

```
In [1]: #world-happiness-report-2021.csv
```

```
In [2]: import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [3]: mpl.rcParams['figure.dpi'] = 200
mpl.rcParams['axes.spines.top'] = False
mpl.rcParams['axes.spines.right'] = False
```

```
In [5]: df = pd.read_csv(r'/Users/clairekraft/Desktop/Python/data/world-happiness-report-2021.csv')
```

In [6]: `df.head()`

Out [6]:

	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	Perceived corruption
0	Finland	Western Europe	7.842	0.032	7.904	7.780	10.775	0.954	72.0	0.949	-0.098	
1	Denmark	Western Europe	7.620	0.035	7.687	7.552	10.933	0.954	72.7	0.946	0.030	
2	Switzerland	Western Europe	7.571	0.036	7.643	7.500	11.117	0.942	74.4	0.919	0.025	
3	Iceland	Western Europe	7.554	0.059	7.670	7.438	10.878	0.983	73.0	0.955	0.160	
4	Netherlands	Western Europe	7.464	0.027	7.518	7.410	10.932	0.942	72.4	0.913	0.175	

In [8]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 149 entries, 0 to 148
```

```
Data columns (total 20 columns):
```

#	Column	Non-Null Count	Dtype
0	Country name	149 non-null	object
1	Regional indicator	149 non-null	object
2	Ladder score	149 non-null	float64
3	Standard error of ladder score	149 non-null	float64
4	upperwhisker	149 non-null	float64
5	lowerwhisker	149 non-null	float64
6	Logged GDP per capita	149 non-null	float64
7	Social support	149 non-null	float64
8	Healthy life expectancy	149 non-null	float64
9	Freedom to make life choices	149 non-null	float64
10	Generosity	149 non-null	float64
11	Perceptions of corruption	149 non-null	float64
12	Ladder score in Dystopia	149 non-null	float64
13	Explained by: Log GDP per capita	149 non-null	float64
14	Explained by: Social support	149 non-null	float64
15	Explained by: Healthy life expectancy	149 non-null	float64
16	Explained by: Freedom to make life choices	149 non-null	float64
17	Explained by: Generosity	149 non-null	float64
18	Explained by: Perceptions of corruption	149 non-null	float64
19	Dystopia + residual	149 non-null	float64

```
dtypes: float64(18), object(2)
```

```
memory usage: 23.4+ KB
```

In [9]: df.info

```
Out[9]: <bound method DataFrame.info of      Country name  Regional indicator  Ladder score  \
0      Finland      Western Europe      7.842
1      Denmark      Western Europe      7.620
2    Switzerland      Western Europe      7.571
3      Iceland      Western Europe      7.554
4    Netherlands      Western Europe      7.464
...
144    Lesotho      Sub-Saharan Africa      3.512
145    Botswana      Sub-Saharan Africa      3.467
146      Rwanda      Sub-Saharan Africa      3.415
147    Zimbabwe      Sub-Saharan Africa      3.145
148  Afghanistan      South Asia      2.523

      Standard error of ladder score  upperwhisker  lowerwhisker  \
0                                0.032      7.904      7.780
1                                0.035      7.687      7.552
2                                0.036      7.643      7.500
3                                0.059      7.670      7.438
4                                0.027      7.518      7.410
```

```
In [10]: df.describe().T.style.bar(subset=['mean'], color='#205ff2')\
        .background_gradient(subset=['std'], cmap='Reds')\
        .background_gradient(subset=['50%'], cmap='coolwarm')
```

Out[10]:

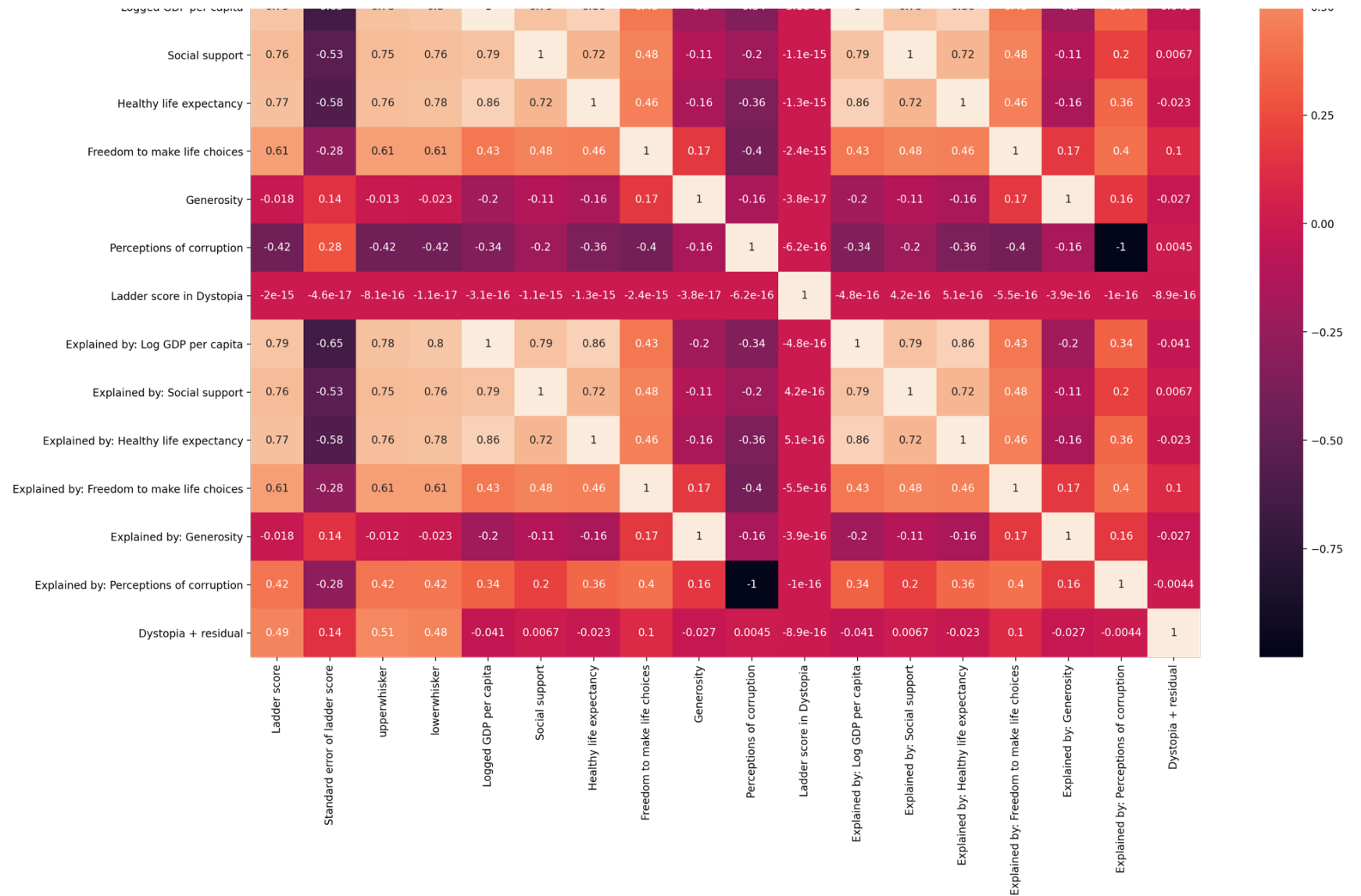
	count	mean	std	min	25%	50%	75%	max
Ladder score	149.000000	5.532839	1.073924	2.523000	4.852000	5.534000	6.255000	7.842000
Standard error of ladder score	149.000000	0.058752	0.022001	0.026000	0.043000	0.054000	0.070000	0.173000
upperwhisker	149.000000	5.648007	1.054330	2.596000	4.991000	5.625000	6.344000	7.904000
lowerwhisker	149.000000	5.417631	1.094879	2.449000	4.706000	5.413000	6.128000	7.780000
Logged GDP per capita	149.000000	9.432208	1.158601	6.635000	8.541000	9.569000	10.421000	11.647000

Social support	149.000000	0.814745	0.114889	0.463000	0.750000	0.832000	0.905000	0.983000
Healthy life expectancy	149.000000	64.992799	6.762043	48.478000	59.802000	66.603000	69.600000	76.953000
Freedom to make life choices	149.000000	0.791597	0.113332	0.382000	0.718000	0.804000	0.877000	0.970000
Generosity	149.000000	-0.015134	0.150657	-0.288000	-0.126000	-0.036000	0.079000	0.542000
Perceptions of corruption	149.000000	0.727450	0.179226	0.082000	0.667000	0.781000	0.845000	0.939000
Ladder score in Dystopia	149.000000	2.430000	0.000000	2.430000	2.430000	2.430000	2.430000	2.430000
Explained by: Log GDP per capita	149.000000	0.977161	0.404740	0.000000	0.666000	1.025000	1.323000	1.751000
Explained by: Social support	149.000000	0.793315	0.258871	0.000000	0.647000	0.832000	0.996000	1.172000
Explained by: Healthy life expectancy	149.000000	0.520161	0.213019	0.000000	0.357000	0.571000	0.665000	0.897000
Explained by: Freedom to make life choices	149.000000	0.498711	0.137888	0.000000	0.409000	0.514000	0.603000	0.716000
Explained by: Generosity	149.000000	0.178047	0.098270	0.000000	0.105000	0.164000	0.239000	0.541000
Explained by: Perceptions of corruption	149.000000	0.135141	0.114361	0.000000	0.060000	0.101000	0.174000	0.547000
Dystopia + residual	149.000000	2.430329	0.537645	0.648000	2.138000	2.509000	2.794000	3.482000

```
In [11]: plt.figure(figsize = (20,15))
sns.heatmap(df.corr(), annot = True)
```

```
Out[11]: <AxesSubplot:>
```

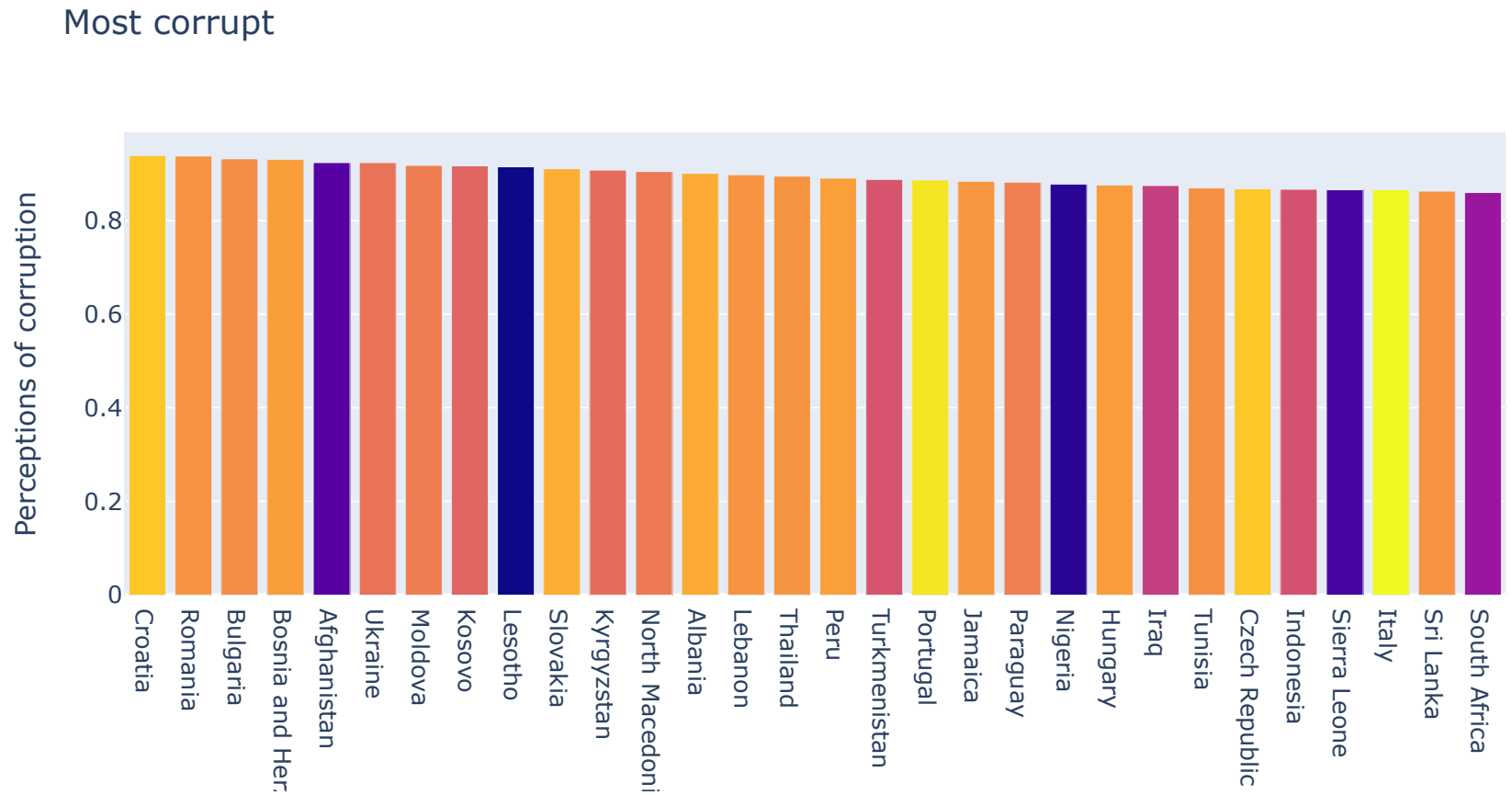




```
In [12]: ['Perceptions of corruption', 'Healthy life expectancy']].sort_values('Perceptions of corruption',
```

```
In [16]: import plotly.express as px
```

```
fig = px.bar(high_corruption[:30], x='Country name', y='Perceptions of corruption', color='Health', title = 'Most corrupt')  
fig.show()
```

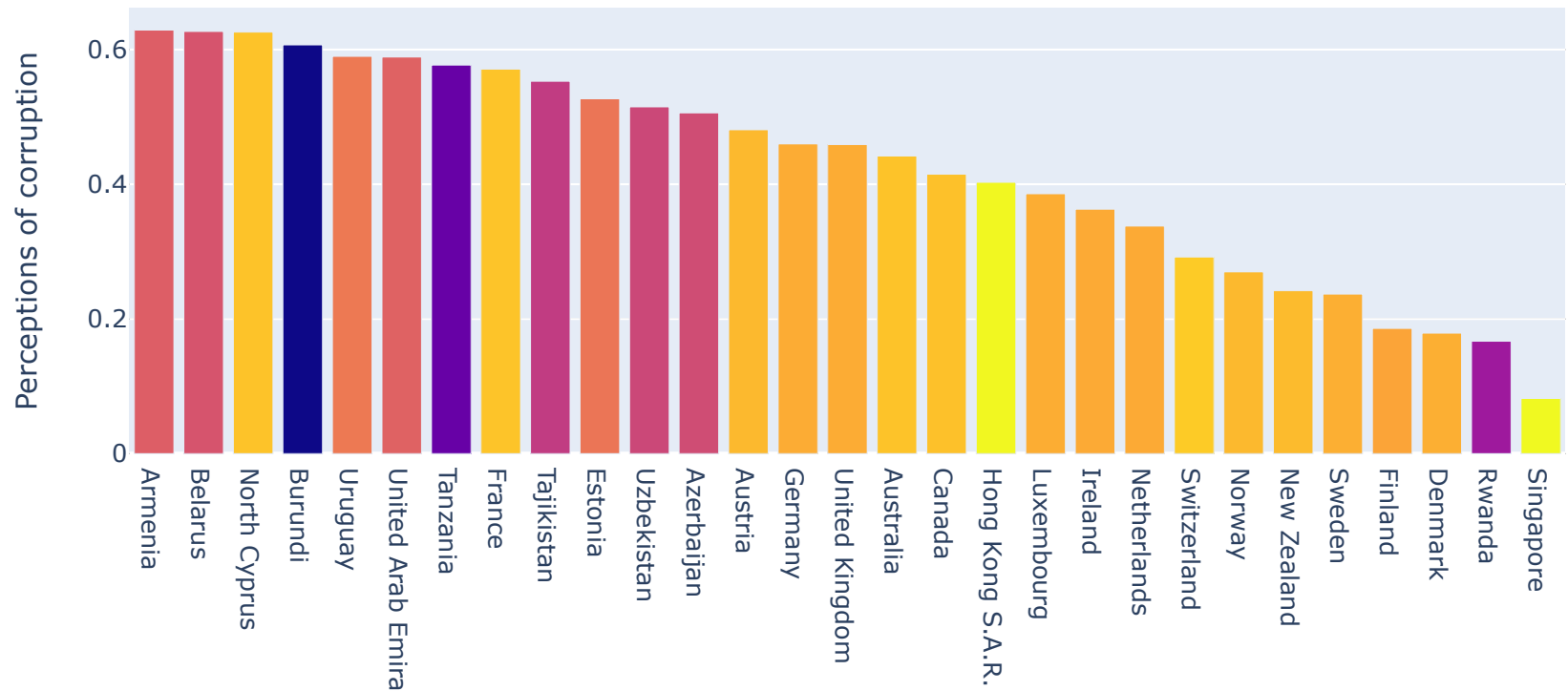


```
In [17]: #Top 30 countries with the highest (perceived) corruption
```



```
In [18]: fig = px.bar(high_corruption[120:], x='Country name', y='Perceptions of corruption', color='Hea  
         title = 'Highest to least corrupt')  
fig.show()
```

Highest to least corrupt

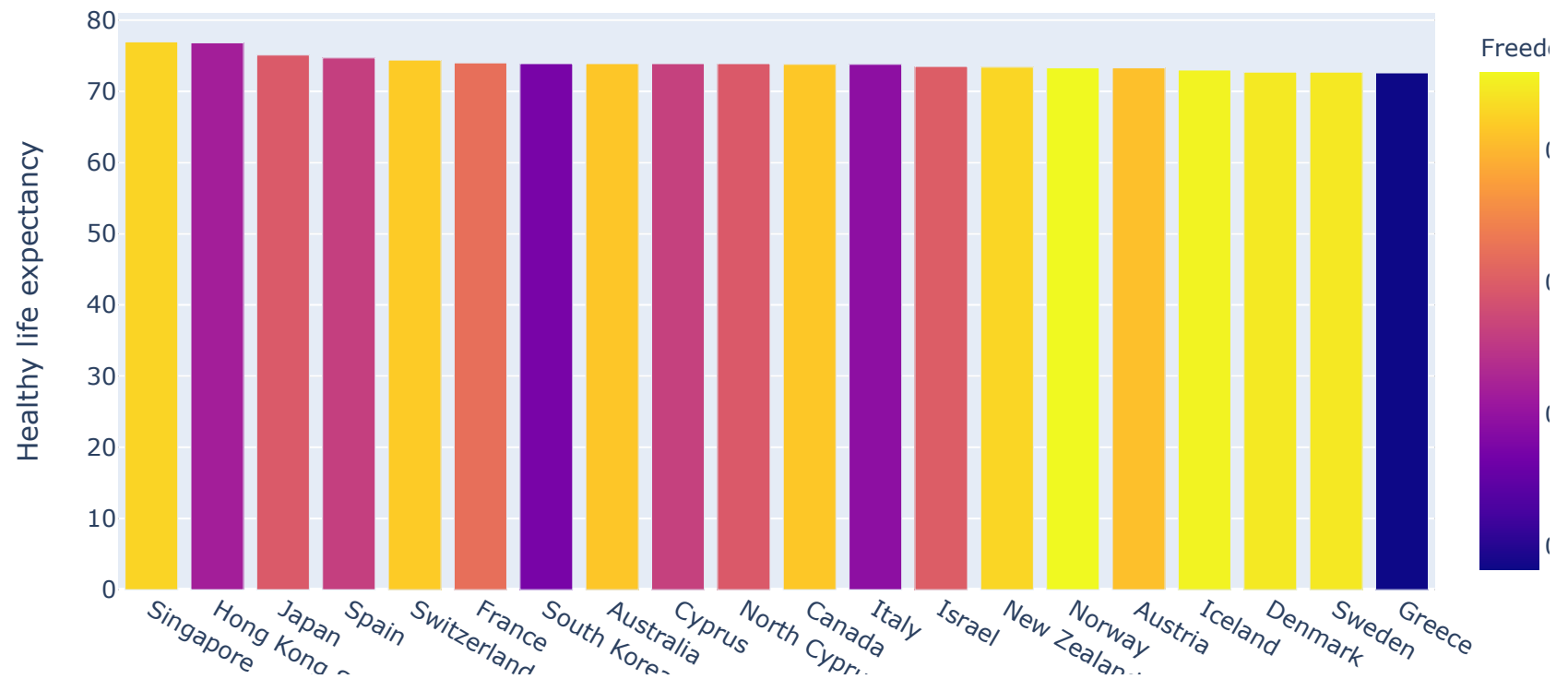


```
In [19]: #Singapore is the least corrupt
```

```
In [20]: Freedom to make life choices', 'Healthy life expectancy' ]].sort_values('Healthy life expectancy',
```

```
In [22]: fig = px.bar(life_exp[:20], x='Country name', y='Healthy life expectancy', color='Freedom to make life choices',
                    title = 'Life expectancy and freedom to make life choices')
fig.show()
```

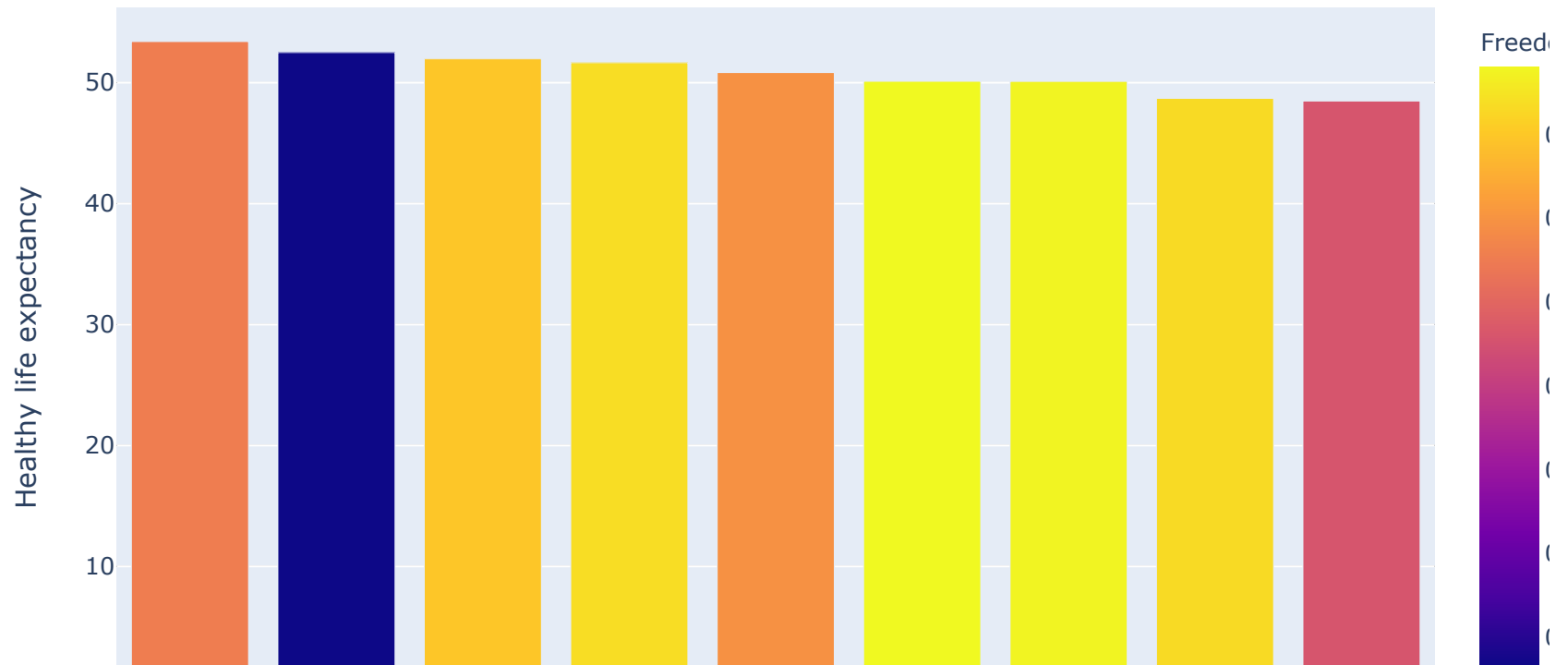
Life expectancy and freedom to make life choices



In [23]: *#Singapore wins for having the highest life expectancy AND freedom or autonomy*

```
In [25]: fig = px.bar(life_exp[140:], x='Country name', y='Healthy life expectancy', color='Freedom to make life choices',  
                    title = 'Life expectancy and freedom to make life choices')  
fig.show()
```

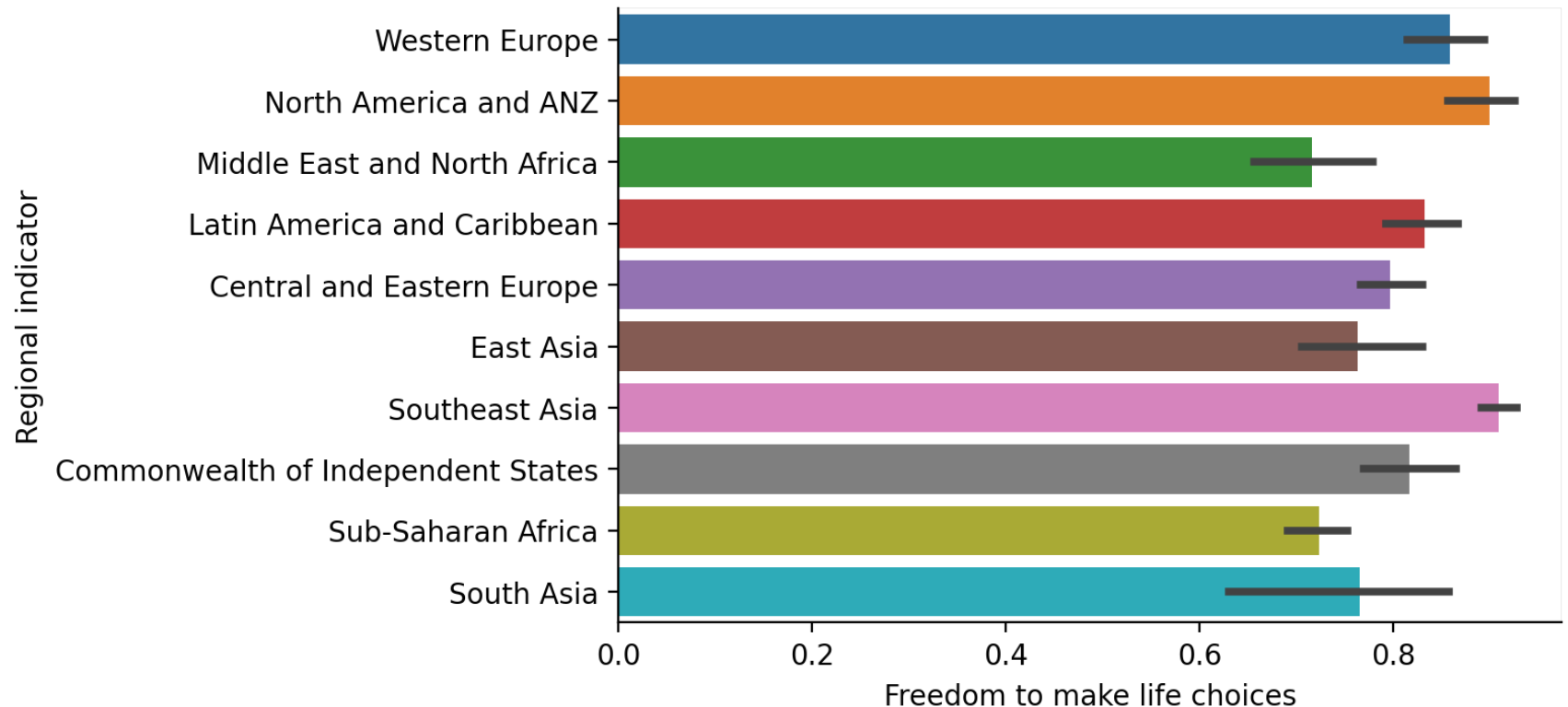
Life expectancy and freedom to make life choices



In [26]: *#This graph shows the other end of the spectrum with the lowest life expectancy and least freedom*

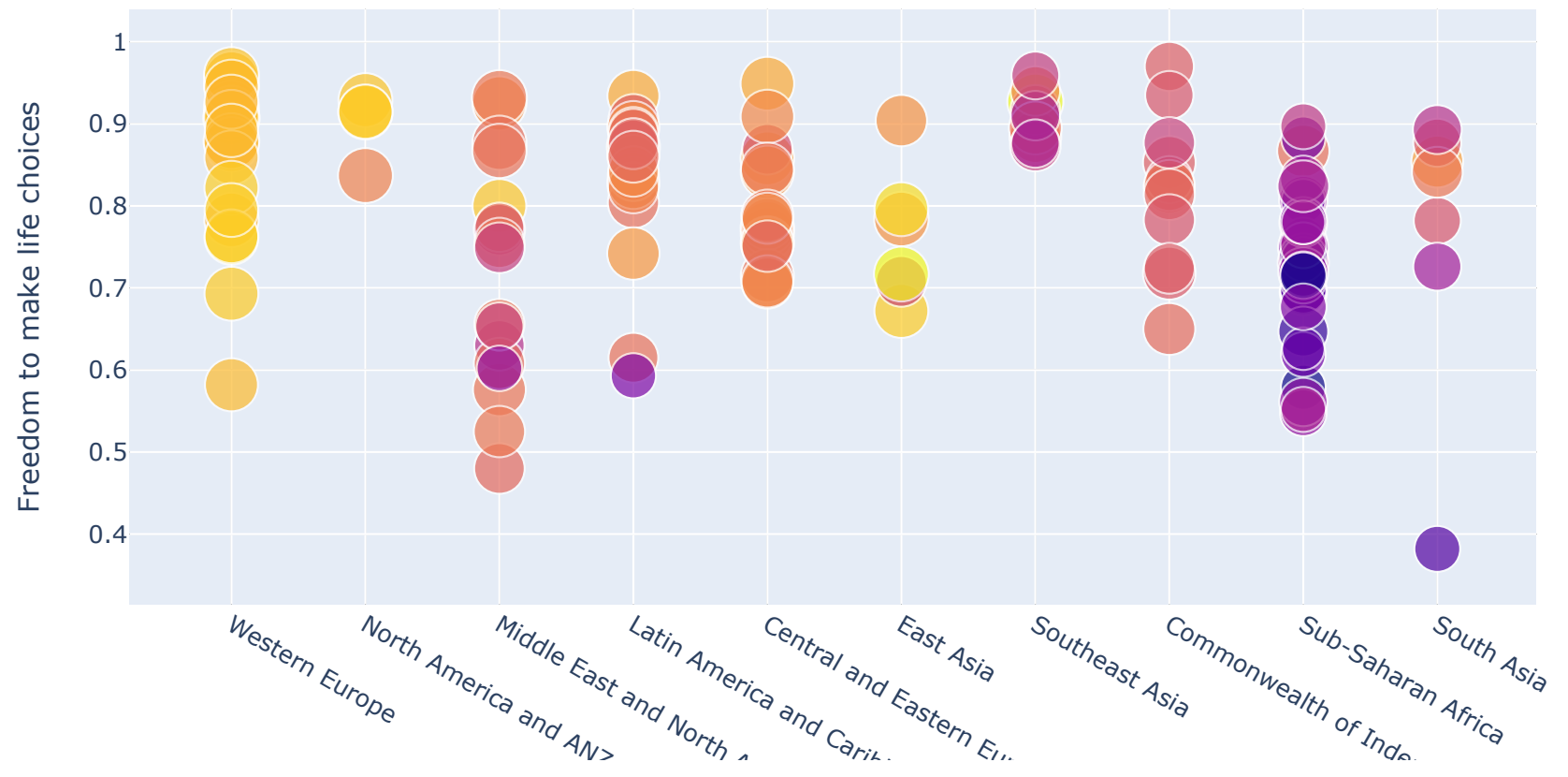
In [27]: `sns.barplot(y = 'Regional indicator', x = 'Freedom to make life choices', data = df)`

Out[27]: `<AxesSubplot:xlabel='Freedom to make life choices', ylabel='Regional indicator'>`



In [28]: *#Freedom broken down by region.
#The Middle East, N Africa, and Sub-Saharan Africa have the least freedom.*

```
In [29]: fig = px.scatter(df, x='Regional indicator', y="Freedom to make life choices", color="Healthy li
size='Logged GDP per capita', hover_data=['Social support'])
fig.show()
```

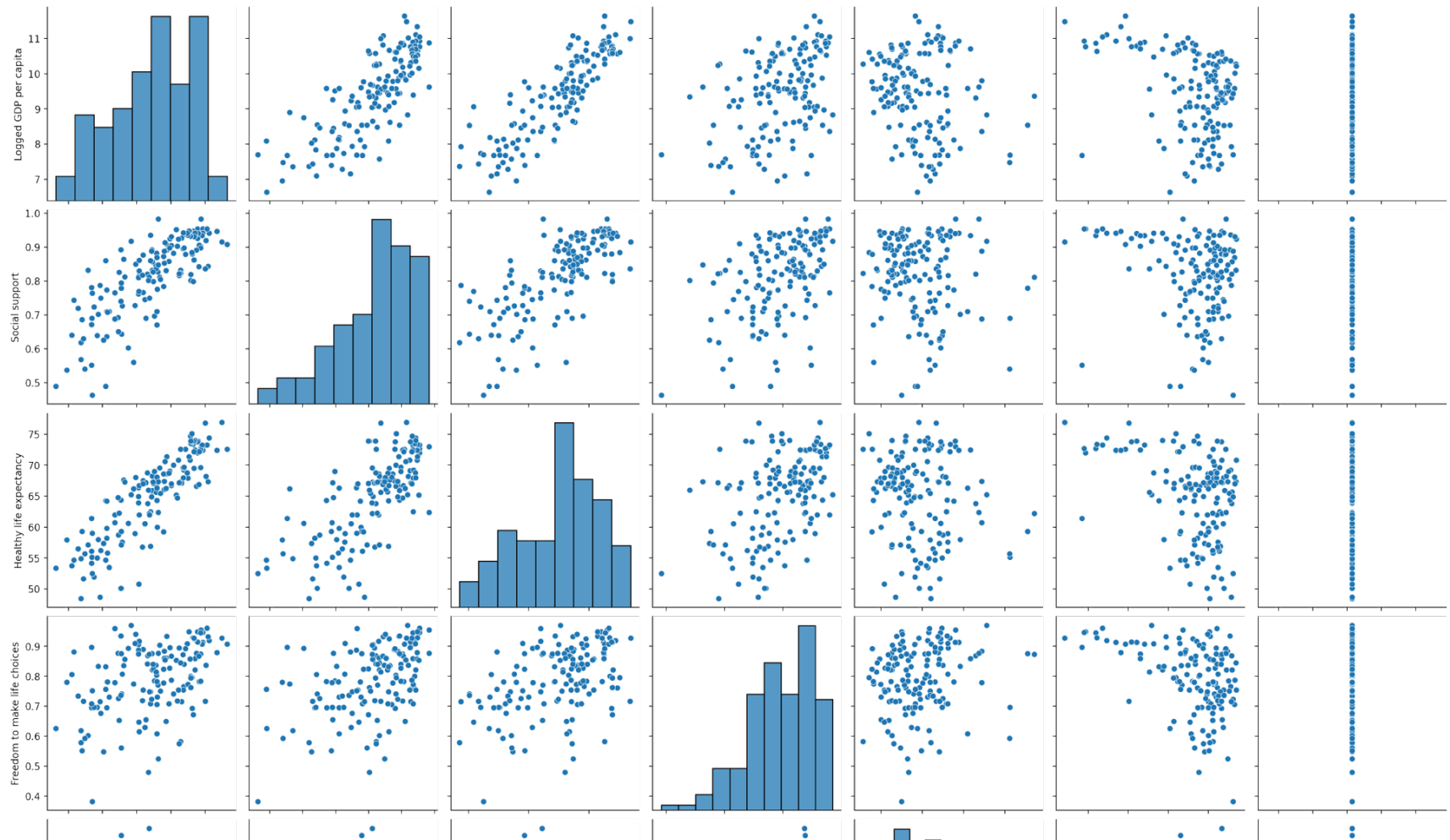


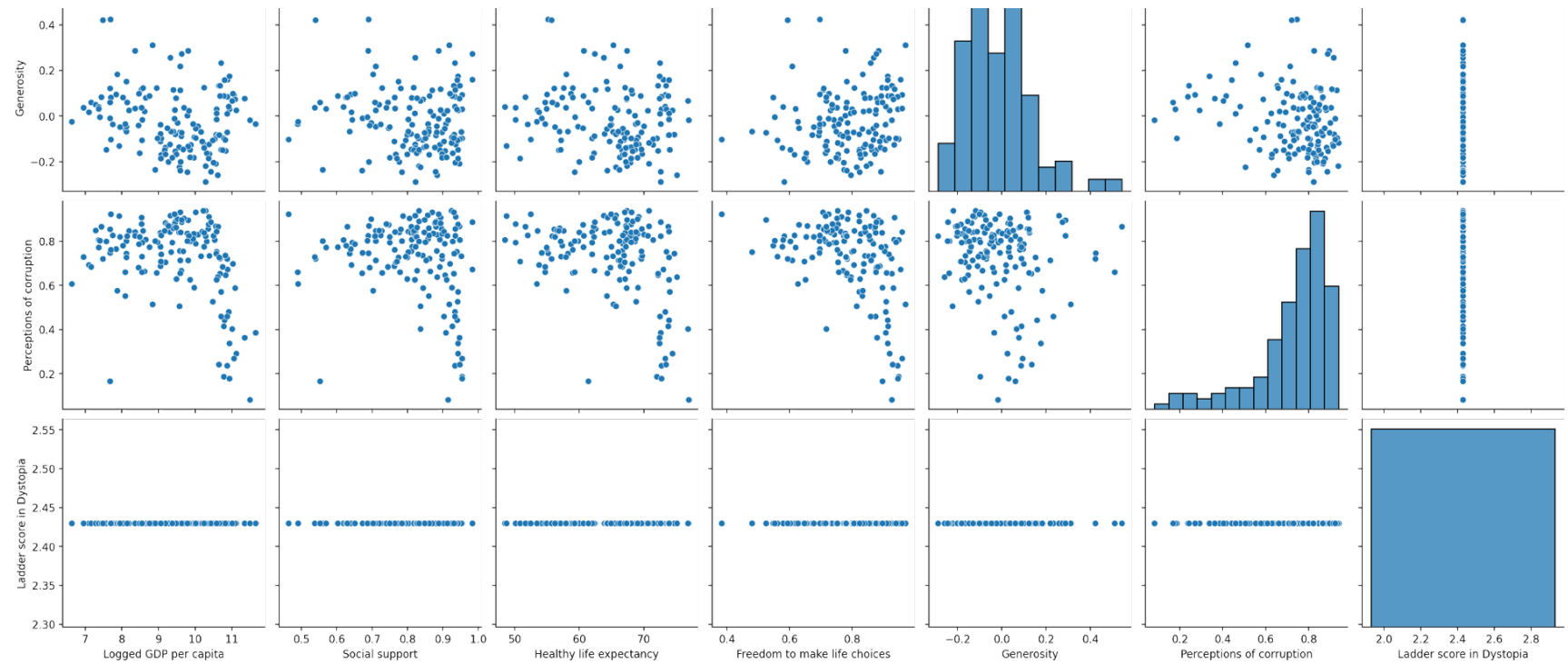
In [30]: *#Life expectancy browken down by region.*

```
In [32]: cols = ['Logged GDP per capita', 'Social support', 'Healthy life expectancy', 'Freedom to make li
              'Generosity', 'Perceptions of corruption', 'Ladder score in Dystopia']

sns.pairplot(df[cols], height = 3)
```

Out[32]: <seaborn.axisgrid.PairGrid at 0x7fd39d4d08b0>

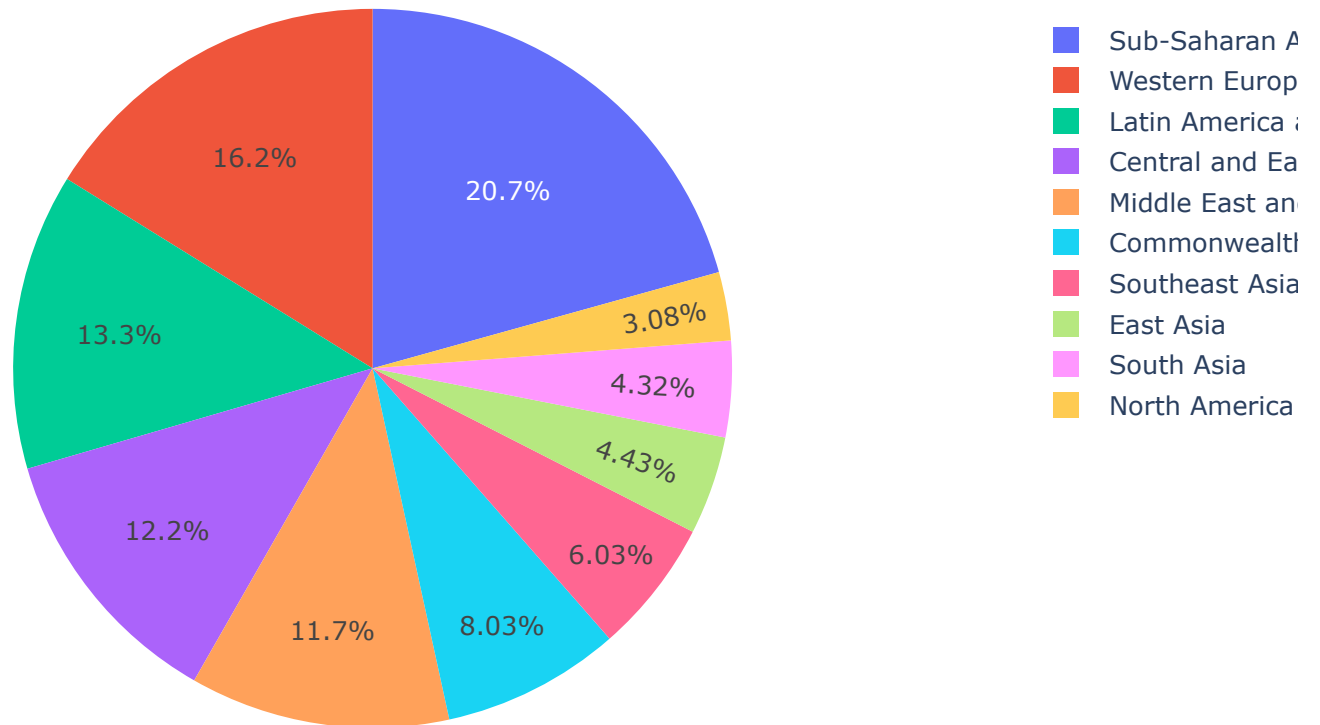




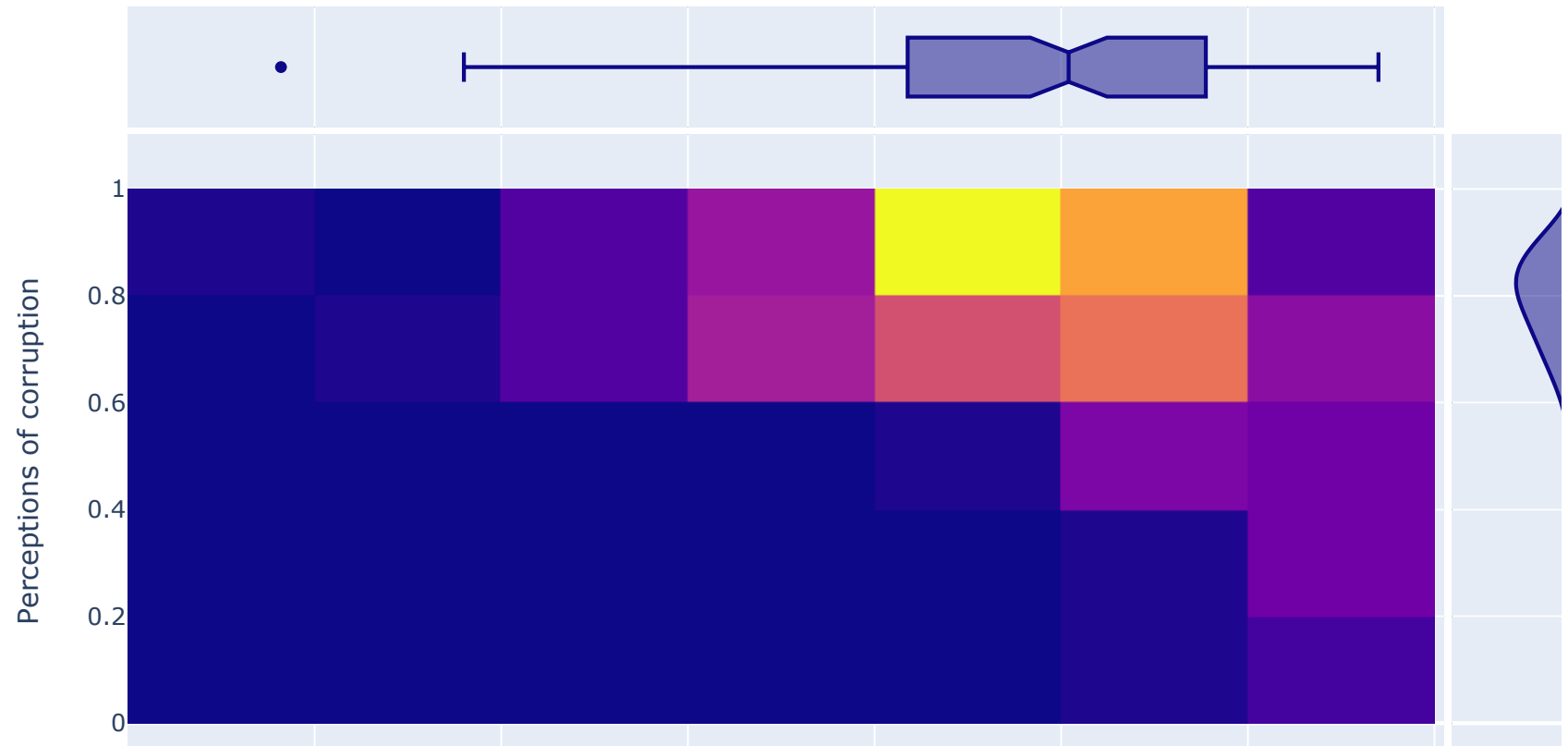
In [33]: *#Strong corr between GDP and well-being (which are defined by life expectancy, social support, and generosity)
#Generosity looks to be less strongly corr with well-being*

```
In [34]: fig = px.pie(df, values='Logged GDP per capita', names='Regional indicator', title='% of Logged GDP of regions from data')
fig.show()
```

% of Logged GDP of regions from data

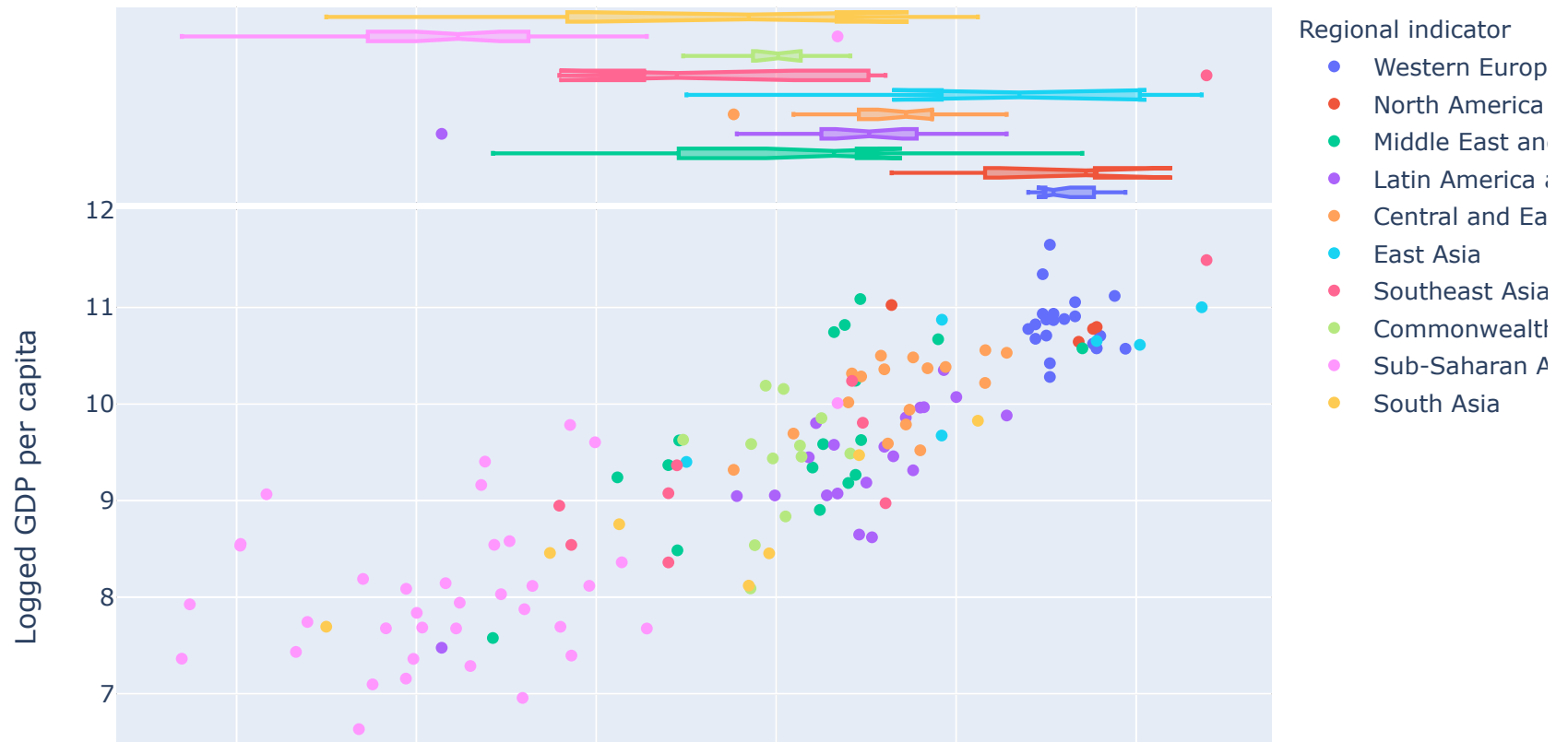


```
In [35]: fig = px.density_heatmap(df, x="Freedom to make life choices", y='Perceptions of corruption', marginal_x=True, marginal_y=True)  
fig.show()
```



```
In [36]: #looking at how freedom and perception of freedom interact with eah other.  
#Medians: freedom = 0.804, perception of corruption = 0.781
```

```
In [37]: fig = px.scatter(df, x="Healthy life expectancy", y="Logged GDP per capita", color="Regional indicator",
                    marginal_x="box")
fig.show()
```



In [38]: *#corr b/w life expectancy and GDP, you can also see easily by region*

```
In [39]: fig = px.violin(df, y="Logged GDP per capita", x="Generosity", color='Regional indicator', box=
           points='all', hover_data=df.columns # can be 'outliers', or False
           )
fig.show()
```



In [41]: *#Are the well off (GDP, social support, freedom, life expectancy) more generous? The conclusion*

In [42]: *#Also I conclude Singapor is the happiest country in 2021.*