importing the Necessary Libirary

In [20]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Uploading csv file

In [21]:
```python
df=pd.read_csv("World Economic Classifications v2.CSV")
df
```

Out[21]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
0	Luxembourg	Developed	Advanced	No	Yes	No	1.0	$142,214.00	$127,046.00
1	Singapore	Developing	Advanced	No	No	No	2.0	$127,565.00	$78,115.00
2	Ireland	Developed	Advanced	No	Yes	No	3.0	$126,905.00	$105,362.00
3	Norway	Developed	Advanced	No	No	No	4.0	$114,899.00	$106,594.00
4	Qatar	Developing	Emerging	No	No	Yes	5.0	$114,648.00	$88,046.00
...	...	...	...	...	...	...	...	...	...
200	Syrian Arab Republic (Syria)	Developing	Emerging	No	No	No	NaN	NaN	NaN
201	Taiwan	Developing	Advanced	No	No	No	NaN	NaN	NaN
202	Venezuela (Bolivarian Republic of)	Developing	Emerging	No	No	Yes	NaN	NaN	NaN
203	Yemen	Developing	Emerging	No	No	Yes	NaN	NaN	NaN
204	Andorra	NaN	Advanced	No	No	Yes	NaN	NaN	$41,993.00

205 rows × 9 columns

Data preprocessing

head()

to Display the First 5 rows in a dataset

In [22]:
```python
df.head()
```

Out[22]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
0	Luxembourg	Developed	Advanced	No	Yes	No	1.0	$142,214.00	$127,046.00
1	Singapore	Developing	Advanced	No	No	No	2.0	$127,565.00	$78,115.00
2	Ireland	Developed	Advanced	No	Yes	No	3.0	$126,905.00	$105,362.00
3	Norway	Developed	Advanced	No	No	No	4.0	$114,899.00	$106,594.00
4	Qatar	Developing	Emerging	No	No	Yes	5.0	$114,648.00	$88,046.00

tail()

To display last 5 rows ina dataset

In [23]:
```python
df.tail()
```

Out[23]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
200	Syrian Arab Republic (Syria)	Developing	Emerging	No	No	No	NaN	NaN	NaN
201	Taiwan	Developing	Advanced	No	No	No	NaN	NaN	NaN
202	Venezuela (Bolivarian Republic of)	Developing	Emerging	No	No	Yes	NaN	NaN	NaN
203	Yemen	Developing	Emerging	No	No	Yes	NaN	NaN	NaN
204	Andorra	NaN	Advanced	No	No	Yes	NaN	NaN	$41,993.00

shape

it shows the Number of Rows & columns in a dataset

In [24]:
```python
df.shape
```

Out[24]:
```
(205, 9)
```

Columns

it shows the each Columns

In [25]:
```python
df.columns
```

Out[25]:
```
Index(['country_name', 'un_class_2014', 'imf_class_2023', 'g7', 'eu_member',
       'fuel_exp_country', 'wealth_rank', 'gdp_ppp_2022', 'gdp_pc_2022'],
      dtype='object')
```

dtypes

it shows The the type of each column

In [26]:
```python
df.dtypes
```

Out[26]:
```
country_name         object
un_class_2014        object
imf_class_2023       object
g7                   object
eu_member            object
fuel_exp_country     object
wealth_rank         float64
gdp_ppp_2022         object
gdp_pc_2022          object
dtype: object
```

unique()

It shows the unique() value of Specific Columns

In [27]:
```python
df['gdp_ppp_2022'].unique()
```

Out[27]:
```
array(['$142,214.00', '$127,565.00', '$126,905.00', '$114,899.00',
       '$114,648.00', '$87,729.00', '$83,598.00', '$76,399.00',
       '$74,005.00', '$69,577.00', '$69,275.00', '$69,081.00',
       '$69,049.00', '$67,936.00', '$65,027.00', '$64,578.00',
       '$63,150.00', '$62,625.00', '$61,228.00', '$59,065.00',
       '$59,027.00', '$58,400.00', '$58,056.00', '$55,928.00',
       '$55,493.00', '$55,344.00', '$54,603.00', '$51,967.00',
       '$51,865.00', '$50,070.00', '$50,032.00', '$49,946.00',
       '$49,931.00', '$49,509.00', '$48,397.00', '$46,697.00',
       '$45,825.00', '$45,573.00', '$43,269.00', '$41,907.00',
       '$41,888.00', '$41,724.00', '$41,452.00', '$40,642.00',
       '$40,380.00', '$40,379.00', '$39,956.00', '$39,280.00',
       '$37,459.00', '$37,274.00', '$36,835.00', '$36,485.00',
       '$35,228.00', '$34,052.00', '$33,582.00', '$33,434.00',
       '$30,810.00', '$30,209.00', '$28,842.00', '$27,778.00',
       '$26,984.00', '$26,906.00', '$26,505.00', '$25,337.00',
       '$24,932.00', '$24,772.00', '$23,911.00', '$23,375.00',
       '$22,834.00', '$22,591.00', '$21,512.00', '$21,476.00',
       '$20,672.00', '$20,377.00', '$20,287.00', '$20,162.00',
       '$20,113.00', '$18,942.00', '$18,552.00', '$18,323.00',
       '$18,075.00', '$17,837.00', '$17,822.00', '$17,764.00',
       '$17,756.00', '$17,620.00', '$17,396.00', '$17,207.00',
       '$16,987.00', '$16,471.00', '$15,977.00', '$15,905.00',
       '$15,238.00', '$15,091.00', '$15,048.00', '$14,653.00',
       '$14,405.00', '$14,230.00', '$14,125.00', '$13,573.00',
       '$13,457.00', '$13,210.00', '$12,822.00', '$12,671.00',
       '$12,490.00', '$11,822.00', '$11,451.00', '$11,206.00',
       '$11,096.00', '$11,003.00', '$10,862.00', '$10,818.00',
       '$10,782.00', '$10,133.00', '$9,684.00', '$9,533.00', '$9,519.00',
       '$9,384.00', '$9,083.00', '$8,379.00', '$7,395.00', '$7,228.00',
       '$6,974.00', '$6,875.00', '$6,757.00', '$6,741.00', '$6,538.00',
       '$6,498.00', '$6,437.00', '$6,424.00', '$6,133.00', '$6,041.00',
       '$5,860.00', '$5,764.00', '$5,421.00', '$4,885.00', '$4,870.00',
       '$4,828.00', '$4,738.00', '$4,725.00', '$4,447.00', '$4,408.00',
       '$4,216.00', '$4,209.00', '$4,056.00', '$3,894.00', '$3,855.00',
       '$3,832.00', '$3,791.00', '$3,305.00', '$3,289.00', '$3,187.00',
       '$3,097.00', '$2,812.00', '$2,792.00', '$2,695.00', '$2,694.00',
       '$2,654.00', '$2,608.00', '$2,546.00', '$2,531.00', '$2,517.00',
       '$2,510.00', '$2,365.00', '$2,190.00', '$1,931.00', '$1,774.00',
       '$1,732.00', '$1,725.00', '$1,668.00', '$1,505.00', '$1,468.00',
       '$1,337.00', '$967.00', '$836.00', nan], dtype=object)
```

nuniuque() it shows The Total Number of unique value in whole data frame

In [28]:
```python
df.nunique()
```

Out[28]:
```
country_name        205
un_class_2014         3
imf_class_2023        3
g7                    2
eu_member             2
fuel_exp_country      2
wealth_rank         176
gdp_ppp_2022        175
gdp_pc_2022         175
dtype: int64
```

describe()

Its count the mean,median,mode,standardeviation,Maximum & Minum Value

In [29]:
```python
df.describe()
```

Out[29]:

	wealth_rank
count	178.000000
mean	89.724719
std	51.818540
min	1.000000
25%	45.250000
50%	89.500000
75%	133.750000
max	177.000000

value_counts()

Its Show all The unique Value with their Counts

In [30]:
```python
df['gdp_pc_2022'].value_counts()
```

Out[30]:
```
gdp_pc_2022
$833.00        2
$127,046.00    1
$5,937.00      1
$4,040.00      1
$3,499.00      1
              ..
$9,721.00      1
$9,926.00      1
$13,904.00     1
$18,745.00     1
$41,993.00     1
Name: count, Length: 175, dtype: int64
```

its shows Information of each column

In [31]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   country_name      205 non-null    object 
 1   un_class_2014     167 non-null    object 
 2   imf_class_2023    193 non-null    object 
 3   g7                205 non-null    object 
 4   eu_member         205 non-null    object 
 5   fuel_exp_country  205 non-null    object 
 6   wealth_rank       178 non-null    float64
 7   gdp_ppp_2022      175 non-null    object 
 8   gdp_pc_2022       176 non-null    object 
dtypes: float64(1), object(8)
memory usage: 14.5+ KB
```

isnull()

its Shows The Null value

In [32]:
```python
df.isnull()
```

Out[32]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
200	False	False	False	False	False	False	True	True	True
201	False	False	False	False	False	False	True	True	True
202	False	False	False	False	False	False	True	True	True
203	False	False	False	False	False	False	True	True	True
204	False	True	False	False	False	False	True	True	False

205 rows × 9 columns

notnull()

its shows THe Notnull Values

In [33]:
```python
df.notnull()
```

Out[33]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
0	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...
200	True	True	True	True	True	True	False	False	False
201	True	True	True	True	True	True	False	False	False
202	True	True	True	True	True	True	False	False	False
203	True	True	True	True	True	True	False	False	False
204	True	False	True	True	True	True	False	False	True

205 rows × 9 columns

info()

In [34]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   country_name      205 non-null    object 
 1   un_class_2014     167 non-null    object 
 2   imf_class_2023    193 non-null    object 
 3   g7                205 non-null    object 
 4   eu_member         205 non-null    object 
 5   fuel_exp_country  205 non-null    object 
 6   wealth_rank       178 non-null    float64
 7   gdp_ppp_2022      175 non-null    object 
 8   gdp_pc_2022       176 non-null    object 
dtypes: float64(1), object(8)
memory usage: 14.5+ KB
```

its fill the 0 if any empty row

In [35]:
```python
df.fillna(0)
```

Out[35]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
0	Luxembourg	Developed	Advanced	No	Yes	No	1.0	$142,214.00	$127,046.00
1	Singapore	Developing	Advanced	No	No	No	2.0	$127,565.00	$78,115.00
2	Ireland	Developed	Advanced	No	Yes	No	3.0	$126,905.00	$105,362.00
3	Norway	Developed	Advanced	No	No	No	4.0	$114,899.00	$106,594.00
4	Qatar	Developing	Emerging	No	No	Yes	5.0	$114,648.00	$88,046.00
...	...	...	...	...	...	...	...	...	...
200	Syrian Arab Republic (Syria)	Developing	Emerging	No	No	No	0.0	0	0
201	Taiwan	Developing	Advanced	No	No	No	0.0	0	0
202	Venezuela (Bolivarian Republic of)	Developing	Emerging	No	No	Yes	0.0	0	0
203	Yemen	Developing	Emerging	No	No	Yes	0.0	0	0
204	Andorra	0	Advanced	No	No	Yes	0.0	0	$41,993.00

205 rows × 9 columns

In [36]:
```python
df.bfill()
```

Out[36]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
0	Luxembourg	Developed	Advanced	No	Yes	No	1.0	$142,214.00	$127,046.00
1	Singapore	Developing	Advanced	No	No	No	2.0	$127,565.00	$78,115.00
2	Ireland	Developed	Advanced	No	Yes	No	3.0	$126,905.00	$105,362.00
3	Norway	Developed	Advanced	No	No	No	4.0	$114,899.00	$106,594.00
4	Qatar	Developing	Emerging	No	No	Yes	5.0	$114,648.00	$88,046.00
...	...	...	...	...	...	...	...	...	...
200	Syrian Arab Republic (Syria)	Developing	Emerging	No	No	No	NaN	NaN	$41,993.00
201	Taiwan	Developing	Advanced	No	No	No	NaN	NaN	$41,993.00
202	Venezuela (Bolivarian Republic of)	Developing	Emerging	No	No	Yes	NaN	NaN	$41,993.00
203	Yemen	Developing	Emerging	No	No	Yes	NaN	NaN	$41,993.00
204	Andorra	NaN	Advanced	No	No	Yes	NaN	NaN	$41,993.00

205 rows × 9 columns

Its Fill The Missing Values By Comparing Both Front and Back datas

In [37]:
```python
df.interpolate(method='linear')
```

```
C:\Users\Muthu\AppData\Local\Temp\ipykernel_1668\1321276365.py:1: FutureWarning: DataFrame.interpolate with object dtype is deprecated and will raise in a future version. Call obj.infer_objects(copy=False) before interpolating instead.
  df.interpolate(method='linear')
```

Out[37]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
0	Luxembourg	Developed	Advanced	No	Yes	No	1.0	$142,214.00	$127,046.00
1	Singapore	Developing	Advanced	No	No	No	2.0	$127,565.00	$78,115.00
2	Ireland	Developed	Advanced	No	Yes	No	3.0	$126,905.00	$105,362.00
3	Norway	Developed	Advanced	No	No	No	4.0	$114,899.00	$106,594.00
4	Qatar	Developing	Emerging	No	No	Yes	5.0	$114,648.00	$88,046.00
...	...	...	...	...	...	...	...	...	...
200	Syrian Arab Republic (Syria)	Developing	Emerging	No	No	No	177.0	NaN	NaN
201	Taiwan	Developing	Advanced	No	No	No	177.0	NaN	NaN
202	Venezuela (Bolivarian Republic of)	Developing	Emerging	No	No	Yes	177.0	NaN	NaN
203	Yemen	Developing	Emerging	No	No	Yes	177.0	NaN	NaN
204	Andorra	NaN	Advanced	No	No	Yes	177.0	NaN	$41,993.00

205 rows × 9 columns

In [38]:
```python
print(df.dropna(axis=0,inplace=True))
```

```
None
```

its ordering The Column by its Alphabet

In [39]:
```python
df.groupby('country_name').sum()
```

Out[39]:

country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
Albania	Transition	Emerging	No	No	No	79.0	$18,552.00	$6,643.00
Algeria	Developing	Emerging	No	No	Yes	102.0	$13,210.00	$4,274.00
Angola	Developing	Emerging	No	No	Yes	123.0	$6,974.00	$2,999.00
Aoeteroa (New Zealand)	Developed	Advanced	No	No	No	28.0	$51,967.00	$47,680.00
Argentina	Developing	Emerging	No	No	No	63.0	$26,505.00	$13,904.00
...	...	...	...	...	...	...	...	...
Uruguay	Developing	Emerging	No	No	No	59.0	$28,842.00	$20,795.00
Uzbekistan	Transition	Emerging	No	No	Yes	116.0	$9,533.00	$2,322.00
Viet Nam (Vietnam)	Developing	Emerging	No	No	Yes	101.0	$13,457.00	$4,164.00
Zambia	Developing	Emerging	No	No	No	147.0	$3,894.00	$1,488.00
Zimbabwe	Developing	Emerging	No	No	No	162.0	$2,531.00	$1,267.00

153 rows × 8 columns

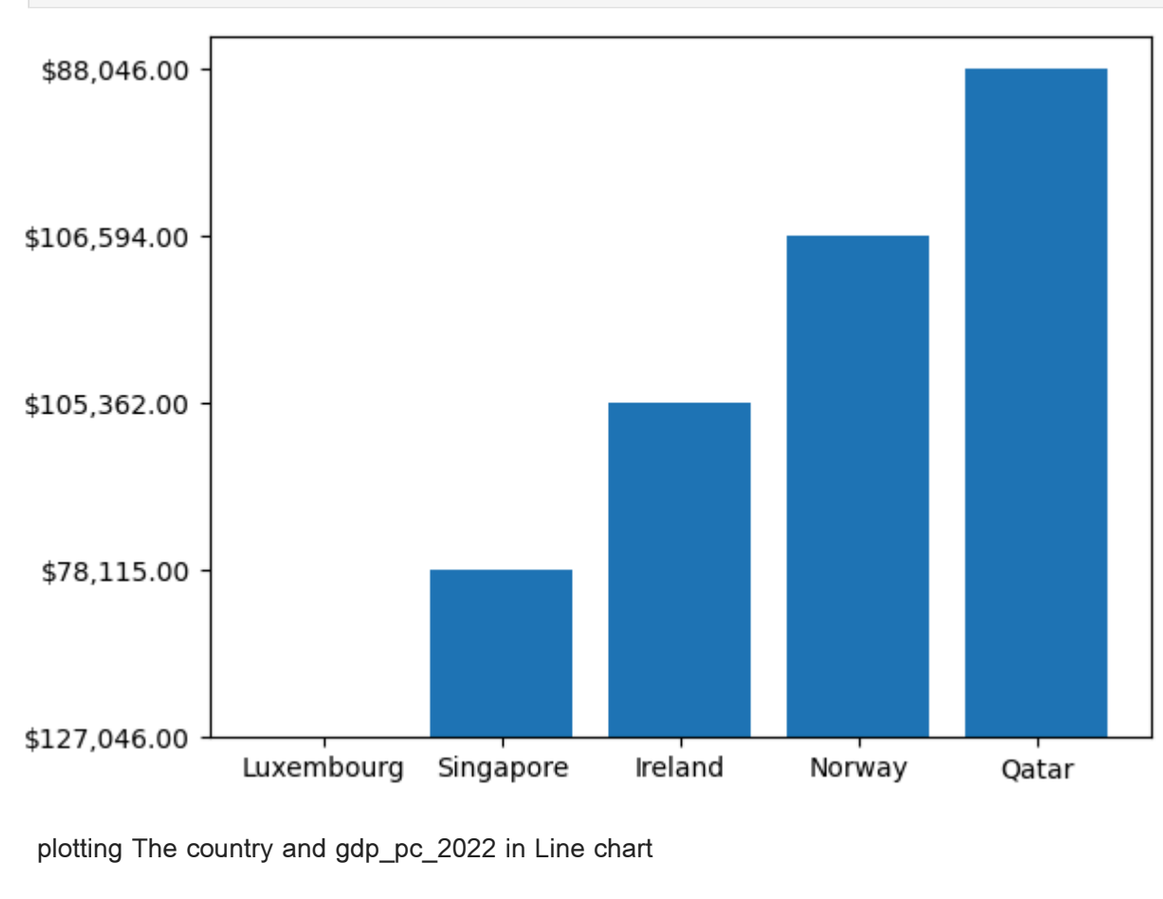
In [40]:
```python
x=df.head()
x
```

Out[40]:

	country_name	un_class_2014	imf_class_2023	g7	eu_member	fuel_exp_country	wealth_rank	gdp_ppp_2022	gdp_pc_2022
0	Luxembourg	Developed	Advanced	No	Yes	No	1.0	$142,214.00	$127,046.00
1	Singapore	Developing	Advanced	No	No	No	2.0	$127,565.00	$78,115.00
2	Ireland	Developed	Advanced	No	Yes	No	3.0	$126,905.00	$105,362.00
3	Norway	Developed	Advanced	No	No	No	4.0	$114,899.00	$106,594.00
4	Qatar	Developing	Emerging	No	No	Yes	5.0	$114,648.00	$88,046.00

plotting The country and gdp_ppp_2022 in bar chart

In [98]:
```python
plt.bar(x.iloc[:,0],x.iloc[:,-1])
plt.show()
```



plotting The country and gdp_pc_2022 in Line chart

In [99]:
```python
plt.plot(x.iloc[:,0],x.iloc[:,-2])
plt.show()
```