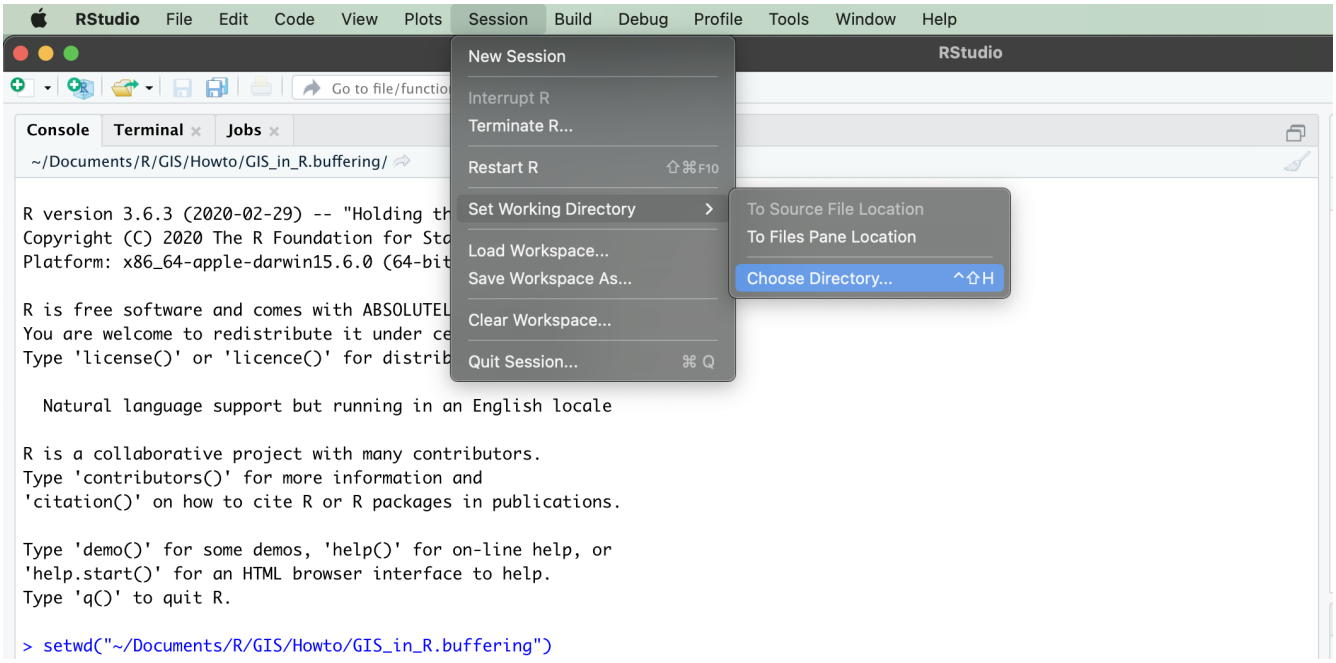
NYU DataServices

How to create buffers around shapefile features in R

GIS in R: Buffering

Buffering allows you to identify or define an area within a specified distance around a feature in order for spatial analysis or to indicate proximity or accessibility conditions. Buffering can be done on all three types of features: point, line, area. You can use the sf package's `st_read()` function to read shapefiles into R, and then convert the coordinate reference systems of the shapefiles to an appropriate one for the geography of your data by using the `st_transform()` function, and then create polygon buffers around features of interest using the `st_buffer()` function. For example, as many research have found that having limited or uncertain capacity for acquiring sufficient, safe, and nutritious food to meet dietary needs, as known as food insecurity, can result in serious consequences such as increased risks of some birth defects, anemia, cognitive problems, aggression and anxiety, etc., you would like to study how many women in New York state live within 20 minutes walking distance to fresh food sources, specifically farmers markets. You now have two shapefiles at hand, one is a New York state census tract shapefile with an attribute of women population and the other is a shapefile of locations of all the farmers markets in New York state. Also, you would like to conduct the analysis in R. 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. Optionally, you can use `View()` function to have a look at your data. For this example, by examining the data there are 4918 census tracts and 87 farmers markets in New York state.

```
NY_women_per_tract <- st_read("NY_women_per_tract.shp")
```

Reading layer `NY_women_per_tract' from data source`/Users/junhui/Documents/R/GIS/Howto/GIS_in_R.buffering/NY_wo using driver 'ESRI Shapefile' Simple feature collection with 4918 features and 14 fields geometry type: MULTIPOLYGON dimension: XY bbox: xmin: -79.76259 ymin: 40.4774 xmax: -71.77749 ymax: 45.01586 CRS: 4269

```
NY_farmers_market <- st_read("NY_farmers_market.shp")
```

Reading layer `NY_farmers_market' from data source`/Users/junhui/Documents/R/GIS/Howto/GIS_in_R.buffering/NY_farn using driver 'ESRI Shapefile' Simple feature collection with 87 features and 57 fields geometry type: POINT dimension: XY bbox: xmin: -79.32609 ymin: 40.58551 xmax: -73.56353 ymax: 44.33117 CRS: 4326

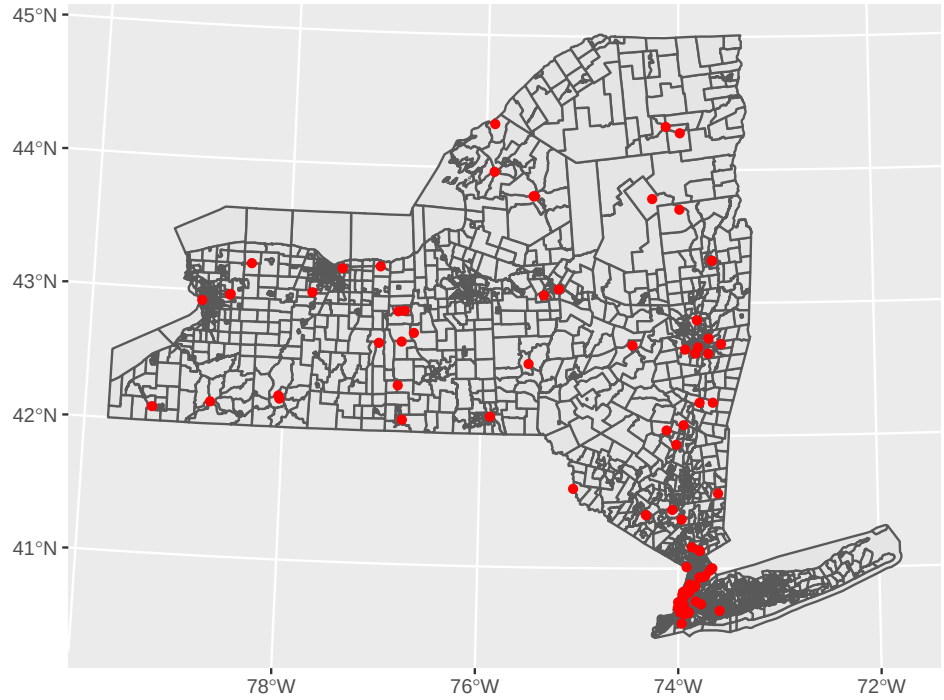
```
# View(NY_women_per_tract)
# View(NY_farmers_market)
```

4. Reproject your spatial objects into a projection appropriate for the geographical area of your data. This is because all layers involved in the analysis have to specify the same coordinate system in order to perform calculations accurately. You can change the coordinate reference systems of spatial objects from one coordinate reference system to another by using the `st_transform()` function. For this example, a bit of research suggests that EPSG:3627 would be an appropriate projection for New York state (http://epsg.io/3627). In the code, "NY_women_per_tract" and "NY_farmers_market" are the names of the objects we want to convert into another coordinate system, and 3627 is the EPSG code of the coordinate system to which we want to convert "NY_women_per_tract" and "NY_farmers_market". 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.

```
NY_women_per_tract_projected <- st_transform(NY_women_per_tract, crs = 3627)
NY_farmers_market_projected <- st_transform(NY_farmers_market, crs = 3627)
```

Optionally, you can use `ggplot()` from ggplot2 package to generate a simple 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 "NY_women_per_tract_projected" by calling `geom_sf()`; to continue adding to the map, farmers markets point layer "NY_farmers_market_projected" is plotted as an additional sf layer using `geom_sf()`.

```
ggplot() + geom_sf(data = NY_women_per_tract_projected) +
           geom_sf(data = NY_farmers_market_projected, color = "red")
```



5. Use `st_buffer()` function to create a buffer around a geometry stored in a sf object. `st_buffer()` takes in a sf object and a buffer distance and outputs a polygon with a boundary the buffer distance away from the input geometry. For this example, the geometry is points stored in the "NY_farmers_market_projected" which are the locations of farmers markets, and we create a 2km radius buffer around the farmers market points.

```
buffer_2000m <- st_buffer(NY_farmers_market_projected, 2000)
```

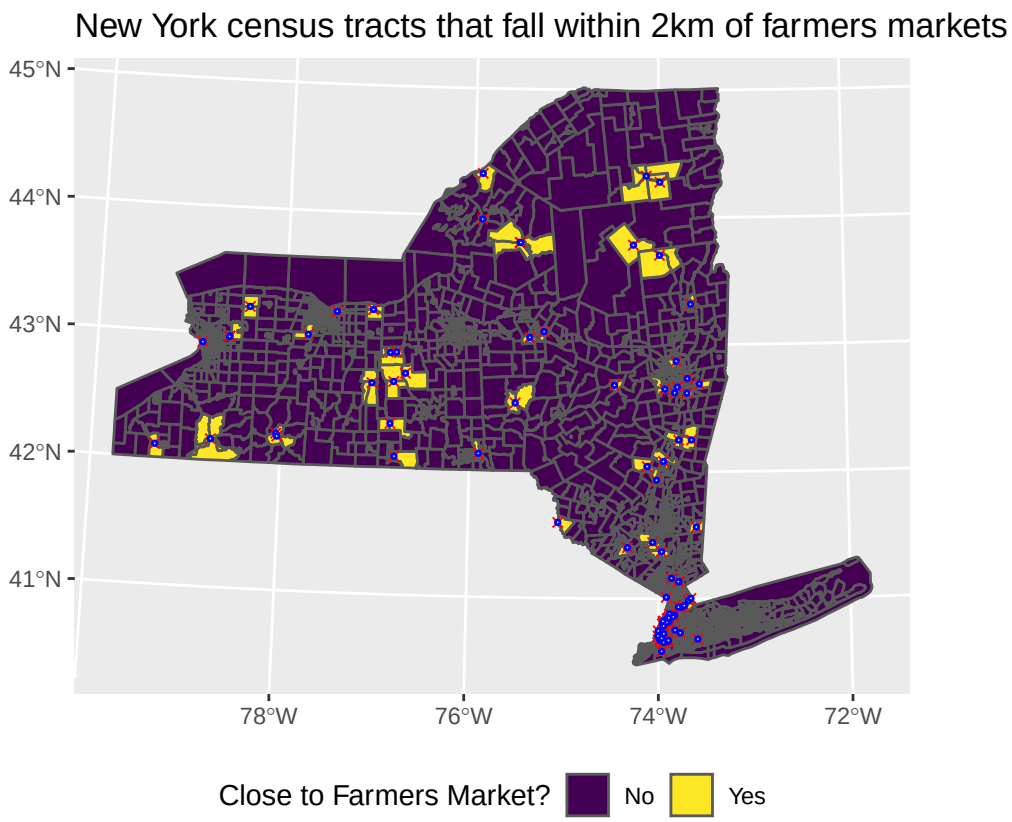
6. Now with the buffers generated, we can study the question how many women in New York state live within 20 minutes walking distance to fresh food sources through the following steps.

- Use `st_intersects()` to select all census tract polygons that intersect with the farmers market buffers. "NY_women_per_tract_projected" and "buffer_2000m" are the two polygons we want to determine the intersection, and by setting sparse = F the `st_intersects()` function returns a matrix of logical values, TRUE and FALSE, indicating if a census tract is within the 2km buffer of each of the 87 farmers markets for all 4918 census tracts in New York state, saved in a new object "tracts_access_markets".
- In the "NY_women_per_tract_projected", define a new variable "close2market" with values "Yes" for having access to fresh food sources within 20 min walking distance if a census tract intersects at least one buffer of farmers markets, and "No" for not having access to fresh food sources within 20 min walking distance if a census tract doesn't intersect with any buffers of farmers markets. In the `apply()` function below, "tracts_access_markets" is the logical value matrix we want to calculate on, and 1 indicates the calculation operates for each row in the matrix, and function(x) sum(x == FALSE) == 87 is to find which rows (tracts) have no intersection with buffers of any of the 87 farmers markets.
- The `apply()` function returns a vector of FALSE and TRUE indicating no intersection and at least one intersection with the buffers of farmers markets. They are then recoded as "No" and "Yes" by the `ifelse()` function.

```
tracts_access_markets <- st_intersects(NY_women_per_tract_projected, buffer_2000m, sparse = F)
NY_women_per_tract_projected$close2market <- ifelse(apply(tracts_access_markets, 1,
                                                  function(x) sum(x == FALSE) == 87), "No","Yes")
```

- To have a look at the buffer areas around farmers markets and the tracts that women have access to fresh food sources, we can again use `ggplot()` to do the visualization.
  - In the first `geom_sf()` call, when drawing the polygon layer "NY_women_per_tract_projected" we use different colors to distinguish tracts that have and don't have access to at least one farmers market within 20 min walking distance (aes(fill = close2market)).
  - Adding on the polygon layer using another call on the `geom_sf()`, the points of farmers markets ("NY_farmers_market_projected") are plotted by cross symbol (pch = 4) and in red color (color = "red").
  - The buffer areas around farmers markets ("buffer_2000m") are plotted on the top as blue circles (color = "blue") by adding another `geom_sf()`.
  - By calling `scale_fill_viridis_d()`, the map uses the color scales that are easier to read by those with colorblindness, and print well in gray scale.
  - We then customize the map by adding a title (ggtitle("Census tracts that fall within 2km of farmers markets")), a legend title (labs(fill = "Close to Farmers Market?")) and place the legend at the bottom (theme(legend.position = "bottom")).

```
ggplot() +
  geom_sf(data = NY_women_per_tract_projected, aes(fill = close2market)) +
  geom_sf(data = NY_farmers_market_projected, pch = 4, color = "red") +
  geom_sf(data = buffer_2000m, color = "blue") +
  scale_fill_viridis_d() +
  ggtitle("New York census tracts that fall within 2km of farmers markets") +
  theme(legend.position = "bottom") + labs(fill = "Close to Farmers Market?")
```



- Now let's calculate how many women in New York state live within 20 minutes walking distance to farmers markets. The object "idx_tracts_close2market" saves the indexes of tracts that intersect with at least one buffer of farmers markets, and then we use the indexes to calculate the number of women who live in these tracts, which is then divided by the total number of women in all 4918 tracts.

```
idx_tracts_close2market <- which(NY_women_per_tract_projected$close2market == "Yes")
women_within_20min <- sum(NY_women_per_tract_projected$WomenPop[idx_tracts_close2market])
women_within_20min/sum(NY_women_per_tract_projected$WomenPop)
```

```
## [1] 0.279542
```

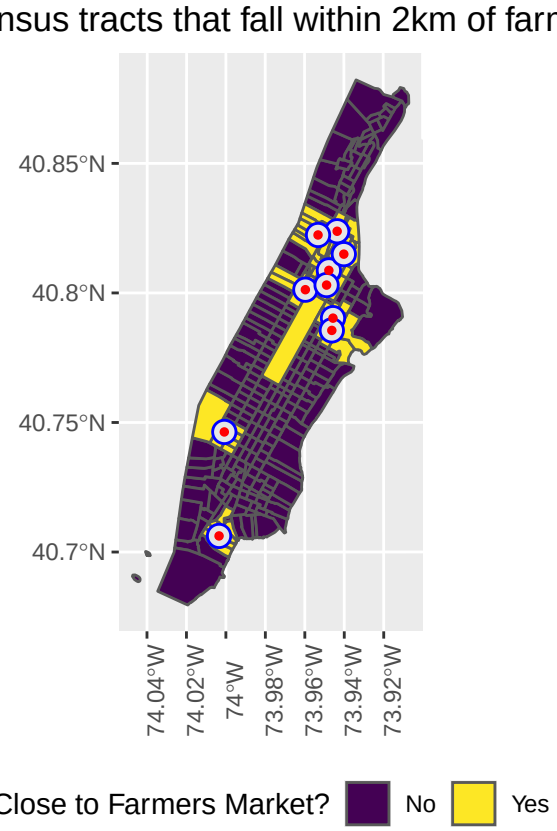
The result shows that only 27.95% women in New York state live within 20 minutes walking distance to fresh food sources.

20 minutes walking with groceries is very long and taxing, so it might be more realistic and fair to examine 5 minutes walking distance buffers around farmers markets. You may have found in the figure above that the 20 minutes walking distance buffers (blue circles) are very small in the whole state map, so we next use the same steps as before to show the 5 minutes walking distance buffers around farmers markets only in New York county/Manhattan borough.

- First use `grep()` function to select rows (tracts) whose names contain characters "New York County" to subset the "NY_women_per_tract_projected" to only census tracts in New York county.
- Then use the condition County == "New York" to subset farmers markets to those in New York county and clean spurious data that is in Brooklyn.
- Next, the procedures to do the analysis are completely same as explained above. Note that by examining the "Manhattan_farmers_market_projected", there are 10 farmers markets in Manhattan, so in the `apply()` function, sum(x == FALSE) == 10. In the `ggplot()`, we further customize the title of the figure to be in the center (plot.title = element_text(hjust = 0.5)), and the X axis labels to be vertical rotated (axis.text.x = element_text(angle = 90)).

```
# subset census tracts only in New York county
Manhattan_women_per_tract_projected <-
  NY_women_per_tract_projected[grep("New York County", NY_women_per_tract_projected$NAME_y), ]
# subset farmers markets only in New York county
Manhattan_farmers_market_projected <-
  NY_farmers_market_projected[which(NY_farmers_market_projected$County == "New York") , ]
Manhattan_farmers_market_projected <- Manhattan_farmers_market_projected[
  -which(Manhattan_farmers_market_projected$city == "Brooklyn"), ]

# create 500m buffers around farmers markets
buffer_500m <- st_buffer(Manhattan_farmers_market_projected, 500)

# determine the intersection between Manhattan census tracts and the 500m buffers
Manhattan_tracts_access_markets <-
  st_intersects(Manhattan_women_per_tract_projected, buffer_500m, sparse = F)

# generate a new variable indicating if a tract has at least one intersection with the buffers
Manhattan_women_per_tract_projected$close2market <-
  ifelse(apply(Manhattan_tracts_access_markets, 1, function(x) sum(x == FALSE) == 10),
         "No","Yes")

# plot tracts, farmers markets and their 500m buffers
```

```
ggplot() +
  geom_sf(data = Manhattan_women_per_tract_projected, aes(fill = close2market)) +
  geom_sf(data = buffer_500m, color = "blue") +
  geom_sf(data = Manhattan_farmers_market_projected, size = 1, color = "red") +
  scale_fill_viridis_d() +
  ggtitle("Manhattan census tracts that fall within 2km of farmers markets") +
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5),
        axis.text.x = element_text(angle = 90)) +
  labs(fill = "Close to Farmers Market?")
```




```
# proportion of women in NY county that live within 5 min walking distance to farmers markets
idx_Manhattan_tracts_close2market <-
  which(Manhattan_women_per_tract_projected$close2market == "Yes")
Manhattan_women_within_5min <-
  sum(Manhattan_women_per_tract_projected$WomenPop[idx_Manhattan_tracts_close2market])
Manhattan_women_within_5min/sum(Manhattan_women_per_tract_projected$WomenPop)
```

```
## [1] 0.2707443
```

The calculation result shows that only 27.07% women in New York county/Manhattan borough live within 5 minutes walking distance to fresh food sources.