IMT 573: Problem Set 2 - Working with Data

Stephen V Tucker

Due: Friday, October 21, 2022

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. 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 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 dont' exist
# if you run this on its own it with give an error
```

6. 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 `ps2_ourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

7. 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.

Setup

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

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

data(flights)
flights <- tibble(flights)


# Item  // \item How many flights out of NYC are there in the data?
## 336776 flights out of NYC airport
flights_out <- flights %>%
  summarise(origin)
count(flights_out)

## # A tibble: 1 × 1
##        n
##    <int>
## 1 336776

#
#\item How many NYC airports are included in this data?  Which airports are
these?
# Newark, Laguardia John F kennedy,
# # "EWR" "LGA" "JFK"
nyc_airport <- unique(flights$origin) %>%
  print()

## [1] "EWR" "LGA" "JFK"

# \item Into how many airports did the airlines fly from NYC in 2013?
# There were 105 flights that flew out of NYC
nyc_airport_dest <- unique(flights$dest)
print(length(nyc_airport_dest))

## [1] 105

# \item How many flights were there from NYC to Seattle (airport code
\texttt{SEA})?
```

```
## there were 3923 from NYC to Seattle
seattle_flights <- flights %>%
  filter(dest == "SEA") %>%
  group_by(total = length(dest))

print(length(seattle_flights$total))

## [1] 3923

#\item Were the any flights from NYC to Spokane \texttt{(GAG)}?
# According to the data no there were no flights from NYC to GAG
spokane_flights <- flights %>%
  filter(dest == "GAG")

print(nrow(spokane_flights))

## [1] 0

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

### # No they all look valid
nyc_airport_dest <- flights
nyc_airport_dest <- nyc_airport_dest[!duplicated(nyc_airport_dest$dest),]
```

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.

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, along with code comments. Your response should also be accompanied by a written explanation, as code alone is not a sufficient response.

1(a) Describe and Summarize

Answer the following questions in order to describe and summarize the `flights` data.

Hint:check the function `grepl` to do regular expression matching. You may use `"^[[:upper:]]{3}$"` for a regular expression that matches three upper case letters. See an example below:

```
grepl("^[[:upper:]]{3}$", c("12AB", "SEA", "ABCD", "ATL"))

# [1] FALSE  TRUE FALSE  TRUE
```

1(b) Reflect and Question

Comment on 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?

Problem 2: NYC Flight Delays

Flights are often delayed. Let's look 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.

2(a) Typical Delays

What is the typical delay of flights in this data?

```
## typical delay is 19.4505. i choose to go for the avergae becaus I perceive typical as average.
print(mean(flights$arr_delay + flights$dep_delay, na.rm=TRUE))

## [1] 19.45053
```

2(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?
the definition of flight delay i used was taking the avg of both departure and arrivals delay, and finding cumulative average.

2(c) Delays by Destination

Now compute flight delay by destination. 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 question isnt straightforward since there is a large variation of total amount of destinations from place to place. But I will attempt to get a straight forward answer.

```
# Step one
flight_delay_dest_nyc <- flights %>%
  drop_na() %>%
  select(dest,dep_delay,arr_delay) %>%
  mutate(total_delay = dep_delay + arr_delay) %>%
  select(dest, total_delay)

# Step Two
delay_dest_nyc <- aggregate(flight_delay_dest_nyc$total_delay, by = list(dest
=flight_delay_dest_nyc$dest), FUN=sum, na.rm = TRUE)

# Step Three
delay_dest_nyc <- delay_dest_nyc[order(delay_dest_nyc$x, decreasing = TRUE),]

print(head(delay_dest_nyc))
```

```
##    dest      x
## 5   ATL 399462
## 69  ORD 319880
## 36  FLL 247052
## 54  MCO 233528
## 24  CLT 226397
## 90  SFO 203017
```

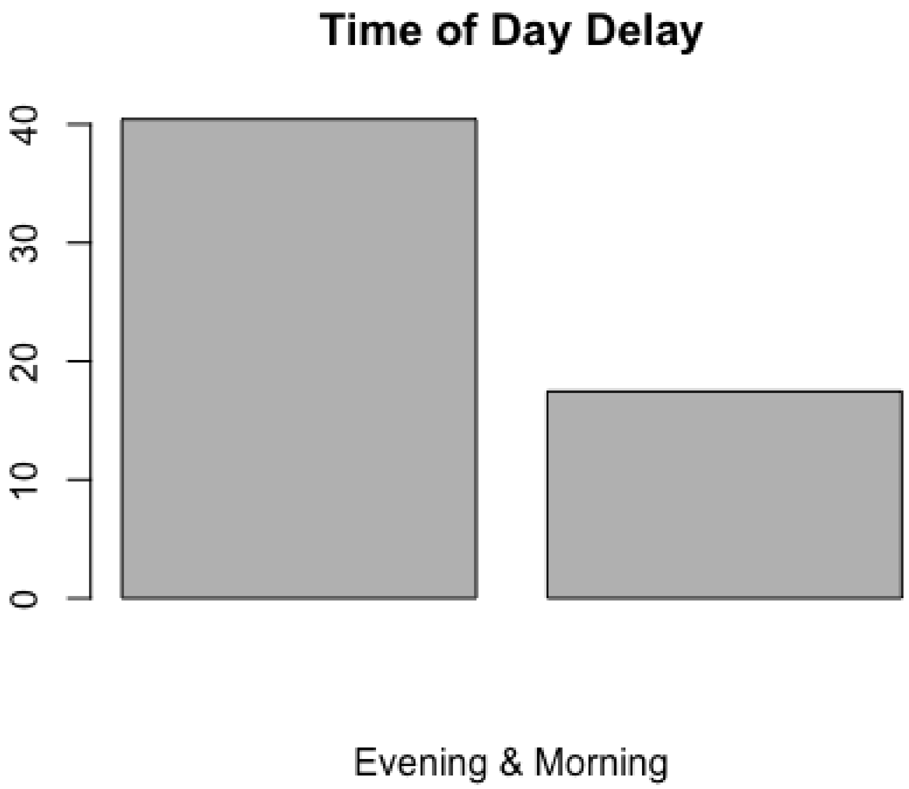
```
## Atlanta has the longest flight delay destination at 399462
```

2(d) Delays by time of day

We'd like to know how much delays depend on the time of day. Are there more delays in the mornings? Late night when all the daily delays may accumulate? Create a visualization (graph or table) to illustrate your findings.

```
## There are more delays in the evening.
#
morning <- flights %>%
  drop_na() %>%
  mutate(bin_a = cut(hour, breaks = c(6, 18))) %>%
  drop_na()

# Bin 2
evening <- flights %>%
  drop_na() %>%
  mutate(bin_b = cut(hour, breaks = c(19, 23))) %>%
  drop_na()

#
morning <- morning %>%
  drop_na() %>%
  group_by(delay_avg = mean(dep_delay + arr_delay)) %>%
  select(year,delay_avg)

print( morning[!duplicated(morning$delay_avg),] )
```

```
## # A tibble: 1 × 2
## # Groups:   delay_avg [1]
##    year delay_avg
##   <int>     <dbl>
## 1  2013      17.5

## bin 2 agg
evening <- evening  %>%
  drop_na() %>%
  group_by(delay_avg = mean(dep_delay + arr_delay)) %>%
  select(year,delay_avg)

print( evening[!duplicated(evening$delay_avg),] )

## # A tibble: 1 × 2
## # Groups:   delay_avg [1]
##    year delay_avg
##   <int>     <dbl>
## 1  2013      40.4

#
morning_evening_join <- full_join(evening,morning)

## Joining, by = c("year", "delay_avg")

morning_evening_join <-
morning_evening_join[!duplicated(morning_evening_join$delay_avg),]


counts_2 <- (morning_evening_join$delay_avg)


# bar_plot
barplot(counts_2,beside = FALSE, main = "Time of Day Delay",
        xlab = "Evening & Morning")
```



```
# links: https://www.tutorialspoint.com/how-to-create-bins-for-a-continuous-
vector-in-r
# https://www.statology.org/data-binning-in-r/
```

2(e) Reflect and Challenge Your Results

After completing the exploratory analyses from Problem 2, do you have any concerns about these questions and your findings? How well defined were the questions? If you feel a question is not defined well enough, re-formulate it in a more specific way so you can actually answer this question. And state clearly what is your more precise question.
Can you formulate any additional questions regarding flight delays?

Reflection Response

My one question and concern is more about the definition of mornings and evening or late. As well as why the delay exist is it because less workers and at the time or more people flying out? How do we find out the root cause of the delays?

Problem 3: Let's Fly Across the Country!

3(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`).

3(b) Reflect and Question

Comment on 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 successfully the data was good enough to answer all questions.

Yes all the questions are well defined

Extra Credit

Seasonal Delays

Let's get back to the question of flight 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.

```
##  It appears that summer has the most delays i decided to use the total of
departure delays and arrival delays to get a gross count, It may be more
useful to get overall average but the question said experience so i wanted to
get a total amount to capture the experience in totality Though average is a
way to consider as well which would of been insightful as well.


# Bin 1 - months 12- 1,2 Winter
seasonal_delay_b1 <- flights %>%
  mutate(bin_1 = cut(month, breaks = c(0,2))) %>%
  drop_na(bin_1)

seasonal_delay_b1_2 <- flights %>%
  mutate(bin_1_2 = cut(month, breaks = c(11,12))) %>%
  drop_na(bin_1_2)

joined_b1 <- full_join(seasonal_delay_b1, seasonal_delay_b1_2)

## Joining, by = c("year", "month", "day", "dep_time", "sched_dep_time",
## "dep_delay", "arr_time", "sched_arr_time", "arr_delay", "carrier",
"flight",
## "tailnum", "origin", "dest", "air_time", "distance", "hour", "minute",
## "time_hour")
```

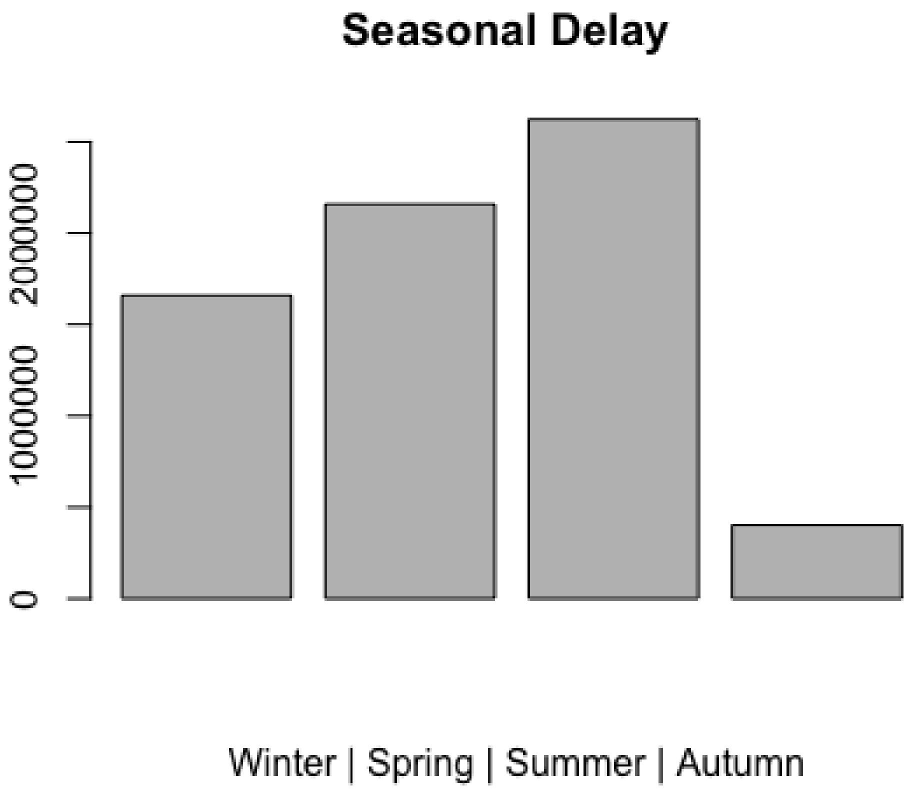
```r
## Spring Delay
spring <- flights %>%
  mutate(bin_2 = cut(month, breaks = c(3,6))) %>%
  drop_na(bin_2)

## Summer

seasonal_delay_b3 <- flights %>%
  mutate(bin_3 = cut(month, breaks = c(6,9))) %>%
  drop_na(bin_3)

##

seasonal_delay_b4 <- flights %>%
  mutate(bin_4 = cut(month, breaks = c(9,12))) %>%
  drop_na(bin_4)

#### IVI
# Add up total delay

## Winter

seasonal_delay_1 <- seasonal_delay_b1 %>%
  drop_na() %>%
  mutate(total_delay_1 = sum(dep_delay + arr_delay)) %>%
  filter(!duplicated(total_delay_1))

winter_delay <- joined_b1  %>%
  select(year,month,dep_delay,arr_delay) %>%
  drop_na() %>%
  mutate(total_delay = sum(dep_delay + arr_delay)) %>%
  mutate(season = c("winter")) %>%
  select(year,total_delay,season) %>%
  filter(! duplicated(total_delay))

print(winter_delay) # 1659150 3
```

```
## # A tibble: 1 × 3
##    year total_delay season
##   <int>       <dbl> <chr>
## 1  2013     1659150 winter
```

```r
## Spring Delay
spring <- flights %>%
  mutate(bin_2 = cut(month, breaks = c(3,6))) %>%
  drop_na(bin_2)
```

```
spring_delay <- spring %>%
  select(year,month,dep_delay,arr_delay) %>%
  drop_na() %>%
  mutate(total_delay = sum(dep_delay + arr_delay)) %>%
  mutate(season = c("spring")) %>%
  select(year,total_delay,season) %>%
  filter(! duplicated(total_delay))

print(spring_delay) #2158845    2

## # A tibble: 1 × 3
##    year total_delay season
##   <int>       <dbl> <chr>
## 1  2013     2158845 spring

## summer


summer <- flights %>%
  mutate(bin_3 = cut(month, breaks = c(5,8))) %>%
  drop_na(bin_3)

summer_delay <- summer %>%
  select(year,month,dep_delay,arr_delay) %>%
  drop_na() %>%
  mutate(total_delay = sum(dep_delay + arr_delay)) %>%
  mutate(season = c("summer")) %>%
  select(year,total_delay,season) %>%
  filter(! duplicated(total_delay))

print(summer_delay) # 2624301 1

## # A tibble: 1 × 3
##    year total_delay season
##   <int>       <dbl> <chr>
## 1  2013     2624301 summer

## autumn
autumn <- flights %>%
  mutate(bin_4 = cut(month, breaks = c(8,11))) %>%
  drop_na(bin_4)

autumn_delay <- autumn %>%
  select(year,month,dep_delay,arr_delay) %>%
  drop_na() %>%
  mutate(total_delay = sum(dep_delay + arr_delay)) %>%
  mutate(season = c("autumn")) %>%
  select(year,total_delay,season) %>%
  filter(! duplicated(total_delay))
```

```
print(autumn_delay$total_delay) # 402783 4

## [1] 402783

## joining Data
full_join_1 <- full_join(winter_delay,spring_delay)

## Joining, by = c("year", "total_delay", "season")

full_join_1 <- full_join_1[!duplicated(full_join_1),]
##
full_join_2 <- full_join(summer_delay,autumn_delay)

## Joining, by = c("year", "total_delay", "season")

full_join_2 <- full_join_2[!duplicated(full_join_2),]
##
full_join_3 <- full_join(full_join_1,full_join_2)

## Joining, by = c("year", "total_delay", "season")

full_join_3 <- full_join_3[!duplicated(full_join_3),]

#


#
as.integer(full_join_3$total_delay)

## [1] 1659150 2158845 2624301  402783

counts <- full_join_3$total_delay
# bar_plot
barplot(counts,beside = FALSE, main = "Seasonal Delay ",
        xlab = " Winter | Spring | Summer | Autumn")
```


Seasonal Delay
0
1000000
2000000
Winter | Spring | Summer | Autumn