ENSF 592 Spring 2021 – Final Project Report

Authors: Bhavyai Gupta, Brandon Attai
Date: 16th June, 2021
Course: ENSF 592

Summary

This program allows a user to request, select, or visualize various statistics surrounding data provided by United Nations to give insight into population and wealth trends for UN Regions, Sub-Regions, and Countries. To achieve this, a total of four Datasets are used as follows:

- UN Region, Sub-Region and Country
- Population Growth, Fertility, Life Expectancy and Mortality
- Population in the Capital City, Urban and Rural Areas
- GDP and GDP per Capita

The program makes use of four python source files to provide an interactive command line user interface. `launch.py` serves as the starting point of the program, as it controls the execution flow by using menu options.

As soon as the program is launched, it will initialize via a five-step process. These five steps are (1) importing the data into pandas' DataFrames, (2) merging DataFrames together, (3) adding computed columns to the merged DataFrame, (4) performing checks for missing values, and (4) export of the final dataset into the project directory. User is given real time updates on each step and its status.

Once user enters the program menu after the initialization, the user would have 7 options to select from. These are summarized in *Table 1*. For example, the user can use menu option number 4 to print the aggregate stats on GDP per capita (US dollars) grouped by either UN Regions or UN Sub-Regions.

Option Number	Option Description
1	Print the imported dataframes
2	Re-export the entire merged hierarchical dataset into Excel
3	Print aggregate stats for the entire dataset
4	Print aggregation stats grouped by UN Region/UN Sub-Region and available years
5	Print the list of countries that have higher GDP per capita than USA, and the year
6	Compare four different countries on various statistical data and plot graphs
0	Exit

Table 1

One thing to note is that some countries may not get accepted as a valid input by the program. Examples include Russia and Iran. This is because these countries do not have adequate data, which results in `NaN` in the merged dataset. `NaN` values are dropped during the initialization process.

The program makes use of object-oriented programming with the help of classes, to efficiently handle data analysis and calculations. Finally, exception handling is performed throughout the program to ensure the program does not terminate if the user enters an invalid input.

A matrix summarizing how the requirements are met is shown in Appendix 1 on page 3.

References

1. UN Region, Sub-Region and Country, Development Data Section of the Development Data and Outreach Branch within the Statistics Division of the Department of Economic and Social Affairs (UN DESA) of the UN Secretariat, June 2019. [Online]. Available: https://data.un.org/_Docs/SYB/CSV/SYB63_1_202105_Population,%20Surface%20Area%20and%20Density.csv

2. Population Growth, Fertility, Life Expectancy and Mortality, Development Data Section of the Development Data and Outreach Branch within the Statistics Division of the Department of Economic and Social Affairs (UN DESA) of the UN Secretariat, Aug. 2019. [Online]. Available: https://data.un.org/_Docs/SYB/CSV/SYB62_246_201907_Population%20Growth,%20Fertility%20and%20Mortality%20Indicators.csv

3. Population in the Capital City, Urban and Rural Areas, Development Data Section of the Development Data and Outreach Branch within the Statistics Division of the Department of Economic and Social Affairs (UN DESA) of the UN Secretariat, May 2018. [Online]. Available: https://data.un.org/_Docs/SYB/CSV/SYB61_253_Population%20Growth%20Rates%20in%20Urban%20areas%20and%20Capital%20cities.csv

4. GDP and GDP per Capita, Development Data Section of the Development Data and Outreach Branch within the Statistics Division of the Department of Economic and Social Affairs (UN DESA) of the UN Secretariat, Nov. 2020. [Online]. Available: https://data.un.org/_Docs/SYB/CSV/SYB63_230_202009_GDP%20and%20GDP%20Per%20Capita.csv

Appendix 1

Item	Stages	Objective	Notes
1	Stage 1: Dataset Selection	Several suggested datasets are included in the project repository. You may use the provide data or select datasets of your own choosing.	See folders "UN Population Datasets" and "CustomUNData" under the project directory
2		You must use at least three separate Excel sheets or files that can be related in some way.	Four separate Excel sheets are used
3		Your final combined dataset (see next stage) must have at least ten columns and 200 rows.	Before adding additional columns, dataset has 11 columns and 479 rows
4		You may edit the given datasets before you begin coding, but your program should not modify the Excel files directly.	Program does not modify the excel files in any way
5		You may not hard-code/copy-paste any information into your program except for the Excel column names.	No information hard-code or copy-paste within the program except the column names
6	Stage 2: DataFrame Creation	Import your chosen data into a Pandas DataFrames.	Import is done using `_import_data` method of the class DataAnalysis
7		You must use at least two merge/join operations and you must delete any duplicated columns/rows that result from the merge.	Merge is done in both the methods `_import_data` and `_merge_data` of the class DataAnalysis
8		You must create a hierarchical index of at least two levels (row or column).	A hierarchical index is created in `_merge_data` method (lines 192-193) of the class DataAnalysis
9		All data should be presented in the correctly sorted order, depending on the index.	The indexes are sorted in `_merge_data` method (line 196) of the class DataAnalysis
11		You may not use global variables. You must import the data within your main function.	No global variables are used in the program
12		Remember to check for null values or data mismatches.	Method `_check_null` checks and prints if any column of the dataset has null values. This is just a formality as null values are dropped using `dropna()`
13	Stage 3: User Entry	Your application must return useful information. Design an interface that allows users to search based on some sort of criteria or keywords.	User can select what information needs to be displayed using menu options, and keying various criteria as program goes along
14		The user must provide at least two pieces of information/selection (e.g. "school name" and "grade")	The program is completely user driven. For instance, menu option 4 requires user to input three different types of data to get to the the results
15		Give the user clear input instructions. If an invalid entry is given, use try/except statements to handle the error. Your program should not terminate.	Try/Except statements are used throughout the program to handle invalid user entries
16		You must not hard-code any data values (the data within your spreadsheets could be changed!).	Except column names, no other hard-coded values are used. The program should be able to deal if the amount of data contained in spreadsheet changes
17		Any output information must be clearly defined using printed headers.	Colorful headers and sub-headers are always printed for every output
18	Stage 4: Analysis and Calculations	You may choose what data trends to presents from your data. However, you must meet the following specifications.	-
19		Use the describe method to print aggregate stats for the entire dataset.	`describe()` is used in the method `print_aggregate_stats` (line 360) of the DataAnalysis class.
20		Add at least two columns to the combined dataset.	Three additional columns are added by the method `_additional_statistics` of the class DataAnalysis
21		Use an aggregation computation for a subset of the data.	Aggregation computations are done on the subset of data in the method `group_by_stats` (lines 451-452) of the class DataAnalysis
22		Use a masking operation.	Masking operations are used quite frequently. For example, methods `_import_data`, `_additional_statistics`, and `higher_gdp_than_usa` are using masking operations to provide their respective functionalities
23		Use the groupby operation at least once.	Groupby operation used in the method `group_by_stats` (lines 451-452) of the class DataAnalysis
24		Create and print a pivot table.	Pivot Table created by the method `pivot_plot` on line 561 and printed on line 566. `pivot_plot` is a method of the class DataAnalysis
25		Include at least two user-defined functions or a class that contains two methods.	Total number of classes: 4 Total number of methods in any class: 11 Total number of functions outside class: 3
27	Stage 5: Export and Matplotlib	Export your entire merged, hierarchical dataset to an Excel file in the working directory. Be sure to include the index and header values. The TAs will use this to verify the structure of your dataset.	Entire merged hierarchical dataset is exported using `export_dataset` method (line 301) of the class DataAnalysis
28		Use your data to create at least one plot using Matplotlib. Save the plot as a .png file and upload to the repository.	Plots are created using Matplotlib in the method `pivot_plot` method (line 570 onwards) of the class DataAnalysis. Plots.png file is available in the repository