WORLD HAPPINESS

WORLD HAPPINESS

The World Happiness Report is an annual publication that ranks countries based on various factors related to happiness and well-being. The report is prepared by the United Nations Sustainable Development Solutions Network in collaboration with experts from around the world.

DESCRIPTION:

The dataset we have likely contains information related to the variables used in calculating the happiness scores and rankings. Here is some information about the common variables included in the World Happiness Report:

1. Country: The name of the country included in the ranking.
2. Region: The region or geographic area to which the country belongs.
3. Happiness Rank: The position of the country in the overall happiness ranking.

4. Happiness Score: A numerical score representing the level of happiness and well-being in the country. This score is typically based on survey responses related to various factors like GDP per capita, social support, life expectancy, freedom, generosity, and

5. Standard Error: The standard error associated with the happiness score, which indicates the precision of the estimate.
6. Economy (GDP per Capita): The country's GDP per capita, representing the economic well-being of the nation.
7. Family: This variable represents the strength of social support systems within the country, including family support.
8. Health (Life Expectancy): The life expectancy of the population, which is an indicator of overall health and well-being.
9. Freedom: The degree of freedom individuals have in making life choices, including political freedom.
10. Trust (Government Corruption): This variable measures the perceptions of corruption in the government and public institutions.

11. Generosity: The level of generosity and willingness to help others in the country.
12. Dystopia Residual: A hypothetical value representing the level of unhappiness or dystopia, used as a benchmark for calculating the happiness scores.

WORKING IN PANDAS (DATA PREPROCESSING AND CLEANING)

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#IMPORTING OF OUR DATASET INTO JUPYTER USING PANDAS
dataset=pd.read_csv("./World_Happiness_2015.csv")
```

In [3]:
```
dataset
```

Out[3]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000
4	4	4	Canada	North America	5	7.427	0.03553	1.326290
...	...	...	...	...	...	...	...	...
153	153	153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.222080
154	154	154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.286650
155	155	155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.663200
156	156	156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.015300
157	157	157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.208680

158 rows × 14 columns

In [4]:
```
dataset.shape
```

Out[4]:
```
(158, 14)
```

In [5]:
```
#this command will show the total elements of our dataset by multiplying ro
dataset.size
```

Out[5]:
```
2212
```

In [6]:
```
dataset.head(6)
```

Out[6]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510	1
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995	1
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480	1
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000	1
4	4	4	Canada	North America	5	7.427	0.03553	1.326290	1
5	5	5	Finland	Western Europe	6	7.406	0.03176	1.290250	1

In [7]:
```
dataset.tail(6)
```

Out[7]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)
152	152	152	Afghanistan	Southern Asia	153	3.575	0.03084	0.31982
153	153	153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208
154	154	154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665
155	155	155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320
156	156	156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530
157	157	157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868

In [8]:
```
#This command shows the columns of our dataset
dataset.columns
```

Out[8]:
```
Index(['Unnamed: 0.1', 'Unnamed: 0', 'Country', 'Region', 'Happiness Ran
k',
       'Happiness Score', 'Standard Error', 'Economy (GDP per Capita)',
       'Family', 'Health (Life Expectancy)', 'Freedom',
       'Trust (Government Corruption)', 'Generosity', 'Dystopia Residua
l'],
      dtype='object')
```

In [9]:
```
#First we will check the structure or get information about our dataset
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 14 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Unnamed: 0.1                   158 non-null    int64  
 1   Unnamed: 0                     158 non-null    int64  
 2   Country                        158 non-null    object 
 3   Region                         158 non-null    object 
 4   Happiness Rank                 158 non-null    int64  
 5   Happiness Score                158 non-null    float64
 6   Standard Error                 158 non-null    float64
 7   Economy (GDP per Capita)       158 non-null    float64
 8   Family                         158 non-null    float64
 9   Health (Life Expectancy)       158 non-null    float64
 10  Freedom                        158 non-null    float64
 11  Trust (Government Corruption)  158 non-null    float64
 12  Generosity                     158 non-null    float64
 13  Dystopia Residual              158 non-null    float64
dtypes: float64(9), int64(3), object(2)
memory usage: 17.4+ KB
```

As from the above output, it can be clearly seen that our data has 9 float datatypes, 1 integer datatype and 2 object(string) datatypes

Check for any null or missing value in dataset.

In [10]:
```
dataset.isnull()
```

Out[10]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Fa
0	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	
153	False	False	False	False	False	False	False	False	F
154	False	False	False	False	False	False	False	False	F
155	False	False	False	False	False	False	False	False	F
156	False	False	False	False	False	False	False	False	F
157	False	False	False	False	False	False	False	False	F

158 rows × 14 columns

In [11]:
```
dataset.isnull().sum()
```

Out[11]:
```
Unnamed: 0.1                     0
Unnamed: 0                       0
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 [12]:
```
dataset.isnull().sum().sum()
```

Out[12]:
```
0
```

There is Total "33" missing or null value in our dataset columns

FILLING MISSING OR NULL VALUES

METHODS:

1. Interpolation: If the data has a time series or a continuous trend, you can use interpolation methods like linear interpolation (interpolate()) to estimate missing values based on the surrounding data points.
2. Mean or median imputation: For numerical data, you can replace null values with the mean or median of the corresponding column using the fillna() method. This method assumes that the missing values are missing at random and does not introduce bias.

In [13]:
```
df1=dataset.fillna(dataset.median())
df1.head()
```

```
C:\Users\Muhammad Rafique\AppData\Local\Temp\ipykernel_6116\3846757236.py:
1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (wi
th 'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  df1=dataset.fillna(dataset.median())
```

Out[13]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510	1
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995	1
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480	1
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000	1
4	4	4	Canada	North America	5	7.427	0.03553	1.326290	1

In [14]:
```
df2=dataset.fillna(dataset.mean())
df2.head()
```

```
C:\Users\Muhammad Rafique\AppData\Local\Temp\ipykernel_6116\2101696174.py:
1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (wi
th 'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  df2=dataset.fillna(dataset.mean())
```

Out[14]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510	1
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995	1
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480	1
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000	1
4	4	4	Canada	North America	5	7.427	0.03553	1.326290	1

In [15]:
```
df3=dataset.fillna(dataset.mode())
df3.head()
```

Out[15]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510	1
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995	1
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480	1
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000	1
4	4	4	Canada	North America	5	7.427	0.03553	1.326290	1

Note Second Method used here that is mean and median.

NOTE: from generosity column first value is NAN in our data, after filling it we get the value that is not suitable for this type of data. So, this method is not valid for this type of data..

In [16]:
```
df4=dataset.interpolate()
df4
```

Out[16]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000
4	4	4	Canada	North America	5	7.427	0.03553	1.326290
...	...	...	...	...	...	...	...	...
153	153	153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.222080
154	154	154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.286650
155	155	155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.663200
156	156	156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.015300
157	157	157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.208680

158 rows × 14 columns

In [17]:
```
dataset.head()
```

Out[17]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510	1
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995	1
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480	1
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000	1
4	4	4	Canada	North America	5	7.427	0.03553	1.326290	1

Note First Method used here that is INTERPOLATION.

NOTE: Note that this method fills missing values very accurately according to that columns values..

In [18]:
```
#Check if there is any other missing value
df4.isnull().sum()
```

Out[18]:
```
Unnamed: 0.1                     0
Unnamed: 0                       0
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 [19]:
```
# to fill last value in Generosity we will use fill_na
df5=df4.fillna(value=(df4["Generosity"][1] and df4["Generosity"][2]).mean()
df5
```

Out[19]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000
4	4	4	Canada	North America	5	7.427	0.03553	1.326290
...	...	...	...	...	...	...	...	...
153	153	153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.222080
154	154	154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.286650
155	155	155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.663200
156	156	156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.015300
157	157	157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.208680

158 rows × 14 columns

NOW ALL MISSING VALUES ARE FILLED, LET'S INSERT THEM IN OUR DATASET

In [20]:
```
dataset=df5
dataset.head()
```

Out[20]:

	Unnamed: 0.1	Unnamed: 0	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	
0	0	0	Switzerland	Western Europe	1	7.587	0.03411	1.396510	1
1	1	1	Iceland	Western Europe	2	7.561	0.04884	1.360995	1
2	2	2	Denmark	Western Europe	3	7.527	0.03328	1.325480	1
3	3	3	Norway	Western Europe	4	7.522	0.03880	1.459000	1
4	4	4	Canada	North America	5	7.427	0.03553	1.326290	1

In [21]:
```
dataset.isnull().sum()
```

Out[21]:
```
Unnamed: 0.1                    0
Unnamed: 0                      0
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 [22]:
```
dataset.isnull().sum().sum()
```

Out[22]:
```
0
```

DATA IS CLEANED NOW.

NOW WE WILL EXPORT OUR CLEANED DATA

In [23]:
```
dataset.to_csv("./World_Happiness_2015.csv")
```

ANALYSIS PART:

- **We can proceed with the analysis phase, specifically focusing on correlation and covariance analysis.**

In [24]:
```
dataset.corr()
```

Out[24]:

	Unnamed: 0.1	Unnamed: 0	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Famil
Unnamed: 0.1	1.000000	1.000000	0.999998	-0.992109	0.143858	-0.789863	-0.73131
Unnamed: 0	1.000000	1.000000	0.999998	-0.992109	0.143858	-0.789863	-0.73131
Happiness Rank	0.999998	0.999998	1.000000	-0.992096	0.143824	-0.789913	-0.73127
Happiness Score	-0.992109	-0.992109	-0.992096	1.000000	-0.162790	0.784994	0.73779
Standard Error	0.143858	0.143858	0.143824	-0.162790	1.000000	-0.184516	-0.09631
Economy (GDP per Capita)	-0.789863	-0.789863	-0.789913	0.784994	-0.184516	1.000000	0.65180
Family	-0.731316	-0.731316	-0.731277	0.737798	-0.096313	0.651801	1.00000

In [25]:
```
dataset.cov()
```

Out[25]:

	Unnamed: 0.1	Unnamed: 0	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Fa
Unnamed: 0.1	2093.500000	2093.500000	2093.477707	-51.973280	0.111772	-14.397865	-9.14
Unnamed: 0	2093.500000	2093.500000	2093.477707	-51.973280	0.111772	-14.397865	-9.14
Happiness Rank	2093.477707	2093.477707	2093.461743	-51.972111	0.111745	-14.398651	-9.14
Happiness Score	-51.973280	-51.973280	-51.972111	1.310896	-0.003165	0.358064	0.23
Standard Error	0.111772	0.111772	0.111745	-0.003165	0.000288	-0.001248	-0.00
Economy (GDP per Capita)	-14.397865	-14.397865	-14.398651	0.358064	-0.001248	0.158716	0.07
Family	-9.145507	-9.145507	-9.144933	0.230881	-0.000447	0.070973	0.07
Health (Life Expectancy)	-8.293348	-8.293348	-8.293952	0.204071	-0.001256	0.078401	0.03
Freedom	-3.796170	-3.796170	-3.794578	0.096602	-0.000332	0.022219	0.01
Trust (Government Corruption)	-2.097011	-2.097011	-2.097004	0.055669	-0.000345	0.015245	0.00
Generosity	-0.916953	-0.916953	-0.916471	0.025868	-0.000137	-0.000617	0.00
Dystopia Residual	-12.966623	-12.966623	-12.966603	0.330360	0.000772	0.014557	0.02

SOME ANALYSIS QUESTIONS RELATED TO OUR DATA

1.Happiness Score and Economy (GDP per Capita):

- Is there a correlation between a country's Happiness Score and its economic prosperity?
- Do countries with higher GDP per capita tend to have higher Happiness Scores?
- How strong is the correlation between Happiness Score and the Economy?

In [26]:
```
correlation = dataset["Happiness Score"].corr(dataset["Economy (GDP per Cap
print("Correlation between Happiness Score and Economy (GDP per Capita):",
```

```
Correlation between Happiness Score and Economy (GDP per Capita): 0.8
```

2.Happiness Score and Social Support (Family):

- Is there a relationship between a country's Happiness Score and the health and life expectancy of its citizens?
- Do countries with better health indicators tend to have higher Happiness Scores?
- How does the correlation between Happiness Score and Health compare to other factors?

In [27]:
```
correlation = dataset["Happiness Score"].corr(dataset["Family"])
print("Correlation between Happiness Score and Family:", correlation.round(
```

```
Correlation between Happiness Score and Family: 0.7
```

3.Happiness Score and Freedom:

- Is there a correlation between a country's Happiness Score and the level of freedom its citizens enjoy?
- Do countries with higher levels of political and personal freedom tend to have higher Happiness Scores?
- How significant is the relationship between Happiness Score and Freedom?

In [28]:
```
correlation = dataset["Happiness Score"].corr(dataset["Freedom"])
print("Correlation between Happiness Score and Freedom:", correlation.round
```

```
Correlation between Happiness Score and Freedom: 0.6
```

4.Happiness Score and Trust (Government Corruption):

- Is there an association between a country's Happiness Score and the level of trust in its government and institutions?
- Do countries with lower levels of corruption and higher trust tend to have higher Happiness Scores?

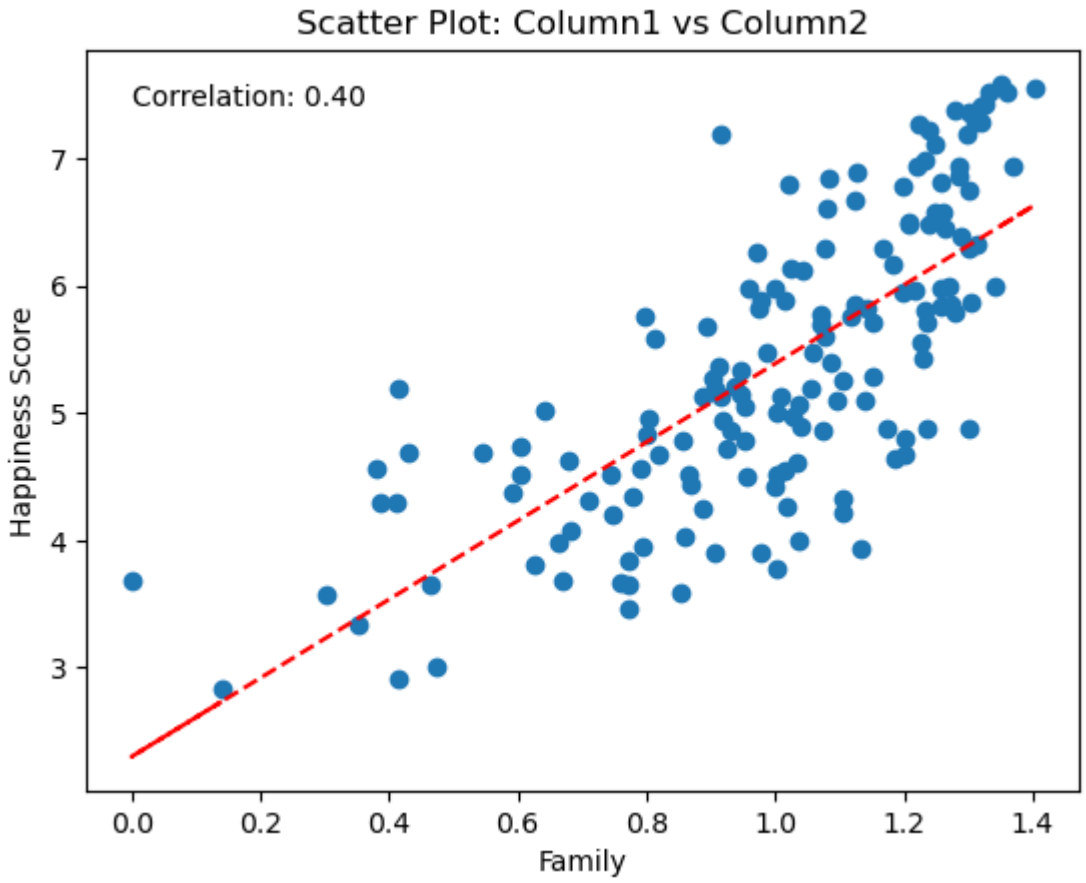
- How does the correlation between Happiness Score and Trust vary across different regions?

In [29]:
```
correlation = dataset["Happiness Score"].corr(dataset["Trust (Government Co
print("Correlation between Happiness Score and Trust (Government Corruption
```

```
Correlation between Happiness Score and Trust (Government Corruption): 0.4
```

Most Country were above 0.85 in Economy (GDP per Capita) and above 5.38 in Happiness Score Qatar had the highest Economy (GDP per Capita) (1.69). Switzerland had the highest Happiness Score (7.59). The Region with the highest Economy (GDP per Capita) was Middle East and Northern Africa. The Region with the highest Happiness Score was Western Europe.

In [30]:
```
plt.scatter(dataset['Family'], dataset['Happiness Score'])
plt.xlabel('Family')
plt.ylabel('Happiness Score')
plt.title('Scatter Plot: Column1 vs Column2')
plt.text(dataset["Family"].min(), dataset["Happiness Score"].max(), f'Corre
coefficients = np.polyfit(dataset['Family'], dataset['Happiness Score'], 1)
trendline = np.poly1d(coefficients)
plt.plot(dataset['Family'], trendline(dataset['Family']), color='r', linest
plt.show()
```



In [ ]: