D17

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\20_states.csv")
df
```

Out[4]:

	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 [5]:
```
df.head(10)
```

Out[5]:

	id	name	country_id	country_code	country_name	state_code	type	latitude	lon
0	3901	Badakhshan	1	AF	Afghanistan	BDS	NaN	36.734772	70.8
1	3871	Badghis	1	AF	Afghanistan	BDG	NaN	35.167134	63.7
2	3875	Baghlan	1	AF	Afghanistan	BGL	NaN	36.178903	68.7
3	3884	Balkh	1	AF	Afghanistan	BAL	NaN	36.755060	66.8
4	3872	Bamyan	1	AF	Afghanistan	BAM	NaN	34.810007	67.8
5	3892	Daykundi	1	AF	Afghanistan	DAY	NaN	33.669495	66.0
6	3899	Farah	1	AF	Afghanistan	FRA	NaN	32.495328	62.2
7	3889	Faryab	1	AF	Afghanistan	FYB	NaN	36.079561	64.9
8	3870	Ghazni	1	AF	Afghanistan	GHA	NaN	33.545059	68.4
9	3888	Ghōr	1	AF	Afghanistan	GHO	NaN	34.099578	64.9

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5077 entries, 0 to 5076
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   id            5077 non-null   int64  
 1   name          5077 non-null   object 
 2   country_id    5077 non-null   int64  
 3   country_code  5063 non-null   object 
 4   country_name  5077 non-null   object 
 5   state_code    5072 non-null   object 
 6   type          1597 non-null   object 
 7   latitude      5008 non-null   float64
 8   longitude     5008 non-null   float64
dtypes: float64(2), int64(2), object(5)
memory usage: 357.1+ KB
```

In [7]:
```
dff=df.dropna()
```

In [8]:
```
dff.describe()
```

Out[8]:

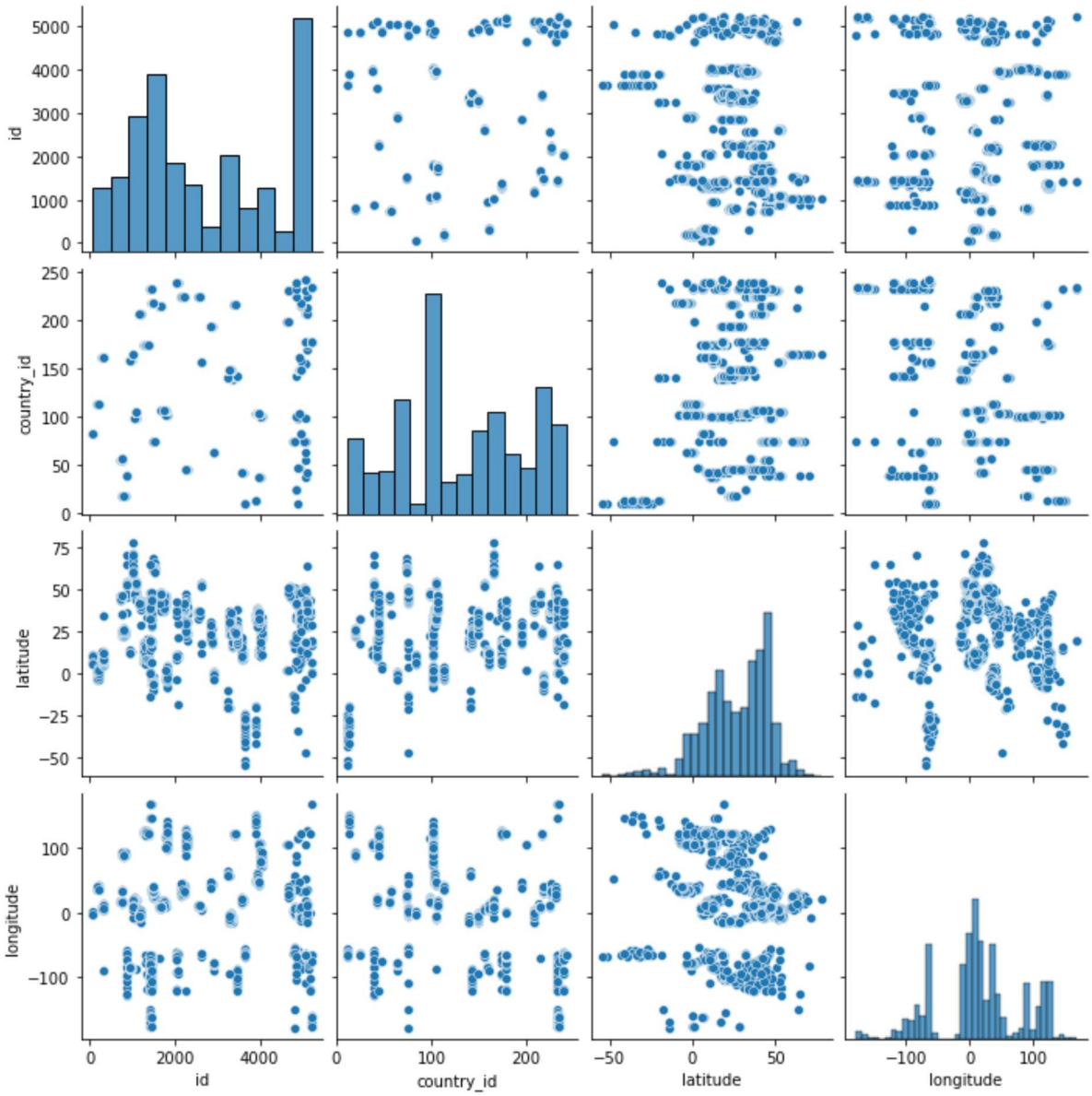
	id	country_id	latitude	longitude
count	1580.000000	1580.000000	1580.000000	1580.000000
mean	2685.916456	134.000633	26.988930	15.009671
std	1611.169440	66.055166	19.635279	66.200355
min	48.000000	11.000000	-54.805400	-178.116500
25%	1339.750000	75.000000	13.752013	-7.622388
50%	2210.500000	139.000000	30.887089	11.665277
75%	4013.250000	178.000000	42.938004	45.682217
max	5220.000000	242.000000	77.874972	166.649935

In [9]:
```
dff.columns
```

Out[9]:
```
Index(['id', 'name', 'country_id', 'country_code', 'country_name',
       'state_code', 'type', 'latitude', 'longitude'],
      dtype='object')
```

In [10]:
```
sns.pairplot(dff)
```

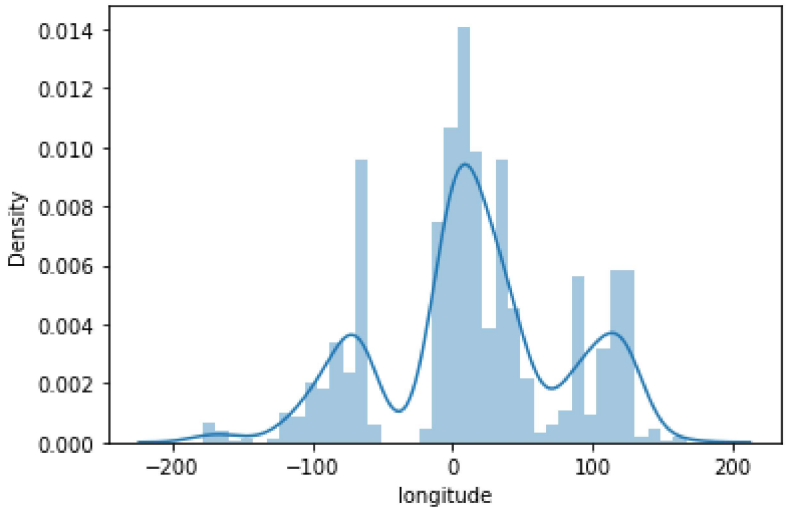
Out[10]: <seaborn.axisgrid.PairGrid at 0x257dacd1310>



In [12]:
```
sns.distplot(dff['longitude'])
```

```
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)
```

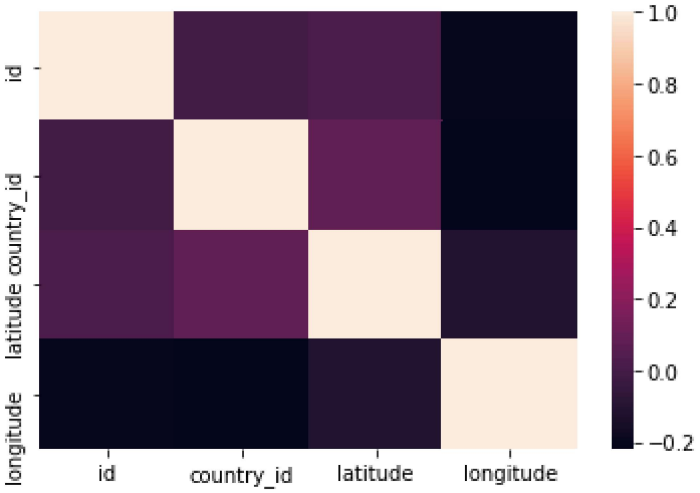
Out[12]: <AxesSubplot:xlabel='longitude', ylabel='Density'>



In [21]:
```
df1=dff[['id', 'name', 'country_id','latitude', 'longitude']]
```

In [22]:
```
sns.heatmap(df1.corr())
```

Out[22]: <AxesSubplot:>



In [24]:
```
x=df1[['id', 'country_id', 'latitude']]
y=df1['longitude']
```

In [25]:
```
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 [26]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[26]:
```
LinearRegression()
```

In [27]:
```
print(lr.intercept_)
```

```
73.68732238398732
```

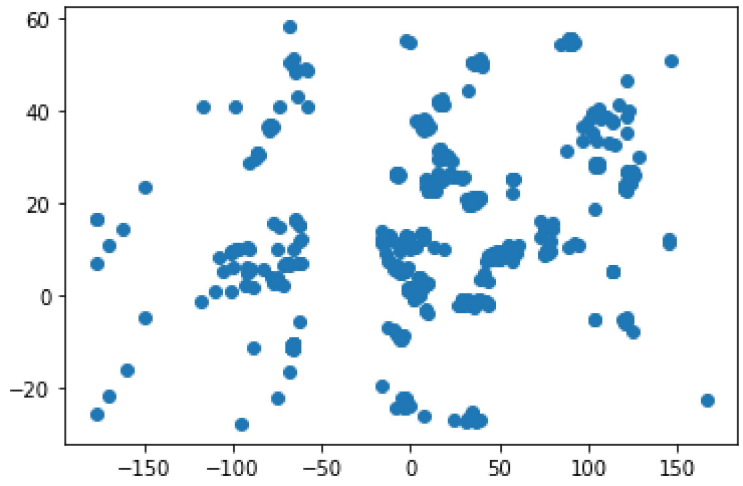
In [28]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[28]:

	Co-efficient
id	-0.008583
country_id	-0.192626
latitude	-0.330770

In [29]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[29]:
```
<matplotlib.collections.PathCollection at 0x257e1b09eb0>
```



In [30]:
```
print(lr.score(x_test,y_test))
```

```
0.0848281224409494
```

In [31]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [32]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[32]:
```
Ridge(alpha=10)
```

In [33]:
```
rr.score(x_test,y_test)
```

Out[33]:
```
0.08482830155137888
```

In [34]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[34]:
```
Lasso(alpha=10)
```

In [35]:
```
la.score(x_test,y_test)
```

Out[35]:
```
0.08524617659173672
```

In [36]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[36]:
```
ElasticNet()
```

In [37]:
```
print(en.coef_)
```

```
[-0.0085836  -0.19253745 -0.32906671]
```

In [38]:
```
print(en.intercept_)
```

```
73.63157790444335
```

In [39]:
```
print(en.predict(x_train))
```

```
[-5.64015808  1.50424685 40.88487469 ... -1.6943404  -8.65480207
 -0.05627478]
```

In [40]:
```
print(en.score(x_train,y_train))
```

```
0.09610391439277566
```

In [41]:
```
from sklearn import metrics
```

In [42]:
```
print("Mean Absolytre Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolytre Error: 49.103874514827076
```

In [43]:
```
print("Mean Square Error:",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Square Error: 4131.857723297188
```

In [44]:
```
print("Root Mean Square Error:",np.sqrt(metrics.mean_absolute_error(y_test,pre
```

```
Root Mean Square Error: 7.00741568017961
```

In [45]:
```
import pickle
```

In [47]:
```
f2="prediction"
pickle.dump(lr,open(f2,'wb'))
```

In [ ]:
```

```