DSCI 503 – Project 01 Instructions

General Instructions

Create a new notebook named **`Project_01_YourLastName.ipynb`** and complete the instructions provided below.

Any set of instructions you see in this document with an orange bar to the left will indicate a place where you should create a markdown cell. If no instructions are provided regarding formatting, then the text should be unformatted.

Any set of instructions you see with a blue bar to the left will provide instructions for creating a single code cell.

Read the instructions carefully.

Assignment Header

Create a markdown cell with a level 1 header that reads: "DSCI 503 – Project 01". Add your name below that as a level 3 header.

Introduction

In this project, you will be working with the gapminder dataset. This dataset contains socioeconomic data for 184 countries recorded over a period of 219 years, from 1800 to 2018.

Create a markdown cell with a level 1 header that reads "Introduction". Also include unformatted text explaining the purpose of this project. You may paraphrase the statement in the paragraph above.

The data you will be working with is contained in the file **`gapminder_data.txt`**. Download this file into the same folder that contains your project notebook.

We will use the **pandas** library to import the data into a **dataframe** object. We will discuss pandas and dataframes later in the course. For now, create and run a code cell containing the code shown below. This cell will load the data, and will display the first ten rows of the dataframe.

```
import pandas as pd
df = pd.read_csv('gapminder_data.txt', sep='\t')
df.head(10)
```

Information in a dataframe is represented as a table. Each row in this table represents a single country during a single year. Each column represents a different piece of information about each of the the countries. This dataframe contains the following columns:

- **`country`** - This column contains the names of the 184 countries. Each country appears in the dataframe 219 times - once for each year for which data was collected.
- **`continent`** - Each element of this column records the continental region in which the associated country belongs. There are four continental regions used in this dataset: `'africa'`, `'americas'`, `'asia'`, and `'europe'`.
- **`year`** - This column contains the year during which that particular record was collected.
- **`population`** - Each entry in this column is the population of a given country in a given year.
- **`life_exp`** - Each entry in this column is the average life expectancy in a given country in a given year.
- **`gdp_per_cap`** - Each entry in this column is the per capita GDP of a given country in a given year. GDP (or Gross Domestic Product) is a measure of the economic output of a country. The per capita GDP is the total GDP of the country divided by the population. This is sometimes use as a measure of the average wealth of a country's citizens.
- **`gini`** - Each entry in this column is the Gini score of a particular country in a given year. The Gini score is a measure of economic inequality and is recorded on a scale from 0 to 100.

Since we do not yet know how to work with DataFrames, we will convert each column of the DataFrame into its own list. Create a cell with the code shown below. This will create the following lists: **`country`**, **`continent`**, **`population`**, **`life_exp`**, **`pcgdp`**, and **`gini`**.

```
country = df.country.to_list()
continent = df.continent.to_list()
year = df.year.to_list()
population = df.population.to_list()
life_exp = df.life_exp.to_list()
pcgdp = df.gdp_per_cap.to_list()
gini = df.gini.to_list()
```

The lists we have created each contain 40,296 elements and are parallel in the sense that for any particular integer **N**, the elements of the lists at index **N** will all refer to the same country, during the same year. To better understand this concept, copy the code in the cell below. Run this cell a few times, each time selecting a different value of **N** between 0 and 40,295.

```
N = xxxx

print('Country:         ', country[N])
print('Continent:       ', continent[N])
print('Year:            ', year[N])
print('Population:      ', population[N])
print('Life Expectancy: ', life_exp[N])
print('Per Capita GDP:  ', pcgdp[N])
print('Gini Index:      ', gini[N])
```

Part 1: Displaying Past 20 Years of US Data

In this section, you will be asked to display the last twenty years of data for the United States.

Create a markdown cell with a level 1 header that reads "**Part 1: Displaying Past 20 Years of US Data**".
Write a brief description of what you will be accomplishing in this part of the project. Do not format this text.

