Import

In [31]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

sns.set(rc={'figure.figsize':(5, 5)})
```

Read Data

In [32]:

```
df = pd.read_csv("world-happiness-report.csv")
df.head(10)
```

Out[32]:

	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
0	Afghanistan	2008	3.724	7.370	0.451	50.80	0.718	0.168	0.882	0.518	0.258
1	Afghanistan	2009	4.402	7.540	0.552	51.20	0.679	0.190	0.850	0.584	0.237
2	Afghanistan	2010	4.758	7.647	0.539	51.60	0.600	0.121	0.707	0.618	0.275
3	Afghanistan	2011	3.832	7.620	0.521	51.92	0.496	0.162	0.731	0.611	0.267
4	Afghanistan	2012	3.783	7.705	0.521	52.24	0.531	0.236	0.776	0.710	0.268
5	Afghanistan	2013	3.572	7.725	0.484	52.56	0.578	0.061	0.823	0.621	0.273
6	Afghanistan	2014	3.131	7.718	0.526	52.88	0.509	0.104	0.871	0.532	0.375
7	Afghanistan	2015	3.983	7.702	0.529	53.20	0.389	0.080	0.881	0.554	0.339
8	Afghanistan	2016	4.220	7.697	0.559	53.00	0.523	0.042	0.793	0.565	0.348
9	Afghanistan	2017	2.662	7.697	0.491	52.80	0.427	-0.121	0.954	0.496	0.371

Explarotary Data Analysis

In [33]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1949 entries, 0 to 1948
Data columns (total 11 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Country name                      1949 non-null   object 
 1   year                              1949 non-null   int64  
 2   Life Ladder                       1949 non-null   float64
 3   Log GDP per capita                1913 non-null   float64
 4   Social support                    1936 non-null   float64
 5   Healthy life expectancy at birth  1894 non-null   float64
 6   Freedom to make life choices      1917 non-null   float64
 7   Generosity                        1860 non-null   float64
 8   Perceptions of corruption         1839 non-null   float64
 9   Positive affect                   1927 non-null   float64
 10  Negative affect                   1933 non-null   float64
dtypes: float64(9), int64(1), object(1)
```

```
memory usage: 167.6+ KB
```

In [34]:

```
df["Country name"].value_counts()
```

Out[34]:

```
Japan         15
Kenya         15
Ecuador       15
Venezuela     15
Canada        15
              ..
Cuba           1
Maldives       1
Oman           1
Suriname       1
Guyana         1
Name: Country name, Length: 166, dtype: int64
```

In [35]:

```
df.describe().T
```

Out[35]:

	count	mean	std	min	25%	50%	75%	max
year	1949.0	2013.216008	4.166828	2005.000	2010.00000	2013.0000	2017.000	2020.000
Life Ladder	1949.0	5.466705	1.115711	2.375	4.64000	5.3860	6.283	8.019
Log GDP per capita	1913.0	9.368453	1.154084	6.635	8.46400	9.4600	10.353	11.648
Social support	1936.0	0.812552	0.118482	0.290	0.74975	0.8355	0.905	0.987
Healthy life expectancy at birth	1894.0	63.359374	7.510245	32.300	58.68500	65.2000	68.590	77.100
Freedom to make life choices	1917.0	0.742558	0.142093	0.258	0.64700	0.7630	0.856	0.985
Generosity	1860.0	0.000103	0.162215	-0.335	-0.11300	-0.0255	0.091	0.698
Perceptions of corruption	1839.0	0.747125	0.186789	0.035	0.69000	0.8020	0.872	0.983
Positive affect	1927.0	0.710003	0.107100	0.322	0.62550	0.7220	0.799	0.944
Negative affect	1933.0	0.268544	0.085168	0.083	0.20600	0.2580	0.320	0.705

Visualize Data & Preprocessing

In [36]:

```
from os import mkdir
```

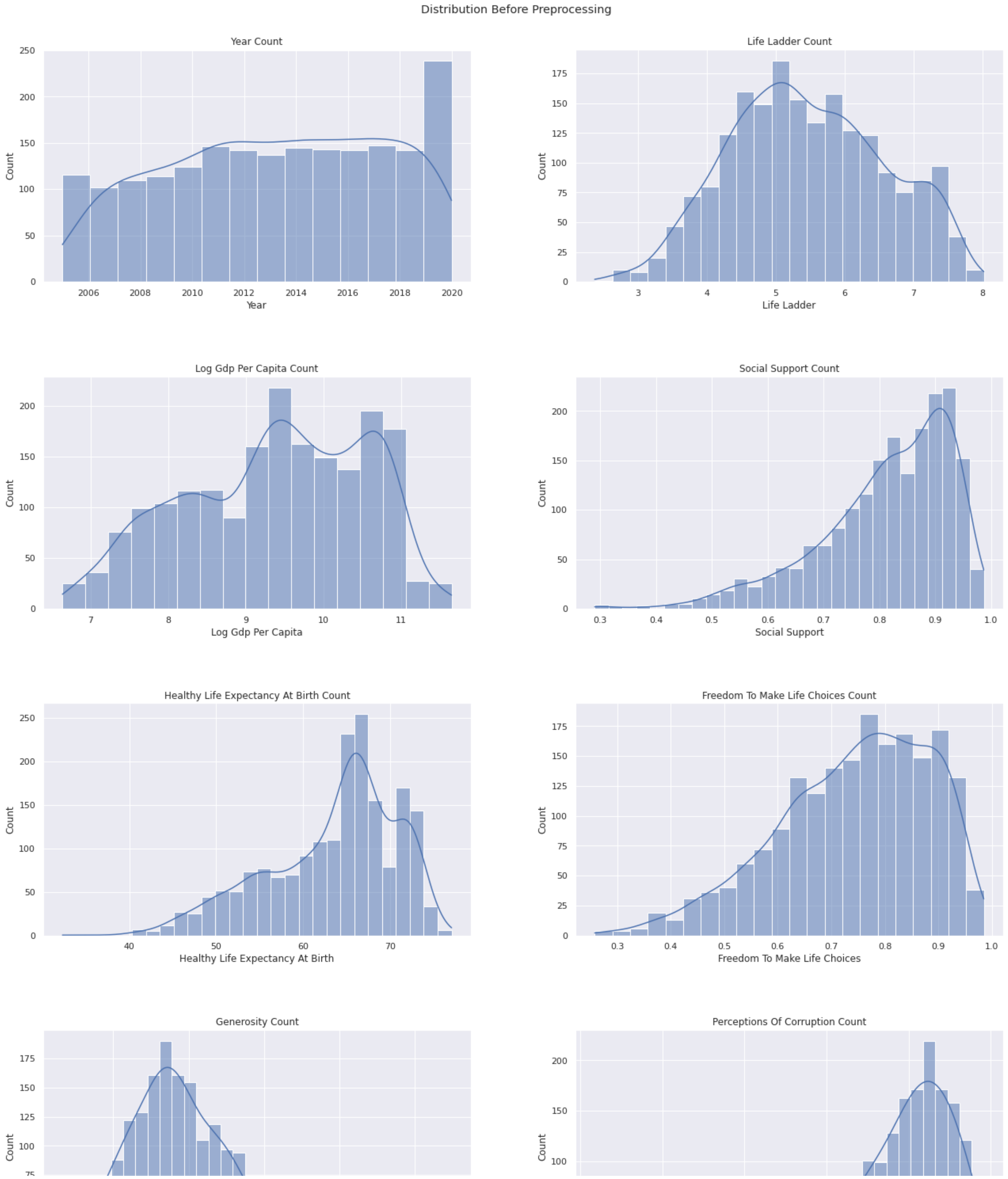
In [37]:

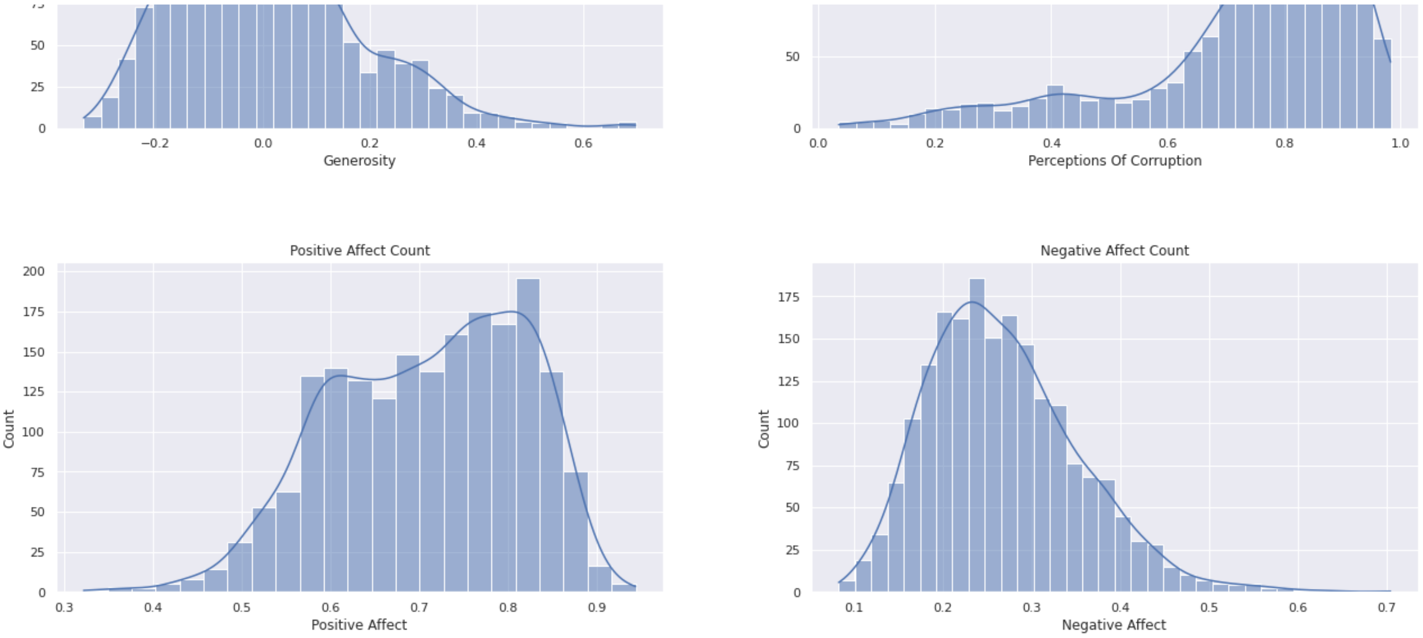
```
try:
  mkdir("Plots")
except:
  pass

mypath = "Plots"
```

In [38]:

```
f, axes = plt.subplots(5, 2, figsize = (20, 30))
f.tight_layout(pad=8)
f.suptitle("Distribution Before Preprocessing")
cols = df.select_dtypes(exclude="object").columns

x_axis = 0
```

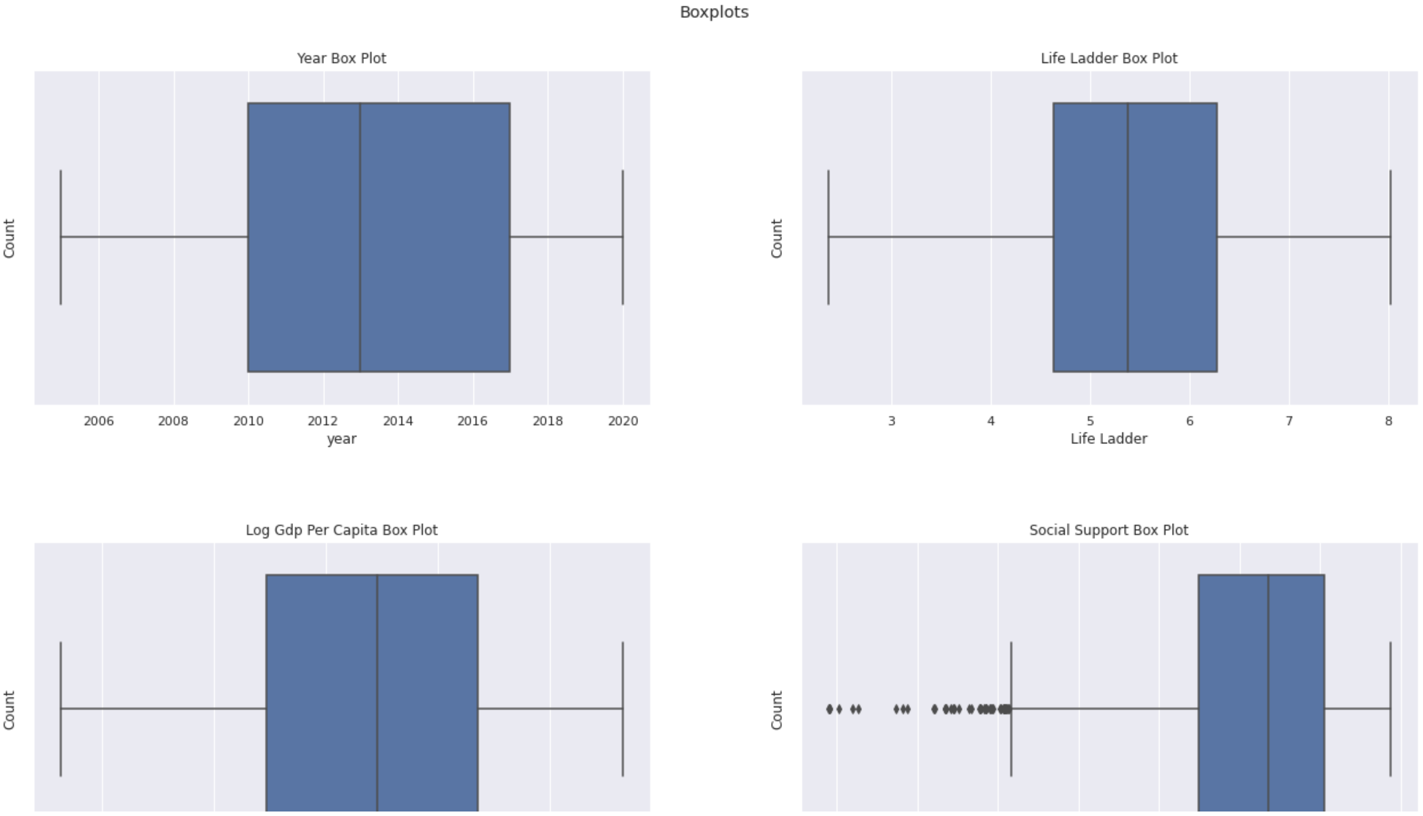
```
y_axis = 0

for col in cols:
  sns.histplot(data=df, x=col, kde=True, ax=axes[x_axis, y_axis])
  axes[x_axis,y_axis].set_xlabel(col.title())
  axes[x_axis,y_axis].set_ylabel("Count")
  axes[x_axis,y_axis].set_title(f"{col.title()} Count")

  if y_axis == 1:
    y_axis = 0
    x_axis += 1
    # continue
  else:
    y_axis += 1

plt.savefig("Plots/histograms.png")
plt.show()
```

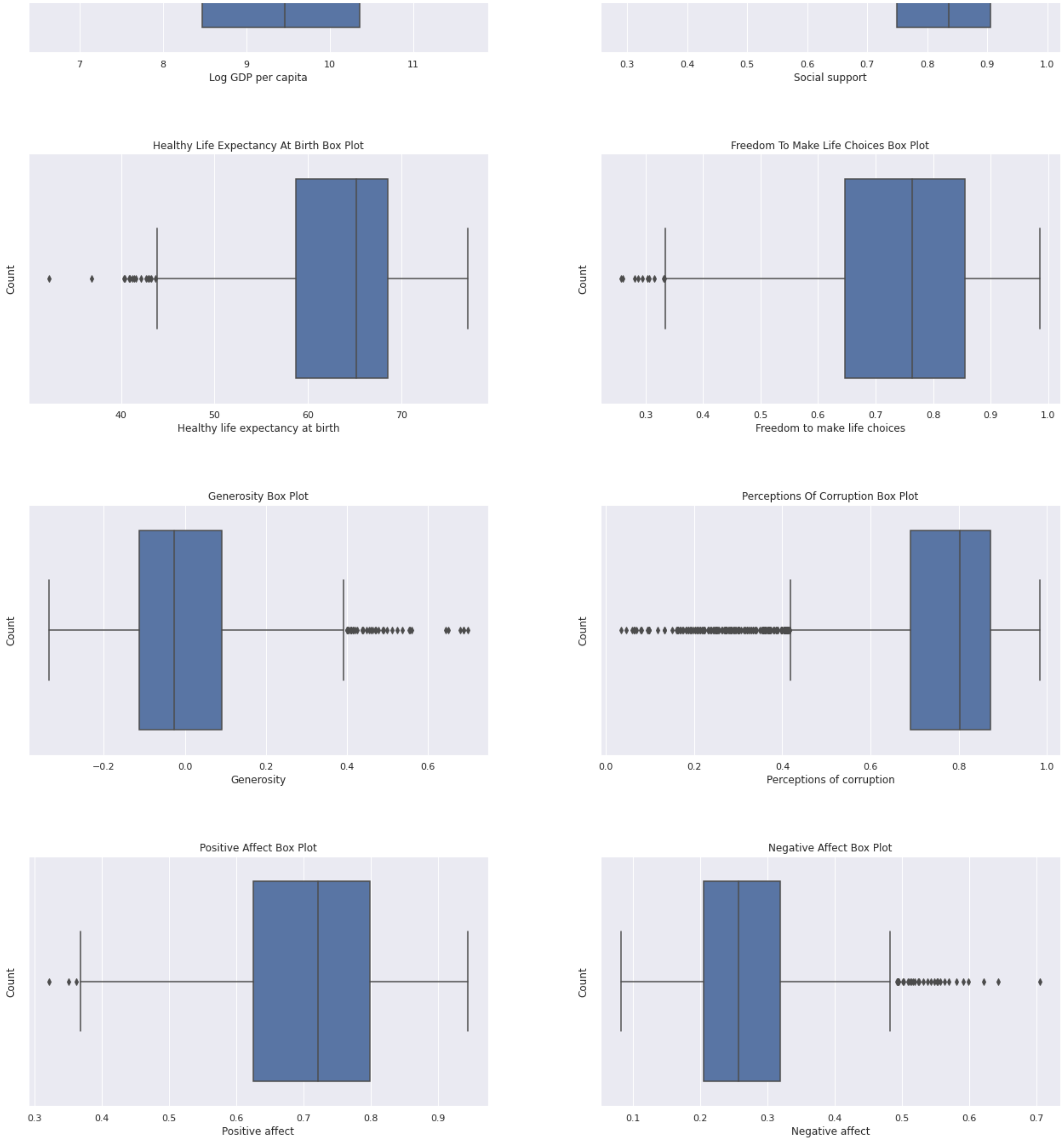




In [39]:

```
f, axes = plt.subplots(5,2, figsize = (20,30))
f.tight_layout(pad=8)
f.suptitle("Boxplots")
cols = df.select_dtypes(exclude="object").columns

x_axis = 0
y_axis = 0

for col in cols:
  sns.boxplot(data=df, x=col, ax=axes[x_axis, y_axis])
  axes[x_axis,y_axis].set_xlabel(col)
  axes[x_axis,y_axis].set_ylabel("Count")
  axes[x_axis,y_axis].set_title(f"{col.title()} Box Plot")
  if y_axis == 1:
    y_axis = 0
    x_axis += 1
  else:
    y_axis += 1

plt.savefig("Plots/boxplots.png")
plt.show()
```





Healthy Life Expectancy At Birth Box Plot

Freedom To Make Life Choices Box Plot

Generosity Box Plot

Perceptions Of Corruption Box Plot

Positive Affect Box Plot

Negative Affect Box Plot

Grouping Data By Year For Visualization

In [40]:

```
year_group = df.groupby("year").sum()
year_group
```

Out[40]:

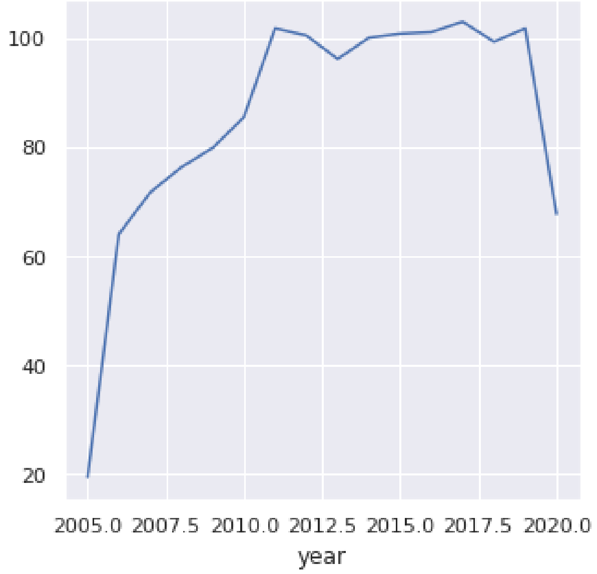
	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption	Positive affect	Negative affect
year									
2005	174.049	273.204	24.230	1809.200	20.738	0.256	17.897	19.464	6.650
2006	462.524	795.897	74.387	5292.980	62.821	0.539	62.727	63.999	22.406
2007	552.664	935.641	80.777	6191.832	69.421	1.513	77.629	71.627	25.522
2008	596.036	1008.400	86.283	6614.995	73.654	2.342	81.761	76.274	26.790

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
2009	622.474	1056.810	92.555	6628.501	76.989	-0.607	84.736	79.789	27.909
2010	681.604	1155.443	99.824	7631.352	86.408	0.416	90.111	85.458	28.881
2011	791.916	1346.593	116.406	8771.909	106.167	-2.079	104.218	101.726	36.328
2012	773.013	1314.683	114.887	8597.511	100.285	-0.274	100.050	100.394	36.958
2013	739.055	1274.760	110.498	8422.422	98.994	-0.021	99.254	96.110	37.157
2014	781.408	1330.678	114.401	8861.433	101.355	2.796	100.413	99.984	38.348
2015	772.777	1315.491	113.345	8849.445	104.815	2.737	97.958	100.723	39.576
2016	766.726	1307.019	115.289	8748.825	106.842	0.096	97.816	101.031	40.564
2017	802.680	1371.967	117.551	9164.903	113.968	-0.952	99.810	102.927	42.517
2018	780.921	1279.317	115.353	8908.642	110.535	-3.140	98.312	99.246	40.977
2019	802.205	1307.830	117.618	9035.544	113.621	-2.725	98.281	101.704	41.043
2020	554.857	858.117	79.697	6173.160	76.870	-0.705	62.990	67.720	27.470

Save Plot as "png"

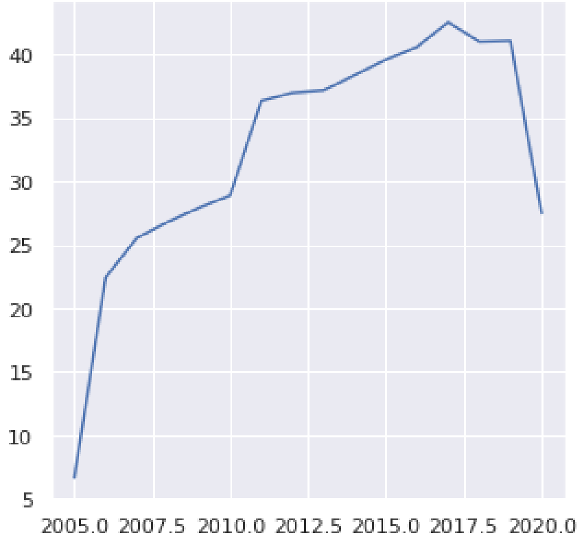
In [41]:

```
year_group["Positive affect"].plot()
plt.savefig("Plots/Positive_Affect_Plot.png")
```



In [42]:

```
year_group["Negative affect"].plot()
plt.savefig("Plots/Negative_Affect_Plot.png")
```



year

In [43]:

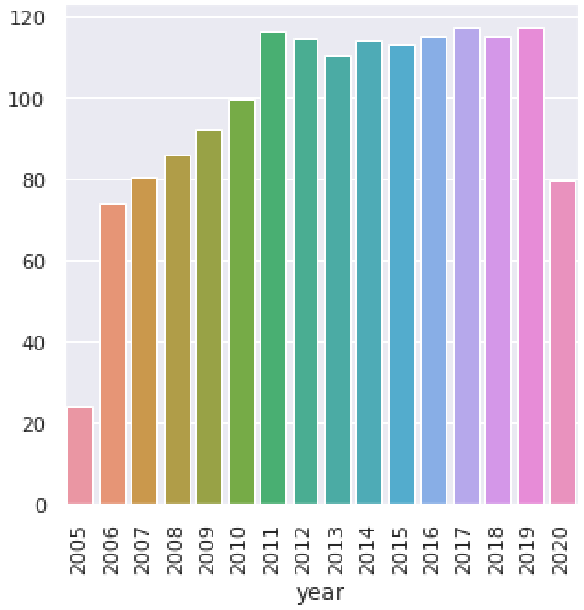
```
year_group
```

Out[43]:

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
2005	174.049	273.204	24.230	1809.200	20.738	0.256	17.897	19.464	6.650
2006	462.524	795.897	74.387	5292.980	62.821	0.539	62.727	63.999	22.406
2007	552.664	935.641	80.777	6191.832	69.421	1.513	77.629	71.627	25.522
2008	596.036	1008.400	86.283	6614.995	73.654	2.342	81.761	76.274	26.790
2009	622.174	1046.810	92.555	6928.501	76.989	-0.607	84.736	79.789	27.909
2010	681.604	1155.443	99.824	7631.352	86.408	0.416	90.111	85.458	28.881
2011	791.916	1346.593	116.406	8771.909	106.167	-2.079	104.218	101.726	36.328
2012	773.013	1314.683	114.887	8597.511	100.285	-0.274	100.050	100.394	36.958
2013	739.055	1274.760	110.498	8422.422	98.994	-0.021	99.254	96.110	37.157
2014	781.408	1330.678	114.401	8861.433	101.355	2.796	100.413	99.984	38.348
2015	772.777	1315.491	113.345	8849.445	104.815	2.737	97.958	100.723	39.576
2016	766.726	1307.019	115.289	8748.825	106.842	0.096	97.816	101.031	40.564
2017	802.680	1371.967	117.551	9164.903	113.968	-0.952	99.810	102.927	42.517
2018	780.921	1279.317	115.353	8908.642	110.535	-3.140	98.312	99.246	40.977
2019	802.205	1307.830	117.618	9035.544	113.621	-2.725	98.281	101.704	41.043
2020	554.857	858.117	79.697	6173.160	76.870	-0.705	62.990	67.720	27.470

In [44]:

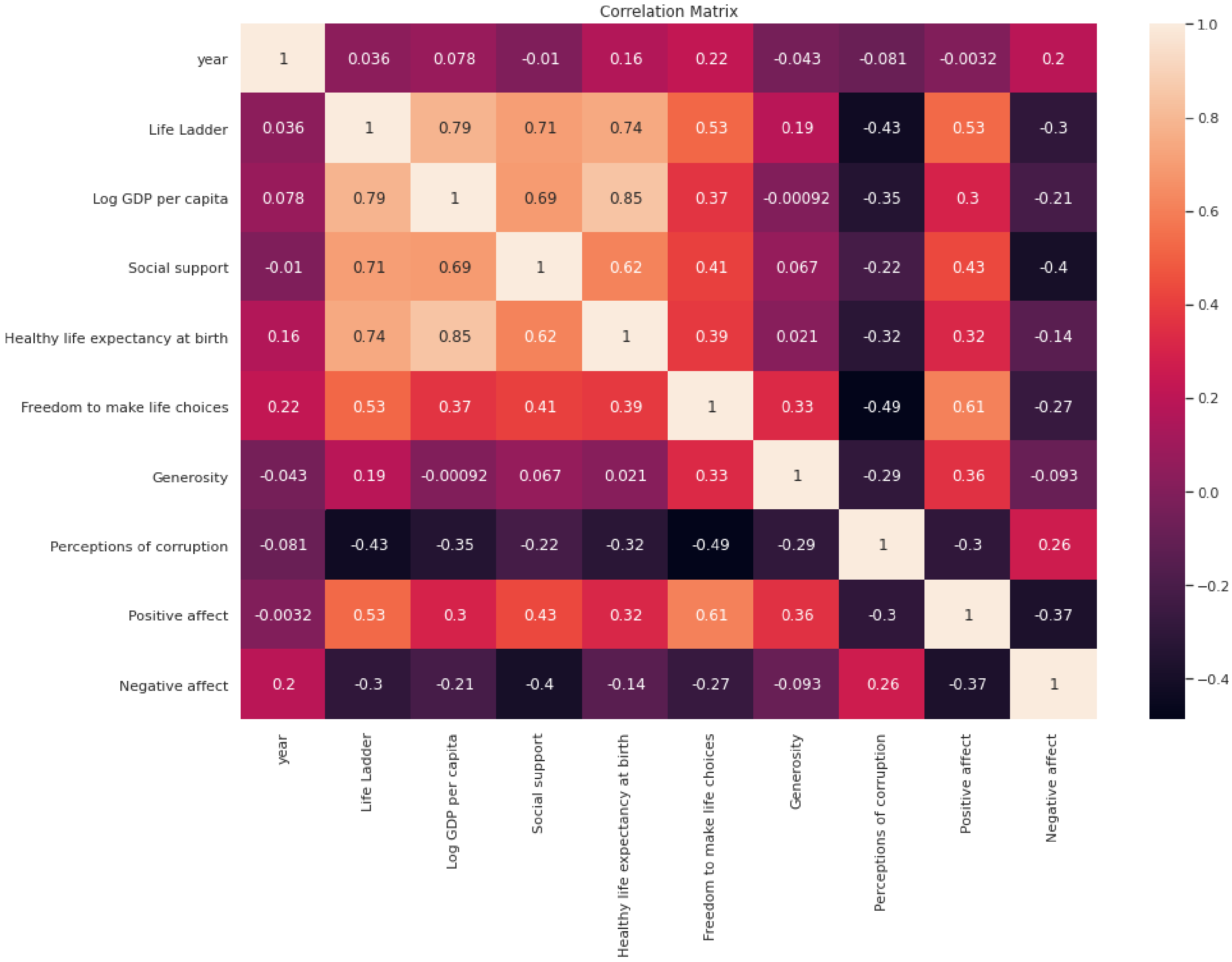
```
ax1 = sns.barplot(x=year_group.index, y=year_group['Social support'].values)
ax1.tick_params(axis='x', rotation=90)
plt.savefig("Plots/SocialSupport.png")
```



In [45]:

```
sns.set(rc={'figure.figsize':(15, 10)})

plt.title('Correlation Matrix')
```

```
sns.heatmap(df.corr(), annot=True)
plt.savefig("Plots/Correlation_matrix.png")
```



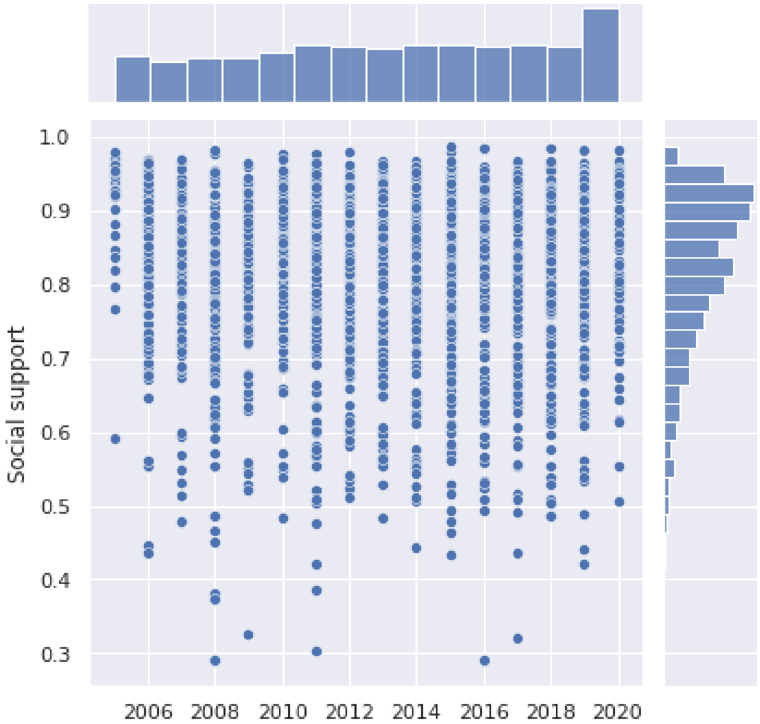
	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption	Positive affect	Negative affect
year	1	0.036	0.078	-0.01	0.16	0.22	-0.043	-0.081	-0.0032	0.2
Life Ladder	0.036	1	0.79	0.71	0.74	0.53	0.19	-0.43	0.53	-0.3
Log GDP per capita	0.078	0.79	1	0.69	0.85	0.37	-0.00092	-0.35	0.3	-0.21
Social support	-0.01	0.71	0.69	1	0.62	0.41	0.067	-0.22	0.43	-0.4
Healthy life expectancy at birth	0.16	0.74	0.85	0.62	1	0.39	0.021	-0.32	0.32	-0.14
Freedom to make life choices	0.22	0.53	0.37	0.41	0.39	1	0.33	-0.49	0.61	-0.27
Generosity	-0.043	0.19	-0.00092	0.067	0.021	0.33	1	-0.29	0.36	-0.093
Perceptions of corruption	-0.081	-0.43	-0.35	-0.22	-0.32	-0.49	-0.29	1	-0.3	0.26
Positive affect	-0.0032	0.53	0.3	0.43	0.32	0.61	0.36	-0.3	1	-0.37
Negative affect	0.2	-0.3	-0.21	-0.4	-0.14	-0.27	-0.093	0.26	-0.37	1

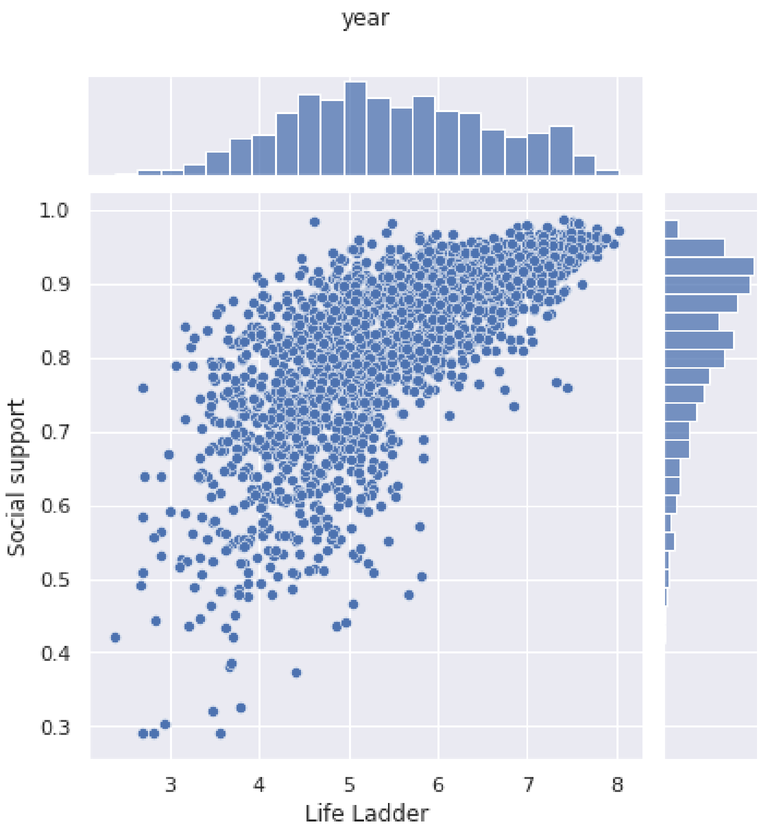
In [46]:

```
sns.jointplot(data=df, x="year", y="Social support")
sns.jointplot(data=df, x="Life Ladder", y="Social support")
```

Out[46]:

```
<seaborn.axisgrid.JointGrid at 0x7f6471666d50>
```





Exporting Reports as PDF

In [47]:

```
import os
from os import listdir, mkdir
from os.path import isfile, join
```

In [48]:

```
all_files = os.listdir("Plots/")
reports = [f"Plots/{file}" for file in all_files]
print(reports)
```

```
['Plots/Correlation_matrix.png', 'Plots/boxplots.png', 'Plots/SocialSupport.png', 'Plots/
Negative_Affect_Plot.png', 'Plots/Positive_Affect_Plot.png', 'Plots/histograms.png']
```

In [49]:

```
!pip install FPDF
```

```
Requirement already satisfied: FPDF in /usr/local/lib/python3.7/dist-packages (1.7.2)
```

In [50]:

```
from fpdf import FPDF
```

In [51]:

```
WIDTH = 210
HEIGHT = 297
```

In [52]:

```
pdf = FPDF()
pdf.set_font("Arial", "B", 56)

pdf.add_page()
pdf.cell(180, 20, txt='REPORT', align='C')

#pdf.add_page()
```

```
for report in reports:
  pdf.add_page()

  pdf.set_font("Arial", "B", 24)
  pdf.cell(180, 20, txt=report, align='C')

  pdf.image(report, 5, 30, WIDTH-5)


pdf.output("Countries_Report.pdf", "F")
```

Out[52]:

```
''
```

In [53]:

```
#os.system("rm -rf Plots")
#os.system("mkdir Plots")
```