In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as py
import seaborn as sns
```

In [2]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\19_nuclear_explosions - 19_nuclear_explosions.c
d
```

Out[2]:

	WEAPON SOURCE COUNTRY	WEAPON DEPLOYMENT LOCATION	Data.Source	Location.Cordinates.Latitude	Location.Cordinates.Longitude
0	USA	Alamogordo	DOE	32.54	-105.57
1	USA	Hiroshima	DOE	34.23	132.27
2	USA	Nagasaki	DOE	32.45	129.52
3	USA	Bikini	DOE	11.35	165.20
4	USA	Bikini	DOE	11.35	165.20
...	...	...	...	...	...
2041	CHINA	Lop Nor	HFS	41.69	88.35
2042	INDIA	Pokhran	HFS	27.07	71.70
2043	INDIA	Pokhran	NRD	27.07	71.70
2044	PAKIST	Chagai	HFS	28.90	64.89
2045	PAKIST	Kharan	HFS	28.49	63.78

2046 rows × 16 columns

In [3]:

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2046 entries, 0 to 2045
Data columns (total 16 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   WEAPON SOURCE COUNTRY          2046 non-null   object 
 1   WEAPON DEPLOYMENT LOCATION     2046 non-null   object 
 2   Data.Source                    2046 non-null   object 
 3   Location.Cordinates.Latitude   2046 non-null   float64
 4   Location.Cordinates.Longitude  2046 non-null   float64
 5   Data.Magnitude.Body            2046 non-null   float64
 6   Data.Magnitude.Surface         2046 non-null   float64
 7   Location.Cordinates.Depth      2046 non-null   float64
 8   Data.Yeild.Lower               2046 non-null   float64
 9   Data.Yeild.Upper               2046 non-null   float64
 10  Data.Purpose                   2046 non-null   object 
 11  Data.Name                      2046 non-null   object 
 12  Data.Type                      2046 non-null   object 
 13  Date.Day                       2046 non-null   int64  
 14  Date.Month                     2046 non-null   int64  
 15  Date.Year                        2046 non-null   int64  
dtypes: float64(7), int64(3), object(6)
memory usage: 255.9+ KB
```

In [4]:

```
d.columns
```

Out[4]:

```
Index(['WEAPON SOURCE COUNTRY', 'WEAPON DEPLOYMENT LOCATION', 'Data.Source',
       'Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
       'Data.Magnitude.Body', 'Data.Magnitude.Surface',
       'Location.Cordinates.Depth', 'Data.Yeild.Lower', 'Data.Yeild.Upper',
       'Data.Purpose', 'Data.Name', 'Data.Type', 'Date.Day', 'Date.Month',
       'Date.Year'],
      dtype='object')
```

In [5]:

```
d1=d.head(100)
d1
```

Out[5]:

	WEAPON SOURCE COUNTRY	WEAPON DEPLOYMENT LOCATION	Data.Source	Location.Cordinates.Latitude	Location.Cordinates.Longitude
0	USA	Alamogordo	DOE	32.54	-105.57
1	USA	Hiroshima	DOE	34.23	132.27
2	USA	Nagasaki	DOE	32.45	129.52
3	USA	Bikini	DOE	11.35	165.20
4	USA	Bikini	DOE	11.35	165.20
...	...	...	...	...	...
95	USSR	Semi Kazakh	DOE	50.00	78.00
96	USA	Nts	DOE	37.00	-116.00
97	USSR	Kazakh	MTM	46.00	61.00
98	USSR	Semi Kazakh	DOE	50.00	78.00
99	USSR	Semi Kazakh	DOE	50.00	78.00

100 rows × 16 columns

In [6]:

```
d1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 16 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   WEAPON SOURCE COUNTRY          100 non-null    object 
 1   WEAPON DEPLOYMENT LOCATION     100 non-null    object 
 2   Data.Source                    100 non-null    object 
 3   Location.Cordinates.Latitude   100 non-null    float64
 4   Location.Cordinates.Longitude  100 non-null    float64
 5   Data.Magnitude.Body            100 non-null    float64
 6   Data.Magnitude.Surface         100 non-null    float64
 7   Location.Cordinates.Depth      100 non-null    float64
 8   Data.Yeild.Lower                 100 non-null    float64
 9   Data.Yeild.Upper                 100 non-null    float64
 10  Data.Purpose                     100 non-null    object
 11  Data.Name                        100 non-null    object
 12  Data.Type                        100 non-null    object
 13  Date.Day                         100 non-null    int64
 14  Date.Month                       100 non-null    int64
 15  Date.Year                        100 non-null    int64
dtypes: float64(7), int64(3), object(6)
memory usage: 12.6+ KB
```

In [7]:

```
x=d1[[  'Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
       'Data.Magnitude.Body', 'Data.Magnitude.Surface',
       'Location.Cordinates.Depth', 'Data.Yeild.Lower', 'Data.Yeild.Upper','Date.Day',
       'Date.Year']]
y=d1['Location.Cordinates.Longitude']
```

In [8]:

```
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 [9]:

```
from sklearn.linear_model import LinearRegression
```

In [10]:

```
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[10]:

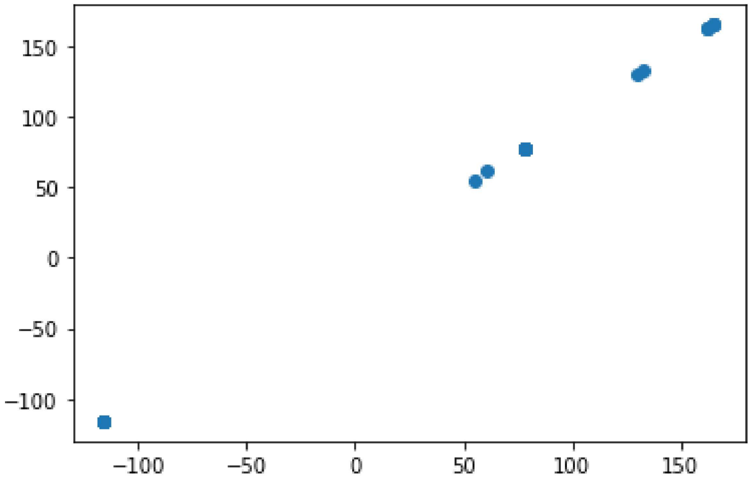
```
LinearRegression()
```

In [13]:

```
prediction =lr.predict(x_test)
py.scatter(y_test,prediction)
```

Out[13]:

```
<matplotlib.collections.PathCollection at 0x22d1dffb100>
```



In [14]:

```
print(lr.score(x_test,y_test))
```

```
1.0
```

In [15]:

```
print(lr.score(x_train,y_train))
```

```
1.0
```

In [16]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [17]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[17]:

```
Ridge(alpha=10)
```

In [18]:

```
rr.score(x_test,y_test)
```

Out[18]:

```
0.9999999998608537
```

In [19]:

```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[19]:

```
Lasso(alpha=10)
```

In [20]:

```
la.score(x_test,y_test)
```

Out[20]:

```
0.9999993187710697
```

In [21]:

```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[21]:

```
ElasticNet()
```

In [22]:

```
print(en.coef_)
```

```
[ 0.00000000e+00  9.99926983e-01  0.00000000e+00  0.00000000e+00
  0.00000000e+00  1.49611139e-06 -1.80911530e-07 -0.00000000e+00
  0.00000000e+00 -0.00000000e+00]
```

In [23]:

```
print(en.intercept_)
```

```
-0.0019103221962488703
```

In [24]:

```
print(en.predict(x_test))
```

```
[-115.99343908 -115.99344013  129.50866018 -115.99341146  165.18605495
  162.13654595   77.99240083   77.99272318 -115.99343119   77.99240753
  162.13847272   77.99247592  165.20575533 -115.99343908  132.33843645
   60.99363606   77.99243121 -115.99342593 -115.99343908 -115.99338384
 -115.99343513 -115.99342987 -115.99344039  162.13629869  165.19510221
   54.99412637 -115.99341146   77.99239964   77.99240201  165.18605495]
```

In [25]:

```
print(en.score(x_test,y_test))
```

```
0.9999999947312163
```

In [26]:

```
from sklearn import metrics
```

In [27]:

```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error: 2.9250675955457456e-13
```

In [28]:

```
print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error: 1.4304846038334128e-25
```

In [29]:

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

```
Root Mean Squared Error: 3.782174776280722e-13
```

In [30]:

```
import pickle
```

In [32]:

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

In [ ]:

```

```