

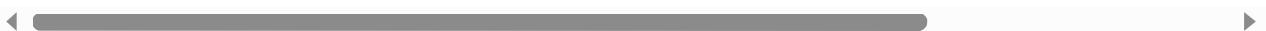
```
In [1]:  
import numpy as np  
import pandas as pd
```

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

Out[2]:

	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	0.66557
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297
...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453

158 rows × 12 columns



```
In [3]:  
a.isnull()
```

Out[3] :

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(Govt Cor
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...
153	False	False	False	False	False	False	False	False	False	False
154	False	False	False	False	False	False	False	False	False	False
155	False	False	False	False	False	False	False	False	False	False
156	False	False	False	False	False	False	False	False	False	False
157	False	False	False	False	False	False	False	False	False	False

158 rows × 12 columns



In [4] :

a.fillna(value=True)

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	0.66557
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297
...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450
155	Syria	Middle East and	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
		Northern Africa							
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453

158 rows × 12 columns

In [5]:

```
a.dropna()
```

Out[5]:

	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	0.66557
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297
...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453

158 rows × 12 columns

In [6]: `a.head(5)`

Out[6]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(Go C
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	

◀ ▶

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

Out[7]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
152	Afghanistan	Southern Asia	153	3.575	0.03084	0.31982	0.30285	0.30335	0.23414
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453

◀ ▶

In [8]: `a.describe()`

Out[8]:

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)
count	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000
mean	79.493671	5.375734	0.047885	0.846137	0.991046	0.630259	0.428615	0.14342
std	45.754363	1.145010	0.017146	0.403121	0.272369	0.247078	0.150693	0.12003
min	1.000000	2.839000	0.018480	0.000000	0.000000	0.000000	0.000000	0.000000
25%	40.250000	4.526000	0.037268	0.545808	0.856823	0.439185	0.328330	0.06167
50%	79.500000	5.232500	0.043940	0.910245	1.029510	0.696705	0.435515	0.10722
75%	118.750000	6.243750	0.052300	1.158448	1.214405	0.811013	0.549092	0.18025
max	158.000000	7.587000	0.136930	1.690420	1.402230	1.025250	0.669730	0.55191

```
◀ ━━━━━━ ▶
```

In [9]:

a.iloc[0:5]

Out[9]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	

```
◀ ━━━━━━ ▶
```

In [10]:

a.loc[0:4]

Out[10]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	

Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(Government Corruption)
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297

In [12]: `np.shape(a)`

Out[12]: (158, 12)

In [13]: `np.size(a)`

Out[13]: 1896

In [14]: `np.ndim(a)`

Out[14]: 2

In [15]: `a.columns`

Out[15]: 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 [16]: `a.index`

Out[16]: RangeIndex(start=0, stop=158, step=1)

In [17]: `a1=a[["Family", "Freedom"]]`
`a1`

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

```
Family Freedom
```

```
156 0.41587 0.11850
157 0.13995 0.36453
```

158 rows × 2 columns

```
In [18]: a1.mean()
```

```
Out[18]: Family      0.991046
Freedom     0.428615
dtype: float64
```

```
In [19]: a1.median()
```

```
Out[19]: Family      1.029510
Freedom     0.435515
dtype: float64
```

```
In [20]: a1.mode()
```

```
Out[20]:      Family Freedom
```

```
0 0.00000 0.00000
1 0.13995 0.07699
2 0.30285 0.09245
3 0.35386 0.10081
4 0.38174 0.10384
...
153 1.34043 0.65821
154 1.34951 0.65980
155 1.36058 0.66246
156 1.36948 0.66557
157 1.40223 0.66973
```

158 rows × 2 columns

```
In [21]: a1.sum()
```

```
Out[21]: Family      156.58526
Freedom     67.72116
dtype: float64
```

```
In [22]: a1.cumsum()
```

```
Out[22]:
```

	Family	Freedom
0	1.34951	0.66557
1	2.75174	1.29434
2	4.11232	1.94372
3	5.44327	2.61345
4	6.76588	3.24642
...
153	155.20069	66.59679
154	155.55455	67.08129
155	156.02944	67.23813
156	156.44531	67.35663
157	156.58526	67.72116

158 rows × 2 columns

```
In [23]:
```

```
a1.max()
```

```
Out[23]: Family      1.40223
Freedom     0.66973
dtype: float64
```

```
In [24]:
```

```
a1.min()
```

```
Out[24]: Family      0.0
Freedom     0.0
dtype: float64
```

```
In [25]:
```

```
a1.count()
```

```
Out[25]: Family      158
Freedom     158
dtype: int64
```

```
In [26]:
```

```
from numpy import cov
from scipy.stats import pearsonr
from scipy.stats import spearmanr
```

```
In [27]:
```

```
b1=a["Family"].values
b2=a["Freedom"].values
print(np.cov(b1,b2))
```

```
[[0.07418492  0.0181217 ]
 [ 0.0181217   0.02270832]]
```

```
In [28]:
```

```
spearman =pd.Series(b1).corr(pd.Series(b2),method='spearman')
pearson =pd.Series(b1).corr(pd.Series(b2),method='pearson')
```

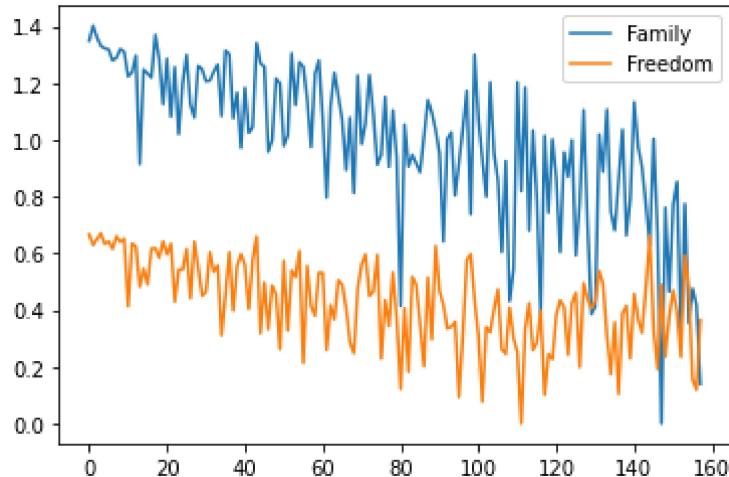
```
print(spearman)
print(pearson)
```

```
0.5281391142435108
0.4415182106228604
```

visualization

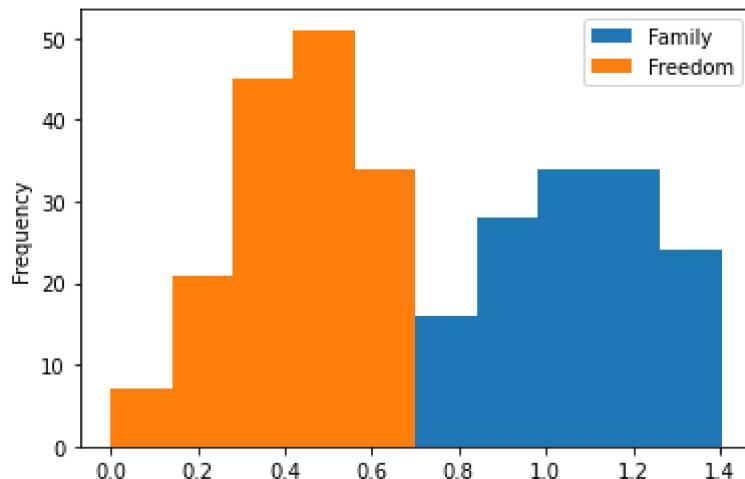
```
In [30]: a1.plot.line()
```

```
Out[30]: <AxesSubplot:>
```



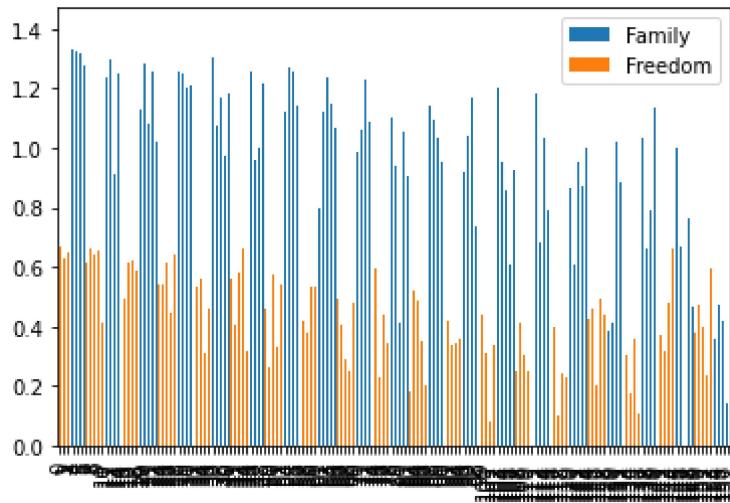
```
In [31]: a1.plot.hist()
```

```
Out[31]: <AxesSubplot:ylabel='Frequency'>
```



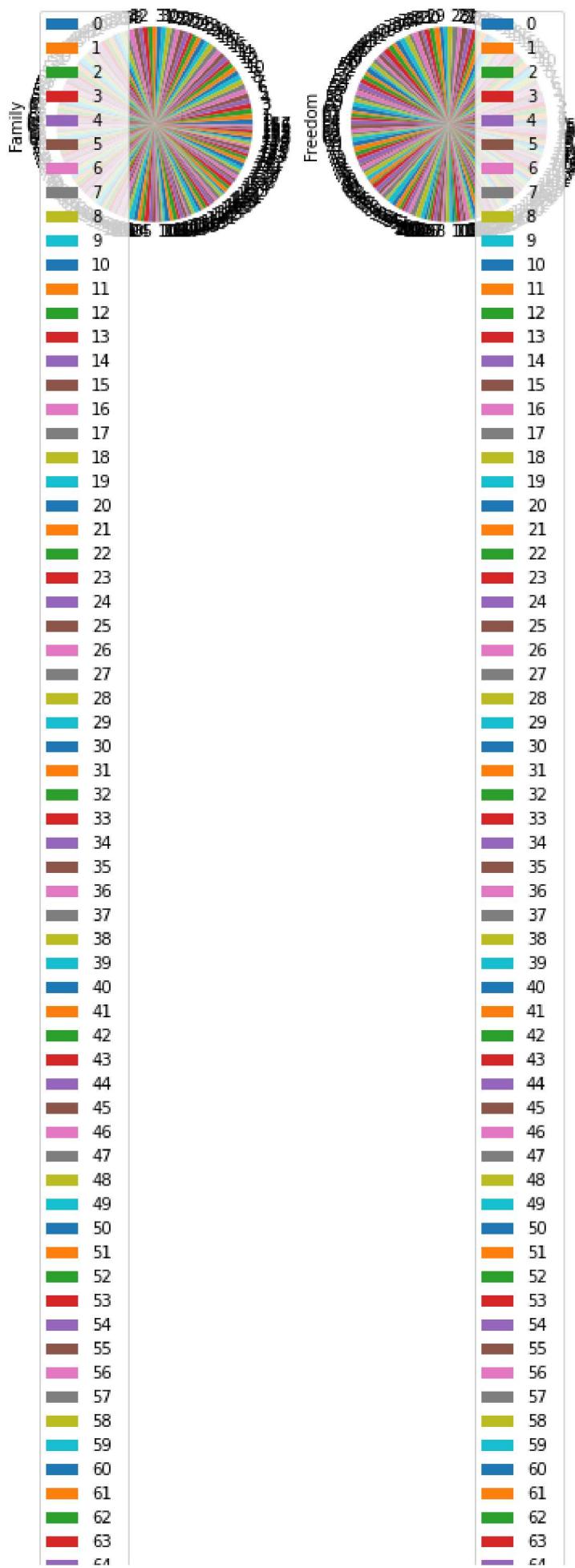
```
In [32]: a1.plot.bar()
```

```
Out[32]: <AxesSubplot:>
```



```
In [34]: a1.plot.pie(subplots=True)
```

```
Out[34]: array([<AxesSubplot:ylabel='Family'>, <AxesSubplot:ylabel='Freedom'>],  
dtype=object)
```



64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130

