IMT 573: Problem Set 2 - Working with Data

Alexander Davis

Due: Tuesday, October 20, 2020 by 9am PT

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 object 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.

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

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, 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.

\begin{enumerate}

```r
data(flights)
flights <- tibble(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>
```

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

```
cat("There are", nrow(flights), "flights out of NYC in the dataset.")
```

```
## There are 336776 flights out of NYC in the dataset.
```

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

```
cat("There are", length(unique(flights$origin)), "airports and they are",
    unique(flights$origin))
```

```
## There are 3 airports and they are EWR LGA JFK
```

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

```
cat("There were", length(unique(flights$dest)), "unique airports that airlines
    flew into from NYC in 2013.")
```

```
## There were 105 unique airports that airlines
##     flew into from NYC in 2013.
```

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

```
to_seattle <- flights %>%
  filter(dest == "SEA") %>%
  nrow()

cat("There were", to_seattle, "flights were there from NYC to Seattle.")
```

```
## There were 3923 flights were there from NYC to Seattle.
```

Were the any flights from NYC to Spokane ?

```
to_spokane <- flights %>%
  filter(dest == "GAG")
to_spokane
```

```
## # A tibble: 0 x 19
## # … with 19 variables: year <int>, month <int>, day <int>, dep_time <int>,
## #   sched_dep_time <int>, dep_delay <dbl>, arr_time <int>,
## #   sched_arr_time <int>, 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>
```

```
cat("No flights from NYC to Spokane")
```

```
## No flights from NYC to Spokane
```

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

```
airports <- unique(flights$dest)
nchar(airports)
```

```
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

```
grepl("^[[:upper:]]{3}$",airports)
```

```
##   [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [61] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [76] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [91] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
## Pulling all unique values from destination airport codes looks like they are
## all valid airports without further verification i.e. I would need a valid
## list of airport codes to compare the ones in this file. Otherwise all
## airports appear to have 3 letters and are uppercase.
```

\end{enumerate}

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
```

(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?

I was able to answer all of the questions with either data or lack there of as it wasn't presenet in the dataset. For example, the airport Spokane was not in the dataset likley as it is a regional airport and not many people travel to this city from NYC. Questions seemed to be defined well enough albeit were simple and easy to understand. The data is in good enough quality to answer these questions.

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.

(1) Typical Delays

What is the typical delay of flights in this data?

```
typical_delay <- flights %>%
  select(dep_delay) %>%
  na.omit()

mean(typical_delay$dep_delay)
```

```
## [1] 12.63907
```

(2) 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 average after removing NA values. I did some exploration to remove the NA values. It would appear that some of the min/max values are probably implausible. Having a delay of 1301 min or 21 hours seems very excessive. Leaving 43 min early seems somewhat possible if the flight is nearly empty.

```
min(typical_delay)
```

```
## [1] -43
```

```
max(typical_delay)
```

```
## [1] 1301
```

(3) 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_dest <- flights %>%
  select(dest, dep_delay) %>%
  na.omit() %>%
  group_by(dest) %>%
  summarise(mean = mean(dep_delay), n = n()) %>%
  arrange(-mean) %>%
  slice(1:3)

delay_dest
```

```
## # A tibble: 3 x 3
##   dest   mean     n
##   <chr> <dbl> <int>
## 1 CAE    35.6   107
## 2 TUL    34.9   299
## 3 OKC    30.6   327
```

I chose to group_by and find the average delay for each destination from NYC along with the count of flights for each destination. This way I can see if there is also a significant difference between the number of flights and what the average delay is. It seems like (from the top three) that number of flights are not a factor in delays and it might even have an inverse effect.

(4) Delays by time of day

We'd like to know how much do 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.

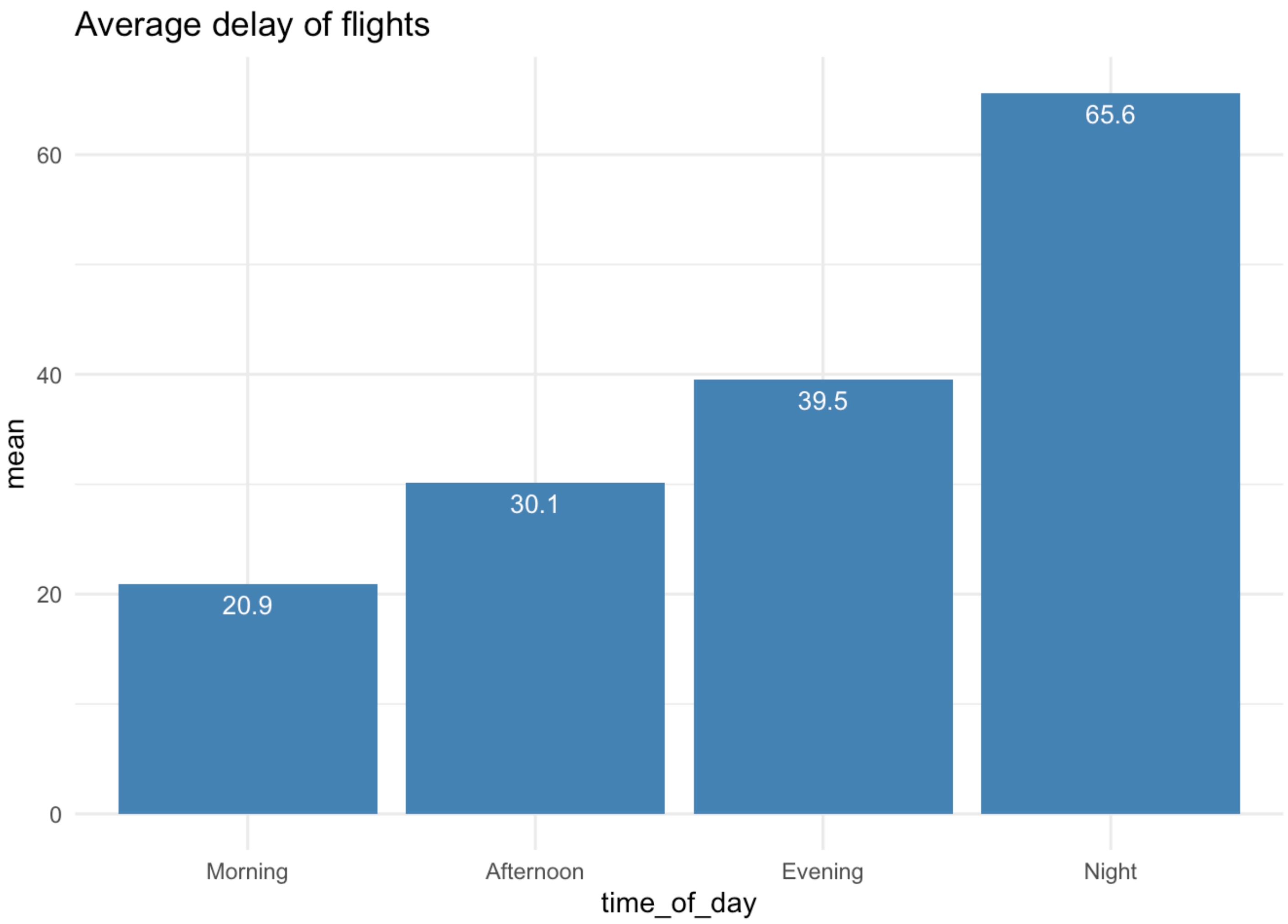
```
temp_df <- flights %>%
  select(year, month, day, dep_time, dep_delay, carrier) %>%
    na.omit() %>%
  filter(dep_delay > 0) # Omit any delays that are below 0 min. These flights would b
e considered leaving early.
temp_df
```

```
## # A tibble: 128,432 x 6
##     year month   day dep_time dep_delay carrier
##    <int> <int> <int>    <int>     <dbl> <chr>
##  1  2013     1     1      517         2 UA
##  2  2013     1     1      533         4 UA
##  3  2013     1     1      542         2 AA
##  4  2013     1     1      601         1 B6
##  5  2013     1     1      608         8 MQ
##  6  2013     1     1      611        11 UA
##  7  2013     1     1      613         3 B6
##  8  2013     1     1      623        13 AA
##  9  2013     1     1      632        24 EV
## 10  2013     1     1      644         8 UA
## # … with 128,422 more rows
```

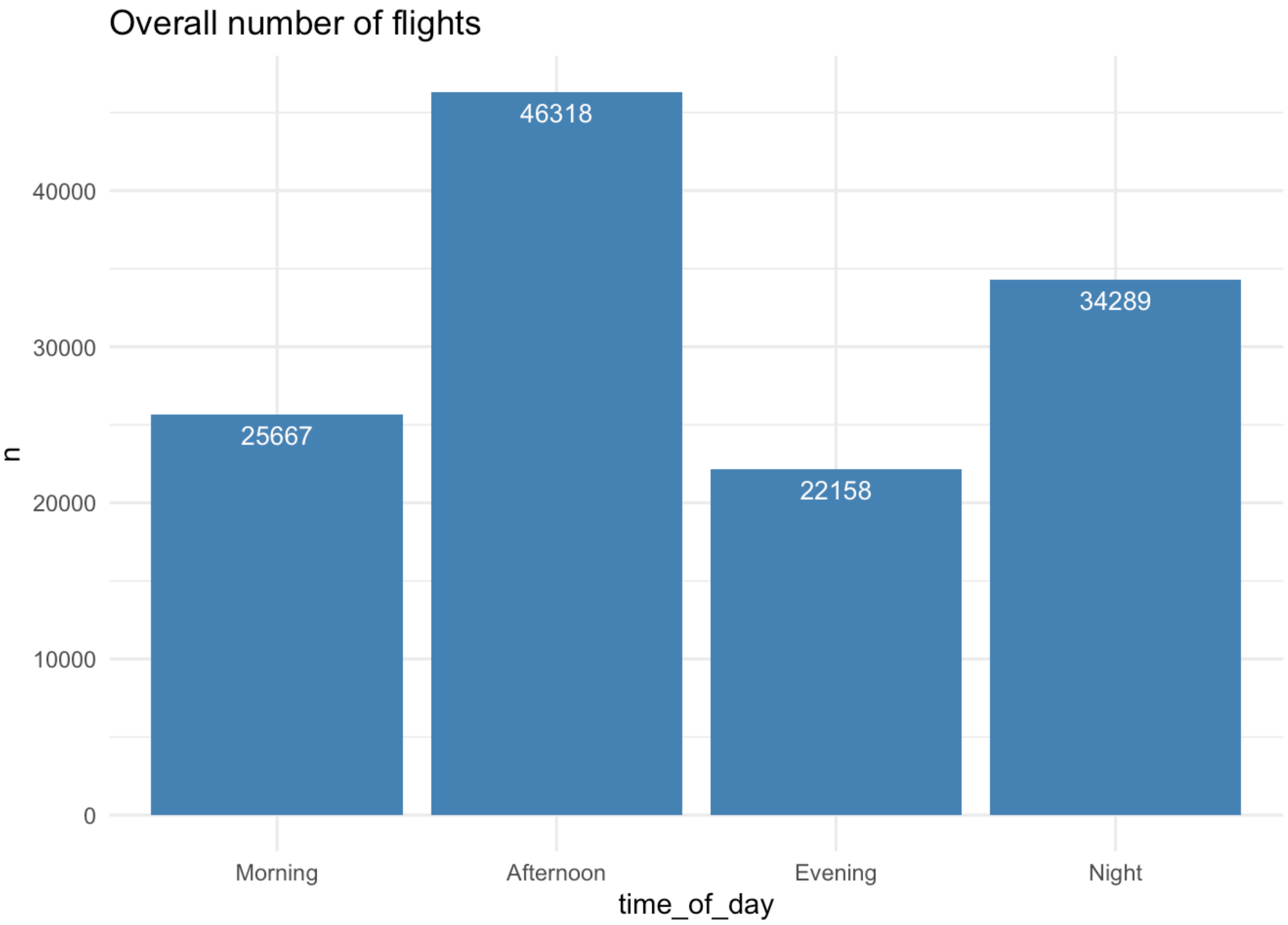
```
# Mutate and assign the times to a specific category
delay_df <- mutate(temp_df, time_of_day =
         ifelse(dep_time %in% 00300:1059, "Morning",
         ifelse(dep_time %in% 1100:1659, "Afternoon",
         ifelse(dep_time %in% 1700:1900, "Evening", "Night")))) %>%
  group_by(time_of_day) %>%
  summarise(mean = round(mean(dep_delay), 1), n = n())
delay_df
```

```
## # A tibble: 4 x 3
##   time_of_day  mean     n
##   <chr>       <dbl> <int>
## 1 Afternoon    30.1 46318
## 2 Evening      39.5 22158
## 3 Morning      20.9 25667
## 4 Night        65.6 34289
```

```
# Looking at the average delay per flight (removed early flights and NA values)
ggplot(data = delay_df,aes(x=time_of_day, y=mean)) +
  geom_bar(stat="identity", fill="steelblue") +
  geom_text(aes(label=mean), vjust=1.6, color="white", size=3.5) +
  scale_x_discrete(limits=c("Morning", "Afternoon", "Evening", "Night")) +
  theme_minimal() +
  ggtitle("Average delay of flights")
```



