In [1]:
```
import matplotlib
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
```

In [2]:
```
df=pd.read_excel('happiness.xls')
```

In [3]:
```
df
```

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
0	Afghanistan	2008	3.723590	7.370100	0.450662	50.799999	0.718114	0.167640	0.881686	0.517637	0.258195
1	Afghanistan	2009	4.401778	7.539972	0.552308	51.200001	0.678896	0.190099	0.850035	0.583926	0.237092
2	Afghanistan	2010	4.758381	7.646709	0.539075	51.599998	0.600127	0.120590	0.706766	0.618265	0.275324
3	Afghanistan	2011	3.831719	7.619532	0.521104	51.919998	0.495901	0.162427	0.731109	0.611387	0.267175
4	Afghanistan	2012	3.782938	7.705479	0.520637	52.240002	0.530935	0.236032	0.775620	0.710385	0.267919
...	...	...	...	...	...	...	...	...	...	...	...
1944	Zimbabwe	2016	3.735400	7.984372	0.768425	54.400002	0.732971	-0.094634	0.723612	0.737636	0.208555
1945	Zimbabwe	2017	3.638300	8.015738	0.754147	55.000000	0.752826	-0.097645	0.751208	0.806428	0.224051
1946	Zimbabwe	2018	3.616480	8.048798	0.775388	55.599998	0.762675	-0.068427	0.844209	0.710119	0.211726
1947	Zimbabwe	2019	2.693523	7.950132	0.759162	56.200001	0.631908	-0.063791	0.830652	0.716004	0.235354
1948	Zimbabwe	2020	3.159802	7.828757	0.717243	56.799999	0.643303	-0.008696	0.788523	0.702573	0.345736

1949 rows × 11 columns

Countries with negative perceptions

In [4]:
```
neg_c=df[['Country name','Perceptions of corruption']].copy()
neg_c.rename(columns={'Country name': 'country', 'Perceptions of corruption': 'corruption'}, inplace=True)
neg_c
```

Out[4]:

	country	corruption
0	Afghanistan	0.881686
1	Afghanistan	0.850035
2	Afghanistan	0.706766
3	Afghanistan	0.731109
4	Afghanistan	0.775620
...	...	...
1944	Zimbabwe	0.723612
1945	Zimbabwe	0.751208
1946	Zimbabwe	0.844209
1947	Zimbabwe	0.830652
1948	Zimbabwe	0.788523

1949 rows × 2 columns

In [5]:
```python
neg_c=neg_c.groupby('country')['corruption'].sum().reset_index()
neg_c
```

Out[5]:

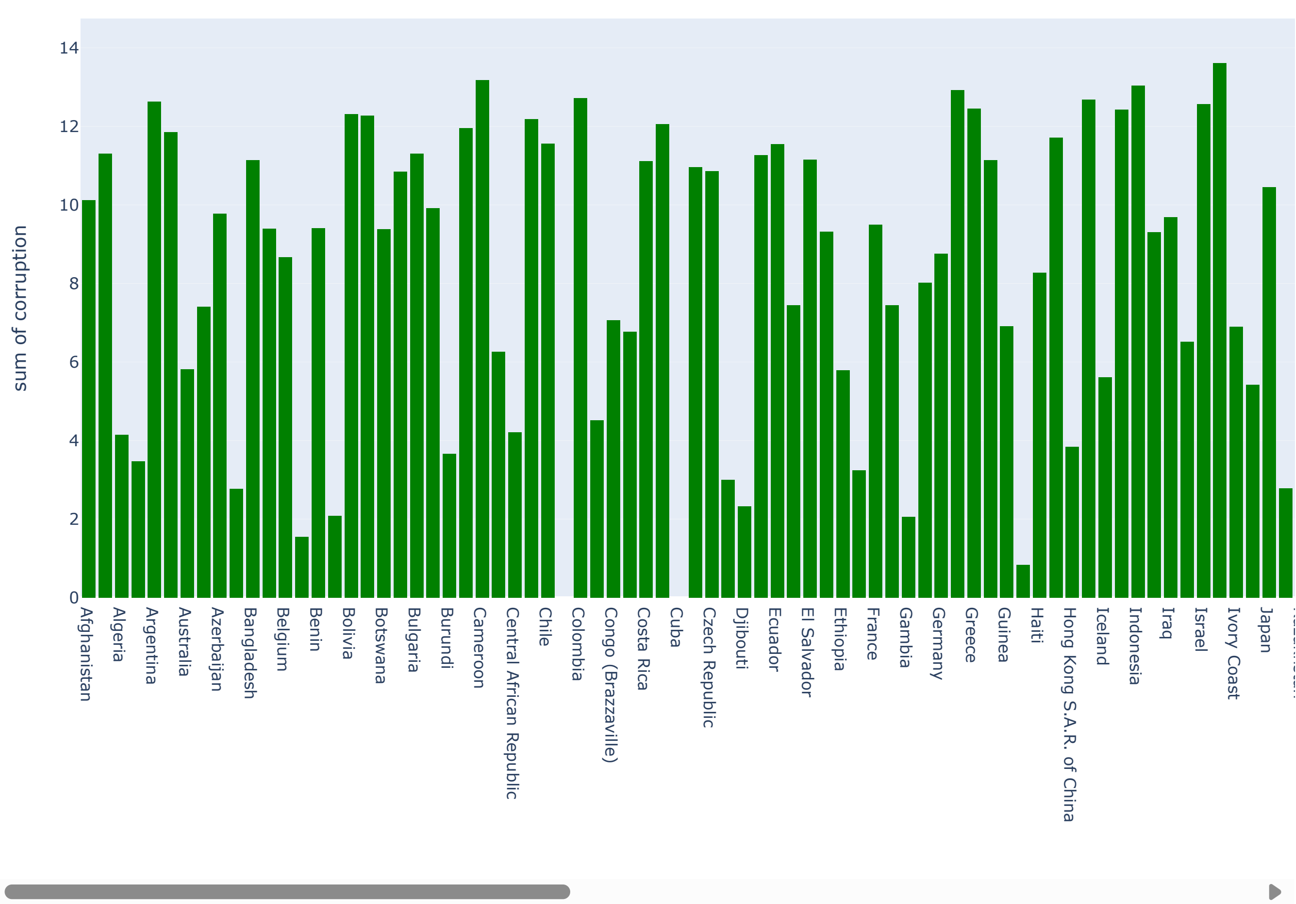
	country	corruption
0	Afghanistan	10.119393
1	Albania	11.301685
2	Algeria	4.145224
3	Angola	3.468072
4	Argentina	12.629954
...	...	...
161	Venezuela	11.928095
162	Vietnam	7.847523
163	Yemen	7.423484
164	Zambia	11.603342
165	Zimbabwe	12.662374

166 rows × 2 columns

In [6]:
```python
fig=px.histogram(neg_c,x='country',y='corruption')
```

In [7]:
```python
fig.update_layout(width=2200, height=700)
fig.update_traces(marker=dict(color='green'))

fig.show()
```



Relation between social support and negative or positive impact?

In [8]:
```
df
```

Out[8]:

	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	3.723590	7.370100	0.450662	50.799999	0.718114	0.167640	0.881686	0.517637	0.258195
1	Afghanistan	2009	4.401778	7.539972	0.552308	51.200001	0.678896	0.190099	0.850035	0.583926	0.237092
2	Afghanistan	2010	4.758381	7.646709	0.539075	51.599998	0.600127	0.120590	0.706766	0.618265	0.275324
3	Afghanistan	2011	3.831719	7.619532	0.521104	51.919998	0.495901	0.162427	0.731109	0.611387	0.267175
4	Afghanistan	2012	3.782938	7.705479	0.520637	52.240002	0.530935	0.236032	0.775620	0.710385	0.267919
...	...	...	...	...	...	...	...	...	...	...	...
1944	Zimbabwe	2016	3.735400	7.984372	0.768425	54.400002	0.732971	-0.094634	0.723612	0.737636	0.208555
1945	Zimbabwe	2017	3.638300	8.015738	0.754147	55.000000	0.752826	-0.097645	0.751208	0.806428	0.224051
1946	Zimbabwe	2018	3.616480	8.048798	0.775388	55.599998	0.762675	-0.068427	0.844209	0.710119	0.211726
1947	Zimbabwe	2019	2.693523	7.950132	0.759162	56.200001	0.631908	-0.063791	0.830652	0.716004	0.235354
1948	Zimbabwe	2020	3.159802	7.828757	0.717243	56.799999	0.643303	-0.008696	0.788523	0.702573	0.345736

1949 rows × 11 columns

In [9]:
```
soc_c=df[['Country name','Social support','Negative affect','Positive affect']].copy()
soc_c
```

Out[9]:

	Country name	Social support	Negative affect	Positive affect
0	Afghanistan	0.450662	0.258195	0.517637
1	Afghanistan	0.552308	0.237092	0.583926
2	Afghanistan	0.539075	0.275324	0.618265
3	Afghanistan	0.521104	0.267175	0.611387
4	Afghanistan	0.520637	0.267919	0.710385
...	...	...	...	...
1944	Zimbabwe	0.768425	0.208555	0.737636
1945	Zimbabwe	0.754147	0.224051	0.806428
1946	Zimbabwe	0.775388	0.211726	0.710119
1947	Zimbabwe	0.759162	0.235354	0.716004
1948	Zimbabwe	0.717243	0.345736	0.702573

1949 rows × 4 columns

In [10]:
```
soc_c.rename(columns={'Country name': 'country', 'Negative affect': 'negative','Social support':'soc_support','Positive affect':'
soc_c
```

Out[10]:

	country	soc_support	negative	positive
0	Afghanistan	0.450662	0.258195	0.517637
1	Afghanistan	0.552308	0.237092	0.583926
2	Afghanistan	0.539075	0.275324	0.618265
3	Afghanistan	0.521104	0.267175	0.611387
4	Afghanistan	0.520637	0.267919	0.710385
...	...	...	...	...
1944	Zimbabwe	0.768425	0.208555	0.737636
1945	Zimbabwe	0.754147	0.224051	0.806428
1946	Zimbabwe	0.775388	0.211726	0.710119
1947	Zimbabwe	0.759162	0.235354	0.716004
1948	Zimbabwe	0.717243	0.345736	0.702573

1949 rows × 4 columns

In [11]:
```
soc_c1=soc_c[['soc_support','negative','positive']].copy()
```

In [12]:
```
soc_c1
```

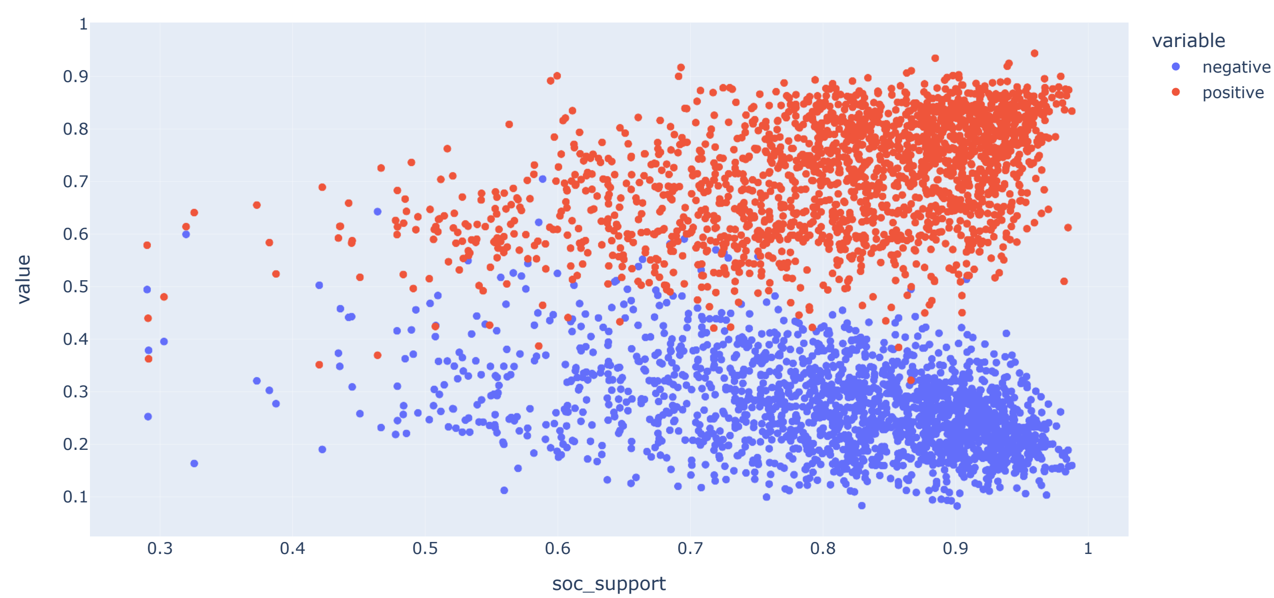
Out[12]:

	soc_support	negative	positive
0	0.450662	0.258195	0.517637
1	0.552308	0.237092	0.583926
2	0.539075	0.275324	0.618265
3	0.521104	0.267175	0.611387
4	0.520637	0.267919	0.710385
...	...	...	...
1944	0.768425	0.208555	0.737636
1945	0.754147	0.224051	0.806428
1946	0.775388	0.211726	0.710119
1947	0.759162	0.235354	0.716004
1948	0.717243	0.345736	0.702573

1949 rows × 3 columns

In [13]:
```
fig=px.scatter(soc_c1,x='soc_support',y=['negative','positive'])
fig.show()


#The reason for negative impact is not social support
```



In [14]:
```
df
```

Out[14]:

	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	3.723590	7.370100	0.450662	50.799999	0.718114	0.167640	0.881686	0.517637	0.258195
1	Afghanistan	2009	4.401778	7.539972	0.552308	51.200001	0.678896	0.190099	0.850035	0.583926	0.237092
2	Afghanistan	2010	4.758381	7.646709	0.539075	51.599998	0.600127	0.120590	0.706766	0.618265	0.275324
3	Afghanistan	2011	3.831719	7.619532	0.521104	51.919998	0.495901	0.162427	0.731109	0.611387	0.267175
4	Afghanistan	2012	3.782938	7.705479	0.520637	52.240002	0.530935	0.236032	0.775620	0.710385	0.267919
...	...	...	...	...	...	...	...	...	...	...	...
1944	Zimbabwe	2016	3.735400	7.984372	0.768425	54.400002	0.732971	-0.094634	0.723612	0.737636	0.208555
1945	Zimbabwe	2017	3.638300	8.015738	0.754147	55.000000	0.752826	-0.097645	0.751208	0.806428	0.224051
1946	Zimbabwe	2018	3.616480	8.048798	0.775388	55.599998	0.762675	-0.068427	0.844209	0.710119	0.211726
1947	Zimbabwe	2019	2.693523	7.950132	0.759162	56.200001	0.631908	-0.063791	0.830652	0.716004	0.235354
1948	Zimbabwe	2020	3.159802	7.828757	0.717243	56.799999	0.643303	-0.008696	0.788523	0.702573	0.345736

1949 rows × 11 columns

Corruption throughout the years

In [15]:
```
year_c=df[['year','Perceptions of corruption']].copy()
year_c
```

Out[15]:

	year	Perceptions of corruption
0	2008	0.881686
1	2009	0.850035
2	2010	0.706766
3	2011	0.731109
4	2012	0.775620
...	...	...
1944	2016	0.723612
1945	2017	0.751208
1946	2018	0.844209
1947	2019	0.830652
1948	2020	0.788523

1949 rows × 2 columns

In [16]:
```
year_c.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1949 entries, 0 to 1948
Data columns (total 2 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   year                       1949 non-null   int64  
 1   Perceptions of corruption  1839 non-null   float64
dtypes: float64(1), int64(1)
memory usage: 30.6 KB
```

In [18]:
```
# year_c.duplicates
```

In [ ]:
```

```