applied_ps1_new

Rabail Sofi

2023-04-05

```
knitr::opts_chunk$set(echo = TRUE)

rm(list=ls())
library(tinytex)
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.1.9000     ✓ readr     2.1.4
## ✓ forcats   1.0.0          ✓ stringr   1.5.0
## ✓ ggplot2   3.4.1          ✓ tibble    3.2.1
## ✓ lubridate 1.9.2          ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(devtools)

## Loading required package: usethis

library(palmerpenguins)
library(ggthemes)
library(rmarkdown)
library(knitr)
library(dplyr)
knitr::opts_chunk$set(error = TRUE)
knitr::opts_chunk$set(echo = TRUE)
```

Front matter

This submission is my work alone and complies with the 30535 integrity policy.

Add your initials to indicate your agreement: << RS >>

Add your collaborators: <<__>>

Late coins used this pset: 1. Late coins left: 3.

R for Data Science Exercises

First Steps

Flight Data: Part 1

Download BTS Data

1.1.1

```
setwd("/Users/rabailsofi/Desktop")
```

1.1.2

```
# repo called "data" saved.
```

1.1.3

```
download.file("https://github.com/datasci-harris/Applied-PS1-Data-
Files/blob/main/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_20
16_1.zip", destfile =
"rabailsofi/Desktop/On_Time_Reporting_Carrier_On_Time_Performance_1987_presen
t_2016_1.zip")
```

```
## Warning in
## download.file("https://github.com/datasci-harris/Applied-PS1-Data-
Files/blob/main/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_20
16_1.zip",
## : URL
## https://github.com/datasci-harris/Applied-PS1-Data-
Files/blob/main/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_20
16_1.zip:
## cannot open destfile
##
'rabailsofi/Desktop/On_Time_Reporting_Carrier_On_Time_Performance_1987_presen
t_2016_1.zip',
## reason 'No such file or directory'

## Warning in
## download.file("https://github.com/datasci-harris/Applied-PS1-Data-
Files/blob/main/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_20
16_1.zip",
## : download had nonzero exit status
```

1.1.4

```
setwd("/Users/rabailsofi/Desktop")


download.file("https://github.com/datasci-harris/Applied-PS1-Data-
Files/blob/main/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_20
```

```
16_1.zip", destfile =
"/Users/rabailsofi/Desktop/data/On_Time_Reporting_Carrier_On_Time_Performance
_1987_present_2016_1 (4).zip")
```

1.1.5

```
website <- "https://github.com/datasci-harris/Applied-PS1-Data-
Files/blob/main/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_20
16_"

path <- str_c("/Users/rabailsofi/Desktop/data",".zip")

files <- str_c(1:12, ".zip")

download.file(url = str_c(website, files), destfile = str_c(path, files))

## Error in download.file(url = str_c(website, files), destfile = str_c(path,
: 'url' must be a length-one character vector
```

1.1.6

```
setwd("/Users/rabailsofi/Desktop")

flights1 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_1.c
sv")
flights2 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_2.c
sv")
flights3 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_3.c
sv")
flights4 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_4.c
sv")
flights5 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_5.c
sv")
flights6 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_6.c
sv")
flights7 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_7.c
sv")
flights8 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_8.c
sv")
flights9 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_9.c
sv")
flights10 <-
```

```
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_10.
csv")
flights11 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_11.
csv")
flights12 <-
read.csv("On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_12.
csv")

flights1 <- flights1 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights2 <- flights2 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights3 <- flights3 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights4 <- flights4 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights5 <- flights5 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights6 <- flights6 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights7 <- flights7 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights8 <- flights8 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights9 <- flights9 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights10 <- flights10 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights11 <- flights11 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
flights12 <- flights12 |>
  filter(OriginState == "IL" | DestStateName == "Illinois")
```

1.1.7

```
allflightsdata <- bind_rows(flights1, flights2, flights3, flights4, flights5,
flights6, flights7, flights8, flights9, flights10, flights11, flights12)

## Error in list2(...): object 'flights1' not found
```

1.1.8

```
setwd("/Users/rabailsofi/Desktop")
read_csv("allflightsdata.csv")
```

1.1.9

```
# Manually knitted PDF, timed up to 2 minutes and 10 seconds.
```

1.1.10

```
view("allflightsdata.csv")
```

1.1.11

```
# *.csv
```

Data Description

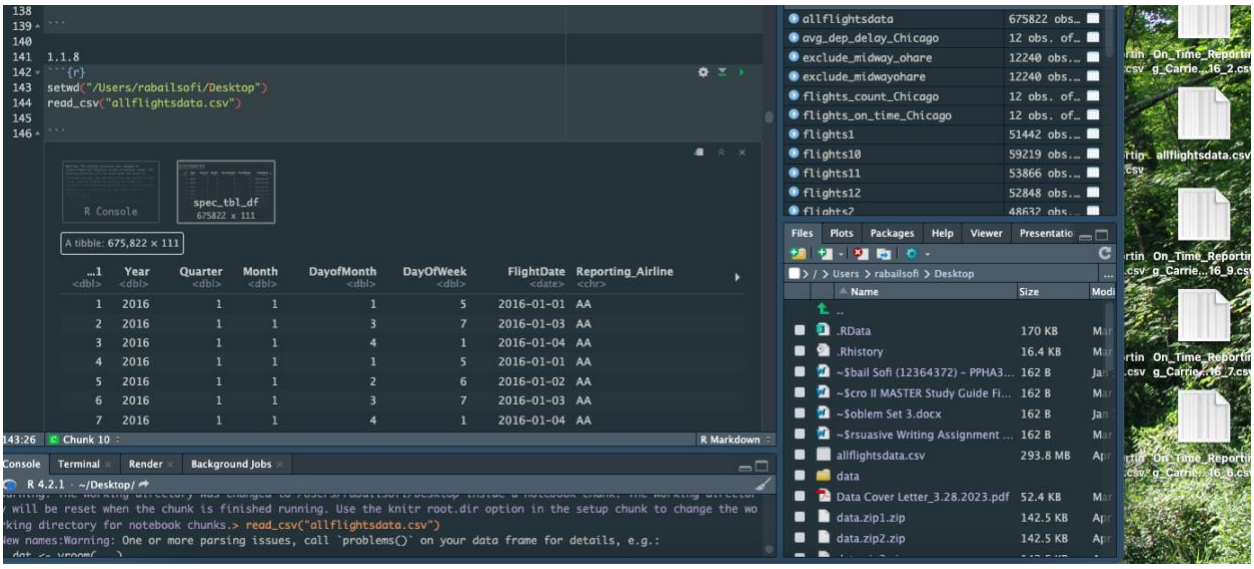
1.2.1 The tail numbers and dates.

1.2.2

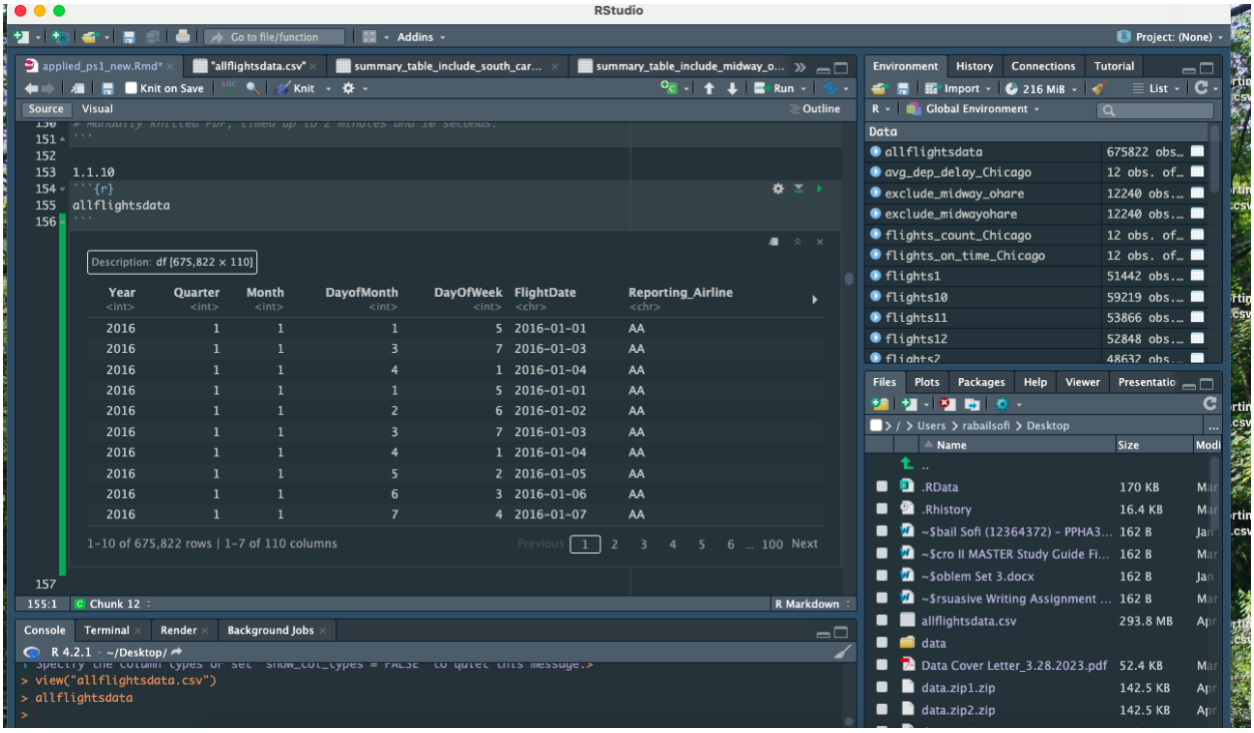
1.) The print() function and head() function are almost identical. However, print() gives a better output because you can still scroll through all the columns and rows while previewing the data, while head() doesn't allow us to scroll and view more data.

2.) The summary() function gives us a better output because its able to give us the averages, minimums, and maximums of each variable in the data set. The glimse() function also gives a good output because it shows us a quick preview of the data and categories in each variable.

```
print(allflightsdata)

## Error in print(allflightsdata): object 'allflightsdata' not found

head(allflightsdata)

## Error in head(allflightsdata): object 'allflightsdata' not found

glimpse(allflightsdata)

## Error in glimpse(allflightsdata): object 'allflightsdata' not found

str(allflightsdata)

## Error in str(allflightsdata): object 'allflightsdata' not found

View(allflightsdata)

## Error in as.data.frame(x): object 'allflightsdata' not found

summary(allflightsdata)

## Error in summary(allflightsdata): object 'allflightsdata' not found
```



*** as you can see in the screenshot, the file "allfightsdata.csv" is right on my desktop, has been loaded correctly in R, and has the correct setwd.



Data Cleaning

1.3.1 110 variables.

1.3.2

```
nycflights <- nycflights13::flights
view(nycflights)
?flights
```

1.3.3 Variables from nycflights such as air time and time hour were missing in the newly downloaded data set.

Data Validation

1.4.1

The test_that function does not work on my end, but my newly compiled data set, allflightsdata, does have a total of 675822 rows.

```
library(testthat)

##
## Attaching package: 'testthat'

## The following object is masked from 'package:devtools':
##
##     test_file

## The following object is masked from 'package:dplyr':
##
##     matches

## The following object is masked from 'package:purrr':
##
##     is_null

## The following objects are masked from 'package:readr':
##
##     edition_get, local_edition

## The following object is masked from 'package:tidyr':
##
##     matches

test_that(
  "we have the right number of rows",
  expect_equal(nrow(allflightsdata), 675822))

## ── Error (???): we have the right number of rows
──────────────────────────────
## Error in `nrow(allflightsdata)`: object 'allflightsdata' not found
## Backtrace:
##  1. testthat::expect_equal(nrow(allflightsdata), 675822)
##  4. base::nrow(allflightsdata)

## Error in `reporter$stop_if_needed()`:
## ! Test failed
```

1.4.2

```
test_that(
  "we have the right number of rows",
  expect_equal(nrow(allflightsdata$OriginState=="IL"), 340625))
```

```
## ── Error (???): we have the right number of rows
──────────────────────────────
## Error in `nrow(allflightsdata$OriginState == "IL")`: object
'allflightsdata' not found
## Backtrace:
##  1. testthat::expect_equal(...)
##  4. base::nrow(allflightsdata$OriginState == "IL")

## Error in `reporter$stop_if_needed()`:
## ! Test failed
```

```
allflightsdata |>
  filter(OriginState=="IL")
```

```
## Error in filter(allflightsdata, OriginState == "IL"): object
'allflightsdata' not found
```

1.4.3

After removing flights to and from ORD and MDW, there are only 12,240 flights left in the dataframe.

Citation: https://stackoverflow.com/questions/6650510/remove-rows-from-data-frame-where-a-row-matches-a-string

```
exclude_midway_ohare <-allflightsdata[(!(allflightsdata$Origin=="ORD") &
!(allflightsdata$Origin=="MDW") & !(allflightsdata$Dest=='ORD') &
!(allflightsdata$Dest=="MDW")),]
```

```
## Error in eval(expr, envir, enclos): object 'allflightsdata' not found
```

1.4.4

````
1.4.4
```{r}

top_5_origins <- exclude_midway_ohare |>
  count(Origin) |>
  slice_max(n, n=5) |>
  print(top_5_origins)

top_5_dests <- exclude_midway_ohare |>
  count(Dest) |>
  slice_max(n, n=5) |>
  print(top_5_dests)

```
````

data.frame 5 x 2 | data.frame 5 x 2

Description: df [5 × 2]

Origin <chr>	n <int>
ATL	2966
PIA	2001
MLI	1984
BMI	1517
DTW	1233


````
1.4.4
```{r}

top_5_origins <- exclude_midway_ohare |>
  count(Origin) |>
  slice_max(n, n=5) |>
  print(top_5_origins)

top_5_dests <- exclude_midway_ohare |>
  count(Dest) |>
  slice_max(n, n=5) |>
  print(top_5_dests)

```
````


data.frame 5 x 2 | data.frame 5 x 2

Description: df [5 × 2]

Dest <chr>	n <int>
ATL	2968
PIA	1999
MLI	1926
BMI	1517
DTW	1228

```
top_5_origins <- exclude_midway_ohare |>
  count(Origin) |>
```

```
  slice_max(n, n=5) |>
  print(top_5_origins)

## Error in count(exclude_midway_ohare, Origin): object
'exclude_midway_ohare' not found

top_5_dests <- exclude_midway_ohare |>
  count(Dest) |>
  slice_max(n, n=5) |>
  print(top_5_dests)

## Error in count(exclude_midway_ohare, Dest): object 'exclude_midway_ohare'
not found
```

1.4.5

1.) When we limit the flights only to and from ORD and MDW, the number of flights should be 663582 (total number of flights, 675822 - flights without ORD and MDW, 12240.

```
include_midway_ohare <- allflightsdata |>
  filter(Origin=="ORD" | Origin=="MDW" | Dest=="ORD" | Dest=="MDW")

## Error in filter(allflightsdata, Origin == "ORD" | Origin == "MDW" | Dest
== : object 'allflightsdata' not found
```

1.4.6 According to an article by Upgraded Points, Ohare and Midway are "a major airport for connections all over the U.S., operating as the main hub for United Airlines and the third-largest hub for American Airlines". Luftansa and Air Canada are next largest airlines that enter and exit Ohare and Midway. While real time data of how many flights enter and exit those two airports isn't avaible, sources say that north of 900 flights enter & exit the city every single day. Knowing that Chicgao is one of the most visited cities in the United States, these numbers and the BTS data seem to match and this dataset can be assessed as believable.

Citation: https://upgradedpoints.com/travel/airports/chicago-ohare-ord-airport/ & https://simpleflying.com/chicago-o-hare-world-top-regional-jet-airport/#:~:text=Up%20To%20914%20Flights%20A,World's%20Top%20Regional%20Jet%20Airport

Flight Data: Part II

2.1.1

November has the lowest average arrival delays. At least 80% of the flights are on time in October as per the table 'flights_on_time_Chicago'. In the ggplot below, we can see that October and November have the lowest average arrival delays while also most of the flights during these months are on time. Some other months that have flights on time are January and February. In terms of average arrival delays, the summer months have the most delays and the colder months have fewer delays.

Citation: https://www.statology.org/ggplot-multiple-data-frames/ & https://r-graph-gallery.com/line-chart-dual-Y-axis-ggplot2.html

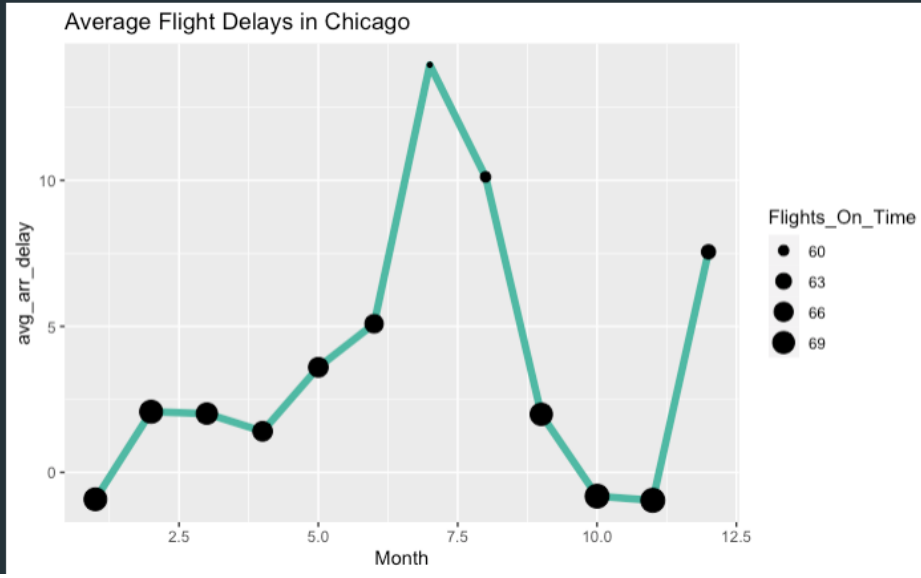
2.1.1

November has the lowest average arrival delays. At least 80% of the flights are on time in October as per the table 'flights_on_time_Chicago'. In the ggplot below, we can see that October and November have the lowest average arrival delays while also most of the flights during these months are on time. Some other months that have flights on time are January and February. In terms of average arrival delays, the summer months have the most delays and the colder months have fewer delays.

Citation: https://www.statology.org/ggplot-multiple-data-frames/ &
https://r-graph-gallery.com/line-chart-dual-Y-axis-ggplot2.html

```{r}
summary_table_Chicago <- allflightsdata |>
  filter(DestCityName=="Chicago, IL") |>
  group_by(Month) |>
  summarize(avg_arr_delay = mean(ArrDelay, na.rm = TRUE),
            Flights_On_Time = mean(ArrDelay <=0, na.rm= TRUE) * 100,
            n=n())

include_midway_ohare_1<- include_midway_ohare
include_midway_ohare_1$ArrDelayMinutes <- ifelse(include_midway_ohare$ArrDelay > 0, '1', '0')

flights_on_time_Chicago<- include_midway_ohare_1 |>
  group_by(Month) |>
  summarize(ave_arr_delay = mean(ArrDelay, na.rm = TRUE))
view(flights_on_time_Chicago)

ggplot(summary_table_Chicago, aes(x=Month, y=avg_arr_delay, size = Flights_On_Time)) +
   geom_line(color="#69b3a2", size=2) +
   ggtitle("Average Flight Delays in Chicago") +
   geom_point()
```

```
284    geom_line(color="#69b3a2", size=2) +
285    ggtitle("Average Flight Delays in Chicago") +
286    geom_point()
287
288  ```
```




```
summary_table_Chicago <- allflightsdata |>
  filter(DestCityName=="Chicago, IL") |>
  group_by(Month) |>
```

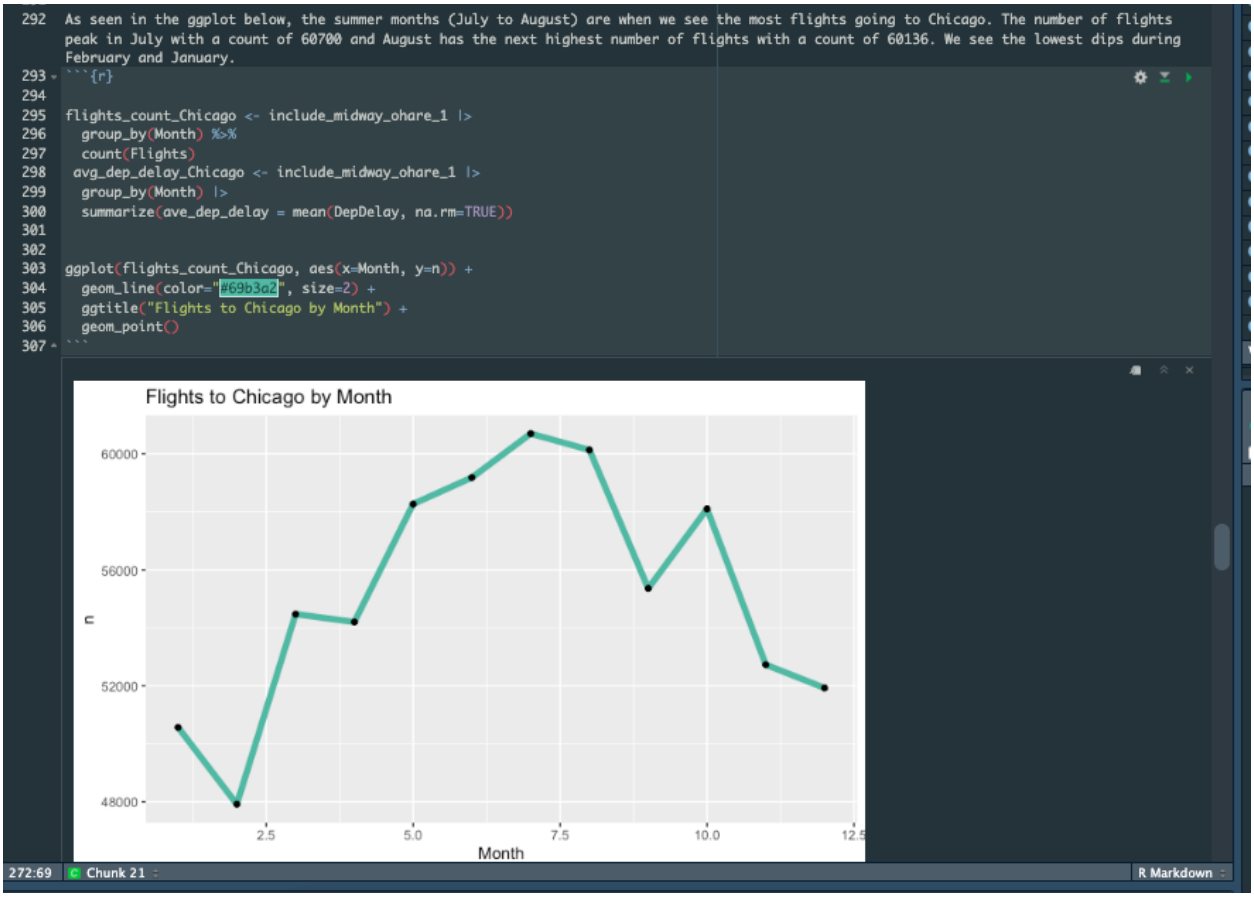
```
  summarize(avg_arr_delay = mean(ArrDelay, na.rm = TRUE),
            Flights_On_Time = mean(ArrDelay <=0, na.rm= TRUE) * 100,
            n=n())

## Error in filter(allflightsdata, DestCityName == "Chicago, IL"): object
'allflightsdata' not found

include_midway_ohare_1<- include_midway_ohare

## Error in eval(expr, envir, enclos): object 'include_midway_ohare' not
found

include_midway_ohare_1$ArrDelayMinutes <-
ifelse(include_midway_ohare$ArrDelay > 0, '1', '0')

## Error in ifelse(include_midway_ohare$ArrDelay > 0, "1", "0"): object
'include_midway_ohare' not found

flights_on_time_Chicago<- include_midway_ohare_1 |>
  group_by(Month) |>
  summarize(ave_arr_delay = mean(ArrDelay, na.rm = TRUE))

## Error in group_by(include_midway_ohare_1, Month): object
'include_midway_ohare_1' not found

view(flights_on_time_Chicago)

## Error in view(flights_on_time_Chicago): object 'flights_on_time_Chicago'
not found

ggplot(summary_table_Chicago, aes(x=Month, y=avg_arr_delay, size =
Flights_On_Time)) +
   geom_line(color="#69b3a2", size=2) +
   ggtitle("Average Flight Delays in Chicago") +
   geom_point()

## Error in ggplot(summary_table_Chicago, aes(x = Month, y = avg_arr_delay, :
object 'summary_table_Chicago' not found
```

2.1.2

As seen in the ggplot below, the summer months (July to August) are when we see the most flights going to Chicago. The number of flights peak in July with a count of 60700 and August has the next highest number of flights with a count of 60136. We see the lowest dips during February and January.

```
flights_count_Chicago <- include_midway_ohare_1 |>
  group_by(Month) %>%
  count(Flights)
```

```
## Error in group_by(include_midway_ohare_1, Month): object
'include_midway_ohare_1' not found

 avg_dep_delay_Chicago <- include_midway_ohare_1 |>
  group_by(Month) |>
  summarize(ave_dep_delay = mean(DepDelay, na.rm=TRUE))

## Error in group_by(include_midway_ohare_1, Month): object
'include_midway_ohare_1' not found

ggplot(flights_count_Chicago, aes(x=Month, y=n)) +
  geom_line(color="#69b3a2", size=2) +
  ggtitle("Flights to Chicago by Month") +
  geom_point()

## Error in ggplot(flights_count_Chicago, aes(x = Month, y = n)): object
'flights_count_Chicago' not found
```

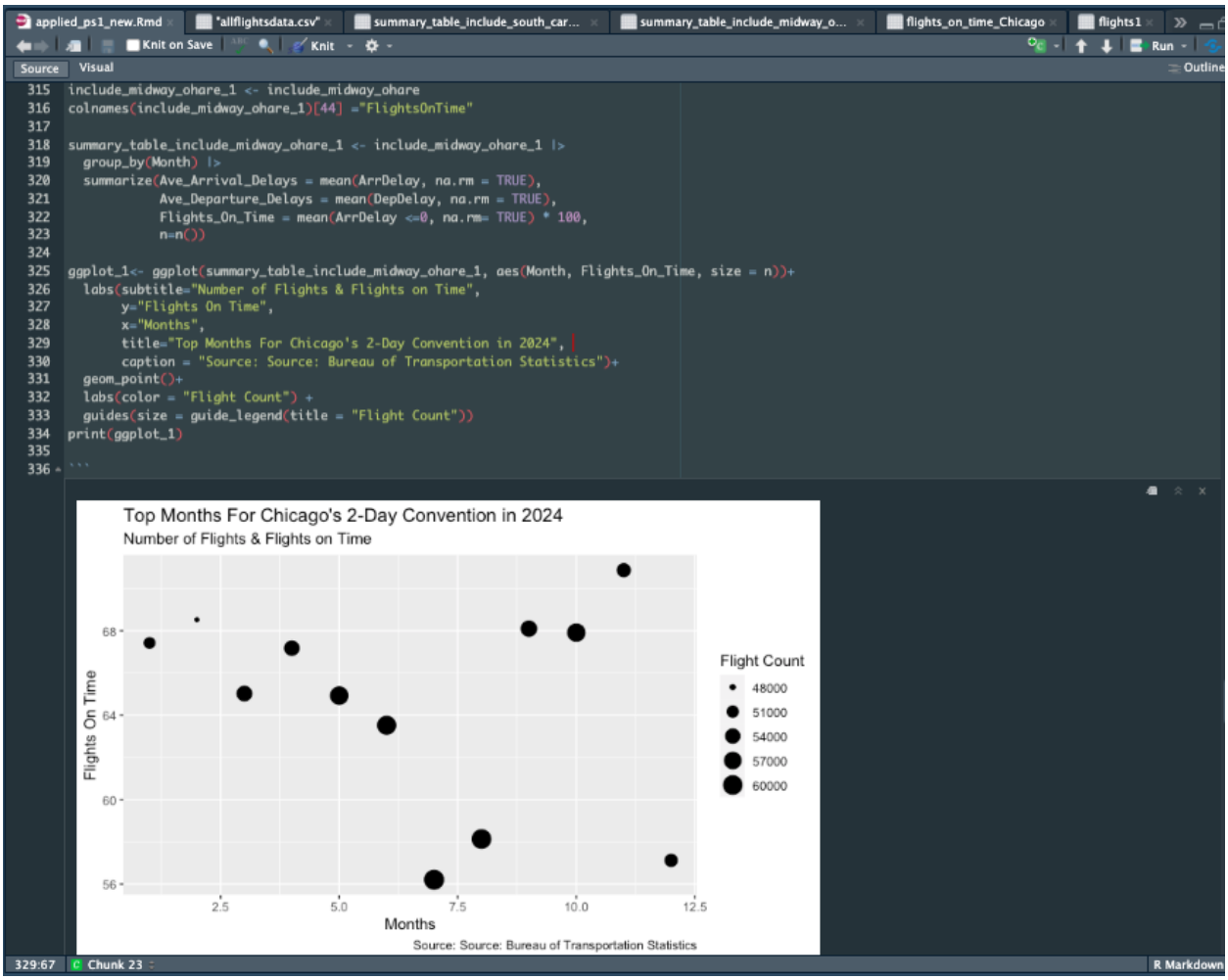


2.1.3

The convention should be held between October and November because those month have some of the largest counts of flights going to Chicago, but don't have as long delays (arrivals as well as departures) as more common months such as July or August.

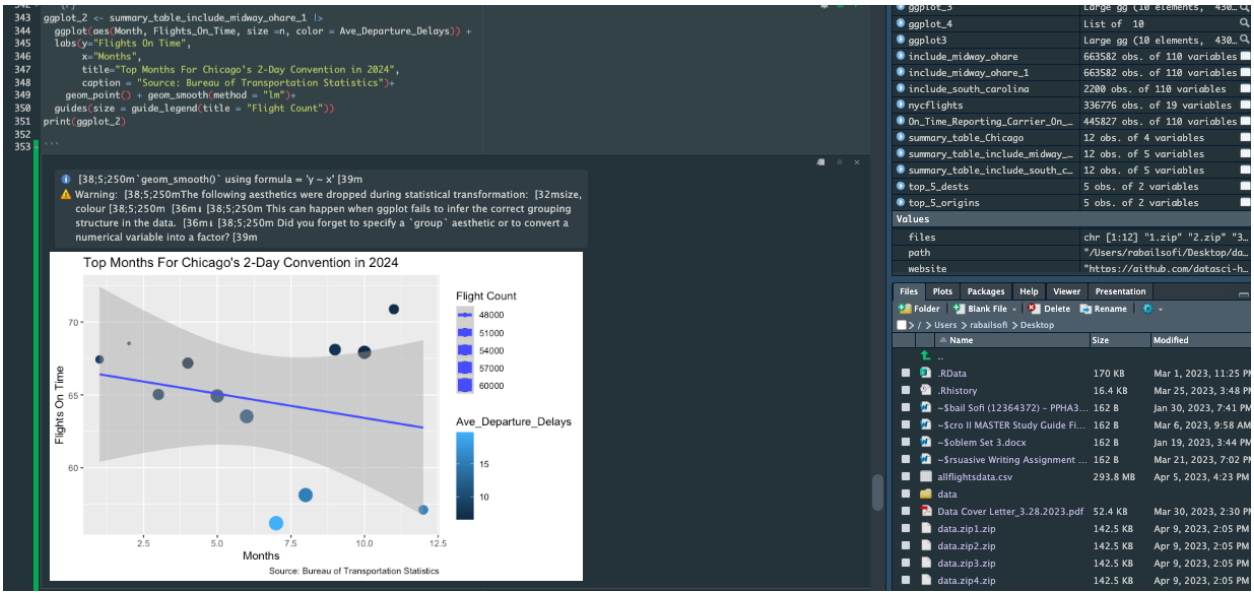
a.)

```
include_midway_ohare_1 <- include_midway_ohare

## Error in eval(expr, envir, enclos): object 'include_midway_ohare' not
found

colnames(include_midway_ohare_1)[44] ="FlightsOnTime"

## Error in colnames(include_midway_ohare_1)[44] = "FlightsOnTime": object
'include_midway_ohare_1' not found

summary_table_include_midway_ohare_1 <- include_midway_ohare_1 |>
  group_by(Month) |>
  summarize(Ave_Arrival_Delays = mean(ArrDelay, na.rm = TRUE),
            Ave_Departure_Delays = mean(DepDelay, na.rm = TRUE),
            Flights_On_Time = mean(ArrDelay <=0, na.rm= TRUE) * 100,
            n=n())

## Error in group_by(include_midway_ohare_1, Month): object
'include_midway_ohare_1' not found

ggplot_1<- ggplot(summary_table_include_midway_ohare_1, aes(Month,
Flights_On_Time, size = n))+
  labs(subtitle="Number of Flights & Flights on Time",
       y="Flights On Time",
       x="Months",
       title="Top Months For Chicago's 2-Day Convention in 2024",
       caption = "Source: Source: Bureau of Transportation Statistics")+
  geom_point()+
  labs(color = "Flight Count") +
  guides(size = guide_legend(title = "Flight Count"))

## Error in ggplot(summary_table_include_midway_ohare_1, aes(Month,
Flights_On_Time, : object 'summary_table_include_midway_ohare_1' not found

print(ggplot_1)

## Error in print(ggplot_1): object 'ggplot_1' not found
```



b.) In the richer plot, I added the average departure delays to the plot to further prove that October and November indeed have some of the lowest departure delays and would ideal options for the convention in 2024. Adding the geom_smooth allows us to visualize the outliers and in this case, its evident that October & November have extraordinary punctual flights and the flights during this time is in high volume still.

Citation: https://www.geeksforgeeks.org/how-to-create-tables-in-r/




```
ggplot_2 <- summary_table_include_midway_ohare_1 |>
  ggplot(aes(Month, Flights_On_Time, size =n, color = Ave_Departure_Delays))
+
  labs(y="Flights On Time",
       x="Months",
       title="Top Months For Chicago's 2-Day Convention in 2024",
       caption = "Source: Bureau of Transportation Statistics")+
    geom_point() + geom_smooth(method = "lm")+
  guides(size = guide_legend(title = "Flight Count"))

## Error in ggplot(summary_table_include_midway_ohare_1, aes(Month,
Flights_On_Time, : object 'summary_table_include_midway_ohare_1' not found

print(ggplot_2)

## Error in print(ggplot_2): object 'ggplot_2' not found
```

c.) I would submit the plot titled ‘ggplot_2’ because its more descriptive and visualizes the distribution within the data better.

d.) The other data sets that I would use “tourism.rds” and “OpenFlights” which tells us about flights and tourism. These data sets help us with planning the date for the convention.

Pick An Airline

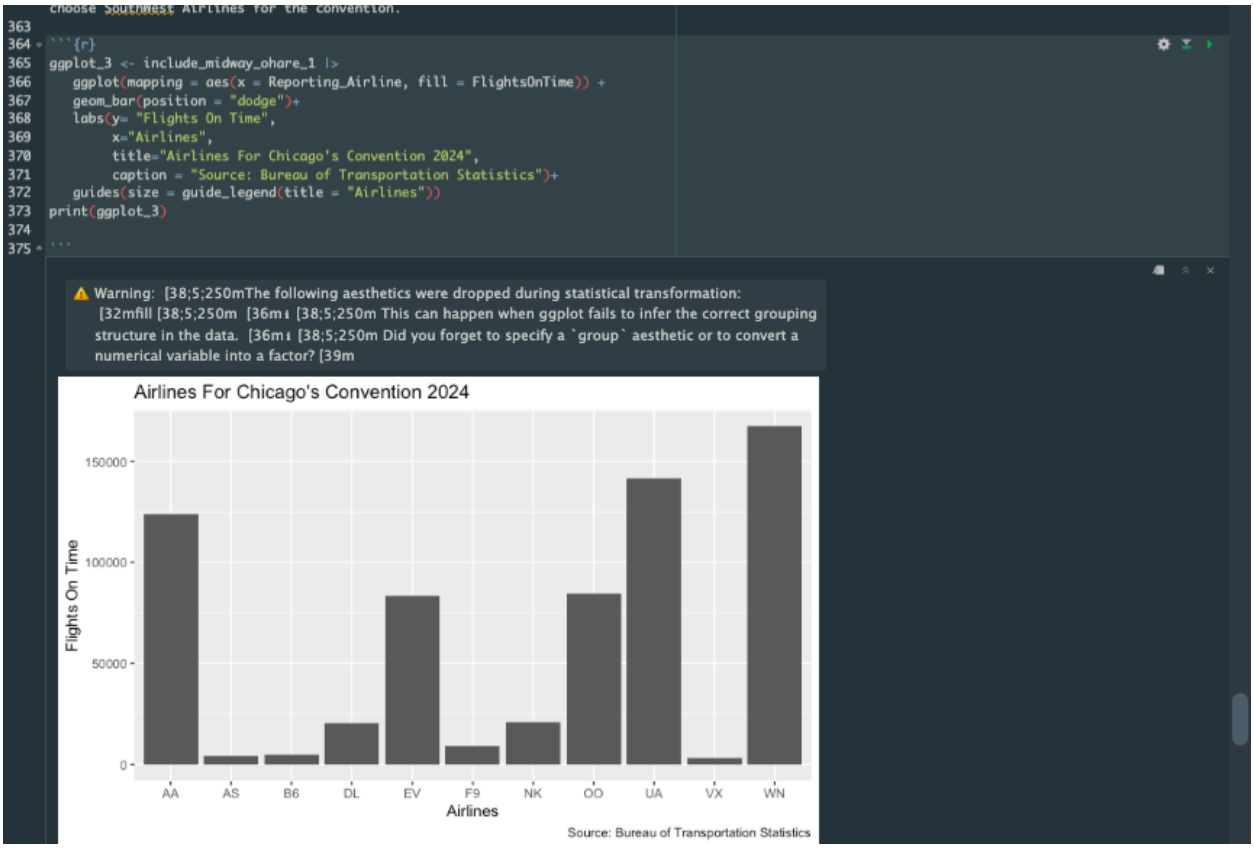
2.2.1 1.) Now that the convention date is set to be in between October and November, we can plot the most frequent airlines that fly to and from Chicago. Based on the plot created below, ggplot3, we can suggest the top airlines that is most on time is SouthWest (WN). Hence, my recommendation would be to choose SouthWest Airlines for the convention.

```r
ggplot_3 <- include_midway_ohare_1 |>
   ggplot(mapping = aes(x = Reporting_Airline, fill = FlightsOnTime)) +
   geom_bar(position = "dodge")+
   labs(y= "Flights On Time",
        x="Airlines",
        title="Airlines For Chicago's Convention 2024",
        caption = "Source: Bureau of Transportation Statistics")+
   guides(size = guide_legend(title = "Airlines"))
```

```
## Error in ggplot(include_midway_ohare_1, mapping = aes(x =
Reporting_Airline, : object 'include_midway_ohare_1' not found
```

```r
print(ggplot_3)
```

```
## Error in print(ggplot_3): object 'ggplot_3' not found
```



Reconsider the date

2.3.1 1.)

Now that the Convention has been moved to Greer, SC, we can see that the best time to host the event is February. In the plot, ggplot4, and in the summary table titles 'summary_table_include_south_carolina', we can see that February has the highest number

of flights on time, has some of the largest volumes of flights, and has relative fewer delays than some of it busier months.

This recommendation is superior than the previous recommendation because Greer, SC, has lower average departure delays and more flights on time in February than Chicago did in October-November. In February, Greer's average arrival delays are an estimated -2 hours and the average departure delays are just 3.6 hours. According to the previous recommendation, during the months of October-November, Chicago's average arrival delays are an estimated 1 hour and the average departure delays are between 6.5 to 6.8 hours. In conclusion, the new recommendation supporting the convention taking place in Greer is a superior to the previous recommendation because Greer's data proves that the travel for members outside SC will be seamless since flights to Greer in February are very smooth. The smooth travel to and from Greer in February can ensure a successful and hassel-free convention for the International Trade Organization.

```
include_south_carolina <- allflightsdata |>
  filter(OriginCityName=="Greer, SC" | DestCityName=="Greer, SC")

## Error in filter(allflightsdata, OriginCityName == "Greer, SC" |
DestCityName == : object 'allflightsdata' not found

summary_table_include_south_carolina <- include_south_carolina |>
  group_by(Month) |>
  summarize(Ave_Arrival_Delays = mean(ArrDelay, na.rm = TRUE),
            Ave_Departure_Delays = mean(DepDelay, na.rm = TRUE),
            Flights_On_Time = mean(ArrDelay <=0, na.rm= TRUE) * 100,
            n=n())

## Error in group_by(include_south_carolina, Month): object
'include_south_carolina' not found

ggplot_4 <- summary_table_include_south_carolina |>
  ggplot(aes(Month, Flights_On_Time, size =n, color = Ave_Departure_Delays))
+
  labs(y="Flights On Time",
       x="Months",
       title="Top Months For Greer's 2-Day Convention in 2024",
       caption = "Source: Bureau of Transportation Statistics")+
    geom_point() + geom_smooth(method = "lm")+
  guides(size = guide_legend(title = "Flight Count"))

## Error in ggplot(summary_table_include_south_carolina, aes(Month,
Flights_On_Time, : object 'summary_table_include_south_carolina' not found

print(ggplot_4)

## Error in print(ggplot_4): object 'ggplot_4' not found
```

***As seen in the screenshot below, 'ggplot_4' is loaded into my R studio and is running successfully on my end: