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

In [2]:
```
df = pd.read_csv(r'C:\Users\HP\Downloads\world-happiness-report-2021.csv')
df.head()
```

Out[2]:

	Country name	Regional indicator	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support
0	Finland	Western Europe	7.842	0.032	7.904	7.780	10.775	0.954
1	Denmark	Western Europe	7.620	0.035	7.687	7.552	10.933	0.954
2	Switzerland	Western Europe	7.571	0.036	7.643	7.500	11.117	0.942
3	Iceland	Western Europe	7.554	0.059	7.670	7.438	10.878	0.983
4	Netherlands	Western Europe	7.464	0.027	7.518	7.410	10.932	0.942

Here, Ladder score is basically the happiness score, explained by six factors. Dystopia is a hypothetical country that has values equal to the world's lowest national averages for each of the six factors. Now, let's move ahead with analyzing this dataset through Data Visualization using Matplotlib. Line Graphs/Plots

LINE GRAPH

In [6]:
```
#Create Series
expectancy = df['Healthy life expectancy']
score = df['Ladder score']
plt.plot(score, expectancy)
plt.plot(score, expectancy)
plt.title('Happiness Plot')
plt.xlabel('Happiness Score')
plt.ylabel('Age')
plt.legend(['Healthy Life Expectancy'])
plt.show()
```

In [4]:
```
#Add color, style, width to line element
plt.plot(score, expectancy, color = 'green', linestyle = '--',
linewidth=1.2)
plt.title('Happiness Plot')
plt.xlabel('Happiness Score')
plt.ylabel('Age')
plt.legend(['Healthy Life Expectancy'])
plt.show()
```

In [7]:
```
#Add color, style, width to line element
plt.plot(score, expectancy, color = 'red', linestyle = '-',
linewidth=1.2)
plt.title('Happiness Plot')
plt.xlabel('Happiness Score')
plt.ylabel('Age')
plt.legend(['Healthy Life Expectancy'])
plt.show()
```

In [7]:
```
#Add grid using grid() method
plt.grid(True)
plt.plot(score, expectancy)
plt.title('Happiness Plot')
plt.xlabel('Happiness Score')
plt.ylabel('Age')
plt.legend(['Healthy Life Expectancy'])
plt.show()
```

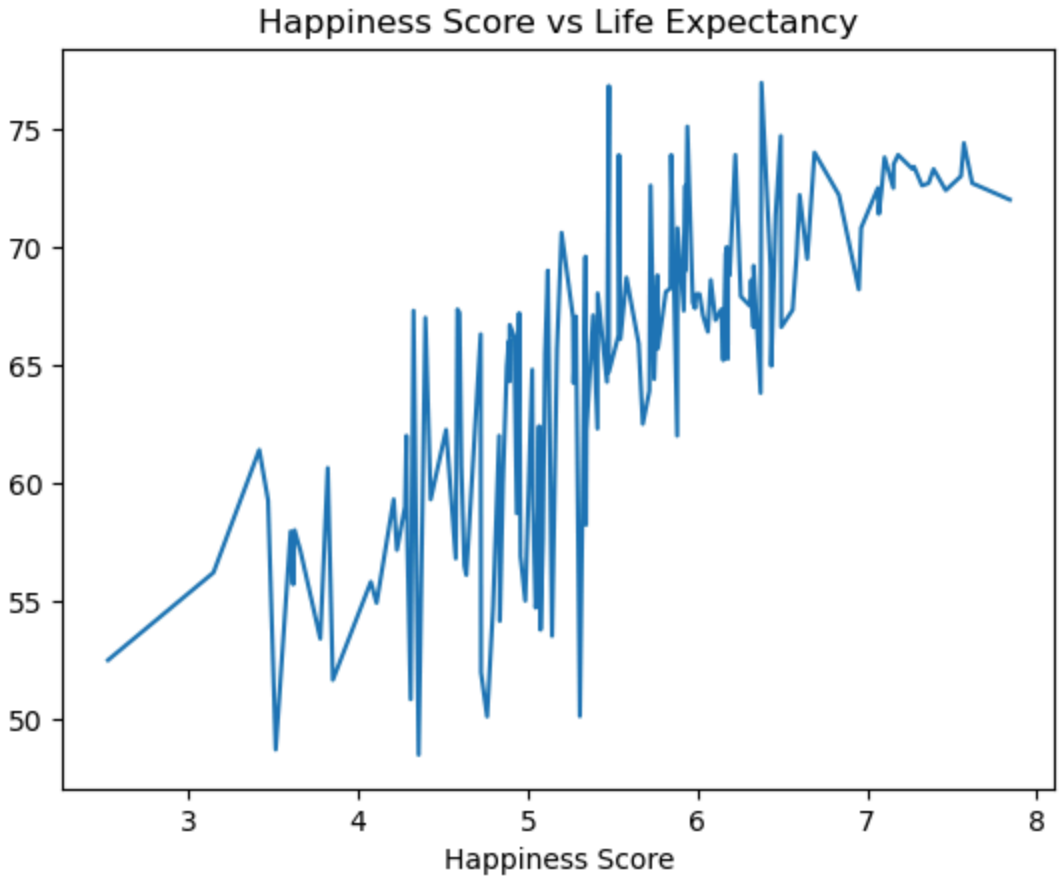
Making Multiple Plots in One Figure

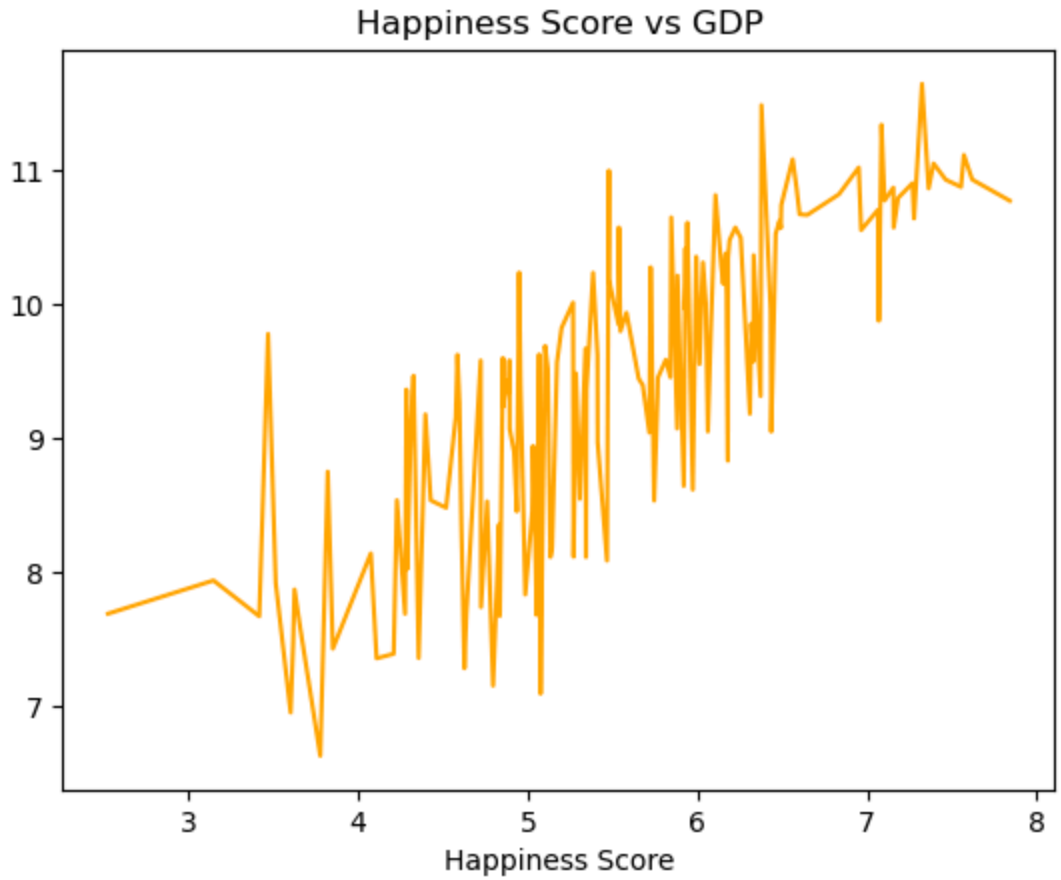
In [ ]:
```
Let's compare the GDP and life expectancy of countries against their happiness scor
comparison, we'll need to plot 'happiness score vs GDP' and 'happiness score vs lif
in a single figure.
```

In [8]:
```
#Create Series for GDP
gdp = df['Logged GDP per capita']
plt.plot(score, expectancy)
plt.plot(score, gdp)
plt.title('Happiness Score vs GDP and Life Expectancy')
plt.xlabel('Happiness Score')
plt.legend(['Life Expectancy','GDP'])
plt.show()
```

In [ ]:
```
From this graph, we can also visually identify a trend – both GDP per capita and li
have higher values than for countries with higher happiness scores.
If you want to display the plots in separate figures, use plt.show() after each plo
shown below:
```

In [10]:
```
plt.plot(score, expectancy)
plt.title('Happiness Score vs Life Expectancy')
plt.xlabel('Happiness Score')
plt.show()
plt.plot(score, gdp, color ='orange')
plt.title('Happiness Score vs GDP')
plt.xlabel('Happiness Score')
plt.show()
```


Happiness Score vs Life Expectancy
75
70
65
60
55
50
3
4
5
6
7
8
Happiness Score


Happiness Score vs GDP
11
10
9
8
7
3
4
5
6
7
8
Happiness Score

In [ ]:
```
Through these separate graphs, we can see that when there is a spike/dip for GDP pe
a given score, there is also a spike/dip for life expectancy for the same score
```

Creating Subplots

In [ ]:
```
We use pyplot.subplots to create a figure and a grid of subplots with a single call
example, for the previous scenario, we could create subplots using the following li
```

In [10]:
```
#Creating two subplots
fig, axs = plt.subplots(2)
fig.suptitle('Vertically stacked subplots')
axs[0].plot(score, expectancy)
axs[1].plot(score, gdp, color = 'orange')
```

Out[10]:
```
[<matplotlib.lines.Line2D at 0x189ea4d55d0>]
```

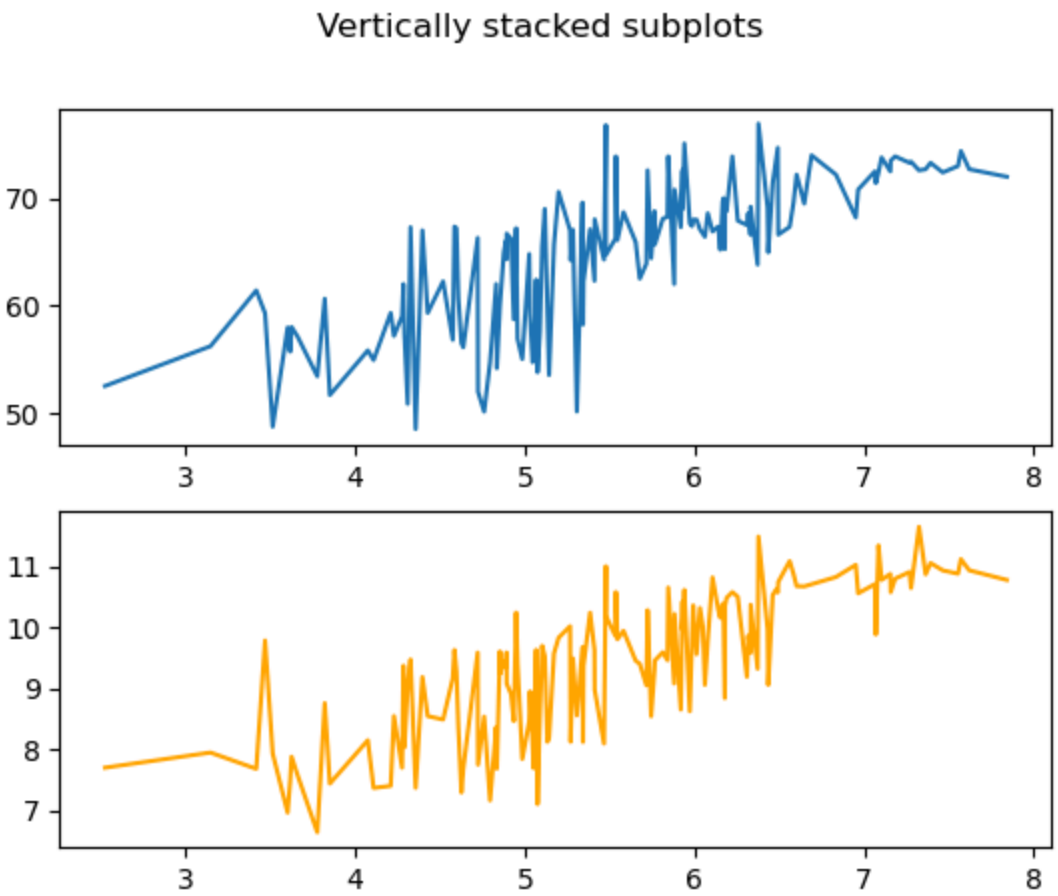


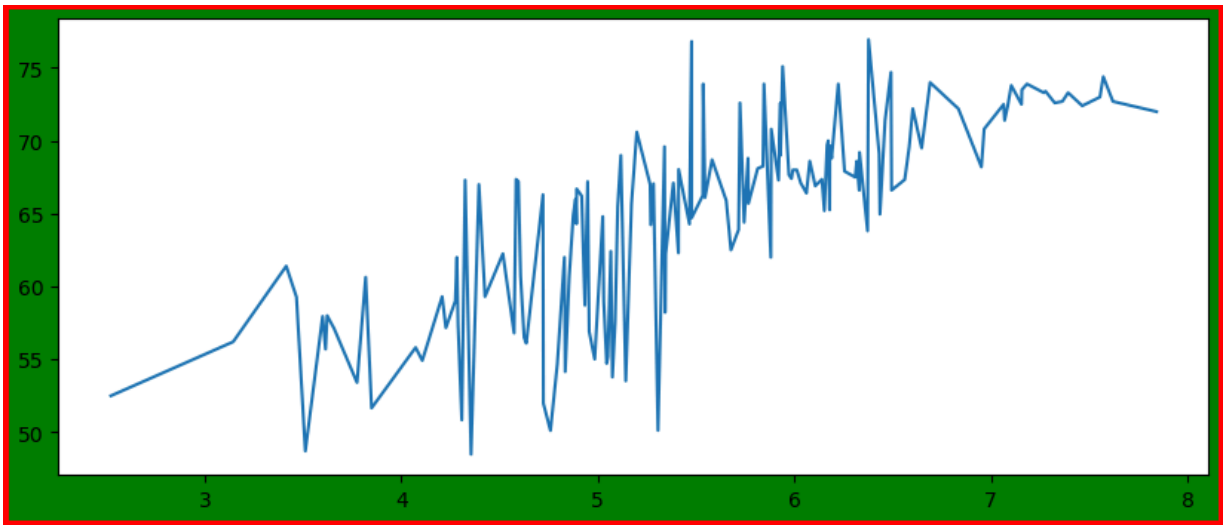
Figure Objects

In [ ]:
```
The matplotlib.figure is a module in Matplotlib that provides the figure object, wh
the plot elements. This module controls the default spacing of the subplots.
matplotlib.figure.Figure() class is the top-level container for the plot elements.
figure instances.
plt.figure() is used to create the empty figure object in Matplotlib. It has the fo
parameters:
```

