Homework 0: Skills Check

Due Sept. 7th at 10pm

1 Class Set Up

1.1 CourseIntro

Introduce yourself by creating a CS333 CourseIntros[1] introduction. Make a brief video in which you pronounce your whole name.

1.2 Gradescope

We will handle homework submissions using Gradescope. Add yourself to the course Gradescope:

- Go to Gradescope and create an account, if you don't have one already.
- Enroll in the course by using the course entry code: WV77YR

2 Python Skills Check

The goal of this assignment is to give you a way of assessing whether your skills are strong enough for this course. CS 333 is a programming-intensive course that involves a lot of data processing. In order to be successful in this course, you should be confident about the following:

- Passing command-line arguments using **`sys.argv`**
- Reading and writing from files
- Reading and writing JSON and CSV data
- Working with dictionaries and lists
- Manipulating strings using **`split`**, **`strip`**, and **`join`**
- Using list comprehensions to filter and process lists
- Printing with f-strings
- Error handling with **`assert`** and **`try/except`**
- Installing Python packages using **`pip`**

Documentation for each of these topics is linked above— if you don't feel confident with any of these, you should spend some time reviewing them before starting the assignment.

2.1 Data Preprocessing

For this assignment, you will work with the data in **`animals.csv`**. It contains information about pets in Austin, Texas. Place your code for this assignment in **`skills_check.py`**. You should

[1] CourseIntros was created by Wellesley faculty alum Susan Buck.

start with a **`main`** function, which should call each of the functions that you will write below.

2.1.1 Reading CSV Data

This class involves a lot of data processing. Usually, we will work with data in one of two formats: JSON or CSV. Our data for this assignment is in CSV format.

Write a function called **`read_data`**, which takes the name of a file as its only argument. It should use a csv DictReader to read in the data from the file as a list of dictionaries, and return the list.

Call this function from **`main`** with the "animals.csv" as an argument, and store the result in a variable for later use.

2.1.2 Data Cleaning

There are three ways that our pet data could be more usefully arranged:

1. Some of the pet names have asterisks in front of them.
2. DateTime contains both the date that the animal was registered, and the time it was registered.
3. Some of the Location entries include a street, but others only have the town and state.

Write a function called **`clean_data`**. Your function should take in the list of dictionaries produced by **`read_data`** and apply the following changes:

1. Strip any preceding asterisks and whitespace from Name.
2. Split DateTime into two separate entries: Date, and Time.
3. Remove DateTime.
4. Standardize Location so that all entries are in the Town (State) format.

Your function should return the cleaned dataset. You should call this function from main, and store the result in a variable for later use.

2.2 Summarizing Data

2.2.1 Summarizing Animal Types

Write a function called **`summarize_types`**. This function should take the cleaned list of dictionaries and print out the frequency counts for each animal type in Animal Type.

Your function should produce a printed summary like the one below:

```
~~~~~~~~~~~~~~~~~~~~~~~~
N Dog:            50349
N Cat:            21993
N Other:          650
N Bird:           107
N Livestock:      2
~~~~~~~~~~~~~~~~~~~~~~~~
```

Hint: using f-strings makes this much simpler.

2.2.2 Most Popular Names

Write a function called **`top_names`**. It should take the cleaned list of dictionaries and a number k as arguments. It should calculate and print out the k most popular names along with their counts. For instance, when called with k=7, it should print:

```
~~~~~~~~~~~~~~~~~~~~~~~
1. Max:                534
2. Bella:              479
3. Luna:               445
4. Daisy:              381
5. Charlie:            375
6. Rocky:              347
7. Lucy:               330
~~~~~~~~~~~~~~~~~~~~~~~
```

2.2.3 Filtering Data

Since there are twice as many dogs as cats in our dataset, it is likely that the most popular names are skewed towards dog names. To find the most popular names for other species, it will be useful to have a general-purpose filtering function.

Write a function called **`filter_data`**. It should take the cleaned dataset as one argument, and a predicate as its second argument (a function that takes a single dictionary and returns True or False). It should return a list of only the dictionaries for which the predicate is true.

2.2.4 Most Popular Cat Names

Now use your filtering function to help you write a function that prints out a summary of the most popular cat names. Call this **`top_cat_names`**.

If you make use of an anonymous function, its body only needs to contain 1 line.

2.2.5 Color Counts

Write a function to summarize how many of animals of each color are in the data. Call this **`count_colors`**; it should take two parameters, data and k.

Print *inclusive* counts: if more than one color is listed for a given animal, include it for each of the colors. You can approach this in a number of ways: hardcoding a list of colors, figuring out what characters to split on to separate colors, or any other strategy that occurs to you.

Your function should print out a summary of the color counts as below (shown with k=7):

```
~~~~~~~~~~~~~~~~~~~~~~~~~
1. White:              35685
2. Black:              23621
3. Brown:              18692
4. Tan:                10345
5. Tabby:              9977
```

```
6. Blue:            5994
7. Brindle:         3409
~~~~~~~~~~~~~~~~~~~~~~~~~
```

2.3 Estimating Probabilities

Next, we will use the data to estimate some probabilities.

2.3.1 Blue Animals

Blue is a surprisingly popular animal color. Which animals are blue?

To answer this question, we could print out another table of counts, filtering our data first to animals that are blue, and then calling **`summarize_types`**. To interpret this data, though, we have to keep in mind how many animals of each type there are— even a small number of blue birds could indicate that this is a popular color for birds, since there are so few birds to start with.

To better understand the relationship between color and animal type, we will instead calculate the *probability* of being blue given an animal type: p(blue|cat), p(blue|bird), etc.

Write a function called **`calculate_conditional_probability`**. Like **`filter_data`**, this will be a higher order function: we will pass in two predicates, A and B.

Your function should calculate and return p(A|B), the probability of predicate A holding if predicate B holds. For instance, to answer our question about whether birds are likely to be blue, predicate A would be a function that returns True if the blue is one of the animal's colors, and predicate B would be a function that returns True if the animal is a bird.

2.3.2 Color Given Animal

We now have a way to calculate the probability of being blue for a single type of animal. Write a function called **`summarize_blue_given_animal`**. It should use your previous function to calculate the probability of being blue for each type of animal. It should print a table of the probabilities in the format that we have been using; round the probabilities to 3 decimal points.

2.3.3 Cat Given Color

You should observe that cats are the most likely animal to be blue, though blue cats are still not very common. If you know an animal is a cat, you now know how likely it is to be blue. But what if you know its color, but not its type?

Write a function called **`summarize_cat_given_color`**. For each of the top 10 colors in our dataset, it should calculate the probability that an animal of that color is a cat. It should print out the results in the format we have been using; round the probabilities to 3 decimal points.

Hint: you may want to modify your **`count_colors`** function to return a list of top colors; your **`summarize_cat_given_color`** can take this as an argument.

Which color is the best predictor that an animal is a cat?

2.4 Wrapping Up

2.4.1 Export Cleaned Data

Write a function that exports the cleaned data to a new CSV file. Your function should be called **`export_data`** and it should take two arguments: the data and the name of the destination file. It should use **`csv.DictWriter`** to write the data. Make sure to write the column names as the first line using **`writeheader`**.

Call this from main with "clean_animals.csv" as the filename argument.

3 Math Skills Check

This part of the assignment is also an opportunity to see whether you are prepared to succeed in this class. As in the Python skills check, you may consult external resources to refresh your knowledge as you work on these problems. But, by the time you submit your answers, you should feel confident that you understand and could reproduce them.

3.1 Axioms of Probability

Assume there are three variables A and B and C, which may or may not be independent. Which of the following statements are always true?

1. $P(A|B) = P(B|A)$
2. $P(A|B) = P(A)$
3. $P(A, B) = P(A|B)P(B)$
4. $P(A, B, C) = P(A)P(B)P(C)$
5. $P(A, B) = P(A)P(B)$

3.2 Axioms of Probability, Again

Now assume that A and B and C are independent of each other. Which of the following statements are always true?

1. $P(A|B) = P(B|A)$
2. $P(A|B) = P(A)$
3. $P(A, B) = P(A|B)P(B)$
4. $P(A, B, C) = P(A)P(B)P(C)$
5. $P(A, B) = P(A)P(B)$

3.3 Vectors and Matrices

Linear algebra is not a prerequisite for CS 333, but you will need some basic familiarity with vectors and matrices in the second half of this course.

Assume you have the following two matrices and vectors:

$$A = \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 1 & 3 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 1 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 4 & 5 & 6 \end{bmatrix}$$

1. Can you multiply A by B? If so, what would be the **dimensions of the resulting matrix**?
2. Can you multiply B by A? If so, what would be the **dimensions of the resulting matrix**?
3. Can you multiply C by A? If so, what would be the **result**?
4. What is the result of taking the dot product of C and D?

4 Submission

When you're finished, you should submit your Python file and your answers to the questions (as a PDF or plain text file) to Gradescope. **Make sure your Python file is called skills_check.py.** The name of your answer file does not matter.