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

In [2]:
```
df=pd.read_csv("19_nuclear_explosions.csv")
df.fillna(0,inplace=True)
df
```

Out[2]:

ody	Data.Magnitude.Surface	Location.Cordinates.Depth	Data.Yeild.Lower	Data.Yeild.Upper	Data.Pur
0.0	0.0	-0.10	21.0	21.0	
0.0	0.0	-0.60	15.0	15.0	Cc
0.0	0.0	-0.60	21.0	21.0	Cc
0.0	0.0	-0.20	21.0	21.0	
0.0	0.0	0.03	21.0	21.0	
...	...	...	...	...	
5.3	0.0	0.00	3.0	12.0	
5.3	0.0	0.00	0.0	20.0	
0.0	0.0	0.00	0.0	1.0	
0.0	0.0	0.00	0.0	35.0	
5.0	0.0	0.00	0.0	18.0	

In [3]:
```
df.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
```

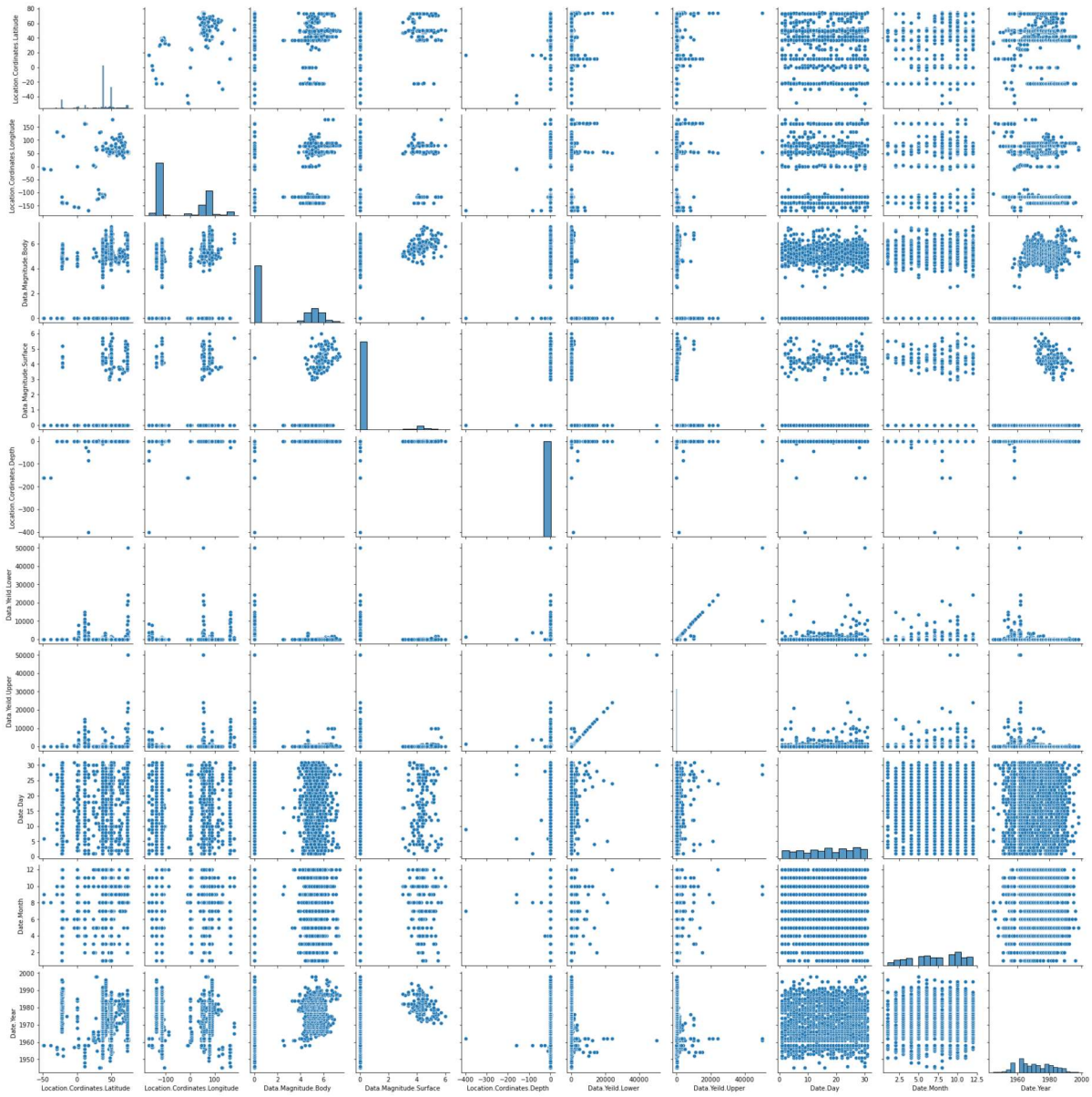
In [4]:
```
df.describe()
```

Out[4]:

tes.Longitude	Data.Magnitude.Body	Data.Magnitude.Surface	Location.Cordinates.Depth	Data.Yeild.L
2046.000000	2046.000000	2046.000000	2046.000000	2046.00
-36.015037	2.145406	0.356696	-0.490829	208.44
100.829355	2.625453	1.203569	10.981072	1641.96
-169.320000	0.000000	0.000000	-400.000000	0.00
-116.051500	0.000000	0.000000	0.000000	0.00
-116.000000	0.000000	0.000000	0.000000	0.00
78.000000	5.100000	0.000000	0.000000	20.00
179.220000	7.400000	6.000000	1.451000	50000.00

