LAB 1: Exploratory Data Analysis and Data Visualization in Python

Name : Mohammed Meraj Mohammed Ashfaque

Class : TY - Computer **BATCH - T2**

Roll no : 32

Date : 25 Feb 2025

Import Libraries

In [95]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import DataSet

In [96]:
```
df = pd.read_csv('CountryUpdt.csv')
```

In [97]:
```
df
```

Out[97]:

	Country	Population	Area	GDP	Continent	Language	Currency	Capital	Category
0	Afghanistan	38928346	652230	20000000000	Asia	Pashto	Afghan Afghani	Kabul	0
1	Albania	2877797	28748	17000000000	Europe	Albanian	Albanian Lek	Tirana	0
2	Algeria	43851044	2381741	160000000000	Africa	Arabic	Algerian Dinar	Algiers	0
3	Andorra	77265	468	3000000000	Europe	Catalan	Euro	Andorra la Vella	1
4	Angola	32866272	1246700	100000000000	Africa	Portuguese	Angolan Kwanza	Luanda	0
5	Argentina	45195774	2780400	450000000000	South America	Spanish	Argentine Peso	Buenos Aires	0
6	Armenia	2963243	29743	14000000000	Asia	Armenian	Armenian Dram	Yerevan	0
7	Australia	25499884	7692024	1400000000000	Oceania	English	Australian Dollar	Canberra	1
8	Austria	9006398	83871	480000000000	Europe	German	Euro	Vienna	1
9	Azerbaijan	10139177	86600	48000000000	Asia	Azerbaijani	Azerbaijani Manat	Baku	0
10	Bahamas	393244	13943	12000000000	North America	English	Bahamian Dollar	Nassau	1
11	Bahrain	1701575	765	38000000000	Asia	Arabic	Bahraini Dinar	Manama	1
12	Bangladesh	164689383	147570	350000000000	Asia	Bengali	Bangladeshi Taka	Dhaka	0
13	Barbados	287375	430	5000000000	North America	English	Barbadian Dollar	Bridgetown	1
14	Belarus	9449323	207600	60000000000	Europe	Belarusian	Belarusian Ruble	Minsk	0
15	Belgium	11589623	30528	550000000000	Europe	Dutch	Euro	Brussels	1
16	Belize	397628	22966	2000000000	North America	English	Belize Dollar	Belmopan	0
17	Benin	12123200	112622	15000000000	Africa	French	West African CFA Franc	Porto-Novo	0
18	Bhutan	771608	38394	2500000000	Asia	Dzongkha	Bhutanese Ngultrum	Thimphu	0
19	Bolivia	11673021	1098581	40000000000	South America	Spanish	Bolivian Boliviano	Sucre	0
20	Bosnia and Herzegovina	3280819	51129	20000000000	Europe	Bosnian	Bosnia and Herzegovina Convertible Mark	Sarajevo	0
21	Botswana	2351627	581730	18000000000	Africa	English	Botswana Pula	Gaborone	0
22	Brazil	212559417	8515767	1800000000000	South America	Portuguese	Brazilian Real	Brasilia	0
23	Brunei	437479	5765	12000000000	Asia	Malay	Brunei Dollar	Bandar Seri Begawan	1
24	Bulgaria	6948445	110879	70000000000	Europe	Bulgarian	Bulgarian Lev	Sofia	1

25	Burkina Faso	20903273	274200	16000000000	Africa	French	West African CFA Franc	Ouagadougou	0
26	Burundi	11890784	27834	3000000000	Africa	Kirundi	Burundian Franc	Bujumbura	0
27	Cabo Verde	555987	4033	2000000000	Africa	Portuguese	Cape Verdean Escudo	Praia	0
28	Cambodia	16718965	181035	27000000000	Asia	Khmer	Cambodian Riel	Phnom Penh	0
29	Cameroon	26545863	475442	40000000000	Africa	French	Central African CFA Franc	Yaounde	0
30	Canada	37742154	9984670	1700000000000	North America	English	Canadian Dollar	Ottawa	1
31	Central African Republic	4829767	622984	2000000000	Africa	French	Central African CFA Franc	Bangui	0
32	Chad	16425864	1284000	11000000000	Africa	French	Central African CFA Franc	N'Djamena	0
33	Chile	19116209	756102	280000000000	South America	Spanish	Chilean Peso	Santiago	1
34	China	1402112000	9596961	18000000000000	Asia	Mandarin	Chinese Yuan	Beijing	0
35	Colombia	50882891	1141748	320000000000	South America	Spanish	Colombian Peso	Bogota	0
36	Comoros	869601	1862	1200000000	Africa	Comorian	Comorian Franc	Moroni	0
37	Congo	5518087	342000	11000000000	Africa	French	Central African CFA Franc	Brazzaville	0
38	Costa Rica	5094118	51100	65000000000	North America	Spanish	Costa Rican Colon	San Jose	1
39	Croatia	4105267	56594	60000000000	Europe	Croatian	Croatian Kuna	Zagreb	1
40	Cuba	11326616	109884	100000000000	North America	Spanish	Cuban Peso	Havana	0
41	Cyprus	1207359	9251	25000000000	Europe	Greek	Euro	Nicosia	1
42	Czech Republic	10708981	78865	250000000000	Europe	Czech	Czech Koruna	Prague	1
43	Denmark	5792202	43094	350000000000	Europe	Danish	Danish Krone	Copenhagen	1
44	Djibouti	988000	23200	3000000000	Africa	French	Djiboutian Franc	Djibouti	0
45	Dominica	71986	751	500000000	North America	English	East Caribbean Dollar	Roseau	1
46	Dominican Republic	10847910	48671	89000000000	North America	Spanish	Dominican Peso	Santo Domingo	0
47	Ecuador	17643054	276841	110000000000	South America	Spanish	United States Dollar	Quito	0
48	Egypt	102334404	1002450	400000000000	Africa	Arabic	Egyptian Pound	Cairo	0
49	El Salvador	6486201	21041	27000000000	North America	Spanish	United States Dollar	San Salvador	0
50	Equatorial Guinea	1402985	28051	10000000000	Africa	Spanish	Central African CFA Franc	Malabo	0
51	Eritrea	3546421	117600	2000000000	Africa	Tigrinya	Eritrean Nakfa	Asmara	0
52	Estonia	1326535	45227	31000000000	Europe	Estonian	Euro	Tallinn	1
53	Eswatini	1160164	17364	4000000000	Africa	Swazi	Swazi Lilangeni	Mbabane	0
54	Ethiopia	114963588	1104300	110000000000	Africa	Amharic	Ethiopian Birr	Addis Ababa	0

MetaData of DataFrame

In [98]:
```
df.head(3)
```

Out[98]:

	Country	Population	Area	GDP	Continent	Language	Currency	Capital	Category
0	Afghanistan	38928346	652230	20000000000	Asia	Pashto	Afghan Afghani	Kabul	0
1	Albania	2877797	28748	17000000000	Europe	Albanian	Albanian Lek	Tirana	0
2	Algeria	43851044	2381741	160000000000	Africa	Arabic	Algerian Dinar	Algiers	0

In [99]:
```
df.tail(3)
```

Out[99]:

	Country	Population	Area	GDP	Continent	Language	Currency	Capital	Category
52	Estonia	1326535	45227	31000000000	Europe	Estonian	Euro	Tallinn	1
53	Eswatini	1160164	17364	4000000000	Africa	Swazi	Swazi Lilangeni	Mbabane	0
54	Ethiopia	114963588	1104300	110000000000	Africa	Amharic	Ethiopian Birr	Addis Ababa	0

In [100…
```
df.shape
```

Out[100… (55, 9)

In [101…
```
df.columns
```

Out[101…
```
Index(['Country', 'Population', 'Area', 'GDP', 'Continent', 'Language',
       'Currency', 'Capital', 'Category'],
      dtype='object')
```

In [102…
```
df.dtypes
```

Out[102…
```
Country       object
Population     int64
Area           int64
GDP            int64
Continent     object
Language      object
Currency      object
Capital       object
Category       int64
dtype: object
```

