Exploring the World Happiness Index and Its Correlations

Welcome to this data exploration project, where I embark on a journey to showcase my Python skills in data cleaning and manipulation. In a world driven by data, understanding the intricate web of factors contributing to happiness becomes more than a mere intellectual curiosity—it delves into the very essence of human well-being. This data analysis report immerses itself in the World Happiness Index, an extensive dataset meticulously quantifying and ranking the happiness of nations.

At the heart of this analysis lies the pursuit of unraveling insights concerning happiness and the multifaceted external factors that exert their influence. My journey commences with a meticulous exploration of the World Happiness Index dataset, focusing on the distribution of happiness scores. Additionally, I set my sights on discerning potential correlations between happiness and key attributes, including GDP per capita and social support. As an experimental extension, I'll also scrutinize the impact of significant world events, such as the outbreak of the conflict in Ukraine, to assess their implications on the happiness scores of the involved nations.

This expedition through data promises to unveil intricate connections and provide profound insights into the intricate interplay between happiness and the myriad social and economic factors shaping our world. I extend to you an invitation to join me in this voyage, where we are poised to gain a deeper understanding of the profound dynamics governing human well-being.

Data Exploration

Data Overview

In [1]:

```
# Load all necessary libraries
import pandas as pd
import os
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set()
```

The dataset comes from: https://www.kaggle.com/datasets/unsdsn/world-happiness.
First I will create a dataframe for each year from 2015-2023.

In [2]:

```
# Initialize variables for each DataFrame
df_2015 = None
df_2016 = None
df_2017 = None
df_2018 = None
df_2019 = None
df_2020 = None
df_2021 = None
df_2022 = None
```

```python
df_2023 = None

# Define the directory path with absolute path
directory_path = os.path.abspath('../happiness')

# Loop through CSV files in the directory
for file in os.listdir(directory_path):
    if file.endswith(".csv"):
        # Extract the year from the file name
        year = file.split("_")[1].split(".")[0]

        file_path = os.path.join(directory_path, file)  # Create full file
        df = pd.read_csv(file_path)

        # Assign the DataFrame to the corresponding variable
        if year == '2015':
            df_2015 = df
        elif year == '2016':
            df_2016 = df
        elif year == '2017':
            df_2017 = df
        elif year == '2018':
            df_2018 = df
        elif year == '2019':
            df_2019 = df
        elif year == '2020':
            df_2020 = df
        elif year == '2021':
            df_2021 = df
        elif year == '2022':
            df_2022 = df
        elif year == '2023':
            df_2023 = df
```

Data Cleaning

In [3]:

```python
# Adding new column "year" for each dataframe
df_2015["year"] = 2015
df_2016["year"] = 2016
df_2017["year"] = 2017
df_2018["year"] = 2018
df_2019["year"] = 2019
df_2020["year"] = 2020
df_2021["year"] = 2021
df_2022["year"] = 2022
df_2023["year"] = 2023
```

In [4]:

```python
# Checking datatypes
df_2015.dtypes
```

Out[4]:

```
country                          object
region                           object
happiness_score                 float64
gdp_per_capita                  float64
social_support                  float64
healthy_life_expectancy         float64
freedom_to_make_life_choices    float64
generosity                      float64
perceptions_of_corruption       float64
year                              int64
dtype: object
```

In [5]:

```
# Converting the year column into a pandas datetime object
dataframes = [df_2015, df_2016, df_2017, df_2018, df_2019, df_2020, df_2021,

for df in dataframes:
    df['year'] = pd.to_datetime(df['year'], format='%Y')

df_2015.dtypes
```

Out[5]:

```
country                                  object
region                                   object
happiness_score                         float64
gdp_per_capita                          float64
social_support                          float64
healthy_life_expectancy                 float64
freedom_to_make_life_choices            float64
generosity                              float64
perceptions_of_corruption               float64
year                             datetime64[ns]
dtype: object
```

In [51]:

```
# Check for missing values in each DataFrame for the year 2015
for df in dataframes:
    if df['year'].iloc[0].year == 2015:
        missing_values = df.isna().sum()
        print(f"Missing values in DataFrame for year {df['year'].iloc[0].yea
```

```
Missing values in DataFrame for year 2015:
country                         0
region                          0
happiness_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
year                            0
dtype: int64
```

Since for all the other years there are also almost no missing values, we can simply continue our analysis.

Data Exploration

Given the minimal presence of missing values, we will proceed with our analysis without concern for such gaps. Let us commence our exploration. Initially, we shall examine the countries boasting the highest happiness scores for each respective year.

In [8]:

```
for df in dataframes:
    # Group by "Country" and calculate the mean happiness score for each cou
    grouped_by_country = df.groupby('country')['happiness_score'].mean()

    # Convert the grouped data to a DataFrame, set "country" as the index, a
    result_df = grouped_by_country.reset_index().set_index('country', drop=T

    # Sort the DataFrame by "Happiness Score" in descending order
    result_df = result_df.sort_values(by='happiness_score', ascending=False)

    # Add a "Rank" column based on the sorted order as an integer
    result_df['Rank'] = result_df['happiness_score'].rank(ascending=False, r
    # Get the top 10 happiest countries with ranks, country, and happiness s
    top_10 = result_df[['Rank', 'happiness_score']].head(10)
    top_10.reset_index(inplace=True)
    top_10['Rank'] = top_10['Rank'] - top_10['Rank'].min() + 1  # Start rank

    # Calculate the maximum width needed for the country column
    max_country_width = top_10['country'].str.len().max()

    # Print the top 10 happiest countries with aligned columns
    print(f"Top 10 Happiest Countries in {df['year'].iloc[0].year}:")
    for index, row in top_10.iterrows():
        print(f"{row['Rank']:2d}. {row['country']:{max_country_width}} {row

    print()
```

```
Top 10 Happiest Countries in 2015:
 1. Switzerland 7.587
 2. Iceland     7.561
 3. Denmark     7.527
 4. Norway      7.522
 5. Canada      7.427
 6. Finland     7.406
 7. Netherlands 7.378
 8. Sweden      7.364
 9. New Zealand 7.286
10. Australia   7.284

Top 10 Happiest Countries in 2016:
 1. Denmark     7.526
 2. Switzerland 7.509
 3. Iceland     7.501
 4. Norway      7.498
 5. Finland     7.413
 6. Canada      7.404
 7. Netherlands 7.339
 8. New Zealand 7.334
 9. Australia   7.313
10. Sweden      7.291

Top 10 Happiest Countries in 2017:
 1. Norway      7.537
 2. Denmark     7.522
 3. Iceland     7.504
 4. Switzerland 7.494
 5. Finland     7.469
 6. Netherlands 7.377
 7. Canada      7.316
 8. New Zealand 7.314
 9. Australia   7.284
10. Sweden      7.284

Top 10 Happiest Countries in 2018:
 1. Finland     7.632
 2. Norway      7.594
 3. Denmark     7.555
 4. Iceland     7.495
 5. Switzerland 7.487
 6. Netherlands 7.441
 7. Canada      7.328
 8. New Zealand 7.324
 9. Sweden      7.314
10. Australia   7.272

Top 10 Happiest Countries in 2019:
 1. Finland     7.769
 2. Denmark     7.600
 3. Norway      7.554
 4. Iceland     7.494
 5. Netherlands 7.488
 6. Switzerland 7.480
 7. Sweden      7.343
 8. New Zealand 7.307
 9. Canada      7.278
10. Austria     7.246

Top 10 Happiest Countries in 2020:
 1. Finland     7.809
 2. Denmark     7.646
 3. Switzerland 7.560
 4. Iceland     7.504
 5. Norway      7.488
 6. Netherlands 7.449
 7. Sweden      7.353
 8. New Zealand 7.300
 9. Austria     7.294
10. Luxembourg  7.238

Top 10 Happiest Countries in 2021:
 1. Finland     7.842
 2. Denmark     7.620
 3. Switzerland 7.571
 4. Iceland     7.554
 5. Netherlands 7.464
 6. Norway      7.392
 7. Sweden      7.363
 8. Luxembourg  7.324
 9. New Zealand 7.277
10. Austria     7.268

Top 10 Happiest Countries in 2022:
 1. Finland     7.821
 2. Denmark     7.636
 3. Iceland     7.557
 4. Switzerland 7.512
 5. Netherlands 7.415
 6. Luxembourg  7.404
 7. Sweden      7.384
 8. Norway      7.365
 9. Israel      7.364
10. New Zealand 7.200

Top 10 Happiest Countries in 2023:
 1. Finland     7.804
 2. Denmark     7.586
 3. Iceland     7.530
 4. Israel      7.473
 5. Netherlands 7.403
 6. Sweden      7.395
 7. Norway      7.315
 8. Switzerland 7.240
 9. Luxembourg  7.228
10. New Zealand 7.123
```

In [9]:

```
# Now lets see what countries score the highest in total:
total_scores_df = result_df.groupby('country')['happiness_score'].sum().rese

# Sort the DataFrame by total happiness scores in descending order
total_scores_df = total_scores_df.sort_values(by='happiness_score', ascendi

# Get the 5 countries with the highest total happiness scores
highest_5_countries = total_scores_df[['country', 'happiness_score']].head(5

print("Top 5 Countries with the Highest Total Happiness Scores:")
for index, row in highest_5_countries.iterrows():
    print(f"{row['country']} - Total Happiness Score: {row['happiness_score
```

```
Top 5 Countries with the Highest Total Happiness Scores:
Finland - Total Happiness Score: 7.804
Denmark - Total Happiness Score: 7.586
Iceland - Total Happiness Score: 7.530
Israel - Total Happiness Score: 7.473
Netherlands - Total Happiness Score: 7.403
```

What could be the common denominator for these countries. Well here are some ideas:

Quality of Life: These countries often have high standards of living, with access to quality healthcare, education, and social services.

Social Support: Strong social support systems and close-knit communities contribute to the well-being of their citizens.

Political Stability: These nations tend to have stable political environments, low levels of corruption, and effective governance.

Safety and Security: Low crime rates and a sense of safety are common features in these countries.

Healthcare: Access to good healthcare and a focus on well-being and public health contribute to the overall happiness of the population.

Education: High-quality education systems often play a role in providing opportunities for personal and professional development.

Work-Life Balance: Many of these countries emphasize a healthy work-life balance and have shorter working hours.

Lets put these assumptions to the test. Fortunately, the dataset contains the requisite information to conduct such an investigation.

Correlation between Generosity and Hapiness Score

In [50]:

```python
for df in dataframes:
    # Calculate the correlation between "Generosity" and "Happiness Score"
    correlation = df['generosity'].corr(df['happiness_score'])

    # Determine the correlation strength and provide an explanation
    if correlation > 0.7:
        strength = "Strong positive"
    elif correlation < -0.7:
        strength = "Strong negative"

    elif 0.3 <= correlation <= 0.7:
        strength = "Moderate positive"

    elif -0.7 <= correlation <= -0.3:
        strength = "Moderate negative"

    else:
        strength = "Weak or no"

    # Print the correlation coefficient and its interpretation
    print(f"Generosity and Happiness in {df['year'].iloc[0].year}: {correla
```

```
Generosity and Happiness in 2015: 0.18 (Weak or no correlation)
Generosity and Happiness in 2016: 0.16 (Weak or no correlation)
Generosity and Happiness in 2017: 0.16 (Weak or no correlation)
Generosity and Happiness in 2018: 0.14 (Weak or no correlation)
Generosity and Happiness in 2019: 0.08 (Weak or no correlation)
Generosity and Happiness in 2020: 0.07 (Weak or no correlation)
Generosity and Happiness in 2021: -0.02 (Weak or no correlation)
Generosity and Happiness in 2022: 0.06 (Weak or no correlation)
Generosity and Happiness in 2023: 0.04 (Weak or no correlation)
```

In summary, it seems that there is a minimal correlation between a country's happiness score and the generosity of its citizens.

Correlation between Healthy Life Expectancy and Hapiness Score

In [49]:

```
for df in dataframes:
    # Calculate the correlation between "Life Expectancy" and "Happiness Sc
    correlation = df['healthy_life_expectancy'].corr(df['happiness_score'])

    # Determine the correlation strength and provide an explanation
    if correlation > 0.7:
        strength = "Strong positive"
    elif correlation < -0.7:
        strength = "Strong negative"

    elif 0.3 <= correlation <= 0.7:
        strength = "Moderate positive"

    elif -0.7 <= correlation <= -0.3:
        strength = "Moderate negative"

    else:
        strength = "Weak or no"

    print(f"HALE and Happiness in {df['year'].iloc[0].year}: {correlation:.
```

```
HALE and Happiness in 2015: 0.72 (Strong positive correlation)
HALE and Happiness in 2016: 0.77 (Strong positive correlation)
HALE and Happiness in 2017: 0.78 (Strong positive correlation)
HALE and Happiness in 2018: 0.78 (Strong positive correlation)
HALE and Happiness in 2019: 0.78 (Strong positive correlation)
HALE and Happiness in 2020: 0.77 (Strong positive correlation)
HALE and Happiness in 2021: 0.77 (Strong positive correlation)
HALE and Happiness in 2022: 0.74 (Strong positive correlation)
HALE and Happiness in 2023: 0.75 (Strong positive correlation)
```

A robust correlation is evident between the happiness score and healthy life expectancy (HALE). To elucidate, HALE at birth represents the mean number of years an individual can anticipate living in a state of "full health," considering the years spent in less-than-optimal health due to illness and/or injury.

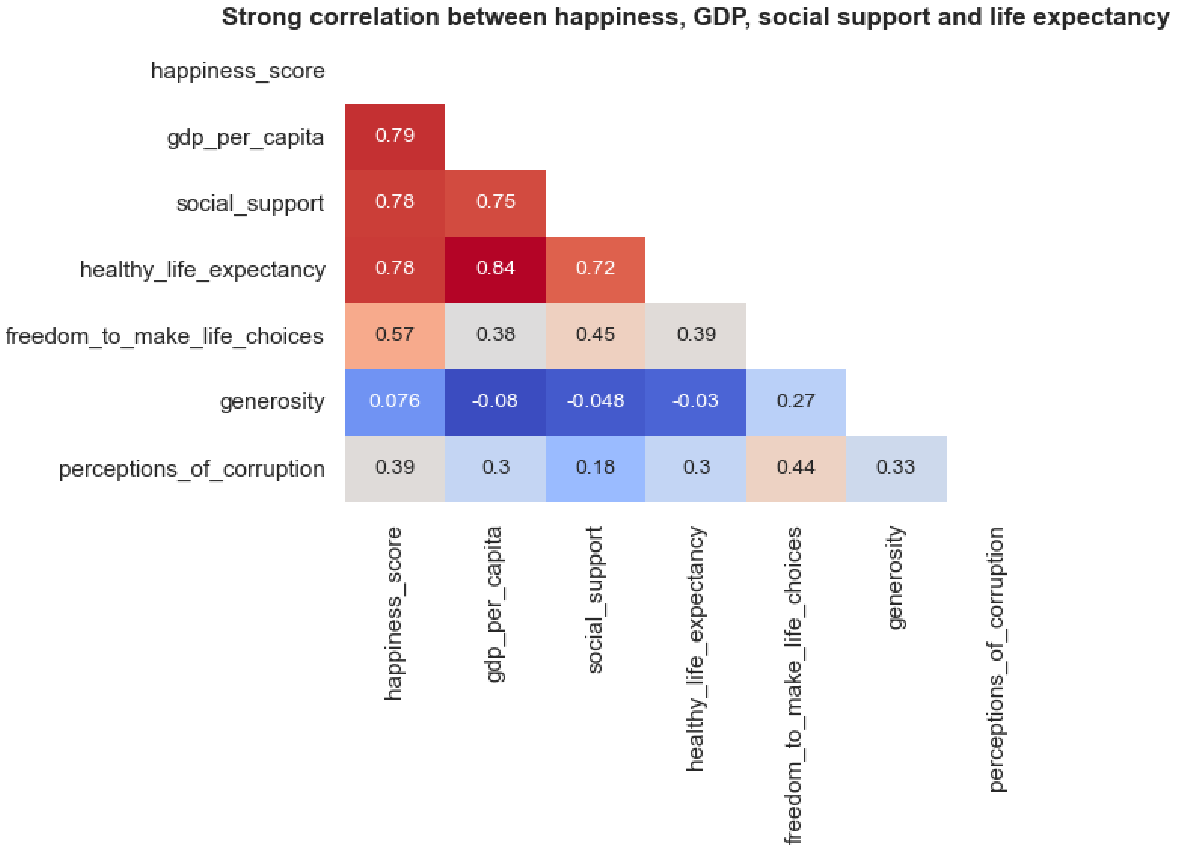
To simplify the analysis, let's generate a heatmap encompassing all factors in the columns to identify strong correlations specifically for the year 2019. This choice is based on the assumption that the impact of these factors is unlikely to fluctuate significantly from year to year. Elements that induce unhappiness in 2018, for instance, are expected to yield similar results in 2019 and subsequent years.

In [12]:

```
# Calculate the correlation matrix
corr_matrix = df_2019.corr()

# Create a mask to only display upper triangle (to avoid duplication)
mask = np.triu(np.ones_like(corr_matrix, dtype=bool))

# Set up the matplotlib figure
plt.figure(figsize=(6, 4))

# Remove grid lines
sns.set_style("white")

# Customize the heatmap using Seaborn
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", mask=mask, cbar=False

# Set the title
plt.title("Strong correlation between happiness, GDP, social support and li
          weight = "bold")
```

```
# Show the plot
plt.show()
```

```
/var/folders/gf/945vyt3n2m3flxbfw360lmb40000gn/T/ipykernel_5152/20106881.p
y:2: FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only vali
d columns or specify the value of numeric_only to silence this warning.
  corr_matrix = df_2019.corr()
```



Region-based Analysis:

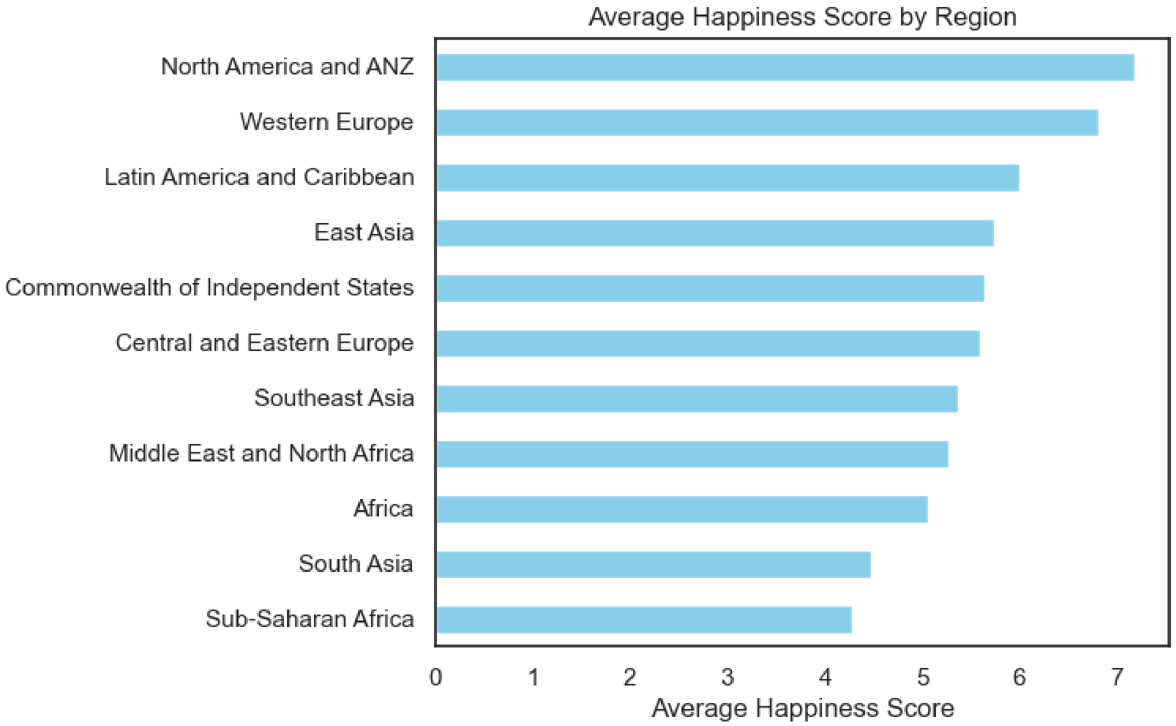
Regional Happiness:

Lets group the data by region and calculate the average happiness score for each. Then we can visualize regional happiness using bar charts.

In [54]:

```
# Group data by region and calculate average happiness score
dataframes = [df_2015, df_2016, df_2017, df_2018, df_2019, df_2020, df_2021,

# Concatenate the DataFrames into one
combined_df = pd.concat(dataframes, ignore_index=True)
regional_happiness = combined_df.groupby('region')['happiness_score'].mean(

# Create the color
colors = ['skyblue' for region in regional_happiness.index]

plt.figure(figsize=(6, 5))
ax = regional_happiness.sort_values(ascending=True).plot(kind='barh', color
plt.title('Average Happiness Score by Region')

# Remove the y-axis label
ax.set_ylabel('')
```

```
plt.xlabel('Average Happiness Score')
plt.show()
```



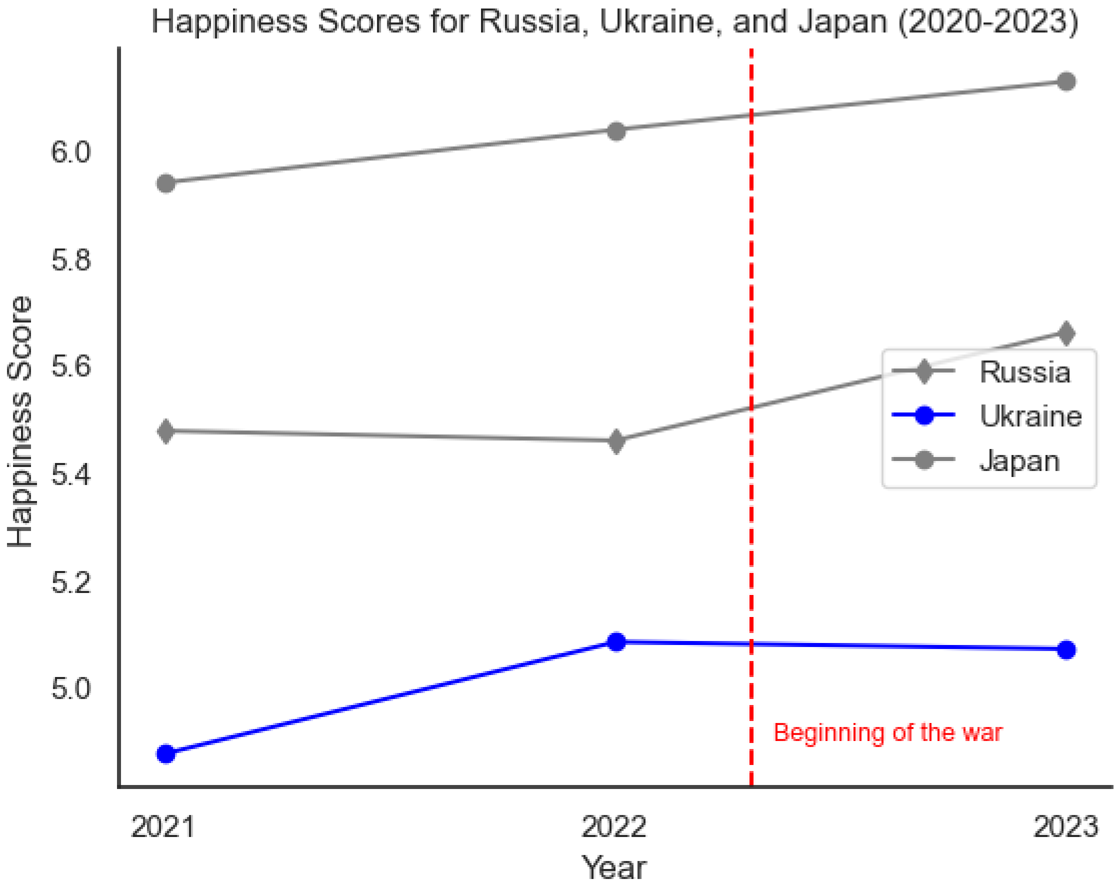
War and conflict:

Out of curiosity, I would like to investigate whether the conflict between Russia and Ukraine had any discernible impact on their happiness scores. If such an impact occurred, we would anticipate a drop in their happiness scores in late February and early March 2021. Let's create line graphs for each of these countries and examine the data to discern any patterns or anomalies during that time frame.

In [14]:

```
# Filter dataframes for the years 2020 to 2023
filtered_dataframes = [df for df in dataframes if 2021 <= df['year'].iloc[0

# Create a figure and axis
fig, ax = plt.subplots()

# Data for Russia, Ukraine, and Japan
data = {'Russia': [], 'Ukraine': [], 'Japan': []}

# Years
years = list(range(2021, 2024))

# Extract happiness scores
for year in years:
    for country in data.keys():
        happiness_scores = [df[df['country'] == country]['happiness_score']
        if happiness_scores:
            data[country].append(happiness_scores[0])
        else:
            data[country].append(None)

# Plot lines for Russia, Ukraine, and Japan
ax.plot(years, data['Russia'], label='Russia', color='grey', marker='d')
ax.plot(years, data['Ukraine'], label='Ukraine', color='blue', marker='o')
ax.plot(years, data['Japan'], label='Japan', color='grey', marker='o')

# Set plot labels and title
```

```
plt.xlabel('Year')
plt.ylabel('Happiness Score')
plt.title('Happiness Scores for Russia, Ukraine, and Japan (2020-2023)')

# Add a vertical line for the beginning of the war
plt.axvline(x=2022.3, color='red', linestyle='--')
plt.text(2022.35, 4.9, 'Beginning of the war', fontsize=9, color='red')

# Set x-axis ticks as integers
plt.xticks(years)
sns.set_style("white")

# Set legend
plt.legend()
sns.despine()

# Show the plot
plt.show()
```



It is not surprising that Ukraine has experienced a small decrease in its happiness score since the onset of the war. This trend, however, is not observable in the case of Russia's happiness score. On the contrary, Russia's happiness score has increased since the war began. To provide a reference point, I have included Japan's happiness score, which serves as an example of a country that has not been directly affected by the conflict.

Countries with the lowest happiness index:

I would like to conclude this analysis by examining the countries with the lowest happiness scores. Are there any discernible patterns, such as political systems in place, that might contribute to these low scores? Let's explore and find out.

In [15]:

```
for df in dataframes:
    # Group by "Country" and calculate the mean happiness score for each cou
    grouped_by_country = df.groupby('country')['happiness_score'].mean()

    # Convert the grouped data to a DataFrame, set "country" as the index, a
    result_df = grouped_by_country.reset_index().set_index('country', drop=T

    # Sort the DataFrame by "Happiness Score" in ascending order to get the
    result_df = result_df.sort_values(by='happiness_score', ascending=True)

    # Add a "Rank" column based on the sorted order as an integer
    result_df['Rank'] = result_df['happiness_score'].rank(ascending=True, me

    # Get the top 10 unhappiest countries with ranks, country, and happiness
    bottom_10 = result_df[['Rank', 'happiness_score']].head(10)
    bottom_10.reset_index(inplace=True)
    bottom_10['Rank'] = bottom_10['Rank'] - bottom_10['Rank'].min() + 1  # S

    # Calculate the maximum width needed for the country column
    max_country_width = bottom_10['country'].str.len().max()

    # Print the top 10 unhappiest countries with aligned columns
    print(f"Top 10 Unhappiest Countries in {df['year'].iloc[0].year}:")
    for index, row in bottom_10.iterrows():
        print(f"{row['Rank']:2d}. {row['country']:{max_country_width}} {row

    print()
```

```
Top 10 Unhappiest Countries in 2015:
 1. Togo         2.839
 2. Burundi      2.905
 3. Syria        3.006
 4. Benin        3.340
 5. Rwanda       3.465
 6. Afghanistan  3.575
 7. Burkina Faso 3.587
 8. Ivory Coast  3.655
 9. Guinea       3.656
10. Chad         3.667

Top 10 Unhappiest Countries in 2016:
 1. Burundi     2.905
 2. Syria       3.069
 3. Togo        3.303
 4. Afghanistan 3.360
 5. Benin       3.484
 6. Rwanda      3.515
 7. Guinea      3.607
 8. Liberia     3.622
 9. Tanzania    3.666
10. Madagascar  3.695

Top 10 Unhappiest Countries in 2017:
 1. Central African Republic 2.693
 2. Burundi                  2.905
 3. Tanzania                 3.349
 4. Syria                    3.462
 5. Rwanda                   3.471
 6. Togo                     3.495
 7. Guinea                   3.507
 8. Liberia                  3.533
 9. South Sudan              3.591
10. Yemen                    3.593

Top 10 Unhappiest Countries in 2018:
 1. Burundi                  2.905
 2. Central African Republic 3.083
 3. South Sudan              3.254
 4. Tanzania                 3.303
 5. Yemen                    3.355
 6. Rwanda                   3.408
 7. Syria                    3.462
 8. Liberia                  3.495
 9. Haiti                    3.582
10. Malawi                   3.587

Top 10 Unhappiest Countries in 2019:
 1. South Sudan              2.853
 2. Central African Republic 3.083
 3. Afghanistan              3.203
 4. Tanzania                 3.231
 5. Rwanda                   3.334
 6. Yemen                    3.380
 7. Malawi                   3.410
 8. Syria                    3.462
 9. Botswana                 3.488
10. Haiti                    3.597

Top 10 Unhappiest Countries in 2020:
 1. Afghanistan              2.567
 2. South Sudan              2.817
 3. Zimbabwe                 3.299
 4. Rwanda                   3.312
 5. Central African Republic 3.476
 6. Tanzania                 3.476
 7. Botswana                 3.479
 8. Yemen                    3.527
 9. Malawi                   3.538
10. India                    3.573

Top 10 Unhappiest Countries in 2021:
 1. Afghanistan 2.523
 2. Zimbabwe    3.145
 3. Rwanda      3.415
 4. Botswana    3.467
 5. Lesotho     3.512
 6. Malawi      3.600
 7. Haiti       3.615
 8. Tanzania    3.623
 9. Yemen       3.658
10. Burundi     3.775

Top 10 Unhappiest Countries in 2022:
 1. Afghanistan  2.404
 2. Lebanon      2.955
 3. Zimbabwe     2.995
 4. Rwanda       3.268
 5. Botswana     3.471
 6. Lesotho      3.512
 7. Sierra Leone 3.574
 8. Tanzania     3.702
 9. Malawi       3.750
10. Zambia       3.760

Top 10 Unhappiest Countries in 2023:
 1. Afghanistan      1.859
 2. Lebanon          2.392
 3. Sierra Leone     3.138
 4. Zimbabwe         3.204
 5. Congo (Kinshasa) 3.207
 6. Botswana         3.435
 7. Malawi           3.495
 8. Comoros          3.545
 9. Tanzania         3.694
10. Zambia           3.982
```

To facilitate the process of identifying the countries that appear most frequently in the analysis, we can use a simple Python code for that purpose.

In [16]:

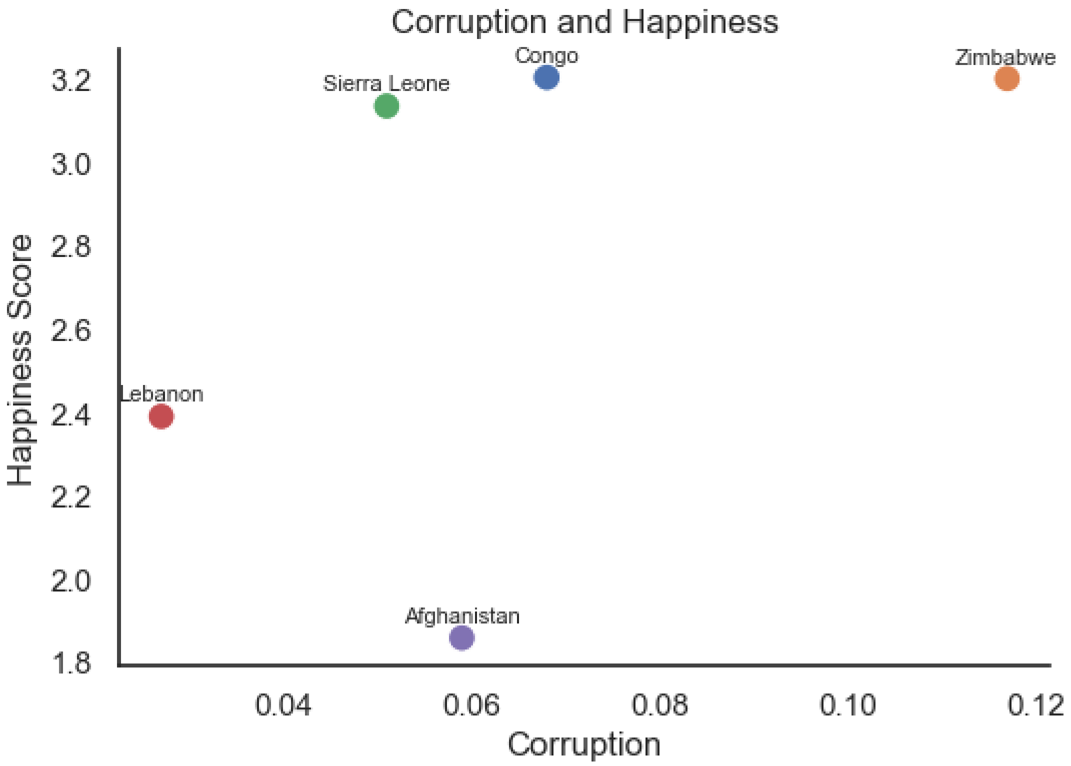
```python
# Create a DataFrame to store the total happiness scores for each country
total_scores_df = result_df.groupby('country')['happiness_score'].sum().rese

# Sort the DataFrame by total happiness scores in ascending order
total_scores_df = total_scores_df.sort_values(by='happiness_score', ascendi

# Get the 5 countries with the lowest total happiness scores
lowest_5_countries = total_scores_df[['country', 'happiness_score']].head(5

print("Top 5 Countries with the Lowest Total Happiness Scores:")
for index, row in lowest_5_countries.iterrows():
    print(f"{row['country']} - Total Happiness Score: {row['happiness_score
Top 5 Countries with the Lowest Total Happiness Scores:
Afghanistan - Total Happiness Score: 1.859
Lebanon - Total Happiness Score: 2.392
Sierra Leone - Total Happiness Score: 3.138
Zimbabwe - Total Happiness Score: 3.204
Congo (Kinshasa) - Total Happiness Score: 3.207
```

Indeed, common denominators among these countries may involve factors like political instability, conflict, economic challenges, and social issues, as you mentioned. Corruption is also a critical factor to consider. To investigate further, let's explore the potential relationship between happiness scores and corruption using the available dataset.

In [47]:

```python
# Filter the DataFrame for the selected countries
selected_countries = ['Afghanistan', 'Lebanon', 'Sierra Leone', 'Zimbabwe',
filtered_df = df[df['country'].isin(selected_countries)]

# Shorten the "Congo (Kinshasa)" label
filtered_df['country'] = filtered_df['country'].replace('Congo (Kinshasa)',

# Select the relevant columns
columns_to_analyze = ['happiness_score', 'social_support', 'perceptions_of_c

# Create a scatterplot
plt.figure(figsize=(6, 4))
sns.set_style("white")

# Plot corruption vs. happiness score
ax = sns.scatterplot(data=filtered_df, x='perceptions_of_corruption', y='hap

# Set axis labels and title
plt.xlabel("Corruption")
plt.ylabel("Happiness Score")
plt.title("Corruption and Happiness")

# Add country labels above data points
for i in range(len(filtered_df)):
    x_offset = 0 if filtered_df['country'].iloc[i] == 'Afghanistan' else 0.0
    ax.text(filtered_df['perceptions_of_corruption'].iloc[i] + x_offset, fi

sns.despine()

# Remove the legend
ax.get_legend().remove()

# Show the plot
plt.show()
```

```
/var/folders/gf/945vyt3n2m3flxbfw360lmb40000gn/T/ipykernel_5152/2840154766.
py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['country'] = filtered_df['country'].replace('Congo (Kinshas
a)', 'Congo')
```



While examining the relationship between corruption and happiness score, it becomes evident that there is no straightforward correlation. The scatterplot shows a scattered distribution of data points, indicating that the level of corruption doesn't consistently align with happiness scores across the selected countries. However, it's important to note that this dataset might not offer a complete picture of the ground reality. High corruption levels in some countries could significantly impact the accuracy of the reported data, making it challenging to draw definitive conclusions about the connection between corruption and happiness. This highlights the need for cautious interpretation and raises questions about the reliability of the dataset.

Conclusion:

Top 5 Countries with the Highest Total Happiness Scores:

Finland - Total Happiness Score: 7.804

Denmark - Total Happiness Score: 7.586

Iceland - Total Happiness Score: 7.530

Israel - Total Happiness Score: 7.473

Netherlands - Total Happiness Score: 7.403

Possible Reasons for Their High Happiness Scores:

These countries may benefit from strong social support systems, promoting overall well-being. High GDP per capita may provide residents with economic stability and access to essential resources. Quality healthcare and healthy life expectancy contribute to a high happiness index.

Regional Happiness Trends:

The highest average happiness scores are observed in Northern America, ANZ (Australia and New Zealand), and Western Europe. Ukraine's happiness index has stagnated and even decreased since the onset of the war, likely due to the ongoing conflict's impact on well-being.

Countries with the Lowest Happiness Scores:

Afghanistan, Lebanon, Sierra Leone, Zimbabwe, and Congo have consistently low happiness scores.

Possible Reasons for Their Low Happiness Scores:

These countries may face a combination of factors, such as political instability, economic challenges, and inadequate social support systems.

Correlation Between Happiness and Generosity:

No visible correlation between happiness and generosity is evident in the dataset, suggesting that generosity alone does not drive happiness.

Correlation Between Happiness and Corruption:

Similarly, there is no clear correlation between happiness and corruption based on the dataset's information. The dataset's reliability may be impacted by corruption in some countries, making it challenging to draw definitive conclusions. In conclusion, these findings shed light on the complex interplay of various factors affecting happiness. While strong correlations are observed between happiness, GDP, social support, and healthy life expectancy, other factors like generosity and corruption do not exhibit clear relationships. It's important to consider the unique circumstances of each country and be cautious in interpreting the data, as corruption may affect the accuracy of reported happiness scores. Thank you for your time and for reviewing this analysis.