

# import libraries

```
In [1]: import numpy as np
import pandas as pd
```

import dataset

```
In [2]: data=pd.read_csv(r"C:\Users\user\Downloads\2015.csv")
data
```

Out[2]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Free
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



mean

```
In [3]: print(data.mean())
```

```
Happiness Rank          79.493671
Happiness Score          5.375734
Standard Error           0.047885
Economy (GDP per Capita) 0.846137
Family                   0.991046
Health (Life Expectancy) 0.630259
Freedom                  0.428615
Trust (Government Corruption) 0.143422
Generosity               0.237296
Dystopia Residual         2.098977
dtype: float64
```

median

```
In [8]: print(data.median())
```

```
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
```

mode

```
In [10]: print(data.describe())
```

	Happiness Rank	Happiness Score	Standard Error	\
count	158.000000	158.000000	158.000000	
mean	79.493671	5.375734	0.047885	
std	45.754363	1.145010	0.017146	
min	1.000000	2.839000	0.018480	
25%	40.250000	4.526000	0.037268	
50%	79.500000	5.232500	0.043940	
75%	118.750000	6.243750	0.052300	
max	158.000000	7.587000	0.136930	

	Economy (GDP per Capita)	Family	Health (Life Expectancy)	\
count	158.000000	158.000000	158.000000	
mean	0.846137	0.991046	0.630259	
std	0.403121	0.272369	0.247078	
min	0.000000	0.000000	0.000000	
25%	0.545808	0.856823	0.439185	
50%	0.910245	1.029510	0.696705	
75%	1.158448	1.214405	0.811013	
max	1.690420	1.402230	1.025250	

	Freedom	Trust (Government Corruption)	Generosity	\
count	158.000000	158.000000	158.000000	
mean	0.428615	0.143422	0.237296	
std	0.150693	0.120034	0.126685	
min	0.000000	0.000000	0.000000	
25%	0.328330	0.061675	0.150553	
50%	0.435515	0.107220	0.216130	
75%	0.549092	0.180255	0.309883	
max	0.669730	0.551910	0.795880	

	Dystopia Residual
count	158.000000
mean	2.098977
std	0.553550
min	0.328580
25%	1.759410
50%	2.095415
75%	2.462415
max	3.602140

In [30]: `print(data.sum())`

```
Country                SwitzerlandIcelandDenmarkNorwayCanadaFinland
Ne...
Region                Western EuropeWestern EuropeWestern EuropeWe
st...
Happiness Rank
12560
Happiness Score                84
9.366
Standard Error                7.
56579
Economy (GDP per Capita)        133.
68968
Family                156.
58526
Health (Life Expectancy)        99.
58098
Freedom                67.
72116
Trust (Government Corruption)    22.
66065
Generosity                37.
49269
Dystopia Residual                331.
63833
dtype: object
```

In [25]: `df = pd.DataFrame(data[["Happiness Rank", "Happiness Score"]])`  
`df`

Out[25]:

	Happiness Rank	Happiness Score
0	1	7.587
1	2	7.561
2	3	7.527
3	4	7.522
4	5	7.427
...	...	...
153	154	3.465
154	155	3.340
155	156	3.006
156	157	2.905
157	158	2.839

158 rows × 2 columns

In [26]: `print(df.mode())`

	Happiness Rank	Happiness Score
0	82	5.192

In [27]: `print(df.mean())`

	Happiness Rank	Happiness Score
	79.493671	5.375734

dtype: float64

In [28]: `print(df.median())`

	Happiness Rank	Happiness Score
	79.5000	5.2325

dtype: float64

In [29]: `print(df.describe())`

	Happiness Rank	Happiness Score
count	158.000000	158.000000
mean	79.493671	5.375734
std	45.754363	1.145010
min	1.000000	2.839000
25%	40.250000	4.526000
50%	79.500000	5.232500
75%	118.750000	6.243750
max	158.000000	7.587000

In [32]: `print(df.sum())`

	Happiness Rank	Happiness Score
	12560.000	849.366

dtype: float64

In [33]: `print(df.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]

```
In [34]: print(df.min())
```

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

```
In [35]: print(df.max())
```

```
Happiness Rank    158.000  
Happiness Score     7.587  
dtype: float64
```

```
In [36]: print(df.count())
```

```
Happiness Rank     158  
Happiness Score     158  
dtype: int64
```

```
In [37]: from numpy import cov
```

```
In [38]: print(cov(df))
```

```
[[ 2.16942845e+01  1.83151535e+01  1.49096745e+01 ... -5.03885739e+02  
 -5.07511882e+02 -5.11022753e+02]  
 [ 1.83151535e+01  1.54623605e+01  1.25873235e+01 ... -4.25399817e+02  
 -4.28461147e+02 -4.31425161e+02]  
 [ 1.49096745e+01  1.25873235e+01  1.02468645e+01 ... -3.46301919e+02  
 -3.48794033e+02 -3.51206924e+02]  
 ...  
 [-5.03885739e+02 -4.25399817e+02 -3.46301919e+02 ...  1.17035820e+04  
  1.17878052e+04  1.18693510e+04]  
 [-5.07511882e+02 -4.28461147e+02 -3.48794033e+02 ...  1.17878052e+04  
  1.18726345e+04  1.19547671e+04]  
 [-5.11022753e+02 -4.31425161e+02 -3.51206924e+02 ...  1.18693510e+04  
  1.19547671e+04  1.20374680e+04]]
```

```
In [40]: from scipy.stats import pearsonr
```

```
In [45]: df1 = df["Happiness Rank"][0:100]
df2 = df["Happiness Score"][0:100]
df1
df2
```

```
Out[45]: 0      7.587
1      7.561
2      7.527
3      7.522
4      7.427
...
95     4.949
96     4.898
97     4.885
98     4.876
99     4.874
Name: Happiness Score, Length: 100, dtype: float64
```

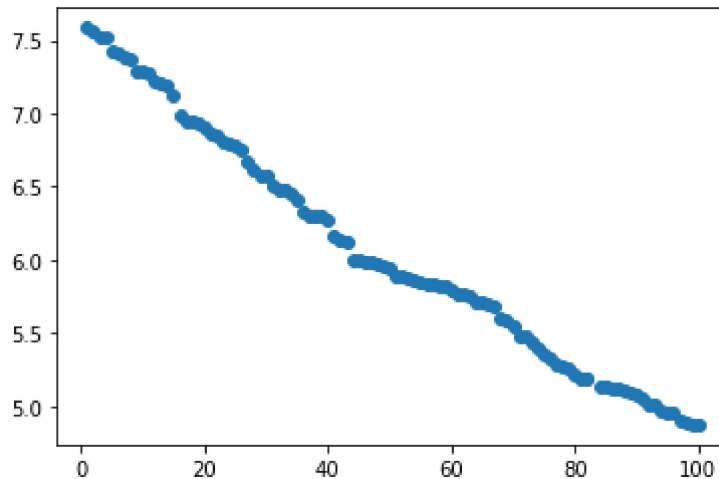
```
In [46]: print(pearsonr(df1,df2))

(-0.9923267072800684, 8.735160340905535e-91)
```

```
In [47]: import matplotlib.pyplot as plot
```

```
In [48]: plot.scatter(df1,df2)
```

```
Out[48]: <matplotlib.collections.PathCollection at 0x2097e220fa0>
```



```
In [ ]:
```