Happiness over the years: exploratory analysis of happiness scores of different countries

Ilef Mhamdi

Alanya University, Faculty of Engineering and Natural Sciences
Computer Engineering Department

TE 1414 Introduction to Data Science, Spring 2024–2025
May 7, 2025

1. Introduction

The World Happiness Report dataset is an annual collection of happiness scores and related socioeconomic indicators for countries around the world. Understanding the things that make people happy has important implications for both researchers and policymakers who want to make the world a better place to live in. This project analyzes happiness scores from 2015 to 2019, focusing on how they differ between countries, geographic regions, and years. The main goals of the project are to clean and merge multi-year data together, examine a geographic region variable, conduct exploratory data analysis (EDA), and create visualizations that show trends and disparities in happiness at a global level. The analysis will look for insights and trends that can inform our understanding of the socioeconomic determinants of happiness as they change over time.

2. Methodology / Project Design

The dataset was obtained from the World Happiness Report series published by Kaggle over the years 2015 through 2019. The dataset was analyzed using Python, with functions from Pandas to manipulate data and functions from Seaborn and Matplotlib for static visualizations. The interactive plots were made using Plotly Express and PyCountry-Convert to plot geographic data.

The steps for the processing and analysis of the data were as follows: First, the annual CSV files were downloaded programmatically. Due to schema differences between years, such as a lack of standardized field names occurring, suggesting “Happiness Score” vs. “Score” and “Family” vs. “Social support”,a mapping dictionary was created to standardize the names of each dataset’s file to present as one. After successfully framing each collection, the annual data was merged into one DataFrame.

The data had to be cleaned of leading spaces in the country name to maintain standardization across iterations, and the 'Year' variable was converted to datetime so a reference for temporal analysis was preserved. After formatting, rows with missing target values (Happiness_Score) were dropped. When other features contained missing values, the median feature value was estimated with individual yearly data in order to create complete datasets to minimize bias and avoid changing reality.

A new 'Region' feature was created by mapping countries to their respective continents using the pycountry_convert library. This feature allowed for further comparisons and analyses that incorporate regional factors.

Descriptive statistics were computed for happiness scores and key features (mean, median and standard deviation). Potential explanatory features were examined in various visualizations (e.g. histograms, scatter plots, correlation heatmaps, boxplots, and line plots) that compared distributions, relationships, and time trends, thus ensuring a standard approach to the analysis annually and globally.

3. Implementation / Execution

The CSV file for each year was individually loaded through pandas from 2015 to 2019. Due to the column naming conventions changing across years, columns were changed to the same schema with a dictionary mapping names. After renaming, all row values from yearly DataFrames were appended to a single DataFrame having all records. Additionally in country names, white spaces were removed to avoid discrepancies while joining and mapping. The 'Year' column was transformed to a datetime type to carry out any time based analysis. Rows that have the target variable (Happiness_Score) missing should also be removed to maintain integrity in the data. For the other features, missing values are imputed with the median for each feature and each year, this reduced the effect of outliers in any data analysis.

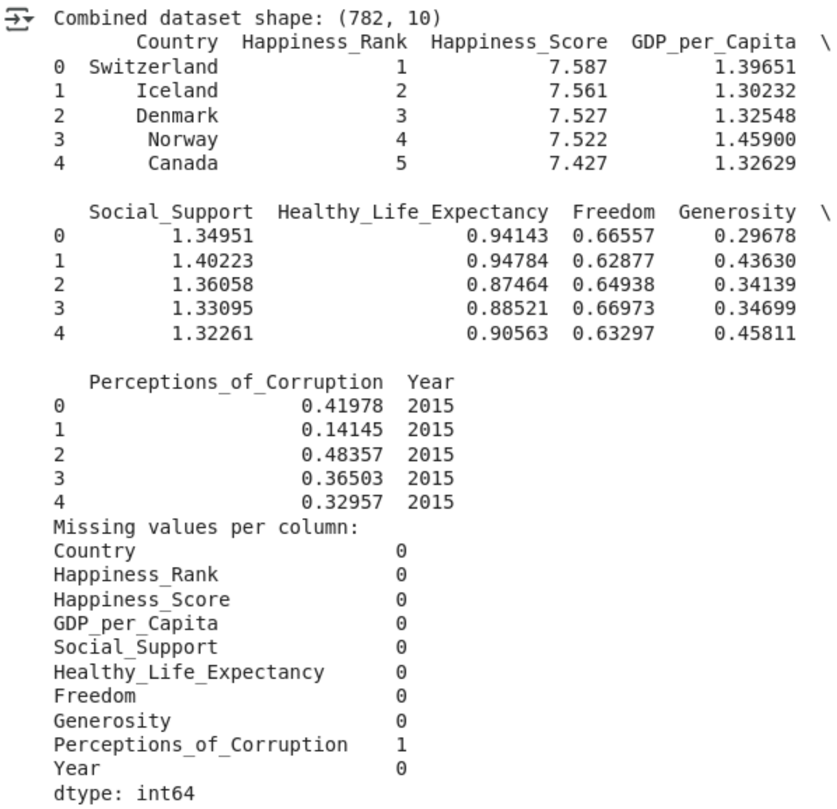
A 'Region' column was produced by mapping countries to continents using the pycountry_convert library, which enabled more group and exploratory analysis by region.

Exploratory data analysis was conducted by calculating descriptive statistics, including the mean, median, and standard deviation for happiness scores and other relevant variables. Descriptive histograms were created to examine the distribution of happiness scores. The histogram indicated an approximately normal distribution with some right skew. Scatter plots were created to visualize the relationship between GDP per capita and happiness scores. The scatter plots demonstrated positive correlations between the two variables and regional clustering. Correlation heatmaps, which illustrated the relationship strength among the different variables, were created. The strongest association was between happiness score and GDP per capita, social support, and healthy life expectancy. Boxplots were created to visualize the distribution of happiness scores by region. The boxplots of happiness scores demonstrated that Europe and North America were regions that tended to report higher median happiness scores. Line plots summarized the average happiness score for each region over time. The line plots indicated that the average happiness scores tended to be relatively stable over time for each region, while some variation occurred between different regions from 2015 to 2019 (e.g., South America and Europe)

4. Results and Analysis

Dataset Overview

After merging and cleaning, the combined dataset had 782 rows and 10 important columns. Some features still had missing values but we were able to fill those by using median values in each year.



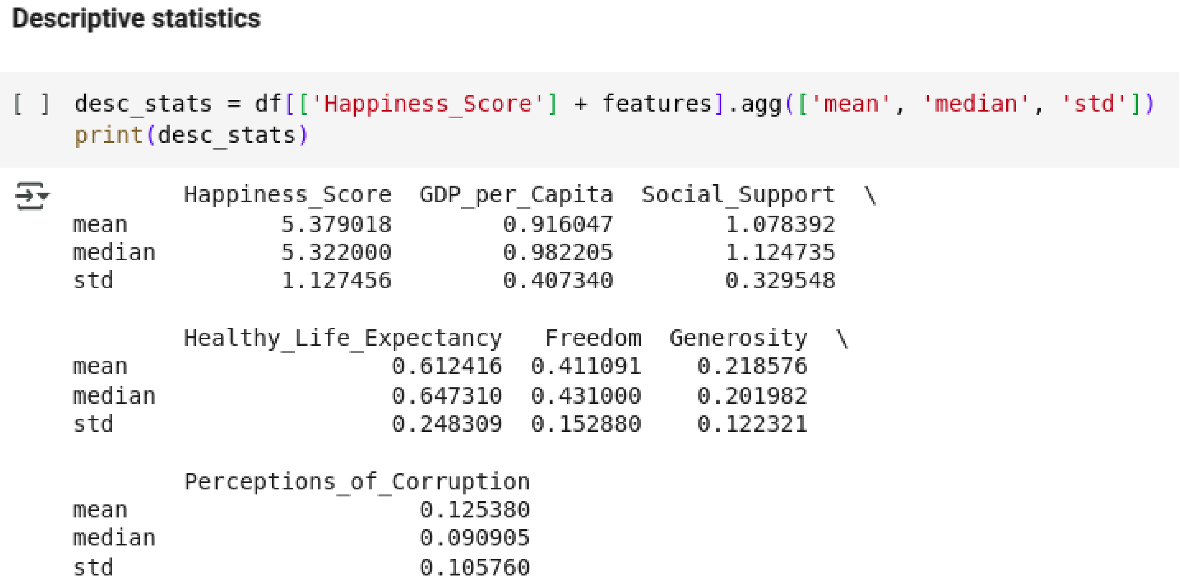
```
Combined dataset shape: (782, 10)
       Country  Happiness_Rank  Happiness_Score  GDP_per_Capita  \
0  Switzerland               1            7.587         1.39651   
1      Iceland               2            7.561         1.30232   
2      Denmark               3            7.527         1.32548   
3       Norway               4            7.522         1.45900   
4       Canada               5            7.427         1.32629   

   Social_Support  Healthy_Life_Expectancy  Freedom  Generosity  \
0         1.34951                  0.94143  0.66557     0.29678   
1         1.40223                  0.94784  0.62877     0.43630   
2         1.36058                  0.87464  0.64938     0.34139   
3         1.33095                  0.88521  0.66973     0.34699   
4         1.32261                  0.90563  0.63297     0.45811   

   Perceptions_of_Corruption  Year  
0                    0.41978  2015  
1                    0.14145  2015  
2                    0.48357  2015  
3                    0.36503  2015  
4                    0.32957  2015  
Missing values per column:
Country                      0
Happiness_Rank               0
Happiness_Score              0
GDP_per_Capita               0
Social_Support               0
Healthy_Life_Expectancy      0
Freedom                      0
Generosity                   0
Perceptions_of_Corruption    1
Year                         0
dtype: int64
```

Descriptive Statistics

Descriptive Statistics of Happiness Score and Features



Descriptive statistics

```
[ ] desc_stats = df[['Happiness_Score'] + features].agg(['mean', 'median', 'std'])
    print(desc_stats)
```

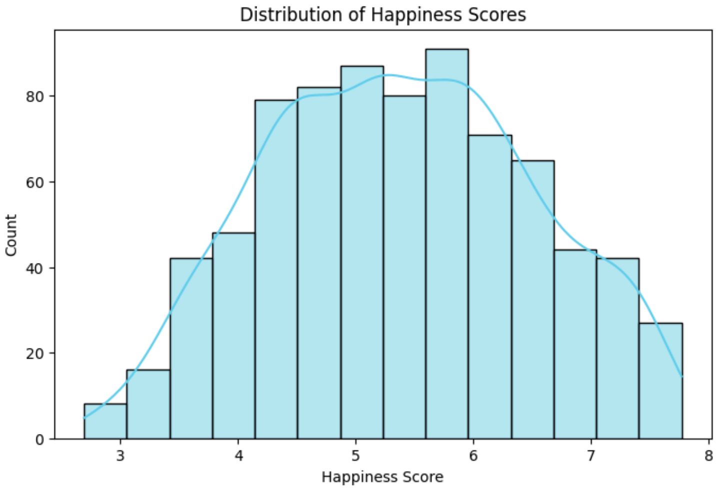
```
        Happiness_Score  GDP_per_Capita  Social_Support  \
mean           5.379018        0.916047        1.078392   
median         5.322000        0.982205        1.124735   
std            1.127456        0.407340        0.329548   

        Healthy_Life_Expectancy   Freedom  Generosity  \
mean                   0.612416  0.411091    0.218576   
median                 0.647310  0.431000    0.201982   
std                    0.248309  0.152880    0.122321   

        Perceptions_of_Corruption  
mean                     0.125380  
median                   0.090905  
std                      0.105760  
```

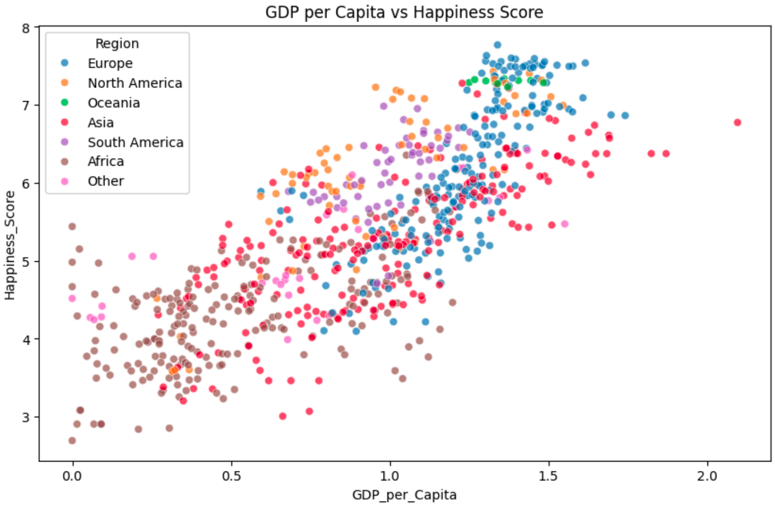
Visualizations

Figure 1: Distribution of Happiness Scores (Histogram)



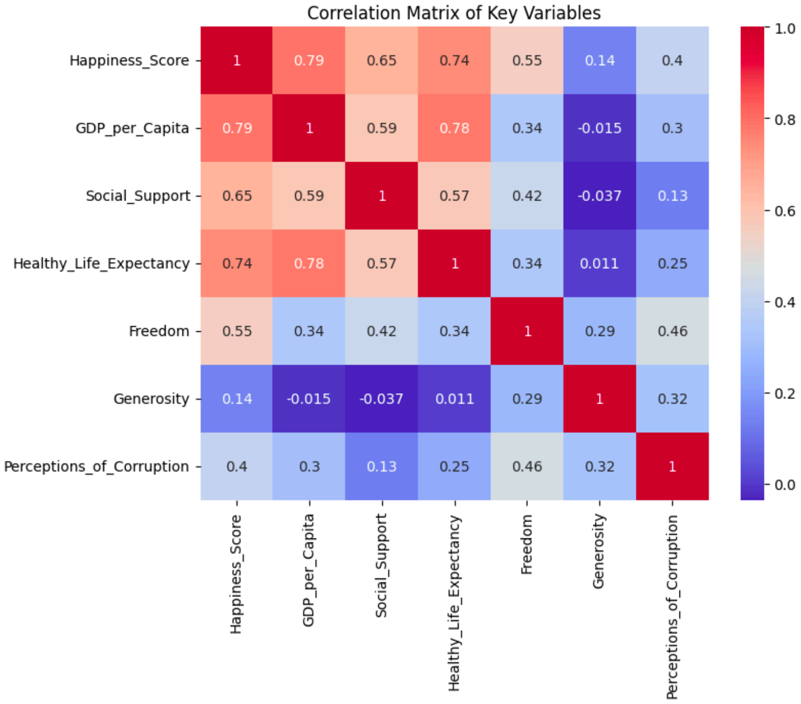
- ➔ The histogram of happiness score suggests that the distribution resembles a normal distribution with a slight skew towards higher values. It is reasonable to conclude that most countries are centered around an average happiness score, but some countries had very high happiness scores.

Figure 2: GDP per Capita vs Happiness Score by Region (Scatter Plot)



- ➔ The scatter plot of GDP per capita versus happiness score represents a strong positive relationship, indicating that overall, higher economic prosperity is related to happiness. There are some regional clusters present in the data since Europe and North America tend to have higher scores of both measures.

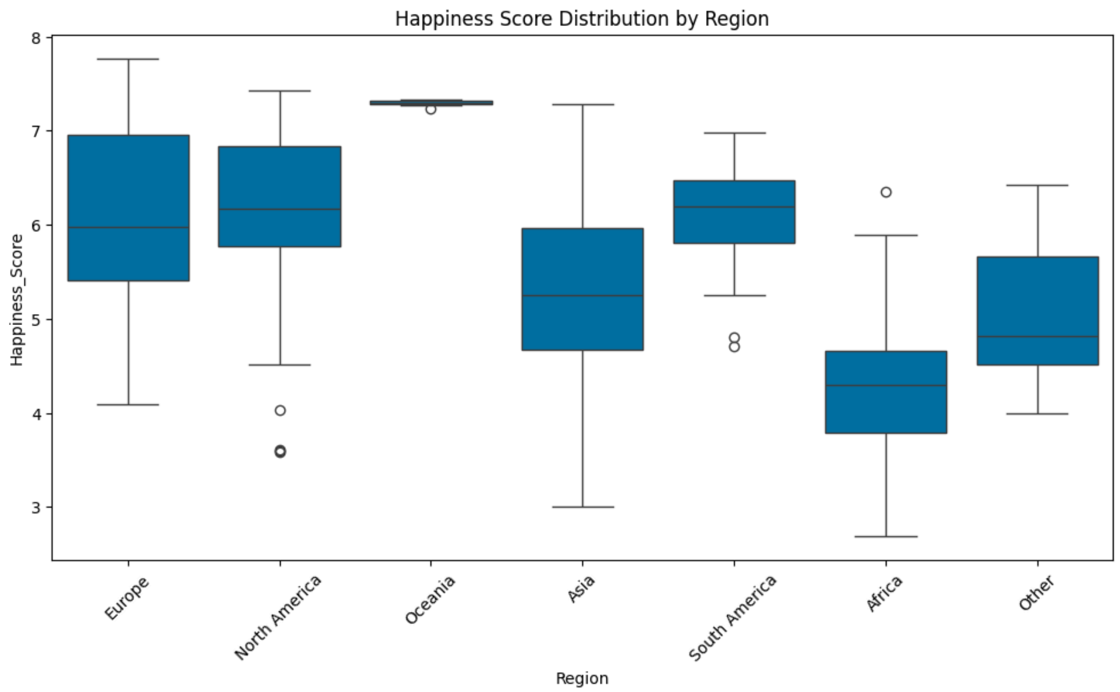
Figure 3: Correlation Matrix of Key Variables (Heatmap)



	Happiness_Score	GDP_per_Capita	Social_Support	Healthy_Life_Expectancy	Freedom	Generosity	Perceptions_of_Corruption
Happiness_Score	1	0.79	0.65	0.74	0.55	0.14	0.4
GDP_per_Capita	0.79	1	0.59	0.78	0.34	-0.015	0.3
Social_Support	0.65	0.59	1	0.57	0.42	-0.037	0.13
Healthy_Life_Expectancy	0.74	0.78	0.57	1	0.34	0.011	0.25
Freedom	0.55	0.34	0.42	0.34	1	0.29	0.46
Generosity	0.14	-0.015	-0.037	0.011	0.29	1	0.32
Perceptions_of_Corruption	0.4	0.3	0.13	0.25	0.46	0.32	1

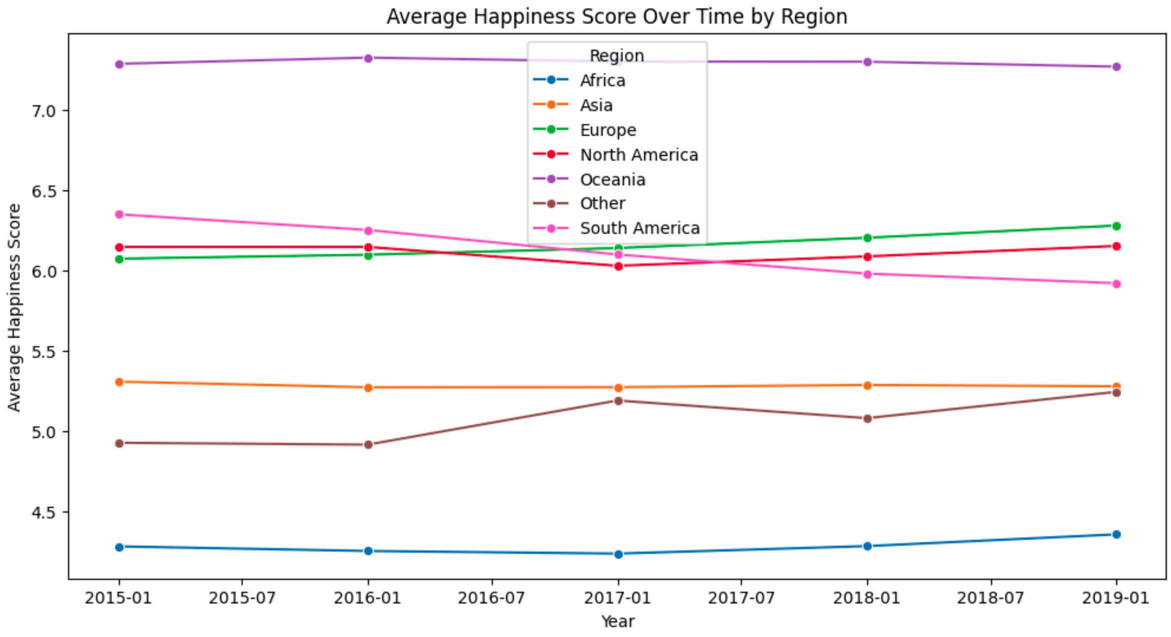
- ➔ The correlation heatmap shows indicators that have the strongest positive correlation to happiness score such as GDP per capita (0.79), social support (0.65), and healthy life expectancy (0.74), signifying the multifaceted representations of what contributes to the measure of happiness.

Figure 4: Happiness Score Distribution by Region (Boxplot)



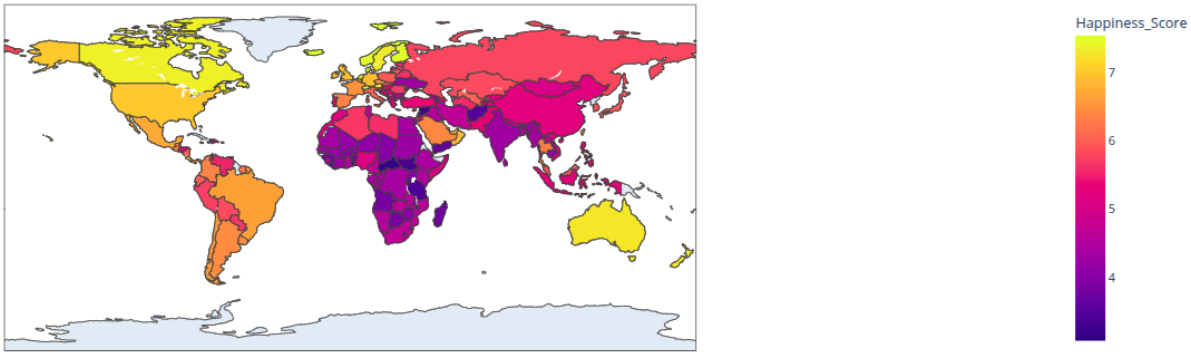
- ➔ The boxplots of happiness score by region demonstrate that Europe and North America have higher median happiness scores than other regions, including Africa and Asia.

Figure 5: Average Happiness Score Over Time by Region (Line Plot)



➔ The average happiness score over time by region was visualized in a line plot which shows some relative stability in happiness score values from 2015 to 2019, but within certain regions (Europe, South America), scores appear to fluctuate, suggesting the temporal aspects of well-being.

Map Visualization:



➔ The choropleth map provides a global perspective on happiness distribution. Countries with warmer colors (yellow/white) indicate higher happiness scores, while cooler colors (purple) represent lower scores. From the map, we can observe:

Scandinavian countries consistently show the highest happiness scores, North America and Europe appear happier than other regions, and Africa shows the lowest happiness scores

There's a clear regional pattern where developed nations tend to be happier than developing ones

5. Conclusion

This project successfully merged and standardized multi-year World Happiness data and developed a regional feature to enable holistic exploratory analysis. The insights underscored the considerable role of economic prosperity and social support in determining happiness while there are also important regional-spatial and temporal variations.

The analytical methods followed suited the analysis, while appropriately identifying data inconsistencies and addressing missing values. Duplicate rows were also checked using \`df.duplicated().sum()\` and removed with \`df.drop_duplicates()\` to ensure data integrity. Countries were mapped to continents using \`pycountry_convert\`, enabling regional analysis. Additionally, this project has established a strong base for further modeling efforts or policy assessments designed to enhance global welfare.

6. References

World Happiness Report Dataset, Kaggle, 2015–2019.

Data Science Practical examples file

PyCountry-Convert Library Documentation, https://pypi.org/project/pycountry-convert/ Waskom, M.

7. Appendix

https://colab.research.google.com/drive/1BK6SfL15MVHX2OW1ck0eU4VQ6jpwt5qt?usp=sharing

World Happiness Final Project

Installing the packages

```
!pip install kagglehub pycountry_convert --quiet
import kagglehub
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import pycountry_convert as pc
```

Download dataset

```
path = kagglehub.dataset_download("unsdsn/world-happiness")
print("Dataset files path:", path)
# Files available (2015-2019)
available_years = [2015, 2016, 2017, 2018, 2019]
# Defining  and unifying consistent columns that I wanna keep across years (different years have
different column names, so I mapped them)
column_mappings = {
  2015: {
      'Country': 'Country',
      'Happiness Rank': 'Happiness_Rank',
      'Happiness Score': 'Happiness_Score',
      'Economy (GDP per Capita)': 'GDP_per_Capita',
      'Family': 'Social_Support',
      'Health (Life Expectancy)': 'Healthy_Life_Expectancy',
      'Freedom': 'Freedom',
      'Generosity': 'Generosity',
      'Trust (Government Corruption)': 'Perceptions_of_Corruption'
  },
  2016: {
      'Country': 'Country',
      'Happiness Rank': 'Happiness_Rank',
      'Happiness Score': 'Happiness_Score',
      'Economy (GDP per Capita)': 'GDP_per_Capita',
      'Family': 'Social_Support',
      'Health (Life Expectancy)': 'Healthy_Life_Expectancy',
      'Freedom': 'Freedom',
      'Generosity': 'Generosity',
      'Trust (Government Corruption)': 'Perceptions_of_Corruption'
  },
  2017: {
      'Country': 'Country',
      'Happiness.Rank': 'Happiness_Rank',
      'Happiness.Score': 'Happiness_Score',
      'Economy..GDP.per.Capita.': 'GDP_per_Capita',
      'Family': 'Social_Support',
      'Health..Life.Expectancy.': 'Healthy_Life_Expectancy',
      'Freedom': 'Freedom',
      'Generosity': 'Generosity',
      'Trust..Government.Corruption.': 'Perceptions_of_Corruption'
  },
  2018: {
      'Country or region': 'Country',
      'Overall rank': 'Happiness_Rank',
      'Score': 'Happiness_Score',
      'GDP per capita': 'GDP_per_Capita',
      'Social support': 'Social_Support',
      'Healthy life expectancy': 'Healthy_Life_Expectancy',
      'Freedom to make life choices': 'Freedom',
      'Generosity': 'Generosity',
      'Perceptions of corruption': 'Perceptions_of_Corruption'
  },
  2019: {
      'Country or region': 'Country',
      'Overall rank': 'Happiness_Rank',
      'Score': 'Happiness_Score',
      'GDP per capita': 'GDP_per_Capita',
      'Social support': 'Social_Support',
      'Healthy life expectancy': 'Healthy_Life_Expectancy',
      'Freedom to make life choices': 'Freedom',
      'Generosity': 'Generosity',
      'Perceptions of corruption': 'Perceptions_of_Corruption'
  }
}
```

Loading and unifying datasets

```
dfs = []
for year in available_years:
    fname = f"{year}.csv"
    fpath = os.path.join(path, fname)
    if os.path.exists(fpath):
        df_year = pd.read_csv(fpath)
        # Renaming columns according to mapping for the year
        df_year = df_year.rename(columns=column_mappings[year])
        # to keep only the columns that I want (some years may have extra columns)
        cols_to_keep = list(column_mappings[year].values())
        df_year = df_year[cols_to_keep]
        df_year['Year'] = year
        dfs.append(df_year)
    else:
        print(f"File not found: {fpath}")
```

```
# Concatenating all years
df = pd.concat(dfs, ignore_index=True)
```

Data cleaning and ensuring consistency (handling also the missing values)

```
# Showing the shape and sample after combination of the database
print(f"Combined dataset shape: {df.shape}")
print(df.head())
# Striping whitespace from country names for the cleaning of the data process
df['Country'] = df['Country'].str.strip()
# Converting Year to datetime
df['Year'] = pd.to_datetime(df['Year'], format='%Y')
# Checking for missing values
print("Missing values per column:")
print(df.isnull().sum())
# Droping rows missing Happiness_Score (target)
df = df.dropna(subset=['Happiness_Score'])
# Imputing the missing values in features with median per year
features = ['GDP_per_Capita', 'Social_Support', 'Healthy_Life_Expectancy', 'Freedom', 'Generosity',
'Perceptions_of_Corruption']
for col in features:
    if col in df.columns:
        df[col] = df.groupby('Year')[col].transform(lambda x: x.fillna(x.median()))
```

Creating Geographic Region feature

```
def country_to_continent(country):
    try:
        code = pc.country_name_to_country_alpha2(country)
        continent_code = pc.country_alpha2_to_continent_code(code)
        continent_name = pc.convert_continent_code_to_continent_name(continent_code)
        return continent_name
    except:
        return 'Other'
df['Region'] = df['Country'].apply(country_to_continent)
```

Descriptive statistics

```
desc_stats = df[['Happiness_Score'] + features].agg(['mean', 'median', 'std'])
print(desc_stats)
```

Visualizations

```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
plt.figure(figsize=(8,5))
sns.histplot(df['Happiness_Score'], kde=True, color='skyblue')
plt.title('Distribution of Happiness Scores')
plt.xlabel('Happiness Score')
plt.ylabel('Count')
plt.show()
plt.figure(figsize=(10,6))
sns.scatterplot(data=df, x='GDP_per_Capita', y='Happiness_Score', hue='Region', alpha=0.7)
plt.title('GDP per Capita vs Happiness Score')
plt.show()
```

Correlation heatmap

```
corr = df[['Happiness_Score'] + features].corr()
plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Key Variables')
plt.show()
```

Boxplot of Happiness Score by Region

```
plt.figure(figsize=(12,6))
sns.boxplot(x='Region', y='Happiness_Score', data=df)
plt.title('Happiness Score Distribution by Region')
plt.xticks(rotation=45)
plt.show()
```

World Map of Average Happiness Scores by Country (2015–2019)

```
# Map Visualization of Average Happiness Scores by Country
fig = px.choropleth(df.groupby('Country')['Happiness_Score'].mean().reset_index(),
                    locations="Country",
                    locationmode='country names',
                    color="Happiness_Score",
                    hover_name="Country",
                    color_continuous_scale=px.colors.sequential.Plasma,
                    title="World Happiness Scores by Country (2015-2019 Average)")
fig.show()
```

Line Plot of Average Happiness Score Over Time by Region

```
df_region_year = df.groupby(['Region', 'Year'])['Happiness_Score'].mean().reset_index()
plt.figure(figsize=(12,6))
sns.lineplot(data=df_region_year, x='Year', y='Happiness_Score', hue='Region', marker='o')
plt.title('Average Happiness Score Over Time by Region')
plt.ylabel('Average Happiness Score')
plt.show()
```