IMT 573: Problem Set 1 - Exploring Data

Chen Hsin Lee

Due: Tuesday, October 12, 2021 by 10am PT

Collaborators:

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

1. Download the `problemset1.Rmd` file from Canvas. Open `problemset1.Rmd` in RStudio and supply your solutions to the assignment by editing `problemset1.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. Collaboration shouldn't be confused with group project work (where each person does a part of the project). Working on problem sets should be your individual contribution. More on that in point 8.
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. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do not need four different visualizations of the same pattern.
4. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment. In particular, note that Stack Overflow is licenses as Creative Commons (CC-BY-SA). This means you have to attribute any code you refer from SO.
5. Partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. But please **DO NOT** submit pages and pages of hard-to-read code and attempts that is impossible to grade. That is, avoid redundancy. Remember that one of the key goals of a data scientist is to produce coherent reports that others can easily follow. Students are *strongly* encouraged to attempt each question and to document their reasoning process even if they cannot find the correct answer. If you would like to include R code to show this process, but it does not run withouth errors you can do so with the `eval=FALSE` option as follows:

```
a + b # these object dont' exist
# if you run this on its own it with give an error
```

7. 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 knitted PDF file to `ps1_ourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.
8. Collaboration is often fun and useful, but each student must turn in an individual write-up in their own words as well as code/work that is their own. Regardless of whether you work with others, what you turn in must be your own work; this includes code and interpretation of results. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.

Problem 1: Basic R Programming Write a function, `calculate_bmi` to calculate a person's body mass index, when given two input parameters, 1). weight in pounds and 2) height in inches.

NOTE: You would have to go to external sources to find the formula of bmi. In your response, before presenting your code for the function, tell us your official reference for the BMI formulate.

Insert Response first calculate_bmi<-function(weight,height){bmi<-weight/height*2;return(bmi)}

calculate_bmi(20,1.3)

```
calculate_bmi<-function(weight,height){bmi<-weight/height*2;return(bmi)}
calculate_bmi(20,1.3)
```

Insert code. Your code should appear within R Code Chunks.

```
## [1] 30.76923
```

Problem 2: Exploring 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.

Setup: Problem 2 You will need, at minimum, the following R packages. The data itself resides in package *nycflights13*. You may need to install both.

```
# Load standard libraries
library(tidyverse)
library('nycflights13')
```

```
# Load the nycflights13 library which includes data on all
# lights departing NYC
data(flights)
# Note the data itself is called flights, we will make it into a local df
# for readability
flights <- tbl_df(flights)
```

```
## Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
# Look at the help file for information about the data
# ?flights
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # ... with 336,766 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
# summary(flights)
```

(a) Importing Data Load the data and describe in a short paragraph how the data was collected and what each variable represents.

```
#I collected data from nycflights13, an existing database in package tidyvers.

#year:Year of departure
#month: Month of departure
#day:Day of departure
#dep_time: Actual departure time
#sched_dep_time: Scheduled departure time
#dep_delay: Departure delays, in minutes
#arr_time: Actual arrival time
#sched_arr_time: Scheduled arrival time
#arr_delay: Arrival delays, in minutes
#carrier: Two letter carrier abbreviation
#flight: Flight number
#tailnum: Plane tail number
#origin: Origin
#dest: Destination
#air_time: Amount of time spent in the air, in minutes
#distance: Distance between airports, in miles
#hour: Time of scheduled departure broken into hour
#minute: Time of scheduled departure broken into minutes
#time_hour: Scheduled date and hour of the flight as a POSIXct date
```

(b) Inspecting Data Perform a basic inspection of the data and discuss what you find. Inspections may involve asking the following questions (the list is not inclusive, you may well ask other questions):

- How many distinct flights do we have in the dataset?
- How many missing values are there in each variable?
- Do you see any unreasonable values? *Hint: Check out min, max and range functions.*

```
#1
distinct(flights,flight)
```

```
## # A tibble: 3,844 x 1
##    flight
##     <int>
##  1   1545
##  2   1714
##  3   1141
##  4    725
##  5    461
##  6   1696
##  7    507
##  8   5708
##  9     79
## 10    301
## # ... with 3,834 more rows
```

```
#There are 3844 distinct flights in the datasets.

#2
```

```
sum(is.na(flights))
```

```
## [1] 46595
```

```
sum(is.na(flights$year))
```

```
## [1] 0
```

```
sum(is.na(flights$month))
```

```
## [1] 0
```

```
sum(is.na(flights$day))
```

```
## [1] 0
```

```
sum(is.na(flights$dep_time))
```

```
## [1] 8255
```

```
sum(is.na(flights$sched_dep_time))
```

```
## [1] 0
```

```
sum(is.na(flights$dep_delay))
```

```
## [1] 8255
```

```
sum(is.na(flights$arr_time))
```

```
## [1] 8713
```

```
sum(is.na(flights$sched_arr_time))
```

```
## [1] 0
```

```
sum(is.na(flights$arr_delay))
```

```
## [1] 9430
```

```
sum(is.na(flights$carrier))
```

```
## [1] 0
```

```
sum(is.na(flights$flight))
```

```
## [1] 0
```

```
sum(is.na(flights$tailnum))
```

```
## [1] 2512
```

```
sum(is.na(flights$origin))
```

```
## [1] 0
```

```
sum(is.na(flights$dest))
```

```
## [1] 0
```

```
sum(is.na(flights$air_time))
```

```
## [1] 9430
```

```
sum(is.na(flights$distance))
```

```
## [1] 0
```

```
sum(is.na(flights$hour))
```

```
## [1] 0
```

```
sum(is.na(flights$minute))
```

```
## [1] 0
```

```
sum(is.na(flights$time_hour))
```

```
## [1] 0
```

```
#3
summary(flights)
```

```
##       year          month             day           dep_time    sched_dep_time
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   :   1   Min.   : 106
##  1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907   1st Qu.: 906
##  Median :2013   Median : 7.000   Median :16.00   Median :1401   Median :1359
##  Mean   :2013   Mean   : 6.549   Mean   :15.71   Mean   :1349   Mean   :1344
##  3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744   3rd Qu.:1729
##  Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2400   Max.   :2359
##                                                  NA's   :8255
##    dep_delay          arr_time    sched_arr_time   arr_delay
##  Min.   : -43.00   Min.   :   1   Min.   :   1   Min.   : -86.000
##  1st Qu.:  -5.00   1st Qu.:1104   1st Qu.:1124   1st Qu.: -17.000
##  Median :  -2.00   Median :1535   Median :1556   Median :  -5.000
##  Mean   :  12.64   Mean   :1502   Mean   :1536   Mean   :   6.895
##  3rd Qu.:  11.00   3rd Qu.:1940   3rd Qu.:1945   3rd Qu.:  14.000
##  Max.   :1301.00   Max.   :2400   Max.   :2359   Max.   :1272.000
##  NA's   :8255      NA's   :8713                  NA's   :9430
##    carrier              flight       tailnum            origin
##  Length:336776      Min.   :   1   Length:336776      Length:336776
##  Class :character   1st Qu.: 553   Class :character   Class :character
##  Mode  :character   Median :1496   Mode  :character   Mode  :character
##                     Mean   :1972
##                     3rd Qu.:3465
##                     Max.   :8500
##
##      dest              air_time        distance         hour
##  Length:336776      Min.   : 20.0   Min.   :  17   Min.   : 1.00
##  Class :character   1st Qu.: 82.0   1st Qu.: 502   1st Qu.: 9.00
##  Mode  :character   Median :129.0   Median : 872   Median :13.00
##                     Mean   :150.7   Mean   :1040   Mean   :13.18
##                     3rd Qu.:192.0   3rd Qu.:1389   3rd Qu.:17.00
##                     Max.   :695.0   Max.   :4983   Max.   :23.00
##                     NA's   :9430
##      minute        time_hour
##  Min.   : 0.00   Min.   :2013-01-01 05:00:00
##  1st Qu.: 8.00   1st Qu.:2013-04-04 13:00:00
##  Median :29.00   Median :2013-07-03 10:00:00
##  Mean   :26.23   Mean   :2013-07-03 05:22:54
##  3rd Qu.:44.00   3rd Qu.:2013-10-01 07:00:00
```

```
##  Max.   :59.00   Max.   :2013-12-31 23:00:00
##
```

```
#for the discrepancy of mean of sched_dep_time and dep_time is 5;however,the mean of dep_delay is 12.64
#for the discrepancy of mean of sched_arr_time and arr_time is 34;however, the mean of arr_delay is 6.8
#There are many NA's in the dep_time,dep_delay,arr_time and arr_delay.
```

(c) Formulating Questions Consider the NYC flights data. Formulate two motivating questions you want to explore using this data. Describe why these questions are interesting and how you might go about answering them.

Example questions:

- Which airport, JFK or LGA, experience more delays?
- What was the worst day to fly out?
- Are there seasonal patterns

```
#1 Are there any route most frequent to be delayed?
# It's interesting to see whether we can find the most frequent delayed route to solve the problem of d
# Check the distinction of orign and dest to see whether there are most frequent delayed routes.

#2 Are there any correlation between month and delaytime of arrival?
# It's interesting to understand whether we can find the seasonal correlation between moth and delaytim
# Check each month of the delaytime of arrival to see whether there is any specific month has the most
```

(d) Exploring Data For each of the questions you proposed in Problem 1c, perform an exploratory data analysis designed to address the question. Produce visualizations (graphics or tables) to answer your question. * You need to explore the data from the point of view of the questions * Depending on the question, you would need to provide precise definition. For example, what does "more delays" mean. * At a minimum, you should produce two visualizations (graphics or tables) related to each question. Be sure to describe what the visuals show and how they speak to your question of interest.

```
#1
newdata<-na.omit(flights)

newdata %>%
  group_by(origin,dest) %>%
  summarise(sumdelay=sum(arr_delay)) %>%

arrange('origin','dest',-sumdelay)
```

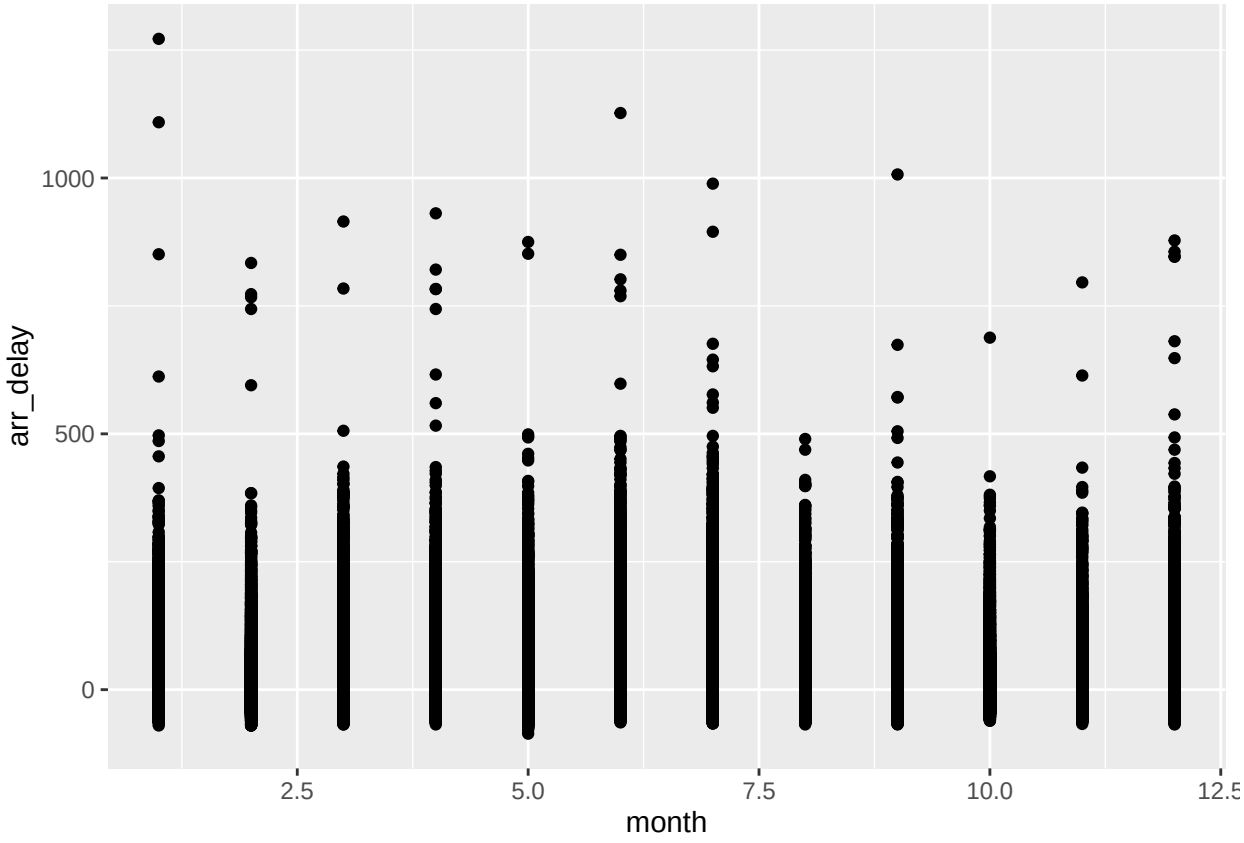
```
## `summarise()` has grouped output by 'origin'. You can override using the `.groups` argument.
```

```
## # A tibble: 223 x 3
## # Groups:   origin [3]
##    origin dest  sumdelay
##    <chr>  <chr>    <dbl>
##  1 LGA    ATL     113689
##  2 EWR    ATL      64525
##  3 EWR    ORD      52436
##  4 LGA    CLT      43991
##  5 EWR    RIC      41700
##  6 EWR    STL      38671
##  7 EWR    CLT      38077
##  8 LGA    FLL      37940
##  9 EWR    CVG      37508
## 10 LGA    BNA      34952
```

```
## # ... with 213 more rows
```

```
#most frequent delayed route means the routes of discrete origin and dest with their sum of arr_depay t

#2
ggplot(data = newdata ) + geom_point(mapping = aes(x = month, y=arr_delay))
```



(e) Challenge Your Results After completing the exploratory analyses from Problem 1d, do you have any concerns about your findings? How well defined was your original question? Do you have concerns regarding your answer? Is additional analysis/different data needed? Comment on any ethical and/or privacy concerns you have with your analysis.

```
#For my first exploring result, in route and delay_time table, I would like to get more information of

#For my second exploring result, in month and arr_delay table, I would like to get more information of
```

#For my first exploring result, in route and delay_time table, I would like to get more information of the origin and destination airport to understand why LGA and ATL route has the highest delay time.

#For my second exploring result, in month and arr_delay table, I would like to get more information of the specific month like Jan and Jun to see why these months have higher arr_delay time.