CptS 575_Assignment 04

Sajjad Uddin Mahmud | WSU ID: 011789534

2022-10-03

Solution of Problem 01 :

Solution of 1(a)

```
# Loading the packages
library(nycflights13)
library(dplyr)

# Reading the data set

weather %>%
  select(origin,year,month,day,hour,humid) %>%
  filter(year==2013, month==11, day==1, hour>=12 & hour<=18) ->
  Weather_Data

flights %>%
  select(dest, dep_time, tailnum, year, month, day, hour, origin) %>%
  filter(dest == "TPA", year == 2013, month == 11, day == 1, dep_time >= 1200
         & dep_time<=1800) %>%
  mutate(hour = round(dep_time/100)) %>%
  left_join(Weather_Data, by = c("origin" = "origin", "year" = "year",
                                 "month" = "month", "day" = "day", "hour"="hour")) ->
  Flight_Data_Tampa

Flight_Data_Tampa
```

```
## # A tibble: 6 x 9
##   dest  dep_time tailnum  year month   day  hour origin humid
##   <chr>    <int> <chr>   <int> <int> <int> <dbl> <chr>  <dbl>
## 1 TPA       1440 N580JB   2013    11     1    14 JFK     63.1
## 2 TPA       1451 N337NB   2013    11     1    15 LGA     50.6
## 3 TPA       1457 N567UA   2013    11     1    15 EWR     52.8
## 4 TPA       1508 N515MQ   2013    11     1    15 JFK     65.1
## 5 TPA       1707 N779JB   2013    11     1    17 EWR     56.5
## 6 TPA       1737 N561JB   2013    11     1    17 LGA     58.6
```

There are **6** flights happened during the given time frame.

Solution of 1(b)

```
Anti_Join_1 <- anti_join(flights, airports, by = c("origin" = "faa"))

Anti_Join_1
```

```
## # A tibble: 0 x 19
## # ... with 19 variables: year <int>, month <int>, day <int>, dep_time <int>,
## #   sched_dep_time <int>, dep_delay <dbl>, arr_time <int>,
## #   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>
```

```
Anti_Join_2 <- anti_join(airports, flights, by = c("faa" = "origin"))

Anti_Join_2
```

```
## # A tibble: 1,455 x 8
##    faa   name                            lat    lon   alt    tz dst   tzone
##    <chr> <chr>                         <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
##  1 04G   Lansdowne Airport              41.1  -80.6  1044    -5 A     America/~
##  2 06A   Moton Field Municipal Airport  32.5  -85.7   264    -6 A     America/~
##  3 06C   Schaumburg Regional            42.0  -88.1   801    -6 A     America/~
##  4 06N   Randall Airport                41.4  -74.4   523    -5 A     America/~
##  5 09J   Jekyll Island Airport          31.1  -81.4    11    -5 A     America/~
##  6 0A9   Elizabethton Municipal Airport 36.4  -82.2  1593    -5 A     America/~
##  7 0G6   Williams County Airport        41.5  -84.5   730    -5 A     America/~
##  8 0G7   Finger Lakes Regional Airport  42.9  -76.8   492    -5 A     America/~
##  9 0P2   Shoestring Aviation Airfield   39.8  -76.6  1000    -5 U     America/~
## 10 0S9   Jefferson County Intl          48.1 -123.    108    -8 A     America/~
## # ... with 1,445 more rows
```

Difference between Anti_Join_1 and Anti_Join_2

We know that anti_join() return all rows from X dataset without a match in Y dataset.

In Anti_Join_1, we are looking into rows of flights dataset that are not in the airports dataset for "origin=faa". As in the flights dataset, there are only EWR, JFK and LGA airports, hence it returns zero rows.

In Anti_Join_2, we are looking into rows of airports dataset that are not in the flights dataset for "faa=origin" condition. As a result, we can see 1455 rows. Because the total rows were 1458 and it excludes 3 (EWR, JFK and LGA) from that.

Difference between semi_join and anti_join

With just the columns from X dataset kept, semi join(X,Y) returns all rows from X where there are matching values in Y. On the contrary, anti join(X,Y) only keeps the columns from X and returns all rows from X where there are no matching values in Y.

Solution of 1(c)

```
flights %>%
  select(origin, dest) ->
  Flight_Routes

airports %>%
  select(faa,lat,lon) ->
  Airport_Locations

Flights_Data <- inner_join(Flight_Routes,Airport_Locations, by = c("dest" = "faa"))

Flights_Data
```

```
## # A tibble: 329,174 x 4
##    origin dest    lat   lon
##    <chr>  <chr> <dbl> <dbl>
##  1 EWR    IAH    30.0 -95.3
##  2 LGA    IAH    30.0 -95.3
##  3 JFK    MIA    25.8 -80.3
##  4 LGA    ATL    33.6 -84.4
##  5 EWR    ORD    42.0 -87.9
##  6 EWR    FLL    26.1 -80.2
##  7 LGA    IAD    38.9 -77.5
##  8 JFK    MCO    28.4 -81.3
##  9 LGA    ORD    42.0 -87.9
## 10 JFK    PBI    26.7 -80.1
## # ... with 329,164 more rows
```

There are **329,174** flights.

Solution of 1(d)

```
flights %>%
  group_by(carrier, dest) %>%
  count(sort = TRUE) ->
  Flights_Summary

Flights_Summary
```

```
## # A tibble: 314 x 3
## # Groups:   carrier, dest [314]
##    carrier dest      n
##    <chr>   <chr> <int>
##  1 DL      ATL   10571
##  2 US      CLT    8632
##  3 AA      DFW    7257
##  4 AA      MIA    7234
##  5 UA      ORD    6984
##  6 UA      IAH    6924
```

```
##  7 UA      SFO    6819
##  8 B6      FLL    6563
##  9 B6      MCO    6472
## 10 AA      ORD    6059
## # ... with 304 more rows
```

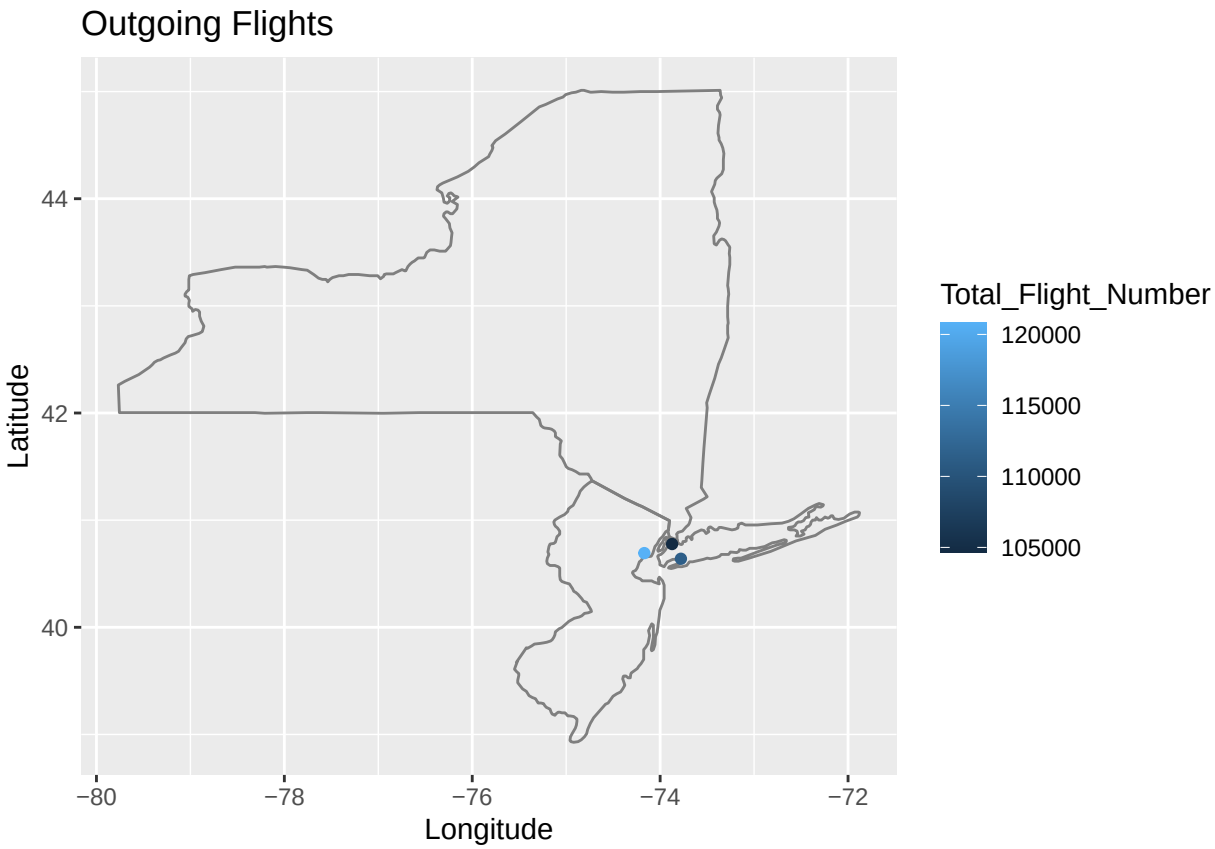
There are **314** unique combinations of carrier/dest present in the dataset.

Solution of 1(e)

```
# Loading the packages
library(maps)
library(usmap)
library(ggplot2)

# Getting the data to map
flights %>%
  select(origin) %>%
  left_join(airports, by = c("origin" = "faa")) %>%
  group_by(origin) %>%
  select(origin,lat,lon) %>%
  summarise(Total_Flight_Number = n()) ->
  Flight_Origin_Map_Data

Flight_Origin_Map_Data
```

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

```
Flight_Origin_Map_Data %>%
  left_join(airports, c("origin" = "faa")) ->
  Flight_Origin_Map_Data

Flight_Origin_Map_Data %>%
  ggplot(aes(lon,lat, label = origin))+
  borders("state", xlim = c(-74.5,-73.5), ylim = c(40.8,41)) +
  geom_point(aes(colour = Total_Flight_Number)) +
  labs(x="Longitude", y="Latitude", title="Outgoing Flights")
```


Outgoing Flights
Latitude
44
42
40
−80
−78
−76
−74
−72
Longitude
Total_Flight_Number
120000
115000
110000
105000

Solution of Problem 02

For this problem, I have chosen US President Election Year 2016 and 2020 for the visualization.

```
# Loading the packages
library(usmap)
library(ggplot2)
library(RColorBrewer)


# Reading the data set
US_President <- read.csv("us-presidents.csv")

# Creating a visualization of the total number of voter in US President Election

# Year 2016
US_President %>%
  group_by(year) %>%
  filter(year=="2016") ->
  US_Map_Data

Plot_US_Election1 <- plot_usmap(data=US_Map_Data,values="totalvotes",
                                labels = TRUE, label_color="white")+
  scale_fill_continuous(name="Total No. of Voter",low="slateblue4",
                        high="skyblue",label=scales::comma)+
  theme(legend.position = "right") + labs(caption="Figure 01: US Election 2016")

Plot_US_Election1
```

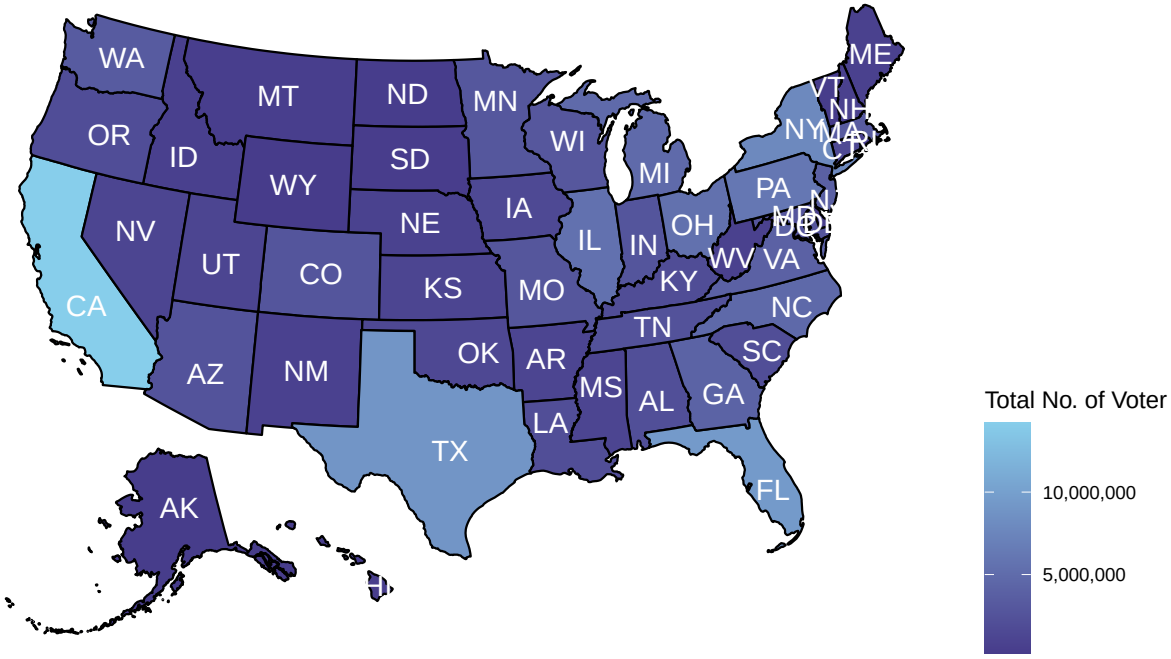


Figure 01: US Election 2016

```
# Year 2020
US_President %>%
  group_by(year) %>%
  filter(year=="2020") ->
  US_Map_Data

Plot_US_Election2 <- plot_usmap(data=US_Map_Data,values="totalvotes",
                                labels = TRUE, label_color="white")+
  scale_fill_continuous(name="Total No. of Voter",low="mediumpurple4",
                        high="mediumorchid1",label=scales::comma)+
  theme(legend.position = "right") + labs(caption="Figure 02: US Election 2020")

Plot_US_Election2
```

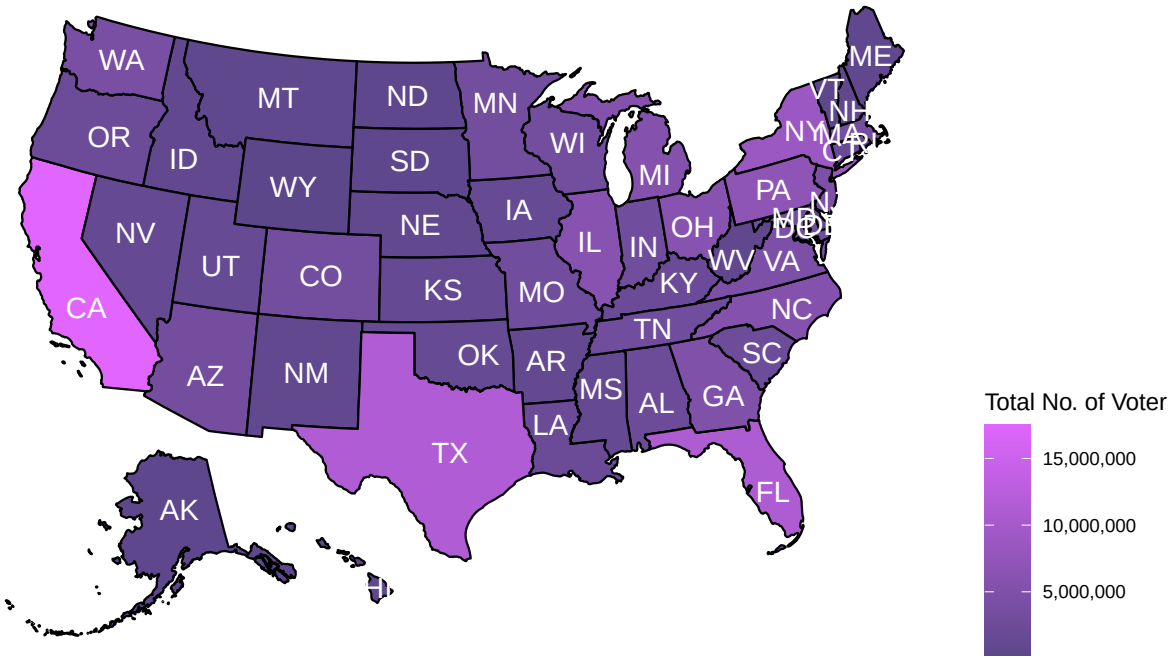


Figure 02: US Election 2020

Solution of Problem 03: Word Cloud

For this problem, I have chosen my "Statement of Purpose" as the input document.

```
# Initialization
library(wordcloud)
library(RColorBrewer)
library(tm)
library(plotly)

# Reading the word document
Word_File_Text <- readLines("D:\\Homeworks\\Data_Science\\Assignment 04\\Statement_of_purpose_Sajjad.tx

Word_File_Text <- Corpus(VectorSource(Word_File_Text))

# Tidying the document
Word_File_Text %>%
  tm_map(content_transformer(tolower)) %>% # Converting all words in lowercase
  tm_map(removeNumbers) %>% # Removing numbers
  tm_map(removeWords, stopwords("english")) %>% # Removing stop words
  tm_map(removeWords, c("also", "become", "sajjad", "mahmud")) %>% # Removing some specific words
  tm_map(removePunctuation) %>% # Removing punctuation
  tm_map(stripWhitespace) -> # Collapsing multiple white space characters to a single blank
  Word_File_Text

# Creating Title
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))
par(mar=rep(0, 4))
plot.new()
text(x=0.5, y=0.5,
     "Sajjad's Statement of Purpose for WSU Application, written in August 2021")

# Generating the wordcloud
wordcloud(Word_File_Text, min.freq = 1, max.words = 200, random.order=FALSE,
          colors=brewer.pal(12,"Set1"))
```

Sajjad's Statement of Purpose for WSU Application, written in August 2021