Assignment 4

2022-09-30

Assignment 4

Problem 1

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
flights = read.csv("flights.csv", sep = ",", header = TRUE)
airlines = read.csv("airlines.csv", sep = ",", header = TRUE)
airports = read.csv("airports.csv", sep = ",", header = TRUE)
planes = read.csv("planes.csv", sep = ",", header = TRUE)
weather = read.csv("weather.csv", sep = ",", header = TRUE)
```

a.)

```
filtered_data = flights %>%
  left_join(weather) %>%
  filter(dest == "TPA", dep_time >= 1200, arr_time <= 1800, month == 11, day == 1, year == 2013) %>%
  select(tailnum, year, month, day, hour, origin, humid)
```

```
## Joining, by = c("year", "month", "day", "origin", "hour", "time_hour")
```

```
filtered_data
```

```
##   tailnum year month day hour origin humid
## 1  N580JB 2013    11   1   14    JFK 63.08
## 2  N337NB 2013    11   1   14    LGA 56.51
## 3  N567UA 2013    11   1   15    EWR 52.80
```

```
count(filtered_data)
```

```
##   n
## 1 3
```

There were three flights during the given time frame.

b.)

The difference between the two lines,

1 - anti_join(flights, airports, by = c("origin" = "faa")) 2 - anti_join(airports, flights, by = c("faa" = "origin"))

is that in line 1, it will filter the left data set with the column "origin" based on an existing entry in the column "faa" in the right data set. While in line 2, it is the opposite case where it filters the left data set (with column "faa") based on the right data set (with the column "origin").

Moreover, semi_join filters the left data set using the right data set, based on similar entries found in columns in both. While a anti_join does the same thing except instead of filtering the left data set when an existing entry is found in the right data set, it filters the left data set if no entry is found in the right.

c.)

```
flight_count = airports %>%
  inner_join(flights, c("faa" = "dest")) %>%
  mutate(dest = faa) %>%
  select(-faa) %>%
  select(origin, dest, lat, lon)

count(flight_count)
```

```
##        n
## 1 329174
```

There are 329,174 flights.

d.)

```
unique_count = flights %>%
  group_by(dest) %>%
  summarise(carrier) %>%
  unique()
```

```
## `summarise()` has grouped output by 'dest'. You can override using the
## `.groups` argument.
```

```
nrow(unique_count)
```

```
## [1] 314
```

There are 314 unique combinations of carrier/dest.

e.)

```
library(ggplot2)
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(maptools)
```

```
## Loading required package: sp

## Checking rgeos availability: FALSE
## Please note that 'maptools' will be retired by the end of 2023,
## plan transition at your earliest convenience;
## some functionality will be moved to 'sp'.
##      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpcl
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

```
library(mapproj)
```

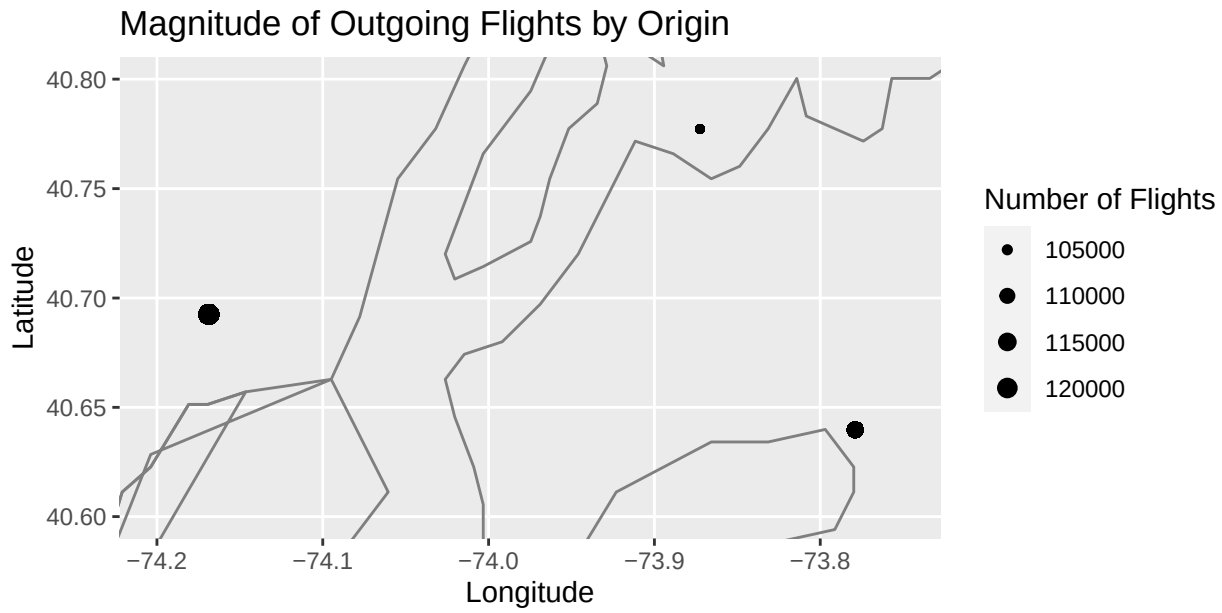
```
## Loading required package: maps

##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map
```

```
outgoing_flights = flights %>%
  left_join(airports, c("origin" = "faa")) %>%
  select(origin, lon, lat)

outgoing_flights$origin = as.factor(outgoing_flights$origin)
str(outgoing_flights)
```

```
## 'data.frame':    336776 obs. of  3 variables:
##  $ origin: Factor w/ 3 levels "EWR","JFK","LGA": 1 3 2 2 3 1 1 3 2 3 ...
##  $ lon   : num  -74.2 -73.9 -73.8 -73.8 -73.9 ...
##  $ lat   : num  40.7 40.8 40.6 40.6 40.8 ...
```

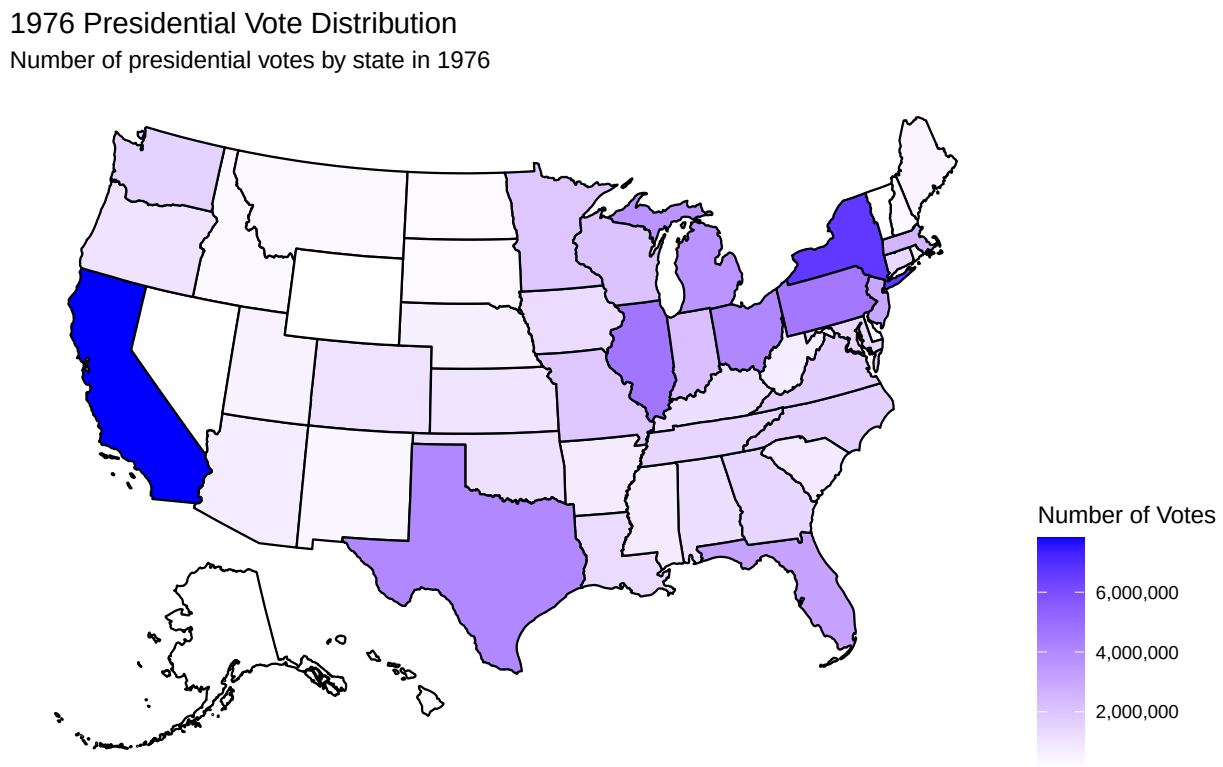
By turning the origin column of 'outgoing_flights' into a factor, we know that there are only three origin points.

```
outgoing_flights = flights %>%
  left_join(airports, c("origin" = "faa")) %>%
  select(origin, lon, lat)

count = c(length(which(outgoing_flights$origin == "EWR")),
          length(which(outgoing_flights$origin == "LGA")),
          length(which(outgoing_flights$origin == "JFK")))
column = c("EWR", "LGA", "JFK")
size = data.frame(column, count)

outgoing_flights = outgoing_flights %>%
  left_join(size, c("origin" = "column"))

ggplot(outgoing_flights, aes(x = lon, y = lat, size = count)) +
  geom_point(alpha= 0.2) +
  scale_size_continuous(range = c(1, 3), name = "Number of Flights") +
  borders("state") +
  geom_point() +
  coord_map(xlim = c(-74.2, -73.75), ylim = c(40.6, 40.8)) +
  labs(x = "Longitude", y = "Latitude", title = "Magnitude of Outgoing Flights by Origin")
```



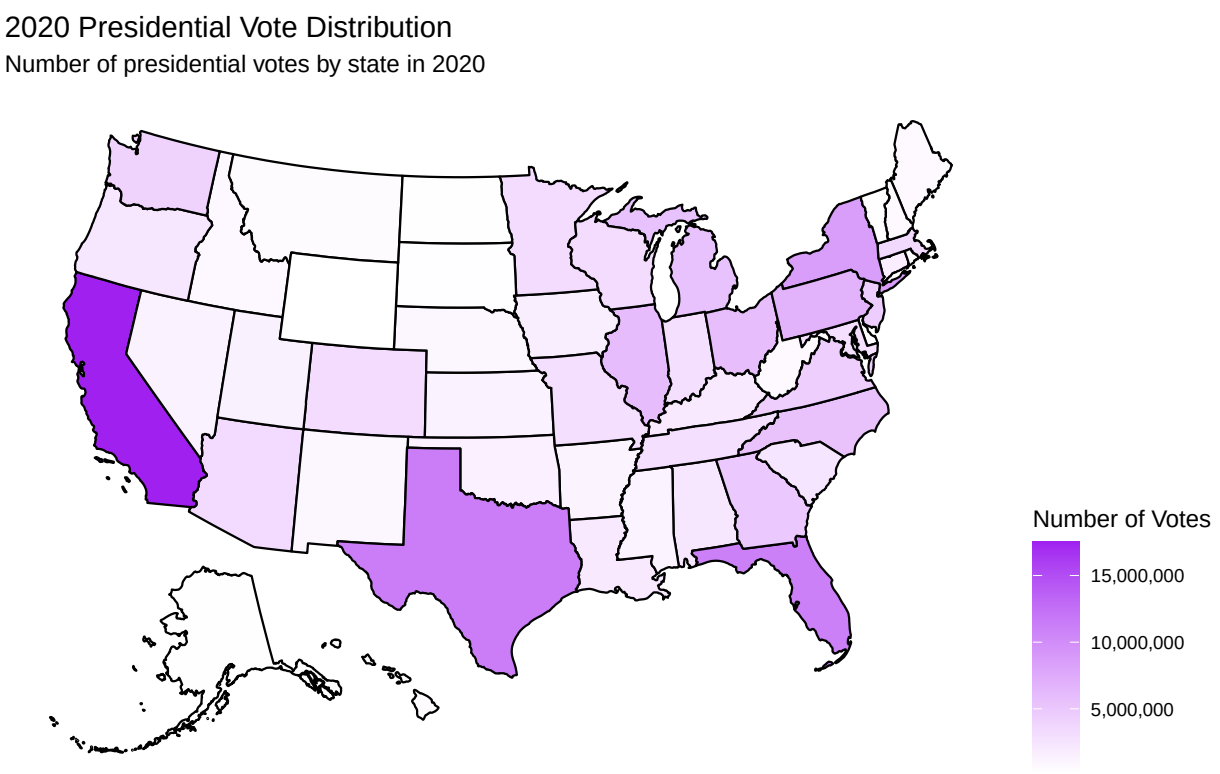
Problem 2

```
library(usmap)

us_predidents = read.csv("us-presidents.csv", sep = ",", header = TRUE)

pres_year1 = us_predidents %>%
  subset(select = -office)
pres_year1 = pres_year1[pres_year1$year == 1976, ]

pres_year2 = us_predidents %>%
  subset(select = -office)
pres_year2 = pres_year2[pres_year2$year == 2020, ]

plot_usmap(data = pres_year1, values = "totalvotes") +
  scale_fill_continuous(low = "white", high = "blue",
                        name = "Number of Votes", label = scales::comma) +
  labs(title = "1976 Presidential Vote Distribution",
       subtitle = "Number of presidential votes by state in 1976") +
  theme(legend.position = "right")
```




```
plot_usmap(data = pres_year2, values = "totalvotes") +
  scale_fill_continuous(low = "white", high = "purple", name = "Number of Votes",
                        label = scales::comma) +
```

```
  labs(title = "2020 Presidential Vote Distribution",
       subtitle = "Number of presidential votes by state in 2020") +
  theme(legend.position = "right")
```



In relation to the other states, Alaska, Hawaii and the center states did not have as many votes relative to the total number of votes, both in the past (1976) and recent past/present (2020). Moreover, California and Texas still have more votes in relation to the other states, both in the past and present, relative to the total number of votes. Furthermore, it seems that the northeastern states have decreased in the number of votes, while Florida has increased, between 1976 and 2020, relative to the total number of votes in both eras.

Problem 3

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
text = readLines(file.choose())
```

```
## Warning in readLines(file.choose()): incomplete final line found on 'C:
## \Users\denis\Documents\WSU\3 Junior\2 Junior 2nd Semester\Data Science (Cpts
## 475)\Assignments\Assignment 4\Assignment 4\The Egg.txt'
```

```
document = Corpus(VectorSource(text))

space = content_transformer(function (x, pattern) gsub(pattern, " ", x))
document = tm_map(document, space, "/")
```

```
## Warning in tm_map.SimpleCorpus(document, space, "/"): transformation drops
## documents
```

```
document = tm_map(document, space, "@")
```

```
## Warning in tm_map.SimpleCorpus(document, space, "@"): transformation drops
## documents
```

```
document = tm_map(document, space, "\\|")
```

```
## Warning in tm_map.SimpleCorpus(document, space, "\\|"): transformation drops
## documents
```

```
document = tm_map(document, space, """)
```

```
## Warning in tm_map.SimpleCorpus(document, space, """): transformation drops
## documents
```

```
document = tm_map(document, space, """)
```

```
## Warning in tm_map.SimpleCorpus(document, space, """): transformation drops
## documents
```

```
document = tm_map(document, space, "'")
```

```
## Warning in tm_map.SimpleCorpus(document, space, "'"): transformation drops
## documents
```

```
document = tm_map(document, space, "...")
```

```
## Warning in tm_map.SimpleCorpus(document, space, "..."): transformation drops
## documents
```

```
document = tm_map(document, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(document, content_transformer(tolower)):
## transformation drops documents
```

```
document = tm_map(document, removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(document, removeNumbers): transformation drops
## documents
```

```
document = tm_map(document, removeWords, stopwords("english"))
```

```
## Warning in tm_map.SimpleCorpus(document, removeWords, stopwords("english")):
## transformation drops documents
```

```
document = tm_map(document, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(document, removePunctuation): transformation
## drops documents
```

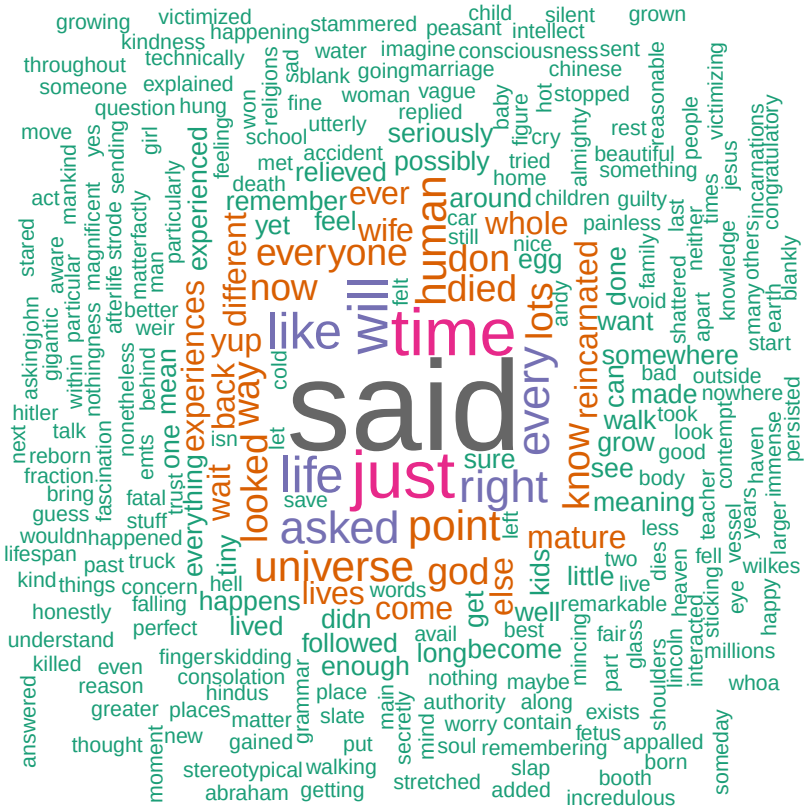
```
document = tm_map(document, stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(document, stripWhitespace): transformation drops
## documents
```

```
tdoc_matrix = TermDocumentMatrix(document)
doc_matrix = as.matrix(tdoc_matrix)
frequent = sort(rowSums(doc_matrix), decreasing = TRUE)
text_data = data.frame(word = names(frequent), freq = frequent)

str(text_data)
```

```
## 'data.frame':    279 obs. of  2 variables:
##  $ word: chr  "said" "just" "time" "will" ...
##  $ freq: num  22 10 10 8 7 7 6 6 6 5 ...
```

```
wordcloud(words = text_data$word, freq = text_data$freq, min.freq = 1, max.words = 300,
          random.order = FALSE,
          rot.per = 0.35, colors = brewer.pal(8, "Dark2"))
```



"The Egg", a short story by Andy Weir.