



Ciencia de datos geograficos

Silvia Laceiras

Felipe Sodré M. Barros

Fabián Rechberger

0. Instalando R y RStudio

1. Organizando carpeta del proyecto

2. Por qué R?

3. R y RStudio

4. Paquetes/Librerías

...

X. Materiales de consulta



0. Instalando R y RStudio



1. Organizando carpeta del proyecto



2. Por qué R?

2. Por qué R?

1. Es **grátis**!
2. Es amigable ("**user-friendly**")!
3. Permite automatizar rutinas (tareas que se repiten)!
4. Nos permite hacer **graficos** de forma eficiente
5. Es considerada una de las **principales lenguajes** para nálsisi de datos

[*] <https://insights.stackoverflow.com/survey/2019#technology>

2. Por qué R?

~~Qué puedo hacer con R?~~

Cómo hacer XXXX en R?



3. R y RStudio

Aclaración importante

3. R y RStudio

3. Entendiendo el RStudio

3. R y RStudio

Ambiente de desarrollo integrado (IDE)

3. R y RStudio

3. R y RStudio

4. Paquetes/Librerías

4. Paquetes/Librerías

Paquetes/Librerías son **colecciones de funciones** creadas para realizar análisis específicos o manejar distintos tipos de datos.

4. Paquetes/Librerías

Instalando nuestro primero paquete

Instalando el paquete geoAr

```
install.packages("geoAr")
```

Cargando el paquete

```
library(geoAr)  
library(sf) # vamos a ocupar el paquete sf tambien
```

[*] <https://github.com/PoliticaArgentina/geoAr>

Instalando nuestro primero paquete

Descargando datos geográficos de Argentina directo del R

```
# descarga de datos
```

```
departamentos <- get_geo(geo="MISIONES", level = "departamento")  
departamentos
```

```
## Simple feature collection with 17 features and 2 fields
```

```
## Geometry type: MULTIPOLYGON
```

```
## Dimension: XY
```

```
## Bounding box: xmin: -56.05506 ymin: -28.15997 xmax: -53.64276 ymax: -25.4984
```

```
## Geodetic CRS: WGS 84
```

```
## # A tibble: 17 × 3
```

```
##   codprov_censo coddepto_censo                                geome
```

```
## * <chr>          <chr>                                <MULTIPOLYGON [
```

```
## 1 54              007      (((-55.49358 -27.70632, -55.5005 -27.76028, -
```

```
## 2 54              014      (((-54.58086 -27.01227, -54.54501 -27.08789,
```

```
## 3 54              021      (((-55.56439 -27.29871, -55.53692 -27.29375,
```

```
## 4 54              028      (((-55.81289 -27.46097, -55.78509 -27.49415,
```

```
## 5 54              035      (((-55.31861 -27.91788, -55.33642 -27.93525,
```

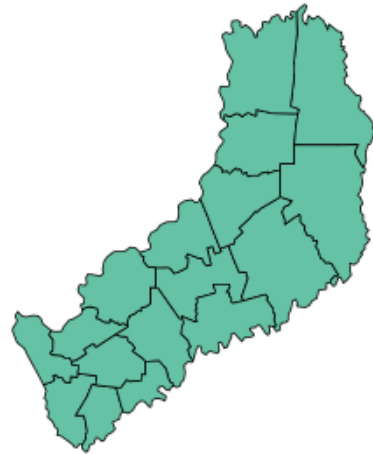
La estructura del dato geográfico

Instalando nuestro primero paquete

creando un grafico (mapa?) simple

```
# mapa de los departamentos  
plot(departamentos)
```

codprov_censo



coddepto_censo



Entendiendo la representación [geo]gráfica

Como se trata de un dato con geometría que posee atributos asociados, el R representa graficamente las geometrías con todos los atributos existentes, generando x 'mapas'.

Por eso, mucho cuidado al usar datos geográficos con muchas columnas en la tabla de atributos!

creando un grafico (mapa?) sin representar los atributos:

```
plot(st_geometry(departamentos))
```





Elaborando mapas

Elaborando mapas

Elaborando mapas

<https://www.jstatsoft.org/article/view/v084i06>



Instalando el tmap

Instalando el tmap

El clásico:

```
install.packages("tmap")
```



Usando el tmap (sintaxis)

Usando el tmap (sintaxis)

El `tmap`, bien como el `ggplot2` (principal librería para gráficos en R), está basado en el abordaje "gramatica de gráficos" ('grammar of graphics', Wilkinson and Wills 2005).

Qué quiere decir eso?

- Esta basado en capas (`layers`)
- Separa el dato de entrada de las configuraciones estéticas (definición del cómo el mismo será presentado);
- El elemento básico es la función `tm_shape()`, en el cual se especifica el dato de entrada;
- El `tm_shape` debe ser seguido por un elemento que define la estetica, como `tm_polygons()` (para representar polígonos) y/o `tm_dots()` (para datos de punto), entre [tantos] otros;

```
# representando el dato de departamentos con tmap  
tm_shape(departamentos) +  
  tm_polygons()
```

Usando el tmap (sintaxis)

Atención! No confundir el término `shape` de la función del `tmap` con `shapefile`.
`tm_shape` será usado mismo si el dato de entrada fuera un raster!

```
tm_shape(raster_entrada) +  
  tm_raster()
```



Manos a la obra

Población de Misiones

Vamos a trabajar con los datos de población del censo.

Cargando el dato de censo

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

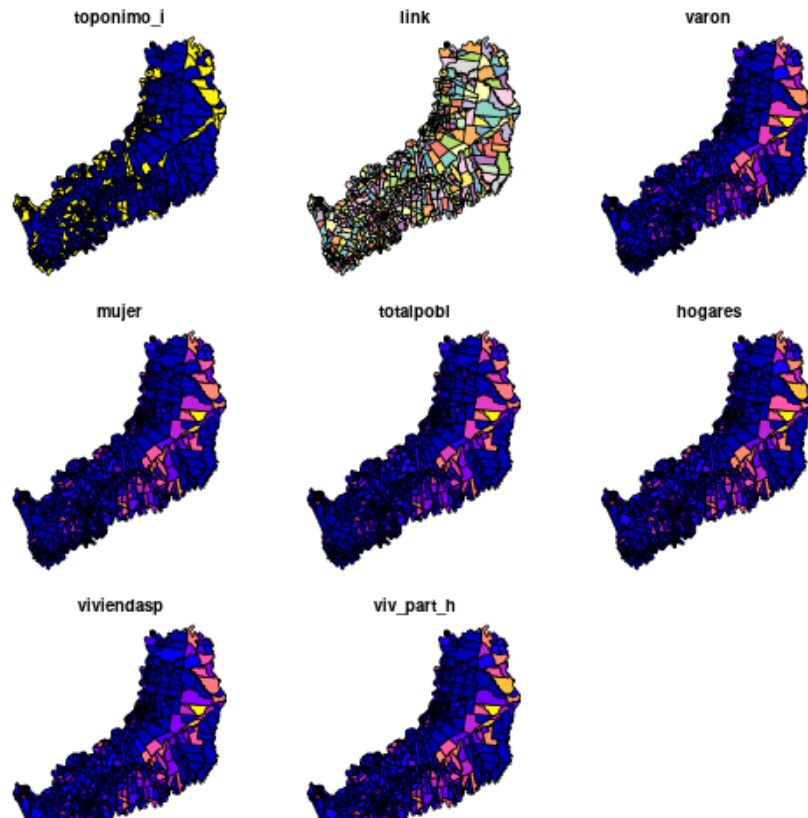
## Reading layer `Misiones_con_datos' from data source
##   `/mnt/Trabalho/repos/UNaM_GeoDataScience/Clase II/datos/vectoriales/Misione
##   using driver `ESRI Shapefile'
## Simple feature collection with 1432 features and 8 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:   xmin: 4486130 ymin: 6841372 xmax: 4740998 ymax: 7126808
## Projected CRS: POSGAR 94 / Argentina 3

