INFX 573: Problem Set 2 - Data Wrangling

Shuyang Wu

Due: Monday, October 18, 2016

Collaborators:

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.
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`, rename the R Markdown file to `YourLastName_YourFirstName_ps2.Rmd`, knit a PDF and submit the PDF 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)
```

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/7ais-f98f.json")
```

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

```
#1a
head(police_incidents)
```

```
##   offense_code_extension   offense_type general_offense_number
## 1                      0         EQUALS             2016239258
## 2                      0   ASSLT-NONAGG             2016340018
## 3                      1 VEH-THEFT-AUTO             2016340045
## 4                      0 THEFT-SHOPLIFT             2016339816
## 5                      0   ASSLT-NONAGG             2016339898
## 6                      1 VEH-THEFT-AUTO             2016339682
##   offense_code rms_cdw_id year zone_beat     latitude
## 1         2903     949463 <NA>      <NA>         <NA>
## 2         1313    1038931 2016        E2 47.615837097
## 3         2404    1038930 2016        U1 47.667503357
## 4         2303    1038854 2016        L3 47.721984863
## 5         1313    1038866 2016        U2 47.659805298
## 6         2404    1038799 2016        N1 47.700145721
##   summarized_offense_description       date_reported
## 1                           <NA>                <NA>
## 2                       ASSAULT 2016-09-19T14:25:00
## 3                  VEHICLE THEFT 2016-09-19T13:21:00
## 4                    SHOPLIFTING 2016-09-19T12:14:00
## 5                       ASSAULT 2016-09-19T11:33:00
## 6                  VEHICLE THEFT 2016-09-19T10:19:00
##   occurred_date_or_date_range_start summary_offense_code month
## 1                              <NA>                 <NA>  <NA>
## 2               2016-09-19T13:00:00                 1300     9
## 3               2016-09-18T15:00:00                 2400     9
## 4               2016-09-19T10:12:00                 2300     9
## 5               2016-09-19T11:33:00                 1300     9
## 6               2016-09-17T17:00:00                 2400     9
##   census_tract_2000 location.latitude location.needs_recoding
## 1              <NA>              <NA>                     NA
## 2         7500.5009      47.615837097                  FALSE
## 3         4400.4003      47.667503357                  FALSE
## 4          100.5005      47.721984863                  FALSE
## 5         5301.3002      47.659805298                  FALSE
## 6         1400.3013      47.700145721                  FALSE
##   location.longitude         hundred_block_location district_sector
## 1               <NA>                           <NA>            <NA>
## 2     -122.31816864            16XX BLOCK OF 11 AV               E
## 3    -122.315200806         52XX BLOCK OF 12 AV NE               U
## 4    -122.293640137 127XX BLOCK OF LAKE CITY WY NE               L
## 5    -122.314323425      NE 43 ST / BROOKLYN AV NE               U
## 6    -122.366722107          8XX BLOCK OF NW 97 ST               N
##        longitude occurred_date_range_end
## 1           <NA>                   <NA>
## 2 -122.318168640                   <NA>
## 3 -122.315200806     2016-09-19T13:00:00
## 4 -122.293640137                   <NA>
## 5 -122.314323425                   <NA>
## 6 -122.366722107     2016-09-19T07:00:00
```

The data represents incidents happened around Seattle recorded in police reports when officers responded to them. Each row in the dataset represents a single incident/police report. Columns represent various information related to the incident. Specifically, offense type, offense code, offense code extension, general offense number, summarized offense description and summary offense code describe in detail the offense

happened in the incidence. Rms cdw id correspond to the original report in the Records Management System (RMS) which was then transmitted out to data.seattle.gov and generated this dataset. Year, month, date reported, occured data or date range start and occured data range end store data on the time that the incidence was reported and happened. Zone beat, latitude, longitude, hundred block location, district sector and census tract 2000 represent location and the precinct(zone/beat) of the incidents.

(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.

```
#1b
str(police_incidents) #check current data type
```

```
## 'data.frame':    1000 obs. of  19 variables:
##  $ offense_code_extension           : chr  "0" "0" "1" "0" ...
##  $ offense_type                     : chr  "EQUALS" "ASSLT-NONAGG" "VEH-THEFT-AUTO" "THEFT-SHOPLIFT"
##  $ general_offense_number           : chr  "2016239258" "2016340018" "2016340045" "2016339816" ...
##  $ offense_code                     : chr  "2903" "1313" "2404" "2303" ...
##  $ rms_cdw_id                       : chr  "949463" "1038931" "1038930" "1038854" ...
##  $ year                             : chr  NA "2016" "2016" "2016" ...
##  $ zone_beat                        : chr  NA "E2" "U1" "L3" ...
##  $ latitude                         : chr  NA "47.615837097" "47.667503357" "47.721984863" ...
##  $ summarized_offense_description   : chr  NA "ASSAULT" "VEHICLE THEFT" "SHOPLIFTING" ...
##  $ date_reported                    : chr  NA "2016-09-19T14:25:00" "2016-09-19T13:21:00" "2016-09-19
##  $ occurred_date_or_date_range_start: chr  NA "2016-09-19T13:00:00" "2016-09-18T15:00:00" "2016-09-19
##  $ summary_offense_code             : chr  NA "1300" "2400" "2300" ...
##  $ month                            : chr  NA "9" "9" "9" ...
##  $ census_tract_2000                : chr  NA "7500.5009" "4400.4003" "100.5005" ...
##  $ location                         :'data.frame':   1000 obs. of  3 variables:
##   ..$ latitude      : chr  NA "47.615837097" "47.667503357" "47.721984863" ...
##   ..$ needs_recoding: logi  NA FALSE FALSE FALSE FALSE FALSE ...
##   ..$ longitude     : chr  NA "-122.31816864" "-122.315200806" "-122.293640137" ...
##  $ hundred_block_location           : chr  NA "16XX BLOCK OF 11 AV" "52XX BLOCK OF 12 AV NE" "127XX B
##  $ district_sector                  : chr  NA "E" "U" "L" ...
##  $ longitude                        : chr  NA "-122.318168640" "-122.315200806" "-122.293640137" ...
##  $ occurred_date_range_end          : chr  NA NA "2016-09-19T13:00:00" NA ...
```

```
colnames(police_incidents)[colSums(is.na(police_incidents)) > 0] #columms that had missing value
```

```
##  [1] "year"
##  [2] "zone_beat"
##  [3] "latitude"
##  [4] "summarized_offense_description"
##  [5] "date_reported"
##  [6] "occurred_date_or_date_range_start"
##  [7] "summary_offense_code"
##  [8] "month"
##  [9] "census_tract_2000"
## [10] "location"
## [11] "hundred_block_location"
## [12] "district_sector"
## [13] "longitude"
```

```
## [14] "occurred_date_range_end"
## [15] NA
## [16] NA
```

```
#recast categorical variables into factor data type, continuous into numeric/integer
police_incidents$offense_code_extension <- as.factor(police_incidents$offense_code_extension)
police_incidents$location$needs_recoding <- as.logical(police_incidents$location$needs_recoding)
police_incidents$offense_code <- as.factor(police_incidents$offense_code)
police_incidents$rms_cdw_id <- as.factor(police_incidents$rms_cdw_id)
police_incidents$general_offense_number <- as.factor(police_incidents$general_offense_number)
police_incidents$year <- as.integer(police_incidents$year)
police_incidents$latitude <- as.numeric(police_incidents$latitude)
police_incidents$longitude <- as.numeric(police_incidents$longitude)
police_incidents$summary_offense_code <- as.factor(police_incidents$summary_offense_code)
police_incidents$month <- as.integer(police_incidents$month)
police_incidents$location$latitude <- as.numeric(police_incidents$location$latitude)
police_incidents$location$longitude <- as.numeric(police_incidents$location$longitude)
police_incidents$district_sector <- as.factor(police_incidents$district_sector)
str(police_incidents) #check final data type
```

```
## 'data.frame':    1000 obs. of  19 variables:
##  $ offense_code_extension           : Factor w/ 20 levels "0","1","18","2",..: 1 1 2 1 1 2 2 2 2 4 .
##  $ offense_type                     : chr  "EQUALS" "ASSLT-NONAGG" "VEH-THEFT-AUTO" "THEFT-SHOPLIFT"
##  $ general_offense_number           : Factor w/ 532 levels "2016239258","2016297802",..: 1 531 532 5
##  $ offense_code                     : Factor w/ 54 levels "1202","1203",..: 38 10 25 19 10 25 25 25 1
##  $ rms_cdw_id                       : Factor w/ 1000 levels "1036240","1036244",..: 1000 992 991 964
##  $ year                             : int  NA 2016 2016 2016 2016 2016 2016 2016 2016 2016 ...
##  $ zone_beat                        : chr  NA "E2" "U1" "L3" ...
##  $ latitude                         : num  NA 47.6 47.7 47.7 47.7 ...
##  $ summarized_offense_description   : chr  NA "ASSAULT" "VEHICLE THEFT" "SHOPLIFTING" ...
##  $ date_reported                    : chr  NA "2016-09-19T14:25:00" "2016-09-19T13:21:00" "2016-09-19
##  $ occurred_date_or_date_range_start: chr  NA "2016-09-19T13:00:00" "2016-09-18T15:00:00" "2016-09-19
##  $ summary_offense_code             : Factor w/ 20 levels "1200","1300",..: NA 2 6 5 2 6 6 6 6 5 ...
##  $ month                            : int  NA 9 9 9 9 9 9 9 9 9 ...
##  $ census_tract_2000                : chr  NA "7500.5009" "4400.4003" "100.5005" ...
##  $ location                         :'data.frame':   1000 obs. of  3 variables:
##   ..$ latitude      : num  NA 47.6 47.7 47.7 47.7 ...
##   ..$ needs_recoding: logi  NA FALSE FALSE FALSE FALSE FALSE ...
##   ..$ longitude     : num  NA -122 -122 -122 -122 ...
##  $ hundred_block_location           : chr  NA "16XX BLOCK OF 11 AV" "52XX BLOCK OF 12 AV NE" "127XX B
##  $ district_sector                  : Factor w/ 19 levels "99","B","C","D",..: NA 5 18 10 18 12 12 6
##  $ longitude                        : num  NA -122 -122 -122 -122 ...
##  $ occurred_date_range_end          : chr  NA NA "2016-09-19T13:00:00" NA ...
```

Offense type, offense code, offense code extension, general offense number, summarized offense description and summary offense code measure the types of offense happened in the incidences. Rms cdw id measures the original report in the Records Management System. Year, month, date reported, occured data or date range start and occured data range end measure time that the incidence was reported and time (period) that it happened. Zone beat, latitude, longitude, hundred block location, district sector and census tract 2000 measure location and the precinct(zone/beat) of the incidents.

Year, zone beat, latitude, summarized offense description, data reported, occured data or date range start, summary offense code, month, census tract 2000, location, hundred block location, district secror, longitude and occured data range end columns have missing value.

(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.

```
#1c
police_incidents_tidy <- police_incidents #duplicate original dataset
police_incidents_tidy$latitude <- NULL #remove duplicated columns
police_incidents_tidy$longitude <- NULL
police_incidents_tidy$hundred_block_location <- NULL #latitude and longitude data are sufficient to rep
police_incidents_tidy$offense_code[police_incidents_tidy$offense_code == "X"] <- NA #recode missing val
police_incidents_tidy$summary_offense_code[police_incidents_tidy$summary_offense_code == "X"] <- NA
save(police_incidents_tidy, file="police_incidents_tidy.RData")
```

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

This data can be a threat for privacy violation because it contains very detailed information of the location and the offense type. It'd be easy to find people living in residences where there was a reported incidence without knowing if the incidence actually represent anything about people who lives there or the safety of the neighborhood.

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.

```
head(nycflights13::flights)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     1     1      517            515         2      830
## 2  2013     1     1      533            529         4      850
## 3  2013     1     1      542            540         2      923
## 4  2013     1     1      544            545        -1     1004
## 5  2013     1     1      554            600        -6      812
## 6  2013     1     1      554            558        -4      740
## # ... with 12 more variables: sched_arr_time <int>, arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <time>
```

```
summary(nycflights13::flights)
```

```
##       year          month             day           dep_time
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   :   1
##  1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907
##  Median :2013   Median : 7.000   Median :16.00   Median :1401
##  Mean   :2013   Mean   : 6.549   Mean   :15.71   Mean   :1349
```

```
##  3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744
##  Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2400
##                                                  NA's   :8255
##  sched_dep_time   dep_delay          arr_time    sched_arr_time
##  Min.   : 106   Min.   : -43.00   Min.   :   1   Min.   :   1
##  1st Qu.: 906   1st Qu.:  -5.00   1st Qu.:1104   1st Qu.:1124
##  Median :1359   Median :  -2.00   Median :1535   Median :1556
##  Mean   :1344   Mean   :  12.64   Mean   :1502   Mean   :1536
##  3rd Qu.:1729   3rd Qu.:  11.00   3rd Qu.:1940   3rd Qu.:1945
##  Max.   :2359   Max.   :1301.00   Max.   :2400   Max.   :2359
##                 NA's   :8255      NA's   :8713
##    arr_delay          carrier              flight       tailnum
##  Min.   : -86.000   Length:336776      Min.   :   1   Length:336776
##  1st Qu.: -17.000   Class :character   1st Qu.: 553   Class :character
##  Median :  -5.000   Mode  :character   Median :1496   Mode  :character
##  Mean   :   6.895                      Mean   :1972
##  3rd Qu.:  14.000                      3rd Qu.:3465
##  Max.   :1272.000                      Max.   :8500
##  NA's   :9430
##     origin             dest              air_time        distance
##  Length:336776      Length:336776      Min.   : 20.0   Min.   :  17
##  Class :character   Class :character   1st Qu.: 82.0   1st Qu.: 502
##  Mode  :character   Mode  :character   Median :129.0   Median : 872
##                                        Mean   :150.7   Mean   :1040
##                                        3rd Qu.:192.0   3rd Qu.:1389
##                                        Max.   :695.0   Max.   :4983
##                                        NA's   :9430
##       hour           minute        time_hour
##  Min.   : 1.00   Min.   : 0.00   Min.   :2013-01-01 05:00:00
##  1st Qu.: 9.00   1st Qu.: 8.00   1st Qu.:2013-04-04 13:00:00
##  Median :13.00   Median :29.00   Median :2013-07-03 10:00:00
##  Mean   :13.18   Mean   :26.23   Mean   :2013-07-03 05:02:36
##  3rd Qu.:17.00   3rd Qu.:44.00   3rd Qu.:2013-10-01 07:00:00
##  Max.   :23.00   Max.   :59.00   Max.   :2013-12-31 23:00:00
##
```

(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 Seattle in 2013?

```
table(nycflights13::flights$year) #confirm all data are from 2013
```

```
##
##   2013
## 336776
```

```
table(nycflights13::flights$origin) #confirm all flights departed from NYC
```

```
##
##    EWR    JFK    LGA
## 120835 111279 104662
```

```
nyc_sea <- subset(nycflights13::flights, dest == "SEA") #Subset dataset to have only "SEA" as destinati
nrow(nyc_sea) #count number of flights (rows) in the new dataset
```

```
## [1] 3923
```

There were 3923 flights from NYC airports to Seattle.

- How many airlines fly from NYC to Seattle?

```
#count unique carriers
length(unique(nyc_sea$carrier))
```

```
## [1] 5
```

5 different airlines fly from NYC to Seattle.

- How many unique air planes fly from NYC to Seattle?

```
#count unique airplanes (combine values from carrier and flight to create new column)
nyc_sea$airplane<- with(nyc_sea, paste0(carrier, flight))
length(unique(nyc_sea$airplane))
```

```
## [1] 166
```

There were 166 unique airplanes flew from NYC to Seattle.

- What is the average arrival delay for flights from NYC to Seattle?

```
#calculate mean value for column(arr_delay)
class(nyc_sea$arr_delay) #check for correct data type
```

```
## [1] "numeric"
```

```
nyc_sea <- na.omit(nyc_sea) # remove mising values from df
mean(nyc_sea$arr_delay)
```

```
## [1] -1.099099
```

The average arrival delay was -1.099.

- What proportion of flights to Seattle come from each NYC airport? Out of all flights leaving from NYC to Seattle, 46.6% of them came from EWR, 53.4% came from JFK.

```
prop.table(table(nyc_sea$origin)) # proportion table for origin column
```

```
##
##       EWR       JFK
## 0.4658945 0.5341055
```