

SUMESH R - 20104169

Basic Analysis using NumPy and Pandas

Import Libraries

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

```
In [2]: import numpy as np
```

Import Dataset

```
In [3]: data = pd.read_csv("2015.csv")
```

```
In [4]: display(data)
```

| | 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 |

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

To display top 10 rows

In [5]: `data.head(10)`

Out[5]:

| | Country | Region | Happiness Rank | Happiness Score | Standard Error | Economy (GDP per Capita) | Family | Health (Life Expectancy) | Freedom | (G |
|---|-------------|---------------------------|----------------|-----------------|----------------|--------------------------|---------|--------------------------|---------|----|
| 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 | |
| 5 | Finland | Western Europe | 6 | 7.406 | 0.03140 | 1.29025 | 1.31826 | 0.88911 | 0.64169 | |
| 6 | Netherlands | Western Europe | 7 | 7.378 | 0.02799 | 1.32944 | 1.28017 | 0.89284 | 0.61576 | |
| 7 | Sweden | Western Europe | 8 | 7.364 | 0.03157 | 1.33171 | 1.28907 | 0.91087 | 0.65980 | |
| 8 | New Zealand | Australia and New Zealand | 9 | 7.286 | 0.03371 | 1.25018 | 1.31967 | 0.90837 | 0.63938 | |
| 9 | Australia | Australia and New Zealand | 10 | 7.284 | 0.04083 | 1.33358 | 1.30923 | 0.93156 | 0.65124 | |

to display last 5 rows

In [6]:

```
data.tail()
```

Out[6]:

| | Country | Region | Happiness Rank | Happiness Score | Standard Error | Economy (GDP per Capita) | Family | Health (Life Expectancy) | Freedom | (Governmer |
|-----|---------|---------------------------------|----------------|-----------------|----------------|--------------------------|---------|--------------------------|---------|------------|
| 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 | |

statistical summary

In [7]:

```
data.describe()
```

Out[7]:

| | Happiness Rank | Happiness Score | Standard Error | Economy (GDP per Capita) | Family | Health (Life Expectancy) | Freedom | True (Governmer |
|-------|----------------|-----------------|----------------|--------------------------|------------|--------------------------|------------|-----------------|
| 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 |

To print number of elements

In [8]: `data.size`

Out[8]: 1896

to print number of row and cols

In [9]: `data.shape`

Out[9]: (158, 12)

to find missing values

In [10]: `data.isna()`

Out[10]:

| | 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



drop null values

In [11]: `data.dropna()`

Out[11]:

| | Country | Region | Happiness Rank | Happiness Score | Standard Error | Economy (GDP per Capita) | Family | Health (Life Expectancy) | Freedom |
|---|-------------|---------|----------------|-----------------|----------------|--------------------------|---------|--------------------------|---------|
| 0 | Switzerland | Western | 1 | 7.587 | 0.03411 | 1.39651 | 1.34951 | 0.94143 | 0.66557 |

| | Country | Region | Happiness Rank | Happiness Score | Standard Error | Economy (GDP per Capita) | Family | Health (Life Expectancy) | Freedom |
|-----|---------|---------------------------------|----------------|-----------------|----------------|--------------------------|---------|--------------------------|---------|
| | | Europe | | | | | | | |
| 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

to select a particular columns

In [13]:

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

Out[13]:

| | Happiness Rank | Happiness Score |
|---|----------------|-----------------|
| 0 | 1 | 7.587 |
| 1 | 2 | 7.561 |
| 2 | 3 | 7.527 |
| 3 | 4 | 7.522 |
| 4 | 5 | 7.427 |

Happiness Rank Happiness Score

| ... | ... | ... |
|------------|-----|-------|
| 153 | 154 | 3.465 |
| 154 | 155 | 3.340 |
| 155 | 156 | 3.006 |
| 156 | 157 | 2.905 |
| 157 | 158 | 2.839 |

158 rows × 2 columns

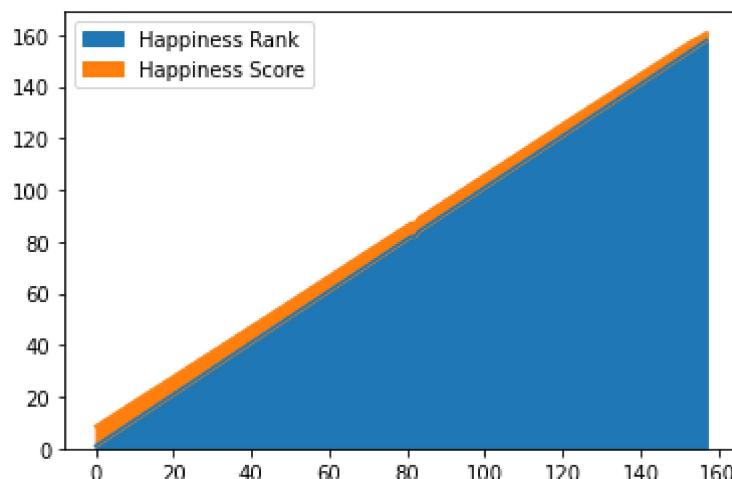
line plot

In [14]: `df.plot.line()`

Out[14]: <AxesSubplot:>

In [15]: `df.plot.area()`

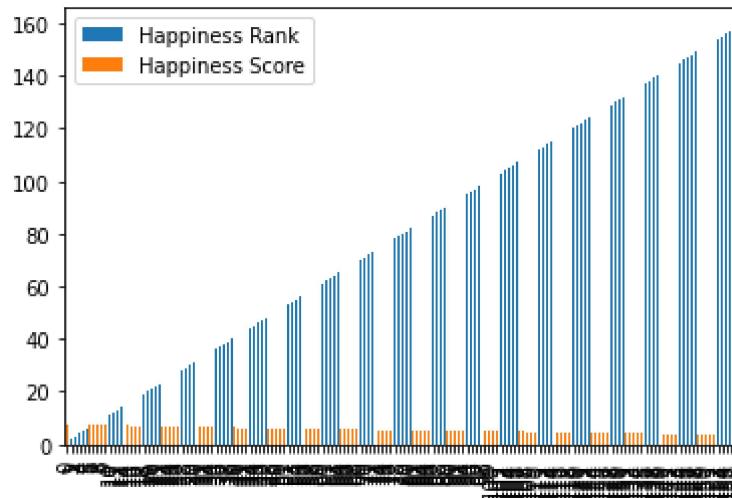
Out[15]: <AxesSubplot:>



bar chart

```
In [16]: df.plot.bar()
```

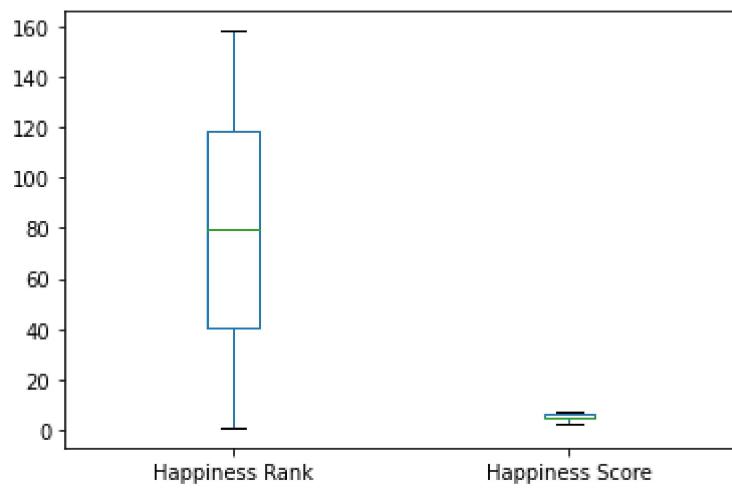
```
Out[16]: <AxesSubplot:>
```



box plot

```
In [17]: df.plot.box()
```

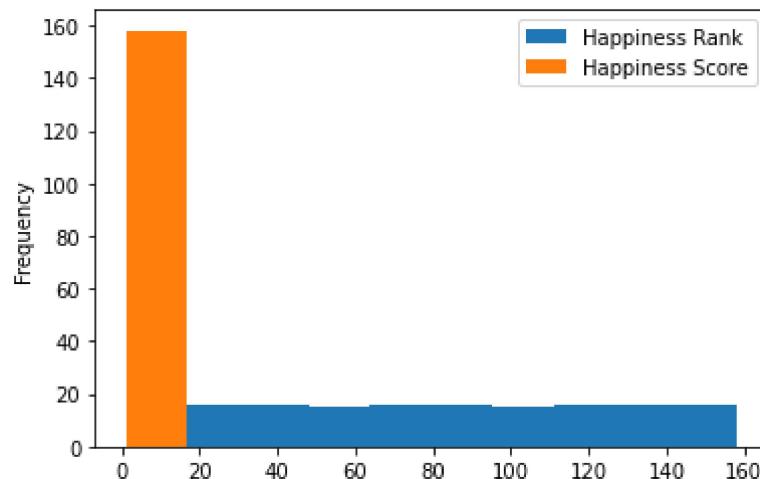
```
Out[17]: <AxesSubplot:>
```



histogram

```
In [18]: df.plot.hist()
```

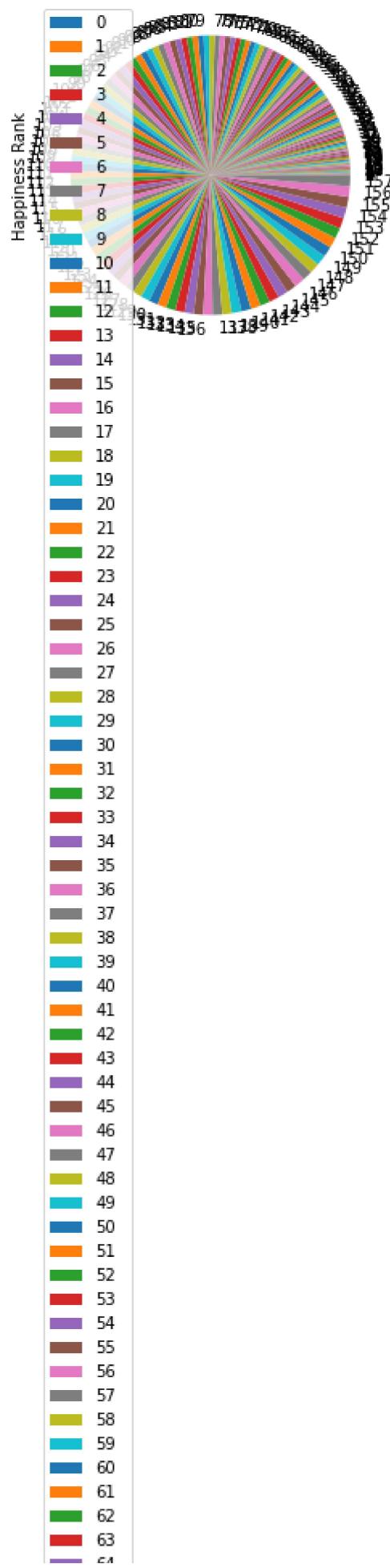
```
Out[18]: <AxesSubplot:ylabel='Frequency'>
```



pie plot

```
In [20]: df.plot.pie(y="Happiness Rank")
```

```
Out[20]: <AxesSubplot:ylabel='Happiness Rank'>
```

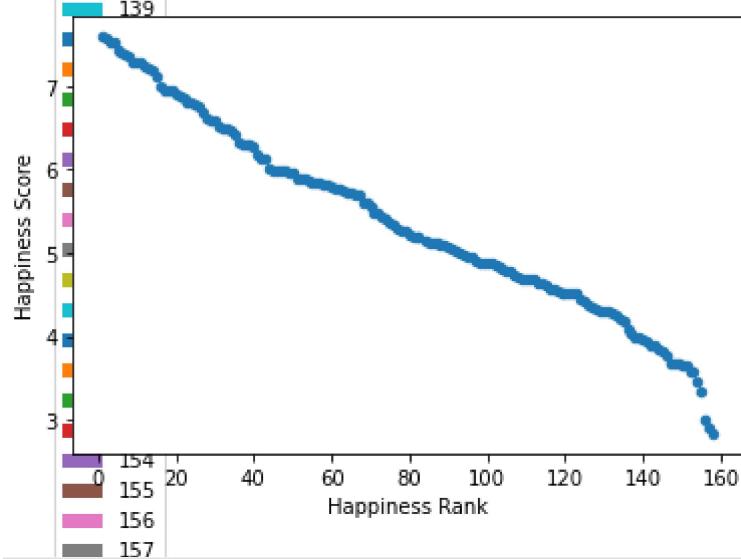


| |
|-----|
| 64 |
| 65 |
| 66 |
| 67 |
| 68 |
| 69 |
| 70 |
| 71 |
| 72 |
| 73 |
| 74 |
| 75 |
| 76 |
| 77 |
| 78 |
| 79 |
| 80 |
| 81 |
| 82 |
| 83 |
| 84 |
| 85 |
| 86 |
| 87 |
| 88 |
| 89 |
| 90 |
| 91 |
| 92 |
| 93 |
| 94 |
| 95 |
| 96 |
| 97 |
| 98 |
| 99 |
| 100 |
| 101 |
| 102 |
| 103 |
| 104 |
| 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 |

Scatter plot

```
In [22]: df.plot.scatter(x="Happiness Rank", y="Happiness Score")
```

```
Out[22]: <AxesSubplot:xlabel='Happiness Rank', ylabel='Happiness Score'>
```



```
In [ ]:
```