A PROJECT REPORT ON

Exploratory analysis

of

World Fact book dataset

BY

Asmita Dhage
Yasaswini Suryadevara

Under The Guidance of

Dr. Vahid Behzadan

OVERVIEW:

For analytical purposes, we have tried to classify all countries of the world into one of three broad categories: developed economies, economies in transition and developing economies. Several countries (in particular the economies in transition) have characteristics that could place them in more than one category; however, for purposes of analysis, the groupings have been made mutually exclusive. Within each broad category, some subgroups are defined based either on population or on ad hoc criteria, such as the subgroup of "major developed economies", which is based on the membership of the Group of Seven. Geographical regions for developing economies are as follows: Africa, East Asia, South Asia, Western Asia etc.
Also we have tried analyzing and visualizing each major category for all the countries.

STATEMENT OF OBJECTIVES:

To use exploratory data analysis to investigate the relationship between two categorical variables. Specifically, to study how the conditional distribution of a categorical response variable changes for different categories of an explanatory variable and to achieve following insights from the data:

1) Evaluating the overall performance of a country depending on major categories such as area, population, GDP, etc.
2) Classifying the countries into developed, developing and under developed categories
3) Ranking the countries based on health, wealth and best to live in

APPROACH:

Task 1: Collecting data set which can summarize all major features required for the final analysis

For this task, we have used the dataset of 264 countries with ~45 characteristics such as GDP, electricity consumption, Internet users, etc. The dataset is available on the https://perso.telecom-paristech.com

```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/deprecation.py:66: DeprecationWarning: Class Imputer is deprecated; Imputer was deprecated in version 0.20 and will be removed in 0.22. Import impute.SimpleImputer from sklearn instead.
  warnings.warn(msg, category=DeprecationWarning)
```

	Country	Area(sq km)	Birth rate(births/1000 population)	Current account balance	Death rate(deaths/1000 population)	Debt - external	Electricity - consumption(kWh)	Electricity - production(kWh)	Exports	GDP	GDP - per capita	GDP - real growth rate(%)	HIV/AIDS - adult prevalence rate(%)	HIV/AIDS - deaths	HIV/AIDS - people living with HIV/AIDS	Highways(km)	Imports	Industrial production growth rate(%)	Infant mor rate(deat live b
1	Afghanistan	647500	47.02	NaN	20.75	8000000000	652200000	540000000	446000000	2.150000e+10	800	7.50	0.01	NaN	NaN	21000	3759000000	NaN	
2	Akrotiri	123	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.925614e+11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	Albania	28748	15.08	-504000000	5.12	1410000000	6760000000	5680000000	552400000	1.746000e+10	4900	5.60	NaN	NaN	NaN	18000	2076000000	3.10	
4	Algeria	2381740	17.13	11900000000	4.60	21900000000	23610000000	25760000000	32160000000	2.123000e+11	6600	6.10	0.10	500	9100	104000	15250000000	6.00	
5	American Samoa	199	23.13	NaN	3.33	NaN	120900000	130000000	30000000	5.000000e+08	8000	NaN	NaN	NaN	NaN	185	123000000	NaN	
6	Andorra	468	9.00	NaN	6.07	NaN	NaN	NaN	58000000	1.900000e+09	26800	2.00	NaN	NaN	NaN	269	1077000000	NaN	
7	Angola	1246700	44.64	-37880000	25.90	10450000000	1587000000	1707000000	12760000000	2.317000e+10	2100	11.70	3.90	21000	240000	51429	4896000000	1.00	
8	Anguilla	102	14.26	NaN	5.43	8800000	42600000	NaN	2600000	1.120000e+08	7500	2.80	NaN	NaN	NaN	105	80900000	3.10	
9	Antarctica	14000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.925614e+11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
10	Antigua and Barbuda	443	17.26	NaN	5.44	231000000	103000000	110800000	689000000	7.500000e+08	11000	3.00	NaN	NaN	NaN	250	692000000	6.00	
11	Argentina	2766890	16.90	5473000000	7.56	157700000000	81650000000	81390000000	33780000000	4.835000e+11	12400	8.30	0.70	1500	130000	215471	22060000000	12.00	
12	Armenia	29800	11.76	-240400000	8.16	905000000	5797000000	6492000000	850000000	1.365000e+10	4600	9.00	0.10	200	2600	8431	1300000000	15.00	
13	Aruba	193	11.26	NaN	6.57	285000000	751200000	807700000	128000000	1.940000e+09	28000	-1.50	NaN	NaN	NaN	800	841000000	NaN	
14	Ashmore and Cartier Islands	5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.925614e+11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
15	Australia	7686850	12.26	-38300000000	7.44	308700000000	195600000000	210300000000	86890000000	6.117000e+11	30700	3.50	0.10	NaN	14000	811603	98100000000	1.90	

Task 2: Cleaning the dataset

In this step we have cleaned the dataset to remove duplicate entries or false entries. Also, we tried to reduce some of the columns from the dataset which are not contributing towards end goal of the project or which are positively co-related to each other. For this task we implemented the techniques mentioned below:

1. **Check for NANs (Imputer)**
2. **Drop/Replace missing data**
3. **Drop a feature**
4. **Taking care of outliers**
5. **Feature engineering (Use of co-relation matrix)**

	Country	Area(sq km)	Birth rate(births/1000 population)	Current account balance	Death rate(deaths/1000 population)	Debt - external	Electricity - consumption(kWh)	Electricity - production(kWh)
1	Afghanistan	647500	47.020000	-8.473897e+08	20.750000	8.000000e+09	6.522000e+08	5.400000e+08
2	Akrotiri	123	22.146667	-8.473897e+08	9.374267	6.317084e+10	8.070919e+10	8.849167e+10
3	Albania	28748	15.080000	-5.040000e+08	5.120000	1.410000e+09	6.760000e+09	5.680000e+09
4	Algeria	2381740	17.130000	1.190000e+10	4.600000	2.190000e+10	2.361000e+10	2.576000e+10
5	American Samoa	199	23.130000	-8.473897e+08	3.330000	6.317084e+10	1.209000e+08	1.300000e+08
6	Andorra	468	9.000000	-8.473897e+08	6.070000	6.317084e+10	8.070919e+10	8.849167e+10
7	Angola	1246700	44.640000	-3.788000e+07	25.900000	1.045000e+10	1.587000e+09	1.707000e+09
8	Anguilla	102	14.260000	-8.473897e+08	5.430000	8.800000e+06	4.260000e+07	8.849167e+10
9	Antarctica	14000000	22.146667	-8.473897e+08	9.374267	6.317084e+10	8.070919e+10	8.849167e+10
10	Antigua and Barbuda	443	17.260000	-8.473897e+08	5.440000	2.310000e+08	1.030000e+08	1.108000e+08
11	Argentina	2766890	16.900000	5.473000e+09	7.560000	1.577000e+11	8.165000e+10	8.139000e+10
12	Armenia	29800	11.760000	-2.404000e+08	8.160000	9.050000e+08	5.797000e+09	6.492000e+09

	Exports	GDP	GDP - per capita	GDP - real growth rate(%)	HIV/AIDS - adult prevalence rate(%)	HIV/AIDS - deaths	HIV/AIDS - people living with HIV/AIDS	Highways(km)	Imports	Industr produc gr rat
1	4.460000e+08	2.150000e+10	800.00000	7.50000	0.010000	19160.864865	233033.719512	21000.000000	3.759000e+09	5.57
2	4.432637e+10	2.925614e+11	10552.76087	4.77783	2.539702	19160.864865	233033.719512	141557.973913	4.438353e+10	5.57
3	5.524000e+08	1.746000e+10	4900.00000	5.60000	2.539702	19160.864865	233033.719512	18000.000000	2.076000e+09	3.10
4	3.216000e+10	2.123000e+11	6600.00000	6.10000	0.100000	500.000000	9100.000000	104000.000000	1.525000e+10	6.00
5	3.000000e+07	5.000000e+08	8000.00000	4.77783	2.539702	19160.864865	233033.719512	185.000000	1.230000e+08	5.57
6	5.800000e+07	1.900000e+09	26800.00000	2.00000	2.539702	19160.864865	233033.719512	269.000000	1.077000e+09	5.57
7	1.276000e+10	2.317000e+10	2100.00000	11.70000	3.900000	21000.000000	240000.000000	51429.000000	4.896000e+09	1.00
8	2.600000e+06	1.120000e+08	7500.00000	2.80000	2.539702	19160.864865	233033.719512	105.000000	8.090000e+07	3.10
9	4.432637e+10	2.925614e+11	10552.76087	4.77783	2.539702	19160.864865	233033.719512	141557.973913	4.438353e+10	5.57
10	6.890000e+08	7.500000e+08	11000.00000	3.00000	2.539702	19160.864865	233033.719512	250.000000	6.920000e+08	6.00
11	3.378000e+10	4.835000e+11	12400.00000	8.30000	0.700000	1500.000000	130000.000000	215471.000000	2.206000e+10	12.00
12	8.500000e+08	1.365000e+10	4600.00000	9.00000	0.100000	200.000000	2600.000000	8431.000000	1.300000e+09	15.00

Task 3: Processing data and finding important observations/values from the dataset using statistical operations

For the task of processing data, we have used some of statistical methods such as mean, Variance, ratio, etc.

Task 4: Visualization

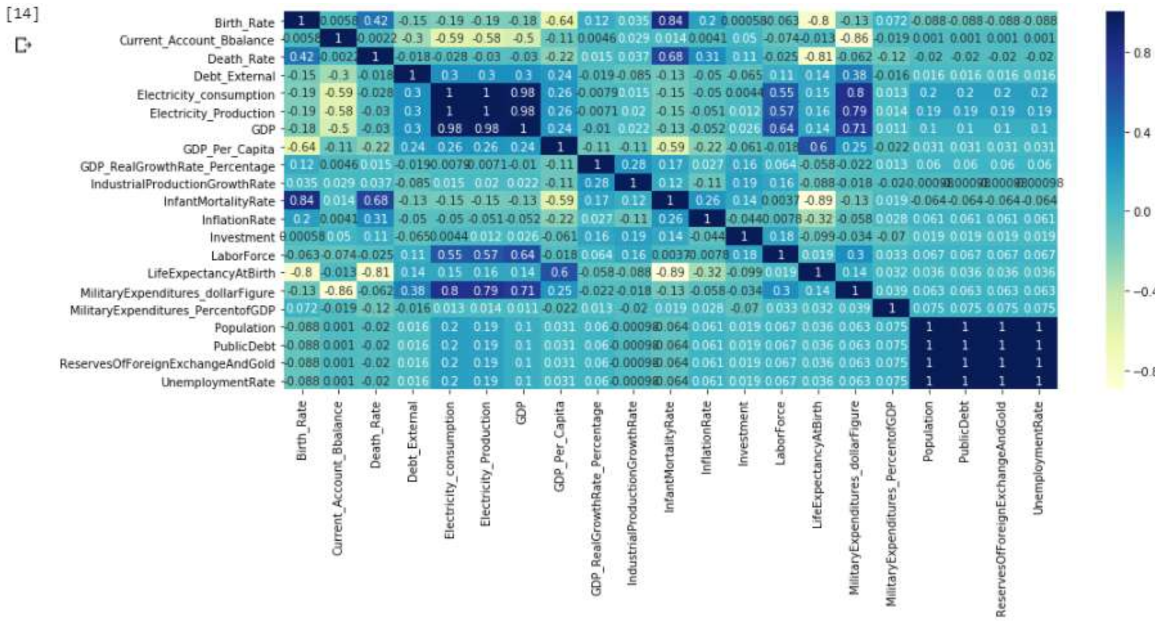
For this step, we used different types of graphs and charts to represent our findings and making it easier by visualizing it in a form of histogram, box plot, etc.

We have used below python libraries for this task:

1. **Matplotlib**
2. **seaborn**
3. **plotly.graph_objs**

RESULT ANALYSIS:

1. Feature engineering using correlation matrix:

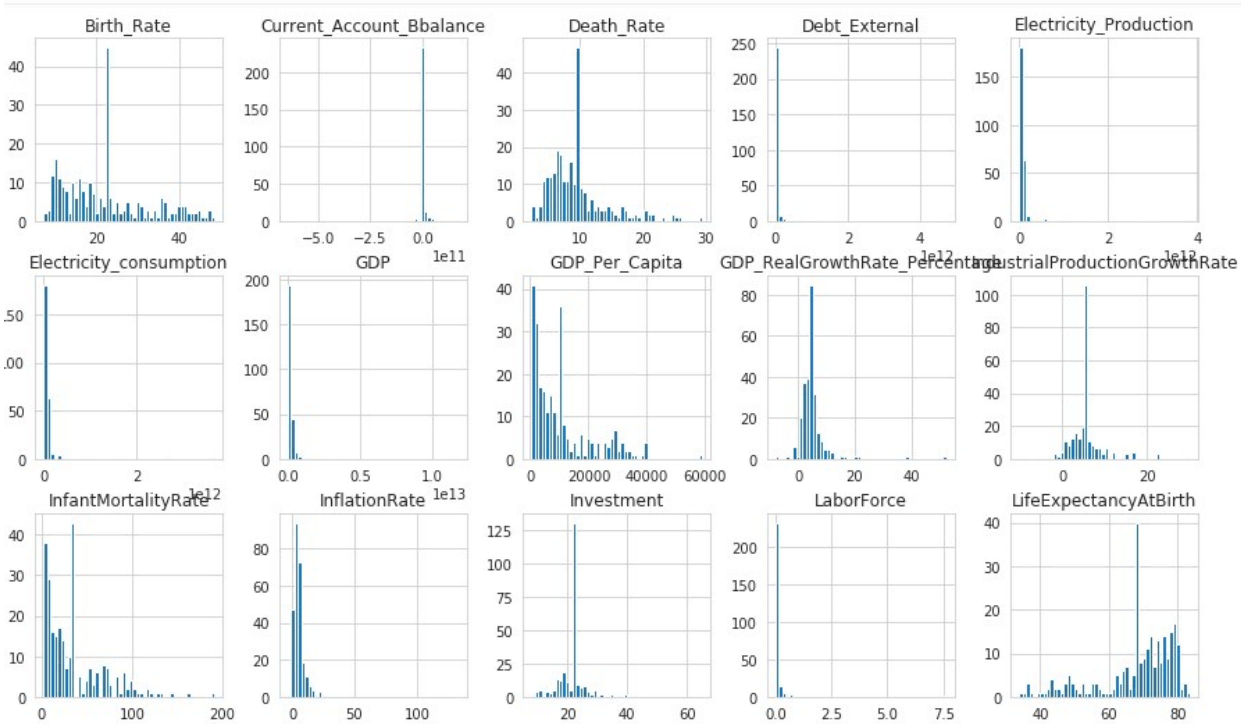


Here we have analyzed the dataset for positively correlated features and deleted unnecessary features.

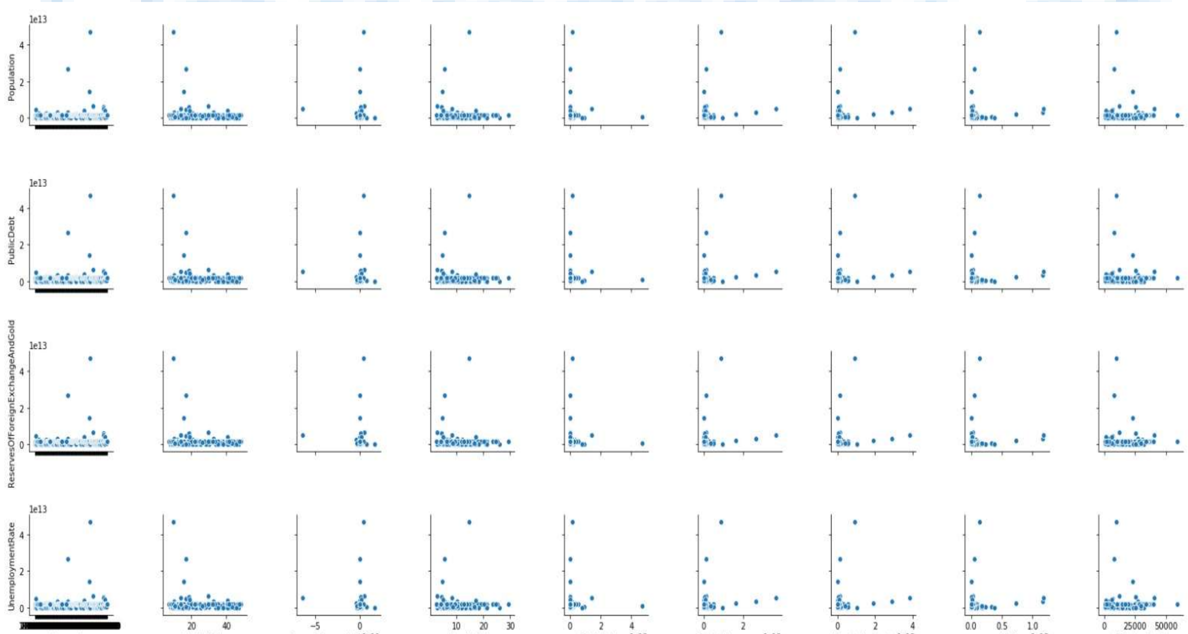
Final dataset column list after reducing correlated features:

```
[9]  print(data.columns)

Index(['Country', 'Area_sqkm', 'Birth_Rate', 'Current_Account_Bbalance',
       'Death_Rate', 'Debt_External', 'Electricity_consumption',
       'Electricity_Production', 'GDP', 'GDP_Per_Capita',
       'GDP_RealGrowthRate_Percentage', 'IndustrialProductionGrowthRate',
       'InfantMortalityRate', 'InflationRate', 'Investment', 'LaborForce',
       'LifeExpectancyAtBirth', 'MilitaryExpenditures_dollarFigure',
       'MilitaryExpenditures_PercentofGDP', 'Population', 'PublicDebt',
       'ReservesOfForeignExchangeAndGold', 'UnemploymentRate'],
      dtype='object')
```

2. Univariate Analysis:



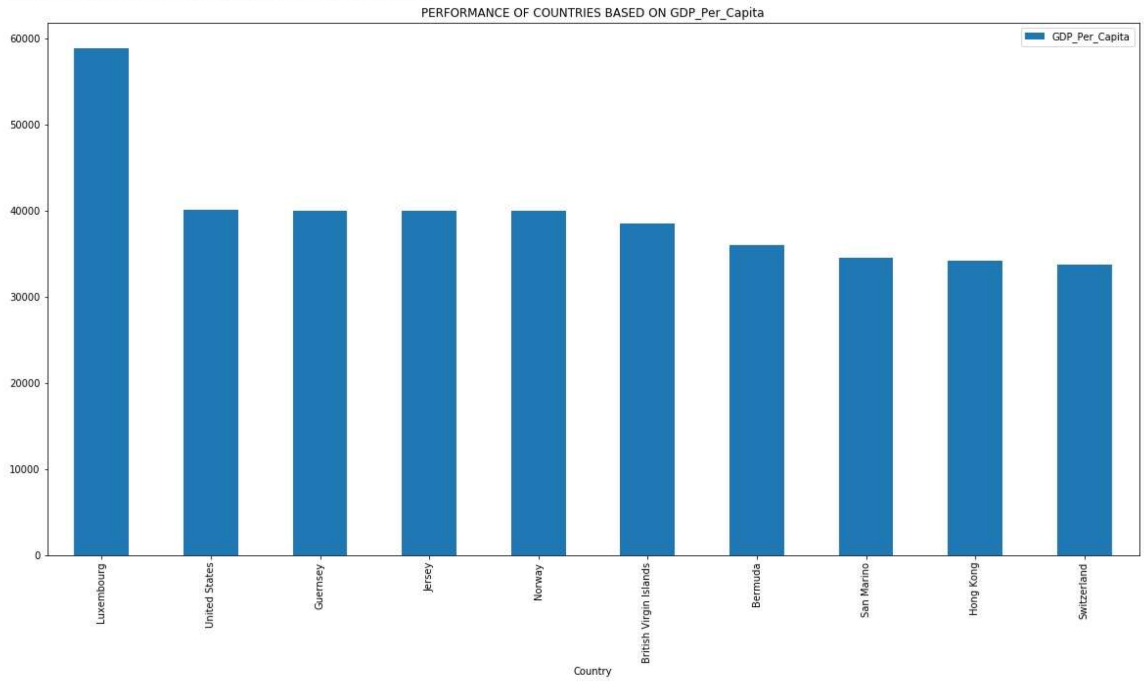
3. Bi-variate Analysis:



4. Visualization of top 10 countries with highest GDP per Capita:

```
data.nlargest(10,'GDP_Per_Capita').plot(kind = 'bar',x = 'Country',y='GDP_Per_Capita',figsize = (20,10),title = "PERFORMANCE OF COUNTRIES BASED ON GDP_Per_Capita")
```

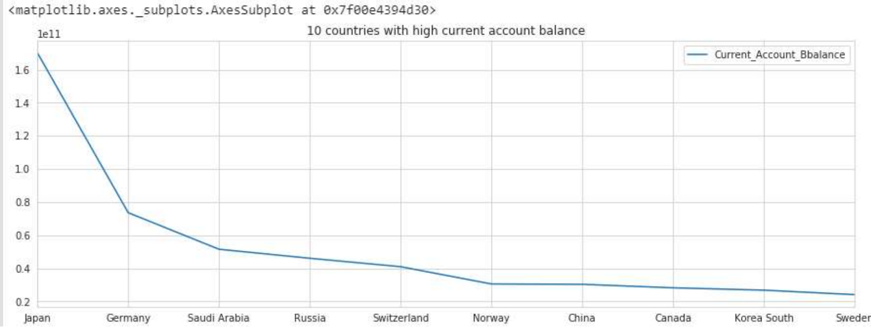
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdcc53cffd0>
```



5. Visualizing top 10 countries with high current account balance:

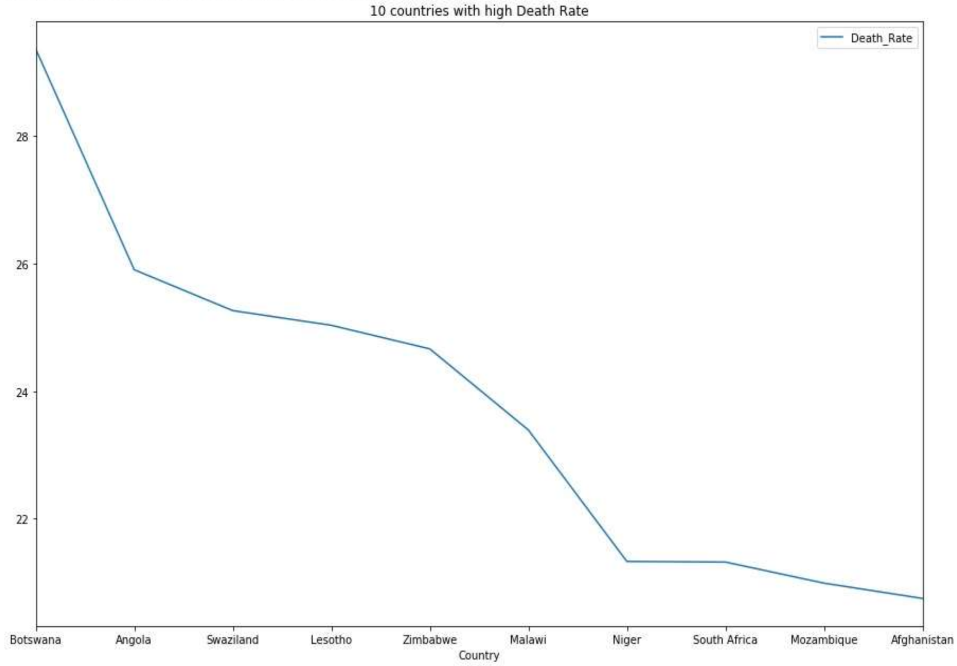
```
data.nlargest(10,'Current_Account_Bbalance').plot(kind = 'line',x = 'Country',y = 'Current_Account_Bbalance',figsize = (15,5), title = "10 countries with high
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f00e4394d30>
```

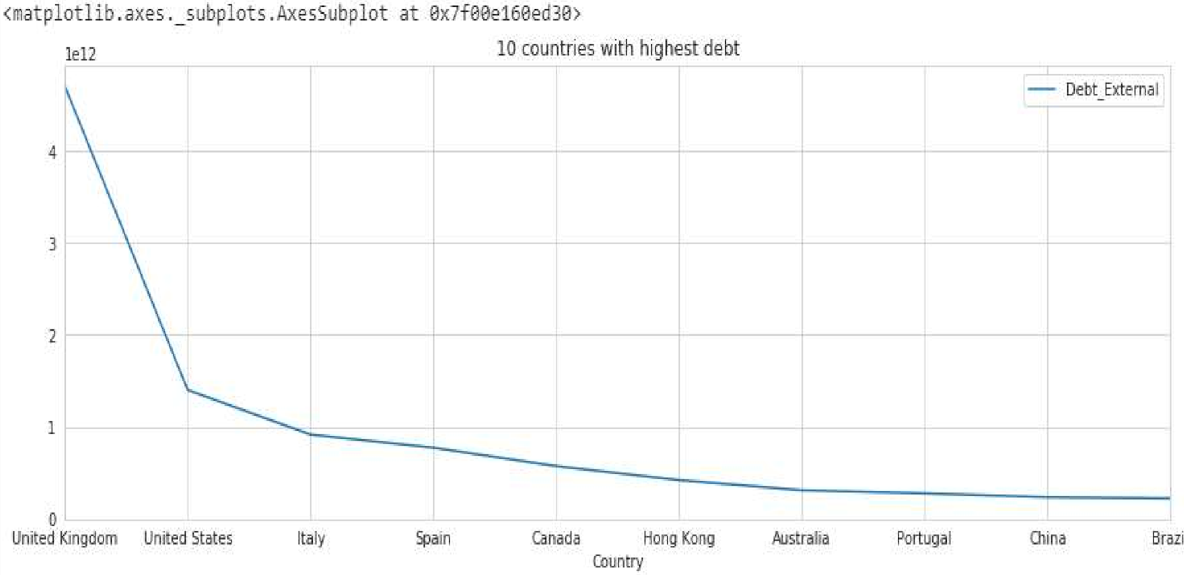


```
[21] data.nlargest(10,'Death_Rate').plot(kind = 'line',x = 'Country',y = 'Death_Rate',figsize = (15,10), title = "10 countries with high Death Rate")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdcc52993c8>
```




```
<matplotlib.axes._subplots.AxesSubplot at 0x7f00e160ed30>
```



6. Classification of countries based on Economic status of country:

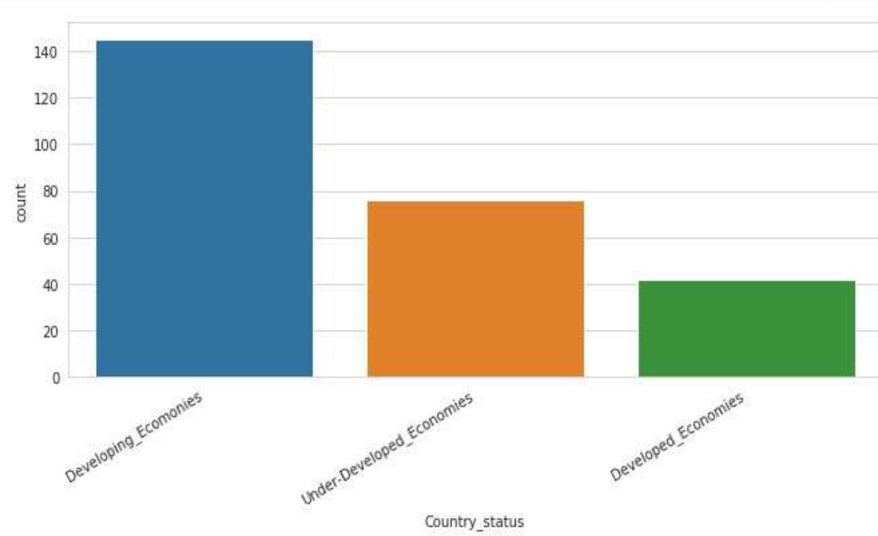
```
import numpy as np;

conditions = [
              (data['GDP_Per_Capita'] >= 22000),
              (data['GDP_Per_Capita'] >= 3000) & (data['GDP_Per_Capita'] < 22000),
              (data['GDP_Per_Capita']>=0) & (data['GDP_Per_Capita'] < 3000)
              ]
choices = ['Developed_Economies','Developing_Ecomonies','Under-Developed_Economies']
data['Country_status'] = np.select(conditions,choices,default = 'Under-Developed_Economies')
data.head()
```

	Country	Area_sqkm	Birth_Rate	Current_Account_Bbalance	Death_Rate	Debt_External	Electricity_consumption	Electricity_Production	GDP	GDP_Per
1	Afghanistan	647500	47.020000	-8.473897e+08	20.750000	8.000000e+09	6.522000e+08	5.400000e+08	2.150000e+10	8
2	Akrotiri	123	22.146667	-8.473897e+08	9.374267	6.317084e+10	8.070919e+10	8.849167e+10	2.925614e+11	1055
3	Albania	28748	15.080000	-5.040000e+08	5.120000	1.410000e+09	6.760000e+09	5.680000e+09	1.746000e+10	49
4	Algeria	2381740	17.130000	1.190000e+10	4.600000	2.190000e+10	2.361000e+10	2.576000e+10	2.123000e+11	66
5	American Samoa	199	23.130000	-8.473897e+08	3.330000	6.317084e+10	1.209000e+08	1.300000e+08	5.000000e+08	80

we have tried to classify all countries of the world into one of three broad categories: developed economies, economies in transition and developing economies.

Graphical representation of economic status of countries:

```
import seaborn as sns
sns.set_style("whitegrid")
fig, ax =plt.subplots(1,2,figsize=(24,5))
sns.countplot(data['Country_status'].sort_values(), ax=ax[0],order = data['Country_status'].value_counts().index)
fig.autofmt_xdate()
fig.show()
```



```python
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot

data6 = dict(type = 'choropleth',
           locations = data['Country'],
           locationmode = 'country names',
           z = data['GDP_Per_Capita'],
           text = data['Country'],
          colorscale = 'Viridis', reversescale = False)
layout = dict(title = 'Economy status of country Across the World',
             geo = dict(showframe = False,
                      projection = {'type': 'mercator'}))
choromap6 = go.Figure(data = [data6], layout=layout)
iplot(choromap6)
```

7. Linear Regression model for predicting GDP per Capita value:

```python
from sklearn.linear_model import LinearRegression
X = dropped_data.drop("GDP_Per_Capita", axis = 1)
lm = LinearRegression()
lm.fit(X, dropped_data['GDP_Per_Capita'])
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```python
print("Estimated Intercept is", lm.intercept_)
```

```
Estimated Intercept is -71173.20070880392
```

```python
print("The number of coefficients in this model are", lm.coef_)
```

```
The number of coefficients in this model are [ 2.27166165e-04  2.83434859e+01  3.84030635e-08  1.63257459e+03
  6.40112711e-10 -3.08499572e-08  1.47481313e-08  4.15537601e-09
 -6.78214181e+01 -2.60846561e+01 -2.96033934e+01 -6.94351843e+01
  1.47985584e+01 -3.40337522e-05  9.81851128e+02  1.46348547e-07
  1.50601602e+02  1.17853953e-01 -1.51588581e-01 -1.57893420e-02
  4.95239701e-02]
```

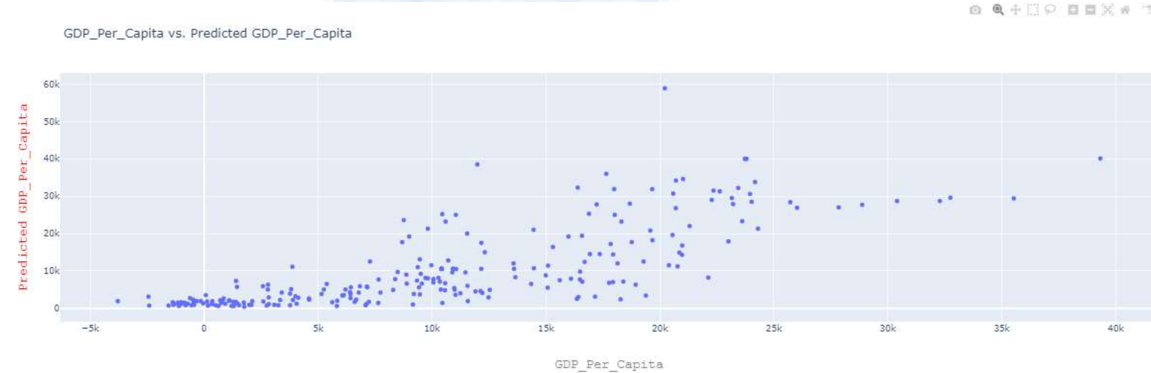
```python
coef = zip(X.columns, lm.coef_)
coef_df = pd.DataFrame(list(zip(X.columns, lm.coef_)), columns=['features', 'coefficients'])
coef_df
```

	features	coefficients
0	Area_sqkm	2.271662e-04
1	Birth_Rate	2.834349e+01
2	Current_Account_Bbalance	3.840306e-08
3	Death_Rate	1.632575e+03
4	Debt_External	6.401127e-10
5	Electricity_consumption	-3.084996e-08
6	Electricity_Production	1.474813e-08
7	GDP	4.155376e-09
8	GDP_RealGrowthRate_Percentage	-6.782142e+01
9	IndustrialProductionGrowthRate	-2.608466e+01
10	InfantMortalityRate	-2.960339e+01
11	InflationRate	-6.943518e+01
12	Investment	1.479856e+01

```
[ ] lm.predict(X)[0:100]
```

```
array([  499.24103791, 10419.89944989, 12535.20314971,  9556.42578656,
        9821.43590313, 20697.64536053,  1116.57812268, 15606.17413861,
       13600.19782025,  9378.01999432, 16697.84282452, 12128.02568585,
       18679.66931683, 10419.87265545, 20583.84604413, 22624.99026495,
        5155.12177235,  8694.27586895,  9000.35077292, 10419.87173993,
        1088.5517183 , 15307.30220087, 10419.87151104, 17772.38249737,
       23953.12549359,  5383.20451886,  -989.39192763, 17645.81172711,
         752.78249127,  4604.98553235, 14349.95955426,  9507.39633709,
       10419.88490826,  9711.23969418, 10419.88513714, 11989.55060345,
        8760.81056744, 22124.29487822,   929.96649701,  1281.13251263,
       -1099.53862995,  -437.94272846,  -137.92067045, 22352.3203941 ,
        7644.45601422, 16395.21314007,   337.6678825 , -1372.49479145,
       14470.73863751,  6441.27338177, 10419.90218121, 10419.87286908,
       10419.87470013,  8894.31574971,  2747.00255353, -1557.7325148 ,
        -808.96537031, 10994.88861615, 10891.18475873,  -854.71989268,
       20776.69990713, 16426.23675686, 18394.12132794, 20979.8378461 ,
       23434.30032641, 10419.90126568,  1460.61885602, 15074.81049115,
        2807.24773175,  1766.57176739,  9467.79279954,  6787.03815991,
        8298.80042035,  -614.5413484 ,  2075.14006573, 20971.21158368,
        -554.50743419, 10419.87787396, 26019.03583294, 11048.4526878 ,
       21305.88231427,  6837.66159588, 22276.41634137, 30405.3461408 ,
       13659.58429486, 12165.8856712 , 10421.6499992 ,  2584.21091467,
         796.38660199,  5826.88557965, 17154.64887005, 32284.01994092,
        -486.4669972 , 23210.84889752, 10419.87265545, 24304.05184449,
       11543.46385724,  5255.49121624, 16096.98802716, 14456.86499554])
```

```
trace = go.Scatter(
    x = lm.predict(X),
    y = dropped_data['GDP_Per_Capita'],
    mode = 'markers'
)
data1 = [trace]
layout = go.Layout(
    title='GDP_Per_Capita vs. Predicted GDP_Per_Capita',
    xaxis=dict(
        title='GDP_Per_Capita',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    ),
    yaxis=dict(
        title='Predicted GDP_Per_Capita',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='Red'
        )
    )
)

fig = go.Figure(data=data1, layout=layout)
iplot(fig)
```



REFERENCES:

[1] The World Factbook, https://en.wikipedia.org/wiki/The_World_Factbook
[2] factbook.csv - World Factbook Country Profile Data in CSV (Comma-Separated Values) Format - Free Open Public Domain Data, https://github.com/thewiremonkey/factbook.csv
[3] Country classification - Data sources, country classifications and aggregation methodology, https://www.un.org/en/development/desa/policy/wesp/wesp_current/2014wesp_country_classification.pdf
[4] Data Analytics with Python by Web scraping: Illustration with CIA World Fact-book, https://towardsdatascience.com/data-analytics-with-python-by-web-scraping-illustration-with-cia-world-factbook-abbdaa687a84
[5] Analyzing-CIA-Factbook-Data-Using-SQLite-and-Python, https://github.com/arjunchndr/Analyzing-CIA-Factbook-Data-Using-SQLite-and-Python
[6] The complete beginner's guide to machine learning: simple linear regression in four lines of code!, https://towardsdatascience.com/simple-linear-regression-in-four-lines-of-code-d690fe4dba84
[7] Linear Regression in Python; Predict The Bay Area's Home Prices, https://towardsdatascience.com/linear-regression-in-python-predict-the-bay-areas-home-price-5c91c8378878