# Homework #5 – JSON & tapply Homework: Accident Analysis

## R Script (Code)

#

# Course: IST687

# Name: Joyce Woznica

# Homework 5 - JSON & tapply Homework: Accident Analysis

# Due Date: 2/12/2019

# Date Submitted:

#

# Library to use

install.packages("RCurl")

install.packages("jsonlite")

library(RCurl)

library(jsonlite)

#install.packages("RJSONIO")

#library(RJSONIO)

# Step 1: Load the data

# 1) read JSON dataset from a URL:

# http://data.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD

accidentURL<-"http://data.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD"

apiResult<-getURL(accidentURL)

accidentData<- fromJSON(apiResult)

# Other Options - # see Gary's document

# Step 2: Clean the data

# first row is about the data

# second has the data

summary(accidentData[[2]])

View(accidentData[[2]])

accidentData<-accidentData[[2]]

#actual number of accidents

numrow <- length(accidentData)

# how many columns

numcol <- ncol(accidentData)

# remove the first 8 columns

accidentData <- accidentData[,9:numcol]

# now rename the columns

# should now be 18 columns

ColNameVect <- 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")

# Update with new column names

colnames(accidentData)<-ColNameVect

View(accidentData)

# need to clean out extra spaces from ends of lines

trim.leading<-function(x) {sub("^\\s+","",x)}

trim.trailing<-function(x) {sub("\\s+$","",x)}

trim<-function(x) {sub("^\\s+|\\s+$","",x)}

accidentData[,6]<-trim(accidentData[,6])

# before using the data - make it a data frame

accidentData<-as.data.frame(accidentData)

# Step 3: Understad the data using SQL (via SQLDF)

install.packages("sqldf")

library(sqldf)

# 1) How many accidents happen on SUNDAY

# ANSWER: 2373

SQLStatement<-"select count(accidentData.CASE\_NUMBER) from accidentData

where accidentData.DAY\_OF\_WEEK = 'SUNDAY'"

numSundayAccidents <- (sqldf(SQLStatement, stringsAsFactors = FALSE))

numSundayAccidents

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

# ignore the na's - but did a check...there are no NA's in INJURY field

# ANSWER: 6433

sqldf("select accidentData.INJURY from accidentData where

accidentData.INJURY = 'NA'")

# this returns 0, so nothing with NA

SQLStatement<-"select count(accidentData.INJURY) from accidentData

where accidentData.INJURY = 'YES'"

numInjuries<-sqldf(SQLStatement,stringsAsFactors=FALSE)

numInjuries

# 3) List the injuries by Day

# ANSWER: See results in injuriesbyDay

# requires grouping the count of injuries by the DAY\_OF\_WEEK

SQLStatement<-"select accidentData.DAY\_OF\_WEEK, count(accidentData.INJURY) from accidentData

where accidentData.INJURY = 'YES' Group by accidentData.DAY\_OF\_WEEK"

injuriesbyDay<-sqldf(SQLStatement,stringsAsFactors=FALSE)

injuriesbyDay

View(injuriesbyDay)

# Step 4: Understand the data using tapply

# Answer the following Questions

# Note: Could use the 'attach' function to reference by DAY\_OF\_WEEK instead

# of accidentData$DAY\_OF\_WEEK

# 1) How many accidents happen on Sunday

# ANSWER: 2373

# find all the match SUNDAY

sunday<-tapply(accidentData$DAY\_OF\_WEEK, accidentData$DAY\_OF\_WEEK=='SUNDAY', length)

sunday["TRUE"]

# 2) How many accidents had injuries

# ANSWER: 6433

injury<-tapply(accidentData$INJURY, accidentData$INJURY=='YES', length)

injury["TRUE"]

# 3) List the injuries by day

# ANSWER: see console

injurydays<-tapply(accidentData$INJURY,

list(accidentData$INJURY=='YES',accidentData$DAY\_OF\_WEEK), length)

injurydays["TRUE",]

View(injurydays["TRUE",])

## Console Log (Executed Code)

> #

> # Course: IST687

> # Name: Joyce Woznica

> # Homework 5 - JSON & tapply Homework: Accident Analysis

> # Due Date: 2/12/2019

> # Date Submitted:

> #

> # Library to use

>

> install.packages("RCurl")

Installing package into ‘C:/Users/Joyce/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/RCurl\_1.95-4.11.zip'

Content type 'application/zip' length 2969979 bytes (2.8 MB)

downloaded 2.8 MB

package ‘RCurl’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Public\Documents\Wondershare\CreatorTemp\RtmpqY9IOn\downloaded\_packages

> install.packages("jsonlite")

Installing package into ‘C:/Users/Joyce/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/jsonlite\_1.6.zip'

Content type 'application/zip' length 1160716 bytes (1.1 MB)

downloaded 1.1 MB

package ‘jsonlite’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Public\Documents\Wondershare\CreatorTemp\RtmpqY9IOn\downloaded\_packages

> library(RCurl)

Loading required package: bitops

Warning messages:

1: package ‘RCurl’ was built under R version 3.5.2

2: package ‘bitops’ was built under R version 3.5.2

> library(jsonlite)

Warning message:

package ‘jsonlite’ was built under R version 3.5.2

> #install.packages("RJSONIO")

> #library(RJSONIO)

>

> # Step 1: Load the data

> # 1) read JSON dataset from a URL:

> # http://data.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD

>

> accidentURL<-"http://data.maryland.gov/api/views/pdvh-tf2u/rows.json?accessType=DOWNLOAD"

> apiResult<-getURL(accidentURL)

> accidentData<- fromJSON(apiResult)

>

> # Other Options - # see Gary's document

>

> # Step 2: Clean the data

> # first row is about the data

> # second has the data

> summary(accidentData[[2]])

V1 V2 V3

row-222r~j9dd~y23n: 1 00000000-0000-0000-0000-3FE73BABB07F: 1 0:18638

row-2264~ysk2~3k7u: 1 00000000-0000-0000-0005-84072304A8CE: 1

row-22bn-39nf.9an9: 1 00000000-0000-0000-000D-B378E10AB61B: 1

row-22cp~ahk3.gdcb: 1 00000000-0000-0000-000E-30C23B3D6312: 1

row-22eg-mdxe~qq28: 1 00000000-0000-0000-000E-7D867C1E97C5: 1

row-22ej.stcc~z83z: 1 00000000-0000-0000-0016-93783A59F3D7: 1

(Other) :18632 (Other) :18632

V4 V5 V6 V7 V8 V9

1425933202:18638 NA's:18638 1425933202:18638 NA's:18638 { }:18638 1257000644: 3

1262006287: 3

1266001445: 3

1250003311: 2

1250005131: 2

1251005139: 2

(Other) :18623

V10 V11 V12 V13 V14

Forestville : 1911 2012-07-21T00:00:00: 113 17:11 : 160 1:1665 FRIDAY :3014

College Park: 1536 2012-01-21T00:00:00: 111 17:12 : 149 2:2645 MONDAY :2554

Frederick : 1501 2012-12-09T00:00:00: 98 17:08 : 136 3:3330 SATURDAY :2732

Bel Air : 1385 2012-06-12T00:00:00: 95 16:05 : 133 4:4109 SUNDAY :2373

Rockville : 1381 2012-08-26T00:00:00: 92 16:08 : 130 5:4540 THURSDAY :2671

(Other) :10194 2012-02-29T00:00:00: 86 14:06 : 129 6:2349 TUESDAY :2676

NA's : 730 (Other) :18043 (Other):17801 WEDNESDAY:2618

V15 V16 V17 V18

IS 00095 CAPITAL BELTWAY : 1163 IS 00695 BALTO BELTWAY : 173 0 :4941 E :2656

IS 00495 CAPITAL BELTWAY : 874 MD 00185 CONNECTICUT AVE: 153 100 :1399 N :4493

IS 00695 BALTO BELTWAY : 840 MD 00100 NO NAME : 133 0.25 :1381 S :4586

US 00301 CRAIN HWY : 672 IS 00095 CAPITAL BELTWAY: 129 0.5 :1226 U :3974

IS 00095 NO NAME : 618 MD 00201 KENILWORTH AVE : 112 500 :1224 W :2548

IS 00095 J F K MEMORIAL HWY: 568 (Other) :17937 (Other):8454 NA's: 381

(Other) :13903 NA's : 1 NA's : 13

V19 V20 V21 V22 V23

Not Applicable:18170 16 :3453 Prince Georges:3453 2 :7816 NO : 7071

Mount Airy : 59 12 :1659 Harford :1659 1 :6864 YES :11566

Westminster : 20 3 :1597 Baltimore :1597 3 :2083 NA's: 1

Berlin : 19 10 :1502 Frederick :1502 4 : 480

Leonardtown : 10 15 :1387 Montgomery :1387 5 : 105

(Other) : 164 (Other):9006 (Other) :9006 (Other): 39

NA's : 196 NA's : 34 NA's : 34 NA's :1251

V24 V25 V26

NO :12204 VEH :10675 OTHER-COLLISION:13644

YES : 6433 FIXED OBJ : 4299 FIXED OBJ : 2644

NA's: 1 OTHER-COLLISION: 2205 VEH : 1759

ANIMAL : 846 NON-COLLISION : 559

NON-COLLISION : 465 PED : 17

(Other) : 147 (Other) : 14

NA's : 1 NA's : 1

> View(accidentData[[2]])

>

> accidentData<-accidentData[[2]]

> #actual number of accidents

> numrow <- length(accidentData)

> # how many columns

> numcol <- ncol(accidentData)

>

> # remove the first 8 columns

> accidentData <- accidentData[,9:numcol]

>

> # now rename the columns

> # should now be 18 columns

> ColNameVect <- 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")

>

> # Update with new column names

> colnames(accidentData)<-ColNameVect

> View(accidentData)

> # need to clean out extra spaces from ends of lines

> trim.leading<-function(x) {sub("^\\s+","",x)}

> trim.trailing<-function(x) {sub("\\s+$","",x)}

> trim<-function(x) {sub("^\\s+|\\s+$","",x)}

> accidentData[,6]<-trim(accidentData[,6])

>

> # before using the data - make it a data frame

> accidentData<-as.data.frame(accidentData)

>

> # Step 3: Understad the data using SQL (via SQLDF)

> install.packages("sqldf")

Installing package into ‘C:/Users/Joyce/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/sqldf\_0.4-11.zip'

Content type 'application/zip' length 78196 bytes (76 KB)

downloaded 76 KB

package ‘sqldf’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Public\Documents\Wondershare\CreatorTemp\RtmpqY9IOn\downloaded\_packages

> library(sqldf)

Loading required package: gsubfn

Loading required package: proto

Loading required package: RSQLite

Warning messages:

1: package ‘sqldf’ was built under R version 3.5.2

2: package ‘gsubfn’ was built under R version 3.5.2

3: package ‘proto’ was built under R version 3.5.2

4: package ‘RSQLite’ was built under R version 3.5.2

>

> # 1) How many accidents happen on SUNDAY

> # ANSWER: 2373

> SQLStatement<-"select count(accidentData.CASE\_NUMBER) from accidentData

+ where accidentData.DAY\_OF\_WEEK = 'SUNDAY'"

> numSundayAccidents <- (sqldf(SQLStatement, stringsAsFactors = FALSE))

> numSundayAccidents

count(accidentData.CASE\_NUMBER)

1 2373

>

> # 2) How many accidents had injuries (might need to remove NAs from the data)

> # ignore the na's - but did a check...there are no NA's in INJURY field

> # ANSWER: 6433

> sqldf("select accidentData.INJURY from accidentData where

+ accidentData.INJURY = 'NA'")

[1] INJURY

<0 rows> (or 0-length row.names)

> # this returns 0, so nothing with NA

> SQLStatement<-"select count(accidentData.INJURY) from accidentData

+ where accidentData.INJURY = 'YES'"

> numInjuries<-sqldf(SQLStatement,stringsAsFactors=FALSE)

> numInjuries

count(accidentData.INJURY)

1 6433

>

> # 3) List the injuries by Day

> # ANSWER: See results in injuriesbyDay

> # requires grouping the count of injuries by the DAY\_OF\_WEEK

> SQLStatement<-"select accidentData.DAY\_OF\_WEEK, count(accidentData.INJURY) from accidentData

+ where accidentData.INJURY = 'YES' Group by accidentData.DAY\_OF\_WEEK"

> injuriesbyDay<-sqldf(SQLStatement,stringsAsFactors=FALSE)

> injuriesbyDay

DAY\_OF\_WEEK count(accidentData.INJURY)

1 FRIDAY 1043

2 MONDAY 915

3 SATURDAY 950

4 SUNDAY 818

5 THURSDAY 968

6 TUESDAY 843

7 WEDNESDAY 896

> View(injuriesbyDay)

>

> # Step 4: Understand the data using tapply

> # Answer the following Questions

> # Note: Could use the 'attach' function to reference by DAY\_OF\_WEEK instead

> # of accidentData$DAY\_OF\_WEEK

> # 1) How many accidents happen on Sunday

> # ANSWER: 2373

> # find all the match SUNDAY

> sunday<-tapply(accidentData$DAY\_OF\_WEEK, accidentData$DAY\_OF\_WEEK=='SUNDAY', length)

> sunday["TRUE"]

TRUE

2373

>

> # 2) How many accidents had injuries

> # ANSWER: 6433

> injury<-tapply(accidentData$INJURY, accidentData$INJURY=='YES', length)

> injury["TRUE"]

TRUE

6433

>

> # 3) List the injuries by day

> # ANSWER: see console

> injurydays<-tapply(accidentData$INJURY,

+ list(accidentData$INJURY=='YES',accidentData$DAY\_OF\_WEEK), length)

> injurydays["TRUE",]

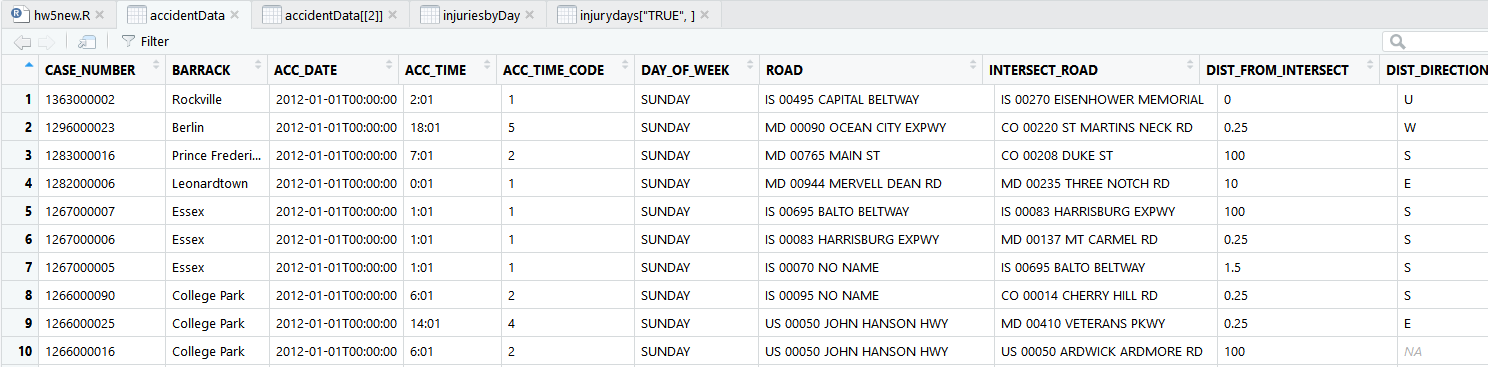
FRIDAY MONDAY SATURDAY SUNDAY THURSDAY TUESDAY WEDNESDAY

1043 915 950 818 968 843 896

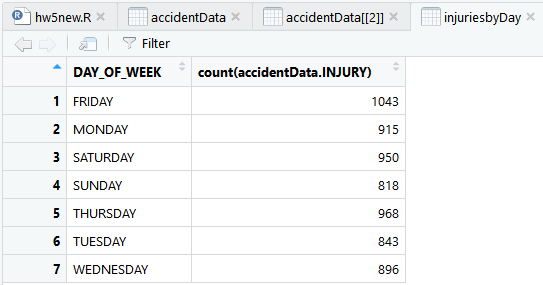
> View(injurydays["TRUE",])

## Other Artifacts

### Accident Data (from “View”)



### Injuries by Day (from “View”) – results of sqldf commands



### Injuries by Day (from “View”) – results from tapply commands

