Assignment 1

November 15, 2022

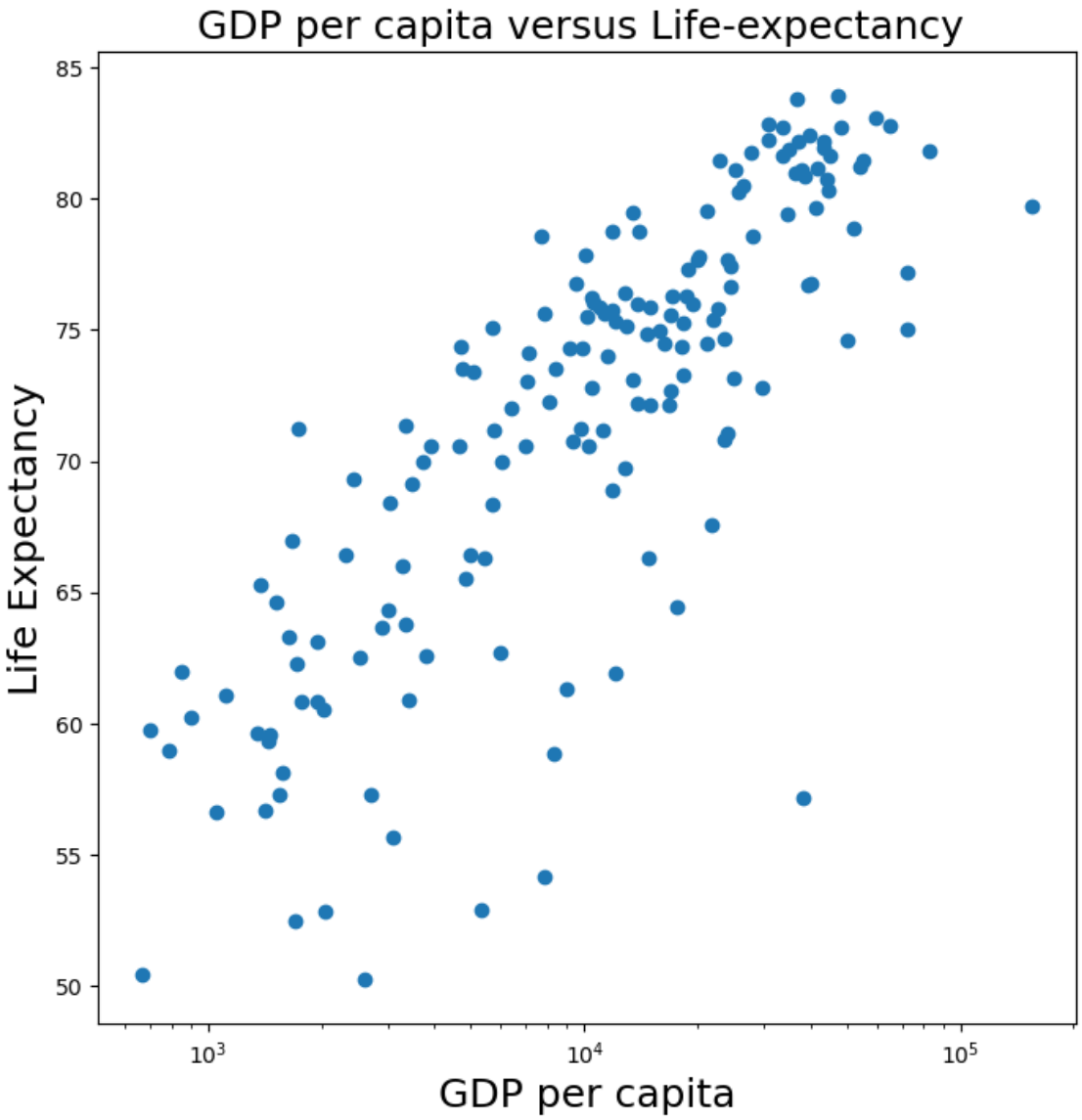
0.0.1 DAT405 Assignment 1 – Group 53

Venkata Sai Dinesh Uddagiri - (14 hrs)

Madhumitha Venkatesan - (14 hrs)

1 Problem 1

[1]:
```
#code to draws a scatter plot for GDP per capita versus Life-expectancy
import pandas as pnd
import matplotlib.pyplot as plt
#Reading csv file in to data frame
le_Gdp_DF_Raw=pnd.read_csv("life-expectancy-vs-gdp-per-capita.csv")
#Data cleaning
#le_gdp_DF_clean = le_Gdp_DF_Raw.dropna(subset = ['Life expectancy', 'GDP per␣
 ↪capita','Population (historical estimates)'])
le_gdp_DF_clean = (le_Gdp_DF_Raw[(le_Gdp_DF_Raw['Year'] >= 2011) &␣
 ↪(le_Gdp_DF_Raw['Year'] <= 2019)]
                   .dropna(subset = ['Life expectancy', 'GDP per␣
 ↪capita','Population (historical estimates)']))
del le_gdp_DF_clean["417485-annotations"], le_gdp_DF_clean["Continent"]
#Obtaining mean 'Life expectancy','GDP per capita' and 'Population (historical␣
 ↪estimates)'of based on 'Entity'
le_gdp_DF_clean =le_gdp_DF_clean.groupby(['Entity'])[['Life expectancy','GDP␣
 ↪per capita','Population (historical estimates)'
                                                      ]].agg('mean')
plt.figure(figsize=(8,8))
#Drawing scatter plot with corresponding axis Labels and title
plt.scatter(le_gdp_DF_clean['GDP per capita'], le_gdp_DF_clean['Life␣
 ↪expectancy'])
plt.title("GDP per capita versus Life-expectancy ", fontsize = 18)
plt.xlabel("GDP per capita", fontsize = 18)
plt.ylabel("Life Expectancy", fontsize = 18)
plt.xscale("log")
plt.show()
```



[2]:
```
#code to obatain data of life expectancy higher than one standard deviation␣
 ↪above the mean
#Calculation of standard deviation for Life expectancy
StandardDeviation = le_gdp_DF_clean.describe().loc['std', 'Life expectancy']
#Calculation of mean for Life expectancy
mean = le_gdp_DF_clean.describe().loc['mean', 'Life expectancy']
#Printing mean and StandardDeviation of Life expectancy
print("Standard Deviation of Life expectancy is", StandardDeviation)
print("mean of Life expectancy is", StandardDeviation)
#calculation of life expectancy higher than one standard deviation above the␣
 ↪mean
```

```
le_gdp_DF_clean[le_gdp_DF_clean['Life expectancy'] > mean + StandardDeviation].
 ↪sort_values(by='Life expectancy', ascending = False)
```

```
Standard Deviation of Life expectancy is 8.235309996831452
mean of Life expectancy is 8.235309996831452
```

[2]:

Entity	Life expectancy	GDP per capita	Population (historical estimates)
Hong Kong	83.912375	47422.371582	7.354659e+06
Japan	83.783125	36879.333984	1.272725e+08
Switzerland	83.050125	59549.271484	8.225456e+06
Spain	82.848500	30969.954834	4.660428e+07
Singapore	82.765625	65233.171875	5.581489e+06
Italy	82.739625	33758.812988	6.013556e+07
Australia	82.706625	48068.301758	2.365665e+07
Iceland	82.412250	39847.472168	
Israel	82.249000	30972.357666	
Sweden	82.176750	43394.454102	
France	82.146875	37097.864746	
Canada	81.948875	43508.422852	
South Korea	81.893375	34936.228027	
Norway	81.833375	82543.058594	
Malta	81.775250	27722.480225	
New Zealand	81.620875	33657.760742	
Netherlands	81.612750	45136.357910	
Luxembourg	81.471500	55286.381348	
Greece	81.436500	22866.235840	
Ireland	81.226000	54351.322266	
Austria	81.126500	41703.555176	
Portugal	81.088250	25180.673828	
Finland	81.078375	37697.282227	
United Kingdom	80.965500	36540.739258	
Belgium	80.875625	38555.545898	
Germany	80.712750	44327.925293	
Slovenia	80.520375	26563.012451	
Denmark	80.337250	44537.237305	
Cyprus	80.246625	25797.837891	

```
Iceland                          3.319149e+05
Israel                           7.946864e+06
Sweden                           9.805371e+06
France                           6.362392e+07
Canada                           3.561487e+07
South Korea                      5.061916e+07
Norway                           5.150985e+06
Malta                            4.541085e+05
New Zealand                      4.574982e+06
Netherlands                      1.699588e+07
Luxembourg                       5.631158e+05
Greece                           1.082847e+07
Ireland                          4.663496e+06
Austria                          8.607922e+06
Portugal                         1.040692e+07
Finland                          5.462579e+06
United Kingdom                   6.494353e+07
Belgium                          1.120814e+07
Germany                          8.205426e+07
Slovenia                         2.082166e+06
Denmark                          5.667706e+06
Cyprus                           1.182269e+06
```

[3]:
```
#code to obtain data of countries with high life expectancy but have low GDP
#Calculation of GDP
le_gdp_DF_clean['GDP'] = le_gdp_DF_clean['GDP per capita'] *␣
 ↪le_gdp_DF_clean['Population (historical estimates)']
#median of GDP
median_gdp = le_gdp_DF_clean['GDP'].median()
#median of life_expectancy
median_life_expectancy = le_gdp_DF_clean['Life expectancy'].median()
#To obtain Countries with high life expectancy but have low GDP
le_low_gdp_median=(le_gdp_DF_clean[(le_gdp_DF_clean['Life expectancy'] >␣
 ↪median_life_expectancy)&(le_gdp_DF_clean['GDP'] < median_gdp)]
            .sort_values(by='Life expectancy', ascending= False))
print("Countries with high life expectancy but have low GDP")
le_low_gdp_median
```

```
Countries with high life expectancy but have low GDP
```

[3]:

Entity	Life expectancy	GDP per capita	\
Iceland	82.412250	39847.472168	
Malta	81.775250	27722.480225	
Luxembourg	81.471500	55286.381348	
Slovenia	80.520375	26563.012451	
Cyprus	80.246625	25797.837891	

Costa Rica	79.492125	13528.503906
Barbados	78.763000	11903.410400
Lebanon	78.726125	14047.593506
Cuba	78.551000	7678.671814
Albania	77.817875	10090.535767
Croatia	77.759750	20144.334473
Panama	77.670500	20082.892334
Estonia	77.436750	24452.233154
Uruguay	77.315375	18956.496582
Bosnia and Herzegovina	76.770625	9487.344482
Bahrain	76.688250	39171.060059
Montenegro	76.271375	17091.506836
Saint Lucia	75.521125	10215.922485
North Macedonia	75.314500	12110.805786
Lithuania	74.633500	23653.969971
Latvia	74.453375	21217.309326
Mauritius	74.366875	18175.644531
Honduras	74.361750	4700.616821
Armenia	74.321250	9887.225342
Dominica	74.270750	9143.899536
Jamaica	74.121625	7091.520020

	Population (historical estimates)	GDP
Entity		
Iceland	3.319149e+05	1.322597e+10
Malta	4.541085e+05	1.258901e+10
Luxembourg	5.631158e+05	3.113263e+10
Slovenia	2.082166e+06	5.530861e+10
Cyprus	1.182269e+06	3.049998e+10
Costa Rica	4.865923e+06	6.582866e+10
Barbados	2.777172e+05	3.305782e+09
Lebanon	5.861784e+06	8.234395e+10
Cuba	1.132602e+07	8.696880e+10
Albania	2.885487e+06	2.911611e+10
Croatia	4.263472e+06	8.588481e+10
Panama	3.924834e+06	7.882202e+10
Estonia	1.319114e+06	3.225529e+10
Uruguay	3.396431e+06	6.438444e+10
Bosnia and Herzegovina	3.556453e+06	3.374129e+10
Bahrain	1.340734e+06	5.251798e+10
Montenegro	6.330304e+05	1.081944e+10
Saint Lucia	1.751548e+05	1.789367e+09
North Macedonia	2.105968e+06	2.550497e+10
Lithuania	2.983035e+06	7.056062e+10
Latvia	2.003633e+06	4.251169e+10
Mauritius	1.291875e+06	2.348066e+10
Honduras	9.209799e+06	4.329174e+10

Armenia	2.883458e+06	2.850940e+10
Dominica	6.966000e+04	6.369640e+08
Jamaica	2.785131e+06	1.975081e+10

[4]:
```
#Countries with strong economy (normally indicated by GDP) have low life␣
 ↪expectancy

le_low_gdp_median=(le_gdp_DF_clean[(le_gdp_DF_clean['Life expectancy'] <␣
 ↪median_life_expectancy)&(le_gdp_DF_clean['GDP'] > median_gdp)]
            .sort_values(by='Life expectancy', ascending= False))
print("Countries with strong economy (normally indicated by GDP) have low life␣
 ↪expectancy")
le_low_gdp_median
```

```
Countries with strong economy (normally indicated by GDP) have low life
expectancy
```

[4]:

Entity	Life expectancy	GDP per capita
Belarus	73.247125	18257.462891
Dominican Republic	73.108250	13497.166870
Guatemala	73.036500	7030.423035
Venezuela	72.648750	16956.385620
World	72.161000	14952.293620
Azerbaijan	72.105750	16795.295166
Bangladesh	71.336750	3362.787903
Ukraine	71.203625	9801.343018
Egypt	71.198625	11172.737549
Russia	71.079625	23963.692871
Kazakhstan	70.790875	23489.933350
Uzbekistan	70.775750	9369.529541
Indonesia	70.582750	10317.286377
Philippines	70.552750	6950.906128
Iraq	69.741500	12883.759399
India	68.372125	5688.076477
Turkmenistan	67.573750	21728.247803
Pakistan	66.412000	4967.703125
Myanmar	65.521875	4854.955444
Ethiopia	64.617000	1511.226059
Kenya	64.313250	3014.418457
Ghana	62.590000	3784.827881
Tanzania	62.516250	2525.604156
South Africa	61.902375	12122.894043
Angola	58.860000	8305.859680
Nigeria	52.870875	5343.100586

	Population (historical estimates)	GDP

Entity		
Belarus	9.699724e+06	1.770923e+11
Dominican Republic	1.034000e+07	1.395607e+11
Guatemala	1.585240e+07	1.114491e+11
Venezuela	3.003227e+07	5.092387e+11
World	7.599029e+09	1.136229e+14
Azerbaijan	9.783098e+06	1.643100e+11
Bangladesh	1.569232e+08	5.276994e+11
Ukraine	4.503724e+07	4.414254e+11
Egypt	9.655608e+07	1.078796e+12
Russia	1.445148e+08	3.463108e+12
Kazakhstan	1.770915e+07	4.159868e+11
Uzbekistan	3.072107e+07	2.878419e+11
Indonesia	2.574170e+08	2.655845e+12
Philippines	1.023262e+08	7.112597e+11
Iraq	3.689230e+07	4.753115e+11
India	1.314399e+09	7.476400e+12
Turkmenistan	5.713902e+06	1.241531e+11
Pakistan	2.093754e+08	1.040115e+12
Myanmar	5.125802e+07	2.488554e+11
Ethiopia	1.012741e+08	1.530481e+11
Kenya	4.632910e+07	1.396553e+11
Ghana	2.853810e+07	1.080118e+11
Tanzania	5.194656e+07	1.311964e+11
South Africa	5.505889e+07	6.674731e+11
Angola	2.768593e+07	2.299554e+11
Nigeria	1.817739e+08	9.712361e+11

[5]:
```
#
#Median of GDP per capita
median_gdp_per_capita = le_gdp_DF_clean['GDP per capita'].median()
#GDP per capita as an indicator of strong economy using median
le_low_gdp_median=(le_gdp_DF_clean[(le_gdp_DF_clean['Life expectancy'] <␣
 ↪median_life_expectancy)&(le_gdp_DF_clean['GDP per capita'] >␣
 ↪median_gdp_per_capita)]
          .sort_values(by='Life expectancy', ascending= False))
print("GDP per capita as an indicator of strong economy")
le_low_gdp_median
```

```
GDP per capita as an indicator of strong economy
```

[5]:

Entity	Life expectancy	GDP per capita	\
Belarus	73.247125	18257.462891	
Seychelles	73.158375	25047.468506	
Dominican Republic	73.108250	13497.166870	
Trinidad and Tobago	72.821000	29839.618652	
Venezuela	72.648750	16956.385620	

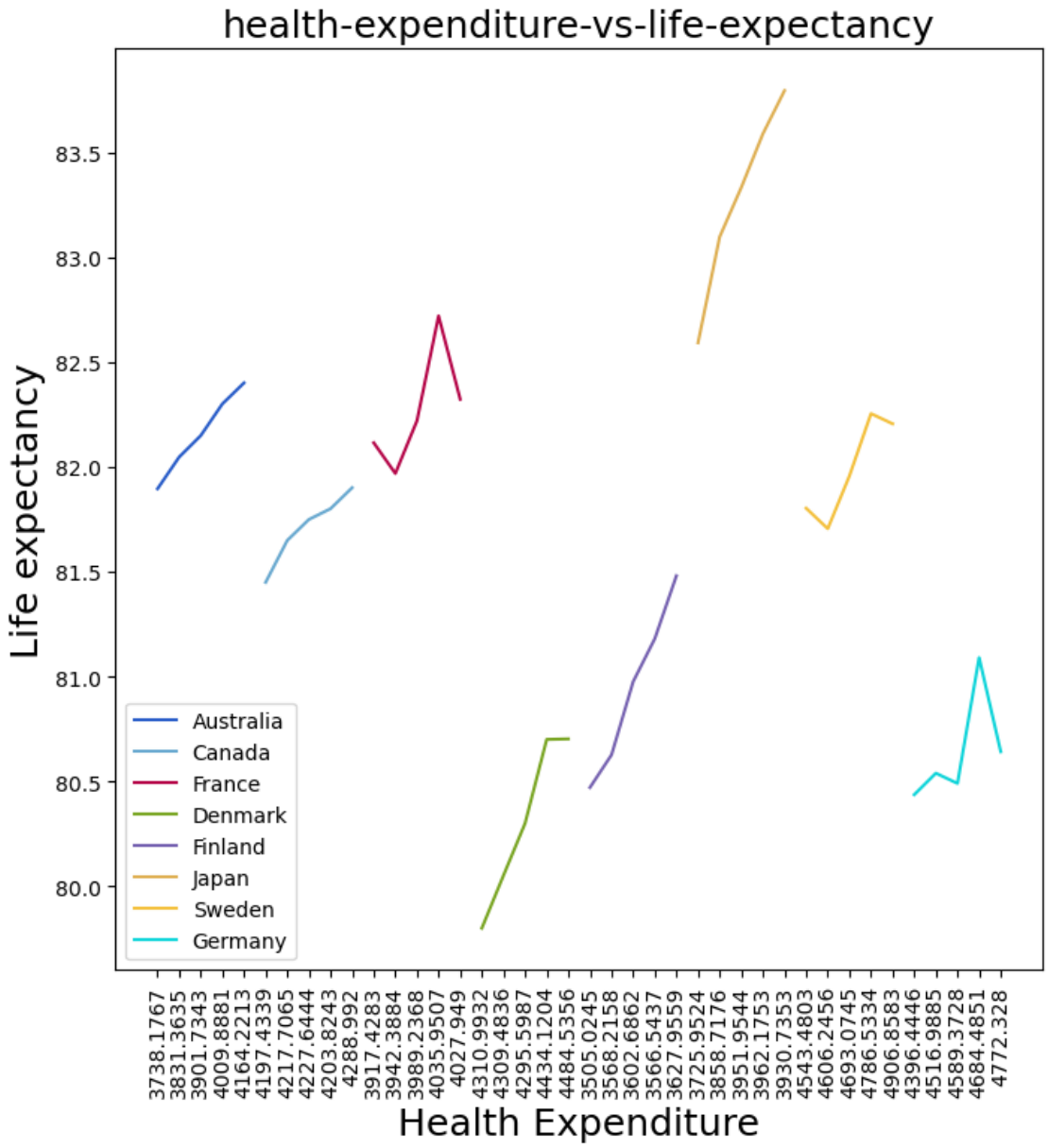
```
Libya                          72.177625    13794.094727
World                          72.161000    14952.293620
Azerbaijan                     72.105750    16795.295166
Russia                         71.079625    23963.692871
Kazakhstan                     70.790875    23489.933350
Iraq                           69.741500    12883.759399
Turkmenistan                   67.573750    21728.247803
Botswana                       66.283125    14787.069946
Gabon                          64.426875    17662.321289
South Africa                   61.902375    12122.894043
Equatorial Guinea              57.176375    38240.972656
```

Entity	Population (historical estimates)	GDP
Belarus	9.699724e+06	1.770923e+11
Seychelles	9.857950e+04	2.469167e+09
Dominican Republic	1.034000e+07	1.395607e+11
Trinidad and Tobago	1.456829e+06	4.347122e+10
Venezuela	3.003227e+07	5.092387e+11
Libya	6.183939e+06	8.530184e+10
World	7.599029e+09	1.136229e+14
Azerbaijan	9.783098e+06	1.643100e+11
Russia	1.445148e+08	3.463108e+12
Kazakhstan	1.770915e+07	4.159868e+11
Iraq	3.689230e+07	4.753115e+11
Turkmenistan	5.713902e+06	1.241531e+11
Botswana	2.287249e+06	3.382172e+10
Gabon	1.990660e+06	3.515967e+10
South Africa	5.505889e+07	6.674731e+11
Equatorial Guinea	1.322011e+06	5.055500e+10

2 Problem 2

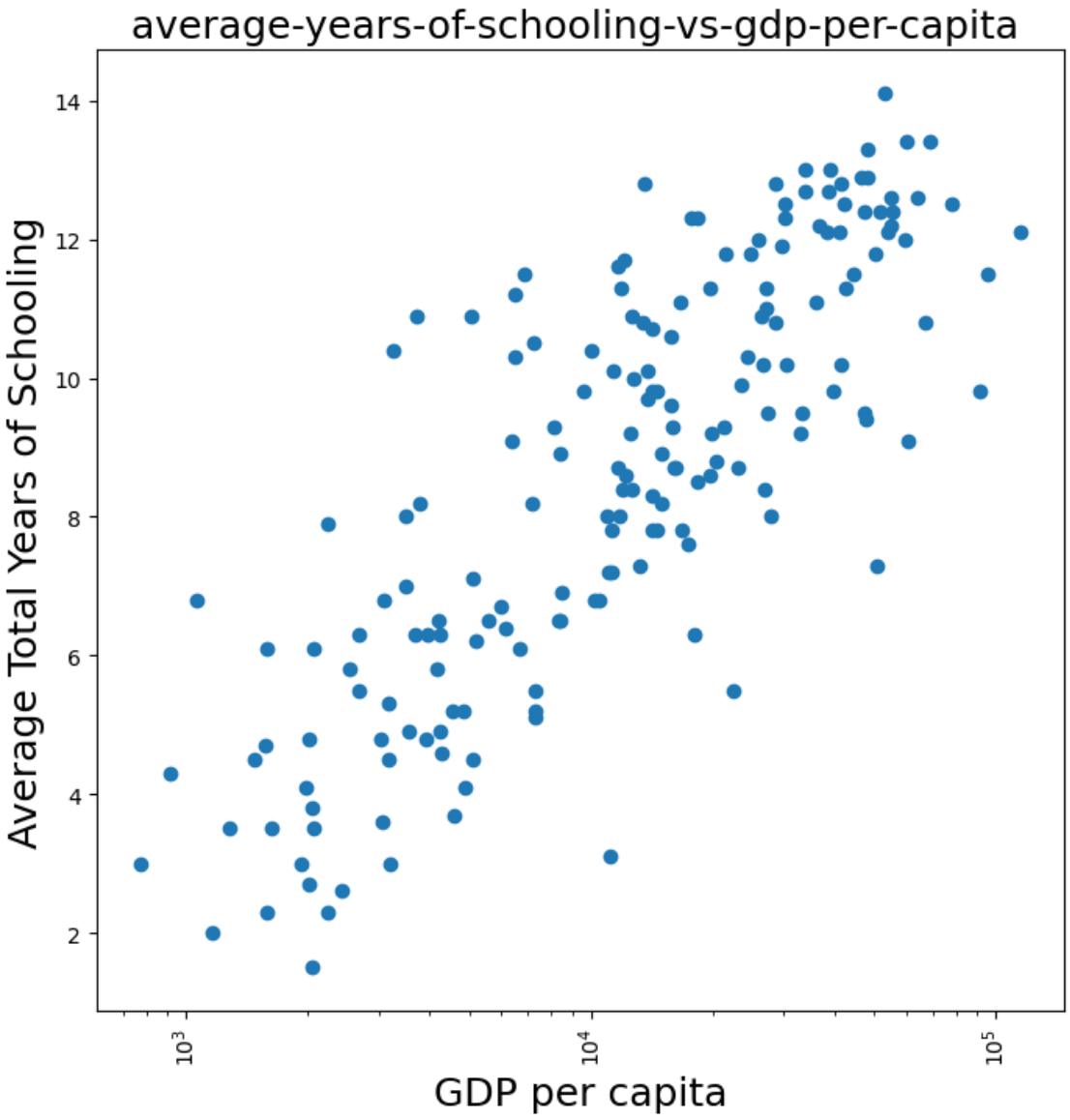
[6]:
```
#code for Health expenditure versus Life-expectancy
import pandas as pnd
import matplotlib.pyplot as plt
#Reading csv file in to data frame
import seaborn as sns
import numpy as np
le_he_DF_Raw=pnd.read_csv("life-expectancy-vs-health-expenditure.csv")
#Data cleaning
le_he_DF_clean = (le_he_DF_Raw[(le_he_DF_Raw['Year'] >= 2011) &␣
 ↪(le_he_DF_Raw['Year'] <= 2020)]
                  .dropna(subset = ['Life expectancy at birth, total (years)',␣
 ↪'Health Expenditure and Financing (per capita) (OECDstat (2017))'']))
```

```
Countries_list=␣
 ↪['Australia','Canada','France','Denmark','Finland','Japan','Sweden','Germany']
#plot size
plt.figure(figsize=(8,8))
for i in range(len(Countries_list)):
    df_Entity = le_he_DF_clean.loc[le_he_DF_clean['Entity'] ==␣
 ↪Countries_list[i]]
    df_Entity_le = df_Entity['Life expectancy at birth, total (years)']
    df_Entity_helExp = df_Entity['Health Expenditure and Financing (per capita)␣
 ↪(OECDstat (2017))']
    #generating random clour for each entity
    clr = (np.random.random(), np.random.random(), np.random.random())
    #Instance of plot
    plt.plot(df_Entity_helExp,df_Entity_le, color = clr)
#Assigning plot with corresponding axis Labels and title
plt.xticks(rotation ='vertical')
plt.legend(Countries_list, loc = "lower left")
fs=18
plt.title('health-expenditure-vs-life-expectancy',fontsize = fs)
plt.xlabel('Health Expenditure',fontsize = fs)
plt.ylabel('Life expectancy',fontsize = fs)
plt.show()
```



[7]:
```
#code for Average years of schooling versus GDP per capita
import pandas as pnd
import matplotlib.pyplot as plt
import seaborn as sns
#Reading csv file in to data frame
ayos_gdp_DF_Raw=pnd.read_csv("average-years-of-schooling-vs-gdp-per-capita.csv")
#Data cleaning
ayos_gdp_DF_clean = (ayos_gdp_DF_Raw[(ayos_gdp_DF_Raw['Year'] == 2017)]
```

```
                    .dropna(subset = ['Average Total Years of Schooling for Adult␣
 ↪Population (Lee-Lee (2016), Barro-Lee (2018) and UNDP (2018))', 'GDP per␣
 ↪capita, PPP (constant 2017 international $)']))
#Plot size
plt.figure(figsize=(8,8))
#Scatter plot with corresponding axis Labels and title
plt.scatter(ayos_gdp_DF_clean['GDP per capita, PPP (constant 2017 international␣
 ↪$)'], ayos_gdp_DF_clean['Average Total Years of Schooling for Adult␣
 ↪Population (Lee-Lee (2016), Barro-Lee (2018) and UNDP (2018))'])
fs=18
plt.title("average-years-of-schooling-vs-gdp-per-capita ", fontsize = fs)
plt.xlabel("GDP per capita", fontsize = 18)
plt.ylabel("Average Total Years of Schooling", fontsize = fs)
plt.xticks(rotation ='vertical')
plt.xscale('log')
plt.show()
```


average-years-of-schooling-vs-gdp-per-capita
Average Total Years of Schooling
GDP per capita
14
12
10
8
6
4
2
10^3
10^4
10^5