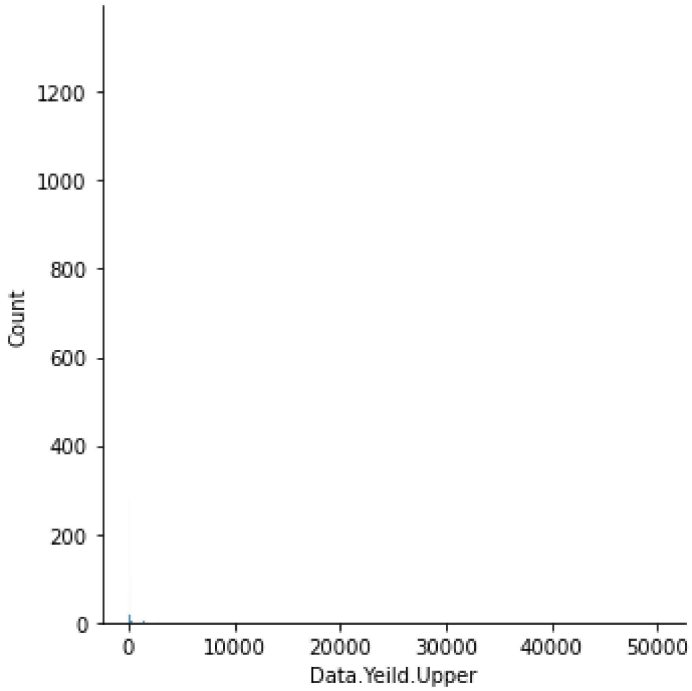
In [5]:
```
sns.pairplot(df)
```

Out[5]:
```
<seaborn.axisgrid.PairGrid at 0x28249786640>
```



In [7]:
```
sns.displot(df['Data.Yeild.Upper'])
```

Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x2825b6faaf0>
```



In [8]:
```
df1=df.drop(['WEAPON SOURCE COUNTRY'],axis=1)
df1
```

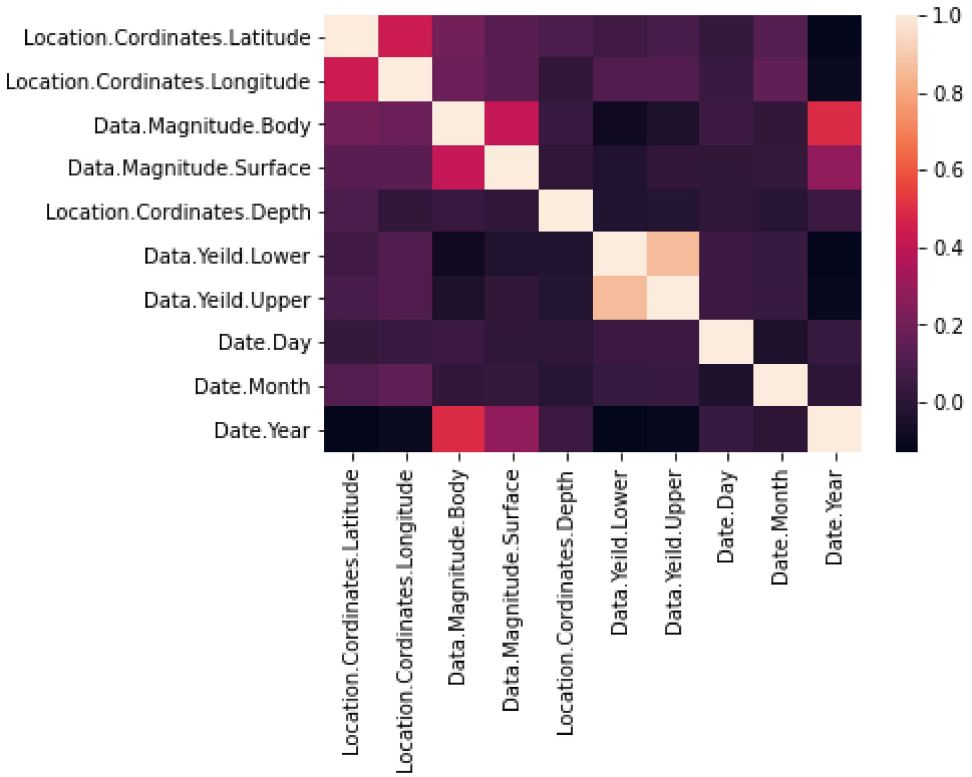
Out[8]:

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

2046 rows × 15 columns

In [9]:
```
sns.heatmap(df1.corr())
```

Out[9]: <AxesSubplot:>



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

In [11]:
```
y=df['Data.Yeild.Upper']
x=df1.drop(['WEAPON DEPLOYMENT LOCATION','Data.Source','Data.Purpose','Data.Nam
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

```
      Location.Cordinates.Latitude  Location.Cordinates.Longitude  \
1315                        49.920                         78.91
407                          2.000                       -157.00
761                        -22.000                       -139.00
1030                       -22.000                       -139.00
131                         50.000                         78.00
...                            ...                           ...
1242                        37.200                       -116.20
202                         11.350                        165.20
98                          50.000                         78.00
1874                        37.225                       -116.08
482                         16.450                       -169.32

      Data.Magnitude.Body  Data.Magnitude.Surface  Location.Cordinates.Depth
\
1315                  5.8                     6.0                      0.000
407                   0.0                     0.0                      0.000
761                   0.0                     0.0                      0.000
1030                  0.0                     0.0                      0.000
131                   0.0                     0.0                      0.000
...                   ...                     ...                        ...
1242                  5.0                     0.0                      0.000
202                   0.0                     0.0                     -0.001
98                    0.0                     0.0                      0.000
1874                  0.0                     0.0                      0.000
482                   0.0                     0.0                      0.000

      Data.Yeild.Lower  Data.Yeild.Upper  Date.Day  Date.Month  Date.Year
1315              20.0             150.0        29          10       1975
407               50.0              50.0        11           5       1962
761                0.0             200.0        19           7       1966
1030               0.0            1000.0        22           5       1970
131               42.0              42.0         3           4       1957
...                ...               ...       ...         ...        ...
1242               0.0              20.0        19           6       1974
202             1360.0            1360.0        11           5       1958
98                14.0              14.0        16           3       1956
1874               0.0              20.0         3           2       1987
482               75.0              75.0         2          10       1962

[1432 rows x 10 columns]
```

In [12]:
```
model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

Out[12]:
```
-3.956301952712238e-11
```

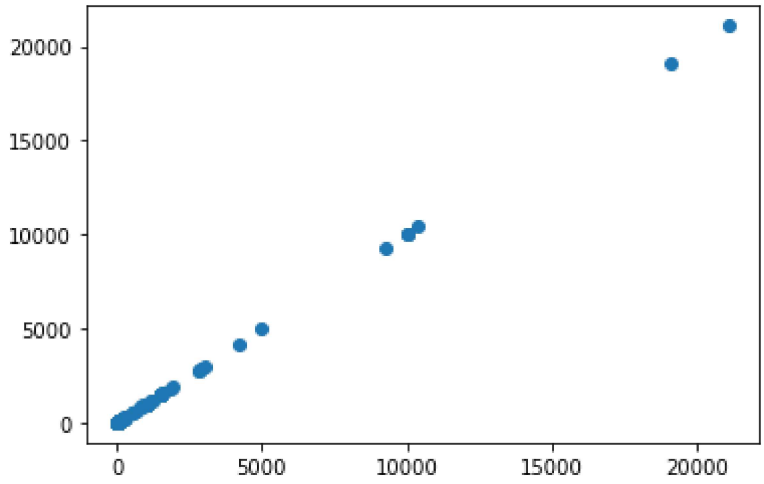
In [13]:
```
coeff=pd.DataFrame(model.coef_,x.columns,columns=["Coefficient"])
coeff
```

Out[13]:

	Coefficient
Location.Cordinates.Latitude	1.270081e-14
Location.Cordinates.Longitude	-2.864375e-14
Data.Magnitude.Body	2.073151e-13
Data.Magnitude.Surface	1.605360e-13
Location.Cordinates.Depth	1.811745e-14
Data.Yeild.Lower	6.161738e-15
Data.Yeild.Upper	1.000000e+00
Date.Day	7.631070e-14
Date.Month	-5.972978e-14
Date.Year	1.812179e-14

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

Out[14]: <matplotlib.collections.PathCollection at 0x2826015dbb0>



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

Out[15]: 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]:
```
1.0
```

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]:
```
0.9999999927707098
```

In [21]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
print(en.coef_)
print(en.intercept_)
print(en.predict(x_test))
print(en.score(x_test,y_test))
from sklearn import metrics
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test,pred
```

```
-0.000947414896245391
[ 1.49980247e+03  2.00023813e+01  1.99917400e+01  1.49987197e+02
  1.49983902e+02  1.49971766e+01  2.00023870e+01  1.99917328e+01
  1.40064758e+01 -4.22391883e-03  7.99077040e+00 -1.22346316e-03
  1.49976549e+02  2.00023999e+01  1.99906671e+01  4.90578450e+00
  2.00023758e+01  1.99979618e+02  1.49987222e+02  2.00036368e+01
  5.99649298e+00  1.49976512e+02  2.00023758e+01  6.01233382e-03
  1.01061795e+01  3.94806618e-01  1.99917290e+01  2.00023868e+01
  1.99979605e+02  8.09963156e+01  1.99979605e+02  2.34794464e-01
  2.00023758e+01  1.59012911e+03  1.10062479e+01  2.00023857e+01
  8.49668288e+00  1.49976502e+02  2.00023758e+01  9.89552825e+00
  3.99990757e+01  2.00023759e+01  9.99263012e+00  1.49976501e+02
  2.09915907e+01  1.99917477e+01  1.99917383e+01  2.99209839e+00
  2.00023758e+01  1.99930667e+01  1.99979606e+02  1.99930298e+01
  1.99917526e+01  2.00023785e+01  5.35746322e-03  2.34794464e-01
  1.99917400e+01  2.00023758e+01  2.00036203e+01  2.00023758e+01
  1.99917252e+01  2.55009533e+02  2.00023758e+01  1.19662179e+00
  1.09909983e+01  1.99916832e+01  1.69492467e+00  1.99917509e+01
  1.20063238e+01  1.99979607e+02  1.49976508e+02  1.60011767e+03
```

In [24]:
```
import pickle
filename='prediction'
pickle.dump(model,open(filename,'wb'))
model = pickle.load(open(filename,'rb'))
real = [[10,20,30,40,50,60,70,80,90,100],[12,13,21,43,12,12,34,53,56,12]]
result = model.predict(real)
result
```

Out[24]:
```
array([70., 34.])
```

In [ ]: