Final Project STAT167

Shreya Mohan, Kalyani Mantirraju, Crystal Arevalo,
Karen Alvarez, Mason Lam Group: Statistically Speaking

06/02/2025

Libraries

```
#  install.packages("dunn.test")
#  install.packages("multcomp")
#  install.packages("nortest")
#  install.packages("rstatix")
#  install.packages("mgcv")
#  install.packages("gridExtra")
library(mgcv)
library(rstatix)
library(nycflights13)
library(tidyverse)
library(car)
library(dunn.test)
library(gridExtra)
library(tidyr)
library(broom)
library(multcomp)
library(nortest)
library(boot)
library(knitr)
library(gridExtra)
library(ggcorrplot)
library(MASS)
library(cowplot)
library(knitr)
```

Project Description:

The primary goal of this research is to explore factors influencing flight delays from New City airports in 2013.

Problem Statement and Motivation

Understanding factors that contribute to flight delays is critical for informing Federal Aviation Administration (FAA) policies and guiding airlines and airports in improving operational efficiency, enhancing weather

preparedness, and reducing delays through controllable factors. By analyzing weather conditions, airline differences, holiday effects, fleet age, and airport specific challenges, this research can provide data-driven insights to optimize air travel and ensure compliance with aviation regulations in heavily congested areas like New York City.

Research Questions

1. How do weather conditions affect flight delays?
2. How do differences between airlines influence flight delays?
3. Are delays more frequent during major holidays?
4. Does the age of the plane affect flight delays?
5. How do environmental factors like humidity, visibility, and wind affect flight delays?

View our GitHub Repository

Datasets

1. Flights dataset: All flights that departed from NYC in 2013

Variables:

- flights (year, month, day, dep_time, arr_time, sched_dep_time, sched_arr_time, dep_delay, arr_delay, carrier, origin, dest, air_time, distance, time_hour)
  - year, month, day : date of departure
  - dep_time, arr_time : actual departure and arrival times in HHMM
  - sched_dep_time, sched_arr_time : scheduled departure and arrival times in HHMM
  - dep_delay, arr_delay : departure and arrival delays in minutes
  - carrier : two letter carrier abbreviation of the carrier
  - origin, dest : origin and destination
  - air_time : amount of time spent in air in minutes
  - distance : distance between airport in miles
  - time_hour : scheduled date and hour of the flight as POSIXct date

2. Airlines dataset: Translation between two letter carrier codes and names

Variables:

- airlines (carrier, name)
  - carrier : two-letter abbreviation of the airlines
  - name : full name of the airlines

3. Airports dataset: Airport names and locations

Variables:

- airports (faa, name, lat, lon)
  - faa : FAA airport code
  - name : usual name of the airport
  - lat, lon : location of airport

4. Planes dataset: Construction information about each plane

Variables:

- planes (year, type, manufacturer, model, engines, seats, speed, engine)
  - year : year manufactured
  - type : type of plane
  - manufacturer, model : manufacturer and model
  - engines, seats : number of engines and seats
  - speed : average cruising speed in mph
  - engine : type in engine

5. Weather dataset: Hourly meterological data for each airport

Variables:

- weather (origin, year, month, day, hour, temp, dewp, humid, wind_dir, wind_speed, wind_gust, precip, pressure, visib, time_hour)
  - origin : weather station
  - year, month, day, hour : time of recording
  - temp, dewp : temperature and dew point in Fahrenheit
  - humid : relative humidity
  - wind_dir, wind_speed, wind_gust : wind direction in degrees, wind speed and gust in mph
  - precip : precipitation in inches
  - pressure : sea level pressure in millibars
  - visib : visibility in miles
  - time_hour : date and hour of the recording as POSIXct date

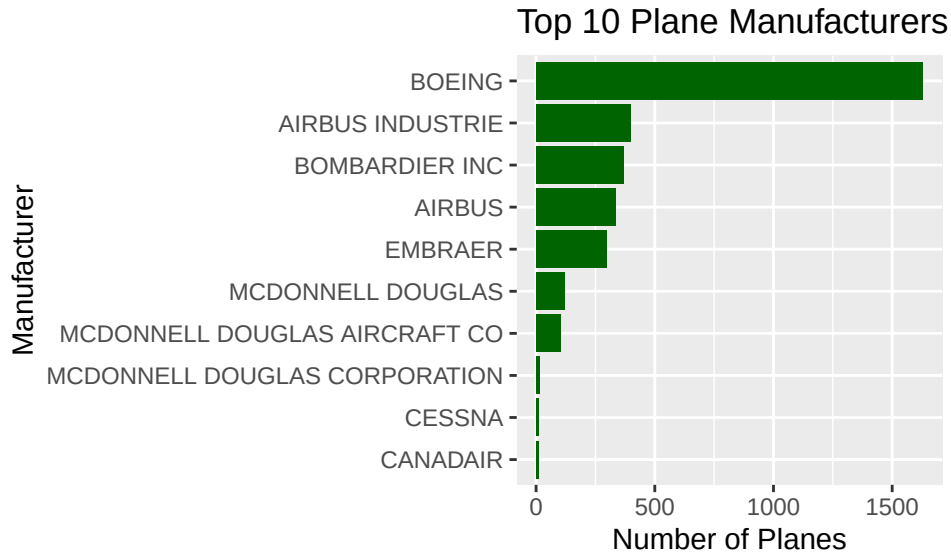
Exploratory Data Analysis

Planes Dataset EDA

```
##
## Missing values per column:

##      tailnum         year         type manufacturer        model      engines
##            0           70            0            0            0            0
##        seats        speed       engine
##            0         3299            0

##
## Column types and structure:

## Rows: 3,322
## Columns: 9
## $ tailnum      <chr> "N10156", "N102UW", "N103US", "N104UW", "N10575", "N105UW~
## $ year         <int> 2004, 1998, 1999, 1999, 2002, 1999, 1999, 1999, 1999, 199~
## $ type         <chr> "Fixed wing multi engine", "Fixed wing multi engine", "Fi~
## $ manufacturer <chr> "EMBRAER", "AIRBUS INDUSTRIE", "AIRBUS INDUSTRIE", "AIRBU~
## $ model        <chr> "EMB-145XR", "A320-214", "A320-214", "A320-214", "EMB-145~
## $ engines      <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ seats        <int> 55, 182, 182, 182, 55, 182, 182, 182, 182, 182, 55, 55, 5~
## $ speed        <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ engine       <chr> "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turb~
## # A tibble: 3,322 x 9
##    tailnum  year type              manufacturer model engines seats speed engine
##    <chr>   <int> <chr>             <chr>        <chr>   <int> <int> <int> <chr>
##  1 N10156   2004 Fixed wing multi~ EMBRAER      EMB-~       2    55    NA Turbo~
##  2 N102UW   1998 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  3 N103US   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  4 N104UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  5 N10575   2002 Fixed wing multi~ EMBRAER      EMB-~       2    55    NA Turbo~
##  6 N105UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  7 N107US   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  8 N108UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  9 N109UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## 10 N110UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## # i 3,312 more rows

##
## First few rows:

## # A tibble: 6 x 9
##   tailnum  year type              manufacturer model engines seats speed engine
##   <chr>   <int> <chr>             <chr>        <chr>   <int> <int> <int> <chr>
## 1 N10156   2004 Fixed wing multi ~ EMBRAER      EMB-~       2    55    NA Turbo~
## 2 N102UW   1998 Fixed wing multi ~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## 3 N103US   1999 Fixed wing multi ~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## 4 N104UW   1999 Fixed wing multi ~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## 5 N10575   2002 Fixed wing multi ~ EMBRAER      EMB-~       2    55    NA Turbo~
## 6 N105UW   1999 Fixed wing multi ~ AIRBUS INDU~ A320~       2   182    NA Turbo~
```

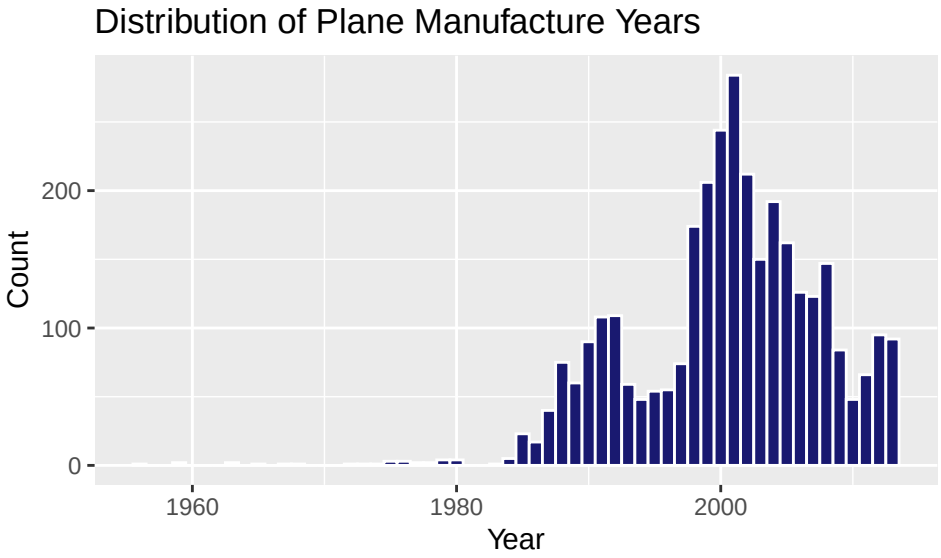
```r
planes %>%
  count(manufacturer, sort = TRUE) %>%
  top_n(10) %>%
  ggplot(aes(x = reorder(manufacturer, n), y = n)) +
  geom_col(fill = "darkgreen") +
  coord_flip() +
  labs(title = "Top 10 Plane Manufacturers", x = "Manufacturer", y = "Number of Planes")
```

```
## Selecting by n
```



The visualization above shows the top 10 plane manufacturers present in the data-set. Boeing has the largest amount of planes with approximately 1750 planes, and Airbus has the second most with approximately 400 planes.

```r
ggplot(planes, aes(x = year)) +
  geom_histogram(binwidth = 1, fill = "midnightblue", color = "white") +
  labs(title = "Distribution of Plane Manufacture Years", x = "Year", y = "Count")
```



This histogram shows the distribution of plane manufacture years, with the majority of planes built between the mid-1990s and early 2000s. There is a notable peak around the year 2000, indicating a surge in plane production during that period.

Airlines Dataset EDA

Dimensions and column names of the airlines dataset

```
##
## Missing values per column:

## carrier    name
##       0       0

##
## Column types and structure:

## Rows: 16
## Columns: 2
## $ carrier <chr> "9E", "AA", "AS", "B6", "DL", "EV", "F9", "FL", "HA", "MQ", "O~
## $ name    <chr> "Endeavor Air Inc.", "American Airlines Inc.", "Alaska Airline~
## # A tibble: 16 x 2
##    carrier name
##    <chr>   <chr>
##  1 9E      Endeavor Air Inc.
##  2 AA      American Airlines Inc.
##  3 AS      Alaska Airlines Inc.
##  4 B6      JetBlue Airways
##  5 DL      Delta Air Lines Inc.
##  6 EV      ExpressJet Airlines Inc.
##  7 F9      Frontier Airlines Inc.
##  8 FL      AirTran Airways Corporation
##  9 HA      Hawaiian Airlines Inc.
```

```
## 10 MQ      Envoy Air
## 11 OO      SkyWest Airlines Inc.
## 12 UA      United Air Lines Inc.
## 13 US      US Airways Inc.
## 14 VX      Virgin America
## 15 WN      Southwest Airlines Co.
## 16 YV      Mesa Airlines Inc.

##
## First few rows:

## # A tibble: 6 x 2
##   carrier name
##   <chr>   <chr>
## 1 9E      Endeavor Air Inc.
## 2 AA      American Airlines Inc.
## 3 AS      Alaska Airlines Inc.
## 4 B6      JetBlue Airways
## 5 DL      Delta Air Lines Inc.
## 6 EV      ExpressJet Airlines Inc.
```

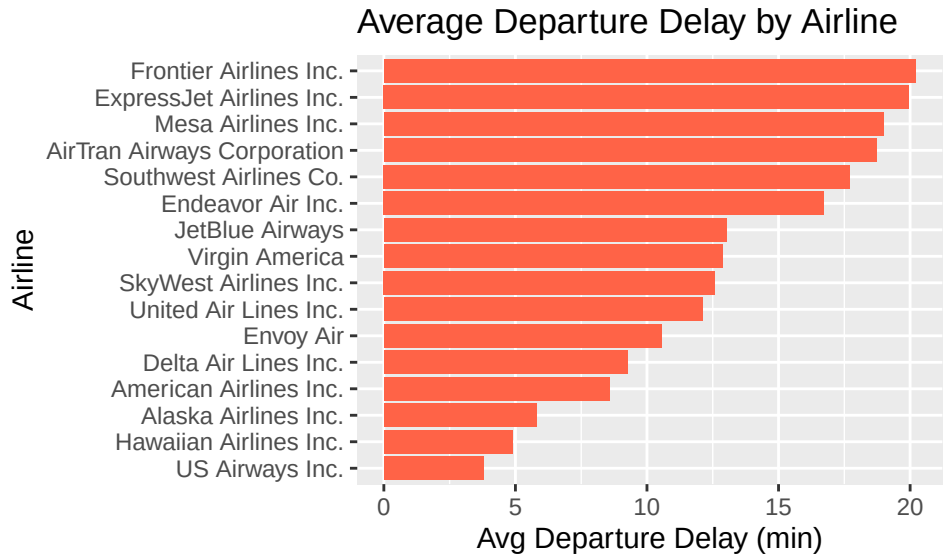
Viewing all the Unique Airlines:

```
airlines %>%
  arrange(name)
```

```
## # A tibble: 16 x 2
##    carrier name
##    <chr>   <chr>
##  1 FL      AirTran Airways Corporation
##  2 AS      Alaska Airlines Inc.
##  3 AA      American Airlines Inc.
##  4 DL      Delta Air Lines Inc.
##  5 9E      Endeavor Air Inc.
##  6 MQ      Envoy Air
##  7 EV      ExpressJet Airlines Inc.
##  8 F9      Frontier Airlines Inc.
##  9 HA      Hawaiian Airlines Inc.
## 10 B6      JetBlue Airways
## 11 YV      Mesa Airlines Inc.
## 12 OO      SkyWest Airlines Inc.
## 13 WN      Southwest Airlines Co.
## 14 US      US Airways Inc.
## 15 UA      United Air Lines Inc.
## 16 VX      Virgin America
```

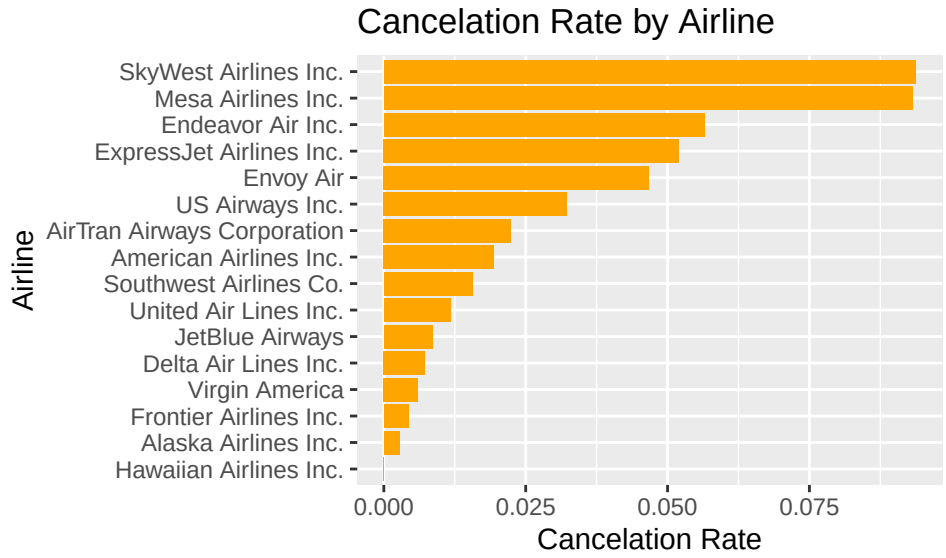
```
# Join flights and airline names
flights_airlines <- flights %>%
  left_join(airlines, by = "carrier")

# Average delay metrics
avg_delays <- flights_airlines %>%
```

```
  group_by(name) %>%
  summarise(
    avg_dep_delay = mean(dep_delay, na.rm = TRUE),
    avg_arr_delay = mean(arr_delay, na.rm = TRUE),
    flights = n()
  )

# Plot: Departure Delay
ggplot(avg_delays, aes(x = reorder(name, avg_dep_delay), y = avg_dep_delay)) +
  geom_col(fill = "tomato") +
  coord_flip() +
  labs(
    title = "Average Departure Delay by Airline",
    x = "Airline",
    y = "Avg Departure Delay (min)"
  )
```



We can see that on average, Frontier Airlines has the most departure delay at around 20 min, with ExpressJet roughly around the same 20 minutes. Less than half the Airlines seem to be past the 13 minute delay mark.

```
cancel_rate <- flights_airlines %>%
  mutate(cancelled = is.na(dep_delay)) %>%
  group_by(name) %>%
  summarise(cancel_rate = mean(cancelled), total_flights = n())

ggplot(cancel_rate, aes(x = reorder(name, cancel_rate), y = cancel_rate)) +
  geom_col(fill = "orange") +
  coord_flip() +
  labs(
    title = "Cancelation Rate by Airline",
    x = "Airline",
    y = "Cancelation Rate"
  )
```



As we can see from above, Skywest Airlines Inc has the highest cancellation rate, with Mesa Airlines very closely behind, and a huge drop off at Endeavor Air Inc.

Flights Dataset EDA:

```
##
## Missing values per column:

##           year          month            day       dep_time sched_dep_time
##              0              0              0           8255              0
##      dep_delay       arr_time sched_arr_time      arr_delay        carrier
##           8255           8713              0           9430              0
##         flight        tailnum         origin           dest       air_time
##              0           2512              0              0           9430
##       distance           hour         minute      time_hour
##              0              0              0              0

##
## Column types and structure:

## Rows: 336,776
## Columns: 19
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2~
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 558, ~
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 600, ~
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1~
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 849,~
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 851,~
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1~
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "~
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301, 4~
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N394~
```

```
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA",~
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IAD",~
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149, 1~
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 733, ~
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 6~
## $ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0~
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0~
## # 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
## # i 336,766 more rows
## # 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>

##
## First few rows:

## # A tibble: 6 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
## # 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>
```

Most of our analysis is based on how other variables and datasets affect and compare to the flights dataset. We are seeing how the arrival time, departure delay time, departure time, arrival delay time, and other variables are affected.

flights that were not canceled - We will be using these the not_canceled data for the rest of the EDA

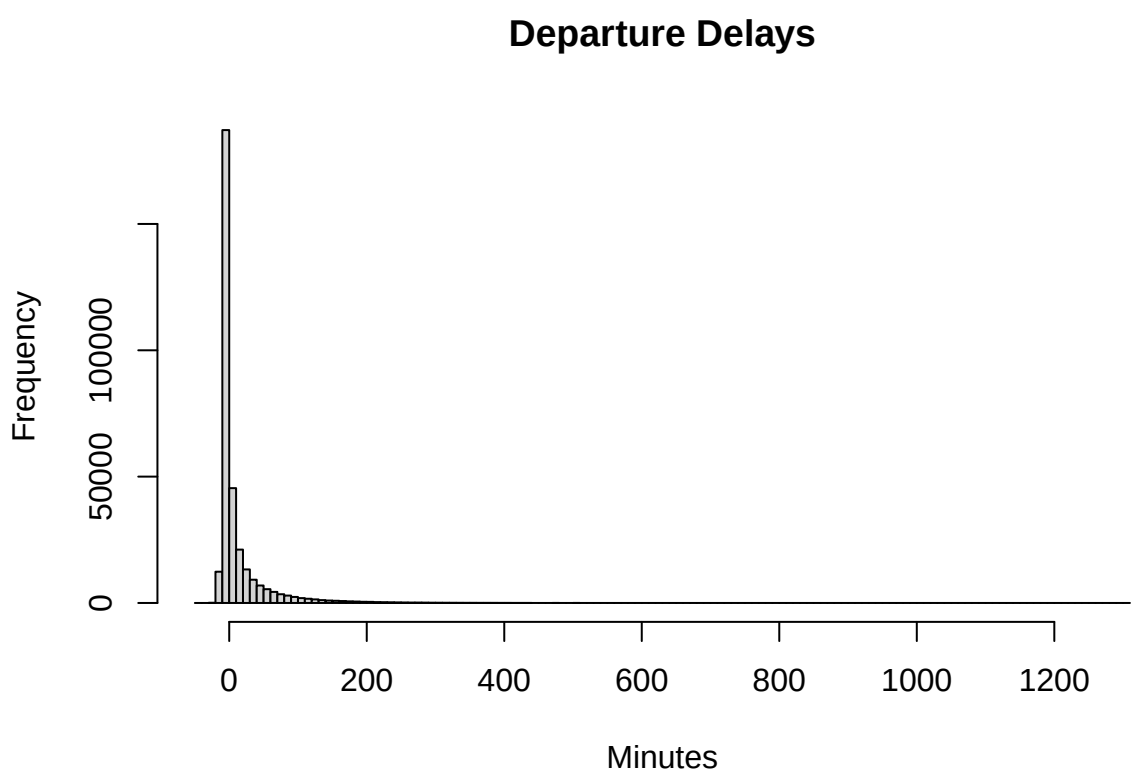
```
not_canceled <- filter(flights, !is.na(dep_delay), !is.na(arr_delay))
not_canceled
```

```
## # A tibble: 327,346 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>
```

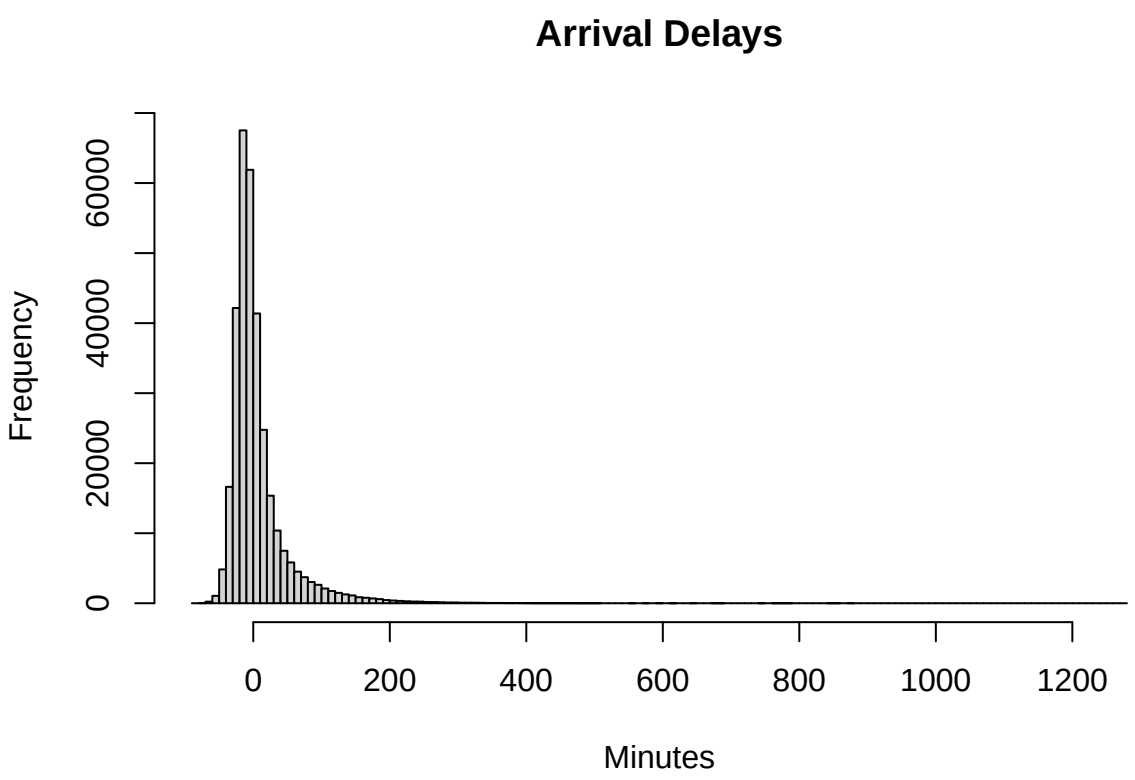
```
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -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
## # i 327,336 more rows
## # 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>
```

Basic delay analysis Distribution and Proportion of delayed flights that were not canceled

```
#histograms
hist(not_canceled$dep_delay, breaks=100, main = "Departure Delays",xlab = "Minutes")
```




```
hist(not_canceled$arr_delay, breaks=100, main ="Arrival Delays", xlab = "Minutes")
```




```
#proportions
mean(not_canceled$dep_delay>0, na.rm=TRUE)
```

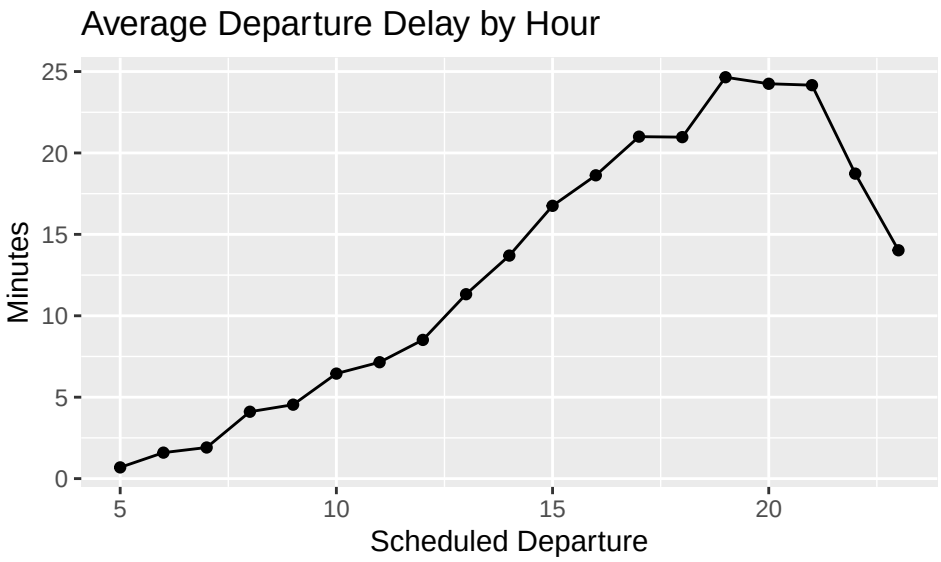
```
## [1] 0.3902446
```

```
mean(not_canceled$arr_delay>0, na.rm=TRUE)
```

```
## [1] 0.4063101
```

Most of the departure delays do not go over 200 minutes and the arrival delays have very few delays past 200 minutes.

```
#convert time to hours
not_canceled$dep_hour <-floor(not_canceled$sched_dep_time/100)
not_canceled$arr_hour <-floor(not_canceled$sched_arr_time/100)

#plot
not_canceled |>
  group_by(dep_hour) |>
  summarize(mean_dep_delay = mean(dep_delay, na.rm=TRUE))|>
  ggplot(aes(x=dep_hour, y =mean_dep_delay))+
  geom_line()+
  geom_point()+
  labs(title = "Average Departure Delay by Hour", x="Scheduled Departure", y="Minutes")
```



We can see that many of the delays happen further in the day and peak at about 18 hours and then it descends from there.

```
not_canceled |>
  group_by(origin)|>
  summarize(avg_dep_delay= mean(dep_delay, na.rm=TRUE), avg_arr_delay= mean(arr_delay, na.rm=TRUE))
```

```
## # A tibble: 3 x 3
##   origin avg_dep_delay avg_arr_delay
##   <chr>          <dbl>         <dbl>
## 1 EWR             15.0          9.11
## 2 JFK             12.0          5.55
## 3 LGA             10.3          5.78
```

EWR has the highest average departure and arrival delay followed by JFK and then LGA

```
not_canceled|>
  group_by(carrier)|>
  summarize(avg_dep_delay = mean(dep_delay, na.rm=TRUE))|>
  arrange(desc(avg_dep_delay))
```

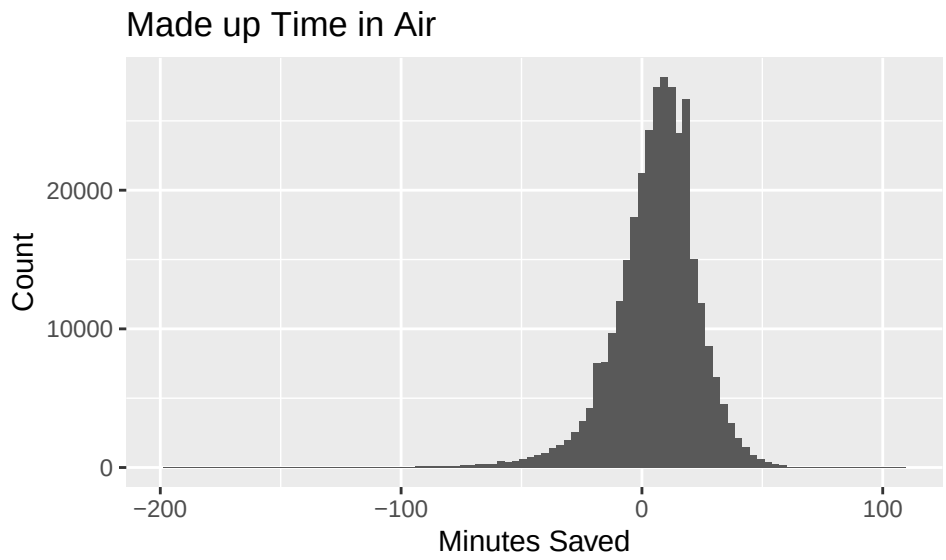
```
## # A tibble: 16 x 2
##    carrier avg_dep_delay
##    <chr>           <dbl>
##  1 F9               20.2
##  2 EV               19.8
##  3 YV               18.9
##  4 FL               18.6
##  5 WN               17.7
##  6 9E               16.4
##  7 B6               13.0
##  8 VX               12.8
##  9 OO               12.6
## 10 UA               12.0
```

```
## 11 MQ              10.4
## 12 DL               9.22
## 13 AA               8.57
## 14 AS               5.83
## 15 HA               4.90
## 16 US               3.74
```

F9 has the highest average departure delay at 20 hours.

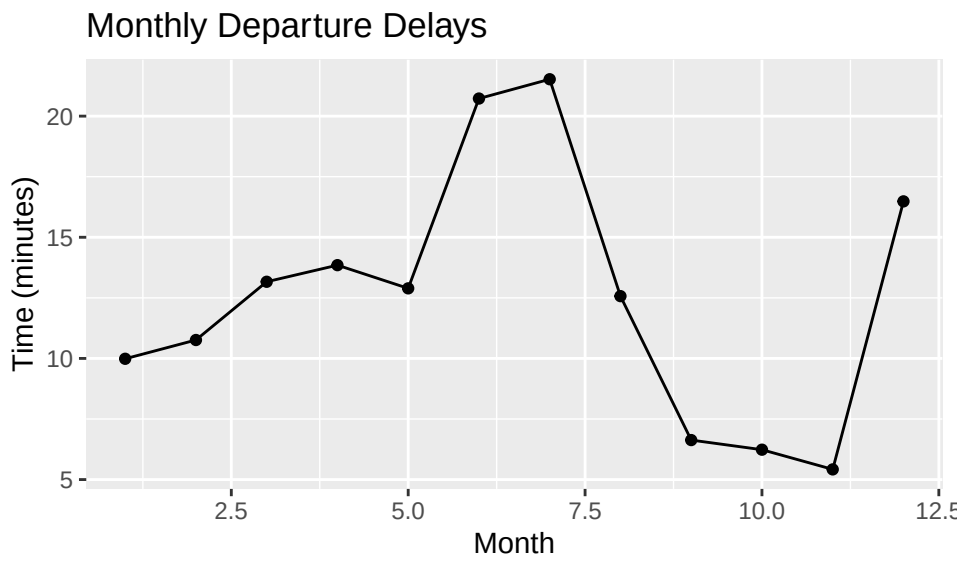
Check to see if the flights that were delayed made up the time in the air

```
not_canceled|>
  mutate(made_up_time = dep_delay- arr_delay)|>
  ggplot(aes(x=made_up_time))+
  geom_histogram(bins=100)+
  labs(title="Made up Time in Air", x= "Minutes Saved", y="Count")
```



We can see that the majority of the flights did not save any minutes on the arrival delay and actually ended up being delayed more. Some flights did in fact save minutes but it was less than 50% of all flights.

```
not_canceled |>
  group_by(month) |>
  summarize(mean_dep_delay = mean(dep_delay, na.rm = TRUE)) |>
  ggplot(aes(x = month, y = mean_dep_delay)) +
  geom_line() +
  geom_point() +
  labs(title = "Monthly Departure Delays", x = "Month", y = "Time (minutes)")
```

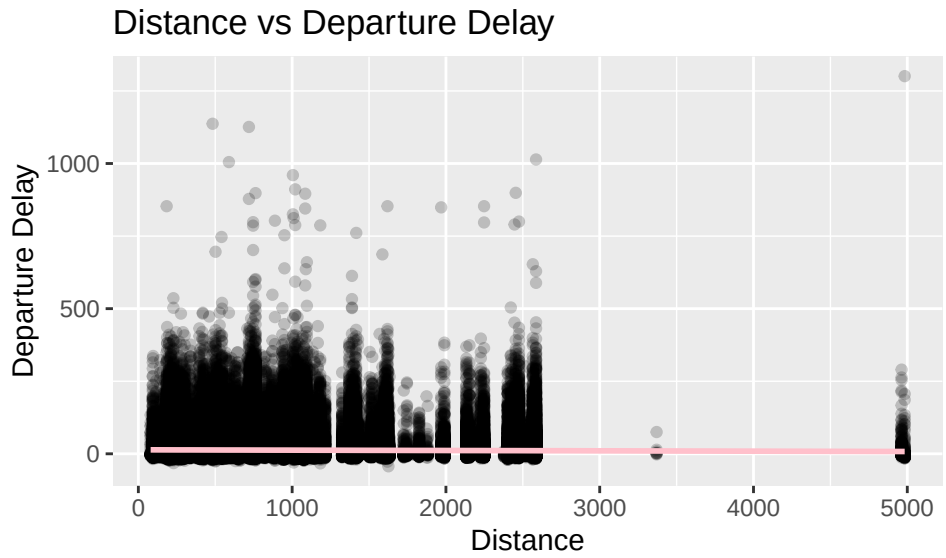


we see that the majority of flights are delayed from May to mid July, and there is another peak at December. The months with the shorest delays are September and October.

```
ggplot(not_canceled, aes(x=distance, y=dep_delay))+
  geom_point(alpha=0.2)+
  geom_smooth(method = "lm", se=TRUE,color= "Pink")+
  labs(title = "Distance vs Departure Delay", x="Distance", y="Departure Delay")
```

Does distance affect the amount of delays?

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



We can see that there is not much of an effect of Distance on Departure delay.

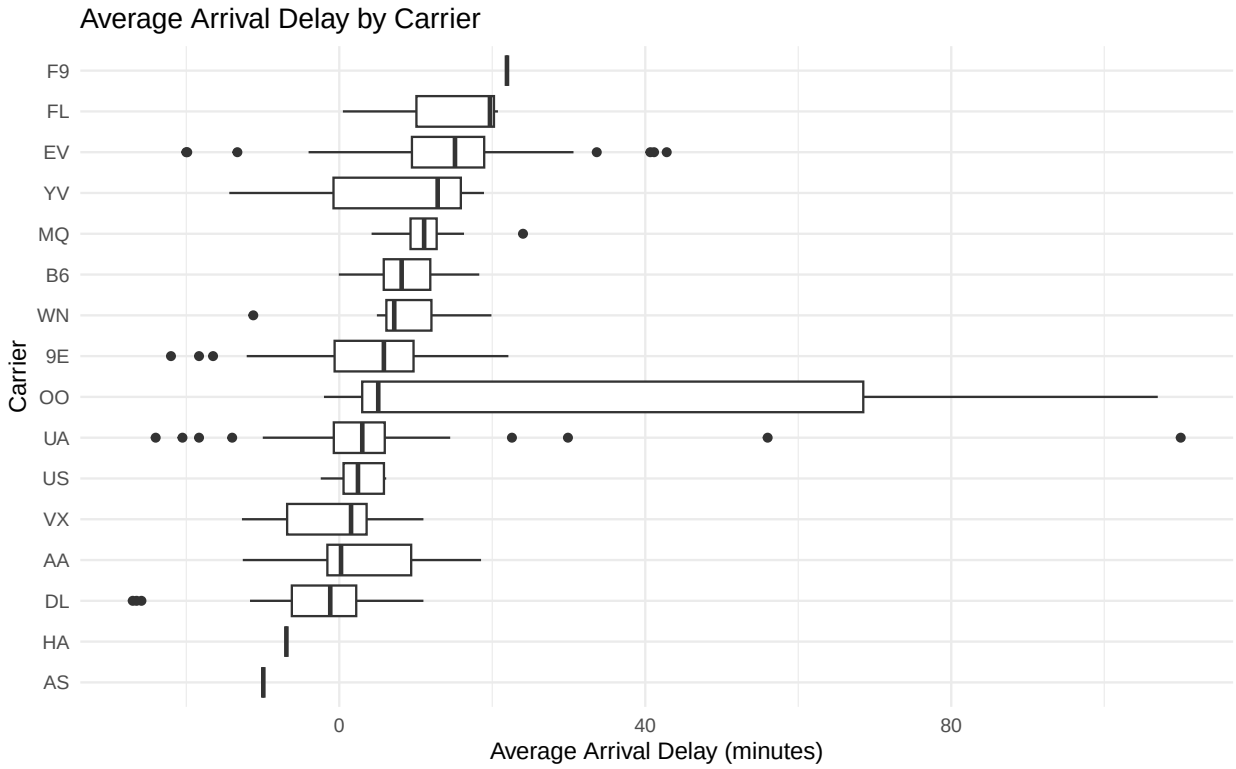
```
longest_Distance <- not_canceled |>
  arrange(desc(distance)) |>
  dplyr::select(carrier, origin, dest)
longest_Distance
```

Flights traveled the longest by distance

```
## # A tibble: 327,346 x 3
##    carrier origin dest
##    <chr>   <chr>  <chr>
##  1 HA      JFK    HNL
##  2 HA      JFK    HNL
##  3 HA      JFK    HNL
##  4 HA      JFK    HNL
##  5 HA      JFK    HNL
##  6 HA      JFK    HNL
##  7 HA      JFK    HNL
##  8 HA      JFK    HNL
##  9 HA      JFK    HNL
## 10 HA      JFK    HNL
## # i 327,336 more rows
```

We see that HA is the carrier with the longest flights and they all start at JFK airport and land at HNL.

```
carrier_Dest <- not_canceled |>
  group_by(carrier, dest) |>
  summarize(avg_arr_Delay = mean(arr_delay, na.rm = TRUE), .groups = "drop")

ggplot(carrier_Dest, aes(x = reorder(carrier, avg_arr_Delay, median), y = avg_arr_Delay)) +
  geom_boxplot() +
  coord_flip() +
  labs(title = "Average Arrival Delay by Carrier", x = "Carrier", y = "Average Arrival Delay (minutes)"
  theme_minimal()
```



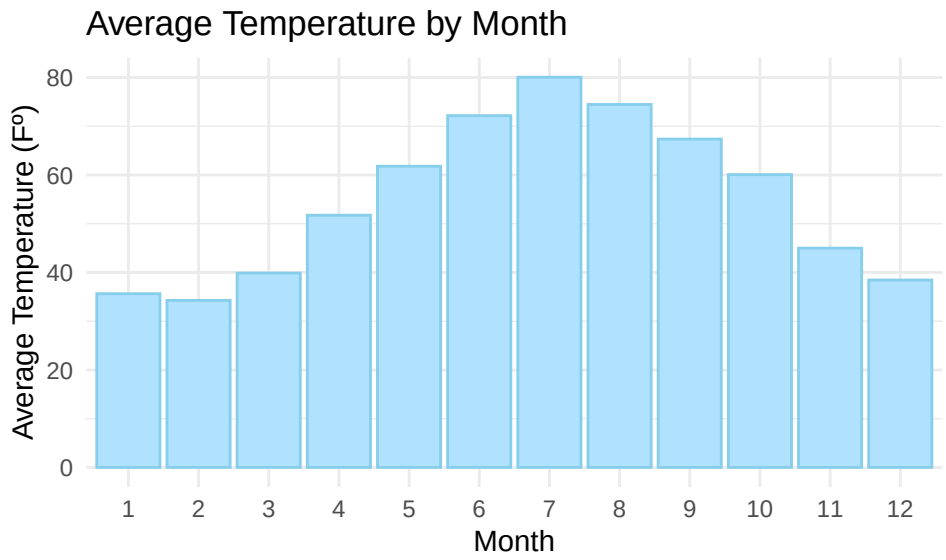
Weather Dataset EDA

```
##
## Missing values per column:

##     origin       year      month        day       hour       temp       dewp
##          0          0          0          0          0          1          1
##      humid   wind_dir wind_speed  wind_gust     precip   pressure      visib
##          1        460          4      20778          0       2729          0
##  time_hour
##          0

##
## Column types and structure:

## Rows: 26,115
## Columns: 15
## $ origin     <chr> "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EW~
## $ year       <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013,~
## $ month      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ day        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ hour       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, ~
## $ temp       <dbl> 39.02, 39.02, 39.02, 39.92, 39.02, 37.94, 39.02, 39.92, 39.~
## $ dewp       <dbl> 26.06, 26.96, 28.04, 28.04, 28.04, 28.04, 28.04, 28.04, 28.~
## $ humid      <dbl> 59.37, 61.63, 64.43, 62.21, 64.43, 67.21, 64.43, 62.21, 62.~
## $ wind_dir   <dbl> 270, 250, 240, 250, 260, 240, 240, 250, 260, 260, 260, 330,~
## $ wind_speed <dbl> 10.35702, 8.05546, 11.50780, 12.65858, 12.65858, 11.50780, ~
```

```
## $ wind_gust  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 20.~
## $ precip     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ pressure   <dbl> 1012.0, 1012.3, 1012.5, 1012.2, 1011.9, 1012.4, 1012.2, 101~
## $ visib      <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,~
## $ time_hour  <dttm> 2013-01-01 01:00:00, 2013-01-01 02:00:00, 2013-01-01 03:00~
## # A tibble: 26,115 x 15
##    origin  year month   day  hour  temp  dewp humid wind_dir wind_speed
##    <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>
##  1 EWR     2013     1     1     1  39.0  26.1  59.4      270      10.4
##  2 EWR     2013     1     1     2  39.0  27.0  61.6      250       8.06
##  3 EWR     2013     1     1     3  39.0  28.0  64.4      240      11.5
##  4 EWR     2013     1     1     4  39.9  28.0  62.2      250      12.7
##  5 EWR     2013     1     1     5  39.0  28.0  64.4      260      12.7
##  6 EWR     2013     1     1     6  37.9  28.0  67.2      240      11.5
##  7 EWR     2013     1     1     7  39.0  28.0  64.4      240      15.0
##  8 EWR     2013     1     1     8  39.9  28.0  62.2      250      10.4
##  9 EWR     2013     1     1     9  39.9  28.0  62.2      260      15.0
## 10 EWR     2013     1     1    10  41    28.0  59.6      260      13.8
## # i 26,105 more rows
## # i 5 more variables: wind_gust <dbl>, precip <dbl>, pressure <dbl>,
## #   visib <dbl>, time_hour <dttm>

##
## First few rows:

## # A tibble: 6 x 15
##   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed wind_gust
##   <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>     <dbl>
## 1 EWR     2013     1     1     1  39.0  26.1  59.4      270      10.4         NA
## 2 EWR     2013     1     1     2  39.0  27.0  61.6      250       8.06        NA
## 3 EWR     2013     1     1     3  39.0  28.0  64.4      240      11.5         NA
## 4 EWR     2013     1     1     4  39.9  28.0  62.2      250      12.7         NA
## 5 EWR     2013     1     1     5  39.0  28.0  64.4      260      12.7         NA
## 6 EWR     2013     1     1     6  37.9  28.0  67.2      240      11.5         NA
## # i 4 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,
## #   time_hour <dttm>
```

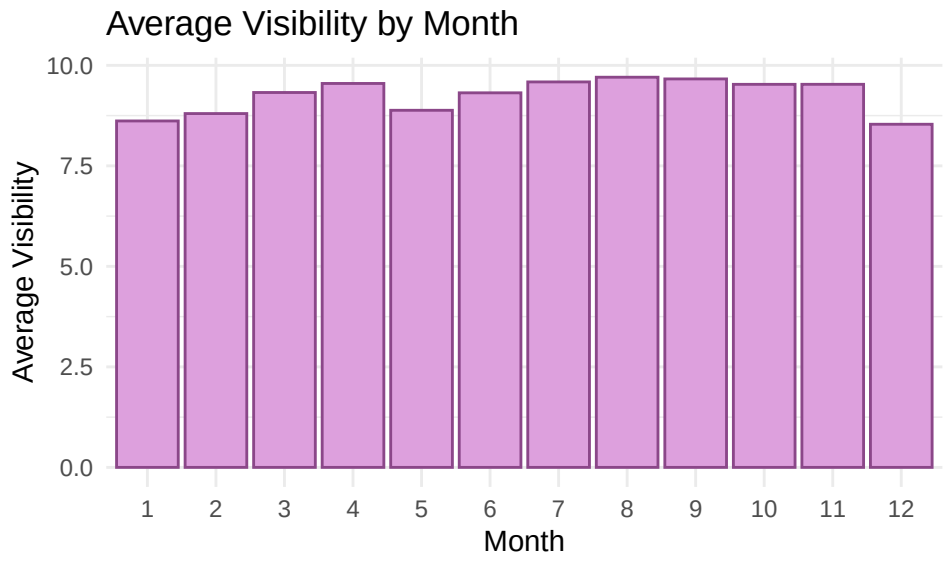
Related Questions from our Proposal: 1) How do weather conditions affect flight delays? 2) How do environmental factors like humidity, visibility, and wind affect flight delays? 3) What impact does precipitation have on specific airports and weather-related delays?

```r
# Average Temperature by Month

monthlyavgtemp <- weather %>%
  group_by(month) %>%
  summarise(monthlyavgtemp = mean(temp, na.rm = TRUE))

ggplot(data = monthlyavgtemp,
       aes(x = factor(month), y = monthlyavgtemp)) +
  geom_col(color = "skyblue", fill = "lightskyblue1") +
  labs(title = "Average Temperature by Month",
       x = "Month",
       y = "Average Temperature (F°)") +
  theme_minimal()
```

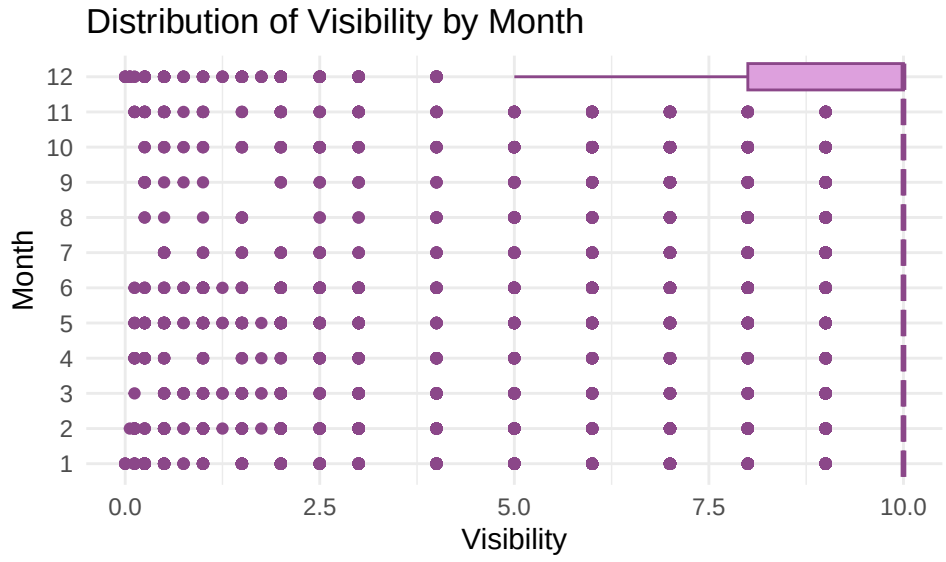


We can see that the average temperature ranges from about 70-80° in the summer, and 35-40° in the winter. When answering our research questions, we can see if there is a correlation between summer/winter weather and flight delays.

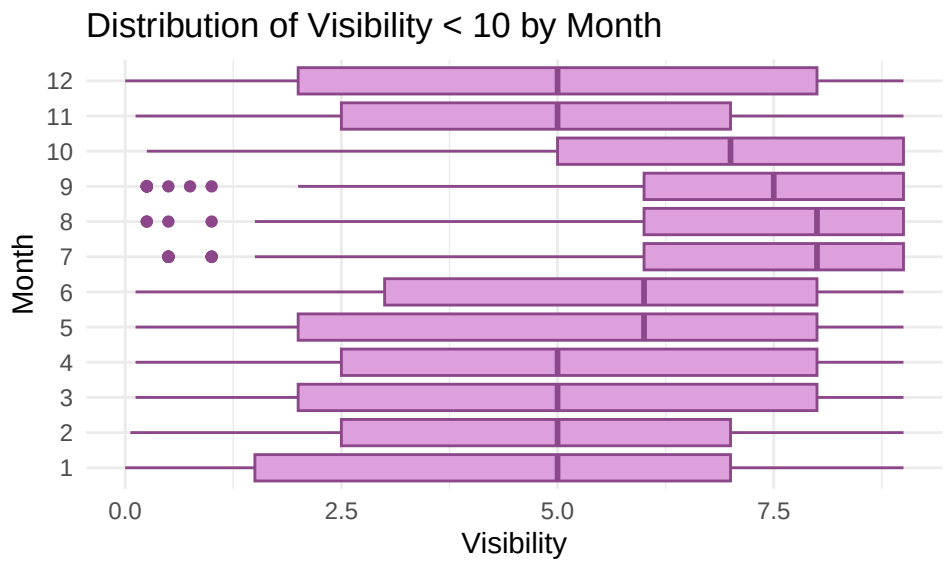
```
# Average Visibility by Month

monthlyavgvisib <- weather %>%
  group_by(month) %>%
  summarise(monthlyavgvisib = mean(visib, na.rm = TRUE))

ggplot(data = monthlyavgvisib) +
    geom_col(aes(x = factor(month), y = monthlyavgvisib),
                  color = "orchid4", fill = "plum") +
  labs(title = "Average Visibility by Month",
       x = "Month",
       y = "Average Visibility") +
  theme_minimal()
```




```
ggplot(weather, aes(x = factor(month), y = visib)) +
  geom_boxplot(fill = "plum", color = "orchid4") +
  labs(title = "Distribution of Visibility by Month",
       x = "Month",
       y = "Visibility") +
  coord_flip() +
  theme_minimal()
```

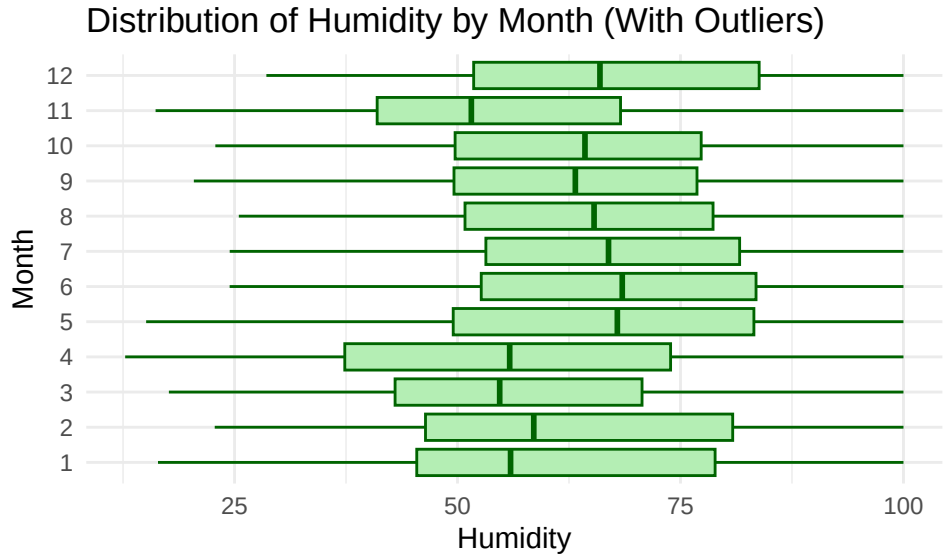



```
ggplot(filter(weather, visib < 10), aes(x = factor(month), y = visib)) +
  geom_boxplot(fill = "plum", color = "orchid4") +
  labs(title = "Distribution of Visibility < 10 by Month",
       x = "Month",
       y = "Visibility") +
  coord_flip() +
  theme_minimal()
```



The average visibility does not greatly vary by month looking at the average value. However, we can see that there is slightly less visiblity in winter months. Looking at the boxplots, we can see that there are a lot of outliers. Removing these outliers and focusing on visib < 10 shows us a better distribution of visiblity. When answering our research questions, we can compare the average visibility during flight delays vs average visibility without flight delays to further explore the role of visibility in flight delays.

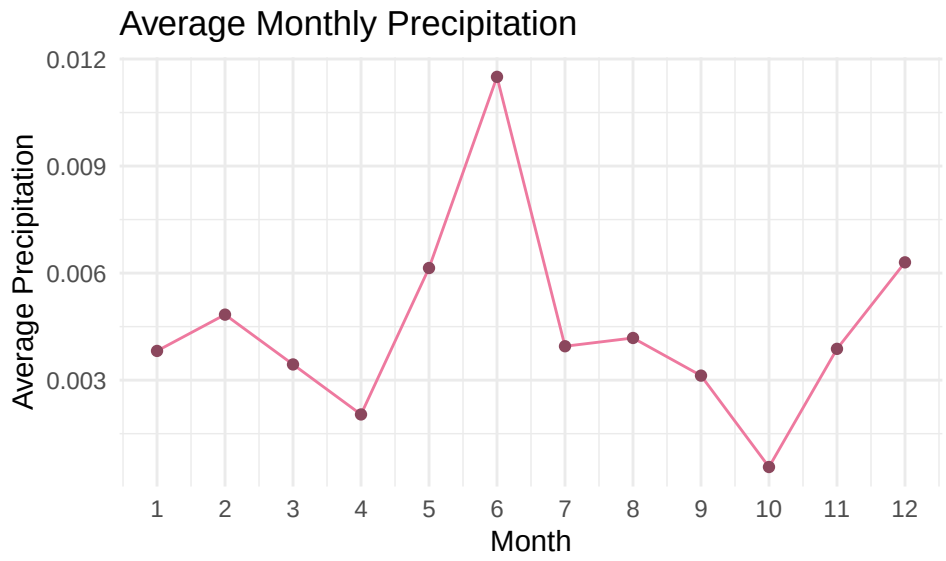
```
# Distribution of Humidity by Month

ggplot(weather, aes(x = factor(month), y = humid)) +
  geom_boxplot(fill = "darkseagreen2", color = "darkgreen") +
  labs(title = "Distribution of Humidity by Month (With Outliers)",
       x = "Month",
       y = "Humidity") +
  coord_flip() +
  theme_minimal()
```



The distribution of humidity varies by month, but there does not seem to be significant differences. We can

further explore the role of humidity by comparing it to other weather variables and flight delays.

```
# Precipitation by Month

monthlyprecip <- weather %>%
  group_by(month) %>%
  summarise(avgmonthlyprecip = mean(precip, na.rm = TRUE))

ggplot(monthlyprecip, aes(x = month, y = avgmonthlyprecip)) +
  geom_line(color = "palevioletred2") +
  geom_point(color = "palevioletred4") +
  labs(title = "Average Monthly Precipitation",
       x = "Month",
       y = "Average Precipitation") +
  scale_x_continuous(breaks = 1:12) +
  theme_minimal()
```



The average precipitation for each month varies greatly. We can see that spring months have the greatest average precipitation. When answering our research question, we can see if greater precipitation correlates to flight delays.

```
# Correlation Between Variables

cor(weather$precip, weather$visib, use = "complete.obs")
```

```
## [1] -0.3199118
```

```
cor(weather$humid, weather$visib, use = "complete.obs")
```

```
## [1] -0.5167424
```

We can explore the correlation between different weather variables and see how they may work together to impact flight delays.

Airport Dataset EDA

```
cat("\nMissing values per column:\n")
```

```
##
## Missing values per column:
```

```
print(colSums(is.na(airports)))
```

```
##   faa  name   lat   lon   alt    tz   dst tzone
##     0     0     0     0     0     0     0     3
```

```
cat("\nColumn types and structure:\n")
```

```
##
## Column types and structure:
```

```
print(glimpse(airports))
```

```
## Rows: 1,458
## Columns: 8
## $ faa   <chr> "04G", "06A", "06C", "06N", "09J", "0A9", "0G6", "0G7", "0P2", "~
## $ name  <chr> "Lansdowne Airport", "Moton Field Municipal Airport", "Schaumbur~
## $ lat   <dbl> 41.13047, 32.46057, 41.98934, 41.43191, 31.07447, 36.37122, 41.4~
## $ lon   <dbl> -80.61958, -85.68003, -88.10124, -74.39156, -81.42778, -82.17342~
## $ alt   <dbl> 1044, 264, 801, 523, 11, 1593, 730, 492, 1000, 108, 409, 875, 10~
## $ tz    <dbl> -5, -6, -6, -5, -5, -5, -5, -5, -5, -8, -5, -6, -5, -5, -5, -5, ~
## $ dst   <chr> "A", "A", "A", "A", "A", "A", "A", "A", "U", "A", "A", "U", "A",~
## $ tzone <chr> "America/New_York", "America/Chicago", "America/Chicago", "Ameri~
## # A tibble: 1,458 x 8
##    faa   name                            lat    lon   alt    tz dst   tzone
##    <chr> <chr>                         <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
##  1 04G   Lansdowne Airport              41.1  -80.6  1044    -5 A     America/~
##  2 06A   Moton Field Municipal Airport  32.5  -85.7   264    -6 A     America/~
##  3 06C   Schaumburg Regional            42.0  -88.1   801    -6 A     America/~
##  4 06N   Randall Airport                41.4  -74.4   523    -5 A     America/~
##  5 09J   Jekyll Island Airport          31.1  -81.4    11    -5 A     America/~
##  6 0A9   Elizabethton Municipal Airport 36.4  -82.2  1593    -5 A     America/~
##  7 0G6   Williams County Airport        41.5  -84.5   730    -5 A     America/~
##  8 0G7   Finger Lakes Regional Airport  42.9  -76.8   492    -5 A     America/~
##  9 0P2   Shoestring Aviation Airfield   39.8  -76.6  1000    -5 U     America/~
## 10 0S9   Jefferson County Intl          48.1 -123.    108    -8 A     America/~
## # i 1,448 more rows
```

```
cat("\nFirst few rows:\n")
```

```
##
## First few rows:
```

```r
print(head(airports))
```

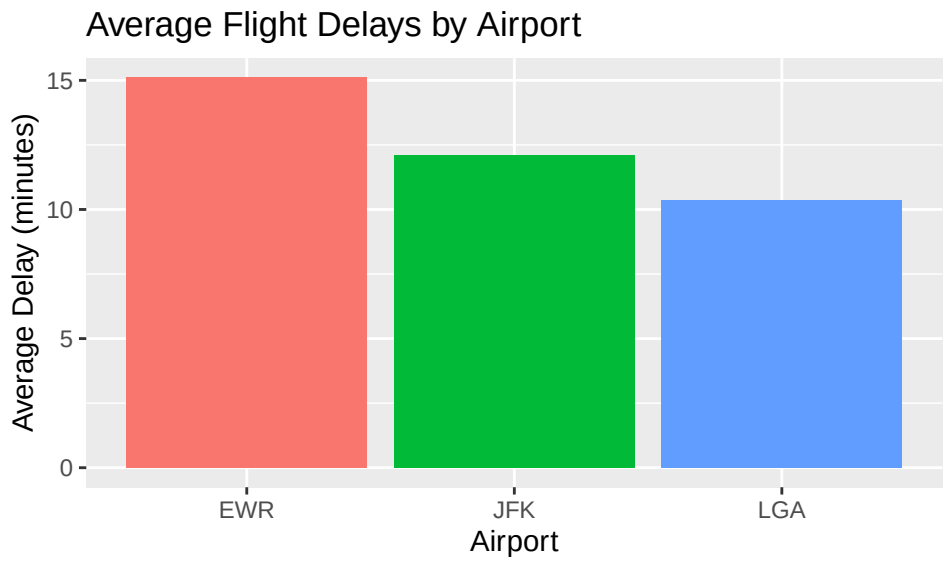
```
## # A tibble: 6 x 8
##   faa   name                            lat   lon   alt    tz dst   tzone
##   <chr> <chr>                         <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 04G   Lansdowne Airport              41.1 -80.6  1044    -5 A     America/Ne~
## 2 06A   Moton Field Municipal Airport  32.5 -85.7   264    -6 A     America/Ch~
## 3 06C   Schaumburg Regional            42.0 -88.1   801    -6 A     America/Ch~
## 4 06N   Randall Airport                41.4 -74.4   523    -5 A     America/Ne~
## 5 09J   Jekyll Island Airport          31.1 -81.4    11    -5 A     America/Ne~
## 6 0A9   Elizabethton Municipal Airport 36.4 -82.2  1593    -5 A     America/Ne~
```

```r
# Looking at the variation of flight delays between airports

flights_mod <- flights |>
  mutate(month = factor(month, levels = 1:12, labels = month.abb))

delays_airport <- flights_mod |>
  group_by(origin) |>
  summarise(mean_delay = mean(dep_delay, na.rm = TRUE),
            median_delay = median(dep_delay, na.rm = TRUE),
            sd_delay = sd(dep_delay, na.rm = TRUE))
airport_avg_delay <- ggplot(delays_airport, aes(x = origin, y = mean_delay, fill = origin)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  labs(title = "Average Flight Delays by Airport", x = "Airport", y = "Average Delay (minutes)")

# Looking at variation of cancelled flights by airports over a period of 12 months

cancelled_flights <- flights_mod |>
  filter(!is.na(month)) |>
  group_by(origin, month) |>
  summarise(cancelled = sum(is.na(dep_time)))
```

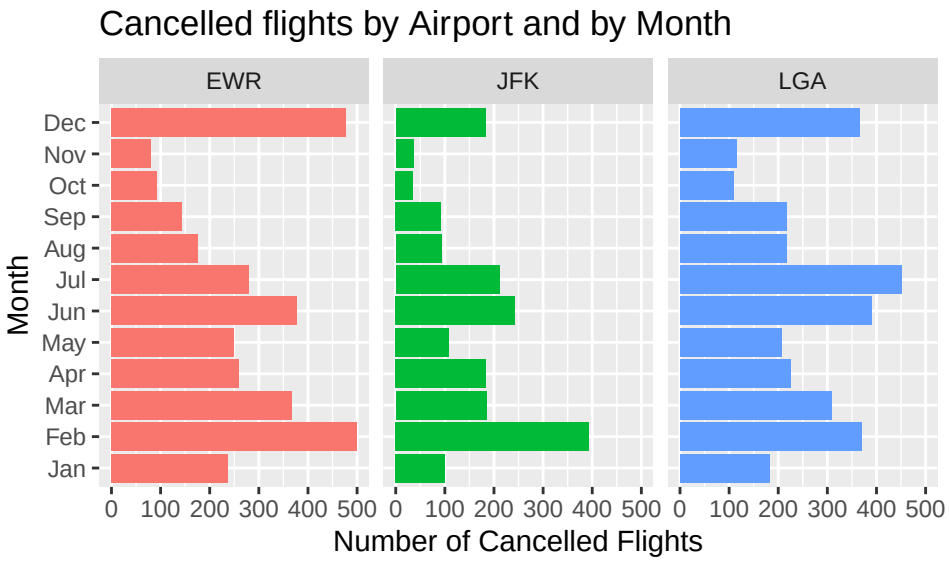
```
## `summarise()` has grouped output by 'origin'. You can override using the
## `.groups` argument.
```

```r
airport_cancelled_month <- ggplot(cancelled_flights, aes(x = month, y = cancelled, fill = origin)) +
  geom_bar(stat = "identity", position = "dodge", show.legend = FALSE) +
  facet_wrap(~origin) +
  coord_flip() +
  labs(title = "Cancelled flights by Airport and by Month", x = "Month", y = "Number of Cancelled Flight

# Looking at the flight patterns for each airport over a period of 12 months
flight_patterns <- flights_mod |>
  group_by(origin, month) |>
  summarise(avg_flights = n() / length(unique(day)), .groups = "drop")

airport_avg_flight_month <- ggplot(flight_patterns, aes(x = month, y = avg_flights, color = origin, grou
  geom_line() +
  labs(title = "Average Flight Patterns from Airport by Month", x = "Month", y = "Average Number of Flig

# Looking at carriers departing from airports over a period of 12 months
```

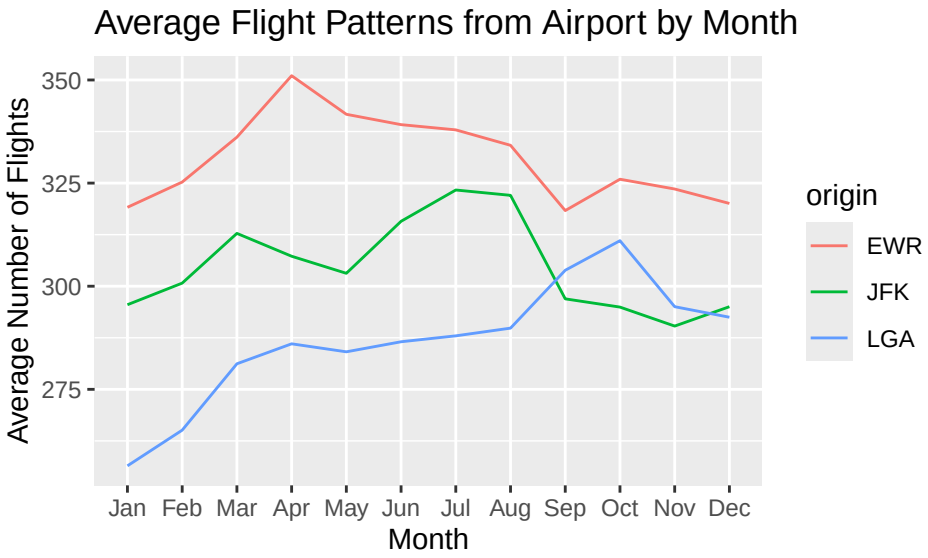
```r
flights_with_airlines <- flights_mod |>
  left_join(airlines, by = "carrier")

carrier_variation <- flights_with_airlines |>
  group_by(origin, name, month)

airport_carriers_month <- ggplot(carrier_variation, aes(x = name, y = month, fill = name)) +
  geom_bar(stat = "identity", position = "dodge", show.legend = FALSE) +
  facet_wrap(~origin) +
  coord_flip() +
  labs(title = "Variation of Carriers Departing from Airports", x = "Carrier", y = "Number of Flights")

# Plots
  airport_avg_delay
```



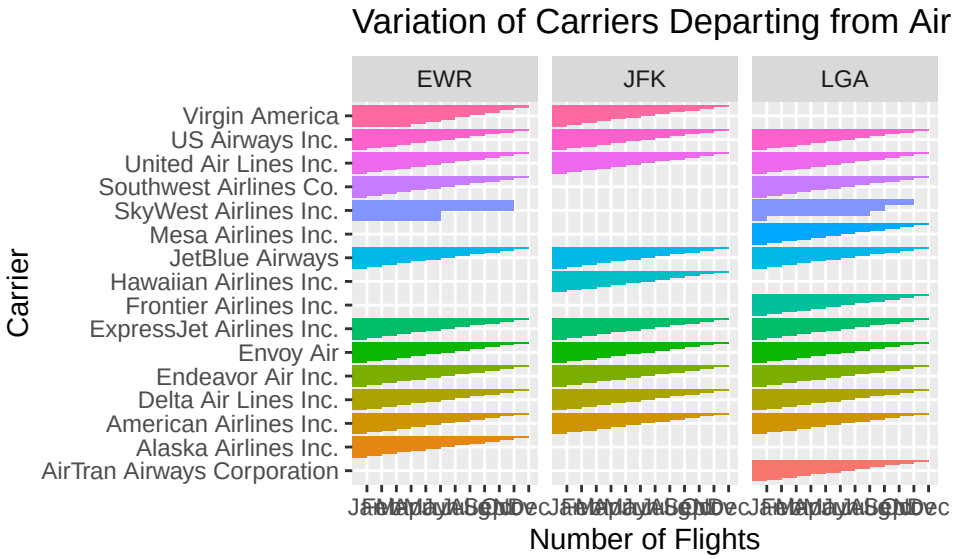

```r
  airport_cancelled_month
```




```
airport_avg_flight_month
```




```
airport_carriers_month
```



Summary: 1: When looking at the Average Flight Delays we can see that Newark Liberty Airport (EWR) has the highest flight delays averaging at 15 minutes, followed by John F. Kennedy Airport (JFK) and LaGuardia Airport (LGA). Based on this information, further investigation into potential causes like weather and airline operations could provide more information concerning the delays at EWR.

2. When looking at cancelled flights by airport and month, distinct patterns emerge for EWR, JFK, and LGA. EWR experiences a higher frequency of cancellations throughout the year, with notable peaks in February, March, and December. This may be attributed to winter weather and holiday congestion. JFK generally has fewer cancellations, expect for a significant spike in February, this may also be weather-related as this pattern is seen in EWR. LGA shows variability in cancellations, with higher instances during the summer months and December, possibly influenced by seasonal weather and holiday travel. These patterns suggest that weather conditions and holiday would be worth investigating to better understand their effects on flight delays.

3. The line plot illustrating average flight patterns by month reveals distinct seasonal trends for each airport. EWR exhibits a significant peak in April, averaging 350 flights, followed by a gradual decrease for the remainder of the year. JFK displays more fluctuation throughout the year, with its highest average number of flights, just under 325, occurring during the summer months, followed by a decrease and a slight uptick during the holiday season. LGA, while experiencing the lowest overall flight numbers, shows a gradual incrase with a peak in October, followed by a decline. These patterns may suggest that EWR and JFK, as international airports, are more influenced by seasonal travel trends, whereas LGA, with its domestic focus, shows a more consistent traffic flow.

4. The bar chart shows the distribution of flights by carrier for each airport. EWR and JFK have a variety of carriers, while LGA shows a more concentrated distribution with fewer carriers. The variety of carriers at EWR and JFK may contribute to increased operational complexity and may potentially lead to higher delays. While LGA's more concentrated carriers might allow for efficient operations, however this isn't the case as LGA experienced a higher number of cancelled flights. This would be interesting to investigate is it implies that factors other than weather, such as operational or logistical challenges, could be influencing flight delays and cancellations.

Analysis Approach Plan

Assumptions: All variables are independent

The process of analysis will involve data cleaning after forming our question, basic exploration of the data, comparison of certain datasets with other datasets, visualization of the data, and an interpretation of the data/results. Cleaning of the data will deal with tasks like handling empty cells/columns and NA values. When it comes to exploratory data analysis, we plan on using tools such as histograms and boxplots to gain an understanding of the data and identify patterns and relationships. The statistical analysis that we plan on performing with the data will most likely involve making comparisons between groups to compare airlines, times, and other metrics to make our overall claim. For example, we might be comparing trends in time performance by weeks or month between different airlines to gain a better understanding of how differences in airlines affect delays. In terms of data visualization, we will most likely be using line graphs for trends over time when it comes to comparing flight time under different variables and heatmaps/scatterplots for flight delays to help communicate our findings. Finally, interpretation of the data will involve us answering the proposed question by summarizing our statistics/findings as well as through the presentation of graphical evidence.

Analysis:

Question 1: How do weather conditions affect flight delays?

1. Are specific weather variables (e.g., precipitation, temperature, humidity) correlated with arrival delays?

```
head(weather)
```

```
## # A tibble: 6 x 15
##   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed wind_gust
##   <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>     <dbl>
## 1 EWR     2013     1     1     1  39.0  26.1  59.4      270      10.4         NA
## 2 EWR     2013     1     1     2  39.0  27.0  61.6      250       8.06        NA
## 3 EWR     2013     1     1     3  39.0  28.0  64.4      240      11.5         NA
## 4 EWR     2013     1     1     4  39.9  28.0  62.2      250      12.7         NA
## 5 EWR     2013     1     1     5  39.0  28.0  64.4      260      12.7         NA
## 6 EWR     2013     1     1     6  37.9  28.0  67.2      240      11.5         NA
## # i 4 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,
## #   time_hour <dttm>
```

```
head(flights)
```

```
## # A tibble: 6 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
```

```
## # 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>
```

```
not_canceled <- filter(flights, !is.na(dep_delay), !is.na(arr_delay))
head(not_canceled)
```

```
## # A tibble: 6 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
## # 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>
```

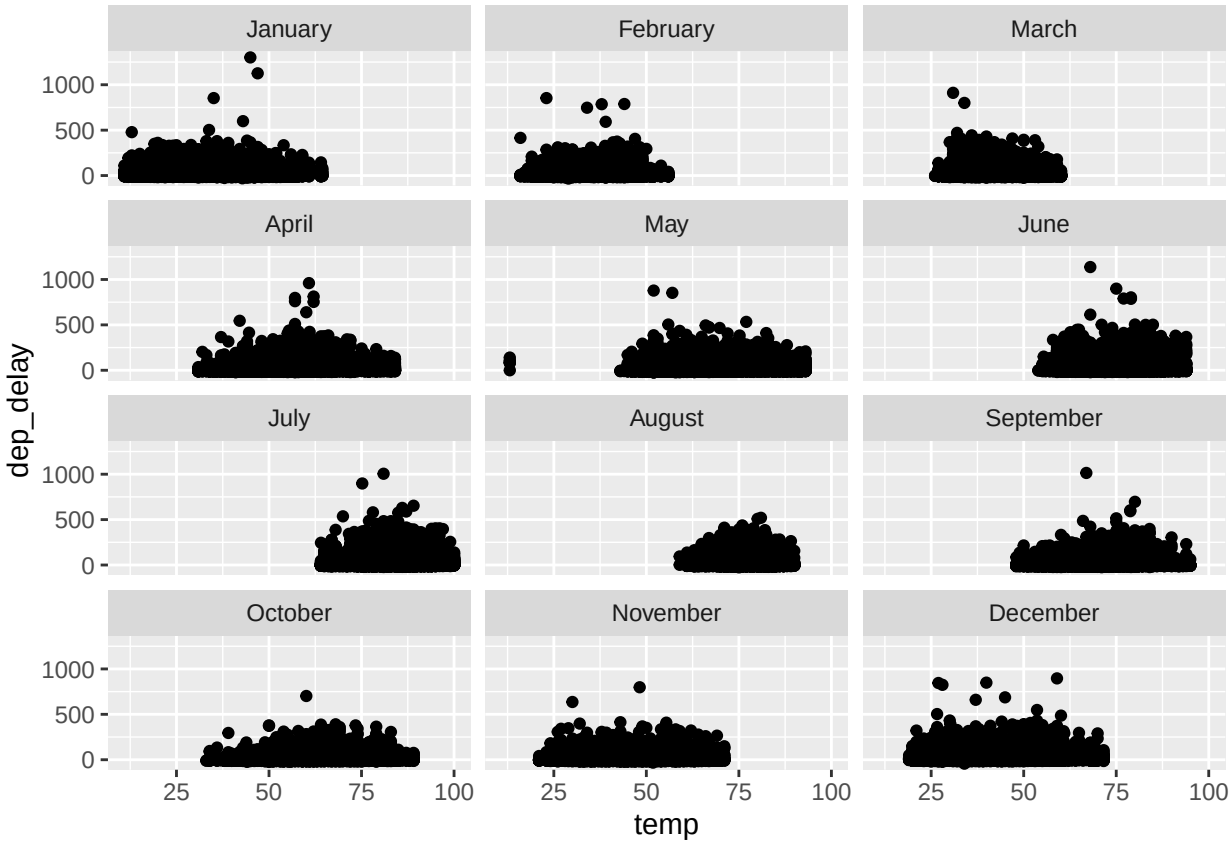
join columns of weather and uncancelled flights

```
flights_weather <- left_join(not_canceled, weather, by = c("year", "month", "day", "hour", "origin"))

flights_weather <- flights_weather |>
  filter(!is.na(dep_delay))
flights_weather
```

