In [91]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [92]:
```
data=pd.read_csv(r"C:\Users\user\Downloads\2015 - 2015.csv")
data
```

Out[92]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Fr
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0
...	...	...	...	...	...	...	...	...	
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0

158 rows × 12 columns

In [93]:
```
df=data.head(100)
df
```

Out[93]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563
...	...	...	...	...	...	...	...	...
95	Bosnia and Herzegovina	Central and Eastern Europe	96	4.949	0.06913	0.83223	0.91916	0.79081
96	Lesotho	Sub-Saharan Africa	97	4.898	0.09438	0.37545	1.04103	0.07612
97	Dominican Republic	Latin America and Caribbean	98	4.885	0.07446	0.89537	1.17202	0.66825
98	Laos	Southeastern Asia	99	4.876	0.06698	0.59066	0.73803	0.54909
99	Mongolia	Eastern Asia	100	4.874	0.03313	0.82819	1.30060	0.60268

100 rows × 12 columns

In [94]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Country                        100 non-null    object 
 1   Region                         100 non-null    object 
 2   Happiness Rank                 100 non-null    int64  
 3   Happiness Score                100 non-null    float64
 4   Standard Error                 100 non-null    float64
 5   Economy (GDP per Capita)       100 non-null    float64
 6   Family                         100 non-null    float64
 7   Health (Life Expectancy)       100 non-null    float64
 8   Freedom                        100 non-null    float64
 9   Trust (Government Corruption)  100 non-null    float64
 10  Generosity                     100 non-null    float64
 11  Dystopia Residual              100 non-null    float64
dtypes: float64(9), int64(1), object(2)
memory usage: 9.5+ KB
```

In [95]:
```
df.describe()
```

Out[95]:

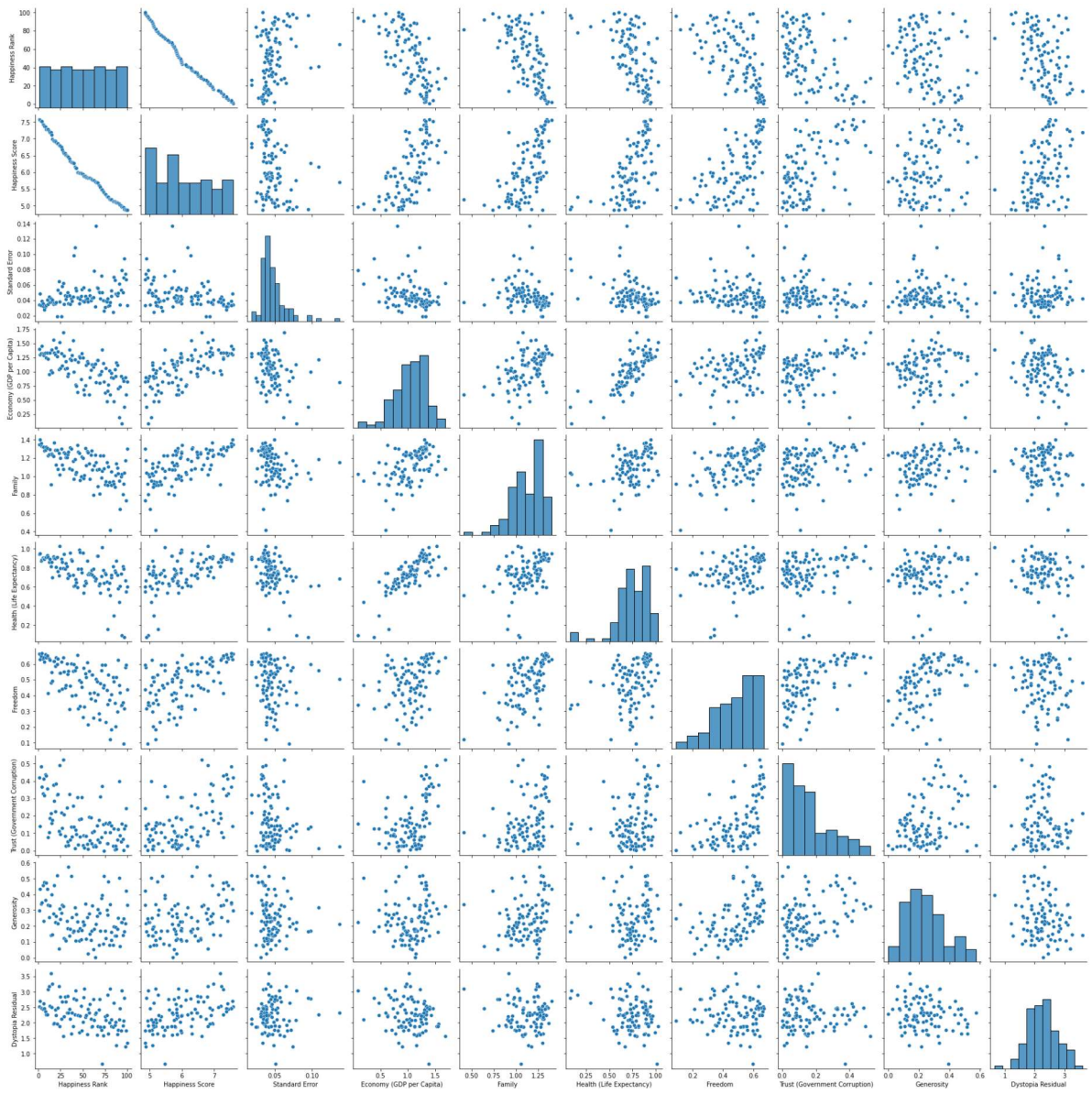
	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(G C
count	100.000000	100.00000	100.000000	100.000000	100.000000	100.000000	100.000000	1
mean	50.490000	6.06081	0.047206	1.045210	1.119594	0.747963	0.480267	
std	29.000347	0.79900	0.017788	0.299610	0.175886	0.175114	0.135930	
min	1.000000	4.87400	0.018480	0.083080	0.414110	0.076120	0.092450	
25%	25.750000	5.35300	0.037135	0.875007	1.007810	0.666432	0.401975	
50%	50.500000	5.91900	0.042650	1.073035	1.140595	0.755560	0.500285	
75%	75.250000	6.75900	0.052268	1.272500	1.258182	0.885710	0.596122	
max	100.000000	7.58700	0.136930	1.690420	1.402230	1.025250	0.669730	

In [96]:
```
df.columns
```

Out[96]:
```
Index(['Country', 'Region', 'Happiness Rank', 'Happiness Score',
       'Standard Error', 'Economy (GDP per Capita)', 'Family',
       'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruptio
n)',
       'Generosity', 'Dystopia Residual'],
      dtype='object')
```

In [97]:
```
sns.pairplot(df)
```

Out[97]:
```
<seaborn.axisgrid.PairGrid at 0x1ed7255e430>
```



In [98]:
```
da=df[[ 'Happiness Rank', 'Happiness Score',
       'Standard Error', 'Economy (GDP per Capita)', 'Family',
       'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)',
       'Generosity', 'Dystopia Residual']]
da
```

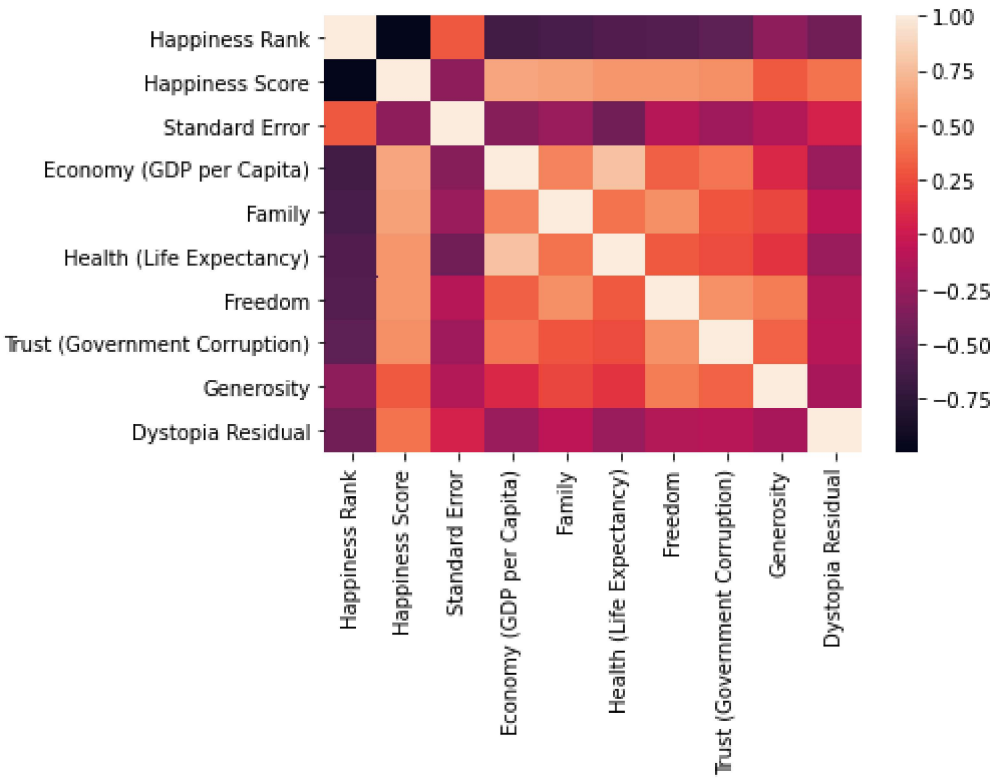
Out[98]:

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	C
0	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	
1	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	
2	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	
3	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	
4	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	
...	...	...	...	...	...	...	...	...	
95	96	4.949	0.06913	0.83223	0.91916	0.79081	0.09245	0.00227	
96	97	4.898	0.09438	0.37545	1.04103	0.07612	0.31767	0.12504	
97	98	4.885	0.07446	0.89537	1.17202	0.66825	0.57672	0.14234	
98	99	4.876	0.06698	0.59066	0.73803	0.54909	0.59591	0.24249	
99	100	4.874	0.03313	0.82819	1.30060	0.60268	0.43626	0.02666	

100 rows × 10 columns

In [99]:
```
sns.heatmap(da.corr())
```

Out[99]: <AxesSubplot:>



In [100]:
```
x=df[['Happiness Rank', 'Happiness Score',
       'Standard Error', 'Economy (GDP per Capita)', 'Family',
       'Health (Life Expectancy)', 'Trust (Government Corruption)',
       'Generosity', 'Dystopia Residual']]
y=df['Freedom']
```

In [101]:
```
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 [102]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[102]: LinearRegression()

In [103]:
```
print(lr.intercept_)
```

```
-0.0010925570149081243
```

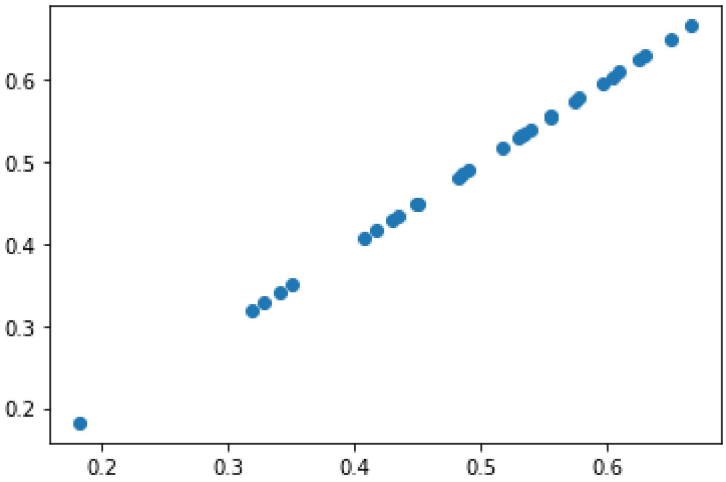
In [104]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[104]:

	Co-efficient
Happiness Rank	0.000005
Happiness Score	1.001063
Standard Error	-0.001779
Economy (GDP per Capita)	-1.000959
Family	-1.001119
Health (Life Expectancy)	-1.000799
Trust (Government Corruption)	-1.000809
Generosity	-1.001252
Dystopia Residual	-1.000977

In [105]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[105]:
```
<matplotlib.collections.PathCollection at 0x1ed77ff5670>
```



In [106]:
```
print(lr.score(x_test,y_test))
```

```
0.9999917973958037
```

In [107]:
```
print(lr.score(x_train,y_train))
```

```
0.9999972497920053
```

In [108]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [109]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[109]:
```
Ridge(alpha=10)
```

In [110]:
```
rr.score(x_test,y_test)
```

Out[110]:
```
0.42844003441559564
```

In [111]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[111]:
```
Lasso(alpha=10)
```

In [112]:
```
la.score(x_test,y_test)
```

Out[112]:
```
-0.035531476597402634
```

In [ ]:
```

```