

1.0_create_adult_inputs

May 26, 2021

These notebooks are to be used to implement the Shakespeare Method. There are 2 studies that will be referred to as the “Transfusion Study” and the “Time” study. The Transfusion study was based on creating a cohort of admissions that had a transfusion and admissions that did not (non-transfused, or control). The Time Study was based on using the Shakespeare Method to identify adverse events in different time periods rather than separating the admissions into treatment-based cohorts, we separate based on time, the way one would want to use a tool in reality. When a notebook is specific to the Transfusion or Time study it will be noted at the top. Most notebooks are general enough to be used for any type of cohort.

1 1.0_create_adult_inputs

1.0.1 Create a table `inputs_all` of adult admissions and their input events from the mimic iii database

- postgres \geq 9.4
- python=2.7 (any version should work here)
- Time and Transfused Study

```
[64]: # only run this cell if you need to reset the connection to postgres database  
→after getting an error when running a query  
conn.commit();  
cur.close();  
conn.close();
```

1.1 import libraries, connect to your postgres mimic iii database

Details on this in the readme

```
[1]: import sys  
from datetime import datetime  
import pandas as pd  
  
from importlib_metadata import version  
  
# things to connect to the postgres database  
import psycopg2  
from sqlalchemy import create_engine, update
```

```

POSTGRES_CONNECT = os.environ.get("POSTGRES_CONNECT")
POSTGRES_ENGINE = os.environ.get("POSTGRES_ENGINE")
conn = psycopg2.connect(POSTGRES_CONNECT)
cur = conn.cursor();
cur.execute("""SET search_path = mimiciii;""")
engine = create_engine(POSTGRES_ENGINE)

libraries = ['pandas','sqlalchemy','psycopg2','tqdm']
print('last ran: ',datetime.now() )
print("Python Version:", sys.version[0:7])
print( "operating system:", sys.platform)

for lib in libraries:
    print(lib + ' version: ' + version(lib))

```

```

last ran: 2019-12-24 16:22:17.780177
Python Version: 3.7.3 (
operating system: darwin
pandas version: 0.24.2
sqlalchemy version: 1.3.3
psycopg2 version: 2.7.6.1
tqdm version: 4.32.1

```

1.2 1.0.1 Create Adult Admissions Table

In this study we are looking at each admission, not each patient (because a patient can have more than one admission and we wanted to treat each admission separately)

The first criteria for our cohort is to keep only adult admissions by filtering the `admissions` table based on information from the `patients` table, resulting in a `patients_adult` table.

- adults = admissions that were 16 years or more at the time of admission
- total admissions in MIMICIII database = 58,976
- total adult admissions (16 or older) = 50,857

```

[2]: cur.execute("""
DROP TABLE IF EXISTS mimiciii.patients_adult;
WITH
first_admission_time AS
(
    SELECT pp.subject_id
           ,MIN (a.admittime) AS first_admittime
           , MIN( ROUND( (CAST(a.admittime AS date) - CAST(pp.dob AS date))/365.
→242,2))AS first_admit_age
    FROM mimiciii.patients pp
      INNER JOIN mimiciii.admissions a
      ON pp.subject_id = a.subject_id
    GROUP BY pp.subject_id
    ORDER BY pp.subject_id)
, age AS

```

```

(
    SELECT subject_id, first_admit_age
        , CASE
            WHEN first_admit_age >= 16
                THEN 'adult'
            ELSE 'pediatric'
            END AS age_group
        FROM first_admission_time)

SELECT p.*,f.first_admit_age
    INTO mimiciii.patients_adult
FROM mimiciii.patients p
    INNER JOIN age f
    ON p.subject_id = f.subject_id
    WHERE f.age_group LIKE 'adult';"""

conn.commit()

```

print total number of unique patient admissions & unique adult patient admissions

```

[3]: cur.execute("""
SELECT COUNT(DISTINCT a.hadm_id) AS admissions_total
FROM admissions a;""")

print(pd.DataFrame(cur.fetchall(), columns=['admissions_total']).
    ↳to_string(index=False))

cur.execute("""
SELECT COUNT(DISTINCT a.hadm_id) AS adult_admissions_total
FROM patients_adult p
INNER JOIN admissions a
    ON a.subject_id = p.subject_id;""")

print(pd.DataFrame(cur.fetchall(), columns=['adult_admissions_total']).
    ↳to_string(index=False))

```

```

admissions_total
          58976
adult_admissions_total
          50857

```

For this new table, the minimum **first_admit_age** should be 16.0. The maximum is 300. If the patient is older than 89, the patient's age is fixed to 300 to de-identify.

1.3 1.0.2 Create **inpuvents_mv_adult**: a table of metavision input events for adults

Filter the **inpuvents_mv** table on the **patients_adult** table using **subject_id**

```
[ ]: cur.execute("""
DROP TABLE IF EXISTS mimiciii.inputevents_mv_adult;

SELECT i.*
      INTO inputevents_mv_adult
FROM mimiciii.inputevents_mv i
      INNER JOIN mimiciii.patients_adult p
      ON p.subject_id = i.subject_id;""")
```

print number of unique adult mv admissions

```
[6]: cur.execute("""
SELECT COUNT(*) AS total_adult_mv_inputs
      , COUNT(DISTINCT hadm_id) AS unique_adult_admissions_mv_inputs
FROM mimiciii.inputevents_mv_adult;""")

print(pd.DataFrame(cur.fetchall(), columns=[
    ↳ 'total_adult_mv_inputs', 'unique_adult_admissions_mv_inputs']).
    ↳ to_string(index=False))
```

total_adult_mv_inputs	unique_adult_admissions_mv_inputs
3618905	21876

1.4 1.0.3 create inputevents_cv: CareVue Input Events for Adults

Filter the inputevents_cv table on the patients_adult table using subject_id

```
[ ]: cur.execute("""
DROP TABLE IF EXISTS mimiciii.inputevents_cv_adult;

SELECT i.*
      INTO mimiciii.inputevents_cv_adult
FROM mimiciii.inputevents_cv i
      INNER JOIN mimiciii.patients_adult p
      ON p.subject_id = i.subject_id;""")
```

print number of total inputs (rows) & unique adult cv admissions

```
[7]: cur.execute("""
SELECT COUNT(*) AS total_adult_cv_inputs
      , COUNT(DISTINCT hadm_id) AS unique_adult_admissions_cv_inputs
FROM mimiciii.inputevents_cv_adult;""")

print(pd.DataFrame(cur.fetchall(), columns=['total_adult_cv_inputs',
    ↳ 'unique_adult_admissions_cv_inputs']).to_string(index=False))
```

total_adult_cv_inputs	unique_adult_admissions_cv_inputs
15229603	27138

1.4.1 1.0.4 Create chartevents_adult: Chart Events for Adults

- It takes a long time to execute this query (more than an hour or 2 (on a laptop). Run the cell and just walk away for a bit.
- extract all chart events for adult patients (inner join chartevents to patients_adult on subject_id)
- total adult chart events = 280231912
- unique adult chart events = 49,282

```
[8]: cur.execute("""
DROP TABLE IF EXISTS mimiciii.chartevents_adult;

SELECT c.*
      INTO mimiciii.chartevents_adult
FROM mimiciii.chartevents c
     INNER JOIN mimiciii.patients_adult p
     ON p.subject_id = c.subject_id;""")
```

print number of inputs (rows) & unique adult cv admissions

```
[9]: conn.commit()
cur.execute("""
SELECT COUNT(*) AS total_adult_chartevents
      , COUNT(distinct hadm_id) AS unique_adult_admissions_chartevents
FROM mimiciii.chartevents_adult;""")

print(pd.DataFrame(cur.fetchall(), columns=[
    ↳ 'total_adult_chartevents', 'unique_adult_admissions_chartevents']).
    ↳ to_string(index=False))
```

total_adult_chartevents	unique_adult_admissions_chartevents
280231912	49282

1.5 1.0.5 Create inputs_all: All Adult Inputs (from mv and cv and chartevents)

- concatenate the tables we created above
- inputevents_cv_adult
- inputevents_mv_adult
- chartevents_adult

This query takes a bit of time as well, this is normal.

```
[10]: cur.execute("""
DROP TABLE IF EXISTS mimiciii.inputs_all;

SELECT subject_id,
       hadm_id,
       icustay_id,
       charttime AS dt,
```

```

        'cv' AS source,
        itemid
    INTO mimiciii.inputs_all
    FROM mimiciii.inpatevents_cv_adult UNION

    SELECT subject_id,
           hadm_id,
           icustay_id,
           starttime as dt,
           'mv' as source,
           itemid
    FROM mimiciii.inpatevents_mv_adult UNION

    SELECT subject_id,
           hadm_id,
           icustay_id,
           charttime as dt,
           'ce' as source,
           itemid
    FROM mimiciii.charthevents_adult;""")

```

- print number of total inputs (rows) & unique adult admissions for the big inputs_all table
- adult_admissions_count = 49284
- inputs_count = 289509356

```

[11]: cur.execute("""

    SELECT COUNT(DISTINCT hadm_id) AS adult_admissions_count
           , COUNT(*) AS inputs_count
    FROM mimiciii.inputs_all;""")

    print(pd.DataFrame(cur.fetchall(),
        ↪columns=['adult_admissions_count', 'inputs_count' ]).to_string(index=False))

```

```

adult_admissions_count  inputs_count
                    49284      289509356

```

1.5.1 Clean Up, Commit, and Close

```

[12]: conn.commit();
      cur.close();
      conn.close();

```