Case Study 4

Hang Tian
UB PERSON ID: 50413372

2022-09-27

1. Import packages and get data

```
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   1.2.1      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()
```

```
library(nycflights13)
library(sp)
library(sf)
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
data(flights)
str(flights)
```

```
## tibble [336,776 x 19] (S3: tbl_df/tbl/data.frame)
##  $ year          : int [1:336776] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int [1:336776] 517 533 542 544 554 554 555 557 557 558 ...
##  $ sched_dep_time: int [1:336776] 515 529 540 545 600 558 600 600 600 600 ...
##  $ dep_delay     : num [1:336776] 2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int [1:336776] 830 850 923 1004 812 740 913 709 838 753 ...
##  $ sched_arr_time: int [1:336776] 819 830 850 1022 837 728 854 723 846 745 ...
##  $ arr_delay     : num [1:336776] 11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr [1:336776] "UA" "UA" "AA" "B6" ...
##  $ flight        : int [1:336776] 1545 1714 1141 725 461 1696 507 5708 79 301 ...
##  $ tailnum       : chr [1:336776] "N14228" "N24211" "N619AA" "N804JB" ...
##  $ origin        : chr [1:336776] "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr [1:336776] "IAH" "IAH" "MIA" "BQN" ...
##  $ air_time      : num [1:336776] 227 227 160 183 116 150 158 53 140 138 ...
##  $ distance      : num [1:336776] 1400 1416 1089 1576 762 ...
##  $ hour          : num [1:336776] 5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num [1:336776] 15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct[1:336776], format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

```
data(airports)
str(airports)
```

```
## tibble [1,458 x 8] (S3: tbl_df/tbl/data.frame)
##  $ faa  : chr [1:1458] "04G" "06A" "06C" "06N" ...
##  $ name : chr [1:1458] "Lansdowne Airport" "Moton Field Municipal Airport" "Schaumburg Regional" "Ra
##  $ lat  : num [1:1458] 41.1 32.5 42 41.4 31.1 ...
##  $ lon  : num [1:1458] -80.6 -85.7 -88.1 -74.4 -81.4 ...
##  $ alt  : num [1:1458] 1044 264 801 523 11 ...
##  $ tz   : num [1:1458] -5 -6 -6 -5 -5 -5 -5 -5 -5 -8 ...
##  $ dst  : chr [1:1458] "A" "A" "A" "A" ...
##  $ tzone: chr [1:1458] "America/New_York" "America/Chicago" "America/Chicago" "America/New_York" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   name = col_character(),
##   ..   city = col_character(),
##   ..   country = col_character(),
##   ..   faa = col_character(),
##   ..   icao = col_character(),
##   ..   lat = col_double(),
##   ..   lon = col_double(),
##   ..   alt = col_double(),
##   ..   tz = col_double(),
##   ..   dst = col_character(),
##   ..   tzone = col_character()
##   .. )
```

2. Find names of airports within NYC

```
#NYC shapefile
nyc=st_read('nyc.shp')
```

```
## Reading layer `nyc' from data source
##   `/Users/hang/Semester files/GEO 511/GEO-511-2022/week_04/case_study/nyc.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 5 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -74.25559 ymin: 40.49613 xmax: -73.70001 ymax: 40.91553
## Geodetic CRS:  WGS84(DD)
```

```
nyc_crs=st_crs(nyc)
#Airports shapefile
airports=st_as_sf(airports,coords=c('lon','lat'))
airports=st_set_crs(airports,nyc_crs)
#Logic choose
is_in=st_within(airports,nyc)
is_in=as.numeric(is_in)
bool=!is.na(is_in)
airports_nyc=airports[bool,]
#Names for airports within NYC
names=airports_nyc$faa
names
```

```
## [1] "IDL" "JFK" "JRA" "JRB" "LGA" "NYC" "TSS" "ZYP"
```

#3. Process data and find result

```
farthest_distance_df <- flights %>%
  filter(origin %in% names)%>%
  group_by(origin)%>%
  summarise(distance=max(distance))


result <- left_join(farthest_distance_df,flights,by=c("origin","distance")) %>%
  select(c('origin','distance','dest'))%>%
  unique()%>%
  left_join(airports,by=c('dest'='faa'))%>%
  select(c("dest",'name'))

result
```

```
## # A tibble: 2 x 2
##   dest  name
##   <chr> <chr>
## 1 HNL   Honolulu Intl
## 2 DEN   Denver Intl
```