Exercise Set 5

Introduction

This exercise set focuses on the **argparse** module of the standard library and Numpy **ndarray** objects. There are some prerequisites for this exercise:

- Install **numpy**
- Download the support files for this exercise set:
  - ex_5_2-data.csv (https://cbu.instructure.com/courses/9073/files/1192773?wrap=1) (https://cbu.instructure.com/courses/9073/files/1192773/download?download_frd=1)
  - ex_5_4-data.csv (https://cbu.instructure.com/courses/9073/files/1192759?wrap=1) (https://cbu.instructure.com/courses/9073/files/1192759/download?download_frd=1)
  - ex_5_2.py (https://cbu.instructure.com/courses/9073/files/1192761?wrap=1) (https://cbu.instructure.com/courses/9073/files/1192761/download?download_frd=1) (starter file)

ex_5_0.py

In a module named ex_5_0.py, implement a function `line_count(infile)` that meets the following requirements:

- takes an input filename `infile`
- opens the file
- counts the number of lines present in the file. *Hint: use readlines()*
- prints the number of lines in the file to standard output (the console/screen)

This module will be used in the next exercise. Note that for this exercise, `infile` is guaranteed to exist.

Optional but helpful: Test your function with simple files of your creation. You might also consider adding an entry point for your function

ex_5_1.py

In this exercise you will implement a command-line interface for the `line_count` function that you implemented in `ex_5_0.py`. Your module named `ex_5_1.py` will consist only of an entry point that meets the following requirements:

- use the appropriate `if` statement and conditional expression to assure that module code is only executed if the module is run from the command line. Note that it would technically execute if `Run -> Run Module` is selected in IDLE but this exercise is command-line focused.
- instantiate an `argparse.ArgumentParser` object

- configure the ArgumentParser object with the following:
  - a description for the program
  - a positional argument `infile`
- parse the arguments
- call `ex_5_0.line_count` with the infile argument.

To keep the problem simple, the filename that is passed to your program is guaranteed to exist. Your entry point only needs to parse arguments, import and call `line_count` with the appropriate argument.

ex_5_2.py

This exercise introduces two numpy functions: `numpy.savetxt` and `numpy.loadtxt`. In this exercise you will begin with a starter module entitled `ex_5_2.py` which includes code to read and write NumPy array data to a file.

Complete `ex_5_2.py` by finishing the `TODO` items included in the comments. These include:

- modify the input data so that it has a mean of 0.
- modify the zero mean data so that it has a standard deviation of 1.
- make sure to save the processed data to a variable called `processed` so that the `np.savetxt` call succeeds.

For more information about loading and saving data with NumPy, see the official documentation notes on **savetxt (https://numpy.org/doc/stable/reference/generated/numpy.savetxt.html?highlight=savetxt#numpy.savetxt)** and **loadtxt (https://numpy.org/doc/stable/reference/generated/numpy.loadtxt.html?highlight=loadtxt#numpy.loadtxt)**.

ex_5_3.py

In a new module caled `ex_5_3.py` re-create the code from `ex_5_2.py` and implement a command-line interace for it by:

- creating an `argparse.ArgumentParser` object
- configure the ArgumentParser object with the following:
  - a description
  - a positional argument `infile` which should be the input filename for the data file that needs to be processed.
  - a positional argument `outfile` which accepts the output filename.

Test your program with the input data from `ex_5_2-data.csv`.

ex_5_4.py

Create a module named `ex_5_4.py` that uses `numpy.loadtxt` to load an array from the file `ex_5_4-data.csv`. Your program should implement the following:

- load the 1000 element `ndarray` from `ex_5_4-data.csv`
- set any negative elements of the array to 0
- write the processed array to a file named `ex_5_4-processed.csv` using `numpy.savetxt`.

To keep this simple, you may assume that the file `ex_5_4-data.csv` will be available to your program.

NOTE: no command-line interface is required for this exercise.