IMT 573: Problem Set 1 - Exploring Data

Jay Kuo

Due: Friday, October 14, 2022 by midnight PST

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 without errors you can do so with the `eval=FALSE` option as follows:

```
a + b # these objects don't 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 formulae.

Insert Response first

https://www.cdc.gov/nccdphp/dnpao/growthcharts/training/bmiage/page5_2.html

```
calculate_bmi <-function(pounds, inches)
  {return(pounds / (inches**2) *703)}
#test
calculate_bmi(154, 70)
```

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

```
## [1] 22.09429
```

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.
```

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

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      517        515       2     830     819      11 UA
##  2  2013     1     1      533        529       4     850     830      20 UA
##  3  2013     1     1      542        540       2     923     850      33 AA
##  4  2013     1     1      544        545      -1    1004    1022     -18 B6
##  5  2013     1     1      554        600      -6     812     837     -25 DL
##  6  2013     1     1      554        558      -4     740     728      12 UA
##  7  2013     1     1      555        600      -5     913     854      19 B6
##  8  2013     1     1      557        600      -3     709     723     -14 EV
##  9  2013     1     1      557        600      -3     838     846      -8 B6
## 10  2013     1     1      558        600      -2     753     745       8 AA
```

```
## # ... with 336,766 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
# summary(flights)
```

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

```
data <- flights
data
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      517        515       2     830     819      11 UA
##  2  2013     1     1      533        529       4     850     830      20 UA
##  3  2013     1     1      542        540       2     923     850      33 AA
##  4  2013     1     1      544        545      -1    1004    1022     -18 B6
##  5  2013     1     1      554        600      -6     812     837     -25 DL
##  6  2013     1     1      554        558      -4     740     728      12 UA
##  7  2013     1     1      555        600      -5     913     854      19 B6
##  8  2013     1     1      557        600      -3     709     723     -14 EV
##  9  2013     1     1      557        600      -3     838     846      -8 B6
## 10  2013     1     1      558        600      -2     753     745       8 AA
## # ... with 336,766 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
# The data was collected from airports information
# dep_time is the departure time in minutes and ached_dep_time is the scheduled departure times in minu
# sched_dep_time is the scheduled departure time in minutes
# arr_time is the arrival times in minutes
# sched_arr_time is the scheduled arrival times in minutes
# dep_delay means departure delay in minutes, if it's negative, then it means depart earlier than sched
# arr_delay is the same concept as dep_delay, except it is the delay of arrival instead of departure.
# carrier is the carrier abbreviation
# flight is the flight number
# tailnum is the plane tail number.
# origin, dest means origin and destination.
# air_time is the amount of time spent in the air in minutes
# distance is the distance between airports in miles
# hour, minute is time of scheduled departure broken into hour and minutes.
# time_hour is 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?

Ans: 3844

```
length(levels(factor(data$flight)))
```

```
## [1] 3844
```

- How many missing values are there in each variable? Ans: dep_time: 8255 dep_delay: 8255 arr_time: 8713 arr_delay: 9430 arr_time: 9430

```
summary(data)
```

```
##       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 05:00:00.00
##  1st Qu.: 8.00   1st Qu.:2013-04-04 13:00:00.00
##  Median :29.00   Median :2013-07-03 10:00:00.00
##  Mean   :26.23   Mean   :2013-07-03 05:22:54.64
##  3rd Qu.:44.00   3rd Qu.:2013-10-01 07:00:00.00
##  Max.   :59.00   Max.   :2013-12-31 23:00:00.00
##
```

- Do you see any unreasonable values? *Hint: Check out min, max and range functions.*

Ans: To me, an unreasonable value happened at arrival delay and departure delay. Because their mean and median are too different. ##### (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?

```
# When is Peak season and slack season in a year?
# Choose the origin column which is JFK, and sum its dep_delay & arr_delay and divided by the total of
JFKsum <- sum(data[data$origin == 'JFK', ]$dep_delay, na.rm = TRUE)+
  sum(data[data$origin == 'JFK', ]$arr_delay, na.rm = TRUE)
JFKsum <- JFKsum / length(data[data$origin == "JFK", ])
LGAsum <- sum(data[data$origin=='LGA', ]$dep_delay, na.rm = TRUE) +
  sum(data[data$origin == 'LGA', ]$arr_delay, na.rm = TRUE)
LGAsum <- LGAsum / length(data[data$origin == "LGA", ])
if (JFKsum > LGAsum){
  cat("JFK experience more delays and the average delay time is ", JFKsum, " minutes")
}else{
  cat("LGA experience more delays and the average delay time is ", LGAsum, " minutes")
}
```

```
## JFK experience more delays and the average delay time is  101621.8  minutes
```

- What was the worst day to fly out?

Ans: 3/8 has the max of daily total delay, so I think it is the worst day to fly out.

```
# The most delayed date might be the worst day to fly out.
data %>%
  group_by(month, day) %>%
  summarise(total_delay = sum(dep_delay, na.rm = TRUE) + sum(arr_delay, na.rm = TRUE)) %>%
  filter(total_delay == max(total_delay), na.rm = TRUE)
```

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

```
## # A tibble: 12 x 3
## # Groups:   month [12]
##    month   day total_delay
##    <int> <int>       <dbl>
##  1     1    31       51578
##  2     2    11       64221
##  3     3     8      135264
##  4     4    19       82556
##  5     5    23       86078
##  6     6    13       97104
##  7     7     1      100654
##  8     8     8       92421
##  9     9    12       85035
## 10    10     7       72458
## 11    11    17       37033
## 12    12     5       84236
```

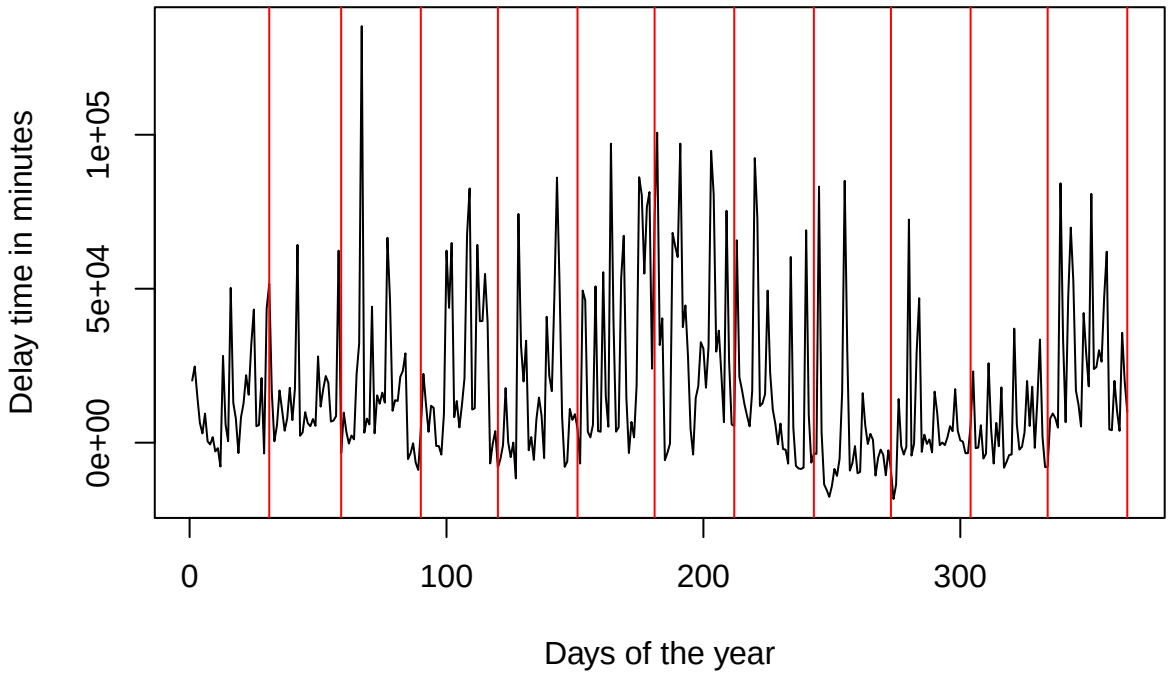
- Are there seasonal patterns to delays?

Ans: Yes, seems like summer (July and August) have more delays than usual. The peak happened at March. The red lines added separate the graph into different months.

```
day_data <- data %>%
  group_by(month, day) %>%
  summarise(total_delay = sum(dep_delay, na.rm = TRUE) + sum(arr_delay, na.rm = TRUE))
```

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

```
plot(day_data$total_delay, type = "l", xlab= "Days of the year", ylab= "Delay time in minutes")
 abline(v = 31, col = 'red')
 abline(v = 59, col = 'red')
 abline(v = 90, col = 'red')
 abline(v = 120, col = 'red')
 abline(v = 151, col = 'red')
 abline(v = 181, col = 'red')
 abline(v = 212, col = 'red')
 abline(v = 243, col = 'red')
 abline(v = 273, col = 'red')
 abline(v = 304, col = 'red')
 abline(v = 334, col = 'red')
 abline(v = 365, col = 'red')
```



“‘

(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 will need to provide a more 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.

Ans: The visualization is shown above in each problem.

(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.

Ans: I am not sure about my finding regarding the seasonal patterns of delay. Because based on the plot I

created, the difference is subtle, and I am wondering if there is any other way to see the seasonal patterns.