DanBurkeHW5

Dan Burke

8/16/2021

Dan Burke

IST687

Homework 5 - JSON & tapply Homework: Accident Analysis

Assignment Due: 8/16/2021

Submitted: 8/16/2021

Step 1: Load the data

Read in the following JSON dataset http://data.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD

```
#The url provided in Step 1 is broken, I've used the following link provided by a classmate (via Slack
#https://opendata.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD

#Install Needed Libraries
#install.packages("jsonlite")
#install.packages("RJSONIO")
#install.packages("RCurl")

#Load the Libraries
library(RCurl)
library(RJSONIO)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
```

```
## The following objects are masked from 'package:RJSONIO':
##
##     fromJSON, toJSON
```

```
library(httr)
library(knitr)
```

```
url <- "https://opendata.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD"


#SSL issues require utilizing the httr lib
jsonData <- fromJSON(url)
jsonData <- jsonData$data

dfMd <- data.frame(jsonData)
```

Step 2: Clean the data

After you load the data, remove the first 8 columns, and then, to make it easier to work with, name the rest of the columns as follows: Note, not surprisingly, it is in JSON format. You should be able to see that the first result is the metadata (information about the data) and the second is the actual data.

```
#Removing the First 8 Columns
dfMd <- dfMd[, -c(1:8)]


#Renaming the Columns

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")

colnames(dfMd) <-namesOfColumns

#Trust, but Verify with a couple rows
colnames(dfMd)
```

```
##  [1] "CASE_NUMBER"         "BARRACK"             "ACC_DATE"
##  [4] "ACC_TIME"            "ACC_TIME_CODE"       "DAY_OF_WEEK"
##  [7] "ROAD"                "INTERSECT_ROAD"      "DIST_FROM_INTERSECT"
## [10] "DIST_DIRECTION"      "CITY_NAME"           "COUNTY_CODE"
## [13] "COUNTY_NAME"         "VEHICLE_COUNT"       "PROP_DEST"
## [16] "INJURY"              "COLLISION_WITH_1"    "COLLISION_WITH_2"
```

```
print(kable(dfMd[1:3,1:5]))
```

```
##
##
## |CASE_NUMBER |BARRACK          |ACC_DATE            |ACC_TIME |ACC_TIME_CODE |
## |:-----------|:----------------|:-------------------|:--------|:-------------|
## |1363000002  |Rockville        |2012-01-01T00:00:00 |2:01     |1             |
## |1296000023  |Berlin           |2012-01-01T00:00:00 |18:01    |5             |
## |1283000016  |Prince Frederick |2012-01-01T00:00:00 |7:01     |2             |
```

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

Answer the following questions: • How many accidents happen on SUNDAY
• How many accidents had injuries (might need to remove NAs from the data) • List the injuries by day

```
#How    many    accidents   happen  on  SUNDAY
#need to clean it up
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
dfMd$DAY_OF_WEEK <- trimws(dfMd$DAY_OF_WEEK)
sundayCases <- sqldf("select count(CASE_NUMBER) from dfMd where DAY_OF_WEEK == 'SUNDAY'")
sundayCases
```

```
##   count(CASE_NUMBER)
## 1               2373
```

```
#How    many    accidents   had injuries (might need    to  remove  NAs from    the data)

#Check for NAs
dfMd$INJURY <- trimws(dfMd$INJURY)
sum(is.na(dfMd$INJURY))
```

```
## [1] 1
```

```
dfMd$INJURY[which(is.na(dfMd$INJURY))] <- "NO"

sqldf("select count(INJURY) from dfMd where INJURY == 'YES'")
```

```
##   count(INJURY)
## 1          6433
```

```
#List   the injuries    by  day
sqldf("select DAY_OF_WEEK, Count(CASE_NUMBER) from dfMd where INJURY == 'YES' group by DAY_OF_WEEK")
```

```
##   DAY_OF_WEEK Count(CASE_NUMBER)
## 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: • How many accidents happen on SUNDAY
• How many accidents had injuries (might need to remove NAs from the data) • List the injuries by day

```
#How    many    accidents   happen  on  SUNDAY
#need to clean it up
dfMd$DAY_OF_WEEK <- trimws(dfMd$DAY_OF_WEEK)
cases <- paste("Accidents on Sunday (Not tapply to check):", length(which(dfMd$DAY_OF_WEEK == "SUNDAY")))
cases
```

```
## [1] "Accidents on Sunday (Not tapply to check): 2373"
```

```
#Using tapply
sundayAcci <- tapply(dfMd$CASE_NUMBER, dfMd$DAY_OF_WEEK == "SUNDAY", length)
cases <- paste("Accidents on Sunday (with tapply):", sundayAcci[2])
cases
```

```
## [1] "Accidents on Sunday (with tapply): 2373"
```

```
#How    many    accidents   had injuries (might need    to  remove  NAs from    the data)

#Check for NAs
sum(is.na(dfMd$INJURY))
```

```
## [1] 0
```

```
dfMd$INJURY[10398] <- "NO"

casesInj <- paste("Accidents with injuries (Not tapply to check):",length(which(dfMd$INJURY == "YES")))
print(casesInj)
```

```
## [1] "Accidents with injuries (Not tapply to check): 6433"
```

```
injuryAcci <- tapply(dfMd$INJURY, dfMd$INJURY == "YES", length)

casesInj <- paste("Accidents with injuries (with tapply):",injuryAcci[2])
print(casesInj)
```

```
## [1] "Accidents with injuries (with tapply): 6433"
```

```
#List   the injuries    by  day
tapply(dfMd$INJURY =="YES", dfMd$DAY_OF_WEEK, length)
```

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