MCCG11503 / MEME19803 / MECG11503
Assignment 3
Session 202301

Instructions:

1. This is an individual assignment.

2. Deadline for submission of the assignment output is **5.00PM, 7 April 2023 (Friday of week 12)**.

3. In the case of late submission for the assignment output, 10% of the maximum marks will be deducted if the work is up to one day late (24 hours) and additional 10% of the maximum marks for each of the subsequent days.

4. Plagiarism is not allowed. If the works are found to be plagiarised, no marks will be given, and the incident will be reported to the university for further action.

5. Your output can be submitted via a link in WBLE or to the email: yeohg@utar.edu.my or through the Teams link.

Assignment Question

The file "whs2022_annex2.xlsx" contains official WHO statistics for selected health-related SDG indicators and selected Thirteenth General Programme of Work indicators, based on data available in early 2022. In addition, summary measures of health, such as (healthy) life expectancy and total population, are included. These statistics have been compiled primarily from publications and databases produced and maintained by WHO. In each instance, the source of the data series is provided.

Write a Python script that performs the following tasks in the given order:

1. Read the dataset into a dictionary called "data". Read only the worksheets "Annex 2-1" to "Annex 2-4" using default settings. Do not perform any other processing, e.g. setting index columns or header rows. Rename the dictionary keys as "annex1", "annex2", "annex3", "annex4".

   (2 marks)

2. Read the footnotes in the worksheet "Footnotes" of the dataset into a Series called "footnotes". This Series should use the footnote markers as its index. (2 marks)

3. The dataset in "Annex 2-1" has a slightly different structure compared to the other datasets. In particular, each of the first three fields have three sub-fields of their own – "Male", "Female", and "Both sexes". Hence, in the corresponding DataFrame, the first 5 rows has the following values:

Table 1

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	...
0	NaN	Total populationa (000s)	NaN	NaN	Life expectancy at birthb (years)	NaN	NaN	Healthy life expectancy at birthb (years)	NaN	NaN	...
1	Data type	Comparable estimates	NaN	NaN	Comparable estimates	NaN	NaN	Comparable estimates	NaN	NaN	...
2	NaN	Male	Female	Both sexes	Male	Female	Both sexes	Male	Female	Both sexes	...
3	Member State	2020	NaN	NaN	2019	NaN	NaN	2019	NaN	NaN	...
4	Afghanistan	19976	18952	38928	63.3	63.2	63.2	54.7	53.2	53.9	...

   Fill the missing values in the first row (index 0) and fourth row (index 3) using forward fill. Then, concatenate to the values in the first row the corresponding gender information. For example, the three values (after forward filled) of "Total populationa (000s)" in the first row become
   - "Total populationa (000s) Male",
   - "Total populationa (000s) Female", and
   - "Total populationa (000s) Both sexes",

   respectively. (2 marks)

4. All four datasets in "Annex 2-1" to "Annex 2-4" have the same general table structure. Create a **function** called "process_df" that takes a DataFrame as input and returns a DataFrame as output. The function performs the following operations in the given order on the input DataFrame:
   - Delete the last column.
   - Delete the second row (with the values ['Data type', 'Comparable estimates', 'NaN', ...]).
   - Delete the third row (with the values ['Male', 'Female', 'Both sexes', ...])
   - Delete all blank rows and all blank columns.
   - Some of the columns are mostly blank except for a few footnote marker symbols. Delete these columns. Use the index of the Series "footnotes" to get all the footnote marker symbols.
   - Set the first two rows as the two-level header of the DataFrame, where the first row is the outer level and the second row is the inner level, with the names "Statistic" and "Year", respectively.
   - Set the first column as the index of the DataFrame with the name 'Member State'.

   For example, calling

   ```
   process_df(data['annex1'])
   ```

   returns the following DataFrame (only a portion is shown in Table 2):

Table 2

Statistic	Total populationa (000s) Male	Total populationa (000s) Female	Total populationa (000s) Both sexes	Life expectancy at birthb (years) Male	Life expectancy at birthb (years) Female	Life expectancy at birthb (years) Both sexes	Healthy life expectancy at birthb (years) Male	Healthy life expectancy at birthb (years) Female
Year	2020	2020	2020	2019	2019	2019	2019	2019
Member State								
Afghanistan	19976	18952	38928	63.3	63.2	63.2	54.7	53.2
Albania	1465	1413	2878	76.3	79.9	78	68	70.3
Algeria	22154	21697	43851	76.2	78.1	77.1	66.7	66.1
Andorra	-	-	77	-	-	-	-	-
Angola	16261	16605	32866	60.7	65.5	63.1	53.6	56.2

   Call the function "process_df" on all member DataFrames of the dictionary "data". Use the variables "a1", "a2", "a3", and "a4" to reference the resulting DataFrames. (6 marks)

5. The outer column headers are too long. Create four Series called "ha1", "ha2", "ha3", and "ha4", respectively, to save the outer column headers for the DataFrames "a1", "a2", "a3", and "a4", respectively. Let these Series have the default index (index 0, 1, 2, …). Use the index of each Series to replace the long headers of the correponding DataFrame. (2 marks)

6. For each DataFrame "a1", "a2", "a3", and "a4", do the following:
   - Print the **total** number of missing values.
   - Display the rows with missing values.

- At this point, you will find that the missing values come from only one row. Remove that row.
- Print the data type of each column.
- At this point, you will find that the data type is "object" for all columns because of the entries "-", "<0.1", and "<1". Replace the entries "<0.1" and "<1" with 0. The entries "-" are used as missing value markers. Replace these entries with "np.nan".
- Hence, convert all columns to numeric data type. Verify that the conversions are successful.
- Now, print the number of missing values of each column as a percentage of the total length of the column.

(4 marks)

7. From the DataFrame "a1", construct a table that shows the following information:
   - Under-five mortality rate (%)
   - Neonatal mortality rate (%)

   for the top 20 countries (lower is better). Sort in ascending order, first by under-five mortality rate, then by neonatal mortality rate. Save the table as "child_mortality" and display it. Note that the raw data is per 1000 live births, not per 100 live births. (2 marks)

[Total: 20 marks]

Assignment Output

The following items need to be submitted:

1. The Python script file as a .PY file or a .IPYNB file.

Marking Allocation

The allocation of marks are as given in the question.