```
import plotly.express as px
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode, iplot
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use("seaborn-notebook")
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv('E:\PORTFOLIO-PROJECTS\World Happiness Report 2005-2021.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Country name	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	Confidence in national government
0	Afghanistan	2008	3.723590	7.302574	0.450662	50.500000	0.718114	0.173169	0.881686	0.414297	0.258195	0.612072
1	Afghanistan	2009	4.401778	7.472446	0.552308	50.799999	0.678896	0.195469	0.850035	0.481421	0.237092	0.611545
2	Afghanistan	2010	4.758381	7.579183	0.539075	51.099998	0.600127	0.125859	0.706766	0.516907	0.275324	0.299357
3	Afghanistan	2011	3.831719	7.552006	0.521104	51.400002	0.495901	0.167723	0.731109	0.479835	0.267175	0.307386
4	Afghanistan	2012	3.782938	7.637953	0.520637	51.700001	0.530935	0.241247	0.775620	0.613513	0.267919	0.435440

In [4]:

```
df.columns
```

Out[4]:

```
Index(['Country name', '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',
       'Confidence in national government'],
      dtype='object')
```

In [5]:

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

Out[5]:

```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Argentina',
       'Armenia', 'Australia', 'Austria', 'Azerbaijan', 'Bahrain',
       'Bangladesh', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bhutan',
       'Bolivia', 'Bosnia and Herzegovina', 'Botswana', 'Brazil',
       'Bulgaria', 'Burkina Faso', 'Burundi', 'Cambodia', 'Cameroon',
       'Canada', 'Central African Republic', 'Chad', 'Chile', 'China',
       'Colombia', 'Comoros', 'Congo (Brazzaville)', 'Congo (Kinshasa)',
       'Costa Rica', 'Croatia', 'Cuba', 'Cyprus', 'Czechia', 'Denmark',
       'Djibouti', 'Dominican Republic', 'Ecuador', 'Egypt',
       'El Salvador', 'Estonia', 'Eswatini', 'Ethiopia', 'Finland',
       'France', 'Gabon', 'Gambia', 'Georgia', 'Germany', 'Ghana',
       'Greece', 'Guatemala', 'Guinea', 'Guyana', 'Haiti', 'Honduras',
       'Hong Kong S.A.R. of China', 'Hungary', 'Iceland', 'India',
       'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Ivory Coast', 'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya',
       'Kosovo', 'Kuwait', 'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon',
       'Lesotho', 'Liberia', 'Libya', 'Lithuania', 'Luxembourg',
       'Madagascar', 'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta',
       'Mauritania', 'Mauritius', 'Mexico', 'Moldova', 'Mongolia',
       'Montenegro', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia',
       'Nepal', 'Netherlands', 'New Zealand', 'Nicaragua', 'Niger',
       'Nigeria', 'North Cyprus', 'North Macedonia', 'Norway', 'Oman',
       'Pakistan', 'Palestinian Territories', 'Panama', 'Paraguay',
       'Peru', 'Philippines', 'Poland', 'Portugal', 'Qatar', 'Romania',
       'Russia', 'Rwanda', 'Saudi Arabia', 'Senegal', 'Serbia',
       'Sierra Leone', 'Singapore', 'Slovakia', 'Slovenia', 'Somalia',
       'Somaliland region', 'South Africa', 'South Korea', 'South Sudan',
       'Spain', 'Sri Lanka', 'Sudan', 'Suriname', 'Sweden', 'Switzerland',
       'Syria', 'Taiwan Province of China', 'Tajikistan', 'Tanzania',
       'Thailand', 'Togo', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
       'Turkmenistan', 'Uganda', 'Ukraine', 'United Arab Emirates',
       'United Kingdom', 'United States', 'Uruguay', 'Uzbekistan',
       'Venezuela', 'Vietnam', 'Yemen', 'Zambia', 'Zimbabwe'],
      dtype=object)
```

In [6]:

```
df.shape
```

Out[6]:

```
(2089, 12)
```

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2089 entries, 0 to 2088
Data columns (total 12 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Country name                      2089 non-null   object 
 1   Year                              2089 non-null   int64  
 2   Life Ladder                       2089 non-null   float64
 3   Log GDP per capita                2062 non-null   float64
 4   Social support                    2076 non-null   float64
 5   Healthy life expectancy at birth  2031 non-null   float64
 6   Freedom to make life choices      2057 non-null   float64
 7   Generosity                        2009 non-null   float64
 8   Perceptions of corruption         1976 non-null   float64
 9   Positive affect                   2065 non-null   float64
 10  Negative affect                   2073 non-null   float64
 11  Confidence in national government  1873 non-null   float64
dtypes: float64(10), int64(1), object(1)
memory usage: 196.0+ KB
```

In [10]:

```
df.describe()
```

Out[10]:

	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	Confidence in national government
count	2089.000000	2089.000000	2062.000000	2076.000000	2031.000000	2057.000000	2009.000000	1976.000000	2065.000000	2073.000000	1873.000000
mean	2013.727621	5.473747	9.378408	0.811542	63.180326	0.745462	0.000102	0.746474	0.651421	0.270544	0.483914
std	4.455614	1.115567	1.143520	0.118935	6.948546	0.140751	0.161082	0.186136	0.105954	0.085849	0.191515
min	2005.000000	2.178809	5.526723	0.290184	6.720000	0.257534	-0.335739	0.035198	0.178886	0.082737	0.068769

	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	Confidence in national government
25%	2010.000000	4.651972	8.473547	0.747664	58.965000	0.651689	-0.113172	0.689840	0.570057	0.207652	0.334808
50%	2014.000000	5.405246	9.463269	0.834770	64.980003	0.767357	-0.023366	0.801339	0.662837	0.260328	0.467046
75%	2017.000000	6.294282	10.347656	0.904682	68.362499	0.857677	0.090584	0.870789	0.737176	0.321706	0.616302
max	2021.000000	8.018934	11.665803	0.987343	74.349998	0.985178	0.706377	0.983276	0.883586	0.704590	0.993604

In [12]:

```
df2021 = pd.read_csv('E:\PORTFOLIO-PROJECTS\world-happiness-report-2021.csv')
df2021.head()
```

Out[12]:

	Country name	Regional indicator	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
0	Finland	Western Europe	7.842	0.032	7.904	7.780	10.775	0.954	72.0	0.949	-0.098	0.186	2.43	1.446	1.106	0.741	0.691	0.124	0.481	3.253
1	Denmark	Western Europe	7.620	0.035	7.687	7.552	10.933	0.954	72.7	0.946	0.030	0.179	2.43	1.502	1.108	0.763	0.686	0.208	0.485	2.868

	Country name	Regional indicator	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
2	Switzerland	Western Europe	7.571	0.036	7.643	7.500	11.117	0.942	74.4	0.919	0.025	0.292	2.43	1.566	1.079	0.816	0.653	0.204	0.413	2.839
3	Iceland	Western Europe	7.554	0.059	7.670	7.438	10.878	0.983	73.0	0.955	0.160	0.673	2.43	1.482	1.172	0.772	0.698	0.293	0.170	2.967
4	Netherlands	Western Europe	7.464	0.027	7.518	7.410	10.932	0.942	72.4	0.913	0.175	0.338	2.43	1.501	1.079	0.753	0.647	0.302	0.384	2.798

In [13]:

```
df2021.columns
```

Out[13]:

```
Index(['Country name', 'Regional indicator', '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'],
      dtype='object')
```

In [14]:

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

Out[14]:

```
array(['Finland', 'Denmark', 'Switzerland', 'Iceland', 'Netherlands',
       'Norway', 'Sweden', 'Luxembourg', 'New Zealand', 'Austria',
       'Australia', 'Israel', 'Germany', 'Canada', 'Ireland',
       'Costa Rica', 'United Kingdom', 'Czech Republic', 'United States',
       'Belgium', 'France', 'Bahrain', 'Malta',
       'Taiwan Province of China', 'United Arab Emirates', 'Saudi Arabia',
       'Spain', 'Italy', 'Slovenia', 'Guatemala', 'Uruguay', 'Singapore',
       'Kosovo', 'Slovakia', 'Brazil', 'Mexico', 'Jamaica', 'Lithuania',
       'Cyprus', 'Estonia', 'Panama', 'Uzbekistan', 'Chile', 'Poland',
       'Kazakhstan', 'Romania', 'Kuwait', 'Serbia', 'El Salvador',
       'Mauritius', 'Latvia', 'Colombia', 'Hungary', 'Thailand',
       'Nicaragua', 'Japan', 'Argentina', 'Portugal', 'Honduras',
       'Croatia', 'Philippines', 'South Korea', 'Peru',
       'Bosnia and Herzegovina', 'Moldova', 'Ecuador', 'Kyrgyzstan',
       'Greece', 'Bolivia', 'Mongolia', 'Paraguay', 'Montenegro',
       'Dominican Republic', 'North Cyprus', 'Belarus', 'Russia',
       'Hong Kong S.A.R. of China', 'Tajikistan', 'Vietnam', 'Libya',
       'Malaysia', 'Indonesia', 'Congo (Brazzaville)', 'China',
       'Ivory Coast', 'Armenia', 'Nepal', 'Bulgaria', 'Maldives',
       'Azerbaijan', 'Cameroon', 'Senegal', 'Albania', 'North Macedonia',
       'Ghana', 'Niger', 'Turkmenistan', 'Gambia', 'Benin', 'Laos',
       'Bangladesh', 'Guinea', 'South Africa', 'Turkey', 'Pakistan',
       'Morocco', 'Venezuela', 'Georgia', 'Algeria', 'Ukraine', 'Iraq',
       'Gabon', 'Burkina Faso', 'Cambodia', 'Mozambique', 'Nigeria',
       'Mali', 'Iran', 'Uganda', 'Liberia', 'Kenya', 'Tunisia', 'Lebanon',
       'Namibia', 'Palestinian Territories', 'Myanmar', 'Jordan', 'Chad',
       'Sri Lanka', 'Swaziland', 'Comoros', 'Egypt', 'Ethiopia',
       'Mauritania', 'Madagascar', 'Togo', 'Zambia', 'Sierra Leone',
       'India', 'Burundi', 'Yemen', 'Tanzania', 'Haiti', 'Malawi',
       'Lesotho', 'Botswana', 'Rwanda', 'Zimbabwe', 'Afghanistan'],
      dtype=object)
```

In [15]:

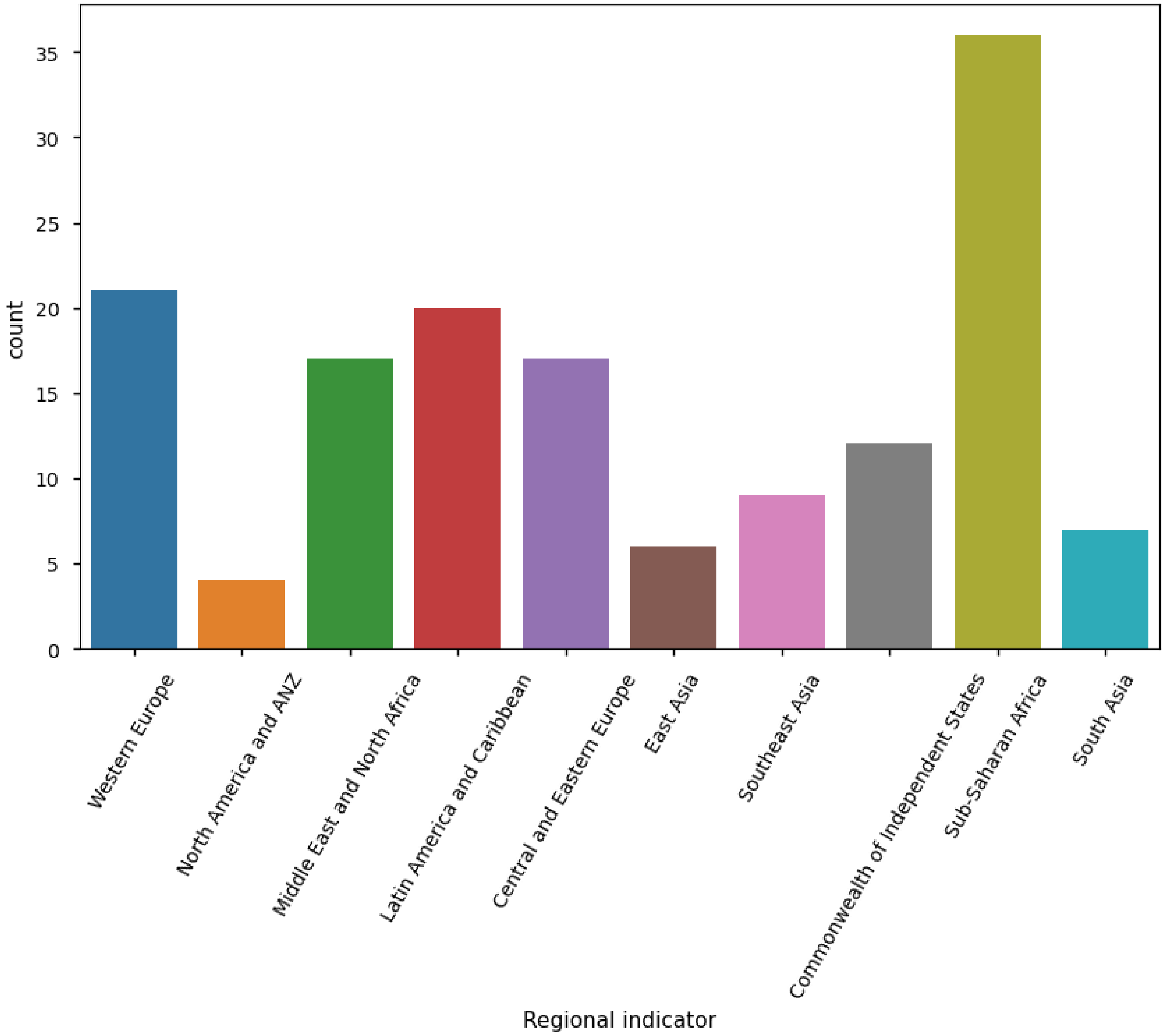
```
len(df2021['Country name'].unique())
```

Out[15]:

```
149
```

In [16]:

```
#count Regional Indicators
plt.figure(figsize = (10,6))
sns.countplot(x=df2021["Regional indicator"])
plt.xticks(rotation = 60)
plt.show()
```


35
30
25
20
15
10
5
0
count
Western Europe
North America and ANZ
Middle East and North Africa
Latin America and Caribbean
Central and Eastern Europe
East Asia
Southeast Asia
Commonwealth of Independent States
Sub-Saharan Africa
South Asia
Regional indicator

In [17]:

```
#Distribution of features
list_features = ["Social support", "Freedom to make life choices", "Generosity", "Perceptions of corruption"]
sns.boxplot(data = df2021.loc[:, list_features], orient = "v", palette = "rainbow")
plt.show()
```

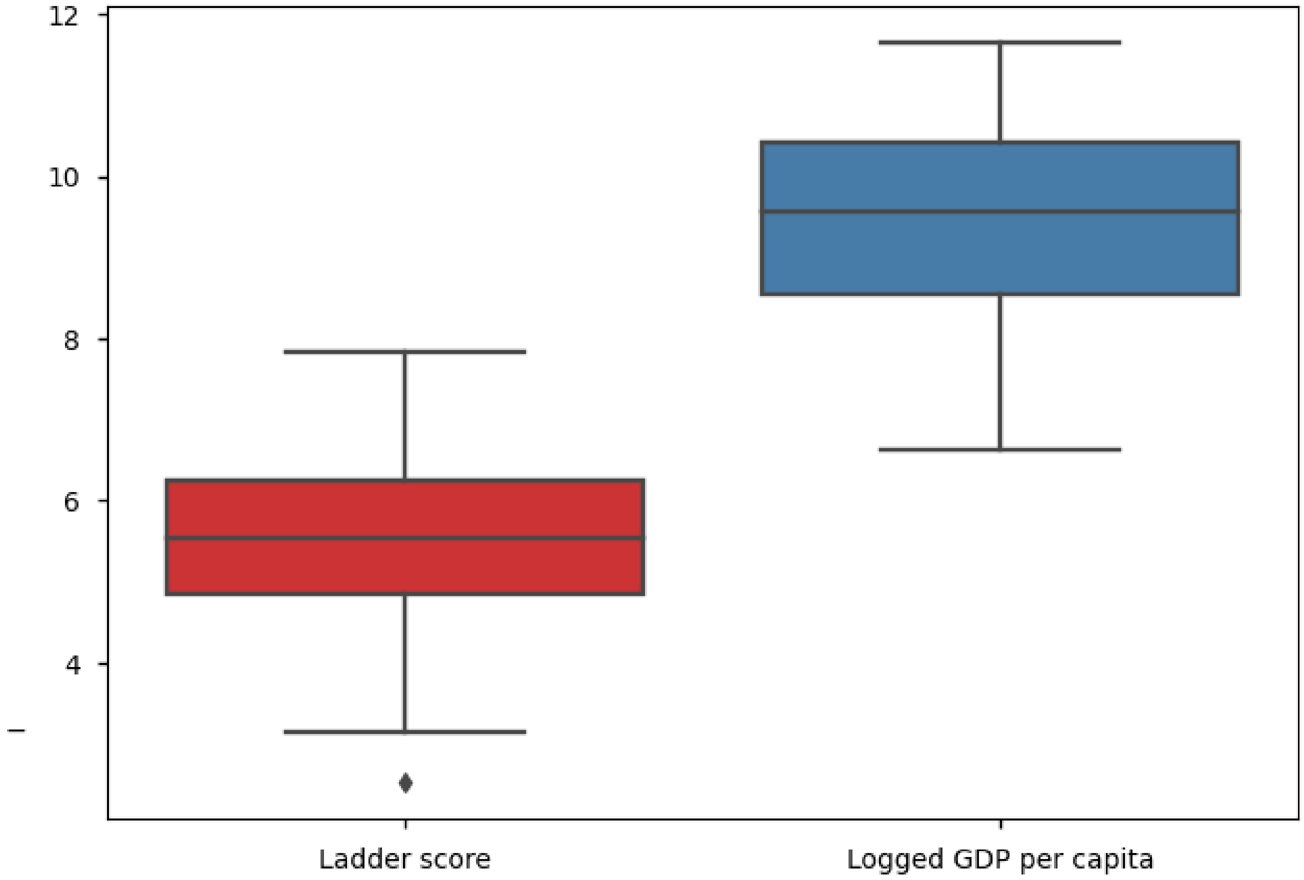
In [18]:

```
list_features = ["Ladder score", "Logged
GDP per capita"]
sns.boxplot(data = df2021.loc[:, list_features], orient = "v", palette = "Set1")
plt.show()
```

In [19]:

```
list_features = ["Healthy life expectancy"]
sns.boxplot(data = df2021.loc[:, list_features], orient = "v", palette = "Set1")
plt.show()
```

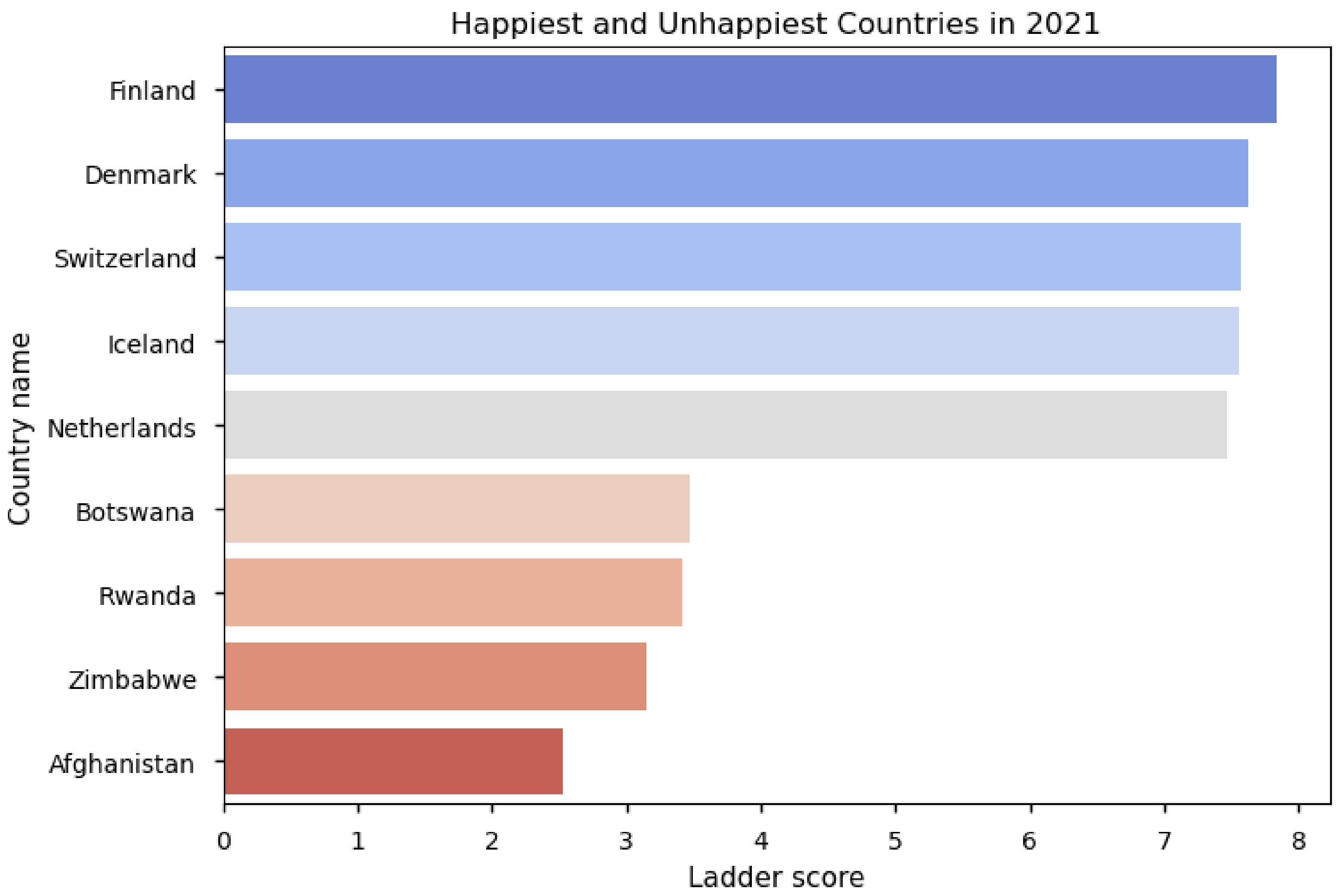
In [20]:

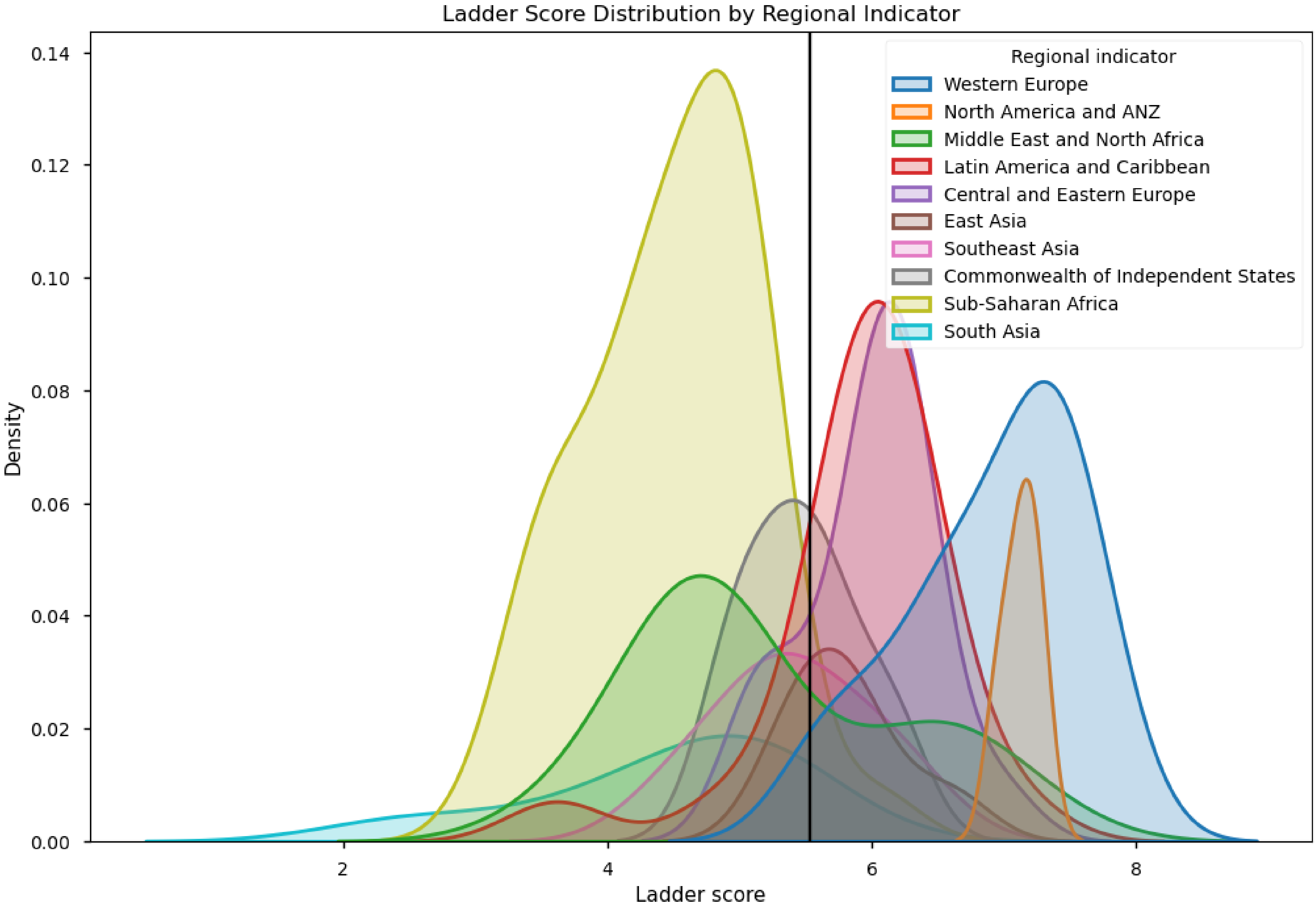


```
df2021_happiest_unhappiest = df2021[(df2021.loc[:, "Ladder score"] > 7.4) | (df2021.loc[:, "Ladder score"] < 3.5)]
sns.barplot(x = "Ladder score", y = "Country name", data=df2021_happiest_unhappiest, palette = "coolwarm")
plt.title("Happiest and Unhappiest Countries in 2021")
plt.show()
```

In [21]:

```
plt.figure(figsize = (12,8))
sns.kdeplot(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()
```




Ladder Score Distribution by Regional Indicator
Regional indicator
Western Europe
North America and ANZ
Middle East and North Africa
Latin America and Caribbean
Central and Eastern Europe
East Asia
Southeast Asia
Commonwealth of Independent States
Sub-Saharan Africa
South Asia
Density
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
2
4
6
8
Ladder score

In [23]:

```
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()
```

In [24]:

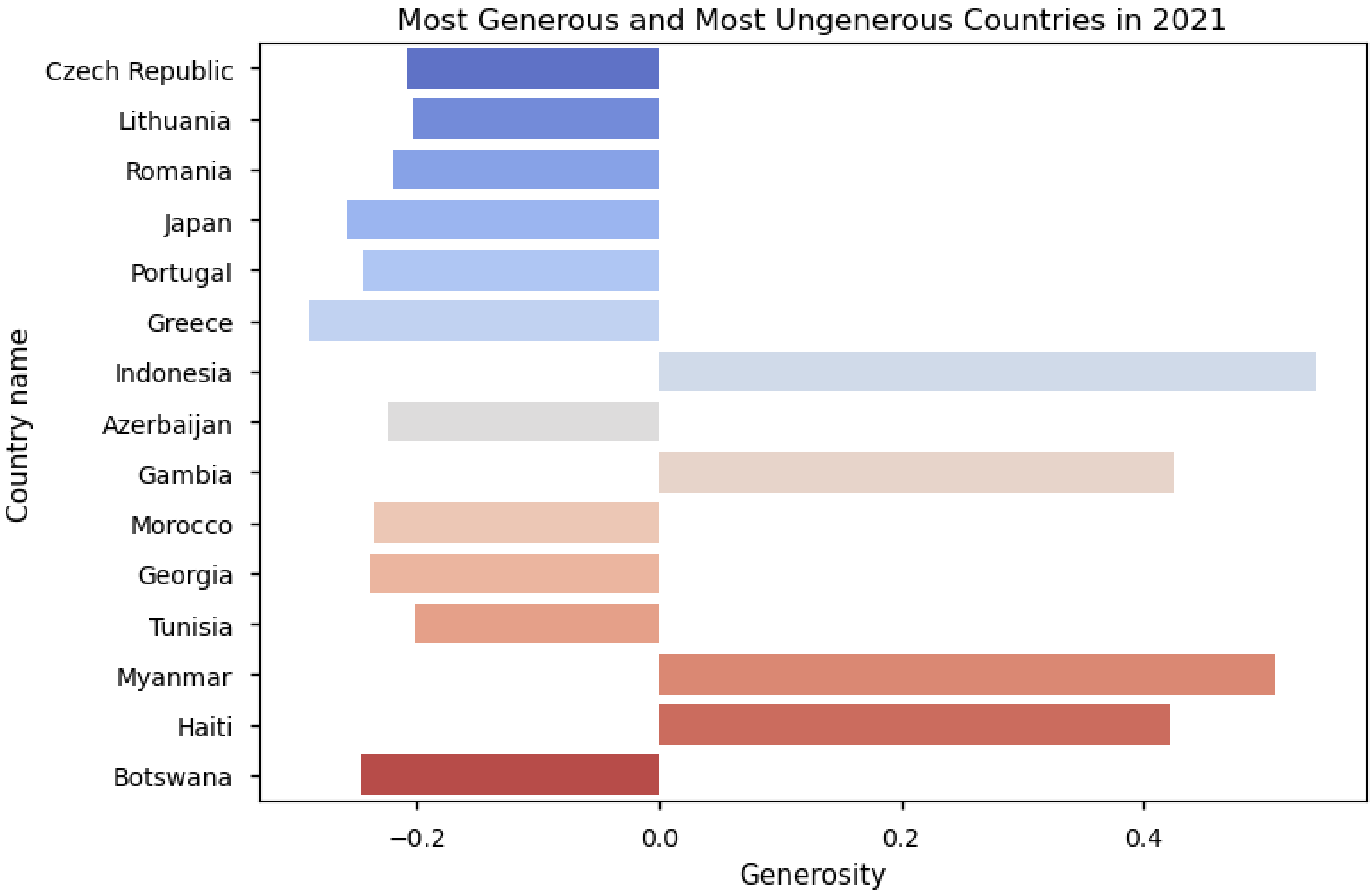
```
df2021_g = df2021[(df2021.loc[:, "Generosity"] > 0.4) | (df2021.loc[:, "Generosity"] < -0.2)]
sns.barplot(x = "Generosity", y = "Country name", data = df2021_g, palette = "coolwarm")
plt.title("Most Generous and Most Ungenerous Countries in 2021")
plt.show()
```

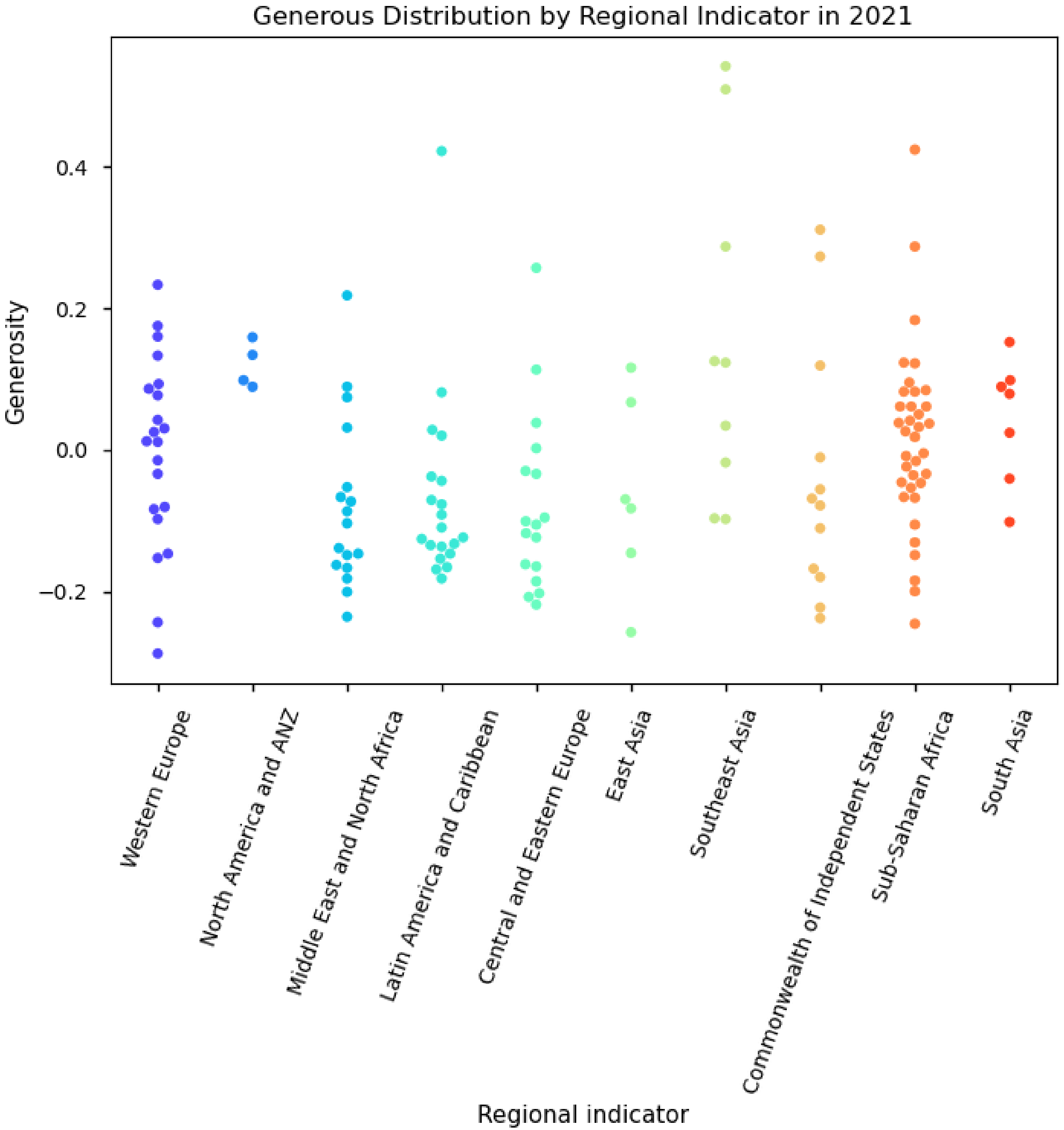
In [25]:

```
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()
```

In [26]:

```
sns.swarmplot(x = "Regional indicator", y = "Generosity", data = df2021, palette = "rainbow" )
plt.xticks(rotation = 70)
plt.title("Generous Distribution by Regional Indicator in 2021")
plt.show()
```


Most Generous and Most Ungenerous Countries in 2021
Country name
Czech Republic
Lithuania
Romania
Japan
Portugal
Greece
Indonesia
Azerbaijan
Gambia
Morocco
Georgia
Tunisia
Myanmar
Haiti
Botswana
−0.2
0.0
0.2
0.4
Generosity


Generous Distribution by Regional Indicator in 2021
Generosity
0.4
0.2
0.0
−0.2
Western Europe
North America and ANZ
Middle East and North Africa
Latin America and Caribbean
Central and Eastern Europe
East Asia
Southeast Asia
Commonwealth of Independent States
Sub-Saharan Africa
South Asia
Regional indicator

In [27]:

```
pop = pd.read_csv('E:\PORTFOLIO-PROJECTS\population_total_long.csv')
```

In [28]:

```
pop.head()
```

Out[28]:

	Country Name	Year	Count
0	Aruba	1960	54211
1	Afghanistan	1960	8996973
2	Angola	1960	5454933
3	Albania	1960	1608800
4	Andorra	1960	13411

In [30]:

```
country_continent={}
for i in range(len(df2021)):
    country_continent[df2021["Country name"][i]]=df2021["Regional indicator"][i]
all_countries=df["Country name"].value_counts().reset_index()["index"].tolist()
all_countries_2021=df2021["Country name"].value_counts().reset_index()["index"].tolist()

for x in all_countries:
    if x not in all_countries_2021:
        print(x)
```

```
Czechia
Congo (Kinshasa)
Syria
Central African Republic
Sudan
Qatar
Trinidad and Tobago
Angola
Somaliland region
South Sudan
```

```
Djibouti
Somalia
Bhutan
Eswatini
Belize
Suriname
Guyana
Cuba
Oman
```

In [41]:

```
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 Caribbean")
    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 Caribbean")
    elif df['Country name'][i] == 'Cuba':
        region.append("Latin America and Caribbean")
    elif df['Country name'][i] == 'Qatar':
        region.append("Middle East and North Africa")
    elif df['Country name'][i] == 'Sudan':
        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] == 'Djibouti':
        region.append("Sub-Saharan Africa")
    elif df['Country name'][i] == 'Somaliland region':
        region.append("Sub-Saharan Africa")
    elif df['Country name'][i] == 'South Sudan':
        region.append("Middle East and North Africa")
    elif df['Country name'][i] == 'Somalia':
        region.append("Sub-Saharan Africa")
    elif df['Country name'][i] == 'Oman':
        region.append("Middle East and North Africa")
    elif df['Country name'][i] == 'Guyana':
        region.append("Latin America and Caribbean")
    elif df['Country name'][i] == 'Guyana':
        region.append("Latin America and Caribbean")
    elif df['Country name'][i] == 'Bhutan':
        region.append("South Asia")
    elif df['Country name'][i] == 'Suriname':
        region.append("Latin America and Caribbean")
    elif df['Country name'][i] == 'Czechia':
        region.append("Europe")
    else:
        region.append(country_continent[df['Country name'].tolist()[i]])
        country_continent['Eswatini'] = 'Africa'

df["region"] = region
```

In [42]:

```
df.head(3)
```

Out[42]:

	Country name	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	Confidence in national government	region
0	Afghanistan	2008	3.723590	7.302574	0.450662	50.500000	0.718114	0.173169	0.881686	0.414297	0.258195	0.612072	South Asia
1	Afghanistan	2009	4.401778	7.472446	0.552308	50.799999	0.678896	0.195469	0.850035	0.481421	0.237092	0.611545	South Asia
2	Afghanistan	2010	4.758381	7.579183	0.539075	51.099998	0.600127	0.125859	0.706766	0.516907	0.275324	0.299357	South Asia

In [44]:

```
all_countries = df["Country name"].value_counts().reset_index()["index"].tolist()
all_countries_pop = pop["Country Name"].value_counts().reset_index()["index"].tolist()

del_cou = []
for x in all_countries:
    if x not in all_countries_pop:
        del_cou.append(x)
del_cou
```

Out[44]:

```
['Egypt',
 'Kyrgyzstan',
 'Russia',
 'South Korea',
 'Venezuela',
 'Palestinian Territories',
 'Taiwan Province of China',
 'Iran',
 'Slovakia',
 'Czechia',
 'Congo (Brazzaville)',
 'Hong Kong S.A.R. of China',
 'Yemen',
 'Laos',
 'Ivory Coast',
 'Congo (Kinshasa)',
 'Syria',
 'North Cyprus',
 'Somaliland region',
 'Gambia']
```

In [55]:

```
df.columns
```

Out[55]:

```
Index(['Country name', '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',
       'Confidence in national government', 'region'],
      dtype='object')
```

In [57]:

```
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',"region",'Perceptions of corruption']].copy()
pop_df.head()
```

Out[57]:

	Log GDP per capita	Life Ladder	Country name	Year	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	region	Perceptions of corruption
0	7.302574	3.723590	Afghanistan	2008	0.450662	50.500000	0.718114	0.173169	South Asia	0.881686
1	7.472446	4.401778	Afghanistan	2009	0.552308	50.799999	0.678896	0.195469	South Asia	0.850035
2	7.579183	4.758381	Afghanistan	2010	0.539075	51.099998	0.600127	0.125859	South Asia	0.706766
3	7.552006	3.831719	Afghanistan	2011	0.521104	51.400002	0.495901	0.167723	South Asia	0.731109
4	7.637953	3.782938	Afghanistan	2012	0.520637	51.700001	0.530935	0.241247	South Asia	0.775620

In [60]:

```
pop_df = pop_df[~pop_df["Country name"].isin(del_cou)]
pop_df = pop_df[~pop_df.Year.isin([2006,2005,2007,2018,2019,2020,2021])]
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]
```

In [63]:

```
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()
```

Out[63]:

	Log GDP per capita	Life Ladder	Country name	Year	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	region	Perceptions of corruption	population
0	7.302574	3.723590	Afghanistan	2008	0.450662	50.500000	0.718114	0.173169	South Asia	0.881686	27722276
1	7.472446	4.401778	Afghanistan	2009	0.552308	50.799999	0.678896	0.195469	South Asia	0.850035	28394813
2	7.579183	4.758381	Afghanistan	2010	0.539075	51.099998	0.600127	0.125859	South Asia	0.706766	29185507
3	7.552006	3.831719	Afghanistan	2011	0.521104	51.400002	0.495901	0.167723	South Asia	0.731109	30117413
4	7.637953	3.782938	Afghanistan	2012	0.520637	51.700001	0.530935	0.241247	South Asia	0.775620	31161376

In [64]:

```
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 Regions for each Year")
fig.show()
```

In [65]:

```
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 Freedom Comparison by Countries via Regions for each Year")
fig.show()
```

In [66]:

```
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 Regions for each Year")
fig.show()
```

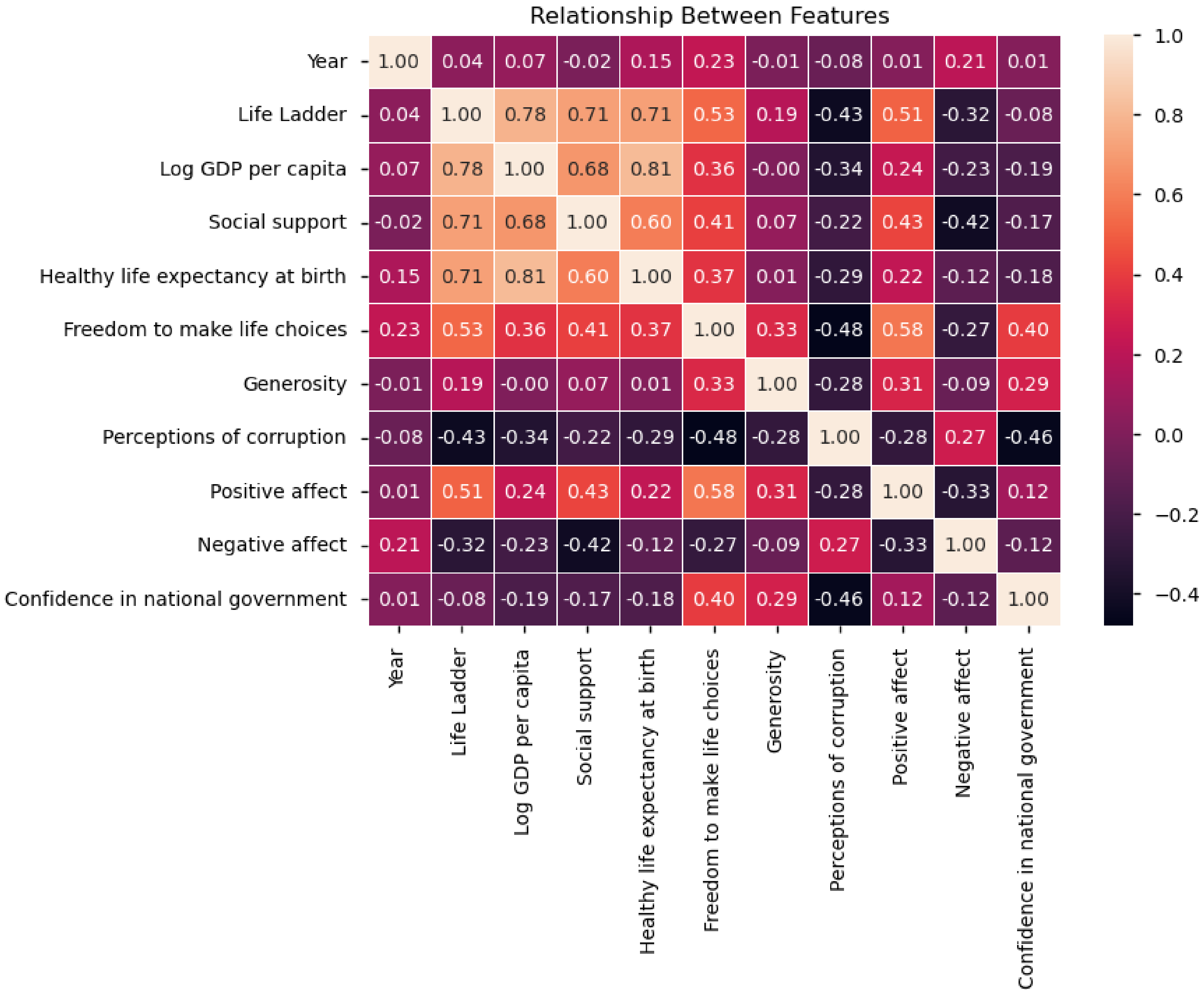
In [53]:

```
sns.heatmap(df.corr(), annot = True, fmt = ".2f", linewidth = .7)
plt.title("Relationship Between Features ")
```

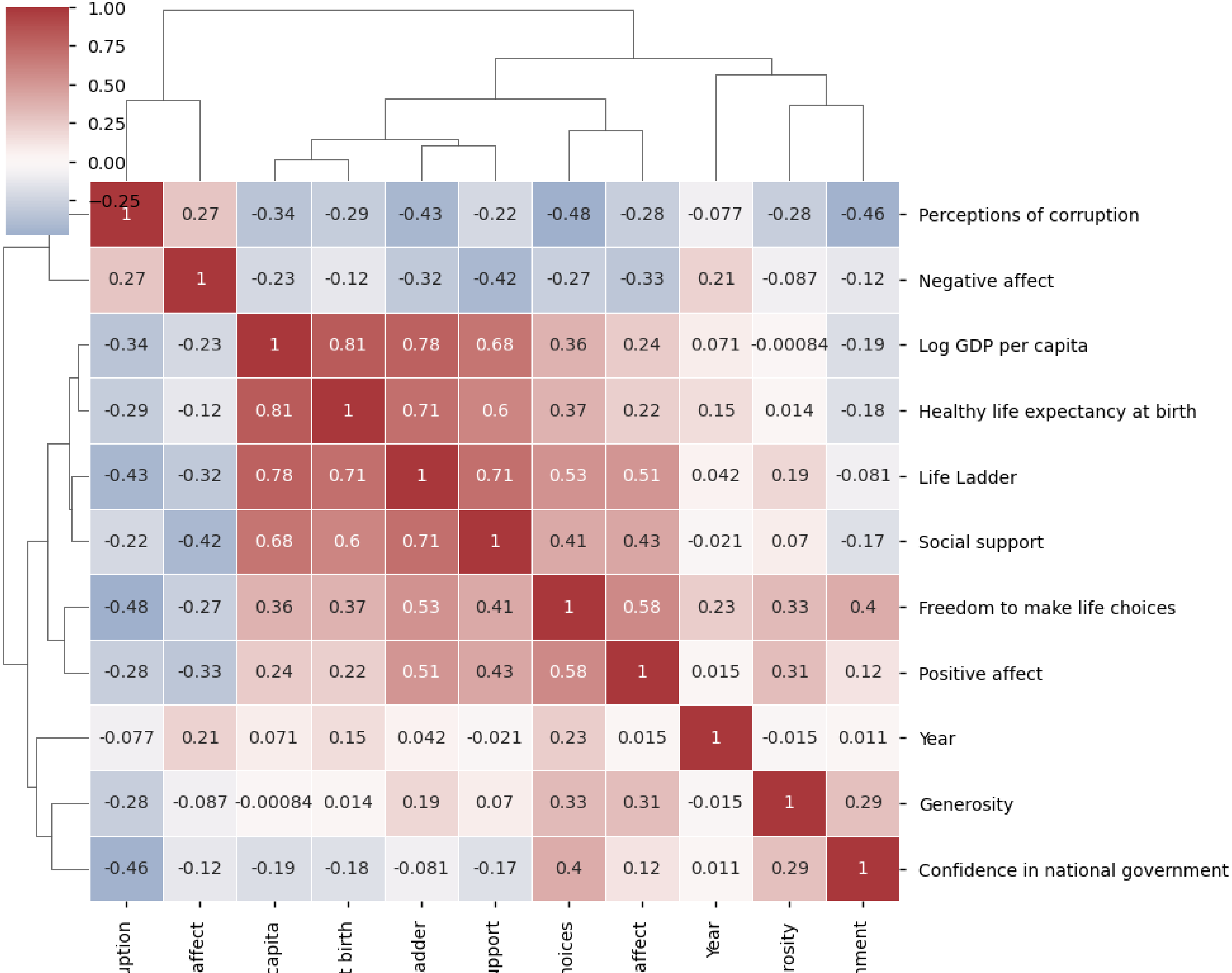
Out[53]:

```
Text(0.5, 1.0, 'Relationship Between Features ')
```

In [54]:


Relationship Between Features
Year 1.00 0.04 0.07 -0.02 0.15 0.23 -0.01 -0.08 0.01 0.21 0.01
Life Ladder 0.04 1.00 0.78 0.71 0.71 0.53 0.19 -0.43 0.51 -0.32 -0.08
Log GDP per capita 0.07 0.78 1.00 0.68 0.81 0.36 -0.00 -0.34 0.24 -0.23 -0.19
Social support -0.02 0.71 0.68 1.00 0.60 0.41 0.07 -0.22 0.43 -0.42 -0.17
Healthy life expectancy at birth 0.15 0.71 0.81 0.60 1.00 0.37 0.01 -0.29 0.22 -0.12 -0.18
Freedom to make life choices 0.23 0.53 0.36 0.41 0.37 1.00 0.33 -0.48 0.58 -0.27 0.40
Generosity -0.01 0.19 -0.00 0.07 0.01 0.33 1.00 -0.28 0.31 -0.09 0.29
Perceptions of corruption -0.08 -0.43 -0.34 -0.22 -0.29 -0.48 -0.28 1.00 -0.28 0.27 -0.46
Positive affect 0.01 0.51 0.24 0.43 0.22 0.58 0.31 -0.28 1.00 -0.33 0.12
Negative affect 0.21 -0.32 -0.23 -0.42 -0.12 -0.27 -0.09 0.27 -0.33 1.00 -0.12
Confidence in national government 0.01 -0.08 -0.19 -0.17 -0.18 0.40 0.29 -0.46 0.12 -0.12 1.00
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
Confidence in national government
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

```
sns.clustermap(df.corr(), center = 0, cmap = "vlag", dendrogram_ratio = (0.1, 0.2), annot = True, linewidths = .7,
figsize=(10,10))
plt.show()
```



	uption	affect	capita	t birth	adder	upport	hoices	affect	Year	rosity	nment
Perceptions of corruption	1	0.27	-0.34	-0.29	-0.43	-0.22	-0.48	-0.28	-0.077	-0.28	-0.46
Negative affect	0.27	1	-0.23	-0.12	-0.32	-0.42	-0.27	-0.33	0.21	-0.087	-0.12
Log GDP per capita	-0.34	-0.23	1	0.81	0.78	0.68	0.36	0.24	0.071	-0.00084	-0.19
Healthy life expectancy at birth	-0.29	-0.12	0.81	1	0.71	0.6	0.37	0.22	0.15	0.014	-0.18
Life Ladder	-0.43	-0.32	0.78	0.71	1	0.71	0.53	0.51	0.042	0.19	-0.081
Social support	-0.22	-0.42	0.68	0.6	0.71	1	0.41	0.43	-0.021	0.07	-0.17
Freedom to make life choices	-0.48	-0.27	0.36	0.37	0.53	0.41	1	0.58	0.23	0.33	0.4
Positive affect	-0.28	-0.33	0.24	0.22	0.51	0.43	0.58	1	0.015	0.31	0.12
Year	-0.077	0.21	0.071	0.15	0.042	-0.021	0.23	0.015	1	-0.015	0.011
Generosity	-0.28	-0.087	-0.00084	0.014	0.19	0.07	0.33	0.31	-0.015	1	0.29
Confidence in national government	-0.46	-0.12	-0.19	-0.18	-0.081	-0.17	0.4	0.12	0.011	0.29	1