ForteLab11.R

andre

2022-07-05

# Install and activate (“library()”) the sqldf package in RStudio. With any new package it is possible to run into installation issues depending on your platform and the versions of software you are running, so monitor your diagnostic messages carefully (2 points: 1 point per task).  
  
# install.packages("sqldf")  
library(sqldf)

## Warning: package 'sqldf' was built under R version 4.2.1

## Loading required package: gsubfn

## Warning: package 'gsubfn' was built under R version 4.2.1

## Loading required package: proto

## Warning: package 'proto' was built under R version 4.2.1

## Loading required package: RSQLite

## Warning: package 'RSQLite' was built under R version 4.2.1

# Review online documentation for sqldf so that you are familiar with the basic concepts and usage of the package and its commands (1point).  
# (a) Make sure the built-in “airquality” dataset is available for use in subsequent commands (hint: print head(airquality)). It would be wise to reveal the first few records of air quality with head() to make sure that air quality is available. This will also show you the names of the columns of the air quality dataframe which you will need to use in SQL commands  
  
head(airquality)

## Ozone Solar.R Wind Temp Month Day  
## 1 41 190 7.4 67 5 1  
## 2 36 118 8.0 72 5 2  
## 3 12 149 12.6 74 5 3  
## 4 18 313 11.5 62 5 4  
## 5 NA NA 14.3 56 5 5  
## 6 28 NA 14.9 66 5 6

# (b) assign airquality to an object “air”  
  
air <- airquality  
  
# c) what is the data type of air? You must use a simple command to reveal the data type (3 points).  
  
str(air)

## 'data.frame': 153 obs. of 6 variables:  
## $ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...  
## $ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...  
## $ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...  
## $ Temp : int 67 72 74 62 56 66 65 59 61 69 ...  
## $ Month : int 5 5 5 5 5 5 5 5 5 5 ...  
## $ Day : int 1 2 3 4 5 6 7 8 9 10 ...

# (a) Using sqldf(), run an SQL select command that calculates the average level of ozone across all records. Assign the resulting value into a variable (average\_ozone)   
  
average\_ozone <- sqldf("SELECT AVG(Ozone) FROM air")  
  
# print it out in the console (2 points).  
  
average\_ozone

## AVG(Ozone)  
## 1 42.12931

# Again using sqldf(), run another SQL command that selects all of the records from air quality where the value of ozone is higher than the average. Note that it is possible to combine steps 4 and 5 into a single SQL command – those who are familiar with SQL syntax and usage should attempt to do so (1 point).  
  
sqldf("SELECT \* FROM air WHERE Ozone > (SELECT AVG(air.Ozone) FROM air)")

## Ozone Solar.R Wind Temp Month Day  
## 1 45 252 14.9 81 5 29  
## 2 115 223 5.7 79 5 30  
## 3 71 291 13.8 90 6 9  
## 4 135 269 4.1 84 7 1  
## 5 49 248 9.2 85 7 2  
## 6 64 175 4.6 83 7 5  
## 7 77 276 5.1 88 7 7  
## 8 97 267 6.3 92 7 8  
## 9 97 272 5.7 92 7 9  
## 10 85 175 7.4 89 7 10  
## 11 48 260 6.9 81 7 16  
## 12 61 285 6.3 84 7 18  
## 13 79 187 5.1 87 7 19  
## 14 63 220 11.5 85 7 20  
## 15 80 294 8.6 86 7 24  
## 16 108 223 8.0 85 7 25  
## 17 52 82 12.0 86 7 27  
## 18 82 213 7.4 88 7 28  
## 19 50 275 7.4 86 7 29  
## 20 64 253 7.4 83 7 30  
## 21 59 254 9.2 81 7 31  
## 22 78 NA 6.9 86 8 4  
## 23 66 NA 4.6 87 8 6  
## 24 122 255 4.0 89 8 7  
## 25 89 229 10.3 90 8 8  
## 26 110 207 8.0 90 8 9  
## 27 44 192 11.5 86 8 12  
## 28 65 157 9.7 80 8 14  
## 29 59 51 6.3 79 8 17  
## 30 44 190 10.3 78 8 20  
## 31 45 212 9.7 79 8 24  
## 32 168 238 3.4 81 8 25  
## 33 73 215 8.0 86 8 26  
## 34 76 203 9.7 97 8 28  
## 35 118 225 2.3 94 8 29  
## 36 84 237 6.3 96 8 30  
## 37 85 188 6.3 94 8 31  
## 38 96 167 6.9 91 9 1  
## 39 78 197 5.1 92 9 2  
## 40 73 183 2.8 93 9 3  
## 41 91 189 4.6 93 9 4  
## 42 47 95 7.4 87 9 5  
## 43 44 236 14.9 81 9 11  
## 44 46 237 6.9 78 9 16

# (a) Refine step 5 to write the result table into a new R data object called “newAQ.”  
  
newAQ <- sqldf("SELECT \* FROM air WHERE Ozone > (SELECT AVG(air.Ozone) FROM air)")  
  
# (b) Then run a command to reveal what type of object newAQ is  
  
str(newAQ)

## 'data.frame': 44 obs. of 6 variables:  
## $ Ozone : int 45 115 71 135 49 64 77 97 97 85 ...  
## $ Solar.R: int 252 223 291 269 248 175 276 267 272 175 ...  
## $ Wind : num 14.9 5.7 13.8 4.1 9.2 4.6 5.1 6.3 5.7 7.4 ...  
## $ Temp : int 81 79 90 84 85 83 88 92 92 89 ...  
## $ Month : int 5 5 6 7 7 7 7 7 7 7 ...  
## $ Day : int 29 30 9 1 2 5 7 8 9 10 ...

# (c) another command to show what its dimensions are (i.e., how many rows and columns),  
  
dim(newAQ)

## [1] 44 6

# (d) a head() command to show the first few rows (4 points: one point per task).   
  
head(newAQ)

## Ozone Solar.R Wind Temp Month Day  
## 1 45 252 14.9 81 5 29  
## 2 115 223 5.7 79 5 30  
## 3 71 291 13.8 90 6 9  
## 4 135 269 4.1 84 7 1  
## 5 49 248 9.2 85 7 2  
## 6 64 175 4.6 83 7 5

# Steps above was done using a SQL way. Now, repeat steps 4, 5 and 6 in an R way, using R commands including str, mean, head, dim, which, and tapply, which is a more “R” like way to do the analysis (7 points (a through g below): one point per each step).   
# Repeat step 4: calculates the average level of ozone across all records.   
# (a) Exclude Missing Values from calculating "Ozone" mean and assign the result to "average\_ozone": Hint:use na.rm  
  
average\_ozone <- mean(air$Ozone, na.rm = TRUE)  
  
# (b) print the result (average\_ozone)  
  
average\_ozone

## [1] 42.12931

# (c) select rows with bigger values than the average ozone value   
# only keep the rows in which the Ozone values are higher than the average, and write the result table into a new R data object called "newAQ2"  
  
biggerThanMean <- function(ozone){  
 if(ozone > average\_ozone && is.na(ozone) == FALSE){  
 return(TRUE)  
 }else{  
 return(FALSE)  
 }  
}  
  
higherOzoneBool <- sapply(air$Ozone, biggerThanMean)  
  
newAQ2 <- air[higherOzoneBool, ]  
  
# (e) reveal what type of object newAQ2 is  
  
str(newAQ2)

## 'data.frame': 44 obs. of 6 variables:  
## $ Ozone : int 45 115 71 135 49 64 77 97 97 85 ...  
## $ Solar.R: int 252 223 291 269 248 175 276 267 272 175 ...  
## $ Wind : num 14.9 5.7 13.8 4.1 9.2 4.6 5.1 6.3 5.7 7.4 ...  
## $ Temp : int 81 79 90 84 85 83 88 92 92 89 ...  
## $ Month : int 5 5 6 7 7 7 7 7 7 7 ...  
## $ Day : int 29 30 9 1 2 5 7 8 9 10 ...

# (f) reveal the number of rows, then reveal the number of columns  
  
nrow(newAQ2)

## [1] 44

ncol(newAQ2)

## [1] 6

# (g) show the first few rows of "newAQ2"  
  
head(newAQ2)

## Ozone Solar.R Wind Temp Month Day  
## 29 45 252 14.9 81 5 29  
## 30 115 223 5.7 79 5 30  
## 40 71 291 13.8 90 6 9  
## 62 135 269 4.1 84 7 1  
## 63 49 248 9.2 85 7 2  
## 66 64 175 4.6 83 7 5