IST687 - Week 5 - JSON & tapply

John Fields

4/27/2019

Step 1: Load the data

Read in the following JSON dataset https://opendata.maryland.gov/resource/pdvh-tf2u.json

```
#NOTE: The R Socrata option was used for this homework since more data was avaiable
#(10,000 vs 18,638 observations)

#install.packages(RSocrata)
library(RSocrata)
url<-"https://opendata.maryland.gov/resource/pdvh-tf2u.json"
rawdata<-read.socrata(url)
str(rawdata)
```

```
## 'data.frame':    18638 obs. of  18 variables:
##  $ acc_date           : POSIXct, format: "2012-01-01" "2012-01-01" ...
##  $ acc_time           : chr  "2:01" "18:01" "7:01" "0:01" ...
##  $ acc_time_code      : chr  "1" "5" "2" "1" ...
##  $ barrack            : chr  "Rockville" "Berlin" "Prince Frederick" "Leonardtown" ...
##  $ case_number        : chr  "1363000002" "1296000023" "1283000016" "1282000006" ...
##  $ city_name          : chr  "Not Applicable" "Not Applicable" "Not Applicable" "Not Applicable" ...
##  $ collision_with_1   : chr  "VEH" "FIXED OBJ" "FIXED OBJ" "FIXED OBJ" ...
##  $ collision_with_2   : chr  "OTHER-COLLISION" "OTHER-COLLISION" "FIXED OBJ" "OTHER-COLLISION" ...
##  $ county_code        : chr  "15" "23" "4" "18" ...
##  $ county_name        : chr  "Montgomery" "Worcester" "Calvert" "St. Marys" ...
##  $ day_of_week        : chr  "SUNDAY   " "SUNDAY   " "SUNDAY   " "SUNDAY   " ...
##  $ dist_direction     : chr  "U" "W" "S" "E" ...
##  $ dist_from_intersect: chr  "0" "0.25" "100" "10" ...
##  $ injury             : chr  "NO" "NO" "NO" "NO" ...
##  $ intersect_road     : chr  "IS 00270 EISENHOWER MEMORIAL" "CO 00220 ST MARTINS NECK RD" "CO 00208 I
##  $ prop_dest          : chr  "YES" "YES" "YES" "YES" ...
##  $ road               : chr  "IS 00495 CAPITAL BELTWAY" "MD 00090 OCEAN CITY EXPWY" "MD 00765 MAIN S
##  $ vehicle_count      : chr  "2" "1" "1" "1" ...
```

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. namesOfColumns <- c("CASE_NUMBER","BARRACK","ACC_DATE","ACC_TIME", "ACC_TIME_CODE","DAY_OF_WE EK","ROAD","INTERSECT_ROAD", "DIST_FROM_INTERSECT", "DIST_DIRECTION","CITY_NA ME","COUNTY_CODE", "COUNTY_NAME","VEHICLE_COUNT", "PROP_DEST","INJURY","COLLI SION_WITH_1", "COLLISION_WITH_2")

```
# NOTE: After importing the data via JSON, there was not a need to remove the 8 columns
#as described in the instructions

#Rename the columns
```

```r
namesOfColumns<-c("ACC_DATE","ACC_TIME","ACC_TIME_CODE","BARRACK","CASE_NUMBER","CITY_NAME",
"COLLISION_WITH_1","COLLISION_WITH_2","COUNTY_CODE","COUNTY_NAME","DAY_OF_WEEK",
"DIST_DIRECTION","DIST_FROM_INTERSECT","INJURY","INTERSECT_ROAD","PROP_DEST","ROAD",
"VEHICLE_COUNT")
cleandata<-function(rawdata,namesOfColumns)
{colnames(rawdata)<-namesOfColumns
return(rawdata)
}
results<-cleandata(rawdata,namesOfColumns)
#The NA values need to be removed from the dataset.
results <- na.omit(results)
summary(results)
```

```
##     ACC_DATE                     ACC_TIME         ACC_TIME_CODE
##  Min.   :2012-01-01 00:00:00   Length:16263       Length:16263
##  1st Qu.:2012-04-11 00:00:00   Class :character   Class :character
##  Median :2012-07-08 00:00:00   Mode  :character   Mode  :character
##  Mean   :2012-07-07 00:57:35
##  3rd Qu.:2012-10-07 00:00:00
##  Max.   :2012-12-31 00:00:00
##    BARRACK           CASE_NUMBER         CITY_NAME
##  Length:16263       Length:16263       Length:16263
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  COLLISION_WITH_1   COLLISION_WITH_2   COUNTY_CODE
##  Length:16263       Length:16263       Length:16263
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  COUNTY_NAME        DAY_OF_WEEK        DIST_DIRECTION
##  Length:16263       Length:16263       Length:16263
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  DIST_FROM_INTERSECT    INJURY          INTERSECT_ROAD
##  Length:16263        Length:16263       Length:16263
##  Class :character    Class :character   Class :character
##  Mode  :character    Mode  :character   Mode  :character
##
##
##
##   PROP_DEST             ROAD           VEHICLE_COUNT
##  Length:16263       Length:16263       Length:16263
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
```

##

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

Answer the following questions:

- How many accidents happen on SUNDAY

```
#The SQL query didn't return results due to spaces after the DAY_OF_WEEK.  The code below
#was added to remove the spaces.

#$ day_of_week        : chr  "SUNDAY   " "SUNDAY   " "SUNDAY   " "SUNDAY   " ...

results$DAY_OF_WEEK<-gsub(" ","",results$DAY_OF_WEEK)

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

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

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

```
## Could not load tcltk.  Will use slower R code instead.
```

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

```
SUNaccidents<- sqldf('select COUNT(DAY_OF_WEEK),DAY_OF_WEEK from results where DAY_OF_WEEK="SUNDAY"')
SUNaccidents
```

```
##   COUNT(DAY_OF_WEEK) DAY_OF_WEEK
## 1               2061      SUNDAY
```

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

```
injuries<- sqldf('select COUNT(INJURY),INJURY from results where INJURY="YES"')
injuries
```

```
##   COUNT(INJURY) INJURY
## 1          5639    YES
```

- List the injuries by day

```
injuriesDay<- sqldf('select COUNT(INJURY),INJURY,DAY_OF_WEEK from results where INJURY="YES" group by DA
injuriesDay
```

```
##   COUNT(INJURY) INJURY DAY_OF_WEEK
## 1           915    YES      FRIDAY
## 2           795    YES      MONDAY
## 3           827    YES    SATURDAY
## 4           705    YES      SUNDAY
## 5           864    YES    THURSDAY
## 6           748    YES     TUESDAY
## 7           785    YES   WEDNESDAY
```

Step 4: Understand the data using tapply

Answer the following questions (same as before) – compare results:

- How many accidents happen on Sunday

```
tapply(results$CASE_NUMBER, results$DAY_OF_WEEK=="SUNDAY", length)
```

```
## FALSE  TRUE
## 14202  2061
```

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

```
tapply(results$CASE_NUMBER, results$INJURY == "YES", length)
```

```
## FALSE  TRUE
## 10624  5639
```

- List the injuries by day

```
injuryByDay <- results[which(results$INJURY == "YES"),]
tapply(injuryByDay$CASE_NUMBER, injuryByDay$DAY_OF_WEEK, length)
```

```
##    FRIDAY    MONDAY  SATURDAY    SUNDAY  THURSDAY   TUESDAY WEDNESDAY
##       915       795       827       705       864       748       785
```