Thisisafunnygroupname's Project Proposal

Richard Zhou

Adam Rui

Jonathan Darius

Ojasvi Godha

Ryan Huang

Isaac Kang

Installation and Packages

```
# install.packages("tidyverse")
# install.packages("nycflights13")
# install.packages("nycflights23")
# install.packages("dplyr")
# install.packages("gridExtra")

library("tidyverse")
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
## Warning: package 'tibble' was built under R version 4.4.2
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
```

```
## Warning: package 'readr' was built under R version 4.4.2
```

```
## Warning: package 'purrr' was built under R version 4.4.2
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
## Warning: package 'stringr' was built under R version 4.4.2
```

```
## Warning: package 'forcats' was built under R version 4.4.2
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── 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 becom
e errors
```

```
library("nycflights13")
```

```
## Warning: package 'nycflights13' was built under R version 4.4.3
```

```
library("nycflights23")
```

```
## Warning: package 'nycflights23' was built under R version 4.4.3
```

```
##
## Attaching package: 'nycflights23'
##
## The following objects are masked from 'package:nycflights13':
##
##     airlines, airports, flights, planes, weather
```

```
library("dplyr")
library("gridExtra")
```

```
## Warning: package 'gridExtra' was built under R version 4.4.3
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

Objectives:

Our overall objective of this project is to analyze and compare flight data from 2013 and 2023 using the nycflights13 and nycflights23 packages. We aim to identify patterns, trends, and key factors affecting flight performance, delays, and operations over time. Our research aims to answer the following questions:

1. Have flight delays improved or gotten worse between 2013 and 2023?
2. To expand on Question #1, have individual airlines gotten better or worse with delays over time?
3. How does weather impact flight performance, and has it changed over time?
4. Which airports have seen the largest increase or decrease in traffic?
5. Have there been any big changes in popular destinations from NYC between 2013 and 2023?
6. Is there a seasonal pattern to flight delays? (Such as holidays)
7. What are the most reliable airlines, and has this changed over time?

Datasets:

[Include your exploratory data analysis and some visualization! Make sure the data is reasonably clean and contains enough information to answer your questions. Which variables in the table are most relevant to your questions? Do you plan to acquire additional datasets?]

We plan on using the packages nycflights13 and nycflights23, using the datasets provided in both. Specifically, we will be focusing on flights, airlines, airports, and weather from both packages.

```
# listing dimensions of datasets
dimensions <- tibble(name = character(), rows = integer(), cols = integer())

dimensions <- add_row(dimensions, name = "flights_2013", rows = nrow(nycflights13::flights), col
s = ncol(nycflights13::flights))
dimensions <- add_row(dimensions, name = "flights_2023", rows = nrow(nycflights23::flights), col
s = ncol(nycflights23::flights))

dimensions <- add_row(dimensions, name = "airlines_2013", rows = nrow(nycflights13::airlines), c
ols = ncol(nycflights13::airlines))
dimensions <- add_row(dimensions, name = "airlines_2023", rows = nrow(nycflights23::airlines), c
ols = ncol(nycflights23::airlines))

dimensions <- add_row(dimensions, name = "airports_2013", rows = nrow(nycflights13::airports), c
ols = ncol(nycflights13::airports))
dimensions <- add_row(dimensions, name = "airports_2023", rows = nrow(nycflights23::airports), c
ols = ncol(nycflights23::airports))

dimensions <- add_row(dimensions, name = "weather_2013", rows = nrow(nycflights13::weather), col
s = ncol(nycflights13::weather))
dimensions <- add_row(dimensions, name = "weather_2023", rows = nrow(nycflights23::weather), col
s = ncol(nycflights23::weather))

print(dimensions)
```

```
## # A tibble: 8 × 3
##   name            rows  cols
##   <chr>          <int> <int>
## 1 flights_2013  336776    19
## 2 flights_2023  435352    19
## 3 airlines_2013     16     2
## 4 airlines_2023     14     2
## 5 airports_2013   1458     8
## 6 airports_2023   1255     8
## 7 weather_2013   26115    15
## 8 weather_2023   26207    15
```

```
# cleaning dataset
flights_combined <- bind_rows(nycflights13::flights %>% mutate(year = 2013), nycflights23::fligh
ts %>% mutate(year = 2023))

flights_clean <- flights_combined %>%
  filter(!is.na(dep_delay), !is.na(arr_delay)) %>%
  left_join(nycflights13::airlines, by = "carrier") %>%
  left_join(nycflights13::airports, by = c("dest" = "faa"))
```

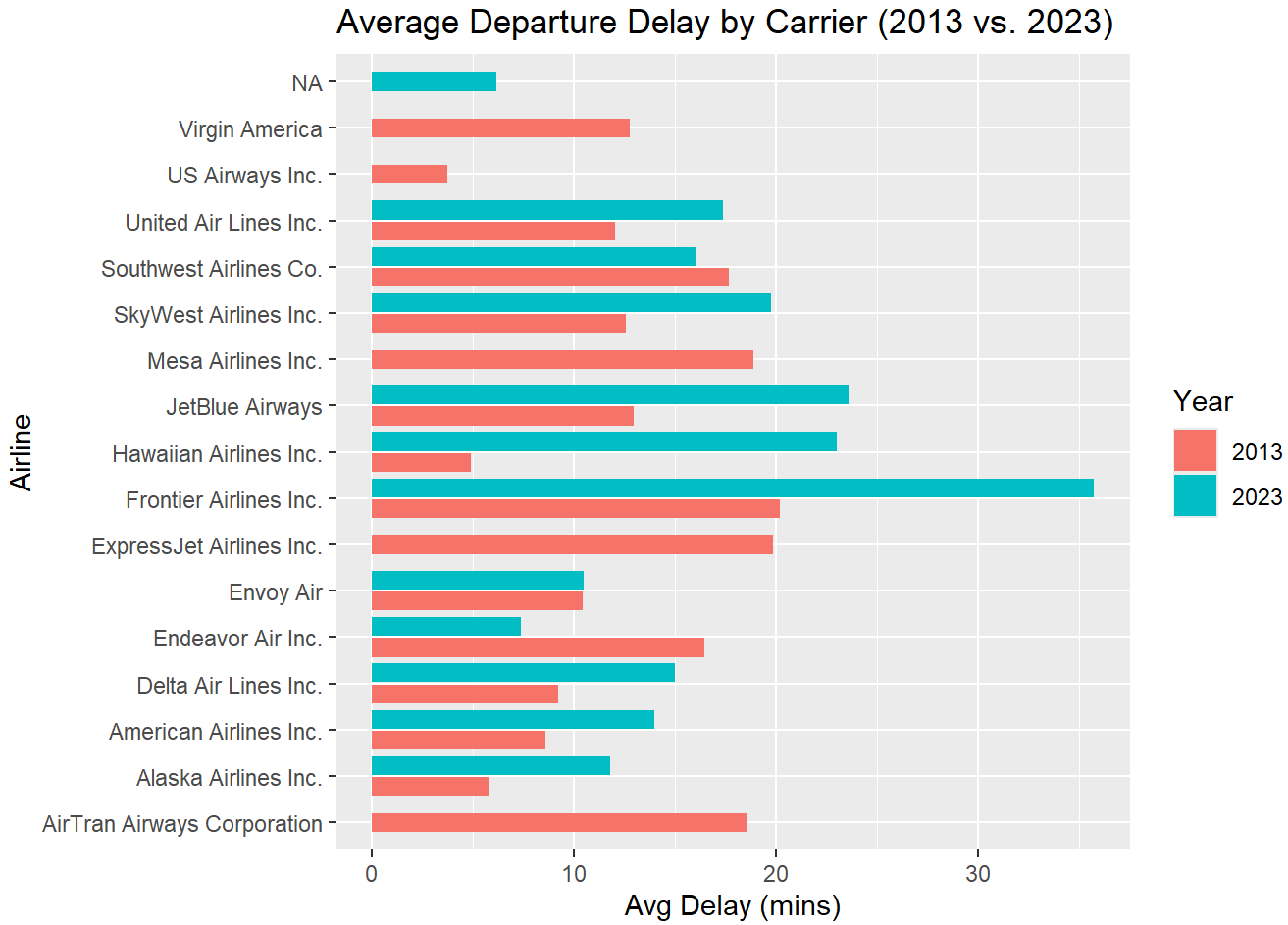
Preliminary Data Analysis

1. General Summary of Departure Delays, and Comparison of delay from year to year based on airline.

```
dep_delay_summary <- flights_clean %>%
  group_by(year) %>%
  summarise(
    avg_dep_delay = mean(dep_delay, na.rm = TRUE), #calculates the average delay for the entire
year
    median_dep_delay = median(dep_delay, na.rm = TRUE),
    perc_flights_delayed = mean(dep_delay > 15) * 100
  )

print(dep_delay_summary)
```

```
## # A tibble: 2 × 4
##    year avg_dep_delay median_dep_delay perc_flights_delayed
##   <dbl>         <dbl>            <dbl>                <dbl>
## 1  2013          12.6               -2                 21.5
## 2  2023          13.7               -2                 20.9
```

```
delay_by_carrier <- flights_clean %>%
  group_by(name.x, year) %>% #specifies to calculate the average fo each airline as well
  summarise(avg_dep_delay = mean(dep_delay), .groups = 'drop')

ggplot(delay_by_carrier, aes(x = name.x, y = avg_dep_delay, fill = factor(year))) +
  geom_col(position = position_dodge2(width = 0.8, preserve = "single")) + #puts the years side by side for each airline
  coord_flip() +
  labs(title = "Average Departure Delay by Carrier (2013 vs. 2023)", x = "Airline", y = "Avg Delay (mins)", fill = "Year")
```



- This table shows the average and median departure delays for 2013 and 2023, along with the percentage of flights delayed by more than 15 minutes. It gives an initial view if flight punctuality has improved or worsened over time.

2. Summary of Arrival Delays

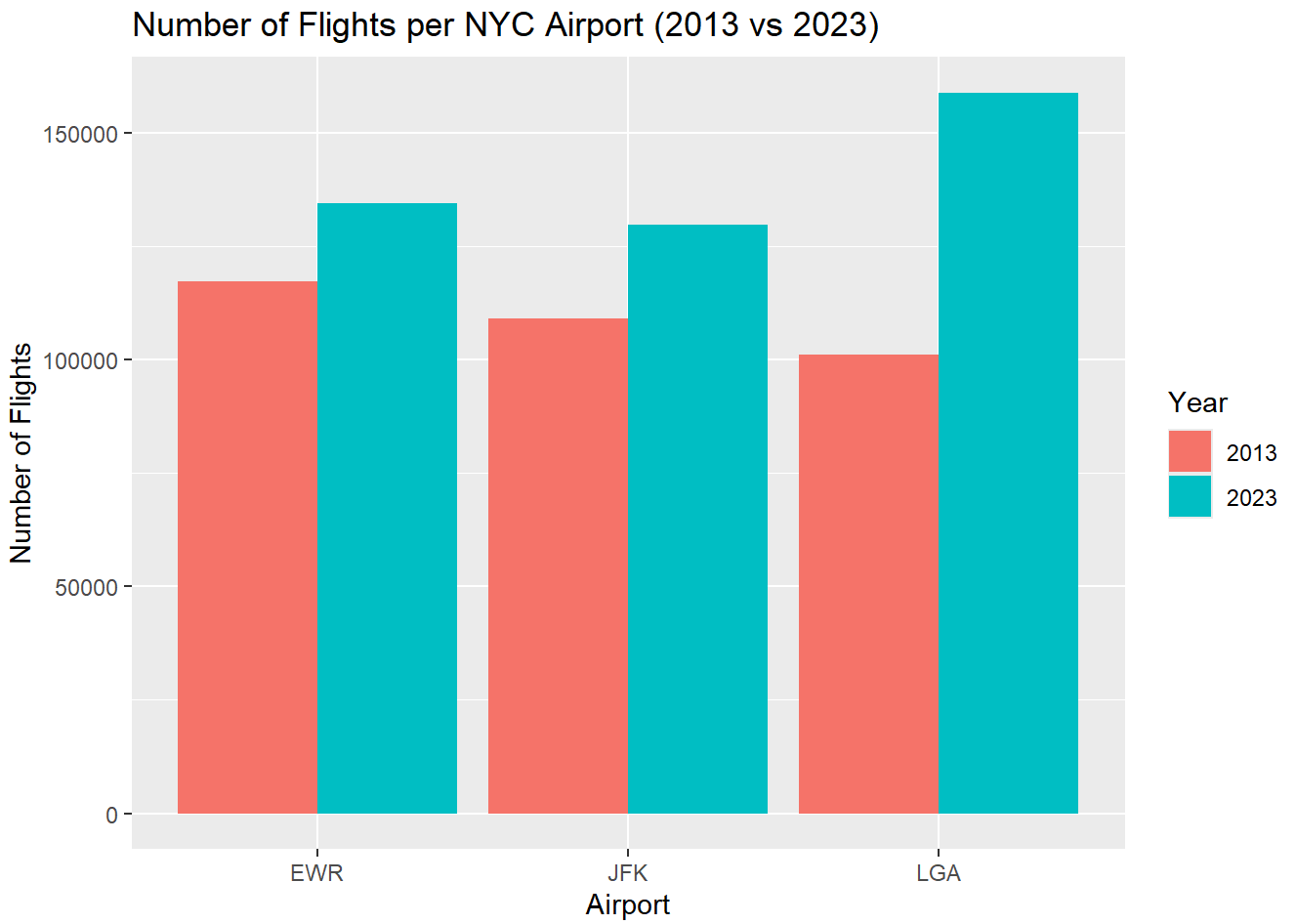
```
arr_delay_summary <- flights_clean %>%
  group_by(year) %>%
  summarise(
    avg_arr_delay = mean(arr_delay, na.rm = TRUE),
    median_arr_delay = median(arr_delay, na.rm = TRUE),
    perc_arrivals_delayed = mean(arr_delay > 15) * 100
  )

print(arr_delay_summary)
```

```
## # A tibble: 2 × 4
##    year avg_arr_delay median_arr_delay perc_arrivals_delayed
##   <dbl>         <dbl>            <dbl>                 <dbl>
## 1  2013          6.90               -5                  23.7
## 2  2023          4.34              -10                  20.7
```

- This summary focuses on arrival delays, calculating similar statistics to departure delays. It helps us check whether late departures also result in late arrivals and if arrival performance changed between 2013 and 2023.

3. Number of departures per airport

```
airport_traffic <- flights_clean %>%
  group_by(origin, year) %>%
  summarise(num_flights = n(), .groups = 'drop') #counts the number of  outgoing flights per air
port per year

ggplot(airport_traffic, aes(x = origin, y = num_flights, fill = factor(year))) +
  geom_col(position = "dodge") +
  labs(title = "Number of Flights per NYC Airport (2013 vs 2023)", x = "Airport", y = "Number of
Flights", fill = "Year")
```

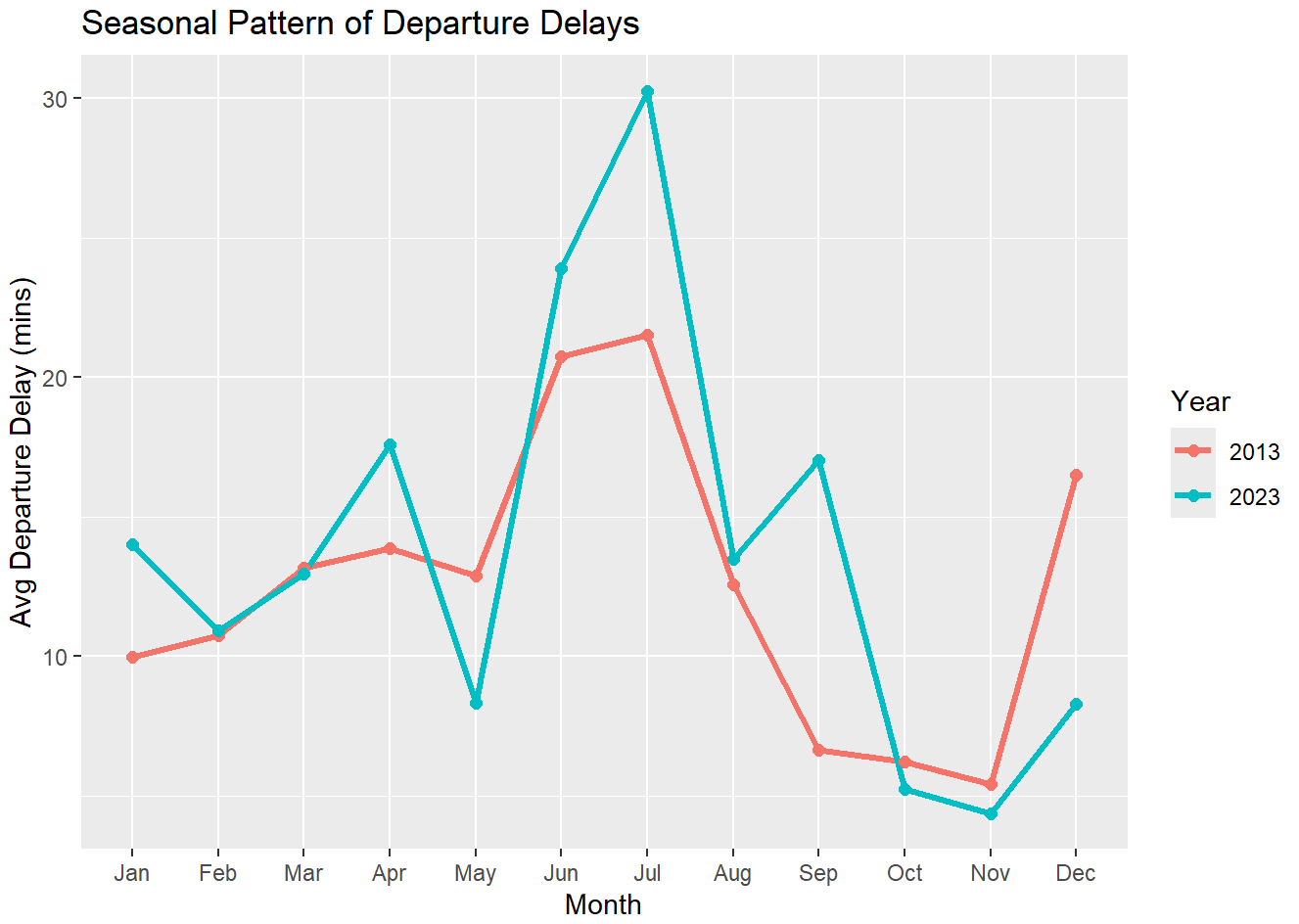


- This plot compares the total number of flights at each major NYC airport (JFK, LGA, EWR) between 2013 and 2023. It highlights if traffic increased or decreased at certain airports, helping answer which airports saw the biggest changes.

4. Seasonality of Delays

```
flights_clean %>%
  mutate(month = month(time_hour, label = TRUE)) %>% #gets the month from time_hour
  group_by(month, year) %>%
  summarise(avg_dep_delay = mean(dep_delay), .groups = 'drop') %>% #computes the average departu
re delay for that month
  ggplot(aes(x = month, y = avg_dep_delay, group = year, color = factor(year))) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Seasonal Pattern of Departure Delays", x = "Month", y = "Avg Departure Delay (mi
ns)", color = "Year")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



- This line chart shows how departure delays vary across months for both years. Peaks in certain months could point to holiday seasons, weather events, or seasonal congestion affecting flight performance.

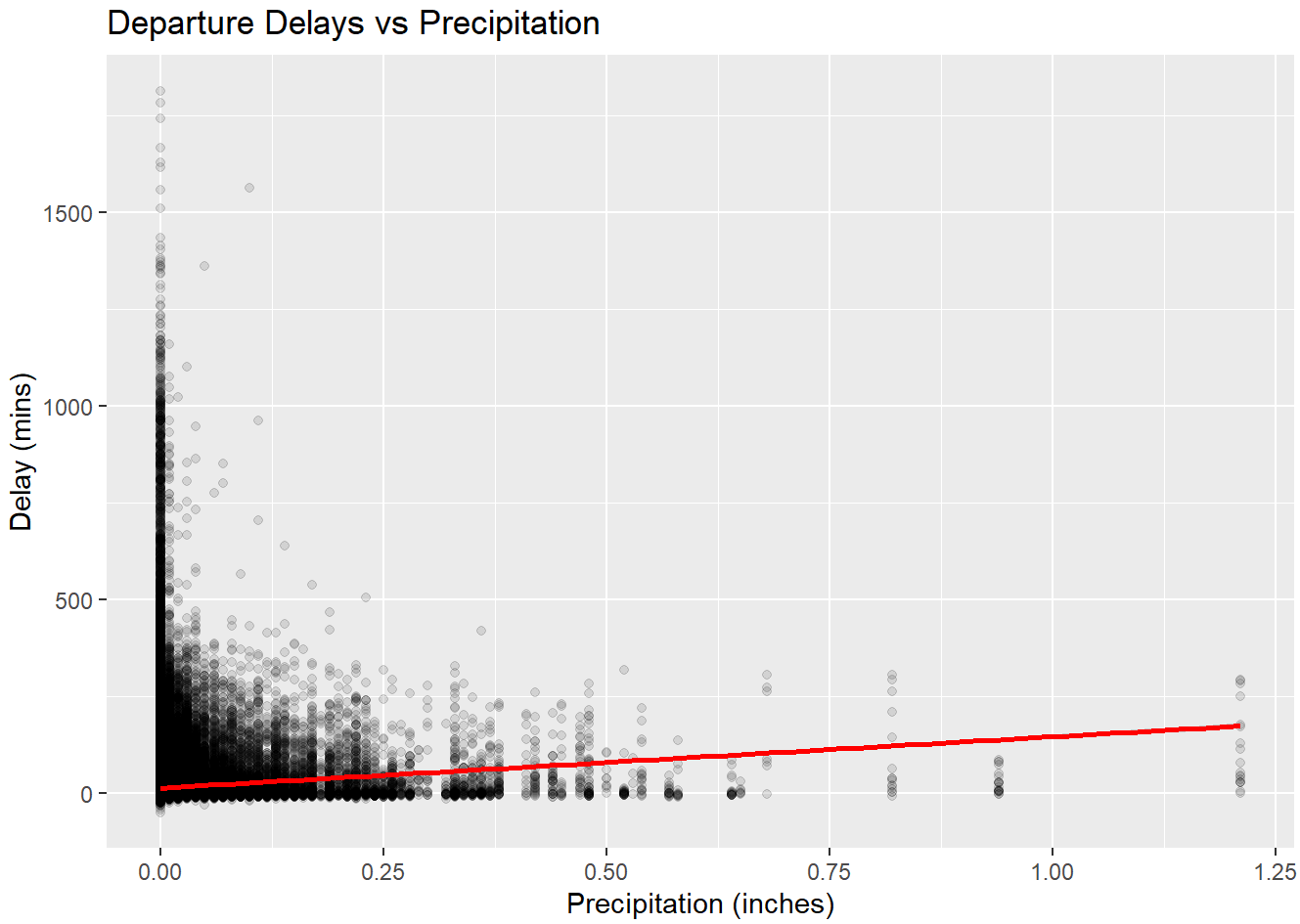
5. Weather Impact Check

```
flights_weather <- bind_rows((nycflights13::flights %>% left_join(nycflights13::weather, by = c
("origin", "time_hour"))), (nycflights23::flights %>% left_join(nycflights23::weather, by = c("o
rigin", "time_hour")))) #joins the precipitation data for 2013 and 2023

flights_weather %>%
  mutate(high_precip = precip > 0.1) %>%
  group_by(high_precip) %>%
  summarise(avg_dep_delay = mean(dep_delay, na.rm = TRUE)) %>%
  print()
```

```
## # A tibble: 3 × 2
##   high_precip avg_dep_delay
##   <lgl>               <dbl>
## 1 FALSE                13.2
## 2 TRUE                 41.6
## 3 NA                   10.1
```

```
flights_weather %>%
  filter(!is.na(dep_delay), !is.na(precip)) %>% #removes missing precipitation values
  ggplot(aes(x = precip, y = dep_delay)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Departure Delays vs Precipitation", x = "Precipitation (inches)", y = "Delay (mi
ns)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

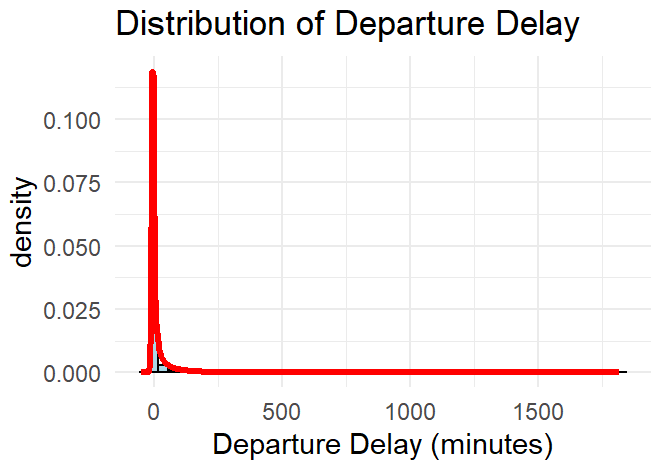


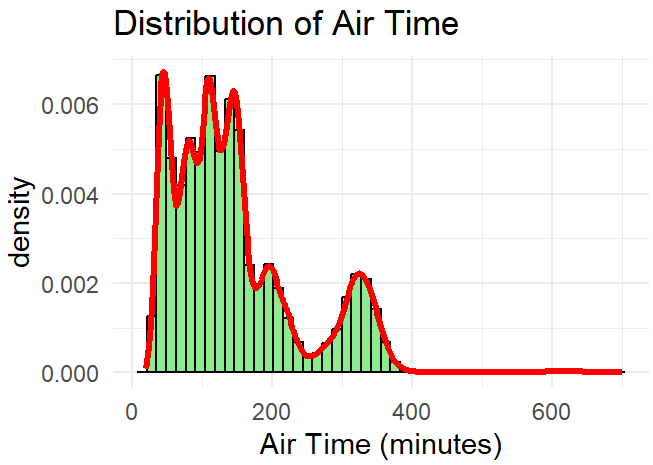
- Here, we compare the average departure delay between flights with high precipitation (rain, snow, etc.) and those with little to no precipitation. It gives quick insight into how weather affects flight delays.

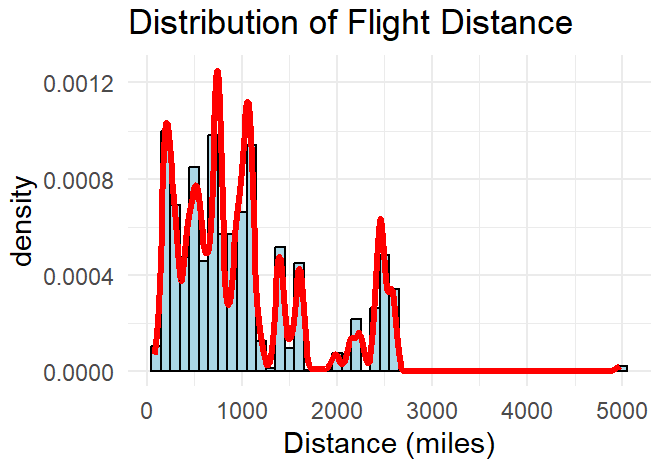
6. Distribution Checks

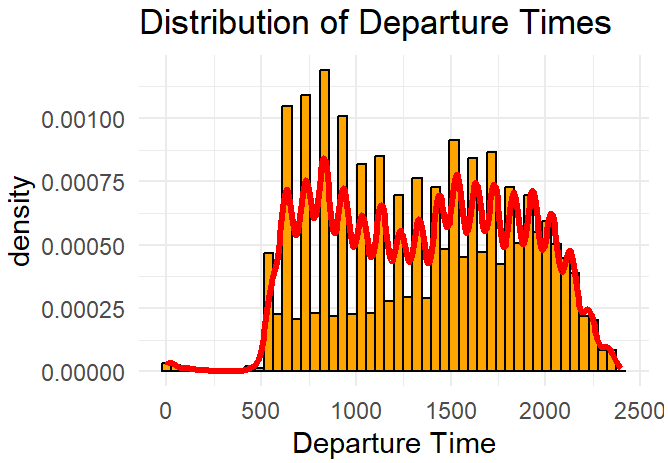
```
# Departure Delay Histogram
plot1 <- ggplot(flights_clean, aes(x = dep_delay)) +
          geom_histogram(aes(y = ..density..), bins = 50, fill = "lightblue", color = "black") +
          geom_density(color = "red", size = 1.2) +
          labs(title = "Distribution of Departure Delay", x = "Departure Delay (minutes)") +
          theme_minimal()

# Air Time Histogram
plot2 <- ggplot(flights_clean, aes(x = air_time)) +
          geom_histogram(aes(y = ..density..), bins = 50, fill = "lightgreen", color = "black")
+
          geom_density(color = "red", size = 1.2) +
          labs(title = "Distribution of Air Time", x = "Air Time (minutes)") +
          theme_minimal()

# Distance Histogram
plot3 <- ggplot(flights_clean, aes(x = distance)) +
          geom_histogram(aes(y = ..density..), bins = 50, fill = "lightblue", color = "black") +
          geom_density(color = "red", size = 1.2) +
          labs(title = "Distribution of Flight Distance", x = "Distance (miles)") +
          theme_minimal()

# Departure Time Histogram
plot4 <- ggplot(flights_clean, aes(x = dep_time)) +
          geom_histogram(aes(y = ..density..), bins = 50, fill = "orange", color = "black") +
          geom_density(color = "red", size = 1.2) +
          labs(title = "Distribution of Departure Times", x = "Departure Time") +
          theme_minimal()

grid.arrange(plot1, plot2, plot3, plot4, ncol=2)
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

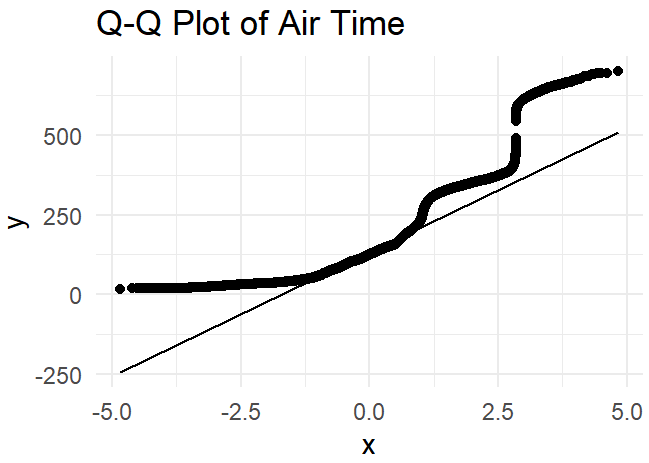


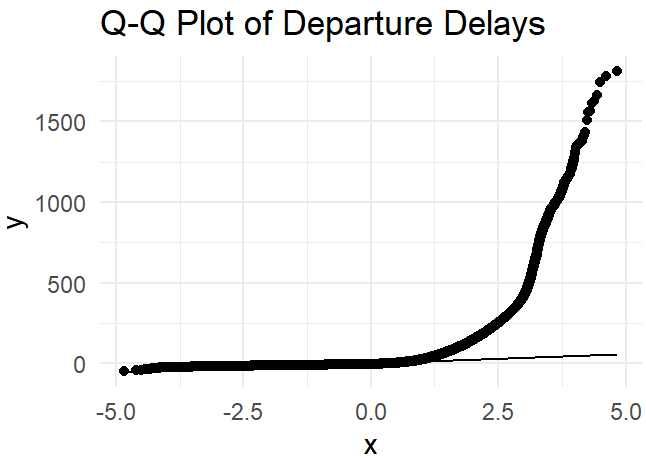


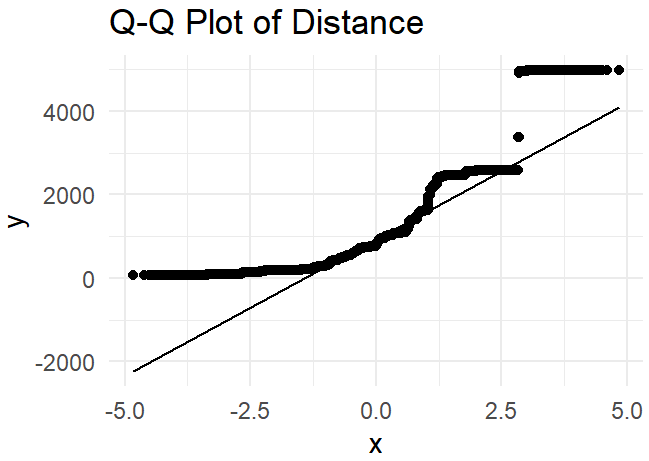


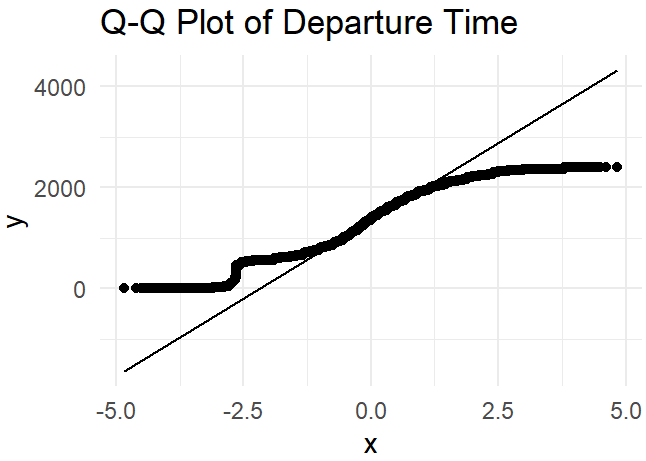



```
p1 <- ggplot(flights_clean, aes(sample = air_time)) +
        stat_qq() +
        stat_qq_line() +
        labs(title = "Q-Q Plot of Air Time") +
        theme_minimal()

p2 <- ggplot(flights_clean, aes(sample = dep_delay)) +
        stat_qq() +
        stat_qq_line() +
        labs(title = "Q-Q Plot of Departure Delays") +
        theme_minimal()

p3 <- ggplot(flights_clean, aes(sample = distance)) +
        stat_qq() +
        stat_qq_line() +
        labs(title = "Q-Q Plot of Distance") +
        theme_minimal()

p4 <- ggplot(flights_clean, aes(sample = dep_time)) +
        stat_qq() +
        stat_qq_line() +
        labs(title = "Q-Q Plot of Departure Time") +
        theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol=2)
```









- The 2 previous visualizations are meant to look at the distributions of distance, departure times, air time, departure delays, and whether or not they follow normality assumptions. If we end up applying a linear regression model, we needed to look at whether the features follow said assumptions, and in our instance they don't. The histograms enforce this due to their either highly skewed distributions or their non-normal distribution. The QQ plot further reinforces this since most QQ lines don't fit on the data properly, which shows that we may need to apply transformations to enforce the normality assumption.

7. Most Reliable Airlines (lowest average departure delay)

```
reliable_airlines <- flights_clean %>%
  group_by(name.x, year) %>%
  summarise(avg_dep_delay = mean(dep_delay, na.rm = TRUE), .groups = 'drop') %>% #calculates the average delay per airline per year
  arrange(avg_dep_delay)

print(reliable_airlines %>% group_by(year) %>% slice_head(n = 5))
```

```
## # A tibble: 10 × 3
## # Groups:   year [2]
##    name.x                 year avg_dep_delay
##    <chr>                 <dbl>         <dbl>
##  1 US Airways Inc.        2013          3.74
##  2 Hawaiian Airlines Inc. 2013          4.90
##  3 Alaska Airlines Inc.   2013          5.83
##  4 American Airlines Inc. 2013          8.57
##  5 Delta Air Lines Inc.   2013          9.22
##  6 <NA>                   2023          6.16
##  7 Endeavor Air Inc.      2023          7.38
##  8 Envoy Air              2023         10.5
##  9 Alaska Airlines Inc.   2023         11.8
## 10 American Airlines Inc. 2023         14.0
```

- This table lists the top 5 airlines with the lowest average departure delays for each year. It helps identify the most reliable carriers in 2013 and 2023, and see if rankings shifted over the decade.

Our Plan:

To answer our research questions, we will first clean, merge, and standardize the 'nycflights13' and 'nycflights23' datasets, ensuring that important variables such as delay times, weather conditions, airline names, and airport locations are consistent across both years. We assume that any missing or inconsistent data can be reasonably filtered out and that the recorded information accurately reflects real-world flight operations.

Throughout the project, we plan to use ggplot2 extensively to generate our visualizations. We will create bar plots comparing average delays across airlines and airports, time series plots showing delay trends over months and seasons, scatter plots with trend lines to examine how weather impacts departure delays, and heatmaps to visualize delays across different times of day and days of the week. For modeling, we will use linear regression to explore how factors like precipitation, wind, and scheduled time influence flight delays. We will evaluate our models using R^2 values for linear regression, matrices for logistic regression, and diagnostic plots to check for model fit and assumptions.

Through this combination of exploratory plots, statistical modeling, and comparative analysis, we aim to gain insight into how NYC flight performance has evolved over the last decade and what factors have influenced those changes.

Alternative Strategies/Backup Plans:

In the case our initial objectives don't work out as planned, here are a couple of alternative questions we came up with:

1. Are there certain aircrafts or tail numbers that seem to be more prone to delays?
    - We can analyze the delay times reported for each flight and see if there is a noticable pattern between the delay times and the types of aircrafts or tail numbers.
2. Does the data support the idea that an aircraft with more flights per day tend to experience more delays due to the tight scheduling?
    - We would look at aircrafts that are listed for multiple flights in a single day and see if the reported delay times are significant enough to prove that there is a reasonable relation between number of flights per day and flight delays.