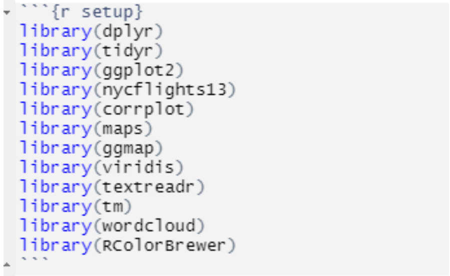
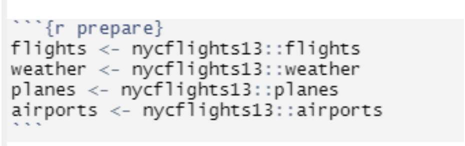
CPTS 575 Data Science
Assignment 4
Jinyang Ruan
011696096

Libraries prepare:



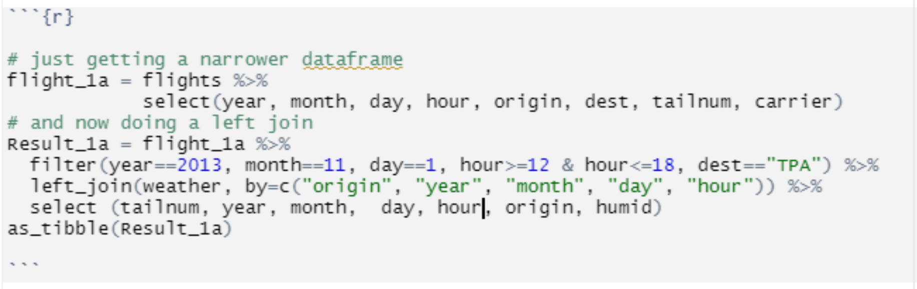
~~~
```{r setup}
library(dplyr)
library(tidyr)
library(ggplot2)
library(nycflights13)
library(corrplot)
library(maps)
library(ggmap)
library(viridis)
library(textreadr)
library(tm)
library(wordcloud)
library(RColorBrewer)
```
~~~

Data prepare:

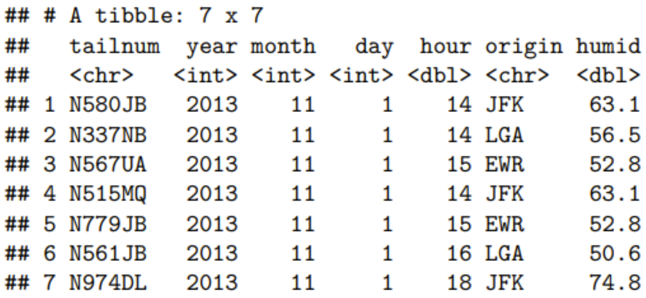


~~~
```{r prepare}
flights <- nycflights13::flights
weather <- nycflights13::weather
planes <- nycflights13::planes
airports <- nycflights13::airports
```
~~~

Problem 1.
1.a Filter the dataset (using a left join) to display the tail number, year, month, day, hour, origin, and humidity for all flights heading to Tampa International Airport (TPA) after 12pm on November 1, 2013.



~~~
```{r}

# just getting a narrower dataframe
flight_1a = flights %>%
            select(year, month, day, hour, origin, dest, tailnum, carrier)
# and now doing a left join
Result_1a = flight_1a %>%
  filter(year==2013, month==11, day==1, hour>=12 & hour<=18, dest=="TPA") %>%
  left_join(weather, by=c("origin", "year", "month", "day", "hour")) %>%
  select (tailnum, year, month,  day, hour, origin, humid)
as_tibble(Result_1a)

```
~~~



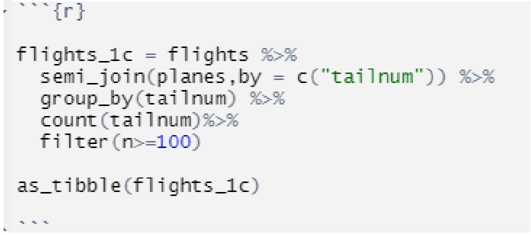
```
## # A tibble: 7 x 7
##   tailnum  year month   day  hour origin humid
##   <chr>   <int> <int> <int> <dbl> <chr>  <dbl>
## 1 N580JB   2013    11     1    14 JFK     63.1
## 2 N337NB   2013    11     1    14 LGA     56.5
## 3 N567UA   2013    11     1    15 EWR     52.8
## 4 N515MQ   2013    11     1    14 JFK     63.1
## 5 N779JB   2013    11     1    15 EWR     52.8
## 6 N561JB   2013    11     1    16 LGA     50.6
## 7 N974DL   2013    11     1    18 JFK     74.8
```

1.b What is the difference between the following two joins?

- anti_join(flights, airports, by = c("dest" = "faa")): this operation will drop from table flights all observations that have a match with the condition ("dest" = "faa") in table airports. the result is a subset of table flights.
- anti_join(airports, flights, by = c("faa" = "dest")): this operation will drop from table airports all observations that have a match with the condition ("dest" = "faa") in table flights the result is a subset of table airports.

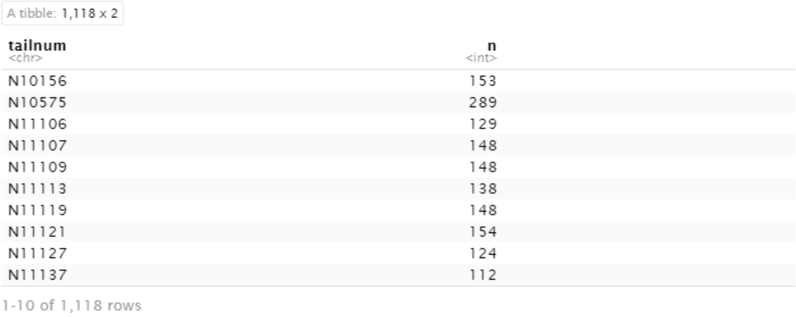
1.c Filter the table flights to only show flights with planes that have flown at least 100 flights. Hint: tailnum is used to identify planes.

I am not sure whether the "year" in the planes table represents the same thing as the "year" in the table flights. If not, semi-join the planes table with flights by "tailnum".



~~~
```{r}

flights_1c = flights %>%
  semi_join(planes,by = c("tailnum")) %>%
  group_by(tailnum) %>%
  count(tailnum)%>%
  filter(n>=100)

as_tibble(flights_1c)

```
~~~

Totally there are 1118 flights with planes that have flown at least 100 flights.

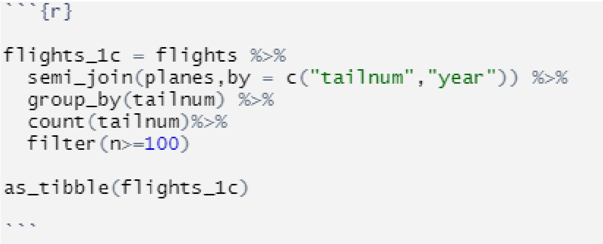


A tibble: 1,118 x 2

tailnum <chr>	n <int>
N10156	153
N10575	289
N11106	129
N11107	148
N11109	148
N11113	138
N11119	148
N11121	154
N11127	124
N11137	112

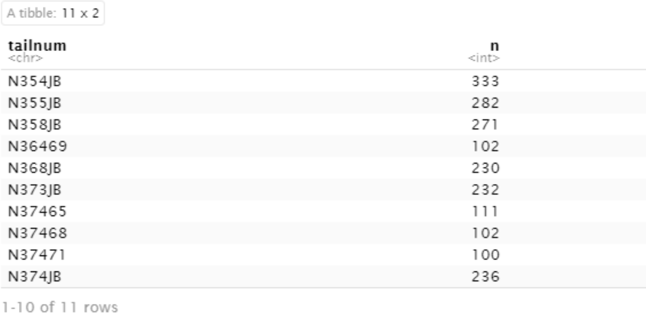
1-10 of 1,118 rows

If we semi-join the planes table with flights by "tailnum" and "year":



~~~
```{r}

flights_1c = flights %>%
  semi_join(planes,by = c("tailnum","year")) %>%
  group_by(tailnum) %>%
  count(tailnum)%>%
  filter(n>=100)

as_tibble(flights_1c)

```
~~~

Totally there are 11 flights with planes that have flown at least 100 flights.

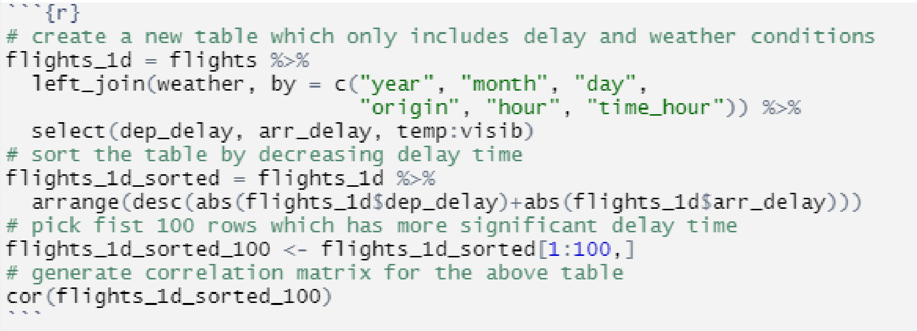


A tibble: 11 x 2

tailnum <chr>	n <int>
N354JB	333
N355JB	282
N358JB	271
N36469	102
N368JB	230
N373JB	232
N37465	111
N37468	102
N37471	100
N374JB	236

1-10 of 11 rows

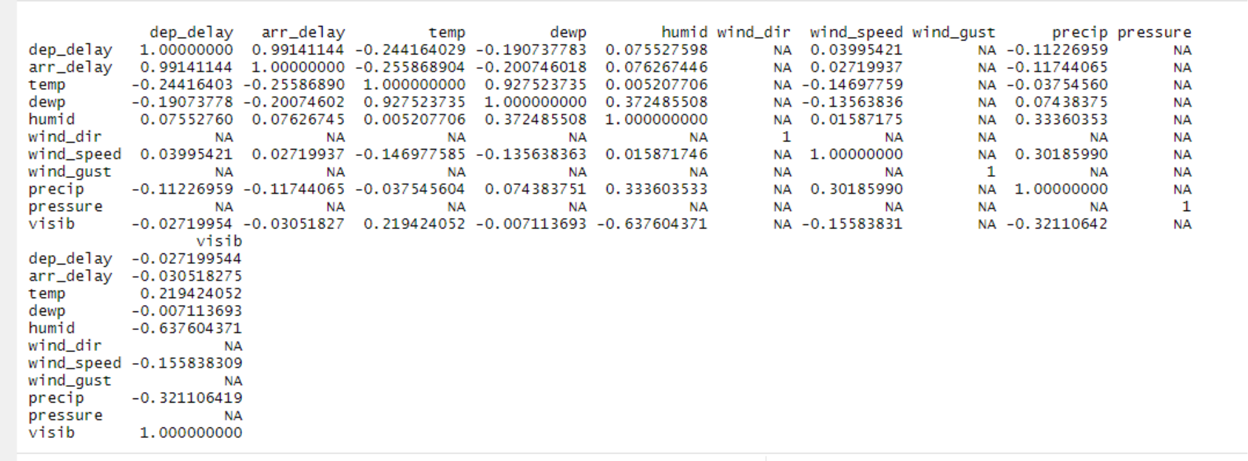
1.d What weather conditions make it more likely to see a delay? Briefly discuss any relations/patterns you found.



~~~
```{r}
# create a new table which only includes delay and weather conditions
flights_1d = flights %>%
  left_join(weather, by = c("year", "month", "day",
                            "origin", "hour", "time_hour")) %>%
  select(dep_delay, arr_delay, temp:visib)
# sort the table by decreasing delay time
flights_1d_sorted = flights_1d %>%
  arrange(desc(abs(flights_1d$dep_delay)+abs(flights_1d$arr_delay)))
# pick fist 100 rows which has more significant delay time
flights_1d_sorted_100 <- flights_1d_sorted[1:100,]
# generate correlation matrix for the above table
cor(flights_1d_sorted_100)
```
~~~

Generally, I left-join the weather table with flights, compute the total delay time (|dep_delay|+|arr_delay|), sort the table by decreasing delay time, pick first 100 rows which has more significant delay time and analyze them.

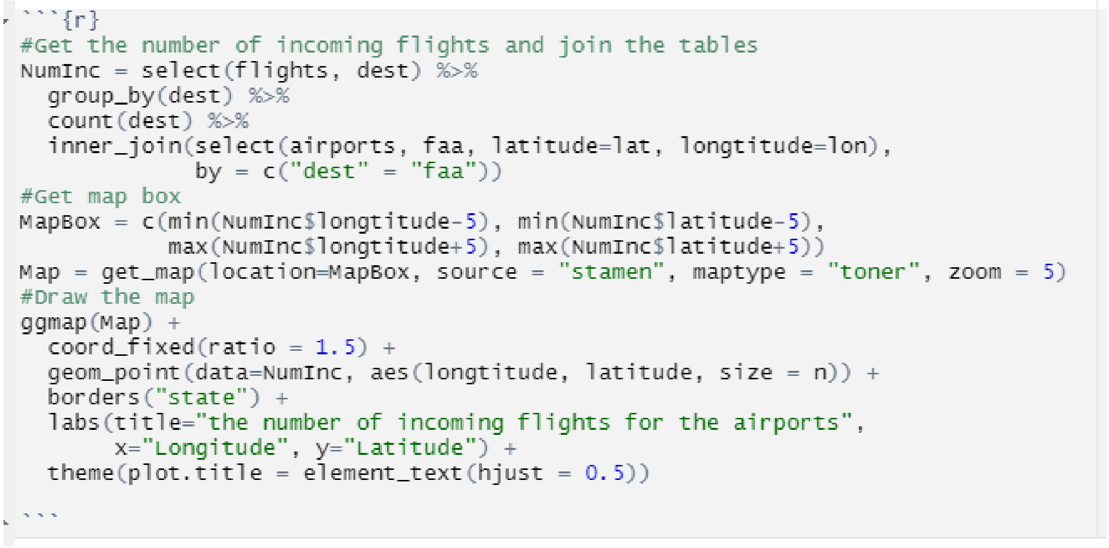
The correlation score matrix of the first 100 rows is shown as below.



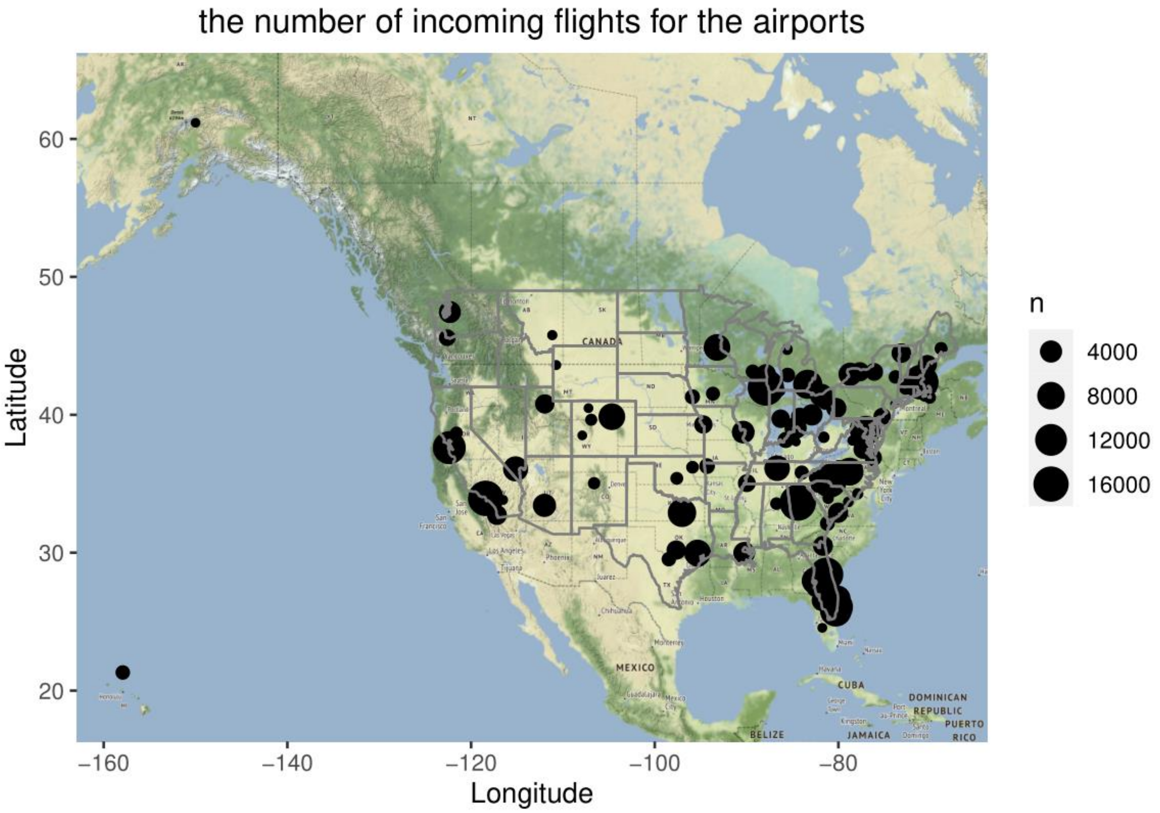
```
             dep_delay   arr_delay         temp         dewp        humid wind_dir  wind_speed wind_gust      precip pressure
dep_delay   1.00000000  0.99141144 -0.244164029 -0.190737783  0.075527598       NA  0.03995421        NA -0.11226959       NA
arr_delay   0.99141144  1.00000000 -0.255868904 -0.200746018  0.076267446       NA  0.02719937        NA -0.11744065       NA
temp       -0.24416403 -0.25586890  1.000000000  0.927523735  0.005207706       NA -0.14697759        NA -0.03754560       NA
dewp       -0.19073778 -0.20074602  0.927523735  1.000000000  0.372485508       NA -0.13563836        NA  0.07438375       NA
humid       0.07552760  0.07626745  0.005207706  0.372485508  1.000000000       NA  0.01587175        NA  0.33360353       NA
wind_dir            NA          NA           NA           NA           NA        1          NA        NA          NA       NA
wind_speed  0.03995421  0.02719937 -0.146977585 -0.135638363  0.015871746       NA  1.00000000        NA  0.30185990       NA
wind_gust           NA          NA           NA           NA           NA       NA          NA         1          NA       NA
precip     -0.11226959 -0.11744065 -0.037545604  0.074383751  0.333603533       NA  0.30185990        NA  1.00000000       NA
pressure            NA          NA           NA           NA           NA       NA          NA        NA          NA        1
visib      -0.02719954 -0.03051827  0.219424052 -0.007113693 -0.637604371       NA -0.15583831        NA -0.32110642       NA
                  visib
dep_delay  -0.027199544
arr_delay  -0.030518275
temp        0.219424052
dewp       -0.007113693
humid      -0.637604371
wind_dir             NA
wind_speed -0.155838309
wind_gust            NA
precip     -0.321106419
pressure             NA
visib       1.000000000
```

As the correlation matrix shows, temperature and dewpoint temperature make it more likely to see a delay. As the temperature and dewp goes down, delay times go up. From my perspective, temperature usually can lead other weather conditions. With the low temperature it is, the weather conditions become worse for flights.

1.e Produce a map that sizes each destination airport by the number of incoming flights. You may use a continuous scale for the size. Here is a code snippet to draw a map of all flight destinations, which you can use as a starting point. You may need to install the maps packages if you have not already. Adjust the title, axis labels and aesthetics to make this visualization as clear as possible.

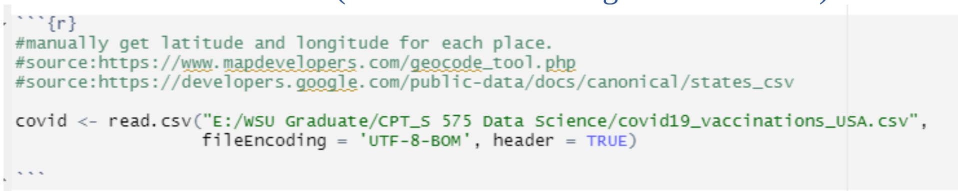


~~~
```{r}
#Get the number of incoming flights and join the tables
NumInc = select(flights, dest) %>%
  group_by(dest) %>%
  count(dest) %>%
  inner_join(select(airports, faa, latitude=lat, longtitude=lon),
             by = c("dest" = "faa"))
#Get map box
MapBox = c(min(NumInc$longtitude-5), min(NumInc$latitude-5),
           max(NumInc$longtitude+5), max(NumInc$latitude+5))
Map = get_map(location=MapBox, source = "stamen", maptype = "toner", zoom = 5)
#Draw the map
ggmap(Map) +
  coord_fixed(ratio = 1.5) +
  geom_point(data=NumInc, aes(longtitude, latitude, size = n)) +
  borders("state") +
  labs(title="the number of incoming flights for the airports",
       x="Longitude", y="Latitude") +
  theme(plot.title = element_text(hjust = 0.5))

```
~~~



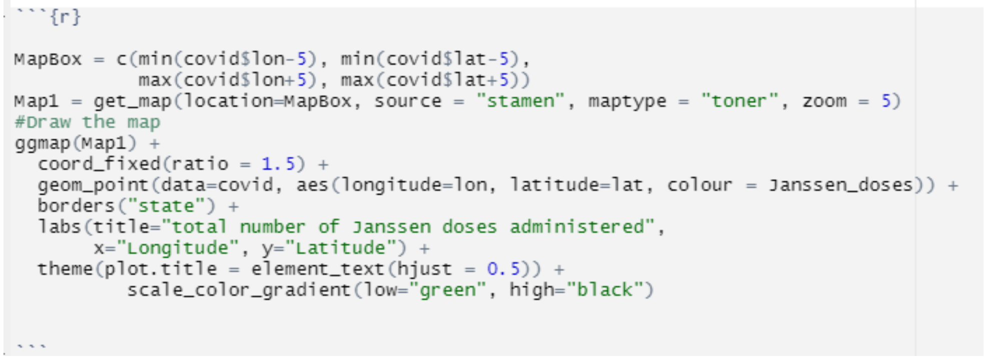
Problem 2

I failed to get the geocode for each place through geocode function from google, so I manually get latitude and longitude for each place, I also post the sources where I got geocode, there might be several little mistakes. (I found there is one geocode is weird)

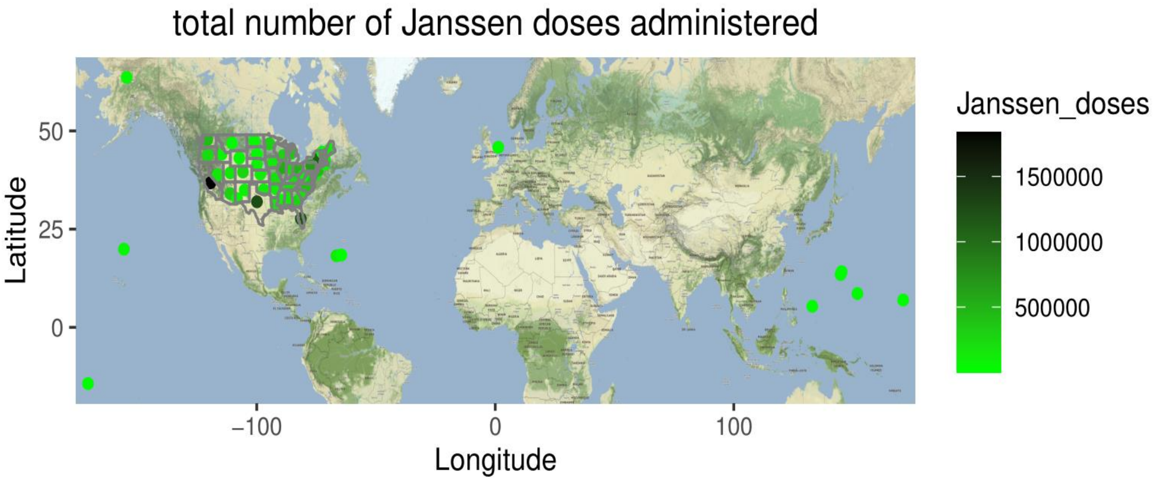


~~~
```{r}
#manually get latitude and longitude for each place.
#source:https://www.mapdevelopers.com/geocode_tool.php
#source:https://developers.google.com/public-data/docs/canonical/states_csv

