Assignment_1_STA5001_Code

2024-03-21

#If you want to knit into word (then export into pdf from word) #Ensure you install packages: webshot and webshot2

#Part a

#Number of observations and variables

```
taxi_df <- read.csv("C:/Users/Michael Le/Desktop/Taxi_Data/taxi.csv", header
= TRUE)

#Check any null values
sum(is.na(taxi_df))

## [1] 0
```

#There are 625134 observations and 9 variables for the taxi dataframe.Assuming the data is cleaned

#Check the head of the taxi dataframe

```
head(taxi_df)

##          id vendor_id     pickup_datetime passenger_count pickup_longitude
## 1 id3004672         1 2016-06-30 23:59:58               1        -73.98813
## 2 id3505355         1 2016-06-30 23:59:53               1        -73.96420
## 3 id1217141         1 2016-06-30 23:59:47               1        -73.99744
## 4 id2150126         2 2016-06-30 23:59:41               1        -73.95607
## 5 id1598245         1 2016-06-30 23:59:33               1        -73.97021
## 6 id0668992         1 2016-06-30 23:59:30               1        -73.99130
##   pickup_latitude dropoff_longitude dropoff_latitude store_and_fwd_flag
## 1        40.73203         -73.99017         40.75668                  N
## 2        40.67999         -73.95981         40.65540                  N
## 3        40.73758         -73.98616         40.72952                  N
## 4        40.77190         -73.98643         40.73047                  N
## 5        40.76147         -73.96151         40.75589                  N
## 6        40.74980         -73.98051         40.78655                  N
```

#Part b

#Compute values and add to the data frame taxi_df the variable dist with the Euclidean distance between pickup and dropoff locations (use the longitude as the x coordinates and the latitude as the y coordinate).

```
taxi_df$dist <-  sqrt((taxi_df$dropoff_latitude-taxi_df$pickup_latitude)^2 +
(taxi_df$dropoff_longitude -taxi_df$pickup_longitude)^2)
```

```r
head(taxi_df)
```

```
##          id vendor_id     pickup_datetime passenger_count pickup_longitude
## 1 id3004672         1 2016-06-30 23:59:58               1        -73.98813
## 2 id3505355         1 2016-06-30 23:59:53               1        -73.96420
## 3 id1217141         1 2016-06-30 23:59:47               1        -73.99744
## 4 id2150126         2 2016-06-30 23:59:41               1        -73.95607
## 5 id1598245         1 2016-06-30 23:59:33               1        -73.97021
## 6 id0668992         1 2016-06-30 23:59:30               1        -73.99130
##   pickup_latitude dropoff_longitude dropoff_latitude store_and_fwd_flag
## 1        40.73203         -73.99017         40.75668                  N
## 2        40.67999         -73.95981         40.65540                  N
## 3        40.73758         -73.98616         40.72952                  N
## 4        40.77190         -73.98643         40.73047                  N
## 5        40.76147         -73.96151         40.75589                  N
## 6        40.74980         -73.98051         40.78655                  N
##         dist
## 1 0.02473523
## 2 0.02497914
## 3 0.01386090
## 4 0.05136275
## 5 0.01034256
## 6 0.03830145
```

#What is the minimum distance between pickup and dropoff locations in this dataset(round and enter the answer with 3 decimal places)? What is the number of trips that have this minimum distance?

```r
sprintf("%.3f",min(taxi_df$dist))
```

```
## [1] "0.000"
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.3
```

```r
taxi_df %>% count(dist == min(taxi_df$dist))
```

```
##   dist == min(taxi_df$dist)      n
## 1                     FALSE 622659
## 2                      TRUE   2475

#There are 2475 trips that has the minimum distance.
```

625134

#Part c

Which first row in the data frame taxi_df corresponds to this distance?

```
#To extract the first row that corresponds the minimum distance.
taxi_df[which(taxi_df$dist==0, arr.ind=TRUE)[1],]

##            id vendor_id     pickup_datetime passenger_count pickup_longitu
de
## 128 id2195452         1 2016-06-30 23:22:00               1        -73.959
99
##     pickup_latitude dropoff_longitude dropoff_latitude store_and_fwd_flag
dist
## 128        40.77075         -73.95999         40.77075                  N
0
```

#Part d

#Use the library lubridate. The date 01/01/2016 can be entered in R by using the command ymd(20160101). Subset the data frame taxi_df by selecting trips with pickup_datetime which is smaller or equal to ymd(20160101).

```
#First we need to get the lubricate package in this order.
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.3.3

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'tibble' was built under R version 4.3.3

## Warning: package 'readr' was built under R version 4.3.3

## Warning: package 'purrr' was built under R version 4.3.3

## Warning: package 'forcats' was built under R version 4.3.3

## Warning: package 'lubridate' was built under R version 4.3.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.
0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.5
## ✓ ggplot2   3.5.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2
```

```
## ── Conflicts ─────────────────────────────────────── tidyverse_conflict
s() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(timechange)
```

```
## Warning: package 'timechange' was built under R version 4.3.3
```

```
library(lubridate)
library(readr)

newdf <- taxi_df[taxi_df$pickup_datetime <= ymd(20160101),]
head(newdf)
```

```
##               id vendor_id     pickup_datetime passenger_count pickup_long
itude
## 621954 id2071782         1 2016-01-01 23:59:08               1        -73.
95511
## 621955 id1078262         1 2016-01-01 23:59:01               2        -73.
98879
## 621956 id0462507         1 2016-01-01 23:58:51               1        -73.
99209
## 621957 id3355801         1 2016-01-01 23:58:09               1        -73.
97931
## 621958 id1493417         2 2016-01-01 23:58:02               5        -73.
82975
## 621959 id2926733         1 2016-01-01 23:58:02               1        -73.
98592
##        pickup_latitude dropoff_longitude dropoff_latitude store_and_fwd_fl
ag
## 621954        40.81504         -73.94047         40.82474
N
## 621955        40.74863         -73.95255         40.77308
N
## 621956        40.72733         -73.98468         40.74580
N
## 621957        40.78446         -73.97331         40.79275
N
## 621958        40.75623         -73.95930         40.81542
N
## 621959        40.75203         -73.97763         40.74547
N
##              dist
## 621954 0.01756298
## 621955 0.04371751
## 621956 0.01990100
## 621957 0.01023101
```

```
## 621958 0.14242814
## 621959 0.01056647
```

```
newdf$dist <-  sqrt((newdf $dropoff_latitude-newdf $pickup_latitude)^2 + (new
df$dropoff_longitude -newdf $pickup_longitude)^2)
```

#What is the maximum distance between pickup and dropoff locations in this subset (round and enter the answer with 3 decimal places)?What is the number of trips in this subset that have this maximum distance?

```
#The maximum distance in this subset.
sprintf("%.3f",max(newdf$dist))
```

```
## [1] "0.407"
```

#What is the number of trips in this subset that have this maximum distance?

```
#There is only one trip occurring in this subset.
newdf %>% count(dist == max(newdf$dist))
```

```
##   dist == max(newdf$dist)    n
## 1                   FALSE 3180
## 2                    TRUE    1
```

#Part e

#Subset the data frame taxi_df and select trips which dropoff longitudes are greater than -74 and dropoff latitudes are greater than 41. How many such trips are in the data frame? (0.5 mark)

```
taxi_df <- read.csv("C:/Users/Michael Le/Desktop/Taxi_Data/taxi.csv", header
= TRUE)
newdf_2<- taxi_df[taxi_df$dropoff_longitude > -74 & taxi_df$dropoff_latitude
> 41,]
head(newdf_2)
```

```
##              id vendor_id     pickup_datetime passenger_count pickup_longi
tude
## 675   id2970680         1 2016-06-30 20:40:14               1        -73.4
8360
## 5339  id3956502         1 2016-06-29 12:02:48               1        -73.9
9069
## 5392  id0609178         1 2016-06-29 11:40:15               1        -73.5
4888
## 6113  id0664915         1 2016-06-29 07:07:13               2        -73.8
8441
## 7165  id1454138         1 2016-06-28 20:16:07               1        -73.9
7695
## 22417 id3114359         2 2016-06-24 03:37:20               1        -73.9
7016
##       pickup_latitude dropoff_longitude dropoff_latitude store_and_fwd_fla
g
```

```
## 675          41.10648         -73.48360         41.10649
N
## 5339         40.75010         -73.75880         41.03841
N
## 5392         41.04432         -73.54888         41.04432
N
## 6113         40.74380         -73.85784         41.06486
N
## 7165         40.75538         -73.76338         41.03668
N
## 22417        40.74892         -73.53598         41.05465
N
```

#There are 94 trips in this dataframe.

#Part f

#Consider only dropoff longitudes and dropoff latitudes of the subset trips as the spatial coordinates and create a SpatialPoints object.

What is the minimum dropoff longitude coordinate in the bounding box (round and enter the answer with 3 decimal places)? What is the minimum dropoff latitude coordinate in the bounding box (round and enter the answer with 3 decimal places)? (1 mark)

```
newdf_3 <- newdf_2[,c('dropoff_longitude','dropoff_latitude')]
library(sp)

## Warning: package 'sp' was built under R version 4.3.3

#The minimum longitude
sprintf("%.3f",min(newdf_3$dropoff_longitude))

## [1] "-73.985"

#The minimum latitude
sprintf("%.3f",min(newdf_3$dropoff_latitude))

## [1] "41.007"
```

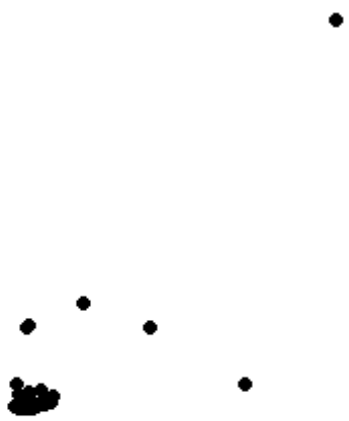
#Part g

Plot dropoff locations of the subset trips as 2D points. How many points are far away from the group with the majority of points?

```
library(sp)
points <- SpatialPoints(newdf_3[,c("dropoff_longitude","dropoff_latitude")])
str(points)
```

```
## Formal class 'SpatialPoints' [package "sp"] with 3 slots
##   ..@ coords     : num [1:94, 1:2] -73.5 -73.8 -73.5 -73.9 -73.8 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:94] "675" "5339" "5392" "6113" ...
##   .. .. ..$ : chr [1:2] "dropoff_longitude" "dropoff_latitude"
##   ..@ bbox       : num [1:2, 1:2] -74 41 -67.5 48.9
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "dropoff_longitude" "dropoff_latitude"
##   .. .. ..$ : chr [1:2] "min" "max"
##   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
##   .. .. ..@ projargs: chr NA
```

```
#It seems there are 5 points away the group with the majority of points.
plot(points,pch=19)
```



#Part hi

#Scale the variable dist in the data frame taxi_df by multiplying it by 20. Save it with the same name dist. Create a SpatialPointsDataFrame object using proj4string CRS("+proj=longlat +ellps=WGS84"), the pickup locations and the updated data frame taxi_df.

#Produce spplot of the trips from 101 to 200 for variables "vendor_id","passenger_count", and "dist". For the first interval (range of smallest values) in the plot legend, what are its (round and enter the answer with 3 decimal places) lower and upper bounds (1 mark)

```
#Refer from the additional workshop
library(sp)
taxi_df <- read.csv("C:/Users/Michael Le/Desktop/Taxi_Data/taxi.csv", header
= TRUE)
taxi_df$dist <-  sqrt((taxi_df$dropoff_latitude-taxi_df$pickup_latitude)^2 +
(taxi_df$dropoff_longitude -taxi_df$pickup_longitude)^2)

head(taxi_df)

##          id vendor_id     pickup_datetime passenger_count pickup_longitude
## 1 id3004672         1 2016-06-30 23:59:58               1        -73.98813
## 2 id3505355         1 2016-06-30 23:59:53               1        -73.96420
## 3 id1217141         1 2016-06-30 23:59:47               1        -73.99744
## 4 id2150126         2 2016-06-30 23:59:41               1        -73.95607
## 5 id1598245         1 2016-06-30 23:59:33               1        -73.97021
## 6 id0668992         1 2016-06-30 23:59:30               1        -73.99130
##   pickup_latitude dropoff_longitude dropoff_latitude store_and_fwd_flag
## 1        40.73203         -73.99017         40.75668                  N
## 2        40.67999         -73.95981         40.65540                  N
## 3        40.73758         -73.98616         40.72952                  N
## 4        40.77190         -73.98643         40.73047                  N
## 5        40.76147         -73.96151         40.75589                  N
## 6        40.74980         -73.98051         40.78655                  N
##         dist
## 1 0.02473523
## 2 0.02497914
## 3 0.01386090
## 4 0.05136275
## 5 0.01034256
## 6 0.03830145

#Using the coordinates
taxi_df1 <- cbind(taxi_df$pickup_latitude,taxi_df$pickup_longitude)

str(taxi_df1)

##  num [1:625134, 1:2] 40.7 40.7 40.7 40.8 40.8 ...
```

```r
llCRS <- CRS("+proj=longlat +ellps=WGS84")
df <- SpatialPoints(taxi_df1, proj4string=llCRS)
summary(df)
```

```
## Object of class SpatialPoints
## Coordinates:
##                   min       max
## coords.x1   37.38959  42.81494
## coords.x2 -121.93313 -69.24892
## Is projected: FALSE
## proj4string : [+proj=longlat +ellps=WGS84 +no_defs]
## Number of points: 625134
```

```r
bbox(df)
```

```
##                   min       max
## coords.x1   37.38959  42.81494
## coords.x2 -121.93313 -69.24892
```

```r
l2lon <- which(df$coords.x2 %in% sort(df$coords.x2)[1:2])
coordinates(df)[l2lon,]
```

```
##      coords.x1 coords.x2
## [1,]  37.38959 -121.9331
## [2,]  40.73916  -79.4879
```

```r
df1 <- SpatialPointsDataFrame(taxi_df1, taxi_df, proj4string=llCRS, match.ID=
TRUE)
str(df1)
```

```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
##   ..@ data       :'data.frame':  625134 obs. of  10 variables:
##   .. ..$ id                : chr [1:625134] "id3004672" "id3505355" "id121
7141" "id2150126" ...
##   .. ..$ vendor_id         : int [1:625134] 1 1 1 2 1 1 1 1 2 2 ...
##   .. ..$ pickup_datetime   : chr [1:625134] "2016-06-30 23:59:58" "2016-06
-30 23:59:53" "2016-06-30 23:59:47" "2016-06-30 23:59:41" ...
##   .. ..$ passenger_count   : int [1:625134] 1 1 1 1 1 1 1 2 2 1 ...
##   .. ..$ pickup_longitude  : num [1:625134] -74 -74 -74 -74 -74 ...
##   .. ..$ pickup_latitude   : num [1:625134] 40.7 40.7 40.7 40.8 40.8 ...
##   .. ..$ dropoff_longitude : num [1:625134] -74 -74 -74 -74 -74 ...
##   .. ..$ dropoff_latitude  : num [1:625134] 40.8 40.7 40.7 40.7 40.8 ...
##   .. ..$ store_and_fwd_flag: chr [1:625134] "N" "N" "N" "N" ...
##   .. ..$ dist              : num [1:625134] 0.0247 0.025 0.0139 0.0514 0.0
103 ...
##   ..@ coords.nrs : num(0)
##   ..@ coords     : num [1:625134, 1:2] 40.7 40.7 40.7 40.8 40.8 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:2] "coords.x1" "coords.x2"
##   ..@ bbox       : num [1:2, 1:2] 37.4 -121.9 42.8 -69.2
```

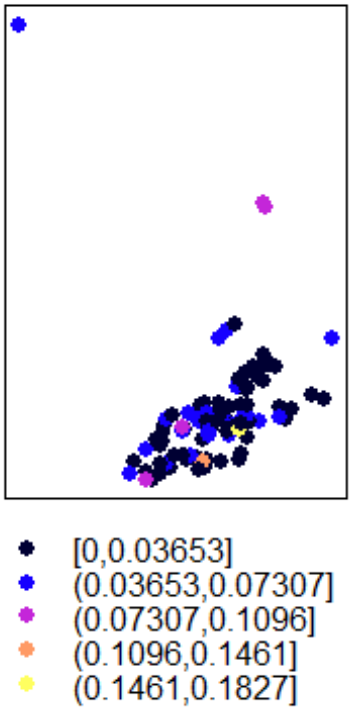
```
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "coords.x1" "coords.x2"
##   .. .. ..$ : chr [1:2] "min" "max"
##   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
##   .. .. ..@ projargs: chr "+proj=longlat +ellps=WGS84 +no_defs"
##   .. .. ..$ comment: chr "GEOGCRS[\"unknown\",\n    DATUM[\"Unknown based
on WGS 84 ellipsoid\",\n        ELLIPSOID[\"WGS 84\",6378137,29"| __truncated
__
```

```
summary(df1)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##                  min       max
## coords.x1   37.38959  42.81494
## coords.x2 -121.93313 -69.24892
## Is projected: FALSE
## proj4string : [+proj=longlat +ellps=WGS84 +no_defs]
## Number of points: 625134
## Data attributes:
##       id              vendor_id     pickup_datetime    passenger_count
##  Length:625134      Min.   :1.000   Length:625134      Min.   :0.000
##  Class :character   1st Qu.:1.000   Class :character   1st Qu.:1.000
##  Mode  :character   Median :2.000   Mode  :character   Median :1.000
##                     Mean   :1.535                      Mean   :1.662
##                     3rd Qu.:2.000                      3rd Qu.:2.000
##                     Max.   :2.000                      Max.   :9.000
##  pickup_longitude  pickup_latitude dropoff_longitude dropoff_latitude
##  Min.   :-121.93   Min.   :37.39   Min.   :-121.93   Min.   :36.60
##  1st Qu.: -73.99   1st Qu.:40.74   1st Qu.: -73.99   1st Qu.:40.74
##  Median : -73.98   Median :40.75   Median : -73.98   Median :40.75
##  Mean   : -73.97   Mean   :40.75   Mean   : -73.97   Mean   :40.75
##  3rd Qu.: -73.97   3rd Qu.:40.77   3rd Qu.: -73.96   3rd Qu.:40.77
##  Max.   : -69.25   Max.   :42.81   Max.   : -67.50   Max.   :48.86
##  store_and_fwd_flag      dist
##  Length:625134      Min.   : 0.00000
##  Class :character   1st Qu.: 0.01259
##  Mode  :character   Median : 0.02122
##                     Mean   : 0.03540
##                     3rd Qu.: 0.03845
##                     Max.   :10.38500
```

```
library(lattice)
```

```
proj4string(df1) <- CRS(as.character(NA))
spplot(df1[101:200,c("vendor_id","passenger_count","dist")], "dist")
```



For the first interval (range of smallest values) in the plot legend, the lower and upper bounds are 0 and 0.037 respectively.

#Part hii

#Convert the SpatialPointsDataFrame object into an sf object and use the mapview command to plot the locations of the trips from 101 to 200 on the New York map. Use the option cex = "dist". In the obtained plot how many locations are shown at the bottom right part of the plot (close to Valley Stream)?

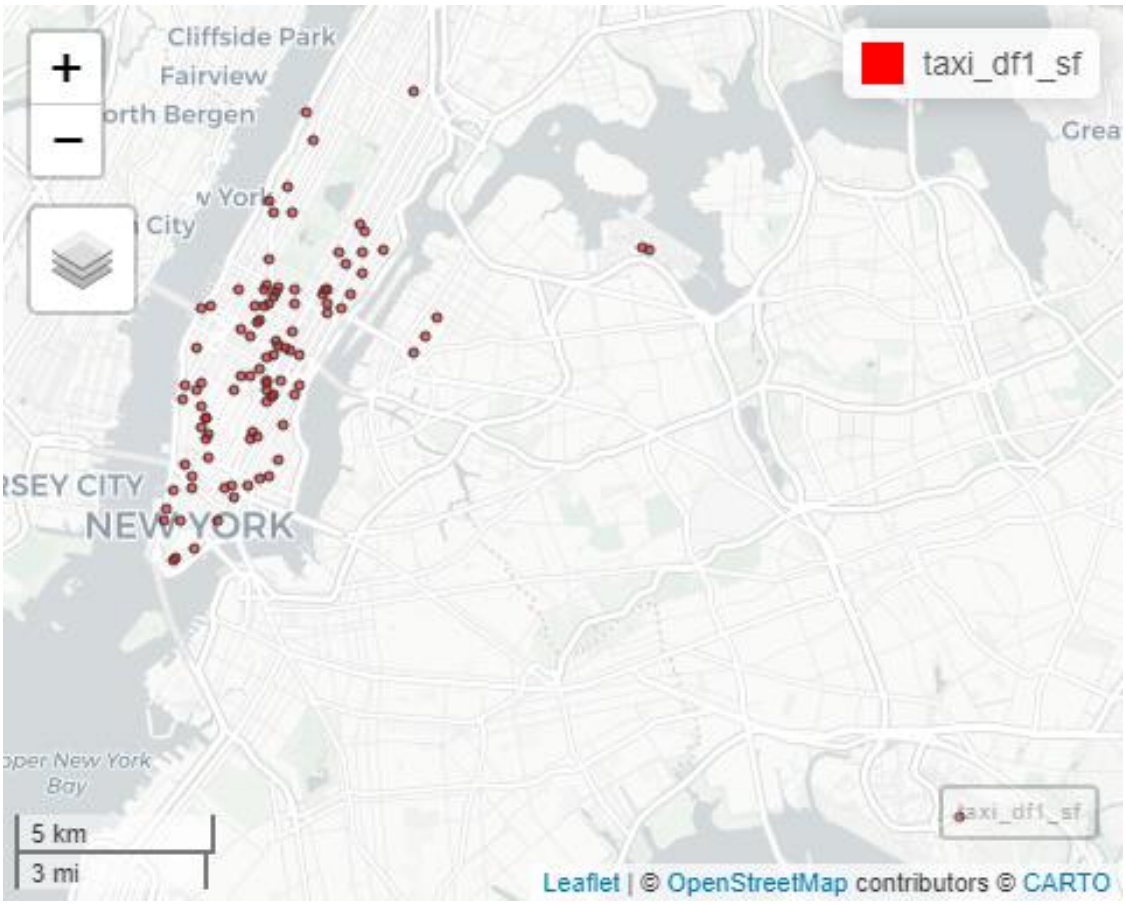
```
taxi_df <- read.csv("C:/Users/Michael Le/Desktop/Taxi_Data/taxi.csv", header
= TRUE)
taxi_df$dist <-  sqrt((taxi_df$dropoff_latitude-taxi_df$pickup_latitude)^2 +
(taxi_df$dropoff_longitude -taxi_df$pickup_longitude)^2)
str(taxi_df)

## 'data.frame':    625134 obs. of  10 variables:
##  $ id                : chr  "id3004672" "id3505355" "id1217141" "id2150126
" ...
##  $ vendor_id         : int  1 1 1 2 1 1 1 1 2 2 ...
##  $ pickup_datetime   : chr  "2016-06-30 23:59:58" "2016-06-30 23:59:53" "2
016-06-30 23:59:47" "2016-06-30 23:59:41" ...
##  $ passenger_count   : int  1 1 1 1 1 1 1 2 2 1 ...
##  $ pickup_longitude  : num  -74 -74 -74 -74 -74 ...
##  $ pickup_latitude   : num  40.7 40.7 40.7 40.8 40.8 ...
##  $ dropoff_longitude : num  -74 -74 -74 -74 -74 ...
##  $ dropoff_latitude  : num  40.8 40.7 40.7 40.7 40.8 ...
```

```
##  $ store_and_fwd_flag: chr  "N" "N" "N" "N" ...
##  $ dist              : num  0.0247 0.025 0.0139 0.0514 0.0103 ...
```

###NOTE: Pick and run only of the options ### Pick A) PICKUP LOCATIONS ### Pick B) DROPOFF LOCATIONS

#A) PICKUP LOCATIONS

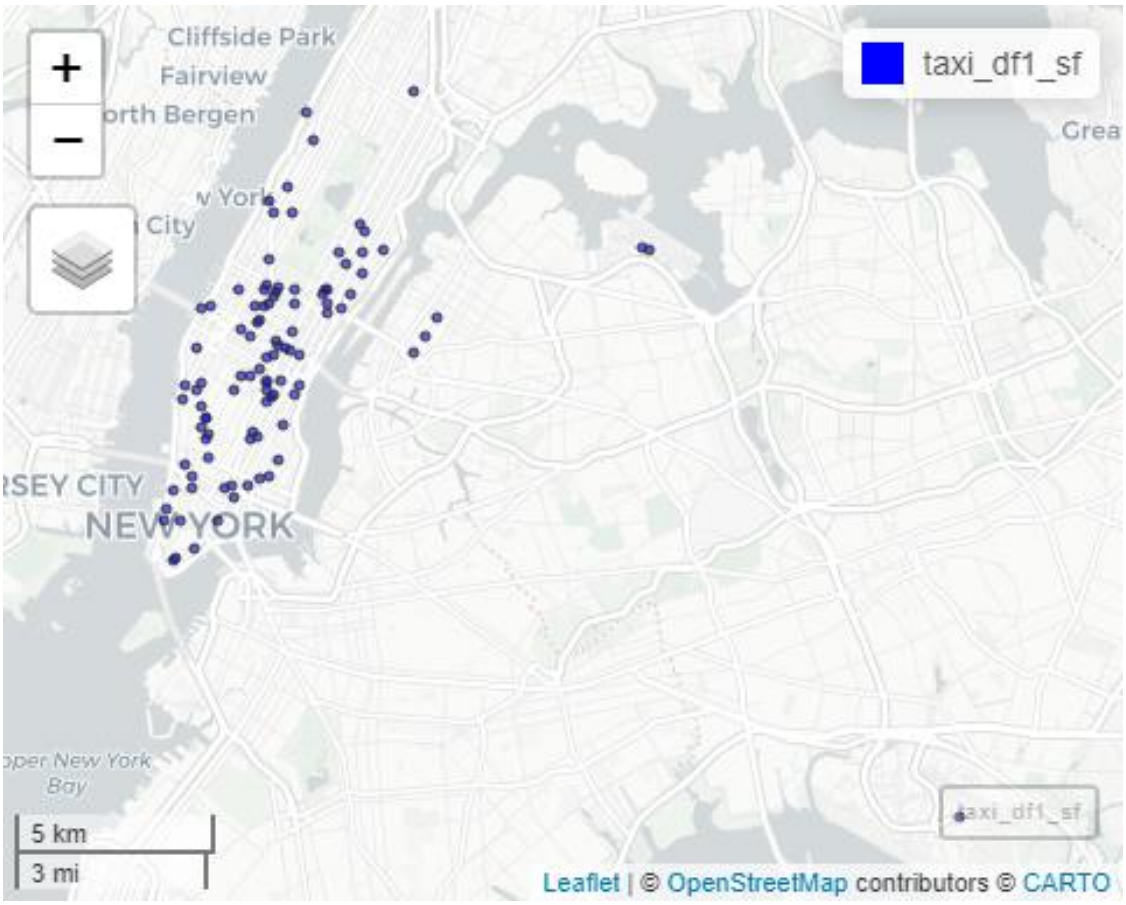
```
library(mapview)

## Warning: package 'mapview' was built under R version 4.3.3

library(sp)
library(sf)

## Warning: package 'sf' was built under R version 4.3.3

## Linking to GEOS 3.11.2, GDAL 3.8.2, PROJ 9.3.1; sf_use_s2() is TRUE

taxi_df <- taxi_df[101:200,]
taxi_df1 <- taxi_df[101:200, c("vendor_id", "passenger_count", "dist")]
taxi_coords <- cbind(taxi_df$pickup_longitude, taxi_df$pickup_latitude)
row.names(taxi_coords) <-  1:nrow(taxi_coords)
row.names(taxi_df1) <-  1:nrow(taxi_df1)
llCRS <- CRS("+proj=longlat +ellps=WGS84")
taxi_df1_sp <- SpatialPoints(taxi_coords, proj4string = llCRS)
taxi_df1_spdf <-SpatialPointsDataFrame(taxi_coords,taxi_df1,proj4string = llC
RS,match.ID = TRUE)
taxi_df1_sf <-  st_as_sf(taxi_df1_spdf)
mapview(taxi_df1_sf, col.regions = "red",cex = "dist",fgb = FALSE)

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## PhantomJS not found. You can install it with webshot::install_phantomjs().
If it is installed, please make sure the phantomjs executable can be found vi
a the PATH variable.
```



#For the pick-up locations in New York there is only one location near Valley Stream within 101 - 200 trips based on Vendor id,

B) DROPOFF LOCATIONS

```
library(mapview)
taxi_df <- taxi_df[101:200,]
taxi_df1 <- taxi_df[101:200, c("vendor_id", "passenger_count", "dist")]
taxi_coords_d <- cbind(taxi_df$dropoff_longitude, taxi_df$dropoff_latitude)
row.names(taxi_coords) <-  1:nrow(taxi_coords)
row.names(taxi_df1) <-  1:nrow(taxi_df1)
llCRS <- CRS("+proj=longlat +ellps=WGS84")
taxi_df1_sp <- SpatialPoints(taxi_coords, proj4string = llCRS)
taxi_df1_spdf <-SpatialPointsDataFrame(taxi_coords,taxi_df1,proj4string = llC
RS,match.ID = TRUE)
taxi_df1_sf <-  st_as_sf(taxi_df1_spdf)
mapview(taxi_df1_sf, col.regions = "blue",cex = "dist",fgb = FALSE)

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf
```



#For the drop-off locations in New York there is only one location near Valley Stream within 101 - 200 trips based on Vendor id,