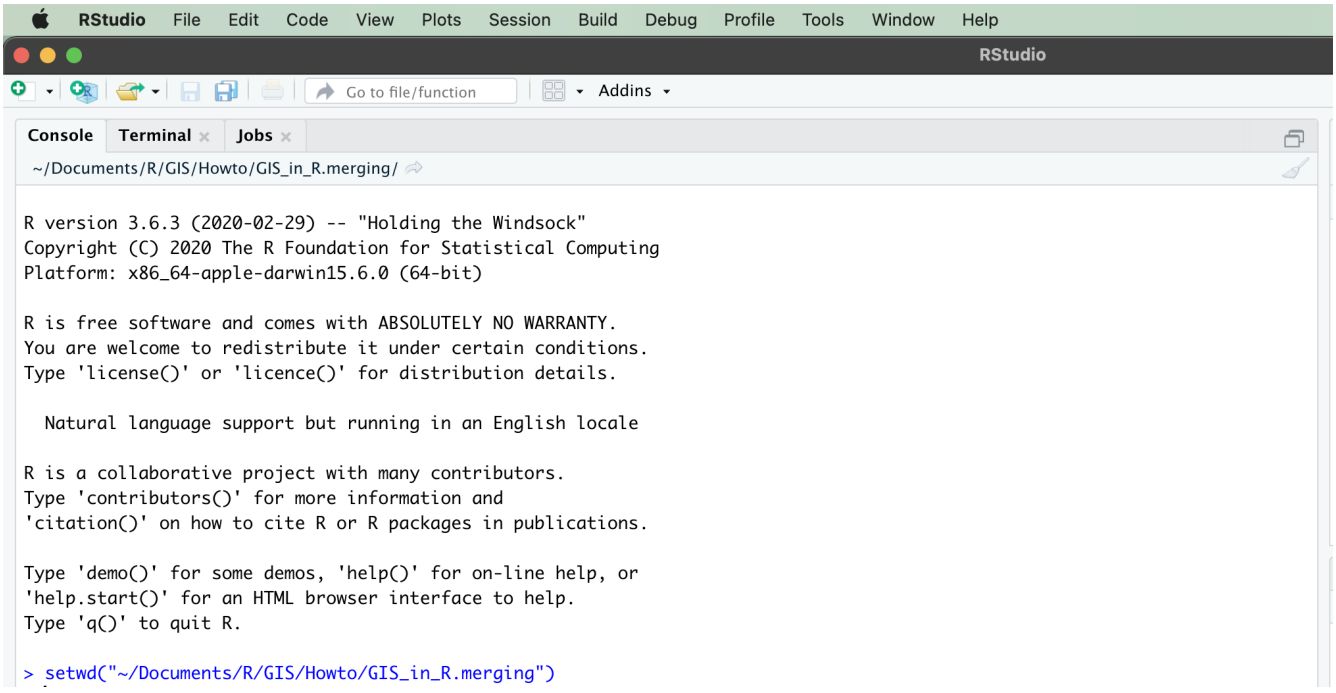
How to merge shapefile in R

GIS in R: Merging Shapefile

Merging shapefile allows you to combine shapefiles from different sources or different time periods into a single shapefile for analysis. You can use the sf package's `st_read()` function to read the shapefiles which you wish to merge into R, and then merge the spatial objects into a new spatial object containing all the features of the original shapefiles using the `rbind()` function. For example, you would like to study stops and frisks by New York city police for the reason of burglary over time in New York city. You might find shapefiles of stop, question and frisk due to burglary by NYPD for each year in your study time period from 2017 to 2019, and you'd like to combine them to view stop, question and frisk due to burglary distribution in the city. Here is the data for this exercise.

1. Set your working directory to the folder where your project locates using `setwd()`. Alternatively, you can also set the working directory using the dropdown menu **Session** in RStudio. Select the **Choose Directory** and then click the project folder. After you've set the working directory, you may verify it by calling the `getwd()` function.



2. Load the following packages in R. If you do not already have the packages installed, be sure to install them using command `install.packages()` before loading them.

```
library(sf)
library(ggplot2)
```

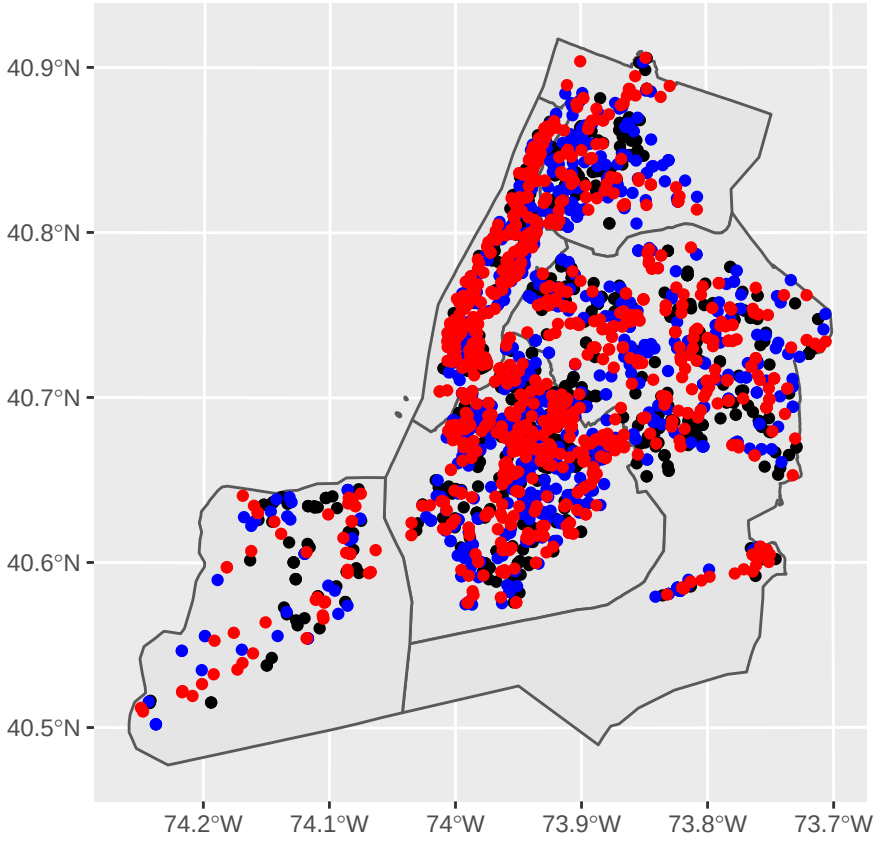
3. Use the sf package's `st_read()` function to read your shapefiles into R. With `stringsAsFactors = F` set,

character vectors will not be converted to factors.

```
NYC_boroughs <- st_read("nyu_2451_34515.shp", stringsAsFactors = F)
sqf2017_shapefile <- st_read("sqf2017_BURGLARY.shp", stringsAsFactors = F)
sqf2018_shapefile <- st_read("sqf2018_BURGLARY.shp", stringsAsFactors = F)
sqf2019_shapefile <- st_read("sqf2019_BURGLARY.shp", stringsAsFactors = F)
```

Optionally, you can use `ggplot()` from ggplot2 package to generate an overview of the different spatial objects on top of each other. In `ggplot()` adding layers is straightforward: as long as your spatial data is properly stored in sf objects, adding additional layers would simply be additional calls to `geom_sf()` in the `ggplot()` sequence. The `geom_sf()` function specifies the sf object you wish to map. For this example, we first tell `ggplot()` to draw a polygon layer "NYC_boroughs" by calling `geom_sf()`; to continue adding to the map, stop and frisk point layer for 2017 "sqf2017_shapefile" is plotted as an additional sf layer using `geom_sf()`; to keep adding stop and frisk point layers for 2018 and 2019 on the map, two additional `geom_sf()` functions are called with color arguments to set the points of 2018 to be blue and the points for 2019 to be red.

```
ggplot() + geom_sf(data = NYC_boroughs) + geom_sf(data = sqf2017_shapefile) +
  geom_sf(data = sqf2018_shapefile, color = "blue") + geom_sf(data = sqf2019_shapefile, color = "red")
```



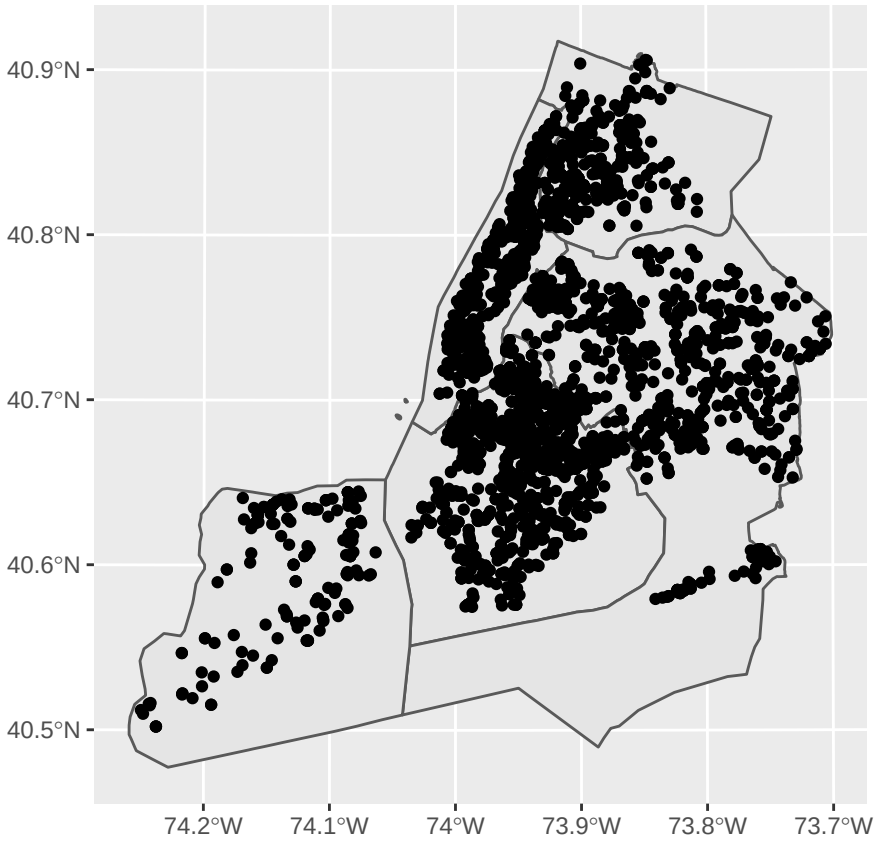
4. Use the `rbind()` function to merge multiple spatial objects into one object. Note to bind the spatial objects, they should have same number of columns and identical column names; also, they should be in the same coordinate reference system. You can use `st_crs()$input` to examine what is the coordinate reference system of each of your spatial objects. If they are not in the same coordinate system, you can reproject your spatial objects into

another coordinate system by using the `st_transform()` function. In this example, the "sqf2017_shapefile", "sqf2018_shapefile", and "sqf2019_shapefile" are all in the EPSG:2236 (http://epsg.io/2236) projection, so we can directly use `rbind()` to merge them together. For more information and resources on coordinate systems and map projections, please see Appendix 1 in NYU Data Services' QGIS tutorial, which is available here.

```
sqf2017to2019 <- rbind(sqf2017_shapefile, sqf2018_shapefile, sqf2019_shapefile)
# st_crs(sqf2017_shapefile)$input
```

Now you can use `ggplot()` to visualize the new spatial object containing all the features of the three original shapefiles. You can also use the sf package's `st_write()` function to write it as a new shapefile. The first argument is the name of the object we want to write to our disk, and the second argument is the name we want to give to the file that will be written to our disk along with its file extension (since we want the file to be written as a shapefile, we use the .shp extension). Note the shapefile will be saved in your current working directory. If you would like to save the file to other folders, you can simply add the path of the folder you wish to save before the file name ("sqf2017to2019.shp" in this example).

```
ggplot() + geom_sf(data = NYC_boroughs) + geom_sf(data = sqf2017to2019)
```




```
# st_write(sqf2017to2019, "sqf2017to2019.shp")
```