final_project_draft

Omer Shasman

5/3/2022

Introduction

On Time Performance analysis of an airline network - This is an important metric for the airline which are delayed while the aircraft arrives at the gate. There are multiple reasons which contribute to the variation in OTP. An analysis of the OTP data breaking it down into its individual components namely different delay and historical delays can provide insights into how the OTP for an airline can be managed by operational/process changes. The Department of Transport releases the flight level, On Time Performance data. This dataset also has various other factors which affect the Arrival Delay of a flight. An exploratory analysis of this data with the Arrival Delay as the response variable analyzed against different dimensions provided in the dataset can reveal several insights to improve the OTP of an Airline.

What

As part of my Final Project, I am planning to use a subset of OTP data to perform analysis of delays on actual file arrivals focusing on one particular Station and Airline. Since airline operations are very complex, the arrival delays itself can be due to varying factors, like weather delay, carrier delays, security delays, Late aircraft delay…etc or any combinations of any of these in general. My focus is only on 2 types of delays so that I can minimize the complexities in data structures and limit any repeating processes or steps, and rather focus on how to explore data and do analysis/inference with few variables. Hence I will be considering only 5 years data ranging from year 2014 till 2018 and two types of delays "Weather Delays" and "Carrier Delays"

Why

I thought airline is an interesting business with lot of complex operation/data and business itself is most of us are familiar with. Also, with the time constraint we have, there were few sites like given below which gives a head start for this project. Kaggle and DOT On-Time_performance

This data is presented as yearly file in csv format, I have to merge different years data using dply bind command to append rows at the end and build one file.

How

Use RMarkdown and explore Rfunctions that can integrate some of the topics we learned in the class for flight arrival delay analysis. The following are steps which will be followed as part of the project

- Load data into R Markdown using R chunks
- Merge Data using ddplyr

- Filter/melt/massage data using Tidy Data approach
- Use sampling strategy for identifying sample observations.
- Use Stats function to determine mean, median, IQR,...
- Regression - Find any co-relation between total flights arriving at a particular airport and delays to identify if it's the airport operational/capacity issue or not.
- Cloud hosted data and R Markdown interface. Pull data from AWS S3 buckets than loading from local machine.

Body

This project perform analysis of flight arrival delays focusing on one particular Station and Airline. Since airline delays are unavoidable there is always a chance that a flight will be delayed. I think this analysis can be used to further study on why a particular delay happens and if the process/schedule/operations can be enhanced or refined to minimize delays and adjust future flight schedules.

Packages Required

```
library(knitr) ##for printing tables in R Markdown
library(dplyr) ##for data munging
```

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

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

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

```
library(ggplot2) ## for charts
library(infer) ## for rep_sample_n used for clustered sampling
```

```
library(readr)
flight.data.y2014 <- read_csv("Data/2014.csv") %>%
  filter(OP_CARRIER == 'DL') %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY, WEATHER_DELAY, CARRIER_DELAY, AIR_TIME, CRS_ARR_TIME))
# head(flight.data.y2014)

flight.data.y2015 <- read_csv("Data/2015.csv") %>%
  filter(OP_CARRIER == 'DL') %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY, WEATHER_DELAY, CARRIER_DELAY, AIR_TIME, CRS_ARR_TIME))
# head(flight.data.y2015)

flight.data.y2016 <- read_csv("Data/2016.csv") %>%
  filter(OP_CARRIER == 'DL') %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY, WEATHER_DELAY, CARRIER_DELAY, AIR_TIME, CRS_ARR_TIME))
```

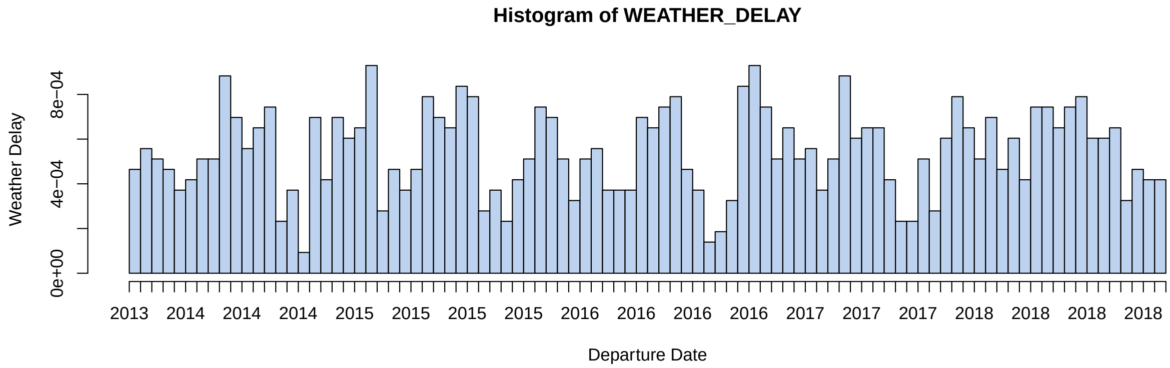
```r
# head(flight.data.y2016)

flight.data.y2017 <- read_csv("Data/2017.csv") %>%
  filter(OP_CARRIER == 'DL') %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY, WEATHER_DELAY, CARRIER_DELAY, AIR_TIME, CRS_ARR_TIME))
# head(flight.data.y2017)

flight.data.y2018 <- read_csv("Data/2018.csv") %>%
  filter(OP_CARRIER == 'DL') %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY, WEATHER_DELAY, CARRIER_DELAY, AIR_TIME, CRS_ARR_TIME))
  # head(flight.data.y2018)

# Since this is a large dataset, sampling/manipualting on all the observations
# is throwing memory error in my machine. So for ease of processing, I am
# considering a subset of data with arrival station as MSP.
```

```r
flight.data.y2014.y2018 <-
  dplyr::bind_rows(flight.data.y2014,
                   flight.data.y2015,
                   flight.data.y2016,
                   flight.data.y2017,
                   flight.data.y2018,
                   )
dim(flight.data.y2014.y2018)
```

