In [5]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import linear_model
```

In [6]:
```python
df=pd.read_csv('D:/shree/happiness_score.csv')
```

In [7]:
```python
df.head()
#Bottom 5 rows
```

Out[7]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176

In [8]:
```python
df.columns
# all columns title
```

Out[8]:
```
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 [41]:
```python
df_cols=['Country', 'Region', 'Happiness_Rank', 'Happiness_Score',
         'Standard_Error', 'Economy(GDP_per_Capita)', 'Family',
         'Health(Life_Expectancy)', 'Freedom', 'Trust(Government_Corruption)',
         'Generosity', 'Dystopia_Residual']

# renaming headers of columns
```

In [42]:
```python
df.columns=df_cols
# replacing with new one
```

In [43]:
```python
df.columns
```

Out[43]:
```
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 [44]:
```python
df.isnull()
# checking for null values (boolean)
```

Out[44]:

	Country	Region	Happiness_Rank	Happiness_Score	Standard_Error	Economy(GDP_per_Capita)	Family	Health(Life_Expectancy)	Freedom	Trust(Government_Corruption)	Generosity	Dystopia_Re
0	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	
153	False	False	False	False	False	False	False	False	False	False	False	
154	False	False	False	False	False	False	False	False	False	False	False	
155	False	False	False	False	False	False	False	False	False	False	False	
156	False	False	False	False	False	False	False	False	False	False	False	
157	False	False	False	False	False	False	False	False	False	False	False	

158 rows × 12 columns

In [45]:
```python
df.isnull().sum()
# counting if any null values
```

Out[45]:
```
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
Generosity                      0
Dystopia_Residual               0
dtype: int64
```

In [46]:
```python
df.shape
# rows and columns
```

Out[46]: (158, 12)

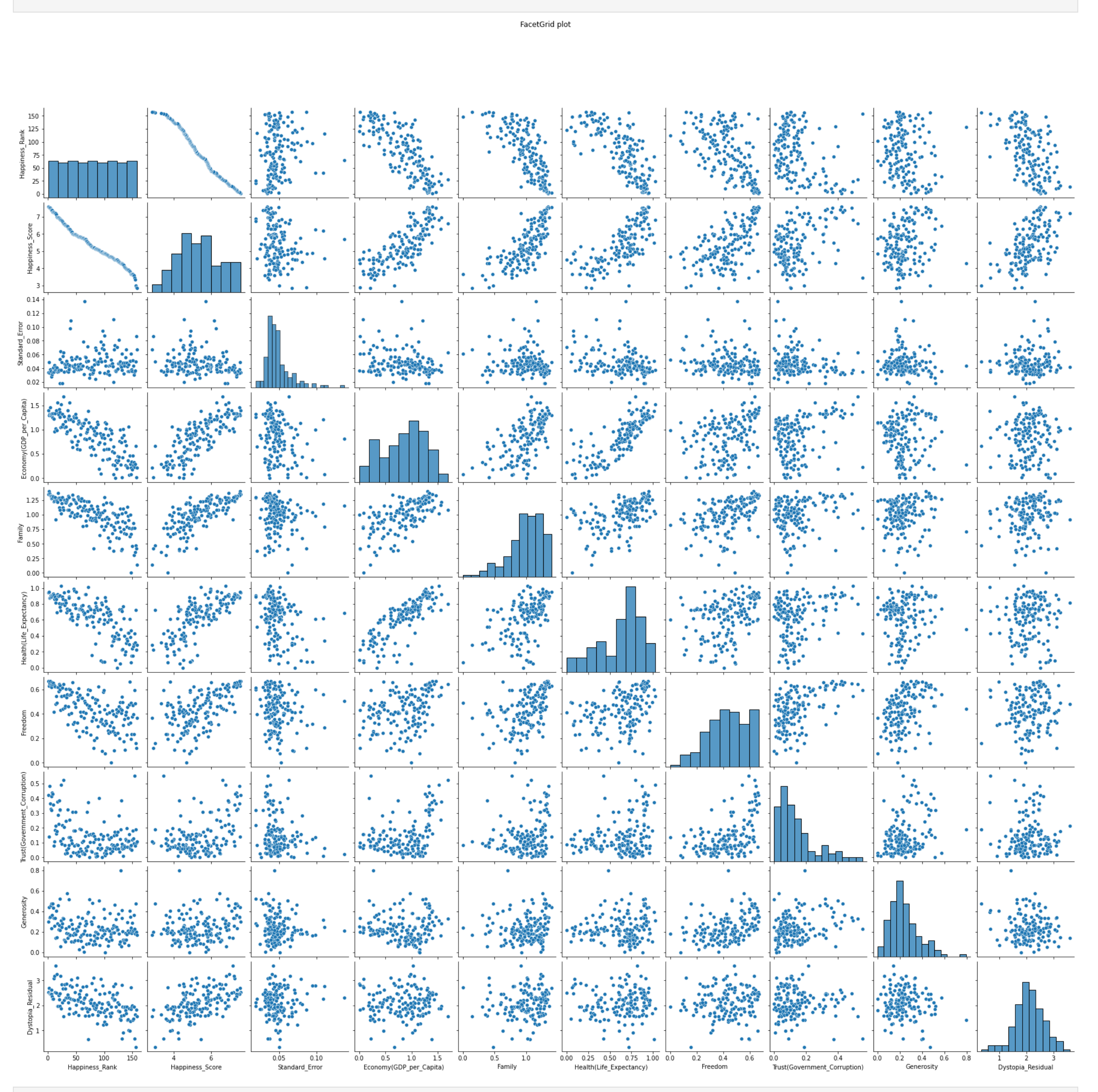
In [47]:
```python
df.dtypes
# columns types
```

Out[47]:
```
Country                          object
Region                           object
Happiness_Rank                    int64
Happiness_Score                 float64
Standard_Error                  float64
Economy(GDP_per_Capita)         float64
Family                          float64
Health(Life_Expectancy)         float64
Freedom                         float64
Trust(Government_Corruption)    float64
Generosity                      float64
Dystopia_Residual               float64
dtype: object
```

In [82]:
```python
# plotting

plot = sns.pairplot(df)
plot.fig.suptitle('FacetGrid plot', fontsize = 12)
plot.fig.subplots_adjust(top= 0.9)

# This shows the relationship between 2 attributes.
```



In [ ]:
```python
# Finding correlation of each attributes with happiness_score
```

In [50]:
```python
economy_happiness = ['Happiness_Score','Economy(GDP_per_Capita)']
economy_corr = df[economy_happiness]
economy_corr.corr()
```

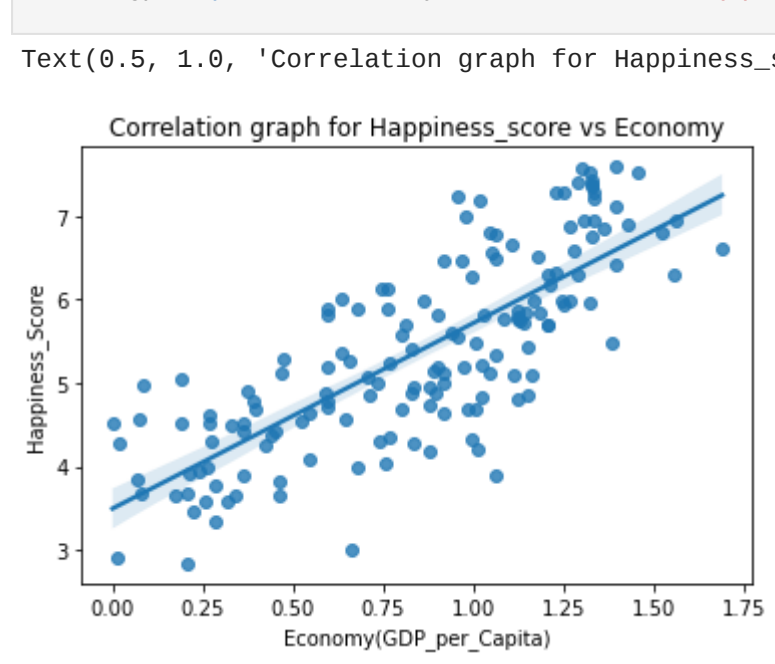
Out[50]:

	Happiness_Score	Economy(GDP_per_Capita)
Happiness_Score	1.000000	0.780966
Economy(GDP_per_Capita)	0.780966	1.000000

In [52]:
```python
#correlation graph

sns.regplot(data = economy_corr, x = 'Economy(GDP_per_Capita)', y = 'Happiness_Score').set_title("Correlation graph for Happiness_score vs Economy")
```

Out[52]: Text(0.5, 1.0, 'Correlation graph for Happiness_score vs Economy')



In [77]:
```python
#Creating R and R Squared for Happiness Score and Economy