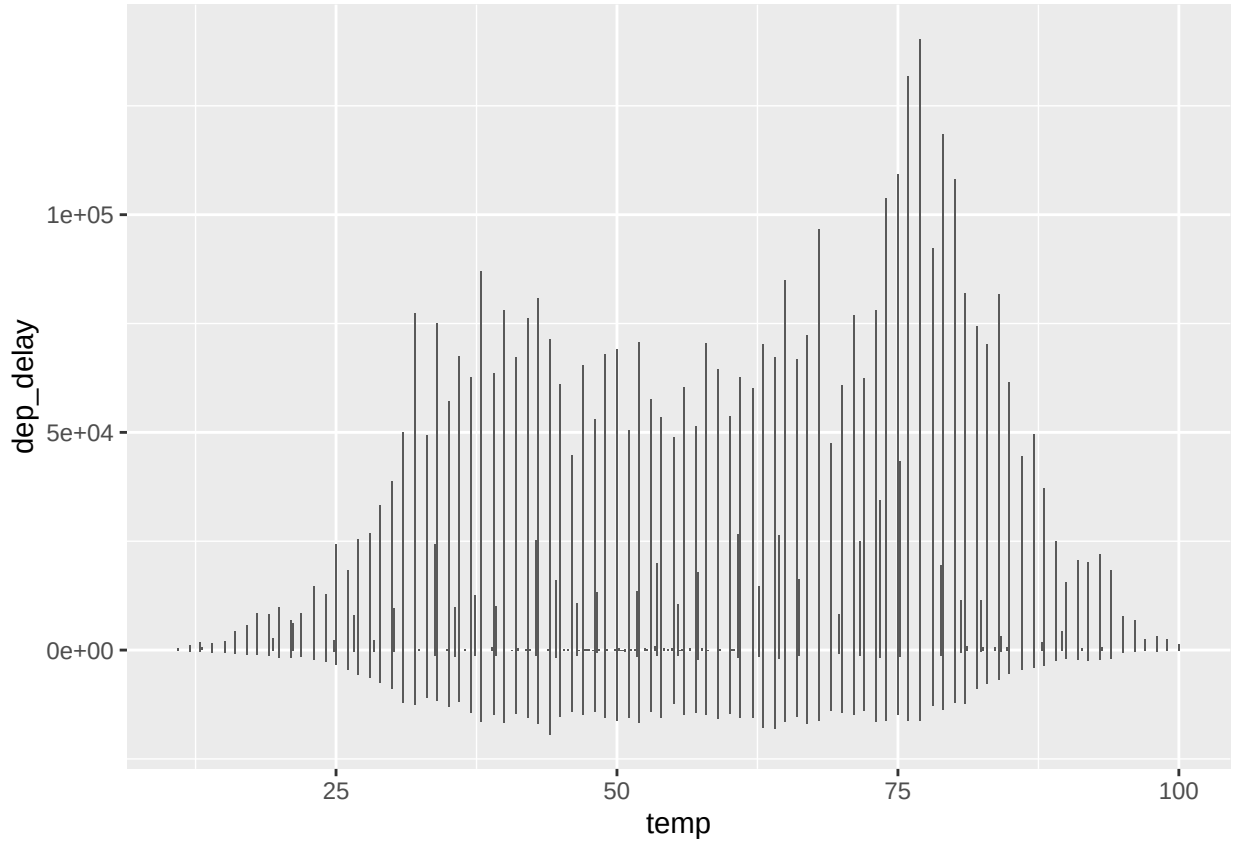
```
## # A tibble: 327,346 x 29
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      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
## # i 327,336 more rows
## # i 21 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.x <dttm>, temp <dbl>, dewp <dbl>,
## #   humid <dbl>, wind_dir <dbl>, wind_speed <dbl>, wind_gust <dbl>,
## #   precip <dbl>, pressure <dbl>, visib <dbl>, time_hour.y <dttm>
```

Temperature by month and departure delays

```r
#mutate to month
f_w_by_month <- flights_weather |>
  mutate(month = factor(month, levels = 1:12, labels = month.name))

ggplot(f_w_by_month, aes(x = temp, y = dep_delay)) +
  geom_point() +
  facet_wrap(~ month, ncol = 3)
```

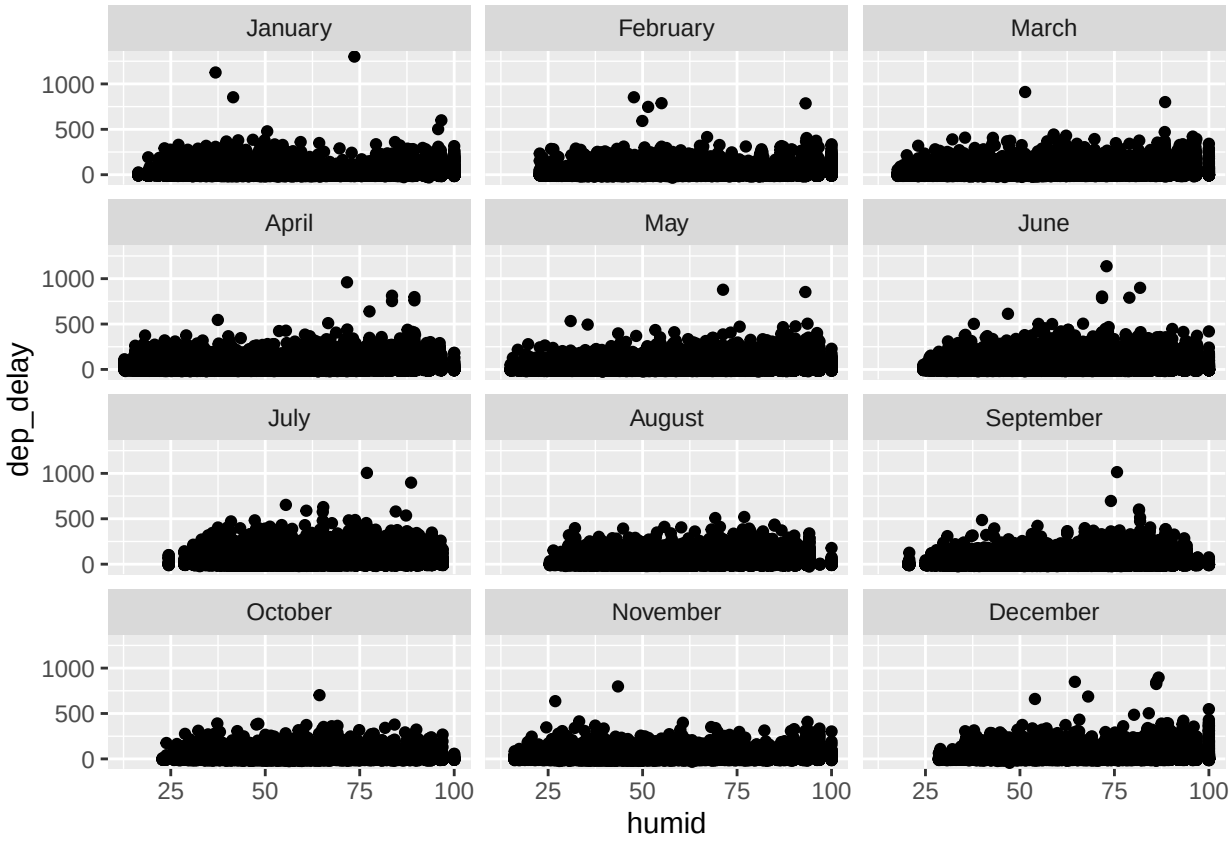



```r
ggplot(f_w_by_month, aes(x = temp, y = dep_delay)) +
  geom_bar(stat = "identity")
```

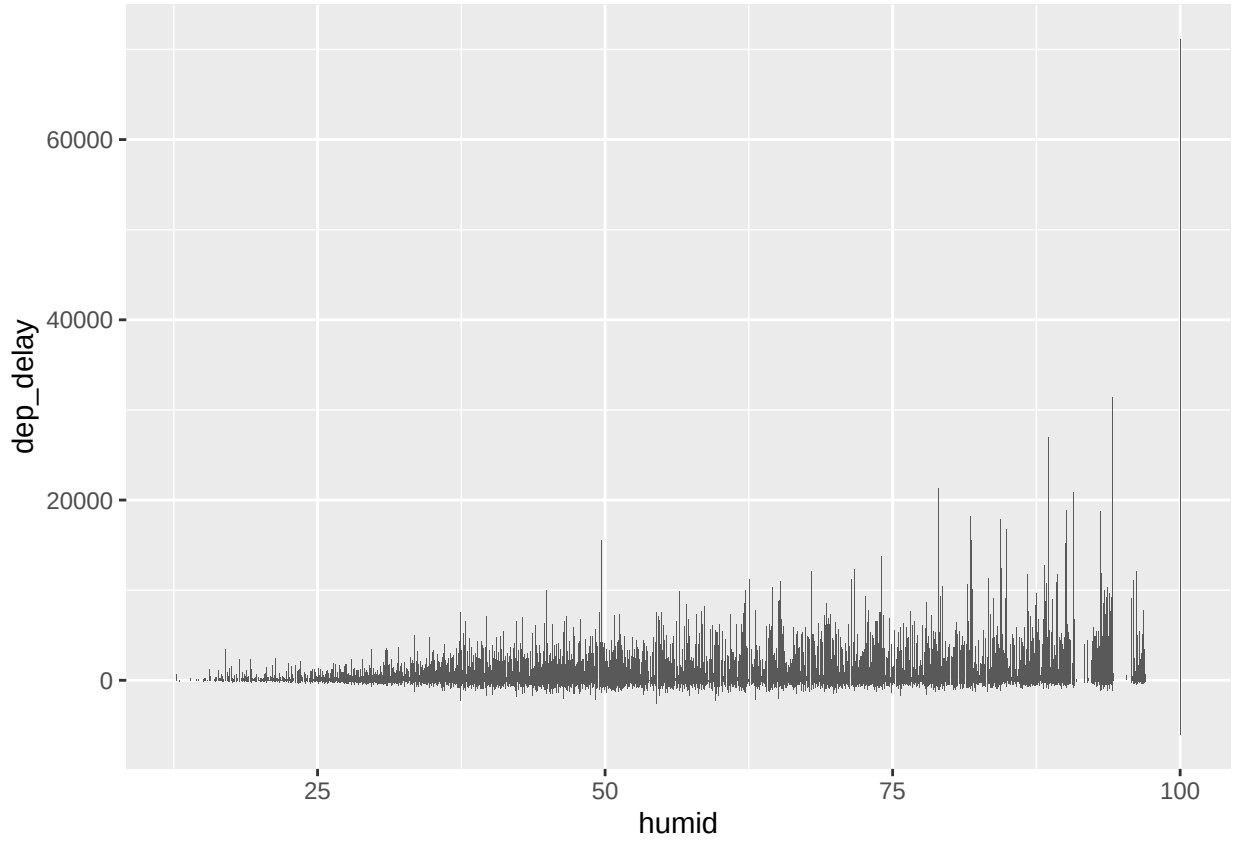


Humidity by month and departure delays

```
ggplot(f_w_by_month, aes(x = humid, y = dep_delay)) +
  geom_point() +
  facet_wrap(~ month, ncol = 3)
```

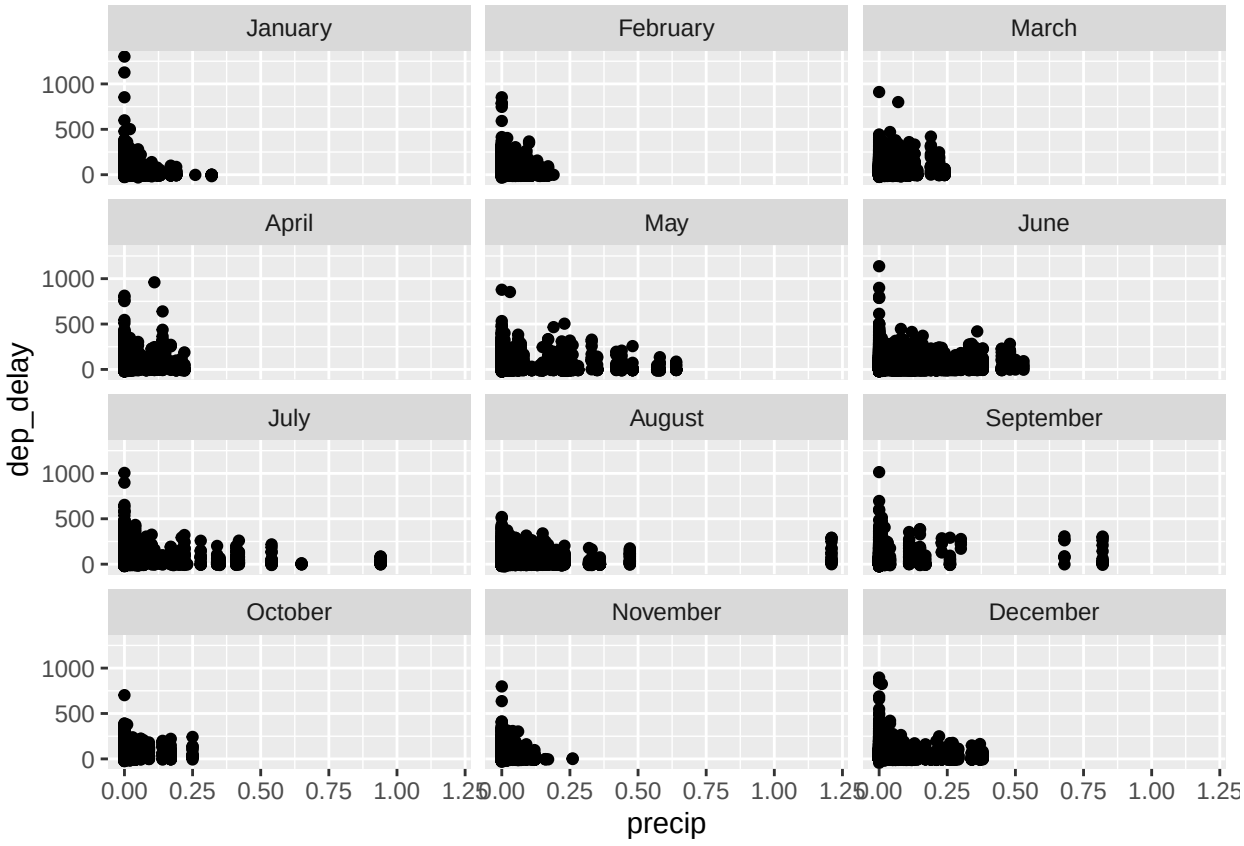



```
ggplot(f_w_by_month, aes(x = humid, y = dep_delay)) +
  geom_bar(stat = "identity")
```

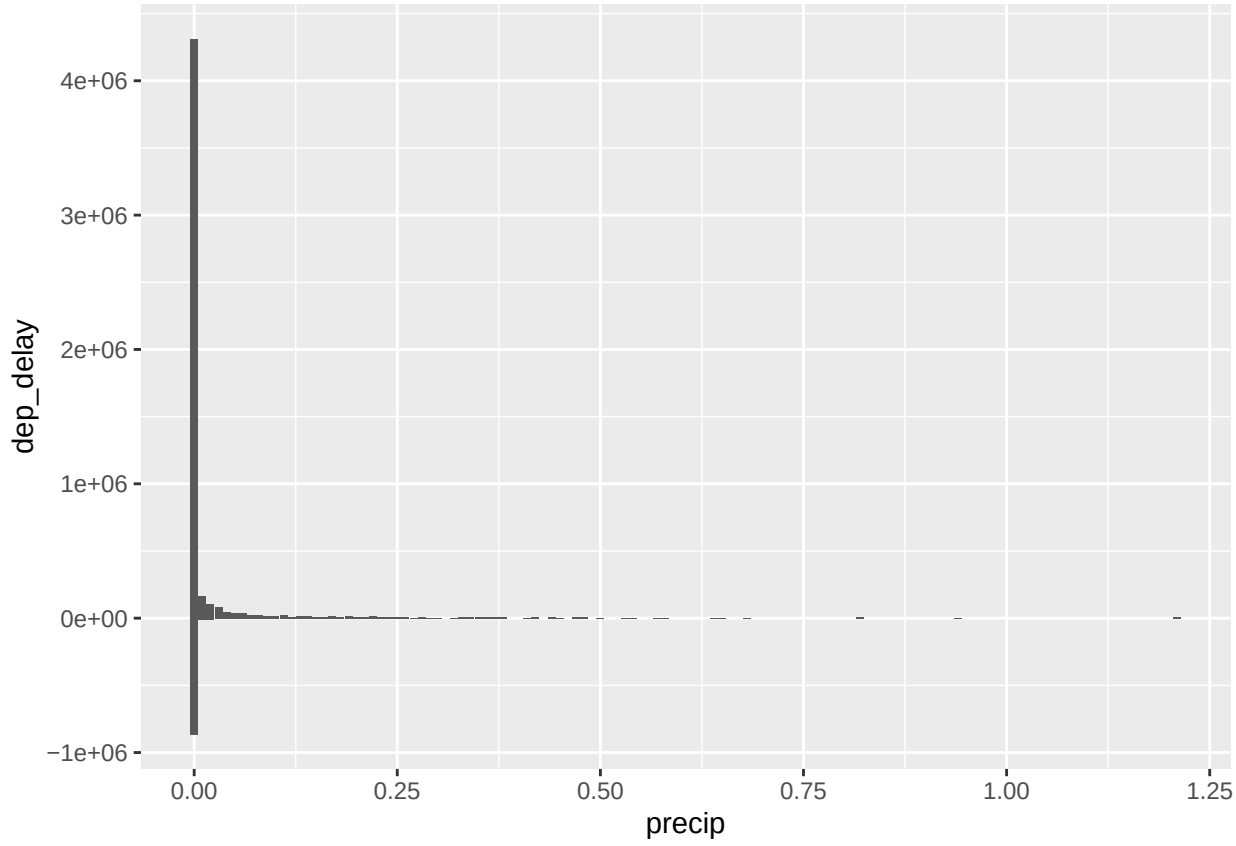


Precipitation by month and departure delays

```
ggplot(f_w_by_month, aes(x = precip, y = dep_delay)) +
  geom_point() +
  facet_wrap(~ month, ncol = 3)
```




```
ggplot(f_w_by_month, aes(x = precip, y = dep_delay)) +
  geom_bar(stat = "identity")
```




```
weather_vars <- f_w_by_month |>
  dplyr::select(temp, wind_speed, precip, visib, dep_delay)

#filter out extreme
weather_vars <- filter(weather_vars, dep_delay <= 300)
cor(weather_vars, use = "complete.obs")
```

```
##                   temp  wind_speed       precip       visib   dep_delay
## temp        1.000000000 -0.14194220  0.009347564  0.09032313  0.06094409
## wind_speed -0.141942195  1.00000000  0.032506147  0.07188803  0.05059973
## precip      0.009347564  0.03250615  1.000000000 -0.32062812  0.09211293
## visib       0.090323128  0.07188803 -0.320628119  1.00000000 -0.09163311
## dep_delay   0.060944091  0.05059973  0.092112931 -0.09163311  1.00000000
```

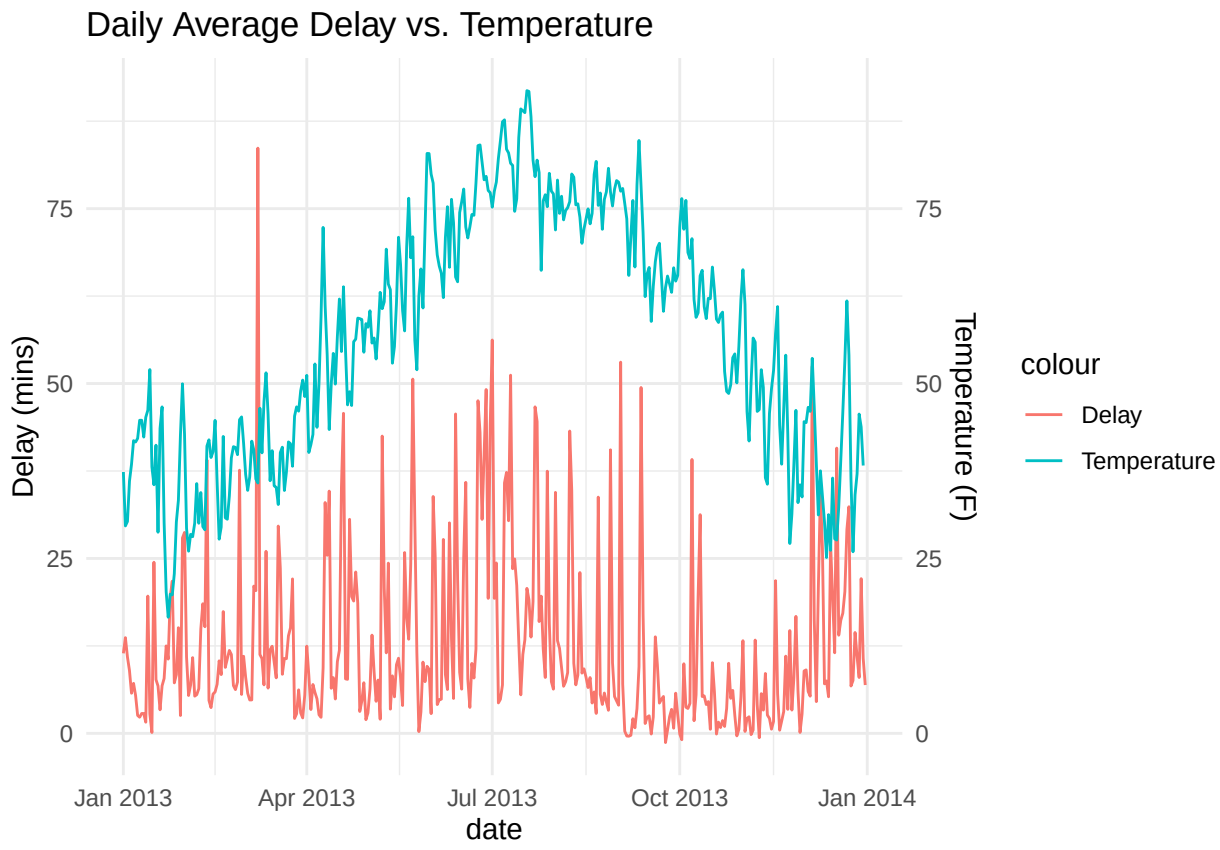
```
summary(f_w_by_month$precip)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.0000  0.0000  0.0042  0.0000  1.2100    1527
```

- Visibility and departure delay has the weakest negative correlation
- Strongest positive linear relationship is between precipitation and departure delay

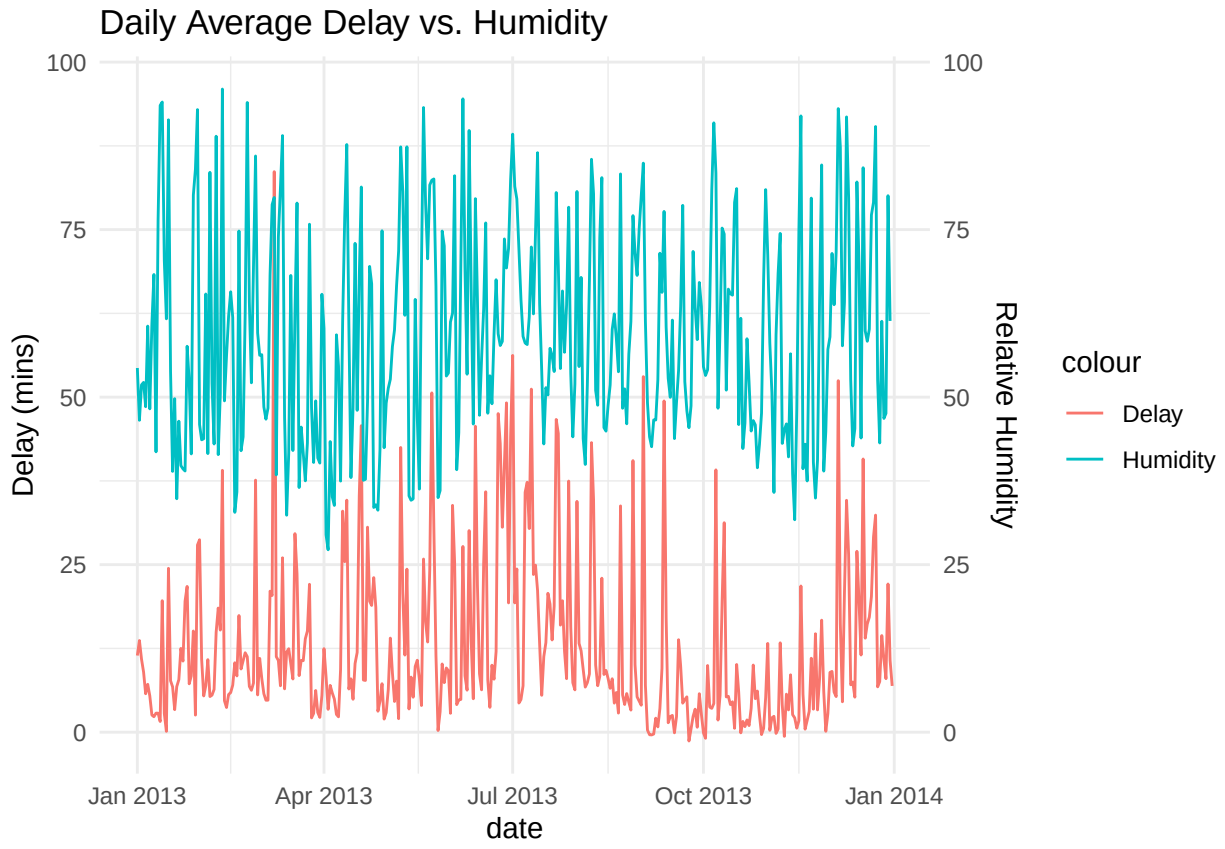
Time Series (Precipitation and Departure Delay)

```
daily_data <- f_w_by_month |>
  group_by(year, month, day) |>
  summarise(
    mean_delay = mean(dep_delay, na.rm = TRUE),
    mean_temp = mean(temp, na.rm = TRUE),
    mean_humid = mean(humid, na.rm = TRUE),
    mean_precip = mean(precip, na.rm = TRUE),
    .groups = "keep"
  )

#make the date column
daily_data <- mutate(daily_data,
                     monthly_num = match(month, month.name),
                     date = as.Date(paste(year, monthly_num, day, sep = "-")))

#delays and weather variables over time
ggplot(daily_data, aes(x = date)) +
  geom_line(aes(y = mean_delay, color = "Delay")) +
  geom_line(aes(y = mean_temp, color = "Temperature")) +
  scale_y_continuous(sec.axis = sec_axis(~., name = "Temperature (F)")) +
  labs(title = "Daily Average Delay vs. Temperature", y = "Delay (mins)") +
  theme_minimal()
```

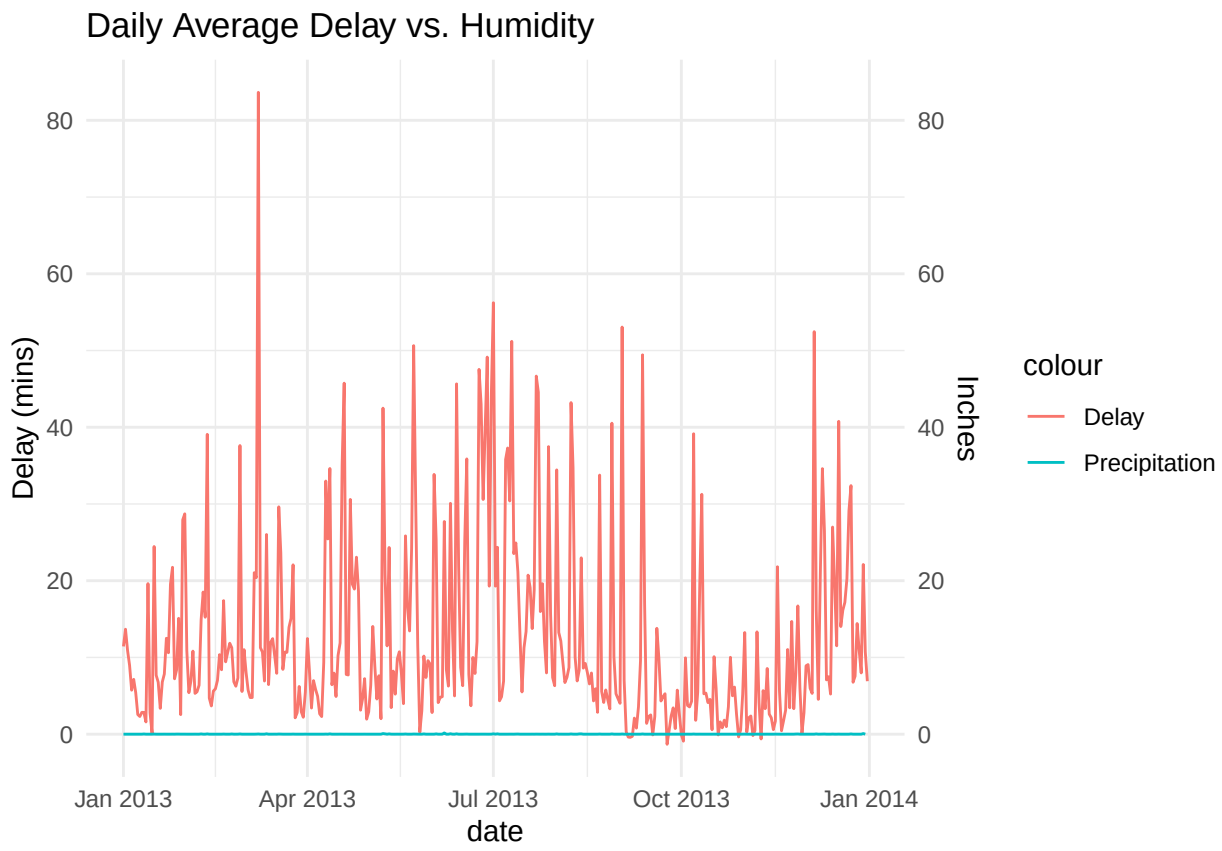



```
ggplot(daily_data, aes(x = date)) +
  geom_line(aes(y = mean_delay, color = "Delay")) +
```

```
  geom_line(aes(y = mean_humid, color = "Humidity")) +
  scale_y_continuous(sec.axis = sec_axis(~., name = "Relative Humidity")) +
  labs(title = "Daily Average Delay vs. Humidity", y = "Delay (mins)") +
  theme_minimal()
```




```
ggplot(daily_data, aes(x = date)) +
  geom_line(aes(y = mean_delay, color = "Delay")) +
  geom_line(aes(y = mean_precip, color = "Precipitation")) +
  scale_y_continuous(sec.axis = sec_axis(~., name = "Inches")) +
  labs(title = "Daily Average Delay vs. Humidity", y = "Delay (mins)") +
  theme_minimal()
```



Both variables rise and fall together for Temperature vs. Average Delay and Humidity vs. Average Delay. This suggests that there is a seasonal or temporal pattern and that a positive association exists. Since precipitation is mostly entered as 0, it will provide little variability for analysis.

Linear Regression Model

```
#omit missing values
model_data <- flights_weather |>
  dplyr::select(arr_delay, precip, temp, humid, visib, carrier, origin, month, hour) |>
  filter(!is.na(arr_delay), !is.na(precip), !is.na(temp), !is.na(humid), !is.na(visib), !is.na(carrier)

#linear regression model
lm_model <- lm(arr_delay ~ precip + temp + humid + visib + carrier + origin + month + hour, data = model
summary(lm_model)
```

```
##
## Call:
## lm(formula = arr_delay ~ precip + temp + humid + visib + carrier +
##     origin + month + hour, data = model_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -158.98  -22.80   -8.39    9.33 1273.59
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -22.654133   0.792132 -28.599  < 2e-16 ***
## precip       88.506538   2.690666  32.894  < 2e-16 ***
## temp          0.054689   0.004390  12.459  < 2e-16 ***
## humid         0.306451   0.004735  64.717  < 2e-16 ***
## visib        -1.362518   0.047087 -28.936  < 2e-16 ***
## carrierAA    -5.209214   0.412458 -12.630  < 2e-16 ***
## carrierAS   -19.179063   1.658764 -11.562  < 2e-16 ***
## carrierB6     2.156817   0.374819   5.754 8.71e-09 ***
## carrierDL    -5.059069   0.388304 -13.029  < 2e-16 ***
## carrierEV     7.500208   0.421570  17.791  < 2e-16 ***
## carrierF9    14.104919   1.682359   8.384  < 2e-16 ***
## carrierFL    12.425374   0.846365  14.681  < 2e-16 ***
## carrierHA    -5.354069   2.343375  -2.285  0.02233 *
## carrierMQ     3.827175   0.438167   8.735  < 2e-16 ***
## carrierOO    -0.952014   7.948453  -0.120  0.90466
## carrierUA    -4.411628   0.412649 -10.691  < 2e-16 ***
## carrierUS    -3.530563   0.465162  -7.590 3.21e-14 ***
## carrierVX    -3.202371   0.684845  -4.676 2.93e-06 ***
## carrierWN     3.104639   0.534568   5.808 6.34e-09 ***
## carrierYV     6.030891   1.875750   3.215  0.00130 **
## originJFK    -4.323413   0.250215 -17.279  < 2e-16 ***
## originLGA    -0.665356   0.229788  -2.896  0.00379 **
## month        -0.359562   0.022856 -15.732  < 2e-16 ***
## hour          1.888051   0.016580 113.878  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.76 on 325778 degrees of freedom
## Multiple R-squared:  0.08357,    Adjusted R-squared:  0.0835
## F-statistic:  1292 on 23 and 325778 DF,  p-value: < 2.2e-16
```

- R^2 is 0.035, 8.35% of the variance in arrival delays is explained
- p < 2.2e-16, model is statistically significant overall
- Heavy precipitation has a strong, positive effect on delay, Estimate = +88.5
- Temperature has a small, positive effect, Estimate = +0.055
- Higher humidity is mildly associated with more delay

While predictive power is mild due to the large amount of noise in the delay data, this model provides useful insight into the impact of weather and delay factors. Since the model only explains 8.35% of the variance, it suggests that while weather has an impact on arrival delays, it only accounts for a small portion of the variance and a lot of other factors such as carrier, traffic, and other specific details relating to the plane contribute heavily to delays.

Cross-validation

```r
model_data <- na.omit(model_data)
lm_model2 <- glm(arr_delay ~ precip + temp + humid + visib, data = model_data)

set.seed(167)
cv_results <- cv.glm(data = model_data, glmfit = lm_model2, K = 10)

cv_results$delta
```

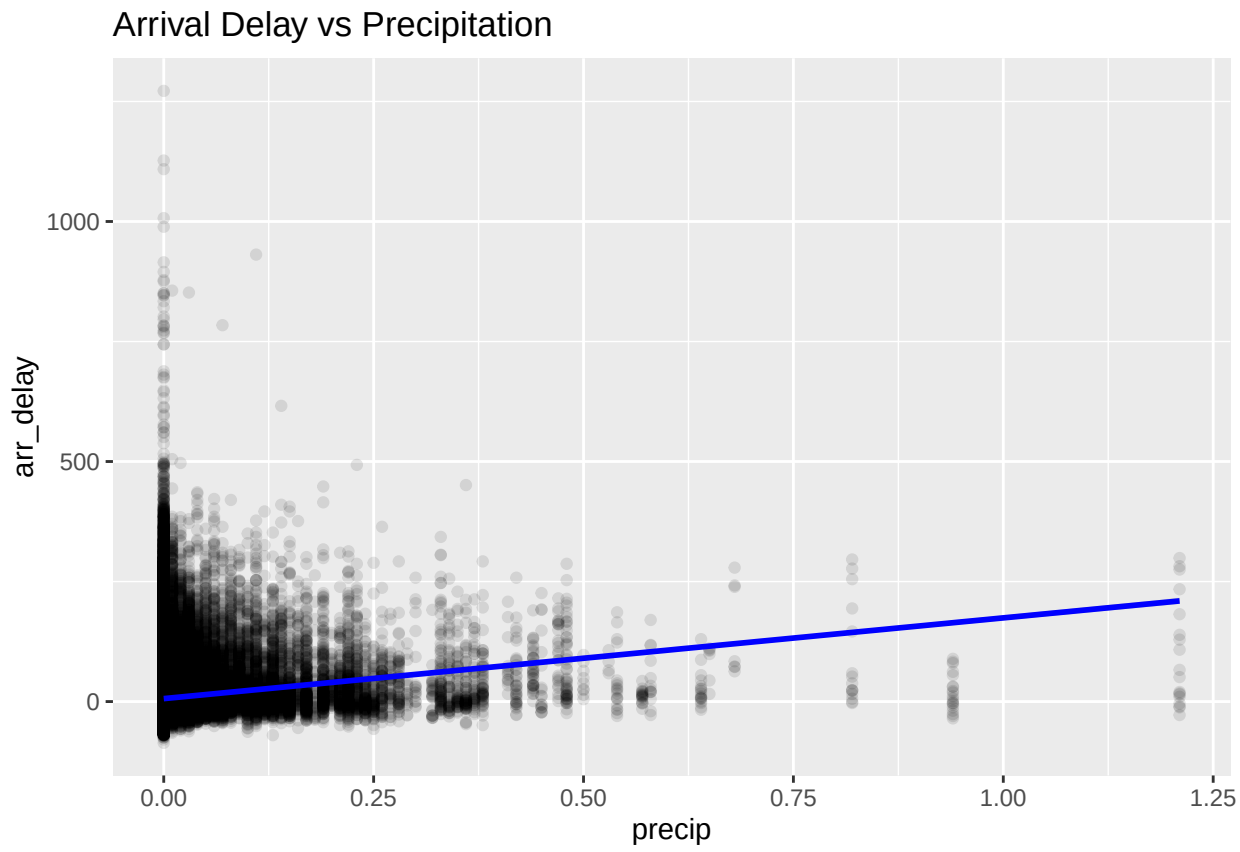
```
## [1] 1934.189 1934.182
```

Linear model has a Root Mean Square Error of ~44 minutes, predictions are about 44 minutes off from the actual arrival delays.

Visualization

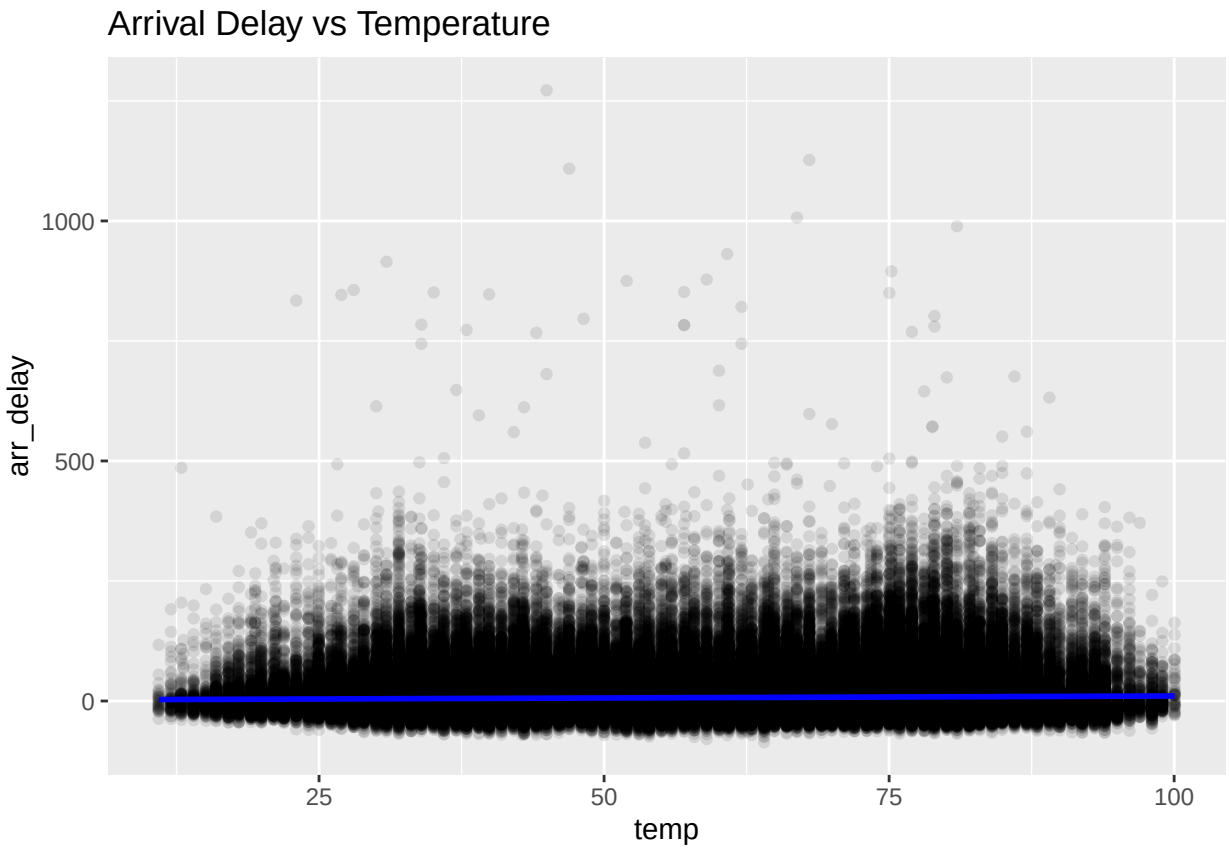
```
ggplot(model_data, aes(x = precip, y = arr_delay)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Arrival Delay vs Precipitation")
```

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



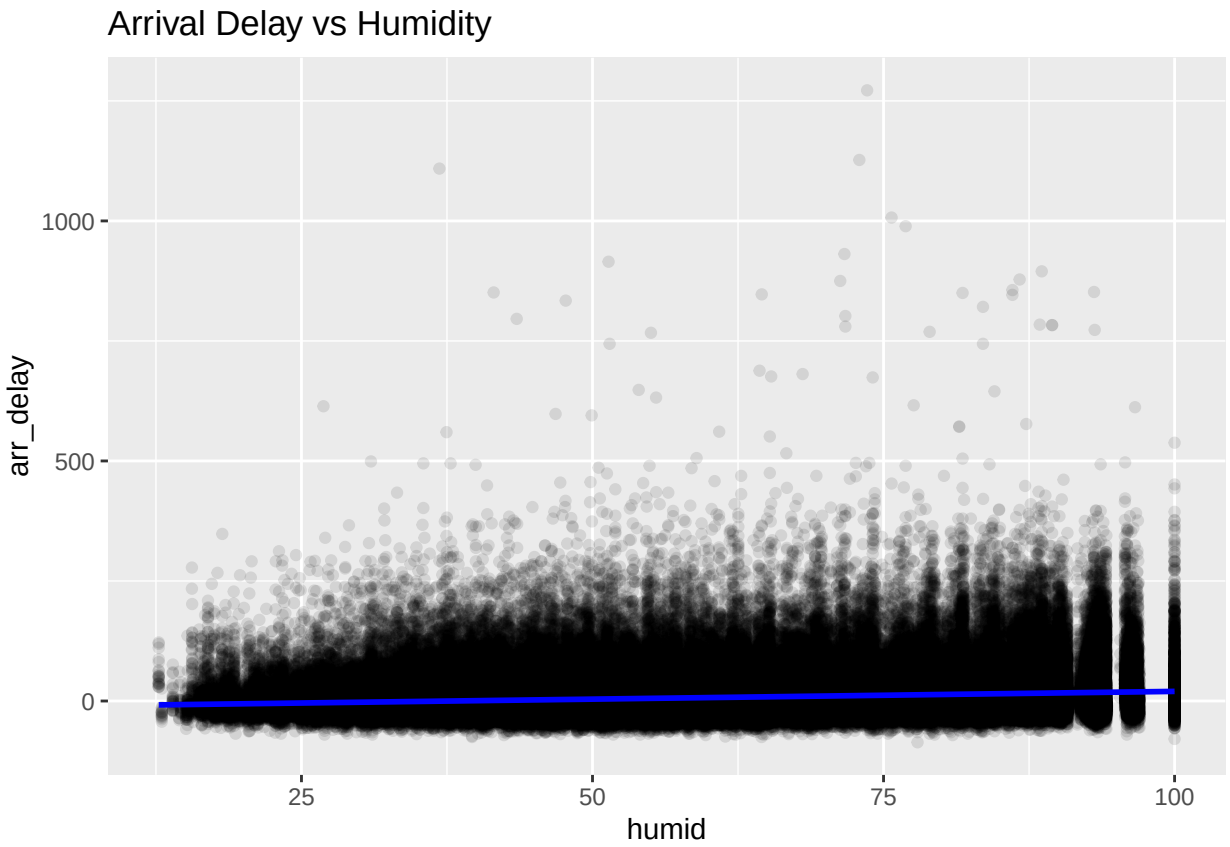

```
ggplot(model_data, aes(x = temp, y = arr_delay)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Arrival Delay vs Temperature")
```

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




```
ggplot(model_data, aes(x = humid, y = arr_delay)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Arrival Delay vs Humidity")
```

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



Interpretation:

The noticeable upward slope indicates a positive linear relationship for precipitation and arrival delays. Since most points cluster near zero precipitation, the delay increase is mostly evident during heavy rainfall. The slope for delays and temperature is close to zero, and the distribution appears to be very even, suggesting that temperature has a very small impact, as evident in the summary of the model. Since less points cluster near the lower end of humidity, and there is a small but existing positive slope, higher humidity is mildly associated with more delay.

```
flights_weather <- mutate(flights_weather, severe_delay = arr_delay > 60)

#create extreme weather vars and conditions
flights_weather <- flights_weather |>
  mutate(
    extreme_precip = precip > 0.5,
    low_visib = visib < 1,
    high_wind = wind_speed > 20,
    cold_temp = temp < 32,
    high_temp = temp > 90,
    extreme_weather = extreme_precip | low_visib | high_wind | cold_temp | high_temp
  )
```

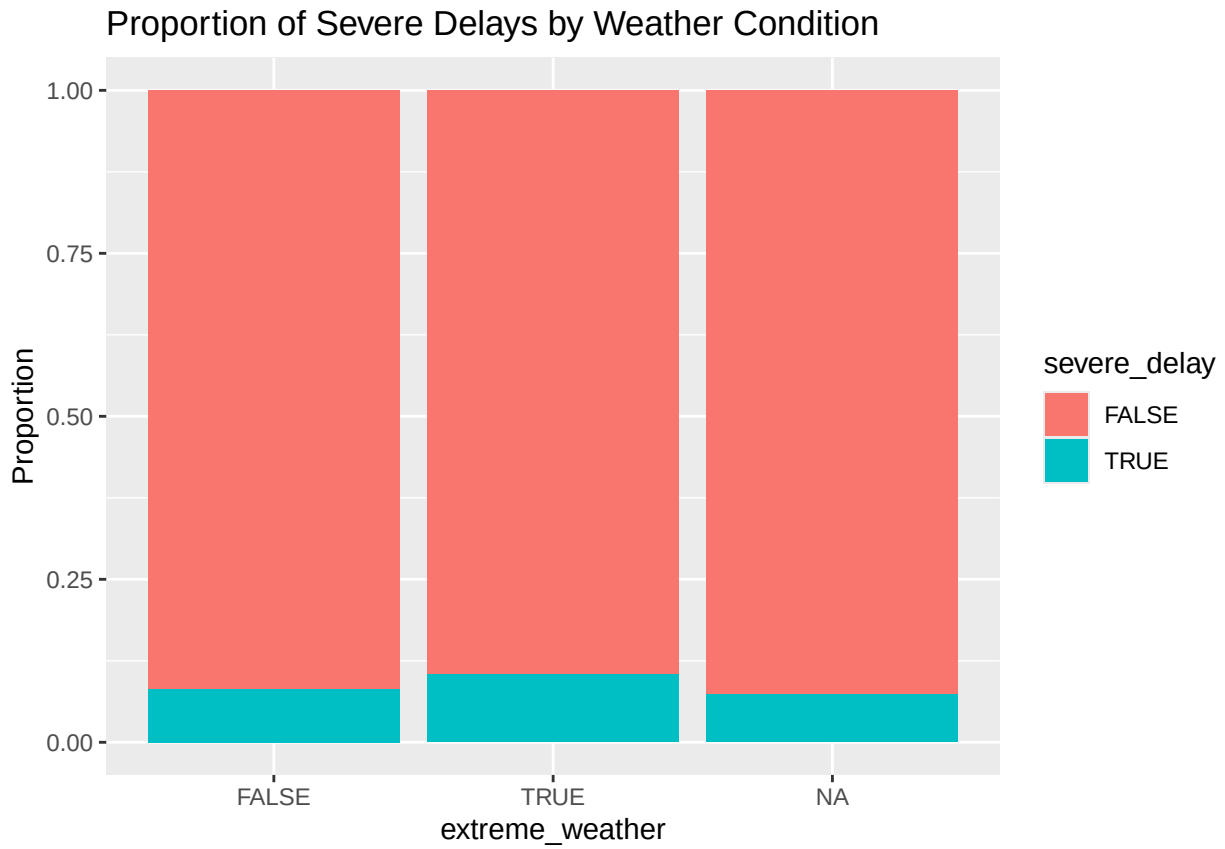
```r
#use glm for logistic regression
lm_severe <- glm(severe_delay ~ extreme_precip + low_visib + high_wind + cold_temp + high_temp, data = f
summary(lm_severe)
```

2. How do extreme weather conditions (e.g. heavy precipitation, low visibility, cold temperature) impact the likelihood of severe delays (e.g. delays > 60 minutes)?

```
##
## Call:
## glm(formula = severe_delay ~ extreme_precip + low_visib + high_wind +
##     cold_temp + high_temp, family = "binomial", data = flights_weather)
##
## Coefficients:
##                     Estimate Std. Error  z value Pr(>|z|)
## (Intercept)        -2.422470   0.006901 -351.032  < 2e-16 ***
## extreme_precipTRUE  1.548281   0.185093    8.365  < 2e-16 ***
## low_visibTRUE       1.079642   0.041145   26.240  < 2e-16 ***
## high_windTRUE       0.374081   0.023281   16.068  < 2e-16 ***
## cold_tempTRUE      -0.134758   0.024973   -5.396 6.81e-08 ***
## high_tempTRUE       0.480824   0.042428   11.333  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 189372  on 325723  degrees of freedom
## Residual deviance: 188367  on 325718  degrees of freedom
##   (1622 observations deleted due to missingness)
## AIC: 188379
##
## Number of Fisher Scoring iterations: 5
```

Flights in heavy rain are 4.66 times more likely to be severely delayed compared to flights without heavy precipitation. The odds of a severe delay is increased by a factor of 2.9 when low visibility (less than 1) is present in the case. In addition, high wind speed increases the odds of a severe delay by a factor of 1.45. Cold temperature seems to decrease the odds of a severe delay by 13%, and high temperature increases the odds of severe delays by 30%.

```r
#proportion chart of severity of weather and delay
flights_weather %>%
  ggplot(aes(x = extreme_weather, fill = severe_delay)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion", title = "Proportion of Severe Delays by Weather Condition")
```



Extreme weather cases make up the majority of severely delayed flights. However, the difference between extreme weather cases and the non extreme weather cases is not overwhelming which further suggests the idea that weather only impacts arrival delays up to a certain point, and that other factors unrelated to weather explain a large chunk of the variance.

Question 2: How do differences between airlines influence flight delays?

To answer this question, we can explore factors such as airlines, engines,

1. Do some airlines have more delays than others? H_0: All airlines have the same average delay.

H_A: All airlines do not have the same average delay.

We can test this by comparing the means of the different airlines.

```
flights_not_missing = flights %>%
  filter(!is.na(arr_delay))

flights_not_missing = flights_not_missing %>%
  left_join(airlines, by = "carrier")

top_airlines = flights_not_missing %>%
  count(name, sort = TRUE) %>%
  top_n(5) %>%
  pull(name)
```

```
## Selecting by n
```

```
flights_subset = flights_not_missing %>%
  filter(name %in% top_airlines)

# Testing assumptions before ANOVA
print("Levene Test for Equal Variances: ")
```

```
## [1] "Levene Test for Equal Variances: "
```

```
leveneTest(arr_delay ~ name, data = flights_subset)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      4  303.16 < 2.2e-16 ***
##       242539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Cannot use Shapiro-Wilk to test for normality due to large sample
# using ad test instead
ad.test(flights_subset$arr_delay)
```

```
##
##  Anderson-Darling normality test
##
## data:  flights_subset$arr_delay
## A = 17700, p-value < 2.2e-16
```

```r
length(flights_subset$arr_delay)
```

```
## [1] 242544
```

Before attempting to test our hypotheses with ANOVA, we check on the assumptions of equal variances and normality. Both are violated, but we can bypass the normality violation because of the large sample size. This means we can use a Welch Anova test instead of the regular Anova, which assumes that the variances are not equal.

```r
# Welch Anova Test
print("Welch Anova Test")
```

```
## [1] "Welch Anova Test"
```

```r
oneway.test(arr_delay ~ name, data = flights_subset, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  arr_delay and name
## F = 887.09, num df = 4, denom df = 113281, p-value < 2.2e-16
```

```r
# Games_howell in place of Tukey
flights_subset %>%
  games_howell_test(arr_delay ~ name)
```

```
## # A tibble: 10 x 8
##    .y.       group1     group2 estimate conf.low conf.high    p.adj p.adj.signif
##  * <chr>     <chr>      <chr>     <dbl>    <dbl>     <dbl>    <dbl> <chr>
##  1 arr_delay American ~ Delta~     1.28    0.426      2.13 4.15e- 4 ***
##  2 arr_delay American ~ Expre~    15.4    14.5       16.3  0        ****
##  3 arr_delay American ~ JetBl~     9.09    8.27       9.91 0        ****
##  4 arr_delay American ~ Unite~     3.19    2.40       3.99 4.50e-14 ****
##  5 arr_delay Delta Air~ Expre~    14.2    13.3       15.0  0        ****
##  6 arr_delay Delta Air~ JetBl~     7.81    7.06       8.56 0        ****
##  7 arr_delay Delta Air~ Unite~     1.91    1.19       2.64 5.61e-12 ****
##  8 arr_delay ExpressJe~ JetBl~    -6.34   -7.12      -5.55 0        ****
##  9 arr_delay ExpressJe~ Unite~   -12.2   -13.0      -11.5  0        ****
## 10 arr_delay JetBlue A~ Unite~    -5.90   -6.58      -5.22 0        ****
```

The F statistic is extremely high, suggesting there is definitely a difference between means. We can reject our null. Since we are using Welch Anova, we can use the Games-Howell test in place of the Tukey test to observe the differences between each airline.

The results from the Games-Howell test show us that there is a stark difference between the means of each airline (0 does not exist within any confidence interval and the p-values are very low). So to answer our question: yes, some airlines have more delays than others.

2: Does arrival delay vary between different engine types? H_0: Arrival delay does not vary between different engine types.

H_A: Arrival delay does vary between different engine types.

We can use ANOVA to compare different engine types. However, we first need to test the assumptions.

```
flights_engines = flights %>%
  filter(!is.na(arr_delay)) %>%
  left_join(planes, by = "tailnum") %>%
  filter(!is.na(engine))

table(flights_engines$engine)
```

```
##
##       4 Cycle Reciprocating     Turbo-fan     Turbo-jet    Turbo-prop
##                          47          1703        236084         40736            46
##   Turbo-shaft
##                         401
```

```
# Testing assumptions before ANOVA
print("Levene Test for Equal Variances: ")
```

```
## [1] "Levene Test for Equal Variances: "
```

```
leveneTest(arr_delay ~ engine, data = flights_engines)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      5  19.879 < 2.2e-16 ***
##       279011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Cannot use Shapiro-Wilk to test for normality due to large sample
# using ad test instead
ad.test(flights_engines$arr_delay)
```

```
##
##  Anderson-Darling normality test
##
## data:  flights_engines$arr_delay
## A = 20772, p-value < 2.2e-16
```

By testing the assumptions of equal variances and normality, we can see that both assumptions are violated. However, due to the large sample size, we can bypass the normality assumption. We can therefore use the Welch ANOVA test to account for the unequal variances.

```
oneway.test(arr_delay ~ engine, data = flights_engines, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  arr_delay and engine
## F = 77.48, num df = 5.00, denom df = 248.04, p-value < 2.2e-16
```

```
flights_engines %>%
  games_howell_test(arr_delay ~ engine)
```

```
## # A tibble: 15 x 8
##    .y.       group1        group2 estimate conf.low conf.high p.adj p.adj.signif
##  * <chr>     <chr>         <chr>     <dbl>    <dbl>     <dbl> <dbl> <chr>
##  1 arr_delay 4 Cycle       Recip~   -4.01   -20.8      12.8   0.98  ns
##  2 arr_delay 4 Cycle       Turbo~   -2.00   -18.6      14.6   0.999 ns
##  3 arr_delay 4 Cycle       Turbo~   -6.53   -23.1      10.1   0.849 ns
##  4 arr_delay 4 Cycle       Turbo~   -4.83   -30.2      20.5   0.994 ns
##  5 arr_delay 4 Cycle       Turbo~   -0.442  -18.3      17.4   1     ns
##  6 arr_delay Reciprocating Turbo~    2.01    -0.935     4.95  0.374 ns
##  7 arr_delay Reciprocating Turbo~   -2.52    -5.51      0.467 0.154 ns
##  8 arr_delay Reciprocating Turbo~   -0.824  -20.9      19.2   1     ns
##  9 arr_delay Reciprocating Turbo~    3.57    -3.87     11.0   0.744 ns
## 10 arr_delay Turbo-fan     Turbo~   -4.53    -5.18     -3.88  0     ****
## 11 arr_delay Turbo-fan     Turbo~   -2.83   -22.7      17.0   0.998 ns
## 12 arr_delay Turbo-fan     Turbo~    1.56    -5.28      8.40  0.987 ns
## 13 arr_delay Turbo-jet     Turbo~    1.70   -18.2      21.6   1     ns
## 14 arr_delay Turbo-jet     Turbo~    6.09    -0.774    13.0   0.115 ns
## 15 arr_delay Turbo-prop    Turbo~    4.39   -16.5      25.3   0.989 ns
```

The results from the Games-Howell test show us that most of the different engine types do not have a different mean arrival delay. However, there is a difference between turbo-fan vs turbo-jet engines. This suggests that for most engine types, arrival delay does not vary significantly between engines. We can check the most common type of engine for each airlines.

```
flights_engines_named = flights_engines %>%
  left_join(airlines, by = "carrier")

# Find most common engine type per airline
most_common_engines = flights_engines_named %>%
  group_by(name, engine) %>%
  summarise(count = n(), .groups = "drop") %>%
  arrange(name, desc(count)) %>%
  group_by(name) %>%
  slice(1)

most_common_engines
```

```
## # A tibble: 16 x 3
## # Groups:   name [16]
##    name                        engine    count
##    <chr>                       <chr>     <int>
##  1 AirTran Airways Corporation Turbo-fan  2958
##  2 Alaska Airlines Inc.        Turbo-fan   675
```

```
##  3 American Airlines Inc.      Turbo-fan  8930
##  4 Delta Air Lines Inc.        Turbo-fan 34916
##  5 Endeavor Air Inc.           Turbo-fan 17294
##  6 Envoy Air                   Turbo-jet   555
##  7 ExpressJet Airlines Inc.    Turbo-fan 50846
##  8 Frontier Airlines Inc.      Turbo-fan   634
##  9 Hawaiian Airlines Inc.      Turbo-fan   342
## 10 JetBlue Airways             Turbo-fan 52407
## 11 Mesa Airlines Inc.          Turbo-fan   544
## 12 SkyWest Airlines Inc.       Turbo-fan    29
## 13 Southwest Airlines Co.      Turbo-fan 11950
## 14 US Airways Inc.             Turbo-fan 17883
## 15 United Air Lines Inc.       Turbo-fan 31560
## 16 Virgin America              Turbo-fan  5116
```

It looks like all the airlines mostly use Turbo-fan engines, which means we don't have much evidence to connect different engine types with the arrival delays of different airlines. We fail to reject the null hypothesis.

3: Does cancellation rate vary across airlines? H_0: Cancellation rate is the same across all airlines.

H_A: Cancellation rate differs between at least some airlines.

We can use a chi-squared test to check our hypotheses.

```
flights_cancel = flights %>%
  mutate(cancelled = is.na(dep_time)) %>%
  left_join(airlines, by = "carrier")

cancel_table = table(flights_cancel$name, flights_cancel$cancelled)
cancel_table
```

```
##
##                               FALSE  TRUE
##   AirTran Airways Corporation  3187    73
##   Alaska Airlines Inc.          712     2
##   American Airlines Inc.      32093   636
##   Delta Air Lines Inc.        47761   349
##   Endeavor Air Inc.           17416  1044
##   Envoy Air                   25163  1234
##   ExpressJet Airlines Inc.    51356  2817
##   Frontier Airlines Inc.        682     3
##   Hawaiian Airlines Inc.        342     0
##   JetBlue Airways             54169   466
##   Mesa Airlines Inc.            545    56
##   SkyWest Airlines Inc.          29     3
##   Southwest Airlines Co.      12083   192
##   United Air Lines Inc.       57979   686
##   US Airways Inc.             19873   663
##   Virgin America               5131    31
```

```
chisq_test_result = chisq.test(cancel_table)
```

```
## Warning in chisq.test(cancel_table): Chi-squared approximation may be incorrect
```

```
chisq_test_result
```

```
##
##  Pearson's Chi-squared test
##
## data:  cancel_table
## X-squared = 4997.8, df = 15, p-value < 2.2e-16
```

Since the p-value is low, we can say that there is evidence that cancellation rate does vary across different airlines and we can reject the null hypothesis.

4: Does speed vary across airlines? H_0: Mean speed is the same across all airlines.

H_A: At least one airline has a different mean speed from the others.

We can test this using ANOVA, first checking assumptions of normality and equal variances.

```
flights_speed = flights %>%
  filter(!is.na(air_time), air_time > 0, !is.na(distance)) %>%
  mutate(speed = distance / (air_time / 60)) %>%
  left_join(airlines, by = "carrier")

# Testing equal variances with Levene's
leveneTest(speed ~ name, data = flights_speed)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group     15  1479.5 < 2.2e-16 ***
##       327330
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Testing for normality with ad test
ad.test(flights_speed$speed)
```

```
##
##  Anderson-Darling normality test
##
## data:  flights_speed$speed
## A = 2978.2, p-value < 2.2e-16
```

Like the previous questions, we can bypass normality and use a Welch ANOVA test for unequal variances.

```
oneway.test(speed ~ name, data = flights_speed)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  speed and name
## F = 7733.1, num df = 15.0, denom df = 1931.8, p-value < 2.2e-16
```

```
flights_speed %>%
  games_howell_test(speed ~ name)
```

```
## # A tibble: 120 x 8
##    .y.   group1         group2 estimate conf.low conf.high    p.adj p.adj.signif
##  * <chr> <chr>          <chr>     <dbl>    <dbl>     <dbl>    <dbl> <chr>
##  1 speed AirTran Airwa~ Alask~    49.3     45.8      52.8  0        ****
##  2 speed AirTran Airwa~ Ameri~    23.1     20.8      25.4  4.36e- 8 ****
##  3 speed AirTran Airwa~ Delta~    24.1     21.9      26.3  1.93e- 8 ****
##  4 speed AirTran Airwa~ Endea~   -48.9    -51.6     -46.2  2.69e-11 ****
##  5 speed AirTran Airwa~ Envoy~   -26.0    -28.3     -23.6  0        ****
##  6 speed AirTran Airwa~ Expre~   -31.4    -33.6     -29.2  3.38e- 8 ****
##  7 speed AirTran Airwa~ Front~    30.8     26.6      35.0  0        ****
##  8 speed AirTran Airwa~ Hawai~    86.0     82.4      89.6  0        ****
##  9 speed AirTran Airwa~ JetBl~     5.61     3.32      7.90 3.91e- 8 ****
## 10 speed AirTran Airwa~ Mesa ~   -62.4    -71.9     -52.9  4.53e-10 ****
## # i 110 more rows
```

The results show us that there is a drastic evidence to show a difference between speed for every airline. All the p-values are below 0.05, so we can reject the null hypothesis and say that speed does vary across different airlines. We can further see how speed interacts with flight delays by testing the correlation.

First, we can see how speed affects delays by each airline.

```
speed_delay_summary = flights_speed %>%
  filter(!is.na(arr_delay)) %>%
  group_by(name) %>%
  summarise(
    avg_speed = mean(speed, na.rm = TRUE),
    avg_arr_delay = mean(arr_delay, na.rm = TRUE)
  )

speed_delay_summary
```

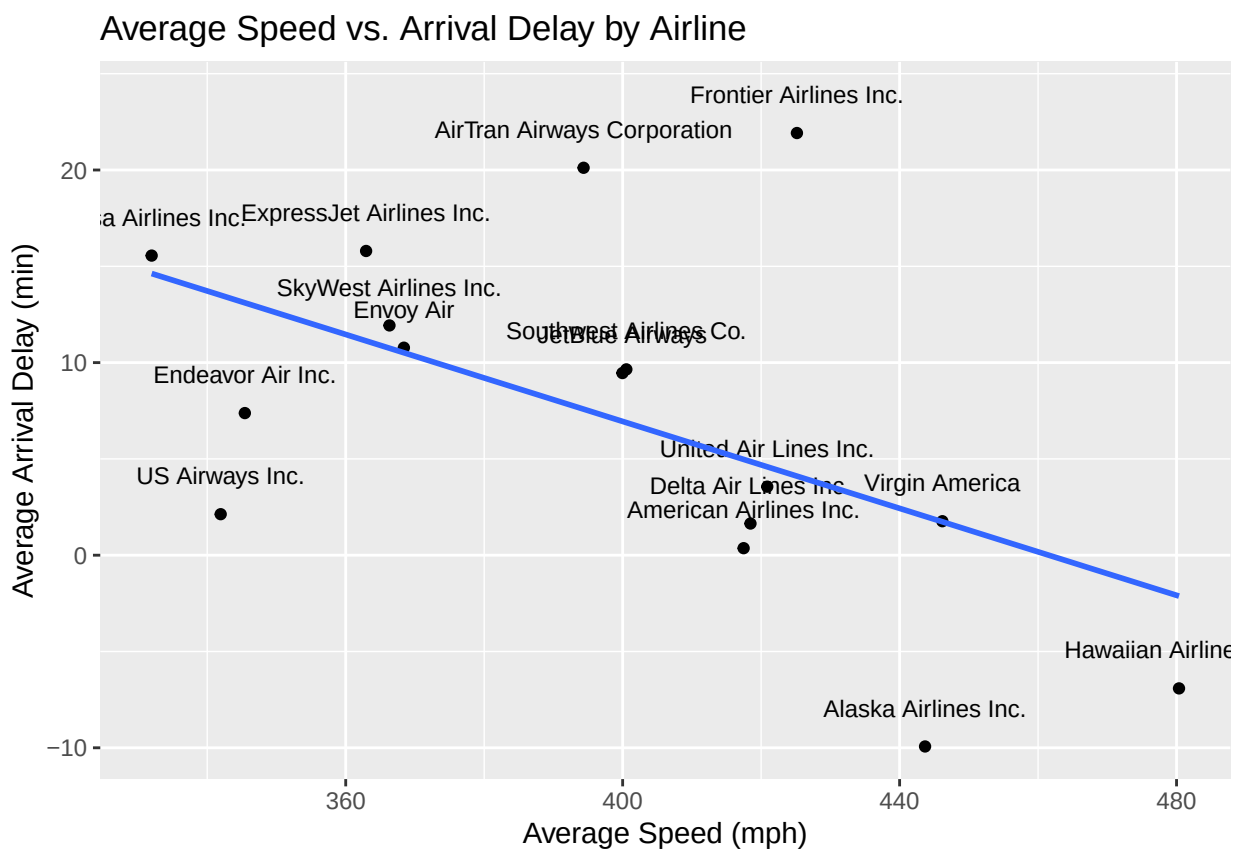
```
## # A tibble: 16 x 3
##    name                          avg_speed avg_arr_delay
##    <chr>                             <dbl>         <dbl>
##  1 AirTran Airways Corporation        394.        20.1
##  2 Alaska Airlines Inc.               444.        -9.93
##  3 American Airlines Inc.             417.         0.364
##  4 Delta Air Lines Inc.               418.         1.64
##  5 Endeavor Air Inc.                  345.         7.38
##  6 Envoy Air                          368.        10.8
##  7 ExpressJet Airlines Inc.           363.        15.8
##  8 Frontier Airlines Inc.             425.        21.9
##  9 Hawaiian Airlines Inc.             480.        -6.92
## 10 JetBlue Airways                    400.         9.46
## 11 Mesa Airlines Inc.                 332.        15.6
## 12 SkyWest Airlines Inc.              366.        11.9
## 13 Southwest Airlines Co.             401.         9.65
## 14 US Airways Inc.                    342.         2.13
## 15 United Air Lines Inc.              421.         3.56
## 16 Virgin America                     446.         1.76
```

```r
cor.test(speed_delay_summary$avg_speed, speed_delay_summary$avg_arr_delay)
```

```
##
##  Pearson's product-moment correlation
##
## data:  speed_delay_summary$avg_speed and speed_delay_summary$avg_arr_delay
## t = -2.3331, df = 14, p-value = 0.03507
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.81187916 -0.04529422
## sample estimates:
##       cor
## -0.5291194
```

```r
ggplot(speed_delay_summary, aes(x = avg_speed, y = avg_arr_delay, label = name)) +
  geom_point() +
  geom_text(nudge_y = 2, size = 3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Average Speed vs. Arrival Delay by Airline",
       x = "Average Speed (mph)",
       y = "Average Arrival Delay (min)")
```

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



This correlation confidence interval does not include 0 but it does come close, and it shows us that the correlation between speed and delay by airline is about -0.529. This suggests that airlines with more speed have less delay. Note that there are some outliers, since we only have a few different airlines to observe.

To provide some insight into our overall findings, we found that: (1) Different airlines have different delays on average, (2) Engine type does not seem to vary across airlines, and it doesn't seem to have a significant impact on delays, (3) Average cancellation rates vary between airlines, and (4) Airlines with faster mean speeds tend to experience lower arrival delays on average.

Question 3. Are delays more frequent during major holidays?

H_0: There is no difference in the distribution of arrival delays during major holidays compared to other days.

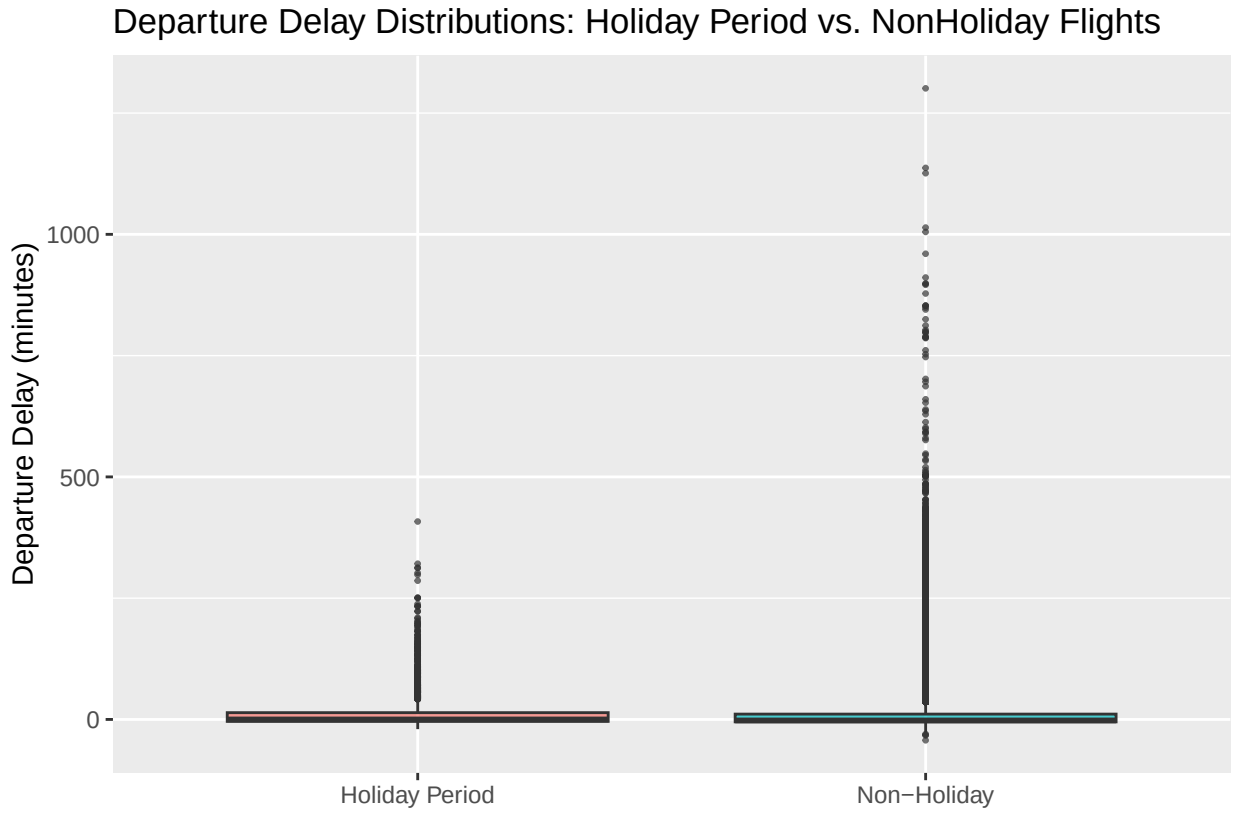
H_1: Arrival delays are different during major holidays compared to other days.

```
#identifying holiday periods
holidayDates<- as.Date(c(
  "2013-11-27",  #day before thanksgiving
  "2013-11-28",  #thanksgiving 2013
  "2013-12-24",  #day before christmas
  "2013-12-25",  #christmas 2013
  "2014-12-31",  #new years eve
  "2014-01-01"   #new years 2014
))

flights_holiday<-flights|>
  mutate(holiday_flag=ifelse(as.Date(time_hour) %in% holidayDates, "Holiday Period", "Non-Holiday"))|>
  filter(!is.na(arr_delay))

table(flights_holiday$holiday_flag)
```

```
##
## Holiday Period    Non-Holiday
##           3330         324016
```

Departure Delays

```
ggplot(flights_holiday, aes(x=holiday_flag, y=dep_delay, fill= holiday_flag))+
  geom_boxplot(outlier.size = 0.5, alpha=0.7)+
  labs(title= "Departure Delay Distributions: Holiday Period vs. NonHoliday Flights",
       x="", y="Departure Delay (minutes)")+
  theme(legend.position = "none")
```




```
set.seed(707)
normality_test<-flights_holiday|> #normality test per group
  group_by(holiday_flag)|>
  summarise(sample_delays= list(sample(dep_delay, min(5000,n()), replace = FALSE)),
            shapiro_p = shapiro.test(unlist(sample_delays))$p.value)|>
  ungroup()|>
  dplyr::select(holiday_flag, shapiro_p)
print(normality_test)
```

```
## # A tibble: 2 x 2
##   holiday_flag   shapiro_p
##   <chr>              <dbl>
## 1 Holiday Period  2.10e-67
## 2 Non-Holiday     3.38e-78
```

```
flights_holiday$holiday_flag<-as.factor(flights_holiday$holiday_flag)

leveneTest(dep_delay~holiday_flag, data=flights_holiday)#levenes test for homogeneity of variance
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value Pr(>F)
## group      1   0.608 0.4355
##       327344
```

```r
holiday_test<-wilcox.test(dep_delay~holiday_flag, data=flights_holiday)#Wilcoxon Rank Sum Test
print(holiday_test)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  dep_delay by holiday_flag
## W = 583916914, p-value = 2.42e-16
## alternative hypothesis: true location shift is not equal to 0
```

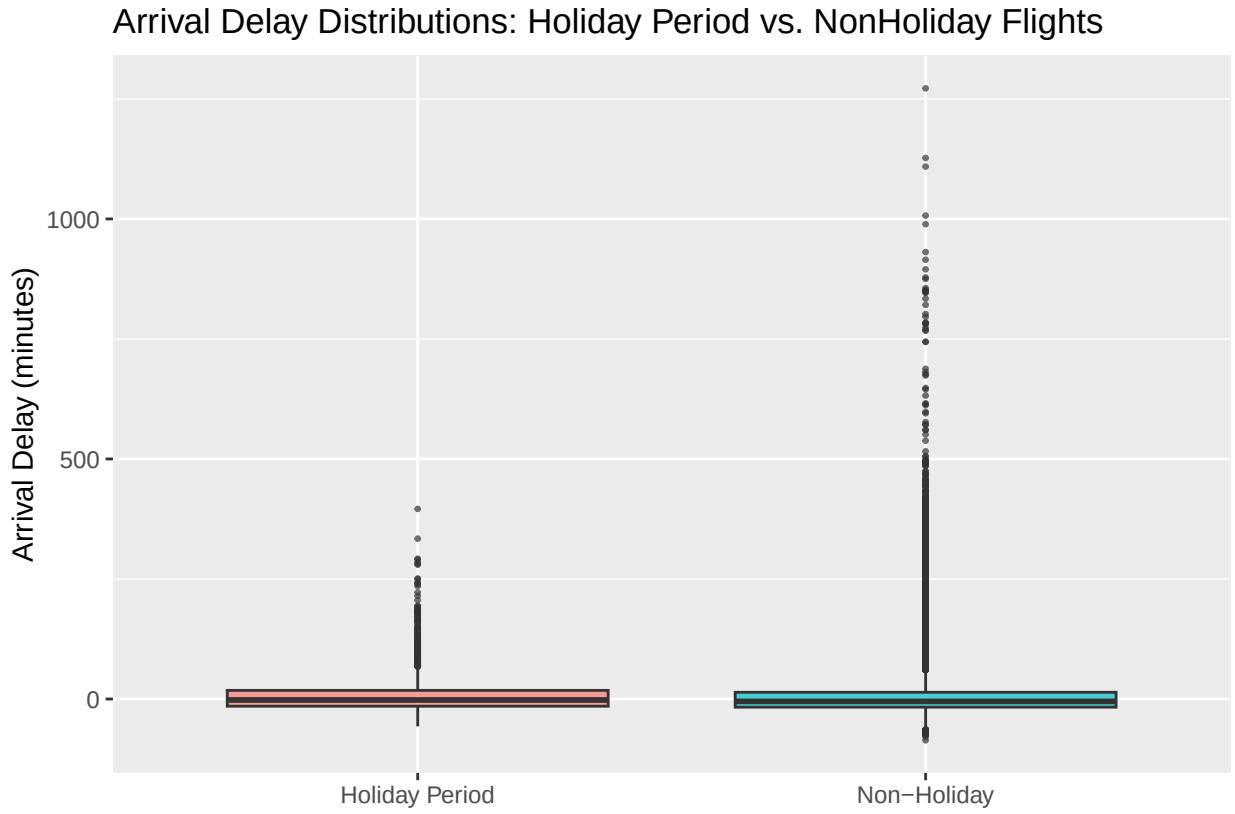
```r
cat("Wilcoxon test p-value: ", signif(holiday_test$p.value,4))
```

```
## Wilcoxon test p-value:  2.42e-16
```

Departure Conclusion: Departure delays has a low p-value for both holiday and non holiday periods which means that the distributions are not normal in either group, this leads us to use a non-parametric test such as the Levene test and Wilcoxon rank sum test. For departure delays, the Levene's test p=0.4355 which is not significant meaning that variances are roughly equal between holiday and non holiday groups for departure delays. The Wilcoxon Rank Sum test p-value is approximately 7.33×10^{-10} which is significant and allows us to conclude that departure delays are significant between holiday and non holiday periods.

#Arrival Delays

```r
ggplot(flights_holiday, aes(x=holiday_flag, y=arr_delay, fill= holiday_flag))+
  geom_boxplot(outlier.size = 0.5, alpha=0.7)+
  labs(title= "Arrival Delay Distributions: Holiday Period vs. NonHoliday Flights",
       x="", y="Arrival Delay (minutes)")+
  theme(legend.position = "none")
```




```
set.seed(707)
normality_test<-flights_holiday|> #normality test per group
  group_by(holiday_flag)|>
  summarise(sample_delays= list(sample(arr_delay, min(5000,n()), replace = FALSE)),
            shapiro_p = shapiro.test(unlist(sample_delays))$p.value)|>
  ungroup()|>
  dplyr::select(holiday_flag, shapiro_p)
print(normality_test)
```

```
## # A tibble: 2 x 2
##   holiday_flag   shapiro_p
##   <fct>              <dbl>
## 1 Holiday Period  9.49e-57
## 2 Non-Holiday     1.14e-67
```

```
flights_holiday$holiday_flag<-as.factor(flights_holiday$holiday_flag)

leveneTest(arr_delay~holiday_flag, data=flights_holiday)#levenes test for homogeneity of variance
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value  Pr(>F)
## group      1  3.9166 0.04781 *
##       327344
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
holiday_test<-wilcox.test(arr_delay~holiday_flag, data=flights_holiday)#Wilcoxon Rank Sum Test
print(holiday_test)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  arr_delay by holiday_flag
## W = 572896209, p-value = 7.329e-10
## alternative hypothesis: true location shift is not equal to 0
```

```r
cat("Wilcoxon test p-value: ", signif(holiday_test$p.value,4))
```

```
## Wilcoxon test p-value:  7.329e-10
```

Arrival Conclusion: Arrival delays has a low p-value for both holiday and non holiday periods which means that the distributions are not normal in either group, this leads us to use a non-parametric test such as the Levene test and Wilcoxon rank sum test. For arrival delays, the Levene's test p=0.04781 which indicates that the variance differs significantly between holiday and non holiday groups for arrival delays. The Wilcoxon Rank Sum test p-value is ~7.33e-10 which is significant and allows us to conclude that departure delays are significant between holiday and non holiday periods.

Overall Conclusion: We can conclude that Departure and Arrival flight delays occur more frequently or are more severe during major holiday periods, defined as the day before and day of the holiday. The analysis shows that arrival delays increase and exhibit significantly greater variability during holiday times, indicating inconsistent and unpredictable arrival times around major holidays. Although departure delays show similar mean increases during holidays, the variance remains stable suggesting that delays at departure are consistently worse but not erratic. This analysis highlights the operational challenges airlines and airports face during major holidays, emphasizing the need for planning and resource allocations to mitigate such delays during holiday periods.

Question 4: Does the age of the plane affect flight delays?

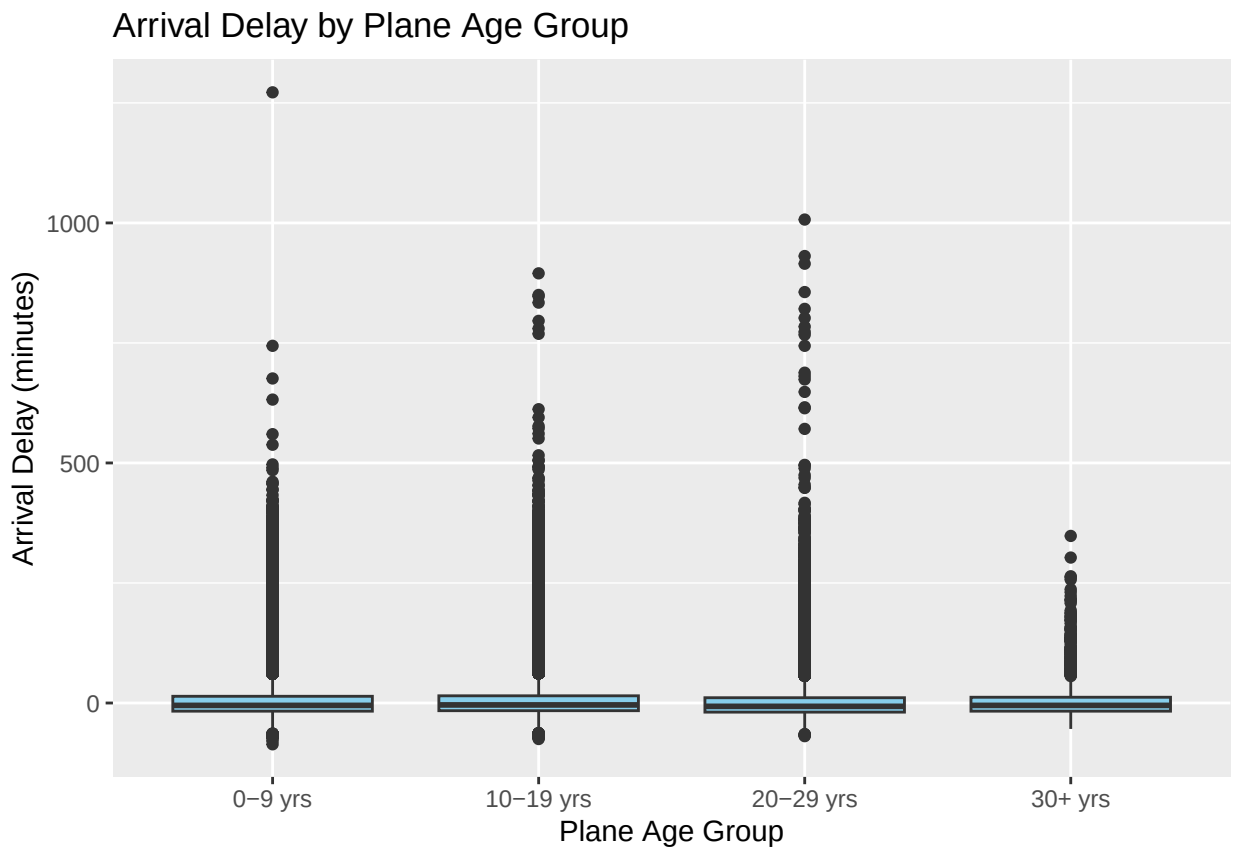
Within this main question we will perform hypothesis tests to answer the two following sub-questions:

1. Do older planes experience more delays compared to newer ones? H_0: There is no difference in the distribution of arrival delays across the different plane age groups.

H_1: At least one plane age group has a different distribution of arrival delays compared to the others.

```
# Join flights with planes to get plane manufacture year
planes_fixed <- planes %>%
  rename(plane_year = year)

flights_planes <- flights %>%
  inner_join(planes %>% rename(plane_year = year), by = "tailnum") %>%
  filter(!is.na(plane_year), !is.na(arr_delay)) %>%
  mutate(
    plane_age = 2013 - plane_year,
    age_group = case_when(
      plane_age < 10 ~ "0-9 yrs",
      plane_age < 20 ~ "10-19 yrs",
      plane_age < 30 ~ "20-29 yrs",
      TRUE ~ "30+ yrs"
    )
  )

head(flights_planes)
```

```
## # A tibble: 6 x 29
##    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
## # i 21 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>, plane_year <int>, type <chr>,
## #   manufacturer <chr>, model <chr>, engines <int>, seats <int>, speed <int>,
## #   engine <chr>, plane_age <dbl>, age_group <chr>
```



The boxplot shows that arrival delays are fairly similar across all plane age groups, with comparable medians and interquartile ranges. While each group has extreme outliers, there is no clear visual trend suggesting that older planes experience more delays than newer ones. We will run a hypothesis test to further our findings.

```
flights_planes$age_group <- as.factor(flights_planes$age_group)

summary_stats <- flights_planes %>%
  group_by(age_group) %>%
  summarise(
    mean_delay = mean(arr_delay, na.rm = TRUE),
    count = n()
  )
print(summary_stats)
```

```
## # A tibble: 4 x 3
##   age_group mean_delay  count
##   <fct>          <dbl>  <int>
## 1 0-9 yrs         7.36 103366
## 2 10-19 yrs       7.61 133479
## 3 20-29 yrs       4.00  35412
## 4 30+ yrs         5.54   1596
```

```
# Normality test with sample per group
set.seed(123)
normality_test <- flights_planes %>%
```

```
  group_by(age_group) %>%
  summarise(
    sample_delays = list(sample(arr_delay[!is.na(arr_delay)], min(5000, n()), replace = FALSE)),
    shapiro_p = shapiro.test(unlist(sample_delays))$p.value
  ) %>%
dplyr::select(-sample_delays)
print(normality_test)
```

```
## # A tibble: 4 x 2
##   age_group shapiro_p
##   <fct>         <dbl>
## 1 0-9 yrs    5.81e-68
## 2 10-19 yrs  1.49e-69
## 3 20-29 yrs  4.87e-75
## 4 30+ yrs    2.11e-45
```

```
# Levene's Test for homogeneity of variances
leveneTest(arr_delay ~ age_group, data = flights_planes)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value  Pr(>F)
## group      3  3.4683 0.01542 *
##       273849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Since homogeneity is violated, run Kruskal-Wallis test (non-parametric)
kruskal.test(arr_delay ~ age_group, data = flights_planes)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  arr_delay by age_group
## Kruskal-Wallis chi-squared = 514.57, df = 3, p-value < 2.2e-16
```

```
# post-hoc test for Kruskal-Wallis (Dunn test) if significant
dunn.test(flights_planes$arr_delay, flights_planes$age_group, method = "bonferroni")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 514.566, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
## Col Mean-|
## Row Mean |    0-9 yrs   10-19 yr   20-29 yr
## ---------+---------------------------------
## 10-19 yr |  -2.397838
##          |     0.0495
```

```
##           |
## 20-29 yr |   19.95894   22.22150
##           |    0.0000*    0.0000*
##           |
##  30+ yrs |   0.872657   1.268699  -3.942415
##           |     1.0000     0.6136    0.0002*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

The Kruskal-Wallis test revealed a highly significant difference in arrival delays across plane age groups (p < 2.2e-16), indicating that at least one group's delay distribution differs from the others. Post-hoc pairwise comparisons using Dunn's test with Bonferroni correction showed that planes aged 20–29 years experience significantly different delay patterns compared to both the 0–9 and 10–19 year groups (adjusted p-values < 0.001). Additionally, planes aged 30+ years differ significantly from the 20–29 year group (adjusted p = 0.0002), but do not differ significantly from the younger 0–9 or 10–19 year groups. The difference between the 10–19 and 0–9 year groups was borderline significant (adjusted p = 0.0495). In summary, planes aged 20–29 years tend to have notably different arrival delays compared to most other age groups, highlighting a possible link between this age range and on-time performance issues.

2. Are there specific plane models or manufactures associated with better on-time performance? H_0: There is no difference in the distribution of arrival delays across different plane models or manufacturers.

H_1: At least one plane model or manufacturer has a different distribution of arrival delays compared to the others.

```
flights_manufacturer <- flights %>%
  inner_join(planes, by = "tailnum") %>%
  filter(!is.na(manufacturer), !is.na(arr_delay)) %>%
  group_by(manufacturer) %>%
  filter(n() > 50)
```


```r
# filter manufacturers with more than 50 flights
flights_manufacturer <- flights %>%
  inner_join(planes, by = "tailnum") %>%
  filter(!is.na(manufacturer), !is.na(arr_delay)) %>%
  group_by(manufacturer) %>%
  filter(n() > 50) %>%
  ungroup()

flights_manufacturer$manufacturer <- as.factor(flights_manufacturer$manufacturer)

# Normality test per manufacturer group
set.seed(123)
normality_test <- flights_manufacturer %>%
  group_by(manufacturer) %>%
  summarise(
    sample_delays = list(sample(arr_delay, min(5000, n()), replace = FALSE)),
    shapiro_p = shapiro.test(unlist(sample_delays))$p.value
  ) %>%
dplyr::select(-sample_delays)
print(normality_test)
```

```
## # A tibble: 22 x 2
##    manufacturer       shapiro_p
##    <fct>                  <dbl>
##  1 AIRBUS              1.03e-66
```

```
##  2 AIRBUS INDUSTRIE    1.17e-71
##  3 BARKER JACK L       1.20e-15
##  4 BELL                2.94e-11
##  5 BOEING              4.68e-67
##  6 BOMBARDIER INC      8.57e-69
##  7 CANADAIR            2.56e-47
##  8 CANADAIR LTD        1.16e-11
##  9 CESSNA              1.67e-30
## 10 CIRRUS DESIGN CORP  1.70e-22
## # i 12 more rows
```

```r
# Levene's Test for homogeneity of variance
leveneTest(arr_delay ~ manufacturer, data = flights_manufacturer)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group     21   43.17 < 2.2e-16 ***
##       278672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Kruskal-Wallis test
kruskal_result <- kruskal.test(arr_delay ~ manufacturer, data = flights_manufacturer)
print(kruskal_result)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  arr_delay by manufacturer
## Kruskal-Wallis chi-squared = 3770.9, df = 21, p-value < 2.2e-16
```

```r
if (kruskal_result$p.value < 0.05) {
  dunn_output <- capture.output(
    dunn_result <- dunn.test(
      flights_manufacturer$arr_delay,
      flights_manufacturer$manufacturer,
      kw = FALSE,
      list = TRUE,
      rmc = FALSE,
      alpha = 0.05
    )
  )

  sig_comparisons <- data.frame(
    comparison = dunn_result$comparisons,
    p_value = dunn_result$P.adjusted
  ) %>%
    filter(p_value < 0.05) %>%
    arrange(p_value) %>%
    slice_head(n = 5) %>%  # Show only top 5
    mutate(p_value = signif(p_value, 4))  # Optional: round p-values
```

```r
  cat("Kruskal-Wallis p-value:", signif(kruskal_result$p.value, 4), "\n")
  cat("Significant differences found. Top 5 manufacturer pairs with different arrival delays:\n\n")
  print(sig_comparisons, row.names = FALSE)

} else {
  cat("Kruskal-Wallis p-value:", signif(kruskal_result$p.value, 4), "\n")
  cat("No significant differences found among manufacturers.\n")
}
```

```
## Kruskal-Wallis p-value: 0
## Significant differences found. Top 5 manufacturer pairs with different arrival delays:
##
##                   comparison    p_value
##   AIRBUS INDUSTRIE - EMBRAER  0.000e+00
##              BOEING - EMBRAER  0.000e+00
##     BOMBARDIER INC - EMBRAER  0.000e+00
##              AIRBUS - EMBRAER 1.240e-223
##  EMBRAER - MCDONNELL DOUGLAS 2.626e-188
```

The Kruskal-Wallis test indicates a highly significant difference in arrival delays across plane manufacturers ($\chi^2 = 3770.9$, df = 21, p < 2.2e-16). This strongly suggests that not all manufacturers have the same on-time performance. The post-hoc Dunn's test with Bonferroni correction highlights specific pairwise differences. Notably, comparisons involving Embraer show consistently significant differences with Airbus Industrie, Boeing, Bombardier Inc, and McDonnell Douglas, indicating that Embraer aircraft tend to have distinct arrival delay patterns compared to these manufacturers. Additionally, Airbus and McDonnell Douglas pairs also appear among the most significant differences. These results indicate that certain manufacturers, especially Embraer, are associated with notably different arrival delay profiles, reflecting differences in punctuality or operational factors across manufacturers.

Question 5. How do environmental factors like humidity, visibility, and wind affect flight delays?

H_0: Environmental factors like humidity, visibility, and wind do not affect flight delays across airports.

H_1: At least one of the environmental factors like humidity, visibility, and wind affect flight delays.

1. Is there a relationship between the predictors To address this question, a correlation matrix was utilized to explore the relationship between departure delays and various environmental factors. Initially, the dataset was filtered to exclude any missing values related to departure delay and air time. Following this, the flight data was merged with a weather dataset subset that included key environmental factors such as humidity, visibility, wind speed, air density, and wind gust.

To prepare for further analysis and cross-validation, the merged dataset, denoted as 'flights_weather', was grouped by airport origin. The airports included in the analysis were Newark Liberty Airport (EWR), La-Guardia Airport (LGA), and John F. Kennedy Airport (JFK). A sample of 1,000 observations was randomly selected from each airport group using the 'sample_n' function. This sampling process was conducted to create balanced and manageable traing data sets for modeling and cross-validation procedures.

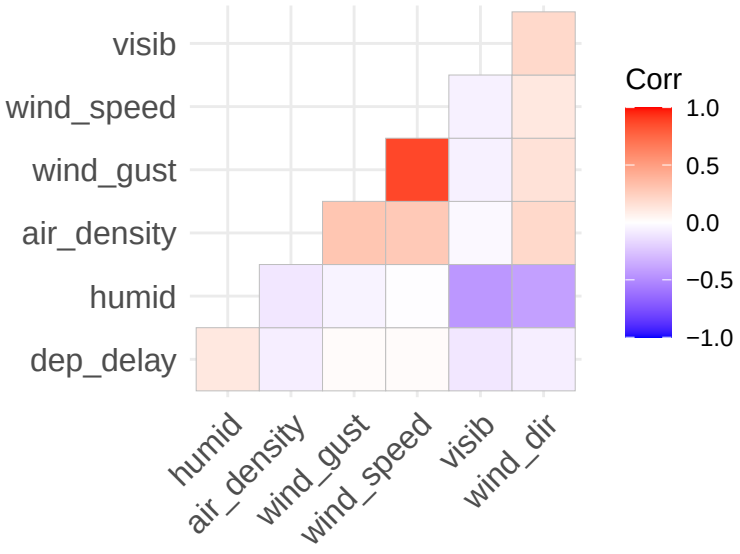
```
##           year          month            day       dep_time sched_dep_time
##              0              0              0              0              0
##      dep_delay       arr_time sched_arr_time      arr_delay        carrier
##              0              0              0              0              0
##         flight        tailnum         origin           dest       air_time
##              0              0              0              0              0
##       distance           hour         minute      time_hour
##              0              0              0              0
```

```
## tibble [327,346 x 19] (S3: tbl_df/tbl/data.frame)
##  $ year          : int [1:327346] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int [1:327346] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int [1:327346] 1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int [1:327346] 517 533 542 544 554 554 555 557 557 558 ...
##  $ sched_dep_time: int [1:327346] 515 529 540 545 600 558 600 600 600 600 ...
##  $ dep_delay     : num [1:327346] 2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int [1:327346] 830 850 923 1004 812 740 913 709 838 753 ...
##  $ sched_arr_time: int [1:327346] 819 830 850 1022 837 728 854 723 846 745 ...
##  $ arr_delay     : num [1:327346] 11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr [1:327346] "UA" "UA" "AA" "B6" ...
##  $ flight        : int [1:327346] 1545 1714 1141 725 461 1696 507 5708 79 301 ...
##  $ tailnum       : chr [1:327346] "N14228" "N24211" "N619AA" "N804JB" ...
##  $ origin        : chr [1:327346] "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr [1:327346] "IAH" "IAH" "MIA" "BQN" ...
##  $ air_time      : num [1:327346] 227 227 160 183 116 150 158 53 140 138 ...
##  $ distance      : num [1:327346] 1400 1416 1089 1576 762 ...
##  $ hour          : num [1:327346] 5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num [1:327346] 15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct[1:327346], format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

1a. Correlation Analysis

```
##               dep_delay        humid       visib   wind_gust   wind_speed
## dep_delay    1.00000000  0.122166383 -0.09745755  0.02441787  0.021001914
```

```
## humid        0.12216638  1.000000000 -0.45001107 -0.05383642 -0.009770943
## visib       -0.09745755 -0.450011070  1.00000000 -0.06001602 -0.063961212
## wind_gust    0.02441787 -0.053836421 -0.06001602  1.00000000  0.874430293
## wind_speed   0.02100191 -0.009770943 -0.06396121  0.87443029  1.000000000
## wind_dir    -0.06842713 -0.411605970  0.20250473  0.14504334  0.116032873
## air_density -0.06533380 -0.100994602 -0.02646630  0.29750449  0.275989839
##                wind_dir air_density
## dep_delay   -0.06842713  -0.0653338
## humid       -0.41160597  -0.1009946
## visib        0.20250473  -0.0264663
## wind_gust    0.14504334   0.2975045
## wind_speed   0.11603287   0.2759898
## wind_dir     1.00000000   0.2012071
## air_density  0.20120708   1.0000000
```



The correlation matrix examined correlation between departure delays and environmental factors. The results show environmental factors like humidity, air density, visibility, and wind direction have weak to moderate correlation with flight delays. However, wind speed and wind gust show no correlation, thus these variables omitted from the models providing us with following equation to build our models: dep delay $= \beta_0 + \beta_1 \cdot \text{humidity} + \beta_2 \cdot \text{visibility} + \beta_3 \cdot \text{wind gust} + \beta_4 \cdot \text{wind speed} + \beta_5 \cdot \text{wind direction} + \beta_6 \cdot \text{air density} + \epsilon$.

2. Linear Regression Models Three linear models were built, and forward selection was employed for model selection, providing the predictive departure delay for for each airport model:

The model for Newark Liberty Airport ('lm.ewr.01') is given by: $\hat{y}_{\text{EWRdep delay}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{humid}$.

The model for LaGuardia Airport ('lm.lga.01') is given by: $\hat{y}_{\text{LGAdep delay}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{humid} + \hat{\beta}_2 \cdot \text{wind dir}$.

The model for John F. Kennedy Airport ('lm.jfk') is given by: $\hat{y}_{\text{JFKdep delay}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{visib} + \hat{\beta}_2 \cdot \text{wind dir}$.

Model Fit: Following the building of our models, residual plots were used to evaluate model fit of the models and identify issues. For the Residuals vs. Fitted Plots, slight curvatures are detected, suggesting some non-linearity in the relationship between the predictors and response variable. This indicates that the linear models may not fully capture underlying patterns in the data. For Normal Q-Q Plots, the residuals deviate from the line at the tails, highlighting some deviation from normality, this could suggest the presence of outliers or a skewed distribution of residuals. For Scale-Location Plots, the spread of the residuals increases as fitted values increase, this suggests some heteroscedasticity. This suggests that variability of residuals is not constant, which can affect the validity of the regression analysis. For the Residuals vs Leverage Plots, we can see a presence of influential observations, which can affect the model's coefficients and overall fit. All trends were similarly observered across all three models. To address these issues, summary statistics (R-squared and Mean squared error) were employed through a validation set to provide further insight into the models performance.

```
##
## Call:
## lm(formula = dep_delay ~ humid + air_density + visib + wind_dir,
##     data = delay_EWR)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -60.58 -27.95 -17.59  10.19 349.65
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.495e+01  3.295e+01   1.364    0.173
## humid        4.623e-01  1.174e-01   3.939 8.77e-05 ***
## air_density -1.784e+03  2.248e+03  -0.794    0.428
## visib       -1.282e+00  1.711e+00  -0.750    0.454
## wind_dir     6.920e-03  2.222e-02   0.311    0.755
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.69 on 995 degrees of freedom
## Multiple R-squared:  0.02733,    Adjusted R-squared:  0.02342
## F-statistic: 6.989 on 4 and 995 DF,  p-value: 1.505e-05


##
## Call:
## lm(formula = dep_delay ~ humid + air_density + visib + wind_dir,
##     data = delay_LGA)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -62.68 -31.50 -18.62   9.05 619.94
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.013e+02  3.853e+01   2.629 0.008704 **
## humid        6.401e-01  1.422e-01   4.502 7.53e-06 ***
## air_density -1.009e+04  2.738e+03  -3.685 0.000241 ***
## visib        9.158e-01  1.843e+00   0.497 0.619289
## wind_dir     3.619e-02  2.520e-02   1.436 0.151236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.88 on 995 degrees of freedom
## Multiple R-squared:  0.03822,    Adjusted R-squared:  0.03435
## F-statistic: 9.884 on 4 and 995 DF,  p-value: 7.624e-08
```

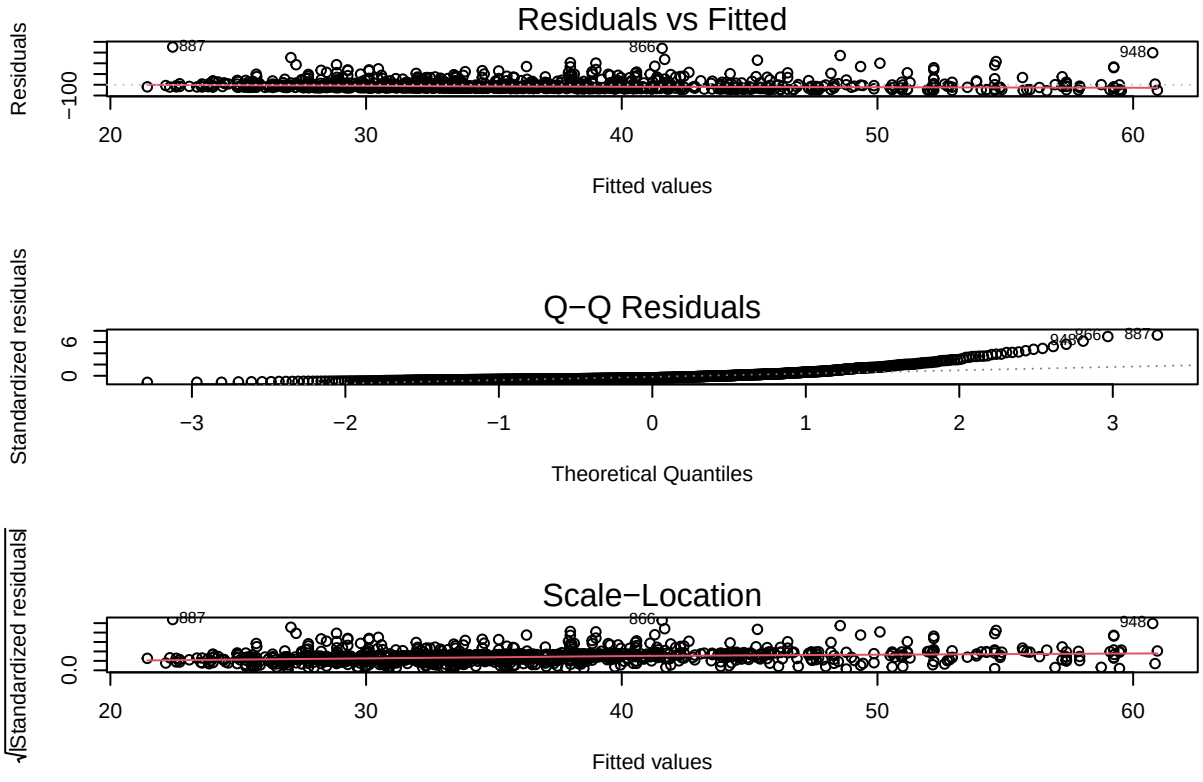
```
##
## Call:
## lm(formula = dep_delay ~ humid + air_density + visib + wind_dir,
##     data = delay_JFK)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -59.13 -27.94 -17.07  12.84 254.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.918e+01  3.095e+01   3.204  0.00140 **
## humid        1.115e-01  9.131e-02   1.221  0.22235
## air_density -3.392e+03  2.475e+03  -1.371  0.17080
## visib       -2.663e+00  8.655e-01  -3.077  0.00215 **
## wind_dir    -2.352e-02  1.819e-02  -1.293  0.19642
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.91 on 995 degrees of freedom
## Multiple R-squared:  0.03257,    Adjusted R-squared:  0.02868
## F-statistic: 8.375 on 4 and 995 DF,  p-value: 1.204e-06
```

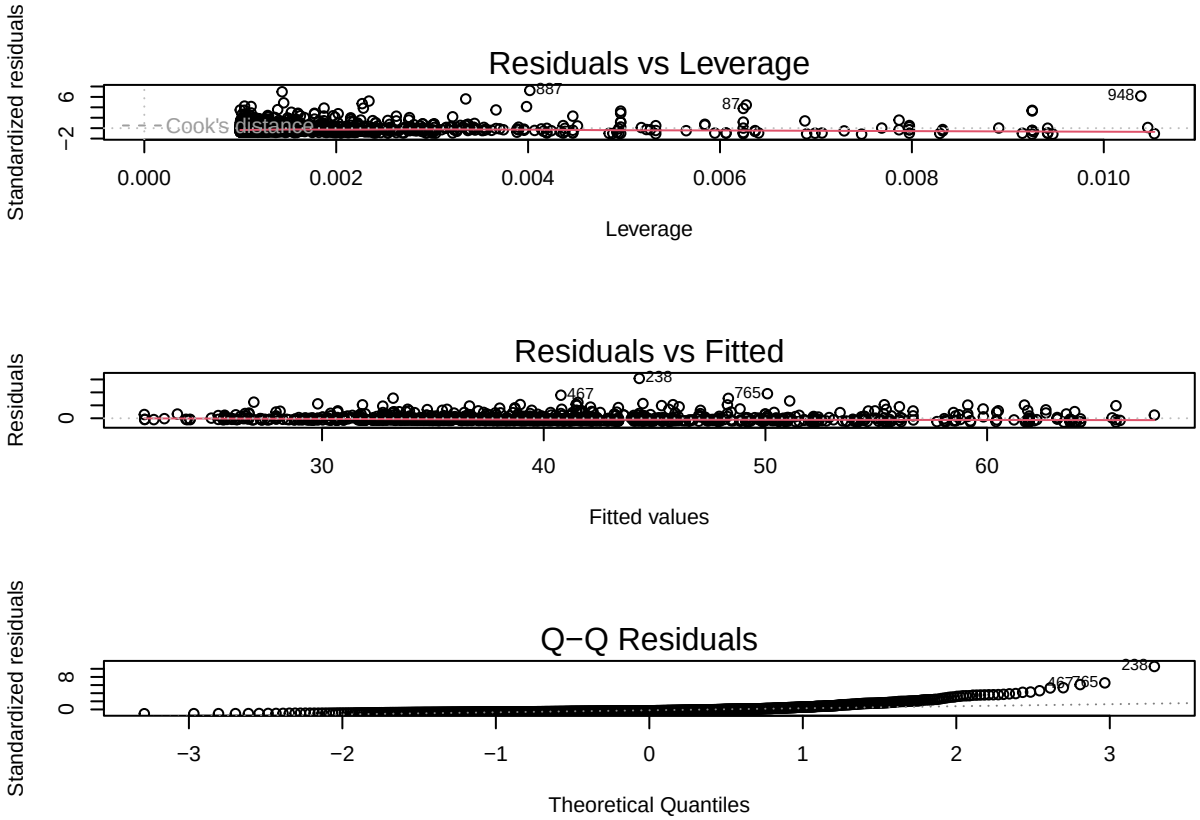
```
## Start:  AIC=7795.78
## dep_delay ~ 1
##
##               Df Sum of Sq     RSS    AIC
## + humid        1     63728 2361733 7771.2
## + visib        1     22817 2402643 7788.3
## + wind_dir     1     10156 2415304 7793.6
## <none>                     2425460 7795.8
## + air_density  1      2576 2422884 7796.7
##
## Step:  AIC=7771.15
## dep_delay ~ humid
##
##               Df Sum of Sq     RSS    AIC
## <none>                     2361733 7771.2
## + air_density  1   1059.77 2360673 7772.7
## + visib        1    990.11 2360742 7772.7
## + wind_dir     1     48.39 2361684 7773.1
```

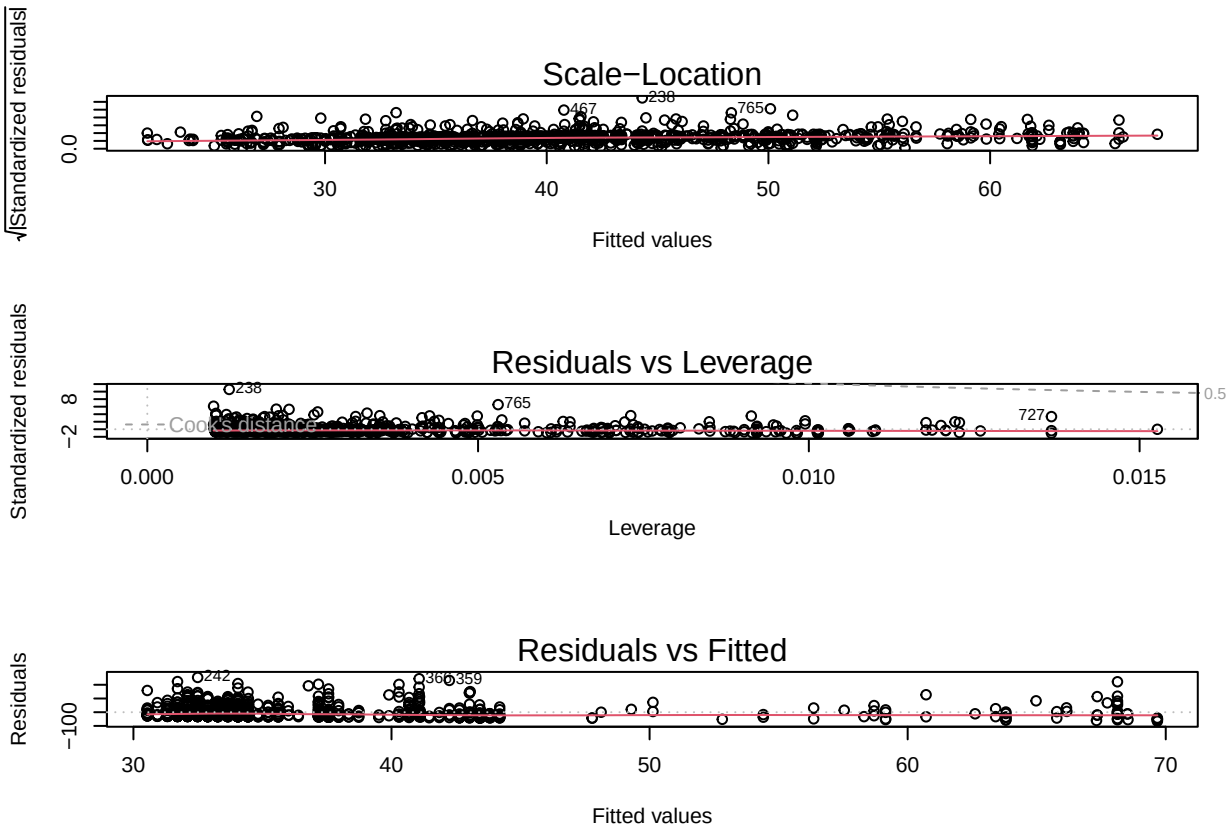
```
## Start:  AIC=8152.86
## dep_delay ~ 1
##
##               Df Sum of Sq     RSS    AIC
## + humid        1     83311 3383040 8130.5
## + air_density  1     55526 3410825 8138.7
## + wind_dir     1      9139 3457212 8152.2
## <none>                     3466350 8152.9
## + visib        1      6004 3460347 8153.1
##
## Step:  AIC=8130.53
## dep_delay ~ humid
##
##               Df Sum of Sq     RSS    AIC
## + air_density  1     41448 3341591 8120.2
## <none>                     3383040 8130.5
## + visib        1      3180 3379860 8131.6
## + wind_dir     1       364 3382676 8132.4
##
## Step:  AIC=8120.2
## dep_delay ~ humid + air_density
##
##            Df Sum of Sq     RSS    AIC
## + wind_dir  1    6886.8 3334705 8120.1
## <none>                  3341591 8120.2
## + visib     1     802.4 3340789 8122.0
##
## Step:  AIC=8120.14
## dep_delay ~ humid + air_density + wind_dir
##
##         Df Sum of Sq     RSS    AIC
## <none>               3334705 8120.1
## + visib  1    827.68 3333877 8121.9

## Start:  AIC=7639.4
## dep_delay ~ 1
##
##               Df Sum of Sq     RSS    AIC
## + visib        1     47763 2026588 7618.1
## + humid        1     42044 2032307 7620.9
## + wind_dir     1     24074 2050276 7629.7
## + air_density  1     10190 2064161 7636.5
## <none>                     2074351 7639.4
##
## Step:  AIC=7618.11
## dep_delay ~ visib
##
##               Df Sum of Sq     RSS    AIC
## + wind_dir     1   11643.7 2014944 7614.3
## + humid        1   11141.9 2015446 7614.6
## + air_density  1    9500.3 2017088 7615.4
## <none>                     2026588 7618.1
##
## Step:  AIC=7614.35
```

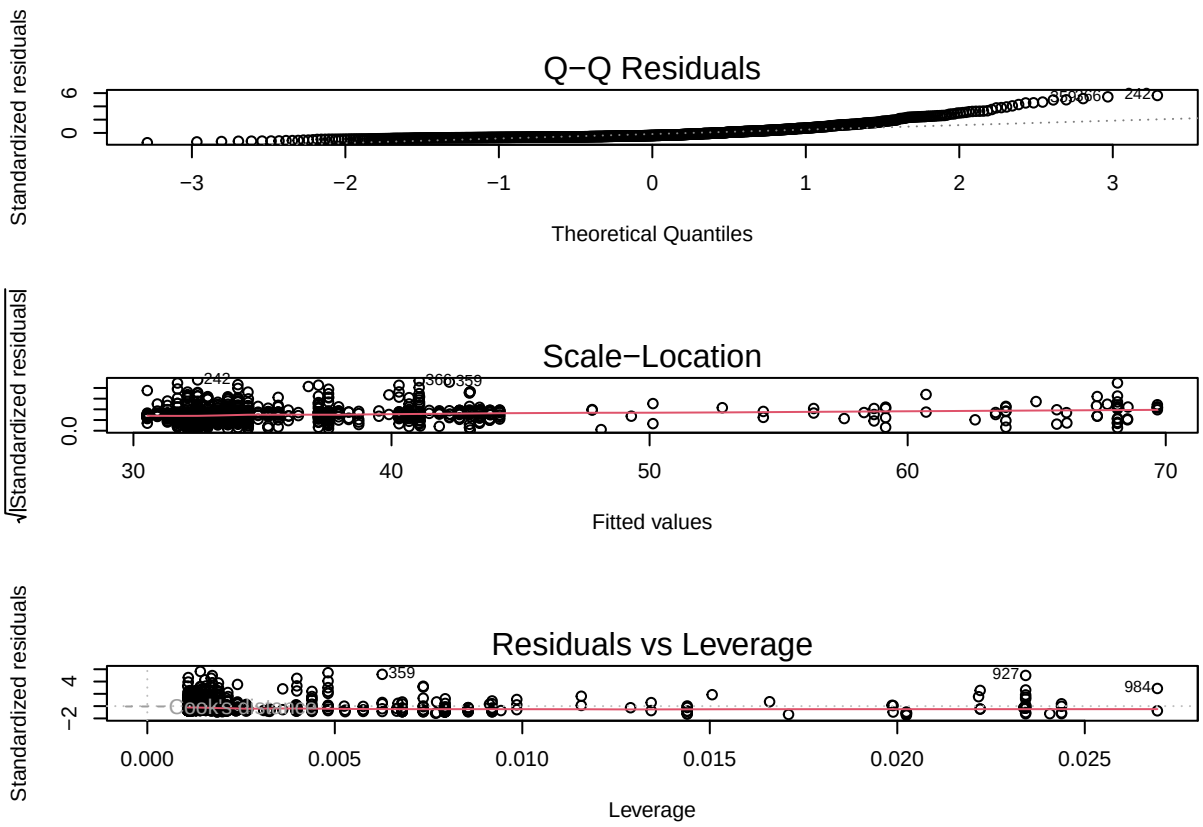
```
## dep_delay ~ visib + wind_dir
##
##               Df Sum of Sq     RSS    AIC
## + air_density  1    5153.9 2009791 7613.8
## + humid        1    4372.2 2010572 7614.2
## <none>                     2014944 7614.3
##
## Step:  AIC=7613.79
## dep_delay ~ visib + wind_dir + air_density
##
##         Df Sum of Sq     RSS    AIC
## <none>               2009791 7613.8
## + humid  1    3007.2 2006783 7614.3

##
## Call:
## lm(formula = dep_delay ~ humid, data = delay_EWR)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -56.24 -28.07 -17.25  10.17 351.56
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.44087    4.66697   2.880  0.00406 **
## humid        0.49434    0.09526   5.189 2.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.65 on 998 degrees of freedom
## Multiple R-squared:  0.02627,    Adjusted R-squared:  0.0253
## F-statistic: 26.93 on 1 and 998 DF,  p-value: 2.556e-07

##
## Call:
## lm(formula = dep_delay ~ humid + wind_dir, data = delay_LGA)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -60.88 -31.58 -19.30   7.64 615.69
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.581226  10.360302   0.925    0.355
## humid        0.605232   0.129128   4.687 3.16e-06 ***
## wind_dir     0.007937   0.024221   0.328    0.743
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.25 on 997 degrees of freedom
## Multiple R-squared:  0.02414,    Adjusted R-squared:  0.02218
## F-statistic: 12.33 on 2 and 997 DF,  p-value: 5.127e-06

##
```

```
## Call:
## lm(formula = dep_delay ~ visib + wind_dir, data = delay_JFK)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -65.68 -28.03 -17.49  12.66 253.51
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 75.93543    7.47569  10.158  < 2e-16 ***
## visib       -3.13529    0.74986  -4.181 3.15e-05 ***
## wind_dir    -0.03901    0.01625  -2.400   0.0166 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.96 on 997 degrees of freedom
## Multiple R-squared:  0.02864,    Adjusted R-squared:  0.02669
## F-statistic:  14.7 on 2 and 997 DF,  p-value: 5.121e-07
```




Residuals vs Leverage
Standardized residuals
Cook's distance
887
876
948
0.000
0.002
0.004
0.006
0.008
0.010
Leverage
Residuals vs Fitted
Residuals
238
467
765
30
40
50
60
Fitted values
Q−Q Residuals
Standardized residuals
238
467
765
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Scale−Location
√|Standardized residuals|
0.0
238
467
765
30
40
50
60
Fitted values
Residuals vs Leverage
Standardized residuals
−2
8
238
765
727
0.5
Cook's distance
0.000
0.005
0.010
0.015
Leverage
Residuals vs Fitted
Residuals
−100
242
366
359
30
40
50
60
70
Fitted values

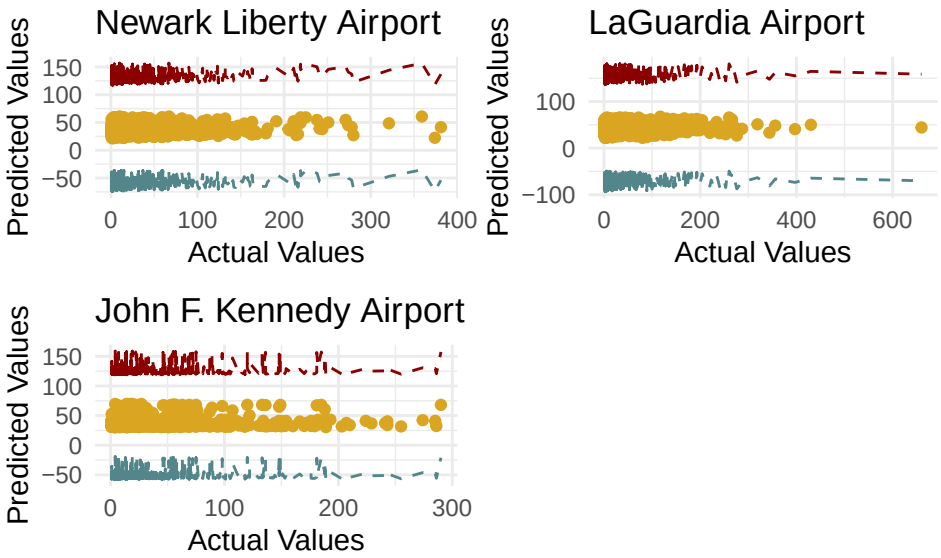


Validation Fit To evaluate predictive performance of the models, predictions were assesed using a validation set, with metrics such as Mean Squared Error (MSE) and adjusted R-squared values being calculated for both the training and test sets. Lower MSE values indicate better model performance. In this analysis, the models 'lm.ewr01', 'lm.lga01', and 'lm.fjk01' all exhibited lower test MSE compared to the full models. However, all models demonstrated low adjusted R-squared values, indicating potential issues with model fit or data quality.

```
## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

##         Model Training.MSE TEST.MSE Training.Adj.R_sq Test.Adj.R_sq
```

```
## 1 lm.ewr.full     2498.796 2536.795        0.02941514  -0.027640061
## 2    lm.ewr01     2504.891 2536.549        0.03117142  -0.002404344
## 3 lm.lga.full     3597.048 2536.795       -0.04056618  -0.057556193
## 4    lm.lga01     3580.118 2536.549       -0.02496154  -0.071492208
## 5 lm.jfk.full     2114.568 2536.795       -0.03597855  -0.030830240
## 6    lm.jfk01     2106.382 2536.549       -0.02917618  -0.007081185
```

```
##   Airport     Model Significant_Predictors Adjusted_R_squared
## 1     EWR lm.ewr.01                 humid            0.02354
## 2     LGA lm.lga.01       humid, wind_dir            0.02106
## 3     JFK lm.jfk.01        visb, wind_dir            0.03648
##   Residual_Standard_Error
## 1                   50.57
## 2                   52.45
## 3                   43.63
```



Validation Fit Summary When examining the variance explained by the models, it was found that approximately 2.35% of the variance in flight delays at Newark Liberty Airport is explained by humidity. For LaGuardia Airport, about 2.10% of the variance is explained by humidity and wind direction. For John F. Kennedy Airport, approximately 3.65% of the variance is explained by visibility and wind direction. All models indicate additional variables might be influenced by other factors.

Overall, while the models provide some predictive capability the low adjusted R-squared values suggest that further adjustments and inclusion of additional predictors could enhance the models performance.

3. Summary of Findings Our findings indicate that environmental factors such as humidity, visibility, and wind direction do have significant effects on flight delays. However, these effects differ across airports. Additionally, The models for each airport showed varying values with adjust R-squared values indicating the proportion of variance in flight delays explained by the models. The residual error provide information concerning the accuracy of model predictions.

Table 1: Summary of Findings for Each Model

Airport	Model	Significant.Predictors	Intercept	Predictor.Estimates	R.squared	Adjusted.R.squared	F.statistic	Residual.Standard.Error
EWR	lm.ewr.01	humid	15.5746 ($p < 0.01$)	humid: 0.5035 ($p < 0.001$)	0.02452	0.02354	25.09 ($p < 0.001$)	50.57
JFK	lm.jfk.01	humid, wind_dir	80.38946 ($p < 0.001$)	visib: -2.87115 ($p < 0.001$), wind_dir: -0.07205 ($p < 0.001$)	0.03840	0.03648	19.91 ($p < 0.001$)	43.63
LGA	lm.lga.01	visib, wind_dir	18.1891 ($p = 0.05$)	humid: 0.49128 ($p < 0.001$), wind_dir: -0.01727 ($p = 0.42$)	0.02302	0.02106	11.74 ($p < 0.001$)	52.45

At Newark Liberty Airport, humidity was identified as a significant predictor of departure delays, with a p-value of less than 0.05. For LaGuardia Airport, both humidity and wind direction were significant predictors, also with p-values less than 0.05. At John F. Kennedy Airport, visibility and wind direction emerged as significant predictors of flight delays, again with p-values less than 0.05. Based on the analysis, we reject the null hypothesis. As the evidence suggests that at least one of the environmental factors significantly affects flight delays across airports.

Conclusion

Overall Summary of Findings

- **RQ1:** Weather conditions, especially heavy precipitation and extreme weather events, significantly contribute to flight delays.
- **RQ2:** Flight delays differ significantly between airlines, influenced by factors like cancellation rates and operational speed.
- **RQ3:** Delays increase and become more variable during major holiday periods, impacting both departures and arrivals.
- **RQ4:** The age and manufacturer of aircraft play a significant role, with older planes and certain manufacturers experiencing more delays.
- **RQ5:** Environmental factors such as humidity, visibility, and wind direction also affect delays, though their impact varies across different airports.

Discussion Flight delays are influenced by multiple interrelated factors including weather, aircraft characteristics, holidays, and airport-specific conditions. This analysis provides valuable insights that can help stakeholders identify key areas for improving on-time performance and reduce delays at airports.

Limitations

- Prediction accuracy is limited by the lack of detailed air traffic control data and potential confounding variables.
- The weather dataset does not include extreme weather event details like thunderstorms or snowstorms.
- The analysis relies on 2013 data, which may not fully capture longer-term trends or recent changes.

Looking Ahead

- Future work should incorporate additional data such as aircraft maintenance records and air traffic control metrics, including employee retention and turnover rates.
- Developing strategies to mitigate delays during extreme weather and peak holiday seasons will help minimize disruptions.
- Infrastructure improvements at airports should be advocated to support smoother operations.
- Further investigation into human factors, such as pilot decision-making and employee well-being, could shed light on additional causes of delays.

Team Member Contributions

- **Shreya**: Planes Dataset EDA & Analysis Question #4
- **Kalyani**: Weather Dataset EDA & Analysis Question #2
- **Karen**: Flights Dataset EDA & Analysis Question #3
- **Crystal**: Airports Dataset EDA & Analysis Question #5
- **Mason**: Analysis Question #1

Alternative Strategies & Back Up Plan:

As a backup idea, we are planning on seeing if there is any correlation between the amount of delays present in the different airports. Our data deals with the airports EWR, JFK, and LGA which are all different airports within New York City. Our first question is to figure out if the JFK airport has a different amount of delays compared to LGA or EWR if there is a higher amount of precipitation in the JFK area. Although all the airports are in New York, within the different areas of the city, there can be different amounts of precipitation and rainfall that occur. Our second question is to decide whether the different airports have different models of planes and if the difference affects the amounts of delays. For example if a plane is older or a different configuration, does that lead to more delays due to cleaning or maintenance? And lastly, our third question is whether the three different airports have different airlines coming in and out and if these differing airlines affect the amount of delays present on a given day. For example, if Delta services one airport and not another, does that increase or decrease the amount of total delays for an airport. These questions can be further investigated if our first set of questions are not approved or if we need more content to explore within our project. These sets of backup questions will further explore the flight data we have.