STATS405_HW3

Yuan Yi Chen (Eve)

2017.6.3

A. Set up a working environment

Step1: Setup

```
#Remove Objects
rm(list=ls())

#Clear Memory
gc(reset=TRUE)
```

```
##          used (Mb) gc trigger (Mb) max used (Mb)
## Ncells 480378 25.7     940480 50.3   480378 25.7
## Vcells 871824  6.7    1650153 12.6   871824  6.7
```

```
#Load packages
library(readr)
library(RMySQL)
library(sqldf)
```

B. Run RSQLite

```
library(RSQLite)
SQLite()
```

```
## <SQLiteDriver>
```

Step1: dbconnect

```
con <- dbConnect(SQLite(), db = "database.sqlite")
```

Step2: Check our database to ensure we have already loaded our two data sets

```
dbListTables(con)
```

```
## [1] "Diag" "Prog"
```

```
#Show colnames of Diag table
dbListFields(con, "Diag")

## [1] "X"           "ID_number"   "Diag"        "mean_radius"

#Show colnames of Prog table
dbListFields(con, "Prog")

## [1] "X"           "ID_number"   "outcome"     "time"        "mean_ra
dius"
```

Step3: View our data sets

```
#1. View Diag data set - 569 obs of 3 variables (ID_number is the key)
junk1 <- dbSendQuery(con, paste("SELECT ID_number, Diag, mean_radius
                         FROM Diag", sep = ""))
diagnosis <- fetch(junk1)
head(diagnosis, 3)

##   ID_number Diag mean_radius
## 1    842302  "M"       17.99
## 2    842517  "M"       20.57
## 3  84300903  "M"       19.69

str(diagnosis)

## 'data.frame':    569 obs. of  3 variables:
##  $ ID_number  : int  842302 842517 84300903 84348301 84358402 843786
 844359 84458202 844981 84501001 ...
##  $ Diag       : chr  "\"M\"" "\"M\"" "\"M\"" "\"M\"" ...
##  $ mean_radius: num  18 20.6 19.7 11.4 20.3 ...



#2. View Prog data set - 198 obs of 4 variables (ID_number is the key)
junk2 <- dbSendQuery(con, paste("SELECT ID_number, outcome, time, mean_
radius
                                FROM Prog", sep = ""))

## Warning: Closing open result set, pending rows

prognosis <- fetch(junk2)
head(prognosis, 3)

##   ID_number outcome time mean_radius
## 1    119513     "N"   31       18.02
## 2      8423     "N"   61       17.99
## 3    842517     "N"  116       21.37

str(prognosis)
```

```
## 'data.frame':    198 obs. of  4 variables:
##  $ ID_number  : int  119513 8423 842517 843483 843584 843786 844359
844582 844981 845010 ...
##  $ outcome    : chr  "\"N\"" "\"N\"" "\"N\"" "\"N\"" ...
##  $ time       : int  31 61 116 123 27 77 60 77 119 76 ...
##  $ mean_radius: num  18 18 21.4 11.4 20.3 ...
```

C. Perform Inner Join - Only 139 obs

```
innerjoin <- dbGetQuery(con, "SELECT * FROM Diag
                        INNER JOIN Prog
                        USING (ID_number);")

## Warning: Closing open result set, pending rows

head(innerjoin)

##      X ID_number Diag mean_radius    X outcome time mean_radius
## 1  "2"    842517  "M"       20.57  "3"     "N"  116       21.37
## 2  "6"    843786  "M"       12.45  "6"     "R"   77       12.75
## 3  "7"    844359  "M"       18.25  "7"     "N"   60       18.98
## 4  "9"    844981  "M"       13.00  "9"     "N"  119       13.00
## 5 "11"    845636  "M"       16.02 "11"     "N"  123       16.02
## 6 "14"    846381  "M"       15.85 "13"     "N"  117       15.85

nrow(innerjoin)

## [1] 139
```

D. Show the processing time of doing inner join

```
system.time(innerjoin <- dbGetQuery(con, "SELECT * FROM Diag
                        INNER JOIN Prog
                                    USING (ID_number);"))

##    user  system elapsed
##       0       0       0
```

This is the end of my homework 2. Thanks for your reading and grading! :)