Assignment 5 - Data Science Revision

January 10, 2025

Stefan Dimitrov Velev, 0MI3400521, Big Data Technologies

Faculty of Mathematics and Informatics, Sofia University

Task 1: In this question you will work with data sets from Our World In Data and Python to produce thoughtful analyses and interesting visualisations

1. Import required Python packages

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import geopandas as gpd
import requests

from sklearn.cluster import KMeans
from sklearn.cluster import DBSCAN
```

2. Read the CSV files 2.1 Self-reported life satisfaction

Data sources: World Happiness Report (2012-2024); Wellbeing Research Centre (2024); Population based on various sources (2023)

https://ourworldindata.org/grapher/happiness-cantril-ladder?time=latest

[2]:
```
df_life_satisfaction = pd.read_csv('./data/happiness-cantril-ladder.csv')
```

[3]:
```
df_life_satisfaction.head()
```

[3]:

	Entity	Code	Year	Cantril ladder score
0	Afghanistan	AFG	2011	4.25835
1	Afghanistan	AFG	2014	3.57500
2	Afghanistan	AFG	2015	3.36000
3	Afghanistan	AFG	2016	3.79400
4	Afghanistan	AFG	2017	3.63150

[4]:
```
print("The number of rows in the life satisfaction data frame is:",␣
↪len(df_life_satisfaction))
```

```
The number of rows in the life satisfaction data frame is: 1787
```

2.2 Share in extreme poverty vs. life expectancy

Data sources: UN, World Population Prospects (2024); World Bank Poverty and Inequality Platform (2024); HYDE (2023); Gapminder - Population v7 (2022); Gapminder - Systema Globalis (2022)

https://ourworldindata.org/grapher/extreme-poverty-headcount-ratio-vs-life-expectancy-at-birth

[5]:
```
df_extreme_poverty_life_expectancy = pd.read_csv('./data/
↪extreme-poverty-headcount-ratio-vs-life-expectancy-at-birth.csv')
```

[6]:
```
df_extreme_poverty_life_expectancy.head()
```

[6]:

	Entity	Code	Year	Life expectancy - Sex: all - Age: 0 - Variant: estimates	$2.15 a day - Share of population in poverty	990305-annotations	Population (historical)	World regions according to OWID
0	Afghanistan	AFG	1950	28.156	NaN	NaN	7776182.0	NaN
1	Afghanistan	AFG	1951	28.584	NaN	NaN	7879343.0	NaN
2	Afghanistan	AFG	1952	29.014	NaN	NaN	7987783.0	NaN
3	Afghanistan	AFG	1953	29.452	NaN	NaN	8096703.0	NaN
4	Afghanistan	AFG	1954	29.698	NaN	NaN	8207953.0	NaN

[7]:
```
print("The number of rows in the extreme poverty vs. life expectancy data frame␣
↪is:", len(df_extreme_poverty_life_expectancy))
```

```
The number of rows in the extreme poverty vs. life expectancy data frame is:
60100
```

2.3 Political corruption index

Data source: V-Dem (2024)

https://ourworldindata.org/grapher/political-corruption-index

[8]:
```
df_political_corruption = pd.read_csv('./data/political-corruption-index.csv')
```

[9]:
```
df_political_corruption.head()
```

[9]:

	Entity	Code	Year	Political corruption index (best estimate, aggregate: average)
0	Afghanistan	AFG	1789	0.438
1	Afghanistan	AFG	1790	0.438
2	Afghanistan	AFG	1791	0.438
3	Afghanistan	AFG	1792	0.438
4	Afghanistan	AFG	1793	0.438

[10]:
```
print("The number of rows in the political corruption data frame is:",␣
↪len(df_political_corruption))
```

```
The number of rows in the political corruption data frame is: 33090
```

3. Data Cleaning 3.1 Self-reported life satisfaction

[11]:
```
# Remove the unnecessary columns in the data frame
df_life_satisfaction = df_life_satisfaction[['Entity', 'Year', 'Cantril ladder␣
↪score']]
```

[12]:
```
# Rename the applicable columns
df_life_satisfaction = df_life_satisfaction.rename(columns={'Entity':␣
↪'Country'})
```

[13]:
```
# Leaving only rows for year 2021
df_life_satisfaction = df_life_satisfaction[df_life_satisfaction['Year'] ==␣
↪2021]
```

[14]:
```
# Remove rows with missing values
df_life_satisfaction = df_life_satisfaction.dropna()
```

[15]:
```
df_life_satisfaction.head(20)
```

[15]:

	Country	Year	Cantril ladder score
8	Afghanistan	2021	2.403800
19	Africa	2021	4.517288
30	Albania	2021	5.198800
41	Algeria	2021	5.122300
57	Argentina	2021	5.967000

```
68                  Armenia  2021              5.398600
79                     Asia  2021              4.892916
90                Australia  2021              7.162100
101                 Austria  2021              7.163000
112              Azerbaijan  2021              5.173400
122                 Bahrain  2021              6.646900
133              Bangladesh  2021              5.155500
144                 Belarus  2021              5.821500
153                 Belgium  2021              6.805000
168                   Benin  2021              4.623200
184                 Bolivia  2021              5.600300
195  Bosnia and Herzegovina  2021              5.768000
206                Botswana  2021              3.471100
217                  Brazil  2021              6.292800
228                Bulgaria  2021              5.370900
```

[16]:
```
# Remove not-country-specific entries
df_life_satisfaction = df_life_satisfaction[df_life_satisfaction['Country'] !=␣
 ↪'High-income countries']
df_life_satisfaction = df_life_satisfaction[df_life_satisfaction['Country'] !=␣
 ↪'Low-income countries']
df_life_satisfaction = df_life_satisfaction[df_life_satisfaction['Country'] !=␣
 ↪'Lower-middle-income countries']
df_life_satisfaction = df_life_satisfaction[df_life_satisfaction['Country'] !=␣
 ↪'Upper-middle-income countries']
df_life_satisfaction = df_life_satisfaction[df_life_satisfaction['Country'] !=␣
 ↪'World']
```

[17]:
```
df_life_satisfaction.head(20)
```

[17]:
```
                  Country  Year  Cantril ladder score
8             Afghanistan  2021              2.403800
19                 Africa  2021              4.517288
30                Albania  2021              5.198800
41                Algeria  2021              5.122300
57              Argentina  2021              5.967000
68                Armenia  2021              5.398600
79                   Asia  2021              4.892916
90              Australia  2021              7.162100
101               Austria  2021              7.163000
112            Azerbaijan  2021              5.173400
122               Bahrain  2021              6.646900
133            Bangladesh  2021              5.155500
144               Belarus  2021              5.821500
153               Belgium  2021              6.805000
168                 Benin  2021              4.623200
184               Bolivia  2021              5.600300
```

```
195  Bosnia and Herzegovina  2021              5.768000
206                Botswana  2021              3.471100
217                  Brazil  2021              6.292800
228                Bulgaria  2021              5.370900
```

3.2 Share in extreme poverty vs. life expectancy

[18]:
```
# Remove the unnecessary columns in the data frame
df_extreme_poverty_life_expectancy =␣
↪df_extreme_poverty_life_expectancy[['Entity', 'Year', 'Life expectancy - Sex:
↪ all - Age: 0 - Variant: estimates', '$2.15 a day - Share of population in␣
↪poverty', 'Population (historical)']]
```

[19]:
```
# Rename the applicable columns
df_extreme_poverty_life_expectancy = df_extreme_poverty_life_expectancy.
↪rename(columns={'Entity': 'Country', 'Life expectancy - Sex: all - Age: 0 -␣
↪Variant: estimates': 'Life expectancy', '$2.15 a day - Share of population␣
↪in poverty': 'Share in extreme poverty', 'Population (historical)':␣
↪'Population'})
```

[20]:
```
# Leaving only rows for year 2021
df_extreme_poverty_life_expectancy =␣
↪df_extreme_poverty_life_expectancy[df_extreme_poverty_life_expectancy['Year']␣
↪== 2021]
```

[21]:
```
# Remove rows with missing values
df_extreme_poverty_life_expectancy = df_extreme_poverty_life_expectancy.dropna()
```

[22]:
```
df_extreme_poverty_life_expectancy.head(20)
```

[22]:
```
                       Country  Year  Life expectancy  \
2588                   Armenia  2021           72.552
3683                   Austria  2021           81.820
5502                   Belgium  2021           81.659
6024                     Benin  2021           59.610
6636                   Bolivia  2021           61.427
7495                    Brazil  2021           73.038
8101                  Bulgaria  2021           71.414
8362              Burkina Faso  2021           60.046
9145                  Cameroon  2021           61.146
9982   Central African Republic  2021           40.279
10765                    China  2021           78.117
11072                 Colombia  2021           72.698
11927               Costa Rica  2021           78.050
12188            Cote d'Ivoire  2021           60.290
12449                  Croatia  2021           77.141
13045                   Cyprus  2021           80.573
13306                  Czechia  2021           77.234
```

```
13914                  Denmark  2021           81.436
14692       Dominican Republic  2021           71.757
15295                  Ecuador  2021           72.746

       Share in extreme poverty    Population
2588                   0.523521  2.870339e+06
3683                   0.485822  8.967053e+06
5502                   0.029965  1.157084e+07
6024                  12.723279  1.341342e+07
6636                   1.964501  1.193737e+07
7495                   5.834562  2.095503e+08
8101                   0.699141  6.877222e+06
8362                  25.277073  2.199525e+07
9145                  22.992220  2.691576e+07
9982                  65.666510  5.112102e+06
10765                  0.000000  1.426437e+09
11072                  7.334666  5.118817e+07
11927                  1.242194  5.059992e+06
12188                  9.733192  2.963974e+07
12449                  0.308004  3.924607e+06
13045                  0.005303  1.317312e+06
13306                  0.059851  1.053068e+07
13914                  0.153680  5.856773e+06
14692                  0.853729  1.112348e+07
15295                  3.583180  1.768245e+07
```

[23]:
```
# Remove not-country-specific entries
df_extreme_poverty_life_expectancy =␣
 ↪df_extreme_poverty_life_expectancy[df_extreme_poverty_life_expectancy['Country']␣
 ↪!= 'World']
```

[24]:
```
df_extreme_poverty_life_expectancy.head(20)
```

[24]:
```
                       Country  Year  Life expectancy  \
2588                   Armenia  2021           72.552
3683                   Austria  2021           81.820
5502                   Belgium  2021           81.659
6024                     Benin  2021           59.610
6636                   Bolivia  2021           61.427
7495                    Brazil  2021           73.038
8101                  Bulgaria  2021           71.414
8362              Burkina Faso  2021           60.046
9145                  Cameroon  2021           61.146
9982   Central African Republic  2021           40.279
10765                    China  2021           78.117
11072                 Colombia  2021           72.698
11927               Costa Rica  2021           78.050
```

```
12188             Cote d'Ivoire  2021           60.290
12449                   Croatia  2021           77.141
13045                    Cyprus  2021           80.573
13306                   Czechia  2021           77.234
13914                   Denmark  2021           81.436
14692        Dominican Republic  2021           71.757
15295                   Ecuador  2021           72.746

       Share in extreme poverty    Population
2588                   0.523521  2.870339e+06
3683                   0.485822  8.967053e+06
5502                   0.029965  1.157084e+07
6024                  12.723279  1.341342e+07
6636                   1.964501  1.193737e+07
7495                   5.834562  2.095503e+08
8101                   0.699141  6.877222e+06
8362                  25.277073  2.199525e+07
9145                  22.992220  2.691576e+07
9982                  65.666510  5.112102e+06
10765                  0.000000  1.426437e+09
11072                  7.334666  5.118817e+07
11927                  1.242194  5.059992e+06
12188                  9.733192  2.963974e+07
12449                  0.308004  3.924607e+06
13045                  0.005303  1.317312e+06
13306                  0.059851  1.053068e+07
13914                  0.153680  5.856773e+06
14692                  0.853729  1.112348e+07
15295                  3.583180  1.768245e+07
```

[25]:
```
df_extreme_poverty_life_expectancy.describe()
```

[25]:
```
          Year  Life expectancy  Share in extreme poverty    Population
count     70.0        70.000000                 70.000000  7.000000e+01
mean    2021.0        73.067557                  5.409042  7.159597e+07
std        0.0         8.103193                 11.956552  2.401526e+08
min     2021.0        40.279000                  0.000000  1.055040e+05
25%     2021.0        69.136250                  0.102991  5.073020e+06
50%     2021.0        73.770500                  0.543886  1.085152e+07
75%     2021.0        80.506750                  3.401814  3.400077e+07
max     2021.0        83.852000                 65.666510  1.426437e+09
```

[26]:
```
df_extreme_poverty_life_expectancy[df_extreme_poverty_life_expectancy['Country']␣
 ↪== 'Central African Republic']
```

[26]:
```
                      Country  Year  Life expectancy  \
9982  Central African Republic  2021           40.279
```

```
      Share in extreme poverty  Population
9982                  65.66651   5112102.0
```

[27]:
```
# Correct the Central African Republic life expectancy according the World Health Organization Data for 2021
# Source: https://data.who.int/countries/140
df_extreme_poverty_life_expectancy.loc[df_extreme_poverty_life_expectancy['Country'] == 'Central African Republic', 'Life expectancy'] = 52.31
```

3.3 Political corruption index

[28]:
```
# Remove the unnecessary columns in the data frame
df_political_corruption = df_political_corruption[['Entity', 'Year', 'Political corruption index (best estimate, aggregate: average)']]
```

[29]:
```
# Rename the applicable columns
df_political_corruption = df_political_corruption.rename(columns={'Entity': 'Country', 'Political corruption index (best estimate, aggregate: average)': 'Political corruption index'})
```

[30]:
```
# Leaving only rows for year 2021
df_political_corruption = df_political_corruption[df_political_corruption['Year'] == 2021]
```

[31]:
```
# Remove rows with missing values
df_political_corruption = df_political_corruption.dropna()
```

[32]:
```
df_political_corruption.head(20)
```

[32]:

	Country	Year	Political corruption index
232	Afghanistan	2021	0.397000
467	Africa	2021	0.624089
579	Albania	2021	0.609000
703	Algeria	2021	0.693000
827	Angola	2021	0.510000
1062	Argentina	2021	0.471000
1168	Armenia	2021	0.315000
1403	Asia	2021	0.555878
1638	Australia	2021	0.031000
1866	Austria	2021	0.123000
2077	Azerbaijan	2021	0.914000
2178	Bahrain	2021	0.706000
2412	Bangladesh	2021	0.907000
2536	Barbados	2021	0.067000
2847	Belarus	2021	0.374000
3082	Belgium	2021	0.031000

3206	Benin	2021	0.192000
3330	Bhutan	2021	0.117000
3529	Bolivia	2021	0.574000
3724	Bosnia and Herzegovina	2021	0.754000

[33]:
```
# Remove not-country-specific entries
df_political_corruption = df_political_corruption[df_political_corruption['Country'] != 'World']
```

[34]:
```
df_political_corruption.head(20)
```

[34]:

	Country	Year	Political corruption index
232	Afghanistan	2021	0.397000
467	Africa	2021	0.624089
579	Albania	2021	0.609000
703	Algeria	2021	0.693000
827	Angola	2021	0.510000
1062	Argentina	2021	0.471000
1168	Armenia	2021	0.315000
1403	Asia	2021	0.555878
1638	Australia	2021	0.031000
1866	Austria	2021	0.123000
2077	Azerbaijan	2021	0.914000
2178	Bahrain	2021	0.706000
2412	Bangladesh	2021	0.907000
2536	Barbados	2021	0.067000
2847	Belarus	2021	0.374000
3082	Belgium	2021	0.031000
3206	Benin	2021	0.192000
3330	Bhutan	2021	0.117000
3529	Bolivia	2021	0.574000
3724	Bosnia and Herzegovina	2021	0.754000

4. Data Segregation

[35]:
```
continents = ['Africa', 'Asia', 'Australia', 'Europe', 'North America', 'South America']
```

4.1 Self-reported life satisfaction

[36]:
```
# Extract data only for continents
df_life_satisfaction_continents = df_life_satisfaction[df_life_satisfaction['Country'].isin(continents)]
```

[37]:
```
df_life_satisfaction_continents
```

[37]:

	Country	Year	Cantril ladder score
19	Africa	2021	4.517288
79	Asia	2021	4.892916

```
90         Australia  2021              7.162100
511           Europe  2021              6.338868
1190  North America  2021              6.692469
1465  South America  2021              5.985671
```

[38]:
```
# Remove continent entries in the original data frame (Australia remains as it is a country as well)
df_life_satisfaction = df_life_satisfaction[(~df_life_satisfaction['Country'].isin(continents)) | (df_life_satisfaction['Country'] == 'Australia')]
```

[39]:
```
df_life_satisfaction.head(20)
```

[39]:

	Country	Year	Cantril ladder score
8	Afghanistan	2021	2.4038
30	Albania	2021	5.1988
41	Algeria	2021	5.1223
57	Argentina	2021	5.9670
68	Armenia	2021	5.3986
90	Australia	2021	7.1621
101	Austria	2021	7.1630
112	Azerbaijan	2021	5.1734
122	Bahrain	2021	6.6469
133	Bangladesh	2021	5.1555
144	Belarus	2021	5.8215
153	Belgium	2021	6.8050
168	Benin	2021	4.6232
184	Bolivia	2021	5.6003
195	Bosnia and Herzegovina	2021	5.7680
206	Botswana	2021	3.4711
217	Brazil	2021	6.2928
228	Bulgaria	2021	5.3709
239	Burkina Faso	2021	4.6705
258	Cambodia	2021	4.6403

[40]:
```
df_life_satisfaction.count()
```

[40]:
```
Country                147
Year                   147
Cantril ladder score   147
dtype: int64
```

4.2 Political corruption index

[41]:
```
# Extract data only for continents
df_political_corruption_continents = df_political_corruption[df_political_corruption['Country'].isin(continents)]
```

[42]:
```
df_political_corruption_continents
```

[42]:
	Country	Year	Political corruption index
467	Africa	2021	0.624089
1403	Asia	2021	0.555878
1638	Australia	2021	0.031000
9709	Europe	2021	0.235659
20953	North America	2021	0.458667
26875	South America	2021	0.472500

[43]:
```
# Remove continent entries in the original data frame (Australia remains as it␣
 ↪is a country as well)
df_political_corruption =␣
 ↪df_political_corruption[(~df_political_corruption['Country'].
 ↪isin(continents)) | (df_political_corruption['Country'] == 'Australia')]
```

[44]:
```
df_political_corruption.head(20)
```

[44]:
	Country	Year	Political corruption index
232	Afghanistan	2021	0.397
579	Albania	2021	0.609
703	Algeria	2021	0.693
827	Angola	2021	0.510
1062	Argentina	2021	0.471
1168	Armenia	2021	0.315
1638	Australia	2021	0.031
1866	Austria	2021	0.123
2077	Azerbaijan	2021	0.914
2178	Bahrain	2021	0.706
2412	Bangladesh	2021	0.907
2536	Barbados	2021	0.067
2847	Belarus	2021	0.374
3082	Belgium	2021	0.031
3206	Benin	2021	0.192
3330	Bhutan	2021	0.117
3529	Bolivia	2021	0.574
3724	Bosnia and Herzegovina	2021	0.754
3848	Botswana	2021	0.161
4083	Brazil	2021	0.543

[45]:
```
df_political_corruption.count()
```

[45]:
```
Country                     180
Year                        180
Political corruption index  180
dtype: int64
```

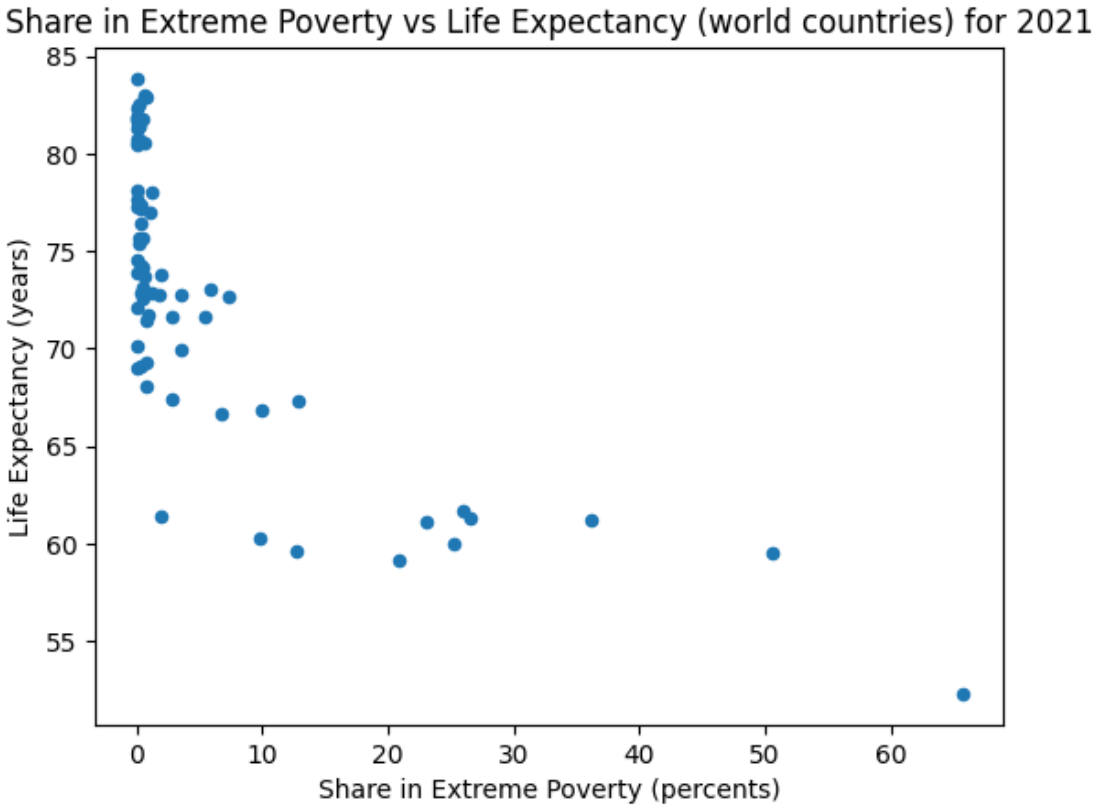
5. Draw a scatter plot of Share of Population in Extreme Poverty vs Life Expectancy for 2021

[46]:
```
df_extreme_poverty_life_expectancy.describe()
```

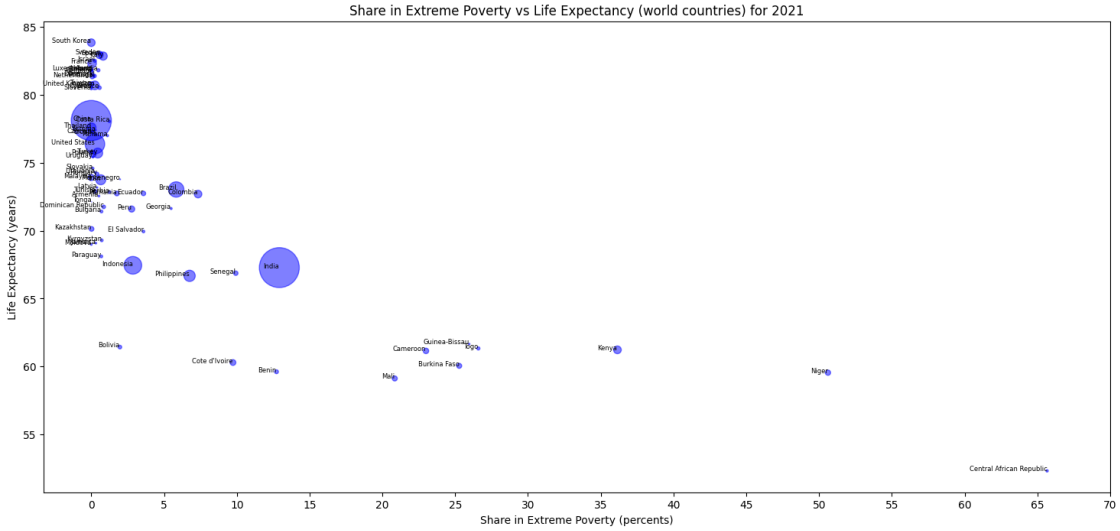
[46]:

	Year	Life expectancy	Share in extreme poverty	Population
count	70.0	70.000000	70.000000	7.000000e+01
mean	2021.0	73.239429	5.409042	7.159597e+07
std	0.0	7.503022	11.956552	2.401526e+08
min	2021.0	52.310000	0.000000	1.055040e+05
25%	2021.0	69.136250	0.102991	5.073020e+06
50%	2021.0	73.770500	0.543886	1.085152e+07
75%	2021.0	80.506750	3.401814	3.400077e+07
max	2021.0	83.852000	65.666510	1.426437e+09

[47]:
```
df_extreme_poverty_life_expectancy.plot.scatter(x='Share in extreme poverty',␣
↪y='Life expectancy')
plt.xlabel('Share in Extreme Poverty (percents)')
plt.ylabel('Life Expectancy (years)')
plt.title('Share in Extreme Poverty vs Life Expectancy (world countries) for␣
↪2021')
plt.show()
```

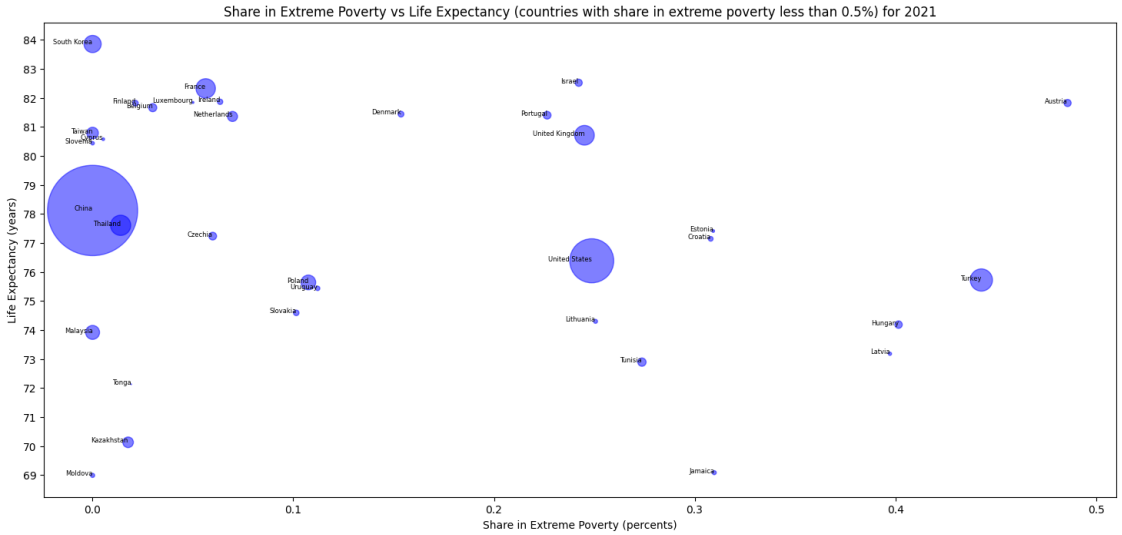


[48]:
```
plt.figure(figsize=(18, 8))
plt.scatter(df_extreme_poverty_life_expectancy['Share in extreme poverty'],␣
 ↪df_extreme_poverty_life_expectancy['Life expectancy'], color='blue', s =␣
 ↪df_extreme_poverty_life_expectancy['Population']/1000000, alpha=0.5)

for i, country in enumerate(df_extreme_poverty_life_expectancy['Country']):
    plt.text(df_extreme_poverty_life_expectancy['Share in extreme poverty'].
 ↪iloc[i], df_extreme_poverty_life_expectancy['Life expectancy'].iloc[i],␣
 ↪df_extreme_poverty_life_expectancy['Country'].iloc[i], fontsize=6,␣
 ↪ha='right')

plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70])
plt.xlabel('Share in Extreme Poverty (percents)')
plt.ylabel('Life Expectancy (years)')
plt.title('Share in Extreme Poverty vs Life Expectancy (world countries) for␣
 ↪2021')
plt.show()
```



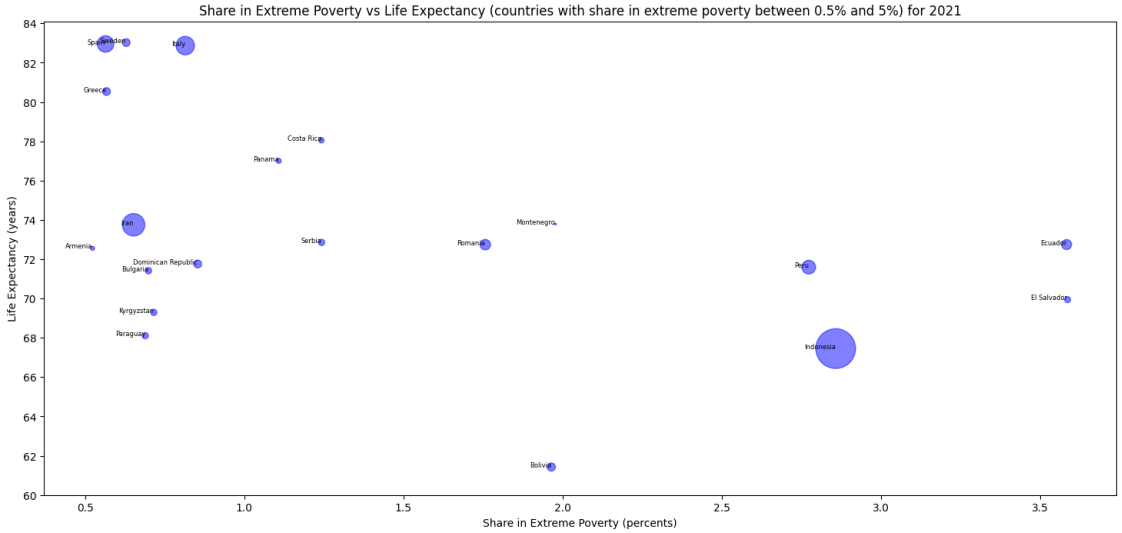
[49]:
```
plt.figure(figsize=(18, 8))
df_low_extreme_poverty =␣
 ↪df_extreme_poverty_life_expectancy[df_extreme_poverty_life_expectancy['Share␣
 ↪in extreme poverty'] <= 0.5]
plt.scatter(df_low_extreme_poverty['Share in extreme poverty'],␣
 ↪df_low_extreme_poverty['Life expectancy'], color='blue', s =␣
 ↪df_low_extreme_poverty['Population']/200000, alpha=0.5)

for i, country in enumerate(df_low_extreme_poverty['Country']):
```

```
    plt.text(df_low_extreme_poverty['Share in extreme poverty'].iloc[i],␣
  ↪df_low_extreme_poverty['Life expectancy'].iloc[i],␣
  ↪df_low_extreme_poverty['Country'].iloc[i], fontsize=6, ha='right')

plt.yticks(range(69, 85))

plt.xlabel('Share in Extreme Poverty (percents)')
plt.ylabel('Life Expectancy (years)')
plt.title('Share in Extreme Poverty vs Life Expectancy (countries with share in␣
  ↪extreme poverty less than 0.5%) for 2021')
plt.show()
```

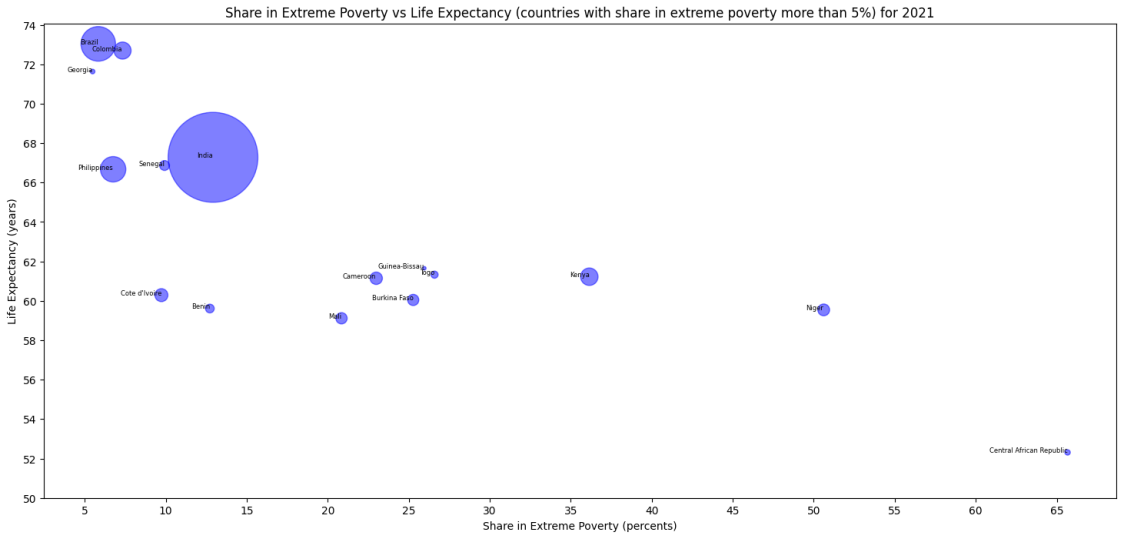


[50]:
```
plt.figure(figsize=(18, 8))
df_middle_extreme_poverty =␣
  ↪df_extreme_poverty_life_expectancy[(df_extreme_poverty_life_expectancy['Share␣
  ↪in extreme poverty'] >= 0.5) & (df_extreme_poverty_life_expectancy['Share in␣
  ↪extreme poverty'] <= 5)]
plt.scatter(df_middle_extreme_poverty['Share in extreme poverty'],␣
  ↪df_middle_extreme_poverty['Life expectancy'], color='blue', s =␣
  ↪df_middle_extreme_poverty['Population']/200000, alpha=0.5)

for i, country in enumerate(df_middle_extreme_poverty['Country']):
    plt.text(df_middle_extreme_poverty['Share in extreme poverty'].iloc[i],␣
  ↪df_middle_extreme_poverty['Life expectancy'].iloc[i],␣
  ↪df_middle_extreme_poverty['Country'].iloc[i], fontsize=6, ha='right')

plt.yticks(range(60, 85, 2))
```

```
plt.xlabel('Share in Extreme Poverty (percents)')
plt.ylabel('Life Expectancy (years)')
plt.title('Share in Extreme Poverty vs Life Expectancy (countries with share in␣
 ↪extreme poverty between 0.5% and 5%) for 2021')
plt.show()
```



[51]:
```
plt.figure(figsize=(18, 8))
df_high_extreme_poverty =␣
 ↪df_extreme_poverty_life_expectancy[df_extreme_poverty_life_expectancy['Share␣
 ↪in extreme poverty'] >= 5]
plt.scatter(df_high_extreme_poverty['Share in extreme poverty'],␣
 ↪df_high_extreme_poverty['Life expectancy'], color='blue', s =␣
 ↪df_high_extreme_poverty['Population']/200000, alpha=0.5)

for i, country in enumerate(df_high_extreme_poverty['Country']):
    plt.text(df_high_extreme_poverty['Share in extreme poverty'].iloc[i],␣
 ↪df_high_extreme_poverty['Life expectancy'].iloc[i],␣
 ↪df_high_extreme_poverty['Country'].iloc[i], fontsize=6, ha='right')

plt.xticks(range(5, 70, 5))
plt.yticks(range(50, 75, 2))

plt.xlabel('Share in Extreme Poverty (percents)')
plt.ylabel('Life Expectancy (years)')
plt.title('Share in Extreme Poverty vs Life Expectancy (countries with share in␣
 ↪extreme poverty more than 5%) for 2021')
plt.show()
```

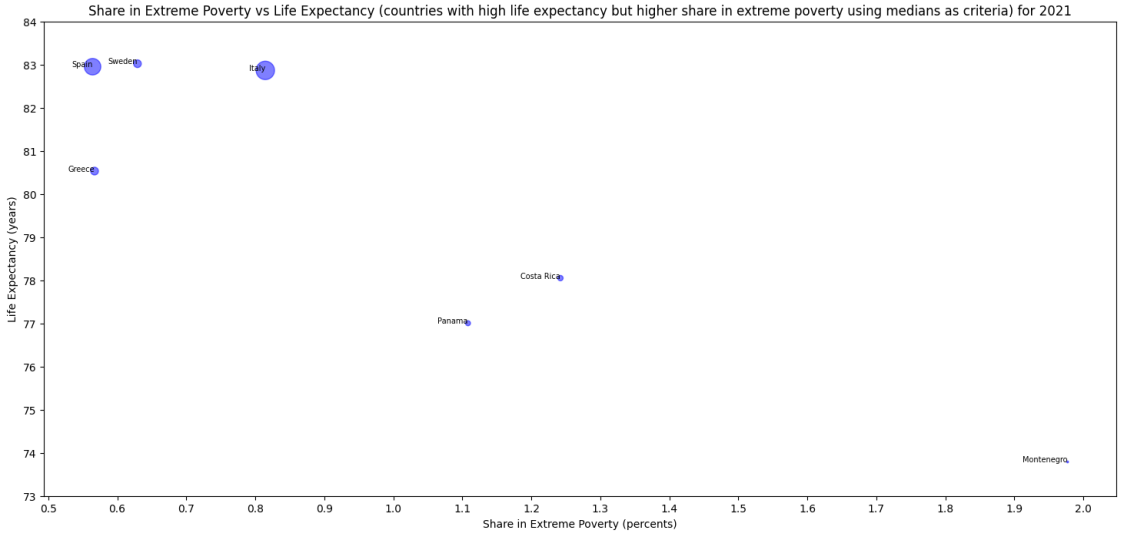


5.1 Check which countries have high life expectancy but have higher share in extreme poverty for 2021?

[52]:
```
# List countries with high life expectancy but higher share in extreme poverty using medians
median_life_expectancy = np.median(df_extreme_poverty_life_expectancy['Life expectancy'])
print('Median life expectancy is', median_life_expectancy)
median_extreme_poverty = np.median(df_extreme_poverty_life_expectancy['Share in extreme poverty'])
print('Median share in extreme poverty is', median_extreme_poverty)
df_higher_life_expectancy_higher_extreme_poverty_medians = df_extreme_poverty_life_expectancy[(df_extreme_poverty_life_expectancy['Life expectancy'] > median_life_expectancy) & (df_extreme_poverty_life_expectancy['Share in extreme poverty'] > median_extreme_poverty)]
print('The number of countries with high life expectancy but higher share in extreme poverty using medians is',len(df_higher_life_expectancy_higher_extreme_poverty_medians))
print('The list of countries with high life expectancy but higher share in extreme poverty using medians is:', ', '.join(df_higher_life_expectancy_higher_extreme_poverty_medians['Country']))
```

```
Median life expectancy is 73.7705
Median share in extreme poverty is 0.5438855499999999
The number of countries with high life expectancy but higher share in extreme poverty using medians is 7
The list of countries with high life expectancy but higher share in extreme poverty using medians is: Costa Rica, Greece, Italy, Montenegro, Panama, Spain, Sweden
```

[53]:
```
plt.figure(figsize=(18, 8))
plt.scatter(df_higher_life_expectancy_higher_extreme_poverty_medians['Share in␣
 ↪extreme poverty'],␣
 ↪df_higher_life_expectancy_higher_extreme_poverty_medians['Life expectancy'],␣
 ↪color='blue', s =␣
 ↪df_higher_life_expectancy_higher_extreme_poverty_medians['Population']/
 ↪200000, alpha=0.5)

for i, country in␣
 ↪enumerate(df_higher_life_expectancy_higher_extreme_poverty_medians['Country']):
 ↪
    plt.text(df_higher_life_expectancy_higher_extreme_poverty_medians['Share in␣
 ↪extreme poverty'].iloc[i],␣
 ↪df_higher_life_expectancy_higher_extreme_poverty_medians['Life expectancy'].
 ↪iloc[i], df_higher_life_expectancy_higher_extreme_poverty_medians['Country'].
 ↪iloc[i], fontsize=7, ha='right')

plt.xticks([0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.
 ↪8, 1.9, 2])
plt.yticks(range(73, 85))

plt.xlabel('Share in Extreme Poverty (percents)')
plt.ylabel('Life Expectancy (years)')
plt.title('Share in Extreme Poverty vs Life Expectancy (countries with high␣
 ↪life expectancy but higher share in extreme poverty using medians as␣
 ↪criteria) for 2021')
plt.show()
```



[54]:
```
# List countries with high life expectancy but higher share in extreme poverty␣
 ↪using means
mean_life_expectancy = np.mean(df_extreme_poverty_life_expectancy['Life␣
 ↪expectancy'])
print('Mean life expectancy is', mean_life_expectancy)
mean_extreme_poverty = np.mean(df_extreme_poverty_life_expectancy['Share in␣
 ↪extreme poverty'])
print('Mean share in extreme poverty is', mean_extreme_poverty)
df_higher_life_expectancy_higher_extreme_poverty_means =␣
 ↪df_extreme_poverty_life_expectancy[(df_extreme_poverty_life_expectancy['Life␣
 ↪expectancy'] > mean_life_expectancy) &␣
 ↪(df_extreme_poverty_life_expectancy['Share in extreme poverty'] >␣
 ↪mean_extreme_poverty)]
print('The number of countries with high life expectancy but higher share in␣
 ↪extreme poverty using means is',␣
 ↪len(df_higher_life_expectancy_higher_extreme_poverty_means))
```

```
Mean life expectancy is 73.23942857142856
Mean share in extreme poverty is 5.409041956905713
The number of countries with high life expectancy but higher share in extreme
poverty using means is 0
```

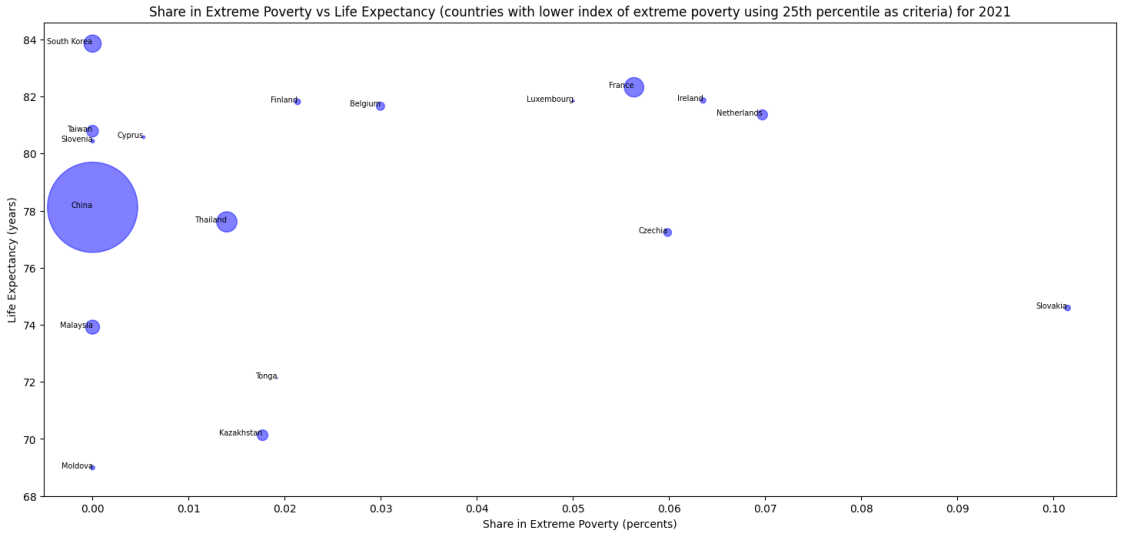
5.2 Find whether each country with lower share in extreme poverty have high life expectancy?

[55]:
```
upper_boundary_low_extreme_poverty = df_extreme_poverty_life_expectancy['Share␣
 ↪in extreme poverty'].quantile(0.25)
print('The upper boundary for low share in extreme poverty is',␣
 ↪upper_boundary_low_extreme_poverty)
df_lower_extreme_poverty =␣
 ↪df_extreme_poverty_life_expectancy[df_extreme_poverty_life_expectancy['Share␣
 ↪in extreme poverty'] <= upper_boundary_low_extreme_poverty]
print('The number of countries with lower index of extreme poverty using 25th␣
 ↪percentile is', len(df_lower_extreme_poverty))
print('The list of countries with lower index of extreme poverty using 25th␣
 ↪percentile is:', ', '.join(df_lower_extreme_poverty['Country']))
```

```
The upper boundary for low share in extreme poverty is 0.1029907725
The number of countries with lower index of extreme poverty using 25th
percentile is 18
The list of countries with lower index of extreme poverty using 25th percentile
is: Belgium, China, Cyprus, Czechia, Finland, France, Ireland, Kazakhstan,
Luxembourg, Malaysia, Moldova, Netherlands, Slovakia, Slovenia, South Korea,
Taiwan, Thailand, Tonga
```

[56]:
```
plt.figure(figsize=(18, 8))
```

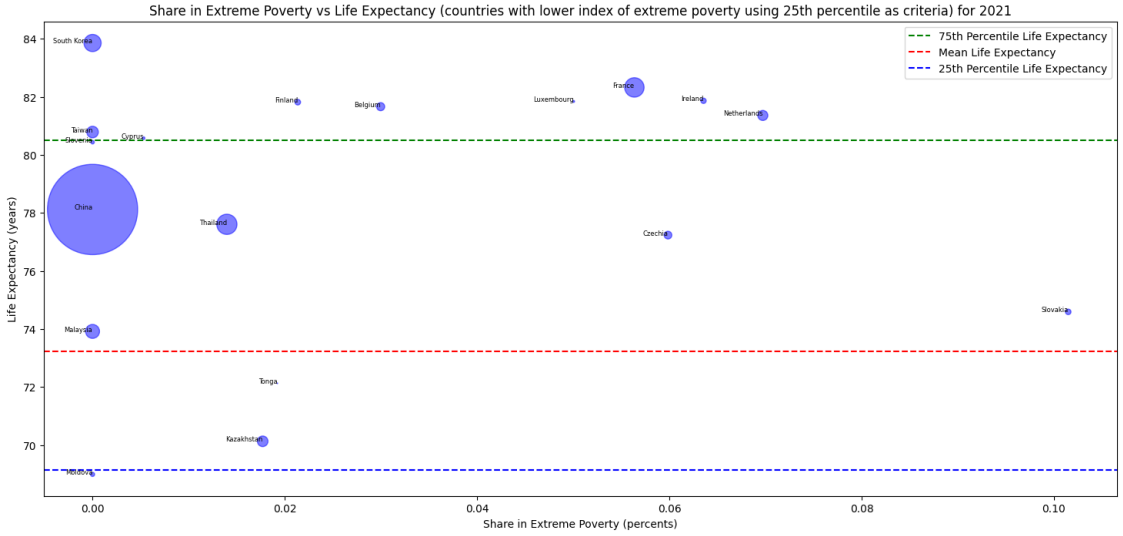
```
plt.scatter(df_lower_extreme_poverty['Share in extreme poverty'],␣
 ↪df_lower_extreme_poverty['Life expectancy'], color='blue', s =␣
 ↪df_lower_extreme_poverty['Population']/200000, alpha=0.5)

for i, country in enumerate(df_lower_extreme_poverty['Country']):
    plt.text(df_lower_extreme_poverty['Share in extreme poverty'].iloc[i],␣
 ↪df_lower_extreme_poverty['Life expectancy'].iloc[i],␣
 ↪df_lower_extreme_poverty['Country'].iloc[i], fontsize=7, ha='right')

plt.xticks([0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10])
plt.yticks(range(68, 86, 2))

plt.xlabel('Share in Extreme Poverty (percents)')
plt.ylabel('Life Expectancy (years)')
plt.title('Share in Extreme Poverty vs Life Expectancy (countries with lower␣
 ↪index of extreme poverty using 25th percentile as criteria) for 2021')
plt.show()
```



[57]:
```
lower_boundary_high_life_expectancy = np.
 ↪quantile(df_extreme_poverty_life_expectancy['Life expectancy'], 0.75)
print('75th Percentile life expectancy is', lower_boundary_high_life_expectancy)
print('Mean life expectancy is', mean_life_expectancy)
upper_boundary_low_life_expectancy = np.
 ↪quantile(df_extreme_poverty_life_expectancy['Life expectancy'], 0.25)
print('25th Percentile life expectancy is', upper_boundary_low_life_expectancy)
```

```
75th Percentile life expectancy is 80.50675000000001
Mean life expectancy is 73.23942857142856
25th Percentile life expectancy is 69.13624999999999
```

[58]:
```
plt.figure(figsize=(18, 8))
plt.scatter(df_lower_extreme_poverty['Share in extreme poverty'],␣
 ↪df_lower_extreme_poverty['Life expectancy'], color='blue', s =␣
 ↪df_lower_extreme_poverty['Population']/200000, alpha=0.5)

for i, country in enumerate(df_lower_extreme_poverty['Country']):
    plt.text(df_lower_extreme_poverty['Share in extreme poverty'].iloc[i],␣
 ↪df_lower_extreme_poverty['Life expectancy'].iloc[i],␣
 ↪df_lower_extreme_poverty['Country'].iloc[i], fontsize=6, ha='right')

plt.axhline(y=lower_boundary_high_life_expectancy, color='green',␣
 ↪linestyle='--', label='75th Percentile Life Expectancy')

plt.axhline(y=mean_life_expectancy, color='red', linestyle='--', label='Mean␣
 ↪Life Expectancy')

plt.axhline(y=upper_boundary_low_life_expectancy, color='blue', linestyle='--',␣
 ↪label='25th Percentile Life Expectancy')

plt.xlabel('Share in Extreme Poverty (percents)')
plt.ylabel('Life Expectancy (years)')
plt.title('Share in Extreme Poverty vs Life Expectancy (countries with lower␣
 ↪index of extreme poverty using 25th percentile as criteria) for 2021')
plt.legend()
plt.show()
```



[59]:
```
df_lower_extreme_poverty
```

[59]:

	Country	Year	Life expectancy	Share in extreme poverty	Population
5502	Belgium	2021	81.659	0.029965	1.157084e+07
10765	China	2021	78.117	0.000000	1.426437e+09
13045	Cyprus	2021	80.573	0.005303	1.317312e+06
13306	Czechia	2021	77.234	0.059851	1.053068e+07
18681	Finland	2021	81.814	0.021343	5.541068e+06
18942	France	2021	82.322	0.056357	6.608355e+07
25393	Ireland	2021	81.861	0.063545	5.028430e+06
27103	Kazakhstan	2021	70.131	0.017695	1.974361e+07
31476	Luxembourg	2021	81.839	0.050037	6.402760e+05
32345	Malaysia	2021	73.917	0.000000	3.428240e+07
34806	Moldova	2021	68.991	0.000000	3.023780e+06
37489	Netherlands	2021	81.356	0.069717	1.773056e+07
48218	Slovakia	2021	74.590	0.101481	5.442767e+06
48479	Slovenia	2021	80.434	0.000000	2.113495e+06
49944	South Korea	2021	83.852	0.000000	5.184840e+07
52397	Taiwan	2021	80.785	0.000000	2.355833e+07
53180	Thailand	2021	77.606	0.013972	7.172734e+07
53781	Tonga	2021	72.130	0.019216	1.055040e+05

6. Draw a scatter plot of Self-Reported Life Satisfaction (Cantril Ladder Score) vs Political Corruption Index for 2021

[60]:
```
df_life_satisfaction.describe()
```

[60]:

	Year	Cantril ladder score
count	147.0	147.000000
mean	2021.0	5.564560

```
std       0.0              1.091243
min    2021.0              2.403800
25%    2021.0              4.889200
50%    2021.0              5.578300
75%    2021.0              6.324950
max    2021.0              7.821000
```

[61]:
```
df_political_corruption.describe()
```

[61]:
```
          Year  Political corruption index
count   180.0                  180.000000
mean   2021.0                    0.482430
std       0.0                    0.296203
min    2021.0                    0.002000
25%    2021.0                    0.192750
50%    2021.0                    0.511500
75%    2021.0                    0.736750
max    2021.0                    0.967000
```

[62]:
```
# Merge the above two data frames using inner join on 'Country' column
df_life_satisfaction_political_corruption = pd.merge(df_life_satisfaction,␣
 ↪df_political_corruption, on='Country', how='inner')
```

[63]:
```
df_life_satisfaction_political_corruption.head(20)
```

[63]:
```
                   Country  Year_x  Cantril ladder score  Year_y  \
0              Afghanistan    2021                2.4038    2021
1                  Albania    2021                5.1988    2021
2                  Algeria    2021                5.1223    2021
3                Argentina    2021                5.9670    2021
4                  Armenia    2021                5.3986    2021
5                Australia    2021                7.1621    2021
6                  Austria    2021                7.1630    2021
7               Azerbaijan    2021                5.1734    2021
8                  Bahrain    2021                6.6469    2021
9               Bangladesh    2021                5.1555    2021
10                 Belarus    2021                5.8215    2021
11                 Belgium    2021                6.8050    2021
12                   Benin    2021                4.6232    2021
13                 Bolivia    2021                5.6003    2021
14  Bosnia and Herzegovina    2021                5.7680    2021
15                Botswana    2021                3.4711    2021
16                  Brazil    2021                6.2928    2021
17                Bulgaria    2021                5.3709    2021
18            Burkina Faso    2021                4.6705    2021
19                Cambodia    2021                4.6403    2021
```

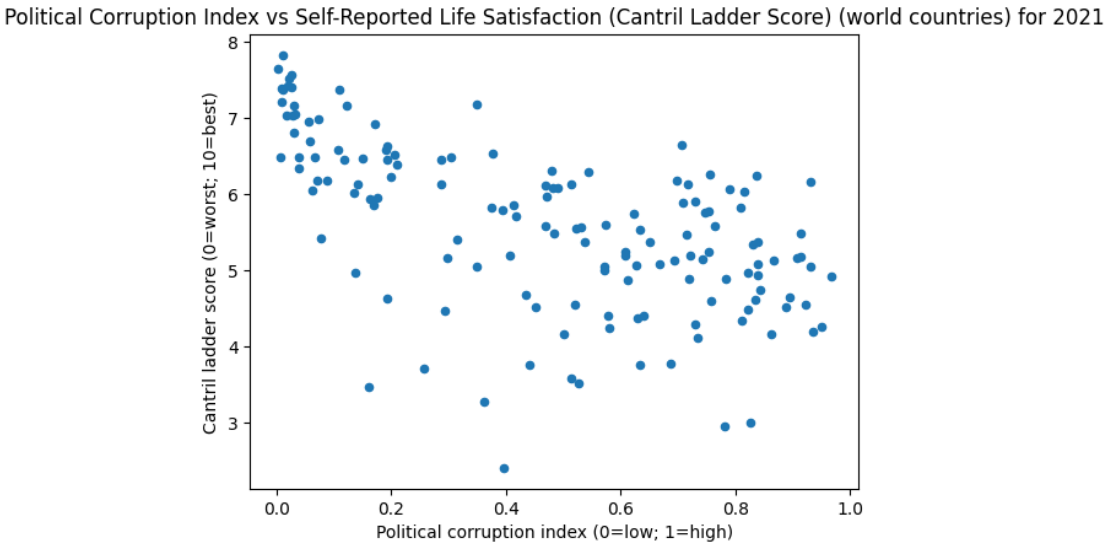
```
    Political corruption index
0                        0.397
1                        0.609
2                        0.693
3                        0.471
4                        0.315
5                        0.031
6                        0.123
7                        0.914
8                        0.706
9                        0.907
10                       0.374
11                       0.031
12                       0.192
13                       0.574
14                       0.754
15                       0.161
16                       0.543
17                       0.537
18                       0.435
19                       0.895
```

[64]:
```
df_life_satisfaction_political_corruption.describe()
```

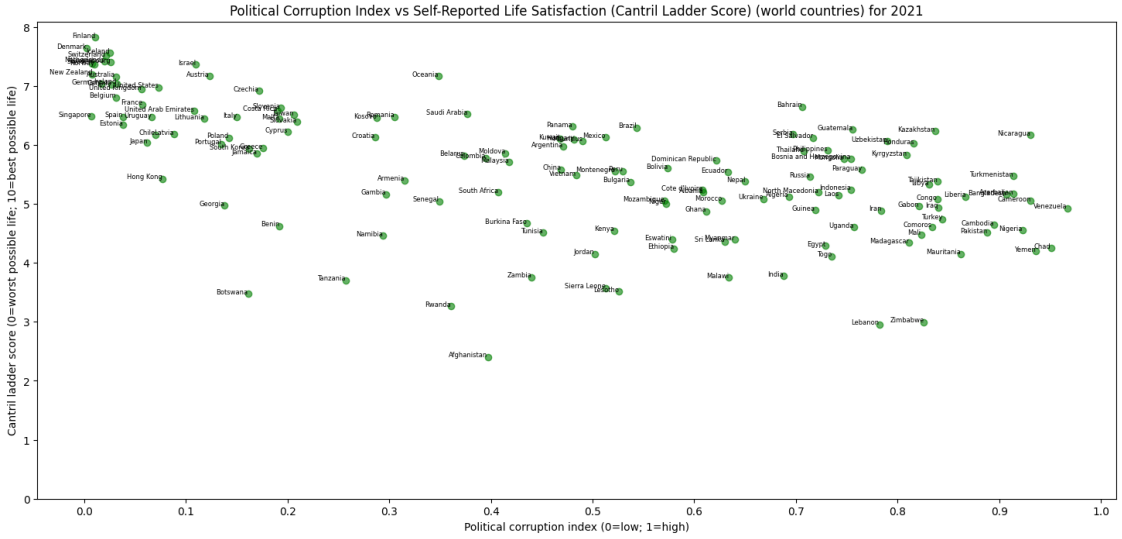
[64]:

	Year_x	Cantril ladder score	Year_y	Political corruption index
count	145.0	145.000000	145.0	145.000000
mean	2021.0	5.572698	2021.0	0.469389
std	0.0	1.095028	0.0	0.302031
min	2021.0	2.403800	2021.0	0.002000
25%	2021.0	4.890500	2021.0	0.172000
50%	2021.0	5.585300	2021.0	0.502000
75%	2021.0	6.340800	2021.0	0.735000
max	2021.0	7.821000	2021.0	0.967000

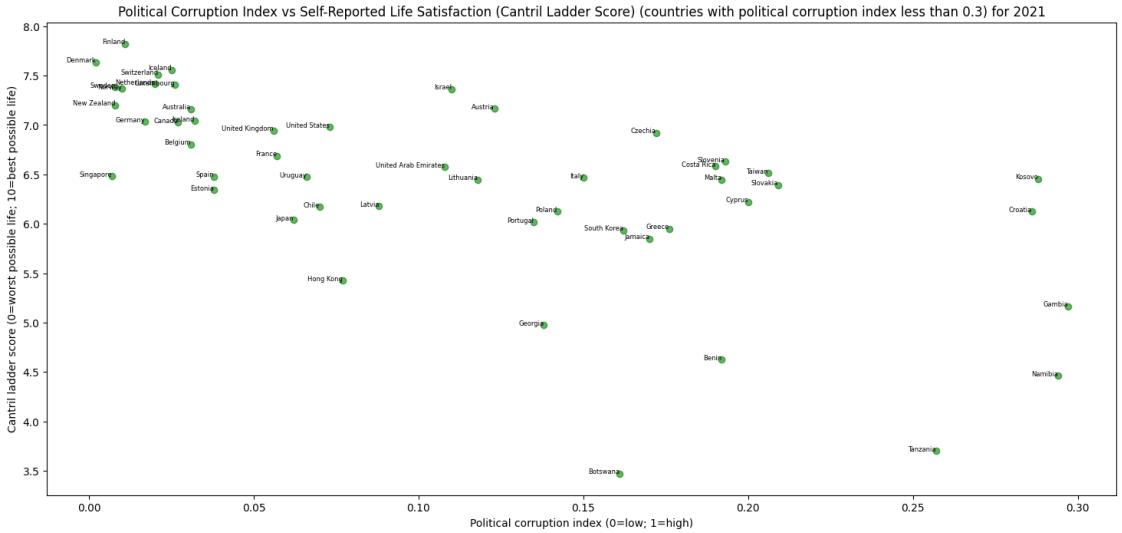
[65]:
```
df_life_satisfaction_political_corruption.plot.scatter(x='Political corruption␣
 ↪index', y='Cantril ladder score')
plt.xlabel('Political corruption index (0=low; 1=high)')
plt.ylabel('Cantril ladder score (0=worst; 10=best)')
plt.title('Political Corruption Index vs Self-Reported Life Satisfaction␣
 ↪(Cantril Ladder Score) (world countries) for 2021')
plt.show()
```



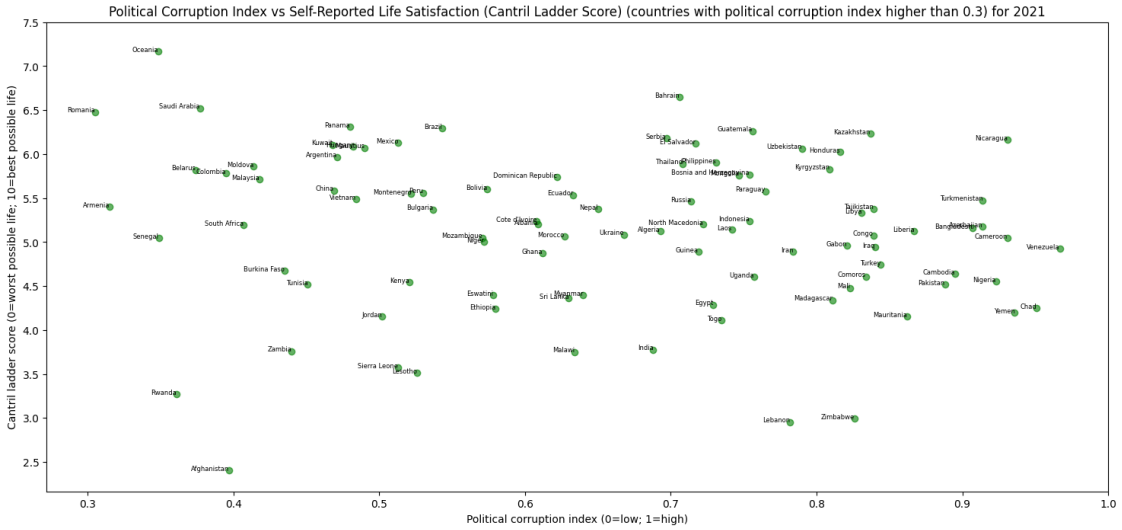
[66]:
```
plt.figure(figsize=(18, 8))
plt.scatter(df_life_satisfaction_political_corruption['Political corruption␣
 ↪index'], df_life_satisfaction_political_corruption['Cantril ladder score'],␣
 ↪color='green', alpha=0.6)

for i, country in␣
 ↪enumerate(df_life_satisfaction_political_corruption['Country']):
    plt.text(df_life_satisfaction_political_corruption['Political corruption␣
 ↪index'].iloc[i], df_life_satisfaction_political_corruption['Cantril ladder␣
 ↪score'].iloc[i], df_life_satisfaction_political_corruption['Country'].
 ↪iloc[i], fontsize=6, ha='right')

plt.xticks([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
plt.yticks(range(0, 9))

plt.xlabel('Political corruption index (0=low; 1=high)')
plt.ylabel('Cantril ladder score (0=worst possible life; 10=best possible␣
 ↪life)')
plt.title('Political Corruption Index vs Self-Reported Life Satisfaction␣
 ↪(Cantril Ladder Score) (world countries) for 2021')
plt.show()
```



[67]:
```
plt.figure(figsize=(18, 8))
df_low_political_corruption =␣
 ↪df_life_satisfaction_political_corruption[df_life_satisfaction_political_corruption['Politi
 ↪corruption index'] <= 0.3]
plt.scatter(df_low_political_corruption['Political corruption index'],␣
 ↪df_low_political_corruption['Cantril ladder score'], color='green', alpha=0.
 ↪6)

for i, country in enumerate(df_low_political_corruption['Country']):
    plt.text(df_low_political_corruption['Political corruption index'].iloc[i],␣
 ↪df_low_political_corruption['Cantril ladder score'].iloc[i],␣
 ↪df_low_political_corruption['Country'].iloc[i], fontsize=6, ha='right')

plt.yticks([3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8])

plt.xlabel('Political corruption index (0=low; 1=high)')
plt.ylabel('Cantril ladder score (0=worst possible life; 10=best possible␣
 ↪life)')
plt.title('Political Corruption Index vs Self-Reported Life Satisfaction␣
 ↪(Cantril Ladder Score) (countries with political corruption index less than␣
 ↪0.3) for 2021')
plt.show()
```



[68]:
```
plt.figure(figsize=(18, 8))
df_high_political_corruption =␣
 ↪df_life_satisfaction_political_corruption[df_life_satisfaction_political_corruption['Politi
 ↪corruption index'] >= 0.3]
plt.scatter(df_high_political_corruption['Political corruption index'],␣
 ↪df_high_political_corruption['Cantril ladder score'], color='green', alpha=0.
 ↪6)

for i, country in enumerate(df_high_political_corruption['Country']):
    plt.text(df_high_political_corruption['Political corruption index'].
 ↪iloc[i], df_high_political_corruption['Cantril ladder score'].iloc[i],␣
 ↪df_high_political_corruption['Country'].iloc[i], fontsize=6, ha='right')

plt.yticks([2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5])

plt.xlabel('Political corruption index (0=low; 1=high)')
plt.ylabel('Cantril ladder score (0=worst possible life; 10=best possible␣
 ↪life)')
plt.title('Political Corruption Index vs Self-Reported Life Satisfaction␣
 ↪(Cantril Ladder Score) (countries with political corruption index higher␣
 ↪than 0.3) for 2021')
plt.show()
```



6.1 Which are the top 10 happiest world countries?

[69]:
```
top_10_happiest_countries = df_life_satisfaction_political_corruption.
↪sort_values(by='Cantril ladder score', ascending=False).head(10)
```

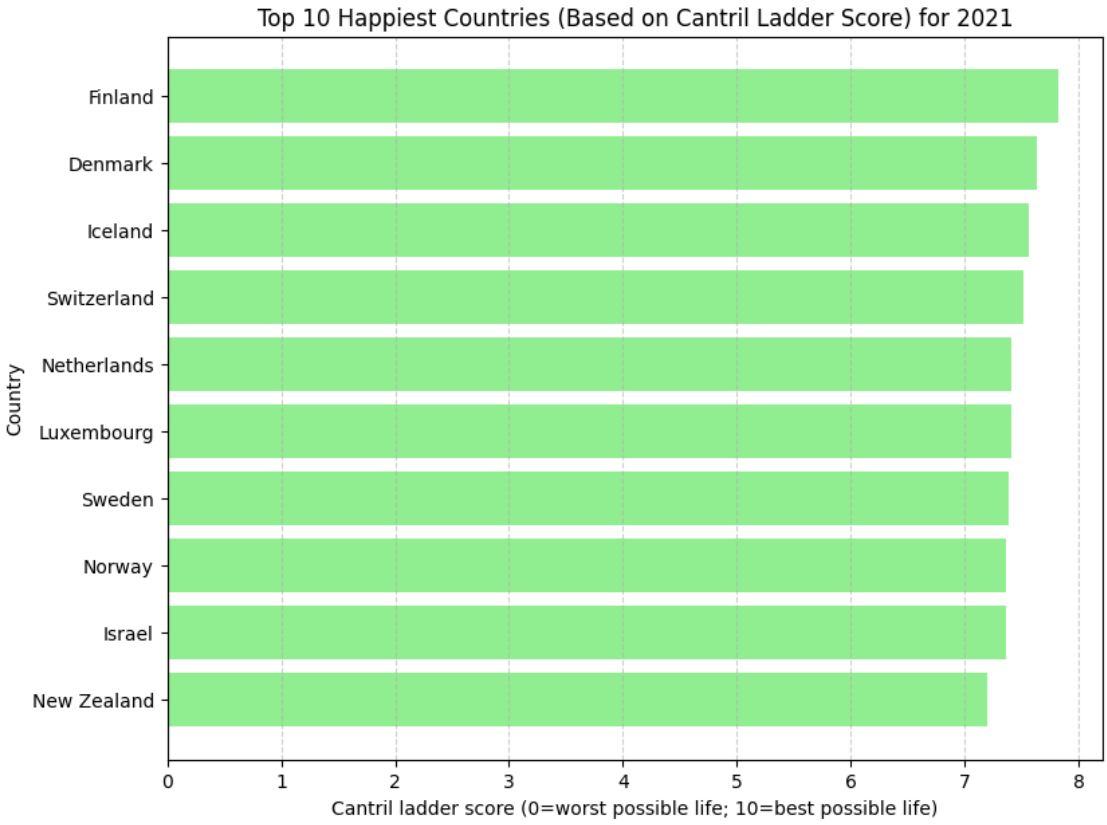
[70]:
```
top_10_happiest_countries
```

[70]:

	Country	Year_x	Cantril ladder score	Year_y	\
41	Finland	2021	7.8210	2021	
33	Denmark	2021	7.6362	2021	
54	Iceland	2021	7.5575	2021	
124	Switzerland	2021	7.5116	2021	
94	Netherlands	2021	7.4149	2021	
77	Luxembourg	2021	7.4040	2021	
123	Sweden	2021	7.3843	2021	
100	Norway	2021	7.3651	2021	
60	Israel	2021	7.3638	2021	
95	New Zealand	2021	7.1998	2021	

	Political corruption index
41	0.011
33	0.002
54	0.025
124	0.021
94	0.020
77	0.026
123	0.008
100	0.010
60	0.110
95	0.008

[71]:
```
plt.figure(figsize=(9, 7))
plt.barh(top_10_happiest_countries['Country'],␣
 ↪top_10_happiest_countries['Cantril ladder score'], color='lightgreen')
plt.xlabel('Cantril ladder score (0=worst possible life; 10=best possible␣
 ↪life)')
plt.ylabel('Country')
plt.title('Top 10 Happiest Countries (Based on Cantril Ladder Score) for 2021')
plt.gca().invert_yaxis()
plt.grid(axis='x', linestyle='--', alpha=0.6)
plt.show()
```



6.2 Which are the top 20 most politically corrupted world countries?

[72]:
```
top_20_most_politically_corrupted_countries =␣
 ↪df_life_satisfaction_political_corruption.sort_values(by='Political␣
 ↪corruption index', ascending=False).head(20)
```

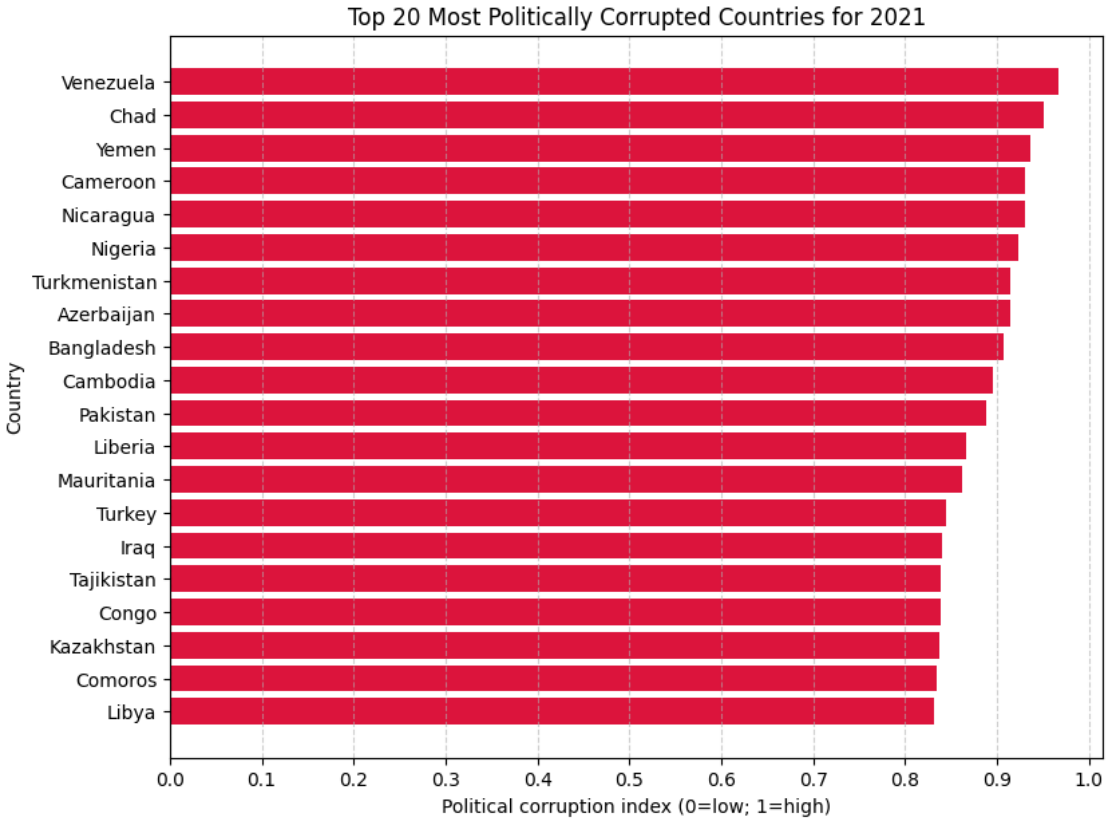
[73]:
```
top_20_most_politically_corrupted_countries
```

[73]:
```
          Country  Year_x  Cantril ladder score  Year_y  \
140     Venezuela    2021                4.9255    2021
```

```
22           Chad    2021                4.2508    2021
142         Yemen    2021                4.1969    2021
20       Cameroon    2021                5.0476    2021
96      Nicaragua    2021                6.1646    2021
98        Nigeria    2021                4.5520    2021
132  Turkmenistan    2021                5.4743    2021
7      Azerbaijan    2021                5.1734    2021
9      Bangladesh    2021                5.1555    2021
19       Cambodia    2021                4.6403    2021
102      Pakistan    2021                4.5158    2021
74        Liberia    2021                5.1215    2021
83     Mauritania    2021                4.1526    2021
131        Turkey    2021                4.7442    2021
58           Iraq    2021                4.9409    2021
126    Tajikistan    2021                5.3771    2021
27          Congo    2021                5.0752    2021
65     Kazakhstan    2021                6.2341    2021
26        Comoros    2021                4.6086    2021
75          Libya    2021                5.3302    2021

     Political corruption index
140                       0.967
22                        0.951
142                       0.936
20                        0.931
96                        0.931
98                        0.923
132                       0.914
7                         0.914
9                         0.907
19                        0.895
102                       0.888
74                        0.867
83                        0.862
131                       0.844
58                        0.840
126                       0.839
27                        0.839
65                        0.837
26                        0.834
75                        0.831
```

[74]:
```
plt.figure(figsize=(9, 7))
plt.barh(top_20_most_politically_corrupted_countries['Country'],␣
 ↪top_20_most_politically_corrupted_countries['Political corruption index'],␣
 ↪color='crimson')
plt.xticks([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
```

```
plt.xlabel('Political corruption index (0=low; 1=high)')
plt.ylabel('Country')
plt.title('Top 20 Most Politically Corrupted Countries for 2021')
plt.gca().invert_yaxis()
plt.grid(axis='x', linestyle='--', alpha=0.6)
plt.show()
```



6.3 Find whether each country with low political corruption index have high Cantril ladder score?

[75]:
```
upper_boundary_low_political_corruption = np.
 ↪quantile(df_life_satisfaction_political_corruption['Political corruption␣
 ↪index'], 0.25)
print('25th Percentile political corruption index is',␣
 ↪upper_boundary_low_political_corruption)
```

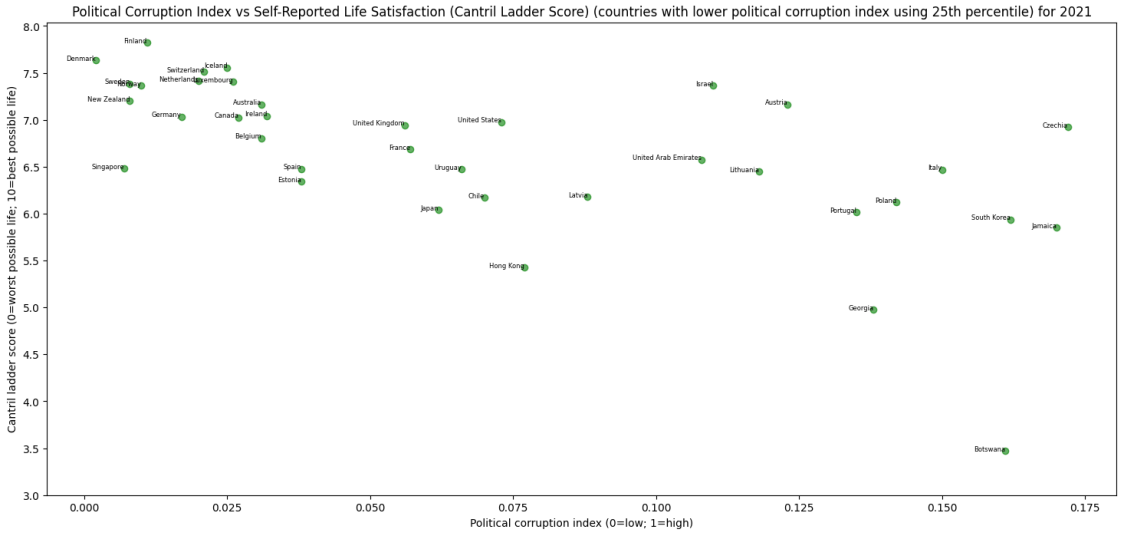
```
25th Percentile political corruption index is 0.172
```

[76]:
```
df_lower_political_corruption = df_life_satisfaction_political_corruption[
    df_life_satisfaction_political_corruption['Political corruption index'] <=␣
 ↪upper_boundary_low_political_corruption]
```

```
print('The number of countries with lower political corruption index using 25th␣
 ↪percentile is', len(df_lower_political_corruption))
print('The list of countries with lower political corruption index using 25th␣
 ↪percentile is:', ', '.join(df_lower_political_corruption['Country']))
```

```
The number of countries with lower political corruption index using 25th
percentile is 37
The list of countries with lower political corruption index using 25th
percentile is: Australia, Austria, Belgium, Botswana, Canada, Chile, Czechia,
Denmark, Estonia, Finland, France, Georgia, Germany, Hong Kong, Iceland,
Ireland, Israel, Italy, Jamaica, Japan, Latvia, Lithuania, Luxembourg,
Netherlands, New Zealand, Norway, Poland, Portugal, Singapore, South Korea,
Spain, Sweden, Switzerland, United Arab Emirates, United Kingdom, United States,
Uruguay
```

[77]:
```
plt.figure(figsize=(18, 8))
plt.scatter(df_lower_political_corruption['Political corruption index'],␣
 ↪df_lower_political_corruption['Cantril ladder score'], color='green',␣
 ↪alpha=0.6)

for i, country in enumerate(df_lower_political_corruption['Country']):
    plt.text(df_lower_political_corruption['Political corruption index'].
 ↪iloc[i], df_lower_political_corruption['Cantril ladder score'].iloc[i],␣
 ↪df_lower_political_corruption['Country'].iloc[i], fontsize=6, ha='right')

plt.yticks([3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8])

plt.xlabel('Political corruption index (0=low; 1=high)')
plt.ylabel('Cantril ladder score (0=worst possible life; 10=best possible␣
 ↪life)')
plt.title('Political Corruption Index vs Self-Reported Life Satisfaction␣
 ↪(Cantril Ladder Score) (countries with lower political corruption index␣
 ↪using 25th percentile) for 2021')
plt.show()
```

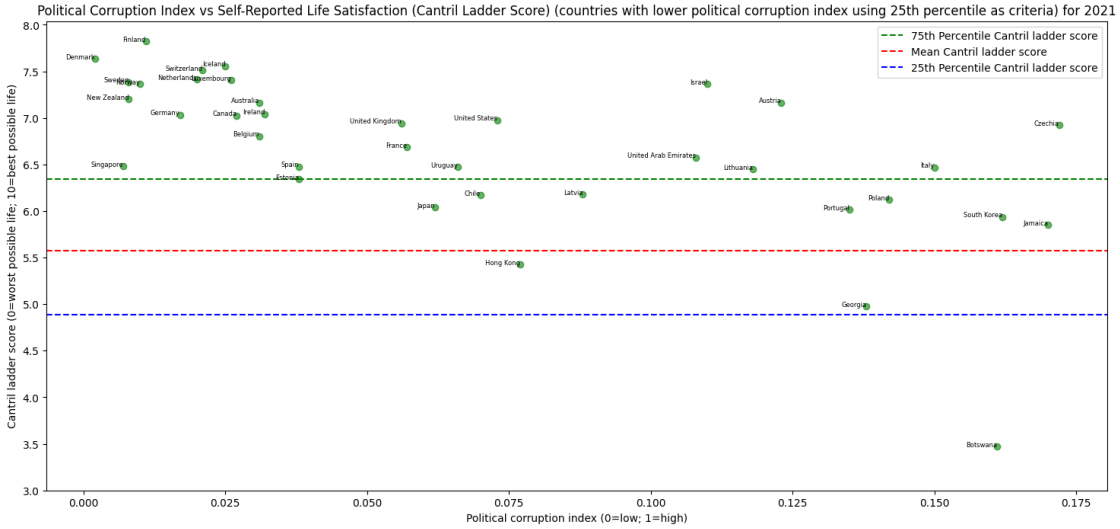


[78]:
```
lower_boundary_high_life_satisfaction = np.
 ↪quantile(df_life_satisfaction_political_corruption['Cantril ladder score'],␣
 ↪0.75)
print('75th Percentile Cantril ladder score is',␣
 ↪lower_boundary_high_life_satisfaction)
mean_life_satisfaction = np.
 ↪mean(df_life_satisfaction_political_corruption['Cantril ladder score'])
print('Mean Cantril ladder score is', mean_life_satisfaction)
upper_boundary_low_life_satisfaction = np.
 ↪quantile(df_life_satisfaction_political_corruption['Cantril ladder score'],␣
 ↪0.25)
print('25th Percentile Cantril ladder score is',␣
 ↪upper_boundary_low_life_satisfaction)
```

```
75th Percentile Cantril ladder score is 6.3408
Mean Cantril ladder score is 5.572698126206896
25th Percentile Cantril ladder score is 4.8905
```

[79]:
```
plt.figure(figsize=(18, 8))
plt.scatter(df_lower_political_corruption['Political corruption index'],␣
 ↪df_lower_political_corruption['Cantril ladder score'], color='green',␣
 ↪alpha=0.6)

for i, country in enumerate(df_lower_political_corruption['Country']):
    plt.text(df_lower_political_corruption['Political corruption index'].
 ↪iloc[i], df_lower_political_corruption['Cantril ladder score'].iloc[i],␣
 ↪df_lower_political_corruption['Country'].iloc[i], fontsize=6, ha='right')

plt.yticks([3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8])
```

```
plt.axhline(y=lower_boundary_high_life_satisfaction, color='green',␣
 ↪linestyle='--', label='75th Percentile Cantril ladder score')

plt.axhline(y=mean_life_satisfaction, color='red', linestyle='--', label='Mean␣
 ↪Cantril ladder score')

plt.axhline(y=upper_boundary_low_life_satisfaction, color='blue',␣
 ↪linestyle='--', label='25th Percentile Cantril ladder score')

plt.xlabel('Political corruption index (0=low; 1=high)')
plt.ylabel('Cantril ladder score (0=worst possible life; 10=best possible␣
 ↪life)')
plt.title('Political Corruption Index vs Self-Reported Life Satisfaction␣
 ↪(Cantril Ladder Score) (countries with lower political corruption index␣
 ↪using 25th percentile as criteria) for 2021')
plt.legend()
plt.show()
```



6.4 Check if there are highly politically corrupted countries that have high Cantril ladder score?

[80]:
```
lower_boundary_high_political_corruption = np.
 ↪quantile(df_life_satisfaction_political_corruption['Political corruption␣
 ↪index'], 0.75)
print('75th Percentile political corruption index is',␣
 ↪lower_boundary_high_political_corruption)
```

```
75th Percentile political corruption index is 0.735
```

[81]:
```
df_higher_political_corruption = df_life_satisfaction_political_corruption[
    df_life_satisfaction_political_corruption['Political corruption index'] >=␣
 ↪lower_boundary_high_political_corruption]
print('The number of countries with higher political corruption index using␣
 ↪75th percentile is', len(df_higher_political_corruption))
print('The list of countries with higher political corruption index using 75th␣
 ↪percentile is:', ', '.join(df_higher_political_corruption['Country']))
```

```
The number of countries with higher political corruption index using 75th
percentile is 37
The list of countries with higher political corruption index using 75th
percentile is: Azerbaijan, Bangladesh, Bosnia and Herzegovina, Cambodia,
Cameroon, Chad, Comoros, Congo, Gabon, Guatemala, Honduras, Indonesia, Iran,
Iraq, Kazakhstan, Kyrgyzstan, Laos, Lebanon, Liberia, Libya, Madagascar, Mali,
Mauritania, Mongolia, Nicaragua, Nigeria, Pakistan, Paraguay, Tajikistan, Togo,
Turkey, Turkmenistan, Uganda, Uzbekistan, Venezuela, Yemen, Zimbabwe
```

[82]:
```
df_higher_political_corruption_higher_life_satisfaction =␣
 ↪df_higher_political_corruption[df_higher_political_corruption['Cantril␣
 ↪ladder score'] > mean_life_satisfaction]
df_higher_political_corruption_higher_life_satisfaction
```

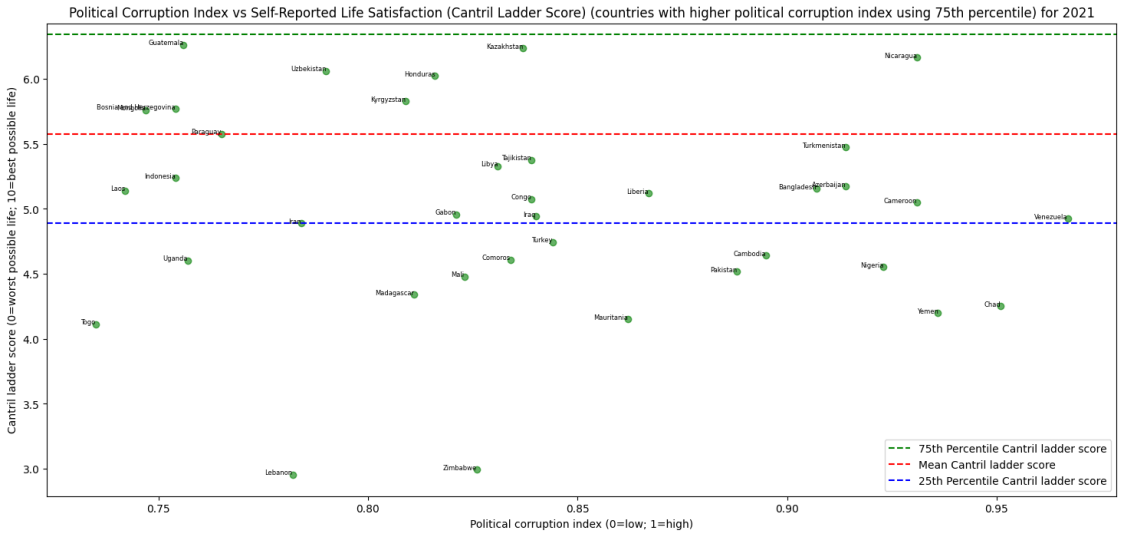
[82]:

	Country	Year_x	Cantril ladder score	Year_y	\
14	Bosnia and Herzegovina	2021	5.7680	2021	
49	Guatemala	2021	6.2622	2021	
51	Honduras	2021	6.0221	2021	
65	Kazakhstan	2021	6.2341	2021	
69	Kyrgyzstan	2021	5.8285	2021	
87	Mongolia	2021	5.7607	2021	
96	Nicaragua	2021	6.1646	2021	
104	Paraguay	2021	5.5783	2021	
139	Uzbekistan	2021	6.0627	2021	

	Political corruption index
14	0.754
49	0.756
51	0.816
65	0.837
69	0.809
87	0.747
96	0.931
104	0.765
139	0.790

[83]:
```
plt.figure(figsize=(18, 8))
```

```
plt.scatter(df_higher_political_corruption['Political corruption index'],␣
 ↪df_higher_political_corruption['Cantril ladder score'], color='green',␣
 ↪alpha=0.6)

for i, country in enumerate(df_higher_political_corruption['Country']):
    plt.text(df_higher_political_corruption['Political corruption index'].
 ↪iloc[i], df_higher_political_corruption['Cantril ladder score'].iloc[i],␣
 ↪df_higher_political_corruption['Country'].iloc[i], fontsize=6, ha='right')

plt.axhline(y=lower_boundary_high_life_satisfaction, color='green',␣
 ↪linestyle='--', label='75th Percentile Cantril ladder score')

plt.axhline(y=mean_life_satisfaction, color='red', linestyle='--', label='Mean␣
 ↪Cantril ladder score')

plt.axhline(y=upper_boundary_low_life_satisfaction, color='blue',␣
 ↪linestyle='--', label='25th Percentile Cantril ladder score')

plt.xlabel('Political corruption index (0=low; 1=high)')
plt.ylabel('Cantril ladder score (0=worst possible life; 10=best possible␣
 ↪life)')
plt.title('Political Corruption Index vs Self-Reported Life Satisfaction␣
 ↪(Cantril Ladder Score) (countries with higher political corruption index␣
 ↪using 75th percentile) for 2021')
plt.legend(loc="lower right")
plt.show()
```



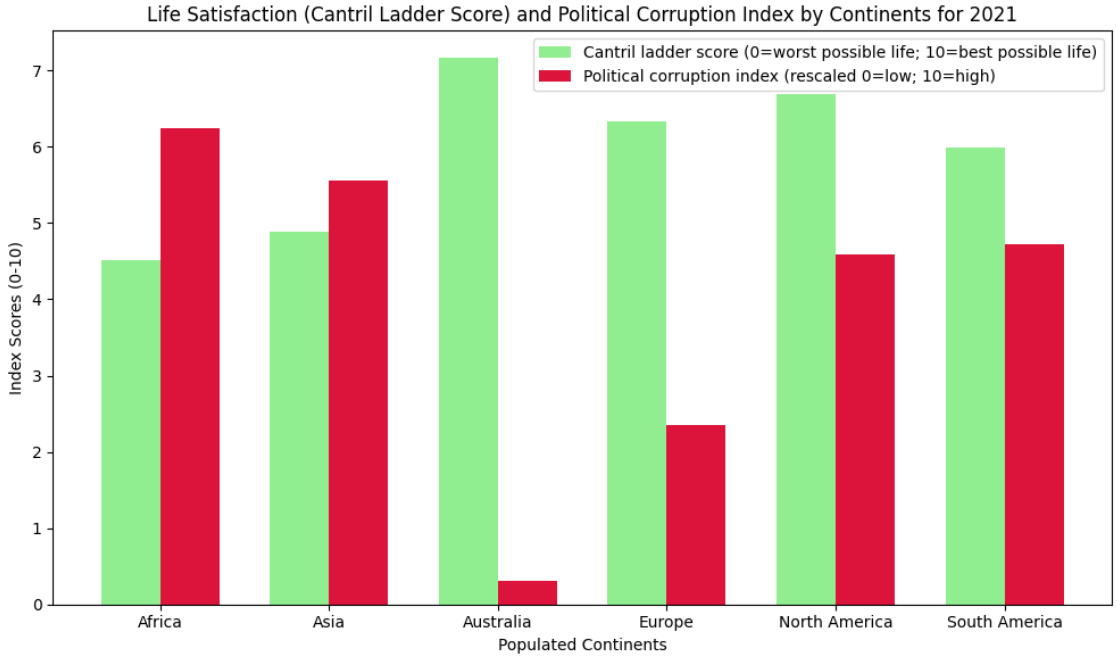
7. Draw a grouped bar plot of Life Satisfaction (Cantril Ladder Score) and Political Corruption Index by Continents for 2021

[84]:
```
df_political_corruption_continents
```

[84]:

	Country	Year	Political corruption index
467	Africa	2021	0.624089
1403	Asia	2021	0.555878
1638	Australia	2021	0.031000
9709	Europe	2021	0.235659
20953	North America	2021	0.458667
26875	South America	2021	0.472500

[85]:
```
x = np.arange(len(continents))
width = 0.35
fig, ax = plt.subplots(figsize=(10, 6))
bars1 = ax.bar(x - width/2, df_life_satisfaction_continents['Cantril ladder␣
 ↪score'], width, label='Cantril ladder score (0=worst possible life; 10=best␣
 ↪possible life)', color='lightgreen')
bars2 = ax.bar(x + width/2, df_political_corruption_continents['Political␣
 ↪corruption index'] * 10, width, label='Political corruption index (rescaled␣
 ↪0=low; 10=high)', color='crimson')

ax.set_xlabel('Populated Continents')
ax.set_ylabel('Index Scores (0-10)')
ax.set_title('Life Satisfaction (Cantril Ladder Score) and Political Corruption␣
 ↪Index by Continents for 2021')
ax.set_xticks(x)
ax.set_xticklabels(continents)
ax.legend()
plt.tight_layout()
plt.show()
```



Task 2: Investigate whether clustering can be used to identify volcanoes that lie on the same tectonic plate boundary

1. Prepare suitable SPARQL query that extracts volcano name, coordinates (latitude, longitude), country, elevation

[86]:
```
sparql_query = """
SELECT ?volcano ?volcanoLabel (SAMPLE(?coordinate) AS ?singleCoordinate)␣
 ↪(SAMPLE(?latitude) AS ?singleLatitude) (SAMPLE(?longitude) AS ?
 ↪singleLongitude) (SAMPLE(?countryLabel) AS ?singleCountryLabel) (SAMPLE(?
 ↪elevation) AS ?singleElevation) WHERE {
  ?volcano wdt:P31/wdt:P279* wd:Q8072 .  # instance of volcano
  ?volcano wdt:P625 ?coordinate . # coordinate location
  ?volcano wdt:P2044 ?elevation . # elevation above sea level
  ?volcano wdt:P17 ?country . # country
  ?country rdfs:label ?countryLabel .  # country label
  FILTER(LANG(?countryLabel) = "en")
  SERVICE wikibase:label { bd:serviceParam wikibase:language␣
 ↪"[AUTO_LANGUAGE],en". }
  BIND(geof:latitude(?coordinate) AS ?latitude).
  BIND(geof:longitude(?coordinate) AS ?longitude).
}
GROUP BY ?volcano ?volcanoLabel
"""
```

2. Execute the query to the Wikidata API with the help of the *request* Python package

[87]:
```
wikidata_url = "https://query.wikidata.org/sparql"

response = requests.get(wikidata_url, params={'query': sparql_query, 'format':␣
 ↪'json'})

if response.status_code == 200:
    data = response.json()
else:
    print("There is an error with status code:", response.status_code)
    data = None
```

3. Process the results and load them in a Pandas data frame

[88]:
```
if data:
    results = data['results']['bindings']

    processed_data = []
    for result in results:
        volcano = result['volcanoLabel']['value'] if 'volcanoLabel' in result␣
 ↪else None
        coordinate = result['singleCoordinate']['value'] if 'singleCoordinate'␣
 ↪in result else None
        latitude = float(result['singleLatitude']['value']) if 'singleLatitude'␣
 ↪in result else None
        longitude = float(result['singleLongitude']['value']) if␣
 ↪'singleLongitude' in result else None
        country = result['singleCountryLabel']['value'] if 'singleCountryLabel'␣
 ↪in result else None
        elevation = result['singleElevation']['value'] if 'singleElevation' in␣
 ↪result else None
        processed_data.append({
            'Volcano': volcano,
            'Coordinate': coordinate,
            'Latitude': latitude,
            'Longitude': longitude,
            'Country': country,
            'Elevation': elevation
        })

    df_volcanoes = pd.DataFrame(processed_data)
```

[89]:
```
df_volcanoes.head(20)
```

[89]:

	Volcano
0	Mount Yōtei
1	Mount Chōkai
2	Mount Iwate
3	Mount Kaimon

```
4              Mount Tateshina
5           Mount Azuma-kofuji
6                Mount Tarumae
7                 Mount Hakone
8              Mount Kirishima
9   Daisetsuzan Volcanic Group
10             Mount Suribachi
11                 Mount Osore
12                  Beerenberg
13                 San Antonio
14                   Novarupta
15             Mount Cleveland
16            Mount Chiginagak
17         Fourpeaked Mountain
18                   Teneguía
19                        Tau

                                          Coordinate   Latitude   Longitude  \
0                  Point(140.806388888 42.827222222)  42.827222  140.806389
1                  Point(140.048888888 39.099166666)  39.099167  140.048889
2                        Point(141.00111111 39.8525)  39.852500  141.001111
3                         Point(130.528333333 31.18)  31.180000  130.528333
4                        Point(138.295 36.103611111)  36.103611  138.295000
5                                Point(140.25 37.73)  37.730000  140.250000
6                         Point(141.378055555 42.69)  42.690000  141.378056
7                         Point(139.023888888 35.23)  35.230000  139.023889
8   http://www.wikidata.org/.well-known/genid/89b6…  31.916667  130.866667
9                                Point(142.85 43.65)  43.650000  142.850000
10                 Point(141.287783333 24.749030555)  24.749031  141.287783
11                             Point(141.12 41.279)  41.279000  141.120000
12                  Point(-8.155880555 71.079666666)  71.079667   -8.155881
13                 Point(-17.848244444 28.485919444)  28.485919  -17.848244
14                Point(-155.156666666 58.266666666)  58.266667 -155.156667
15                     Point(-169.945 52.822222222)  52.822222 -169.945000
16                Point(-156.991388888 57.133611111)  57.133611 -156.991389
17                       Point(-153.671944444 58.77)  58.770000 -153.671944
18                  Point(-17.85166667 28.47333333)  28.473333  -17.851667
19                       Point(-170.72806 -14.31778) -14.317780 -170.728060

                  Country Elevation
0                   Japan      1898
1                   Japan      2236
2                   Japan    2038.2
3                   Japan       924
4                   Japan      2530
5                   Japan      1707
6                   Japan      1041
```

```
7                        Japan      1438
8                        Japan      1700
9                        Japan      2290
10                       Japan       169
11                       Japan       879
12                      Norway      2277
13                       Spain       631
14  United States of America         841
15  United States of America        1730
16  United States of America        2134
17  United States of America        1999
18                       Spain     430.7
19  United States of America         931
```

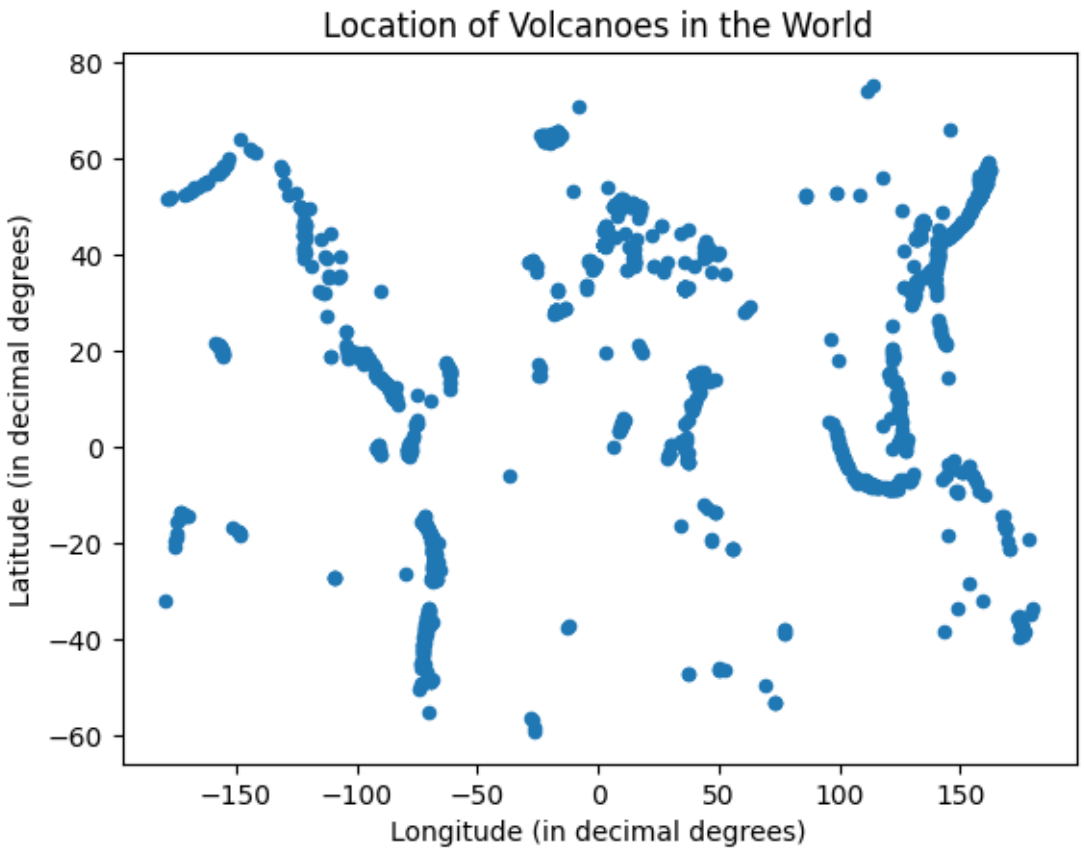
[90]:
```
# Remove rows with missing values (if there are any)
df_volcanoes = df_volcanoes.dropna()
```

[91]:
```
df_volcanoes.describe()
```

[91]:
```
          Latitude    Longitude
count  1655.000000  1655.000000
mean     20.286162    21.617850
std      29.777613   101.565439
min     -59.050000  -179.183333
25%      -1.468056   -71.474861
50%      21.450000     8.539167
75%      45.146525   131.034722
max      75.313056   179.871000
```

4. Investigate with visualisations whether clustering techniques can help in solving the problem

[92]:
```
# Create scatter plot of volcano locations using Pandas
df_volcanoes.plot.scatter(x='Longitude', y='Latitude')
plt.xlabel('Longitude (in decimal degrees)')
plt.ylabel('Latitude (in decimal degrees)')
plt.title('Location of Volcanoes in the World')
plt.show()
```



Tectonic Plate Boundaries

Data sources: Bird (2003); Argus et al. (2011)

https://www.kaggle.com/datasets/cwthompson/tectonic-plate-boundaries/data

[93]:
```
# Read latitude and longitude data that completely encloses 56 tectonic plates␣
↪so that we can validate the results visually
df_tectonic_plate_boundaries = pd.read_csv('./data/tectonic-plate-boundaries.
↪csv')
```

[94]:
```
df_tectonic_plate_boundaries.head()
```

[94]:

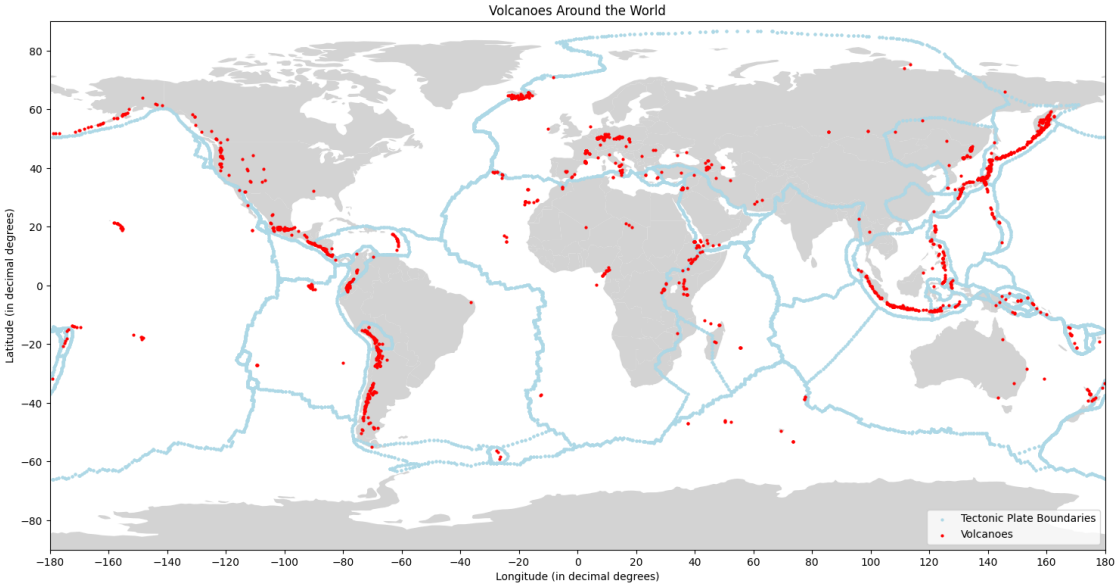
	plate	lat	lon
0	am	30.754	132.824
1	am	30.970	132.965
2	am	31.216	133.197
3	am	31.515	133.500
4	am	31.882	134.042

1:110m Cultural Vectors: Admin 0 - Countries

Data source: Natural Earth

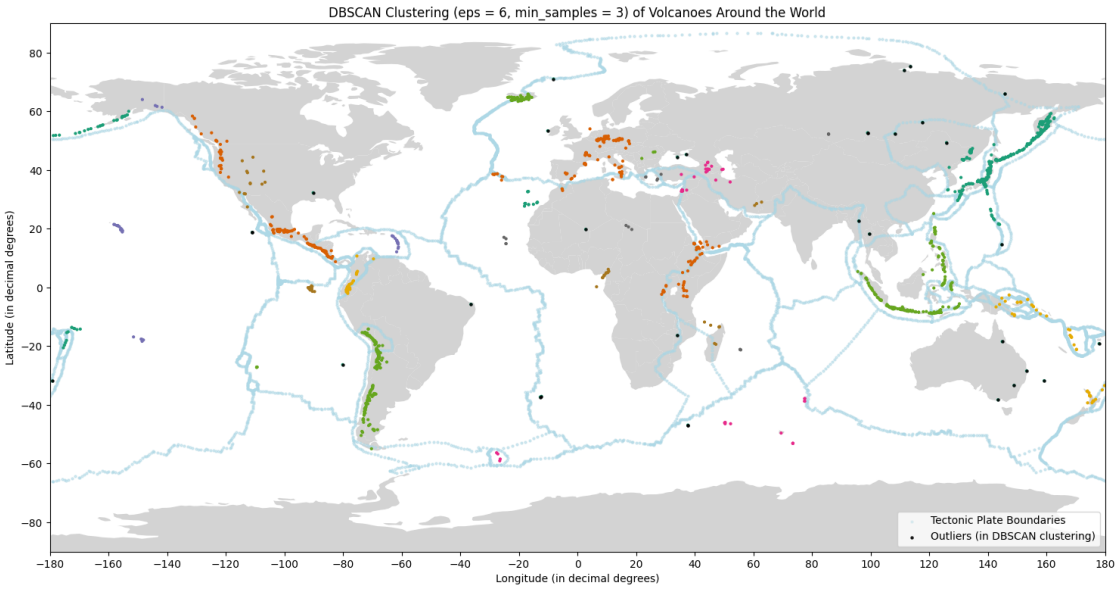
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/110m-admin-0-countries/

[95]:
```
# Use Python GeoPandas package to display world map for better clarity of the␣
 ↪volcano locations
world = gpd.read_file('./data/map/ne_110m_admin_0_countries_lakes.shp')

gdf_tectonic_plate_boundaries = gpd.GeoDataFrame(df_tectonic_plate_boundaries,␣
 ↪geometry=gpd.points_from_xy(df_tectonic_plate_boundaries['lon'],␣
 ↪df_tectonic_plate_boundaries['lat']))

gdf_volcanoes = gpd.GeoDataFrame(df_volcanoes, geometry=gpd.
 ↪points_from_xy(df_volcanoes['Longitude'], df_volcanoes['Latitude']))

fig, ax = plt.subplots(figsize=(18, 14))
world.plot(ax=ax, color='lightgray')

gdf_tectonic_plate_boundaries.plot(ax=ax, color='lightblue', markersize=4,␣
 ↪label='Tectonic Plate Boundaries')
gdf_volcanoes.plot(ax=ax, color='red', markersize=4, label='Volcanoes')

ax.set_xlim([-180, 180])
ax.set_xticks(range(-180, 181, 20))

ax.set_ylim([-90, 90])

plt.title('Volcanoes Around the World')
plt.xlabel('Longitude (in decimal degrees)')
plt.ylabel('Latitude (in decimal degrees)')
plt.legend(loc="lower right")
plt.show()
```



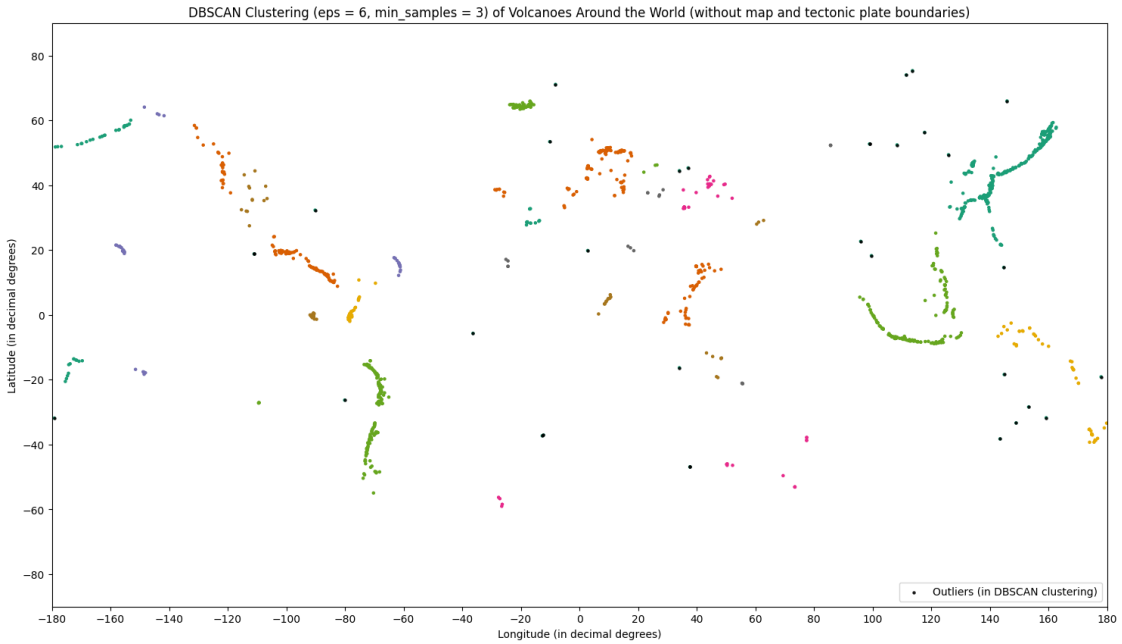
[96]:
```
# Apply DBSCAN clustering of world volcanoes
world = gpd.read_file('./data/map/ne_110m_admin_0_countries_lakes.shp')

gdf_tectonic_plate_boundaries = gpd.GeoDataFrame(df_tectonic_plate_boundaries,␣
 ↪geometry=gpd.points_from_xy(df_tectonic_plate_boundaries['lon'],␣
 ↪df_tectonic_plate_boundaries['lat']))

gdf_volcanoes = gpd.GeoDataFrame(df_volcanoes, geometry=gpd.
 ↪points_from_xy(df_volcanoes['Longitude'], df_volcanoes['Latitude']))

db = DBSCAN(eps=6, min_samples=3)
y_db = db.fit_predict(df_volcanoes[['Longitude', 'Latitude']])

core_samples_mask = np.zeros_like(y_db, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = y_db

num_clusters = len(set(labels)) - (1 if -1 in labels else 0)
num_outliers = list(labels).count(-1)

fig, ax = plt.subplots(figsize=(18, 14))

world.plot(ax=ax, color='lightgray')

gdf_tectonic_plate_boundaries.plot(ax=ax, color='lightblue', markersize=4,␣
 ↪alpha=0.3, label='Tectonic Plate Boundaries')
```

```
plt.scatter(df_volcanoes['Longitude'], df_volcanoes['Latitude'], c=y_db,␣
 ↪cmap='Dark2', s=4)

outliers_mask = labels == -1
plt.scatter(df_volcanoes['Longitude'][outliers_mask],␣
 ↪df_volcanoes['Latitude'][outliers_mask], c='black', s=5, alpha=0.8,␣
 ↪label='Outliers (in DBSCAN clustering)')

ax.set_xlim([-180, 180])
ax.set_xticks(range(-180, 181, 20))

ax.set_ylim([-90, 90])

plt.title('DBSCAN Clustering (eps = 6, min_samples = 3) of Volcanoes Around the␣
 ↪World')
plt.xlabel('Longitude (in decimal degrees)')
plt.ylabel('Latitude (in decimal degrees)')
plt.legend(loc="lower right")
plt.show()

print(f'Number of clusters: {num_clusters}')
print(f'Number of outliers: {num_outliers}')
```




```
Number of clusters: 37
Number of outliers: 33
```

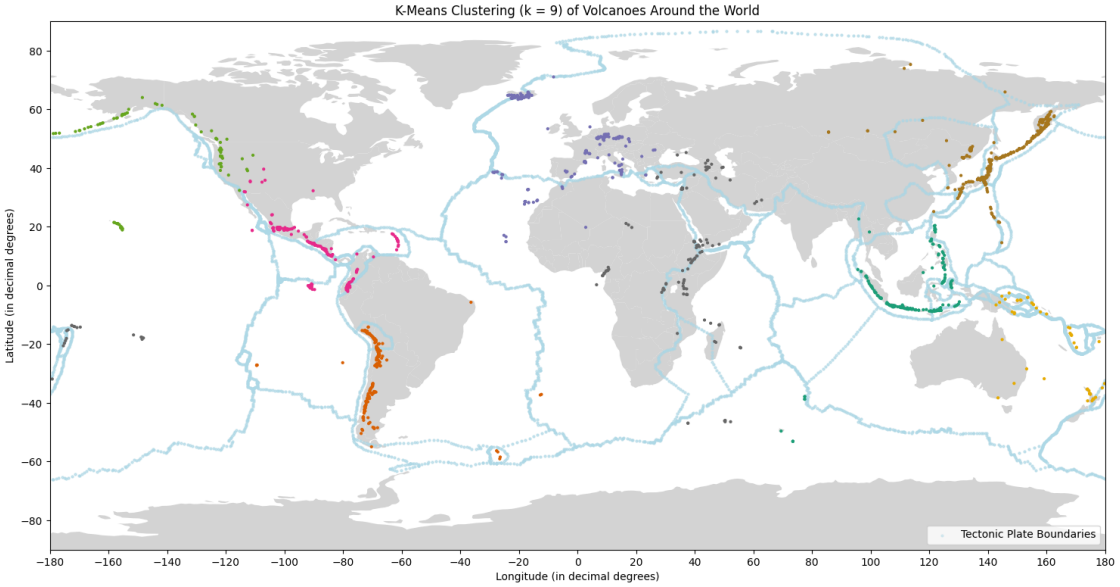
[97]:
```
# Apply DBSCAN clustering of world volcanoes (without loading map and tectonic␣
 ↪plate boundaries)
db = DBSCAN(eps=6, min_samples=3)
y_db = db.fit_predict(df_volcanoes[['Longitude', 'Latitude']])

core_samples_mask = np.zeros_like(y_db, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = y_db

num_clusters = len(set(labels)) - (1 if -1 in labels else 0)
num_outliers = list(labels).count(-1)

fig, ax = plt.subplots(figsize=(18, 10))

plt.scatter(df_volcanoes['Longitude'], df_volcanoes['Latitude'], c=y_db,␣
 ↪cmap='Dark2', s=5)

outliers_mask = labels == -1
plt.scatter(df_volcanoes['Longitude'][outliers_mask],␣
 ↪df_volcanoes['Latitude'][outliers_mask], c='black', s=5, alpha=0.8,␣
 ↪label='Outliers (in DBSCAN clustering)')

ax.set_xlim([-180, 180])
ax.set_xticks(range(-180, 181, 20))

ax.set_ylim([-90, 90])

plt.title('DBSCAN Clustering (eps = 6, min_samples = 3) of Volcanoes Around the␣
 ↪World (without map and tectonic plate boundaries)')
plt.xlabel('Longitude (in decimal degrees)')
plt.ylabel('Latitude (in decimal degrees)')
plt.legend(loc="lower right")
plt.show()

print(f'Number of clusters: {num_clusters}')
print(f'Number of outliers: {num_outliers}')
```




```
Number of clusters: 37
Number of outliers: 33
```

[98]:
```
# Apply KMeans clustering for world volcanoes
world = gpd.read_file('./data/map/ne_110m_admin_0_countries_lakes.shp')

gdf_tectonic_plate_boundaries = gpd.GeoDataFrame(df_tectonic_plate_boundaries,␣
 ↪geometry=gpd.points_from_xy(df_tectonic_plate_boundaries['lon'],␣
 ↪df_tectonic_plate_boundaries['lat']))

gdf_volcanoes = gpd.GeoDataFrame(df_volcanoes, geometry=gpd.
 ↪points_from_xy(df_volcanoes['Longitude'], df_volcanoes['Latitude']))

kmeans = KMeans(n_clusters=9, n_init=10)
kmeans.fit(df_volcanoes[['Longitude', 'Latitude']])
y_kmeans = kmeans.predict(df_volcanoes[['Longitude', 'Latitude']])

fig, ax = plt.subplots(figsize=(18, 14))

world.plot(ax=ax, color='lightgray')

gdf_tectonic_plate_boundaries.plot(ax=ax, color='lightblue', markersize=4,␣
 ↪alpha=0.4, label='Tectonic Plate Boundaries')

plt.scatter(df_volcanoes['Longitude'], df_volcanoes['Latitude'], c=y_kmeans,␣
 ↪cmap='Dark2', s=4)
```

```
ax.set_xlim([-180, 180])
ax.set_xticks(range(-180, 181, 20))

ax.set_ylim([-90, 90])

plt.title('K-Means Clustering (k = 9) of Volcanoes Around the World')
plt.xlabel('Longitude (in decimal degrees)')
plt.ylabel('Latitude (in decimal degrees)')
plt.legend(loc="lower right")
plt.show()
```



[99]:
```
# Apply KMeans clustering for world volcanoes (without loading map and tectonic␣
↪plate boundaries)
kmeans = KMeans(n_clusters=9, n_init=10)
kmeans.fit(df_volcanoes[['Longitude', 'Latitude']])
y_kmeans = kmeans.predict(df_volcanoes[['Longitude', 'Latitude']])

fig, ax = plt.subplots(figsize=(18, 10))

plt.scatter(df_volcanoes['Longitude'], df_volcanoes['Latitude'], c=y_kmeans,␣
↪cmap='Dark2', s=4)

ax.set_xlim([-180, 180])
ax.set_xticks(range(-180, 181, 20))

ax.set_ylim([-90, 90])
```

```
plt.title('K-Means Clustering (k = 9) of Volcanoes Around the World (without␣
 ↪map and tectonic plate boundaries)')
plt.xlabel('Longitude (in decimal degrees)')
plt.ylabel('Latitude (in decimal degrees)')
plt.show()
```