```
# Looking at the overall number of flights
ggplot(data = delay_df,aes(x=time_of_day, y=n)) +
  geom_bar(stat="identity", fill="steelblue") +
  geom_text(aes(label=n), vjust=1.6, color="white", size=3.5) +
  scale_x_discrete(limits=c("Morning", "Afternoon", "Evening", "Night")) +
  theme_minimal() +
  ggtitle("Overall number of flights")
```



(5) 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?

I think the questions asked are fine. The problem with looking at time of day is that it can be over-simplified. There can be a lot of variability among airlines and airports so this wouldn't help a whole lot except for figuring out when the best time of the day it is to fly. I would specify in this question if we need to see when is the *longest delay or how many (n) delays are there in each part of the day. It would also be nice to know if there is a convention when it comes to grouping the times. I would want to break down this exploration into the three different kind of airports in NYC and also take into consideration the airlines that operate here.*

Problem 3: Let's Fly 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`).

\begin{enumerate}

```
pdx_flights <- flights %>%
  filter(dest == "PDX")
```

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

```
cat("There were", nrow(pdx_flights),"flights to Portland in 2013.")
```

```
## There were 1354 flights to Portland in 2013.
```

How many airlines fly from NYC to Portland?

```
cat(length(unique(pdx_flights$carrier)), "airlines flew from NYC to Portland.")
```

```
## 3 airlines flew from NYC to Portland.
```

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

```
pdx_airlines <- pdx_flights %>%
  select(carrier, dest) %>%
  group_by(carrier) %>%
  summarize(count = n())

pdx_airlines
```

```
## # A tibble: 3 x 2
##   carrier count
##   <chr>   <int>
## 1 B6        325
## 2 DL        458
## 3 UA        571
```

How many unique airplanes fly from NYC to PDX? \

```
cat(length(unique(pdx_flights$tailnum)),
    "unique airplines flew from NYC to PDX.")
```

```
## 492 unique airplines flew from NYC to PDX.
```

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

```
three_nyc_airports <- pdx_flights %>%
  select(origin, tailnum) %>%
  group_by(origin) %>%
  summarize(n = n())

three_nyc_airports
```

```
## # A tibble: 2 x 2
##   origin     n
##   <chr>  <int>
## 1 EWR      571
## 2 JFK      783
```

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

```
more_than <- pdx_flights %>%
  filter(dep_delay > 15) %>%
  nrow()

perc_flights <- more_than / nrow(pdx_flights)

cat(perc_flights * 100,
    "percentage of flights were delayed by more than 15 min.")
```

```
## 26.66174 percentage of flights were delayed by more than 15 min.
```

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

```
worse_airport <- pdx_flights %>%
  select(origin, dep_delay) %>%
  group_by(origin) %>%
  na.omit() %>%
  summarize(mean = mean(dep_delay)) %>%
  arrange(-mean)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
worse_airport[1,]
```

```
## # A tibble: 1 x 2
##   origin  mean
##   <chr>  <dbl>
## 1 EWR     16.6
```

\end{enumerate}

(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 questions. I think the continuous issue is not being specific enough when it comes to flying from NYC to PDX. Unfortunately there are three airports for NYC so this wouldn't be useful information to any of the specific airlines or airports until we were able to break it out by airport. The questions could be refined a bit more in this respect. The data is good enough though to answer all of these questions.

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.

```
#install.packages("flextable")
library(flextable)
```

In my analysis of the average flight delays broken down by season, summer appears to be the season with the biggest average delay. It also has by far the most amount of flights and this is probably due to tourist season during this period of months; July - September. This is often when schools are on summer break and families are traveling. This despite NYC typically have weather during the winter season that might greatly effect flight travel.

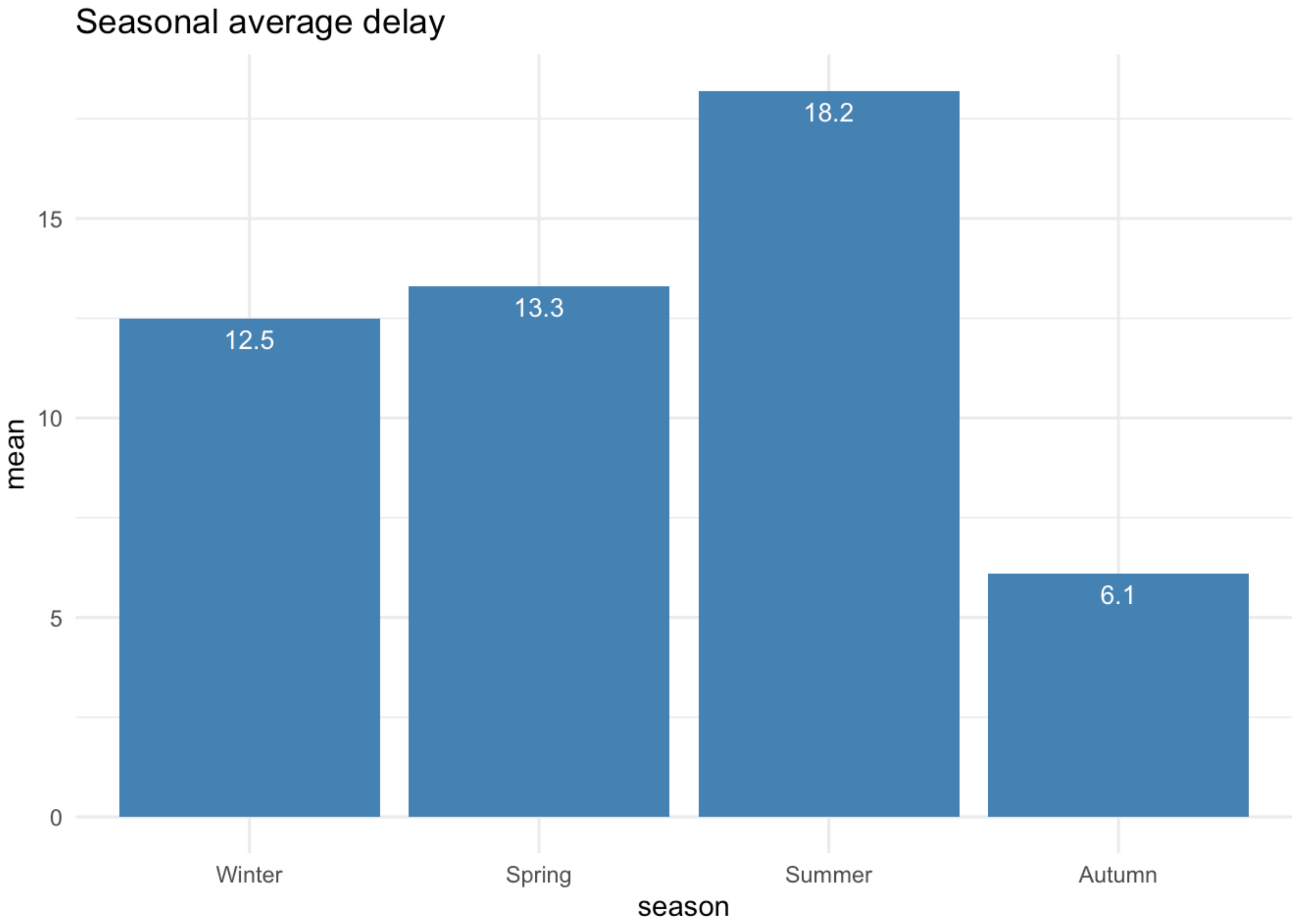
```
# For reference:
# Winter — December, January and February.
# Spring — March, April and May.
# Summer — June, July and August.
# Autumn — September, October and November.

seasonal_df <- mutate(flights, season =
         ifelse(month %in% 3:5, "Spring",
         ifelse(month %in% 6:8, "Summer",
         ifelse(month %in% 9:11, "Autumn", "Winter")))) %>%
  na.omit %>%
  group_by(season) %>%
  summarise(mean = round(mean(dep_delay), 1), n = n())

flextable(
  seasonal_df
          )
```

season	mean	n
Autumn	6.1	82599
Spring	13.3	83594
Summer	18.2	84124
Winter	12.5	77029

```
ggplot(data = seasonal_df,aes(x=season, y=mean)) +
  geom_bar(stat="identity", fill="steelblue") +
  geom_text(aes(label=mean), vjust=1.6, color="white", size=3.5) +
  scale_x_discrete(limits=c("Winter", "Spring", "Summer", "Autumn")) +
  theme_minimal() +
  ggtitle("Seasonal average delay")
```


Seasonal average delay
mean
season
0
5
10
15
Winter
Spring
Summer
Autumn
12.5
13.3
18.2
6.1