# IMT 573: Problem Set 3

Working with Data: Part I

Jenny Skytta

Due: April 18, 2022

**Collaborators:** ***independent work***

**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)
library(nycflights13)
library(knitr) # this will keep code on the page!
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=FALSE)
tinytex::install_tinytex()
```

**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 <- tibble::as_tibble(flights)
# 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 2 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.

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

2. How many NYC airports are included in this data? Which airports are these?
3. Into how many airports did the airlines fly from NYC in 2013?
4. How many flights were there from NYC to Seattle (airport code `SEA`)?
5. Were the any flights from NYC to Spokane `(GAG)`?
6. What about missing destination codes? Are there any destinations that do not look like valid airport codes (i.e. three-letter-all-upper case)?

**(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?

```
# How many flights out of NYC are there in the data?
dep_NYC <- flights %>%
  group_by(origin) %>%
  filter(origin=="JFK" | origin=="LGA") %>%
  pull(origin)
# There are 215,941 flights out of NYC.

#How many NYC airports are included in this data?  Which airports are these?
unique_airports <- flights %>%
  group_by(origin) %>%
  summarise(unique(origin))
# There are 2 NYC airports included in this dataset.
#These airports are La Guardia and JFK International airports; both located in Queens.

#Into how many airports did the airlines fly from NYC in 2013?
num_dest_NYC <- flights %>%
  group_by(origin) %>%  # grouping by origin airline
  filter(origin=="JFK" | origin=="LGA") %>% #filtering for NYC airports
  summarise(total = unique(dest)) # summarizing by unique destination airports.
```

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

```
# The flights that departed NYC, flew into 138 different airports.

# How many flights were there from NYC to Seattle?
NYC_to_SEA <- flights %>%
  group_by(origin) %>% # grouping by origin airline
  filter(origin=="JFK" | origin=="LGA") %>% #filtering for NYC airports
  filter(dest=="SEA") #filtering for Seattle airport
# There were 2092 flights from NYC to Seattle in 2013.

# Were the any flights from NYC to Spokane
NYC_to_GAG <- flights %>%
  group_by(origin) %>%
  filter(origin=="JFK" | origin=="LGA") %>%
  filter(dest=="GAG") #filtering for Spokane airport
# There were zero flights from NYC to Spokane in 2013.

#What about missing destination codes?
```

```r
#Are there any destinations that do not look like valid airport codes
# (i.e. three-letter-all-upper case)?
dest_code_odd <- max(nchar(flights$dest))
#grabbing the highest character count from destination column
dest_code_small <-min(nchar(flights$dest))
#grabbing the lowest character count from destination column

dest_airports <- flights %>%
  group_by(dest) %>% #grouping by destination airports
  summarise(unique(dest)) # summarizing by unique destination airports
# The minimal and maximal character count is 3 which suggests the full column
#contains 3 character strings. Looking over the 105 unique airport codes,
#nothing appears to be an outlier.
```

While inspecting the dataset, I certainly feel that I was able to answer the specific questions. The questions were defined well enough to ascertain the answers but they did require additional querying beyond the dataset. For instance, I was not certain which airports were in NYC so I did need to look that up. For the question about Spokane, I deferred to the label of GAG to query whether there was a flight to that location. I did not inspect further to assess if there are additional airports in Spokane. I also did not look at any airport beyond SEA for Seattle.

**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?

```r
flight_delay <- flights %>%
  filter(dep_delay >= 1) %>% #filtering for departure delays over 0 minutes
  filter(arr_delay >= 1) %>% #filtering for arrival delays over 0 minutes
  summarise("total_delay" = mean(arr_delay + dep_delay))
# summarizing total delays by calculating the average of the total arrival and departure
# delays total.
```

The average delay time is 104 minutes or roughly 1 hour and 44 minutes.

**(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?

```r
flight_delay_med <- flights %>%
  filter(dep_delay >= 1) %>% #filtering for departure delays over 0 minutes
  filter(arr_delay >= 1) %>% #filtering for arrival delays over 0 minutes
  summarise("total_delay" = median(arr_delay + dep_delay))

delays_by_carrier <- flights %>%
  filter(dep_delay >= 1) %>%
  filter(arr_delay >= 1) %>%
  group_by(carrier) %>%
  summarise("total_delay" = mean(arr_delay + dep_delay)) %>%
  filter(total_delay == max(total_delay)) %>%
  pull(carrier)
```

I used an average of delay time in minutes in arrival and departure to answer the question of typical delay in the previous question but one could also assume median is akin to typical as a descriptor. I have recalculated using median. This provides a new answer of 63 minutes for typical delay. Typical could also be asking about airlines that have the most delays as in a typical profile for delays. The data exploration reveals that the HA airline was the airline with the most delays. One issue with delay is that delayed departure invariably leads to a delay arrival. Timezones will affect the overall hours so the safest method is to measure in minutes but the minutes are a number that is based on the change from the scheduled time.

**(c) 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.

```
delay_by_dest <- flights %>%
  filter(dep_delay >= 1) %>%
  filter(arr_delay >= 1) %>%
  filter(origin=="JFK" | origin=="LGA") %>%
  mutate(total_delay = (arr_delay + dep_delay)) %>%
  mutate(delayhrs = total_delay/60) %>%
  group_by(dest) %>%
  select(dest, total_delay, arr_delay, dep_delay, delayhrs) %>%
  arrange(desc(total_delay))

freq_delay_by_dest <- flights %>%
  filter(dep_delay >= 1) %>%
  filter(arr_delay >= 1) %>%
  filter(origin=="JFK" | origin=="LGA") %>%
  group_by(dest) %>%
  select(dest, arr_delay, dep_delay) %>%
  arrange(desc(dest)) %>%
  summarise(n = n())

kable(head(delay_by_dest))
```

| dest | total_delay | arr_delay | dep_delay | delayhrs |
|---|---|---|---|---|
| HNL | 2573 | 1272 | 1301 | 42.88333 |
| CMH | 2264 | 1127 | 1137 | 37.73333 |
| SFO | 2021 | 1007 | 1014 | 33.68333 |
| CVG | 1994 | 989 | 1005 | 33.23333 |
| TPA | 1891 | 931 | 960 | 31.51667 |
| MSP | 1826 | 915 | 911 | 30.43333 |

I created a table that captures all the destinations that encounter delays from NYC. The top 3 destinations that from NYC that encounter delays can be assessed by the longest delay or freqency of delays as the question itself is vague. The top 3 airports with the longest delays is calculated above by looking at the total delay time in minutes. These 3 cities are Honolulu, Columbus, and San Francisco.

```
kable(head(freq_delay_by_dest))
```

| dest | n |
|---|---|
| ABQ | 77 |

| dest | n |
|---|---|
| ACK | 71 |
| ATL | 3332 |
| AUS | 436 |
| AVL | 2 |
| BGR | 116 |

Returning to the question, "If you don't like delays" suggests frequency of delay rather than largest delay. I created a tibble that looks at frequency of delay. The data suggest that Chicago, Atlanta, and Fort Lauderdale have the most frequent flights with delays.

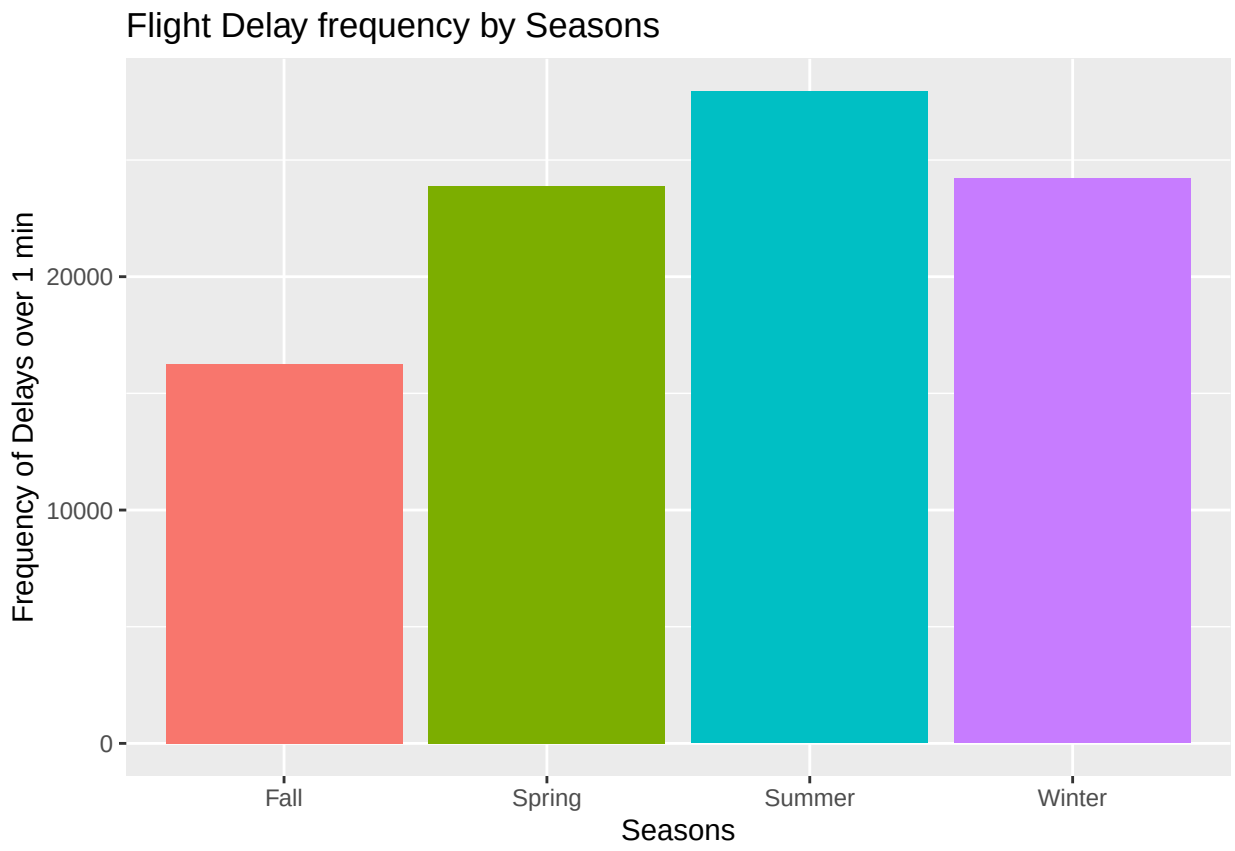
**(d) Seasonal Delays** Flight delays may be partly related to weather, as you might have experienced for yourself. We do not have weather information here but let's analyze how it is related to season. Which seasons have the worst flights delays? Why might this be the case? In your communication of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

```
seasons <- flights %>%
  filter(dep_delay >= 1) %>% #filtering for departure delays
  filter(arr_delay >= 1) %>%  #filtering for arrival delays
  select(arr_delay, dep_delay, month, origin, dest) # selecting columns

seasons$month[seasons$month == 1] <- "Winter"  #changing months to character names
seasons$month[seasons$month == 2] <- "Winter"
seasons$month[seasons$month == 3] <- "Spring"
seasons$month[seasons$month == 4] <- "Spring"
seasons$month[seasons$month == 5] <- "Spring"
seasons$month[seasons$month == 6] <- "Summer"
seasons$month[seasons$month == 7] <- "Summer"
seasons$month[seasons$month == 8] <- "Summer"
seasons$month[seasons$month == 9] <- "Fall"
seasons$month[seasons$month == 10] <- "Fall"
seasons$month[seasons$month == 11] <- "Fall"
seasons$month[seasons$month == 12] <- "Winter"

seasons2 <- seasons %>%
  group_by(month) %>%
  summarise(total_delays = n())

kable(seasons2)
```

| month | total_delays |
|---|---|
| Fall | 16272 |
| Spring | 23886 |
| Summer | 27939 |
| Winter | 24206 |

```
ggplot(data = seasons2) +
  geom_col(mapping = aes(x = month, y = total_delays, fill = month),show.legend = FALSE) +
  labs(
    title = "Flight Delay frequency by Seasons", # plot title
```

```
    x = "Seasons", # x-axis label
    y = "Frequency of Delays over 1 min", # y-axis label
  )
```




**(3) 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.

After really delving into this dataset, I do think my initial questions (below) were fairly vague, even after reflection. My thought process at the time about what is a delay wasn't as fully fleshed out as it may have needed to be for a comprehensive answer. My second question almost mirrors the question within this problem set regarding seasonality. In that breakdown, I looked at delays over the year as grouped by months. The months with the most delays were December and July which somewhat made sense. I don't know the underlying root of why summer is higher in delays. If we are exploring the question directly as a query of frequency as number of delays (arrival and departure) over time, this dataset appears to provide the answer that Summer has the most delays with Winter and Spring close behind.

## Problem 2 questions:

- *Does one carrier have more total delays?-*
- *Is there a time of year when there are more frequent delays?*

**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`).

1. How many flights were there from NYC airports to Portland in 2013?
2. How many airlines fly from NYC to Portland?
3. Which are these airlines (find the 2-letter abbreviations)? How many times did each of these go to Portland?
4. How many unique airplanes fly from NYC to PDX? Hint: airplane tail number is a unique identifier of an airplane.
5. How many different airplanes arrived from each of the three NYC airports to Portland?
6. What percentage of flights to Portland were delayed at departure by more than 15 minutes?
7. Is one of the New York airports noticeably worse in terms of departure delays for flights to Portland, OR than others?

**(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?

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

NYC_to_PDX <- flights %>%
  group_by(origin) %>% # grouping by origin airline
  filter(origin=="JFK" | origin=="LGA") %>% #filtering for NYC airports
  filter(dest=="PDX") #filtering for Portland airport
# There were 783 flights from NYC to Portland in 2013.

#How many airlines fly from NYC to Portland?
NYC_AL_PDX <- flights %>%
  group_by(origin) %>% # grouping by origin airline
  filter(origin=="JFK" | origin=="LGA") %>% #filtering for NYC airports
  filter(dest=="PDX") %>% #filtering for Portland airport
  summarise(airline = unique(carrier))
```

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

```
# Only two airlines in NYC has flights from NYC to Portland and both are out of JFK airport.

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

freq_NYC_AL_PDX <- flights %>%
  group_by(origin) %>% # grouping by origin airline
  filter(origin=="JFK" | origin=="LGA") %>% #filtering for NYC airports
  filter(dest=="PDX") %>% #filtering for Portland airport
  group_by(carrier) %>% # grouping by carrier (airline)
  summarise(carrier, total = n()) # totaling number of flights by carrier
```

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

```
  # There were 458 flights by DL and 325 flights by B6 from NYC to PDX in 2013.

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

plane_NYC_AL_PDX <- flights %>%
  group_by(origin) %>% # grouping by origin airline
  filter(origin=="JFK" | origin=="LGA") %>% #filtering for NYC airports
  filter(dest=="PDX") %>% #filtering for Portland airport
  group_by(tailnum) %>% # grouping by plane (unique tail numbers)
  summarise(total = n()) #total of unique plans

#There are 192 unique planes flying from NYC to PDX.

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

plane_arr_NYC_PDX <- flights %>%
  group_by(origin) %>% # grouping by origin airline
  filter(dest=="PDX") %>% #filtering for Portland airport
  group_by(tailnum) %>% # grouping by plane (tail numbers)
  filter(!is.na(tailnum)) %>%
  summarise(total = n()) #total of different planes

#There were 491 different airplanes that arrived from each of the
#three NYC airports (including Newark NJ) to Portland.

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

delay_15min_perct <- flights %>%
  filter(dest=="PDX") %>% #filtering flights to Portland
  mutate(pdxflightsall = n()) %>% #create total of PDX flights
   filter(dep_delay>= 15) %>% #filtering for delayed departure 15 mins
    mutate(delay15flt = n()) %>%   #create total for delayed PDX flights
    summarise(proportion = round((delay15flt/pdxflightsall)*100)) #calculate proportion

#It looks like 27% of flights to Portland had at least a 15 minute departure delay.

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

delay_NYC_15 <- flights %>%
  filter(dest=="PDX") %>% #filtering flights to Portland
  group_by(origin) %>% #group by NYC airports including Newark
  mutate(pdxflightsall = n()) %>%  #create total of PDX flights
   filter(dep_delay>= 15) %>% #filtering for delayed departure 15 mins
    mutate(delay15flt = n()) %>% #create total for delayed PDX flights
    summarise(proportion = round((delay15flt/pdxflightsall)*100))
```

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

```
# Newark accounts for 30% of the delayed flights while JFK accounts for 25%.
#This doesn't seem to be a noticeable distinction.
```

**Were you able to answer all of these questions? Are all questions well defined? Is the data good enough to answer all these?** All of these questions were answerable but terms like "typical" and "noticeably worse" aren't well defined. The dataset does seem to be decent enough to answer the questions within the paremeters of 2013 flights from New York. There were items that required additional searches such as airport names from the codes, as well as airline names. I wouldn't have intuitively known that tailnumbers were a unique plane identifier either.

**Citations:**

RITA, Bureau of transportation statistics https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236

Code written above is from the previous course IMT 511 which used the below text to support class scripts. https://www.google.com/books/edition/Programming_Skills_for_Data_Science/BnB6DwAAQBAJ?hl=en&gbpv=1&printsec=frontcover

NEWS: Visualizing the U.S. Airports with the Worst Flight Delays https://www.visualcapitalist.com/visualizing-the-u-s-airports-with-the-worst-flight-delays/