# CIA Country Analysis and Clustering

Source: All these data sets are made up of data from the US government. https://www.cia.gov/library/publications/the-world-factbook/docs/faqs.html

## Goal:

### Gain insights into similarity between countries and regions of the world by experimenting with different cluster amounts.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('CIA_Country_Facts.csv')
```

# Exploratory Data Analysis

**TASK: Explore the rows and columns of the data as well as the data types of the columns.**

In [3]:
```
df.head()
```

Out[3]:

| | Country | Region | Population | Area (sq. mi.) | Pop. Density (per sq. mi.) | Coastline (coast/area ratio) | Net migration | Infant mortality (per 1000 births) | GDP ($ per capita) | Literacy (%) | Ph |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Afghanistan | ASIA (EX. NEAR EAST) | 31056997 | 647500 | 48.0 | 0.00 | 23.06 | 163.07 | 700.0 | 36.0 | |
| **1** | Albania | EASTERN EUROPE | 3581655 | 28748 | 124.6 | 1.26 | -4.93 | 21.52 | 4500.0 | 86.5 | |
| **2** | Algeria | NORTHERN AFRICA | 32930091 | 2381740 | 13.8 | 0.04 | -0.39 | 31.00 | 6000.0 | 70.0 | |
| **3** | American Samoa | OCEANIA | 57794 | 199 | 290.4 | 58.29 | -20.71 | 9.27 | 8000.0 | 97.0 | |
| **4** | Andorra | WESTERN EUROPE | 71201 | 468 | 152.1 | 0.00 | 6.60 | 4.05 | 19000.0 | 100.0 | |

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 227 entries, 0 to 226
Data columns (total 20 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Country                             227 non-null    object
 1   Region                              227 non-null    object
 2   Population                          227 non-null    int64
```

```
 3   Area (sq. mi.)                        227 non-null    int64
 4   Pop. Density (per sq. mi.)            227 non-null    float64
 5   Coastline (coast/area ratio)          227 non-null    float64
 6   Net migration                         224 non-null    float64
 7   Infant mortality (per 1000 births)    224 non-null    float64
 8   GDP ($ per capita)                    226 non-null    float64
 9   Literacy (%)                          209 non-null    float64
 10  Phones (per 1000)                     223 non-null    float64
 11  Arable (%)                            225 non-null    float64
 12  Crops (%)                             225 non-null    float64
 13  Other (%)                             225 non-null    float64
 14  Climate                               205 non-null    float64
 15  Birthrate                             224 non-null    float64
 16  Deathrate                             223 non-null    float64
 17  Agriculture                           212 non-null    float64
 18  Industry                              211 non-null    float64
 19  Service                               212 non-null    float64
dtypes: float64(16), int64(2), object(2)
memory usage: 35.6+ KB
```

In [5]:
```
df.describe().transpose()
```

Out[5]:

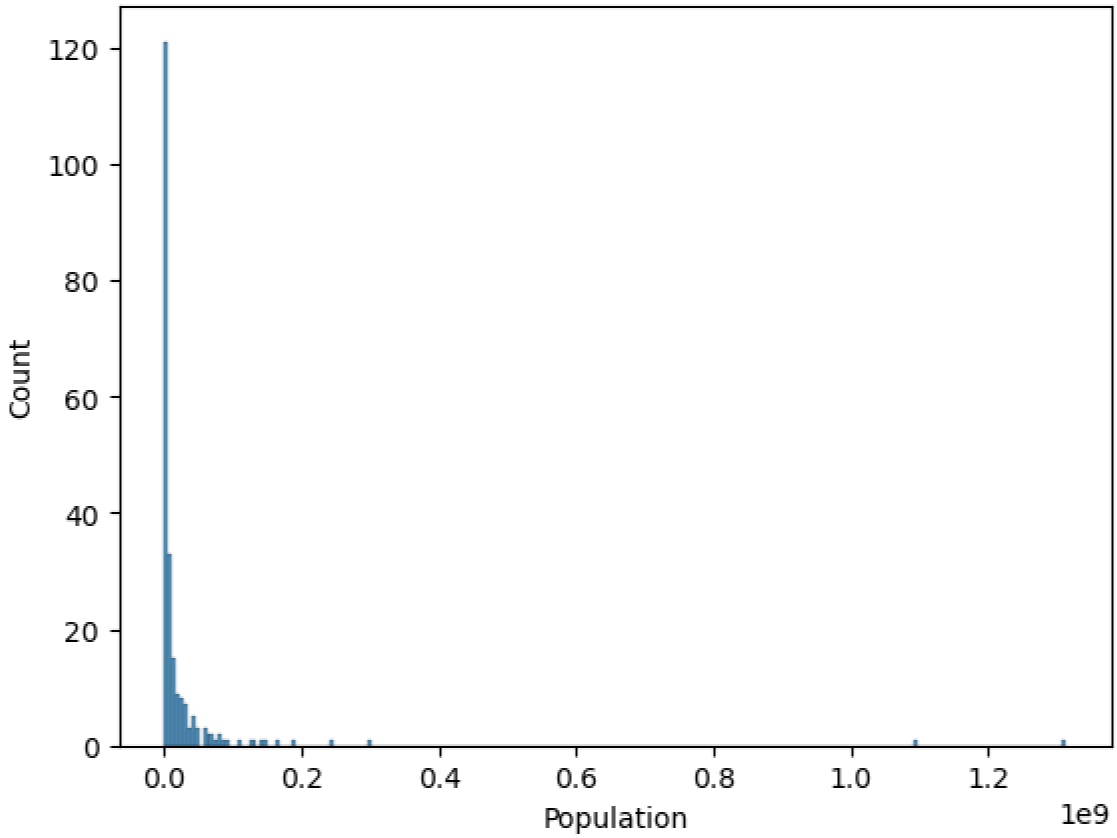
| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Population** | 227.0 | 2.874028e+07 | 1.178913e+08 | 7026.000 | 437624.00000 | 4786994.000 | 1.749777e+07 | 1.313974e+09 |
| **Area (sq. mi.)** | 227.0 | 5.982270e+05 | 1.790282e+06 | 2.000 | 4647.50000 | 86600.000 | 4.418110e+05 | 1.707520e+07 |
| **Pop. Density (per sq. mi.)** | 227.0 | 3.790471e+02 | 1.660186e+03 | 0.000 | 29.15000 | 78.800 | 1.901500e+02 | 1.627150e+04 |
| **Coastline (coast/area ratio)** | 227.0 | 2.116533e+01 | 7.228686e+01 | 0.000 | 0.10000 | 0.730 | 1.034500e+01 | 8.706600e+02 |
| **Net migration** | 224.0 | 3.812500e-02 | 4.889269e+00 | -20.990 | -0.92750 | 0.000 | 9.975000e-01 | 2.306000e+01 |
| **Infant mortality (per 1000 births)** | 224.0 | 3.550696e+01 | 3.538990e+01 | 2.290 | 8.15000 | 21.000 | 5.570500e+01 | 1.911900e+02 |
| **GDP ($ per capita)** | 226.0 | 9.689823e+03 | 1.004914e+04 | 500.000 | 1900.00000 | 5550.000 | 1.570000e+04 | 5.510000e+04 |
| **Literacy (%)** | 209.0 | 8.283828e+01 | 1.972217e+01 | 17.600 | 70.60000 | 92.500 | 9.800000e+01 | 1.000000e+02 |
| **Phones (per 1000)** | 223.0 | 2.360614e+02 | 2.279918e+02 | 0.200 | 37.80000 | 176.200 | 3.896500e+02 | 1.035600e+03 |
| **Arable (%)** | 225.0 | 1.379711e+01 | 1.304040e+01 | 0.000 | 3.22000 | 10.420 | 2.000000e+01 | 6.211000e+01 |
| **Crops (%)** | 225.0 | 4.564222e+00 | 8.361470e+00 | 0.000 | 0.19000 | 1.030 | 4.440000e+00 | 5.068000e+01 |
| **Other (%)** | 225.0 | 8.163831e+01 | 1.614083e+01 | 33.330 | 71.65000 | 85.700 | 9.544000e+01 | 1.000000e+02 |
| **Climate** | 205.0 | 2.139024e+00 | 6.993968e-01 | 1.000 | 2.00000 | 2.000 | 3.000000e+00 | 4.000000e+00 |
| **Birthrate** | 224.0 | 2.211473e+01 | 1.117672e+01 | 7.290 | 12.67250 | 18.790 | 2.982000e+01 | 5.073000e+01 |
| **Deathrate** | 223.0 | 9.241345e+00 | 4.990026e+00 | 2.290 | 5.91000 | 7.840 | 1.060500e+01 | 2.974000e+01 |
| **Agriculture** | 212.0 | 1.508443e-01 | 1.467980e-01 | 0.000 | 0.03775 | 0.099 | 2.210000e-01 | 7.690000e-01 |
| **Industry** | 211.0 | 2.827109e-01 | 1.382722e-01 | 0.020 | 0.19300 | 0.272 | 3.410000e-01 | 9.060000e-01 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Service** | 212.0 | 5.652830e-01 | 1.658410e-01 | 0.062 | 0.42925 | 0.571 | 6.785000e-01 | 9.540000e-01 |

# Exploratory Data Analysis

In [6]:
```
sns.histplot(data=df, x='Population')
```

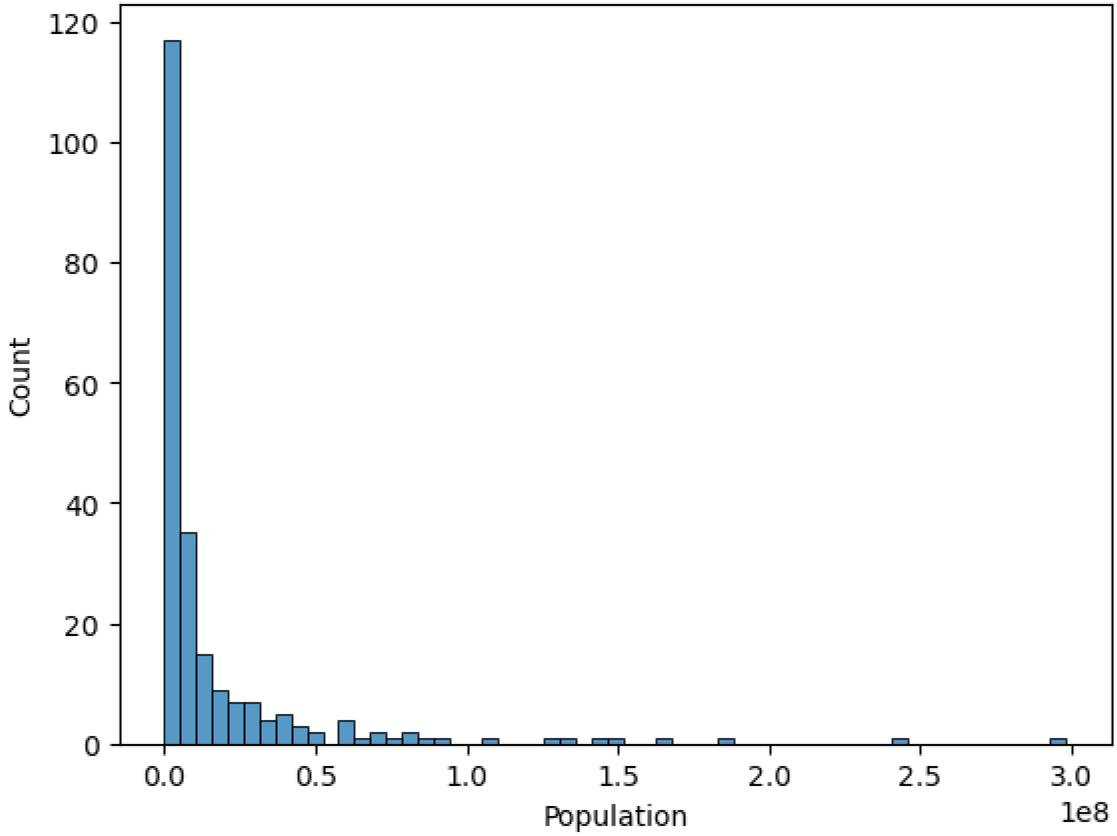
Out[6]:
```
<AxesSubplot:xlabel='Population', ylabel='Count'>
```




### The same histplot but without the great populated countries

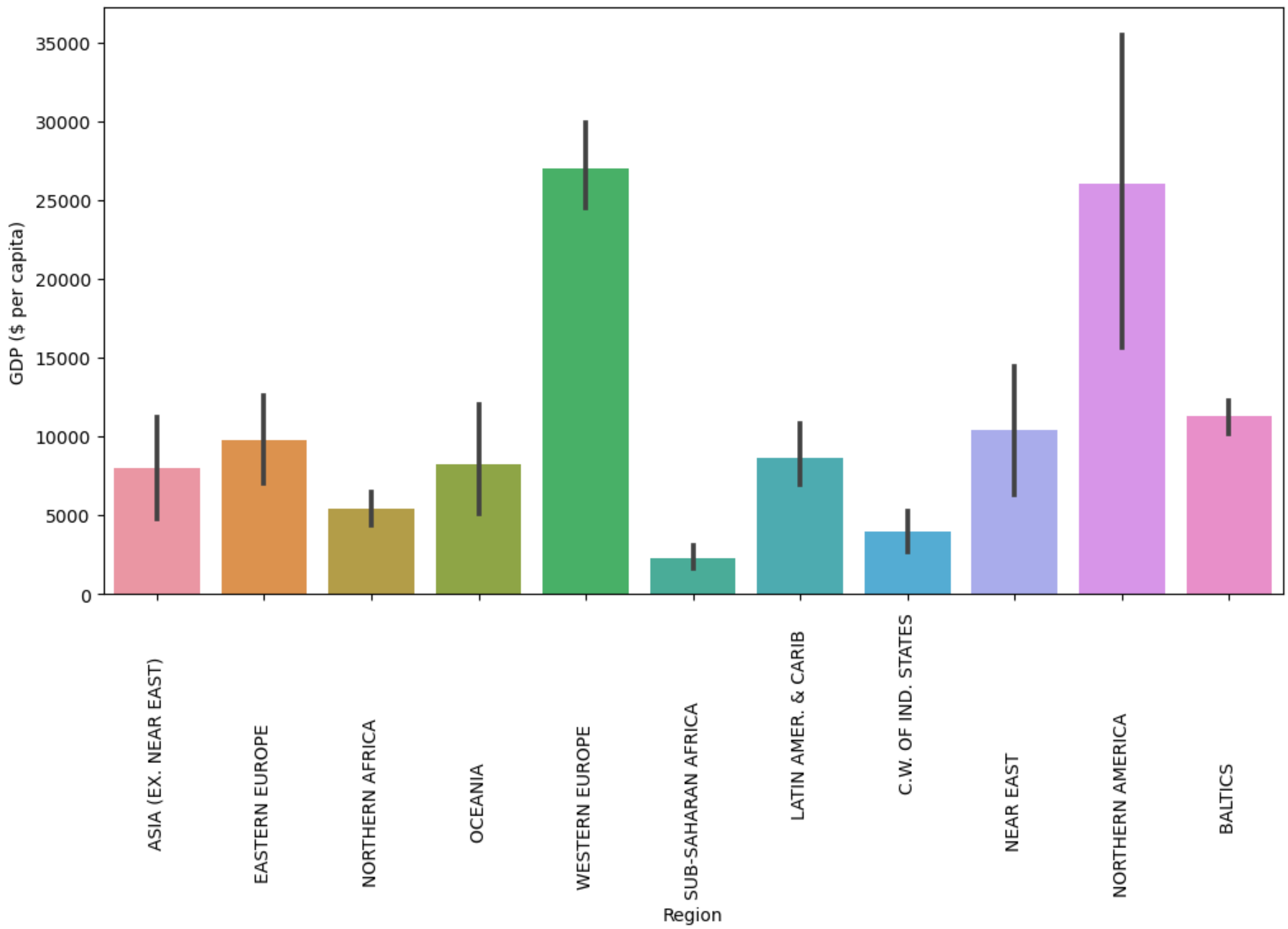
In [7]:
```
sns.histplot(data=df[df['Population']<0.5e+09], x='Population')
```

Out[7]:
```
<AxesSubplot:xlabel='Population', ylabel='Count'>
```




**Let's explore GDP and Regions. Create a bar chart showing the mean GDP per Capita per region (recall the black bar represents std).**

In [8]:
```
plt.figure(figsize=(12,6))
sns.barplot(data=df, x='Region', y='GDP ($ per capita)')
plt.xticks(rotation=90);
```

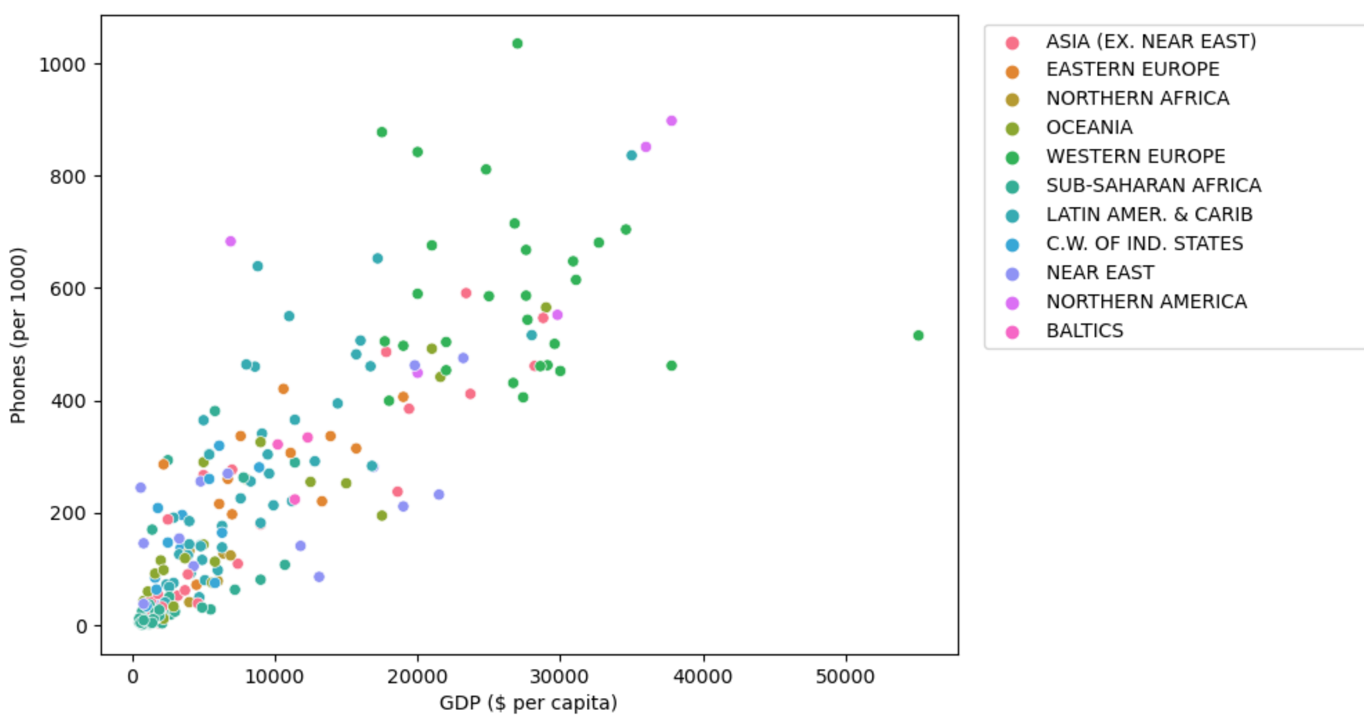



### With a scatterplot we will find the relationship between Phones per 1000 people and the GDP per Capita shown by Region. Aslo we will find any outliers.

In [9]:

```
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='GDP ($ per capita)', y='Phones (per 1000)', hue='Region')
plt.legend(bbox_to_anchor=(1.02, 1))
```

Out[9]: <matplotlib.legend.Legend at 0x21654be4100>



In [10]:
```python
df[df['GDP ($ per capita)']>50000]
```

Out[10]:

| | Country | Region | Population | Area (sq. mi.) | Pop. Density (per sq. mi.) | Coastline (coast/area ratio) | Net migration | Infant mortality (per 1000 births) | GDP ($ per capita) | Literacy (%) | Pho ( 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **121** | Luxembourg | WESTERN EUROPE | 474413 | 2586 | 183.5 | 0.0 | 8.97 | 4.81 | 55100.0 | 100.0 | 51 |

In [11]:
```python
df[df['Phones (per 1000)']>900]
```

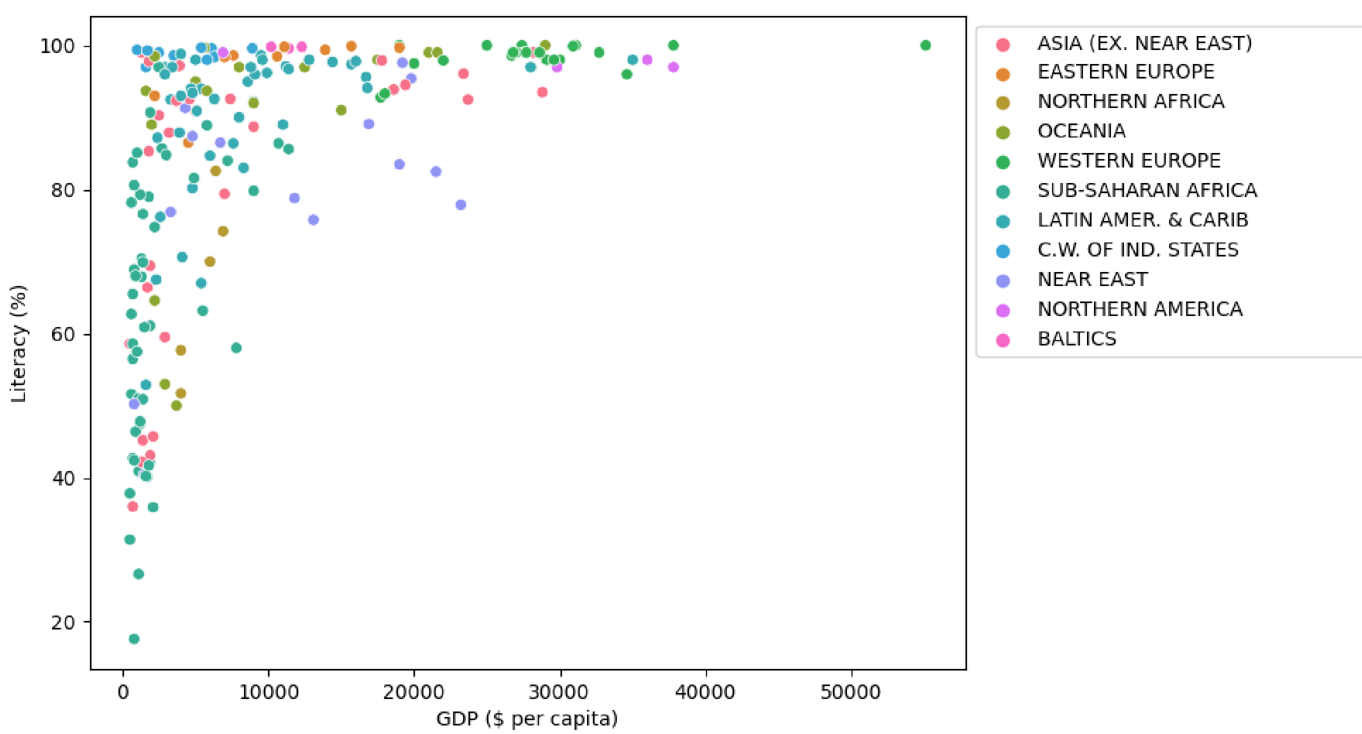
Out[11]:

| | Country | Region | Population | Area (sq. mi.) | Pop. Density (per sq. mi.) | Coastline (coast/area ratio) | Net migration | Infant mortality (per 1000 births) | GDP ($ per capita) | Literacy (%) | Phones (per 1000) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **138** | Monaco | WESTERN EUROPE | 32543 | 2 | 16271.5 | 205.0 | 7.75 | 5.43 | 27000.0 | 99.0 | 1035.6 |

## With the same plot we will find the relationship between GDP per Capita and Literacy shown by Region and find any outliers.

In [12]:
```python
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='GDP ($ per capita)', y='Literacy (%)', hue='Region')
plt.legend(bbox_to_anchor=(1, 1))
```

Out[12]: <matplotlib.legend.Legend at 0x216549cbd00>




In [13]:
```python
df[df['Literacy (%)']<35]
```

Out[13]:

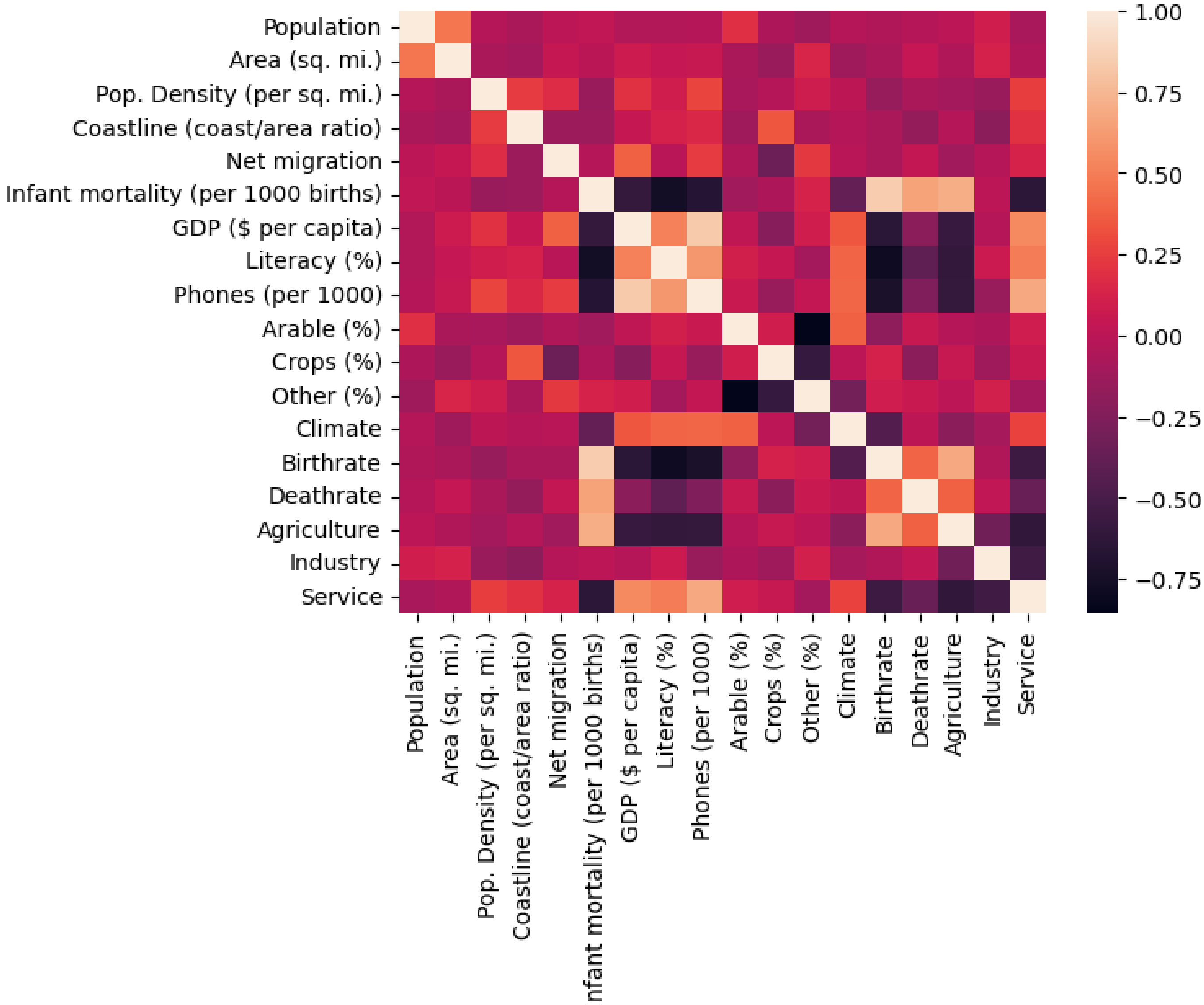
| | Country | Region | Population | Area (sq. mi.) | Pop. Density (per sq. mi.) | Coastline (coast/area ratio) | Net migration | Infant mortality (per 1000 births) | GDP ($ per capita) | Literacy (%) | Phor (p 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **31** | Burkina Faso | SUB-SAHARAN AFRICA | 13902972 | 274200 | 50.7 | 0.00 | 0.00 | 97.57 | 1100.0 | 26.6 |
| **151** | Niger | SUB-SAHARAN AFRICA | 12525094 | 1267000 | 9.9 | 0.00 | -0.67 | 121.69 | 800.0 | 17.6 |
| **183** | Sierra Leone | SUB-SAHARAN AFRICA | 6005250 | 71740 | 83.7 | 0.56 | 0.00 | 143.64 | 500.0 | 31.4 |

### We will use a Heatmap to find any Correlation between columns in the DataFrame and a clustemap for some first insights.

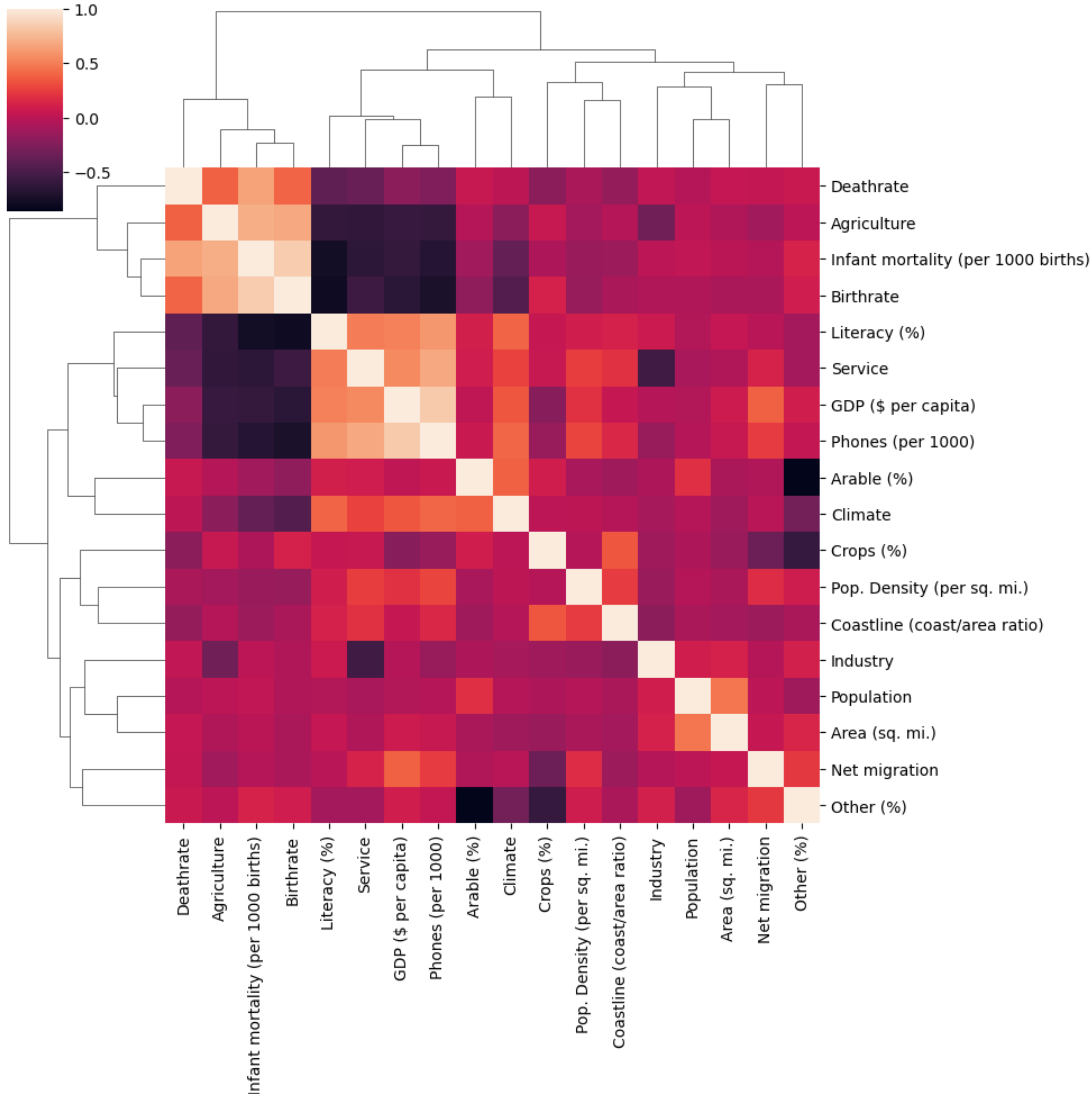
In [14]:
```
sns.heatmap(data=df.drop(['Country', 'Region'], axis=1).corr())
```

