INFO 550 Homework 1 Solutions

Scott Gillespie

March 23, 2016

PART I:

Solutions for SQLite questions #1 - 7 (copied from SQLite database)

The proceeding chunk performs the following tasks:

1.) Creates a character string containing SQL code for each question.  
2.) Pastes the character string for each question into a single list named 'sqlList' using a for loop.  
3.) Utilizes a function ("sqlFunction") to calculate the results of an input SQL query using the 'dbFetch' and 'dbSendQuery' functions from the DBI package.  
4.) Outputs the results of each query into a list.

sqlQuestion1<-("SELECT COUNT(\*) FROM violations  
 WHERE description LIKE ('%vermin%');")   
  
  
sqlQuestion2<-("SELECT COUNT(\*) FROM violations  
 WHERE ((risk\_category = 'High Risk') AND description NOT LIKE ('%vermin%'));")  
  
  
sqlQuestion3<-("SELECT COUNT(\*) FROM inspections  
 WHERE ((date BETWEEN 20140401 AND 20140501) AND (type LIKE ('%routine%')));")  
  
  
sqlQuestion4<-("SELECT business\_id, score, date, type FROM inspections  
 WHERE score != 'NA'   
 ORDER BY score ASC  
 LIMIT 10;")  
  
  
sqlQuestion5<-("SELECT DISTINCT postal\_code FROM businesses  
 WHERE LENGTH(postal\_code) = 5 AND postal\_code != '00000';")  
  
  
sqlQuestion6<-("SELECT name, postal\_code, COUNT(name) AS mex\_count FROM businesses  
 WHERE name LIKE ('%mex%')  
 GROUP BY postal\_code  
 ORDER BY mex\_count DESC;")  
  
  
sqlQuestion7<-("SELECT businesses.business\_id, name, score FROM businesses, inspections  
 WHERE ((businesses.business\_id = inspections.business\_id) AND (name LIKE ('%mex%')) AND (score != 'NA'))   
 ORDER BY score ASC  
 LIMIT 10;")  
  
sqlList<-list()  
for (i in 1:7){  
 sqlList[i] <- list(get(paste0('sqlQuestion',i)))  
}  
  
sqlFunction<-function(x){  
 dbFetch(dbSendQuery(hwDB, x))  
}  
  
output<-list()  
for (i in 1:7){  
 output[[i]] <- sqlFunction(sqlList[[i]])  
}  
output

## [[1]]  
## COUNT(\*)  
## 1 4151  
##   
## [[2]]  
## COUNT(\*)  
## 1 6424  
##   
## [[3]]  
## COUNT(\*)  
## 1 660  
##   
## [[4]]  
## business\_id score date type  
## 1 7643 42 20120125 Routine - Unscheduled  
## 2 74522 42 20131003 Routine - Unscheduled  
## 3 6695 44 20120813 Routine - Unscheduled  
## 4 10413 45 20120622 Routine - Unscheduled  
## 5 3151 47 20111116 Routine - Unscheduled  
## 6 3309 47 20120319 Routine - Unscheduled  
## 7 6695 48 20130325 Routine - Unscheduled  
## 8 286 49 20130725 Routine - Unscheduled  
## 9 853 49 20130710 Routine - Unscheduled  
## 10 3817 50 20110601 Routine - Unscheduled  
##   
## [[5]]  
## postal\_code  
## 1 94104  
## 2 94124  
## 3 94109  
## 4 94103  
## 5 94133  
## 6 94118  
## 7 94110  
## 8 94122  
## 9 94115  
## 10 94131  
## 11 94111  
## 12 94117  
## 13 94107  
## 14 94108  
## 15 94102  
## 16 94132  
## 17 94105  
## 18 94134  
## 19 94116  
## 20 94121  
## 21 94112  
## 22 94127  
## 23 94123  
## 24 94114  
## 25 94513  
## 26 94545  
## 27 94066  
## 28 94158  
## 29 95105  
## 30 94140  
## 31 94013  
## 32 94130  
## 33 92672  
## 34 94120  
## 35 94143  
## 36 94609  
## 37 94101  
## 38 94188  
## 39 94014  
## 40 04102  
## 41 94129  
##   
## [[6]]  
## name postal\_code mex\_count  
## 1 CHIPOTLE MEXICAN GRILL #2080 94110 7  
## 2 CHIPOTLE MEXICAN GRILL #1460 94103 4  
## 3 ROSA MEXICANO 94105 4  
## 4 THE LITTLE CHIHUAHUA MEXICAN 94114 4  
## 5 CHIPOTLE MEXICAN GRILL #1230 94102 2  
## 6 CHAVO'S MEXICAN RESTAURANT 94107 2  
## 7 CHIPOTLE MEXICAN GRILL #1566 94111 2  
## 8 TAQUERIA MEXICO TIPICO 94112 2  
## 9 CHIPOTLE MEXICAN GRILL #1560 94133 2  
## 10 CHIPOTLE MEXICAN GRILL #1857 94104 1  
## 11 EL FARO MEXICAN FOOD 94108 1  
## 12 PAPALOTE MEXICAN GRILL 94117 1  
## 13 LA CIVDAD DE MEXICO 94118 1  
## 14 TOMMYS MEXICAN RESTAURANT 94121 1  
## 15 EL TOREADOR FONDA MEXICANA RESTAURANT 94127 1  
## 16 CHIPOTLE MEXICAN GRILL 94132 1  
## 17 LA PLACITA MEXICATESSEN 94134 1  
##   
## [[7]]  
## business\_id name score  
## 1 1005 CHAVITA'S MEXICAN RESTAURANT 69  
## 2 3229 TOMMYS MEXICAN RESTAURANT 70  
## 3 5140 EL FARO MEXICAN FOOD 72  
## 4 1702 EL TOREADOR FONDA MEXICANA RESTAURANT 73  
## 5 74593 LA PLACITA MEXICATESSEN 73  
## 6 1702 EL TOREADOR FONDA MEXICANA RESTAURANT 74  
## 7 70019 EL METATE CON SABOR AMEXICO LLC 74  
## 8 7770 DOMINGUEZ MEXICAN BAKERY 75  
## 9 74593 LA PLACITA MEXICATESSEN 76  
## 10 1005 CHAVITA'S MEXICAN RESTAURANT 77

Creating dplyr object tables from the SQLite database

library(dplyr)  
hwDBdplyr <- src\_sqlite(path="restaurants.db")  
businesses <- tbl(hwDBdplyr, "businesses")  
inspections <- tbl(hwDBdplyr, "inspections")  
violations <- tbl(hwDBdplyr, "violations")

Running each query as before, except now utilizing dplyr package functionalities

dplyrQuestion8 <- violations %>%   
 filter(description %like% ("%vermin%")) %>%   
 summarise(Count=n())  
dplyrQuestion8

## Source: sqlite 3.8.6 [restaurants.db]  
## From: <derived table> [?? x 1]  
##   
## Count  
## (int)  
## 1 4151  
## .. ...

dplyrQuestion9 <- violations %>%   
 filter(description %not like% ("%vermin%"), (risk\_category == "High Risk")) %>%   
 summarise(Count=n())  
dplyrQuestion9

## Source: sqlite 3.8.6 [restaurants.db]  
## From: <derived table> [?? x 1]  
##   
## Count  
## (int)  
## 1 6424  
## .. ...

dplyrQuestion10 <- inspections %>%   
 filter(between(date, '20140401', '20140501'), (type %like% ('%routine%'))) %>%   
 summarise(Count=n())  
dplyrQuestion10

## Source: sqlite 3.8.6 [restaurants.db]  
## From: <derived table> [?? x 1]  
##   
## Count  
## (int)  
## 1 660  
## .. ...

dplyrQuestion11 <- head(inspections %>%   
 select(business\_id, score, date, type) %>%   
 filter(score != 'NA') %>%   
 arrange((score)),10)  
dplyrQuestion11

## business\_id score date type  
## 1 7643 42 20120125 Routine - Unscheduled  
## 2 74522 42 20131003 Routine - Unscheduled  
## 3 6695 44 20120813 Routine - Unscheduled  
## 4 10413 45 20120622 Routine - Unscheduled  
## 5 3151 47 20111116 Routine - Unscheduled  
## 6 3309 47 20120319 Routine - Unscheduled  
## 7 6695 48 20130325 Routine - Unscheduled  
## 8 286 49 20130725 Routine - Unscheduled  
## 9 853 49 20130710 Routine - Unscheduled  
## 10 3817 50 20110601 Routine - Unscheduled

dplyrQuestion12 <- head(distinct(businesses %>%   
 select(postal\_code) %>%   
 filter(length(postal\_code) == 5, (postal\_code != '00000'))),50)  
dplyrQuestion12

## postal\_code  
## 1 94104  
## 2 94124  
## 3 94109  
## 4 94103  
## 5 94133  
## 6 94118  
## 7 94110  
## 8 94122  
## 9 94115  
## 10 94131  
## 11 94111  
## 12 94117  
## 13 94107  
## 14 94108  
## 15 94102  
## 16 94132  
## 17 94105  
## 18 94134  
## 19 94116  
## 20 94121  
## 21 94112  
## 22 94127  
## 23 94123  
## 24 94114  
## 25 94513  
## 26 94545  
## 27 94066  
## 28 94158  
## 29 95105  
## 30 94140  
## 31 94013  
## 32 94130  
## 33 92672  
## 34 94120  
## 35 94143  
## 36 94609  
## 37 94101  
## 38 94188  
## 39 94014  
## 40 04102  
## 41 94129

dplyrQuestion13 <- head(businesses %>%   
 select(name, postal\_code) %>%   
 filter(name %like% ('%MEX%')) %>%   
 group\_by(postal\_code) %>%   
 summarise(mex\_count=count(name)) %>%   
 arrange(desc(mex\_count)), 50)  
dplyrQuestion13

## postal\_code mex\_count  
## 1 94110 7  
## 2 94103 4  
## 3 94105 4  
## 4 94114 4  
## 5 94102 2  
## 6 94107 2  
## 7 94111 2  
## 8 94112 2  
## 9 94133 2  
## 10 94104 1  
## 11 94108 1  
## 12 94117 1  
## 13 94118 1  
## 14 94121 1  
## 15 94127 1  
## 16 94132 1  
## 17 94134 1

dplyrQuestion14 <- head(inner\_join(businesses, inspections,"business\_id") %>%   
 select(business\_id, name, score) %>%   
 filter(name %like% ('%MEX%'), (score != 'NA')) %>% arrange(score),10)  
dplyrQuestion14

## business\_id name score  
## 1 1005 CHAVITA'S MEXICAN RESTAURANT 69  
## 2 3229 TOMMYS MEXICAN RESTAURANT 70  
## 3 5140 EL FARO MEXICAN FOOD 72  
## 4 1702 EL TOREADOR FONDA MEXICANA RESTAURANT 73  
## 5 74593 LA PLACITA MEXICATESSEN 73  
## 6 1702 EL TOREADOR FONDA MEXICANA RESTAURANT 74  
## 7 70019 EL METATE CON SABOR AMEXICO LLC 74  
## 8 7770 DOMINGUEZ MEXICAN BAKERY 75  
## 9 74593 LA PLACITA MEXICATESSEN 76  
## 10 1005 CHAVITA'S MEXICAN RESTAURANT 77

PART II:

A function to quickly pull and select variables from JSON-formatted data, utilizing a web-based API provided by the NY Times. The data was copied from the NY Times website and stored locally as a file called "hw1.JSON". This file is provided on github and the direct weblink is commented-out within the function.

library(XML)  
library(dplyr)  
library(jsonlite)  
  
nytimes\_movies <- function(){  
  
 jsonAPI <- fromJSON("C:/Users/sgille3/Projects/Course Work/INFO - Software Engineering/Homework/hw1.json")  
 #http://api.nytimes.com/svc/movies/v2/reviews/all.json?api-key=4ce21e5e6b299a7e95d2a14a12345f5e:12:74694642  
 retdf <- jsonAPI$results %>%   
 select(display\_title, mpaa\_rating, critics\_pick, thousand\_best, headline, opening\_date) %>%   
 rename(names=display\_title)  
   
 return(retdf)  
}  
nyt <- nytimes\_movies()  
names(nyt)

## [1] "names" "mpaa\_rating" "critics\_pick" "thousand\_best"  
## [5] "headline" "opening\_date"

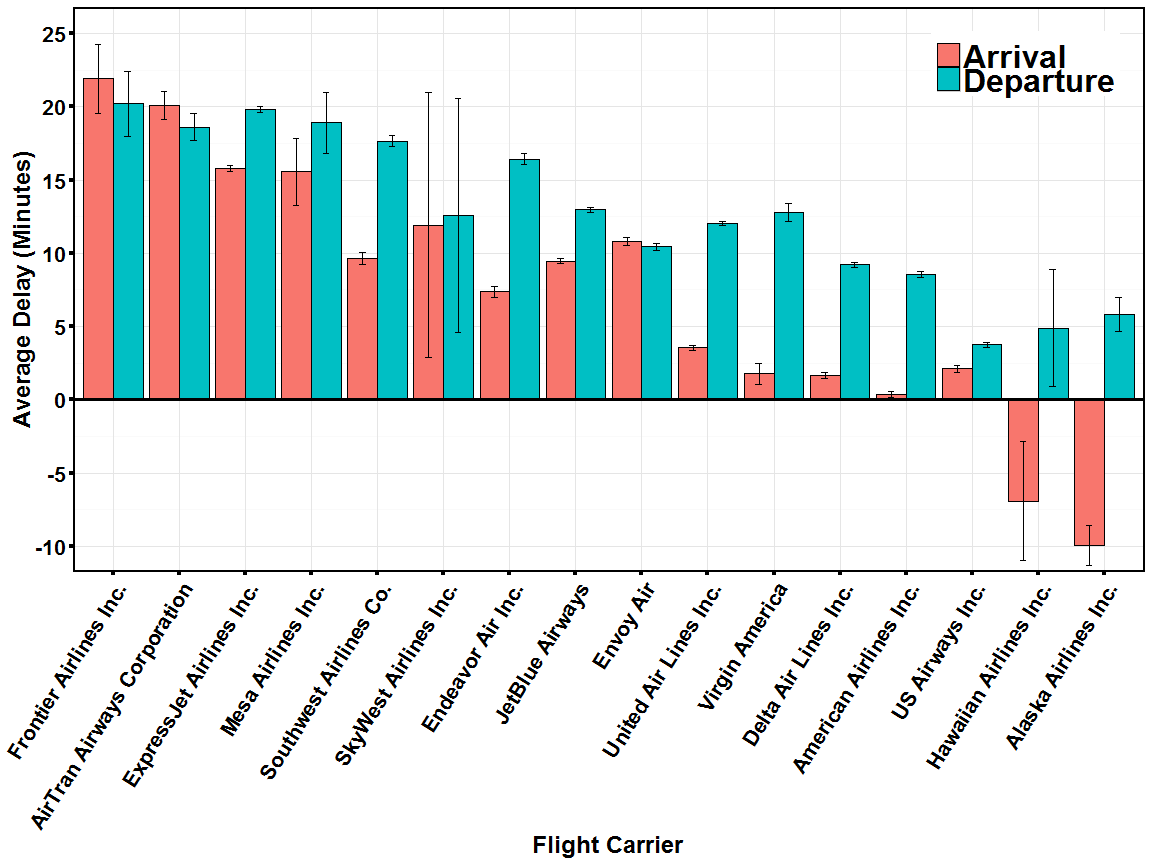
nyt %>%   
 select(names, critics\_pick, headline)

## names critics\_pick  
## 1 Big Night 1  
## 2 The Big Chill 1  
## 3 The Big Red One 0  
## 4 Little Big Man 1  
## 5 Big Deal on Madonna Street 1  
## 6 The Big Heat 1  
## 7 The Big Sky 1  
## 8 The Big Clock 1  
## 9 Miracle on 34th Street 1  
## 10 The Big Sleep 1  
## headline  
## 1   
## 2   
## 3 There's No Glory in War, Unless You Mean Survival  
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10

PART III:

Plot 1:

Average flight delays in minutes for US carriers. Arrival delays are summarized in red, and departure delays are summarized in blue. The research hypothesis concerned if more regional airlines (e.g. Hawaiian Airlines, Alaska Airlines, and Virgin America) performed better than major carriers (e.g. Frontier Airlines, Southwest Airlines, and United Airlines). Evidence from the plot seems to indicate a possible association between flight carrier type (regional vs. national) and average flight delays.



Plot 2:

Pearson correlations between arrival delay, flight speed, and number of seats and engines. Lighter blue indicates stronger associations; whereas, darker blue indicates weaker association between the variables. From the plot, number of seats seems to correlate highly with speed of the jet; likewise, number of engines has moderate correlation with seats and speed. Interestingly, arrival delay fails to associate strongly with any of these characteristics.

