31-07-2023

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

In [390]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\19_nuclear_explosions.csv")
a
```

Out[390]:

	WEAPON SOURCE COUNTRY	WEAPON DEPLOYMENT LOCATION	Data.Source	Location.Cordinates.Latitude	Location.Cordinates.Longitude	Da
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 [391]:

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

Out[391]:

Latitude	Location.Cordinates.Longitude	Data.Magnitude.Body	Data.Magnitude.Surface	Location.Cordinates.Dept
32.54	-105.57	0.0	0.0	-0.1
34.23	132.27	0.0	0.0	-0.6
32.45	129.52	0.0	0.0	-0.6
11.35	165.20	0.0	0.0	-0.2
11.35	165.20	0.0	0.0	0.0
11.30	162.15	0.0	0.0	-0.0
11.30	162.15	0.0	0.0	-0.0
11.30	162.15	0.0	0.0	-0.0
48.00	76.00	0.0	0.0	0.0
37.00	-116.00	0.0	0.0	-0.3

In [392]:

```
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 [393]:

```
a.columns
```

Out[393]:

```
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 [394]:

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

Out[394]:

	WEAPON SOURCE COUNTRY	WEAPON DEPLOYMENT LOCATION	Data.Source	Location.Cordinates.Latitude	Location.Cordinates.Longitude	Data.I
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	
5	USA	Enewetak	DOE	11.30	162.15	
6	USA	Enewetak	DOE	11.30	162.15	
7	USA	Enewetak	DOE	11.30	162.15	
8	USSR	Semi Kazakh	DOE	48.00	76.00	
9	USA	Nts	DOE	37.00	-116.00	

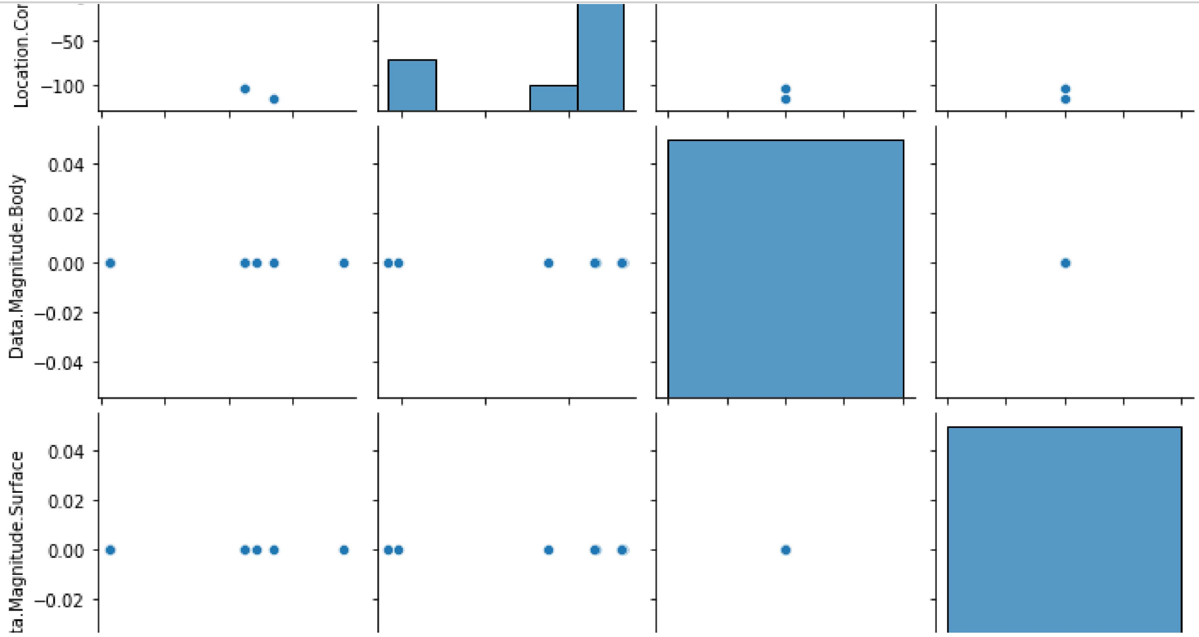
In [395]:

```
d.describe()
```

Out[395]:

	Location.Cordinates.Latitude	Location.Cordinates.Longitude	Data.Magnitude.Body	Data.Magnitude.Surfa
count	10.000000	10.000000	10.0	1
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	

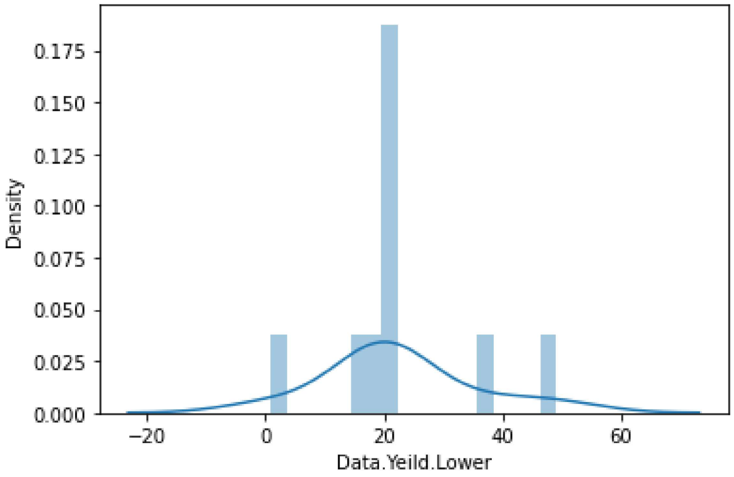
In [396]:
```
sns.pairplot(d)
```



In [397]:
```
sns.distplot(a['Data.Yeild.Lower'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Please
adapt your code to use either `displot` (a figure-level function with similar flexibil
ity) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

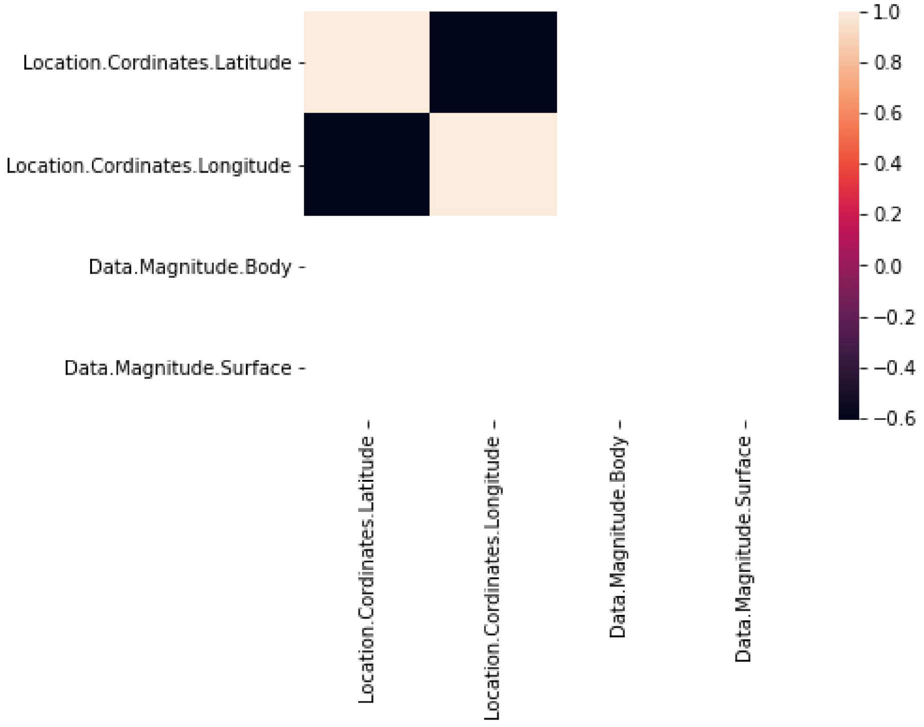
Out[397]:
```
<AxesSubplot:xlabel='Data.Yeild.Lower', ylabel='Density'>
```



In [404]:
```
x1=a[['Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
      'Data.Magnitude.Body', 'Data.Magnitude.Surface']]
```

In [405]:
```
sns.heatmap(x1.corr())
```

Out[405]:
```
<AxesSubplot:>
```



In [406]:
```
x=a[['Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
       'Data.Magnitude.Body', 'Data.Magnitude.Surface']]
y=a['Data.Yeild.Lower']
```

In [407]:
```
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 [408]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[408]:
```
LinearRegression()
```

In [409]:
```
print(lr.intercept_)
```

```
24.08485018600524
```

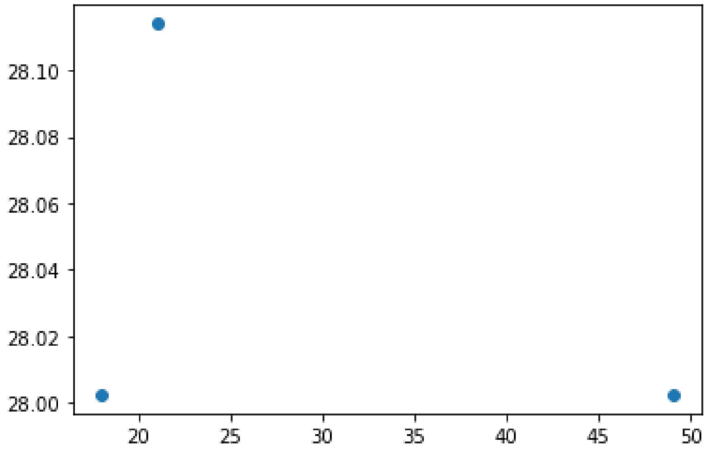
In [410]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[410]:

	Co-efficient
Location.Cordinates.Latitude	-0.234774
Location.Cordinates.Longitude	0.040521
Data.Magnitude.Body	0.000000
Data.Magnitude.Surface	0.000000

In [411]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[411]:
```
<matplotlib.collections.PathCollection at 0x190c53e6220>
```



In [412]:
```
print(lr.score(x_test,y_test))
```

```
-0.011790784384103636
```

In [413]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [414]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[414]:
```
Ridge(alpha=10)
```

In [415]:
```
rr.score(x_test,y_test)
```

Out[415]:
```
-0.012285783192243382
```

In [416]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[416]:
```
Lasso(alpha=10)
```

In [417]:
```
la.score(x_test,y_test)
```

Out[417]:
```
-0.03278531441016108
```

In [418]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[418]:
```
ElasticNet()
```

In [419]:
```
print(en.coef_)
```

```
[-0.22993566  0.04073522  0.          0.        ]
```

```
In [420]: print(en.intercept_)

          23.92826707308187
```

```
In [421]: print(en.predict(x_test))

          [28.04795525 27.93520962 27.93520962]
```

```
In [422]: en.score(x_test,y_test)

Out[422]: -0.012726551515849005
```

```
In [423]: from sklearn import metrics
```

```
In [424]: print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))

          Mean Absolute Error 12.704724310837811
```

```
In [425]: print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))

          Mean Squared Error 197.18678175663533
```

```
In [426]: print(" Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction))

           Root Mean Squared Error 14.042321095767441
```

```
In [427]: import pickle
```

```
In [428]: filename="prediction"
          pickle.dump(lr,open(filename,'wb'))
```

```
In [429]: import pandas as pd
          import pickle
```

```
In [430]: filename="prediction"
          model=pickle.load(open(filename,"rb"))
```

```
In [434]: real=[[10,20,24,25],[15,30,36,40]]
          result=model.predict(real)
```

```
In [435]: result

Out[435]: array([22.54752329, 21.77885984])
```