Alonso\_Week 5 Homework Assignment

Martin Alonso

8/16/2018

IST687 Introduction to Data Science: Week 5 Homework.

JSON and tapply Homework: Accident Analysis

### Part 1: The Code

require(RJSONIO)  
require(RCurl)  
  
#Step 1: Load the data  
url <- 'http://data.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD'  
apiResult <- getURL(url)  
df <- fromJSON(apiResult)  
  
accidents <- df[[2]]  
numRows <- length(accidents)  
  
#There are 'NULL' values in the JSON file that need to be replaced with NA.   
accidents <- sapply(accidents, function(x) ifelse(x == 'NULL', NA, x))  
dfAccident <- data.frame(matrix(unlist(accidents), nrow = numRows, byrow = T), stringsAsFactors = F)   
  
#The data.frame has successfully been imported.  
  
#Step 2: Clean the data  
#Remove the first 8 columns, and rename the remaining ones.   
dfAccident <- dfAccident[-1:-8]  
namesOfColumns <- 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")  
  
names(dfAccident) <- namesOfColumns  
  
#Remove spaces from columns  
dfAccident <- data.frame(lapply(dfAccident, function(x) gsub("\\s+", "", x)), stringsAsFactors = F)  
  
str(dfAccident)  
  
#Step 3: Understand the data using SQL (via SQLDF)  
require(sqldf)  
  
#Answer the following questions:  
#How many accidents happened on Sunday?  
sqldf("SELECT COUNT(1) AS NUM\_SUNDAYS FROM dfAccident WHERE DAY\_OF\_WEEK = 'SUNDAY'")  
#2,373 accidents happened on Sundays.   
  
#How many accidents had injuries (might need to remove NAs from data).  
#By having "INJURY = 'YES'" in the WHERE clause, we omit both NAs and "INJURY = 'NO'".   
sqldf("SELECT COUNT(1) AS NUM\_INJURIES FROM dfAccident WHERE INJURY = 'YES'")  
#6,433 accidents resulted in injuries.   
  
#List the injuries by day  
sqldf("SELECT DAY\_OF\_WEEK, COUNT(1) AS NUM\_INJURIES FROM dfAccident WHERE INJURY = 'YES' GROUP BY DAY\_OF\_WEEK")  
  
  
#Step 4: Understand the data using tapply()  
#Answer the following questions and compare results with the previous Step.  
#How many accidents happened on Sunday?  
tapply(dfAccident[dfAccident$DAY\_OF\_WEEK == 'SUNDAY', ]$DAY\_OF\_WEEK, dfAccident[dfAccident$DAY\_OF\_WEEK == 'SUNDAY', ]$DAY\_OF\_WEEK, length)  
#2,373 accidents happened on Sundays; the same number as when using 'sqldf()'.  
  
#How many accidents had injuries (might need to remove NAs from data)  
tapply(dfAccident[dfAccident$INJURY == 'YES', ]$INJURY, dfAccident[dfAccident$INJURY == 'YES', ]$INJURY, length)  
#Likewise, the 'tapply()' function shows 6,433 accidents with injuries, same number as when using the 'sqldf()' function.   
  
#List the injuries by day  
tapply(dfAccident[dfAccident$INJURY == 'YES', ]$INJURY, dfAccident[dfAccident$INJURY == 'YES', ]$DAY\_OF\_WEEK, length)

### Part 2: Running the Code

require(RJSONIO)

## Loading required package: RJSONIO

require(RCurl)

## Loading required package: RCurl

## Loading required package: bitops

#Step 1: Load the data  
url <- 'http://data.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD'  
apiResult <- getURL(url)  
df <- fromJSON(apiResult)  
  
accidents <- df[[2]]  
numRows <- length(accidents)  
  
#There are 'NULL' values in the JSON file that need to be replaced with NA.   
accidents <- sapply(accidents, function(x) ifelse(x == 'NULL', NA, x))  
dfAccident <- data.frame(matrix(unlist(accidents), nrow = numRows, byrow = T), stringsAsFactors = F)   
  
#The data.frame has successfully been imported.  
  
#Step 2: Clean the data  
#Remove the first 8 columns, and rename the remaining ones.   
dfAccident <- dfAccident[-1:-8]  
namesOfColumns <- 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")  
  
names(dfAccident) <- namesOfColumns  
  
#Remove spaces from columns  
dfAccident <- data.frame(lapply(dfAccident, function(x) gsub("\\s+", "", x)), stringsAsFactors = F)  
  
str(dfAccident)

## 'data.frame': 18638 obs. of 18 variables:  
## $ CASE\_NUMBER : chr "1363000002" "1296000023" "1283000016" "1282000006" ...  
## $ BARRACK : chr "Rockville" "Berlin" "PrinceFrederick" "Leonardtown" ...  
## $ ACC\_DATE : chr "2012-01-01T00:00:00" "2012-01-01T00:00:00" "2012-01-01T00:00:00" "2012-01-01T00:00:00" ...  
## $ ACC\_TIME : chr "2:01" "18:01" "7:01" "0:01" ...  
## $ ACC\_TIME\_CODE : chr "1" "5" "2" "1" ...  
## $ DAY\_OF\_WEEK : chr "SUNDAY" "SUNDAY" "SUNDAY" "SUNDAY" ...  
## $ ROAD : chr "IS00495CAPITALBELTWAY" "MD00090OCEANCITYEXPWY" "MD00765MAINST" "MD00944MERVELLDEANRD" ...  
## $ INTERSECT\_ROAD : chr "IS00270EISENHOWERMEMORIAL" "CO00220STMARTINSNECKRD" "CO00208DUKEST" "MD00235THREENOTCHRD" ...  
## $ DIST\_FROM\_INTERSECT: chr "0" "0.25" "100" "10" ...  
## $ DIST\_DIRECTION : chr "U" "W" "S" "E" ...  
## $ CITY\_NAME : chr "NotApplicable" "NotApplicable" "NotApplicable" "NotApplicable" ...  
## $ COUNTY\_CODE : chr "15" "23" "4" "18" ...  
## $ COUNTY\_NAME : chr "Montgomery" "Worcester" "Calvert" "St.Marys" ...  
## $ VEHICLE\_COUNT : chr "2" "1" "1" "1" ...  
## $ PROP\_DEST : chr "YES" "YES" "YES" "YES" ...  
## $ INJURY : chr "NO" "NO" "NO" "NO" ...  
## $ COLLISION\_WITH\_1 : chr "VEH" "FIXEDOBJ" "FIXEDOBJ" "FIXEDOBJ" ...  
## $ COLLISION\_WITH\_2 : chr "OTHER-COLLISION" "OTHER-COLLISION" "FIXEDOBJ" "OTHER-COLLISION" ...

#Step 3: Understand the data using SQL (via SQLDF)  
require(sqldf)

## Loading required package: sqldf

## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

#Answer the following questions:  
#How many accidents happened on Sunday?  
sqldf("SELECT COUNT(1) AS NUM\_SUNDAYS FROM dfAccident WHERE DAY\_OF\_WEEK = 'SUNDAY'")

## NUM\_SUNDAYS  
## 1 2373

#2,373 accidents happened on Sundays.   
  
#How many accidents had injuries (might need to remove NAs from data).  
#By having "INJURY = 'YES'" in the WHERE clause, we omit both NAs and "INJURY = 'NO'".   
sqldf("SELECT COUNT(1) AS NUM\_INJURIES FROM dfAccident WHERE INJURY = 'YES'")

## NUM\_INJURIES  
## 1 6433

#6,433 accidents resulted in injuries.   
  
#List the injuries by day  
sqldf("SELECT DAY\_OF\_WEEK, COUNT(1) AS NUM\_INJURIES FROM dfAccident WHERE INJURY = 'YES' GROUP BY DAY\_OF\_WEEK")

## DAY\_OF\_WEEK NUM\_INJURIES  
## 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()  
#Answer the following questions and compare results with the previous Step.  
#How many accidents happened on Sunday?  
tapply(dfAccident[dfAccident$DAY\_OF\_WEEK == 'SUNDAY', ]$DAY\_OF\_WEEK, dfAccident[dfAccident$DAY\_OF\_WEEK == 'SUNDAY', ]$DAY\_OF\_WEEK, length)

## SUNDAY   
## 2373

#2,373 accidents happened on Sundays; the same number as when using 'sqldf()'.  
  
#How many accidents had injuries (might need to remove NAs from data)  
tapply(dfAccident[dfAccident$INJURY == 'YES', ]$INJURY, dfAccident[dfAccident$INJURY == 'YES', ]$INJURY, length)

## YES   
## 6433

#Likewise, the 'tapply()' function shows 6,433 accidents with injuries, same number as when using the 'sqldf()' function.   
  
#List the injuries by day  
tapply(dfAccident[dfAccident$INJURY == 'YES', ]$INJURY, dfAccident[dfAccident$INJURY == 'YES', ]$DAY\_OF\_WEEK, length)

## FRIDAY MONDAY SATURDAY SUNDAY THURSDAY TUESDAY WEDNESDAY   
## 1043 915 950 818 968 843 896