covid <- read.csv("E:/WSU Graduate/CPT_S 575 Data Science/covid19_vaccinations_USA.csv",
                  fileEncoding = 'UTF-8-BOM', header = TRUE)

```
~~~

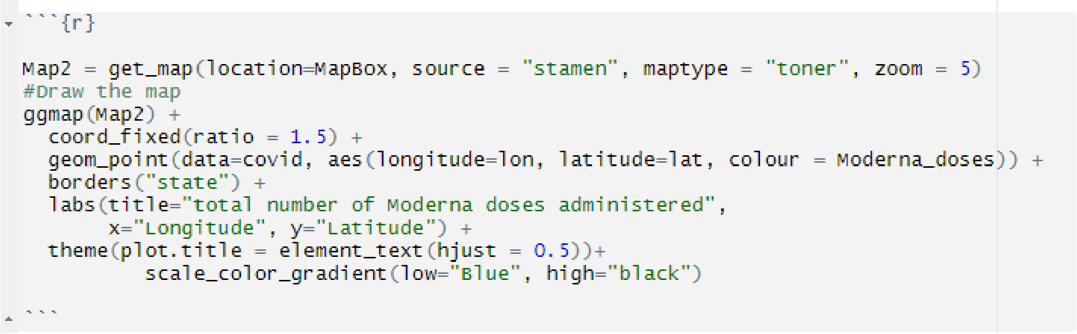
Draw the first map which shows the total number of Janssen doses administered.



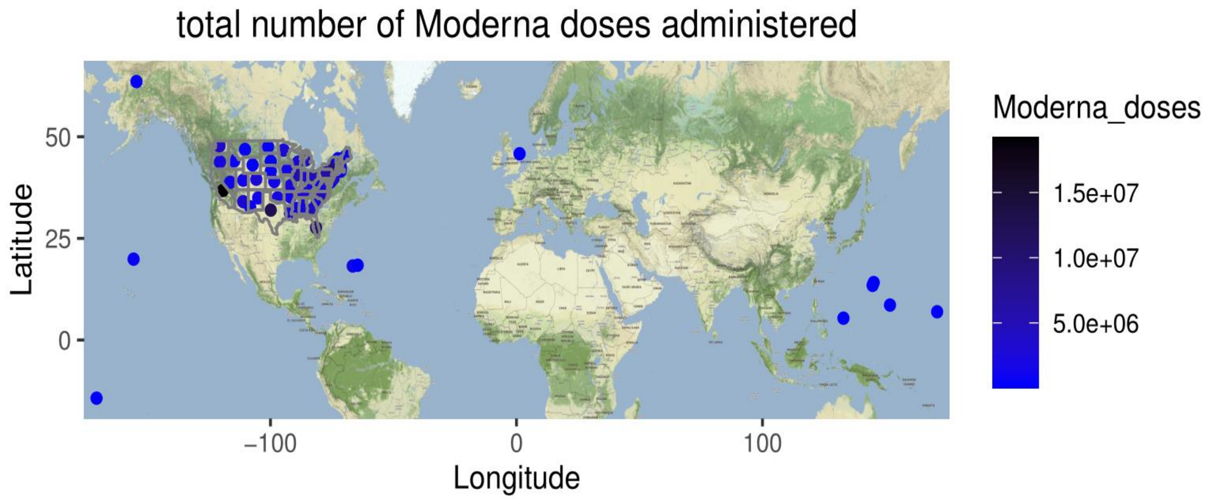
~~~
```{r}

MapBox = c(min(covid$lon-5), min(covid$lat-5),
           max(covid$lon+5), max(covid$lat+5))
Map1 = get_map(location=MapBox, source = "stamen", maptype = "toner", zoom = 5)
#Draw the map
ggmap(Map1) +
  coord_fixed(ratio = 1.5) +
  geom_point(data=covid, aes(longitude=lon, latitude=lat, colour = Janssen_doses)) +
  borders("state") +
  labs(title="total number of Janssen doses administered",
       x="Longitude", y="Latitude") +
  theme(plot.title = element_text(hjust = 0.5)) +
          scale_color_gradient(low="green", high="black")


```
~~~



Draw the second map which shows the total number of Moderna doses administered.

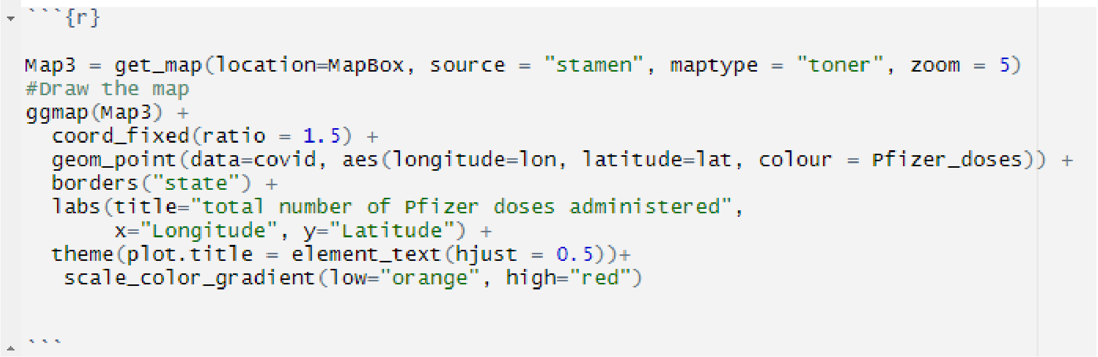


~~~
```{r}

Map2 = get_map(location=MapBox, source = "stamen", maptype = "toner", zoom = 5)
#Draw the map
ggmap(Map2) +
  coord_fixed(ratio = 1.5) +
  geom_point(data=covid, aes(longitude=lon, latitude=lat, colour = Moderna_doses)) +
  borders("state") +
  labs(title="total number of Moderna doses administered",
       x="Longitude", y="Latitude") +
  theme(plot.title = element_text(hjust = 0.5))+
          scale_color_gradient(low="Blue", high="black")

```
~~~

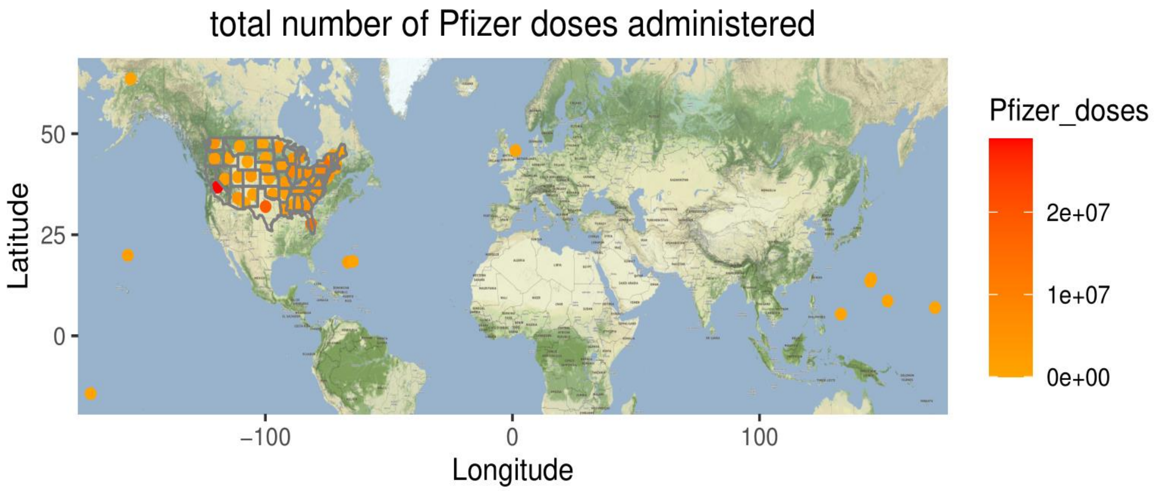


total number of Moderna doses administered

Draw the third map which shows the total number of Pfizer doses administered.



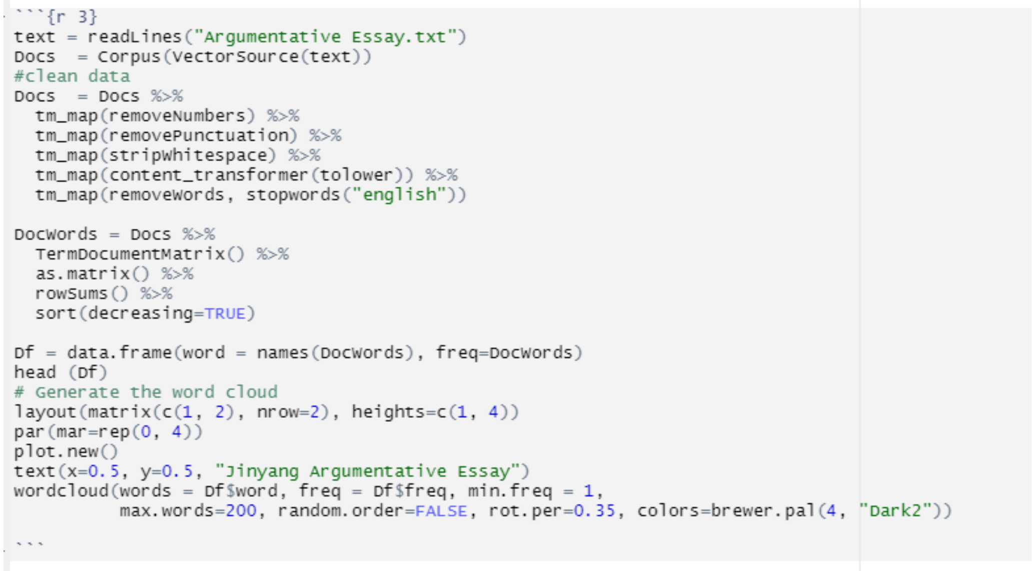
~~~
```{r}

Map3 = get_map(location=MapBox, source = "stamen", maptype = "toner", zoom = 5)
#Draw the map
ggmap(Map3) +
  coord_fixed(ratio = 1.5) +
  geom_point(data=covid, aes(longitude=lon, latitude=lat, colour = Pfizer_doses)) +
  borders("state") +
  labs(title="total number of Pfizer doses administered",
       x="Longitude", y="Latitude") +
  theme(plot.title = element_text(hjust = 0.5))+
   scale_color_gradient(low="orange", high="red")


```
~~~



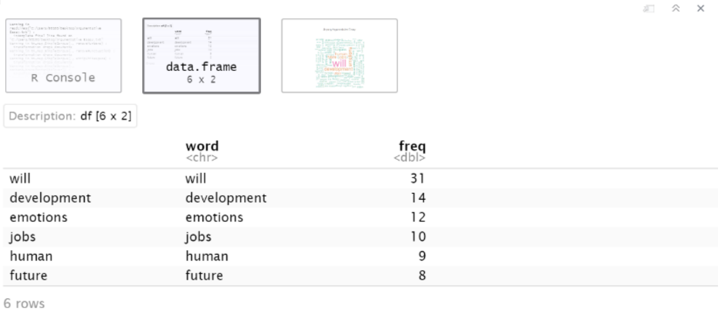
total number of Pfizer doses administered

Problem 3

I chose an argumentative essay I wrote when I was learning English last year. I also generate the word frequency data frame for the figure.



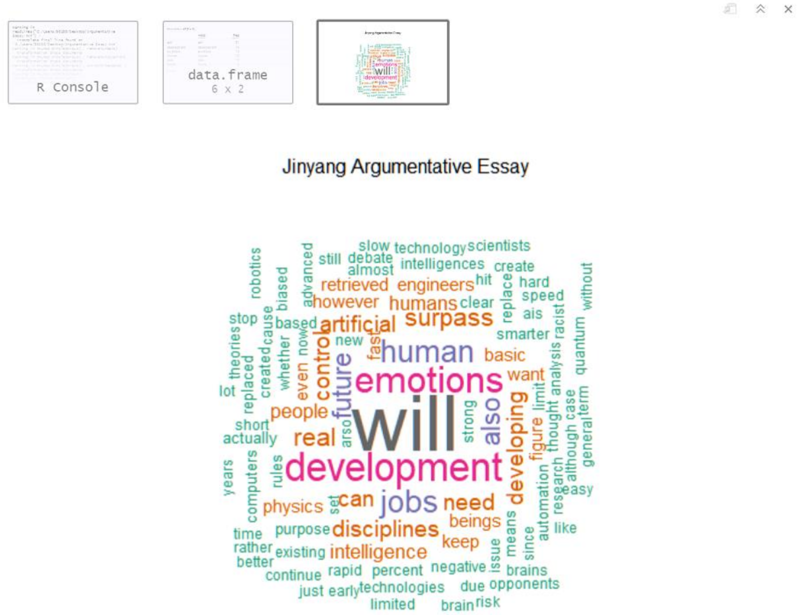
~~~
```{r 3}
text = readLines("Argumentative Essay.txt")
Docs   = Corpus(VectorSource(text))
#clean data
Docs   = Docs %>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace) %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removeWords, stopwords("english"))

DocWords = Docs %>%
  TermDocumentMatrix() %>%
  as.matrix() %>%
  rowSums() %>%
  sort(decreasing=TRUE)

Df = data.frame(word = names(DocWords), freq=DocWords)
head (Df)
# Generate the word cloud
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, "Jinyang Argumentative Essay")
wordcloud(words = Df$word, freq = Df$freq, min.freq = 1,
          max.words=200, random.order=FALSE, rot.per=0.35, colors=brewer.pal(4, "Dark2"))

```
~~~



Description: df [6 x 2]

	word <chr>	freq <dbl>
will	will	31
development	development	14
emotions	emotions	12
jobs	jobs	10
human	human	9
future	future	8

6 rows



Jinyang Ruan’s argumentative essay on AI future, written in December 2020.