

Statistics

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

File directory

```
In [2]: data=pd.read_csv(r"C:\Users\user\Desktop\Vicky\2_2015.csv")
```

All Mathematical function

```
In [3]: data.describe()
```

Out[3]:

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(G C
count	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	1
mean	79.493671	5.375734	0.047885	0.846137	0.991046	0.630259	0.428615	
std	45.754363	1.145010	0.017146	0.403121	0.272369	0.247078	0.150693	
min	1.000000	2.839000	0.018480	0.000000	0.000000	0.000000	0.000000	
25%	40.250000	4.526000	0.037268	0.545808	0.856823	0.439185	0.328330	
50%	79.500000	5.232500	0.043940	0.910245	1.029510	0.696705	0.435515	
75%	118.750000	6.243750	0.052300	1.158448	1.214405	0.811013	0.549092	
max	158.000000	7.587000	0.136930	1.690420	1.402230	1.025250	0.669730	

To display the top portion of the dataset

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

Out[4]:

	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	0.661
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.621
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	0.631

To display the mean median mode of the dataset for only numerical value

In [7]: `data1=data[["Happiness Rank","Happiness Score"]]
print(data1.mean())
print(data1.mode())
print(data1.median())
print()`

```
Happiness Rank      79.493671
Happiness Score      5.375734
dtype: float64
Happiness Rank      Happiness Score
0                82                5.192
Happiness Rank      79.500000
Happiness Score      5.232500
Standard Error      0.043940
Economy (GDP per Capita)  0.910245
Family              1.029510
Health (Life Expectancy)  0.696705
Freedom             0.435515
Trust (Government Corruption)  0.107220
Generosity          0.216130
Dystopia Residual    2.095415
dtype: float64
```

To display the total of each columns

In [8]: `print(data1.sum())`

```
Happiness Rank      12560.000
Happiness Score      849.366
dtype: float64
```

To displayn the minimum value

In [9]: `print(data1.min())`

```
Happiness Rank      1.000
Happiness Score      2.839
dtype: float64
```

To displayn theCumulative sum

In [10]: `print(data1.cumsum())`

	Happiness Rank	Happiness Score
0	1	7.587
1	3	15.148
2	6	22.675
3	10	30.197
4	15	37.624
..
153	11934	837.276
154	12089	840.616
155	12245	843.622
156	12402	846.527
157	12560	849.366

[158 rows x 2 columns]

To count the total number of values in columns

In [11]: `print(data.count())`

Country	158
Region	158
Happiness Rank	158
Happiness Score	158
Standard Error	158
Economy (GDP per Capita)	158
Family	158
Health (Life Expectancy)	158
Freedom	158
Trust (Government Corruption)	158
Generosity	158
Dystopia Residual	158
dtype: int64	

In [12]: `print(data1.cov())`

	Happiness Rank	Happiness Score
Happiness Rank	2093.461743	-51.975613
Happiness Score	-51.975613	1.311048

In [13]: `from scipy.stats import spearmanr`
`from scipy.stats import pearsonr`

In [17]: `data2=data[["Family","Freedom"]]`
`data2`

Out[17]:

	Family	Freedom
0	1.34951	0.66557
1	1.40223	0.62877
2	1.36058	0.64938
3	1.33095	0.66973
4	1.32261	0.63297
...
153	0.77370	0.59201
154	0.35386	0.48450
155	0.47489	0.15684
156	0.41587	0.11850
157	0.13995	0.36453

158 rows × 2 columns

In [18]: `print(spearmanr(data1,data2))`

```
SpearmanrResult(correlation=array([[ 1.          , -1.          , -0.77003796, -
0.57635217],
[ -1.          ,  1.          ,  0.77003796,  0.57635217],
[ -0.77003796,  0.77003796,  1.          ,  0.52813911],
[ -0.57635217,  0.57635217,  0.52813911,  1.          ]]), pvalue=array
([[0.00000000e+00, 0.00000000e+00, 2.93848445e-32, 2.30130102e-15],
[0.00000000e+00, 0.00000000e+00, 2.93848445e-32, 2.30130102e-15],
[2.93848445e-32, 2.93848445e-32, 0.00000000e+00, 9.93778697e-13],
[2.30130102e-15, 2.30130102e-15, 9.93778697e-13, 0.00000000e+00]]))
```

In []:

In []:

