Mini Project 1

PSTAT 100: Spring 2024 (Instructor: Ethan P. Marzban)

AARTI GARAYE (aartigaraye)

April 23, 2024

As a part of this project, I am going to present a report/analysis of flights in CA and help visualize it through the maps.

The data for this project is spread across several files (which is fairly common in data science projects);

a series of 12 files containing flight informations for each of the 12 months in 2023 (these files all have the name CA_Flights_, where represents the month represented in the file)

a file called Carrier_Codes.csv, which includes the full names for the various airline carriers included in the dataset

a filed called Airport_Info.csv, which contains geographical information about major US airports.

Each of the CA_flights_.csv files contain the following column names (and their description):

Variable Name Description year the year of observation month the month of observation day_of_month the day of month of observation op_unique_carrier the airline carrier associated with the observation origin the airport code of the origin (i.e. point-of-departure) of the observation dest the airport code of the destination crs_dep_time the scheduled departure time dep_time the actual departure time dep_delay the amount of delay in departure; i.e. actual departure minus schedule departure (flights that departed early have a negative dep_delay value) crs_arr_time the scheduled arrival time arr_time the actual arrival time arr_delay the amount of delay in arrival; i.e. actual arrival minus schedule arrival (flights that arrived early have a negative dep_delay value) crs_elapsed_time the scheduled flight duration (in minutes) actual_elapsed_time the actual flight duration (in minutes) Additionally, all times are listed in the local time zone.

Abstract

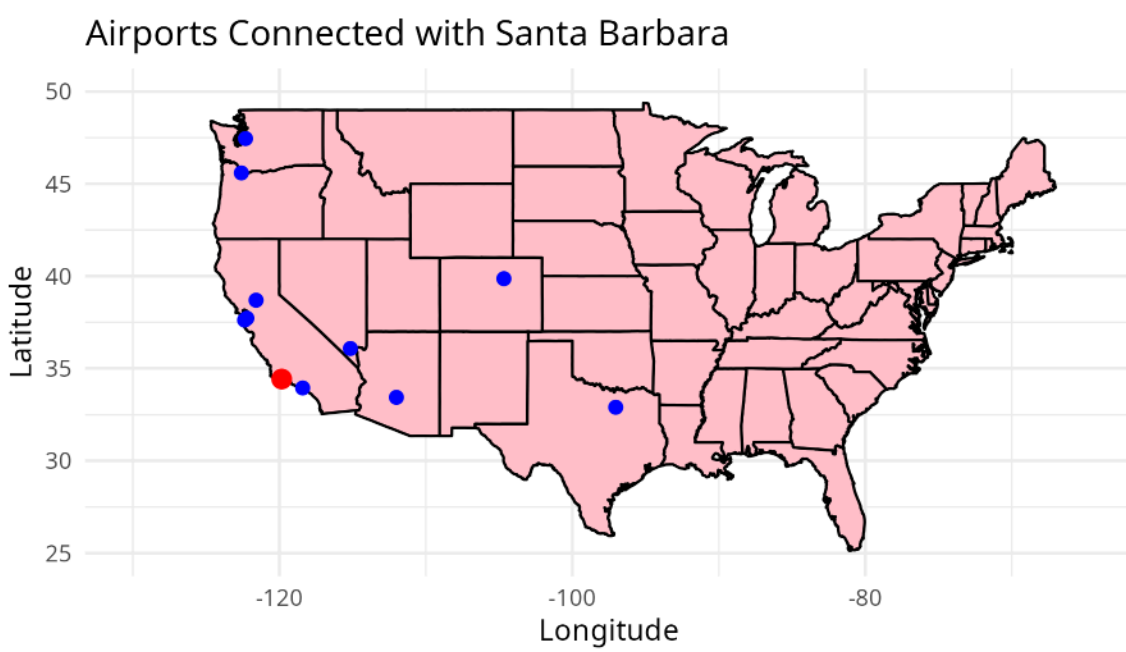
Below is a mini analytical overview of flights coming in and out of the Santa Barbara airport. All the data has been provided by the Bureau of Transportation Statistics. In this mini-project, we will explore some of the aviation data that the BTS provides. Specifically, we will examine only flights from 2023 that routed through California. The main part of this project will be visualizing this giant chunk of data using different types of graphs.

Section 1 Reference the first code section in the appendix for the source of this part of the answers. After combining the data in the combined data dataframe, we can see that the each observation is about each flight so the flights that have California airports either as origin or their destination. Missing values are generally encoded as 0 or NA, to tell if there are any missing values we can use funtioncs like is.na() or to remove any missing values we can say na.rm = TRUE

Section 2 Refer to Section 2 of the Appendix, Airports that have connection with the SBA airport are: Dallas Fort Worth International Airport, Phoenix Sky Harbor International Airport, Seattle–Tacoma International Airport, Los Angeles International Airport, Portland International Airport, San Francisco International Airport, Denver International Airport, Harry Reid International Airport Las Vegas, Oakland International Airport, and Sacramento International Airport. That makes it total of 11 airports having a connecting flights to Santa Barbara airport.

