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

In [3]:
```
d=pd.read_csv(r"C:\Users\user\Downloads\21_cities - 21_cities.csv")
d
```

Out[3]:

	id	name	state_id	state_code	state_name	country_id	country_code	country_name
0	52	Ashkāsham	3901	BDS	Badakhshan	1	AF	Afghanistan
1	68	Fayzabad	3901	BDS	Badakhshan	1	AF	Afghanistan
2	78	Jurm	3901	BDS	Badakhshan	1	AF	Afghanistan
3	84	Khandūd	3901	BDS	Badakhshan	1	AF	Afghanistan
4	115	Rāghistān	3901	BDS	Badakhshan	1	AF	Afghanistan
...	...	...	...	...	...	...	...	...
150449	131496	Redcliff	1957	MI	Midlands Province	247	ZW	Zimbabwe
150450	131502	Shangani	1957	MI	Midlands Province	247	ZW	Zimbabwe
150451	131503	Shurugwi	1957	MI	Midlands Province	247	ZW	Zimbabwe
150452	131504	Shurugwi District	1957	MI	Midlands Province	247	ZW	Zimbabwe
150453	131508	Zvishavane District	1957	MI	Midlands Province	247	ZW	Zimbabwe

150454 rows × 11 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150454 entries, 0 to 150453
Data columns (total 11 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   id            150454 non-null  int64  
 1   name          150454 non-null  object 
 2   state_id      150454 non-null  int64  
 3   state_code    150129 non-null  object 
 4   state_name    150454 non-null  object 
 5   country_id    150454 non-null  int64  
 6   country_code  150406 non-null  object 
 7   country_name  150454 non-null  object 
 8   latitude      150454 non-null  float64
 9   longitude     150454 non-null  float64
 10  wikiDataId    147198 non-null  object 
```

```
dtypes: float64(2), int64(3), object(6)
memory usage: 12.6+ MB
```

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

Out[5]:
```
Index(['id', 'name', 'state_id', 'state_code', 'state_name', 'country_id',
       'country_code', 'country_name', 'latitude', 'longitude', 'wikiDataId'],
      dtype='object')
```

In [6]:
```
d1=d.head(100)
d1
```

Out[6]:

	id	name	state_id	state_code	state_name	country_id	country_code	country_name	latitud
0	52	Ashkāsham	3901	BDS	Badakhshan	1	AF	Afghanistan	36.6833
1	68	Fayzabad	3901	BDS	Badakhshan	1	AF	Afghanistan	37.1166
2	78	Jurm	3901	BDS	Badakhshan	1	AF	Afghanistan	36.8647
3	84	Khandūd	3901	BDS	Badakhshan	1	AF	Afghanistan	36.9512
4	115	Rāghistān	3901	BDS	Badakhshan	1	AF	Afghanistan	37.6607
...	...	...	...	...	...	...	...	...	
95	180	Bashkia Poliçan	629	BR	Berat District	3	AL	Albania	40.5860
96	186	Bashkia Skrapar	629	BR	Berat District	3	AL	Albania	40.5603
97	191	Berat	629	BR	Berat District	3	AL	Albania	40.7058
98	280	Çorovodë	629	BR	Berat District	3	AL	Albania	40.5041
99	219	Kuçovë	629	BR	Berat District	3	AL	Albania	40.8002

100 rows × 11 columns

In [7]:
```
d1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   id            100 non-null    int64  
 1   name          100 non-null    object 
 2   state_id      100 non-null    int64  
 3   state_code    100 non-null    object 
 4   state_name    100 non-null    object 
 5   country_id    100 non-null    int64  
 6   country_code  100 non-null    object 
 7   country_name  100 non-null    object 
 8   latitude      100 non-null    float64
 9   longitude     100 non-null    float64
```

```
 10  wikiDataId    100 non-null    object
dtypes: float64(2), int64(3), object(6)
memory usage: 8.7+ KB
```

In [8]:
```
x=d1[[ 'id','state_id', 'country_id', 'longitude']]
y=d1['latitude']
```

In [9]:
```
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)
```

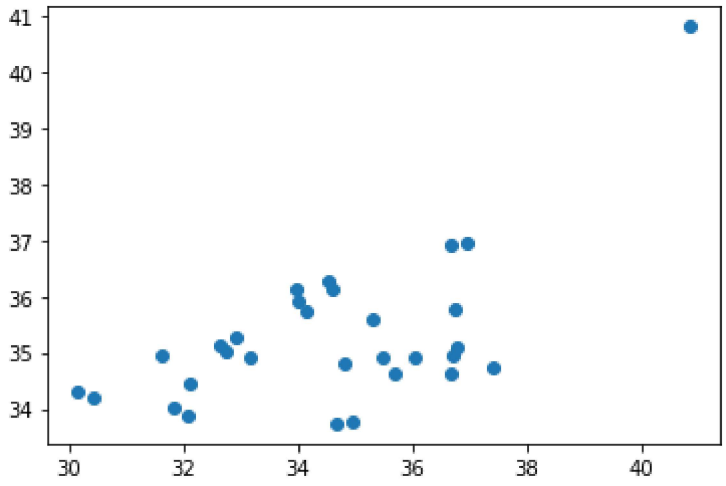
In [10]:
```
from sklearn.linear_model import LinearRegression
```

In [11]:
```
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[11]:
```
LinearRegression()
```

In [14]:
```
prediction =lr.predict(x_test)
py.scatter(y_test,prediction)
```

Out[14]:
```
<matplotlib.collections.PathCollection at 0x203effaabb0>
```



In [15]:
```
print(lr.score(x_test,y_test))
```

```
0.2694790314392367
```

In [16]:
```
print(lr.score(x_train,y_train))
```

```
0.6620595106329579
```

In [17]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [18]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[18]:
```
Ridge(alpha=10)
```

In [19]:
```
rr.score(x_test,y_test)
```

Out[19]:
```
0.3040839048065578
```

In [20]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[20]:
```
Lasso(alpha=10)
```

In [21]:
```
la.score(x_test,y_test)
```

Out[21]:
```
0.1708569275265509
```

In [22]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[22]:
```
ElasticNet()
```

In [23]:
```
print(en.coef_)
```

```
[-0.00463947 -0.0045152   0.          0.18374318]
```

In [24]:
```
print(en.intercept_)
```

```
40.7278854284835
```

In [25]:
```
print(en.predict(x_test))
```

```
[35.44339147 35.30096797 34.10080963 34.76777647 34.7447743  35.34212939
 35.22758113 36.012313   35.70360664 34.86174759 35.24084954 34.01321921
 35.04968063 35.00857302 34.64306617 34.78005829 35.07198363 35.45299977
 35.02424659 40.82358561 35.54128506 34.01370335 34.60621166 35.35242517
 35.32282383 36.01659841 34.69744865 34.03508404 35.75584012 34.79261438]
```

In [26]:
```
print(en.score(x_test,y_test))
```

```
0.2904952854328062
```

In [27]:
```
from sklearn import metrics
```

In [28]:
```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error: 1.664129881043903
```

In [29]:
```
print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error: 3.865186586983291
```

In [30]:
```
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test,prediction))
```

```
Root Mean Squared Error: 1.9660077789732398
```

In [31]:
```
import pickle
```

In [33]:
```
filename="cities"
pickle.dump(lr,open(filename,'wb'))
```

In [ ]:
```

```