# Week 5 Exercise – NumPy Arrays and matplotlib Plotting

## Before attempting either part of this Week 5 exercise, be sure you have completed (and studied) the Python Tutorials. Include all relevant code and visualizations/plots.

## Do not forgot to thoroughly comment your code.

There are three separate pages of instructions. The results of your code execution should be pasted into the empty space below each set of instructions

## Part A: looping\_arrays.py

Exercise on Looping three dimensional numpy arrays

A.1. Read, study and execute looping\_arrays.py

#numpy import only needed if not imported already:

import numpy as np

arr3 = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])

print(arr3)

print(np.shape(arr3))

A.2. Within a single statement, create three nested loops to access all elements of the three-dimensional array arr3

Cut and paste your python code below:

# import numpy

import numpy as np

# looping & printing a 1d array

arr1 = np.array([1, 2, 3, 4])

print(arr1)

#display the shape(dimensions) of the array

print(np.shape(arr1))

# loop through the 1 dimensional array

for i in range(np.shape(arr1)[0]):

    print(arr1[i])

# This is an example of a two dimensional array

arr2 = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])

print(arr2)

print(np.shape(arr2))

# two nested loops to access all elements of

# the two dimensional array arr2

for ii in range(np.shape(arr2)[0]):

    for jj in range(np.shape(arr2)[1]):

        print(arr2[ii, jj])

# Now, create a 3-dimensional array

arr3 = np.array([

    [[1, 2, 3], [4, 5, 6]],

    [[7, 8, 9], [10, 11, 12]]

])

print(arr3)

print(np.shape(arr3))  # Should print (2, 2, 3)

# three nested loops to access all elements of the 3D array arr3

for i in range(np.shape(arr3)[0]):

    for j in range(np.shape(arr3)[1]):

        for k in range(np.shape(arr3)[2]):

            print(arr3[i, j, k])

## Part B: matplotlib Plotting

In Part B, you will use two datafiles (attached here):

* gapminder\_gdp\_asia.csv
* gapminder\_all.csv

B.1. Using gapminder\_gdp\_asia.csv create a line plot (plot) comparing the GDP of China and India for all years from 1952 to 2007. Include labels on the X and Y axis and a legend, with each country identified, in the lower right corner of the plot.

Cut and paste your all your code (from importing pandas, the dataset, preparation of the dataset for analysis and creation of the plot, with labels and legend and be sure to include a title above the plot--generated with code, NOT added manually). Your code should be complete and executable from beginning to end. Also include an image of your final plot below:

1# import matplotlib & pandas

import matplotlib.pyplot as plt

import pandas as pd

2# Read gapminder\_gdp\_asia.csv

data = pd.read\_csv('gapminder\_gdp\_asia.csv', index\_col= 'country')

3# Extract years from colunm

years = data.columns.str.strip('gdpPercap\_')

4# Convert year values to integers, saving results back to dataframe

data.columns = years.astype(int)

5# Plot China and india GDP from 1952-2007

data.loc['China'].plot()

data.loc['India'].plot()

#6 Labelling

plt.title('GDP of China vs. India from 1952 to 2007')

plt.legend(['China', 'India'], loc='lower right')

plt.ylabel('GDP per capita')

plt.xlabel('Years')

plt.savefig('my\_figure.png')

#7 Plot the results

plt.show()

A graph with blue and orange lines

AI-generated content may be incorrect.

B.2. This short script creates a plot showing the correlation between GDP and life expectancy for 2007, normalizing marker size by population:

<Import pandas and load the gapminder\_all.csv as data\_all. Declare ‘country’ as the column index.>

data\_all.plot(kind='scatter', x='gdpPercap\_2007', y='lifeExp\_2007',

s=data\_all['pop\_2007']/1e6)

Complete the code above and execute the code. Cut and paste your code and an image of the plot below. Be sure to include a title above the plot--generated with code, NOT added manually.

1# import pandas

import pandas as pd

import matplotlib.pyplot as plt

2# Read gapminder\_all.csv

data\_all = pd.read\_csv('gapminder\_all.csv', index\_col= 'country')

3# Create scatter plot for 2007 GDP per capita vs life expectancy

ax = data\_all.plot(kind='scatter',

x='gdpPercap\_2007',

y='lifeExp\_2007',

s=data\_all['pop\_2007']/1e6, figsize=(16, 8))

4# Labelling

ax.set\_xlabel('GDP per Capita in 2007')

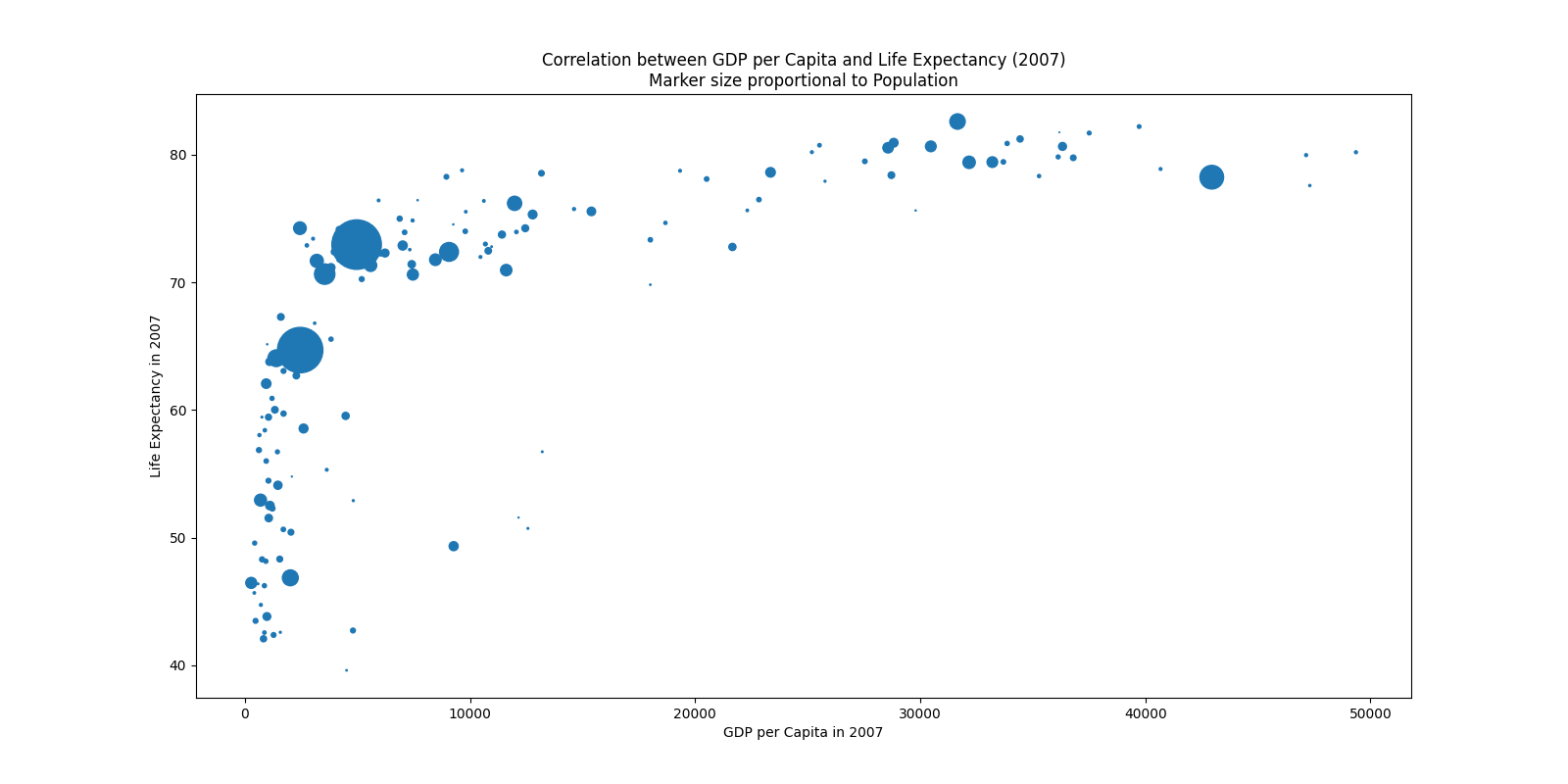
ax.set\_ylabel('Life Expectancy in 2007')

ax.set\_title('Correlation between GDP per Capita and Life Expectancy (2007)\nMarker size proportional to Population')

plt.savefig('my\_figure.png')

5# Plot the results

plt.show()



Using online help and documentation, explain in a sentence or two the meaning of GDP per Capita. Then define what each argument in plot does (**there are 4 arguments**, kind, x, y, s**).**  **Be specific as to how these arguments relate to your plot (no generic explanations that would apply to any plot).**

**Per capita GDP** is a measure of a country's economic output per person, calculated by dividing the total gross domestic product (GDP) by the population. This is commonly used to compare the average standard of living and economic health between countries.

The following represents:

* **kind='scatter':** Specifies that the plot should be a scatter plot, with each country represented as a point.
* **x='gdpPercap\_2007':** Plots the GDP per capita for 2007 on the horizontal axis.
* **y='lifeExp\_2007':** Plots the life expectancy for 2007 on the vertical axis.
* **s=data['pop\_2007']/1e6:** Determines the size of each point based on the country’s 2007 population (divided by 1 million), so larger countries visually stand out more.