WorldHappinessRecord

March 5, 2024

[1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```python
# 2015 Analysis
```

[3]:
```python
y15 = pd.read_csv("2015.csv")
```

[4]:
```python
y15.head(5)
```

[4]:
```
       Country          Region  Happiness Rank  Happiness Score  \
0  Switzerland  Western Europe               1            7.587
1      Iceland  Western Europe               2            7.561
2      Denmark  Western Europe               3            7.527
3       Norway  Western Europe               4            7.522
4       Canada   North America               5            7.427

   Standard Error  Economy (GDP per Capita)   Family  \
0         0.03411                   1.39651  1.34951
1         0.04884                   1.30232  1.40223
2         0.03328                   1.32548  1.36058
3         0.03880                   1.45900  1.33095
4         0.03553                   1.32629  1.32261

   Health (Life Expectancy)  Freedom  Trust (Government Corruption)  \
0                   0.94143  0.66557                        0.41978
1                   0.94784  0.62877                        0.14145
2                   0.87464  0.64938                        0.48357
3                   0.88521  0.66973                        0.36503
4                   0.90563  0.63297                        0.32957

   Generosity  Dystopia Residual
0     0.29678            2.51738
1     0.43630            2.70201
2     0.34139            2.49204
3     0.34699            2.46531
4     0.45811            2.45176
```

[5]:
```
y15.shape
```

[5]:
```
(158, 12)
```

[6]:
```
y15.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Country                        158 non-null    object
 1   Region                         158 non-null    object
 2   Happiness Rank                 158 non-null    int64
 3   Happiness Score                158 non-null    float64
 4   Standard Error                 158 non-null    float64
 5   Economy (GDP per Capita)       158 non-null    float64
 6   Family                         158 non-null    float64
 7   Health (Life Expectancy)       158 non-null    float64
 8   Freedom                        158 non-null    float64
 9   Trust (Government Corruption)  158 non-null    float64
 10  Generosity                     158 non-null    float64
 11  Dystopia Residual              158 non-null    float64
dtypes: float64(9), int64(1), object(2)
memory usage: 14.9+ KB
```

[7]:
```
y15.isnull().sum()
```

[7]:
```
Country                          0
Region                           0
Happiness Rank                   0
Happiness Score                  0
Standard Error                   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
```

[8]:
```
y15 = y15.drop_duplicates()
```

[9]:
```
y15 = y15.drop( columns= ['Happiness Rank','Standard Error','Dystopia␣
 ↪Residual'])
```

[10]:
```
# 2016 Analysis
```

[11]:

```python
y16 = pd.read_csv("2016.csv")
```

[12]:

```python
y16.head(5)
```

[12]:

```
       Country          Region  Happiness Rank  Happiness Score  \
0      Denmark  Western Europe               1            7.526
1  Switzerland  Western Europe               2            7.509
2      Iceland  Western Europe               3            7.501
3       Norway  Western Europe               4            7.498
4      Finland  Western Europe               5            7.413

   Lower Confidence Interval  Upper Confidence Interval  \
0                      7.460                      7.592
1                      7.428                      7.590
2                      7.333                      7.669
3                      7.421                      7.575
4                      7.351                      7.475

   Economy (GDP per Capita)   Family  Health (Life Expectancy)  Freedom  \
0                   1.44178  1.16374                   0.79504  0.57941
1                   1.52733  1.14524                   0.86303  0.58557
2                   1.42666  1.18326                   0.86733  0.56624
3                   1.57744  1.12690                   0.79579  0.59609
4                   1.40598  1.13464                   0.81091  0.57104

   Trust (Government Corruption)  Generosity  Dystopia Residual
0                        0.44453     0.36171            2.73939
1                        0.41203     0.28083            2.69463
2                        0.14975     0.47678            2.83137
3                        0.35776     0.37895            2.66465
4                        0.41004     0.25492            2.82596
```

[13]:

```python
y16.shape
```

[13]:

```
(157, 13)
```

[14]:

```python
y16.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 157 entries, 0 to 156
Data columns (total 13 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Country                        157 non-null    object
 1   Region                         157 non-null    object
 2   Happiness Rank                 157 non-null    int64
 3   Happiness Score                157 non-null    float64
 4   Lower Confidence Interval      157 non-null    float64
 5   Upper Confidence Interval      157 non-null    float64
 6   Economy (GDP per Capita)       157 non-null    float64
 7   Family                         157 non-null    float64
 8   Health (Life Expectancy)       157 non-null    float64
 9   Freedom                        157 non-null    float64
 10  Trust (Government Corruption)  157 non-null    float64
 11  Generosity                     157 non-null    float64
 12  Dystopia Residual              157 non-null    float64
dtypes: float64(10), int64(1), object(2)
memory usage: 16.1+ KB
```

```
[15]: y16.isnull().sum()
```

```
[15]: 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
```

```
[16]: y16 = y16.drop_duplicates()
```

```
[17]: y16 = y16.drop( columns= ['Happiness Rank','Dystopia Residual','Lower␣
        ↪Confidence Interval', 'Upper Confidence Interval'])
```

```
[18]: # 2017 Analysis
```

```
[19]: y17 = pd.read_csv("2017.csv")
```

```
[20]: y17.head(5)
```

```
[20]:        Country  Happiness.Rank  Happiness.Score  Whisker.high  Whisker.low  \
      0       Norway               1            7.537      7.594445     7.479556
      1      Denmark               2            7.522      7.581728     7.462272
      2      Iceland               3            7.504      7.622030     7.385970
      3  Switzerland               4            7.494      7.561772     7.426227
      4      Finland               5            7.469      7.527542     7.410458

         Economy..GDP.per.Capita.    Family  Health..Life.Expectancy.   Freedom  \
      0                  1.616463  1.533524                  0.796667  0.635423
```

```
1                  1.482383  1.551122                  0.792566  0.626007
2                  1.480633  1.610574                  0.833552  0.627163
3                  1.564980  1.516912                  0.858131  0.620071
4                  1.443572  1.540247                  0.809158  0.617951

   Generosity  Trust..Government.Corruption.  Dystopia.Residual
0    0.362012                       0.315964           2.277027
1    0.355280                       0.400770           2.313707
2    0.475540                       0.153527           2.322715
3    0.290549                       0.367007           2.276716
4    0.245483                       0.382612           2.430182
```

[21]:
```
y17.shape
```

[21]:
```
(155, 12)
```

[22]:
```
y17.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 155 entries, 0 to 154
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Country                        155 non-null    object
 1   Happiness.Rank                 155 non-null    int64
 2   Happiness.Score                155 non-null    float64
 3   Whisker.high                   155 non-null    float64
 4   Whisker.low                    155 non-null    float64
 5   Economy..GDP.per.Capita.       155 non-null    float64
 6   Family                         155 non-null    float64
 7   Health..Life.Expectancy.       155 non-null    float64
 8   Freedom                        155 non-null    float64
 9   Generosity                     155 non-null    float64
 10  Trust..Government.Corruption.  155 non-null    float64
 11  Dystopia.Residual              155 non-null    float64
dtypes: float64(10), int64(1), object(1)
memory usage: 14.7+ KB
```

[23]:
```
y17.isnull().sum()
```

[23]:
```
Country                          0
Happiness.Rank                   0
Happiness.Score                  0
Whisker.high                     0
Whisker.low                      0
Economy..GDP.per.Capita.         0
Family                           0
Health..Life.Expectancy.         0
```

```
Freedom                          0
Generosity                       0
Trust..Government.Corruption.    0
Dystopia.Residual                0
dtype: int64
```

```
[24]: y17 = y17.drop_duplicates()
```

```
[25]: y17 = y17.drop( columns= ['Happiness.Rank','Dystopia.Residual','Whisker.
       ↪high','Whisker.low'])
```

```
[26]: # 2018 Analysis
```

```
[27]: y18 = pd.read_csv("2018.csv")
```

```
[28]: y18.head(5)
```

```
[28]:    Overall rank Country or region  Score  GDP per capita  Social support  \
      0             1           Finland  7.632           1.305           1.592
      1             2            Norway  7.594           1.456           1.582
      2             3           Denmark  7.555           1.351           1.590
      3             4           Iceland  7.495           1.343           1.644
      4             5       Switzerland  7.487           1.420           1.549

         Healthy life expectancy  Freedom to make life choices  Generosity  \
      0                    0.874                         0.681       0.202
      1                    0.861                         0.686       0.286
      2                    0.868                         0.683       0.284
      3                    0.914                         0.677       0.353
      4                    0.927                         0.660       0.256

         Perceptions of corruption
      0                      0.393
      1                      0.340
      2                      0.408
      3                      0.138
      4                      0.357
```

```
[29]: y18.shape
```

```
[29]: (156, 9)
```

```
[30]: y18.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype
```

```
---  ------                        --------------  -----
 0   Overall rank                  156 non-null    int64
 1   Country or region             156 non-null    object
 2   Score                         156 non-null    float64
 3   GDP per capita                156 non-null    float64
 4   Social support                156 non-null    float64
 5   Healthy life expectancy       156 non-null    float64
 6   Freedom to make life choices  156 non-null    float64
 7   Generosity                    156 non-null    float64
 8   Perceptions of corruption     155 non-null    float64
dtypes: float64(7), int64(1), object(1)
memory usage: 11.1+ KB
```

[31]:

```python
y18.isnull().sum()
```

[31]:

```
Overall rank                    0
Country or region               0
Score                           0
GDP per capita                  0
Social support                  0
Healthy life expectancy         0
Freedom to make life choices    0
Generosity                      0
Perceptions of corruption       1
dtype: int64
```

[32]:

```python
m = y18["Perceptions of corruption"].mean()
y18["Perceptions of corruption"] = y18["Perceptions of corruption"].fillna(m)
```

[33]:

```python
y18 = y18.drop_duplicates()
```

[34]:

```python
y18 = y18.drop(columns= ['Overall rank'])
```

[35]:

```python
# 2019 Analysis
```

[36]:

```python
y19 = pd.read_csv("2019.csv")
```

[37]:

```python
y19.head(5)
```

[37]:

```
   Overall rank Country or region  Score  GDP per capita  Social support  \
0             1           Finland  7.769           1.340           1.587
1             2           Denmark  7.600           1.383           1.573
2             3            Norway  7.554           1.488           1.582
3             4           Iceland  7.494           1.380           1.624
4             5       Netherlands  7.488           1.396           1.522

   Healthy life expectancy  Freedom to make life choices  Generosity  \
0                    0.986                         0.596       0.153
```

```
1                    0.996                         0.592       0.252
2                    1.028                         0.603       0.271
3                    1.026                         0.591       0.354
4                    0.999                         0.557       0.322

   Perceptions of corruption
0                      0.393
1                      0.410
2                      0.341
3                      0.118
4                      0.298
```

```
[38]: y19.shape
```

```
[38]: (156, 9)
```

```
[39]: y19.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Overall rank                  156 non-null    int64
 1   Country or region             156 non-null    object
 2   Score                         156 non-null    float64
 3   GDP per capita                156 non-null    float64
 4   Social support                156 non-null    float64
 5   Healthy life expectancy       156 non-null    float64
 6   Freedom to make life choices  156 non-null    float64
 7   Generosity                    156 non-null    float64
 8   Perceptions of corruption     156 non-null    float64
dtypes: float64(7), int64(1), object(1)
memory usage: 11.1+ KB
```

```
[40]: y19.isnull().sum()
```

```
[40]: Overall rank                    0
      Country or region               0
      Score                           0
      GDP per capita                  0
      Social support                  0
      Healthy life expectancy         0
      Freedom to make life choices    0
      Generosity                      0
      Perceptions of corruption       0
      dtype: int64
```

[41]:

```
y19 = y19.drop(columns= ['Overall rank'])
```

[42]:

```
y17.rename(columns={"Happiness.Score" : "Happiness Score", "Economy..GDP.per.
 ↪Capita.":"Economy (GDP per Capita)", "Trust..Government.Corruption.":"Trust␣
 ↪(Government Corruption)",  "Health..Life.Expectancy." : "Health (Life␣
 ↪Expectancy)"}, inplace= True)
```

[43]:

```
y18.rename(columns={"Score":"Happiness Score", "Country or region":
 ↪"Country","Freedom to make life choices":"Freedom", "Healthy life␣
 ↪expectancy":"Health (Life Expectancy)", "GDP per capita":"Economy (GDP per␣
 ↪Capita)", "Social support":"Family", "Perceptions of corruption":"Trust␣
 ↪(Government Corruption)"}, inplace= True)
```

[44]:

```
y19.rename(columns={"Score":"Happiness Score", "Country or region":
 ↪"Country","Freedom to make life choices":"Freedom", "Healthy life␣
 ↪expectancy":"Health (Life Expectancy)", "GDP per capita":"Economy (GDP per␣
 ↪Capita)", "Social support":"Family", "Perceptions of corruption":"Trust␣
 ↪(Government Corruption)"}, inplace= True)
```

[45]:

```
y15["Year"]= "2015"
y16["Year"]= "2016"
y17["Year"]= "2017"
y18["Year"]= "2018"
y19["Year"]= "2019"
```

[46]:

```
y_overall = pd.concat([y15,y16,y17,y18,y19])
y_overall
```

[46]:

```
                      Country          Region  Happiness Score  \
0                 Switzerland  Western Europe            7.587
1                     Iceland  Western Europe            7.561
2                     Denmark  Western Europe            7.527
3                      Norway  Western Europe            7.522
4                      Canada   North America            7.427
..                        …               …              …
151                    Rwanda             NaN            3.334
152                  Tanzania             NaN            3.231
153               Afghanistan             NaN            3.203
154  Central African Republic             NaN            3.083
155               South Sudan             NaN            2.853

     Economy (GDP per Capita)   Family  Health (Life Expectancy)  Freedom  \
0                     1.39651  1.34951                   0.94143  0.66557
1                     1.30232  1.40223                   0.94784  0.62877
2                     1.32548  1.36058                   0.87464  0.64938
3                     1.45900  1.33095                   0.88521  0.66973
4                     1.32629  1.32261                   0.90563  0.63297
```

```
..                        …      …                       …      …
151                   0.35900  0.71100                 0.61400  0.55500
152                   0.47600  0.88500                 0.49900  0.41700
153                   0.35000  0.51700                 0.36100  0.00000
154                   0.02600  0.00000                 0.10500  0.22500
155                   0.30600  0.57500                 0.29500  0.01000

     Trust (Government Corruption)  Generosity  Year
0                          0.41978     0.29678  2015
1                          0.14145     0.43630  2015
2                          0.48357     0.34139  2015
3                          0.36503     0.34699  2015
4                          0.32957     0.45811  2015
..                             …         …   …
151                        0.41100     0.21700  2019
152                        0.14700     0.27600  2019
153                        0.02500     0.15800  2019
154                        0.03500     0.23500  2019
155                        0.09100     0.20200  2019

[782 rows x 10 columns]
```

[47]:
```
y_overall.isna().sum()
```

[47]:
```
Country                            0
Region                           467
Happiness Score                    0
Economy (GDP per Capita)           0
Family                             0
Health (Life Expectancy)           0
Freedom                            0
Trust (Government Corruption)      0
Generosity                         0
Year                               0
dtype: int64
```

[48]:
```
grouped = y_overall.groupby('Country')
y_overall['Region'] = grouped['Region'].ffill( )
y_overall['Region'] = grouped['Region'].bfill( )
```

[49]:
```
missing = y_overall["Region"].isna()
indices = y_overall[missing].index
print(indices)
```

```
Index([32, 70, 37, 57, 38, 63, 83, 119], dtype='int64')
```

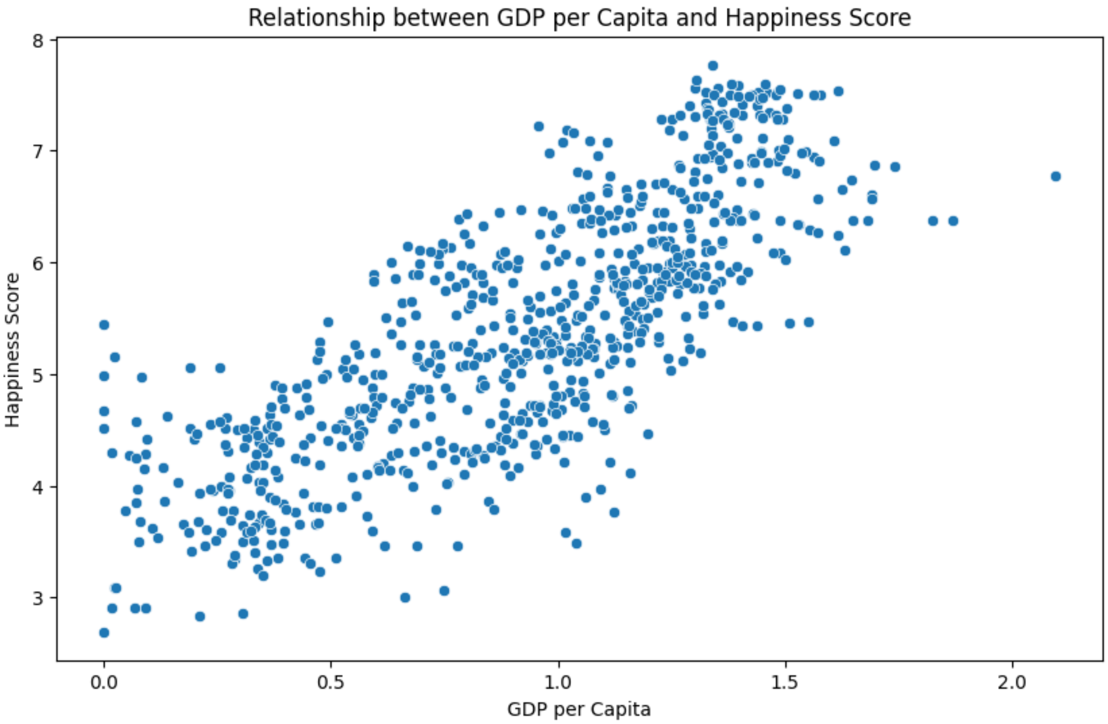
[50]:
```
y_overall.describe()
```

[50]:

```
       Happiness Score  Economy (GDP per Capita)      Family  \
count       782.000000                782.000000  782.000000
mean          5.379018                  0.916047    1.078392
std           1.127456                  0.407340    0.329548
min           2.693000                  0.000000    0.000000
25%           4.509750                  0.606500    0.869363
50%           5.322000                  0.982205    1.124735
75%           6.189500                  1.236187    1.327250
max           7.769000                  2.096000    1.644000

       Health (Life Expectancy)     Freedom  Trust (Government Corruption)  \
count                782.000000  782.000000                     782.000000
mean                   0.612416    0.411091                       0.125418
std                    0.248309    0.152880                       0.105750
min                    0.000000    0.000000                       0.000000
25%                    0.440183    0.309768                       0.054250
50%                    0.647310    0.431000                       0.091033
75%                    0.808000    0.531000                       0.155861
max                    1.141000    0.724000                       0.551910

       Generosity
count  782.000000
mean     0.218576
std      0.122321
min      0.000000
25%      0.130000
50%      0.201982
75%      0.278832
max      0.838075
```

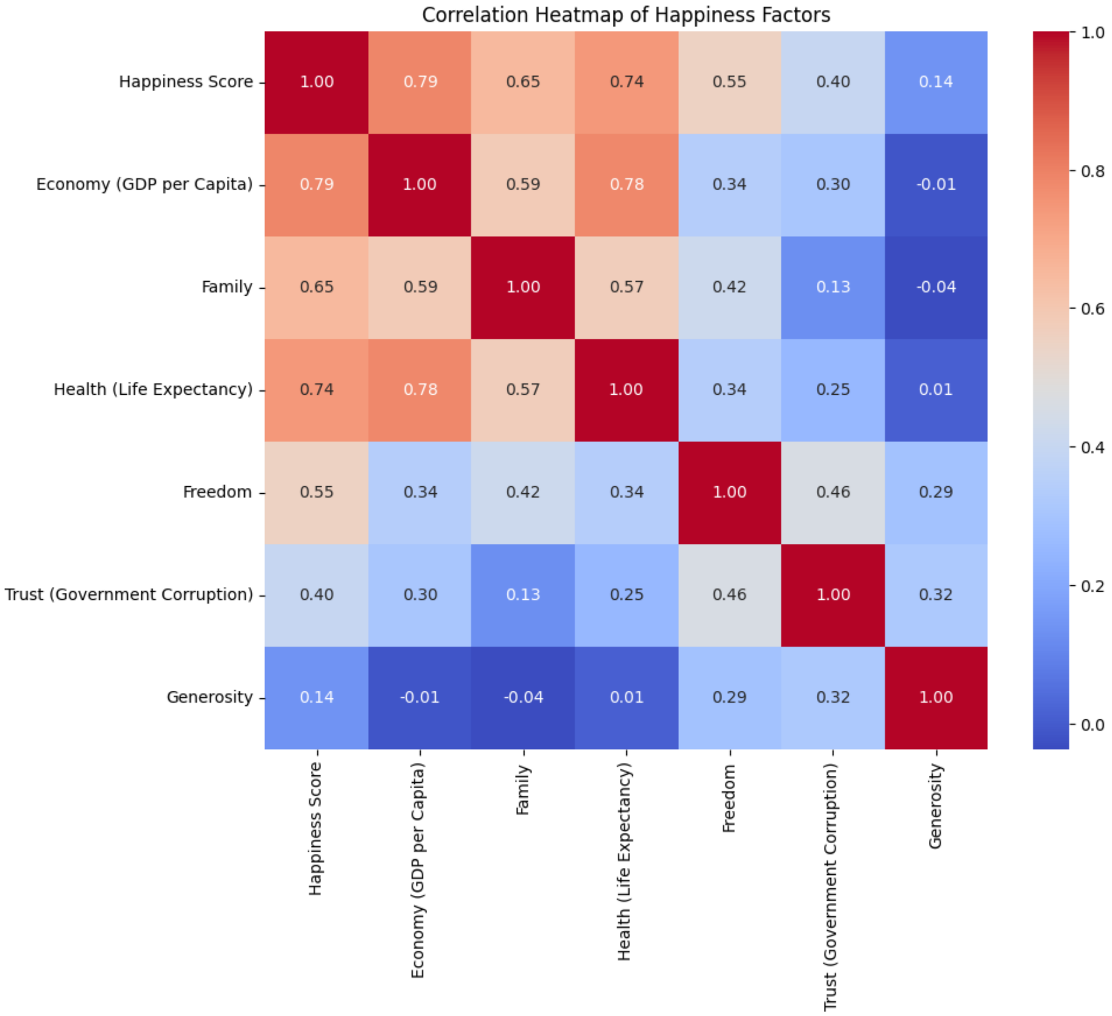
[51]:

```python
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Economy (GDP per Capita)', y='Happiness Score',␣
 ↪data=y_overall)
plt.title("Relationship between GDP per Capita and Happiness Score")
plt.xlabel("GDP per Capita")
plt.ylabel("Happiness Score")
plt.show()
```



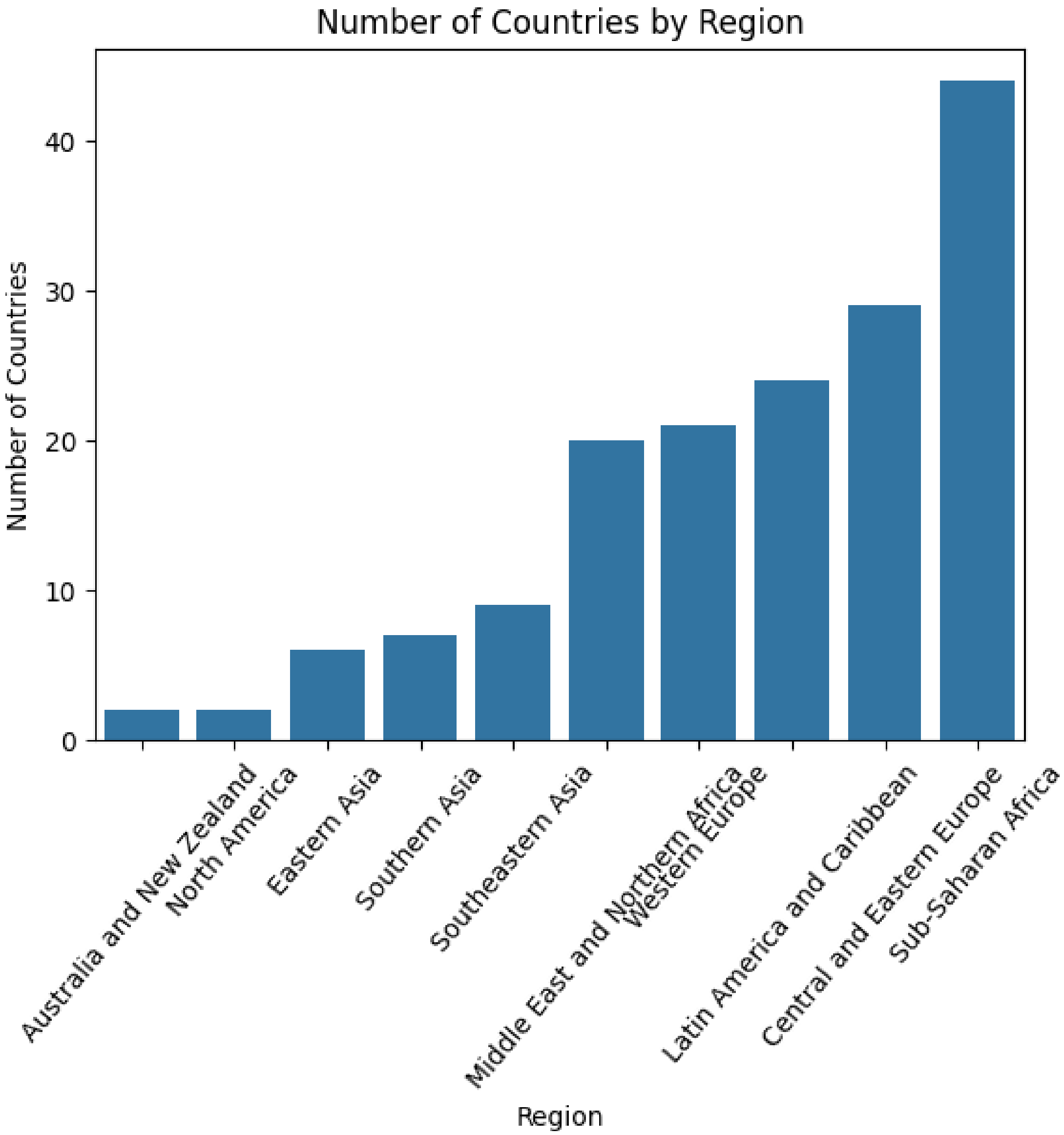
[52]:

```
number_df = y_overall.select_dtypes(include=[np.number])

# Calculating correlations
correlation = number_df.corr()

# Plotting the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap of Happiness Factors")
plt.show()
```



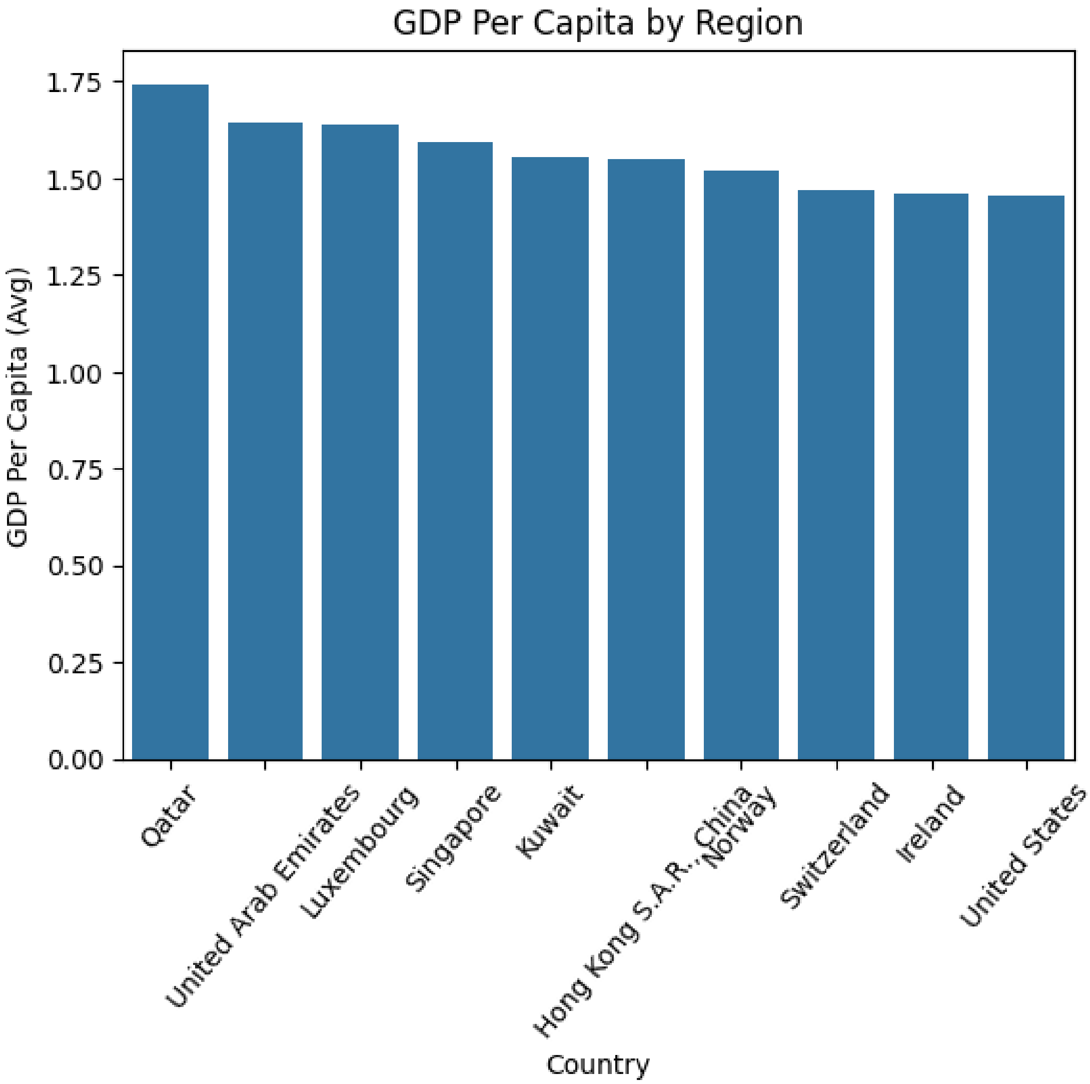
[53]:
```
df_count = y_overall.groupby('Region')['Country'].nunique().
  ↪reset_index(name='num')
df_count = df_count.sort_values( by="num")
sns.barplot(df_count, x='Region', y ='num')
plt.xticks(rotation=50)
plt.xlabel('Region')
plt.ylabel('Number of Countries')
plt.title('Number of Countries by Region')
```

[53]:
```
Text(0.5, 1.0, 'Number of Countries by Region')
```



```
[55]: Avg_gdp = y_overall.groupby('Country')['Economy (GDP per Capita)'].mean()
      Avg_gdp= Avg_gdp.sort_values(ascending=False).iloc[0:10]
      sns.barplot(Avg_gdp)
      plt.title('GDP Per Capita by Region')
      plt.xlabel('Country')
      plt.ylabel('GDP Per Capita (Avg)')
      plt.xticks(rotation= 50)
```

```
[55]: ([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
       [Text(0, 0, 'Qatar'),
        Text(1, 0, 'United Arab Emirates'),
        Text(2, 0, 'Luxembourg'),
        Text(3, 0, 'Singapore'),
        Text(4, 0, 'Kuwait'),
        Text(5, 0, 'Hong Kong S.A.R., China'),
        Text(6, 0, 'Norway'),
        Text(7, 0, 'Switzerland'),
        Text(8, 0, 'Ireland'),
        Text(9, 0, 'United States')])
```



[56]:
```
reg_year = y_overall.pivot_table(index=['Region','Year'],values='Happiness␣
  ↪Score',aggfunc='mean')
reg_year = reg_year.reset_index()
reg_year
```

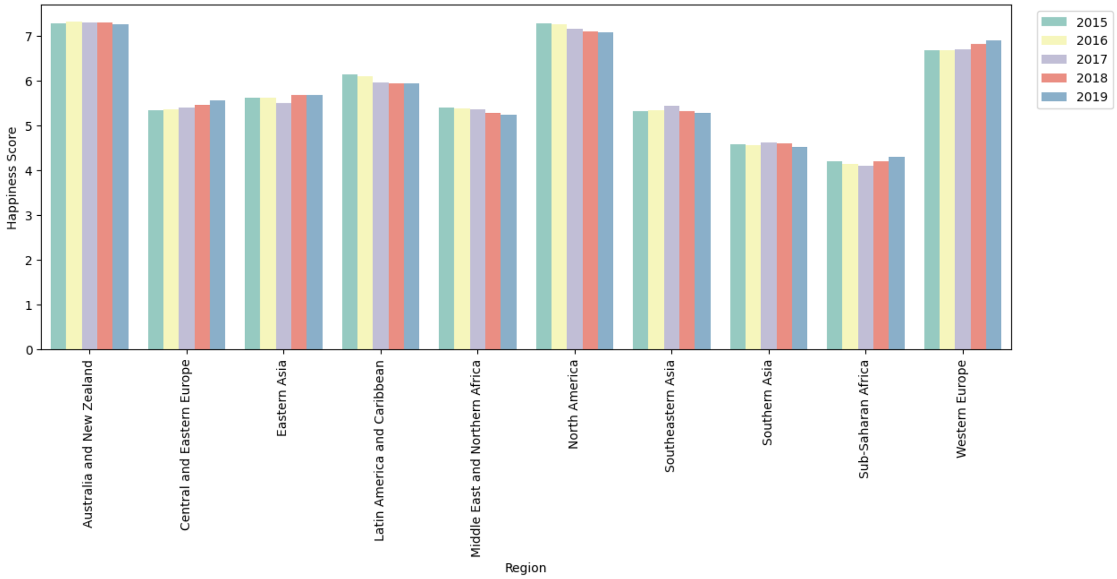
```
[56]:                              Region  Year  Happiness Score
      0         Australia and New Zealand  2015         7.285000
      1         Australia and New Zealand  2016         7.323500
      2         Australia and New Zealand  2017         7.299000
      3         Australia and New Zealand  2018         7.298000
      4         Australia and New Zealand  2019         7.267500
      5        Central and Eastern Europe  2015         5.332931
      6        Central and Eastern Europe  2016         5.370690
      7        Central and Eastern Europe  2017         5.409931
      8        Central and Eastern Europe  2018         5.463966
      9        Central and Eastern Europe  2019         5.571786
      10                     Eastern Asia  2015         5.626167
      11                     Eastern Asia  2016         5.624167
      12                     Eastern Asia  2017         5.496500
      13                     Eastern Asia  2018         5.672000
      14                     Eastern Asia  2019         5.688833
      15      Latin America and Caribbean  2015         6.144682
      16      Latin America and Caribbean  2016         6.101750
      17      Latin America and Caribbean  2017         5.957818
      18      Latin America and Caribbean  2018         5.938619
      19      Latin America and Caribbean  2019         5.942550
      20  Middle East and Northern Africa  2015         5.406900
      21  Middle East and Northern Africa  2016         5.386053
      22  Middle East and Northern Africa  2017         5.369684
      23  Middle East and Northern Africa  2018         5.282737
      24  Middle East and Northern Africa  2019         5.237000
      25                    North America  2015         7.273000
      26                    North America  2016         7.254000
      27                    North America  2017         7.154500
      28                    North America  2018         7.107000
      29                    North America  2019         7.085000
      30                Southeastern Asia  2015         5.317444
      31                Southeastern Asia  2016         5.338889
      32                Southeastern Asia  2017         5.444875
      33                Southeastern Asia  2018         5.313444
      34                Southeastern Asia  2019         5.273667
      35                    Southern Asia  2015         4.580857
      36                    Southern Asia  2016         4.563286
      37                    Southern Asia  2017         4.628429
      38                    Southern Asia  2018         4.603857
      39                    Southern Asia  2019         4.526857
      40               Sub-Saharan Africa  2015         4.202800
      41               Sub-Saharan Africa  2016         4.136421
      42               Sub-Saharan Africa  2017         4.111949
      43               Sub-Saharan Africa  2018         4.195026
      44               Sub-Saharan Africa  2019         4.294513
      45                   Western Europe  2015         6.689619
```

```
46                   Western Europe  2016         6.685667
47                   Western Europe  2017         6.703714
48                   Western Europe  2018         6.829100
49                   Western Europe  2019         6.898400
```

[57]:

```
plt.figure(figsize=(14,5))
g1 = sns.barplot(data=reg_year,x='Region',y='Happiness␣
  ↪Score',hue='Year',palette='Set3')
g1.set_xticklabels(g1.get_xticklabels(),rotation=90)
plt.legend(bbox_to_anchor=(1.02, 1),loc="upper left")
plt.show()
```

```
C:\Users\sapna\AppData\Local\Temp\ipykernel_8584\1993896136.py:3: UserWarning:
set_ticklabels() should only be used with a fixed number of ticks, i.e. after
set_ticks() or using a FixedLocator.
  g1.set_xticklabels(g1.get_xticklabels(),rotation=90)
```



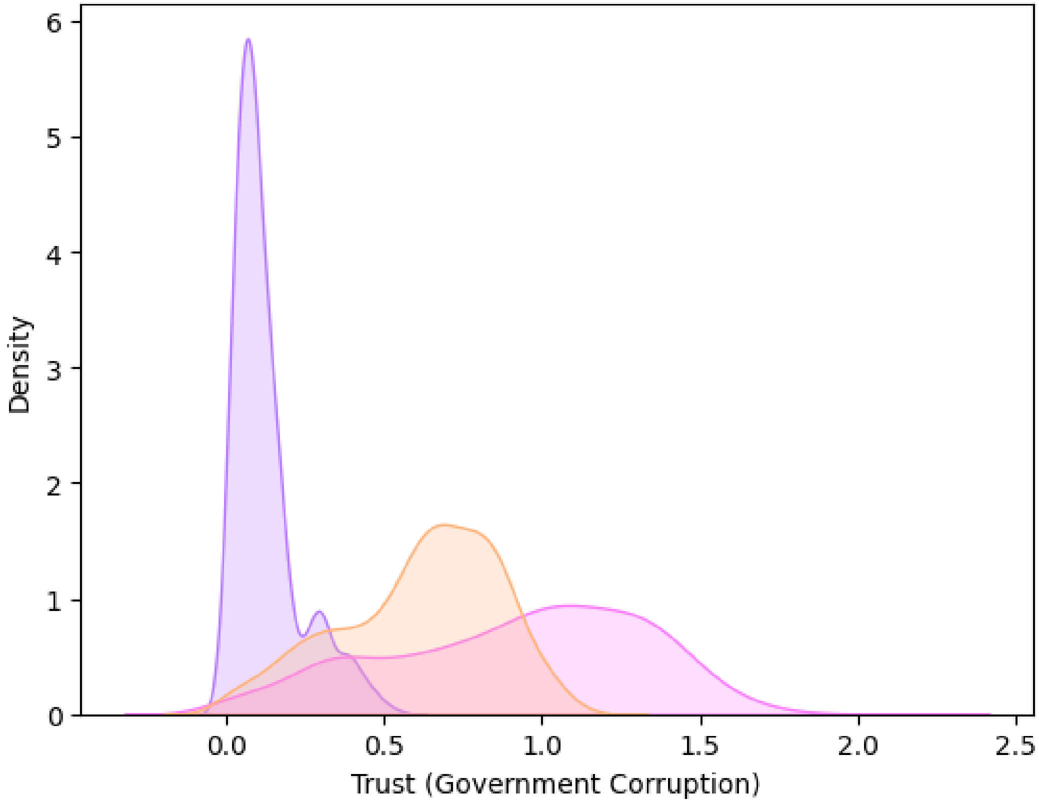
[58]:

```
color=["#B77AFF","#FD7AFF","#FFB27A","#A9FF7A","#7AFFD4","#FF7A7A"]
sns.kdeplot(y_overall['Trust (Government␣
  ↪Corruption)'],shade=True,color=color[0],label='Trust (Government␣
  ↪Corruption)')
sns.kdeplot(y_overall['Economy (GDP per␣
  ↪Capita)'],shade=True,color=color[1],label='Economy (GDP per Capita)')
sns.kdeplot(y_overall['Health (Life␣
  ↪Expectancy)'],shade=True,color=color[2],label='Health (Life Expectancy)')
sns.kdeplot(df['Freedom'],shade=True,color=color[3],label='Freedom')
sns.kdeplot(df['Generosity'],shade=True,color=color[4],label='Generosity')
```

```
plt.xlabel('factores')
plt.legend(fontsize=8)
```

```
C:\Users\sapna\AppData\Local\Temp\ipykernel_8584\1153193609.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(y_overall['Trust (Government
Corruption)'],shade=True,color=color[0],label='Trust (Government Corruption)')
C:\Users\sapna\AppData\Local\Temp\ipykernel_8584\1153193609.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(y_overall['Economy (GDP per
Capita)'],shade=True,color=color[1],label='Economy (GDP per Capita)')
C:\Users\sapna\AppData\Local\Temp\ipykernel_8584\1153193609.py:4: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(y_overall['Health (Life
Expectancy)'],shade=True,color=color[2],label='Health (Life Expectancy)')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[58], line 5
      3 sns.kdeplot(y_overall['Economy (GDP per␣
 ↪Capita)'],shade=True,color=color[1],label='Economy (GDP per Capita)')
      4 sns.kdeplot(y_overall['Health (Life␣
 ↪Expectancy)'],shade=True,color=color[2],label='Health (Life Expectancy)')
----> 5 sns.kdeplot(df['Freedom'],shade=True,color=color[3],label='Freedom')
      6 sns.
 ↪kdeplot(df['Generosity'],shade=True,color=color[4],label='Generosity')
      8 plt.xlabel('factores')

NameError: name 'df' is not defined
```


Density
6
5
4
3
2
1
0
0.0
0.5
1.0
1.5
2.0
2.5
Trust (Government Corruption)

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```