Social Data Sample

Due: *11:59 pm Thursday December 12, 2019 (Late penalty waived until Sunday December 15)*
Worth: *15% of your final grade*
Submit to: *MarkUS (markus.teach.cs.toronto.edu/inf1340-2019-09)*
Work: *In pairs using the pair programming paradigm (en.wikipedia.org/wiki/Pair_programming)*

Introduction

Statistics Canada, one of the most advanced and transparent national statistics agencies in the world, conducts a variety of surveys. We will focus on a small (1500 row / 20 column) sample of **The Canadian Community Health Survey (CCHS)**, including information on height and weight, smoking, alcohol consumption, general health, socio-demographics, income, and labour force characteristics of the population.

Starter Files

- `CCHSX.csv`: 1500 data rows (and 1 header row) with a subset of the columns from the CCHS
- `cchs.py`: Starter code and testing code for the Part 1 constants and Part 2 functions
- `cchs.ipynb`: Notebook with the same starter code as `cchs.py` that you can use for Part 3
- csv: an empty directory where all your generated `.csv` files should go.

Part 1: Preparation

- Consult **Appendix A** for data measurement scales
- Consult **Appendix B** for the CCHS data columns, the original column names, and the readable names we will use in the assignment.
- Check the `DATA_COLUMNS` tuple in `cchs.py` to see how the original column names in the dataset will be changed to readable column names.
- Fill out the data types in the `DATA_COLUMNS` constant tuple as follows:
    a. **Nominal** columns should be set to **`'U2'`**, a two-digit unicode string
    b. **Ordinal** columns should be set to Python **`int`**
    c. **Interval** columns should be set to Python **`float`**
    d. **Ratio** columns should be set to Python **`float`**
- Fill out the tuples **`ORDINAL`**, **`INTERVAL`**, and **`RATIO`** with column names based on your understanding of the data. **`NOMINAL`** has been filled out for you.

Part 2: Functions

All functions in Part 2 share a similar pattern:

- They accept the **complete array** and a **column name** as the first two arguments
- They work on the array in place and **return nothing**
- The functions that write to files accept **open TextIO file streams** like in Assignment 2, meaning you must open a file for writing before calling one
- You should **NOT** be running for loops to change array values.
  Use array indexing, slicing, and/or Boolean masks instead.

Cleaning functions

`replace_nominal_codes` changes the original integer category codes in nominal scale columns (that we've stored as strings) with non-numerical string codes. This function can be completed in **3 lines**.

`replace_missing_with_nan` replaces the arbitrary numbers in the data that denote that the data is missing or not applicable (e.g., 996) with a value that indicates that these values should not be used in calculations. This is done so that missing data codes are not mistakenly used in, e.g., calculations of mean. Use `np.nan` as the value of Not-A-Number values. This function can be completed in **3 lines**.

File writing functions

`write_categorical_csv` writes the entirety of a CSV file called `[readable_column_name].csv` in subdirectory `csv/` containing category frequency data for a nominal or ordinal data column. This function can be completed in **4 lines**. It should use the `np.unique` function to get all unique category values and the counts associated with each one.
The CSV file's format is as follows:

- **Header line:** readable column name (e.g., `biosex`, not `dhh_sex`), then string `"count"`
- **Other lines:** category in the first column and number of entries in the second. The order of categories is not important.

`write_column_summary_csv` writes a single line to an open CSV file called `'csv/summary.csv'` containing, in order:

- The readable column name (e.g., `alcoweek`, not `alc_2`)
- The median of values in the column for all data (use the `np.nanmedian` function)
- The mean of values in the column for interval and ratio data (use the `np.nanmean` function)
- The standard deviation of values in the column for interval and ratio data (use the `np.nanstd` function)

Every line this function writes should contain 3 commas and end with a newline character.
This function can be completed in **10 lines**.

Consult the numpy documentation for proper usage of its functions:
https://docs.scipy.org/doc/numpy/reference/index.html

Part 3: Plot and Play

Migrate your code into a Jupyter notebook and prepare a short data exploration notebook. The purpose of this part is for you to show your work and your thinking through the different stages of the data analysis process.

- Remember that each row in the data set describes a single respondent in the sample. You can use this information to slice and filter the data along different axes.
- Focus on 2-3 columns that you find interesting
- Think about the real-world properties they represent and type in some questions you would like to answer in your notebook before you start working.

Some things to try:

- Plot a variable (e.g., as a histogram) or a relationship between variables (e.g., as a scatter or box plot) using **`matplotlib`** and see what they tell you
- Extract a subset of the sample based on a particular characteristic (e.g., all smokers) and compare it to the rest of the sample

Overall, this part of A3 is a chance to show your thinking process and use a data set you've processed and understand to do some exploration.

Requirements:

- Do something with the data
- Use at least 3 variables / columns
- Use numpy and/or matplotlib and/or anything else you stumble across
- Create a self-contained story in your Jupyter notebook regarding your thinking and data exploration.

Using numpy with Python

Because **numpy** is not included in the base distribution of Python, you will likely get an error trying to **import numpy** directly into Wing. However, Anaconda does include **numpy** in its installation.

Therefore, you have three options for this assignment.

- Change your Wing settings to use the Anaconda version of Python as its executable: https://docs.anaconda.com/anaconda/user-guide/tasks/integration/wing/
- Paste the starter code from `cchs.py` into a code cell in Jupyter and write/test all your code there. Then, to submit to MarkUs, paste it back into Wing to save as `cchs.py`
- Use instructions here: https://scipy.org/install.html to install SciPy (including **numpy**) on your computer.

What to submit

- **cchs.py** with completed functions and constants
- **cchs.ipynb** with your own data explorations for Part 3

There is no need to submit any of your generated csv files

Component weight

Component	%	Criteria
Part 1&2 Correctness	60	Your functions and constants pass automated tests of correctness
Coding style and variable naming conventions	20	Code is readable and robust, if statements used appropriately, minimal code repetition, informative variable names follow pothole_case
Part 3: Exploration	20	An interesting question has been asked about the dataset. Data has been gathered using additional Python code, and the results have been described well.

How to submit

1. Find a **partner** and follow the pair programming paradigm.
2. Ensure you can log in to **MarkUs** at https://markus.teach.cs.toronto.edu/inf1340-2019-09/
3. Go to Assignment 3 and under "Group Information" invite your **partner** or join their group.
4. You may **submit** as many times as you like before the due date.

Appendix A: Identifying Data Measurement Scales

Nominal Data Scale

Each data point belongs to a discrete category. Category membership can be **counted** and expressed as a **proportion** of the total, but data points cannot be compared, subtracted, or ordered.

E.g., *Where were you born?*

France	**154**	**45.3%**
Germany	**77**	**22.6%**
Spain	**109**	**32.0%**

Ordinal Data Scale

Data points are categorical, but they can be ordered. Membership can be expressed as a **proportion** of the total, and **median and mode** descriptions are meaningful, but data points cannot be subtracted or divided.

E.g., *How satisfied were you with your meal?*

Very dissatisfied	**23**	**38.3%**
Dissatisfied	**18**	**30.0%**
Neither	**5**	**8.3%**
Satisfied	**11**	**18.3%**
Very satisfied	**3**	**5.0%**

Interval Data Scale

Data points may be discrete or continuous, but they are ordered, and the units of the scale are evenly spaced. Membership does not make sense, but **mean, median, and mode** descriptions can be calculated and data point differences are meaningful.

E.g., *What is the average daily temperature in April in your city?*

London, UK:	15 C
Los Angeles, USA:	23 C
Cairo, Egypt:	30 C

The **difference** between London and LA is similar to the difference between LA and Cairo.

Ratio Data Scale

Ratio scale data os ordered, continuous, evenly spaced, and has a meaningful zero. **Mean, median, and mode** apply, as do **differences** and **ratios** between values.

E.g., *Adventurer height:*

Grog:	262 cm
Keyleth:	183 cm
Pike:	99 cm

Grog is **1.5 times** taller than Keyleth and **3 times** taller than Pike. There is such a thing as a 0 cm height.

Appendix B: CCHS Columns and Codes

`original_column_name -> readable_column_name`

alc_2 -> alcofreq During the past 12 months, how often did you drink alcoholic beverages?	1. No drinks 2. < Once a month 3. Once a month 4. 2 to 3 times a month 5. Once a week 6. 2 to 3 times a week 7. 4 to 6 times a week 8. Every day
alwdwky -> alcoweek Average number of drinks consumed per day in the past week?	996. Did not drink in the last 12 months
dhh_sex -> biosex Sex.	1. Male 2. Female
dhhgage -> agegroup Age.	3. 18 to 19 years 4. 20 to 24 years 5. 25 to 29 years 6. 30 to 34 years 7. 35 to 39 years 8. 40 to 44 years 9. 45 to 49 years 10. 50 to 54 years 11. 55 to 59 years 12. 60 to 64 years 13. 65 to 69 years 14. 70 to 74 years 15. 75 to 79 years 16. 80 years or more
edudr04 -> education Highest level of education	1. < Secondary school 2. Secondary school graduate 3. Some post-secondary education 4. Post-secondary certificate
fvcdtot -> fruitvegtot Total number of times per day eats fruits and vegetables	

gen_07 -> stressgen Thinking about the amount of stress in your life, would you say that most days are stressful?	1. Not at all 2. Not very 3. A bit 4. Quite a bit 5. Extremely
gen_09 -> stresswork Would you say that most days at work are stressful?	0. Not in labour force 1. Not at all 2. Not very 3. A bit 4. Quite a bit 5. Extremely
gendhdi -> healthphys Perceived health status	0. Poor 1. Fair 2. Good 3. Very good 4. Excellent
gendmhi -> healthment Perceived mental health status	0. Poor 1. Fair 2. Good 3. Very good 4. Excellent
gengswl -> satisfaction Satisfaction with life in general.	1. Very satisfied 2. Satisfied 3. Neither 4. Dissatisfied 5. Very dissatisfied
geogprv -> province Province of residence	10 Newfoundland & Labrador 11 Prince Edward Island 12 Nova Scotia 13 New Brunswick 24 Quebec 35 Ontario 46 Manitoba 47 Saskatchewan 48 Alberta

	59 British Columbia 60 Yukon / NWT / Nunavut
hcu_1aa -> hasdoctor Do you have a regular medical doctor?	1. Yes 2. No
hwtgbmi -> bmi Body Mass Index (BMI)	
hwtghtm -> height Height (in m)	
hwtgwtk -> weight Weight (in kg)	
incghh -> incomegroup Household income	1. < $20 000 2. $20 000 - $39,999 3. $40 000 - $59,999 4. $60 000 - $79,999 5. $80 000 or more
lbsghpw -> workhoursperweek Total usual hours worked per week in current job(s)	996. Not in labour force
sdcdfols -> firstlanguage First official language spoken	1. English 2. French 3. English & French 4. Neither
smkdsty -> yrsmokeddaily Number of years smoked daily	996. Not a daily smoker