```
import numpy as np
import pandas as pd
import matplotlib as lib
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
from google.colab import drive
drive.mount('2022.csv')
```

```
Drive already mounted at 2022.csv; to attempt to forcibly remount, call drive.mount("2022.csv", force_remount=True).
```

```
df=pd.read_csv('/2022.csv')
df
```

	RANK	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	nan	Explained by: Perceptions of corruption
0	1.0	Finland	7821.0	7886.0	7756.0	2518.0	1892.0	1258.0	775.0	736.0	109.0	534.0
1	2.0	Denmark	7636.0	7710.0	7563.0	2226.0	1953.0	1243.0	777.0	719.0	188.0	532.0
2	3.0	Iceland	7557.0	7651.0	7464.0	2320.0	1936.0	1320.0	803.0	718.0	270.0	191.0
3	4.0	Switzerland	7512.0	7586.0	7437.0	2153.0	2026.0	1226.0	822.0	677.0	147.0	461.0
4	5.0	Netherlands	7415.0	7471.0	7359.0	2137.0	1945.0	1206.0	787.0	651.0	271.0	419.0
...	...	...	...	...	...	...	...	...	...	...	...	...
142	143.0	Rwanda*	3268.0	3462.0	3074.0	536.0	785.0	133.0	462.0	621.0	187.0	544.0
143	144.0	Zimbabwe	2995.0	3110.0	2880.0	548.0	947.0	690.0	270.0	329.0	106.0	105.0
144	145.0	Lebanon	2955.0	3049.0	2862.0	216.0	1392.0	498.0	631.0	103.0	82.0	34.0
145	146.0	Afghanistan	2404.0	2469.0	2339.0	1263.0	758.0	0.0	289.0	0.0	89.0	5.0
146	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
print(df.isnull().sum())
```

```
RANK                                         1
Country                                      1
Happiness score                              1
Whisker-high                                 1
Whisker-low                                  1
Dystopia (1.83) + residual                   1
Explained by: GDP per capita                 1
Explained by: Social support                 1
Explained by: Healthy life expectancy        1
Explained by: Freedom to make life choices   1
nan                                          1
Explained by: Perceptions of corruption      1
dtype: int64
```

```
df_new=df.head(30)
```

VISUALIZATION

```
df_col1=(df.columns)
df_col1
```

```
Index(['RANK', 'Country', 'Happiness score', 'Whisker-high', 'Whisker-low',
       'Dystopia (1.83) + residual', 'Explained by: GDP per capita',
       'Explained by: Social support', 'Explained by: Healthy life expectancy',
       'Explained by: Freedom to make life choices', 'nan',
       'Explained by: Perceptions of corruption'],
      dtype='object')
```

```
df_sorted = df_new.sort_values(by='Explained by: Social support', ascending = False)
df_sorted
```

	RANK	Country	score	high	low	(1.83) + residual	per capita	Social support
2	3.0	Iceland	7557.0	7651.0	7464.0	2320.0	1936.0	1320.0
17	18.0	Czechia	6920.0	7029.0	6811.0	2263.0	1815.0	1260.0
0	1.0	Finland	7821.0	7886.0	7756.0	2518.0	1892.0	1258.0
21	22.0	Slovenia	6630.0	6718.0	6542.0	1885.0	1810.0	1249.0
1	2.0	Denmark	7636.0	7710.0	7563.0	2226.0	1953.0	1243.0
7	8.0	Norway	7365.0	7440.0	7290.0	1925.0	1997.0	1239.0
9	10.0	New Zealand	7200.0	7279.0	7120.0	1954.0	1852.0	1235.0
3	4.0	Switzerland	7512.0	7586.0	7437.0	2153.0	2026.0	1226.0
8	9.0	Israel	7364.0	7426.0	7301.0	2634.0	1826.0	1221.0
19	20.0	France	6687.0	6758.0	6615.0	1895.0	1863.0	1219.0
28	29.0	Spain	6476.0	6560.0	6392.0	1893.0	1808.0	1211.0
4	5.0	Netherlands	7415.0	7471.0	7359.0	2137.0	1945.0	1206.0
6	7.0	Sweden	7384.0	7454.0	7315.0	2003.0	1920.0	1204.0
11	12.0	Australia	7162.0	7244.0	7081.0	2011.0	1900.0	1203.0
14	15.0	Canada	7025.0	7107.0	6943.0	1924.0	1886.0	1188.0
15	16.0	United States	6977.0	7065.0	6888.0	2214.0	1982.0	1182.0
29	30.0	Uruguay	6474.0	6562.0	6386.0	1974.0	1615.0	1180.0
12	13.0	Ireland	7041.0	7121.0	6961.0	1743.0	2129.0	1166.0
10	11.0	Austria	7163.0	7237.0	7089.0	2148.0	1931.0	1165.0
5	6.0	Luxembourg*	7404.0	7501.0	7307.0	2042.0	2209.0	1155.0
16	17.0	United Kingdom	6943.0	7018.0	6867.0	1967.0	1867.0	1143.0
26	27.0	Singapore	6480.0	6569.0	6392.0	932.0	2149.0	1127.0
18	19.0	Belgium	6805.0	6890.0	6720.0	2283.0	1907.0	1106.0
25	26.0	Taiwan Province of China	6512.0	6596.0	6429.0	2002.0	1897.0	1095.0
24	25.0	Saudi Arabia	6523.0	6637.0	6409.0	2075.0	1870.0	1092.0
13	14.0	Germany	7034.0	7122.0	6947.0	2142.0	1924.0	1088.0
22	23.0	Costa Rica	6582.0	6683.0	6481.0	2346.0	1584.0	1054.0

```
df_sorted = df.sort_values(by='RANK', ascending=True)

columns_to_fill = ['Happiness score', 'Whisker-high', 'Whisker-low', 'Dystopia (1.83) + residual',
                   'Explained by: GDP per capita', 'Explained by: Social support',
                   'Explained by: Healthy life expectancy', 'Explained by: Freedom to make life choices',
                   'Explained by: Perceptions of corruption']

df_sorted[columns_to_fill] = df_sorted[columns_to_fill].fillna(df_sorted[columns_to_fill].mean())

df_sorted
```

	RANK	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	Explained by: GDP per capita	b
0	1.0	Finland	7821.000000	7886.000000	7756.000000	2518.000000	1892.000000	12
1	2.0	Denmark	7636.000000	7710.000000	7563.000000	2226.000000	1953.000000	12
2	3.0	Iceland	7557.000000	7651.000000	7464.000000	2320.000000	1936.000000	13
3	4.0	Switzerland	7512.000000	7586.000000	7437.000000	2153.000000	2026.000000	12
4	5.0	Netherlands	7415.000000	7471.000000	7359.000000	2137.000000	1945.000000	12
...	...	...	...	...	...	...	...	
142	143.0	Rwanda*	3268.000000	3462.000000	3074.000000	536.000000	785.000000	1
143	144.0	Zimbabwe	2995.000000	3110.000000	2880.000000	548.000000	947.000000	6
144	145.0	Lebanon	2955.000000	3049.000000	2862.000000	216.000000	1392.000000	4
145	146.0	Afghanistan	2404.000000	2469.000000	2339.000000	1263.000000	758.000000	

```
print(df_sorted.isnull().sum())
```

```
    RANK                                        1
    Country                                     1
    Happiness score                             0
    Whisker-high                                0
    Whisker-low                                 0
    Dystopia (1.83) + residual                  0
    Explained by: GDP per capita                0
    Explained by: Social support                0
    Explained by: Healthy life expectancy       0
    Explained by: Freedom to make life choices  0
    nan                                         1
    Explained by: Perceptions of corruption     0
    dtype: int64
```

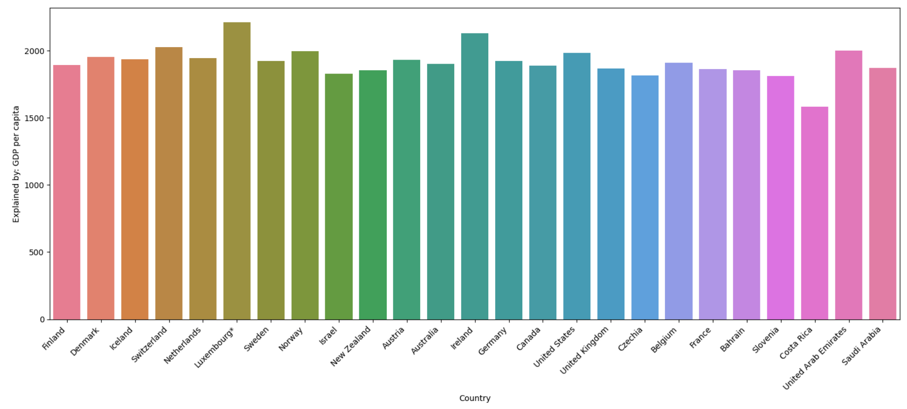
```
df_sorted=df_sorted.dropna()
print(df_sorted.isnull().sum())
```

