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

```
sns.set_style('darkgrid')
plt.rcParams['font.size'] = 15
plt.rcParams['figure.figsize'] = (10,7)
plt.rcParams['figure.facecolor'] = '#FFE5B4'
```

```
data = pd.read_csv('/content/world-happiness-report-2021.csv')
data.head()
```

	Country name	Regional indicator	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Soci suppo
0	Finland	Western Europe	7.842	0.032	7.904	7.780	10.775	0.9
1	Denmark	Western Europe	7.620	0.035	7.687	7.552	10.933	0.9
2	Switzerland	Western Europe	7.571	0.036	7.643	7.500	11.117	0.9
3	Iceland	Western Europe	7.554	0.059	7.670	7.438	10.878	0.9
4	Netherlands	Western Europe	7.464	0.027	7.518	7.410	10.932	0.9

```
data_columns = ['Country name','Regional indicator','Ladder score','Logged GDP per capita','Social support','Healthy lif
```

```
data = data[data_columns].copy()
```

```
happy_df = data.rename({'Country name':'country','Regional indicator':'RI','Ladder score':'Happiness score','Logged GDP
```

```
happy_df = data.rename(columns={'Country name':'country',
                                'Regional indicator':'RI',
                                'Ladder score':'Happiness score',
                                'Logged GDP per capita':'GDP',
                                'Freedom to make life choices':'freedom',
                                'Perceptions of corruption':'perception'})

print(happy_df.columns)
```

```
Index(['country', 'RI', 'Happiness score', 'GDP', 'Social support',
       'Healthy life expectancy', 'freedom', 'Generosity', 'perception'],
      dtype='object')
```

```
happy_df.head(5)
```

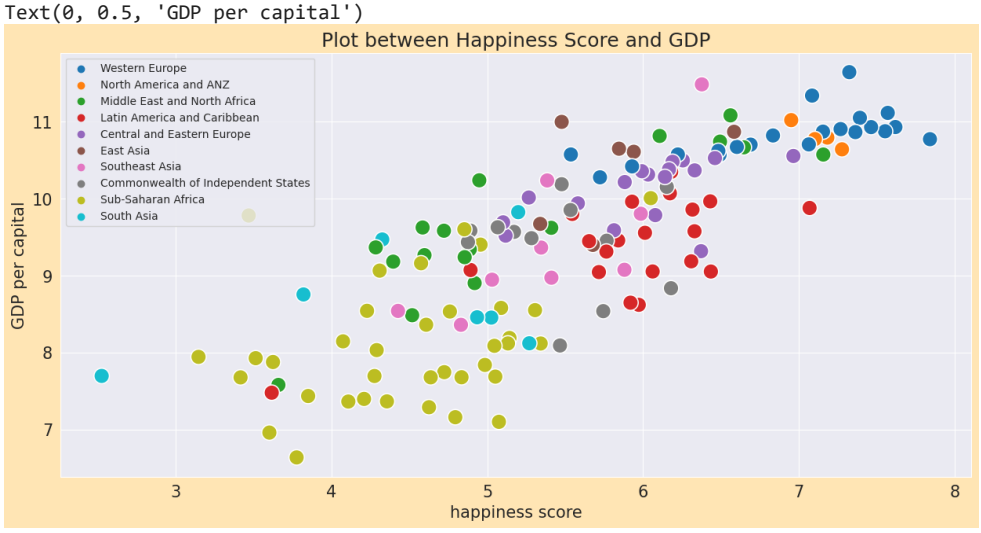
	country	RI	Happiness score	GDP	Social support	Healthy life expectancy	freedom	Generosity	
0	Finland	Western Europe	7.842	10.775	0.954	72.0	0.949	-0.098	
1	Denmark	Western Europe	7.620	10.933	0.954	72.7	0.946	0.030	
2	Switzerland	Western Europe	7.571	11.117	0.942	74.4	0.919	0.025	
3	Iceland	Western Europe	7.554	10.878	0.983	73.0	0.955	0.160	
		Western							

```
happy_df.isnull().sum()
```

```
country                   0
RI                        0
Happiness score           0
GDP                       0
Social support            0
Healthy life expectancy   0
freedom                   0
Generosity                0
perception                0
dtype: int64
```

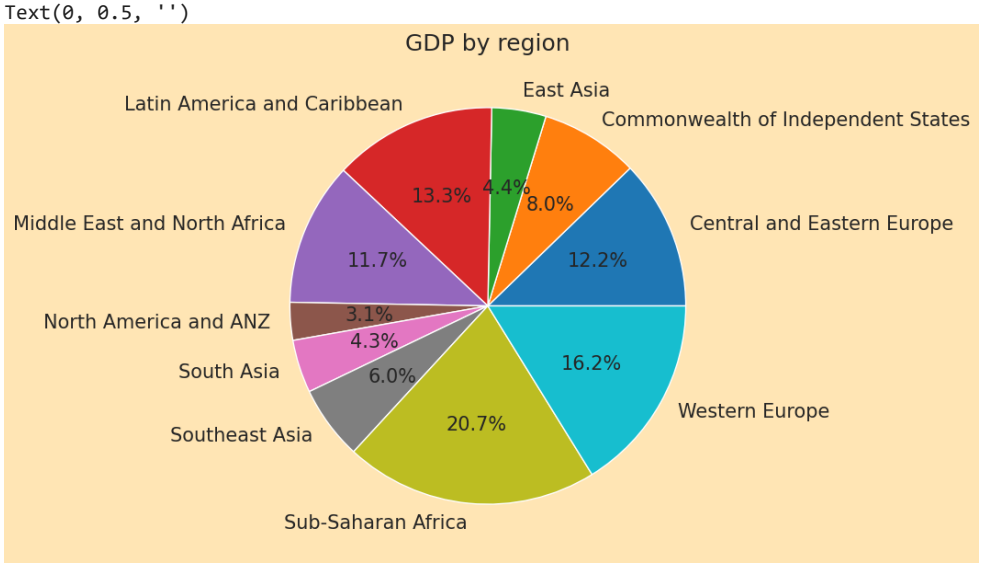
```
from seaborn._core.properties import FontSize
plt.rcParams['figure.figsize'] = (15,7)
plt.title('Plot between Happiness Score and GDP')
sns.scatterplot(x=happy_df['Happiness score'], y=happy_df['GDP'], hue=happy_df['RI'], s=200)
plt.legend(loc = 'upper left', fontsize = '10')
plt.xlabel('happiness score')
plt.ylabel('GDP per capital')
```

```
Text(0, 0.5, 'GDP per capital')
```




```
gdp_region = happy_df.groupby('RI')['GDP'].sum()
gdp_region
gdp_region.plot.pie(autopct = '%1.1f%%')
plt.title('GDP by region')
plt.ylabel('')
```

```
Text(0, 0.5, '')
```

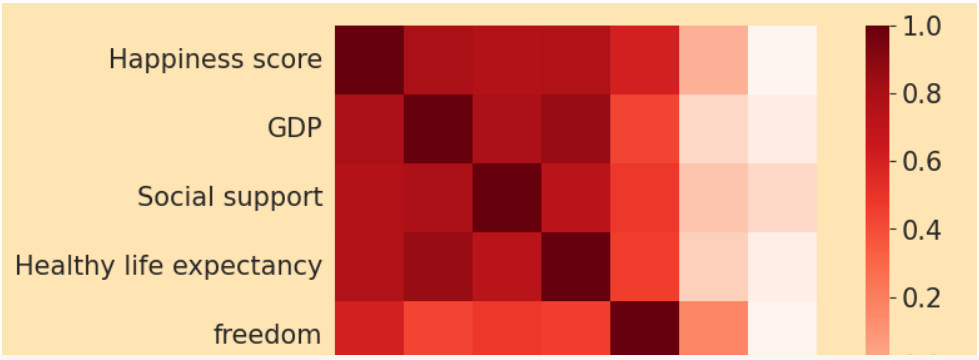



```
total_country = happy_df.groupby('RI')[['country']].count()
print(total_country)
```

```
                                  country
RI
Central and Eastern Europe             17
Commonwealth of Independent States      12
East Asia                               6
Latin America and Caribbean            20
Middle East and North Africa           17
North America and ANZ                   4
South Asia                              7
Southeast Asia                          9
Sub-Saharan Africa                     36
Western Europe                         21
```

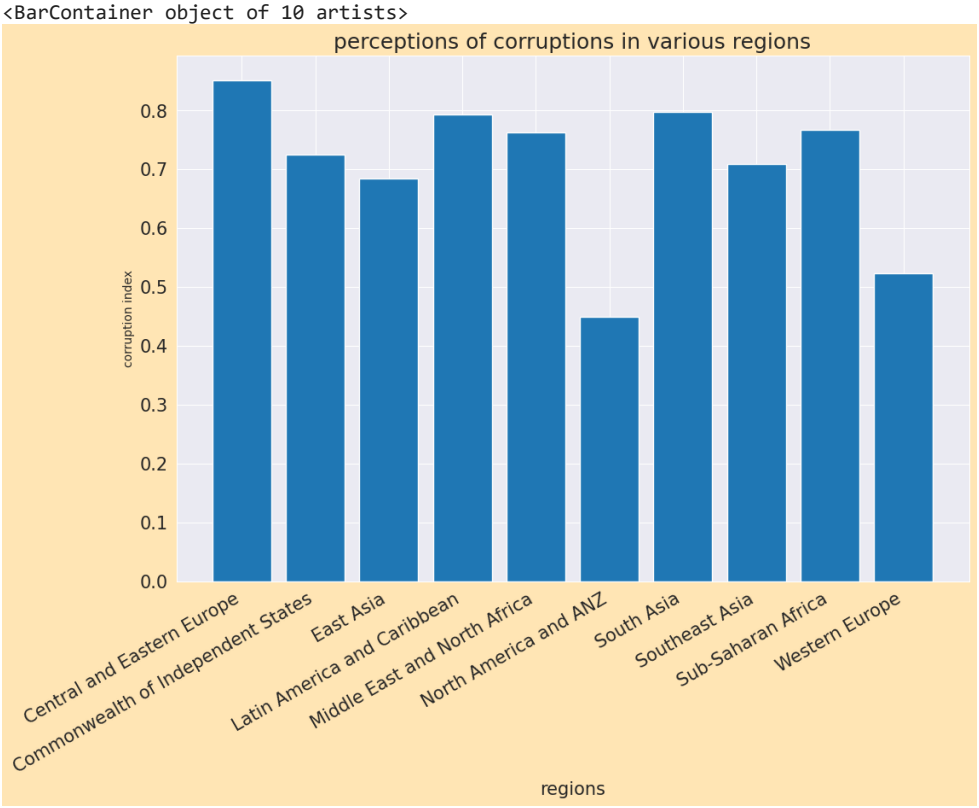
```
cor = happy_df.corr(method = "pearson")
f, ax = plt.subplots(figsize = (10,5))
sns.heatmap(cor, mask = np.zeros_like(cor,dtype=np.bool),cmap = "Reds",square = True,ax = ax)
```

```
<ipython-input-13-d22a37657c58>:1: FutureWarning: The default value of numeric_only
  cor = happy_df.corr(method = "pearson")
<ipython-input-13-d22a37657c58>:3: DeprecationWarning: `np.bool` is a deprecated ali
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  sns.heatmap(cor, mask = np.zeros_like(cor,dtype=np.bool),cmap = "Reds",square = Tr
<Axes: >
```




```
corruption = happy_df.groupby('RI')[['perception']].mean()
corruption
plt.rcParams['figure.figsize']=(12,8)
plt.title("perceptions of corruptions in various regions")
plt.xlabel("regions", fontsize = 15)
plt.ylabel("corruption index", fontsize = 10)
plt.xticks(rotation = 30, ha = 'right')
plt.bar(corruption.index,corruption.perception)
```

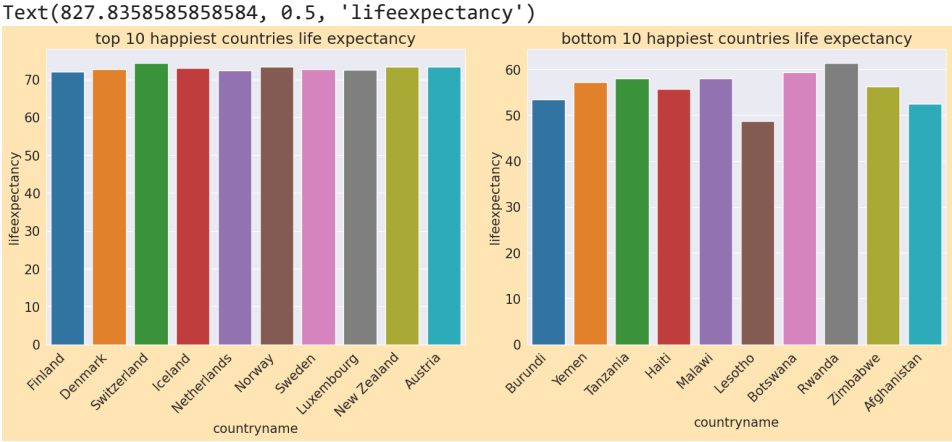
```
<BarContainer object of 10 artists>
```




```
top_10 = happy_df.head(10)
bottom_10 = happy_df.tail(10)
```

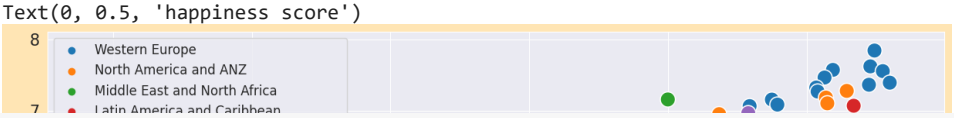
```
fig, axes = plt.subplots(1,2,figsize =(16,6))
plt.tight_layout(pad =2)
xlabels = top_10.country
axes[0].set_title("top 10 happiest countries life expectancy")
axes[0].set_xticklabels(xlabels,rotation = 45,ha = "right")
sns.barplot(x=top_10.country, y=top_10['Healthy life expectancy'], ax=axes[0])
axes[0].set_xlabel('countryname')
axes[0].set_ylabel('lifeexpectancy')
xlabels = bottom_10.country
axes[1].set_title("bottom 10 happiest countries life expectancy")
axes[1].set_xticklabels(xlabels,rotation = 45,ha = "right")
sns.barplot(x = bottom_10.country, y = bottom_10['Healthy life expectancy'], ax=axes[1])
axes[1].set_xlabel('countryname')
axes[1].set_ylabel('lifeexpectancy')
```

```
<ipython-input-16-e4b30b7e1a5c>:5: UserWarning: FixedFormatter should only be used
  axes[0].set_xticklabels(xlabels,rotation = 45,ha = "right")
<ipython-input-16-e4b30b7e1a5c>:11: UserWarning: FixedFormatter should only be use
  axes[1].set_xticklabels(xlabels,rotation = 45,ha = "right")
Text(827.8358585858584, 0.5, 'lifeexpectancy')
```



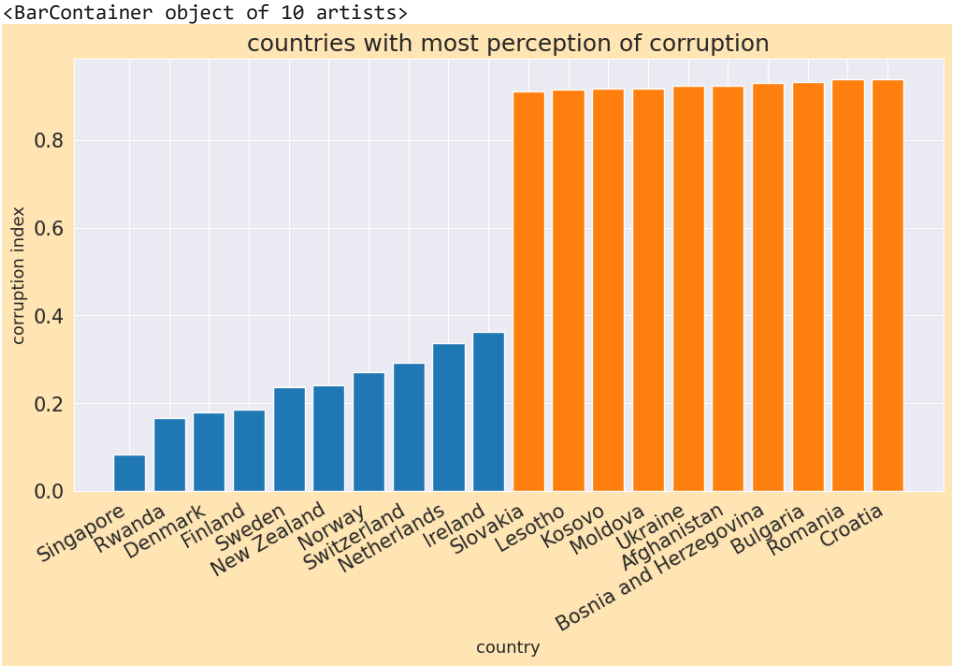

```
plt.rcParams['figure.figsize'] = (15, 7)
sns.scatterplot(x=happy_df['freedom'], y=happy_df['Happiness score'], hue=happy_df['RI'], s=200)
plt.legend(loc='upper left', fontsize=12)
plt.xlabel("freedom to make life choices")
plt.ylabel("happiness score")
```

```
Text(0, 0.5, 'happiness score')
```



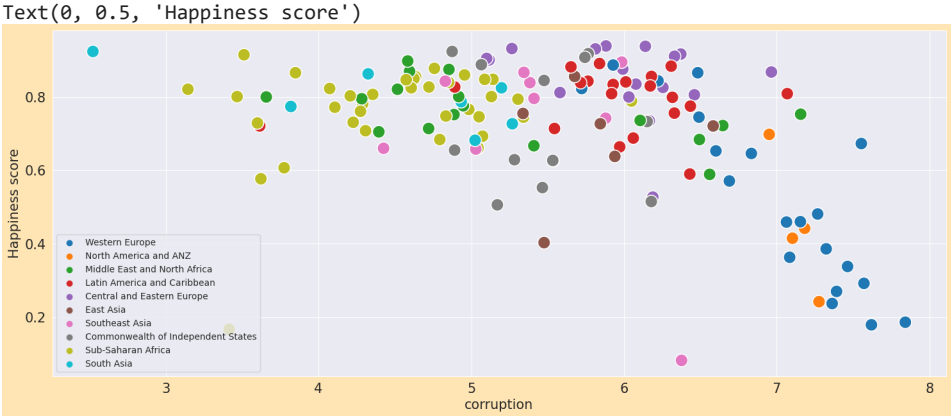

```
country = happy_df.sort_values(by = 'perception').head(10)
plt.rcParams['figure.figsize'] = (12,6)
plt.title('countries with most perception of corruption')
plt.xlabel('country', fontsize = 13)
plt.ylabel('corruption index', fontsize =13)
plt.xticks(rotation = 30, ha = 'right')
plt.bar(country.country, country.perception)
country = happy_df.sort_values(by = 'perception').tail(10)
plt.rcParams['figure.figsize'] = (12,6)
plt.title('countries with most perception of corruption')
plt.xlabel('country', fontsize = 13)
plt.ylabel('corruption index', fontsize =13)
plt.xticks(rotation = 30, ha = 'right')
plt.bar(country.country, country.perception)
```

```
<BarContainer object of 10 artists>
```




```
plt.rcParams['figure.figsize'] = (18,7)
sns.scatterplot(x = happy_df['Happiness score'], y = happy_df['perception'], hue = happy_df['RI'], s =200)
plt.legend(loc='lower left', fontsize=10)
plt.xlabel('corruption')
plt.ylabel('Happiness score')
```

```
Text(0, 0.5, 'Happiness score')
```



Colab paid products - Cancel contracts here

✓ 0s completed at 12:32 PM