In [171]:

```
# IMPORT LIBRARIES
import numpy  as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [172]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\19_nuclear_explosions.csv")
a
```

Out[172]:

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

2046 rows × 16 columns

In [173]:

```
a=a.head(10)
a
```

Out[173]:

Body	Data.Magnitude.Surface	Location.Cordinates.Depth	Data.Yeild.Lower	Data.Yeild.Upper	Dat
0.0	0.0	-0.10	21.0	21.0	
0.0	0.0	-0.60	15.0	15.0	
0.0	0.0	-0.60	21.0	21.0	
0.0	0.0	-0.20	21.0	21.0	
0.0	0.0	0.03	21.0	21.0	
0.0	0.0	-0.08	37.0	37.0	
0.0	0.0	-0.08	49.0	49.0	
0.0	0.0	-0.08	18.0	18.0	
0.0	0.0	0.00	22.0	22.0	
0.0	0.0	-0.35	1.0	1.0	

In [174]:

```
# to find
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   WEAPON SOURCE COUNTRY           10 non-null     object 
 1   WEAPON DEPLOYMENT LOCATION      10 non-null     object 
 2   Data.Source                     10 non-null     object 
 3   Location.Cordinates.Latitude    10 non-null     float64
 4   Location.Cordinates.Longitude   10 non-null     float64
 5   Data.Magnitude.Body             10 non-null     float64
 6   Data.Magnitude.Surface          10 non-null     float64
 7   Location.Cordinates.Depth       10 non-null     float64
 8   Data.Yeild.Lower                10 non-null     float64
 9   Data.Yeild.Upper                10 non-null     float64
 10  Data.Purpose                    10 non-null     object 
 11  Data.Name                       10 non-null     object 
 12  Data.Type                       10 non-null     object 
 13  Date.Day                        10 non-null     int64  
 14  Date.Month                      10 non-null     int64  
 15  Date.Year                       10 non-null     int64  
dtypes: float64(7), int64(3), object(6)
memory usage: 1.4+ KB
```

In [175]:

```
# to display summary of statastic
a.describe()
```

Out[175]:

	Location.Cordinates.Latitude	Location.Cordinates.Longitude	Data.Magnitude.Body	Data.
count	10.000000	10.000000	10.0	
mean	24.082000	93.307000	0.0	
std	14.133627	111.078447	0.0	
min	11.300000	-116.000000	0.0	
25%	11.312500	89.380000	0.0	
50%	21.900000	147.210000	0.0	
75%	33.807500	162.150000	0.0	
max	48.000000	165.200000	0.0	

In [176]:

```
# to display colum heading
a.columns
```

Out[176]:

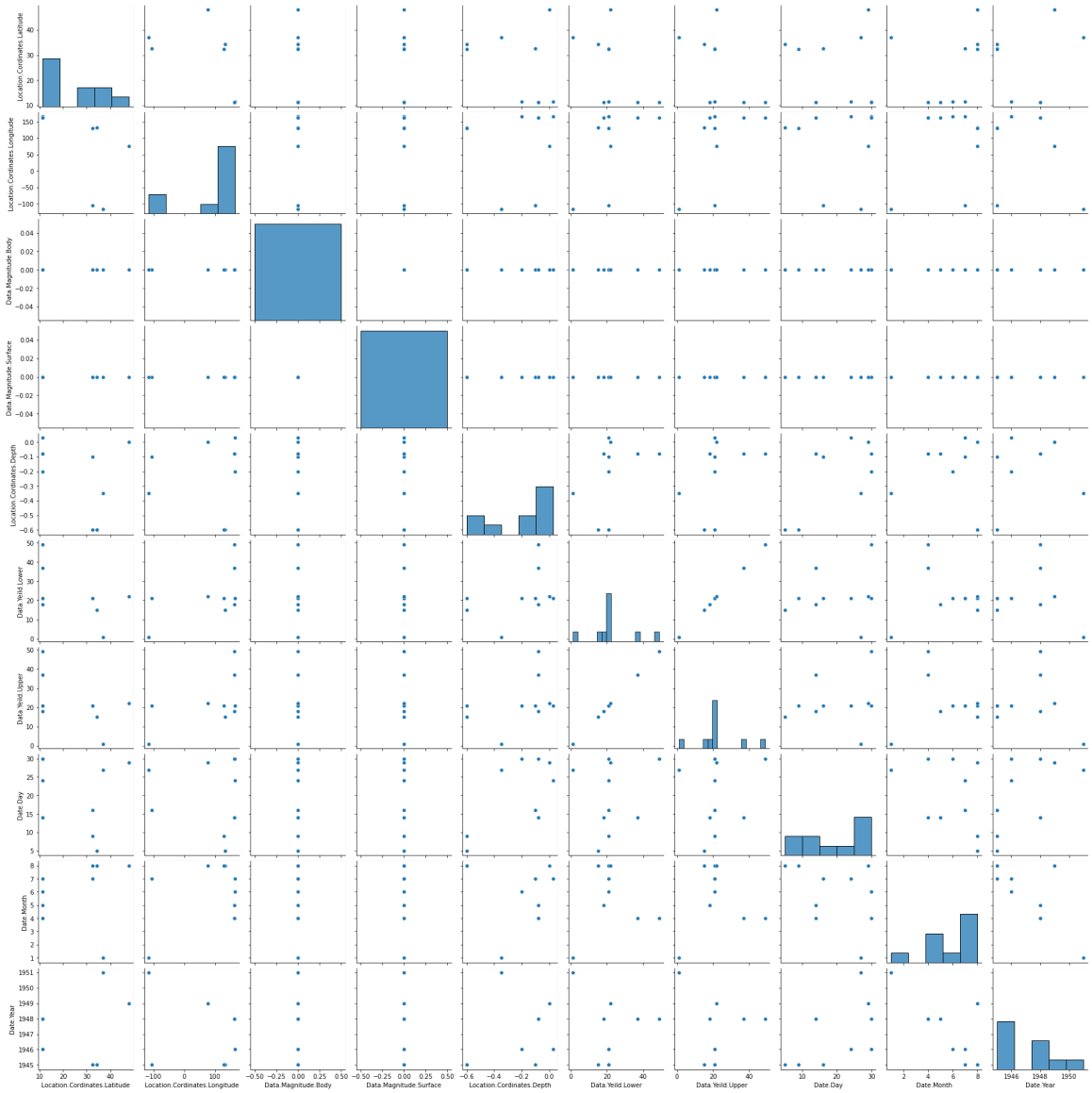
```
Index(['WEAPON SOURCE COUNTRY', 'WEAPON DEPLOYMENT LOCATION', 'Data.Sourc
e',
       'Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
       'Data.Magnitude.Body', 'Data.Magnitude.Surface',
       'Location.Cordinates.Depth', 'Data.Yeild.Lower', 'Data.Yeild.Uppe
r',
       'Data.Purpose',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

In [177]:

```
sns.pairplot(a)
```

Out[177]:

```
<seaborn.axisgrid.PairGrid at 0x243f31f3cd0>
```

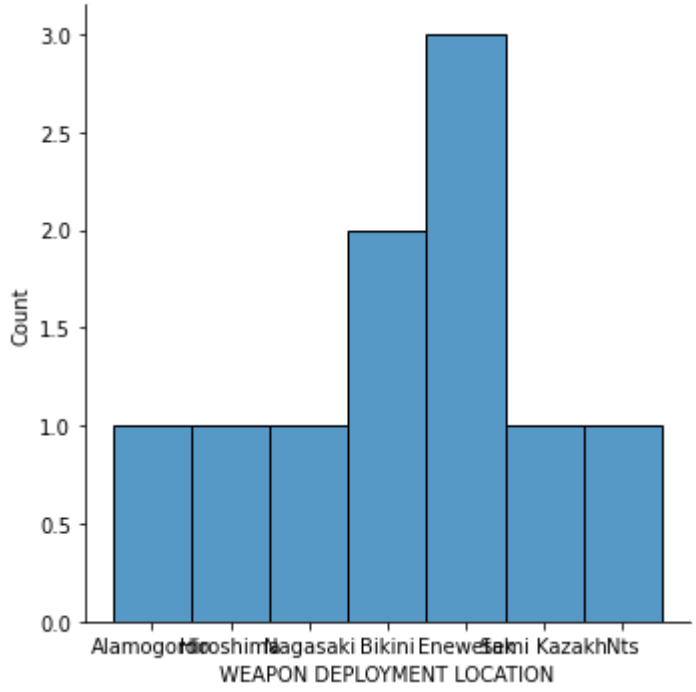


In [178]:

```
sns.displot(a["WEAPON DEPLOYMENT LOCATION"])
```

Out[178]:

```
<seaborn.axisgrid.FacetGrid at 0x243f6c30280>
```



In [179]:

```
b=a[['WEAPON SOURCE COUNTRY', 'WEAPON DEPLOYMENT LOCATION', 'Data.Source',
       'Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
       'Data.Magnitude.Body', 'Data.Magnitude.Surface',
       'Location.Cordinates.Depth']]
b
```

Out[179]:

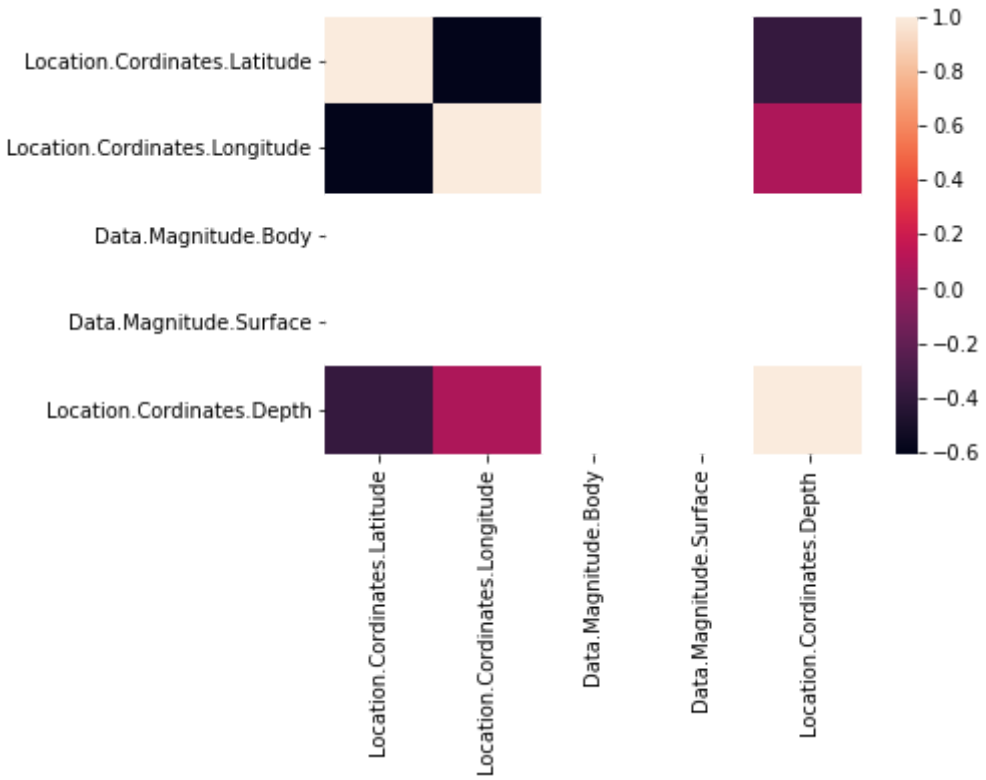
	WEAPON SOURCE COUNTRY	WEAPON DEPLOYMENT LOCATION	Data.Source	Location.Cordinates.Latitude	Location.Cordinates.Lc
0	USA	Alamogordo	DOE	32.54	
1	USA	Hiroshima	DOE	34.23	
2	USA	Nagasaki	DOE	32.45	
3	USA	Bikini	DOE	11.35	
4	USA	Bikini	DOE	11.35	
5	USA	Enewetak	DOE	11.30	
6	USA	Enewetak	DOE	11.30	
7	USA	Enewetak	DOE	11.30	
8	USSR	Semi Kazakh	DOE	48.00	
9	USA	Nts	DOE	37.00	

In [180]:

```
sns.heatmap(b.corr())
```

Out[180]:

<AxesSubplot:>



In [182]:

```
x=a[['Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
       'Data.Magnitude.Body', 'Data.Magnitude.Surface',
       'Location.Cordinates.Depth', 'Data.Yeild.Lower']]
y=a['Data.Magnitude.Surface']
```

In [183]:

```
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 [184]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[184]:

LinearRegression()

In [185]:

```
lr.intercept_
```

Out[185]:

```
0.0
```

In [186]:

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

Out[186]:

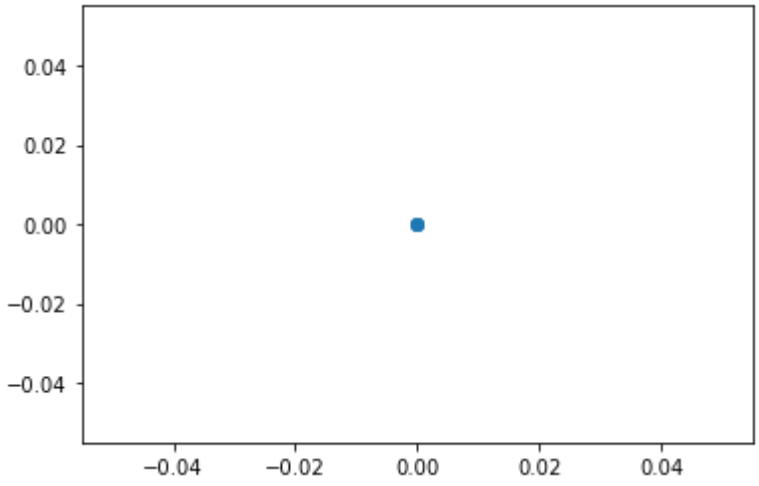
	Co-efficient
Location.Cordinates.Latitude	0.0
Location.Cordinates.Longitude	0.0
Data.Magnitude.Body	0.0
Data.Magnitude.Surface	0.0
Location.Cordinates.Depth	0.0
Data.Yeild.Lower	0.0

In [187]:

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

Out[187]:

```
<matplotlib.collections.PathCollection at 0x243f8b88fa0>
```



In [188]:

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

Out[188]:

```
1.0
```

In [189]:

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

Out[189]:

```
1.0
```

In [190]:

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

In [191]:

```
rr=Ridge(alpha=10)
rr.fit(x_test,y_test)
```

Out[191]:

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

In [192]:

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

Out[192]:

```
1.0
```

In [193]:

```
la=Lasso(alpha=10)
la.fit(x_test,y_test)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 0.0, tolerance:
0.0
  model = cd_fast.enet_coordinate_descent(
```

Out[193]:

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

In [194]:

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

Out[194]:

```
1.0
```

In [195]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 0.0, tolerance:
0.0
  model = cd_fast.enet_coordinate_descent(
```

Out[195]:

```
ElasticNet()
```

In [196]:

```
en.coef_
```

Out[196]:

```
array([0., 0., 0., 0., 0., 0.])
```

In [197]:

```
en.intercept_
```

Out[197]:

```
0.0
```

In [198]:

```
prediction=en.predict(x_test)
prediction
```

Out[198]:

```
array([0., 0., 0.])
```

In [199]:

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

Out[199]:

```
1.0
```

EVALUATION METRICS

In [200]:

```
from sklearn import metrics
```

In [201]:

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

```
Mean Absolute Error: 0.0
```

In [202]:

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

```
Mean Squared Error 0.0
```

In [203]:

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

```
Root Mean Squared Error 0.0
```

MODEL SAVING

In [204]:

```
import pickle
```

In [205]:

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

In [206]:

```
import pandas as pd
import pickle
```

In [207]:

```
filename='prediction'
model=pickle.load(open(filename,'rb'))
```

In [209]:

```
real=[[10,20,30,40,50,60],[13,23,33,43,53,63]]
result=model.predict(real)
result
```

Out[209]:

```
array([0., 0.])
```

In [ ]:

```

```