In [97]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [98]:
```
df = pd.read_csv('CIA_Country_Facts.csv')
```

In [99]:
```
df.head(50)
```

Out[99]:

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

In [100]:
```
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 [101]:
```
df.shape
```

Out[101]: (227, 20)

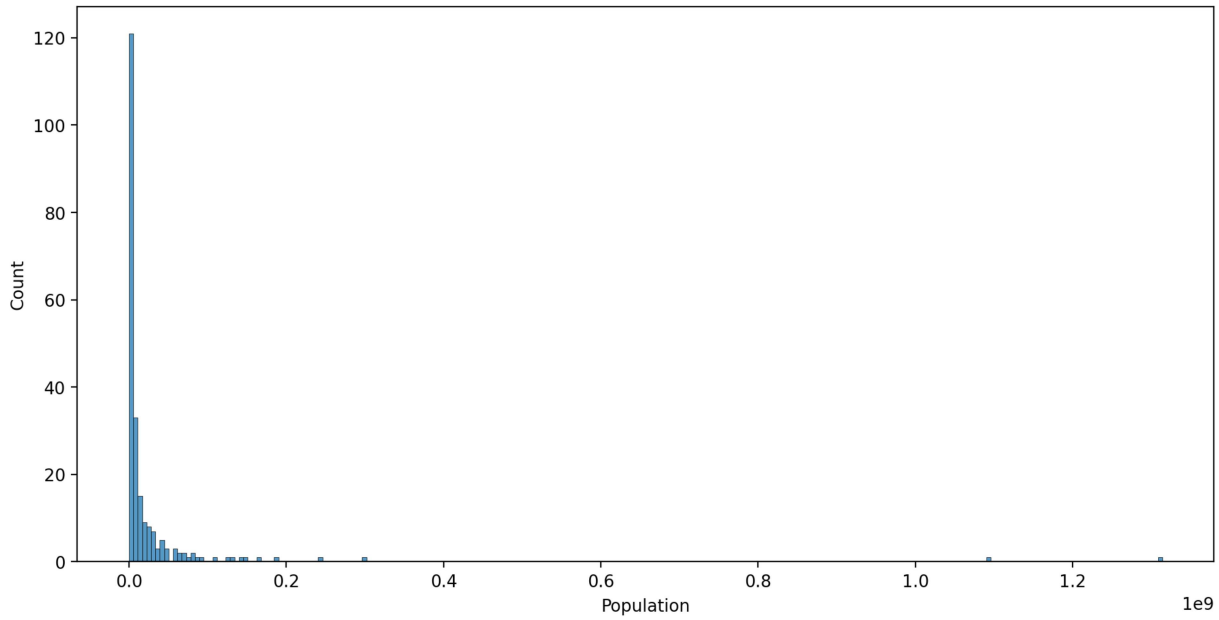
In [102]:
```
df.describe()
```

Out[102]:

	Population	Area (sq. mi.)	Pop. Density (per sq. mi.)	Coastline (coast/area ratio)	Net migration	Infant mortality (per 1000 births)	GDP ($ pe capita
count	2.270000e+02	2.270000e+02	227.000000	227.000000	224.000000	224.000000	226.00000
mean	2.874028e+07	5.982270e+05	379.047137	21.165330	0.038125	35.506964	9689.82300
std	1.178913e+08	1.790282e+06	1660.185825	72.286863	4.889269	35.389899	10049.13851
min	7.026000e+03	2.000000e+00	0.000000	0.000000	-20.990000	2.290000	500.00000
25%	4.376240e+05	4.647500e+03	29.150000	0.100000	-0.927500	8.150000	1900.00000
50%	4.786994e+06	8.660000e+04	78.800000	0.730000	0.000000	21.000000	5550.00000
75%	1.749777e+07	4.418110e+05	190.150000	10.345000	0.997500	55.705000	15700.00000
max	1.313974e+09	1.707520e+07	16271.500000	870.660000	23.060000	191.190000	55100.00000

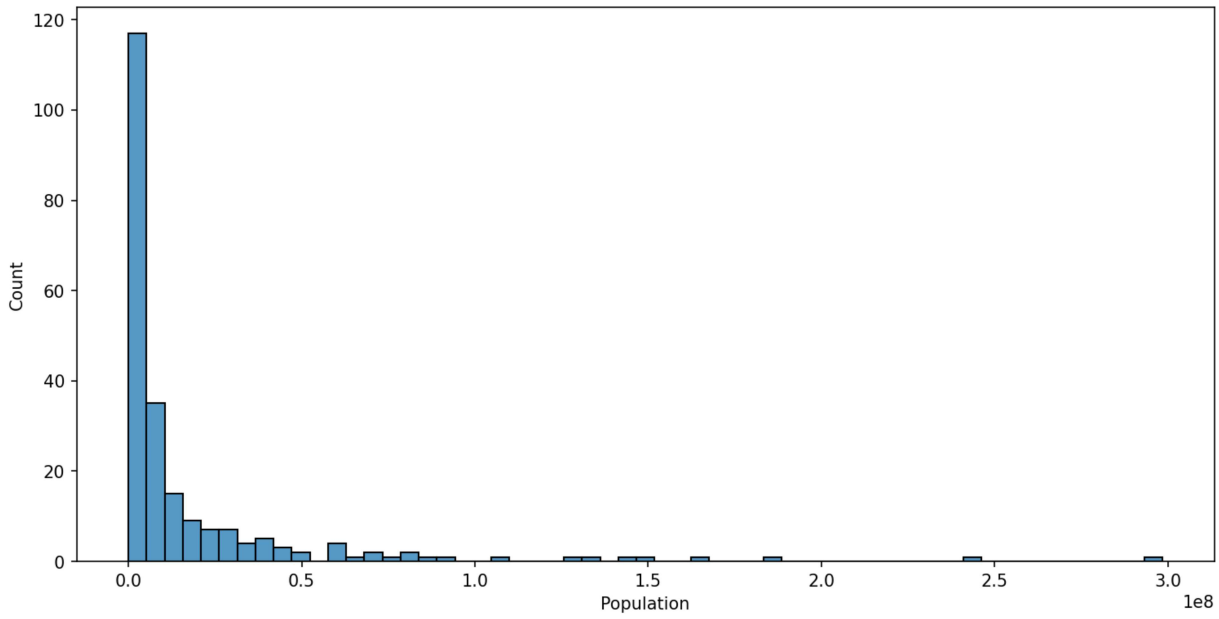
In [103]:
```
plt.figure(figsize=(12,6),dpi=200)
sns.histplot(data=df,x='Population')
```

Out[103]: <AxesSubplot:xlabel='Population', ylabel='Count'>

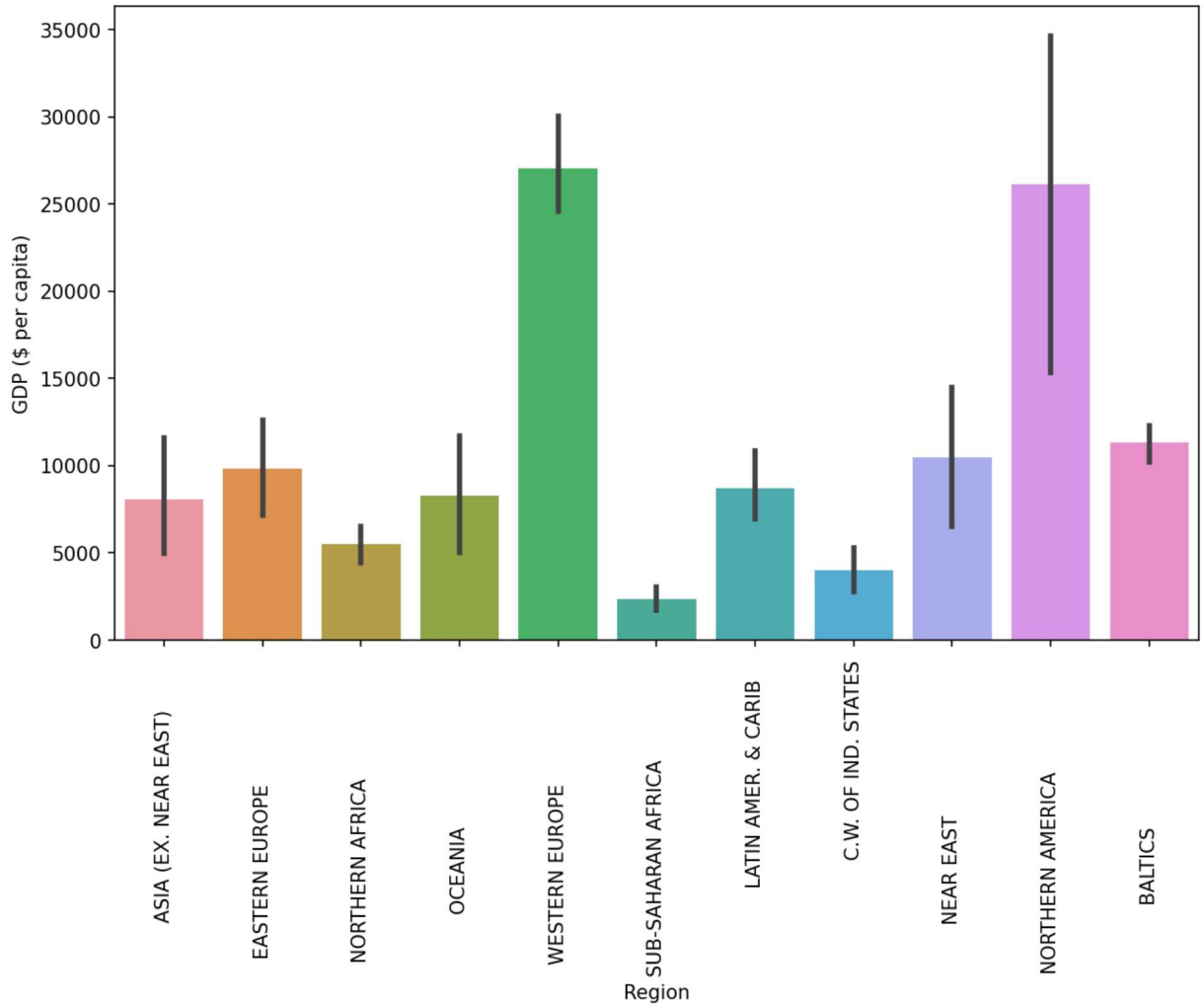


In [104]:
```
plt.figure(figsize=(12,6),dpi=150)
sns.histplot(data=df[df['Population']<500000000],x='Population')
```

Out[104]: <AxesSubplot:xlabel='Population', ylabel='Count'>

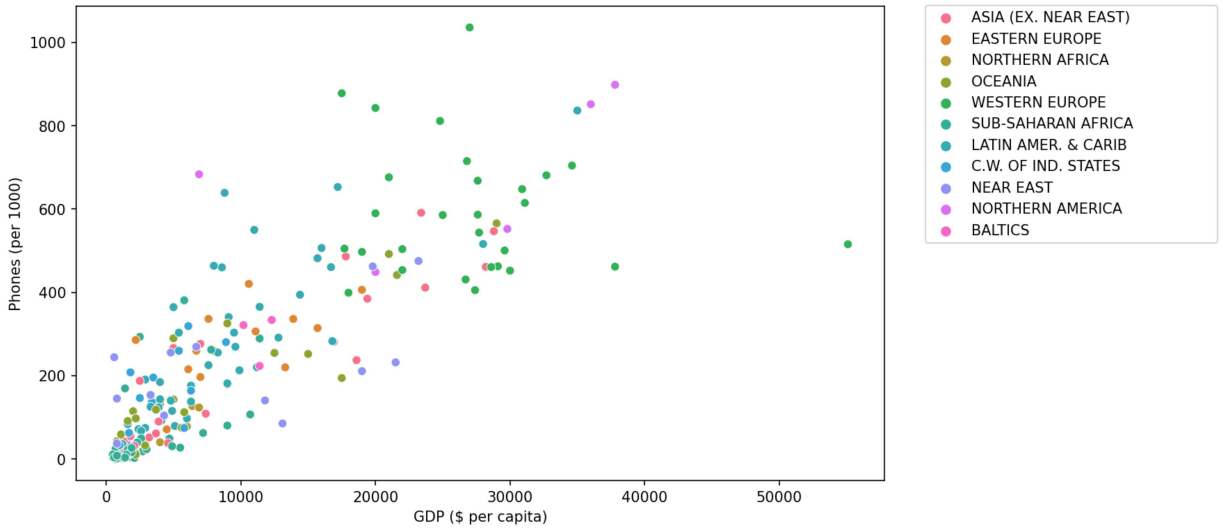


In [105]:
```
plt.figure(figsize=(10,6),dpi=150)
sns.barplot(data=df,x='Region',y='GDP ($ per capita)')
plt.xticks(rotation=90);
```



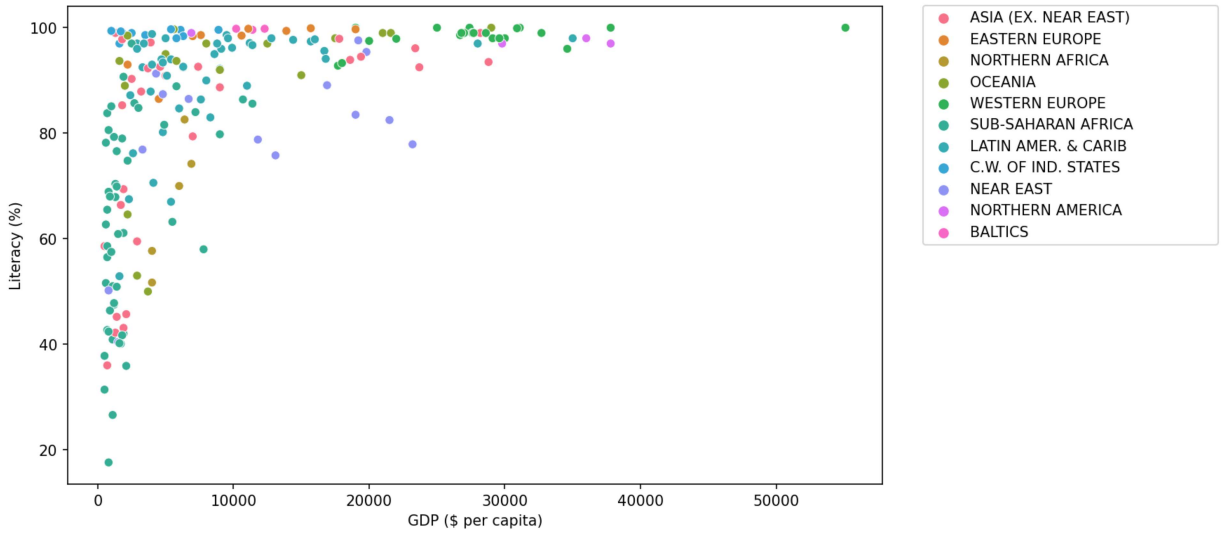
In [106]:
```
plt.figure(figsize=(10,6),dpi=150)
sns.scatterplot(data=df,x='GDP ($ per capita)',y='Phones (per 1000)',hue='Region'
plt.legend(loc=(1.05,0.5))
```

Out[106]: <matplotlib.legend.Legend at 0x16b96d8c310>



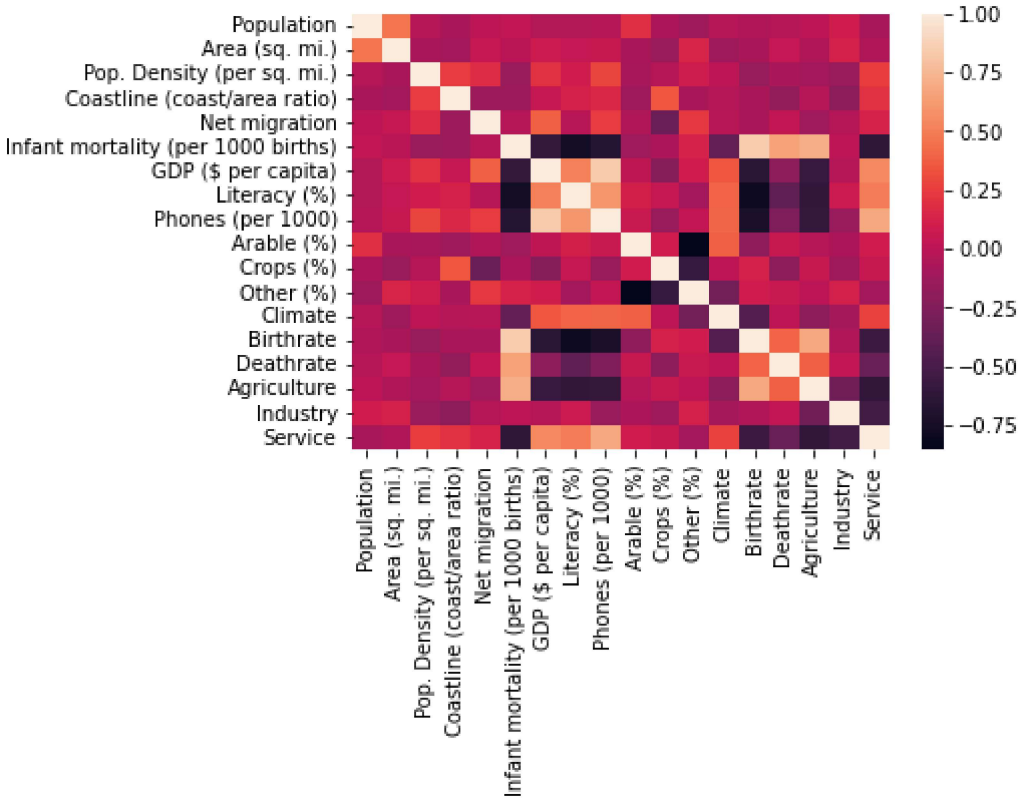
In [107]:
```
plt.figure(figsize=(10,6),dpi=150)
sns.scatterplot(data=df,x='GDP ($ per capita)',y='Literacy (%)',hue='Region')
plt.legend(loc=(1.05,0.5))
```

Out[107]:
```
<matplotlib.legend.Legend at 0x16b96f2cb50>
```



In [108]:
```
sns.heatmap(data=df.corr())
plt.figure(figsize=(10,6),dpi=150)
```

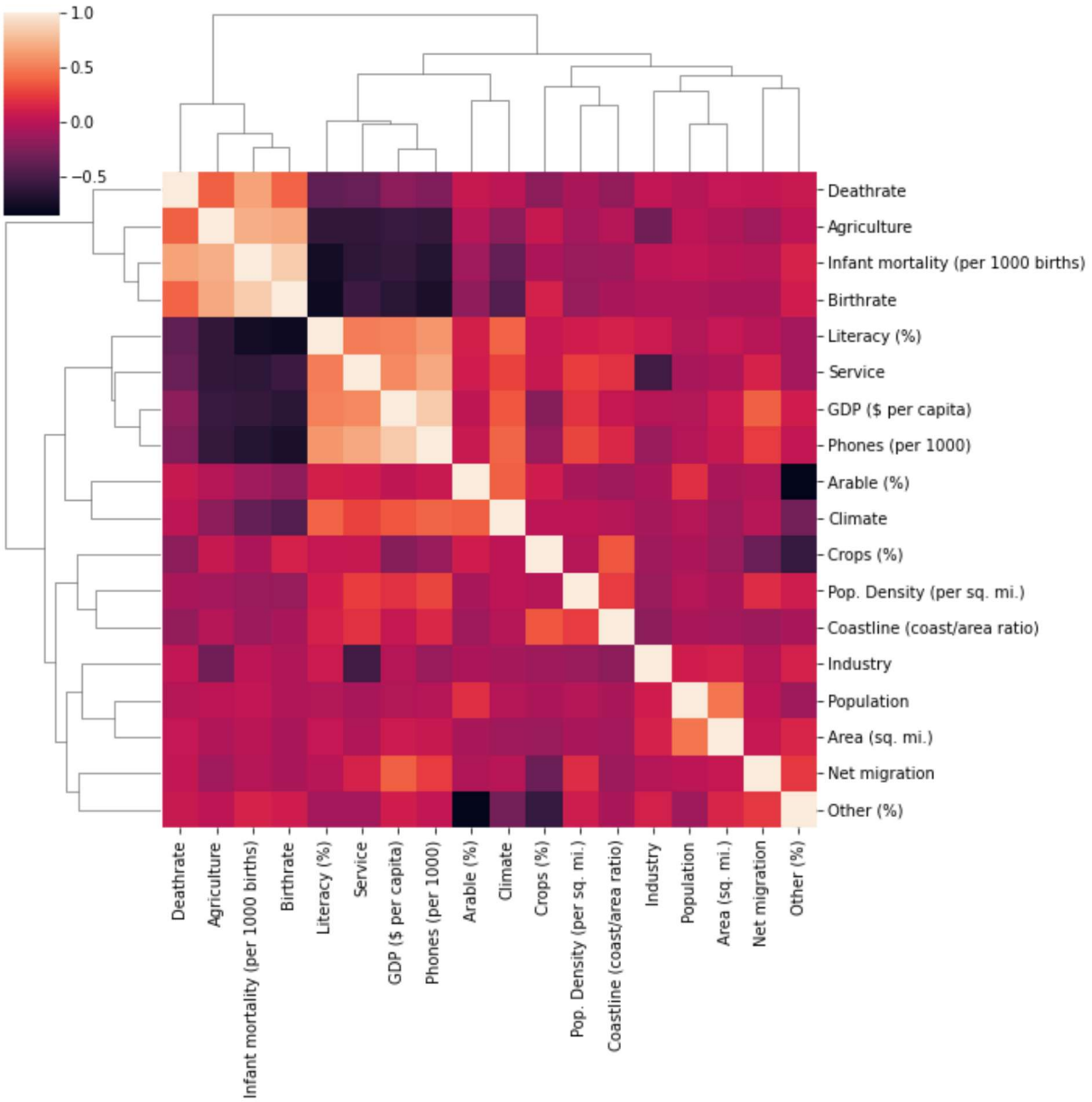
Out[108]:
```
<Figure size 1500x900 with 0 Axes>
```




```
<Figure size 1500x900 with 0 Axes>
```

In [109]:
```
sns.clustermap(data=df.corr())
```

Out[109]: <seaborn.matrix.ClusterGrid at 0x16b945d8be0>



In [110]:
```
df.isnull().sum()
```

Out[110]:
```
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 [111]:
```
df[df['Agriculture'].isnull() == True]['Country']
```

Out[111]:
```
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
```

In [112]:
```
df[df['Agriculture'].isnull()] = df[df['Agriculture'].isnull()].fillna(0)
```

In [113]:
```
df.isnull().sum()
```

Out[113]:
```
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 [114]:
```
df[df['Climate'].isnull()==True]['Region']
```

Out[114]:
```
5      SUB-SAHARAN AFRICA
36     NORTHERN AMERICA
50     EASTERN EUROPE
66     WESTERN EUROPE
101    WESTERN EUROPE
115    NEAR EAST
118    NORTHERN AFRICA
120    BALTICS
121    WESTERN EUROPE
129    WESTERN EUROPE
137                  C.W. OF IND. STATES
138    WESTERN EUROPE
141    NORTHERN AFRICA
145          ASIA (EX. NEAR EAST)
169                  C.W. OF IND. STATES
181    EASTERN EUROPE
186    EASTERN EUROPE
200    SUB-SAHARAN AFRICA
Name: Region, dtype: object
```

In [115]:
```
df['Climate'] = df['Climate'].fillna(df.groupby('Region')['Climate'].transform('
```

In [116]:
```
df.isnull().sum()
```

Out[116]:
```
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 [117]:
```
df['Literacy (%)'] = df['Literacy (%)'].fillna(df.groupby('Region')['Literacy (%)
```

In [118]:
```
df.isnull().sum()
```

Out[118]:
```
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
```

In [119]:
```
df = df.dropna()
```

In [120]:
```
df.isnull().sum()
```

Out[120]:
```
Country                               0
Region                                0
Population                            0
Area (sq. mi.)                        0
Pop. Density (per sq. mi.)            0
Coastline (coast/area ratio)          0
Net migration                         0
Infant mortality (per 1000 births)    0
GDP ($ per capita)                    0
Literacy (%)                          0
Phones (per 1000)                     0
Arable (%)                            0
Crops (%)                             0
Other (%)                             0
Climate                               0
Birthrate                             0
Deathrate                             0
Agriculture                           0
Industry                              0
Service                               0
dtype: int64
```

In [122]:
```
X = df.drop('Country',axis=1)
```

In [123]:
```
X = pd.get_dummies(X)
```

In [124]:
```
X
```

Out[124]:

STERN UROPE	Region_LATIN AMER. & CARIB	Region_NEAR EAST	Region_NORTHERN AFRICA	Region_NORTHERN AMERICA	Region_OCEANIA	Re
0	0	0	0	0	0	
1	0	0	0	0	0	
0	0	0	1	0	0	
0	0	0	0	0	1	
0	0	0	0	0	0	
...	...	...	...	...	...	
0	0	1	0	0	0	
0	0	0	1	0	0	
0	0	1	0	0	0	
0	0	0	0	0	0	
0	0	0	0	0	0	

In [125]:
```
X.head()
```

Out[125]:

	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)	Arab (%
0	31056997	647500	48.0	0.00	23.06	163.07	700.0	36.0	3.2	12.1
1	3581655	28748	124.6	1.26	-4.93	21.52	4500.0	86.5	71.2	21.0
2	32930091	2381740	13.8	0.04	-0.39	31.00	6000.0	70.0	78.1	3.2
3	57794	199	290.4	58.29	-20.71	9.27	8000.0	97.0	259.5	10.0
4	71201	468	152.1	0.00	6.60	4.05	19000.0	100.0	497.2	2.2

5 rows × 29 columns

In [126]:
```
from sklearn.preprocessing import StandardScaler
```

In [127]:
```
scaler = StandardScaler()
```

In [128]:
```
scaled_X = scaler.fit_transform(X)
```

In [129]:
```
scaled_X
```

Out[129]:
```
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]])
```

In [131]:
```
from sklearn.cluster import KMeans
```

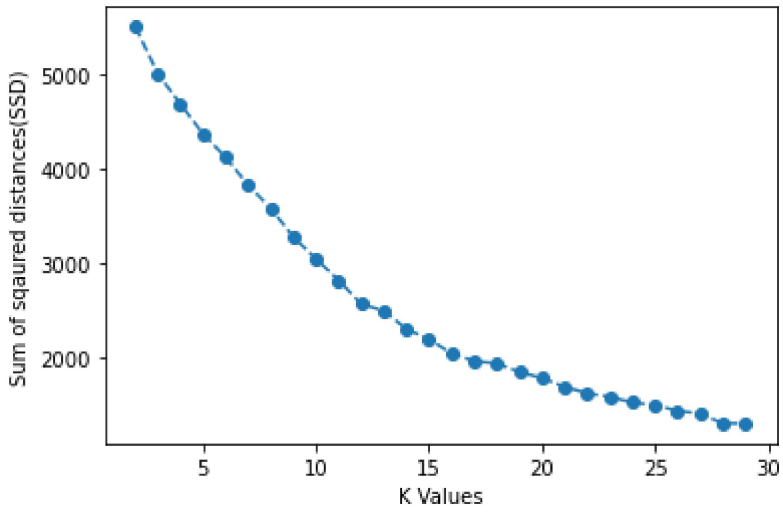
In [138]:
```
ssd = []

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

In [139]:
```
ssd
```

Out[139]:
```
[5496.1778057452575,
 5001.629405600431,
 4679.30415416996,
 4367.373633918766,
 4118.8194771595845,
 3826.6048798075312,
 3580.811856436625,
 3277.7890664115857,
 3048.633805807907,
 2823.9554586293207,
 2571.7812468600337,
 2502.0969583545484,
 2307.4641227674574,
 2201.341313463305,
 2050.51032564894,
 1970.9836256774315,
 1946.204784098216,
 1857.070449444394,
 1791.7948563475438,
 1700.5087469310342,
 1631.1163425260809,
 1593.551193788919,
 1534.3051489134834,
 1506.3533836956535,
 1447.4141430243892,
 1415.502681920085,
 1313.1261131272106,
 1313.3244976739136]
```

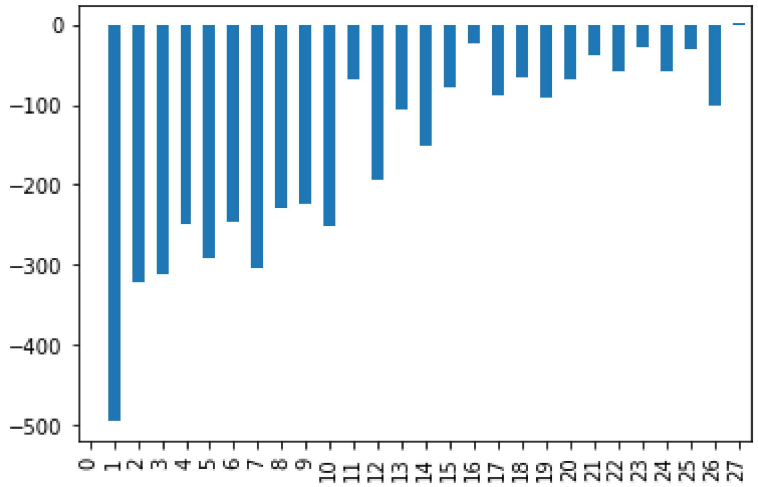
In [141]:
```
plt.plot(range(2,30),ssd,'o--')
plt.xlabel('K Values')
plt.ylabel('Sum of sqaured distances(SSD)')
```

Out[141]:
```
Text(0, 0.5, 'Sum of sqaured distances(SSD)')
```



In [149]:
```
pd.Series(ssd).diff().plot(kind='bar')
```

Out[149]: <AxesSubplot:>



In [176]:
```
# Let us take the value for k = 3
model = KMeans(n_clusters=3)
model.fit(scaled_X)
```

Out[176]:
```
KMeans(n_clusters=3)
```

In [178]:
```
model.labels_
```

Out[178]:
```
array([1, 0, 0, 0, 2, 1, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 0, 1,
       2, 1, 0, 2, 1, 0, 2, 0, 2, 1, 0, 1, 0, 1, 2, 0, 2, 1, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 2, 0, 2, 2, 1, 0, 0, 0, 0, 0, 1, 1, 2, 1, 2, 0, 2,
       2, 0, 0, 1, 1, 0, 0, 2, 1, 2, 2, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 2,
       2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 0, 2, 2, 0, 0, 1, 0, 0, 2, 0, 0, 1,
       2, 0, 1, 1, 0, 2, 2, 2, 2, 2, 1, 1, 0, 0, 1, 2, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 1, 2, 0, 0, 2, 0, 1, 1, 0, 2, 0, 0, 0, 0,
       0, 0, 0, 0, 2, 2, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 1,
       0, 1, 2, 2, 2, 0, 1, 1, 2, 0, 1, 0, 1, 2, 2, 0, 2, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1])
```

In [ ]:
```

```