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

In [169]:
```
s=pd.read_csv(r"C:\Users\user\Downloads\22_countries - 22_countries.csv")
s
```

Out[169]:

| | id | name | iso3 | iso2 | numeric_code | phone_code | capital | currency | currency_nai |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | Afghanistan | AFG | AF | 4 | 93 | Kabul | AFN | Afghan afgh |
| **1** | 2 | Aland Islands | ALA | AX | 248 | +358-18 | Mariehamn | EUR | Eu |
| **2** | 3 | Albania | ALB | AL | 8 | 355 | Tirana | ALL | Albanian |
| **3** | 4 | Algeria | DZA | DZ | 12 | 213 | Algiers | DZD | Algerian dir |
| **4** | 5 | American Samoa | ASM | AS | 16 | +1-684 | Pago Pago | USD | US Dol |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **245** | 243 | Wallis And Futuna Islands | WLF | WF | 876 | 681 | Mata Utu | XPF | CFP fra |
| **246** | 244 | Western Sahara | ESH | EH | 732 | 212 | El-Aaiun | MAD | Morocc Dirh |
| **247** | 245 | Yemen | YEM | YE | 887 | 967 | Sanaa | YER | Yemeni |
| **248** | 246 | Zambia | ZMB | ZM | 894 | 260 | Lusaka | ZMW | Zambi kwac |
| **249** | 247 | Zimbabwe | ZWE | ZW | 716 | 263 | Harare | ZWL | Zimbab Dol |

250 rows × 19 columns

In [170]:
```
s=s.head(50)
s
```

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **37** | 38 | Cameroon | CMR | CM | 120 | 237 | Yaounde | XAF | Centra C |
| **38** | 39 | Canada | CAN | CA | 124 | 1 | Ottawa | CAD | Canadi |
| **39** | 40 | Cape Verde | CPV | CV | 132 | 238 | Praia | CVE | Cape |
| **40** | 41 | Cayman Islands | CYM | KY | 136 | +1-345 | George Town | KYD | Caymar |
| **41** | 42 | Central African Republic | CAF | CF | 140 | 236 | Bangui | XAF | Centra C |
| **42** | 43 | Chad | TCD | TD | 148 | 235 | N'Djamena | XAF | Centra C |
| **43** | 44 | Chile | CHL | CL | 152 | 56 | Santiago | CLP | Chile |
| **44** | 45 | China | CHN | CN | 156 | 86 | Beijing | CNY | Chine |

In [171]:
```
s.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               50 non-null     int64  
 1   name             50 non-null     object 
 2   iso3             50 non-null     object 
 3   iso2             50 non-null     object 
 4   numeric_code     50 non-null     int64  
 5   phone_code       50 non-null     object 
 6   capital          48 non-null     object 
 7   currency         50 non-null     object 
 8   currency_name    50 non-null     object 
 9   currency_symbol  50 non-null     object 
 10  tld              50 non-null     object 
 11  native           50 non-null     object 
 12  region           49 non-null     object 
 13  subregion        48 non-null     object 
 14  timezones        50 non-null     object 
 15  latitude         50 non-null     float64
 16  longitude        50 non-null     float64
 17  emoji            50 non-null     object 
 18  emojiU           50 non-null     object 
dtypes: float64(2), int64(2), object(15)
memory usage: 7.5+ KB
```

```
In [172]: s.describe()
```

Out[172]:

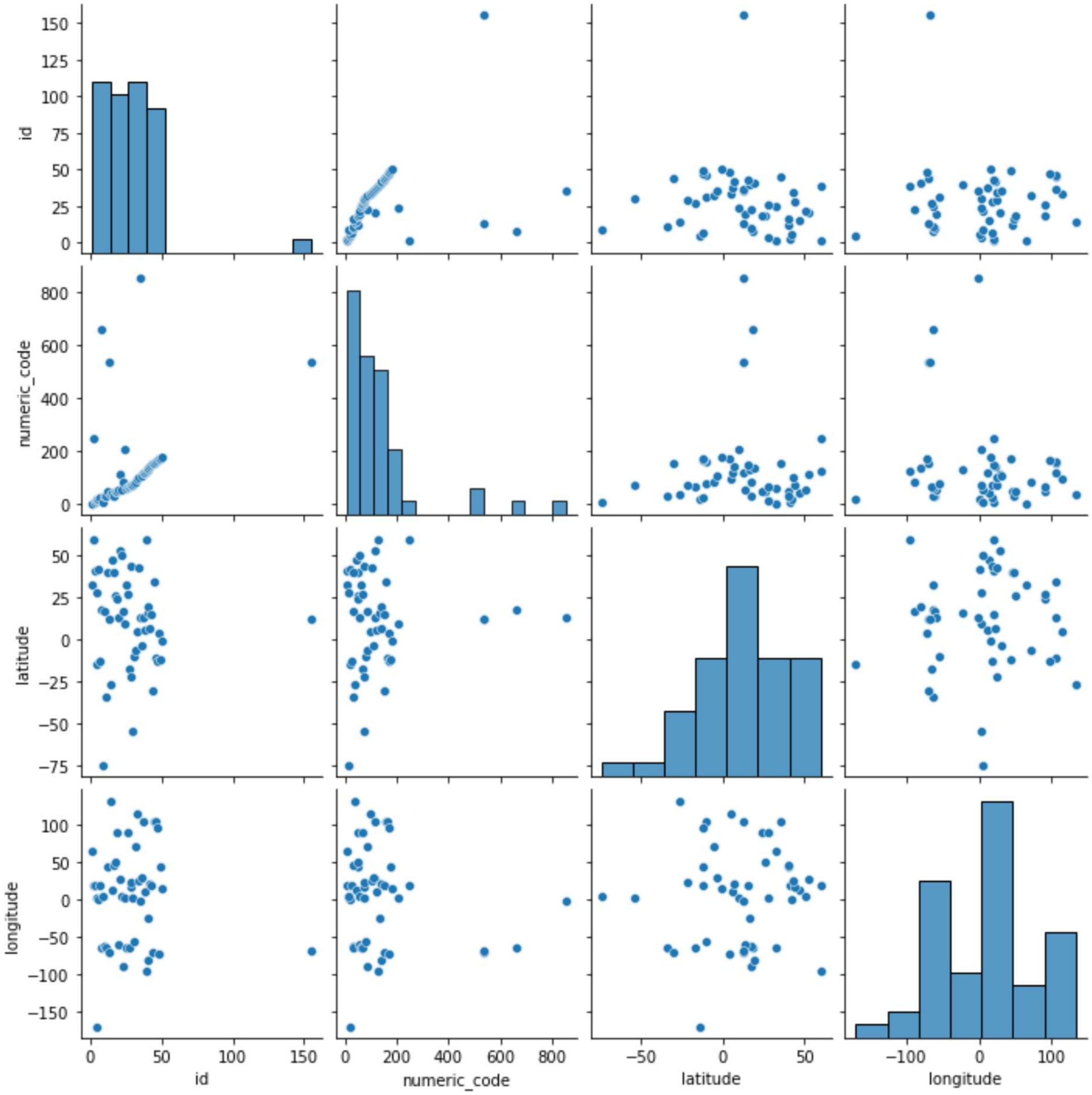
| | id | numeric_code | latitude | longitude |
|---|---|---|---|---|
| **count** | 50.000000 | 50.000000 | 50.000000 | 50.000000 |
| **mean** | 28.260000 | 133.920000 | 11.688000 | 5.441933 |
| **std** | 23.355994 | 166.988676 | 28.752855 | 65.724453 |
| **min** | 1.000000 | 4.000000 | -74.650000 | -170.000000 |
| **25%** | 13.250000 | 48.500000 | -9.000000 | -61.233333 |
| **50%** | 26.500000 | 85.000000 | 13.083333 | 14.166667 |
| **75%** | 38.750000 | 151.000000 | 32.833333 | 44.812500 |
| **max** | 155.000000 | 854.000000 | 60.116667 | 133.000000 |

```
In [173]: s.columns
```

```
Out[173]: Index(['id', 'name', 'iso3', 'iso2', 'numeric_code', 'phone_code', 'capital',
                 'currency', 'currency_name', 'currency_symbol', 'tld', 'native',
                 'region', 'subregion', 'timezones', 'latitude', 'longitude', 'emoji',
                 'emojiU'],
                dtype='object')
```

```
In [174]: sns.pairplot(s)
```

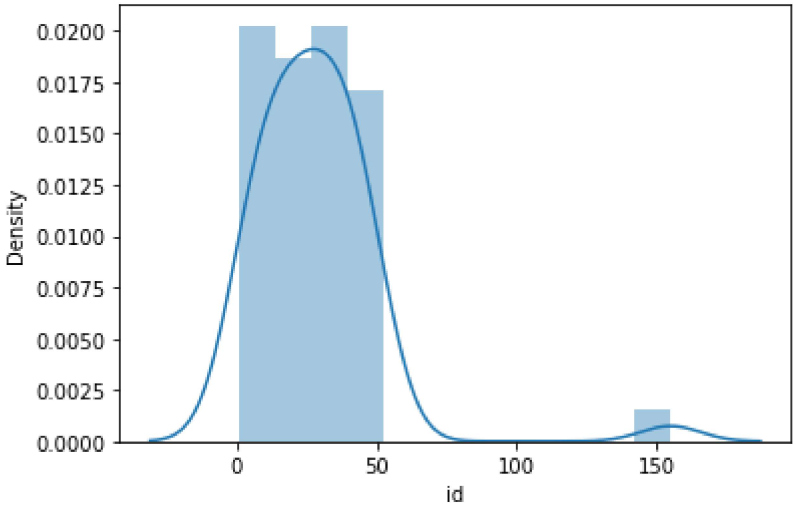
```
Out[174]: <seaborn.axisgrid.PairGrid at 0x23cfadd1af0>
```



```
In [175]: sns.distplot(s['id'])
```

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

```
Out[175]: <AxesSubplot:xlabel='id', ylabel='Density'>
```



In [177]:

```
s1=s[['id','latitude', 'longitude','numeric_code','phone_code']]
s1
```

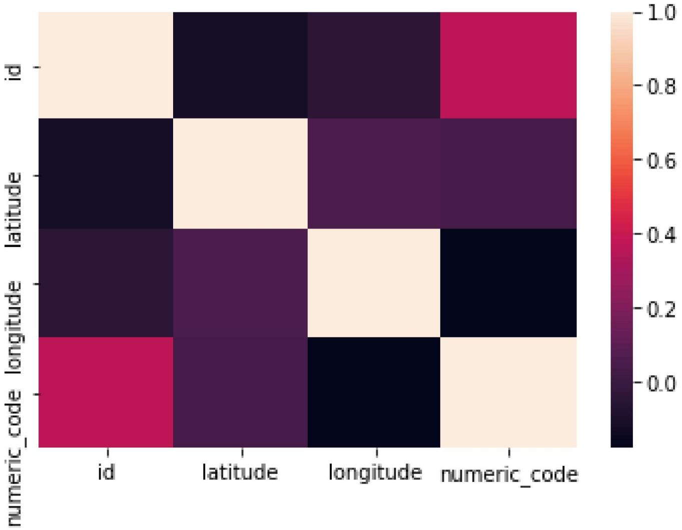
Out[177]:

| | id | latitude | longitude | numeric_code | phone_code |
|---|---|---|---|---|---|
| **0** | 1 | 33.000000 | 65.000000 | 4 | 93 |
| **1** | 2 | 60.116667 | 19.900000 | 248 | +358-18 |
| **2** | 3 | 41.000000 | 20.000000 | 8 | 355 |
| **3** | 4 | 28.000000 | 3.000000 | 12 | 213 |
| **4** | 5 | -14.333333 | -170.000000 | 16 | +1-684 |
| **5** | 6 | 42.500000 | 1.500000 | 20 | 376 |
| **6** | 7 | -12.500000 | 18.500000 | 24 | 244 |
| **7** | 8 | 18.250000 | -63.166667 | 660 | +1-264 |
| **8** | 9 | -74.650000 | 4.480000 | 10 | 672 |
| **9** | 10 | 17.050000 | -61.800000 | 28 | +1-268 |
| **10** | 11 | -34.000000 | -64.000000 | 32 | 54 |
| **11** | 12 | 40.000000 | 45.000000 | 51 | 374 |
| **12** | 13 | 12.500000 | -69.966667 | 533 | 297 |
| **13** | 14 | -27.000000 | 133.000000 | 36 | 61 |
| **14** | 15 | 47.333333 | 13.333333 | 40 | 43 |
| **15** | 16 | 40.500000 | 47.500000 | 31 | 994 |
| **16** | 18 | 26.000000 | 50.550000 | 48 | 973 |
| **17** | 19 | 24.000000 | 90.000000 | 50 | 880 |
| **18** | 20 | 13.166667 | -59.533333 | 52 | +1-246 |
| **19** | 21 | 53.000000 | 28.000000 | 112 | 375 |
| **20** | 22 | 50.833333 | 4.000000 | 56 | 32 |
| **21** | 23 | 17.250000 | -88.750000 | 84 | 501 |
| **22** | 24 | 9.500000 | 2.250000 | 204 | 229 |
| **23** | 25 | 32.333333 | -64.750000 | 60 | +1-441 |
| **24** | 26 | 27.500000 | 90.500000 | 64 | 975 |
| **25** | 27 | -17.000000 | -65.000000 | 68 | 591 |
| **26** | 155 | 12.150000 | -68.266667 | 535 | 599 |
| **27** | 28 | 44.000000 | 18.000000 | 70 | 387 |
| **28** | 29 | -22.000000 | 24.000000 | 72 | 267 |
| **29** | 30 | -54.433333 | 3.400000 | 74 | 55 |
| **30** | 31 | -10.000000 | -55.000000 | 76 | 55 |
| **31** | 32 | -6.000000 | 71.500000 | 86 | 246 |
| **32** | 33 | 4.500000 | 114.666667 | 96 | 673 |
| **33** | 34 | 43.000000 | 25.000000 | 100 | 359 |
| **34** | 35 | 13.000000 | -2.000000 | 854 | 226 |

| | id | latitude | longitude | numeric_code | phone_code |
|---|---|---|---|---|---|
| **35** | 36 | -3.500000 | 30.000000 | 108 | 257 |
| **36** | 37 | 13.000000 | 105.000000 | 116 | 855 |
| **37** | 38 | 6.000000 | 12.000000 | 120 | 237 |
| **38** | 39 | 60.000000 | -95.000000 | 124 | 1 |
| **39** | 40 | 16.000000 | -24.000000 | 132 | 238 |
| **40** | 41 | 19.500000 | -80.500000 | 136 | +1-345 |
| **41** | 42 | 7.000000 | 21.000000 | 140 | 236 |
| **42** | 43 | 15.000000 | 19.000000 | 148 | 235 |
| **43** | 44 | -30.000000 | -71.000000 | 152 | 56 |
| **44** | 45 | 35.000000 | 105.000000 | 156 | 86 |
| **45** | 46 | -10.500000 | 105.666667 | 162 | 61 |
| **46** | 47 | -12.500000 | 96.833333 | 166 | 61 |
| **47** | 48 | 4.000000 | -72.000000 | 170 | 57 |
| **48** | 49 | -12.166667 | 44.250000 | 174 | 269 |
| **49** | 50 | -1.000000 | 15.000000 | 178 | 242 |

In [178]:
```
sns.heatmap(s1.corr())
```

Out[178]:
```
<AxesSubplot:>
```



In [191]:
```
x=s1[['id','latitude', 'longitude','numeric_code']]
y=s1['longitude']
```

In [192]:
```
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)
```

```
In [193]: from sklearn.linear_model import LinearRegression
          lr=LinearRegression()
          lr.fit(x_train,y_train)

Out[193]: LinearRegression()
```

```
In [194]: lr.intercept_

Out[194]: -4.085620730620576e-14
```

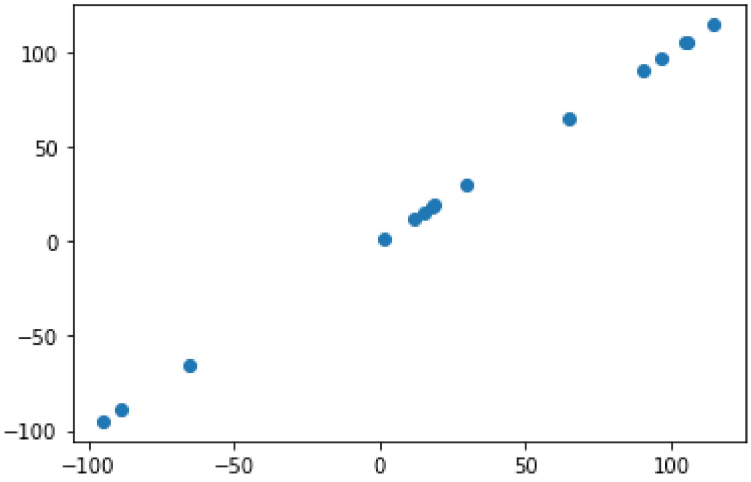
```
In [195]: coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
          coeff

Out[195]:
```

| | Co-efficient |
|---|---|
| **id** | 5.331691e-17 |
| **latitude** | 5.677978e-16 |
| **longitude** | 1.000000e+00 |
| **numeric_code** | 2.242329e-16 |

```
In [196]: prediction=lr.predict(x_test)
          plt.scatter(y_test,prediction)

Out[196]: <matplotlib.collections.PathCollection at 0x23cfbab5be0>
```



```
In [197]: print(lr.score(x_test,y_test))

          1.0
```

```
In [198]: from sklearn.linear_model import Ridge,Lasso
          from sklearn.linear_model import Ridge,Lasso
```

```
In [199]: rr=Ridge(alpha=10)
          rr.fit(x_train,y_train)
          rr.score(x_test,y_test)

Out[199]: 0.999999991552783
```

```
In [200]: la=Lasso(alpha=10)
          la.fit(x_train,y_train)
          la.score(x_test,y_test)

Out[200]: 0.9999913569026688
```

```
In [201]: from sklearn.linear_model import ElasticNet
          en=ElasticNet()
          en.fit(x_train,y_train)

Out[201]: ElasticNet()
```

```
In [202]: print(en.coef_)

          [-0.00000000e+00  0.00000000e+00  9.99734891e-01 -5.02329370e-06]
```

```
In [203]: print(en.intercept_)

          -0.000417771470554662
```

```
In [204]: print(en.predict(x_test))

          [-94.97585527 -88.72731128  29.99108643 -64.98352725  64.98233003
            96.80641027  14.99471144  90.47526834 105.63742194   1.4990841
           114.63536749 104.97096212  18.9938017   11.99579812  17.99445863]
```

```
In [205]: print(en.score(x_test,y_test))

          0.9999999136285275
```

```
In [206]: from sklearn import metrics
```

```
In [207]: print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))

          Mean Absolute Error 1.5380289634473835e-14
```

```
In [208]: print("Mean squared Error",metrics.mean_squared_error(y_test,prediction))

          Mean squared Error 3.5767282134764515e-28
```

```
In [209]: import pickle
```

In [210]:

```
filename="prediction"
pickle.dump(lr,open(filename,'wb'))
```

In [211]:

```
import pandas as pd
import pickle
```

In [212]:

```
filename='prediction'
model=pickle.load(open(filename,'rb'))
```

In [213]:

```
real=[[55,200,50,40],[80,55,66,88]]
result=model.predict(real)
```

In [214]:

```
result
```

Out[214]:

```
array([50., 66.])
```

In [ ]:

In [ ]: