BELII_Homework3.R

Nikita

2023-10-13

```
#HW 3 by Nikita Belii

# Part 1

# QUESTION 1

options(repos = c(CRAN = "https://cran.r-project.org/"))  # I had to specify a CRAN mirror because othe
install.packages("nycflights13")
```

```
##
## The downloaded binary packages are in
##  /var/folders/q0/h3_dxphx6d3c85h9mbxhp5v00000gp/T//RtmpgtmN0k/downloaded_packages
```

```
library(nycflights13)
flights_nyc_2013 <- flights
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Count the number of flights for each airport
airport_counts <- flights_nyc_2013 %>%
  group_by(origin) %>% #groups the data by the origin column
  summarize(n_flights = n()) %>% #calculates the total number of rows for each airport
  arrange(-n_flights) #arranges in descending order

# Print out the number of unique airports and the counts
print(airport_counts)
```

```
## # A tibble: 3 x 2
##   origin n_flights
##   <chr>      <int>
## 1 EWR       120835
## 2 JFK       111279
## 3 LGA       104662
```

```
# Display the busiest airport
busiest_airport <- airport_counts[1, ]
```

```
busiest_airport
```

```
## # A tibble: 1 x 2
##   origin n_flights
##   <chr>      <int>
## 1 EWR       120835
```

```
# QUESTION 2
# Count the number of flights for each destination airport
destination_counts <- flights_nyc_2013 %>%
  group_by(dest) %>% #groups the data by the dest column
  summarize(n_flights = n()) %>%  #calculates the total number of rows for each destination airport
  arrange(-n_flights)  #arranges in descending order
print(destination_counts)
```

```
## # A tibble: 105 x 2
##    dest  n_flights
##    <chr>     <int>
##  1 ORD       17283
##  2 ATL       17215
##  3 LAX       16174
##  4 BOS       15508
##  5 MCO       14082
##  6 CLT       14064
##  7 SFO       13331
##  8 FLL       12055
##  9 MIA       11728
## 10 DCA        9705
## # i 95 more rows
```

```
# Identify the most popular destination airport
most_popular_destination <- destination_counts[1, ]
most_popular_destination
```

```
## # A tibble: 1 x 2
##   dest  n_flights
##   <chr>     <int>
## 1 ORD       17283
```

```
# QUESTION 3: How many flights departed from LGA on July 4, 2013?

# Filter the data and count the flights
flights_on_july4 <- flights_nyc_2013 %>%
  filter(origin == "LGA" & year == 2013 & month == 7 & day == 4) %>%
  summarize(n_flights = n())
print(flights_on_july4)
```

```
## # A tibble: 1 x 1
##   n_flights
##       <int>
## 1       187
```

```
# QUESTION 4: What was the busiest day of the year?
days <- flights_nyc_2013 %>%
  group_by(day) %>% #groups the data by the day column
  summarize(n_flights = n()) %>%  #calculates the total number of rows for each day
  arrange(-n_flights)  #arranges in descending order
```

```r
print(days)
```

```
## # A tibble: 31 x 2
##      day n_flights
##    <int>     <int>
##  1    18     11399
##  2    11     11359
##  3    22     11345
##  4    15     11317
##  5     8     11271
##  6    10     11227
##  7    17     11222
##  8     3     11211
##  9    21     11141
## 10    20     11111
## # i 21 more rows
```

```r
# Identify the busiest day
busiest_day <- days[1, ]
busiest_day
```

```
## # A tibble: 1 x 2
##     day n_flights
##   <int>     <int>
## 1    18     11399
```

```r
# QUESTION 5: What was the busiest month of the year?
months <- flights_nyc_2013 %>%
  group_by(month) %>% #groups the data by the month column
  summarize(n_flights = n()) %>%  #calculates the total number of rows for each month
  arrange(-n_flights)  #arranges in descending order
print(months)
```

```
## # A tibble: 12 x 2
##    month n_flights
##    <int>     <int>
##  1     7     29425
##  2     8     29327
##  3    10     28889
##  4     3     28834
##  5     5     28796
##  6     4     28330
##  7     6     28243
##  8    12     28135
##  9     9     27574
## 10    11     27268
## 11     1     27004
## 12     2     24951
```

```r
# Identify the busiest month
busiest_month <- months[1, ]
busiest_month
```

```
## # A tibble: 1 x 2
##   month n_flights
##   <int>     <int>
## 1     7     29425
```

```
# QUESTION 6: What is the longest flight in the dataset?

longest_flight <- flights_nyc_2013 %>%
  filter(air_time == max(air_time, na.rm = TRUE))  #calculates the maximum airtime from the entire data
longest_flight
```

```
## # A tibble: 1 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     3    17     1337           1335         2     1937           1836
## # i 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>
```

```
# QUESTION 7: What is the shortest flight in the dataset?

shortest_flight <- flights_nyc_2013 %>%
  filter(air_time == min(air_time, na.rm = TRUE))  #calculates the minimum airtime from the entire data
shortest_flight
```

```
## # A tibble: 2 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    16     1355           1315        40     1442           1411
## 2  2013     4    13      537            527        10      622            628
## # i 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>
```

```
# QUESTION 8: Which carrier had the largest number of flights?

# Group by carrier and count the number of flights
carrier_counts <- flights_nyc_2013 %>%
  group_by(carrier) %>%
  summarize(n_flights = n()) %>%
  arrange(-n_flights)
#Show the carrier with the most flights (the first one)
top_carrier <- carrier_counts[1, ]
top_carrier
```

```
## # A tibble: 1 x 2
##   carrier n_flights
##   <chr>       <int>
## 1 UA          58665
```

```
# QUESTION 9: Which destination (airport code) had the shortest average arr_delay?

# Group by destination and calculate the average arrival delay
avg_delays <- flights_nyc_2013 %>%
  group_by(dest) %>%
  summarize(avg_arr_delay = mean(arr_delay, na.rm = TRUE)) %>%  #calculates the average of the arr_dela
  arrange(avg_arr_delay)

# Identify the destination with the shortest average arrival delay
shortest_avg_delay_destination <- avg_delays[1, ]
```

```
shortest_avg_delay_destination
```

```
## # A tibble: 1 x 2
##   dest  avg_arr_delay
##   <chr>         <dbl>
## 1 LEX             -22
```

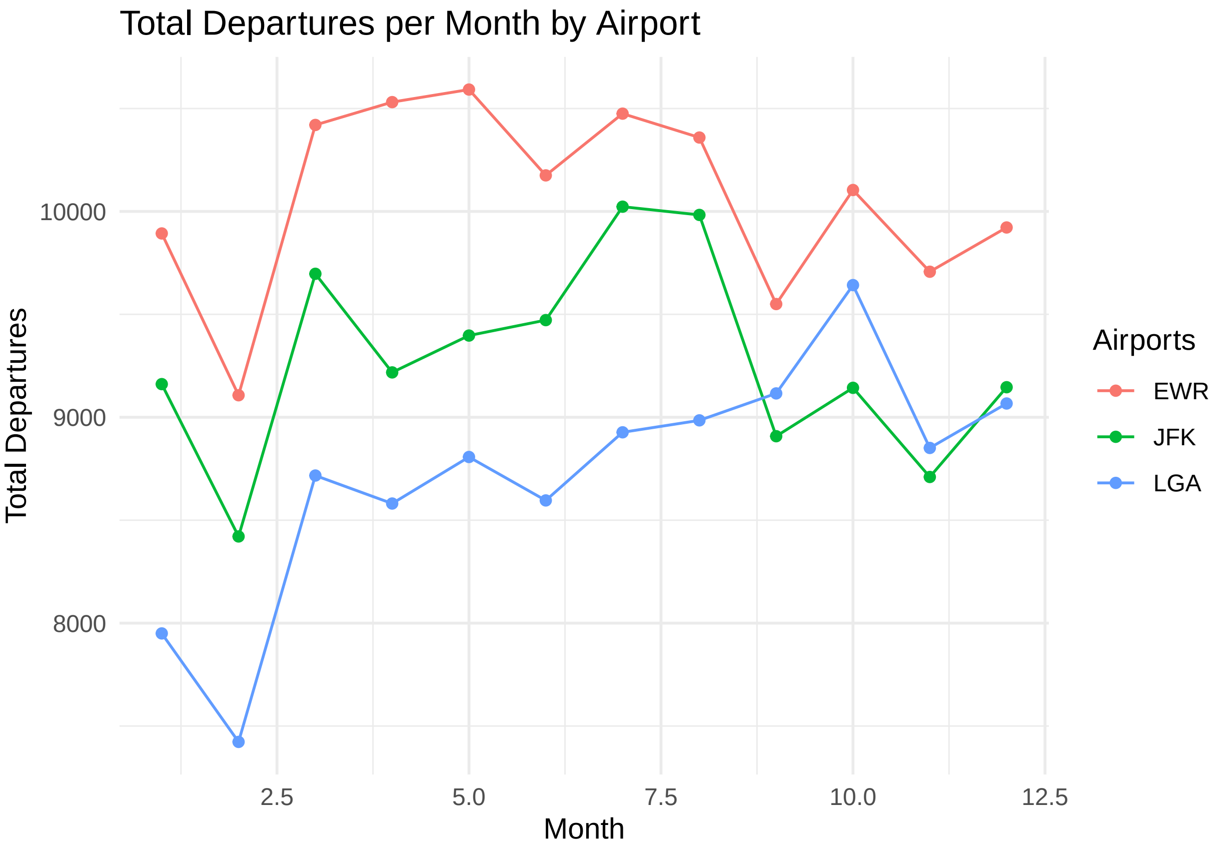
```
# QUESTION 10: What month experienced the highest average departure delay?

# Group by month and calculate the average departure delay
average_monthly_delays <- flights_nyc_2013 %>%
  group_by(month) %>%
  summarize(avg_dep_delay = mean(dep_delay, na.rm = TRUE)) %>% ##calculates the average of the dep_dela
  arrange(-avg_dep_delay)

# Identify the month with the highest average departure delay
highest_delay_month <- average_monthly_delays[1, ]
highest_delay_month
```

```
## # A tibble: 1 x 2
##   month avg_dep_delay
##   <int>         <dbl>
## 1     7          21.7
```

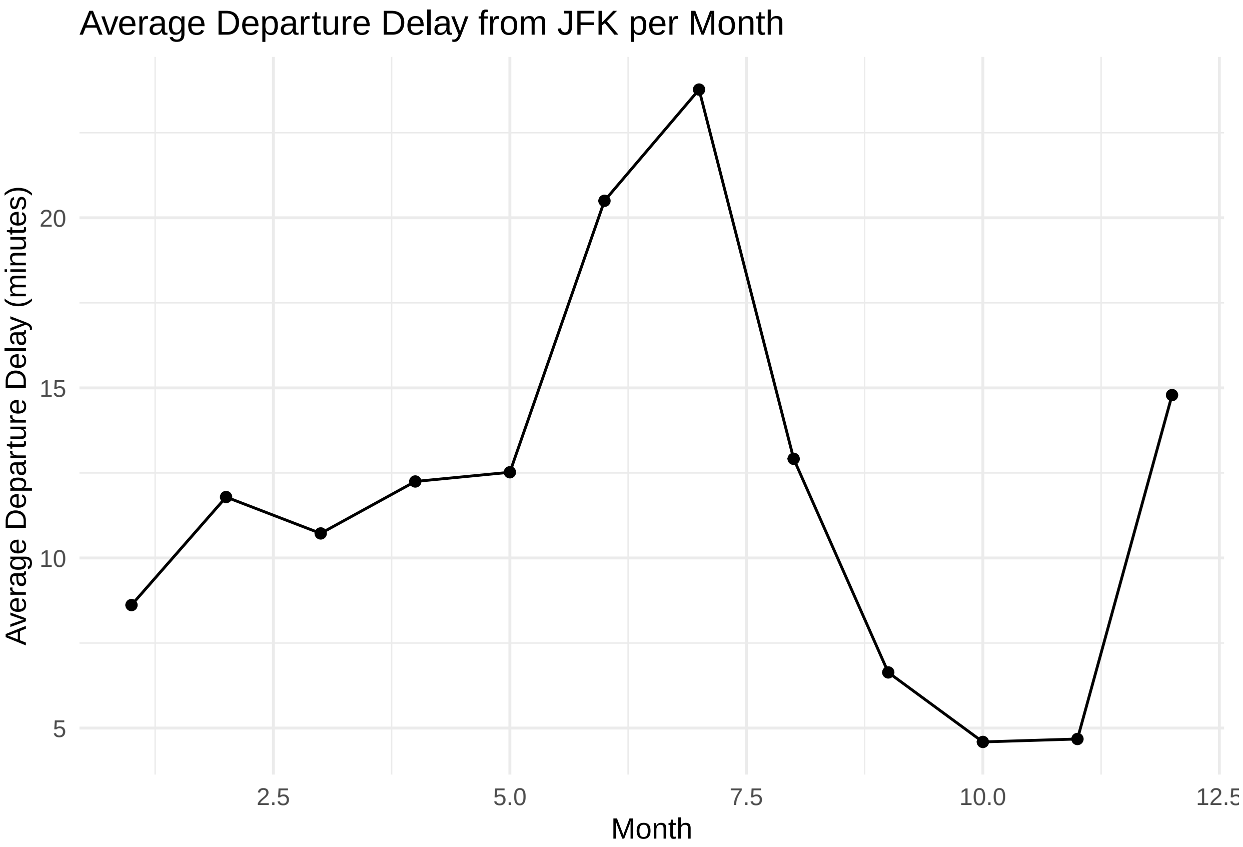
```
# Part 2

library(ggplot2)

# PLOT 1: Total number of departures per month per departure airport ("origin") [line plot?]

# Calculate total departures per month for each airport
monthly_departures <- flights_nyc_2013 %>%
  group_by(month, origin) %>%  #groups by month and origin
  summarize(total_departures = n())  #find the total number of departures
```

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

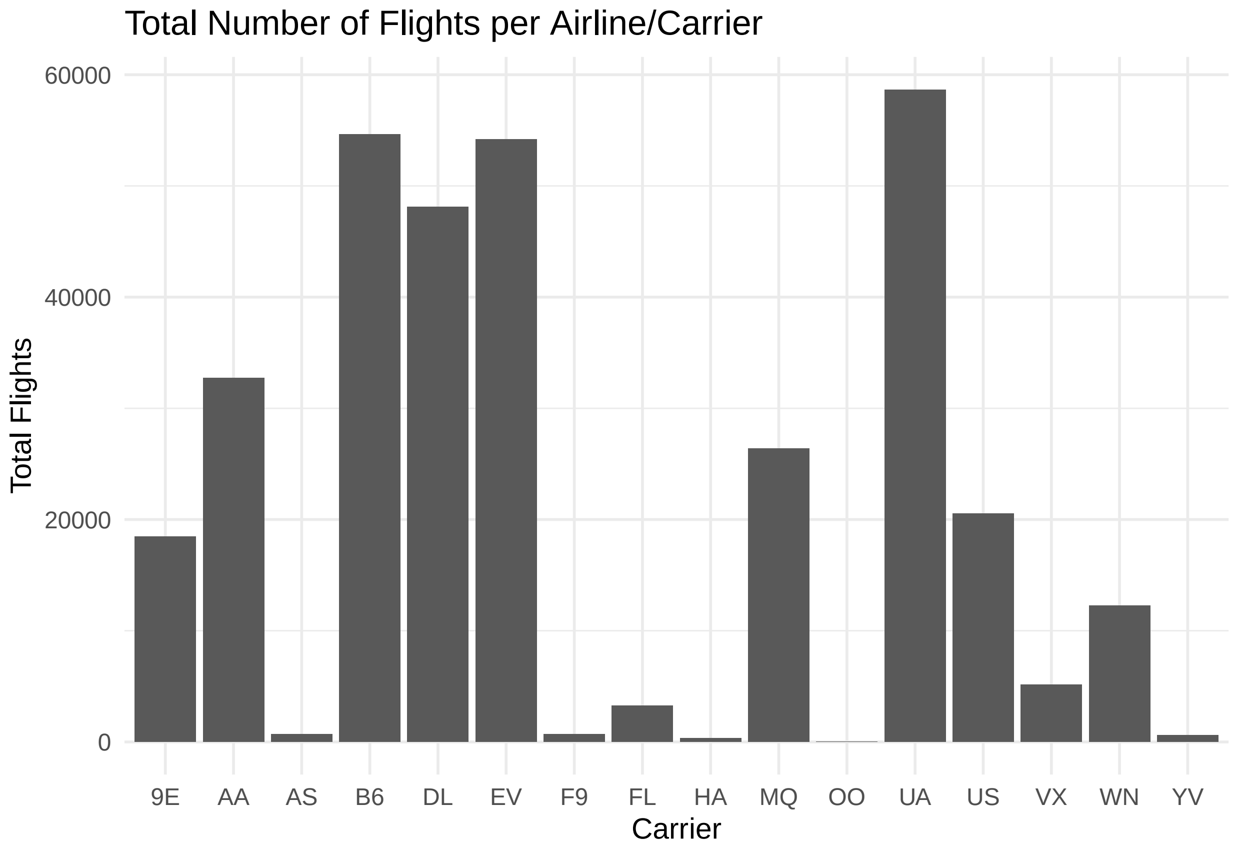
```
# Create the line plot
ggplot(data = monthly_departures, aes(x = month, y = total_departures, group = origin, color = origin))
  geom_line() +
  geom_point() +
  labs(title = "Total Departures per Month by Airport",
       x = "Month",
       y = "Total Departures",
       color = "Airports") +
  theme_minimal()  #applies a minimalistic theme
```



```
#PLOT 2: Average departure delay for flights departing from JFK per month [line plot?]

# Filter for JFK flights and calculate average departure delay per month
jfk_monthly_delays <- flights_nyc_2013 %>%
  filter(origin == "JFK") %>%
  group_by(month) %>%
  summarize(avg_dep_delay = mean(dep_delay, na.rm = TRUE))

# Create the line plot
ggplot(data = jfk_monthly_delays, aes(x = month, y = avg_dep_delay)) +
  geom_line() +
  geom_point() +
  labs(title = "Average Departure Delay from JFK per Month",
       x = "Month",
       y = "Average Departure Delay (minutes)") +
  theme_minimal()
```

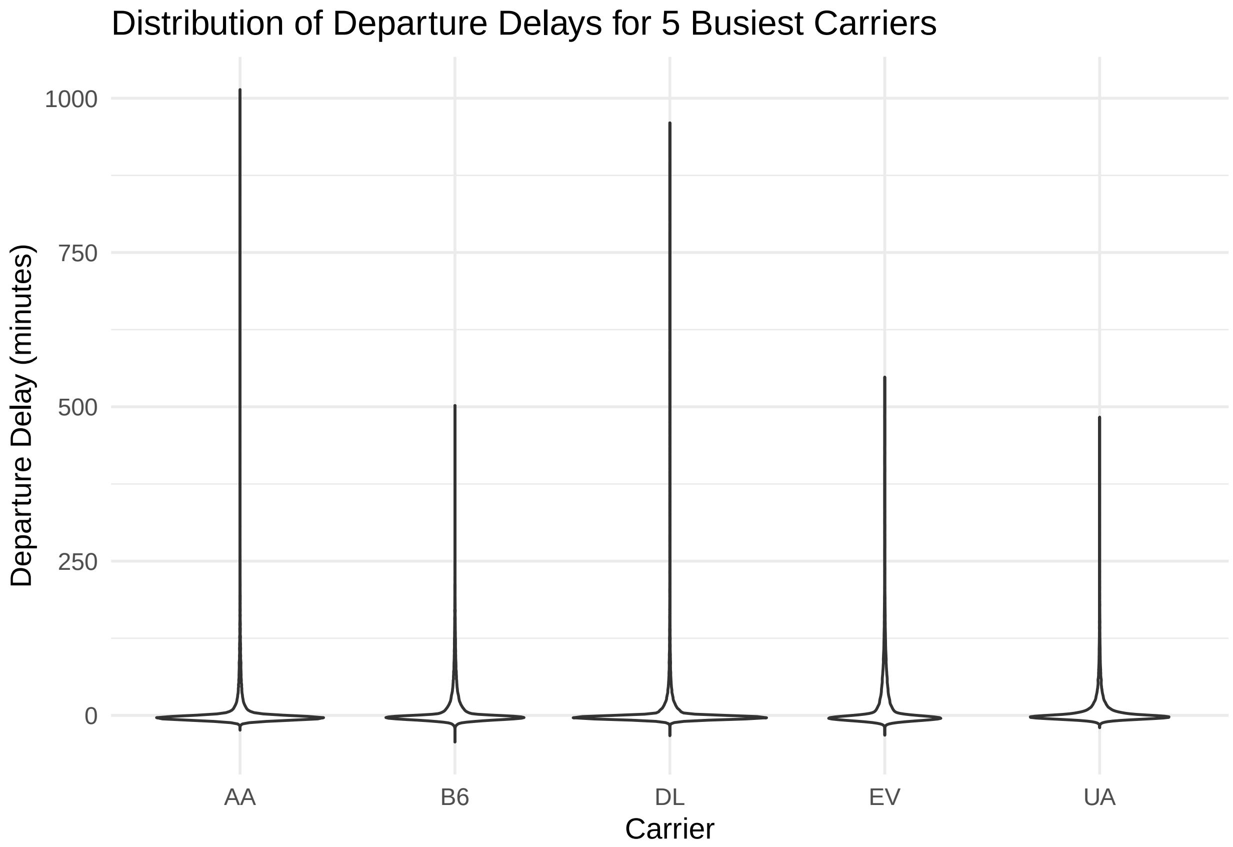


```r
# PLOT 3: Total number of flights per airline/carrier [bar plot?] [pie chart?]

# Calculate total flights per carrier
carrier_counts <- flights_nyc_2013 %>%
  group_by(carrier) %>%
  summarize(total_flights = n()) %>%
  arrange(-total_flights)

# Create the bar plot
ggplot(data = carrier_counts, aes(x = carrier, y = total_flights)) +
  geom_bar(stat = "identity") +
  labs(title = "Total Number of Flights per Airline/Carrier",
       x = "Carrier",
       y = "Total Flights") +
  theme_minimal()
```

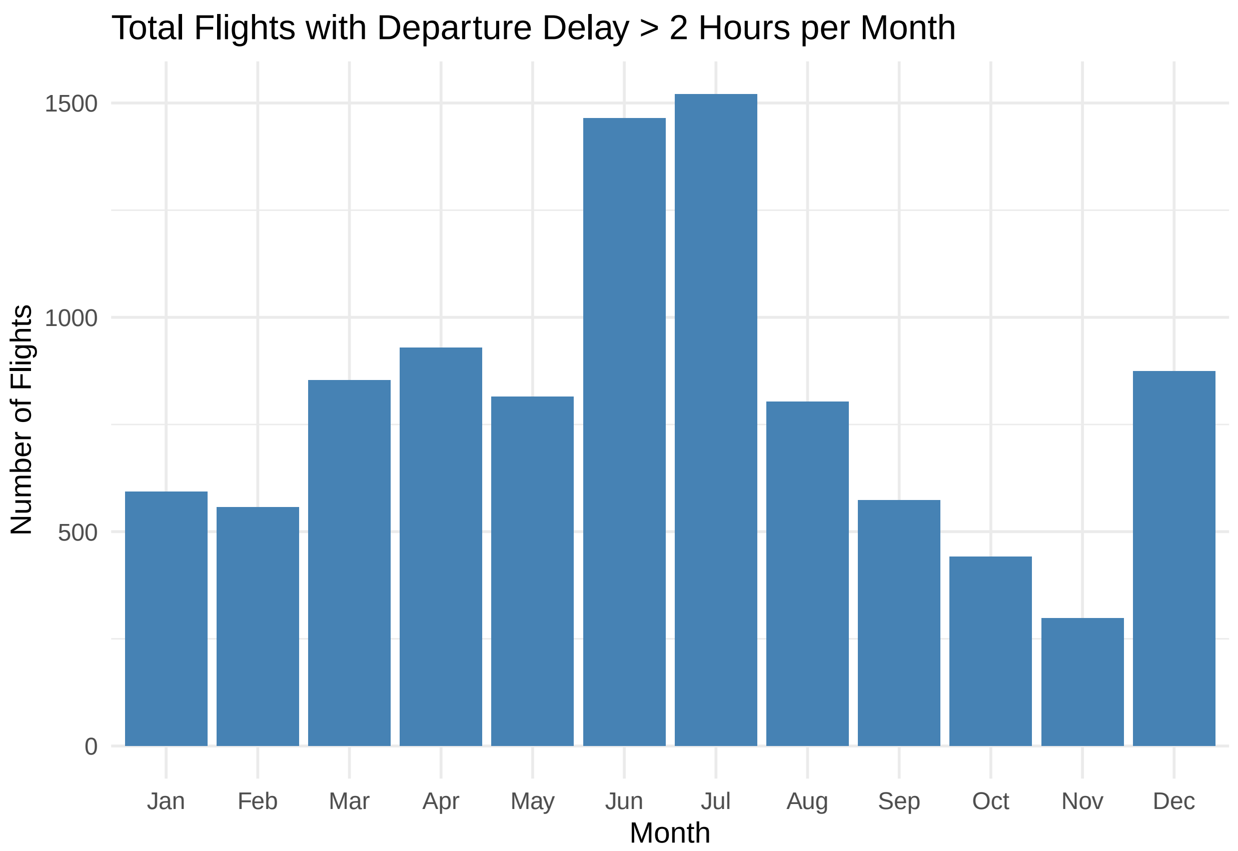


```r
# PLOT 4: Statistical distribution of departure delays for the 5 busiest carriers [box plot?] [violin p

# Identify the 5 busiest carriers
top5_carriers <- flights_nyc_2013 %>%
  count(carrier, sort = TRUE) %>%
  head(5) %>%
  pull(carrier)

# Filter data for these carriers
top5_data <- flights_nyc_2013 %>% filter(carrier %in% top5_carriers)

# Create the box plot
ggplot(data = top5_data, aes(x = carrier, y = dep_delay)) +
  geom_violin() +
  labs(title = "Distribution of Departure Delays for 5 Busiest Carriers",
       x = "Carrier",
       y = "Departure Delay (minutes)") +
  theme_minimal()
```

```
## Warning: Removed 4954 rows containing non-finite values (`stat_ydensity()`).
```



```r
# PLOT 5: Total number of flights with departure delay greater than 2 hours per month [bar plot?]

# Filter for flights with departure delay greater than 2 hours and then calculate count per month
delayed_flights_monthly <- flights_nyc_2013 %>%
  filter(dep_delay > 120) %>%
  count(month)

# Create the bar plot
ggplot(data = delayed_flights_monthly, aes(x = factor(month), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Total Flights with Departure Delay > 2 Hours per Month",
       x = "Month",
       y = "Number of Flights") +
  theme_minimal() +
  scale_x_discrete(labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "No
```



```
# Part 3 (Optional)

# BONUS QUESTION 1: Which airline had the shortest average delay per flight?

# Create the gain column
flights_nyc_2013 <- flights_nyc_2013 %>%
  mutate(gain = dep_delay - arr_delay)  # hint

# Compute the average gain per carrier
avg_gain_per_carrier <- flights_nyc_2013 %>%
  group_by(carrier) %>%
  summarize(average_gain = mean(gain, na.rm = TRUE)) %>%
  arrange(average_gain)

# Extract the carrier with the shortest average delay per flight
best_carrier <- avg_gain_per_carrier[1,]
best_carrier
```

```
## # A tibble: 1 x 2
##   carrier average_gain
##   <chr>          <dbl>
## 1 F9             -1.72
```

```
# BONUS QUESTION 2: Which airline had the longest average delay per flight?

flights_nyc_2013 <- flights_nyc_2013 %>%
  mutate(gain = dep_delay - arr_delay)
```

```r
# Compute the average gain per carrier
avg_gain_per_carrier <- flights_nyc_2013 %>%
  group_by(carrier) %>%
  summarize(average_gain = mean(gain, na.rm = TRUE)) %>%
  arrange(average_gain)

# Extract the carrier with the longest average delay per flight
longest_avg_del_carrier <- avg_gain_per_carrier %>% tail(1)
longest_avg_del_carrier
```

```
## # A tibble: 1 x 2
##   carrier average_gain
##   <chr>          <dbl>
## 1 AS              15.8
```

```r
# BONUS QUESTION 3: What was the worst day of the year (i.e., longest average dep_delay) to catch a fli

# Filter the data for JFK departures
jfk_flights <- flights_nyc_2013 %>% filter(origin == "JFK")

# Calculate the average dep_delay for each day
avg_delay_per_day <- jfk_flights %>%
  group_by(year, month, day) %>%
  summarize(average_dep_delay = mean(dep_delay, na.rm = TRUE)) %>%
  arrange(desc(average_dep_delay))
```

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

```r
# Extract the day with the longest average dep_delay
worst_day <- avg_delay_per_day[1,]
worst_day
```

```
## # A tibble: 1 x 4
## # Groups:   year, month [1]
##    year month   day average_dep_delay
##   <int> <int> <int>             <dbl>
## 1  2013     7    10              63.6
```

```r
# BONUS QUESTION 4: What percentage of flights departing from JFK had a delay of less than 10% of the t

# Filter for JFK departures
jfk_flights <- flights_nyc_2013 %>% filter(origin == "JFK")

# Create a new column representing 10% of the total flight time
flights_with_less_than_10_percent_delay <- jfk_flights %>%
  filter(!is.na(dep_delay) & !is.na(air_time), dep_delay < 0.10 * air_time)  # filter rows where the de

# Calculate the percentage
percentage <- nrow(flights_with_less_than_10_percent_delay) / nrow(jfk_flights) * 100
percentage
```

```
## [1] 77.46565
```

```r
# BONUS QUESTION 5: Which airline had the shortest number of flights delayed by more than 2 hours betwe
# Filter data for flights between May and September with dep_delay > 120
delayed_flights <- flights_nyc_2013 %>%
  filter(month %in% 5:9, dep_delay > 120)

# Count the number of the flights for each airline
delayed_counts_per_airline <- delayed_flights %>%
  group_by(carrier) %>%
  tally(sort = TRUE)

# Identify the airline with the shortest number of flights delayed by more than 2 hours
least_delayed_airline <- head(delayed_counts_per_airline, 1)
least_delayed_airline
```

```
## # A tibble: 1 x 2
##   carrier     n
##   <chr>   <int>
## 1 EV       1049
```