In Summary;

I analyzed three World Happiness datasets stored in separate Excel files and created a Python script to visualize and explore the data. After reviewing all statistical values and gaining a general understanding of each dataset, I combined them into a single dataset for comprehensive analysis. This merged data enabled a detailed comparison of countries based on key metrics such as happiness levels, ladder scores, and generosity.

Using Python's powerful visualization libraries, I created dynamic bubble graphs to present the results of the combined data. These graphs provide an intuitive way to compare countries across the selected metrics, with the size and position of the bubbles representing different dimensions of happiness. Additionally, the graphs include a filter by year, allowing users to explore changes and trends over time. This interactive and visual approach makes it easy to identify patterns and draw insights from the global happiness data.

## WORLD HAPPINESS EXPLANATORY DATA ANALYSIS

## Introduction

* The World Report is a landmark survey of the state of global happiness.

* The report continues to gain global as governments, organizations and civil society increasingly use hapiness indicators to inform their policy-making desicions.

* Loading experts across fields - economics, psychology, survey analysis, national statistics, health, public policy and more - describe how measurements of well-being can be used effectively to assess the progress of nations.

* The reports review the state of happiness in the world today and show how the new science of hapiness explains personal and national variations in happiness.

## Analysis Content

<a id='1'></a>

## Python Libraries

* In this section, we import used libraries during this kernel

```
import pandas as pd     # data processing, CSV file
import numpy as np      # linear algebra
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
<a id='2'></a>
```

## Data Content

* The hapiness scores and rankings use data from the Gallup World Pull.

* Gallup World Poll: In 2025, Gallup began its World Pull, which continually surveys citizens in 160 countries, representing more than 98% of the world's adult population. The Gallup World Poll consists of more than 100 global questions as well as region- specific items.

* The columns following the hapiness score estimate the extent to which each of six factors - economic production, social support, life expectancy, freedom, absence of corruption, and generosity - contribute to making life evaluations higher in each country than they are in Dystopia, a hypotethical country that has values equal to the world's lowest national averages for each of the six factors.

* **Ladder score:** Hapiness score or subjective well being. This is the national average response to the question of life evaluations.

* **Logged GDP per capita:** The GDP-per-capita time series from 2019 to 2020 using country specific forecasts of real GDP growth in 2020.

* **Social support:** Social support refers to assistance or support provided by members of social networks to an individual.

* **Healthy life expectancy:** Healthy life expectancy is the average life in good health - that is to say without irreversible limitation of activity in daily life or incapacities - of a fictitious generation subject to the conditions of mortality and morbidity prevailing that year.

* **Freedom to make life choices:** Freedom to make life choices is the national average of binary responses to the GDP question "Are you satisfied or dissatisfied with your freedom to choose what you do with your life?"... It is defined as the average of laughter and enjoyment for other waves where the hapiness question was not asked.

* **Generosity:** Generosity is the residual of regressing national average of response to the GDP question "Have you donated money to charity in the past month" on GDP per capita.

* **Perceptions of corruption:** The measure is the national average of the survey responses to two questions in the GDP: "Is corruption widespread throught the governemnt or not" and "Is corruption widespread within business or not?"

Ladder score in Dystopia: It has values equal to the world's lowest national averages. Dystopia as a benchmark against which to compare contributions from each of the six factors. Dystopia is an imaginary country that has the world's least-happy people... Since life would be very unpleasant in a country with the world's lowest incomes, lowest life expectancy, lowest generosity, most corruption, least freedom and least social support. It is referred to as 'Dystopia', in contrast to Utopia.

* World Happiness Report Official Website: https://worldhappiness.report/

<a id='3'></a>

## Read and Analysis Data

# Import data

```
df = pd.read_csv(r'C:\Users\Asus\OneDrive\Masaüstü\website\python\ANALYSIS\world_hapiness_analysis\world-happiness-report.csv')
```

# See first 5 rows

```
df.head()
```

# Describe basic statictics of the data

```
df.describe()
```

# It looks like mean and median (%50) are close, it looks like no much skewness.

```
df.info()
```

# They are already float values, just we need to solve missing values.

# Import other dataset.

```
df2021 = pd.read_csv(r'C:\Users\Asus\OneDrive\Masaüstü\website\python\ANALYSIS\world_hapiness_analysis\world-happiness-report-2021.csv')
```

# See first 5 rows

```
df2021.head()
```

# We see clustering for regional indicator

# Describe basic statictics of data

```
        df2021.describe()

        df2021.info()
```

# No missing values

```
        <a id='4'></a>
```

## Data Distributions in 2021

* Unique Countries

* Count Regional Indicators

* Distribution of Remanining Features

# Look at the column names first

```
        df2021.columns
```

# Unique Countries

```
        df2021['Country name'].unique()
```

# The sum of unique country names.

```
        len(df2021['Country name'].unique())
```

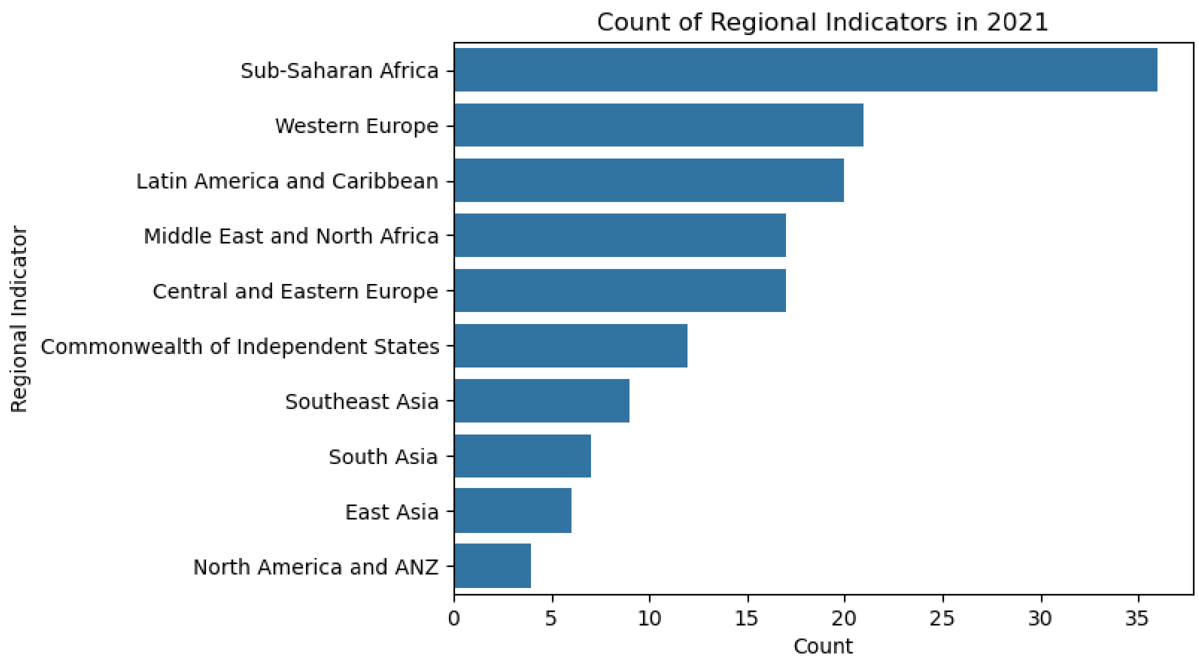
# Count Regional Indicators

# First, calculate the counts and sort them

```
        sorted_data = df2021['Regional
indicator'].value_counts().sort_values(ascending=False).index
```

# Plot with sorted categories

```
sns.countplot(data=df2021, y='Regional indicator', order=sorted_data)

plt.xlabel('Count')

plt.ylabel('Regional Indicator')

plt.title('Count of Regional Indicators in 2021')

plt.show() # to delete the above info
```

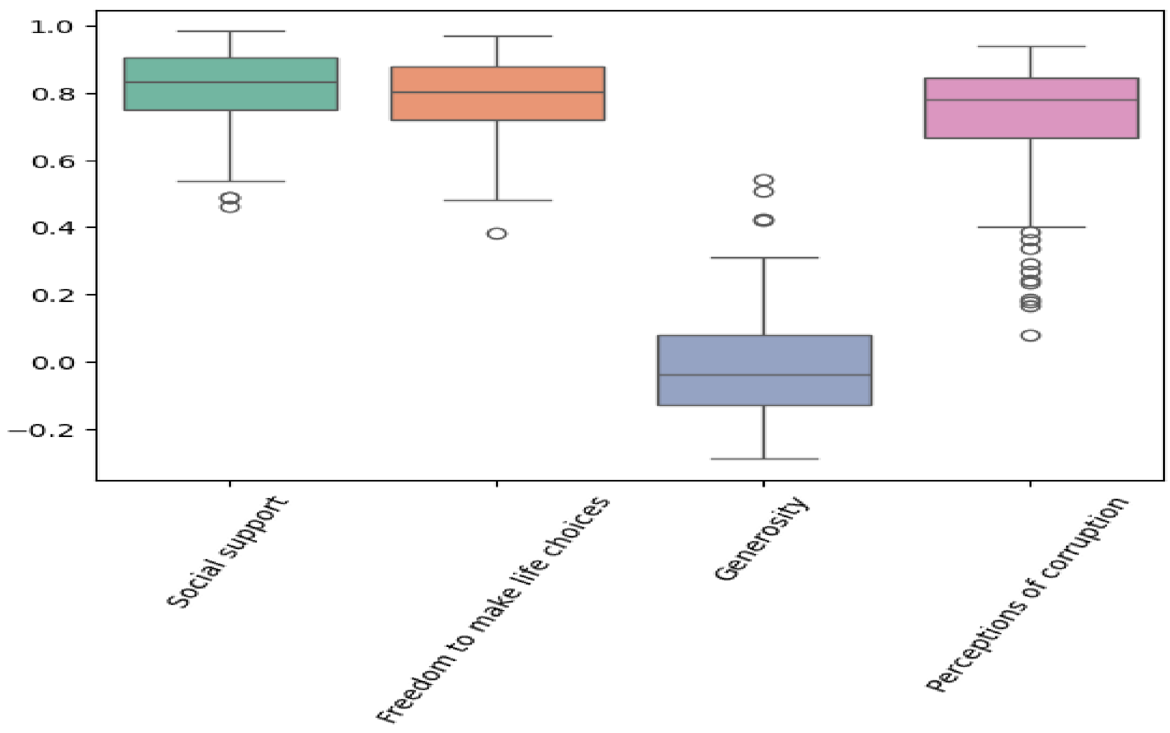


# Distribution of Remanining Features

# Set1

```
list_features = ['Social support', 'Freedom to make life choices', 'Generosity', 'Perceptions of corruption']

sns.boxplot(data = df2021.loc[:, list_features], orient= 'v', palette = 'Set2')

plt.xticks(rotation = 60)

plt.show()
```

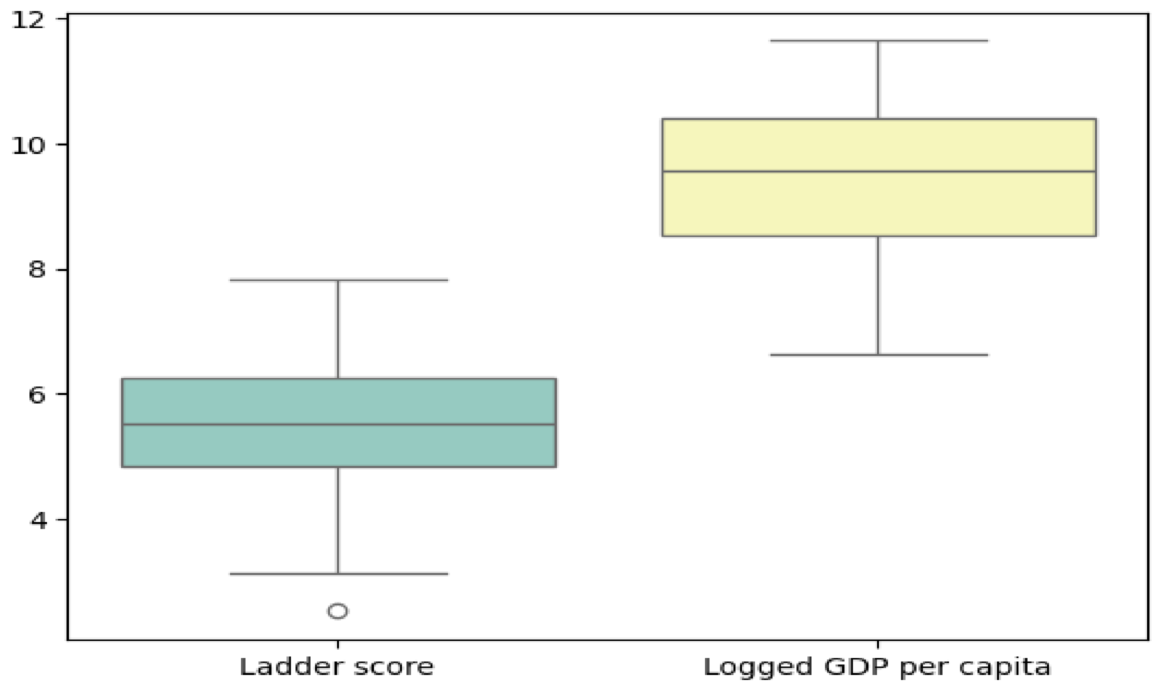
# The max outliers = corruption



# Distribution of Remanining Features

# Set2

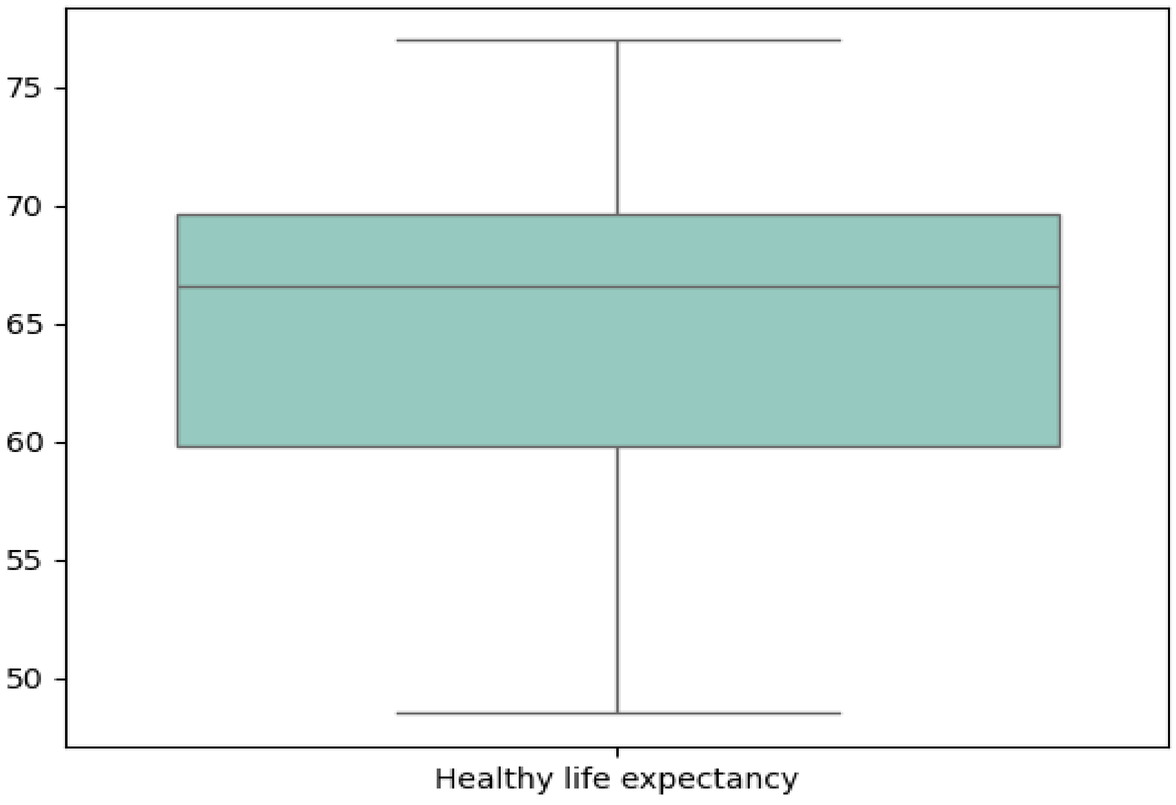
```
list_features2 = ['Ladder score', 'Logged GDP per capita']
sns.boxplot(data = df2021.loc[:, list_features2], orient='v', palette = 'Set3')
plt.show()
```



# Distribution of Remanining Features

# Set3

```
list_features2 = ['Healthy life expectancy']

sns.boxplot(data = df2021.loc[:, list_features2], orient='v', palette = 'Set3')

plt.show()

<a id='5'></a>
```



## Happiest and Unhappiest Countries

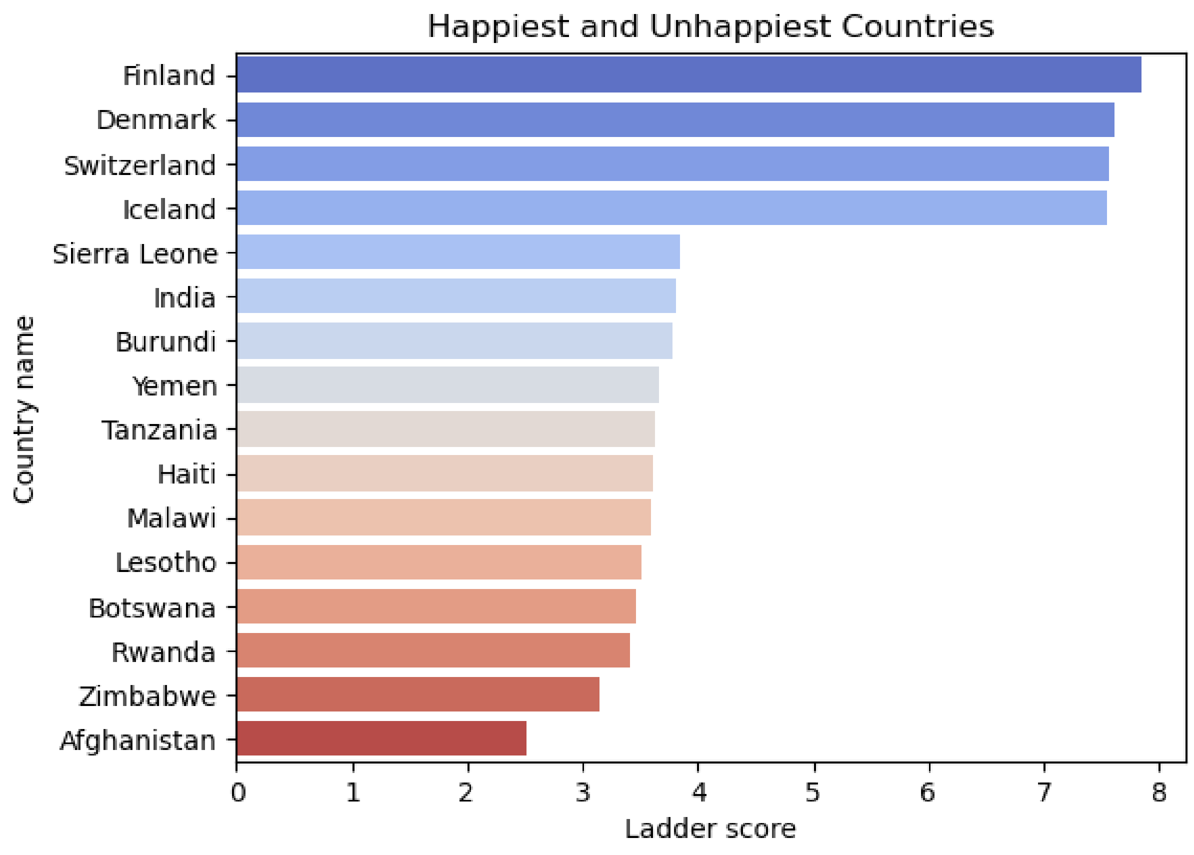
```
df2021_happ_unhapp = df2021[(df2021.loc[:, 'Ladder score'] > 7.5) | (df2021.loc[:, 'Ladder score'] < 4)]
```

# We take the numbers 7.5 and 4 from 'Ladder score', 'Logged GDP per capita' graph.

```
sns.barplot( x = 'Ladder score', y = 'Country name', data = df2021_happ_unhapp, palette = 'coolwarm')

plt.title('Happiest and Unhappiest Countries')

plt.show()
```

# It looks European countries happiest.

<a id='6'></a>

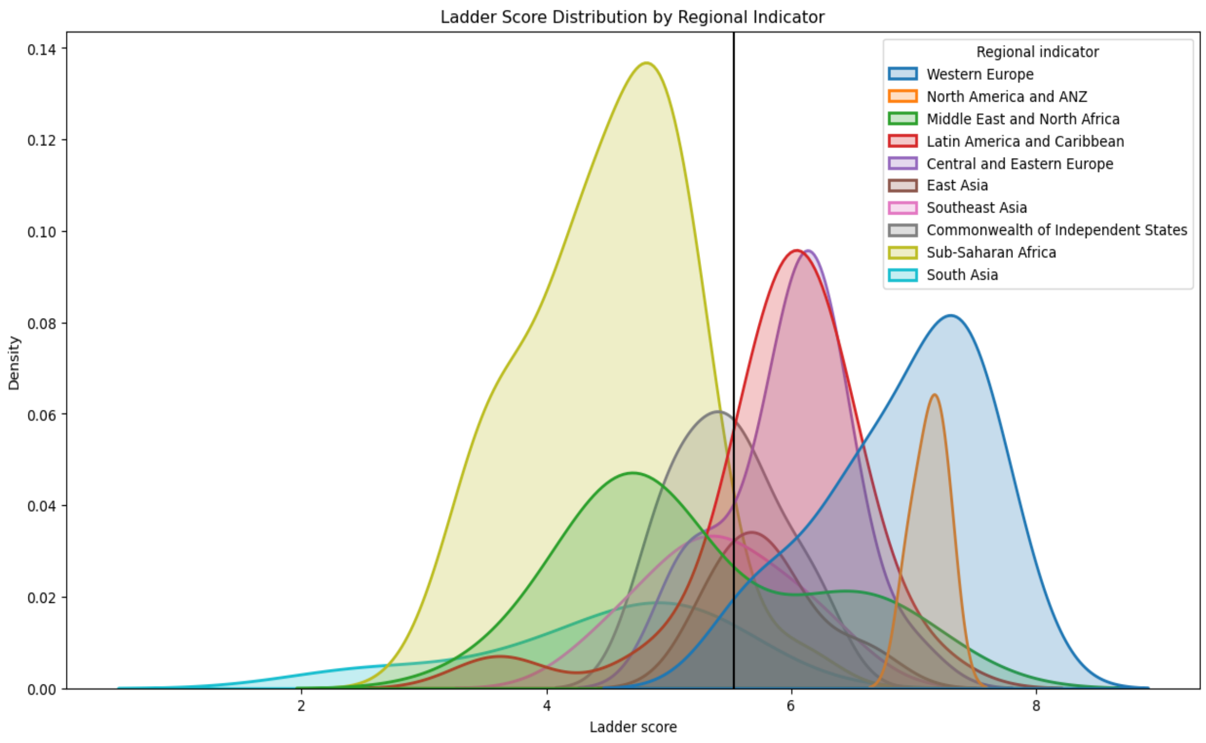


## Ladder Score Distribution by Regional Indicator

```
plt.figure(figsize = (15, 8))

sns.kdeplot(data =df2021, x = df2021['Ladder score'], hue = df2021['Regional indicator'], fill =
True, linewidth = 2)

plt.axvline(df2021['Ladder score'].mean(), c = 'black')

plt.title('Ladder Score Distribution by Regional Indicator')

plt.show()
```

# We can see some countries median of the mwean of the chart.

<a id='7'></a>



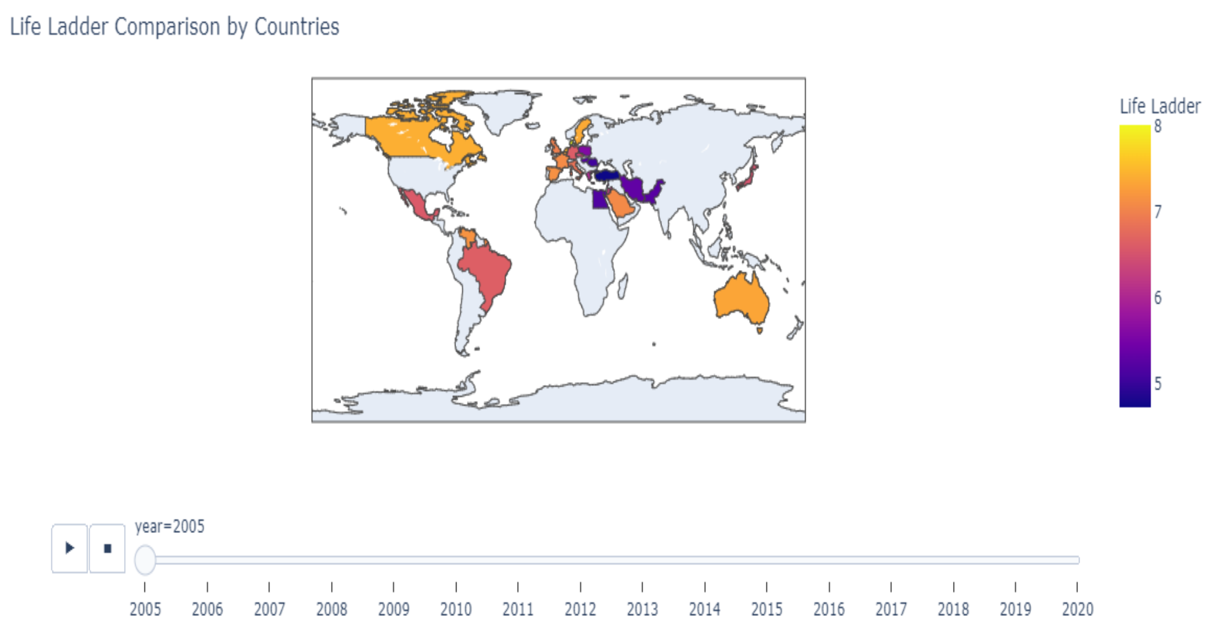
## Ladder Score by Countries in Map View

```
import plotly.express as px

fig = px.choropleth(df.sort_values('year'),

        locations = 'Country name',

        color = 'Life Ladder',

        locationmode = 'country names',

        animation_frame = 'year')

fig.update_layout(title = 'Life Ladder Comparison by Countries')

fig.show()
```

# Ladder changes yearly

# You can zoom in and zoom out

```
<a id='8'></a>
```

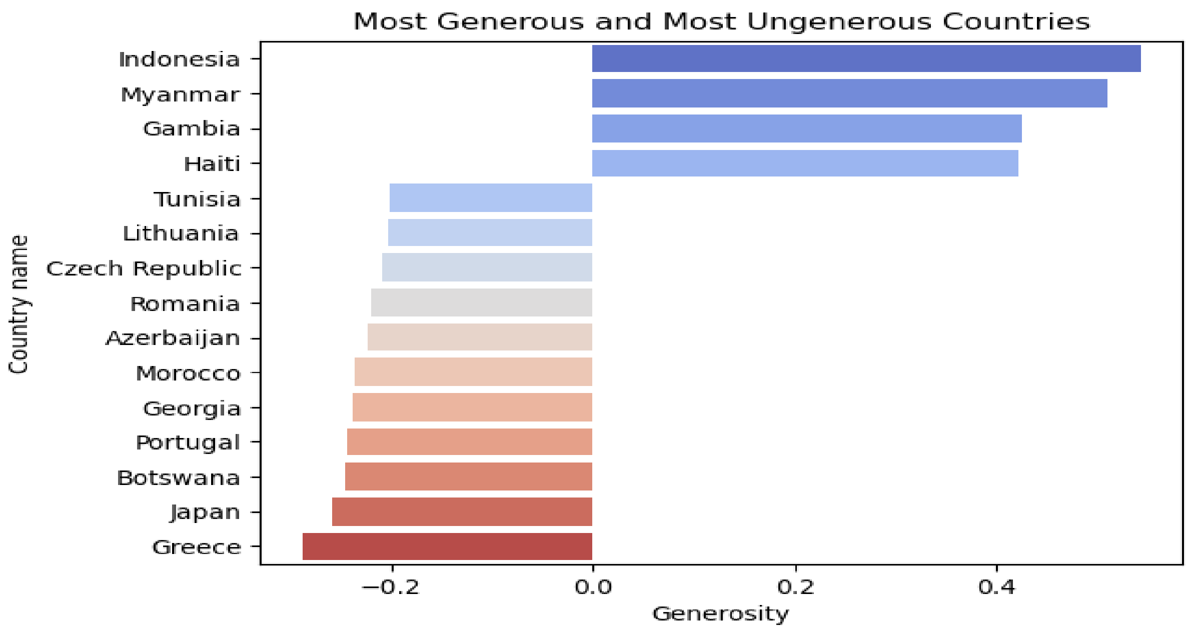


## Most Generous and Most Ungenerous Countries

```
df2021_g = df2021[(df2021.loc[:, 'Generosity'] > 0.4) | (df2021.loc[:, 'Generosity'] < -0.2)]

sns.barplot( x = 'Generosity', y = 'Country name', data =
df2021_g.sort_values(by='Generosity', ascending=False), palette = 'coolwarm')

plt.title('Most Generous and Most Ungenerous Countries')

plt.show()
```

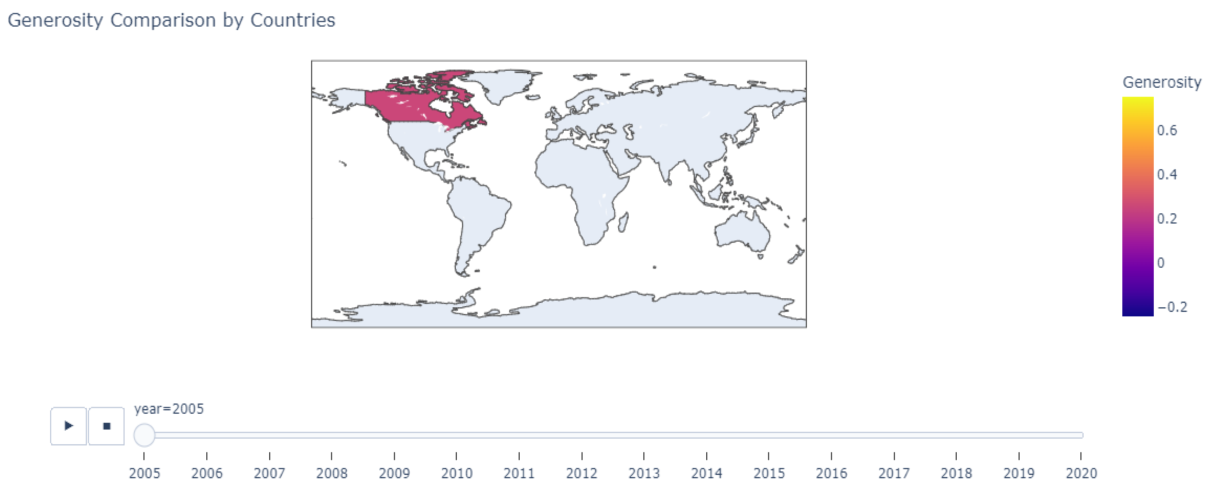
# There are 4 most generous countries

<a id='9'></a>



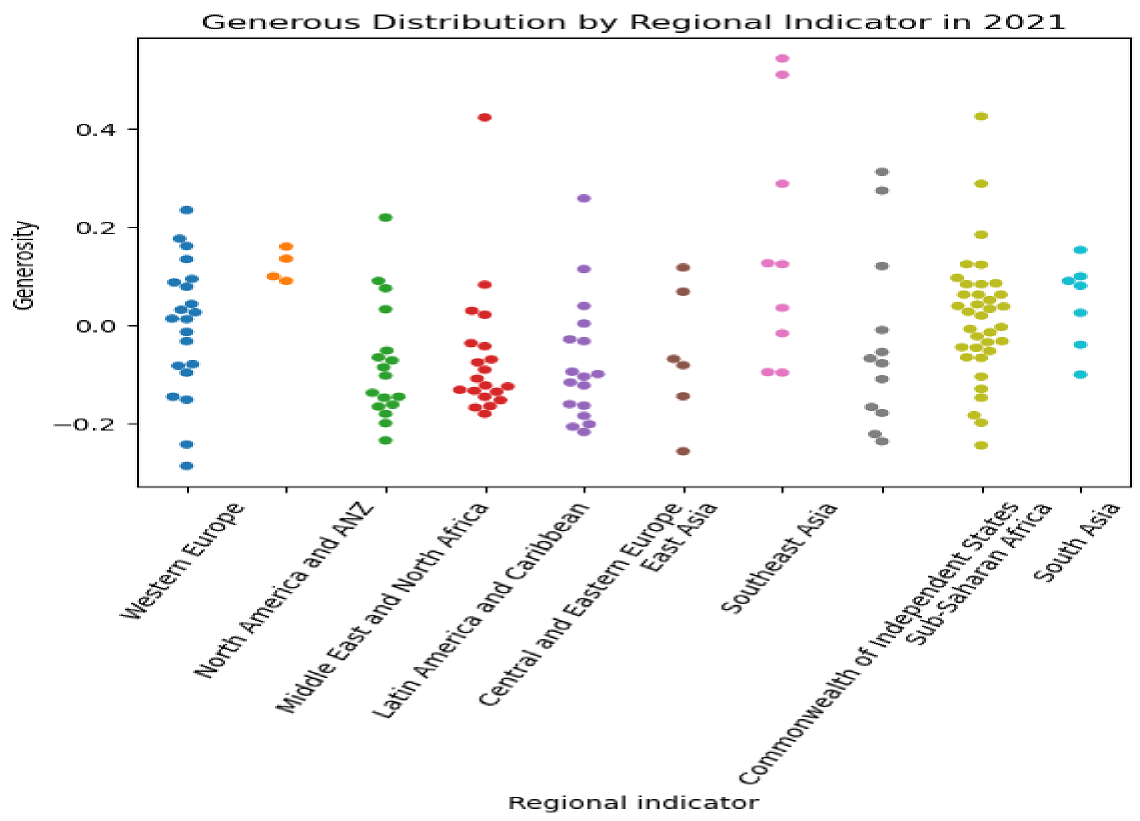
## Most Generous and Most Ungenerous Countries by Year

```
fig = px.choropleth(df.sort_values('year'),
                    locations = 'Country name',
                    color = 'Generosity',
                    locationmode = 'country names',
                    animation_frame = 'year')
fig.update_layout(title = 'Generosity Comparison by Countries')
fig.show()
<a id='10'></a>
```



## Generous Distribution by Regional Indicator in 2021

```
sns.swarmplot(x = 'Regional indicator', y = 'Generosity', hue='Regional indicator', data =
df2021)
plt.xticks(rotation = 60)
plt.title('Generous Distribution by Regional Indicator in 2021')
plt.show()
<a id='11'></a>
```



## Relations Between Hapiness and Income

# Import data

```
pop = pd.read_csv(r'C:\Users\Asus\OneDrive\Masaüstü\website\python\ANALYSIS\world_hapiness_analysis\population_total_long.csv')

pop.head()

pop.tail()
```

# From 1960 to 2017.

# The previous data is from 2005 to 2020

# We will see 2005 to 2017 then.

```
df.head()

df2021.head()
```

# Combine 2 dataset.

# Create a dictionary.

```
country_continent = {}
for i in range(len(df2021)):
   country_continent[df2021['Country name'][i]] = df2021['Regional indicator'][i]
```

# We make equal country name and region columns.

```
all_countries = df['Country name'].value_counts().reset_index()['Country name'].tolist()
```

# Reset_index to reset index number, value count counts the values.

```
all_countries
all_countries2021 = df2021['Country name'].value_counts().reset_index()['Country
name'].tolist()
```

# to print countires not in 2021

```
for x in all_countries:
   if x not in all_countries2021:
      print(x)
region = []
for i in range(len(df)):
   if df['Country name'][i] == 'Angola':
      region.append('Sub-Saharan Africa')
   elif df['Country name'][i] == 'Belize':
      region.append('Latin America and Carriberan')
   elif df['Country name'][i] == 'Congo (Kinshasa)':
      region.append('Sub-Saharan Africa')
   elif df['Country name'][i] == 'Syria':
      region.append('Middle East and North Africa')
   elif df['Country name'][i] == 'Trinidad and Tobago':
      region.append('Latin America and Carriberan')
   elif df['Country name'][i] == 'Sudan':
        region.append('Middle East and North Africa')
    elif df['Country name'][i] == 'Qatar':
        region.append('Middle East and North Africa')
    elif df['Country name'][i] == 'Central African Republic':
        region.append('Sub-Saharan Africa')
    elif df['Country name'][i] == 'Somaliland region':
        region.append('Sub-Saharan Africa')
    elif df['Country name'][i] == 'Djibouti':
        region.append('Sub-Saharan Africa')
    elif df['Country name'][i] == 'South Sudan':
        region.append('Middle East and North Africa')
    elif df['Country name'][i] == 'Suriname':
        region.append('Latin America and Carriberan')
    elif df['Country name'][i] == 'Bhutan':
        region.append('South Asia')
    elif df['Country name'][i] == 'Somalia':
        region.append('Sub-Saharan Africa')
    elif df['Country name'][i] == 'Cuba':
        region.append('Latin America and Carriberan')
    elif df['Country name'][i] == 'Oman':
        region.append('Middle East and North Africa')
    elif df['Country name'][i] == 'Guyana':
        region.append('Latin America and Carriberan')
    else:
        region.append(country_continent[df['Country name'][i]])

df['Region'] = region
```

# We also want to combine population with this datafrrmae

```
all_countries_pop = pop['Country Name'].value_counts().reset_index()['Country Name'].tolist()

del_cou = []
for x in all_countries:
   if x not in all_countries_pop:
      del_cou.append(x)
del_cou
df.head()
pop_df = df[['Log GDP per capita', 'Life Ladder', 'Country name', 'year', 'Social support',
'Healthy life expectancy at birth', 'Freedom to make life choices', 'Generosity', 'Perceptions of
corruption', 'Region' ]].copy()
```

# Make copy to not affect changes each other.

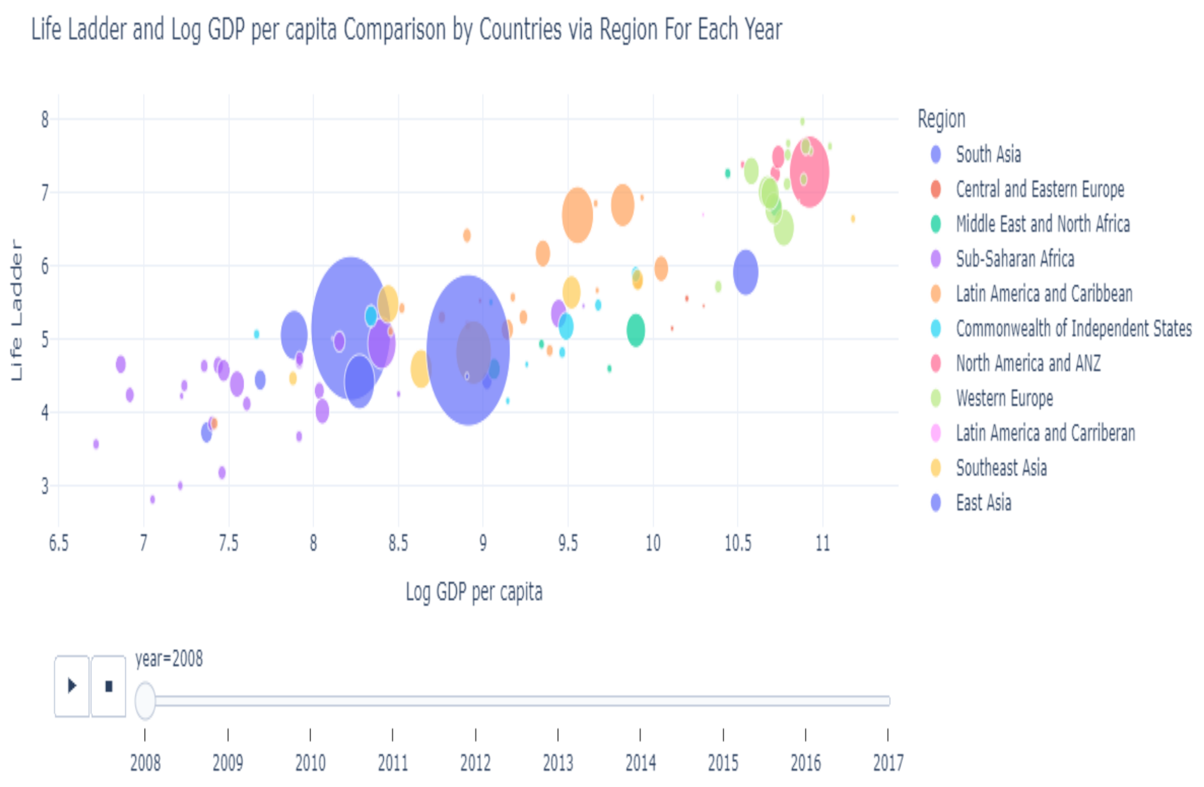
```
pop_df.head()
pop_df = pop_df[~pop_df['Country name'].isin(del_cou)]
```

# ~ = not include, the opposite

```
pop_df = pop_df[~pop_df.year.isin([2005, 2006, 2007, 2018, 2019, 2020, 2021])]
```

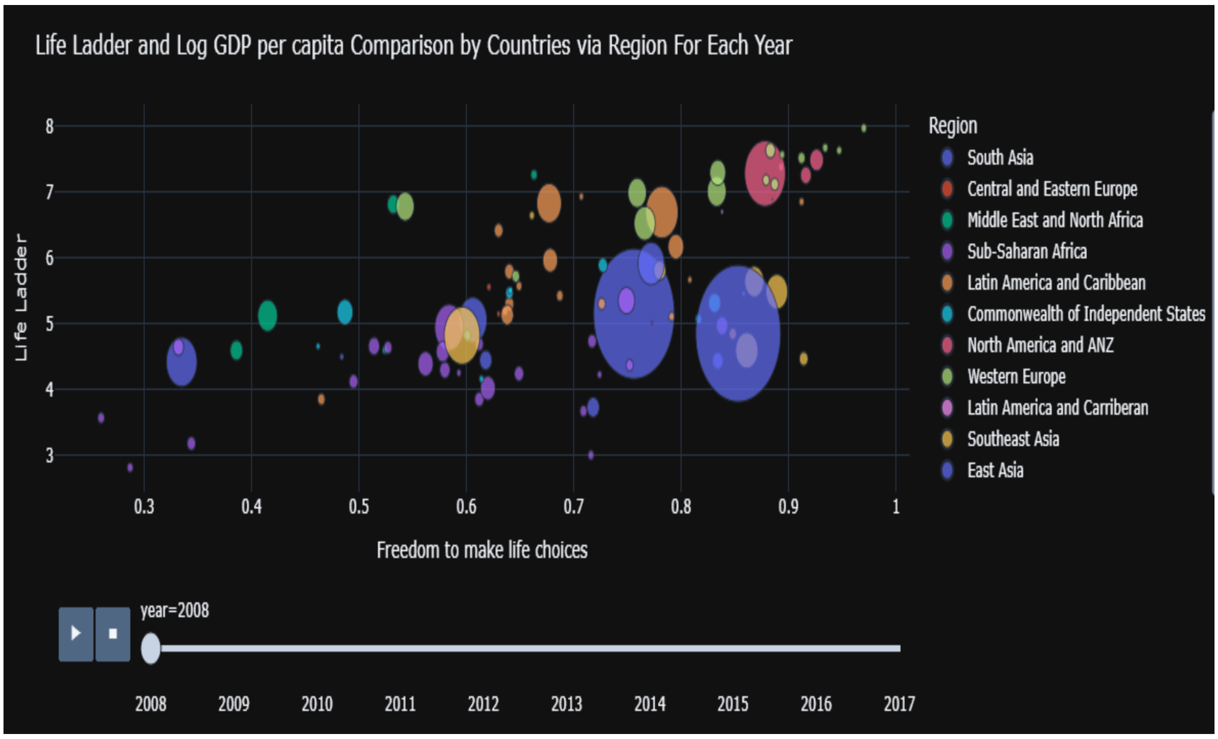
# Create a dictinory and fill it

```
pop_dict = {x: {} for x in range (2008, 2018)}
for i in range(len(pop)):
   if(pop['Year'][i] in range(2008, 2018)):
      pop_dict[pop['Year'][i]][pop['Country Name'][i]] = pop['Count'][i]
population = []
for i in pop_df.index:
   population.append(pop_dict[pop_df['year'][i]][pop_df['Country name'][i]])
pop_df['population'] = population
pop_df.head()
fig = px.scatter(pop_df,
            x = 'Log GDP per capita',
            y = 'Life Ladder',
            animation_frame='year',
            animation_group= 'Country name',
            size = 'population',
            template= 'plotly_white',
            color = 'Region',
            hover_name= 'Country name',
            size_max=60)
fig.update_layout(title = 'Life Ladder and Log GDP per capita Comparison by Countries via Region For Each Year')
fig.show()
<a id='12'></a>
```



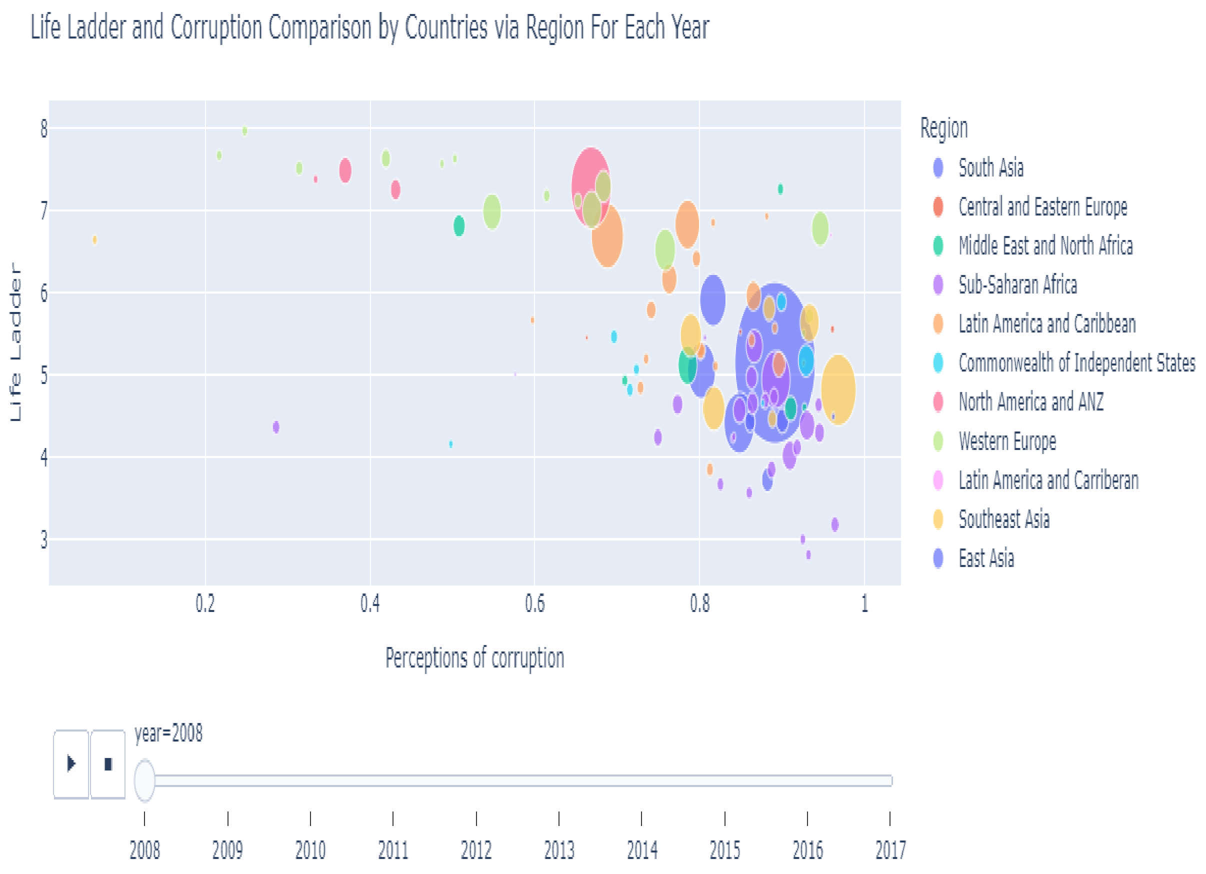
## Relations Between Hapiness and Freedom

```
fig = px.scatter(pop_df,

        x = 'Freedom to make life choices',

        y = 'Life Ladder',

        animation_frame='year',

        animation_group= 'Country name',

        size = 'population',

        template= 'plotly_dark',

        color = 'Region',

        hover_name= 'Country name',

        size_max=60)

fig.update_layout(title = 'Life Ladder and Log GDP per capita Comparison by Countries via Region For Each Year')

fig.show()

<a id='13'></a>
```



## Relations Between Hapiness and Corruption

```
fig = px.scatter(pop_df,
        x = 'Perceptions of corruption',
        y = 'Life Ladder',
        animation_frame='year',
        animation_group= 'Country name',
        size = 'population',
        color = 'Region',
        hover_name= 'Country name',
        size_max=60)
fig.update_layout(title = 'Life Ladder and Corruption Comparison by Countries via Region For Each Year')
fig.show()
<a id='14'></a>
```



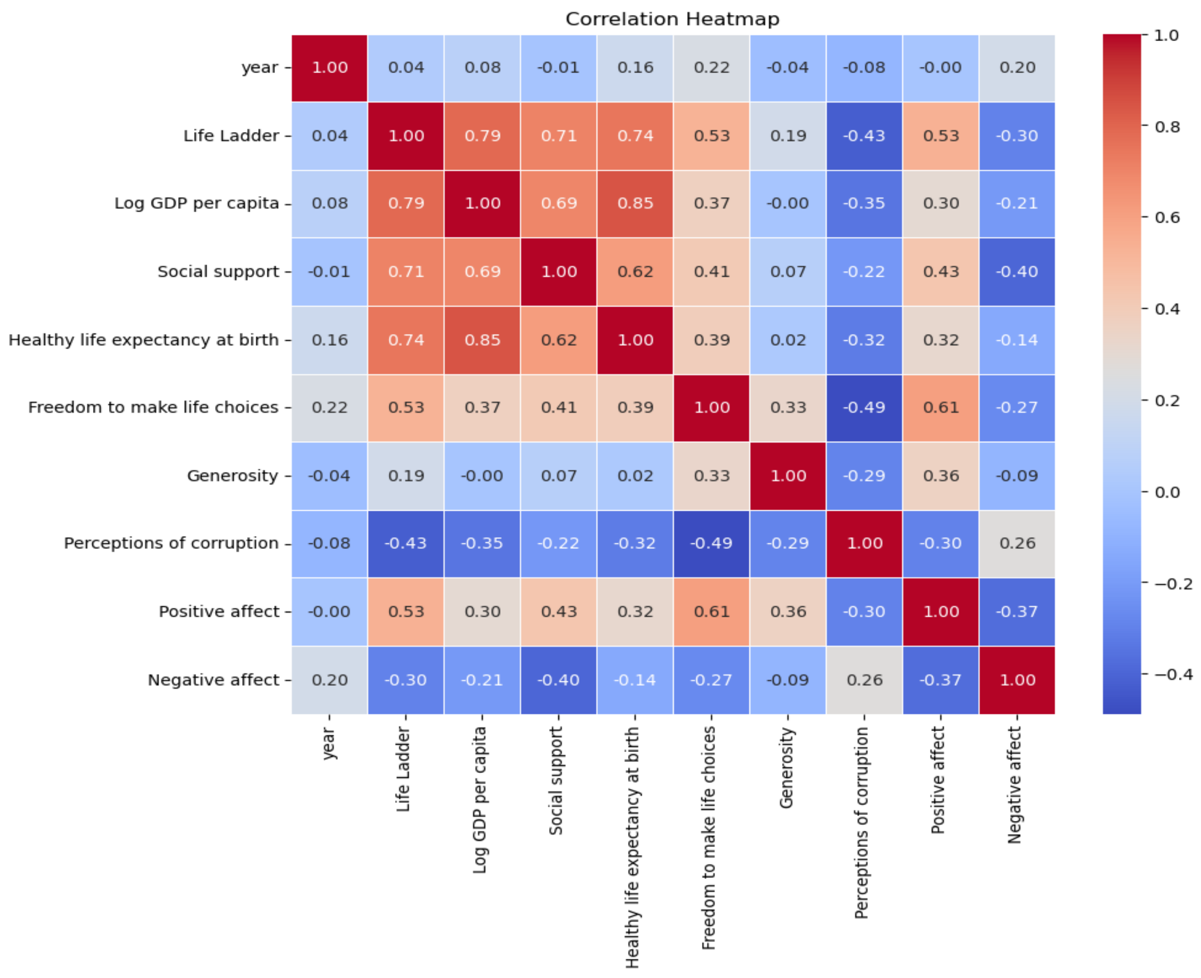
## Relations Between Features

```
numeric_df = df.drop(columns=['Country name', 'Region'])  # Drop string columns

numeric_df.corr()
```

# There is a strong relation between life ladder and social support.

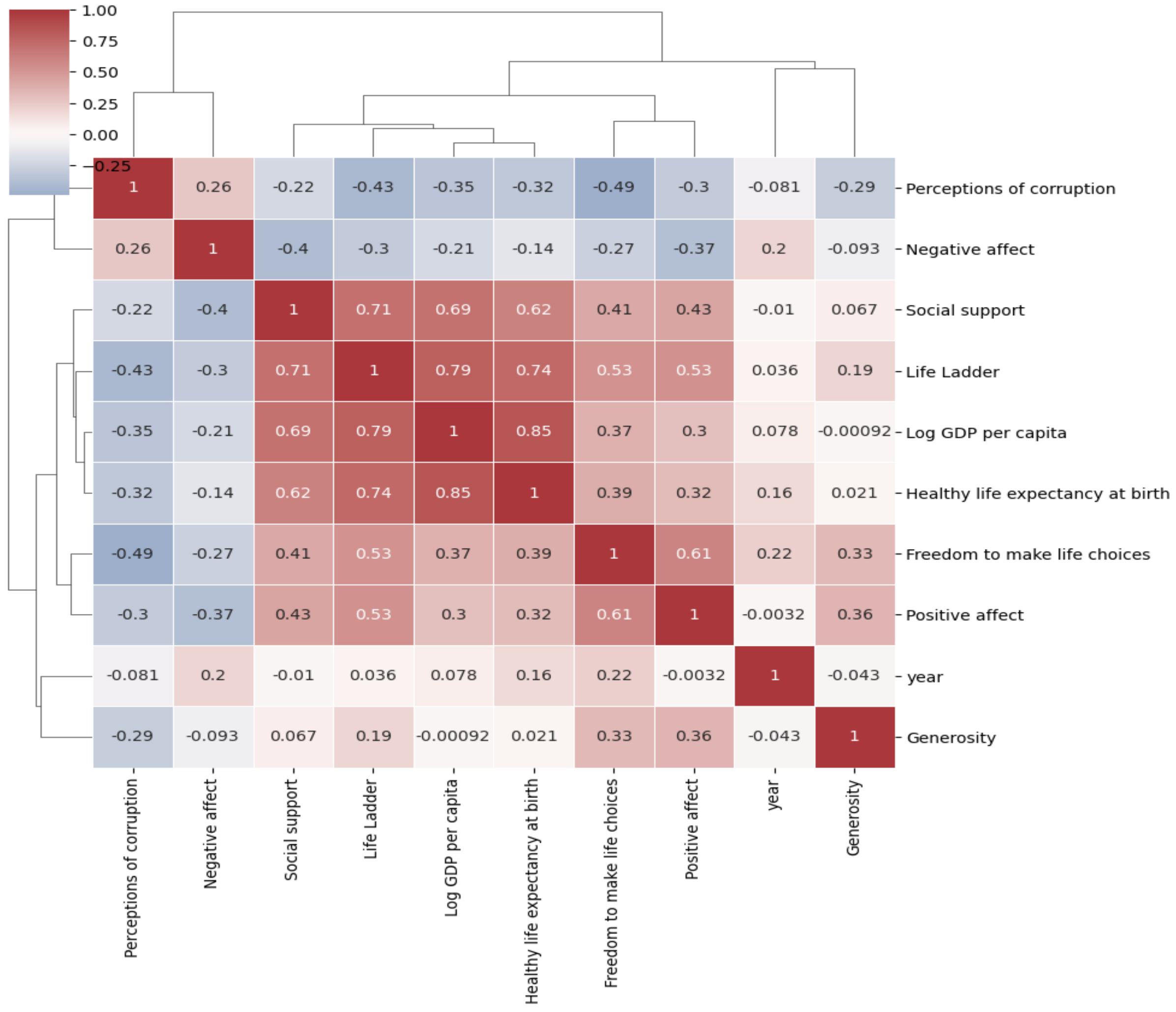
```
plt.figure(figsize=(10, 8))

sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)

plt.title('Correlation Heatmap')

plt.show()
```

# Corruption almost has negative relations with everything, except negative effect.



	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption	Positive affect	Negative affect
year	1.00	0.04	0.08	-0.01	0.16	0.22	-0.04	-0.08	-0.00	0.20
Life Ladder	0.04	1.00	0.79	0.71	0.74	0.53	0.19	-0.43	0.53	-0.30
Log GDP per capita	0.08	0.79	1.00	0.69	0.85	0.37	-0.00	-0.35	0.30	-0.21
Social support	-0.01	0.71	0.69	1.00	0.62	0.41	0.07	-0.22	0.43	-0.40
Healthy life expectancy at birth	0.16	0.74	0.85	0.62	1.00	0.39	0.02	-0.32	0.32	-0.14
Freedom to make life choices	0.22	0.53	0.37	0.41	0.39	1.00	0.33	-0.49	0.61	-0.27
Generosity	-0.04	0.19	-0.00	0.07	0.02	0.33	1.00	-0.29	0.36	-0.09
Perceptions of corruption	-0.08	-0.43	-0.35	-0.22	-0.32	-0.49	-0.29	1.00	-0.30	0.26
Positive affect	-0.00	0.53	0.30	0.43	0.32	0.61	0.36	-0.30	1.00	-0.37
Negative affect	0.20	-0.30	-0.21	-0.40	-0.14	-0.27	-0.09	0.26	-0.37	1.00

```
sns.clustermap(numeric_df.corr(), center = 0, cmap = 'vlag', dendrogram_ratio = (0.1, 0.2),
annot = True, linewidths=0.5, figsize= (10,10))

plt.show()

# The lines show the relations. Smaller lines mean stronger relations.
```


Perceptions of corruption
Negative affect
Social support
Life Ladder
Log GDP per capita
Healthy life expectancy at birth
Freedom to make life choices
Positive affect
year
Generosity