ktqoeydts

July 31, 2023

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

[ ]:
```
df=pd.read_csv("/content/18_world-data-2023.csv")
df
```

[ ]:
```
         Country Density\n(P/Km2) Abbreviation Agricultural Land( %)  \
0    Afghanistan               60           AF               58.10%
1        Albania              105           AL               43.10%
2        Algeria               18           DZ               17.40%
3        Andorra              164           AD               40.00%
4         Angola               26           AO               47.50%
..           …                …          …                    …
190    Venezuela               32           VE               24.50%
191      Vietnam              314           VN               39.30%
192        Yemen               56           YE               44.60%
193       Zambia               25           ZM               32.10%
194     Zimbabwe               38           ZW               41.90%

    Land Area(Km2) Armed Forces size  Birth Rate  Calling Code  \
0          652,230           323,000       32.49          93.0
1           28,748             9,000       11.78         355.0
2        2,381,741           317,000       24.28         213.0
3              468               NaN        7.20         376.0
4        1,246,700           117,000       40.73         244.0
..             …                 …          …            …
190        912,050           343,000       17.88          58.0
191        331,210           522,000       16.75          84.0
192        527,968            40,000       30.45         967.0
193        752,618            16,000       36.19         260.0
194        390,757            51,000       30.68         263.0

    Capital/Major City Co2-Emissions  … Out of pocket health expenditure  \
0                Kabul         8,672  …                           78.40%
1               Tirana         4,536  …                           56.90%
```

```
2              Algiers       150,006  …                         28.10%
3     Andorra la Vella           469  …                         36.40%
4               Luanda        34,693  …                         33.40%
..                 …             …  …                          …
190            Caracas       164,175  …                         45.80%
191              Hanoi       192,668  …                         43.50%
192              Sanaa        10,609  …                         81.00%
193             Lusaka         5,141  …                         27.50%
194             Harare        10,983  …                         25.80%

    Physicians per thousand  Population  \
0                      0.28  38,041,754
1                      1.20   2,854,191
2                      1.72  43,053,054
3                      3.33      77,142
4                      0.21  31,825,295
..                      …           …
190                    1.92  28,515,829
191                    0.82  96,462,106
192                    0.31  29,161,922
193                    1.19  17,861,030
194                    0.21  14,645,468

     Population: Labor force participation (%) Tax revenue (%) Total tax rate  \
0                                        48.90%           9.30%         71.40%
1                                        55.70%          18.60%         36.60%
2                                        41.20%          37.20%         66.10%
3                                           NaN             NaN            NaN
4                                        77.50%           9.20%         49.10%
..                                          …              …             …
190                                      59.70%             NaN         73.30%
191                                      77.40%          19.10%         37.60%
192                                      38.00%             NaN         26.60%
193                                      74.60%          16.20%         15.60%
194                                      83.10%          20.70%         31.60%

    Unemployment rate Urban_population   Latitude   Longitude
0              11.12%        9,797,273  33.939110   67.709953
1              12.33%        1,747,593  41.153332   20.168331
2              11.70%       31,510,100  28.033886    1.659626
3                 NaN           67,873  42.506285    1.521801
4               6.89%       21,061,025 -11.202692   17.873887
..                 …              …        …          …
190             8.80%       25,162,368   6.423750  -66.589730
191             2.01%       35,332,140  14.058324  108.277199
192            12.91%       10,869,523  15.552727   48.516388
193            11.43%        7,871,713 -13.133897   27.849332
```

```
194             4.95%        4,717,305 -19.015438   29.154857

[195 rows x 35 columns]
```

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

[ ]:
```
       Country Density\n(P/Km2) Abbreviation Agricultural Land( %)  \
0  Afghanistan               60           AF                58.10%
1      Albania              105           AL                43.10%
2      Algeria               18           DZ                17.40%
3      Andorra              164           AD                40.00%
4       Angola               26           AO                47.50%

  Land Area(Km2) Armed Forces size  Birth Rate  Calling Code  \
0        652,230           323,000       32.49          93.0
1         28,748             9,000       11.78         355.0
2      2,381,741           317,000       24.28         213.0
3            468               NaN        7.20         376.0
4      1,246,700           117,000       40.73         244.0

  Capital/Major City Co2-Emissions  … Out of pocket health expenditure  \
0              Kabul         8,672  …                           78.40%
1             Tirana         4,536  …                           56.90%
2            Algiers       150,006  …                           28.10%
3   Andorra la Vella           469  …                           36.40%
4             Luanda        34,693  …                           33.40%

  Physicians per thousand  Population  \
0                    0.28  38,041,754
1                    1.20   2,854,191
2                    1.72  43,053,054
3                    3.33      77,142
4                    0.21  31,825,295

   Population: Labor force participation (%) Tax revenue (%) Total tax rate  \
0                                      48.90%           9.30%         71.40%
1                                      55.70%          18.60%         36.60%
2                                      41.20%          37.20%         66.10%
3                                         NaN             NaN            NaN
4                                      77.50%           9.20%         49.10%

  Unemployment rate Urban_population   Latitude  Longitude
0            11.12%        9,797,273  33.939110  67.709953
1            12.33%        1,747,593  41.153332  20.168331
2            11.70%       31,510,100  28.033886   1.659626
3               NaN           67,873  42.506285   1.521801
4             6.89%       21,061,025 -11.202692  17.873887
```

```
[5 rows x 35 columns]
```

1 DATA CLEANING AND DATA PREPROCESSING

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 35 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Country                                     195 non-null    object
 1   Density
(P/Km2)                           195 non-null    object
 2   Abbreviation                                188 non-null    object
 3   Agricultural Land( %)                       188 non-null    object
 4   Land Area(Km2)                              194 non-null    object
 5   Armed Forces size                           171 non-null    object
 6   Birth Rate                                  189 non-null    float64
 7   Calling Code                                194 non-null    float64
 8   Capital/Major City                          192 non-null    object
 9   Co2-Emissions                               188 non-null    object
 10  CPI                                         178 non-null    object
 11  CPI Change (%)                              179 non-null    object
 12  Currency-Code                               180 non-null    object
 13  Fertility Rate                              188 non-null    float64
 14  Forested Area (%)                           188 non-null    object
 15  Gasoline Price                              175 non-null    object
 16  GDP                                         193 non-null    object
 17  Gross primary education enrollment (%)      188 non-null    object
 18  Gross tertiary education enrollment (%)     183 non-null    object
 19  Infant mortality                            189 non-null    float64
 20  Largest city                                189 non-null    object
 21  Life expectancy                             187 non-null    float64
 22  Maternal mortality ratio                    181 non-null    float64
 23  Minimum wage                                150 non-null    object
 24  Official language                           194 non-null    object
 25  Out of pocket health expenditure            188 non-null    object
 26  Physicians per thousand                     188 non-null    float64
 27  Population                                  194 non-null    object
 28  Population: Labor force participation (%)   176 non-null    object
 29  Tax revenue (%)                             169 non-null    object
 30  Total tax rate                              183 non-null    object
 31  Unemployment rate                           176 non-null    object
 32  Urban_population                            190 non-null    object
```

```
 33  Latitude                                   194 non-null    float64
 34  Longitude                                  194 non-null    float64
dtypes: float64(9), object(26)
memory usage: 53.4+ KB
```

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

[ ]:
```
       Birth Rate  Calling Code  Fertility Rate  Infant mortality  \
count  189.000000    194.000000      188.000000        189.000000
mean    20.214974    360.546392        2.698138         21.332804
std      9.945774    323.236419        1.282267         19.548058
min      5.900000      1.000000        0.980000          1.400000
25%     11.300000     82.500000        1.705000          6.000000
50%     17.950000    255.500000        2.245000         14.000000
75%     28.750000    506.750000        3.597500         32.700000
max     46.080000   1876.000000        6.910000         84.500000

       Life expectancy  Maternal mortality ratio  Physicians per thousand  \
count       187.000000                181.000000               188.000000
mean         72.279679                160.392265                 1.839840
std           7.483661                233.502024                 1.684261
min          52.800000                  2.000000                 0.010000
25%          67.000000                 13.000000                 0.332500
50%          73.200000                 53.000000                 1.460000
75%          77.500000                186.000000                 2.935000
max          85.400000               1150.000000                 8.420000

         Latitude   Longitude
count  194.000000  194.000000
mean    19.092351   20.232434
std     23.961779   66.716110
min    -40.900557 -175.198242
25%      4.544175   -7.941496
50%     17.273849   20.972652
75%     40.124603   48.281523
max     64.963051  178.065032
```

[ ]:
```
df.columns
```

[ ]:
```
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')
```

[ ]:
```
df1=df.dropna()
df1
```

[ ]:
```
            Country Density\n(P/Km2) Abbreviation Agricultural Land( %)  \
0       Afghanistan               60           AF                58.10%
1           Albania              105           AL                43.10%
2           Algeria               18           DZ                17.40%
4            Angola               26           AO                47.50%
6         Argentina               17           AR                54.30%
..              ...              ...          ...                   ...
185  United Kingdom              281           GB                71.70%
186   United States               36           US                44.40%
187         Uruguay               20           UY                82.60%
191         Vietnam              314           VN                39.30%
193          Zambia               25           ZM                32.10%

    Land Area(Km2) Armed Forces size  Birth Rate  Calling Code  \
0          652,230           323,000       32.49          93.0
1           28,748             9,000       11.78         355.0
2        2,381,741           317,000       24.28         213.0
4        1,246,700           117,000       40.73         244.0
6        2,780,400           105,000       17.02          54.0
..             ...               ...         ...           ...
185        243,610           148,000       11.00          44.0
186      9,833,517         1,359,000       11.60           1.0
187        176,215            22,000       13.86         598.0
191        331,210           522,000       16.75          84.0
193        752,618            16,000       36.19         260.0

    Capital/Major City Co2-Emissions  … Out of pocket health expenditure  \
0                Kabul         8,672  …                           78.40%
1               Tirana         4,536  …                           56.90%
2              Algiers       150,006  …                           28.10%
4               Luanda        34,693  …                           33.40%
6         Buenos Aires       201,348  …                           17.60%
..                 ...           ...  …                              ...
185             London       379,025  …                           14.80%
186   Washington, D.C.     5,006,302  …                           11.10%
187         Montevideo         6,766  …                           16.20%
191              Hanoi       192,668  …                           43.50%
193             Lusaka         5,141  …                           27.50%
```

```
    Physicians per thousand   Population  \
0                      0.28   38,041,754
1                      1.20    2,854,191
2                      1.72   43,053,054
4                      0.21   31,825,295
6                      3.96   44,938,712
..                      …            …
185                    2.81   66,834,405
186                    2.61  328,239,523
187                    5.05    3,461,734
191                    0.82   96,462,106
193                    1.19   17,861,030

     Population: Labor force participation (%) Tax revenue (%) Total tax rate  \
0                                       48.90%           9.30%         71.40%
1                                       55.70%          18.60%         36.60%
2                                       41.20%          37.20%         66.10%
4                                       77.50%           9.20%         49.10%
6                                       61.30%          10.10%        106.30%
..                                         …              …             …
185                                     62.80%          25.50%         30.60%
186                                     62.00%           9.60%         36.60%
187                                     64.00%          20.10%         41.80%
191                                     77.40%          19.10%         37.60%
193                                     74.60%          16.20%         15.60%

    Unemployment rate Urban_population   Latitude   Longitude
0              11.12%        9,797,273  33.939110   67.709953
1              12.33%        1,747,593  41.153332   20.168331
2              11.70%       31,510,100  28.033886    1.659626
4               6.89%       21,061,025 -11.202692   17.873887
6               9.79%       41,339,571 -38.416097  -63.616672
..                …             …          …          …
185             3.85%       55,908,316  55.378051   -3.435973
186            14.70%      270,663,028  37.090240  -95.712891
187             8.73%        3,303,394 -32.522779  -55.765835
191             2.01%       35,332,140  14.058324  108.277199
193            11.43%        7,871,713 -13.133897   27.849332

[110 rows x 35 columns]
```

[ ]:
```
df1.columns
```

[ ]:
```
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 (%)',
```

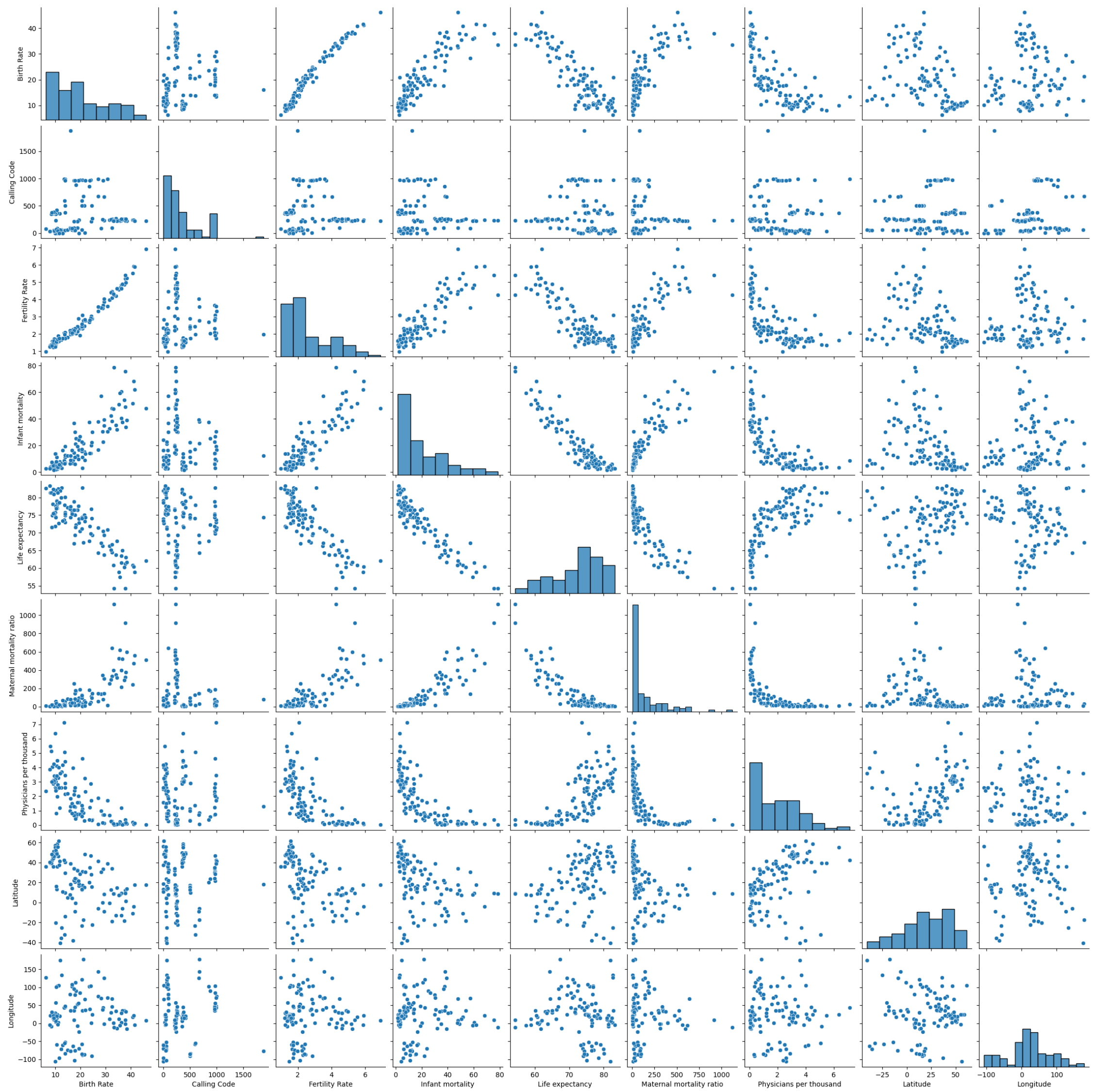
```
       'Currency-Code', 'Fertility Rate', 'Forested Area (%)',
       'Gasoline Price', 'GDP', 'Gross primary education enrollment (%)',
       'Gross tertiary education enrollment (%)', 'Infant mortality',
       'Largest city', 'Life expectancy', 'Maternal mortality ratio',
       'Minimum wage', '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')
```

[ ]:
```
df1=df1[['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']]
```

2 EDA AND VISUALIZATION

[ ]:
```
sns.pairplot(df1)
```

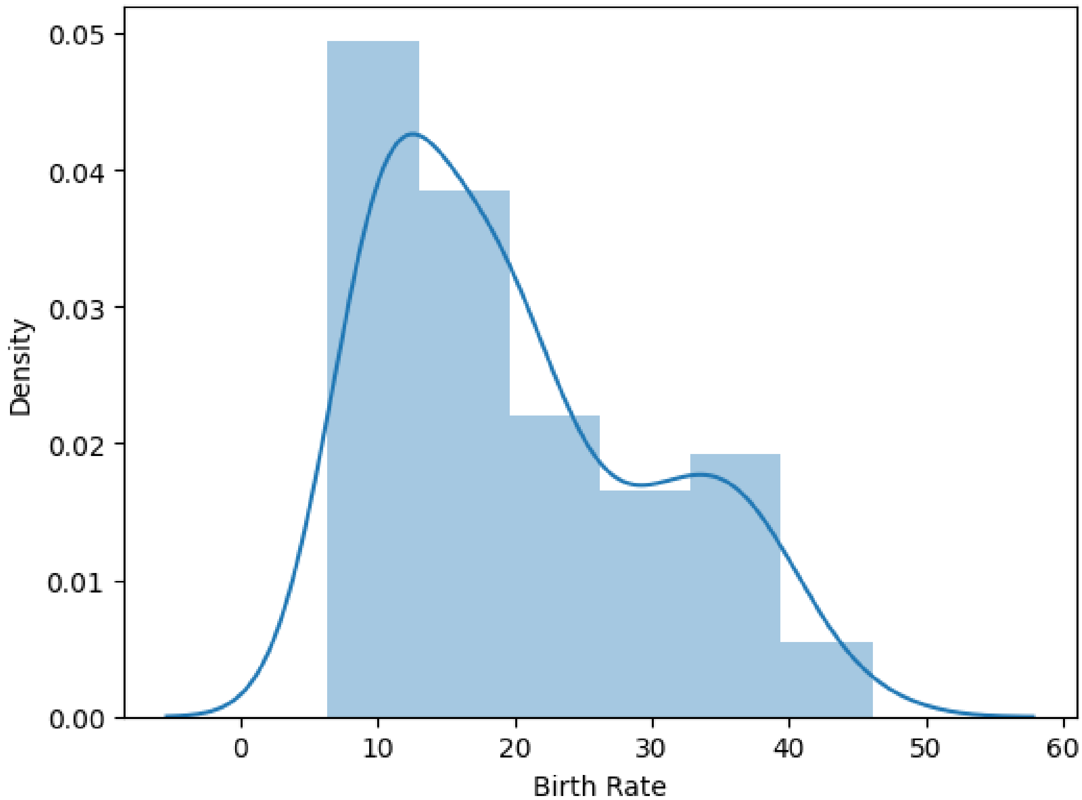
[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7d9569c42020>
```



[ ]:
```
sns.distplot(df1['Birth Rate'])
```

```
<ipython-input-11-a422519242bd>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df1['Birth Rate'])
```

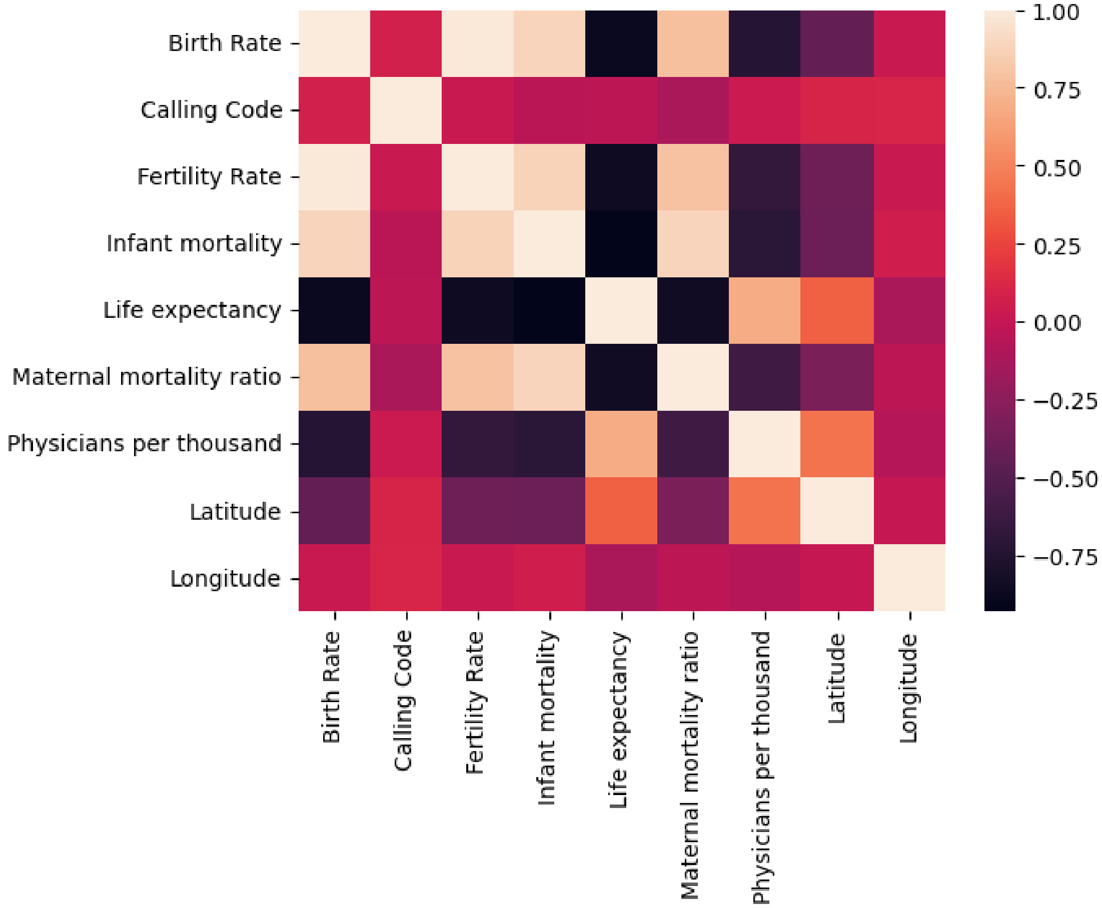
[ ]: <Axes: xlabel='Birth Rate', ylabel='Density'>



[ ]:
```
sns.heatmap(df1.corr())
```

```
<ipython-input-12-3ed1a1a51dc0>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df1.corr())
```

[ ]: <Axes: >



3 TO TRAIN THE MODEL AND MODEL BULDING

[ ]:
```
df2=df[['Calling Code','Fertility Rate', 'Infant mortality', 'Life expectancy',␣
↪'Maternal mortality ratio',
       'Physicians per thousand', 'Latitude',
       'Longitude','Birth Rate']].dropna()
df2=df2[df['Calling Code']!="NaN"]
df2=df2[df['Fertility Rate']!="NaN"]
df2=df2[df['Infant mortality']!="NaN"]
df2=df2[df['Life expectancy']!="NaN"]
df2=df2[df['Maternal mortality ratio']!="NaN"]
df2=df2[df['Physicians per thousand']!="NaN"]
df2=df2[df['Latitude']!="NaN"]
df2=df2[df['Longitude']!="NaN"]
df2=df2[df['Birth Rate']!="NaN"]
x=df2[['Calling Code','Fertility Rate', 'Infant mortality', 'Life expectancy',␣
↪'Maternal mortality ratio',
```

```
       'Physicians per thousand', 'Latitude',
       'Longitude']]
y=df2['Birth Rate']
```

```
<ipython-input-13-de6015c771a4>:4: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Calling Code']!="NaN"]
<ipython-input-13-de6015c771a4>:5: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Fertility Rate']!="NaN"]
<ipython-input-13-de6015c771a4>:6: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Infant mortality']!="NaN"]
<ipython-input-13-de6015c771a4>:7: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Life expectancy']!="NaN"]
<ipython-input-13-de6015c771a4>:8: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Maternal mortality ratio']!="NaN"]
<ipython-input-13-de6015c771a4>:9: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Physicians per thousand']!="NaN"]
<ipython-input-13-de6015c771a4>:10: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Latitude']!="NaN"]
<ipython-input-13-de6015c771a4>:11: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Longitude']!="NaN"]
<ipython-input-13-de6015c771a4>:12: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df2=df2[df['Birth Rate']!="NaN"]
```

[ ]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

[ ]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

[ ]: LinearRegression()

[ ]:
```
lr.intercept_
```

[ ]: 12.54917614941365

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

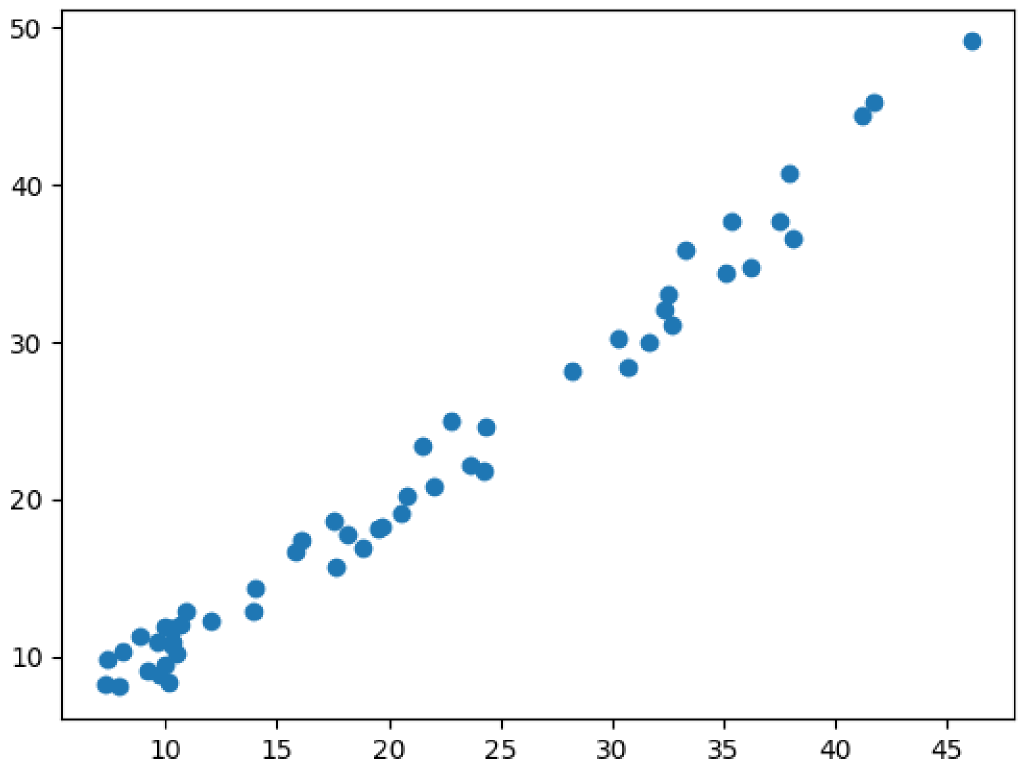
[ ]:

```
                              Co-efficient
Calling Code                      0.001070
Fertility Rate                    6.198562
Infant mortality                  0.050441
Life expectancy                  -0.120693
Maternal mortality ratio         -0.002693
Physicians per thousand          -0.563243
Latitude                         -0.005451
Longitude                         0.000596
```

[ ]:

```
prediction =lr.predict(x_test)
plt.scatter(y_test,prediction)
```

[ ]: <matplotlib.collections.PathCollection at 0x7d955eccf490>



4 ACCURACY

[ ]:

```
lr.score(x_test,y_test)
```

[ ]: 0.9781707675313001

[ ]:
```
lr.score(x_train,y_train)
```

[ ]: 0.9768360452532864

[ ]:
```
from sklearn.linear_model import Ridge,Lasso
```

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

[ ]: Ridge(alpha=10)

[ ]:
```
rr.score(x_test,y_test)
```

[ ]: 0.9690852447922472

[ ]:
```
rr.score(x_train,y_train)
```

[ ]: 0.9712293240075267

[ ]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

[ ]: Lasso(alpha=10)

[ ]:
```
la.score(x_test,y_test)
```

[ ]: 0.7802205081238347

[ ]:
```
la.score(x_train,y_train)
```

[ ]: 0.7844134640141122

[ ]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

[ ]: ElasticNet()

[ ]:
```
print(en.coef_)
print(en.intercept_)
```

```
[ 2.41178946e-03  1.95584230e+00  1.92665309e-01 -2.43358181e-01
  1.52906232e-03 -5.11642416e-01 -2.44267326e-02 -1.97579485e-03]
29.109323628499595
```

[ ]:
```
prediction = en.predict(x_test)
prediction
```

[ ]:
```
array([30.4178063 , 18.04863424, 12.48083622, 13.77545893, 29.71875542,
       37.64315632, 31.47778011, 35.7578751 , 15.72619536, 13.2586933 ,
       11.53827816, 15.91147254, 36.55271231, 21.67342203,  9.55507551,
       28.33968038,  8.96186041, 10.25009375, 43.37502248, 11.36172344,
       15.37353694, 24.73735873, 27.67920334, 19.46442016, 11.16012541,
       35.44657516, 29.72101406, 31.10500217, 11.40745272,  9.19691672,
       14.2136627 ,  9.34117635, 14.73866027, 20.77266582,  8.91068521,
       29.46135333, 27.99886105, 20.6480372 , 14.55953288, 19.39510321,
       12.79463541, 29.63241311, 16.65266665, 16.11832367, 30.99731576,
       20.95204915, 42.55482893, 40.45799896, 23.66234339,  8.97205267,
       43.49169656, 18.19848977, 19.33985349, 12.54930301])
```

[ ]:
```
en.score(x_test,y_test)
```

[ ]:
```
0.8926803664183731
```

[ ]:
```
from sklearn import metrics
print("Mean Absolute Error: ", metrics.mean_absolute_error(y_test,prediction))
print("Mean Squared Error: ", metrics.mean_squared_error(y_test,prediction))
print("Root Mean Squared Error: ", np.sqrt(metrics.
 ↪mean_squared_error(y_test,prediction)))
```

```
Mean Absolute Error:  2.9056740074901657
Mean Squared Error:  13.010233417125345
Root Mean Squared Error:  3.606970115917977
```