HW3

Due Mar 29 by 11:59pm **Points** 25

Instructions: **hw3.pdf (https://canvas.tufts.edu/courses/44718/files/5779650?wrap=1)** ⬇ **(https://canvas.tufts.edu/courses/44718/files/5779650/download?download_frd=1)**

Code: **assignment.zip (https://canvas.tufts.edu/courses/44718/files/5779991?wrap=1)** ⬇ **(https://canvas.tufts.edu/courses/44718/files/5779991/download?download_frd=1)**

The zip folder assignment contains four items: 3 files and 1 data directory.

1. `HW3.py`: a skeleton for your functions.
2. `HW3.ipynb`: a jupyter notebook tasks the tasks of `HW3.py` and displays results.
3. 'README.md': formatted markdown file used to generate `HW.pdf` (i.e., PDF file attached separately).
4. `data/`: a folder containing data

We will review again on Thursday, but the workflow is as follows:

Please start with the notebook and the PDF. The goal is to complete the demo in the notebook, which corresponds, in order, with the PDF homework instructions.

- Unzip the folder in your PyCharm project.
- Again, README.md is the same as HW.PDF. If opened in PyCharm, a preview should render in preview (a standard format for coding projects for documentation).
- Look over the assignment, open the notebook, and start from the top - the notebook and PDF follow in order. Work through the assignment (PDF), as you step through the notebook. Use the notebook to show results and answer questions. This will be the file used for grading (although all files are included in the submission.
- HW3.py contains a template for the functions you will implement: it is set up so you can implement the function as it comes up in the notebook (i.e., follow the assignment, and it should be clear when a function is needed). At this time, implement it.
- For those that prefer development in a script, the `__main__()` also follow in order. You can debug from the notebook, too. Nonetheless, if you prefer this development style, it is there for you (it is the exact contents of the notebook. Make sure to complete the notebook for submission. Feel free to remove from the `__main__()` to the end.

Why this workflow? The reason for this is two-fold:

1. Gradescope tests the functions of the functions. This allows code to be submitted for tests and immediate feedback based on the test fails; automated grading is baked in.

2. This is a typical workflow: implementing functions is more module, allowing for ease of re-use and extendability. Notebooks are then used to run experiments - imagine having similar problems or other tasks, another notebook is created, and existing functions are imported and used to build more notebooks.

Functions are implemented in a .py file, then demos, tutorials, workbooks, etc., are done in a notebook. In this way, it is modular and allows for re-use of the .py file (i.e., for other notebook demos showing different experiments).