In [1]:

```
import time                          # To time processes
import warnings                      # To suppress warnings

import numpy as np                   # Data manipulation
import pandas as pd                  # Dataframe manipulatio
import matplotlib.pyplot as plt                         # For graphics
import seaborn as sns
import plotly.graph_objects as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot,
init_notebook_mode(connected=True)

from sklearn.preprocessing import StandardScaler  # For scaling dataset
from sklearn.cluster import KMeans, AgglomerativeClustering, AffinityPro
from sklearn.mixture import GaussianMixture #For GMM clustering

import os                            # For os related operations
import sys
```

In [2]:

```
# world happiness score: https://worldhappiness.report/ed/2019/

wh = pd.read_csv("2017.csv") #Read the dataset
#wh = wh.set_index('Country')
print('The shape of this dataset: ', wh.shape)
wh.head(5)
```

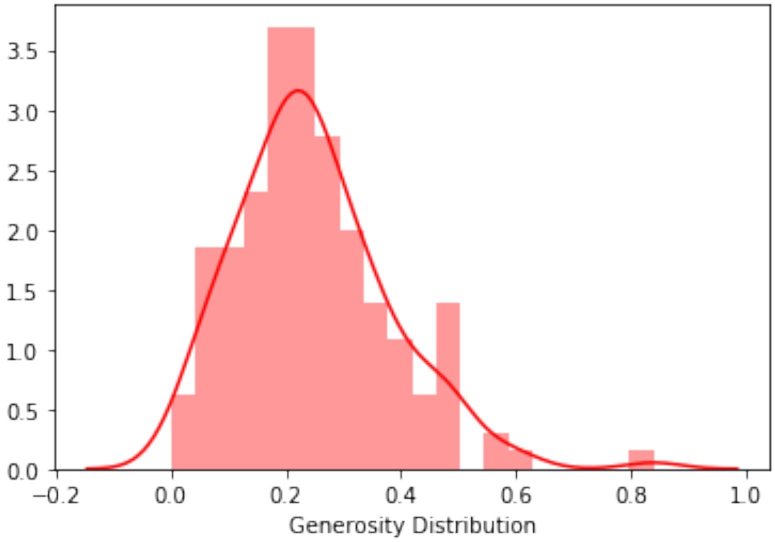
```
The shape of this dataset:  (155, 10)
```

Out[2]:

	Country	Happiness.Rank	Happiness.Score	Economy..GDP.per.Capita.	Family	Health..Life
0	Norway	1	7.537	1.616463	1.533524	
1	Denmark	2	7.522	1.482383	1.551122	
2	Iceland	3	7.504	1.480633	1.610574	
3	Switzerland	4	7.494	1.564980	1.516912	
4	Finland	5	7.469	1.443572	1.540247	

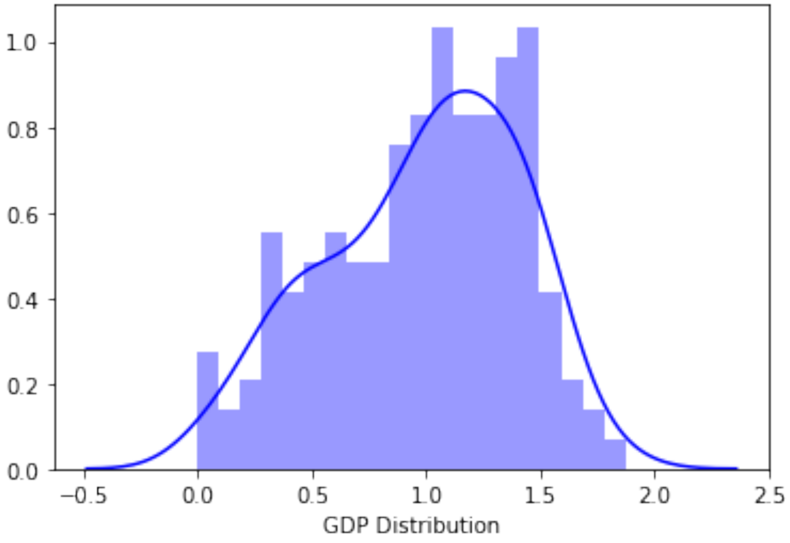
```
In [3]: Generosity = wh[['Generosity']] #explore distribution of factor Generosi
        sns.distplot(Generosity,bins=20,color="r",
                     axlabel='Generosity Distribution')
```

```
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1b53c050>
```



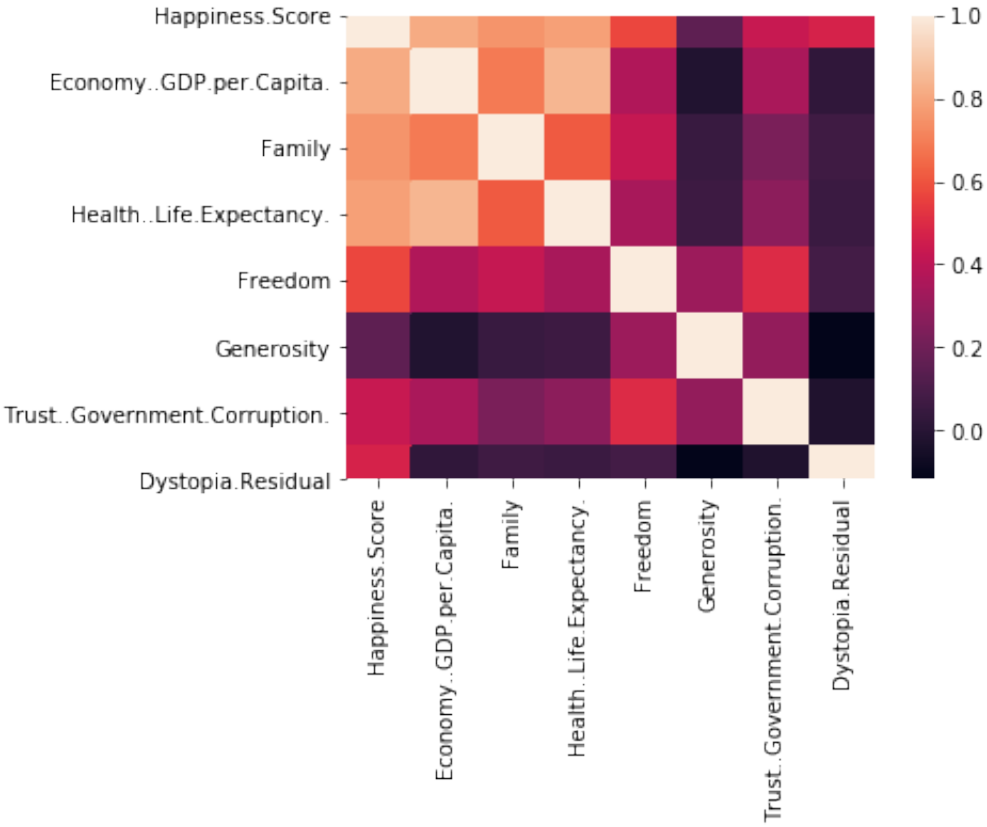
```
In [4]: GDP = wh[['Economy..GDP.per.Capita.']] #explore distribution of factor G
        sns.distplot(GDP,bins=20,color="b",
                     axlabel='GDP Distribution')
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1be18350>
```



```
In [5]: wh1 = wh[['Happiness.Score','Economy..GDP.per.Capita.','Family','Health.
                  'Generosity','Trust..Government.Corruption.','Dystopia.Residua
        cor = wh1.corr() #Calculate the correlation of the above variables
        sns.heatmap(cor, square = True) #Plot the correlation as heat map
```

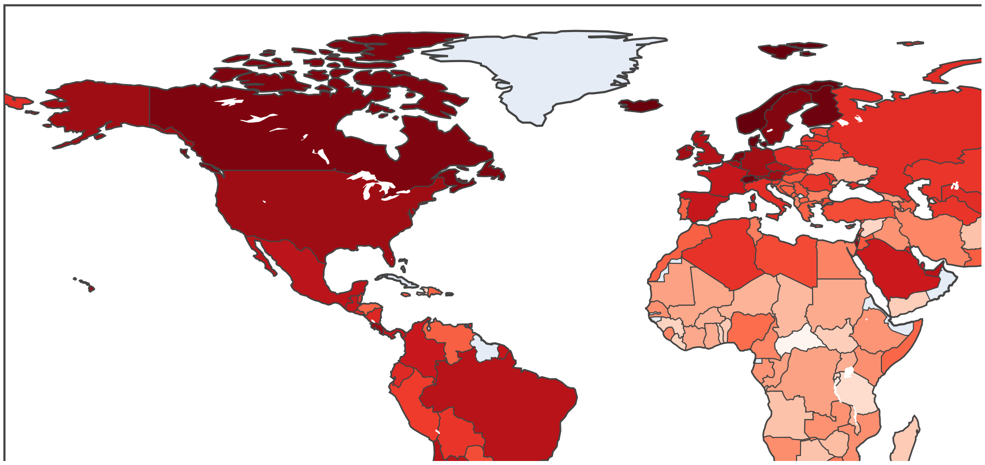
```
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1c034950>
```



In [35]:

```
fig = go.Figure(data=go.Choropleth(
    locations = wh['Country'], # set up locations(countries)
    locationmode='country names',
    z = wh['Happiness.Score'],# this is the value we try to display
    text = wh['Country'],# let the map show country name
    colorscale = 'Reds',# color scale
    autocolorscale=False,
    reversescale=False,
    marker_line_width=0.5,
    colorbar_title = 'Happiness',
)) # visulize happiness score world distribution

fig.show()
```



In [ ]:

In [ ]: