IMT 573: Problem Set 2 - Data Wrangling

Naga Soundari Balamurugan

Due: Tuesday, October 16, 2018 11:59AM

Collaborators: Jayashree Raman

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problemset2.Rmd` file from Canvas. Open `problemset2.Rmd` in RStudio and supply your solutions to the assignment by editing `problemset2.Rmd`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. If you are using more than just a standard function that you found from another source, please credit the source in the comments. For example:
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF` or `Knit Word`, rename the R Markdown file to `YourLastName_YourFirstName_ps1.Rmd`, knit a PDF or DOC and submit both the PDF/DOC and the Rmd file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
library(tidyverse)
library(nycflights13)
library(jsonlite)
library(kableExtra)
```

Problem 1: Open Government Data

Use the following code to obtain data on the Seattle Police Department Police Report Incidents.

```
police_incidents <- fromJSON("https://data.seattle.gov/resource/hapq-73pk.json")

#To find the no of rows and columns in the dataset
no_of_rows <- nrow(police_incidents)
no_of_cols <- ncol(police_incidents)

#To find the dates for which the crime reports are available
unique(police_incidents$report_date)
```

```
## [1] "2008-01-01T00:00:00.000" "2008-02-01T00:00:00.000"
## [3] "2008-03-01T00:00:00.000" "2008-04-01T00:00:00.000"
```

(a) Describe, in detail, what the data represents.

The data contains the list of crimes reported at the police beats from Jan 1st of 2008 to Jan 4th of 2008 including the location, description of the crime etc., Each row in the dataset is a type of crime reported in a police beat. There are 1000 rows and 8 columns in total.

(b) Describe each variable and what it measures. Be sure to note when data is missing. Confirm that each variable is appropriately cast - it has the correct data type. If any are incorrect, recast them to be in the appropriate format.

```r
#List all the columns with data type
str(police_incidents)
```

```
## 'data.frame':    1000 obs. of  8 variables:
##  $ crime_description: chr  "Homicide" "Rape" "Robbery" "Assault" ...
##  $ crime_type       : chr  "Homicide" "Rape" "Robbery" "Assault" ...
##  $ police_beat      : chr  "B1" "B1" "B1" "B1" ...
##  $ precinct         : chr  "N" "N" "N" "N" ...
##  $ report_date      : chr  "2008-01-01T00:00:00.000" "2008-01-01T00:00:00.000" "2008-01-01T00:00:00.0
##  $ row_value_id     : chr  "1" "2" "3" "4" ...
##  $ sector           : chr  "B" "B" "B" "B" ...
##  $ stat_value       : chr  "0" "0" "5" "1" ...
```

```r
#To find the number of sectors in the dataset
#length(unique(police_incidents$sector))

#To find the number of police beats in the dataset
#length(unique(police_incidents$police_beat))

#To find the number of precincts in the dataset
#length(unique(police_incidents$precincts))
```

The variables in the dataset are listed below * crime_description = Description of the crime reported * crime_type = Type of the crime reported * police_beat = ID of the police beat where the crime was reported (for 51 police beats) * precinct = Boundary/limit to which the police beat belongs (The data includes precincts N-North, W-West, E-East, SE-SouthEast, SW-SouthWest) * report_date = Crime reported date * row_value_id = Unique id for each row * sector = Police sector to which the police beat belongs (17 sectors) *stat_value = Number of times crime occurred in a beat for reported month

All the variables in the dataset are of type character. As the row_value_id and stat_value denotes a number, lets recast them as integer. All the values in the report_date column has the time values as 00:00:00. Hence it is to be recasted as date datatype.

```r
#Change the dat type to integer
police_incidents$row_value_id <- as.integer(police_incidents$row_value_id)
police_incidents$stat_value <- as.integer(police_incidents$stat_value)
police_incidents$report_date <- as.Date.character(police_incidents$report_date)
```

(c) Produce a clean dataset, according to the rules of tidy data discussed in class. Export the data for future analysis using the Rdata format.

Lets rearrange the columns for more readability. There are 314 rows with stats_value as 0 which indicates no cases where filed in the corresponding police beat of that crime type. It is not sure that if the data is missing or is recorded as 0.

```
#Rearrange the column order
police_incidents <- police_incidents[c("row_value_id", "report_date", "police_beat",
                                       "precinct", "sector", "crime_type",
                                       "crime_description", "stat_value")]

#No of entries with 0 cases
nilCasesCount <- police_incidents %>% filter(stat_value == 0) %>% dplyr::summarise(total = n())

#Exporting the data in Rdata format
save(police_incidents, file = "police_incidents.RData")
```

(d) Describe any concerns you might have about this data. This may include biases, missing data, or ethical concerns.

One of the major concern is we do not know any background about the data. Also the data is available only for 4 days(Jan 1 to Jan 4 of 2008). With only such a small amount of data, no useful analysis can be done. Also as this data corresponds to the days close to new year eve, there might be less or more number of cases be filed than normal days.

Also sector wise, there is data available for 17 sectors where sector A, H, I, P, T and V are missing(As there could be only sectors upto W, I have mentioned only the missing alphabets). If there was no cases filed, then those sectors should also be listed with stat_value as 0.

Problem 2: Wrangling the NYC Flights Data

In this problem set we will use the data on all flights that departed NYC (i.e. JFK, LGA or EWR) in 2013. You can find this data in the nycflights13 R package.

(a) Importing Data:

Load the data.

```
#Loading flight data
#List all the functions in the nycflights13 package
ls("package:nycflights13")
```

```
## [1] "airlines" "airports" "flights"  "planes"   "weather"
```

(b) Data Manipulation:

Use the flights data to answer each of the following questions. Be sure to answer each question with a written response and supporting analysis.

- How many flights were there from NYC airports to Minneapolis/St.Paul in 2013?

  As the data contains only the list of flights departured from NYC airports in 2013, lets filter the data only based on destination. From the nycflights13::airports data, we can find the code for Minneapolis/St.Paul airport and is **MSP**.

```
#Save the flights data from nycflights13 package
flightsData <- nycflights13::flights
```

```r
#To find the number of flights from NYC to Minneapolis/St.Paul
noOfFlights <- flightsData %>% filter(dest == 'MSP') %>% dplyr::summarise(countFlights = n())
noOfFlights
```

```
## # A tibble: 1 x 1
##   countFlights
##          <int>
## 1         7185
```

There were **7185** flights from NYC airports to Minneapolis/St.Paul in 2013

- How many airlines fly from NYC to Minneapolis/St.Paul?

  carrier variable denotes the airlines and hence that should be used to explore the answer.

```r
#To find the no of airlines that provide service between NYC airports and Minneapolis/St.Paul
noOfAirlines <- flightsData %>% filter(dest == 'MSP') %>%
  dplyr::summarise(countAirlines = n_distinct(carrier))

noOfAirlines
```

```
## # A tibble: 1 x 1
##   countAirlines
##           <int>
## 1             6
```

There are **6** airlines that provide service between NYC airports and Minneapolis/St.Paul

- How many unique airplanes fly from NYC to Minneapolis/St.Paul?

  In order to find the number of unique planes, the flight column should be explored as it contains the flight number.

```r
#To find the no of unique airplanes that fly between NYC and Minneapolis/St.Paul
noOfPlanes <- flightsData %>% filter(dest == 'MSP') %>%
  dplyr::summarise(countPlanes = n_distinct(flight))

noOfPlanes
```

```
## # A tibble: 1 x 1
##   countPlanes
##         <int>
## 1         170
```

There are **170** unique planes that fly between NYC and Minneapolis/St.Paul.

- What is the average arrival delay for flights from NYC to Minneapolis/St.Paul?

  In order to find the average time delay, lets first filter the rows that has the destination as Mineapolis and the remove NAs so that we can calculate the average. Mean function is applied on the column arr_delay to calculate the average arrival time delay.

```r
#Find the average arrival delay for flights from NYC to Minneapolis/St.Paul
arrDelay <- flightsData %>% filter(dest == 'MSP') %>% na.omit() %>%
  dplyr::summarise(delay = mean(arr_delay))

arrDelay
```

```
## # A tibble: 1 x 1
##   delay
##   <dbl>
## 1  7.27
```

The average time delay is **7.27 minute** for flights from NYC to Minneapolis/St.Paul.

- What proportion of flights to Minneapolis/St.Paul come from each NYC airport? >As a first step, the data is filtered based on the destination. Then, as we need to find the propotion based on each NYC airport,the data is grouped by origin. Now a new column named ratio is mutated into dataset which has the ratio of flights(no of flights from each airport/total no of flights) from each airport.

  A histogram of this ratio is plotted using ggplot.

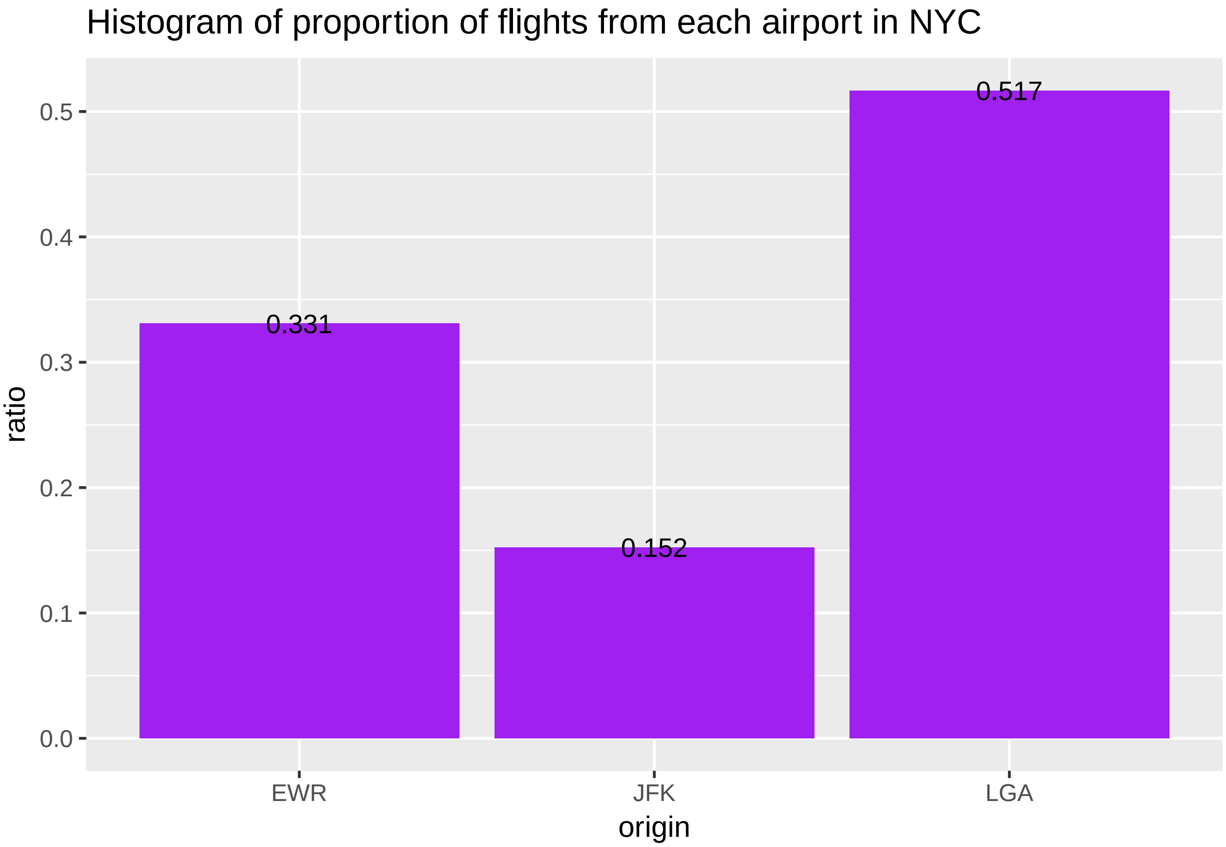
```
#To find the proportion of flights from each airport in NYC
airportProp <- flightsData %>% filter(dest == 'MSP') %>% group_by(origin) %>%
  dplyr::summarise(count = n())

airportProp <- airportProp %>% mutate(ratio = count/sum(count))
kable(airportProp, "latex") %>% kable_styling(bootstrap_options = c("striped", "hover"))
```

origin	count	ratio
EWR	2377	0.3308281
JFK	1095	0.1524008
LGA	3713	0.5167711

```
#Histogram of proportion of flights
proportionHist <- ggplot(data = airportProp, aes(x = origin, y = ratio)) +
  geom_bar(stat="identity", fill = "purple") +
  geom_text(aes(label=round(ratio, digits = 3)), color="black", size=3.5) +
  ggtitle("Histogram of proportion of flights from each airport in NYC")

proportionHist
```



Histogram of proportion of flights from each airport in NYC