DataPreparation

2/25/2020

Overview:

This exercise contains two datasets of format csv and one dataset in json format. Will do some basic necessary steps for data preparation. Details are as follows:

Excercise-1 The Open Data 500 is the first comprehensive study of U.S. companies that use open government data to generate new business and develop new products and services. Open Data is free, public data that can be used to launch commercial and nonprofit ventures, conduct research, make data-driven decisions, and solve complex problems [<https://www.opendata500.com/>] In this assignment you will be analyzing two datasets – “US\_agency.csv” and US\_companies.csv”. You can download the data from the website: <https://www.opendata500.com/>

Libraries Used in this two excercises for data preparation including r-essential bundle are

library(tidyverse)

## -- Attaching packages -------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ----------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(jsonlite)

##   
## Attaching package: 'jsonlite'

## The following object is masked from 'package:purrr':  
##   
## flatten

library(dplyr)

Import two datasets In order to emphasis the difference between read.csv and read\_csv used two methods.

data1 <- read.csv("us\_agencies.csv")  
data2 <- read.csv("us\_companies.csv")  
#data1 <- read\_csv("us\_agencies.csv")  
#data2 <- read\_csv("us\_companies.csv")

1. Structure of the datasets

str(data1)

## 'data.frame': 1123 obs. of 11 variables:  
## $ agency\_name : Factor w/ 70 levels "Administrative Office of the United States Courts",..: 63 2 3 4 19 19 19 19 19 19 ...  
## $ agency\_abbrev : Factor w/ 61 levels "","AO","AR","ATL",..: 1 3 4 5 18 18 18 18 18 18 ...  
## $ agency\_type : Factor w/ 5 levels "City/County",..: 3 5 1 1 2 2 2 2 2 2 ...  
## $ subagency\_name : Factor w/ 56 levels "Agricultural Marketing Service",..: 26 26 26 26 26 26 7 7 7 7 ...  
## $ subagency\_abbrev: Factor w/ 54 levels "","ACE","AMS",..: 1 1 1 1 1 1 9 9 9 9 ...  
## $ url : Factor w/ 112 levels "","http://catalog.data.gov/dataset?q=organization:((ecab-dol-gov)+OR+(whd-dol-gov)+OR+(esba-dol-gov)+OR+(ojc-dol-g"| \_\_truncated\_\_,..: 1 7 92 80 2 2 13 13 13 13 ...  
## $ used\_by : Factor w/ 526 levels "(Leg)Cyte","3 Round Stones, Inc.",..: 89 350 520 101 58 281 180 180 240 240 ...  
## $ used\_by\_category: Factor w/ 21 levels "","Aerospace and Defense",..: 12 9 11 12 9 4 17 17 9 9 ...  
## $ used\_by\_fte : Factor w/ 8 levels "1-10","1,001-5,000",..: 1 1 1 8 8 4 1 1 1 1 ...  
## $ dataset\_name : Factor w/ 507 levels "","18 State ALL Payer Hospital Dataset",..: 422 323 88 1 355 417 374 239 86 325 ...  
## $ dataset\_url : Factor w/ 277 levels "","asterweb.jpl.nasa.gov",..: 39 1 1 1 132 101 100 100 95 97 ...

str(data2)

## 'data.frame': 529 obs. of 22 variables:  
## $ company\_name\_id : Factor w/ 529 levels "3-round-stones-inc",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ company\_name : Factor w/ 529 levels "(Leg)Cyte","3 Round Stones, Inc.",..: 2 3 4 5 6 7 8 9 10 11 ...  
## $ url : Factor w/ 528 levels "abtassoc.com",..: 29 387 457 1 112 113 114 31 115 116 ...  
## $ year\_founded : int 2010 2014 2007 1965 1999 1989 1962 1969 2001 2009 ...  
## $ city : Factor w/ 203 levels ""," Philadelphia",..: 190 2 67 33 157 39 174 90 81 154 ...  
## $ state : Factor w/ 39 levels "AL","AR","AZ",..: 7 31 36 15 4 11 31 2 4 4 ...  
## $ country : Factor w/ 1 level "us": 1 1 1 1 1 1 1 1 1 1 ...  
## $ zip\_code : int 20004 19087 22003 2138 94583 60601 16803 72201 92618 95510 ...  
## $ full\_time\_employees: Factor w/ 8 levels "1-10","1,001-5,000",..: 1 8 1 2 7 3 5 6 4 3 ...  
## $ company\_type : Factor w/ 10 levels "","Nonprofit",..: 7 7 7 7 7 8 7 8 7 8 ...  
## $ company\_category : Factor w/ 21 levels "","Aerospace and Defense",..: 4 9 4 19 12 1 8 4 3 4 ...  
## $ revenue\_source : Factor w/ 101 levels "","Advertising",..: 38 54 98 27 72 72 88 72 72 72 ...  
## $ business\_model : Factor w/ 28 levels "","academia",..: 5 3 9 1 18 3 9 3 3 22 ...  
## $ social\_impact : Factor w/ 13 levels "","Citizen engagement and participation, Consumer empowerment",..: 1 13 1 1 6 1 1 1 1 1 ...  
## $ description : Factor w/ 529 levels " Lose It! helps you make smart, healthy eating choices that still leave room for your favorite foods. Lose It! "| \_\_truncated\_\_,..: 18 451 47 19 20 22 23 24 25 26 ...  
## $ description\_short : Factor w/ 529 levels "48 Factoring Inc. is one of the best financial services company using unique factoring 2.0 financial product wh"| \_\_truncated\_\_,..: 329 1 2 3 4 6 8 9 10 11 ...  
## $ source\_count : Factor w/ 5 levels "","1-10","101+",..: NA 4 NA 3 3 NA NA 3 NA 3 ...  
## $ data\_types : Factor w/ 39 levels "","Agriculture & Food",..: 1 8 1 1 31 30 13 1 1 1 ...  
## $ example\_uses : Factor w/ 9 levels "","FIVAR, one of our products, is a web application that uses real-time open and public data to make food inspecti"| \_\_truncated\_\_,..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ data\_impacts : Factor w/ 10 levels "[]","[u'Cost efficiency', u'Job growth', u'Revenue growth']",..: 1 2 1 1 1 1 1 1 1 1 ...  
## $ financial\_info : Factor w/ 143 levels "","$0 raised -- we are profitable\nOur business model supports a team of 50 (as of 12/10/13) and we are looking to"| \_\_truncated\_\_,..: 10 1 1 35 1 1 1 1 1 1 ...  
## $ last\_updated : Factor w/ 529 levels "2014-03-18 21:04:05.426443",..: 285 522 327 59 60 64 234 65 66 67 ...

1)No missing columns in both us\_agencies and us\_companies datasets.

1. colnames function

colnames(data1)

## [1] "agency\_name" "agency\_abbrev" "agency\_type"   
## [4] "subagency\_name" "subagency\_abbrev" "url"   
## [7] "used\_by" "used\_by\_category" "used\_by\_fte"   
## [10] "dataset\_name" "dataset\_url"

colnames(data2)

## [1] "company\_name\_id" "company\_name" "url"   
## [4] "year\_founded" "city" "state"   
## [7] "country" "zip\_code" "full\_time\_employees"  
## [10] "company\_type" "company\_category" "revenue\_source"   
## [13] "business\_model" "social\_impact" "description"   
## [16] "description\_short" "source\_count" "data\_types"   
## [19] "example\_uses" "data\_impacts" "financial\_info"   
## [22] "last\_updated"

no missing columns names or errors in the colummn names.

3)Finding missing values Initially converting empty column values to NA

data1[data1 == ""] <- NA  
sapply(data1,function(x) sum(is.na(x)))

## agency\_name agency\_abbrev agency\_type subagency\_name   
## 0 313 0 0   
## subagency\_abbrev url used\_by used\_by\_category   
## 709 292 0 4   
## used\_by\_fte dataset\_name dataset\_url   
## 44 548 808

sum(is.na(data1))

## [1] 2718

data2[data2 == ""] <- NA  
sapply(data2,function(x) sum(is.na(x)))

## company\_name\_id company\_name url   
## 0 0 0   
## year\_founded city state   
## 1 33 0   
## country zip\_code full\_time\_employees   
## 0 37 29   
## company\_type company\_category revenue\_source   
## 16 3 10   
## business\_model social\_impact description   
## 76 512 0   
## description\_short source\_count data\_types   
## 0 303 387   
## example\_uses data\_impacts financial\_info   
## 521 0 387   
## last\_updated   
## 0

sum(is.na(data2))

## [1] 2315

Us\_agencies (data1) has 2718 missing values us\_companies (data2) has 2315 missing values

4)Data is not organized in us\_agencies and us\_companies since some columns are not significant and too many categorical variables.

5)Data is not in good shape for further analysis. In dataset like us\_agenies: columns like dataset\_url has many missing values. used\_by\_fte data is not organized.

In datasetlike us\_companies: columns like full\_time\_ employees,source\_count,data\_impacts has unreadable characters and is not properly parsed. financial\_info has lot of missing values almost 73%.

6)Before doing any further analysis we need to do data preprocessing which involves steps like Finding: dulplicate values in dataset. Outliers missing values Correlation between variables.

#Finding Distinct values  
data1\_clean <- data1 %>% unique()  
data2\_clean <- data2 %>% unique()

Missing values can be impute by using some of the techniques as knn-imputation,mean,median.Some of the Libraries like Amelia,Hmisc,MICE can be used to deal with missing values.

1. There is primary key to connect two datasets.Having unique indentifier is needed for further analysis of two datasets. In my opinion dataset of us\_companies has column company\_name can be used as primary key which has unique value since in us\_agencies has used\_by column which is using this values.

Exercise-2 JSON (JavaScript Object Notation) is a most commonly used data format today and as a data scientist, you must know how to access JSON data sets. JSON is easy for machines to parse and generate. “It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent [JSON.ORG].” For this case study, you will parse JSON file, which has city traffic details. “Average Daily Traffic (ADT)” counts are analogous to a census count of vehicles on city streets. These counts provide a close approximation to the actual number of vehicles passing through a given location on an average weekday. Since it is not possible to count every vehicle on every city street, sample counts are taken along larger streets to get an estimate of traffic on half-mile or one-mile street segments. ADT counts are used by city planners, transportation engineers, real-estate developers, marketers and many others for myriad planning and operational purposes. Data Owner: Transportation. Time Period: 2006. Frequency: A citywide count is taken approximately every 10 years. A limited number of traffic counts will be taken and added to the list periodically [<https://catalog.data.gov/>]”.

datafile <- fromJSON("ChicagoTraffic.json")

23 Variables in our dataset

print(nrow(datafile$meta$view$columns))

## [1] 23

To name all variables in json file using

print(datafile$meta$view$columns$name)

## [1] "sid"   
## [2] "id"   
## [3] "position"   
## [4] "created\_at"   
## [5] "created\_meta"   
## [6] "updated\_at"   
## [7] "updated\_meta"   
## [8] "meta"   
## [9] "ID "   
## [10] "Traffic Volume Count Location Address"   
## [11] "Street"   
## [12] "Date of Count"   
## [13] "Total Passing Vehicle Volume"   
## [14] "Vehicle Volume By Each Direction of Traffic"  
## [15] "Latitude"   
## [16] "Longitude"   
## [17] "Location"   
## [18] "Boundaries - ZIP Codes"   
## [19] "Community Areas"   
## [20] "Zip Codes"   
## [21] "Census Tracts"   
## [22] "Wards"   
## [23] ":@computed\_region\_awaf\_s7ux"

3)Total traffic of vechicle on 100 to 115 street

traffic <- datafile$data  
  
for(i in 1:1279){  
 if(traffic[[i]][[11]] == "100th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "101th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "102th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "103th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "104th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "105th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "106th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "107th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "108th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "109th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "110th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "111th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "112th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "113th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "114th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
 if(traffic[[i]][[11]] == "115th St"){  
 print(as.numeric(traffic[[i]][[13]]))  
 }  
}

## [1] 8200  
## [1] 22600  
## [1] 8000  
## [1] 6300  
## [1] 9100  
## [1] 11000  
## [1] 12800  
## [1] 11700  
## [1] 15500  
## [1] 7900  
## [1] 12600  
## [1] 12200  
## [1] 16500  
## [1] 10900  
## [1] 16500  
## [1] 19800  
## [1] 6300  
## [1] 800  
## [1] 12200  
## [1] 20200  
## [1] 29800  
## [1] 7700  
## [1] 12800

1. total traffic of vehicles on geolocations, (41.651861, -87.54501) and (41.66836, -87.620176) is 13600

for(j in 1:1279){  
 if(traffic[[j]][[15]]=="41.651861" && traffic[[j]][[16]]== "-87.54501"){t1<-as.numeric(traffic[[j]][[13]])}  
 if(traffic[[j]][[15]]=="41.66836" && traffic[[j]][[16]]== "-87.620176"){t2<-as.numeric(traffic[[j]][[13]])}  
}  
Total\_Traffic=t1+t2  
print(Total\_Traffic)

## [1] 13600