Scientific Programming in Python Project

DUE DATE: Sunday December 20th, 20:00

Dataset Overview:

The National Health and Nutrition Examination Survey (NHANES) is a program of studies designed to assess the health and nutritional status of adults and children in the United States. The survey is unique in that it combines interviews and physical examinations. NHANES is a major program of the National Center for Health Statistics (NCHS). NCHS is part of the Centers for Disease Control and Prevention (CDC) and has the responsibility for producing vital and health statistics for the Nation.

The first dataset provided (‘*NhanesDemoAdapted.csv*’) has been adapted and changed from the demographic set. You should

1. SEQN – Integer: ID of person
2. Gender – String: ['Female', 'Male']
3. Age – Integer: age of person (0-80, 80 given for anyone aged 80 or over)
4. Ethnicity – String: ['Black', 'White', 'Asian', 'Mexican-American', 'Other Hispanic', 'Others']
5. US born – Integer: {1: Us born, 2: Other}
6. Education – Integer: Highest education level achieved for persons aged 20+ {1:<9th Grade, 2: 9th-11th grade, 3:HighSchool graduate, 4:Some college, 5: College graduate or above}
7. Marital Status – Integer: Persons aged 20+ {1:Married, 2:Widowed, 3:Divorced, 4:Separate, 5: Single, 6: Living with Partner}
8. HouseholdSize – Integer: Number of people in house {1-7, 7 given for houses with 7 or more people}
9. AgeUnder6 – Number of household members under the age of 6
10. Age6to18– Number of household members between the ages of 6 and 18
11. AgeOver60 – Number of household members over the age of 60
12. HouseholdIncome – Total household income in 1000s of dollars
13. IncomePovertyRatio – Float: Ratio of family income to poverty {0.0-5.0, 5 given for ratios of 5 or higher}

The second dataset (‘NhanesFoodAdapted.csv’) involves the diet of individuals. Again this has been adapted and changed from the original Nhanes dataset. The reduced dataset has multiple entries for each individual (as identified by the same SEQN as in the first dataset). Each line involves a different recorded meal for an individual and contains the nutritional information of the meal such as grams, calories, protein, etc.

1. SEQN – Integer: ID of person
2. dGRMS – Float: Gram weight of meal
3. dKCAL – Integer: Energy (kcal) of meal
4. dPROT – Float: Protein content (gm) of meal
5. dCARB – Float: Carbohydrate content (gm) of meal
6. dSUGR – Float: Total sugars content (gm)t of meal
7. dFIBE – Float: Dietary fibre content (gm) of meal
8. dTFAT – Float: Total fat content (gm) of meal
9. dSFAT – Float: Total saturated fatty acids (gm) of meal
10. dCHOL – Float: Cholestorol (mg) of meal
11. dVITC – Float: Vitamin C content (mg) of meal
12. dVITD – Float: Vitamin D content (mcg) of meal
13. dCALC – Float: Calcium content (mg) of meal
14. dCAFF – Float: Caffeine content (mg) of meal
15. dALCO – Float: Alcohol content (gm) of meal

You should read these datasets in as pandas dataframes. You should use the ***describe*** function to familiarise yourself with the contents of each dataframe.

The project will involve analysis of two cases. The first case involves just the first dataset. The second case involves merging the two datasets.

Project Specification.

The objective of this project is to produce an application that allows the user to explore some of the most interesting aspects of these two adapted Nhanes datasets. Please note that where possible you should use **Pandas** as a means of analysing the data. Where requested please incorporate visualisation as a method of illustrating your results. Please be aware that the dataset does contain some missing values.

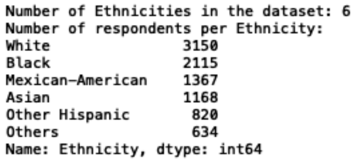
When you run your program it should display the following menu:

```
Please select one of the following options:

    1.  Household income per ethnicity
    2.  Marital status
    3.  Income and education level
    4.  Diet analysis
    5.  Exit
```

1. **Menu Option 1 – Household Income of ethnicities**

   In order to assess if there are societal disadvantages we investigate the relationship between the average HouseholdIncome attained by respondents and their ethnicity. The user should first be told how many ethnicities are represented, and a sorted list of number of respondents per ethnicity. The output should look something like:

   ```
   Number of Ethnicities in the dataset: 6
   Number of respondents per Ethnicity:
   White               3150
   Black               2115
   Mexican-American    1367
   Asian               1168
   Other Hispanic       820
   Others               634
   Name: Ethnicity, dtype: int64
   ```

   The average HouseholdIncome per ethnicity (only considering adult respondents) should then be conveyed using a **horizontal bar graph**.
   The report should contain one graph generated using the code and discussion of the results depicted in the graph.

   [15 Marks]

2. **Menu Option 2 – Marital status**

   Investigate the relationship between age and marriage for adult respondents only (aged 20 and over). You should first print out a sorted list indicating the number of respondents per marital status category.
   Using a **line graph** it should display the 1st, 2nd, and 3rd quartiles of age for each status type. The line graph should have three lines, one for each quartile.
   (Note the 1st quartile is also referred to as the 25th percentile, the 2nd as the median and the 3rd as the 75th percentile.)
   Your report should contain the line graph that is outputted and discussion of the results depicted in the graph.

   [15 Marks]

3. **Menu Option 3 – Education and income**

   In this case, we wish to assess the correlation between education and income. The user will be given the option of

   1. IncomePovertyRatio
   2. HouseholdIncome

   A **simple bar graph** comparing the average of the user option for the different education levels will be given. For example, if the user selects the second option then a bar graph containing the average income for each education level should be generated.
   Your report should contain a sample bar graph that is outputted and discussion of the results depicted in the graph (e.g. does higher level of education suggest a higher income?).

   [15 Marks]

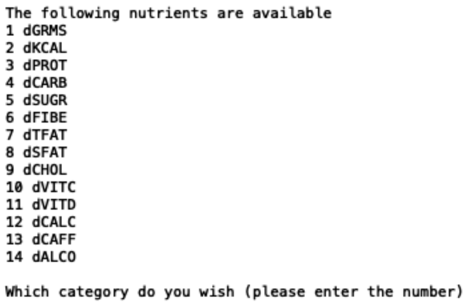
4. **Menu Option 4 – Diet**

   Your function should take as input two dataframes, the first containing the demographic data, the second containing the dietary data.
   You must first reduce the food dataset (using pandas ***groupby*** functionality) to the average food intake per meal per individual. This new dataframe (containing only one row per individual) will then be merged with the demographic dataframe via pandas, such that it **only** contains entries for individuals that were in the Food dataset.

Your code should then write this merged dataframe to a csv file called (‘*studentName_merged.csv’*). A sample file is included of what your output should like.

[15 Marks]

For this merged dataframe (if unsuccessful in creation, use the sample dataset), the user will be given the list of nutrients and asked to select a category.



```
The following nutrients are available
1 dGRMS
2 dKCAL
3 dPROT
4 dCARB
5 dSUGR
6 dFIBE
7 dTFAT
8 dSFAT
9 dCHOL
10 dVITC
11 dVITD
12 dCALC
13 dCAFF
14 dALCO

Which category do you wish (please enter the number)
```

Boxplots will then be generated for the categories of Gender, Ethnicity and Education. Scatter plot will be generated for comparing with HouseholdIncome, and comparing with Age.

You should provide some basic error checking in your code. You should make sure that the user cannot enter an invalid value for the category. If the user does select an invalid value, your code should output an error message and ask the user to re-enter a valid value. Your code should continue to do this until the user enters a valid value.

[25 Marks]

5. Menu Option 5 – Exit

If the user selects an option 1-5 then your program should display the associated output and will subsequently display the main menu again.
If the user selects option 5 the application should exit.

The **final 15 marks** will be awarded based on **general coding** (menu exiting/returning correctly, use of functions, coding style, variable-naming convention, commenting appropriately, clarity, handling missing data, etc).

Note: You will be submitting both a report (.pdf or .docx format) as well as a python file. It is very important to make sure you include a clear reference between the results presented in the report and the python file. I should be able to easily find the code that was used to generate the results presented in the report.

Academic Integrity

This is an individual assignment. The work you submit must be your own. In no way, shape or form should you submit work as if it were your own when some or all of it is not.

Given how much freedom there is in the assignment, everybody's work will be different. It will be obvious if there is collusion. All parties to collusion will be penalized.

Deliberate plagiarism: You must not plagiarise the programs, results, writings or other efforts of another student or any other third-party. Plagiarism will meet with severe penalties, which can include exclusion from the University.

Your report will be checked for signs of collusion, plagiarism, falsification and fabrication. You may be called to discuss your submission and implementation with me and this will inform the grading, any penalties and any disciplinary actions.

You may, of course, ask me questions. I may share questions and answers with the class, if I feel they are general matters, for example, of clarification. But I will also discuss with you questions that relate to your own Python, your experiments and your reading and, in the interest of giving you proper credit for your endeavours, these will not be shared with the class.

A declaration must be included in your submission to state that the work included is all your own work.

Submission Instructions:

1. Submit your report (.pdf format) on Canvas (via the *submit pdf for turnitin* link in the Assignments unit) before **20:00 on Sunday Dec 20th**. Submit your solution python file (.py file) via the *submit code for project* link in the Assignments unit.
   As per CIT regulations, submitting within 7 days of the deadline will result in a 10% penalty. However, as per CIT regulations, submitting **between 7 and 14 days late will result in a 20% penalty**, and later than 14 days after the due date will result in a 100% penalty applied.
   Your file names should be *Surname_StudentNo*_Report.pdf and *Surname_StudentNo*_Project.py.

2. Once you have submitted your files you should verify that you have correctly uploaded them. It is your responsibility to make sure you upload the correct files.

3. Please make sure you fully comment your code.

4. Please also put your student name and number as comments at the top of your python file.