INFO7250 - US On-time Performance Flight Data Analysis

Bowei Wang, Dongyue Li, Zelong Zhao, Xiaoyu Zheng

12 Dec 2016

Contents

1. Introduction

In this project, we used R to analysis flights dataset and do text mining on airline reviews dataset. We investigated the two datasets from several aspects: 1)Number of flights per carrier 2)Flight cancellation rate per carrier 3)Number of flights operated by day of the week 4)Number of flights operated by month 5)Top 10 Worst Airports by Average Arrival Delay Time and Delay Rate 6)Top 10 Worst Airports by Average Departure Delay Time and Delay Rate 7)On-time arrival performance 8)Departure delay distribution over the day period 9)Departure and arrival airport count by carrier 10)Text mining on passengers reviews for each airline(carrier) and plot word cloud for each airline(carrier).

2. Dataset description

The flight dataset which is used for our investigation is derived from the Bureau of Transportation Statistics. Our goal is to investigate flights data of past year (from Oct 2015 to Sep 2016) provided by BTS.

This dataset contains on-time arrival data for non-stop domestic flights by major air carriers, and provides such additional items as departure and arrival delays, origin and destination airports, flight numbers, scheduled and actual departure and arrival times, cancelled or diverted flights, taxi-out and taxi-in times, air time, and non-stop distance.

Another dataset we used for text mining is passengers airline reviews from SKYTRAX. It contains 12 airlines reviews including AA, AS, B6, DL, F9, HA, MQ, NK, OO, UA, VX, WN.

3. Dataset preparation

3.1 Load libraries

3.2 Load the flights dataset

Loading multiple .csv files into the same data frame

```
folder <- "/Users/dongyueli/Desktop/flight_raw_data_new/"  # path to folder that holds multiple .csv fi
file_list <- list.files(path=folder, pattern="*.csv") # create list of all .csv files in folder
# read in each .csv file in file_list and rbind them into a data frame called data
data <-
  do.call("rbind",
          lapply(file_list,
                 function(x)
                 read.csv(paste(folder, x, sep=''),
                 stringsAsFactors = FALSE)))
str(data)
```

```
## 'data.frame':    5666512 obs. of  44 variables:
##  $ YEAR                 : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ QUARTER              : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ MONTH                : int  10 10 10 10 10 10 10 10 10 10 ...
##  $ DAY_OF_WEEK          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ FL_DATE              : chr  "2015-10-05" "2015-10-05" "2015-10-05" "2015-10-05" ...
##  $ UNIQUE_CARRIER       : chr  "DL" "DL" "DL" "DL" ...
##  $ FL_NUM               : int  1847 1848 1849 1851 1852 1853 1854 1855 1856 1856 ...
##  $ ORIGIN_CITY_MARKET_ID: int  34783 31295 31057 30397 30397 33485 31295 30397 30397 33360 ...
##  $ ORIGIN               : chr  "SGF" "DTW" "CLT" "ATL" ...
##  $ ORIGIN_CITY_NAME     : chr  "Springfield, MO" "Detroit, MI" "Charlotte, NC" "Atlanta, GA" ...
##  $ ORIGIN_STATE_ABR     : chr  "MO" "MI" "NC" "GA" ...
##  $ DEST_CITY_MARKET_ID  : int  30397 31703 31295 31650 30559 30397 30325 30852 33360 30397 ...
##  $ DEST                 : chr  "ATL" "LGA" "DTW" "MSP" ...
##  $ DEST_CITY_NAME       : chr  "Atlanta, GA" "New York, NY" "Detroit, MI" "Minneapolis, MN" ...
##  $ DEST_STATE_ABR       : chr  "GA" "NY" "MI" "MN" ...
##  $ CRS_DEP_TIME         : int  605 730 600 1454 1100 600 1555 1220 908 1116 ...
##  $ DEP_TIME             : int  604 740 557 1459 1059 554 1547 1217 903 1112 ...
##  $ DEP_DELAY            : num  -1 10 -3 5 -1 -6 -8 -3 -5 -4 ...
##  $ DEP_DELAY_NEW        : num  0 10 0 5 0 0 0 0 0 0 ...
##  $ DEP_DEL15            : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ DEP_TIME_BLK         : chr  "0600-0659" "0700-0759" "0600-0659" "1400-1459" ...
##  $ TAXI_OUT             : num  31 16 17 25 21 12 15 12 16 9 ...
##  $ WHEELS_OFF           : int  635 756 614 1524 1120 606 1602 1229 919 1121 ...
##  $ WHEELS_ON            : int  855 917 726 1635 1300 844 1640 1346 1020 1233 ...
##  $ TAXI_IN              : num  7 7 4 3 29 11 7 4 6 13 ...
##  $ CRS_ARR_TIME         : int  851 930 750 1642 1335 914 1705 1408 1036 1257 ...
##  $ ARR_TIME             : int  902 924 730 1638 1329 855 1647 1350 1026 1246 ...
##  $ ARR_DELAY            : num  11 -6 -20 -4 -6 -19 -18 -18 -10 -11 ...
##  $ ARR_DELAY_NEW        : num  11 0 0 0 0 0 0 0 0 0 ...
##  $ ARR_DEL15            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ARR_TIME_BLK         : chr  "0800-0859" "0900-0959" "0700-0759" "1600-1659" ...
##  $ CANCELLED            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CANCELLATION_CODE    : chr  "" "" "" "" ...
##  $ DIVERTED             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CRS_ELAPSED_TIME     : num  106 120 110 168 335 134 190 108 88 101 ...
##  $ ACTUAL_ELAPSED_TIME  : num  118 104 93 159 330 121 180 93 83 94 ...
##  $ AIR_TIME             : num  80 81 72 131 280 98 158 77 61 72 ...
##  $ DISTANCE             : num  563 502 500 907 2182 ...
##  $ CARRIER_DELAY        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ WEATHER_DELAY        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ NAS_DELAY            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ SECURITY_DELAY       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ LATE_AIRCRAFT_DELAY  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ X                    : logi  NA NA NA NA NA NA ...
```

```
data$X <- NULL
flight.df <- data
dim(flight.df)
```

```
## [1] 5666512      43
```

```
str(flight.df)
```

```
## 'data.frame':    5666512 obs. of  43 variables:
##  $ YEAR                 : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ QUARTER              : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ MONTH                : int  10 10 10 10 10 10 10 10 10 10 ...
##  $ DAY_OF_WEEK          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ FL_DATE              : chr  "2015-10-05" "2015-10-05" "2015-10-05" "2015-10-05" ...
##  $ UNIQUE_CARRIER       : chr  "DL" "DL" "DL" "DL" ...
##  $ FL_NUM               : int  1847 1848 1849 1851 1852 1853 1854 1855 1856 1856 ...
##  $ ORIGIN_CITY_MARKET_ID: int  34783 31295 31057 30397 30397 33485 31295 30397 30397 33360 ...
##  $ ORIGIN               : chr  "SGF" "DTW" "CLT" "ATL" ...
##  $ ORIGIN_CITY_NAME     : chr  "Springfield, MO" "Detroit, MI" "Charlotte, NC" "Atlanta, GA" ...
##  $ ORIGIN_STATE_ABR     : chr  "MO" "MI" "NC" "GA" ...
##  $ DEST_CITY_MARKET_ID  : int  30397 31703 31295 31650 30559 30397 30325 30852 33360 30397 ...
##  $ DEST                 : chr  "ATL" "LGA" "DTW" "MSP" ...
##  $ DEST_CITY_NAME       : chr  "Atlanta, GA" "New York, NY" "Detroit, MI" "Minneapolis, MN" ...
##  $ DEST_STATE_ABR       : chr  "GA" "NY" "MI" "MN" ...
##  $ CRS_DEP_TIME         : int  605 730 600 1454 1100 600 1555 1220 908 1116 ...
##  $ DEP_TIME             : int  604 740 557 1459 1059 554 1547 1217 903 1112 ...
##  $ DEP_DELAY            : num  -1 10 -3 5 -1 -6 -8 -3 -5 -4 ...
##  $ DEP_DELAY_NEW        : num  0 10 0 5 0 0 0 0 0 0 ...
##  $ DEP_DEL15            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DEP_TIME_BLK         : chr  "0600-0659" "0700-0759" "0600-0659" "1400-1459" ...
```

```
##  $ TAXI_OUT             : num  31 16 17 25 21 12 15 12 16 9 ...
##  $ WHEELS_OFF           : int  635 756 614 1524 1120 606 1602 1229 919 1121 ...
##  $ WHEELS_ON            : int  855 917 726 1635 1300 844 1640 1346 1020 1233 ...
##  $ TAXI_IN              : num  7 7 4 3 29 11 7 4 6 13 ...
##  $ CRS_ARR_TIME         : int  851 930 750 1642 1335 914 1705 1408 1036 1257 ...
##  $ ARR_TIME             : int  902 924 730 1638 1329 855 1647 1350 1026 1246 ...
##  $ ARR_DELAY            : num  11 -6 -20 -4 -6 -19 -18 -18 -10 -11 ...
##  $ ARR_DELAY_NEW        : num  11 0 0 0 0 0 0 0 0 0 ...
##  $ ARR_DEL15            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ARR_TIME_BLK         : chr  "0800-0859" "0900-0959" "0700-0759" "1600-1659" ...
##  $ CANCELLED            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CANCELLATION_CODE    : chr  "" "" "" "" ...
##  $ DIVERTED             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CRS_ELAPSED_TIME     : num  106 120 110 168 335 134 190 108 88 101 ...
##  $ ACTUAL_ELAPSED_TIME  : num  118 104 93 159 330 121 180 93 83 94 ...
##  $ AIR_TIME             : num  80 81 72 131 280 98 158 77 61 72 ...
##  $ DISTANCE             : num  563 502 500 907 2182 ...
##  $ CARRIER_DELAY        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ WEATHER_DELAY        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ NAS_DELAY            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ SECURITY_DELAY       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ LATE_AIRCRAFT_DELAY  : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
head(flight.df)
```

```
##   YEAR QUARTER MONTH DAY_OF_WEEK    FL_DATE UNIQUE_CARRIER FL_NUM
## 1 2015       4    10           1 2015-10-05             DL   1847
## 2 2015       4    10           1 2015-10-05             DL   1848
## 3 2015       4    10           1 2015-10-05             DL   1849
## 4 2015       4    10           1 2015-10-05             DL   1851
## 5 2015       4    10           1 2015-10-05             DL   1852
## 6 2015       4    10           1 2015-10-05             DL   1853
##   ORIGIN_CITY_MARKET_ID ORIGIN ORIGIN_CITY_NAME ORIGIN_STATE_ABR
## 1                 34783    SGF  Springfield, MO               MO
## 2                 31295    DTW      Detroit, MI               MI
## 3                 31057    CLT    Charlotte, NC               NC
## 4                 30397    ATL      Atlanta, GA               GA
## 5                 30397    ATL      Atlanta, GA               GA
## 6                 33485    MSN      Madison, WI               WI
##   DEST_CITY_MARKET_ID DEST  DEST_CITY_NAME DEST_STATE_ABR CRS_DEP_TIME
## 1               30397  ATL     Atlanta, GA             GA          605
## 2               31703  LGA    New York, NY             NY          730
## 3               31295  DTW     Detroit, MI             MI          600
## 4               31650  MSP Minneapolis, MN             MN         1454
## 5               30559  SEA     Seattle, WA             WA         1100
## 6               30397  ATL     Atlanta, GA             GA          600
##   DEP_TIME DEP_DELAY DEP_DELAY_NEW DEP_DEL15 DEP_TIME_BLK TAXI_OUT
## 1      604        -1             0         0    0600-0659       31
## 2      740        10            10         0    0700-0759       16
## 3      557        -3             0         0    0600-0659       17
## 4     1459         5             5         0    1400-1459       25
## 5     1059        -1             0         0    1100-1159       21
## 6      554        -6             0         0    0600-0659       12
##   WHEELS_OFF WHEELS_ON TAXI_IN CRS_ARR_TIME ARR_TIME ARR_DELAY
## 1        635       855       7          851      902        11
```

```
## 2        756       917       7          930      924        -6
## 3        614       726       4          750      730       -20
## 4       1524      1635       3         1642     1638        -4
## 5       1120      1300      29         1335     1329        -6
## 6        606       844      11          914      855       -19
##   ARR_DELAY_NEW ARR_DEL15 ARR_TIME_BLK CANCELLED CANCELLATION_CODE
## 1            11         0    0800-0859         0
## 2             0         0    0900-0959         0
## 3             0         0    0700-0759         0
## 4             0         0    1600-1659         0
## 5             0         0    1300-1359         0
## 6             0         0    0900-0959         0
##   DIVERTED CRS_ELAPSED_TIME ACTUAL_ELAPSED_TIME AIR_TIME DISTANCE
## 1        0              106                 118       80      563
## 2        0              120                 104       81      502
## 3        0              110                  93       72      500
## 4        0              168                 159      131      907
## 5        0              335                 330      280     2182
## 6        0              134                 121       98      707
##   CARRIER_DELAY WEATHER_DELAY NAS_DELAY SECURITY_DELAY LATE_AIRCRAFT_DELAY
## 1            NA            NA        NA             NA                  NA
## 2            NA            NA        NA             NA                  NA
## 3            NA            NA        NA             NA                  NA
## 4            NA            NA        NA             NA                  NA
## 5            NA            NA        NA             NA                  NA
## 6            NA            NA        NA             NA                  NA
```

