

INTRODUCTION

My Objective:

What countries or regions rank the highest in overall happiness and each of the six factors contributing to happiness? How did country ranks or scores change between 2016 and 2019 reports?

1.Data processing with pandas(report 2016)

1.1 Loading data:

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt

In [2]: #Loading file
path16="C://Users//SS//Desktop//World Happiness Report//Data//"
file_name16="2016.csv"
df16=pd.read_csv(path16+file_name16,sep=',')
df16
```

Out[2]:

	Country	Region	Happiness Rank	Happiness Score	Lower Confidence Interval	Upper Confidence Interval	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity
0	Denmark	Western Europe	1	7.526	7.460	7.592	1.44178	1.16374	0.79504	0.57941	0.44453	0.3617
1	Switzerland	Western Europe	2	7.509	7.428	7.590	1.52733	1.14524	0.86303	0.58557	0.41203	0.2808
2	Iceland	Western Europe	3	7.501	7.333	7.669	1.42666	1.18326	0.86733	0.56624	0.14975	0.4767
3	Norway	Western Europe	4	7.498	7.421	7.575	1.57744	1.12690	0.79579	0.59609	0.35776	0.3789
4	Finland	Western Europe	5	7.413	7.351	7.475	1.40598	1.13464	0.81091	0.57104	0.41004	0.2549
...
152	Benin	Sub-Saharan Africa	153	3.484	3.404	3.564	0.39499	0.10419	0.21028	0.39747	0.06681	0.2018
153	Afghanistan	Southern Asia	154	3.360	3.288	3.432	0.38227	0.11037	0.17344	0.16430	0.07112	0.3126
154	Togo	Sub-Saharan Africa	155	3.303	3.192	3.414	0.28123	0.00000	0.24811	0.34678	0.11587	0.1751
155	Syria	Middle East and Northern Africa	156	3.069	2.936	3.202	0.74719	0.14866	0.62994	0.06912	0.17233	0.4839
156	Burundi	Sub-Saharan Africa	157	2.905	2.732	3.078	0.06831	0.23442	0.15747	0.04320	0.09419	0.2029

157 rows × 13 columns

1.2 Data exploration:

```
In [3]: #The data type of each column
df16.dtypes
```

Out[3]:

Country	object
Region	object
Happiness Rank	int64
Happiness Score	float64
Lower Confidence Interval	float64
Upper Confidence Interval	float64
Economy (GDP per Capita)	float64
Family	float64
Health (Life Expectancy)	float64
Freedom	float64
Trust (Government Corruption)	float64
Generosity	float64
Dystopia Residual	float64
dtype: object	

```
In [4]: #Generate descriptive statistics
df16.describe()
```

Out[4]:

	Happiness Rank	Happiness Score	Lower Confidence Interval	Upper Confidence Interval	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
count	157.000000	157.000000	157.000000	157.000000	157.000000	157.000000	157.000000	157.000000	157.000000	157.000000	157.000000
mean	78.980892	5.382185	5.282395	5.481975	0.953880	0.793621	0.557619	0.370994	0.137624	0.242635	2.325807
std	45.466030	1.141674	1.148043	1.136493	0.412595	0.266706	0.229349	0.145507	0.111038	0.133756	0.542220
min	1.000000	2.905000	2.732000	3.078000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.817890
25%	40.000000	4.404000	4.327000	4.465000	0.670240	0.641840	0.382910	0.257480	0.061260	0.154570	2.031710
50%	79.000000	5.314000	5.237000	5.419000	1.027800	0.841420	0.596590	0.397470	0.105470	0.222450	2.290740
75%	118.000000	6.269000	6.154000	6.434000	1.279640	1.021520	0.729930	0.484530	0.175540	0.311850	2.664650
max	157.000000	7.526000	7.460000	7.669000	1.824270	1.183260	0.952770	0.608480	0.505210	0.819710	3.837720



1.3 Cleaning and preparation:

In [5]:

```
#Detect missing values
df16.isna().sum()
```

Out[5]:

```
Country          0
Region          0
Happiness Rank 0
Happiness Score 0
Lower Confidence Interval 0
Upper Confidence Interval 0
Economy (GDP per Capita) 0
Family          0
Health (Life Expectancy) 0
Freedom          0
Trust (Government Corruption) 0
Generosity        0
Dystopia Residual 0
dtype: int64
```

In [6]:

```
#Detect missing values
df16.isnull().sum()
```

Out[6]:

```
Country          0
Region          0
Happiness Rank 0
Happiness Score 0
Lower Confidence Interval 0
Upper Confidence Interval 0
Economy (GDP per Capita) 0
Family          0
Health (Life Expectancy) 0
Freedom          0
Trust (Government Corruption) 0
Generosity        0
Dystopia Residual 0
dtype: int64
```

1.4 Analysye

1.4.1 How many counties are there in the report?

In [7]:

```
len(df16['Country'])
```

Out[7]:

157

1.4.2 The number of countries each region

In [8]:

```
df16.groupby('Region').count()['Country']
```

Out[8]:

Region	Count
Australia and New Zealand	2
Central and Eastern Europe	29
Eastern Asia	6
Latin America and Caribbean	24
Middle East and Northern Africa	19
North America	2
Southeastern Asia	9
Southern Asia	7
Sub-Saharan Africa	38
Western Europe	21

Name: Country, dtype: int64

1.4.3 Top 10 countries which have highest happiness score?

In [9]:

```
df16_1=df16.sort_values('Happiness Rank', ascending=True)[:10][['Country', 'Region', 'Happiness Rank']]
df16_1
```

Out[9]:

	Country	Region	Happiness Rank
0	Denmark	Western Europe	1

Country	Region	Happiness Rank
1 Switzerland	Western Europe	2
2 Iceland	Western Europe	3
3 Norway	Western Europe	4
4 Finland	Western Europe	5
5 Canada	North America	6
6 Netherlands	Western Europe	7
7 New Zealand	Australia and New Zealand	8
8 Australia	Australia and New Zealand	9
9 Sweden	Western Europe	10

1.4.4 Which region has the most countries in top 10?

```
In [10]: df16_2=df16_1.groupby('Region').count().iloc[:, :1]
df16_2
```

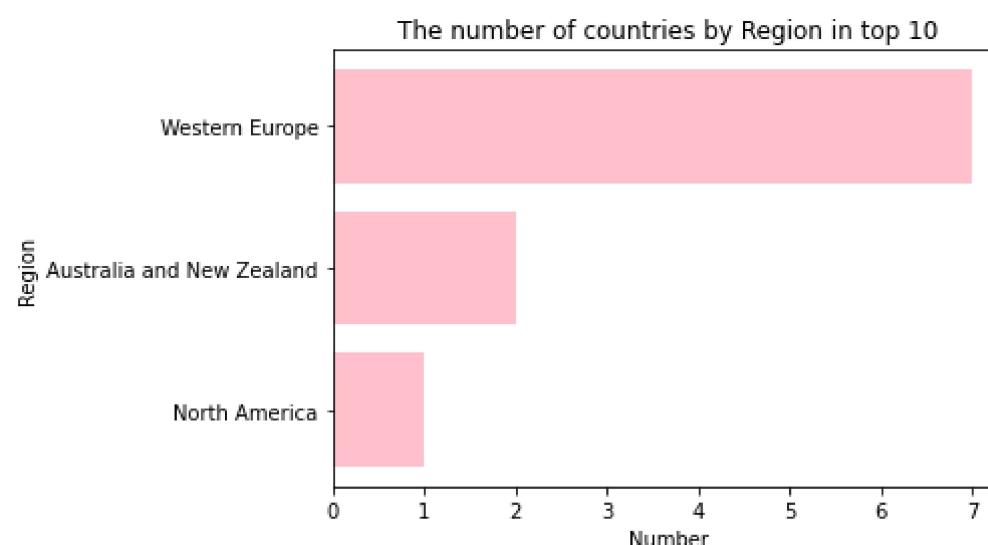
Out[10]:

Country	Region
Australia and New Zealand	2
North America	1
Western Europe	7

1.4.5 Chart 1: The number of countries by region in top 10

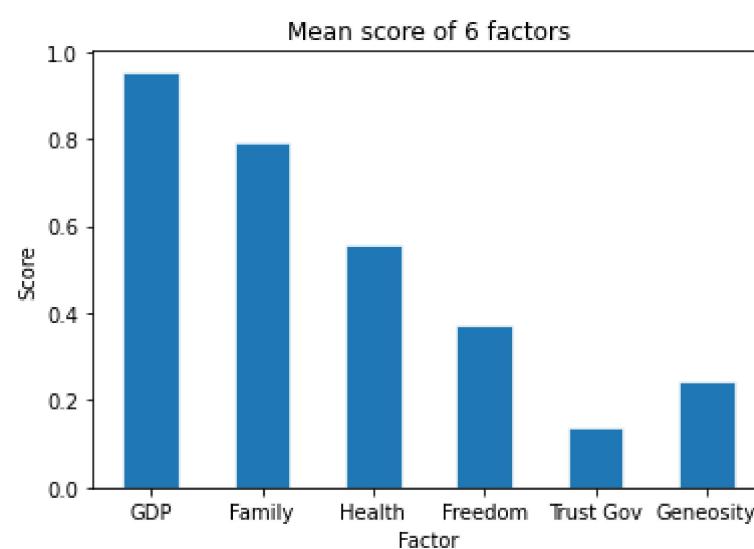
```
In [15]: Num_country=list(df16_2['Country'])
Num_country.sort()
Region=['North America','Australia and New Zealand','Western Europe']
fig,ax=plt.subplots()
chart_1=ax.barh(Region,Num_country, align='center',color='pink')
ax.set_xlabel('Number')
ax.set_ylabel('Region')
ax.set_title('The number of countries by Region in top 10')
```

Out[15]: Text(0.5, 1.0, 'The number of countries by Region in top 10')



1.4.6 Chart 2: The mean score of 6 factors

```
In [18]: six_factors=list(df16.head(0).iloc[:, 6:12])# six factors affect happiness score
df16_4=df16[six_factors].mean()
df16_5=pd.DataFrame(df16_4,columns=['Mean_Score'],index=None)
df16_4.index
#make data
x=['GDP','Family','Health','Freedom','Trust Gov','Generosity']
y=df16_5['Mean_Score']
fig, ax = plt.subplots()
Chart2=ax.bar(x, y,width=0.5, edgecolor="white", linewidth=0.7)
ax.set_xlabel('Factor')
ax.set_ylabel('Score')
ax.set_title('Mean score of 6 factors')
plt.show()
```



2. The changes from 2016-2019

```
In [85]: #Loading and rename columns 2017 report
path17="C://Users//SS//Desktop//World Happiness Report//Data//"
file_name17="2017.csv"
df17=pd.read_csv(path17+file_name17,sep=',')
df17_1=df17.rename(columns={
    'Happiness.Rank': 'Happiness Rank',
    'Happiness.Score':'Happiness Score'})
```

```
In [109...]: # six factors contributing happiness score in 2017 report
six_factors17=list(df17_1.head(0).iloc[:,5:11])
df17_2=df17_1[six_factors17].mean()
df17_2
```

```
Out[109...]: Economy..GDP.per.Capita.      0.984718
Family                  1.188898
Health..Life.Expectancy. 0.551341
Freedom                 0.408786
Generosity               0.246883
Trust..Government.Corruption. 0.123120
dtype: float64
```

```
In [87]: #Top 3 countries which have highest score in 2017
df17_3=df17_1.sort_values('Happiness Rank',ascending=True)[:3][['Happiness Rank','Country',]]#sort by "Happiness Rank"-->
df17_3['Year']=2017
df17_3
```

```
Out[87]:   Happiness Rank Country Year
0             1    Norway  2017
1             2  Denmark  2017
2             3   Iceland  2017
```

```
In [88]: #top 3 countries which have the highest score 2016
df16_4=df16_1.sort_values('Happiness Rank',ascending=True)[:3][['Happiness Rank','Country']]
df16_4['Year']=2016
df16_4
```

```
Out[88]:   Happiness Rank Country Year
0             1  Denmark  2016
1             2  Switzerland  2016
2             3   Iceland  2016
```

```
In [89]: #Combine top 3 countries which have the highest score in 2016 và 2017
df16_17=pd.concat([df17_3,df16_4],ignore_index=False).sort_values(['Happiness Rank','Year'],ascending=True)
df16_17
```

```
Out[89]:   Happiness Rank Country Year
0             1  Denmark  2016
0             1    Norway  2017
1             2  Switzerland  2016
1             2  Denmark  2017
2             3   Iceland  2016
2             3   Iceland  2017
```

```
In [90]: #Loading 2018 report
path18="C://Users//SS//Desktop//World Happiness Report//Data//"
file_name18="2018.csv"
```

```
df18=pd.read_csv(path18+file_name18,sep=',')
df18['Year']=2018
```

In [91]:

```
#Loading 2019 report
path19="C://Users//SS//Desktop//World Happiness Report//Data//"
file_name19="2019.csv"
df19=pd.read_csv(path19+file_name19,sep=',')
df19['Year']=2019
```

In [92]:

```
# concat df18 and df19
df18_19=pd.concat([df18,df19],ignore_index=False)

# rename 2 columns which
#are similar names of df16 and df17
df18_19_1=df18_19.rename(columns={'Overall rank':'Happiness Rank','Country or region':'Country'})
```

In [93]:

```
# sort top 3 happiest countries
top3=[]
for i in df18_19_1['Happiness Rank']:
    if i<=3:
        top3.append('Y')
    else:
        top3.append('N')
df18_19_1['top3']=top3
df18_19_2=df18_19_1[df18_19_1['top3']=='Y'].sort_values(['Happiness Rank','Year'],ascending=True)[['Happiness Rank','Cour
```

2.1 Top 3 happiest countries between 2016 and 2019

In [94]:

```
df_4y=pd.concat([df16_17,df18_19_2],ignore_index=False).sort_values(['Happiness Rank','Year'])
```

Out[94]:

	Happiness Rank	Country	Year
0	1	Denmark	2016
0	1	Norway	2017
0	1	Finland	2018
0	1	Finland	2019
1	2	Switzerland	2016
1	2	Denmark	2017
1	2	Norway	2018
1	2	Denmark	2019
2	3	Iceland	2016
2	3	Iceland	2017
2	3	Denmark	2018
2	3	Norway	2019

In [95]:

```
# six factors contributing happiness score 2018-2019
six_factors18_19=list(df18.head(0).iloc[:,3:9])
six_factors18_19
```

Out[95]:

```
['GDP per capita',
 'Social support',
 'Healthy life expectancy',
 'Freedom to make life choices',
 'Generosity',
 'Perceptions of corruption']
```

In [96]:

```
#Mean score 6 factors in 2018 and 2019 reports
df18_19_1=df18_19.groupby('Year').mean()[six_factors18_19]
df18_19_1
```

Out[96]:

Year	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
2018	0.891449	1.213237	0.597346	0.454506	0.181006	0.112000
2019	0.905147	1.208814	0.725244	0.392571	0.184846	0.110603

In [108...]:

```
#Mean score of 6 factors in 2017 report
df16_4=df16[six_factors].mean()
df16_4
```

Out[108...]:

Economy (GDP per Capita)	0.953880
Family	0.793621
Health (Life Expectancy)	0.557619
Freedom	0.370994
Trust (Government Corruption)	0.137624

```
Generosity          0.242635
dtype: float64
```

In [107...]: #Mean score of 6 factors in 2017 report
df17_2

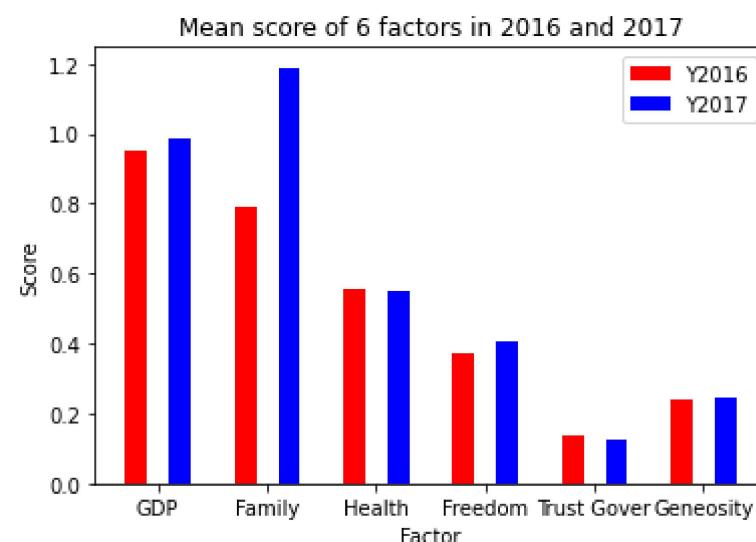
```
Economy..GDP.per.Capita.    0.984718
Family                      1.188898
Health..Life.Expectancy.    0.551341
Freedom                     0.408786
Generosity                  0.246883
Trust..Government.Corruption. 0.123120
dtype: float64
```

2.2 Mean- score of 6 factor in 2016 and 2017

In [106...]: #Chart 3: Mean score of 6 factors in 2016 and 2017

```
import numpy as np
x = np.arange(6)# 6 factors
y1 = [0.953880,0.793621,0.557619,0.370994,0.137624,0.242635]#2016
y2 = [ 0.984718,1.188898,0.551341,0.408786,0.123120,0.246883]#2017
width = 0.2

# plot data in grouped manner of bar type
plt.bar(x-0.2, y1, width, color='red')
plt.bar(x+0.2, y2, width, color='blue')
plt.xticks(x, ['GDP','Family','Health','Freedom','Trust Gover','Geneosity'])
plt.xlabel('Factor')
plt.ylabel("Score")
plt.title('Mean score of 6 factors in 2016 and 2017')
plt.legend(['Y2016','Y2017'])
plt.show()
```



In [105...]: #Mean score of 6 factors in 2018 and 2019

```
df18_19_1=df18_19.groupby('Year').mean()[six_factors18_19]
df18_19_1
```

Out[105...]: GDP per capita Social support Healthy life expectancy Freedom to make life choices Generosity Perceptions of corruption

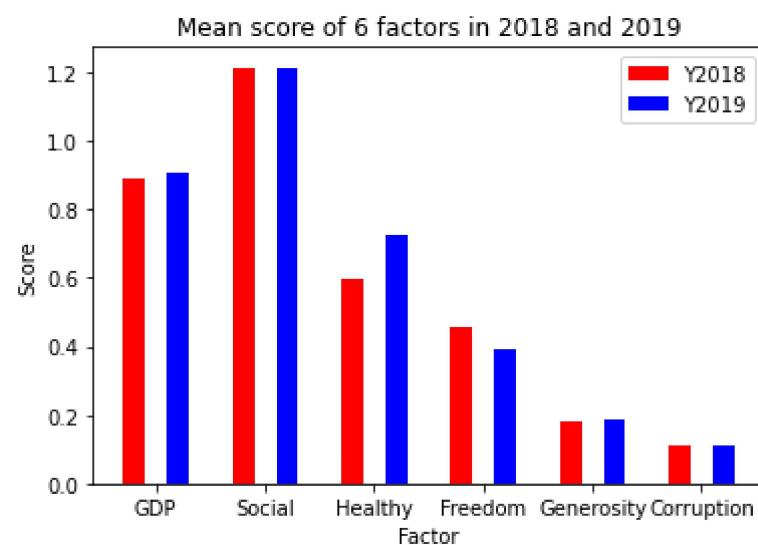
Year	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
2018	0.891449	1.213237	0.597346	0.454506	0.181006	0.112000
2019	0.905147	1.208814	0.725244	0.392571	0.184846	0.110603

2.3 Mean score of 6 factors in 2018 and 2019

In [104...]: #Chart 4: Mean- score of 6 factors in 2018 and 2019

```
import numpy as np
x = np.arange(6)
y1 = [0.891449,1.213237,0.597346,0.454506,0.181006,0.112000]
y2 = [0.905147,1.208814,0.725244,0.392571,0.184846,0.110603]
width = 0.2

# plot data in grouped manner of bar type
plt.bar(x-0.2, y1, width, color='red')
plt.bar(x+0.2, y2, width, color='blue')
plt.xticks(x, ['GDP', 'Social', 'Healthy', 'Freedom', 'Generosity','Corruption'])
plt.xlabel('Factor')
plt.ylabel("Score")
plt.title('Mean score of 6 factors in 2018 and 2019')
plt.legend(['Y2018','Y2019'])
plt.show()
```



Conclusions

- From 2016 and 2019, Denmark, Norway, Finland are happiest countries in top 3. The reason comes down to neighborly support between citizens and state support programs for those in need. People want to feel secure and they also benefit from having a community that they can count on — an environment these countries do better than most in creating.
- In 2016 and 2017 ,there are 6 factors contributing to happiness. They are GDP, family, health, freedom, trust goverment, geneosity. However, in 2018 and 2019, family and trust_goverment factors were replaced social support, Perceptions of corruption respectively.
- Through chart 3 and 4, the healthy score increased, the score GDP decreased slightly. The freedom score maintained stable.