In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [75]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\2_2015.csv")
a
```

Out[75]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(G
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	
...	...	...	...	...	...	...	...	...	...	
		Sub-								

In [76]:
```
a=a.head(50)
a
```

Out[76]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	
5	Finland	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	0.64169	
6	Netherlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	0.61576	
7	Sweden	Western Europe	8	7.364	0.03157	1.33171	1.28907	0.91087	0.65980	
8	New Zealand	Australia and New Zealand	9	7.286	0.03371	1.25018	1.31967	0.90837	0.63938	
9	Australia	Australia and New Zealand	10	7.284	0.04083	1.33358	1.30923	0.93156	0.65124	
10	Israel	Middle East and Northern Africa	11	7.278	0.03470	1.22857	1.22393	0.91387	0.41319	
11	Costa Rica	Latin America and Caribbean	12	7.226	0.04454	0.95578	1.23788	0.86027	0.63376	
12	Austria	Western Europe	13	7.200	0.03751	1.33723	1.29704	0.89042	0.62433	
13	Mexico	Latin America and Caribbean	14	7.187	0.04176	1.02054	0.91451	0.81444	0.48181	
14	United States	North America	15	7.119	0.03839	1.39451	1.24711	0.86179	0.54604	
15	Brazil	Latin America and Caribbean	16	6.983	0.04076	0.98124	1.23287	0.69702	0.49049	
16	Luxembourg	Western Europe	17	6.946	0.03499	1.56391	1.21963	0.91894	0.61583	
17	Ireland	Western Europe	18	6.940	0.03676	1.33596	1.36948	0.89533	0.61777	
18	Belgium	Western Europe	19	6.937	0.03595	1.30782	1.28566	0.89667	0.58450	
19	United Arab Emirates	Middle East and Northern Africa	20	6.901	0.03729	1.42727	1.12575	0.80925	0.64157	
20	United Kingdom	Western Europe	21	6.867	0.01866	1.26637	1.28548	0.90943	0.59625	
21	Oman	Middle East and Northern Africa	22	6.853	0.05335	1.36011	1.08182	0.76276	0.63274	

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
22	Venezuela	Latin America and Caribbean	23	6.810	0.06476	1.04424	1.25596	0.72052	0.42908
23	Singapore	Southeastern Asia	24	6.798	0.03780	1.52186	1.02000	1.02525	0.54252
24	Panama	Latin America and Caribbean	25	6.786	0.04910	1.06353	1.19850	0.79661	0.54210
25	Germany	Western Europe	26	6.750	0.01848	1.32792	1.29937	0.89186	0.61477
26	Chile	Latin America and Caribbean	27	6.670	0.05800	1.10715	1.12447	0.85857	0.44132
27	Qatar	Middle East and Northern Africa	28	6.611	0.06257	1.69042	1.07860	0.79733	0.64040
28	France	Western Europe	29	6.575	0.03512	1.27778	1.26038	0.94579	0.55011
29	Argentina	Latin America and Caribbean	30	6.574	0.04612	1.05351	1.24823	0.78723	0.44974
30	Czech Republic	Central and Eastern Europe	31	6.505	0.04168	1.17898	1.20643	0.84483	0.46364
31	Uruguay	Latin America and Caribbean	32	6.485	0.04539	1.06166	1.20890	0.81160	0.60362
32	Colombia	Latin America and Caribbean	33	6.477	0.05051	0.91861	1.24018	0.69077	0.53466
33	Thailand	Southeastern Asia	34	6.455	0.03557	0.96690	1.26504	0.73850	0.55664
34	Saudi Arabia	Middle East and Northern Africa	35	6.411	0.04633	1.39541	1.08393	0.72025	0.31048
35	Spain	Western Europe	36	6.329	0.03468	1.23011	1.31379	0.95562	0.45951
36	Malta	Western Europe	37	6.302	0.04206	1.20740	1.30203	0.88721	0.60365
37	Taiwan	Eastern Asia	38	6.298	0.03868	1.29098	1.07617	0.87530	0.39740
38	Kuwait	Middle East and Northern Africa	39	6.295	0.04456	1.55422	1.16594	0.72492	0.55499
39	Suriname	Latin America and Caribbean	40	6.269	0.09811	0.99534	0.97200	0.60820	0.59657
40	Trinidad and Tobago	Latin America and Caribbean	41	6.168	0.10895	1.21183	1.18354	0.61483	0.55884
41	El Salvador	Latin America and Caribbean	42	6.130	0.05618	0.76454	1.02507	0.67737	0.40350

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(
42	Guatemala	Latin America and Caribbean	43	6.123	0.05224	0.74553	1.04356	0.64425	0.57733	
43	Uzbekistan	Central and Eastern Europe	44	6.003	0.04361	0.63244	1.34043	0.59772	0.65821	
44	Slovakia	Central and Eastern Europe	45	5.995	0.04267	1.16891	1.26999	0.78902	0.31751	
45	Japan	Eastern Asia	46	5.987	0.03581	1.27074	1.25712	0.99111	0.49615	
46	South Korea	Eastern Asia	47	5.984	0.04098	1.24461	0.95774	0.96538	0.33208	
47	Ecuador	Latin America and Caribbean	48	5.975	0.04528	0.86402	0.99903	0.79075	0.48574	
48	Bahrain	Middle East and Northern Africa	49	5.960	0.05412	1.32376	1.21624	0.74716	0.45492	
49	Italy	Western Europe	50	5.948	0.03914	1.25114	1.19777	0.95446	0.26236	

```
In [77]: a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Country                        50 non-null     object 
 1   Region                         50 non-null     object 
 2   Happiness Rank                 50 non-null     int64  
 3   Happiness Score                50 non-null     float64
 4   Standard Error                 50 non-null     float64
 5   Economy (GDP per Capita)       50 non-null     float64
 6   Family                         50 non-null     float64
 7   Health (Life Expectancy)       50 non-null     float64
 8   Freedom                        50 non-null     float64
 9   Trust (Government Corruption)  50 non-null     float64
 10  Generosity                     50 non-null     float64
 11  Dystopia Residual              50 non-null     float64
dtypes: float64(9), int64(1), object(2)
memory usage: 4.8+ KB
```

```
In [78]: a.columns
```

```
Out[78]: Index(['Country', 'Region', 'Happiness Rank', 'Happiness Score',
                'Standard Error', 'Economy (GDP per Capita)', 'Family',
                'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)',
                'Generosity', 'Dystopia Residual'],
               dtype='object')
```

In [79]:
```
a.head()
```

Out[79]:

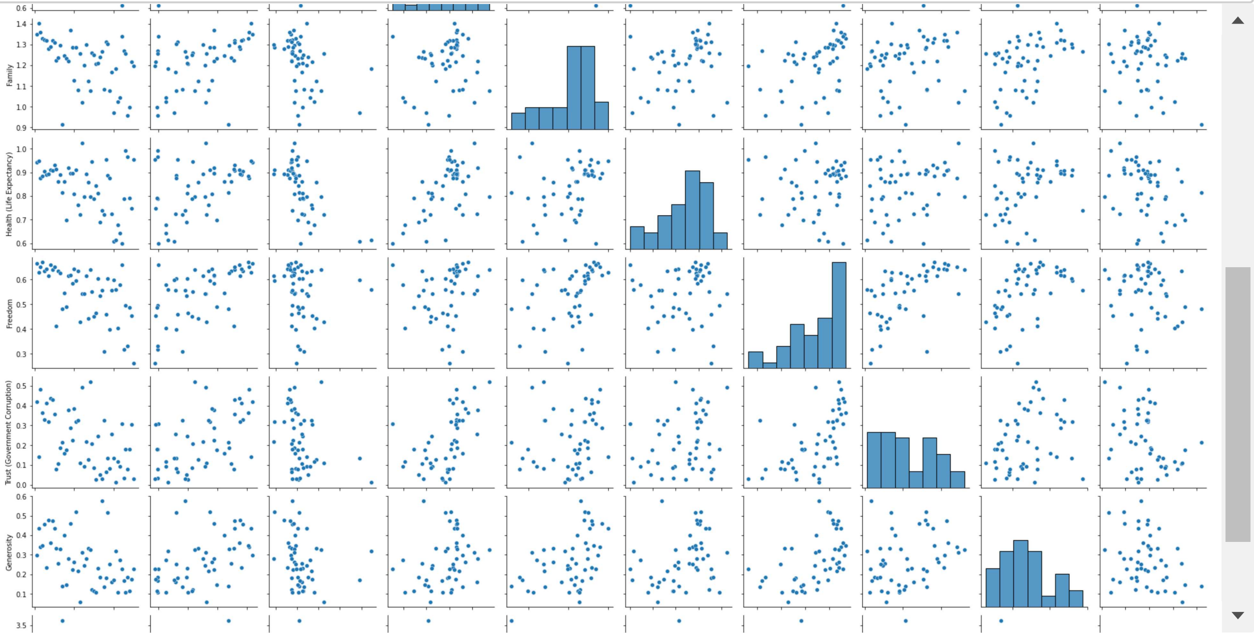
	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(Goveri Corru
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.

In [80]:
```
a.describe()
```

Out[80]:

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Gene
count	50.00000	50.000000	50.000000	50.000000	50.000000	50.000000	50.000000	50.000000	50.0
mean	25.50000	6.729040	0.043584	1.217752	1.212277	0.835402	0.543408	0.221732	0.2
std	14.57738	0.511449	0.015482	0.216687	0.118959	0.105584	0.105589	0.142510	0.1
min	1.00000	5.948000	0.018480	0.632440	0.914510	0.597720	0.262360	0.011400	0.0
25%	13.25000	6.299000	0.035540	1.062128	1.135798	0.768877	0.468183	0.097498	0.1
50%	25.50000	6.768000	0.040795	1.268555	1.243645	0.868215	0.568085	0.196610	0.2
75%	37.75000	7.196750	0.046277	1.333112	1.298787	0.909165	0.631748	0.325240	0.3
max	50.00000	7.587000	0.108950	1.690420	1.402230	1.025250	0.669730	0.522080	0.5

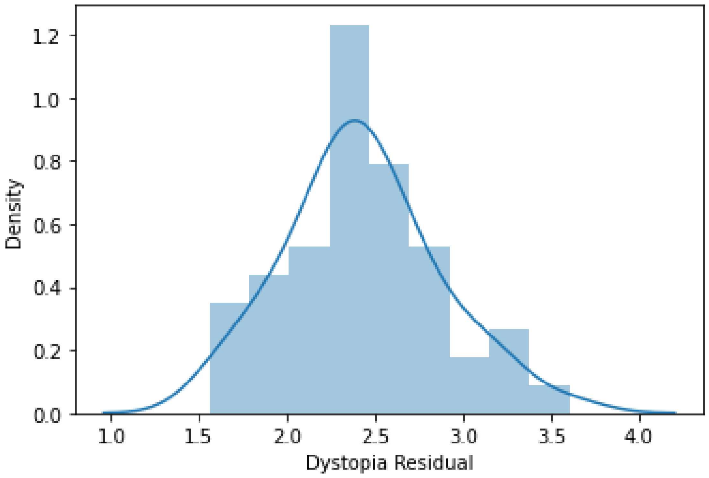
In [81]:
```
sns.pairplot(a)
```



In [82]:
```
sns.distplot(a['Dystopia Residual'])
```

```
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)
```

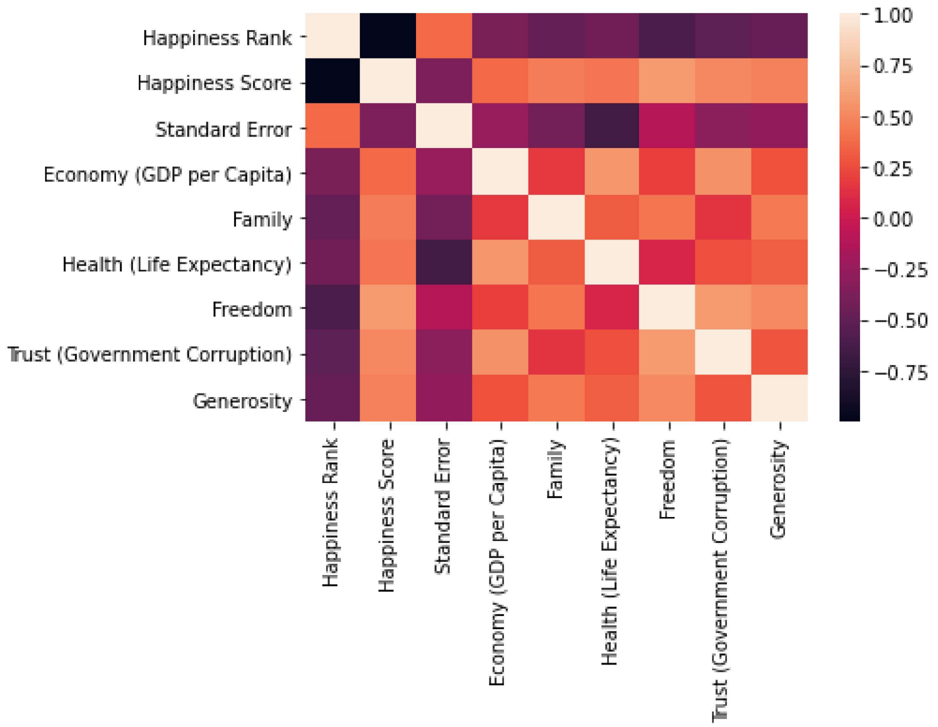
Out[82]:
```
<AxesSubplot:xlabel='Dystopia Residual', ylabel='Density'>
```



In [83]:
```
x1=a[['Happiness Rank', 'Happiness Score',
       'Standard Error', 'Economy (GDP per Capita)', 'Family',
       'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)',
       'Generosity']]
```

In [84]:
```
sns.heatmap(x1.corr())
```

Out[84]:
```
<AxesSubplot:>
```



In [86]:
```
x=a[['Happiness Rank', 'Happiness Score',
       'Standard Error', 'Economy (GDP per Capita)', 'Family',
       'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)',
       'Generosity']]
y=a['Dystopia Residual']
```

In [87]:
```
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 [88]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[88]:
```
LinearRegression()
```

In [89]:
```
print(lr.intercept_)
```

```
0.004859264853705358
```

In [90]:

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

Out[90]:

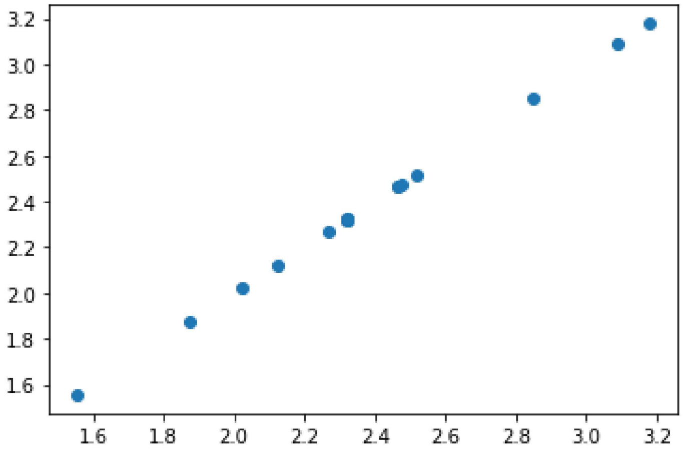
	Co-efficient
Happiness Rank	-0.000021
Happiness Score	0.999387
Standard Error	-0.002848
Economy (GDP per Capita)	-0.999397
Family	-1.000531
Health (Life Expectancy)	-1.000615
Freedom	-0.998902
Trust (Government Corruption)	-1.000772
Generosity	-1.000018

In [91]:

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

Out[91]:

```
<matplotlib.collections.PathCollection at 0x22eb35b1220>
```



In [92]:

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

```
0.999999523602235
```

In [93]:

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

In [94]:

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

Out[94]:

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

In [95]:

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

Out[95]:

```
0.37967082440574806
```

In [96]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[96]:
```
Lasso(alpha=10)
```

In [97]:
```
la.score(x_test,y_test)
```

Out[97]:
```
-0.015015955172610562
```

In [98]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[98]:
```
ElasticNet()
```

In [99]:
```
print(en.coef_)
```

```
[-0.00351865  0.          0.         -0.         -0.         -0.
 -0.         -0.         -0.        ]
```

In [100]:
```
print(en.intercept_)
```

```
2.530602103288059
```

In [101]:
```
print(en.predict(x_test))
```

```
[2.50597154 2.45319176 2.39689332 2.48837828 2.38633737 2.51652749
 2.52708345 2.40393063 2.41800524 2.3546695  2.41096793 2.43207984
 2.4426358  2.49189693 2.39337467]
```

In [102]:
```
en.score(x_test,y_test)
```

Out[102]:
```
0.11248425296623765
```

In [103]:
```
from sklearn import metrics
```

In [104]:
```
print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error 0.00022193162481184045
```

In [105]:
```
print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error 8.079143874781493e-08
```

In [106]:
```
print(" Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction)))
```

```
 Root Mean Squared Error 0.0002842383484820705
```