We see that there are 5666512 number of flights during 2015.10.01 to 2016.09.01 and there are 41 variables selected.

4. Variable summaries and visualizations

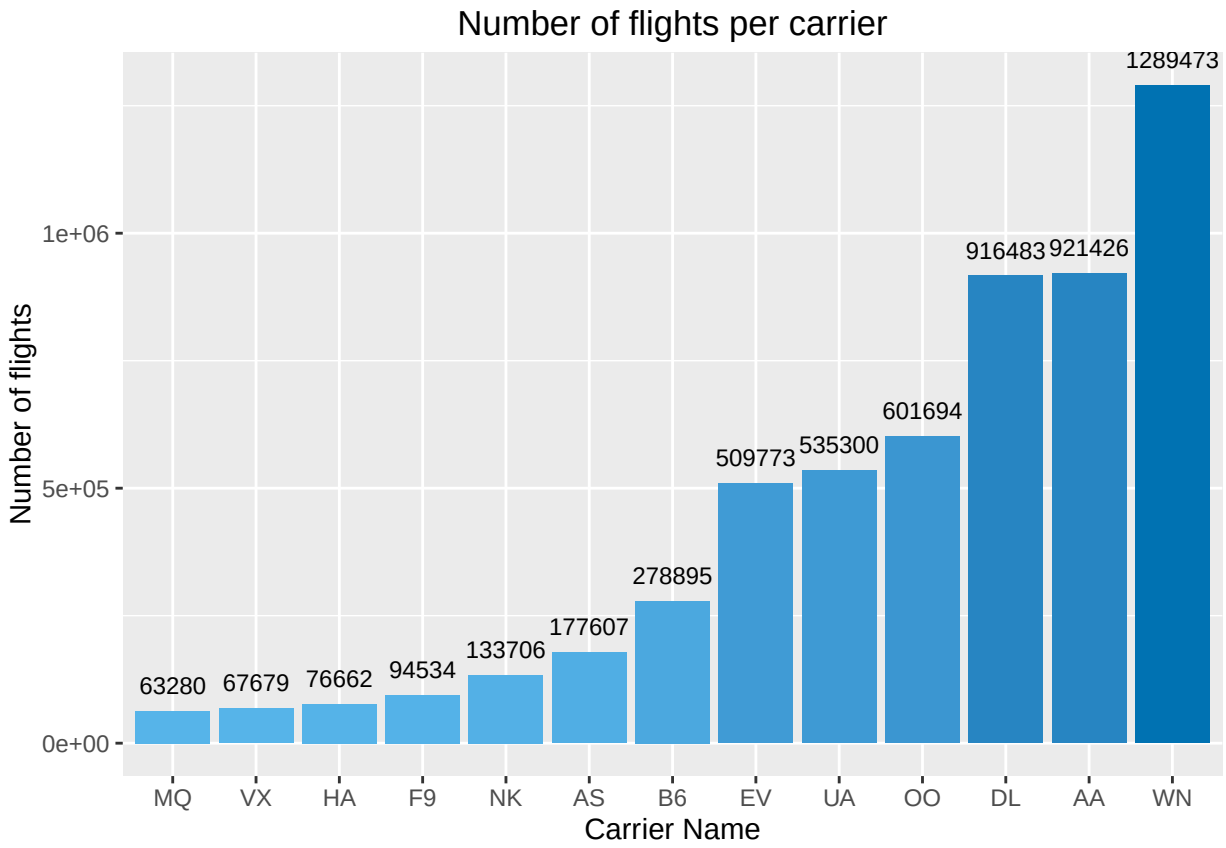
In the next few sections, we observe the significant single and multiple variables for our analysis.

4.1 Number of flights per carrier

We analyze the total number of flights for each carrier to get an understanding on the carriers and the number of flights they are operating during the last year.

We visualize this using a bar graph.

```
carrier_count <- count(flight.df, UNIQUE_CARRIER)
p1 <- ggplot(carrier_count, aes(x = reorder(UNIQUE_CARRIER, n), y = n, fill = n)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = n), vjust=-1, position = position_dodge(0.9),size = 3) +
  ggtitle("Number of flights per carrier") +
  xlab("Carrier Name") + ylab("Number of flights") +
  scale_fill_gradient(low = '#56B4E9', high = '#0072B2')+
  guides(fill=FALSE)
p1
```



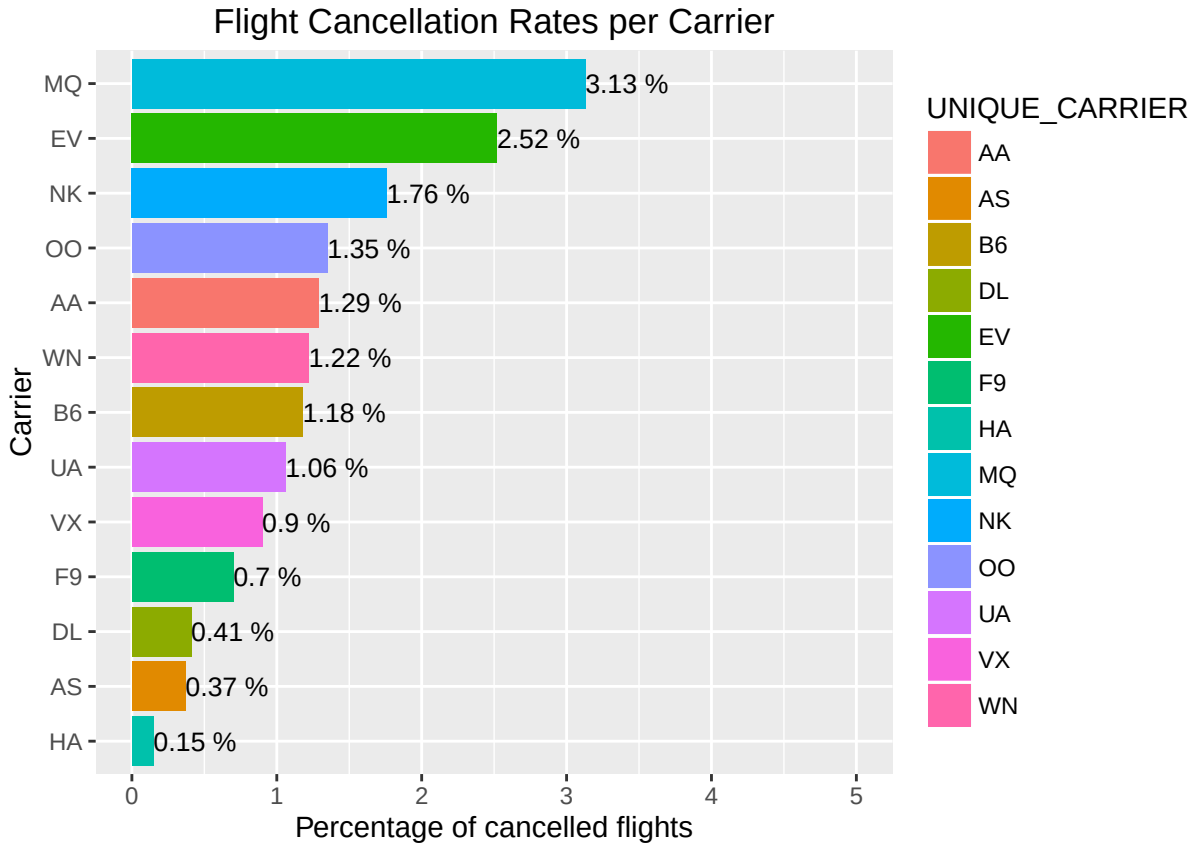

```
# gg1 <- ggplotly(p1)
# plotly_POST(p1, filename = "NumberOfFlightsPerCarrier")
```

We observe that Southwest Airlines(WN) has the largest number of flights followed by American Airlines(AA). Envoy Air(MQ) has the least.

4.2 Flight Cancellation Rates per Carrier

We analyze the number of flights canceled by each carrier and we represent it using a bar graph.

```
p2 <- flight.df %>%
  group_by(UNIQUE_CARRIER) %>%
  summarise(pct = (length(which(CANCELLED==1))*100/length(CANCELLED)) %>%
              round(2)) %>%
  ggplot(aes(x = reorder(UNIQUE_CARRIER,pct),y=pct, fill = UNIQUE_CARRIER))+
  geom_bar(stat="identity", position = 'dodge') +
  geom_text(aes(label=paste(pct,"%")), vjust=0.5, hjust=0, color="black",
            position = position_dodge(0.8), size=3.5)+
  coord_flip() +
  ggtitle("Flight Cancellation Rates per Carrier") +
  xlab("Carrier") + ylab("Percentage of cancelled flights")+
  ylim(0, 5)
p2
```



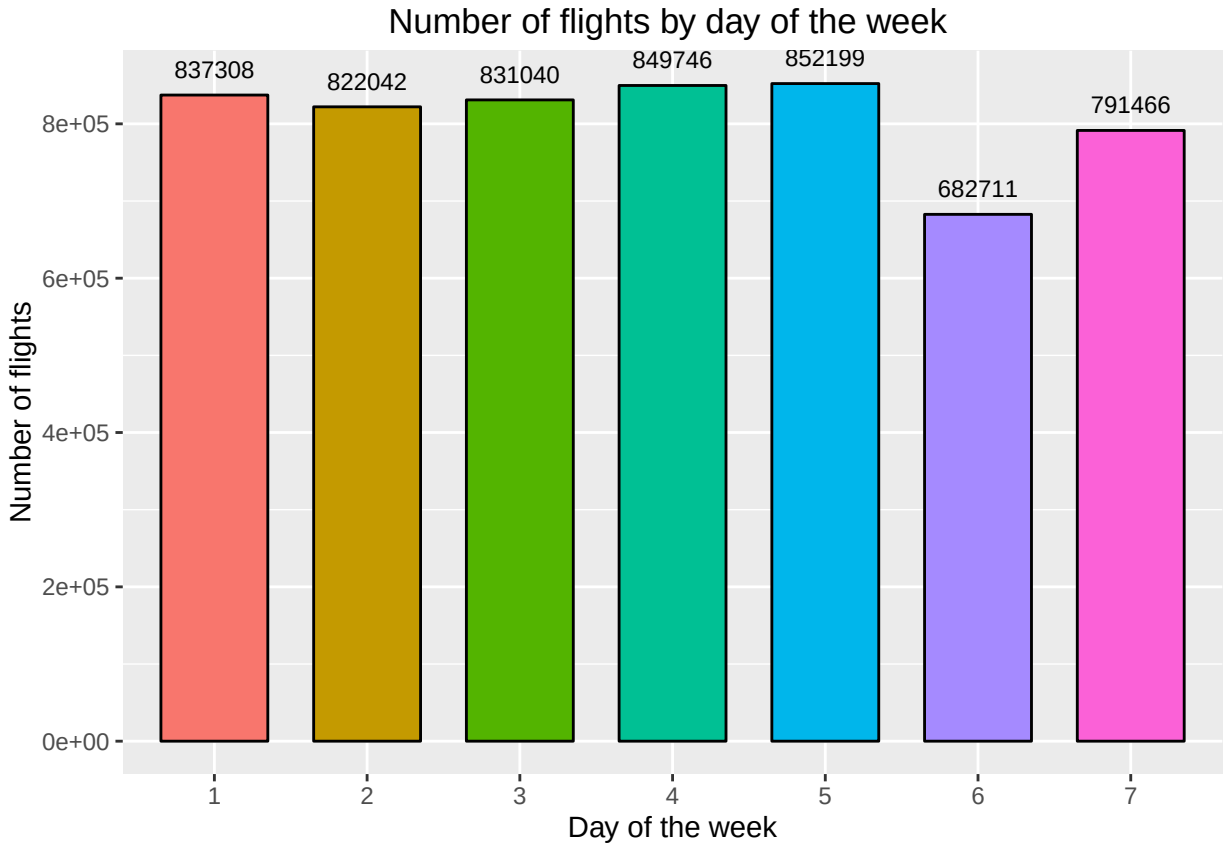

```
# gg2 <- ggplotly(p2)
# plotly_POST(p2, filename = "FlightCancellationRatesPerCarrier")
```

We observe that Envoy Air (MQ) has the maximum percentage of cancellations followed by EV (ExpressJet Airlines). HA (Hawaiian Airlines) has the least percentage of cancellations.

4.3 Number of flights by day of the week

In this analysis we understand the distribution of number of flights over each day of the week and visualize the number of flights operating per week using bar chart. This will help us in deeper understanding of flight delay patterns.

```
flight_count_week <- count(flight.df, DAY_OF_WEEK)
p3 <- ggplot(flight_count_week, aes(x = reorder(DAY_OF_WEEK, DAY_OF_WEEK), y = n, fill = factor(DAY_OF_
  geom_bar(stat = "identity", width = 0.7,color = "black") +
  geom_text(aes(label = n), vjust=-1, position = position_dodge(0.9),size = 3) +
  ggtitle("Number of flights by day of the week ") +
  xlab("Day of the week") + ylab("Number of flights")+
  guides(fill=FALSE)
p3
```

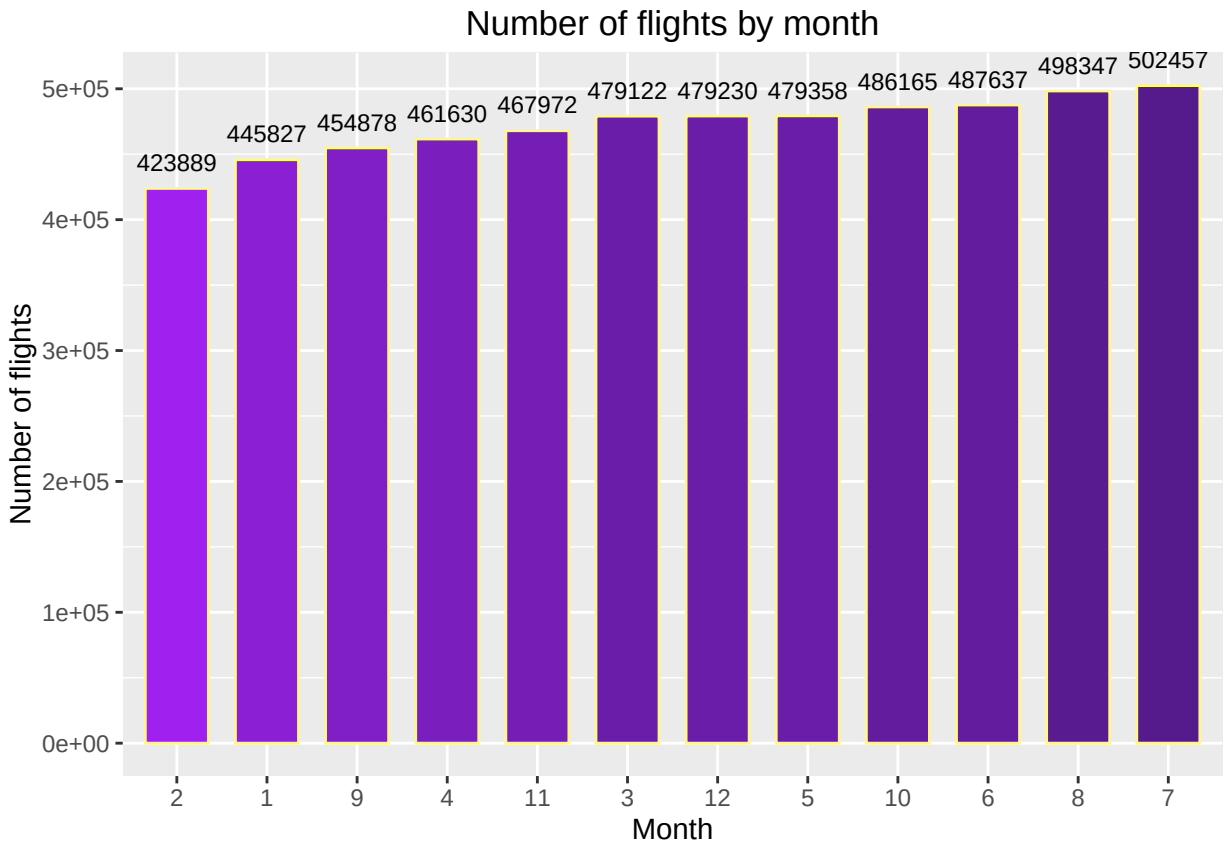



```
# gg3 <- ggplotly(p3)
# plotly_POST(p3, filename = "NumberOfFlightsByDayOfTheWeek")
```

We observe that higher number of flights are operated on Thursday and Friday. A samll number of flights are operated on Saturday.

4.4 Number of flights by month

```
flight_count_month <- count(flight.df, MONTH)
p4 <- ggplot(flight_count_month, aes(x = reorder(MONTH, n), y = n, fill = n)) +
  geom_bar(stat = "identity", width = 0.7,color = "khaki1") +
  geom_text(aes(label = n), vjust=-1, position = position_dodge(0.9),size = 3) +
  ggtitle("Number of flights by month") +
  xlab("Month") + ylab("Number of flights")+
  scale_fill_gradient(low = 'purple', high = 'purple4')+
  guides(fill=FALSE)
p4
```




```
# gg4 <- ggplotly(p4)
# plotly_POST(p4, filename = "NumberOfFlightsByMonth")
```

4.5 Top 10 Worst Airports by Average Arrival Delay Time and Delay Rate

In this section, we analyze the average delay time and percent delayed for top 10 destination airports(that is, airports having largest arrival delay).

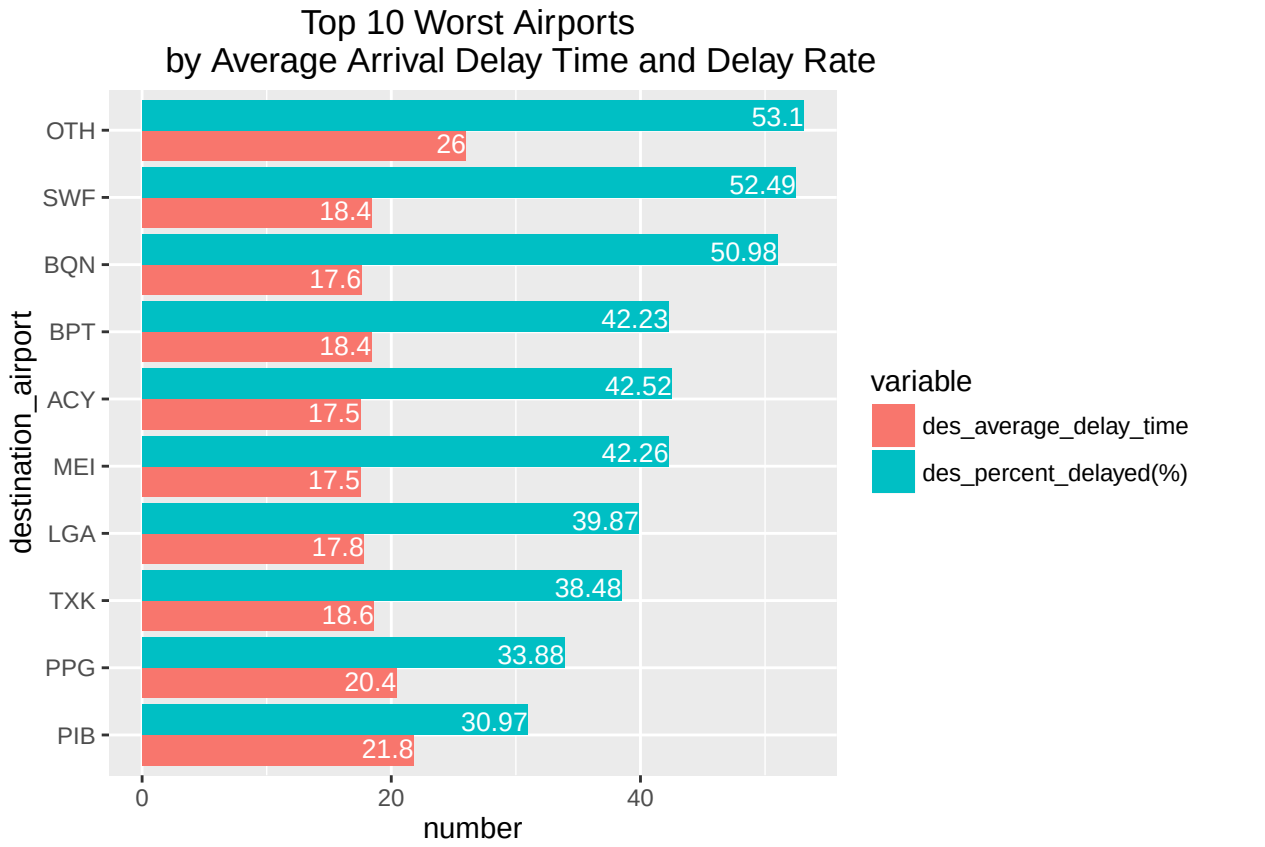
We group by `DEST` and calculate average arrival delay time and percent delayed for each destination airport and sort it by choosing the top 10 records that have the largest average delay time by using `top_n`.

Finally, we use `melt` function to stack the columns `des_average_delay_time` and `des_percent_delayed(%)` into a single column to be clearly displayed in graph.

```
flight.df %>%
  select(DEST, ARR_DELAY_NEW) %>%
  group_by(DEST)%>%
  summarise("des_average_delay_time" =
              mean(ARR_DELAY_NEW,na.rm = TRUE) %>% round(1),
            "des_percent_delayed(%)" =
              (length(which(ARR_DELAY_NEW > 0))*100/length(which(ARR_DELAY_NEW >= 0))) %>%
              round(2)) %>%
  top_n(10,des_average_delay_time) %>%
  melt(id.vars = c("DEST")) -> graph4.df
```

We plot a bar chart showing average delay time and percent delayed for top 10 destination airports that have the largest arrive delay time. Below code represents the bar graph which represents total number along x-axis and destination airport along y-axis.

```
p5<- graph4.df %>%
  ggplot(aes(x = reorder(DEST, value),
             y = value, fill = variable) ) +
  geom_bar(stat="identity", position = 'dodge') +
  ggtitle("Top 10 Worst Airports
          by Average Arrival Delay Time and Delay Rate") +
  geom_text(aes(label=value), vjust=0.5, hjust=1, color="white",
            position = position_dodge(0.8), size=3.5)+
  xlab("destination_airport") +
  ylab("number") +
  coord_flip()
p5
```




```
# gg5 <- ggplotly(p5)
# plotly_POST(p5, filename = "Top10WorstAirportsByAverageArrivalDelayTimeAndDelayRate")
```

We observe that during last year, all top 10 airports that have highest arrival delay have more than 17 minutes arrival delay time and more than 30% flights arrived late. OTH (Southwest Oregon Regional Airport) as the destination airport has the maximum average delay time of 26 minutes and its percent delayed is also high that is 53.1% of flights that arrived late at this airport.

4.6 Top 10 Worst Airports by Average Depature Delay Time and Delay Rate

We analyze the average delay time and percent delayed for top 10 origin airports that have the largest departure delay time.

We group by `ORIGIN` and calculate average departure delay time and percent delayed for each origin airport

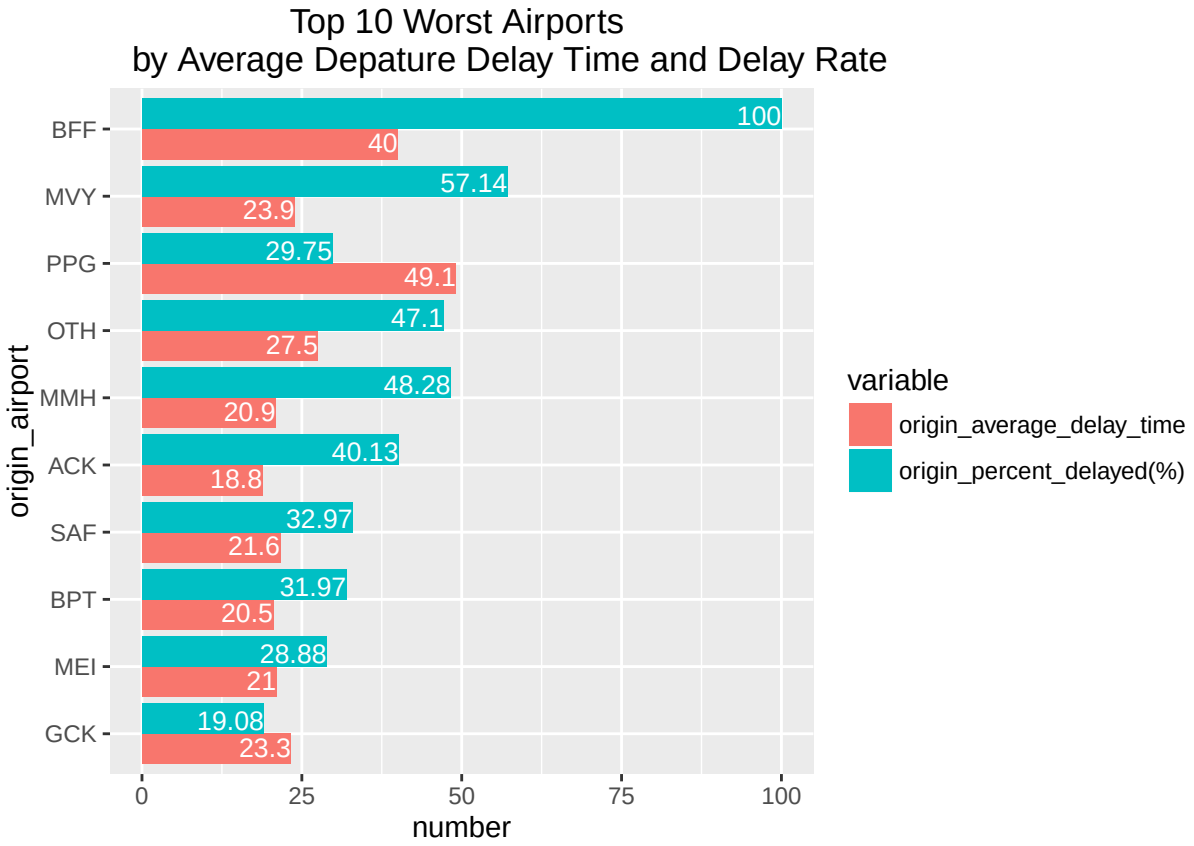
and sort it by choosing the top 10 records that have the largest average delay time by using `top_n`.

Finally, we use `melt` function to stack the columns `origin_average_delay_time` and `origin_percent_delayed(%)` into a single column to be clearly displayed in graph.

```
flight.df %>%
  select(ORIGIN, DEP_DELAY_NEW) %>%
  group_by(ORIGIN)%>%
  summarise("origin_average_delay_time" =
              mean(DEP_DELAY_NEW,na.rm = TRUE) %>% round(1),
            "origin_percent_delayed(%)" =
              (length(which(DEP_DELAY_NEW > 0))*100/length(which(DEP_DELAY_NEW >= 0))) %>%
              round(2)) %>%
  top_n(10,origin_average_delay_time) %>%
  melt(id.vars = c("ORIGIN")) -> graph5.df
```

We plot a bar chart showing average delay time and percent delayed for top 10 origin airports that have the largest departure delay time. Below code represents the bar graph which represents total number along x-axis and origin airport along y-axis.

```
p6<- graph5.df %>%
  ggplot(aes(x = reorder(ORIGIN,value),
             y = value, fill = variable) ) +
  geom_bar(stat="identity", position = 'dodge') +
  ggtitle("Top 10 Worst Airports
          by Average Depature Delay Time and Delay Rate") +
  geom_text(aes(label=value), vjust=0.5, hjust=1, color="white",
            position = position_dodge(0.8), size=3.5)+
  xlab("origin_airport") +
  ylab("number") +
  coord_flip()
p6
```




```
# gg6 <- ggplotly(p6)
# plotly_POST(p6, filename = "Top10WorstAirportsByAverageDepatureDelayTimeAndDelayRate")
```

We observe that during last year, all top 10 airports that have highest arrival delay time have more than 18 minutes arrival delay time and more than 19% flights arrived late. BFF (Western Nebraska Regional Airport) has the maximum percent delayed which is 100% of flights departured late at this airport and its average delay time is also high that is 40 minutes. PPG (Pago Pago International Airport) has the maximum average delay time of 49.1 minutes and 29.15% of flights departured late at this airport.

4.7 On-time arrival performance

In this section we analyze the airline service quality performance and visualize them on a bar graph. We researched the airline on-time statistics and delay causes and observed that the displayed numbers are rounded and may not add up to the total.

First, we calculate 5 delay reasons average percentage. While doing this calculation we observe a business rule that a flight is considered delayed when it arrives 15 or more minutes ahead of the scheduled time. When multiple causes are assigned to one delayed flight, each cause is prorated based on delayed minutes it is responsible for.

We calculate all 5 reasons responsible percentage in each flights and then get the mean percentage value for each reason.

```
delay_reason_avg_pct <- flight.df %>%
  filter(ARR_DELAY_NEW >= 15 ) %>%
  select(CARRIER_DELAY,WEATHER_DELAY, NAS_DELAY,
         SECURITY_DELAY,LATE_AIRCRAFT_DELAY) %>%
  mutate(rowSum = rowSums(.)) %>%
```

```
  transmute(CARRIER_DELAY_PCT = CARRIER_DELAY / rowSum,
         WEATHER_DELAY_PCT = WEATHER_DELAY / rowSum,
         NAS_DELAY_PCT = NAS_DELAY / rowSum,
         SECURITY_DELAY_PCT = SECURITY_DELAY / rowSum,
         LATE_AIRCRAFT_DELAY_PCT = LATE_AIRCRAFT_DELAY / rowSum) %>%
  colMeans()
delay_reason_avg_pct
```

```
##       CARRIER_DELAY_PCT       WEATHER_DELAY_PCT           NAS_DELAY_PCT
##             0.297590843             0.030790997             0.309002393
##      SECURITY_DELAY_PCT LATE_AIRCRAFT_DELAY_PCT
##             0.002053001             0.360562766
```

From above, we learned that when a flight was delayed, the percentage of causes of the plane was delayed are 29.76%(carrier), 3.08%(weather), 30.9%(NAS), 0.21% (security) and 36.06%(late aircraft) respectively. So we can calculate the number of flights which is delayed by each reason.

```
delay_reason_count <-
  delay_reason_avg_pct * length(which(flight.df$ARR_DELAY_NEW>=15))
names(delay_reason_count)<-c("CARRIER_DELAY_COUNT","WEATHER_DELAY_COUNT",
                         "NAS_DELAY_COUNT","SECURITY_DELAY_COUNT",
                         "LATE_AIRCRAFT_DELAY_COUNT")
delay_reason_count
```

```
##       CARRIER_DELAY_COUNT       WEATHER_DELAY_COUNT
##                 287391.512                 29735.697
##           NAS_DELAY_COUNT      SECURITY_DELAY_COUNT
##                 298411.954                  1982.638
## LATE_AIRCRAFT_DELAY_COUNT
##                 348205.199
```

From above we learned that from all 965727 delayed flights, there are 287392 flights delayed due to carrier, 29736 flights delayed due to weather, 298412 flights due to NAS, 1983 flights due to security, and 348205 due to late aircraft.

Next, we calculate the count of flights that arrived on time, are cancelled and diverted.

```
arrive_on_time_count <- length(which(flight.df$ARR_DELAY_NEW<15))
arrive_on_time_count
```

```
## [1] 4618754
```

```
cancelled_count <- length(which(flight.df$CANCELLED==1))
cancelled_count
```

```
## [1] 67656
```

```
diverted_count <- length(which(flight.df$DIVERTED==1))
diverted_count
```

```
## [1] 14375
```

After, we calculated the ratio of above data to all flights and convert them into a data frame.

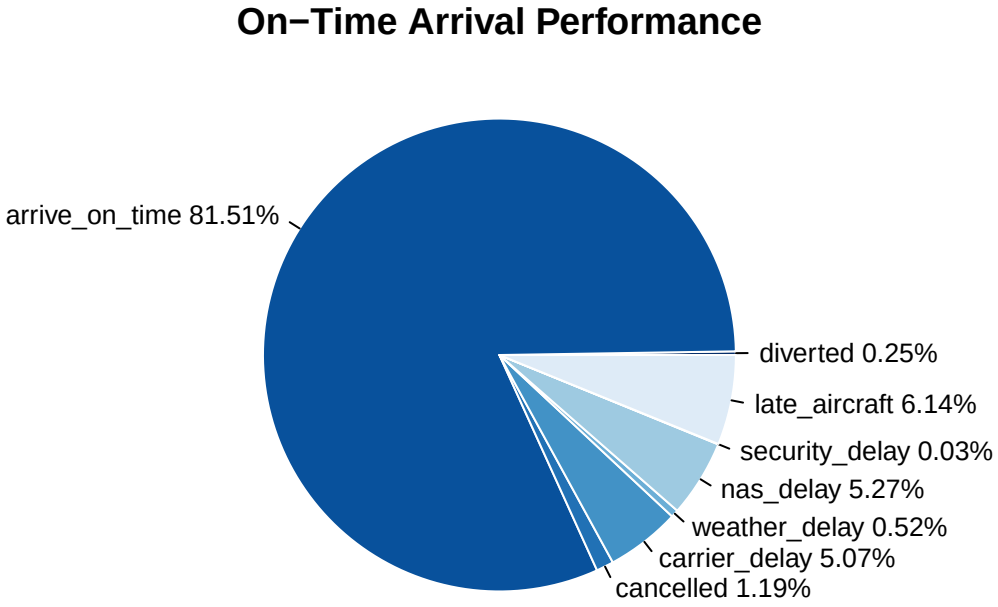
```
slices <- c(diverted_count,arrive_on_time_count,cancelled_count,
            delay_reason_count[1],
            delay_reason_count[2],
            delay_reason_count[3],
            delay_reason_count[4],
```

```
            delay_reason_count[5])
lbls <- c("diverted","arrive_on_time","cancelled",
          "carrier_delay", "weather_delay" , "nas_delay",
          "security_delay", "late_aircraft")
pct <- round(slices/sum(slices)*100,2)
graph6.df <- data.frame(lbls,pct)
graph6.df
```

```
##             lbls   pct
## 1       diverted  0.25
## 2 arrive_on_time 81.51
## 3      cancelled  1.19
## 4  carrier_delay  5.07
## 5  weather_delay  0.52
## 6      nas_delay  5.27
## 7 security_delay  0.03
## 8  late_aircraft  6.14
```

```
lbls1 <- paste(lbls, pct) # add percents to labels
lbls1 <- paste(lbls1,"%",sep="") # ad % to labels

pie(pct, labels = lbls1, col=rev(blues9), border="white",
    main="On-Time Arrival Performance", radius = 1, cex = 0.85)
```

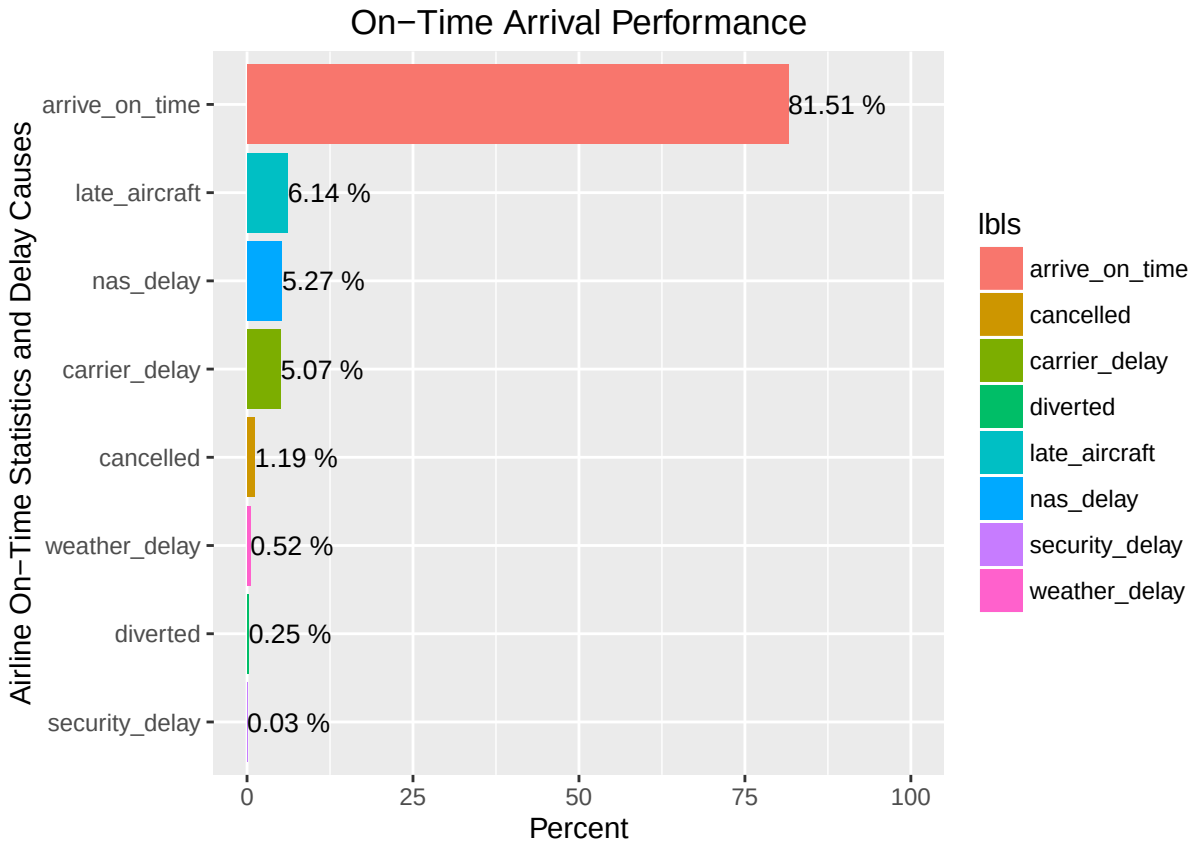



```
# gg7 <- plot_ly(values = pct, labels = lbls1, type = 'pie',
#                insidetextfont = list(color = '#FFFFFF'),
#                marker = list(colors = rev(blues9),line = list(color = '#FFFFFF', width = 1))) %>%
#         layout(title = 'On-Time Arrival Performance')
# plotly_POST(gg7, filename = "On-TimeArrivalPerformancePie")
```

Finally, we plot a bar chart by using the above data.

```
p8<- graph6.df %>%
  ggplot(aes(x = reorder(lbls,pct), y =pct , fill = lbls))+
  geom_bar(stat="identity")+
  ggtitle("On-Time Arrival Performance") +
```

```
  geom_text(aes(label=paste(pct,"%")), vjust=0.5, hjust=0, color="black",
            position = position_dodge(0.8), size=3.5)+
  xlab("Airline On-Time Statistics and Delay Causes") +
  ylab("Percent") +
  coord_flip()+
  ylim(0, 100)
p8
```




```
# gg8 <- ggplotly(p8)
# plotly_POST(p8, filename = "On-TimeArrivalPerformanceBar")
```

We observe that 81.51% flights arrived on time, 1.19% flights were cancelled and 0.25% flights were diverted. About 18.49% flights are delayed and most of the delays occur due to `late aircraft` which contributes to 6.14% of all flights. It is followed by `nas delay`, `carrier delay`, `weather delay` and `security delay`.

4.8 Departure delay distribution over the day period

We plot the departure delays calculated for each period of the day using a bar graph.

First, we make a copy of `DEP_TIME_BLK` and regroup the `DEP_TIME_PERIOD` set to MIDNIGHT, MORNING, AFTERNOON and EVENING.

```
flight.df$DEP_TIME_PERIOD <- flight.df$DEP_TIME_BLK
flight.df$DEP_TIME_PERIOD <- factor(flight.df$DEP_TIME_PERIOD)
#Regrouping factor levels
levels(flight.df$DEP_TIME_PERIOD) <-
  c("MIDNIGHT", "MORNING", "MORNING", "MORNING", "MORNING",
```

```
    "MORNING", "MORNING", "AFTERNOON", "AFTERNOON", "AFTERNOON",
    "AFTERNOON", "AFTERNOON", "AFTERNOON", "EVENING","EVENING",
    "EVENING", "EVENING", "EVENING", "EVENING")
```

Second, we filter out the rows where `DEP_DELAY_NEW` is NA.

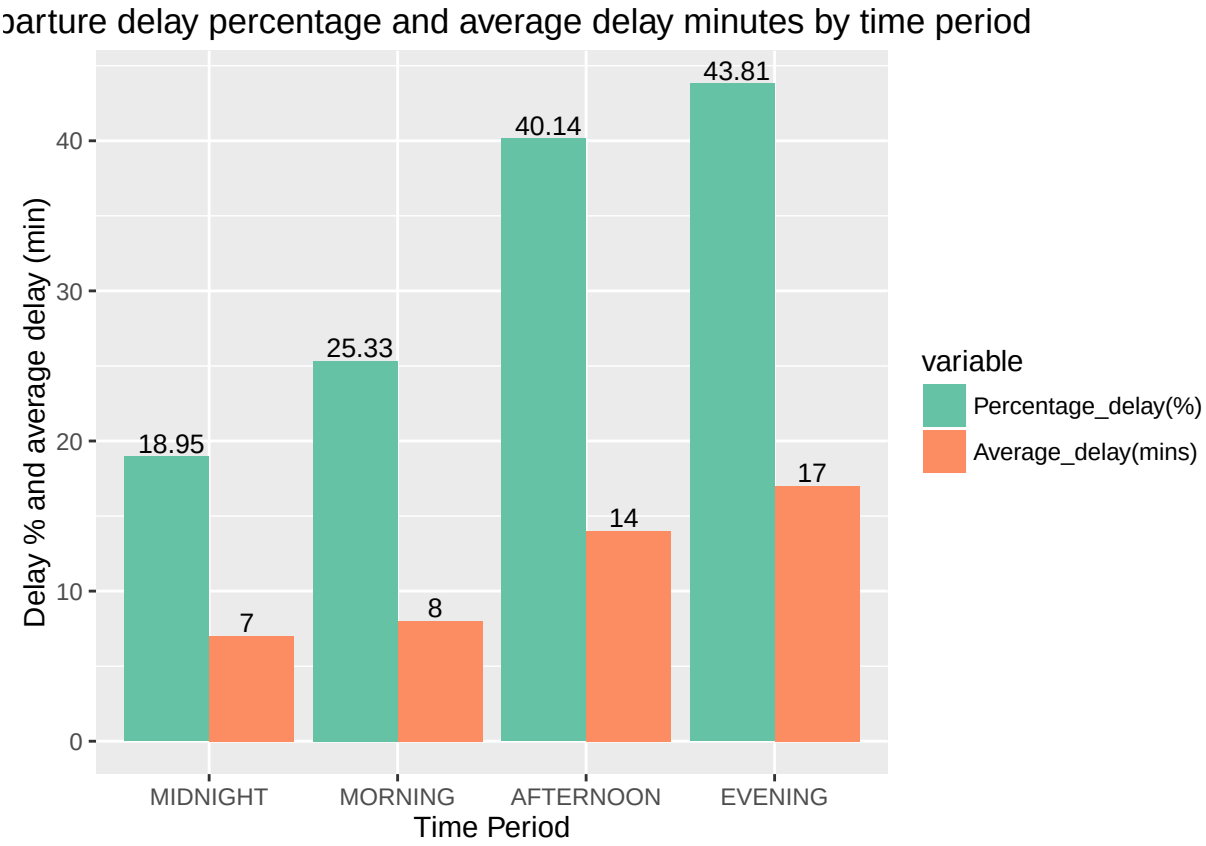
Third, we calculate the delay rate and average delay time in each time period by using `group_by` and `summarise`.

Fourth, we melt the dataframe for multi category bar plot.

```
flight.df %>%
  subset(!is.na(DEP_DELAY_NEW)) %>%
  select(DEP_TIME_PERIOD, DEP_DELAY_NEW) %>%
  group_by(DEP_TIME_PERIOD) %>%
  summarise("Percentage_delay(%)" =
            (length(which(DEP_DELAY_NEW > 0))*100/length(which(DEP_DELAY_NEW >= 0))) %>%
              round(2),
            "Average_delay(mins)" = mean(DEP_DELAY_NEW)%>% round())%>%
  melt(id.vars = c('DEP_TIME_PERIOD')) -> graph7.df
```

Finally, we plot a bar chart by using above data.

```
p9<- graph7.df %>%
  ggplot(aes(x = DEP_TIME_PERIOD, y = value, fill = variable) ) +
  geom_bar(stat="identity", position = 'dodge') +
  ggtitle("Departure delay percentage and average delay minutes by time period") +
  geom_text(aes(label=value), vjust=-0.2, hjust=0.5, color="black",
            position = position_dodge(0.8), size=3.5)+
  xlab("Time Period") +
  ylab("Delay % and average delay (min)") +
  scale_fill_brewer(palette = "Set2")
p9
```



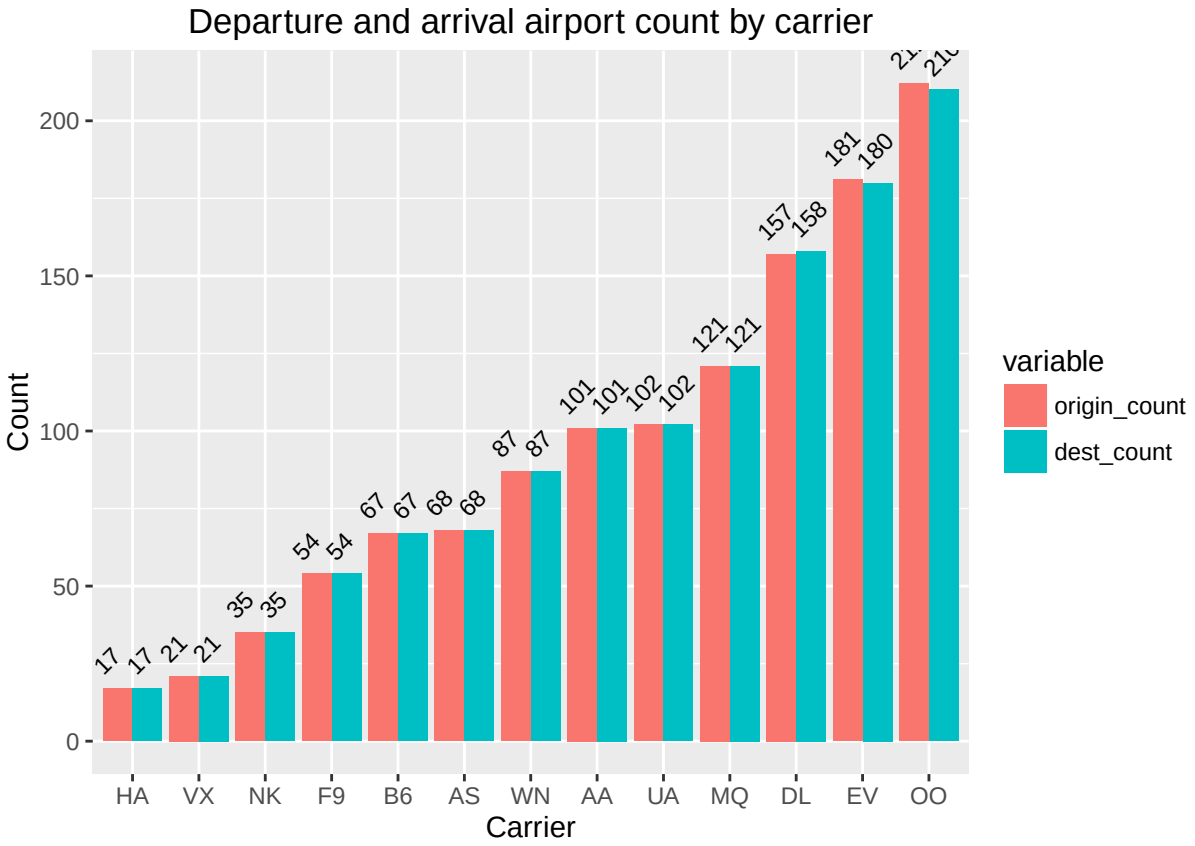

```
# gg9 <- ggplotly(p9)
# plotly_POST(p9, filename = "DepartureDelayDistributionOverTheDayPeriod")
```

We observe that in the evening period, both of departure delay percentage and average delay time reached the highest value of 43.81% and 17 minutes respectively. On the contrary, around midnight both of these values are relatively low; that is 18.95% and 7 minutes respectively.

4.9 Departure and arrival airport count by carrier

```
flight.df %>%
  select(UNIQUE_CARRIER,ORIGIN, DEST) %>%
  group_by(UNIQUE_CARRIER) %>%
  summarise(origin_count = n_distinct(ORIGIN),
            dest_count = n_distinct(DEST)) %>%
  melt(id.vars = c("UNIQUE_CARRIER")) -> graph8.df
```

```
p10<- graph8.df %>%
  ggplot(aes(x = reorder(UNIQUE_CARRIER,value),
             y = value, fill = variable) ) +
  geom_bar(stat="identity", position = 'dodge') +
  ggtitle("Departure and arrival airport count by carrier") +
  geom_text(aes(label=value), vjust=-1, hjust=0, color="black",
            position = position_dodge(1), size=3, angle = 45)+
  xlab("Carrier") +
  ylab("Count")
p10
```




```
# gg10 <- ggplotly(p10)
# plotly_POST(p10, filename = "DepartureAndArrivalAirportCountByCarrier")
```

From above, we learned that OO (SkyWest Airlines) has the maximum number of depature and arrival airports which is more than 210. HA (Hawaiian Airlines) has the least number of depature and arrival airports which is only 17.

5. Text mining on airline reviews – Word Cloud

5.1 AA

```
reviews.df <- read.csv("/Users/dongyueli/Desktop/airline.csv")
AA_reviews <- filter(reviews.df, airline_name == "american-airlines")  #612
AA_Corpus <- Corpus(VectorSource(AA_reviews$content))
#Removing characters
for(j in seq(AA_Corpus))
{
  AA_Corpus[[j]] <- gsub("/", " ", AA_Corpus[[j]])
  AA_Corpus[[j]] <- gsub("@", " ", AA_Corpus[[j]])
  AA_Corpus[[j]] <- gsub("\\|", " ", AA_Corpus[[j]])
}
#Remove punctuations
AA_Corpus <- tm_map(AA_Corpus, removePunctuation)
#Remove numbers
AA_Corpus <- tm_map(AA_Corpus, removeNumbers)
```

```
#Convert the text to lower case
AA_Corpus <- tm_map(AA_Corpus, tolower)
#Remove English Stop Words
AA_Corpus <- tm_map(AA_Corpus, removeWords, stopwords("english"))
AA_Corpus <- tm_map(AA_Corpus, removeWords, stopwords("SMART"))

#Remove particular words
AA_Corpus <- tm_map(AA_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
AA_Corpus <- tm_map(AA_Corpus, stripWhitespace)
#To Finish
AA_Corpus <- tm_map(AA_Corpus, PlainTextDocument)
#Build a document term matrix
AA_dtm <- DocumentTermMatrix(AA_Corpus)
#Frequent Terms
AA_m <- as.matrix(AA_dtm)
AA_v <- sort(colSums(AA_m),decreasing=TRUE)
AA_d <- data.frame(word = names(AA_v),freq=AA_v)
#Plot the 100 most frequently occurring words.
wordcloud(AA_d$word,AA_d$freq, max.words=50, ,scale=c(3,0.5),colors=brewer.pal(6, "Dark2"))
```

5.2 AS

```
AS_reviews <- filter(reviews.df, airline_name == "alaska-airlines")
AS_Corpus <- Corpus(VectorSource(AS_reviews$content))

#Removing characters
```

```
for(j in seq(AS_Corpus))
{
  AS_Corpus[[j]] <- gsub("/", " ", AS_Corpus[[j]])
  AS_Corpus[[j]] <- gsub("@", " ", AS_Corpus[[j]])
  AS_Corpus[[j]] <- gsub("\\|", " ", AS_Corpus[[j]])
}

#Remove punctuations
AS_Corpus <- tm_map(AS_Corpus, removePunctuation)
#Remove numbers
AS_Corpus <- tm_map(AS_Corpus, removeNumbers)
#Convert the text to lower case
AS_Corpus <- tm_map(AS_Corpus, tolower)
#Remove English Stop Words
AS_Corpus <- tm_map(AS_Corpus, removeWords, stopwords("english"))
AS_Corpus <- tm_map(AS_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
AS_Corpus <- tm_map(AS_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
AS_Corpus <- tm_map(AS_Corpus, stripWhitespace)
#To Finish
AS_Corpus <- tm_map(AS_Corpus, PlainTextDocument)
#Build a document term matrix
AS_dtm <- DocumentTermMatrix(AS_Corpus)
#Frequent Terms
AS_m <- as.matrix(AS_dtm)
AS_v <- sort(colSums(AS_m),decreasing=TRUE)
AS_d <- data.frame(word = names(AS_v),freq=AS_v)
#Plot the 100 most frequently occurring words.
wordcloud(AS_d$word,AS_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```

5.3 B6

```
B6_reviews <- filter(reviews.df, airline_name == "jetblue-airways")
B6_Corpus <- Corpus(VectorSource(B6_reviews$content))

#Removing characters
for(j in seq(B6_Corpus))
{
  B6_Corpus[[j]] <- gsub("/", " ", B6_Corpus[[j]])
  B6_Corpus[[j]] <- gsub("@", " ", B6_Corpus[[j]])
  B6_Corpus[[j]] <- gsub("\\|", " ", B6_Corpus[[j]])
}

#Remove punctuations
B6_Corpus <- tm_map(B6_Corpus, removePunctuation)
#Remove numbers
B6_Corpus <- tm_map(B6_Corpus, removeNumbers)
#Convert the text to lower case
B6_Corpus <- tm_map(B6_Corpus, tolower)
#Remove English Stop Words
B6_Corpus <- tm_map(B6_Corpus, removeWords, stopwords("english"))
B6_Corpus <- tm_map(B6_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
B6_Corpus <- tm_map(B6_Corpus, removeWords,c("flight","flights",
                                  "plane", "airlines",
                                   "seat","airline",
                                  "flying"))
#Eliminate extra white spaces
B6_Corpus <- tm_map(B6_Corpus, stripWhitespace)
#To Finish
B6_Corpus <- tm_map(B6_Corpus, PlainTextDocument)
#Build a document term matrix
B6_dtm <- DocumentTermMatrix(B6_Corpus)
#Frequent Terms
B6_m <- as.matrix(B6_dtm)
B6_v <- sort(colSums(B6_m),decreasing=TRUE)
B6_d <- data.frame(word = names(B6_v),freq=B6_v)
#Plot the 100 most frequently occurring words.
wordcloud(B6_d$word,B6_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```

5.4 DL

```
DL_reviews <- filter(reviews.df, airline_name == "delta-air-lines")
DL_Corpus <- Corpus(VectorSource(DL_reviews$content))

#Removing characters
for(j in seq(DL_Corpus))
{
  DL_Corpus[[j]] <- gsub("/", " ", DL_Corpus[[j]])
  DL_Corpus[[j]] <- gsub("@", " ", DL_Corpus[[j]])
  DL_Corpus[[j]] <- gsub("\\|", " ", DL_Corpus[[j]])
}

#Remove punctuations
DL_Corpus <- tm_map(DL_Corpus, removePunctuation)
#Remove numbers
DL_Corpus <- tm_map(DL_Corpus, removeNumbers)
#Convert the text to lower case
DL_Corpus <- tm_map(DL_Corpus, tolower)
#Remove English Stop Words
DL_Corpus <- tm_map(DL_Corpus, removeWords, stopwords("english"))
DL_Corpus <- tm_map(DL_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
DL_Corpus <- tm_map(DL_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
```

```
DL_Corpus <- tm_map(DL_Corpus, stripWhitespace)
#To Finish
DL_Corpus <- tm_map(DL_Corpus, PlainTextDocument)
#Build a document term matrix
DL_dtm <- DocumentTermMatrix(DL_Corpus)
#Frequent Terms
DL_m <- as.matrix(DL_dtm)
DL_v <- sort(colSums(DL_m),decreasing=TRUE)
DL_d <- data.frame(word = names(DL_v),freq=DL_v)
#Plot the 100 most frequently occurring words.
wordcloud(DL_d$word,DL_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```

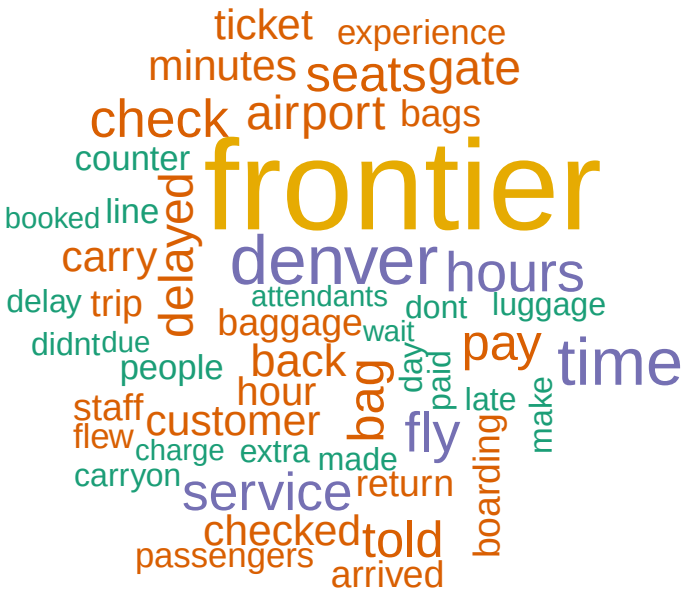
5.5 F9

```
F9_reviews <- filter(reviews.df, airline_name == "frontier-airlines")
F9_Corpus <- Corpus(VectorSource(F9_reviews$content))

#Removing characters
for(j in seq(F9_Corpus))
{
  F9_Corpus[[j]] <- gsub("/", " ", F9_Corpus[[j]])
  F9_Corpus[[j]] <- gsub("@", " ", F9_Corpus[[j]])
  F9_Corpus[[j]] <- gsub("\\|", " ", F9_Corpus[[j]])
}

#Remove punctuations
F9_Corpus <- tm_map(F9_Corpus, removePunctuation)
#Remove numbers
F9_Corpus <- tm_map(F9_Corpus, removeNumbers)
#Convert the text to lower case
F9_Corpus <- tm_map(F9_Corpus, tolower)
#Remove English Stop Words
F9_Corpus <- tm_map(F9_Corpus, removeWords, stopwords("english"))
F9_Corpus <- tm_map(F9_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
```

```
F9_Corpus <- tm_map(F9_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
F9_Corpus <- tm_map(F9_Corpus, stripWhitespace)
#To Finish
F9_Corpus <- tm_map(F9_Corpus, PlainTextDocument)
#Build a document term matrix
F9_dtm <- DocumentTermMatrix(F9_Corpus)
#Frequent Terms
F9_m <- as.matrix(F9_dtm)
F9_v <- sort(colSums(F9_m),decreasing=TRUE)
F9_d <- data.frame(word = names(F9_v),freq=F9_v)
#Plot the 100 most frequently occurring words.
wordcloud(F9_d$word,F9_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```



5.6 HA

```
HA_reviews <- filter(reviews.df, airline_name == "hawaiian-airlines")
HA_Corpus <- Corpus(VectorSource(HA_reviews$content))

#Removing characters
for(j in seq(HA_Corpus))
{
  HA_Corpus[[j]] <- gsub("/", " ", HA_Corpus[[j]])
  HA_Corpus[[j]] <- gsub("@", " ", HA_Corpus[[j]])
  HA_Corpus[[j]] <- gsub("\\|", " ", HA_Corpus[[j]])
}

#Remove punctuations
HA_Corpus <- tm_map(HA_Corpus, removePunctuation)
#Remove numbers
```

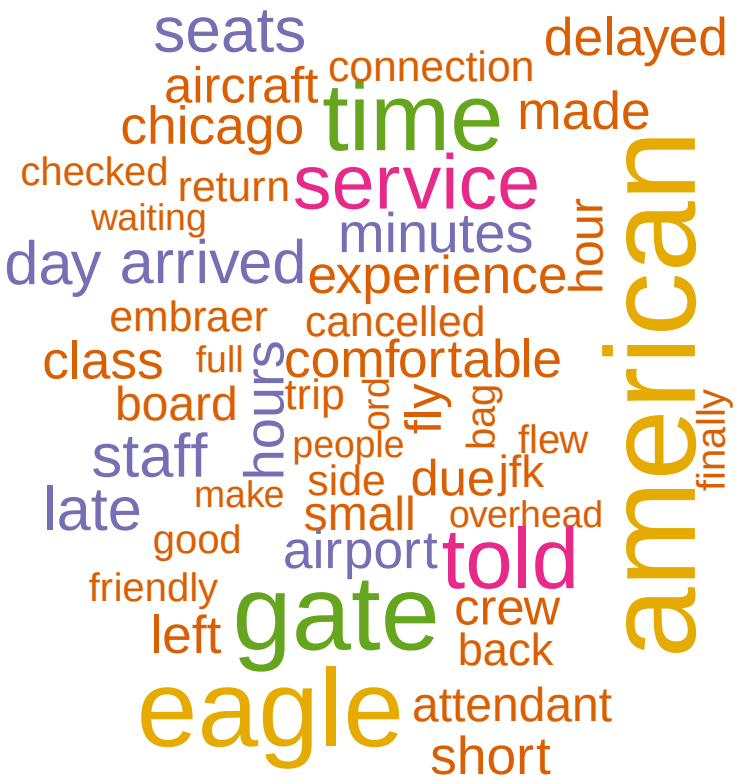
```
HA_Corpus <- tm_map(HA_Corpus, removeNumbers)
#Convert the text to lower case
HA_Corpus <- tm_map(HA_Corpus, tolower)
#Remove English Stop Words
HA_Corpus <- tm_map(HA_Corpus, removeWords, stopwords("english"))
HA_Corpus <- tm_map(HA_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
HA_Corpus <- tm_map(HA_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
HA_Corpus <- tm_map(HA_Corpus, stripWhitespace)
#To Finish
HA_Corpus <- tm_map(HA_Corpus, PlainTextDocument)
#Build a document term matrix
HA_dtm <- DocumentTermMatrix(HA_Corpus)
#Frequent Terms
HA_m <- as.matrix(HA_dtm)
HA_v <- sort(colSums(HA_m),decreasing=TRUE)
HA_d <- data.frame(word = names(HA_v),freq=HA_v)
#Plot the 100 most frequently occurring words.
wordcloud(HA_d$word,HA_d$freq, max.words=50, ,scale=c(3,0.5),colors=brewer.pal(6, "Dark2"))
```

5.7 MQ

```
MQ_reviews <- filter(reviews.df, airline_name == "american-eagle")
MQ_Corpus <- Corpus(VectorSource(MQ_reviews$content))
```

```
#Removing cMQracters
for(j in seq(MQ_Corpus))
{
  MQ_Corpus[[j]] <- gsub("/", " ", MQ_Corpus[[j]])
  MQ_Corpus[[j]] <- gsub("@", " ", MQ_Corpus[[j]])
  MQ_Corpus[[j]] <- gsub("\\|", " ", MQ_Corpus[[j]])
}

#Remove punctuations
MQ_Corpus <- tm_map(MQ_Corpus, removePunctuation)
#Remove numbers
MQ_Corpus <- tm_map(MQ_Corpus, removeNumbers)
#Convert the text to lower case
MQ_Corpus <- tm_map(MQ_Corpus, tolower)
#Remove English Stop Words
MQ_Corpus <- tm_map(MQ_Corpus, removeWords, stopwords("english"))
MQ_Corpus <- tm_map(MQ_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
MQ_Corpus <- tm_map(MQ_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
MQ_Corpus <- tm_map(MQ_Corpus, stripWhitespace)
#To Finish
MQ_Corpus <- tm_map(MQ_Corpus, PlainTextDocument)
#Build a document term matrix
MQ_dtm <- DocumentTermMatrix(MQ_Corpus)
#Frequent Terms
MQ_m <- as.matrix(MQ_dtm)
MQ_v <- sort(colSums(MQ_m),decreasing=TRUE)
MQ_d <- data.frame(word = names(MQ_v),freq=MQ_v)
#Plot the 100 most frequently occurring words.
wordcloud(MQ_d$word,MQ_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```



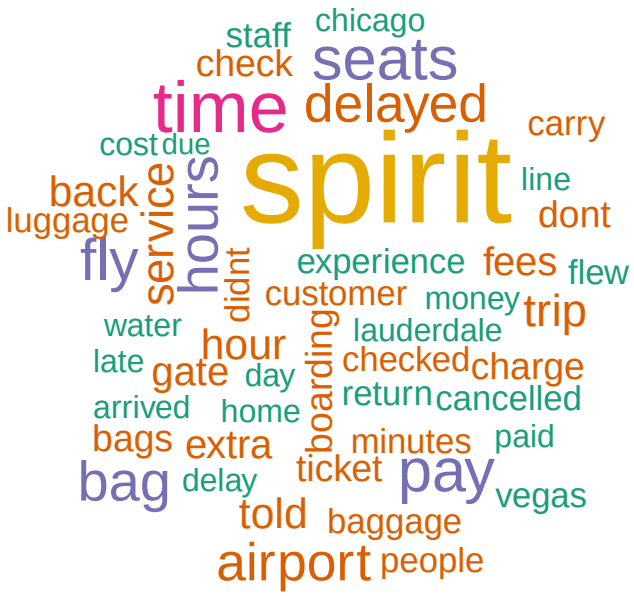
5.8 NK

```
NK_reviews <- filter(reviews.df, airline_name == "spirit-airlines")
NK_Corpus <- Corpus(VectorSource(NK_reviews$content))

#Removing cNKracters
for(j in seq(NK_Corpus))
{
  NK_Corpus[[j]] <- gsub("/", " ", NK_Corpus[[j]])
  NK_Corpus[[j]] <- gsub("@", " ", NK_Corpus[[j]])
  NK_Corpus[[j]] <- gsub("\\|", " ", NK_Corpus[[j]])
}

#Remove punctuations
NK_Corpus <- tm_map(NK_Corpus, removePunctuation)
#Remove numbers
NK_Corpus <- tm_map(NK_Corpus, removeNumbers)
#Convert the text to lower case
NK_Corpus <- tm_map(NK_Corpus, tolower)
#Remove English Stop Words
NK_Corpus <- tm_map(NK_Corpus, removeWords, stopwords("english"))
NK_Corpus <- tm_map(NK_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
NK_Corpus <- tm_map(NK_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
```

```
#Eliminate extra white spaces
NK_Corpus <- tm_map(NK_Corpus, stripWhitespace)
#To Finish
NK_Corpus <- tm_map(NK_Corpus, PlainTextDocument)
#Build a document term matrix
NK_dtm <- DocumentTermMatrix(NK_Corpus)
#Frequent Terms
NK_m <- as.matrix(NK_dtm)
NK_v <- sort(colSums(NK_m),decreasing=TRUE)
NK_d <- data.frame(word = names(NK_v),freq=NK_v)
#Plot the 100 most frequently occurring words.
wordcloud(NK_d$word,NK_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```



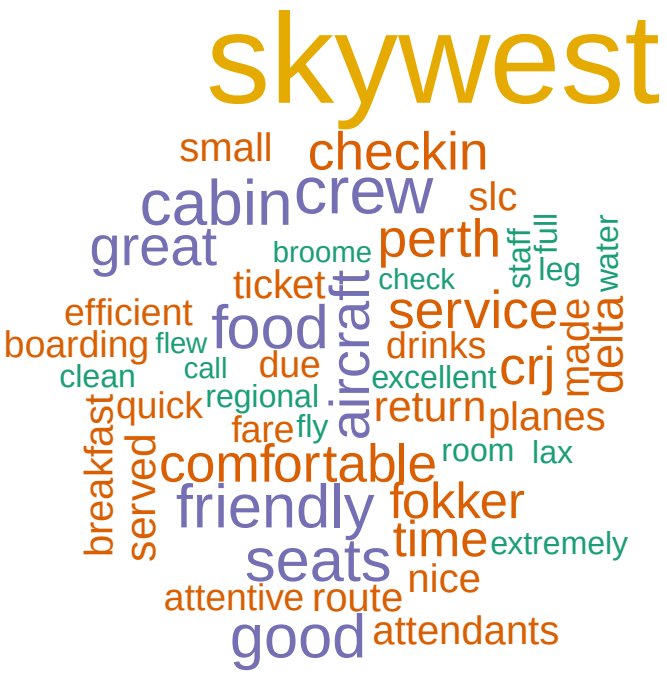
5.9 OO

```
OO_reviews <- filter(reviews.df, airline_name == "skywest-airlines")
OO_Corpus <- Corpus(VectorSource(OO_reviews$content))

#Removing cOOracters
for(j in seq(OO_Corpus))
{
  OO_Corpus[[j]] <- gsub("/", " ", OO_Corpus[[j]])
  OO_Corpus[[j]] <- gsub("@", " ", OO_Corpus[[j]])
  OO_Corpus[[j]] <- gsub("\\|", " ", OO_Corpus[[j]])
}

#Remove punctuations
OO_Corpus <- tm_map(OO_Corpus, removePunctuation)
#Remove numbers
OO_Corpus <- tm_map(OO_Corpus, removeNumbers)
#Convert the text to lower case
OO_Corpus <- tm_map(OO_Corpus, tolower)
#Remove English Stop Words
```

```
OO_Corpus <- tm_map(OO_Corpus, removeWords, stopwords("english"))
OO_Corpus <- tm_map(OO_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
OO_Corpus <- tm_map(OO_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
OO_Corpus <- tm_map(OO_Corpus, stripWhitespace)
#To Finish
OO_Corpus <- tm_map(OO_Corpus, PlainTextDocument)
#Build a document term matrix
OO_dtm <- DocumentTermMatrix(OO_Corpus)
#Frequent Terms
OO_m <- as.matrix(OO_dtm)
OO_v <- sort(colSums(OO_m),decreasing=TRUE)
OO_d <- data.frame(word = names(OO_v),freq=OO_v)
#Plot the 100 most frequently occurring words.
wordcloud(OO_d$word,OO_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```



5.10 UA

```
UA_reviews <- filter(reviews.df, airline_name == "united-airlines")
UA_Corpus <- Corpus(VectorSource(UA_reviews$content))

#Removing cUAracters
for(j in seq(UA_Corpus))
{
  UA_Corpus[[j]] <- gsub("/", " ", UA_Corpus[[j]])
  UA_Corpus[[j]] <- gsub("@", " ", UA_Corpus[[j]])
```

```
  UA_Corpus[[j]] <- gsub("\\|", " ", UA_Corpus[[j]])
}

#Remove punctuations
UA_Corpus <- tm_map(UA_Corpus, removePunctuation)
#Remove numbers
UA_Corpus <- tm_map(UA_Corpus, removeNumbers)
#Convert the text to lower case
UA_Corpus <- tm_map(UA_Corpus, tolower)
#Remove English Stop Words
UA_Corpus <- tm_map(UA_Corpus, removeWords, stopwords("english"))
UA_Corpus <- tm_map(UA_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
UA_Corpus <- tm_map(UA_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
UA_Corpus <- tm_map(UA_Corpus, stripWhitespace)
#To Finish
UA_Corpus <- tm_map(UA_Corpus, PlainTextDocument)
#Build a document term matrix
UA_dtm <- DocumentTermMatrix(UA_Corpus)
#Frequent Terms
UA_m <- as.matrix(UA_dtm)
UA_v <- sort(colSums(UA_m),decreasing=TRUE)
UA_d <- data.frame(word = names(UA_v),freq=UA_v)
#Plot the 100 most frequently occurring words.
wordcloud(UA_d$word,UA_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```

5.11 VX

```
VX_reviews <- filter(reviews.df, airline_name == "virgin-america")
VX_Corpus <- Corpus(VectorSource(VX_reviews$content))

#Removing cVXracters
for(j in seq(VX_Corpus))
{
  VX_Corpus[[j]] <- gsub("/", " ", VX_Corpus[[j]])
  VX_Corpus[[j]] <- gsub("@", " ", VX_Corpus[[j]])
  VX_Corpus[[j]] <- gsub("\\|", " ", VX_Corpus[[j]])
}

#Remove punctVXtions
VX_Corpus <- tm_map(VX_Corpus, removePunctuation)
#Remove numbers
VX_Corpus <- tm_map(VX_Corpus, removeNumbers)
#Convert the text to lower case
VX_Corpus <- tm_map(VX_Corpus, tolower)
#Remove English Stop Words
VX_Corpus <- tm_map(VX_Corpus, removeWords, stopwords("english"))
VX_Corpus <- tm_map(VX_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
VX_Corpus <- tm_map(VX_Corpus, removeWords,c("flight","flights",
                                  "plane", "airlines",
                                   "seat","airline",
                                  "flying"))
#Eliminate extra white spaces
VX_Corpus <- tm_map(VX_Corpus, stripWhitespace)
#To Finish
VX_Corpus <- tm_map(VX_Corpus, PlainTextDocument)
#Build a document term matrix
VX_dtm <- DocumentTermMatrix(VX_Corpus)
#Frequent Terms
VX_m <- as.matrix(VX_dtm)
VX_v <- sort(colSums(VX_m),decreasing=TRUE)
VX_d <- data.frame(word = names(VX_v),freq=VX_v)
#Plot the 100 most frequently occurring words.
wordcloud(VX_d$word,VX_d$freq, max.words=50, ,scale=c(3,0.5),colors=brewer.pal(6, "Dark2"))
```

5.12 WN

```
WN_reviews <- filter(reviews.df, airline_name == "southwest-airlines")
WN_Corpus <- Corpus(VectorSource(WN_reviews$content))

#Removing cWNracters
for(j in seq(WN_Corpus))
{
  WN_Corpus[[j]] <- gsub("/", " ", WN_Corpus[[j]])
  WN_Corpus[[j]] <- gsub("@", " ", WN_Corpus[[j]])
  WN_Corpus[[j]] <- gsub("\\|", " ", WN_Corpus[[j]])
}

#Remove punctuations
WN_Corpus <- tm_map(WN_Corpus, removePunctuation)
#Remove numbers
WN_Corpus <- tm_map(WN_Corpus, removeNumbers)
#Convert the text to lower case
WN_Corpus <- tm_map(WN_Corpus, tolower)
#Remove English Stop Words
WN_Corpus <- tm_map(WN_Corpus, removeWords, stopwords("english"))
WN_Corpus <- tm_map(WN_Corpus, removeWords, stopwords("SMART"))
#Remove particular words
WN_Corpus <- tm_map(WN_Corpus, removeWords,c("flight","flights",
                                    "plane", "airlines",
                                     "seat","airline",
                                    "flying"))
#Eliminate extra white spaces
WN_Corpus <- tm_map(WN_Corpus, stripWhitespace)
#To Finish
```

```
WN_Corpus <- tm_map(WN_Corpus, PlainTextDocument)
#Build a document term matrix
WN_dtm <- DocumentTermMatrix(WN_Corpus)
#Frequent Terms
WN_m <- as.matrix(WN_dtm)
WN_v <- sort(colSums(WN_m),decreasing=TRUE)
WN_d <- data.frame(word = names(WN_v),freq=WN_v)
#Plot the 100 most frequently occurring words.
wordcloud(WN_d$word,WN_d$freq, max.words=50, ,scale=c(4,0.5),colors=brewer.pal(6, "Dark2"))
```