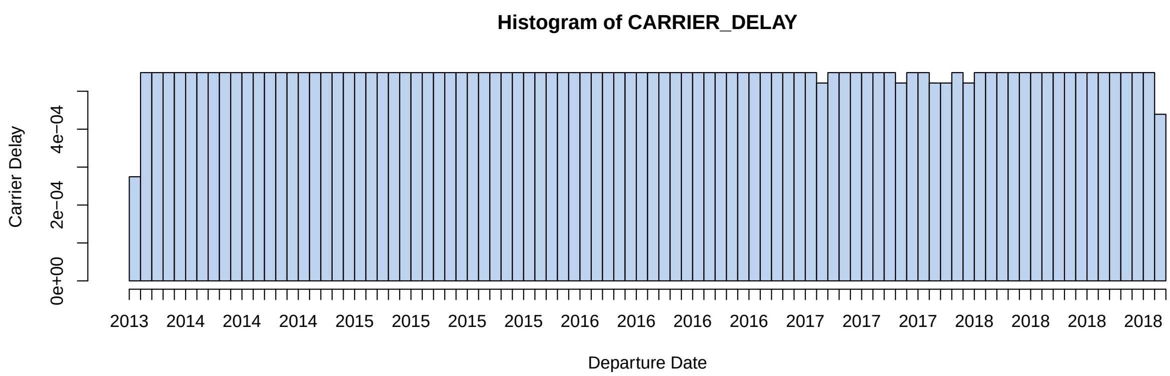
```
## [1] 4471845       6
```

```r
flight.data.y2014.y2018$WEATHER_DELAY <-
  flight.data.y2014.y2018$WEATHER_DELAY %>%
  replace(is.na(.), 0)

flight.data.y2014.y2018$CARRIER_DELAY <-
  flight.data.y2014.y2018$CARRIER_DELAY %>%
  replace(is.na(.), 0)

flight.data.y2014.y2018$AIR_TIME <-
  flight.data.y2014.y2018$AIR_TIME %>%
  replace(is.na(.), 0)
```

```r
flight.data.y2014.y2018.wd <- flight.data.y2014.y2018 %>%
  filter(WEATHER_DELAY > 0) %>%
  filter(DEST == 'MSP') %>%
  group_by(FL_DATE) %>%
  summarize(countDelay = n())

hist(flight.data.y2014.y2018.wd$FL_DATE,
    flight.data.y2014.y2018.wd$countDelay,
     main = paste("Histogram of WEATHER_DELAY" ),
     xlab = 'Departure Date',
     ylab = 'Weather Delay',
     breaks = 100,
     col='lightsteelblue2')
```




```
flight.data.y2014.y2018.cd <- flight.data.y2014.y2018 %>%
  filter(CARRIER_DELAY > 0) %>%
  filter(DEST == 'MSP') %>%
  group_by(FL_DATE) %>%
  summarize(countDelay = n())

hist(flight.data.y2014.y2018.cd$FL_DATE,
    flight.data.y2014.y2018.cd$countDelay,
     main = paste("Histogram of CARRIER_DELAY" ),
     xlab = 'Departure Date',
     ylab = 'Carrier Delay',
     breaks = 100,
     col='lightsteelblue2')
```




```
summary(flight.data.y2014.y2018.wd)
```

```
##     FL_DATE            countDelay
##  Min.   :2014-01-01   Min.   : 1.000
##  1st Qu.:2015-04-08   1st Qu.: 1.000
##  Median :2016-07-18   Median : 2.000
##  Mean   :2016-07-09   Mean   : 2.812
##  3rd Qu.:2017-10-27   3rd Qu.: 3.000
##  Max.   :2018-12-31   Max.   :24.000
```

```
summary(flight.data.y2014.y2018.cd)
```

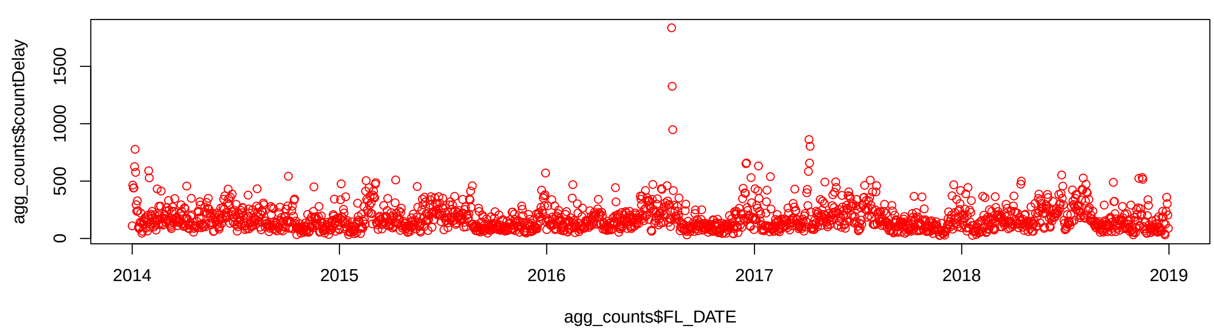
```
##     FL_DATE              countDelay    
##  Min.   :2014-01-01   Min.   :  1.00  
##  1st Qu.:2015-04-01   1st Qu.:  6.00  
##  Median :2016-06-29   Median :  9.00  
##  Mean   :2016-06-30   Mean   : 10.97  
##  3rd Qu.:2017-09-29   3rd Qu.: 14.00  
##  Max.   :2018-12-31   Max.   :137.00
```

Below I am finding if carrier delays surged during a particular time of a day or not.

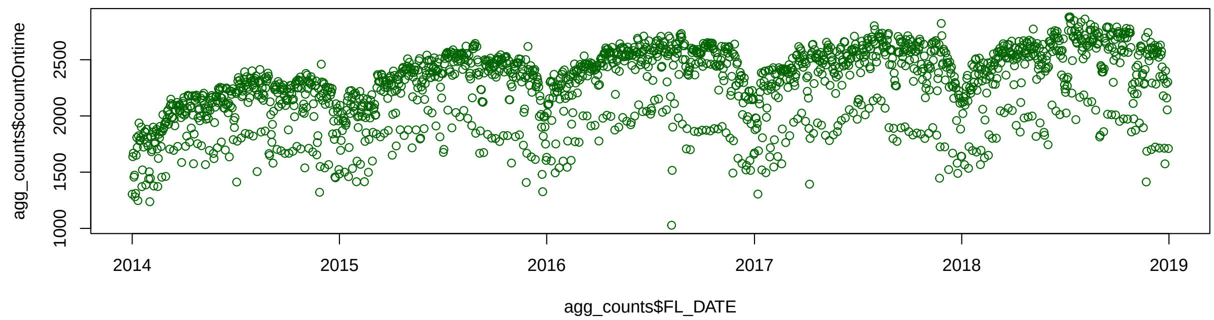
```
flight.data.y2014.y2018$year <- format(flight.data.y2014.y2018$FL_DATE, "%Y")
flight.data.y2014.y2018
```

```
## # A tibble: 4,471,845 x 7
##    FL_DATE    DEST  CARRIER_DELAY WEATHER_DELAY AIR_TIME CRS_ARR_TIME year 
##    <date>     <chr>         <dbl>         <dbl>    <dbl>        <dbl> <chr>
##  1 2014-01-01 MIA              62             0       85          925 2014 
##  2 2014-01-01 MEM               0             0       60         2052 2014 
##  3 2014-01-01 ATL               0             0      171         1521 2014 
##  4 2014-01-01 SLC               0             0      229         1900 2014 
##  5 2014-01-01 JFK               0             0      276         1650 2014 
##  6 2014-01-01 DTW               0             0      233         2059 2014 
##  7 2014-01-01 LAX               0            16      269         1720 2014 
##  8 2014-01-01 DTW              34             0      128         1651 2014 
##  9 2014-01-01 ATL               0             0      203         1844 2014 
## 10 2014-01-01 ATL               0             0       61         1500 2014 
## # ... with 4,471,835 more rows
```

```
agg_counts <- aggregate(cbind(countDelay=CARRIER_DELAY>0,
                              countOntime=CARRIER_DELAY==0) ~ FL_DATE,
                        data = flight.data.y2014.y2018, FUN = sum,
                        na.rm = TRUE)

plot(agg_counts$FL_DATE, agg_counts$countDelay, col = "red")
```



```r
plot(agg_counts$FL_DATE, agg_counts$countOntime, col = "darkgreen")
```




```r
simple.sampling <- dplyr::sample_n(flight.data.y2014.y2018, 1000, replace=FALSE)
# View(simple.sampling)
simple.sampling
```

```
## # A tibble: 1,000 x 7
##    FL_DATE    DEST  CARRIER_DELAY WEATHER_DELAY AIR_TIME CRS_ARR_TIME year
##    <date>     <chr>         <dbl>         <dbl>    <dbl>        <dbl> <chr>
##  1 2017-09-12 ATL               0             0       61          815 2017
##  2 2017-11-18 LAX               8             0      325         1006 2017
##  3 2016-05-06 ANC               0             0      187         1209 2016
##  4 2017-01-21 ATL              25             0       85         1602 2017
##  5 2016-11-29 RDU               0             0       56         1348 2016
##  6 2017-08-21 CHA               0             0       27         1719 2017
##  7 2016-01-05 BHM               0             0       30         1541 2016
##  8 2016-03-19 TPA               0             0       64          850 2016
##  9 2018-08-08 BOS               0             0       54         1611 2018
## 10 2016-10-28 TLH               0             0       42         1802 2016
## # ... with 990 more rows
```

```r
flight.data.y2014.y2018
```

```
## # A tibble: 4,471,845 x 7
##    FL_DATE    DEST  CARRIER_DELAY WEATHER_DELAY AIR_TIME CRS_ARR_TIME year
##    <date>     <chr>         <dbl>         <dbl>    <dbl>        <dbl> <chr>
##  1 2014-01-01 MIA              62             0       85          925 2014
##  2 2014-01-01 MEM               0             0       60         2052 2014
##  3 2014-01-01 ATL               0             0      171         1521 2014
##  4 2014-01-01 SLC               0             0      229         1900 2014
##  5 2014-01-01 JFK               0             0      276         1650 2014
##  6 2014-01-01 DTW               0             0      233         2059 2014
##  7 2014-01-01 LAX               0            16      269         1720 2014
##  8 2014-01-01 DTW              34             0      128         1651 2014
##  9 2014-01-01 ATL               0             0      203         1844 2014
## 10 2014-01-01 ATL               0             0       61         1500 2014
## # ... with 4,471,835 more rows
```

```r
strata.MSP <- flight.data.y2014.y2018[flight.data.y2014.y2018$DEST %in% 'MSP',]
strata.ATL <- flight.data.y2014.y2018[flight.data.y2014.y2018$DEST %in% 'ATL',]
strata.JFL <- flight.data.y2014.y2018[flight.data.y2014.y2018$DEST %in% 'JFK',]
strata.LAX <- flight.data.y2014.y2018[flight.data.y2014.y2018$DEST %in% 'LAX',]

stratified.sampling <- dplyr::sample_n((strata.MSP), 100000, replace=FALSE)
dim(stratified.sampling)
```

```
## [1] 100000      7
```

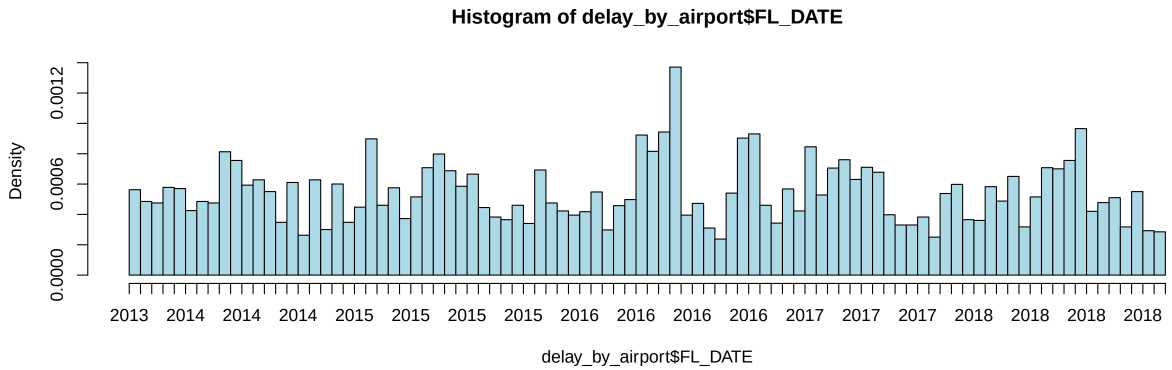
```r
#randomly choose 10 groups out of the n
clusters <-
  sample(unique(flight.data.y2014.y2018$DEST), size=10, replace=FALSE)

#define sample as all obervations belonging to one of the 10 airpots
clustered_by_airport <-
  flight.data.y2014.y2018[flight.data.y2014.y2018$DEST %in% clusters, ]

#view how many observations came from each airport codes
table(clustered_by_airport$DEST)
```

```
##
##   BOS   DAL   EWR   GNV   GSO   JAC   LEX   ONT   PHF   SHV
## 75725  6849 27172  1583 11500  4633  5156  1355  1032   823
```

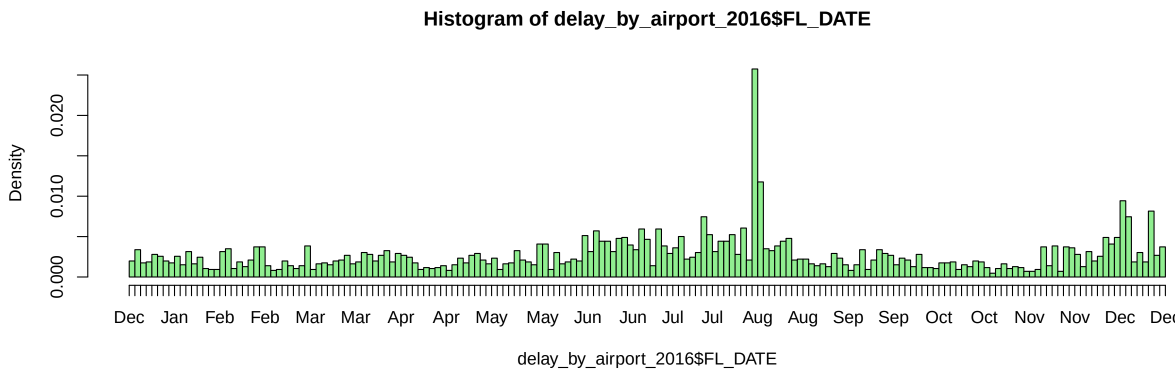
```r
head(clustered_by_airport)
```

```
## # A tibble: 6 x 7
##   FL_DATE    DEST  CARRIER_DELAY WEATHER_DELAY AIR_TIME CRS_ARR_TIME year
##   <date>     <chr>         <dbl>         <dbl>    <dbl>        <dbl> <chr>
## 1 2014-01-01 BOS              67             0       98         1130 2014
## 2 2014-01-01 ONT               0             0      255         2150 2014
## 3 2014-01-01 EWR               0             0       93         1059 2014
## 4 2014-01-01 BOS               0             0      121         1342 2014
## 5 2014-01-01 BOS               0             0      122         2111 2014
## 6 2014-01-01 BOS               0             0      125         2313 2014
```

```r
delay_by_airport <-
  strata.MSP %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY)) %>%
  filter(CARRIER_DELAY > 0)

hist(delay_by_airport$FL_DATE,
     delay_by_airport$CARRIER_DELAY,
     breaks = 100,
     col = "lightblue")
```



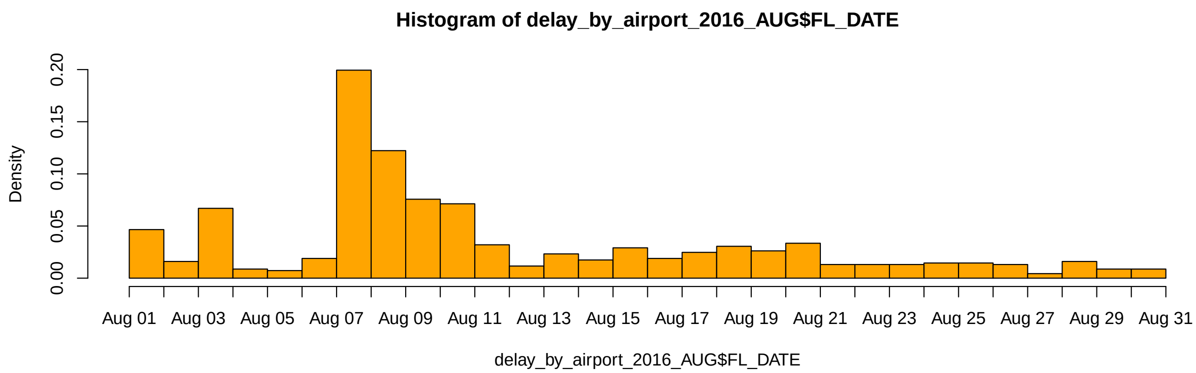

```
delay_by_airport_2016 <-
  delay_by_airport

delay_by_airport_2016$YEAR <-
  format(delay_by_airport$FL_DATE, "%Y")

delay_by_airport_2016 <-
  delay_by_airport_2016 %>%
  filter(YEAR == '2016')

delay_by_airport_2016$MONTH <-
  format(delay_by_airport_2016$FL_DATE, "%m")

hist(delay_by_airport_2016$FL_DATE,
     delay_by_airport_2016$CARRIER_DELAY,
     breaks =200,
     col = "lightgreen")
```



From the above plotting we can visually come to conclusion that August2016 has an increase in carrier delay. The following RChunk code will plot by Days in Aug2016 to determine which day have an surge in Carrier Delays.

```
delay_by_airport_2016_AUG <-
  delay_by_airport_2016

delay_by_airport_2016_AUG <-
```

```r
  delay_by_airport_2016_AUG %>%
  filter(MONTH == '08')

delay_by_airport_2016_AUG$DAY <-
  format(delay_by_airport_2016_AUG$FL_DATE,
         "%d")

hist(delay_by_airport_2016_AUG$FL_DATE,
     delay_by_airport_2016_AUG$CARRIER_DELAY,
     breaks =40,
     col = "orange")
```



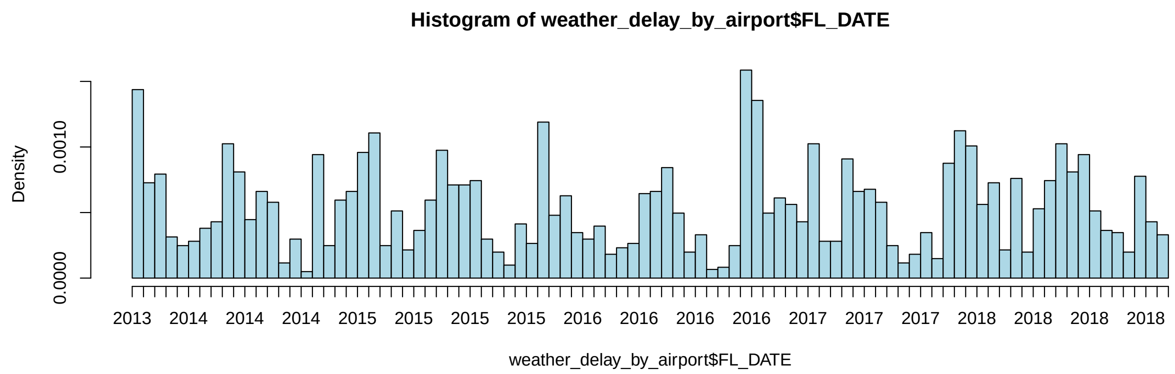

```r
summary(delay_by_airport_2016_AUG$CARRIER_DELAY)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.00    9.00   22.00   70.74   76.00 1167.00
```

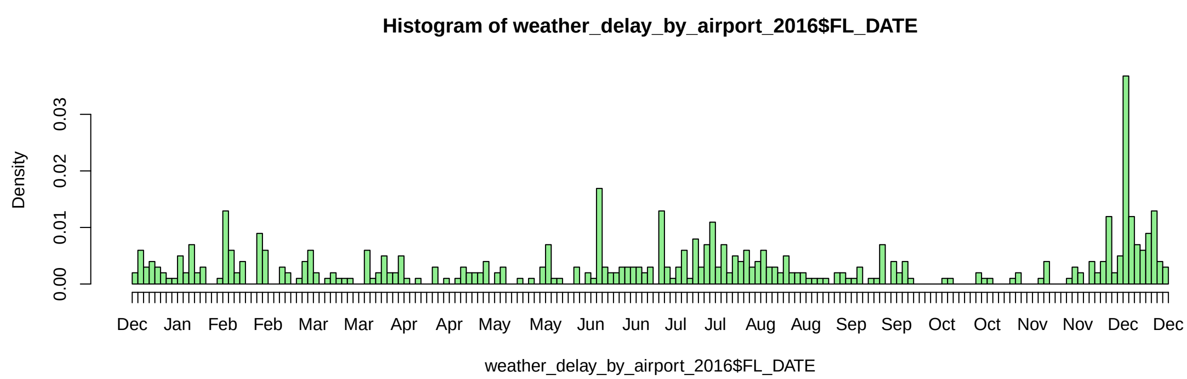
From the news archives from that day we can conclude that there was a system outage which caused massive delay/cancellations and the plotting above matches with that conclusion. Delta System Outage - Aug2016

WEATHER DELAY

```r
# First select the sub-vectors which contains only he coulums we are interested in
weather_delay_by_airport <-
  strata.MSP %>%
  select(c(FL_DATE,
           WEATHER_DELAY)) %>%
  filter(WEATHER_DELAY > 0
         )

hist(weather_delay_by_airport$FL_DATE,
     weather_delay_by_airport$WEATHER_DELAY,
     breaks = 100,
     col = "lightblue")
```




```
# In 2016 we have a increase in delay. Let's find out month by analysis to approximate on which month i
weather_delay_by_airport_2016 <-
  weather_delay_by_airport

weather_delay_by_airport_2016$YEAR <-
  format(weather_delay_by_airport$FL_DATE, "%Y")

weather_delay_by_airport_2016 <-
  weather_delay_by_airport_2016 %>%
  filter(YEAR == '2016')

weather_delay_by_airport_2016$MONTH <-
  format(weather_delay_by_airport_2016$FL_DATE, "%m")

hist(weather_delay_by_airport_2016$FL_DATE,
     weather_delay_by_airport_2016$WEATHER_DELAY,
     breaks =200, col = "lightgreen")
```

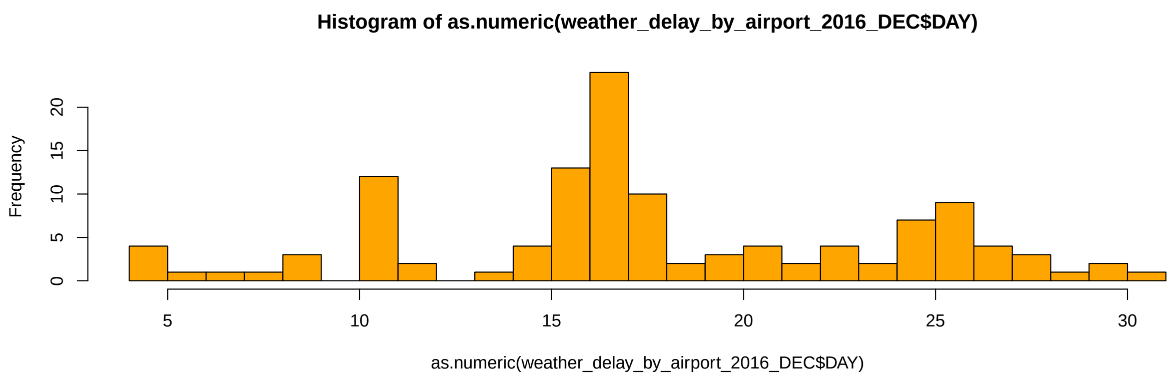


The above histogram shows that whether delays are massive in December month in MSP airport. This could be explained by winter storms related delays.

```
weather_delay_by_airport_2016_DEC <-
  weather_delay_by_airport_2016 %>%
  filter(MONTH == '12')

weather_delay_by_airport_2016_DEC$DAY <-
```

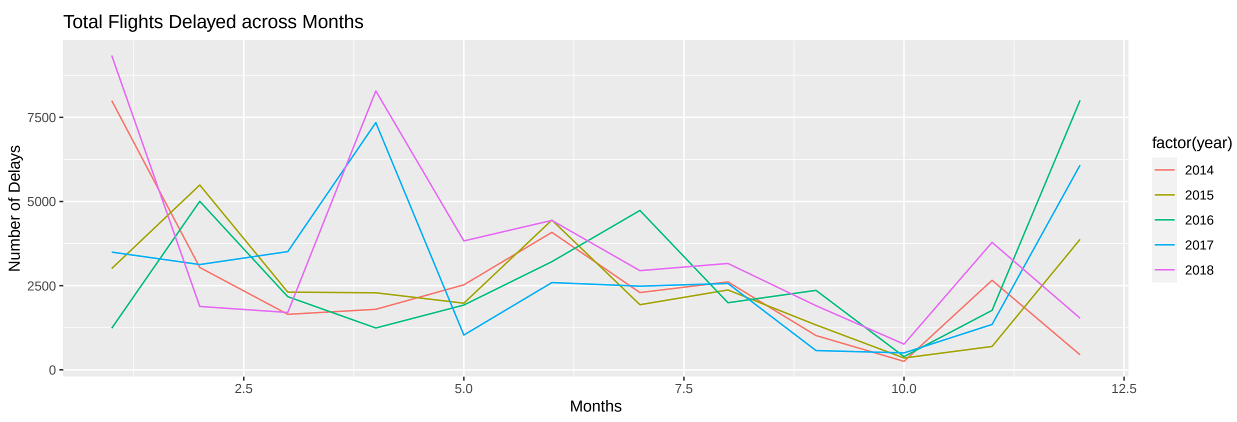
```
  format(weather_delay_by_airport_2016_DEC$FL_DATE, "%d")

hist(as.numeric(weather_delay_by_airport_2016_DEC$DAY),
     weather_delay_by_airport_2016_DEC$WEATHER_DELAY,
     breaks =30, col = "orange")
```

```
## Warning in if (freq) x$counts else x$density: the condition has length > 1 and
## only the first element will be used

## Warning in if (!freq) "Density" else "Frequency": the condition has length > 1
## and only the first element will be used
```




```
airport = c('MSP')

weather_delay_by_airport_2014.gorupby <-
  flight.data.y2014 %>%
  select(c(FL_DATE, DEST, WEATHER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(WEATHER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(WEATHER_DELAY)) %>%
  mutate(year=2014)

weather_delay_by_airport_2015.gorupby <-
  flight.data.y2015 %>%
  select(c(FL_DATE, DEST, WEATHER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(WEATHER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(WEATHER_DELAY)) %>%
  mutate(year=2015)

weather_delay_by_airport_2016.gorupby <-
  flight.data.y2016 %>%
  select(c(FL_DATE, DEST, WEATHER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(WEATHER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
```

```
  summarize(total_delayed=sum(WEATHER_DELAY)) %>%
  mutate(year=2016)

weather_delay_by_airport_2017.gorupby <-
  flight.data.y2017 %>%
  select(c(FL_DATE, DEST, WEATHER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(WEATHER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(WEATHER_DELAY)) %>%
  mutate(year=2017)

weather_delay_by_airport_2018.gorupby <-
  flight.data.y2018 %>%
  select(c(FL_DATE, DEST, WEATHER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(WEATHER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(WEATHER_DELAY)) %>%
  mutate(year=2018)

month_Delay<-rbind(weather_delay_by_airport_2014.gorupby,
                   weather_delay_by_airport_2015.gorupby,
                   weather_delay_by_airport_2016.gorupby,
                   weather_delay_by_airport_2017.gorupby,
                   weather_delay_by_airport_2018.gorupby)

ggplot(month_Delay,
       aes(x = `as.numeric(format(FL_DATE, "%m"))`,
           y = total_delayed,
           color = factor(year), group = factor(year))) +
geom_line(linetype = 1) +
  labs(title="Total Flights Delayed across Months",y = 'Number of Delays',x = 'Months', fill='YEAR')
```

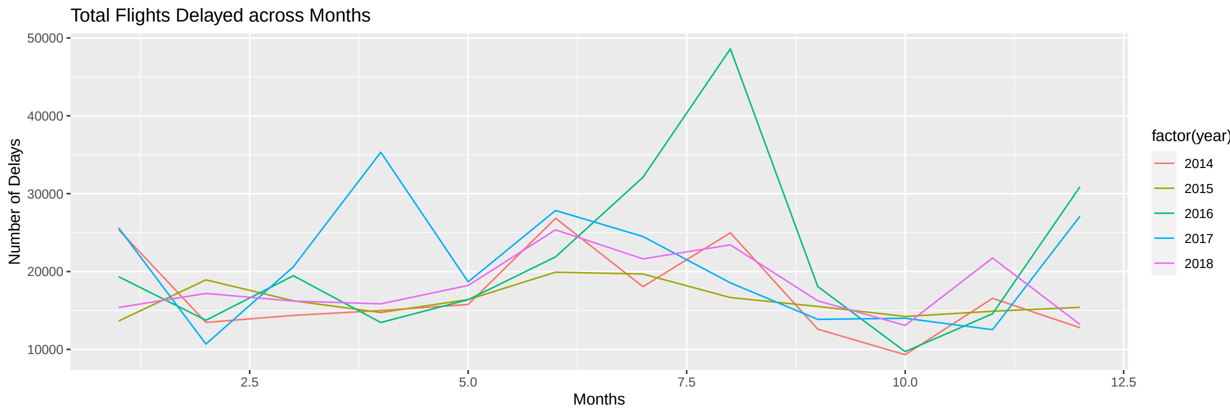


The above plot shows weather delays during 2014-2018 and it's clear that most of the weather related delays happens during year start/end.

```
airport = c('MSP')

weather_delay_by_airport_2014.gorupby <-
```

```
  flight.data.y2014 %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(CARRIER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(CARRIER_DELAY)) %>%
  mutate(year=2014)

weather_delay_by_airport_2015.gorupby <-
  flight.data.y2015 %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(CARRIER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(CARRIER_DELAY)) %>%
  mutate(year=2015)

weather_delay_by_airport_2016.gorupby <-
  flight.data.y2016 %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(CARRIER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(CARRIER_DELAY)) %>%
  mutate(year=2016)

weather_delay_by_airport_2017.gorupby <-
  flight.data.y2017 %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(CARRIER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(CARRIER_DELAY)) %>%
  mutate(year=2017)

weather_delay_by_airport_2018.gorupby <-
  flight.data.y2018 %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY)) %>%
  filter(DEST == airport) %>%
  filter(CARRIER_DELAY > 0) %>%
  group_by(as.numeric(format(FL_DATE, "%m"))) %>%
  summarize(total_delayed=sum(CARRIER_DELAY)) %>%
  mutate(year=2018)

month_Delay<-rbind(weather_delay_by_airport_2014.gorupby,
                   weather_delay_by_airport_2015.gorupby,
                   weather_delay_by_airport_2016.gorupby,
                   weather_delay_by_airport_2017.gorupby,
                   weather_delay_by_airport_2018.gorupby)

ggplot(month_Delay,
       aes(x = `as.numeric(format(FL_DATE, "%m"))`,
           y = total_delayed,
```

```
           color = factor(year), group = factor(year))) +
geom_line(linetype = 1) +
  labs(title="Total Flights Delayed across Months",y = 'Number of Delays',x = 'Months', fill='YEAR')
```




```
flight.data.y2018
```

```
## # A tibble: 949,283 x 6
##    FL_DATE    DEST  CARRIER_DELAY WEATHER_DELAY AIR_TIME CRS_ARR_TIME
##    <date>     <chr>         <dbl>         <dbl>    <dbl>        <dbl>
##  1 2018-01-01 TPA              NA            NA      158         2325
##  2 2018-01-01 JFK              NA            NA      218         1756
##  3 2018-01-01 SLC              NA            NA       83         1605
##  4 2018-01-01 LAX              NA            NA       85          750
##  5 2018-01-01 MCI              NA            NA       60         2138
##  6 2018-01-01 ATL               0             0       68         1523
##  7 2018-01-01 TPA              NA            NA       65         2015
##  8 2018-01-01 DTW              NA            NA      141         1453
##  9 2018-01-01 MCO              27             0       66         2131
## 10 2018-01-01 ATL              NA            NA       64         1004
## # ... with 949,273 more rows
```

```
carrier_delay_by_airport_2018.totalFlights <-
  flight.data.y2018 %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY)) %>%
  group_by(FL_DATE) %>% count() %>% mutate(year=2018)

carrier_delay_by_airport_2018.carrierDelay <-
  flight.data.y2018 %>%
  select(c(FL_DATE, DEST, CARRIER_DELAY)) %>%
  filter(CARRIER_DELAY >= 0) %>%
  group_by(FL_DATE) %>%
  summarize(total_delayed=sum(CARRIER_DELAY > 0)) %>%
  mutate(year=2018)

dataset <- bind_cols(carrier_delay_by_airport_2018.totalFlights,
                     carrier_delay_by_airport_2018.carrierDelay)
```

```
## New names:
```

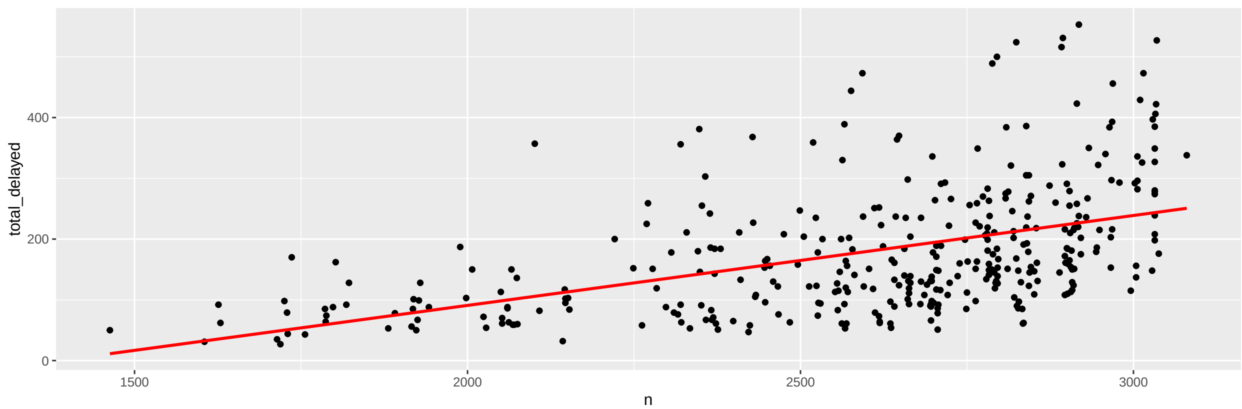
```
## * ‘FL_DATE‘ -> ‘FL_DATE...1‘
## * ‘year‘ -> ‘year...3‘
## * ‘FL_DATE‘ -> ‘FL_DATE...4‘
## * ‘year‘ -> ‘year...6‘
```

```
linear_model <- lm(total_delayed ~ n,
                   data=dataset)
summary(linear_model)
```

```
##
## Call:
## lm(formula = total_delayed ~ n, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -153.25  -65.22  -16.27   46.79  326.31
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -205.18138   37.26136  -5.507 6.94e-08 ***
## n              0.14800    0.01421  10.418  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.41 on 363 degrees of freedom
## Multiple R-squared:  0.2302, Adjusted R-squared:  0.2281
## F-statistic: 108.5 on 1 and 363 DF,  p-value: < 2.2e-16
```

```
ggplot(dataset, aes(x=n,
                    y=total_delayed)) +
  geom_point() +
  geom_smooth(method='lm', se=FALSE, col="red", size=1)
```

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



There appears to be have a linear relation between carrier delay and total flights. Since there is a linear relation between number of flights arrived and carrier delay, it could be due to airline related issue(like crew/pilot scheduling issue or some other operational issues when there is an increase in number of flights operated by the airline.)

Topics From Class

Topic 1:

R Markdown - I will be presenting the project in R Markdown and knit the file to a pdf document. Will be using R chunks to demonstrate and build the project components.

Topic 2:

GitHub - Will host the project in github repository for others to view my project components.

Topic 3:

Sampling strategies for an Observational study - Will be using sampling strategies - Simple random sampling, Stratified sampling, Cluster sampling and multistage sampling to group the data together by using different variables from the dataset and then use one of the sampling result to build topic#4 and 5.

Topic 4:

Detailing Summary statistics (Min. , 1st Qu., Median, Mean, 3rd Qu., Max.) of a variable and plotting graphs using ggplot2

Topic 5:

Regression (if an increase in number of schedules has any impact/variace on carrier delays).

Conclusion

I designed this project as a way to review some of the topics we learned in the class/homework/assignments to reinforce some topics learned and also as an opportunity to refer back some of the materials. Hence I thought of picking a variety of topics like sampling strategies, summary statistics, ANOVA and regressions will be the best approach and most I can get from this project. If I have more time, I would have included some more topics (like binom, dbinom, geom…etc distributions) and see if my dataset have variables that can fit these distributions. Given only a academic background in statistics almost almost 20 years ago, I think this subject has given me much learning experience in statistics and I appreciate how these topics are applicable to find solutions in reality.