```
#Linear Regression
import pandas as pd
```

```
import numpy as numpy
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
covid=pd.read_csv('/content/Covid_datset.zip')
```

```
covid
```

Saved successfully! ×

	Sno	Date	Time	State_UnionTerritory	ConfirmedIndianNational	Confirn
0	447	3/29/2020	7:30 PM	Andhra Pradesh	25	
1	448	3/29/2020	7:30 PM	Andaman and Nicobar Islands	5	
2	449	3/29/2020	7:30 PM	Bihar	3	
3	450	3/29/2020	7:30 PM	Chandigarh	23	
4	451	3/29/2020	7:30 PM	Chhattisgarh	27	
5	452	3/29/2020	7:30 PM	Delhi	15	
6	453	3/29/2020	7:30 PM	Goa	10	
7	454	3/29/2020	7:30 PM	Gujarat	16	
8	455	3/29/2020	7:30 PM	Haryana	40	
9	456	3/29/2020	7:30 PM	Himachal Pradesh	15	
10	457	3/29/2020	7:30 PM	Jammu and Kashmir	15	
11	458	3/29/2020	7:30 PM	Karnataka	18	
12	459	3/29/2020	7:30 PM	Kerala	8	
13	460	3/29/2020	7:30 PM	Ladakh	9	
14	461	3/29/2020	7:30 PM	Madhya Pradesh	8	
				Maharashtra	6	
16	463	3/29/2020	7:30 PM	Manipur	21	
17	464	3/29/2020	7:30 PM	Mizoram	20	
18	465	3/29/2020	7:30 PM	Odisha	23	

Saved successfully! ×

```
covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54 entries, 0 to 53
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Sno                       54 non-null     int64 
 1   Date                      54 non-null     object
 2   Time                      54 non-null     object
 3   State_UnionTerritory      54 non-null     object
 4   ConfirmedIndianNational   54 non-null     int64 
 5   ConfirmedForeignNational  54 non-null     int64 
 6   Cured                     54 non-null     int64 
 7   Deaths                    54 non-null     int64 
 8   Confirmed                 54 non-null     int64 
dtypes: int64(6), object(3)
memory usage: 3.9+ KB
```

```
covid.describe()
```

	Sno	ConfirmedIndianNational	ConfirmedForeignNational	Cured	
count	54.000000	54.000000	54.000000	54.000000	5
mean	473.500000	19.685185	11.907407	3.666667	
std	15.732133	13.095908	7.549256	6.407543	
min	447.000000	2.000000	1.000000	0.000000	
25%	460.250000	9.000000	6.000000	0.000000	
50%	473.500000	19.000000	12.500000	1.000000	
75%	486.750000	26.000000	17.000000	3.750000	
max	500.000000	65.000000	32.000000	25.000000	

```
covid.corr()
```

Saved successfully!

	Sno	ConfirmedIndianNational	ConfirmedForeignNatior
Sno	1.000000	0.125968	0.0024
ConfirmedIndianNational	0.125968	1.000000	0.0441
ConfirmedForeignNational	0.002462	0.044167	1.0000
Cured	0.050724	-0.033653	0.0274
Deaths	0.007600	-0.139222	-0.1823
Confirmed	0.120759	-0.064536	0.0335

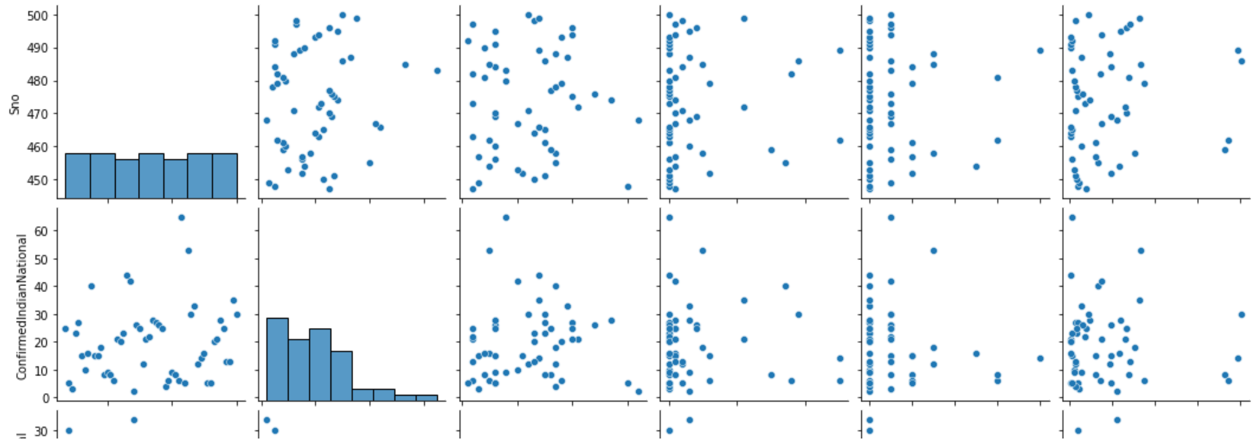
```
covid.columns
```

```
     Index(['Sno', 'Date', 'Time', 'State_UnionTerritory',
            'ConfirmedIndianNational', 'ConfirmedForeignNational', 'Cured',
            'Deaths', 'Confirmed'],
           dtype='object')
```

```
sns.pairplot(covid)
```

Saved successfully! ×

```
<seaborn.axisgrid.PairGrid at 0x7faf12cd4310>
```



```
covid.columns
```

```
Index(['Sno', 'Date', 'Time', 'State_UnionTerritory',
       'ConfirmedIndianNational', 'ConfirmedForeignNational', 'Cured',
       'Deaths', 'Confirmed'],
      dtype='object')
```

```
y=covid['ConfirmedIndianNational']
```

```
y.shape
```

```
(54,)
```

```
X=covid[['Deaths']]
```

```
X.shape
```

```
(54, 1)
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.7,random_state=2529)
```

Saved successfully! ×

```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

```
((37, 1), (17, 1), (37,), (17,))
```

```
X_train
```

```
31  0
30  0
 4  0
 3  0
18  0
33  0
10  2
38  3
 2  1
35  0
28  0
27  0
22  1
52  0
 7  5
41  3
53  1
13  0
37  2
40  0
23  1
25  0
43  0
21  0
24  0
 8  0
47  1
36  1
15  6
50  1
32  2
```

Saved successfully! ×

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

```
reg=LinearRegression()
```

```
  reg.fit(X_train,y_train)

     LinearRegression()
```

```
reg.intercept_

     20.5944229707344
```

```
reg.coef_

     array([-1.28188846])
```

```
y_predict=reg.predict(X_test)
```

```
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error ,r2_score
```

```
mean_absolute_error(y_test,y_predict)

     8.981907948159938
```

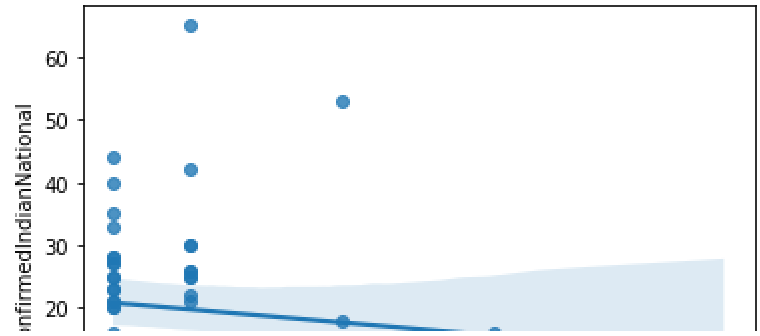
```
mean_absolute_percentage_error(y_test,y_predict)

     0.7249269264351927
```

Saved successfully! ×

```
sns.regplot(x='Deaths',y='ConfirmedIndianNational',data=covid)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7faf0c8e9790>
```



```
#multiple regression
```

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

```
covid.head()
```

	Sno	Date	Time	State_UnionTerritory	ConfirmedIndianNational	ConfirmedF
0	447	3/29/2020	7:30 PM	Andhra Pradesh	25	
1	448	3/29/2020	7:30 PM	Andaman and Nicobar Islands	5	
2	449	3/29/2020	7:30 PM	Bihar	3	

Saved successfully! ×

```
covid.describe()
```

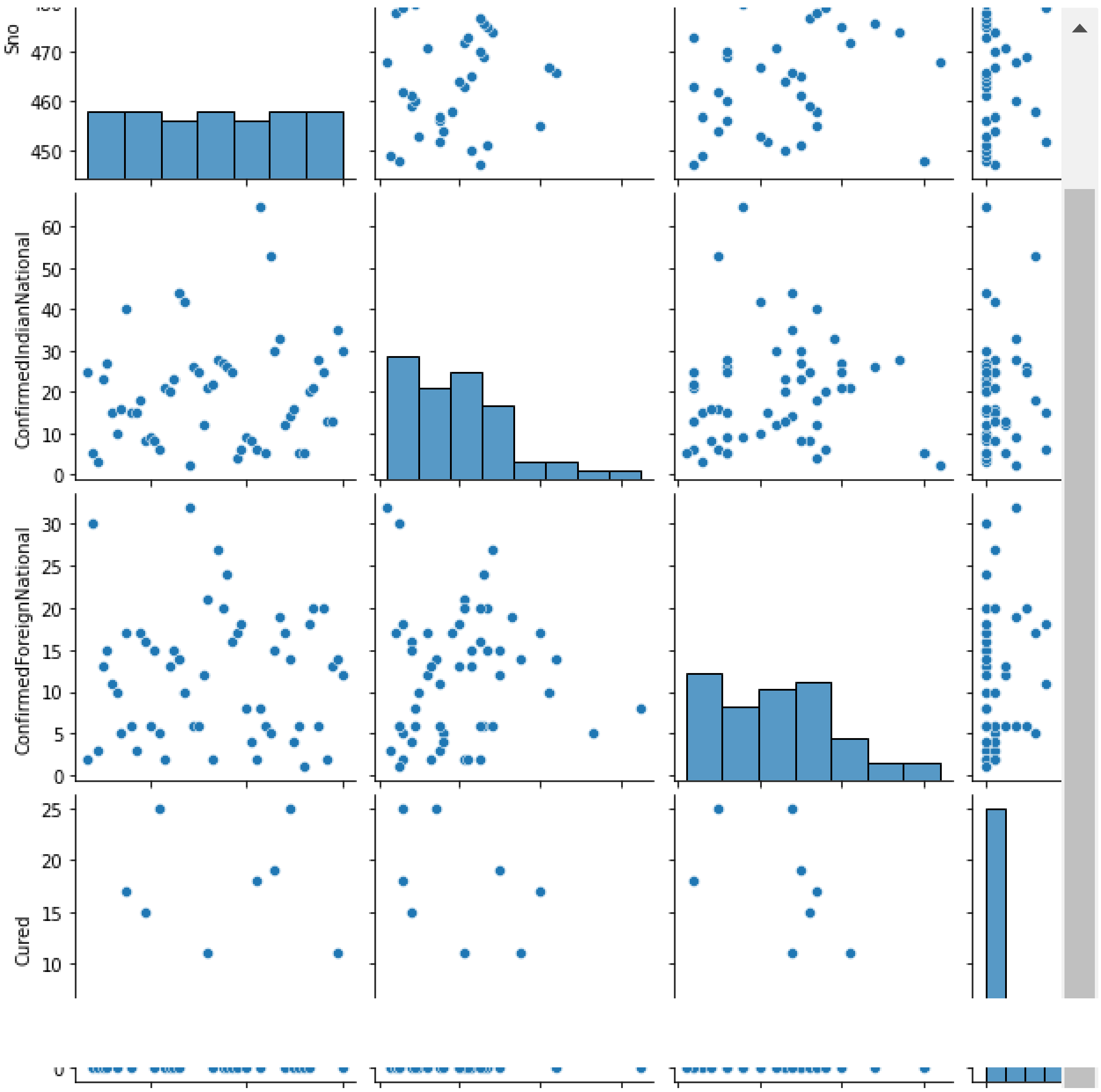
	Sno	ConfirmedIndianNational	ConfirmedForeignNational	Cured	
count	54.000000	54.000000	54.000000	54.000000	5
mean	473.500000	19.685185	11.907407	3.666667	

```
covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54 entries, 0 to 53
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Sno                       54 non-null     int64 
 1   Date                      54 non-null     object
 2   Time                      54 non-null     object
 3   State_UnionTerritory      54 non-null     object
 4   ConfirmedIndianNational   54 non-null     int64 
 5   ConfirmedForeignNational  54 non-null     int64 
 6   Cured                     54 non-null     int64 
 7   Deaths                    54 non-null     int64 
 8   Confirmed                 54 non-null     int64 
dtypes: int64(6), object(3)
memory usage: 3.9+ KB
```

```
sns.pairplot(df)
```

Saved successfully! ×



```
covid.columns
```

```
Index(['Sno', 'Date', 'Time', 'State_UnionTerritory',
       'ConfirmedIndianNational', 'ConfirmedForeignNational', 'Cured',
       'Deaths', 'Confirmed'],
      dtype='object')
```

```
covid['ConfirmedIndianNational']]
```

Saved successfully! ×

```
X=covid[['Sno',
        'ConfirmedForeignNational', 'Cured',
       'Deaths', 'Confirmed']]
```

```
X.shape
```

```
(54, 5)
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.7,random_state=2529)
```

```
from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
```

```
X_train=sc.fit_transform(X_train)
```

```
X_test=sc.fit_transform(X_test)
```

```
X_train
```

```
array([[ 0.45268182, -1.02896166, -0.37515299,  3.04072305,  0.92865847],
       [ 1.20468795, -1.43254733, -0.54866124, -0.64802295, -0.85559329],
       [-0.91460205,  0.45085246, -0.54866124,  0.58155905, -0.12567211],
       [-0.70950947,  0.18179535, -0.54866124, -0.64802295, -0.90966153],
       [-1.25642301, -0.75990455, -0.54866124, -0.03323195, -0.85559329],
       [ 0.24758924,  0.71990957, -0.54866124, -0.64802295, -0.74745682],
       [ 0.17922505,  0.58538101, -0.54866124, -0.64802295, -0.7204227 ],
       [-1.59824398,  0.45085246, -0.54866124, -0.64802295, -0.74745682],
       [-1.66660817,  0.18179535, -0.54866124, -0.64802295, -0.7204227 ],
       [-0.64114527,  0.45085246, -0.54866124, -0.64802295, -0.85559329],
       [ 0.38431763, -0.49084743, -0.54866124, -0.64802295, -0.80152506],
       [-1.18805882, -1.16349022, -0.37515299,  0.58155905, -0.098638  ],
       [ 0.72613859, -0.8944331 ,  0.31888004,  1.19635005,  1.30713612],
       [-1.73497237, -1.16349022, -0.54866124, -0.03323195, -0.63932035],
       [ 0.52104601, -1.29801877,  2.57448737, -0.64802295,  0.03653259],
       [ 0.04249666,  1.12349524, -0.54866124, -0.64802295, -0.69338859],
       [-0.02586753,  2.06519513, -0.37515299, -0.64802295, -0.31491094],
       [-0.3676885 , -0.75990455,  0.14537178, -0.03323195,  0.38797612],
       [ 1.6832373 ,  0.3163239 ,  1.35992958, -0.64802295,  1.280102  ],
       [-1.3931514 , -0.8944331 , -0.37515299,  2.42593205,  0.63128318],
       [ 0.93123117,  0.71990957, -0.54866124,  1.19635005,  0.33390789],
       [ 1.7516015 ,  0.04726679, -0.54866124, -0.03323195, -0.34194506],
       [-0.98296624, -0.75990455, -0.02813647, -0.64802295, -0.58525212],
       [ 0.6577744 , -0.75990455, -0.20164473,  0.58155905,  0.360942  ],
       [ 0.86286698,  0.98896668, -0.02813647, -0.64802295, -0.58525212],
       [-0.29932431, -0.75990455, -0.37515299, -0.03323195,  0.84755612],
       [-0.16259592,  1.2580238 ,  1.35992958, -0.64802295,  0.820522  ],
       [ 1.06795956, -1.02896166, -0.54866124, -0.64802295, -0.90966153],
                            91, -0.02813647, -0.64802295,  0.55018083],
                            79, -0.20164473, -0.64802295, -0.74745682],
                            13, -0.54866124, -0.64802295, -0.90966153],
       [-1.32478721,  0.71990957,  2.40097911, -0.64802295, -0.04456976],
       [ 1.34141633,  1.12349524, -0.37515299, -0.03323195,  0.09060083],
       [ 0.58941021, -0.49084743, -0.54866124, -0.03323195, -0.85559329],
       [-0.84623785, -0.8944331 ,  3.78904517,  3.04072305,  4.09165024],
       [ 1.54650892, -1.29801877, -0.37515299, -0.03323195,  0.98272671],
       [ 0.31595343,  0.85443813,  0.49238829,  0.58155905,  1.41527259]])
```

Saved successfully! ×

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

```
Model=LinearRegression()
```