Code Report

The code is divided into two parts:

1. Medianvals\_by\_zip.txt
2. Medianvals\_by\_date.txt

**Logic for Medianvals\_by\_zip.txt**

The given conditions for this case are:

1. Other\_ID should be empty
2. Consider only the first 5 chars from the 9 char zip code, ignore if empty or less than 5 chars
3. Ignore record if CMTE\_ID or TRANSACTION\_AMT are empty.
4. The records are to be fed into the code sequentially
5. We can consider each record to be new, duplicates need to be addressed as new for simplicity.

The code logic starts by reading the file using infile method and not file method. If infile is not used then the program runs out of memory and as the file is huge and contains a lot of data so the method needs to be such which can be used to import each record sequentially without the program memory getting exhausted.

The records are read from the file and are split based on the new line character (‘\n’) to have one record per line and then we process it.

We then take each line and break the entire array of information on the pipe character ‘|’. Now, we get access to the information we want at index values 0,14 and 15.

Index 0 holds CMTE\_ID

Index 10 holds the zip code which is then sliced to only consider first 5 characters.

Index 13 holds the Transaction\_DT

Index 14 holds the Transaction\_AMT

We add zip code to the dictionary with a counter as key value pair. The counter is checked each time the loop runs, its incremented if the record already exists else it adds the record to the dictionary.

In the other dictionary i.e. zipcode\_sum we add the key and value pair of zipcode and the corresponding sum.

The mean is simply the zipcode\_sum divided by the final count from dictionary zip\_count.

These values are then all clubbed to be written into the txt file 🡪 medianvals\_by\_zip.txt

**Logic for Medianvals\_by\_date.txt**

The considerations to be taken here are:

1. Other\_ID should be empty
2. Ignore Malformed Dates
3. CMTE shouldn't be empty
4. Transaction Date not empty

The logic used here is that we a create a dictionary inside a dictionary to hold the values.

The outer dictionary is the one which holds the CMTE\_ID as the key and the other dictionary as its value.

The other dictionary holds the key as “date” and a list as its value. The list contains two elements:

1. Total Count for unique date and same CMTE\_ID
2. The sum money donated on these dates for this CMTE\_ID

For example, the structure looks like:

Outer\_dictionary

Inner\_Dictionary

List Val[1] , List Val[2]

With values it will look like:

C0008938

01022017

3, 350

02022017

5, 460

C0008945

01022017

5, 3650

02022017

6, 4550

With this pattern a final dictionary is created, and two loops are used to loop the structure and print the values in the file 🡪 medianvals\_by\_date.txt

\*\*\*\*