Assignment 4: Joins and Visualization

WenLi

2020/10/3

Problem 1

This problem will involve the nycflights13 dataset (including tables airlines, airports, planes and weather), which we saw in class. Start by installing and importing the dataset to your chosen platform. We will first use joins to search and manipulate the dataset, then we will produce a flightpath visualization.

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) on the afternoon of November 1, 2013.

```r
# just getting a narrower dataframe
MyFlights = flights %>%
            select(year:day, hour, origin, dest, tailnum, carrier)
# and now doing a left join
Result = MyFlights %>%
  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)
```

```
## # 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
```

b. What is the difference between the following two joins? anti_join(flights, airports, by = c("dest" = "faa")) anti_join(airports, flights, by = c("faa" = "dest"))

- 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 results 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 results is a subset of table airports

c. Select the origin and destination airports and their latitude and longitude for all fights in the dataset (using one or more inner joins).

```
Result = MyFlights %>%
  select (origin, dest) %>%
  inner_join(select(airports, faa, origin_lat=lat, origin_lon=lon), by = c("origin" = "faa")) %>%
  inner_join(select(airports, faa, dest_lat=lat, dest_lon=lon), by = c("dest" = "faa"))
as_tibble(Result)
```

```
## # A tibble: 329,174 x 6
##    origin dest  origin_lat origin_lon dest_lat dest_lon
##    <chr>  <chr>      <dbl>      <dbl>    <dbl>    <dbl>
##  1 EWR    IAH         40.7      -74.2     30.0    -95.3
##  2 LGA    IAH         40.8      -73.9     30.0    -95.3
##  3 JFK    MIA         40.6      -73.8     25.8    -80.3
##  4 LGA    ATL         40.8      -73.9     33.6    -84.4
##  5 EWR    ORD         40.7      -74.2     42.0    -87.9
##  6 EWR    FLL         40.7      -74.2     26.1    -80.2
##  7 LGA    IAD         40.8      -73.9     38.9    -77.5
##  8 JFK    MCO         40.6      -73.8     28.4    -81.3
##  9 LGA    ORD         40.8      -73.9     42.0    -87.9
## 10 JFK    PBI         40.6      -73.8     26.7    -80.1
## # ... with 329,164 more rows
```

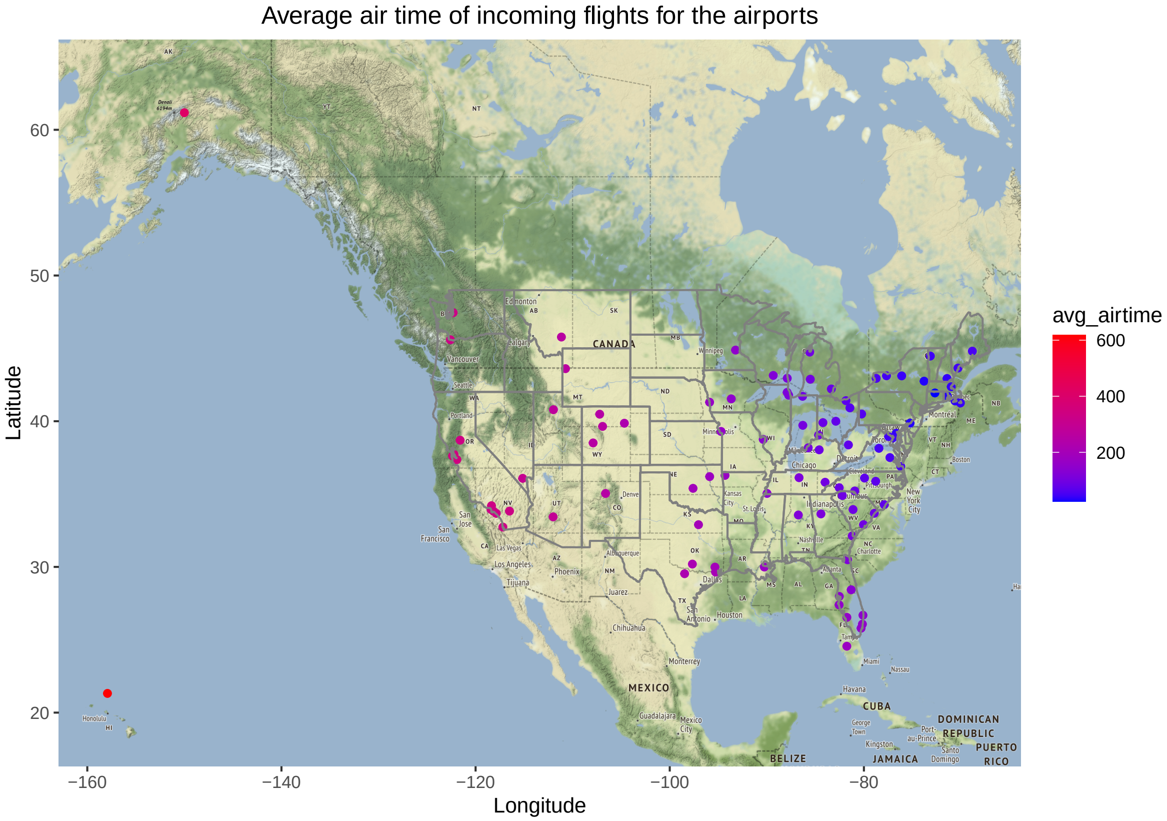
d. Use groupby and count to get the number of flights to each unique origin/destination combination.

```
Result = MyFlights %>%
        group_by (origin, dest) %>%
        dplyr::count (origin, dest)
as_tibble (Result)
```

```
## # A tibble: 224 x 3
##    origin dest      n
##    <chr>  <chr> <int>
##  1 EWR    ALB     439
##  2 EWR    ANC       8
##  3 EWR    ATL    5022
##  4 EWR    AUS     968
##  5 EWR    AVL     265
##  6 EWR    BDL     443
##  7 EWR    BNA    2336
##  8 EWR    BOS    5327
##  9 EWR    BQN     297
## 10 EWR    BTV     931
## # ... with 214 more rows
```

e. Produce a map that colors each destination airport by the average air time of its incoming flights. 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.

```
# Get average time and join the tables
AvgairTime = select(flights, dest, air_time) %>%
  group_by(dest) %>%
  dplyr::summarise(avg_airtime = mean(air_time, na.rm = TRUE)) %>%
  inner_join(select(airports, faa, latitude=lat, longtitude=lon), by = c("dest" = "faa"))
# Get map box
MapBox = c(min(AvgairTime$longtitude)-5, min(AvgairTime$latitude)-5,
           max(AvgairTime$longtitude)+5, max(AvgairTime$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=AvgairTime, aes(longtitude, latitude, colour=avg_airtime)) +
  borders("state") +
  labs(title="Average air time of incoming flights for the airports", x="Longitude", y="Latitude") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_gradient(low="blue", high="red")
```



Problem 2

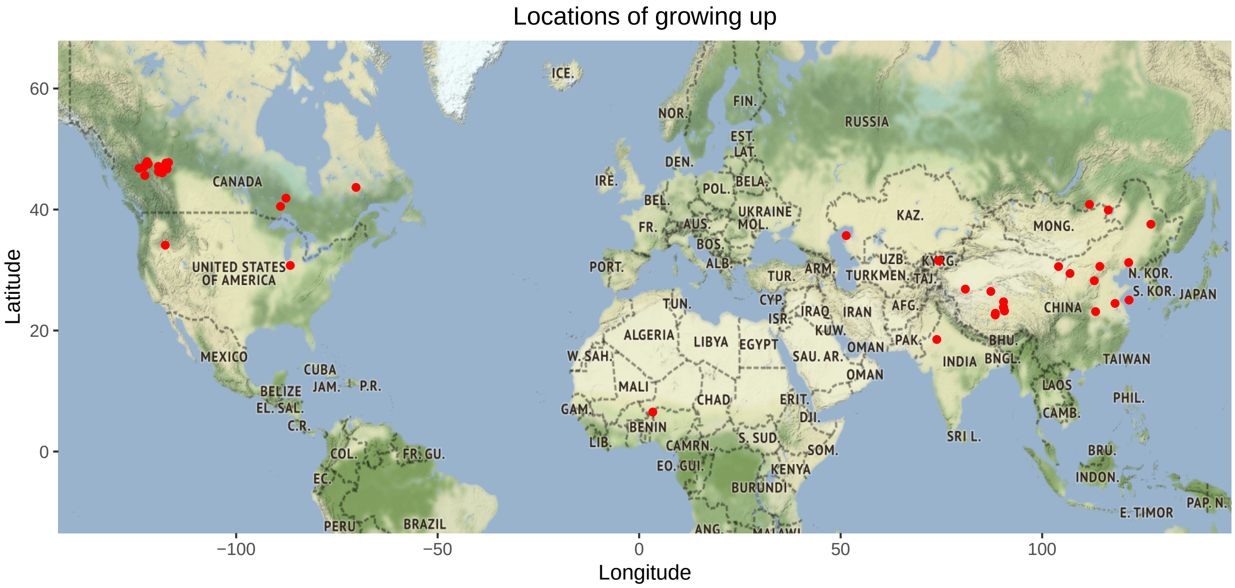
You may recall on the lecture on Friday, Sep 25 (when we had Dr. Ofer Amram as a guest speaker), the warm-up question that day was to type in the city and state (or city and country) where you grew up in. The result of that warm-up question is summarized in a 2 file that is posted under Lecture/Sep 25. The task you have for this problem is to visualize that list on a world map, indicating in some way the cities. You could use the same mark to denote the cities. Try to make your visualization as nice looking as possible.

```
register_google(key = APIKey)
# Get the coordinates of all locations
```

```
GupLoc     = read_rtf("warmUpQuestion-Sep25.RTF")
GupLocDf   = data.frame(Location=GupLoc, stringsAsFactors=FALSE)
GupLocMap = geocode(location=GupLoc, output="more", source="google")
GupLocMap = cbind(GupLocDf, GupLocMap) %>%
            select (Location, Longtitude=lon, Latitude=lat)
head (GupLocMap)
```

```
##                Location Longtitude Latitude
## 1   Seoul, South Korea  126.97797 37.56654
## 2  Olympia, Washington -122.90070 47.03787
## 3        Lucknow ,India   80.94617 26.84669
## 4          Richland, WA -119.27520 46.28042
## 5 Chandpur, Bangladesh   90.66307 23.23210
## 6    Lahore, Pakistan.   74.35875 31.52037
```

```
# Get map box according to the coordinates
MapBox = c(min(GupLocMap$Longtitude)-20, min(GupLocMap$Latitude)-20,
           max(GupLocMap$Longtitude)+20, max(GupLocMap$Latitude)+20)
Map = get_map(location=MapBox, source = "stamen", maptype = "toner", zoom = 3)
#Draw the map
ggmap(Map) +
  coord_fixed(ratio = 1.5) +
  geom_point(data=GupLocMap, aes(x=Longtitude, y=Latitude), color="red") +
  labs(title="Locations of growing up", x="Longitude", y="Latitude") +
  theme(plot.title = element_text(hjust = 0.5))
```



Problem 3

Create a word cloud for an interesting (relatively short, say a couple pages) document of your own choice. Examples of suitable documents include: summary of a recent project you are working or have worked on; your own recent Statement of Purpose or Research Statement or some other similar document.

```
# Load texts
Texts = readLines("PCAIntroduction.txt")
Docs  = Corpus(VectorSource(Texts))
# Clean the data
Docs  = Docs %>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace) %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removeWords, stopwords("english"))
# Create a document-term-matrix
DocWords = Docs %>%
  TermDocumentMatrix() %>%
  as.matrix() %>%
  rowSums() %>%
  sort(decreasing=TRUE)
Df = data.frame(word = names(DocWords), freq=DocWords)
head (Df)
```

```
##                              word freq
## analysis                 analysis   24
## data                         data   18
## pca                           pca   14
## dependence             dependence   11
## interprocedural interprocedural   10
## llvm                         llvm    8
```

```
# 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, "Introduction of PCA: a new program analysis tool")
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"))
```

Introduction of PCA: a new program analysis tool