Out[14]: <AxesSubplot:>




In [15]:
```
sns.clustermap(data=df.drop(['Country', 'Region'], axis=1).corr())
```

Out[15]: <seaborn.matrix.ClusterGrid at 0x21654a56520>




---

# Data Preparation and Model Discovery

We will prepare our data for Kmeans Clustering

## Missing Data

In [16]:

```
df.isnull().sum()
```

Out[16]:

```
Country                               0
Region                                0
Population                            0
Area (sq. mi.)                        0
Pop. Density (per sq. mi.)            0
Coastline (coast/area ratio)          0
Net migration                         3
Infant mortality (per 1000 births)    3
```

```
GDP ($ per capita)                      1
Literacy (%)                           18
Phones (per 1000)                       4
Arable (%)                              2
Crops (%)                               2
Other (%)                               2
Climate                                22
Birthrate                               3
Deathrate                               4
Agriculture                            15
Industry                               16
Service                                15
dtype: int64
```

In [17]:
```python
df['Country'][df['Agriculture'].isnull()]
```

Out[17]:
```
3             American Samoa
4                    Andorra
78                 Gibraltar
80                 Greenland
83                      Guam
134                  Mayotte
140               Montserrat
144                    Nauru
153       N. Mariana Islands
171             Saint Helena
174     St Pierre & Miquelon
177               San Marino
208         Turks & Caicos Is
221         Wallis and Futuna
223           Western Sahara
Name: Country, dtype: object
```

### Filling missing values and droping insignificant.

In [18]:
```python
df[df['Agriculture'].isnull()] = df[df['Agriculture'].isnull()].fillna(value=0)
```

In [19]:
```python
df['Country'][df['Agriculture'].isnull()]
```

Out[19]:
```
Series([], Name: Country, dtype: object)
```

In [20]:
```python
df.isnull().sum()
```

Out[20]:
```
Country                                 0
Region                                  0
Population                              0
Area (sq. mi.)                          0
Pop. Density (per sq. mi.)              0
Coastline (coast/area ratio)            0
Net migration                           1
Infant mortality (per 1000 births)      1
GDP ($ per capita)                      0
Literacy (%)                           13
Phones (per 1000)                       2
Arable (%)                              1
Crops (%)                               1
Other (%)                               1
Climate                                18
Birthrate                               1
Deathrate                               2
Agriculture                             0
Industry                                1
Service                                 1
dtype: int64
```

In [21]:
```
df['Climate'] = df['Climate'].fillna(df.groupby('Region')['Climate'].transform('mean'))
```

In [22]:
```
df.isnull().sum()
```

Out[22]:
```
Country                               0
Region                                0
Population                            0
Area (sq. mi.)                        0
Pop. Density (per sq. mi.)            0
Coastline (coast/area ratio)          0
Net migration                         1
Infant mortality (per 1000 births)    1
GDP ($ per capita)                    0
Literacy (%)                         13
Phones (per 1000)                     2
Arable (%)                            1
Crops (%)                             1
Other (%)                             1
Climate                               0
Birthrate                             1
Deathrate                             2
Agriculture                           0
Industry                              1
Service                               1
dtype: int64
```

In [23]:
```
df['Literacy (%)'] = df['Literacy (%)'].fillna(df.groupby('Region')['Literacy (%)'].tran
```

In [24]:
```
df.isnull().sum()
```

Out[24]:
```
Country                               0
Region                                0
Population                            0
Area (sq. mi.)                        0
Pop. Density (per sq. mi.)            0
Coastline (coast/area ratio)          0
Net migration                         1
Infant mortality (per 1000 births)    1
GDP ($ per capita)                    0
Literacy (%)                          0
Phones (per 1000)                     2
Arable (%)                            1
Crops (%)                             1
Other (%)                             1
Climate                               0
Birthrate                             1
Deathrate                             2
Agriculture                           0
Industry                              1
Service                               1
dtype: int64
```

### I could drop or fill missing values. For simplicity, we will drop these.

In [25]:
```
df = df.dropna()
```

In [26]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 221 entries, 0 to 226
Data columns (total 20 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Country                             221 non-null    object
```

```
 1   Region                              221 non-null    object 
 2   Population                          221 non-null    int64  
 3   Area (sq. mi.)                      221 non-null    int64  
 4   Pop. Density (per sq. mi.)          221 non-null    float64
 5   Coastline (coast/area ratio)        221 non-null    float64
 6   Net migration                       221 non-null    float64
 7   Infant mortality (per 1000 births)  221 non-null    float64
 8   GDP ($ per capita)                  221 non-null    float64
 9   Literacy (%)                        221 non-null    float64
 10  Phones (per 1000)                   221 non-null    float64
 11  Arable (%)                          221 non-null    float64
 12  Crops (%)                           221 non-null    float64
 13  Other (%)                           221 non-null    float64
 14  Climate                             221 non-null    float64
 15  Birthrate                           221 non-null    float64
 16  Deathrate                           221 non-null    float64
 17  Agriculture                         221 non-null    float64
 18  Industry                            221 non-null    float64
 19  Service                             221 non-null    float64
dtypes: float64(16), int64(2), object(2)
memory usage: 36.3+ KB
```

# Data Feature Preparation

In [27]:
```
df
```

Out[27]:

| | Country | Region | Population | Area (sq. mi.) | Pop. Density (per sq. mi.) | Coastline (coast/area ratio) | Net migration | Infant mortality (per 1000 births) | GDP ($ per capita) | Literac (% |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Afghanistan | ASIA (EX. NEAR EAST) | 31056997 | 647500 | 48.0 | 0.00 | 23.06 | 163.07 | 700.0 | 36.00000 |
| **1** | Albania | EASTERN EUROPE | 3581655 | 28748 | 124.6 | 1.26 | -4.93 | 21.52 | 4500.0 | 86.50000 |
| **2** | Algeria | NORTHERN AFRICA | 32930091 | 2381740 | 13.8 | 0.04 | -0.39 | 31.00 | 6000.0 | 70.00000 |
| **3** | American Samoa | OCEANIA | 57794 | 199 | 290.4 | 58.29 | -20.71 | 9.27 | 8000.0 | 97.00000 |
| **4** | Andorra | WESTERN EUROPE | 71201 | 468 | 152.1 | 0.00 | 6.60 | 4.05 | 19000.0 | 100.00000 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | . |
| **222** | West Bank | NEAR EAST | 2460492 | 5860 | 419.9 | 0.00 | 2.98 | 19.62 | 800.0 | 79.52142 |
| **223** | Western Sahara | NORTHERN AFRICA | 273008 | 266000 | 1.0 | 0.42 | 0.00 | 0.00 | 0.0 | 0.00000 |
| **224** | Yemen | NEAR EAST | 21456188 | 527970 | 40.6 | 0.36 | 0.00 | 61.50 | 800.0 | 50.20000 |
| **225** | Zambia | SUB-SAHARAN AFRICA | 11502010 | 752614 | 15.3 | 0.00 | 0.00 | 88.29 | 800.0 | 80.60000 |
| **226** | Zimbabwe | SUB-SAHARAN AFRICA | 12236805 | 390580 | 31.3 | 0.00 | 0.00 | 67.69 | 1900.0 | 90.70000 |

221 rows × 20 columns

In [28]:
```python
X = df.drop('Country', axis=1)
```

In [29]:
```python
X = pd.get_dummies(X)
X
```

Out[29]:

| | Population | Area (sq. mi.) | Pop. Density (per sq. mi.) | Coastline (coast/area ratio) | Net migration | Infant mortality (per 1000 births) | GDP ($ per capita) | Literacy (%) | Phones (per 1000) | Arable (%) | ... | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 31056997 | 647500 | 48.0 | 0.00 | 23.06 | 163.07 | 700.0 | 36.000000 | 3.2 | 12.13 | ... | |
| **1** | 3581655 | 28748 | 124.6 | 1.26 | -4.93 | 21.52 | 4500.0 | 86.500000 | 71.2 | 21.09 | ... | |
| **2** | 32930091 | 2381740 | 13.8 | 0.04 | -0.39 | 31.00 | 6000.0 | 70.000000 | 78.1 | 3.22 | ... | |
| **3** | 57794 | 199 | 290.4 | 58.29 | -20.71 | 9.27 | 8000.0 | 97.000000 | 259.5 | 10.00 | ... | |
| **4** | 71201 | 468 | 152.1 | 0.00 | 6.60 | 4.05 | 19000.0 | 100.000000 | 497.2 | 2.22 | ... | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **222** | 2460492 | 5860 | 419.9 | 0.00 | 2.98 | 19.62 | 800.0 | 79.521429 | 145.2 | 16.90 | ... | |
| **223** | 273008 | 266000 | 1.0 | 0.42 | 0.00 | 0.00 | 0.0 | 0.000000 | 0.0 | 0.02 | ... | |
| **224** | 21456188 | 527970 | 40.6 | 0.36 | 0.00 | 61.50 | 800.0 | 50.200000 | 37.2 | 2.78 | ... | |
| **225** | 11502010 | 752614 | 15.3 | 0.00 | 0.00 | 88.29 | 800.0 | 80.600000 | 8.2 | 7.08 | ... | |
| **226** | 12236805 | 390580 | 31.3 | 0.00 | 0.00 | 67.69 | 1900.0 | 90.700000 | 26.8 | 8.32 | ... | |

221 rows × 29 columns

## Scaling

In [30]:
```python
from sklearn.preprocessing import StandardScaler
```

In [31]:
```python
scaler = StandardScaler()

