In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:

```
df=pd.read_csv(r"C:\Users\user\Downloads\2_2015.csv")
df.fillna(0,inplace=True)
df
```

Out[3]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Fre
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.6
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.6
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.6
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.6
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.6
...	...	...	...	...	...	...	...	...	
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.5
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.4
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.1
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.1
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.3

158 rows × 12 columns

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

Out[4]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedo
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.665
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.628
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.649
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.669
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.632

In [5]:
```
df.info()
```

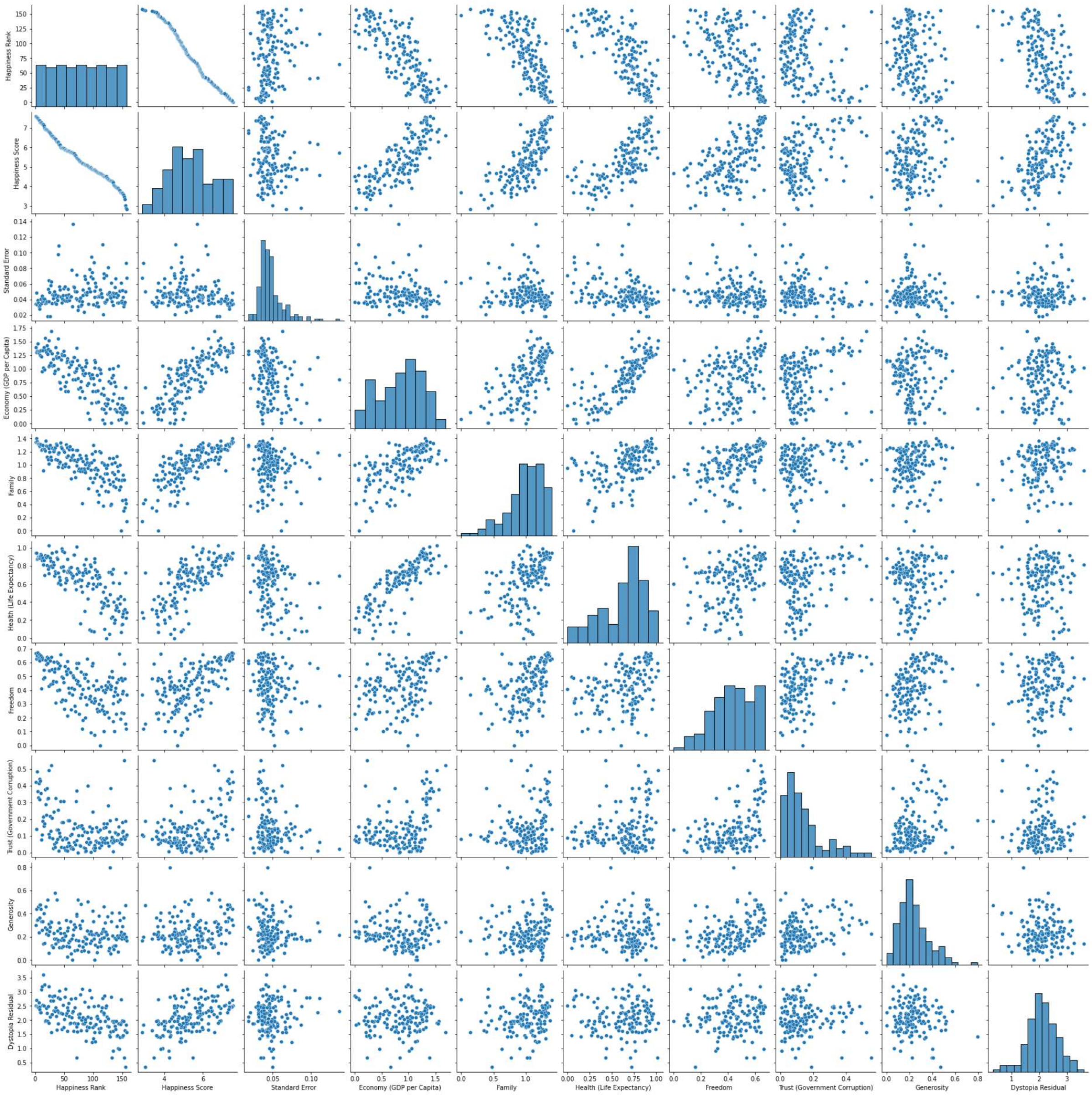
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Country                        158 non-null    object 
 1   Region                         158 non-null    object 
 2   Happiness Rank                 158 non-null    int64  
 3   Happiness Score                158 non-null    float64
 4   Standard Error                 158 non-null    float64
 5   Economy (GDP per Capita)       158 non-null    float64
 6   Family                         158 non-null    float64
 7   Health (Life Expectancy)       158 non-null    float64
 8   Freedom                        158 non-null    float64
 9   Trust (Government Corruption)  158 non-null    float64
 10  Generosity                     158 non-null    float64
 11  Dystopia Residual              158 non-null    float64
dtypes: float64(9), int64(1), object(2)
memory usage: 14.9+ KB
```

In [6]:
```
df=pd.read_csv("2_2015.csv")
```

In [7]:
```
import seaborn as sns
```

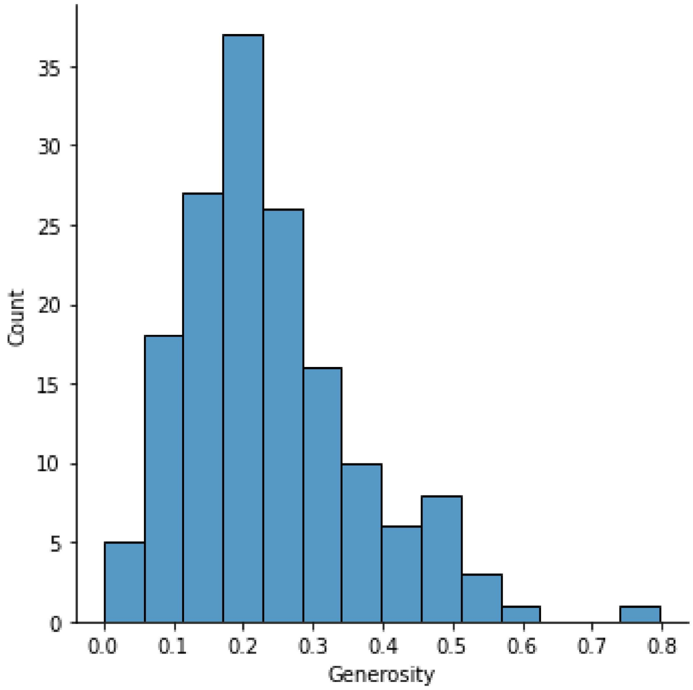
In [8]:
```
sns.pairplot(df)
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x1e0a1080c10>
```



```
In [9]: sns.displot(df['Generosity'])
```

```
Out[9]: <seaborn.axisgrid.FacetGrid at 0x1e0a6d47dc0>
```

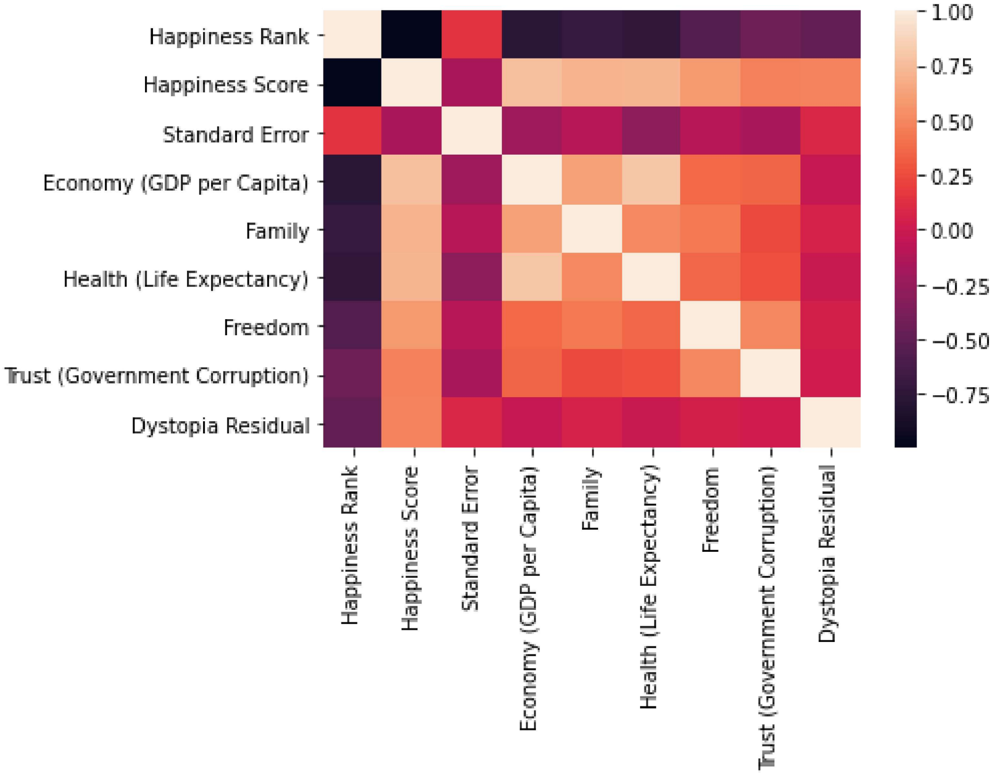


