New York University
School of Continuing and Professional Studies
Division of Programs in Information Technology

Introduction to Python
Homework, Session 4

4.1 Calculate the average Mkt-RF value for a given year using a compiled list. Looping through the F-F_Research_Data_Factors_daily.txt file (this is the file with headers and a footer), prepare a script that asks the user for a 4-digit year. Loop through the file and build a new list of values listed for that factor and that year. Once the list is built:

a. report the count and average using len() and sum()
b. report the maximum and minimum value with max() and min()
c. extra credit: report the median value by using sorted() and list slices

F-F_Research_Data_Factors_daily.txt is a more complete file of the Fama French data. The start of the file looks like this:

```
This file was created by CMPT_ME_BEME_RETS_DAILY using the 201211 ...
       The Tbill return is the simple daily rate that, over the number of ...
       in the month, compounds to 1-month TBill rate from Ibbotson and ...

                Mkt-RF     SMB     HML      RF
       19260701    0.09   -0.22   -0.30   0.009
       19260702    0.44   -0.35   -0.08   0.009
       19260706    0.17    0.26   -0.37   0.009
```

The end of the file looks like this:

```
       20121128    0.83    0.01   -0.26   0.000
       20121129    0.54    0.68   -0.05   0.000
       20121130    0.02   -0.15    0.35   0.000

       Copyright 2012 Kenneth R. French
```

The new wrinkle in reading the file will be to avoid the first 5 lines of the file which are the headers, as well as the last 2 lines of the file, which are the file's footer.

The other major difference in this assignment from last week's FF assignment is the use of a list to get an average value, instead of using a counter integer and sum float.

Important: the point of this exercise is that by compiling a list of values we can then analyze the list (with len() and sum()) to find out more about the values. So, there is no need to keep an independent count or sum in an integer and float. Instead, use len() and sum() on the compiled list. Do not keep an integer count or float sum, as we did in last week's solution. This time we'll use the list of values directly.

Expected output (multiple runs, including error checking):

```
please enter a 4-digit year:  nineteen_twenty
ERROR:  arg must be 4-digit year

please enter a 4-digit year:  1972
1972 (Mkt-RF):  251 values, max 1.38, min -1.45, avg 0.0486055776892

please enter a 4-digit year:  2001
2001 (Mkt-RF):  248 values, max 5.36, min -5.03, avg -0.0529838709677

please enter a 4-digit year:  1900
no values found
```

After reading the selected data, report the year, # of days and average Mkt-RF value. Important: if the year does not appear in the file, the program should still loop through the entire file: it will of course retrieve no values, and the list intended to hold these values will be empty. Test the presence of values in the values list using a boolean test:

```
if not year_data:
    print('no values found')
```

The median value is the "middle" value, i.e. the one that would appear in the middle of a sorted list. In an odd-numbered list of values (i.e., 3 values, 5 values, etc.), the median is the one in the middle (you can use list slicing in conjunction with len() to determine this value). In an even-numbered list of values, the median is halfway between the two middle-most values.

```
x = [1, 3, 5, 6, 7, 12.5, 15]    # median is 6 (middle value)
y = [1, 3, 5, 10, 12.5, 15]      # median is 7.5 (halfway between 5 and 10)
```

In order to acquire the median, you must know about the sorted() function. sorted() takes any sequence and returns the list sorted:

```
us = [1, 0.9, 1.3, 0.03, 11, 5]
s = sorted(us)
print(s)                          # [0.03, 0.9, 1, 1.3, 5, 11]
```

Note: be sure that your values are numbers and not strings!

For more extra credit, allow the user to additionally select a factor (Mkt-RF, SMB, HML, RF) and have the program compile a list of the values for the selected factor. The program should display the factors to choose and verify that the user's choice is correct.

Expected output (multiple runs, including error checking):

```
please enter a 4-digit year:  1972
available factors:  ['Mkt-RF', 'SMB', 'HML', 'RF']
please enter a factor:  Mkt-RF
1972 (Mkt-RF):  251 values, avg 0.0486055776892

please enter a 4-digit year:  2001
available factors:  ['Mkt-RF', 'SMB', 'HML', 'RF']
please enter a factor:  XXL
Sorry, that factor does not exist.
please enter a 4-digit year:  2001
available factors:  ['Mkt-RF', 'SMB', 'HML', 'RF']
please enter a factor:  SMB
2001 (SMB):  248 values, avg [your calculated value here]

please enter a 4-digit year:  1900
available factors:  ['Mkt-RF', 'SMB', 'HML', 'RF']
please enter a factor:  Mkt-RF
no values found
```

4.2 Spell checker: loading spelling words into a set from the words.txt file (which contains a dictionary of 25,225 words, one per line) and then reading and splitting out the words from file sawyer.txt, print all misspelled words (i.e., words not found in the dictionary set). Make sure to rstrip() all punctuation (comma, semicolon, period) and lowercase each word you handle (whether a spelling word or a "check" word).

The spelling words should be loaded into a set, but the sawyer.txt words should not be loaded into a structure -- they should simply be looped over from data in the file.

Attempt to keep track of the line of the file by splitting each line individually and using a loop within a loop (i.e., not by using read().split() the way we did it in the exercises.

Expected Output:

```
25185 words in spelling words

misspelled word:  russling
misspelled word:  unconshiously
misspelled word:  interlarded
misspelled word:  minst
misspelled word:  coattails
misspelled word:  Hhe
misspelled word:  sentense
misspelled word:  akt
```

Side note on this word list: many Unix distributions come with a word list for spell check purposes. I have doctored this list for the purposes of this assignment, adding a number of plurals, etc. - to manage the results and simplify the assignment.

For extra credit, change the script so that it looks at the file one line at a time, and keeps track of the line number so it can report the line number where the misspelling is located.

Expected Output:

```
25225 words in spelling words

misspelled word on line 1:  russling
misspelled word on line 4:  unconshiously
misspelled word on line 6:  interlarded
misspelled word on line 8:  minst
misspelled word on line 14:  coattails
misspelled word on line 16:  hhe
misspelled word on line 18:  sentense
misspelled word on line 20:  akt
```