## Simple feature collection with 1432 features and 8 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:   xmin: 4486130 ymin: 6841372 xmax: 4740998 ymax: 7126808
## Projected CRS: POSGAR 94 / Argentina 3
```

Población de Misiones

Inspeccionando la representación geográfica de los atributos

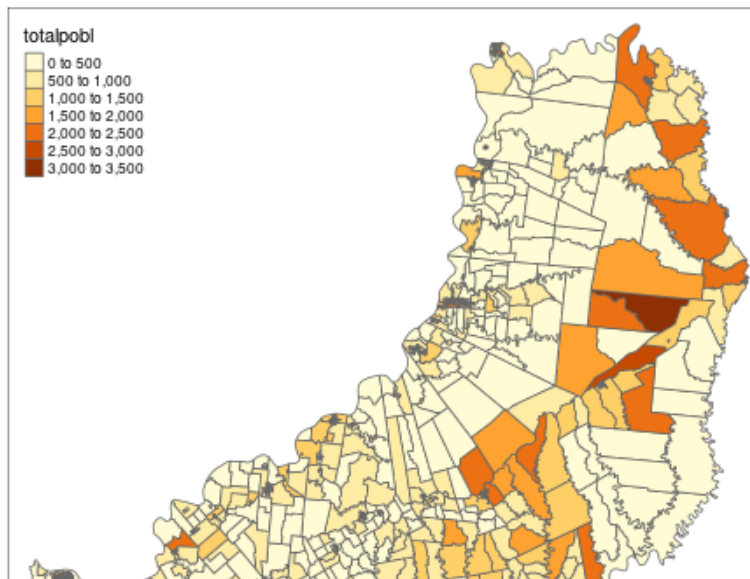
```
plot(poblacion)
```



Elaborando un mapa con `tmap` de la Población de Misiones

En este mapa queremos representar solamente la geometría con los colores representando los valores presentes en el campo `totalpobl`:

```
tm_shape(poblacion) +  
  tm_polygons(col = 'totalpobl')
```

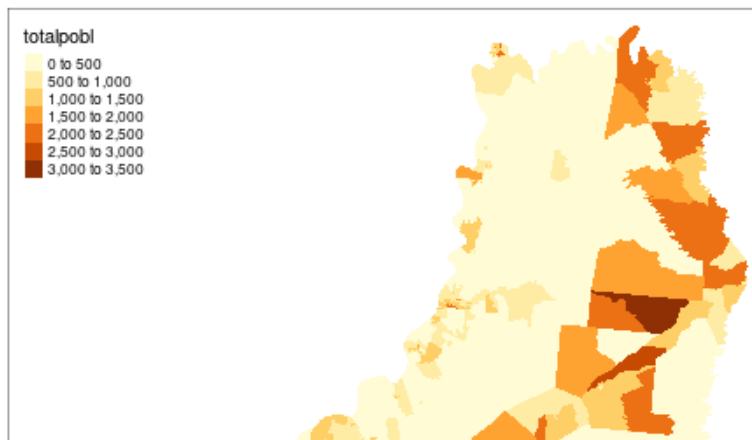


Elaborando un mapa con `tmap` de la Población de Misiones

Como los radios censal suelen tener tamaños variados y muchos de ellos, chiquitos, la línea de perímetro del polígono suele dificultar la visualización/interpretación...

Por eso, podríamos cambiar los parámetros de `tm_polygons`:

```
tm_shape(poblacion) +  
  tm_polygons(col = 'totalpobl', border.col = NULL)
```



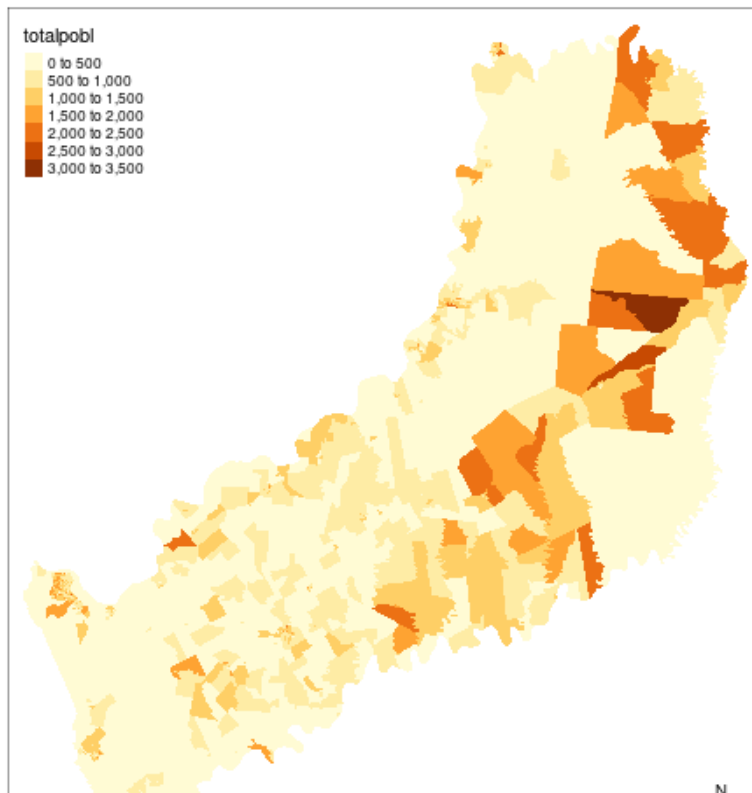
tmap

Pero nos faltan los elementos básicos del mapa:

1. Flecha de norte;
2. Coordenadas de referencia del mapa (grilla);
3. Barra de escala;

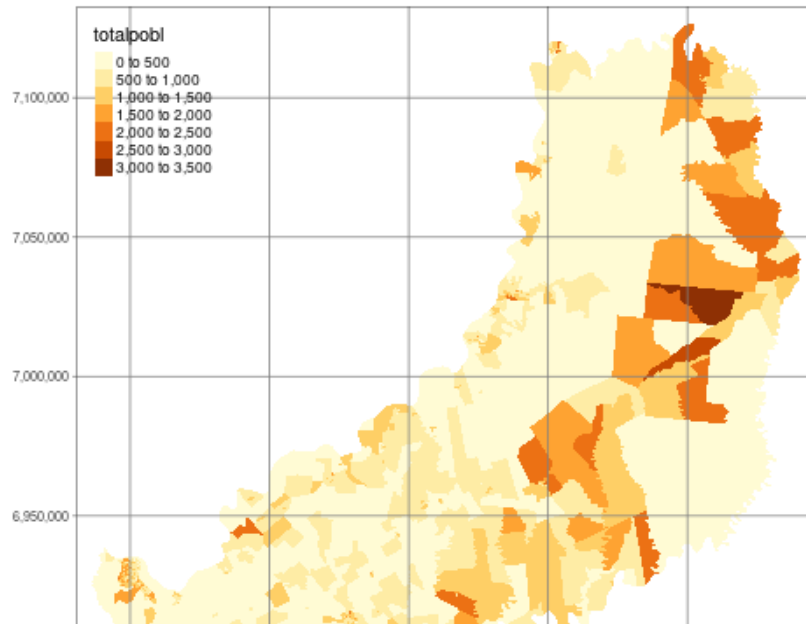
Flecha de norte

```
tm_shape(poblacion) +  
  tm_polygons(col = 'totalpobl', border.col = NULL) +  
  tm_compass()
```



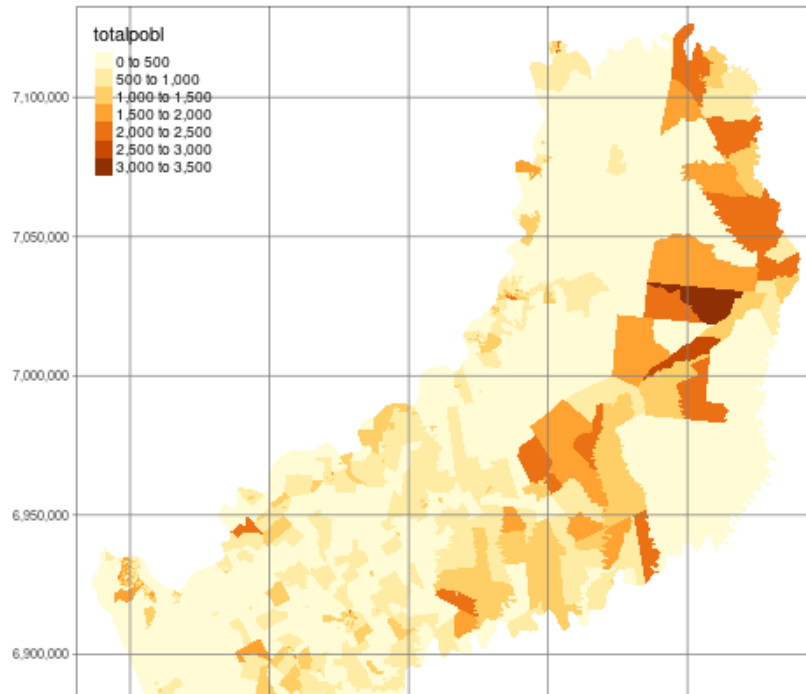
Coordenadas de referencia del mapa (grilla)

```
tm_shape(poblacion) +  
  tm_fill(col = 'totalpobl') +  
  tm_compass() +  
  tm_grid()
```



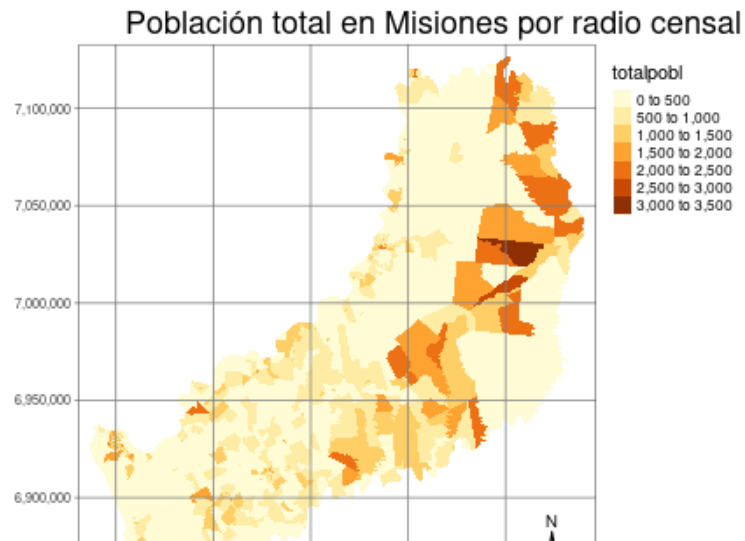
Barra de escala

```
tm_shape(poblacion) +  
  tm_fill(col = 'totalpobl') +  
  tm_compass() +  
  tm_grid()+  
  tm_scale_bar()
```



Titulo

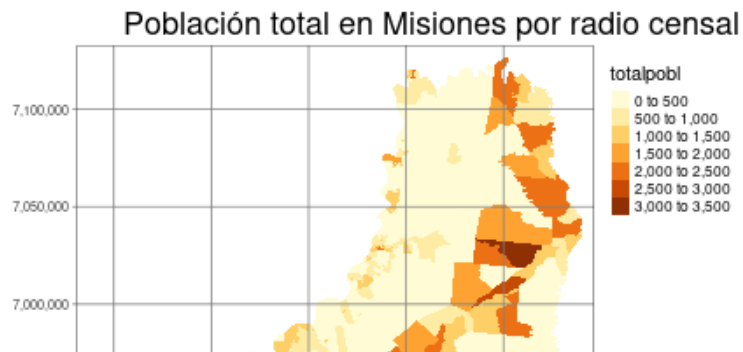
```
tm_shape(poblacion) +  
  tm_fill(col = 'totalpobl') +  
  tm_compass() +  
  tm_grid()+  
  tm_scale_bar() +  
  tm_layout(main.title = "Población total en Misiones por radio censal ", le
```



Guardando el mapa como objeto R

Podemos crear un objeto que almacena el mapa e con eso facilitar la configuración del mismo.

```
mapa <- tm_shape(poblacion) +  
  tm_fill(col = 'totalpobl') +  
  tm_compass() +  
  tm_grid()+  
  tm_scale_bar() +  
  tm_layout(main.title = "Población total en Misiones por radio censal ", le  
mapa
```



Usando estilo pre definidos

tmap_style

- `tm_style("bw")`
- `tm_style("classic")`
- `tm_style("cobalt")`
- `tm_style("col_blind")`
- `tm_style("natural")`

`mapa + tm_style("bw")`

`## Note that tm_style("bw") resets all options set with tm_layout, tm_view, tm_f`



Guardando el mapa

```
tmap::tmap_save(mapa, filename = "mapa_poblacion_misiones.png")
```

tmap_mode()

Otra funcionalidad interesante del tmap es permitir visualizar el mapa en formato webmap. Para eso, basta alterar el parámetro del `tmap_mode()`, para `view`.

```
tmap_mode("view")  
mapa
```

Guardando en un html

El `tmap` permite también guardar el mapa en formato webmap. Basta usar `tmap_save` y guardarlo como `.html`:

```
tmap::tmap_save(filename = "mapa_población_misiones.html")
```



Materiales para estudio

Materiales para estudio

Análise espacial com R (2019)

[*] <https://www.dropbox.com/s/blgtp2bmpdghol7/AnaliseEspacialComR.pdf>

Materiales para estudio

Geocomputation with R (2019)

[*] <https://geocompr.robinlovelace.net/>

Materiales para estudio

Geospatial Health Data (2019)

[*] <http://www.paulamoraga.com/book-geospatial/>