

Homework 1

Due Tuesday, February 24th at 11:59pm

You are encouraged to discuss the assignment in general with your classmates, and may collaborate closely with 1-2 other students on the design and logic of your solution. If you choose to do so, you must indicate with whom you worked. In addition, the code you submit must be entirely your own; two students submitting the same code will be considered plagiarism.

Code must be written in a reasonably current version of Python (>3.0), and be executable from a Unix command line. You are free to use Python's standard modules for data structures and utilities.

Data Wrangling with Python

For reasons I can't quite explain, I keep a detailed record of every time I put gasoline in my car. For this assignment, you will be writing Python code to read, process, and analyze this data using common Python tools and idioms. Whenever possible, you are encouraged to take advantage of Python's extensive library of built-in modules ("Don't reinvent the wheel!"). Remember, the [Python docs](#) are your friend.

Data Format

The file named `mustard_data.csv` contains almost 16 years of refueling records in [comma-separated value](#) (csv) format. The file has five columns (date, mileage, location, gallons, cost per gallon) and 427 rows (including the header row). Note that some of the rows are missing one or more fields. When dealing with these cases, you'll have to do something graceful.

Code

The included file `mustard_analytics.py` contains skeleton code to help get you started. For each of the exercises, you'll need to flesh out the body of one of the function stubs (see the comments in the code for more detail). You are welcome to define additional functions if you'd like, but do not change the names of the existing functions.

Your program must be executable from a Unix(-like) command line. For example:

```
purple:~/$ python mustard_analytics.py /path/to/mustard_data.csv
```

Grading

We will run your program and examine the output for correctness. Note that we may run it on a data file with a different number of rows (but the same column format), so your code should be able to handle data files of arbitrary lengths. Your grade will be determined by how many of the exercises achieve the correct output, with partial credit being awarded.

In addition, up to 10% will be added as a bonus for code that is written in a clean, Pythonic style. For some style guidelines, check out [PEP-8](#).

What to Submit

You should submit:

- A modified `mustard_analytics.py`
- The pdf files generated for exercises 8-10 (`mustard_months.pdf`, `mustard_mpg_time.pdf`, and `mustard_in_between_days.pdf`)
- A `readme.txt`, containing
 - Your name
 - A fun fact about yourself
 - Anyone with whom you worked with on the assignment (see note above)
 - Notes or warnings about what you got working, what is partially working, and what is broken