

# Final Project STAT167

Group: Statistically Speaking

Shreya Mohan, Kalyani Mantiraju, Crystal Arevalo, Karen Alvarez, Mason Lam, Eric Yang

06/02/2025

## Libraries

```
#install.packages("dunn.test")
#install.packages("multcomp")
# install.packages("nortest")
# install.packages("rstatix")
# install.packages("mgcv")
library(mgcv)
library(rstatix)
library(nycflights13)
library(tidyverse)
library(car)
library(dunn.test)
library(gridExtra)
library(nycflights13)
library(dplyr)
library(ggplot2)
library(tidyr)
library(broom)
library(car)
library(multcomp)
library(nortest)
library(rstatix)
library(dplyr)
```

## 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?
6. What impact does precipitation have on specific airports and weather-related delays?

## 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

```
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-
```

```
names(airports)
```

```
## [1] "faa"    "name"   "lat"    "lon"    "alt"    "tz"    "dst"    "tzone"
```

```
str(airports)
```

```
## # tibble [1,458 x 8] (S3: tbl_df/tbl/data.frame)
## # $ faa : chr [1:1458] "04G" "06A" "06C" "06N" ...
## # $ name : chr [1:1458] "Lansdowne Airport" "Moton Field Municipal Airport" "Schaumburg Regional" "Ran...
## # $ lat : num [1:1458] 41.1 32.5 42.4 41.4 31.1 ...
## # $ lon : num [1:1458] -80.6 -85.7 -88.1 -74.4 -81.4 ...
## # $ alt : num [1:1458] 1044 264 801 523 11 ...
## # $ tz : num [1:1458] -5 -6 -6 -5 -5 -5 -5 -5 -8 ...
## # $ dst : chr [1:1458] "A" "A" "A" "A" ...
## # $ timezone: chr [1:1458] "America/New_York" "America/Chicago" "America/Chicago" "America/New_York" ...
## # - attr(*, "spec")=
## #   .. cols(
## #     .. id = col_double(),
## #     .. name = col_character(),
## #     .. city = col_character(),
## #     .. country = col_character(),
## #     .. faa = col_character(),
## #     .. icao = col_character(),
## #     .. lat = col_double(),
## #     .. lon = col_double(),
## #     .. alt = col_double(),
## #     .. tz = col_double(),
## #     .. dst = col_character(),
## #     .. timezone = col_character()
## #     .. )
```

```
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, -8, -5, -6, -5, -5, -5, -5, ~
## #> #> $ dst <chr> "A", "A", "A", "A", "A", "A", "U", "A", "A", "U", "A", "A", "A", ~
## #> #> $ timezone <chr> "America/New_York", "America/Chicago", "America/Chicago", "Ameri~
```

## 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

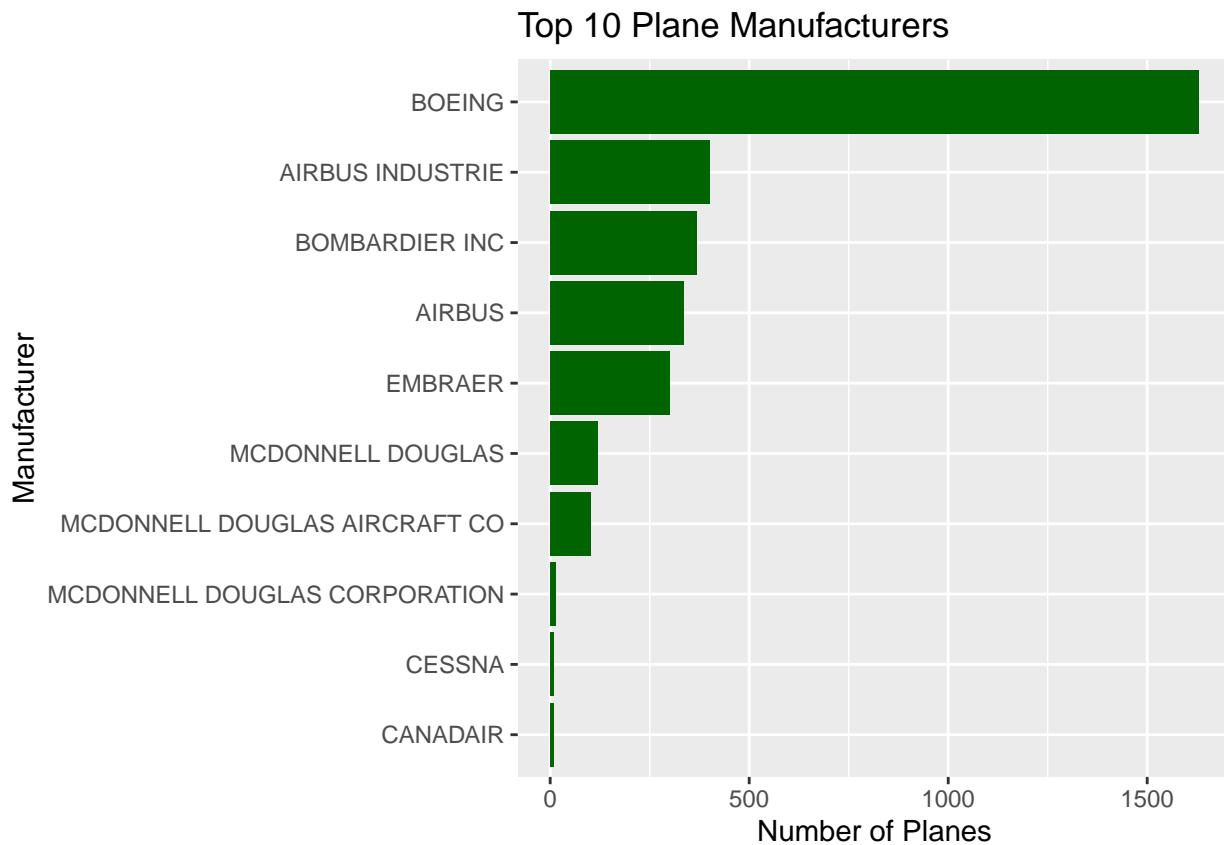
```
##  
## Missing values per column:  
  
##      tailnum      year      type manufacturer      model engines  
##      0            70          0          0            0            0  
##      seats       speed     engine  
##      0            3299        0  
  
##  
## Column types and structure:  
  
## 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, ~  
## $ seats         <int> 55, 182, 182, 182, 55, 182, 182, 182, 182, 182, 55, 55, 5~  
## $ speed         <int> 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~
```

```

## 
## Summary statistics:

##      tailnum          year       type      manufacturer
##  Length:3322   Min.   :1956   Length:3322   Length:3322
##  Class  :character 1st Qu.:1997   Class  :character  Class  :character
##  Mode   :character Median :2001    Mode   :character  Mode   :character
##                               Mean   :2000
##                               3rd Qu.:2005
##                               Max.   :2013
##                               NA's    :70
##      model           engines      seats      speed
##  Length:3322   Min.   :1.000   Min.   : 2.0   Min.   : 90.0
##  Class  :character 1st Qu.:2.000   1st Qu.:140.0  1st Qu.:107.5
##  Mode   :character Median :2.000    Median :149.0  Median :162.0
##                               Mean   :1.995   Mean   :154.3  Mean   :236.8
##                               3rd Qu.:2.000   3rd Qu.:182.0  3rd Qu.:432.0
##                               Max.   :4.000    Max.   :450.0  Max.   :432.0
##                                         NA's   :3299
##      engine
##  Length:3322
##  Class  :character
##  Mode   :character
## 
## 
## 
## 

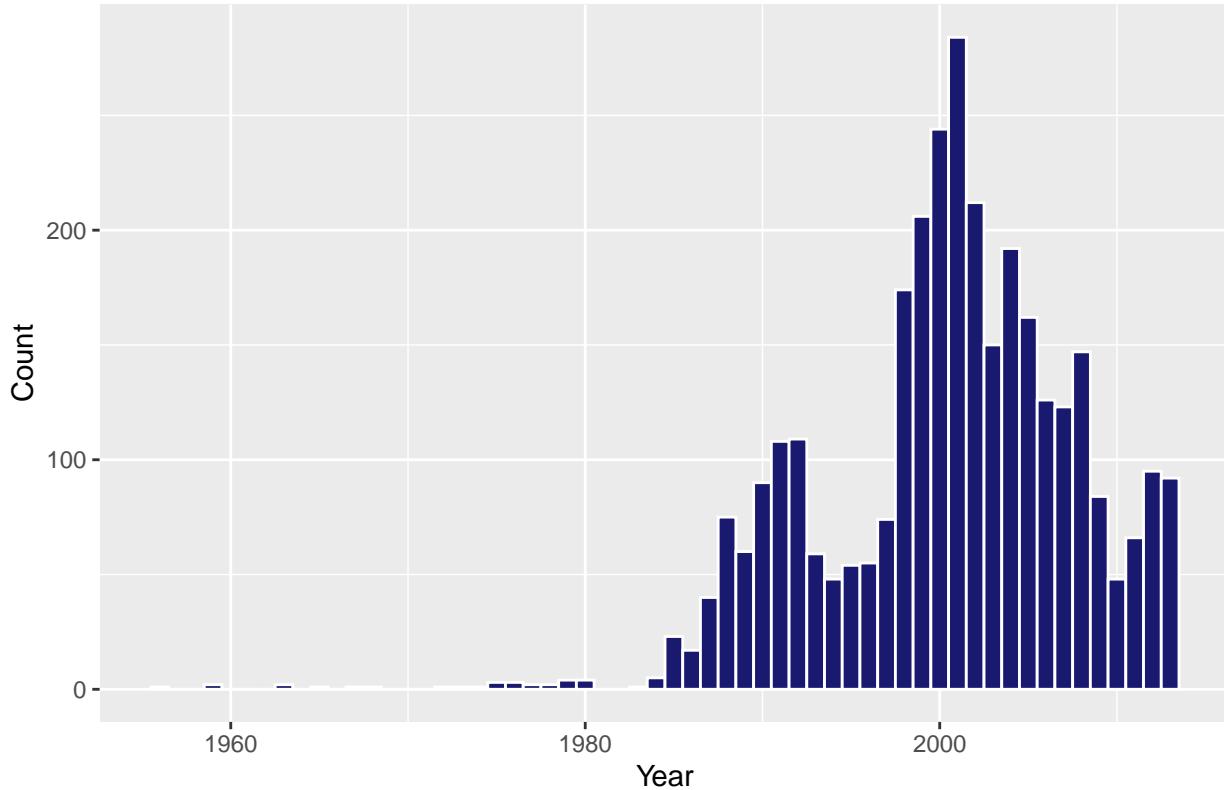
## Selecting by n
```



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.

```
## Warning: Removed 70 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

## Distribution of Plane Manufacture Years



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.
```

```

##
## Summary statistics:
```

```

##   carrier          name
##   Length:16        Length:16
##   Class :character Class :character
##   Mode  :character Mode  :character
```

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
```

```

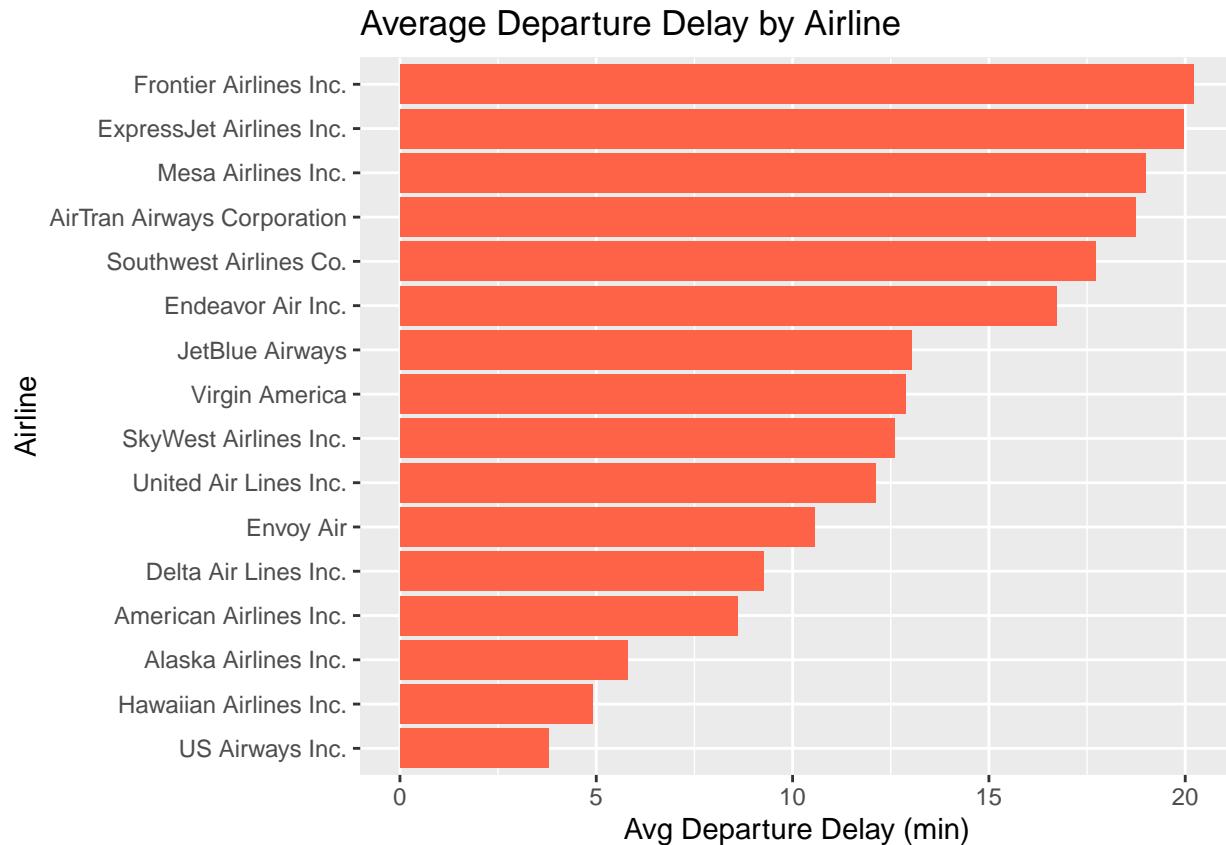
## 7 EV      ExpressJet Airlines Inc.
## 8 F9      Frontier Airlines Inc.
## 9 HA      Hawaiian Airlines Inc.
## 10 B6     JetBlue Airways
## 11 YV     Mesa Airlines Inc.
## 12 OO     SkyWest Airlines Inc.
## 13 WN     Southwest Airlines Co.
## 14 US     US Airways Inc.
## 15 UA     United Air Lines 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)"
  )

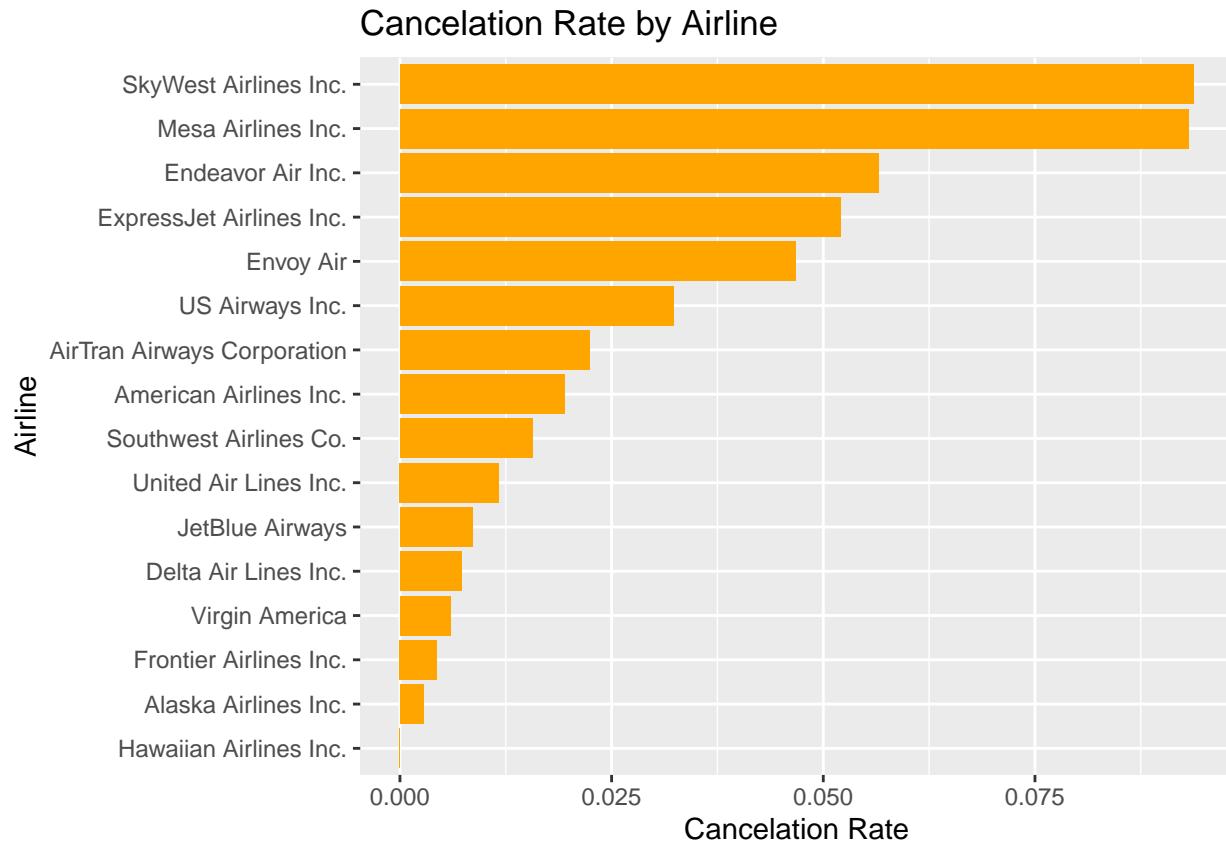
```



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 = "Cancellation Rate by Airline",
    x = "Airline",
    y = "Cancellation 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. ## Planes Dataset EDA

### 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           0        9430  
##    distance        hour      minute      time_hour  
##      0            0            0            0            0  
  
##  
## Column types and structure:  
  
## Rows: 336,776  
## Columns: 19  
## $ year      <int> 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~  
## $ day       <int> 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, -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, 5, 6, 6, 6~
## $ minute          <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 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>

##
## Summary statistics:

##   year      month      day      dep_time      sched_dep_time

```

```

## Min.    :2013   Min.    : 1.000   Min.    : 1.00   Min.    : 1   Min.    : 106
## 1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907  1st Qu.: 906
## Median  :2013   Median  : 7.000   Median  :16.00   Median  :1401  Median  :1359
## Mean    :2013   Mean    : 6.549   Mean    :15.71   Mean    :1349  Mean    :1344
## 3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744 3rd Qu.:1729
## Max.    :2013   Max.    :12.000   Max.    :31.00   Max.    :2400  Max.    :2359
##                                     NA's    :8255
##      dep_delay        arr_time     sched_arr_time arr_delay
## Min.    :-43.00   Min.    : 1   Min.    : 1   Min.    :-86.000
## 1st Qu.:-5.00    1st Qu.:1104  1st Qu.:1124  1st Qu.:-17.000
## Median :-2.00    Median :1535  Median :1556  Median :-5.000
## Mean   :12.64   Mean   :1502  Mean   :1536  Mean   : 6.895
## 3rd Qu.:11.00   3rd Qu.:1940  3rd Qu.:1945  3rd Qu.: 14.000
## Max.   :1301.00  Max.   :2400  Max.   :2359  Max.   :1272.000
## NA's   :8255   NA's   :8713  NA's   :9430
##      carrier       flight      tailnum      origin
## Length:336776   Min.    : 1   Length:336776   Length:336776
## Class  :character 1st Qu.: 553  Class  :character  Class  :character
## Mode   :character Median :1496  Mode   :character  Mode   :character
##                      Mean   :1972
##                      3rd Qu.:3465
##                      Max.   :8500
##
##      dest          air_time     distance      hour
## Length:336776   Min.    :20.0  Min.    : 17  Min.    : 1.00
## Class  :character 1st Qu.: 82.0  1st Qu.: 502  1st Qu.: 9.00
## Mode   :character Median :129.0  Median : 872  Median :13.00
##                      Mean   :150.7  Mean   :1040  Mean   :13.18
##                      3rd Qu.:192.0  3rd Qu.:1389  3rd Qu.:17.00
##                      Max.   :695.0   Max.   :4983  Max.   :23.00
##                      NA's   :9430
##      minute        time_hour
## Min.    : 0.00  Min.    :2013-01-01 05:00:00.00
## 1st Qu.: 8.00  1st Qu.:2013-04-04 13:00:00.00
## Median :29.00  Median :2013-07-03 10:00:00.00
## Mean   :26.23  Mean   :2013-07-03 05:22:54.64
## 3rd Qu.:44.00  3rd Qu.:2013-10-01 07:00:00.00
## Max.   :59.00  Max.   :2013-12-31 23:00:00.00
##

```

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>

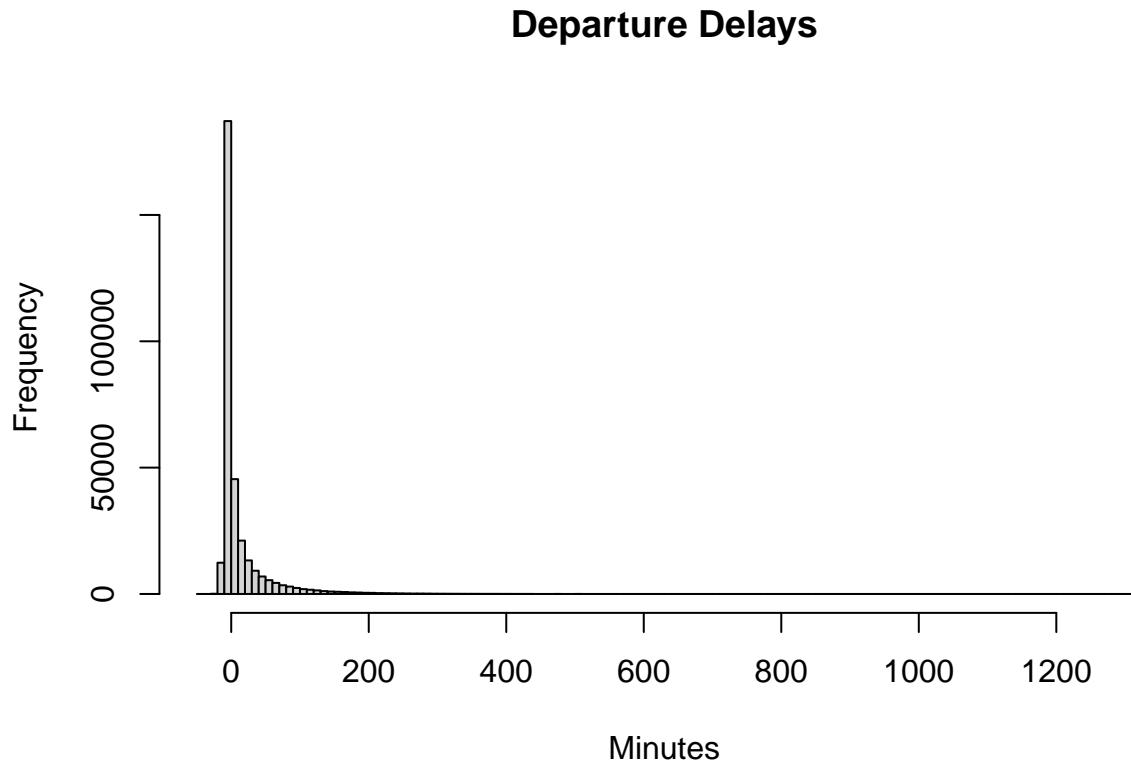
```

#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")

```

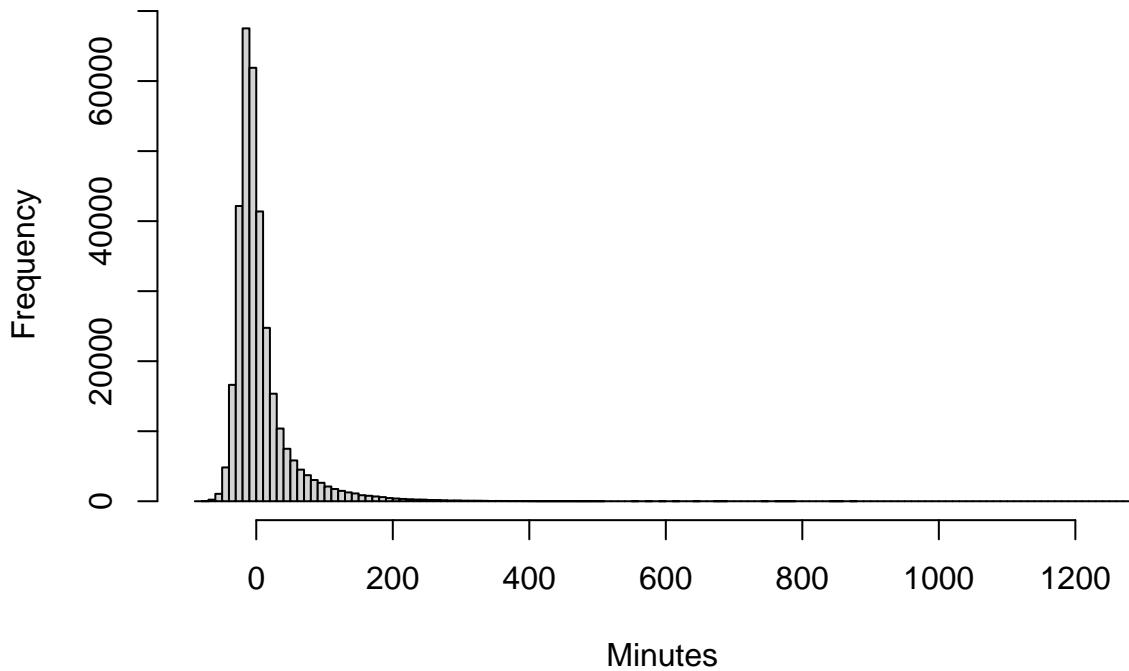


```

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

```

## Arrival Delays



```
#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

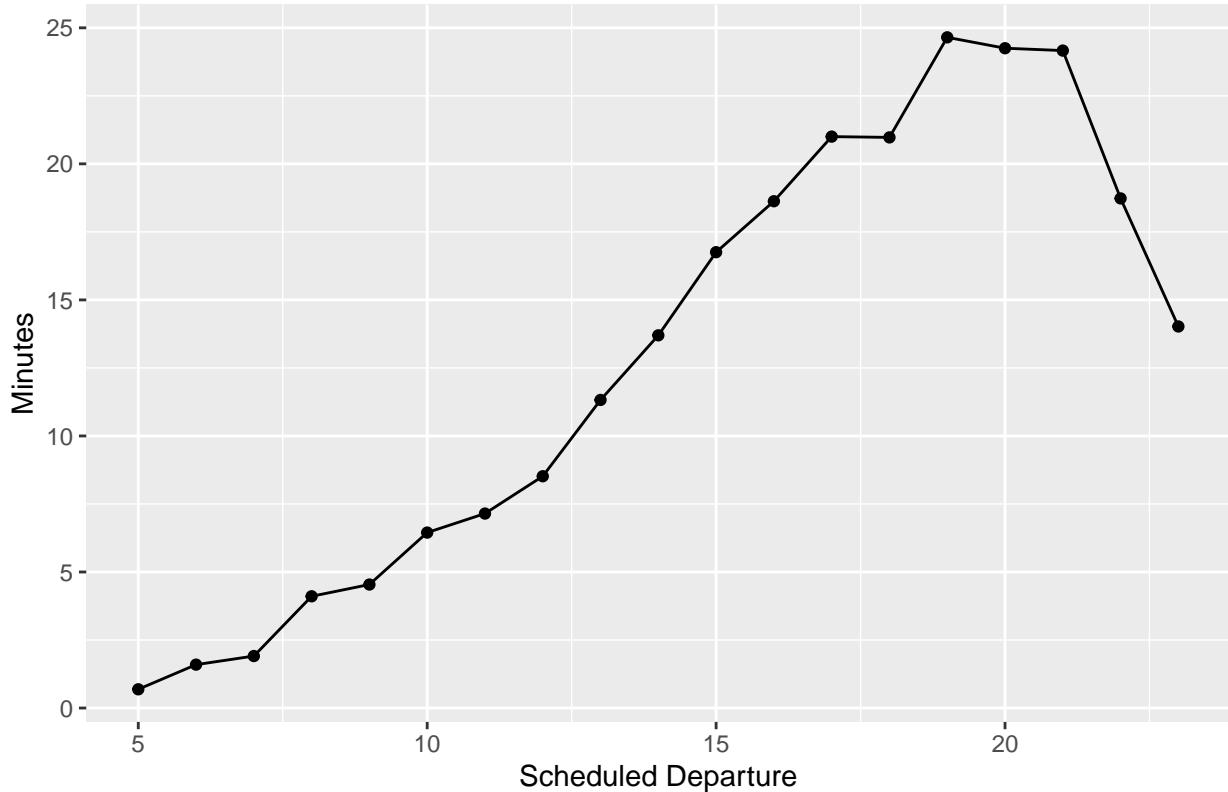
#most of the departure delays do not go over 200 minutes and the arrival delays have very few delays past 200 minutes.

#Delay patterns

#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")
```

## Average Departure Delay by Hour



#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.

#Delays by Airport

```
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

#Ranking airlines by delay

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

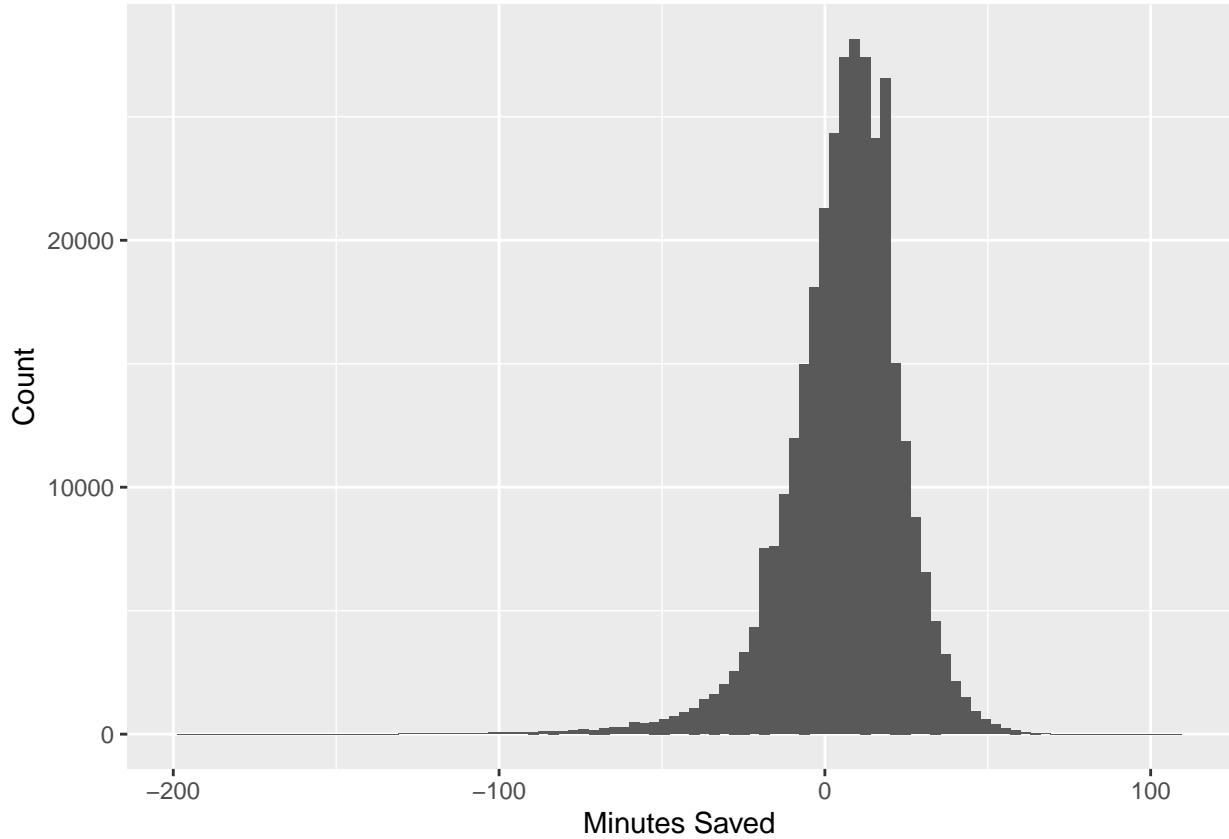
## # A tibble: 16 x 2

```
##      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")
```

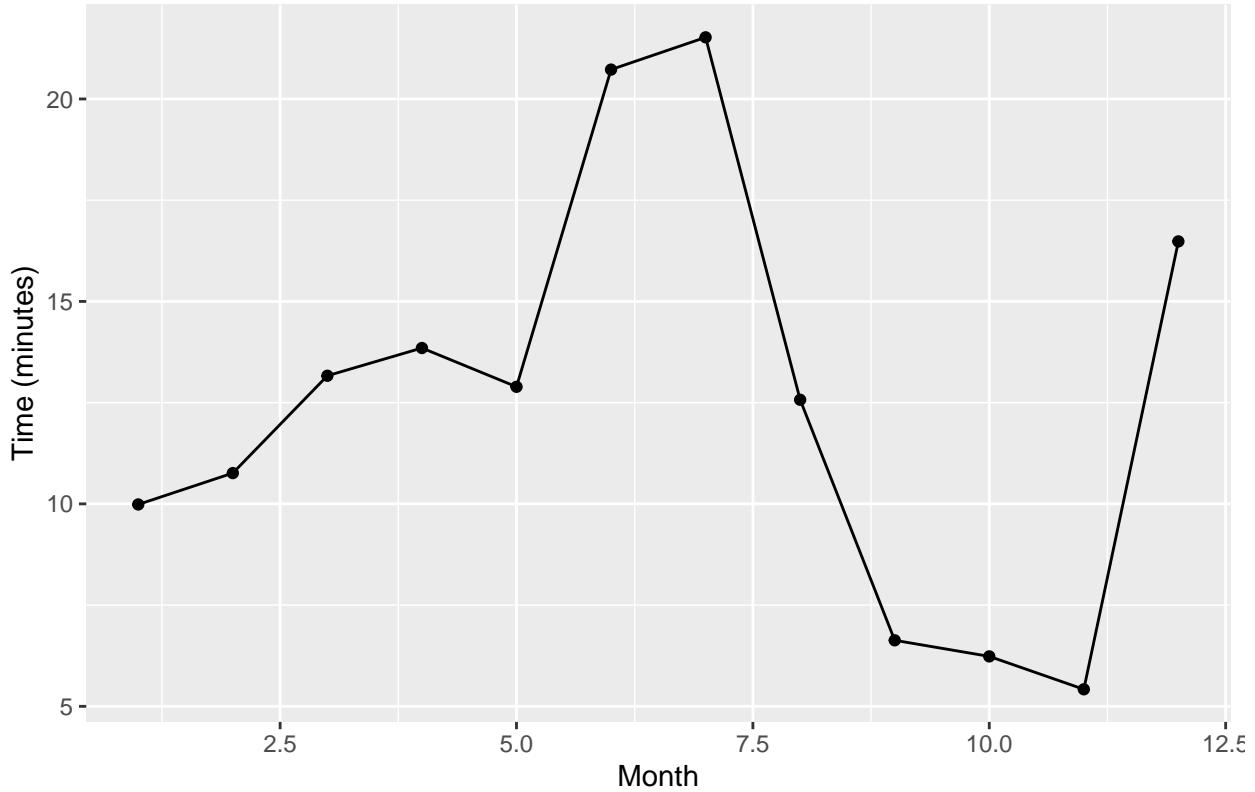


#We can see that the majority of the flights did not save any minutes on the arrival delay and actually ended up being delayed more. Some flights did in fact save minutes but it was less than 50% of all flights.

#Delays by Month

```
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)")
```

## Monthly Departure Delays



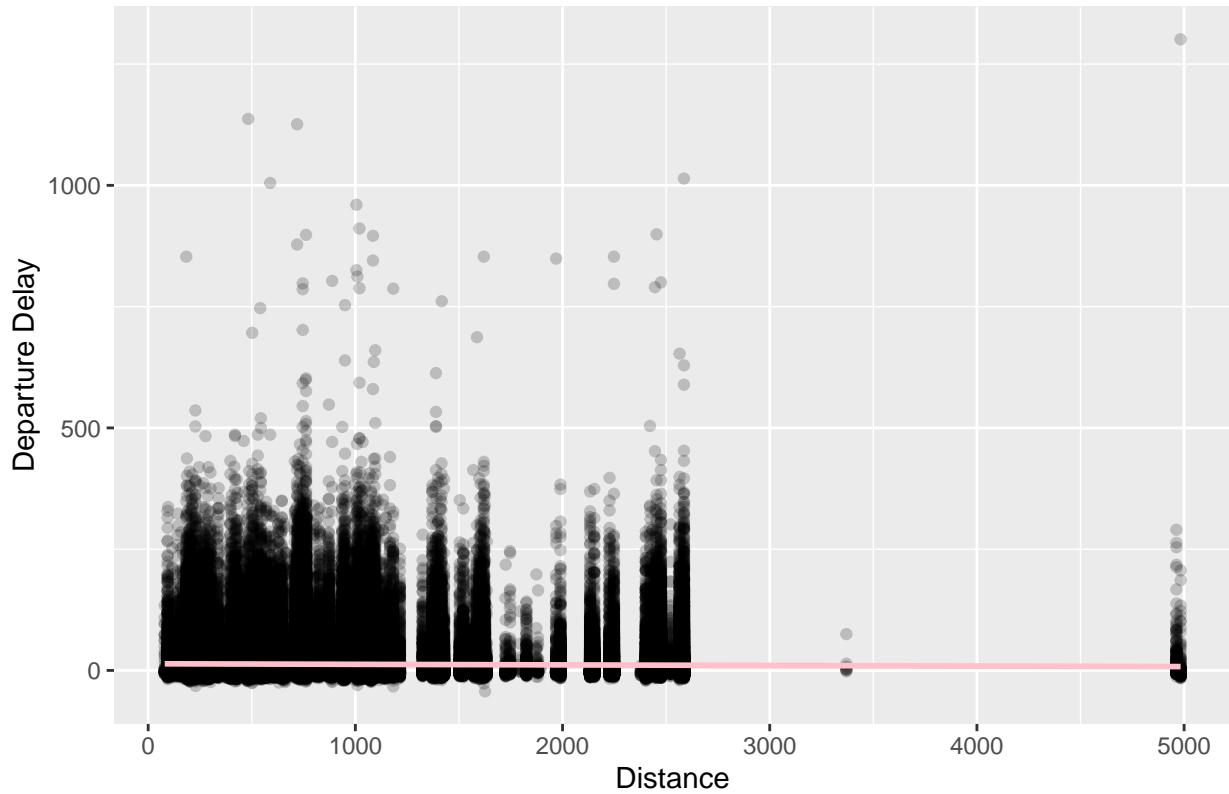
#we see that the majority of flights are delayed from May to mid July, and there is another peak at December. The months with the shorest delays are September and October.

#Does distance affect the amount of delays?

```
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")
```

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

## Distance vs Departure Delay



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

#Flights traveled the longest by distance

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

#arrival delays per carrier

```

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

```

```

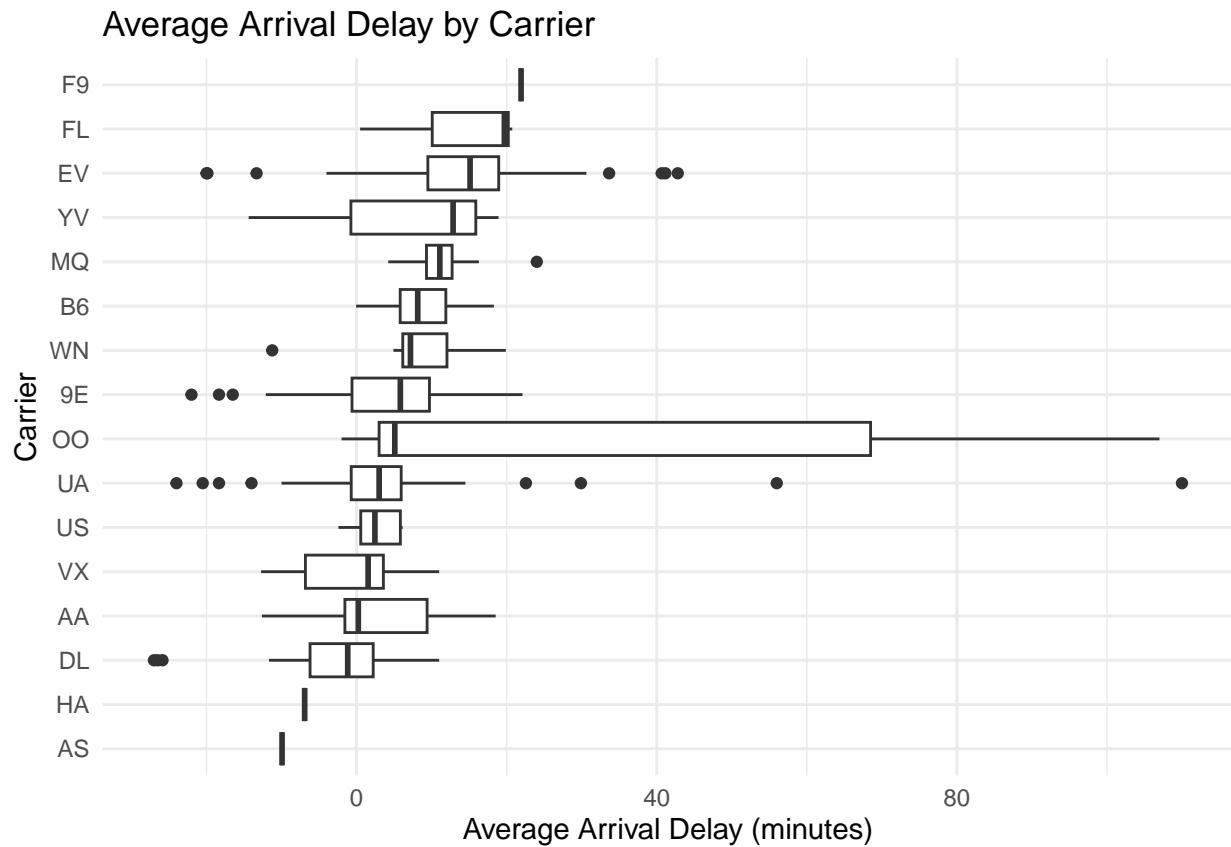
## `summarise()` has grouped output by 'carrier'. You can override using the
## `.`groups` argument.

```

```

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", ~
## $ year       <int> 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, ~
## $ day        <int> 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, 39.92, 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.~
## $ 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, 20.~
## $ precip      <dbl> 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, 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> <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  41.0  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> <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>

## 
## Summary statistics:

```

```

##      origin          year        month       day
##  Length:26115    Min.   :2013   Min.   : 1.000   Min.   : 1.00
##  Class :character 1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00
##  Mode  :character Median :2013   Median : 7.000   Median :16.00
##                               Mean   :2013   Mean   : 6.504   Mean   :15.68
##                               3rd Qu.:2013   3rd Qu.: 9.000   3rd Qu.:23.00
##                               Max.   :2013   Max.   :12.000   Max.   :31.00
##
##      hour         temp        dewp       humid
##  Min.   : 0.00   Min.   :10.94   Min.   :-9.94   Min.   : 12.74
##  1st Qu.: 6.00   1st Qu.:39.92   1st Qu.:26.06   1st Qu.: 47.05
##  Median :11.00   Median :55.40   Median :42.08   Median : 61.79
##  Mean   :11.49   Mean   :55.26   Mean   :41.44   Mean   : 62.53
##  3rd Qu.:17.00   3rd Qu.:69.98   3rd Qu.:57.92   3rd Qu.: 78.79
##  Max.   :23.00   Max.   :100.04  Max.   :78.08   Max.   :100.00
##  NA's   : 1       NA's   : 1       NA's   : 1       NA's   : 1
##      wind_dir     wind_speed     wind_gust     precip
##  Min.   : 0.0   Min.   : 0.000   Min.   :16.11   Min.   :0.0000000
##  1st Qu.:120.0  1st Qu.: 6.905   1st Qu.:20.71   1st Qu.:0.0000000
##  Median :220.0  Median : 10.357   Median :24.17   Median :0.0000000
##  Mean   :199.8   Mean   : 10.518   Mean   :25.49   Mean   :0.004469
##  3rd Qu.:290.0  3rd Qu.: 13.809   3rd Qu.:28.77   3rd Qu.:0.0000000
##  Max.   :360.0   Max.   :1048.361  Max.   :66.75   Max.   :1.2100000
##  NA's   :460     NA's   : 4       NA's   :20778
##      pressure     visib       time_hour
##  Min.   : 983.8   Min.   : 0.000   Min.   :2013-01-01 01:00:00.0
##  1st Qu.:1012.9   1st Qu.:10.000   1st Qu.:2013-04-01 21:30:00.0
##  Median :1017.6   Median :10.000   Median :2013-07-01 14:00:00.0
##  Mean   :1017.9   Mean   : 9.255   Mean   :2013-07-01 18:26:37.7
##  3rd Qu.:1023.0   3rd Qu.:10.000   3rd Qu.:2013-09-30 13:00:00.0
##  Max.   :1042.1   Max.   :10.000   Max.   :2013-12-30 18:00:00.0
##  NA's   :2729

```

*# Related Questions from our Proposal:*

```

# --- How do weather conditions affect flight delays?
# --- How do environmental factors like humidity, visibility, and wind affect flight delays?
# --- What impact does precipitation have on specific airports and weather-related delays?

```

*# Average Temperature by Month*

```

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

```

```

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

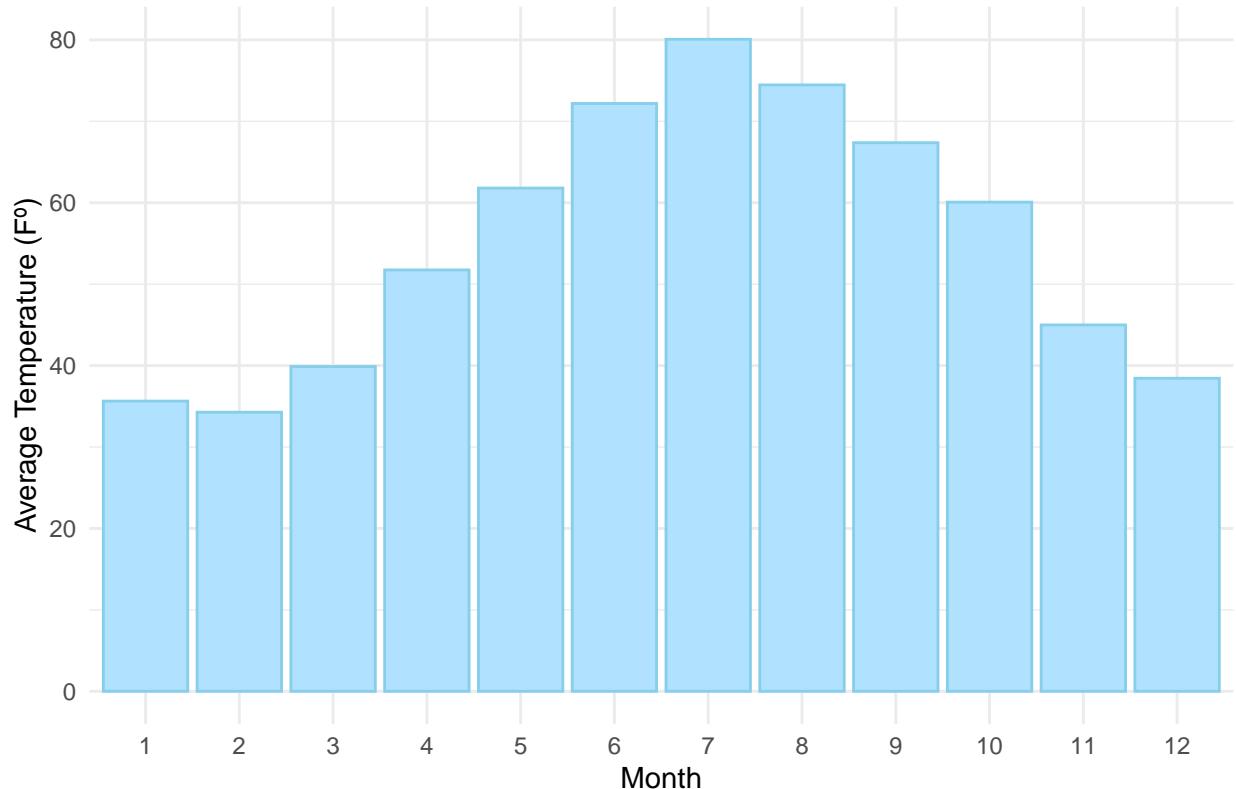
```

```

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()

```

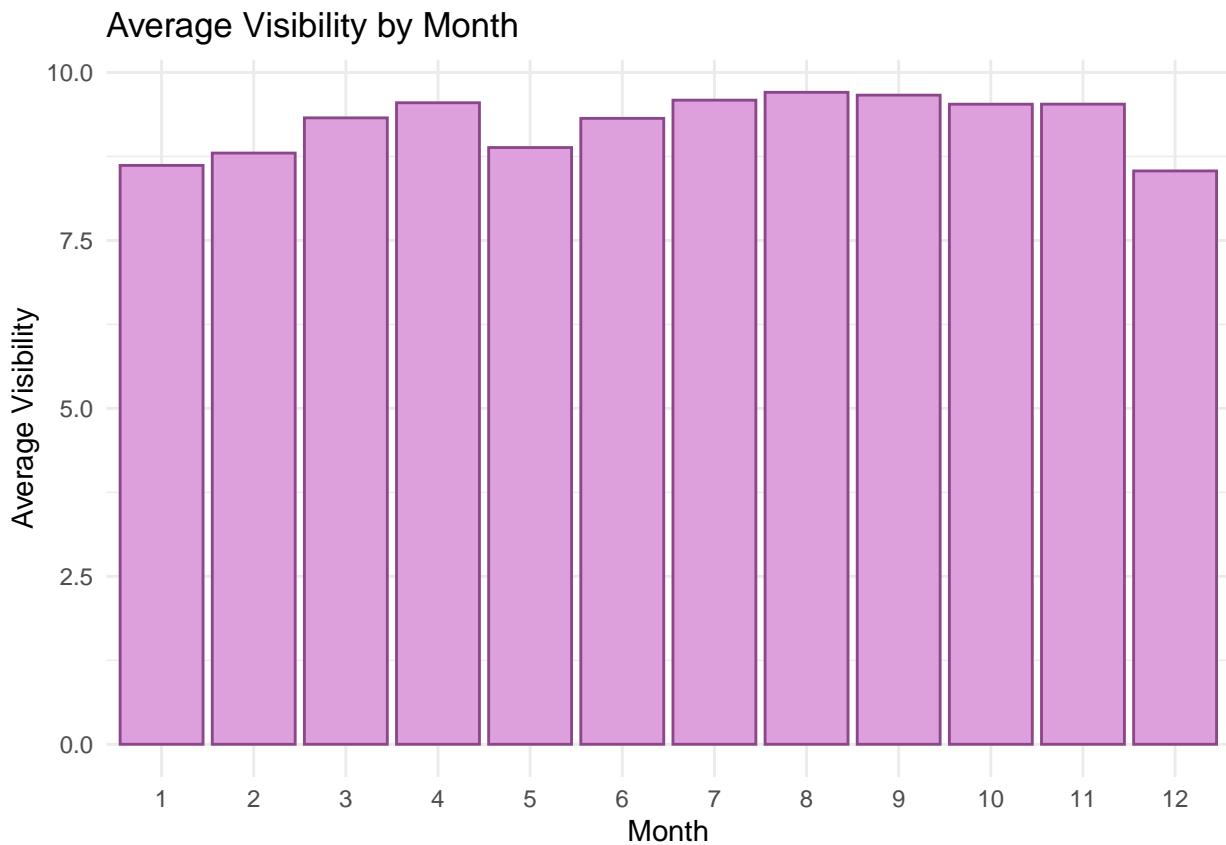
## Average Temperature by Month



```
# Average Visibility by Month
# --- The average visibility does not greatly vary by month looking at the average value. However, we can see that there are a lot of outliers. Removing these outliers and comparing the average visibility during flight delays will give us a more accurate picture.
# --- Looking at the boxplots, we can see that there are a lot of outliers. Removing these outliers and comparing the average visibility during flight delays will give us a more accurate picture.

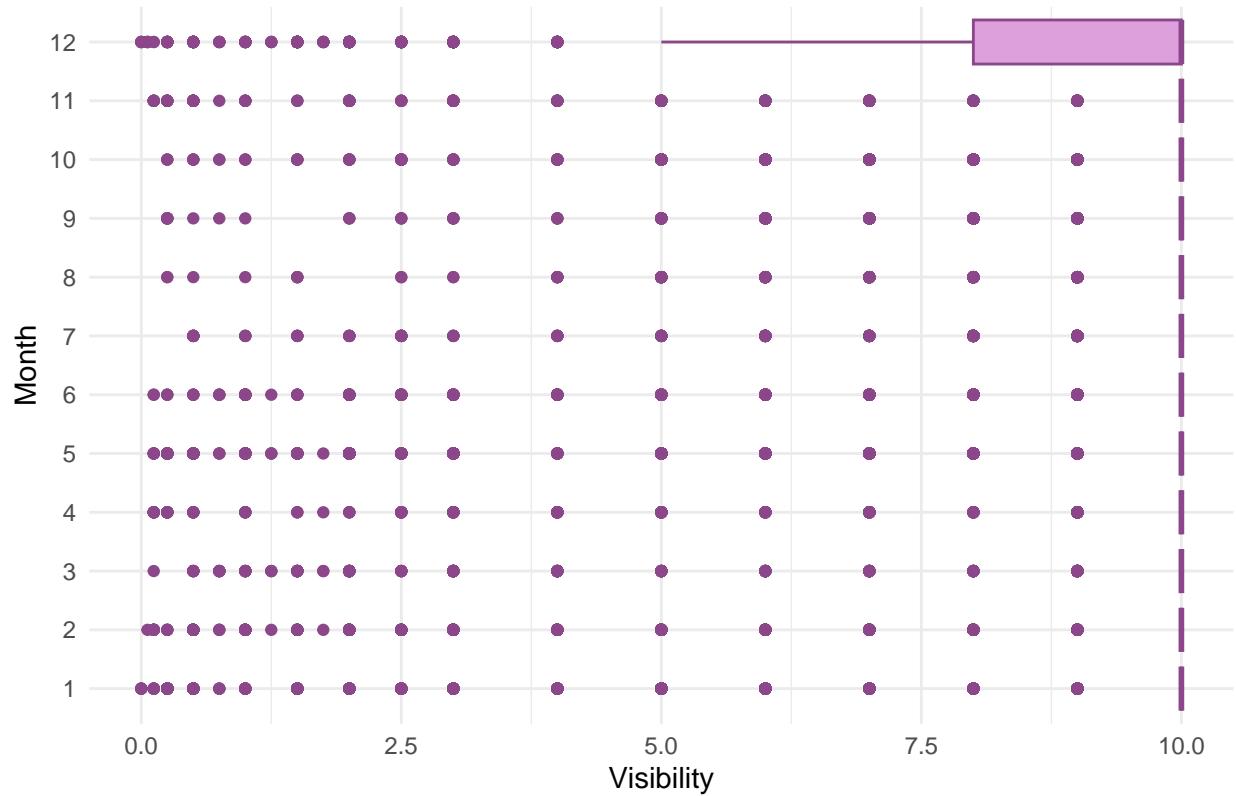
monthlyavgvisib <- weather %>%
  group_by(month) %>%
  summarise(monthlyavgvisib = mean(visib, na.rm = TRUE))

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()
```



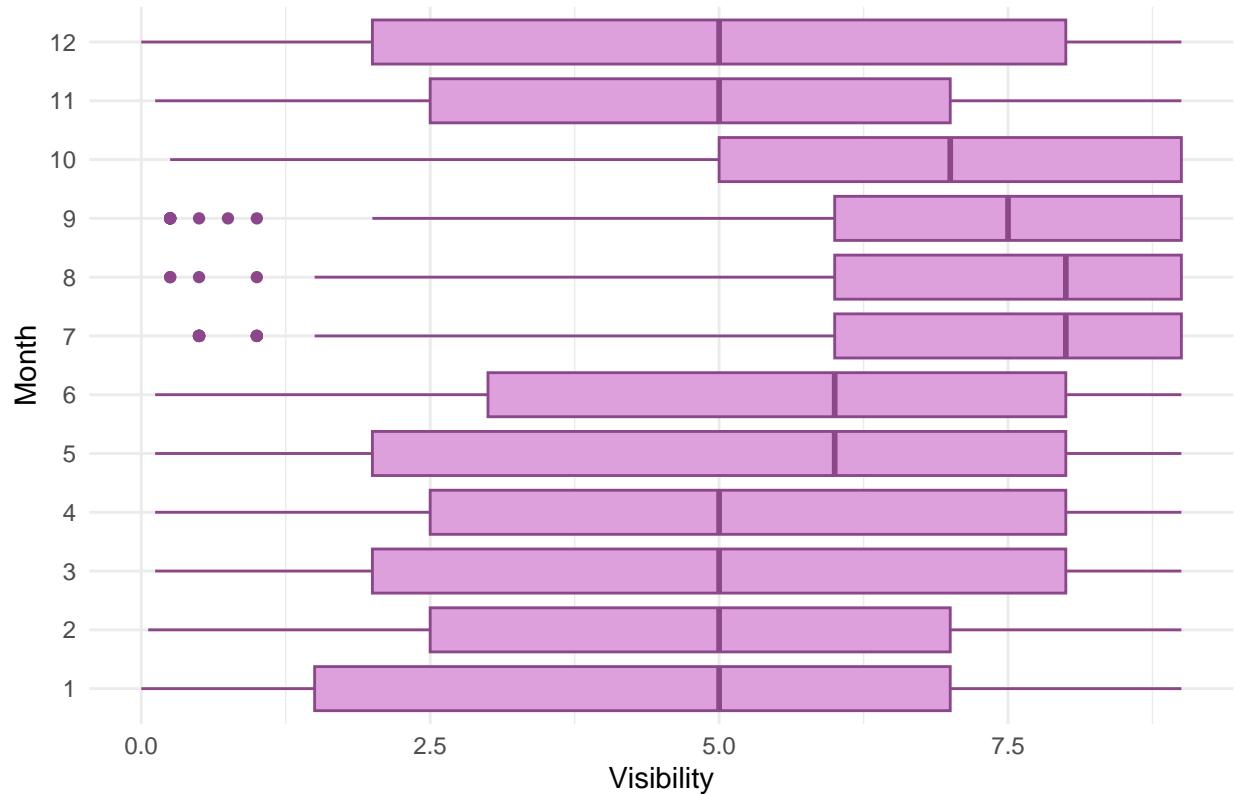
```
ggplot(weather, aes(x = factor(month), y = visib)) +
  geom_boxplot(fill = "plum", color = "orchid4") +
  labs(title = "Distribution of Visibility by Month",
       x = "Month",
       y = "Visibility") +
  coord_flip() +
  theme_minimal()
```

## Distribution of Visibility by Month



```
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()
```

## Distribution of Visibility < 10 by Month

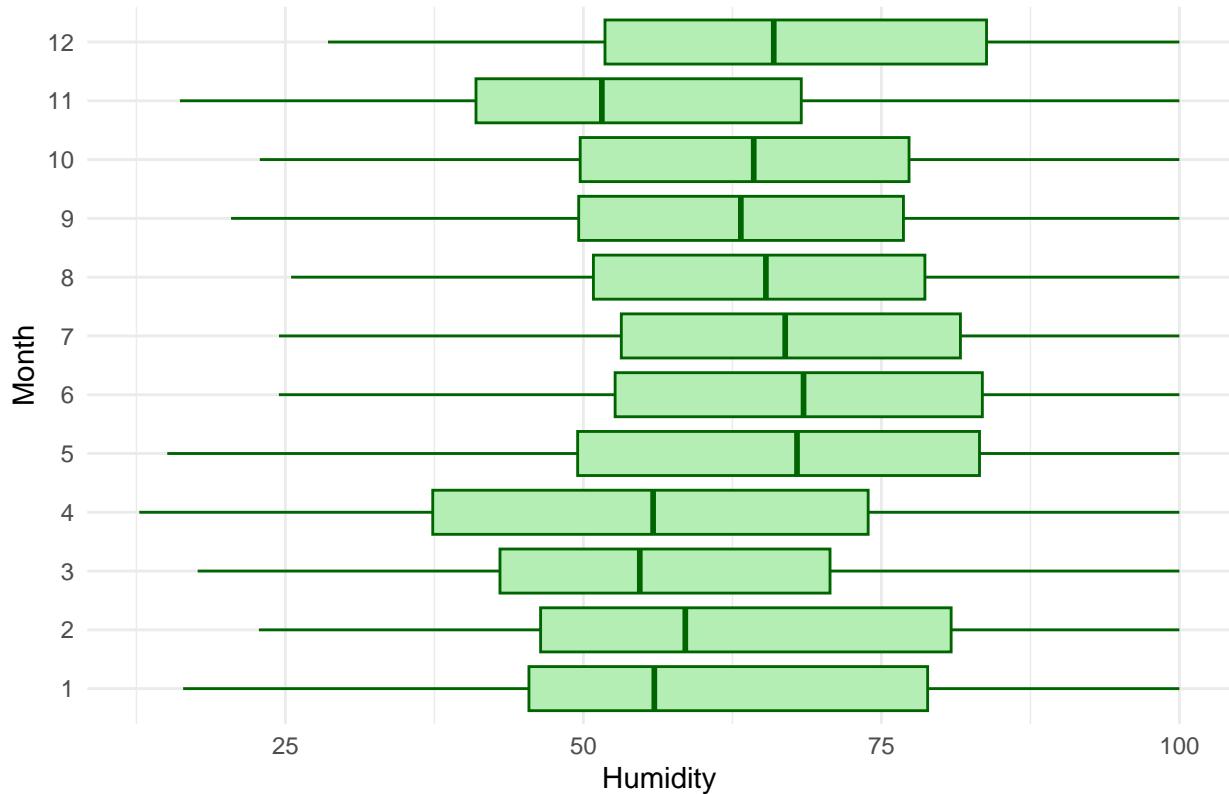


```
# Distribution of Humidity by Month
# --- The distribution of humidity varies by month, but there does not seem to be significant difference
# --- We can further explore the role of humidity by comparing it to other weather variables and flight

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()

## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

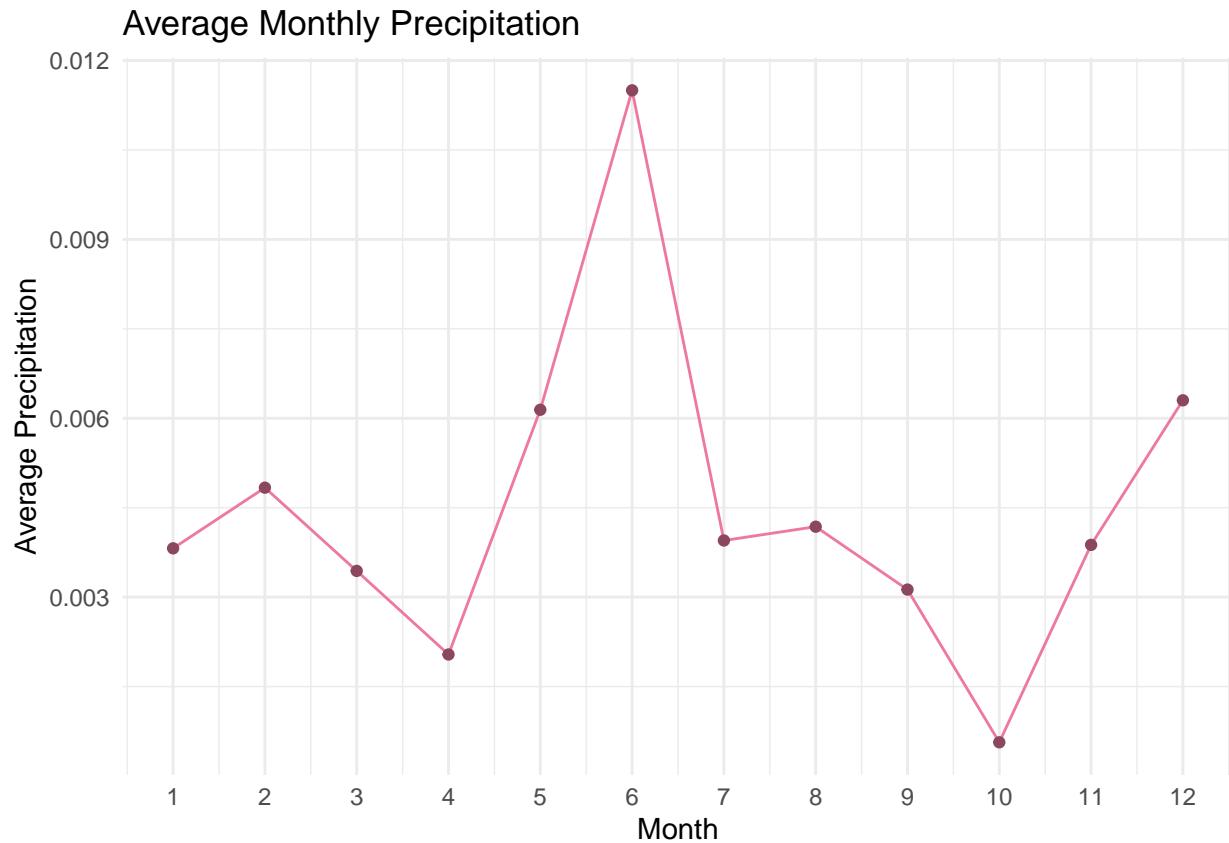
## Distribution of Humidity by Month (With Outliers)



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

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()
```



```
# Correlation Between Variables
# --- We can explore the correlation between different weather variables and see how they may work together
cor(weather$precip, weather$visib, use = "complete.obs")

## [1] -0.3199118

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

## [1] -0.5167424
```

# 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 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)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## 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

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.

```

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

## [1] "Welch Anova Test"

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

# 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)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## 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 Cycle Recip~ -4.01    -20.8     12.8    0.98    ns
## 2 arr_delay Cycle Turbo~ -2.00    -18.6     14.6    0.999   ns
## 3 arr_delay Cycle Turbo~ -6.53    -23.1     10.1    0.849   ns
## 4 arr_delay Cycle Turbo~ -4.83    -30.2     20.5    0.994   ns
## 5 arr_delay 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 difference between engines. 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 airline.

```

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

```

## # 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)

```

```

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## 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

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

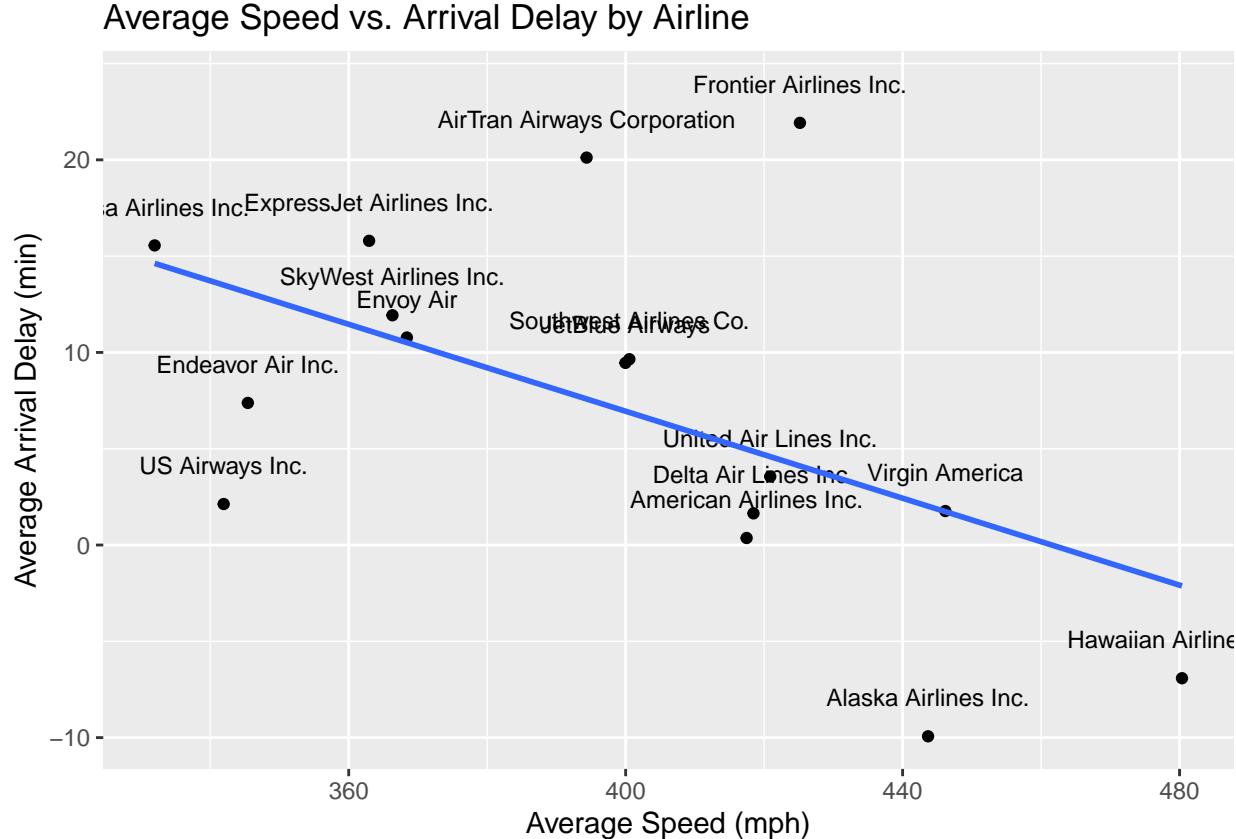
## 
## Pearson's product-moment correlation
##
## data: speed_delay_summary$avg_speed and speed_delay_summary$avg_arr_delay
## t = -2.3331, df = 14, p-value = 0.03507
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.81187916 -0.04529422
## sample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

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'

## Warning: The following aesthetics were dropped during statistical transformation: label.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?

```



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))

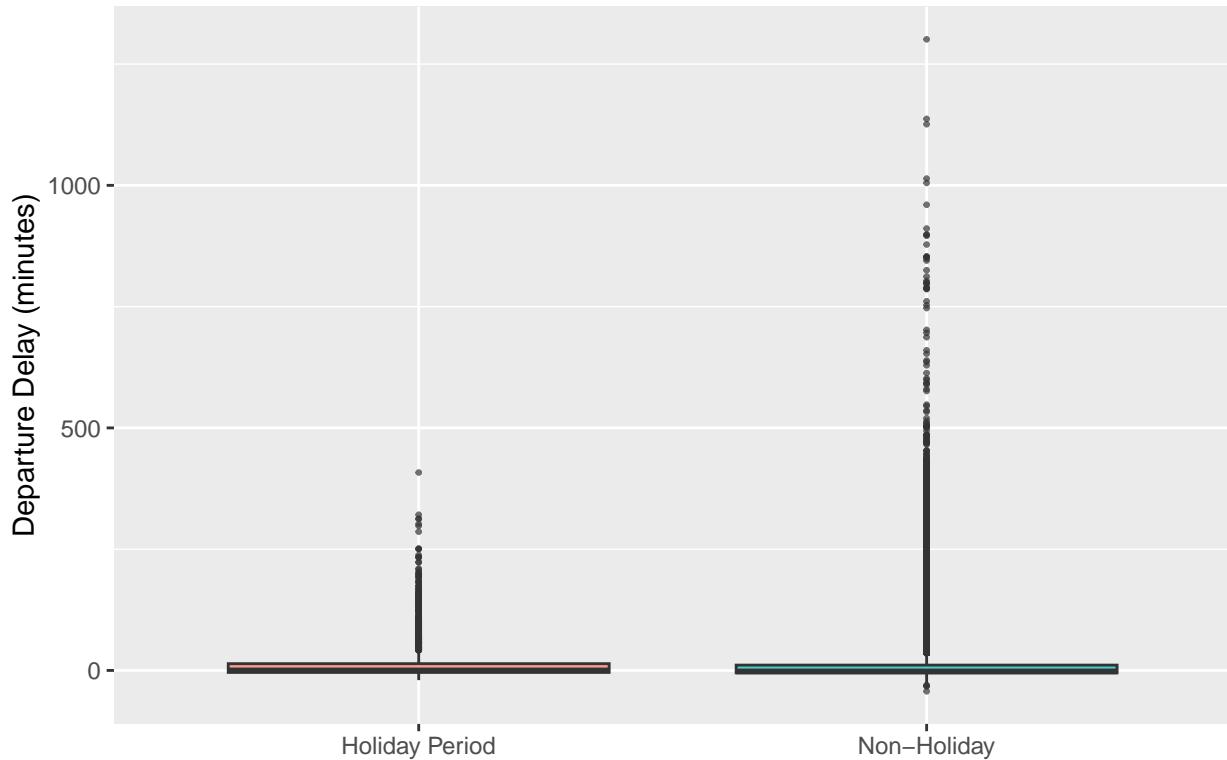
table(flights_holiday$holiday_flag)

##
## 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")
```

## Departure Delay Distributions: Holiday Period vs. NonHoliday Flights



```

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

```

```
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
```

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

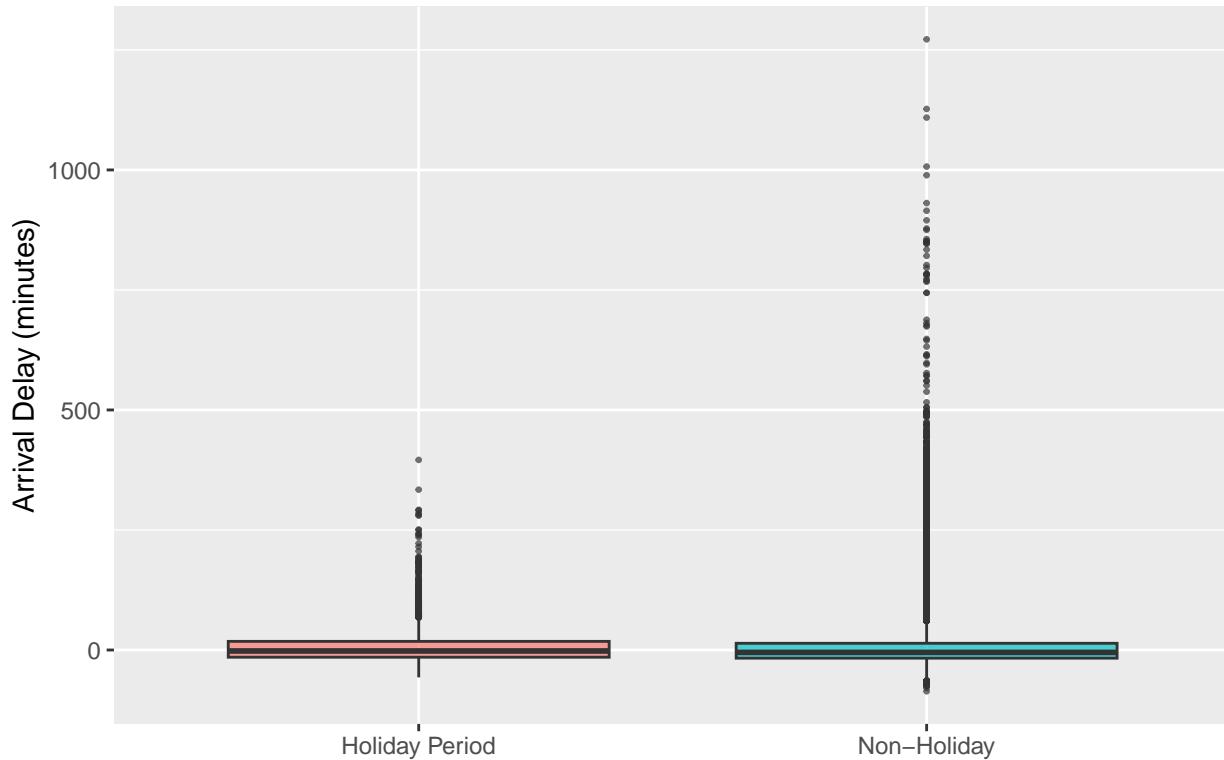
```
## Wilcoxon test p-value: 2.42e-16
```

##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 ~2.42e-16 which is significant and allows us to conclude that departure delays are significant between holiday and non holiday periods.

```
#Arrival Delays
```

```
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")
```

## Arrival Delay Distributions: Holiday Period vs. NonHoliday Flights



```

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

```

```

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 = 572896208, p-value = 7.329e-10
## alternative hypothesis: true location shift is not equal to 0

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)

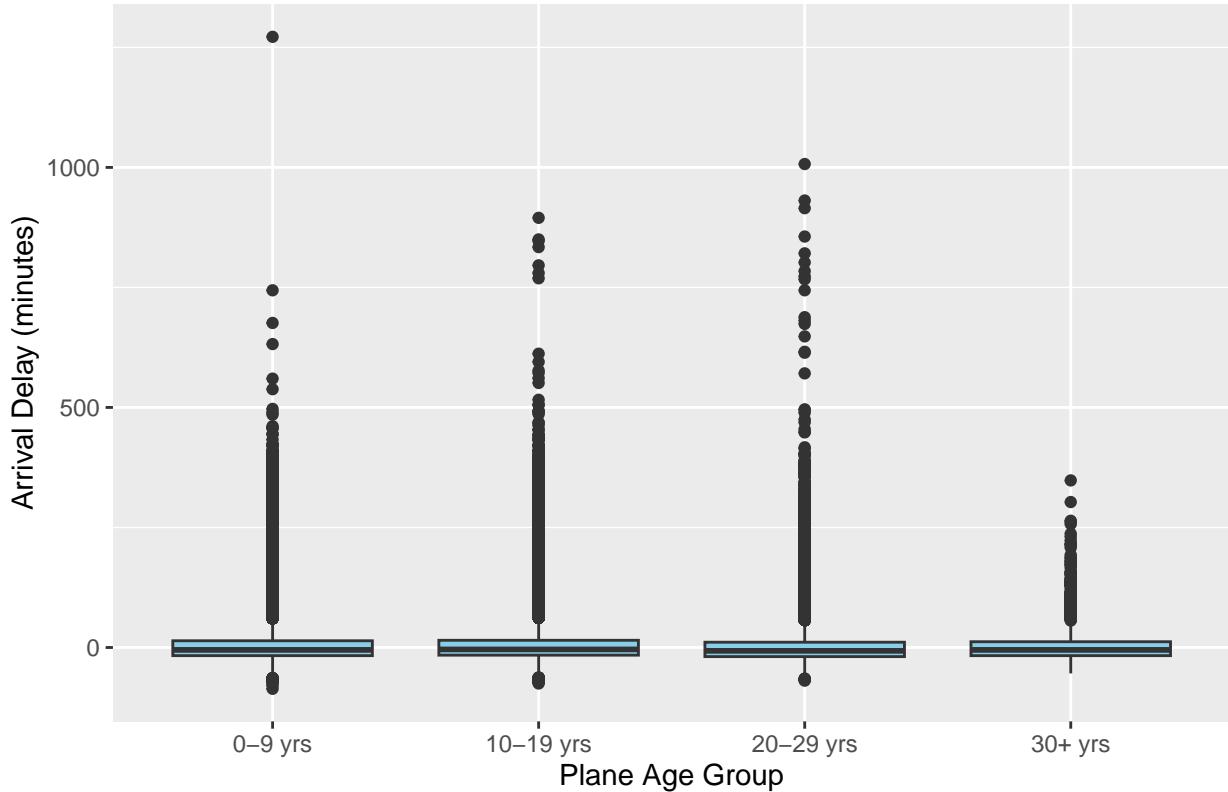
```

## # 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>

### Arrival Delay by Plane Age Group



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

```

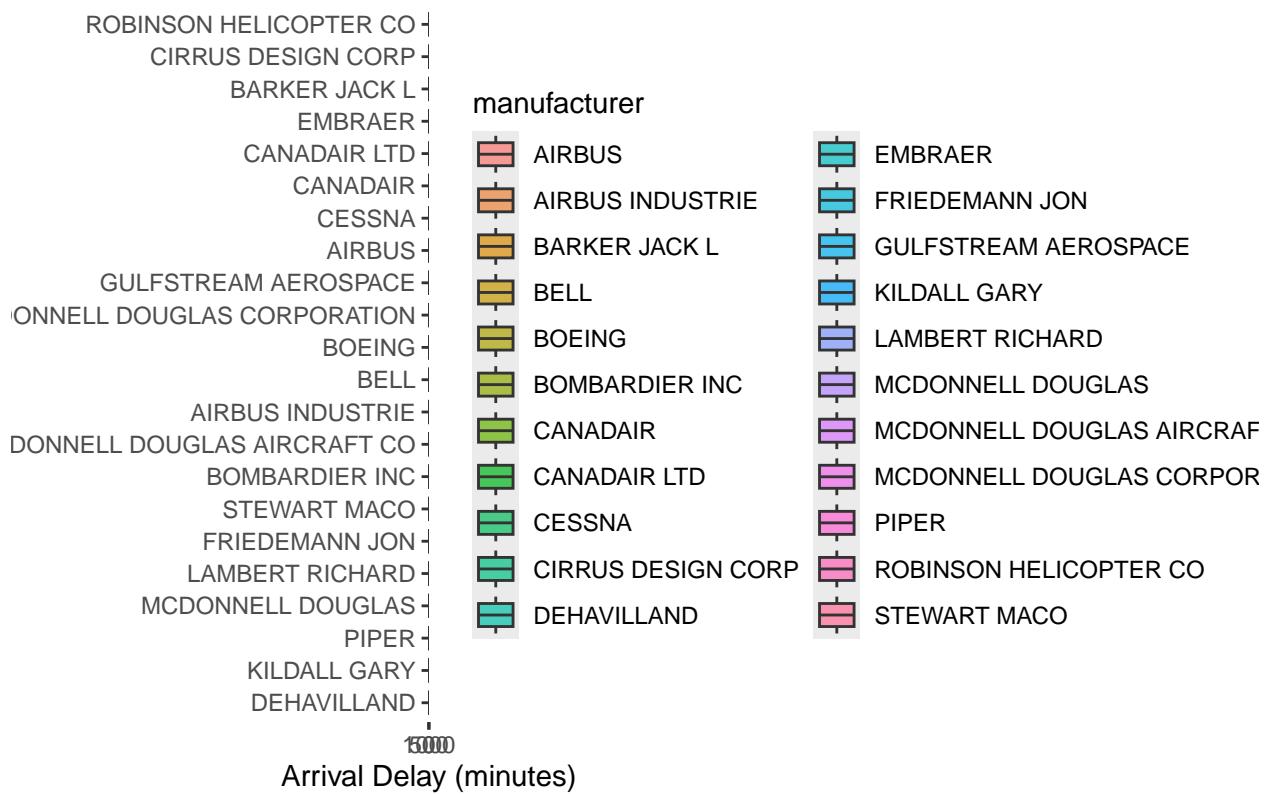
The Kruskal-Wallis test revealed a highly significant difference in arrival delays across plane age groups ( $p < 2.2e-16$ ), indicating that at least one group's delay distribution differs from the others. Post-hoc pairwise comparisons using Dunn's test with Bonferroni correction showed that planes 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)
```

Arrival Delay Distribution by Plane Manufacturer



```
# 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

# 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

# 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

if (kruskal_result$p.value < 0.05) {
  dunn_result <- dunn.test(flights_manufacturer$arr_delay, flights_manufacturer$manufacturer, method =
    sig_comparisons <- data.frame(
      comparison = dunn_result$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

  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 rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 3770.9154, df = 21, p-value = 0
##
##
## Comparison of x by group
## (Bonferroni)
## Col Mean-
## Row Mean | AIRBUS   AIRBUS I  BARKER J    BELL     BOEING   BOMBARDI
## -----+-----
## AIRBUS I | 9.931131
##           | 0.0000*
##           |
## BARKER J | -2.525521 -3.585438
##           | 1.0000   0.0389
##           |
## BELL    | 1.203695  0.669759  2.212547
##           | 1.0000   1.0000   1.0000
##           |
## BOEING  | 12.65471   0.975794  3.684397 -0.623082
##           | 0.0000*   1.0000   0.0265   1.0000
##           |
## BOMBARDI | 11.68778   2.756355  3.919185 -0.498747  2.240740
##           | 0.0000*   0.6751   0.0103*   1.0000   1.0000
##           |
## CANADAIR | -6.410582 -8.948789 -0.123357 -2.490554 -9.256539 -9.683654
##           | 0.0000*   0.0000*   1.0000   1.0000   0.0000*   0.0000*
##           |
## CANADAIR | -1.781623 -2.429688 -0.203980 -2.063333 -2.488113 -2.635713
##           | 1.0000   1.0000   1.0000   1.0000   1.0000   0.9697
##           |
## CESSNA   | -1.650045 -3.315613  1.250314 -1.651213 -3.475462 -3.835224
##           | 1.0000   0.1056   1.0000   1.0000   0.0589   0.0145*
##           |
## CIRRUS D | -2.501401 -3.640381  0.145408 -2.152896 -3.747300 -3.998451
##           | 1.0000   0.0314   1.0000   1.0000   0.0206*  0.0074*
##           |
## DEHAVILL | 3.309480  2.783866  4.090625  1.515670  2.738733  2.614826
##           | 0.1080   0.6204   0.0050*  1.0000   0.7124   1.0000

```

##								
##	EMBRAER	-31.90230	-41.17496	-0.537743	-2.744796	-50.63987	-39.18125	
##		0.0000*	0.0000*	1.0000	0.6993	0.0000*	0.0000*	
##								
##	FRIEDEMA	0.149903	-0.353436	1.218851	-0.717146	-0.397816	-0.514335	
##		1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
##								
##	GULFSTRE	-1.424144	-2.879936	1.207840	-1.621899	-3.016807	-3.336904	
##		1.0000	0.4594	1.0000	1.0000	0.2950	0.0978	
##								
##	KILDALL	2.757056	2.276466	3.556354	1.245048	2.234952	2.122105	
##		0.6736	1.0000	0.0434	1.0000	1.0000	1.0000	
##								
##	LAMBERT	0.991503	0.501741	1.960196	-0.083066	0.458850	0.344931	
##		1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
##								
##	MCDONNEL	17.38324	13.29676	6.924364	1.102706	13.23871	11.77878	
##		0.0000*	0.0000*	0.0000*	1.0000	0.0000*	0.0000*	
##								
##	MCDONNEL	8.613098	2.774816	4.054407	-0.410345	2.380882	0.901520	
##		0.0000*	0.6379	0.0058*	1.0000	1.0000	1.0000	
##								
##	MCDONNEL	2.534726	0.188565	3.360667	-0.611980	-0.017347	-0.556644	
##		1.0000	1.0000	0.0898	1.0000	1.0000	1.0000	
##								
##	PIPER	4.547819	3.713007	5.146813	1.439755	3.643210	3.443095	
##		0.0006*	0.0237*	0.0000*	1.0000	0.0311	0.0664	
##								
##	ROBINSON	-2.003523	-3.126982	0.463251	-1.948603	-3.231432	-3.481042	
##		1.0000	0.2040	1.0000	1.0000	0.1423	0.0577	
##								
##	STEWART	1.799910	1.310077	2.694919	0.513029	1.267453	1.153014	
##		1.0000	1.0000	0.8132	1.0000	1.0000	1.0000	
##	Col Mean	-						
##	Row Mean	CANADAIR	CANADAIR	CESSNA	CIRRUS D	DEHAVILL	EMBRAER	
##								
##	CANADAIR	-0.152935						
##		1.0000						
##								
##	CESSNA	2.139120	1.064927					
##		1.0000	1.0000					
##								
##	CIRRUS D	0.326588	0.313165	-1.138783				
##		1.0000	1.0000	1.0000				
##								
##	DEHAVILL	4.536502	3.695902	3.656535	4.056727			
##		0.0007*	0.0253	0.0295	0.0057*			
##								
##	EMBRAER	-0.979205	-0.089631	-3.171492	-0.791015	-4.826300		
##		1.0000	1.0000	0.1752	1.0000	0.0002*		
##								
##	FRIEDEMA	1.385788	1.211819	0.620677	1.148166	-2.182791	1.602771	
##		1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
##								

## GULFSTRE	1.950500	1.050907	0.013166	1.096731	-3.600436	2.786985
##	1.0000	1.0000	1.0000	1.0000	0.0367	0.6145
##						
## KILDALL	3.896446	3.278208	3.109032	3.515032	-0.198882	4.143972
##	0.0113*	0.1207	0.2168	0.0508	1.0000	0.0039*
##						
## LAMBERT	2.180970	1.866283	1.417459	1.899699	-1.532327	2.405077
##	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
##						
## MCDONNEL	15.09122	4.542870	8.296791	7.203749	-1.024365	29.24464
##	0.0000*	0.0006*	0.0000*	0.0000*	1.0000	0.0000*
##						
## MCDONNEL	9.591538	2.732172	4.014678	4.139773	-2.522928	25.90790
##	0.0000*	0.7267	0.0069*	0.0040*	1.0000	0.0000*
##						
## MCDONNEL	6.289239	2.396971	2.838717	3.374760	-2.678874	9.335853
##	0.0000*	1.0000	0.5232	0.0853	0.8532	0.0000*
##						
## PIPER	6.328748	4.203350	4.818408	5.158643	-0.376350	6.957186
##	0.0000*	0.0030*	0.0002*	0.0000*	1.0000	0.0000*
##						
## ROBINSON	0.748966	0.543767	-0.742171	0.330825	-3.850482	1.243623
##	1.0000	1.0000	1.0000	1.0000	0.0136*	1.0000
##						
## STEWART	2.975839	2.511849	2.193966	2.643250	-0.940646	3.213603
##	0.3375	1.0000	1.0000	0.9484	1.0000	0.1514
## Col Mean-						
## Row Mean	FRIEDEMA	GULFSTRE	KILDALL	LAMBERT	MCDONNEL	MCDONNEL
##						
## GULFSTRE	-0.607105					
##	1.0000					
##						
## KILDALL	1.892094	3.067257				
##	1.0000	0.2495				
##						
## LAMBERT	0.606492	1.394767	-1.274542			
##	1.0000	1.0000	1.0000			
##						
## MCDONNEL	2.019089	7.344407	-0.670132	1.124655		
##	1.0000	0.0000*	1.0000	1.0000		
##						
## MCDONNEL	0.595487	3.512714	-2.039462	-0.264204	-9.930511	
##	1.0000	0.0512	1.0000	1.0000	0.0000*	
##						
## MCDONNEL	0.385683	2.566185	-2.194748	-0.453073	-6.721640	-0.897308
##	1.0000	1.0000	1.0000	1.0000	0.0000*	1.0000
##						
## PIPER	2.223891	4.666325	-0.118841	1.449297	0.915747	3.287409
##	1.0000	0.0004*	1.0000	1.0000	1.0000	0.1168
##						
## ROBINSON	-0.956128	-0.719028	-3.325453	-1.710445	-6.661752	-3.626467
##	1.0000	1.0000	0.1020	1.0000	0.0000*	0.0332
##						
## STEWART	1.186262	2.161880	-0.708115	0.571952	-0.321318	1.070653

```

##          | 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
## Col Mean-|
## Row Mean | MCDONNEL PIPER ROBINSON
## -----
##    PIPER | 3.446159
##           | 0.0657
##           |
## ROBINSON | -2.916703 -4.856862
##           | 0.4086 0.0001*
##           |
## STEWART  | 1.245277 -0.751627 2.452362
##           | 1.0000 1.0000 1.0000
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
## 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 2.865e-221
## EMBRAER - MCDONNELL DOUGLAS 6.067e-186

```

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.

## **Conclusion**

### **Member Contributions**

#### **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.