Dictionaries Challenge 24: Frequency Analysis App

Description:
You are responsible for writing a program that will analyse the letter distribution of a given text. Your program will take any text, remove all non-alpha characters, count the frequency of each letter within the text, calculate the percentage of occurrence for each letter, and create a list of letters ordered from highest occurrence to lowest occurrence. Your program will perform these operations for two different bodies of text.

Step By Step Guide:

- Print a welcome message.
- Create a list called non_letters.
    - This will hold all non letter characters that may appear in a phrase the user enters.
    - Make sure to include all punctuation marks, numbers, a blank space, the newline character, and the tab character.

- Get user input for a phrase to be analyzed.
    - Store this phrase in a variable called key_phrase_1.
    - Take proper precautions to standardize the user input such that it will always be lower case.

- Remove all non letters from the phrase entered by the user.
- To do this use a for loop to loop through your list of non letters.
    - Each iteration, use the .replace() method for strings to replace the current non_letter with ''; or nothing.
    - This will remove any non letters from your phrase and replace them with an empty character.
    - I would suggest looking up how to use this new method.
    - After this step is done a previously entered string such as "Hello! How are you doing today? 32 years old I'm today." would appear as, "hellohowareyoudoingtodayyearsoldimtoday"

- Store the total length of the new "cleaned up" phrase entered by the user in a variable called total_occurrences.

- Create a Counter object called letter_count.
    - A Counter is a collection where elements are stored as dictionary keys and their counts are stored as dictionary values. Counts are allowed to be any integer value including zero or negative counts.
    - For our purpose, each letter in our phrase will be a key to this dictionary and the number of occurrences will be the value.
    - Counters are outside the scope of basic Python so we will need to import an extra library of code.

    - Type “from collections import Counter” as your first line of code in your program.
    - This will import the Counter dictionary subclass.

- To create the Counter type the following:
    - letter_count = Counter(key_phrase_1)
- letter_count will be a dictionary that has every letter as a key and the associated number of occurrences of that letter as the value.

- Print a frequency analysis of the letters used in the phrase entered by the user.
    - This should show the letter, the total number of occurrences, and the percentage that it occurred in the phrase.
    - Sort the results such that they are in alphabetical order.
    - Round the percentage to two decimal places.

- After you display the frequency analysis for key_phrase_1, order the letters from highest to lowest occurrence.
    - In order to do this create a variable called ordered_letter_count.
    - Set the value of ordered_letter_count equal to your Counter you created from part 1, letter_count and use the .most_common() method.
    - I would highly suggest looking up "python counter most common" and see how this method works.
    - I would also suggest experimenting with this method and see what kind of data type it returns and what each piece of information represents.

- Create a blank list called key_phrase_1_ordered_letters.
- Your goal is to append all of the letters from key_phrase_1 to this list in order from most occurrences to least occurrences.
- Once you have done this, print all the letters in key_phrase_1_ordered_letters on one line.
    - If you can't remember how to do this look into the end= argument of the print function.

- Expand your code to allow the user to enter in a second message after the first.
- Display all the same statistics for the second message.
- Choose your variable names wisely!
- Don't overwrite any information from the first message when you run the code a second time.

- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output

```
Welcome to the Frequency Analysis App
```

```
Enter a word or phrase to count the occurrence of each letter:  Hello!  How are you doing today?
32 years old I'm today.

Here is the frequency analysis from key phrase 1:

        Letter          Occurrence      Percentage
        a               4               10.26%
        d               4               10.26%
        e               3               7.69%
        g               1               2.56%
        h               2               5.13%
        i               2               5.13%
        l               3               7.69%
        m               1               2.56%
        n               1               2.56%
        o               7               17.95%
        r               2               5.13%
        s               1               2.56%
        t               2               5.13%
        u               1               2.56%
        w               1               2.56%
        y               4               10.26%

Letters ordered from highest occurrence to lowest:
oaydelhritwungsm

Enter a word or phrase to count the occurrence of each letter:  This is prety interesting.  I feel
like i'm learning something new!

Here is the frequency analysis from key phrase 2:

        Letter          Occurrence      Percentage
        a               1               1.89%
        e               9               16.98%
        f               1               1.89%
        g               3               5.66%
        h               2               3.77%
        i               9               16.98%
        k               1               1.89%
        l               3               5.66%
        m               2               3.77%
        n               6               11.32%
        o               1               1.89%
        p               1               1.89%
```

r	3	5.66%
s	4	7.55%
t	5	9.43%
w	1	1.89%
y	1	1.89%

Letters ordered from highest occurrence to lowest:
ientsrglhmpyfkaow