Problems

Setup

For this week, we've provided an image called `pementorship/w4_problems`. However, all it has is a bunch of data files, which you can also find under the week 4 folder in the repo.

We'll be working with files, data, and different techniques for joining data.

Question 1

For this problem, we'll be reusing the CSV file from week 2 (called `people.csv`). On the previous problem, we looked at calculating averages with the help of a script. For this week, implement the following things using your language of choice:

1. Write a simple program that opens the file, and prints out only the first name of each person on the file.
2. What syscalls do you expect this operation to make? Confirm by using `strace`.
3. Calculate the average height of people in the dataset using the language of your choice.
4. Averages are often not the only useful statistic when you're looking at data. Calculate the standard deviation, 5th percentile, 95th percentile and 99th percentile of the heights? If you're not sure how to calculate those, take a look at this: https://en.wikipedia.org/wiki/Percentile (all you have to do is order the list and find the value at which the given percent of values is below that threshold).
5. Why would calculating percentiles when looking at data be useful? What does it give you that an average and standard deviation does not?

Question 2

Let's say you're given a directory filled with logs, passed in as an argument on the CLI. The logs have a common format that looks as follows (assuming we're looking at the logs under `/var/facebook/logs/hphp/logmaschine`):

```
/var/facebook/logs/hphp/logmaschine
|-- 2020-08-15_logmaschine_dyno.binlog
|-- 2020-08-15_logmaschine_error.binlog
|-- 2020-08-14_logmaschine_access.binlog
|-- 2020-08-14_random_log.binlog
|-- NOT_A_LOG.md
|-- syslogs
|   |-- Makefile
|   |-- 2020-07-11_syslog.binlog
|   +-- 2020-07-12_syslog.binlog
+-- misc
    +-- main
        +-- 2019-12-31_main.binlog
[...]
```

Basically, every log in this folder will follow the format `YYYY-MM-DD_*.binlog`, where YYYY, MM and DD are respectively the year, month and day for the log file. Every file that doesn't follow this format should be ignored and kept as is by your program.

1. Write a piece of code that deletes all files older than 5 days old, based on the date in the file name. Your program should work recursively in every folder, and also delete files in subfolders. We recommend implementing this in python as it has a powerful path library included in the language. Take in an argument for the folder where you'll be analyzing and deleting folders (this can be a relative path). As an example, imagine you call your program `log_clear`. Then, for the example above, you'd run `log_clear /var/facebook/logs/hphp/logmaschine` from your terminal.

2. What syscalls is your program using to explore the filesystem? Use strace to find out. What does it use to get a list of all files in a directory? What about to delete files?

3. Re-implement your program to use the file's creation time instead.

   HINT: you can use the `stat()` syscall (i.e., this is python's interface to stat) for this purpose.