The University of North Carolina at Charlotte

Group Project Report:

Alpha Group 1

Tony Bejos, Josh Peterson, Michael Lewis and Kevin Ovendorf

Dr. Tao Feng

DSBA 6160: Big Data Design, Storage, and Provenance

May 7, 2020

TABLE OF CONTENTS

INTRODUCTION

The goal of this report is to utilize MySQL to analyze a hospital database and write scripts as well as queries to make interesting findings and extract data in preparation for analysis. In order to obtain access to the dataset, each of us underwent training through the CITI program website. We learned about various topics from The Golden Rule to institutional boards that regulate human research conducted in the US. Subject privacy is of the utmost importance when collecting sensitive health-related information and this training course provided a useful introduction before we began to conduct our own research and analysis. Once we passed all necessary modules, we were granted access to the MIMIC-III database and dove into the various requirements of the project.

After obtaining access to the MIMIC-III database, we loaded our data into a MySQL database using the CSV files downloaded from the MIMIC-III repository. Once extracted from their zip files, a schema was created that contained each table downloaded; and we proceeded to load the data using *LOAD DATA LOCAL IN FILE* code (please see Code Appendix A). Indexes were also created on most tables within the database to help sort specific fields and increase our processing time of complex queries and views (please see Code Appendix B). We mostly indexed our tables by primary key (automatic so no code needed), foreign keys, and any relevant dates within the tables.

DATABASE OVERVIEW

The MIMIC-III database contains over a decade's worth of de-identified health-related data from intensive care unit (ICU) patients at Beth Israel Deaconess Medical Center (MIMIC-III Critical Care Database, n.d.). This database was utilized throughout the entirety of this

project. Therefore, a thorough description is provided to better provide a clear overview of the MIMIC-III database.

Key Tables, Columns and Keys

ADMISSIONS

- 58,976 rows
- Entity table used to show all hospital admission data. Includes 2 primary keys: row_id (row number) and hadm_id (admission number). It also includes a foreign key for subject_id that links to a PATIENTS table. The PATIENTS table has a one-to-many relationship to the ADMISSIONS table.
- Other notable fields:
    - ADMITTIME/DISCHTIME – patient admit and discharge time columns
    - DEATHTIME – time of death if applicable, will match DISCHTIME column
    - ADMISSION_TYPE – type of admission
        - 4 unique values – *URGENT, NEWBORN, ELECTIVE, EMERGENCY*
    - ADMISSION_LOCATION/DISCHARGE_LOCATION – location of admit/discharge
    - LANGUAGE/RELIGION/MARITAL_STATUS/ETHNICITY – self-explanatory, demographics
    - EDREGTIME/EDOUTTIME – registration datetime when patient entered/exited emergency department
    - DIAGNOSIS – admitting clinician's fully written diagnosis
    - HOSPITAL_EXPIRE_FLAG – enum(1,0) death indicator
    - HAS_CHARTEVENTS_DATA – enum(1,0) chartevents indicator

CALLOUT

- 34,499 rows
- Entity table used to provide information about ICU discharge planning. Foreign keys include subject_id that links to PATIENTS table with one-to-many relationship and hadm_id that links to ADMISSIONS table with a one-to-one relationship.
- Other notable fields:
    - SUBMIT_WARDID/CURR_WARDID
    - SUBMIT_CAREUNIT/CURR_CAREUNIT
    - CALLOUT_SERVICE
    - REQUEST_/TELE/RESP/CDIFF/MRSA/VRE – ENUM(1,0) – indicator for specific precautions (i.e. REQUEST_CDIFF indicator of 1 shows client has CDIFF)
    - CALLOUT_STATUS – indicates if callout is active or not
    - CALLOUT_OUTCOME – discharged or cancelled, indicates if patient called out
    - DISCHARGE_WARDID – ward where patient was discharged (0 is home, other numbers are hospital ward codes)
    - ACKNOWLEDGE_STATUS – response to callout event
    - CREATETIME – time and date callout was created
    - UPDATETIME – last update
    - ACKNOWLEDGETIME - time at which callout outcome was acknowledged
    - OUTCOMETIME – time at which callout outcome occurred
    - FIRSTRESERVATIONTIME/CURRENTRESERVATIONTIME – information regarding ward reservations

CAREGIVERS

- 7,567 rows
- Entity table that describes the role of specific caregivers
- Single primary key (CGID) that joins to the CHARTEVENTS table with single to multiple relationship.
- Other notable fields:
    - LABEL – type of caregiver – many unique fields that appear to be manually entered with misspellings and typographical errors
    - DESCRIPTION – additional info regarding caregiver. 17 unique values

CPTEVENTS

- 573,146 rows
- Association table that shows a patient's CPT codes as billed
- 2 foreign keys
    - SUBJECT_ID joins PATIENTS table with one-to-many relationship
    - HADM_ID joins ADMISSIONS with one-to-many relationship
- Other notable fields:
    - COSTCENTER – cost center that was billed
    - CHARTDATE – procedure date
    - CPT_CD – CPT code (varchar)
    - CPT_NUMBER – numeric code
    - CPT_SUFFIX – text suffix for CPT code
    - TICKET_ID_SEQ – order of code
    - SECTIONHEADER – category for code

- SUBSECTIONHEADER – category for code
- DESCRIPTION – meaning of respiratory code, if not respiratory related this field is NULL

CHARTEVENTS

- 330,712,483 rows
- Association table that is a list of "chartable events" (measurements like height, weight, vitals, etc.)
- 5 different foreign keys
  - SUBJECT_ID joins PATIENTS table with one-to-many relationship
  - HADM_ID joins ADMISSIONS table with one-to-many relationship
  - ICUSTAY_ID joins ICUSTAYS table with one-to-many relationship
  - ITEMID joins D_ITEMS table with one-to-many relationship
  - CGID joins CAREGIVERS table with one-to-many relationship
- Other notable fields:
  - CHARTIME/STORETIME time at which observation was made/time at which observation was input by staff member
  - VALUE – measurement associated with the ITEMID field (varchar)
  - VALUENUM – numeric value of VALUE
  - VALUEUOM – unit of measure for value
  - WARNING/ERROR – Metavision column, binary for whether error or warning was raised

- RESULTSTATUS/STOPPED – Carevue fields. RESULTSTATUS is either ‘manual’ or ‘automatic’ (shows type of measurement) and STOPPED indicates if the measurement was halted (binary)

DICTIONARIES

- 31,982 Rows (spread out through five tables)
- Association tables:
  - D_CPT, D_ICD_DIAGNOSES
  - D_ICD_PROCEDURES
  - D_ITEMS
  - D_LABITEMS
    - The above tables provide a high-level dictionary of procedural terminology; this ranges from the international classification of diseases to laboratory, and non-laboratory related items
- 2 different foreign keys:
  - ICD9_CODE:
    - Joins D_ICD_DIAGNOSES with DIAGNOSES_ICD with a one-to-one relationship
    - Joins D_ICD_PROCEDURES with PROCEDURES_ICD with a one-to-one relationship
  - ITEM_ID:
    - joins D_ITEMS and has a one-to-one relationship with:
      - CHARTEVENTS
      - DATETIMEEVENTS

- INPUTEVENTS_MV
- MICROBIOLOGYEVENTS
- OUTPUTEVENTS
- PROCEDUREEEVNTS_MV
- Joins D_LABITEMS with LABEVENTS and has a one-to-one relationship

DATETIMEEVENTS

- 4,485,937 Rows
- Association table that shows the datetime of all measurements of a patient. Values measured in timestamp format
- 5 different foreign keys:
    - SUBJECT_ID joins PATIENTS table with one-to-many relationship
    - HADM_ID joins ADMISSIONS table with one-to-many relationship
    - ICUSTAY_ID joins ICUSTAYS table with one-to-many relationship
    - ITEMID joins D_ITEMS with one-to-many relationship
    - CGID joins CAREGIVERS table with one-to-many relationship
- Other notable fields:
    - CHARTIME/STORETIME
    - VALUE – timestamp format, NOT VARCHAR or NUMERIC
    - VALUEUOM – unit of measurement, varchar
    - WARNING/ERROR enum(1,0) binary metavision indicator
    - RESULTSTATUS/STOPPED

DIAGNOSES_ICD

- 651,047 Rows
- Association table that contains diagnoses which relate to a specific hospital's admissions using the ICD9 system.
- 2 Forgeign Keys:
    - SUBJECT_ID joins PATIENTS table with a one-to-many relationship
    - HADM_ID joins ADMISSIONS table with a one-to-one relationship

DRGCODES

- 125,557 Rows
- Entity Table that contains Diagnosis Related Groups for patients.
- 2 Foreign Keys:
    - SUBJECT_ID joins PATIENTS on a one-to-many relationship
    - HADM_ID joins ADMISSIONS table with one-to-many relationship
- Other Notable Fields:
    - DRG_TYPE refers to the Type and can be HCFA or MS and they can have multiple descriptions since they change over time.
    - DRG_CODES have multiple versions so must link DRG_TYPE with these when extracting data
    - DESCRIPTION: Includes abbreviations that explain severity of diagnoses

ICUSTAYS

- 61,532 Rows
- Entity Table that is derived from the TRANSFERS table that defines a single ICU stay. It includes different databases (CareVue & Metavision) which were active at different times.

- 2 Foreign Keys:
    - SUBJECT_ID joins PATIENTS on a one-to-many relationship
    - HADM_ID joins ADMISSIONS table with one-to-many relationship
- Other Notable Fields:
    - ICUSTAY_ID is UNIQUE to each stay
    - DBSOURCE can only be two values: CareVue and Metavision
    - LOS is Length of Stay measured by INTIME and OUTTIME

INPUTEVENTS_CV

- 17,527,935 Rows
- Association Table that contains various event information for each patient in the CareVue Database.
- 5 different foreign keys:
    - SUBJECT_ID joins PATIENTS table with one-to-many relationship
    - HADM_ID joins ADMISSIONS table with one-to-many relationship
    - ICUSTAY_ID joins ICUSTAYS table with one-to-many relationship
    - ITEMID joins D_ITEMS with one-to-many relationship
    - CGID joins CAREGIVERS table with one-to-many relationship
- Other Notable Fields:
    - CHARTTIME/STORETIME
    - ITEMID identifies a single measurement type and these values are specific to the DB. For CareVue, it will be in range of 30000-39999.
    - AMOUNT/RATE – Amount and rate of drug administered
    - AMOUNTUOM/RATEUOM – unit of measurement, varchar

- ORDERID/LINKORDERID - Combines the rows where more than one ITEMID is included in the solution

INPUTEVENTS_MV

- 3,618,991 Rows
- Association Table that contains various event information for each patient in the Metavision Database.
- 5 different foreign keys:
  - SUBJECT_ID joins PATIENTS table with one-to-many relationship
  - HADM_ID joins ADMISSIONS table with one-to-many relationship
  - ICUSTAY_ID joins ICUSTAYS table with one-to-many relationship
  - ITEMID joins D_ITEMS with one-to-many relationship
  - CGID joins CAREGIVERS table with one-to-many relationship
- Other Notable Fields:
  - STARTTIME/ENDTIME and will typically be one minute apart.
  - ITEMID identifies a single measurement type and these values are specific to the DB. For Metavision, it will be greater than 220000.
  - AMOUNT/RATE – Amount and rate of drug administered
  - AMOUNTUOM/RATEUOM – unit of measurement, varchar
  - ORDERID/LINKORDERID - Combines the rows where more than one ITEMID is included in the solution
  - STATUSDESCRIPTION has 6 possible values: Changed, Paused, FinishedRunning, Stopped, Rewritten, Flushed

    - ORIGINALAMOUNT/ORIGINALRATE - Original amounts and rates at STARTTIME. ORIGINALRATE may vary from RATE since that is the ACTUAL, not predicted value.

LABEVENTS

- 27,854,055 Rows
- Association table that contains each patient's lab measurements
- 5 different foreign keys:
    - SUBJECT_ID joins PATIENTS table with one-to-many relationship
    - HADM_ID joins ADMISSIONS table with one-to-many relationship
    - ITEMID joins D_LABITEMS table with one-to-many relationship
- Other Notable Fields:
    - CHARTTIME is recorded at time when fluid is acquired
    - VALUE/VALUENUM contains value measured in Lab, can be numeric
    - VALUEUOM is unit of measure for the VALUE
    - FLAG indicates any abnormal results from the lab results

MICROBIOLOGYEVENTS

- 631,726 Rows
- Association table that contains microbiology information for each patient
- 5 Different Foreign Keys:
    - SUBJECT_ID joins PATIENTS table with one-to-many relationship
    - HADM_ID joins ADMISSIONS table with one-to-many relationship
    - SPEC_ITEMID joins D_ITEMS table with one-to-many relationship
    - ORG_ITEMID joins D_ITEMS table with one-to-many relationship

    - AB_ITEMID joins D_ITEMS table with one-to-many relationship
- Other Notable Fields:
    - CHARTDATE/CHARTTIME - not all rows have CHARTTIME but they will all have CHARTDATE since cultures can take days to complete
    - SPEC_ITEMID/SPEC_TYPE_DESC provides a unique ID and tells us what type of specimen is collected
    - ORG_ITEMID/ORG_NAME shows the organism that grew, if any. Will be NULL if none grew.
    - AB_ITEMID/AB_NAME shows antibiotic tested against organism for sensitivity
    - INTERPRETATION indicates results of AB Sensitivity: Sensitive “S”, Resistant “R”, Intermediate “I” and Pending “P.”

<u>NOTEEVENTS</u>

- 2,083,180 Rows
- Association Table that contains all Patient notes
- 3 Different Foreign Keys:
    - SUBJECT_ID joins PATIENTS table with one-to-many relationship
    - HADM_ID joins ADMISSIONS table with one-to-many relationship
    - CGID joins CAREGIVERS table with one-to-many relationship
- Other Notable Fields:
    - CHARTDATE/CHARTTIME all records will have a CHARTDATE but may not have CHARTTIME
    - STORETIME records the time at which notes were loaded into system

- CATEGORY/DESCRIPTION defines the type of note and can be used to identify Discharges for patients.
- TEXT contains the note and they can be very lengthy

OUTPUTEVENTS

- 4,349,218 rows
- Entity table that contains all item output event information such as solutions administered, solution details and measurements, and the caregiver responsible for administering the solution.
- ROW_ID is the primary key
- The table can join with the following tables:
  - SUBJECT_ID joins PATIENTS table with one-to-many relationship
  - HADM_ID joins ADMISSIONS table with one-to-many relationship
  - ICUSTAY_ID joins ICUSTAYS table with one-to-many relationship
  - ITEMID joins D_ITEMS with one-to-many relationship
  - CGID joins CAREGIVERS table with one-to-many relationship

PATIENTS

- 46,520 rows
- Entity table that contains all patient information such as sex of the patient, date of birth, various dates of death and indicates whether a patient died
- ROW_ID is the primary key
- The table can join with the following tables:
  - ADMISSIONS
  - ICUSTAYS

PRESCRIPTIONS

- 4,156,450 rows
- Entity table that contains all prescription information such as prescription type, name of prescription, and dosage
- ROW_ID is the primary key
- The table can join with the following tables:
    - ADMISSIONS
    - PATIENTS
    - ICUSTAYS

PROCEDUREEVENTS_MV

- 258,066 rows
- Entity table that contains all procedure event information such as location, category, status, and comments
- ROW_ID is the primary key
- The table can join with the following tables:
    - ADMISSIONS
    - PATIENTS
    - D_ITEMS
    - ICUSTAYS

PROCEDURES_ICD

- 240,095 rows
- Association table that contains ICD9_CODEs and the sequence of those codes
- ROW_ID is the primary key

- The table can join with the following tables:
    - ADMISSIONS
    - PATIENTS
    - D_ICD_PROCEDURES

SERVICES

- 73,343 rows
- Association table that contains the current and previous service as well as the transfer time between those services
- ROW_ID is the primary key
- The table can join with the following tables:
    - ADMISSIONS
    - PATIENTS

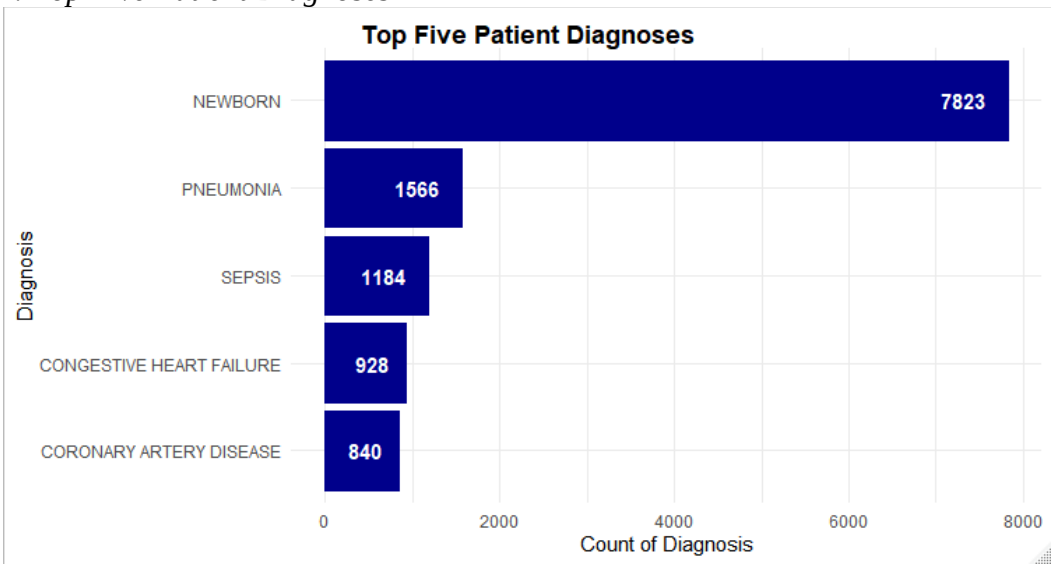
TRANSFERS

- 261,897 rows
- Entity table that contains transfer information such as previous and current care unit, ward number, length of stay and reason for transfer
- ROW_ID is the primary key
- The table can join with the following tables:
    - ADMISSIONS
    - PATIENTS
    - ICUSTAYS

MAJOR CHARACTERISTICS OF THE MIMIC-III DATABASE

Once an understanding of the MIMIC-III database was developed, it was time to start exploring the major characteristics of the MIMIC-III database. Due to the expansiveness of the database, it would be rather cumbersome to fully explore all the characteristics of the database. Therefore, in this section, some major characteristics of the database are explored and examples of the information that can be gleaned is provided (please see Code Appendix C for all SQL queries related to the "Major Characteristics of the MIMIC-III Database" section).

Figure 1: Top Five Patient Diagnoses

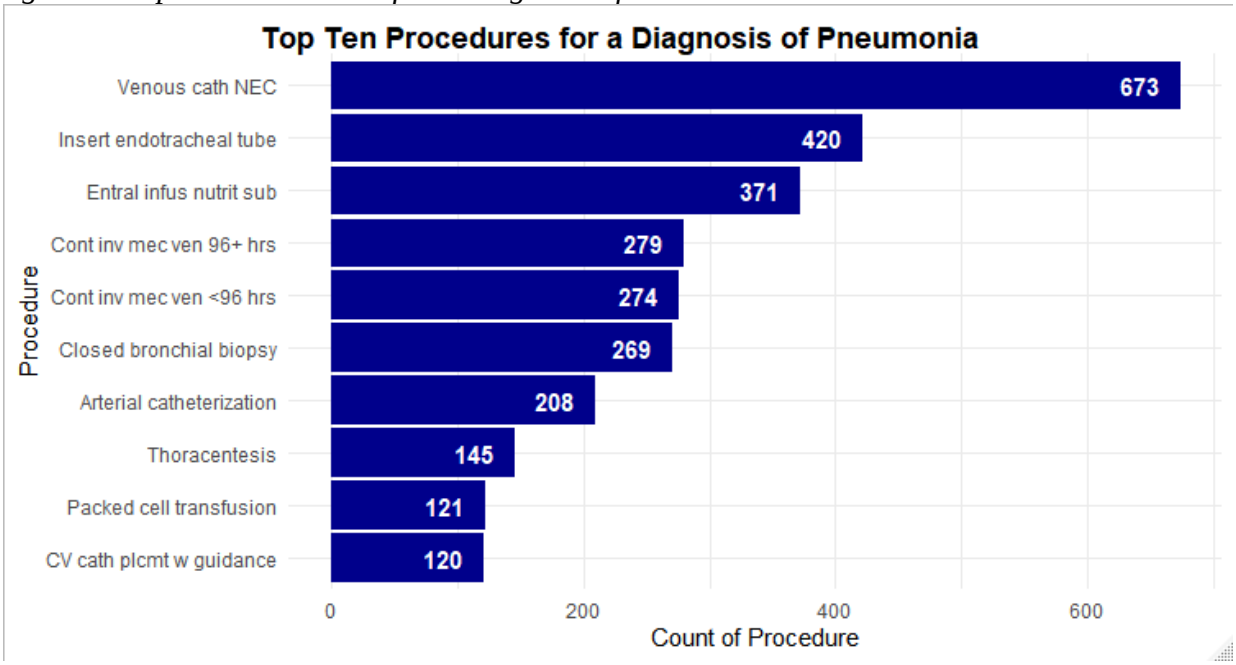


One of the first explorations conducted was finding out what the most common diagnoses were for ICU patients. Using the ADMISSIONS table, a query for the top five patient diagnoses was able to be made (please see Figure 1). Once the top diagnoses were identified, we further drilled down into the available data to understand what types of data and information could be gleaned from the database.

Pneumonia was the diagnosis chosen to further explore what patient characteristics, trends and information could be found within the database. Using a join of the ADMISSIONS and PROCEDURES_ICD tables, the icd9_codes for only those admitted patients with

pneumonia could be identified. After which, the result of that join could be joined with the D_ICD_PROCEDURES table to identify what the top ten procedures were for those admitted with a diagnosis of pneumonia (please see Figure 2). Exploring the database and making these initial queries allowed us to begin to understand the MIMIC-III database and its characteristics.

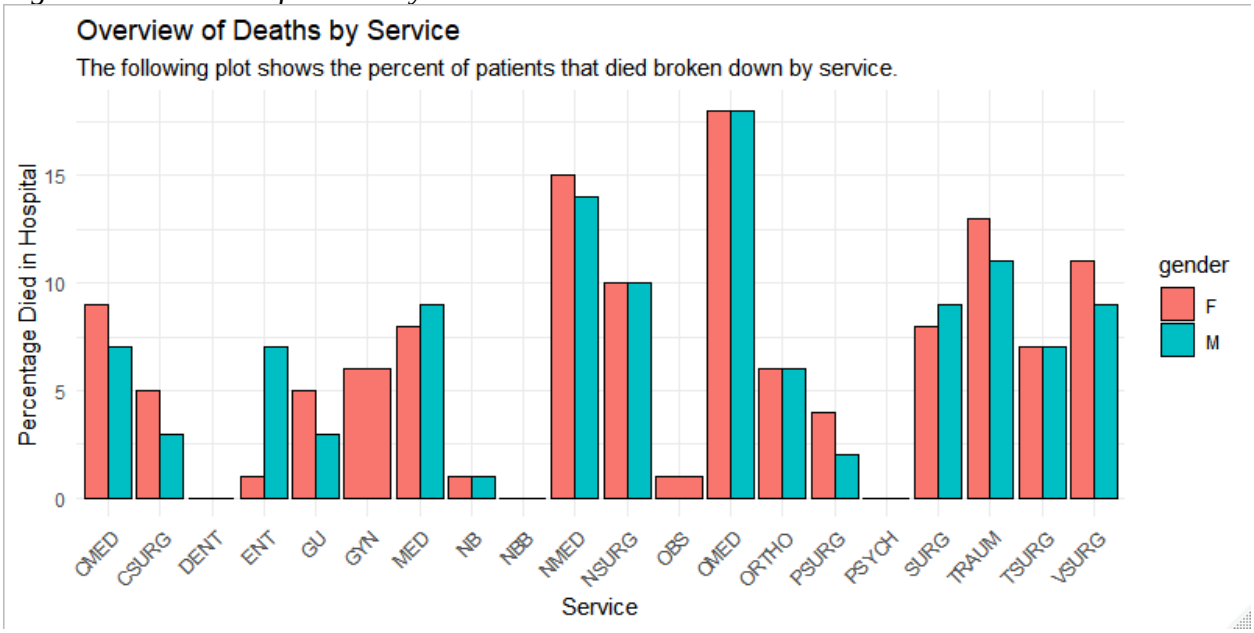
Figure 2: Top Ten Procedures for a Diagnosis of Pneumonia



Moving away from diagnoses, other tables within the database were explored as well to better understand their characteristics. For example, the proportion of individuals that died within the hospital during a particular service was investigated. Any potential disparities in death rates for different genders was explored as well to better understand any possible disparity issues within different services. Most services see rather similar mortality rates between male and female patients. However, there are a couple services where there appear to be potential disparities (please see Figure 3). For example, ENT (ear, nose and throat) shows a greater than five percent difference in the death rate between male and female patients; which is somewhat unusual when compared to other services. Therefore, it may be useful to investigate this disparity for potential causes.

Taking the services analysis one step further, patients were bucketed into age groups for each service provided during a patients' first visit. By joining the patient table with the admissions table, the first admission date was identified and this temporary view of patients' first

Figure 3: Overview of Deaths by Service

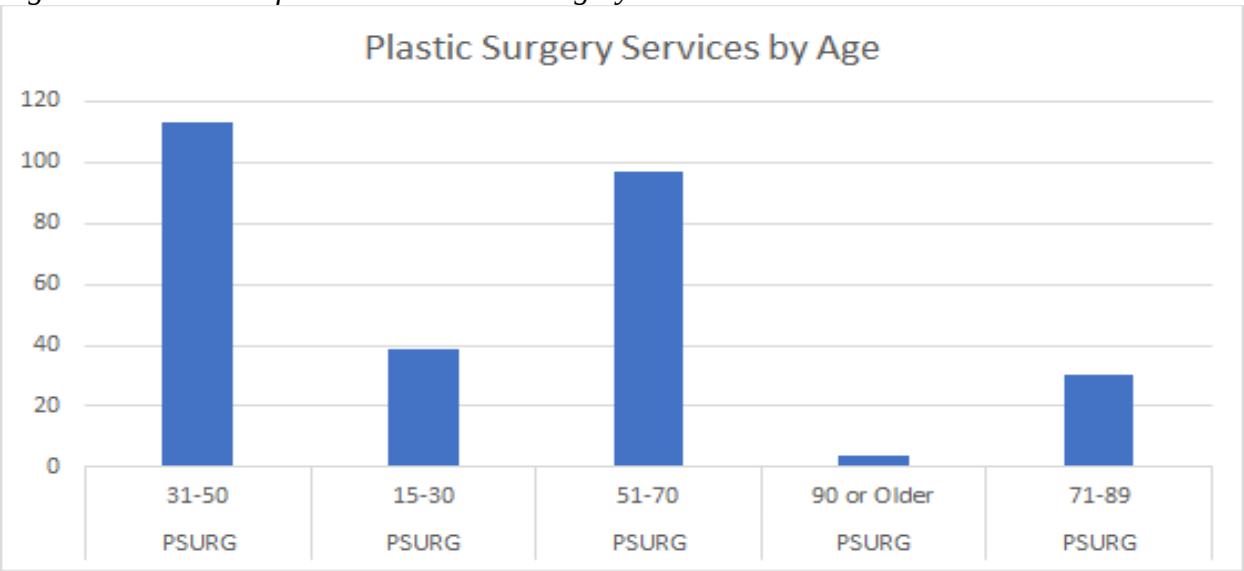


visit was created. In addition, the patient table was used to calculate the age buckets; which enables these views to be joined to produce a count for each service and age bucket. This dataset can be utilized to identify which services lean towards certain age groups. For example, plastic surgery services (PSURG) are far more common in younger populations than other services (please see Figure 4).

Furthermore, this specific query enables analysts to filter by diagnosis by joining the DIAGNOSIS_ICD table and inserting a WHERE clause that allows users to search for service information as it relates to their diagnosis. Going forward, this information can be used to predict services rendered for a given diagnosis while also considering the patient's age. These datasets provide the hospital with structured data from which to analyze and continue to maintain their data as it relates to first-visit patients. Furthermore, if we want to focus on patient diagnoses by

age group, then this query will allow that too by specifying the diagnosis and commenting out the services portion of the script in the select statement.

Figure 4: Overview of First-Visit Plastic Surgery Services



Another of our analyses involved researching mortality details within subjects' hospital stays. Our goal was to find the average hospital stay length and most common diagnosis among the deceased population in the MIMIC III database. We were able to create a query that achieved this goal (please see Code Appendix C).

The first thing we needed to find to be able to analyze mortality statistics was to find the actual population of deceased individuals among the admissions data provided in the MIMIC III database. This was achieved by pulling all subject and admission ids from the ADMISSIONS table with an expire flag of 1 (death indicator). Once we had all subject and admission ids, we were able to join this table with the PATIENTS, DIAGNOSIS_ICD, and D_ICD_DIAGNOSES tables to create our final query view.

The final view of this complex query is made up of two different subqueries. The first subquery, *dx_query*, selects the datetime of the subjects admittance using ADMITTIME (can be adjusted for the time series we wish to pull; Code Appendix C shows decade as the time

grouping for this example). It also selects all distinct ICD9_CODEs within that time series, the total count associated with the diagnoses, and the gender groupings associated with the time series found by joining the PATIENTS table. The data is grouped by ADMITTIME, ICD9_CODE, and GENDER. The WHERE clause specifically pulls data that has a HOSPITAL_EXPIRE_FLAG equal to 1.

The second subquery, *days_query*, selects the same adjustable time series to be analyzed in *dx_query,* the average hospital stay prior to death among that time series is calculated by finding the average days difference between DISCHTIME and ADMITTIME, and the same gender grouping previously mentioned. This also includes a WHERE clause to filter out all patients with a 0 for HOSPITAL_EXPIRE_FLAG.

After these subqueries were created, they were joined in the final query, along with the D_ICD_DIAGNOSIS to create a final query view. The fields selected in this view were *decade* (adjustable to the time series grouping desired by the analyst using this view), *gender*, *avg_days_since_admission*, *short_title* (of the diagnosis), and *dx_count;* which is in the count of the most common diagnosis within the grouped gender and time series.

The results proved to be interesting. It was found that in every decade other than 2200, men's most common diagnosis among the deceased (not necessarily cause of death, but diagnoses shared among the population of men) was acute respiratory failure. Men's average time of stay prior to death ranged from 8.29 days to 11.46 in any given decade. 7 out of 10 decades, the most common diagnosis among deceased females was unspecified essential hypertension. Their average hospital stay ranged from 8.15 days to 10.85 days. All considered, these statistics proved to be insightful and allowed our group to delve deeper into the intricacies of the MIMIC III database.

Lastly, another simple analysis was done to present a key for understanding all of the ICD9_CODEs and its associated diagnoses and procedures. This was done by first selecting the DIAGNOSES_ICD table along with the two dictionary tables, D_ICD_PROCEDURE and D_ICD_DIAGNOSES, more specifically the short and long titles of both the diagnosis and procedure. Then, by joining them in a union using the ICD9_CODE through three simple left outer joins, an exportable table is created that not only defines every ICD9_CODE with its diagnosis and procedure; but the overall quantity of these diagnoses and associated procedures that were recorded in the database.

This leads to various important conclusions such as that of the top ten most common ICD9_CODES, only three of them have an associated procedure and overall it is more common for a procedure to not follow a diagnosis with every ICU visit. To continue this thread, the most common diagnoses without a procedure are the following: atrial fibrillation, coronary atherosclerosis of native coronary artery, and congestive heart failure. The most common diagnoses with an associated procedure are the following: essential hypertensions (which includes a diagnostic procedure on the lymphatic structures), acute kidney failure (which includes the repair of the urethra), and lastly hyperlipidemia (which includes a biopsy of the mouth). This basic table can still be further manipulated as features such as datetime, patient ID, and lab items to name a few can be added for further analysis, so this initial table is suitable as a starting point for a vast number of research topics the analyst may want to pursue.

CHAPTER 21 – OVERVIEW

Chapter 21 outlines the process of extracting predictor variables with the intent of building various models to predict mortality in the ICU. When it comes to clinical practice, knowing the mortality rates of specific illness or injuries is crucial for determining how to

provide the proper care for the patient. Now the goal of predicting mortality with this study is to perform analytics on the average severity of different illnesses between groups of patients, which are either critically ill or involved in clinical trials. Existing ICU performance metrics include: APACHE scores, the simplified acute physiology score, the mortality probability model, and the sequential organ failure assessment score. For the purpose of our study the new performance metrics will be extracted and analyzed (MIT Critical Data, 2016).

CHAPTER 21 – EXTRACTION

The goal of chapter 21 is to develop mortality prediction models using the first ICU admissions of adult patients. However, because this chapter bases its work on the MIMIC-II database, several adjustments had to be made to accommodate the requirements of the chapter. For example, many identifiers to develop the dataset came from the "icustay_detail" table in the MIMIC-II database. However, this table is not in the MIMIC-III database. This required us to build a query that would recreate certain columns that would be found in the "icustay_detail" previously.

Elements of the "icustay_detail" table that needed to be recreated for the data extract were developed in the "icustay_detail" query that we developed (please see Code Appendix D). A DENSE_RANK function PARTITIONED over "subject_id" or "hadm_id" was used to get various ICU sequences. Additionally, calculations to determine "age" and "days_till_death_after_discharge" were made within this query as well. The development of this query was the base from which the remainder of the data extract for Chapter 21 would be based.

After the necessary elements of the "icustay_detail" table were queried, it was necessary to pull in the lab tests and vital signs data. The lab tests were gathered using the "lab_tests" query (please see Code Appendix D). For each admission, we were able to pull in the appropriate

lab tests from the “labevents” table that were associated with each hospital stay. After which, the same methodology was used to pull in the vital signs from the “chartevents” table to be used in the final query.

Finally, all the variables required for the mortality prediction models were pulled together in a final query that utilized all the aforementioned queries (please see Code Appendix D). Additionally, a target variable was created in order to determine whether or not someone died within thirty days of being discharged from the ICU. Furthermore, this final query was used to filter the data down to only the first ICU admissions and only adults; which was required of the data extract for the prediction models.

CHAPTER 21 – DATA PREPARATION

There are several steps that were taken after the data was extracted from the database to further prep the data to be used in the various mortality prediction models. Every patient older than 89 had their age set to 300 to mask it. It was suggested to replace those age records with the median age of 91.4 for elderly individuals. This record replacement was done in R after the data was extracted using SQL. Furthermore, any records with incomplete data were removed in R as well. This final data cleaning in R allowed us to properly prep and format the data in order to make it ready to be used to develop mortality prediction models.

CHAPTER 23 – OVERVIEW

The goal of Chapter 23 is to use propensity score analysis to estimate and model the probability of a subject in the study being assigned to either the control or treatment groups given the subjects’ pre-treatment conditions. This specific study is aiming to identify propensity scores for subjects that have been diagnosed with atrial fibrillation (AFIB) and rapid ventricular response (RVR). A multivariate logistic regression model will be built to accomplish this goal.

Extracting the covariates to be used for the regression and transforming them into a single view was the goal of our group. These covariates include demographics, vital signs, basic metabolic panels, past medical conditions, types of admission, and types of ICU (MIT Critical Data, 2016).

CHAPTER 23 – EXTRACTION

The first thing we did to narrow our data down was identify all subject ids and their admissions with a history of AFIB and RVR. We found that an ICD9_code of 42731 identified an AFIB diagnosis. Once we understood the ICD9_code, we were able to query the Diagnoses_ICD table for all distinct admission ids with an ICD9_code of 42731. Through this query, we were able to identify 10,271 subjects with 12,886 unique hospital admissions with a history of AFIB (please see Code Appendix E). Next, we joined these subjects with the NOTEEVENTS table to identify all patients with an RVR note in their discharge summary. To do this, we added a condition (WHERE clause) to the NOTEEVENTS query to only pull in subjects with a category of discharge summary and text that includes language like "RVR" or "rapid ventricular response". This narrowed our query results down to 806 unique subjects with 1,324 unique hospital admissions in their history.

After identifying the distinct subjects with a history of AFIB and RVR, we began to build our view including all other covariates previously described. Demographics outlined in the view were mainly found by joining the AFIB patient data from the DIAGNOSIS_ICD table with the ADMISSIONS table. The demographics we pulled from the admissions table were race, language, religion, marital status, ethnicity. We then joined the subject id to the patients table to receive gender, date of birth, and date of death (if relevant). The reason we used subject id as the key is because this basic demographic data is unlikely to change between hospital visits.

Unrelated to demographics; we were able to pull admission types from the admissions table as well, using HADM_ID as the key between our base subject query and admissions table.

All physical parameters of the subjects were pulled from the CHARTEVENTS table. The physical parameters of each admission we pulled were height, weight, heart rate, temperature, mean BP, SpO2, and SaO2. In order to pull this data, we needed to create a subquery to pivot these parameters by each admission rather than by item id. Once the subquery was created (please see Code Appendix E) we were able to simply join the admission id to the base table we created. We specifically are joining the admissions id key here because these are vitals/physical parameters and can change between different hospital visits.

Lab results were pulled in a similar way to physical parameters. A pivot subquery was made (please see Code Appendix E) to pull in the max lab results for the relevant metabolic labs for this study by admission id and subject. After the pivot table was created, the labs were easily joined to the final view by admission id.

Now that we have the views to identify our relevant subjects, we must also review their previous medical and prescription history. For complete medical history, we joined the DIAGNOSIS_ICD table to our base table using Subject_ID and the D_ICD_DIAGNOSIS using the ICD_9 code identified previously. We identified 809 distinct patients who had the conditions we found and those patients had a total of 1,908 various individual diagnoses in their history. By including the admit and discharge times, the analysis can properly analyze any conditions the patient experienced prior to these specific heart issues.

For the prescription history, we performed a similar join from PRESCRIPTIONS using Subject_ID and made sure to include the start and end dates. The new view shows all prescriptions administered to our subset of patients with RVR or AFIB diagnosis and resulted in

1,290 distinct prescription records from which any subject's prescription history can be extracted. Now that we have all the information at hand, we can run the propensity model to identify the test and control groups.

CODE APPENDIX

A. **appendix-a_alpha-group-1_db-creation.sql:** Based on MIMIC-III MySQL build found at https://github.com/MIT-LCP/mimic-code/blob/master/buildmimic/mysql/1-define.sql retrieved in April 2019. This code was used to load the database into MySQL.

B. **appendix-b_alpha-group-1_indexes.sql:** Based on MIMIC-III MySQL build found at https://github.com/MIT-LCP/mimic-code/blob/master/buildmimic/mysql/2-index.sql retrieved in April 2019. This code was used to create the indexes.

C. **appendix-c_alpha-group-1_major-characteristics.sql:** This file contains all the queries that were created to explore the various major characteristics of the MIMIC-III database that were explored.

D. **appendix-d_alpha-group-1_chapter-21-code.sql:** This file contains all the code we utilized to extract the needed data from the MIMIC-III database to meet the requirements of Chapter 21.

E. **appendix-e_alpha-group-1_chapter-23-code.sql:** This file contains all the code we utilized to extract the needed data from the MIMIC-III database to meet the requirements of Chapter 23.

REFERENCES

1. MIT Critical Data. (2016). In M. C. Data, *Secondary Analysis of Electronic Health Records*. Cambridge, MA: Springer Open.
2. "MIMIC-III Critical Care Database". *mimic.physionet.org*. Retreived in April 2019 from https://mimic.physionet.org/about/mimic/.

BIBLIOGRAPHY

1. Johnson, Alistair EW, David J. Stone, Leo A. Celi, and Tom J. Pollard. “The MIMIC Code Repository: enabling reproducibility in critical care research.” Journal of the American Medical Informatics Association (2017): ocx084.
2. MIMIC-III, a freely accessible critical care database. Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. Scientific Data (2016). DOI: 10.1038/sdata.2016.35. Available at: http://www.nature.com/articles/sdata201635
3. “MIMIC-III Critical Care Database”. *mimic.physionet.org*. Retreived in April 2019 from https://mimic.physionet.org/about/mimic/.
4. MIT Critical Data. (2016). In M. C. Data, *Secondary Analysis of Electronic Health Records*. Cambridge, MA: Springer Open.
5. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov P, Mark RG, Mietus JE, Moody GB, Peng C, and Stanley HE. Circulation. 101(23), pe215–e220. 2000.