Print the population, life expectancy, per capita GDP, and Gini index for the United States for every year since 1999. The information should be arranged in columns, with a header explaining what the columns mean. Use 4 spaces to separate the values in adjacent columns. The first few rows of your output should look exactly as shown below, without the indentation.

```
Year    Country          Population   LExp    pcGDP    Gini
-----------------------------------------------------------
1999    United States    279000000    76.8    44700    40.5
2000    United States    282000000    76.9    46000    40.5
```

You can accomplish this by looping over the records. With each new record considered, check to see if the year for the record is greater than or equal to 1999, and if the country associated with the record is equal to the string **`'United States'`**. If both criteria are true, then print the record as a new row in the output.

Part 2: Selecting the 2018 Data

From this point on in this project, we will only be interested in the data for the year 2018. We will now create six new lists, each of which will contain 2018 data for each of the 184 countries.

Create a markdown cell with a level 1 header that reads "**Part 2: Selecting the 2018 Data**".
Write a brief description of what you will be accomplishing in this part of the project. Do not format this text.

Create a code cell and perform the following tasks:

1. Create six empty lists named: **`country_2018`**, **`population_2018`**, **`continent_2018`**, **`life_exp_2018`**, **`pcgdp_2018`**, and **`gini_2018`**.
2. Populate the new lists by looping over the original lists. At each iteration of the loop, check to see if the associated record is from 2018. It is is, store each piece of the record in one of the six new lists, as appopriate.

When you have created the lists above, you might wish to check there length to confirm that each contains 184 elements. Do not include this code in the final version of your notebook, however.

Use the **`sum()`** function to calculate the total population of the world in 2018, storing the result in a variable **`global_population_2018`**. Print this result in the following format:

```
The global population in 2018 was xxxx.
```

I would suggest performing a Google search to check to see if the value you calculate is reasonable. You should expect the value you calculated to be exactly what you find on Google since both will just be estimates, but the two numbers should be relatively close to each other.

Part 3: Identifying Countries with Largest and Smallest Populations

In this part, you will be asked to display a list of the countries with the largest populations in 2018, as well as a list of countries with the smallest populations in 2018.

Create a markdown cell with a level 1 header that reads "**Part 3: Identifying Countries with Largest and Smallest Populations** ". Write a brief description of what you will be accomplishing in this part of the project. Do not format this text.

Print several lines of output displaying information for the ten countries with the largest populations in 2018. You can accomplish this task by performing the following steps:

1. Create a sorted copy of the list **`population_2018`**. Store this new list in a variable. Note the the list **`population_2018`** should NOT itself be sorted.
2. Use the sorted list to determine the population of the country with the tenth largest population. Store the results in a variable.
3. Print the first two lines displayed in the sample output shown below.
4. Use a loop to print information for each country with a population **greater than or equal to** the value you found in Step 2 above. Print the results as shown in the sample output below, including a period at the end of each line.

```
Countries with Largest Populations in 2018
----------------------------------------------------
The population of xxxx in 2018 was xxxx.
The population of xxxx in 2018 was xxxx.
```

Print several lines of output displaying information for the ten countries with the smallest populations in 2018. You can accomplish this task by performing the following steps:

1. Use the sorted list from the previous cell to determine the population of the country with the tenth smallest population. Store the results in a variable.
2. Print the first two lines displayed in the sample output shown below.
3. Use a loop to print information for each country with a population **less than or equal to** the value you found in Step 1 above. Print the results as shown in the sample output below, including a period at the end of each line.

```
Countries with Smallest Populations in 2018
-----------------------------------------------------
The population of xxxx in 2018 was xxxx.
The population of xxxx in 2018 was xxxx.
```

Part 4: Identifying Countries with Highest and Lowest Life Expectancies

In this part, you will be asked to display a list of the countries with the highest life expectancies in 2018, as well as a list of countries with the lowest life expectancies in 2018.

Create a markdown cell with a level 1 header that reads "**Part 4: Identifying Countries with Highest and Lowest Life Expectancies**". Write a brief description of what you will be accomplishing in this part of the project. Do not format this text.

Print several lines of output displaying information for the ten countries with the highest life expectancies in 2018. You can follow an approach similar to what you used in Part 3. Format your output as shown below.

```
Countries with Highest Life Expectancy in 2018
-----------------------------------------------------
The life expectancy of xxxx in 2018 was xxxx.
The life expectancy of xxxx in 2018 was xxxx.
```

Print several lines of output displaying information for the ten countries with the lowest life expectancies in 2018. You can follow an approach similar to what you used in Part 3. Format your output as shown below.

```
Countries with Lowest Life Expectancy in 2018
-----------------------------------------------------
The life expectancy of xxxx in 2018 was xxxx.
The life expectancy of xxxx in 2018 was xxxx.
```

Part 5: Calculating GDP by Country

The total GDP for a country is equal to its per capita GDP multiplied by its population. In this part, you will be asked to create a list that contains the GDP for each country in 2018, as well as the total GDP of the world as a whole.

Create a markdown cell with a level 1 header that reads "**Part 5: Calculating GDP by Country**".
Write a brief description of what you will be accomplishing in this part of the project. Do not format this text.

Create an empty list named **`gdp_2018`**. Use a loop to populate this list. Each new element of the list should be equal to the total GDP of a single country in 2018. The order of the elements in this new list should match that of the other lists for the 2018 data.

Use **`sum()`** to find the sum of **`gdp_2018`**, displaying the result with a message in the format shown below.

```
The total global GDP in 2018 was $xxxx.
```

Next, we will determine the country with the highest GDP in 2018 as well as the country with the lowest GDP in 2018.

Use the **`min()`** and **`max()`** functions and the **`index()`** method to find the country with the highest GDP in 2018 as well as the country with the lowest GDP in 2018. Print your results in the format shown below.

```
The country with the highest GDP in 2018 was xxxx with a GDP of xxxx.
The country with the lowest GDP in 2018 was xxxx with a GDP of xxxx.
```

Part 6: Grouping by Continent

The countries in this dataset are grouped into four continental regions: `africa`, `americas`, `asia`, and `europe`. In this part of the project, you will be asked to calculate the total population, per capita GDP, and life expectancy for each of these regions in 2018.

Create a markdown cell with a level 1 header that reads "**Part 6: Grouping by Continent**".
Write a brief description of what you will be accomplishing in this part of the project. Do not format this text.

In this cell, we will calculate the total population, per capita GDP, and life expectancy for each of the four continental regions in 2018. You can accomplish this by following the steps described below.

1. Create a list named **`continent_names`** containing four strings representing the names of the four continental regions, in alphabetical order. The names should be in lowercase.

2. Create three empty lists: **`pcgdp_2018_by_continent`**, **`pop_2018_by_continent`**, **`life_exp_2018_by_continent`**.

3. Loop over the elements of **`continent_names`**, considering a single continent during each iteration. Each time this loop executes, perform the following steps:

    - Create temp variables called **`temp_life_exp`**, **`temp_pop`**, and **`temp_gdp`**. Initialize all three variables to 0.
    - Loop over all 184 countries in the 2018 data. Each time this loop executes, check to see if the given country is in the continent that we are currently considering. If it is, then update the three temp variables. When the loop finishes, **`temp_pop`** should contain the total population for countries in the given continent and **`temp_gdp`** should contain the total GDP for countries in the given continent. The **`temp_life_exp`** variable should contain the weighted sum of the life expectancies for the countries in the given continent, with population being used as the weight.
    - After looping over all 184 countries, divide both **`temp_life_exp`** and **`temp_gdp`** by **`temp_pop`**, storing the results back into the variables **`temp_life_exp`** and **`temp_gdp`**.
    - Round **`temp_gdp`** to the nearest whole number, and round **`temp_life_exp`** to 1 decimal place.
    - Append the values of the three temp variables into the lists created in Step 2 above.

When your loop finishes, the four lists you created should each contain four elements. You might want to check this before moving on.

In the cell cell, add code to print the results of your calculations in the format shown below. The contents of the first column should be left-aligned, while the contents of the remaining columns should be right-aligned.

```
Continent    Population      pcGDP    Life Exp
-----------------------------------------------
Africa             xxxx       xxxx        xxxx
Americas           xxxx       xxxx        xxxx
Asia               xxxx       xxxx        xxxx
Europe             xxxx       xxxx        xxxx
```

Submission Instructions

When you are done, click **Kernel > Restart and Run All**. If any cell produces an error, then manually run every cell after that one, in order. Save your notebook, and then export the notebook as an HTML file. Upload the HTML file to Canvas and upload the IPYNB file to CoCalc, placing the file into the folder **Projects/Project 01**.