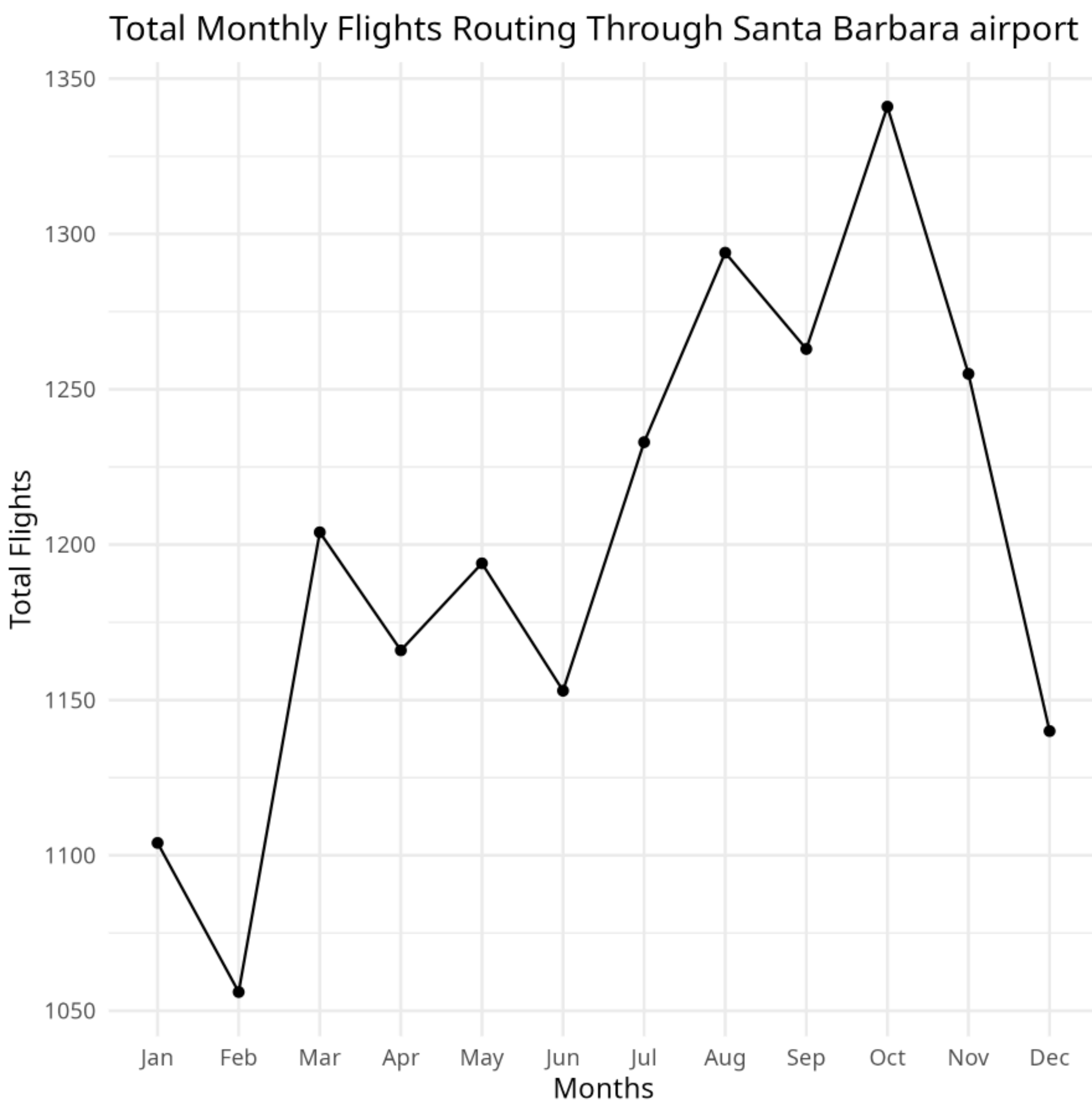
The red dot is the Santa Barbara airport and the blue airports are where there are connecting flights from santa barbara.

```r
knitr::include_graphics("airport_map.png")
```



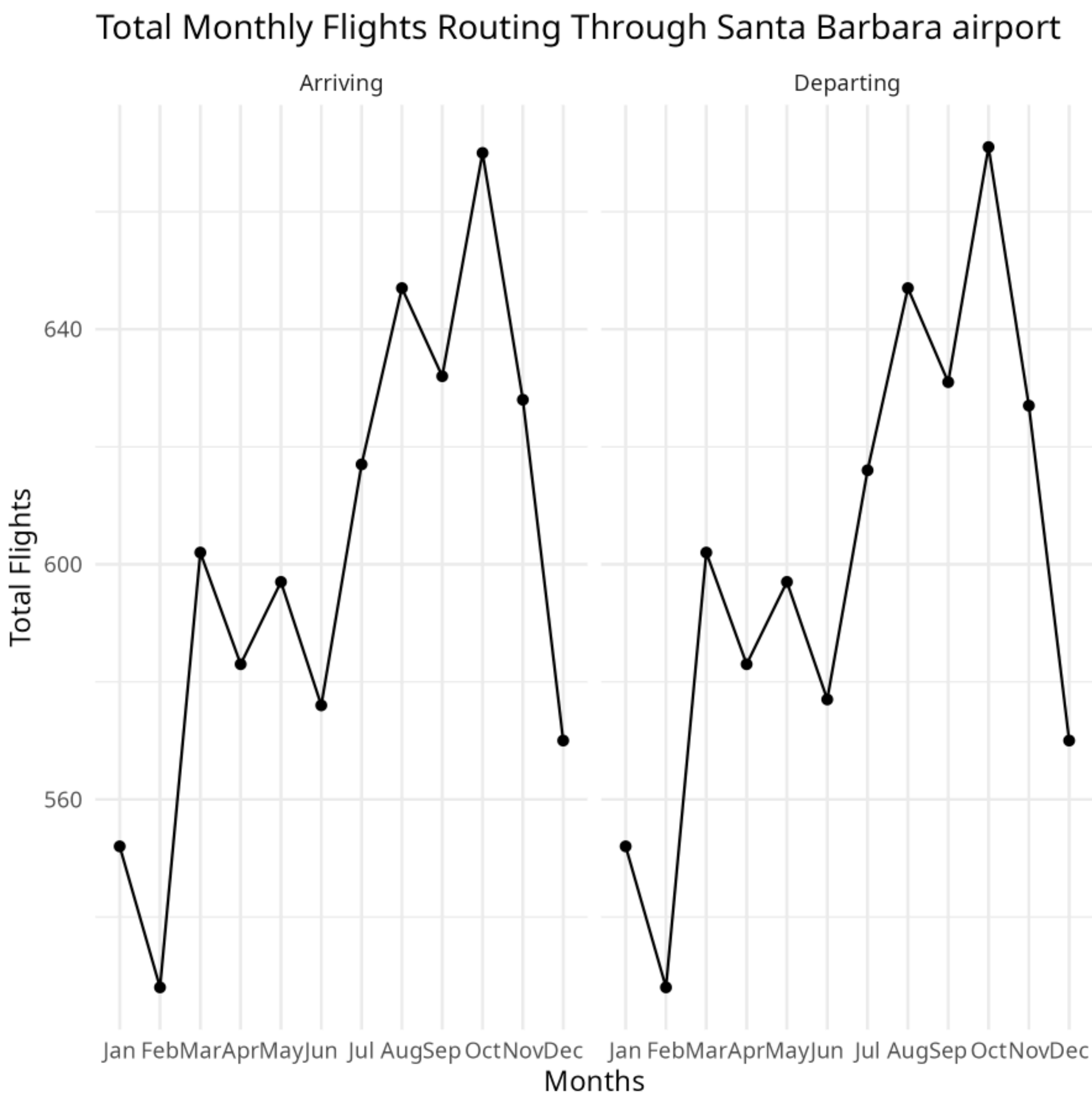
From the graph below, we can see that the trend of having the most flights leaving or landing in SBA is in June-July and in October which makes sense because students studying in UCSB are leaving for their homes and in October coming back here to resume their studies.

```r
knitr::include_graphics("sb_route_through.png")
```



Both of the graphs are almost similar except for a slight difference in months June onwards. I think it is because Santa Barbara is a vacation spot as well, many people come here to spend the summer by the beach and a lot of people have their vacation houses in SB.

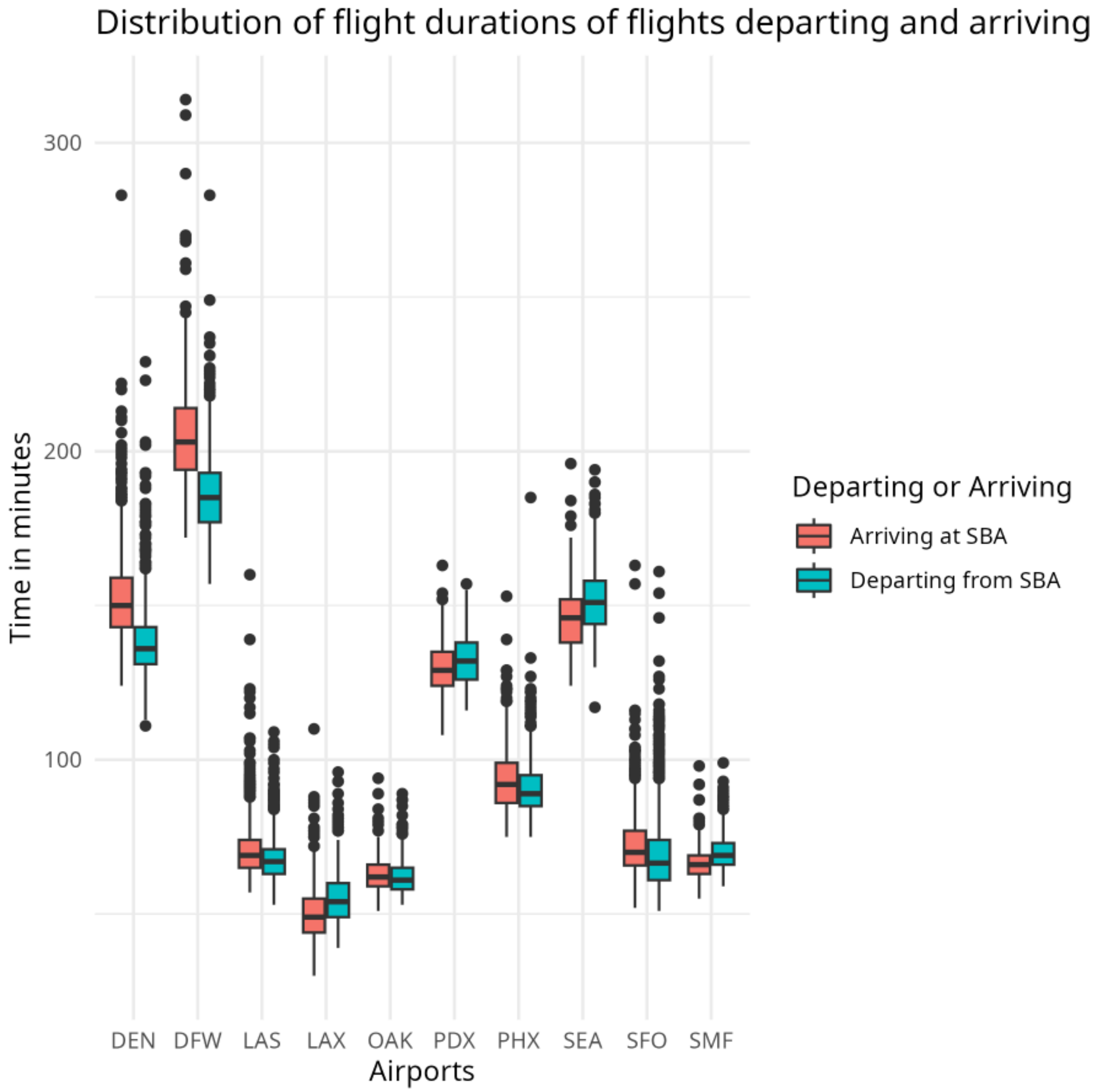
```r
knitr::include_graphics("sb_monthly_direction_route_through.png")
```



From the table below we can see the months where the there is a difference between arriving and departing flights are June, July, September, October, and November. Refer to the table in Appendix for this.

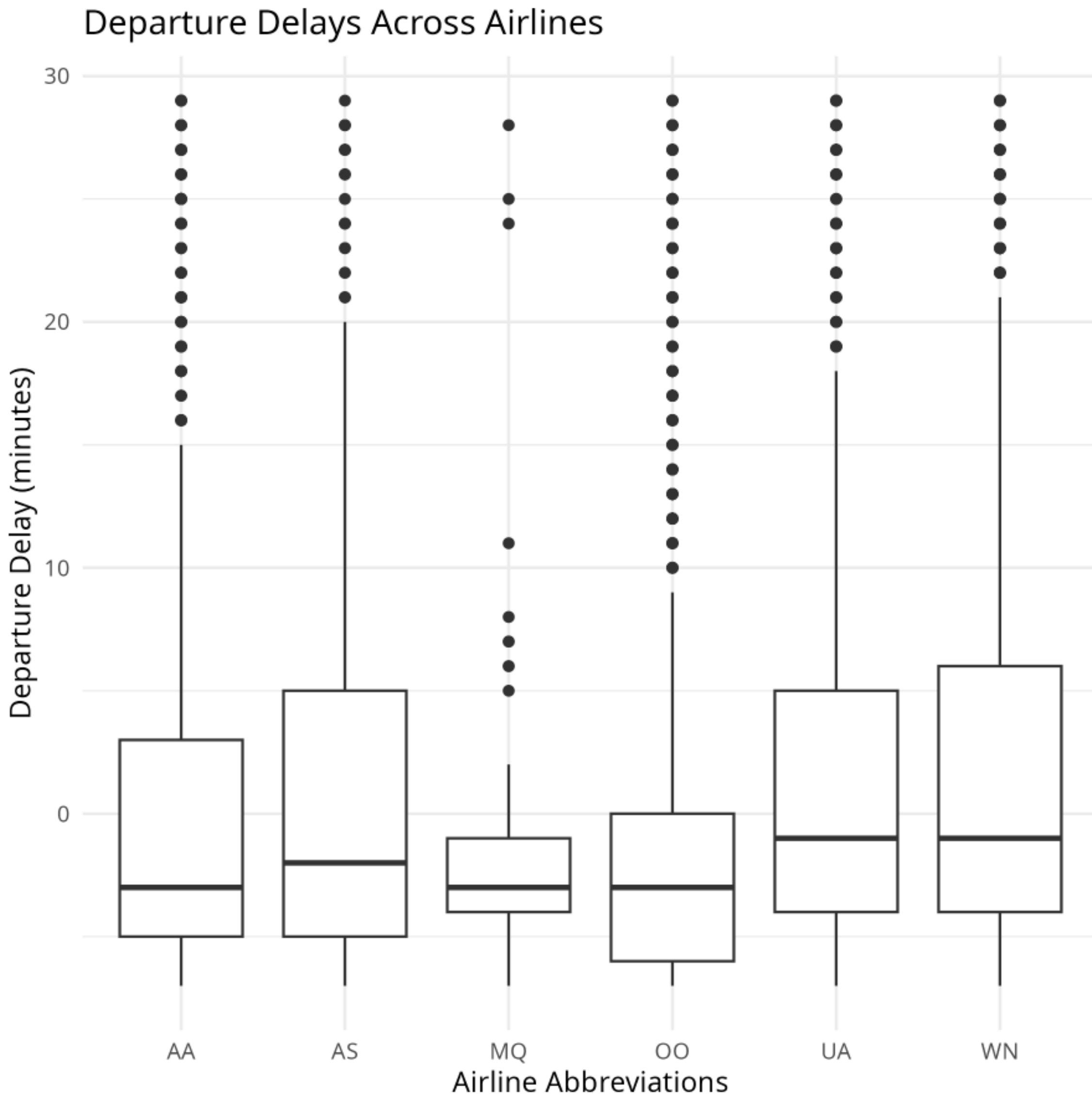
From the graph below we can see the distribution of flight durations of flights departing and arriving at Santa Barbara airport. Clearly the airports that are father away from SB, like Denver, Dallas, and even Seattle have more time duration compared to the airports that are in California like Los Angeles, and Oakland. Even Las Vegas is closer to Santa Barbara geographically making the flight duration very little.

```r
knitr::include_graphics("flight_duration.png")
```



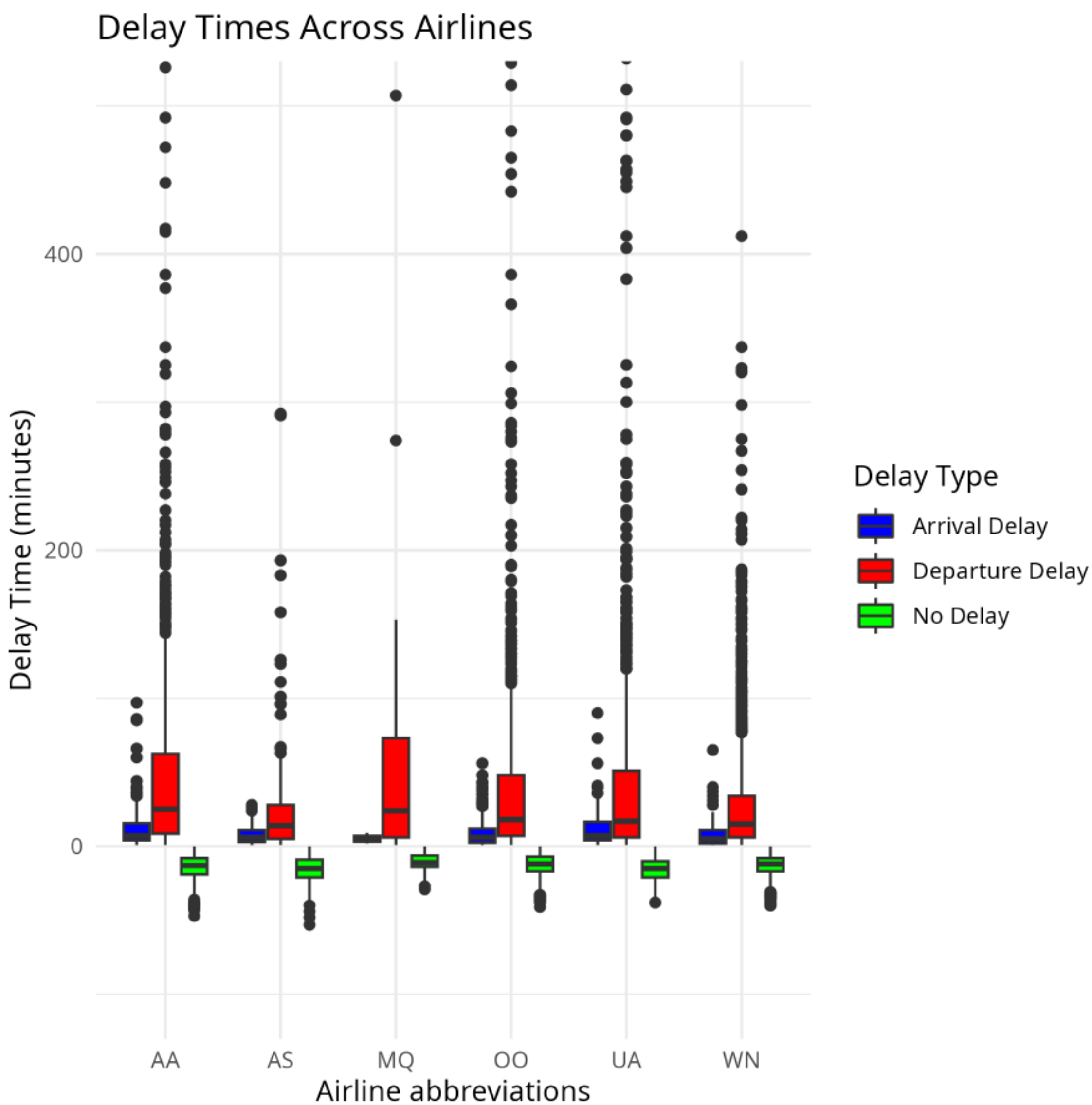
From the graph below, usually airlines have not been delayed but WN has the most delayed rates on average. On average, airlines had flights depareted after the scheduled time.

```
knitr::include_graphics("delay_boxplot.png")
```



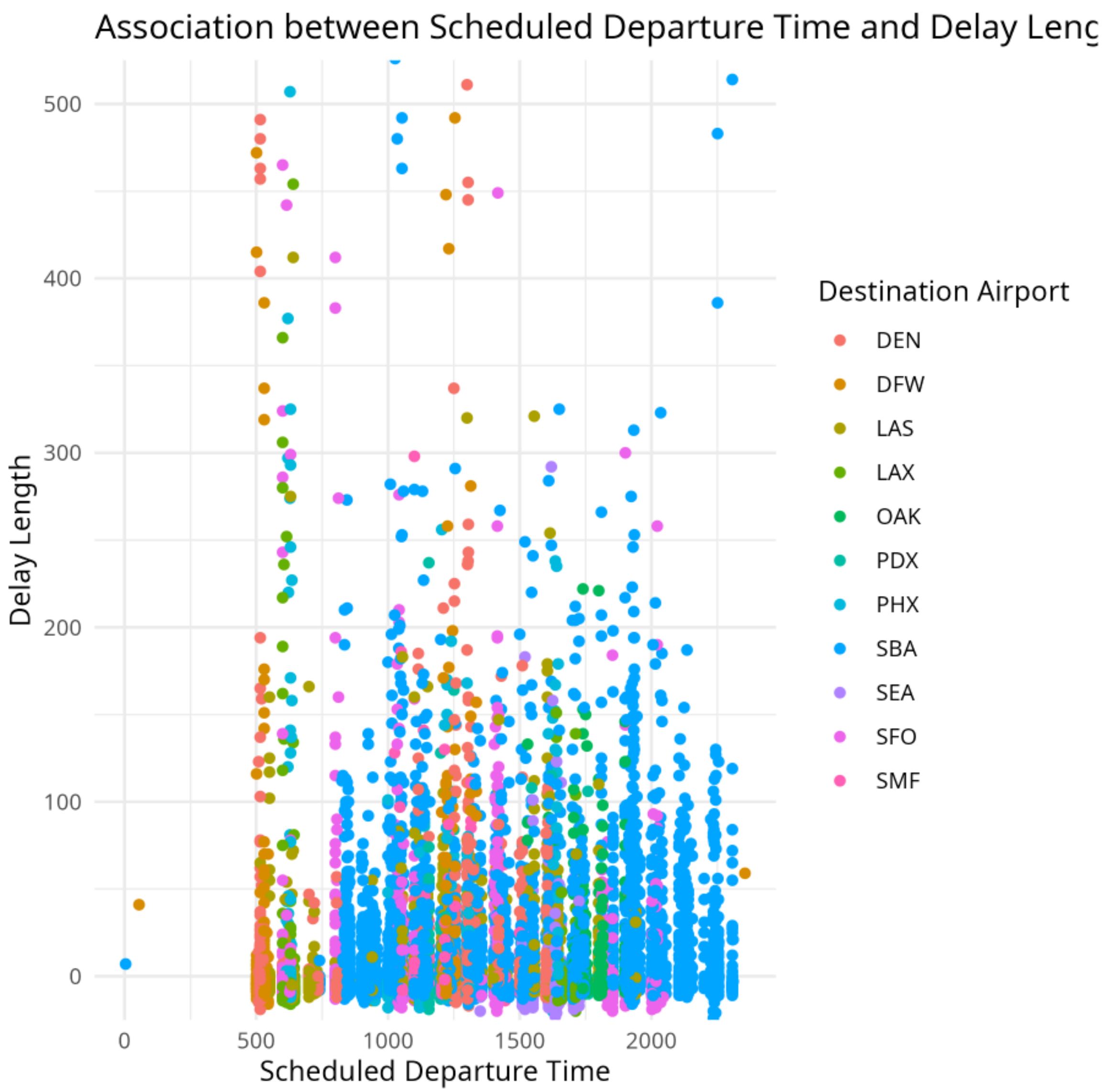
From the plot we can clearly see that there have been more departure delays from the Santa Barbara airport than arrival delays or no delays.

```
knitr::include_graphics("boxplot.png")
```



There seems to be a lot of blue overpowering the scatterplot meaning that most delays happen in Santa Barbara airport. There are 11 airports, so maybe color is not the best aesthetic but it does the job. The colors are distinct enough and the graph could be understood if just looked more closely. For flights departing from Santa Barbara there seems to be a somewhat association between the delay time and the scheduled time. If the flight is scheduled to leave around the afternoon, usually there is no delay however once it hits evenings, there is usually delays and the length of delays is more from 3-8 pm.

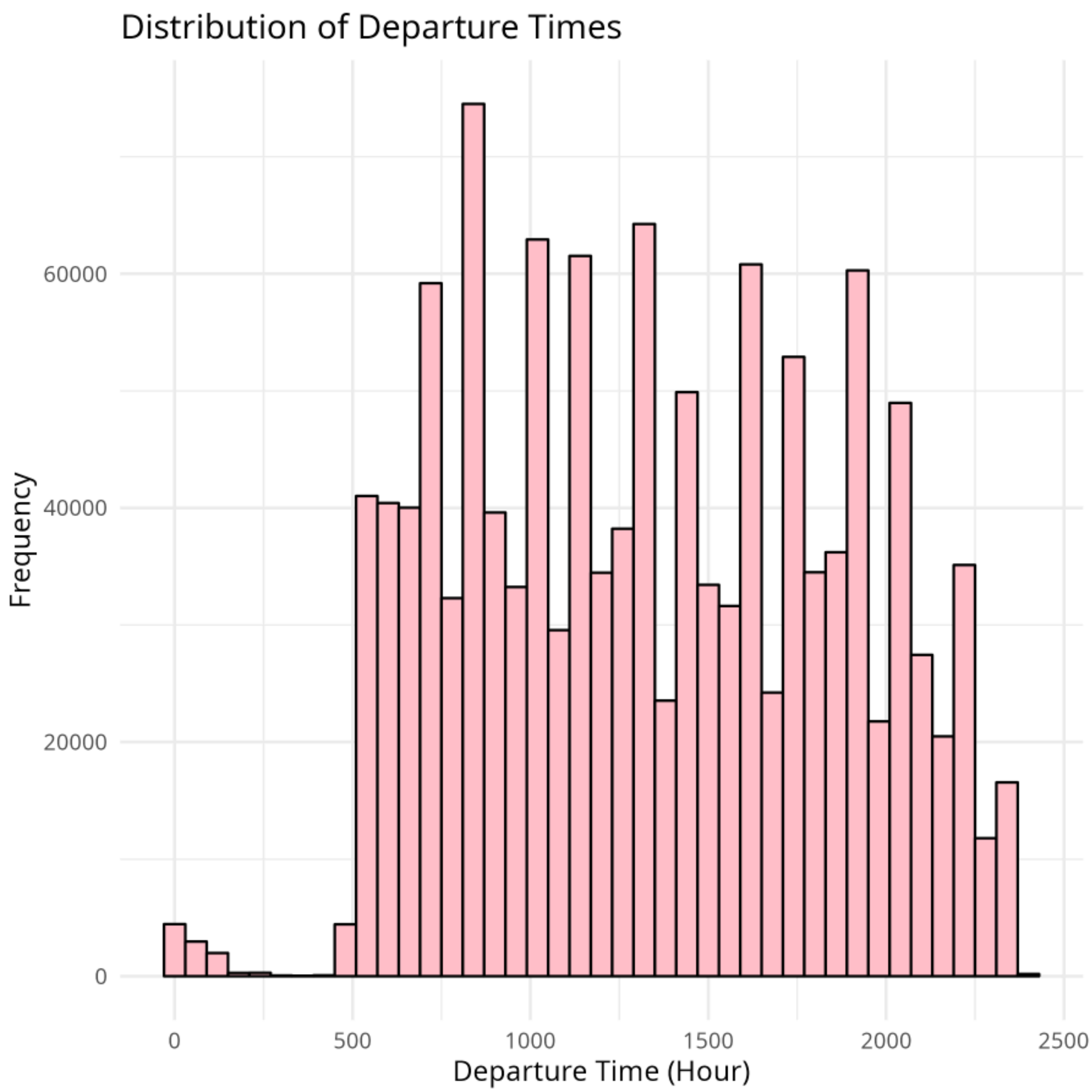
```
knitr::include_graphics("scatter.png")
```



Section 3

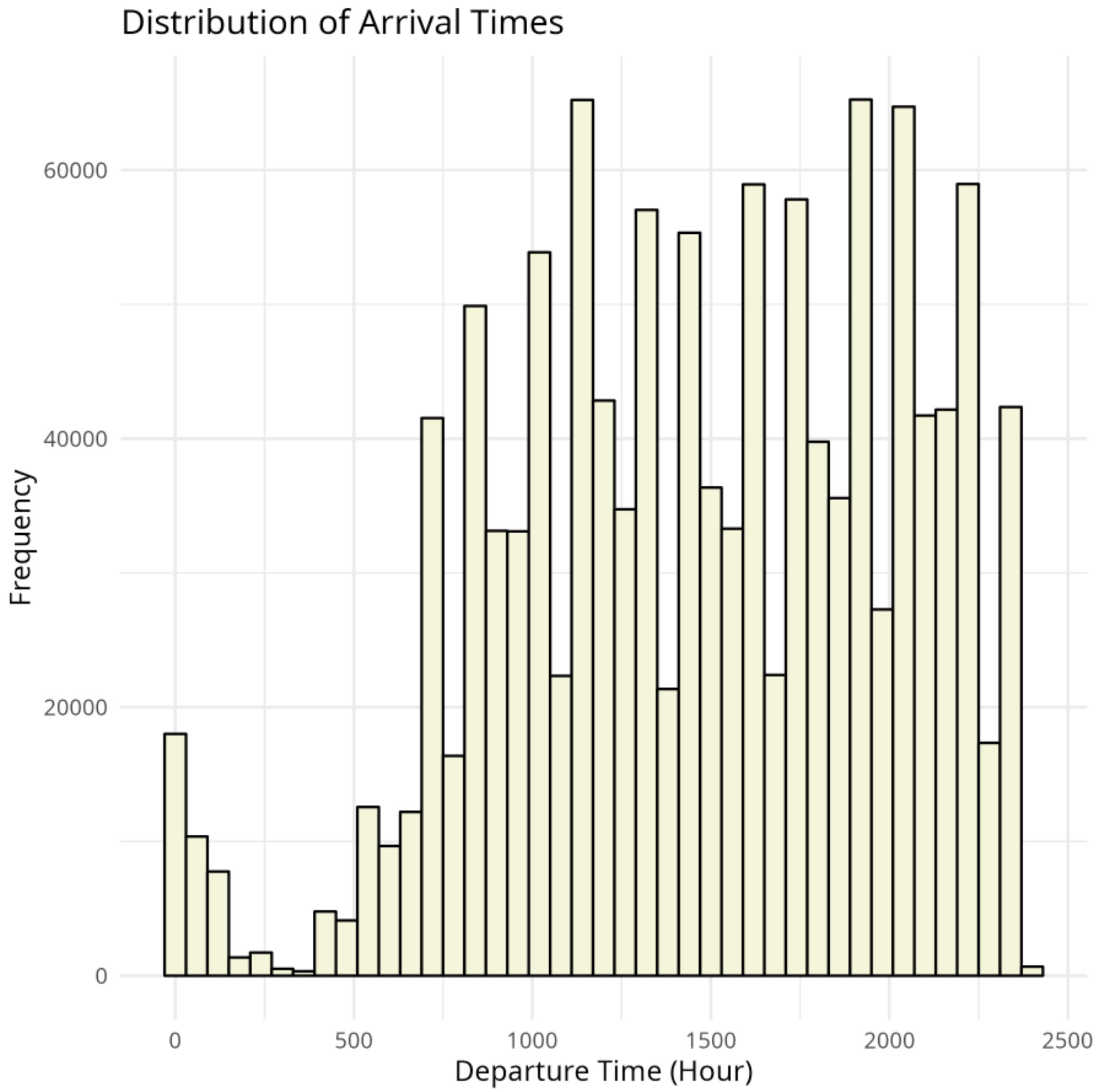
The distribution of depature times looks somewhat unimodal. The peak of depature times is around 9 am in the morning. Other usually peak hours include departure during the afternoon before 3pm and then some before 10pm.

```
knitr::include_graphics("departure_hist.png")
```



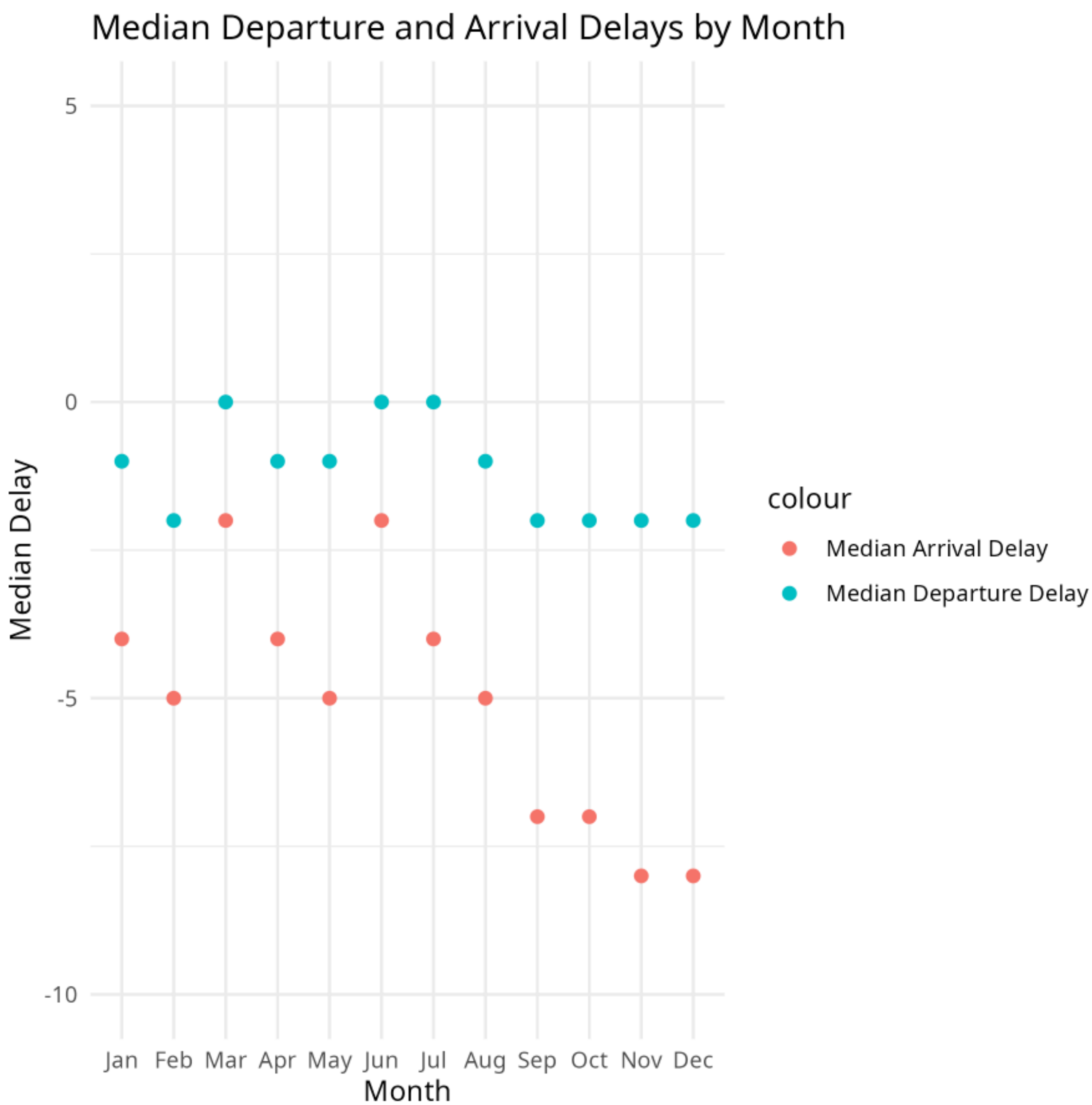
The arrival distribution is the somewhat opposite of the departure distribution which makes sense. There are three major peaks for arrival sometime around 10am, and sometime around 10 pm. We can almost classify this as multimodal distribution.

```
knitr::include_graphics("arrival_hist.png")
```



From the graph below, we can see that June was the month of most delays, departure and arrival both, and December too saw a lot of arrival delays which makes sense because it's the holiday season.

```
knitr::include_graphics("delays.png")
```



Sometimes there are more than one airports in cities, so if we filter based on cities then there might be some data that won't be included in our analysis. Filtering based on departure and arrival airports is a good idea but also to include more filters so we can go through the data more finely. Furthermore, if we include layovers, journey duration, coordinates, and other variables it would be easier to manage the big data.

Appendix

Section 1

```
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
```

```
# Set working directory if necessary
setwd("/home/jovyan/100-sp24/Mini_Projects/MP01/data")

# Load data for each month into a list
file_names <- list.files(pattern = "^CA_Flights_[A-Za-z]*.csv$")
flight_data <- lapply(file_names, read.csv)

# Combine all monthly data into a single dataframe
combined_data <- bind_rows(flight_data)
#View(combined_data)

# Load airport information
airport_info <- read.csv("/home/jovyan/100-sp24/Mini_Projects/MP01/data/Airport_Info.csv")

combined_data <- left_join(
  combined_data,
  airport_info,
  by = c("ORIGIN" = "ARPT_ID")
) %>%
  rename(
    ORIGIN_ARPT_NAME = ARPT_NAME,
    lat_origin = x,
    lon_origin = y
  ) %>%
  left_join(
    airport_info,
    by = c("DEST" = "ARPT_ID")
  ) %>%
  rename(
    DEST_ARPT_NAME = ARPT_NAME,
    lat_dest = x,
    lon_dest = y
  )

#head(combined_data, 6)

# Clean up variable types and encode months with descriptive names
combined_data$MONTH <- factor(combined_data$MONTH, labels = c("Jan", "Feb", "Mar", "Apr", "Ma
```

```r
#head(combined_data, 6)
```

Section 2

For how many flights are arriving / departing from SBA

```r
sb_flights <- combined_data %>%
  filter(ORIGIN == "SBA" | DEST == "SBA")

#print(sb_flights)

connecting_sb_airports <- unique(c(sb_flights$ORIGIN, sb_flights$DEST))

total_connecting_airports <- length(connecting_sb_airports)
paste(total_connecting_airports)
```

```
[1] "11"
```

```r
paste(connecting_sb_airports)
```

```
 [1] "DFW" "PHX" "SBA" "SEA" "LAX" "PDX" "SFO" "DEN" "LAS" "OAK" "SMF"
```

Mapping airports on the map

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v readr     2.1.4
v ggplot2   3.5.0     v stringr   1.5.1
v lubridate 1.9.3     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become e
```

```r
states <- map_data("state")

US_map <- ggplot() +
  geom_polygon(data = states,
               aes(x = long, y = lat, group = group),
               fill = "pink",
               colour = "black") +
```

```r
  coord_quickmap(xlim = c(-130, -65), ylim = c(25, 50)) +
  geom_point(data = filter(airport_info,  ARPT_ID == "DFW"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "PHX"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "SEA"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "LAX"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "PDX"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "SFO"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "DEN"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "LAS"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "OAK"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "SMF"), aes(x = x, y = y), color = "blue
  geom_point(data = filter(airport_info,  ARPT_ID == "SBA"), aes(x = x, y = y), color = "red"
  labs(title = "Airports Connected with Santa Barbara",
       x = "Longitude",
       y = "Latitude") +
  theme_minimal()

ggsave("airport_map.png", US_map, width = 6, height = 6)
```

Code for the line graph

```r
library(tidyverse)
library(ggplot2)

sb_route_through <- combined_data %>%
  filter(ORIGIN == "SBA" | DEST == "SBA")

monthly_flights <- sb_route_through %>%
  mutate(MONTH = factor(MONTH, levels = month.abb)) %>%
  group_by(MONTH) %>%
  summarise(total_flights = n())

line_graph_sb <- ggplot(monthly_flights, aes(x = MONTH, y = total_flights, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title = "Total Monthly Flights Routing Through Santa Barbara airport",
      x = "Months",
      y = "Total Flights") +
  theme_minimal()

ggsave("sb_route_through.png", line_graph_sb, width = 6, height = 6)
```

Facet based graphic

```r
library(tidyverse)
library(ggplot2)
```

```
sb_route_through <- combined_data %>%
  filter(ORIGIN == "SBA" | DEST == "SBA")

monthly_flights <- sb_route_through %>%
  mutate(MONTH = factor(MONTH, levels = month.abb)) %>%
  group_by(MONTH, takeoff_land = if_else(ORIGIN == "SBA", "Departing", "Arriving")) %>%
  summarise(total_flights = n())
```

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

```
#print(monthly_flights)

facet_wrap1 <- ggplot(monthly_flights, aes(x = MONTH, y = total_flights, group = takeoff_land
  geom_line() +
  geom_point() +
  labs(title = "Total Monthly Flights Routing Through Santa Barbara airport",
       x = "Months",
       y = "Total Flights") +
  facet_wrap(~ takeoff_land) +
  theme_minimal()

ggsave("sb_monthly_direction_route_through.png", facet_wrap1, width = 6, height = 6)
```

to make print a table to see the difference between arriving and departing flights:

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

Generating multiple barplots to assess the distribution of flight durations.

```
set.seed(123)

sb_flights <- combined_data %>%
  filter(ORIGIN == "SBA" | DEST == "SBA") %>%
  mutate(Direction = if_else(ORIGIN == "SBA", "Departing from SBA", "Arriving at SBA"))

duration_box_plot <- ggplot(sb_flights, aes(x = if_else(ORIGIN == "SBA", DEST, ORIGIN), y = A
  geom_boxplot() +
  labs(title = "Distribution of flight durations of flights departing and arriving at SBA",
       x = "Airports",
       y = "Time in minutes",
       fill = "Departing or Arriving") +
  theme_minimal()
```

```r
ggsave("flight_duration.png", duration_box_plot, width = 6, height = 6)
```

```
Warning: Removed 258 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

boxplot for delays

```r
sb_flights <- combined_data %>%
  filter(ORIGIN == "SBA" | DEST == "SBA")

delay_boxplot <- ggplot(sb_flights, aes(x = OP_UNIQUE_CARRIER, y = DEP_DELAY)) +
  geom_boxplot() +
  labs(title = "Departure Delays Across Airlines",
       x = "Airline Abbreviations",
       y = "Departure Delay (minutes)") +
  theme_minimal()

quantiles <- quantile(sb_flights$DEP_DELAY, c(0.25, 0.75), na.rm = TRUE)

delay_boxplot <- delay_boxplot + ylim(quantiles + c(-1, 1.5 * IQR(sb_flights$DEP_DELAY, na.rm


ggsave("delay_boxplot.png", delay_boxplot, width = 6, height = 6)
```

```
Warning: Removed 4428 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```r
sb_flights <- combined_data %>%
  filter(ORIGIN == "SBA" | DEST == "SBA")

sb_flights <- sb_flights %>%
  mutate(delay_type = if_else(DEP_DELAY > 0, "Departure Delay",
                              if_else(ARR_DELAY > 0, "Arrival Delay", "No Delay")))

boxplot <- ggplot(sb_flights, aes(x = OP_UNIQUE_CARRIER, y = if_else(delay_type == "Departure
  geom_boxplot() +
  labs(title = "Delay Times Across Airlines",
       x = "Airline abbreviations",
       y = "Delay Time (minutes)",
       fill = "Delay Type") +
  theme_minimal() +
  scale_fill_manual(values = c("Departure Delay" = "red", "Arrival Delay" = "blue", "No Delay
  coord_cartesian(ylim = c(-100, 500))
```

```
ggsave("boxplot.png", boxplot, width = 6, height = 6)
```

```
Warning: Removed 226 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```
scatter_plot <- ggplot(sb_flights, aes(x = CRS_DEP_TIME, y = DEP_DELAY, color = DEST)) +
  geom_point() +
  labs(title = "Association between Scheduled Departure Time and Delay Length",
       x = "Scheduled Departure Time",
       y = "Delay Length",
       color = "Destination Airport") +
  theme_minimal() +
  coord_cartesian(ylim = c(0, 500))

ggsave("scatter.png", scatter_plot, width = 6, height = 6)
```

```
Warning: Removed 200 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Section 3

Let's see how the distribution looks like

```
library(ggplot2)

departure_data <- data.frame(Departure_Time = combined_data$DEP_TIME)

departure_plot <- ggplot(departure_data, aes(x = Departure_Time)) +
  geom_histogram(binwidth = 60, fill = "pink", color = "black") +
  labs(x = "Departure Time (Hour)", y = "Frequency",
       title = "Distribution of Departure Times") +
  theme_minimal()

ggsave("departure_hist.png", departure_plot, width = 6, height = 6)
```

```
Warning: Removed 11748 rows containing non-finite outside the scale range
(`stat_bin()`).
```

Distribution of arrival times

```
library(ggplot2)

arrival_data <- data.frame(Arrival_Time = combined_data$ARR_TIME)
```

```r
arrival_plot <- ggplot(arrival_data, aes(x = Arrival_Time)) +
  geom_histogram(binwidth = 60, fill = "beige", color = "black") +
  labs(x = "Departure Time (Hour)", y = "Frequency",
       title = "Distribution of Arrival Times") +
  theme_minimal()

ggsave("arrival_hist.png", arrival_plot, width = 6, height = 6)
```

```
Warning: Removed 12591 rows containing non-finite outside the scale range
(`stat_bin()`).
```

Let's try answering this question using plots

```r
library(dplyr)

dep_delay <- combined_data %>%
  group_by(MONTH) %>%
  summarise(median_dep_delay = median(DEP_DELAY, na.rm = TRUE))

#head(dep_delay, 10)

arr_delay <- combined_data %>%
  group_by(MONTH) %>%
  summarise(median_arr_delay = median(ARR_DELAY, na.rm = TRUE))

#head(arr_delay, 10)

delayed_flights <- dep_delay %>%
  left_join(arr_delay, by = "MONTH") %>%
  select(MONTH, median_dep_delay, median_arr_delay)

#head(delayed_flights, 15)

median_delays <- ggplot(delayed_flights, aes(x = MONTH)) +
  geom_point(aes(y = median_dep_delay, color = "Median Departure Delay"), size = 2) +
  geom_point(aes(y = median_arr_delay, color = "Median Arrival Delay"), size = 2) +
  labs(title = "Median Departure and Arrival Delays by Month",
       x = "Month",
       y = "Median Delay") +
  coord_cartesian(ylim = c(-10, 5)) +
  theme_minimal()

ggsave("delays.png", median_delays, width = 6, height = 6)
```

Sources:

Google for finding and verifying the names of the airports with the abbreviations.

BTS for the statistical data