Hey PM,

I just finished aggregating and standardizing the member rosters. The sqlite table is located at `**./interview.db**` under the name `**std_member_info**` with column names `**member_id, member_first_name, member_last_name, date_of_birth, main_address, city, state, zip_code, payer**`.

Here are the summary statistics you requested.
- Of the members in the 5 rosters, 86418/109213 members were distinct and 22795/109213 were duplicates.
- By payer, 32347 distinct members were 'Madv' and 54071 distinct members were 'Mdcd'

These queries were done under the assumption that each distinct member had a zip code with a corresponding score located in the `**model_scores_by_zip**` table.
- 6676 distinct members had a `food_access_score` less than 2.
- 3.068 was the average `**social_isolation_score**`.
- 38 members were living in the zipcode with the highest `**algorex_sdoh_composite_score**`. This zip code was 95950, and the SDOH score was 8.77. If you would like a specific `**member_id**` of the people living in this zipcode, please let me know.

I first created the `std_member_info` table with the specified column names to get these results. Then for each member roster table, I took a peek at the first few rows to figure out the data format. These are some observations I had:
1) The columns of the rosters are named differently (Person_Id, First_Name, Last_Name, Dob, Street_Address, City, State, Zip, payer).
2) In `**roster_4**`, the state is represented by the state abbreviation 'CA' instead of 'California'. This differs from the other 4 rosters as they all use the full state name. I assumed that full-state names were desired, so I converted the abbreviations in this roster to full-state names.
3) In `**roster_2**` the date is formatted as `**mm/dd/yyyy**` instead of `**yyyy-mm-dd**` which is the format that the other 4 rosters use. I assumed that the most common format was desired so I converted the dates in this roster to be formatted as `**yyyy-mm-dd**`.

After looking at the rosters, I inserted them 1 by 1 into the `**std_member_info**` table. This allows for duplicate member_ids to be added to the table if any members are in more than 1 roster table. I have also made a version of this script to skip any duplicate member_ids so the table is duplicate-free. Let me know which version you prefer.

With the table created, I ran the queries that you requested. Please let me know if you would like me to explain how the queries work in further detail.

Best,
Albert