Prework Answer Key

Ted Laderas

August 17, 2016

This is the answer key for the prework assignment.

Loading the Data into SQLite/R

Here I first load all of my data into R as tables. Then I store the results in a new SQLite database.

```
patient= read.table("RawSyntheticData/patient.txt", header=TRUE, sep="|")
patient_encounter= read.table("RawSyntheticData/patient_encounter.txt",
                              header=TRUE, sep="|")
patient_diagnosis= read.table("RawSyntheticData/patient_diagnosis.txt",
                              header=TRUE, sep="|")
patient_encounter_hosp = read.table("RawSyntheticData/patient_encounter_hosp.txt",
                                    header=TRUE, sep="|")
race= read.table("RawSyntheticData/race.txt", header=TRUE, sep="|")
t_encounter_type= read.table("RawSyntheticData/t_encounter_type.txt",
                             header=TRUE, sep="|")
t_encounter_reason=read.table("RawSyntheticData/t_encounter_reason.txt",
                              header=TRUE, sep="|")

#load the RSQLite library
library(RSQLite)
```

```
## Loading required package: DBI
```

```
#initialize a database connection (and initialize our database)
con <- dbConnect(SQLite(),dbname="patient2.sqlite")

dbWriteTable(con, "patient", patient)
```

```
## [1] TRUE
```

```
dbWriteTable(con, "patient_encounter", patient_encounter)
```

```
## [1] TRUE
```

```
dbWriteTable(con, "patient_encounter_hosp", patient_encounter_hosp)
```

```
## [1] TRUE
```

```
dbWriteTable(con, "patient_diagnosis", patient_diagnosis)
```

```
## [1] TRUE
```

```
dbWriteTable(con, "t_encounter_type", t_encounter_type)
```

```
## [1] TRUE
```

```
dbWriteTable(con, "t_encounter_reason", t_encounter_reason)
```

```
## [1] TRUE
```

```
dbWriteTable(con, "race", race)
```

```
## [1] TRUE
```

If you want to save the tables in the database, you need to disconnect.

```
#save results (I don't save it here)
dbDisconnect(con)
```

Calculating 30 day readmissions

For the readmits and index cases, there are two approaches we can do. We can take the current table with the index cases and save it as a new table, and then produce the queries on that.

```
#if you disconnected, remember you have to reconnect to the database!

#original index query
sqlStatement <- "select peh.*, case when peh2.admit_date is null then 1
                          else 0 end as index_admit
                          from patient_encounter_hosp peh
                          left join patient_encounter_hosp peh2 on
                          peh.patientID=peh2.patientID and
                          date(peh2.admit_date) < date(peh.admit_date)"
indexTable <- dbGetQuery(con, sqlStatement)

dbWriteTable(con, name="indexAdmitTable", value=indexTable)
```

```
## [1] TRUE
```

Now that we have our `indexAdmitTable`, we can concentrate on the second query, that of identifying 30 day readmits. We need to do something similar to the above case statement, and do another self join. However, the filtering criteria for the self join is different. We want our second self-joined table (which I call `iah2`) to have dates where `iah2.admit_date` is greater than `iah.admit_date`.

We also need to filter on a difference of 30 days between the second `Admit_date` and the first `Discharge_date`. To do this, we can use the date modifiers to modify the `Discharge_date` by adding 30 days to it. Then we can compare this new date to the second admit date to complete our case statement.

```
sqlStatement <-  "select iad.*, iad2.Admit_date, CASE WHEN
                  date(iad2.Admit_date) < date(iad.Discharge_date, '+30 day')
                  THEN 1 ELSE 0 END AS Readmit30
                  from indexAdmitTable as iad
                  left join indexAdmitTable iad2 on  iad.patientID=iad2.patientID
                  and date(iad2.admit_date) > date(iad.admit_date)"

readmitTable <- dbGetQuery(con, sqlStatement)
#show number of 30 day readmits (Readmit30 == 1)
table(readmitTable$Readmit30)
```

```
##
##     0     1
## 34532  5169
```

```
readmitTable[1:20,]
```

```
##    patientid Event_ID encounter_type         outcome Admit_date
## 1          1      108             22             SNF 2014-01-01
## 2          1      109             22             SNF 2014-01-13
## 3          2     1333             22 Discharged Home 2014-01-01
## 4          3       71             22             SNF 2014-01-01
## 5          4      886             22 Discharged Home 2014-01-01
## 6          5       73             22 Discharged Home 2014-01-01
## 7          5       74             22 Discharged Home 2014-01-16
## 8          6       98             22             SNF 2014-01-01
## 9          6       99             22             SNF 2014-01-19
## 10         7      893             22           Rehab 2014-01-01
## 11         8     2556             22             SNF 2014-01-01
## 12         9      649             22             SNF 2014-01-01
## 13        10      979             22             SNF 2014-01-01
## 14        11     1815             22             SNF 2014-01-01
## 15        12      863             22 Discharged Home 2014-01-01
## 16        13     1663             22             SNF 2014-01-01
## 17        14      999             22             SNF 2014-01-01
## 18        15     1699             22 Discharged Home 2014-01-01
## 19        16       11             22           Rehab 2014-01-01
## 20        17      587             22             SNF 2014-01-01
##    Discharge_date   Admit_source index_admit Admit_date Readmit30
## 1      2014-01-08 Emergency Room           1 2014-01-13         1
## 2      2014-01-20       Transfer           0       <NA>         0
## 3      2014-01-13         Clinic           1       <NA>         0
## 4      2014-01-07       Transfer           1       <NA>         0
## 5      2014-01-07 Emergency Room           1       <NA>         0
## 6      2014-01-08 Emergency Room           1 2014-01-16         1
## 7      2014-01-25 Emergency Room           0       <NA>         0
## 8      2014-01-08 Emergency Room           1 2014-01-19         1
## 9      2014-02-01       Transfer           0       <NA>         0
## 10     2014-01-16 Emergency Room           1       <NA>         0
## 11     2014-01-06       Transfer           1       <NA>         0
## 12     2014-01-08 Emergency Room           1       <NA>         0
## 13     2014-01-04 Emergency Room           1       <NA>         0
## 14     2014-01-02       Transfer           1       <NA>         0
```

```
## 15     2014-01-20 Emergency Room           1       <NA>         0
## 16     2014-01-04       Transfer           1       <NA>         0
## 17     2014-01-02 Emergency Room           1       <NA>         0
## 18     2014-01-09         Clinic           1       <NA>         0
## 19     2014-01-08            SNF           1       <NA>         0
## 20     2014-01-03       Transfer           1       <NA>         0
```

We can also do both cases as a single query, though it is a little messy. Note that we define a third table, `peh3`, filtered on identical criteria as the table `iad2` above.

```
sqlStatement<-  "SELECT peh.*,
                CASE WHEN peh2.Admit_date IS NULL THEN 1 ELSE 0 END AS indexadmit,
                CASE WHEN date(peh3.Admit_date) < date(peh.Discharge_date, '+30 day')
                  THEN 1 ELSE 0 END AS Readmit30
                FROM patient_encounter_hosp peh
                LEFT JOIN patient_encounter_hosp peh2
                ON peh.patientid = peh2.patientid AND date(peh2.Admit_date) < date(peh.Admit_date)
                LEFT JOIN patient_encounter_hosp peh3
                ON peh.patientid = peh3.patientid AND date(peh.Admit_date) < date(peh3.Admit_date)"

bothCases <- dbGetQuery(con, sqlStatement)
bothCases[1:5,]
```

```
##   patientid Event_ID encounter_type         outcome Admit_date
## 1         1      108             22             SNF 2014-01-01
## 2         1      109             22             SNF 2014-01-13
## 3         2     1333             22 Discharged Home 2014-01-01
## 4         3       71             22             SNF 2014-01-01
## 5         4      886             22 Discharged Home 2014-01-01
##   Discharge_date   Admit_source indexadmit Readmit30
## 1     2014-01-08 Emergency Room          1         1
## 2     2014-01-20       Transfer          0         0
## 3     2014-01-13         Clinic          1         0
## 4     2014-01-07       Transfer          1         0
## 5     2014-01-07 Emergency Room          1         0
```

Finally, let's save our results back into the database and save everything.

```
dbWriteTable(con, "analytic", bothCases)
```

```
## [1] TRUE
```

```
dbDisconnect(con)
```

```
## [1] TRUE
```

We can also write our results to a file using `write.table()`. Here we save the result as a tab-delimited file:

```
write.table(bothCases, file = "analytic.txt", quote=F, sep="\t", row.names = FALSE)
```

Here are the numbers you should have:

```r
#total number of patient hospital encounters:
nrow(bothCases)
```

```
## [1] 39701
```

```r
#total number of index cases
sum(bothCases$indexadmit)
```

```
## [1] 34532
```

```r
#total number of readmits in table
sum(bothCases$Readmit30)
```

```
## [1] 5169
```