Ciencia de datos geográficos

Silvia Laceiras
Felipe Sodré M. Barros

Fabián Rechberger

Usando R para acceder a datos espaciales en *PostGIS*

El paquete *sf*

Como vmos en las primeras clases, el paquete `sf` es el paquete principal para trabajarmos con datos vectoriales en `R`.

Con su función `st_read()` pudimos cargar datos que estaban en el formato *shapefile*. Se acuerdan cuando trabajamos con los datos de población?

```
library(sf)
library(tmap)
poblacion <- st_read(
  "./datos/vectoriales/Misiones_con_datos.shp")
```

Cargando datos vectoriales desde PostGIS

El paquete `sf` es tán bien estructurado que nos permite leer el dato almacenado en una base de datos de PostGIS con la misma función.

Vamos a probar?

Informaciones que vamos a necesitar:

Para acceder al dato de la base de datos, vamos a necesitar saber:

- el nombre de la base de datos donde está almacenado el dato;
- las credenciales de acceso (nombre de usuário y contraseña);
- el driver a ser usado;
- el *IP* donde se encuentra la base. Como la misma se encuentra en nuestra propia computadora, podemos usar 'localhost';

```
dsn = "PG:dbname='nyc' host='localhost' port='5432' user='postgres' passworc
```

Identificando la base de datos:

Identificando las capas disponibles:

Cargando datos vectoriales desde PostGIS

```
library(sf)
dsn = "PG:dbname='nyc' host='localhost' port='5432' user='postgres' passworc

homicios <- st_read(dsn, "nyc_homicides")
```

```
## Reading layer `nyc_homicides' from data source
##   `PG:dbname='nyc' host='172.17.0.2' port='5432' user='postgres' password='po
##   using driver `PostgreSQL'
## Simple feature collection with 3982 features and 8 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 563897.6 ymin: 4484964 xmax: 609511 ymax: 4529499
## Projected CRS: NAD83 / UTM zone 18N
```

Usando la capa cargada

```
 homicios
```

```
## Simple feature collection with 3982 features and 8 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 563897.6 ymin: 4484964 xmax: 609511 ymax: 4529499
## Projected CRS: NAD83 / UTM zone 18N
## First 10 features:
##    incident_d      boroname num_victim primary_mo id weapon light_dark year
## 1  2008-01-01      Brooklyn          1       <NA>  7    gun          D 2008
## 2  2008-01-04     Manhattan          1       <NA> 14    gun          D 2008
## 3  2008-01-05        Queens          1       <NA> 15    gun          D 2008
## 4  2008-01-04        Queens          1       <NA> 16  knife          D 2008
## 5  2008-01-05        Queens          1       <NA> 18    gun          D 2008
## 6  2008-01-07      Brooklyn          1       <NA> 20    gun          D 2008
## 7  2008-01-10     Manhattan          1       <NA> 22    gun          D 2008
## 8  2008-01-10     Manhattan          1       <NA> 23    gun          D 2008
## 9  2008-01-13 Staten Island          1       <NA> 25    gun          D 2008
## 10 2008-01-16        Queens          1       <NA> 27    gun          D 2008
##                         geom
## 1  POINT (592158.7 4502211)
```

Listando las capas espaciales almacenadas en la base de datos:

As veces necesitamos saber desdel R qué otras capas están almacenadas en la base de datos... Para eso podríamos ocupar la función `st_layers`:

```
st_layers(dsn)
```

```
## Driver: PostgreSQL
## Available layers:
##            layer_name     geometry_type features fields            crs_name
## 1   nyc_census_blocks     Multi Polygon    38794      8 NAD83 / UTM zone 18N
## 2       nyc_homicides             Point     3982      8 NAD83 / UTM zone 18N
## 3   nyc_neighborhoods     Multi Polygon      129      2 NAD83 / UTM zone 18N
## 4         nyc_streets Multi Line String    19091      4 NAD83 / UTM zone 18N
## 5 nyc_subway_stations             Point      491     14 NAD83 / UTM zone 18N
```

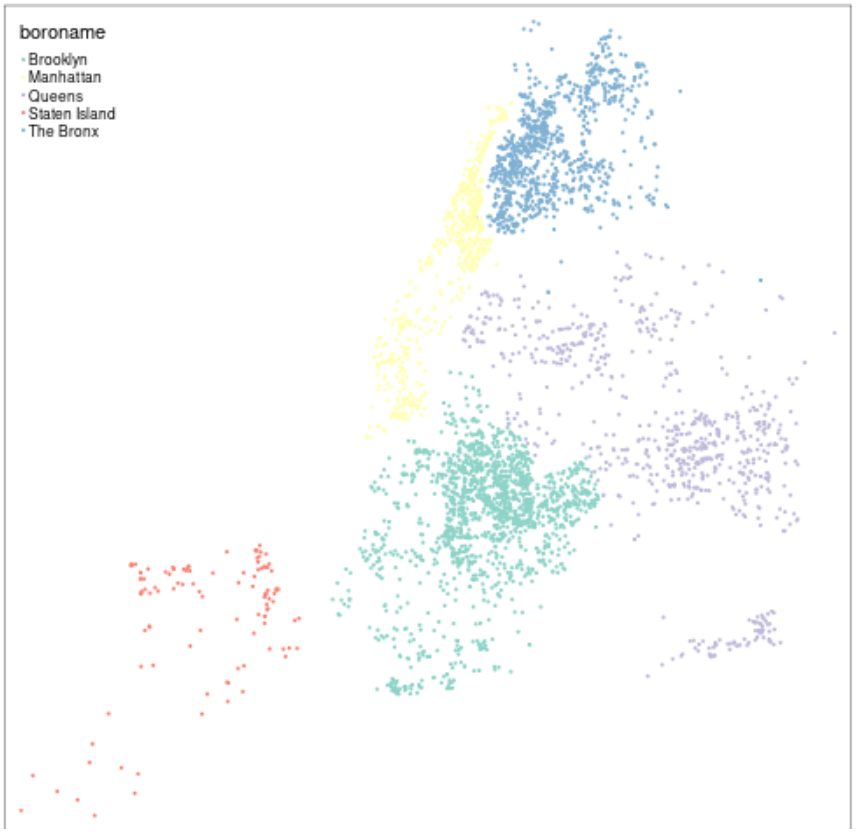
Usando la capa cargada

```
tm_shape(homicios) +
  tm_dots()
```

Como podríamos hacer un mapa de los homicidios diferenciando los barrios por color (*boroname*)?

Mapa de homicidios por barrios

```
tm_shape(homicios) +
  tm_dots(col = "boroname")
```



Podríamos hacer un mapa con el tamaño de los puntos representando la cantidad de victimas (*num_victim*)?

Cambiando el tipo de dato

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts()
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all cor
```

```
homicios <-
  homicios %>%
  mutate(num_victim =
           as.integer(num_victim))
```

Mapa con puntos representando cantidad de victimas

```
tm_shape(homicios) +
  tm_dots(size = "num_victim")
```

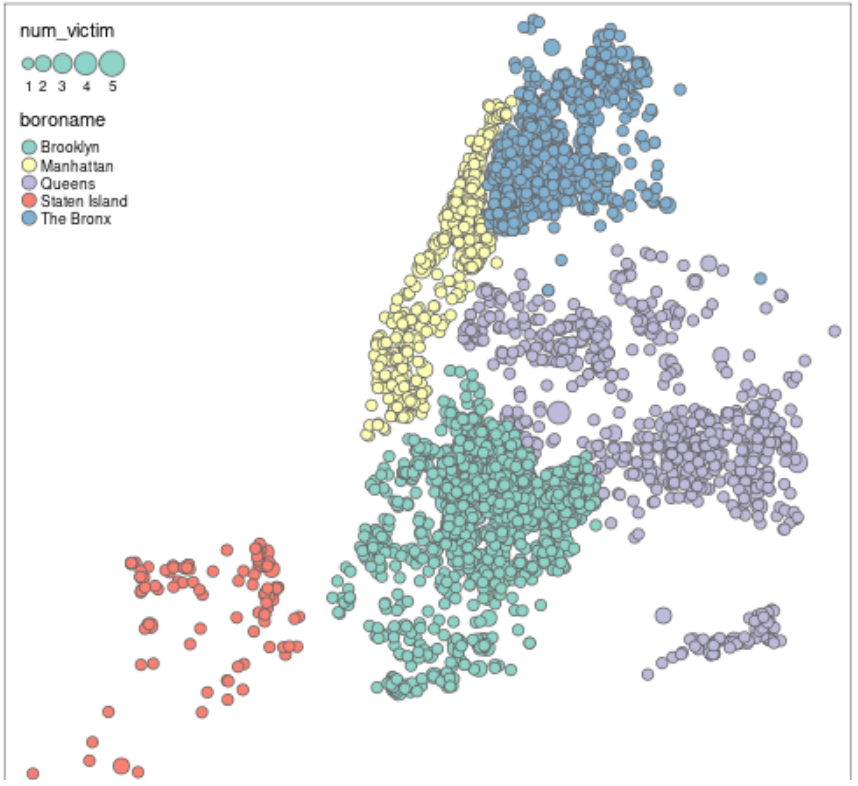
Probando *tm_bubbles*

```
tm_shape(homicios) +
  tm_bubbles(size = "num_victim")
```

Cómo podríamos representar la cantidad de victimas por tamaño de los puntos y los barrios por color?

Victimas y barrios

```
tm_shape(homicios) +
  tm_bubbles(size = "num_victim",
             col = "boroname")
```



Victimas y barrios en mapas distintos

```
tm_shape(homicios) +
  tm_bubbles(size = "num_victim") +
  tm_facets(by = "boroname")
```