In [103…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55 entries, 0 to 54
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Country     55 non-null     object
 1   Population  55 non-null     int64 
 2   Area        55 non-null     int64 
 3   GDP         55 non-null     int64 
 4   Continent   55 non-null     object
 5   Language    55 non-null     object
 6   Currency    55 non-null     object
 7   Capital     55 non-null     object
 8   Category    55 non-null     int64 
dtypes: int64(4), object(5)
memory usage: 4.0+ KB
```

Descriptive Statistics

In [104…
```
# this is for numerical column features
df.describe()
```

Out[104…

	Population	Area	GDP	Category
count	5.500000e+01	5.500000e+01	5.500000e+01	55.000000
mean	4.660320e+07	9.754245e+05	5.042945e+11	0.327273
std	1.903955e+08	2.336911e+06	2.432775e+12	0.473542
min	7.198600e+04	4.300000e+02	5.000000e+08	0.000000
25%	1.552280e+06	2.794250e+04	1.100000e+10	0.000000
50%	9.006398e+06	8.660000e+04	3.100000e+10	0.000000
75%	2.000974e+07	6.376070e+05	1.350000e+11	1.000000
max	1.402112e+09	9.984670e+06	1.800000e+13	1.000000

In [105…
```
# for categooraical/object features/columns
df.describe(include = 'object')
```

Out[105…

	Country	Continent	Language	Currency	Capital
count	55	55	55	55	55
unique	55	6	30	45	55
top	Afghanistan	Africa	Spanish	Central African CFA Franc	Kabul
freq	1	18	10	5	1

In [106…
```
df['Language'].value_counts()
```

Out[106…
```
Language
Spanish        10
English         7
French          7
Arabic          3
Portuguese      3
Pashto          1
Greek           1
Mandarin        1
Comorian        1
Croatian        1
Tigrinya        1
Czech           1
Danish          1
Kirundi         1
Estonian        1
Swazi           1
Khmer           1
Bosnian         1
Bulgarian       1
Malay           1
Albanian        1
Dzongkha        1
Dutch           1
Belarusian      1
Bengali         1
Azerbaijani     1
German          1
Armenian        1
Catalan         1
Amharic         1
Name: count, dtype: int64
```

In [107…
```
df['Currency'].value_counts()
```

Out[107…

```
Currency
Central African CFA Franc                  5
Euro                                       5
United States Dollar                       2
West African CFA Franc                     2
Afghan Afghani                             1
Croatian Kuna                              1
Canadian Dollar                            1
Chilean Peso                               1
Chinese Yuan                               1
Colombian Peso                             1
Comorian Franc                             1
Costa Rican Colon                          1
Czech Koruna                               1
Cuban Peso                                 1
Cape Verdean Escudo                        1
Danish Krone                               1
Djiboutian Franc                           1
East Caribbean Dollar                      1
Dominican Peso                             1
Egyptian Pound                             1
Eritrean Nakfa                             1
Swazi Lilangeni                            1
Cambodian Riel                             1
Bulgarian Lev                              1
Burundian Franc                            1
Bangladeshi Taka                           1
Algerian Dinar                             1
Angolan Kwanza                             1
Argentine Peso                             1
Armenian Dram                              1
Australian Dollar                          1
Azerbaijani Manat                          1
Bahamian Dollar                            1
Bahraini Dinar                             1
Barbadian Dollar                           1
Albanian Lek                               1
Belarusian Ruble                           1
Belize Dollar                              1
Bhutanese Ngultrum                         1
Bolivian Boliviano                         1
Bosnia and Herzegovina Convertible Mark    1
Botswana Pula                              1
Brazilian Real                             1
Brunei Dollar                              1
Ethiopian Birr                             1
Name: count, dtype: int64
```

In [108…

```
df['Currency'].value_counts(normalize = True)*100
```

Out[108…

```
Currency
Central African CFA Franc                    9.090909
Euro                                         9.090909
United States Dollar                         3.636364
West African CFA Franc                       3.636364
Afghan Afghani                               1.818182
Croatian Kuna                                1.818182
Canadian Dollar                              1.818182
Chilean Peso                                 1.818182
Chinese Yuan                                 1.818182
Colombian Peso                               1.818182
Comorian Franc                               1.818182
Costa Rican Colon                            1.818182
Czech Koruna                                 1.818182
Cuban Peso                                   1.818182
Cape Verdean Escudo                          1.818182
Danish Krone                                 1.818182
Djiboutian Franc                             1.818182
East Caribbean Dollar                        1.818182
Dominican Peso                               1.818182
Egyptian Pound                               1.818182
Eritrean Nakfa                               1.818182
Swazi Lilangeni                              1.818182
Cambodian Riel                               1.818182
Bulgarian Lev                                1.818182
Burundian Franc                              1.818182
Bangladeshi Taka                             1.818182
Algerian Dinar                               1.818182
Angolan Kwanza                               1.818182
Argentine Peso                               1.818182
Armenian Dram                                1.818182
Australian Dollar                            1.818182
Azerbaijani Manat                            1.818182
Bahamian Dollar                              1.818182
Bahraini Dinar                               1.818182
Barbadian Dollar                             1.818182
Albanian Lek                                 1.818182
Belarusian Ruble                             1.818182
Belize Dollar                                1.818182
Bhutanese Ngultrum                           1.818182
Bolivian Boliviano                           1.818182
Bosnia and Herzegovina Convertible Mark      1.818182
Botswana Pula                                1.818182
Brazilian Real                               1.818182
Brunei Dollar                                1.818182
Ethiopian Birr                               1.818182
Name: proportion, dtype: float64
```

Querying a dataset or DataFrame

In [109…

```
# Access first 20 Rows and 4 Columns of the dataFrame
df.iloc[0:20, 0:4]
```

Out[109…

	Country	Population	Area	GDP
0	Afghanistan	38928346	652230	20000000000
1	Albania	2877797	28748	17000000000
2	Algeria	43851044	2381741	160000000000
3	Andorra	77265	468	3000000000
4	Angola	32866272	1246700	100000000000
5	Argentina	45195774	2780400	450000000000
6	Armenia	2963243	29743	14000000000
7	Australia	25499884	7692024	1400000000000
8	Austria	9006398	83871	480000000000
9	Azerbaijan	10139177	86600	48000000000
10	Bahamas	393244	13943	12000000000
11	Bahrain	1701575	765	38000000000
12	Bangladesh	164689383	147570	350000000000
13	Barbados	287375	430	5000000000
14	Belarus	9449323	207600	60000000000
15	Belgium	11589623	30528	550000000000
16	Belize	397628	22966	2000000000
17	Benin	12123200	112622	15000000000
18	Bhutan	771608	38394	2500000000
19	Bolivia	11673021	1098581	40000000000

In [110…

```
# Access frst 30 rows and first 4 columns of dataFrame
df.loc[0:30, 'Country':'GDP']
```

Out[110…

	Country	Population	Area	GDP
0	Afghanistan	38928346	652230	20000000000
1	Albania	2877797	28748	17000000000
2	Algeria	43851044	2381741	160000000000
3	Andorra	77265	468	3000000000
4	Angola	32866272	1246700	100000000000
5	Argentina	45195774	2780400	450000000000
6	Armenia	2963243	29743	14000000000
7	Australia	25499884	7692024	1400000000000
8	Austria	9006398	83871	480000000000
9	Azerbaijan	10139177	86600	48000000000
10	Bahamas	393244	13943	12000000000
11	Bahrain	1701575	765	38000000000
12	Bangladesh	164689383	147570	350000000000
13	Barbados	287375	430	5000000000
14	Belarus	9449323	207600	60000000000
15	Belgium	11589623	30528	550000000000
16	Belize	397628	22966	2000000000
17	Benin	12123200	112622	15000000000
18	Bhutan	771608	38394	2500000000
19	Bolivia	11673021	1098581	40000000000
20	Bosnia and Herzegovina	3280819	51129	20000000000
21	Botswana	2351627	581730	18000000000
22	Brazil	212559417	8515767	1800000000000
23	Brunei	437479	5765	12000000000
24	Bulgaria	6948445	110879	70000000000
25	Burkina Faso	20903273	274200	16000000000
26	Burundi	11890784	27834	3000000000
27	Cabo Verde	555987	4033	2000000000
28	Cambodia	16718965	181035	27000000000
29	Cameroon	26545863	475442	40000000000
30	Canada	37742154	9984670	1700000000000

Country with Maximum GDP

In [111…

```
df[df['GDP'] == df[df['Language'] == 'English']['GDP'].max()]['Country']
```

Out[111…

```
30    Canada
Name: Country, dtype: object
```

Sorting

In [112…

```
# Sort the dataFrame by Country Name Ascendind order
df.sort_values(by= 'Country').head()
```

Out[112…

	Country	Population	Area	GDP	Continent	Language	Currency	Capital	Category
0	Afghanistan	38928346	652230	20000000000	Asia	Pashto	Afghan Afghani	Kabul	0
1	Albania	2877797	28748	17000000000	Europe	Albanian	Albanian Lek	Tirana	0
2	Algeria	43851044	2381741	160000000000	Africa	Arabic	Algerian Dinar	Algiers	0
3	Andorra	77265	468	3000000000	Europe	Catalan	Euro	Andorra la Vella	1
4	Angola	32866272	1246700	100000000000	Africa	Portuguese	Angolan Kwanza	Luanda	0

In [113…

```
df.sort_values(by = 'Country', ascending = False).head()
```

Out[113…

	Country	Population	Area	GDP	Continent	Language	Currency	Capital	Category
54	Ethiopia	114963588	1104300	110000000000	Africa	Amharic	Ethiopian Birr	Addis Ababa	0
53	Eswatini	1160164	17364	4000000000	Africa	Swazi	Swazi Lilangeni	Mbabane	0
52	Estonia	1326535	45227	31000000000	Europe	Estonian	Euro	Tallinn	1
51	Eritrea	3546421	117600	2000000000	Africa	Tigrinya	Eritrean Nakfa	Asmara	0
50	Equatorial Guinea	1402985	28051	10000000000	Africa	Spanish	Central African CFA Franc	Malabo	0

In [114…
```
df['Continent'].unique()
```

Out[114…
```
array(['Asia', 'Europe', 'Africa', 'South America', 'Oceania',
       'North America'], dtype=object)
```

Replacing Values in columns

In [115…
```
d = {'Asia' : 0 , 'Europe' : 1, 'Africa' : 2 , 'South America' : 3 , 'Oceania' : 4 ,'North America' : 5 }
print('Before Replacement : ')
print(df['Continent'].head(20))
print('After Replacement : ')
df['Continent'] = df['Continent'].map(d)
print(df['Continent'].head(20))
```

```
Before Replacement :
0              Asia
1            Europe
2            Africa
3            Europe
4            Africa
5     South America
6              Asia
7           Oceania
8            Europe
9              Asia
10    North America
11             Asia
12             Asia
13    North America
14           Europe
15           Europe
16    North America
17           Africa
18             Asia
19    South America
Name: Continent, dtype: object
After Replacement :
0     0
1     1
2     2
3     1
4     2
5     3
6     0
7     4
8     1
9     0
10    5
11    0
12    0
13    5
14    1
15    1
16    5
17    2
18    0
19    3
Name: Continent, dtype: int64
```

Grouping columns value-wise:

In [116…
```
# one column is categorical and one is numenrical
# here I want to know the GDP statastics as per Develpement in Countries '
# 0 Non Developed Countries
# 1 Developed Countries
df.groupby(by = 'Category')['GDP'].describe()
```

Out[116…

Category	count	mean	std	min	25%	50%	75%	max
0	37.0	6.055324e+11	2.954993e+12	1.200000e+09	1.000000e+10	2.000000e+10	1.000000e+11	1.800000e+13
1	18.0	2.961944e+11	4.900910e+11	5.000000e+08	1.525000e+10	6.250000e+10	3.325000e+11	1.700000e+12

In [117…

```
# categorical valuue
# use cross tab
# find the delevoped and non developed countries GDP
pd.crosstab(df['Category'],df['GDP'], normalize=True)
```

Out[117…

Category	GDP	500000000	1200000000	2000000000	2500000000	3000000000	4000000000	5000000000	10000000000	11000000000	1200
0		0.000000	0.018182	0.072727	0.018182	0.036364	0.018182	0.000000	0.018182	0.036364	
1		0.018182	0.000000	0.000000	0.000000	0.018182	0.000000	0.018182	0.000000	0.000000	

2 rows × 41 columns

In [118…

```
# more than one value need to check against target column
# use pivot table
# 0 means undeveloped Countries
df.pivot_table(['GDP', 'Area'],['Category'],
               aggfunc='mean')
```

Out[118…

Category	Area	GDP
0	9.374060e+05	6.055324e+11
1	1.053574e+06	2.961944e+11

In [119…

```
df.pivot_table(['GDP' , 'Area'],['Category'], aggfunc='max')
```

Out[119…

Category	Area	GDP
0	9596961	18000000000000
1	9984670	1700000000000

Data Visualization

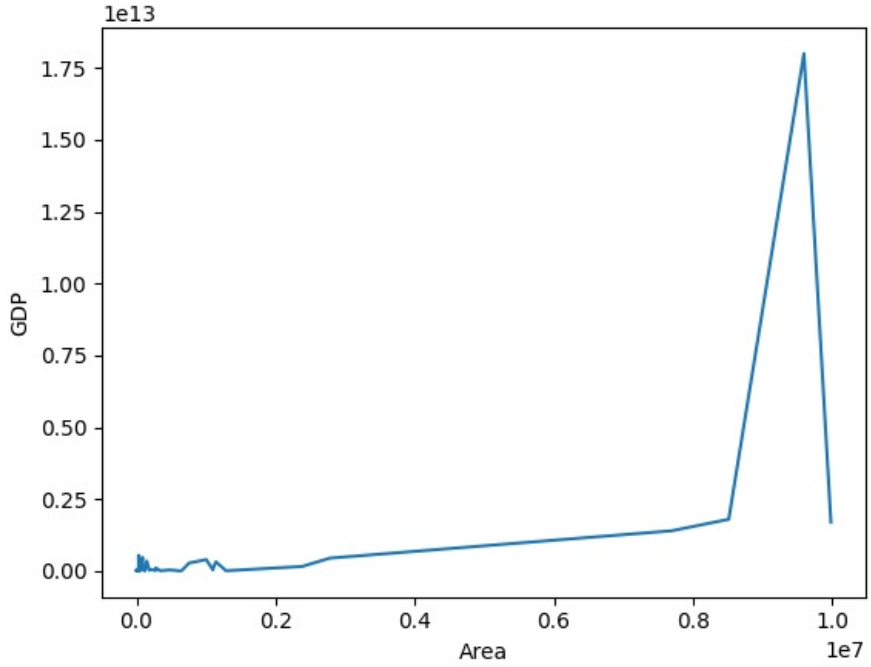
Line Plot

In [120…

```
sns.lineplot(x = 'Area', y = 'GDP', data = df)
```

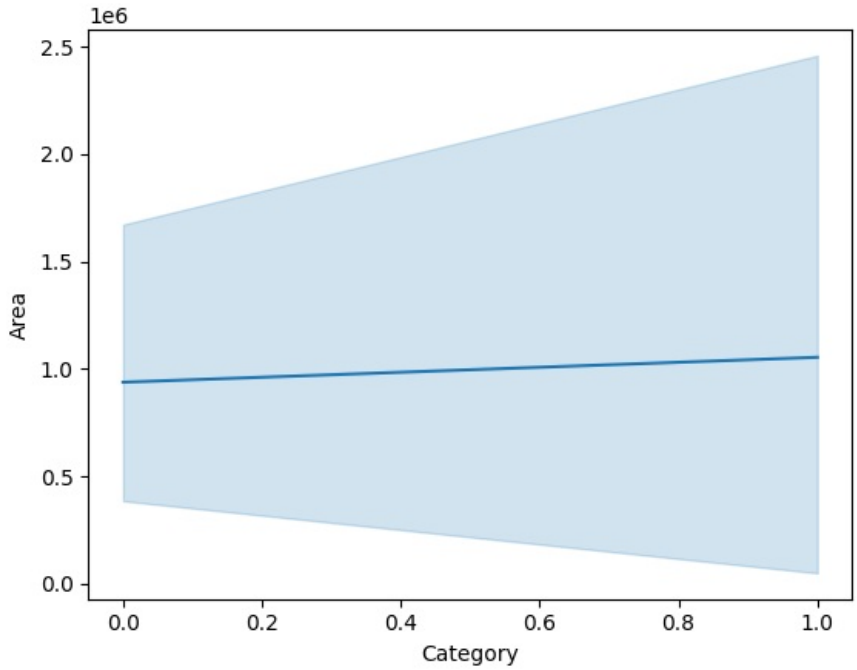
Out[120…

```
<Axes: xlabel='Area', ylabel='GDP'>
```



In [121…
```
# cateogory 0 means undeveloped country
sns.lineplot(x = 'Category' , y = 'Area', data = df)
```

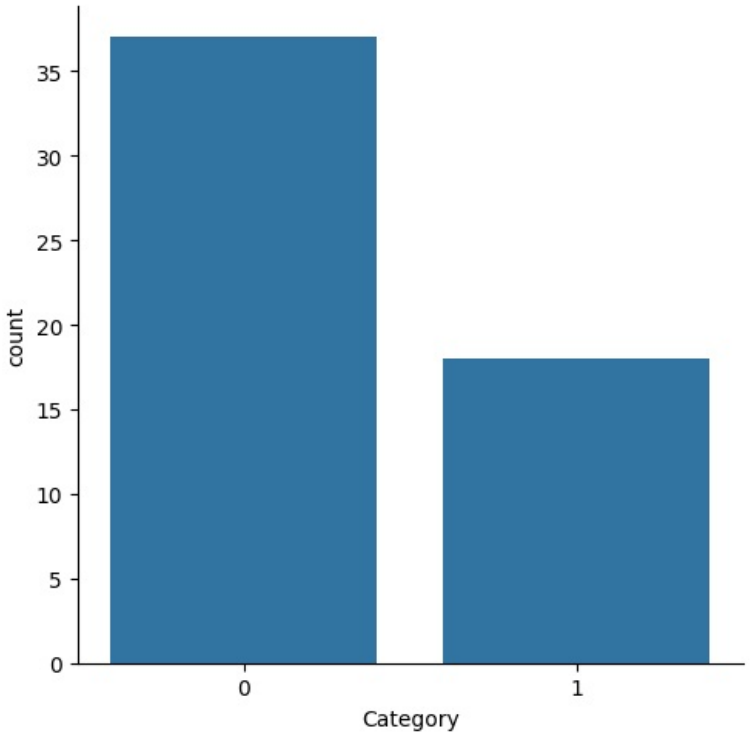
Out[121… <Axes: xlabel='Category', ylabel='Area'>



Bar Graph

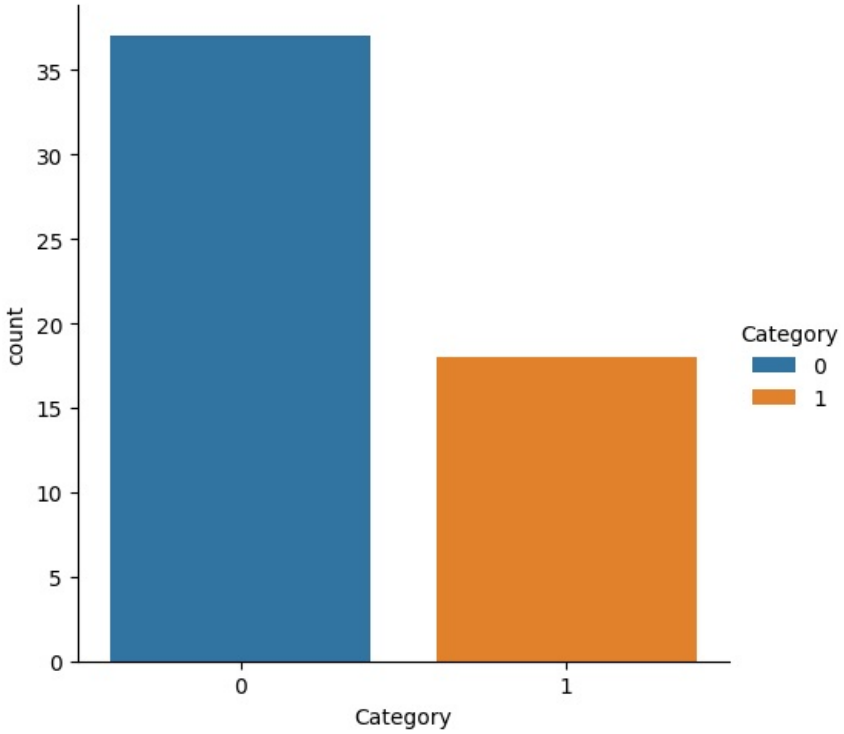
In [122…
```
sns.catplot(x = 'Category' , data = df, kind = 'count')
```

Out[122…
```
<seaborn.axisgrid.FacetGrid at 0x1854e92fb90>
```



In [123…
```
# 0 means Underdeveloped Country
sns.catplot(x = 'Category', data = df, kind='count', hue = 'Category')
```

Out[123…
```
<seaborn.axisgrid.FacetGrid at 0x1854e844470>
```

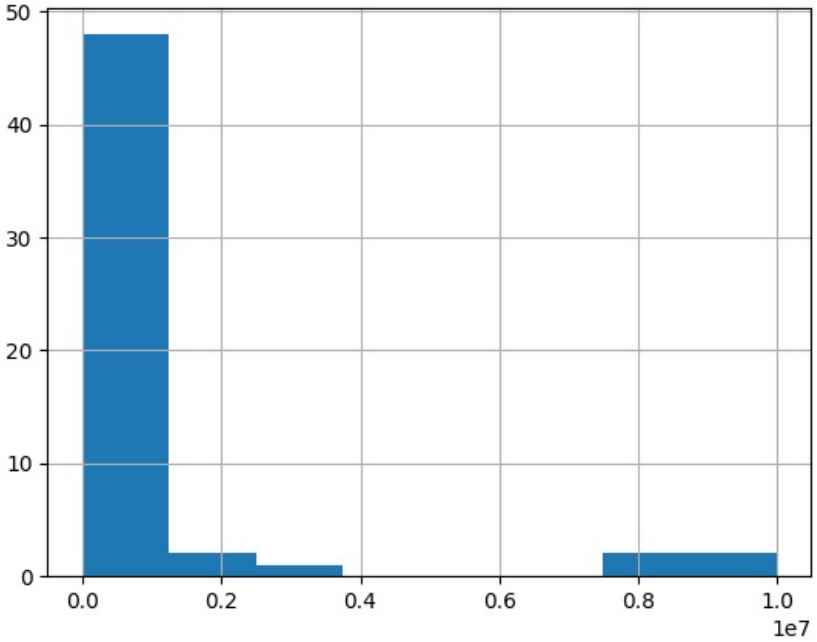


Histogram

Area

In [124…
```
df['Area'].hist(bins=8)
```

Out[124…
```
<Axes: >
```

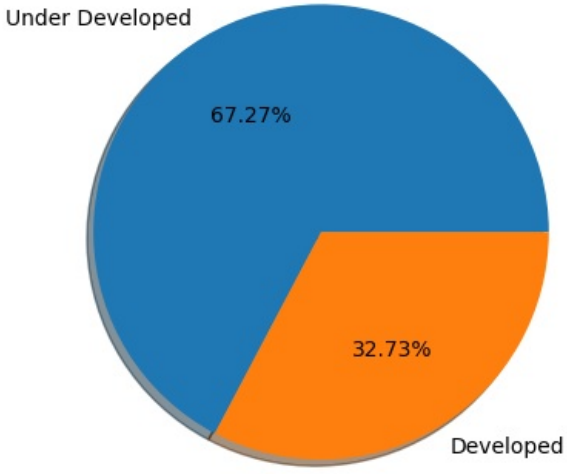


Pie Chart

Developed vs UnderDeveloped Countries

In [125…

```
sizes = df["Category"].value_counts()
sizes
fig1, ax1 = plt.subplots()
ax1.pie(sizes,
        labels=['Under Developed', 'Developed'],
        autopct='%1.2f%%',
        shadow=True)
plt.show()
```



In [ ]: