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

In [2]:
```
df=pd.read_csv(r"C:\Users\Admin\Downloads\22_countries.csv")
df
```

Out[2]:

	id	name	iso3	iso2	numeric_code	phone_code	capital	currency	currency_nar
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 Dirha
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 [3]:
```
df1=df.head(100)
```

In [4]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               100 non-null    int64  
 1   name             100 non-null    object 
 2   iso3             100 non-null    object 
 3   iso2             100 non-null    object 
 4   numeric_code     100 non-null    int64  
 5   phone_code       100 non-null    object 
 6   capital          97 non-null     object 
 7   currency         100 non-null    object 
 8   currency_name    100 non-null    object 
 9   currency_symbol  100 non-null    object 
 10  tld              100 non-null    object 
 11  native           99 non-null     object 
 12  region           98 non-null     object 
 13  subregion        97 non-null     object 
 14  timezones        100 non-null    object 
 15  latitude         100 non-null    float64
 16  longitude        100 non-null    float64
 17  emoji            100 non-null    object 
 18  emojiU           100 non-null    object 
dtypes: float64(2), int64(2), object(15)
memory usage: 15.0+ KB
```

In [5]:
```
df1.columns
```

Out[5]:
```
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 [6]:
```
x=df1[['numeric_code','latitude']]
y=df1[['longitude']]
```

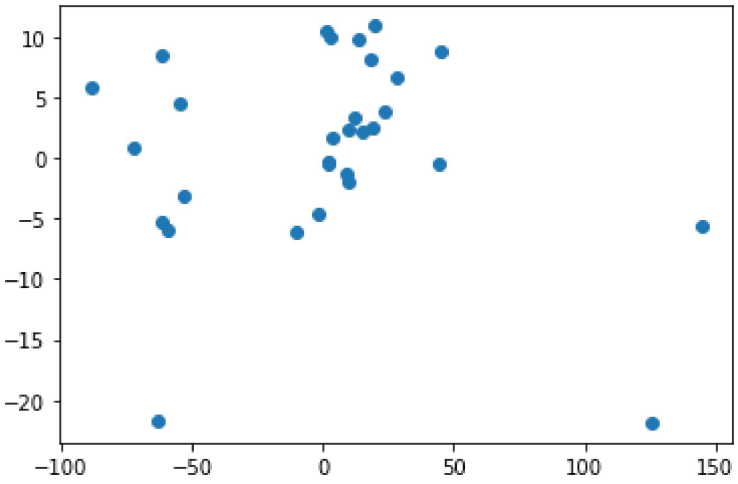
In [7]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [8]:
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(x_train,y_train)
```

Out[8]:
```
LinearRegression()
```

In [9]:
```
prediction= lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[9]: <matplotlib.collections.PathCollection at 0x267ddb422e0>



In [10]:
```
print('Linear Regresion(score):',lr.score(x_test,y_test))
print('Linear Regression(train score)',lr.score(x_train,y_train))
```

```
Linear Regresion(score): -0.05756899428301443
Linear Regression(train score) 0.014278302041548785
```

In [11]:
```
from sklearn.linear_model import Ridge,Lasso
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[11]: Ridge(alpha=10)

In [12]:
```
print('Ridge(test score):',rr.score(x_test,y_test))
```

```
Ridge(test score): -0.05756902281977183
```

In [13]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[13]: Lasso(alpha=10)

In [14]:
```
print('Lasso (test score)',la.score(x_test,y_test))
```

```
Lasso (test score) -0.057534003174442905
```

In [15]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[15]: ElasticNet()

In [16]:
```
print(en.score(x_test,y_test))
```

```
-0.057564297216509575
```

In [17]:
```
import pickle
file="predict"
pickle.dump(lr,open(file,'wb'))
```

In [ ]:
```

```