```
    RANK                                        0
    Country                                     0
    Happiness score                             0
    Whisker-high                                0
    Whisker-low                                 0
    Dystopia (1.83) + residual                  0
    Explained by: GDP per capita                0
    Explained by: Social support                0
    Explained by: Healthy life expectancy       0
    Explained by: Freedom to make life choices  0
    nan                                         0
    Explained by: Perceptions of corruption     0
    dtype: int64
```

VISUALIZATION

GDP rates of Countries

```
df_n = df_sorted.head(25)
plt.rcParams['figure.figsize'] = (19, 7)
sns.barplot(x=df_n['Country'], y=df_n['Explained by: GDP per capita'], hue =df_n['Country'])
plt.xlabel=("Country")
plt.ylabel=("Happiness score")
plt.xticks( rotation=45, ha='right')
plt.show()
```



```
converted_columns = ['Whisker-low', 'Whisker-high']

df_sorted['Average'] =  df_sorted[converted_columns].mean(axis = 1)
```

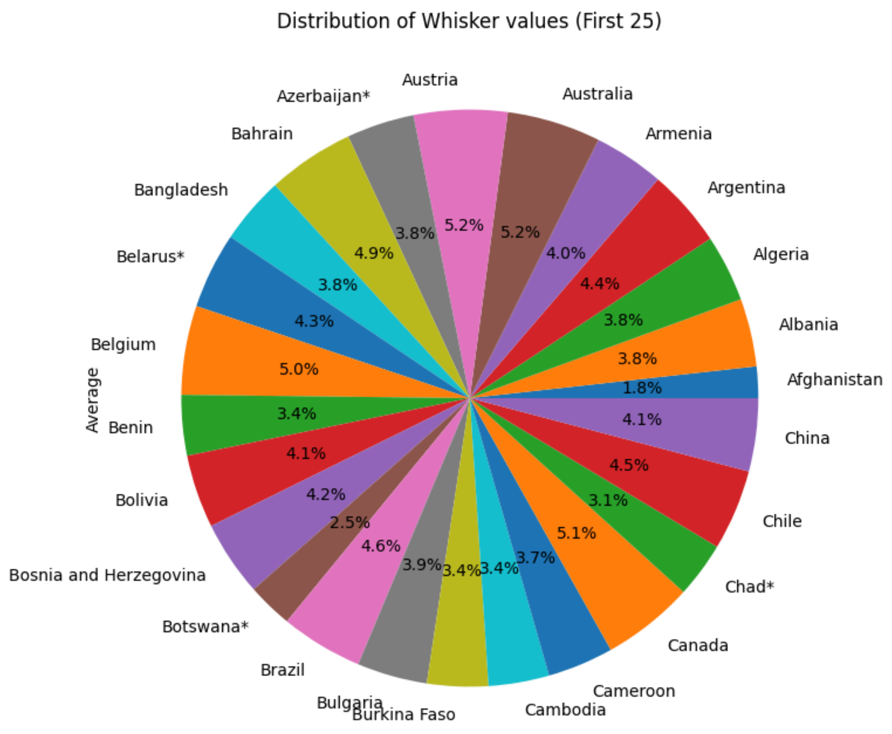
```
    <ipython-input-319-320e3971dfb5>:2: SettingWithCopyWarning:
    A value is trying to be set on a copy of a slice from a DataFrame.
    Try using .loc[row_indexer,col_indexer] = value instead

    See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
      df_sorted.loc[:, 'Average'] = df_sorted[converted_columns].mean(axis=1)
```

```
whisker = df_sorted.groupby('Country')['Average'].sum()
whisker
```

```
    Country
    Afghanistan    2404.0
    Albania        5198.5
    Algeria        5122.5
    Argentina      5967.0
    Armenia        5398.5
                    ...
    Venezuela      4925.5
    Vietnam        5485.0
    Yemen*         4197.0
    Zambia         3760.0
    Zimbabwe       2995.0
    Name: Average, Length: 146, dtype: float64
```

Average Whisker Rates of Countries

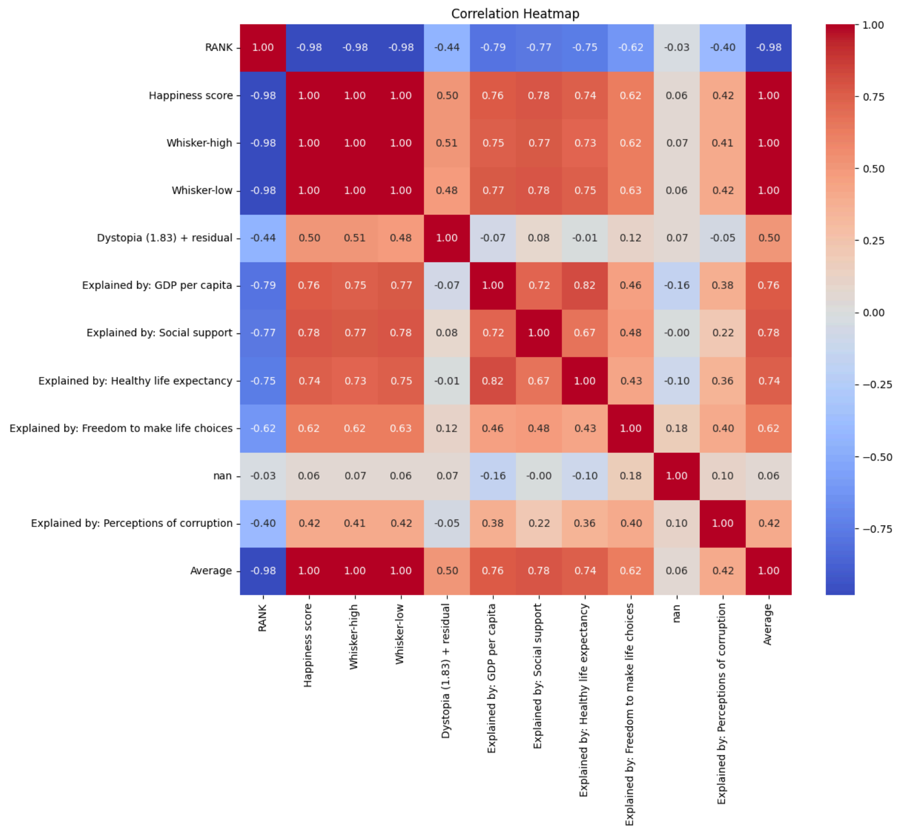
```
whisker = whisker.head(25)
plt.figure(figsize=(8, 8))
whisker.plot.pie(autopct='%1.1f%%')
plt.title("Distribution of Whisker values (First 25)")
plt.show()
```



Correaltion of Whisker High rates and Whisker low rates with their average

```
correlation_matrix = df_sorted.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```

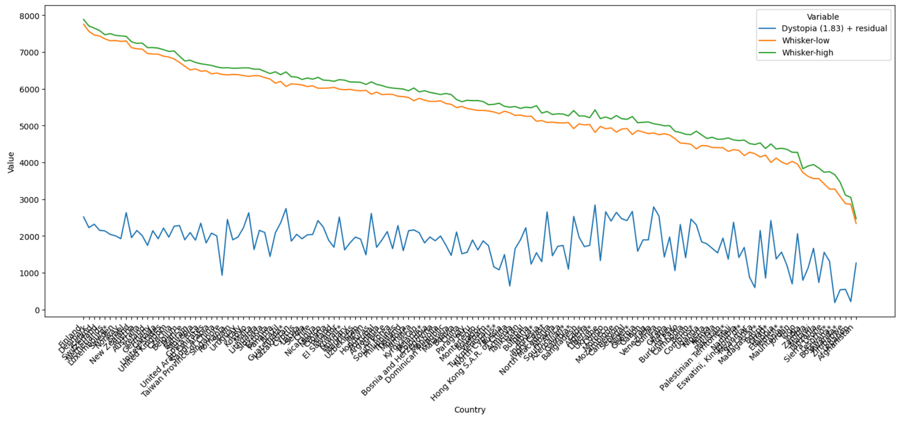
```
<ipython-input-312-ddd84dfbb460>:1: FutureWarning: The default value of numeric_only in
  correlation_matrix = df_sorted.corr()
```



	RANK	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	nan	Explained by: Perceptions of corruption	Average
RANK	1.00	-0.98	-0.98	-0.98	-0.44	-0.79	-0.77	-0.75	-0.62	-0.03	-0.40	-0.98
Happiness score	-0.98	1.00	1.00	1.00	0.50	0.76	0.78	0.74	0.62	0.06	0.42	1.00
Whisker-high	-0.98	1.00	1.00	1.00	0.51	0.75	0.77	0.73	0.62	0.07	0.41	1.00
Whisker-low	-0.98	1.00	1.00	1.00	0.48	0.77	0.78	0.75	0.63	0.06	0.42	1.00
Dystopia (1.83) + residual	-0.44	0.50	0.51	0.48	1.00	-0.07	0.08	-0.01	0.12	0.07	-0.05	0.50
Explained by: GDP per capita	-0.79	0.76	0.75	0.77	-0.07	1.00	0.72	0.82	0.46	-0.16	0.38	0.76
Explained by: Social support	-0.77	0.78	0.77	0.78	0.08	0.72	1.00	0.67	0.48	-0.00	0.22	0.78
Explained by: Healthy life expectancy	-0.75	0.74	0.73	0.75	-0.01	0.82	0.67	1.00	0.43	-0.10	0.36	0.74
Explained by: Freedom to make life choices	-0.62	0.62	0.62	0.63	0.12	0.46	0.48	0.43	1.00	0.18	0.40	0.62
nan	-0.03	0.06	0.07	0.06	0.07	-0.16	-0.00	-0.10	0.18	1.00	0.10	0.06
Explained by: Perceptions of corruption	-0.40	0.42	0.41	0.42	-0.05	0.38	0.22	0.36	0.40	0.10	1.00	0.42
Average	-0.98	1.00	1.00	1.00	0.50	0.76	0.78	0.74	0.62	0.06	0.42	1.00

Analysing the factors of Dystopia and Whisker of Countries

```
selected_columns = ['Country', 'Dystopia (1.83) + residual', 'Whisker-low', 'Whisker-high']
df_selected = df[selected_columns]
df_selected_melted = pd.melt(df_selected, id_vars=['Country'], var_name='Variable', value_name='Value')
plt.xticks( rotation=45, ha='right')
sns.lineplot(data=df_selected_melted, x='Country', y='Value', hue='Variable', markers=True, dashes=False)
plt.show()
```

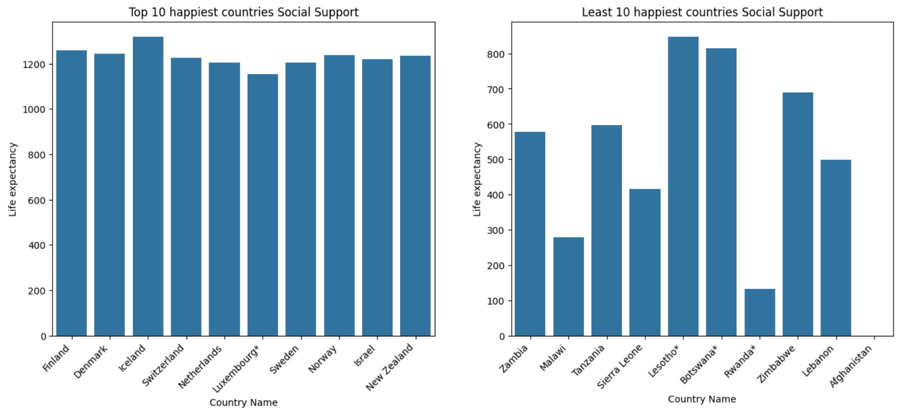


```
first_ten = df_sorted.head(10)
last_ten = df_sorted.tail(10)
```

Comparing the Social support of first 25 countries and the last 25 countries

```
fig, axes = plt.subplots(1, 2, figsize=(16, 6))
labels_top = first_ten.Country
axes[0].set_title('Top 10 happiest countries Social Support')
axes[0].set_xticklabels(labels_top, rotation=45, ha='right')
sns.barplot(x=first_ten.Country, y=first_ten['Explained by: Social support'], ax=axes[0] )
axes[0].set_xlabel('Country Name')
axes[0].set_ylabel('Life expectancy')


labels_bottom = last_ten.Country
axes[1].set_title('Least 10 happiest countries Social Support')
axes[1].set_xticklabels(labels_bottom, rotation=45, ha='right')
sns.barplot(x=last_ten.Country, y=last_ten['Explained by: Social support'], ax=axes[1])
axes[1].set_xlabel('Country Name')
axes[1].set_ylabel('Life expectancy')

plt.show()
```

```
<ipython-input-315-67a989f12243>:4: UserWarning: FixedFormatter should only be used toge
  axes[0].set_xticklabels(labels_top, rotation=45, ha='right')
<ipython-input-315-67a989f12243>:12: UserWarning: FixedFormatter should only be used tog
  axes[1].set_xticklabels(labels_bottom, rotation=45, ha='right')
```



```
print(df_sorted['Happiness score'].dtype)
```

```
float64
```

**How the Freedom to make choices of a country affects the Happiness Score **

```
df_sort = df_sorted.head(25)
plt.rcParams['figure.figsize'] = (15, 7)
sns.scatterplot(x=df_sort['Explained by: Freedom to make life choices'], y=df_sort['Happiness score'], hue=df_sort['Country'])
plt.show()
```

Perceptions of corruption

```
n_n_v = pd.to_numeric(df_sorted['Explained by: Perceptions of corruption'])
df_sorted['Explained by: Perceptions of corruption'].hist(bins=30, color='black')
plt.title("Perceptions of corruption")
```