Team094 – Yuqing (Ida) Wu and Baiyang Liu

5.1 Additional Feature: market value per livable sqft  
Our additional feature computes the average market value per livable square foot for a given ZIP code by dividing the total market value by the total livable area of properties in that ZIP code.

This calculation uses the **properties csv** for both market value and livable area. To ensure correctness, we validated the output against manually calculated averages from a reduced-size dataset in both excel and pandas and confirmed that the number of records used matched expectations. We also included logging and debug during development to trace data inclusion and confirm proper filtering logic.

5.2 Use of Data Structures

1. ArrayList used in Loaders, such as propertyLoader, covidLoader, populationLoader

We used ArrayList to store rows from input files since we needed fast sequential access and preserved ordering. It was preferred over LinkedList because we didn’t require frequent insertions or deletions, and ArrayList offers better cache locality and memory efficiency for iteration-heavy tasks.

2. HashMap used in DataProcessor for ZIP-to-population mapping

To enable fast population lookups during per capita calculations, we used HashMap for expected constant-time access. We considered TreeMap for sorted output, but since sorting was deferred until display time, HashMap offered better performance for frequent data access.

3. HashSet used in Main for duplicate argument detection

We used HashSet to track which argument names had already been seen, allowing efficient O(1) checks for duplicates. A List was considered but would require linear scans, making it less suitable for ensuring uniqueness.

5.3 Lesson Learned

My teammate and I have experience working together at MCIT before. One of us lives in the USA and the other in China, we make use of the time zone difference to accomplish ‘non-stop’ working mode for this final project.

For the project, we divided responsibilities into two groups—one of us focused on data parsing and loaders, while the other handled user interface and processor logic. However we also cross check each other’s work periodically. We primarily communicated through Zoom meetings and messages whenever one person goes to sleep and the other person picks up the project, which allowed for quick decision-making and clarification. We went to TA hours together when we couldn’t agree on each other, and used GitHub for version control, committing changes frequently and using branches to test features before merging.

One challenge we encountered was ensuring consistent understanding of CSV data handling, especially with edge cases like quoted fields with commas. Unfortunately we were under the weather right before the submission date and had to work late night to fix bugs before deadline. In hindsight, earlier alignment could have helped us avoid this last-minute sprint. Going forward, we would set up a clearer workflow with regular code reviews and shared design docs to improve collaboration.