



Laboratorio Nacional de Políticas Públicas



CENTRO DE INVESTIGACIÓN  
Y DOCENCIA ECONÓMICAS A.C.

# Mapas - ggplot2

## Periodismo de Datos PdD - 2021

### Febrero, 2021

**M.C. JORGE JUVENAL CAMPOS FERREIRA.**

Investigador Asociado.

Laboratorio Nacional de Políticas Públicas

CIDE

# **Hoja de Ruta.**



- 1. Revisión de conceptos básicos.**
- 2. Revisar las principales fuentes de datos geográficos (caso mexicano).**
- 3. Hacer ejemplo práctico.**
- 4. Actividad práctica (haciendo un mapa en equipos)**

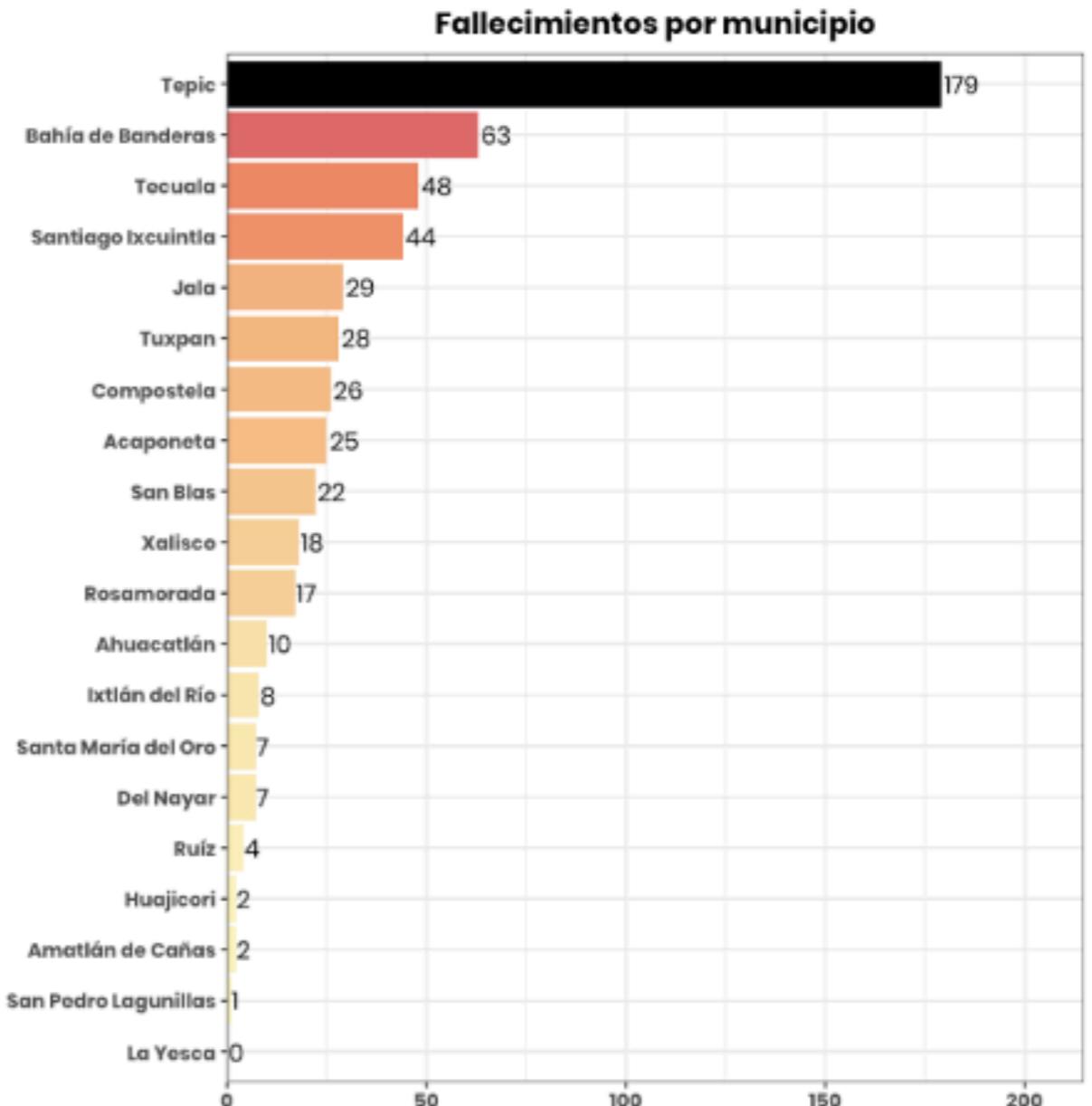
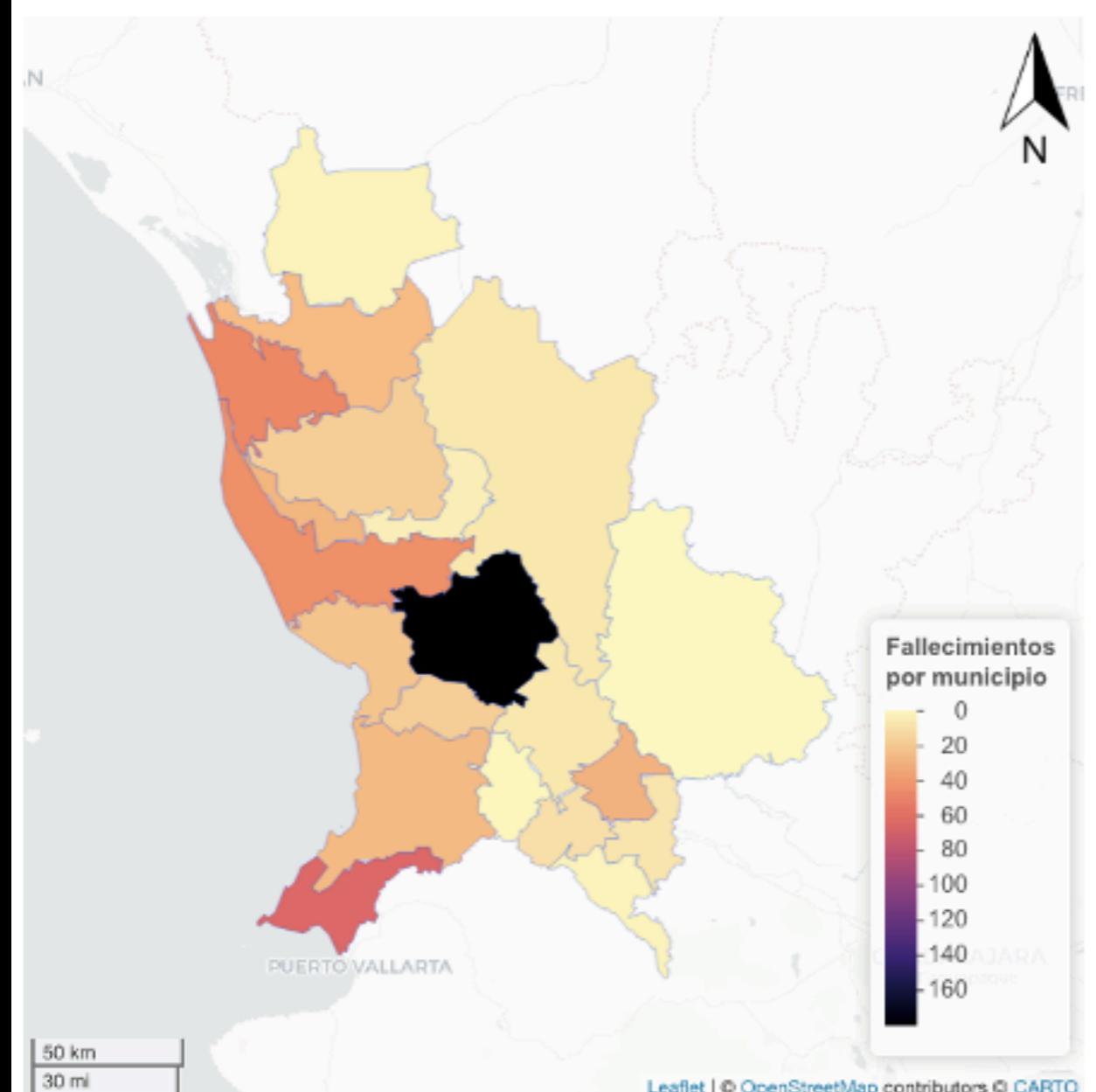
# **Conceptos Básicos**

