Amrita Natarajan
DATA_ENG 300
Homework 2
May 13, 2025

Part 1 (Relational)

1. Create a summary of type of drugs and their total amount used by ethnicity. Report the top usage in each ethnicity group. *You may have to make certain assumptions in calculating their total amount*
    a. See Docker Container
    b. In the SQL query for question 1, there are three main parts to it. First, the ethnicity, drug, and number of times the drug appears for each ethnicity is counted and returned as drug_count (by grouping by ethnicity). Total amount in this query is determined as number of instances because the units of measure for each drug varies and can often not yield an equivalent form of measure, for example a puff of one drug may not transfer well to mg of another. Therefore, frequency will be tracked instead. This requires the ADMISSIONS table to connect to PRESCRIPTIONS through hospital admissions (hadm_id) in order to account for every visit to the hospital. Next, the number of occurrences of drugs for each ethnicity is ranked from greatest to least (DESC) as a rank. Finally, ethnicity, drug, and drug_count are returned in a table in which only the top most (by the greatest number of instances) used drug was returned for each ethnicity (with rank 1).
    c. See Docker Container
    d. The table suggests that “MAIN” is the most common drug type across all ethnicities (returned as highest rank for each) which is also seen in the figure below. Here we see that “MAIN” leads with “BASE” being the second most common drug type across all ethnicities. It is interesting to notice that overall, there are the most prescription counts for those who visited whose ethnicity is white in this dataset (see Fig. 1). An interesting step further would be to consider the proportion of each type of drug for each ethnicity, and this is already explored on a basic level visually through the figure, noticing the ratio

between “BASE” and “MAIN” is always yielding a higher proportion as “MAIN”.

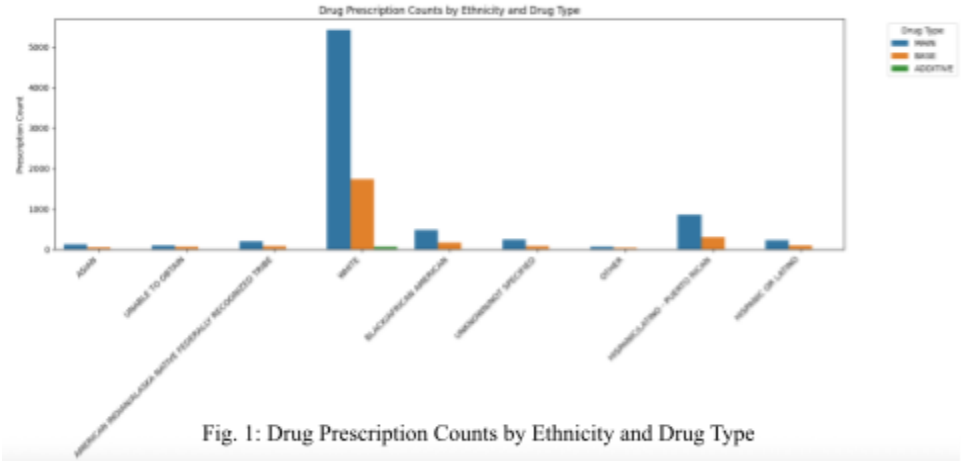

Fig. 1: Drug Prescription Counts by Ethnicity and Drug Type

2. Create a summary of procedures performed on patients by age groups (<=19, 20-49, 50-79, >80). Report the top three procedures, along with the name of the procedures, performed in each age group.
    a. See Docker Container
    b. After using the date of birth and date of death from PATIENTS to determine the age of each patient in the database, the query itself has three main parts to it. First, within patient_procedures, CASE is used to create the distinction between each age range in order to separate the rest of the information by age category. Additionally, ADMISSIONS, PROCEDURES_ICD, and D_ICD_PROCEDURES are all joined together to integrate information on hadm_id (who each admission is) and icd9_code (the numerical representation of each procedure) to generate the name associated with each procedure and to each patient admitted. Second, the procedure_counts group counts the number of each procedure within each age category to return the total number of admissions in that category to have that procedure done to them. Finally, the ranked_procedures ensures that within the table the procedures are ranked from most frequent to least and the top three are returned. The resulting table is ordered by age group (from youngest to oldest) then lists the most frequent procedures with the number of admissions in that group returned with it. Note that there is no <=19 category which is why there are no values returned for it in the table.
    c. See Docker Container
    d. From the table, it is evident that Venous cath NEC is the most common procedure across all age groups, consistently ranked first regardless of age. For individuals aged 20–79, Enteral infusion of nutritional substances appears as the second most common procedure, while for those aged 80 and above, the second most frequent procedure is Packed cell transfusion. This change suggests a potential increase in transfusion frequency with age, which is further supported by its appearance as the third most common procedure in the 50–79 group. However, since the procedure counts are not standardized by the number of individuals in each age group, we cannot conclude a definitive relationship between age and frequency of this procedure. Additional analysis would be needed to normalize for population size per age group and assess relative procedure frequencies. Furthermore, certain procedures, such as Fus/refus 2-3 vertebrae and Applic ext fix dev-femur, appear

only in the youngest group (0–19), while others like Insert endotracheal tube are unique to the oldest group (>80), indicating possible age-specific treatment patterns.

3. How long do patients stay in the ICU? Is there a difference in the ICU length of stay among gender or ethnicity?
    a. See Docker Container
    b. The ICUSTAYS table is altered to add a column that takes the outtime date and subtracts the intime date to yield how many days a patient stays in ICU (as stay_length). This query is converted into a database (while excluding where stay_length is null) in order to graph the distribution of days and to generate the measures of spread. The next query combines ADMISSIONS and ICUSTAYS tables through the hadm_id in order to link stay length (stay_length) in the ICU to each admitted patient. It also connects PATIENTS to ADMISSIONS through subject_id in order to attribute an individuals' gender to the admission. It also puts this information into a dataframe in order to manipulate it into making graphs and calculating measures of spread. For the last one, similar to for gender, this query creates a dataset for icu_stays and their link to ethnicity by joining admissions with patients, and linking ADMISSIONS to ICUSTAYS through hadm_id because ICU stays are not initially linked with admissions data. It also checks that stay length is not null because that data would not be useful in calculating stay length for each category.
    c. See Docker Container
    d. For all values in stay_length, The distribution for ICU stay length is heavily skewed right with the median at 2 days and the mean around 4.42 days. It is evident from the histogram that most patients stay between 1 and 4 days (see Fig. 2). In the next set of data, for males in the ICU, the median is still at 2 but the mean has shifted to 3.52 days, with a smaller standard deviation at 4.18 suggesting a smaller spread which makes sense as there is a smaller skew in the data. Additionally, the IQR is only 2 days, while for all the data points it was 3 (1 to 4 days), suggesting a smaller spread as well (see Fig. 3). For females, the median remains at 2 but the spread of data is much larger than for males, as the IQR (from 1 to 4.5) is 3.5 days, and the mean is much farther from the median at 5.48 days with a standard deviation at 7.8 days. The skew appears to be greater in this set as there is an individual who stayed in the ICU for 35 days, which is more than 6 days longer than the individual who stayed the longest in the male data (22 days) (see Fig. 4). From the last dataset, there is a much bigger range of differences between ethnicities than for gender. For example, for the category "AMERICAN INDIAN/ALASKA NATIVE FEDERALLY RECOGNIZED TRIBE" it is identified that the median is 11.5 days with an IQR from 1 to 22 days. This is also characterized by there being less individuals with that ethnicity in the dataset, so these measures are not reflective of people of this ethnicity overall, just in terms of the data presented in this set. White individuals are more closely aligned to the initial spread of all the data with a median of 2 days (same as intial) and an IQR from 1 to 3 days (initial was 1 to 4 days). The mean also falls at 4.02, similar to the mean of the initial set at 4.42 days. There is a greater number of White individuals (122) in the dataset compared to American Indian/Alaska Native (4), so it makes sense for the measures of spread to more closely align to the overall (see Fig. 5).

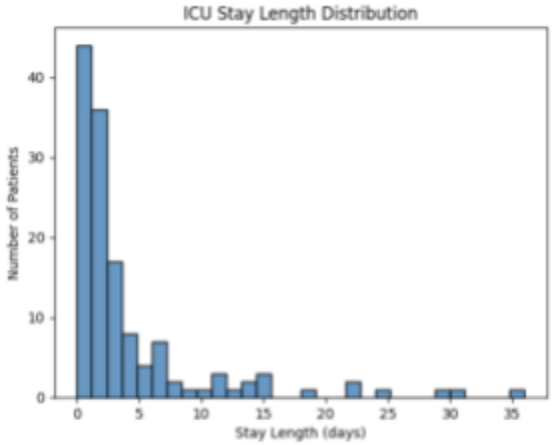

Fig. 2: Stay Length (in days) by Frequency of Number of Patients

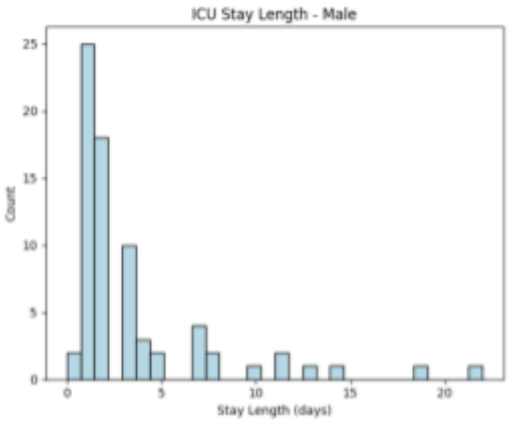

Fig. 3: Frequency of Male Patients by ICU Stay Length (in days)

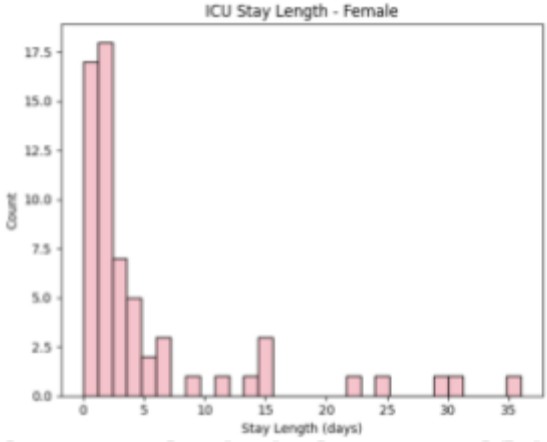

Fig. 4: Frequency of Female Patients by ICU Stay Length (in days)

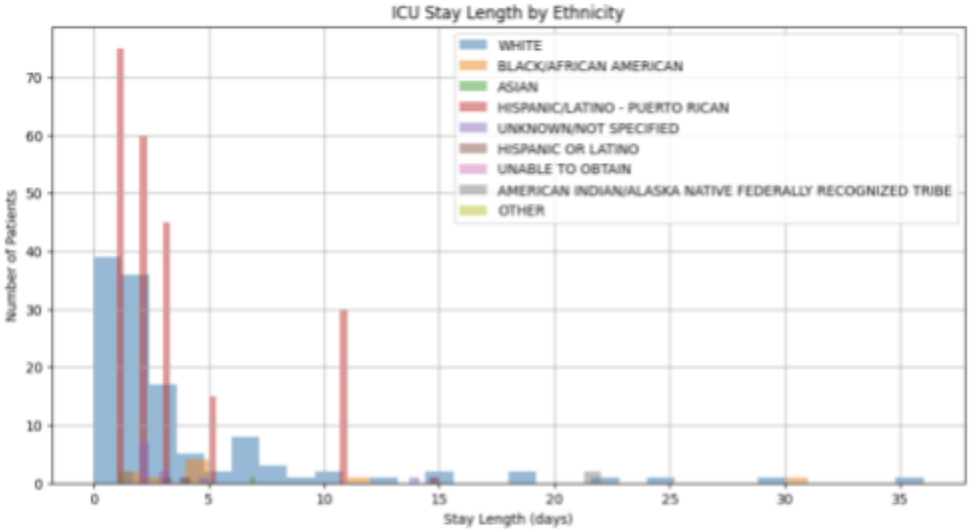

Fig. 5: Frequency of Patients of each Ethnicity Category by ICU Stay Length (in days)

Part 2 (Non-relational):

In Jupyter notebook. Note that for all three questions, aggregating data was not done in Cassandra as it did not well support joining tables and calculating new columns. Instead, pandas was used to merge data after tables were generated and pulled into dataframes to answer the initial question.

AI Disclosure:

Chat-GPT was used in this assignment in Part 1 to aid in developing graphs to show data visually. For this I asked it to generate a histogram for the given fields in the way I wanted it to be structured (for example for Ethnicity against stay length, I asked for stay length to be on the x axis with each ethnicity in a different color on the graph). Additionally, within the SQL component it was referenced when rank and case were involved as this was a new feature I had not learned previously, so in separating a query into three components where one was rank dependent and the other split into cases was useful (I asked for a way to separate my data into the specified categories using the age category, and I asked how to order procedures within a table when not all of them were under the same age category).

For Part 2, I relied a bit more on Chat-GPT initially in figuring out how to structure my data pulling, cleaning and uploading as with many moving parts I was unsure of the limitations of Cassandra and what needed to be pre-processed beforehand. It was also used in debugging like when for my age category I was getting values that were too large, so I capped the age category at 150 years. It was especially useful in understanding the “merge” feature, as I asked it how can I create an association between ADMISSIONS and PATIENTS on subject_id and it told me how as well as that it allows for directionality (left join so all data is matched with the first set). Finally, for all questions it helped me associate a uuid (unique id number) with each value to ensure all were being uploaded and that duplicates were not retained for the primary key, which allowed me to debug. This was especially helpful as data was not aggregated in Cassandra but was done afterwards.