IMT 573: Problem Set 1 - Exploring Data

Saurabh Sharma

Due: Tuesday, October 08, 2019

Collaborators:

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Server.

1. Download the `problemset1.Rmd` file from Canvas or save a copy to your local directory on RStudio Server. Open `problemset1.Rmd` in RStudio and supply your solutions to the assignment by editing `problemset1.Rmd`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures, and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do no need four different visualizations of the same pattern.
4. Collaboration on problem sets is fun and useful, and we encourage it, but each student must turn in an individual write-up in their own words as well as code/work that is their own. Regardless of whether you work with others, what you turn in must be your own work; this includes code and interpretation of results. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.
6. Remember partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. Students are *strongly* encouraged to attempt each question and to document their reasoning process even if they cannot find the correct answer. If you would like to include R code to show this process, but it does not run withouth errors you can do so with the `eval=FALSE` option as follows:

```
a + b # these object dont' exist
# if you run this on its own it with give an error
```

7. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the knitted PDF file to `Yps1_ourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries

install.packages("nycflights13",repos = "http://cran.us.r-project.org")
```

```
## package 'nycflights13' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Swati\AppData\Local\Temp\RtmpQR8IGI\downloaded_packages
```

```
library(tidyverse)
library(nycflights13)
```

Problem 1: Exploring the NYC Flights Data

In this problem set we will use the data on all flights that departed NYC (i.e. JFK, LGA or EWR) in 2013. You can find this data in the `nycflights13` R package.

```
# Load the nycflights13 library which includes data on all
# lights departing NYC
data(flights)
# Note the data itself is called flights, we will make it into a local df
# for readability
flights <- tbl_df(flights)
# Look at the help file for information about the data
# ?flights
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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(flights)
```

(a) Importing and Inspecting Data

Load the data and describe in a short paragraph how the data was collected and what each variable represents. Perform a basic inspection of the data and discuss what you find.

```
head(flights,6)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     1     1      517            515         2      830
## 2  2013     1     1      533            529         4      850
## 3  2013     1     1      542            540         2      923
## 4  2013     1     1      544            545        -1     1004
## 5  2013     1     1      554            600        -6      812
## 6  2013     1     1      554            558        -4      740
## # ... with 12 more variables: sched_arr_time <int>, 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>
```

```
tail(flights,6)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     9    30       NA           1842        NA       NA
## 2  2013     9    30       NA           1455        NA       NA
## 3  2013     9    30       NA           2200        NA       NA
## 4  2013     9    30       NA           1210        NA       NA
## 5  2013     9    30       NA           1159        NA       NA
## 6  2013     9    30       NA            840        NA       NA
## # ... with 12 more variables: sched_arr_time <int>, 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>
```

```
nrow(flights)
```

```
## [1] 336776
```

```
ncol(flights)
```

```
## [1] 19
```

```
summary(flights)
```

```
##       year          month             day           dep_time
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   :   1
##  1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907
##  Median :2013   Median : 7.000   Median :16.00   Median :1401
##  Mean   :2013   Mean   : 6.549   Mean   :15.71   Mean   :1349
##  3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744
##  Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2400
##                                                  NA's   :8255
##  sched_dep_time   dep_delay          arr_time    sched_arr_time
##  Min.   : 106   Min.   : -43.00   Min.   :   1   Min.   :   1
##  1st Qu.: 906   1st Qu.:  -5.00   1st Qu.:1104   1st Qu.:1124
##  Median :1359   Median :  -2.00   Median :1535   Median :1556
##  Mean   :1344   Mean   :  12.64   Mean   :1502   Mean   :1536
##  3rd Qu.:1729   3rd Qu.:  11.00   3rd Qu.:1940   3rd Qu.:1945
##  Max.   :2359   Max.   :1301.00   Max.   :2400   Max.   :2359
##                 NA's   :8255      NA's   :8713
##    arr_delay          carrier              flight       tailnum
##  Min.   : -86.000   Length:336776      Min.   :   1   Length:336776
##  1st Qu.: -17.000   Class :character   1st Qu.: 553   Class :character
##  Median :  -5.000   Mode  :character   Median :1496   Mode  :character
##  Mean   :   6.895                      Mean   :1972
##  3rd Qu.:  14.000                      3rd Qu.:3465
##  Max.   :1272.000                      Max.   :8500
##  NA's   :9430
##     origin              dest              air_time        distance
##  Length:336776      Length:336776      Min.   : 20.0   Min.   :  17
##  Class :character   Class :character   1st Qu.: 82.0   1st Qu.: 502
##  Mode  :character   Mode  :character   Median :129.0   Median : 872
##                                          Mean   :150.7   Mean   :1040
##                                          3rd Qu.:192.0   3rd Qu.:1389
##                                          Max.   :695.0   Max.   :4983
##                                          NA's   :9430
##       hour           minute        time_hour
##  Min.   : 1.00   Min.   : 0.00   Min.   :2013-01-01 05:00:00
##  1st Qu.: 9.00   1st Qu.: 8.00   1st Qu.:2013-04-04 13:00:00
##  Median :13.00   Median :29.00   Median :2013-07-03 10:00:00
##  Mean   :13.18   Mean   :26.23   Mean   :2013-07-03 05:22:54
##  3rd Qu.:17.00   3rd Qu.:44.00   3rd Qu.:2013-10-01 07:00:00
##  Max.   :23.00   Max.   :59.00   Max.   :2013-12-31 23:00:00
##
```

```
dim(flights)
```

```
## [1] 336776     19
```

#To get tge glimpse of data along with the data types and variable names

```
glimpse(flights)
```

```
## Observations: 336,776
## Variables: 19
## $ year           <int> 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,...
## $ day            <int> 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, 55...
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 60...
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2...
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 7...
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 7...
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -...
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV",...
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79...
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN...
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR"...
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL"...
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138...
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 94...
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5,...
## $ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013...
```

#To check the data type of the variables

```
sapply(flights,class)
```

```
## $year
## [1] "integer"
##
## $month
## [1] "integer"
##
## $day
## [1] "integer"
##
```

```
## $dep_time
## [1] "integer"
##
## $sched_dep_time
## [1] "integer"
##
## $dep_delay
## [1] "numeric"
##
## $arr_time
## [1] "integer"
##
## $sched_arr_time
## [1] "integer"
##
## $arr_delay
## [1] "numeric"
##
## $carrier
## [1] "character"
##
## $flight
## [1] "integer"
##
## $tailnum
## [1] "character"
##
## $origin
## [1] "character"
##
## $dest
## [1] "character"
##
## $air_time
## [1] "numeric"
##
## $distance
## [1] "numeric"
##
## $hour
## [1] "numeric"
##
## $minute
## [1] "numeric"
##
## $time_hour
## [1] "POSIXct" "POSIXt"
```

NUmber of rows: 336776 Numbre of columns: 19

On inspecting the data, the variables I have found out are as follows:

year (integer), month (integer),day (integer) : Day of Departure of the flight

dep_time (integer) : Departure Time (actual) sched_dep_time (integer) : scheduled departure time

dep_delay (numeric) : delay in departure

arr_time (integer) : Arrival time (actual)

sched_arr_time (integer) : scheduled arrival

arr_delay (numeric) : Arrival Delay

carrier(character) : Carrier information

flight (integer) : flight number

tailnum (character) : Tail number

origin (integer) : Origin

dest (character) : Final destination

air_time (numeric) : Time of flight in air

distance (numeric) : Distance between ports

hour (numeric) : Hour

minute (numeric) : Minute

time_hour (POSIXt) : Time in POSIXt format

The data is for the year 2013.

On using the summary function we also found that there are missing values for several columns such as dep_time,dep_delay,arr_time,arr_delay and air_time.

To find the origin airports

```
unique(flights$origin)
```

```
## [1] "EWR" "LGA" "JFK"
```

```
unique(flights$dest)
```

```
##   [1] "IAH" "MIA" "BQN" "ATL" "ORD" "FLL" "IAD" "MCO" "PBI" "TPA" "LAX"
##  [12] "SFO" "DFW" "BOS" "LAS" "MSP" "DTW" "RSW" "SJU" "PHX" "BWI" "CLT"
##  [23] "BUF" "DEN" "SNA" "MSY" "SLC" "XNA" "MKE" "SEA" "ROC" "SYR" "SRQ"
##  [34] "RDU" "CMH" "JAX" "CHS" "MEM" "PIT" "SAN" "DCA" "CLE" "STL" "MYR"
##  [45] "JAC" "MDW" "HNL" "BNA" "AUS" "BTV" "PHL" "STT" "EGE" "AVL" "PWM"
##  [56] "IND" "SAV" "CAK" "HOU" "LGB" "DAY" "ALB" "BDL" "MHT" "MSN" "GSO"
##  [67] "CVG" "BUR" "RIC" "GSP" "GRR" "MCI" "ORF" "SAT" "SDF" "PDX" "SJC"
##  [78] "OMA" "CRW" "OAK" "SMF" "TUL" "TYS" "OKC" "PVD" "DSM" "PSE" "BHM"
##  [89] "CAE" "HDN" "BZN" "MTJ" "EYW" "PSP" "ACK" "BGR" "ABQ" "ILM" "MVY"
## [100] "SBN" "LEX" "CHO" "TVC" "ANC" "LGA"
```

```
length(unique(flights$dest))
```

```
## [1] 105
```

Here we are looking at the flights dataset which has 19 columns and 336776 rows.There are 3 airports in the origin column. These are the NYC airports. Also, there are 105 destinations.Three airports in the NYC area are JFK,EWR,LGA.

(b) Formulating Questions

Consider the NYC flights data. Formulate two motivating questions you want to explore using this data. Describe why these questions are interesting and how you might go about answering them.

1. We need to find out the trend of flight delays. We can find out how different variables are associated with each other and cause the flight delays. 2.1 Is there any relation between the month of the year and flight getting delayed. For example a particular month experiencing more flight delays than the other months? 2.2 Is there a relation between hour of the day and the delay? 2.3 Is there a relation between distance between the destination and the delay.? 2.4 Is there a relation between the origin airport and the delay?

These questions are interesting as they will help us in finding the causes for the delay of flights. If we are able to find out the patterns from the data which point towards the causes for the delay, we will be able to reach to the root causes of these delays and probably it can help in eradicating those frequently occuring activities which cause delay. It will be interesting to see whether the factors as small as the hour of the day can cause a delay in the flight timings.

(c) Exploring Data

For each of the questions you proposed in Problem 1b, perform an exploratory data analysis designed to address the question. At a minimum, you should produce two visualizations (graphics or tables) related to each question. Be sure to describe what the visuals show and how they speak to your question of interest.

Analyzing the relation with respect to the NYC airports:

#month vs delay

```r
ggplot(data = flights, aes(x = factor(month),y=dep_delay)) + stat_summary(fun.y = "mean",
  geom = "bar")
```

Using the above plot, we tried to identify whether there is more delay in some months as compared to some other months. We found that the maximum delay is in months of JUNE and JULY. It may be due to various number of reasons. One of the reasons maybe there is more air traffic during these months as flight carriers might be more operational during this months due to some festivals or other probable reasons.

#Hour vs delay

```r
ggplot(data = flights, aes(x = factor(hour),y=dep_delay)) + stat_summary(fun.y = "mean",
  geom = "bar")
```

Using the above plot we tried we tried to indentify whether there is a relation between the hours of the day and delay in flight. We found out that the delay is moer in later part of the days as compared to the early mornings. It might be because more flights might be operational during the day in comparison to those in early morning.

```r
ggplot(flights, aes(x = factor(month), y = dep_delay)) +
  geom_boxplot()
```

#origin vs delay

```r
flights <- flights %>%
  mutate(delay_type = ifelse(dep_delay < 10, "on time", "delayed"))

  ggplot(data = flights, aes(x = origin, fill = delay_type)) +
  geom_bar()
```

In the above function, we have defined a new data delay_type which filters the departure delay times. Then we have plot the graph for the three airports of NYC according to our own criteria of on time and delayed. This plots shows the number of flights which were on time and which were delayed from a particular airport.

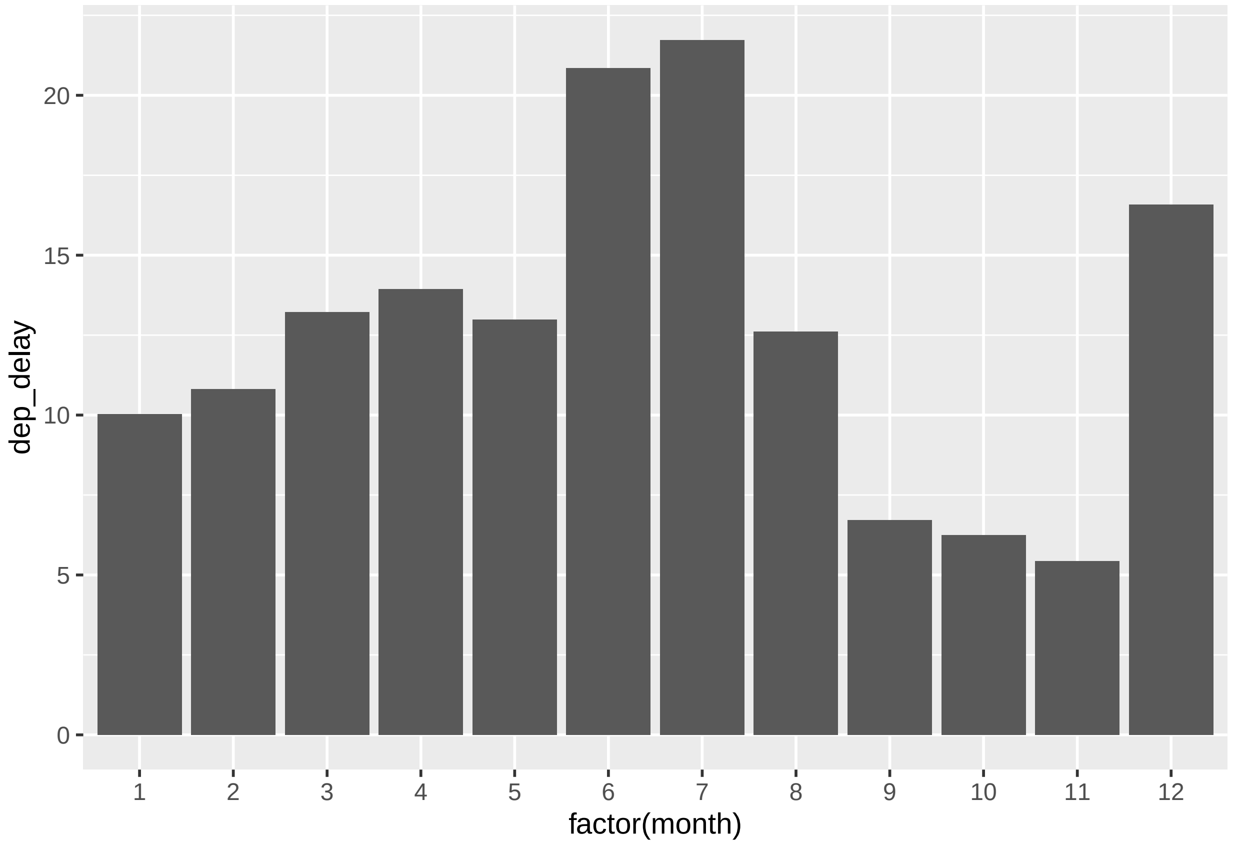

Figure 1: months and delay

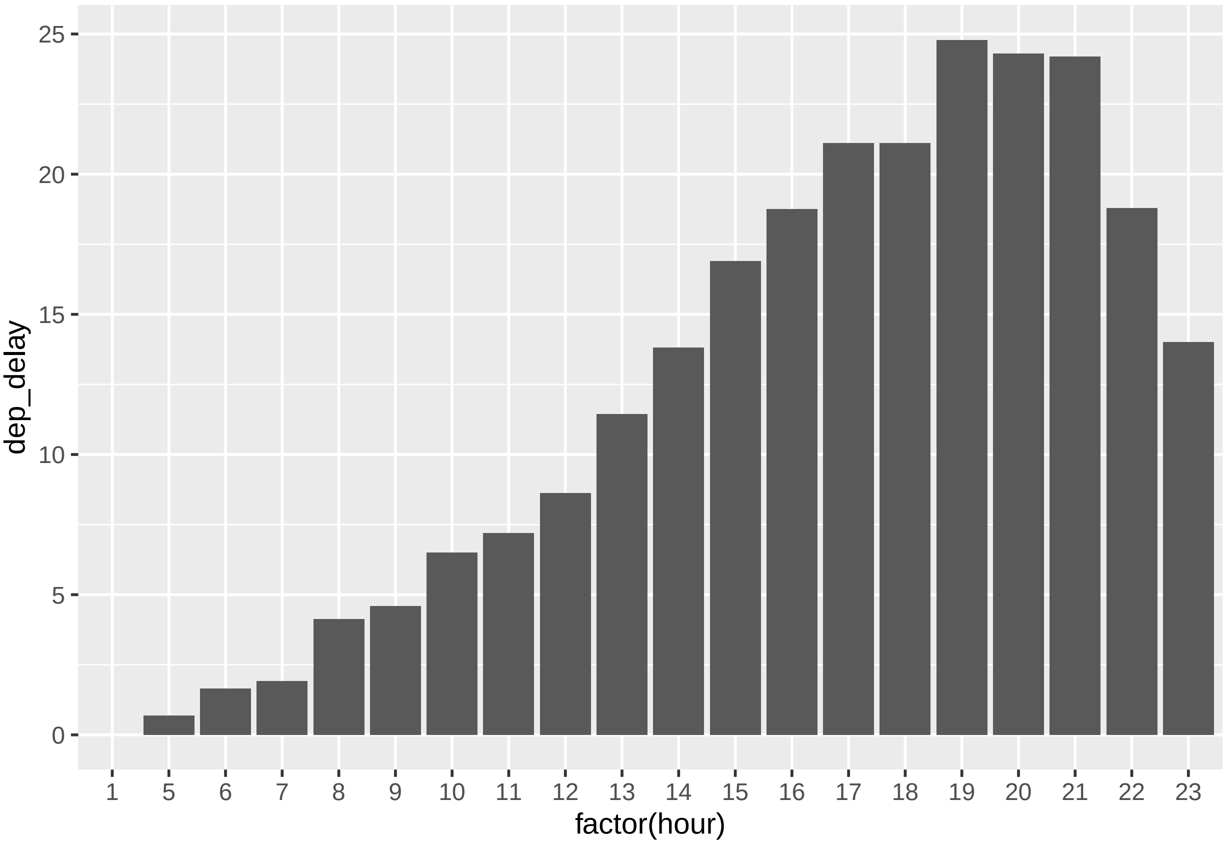

Figure 2: Hour and delay

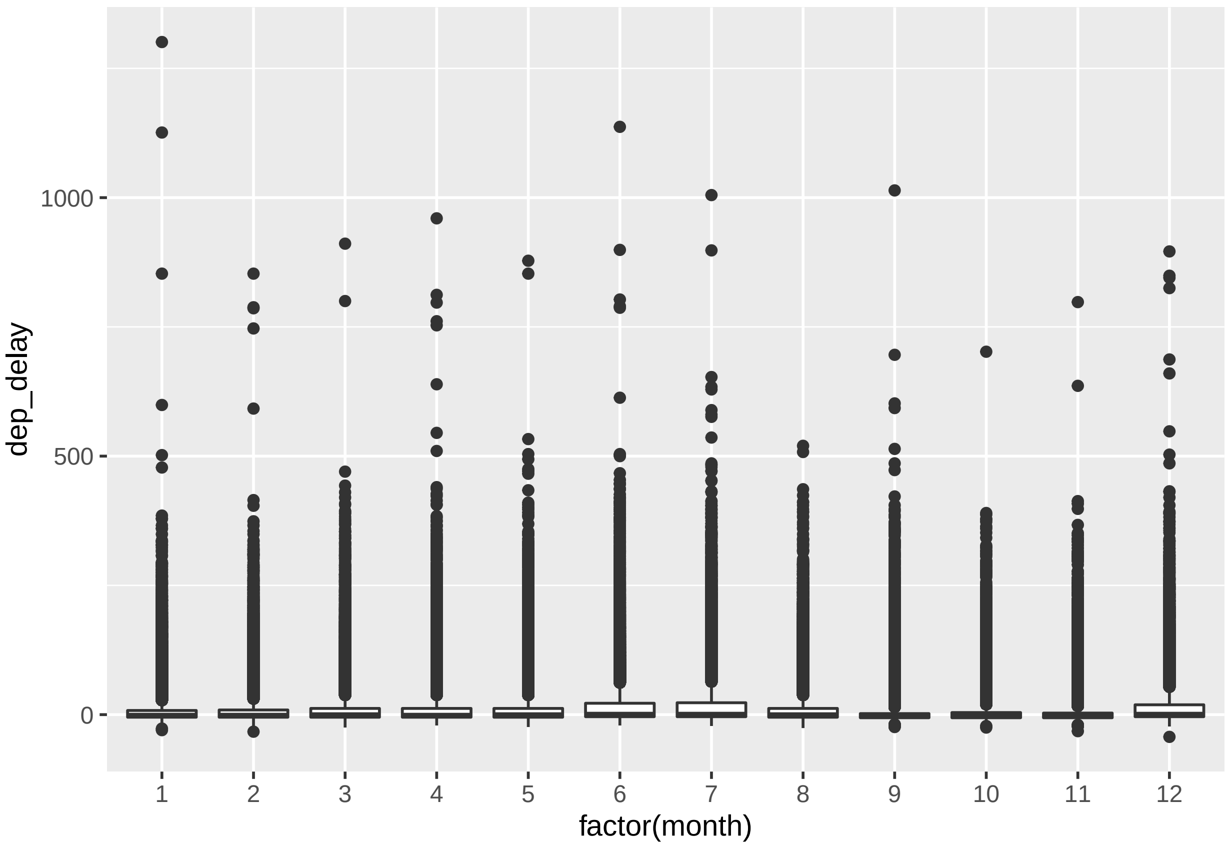

Figure 3: month and delay

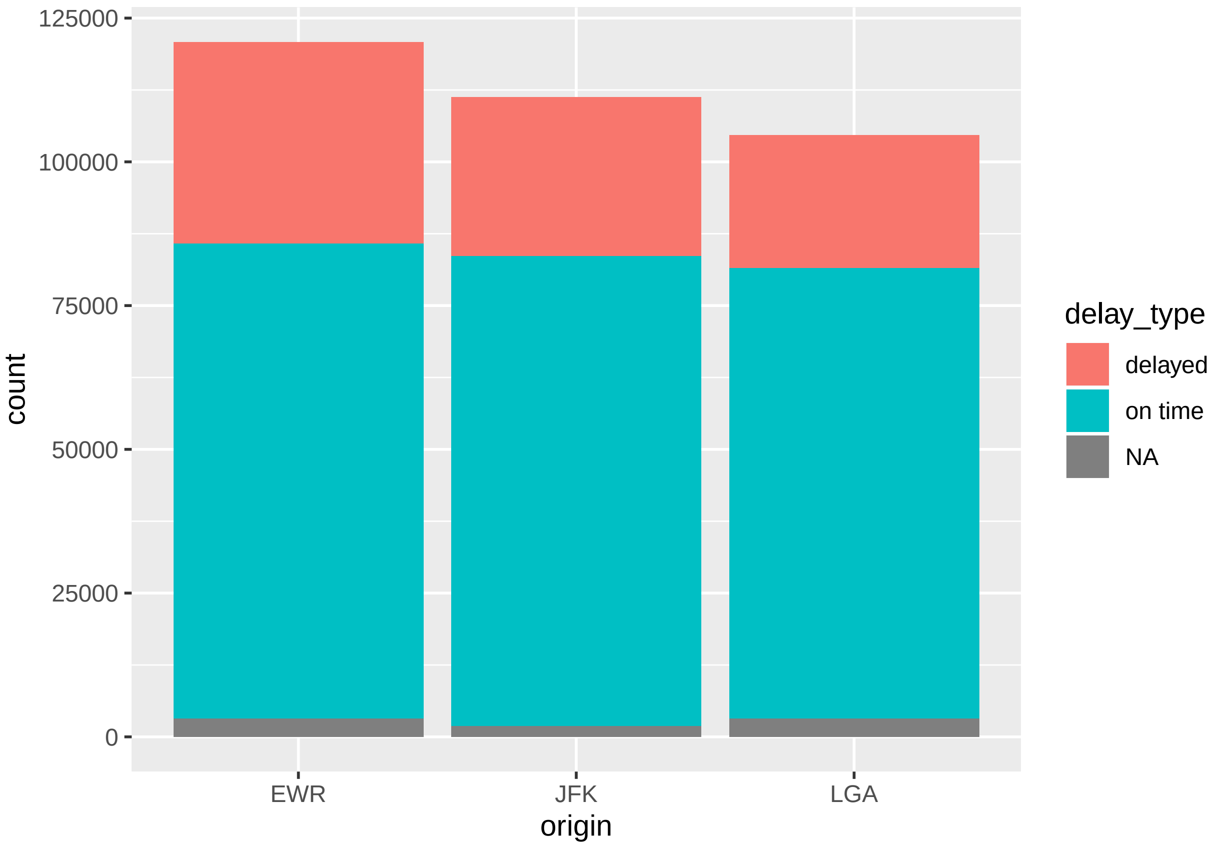

Figure 4: origin and delay

(d) Challenge Your Results

After completing the exploratory analyses from Problem 1c, do you have any concerns about your findings? How well defined was your original question? Do you still believe this question can be answered using this dataset? Comment on any ethical and/or privacy concerns you have with your analysis.

1. In the month vs delay relation, we have reached to a conclusion that the delays are more in June,July as compared to other months. Here we have cited the reason that there might be an increase in carriers causing air traffic. But to verify this assumption we need to find the number of carriers flying each month from the three airports.

2. In the hour vs delay relation, we have assumed that their are more carriers in the day as compared to the early mornings. Again, we need to check this fact in order to prove our relation.

3.In the origin vs delay plot, we have plot the delay and on time data for three different airports individually, but we have not kept into consideration the number of flights flying from each of these airports are different, so a direct comparison can misrepresent the actual data.

One of the concerns related to my analysis is regarding the cleaning of the dataset.