scaled_X = scaler.fit_transform(X)
```

In [32]:
```python
scaled_X
```

Out[32]:
```
array([[ 0.0133285 ,  0.01855412, -0.20308668, ..., -0.31544015,
        -0.54772256, -0.36514837],
       [-0.21730118, -0.32370888, -0.14378531, ..., -0.31544015,
        -0.54772256, -0.36514837],
       [ 0.02905136,  0.97784988, -0.22956327, ..., -0.31544015,
        -0.54772256, -0.36514837],
       ...,
       [-0.06726127, -0.04756396, -0.20881553, ..., -0.31544015,
        -0.54772256, -0.36514837],
       [-0.15081724,  0.07669798, -0.22840201, ..., -0.31544015,
         1.82574186, -0.36514837],
       [-0.14464933, -0.12356132, -0.2160153 , ..., -0.31544015,
         1.82574186, -0.36514837]])
```

## Creating and Fitting Kmeans Model

In [33]:
```python
from sklearn.cluster import KMeans
```

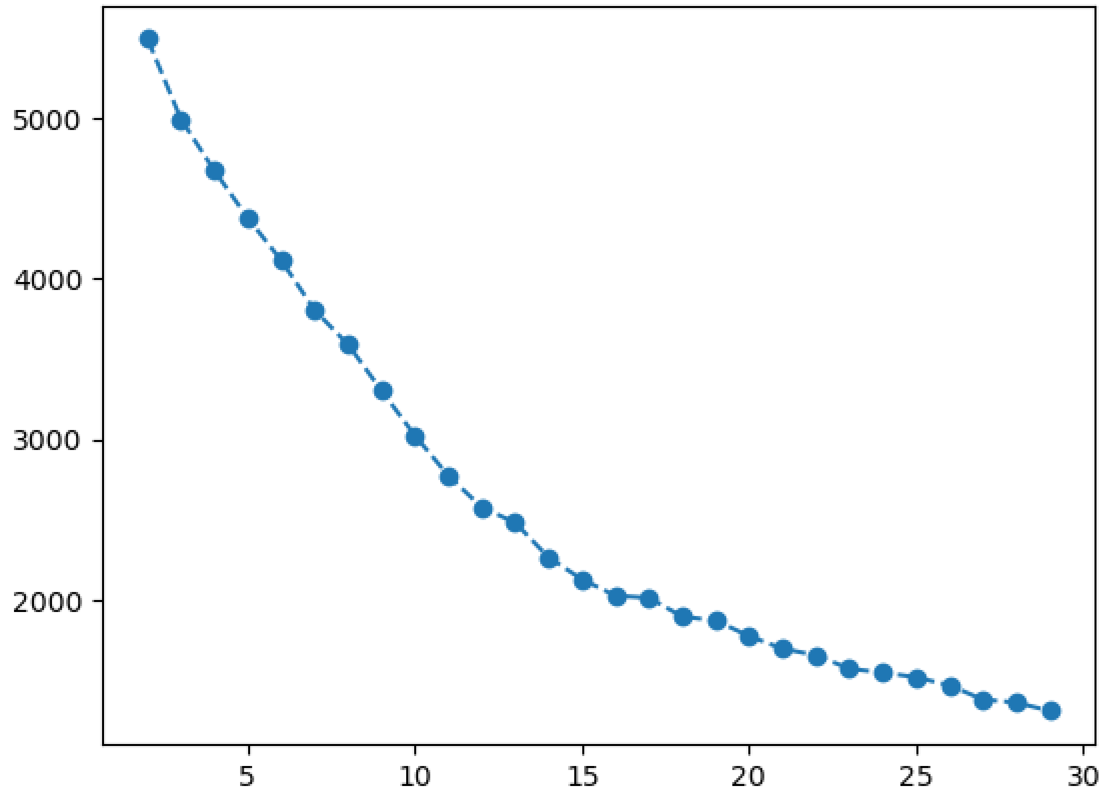
In [34]:
```
ssd = []

for k in range(2,30):
    model = KMeans(n_clusters=k)
    model.fit(scaled_X)

    ssd.append(model.inertia_)
```

In [35]:
```
ssd
```

Out[35]:
```
[5496.451168958374,
 4994.932203262997,
 4674.78039594234,
 4379.3333360302395,
 4115.59696651949,
 3804.817898644623,
 3590.751033971572,
 3303.6132345548303,
 3023.1580790576236,
 2773.0264833679976,
 2575.7584999177484,
 2480.220444234848,
 2263.270186701285,
 2126.0697720836633,
 2024.9163794204017,
 2011.4275475358675,
 1894.6577853890708,
 1869.636532092553,
 1772.816399290193,
 1693.822762101754,
 1653.7495589568998,
 1569.2256735049286,
 1545.3338939182934,
 1516.2694715715406,
 1465.1215096852623,
 1376.9669319457207,
 1358.4559420395449,
 1305.8669884925791]
```

In [36]:
```
plt.plot(range(2,30), ssd, 'o--')
```

Out[36]:
```
[<matplotlib.lines.Line2D at 0x21657653460>]
```

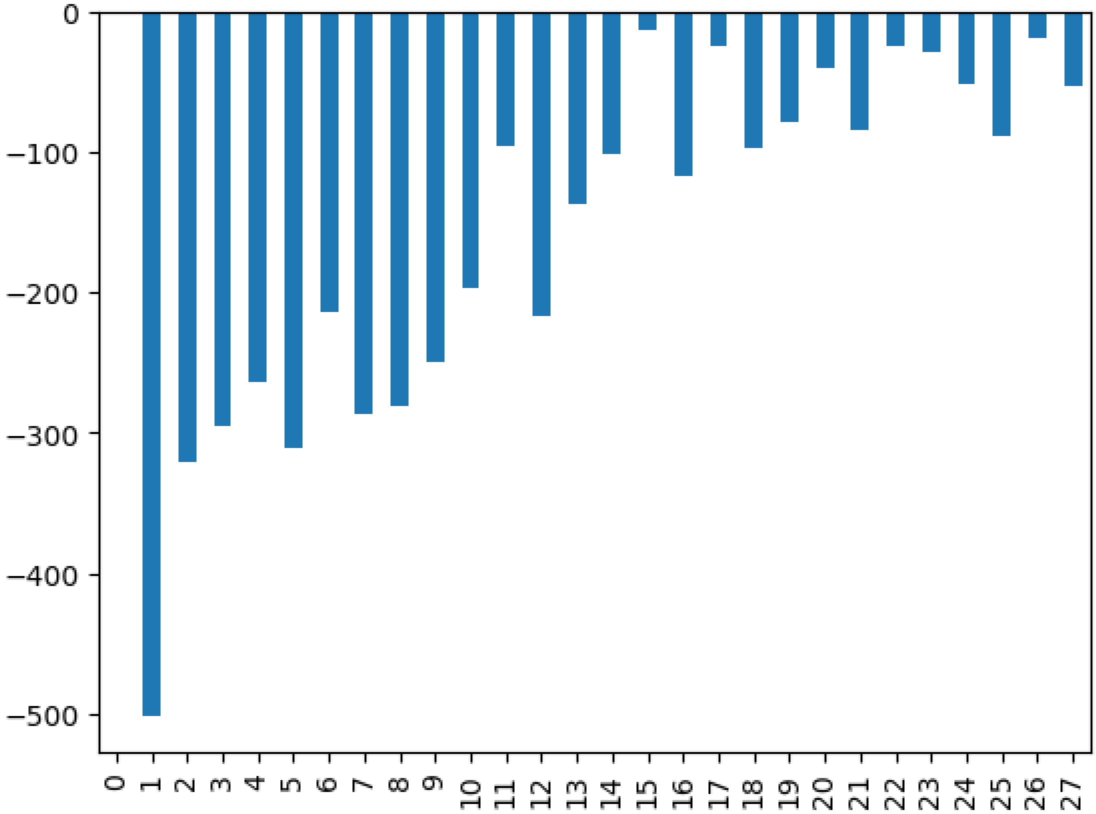



In [37]:
```
pd.Series(ssd).diff()
```

Out[37]:
```
0            NaN
1    -501.518966
2    -320.151807
3    -295.447060
4    -263.736370
5    -310.779068
6    -214.066865
7    -287.137799
8    -280.455155
9    -250.131596
10   -197.267983
11    -95.538056
12   -216.950258
13   -137.200415
14   -101.153393
15    -13.488832
16   -116.769762
17    -25.021253
18    -96.820133
19    -78.993637
20    -40.073203
21    -84.523885
22    -23.891780
23    -29.064422
24    -51.147962
25    -88.154578
26    -18.510990
27    -52.588954
dtype: float64
```

In [38]:
```
pd.Series(ssd).diff().plot(kind='bar')
```

Out[38]:
```
<AxesSubplot:>
```




---

# Model Interpretation

---

### Example Interpretation: Choosing K=3

In [39]:
```
model = KMeans(n_clusters=3)
model.fit(scaled_X)
```

Out[39]:
```
KMeans(n_clusters=3)
```

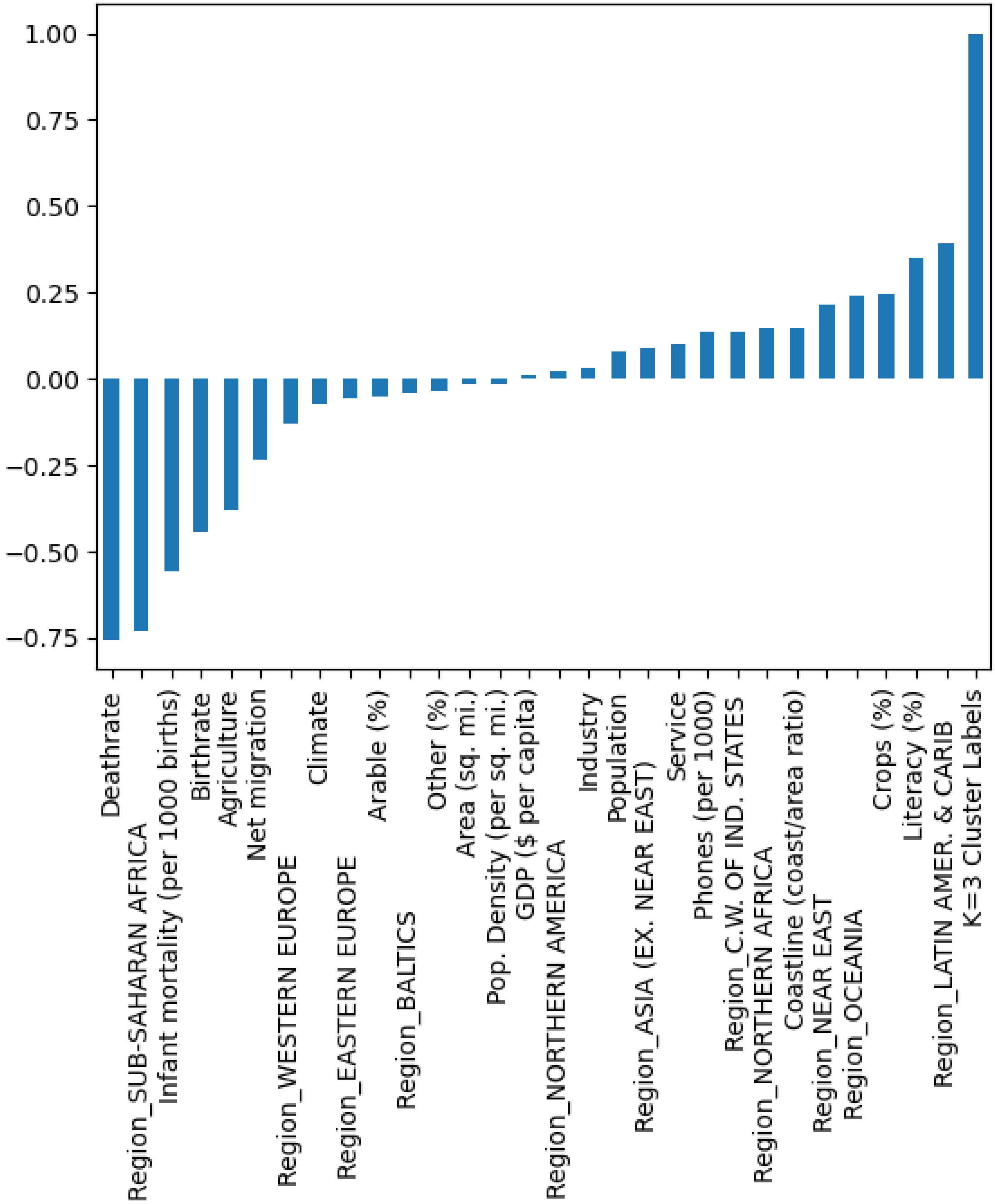
In [40]:
```
model.labels_
```

Out[40]:
```
array([0, 2, 2, 2, 1, 0, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 2, 0,
       1, 0, 2, 1, 0, 2, 1, 2, 1, 0, 2, 0, 2, 0, 1, 2, 1, 0, 0, 2, 2, 2,
       0, 0, 0, 2, 0, 1, 2, 1, 1, 0, 2, 2, 2, 2, 2, 0, 0, 1, 0, 1, 2, 1,
       1, 2, 2, 0, 0, 2, 2, 1, 0, 1, 1, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2, 1,
       1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2, 0, 2, 2, 1, 2, 2, 0,
       1, 2, 0, 0, 2, 1, 1, 1, 1, 1, 0, 0, 2, 2, 0, 1, 2, 2, 0, 1, 0, 2,
       2, 2, 2, 2, 2, 0, 0, 2, 0, 1, 2, 2, 1, 2, 0, 0, 2, 1, 2, 2, 2, 2,
       2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 0, 2, 2, 2, 2, 2, 2, 1, 2, 2, 0,
       2, 0, 1, 1, 1, 2, 0, 0, 1, 2, 0, 2, 0, 1, 1, 2, 1, 2, 0, 2, 0, 2,
       2, 2, 2, 2, 2, 2, 0, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0,
       0])
```

In [41]:
```
X ['K=3 Cluster Labels'] = model.labels_
```

In [42]:
```
X.corr()['K=3 Cluster Labels'].sort_values().plot(kind='bar')
```

Out[42]:
```
<AxesSubplot:>
```




---

## Geographical Model Interpretation

### We will use K=15 for better visualization performance

In [43]:
```
model = KMeans(n_clusters=15)
model.fit(scaled_X)
```

Out[43]:
```
KMeans(n_clusters=15)
```

In [44]:
```
model.labels_
```

Out[44]:
```
array([ 3,  8, 13,  9,  1,  3,  2,  2,  2,  5,  2,  9,  1,  5,  2, 10, 12,
        2,  5,  1,  2,  3,  7,  0,  2,  8,  3,  2,  2,  0,  8,  3,  0, 12,
        0,  3,  7,  2,  1,  3,  3,  2, 11,  2, 12,  3,  3,  2,  3,  8,  2,
        8,  1,  3,  2,  2,  2, 13,  2,  3,  3,  6,  3,  1,  9,  1,  1,  2,
        9,  3,  3, 10,  5,  1,  3, 14,  1,  7,  2,  2,  9,  2,  3,  3,  2,
       12,  2, 14,  8,  1, 11,  0,  0, 10,  1,  1, 10,  1,  2,  0,  1, 10,
```

```
        5,  3,  4,  0,  0, 10,  5,  0,  6, 10,  3,  3, 13,  1,  6,  1, 14,
        8,  3,  3,  0,  0,  3,  1,  4,  2,  3, 12, 12,  2,  4,  5,  0,  2,
       13,  3,  3,  9,  0,  1,  2,  9,  9,  2,  3,  3,  9,  1, 10,  0,  9,
        2,  9,  2,  2,  0,  8,  1,  2, 10,  2,  8,  5, 12,  2,  2,  2,  7,
        2,  4,  1,  4, 10,  3,  2,  3, 14,  8,  8,  9,  3,  3,  1,  0,  3,
        2,  3,  1,  1, 10,  0,  5,  3,  0, 12,  4,  2, 13, 10,  5,  2,  9,
       12,  5, 10,  1,  7,  2,  5,  9,  2,  0,  2,  9, 10, 13, 10,  3,  3])
```

In [45]:

```
X ['K=15 Cluster Labels'] = model.labels_
X.corr()['K=15 Cluster Labels'].sort_values()
```

Out[45]:

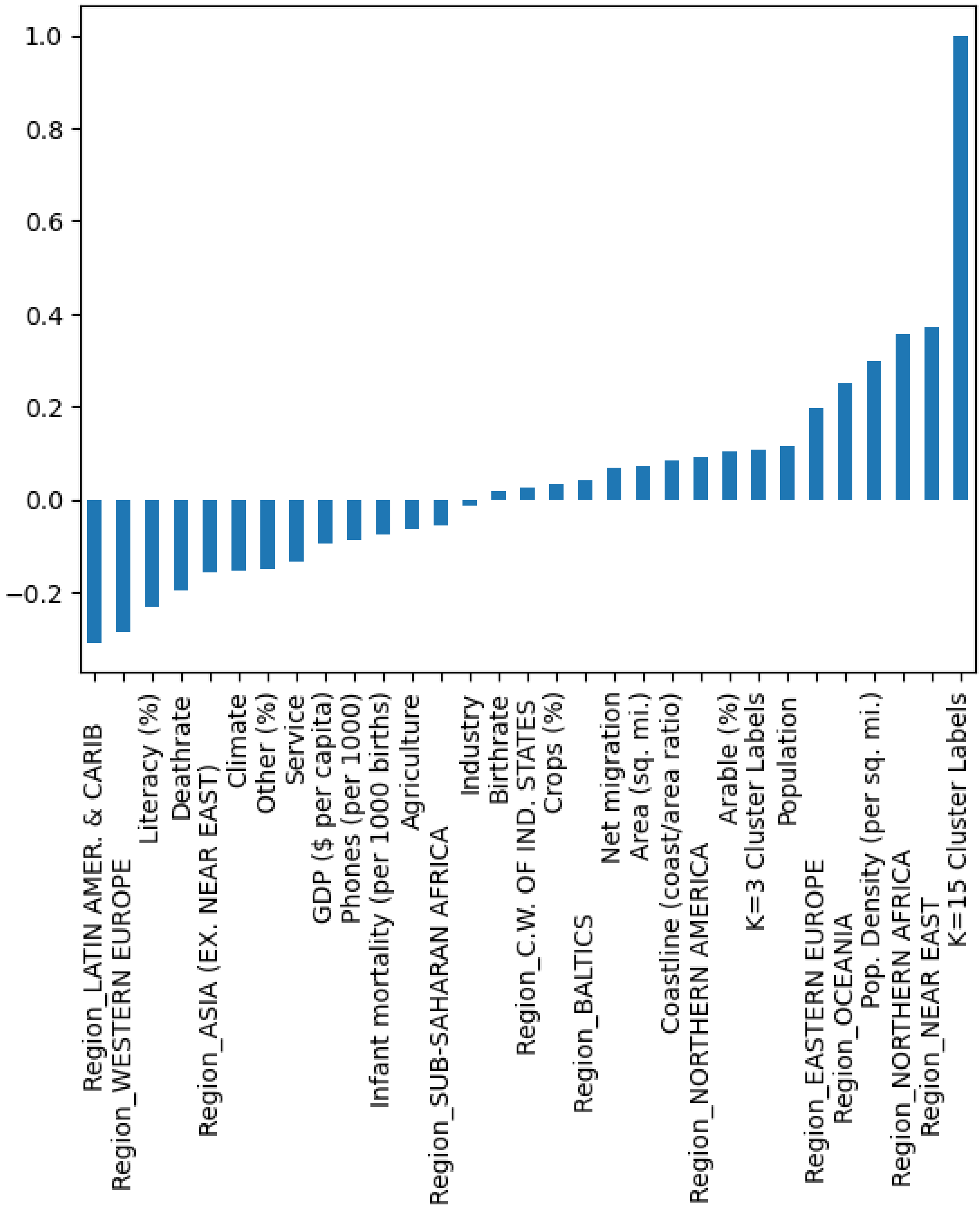
```
Region_LATIN AMER. & CARIB                 -0.306732
Region_WESTERN EUROPE                      -0.286426
Literacy (%)                               -0.231224
Deathrate                                  -0.194070
Region_ASIA (EX. NEAR EAST)                -0.157516
Climate                                    -0.152689
Other (%)                                  -0.147482
Service                                    -0.132461
GDP ($ per capita)                         -0.092500
Phones (per 1000)                          -0.088083
Infant mortality (per 1000 births)         -0.074780
Agriculture                                -0.062085
Region_SUB-SAHARAN AFRICA                  -0.057285
Industry                                   -0.010839
Birthrate                                   0.016679
Region_C.W. OF IND. STATES                  0.025061
Crops (%)                                   0.035549
Region_BALTICS                              0.042066
Net migration                               0.068755
Area (sq. mi.)                              0.073094
Coastline (coast/area ratio)                0.083887
Region_NORTHERN AMERICA                     0.093202
Arable (%)                                  0.104151
K=3 Cluster Labels                          0.106062
Population                                  0.115369
Region_EASTERN EUROPE                       0.198334
Region_OCEANIA                              0.253325
Pop. Density (per sq. mi.)                  0.297359
Region_NORTHERN AFRICA                      0.356924
Region_NEAR EAST                            0.370923
K=15 Cluster Labels                         1.000000
Name: K=15 Cluster Labels, dtype: float64
```

In [46]:

```
X.corr()['K=15 Cluster Labels'].sort_values().plot(kind='bar')
```

Out[46]:

```
<AxesSubplot:>
```




In [47]:
```
!pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\user\anaconda3\lib\site-packages (5.9.
0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\user\anaconda3\lib\site-packa
ges (from plotly) (8.0.1)
```

In [48]:
```
iso_codes = pd.read_csv('country_iso_codes.csv')
```

In [49]:
```
iso_codes
```

Out[49]:

| | Country | ISO Code |
|---|---|---|
| **0** | Afghanistan | AFG |
| **1** | Akrotiri and Dhekelia – See United Kingdom, The | Akrotiri and Dhekelia – See United Kingdom, The |
| **2** | Åland Islands | ALA |
| **3** | Albania | ALB |
| **4** | Algeria | DZA |

| | | |
|---|---|---|
| ... | ... | ... |
| **296** | Congo, Dem. Rep. | COD |
| **297** | Congo, Repub. of the | COG |
| **298** | Tanzania | TZA |
| **299** | Central African Rep. | CAF |
| **300** | Cote d'Ivoire | CIV |

301 rows × 2 columns

In [50]:
```
iso_codes.set_index('Country')['ISO Code'].to_dict()
```

Out[50]:
```
{'Afghanistan': 'AFG',
 'Akrotiri and Dhekelia – See United Kingdom, The': 'Akrotiri and Dhekelia – See United Kingdom, The',
 'Åland Islands': 'ALA',
 'Albania': 'ALB',
 'Algeria': 'DZA',
 'American Samoa': 'ASM',
 'Andorra': 'AND',
 'Angola': 'AGO',
 'Anguilla': 'AIA',
 'Antarctica\u200a[a]': 'ATA',
 'Antigua and Barbuda': 'ATG',
 'Argentina': 'ARG',
 'Armenia': 'ARM',
 'Aruba': 'ABW',
 'Ashmore and Cartier Islands – See Australia.': 'Ashmore and Cartier Islands – See Australia.',
 'Australia\u200a[b]': 'AUS',
 'Austria': 'AUT',
 'Azerbaijan': 'AZE',
 'Bahamas (the)': 'BHS',
 'Bahrain': 'BHR',
 'Bangladesh': 'BGD',
 'Barbados': 'BRB',
 'Belarus': 'BLR',
 'Belgium': 'BEL',
 'Belize': 'BLZ',
 'Benin': 'BEN',
 'Bermuda': 'BMU',
 'Bhutan': 'BTN',
 'Bolivia (Plurinational State of)': 'BOL',
 'Bonaire\xa0Sint Eustatius\xa0Saba': 'BES',
 'Bosnia and Herzegovina': 'BIH',
 'Botswana': 'BWA',
 'Bouvet Island': 'BVT',
 'Brazil': 'BRA',
 'British Indian Ocean Territory (the)': 'IOT',
 'British Virgin Islands – See Virgin Islands (British).': 'British Virgin Islands – See Virgin Islands (British).',
 'Brunei Darussalam\u200a[e]': 'BRN',
 'Bulgaria': 'BGR',
 'Burkina Faso': 'BFA',
 'Burma – See Myanmar.': 'Burma – See Myanmar.',
 'Burundi': 'BDI',
 'Cabo Verde\u200a[f]': 'CPV',
 'Cambodia': 'KHM',
 'Cameroon': 'CMR',
 'Canada': 'CAN',
 'Cape Verde – See Cabo Verde.': 'Cape Verde – See Cabo Verde.',
 'Caribbean Netherlands – See Bonaire, Sint Eustatius and Saba.': 'Caribbean Netherlands
```

```
- See Bonaire, Sint Eustatius and Saba.',
 'Cayman Islands (the)': 'CYM',
 'Central African Republic (the)': 'CAF',
 'Chad': 'TCD',
 'Chile': 'CHL',
 'China': 'CHN',
 'China, The Republic of - See Taiwan (Province of China).': 'China, The Republic of - S
ee Taiwan (Province of China).',
 'Christmas Island': 'CXR',
 'Clipperton Island - See France.': 'Clipperton Island - See France.',
 'Cocos (Keeling) Islands (the)': 'CCK',
 'Colombia': 'COL',
 'Comoros (the)': 'COM',
 'Congo (the Democratic Republic of the)': 'COD',
 'Congo (the)\u200a[g]': 'COG',
 'Cook Islands (the)': 'COK',
 'Coral Sea Islands - See Australia.': 'Coral Sea Islands - See Australia.',
 'Costa Rica': 'CRI',
 "Côte d'Ivoire\u200a[h]": 'CIV',
 'Croatia': 'HRV',
 'Cuba': 'CUB',
 'Curaçao': 'CUW',
 'Cyprus': 'CYP',
 'Czechia\u200a[i]': 'CZE',
 "Democratic People's Republic of Korea - See Korea, The Democratic People's Republic o
f.": "Democratic People's Republic of Korea - See Korea, The Democratic People's Republi
c of.",
 'Democratic Republic of the Congo - See Congo, The Democratic Republic of the.': 'Democ
ratic Republic of the Congo - See Congo, The Democratic Republic of the.',
 'Denmark': 'DNK',
 'Djibouti': 'DJI',
 'Dominica': 'DMA',
 'Dominican Republic (the)': 'DOM',
 'East Timor - See Timor-Leste.': 'East Timor - See Timor-Leste.',
 'Ecuador': 'ECU',
 'Egypt': 'EGY',
 'El Salvador': 'SLV',
 'England - See United Kingdom, The.': 'England - See United Kingdom, The.',
 'Equatorial Guinea': 'GNQ',
 'Eritrea': 'ERI',
 'Estonia': 'EST',
 'Eswatini\u200a[j]': 'SWZ',
 'Ethiopia': 'ETH',
 'Falkland Islands (the) [Malvinas]\u200a[k]': 'FLK',
 'Faroe Islands (the)': 'FRO',
 'Fiji': 'FJI',
 'Finland': 'FIN',
 'France\u200a[l]': 'FRA',
 'French Guiana': 'GUF',
 'French Polynesia': 'PYF',
 'French Southern Territories (the)\u200a[m]': 'ATF',
 'Gabon': 'GAB',
 'Gambia (the)': 'GMB',
 'Georgia': 'GEO',
 'Germany': 'DEU',
 'Ghana': 'GHA',
 'Gibraltar': 'GIB',
 'Great Britain - See United Kingdom, The.': 'Great Britain - See United Kingdom, The.',
 'Greece': 'GRC',
 'Greenland': 'GRL',
 'Grenada': 'GRD',
 'Guadeloupe': 'GLP',
 'Guam': 'GUM',
 'Guatemala': 'GTM',
 'Guernsey': 'GGY',
 'Guinea': 'GIN',
```

'Guinea-Bissau': 'GNB',
'Guyana': 'GUY',
'Haiti': 'HTI',
'Hawaiian Islands – See United States of America, The.': 'Hawaiian Islands – See United States of America, The.',
'Heard Island and McDonald Islands': 'HMD',
'Holy See (the)\u200a[n]': 'VAT',
'Honduras': 'HND',
'Hong Kong': 'HKG',
'Hungary': 'HUN',
'Iceland': 'ISL',
'India': 'IND',
'Indonesia': 'IDN',
'Iran (Islamic Republic of)': 'IRN',
'Iraq': 'IRQ',
'Ireland': 'IRL',
'Isle of Man': 'IMN',
'Israel': 'ISR',
'Italy': 'ITA',
"Ivory Coast – See Côte d'Ivoire.": "Ivory Coast – See Côte d'Ivoire.",
'Jamaica': 'JAM',
'Jan Mayen – See Svalbard and Jan Mayen.': 'Jan Mayen – See Svalbard and Jan Mayen.',
'Japan': 'JPN',
'Jersey': 'JEY',
'Jordan': 'JOR',
'Kazakhstan': 'KAZ',
'Kenya': 'KEN',
'Kiribati': 'KIR',
"Korea (the Democratic People's Republic of)\u200a[o]": 'PRK',
'Korea (the Republic of)\u200a[p]': 'KOR',
'Kuwait': 'KWT',
'Kyrgyzstan': 'KGZ',
"Lao People's Democratic Republic (the)\u200a[q]": 'LAO',
'Latvia': 'LVA',
'Lebanon': 'LBN',
'Lesotho': 'LSO',
'Liberia': 'LBR',
'Libya': 'LBY',
'Liechtenstein': 'LIE',
'Lithuania': 'LTU',
'Luxembourg': 'LUX',
'Macao\u200a[r]': 'MAC',
'North Macedonia\u200a[s]': 'MKD',
'Madagascar': 'MDG',
'Malawi': 'MWI',
'Malaysia': 'MYS',
'Maldives': 'MDV',
'Mali': 'MLI',
'Malta': 'MLT',
'Marshall Islands (the)': 'MHL',
'Martinique': 'MTQ',
'Mauritania': 'MRT',
'Mauritius': 'MUS',
'Mayotte': 'MYT',
'Mexico': 'MEX',
'Micronesia (Federated States of)': 'FSM',
'Moldova (the Republic of)': 'MDA',
'Monaco': 'MCO',
'Mongolia': 'MNG',
'Montenegro': 'MNE',
'Montserrat': 'MSR',
'Morocco': 'MAR',
'Mozambique': 'MOZ',
'Myanmar\u200a[t]': 'MMR',
'Namibia': 'NAM',
'Nauru': 'NRU',

```
 'Nepal': 'NPL',
 'Netherlands (the)': 'NLD',
 'New Caledonia': 'NCL',
 'New Zealand': 'NZL',
 'Nicaragua': 'NIC',
 'Niger (the)': 'NER',
 'Nigeria': 'NGA',
 'Niue': 'NIU',
 'Norfolk Island': 'NFK',
 "North Korea - See Korea, The Democratic People's Republic of.": "North Korea - See Kor
ea, The Democratic People's Republic of.",
 'Northern Ireland - See United Kingdom, The.': 'Northern Ireland - See United Kingdom,
The.',
 'Northern Mariana Islands (the)': 'MNP',
 'Norway': 'NOR',
 'Oman': 'OMN',
 'Pakistan': 'PAK',
 'Palau': 'PLW',
 'Palestine, State of': 'PSE',
 'Panama': 'PAN',
 'Papua New Guinea': 'PNG',
 'Paraguay': 'PRY',
 "People's Republic of China - See China.": "People's Republic of China - See China.",
 'Peru': 'PER',
 'Philippines (the)': 'PHL',
 'Pitcairn\u200a[u]': 'PCN',
 'Poland': 'POL',
 'Portugal': 'PRT',
 'Puerto Rico': 'PRI',
 'Qatar': 'QAT',
 'Republic of China - See Taiwan (Province of China).': 'Republic of China - See Taiwan
(Province of China).',
 'Republic of Korea - See Korea, The Republic of.': 'Republic of Korea - See Korea, The
Republic of.',
 'Republic of the Congo - See Congo, The.': 'Republic of the Congo - See Congo, The.',
 'Réunion': 'REU',
 'Romania': 'ROU',
 'Russian Federation (the)\u200a[v]': 'RUS',
 'Rwanda': 'RWA',
 'Saba - See Bonaire, Sint Eustatius and Saba.': 'Saba - See Bonaire, Sint Eustatius and
Saba.',
 'Sahrawi Arab Democratic Republic - See Western Sahara.': 'Sahrawi Arab Democratic Repu
blic - See Western Sahara.',
 'Saint Barthélemy': 'BLM',
 'Saint Helena\xa0Ascension Island\xa0Tristan da Cunha': 'SHN',
 'Saint Kitts and Nevis': 'KNA',
 'Saint Lucia': 'LCA',
 'Saint Martin (French part)': 'MAF',
 'Saint Pierre and Miquelon': 'SPM',
 'Saint Vincent and the Grenadines': 'VCT',
 'Samoa': 'WSM',
 'San Marino': 'SMR',
 'Sao Tome and Principe': 'STP',
 'Saudi Arabia': 'SAU',
 'Scotland - See United Kingdom, The.': 'Scotland - See United Kingdom, The.',
 'Senegal': 'SEN',
 'Serbia': 'SRB',
 'Seychelles': 'SYC',
 'Sierra Leone': 'SLE',
 'Singapore': 'SGP',
 'Sint Eustatius - See Bonaire, Sint Eustatius and Saba.': 'Sint Eustatius - See Bonair
e, Sint Eustatius and Saba.',
 'Sint Maarten (Dutch part)': 'SXM',
 'Slovakia': 'SVK',
 'Slovenia': 'SVN',
 'Solomon Islands': 'SLB',
```

```
 'Somalia': 'SOM',
 'South Africa': 'ZAF',
 'South Georgia and the South Sandwich Islands': 'SGS',
 'South Korea – See Korea, The Republic of.': 'South Korea – See Korea, The Republic o
f.',
 'South Sudan': 'SSD',
 'Spain': 'ESP',
 'Sri Lanka': 'LKA',
 'Sudan (the)': 'SDN',
 'Suriname': 'SUR',
 'Svalbard\xa0Jan Mayen': 'SJM',
 'Sweden': 'SWE',
 'Switzerland': 'CHE',
 'Syrian Arab Republic (the)\u200a[x]': 'SYR',
 'Taiwan (Province of China)\u200a[y]': 'TWN',
 'Tajikistan': 'TJK',
 'Tanzania, the United Republic of': 'TZA',
 'Thailand': 'THA',
 'Timor-Leste\u200a[aa]': 'TLS',
 'Togo': 'TGO',
 'Tokelau': 'TKL',
 'Tonga': 'TON',
 'Trinidad and Tobago': 'TTO',
 'Tunisia': 'TUN',
 'Turkey': 'TUR',
 'Turkmenistan': 'TKM',
 'Turks and Caicos Islands (the)': 'TCA',
 'Tuvalu': 'TUV',
 'Uganda': 'UGA',
 'Ukraine': 'UKR',
 'United Arab Emirates (the)': 'ARE',
 'United Kingdom of Great Britain and Northern Ireland (the)': 'GBR',
 'United States Minor Outlying Islands (the)\u200a[ac]': 'UMI',
 'United States of America (the)': 'USA',
 'United States Virgin Islands – See Virgin Islands (U.S.).': 'United States Virgin Isla
nds – See Virgin Islands (U.S.).',
 'Uruguay': 'URY',
 'Uzbekistan': 'UZB',
 'Vanuatu': 'VUT',
 'Vatican City – See Holy See, The.': 'Vatican City – See Holy See, The.',
 'Venezuela (Bolivarian Republic of)': 'VEN',
 'Viet Nam\u200a[ae]': 'VNM',
 'Virgin Islands (British)\u200a[af]': 'VGB',
 'Virgin Islands (U.S.)\u200a[ag]': 'VIR',
 'Wales – See United Kingdom, The.': 'Wales – See United Kingdom, The.',
 'Wallis and Futuna': 'WLF',
 'Western Sahara\u200a[ah]': 'ESH',
 'Yemen': 'YEM',
 'Zambia': 'ZMB',
 'Zimbabwe': 'ZWE',
 'United States': 'USA',
 'United Kingdom': 'GBR',
 'Venezuela': 'VEN',
 'Australia': 'AUS',
 'Iran': 'IRN',
 'France': 'FRA',
 'Russia': 'RUS',
 'Korea, North': 'PRK',
 'Korea, South': 'KOR',
 'Myanmar': 'MMR',
 'Burma': 'MMR',
 'Vietnam': 'VNM',
 'Laos': 'LAO',
 'Bolivia': 'BOL',
 'Niger': 'NER',
 'Sudan': 'SDN',
```

```
 'Congo, Dem. Rep.': 'COD',
 'Congo, Repub. of the': 'COG',
 'Tanzania': 'TZA',
 'Central African Rep.': 'CAF',
 "Cote d'Ivoire": 'CIV'}
```

In [51]:
```
iso_map = iso_codes.set_index('Country')['ISO Code'].to_dict()
```

In [52]:
```
df['ISO Code'] = df['Country'].map(iso_map)
```

In [53]:
```
df
```

Out[53]:

| | Country | Region | Population | Area (sq. mi.) | Pop. Density (per sq. mi.) | Coastline (coast/area ratio) | Net migration | Infant mortality (per 1000 births) | GDP ($ per capita) | Literac (% |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Afghanistan | ASIA (EX. NEAR EAST) | 31056997 | 647500 | 48.0 | 0.00 | 23.06 | 163.07 | 700.0 | 36.00000 |
| **1** | Albania | EASTERN EUROPE | 3581655 | 28748 | 124.6 | 1.26 | -4.93 | 21.52 | 4500.0 | 86.50000 |
| **2** | Algeria | NORTHERN AFRICA | 32930091 | 2381740 | 13.8 | 0.04 | -0.39 | 31.00 | 6000.0 | 70.00000 |
| **3** | American Samoa | OCEANIA | 57794 | 199 | 290.4 | 58.29 | -20.71 | 9.27 | 8000.0 | 97.00000 |
| **4** | Andorra | WESTERN EUROPE | 71201 | 468 | 152.1 | 0.00 | 6.60 | 4.05 | 19000.0 | 100.00000 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | . |
| **222** | West Bank | NEAR EAST | 2460492 | 5860 | 419.9 | 0.00 | 2.98 | 19.62 | 800.0 | 79.52142 |
| **223** | Western Sahara | NORTHERN AFRICA | 273008 | 266000 | 1.0 | 0.42 | 0.00 | 0.00 | 0.0 | 0.00000 |
| **224** | Yemen | NEAR EAST | 21456188 | 527970 | 40.6 | 0.36 | 0.00 | 61.50 | 800.0 | 50.20000 |
| **225** | Zambia | SUB-SAHARAN AFRICA | 11502010 | 752614 | 15.3 | 0.00 | 0.00 | 88.29 | 800.0 | 80.60000 |
| **226** | Zimbabwe | SUB-SAHARAN AFRICA | 12236805 | 390580 | 31.3 | 0.00 | 0.00 | 67.69 | 1900.0 | 90.70000 |

221 rows × 21 columns

In [54]:
```
df['Cluster'] = model.labels_
```

In [55]:
```
df
```

Out[55]:

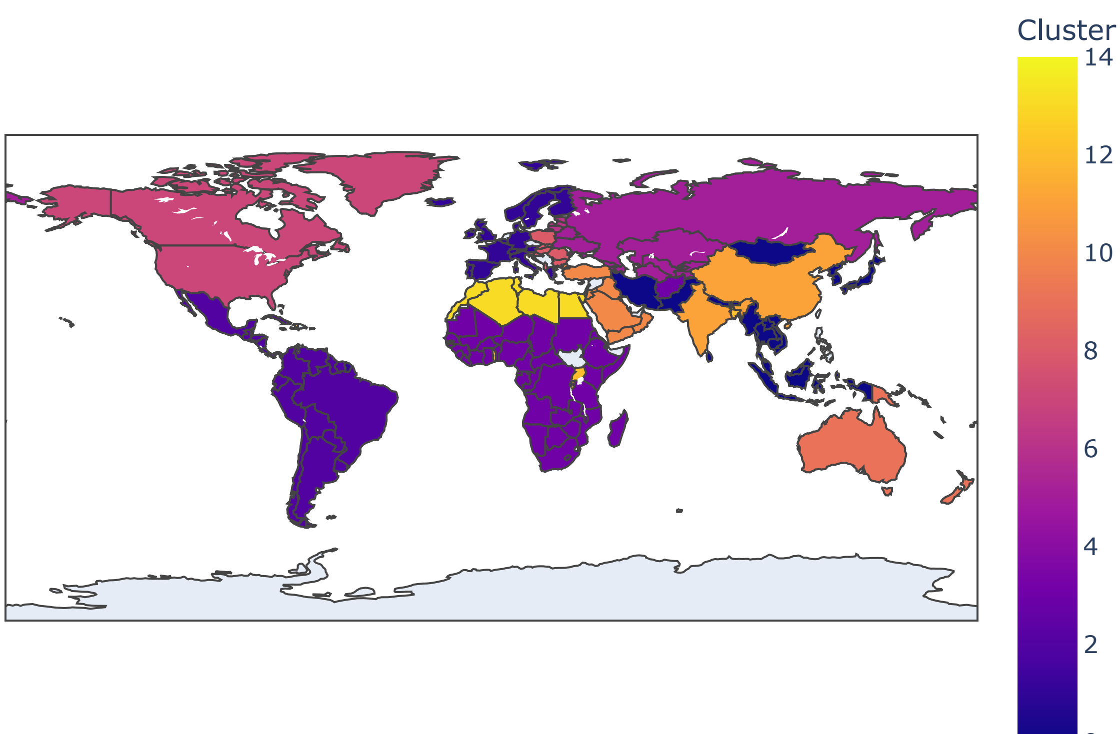
| | Country | Region | Population | Area (sq. mi.) | Pop. Density (per sq. mi.) | Coastline (coast/area ratio) | Net migration | Infant mortality (per 1000 births) | GDP ($ per capita) | Literac (% |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Afghanistan | ASIA (EX. NEAR | 31056997 | 647500 | 48.0 | 0.00 | 23.06 | 163.07 | 700.0 | 36.00000 |

EAST)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | Albania | EASTERN EUROPE | 3581655 | 28748 | 124.6 | 1.26 | -4.93 | 21.52 | 4500.0 | 86.50000 |
| **2** | Algeria | NORTHERN AFRICA | 32930091 | 2381740 | 13.8 | 0.04 | -0.39 | 31.00 | 6000.0 | 70.00000 |
| **3** | American Samoa | OCEANIA | 57794 | 199 | 290.4 | 58.29 | -20.71 | 9.27 | 8000.0 | 97.00000 |
| **4** | Andorra | WESTERN EUROPE | 71201 | 468 | 152.1 | 0.00 | 6.60 | 4.05 | 19000.0 | 100.00000 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **222** | West Bank | NEAR EAST | 2460492 | 5860 | 419.9 | 0.00 | 2.98 | 19.62 | 800.0 | 79.52142 |
| **223** | Western Sahara | NORTHERN AFRICA | 273008 | 266000 | 1.0 | 0.42 | 0.00 | 0.00 | 0.0 | 0.00000 |
| **224** | Yemen | NEAR EAST | 21456188 | 527970 | 40.6 | 0.36 | 0.00 | 61.50 | 800.0 | 50.20000 |
| **225** | Zambia | SUB-SAHARAN AFRICA | 11502010 | 752614 | 15.3 | 0.00 | 0.00 | 88.29 | 800.0 | 80.60000 |
| **226** | Zimbabwe | SUB-SAHARAN AFRICA | 12236805 | 390580 | 31.3 | 0.00 | 0.00 | 67.69 | 1900.0 | 90.70000 |

221 rows × 22 columns

In [56]:
```
import plotly.express as px

fig = px.choropleth(df, locations="ISO Code",color="Cluster", hover_name="Country")
fig.show()
```



In [ ]: