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. What do these clusters represent?

Imports and Data

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

warnings.filterwarnings('ignore')
```

In [2]:
```python
data = pd.read_csv('../DATA/CIA_Country_Facts.csv')
```

Exploratory Data Analysis

> ### Data

In [3]:
```python
data.head(5)
```

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 capi
0	Afghanistan	ASIA (EX. NEAR EAST)	31056997	647500	48.0	0.00	23.06	163.07	70
1	Albania	EASTERN EUROPE	3581655	28748	124.6	1.26	-4.93	21.52	450
2	Algeria	NORTHERN AFRICA	32930091	2381740	13.8	0.04	-0.39	31.00	600
3	American Samoa	OCEANIA	57794	199	290.4	58.29	-20.71	9.27	800
4	Andorra	WESTERN EUROPE	71201	468	152.1	0.00	6.60	4.05	1900

General

In [4]:
```
data.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]:
```
data.describe().transpose()
```

Out[5]:

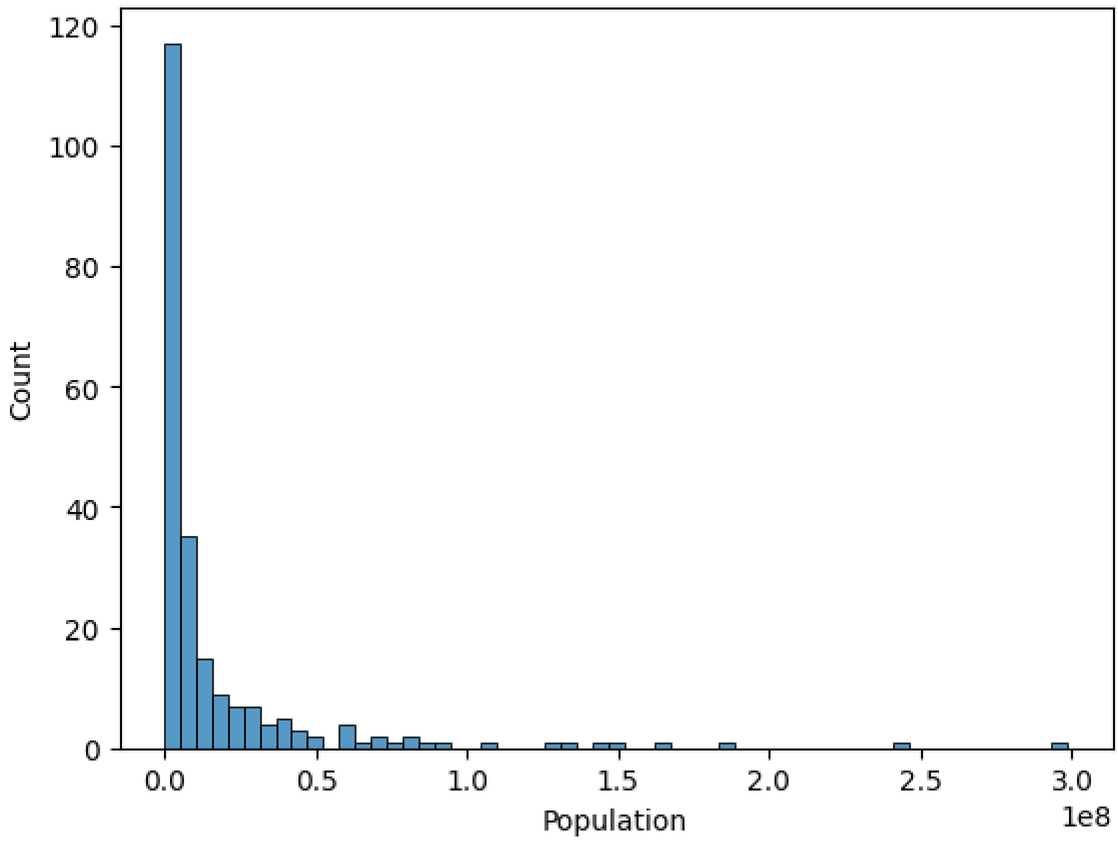
	count	mean	std	min	25%	50%	75
Population	227.0	2.874028e+07	1.178913e+08	7026.000	437624.00000	4786994.000	1.749777e+(
Area (sq. mi.)	227.0	5.982270e+05	1.790282e+06	2.000	4647.50000	86600.000	4.418110e+(
Pop. Density (per sq. mi.)	227.0	3.790471e+02	1.660186e+03	0.000	29.15000	78.800	1.901500e+(
Coastline (coast/area ratio)	227.0	2.116533e+01	7.228686e+01	0.000	0.10000	0.730	1.034500e+(
Net migration	224.0	3.812500e-02	4.889269e+00	-20.990	-0.92750	0.000	9.975000e-(
Infant mortality (per 1000 births)	224.0	3.550696e+01	3.538990e+01	2.290	8.15000	21.000	5.570500e+(
GDP ($ per capita)	226.0	9.689823e+03	1.004914e+04	500.000	1900.00000	5550.000	1.570000e+(
Literacy (%)	209.0	8.283828e+01	1.972217e+01	17.600	70.60000	92.500	9.800000e+(
Phones (per 1000)	223.0	2.360614e+02	2.279918e+02	0.200	37.80000	176.200	3.896500e+(
Arable (%)	225.0	1.379711e+01	1.304040e+01	0.000	3.22000	10.420	2.000000e+(
Crops (%)	225.0	4.564222e+00	8.361470e+00	0.000	0.19000	1.030	4.440000e+(
Other (%)	225.0	8.163831e+01	1.614083e+01	33.330	71.65000	85.700	9.544000e+(
Climate	205.0	2.139024e+00	6.993968e-01	1.000	2.00000	2.000	3.000000e+(
Birthrate	224.0	2.211473e+01	1.117672e+01	7.290	12.67250	18.790	2.982000e+(
Deathrate	223.0	9.241345e+00	4.990026e+00	2.290	5.91000	7.840	1.060500e+(
Agriculture	212.0	1.508443e-01	1.467980e-01	0.000	0.03775	0.099	2.210000e-(
Industry	211.0	2.827109e-01	1.382722e-01	0.020	0.19300	0.272	3.410000e-(
Service	212.0	5.652830e-01	1.658410e-01	0.062	0.42925	0.571	6.785000e-(

Visualizations

- **Histogram of population for countries having population less than 0.5 billion**

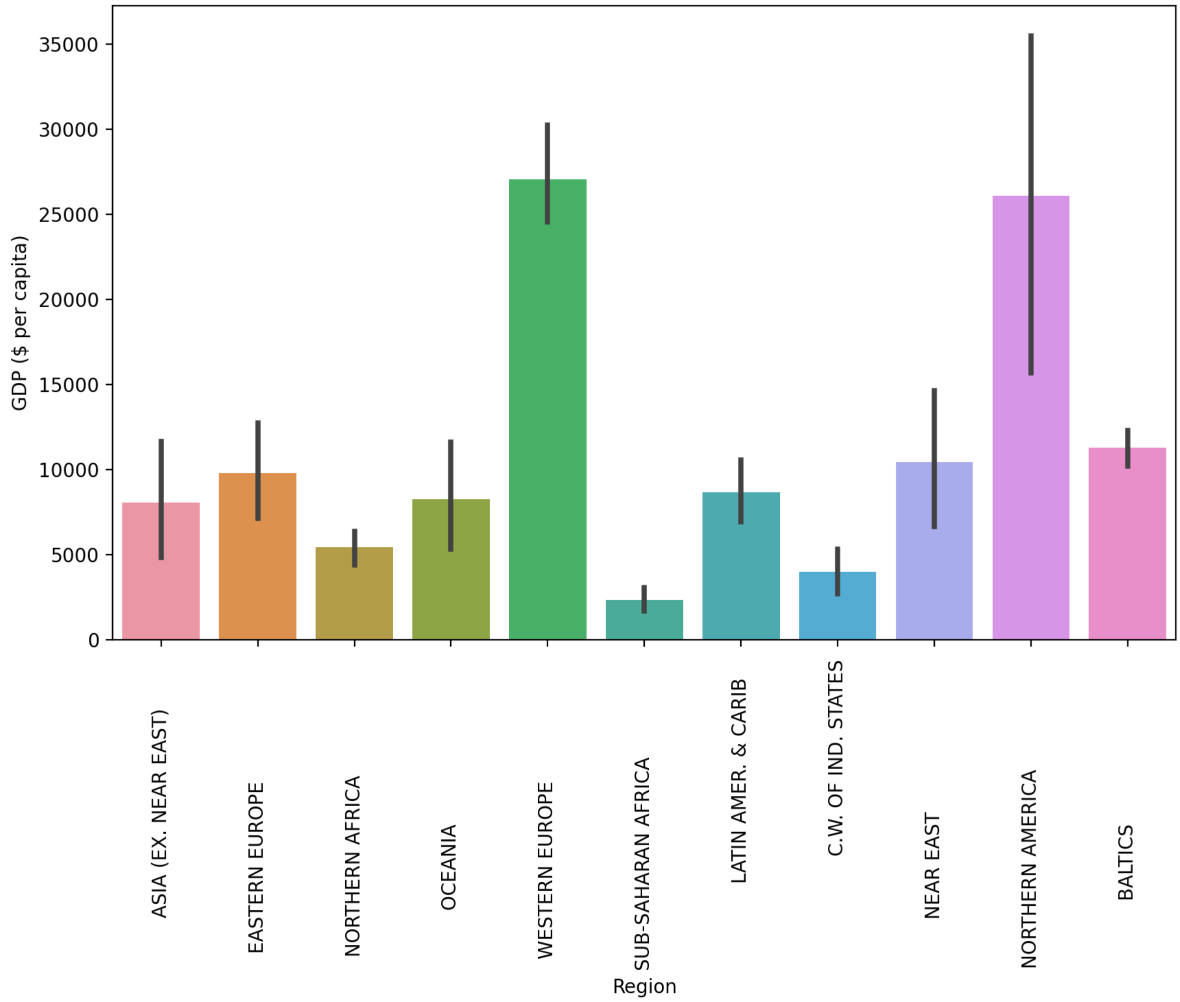
In [6]:

```
sns.histplot(data=data[data['Population']<500000000], x='Population');
```



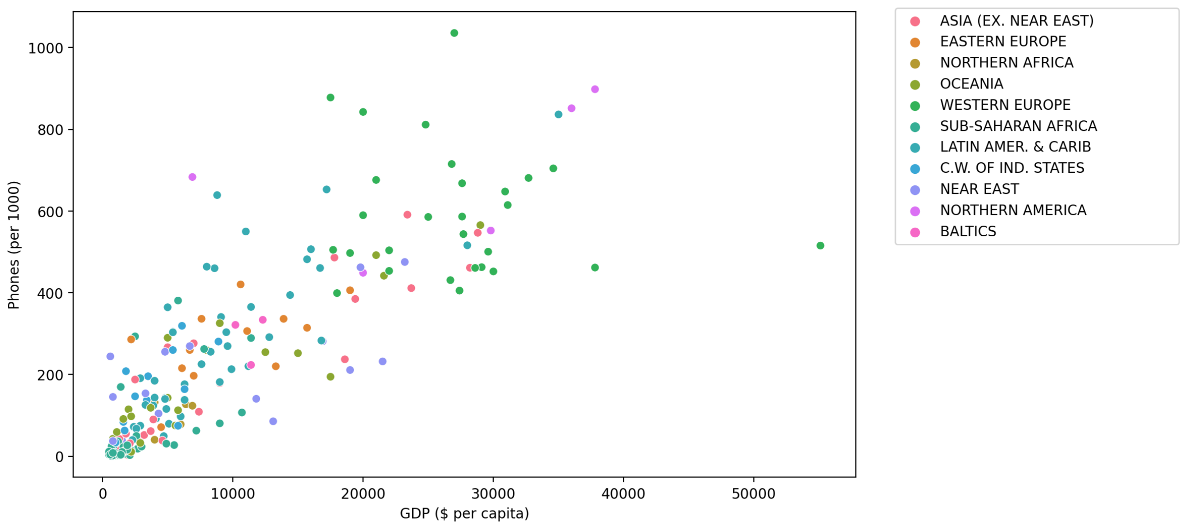
- **Barplot of GDP per capita per region *(black bar represents std)***

In [7]:
```
plt.figure(figsize=(10, 6), dpi=200)
sns.barplot(data=data, x='Region', y='GDP ($ per capita)', estimator=np.mean)
plt.xticks(rotation=90);
```



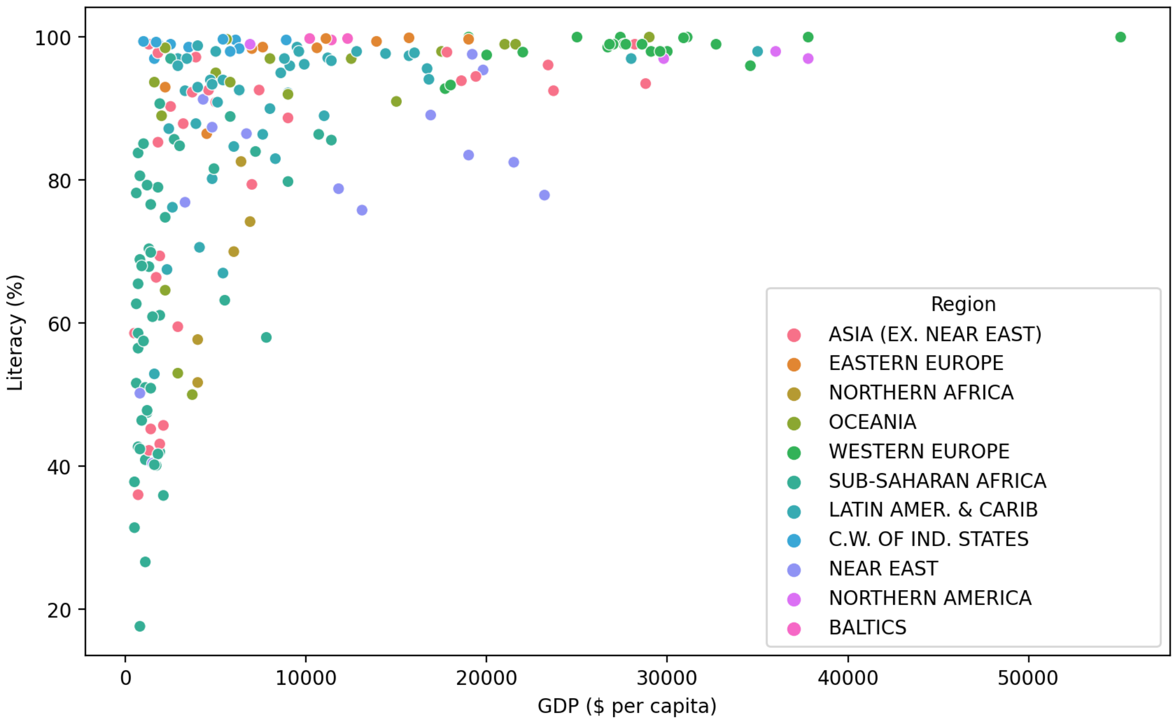
- **Scatterplot showing the relationship between Phones per 1000 people and the GDP per Capita**

In [8]:
```
plt.figure(figsize=(10, 6), dpi=200)
sns.scatterplot(data=data, x='GDP ($ per capita)', y='Phones (per 1000)', hue='Regi
plt.legend(loc=(1.05, 0.5));
```



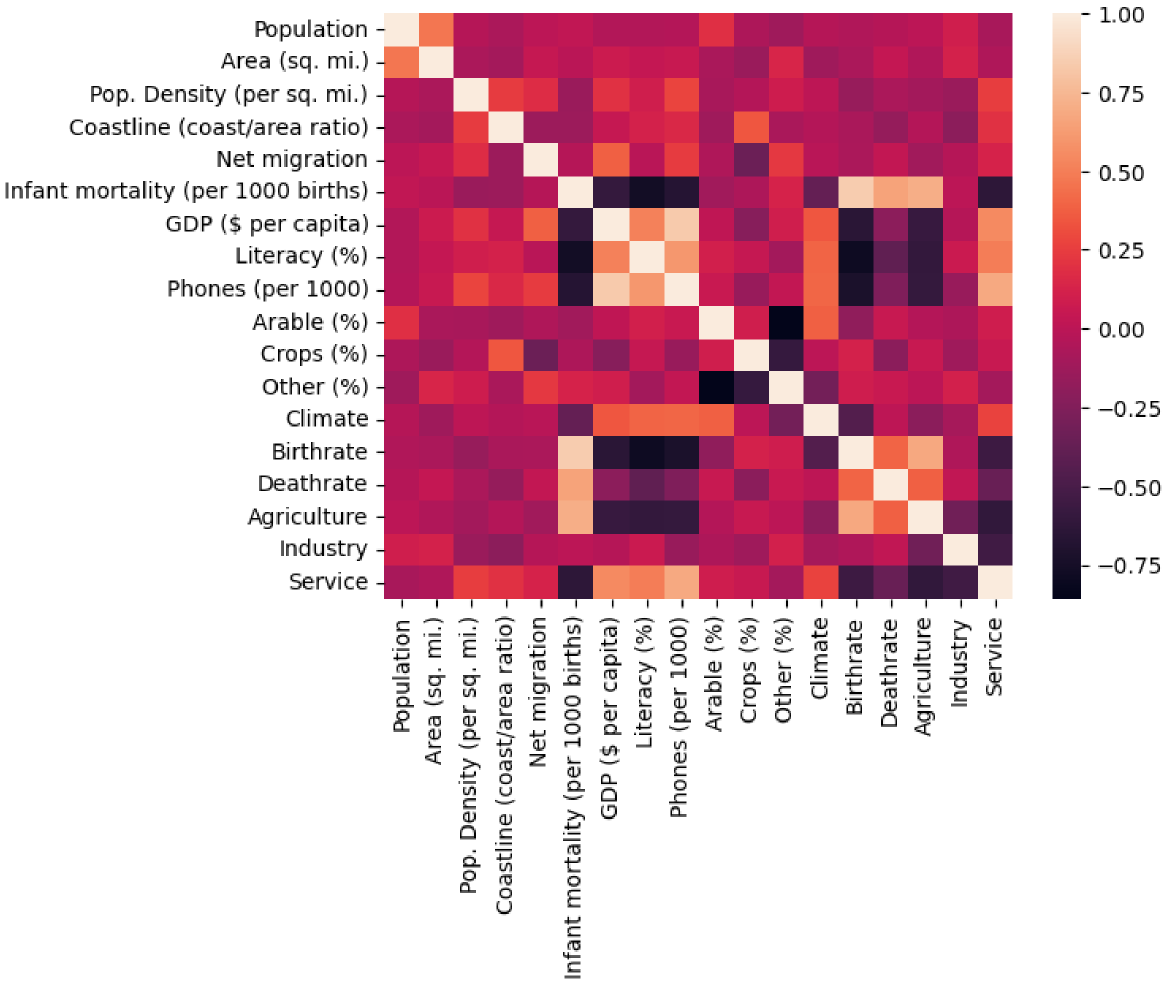
- **Scatterplot showing the relationship between Literacy and the GDP per Capita**

In [9]:
```
plt.figure(figsize=(10, 6), dpi=200)
sns.scatterplot(data=data, x='GDP ($ per capita)', y='Literacy (%)', hue='Region');
```



- **Heatmap of the Correlation between columns in the data**

In [10]:
```
sns.heatmap(data.corr());
```



Data Preparation and Model Discovery

Missing Data

In [11]:
```
data.isnull().sum()
```

Out[11]:
```
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
```

1. Agriculture

In [12]:
```
data[data['Agriculture'].isnull()]['Country']
```

Out[12]:
```
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
```

Notice, most of these countries are tiny islands, with the exception of Greenland and Western Sahara. fill any of these countries missing NaN values with 0, since they are so small or essentially non-existant.

In [13]:
```
data[data['Agriculture'].isnull()] = data[data['Agriculture'].isnull()].fillna(0)
```

Notice, climate is missing for a few countries, but not the Region! filling in the missing Climate values based on the mean climate value for its region.

In [14]:
```
data['Climate'] = data['Climate'].fillna(data.groupby('Region')['Climate'].transfor
```

Notice, Literacy percentage is missing. Using the same tactic as we did with Climate missing values and filling in any missing Literacy % values with the mean Literacy % of

the Region.

In [15]:
```
data['Literacy (%)'] = data['Literacy (%)'].fillna(data.groupby('Region')['Literacy
```

Checking again on the remaining missing values:

In [16]:
```
data.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                         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
```

Notice, We are now missing values for only a few countries. Dropping these countries.

In [17]:
```
data = data.dropna()
```

Data Feature Preparation

It is now time to prepare the data for clustering. The Country column is still a unique identifier string, so it won't be useful for clustering, since its unique for each point. Dropping this Country column.

In [18]:
```
X = data.drop('Country', axis=1)
```

Now creating the X array of features, the Region column is still categorical strings, using Pandas to create dummy variables from this column to create a finalzed X matrix of continuous features along with the dummy variables for the Regions.

In [19]:
```
X = pd.get_dummies(X)
```

In [20]:
```
X.head(5)
```

Out[20]:

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

5 rows × 29 columns

Due to some measurements being in terms of percentages and other metrics being total counts (population), we should scale this data first. Using Sklearn to scale the X feature matrics.

In [21]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

In [22]:
```
Scaled_X = scaler.fit_transform(X)
Scaled_X
```

Out[22]:
```
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

Using a for loop to create and fit multiple KMeans models, testing from K=2-30 clusters. tracking of the Sum of Squared Distances for each K value, then ploting this out to create an "elbow" plot of K versus SSD.

In [23]:
```
from sklearn.cluster import KMeans
```

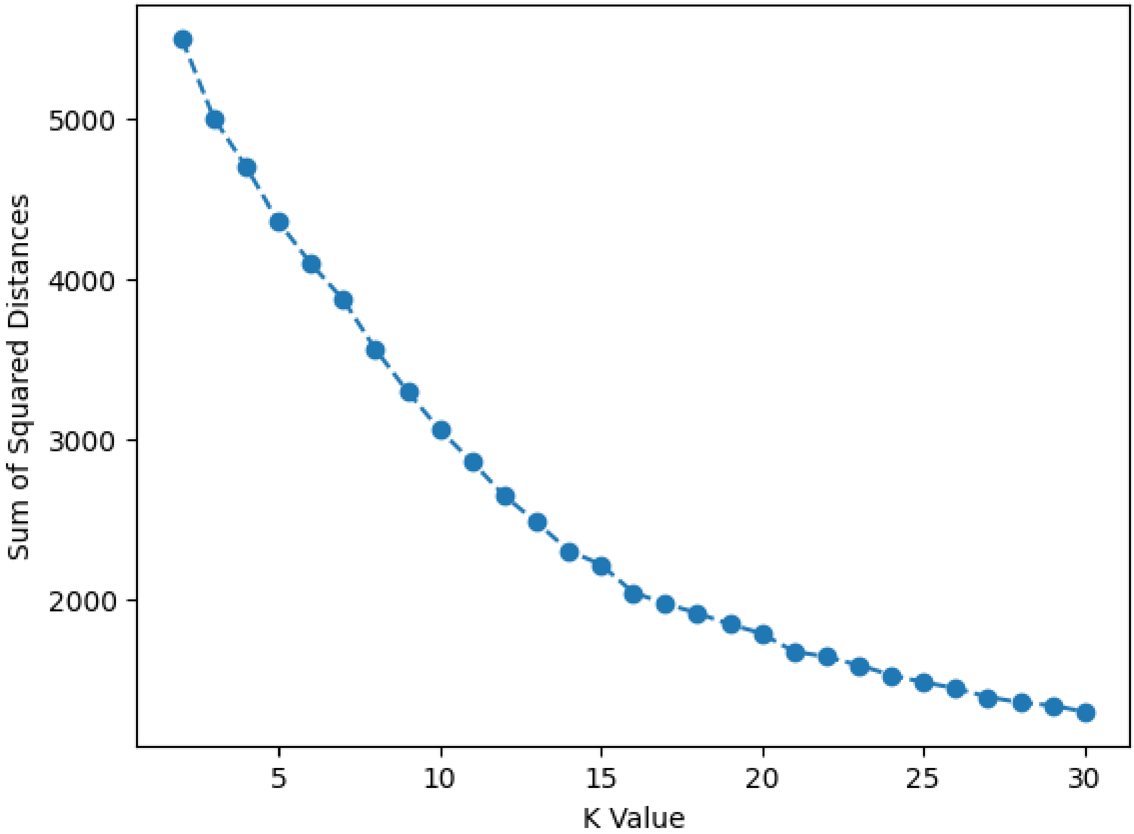
In [24]:
```
sum_of_squared_distances = []

for k in range(2, 31):
```

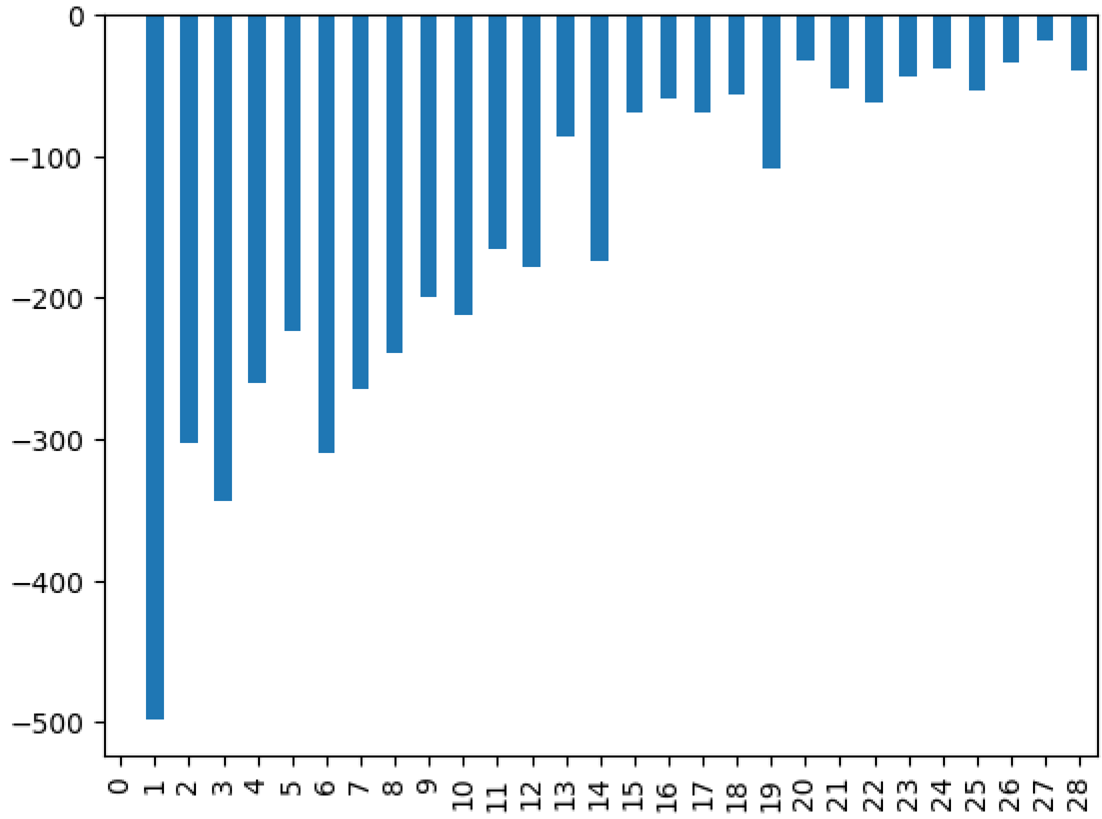
```
    model = KMeans(n_clusters=k)
    model.fit(Scaled_X)

    sum_of_squared_distances.append(model.inertia_)
```

In [25]:
```
plt.plot(range(2,31),sum_of_squared_distances,'o--')
plt.xlabel("K Value")
plt.ylabel("Sum of Squared Distances");
```



In [26]:
```
pd.Series(sum_of_squared_distances).diff().plot(kind='bar');
```



Model Interpretation

Choosing K=15 from above

Remember, there is no 100% correct answer here!

Example Interpretation: Choosing K=15

Let's explore which features are important in the decision of 15 clusters!

In [27]:
```
model = KMeans(n_clusters=15)
model.fit(Scaled_X)
```

Out[27]:
```
▾ KMeans
KMeans(n_clusters=15)
```

In [28]:
```
model.labels_
```

Out[28]:
```
array([ 0,  8, 14,  9,  7,  0,  5,  5,  5, 10,  5,  9,  7, 10,  5,  4,  2,
        5, 10,  7,  5,  3, 13,  2,  5,  8,  3,  5,  5,  2,  8,  0,  2,  3,
        2,  3, 13,  3,  7,  0,  0,  5, 11,  5,  3,  0,  3,  5,  3,  8,  5,
        8,  7,  3,  5,  5,  5, 14,  5,  3,  3,  6,  0,  7,  9,  7,  7,  5,
        9,  3,  3,  4, 10,  7,  3,  9,  7, 13,  5,  5,  9,  5,  3,  0,  5,
        3,  5,  1,  8,  7, 11,  2,  2,  4,  7,  7,  4,  7,  5,  2,  7,  4,
       10,  3, 12,  2,  2,  4, 10,  2,  6,  4,  3,  0, 14,  7,  6,  7,  1,
        8,  3,  3,  2,  2,  0,  7, 12,  5,  3,  5,  3,  5, 12, 10,  2,  5,
       14,  0,  3,  9,  2,  7,  5,  9,  9,  5,  0,  3,  9,  7,  4,  2,  9,
        5,  9,  5,  5,  2,  8,  7,  5,  4,  5,  8, 10,  3,  9,  5,  5, 13,
        5, 12,  7, 12,  4,  3,  5,  0,  1,  8,  8,  9,  0,  3,  7,  2,  3,
        5,  3,  7,  7,  4,  2, 10,  0,  2,  3, 12,  5, 14,  4, 10,  5,  9,
        3, 10,  4,  7, 13,  5, 10,  9,  5,  2,  5,  9,  4, 14,  4,  3,  3])
```

In [29]:
```
X['K=15 Clusters'] = model.labels_
```

In [30]:
```
X.corr()['K=15 Clusters'].sort_values()
```

Out[30]:
```
Infant mortality (per 1000 births)          -0.502814
Birthrate                                   -0.491648
Region_SUB-SAHARAN AFRICA                   -0.478571
Deathrate                                   -0.383261
Agriculture                                 -0.369689
Region_ASIA (EX. NEAR EAST)                 -0.337509
Other (%)                                   -0.151097
Pop. Density (per sq. mi.)                  -0.129180
Net migration                               -0.128289
Region_NEAR EAST                            -0.125322
Region_LATIN AMER. & CARIB                  -0.088229
Industry                                    -0.077783
Region_BALTICS                               0.013543
Service                                      0.050229
Arable (%)                                   0.092918
Population                                   0.119624
Climate                                      0.126397
Region_WESTERN EUROPE                        0.156170
Region_EASTERN EUROPE                        0.159138
Area (sq. mi.)                               0.187606
Coastline (coast/area ratio)                 0.201745
Crops (%)                                    0.202589
GDP ($ per capita)                           0.241492
Literacy (%)                                 0.274769
Region_C.W. OF IND. STATES                   0.305559
Region_NORTHERN AMERICA                      0.326353
Phones (per 1000)                            0.326466
Region_OCEANIA                               0.379385
Region_NORTHERN AFRICA                       0.406768
K=15 Clusters                                1.000000
Name: K=15 Clusters, dtype: float64
```