- figsize: Figure dimension (width, height) in inches
- dpi: Dots per inch
- facecolor: Figure patch facecolor
- edgecolor: Figure patch edge color
- linewidth: Linewidth of the frame

In [11]:
```
#Creating a figure object fig
fig=plt.figure(figsize=(10,4), facecolor ='green',
edgecolor='r',linewidth=5)
plt.plot(score, expectancy)
```

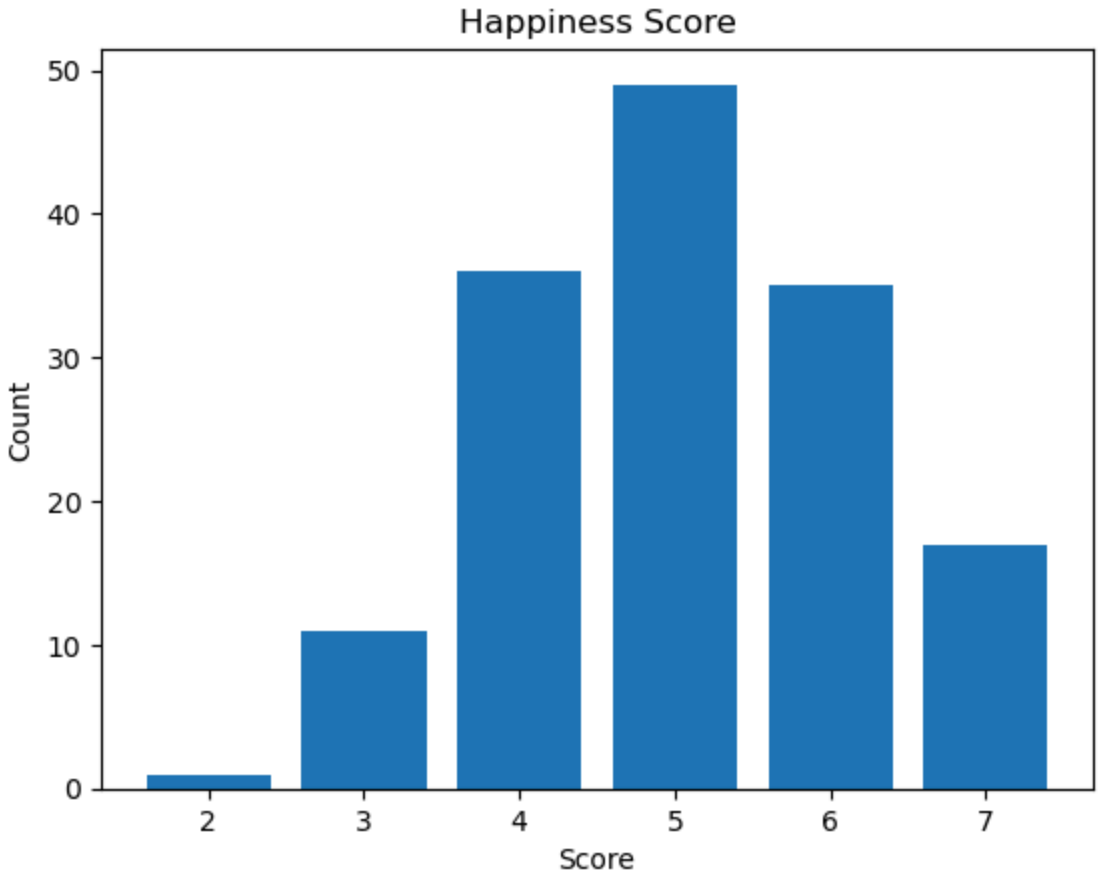
Out[11]: [<matplotlib.lines.Line2D at 0x189ea594790>]



Bar Graphs/Plots

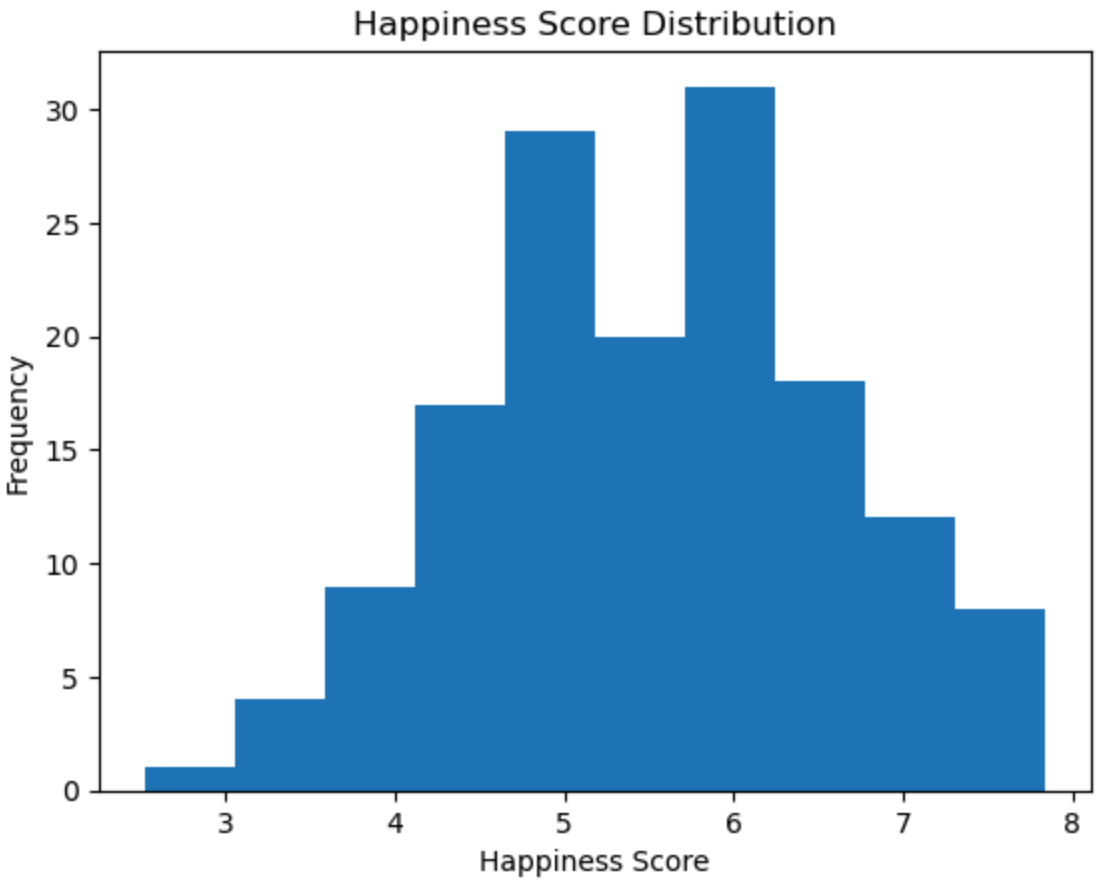
In [12]:
```
#Converting to int
HappinessScore = score.apply(int)
#Counting the number of times each score occurs - the height of the bars

count = HappinessScore.value_counts()
#Score of each count - X-axis
HapScore = count.index
#Plotting the bar graph
plt.bar(HapScore, count)
plt.title('Happiness Score')
plt.xlabel('Score')
plt.ylabel('Count')
plt.show()
```



Histograms

In [13]:
```
plt.hist(score)
plt.title('Happiness Score Distribution')
plt.xlabel('Happiness Score')
plt.ylabel('Frequency')
plt.show()
```



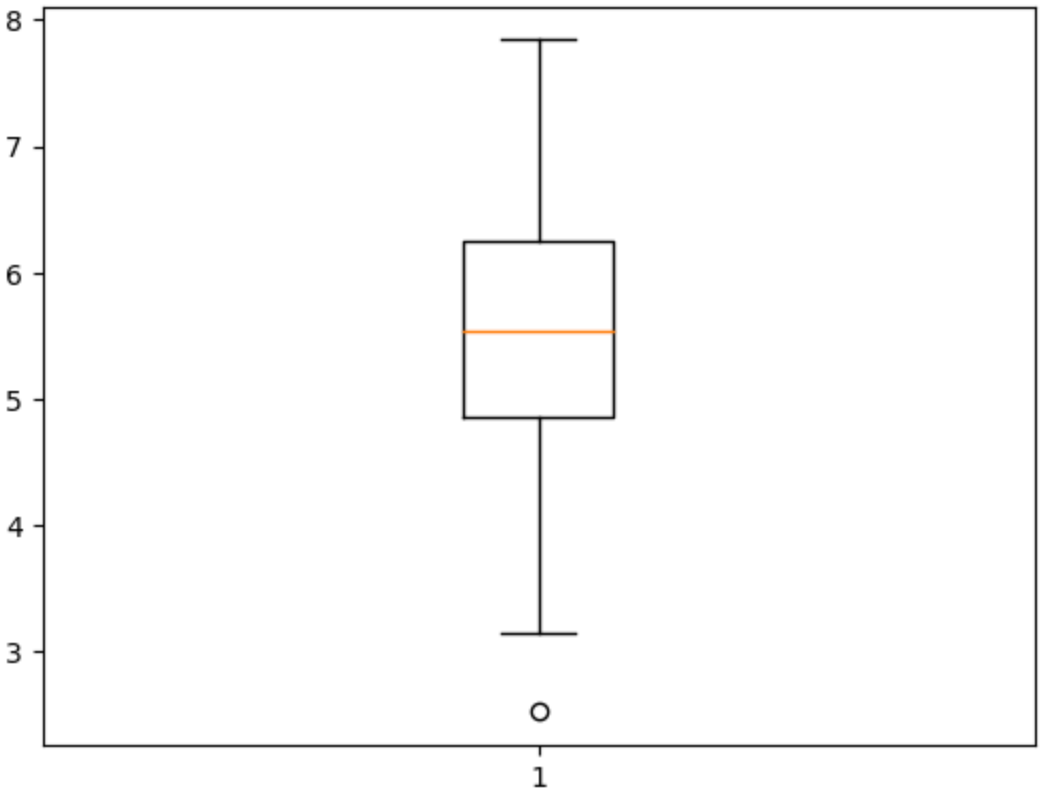
In [ ]:
```
Technically, the happiness score takes a continuous range of values, so we can get
of the score distribution through the above histogram. Though we can't get exact da
just by looking at them.
```

Boxplots

In [ ]:
```
A boxplot is used to display data distribution through quartiles. Quartiles (Q1, Q2
basically the division of data into four equal groups or intervals. A median separa
half and upper half of the data.
The function used for the scatter plot is boxplot(). Used to detect outliers in dat
the data is grouped.
```

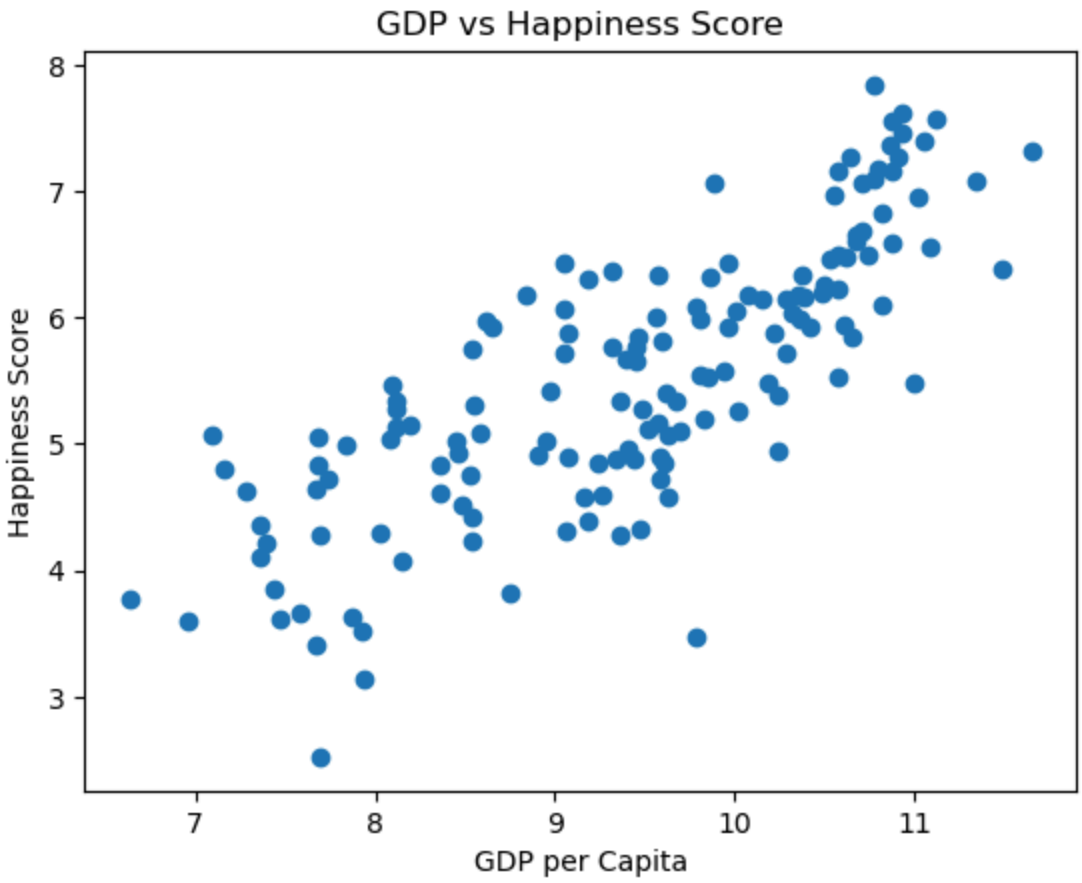
In [ ]:
```
Let's see if our Ladder score has any outlier data:
```

In [14]:
```
plt.boxplot(df['Ladder score'])
plt.show()
```



Scatter Plots

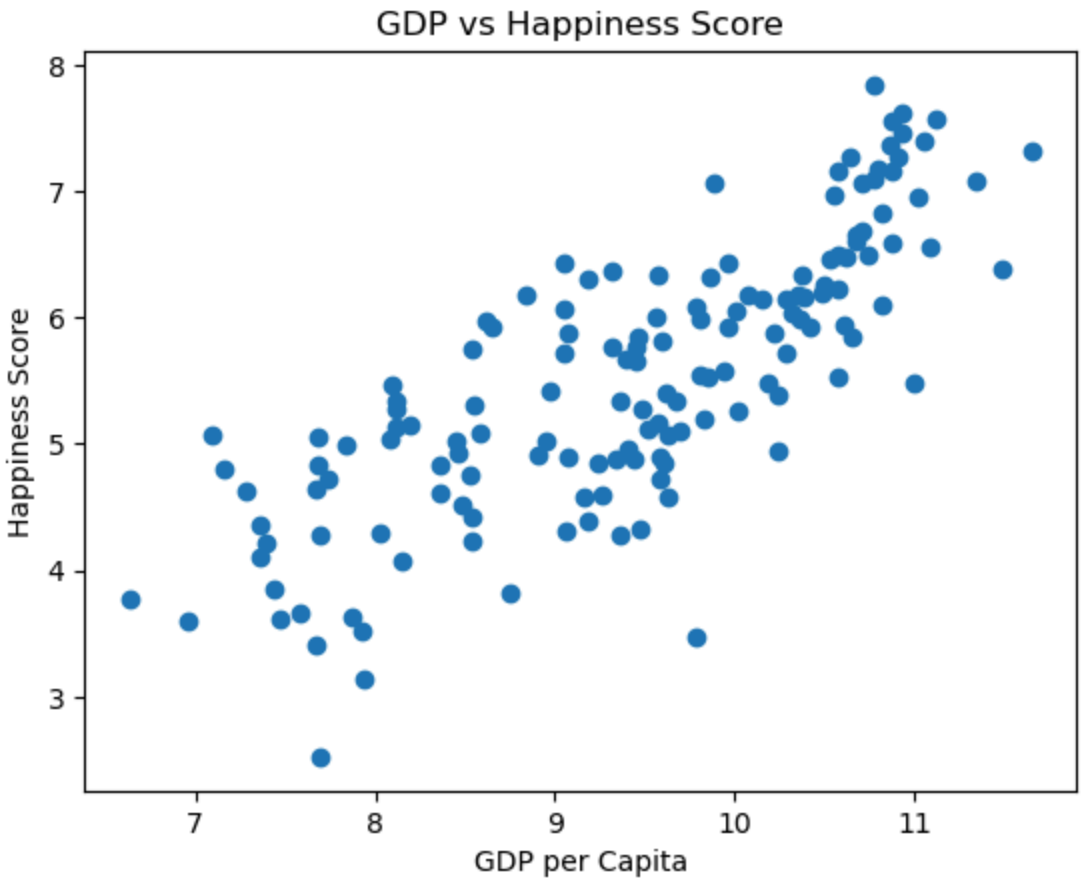
In [15]:
```
plt.scatter(gdp, score)
plt.title('GDP vs Happiness Score')
plt.xlabel('GDP per Capita')
plt.ylabel('Happiness Score')
plt.show()
```



In [ ]:
```
As expected, the higher the score for GDP per Capita, the higher is the happiness s
certain country.
```

Saving Plots

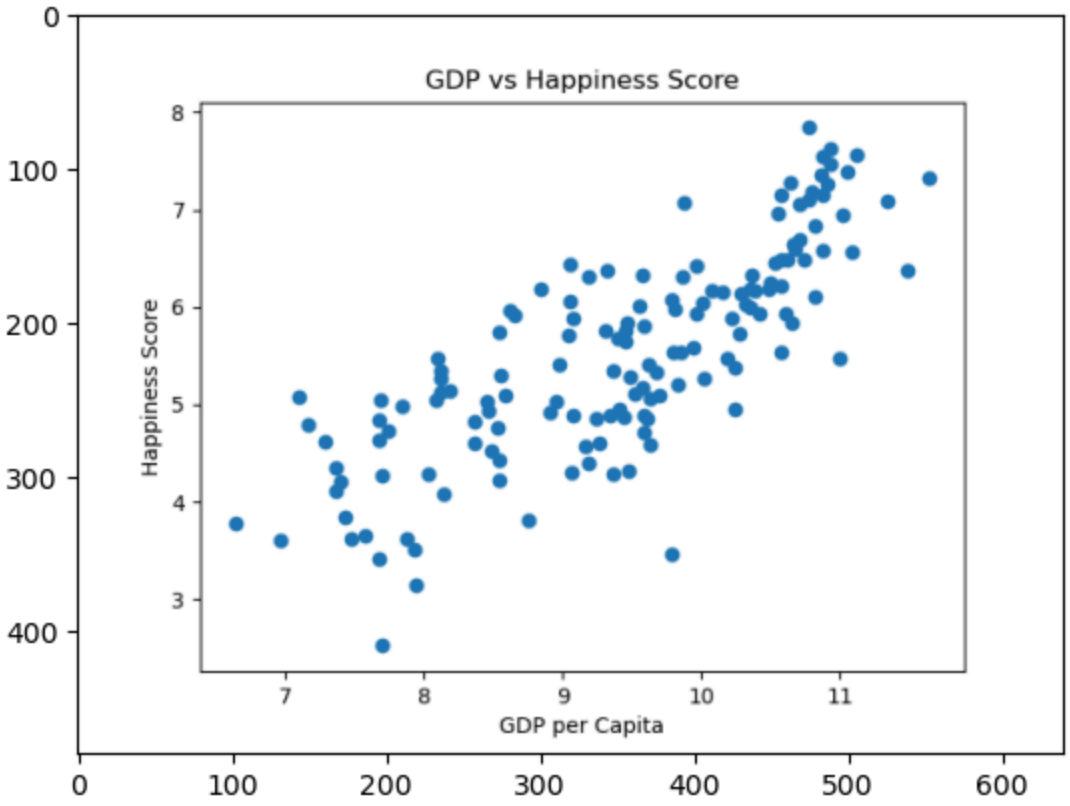
In [16]:
```
fig = plt.figure()
plt.scatter(gdp, score)
plt.title('GDP vs Happiness Score')
plt.xlabel('GDP per Capita')
plt.ylabel('Happiness Score')
fig.savefig('scatterplot.png')
```



In [ ]:
```
The image would have been saved with the filename ‘saveimage.png’.
To view the saved image, we’ll use the matplotlib.image module, as shown below:
```

In [ ]:
```
#Displaying the saved image
```

In [17]:
```
import matplotlib.image as mpimg
image = mpimg.imread("scatterplot.png")
plt.imshow(image)
plt.show()
```



Problem Statement

In [ ]:
```
I am tasked with analyzing the World Happiness Report 2021 dataset to uncover insig
global happiness levels. The dataset includes various factors such as GDP per capit
support, healthy life expectancy, freedom to make life choices, generosity, and per
corruption, all of which potentially influence the happiness scores (Ladder Score)
My goal is to create visualizations that highlight the relationships and distributi
factors, and to identify which factors are most strongly associated with higher hap
```

In [ ]:
```
PS1: How is the distribution of happiness scores (Ladder Score) across different re
PS2: What is the correlation between different factors in the dataset?
PS3: Which are the top 10 happiest countries according to the Ladder Score?
PS4: What is the relationship between GDP per capita and the happiness score?
PS5: How do various factors in the dataset relate to each other and to the happines
```

In [18]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [19]:
```
# Load the dataset
df = pd.read_csv(r'C:\Users\HP\Downloads\world-happiness-report-2021.csv')
df.head(10)
```

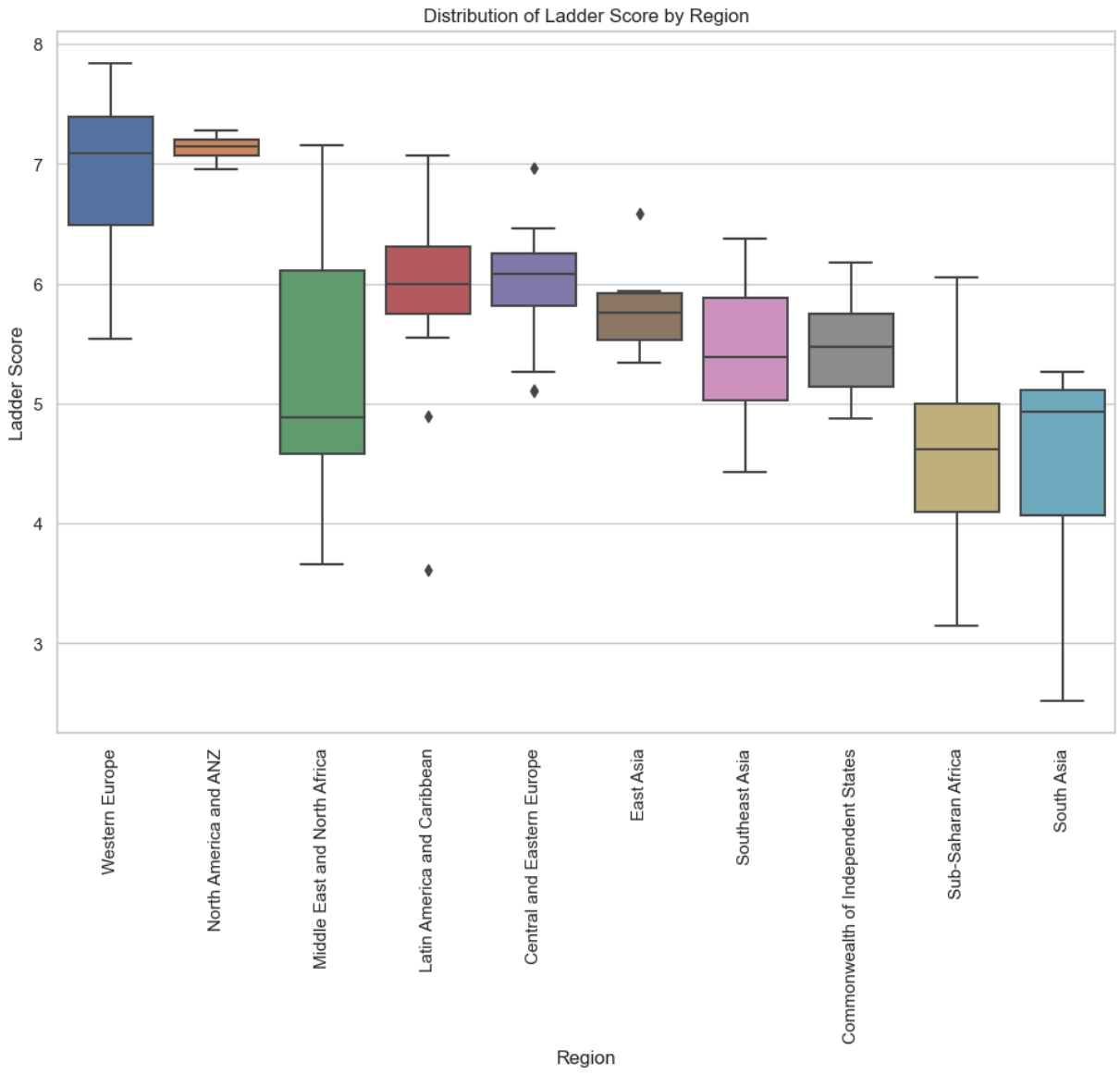
Out[19]:

	Country name	Regional indicator	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Socia support
0	Finland	Western Europe	7.842	0.032	7.904	7.780	10.775	0.954
1	Denmark	Western Europe	7.620	0.035	7.687	7.552	10.933	0.954
2	Switzerland	Western Europe	7.571	0.036	7.643	7.500	11.117	0.942
3	Iceland	Western Europe	7.554	0.059	7.670	7.438	10.878	0.983
4	Netherlands	Western Europe	7.464	0.027	7.518	7.410	10.932	0.942
5	Norway	Western Europe	7.392	0.035	7.462	7.323	11.053	0.954
6	Sweden	Western Europe	7.363	0.036	7.433	7.293	10.867	0.934
7	Luxembourg	Western Europe	7.324	0.037	7.396	7.252	11.647	0.908
8	New Zealand	North America and ANZ	7.277	0.040	7.355	7.198	10.643	0.948
9	Austria	Western Europe	7.268	0.036	7.337	7.198	10.906	0.934

In [ ]:
```
Question1: How is the distribution of happiness scores (Ladder Score) across differ
```

Visualization 1: Distribution of Ladder Score by Region

In [20]:
```
# Set the style for the plots
sns.set(style="whitegrid")
# Distribution of Ladder Score by Region
plt.figure(figsize=(12, 8))
sns.boxplot(x='Regional indicator', y='Ladder score', data=df)
plt.xticks(rotation=90)
plt.title('Distribution of Ladder Score by Region')
plt.xlabel('Region')
plt.ylabel('Ladder Score')
plt.show()
```



Explanation

In [ ]:
- Purpose: To show the distribution of happiness scores across different regions.
- Visualization: A box plot that displays the spread of Ladder Scores within each r
- Insight: This helps identify regions with higher or lower happiness scores and th variation within each region

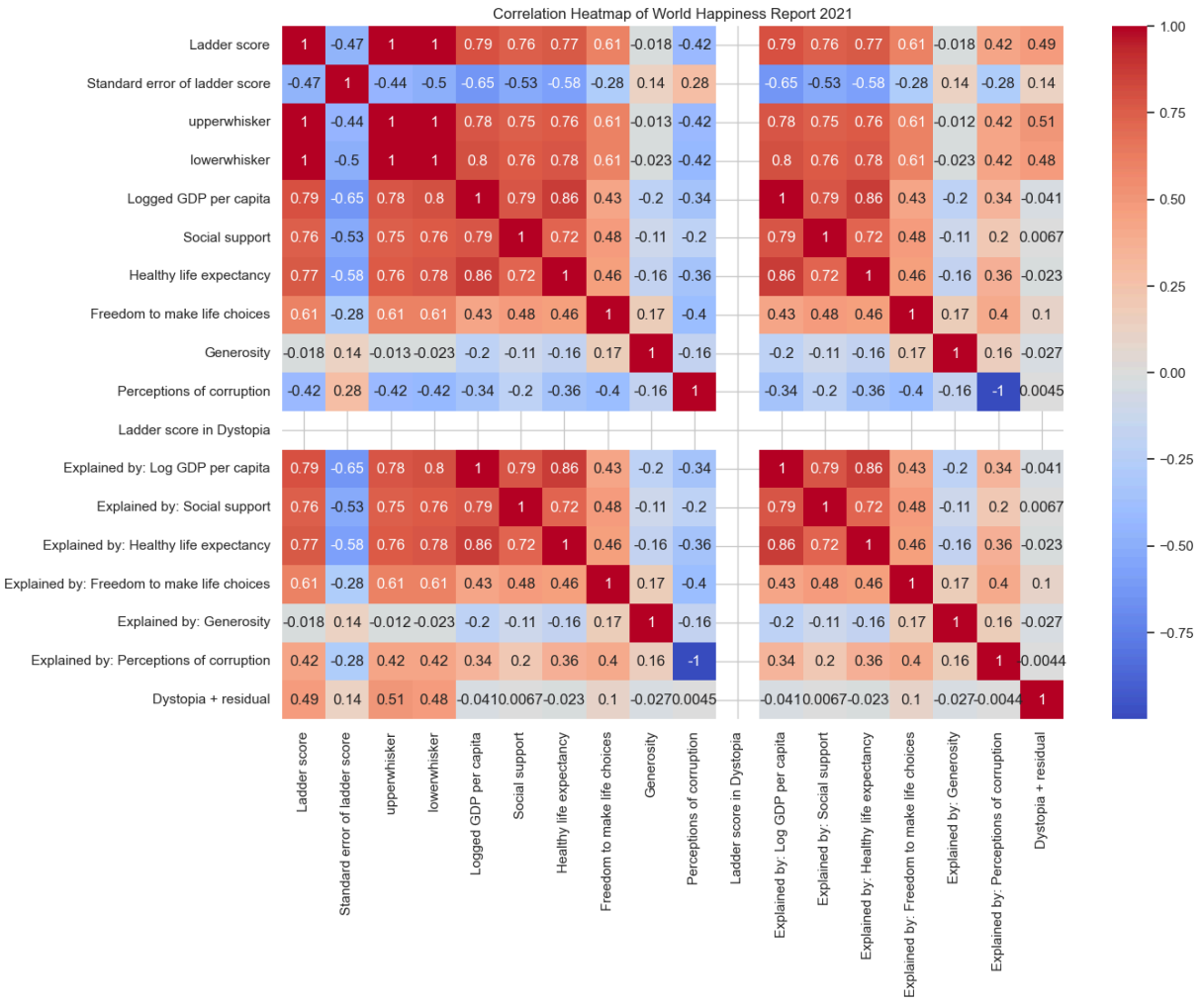
In [ ]:
```
Question2: What is the correlation between different factors in the dataset?
```

Visualization 2: Correlation Heatmap

In [ ]:
```
the corr() method in pandas only works with numerical data, and your DataFrame cont
non-numeric columns. We need to select only the numeric columns before computing th
correlation matrix.
```

In [21]:
```
# Select only numeric columns for correlation
numeric_columns = [
 'Ladder score',
```

```
 'Standard error of ladder score',
 'upperwhisker',
 'lowerwhisker',
 'Logged GDP per capita',
 'Social support',
 'Healthy life expectancy',
 'Freedom to make life choices',
 'Generosity',
 'Perceptions of corruption',
 'Ladder score in Dystopia',
 'Explained by: Log GDP per capita',
 'Explained by: Social support',
 'Explained by: Healthy life expectancy',
 'Explained by: Freedom to make life choices',
 'Explained by: Generosity',
 'Explained by: Perceptions of corruption',
 'Dystopia + residual'
]
# Compute correlation matrix
correlation = df[numeric_columns].corr()
# Plot Correlation Heatmap
plt.figure(figsize=(14, 10))
sns.heatmap(correlation, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Heatmap of World Happiness Report 2021')
plt.show()
```



Explanation:

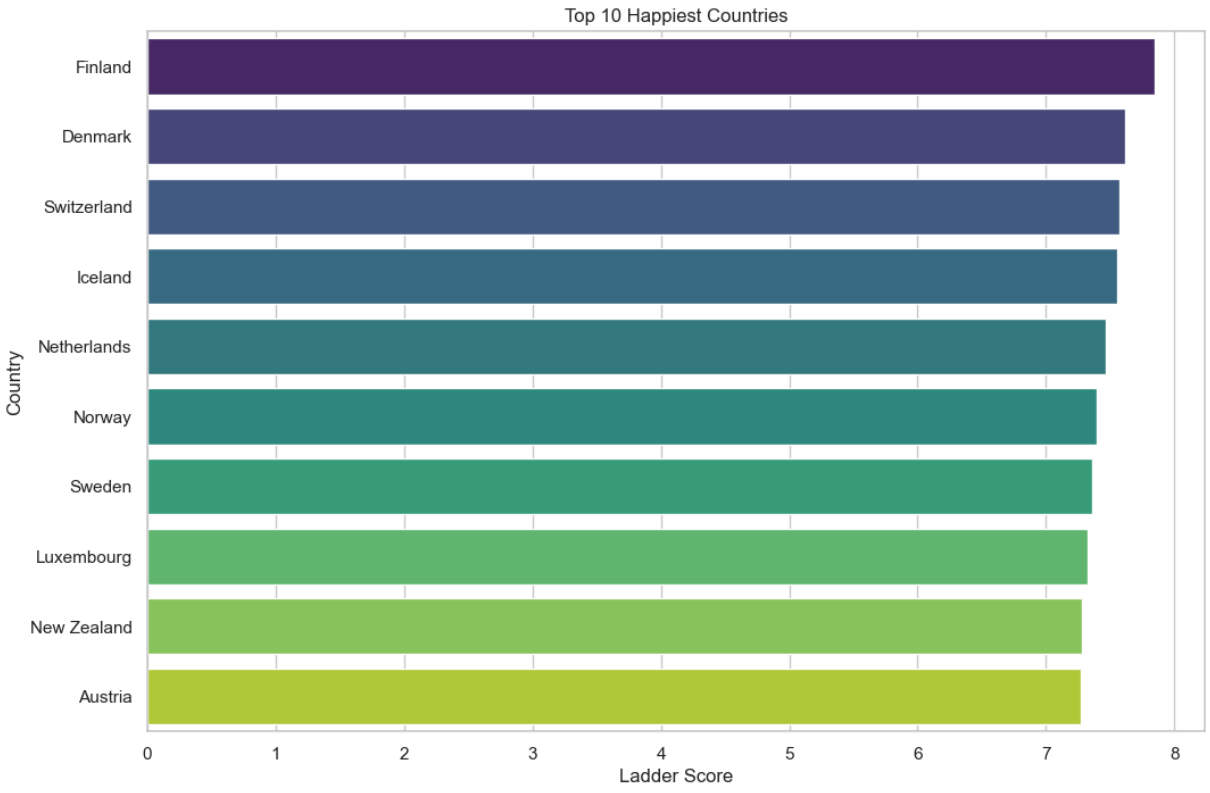
In [ ]:
```
• Purpose: To display the correlation between different numerical variables in the
• Visualization: A heatmap where the color intensity indicates the strength of corr
• Insight: Identifies which factors are most strongly associated with the happiness
```

Question3:

In [ ]:
```
Which are the top 10 happiest countries according to the Ladder Score?
```

Visualization 3: Bar Plot of Top 10 Happiest Countries

In [22]:
```
# Bar Plot of Top 10 Happiest Countries
top_10_happiest = df.nlargest(10, 'Ladder score')
plt.figure(figsize=(12, 8))
sns.barplot(x='Ladder score', y='Country name', data=top_10_happiest,
palette='viridis')
plt.title('Top 10 Happiest Countries')
plt.xlabel('Ladder Score')
plt.ylabel('Country')
plt.show()
```



Explanation:

In [ ]:
```
• Purpose: To highlight the top 10 countries with the highest happiness scores.
• Visualization: A horizontal bar plot showing the Ladder Scores of the top 10 happ
```

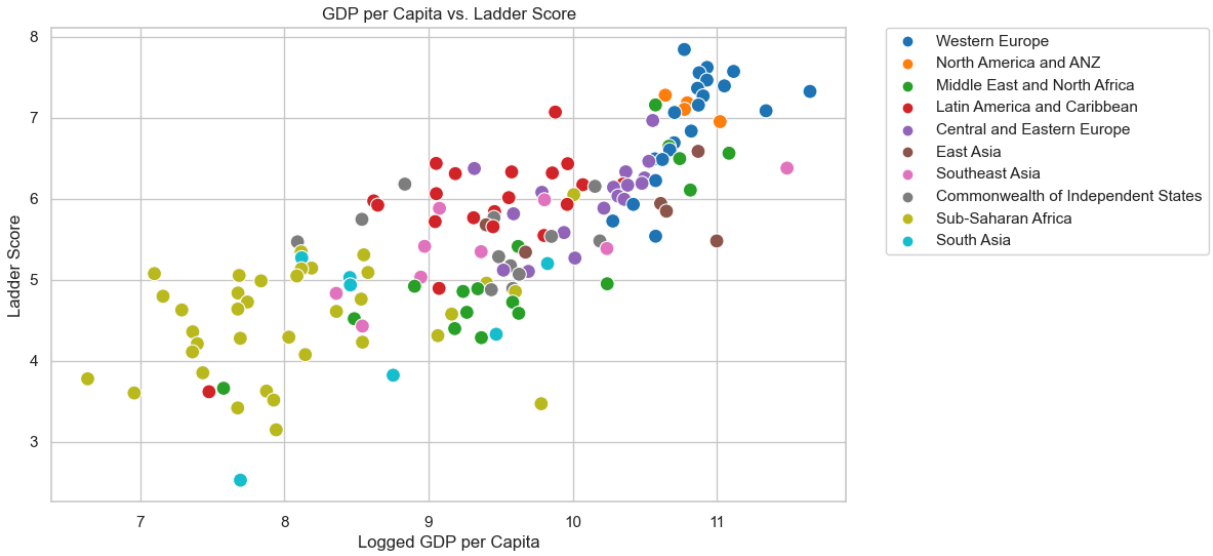
countries.
• Insight: Provides a quick comparison of the happiest countries in the dataset.

Question4:

In [ ]:
```
What is the relationship between GDP per capita and the happiness score?
```

Visualization 4: GDP per Capita vs. Ladder Score

In [23]:
```
# GDP per Capita vs. Ladder Score
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Logged GDP per capita', y='Ladder score', data=df,
hue='Regional indicator', palette='tab10', s=100)
plt.title('GDP per Capita vs. Ladder Score')
plt.xlabel('Logged GDP per Capita')
plt.ylabel('Ladder Score')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```



In [ ]:
```
The plt.legend function can be used to adjust the positioning of the legend in a pl
break down its usage:
```

In [ ]:
```
• bbox_to_anchor=(1.05, 1): This argument specifies the bounding box for the
legend. The coordinates (1.05, 1) place the legend slightly outside the plot area t
right.
• loc=2: This locates the legend at the upper left corner of the bounding box speci
bbox_to_anchor.
• borderaxespad=0.: This sets the padding between the axes and the legend box to
zero.
```

In [ ]:
```
Explanation:
The legend is being positioned outside the main plotting area, to the right, to avo
with the data points. This is particularly useful in scatter plots with many catego
```

Explanation:

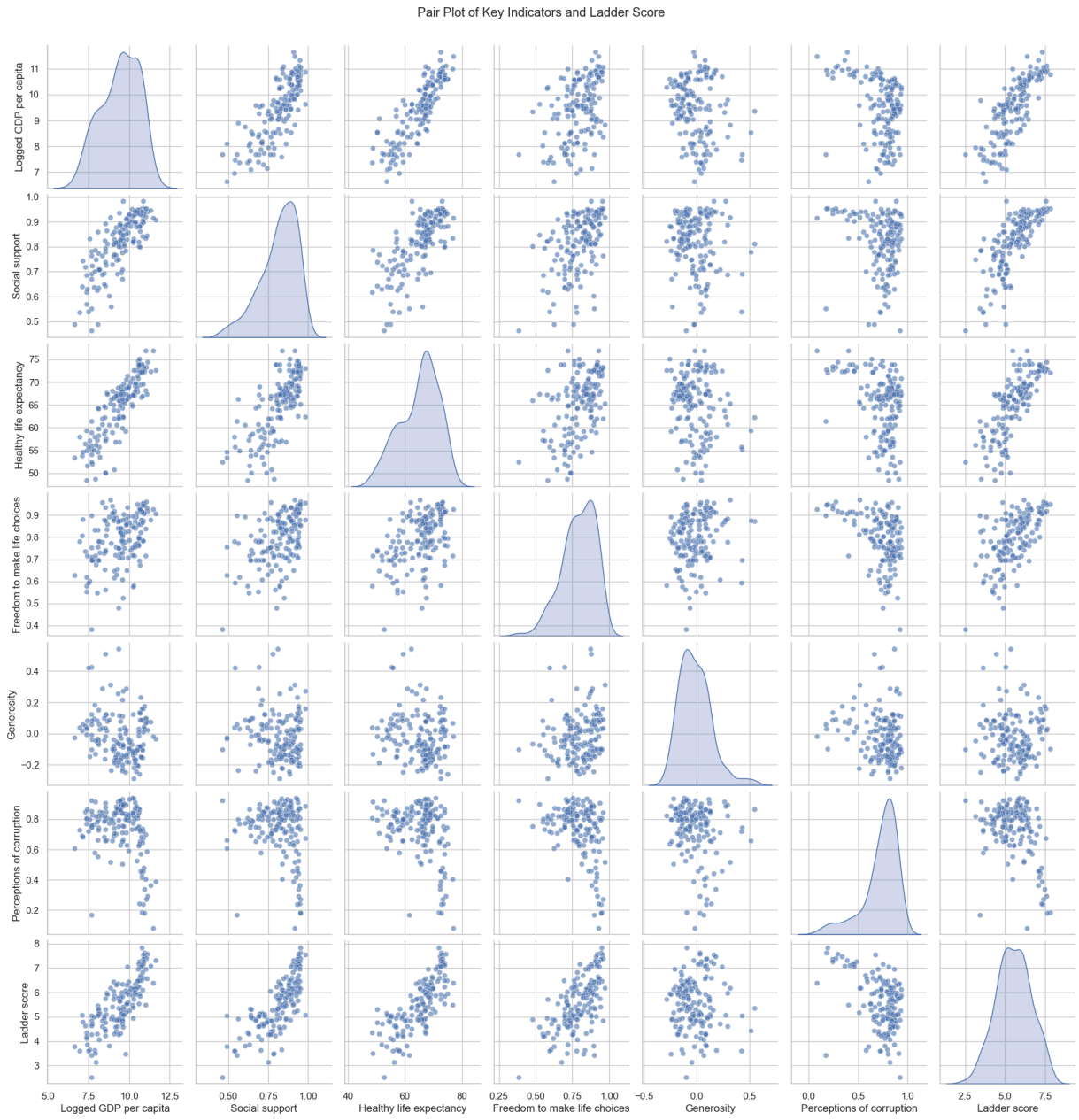
In [ ]:
```
• Purpose: To show the relationship between GDP per capita and the happiness score.
• Visualization: A scatter plot where each point represents a country, color-coded
region.
• Insight: Helps determine if wealthier countries tend to have higher happiness sco
how this relationship varies by region.
```

Question5:

In [ ]:
```
How do various factors in the dataset relate to each other and to the happiness sco
```

Visualization 5: Pair Plot to Explore Relationships

In [30]:
```
# Pair Plot to Explore Relationships
pair_columns = ['Logged GDP per capita', 'Social support', 'Healthy life expectancy
 'Freedom to make life choices', 'Generosity', 'Perceptions of corruption', 'Ladder
sns.pairplot(df[pair_columns],diag_kind='kde',plot_kws={'alpha': 0.6})
plt.suptitle('Pair Plot of Key Indicators and Ladder Score', y=1.02)
plt.show()
```



Explanation of above code:

In [ ]:
```
python sns.pairplot(df[pair_columns],
diag_kind='kde', plot_kws={'alpha': 0.6})
```

In [ ]:
```
• sns.pairplot: A function in Seaborn to create pair plots.
• df[pair_columns]: Selects only the specified columns from the DataFrame.
• diag_kind='kde': Specifies that the diagonal plots should be Kernel Density Estim
(KDE) plots instead of histograms.
• plot_kws={'alpha': 0.6}: Keyword arguments for the individual plots. Here,
alpha=0.6 sets the transparency of the points in the scatter plots to 60%.
```

In [ ]:
```
Add a Title to the Entire Plot: python plt.suptitle('Pair Plot of Key
Indicators and Ladder Score', y=1.02)
```

In [ ]:
```
• plt.suptitle: Adds a centered title to the figure.
• y=1.02: Positions the title slightly above the top of the plot.
```

Explanation:

In [ ]:
```
• Purpose: To visualize relationships and potential correlations between various fa
and the happiness score.
• Visualization: A matrix of scatter plots for each pair of variables, with KDE plo
diagonal.
• Insight: Provides a comprehensive view of how different factors are related to ea
and to the happiness score.
```

Conclusion:

In [ ]:
```
These visualizations help in understanding the factors that influence happiness acr
countries and regions. They are designed to teach how to use matplotlib and seaborn
and visualize data effectively. Each visualization addresses a specific question, p
into the dataset.
```

In [ ]: