

Exploratory Data Analysis on World Hapiness Report

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

In [2]: sns.set_style('darkgrid')
plt.rcParams['font.size'] = 15
plt.rcParams['figure.figsize'] = (10,7)
plt.rcParams['figure.facecolor'] = '#FFE584'

In [3]: data = pd.read_csv('C:\\Users\\windows 10 pro\\Documents\\archive (27)\\2016.csv')

In [4]: data.head(5)

Out[4]:
```

	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	Dystopia Residual
0	Denmark	Western Europe	1	7.526	7.460	7.592	1.44178	1.16374	0.79504	0.57941	0.44453	0.36171	2.73936
1	Switzerland	Western Europe	2	7.509	7.428	7.590	1.52733	1.14524	0.86303	0.58557	0.41203	0.28083	2.69466
2	Iceland	Western Europe	3	7.501	7.333	7.669	1.42666	1.18326	0.86733	0.56624	0.14975	0.47678	2.83137
3	Norway	Western Europe	4	7.498	7.421	7.575	1.57744	1.12690	0.79579	0.59609	0.35776	0.37895	2.66466
4	Finland	Western Europe	5	7.413	7.351	7.475	1.40598	1.13464	0.81091	0.57104	0.41004	0.25492	2.82596

```

In [5]: data_columns = ['Country', 'Region', 'Happiness Score', 'Economy (GDP per Capita)', 'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)', 'Generosity', 'Dystopia Residual']

In [6]: data = data[data_columns].copy()

In [7]: happy_df = data.rename({'Country':'Country', 'Happiness Score':'Happiness_Score', 'Economy (GDP per Capita)':'GDP_per_Capita', 'Health (Life Expectancy)':'Life_Expectancy', 'Freedom':'Freedom_to_life_choices', 'Trust (Government Corruption)':'Perceptions_of_Corruption', 'Generosity':'Generosity'})

In [8]: happy_df.head()

Out[8]:
```

	Country	Region	Happiness_Score	GDP_per_Capita	Life_Expectancy	Freedom_to_life_choices	Perceptions_of_Corruption	Generosity
0	Denmark	Western Europe	7.526	1.44178	0.79504	0.57941	0.44453	0.36171
1	Switzerland	Western Europe	7.509	1.52733	0.86303	0.58557	0.41203	0.28083
2	Iceland	Western Europe	7.501	1.42666	0.86733	0.56624	0.14975	0.47678
3	Norway	Western Europe	7.498	1.57744	0.79579	0.59609	0.35776	0.37895
4	Finland	Western Europe	7.413	1.40598	0.81091	0.57104	0.41004	0.25492

```

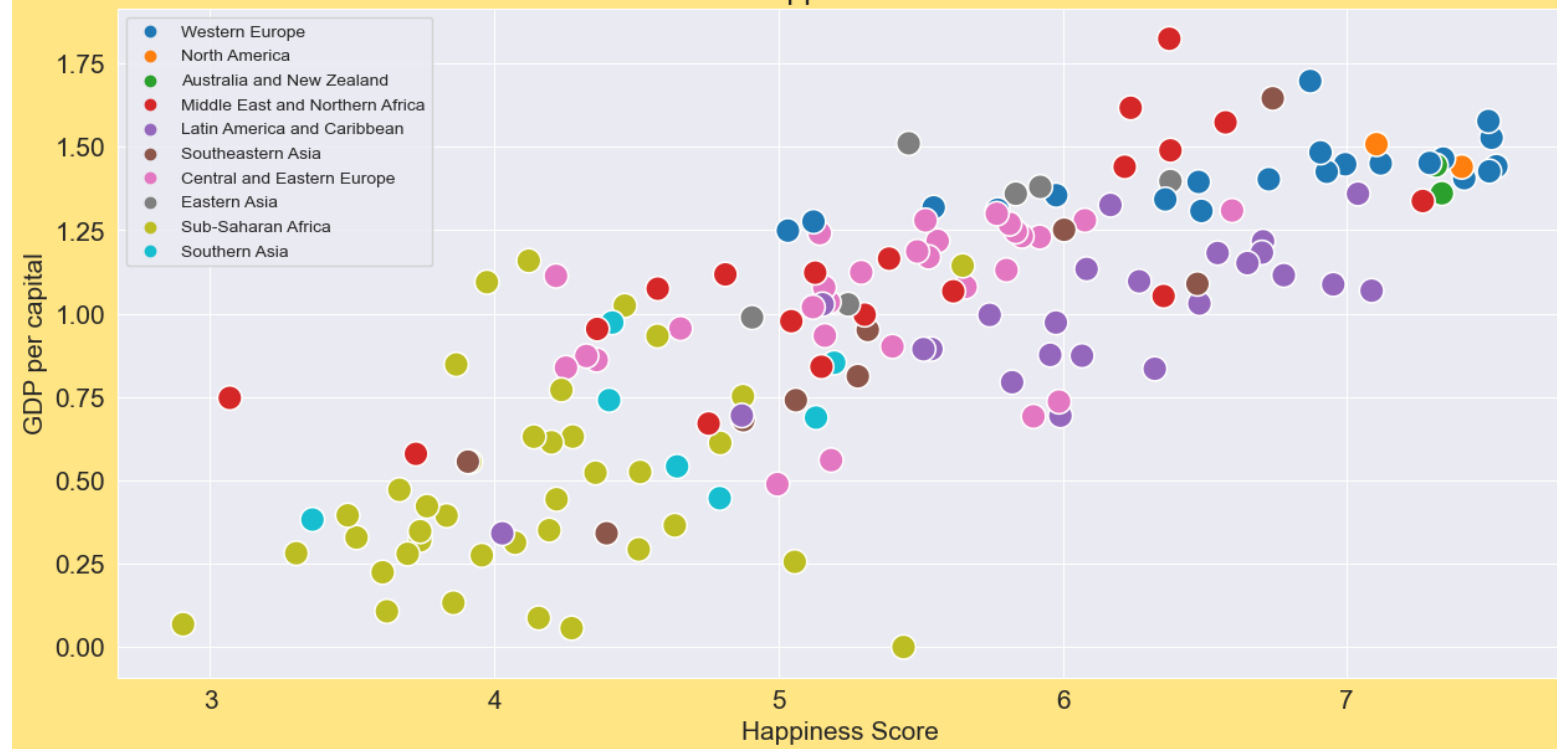
In [9]: happy_df.isnull().sum()

Out[9]:Country      0
Region      0
Happiness_Score      0
GDP_per_Capita      0
Life_Expectancy      0
Freedom_to_life_choices      0
Perceptions_of_Corruption      0
Generosity      0
dtype: int64

In [10]: # Plot between happiness and GDP
plt.rcParams['figure.figsize'] = (15,7)
plt.title('Plot between happiness score and GDP')
sns.scatterplot(x=happy_df.Happiness_Score, y=happy_df.GDP_per_Capita, hue=happy_df.Region, s=200);
plt.legend(loc='upper left', fontsize=10)
plt.xlabel('Happiness Score')
plt.ylabel('GDP per capital')
```

Out[10]:Text(0, 0.5, 'GDP per capital')

Plot between happiness score and GDP



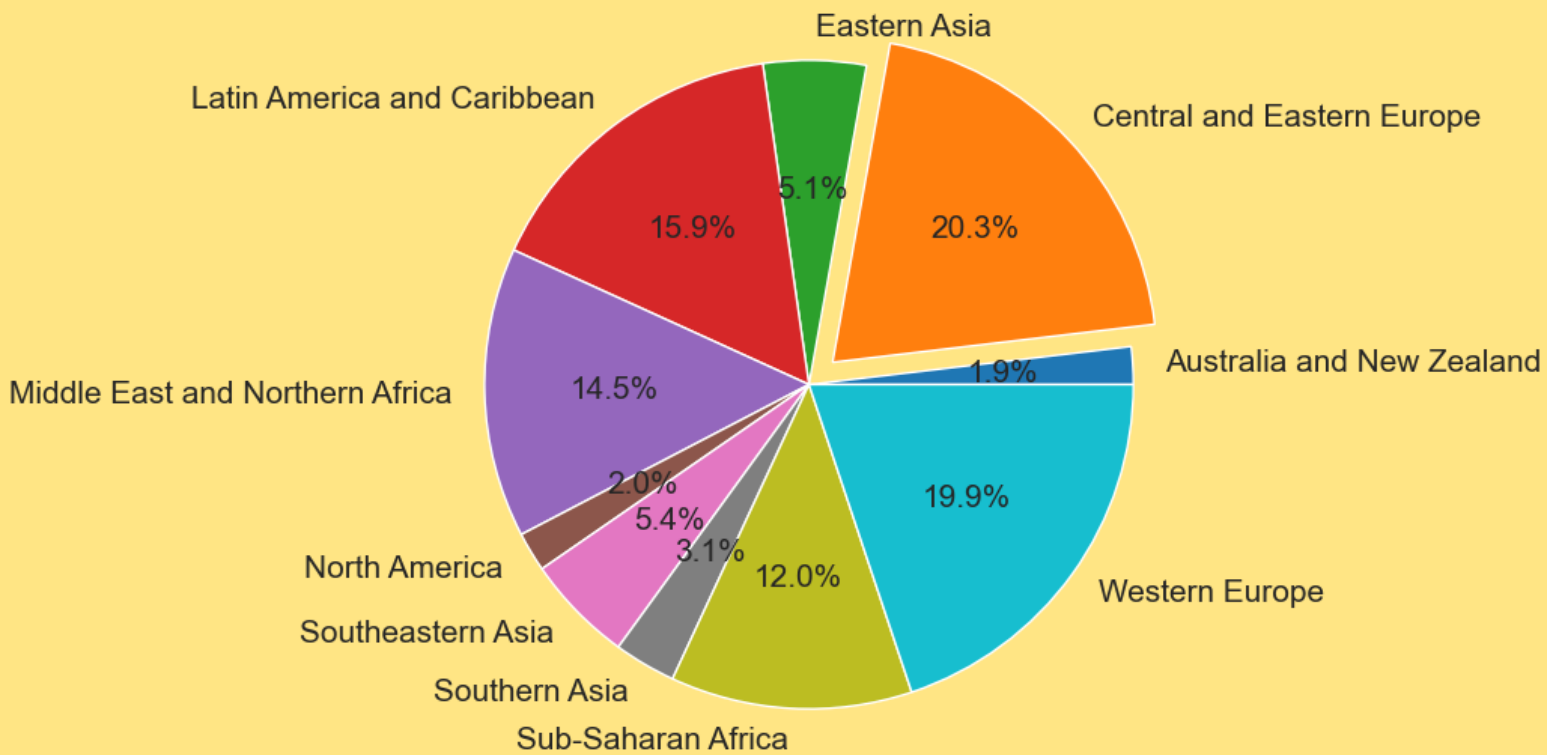
```
In [14]: # Total GDP by regions
gdp_region = happy_df.groupby('Region')['GDP_per_Capita'].sum()
gdp_region
```

```
Out[14]:Region
Australia and New Zealand    2.80509
Central and Eastern Europe   30.37857
Eastern Asia                 7.66387
Latin America and Caribbean  23.84184
Middle East and Northern Africa 21.64714
North America               2.94811
Southeastern Asia           8.06743
Southern Asia               4.62470
Sub-Saharan Africa          18.02421
Western Europe              29.75817
Name: GDP_per_Capita, dtype: float64
```

```
In [15]: explode = (0,0.1,0,0,0,0,0,0,0)
gdp_region.plot.pie(explode=explode, autopct='%1.1f%%')
plt.title('GDP by Region')
plt.ylabel('')
```

Out[15]:Text(0, 0.5, ")

GDP by Region

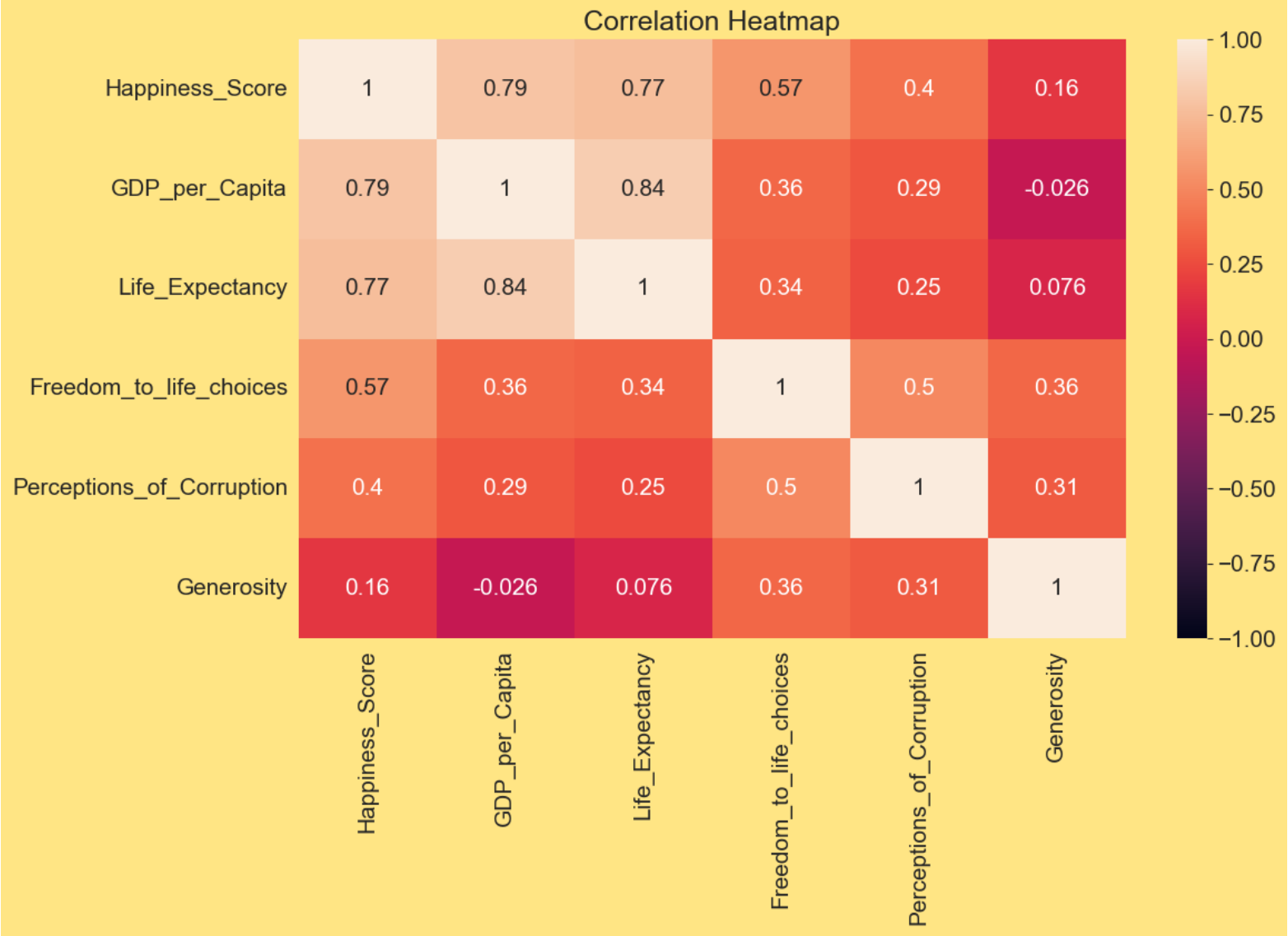


```
In [16]: # Total countries
total_country = happy_df.groupby('Region')[['Country']].count()
total_country
```

```
Out[16]:
```

Region	Country
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

```
In [17]: # Correlation map
plt.figure(figsize=(12,7))
sns.heatmap(happy_df.drop(['Country','Region'],axis=1).corr(), annot = True, vmin = -1, vmax = 1)
plt.title('Correlation Heatmap')
plt.show()
```

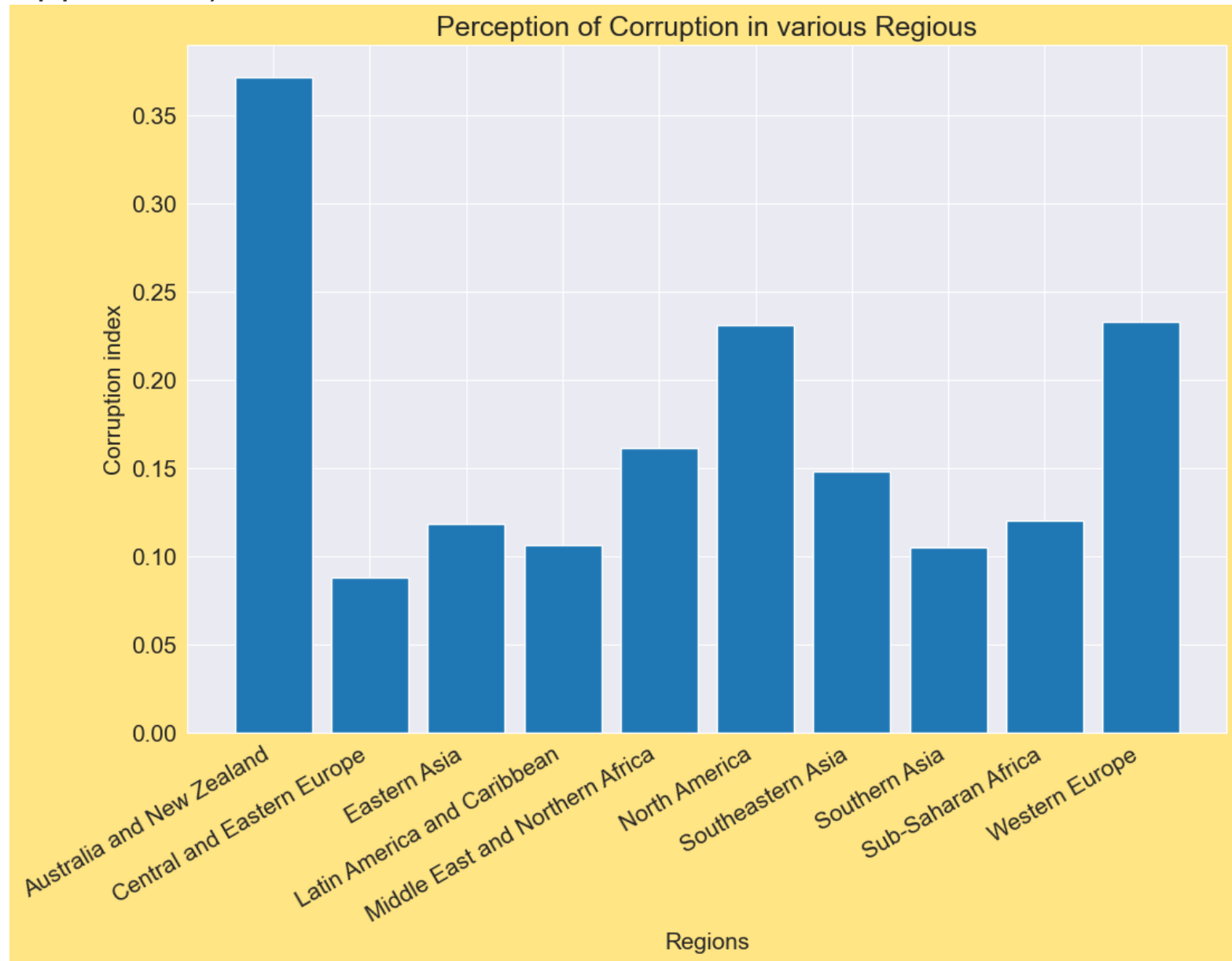


```
In [19]: # Corruption on Regions
corruption = happy_df.groupby('Region')[['Perceptions_of_Corruption']].mean()
corruption
```

Out[19]:

Perceptions_of_Corruption	
Region	
Australia and New Zealand	0.371175
Central and Eastern Europe	0.088072
Eastern Asia	0.118347
Latin America and Caribbean	0.106016
Middle East and Northern Africa	0.161526
North America	0.230985
Southeastern Asia	0.148104
Southern Asia	0.104899
Sub-Saharan Africa	0.120379
Western Europe	0.232544

```
In [24]: # Corruption Regions visualization
plt.rcParams['figure.figsize']=(12,8)
plt.title('Perception of Corruption in various Regions')
plt.xlabel('Regions', fontsize = 15)
plt.ylabel('Corruption index', fontsize = 15)
plt.xticks(rotation = 30, ha = 'right')
plt.bar(corruption.index, corruption.Perceptions_of_Corruption)
```



```
In [25]: top_10 = happy_df.head(10)
bottom_10 = happy_df.tail(10)
```

```
In [31]: # Top 10 country
fig, axes = plt.subplots(1,2, figsize = (16,6))
plt.tight_layout(pad=2)
xlabels = top_10.Country
axes[0].set_title('Top 10 happiest countries life expectancy')
axes[0].set_xticklabels(xlabels,rotation=45,ha='right')
sns.barplot(x = top_10.Country, y=top_10.Life_Expectancy, ax = axes[0])
axes[0].set_xlabel('Country')
axes[0].set_ylabel('Life_Expectancy')

xlabels = bottom_10.Country
axes[1].set_title('Bottom 10 least happy countries life expectancy')
axes[1].set_xticklabels(xlabels,rotation=45,ha='right')
sns.barplot(x = bottom_10.Country, y=bottom_10.Life_Expectancy, ax = axes[1])
axes[1].set_xlabel('Country')
axes[1].set_ylabel('Life_Expectancy')
```

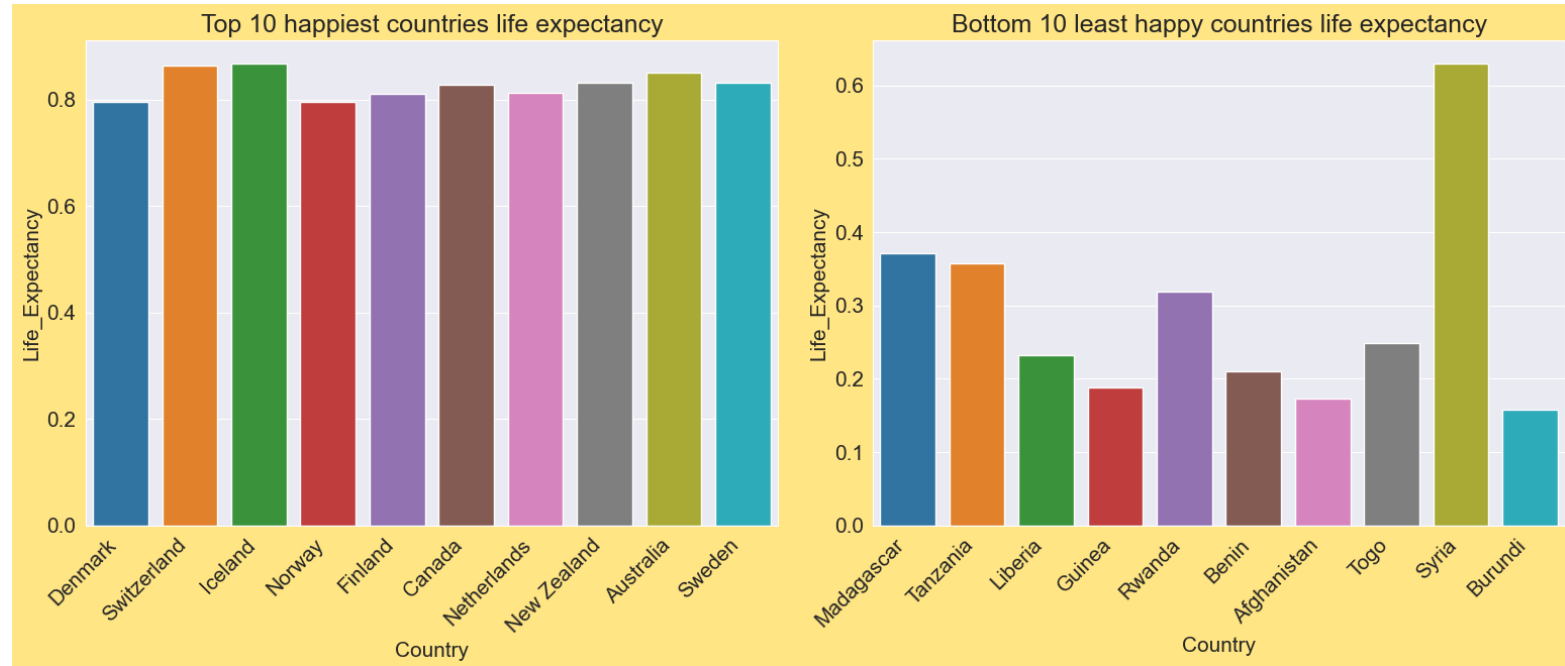
C:\Users\windows 10 pro\AppData\Local\Temp\ipykernel_8828\3278820332.py:6: UserWarning: FixedFormatter should only be used together with FixedLocator

```
axes[0].set_xticklabels(xlabels,rotation=45,ha='right')
```

C:\Users\windows 10 pro\AppData\Local\Temp\ipykernel_8828\3278820332.py:13: UserWarning: FixedFormatter should only be used together with FixedLocator

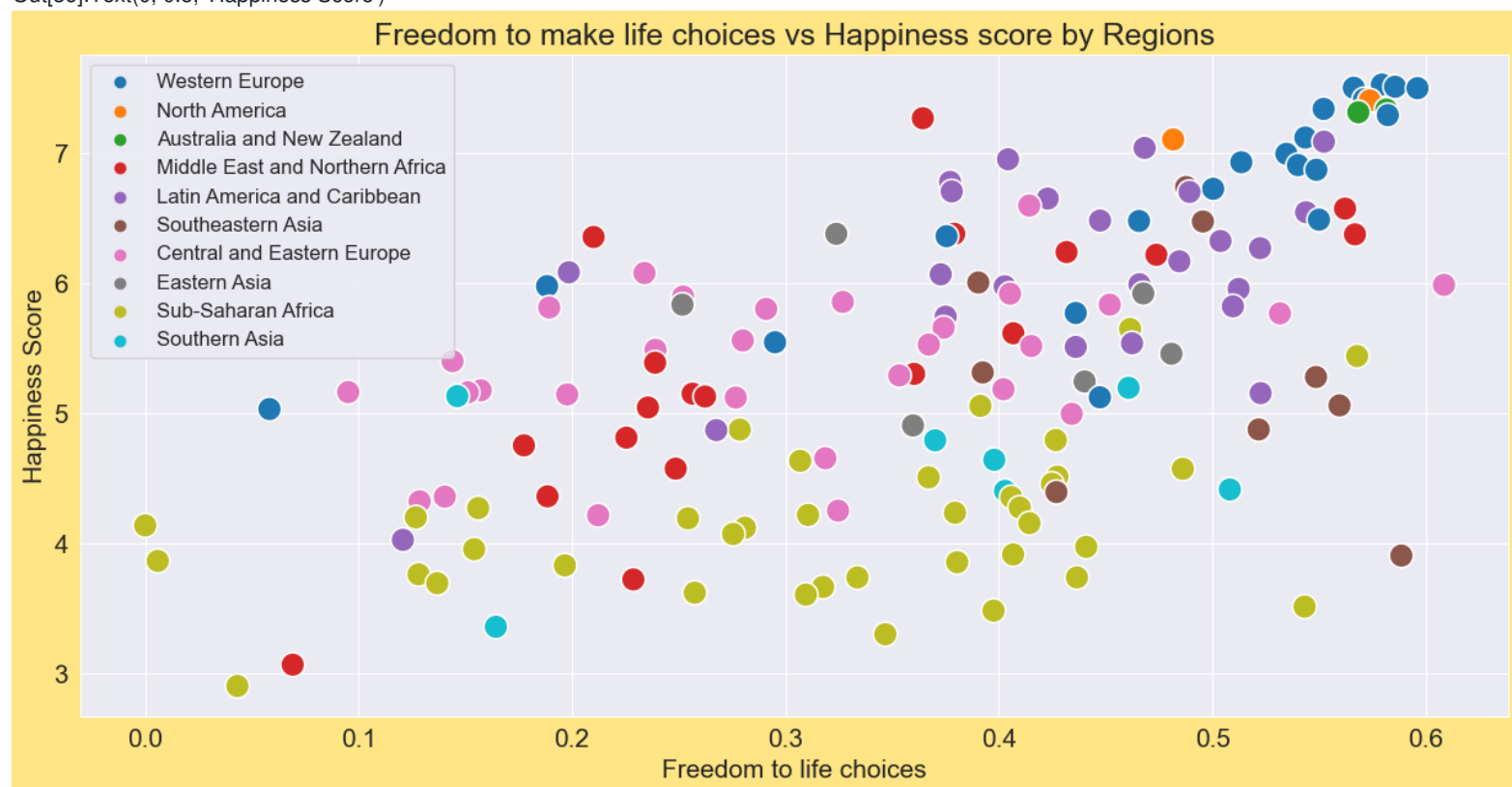
```
axes[1].set_xticklabels(xlabels,rotation=45,ha='right')
```

Out[31]:Text(832.0858585858584, 0.5, 'Life_Expectancy')



In [36]: `plt.rcParams['figure.figsize'] = (15,7)`
`sns.scatterplot(x= happy_df.Freedom_to_life_choices, y= happy_df.Happiness_Score, hue = happy_df.Region, s= 200)`
`plt.legend(loc = 'upper left', fontsize = 12)`
`plt.title('Freedom to make life choices vs Happiness score by Regions')`
`plt.xlabel('Freedom to life choices')`
`plt.ylabel('Happiness Score')`

Out[36]:Text(0, 0.5, 'Happiness Score')



In [45]: `top_10_c = happy_df.sort_values(by = 'Perceptions_of_Corruption', ascending = False).head(10)`
`bottom_10_c = happy_df.sort_values(by = 'Perceptions_of_Corruption', ascending = True).head(10)`

In [47]: `# Top 10 country`
`fig, axes = plt.subplots(1,2, figsize = (16,6))`
`plt.tight_layout(pad=2)`
`xlabels = top_10_c.Country`
`axes[0].set_title('Top 10 countries with most perception of corruption')`
`axes[0].set_xticklabels(xlabels,rotation=45,ha='right')`
`sns.barplot(x = top_10_c.Country, y=top_10_c.Perceptions_of_Corruption, ax = axes[0])`
`axes[0].set_xlabel('Country')`
`axes[0].set_ylabel('Perceptions_of_Corruption')`

```

xlabels = bottom_10_c.Country
axes[1].set_title('Less 10 countries with perception of corruption')
axes[1].set_xticklabels(xlabels,rotation=45,ha='right')
sns.barplot(x = bottom_10_c.Country, y=bottom_10_c.Perceptions_of_Corruption, ax = axes[1])
axes[1].set_xlabel('Country')
axes[1].set_ylabel('Perceptions_of_Corruption')

```

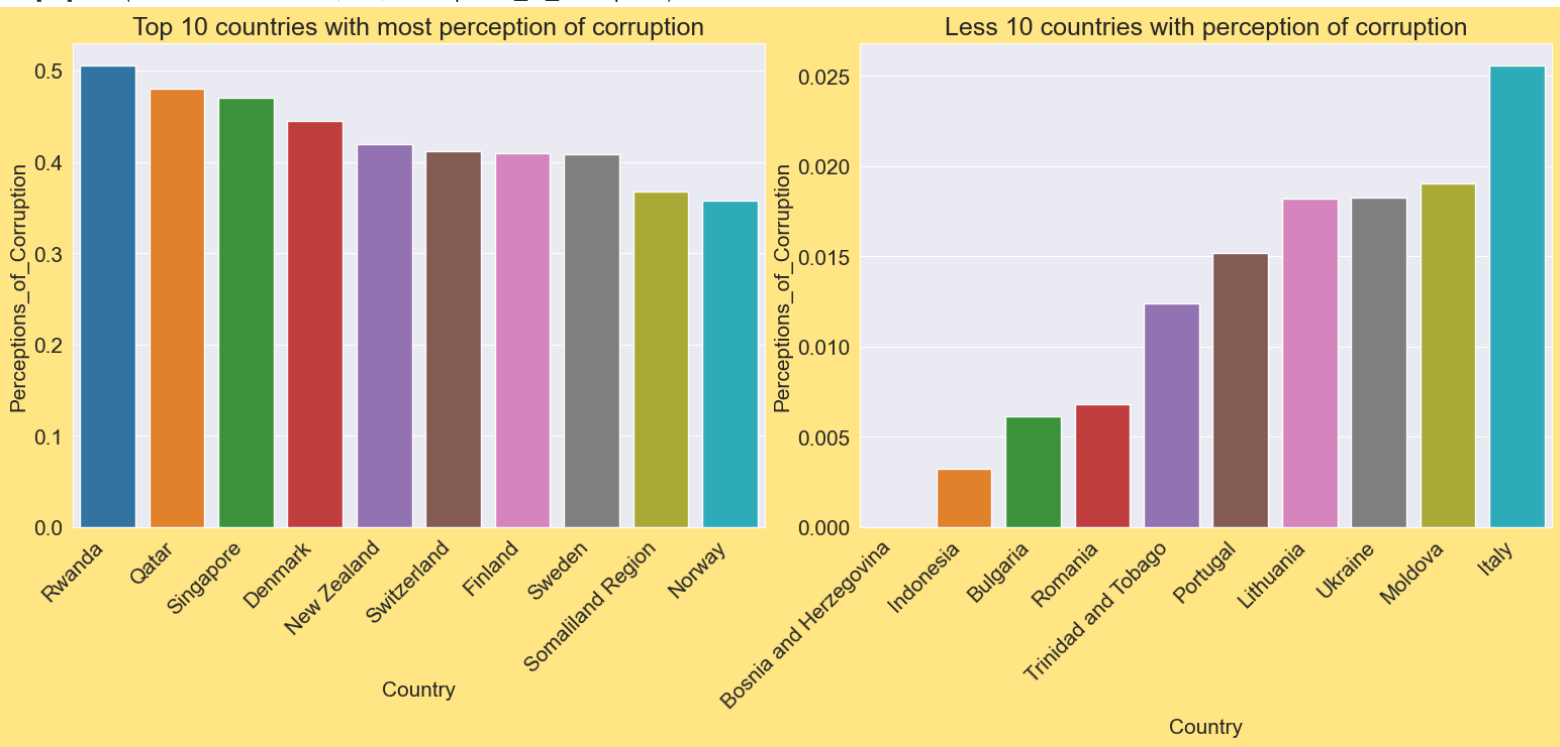
C:\Users\windows 10 pro\AppData\Local\Temp\ipykernel_8828\3433443354.py:6: UserWarning: FixedFormatter should only be used together with FixedLocator

axes[0].set_xticklabels(xlabels,rotation=45,ha='right')

C:\Users\windows 10 pro\AppData\Local\Temp\ipykernel_8828\3433443354.py:13: UserWarning: FixedFormatter should only be used together with FixedLocator

axes[1].set_xticklabels(xlabels,rotation=45,ha='right')

Out[47]:Text(832.0858585858584, 0.5, 'Perceptions_of_Corruption')

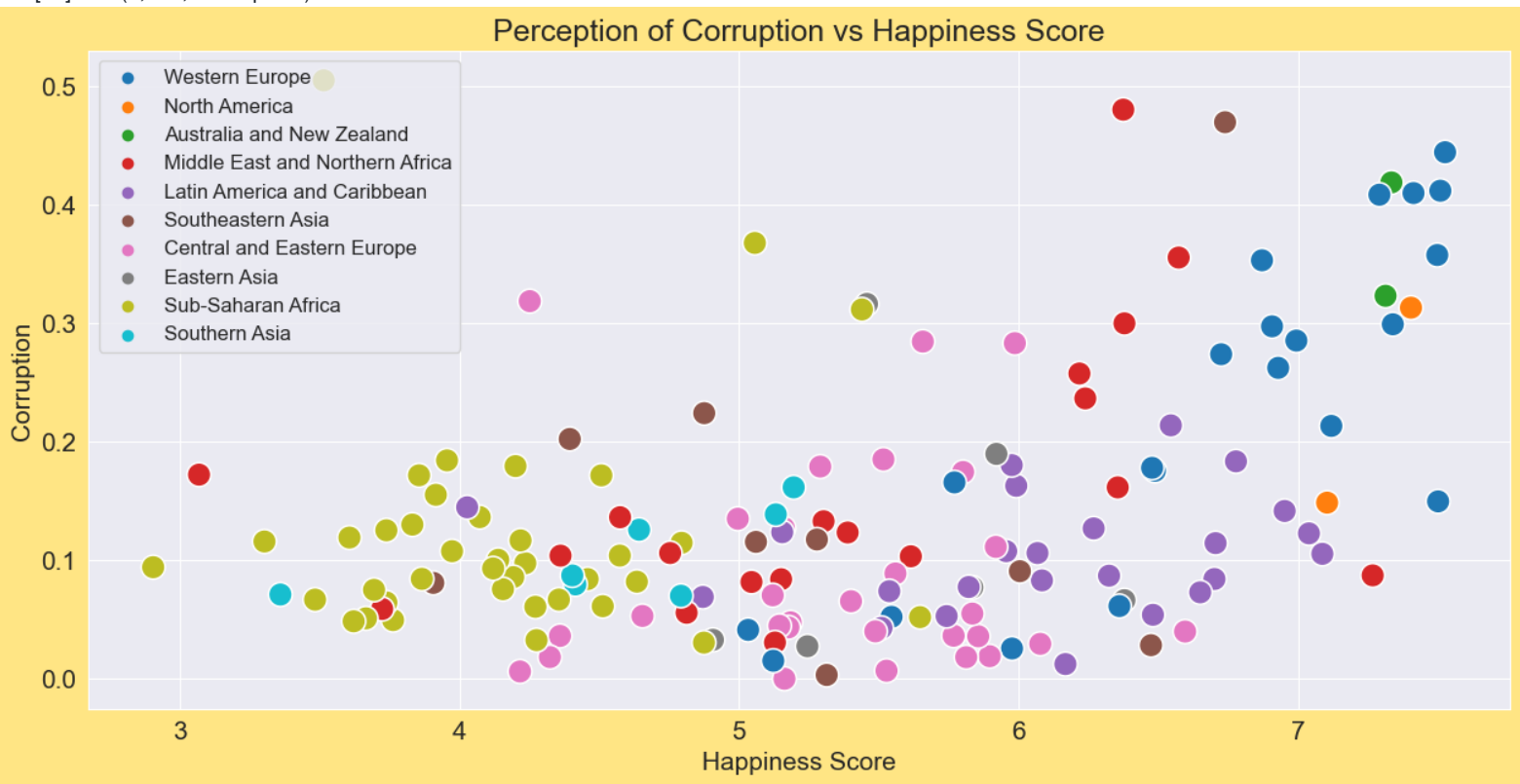


```

In [58]: # Corruption vs Happiness
plt.rcParams['figure.figsize']=(15,7)
sns.scatterplot(x= happy_df.Happiness_Score, y= happy_df.Perceptions_of_Corruption, hue= happy_df.Region, s= 200)
plt.legend(loc= 'upper left', fontsize= 12)
plt.title('Perception of Corruption vs Happiness Score')
plt.xlabel('Happiness Score')
plt.ylabel('Corruption')

```

Out[58]:Text(0, 0.5, 'Corruption')



In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