R = np.array(economy_corr.corr())
R2 = np.array(economy_corr.corr()**2)

print('The data shows we have an R value of: ' + str(R[1]) + ' and and R2 value of ' + str(R2[1]))
```

The data shows we have an R value of: [0.78096553 1.   ] and and R2 value of [0.60990715 1.   ]

In [ ]:
```python
#############
```

In [54]:
```python
family_happiness = ['Happiness_Score','Family']
family_corr = df[family_happiness]
family_corr.corr()
```

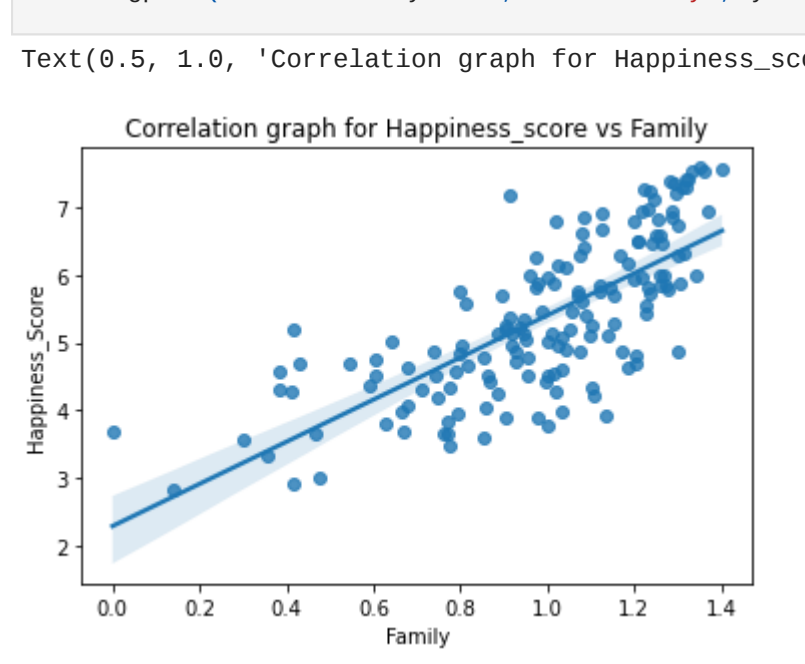
Out[54]:

	Happiness_Score	Family
Happiness_Score	1.000000	0.740605
Family	0.740605	1.000000

In [57]:
```python
#correlation graph

sns.regplot(data = family_corr, x = 'Family', y = 'Happiness_Score').set_title("Correlation graph for Happiness_score vs Family")
```

Out[57]: Text(0.5, 1.0, 'Correlation graph for Happiness_score vs Family')



In [76]:
```python
#Creating R and R Squared for Happiness_Score and Family

R = np.array(family_corr.corr())
R2 = np.array(family_corr.corr()**2)

print('The data shows we have an R value of: ' + str(R[1]) + ' and and R2 value of ' + str(R2[1]))
```

The data shows we have an R value of: [0.7406052 1.   ] and and R2 value of [0.54849606 1.   ]

In [ ]:
```python
##########
```

In [59]:
```python
health_happiness = ['Happiness_Score','Health(Life_Expectancy)']
health_corr = df[health_happiness]
health_corr.corr()
```

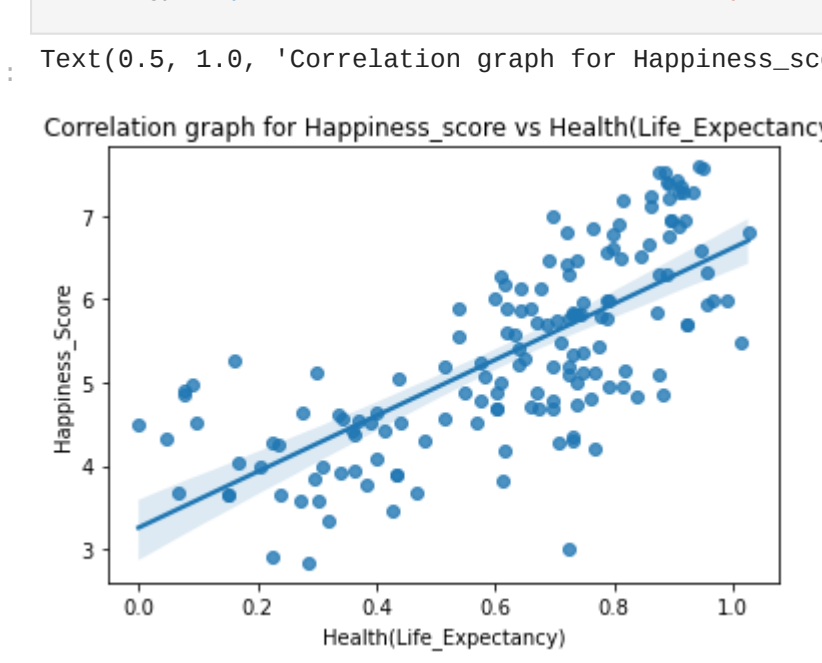
Out[59]:

	Happiness_Score	Health(Life_Expectancy)
Happiness_Score	1.0000	0.7242
Health(Life_Expectancy)	0.7242	1.0000

In [63]:
```python
#correlation graph

sns.regplot(data = health_corr, x = 'Health(Life_Expectancy)', y = 'Happiness_Score').set_title("Correlation graph for Happiness_score vs Health(Life_Expectan
```

Out[63]: Text(0.5, 1.0, 'Correlation graph for Happiness_score vs Health(Life_Expectancy)')



In [75]:
```python
#Creating R and R Squared for Happiness_Score and Health

R = np.array(health_corr.corr())
R2 = np.array(health_corr.corr()**2)

print('The data shows we have an R value of: ' + str(R[1]) + ' and and R2 value of ' + str(R2[1]))
```

The data shows we have an R value of: [0.7241996 1.   ] and and R2 value of [0.52446505 1.   ]

In [ ]:
```python
###############
```

In [65]:
```python
freedom_happiness = ['Happiness_Score','Freedom']
freedom_corr = df[freedom_happiness]
freedom_corr.corr()
```

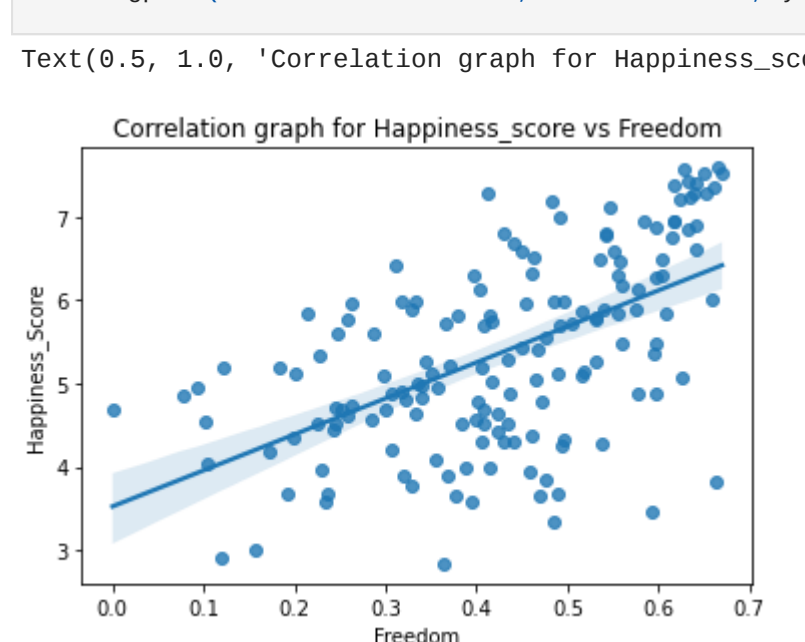
Out[65]:

	Happiness_Score	Freedom
Happiness_Score	1.000000	0.568211
Freedom	0.568211	1.000000

In [66]:
```python
#correlation graph

sns.regplot(data = freedom_corr, x = 'Freedom', y = 'Happiness_Score').set_title("Correlation graph for Happiness_score vs Freedom")
```

Out[66]: Text(0.5, 1.0, 'Correlation graph for Happiness_score vs Freedom')



In [74]:
```python
#Creating R and R Squared for Happiness_Score and freedom

R = np.array(freedom_corr.corr())
R2 = np.array(freedom_corr.corr()**2)

print('The data shows we have an R value of: ' + str(R[1]) + ' and and R2 value of ' + str(R2[1]))
```

The data shows we have an R value of: [0.5682109 1.   ] and and R2 value of [0.32286363 1.   ]

In [ ]:
```python
################
```

In [67]:
```python
trust_happiness = ['Happiness_Score','Trust(Government_Corruption)']
trust_corr = df[trust_happiness]
trust_corr.corr()
```

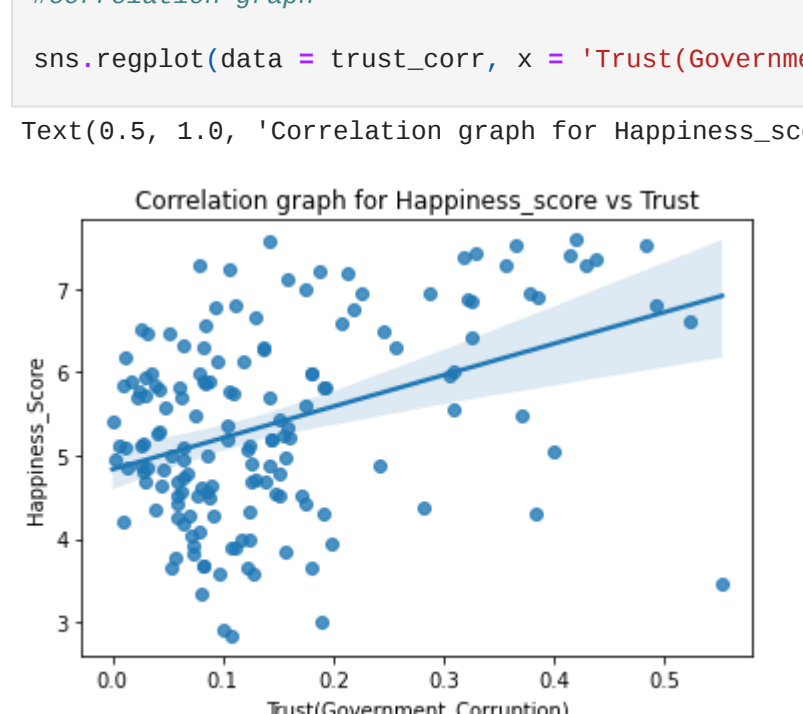
Out[67]:

	Happiness_Score	Trust(Government_Corruption)
Happiness_Score	1.000000	0.395199
Trust(Government_Corruption)	0.395199	1.000000

In [68]:
```python
#correlation graph

sns.regplot(data = trust_corr, x = 'Trust(Government_Corruption)', y = 'Happiness_Score').set_title("Correlation graph for Happiness_score vs Trust")
```

Out[68]: Text(0.5, 1.0, 'Correlation graph for Happiness_score vs Trust')



In [73]:
```python
#Creating R and R Squared for Happiness_Score and trust

