EDA

Importing Python Libraries for EDA

In [35]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

Loading Dataset

In [5]:
```
df=pd.read_csv('D:\GWC\Github\Streamlit_Project\World\World.csv')
```

Displaying Top 5 and Bottom 5 of DataFrame

In [6]:
```
df.head()
```

Out[6]:

	country	year	pop	continent	lifeExp	gdpPercap
0	Afghanistan	1952	8425333.0	Asia	28.801	779.445314
1	Afghanistan	1957	9240934.0	Asia	30.332	820.853030
2	Afghanistan	1962	10267083.0	Asia	31.997	853.100710
3	Afghanistan	1967	11537966.0	Asia	34.020	836.197138
4	Afghanistan	1972	13079460.0	Asia	36.088	739.981106

In [7]:
```
df.tail()
```

Out[7]:

	country	year	pop	continent	lifeExp	gdpPercap
1699	Zimbabwe	1987	9216418.0	Africa	62.351	706.157306
1700	Zimbabwe	1992	10704340.0	Africa	60.377	693.420786
1701	Zimbabwe	1997	11404948.0	Africa	46.809	792.449960
1702	Zimbabwe	2002	11926563.0	Africa	39.989	672.038623
1703	Zimbabwe	2007	12311143.0	Africa	43.487	469.709298

Displaying Shape of DataFrame

In [20]:
```
df.size
```

Out[20]:
```
10224
```

In [21]:
```
df.shape
```

Out[21]:
```
(1704, 6)
```

In [19]:
```
df.ndim
```

Out[19]:
```
2
```

- DataFrame have **10224** elements
- DataFrame have **1704 Rows** and **6 columns**
- DataFrame is a 2-Dimensional

Displaying Basic Info of DataFrame

In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1704 entries, 0 to 1703
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   country    1704 non-null   object 
 1   year       1704 non-null   int64  
 2   pop        1704 non-null   float64
 3   continent  1704 non-null   object 
 4   lifeExp    1704 non-null   float64
 5   gdpPercap  1704 non-null   float64
dtypes: float64(3), int64(1), object(2)
memory usage: 80.0+ KB
```

- This DataFrame doesnot contains any null values.
- This dataFrame have
  - 3 columns with **float64** datatype
  - 2 columns with **object** datatype
  - 1 column with **int64** datatype

Descriptive Statistics

- statistical information of all numerical columns in our data frame

In [14]:
```
df.describe()
```

Out[14]:

	year	pop	lifeExp	gdpPercap
count	1704.00000	1.704000e+03	1704.000000	1704.000000
mean	1979.50000	2.960121e+07	59.474439	7215.327081
std	17.26533	1.061579e+08	12.917107	9857.454543
min	1952.00000	6.001100e+04	23.599000	241.165876
25%	1965.75000	2.793664e+06	48.198000	1202.060309
50%	1979.50000	7.023596e+06	60.712500	3531.846988
75%	1993.25000	1.958522e+07	70.845500	9325.462346
max	2007.00000	1.318683e+09	82.603000	113523.132900

- This DataFrame has data from year **1952** to **2007**
- The Average Life Expectancy is **59.47**
- The Average GDP per Capita is **7215.32**

Displaying Unique Values

In [25]:
```
df.nunique()
```

Out[25]:
```
country       142
year           12
pop          1704
continent       5
lifeExp      1626
gdpPercap    1704
dtype: int64
```

- DataFrame have **5** Continent.
- DataFrame have **142** Countries.
- DataFrame have **12** Years.

In [29]:
```
# List of Continent in DataFrame
df['continent'].unique()
```

Out[29]:
```
array(['Asia', 'Europe', 'Africa', 'Americas', 'Oceania'], dtype=object)
```

In [30]:
```
#List of Countries in Dataframe
df['country'].unique()
```

Out[30]:
```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Argentina',
       'Australia', 'Austria', 'Bahrain', 'Bangladesh', 'Belgium',
       'Benin', 'Bolivia', 'Bosnia and Herzegovina', 'Botswana', 'Brazil',
       'Bulgaria', 'Burkina Faso', 'Burundi', 'Cambodia', 'Cameroon',
       'Canada', 'Central African Republic', 'Chad', 'Chile', 'China',
       'Colombia', 'Comoros', 'Congo, Dem. Rep.', 'Congo, Rep.',
       'Costa Rica', "Cote d'Ivoire", 'Croatia', 'Cuba', 'Czech Republic',
       'Denmark', 'Djibouti', 'Dominican Republic', 'Ecuador', 'Egypt',
       'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Ethiopia',
       'Finland', 'France', 'Gabon', 'Gambia', 'Germany', 'Ghana',
       'Greece', 'Guatemala', 'Guinea', 'Guinea-Bissau', 'Haiti',
       'Honduras', 'Hong Kong, China', 'Hungary', 'Iceland', 'India',
       'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Jamaica', 'Japan', 'Jordan', 'Kenya', 'Korea, Dem. Rep.',
       'Korea, Rep.', 'Kuwait', 'Lebanon', 'Lesotho', 'Liberia', 'Libya',
       'Madagascar', 'Malawi', 'Malaysia', 'Mali', 'Mauritania',
       'Mauritius', 'Mexico', 'Mongolia', 'Montenegro', 'Morocco',
       'Mozambique', 'Myanmar', 'Namibia', 'Nepal', 'Netherlands',
       'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Norway', 'Oman',
       'Pakistan', 'Panama', 'Paraguay', 'Peru', 'Philippines', 'Poland',
       'Portugal', 'Puerto Rico', 'Reunion', 'Romania', 'Rwanda',
       'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia',
       'Sierra Leone', 'Singapore', 'Slovak Republic', 'Slovenia',
       'Somalia', 'South Africa', 'Spain', 'Sri Lanka', 'Sudan',
       'Swaziland', 'Sweden', 'Switzerland', 'Syria', 'Taiwan',
       'Tanzania', 'Thailand', 'Togo', 'Trinidad and Tobago', 'Tunisia',
       'Turkey', 'Uganda', 'United Kingdom', 'United States', 'Uruguay',
       'Venezuela', 'Vietnam', 'West Bank and Gaza', 'Yemen, Rep.',
       'Zambia', 'Zimbabwe'], dtype=object)
```

In [31]:
```
#List of year in DataFrame
df['year'].unique()
```

Out[31]:
```
array([1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, 2002,
       2007], dtype=int64)
```

- In this DataFrame we have list of 12 Years with **5 years** intervals starting from **1952** to **2007**

Relationship between Columns in DataFrame

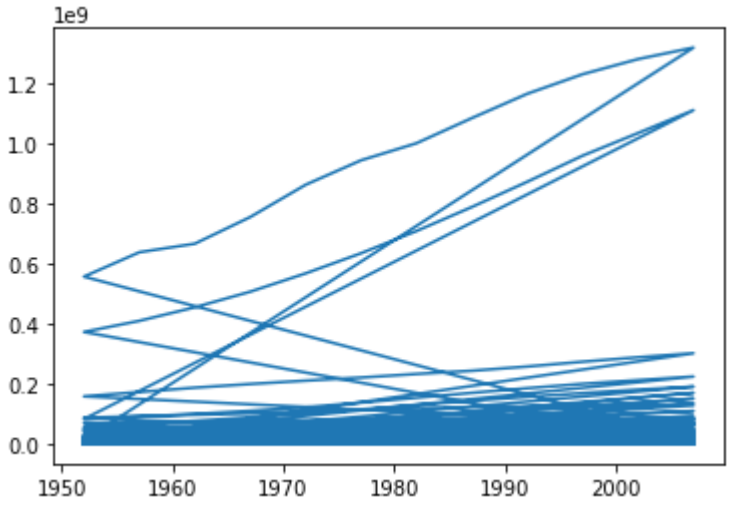
In [32]:
```
df.corr()
```

Out[32]:

	year	pop	lifeExp	gdpPercap
year	1.000000	0.082308	0.435611	0.227318
pop	0.082308	1.000000	0.064955	-0.025600
lifeExp	0.435611	0.064955	1.000000	0.583706
gdpPercap	0.227318	-0.025600	0.583706	1.000000

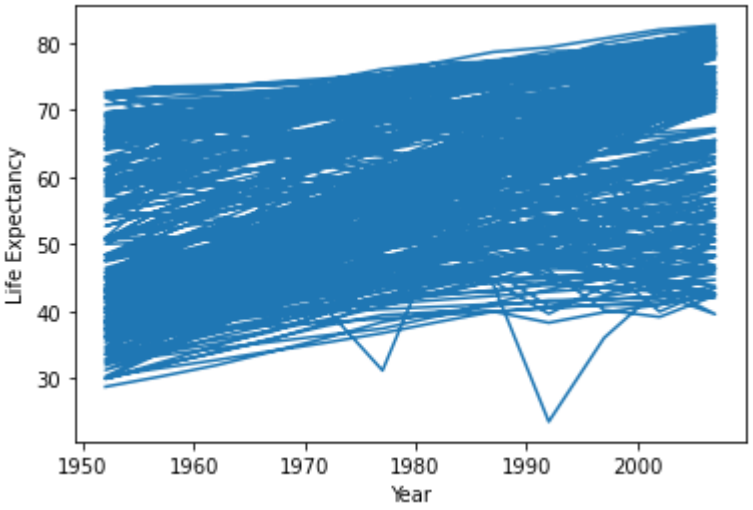
In [37]:
```
plt.plot(df['year'],df['pop'])
```

Out[37]:
```
[<matplotlib.lines.Line2D at 0xd377c70>]
```



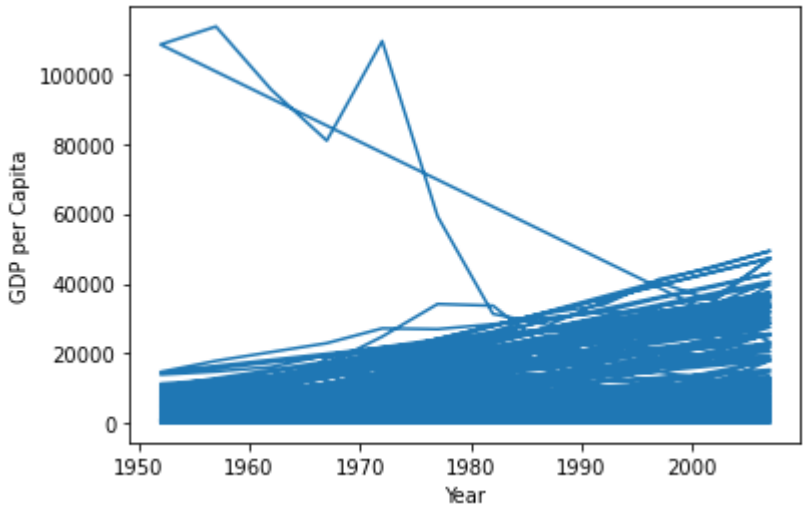
In [43]:
```
plt.plot(df['year'],df['lifeExp'])
plt.xlabel('Year')
plt.ylabel('Life Expectancy')
```

Out[43]:
```
Text(0, 0.5, 'Life Expectancy')
```



In [42]:
```
plt.plot(df['year'],df['gdpPercap'])
plt.xlabel('Year')
plt.ylabel('GDP per Capita')
```

Out[42]:
```
Text(0, 0.5, 'GDP per Capita')
```



Year has postive linear relationship with Population ,Life Expectancy, GDP per Capita

- Year is directly propotional to Population,Life Expecttancy,GDP percapita.If Year goes on the Population,life expectancy and GDP per capita also increases.

Population and Life Expectancy have a moderate positive linear relationship.

- Population is directly propotional to Life Expecttancy.If population increases the life expectancy also increases and vice-versa.

Population and GDP per Capita have a very weak negative linear relationship.

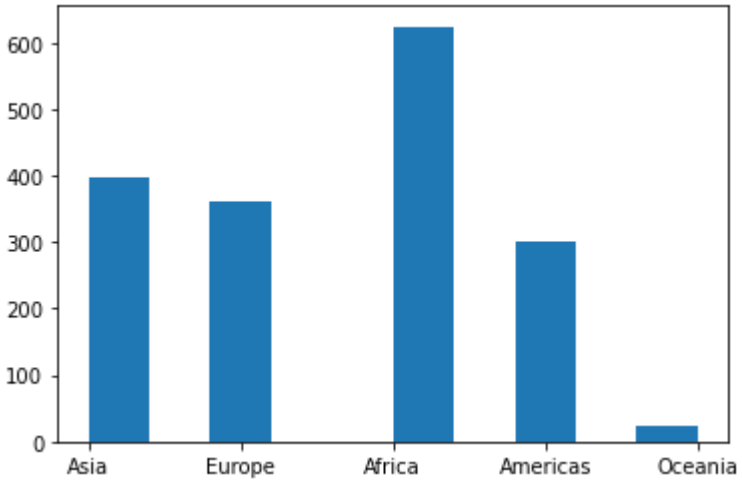
- population is indirectly proportional to GDP per Capita.If Population increases the GDP per capita slightly decreases and vice versa

Life Expactancy and GDP per Capita have a moderate positive linear relationship.

- GDP per Capita is directly propotional to Life Expecttancy.If GDP per capita increases the life expectancy also increases and vice-versa.

In [51]:
```
plt.hist(df['continent'])
```

Out[51]:
```
(array([396.,   0., 360.,   0.,   0., 624.,   0., 300.,   0.,  24.]),
 array([0. , 0.4, 0.8, 1.2, 1.6, 2. , 2.4, 2.8, 3.2, 3.6, 4. ]),
 <BarContainer object of 10 artists>)
```



- Africa have more records than other countries

In [ ]: