IMT 573: Problem Set 3

Working with Data: Part I

Tommy Huynh

Due: July 11, 2021

Collaborators:

Instructions: Before beginning this assignment, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Cloud.

1. Download the `03_ps_workingdata.Rmd` file from Canvas or save a copy to your local directory on RStudio Cloud. Supply your solutions to the assignment by editing `03_ps_workingdata.Rmd`.
2. Replace the "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. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do no need four different visualizations of the same pattern.
4. Collaboration on problem sets is fun and useful, and we encourage it, 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.
5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.
6. Remember partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. 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 object 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`, download and rename the knitted PDF file to `ps3_YourLastName_YourFirstName.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)
```

```
## Warning: package 'tidyr' was built under R version 4.1.1
```

```
## Warning: package 'readr' was built under R version 4.1.1
```

```
library(nycflights13)
```

```
## Warning: package 'nycflights13' was built under R version 4.1.1
```

```
library(dplyr)
library(scales)
```

```
## Warning: package 'scales' was built under R version 4.1.1
```

```
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.1.1
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.1.1
```

Problem 1: Describing the NYC Flights Data In this problem set we will continue to use the data on all flights that departed NYC (i.e. JFK, LGA or EWR) in 2013. Recall, you can find this data in the `nycflights13` R package. Load the data in R and ensure you know the variables in the data. Keep the documentation of the dataset (e.g. the help file) nearby.

```
# 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)
```

In Problem Set 1 you started to explore this data. Now we will perform a more thorough description and summarization of the data, making use of our new data manipulation skills to answer a specific set of questions. When answering these questions be sure to include the code you used in computing empirical responses, this code should include code comments. Your response should also be accompanied by a written explanation, code alone is not a sufficient response.

(a) Describe and Summarize Answer the following questions in order to describe and summarize the `flights` data.

How many flights out of NYC are there in the data?

```
#This code subsets the flights data into flights that
#are originating from NYC, it then counts the number
#of rows in the new dataframe
nyc_flights <- subset(flights, origin == "JFK" | origin == "LGA" | origin == "EWR")

nyc_originating <- nrow(nyc_flights)
print(nyc_originating)
```

```
## [1] 336776
```

How many NYC airports are included in this data? Which airports are these?

```
#There are three NYC airports included in this data.
#John F. Kennnedy, Laguardia, and Newark.
```

Into how many airports did the airlines fly from NYC in 2013?

```
#This code counts the number of unique airports in the dataframes
#destination column.Flights out of NYC went to 105 different
#airports in 2013
sum(!duplicated(nyc_flights$dest))
```

```
## [1] 105
```

How many flights were there from NYC to Seattle (airport code SEA)?

```
#This code subsets the nyc_flights dataframe into a dataframe that
#contains only flights to Seattle. It then counts the number of rows
#in the new data frame. There were a total of 3923 flights from NYC
#to Seattle in 2013.
nyc_sea <- subset(nyc_flights, dest == "SEA")
nrow(nyc_sea)
```

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

Were the any flights from NYC to Spokane (GAG)?

```
#This code subsets the nyc_flights dataframe into a dataframe
#that contains only flights to Spokane. It then counts the rows
#in the new data frame. There were no direct flights from NYC to
#Spokane in 2013.
nyc_geg <- subset(nyc_flights, dest == "GEG")
nrow(nyc_geg)
```

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

What about missing destination codes? Are there any destinations that do not look like valid airport codes (i.e. three-letter-all-upper case)?

```
#There are no missing destination codes.
```

(b) Reflect and Question Comment the questions (and answers) so far. Were you able to answer all of these questions? Are all questions well defined? Is the data good enough to answer all these?

```
#So far, the data is sufficient to answer the questions.
#I think that all the questions were well defined.
```

Problem 2: NYC Flight Delays Flights are often delayed. Let's look at closer at this topic using the NYC Flight dataset. Answer the following questions about flight delays using the `dplyr` data manipulation verbs we talked about in class.

(a) Typical Delays What is the typical delay of flights in this data?

```
#This function calculates the mean arrival delay of all
#flights departing from NYC.This mean calculation excludes
#NA values. The average arrival delay for flights out of
#New York City was about 6.8 minutes.
mean(nyc_flights$arr_delay + nyc_flights$dep_delay, na.rm=TRUE)
```

```
## [1] 19.45053
```

(b) Defining Flight Delays What definition of flight delay did you use to answer part (a)? Did you do any specific exploration and description of this variable prior to using it? If no, please do so now. Is there any missing data? Are there any implausible or invalid entries?

```
#I used the definition of the flight arriving and departing
#to its destination late.There are missing values that state NA,
#but I used the na.rm feature of the mean function to exclude those values.
```

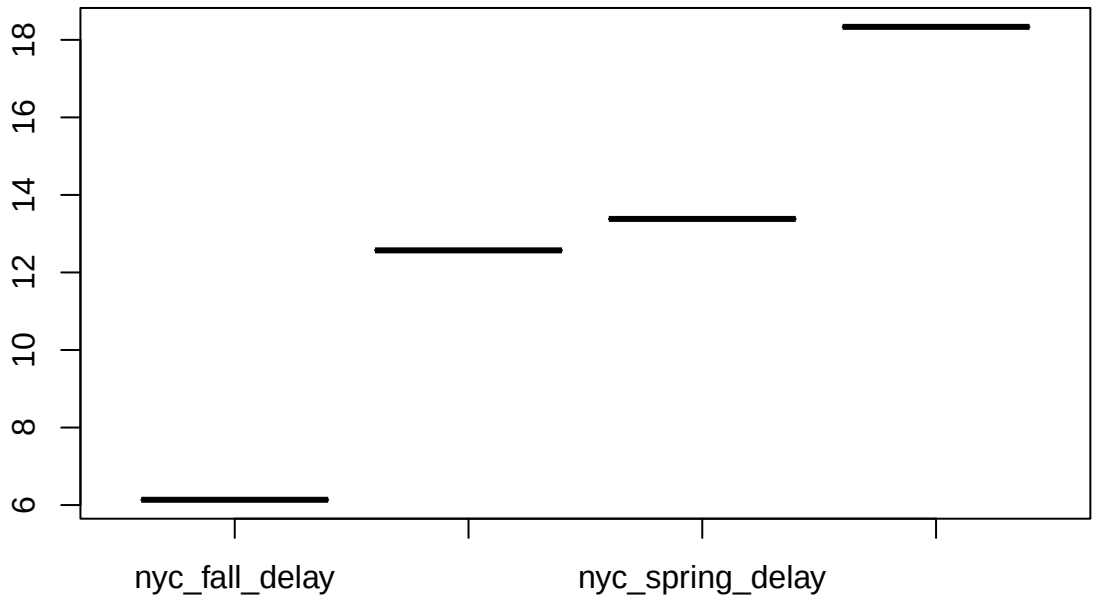
(b) Delays by Destination Now compute flight delay by destinations. Which ones are the worst three destinations from NYC if you don't like flight delays? Be sure to justify your delay variable choice.

```
#This code creates a new column with total delay time
#(departing and arrival). It then aggregates by destination
#airport for a total delay time of all flights.
#Airport codes SNA, PSP, and LEX had the most total delay minutes.
nyc_flights <- cbind(nyc_flights, total = nyc_flights$arr_delay + nyc_flights$dep_delay)
nyc_flights_delays <- aggregate(nyc_flights$total, by=list(dest=nyc_flights$dest), FUN=sum, na.rm = TRU

nyc_flights_delays <- nyc_flights_delays[order(nyc_flights_delays$x, decreasing = FALSE),]

head(nyc_flights_delays)
```

```
##    dest    x
## 96  SNA -883
## 78  PSP -282
## 51  LEX  -31
## 52  LGA    0
## 4   ANC   83
## 35  EYW  170
```

(b) Seasonal Delays Flight delays may be partly related to weather, as you might have experienced for yourself. We do not have weather information here but let's analyze how it is related to season. Which seasons have the worst flights delays? Why might this be the case? In your communication of your analysis use one graphical visualization and one tabular representation of your findings.

```r
#This code calculates the mean departure delay time for each
#season by the corresponding numerical month. It then creates a
#new dataframe with the means of all the seasons in order to create a box plot.
#It appears that winter has the most delays out of the four seaons.
nyc_fall <- subset(nyc_flights, month == '9' | month == '10' | month == '11')
nyc_fall_delay <- mean(nyc_fall$dep_delay, na.rm = TRUE)

nyc_winter <-subset(nyc_flights, month == '12' | month == '1' | month == '2')
nyc_winter_delay <- mean(nyc_winter$dep_delay, na.rm = TRUE)

nyc_spring <- subset(nyc_flights, month == '3' | month == '4' | month == '5')

nyc_spring_delay <- mean(nyc_spring$dep_delay, na.rm = TRUE)

nyc_summer <- subset(nyc_flights, month == '6' | month == '7' | month == '8')
nyc_summer_delay <- mean(nyc_summer$dep_delay, na.rm = TRUE)

nyc_seasonal_delay <- data.frame(nyc_fall_delay, nyc_winter_delay, nyc_spring_delay, nyc_summer_delay)

boxplot(nyc_seasonal_delay)
```



(d) Challenge Your Results After completing the exploratory analyses from Problem 2, do you have any concerns about your findings? How well defined was your original question? Do you still believe this

question can be answered using this dataset? Comment on any ethical and/or privacy concerns you have with your analysis.

```
#I do not have any concerns regarding my findings. I think all of the questions are
#well defined and that all the questions can be properly answered with the data set.
#I do not see any ethifcal or privacy concerns with my analysis because the data is
#essentially public information and historical.
```

Problem 3: Let's Fly to Across the Country!

(a) Describe and Summarize Answer the following questions in order to describe and summarize the `flights` data, focusing on flights from New York to Portland, OR (airport code `PDX`).

How many flights were there from NYC airports to Portland in 2013?

```
#This code subsets the nyc_flights dataframe into a dataframe that
#only contains flights between NYC and Portland. It then calculates the
#number of rows giving us the total number of flights which is 1354.
nyc_pdx <- subset(nyc_flights, dest == "PDX")
nrow(nyc_pdx)
```

```
## [1] 1354
```

How many airlines fly from NYC to Portland?

```
#This code takes all carriers that fly from NYC to Portland and removes
#all duplicates. It then converts it to a dataframe and sums up the number
#of rows, giving us the number of carriers that fly from NYC to Portland which is 3.
airlines <- nyc_pdx[!duplicated(nyc_pdx$carrier),]
airlines <- airlines$carrier
airlines <- as.data.frame(airlines)

nrow(airlines)
```

```
## [1] 3
```

Which are these airlines (find the 2-letter abbreviations)? How many times did each of these go to Portland?

```
#This code prints the airline 2 letter abbreviation. It then counts the
#number of flights by each carrier. From NYC to PDX, B6 flew 325 flights,
#Delta flew 458 flights, and United flew 571 flights.
print(airlines)
```

```
##   airlines
## 1       DL
## 2       UA
## 3       B6
```

```
flight_count <- count(nyc_pdx, vars = carrier)
print(flight_count)
```

```
##   vars   n
## 1   B6 325
## 2   DL 458
## 3   UA 571
```

How many unique airplanes fly from NYC to PDX? {Hint: airplane tail number is a unique identifier of an airplane.}

```
#This code calculates how many flights are flown per plane,
#it then counts the number of rows to determine the number
#of unique planes that are flown on this route. There were
#a total of 492 unique planes.
plane_count <- count(nyc_pdx, vars = tailnum)
nrow(plane_count)
```

```
## [1] 492
```

How many different airplanes arrived from each of the three NYC airports to Portland?

```
#This code subsets each airport into different data frames.
#It then counts the number of unique tailnumbers in each
#dataframe and returns the amount. There were 195 unique planes from JFK and 297 unique planes from EWR
jfk_pdx <- subset(nyc_pdx, origin == "JFK")
jfk_count <- count(jfk_pdx, vars = tailnum)
nrow(jfk_count)
```

```
## [1] 195
```

```
ewr_pdx <- subset(nyc_pdx, origin == "EWR")
ewr_count <- count(ewr_pdx, vars = tailnum)
nrow(ewr_count)
```

```
## [1] 297
```

What percentage of flights to Portland were delayed at departure by more than 15 minutes?

```
#This code subsets the nyc_pdx data frame into a data frame that only contains
#flights that were delayed at departure by more than 15 minutes.
#It then calculates the percentage by dividing the number of rows in the
#new data frame by the nyc_pdx frame. The percentage of flights
#that were delayed at departure by more than 15 minutes is 27%
delay_pdx_15 <- subset(nyc_pdx, dep_delay > 15)
delay_pdx <- percent(nrow(delay_pdx_15) / nrow(nyc_pdx))
print(delay_pdx)
```

```
## [1] "27%"
```

Is one of the New York airports noticeably worse in terms of departure delays for flights to Portland, OR than others?

```
#This code subsets the data for JFK and EWR.
#It the ncalculates the mean departure
#delay for each airport. They are roughly the same.
delay_jfk <- subset(nyc_pdx, origin == "JFK")
print(mean(delay_jfk$dep_delay, na.rm = TRUE))
```

```
## [1] 16.02945
```

```
delay_ewr <- subset(nyc_pdx, origin == "EWR")
print(mean(delay_ewr$dep_delay, na.rm = TRUE))
```

```
## [1] 16.5679
```

(b) Reflect and Question Comment the questions (and answers) in this analysis. Were you able to answer all of these questions? Are all questions well defined? Is the data good enough to answer all these?

```
#Yes, I was able to answer all of the questions. I think that
#the data was good enough to answer all the questions
```