R = np.array(trust_corr.corr())
R2 = np.array(trust_corr.corr()**2)

print('The data shows we have an R value of: ' + str(R[1]) + ' and and R2 value of ' + str(R2[1]))
```

The data shows we have an R value of: [0.39519858 1.   ] and and R2 value of [0.15618192 1.   ]

In [ ]:
```python
###########
```

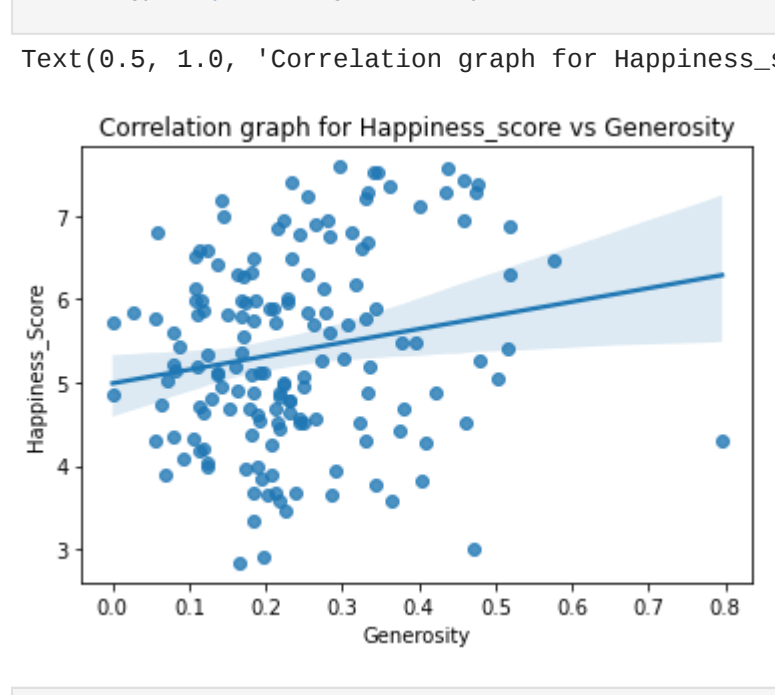
In [69]:
```python
generosity_happiness = ['Happiness_Score','Generosity']
generosity_corr = df[generosity_happiness]
generosity_corr.corr()
```

Out[69]:

	Happiness_Score	Generosity
Happiness_Score	1.000000	0.180319
Generosity	0.180319	1.000000

In [70]:
```python
sns.regplot(data = generosity_corr, x = 'Generosity', y = 'Happiness_Score').set_title("Correlation graph for Happiness_score vs Generosity")
```

Out[70]: Text(0.5, 1.0, 'Correlation graph for Happiness_score vs Generosity')



In [72]:
```python
#Creating R and R Squared for Happiness_Score and Generosity

R = np.array(generosity_corr.corr())
R2 = np.array(generosity_corr.corr()**2)

print('The data shows we have an R value of: ' + str(R[1]) + ' and and R2 value of ' + str(R2[1]))
```

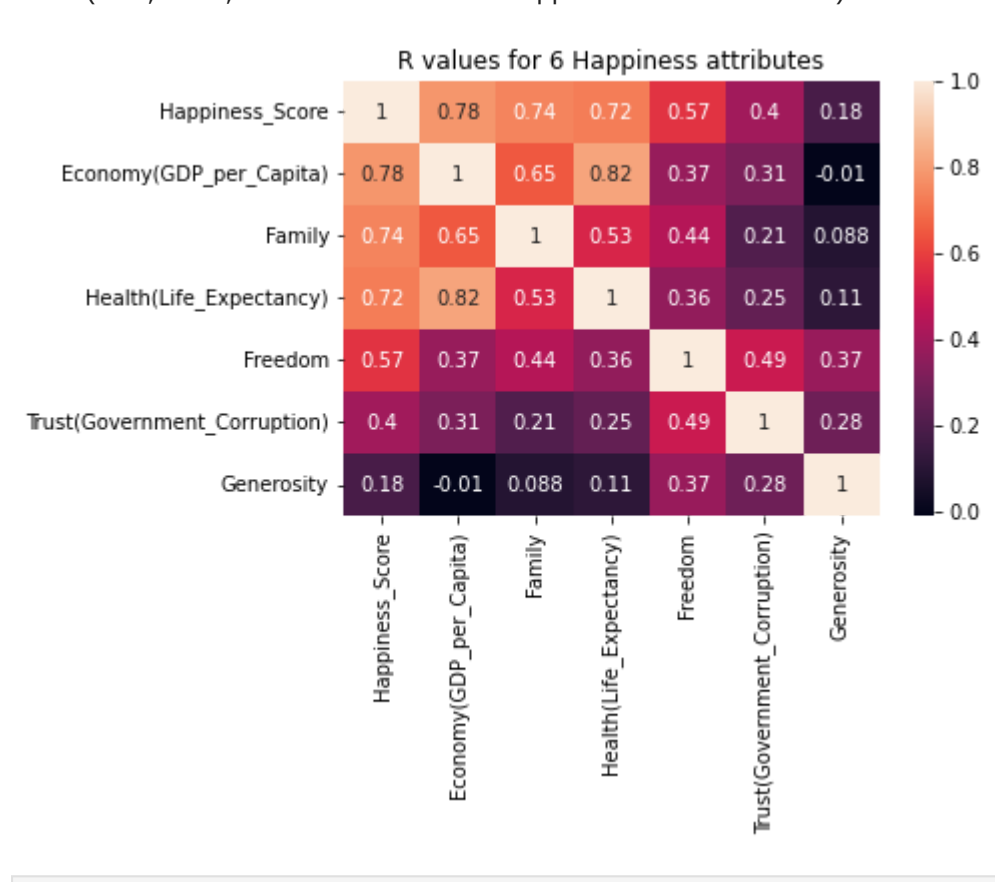
The data shows we have an R value of: [0.18031853 1.   ] and and R2 value of [0.03251477 1.   ]

In [ ]:
```python
#########
```

In [79]:
```python
# Creating a correlation matrix for R values

all_data = ['Happiness_Score', 'Economy(GDP_per_Capita)', 'Family','Health(Life_Expectancy)', 'Freedom', 'Trust(Government_Corruption)', 'Generosity']
all_data_dataframe = df[all_data]
sns.heatmap(all_data_dataframe.corr(), annot=True).set_title("R values for 6 Happiness attributes")
```

Out[79]: Text(0.5, 1.0, 'R values for 6 Happiness attributes')

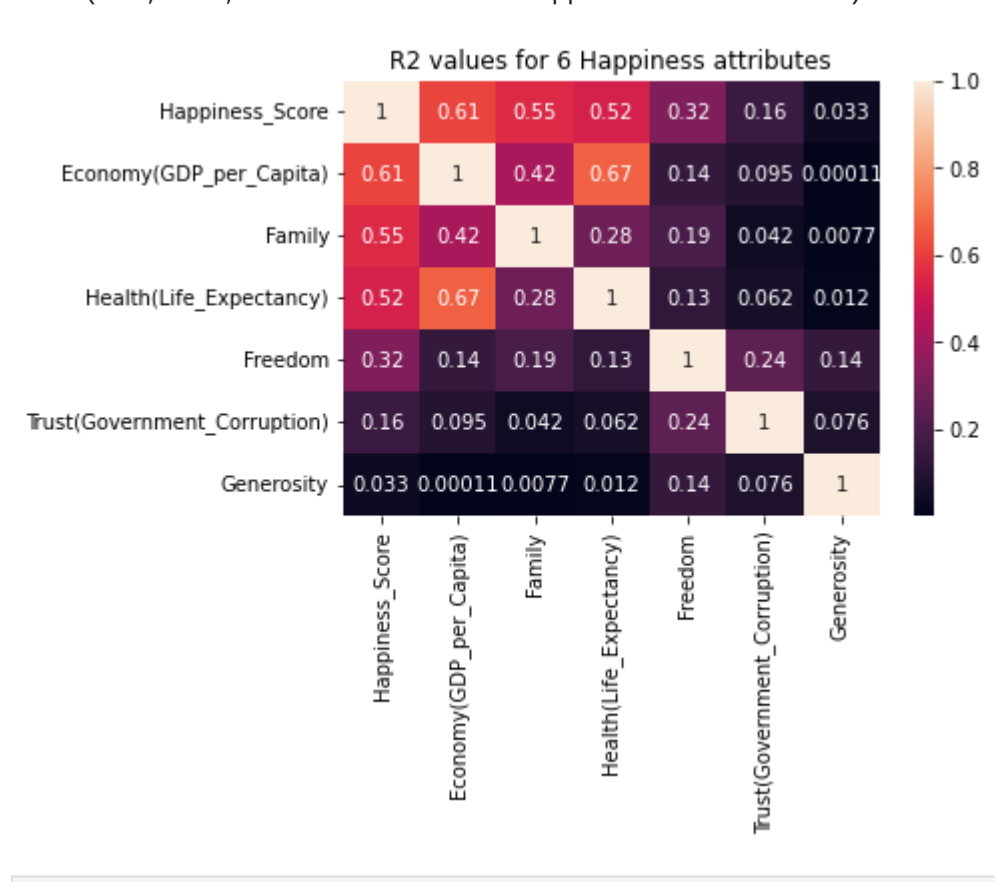


In [ ]:
```python
###########
```

In [80]:
```python
# Creating a correlation matrix for R2 values

all_data = ['Happiness_Score', 'Economy(GDP_per_Capita)', 'Family','Health(Life_Expectancy)', 'Freedom', 'Trust(Government_Corruption)', 'Generosity']
all_data_dataframe = df[all_data]
sns.heatmap(all_data_dataframe.corr()**2, annot=True).set_title("R2 values for 6 Happiness attributes")
```

Out[80]: Text(0.5, 1.0, 'R2 values for 6 Happiness attributes')



In [ ]:
```python
###############
```

In [10]:
```python
df.dtypes
```

Out[10]:
```
Country                           object
Region                            object
Happiness Rank                     int64
Happiness Score                  float64
Standard Error                   float64
Economy (GDP per Capita)         float64
Family                           float64
Health (Life Expectancy)         float64
Freedom                          float64
Trust (Government Corruption)    float64
Generosity                       float64
Dystopia Residual                float64
dtype: object
```

In [11]:
```python
df.head(10)
```

Out[11]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176
5	Finland	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	0.64169	0.41372	0.23351	2.61955
6	Netherlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	0.61576	0.31814	0.47610	2.46570
7	Sweden	Western Europe	8	7.364	0.03157	1.33171	1.28907	0.91087	0.65980	0.43844	0.36262	2.37119
8	New Zealand	Australia and New Zealand	9	7.286	0.03371	1.25018	1.31967	0.90837	0.63938	0.42922	0.47501	2.26425
9	Australia	Australia and New Zealand	10	7.284	0.04083	1.33358	1.30923	0.93156	0.65124	0.35637	0.43562	2.26646

In [17]:
```python
df.columns
```

Out[17]:
```
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 [20]:
```python
df_cols=['Country', 'Region', 'HappinessRank', 'HappinessScore',
         'StandardError', 'EconomyGDPperCapita', 'Family',
         'HealthLifeExpectancy', 'Freedom', 'TrustGovernmentCorruption',
         'Generosity', 'DystopiaResidual']
```

In [21]:
```python
df.columns=df_cols
```

In [24]:
```python
df.columns
```

Out[24]:
```
Index(['Country', 'Region', 'HappinessRank', 'HappinessScore', 'StandardError',
       'EconomyGDPperCapita', 'Family', 'HealthLifeExpectancy', 'Freedom',
       'TrustGovernmentCorruption', 'Generosity', 'DystopiaResidual'],
      dtype='object')
```

In [25]:
```python
reg=linear_model.LinearRegression()
reg.fit(df[['EconomyGDPperCapita', 'Family','HealthLifeExpectancy', 'Freedom', 'TrustGovernmentCorruption','Generosity']],df.HappinessScore)
```

Out[25]: LinearRegression()

In [26]:
```python
reg.coef_
```

Out[26]:
```
array([0.8606572 , 1.40889159, 0.97530903, 1.33343293, 0.78453814,
       0.38893285])
```

In [27]:
```python
reg.intercept_
```

Out[27]: 1.8601849449130587

In [ ]:
```python
#Summary:

#Top attributes that contribute to Happiness:

# 1 Economy (R=0.78096553, R2=0.60990715)
# 2 Family (R=0.7406052, R2=0.54849606)
# 3 Health (R=0.7241996, R2=0.52446505)
# 4 Freedom (R=0.5682109, R2=0.32286363)
# 5 Trust (R=0.39519858, R2=0.15618192)
# 6 Generosity (R=0.18031853, R2=0.03251477)

# From the data we can see that the three most important attributes that impact happiness are economy, family and health.
```