Infraestrutura

> Versão do Python e ambiente virtual

```
humberto@humberto-linux:~/Documentos/PROJ/POS_INFINET/classificacao-nao-supervisionada$ source /home/humberto/Documentos/PROJ/POS_INFINET/classificacao-nao-supervisionada/venv/bin/activate
(venv) humberto@humberto-linux:~/Documentos/PROJ/POS_INFINET/classificacao-nao-supervisionada$ python --version
Python 3.10.6
```

requirements.txt

```
matplotlib==3.6.2
matplotlib-inline==0.1.6
numpy==1.24.1
pandas==1.5.2
scikit-learn==1.2.0
scipy==1.9.3
seaborn==0.12.2
```

Libs

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.spatial import distance
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
```

Escolha de base de dados

1. Baixe os dados disponibilizados na plataforma Kaggle sobre dados sócio-econômicos e de saúde que determinam o índice de desenvolvimento de um país. Esses dados estão disponibilizados através do link: https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data

In [ ]:
```
df = pd.read_csv('Country-data.csv')
df
```

Out[ ]:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fe
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.8
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.6
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.8
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.1
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.1
...	...	...	...	...	...	...	...	...	.
162	Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.5
163	Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.4
164	Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.9
165	Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.6
166	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.4

167 rows × 10 columns

2. Quantos paises existem no dataset?

Resposta:

In [ ]:
```
df.country.nunique()
```

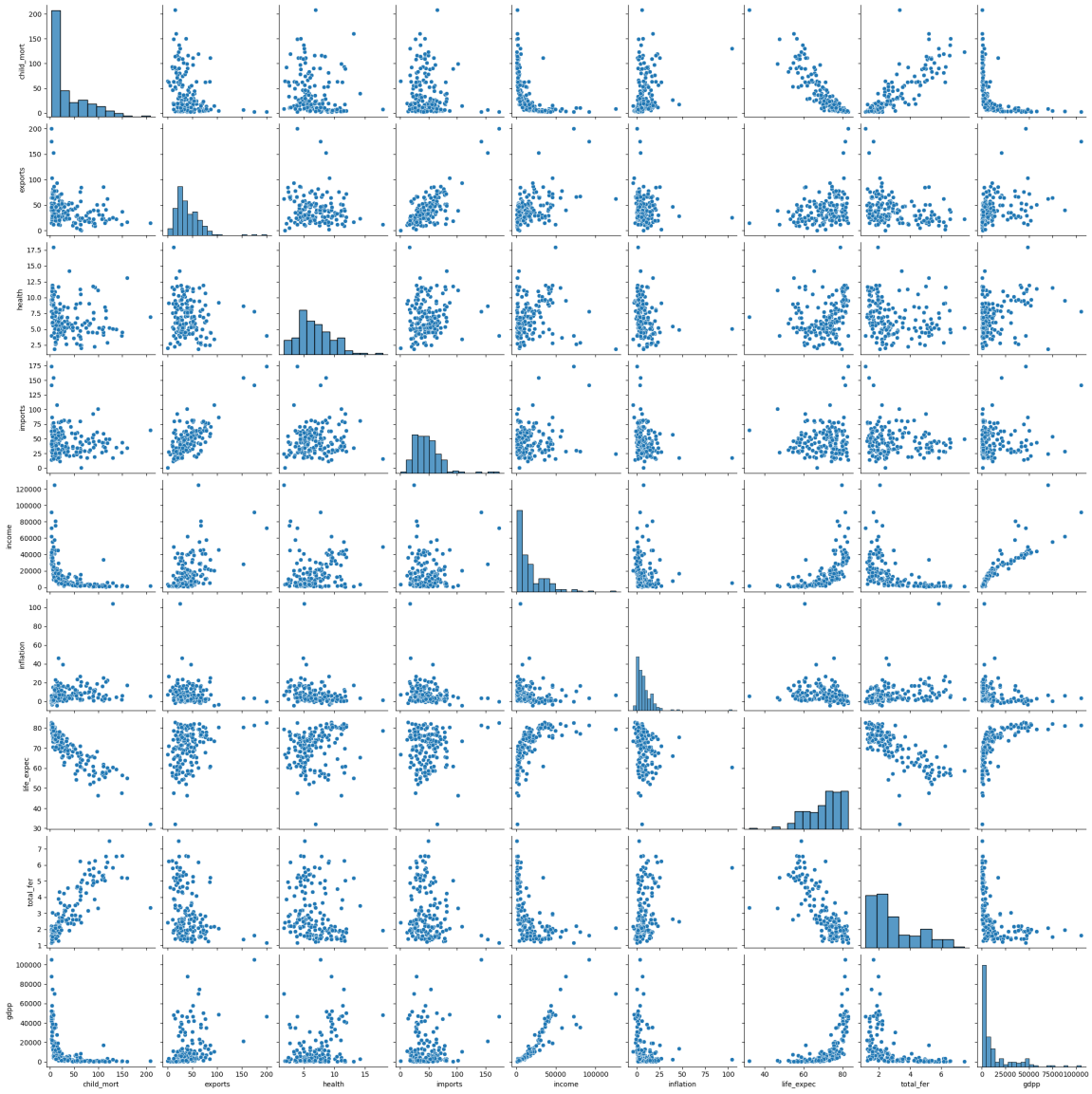
Out[ ]:
```
167
```

3. Mostre através de gráficos a faixa dinâmica das variáveis que serão usadas nas tarefas

de clusterização. Analise os resultados mostrados. O que deve ser feito com os dados antes da etapa de clusterização?

In [ ]:
```
sns.pairplot(df)
```

Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7f8e4e04e230>
```



Resposta: As variáveis devem ser padronizadas, pois estão fora de escala

4. Realize o pré-processamento adequado dos dados.

In [ ]:
```
df_norm = (df-df.mean())/df.std()
df_norm['country'] = df.country
df_norm
```

```
/tmp/ipykernel_8247/1760102069.py:1: FutureWarning: The default value
of numeric_only in DataFrame.mean is deprecated. In a future version,
it will default to False. In addition, specifying 'numeric_only=None'
is deprecated. Select only valid columns or specify the value of numer
ic_only to silence this warning.
  df_norm = (df-df.mean())/df.std()
/tmp/ipykernel_8247/1760102069.py:1: FutureWarning: The default value
of numeric_only in DataFrame.std is deprecated. In a future version, i
t will default to False. In addition, specifying 'numeric_only=None' i
s deprecated. Select only valid columns or specify the value of numeri
c_only to silence this warning.
  df_norm = (df-df.mean())/df.std()
```

Out[ ]:

	child_mort	country	exports	gdpp	health	imports	income	inflat
0	1.287660	Afghanistan	-1.134867	-0.677143	0.278251	-0.082208	-0.805822	0.156
1	-0.537333	Albania	-0.478220	-0.484167	-0.096725	0.070624	-0.374243	-0.311
2	-0.272015	Algeria	-0.098824	-0.463980	-0.963176	-0.639838	-0.220182	0.786
3	2.001787	Angola	0.773056	-0.514720	-1.443729	-0.164820	-0.583289	1.382
4	-0.693548	Antigua and Barbuda	0.160186	-0.041692	-0.286034	0.496076	0.101427	-0.599
...	...	...	...	...	...	...	...	
162	-0.224902	Vanuatu	0.200315	-0.545273	-0.569997	0.239979	-0.736313	-0.488
163	-0.524935	Venezuela	-0.459980	0.029235	-0.693776	-1.209860	-0.033442	3.606
164	-0.371199	Vietnam	1.126916	-0.635842	0.008851	1.375892	-0.656429	0.408
165	0.447072	Yemen	-0.405259	-0.635842	-0.595481	-0.515920	-0.656948	1.496
166	1.111607	Zambia	-0.149897	-0.627658	-0.337002	-0.660491	-0.719195	0.588

167 rows × 10 columns

Clusterização

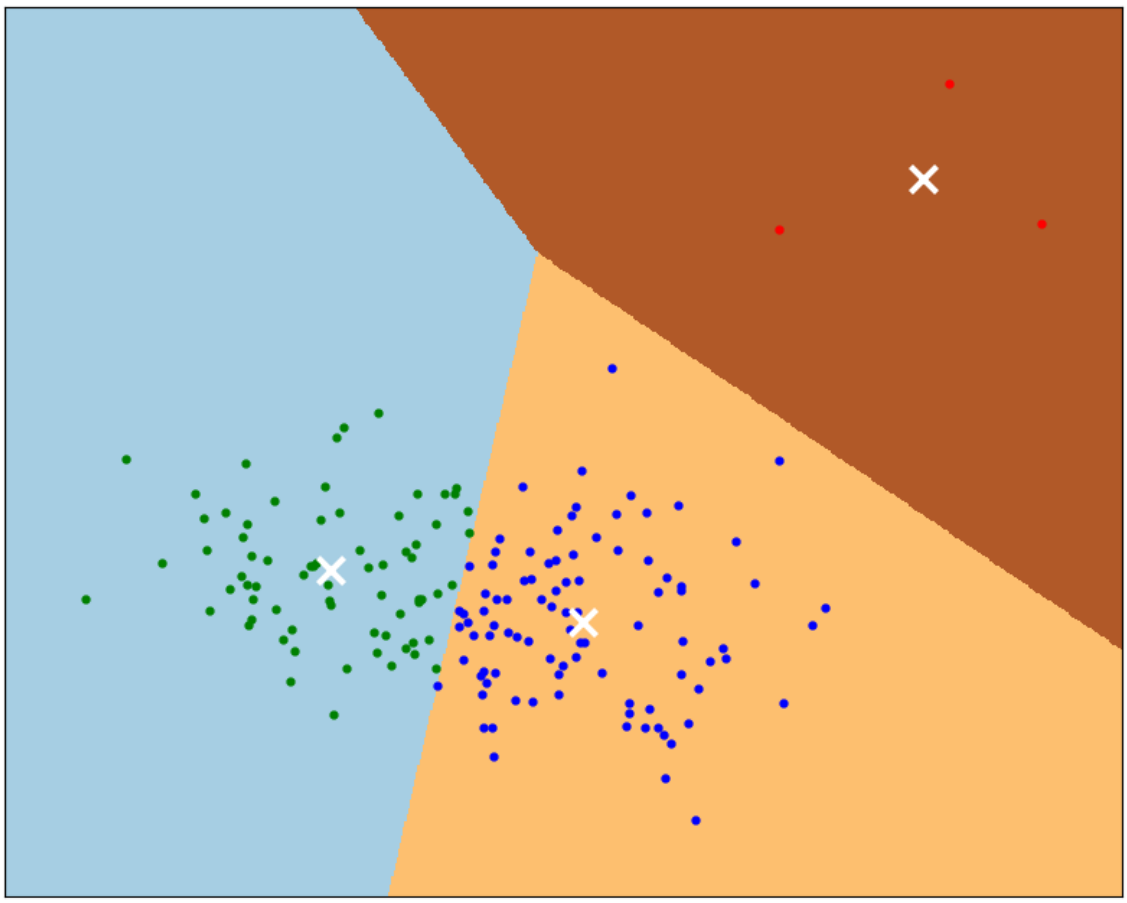
1. Realizar o agrupamento dos países em 3 grupos distintos. Para tal, use:

    a. K-Médias

In [ ]:
```
reduced_data = PCA(n_components=2).fit_transform(df_norm.drop(columns=
['country']))
kmeans = KMeans(n_clusters=3, random_state=0)
fit = kmeans.fit(reduced_data)
# Step size of the mesh. Decrease to increase the quality of the VQ.
h = 0.02
# point in the mesh [x_min, x_max]x[y_min, y_max].
# Plot the decision boundary. For that, we will assign a color to each
x_min, x_max = reduced_data[:, 0].min() - 1, reduced_data[:, 0].max() + 1
y_min, y_max = reduced_data[:, 1].min() - 1, reduced_data[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max,
# Obtain labels for each point in mesh. Use last trained model.
Z = kmeans.predict(np.c_[xx.ravel(), yy.ravel()])
# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure(1, figsize=[10, 8])
plt.clf()
plt.imshow(
    Z,
    interpolation="nearest",
    extent=(xx.min(), xx.max(), yy.min(), yy.max()),
    cmap=plt.cm.Paired,
    aspect="auto",
    origin="lower",
)
```

```
labels = pd.Series(kmeans.labels_)
for label in [0, 1, 2]:
    dict_color = {0: 'green', 1: 'blue', 2: 'red'}
    plt.plot(reduced_data[labels[labels == label].index, 0],
reduced_data[labels[labels == label].index, 1], "k.", markersize=7,
color=dict_color[label])
# Plot the centroids as a white X
centroids = kmeans.cluster_centers_
plt.scatter(
centroids[:, 0],
centroids[:, 1],
marker="x",
s=169,
linewidths=3,
color="w",
zorder=10,
)
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()
```

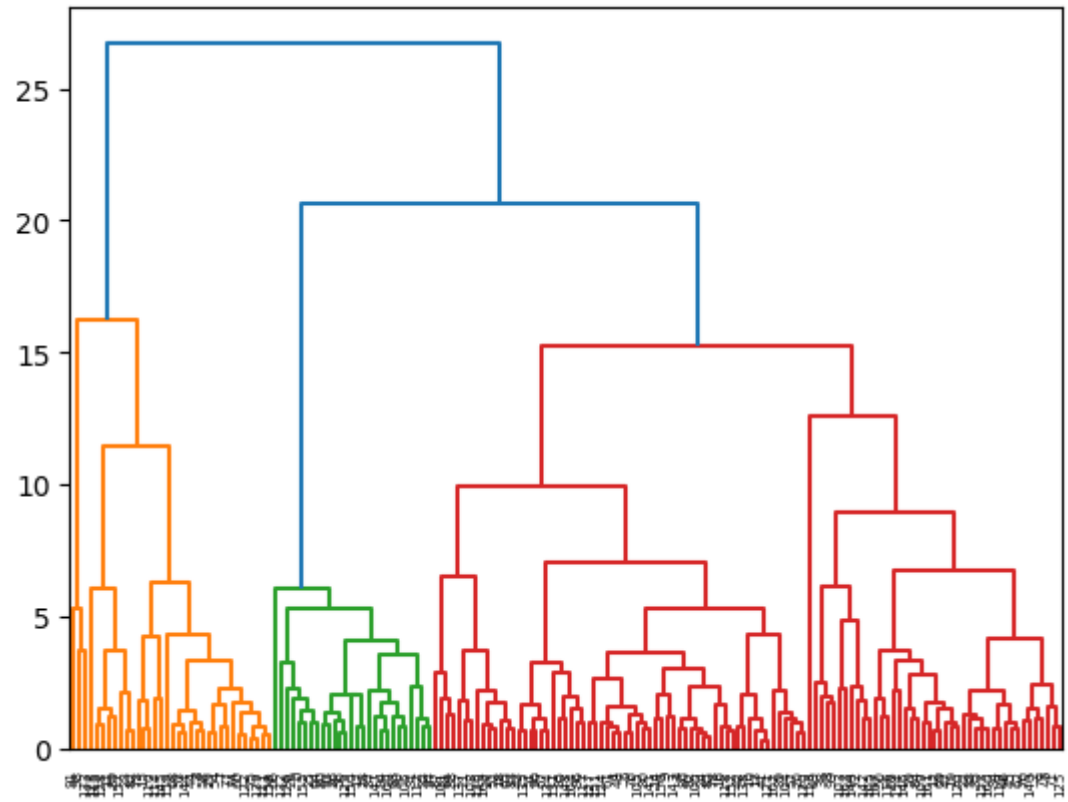
```
/home/humberto/Documentos/PROJ/POS_INFINET/classificacao-nao-supervision
ada/venv/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: Fu
tureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/tmp/ipykernel_8247/2892276334.py:29: UserWarning: color is redundantly
defined by the 'color' keyword argument and the fmt string "k." (-> colo
r='k'). The keyword argument will take precedence.
  plt.plot(reduced_data[labels[labels == label].index, 0],
```



b. Clusterização Hierárquica

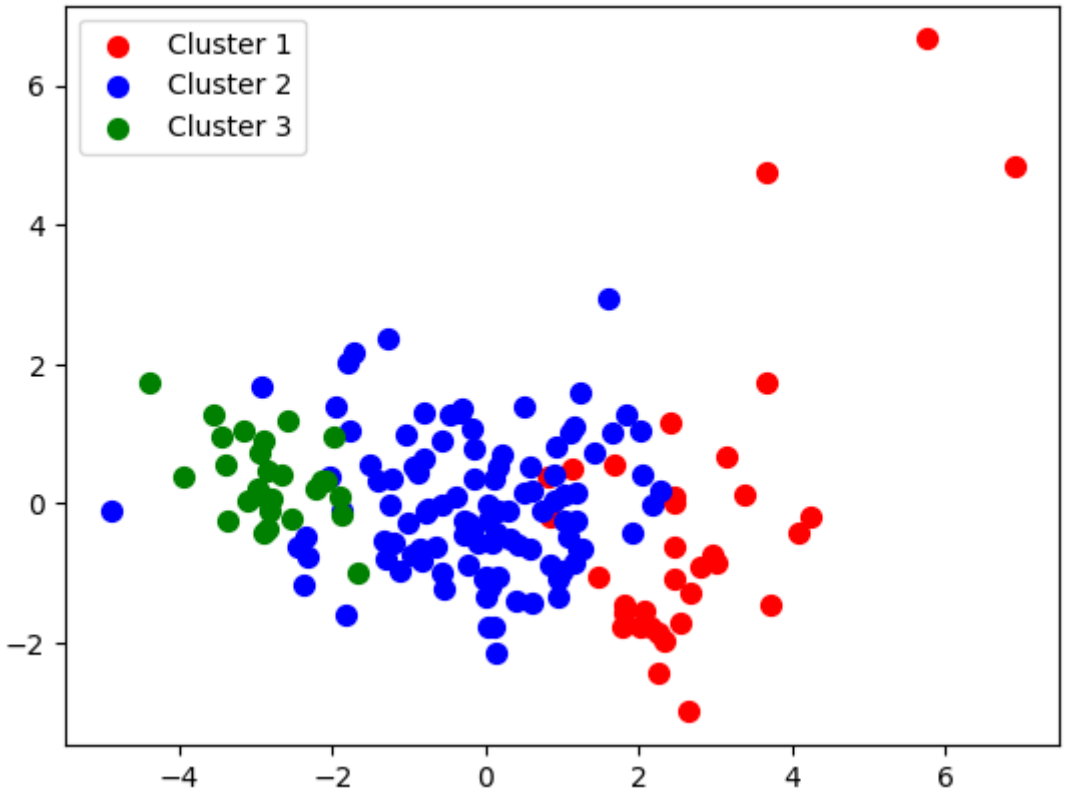
In [ ]:
```
base = df_norm.drop(columns=['country'])
dendrograma = dendrogram(linkage(base, method = 'ward'))
hc = AgglomerativeClustering(n_clusters = 3, affinity = 'euclidean', link
'ward')
previsoes = hc.fit_predict(base)
```

```
/home/humberto/Documentos/PROJ/POS_INFINET/classificacao-nao-supervision
ada/venv/lib/python3.10/site-packages/sklearn/cluster/_agglomerative.py:
983: FutureWarning: Attribute `affinity` was deprecated in version 1.2 a
nd will be removed in 1.4. Use `metric` instead
  warnings.warn(
```



In [ ]:
```
plt.scatter(reduced_data[previsoes == 0, 0], reduced_data[previsoes == 0,
50, c = 'red', label = 'Cluster 1')
plt.scatter(reduced_data[previsoes == 1, 0], reduced_data[previsoes == 1,
50, c = 'blue', label = 'Cluster 2')
plt.scatter(reduced_data[previsoes == 2, 0], reduced_data[previsoes == 2,
50, c = 'green', label = 'Cluster 3')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f8e35fd0a90>
```



In [ ]:
```
df_norm['distance_to_center'] = np.min(fit.transform(reduced_data), axis=
df_norm['label_kmeans'] = fit.labels_
df_norm['label_hclust'] = previsoes
```

2. Para os resultados, do K-Médias:

a. Interprete cada um dos clusters obtidos citando:

i. Qual a distribuição das dimensões em cada grupo

> Resposta: Dataset abaixo

In [ ]:
```
df['label_kmeans'] = df_norm['label_kmeans']
columns = ['child_mort', 'exports', 'gdpp', 'health', 'imports', 'income'
df.groupby('label_kmeans')[columns].mean()
```

Out[ ]:

label_kmeans	child_mort	exports	gdpp	health	imports	income	i
0	74.156338	30.611254	2019.211268	6.015915	44.071351	4363.281690	11
1	11.974194	44.772043	19881.182796	7.426989	45.501075	25390.000000	5
2	4.133333	176.000000	57566.666667	6.793333	156.666667	64033.333333	2

ii. O país, de acordo com o algoritmo, melhor representa o seu agrupamento. Justifique

> Resposta:

- Grupo 0: Senegal
- Grupo 1: Lebanon
- Grupo 2: Singapore

São os países que tem a menor distância ao seu respctivo centroide, e possuem a distribuição das variáveis próximas a média dos seus respectivos grupos.

In [ ]:
```
df_norm.sort_values('distance_to_center').groupby('label_kmeans').agg(
    country=('country', 'first'),
    distance_to_center=('distance_to_center', 'min')
)
```

Out[ ]:

label	country	distance_to_center
0	Senegal	0.200534
1	Lebanon	0.161898
2	Singapore	1.286854

In [ ]:
```
df.query('country in ["Senegal", "Lebanon", "Singapore"]')
```

Out[ ]:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer
86	Lebanon	10.3	35.8	7.03	60.2	16300	0.238	79.8	1.61
129	Senegal	66.8	24.9	5.66	40.3	2180	1.850	64.0	5.06
133	Singapore	2.8	200.0	3.96	174.0	72100	-0.046	82.7	1.15

3. Para os resultados da Clusterização Hierárquica, apresente o dendograma e interprete

os resultados

> Resposta:

- De acordo com a distribuição das variáveis, o algoritimo Clusterização Hierárquica formou os grupos separando as cidades em níveis socioeconômicos, onde no grupo 2 fazem parte os países com piores índices de desenvolvimento. No grupo 1 foram agrupados os países com as melhores medidas, em comparação ao grupo 2 e no grupo 0 foram reunidas os melhores países com os melhores indicadores entre os três grupos.

In [ ]:
```
df['label_hclust'] = df_norm['label_hclust']
columns = ['child_mort', 'exports', 'gdpp', 'health', 'imports', 'income'
df.groupby('label_hclust')[columns].mean()
```

Out[ ]:

label_hclust	child_mort	exports	gdpp	health	imports	income	inflat
0	5.961765	58.508824	43170.588235	8.501176	48.902941	47588.235294	4.115
1	31.617925	39.990368	6407.367925	6.353679	48.085527	11341.886792	9.120
2	105.070370	23.589630	667.888889	6.507037	39.662963	1589.740741	7.142

In [ ]:
```
df.groupby('label_kmeans')[columns].mean()
```

Out[ ]:

label_kmeans	child_mort	exports	gdpp	health	imports	income	i
0	74.156338	30.611254	2019.211268	6.015915	44.071351	4363.281690	11
1	11.974194	44.772043	19881.182796	7.426989	45.501075	25390.000000	5
2	4.133333	176.000000	57566.666667	6.793333	156.666667	64033.333333	2

4. Compare os dois resultados, aponte as semelhanças e diferenças e interprete

> Resposta:

- A classificação geral dos grupos tiveram um resultados seguindo o mesmo intuito, em seprarar os grupos por seus índices socioeconômicos, com maior semelhança entre os grupos de piores indicadores e de valores medianos entre os três grupos (representado pelo label 1 nos dois algorítmos), e com maiores diferenças nas distribuições entre o grupo com os melhores parâmetros das variáveis, sendo possível verificar pela distribuição no gráfico e pela quantidade de países em cada grupo.

In [ ]:
```
df.value_counts('label_hclust')
```

Out[ ]:
```
label_hclust
1    106
0     34
2     27
dtype: int64
```

In [ ]:
```
df.value_counts('label_kmeans')
```

Out[ ]:
```
label_kmeans
1    93
0    71
2     3
dtype: int64
```

Escolha de algoritmos

1. Escreva em tópicos as etapas do algoritmo de K-médias até sua convergência.

> Resposta

A. São escolhidos pontos aletorios para representar os centroides
B. A distâncias entre os centroides e os pontos são calculadas e armazenadas
C. Com base nos cálculos de distância, cada ponto é atribuído ao cluster mais próximo
D. As novas posições do centróide do cluster são atualizadas para o ponto de melhor representatividade do cluster
E. Se os locais dos centróides mudaram, o processo se repete a partir da etapa 2, até que o novo centro calculado permaneça o mesmo ou muito próximo, o que sinaliza que os membros e os centróides dos clusters agora estão definidos.

2. Refaça o algoritmo apresentado na questão 1 a fim de garantir que o cluster seja representado pelo dado mais próximo ao seu baricentro em todas as iterações do algoritmo.

In [ ]:
```
df.query('country in ["Senegal", "Lebanon", "Singapore"]')
```

Out[ ]:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer
86	Lebanon	10.3	35.8	7.03	60.2	16300	0.238	79.8	1.61
129	Senegal	66.8	24.9	5.66	40.3	2180	1.850	64.0	5.06
133	Singapore	2.8	200.0	3.96	174.0	72100	-0.046	82.7	1.15

In [ ]:
```
list_medoides = reduced_data[[129, 86, 133]]
labels_medoides = []
for point in reduced_data:
    list_distance = []
    for medoid in list_medoides:
        list_distance.append(distance.euclidean(point, medoid))
    min_distance = min(list_distance)
    labels_medoides.append(list_distance.index(min_distance))
pd.Series(labels_medoides).value_counts()
```

Out[ ]:
```
1    94
0    70
2     3
dtype: int64
```

In [ ]:
```
df_norm.value_counts('label_kmeans')
```

Out[ ]:
```
label_kmeans
1    93
0    71
2     3
dtype: int64
```

3. O algoritmo de K-médias é sensível a outliers nos dados. Explique.

> Resposta:

- sim, pois usa a média dos pontos para encontrar o centro do cluster,

fazendo com que ele seja suscetível a erros provocados por outliers

4. Por que o algoritmo de DBScan é mais robusto à presença de outliers?

> Resposta:

- O algoritmo do Dbscan agrupa pontos que estão próximos, marcando como outliers pontos que estão sozinhos em regiões de baixa densidade. Fazendo com que ele identifique melhor os outliers.