# Mapa

- Un mapa es una *representación gráfica simplificada de un territorio con propiedades métricas sobre una superficie bidimensional (Wikipedia, 2020)*.
- Un mapa sirve para poder visualizar información que varía o se distribuye a lo largo del espacio.

**Si bien un mapa es una muy buena alternativa para visualizar algún fenómeno, siempre hay que tener en cuenta el propósito al elaborar uno de estos.**

# Ejemplo de uso de mapas



Fuente: Elaboración propia con datos de la Secretaría de Salud, 24 de Agosto del 2020.

# Información Geográfica

- La información geográfica es aquella información que tiene algún componente espacial, es decir, una ubicación, y además, una información atributiva que nos detalle más sobre ese elemento en cuestión.

Info geográfica: **componente espacial + atributo(s)**



- Puntos
- Líneas
- Polígonos
- Áreas
- Matrices de pixeles



- Nombres de calles
- Estadísticas de un lugar
- Precios
- Población
- Índices

# Tipos de Información Geográfica

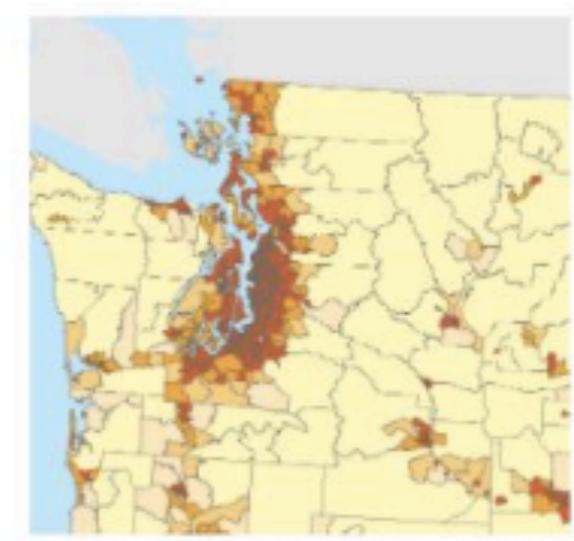
## Información Vectorial



Puntos

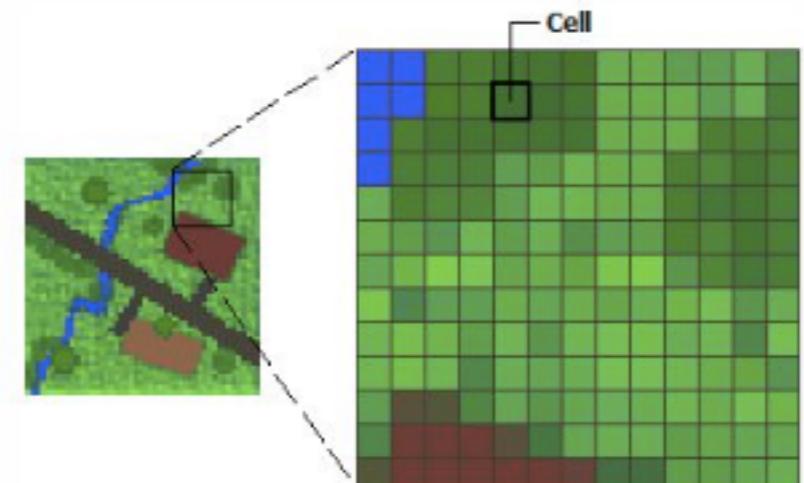


Líneas



Polígonos

## Información Raster

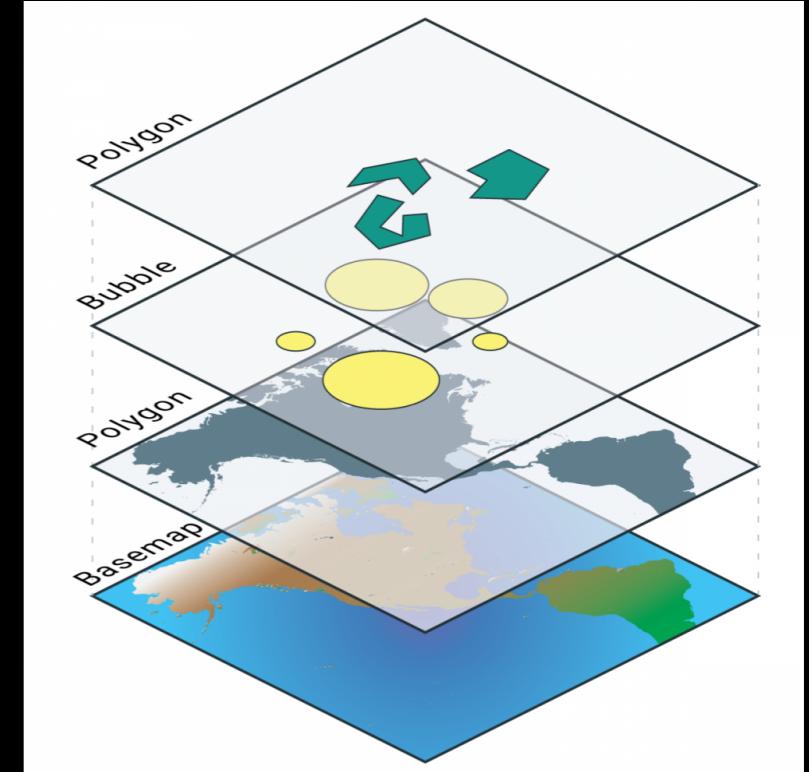
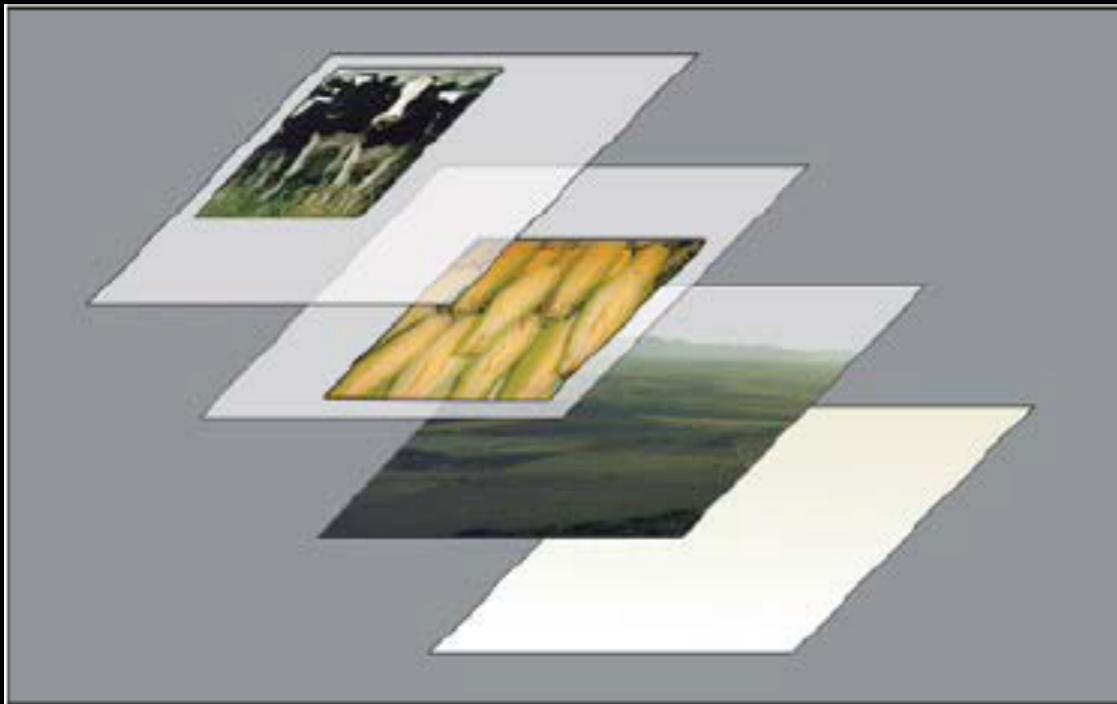


Píxeles (Celdas)

# Capas

Las capas son una **herramienta de trabajo que se asemeja a las hojas apiladas de acetato**.

Las capas **se modifican de manera individual, y al final se juntan** (como un acetato) de tal forma que generen una visualización final.

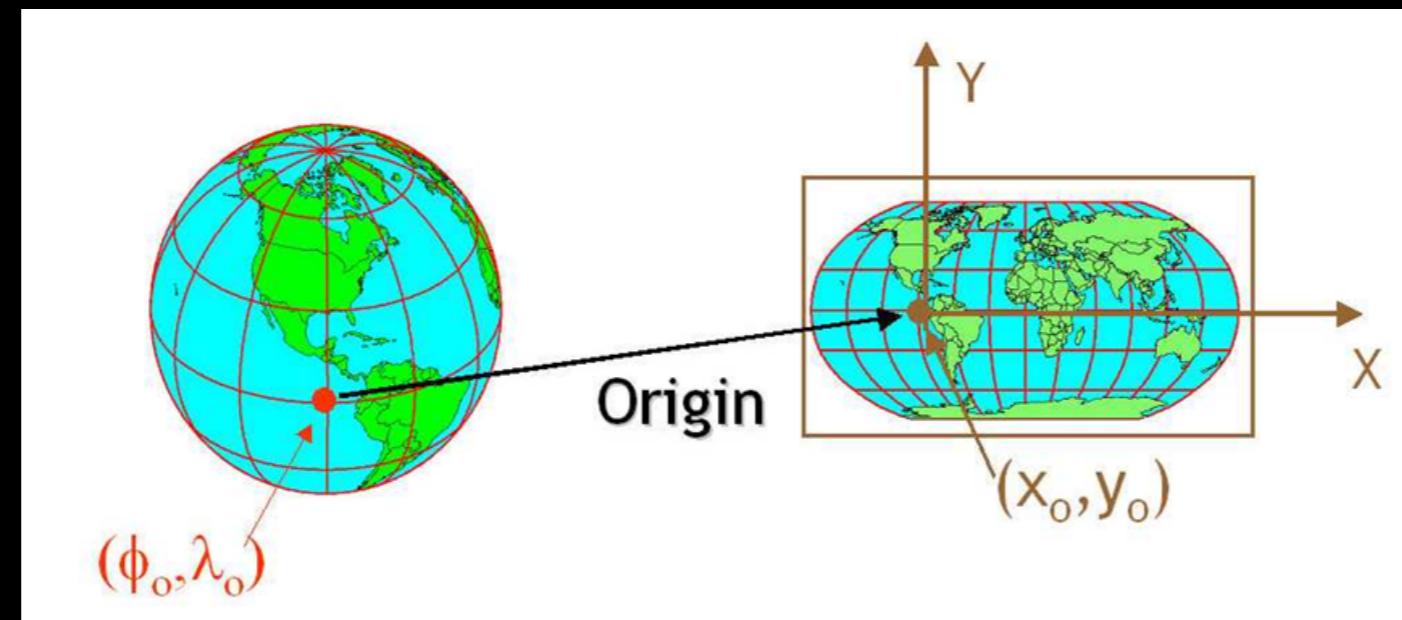


# CRS (Sistema de Coordenadas de Referencia)

Un CRS es el **sistema de coordenadas que se utiliza para localizar** las entidades geográficas.

A grandes rasgos, un crs debe incluir: **a) una proyección geográfica, b) un punto de referencia c) un sistema de traducción a otro crs y d) un datum o elipsoide de referencia.**

**En esta clase utilizaremos el crs 4326** (WGS84 proyección lat/long) por ser el más común y el que utilizan por default librerías como leaflet y google maps.



# Archivos de almacenamiento de info geográfica

**La información geográfica se almacena en archivos especiales.** Los más comunes son los siguientes:



## **ESRI Shapefile.**

Es el mas común, diseñado para funcionar con ArcGIS y QGis. Está conformado por 5 archivos diferentes que almacenan geometrías, atributos y metadatos.



## **{JSON, GeoJSON, TopoJSON}**

Uno de los más populares, actualmente entre gente que usa R o Python. Es ligero, fácil de almacenar y de acceder desde la web.



## **Google \*.kml\*.kmz**

Es el formato diseñado por Google para trabajar utilizando las herramientas de Google Maps.

# Archivos de almacenamiento de info geográfica

**La información geográfica se almacena en archivos especiales.** Los más comunes son los siguientes:



**Una hoja de excel con coordenadas (.csv, .xlsx)**

Una hoja de excel también puede almacenar información geográfica, especialmente si almacena coordenadas de puntos o ubicaciones.



**Imagen .TIFF, .GEOTIFF**  
Formato más utilizado para almacenar bases de datos Raster e imágenes satelitales. En términos prácticos, es una imagen grande y georreferenciada.

Las claves Geoestadísticas de INEGI son números que identifican a un estado, municipio, AGEB, comunidad, colonia o manzana en particular.

Estas claves funcionan como el **Id** de un polígono, y son muy importantes para utilizarlas como variables “Llave” para pegar bases de datos de atributos con bases de datos geográficas.

	CVEGEO	CVE_ENT	CVE_MUN	NOM_MUN
1	01001	01	001	Aguascalientes
2	01002	01	002	Asientos
3	01003	01	003	Calvillo
4	01004	01	004	Cosío

# Fuentes de Información

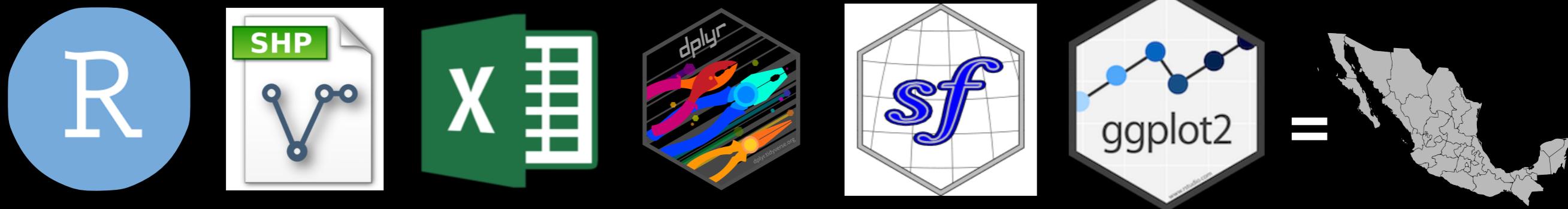
# Actividad 1

- Vamos a **dividir el grupo en dos equipos para explorar las fuentes de información** del Archivo 1 de la carpeta “Actividades”.
- Cada equipo va a revisar las fuentes de información geográfica de cada archivo, y tienen que **responder las siguientes preguntas:**
  1. ¿Qué página visitaron?
  2. ¿Qué bases de datos observaron?
  3. ¿En qué formatos estaban?
  4. ¿Qué bases se le hicieron interesantes?
  5. ¿Qué mapa se les ocurre hacer con estas bases?

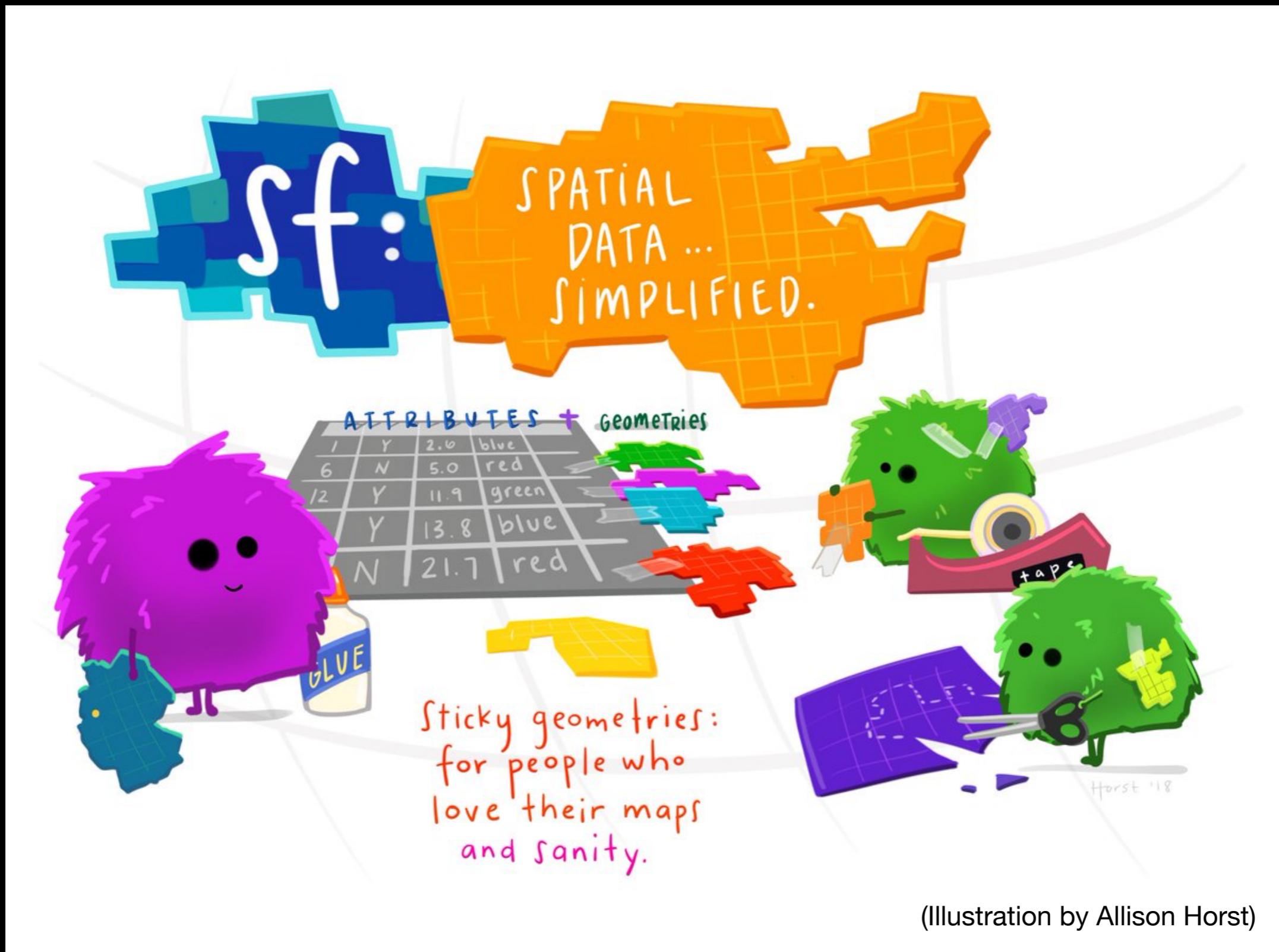
# Actividad práctica

# Haciendo mapas en R

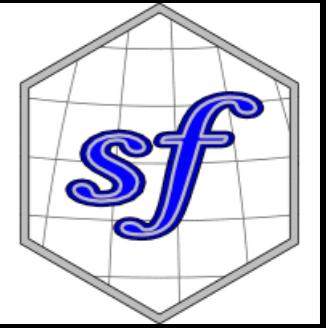
- Para hacer mapas en R, necesitamos lo siguiente:
  1. Un **archivo que almacene la información geográfica** (lo que vimos en la anterior actividad)
  2. Una **librería para abrir ese archivo {sf}**
  3. Una **base de atributos** (una tabla de datos)
  4. Una **librería para manejar esos atributos {dplyr,tidyverse}**
  5. Una **librería para visualizar esa información {ggplot, leaflet}**



# Librería {sf}

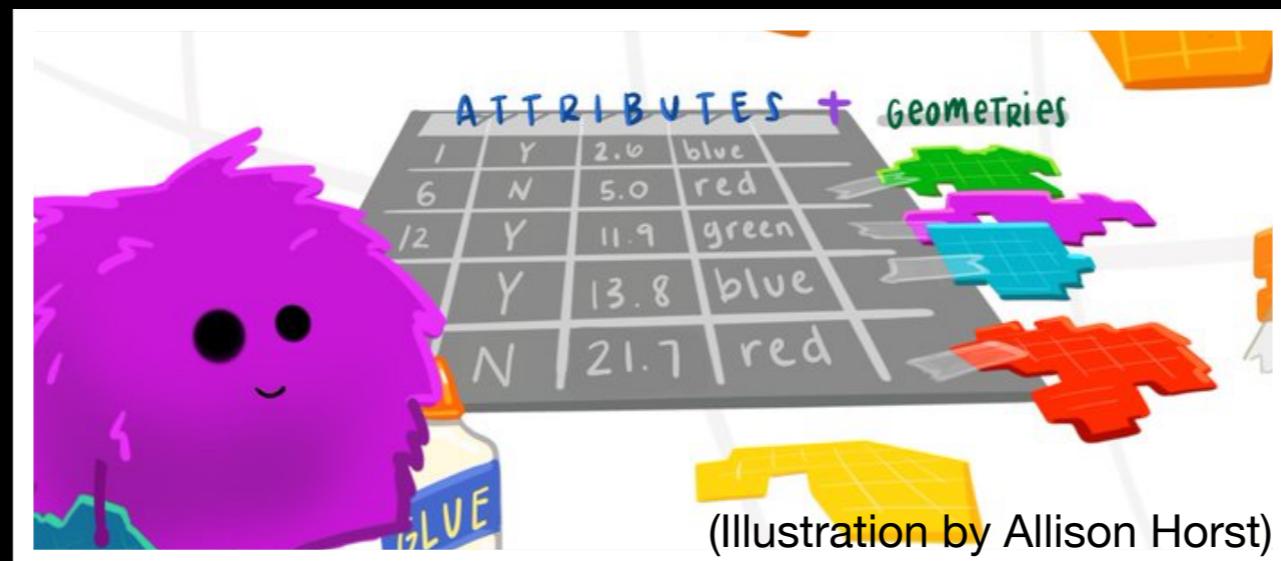


# Librería {sf}



Con la librería sf los objetos espaciales son **almacenados como un data frame simple** con **una columna que contiene la información de las coordenadas de las geometrías. (Objeto clase sf)**

Esa columna especial es una lista de la misma longitud que el número de filas en el data frame.



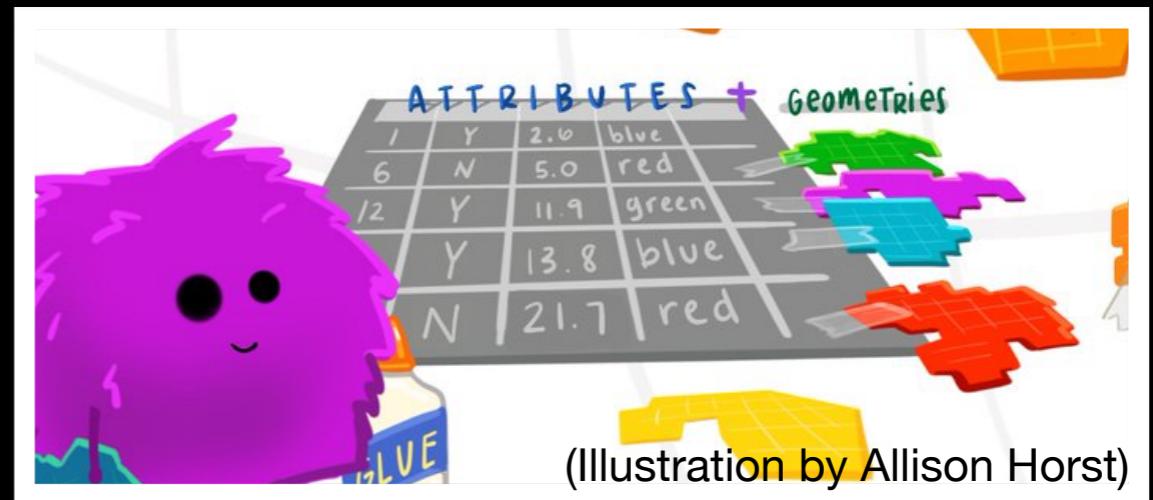
# 1. Función `sf::st_read()`



La función `st_read()` lee un archivo o base de datos vectorial como un objeto `sf`.



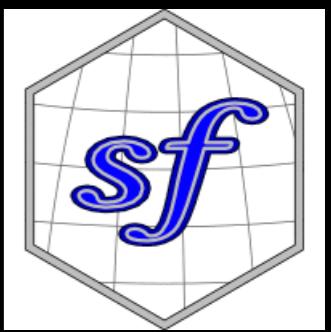
1. `sf::st_read()`



Objeto clase **sf** - `data.frame`, que se puede manipular con los verbos `dplyr` que vimos con Segasi.

**Argumentos:** `dsn`: ubicación del archivo como cadena de texto. Acepta direcciones url, por lo que podemos leer archivos de internet.

# 1. Función `sf::st_read()`



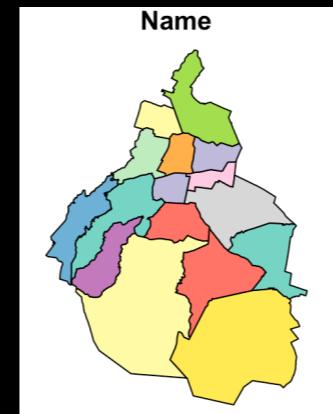
Ejemplo de uso ya en R. (Previamente hay que descargar la librería).

```
library(sf)
# 1. Podemos leer archivos de internet
mpios <- st_read("https://raw.githubusercontent.com/JuveCampos/mpios.geojson")
# 2. Podemos leer archivos localmente
mpios <- st_read("01_Datos/mpios.geojson")
# 3. Podemos leer shps o kml
alcaldias <- st_read("01_Datos/alcaldias.kml")
# Hay que tener todos los archivos para que funcione
frontera <- st_read("01_Datos/Shape Frontera/Mexico_and_US_Border.shp")
```

`plot(mprios, max.plot = 1)`



`plot(alcaldias, max.plot = 1)`



`plot(fronteras, max.plot = 1)`



# Objeto *sf*



La clase *sf* es la clase de un objeto que almacena las coordenadas de los polígonos, líneas o puntos necesarios para crear un mapa.

Generalmente, viene acompañado de la clase *data.frame*, lo cual permite que podamos utilizar en estos objetos los verbos *dplyr* para trabajar con la información de las variables que acompañan a dicha geometría (con las funciones *filter*, *summarise*, *select*, *arrange*, *mutate*, etc.).

```
> class(mprios)
[1] "sf"           "data.frame"
```

Para verificar que el objeto sea clase *sf*, utilizamos la función *class()*.

# Objeto sf



```
> mprios
Simple feature collection with 2465 features and 3 fields
geometry type:  MULTIPOLYGON
dimension:      XY
bbox:            xmin: -118.3651 ymin: 14.5321 xmax: -86.71041 ymax: 32.71865
geographic CRS: WGS 84
First 10 features:
#  CVEGEO    NOM_ENT        NOM_MUN   geometry
1  01001 Aguascalientes Aguascalientes MULTIPOLYGON (((-102.1064 2...
2  01002 Aguascalientes Asientos       MULTIPOLYGON (((-102.0519 2...
3  01003 Aguascalientes Calvillo       MULTIPOLYGON (((-102.6857 2...
4  01004 Aguascalientes Cosío         MULTIPOLYGON (((-102.2879 2...
5  01005 Aguascalientes Jesús María    MULTIPOLYGON (((-102.3357 2...
6  01006 Aguascalientes Pabellón de Arteaga MULTIPOLYGON (((-102.2535 2...
7  01007 Aguascalientes Rincón de Rosos MULTIPOLYGON (((-102.2268 2...
8  01008 Aguascalientes San José de Gracia MULTIPOLYGON (((-102.4561 2...
9  01009 Aguascalientes Tepezalá       MULTIPOLYGON (((-102.1799 2...
10 01010 Aguascalientes El Llano       MULTIPOLYGON (((-102.0295 2...
```

## Atributos:

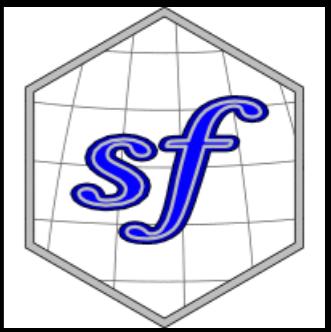
- Incluye las variables asociadas a cada una de las geometrías.

## Información de la geometría

- Incluye el número de filas (features) y columnas (fields)
- El tipo de la geometría,
- las dimensiones (XY y Z)
- Las coordenadas extremas (bbox)
- El Sistema de coordenadas de referencia

## Geometrías:

- Incluye el tipo de la geometría y, en este caso, los vértices de los polígonos de los municipios.



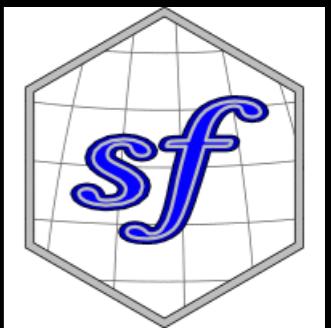
## 2. Función `sf::st_crs()`

La función `st_crs()` nos dice qué sistema de coordenadas de referencia utiliza nuestro objeto sf.

```
st_crs(mpios)
# > Coordinate Reference System:
#   User input: WGS 84

st_crs(alcaldias)
# > Coordinate Reference System:
#   User input: WGS 84

st_crs(frontera)
# > Coordinate Reference System:
#   User input: WGS 84
```



# Función `sf::st_transform()`

La función `st_transform()` nos permite transformar el crs a otro crs que nos convenga más (generalmente, el crs = 4326).

Ejemplo de uso:

```
encharcamientos_2018 <- st_read("http://www.atlas.cdmx.gob.mx/datosAbiertos  
/SPC_Encharcamientos_2018.geojson")  
st_crs(encharcamientos_2018)  
# Coordinate Reference System:  
#   User input: Mexico ITRF2008  
#   ID["EPSG",6365]  
  
# Si lo quiero transformar a 4326:  
encharcamientos_2018 <- encharcamientos_2018 %>%  
  st_transform(crs = 4326)
```

> encharcamientos\_2018  
Simple feature collection with 121 features and 20 fields  
geometry type: POINT  
dimension: XY  
bbox: xmin: -99.24384 ymin: 19.19662 xmax: -98.95801 ymax: 19.53632  
geographic CRS: WGS 84



# Función *plot()*

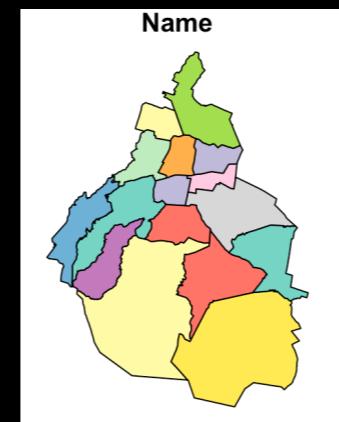


La función **plot()** nos permite hacer un mapa exploratorio de las geometrías que contiene el objeto clase *sf*,

`plot(mprios, max.plot = 1)`



`plot(alcaldias, max.plot = 1)`



`plot(fronteras, max.plot = 1)`

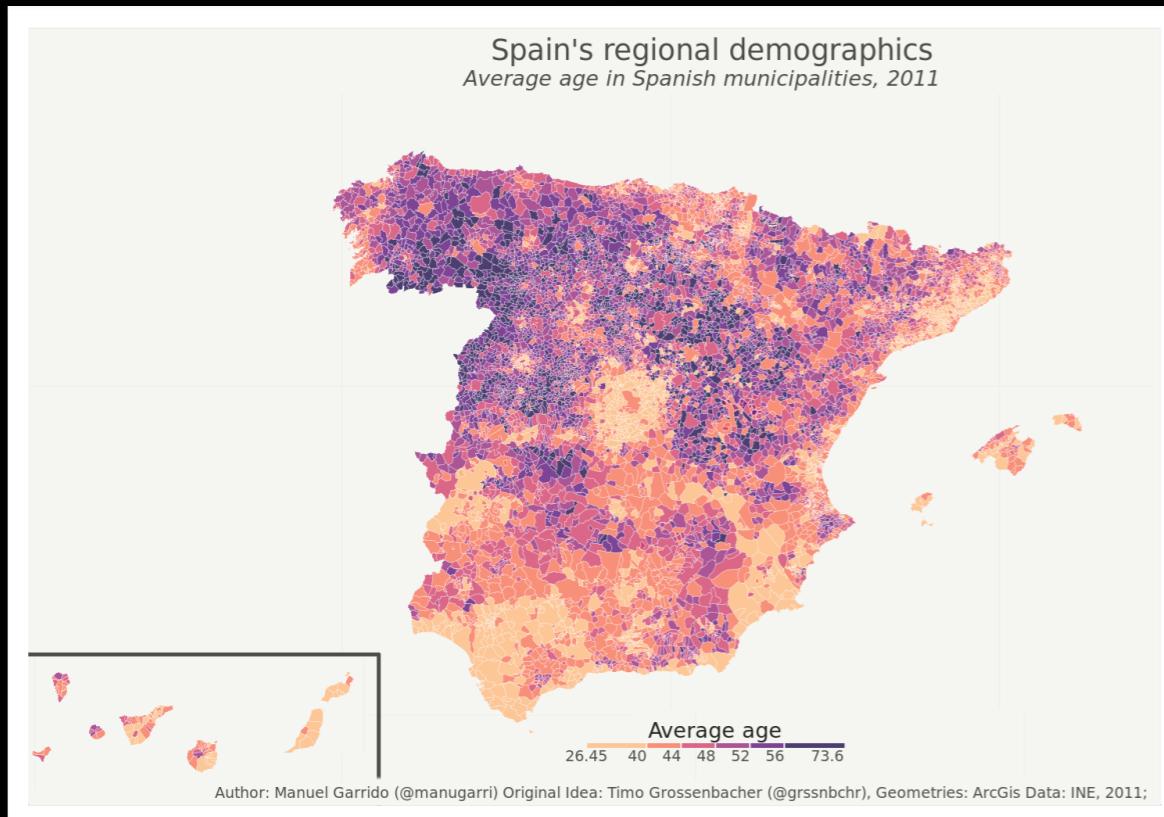


**El argumento `max.plot = 1` imprime solo un mapa de la primera variable.** Si no se incluye este argumento, imprimirá 8/9 mapas de las primeras variables (atributos) del objeto *sf*, **y esto puede congelar la computadora**, si el objeto es muy pesado.

# Visualización

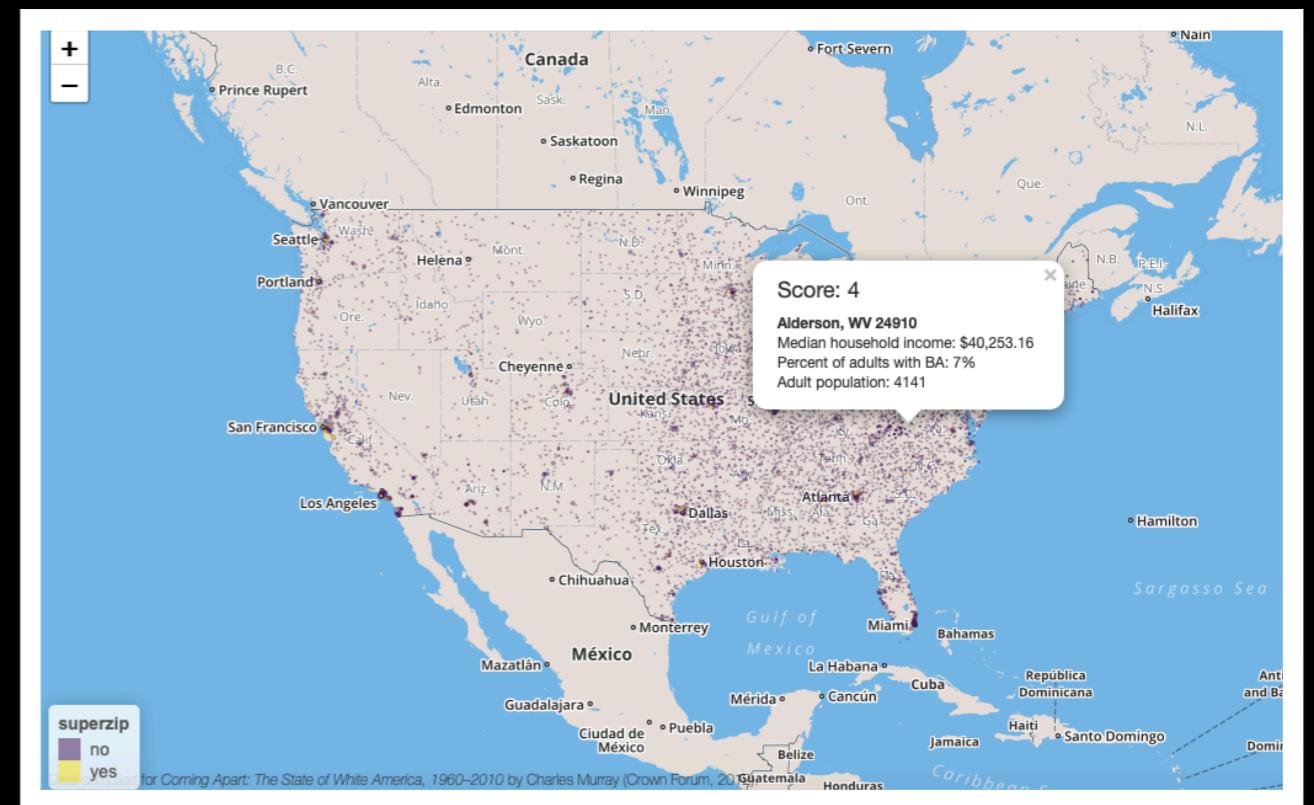
La visualización de mapas se puede hacer de dos maneras: de manera estática **{ggplot2}** e interactiva **{leaflet}**.

## {ggplot2}



## {leaflet}

(tema clase 4)



Autor: Manuel Garrido @manugarri

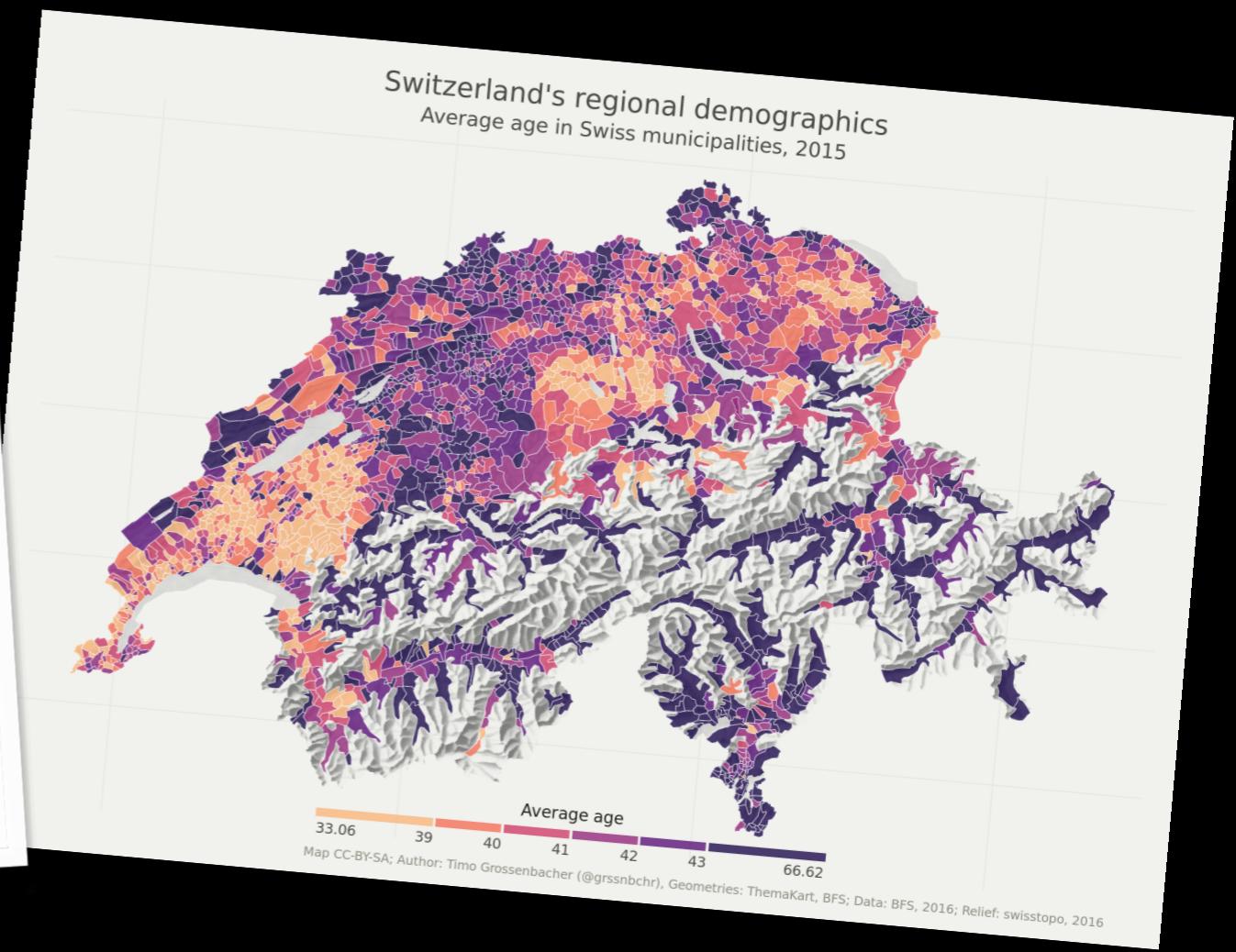
Autor: Charles Murray. Ver enlace para más referencias.

# Librería {ggplot2}



## 1. Función ggplot2::geom\_sf()

La función **geom\_sf()** es la función que nos va a permitir añadir la capa del elemento geométrico en una gráfica ggplot. Es similar a `geom_histogram`, `geom_point()` o `geom_line()`, solo que esta forzosamente requiere un objeto clase `sf` para funcionar.



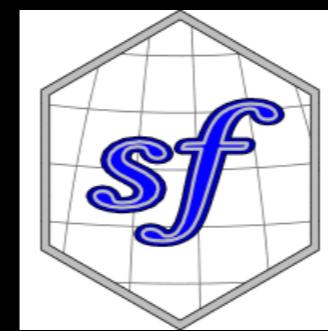
# ggplot2::geom\_sf()



## Ejemplo de uso:

```
# Hacemos el mapa
mapx <- map_estado %>%
  # Declaramos los elementos estéticos
  ggplot(aes(fill = Valor)) +
  # Generamos el mapa (y ponemos los perimetros en blanco)
  geom_sf(color = "gray90") + ←
  # Cambiamos los colores del relleno
  scale_fill_gradientn(colors = viridis(begin = 0,
                                         end = 1,
                                         n = 10))|
```

# Sección práctica.



# Sección práctica.



**Objetivo:** Realizar un mapa del IDH para los municipios del Estado de Morelos.

## Procedimiento:

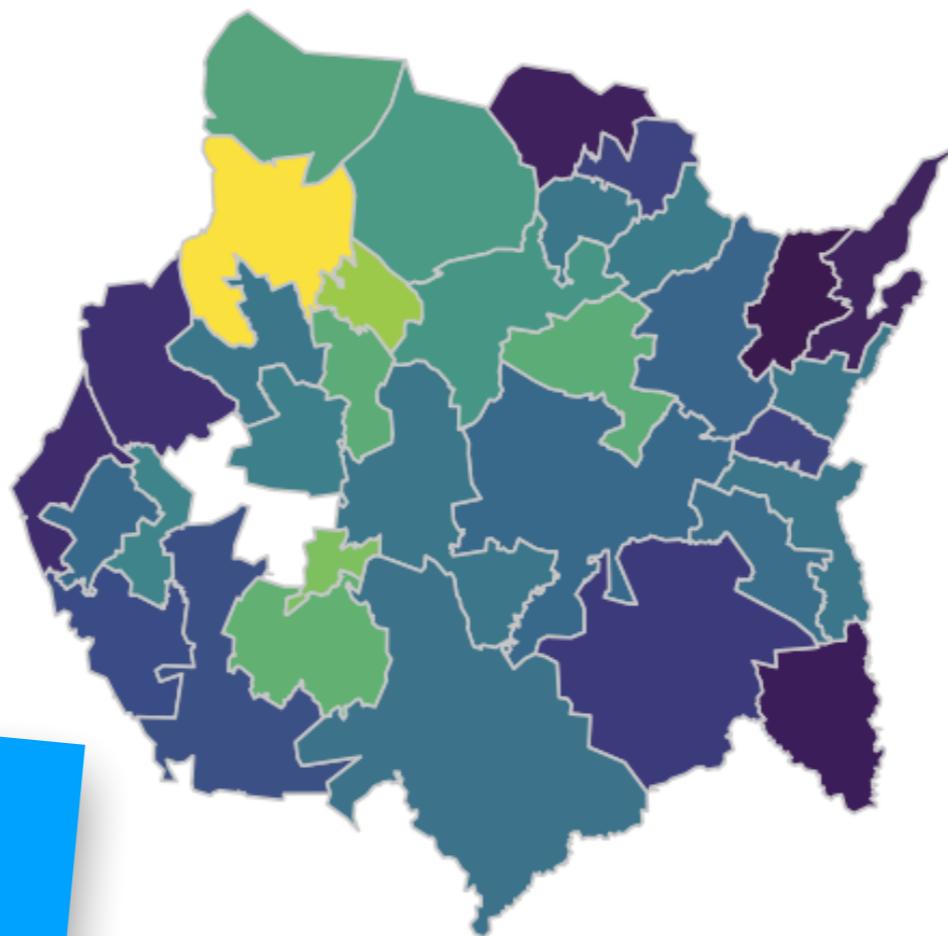
1. **Tener la base de datos del IDH a nivel municipal, y la base de geometrías de los municipios del estado de Morelos.**
2. **Ubicar cual es la variable “llave” (la variable común entre las dos bases).**
3. **Pegar la base de datos de atributos con la base de datos geográficos para tener lista la base de datos que vamos a mapear.**
4. **Generar el mapa, dando color a cada polígono en función del valor del IDH.**
5. **Editar el tema del mapa.**

# Sección práctica.

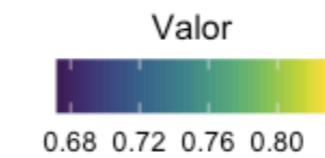


Índice de Desarrollo Humano

Año: 2015



*Hay que replicar esto  
en RStudio!*



Fuente: Informe de Desarrollo Humano Municipal 2010-2015.  
Transformando México Desde lo Local

# Manos a la obra en R.

