IMT 573: Problem Set 2 - Data manipulations

Shree Priya

Due: Wednesday, October 16 2019

```
# Load standard libraries
library("tidyverse")
library("nycflights13")
data(flights)
library(plotly)
library(stringr)
```

1.1 Explore the data

1: How many flights out of NYC are in the data?

```
#To explore the number of rows and columns in our data set.
dim(flights)
```

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

There are totally 336776 flights originating from NYC.

2:How many NYC airports are included in this data? Which airports are these?

```
#Grouping by origin to check the number of flights per origin
nyc1 = flights %>% group_by(origin) %>% summarise(count=n())
nyc1
```

```
## # A tibble: 3 x 2
##   origin  count
##   <chr>   <int>
## 1 EWR    120835
## 2 JFK    111279
## 3 LGA    104662
```

There are 3 airports in NYC. They are:

1. EWR having 120835 flights originating from it

2. JFK having 111279 flights originating from it.

3. LGA having 104662 flights originating from it.

3: Into how many airports did the airlines fly from NYC in 2013?

```
#Grouping by the destination and checking the count.
nyc2 = flights %>% filter(is.na(dest)==FALSE) %>% group_by(dest) %>% summarise(count=n())
dim(nyc2)
```

```
## [1] 105   2
```

There are 105 different destinations of the flights in 2013.

4: How many flights were there from NYC to Seattle(SEA)?

```
#Filtering destination at SEA
nyc3 = flights %>% filter((origin=="EWR" | origin=="JFK" | origin=="LGA") & dest=="SEA")

#Exploring the number of rows and columns
dim(nyc3)
```

```
## [1] 3923   19
```

```
head(nyc3)
```

```
## # 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      724            725        -1     1020
## 2  2013     1     1      743            730        13     1059
## 3  2013     1     1      857            851         6     1157
## 4  2013     1     1     1418           1419        -1     1726
## 5  2013     1     1     1421           1355        26     1735
## 6  2013     1     1     1730           1729         1     2039
## # ... 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>
```

There were 3923 flights that originated from NYC to SEA.

5: Were there any flights from NYC to Spokane(GAG)?

```
#Filtering out destination GAG
nyc4 = flights %>% filter((origin=="EWR" | origin=="JFK" | origin=="LGA") & dest=="GAG")
dim(nyc4)
```

```
## [1]  0 19
```

No, we can see that the number of rows in the above data is 0 so, there were no flights from any airport in NYC to Spokane.

6: What about the missing destination codes? Are there any destinations that do not look like valid airport codes?(three-letters-all-uppercase)

```
#Finding all the unique destinations
nyc5 = flights %>% group_by(dest) %>% summarise(count=n())

#Finding the number of rows
dim(nyc5)
```

```
## [1] 105   2
```

```
#Writing a regex to filter out any destination that does not have 3 uppercase
nyc5 = nyc5 %>% filter(!str_detect(dest, "[[:upper:]]{3}"))

#Checking the number of rows
nrow(nyc5)
```

```
## [1] 0
```

We can see from the above results that the number of rows for the query is 0, therefore all the destinations are valid(three-letters-uppercase)

7:Comment the questions (and answers) so far. Were you able to answer all of these questions? Are all questions well defined? Is the data good enough to answer all these?

Ans: Yes, I was able to answer all of the above questionsso far. The data given was sufficient to solve the above questions so far.

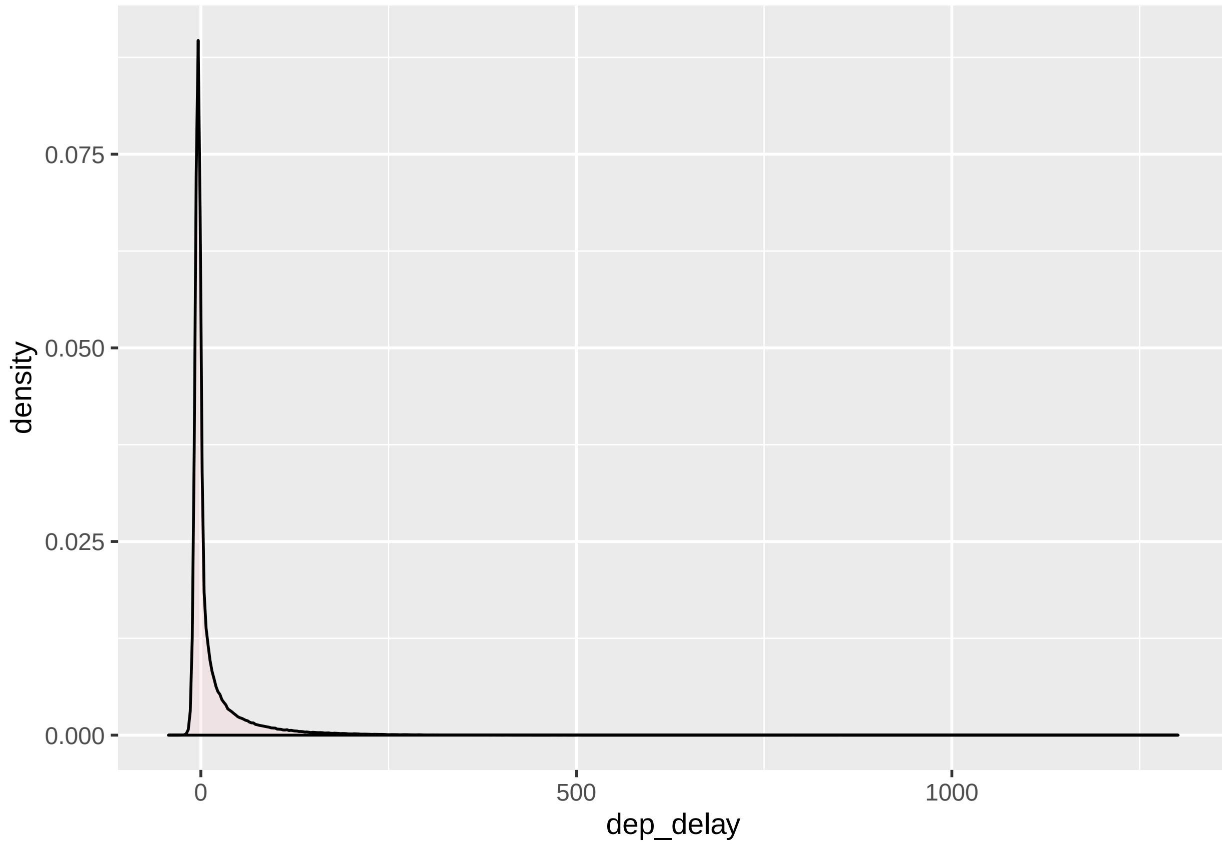
1.2 Flights are delayed...

1. What is the typical delay of the flights in the data?

```
#Filtering out null values
nyc6 = flights %>% filter(is.na(dep_delay)==FALSE)

#Finding the mean
mean(nyc6$dep_delay)
```

```
## [1] 12.63907
```

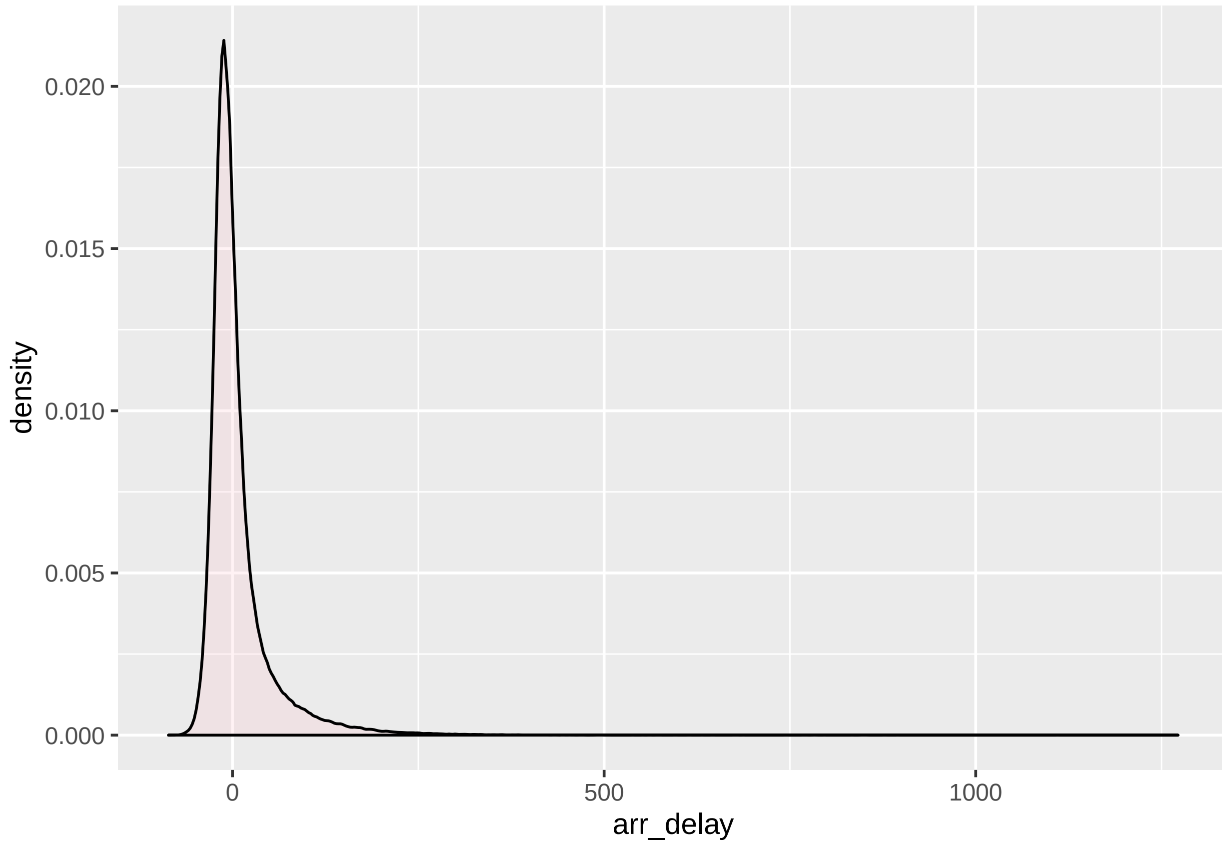
```
#Plotting the data
ggplot(nyc6, aes(x=dep_delay)) + geom_density(alpha=0.2, fill="pink")
```




```
#Filtering out null values in arrival delay
nyc7 = flights %>% filter(is.na(arr_delay)==FALSE)

#Finding the mean for arrival delay
mean(nyc7$arr_delay)
```

```
## [1] 6.895377
```

```
#Plotting the data
ggplot(nyc7, aes(x=arr_delay)) + geom_density(alpha=0.2, fill="pink")
```



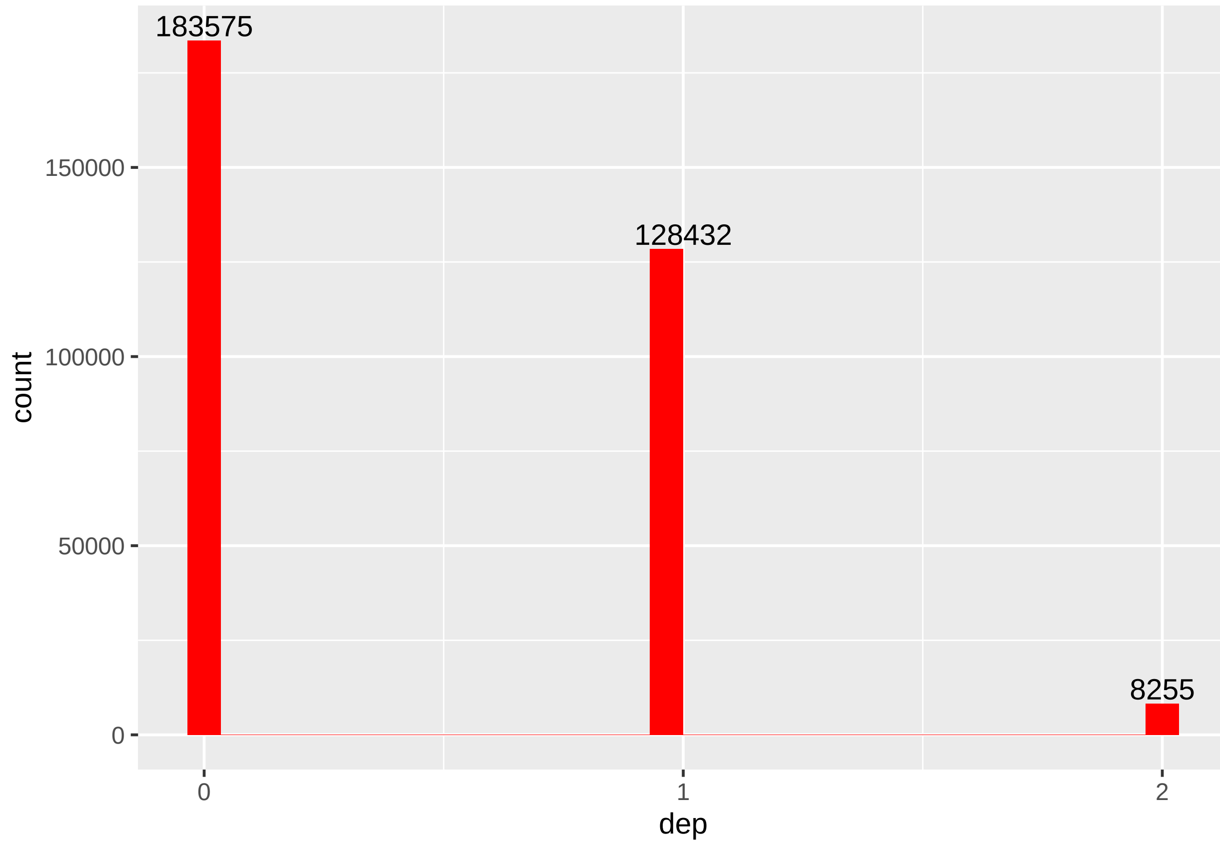
The question is not clear about what does "typical delay" mean. Typical delay with respect to what? From what I understood, find the mean arrival and departure delay: That from the above data is 6.89 and 12.63 respectively.

2: Did you remember to check how good is the delay variable? Are there missings? Are there any implausible or invalid entries? Go and check this.

```
#categorizing departure delay as 0, if its +ve, 0 is its negative, and 2 is it's NA
nyc9 = flights %>% mutate(dep = case_when(dep_delay>0 ~ 1,
                                          dep_delay<0 ~ 0,
                                          is.na(dep_delay)==TRUE ~ 2))
#Removing the NA values for dep
nyc9 = nyc9 %>% filter(is.na(dep)==FALSE)

#Checking the number of values for 0,1 and 2
nyc9 %>% group_by(dep) %>% summarise(count=n())
```

```
## # A tibble: 3 x 2
##     dep  count
##   <dbl>  <int>
## 1     0 183575
## 2     1 128432
## 3     2   8255
```

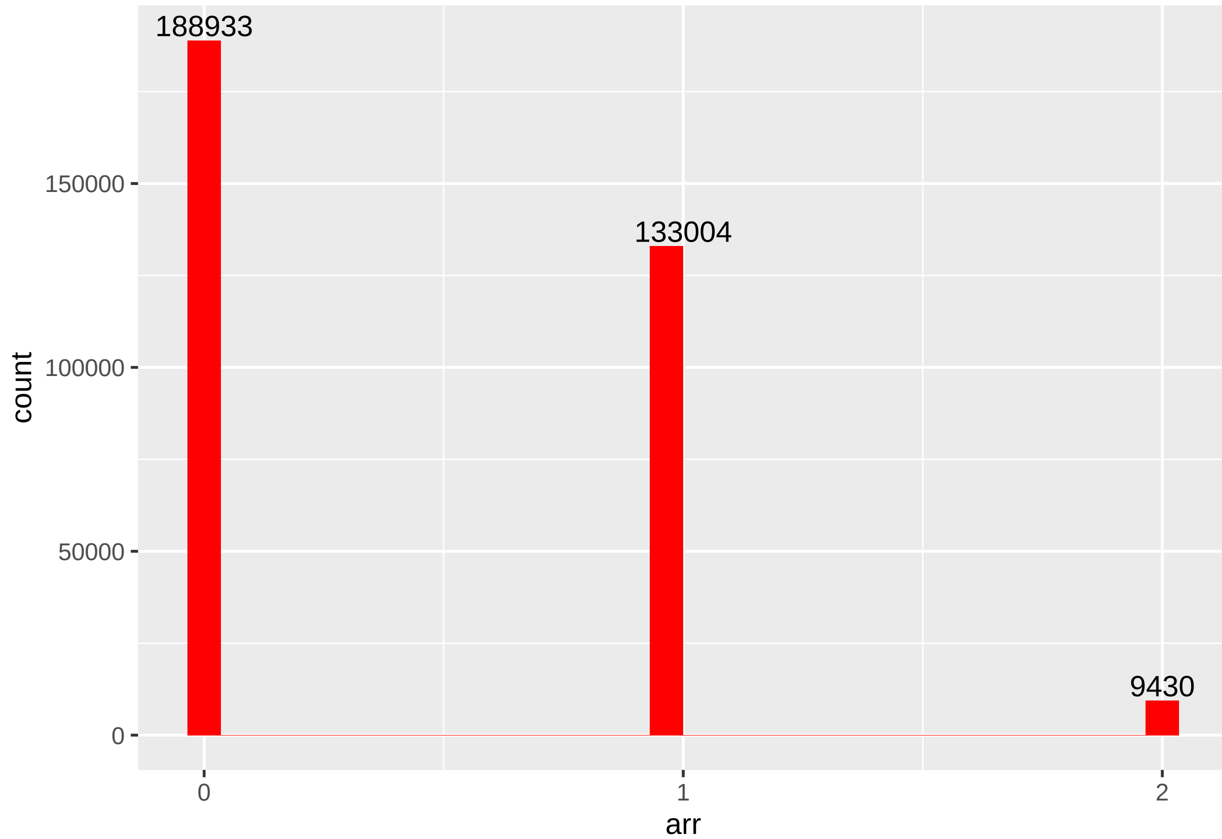
```r
#Plotting the data
ggplot(nyc9, aes(x=dep)) + geom_histogram(fill="red") + geom_text(stat = "count", aes(label=..count..),
```




```r
#categorizing arrival delay as 0, if its +ve, 0 is its negative, and 2 is it's NA
nyc10 = flights %>% mutate(arr = case_when(arr_delay>0 ~ 1,
                                          arr_delay<0 ~ 0,
                                          is.na(arr_delay)==TRUE ~ 2))

#Removing the NA values for arr
nyc10 = nyc10 %>% filter(is.na(arr)==FALSE)

#Checking the number of values for 0,1 and 2
nyc10 %>% group_by(arr) %>% summarise(count=n())
```

```
## # A tibble: 3 x 2
##     arr  count
##   <dbl>  <int>
## 1     0 188933
## 2     1 133004
## 3     2   9430
```

```r
#Plotting the data
ggplot(nyc10, aes(x=arr)) + geom_histogram(fill="red") + geom_text(stat = "count", aes(label=..count..)
```



It is not clear if the question is asking for arrival delay or departure delay, but yes, both of them have NA varibles. Departure delay has 8255 NA values, Arrival delay has 9430 NA values.

3. Now compute the delay by destinations. Which ones are the worst three destinations in terms of the longest delay?

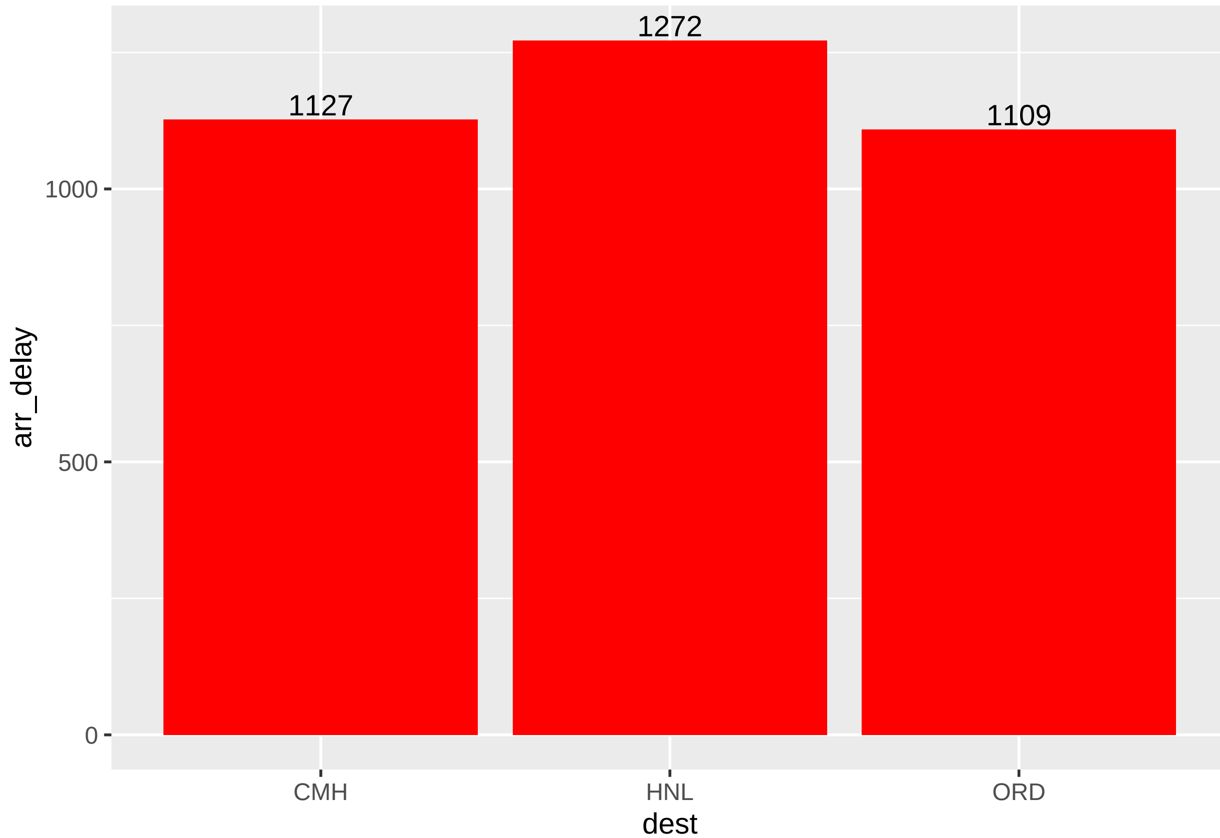
```
#Arranging the arrival delay at the destination
nyc11 = flights %>% filter(arr_delay>0) %>% arrange(desc(arr_delay))

#Finding the destinations for 3 longest delays
nyc11 = head(nyc11,3)

nyc11
```

```
## # A tibble: 3 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     1     9      641            900      1301     1242
## 2  2013     6    15     1432           1935      1137     1607
## 3  2013     1    10     1121           1635      1126     1239
## # ... 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>
```

```
#Plotting the data
ggplot(nyc11, aes(x=dest, y=arr_delay)) + geom_histogram(stat="identity", fill="red")  + geom_text(stat
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



The 3 worst destination with the highest arrival delays are CMH, HNL and ORD with HNL being the highest amongst the three having 21 hrs of delay.

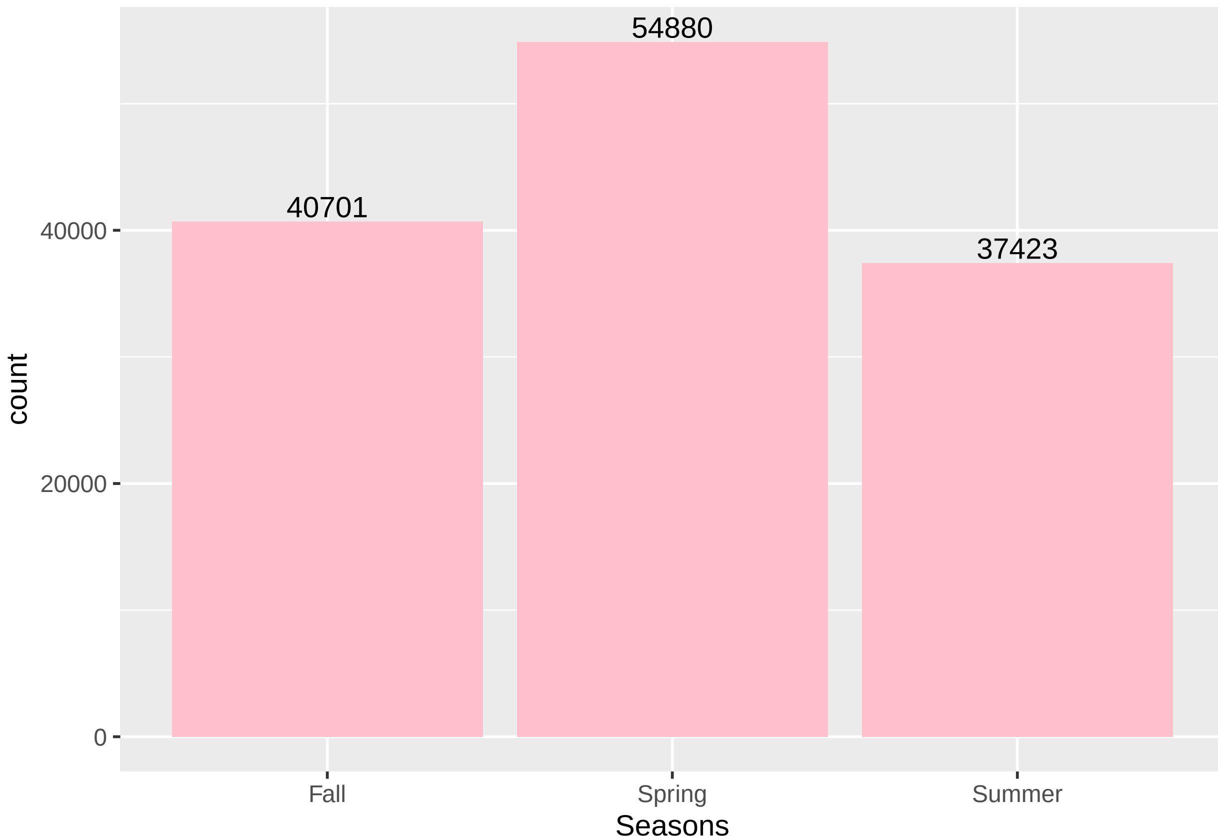
4. Delays may be partly related to weather. We do not have weather information here but let's analyze how it is related to season. Do it in two (or more) ways: one graphical, and one in a table form.

```
#Arrival Delay
#Categorizing the seasons, 1-5: Spring; 6-8: Summer; 9-12: Fall
nyc12 = flights %>% filter(arr_delay>0) %>%
  mutate(dep = case_when(month>=1 &month<=5 ~ "Spring",
                         month>=6 &month<=8 ~ "Summer",
                         month>=9 &month<=12 ~ "Fall"))
#Getting the individual counts
nyc12 = nyc12 %>% group_by(dep) %>% summarise(count=n())

nyc12
```

```
## # A tibble: 3 x 2
##   dep    count
##   <chr>  <int>
## 1 Fall   40701
## 2 Spring 54880
## 3 Summer 37423
```

```
#Plotting the data
ggplot(nyc12, aes(x=dep, y=count)) + geom_histogram(stat="identity", fill="pink") + geom_text(stat = "i
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



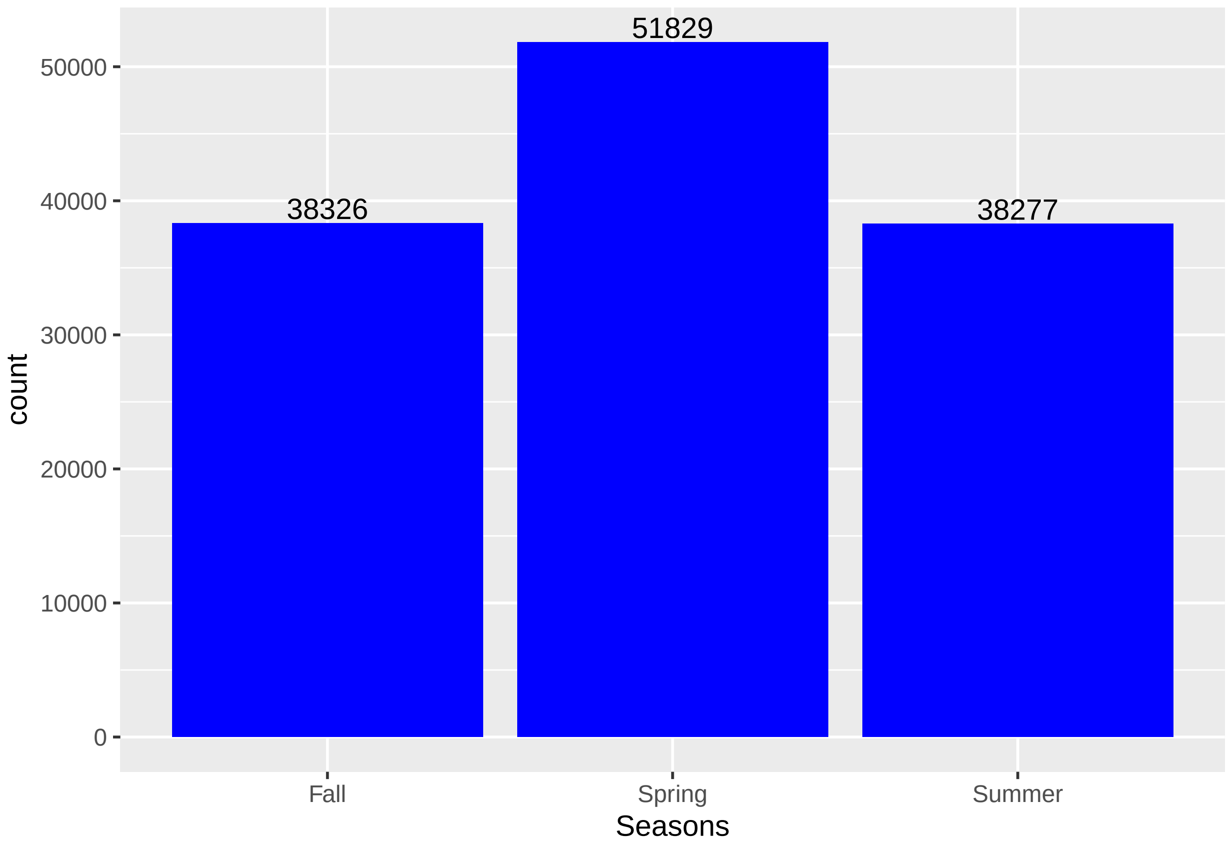

```
#Departure delay
#Categorizing the seasons, 1-5: Spring; 6-8: Summer; 9-12: Fall
nyc13 = flights %>% filter(dep_delay>0) %>%
  mutate(dep = case_when(month>=1 &month<=5 ~ "Spring",
                         month>=6 &month<=8 ~ "Summer",
                         month>=9 &month<=12 ~ "Fall"))
#Getting the individual counts
nyc13 = nyc13 %>% group_by(dep) %>% summarise(count=n())

nyc13
```

```
## # A tibble: 3 x 2
```

```
##   dep    count
##   <chr>  <int>
## 1 Fall   38326
## 2 Spring 51829
## 3 Summer 38277
```

```r
#Plotting the data
ggplot(nyc13, aes(x=dep, y=count)) + geom_histogram(stat="identity", fill="blue") + geom_text(stat = "i
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



From the seasons category, we can see that the highest number of arrival and departure delays are in spring(jan-may). The number of flights that were delayed is displayed in the graph on top of the season.

5. We'd also like to know how much do delays depend on the time of day. Are there more delays in foggy morning hours? Late night when all the daily delays may accumulate? Create a visualization (graph or table) using a different approach than what you did above.

```r
#Departure delay

#Grouping by every hour to check the mean departure delay every hour.
nyc13 = flights %>% filter(dep_delay>0) %>% group_by(hour) %>% summarise(mean_delay = mean(dep_delay))
```

```r
#Plotting the data
p = ggplot(nyc13, aes(x=hour, y=mean_delay)) + geom_smooth() + scale_x_continuous(breaks = c(1:24))

#Making it an interactive plot!
ggplotly(p)
```

```r
#Arrival delay
#Grouping by every hour to check the mean arrival delay every hour.
nyc13 = flights %>% filter(arr_delay>0) %>% group_by(hour) %>% summarise(mean_delay = mean(arr_delay))

#Plotting the data
p = ggplot(nyc13, aes(x=hour, y=mean_delay)) + geom_smooth() + scale_x_continuous(breaks = c(1:24))

#Making it an interactive plot!
ggplotly(p)
```

From the above plot, we can see that there were no delays in the first 4 hours of the day. The mean of the departure delays is the highest in the 19th hour.

6:Do you see any problems with these questions(and answers)?

FOR EVERY QUESTION WITH DELAY, IT IS NOT SPECIFIED WHICH DELAY ARE WE CONSIDERING!!

1.3: Let's fly to portland

1: How many flights were there from NYC airports to Portland in 2013?

```r
#Filtering out all the flights with NYC as origin and PDX as destination
nyc14 = flights %>% filter((origin == "JFK" | origin == "LGA" | origin == "EWR") & dest=="PDX")

#Checking the number of rows
nrow(nyc14)
```

```
## [1] 1354
```

There are 1354 from NYC to Portland in 2013.

2: How many airlines fly from NYC to Portland? Which are these airlines (find the 2-letter abbreviations)? How many times did each of these go to Portland?

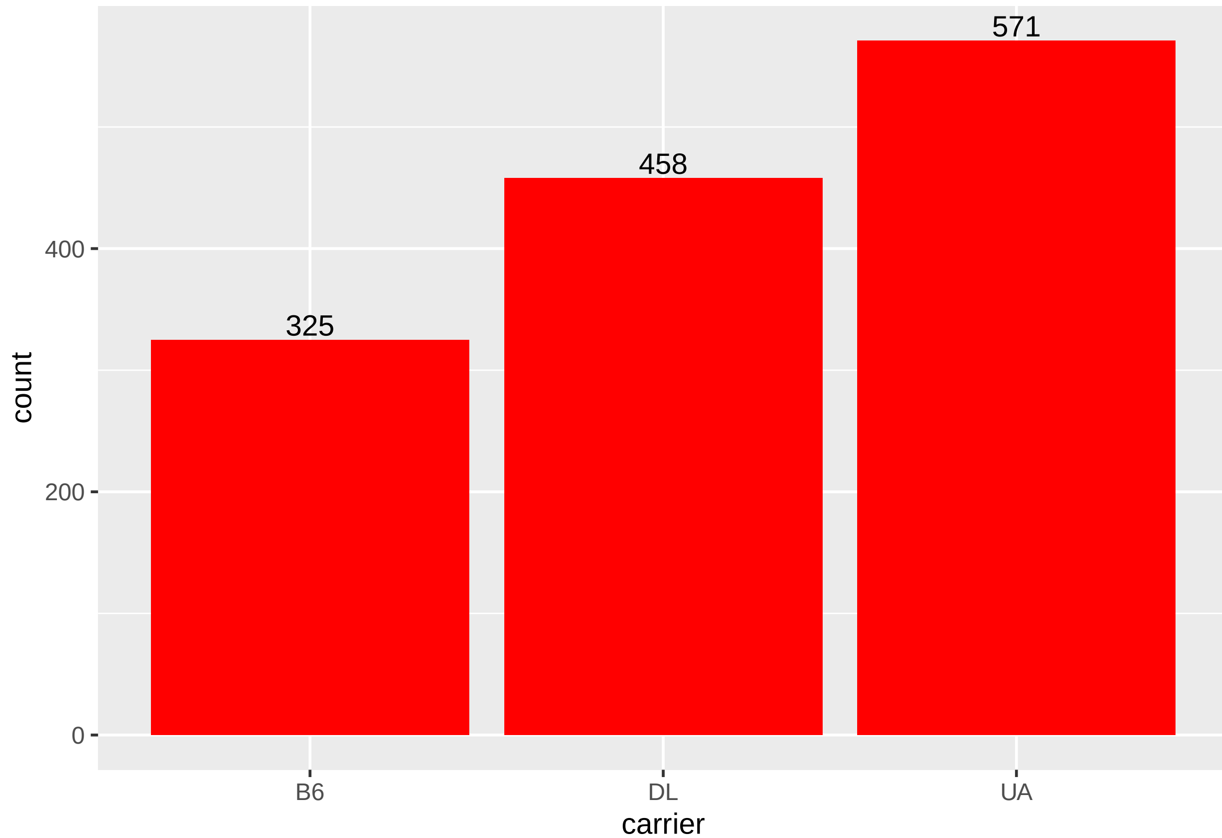
```r
#Grouping by carrier and summrizing
nyc15 = nyc14 %>% filter(is.na(carrier)==FALSE) %>% group_by(carrier) %>% summarise(count=n())

nyc15
```

```
## # A tibble: 3 x 2
##   carrier count
```

```
##   <chr>   <int>
## 1 B6        325
## 2 DL        458
## 3 UA        571
```

```
#Plotting the data
ggplot(nyc15, aes(x=carrier, y=count)) + geom_histogram(stat="identity", fill="red") + geom_text(stat =
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



There are 3 carriers that go from NYC to Portland, they are B6, DL and UA. B6 flew 325 times, DL flew 458 times and UA flew 571 times.

4: How many unique airliners fly from NYC to PDX?

```
#Grouping by tailnumber and summarizing
nyc16 = nyc14 %>% filter(is.na(tailnum)==FALSE) %>% group_by(tailnum) %>% summarise(count=n())

#Checking the number of rows and columns in the dataset.
dim(nyc16)
```

```
## [1] 491   2
```

There are 491 different airliners that fly from NYC to PDX.

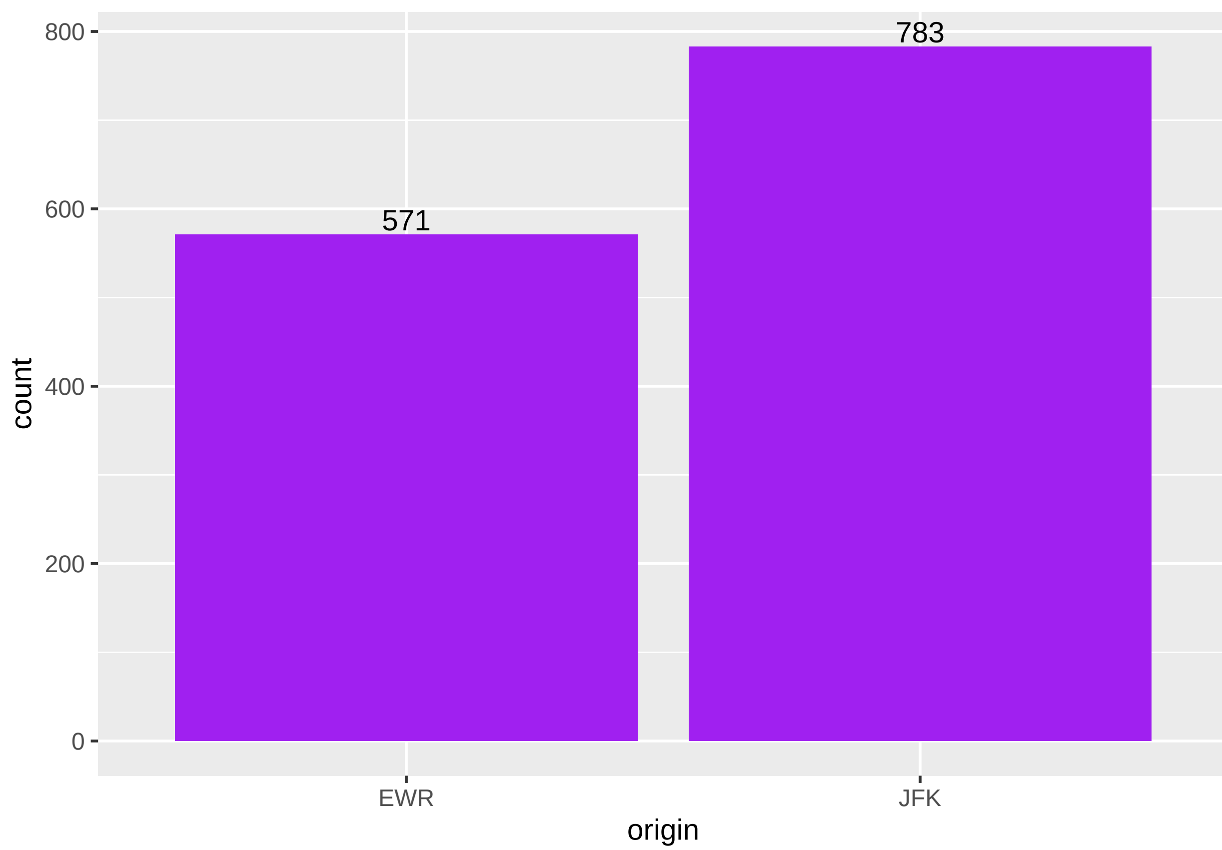
5: How many different airplanes arrived from each of the three NYC airports to Portland?

```
#Grouping by the origin and summarizing
nyc17 = nyc14 %>% group_by(origin) %>% summarise(count=n())

nyc17
```

```
## # A tibble: 2 x 2
##   origin count
##   <chr>  <int>
## 1 EWR      571
## 2 JFK      783
```

```
#Plotting the data
ggplot(nyc17, aes(x=origin, y=count)) + geom_histogram(stat="identity", fill="purple") + geom_text(stat
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



571 flights arrived from EWR, 783 flights arrived from JFK and no flights arrived from LGA.

6: What percentage of flights to Portland were delayed at departure by more than 15 minutes?

```
#Number of rows of flights with origin NYC and dest as PDX
nrow(nyc14)
```

```
## [1] 1354
```

```
#Filtering out departure delay greater than 15 minutes
nyc18 = nyc14 %>% filter(dep_delay>15)

#Getting the number of rows
nrow(nyc18)
```

```
## [1] 361
```

```
p = round((nrow(nyc18) / nrow(nyc14)) *100)
cat(paste("The percentage of NYC flights to portland with delay of more than 15 mins is:\n"))
```

```
## The percentage of NYC flights to portland with delay of more than 15 mins is:
```

```
p
```

```
## [1] 27
```

The percentage of NYC flights to portland with delay of more than 15 mins is 27%

7: And finally answer the question above for each origin airport separately. Is one of the airports noticeably worse than others?

```
#To check the total number of flights from each origin
nyc14 %>% group_by(origin) %>% summarise(count=n())
```

```
## # A tibble: 2 x 2
##   origin count
##   <chr>  <int>
## 1 EWR      571
## 2 JFK      783
```

```
#To check the flights that had more than 15 minutes of delay
#Categorizing as 1 for more than 15 minutes delay and 0 for not.
nyc15 = nyc14 %>% filter(dep_delay>0) %>% mutate(delay_15 = ifelse(dep_delay>15 , 1, 0)) %>% group_by(o

nyc15
```

```
## # A tibble: 4 x 3
## # Groups:   origin [2]
##   origin delay_15 count
##   <chr>     <dbl> <int>
## 1 EWR           0   143
## 2 EWR           1   168
## 3 JFK           0   177
## 4 JFK           1   193
```

Percentage of flights delayed for more than 15 mins from EWR: (168/571) * 100 = 29.42

Percentage of flights delayed for more than 15 mins from JFK: (193/783) * 100 = 24.64

To answer the question, there is no significant difference in both the origins, EWR is a higher percentage as compared to JFK.

1.4: Think about all this?

1: Do you see any issues with data?

Yes, there are issues with the data. Data for arrival delay and departure delay is NA for most of the columns. What does that mean? Does it mean that the flights were on time? This is not clear. The tailnumber for few flights is missing. The data is incomplete.

2: Ethical concerns?

Ethical concerns with this data could be about how accurate this is. If the data is not accurate enough, coming to conclusions with this data would be ethically wrong.

3: Can these questions be answered? Are these questions meaningful?

The questions are very vague. The questions on delay are not clear on which delay are we considering. Is it both arrival delay or departure delay or both?

Few of the questions, it's not specified if we have to remove null values. Few of the flights don't have the airtime. These kind of discrpencies in the data are not enough for a thorough analysis.