In [1]:
```
# import libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
x=pd.read_csv(r"C:\Users\user\Downloads\18_world-data-2023 - 18_world-data-202
```

Out[2]:

	Country	Density\n(P/Km2)	Abbreviation	Agricultural Land(%)	Land Area(Km2)	Armed Forces size	Birth Rate	Calling Code
0	Afghanistan	60	AF	58.10%	652,230	323,000	32.49	93.0
1	Albania	105	AL	43.10%	28,748	9,000	11.78	355.0
2	Algeria	18	DZ	17.40%	2,381,741	317,000	24.28	213.0
3	Andorra	164	AD	40.00%	468	NaN	7.20	376.0
4	Angola	26	AO	47.50%	1,246,700	117,000	40.73	244.0
...	...	...	...	...	...	...	...	...
190	Venezuela	32	VE	24.50%	912,050	343,000	17.88	58.0
191	Vietnam	314	VN	39.30%	331,210	522,000	16.75	84.0
192	Yemen	56	YE	44.60%	527,968	40,000	30.45	967.0
193	Zambia	25	ZM	32.10%	752,618	16,000	36.19	260.0
194	Zimbabwe	38	ZW	41.90%	390,757	51,000	30.68	263.0

195 rows × 35 columns

In [3]:
```
x=x.head(10)
```

Out[3]:

	Country	Density\n(P/Km2)	Abbreviation	Agricultural Land(%)	Land Area(Km2)	Armed Forces size	Birth Rate	Calling Code
0	Afghanistan	60	AF	58.10%	652,230	323,000	32.49	93.0
1	Albania	105	AL	43.10%	28,748	9,000	11.78	355.0
2	Algeria	18	DZ	17.40%	2,381,741	317,000	24.28	213.0
3	Andorra	164	AD	40.00%	468	NaN	7.20	376.0
4	Angola	26	AO	47.50%	1,246,700	117,000	40.73	244.0
5	Antigua and Barbuda	223	AG	20.50%	443	0	15.33	1.0
6	Argentina	17	AR	54.30%	2,780,400	105,000	17.02	54.0
7	Armenia	104	AM	58.90%	29,743	49,000	13.99	374.0
8	Australia	3	AU	48.20%	7,741,220	58,000	12.60	61.0
9	Austria	109	AT	32.40%	83,871	21,000	9.70	43.0

10 rows × 35 columns

In [4]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 35 columns):
 #   Column                                     Non-Null Count  Dtype  
---  ------                                     --------------  -----  
 0   Country                                    10 non-null     object 
 1   Density
(P/Km2)                          10 non-null     object 
 2   Abbreviation                               10 non-null     object 
 3   Agricultural Land( %)                      10 non-null     object 
 4   Land Area(Km2)                             10 non-null     object 
 5   Armed Forces size                          9 non-null      object 
 6   Birth Rate                                 10 non-null     float64
 7   Calling Code                               10 non-null     float64
 8   Capital/Major City                         10 non-null     object 
 9   Co2-Emissions                              10 non-null     object 
 10  CPI                                        9 non-null      object 
 11  CPI Change (%)                             9 non-null      object 
 12  Currency-Code                              10 non-null     object 
 13  Fertility Rate                             10 non-null     float64
 14  Forested Area (%)                          10 non-null     object 
 15  Gasoline Price                             10 non-null     object 
 16  GDP                                        10 non-null     object 
 17  Gross primary education enrollment (%)     10 non-null     object 
 18  Gross tertiary education enrollment (%)    9 non-null      object 
 19  Infant mortality                           10 non-null     float64
 20  Largest city                               10 non-null     object 
 21  Life expectancy                            9 non-null      float64
 22  Maternal mortality ratio                   9 non-null      float64
 23  Minimum wage                               9 non-null      object 
 24  Official language                          10 non-null     object 
 25  Out of pocket health expenditure           10 non-null     object 
 26  Physicians per thousand                    10 non-null     float64
 27  Population                                 10 non-null     object 
 28  Population: Labor force participation (%)  8 non-null      object 
 29  Tax revenue (%)                            9 non-null      object 
 30  Total tax rate                             9 non-null      object 
 31  Unemployment rate                          8 non-null      object 
 32  Urban_population                           10 non-null     object 
 33  Latitude                                   10 non-null     float64
 34  Longitude                                  10 non-null     float64
dtypes: float64(9), object(26)
memory usage: 2.9+ KB
```

In [5]:

Out[5]:
```
Index(['Country', 'Density\n(P/Km2)', 'Abbreviation', 'Agricultural Land(
%)',
       'Land Area(Km2)', 'Armed Forces size', 'Birth Rate', 'Calling Code',
       'Capital/Major City', 'Co2-Emissions', 'CPI', 'CPI Change (%)',
       'Currency-Code', 'Fertility Rate', 'Forested Area (%)',
       'Gasoline Price', 'GDP', 'Gross primary education enrollment (%)',
       'Gross tertiary education enrollment (%)', 'Infant mortality',
       'Largest city', 'Life expectancy', 'Maternal mortality ratio',
       'Minimum wage', 'Official language', 'Out of pocket health expenditure
',
       'Physicians per thousand', 'Population',
       'Population: Labor force participation (%)', 'Tax revenue (%)',
       'Total tax rate', 'Unemployment rate', 'Urban_population', 'Latitude',
       'Longitude'],
      dtype='object')
```

In [6]:
```
d=x[['Co2-Emissions','Birth Rate','Fertility Rate']]
```

Out[6]:

	Co2-Emissions	Birth Rate	Fertility Rate
0	8,672	32.49	4.47
1	4,536	11.78	1.62
2	150,006	24.28	3.02
3	469	7.20	1.27
4	34,693	40.73	5.52
5	557	15.33	1.99
6	201,348	17.02	2.26
7	5,156	13.99	1.76
8	375,908	12.60	1.74
9	61,448	9.70	1.47

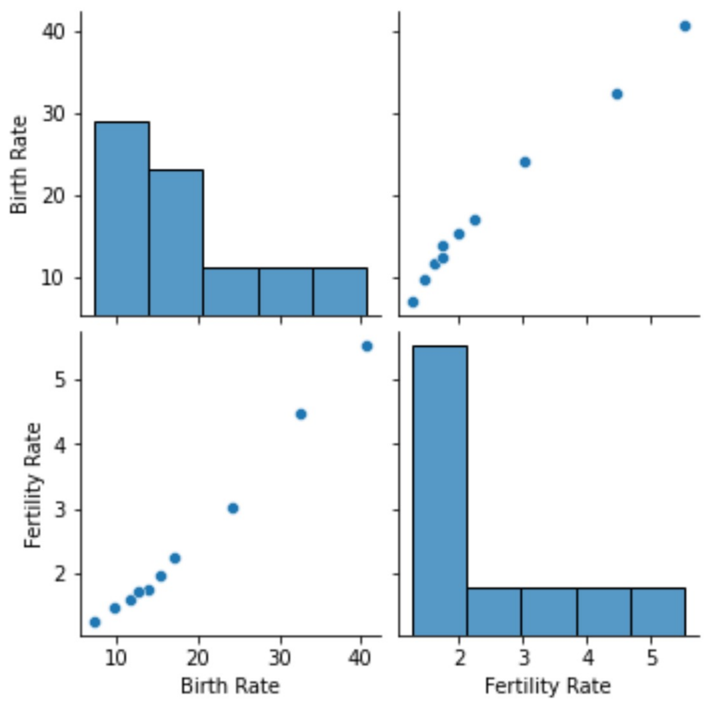
In [7]:

Out[7]:

	Birth Rate	Calling Code	Fertility Rate	Infant mortality	Life expectancy	Maternal mortality ratio	Physicians per thousand	Latit
count	10.000000	10.000000	10.000000	10.000000	9.000000	9.000000	10.000000	10.000
mean	18.512000	181.400000	2.512000	16.090000	74.788889	124.888889	2.671000	17.538
std	10.754729	149.167467	1.416622	18.504321	7.376897	206.621904	1.738387	31.192
min	7.200000	1.000000	1.270000	2.700000	60.800000	5.000000	0.210000	-38.416
25%	11.985000	55.750000	1.650000	3.575000	74.900000	15.000000	1.330000	-4.136
50%	14.660000	153.000000	1.875000	8.300000	76.700000	39.000000	3.045000	30.986
75%	22.465000	327.250000	2.830000	17.825000	78.500000	112.000000	3.890000	40.882
max	40.730000	376.000000	5.520000	51.600000	82.700000	638.000000	5.170000	47.516

In [8]:

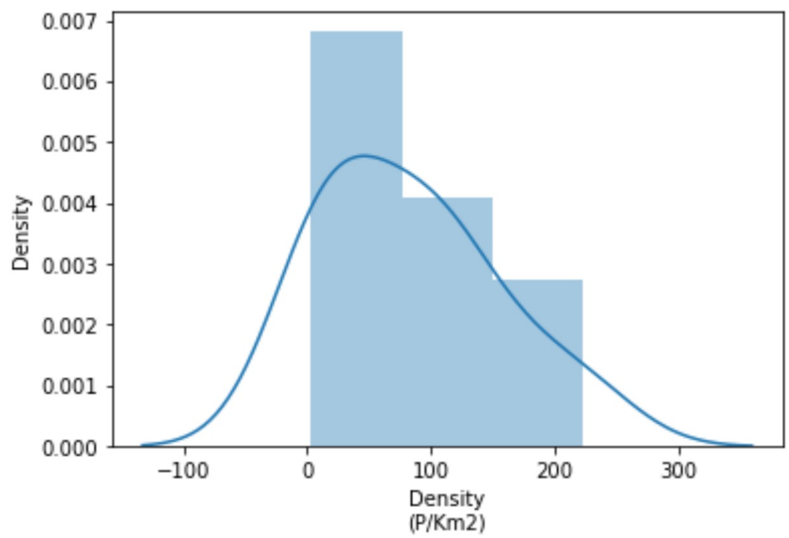
Out[8]: `<seaborn.axisgrid.PairGrid at 0x190982c9610>`



In [9]:

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

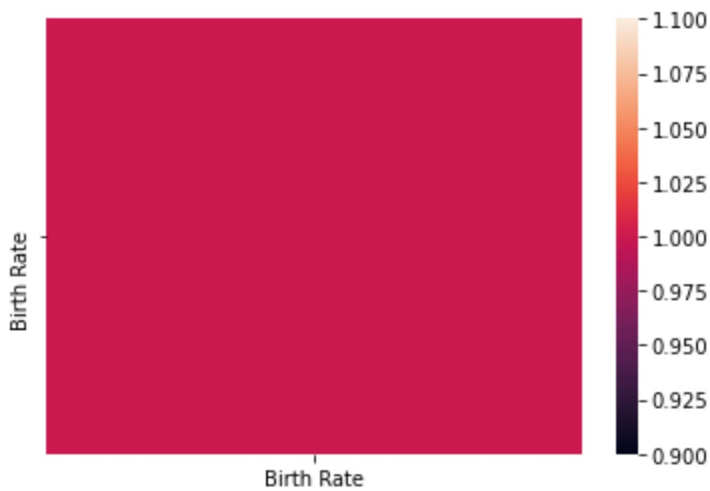
Out[9]: <AxesSubplot:xlabel='Density\n(P/Km2)', ylabel='Density'>



In [10]:

In [11]:

Out[11]: <AxesSubplot:>



In [12]:

```
x=x1[['Birth Rate']]
```

In [13]:
```
# to split my dataset into traning and test date

from sklearn.model_selection import train_test_split
```

In [14]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
```

Out[14]:
```
LinearRegression()
```

In [15]:

```
3.552713678800501e-15
```

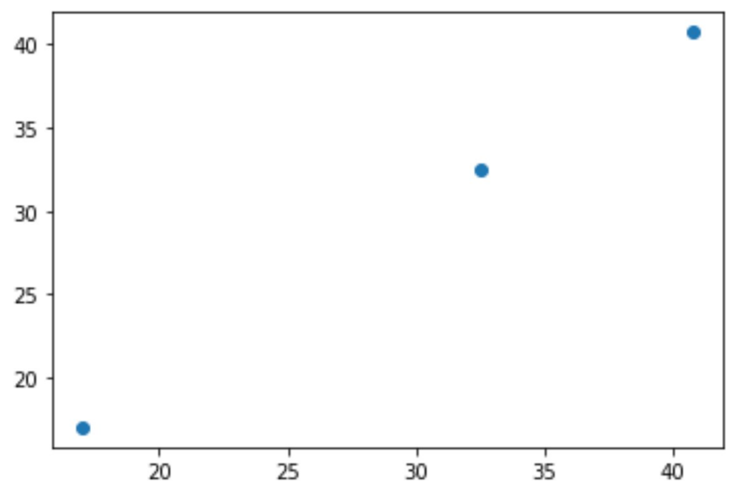
In [16]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[16]:

	Co-efficient
Birth Rate	1.0

In [17]:
```
prediction=lr.predict(x_test)
```

Out[17]:
```
<matplotlib.collections.PathCollection at 0x19099579250>
```



In [18]:

Out[18]:
```
1.0
```

In [19]:

Out[19]:
```
1.0
```

In [20]:

In [21]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[21]: 0.9891341375985607

In [22]:
```
la=Lasso(alpha=10)
```

Out[22]: Lasso(alpha=10)

In [23]:

Out[23]: 0.40595350565336585

In [25]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
```

Out[25]: ElasticNet()

In [26]:

Out[26]: array([0.96136336])

In [27]:

Out[27]: array([39.68002169, 31.75838761, 16.88609644])

In [28]:

Out[28]: 0.5236920669199243

In [29]:

Out[29]: 0.9942868376994006

In [30]:

In [31]:

```
Mean Absolute Error 4.736951571734001e-15
```

In [32]:

```
Mean Squared Error 3.3658065289429835e-29
```

In [33]:

```
Root Mean Squared Error 5.801557143511545e-15
```

In [ ]: