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

In [93]:
```
s=pd.read_csv(r"C:\Users\user\Downloads\20_states - 20_states.csv")
s
```

Out[93]:

	id	name	country_id	country_code	country_name	state_code	type	latitude
0	3901	Badakhshan	1	AF	Afghanistan	BDS	NaN	36.734772
1	3871	Badghis	1	AF	Afghanistan	BDG	NaN	35.167134
2	3875	Baghlan	1	AF	Afghanistan	BGL	NaN	36.178903
3	3884	Balkh	1	AF	Afghanistan	BAL	NaN	36.755060
4	3872	Bamyan	1	AF	Afghanistan	BAM	NaN	34.810007
...	...	...	...	...	...	...	...	...
5072	1953	Mashonaland West Province	247	ZW	Zimbabwe	MW	NaN	-17.485103
5073	1960	Masvingo Province	247	ZW	Zimbabwe	MV	NaN	-20.624151
5074	1954	Matabeleland North Province	247	ZW	Zimbabwe	MN	NaN	-18.533157
5075	1952	Matabeleland South Province	247	ZW	Zimbabwe	MS	NaN	-21.052337
5076	1957	Midlands Province	247	ZW	Zimbabwe	MI	NaN	-19.055201

5077 rows × 9 columns

In [94]:
```
s=s.head(20)
s
```

Out[94]:

	id	name	country_id	country_code	country_name	state_code	type	latitude	lo
0	3901	Badakhshan	1	AF	Afghanistan	BDS	NaN	36.734772	70
1	3871	Badghis	1	AF	Afghanistan	BDG	NaN	35.167134	63
2	3875	Baghlan	1	AF	Afghanistan	BGL	NaN	36.178903	68
3	3884	Balkh	1	AF	Afghanistan	BAL	NaN	36.755060	66
4	3872	Bamyan	1	AF	Afghanistan	BAM	NaN	34.810007	67
5	3892	Daykundi	1	AF	Afghanistan	DAY	NaN	33.669495	66
6	3899	Farah	1	AF	Afghanistan	FRA	NaN	32.495328	62
7	3889	Faryab	1	AF	Afghanistan	FYB	NaN	36.079561	64
8	3870	Ghazni	1	AF	Afghanistan	GHA	NaN	33.545059	68
9	3888	Ghōr	1	AF	Afghanistan	GHO	NaN	34.099578	64
10	3873	Helmand	1	AF	Afghanistan	HEL	NaN	39.298936	-76
11	3887	Herat	1	AF	Afghanistan	HER	NaN	34.352865	62
12	3886	Jowzjan	1	AF	Afghanistan	JOW	NaN	36.896969	65
13	3902	Kabul	1	AF	Afghanistan	KAB	NaN	34.555349	69
14	3890	Kandahar	1	AF	Afghanistan	KAN	NaN	31.628871	65
15	3879	Kapisa	1	AF	Afghanistan	KAP	NaN	34.981057	69
16	3878	Khost	1	AF	Afghanistan	KHO	NaN	33.333847	69
17	3876	Kunar	1	AF	Afghanistan	KNR	NaN	34.846589	71
18	3900	Kunduz Province	1	AF	Afghanistan	KDZ	NaN	36.728551	68
19	3891	Laghman	1	AF	Afghanistan	LAG	NaN	34.689769	70

In [95]:
```
s.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   id            20 non-null     int64  
 1   name          20 non-null     object 
 2   country_id    20 non-null     int64  
 3   country_code  20 non-null     object 
 4   country_name  20 non-null     object 
 5   state_code    20 non-null     object 
 6   type          0 non-null      object 
 7   latitude      20 non-null     float64
 8   longitude     20 non-null     float64
dtypes: float64(2), int64(2), object(5)
memory usage: 1.5+ KB
```

In [96]:
```
s.describe()
```

Out[96]:

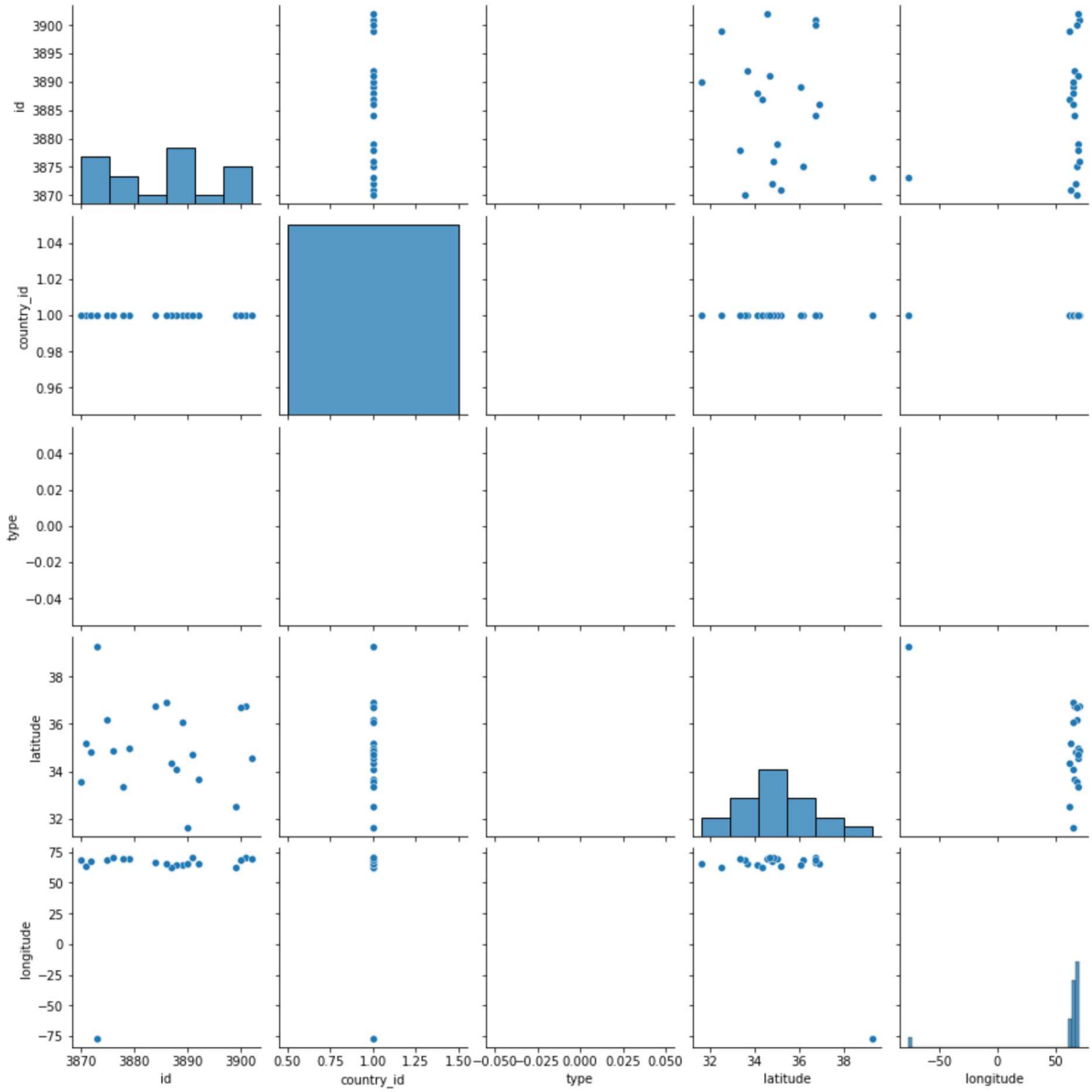
	id	country_id	latitude	longitude
count	20.000000	20.0	20.000000	20.000000
mean	3885.150000	1.0	35.042385	60.022591
std	10.529532	0.0	1.770946	32.274834
min	3870.000000	1.0	31.628871	-76.616047
25%	3875.750000	1.0	33.992057	64.905955
50%	3886.500000	1.0	34.828298	67.359374
75%	3891.250000	1.0	36.316315	69.310979
max	3902.000000	1.0	39.298936	71.097317

In [97]:
```
s.columns
```

Out[97]:
```
Index(['id', 'name', 'country_id', 'country_code', 'country_name',
       'state_code', 'type', 'latitude', 'longitude'],
      dtype='object')
```

In [98]:
```
sns.pairplot(s)
```

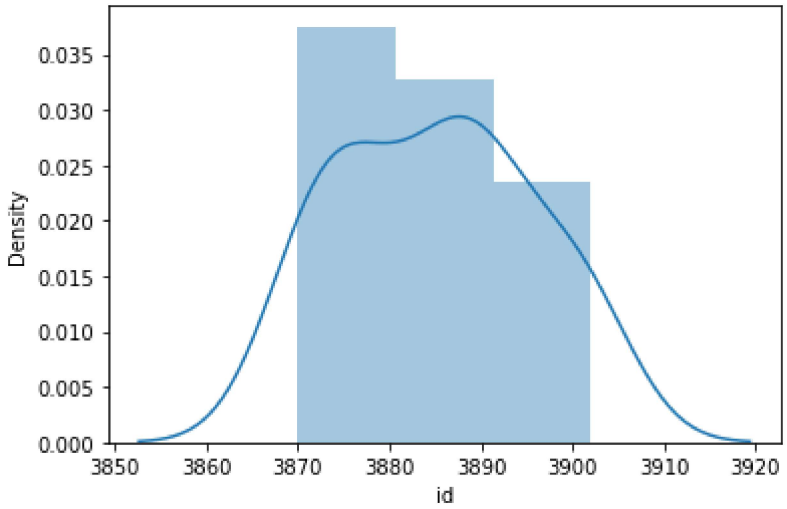
Out[98]: <seaborn.axisgrid.PairGrid at 0x23cf6b95f40>



In [99]:
```
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[99]: <AxesSubplot:xlabel='id', ylabel='Density'>



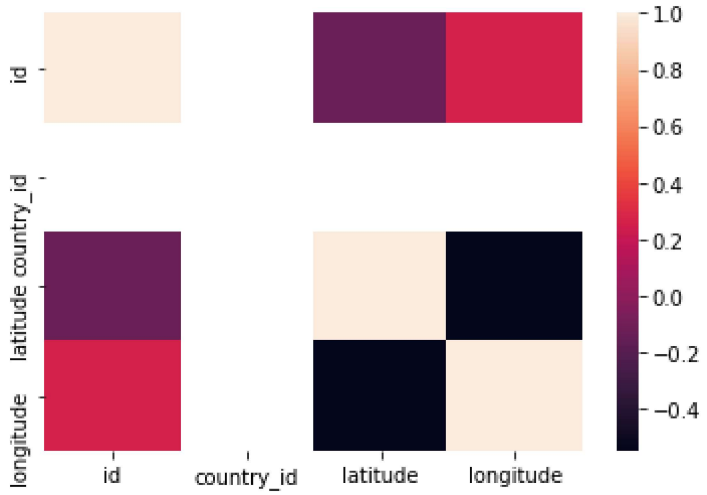
In [101]:
```
s1=s[['id','country_id','latitude', 'longitude']]
s1
```

Out[101]:

	id	country_id	latitude	longitude
0	3901	1	36.734772	70.811995
1	3871	1	35.167134	63.769538
2	3875	1	36.178903	68.745306
3	3884	1	36.755060	66.897537
4	3872	1	34.810007	67.821210
5	3892	1	33.669495	66.046353
6	3899	1	32.495328	62.262663
7	3889	1	36.079561	64.905955
8	3870	1	33.545059	68.417397
9	3888	1	34.099578	64.905955
10	3873	1	39.298936	-76.616047
11	3887	1	34.352865	62.204029
12	3886	1	36.896969	65.665857
13	3902	1	34.555349	69.207486
14	3890	1	31.628871	65.737175
15	3879	1	34.981057	69.621456
16	3878	1	33.333847	69.937167
17	3876	1	34.846589	71.097317
18	3900	1	36.728551	68.867898
19	3891	1	34.689769	70.145580

In [102]:
```
sns.heatmap(s1.corr())
```

Out[102]: <AxesSubplot:>



In [104]:
```
x=s1[['id','country_id','latitude', 'longitude']]
y=s1['id']
```

In [105]:
```
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 [106]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[106]:
```
LinearRegression()
```

In [107]:
```
lr.intercept_
```

Out[107]:
```
9.094947017729282e-13
```

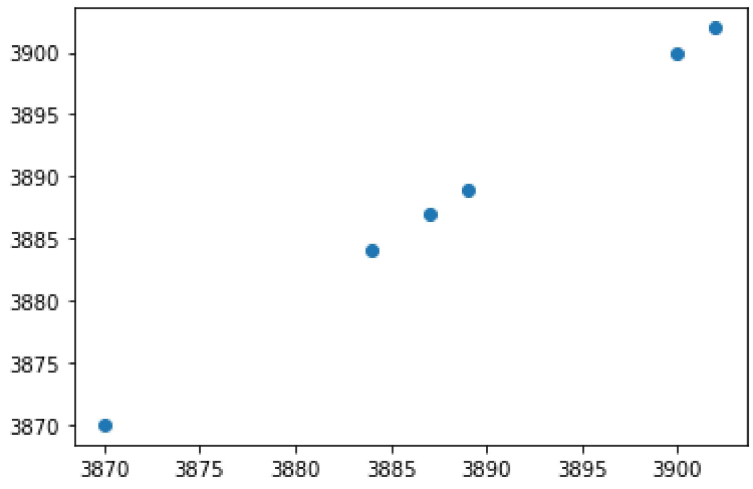
In [108]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[108]:

	Co-efficient
id	1.000000e+00
country_id	-1.429810e-30
latitude	-1.927181e-16
longitude	9.356667e-17

In [109]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[109]:
```
<matplotlib.collections.PathCollection at 0x23cf7d46c10>
```



In [110]:
```
print(lr.score(x_test,y_test))
```

```
1.0
```

In [111]:
```
from sklearn.linear_model import Ridge,Lasso
from sklearn.linear_model import Ridge,Lasso
```

In [112]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

Out[112]: 0.9999034353761045

In [113]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
la.score(x_test,y_test)
```

Out[113]: 0.9860986170714517

In [114]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[114]: ElasticNet()

In [115]:
```
print(en.coef_)
```

```
[ 9.88930404e-01  0.00000000e+00 -0.00000000e+00  4.80275586e-04]
```

In [116]:
```
print(en.intercept_)
```

```
42.96291569625009
```

In [117]:
```
print(en.predict(x_test))
```

```
[3886.9652702  3899.82456595 3888.94442868 3884.00073317 3901.80258985
 3870.15643747]
```

In [118]:
```
print(en.score(x_test,y_test))
```

```
0.9998549822023621
```

In [119]:
```
from sklearn import metrics
```

In [120]:
```
print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error 0.0
```

In [121]:
```
print("Mean squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean squared Error 0.0
```

In [ ]:
```

```

In [122]:
```
import pickle
```

In [123]:
```
filename="prediction"
pickle.dump(lr,open(filename,'wb'))
```

In [124]:
```
import pandas as pd
import pickle
```

In [125]:
```
filename='prediction'
model=pickle.load(open(filename,'rb'))
```

In [126]:
```
real=[[10,20,30,40],[40,55,66,88]]
result=model.predict(real)
```

In [127]:
```
result
```

Out[127]:
```
array([10., 40.])
```

In [ ]:
```

```