.114260	3.744059	3.553016	5.106947
3	116.636510	96.204262	2.924005	6.471653	5.431444	8.614998
4	78.780800	63.200000	2.523520	4.965440	4.063040	5.989760
5	131.880342	109.316239	7.192308	9.741453	11.051282	12.872863
6	87.643098	65.925926	1.811448	2.952862	3.060606	3.727273
7	100.850727	82.090449	3.612903	7.503795	6.577166	10.121126
8	90.889033	67.611940	3.257625	6.132382	6.478261	9.214796
9	100.994922	91.428901	5.376500	8.788319	8.398430	11.842798
10	77.695297	61.550719	2.249126	6.062962	3.918383	7.446560

[101]:
```
features = ['dam_gir', 'inv', 'shr_danax', 'dasaqm', 'daqir', 'shual_moxm',␣
 ↪'saq_momsax', 'prod_gam', 'brunva']
x = df1.loc[:, features].values
x = StandardScaler().fit_transform(x)

y = df2.loc[:, features].values
y = StandardScaler().fit_transform(y)

pca = PCA(n_components=1)
principalComponents1 = pca.fit_transform(x)
principalDf1 = pd.DataFrame(data = principalComponents1, columns = ['absolute'])
principalComponents2 = pca.fit_transform(y)
principalDf2 = pd.DataFrame(data = principalComponents2, columns = ['relative'])
```

[102]:
```
finalDf = pd.concat([df1['region'], principalDf1, principalDf2], axis = 1)
finalDf
```

[102]:

	region	absolute	relative
0	Tbilisi	9.367154	8.465208
1	Adjara	0.067038	2.190107
2	Guria	-1.387287	-1.946357
3	Imereti	-0.538945	-1.152786
4	Kakheti	-1.081653	-1.851998
5	Mtskheta_Mtianeti	-1.248975	0.376341
6	Racha_Lechkhumi_Qvemo_Svaneti	-1.497164	-1.824770
7	Samegrelo_Zemo_Svaneti	-0.851227	-0.953600
8	Samtskhe_Javakheti	-1.184141	-0.831067
9	Qvemo_Qartli	-0.479807	-0.598172
10	Shida_Qartli	-1.164993	-1.872905

[100]:
```
print('explained variance ratio ', round(pca.
 ↪explained_variance_ratio_[0]*100,2))
```

```
explained variance ratio  94.89
```

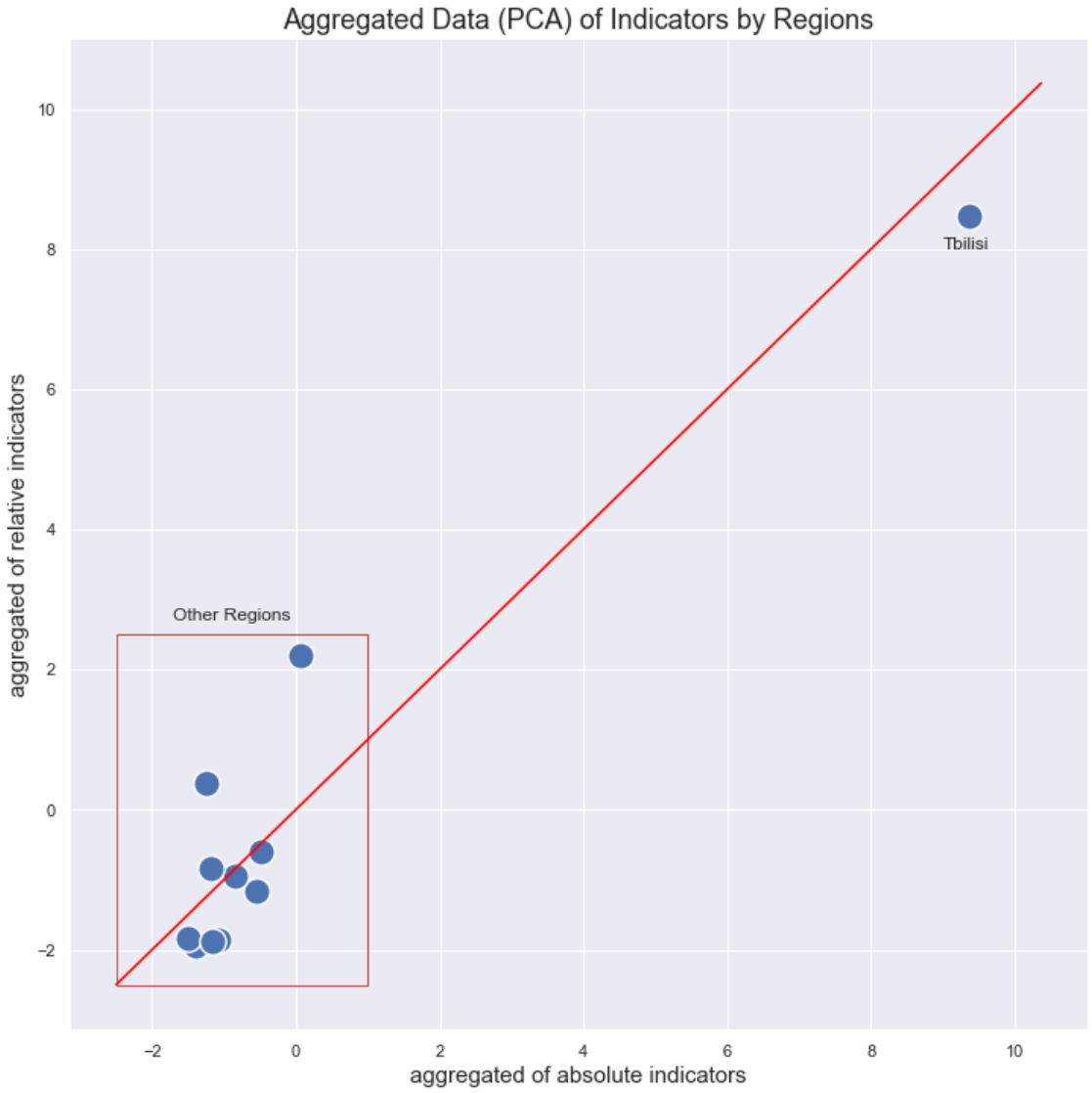
[216]:
```
cluster = np.matrix(finalDf.loc[1:,['absolute', 'relative']])

centroide = (cluster.sum(axis=0)[0,0]/cluster.shape[0], cluster.
 ↪sum(axis=0)[0,1]/cluster.shape[0])
print('Centroide for cluster of regions without Tbilisi ',centroide)

distance =np.power(np.power((centroide[0]-cluster[:,0]),2)+np.
 ↪power((centroide[1]-cluster[:,1]),2), 0.5)
cluster = np.hstack((cluster,distance))
mean_distance_other = cluster[:,2].mean()
distance_tbs = ((centroide[0]-finalDf.loc[0,'absolute'])**2 +␣
 ↪(centroide[1]-finalDf.loc[0,'relative'])**2)**0.5
print('Metrics (euclidean distance) for Tbilisi to centroide of other regions␣
 ↪cluster - ',distance_tbs)
print('Mean of metrics(euclidean distans) for other regions to cluster␣
 ↪centroide - ', mean_distance_other)
print('ratio - ', round(distance_tbs/mean_distance_other,1))
```

```
Centroide for cluster of regions without Tbilisi  (-0.9367153982335654,
-0.8465207822709667)
Metrics (euclidean distance) for Tbilisi to centroide of other regions cluster -
13.888052916973152
Mean of metrics(euclidean distans) for other regions to cluster centroide -
1.0250608719834524
ratio -  13.5
```

[164]:
```
finalDf_graf = finalDf

fig = plt.figure(figsize = (12,12))
ax = fig.add_subplot(1,1,1)
```

```
ax.set_xlabel('aggregated of absolute indicators', fontsize = 15)
ax.set_ylabel('aggregated of relative indicators', fontsize = 15)
ax.set_title('Aggregated Data (PCA) of Indicators by Regions', fontsize = 18)

sns.set_theme(style="darkgrid")
sns.scatterplot(data=finalDf_graf, x="absolute",
                y="relative", s=300)

x = np.linspace(min(finalDf_graf["absolute"])-1,
                max(finalDf_graf["absolute"])+1, 100)
sns.lineplot(x=x, y=x, color='red')

rect = patches.Rectangle((-2.5,-2.5),3.
 ↪5,5,linewidth=1,edgecolor='r',facecolor='none')
ax.add_patch(rect)

ax.annotate('Tbilisi', xy=(9,8), xytext=(9,8))
ax.annotate('Other Regions', xy=(-1.7,2.7), xytext=(-1.7,2.7))
```

[164]: Text(-1.7, 2.7, 'Other Regions')



[169]:
```
finalDf_graf = finalDf.loc[1:, :]

fig = plt.figure(figsize = (12,12))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('aggregated of absolute indicators', fontsize = 15)
ax.set_ylabel('aggregated of relative indicators', fontsize = 15)
ax.set_title('Aggregated Data (PCA) of Indicators by Regions(without Tbilisi)',␣
 ↪fontsize = 18)

sns.set_theme(style="darkgrid")
sns.scatterplot(data=finalDf_graf, x="absolute",
                y="relative", style ="region", s=300)
```

```
x = np.linspace(min(finalDf_graf["absolute"])-1,
                max(finalDf_graf["absolute"])+1, 100)
sns.lineplot(x=x, y=x, color='red')

names = ['Adjara', 'Guria', 'Imereti', 'Kakheti', 'Mtskheta_Mtianeti',␣
 ↪'Racha_Lechkhumi_Qvemo_Svaneti', 'Samegrelo_Zemo_Svaneti',␣
 ↪'Samtskhe_Javakheti', 'Qvemo_Qartli', 'Shida_Qartli']
pos = [(0,2), (-1.5, -2.1), (-0.6, -1.3), (-1, -1.8), (-1.5, 0.5), (-2.4, -1.
 ↪7), (-1.5, -1.1), (-1.4, -0.7), (-0.4, -0.6), (-1.2, -2)]
for name,p in zip(names,pos):
    ax.annotate(name, xy=p, xytext=p)
```