Data Science Problem Assignment:
The Corrupted Ledger Problem

You are a data science consultant!

Your client has made a terrible mistake. They have acquired a company from a bankruptcy proceeding. The acquired company, called Vanish, had only one asset of value: their receivables which are recorded in a ledger. The ledger file (csv format) has a column for each of the following fields (all integers):

customer_id: a sequential number, unique, given to each customer
age: the age of the customer in years
amount: the amount in dollars that are owed

Vanish has separate records which have the name and addresses for each customer_id. This is enough information to bill the people and collect the money. What could go wrong with that?

The trouble is that the ledger file has been corrupted. What appears to have happened is the following. The file was written out one buffer at a time. A buffer is constant number of lines. You've figured out that the buffer size is 3 lines.

A bug seems to have created the situation where, before every buffer write, there is some fixed but unknown probability of a kind of random mutation occurring. The effect of that mutation is that the order of two randomly chosen columns gets swapped from the previous state. Most often, there is no change but sometimes this swap occurs. After a few such mutations, the order is completely shuffled from what the header row says the column should mean. The column ordering is however constant within a buffer write, which you know is 3 lines.

See the header of the file below. You can see that, by the third buffer (7th row), at least one swap has occurred because one of the "ages" is 9864 which can't be right. Clearly ages have moved into the first column "customer_id".

customer_id	age	amount
12221	26	358
4282	21	1050
13875	34	795
48	20	17879
22378	18	24682
9	26	22146
38	9864	11622
51	72	4852
18	9	17856
32	15119	2825
28	0	23778
31	446	1304
30	714	6600
31	148	14872
28	115	3124
33	273	17992
31	6	4538

Your job as a data scientist consultant is to find a way of correcting the ledger as best you can so that your client can mail the bills to the right people at the right address with the right amount in order to recover the maximum value.

You can assume that all customers who receive a bill for the correct value or less than the correct value will pay what was asked for, and the others will not pay at all. If you guess a customer id that does not exist, you also won't collect anything.

You are scored on the percentage of the total recovered to the total actually due.

You can assume:

- The people are all adults (age 18 or greater)
- The buffer size is 3
- Customer ids of customers no longer active are not in the file (so more ids than lines)
- Whatever the probability of a mutation happening, it is fixed. Only one mutation can occur at a buffer boundary (i.e. one column always remains unchanged from before).
- The total amount is more than $1 million but less than $2 million.

Instructions

We will send you these instructions 1 day ahead of time. Feel free to think about it ahead of time but we do not want you writing code to try to solve it until we meet during the interview. We will provide you with some starter code at interview time which does a few useful things like read the file and make histograms etc.

We don't believe thinking about the problem excessively before the interview is going to greatly raise your chance of passing the interview so don't do that. We will likely steer you in a different direction anyway.

We will start the interview by talking through the problem. You can offer a few different ideas for solving it. We will choose one of those approaches, and have you start writing code to do it.

We have the correct ledger so we can validate the results when you have them and for each iteration of your code. For example, for the 0^{th} iteration, we can just take the corrupted file as the solution and see how that does. It will only collect about 20%. Bad but better than nothing. Then we want you to move towards the best solution in iterations. Start with something simple that you can complete quickly and add more sophistication to try to get better results.

Scoring

You will be scored on the following things:

- The quality of the ideas you have for solving the problem
- The ability and efficiency at writing code to implement those ideas
- The quality of your code and program design
- The score that you get on your best version
- Your ability to iterate, improve and collaborate with the interviewers
- Your ability to discuss the ideas you have as well as what ideas you would look into if you had more time to improve it further

Further points:

- We do not expect you to come up with the absolute best solution in this short amount of time. A good solution is one that is simple and easy to implement, without relying heavily on libraries is preferred.
- When we have a decent solution in place we will talk about how to make it better if we had more time to work on it
- Treat this like a real session solving a real problem. If you forget syntax, just Google it or whatever. No artificial constraints other than not discussing it with anyone other than the interviewers. Also feel free to ask questions.
- Don't stress too much about style and creating polished code. We know you could polish it later on if you had time to refactor. We are more focused on the overall form of what you do which gives more of an indication of how well you write code.
- We expect you'll write code in an IDE of some kind (like PyCharm). If you like using notebooks as a front end, or ipython shell for experimentation, you may but we expect you to have a command line script that takes in the corrupted file and creates the uncorrupted file of the same format, headers and row ordering that we can score.

Good luck!