```
In [10]: df1=df.drop(['Generosity'],axis=1)
         df1
         df1=df1.drop(df1.index[153:])
         df1.isna().sum()
```

```
Out[10]: Country                          0
         Region                           0
         Happiness Rank                   0
         Happiness Score                  0
         Standard Error                   0
         Economy (GDP per Capita)         0
         Family                           0
         Health (Life Expectancy)         0
         Freedom                          0
         Trust (Government Corruption)    0
         Dystopia Residual                0
         dtype: int64
```

In [11]:
```
sns.heatmap(df1.corr())
```

Out[11]:
```
<AxesSubplot:>
```



In [12]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [13]:
```
df1.isna().sum()
```

Out[13]:
```
Country                          0
Region                           0
Happiness Rank                   0
Happiness Score                  0
Standard Error                   0
Economy (GDP per Capita)         0
Family                           0
Health (Life Expectancy)         0
Freedom                          0
Trust (Government Corruption)    0
Dystopia Residual                0
dtype: int64
```

In [23]:

```
y=df1['Happiness Rank']
x=df1.drop(['Country','Happiness Rank','Region'],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

```
     Happiness Score  Standard Error  Economy (GDP per Capita)   Family  \
138            3.989         0.06682                   0.67866  0.66290   
22             6.810         0.06476                   1.04424  1.25596   
70             5.477         0.07197                   1.00761  0.98521   
101            4.857         0.05062                   1.15406  0.92933   
6              7.378         0.02799                   1.32944  1.28017   
..               ...             ...                       ...      ...   
48             5.960         0.05412                   1.32376  1.21624   
49             5.948         0.03914                   1.25114  1.19777   
69             5.548         0.04175                   0.95847  1.22668   
37             6.298         0.03868                   1.29098  1.07617   
87             5.102         0.04802                   1.15991  1.13935   

     Health (Life Expectancy)  Freedom  Trust (Government Corruption)  \
138                   0.31051  0.41466                        0.11686   
22                    0.72052  0.42908                        0.11069   
70                    0.70950  0.56066                        0.07521   
101                   0.88213  0.07699                        0.01397   
6                     0.89284  0.61576                        0.31814   
..                        ...      ...                            ...   
48                    0.74716  0.45492                        0.30600   
49                    0.95446  0.26236                        0.02901   
69                    0.53886  0.47610                        0.30844   
37                    0.87530  0.39740                        0.08129   
87                    0.87519  0.51469                        0.01078   

     Dystopia Residual  
138            1.68135  
22             3.19131  
70             1.76145  
101            1.80101  
6              2.46570  
..                 ...  
48             1.73797  
49             2.02518  
69             1.86984  
37             2.32323  
87             1.26462  

[107 rows x 8 columns]
```

In [24]:

```
model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

Out[24]:

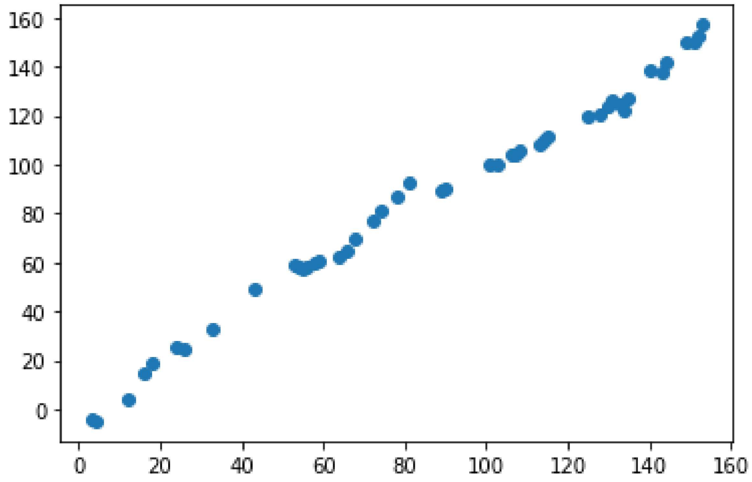
```
297.1524499401098
```

```
In [25]: model.coef_
```

```
Out[25]: array([-26.92401634, -29.70457572, -12.69173816, -15.35442221,
                -17.13611956, -18.78902373,  -2.93847116, -12.41537647])
```

```
In [26]: 
         prediction=model.predict(x_test)
         plt.scatter(y_test,prediction)
```

```
Out[26]: <matplotlib.collections.PathCollection at 0x1e0a8dd45b0>
```



```
In [27]: 
         model.score(x_test,y_test)
```

```
Out[27]: 0.9876499329152798
```

```
In [ ]: 
```