IMT 573: Problem Set 1 - Exploring Data

Naga Soundari Balamurugan

Due: Tuesday, October 9, 2018 at or before 11:59AM

Collaborators: Jayashree Raman

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problemset1.Rmd` file from Canvas. 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. If you are using more than just a standard function that you found from another source, please credit the source in the comments. For example:

```r
# code adapted from "Example: Multiplication Table"  https://www.datamentor.io/r-programming/examples/m

# assign num
num = 8
# use for loop to iterate 10 times
for(i in 1:10) {
print(paste(num,'x', i, '=', num*i))
}
```

```
## [1] "8 x 1 = 8"
## [1] "8 x 2 = 16"
## [1] "8 x 3 = 24"
## [1] "8 x 4 = 32"
## [1] "8 x 5 = 40"
## [1] "8 x 6 = 48"
## [1] "8 x 7 = 56"
## [1] "8 x 8 = 64"
## [1] "8 x 9 = 72"
## [1] "8 x 10 = 80"
```

4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. 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 R Markdown file to `YourLastName_YourFirstName_ps1.Rmd`, knit a PDF and submit both the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages. If you haven't yet installed them you will need to begin by using `install.packages()`

```r
# Load standard libraries
library(tidyverse)
library(nycflights13)
library(dplyr)
library(kableExtra)
```

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.

(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.

```r
#List all the functions in the nycflights13 package
ls("package:nycflights13")
```

```
## [1] "airlines" "airports" "flights"  "planes"   "weather"
```

```r
#Exploring the airlines dataset
nycflights_airlines <- nycflights13::airlines
str(nycflights_airlines)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    16 obs. of  2 variables:
##  $ carrier: chr  "9E" "AA" "AS" "B6" ...
##  $ name   : chr  "Endeavor Air Inc." "American Airlines Inc." "Alaska Airlines Inc." "JetBlue Airways
```

```r
nrow(nycflights_airlines)
```

```
## [1] 16
```

```r
head(nycflights_airlines)
```

```
## # 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.
```

```r
tail(nycflights_airlines)
```

```
## # A tibble: 6 x 2
##   carrier name
##   <chr>   <chr>
## 1 OO      SkyWest Airlines Inc.
## 2 UA      United Air Lines Inc.
## 3 US      US Airways Inc.
## 4 VX      Virgin America
## 5 WN      Southwest Airlines Co.
## 6 YV      Mesa Airlines Inc.
```

```r
#Exploring the airports dataset
nycflights_airports <- nycflights13::airports
str(nycflights_airports)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1458 obs. of  8 variables:
##  $ faa  : chr  "04G" "06A" "06C" "06N" ...
##  $ name : chr  "Lansdowne Airport" "Moton Field Municipal Airport" "Schaumburg Regional" "Randall Ai
##  $ lat  : num  41.1 32.5 42 41.4 31.1 ...
##  $ lon  : num  -80.6 -85.7 -88.1 -74.4 -81.4 ...
##  $ alt  : int  1044 264 801 523 11 1593 730 492 1000 108 ...
##  $ tz   : num  -5 -6 -6 -5 -5 -5 -5 -5 -5 -8 ...
##  $ dst  : chr  "A" "A" "A" "A" ...
##  $ tzone: chr  "America/New_York" "America/Chicago" "America/Chicago" "America/New_York" ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 12
##   .. ..$ id     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ name   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ city   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ country: list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ faa    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ icao   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ lat    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ lon    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ alt    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ tz     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ dst    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ tzone  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

```r
nrow(nycflights_airports)
```

```
## [1] 1458
```

```r
head(nycflights_airports)
```

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

```r
tail(nycflights_airports)
```

```
## # A tibble: 6 x 8
##   faa   name                          lat    lon   alt    tz dst   tzone
##   <chr> <chr>                       <dbl>  <dbl> <int> <dbl> <chr> <chr>
## 1 ZTY   Boston Back Bay Station    42.3  -71.1    20   -5. A     America/~
## 2 ZUN   Black Rock                 35.1 -109.   6454   -7. A     America/~
## 3 ZVE   New Haven Rail Station     41.3  -72.9     7   -5. A     America/~
## 4 ZWI   Wilmington Amtrak Station  39.7  -75.6     0   -5. A     America/~
## 5 ZWU   Washington Union Station   38.9  -77.0    76   -5. A     America/~
## 6 ZYP   Penn Station               40.8  -74.0    35   -5. A     America/~
```

```r
#Exploring the flights dataset
nycflights_flights <- nycflights13::flights
str(nycflights_flights)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    336776 obs. of  19 variables:
##  $ year          : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int  517 533 542 544 554 554 555 557 557 558 ...
##  $ sched_dep_time: int  515 529 540 545 600 558 600 600 600 600 ...
##  $ dep_delay     : num  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int  830 850 923 1004 812 740 913 709 838 753 ...
##  $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
##  $ arr_delay     : num  11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr  "UA" "UA" "AA" "B6" ...
##  $ flight        : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
##  $ tailnum       : chr  "N14228" "N24211" "N619AA" "N804JB" ...
##  $ origin        : chr  "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr  "IAH" "IAH" "MIA" "BQN" ...
##  $ air_time      : num  227 227 160 183 116 150 158 53 140 138 ...
##  $ distance      : num  1400 1416 1089 1576 762 ...
##  $ hour          : num  5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num  15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct, format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

```r
nrow(nycflights_flights)
```

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

```r
head(nycflights_flights)
```

```
## # 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(nycflights_flights)
```

```
## # 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>
```

```
#Exploring the planes dataset
nycflights_planes <- nycflights13::planes
str(nycflights_planes)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    3322 obs. of  9 variables:
##  $ tailnum     : chr  "N10156" "N102UW" "N103US" "N104UW" ...
##  $ year        : int  2004 1998 1999 1999 2002 1999 1999 1999 1999 1999 ...
##  $ type        : chr  "Fixed wing multi engine" "Fixed wing multi engine" "Fixed wing multi engine"
##  $ manufacturer: chr  "EMBRAER" "AIRBUS INDUSTRIE" "AIRBUS INDUSTRIE" "AIRBUS INDUSTRIE" ...
##  $ model       : chr  "EMB-145XR" "A320-214" "A320-214" "A320-214" ...
##  $ engines     : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ seats       : int  55 182 182 182 55 182 182 182 182 182 ...
##  $ speed       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ engine      : chr  "Turbo-fan" "Turbo-fan" "Turbo-fan" "Turbo-fan" ...
```

```
nrow(nycflights_planes)
```

```
## [1] 3322
```

```
head(nycflights_planes)
```

```
## # 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 win~ EMBRAER       EMB-1~        2    55    NA Turbo~
## 2 N102UW   1998 Fixed win~ AIRBUS INDUS~ A320-~        2   182    NA Turbo~
## 3 N103US   1999 Fixed win~ AIRBUS INDUS~ A320-~        2   182    NA Turbo~
## 4 N104UW   1999 Fixed win~ AIRBUS INDUS~ A320-~        2   182    NA Turbo~
## 5 N10575   2002 Fixed win~ EMBRAER       EMB-1~        2    55    NA Turbo~
## 6 N105UW   1999 Fixed win~ AIRBUS INDUS~ A320-~        2   182    NA Turbo~
```

```
tail(nycflights_planes)
```

```
## # A tibble: 6 x 9
##   tailnum  year type      manufacturer    model engines seats speed engine
##   <chr>   <int> <chr>     <chr>           <chr>   <int> <int> <int> <chr>
## 1 N996DL   1991 Fixed wi~ MCDONNELL DOUG~ MD-88       2   142    NA Turbo~
## 2 N997AT   2002 Fixed wi~ BOEING          717-~       2   100    NA Turbo~
## 3 N997DL   1992 Fixed wi~ MCDONNELL DOUG~ MD-88       2   142    NA Turbo~
## 4 N998AT   2002 Fixed wi~ BOEING          717-~       2   100    NA Turbo~
## 5 N998DL   1992 Fixed wi~ MCDONNELL DOUG~ MD-88       2   142    NA Turbo~
## 6 N999DN   1992 Fixed wi~ MCDONNELL DOUG~ MD-88       2   142    NA Turbo~
```

```
#Exploring the weather dataset
nycflights_weather <- nycflights13::weather
str(nycflights_planes)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    3322 obs. of  9 variables:
##  $ tailnum     : chr  "N10156" "N102UW" "N103US" "N104UW" ...
##  $ year        : int  2004 1998 1999 1999 2002 1999 1999 1999 1999 1999 ...
##  $ type        : chr  "Fixed wing multi engine" "Fixed wing multi engine" "Fixed wing multi engine"
##  $ manufacturer: chr  "EMBRAER" "AIRBUS INDUSTRIE" "AIRBUS INDUSTRIE" "AIRBUS INDUSTRIE" ...
##  $ model       : chr  "EMB-145XR" "A320-214" "A320-214" "A320-214" ...
##  $ engines     : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ seats       : int  55 182 182 182 55 182 182 182 182 182 ...
##  $ speed       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ engine      : chr  "Turbo-fan" "Turbo-fan" "Turbo-fan" "Turbo-fan" ...
```

```
nrow(nycflights_planes)
```

```
## [1] 3322
```

```
head(nycflights_planes)
```

```
## # 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 win~ EMBRAER       EMB-1~       2    55    NA Turbo~
## 2 N102UW   1998 Fixed win~ AIRBUS INDUS~ A320-~       2   182    NA Turbo~
## 3 N103US   1999 Fixed win~ AIRBUS INDUS~ A320-~       2   182    NA Turbo~
## 4 N104UW   1999 Fixed win~ AIRBUS INDUS~ A320-~       2   182    NA Turbo~
## 5 N10575   2002 Fixed win~ EMBRAER       EMB-1~       2    55    NA Turbo~
## 6 N105UW   1999 Fixed win~ AIRBUS INDUS~ A320-~       2   182    NA Turbo~
```

```
tail(nycflights_planes)
```

```
## # A tibble: 6 x 9
##   tailnum  year type      manufacturer    model engines seats speed engine
##   <chr>   <int> <chr>     <chr>           <chr>   <int> <int> <int> <chr>
## 1 N996DL   1991 Fixed wi~ MCDONNELL DOUG~ MD-88       2   142    NA Turbo~
## 2 N997AT   2002 Fixed wi~ BOEING          717-~       2   100    NA Turbo~
## 3 N997DL   1992 Fixed wi~ MCDONNELL DOUG~ MD-88       2   142    NA Turbo~
## 4 N998AT   2002 Fixed wi~ BOEING          717-~       2   100    NA Turbo~
## 5 N998DL   1992 Fixed wi~ MCDONNELL DOUG~ MD-88       2   142    NA Turbo~
## 6 N999DN   1992 Fixed wi~ MCDONNELL DOUG~ MD-88       2   142    NA Turbo~
```

> The nycflights13 package has 5 different datasets which includes the details of airlines, airports, flights, planes and weather. It includes very detailed data of each segment which are as follows.

The **airlines** dataframe has 16 rows of data with 2 columns which are the airplane code and its name.

The **airports** dataframe has 8 columns and of 1458 rows. This dataframe has details specific to an airport location like latitude, longitude, altitude, airport's name, zone etc.,

The **flights** dataframe has 19 columns and of 336, 776 rows. It has all the details of the flights from the year 2013. The details include date, departure time, scheduled departure time, delay, arrival time, scheduled arrival time, flying time, distance, origin, destination etc., The **planes** dataframe has 9 columns and of 3322 rows. As the name indicates, this dataframe has all the details related to the planes like its number, type, manufactured yaer, model, engine, no of seats, speed etc.,

The **weather** dataframe has 15 columns and of 26115 rows. This dataframe has a hour specific weather information for the year 2013. The details include temperature, humidity, wind direction, wind speed, precipitation, pressure, visibility etc.,

(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.

```r
#Explore all the columns to find interesting connections
str(nycflights_flights)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    336776 obs. of  19 variables:
##  $ year          : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int  517 533 542 544 554 554 555 557 557 558 ...
##  $ sched_dep_time: int  515 529 540 545 600 558 600 600 600 600 ...
##  $ dep_delay     : num  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int  830 850 923 1004 812 740 913 709 838 753 ...
##  $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
##  $ arr_delay     : num  11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr  "UA" "UA" "AA" "B6" ...
##  $ flight        : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
##  $ tailnum       : chr  "N14228" "N24211" "N619AA" "N804JB" ...
##  $ origin        : chr  "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr  "IAH" "IAH" "MIA" "BQN" ...
##  $ air_time      : num  227 227 160 183 116 150 158 53 140 138 ...
##  $ distance      : num  1400 1416 1089 1576 762 ...
##  $ hour          : num  5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num  15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct, format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

```r
#To find the unique flights in nycflights dataset
uniqueFlights <- unique(nycflights_flights$flight)
NoOfUniqueFlights <- length(uniqueFlights)

uniqueCarriers <- unique(nycflights_flights$carrier)
NoOfUniqueCarriers <- length(uniqueCarriers)
```

At a very first glance, the factor that caught me are the delay in departure and arrival timings of the flight. There dataset contains flight details of 3844 planes of 16 carriers and hence I would like to pose a question, **"which are the top 5 airlines that got delayed the most?"**. By finding the answer to this question, the reason for delay could also then be explored by drilling down to flight details(i.e., flight number) and comparing it against weather(if it is because of bad weather condition) and planes(if the model is obselete, engine condition etc.,) dataset. This could help to improve the airline services by eliminating the delays and the passengers could be satisfied.

> This can be found from the variable dep_delay(delay in departure) and carrier(Two letter carrier abbreviation). The data can be sorted on the basis of dep_delay variable and the top 10 could be filtered out.

Another question that strike my mind of is **To which cities are there most and least flights from Newyork?**. This could help us to find the frequency of flights to different locations and if there is any specific reason behind them. As a next step, we could also explore if the frequency need to be increased or decreased to certain locations.

In order to answer this, we could go with exploring the variable dest(destination). The frequency of each destination needs to be measured to find the most and least accessible location from Newyork through air.

(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 related to each question. Be sure to describe what the visuals show and how they speak to your question of interest.

```
#Filter the flights that had delays
delayedFlights <- subset(nycflights_flights, nycflights_flights$dep_delay > 0)

#Group the flights by carrier and sum the delay time
delayByCarrier <- delayedFlights %>%  group_by(carrier) %>%
  dplyr::summarize(count = n(), TotalDelay = sum(dep_delay)) %>%
  select(carrier, count, TotalDelay)

#Display the table that contains the list of carriers with total delay time and the no of times delayed
kable(delayByCarrier) %>% kable_styling(bootstrap_options = c("striped", "hover"))
```

```
## Warning in kable_styling(., bootstrap_options = c("striped", "hover")):
## Please specify format in kable. kableExtra can customize either HTML or
## LaTeX outputs. See https://haozhu233.github.io/kableExtra/ for details.
```

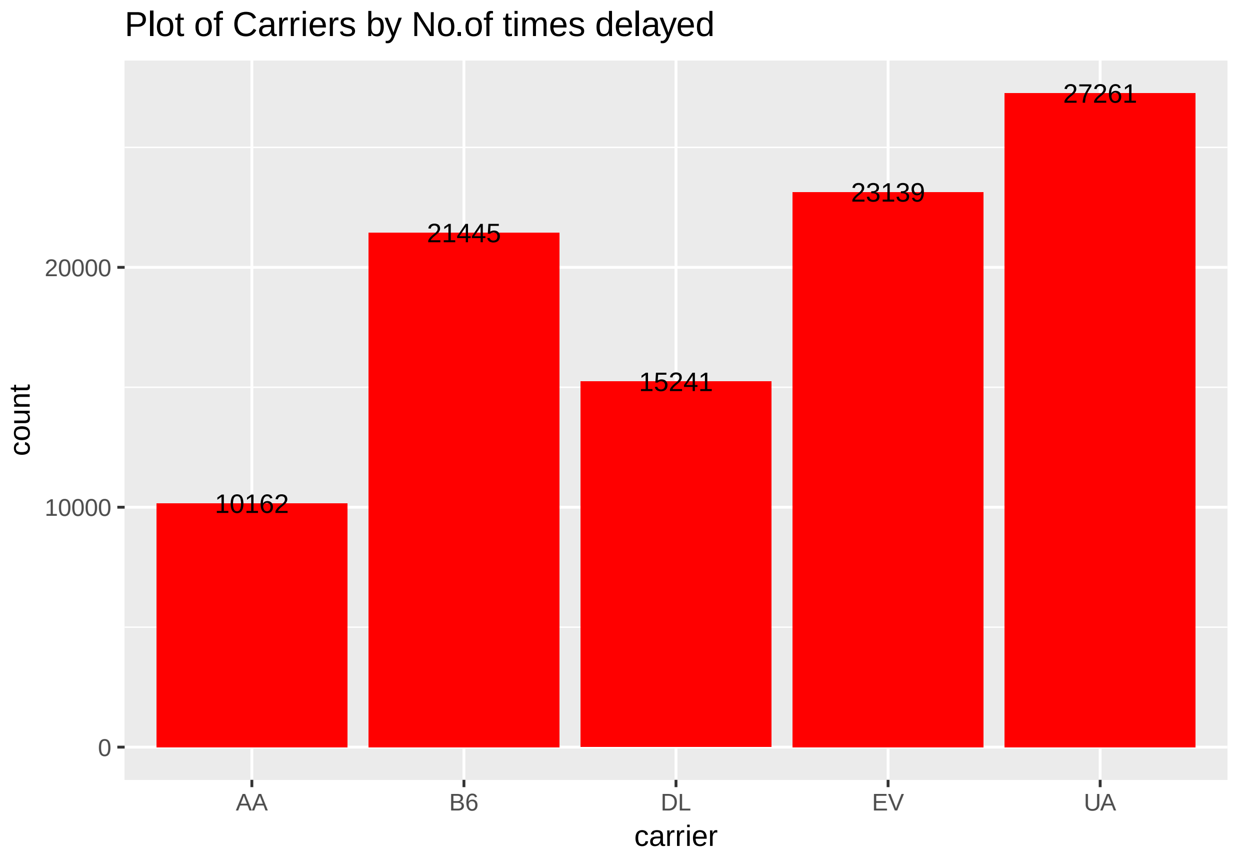
carrier	count	TotalDelay
9E	7063	345522
AA	10162	377714
AS	226	7083
B6	21445	853387
DL	15241	570017
EV	23139	1164581
F9	341	15392
FL	1654	67526
HA	69	3094
MQ	8031	360715
OO	9	522
UA	27261	815818
US	4775	157817
VX	2225	76662
WN	6558	228595
YV	233	12338

To explore in detail about each variable ?nycflights13::flights is used. The variable dep_delay denotes delay in minutes and negative number indicates early departure.

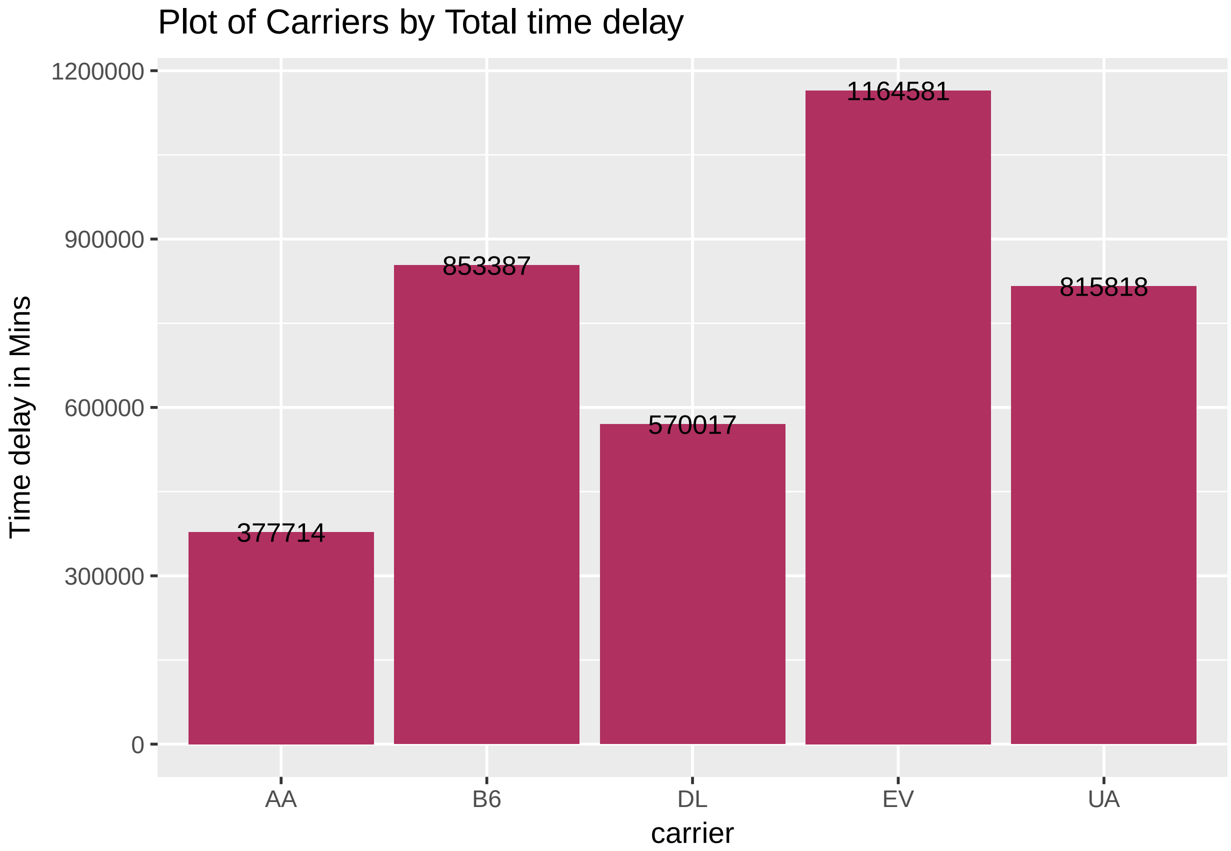
From the table delayByCarrier, we can see the number of times each airlines delayed and the total time delayed. This table is used to get the below visualization.

```
#Sort the data by no of times delayed
delayByCarrier_count <- delayByCarrier %>% arrange(desc(count))
```

```
#Sort the data by total time delay
delayByCarrier_time <- delayByCarrier %>% arrange(desc(TotalDelay))

delaybyCount <- ggplot(data = head(delayByCarrier_count, 5), aes(x = carrier, y = count)) +
  geom_bar(stat="identity", fill = "red") +
  geom_text(aes(label=count), color="black", size=3.5) +
  ggtitle("Plot of Carriers by No.of times delayed")

delaybyCount
```



```
delaybyTime <- ggplot(data = head(delayByCarrier_time, 5), aes(x = carrier, y = TotalDelay)) +
  geom_bar(stat="identity", fill = "maroon") +
  geom_text(aes(label=TotalDelay), color="black", size=3.5) +
  ggtitle("Plot of Carriers by Total time delay") + ylab("Time delay in Mins")

delaybyTime
```



From the above plots we can see that the airlines UA(United Airlines Inc.), EV(Express Jet Airlines Inc.), B6(JetBlue Airways), DL(Delta Air Lines Inc.) and AA(American Airlines Inc.) got delayed the most by both factors(no of times delayed and total time delayed). The Delta Airlines and American Airlines remains in fourth and fifth place respectively for both the factors. But the order of top 3 airlines is affected for both the factors. The top 3 most delayed airlines in order, By total time delayed - Express Jet Airlines, JetBlue Airways and United Airlines. By No of times delayed - United Airlines, Express Jet Airlines and JetBlue Airlines.

Though all these airlines does not fall under the budget airlines category except JetBlue and Express Jet, these has the most frequent delays. Hence there should be some other reason that needs to be explored.

> To explore the second question, "To which cities are there most and least flights from Newyork?", we would use the variable dest which is a three letter representation of the cities and it can then be mapped through nycflights13::airports data.

```
#To find the number of cities that has flights from Newyork
destinationcities <- unique(nycflights_flights$dest)
NoOfDestinationCities <- length(destinationcities)
NoOfDestinationCities
```

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

> There are flights to 105 differet cities from Newyork.

```
#Group the flights by destination cities and calculate the frequency
citiesFrequency <- nycflights_flights %>%  group_by(dest) %>%
  dplyr::summarize(count = n()) %>%
  select(dest, count)
```

```r
#Sort the data by frequency of flights
citiesFrequencySorted <- citiesFrequency %>% arrange(desc(count))

#Top 10 cities that has most flights from Newyork
MostFreqCities <- head(citiesFrequencySorted, 10)
kable(MostFreqCities) %>% kable_styling(bootstrap_options = c("striped", "hover"))
```

```
## Warning in kable_styling(., bootstrap_options = c("striped", "hover")):
## Please specify format in kable. kableExtra can customize either HTML or
## LaTeX outputs. See https://haozhu233.github.io/kableExtra/ for details.
```

dest	count
ORD	17283
ATL	17215
LAX	16174
BOS	15508
MCO	14082
CLT	14064
SFO	13331
FLL	12055
MIA	11728
DCA	9705

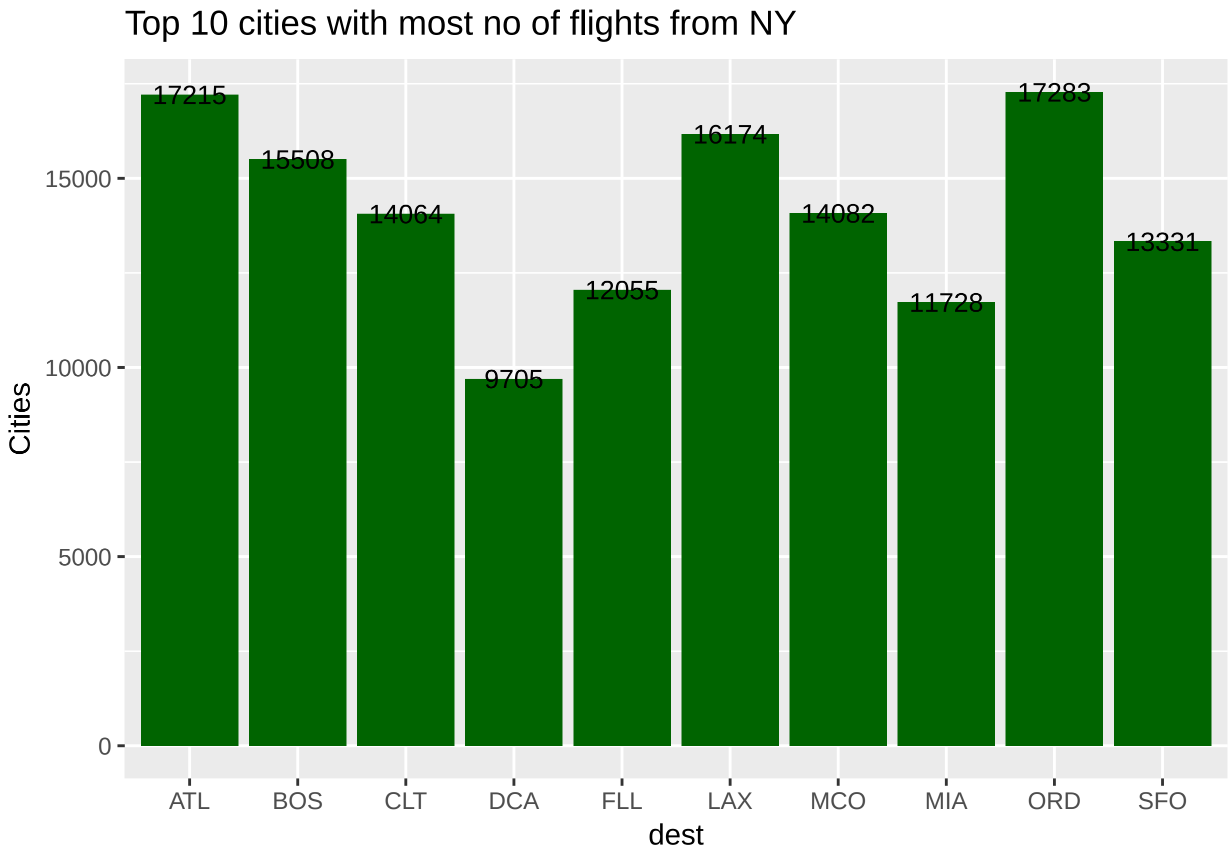
```r
#Top 10 cities that has least number of flights from Newyork
LeastFreqCities <- tail(citiesFrequencySorted, 10)
kable(LeastFreqCities) %>% kable_styling(bootstrap_options = c("striped", "hover"))
```

```
## Warning in kable_styling(., bootstrap_options = c("striped", "hover")):
## Please specify format in kable. kableExtra can customize either HTML or
## LaTeX outputs. See https://haozhu233.github.io/kableExtra/ for details.
```

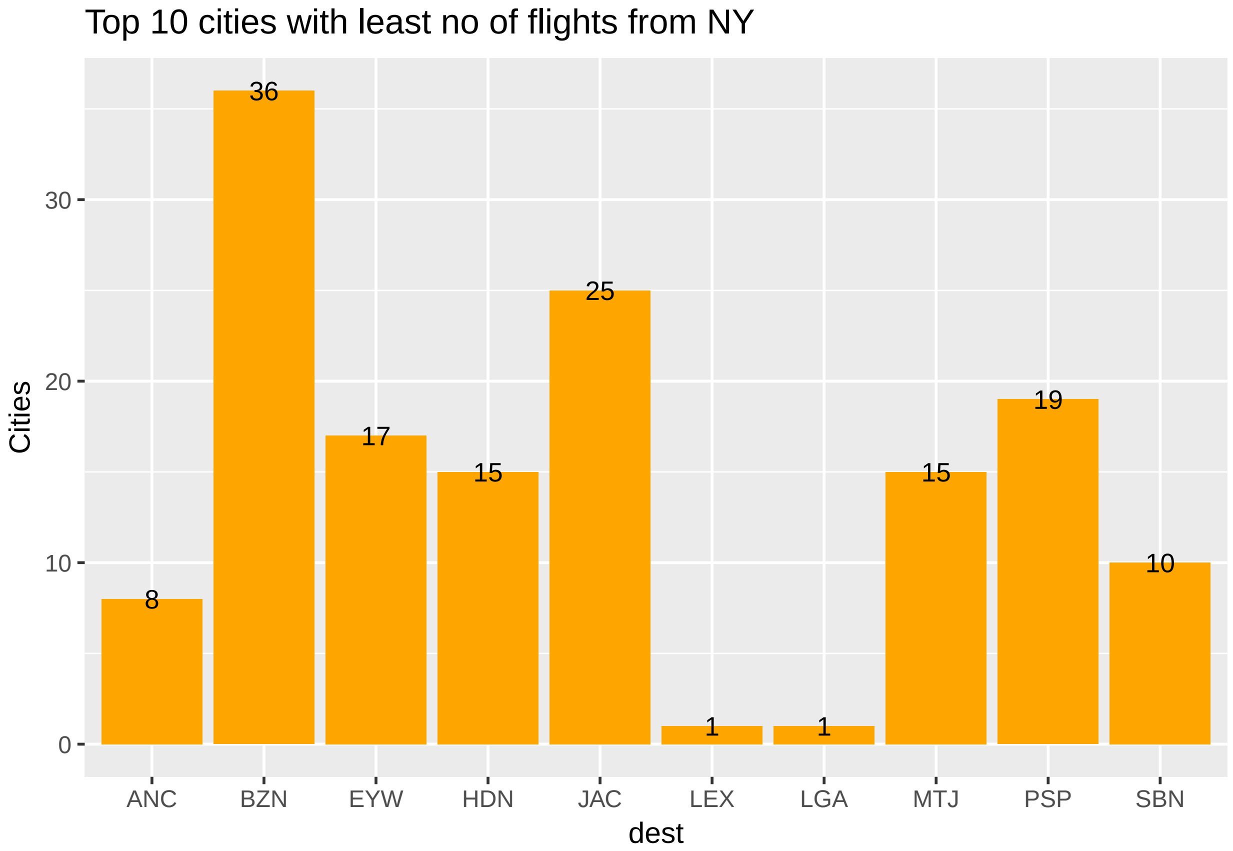
dest	count
BZN	36
JAC	25
PSP	19
EYW	17
HDN	15
MTJ	15
SBN	10
ANC	8
LEX	1
LGA	1

The plots for these tables are shown below.

```r
#cities that has most flights from Newyork
MostFreqCitiesViz <- ggplot(data = MostFreqCities, aes(x = dest, y = count)) +
  geom_bar(stat="identity", fill = "darkgreen") +
  geom_text(aes(label=count), color="black", size=3.5) +
  ggtitle("Top 10 cities with most no of flights from NY") + ylab("Cities")

MostFreqCitiesViz
```



```r
#cities that has most flights from Newyork
LeastFreqCitiesViz <- ggplot(data = LeastFreqCities, aes(x = dest, y = count)) +
  geom_bar(stat="identity", fill = "orange") +
  geom_text(aes(label=count), color="black", size=3.5) +
  ggtitle("Top 10 cities with least no of flights from NY") + ylab("Cities")

LeastFreqCitiesViz
```



From the visualization that shows the cities with most no of flights, we can see that the cities *Atlanta, Boston, Charlotte(NC), Ronald Reagan(VA), Fort Lauderdale(FL), Los Angeles, Orlando(FL), Miami, Ohare(chicago), San Francisco* has most number of flights from Newyork. As I guessed, most of these cities are in the east side of the United States except Los angeles and San Franciso. Since these both cities are the central hub of international arrivals and departures, there should be frequent flights from NewYork.

The vizualisation with the least no of flights shows that, the cities *Ketucky, LaGuardia(NY), Anchorage(AK), Indiana, Colorado, Hayden(Colorado), KeyWest(FL), Palm springs(CA), Wyoming, Montana* has the least number of flights. By just glancing through the list we can know that these airports are of smaller size and does not have much activity. Also these cities are not hub for any major industries or businesses. Thus this explains the reason behind the less frequency of flights.

(d) Challenge Your Results:

After completing the exploratory analysis from Problem 1c, do you have any concerns about your findings? Comment on any ethical and/or privacy concerns you have with your analysis.

> This dataset was really a good one and has so many interesting variables that could be explored. I would like to explore deeper on the questions I have analysed. In the flights dataset, the variables dep_delay and arr_delay had negative values which means that those flights were early. Hence it does not make sense to have them in the dataset.
>
> I was considered about if these airlines have any privacy concerns on the details about the flight model, engine details etc., Listing the most delayed airlines could affect the brand value of the airlines as well. Also, as these data are opensourced and the visualization about the cities gives a clear cut picture of flights with most and least passengers, anyone could access it for a negative cause.