murphy-lab8.R

William Murphy

Fri Nov 30 14:52:03 2018

#install.packages("RCurl")  
library(RCurl)

## Loading required package: bitops

#install.packages("RJSONIO")  
library(RJSONIO)  
library(sqldf)

## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

# Step 1: Load the data  
# get data from the given URL  
dataset <- getURL("http://data.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD")  
# covert the JSON format dataset into R objects  
mydata <- fromJSON(dataset, simplify = FALSE, nullValue = NA)  
# look into the data summary  
summary(mydata)

## Length Class Mode  
## meta 1 -none- list  
## data 18638 -none- list

# Step 2: Clean the data  
#q1: get rid of metadata and only keep the second element in the list "mydata"  
mydata <- mydata[[2]]  
  
#q2: count how many elements are in "myList" and store the number into variable "numRows"  
numRows <- length(mydata)  
  
# If we didn't have nullValue = NA, then we would need to do the following  
# Test every elements in "myList" one by one, for each element:  
# First, record the element into a temporary variable "record",  
# Second, if the original element is NULL, assign NA to "record". If not, keep the original value.  
# Third, re-assign the value of "record" to the element in "myList"  
for(i in 1:numRows) {   
 record <- mydata[[i]]   
 record[sapply(record, is.null)] <- NA   
 mydata[[i]] <- record  
}  
  
# Unlist "myList" and transform it to a dataframe, whose number of rows is the value of "numRows"  
df <- data.frame(matrix(unlist(mydata), nrow=numRows, byrow=T), stringsAsFactors = FALSE)  
  
# q3: remove the first 8 columns  
df <- df[,-1:-8]  
  
# rename the rest of columns  
colnames(df) <- c("CASE\_NUMBER","BARRACK","ACC\_DATE","ACC\_TIME","ACC\_TIME\_CODE","DAY\_OF\_WEEK","ROAD","INTERSECT\_ROAD","DIST\_FROM\_INTERSECT","DIST\_DIRECTION","CITY\_NAME","COUNTY\_CODE","COUNTY\_NAME","VEHICLE\_COUNT","PROP\_DEST","INJURY","COLLISION\_WITH\_1","COLLISION\_WITH\_2")

# Step 3: Understand the data using SQL (via SQLDF)  
# how many accidents happen on SUNDAY  
# Use sql to count how many obersavations meet the criterion that DAT\_OF\_WEEK is "SUNDAY"  
sun\_acc <- sqldf("select count(DAY\_OF\_WEEK) from df where TRIM(DAY\_OF\_WEEK) = 'SUNDAY'")  
# Print the result  
print(sun\_acc)

## count(DAY\_OF\_WEEK)  
## 1 2373

# how many accidents had injuries(might need to remove NAs from the data)  
# q4: Use sql to count how many obersavations meet the criterion that INJURY value is "YES," then assign the value into "inj\_acc" Populate XXX below to complete the code.  
inj\_acc <- sqldf("select count(INJURY) from df where TRIM(INJURY) = 'YES'")  
  
# q5: Print the result  
print(inj\_acc)

## count(INJURY)  
## 1 6433

# list the injuries by day  
# count the number of injuries for each day of the week  
list\_inj <- sqldf("select DAY\_OF\_WEEK, count(CASE\_NUMBER) as Totalnumber from df where INJURY='YES' GROUP BY TRIM(DAY\_OF\_WEEK)")  
# Print the result  
print(list\_inj)

## DAY\_OF\_WEEK Totalnumber  
## 1 FRIDAY 1043  
## 2 MONDAY 915  
## 3 SATURDAY 950  
## 4 SUNDAY 818  
## 5 THURSDAY 968  
## 6 TUESDAY 843  
## 7 WEDNESDAY 896

# Step 4: Understand the data using tapply  
# how many accidents happen on SUNDAY  
# replace spaces by nothing (remove all spaces) in column "DAT\_OF\_WEEK" use gsub  
  
df$DAY\_OF\_WEEK <- gsub(" ", "", df$DAY\_OF\_WEEK)  
  
  
# tapply(Summary Variable, Group Variable, Function):   
# q6: apply the length function on the "SUNDAY" subset of the column DAY\_OF\_WEEK. populate the XXX below.  
tapply(df$DAY\_OF\_WEEK, df$DAY\_OF\_WEEK=="SUNDAY", length)

## FALSE TRUE   
## 16265 2373

# how many accidents had injuries  
# apply the length function on the "YES" subset of the column INJURY  
tapply(df$INJURY, df$INJURY == 'YES', length)

## FALSE TRUE   
## 12204 6433

# list the injuries by day  
# q7: apply the length function on subset of the column INJURY broken down by the value in WEEK\_OF\_DAY as well as INJURY. Populate XXX below  
tapply(df$INJURY, df$DAY\_OF\_WEEK, length)

## FRIDAY MONDAY SATURDAY SUNDAY THURSDAY TUESDAY WEDNESDAY   
## 3014 2554 2732 2373 2671 2676 2618

#Based on the analysis results, please answer to the following questions, if you can write your r script to draw answers:  
  
 #q8: how many accidents happen on Sunday?  
 #2372  
 #q9: What is the percentage of injury for all accidents?   
 #6433  
 #q10: Which day of a week do you observe the most injury?  
 #Friday