

# Gestión de información con atributos espaciales en R

José Luis Texcalac Sangrador

# Modelo geométrico de rasgo simple (*simple feature* ○ *simple feature access*)

- Estándar formal (ISO 19125-1: 2004) que describe cómo los objetos en el mundo real pueden ser representados en las computadoras, con énfasis en la geometría espacial de estos objetos.
- Estándar de código abierto desarrollado y respaldado por el Open Geospatial Consortium (OGC)
- Permite representar una amplia gama de información geográfica.
- Modelo de datos jerárquico que simplifica los datos geográficos al condensar un amplio rango de formas geográficas en una única clase de geometría.
- Describe cómo los objetos se pueden almacenar y recuperar de las bases de datos, y las operaciones geométricas que deberían definirse para ellos.
- Modelo de datos ampliamente respaldado que subyace a las estructuras de datos en muchas aplicaciones de SIG, permitiendo la transferencia de datos a otras configuraciones (software, librerías, etc.).

# *feature*

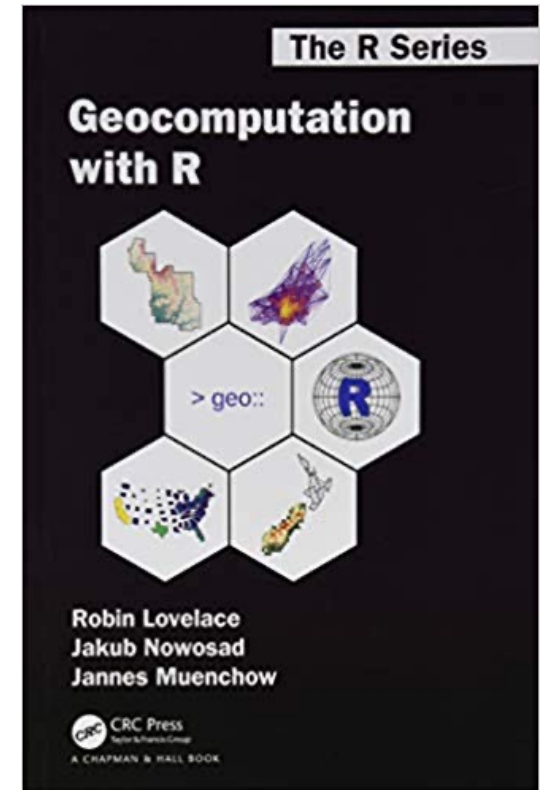
- Se considera una cosa o un objeto en el mundo real (un edificio o un árbol)
- Como en la vida real, los objetos, a menudo conforman otros objetos. Un conjunto de *features* puede formar un solo *feature*.
  - Un árbol puede ser un *feature*, un bosque es un *feature*.
  - Un edificio puede ser un *feature* y una ciudad es un *feature*.
  - Un píxel de una imagen satelital es un *feature*, y la imagen completa también es un *feature*.
- Tienen una geometría que describe su posición sobre la superficie terrestre.
- Tienen atributos que describen propiedades adicionales.
  - La geometría de un árbol puede ser la delineación de su corona, de su tallo o el punto que indica su centro.
  - Otras propiedades pueden incluir su altura, color, diámetro a la altura del pecho en una fecha particular, y así sucesivamente.
- El estándar dice: "La OpenGIS Abstract especifica que un *simple feature* posee ambas características, es decir, atributos espaciales y no espaciales."

# Paquete *sf*

- Compatible con los tipos de *simple feature* utilizados en la mayoría de las operaciones de análisis espacial: puntos, líneas, polígonos y sus respectivas versiones “múltiples”.
- Permite también crear colecciones de geometrías, que pueden contener diferentes tipos de geometría en un solo objeto.
- Incorpora de forma integral la funcionalidad de los tres paquetes principales de *sp*.
  - *sp*: Para el sistema de clases de objetos espaciales;
  - *rgdal*: para leer y escribir datos;
  - *rgeos*: para operaciones espaciales.

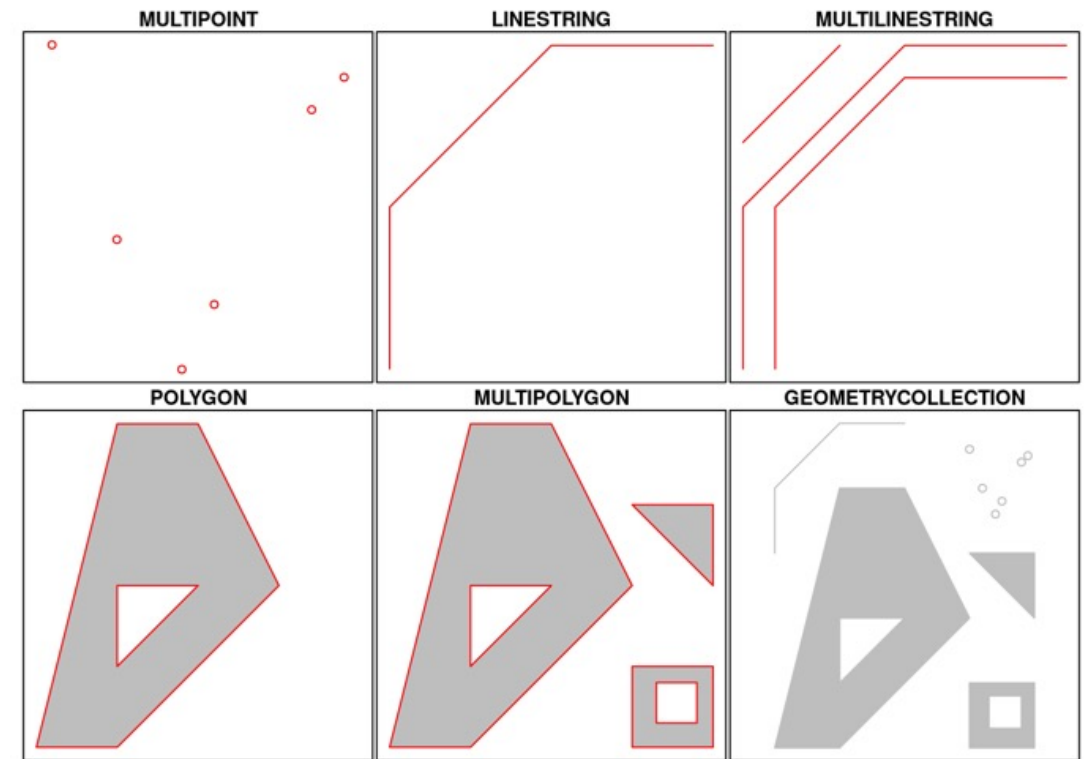
# Ventajas del uso del paquete *sf*

- Proporciona una interfaz casi directa para las funciones GDAL y GEOS C ++
- La lectura y escritura de datos son rápidas.
- El rendimiento de despliegue.
- Los nombres de las funciones de *sf* son relativamente consistentes e intuitivas (todos comienzan con *st\_*).
- Los objetos *sf* pueden procesarse como un *tibble* o *data frame*.
- Las funciones pueden combinarse usando el operador pipe *%>%*
- Funcionan bien con la colección de librerías de tidyverse.



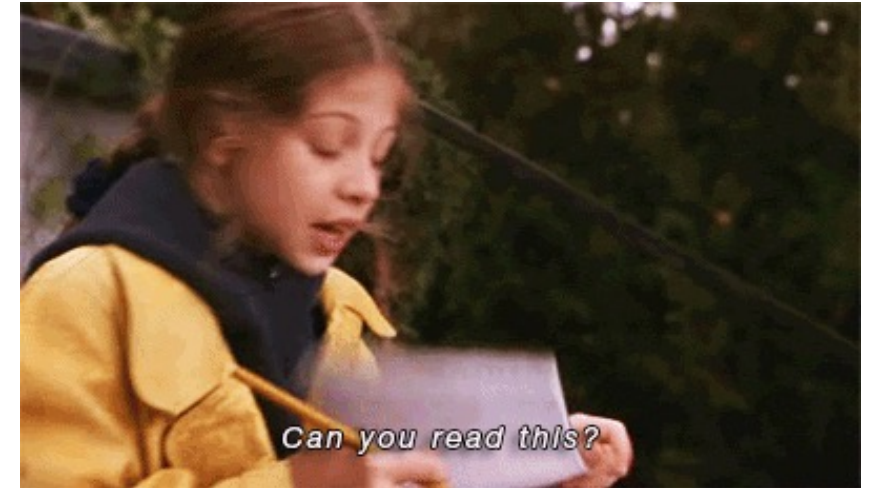
# Objetos espaciales en el paquete *sf*

- Existen diferentes clases de objetos espaciales dependiendo del tipo de información que almacenan
- Permiten gestionar coberturas de puntos, líneas y polígonos.
  - *Point* y *Multipoint*,
  - *Linestring* y *Multilinestring*
  - *Polygon* y *Multipolygon*
- La clase *Geometrycollection* permite juntar, en un mismo objeto, diferentes geometrías.



# Shapefiles

- El formato ESRI Shapefile (SHP) es un formato de archivo de datos espaciales desarrollado por la compañía ESRI, quien crea y comercializa software para Sistemas de Información Geográfica como Arc/Info o ArcGIS.
- Originalmente se creó para la utilización con su producto ArcView GIS, pero actualmente se ha convertido en formato estándar de facto para el intercambio de información geográfica entre Sistemas de Información Geográfica.
  - Requiere de 4 archivos para su uso (.shp, .shx, .dbf, .prj)
  - Límite de 10 caracteres para nombres de campo
  - Problemas con el almacenamiento de fechas
  - Limitado para almacenar valores nulos
  - Límite de 2 Gb
  - No puede contener algunos tipos de geometrías
- El más usado y el que trabajaremos durante el curso.





# Shapefiles del marco geoestadístico nacional

< > shp_inegi_2019	
Nombre	^
02_bajacalifornia	↻
catalogos	↻
contenido.pdf	↻
contenido.txt	↻
entidades_federativas.csv	↻
entidades_federativas.pdf	↻
leme.pdf	↻
leme.txt	↻
localidades_islas.csv	↻
localidades_islas.pdf	↻
localidades_rura...basan_AGEb.csv	↻
localidades_rura...basan_AGEb.pdf	↻
localidades_rura...su_municipio.csv	↻
localidades_rura...su_municipio.pdf	↻
localidades_urb...su_municipio.csv	↻
localidades_urb...su_municipio.pdf	↻
localidades_urb...amanzanadas.csv	↻
localidades_urb...amanzanadas.pdf	↻
municipios.csv	↻
municipios.pdf	↻
conjunto_de_datos	↻
metadatos	↻

< > shp_inegi_2019	
Nombre	^
02_bajacalifornia	✓
catalogos	✓
conjunto_de_datos	✓
02a.cpg	✓
02a.dbf	✓
02a.prj	✓
02a.sbn	✓
02a.sbx	✓
02a.shp	✓
02a.shx	✓
02ar.cpg	✓
02ar.dbf	✓
02ar.prj	✓
02ar.shp	✓
02ar.shx	✓
02cd.cpg	✓
02cd.dbf	✓
02cd.prj	✓
02cd.sbn	✓
02cd.sbx	✓
02cd.shp	✓
02cd.shx	✓

< > shp_inegi_2019	
Nombre	^
02_bajacalifornia	↻
catalogos	↻
conjunto_de_datos	↻
metadatos	↻
02_mg_septiembre2019.rtf	↻
02_mg_septiembre2019.txt	↻
02_mg_septiembre2019.xml	↻
metadatos_mg_...ptiembre2019.txt	↻

# Shapefiles en nuestro proyecto

< > epi_esp		⋮ ⬆	⋮ ⬆
Nombre			^
>	data		✓
>	output		✓
>	scripts		✓
▼	shapefiles		↻
▼	02_bajacalifornia		↻
>	catalogos		✓
>	conjunto_de_datos		↻
>	metadatos		✓
	epi_esp.Rproj		✓

# Importando datos con atributos espaciales en R

```
library(tidyverse)
```

```
library(sf)
```

```
objeto <- st_read("ruta_archivo.shp")
```

- Alias

```
objeto <- read_sf("ruta_archivo.shp")
```

# st\_read( )

```
objeto <-  
  st_read("ruta_archivo.shp",  
          stringsAsFactors = FALSE,  
          options = "ENCODING = WINDOWS-1252")
```

# Estructura de un *simple feature*

```
> bc_mun
```

```
Simple feature collection with 5 features and 4 fields
```

```
geometry type:  MULTIPOLYGON
```

```
dimension:      XY
```

```
bbox:           xmin: 911292 ymin: 1810279 xmax: 1493532 ymax: 2349615
```

```
projected CRS:  MEXICO_ITRF_2008_LCC
```

	CVEGEO	CVE_ENT	CVE_MUN	NOMGEO	geometry
1	02001	02	001	Ensenada	MULTIPOLYGON (((1493197 184...
2	02002	02	002	Mexicali	MULTIPOLYGON (((1437921 188...
3	02003	02	003	Tecate	MULTIPOLYGON (((1113633 234...
4	02004	02	004	Tijuana	MULTIPOLYGON (((1058854 233...
5	02005	02	005	Playas de Rosarito	MULTIPOLYGON (((1077326 233...

*Simple feature*

*Simple feature geometry list-column (sfc)*

*Simple feature geomtery (sfg)*

# Polígonos

# Su turno...

- Cargue la capa "02mun"
- Nombre al nuevo objeto como bc\_mun

```
library(tidyverse)
```

```
library(sf)
```

```
objeto <- st_read("ruta_archivo.shp")
```

# Leyendo datos de Baja California

```
bc_mun <-  
  st_read("./shapefiles/02_bajacalifornia/conjunto_de_datos/02mun.shp") %>%  
  clean_names() %>%  
  print( )
```



```

> bc_mun <-
+   st_read("./shapefiles/02_bajacalifornia/conjunto_de_datos/02mun.shp") %>%
+   clean_names() %>%
+   print( )
Reading layer `02mun' from data source
  `/Users/tex/OneDrive/Docencia/PASPE_R_EPI_Espacial/epi_esp/shapefiles/02_bajacalifornia/conjunto_de_datos/02mun.shp'
  using driver `ESRI Shapefile'
Simple feature collection with 5 features and 4 fields
Geometry type: MULTIPOLYGON
Dimension:      XY
Bounding box:   xmin: 911292 ymin: 1810279 xmax: 1493532 ymax: 2349615
Projected CRS: MEXICO_ITRF_2008_LCC
Simple feature collection with 5 features and 4 fields
Geometry type: MULTIPOLYGON
Dimension:      XY
Bounding box:   xmin: 911292 ymin: 1810279 xmax: 1493532 ymax: 2349615
Projected CRS: MEXICO_ITRF_2008_LCC
  cvegeo cve_ent cve_mun      nomgeo      geometry
1  02001      02    001    Ensenada MULTIPOLYGON (((1493197 184...
2  02002      02    002    Mexicali MULTIPOLYGON (((1437921 188...
3  02003      02    003      Tecate MULTIPOLYGON (((1113633 234...
4  02004      02    004      Tijuana MULTIPOLYGON (((1058854 233...
5  02005      02    005 Playas de Rosarito MULTIPOLYGON (((1077326 233...

```

# `plot(st_geometry(bc_mun))`

- `st_geometry` nos permite visualizar sólo la geometría de la capa.
- Intente el plot omitiendo `st_geometry`



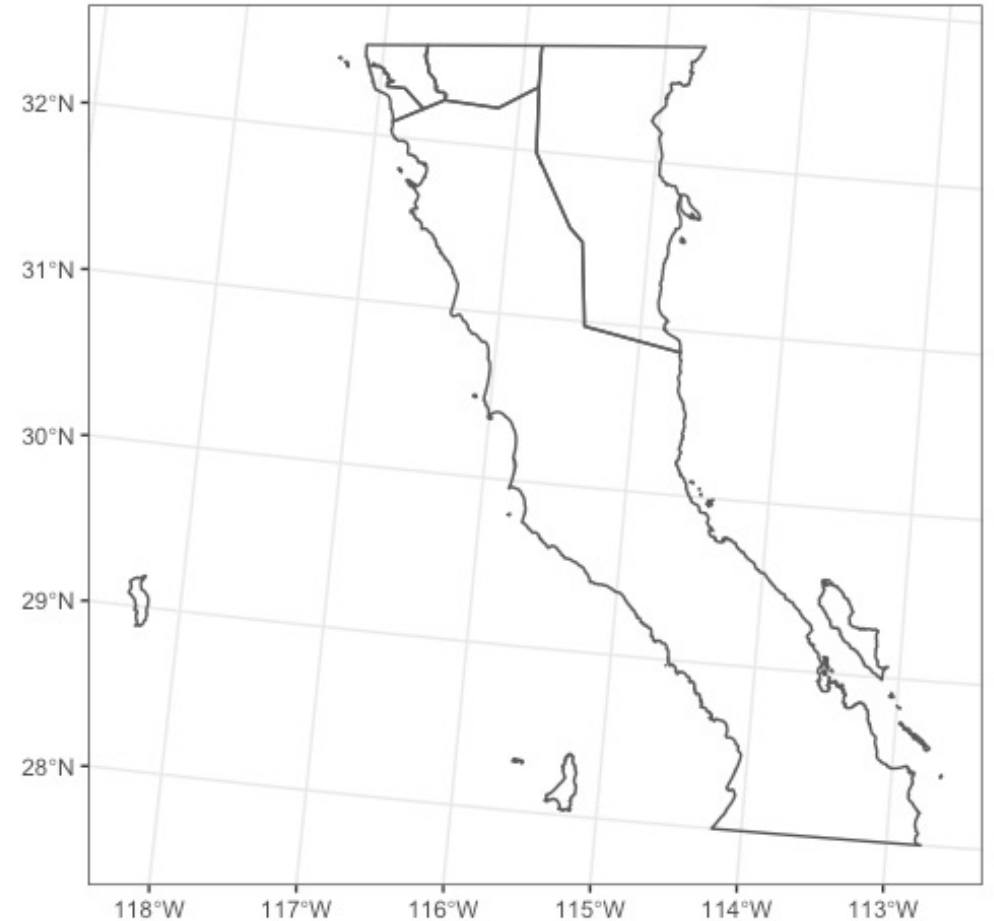
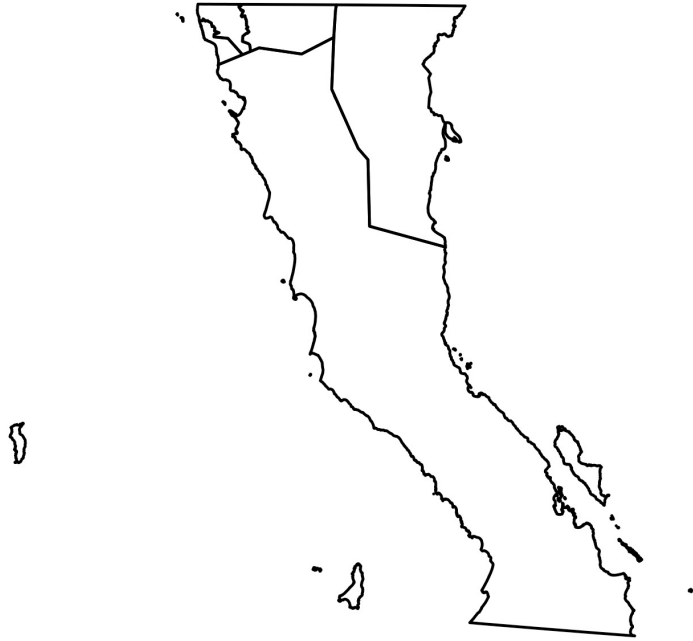
# Mapa básico con ggplot

```
ggplot() %>%  
  geom_sf(...) %>%  
  print( )
```

`ggplot()` +

`geom_sf(data = bc_mun, fill = "transparent")` +

`theme_bw()`



# Error de proyección en capa INEGI

```
> ggplot() +  
+   geom_sf(data = bc_mun, fill = "transparent") +  
+   theme_bw()
```

```
Error in CPL_transform(x, crs, aoi, pipeline, reverse, desired_accuracy, :  
  OGRCreateCoordinateTransformation() returned NULL: PROJ available?
```

Además: Warning messages:

```
1: In CPL_transform(x, crs, aoi, pipeline, reverse, desired_accuracy, :  
  GDAL Error 1: PROJ: proj_create_operations: Inconsistent chaining of CRS in operations
```

```
2: In CPL_transform(x, crs, aoi, pipeline, reverse, desired_accuracy, :  
  GDAL Error 6: Cannot find coordinate operations from `PROJCRS["MEXICO_ITRF_2008_LCC",BASEGEOGCRS["ITRF2008",DATUM["In  
ternational Terrestrial Reference Frame 2008",ELLIPSOID["GRS 1980",6378137,298.257222101,LENGTHUNIT["metre",1]],ID["EPS  
G",1061]],PRIMEM["Greenwich",0,ANGLEUNIT["Degree",0.0174532925199433]]],CONVERSION["unnamed",METHOD["Lambert Conic Conf  
ormal (2SP)",ID["EPSG",9802]],PARAMETER["Latitude of false origin",12,ANGLEUNIT["Degree",0.0174532925199433],ID["EPSG",  
8821]],PARAMETER["Longitude of false origin",-102,ANGLEUNIT["Degree",0.0174532925199433],ID["EPSG",8822]],PARAMETER["La  
titude of 1st standard parallel",17.5,ANGLEUNIT["Degree",0.0174532925199433],ID["EPSG",8823]],PARAMETER["Latitude of 2n  
d standard parallel",29.5,ANGLEUNIT["Degree",0.0174532925199433],ID["EPSG",8824]],PARAMETER["Easting at false origin",2  
500000,LENGTHUNIT["metre",1],ID["EPSG",8826]],PARAMETER["Northing at false origin",0,LENGTHUNIT["metre",1],ID["EPSG",88  
27]]],CS[Cartesian,2],AXIS["(E)",east,ORDER[1],LEN [... truncated]
```

# st\_crs(bc\_mun)

```
> st_crs(bc_mun)
```

Coordinate Reference System:

User input: MEXICO\_ITRF\_2008\_LCC

wkt:

```
PROJCRS["MEXICO_ITRF_2008_LCC",  
  BASEGEOGCRS["ITRF2008",  
    DATUM["International Terrestrial Reference Frame 2008",  
      ELLIPSOID["GRS 1980",6378137,298.257222101,  
        LENGTHUNIT["metre",1]],  
      ID["EPSG",1061]],  
    PRIMEM["Greenwich",0,  
      ANGLEUNIT["Degree",0.0174532925199433]]],  
  CONVERSION["unnamed",  
    METHOD["Lambert Conic Conformal (2SP)",  
      ID["EPSG",9802]],  
    PARAMETER["Latitude of false origin",12,  
      ANGLEUNIT["Degree",0.0174532925199433],  
      ID["EPSG",8821]],  
    PARAMETER["Longitude of false origin",-102,  
      ANGLEUNIT["Degree",0.0174532925199433],  
      ID["EPSG",8822]],  
    PARAMETER["Latitude of 1st standard parallel",17.5,  
      ANGLEUNIT["Degree",0.0174532925199433],  
      ID["EPSG",8823]],  
    PARAMETER["Latitude of 2nd standard parallel",29.5,  
      ANGLEUNIT["Degree",0.0174532925199433],  
      ID["EPSG",8824]],  
    PARAMETER["Easting at false origin",2500000,  
      LENGTHUNIT["metre",1],  
      ID["EPSG",8826]],  
    PARAMETER["Northing at false origin",0,  
      LENGTHUNIT["metre",1],  
      ID["EPSG",8827]]],  
  CS[Cartesian,2],  
  AXIS["(E)",east,  
    ORDER[1],  
    LENGTHUNIT["metre",1,  
      ID["EPSG",9001]]],  
  AXIS["(N)",north,  
    ORDER[2],  
    LENGTHUNIT["metre",1,  
      ID["EPSG",9001]]]]
```



# bc\_mun <- st\_transform(bc\_mun, 6372)

Original

```
> st_crs(bc_mun)
Coordinate Reference System:
User input: MEXICO_ITRF_2008_LCC
wkt:
PROJCRS["MEXICO_ITRF_2008_LCC",
  BASEGEOGCRS["ITRF2008",
    DATUM["International Terrestrial Reference Frame 2008",
      ELLIPSOID["GRS 1980",6378137,298.257222101,
        LENGTHUNIT["metre",1]],
      ID["EPSG",1061]],
    PRIMEM["Greenwich",0,
      ANGLEUNIT["Degree",0.0174532925199433]],
    CONVERSION["unnamed",
      METHOD["Lambert Conic Conformal (2SP)",
        ID["EPSG",9802]],
      PARAMETER["Latitude of false origin",12,
        ANGLEUNIT["Degree",0.0174532925199433],
        ID["EPSG",8821]],
      PARAMETER["Longitude of false origin",-102,
        ANGLEUNIT["Degree",0.0174532925199433],
        ID["EPSG",8822]],
      PARAMETER["Latitude of 1st standard parallel",17.5,
        ANGLEUNIT["Degree",0.0174532925199433],
        ID["EPSG",8823]],
      PARAMETER["Latitude of 2nd standard parallel",29.5,
        ANGLEUNIT["Degree",0.0174532925199433],
        ID["EPSG",8824]],
      PARAMETER["Easting at false origin",2500000,
        LENGTHUNIT["metre",1],
        ID["EPSG",8826]],
      PARAMETER["Northing at false origin",0,
        LENGTHUNIT["metre",1],
        ID["EPSG",8827]]],
    CS[Cartesian,2],
    AXIS["(E)",east,
      ORDER[1],
      LENGTHUNIT["metre",1,
        ID["EPSG",9001]]],
    AXIS["(N)",north,
      ORDER[2],
      LENGTHUNIT["metre",1,
        ID["EPSG",9001]]]]]
```

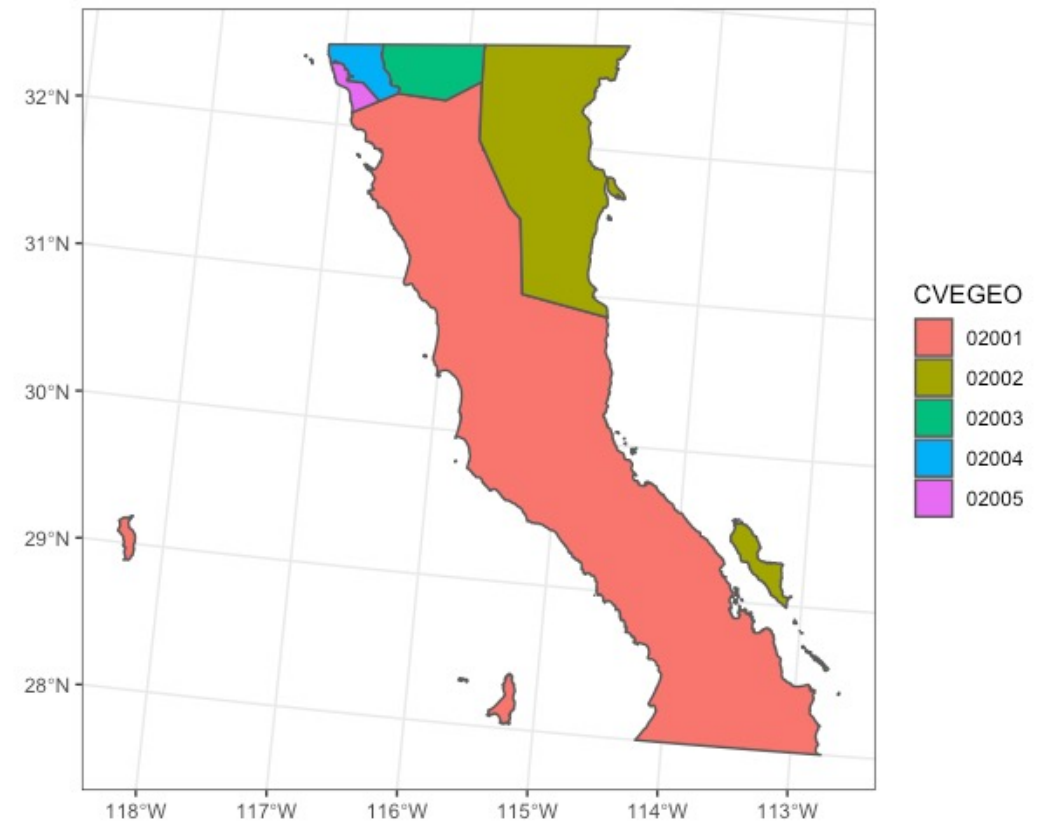
Reproyectado

```
> st_crs(bc_mun)
Coordinate Reference System:
User input: EPSG:6372
wkt:
PROJCRS["Mexico ITRF2008 / LCC",
  BASEGEOGCRS["Mexico ITRF2008",
    DATUM["Mexico ITRF2008",
      ELLIPSOID["GRS 1980",6378137,298.257222101,
        LENGTHUNIT["metre",1]]],
    PRIMEM["Greenwich",0,
      ANGLEUNIT["degree",0.0174532925199433]],
    ID["EPSG",6365]],
  CONVERSION["Mexico LCC",
    METHOD["Lambert Conic Conformal (2SP)",
      ID["EPSG",9802]],
    PARAMETER["Latitude of false origin",12,
      ANGLEUNIT["degree",0.0174532925199433],
      ID["EPSG",8821]],
    PARAMETER["Longitude of false origin",-102,
      ANGLEUNIT["degree",0.0174532925199433],
      ID["EPSG",8822]],
    PARAMETER["Latitude of 1st standard parallel",17.5,
      ANGLEUNIT["degree",0.0174532925199433],
      ID["EPSG",8823]],
    PARAMETER["Latitude of 2nd standard parallel",29.5,
      ANGLEUNIT["degree",0.0174532925199433],
      ID["EPSG",8824]],
    PARAMETER["Easting at false origin",2500000,
      LENGTHUNIT["metre",1],
      ID["EPSG",8826]],
    PARAMETER["Northing at false origin",0,
      LENGTHUNIT["metre",1],
      ID["EPSG",8827]]],
  CS[Cartesian,2],
  AXIS["northing (N)",north,
    ORDER[1],
    LENGTHUNIT["metre",1]],
  AXIS["easting (E)",east,
    ORDER[2],
    LENGTHUNIT["metre",1]],
  USAGE[
    SCOPE["Topographic mapping (small scale)."],
    AREA["Mexico - onshore and offshore."],
    BBOX[12.1,-122.19,32.72,-84.64]],
  ID["EPSG",6372]]]
```

```
ggplot() +  
  geom_sf(data = bc_mun, aes(fill = cvegeo)) +  
  theme_bw()
```

Podemos colorear polígonos tomando como base los valores de sus atributos.

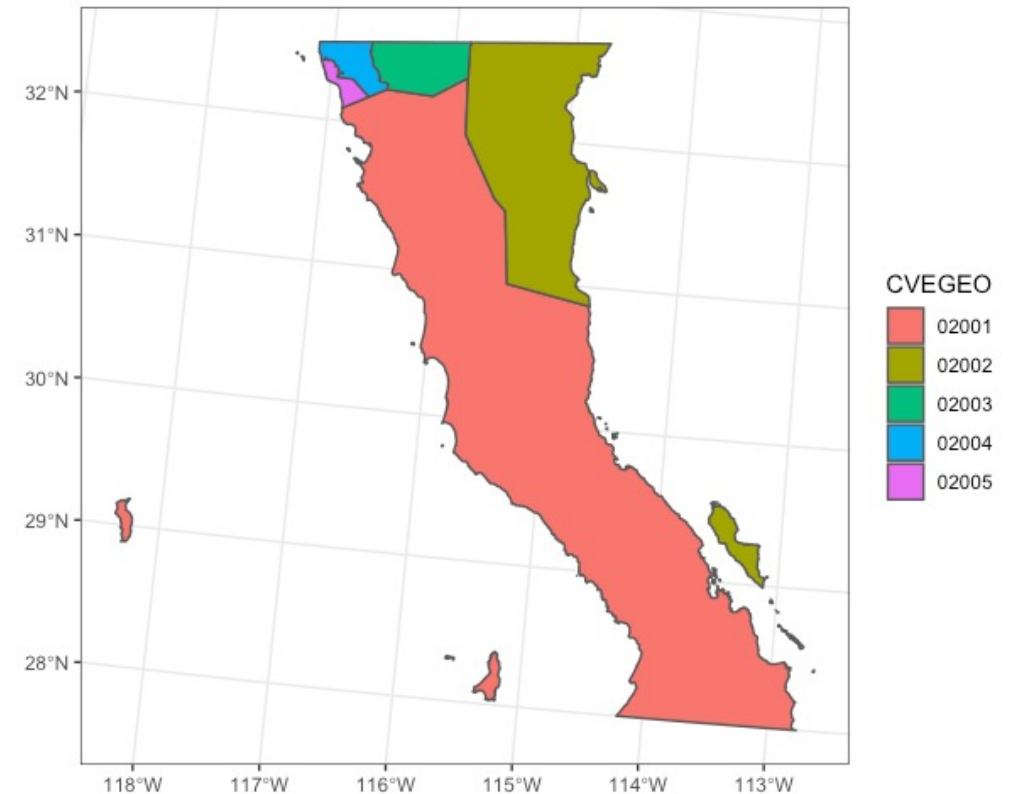
- Población
- Contaminación
- Pobreza
- Vulnerabilidad



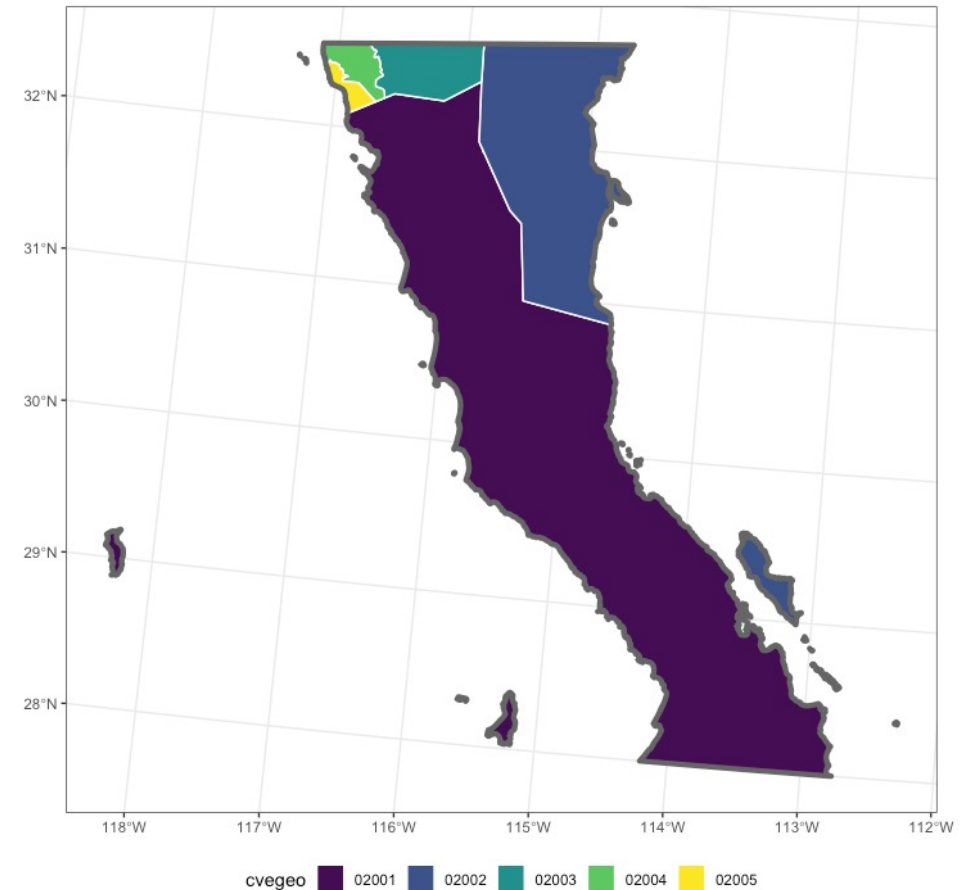


# Su turno... generando mapa

- Cargue la capa de entidad como `bc_ent`
- Contorno de municipios blanco
- Contorno de entidad con grosor de 1.5
- Paleta de colores viridis
- Aplique el tema `bw`
- Leyenda en la parte baja



```
ggplot() +  
  geom_sf(data = bc_mun, aes(fill = CVEGEO), colour = "white") +  
  geom_sf(data = bc_ent, fill = "transparent", colour = "grey30", size = 1.5) +  
  scale_fill_viridis(discrete = T) +  
  theme_bw() +  
  theme(legend.position = "bottom")
```



líneas

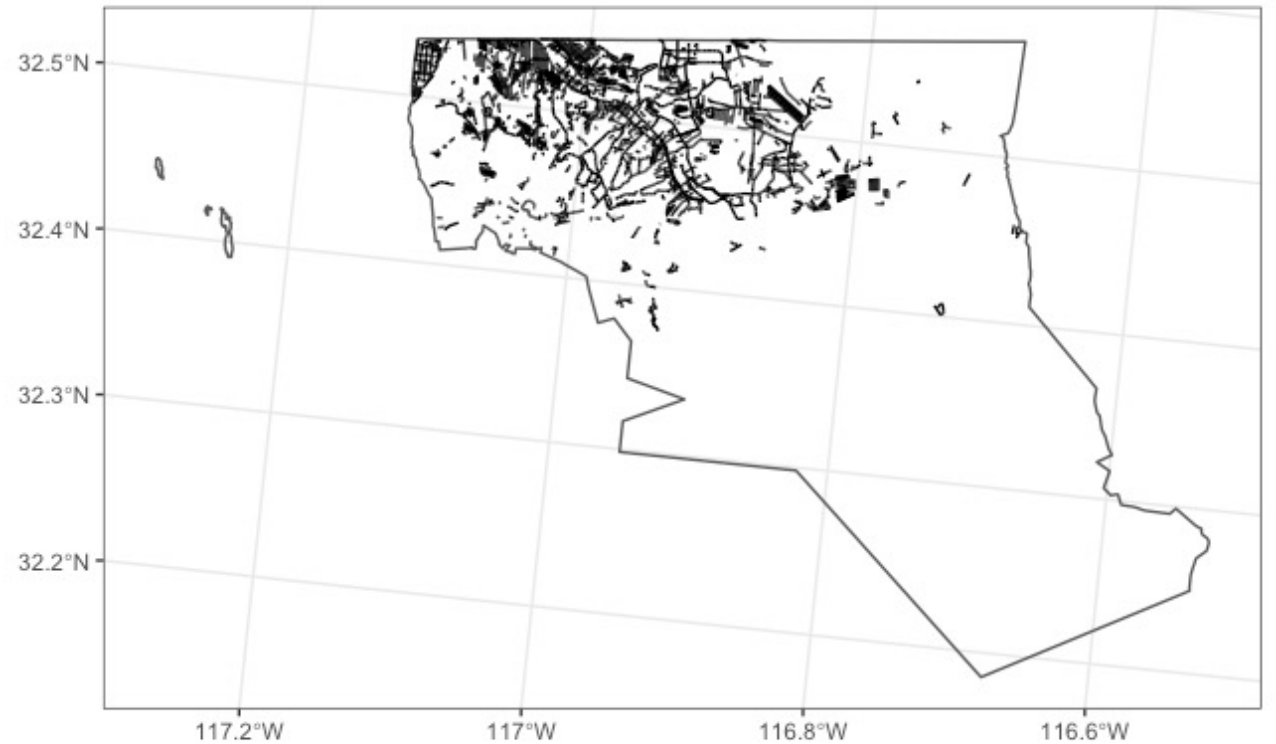
# Su turno

- Cargue la capa "02e.shp"
- Nombre al objeto como `bc_vial`
- ¿Qué valores distintos tiene la columna `tipovial`?
  - Filtre por `Avenida, Boulevard, Calzada y Carretera`
  - Filtre sólo los *features* del municipio de Tijuana
  - Llame a la capa `tij_vial`
- Filtre el municipio de Tijuana en `bc_mun` y llame a la nueva capa `tij_mun`
- Genere un mapa con ambas capas

```
ggplot() +  
  geom_sf(data = tij_mun, fill = "transparent") +  
  geom_sf(data = tij_vial) +  
  theme_bw()
```

¿Qué más podemos hacer?

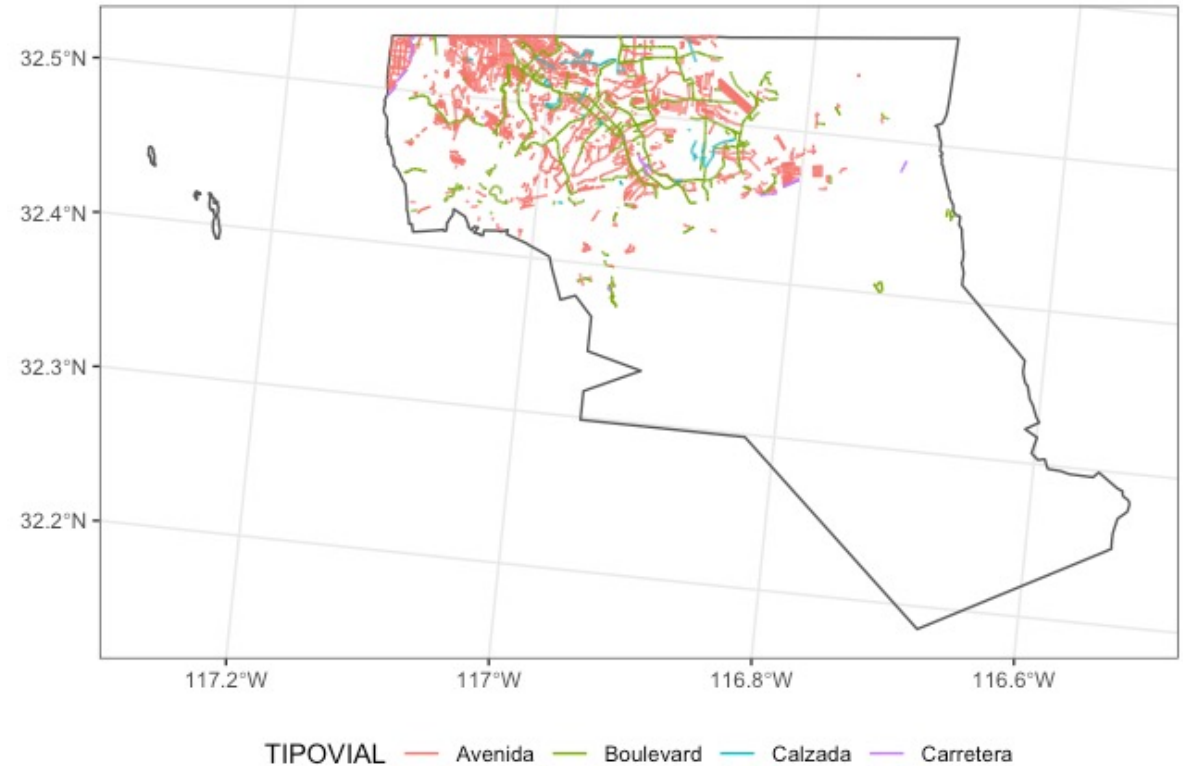
¿Colores de avenidas?



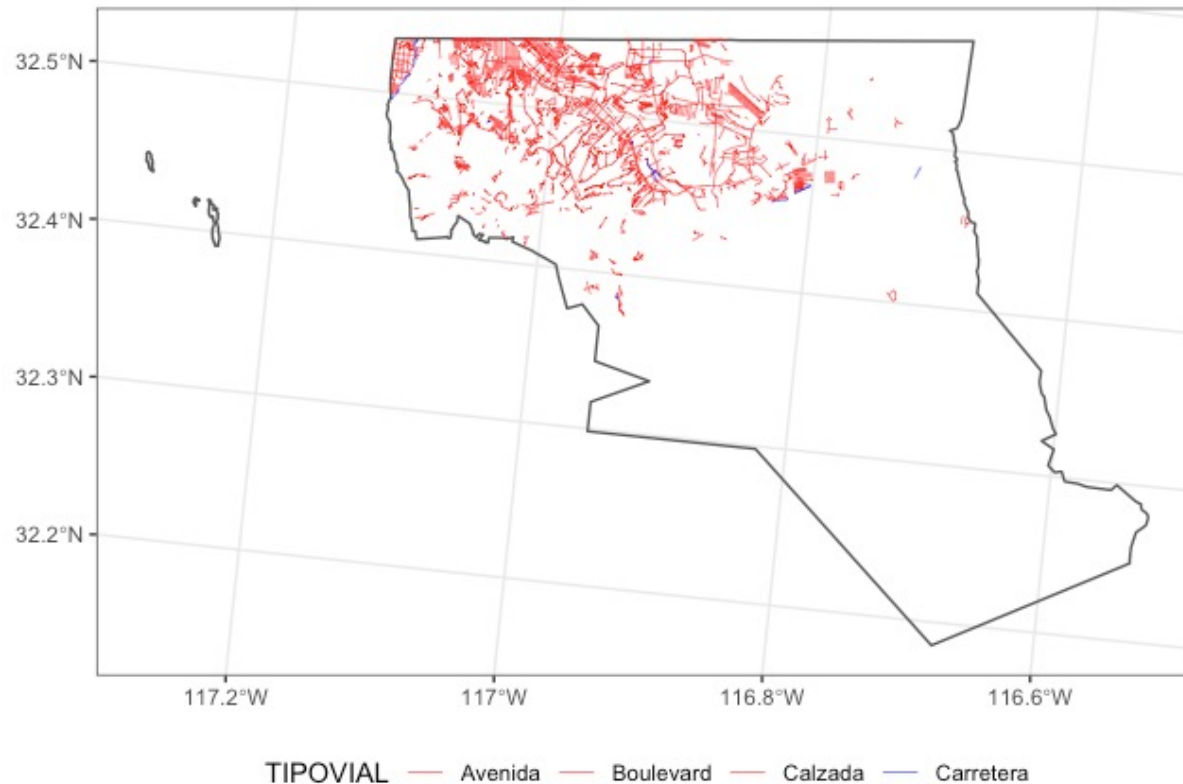
```
ggplot() +  
  geom_sf(data = tij_mun, fill = "transparent") +  
  geom_sf(data = tij_vial, aes(colour = tipovial)) +  
  theme_bw() +  
  theme(legend.position = "bottom")
```

¿Qué más podemos hacer?

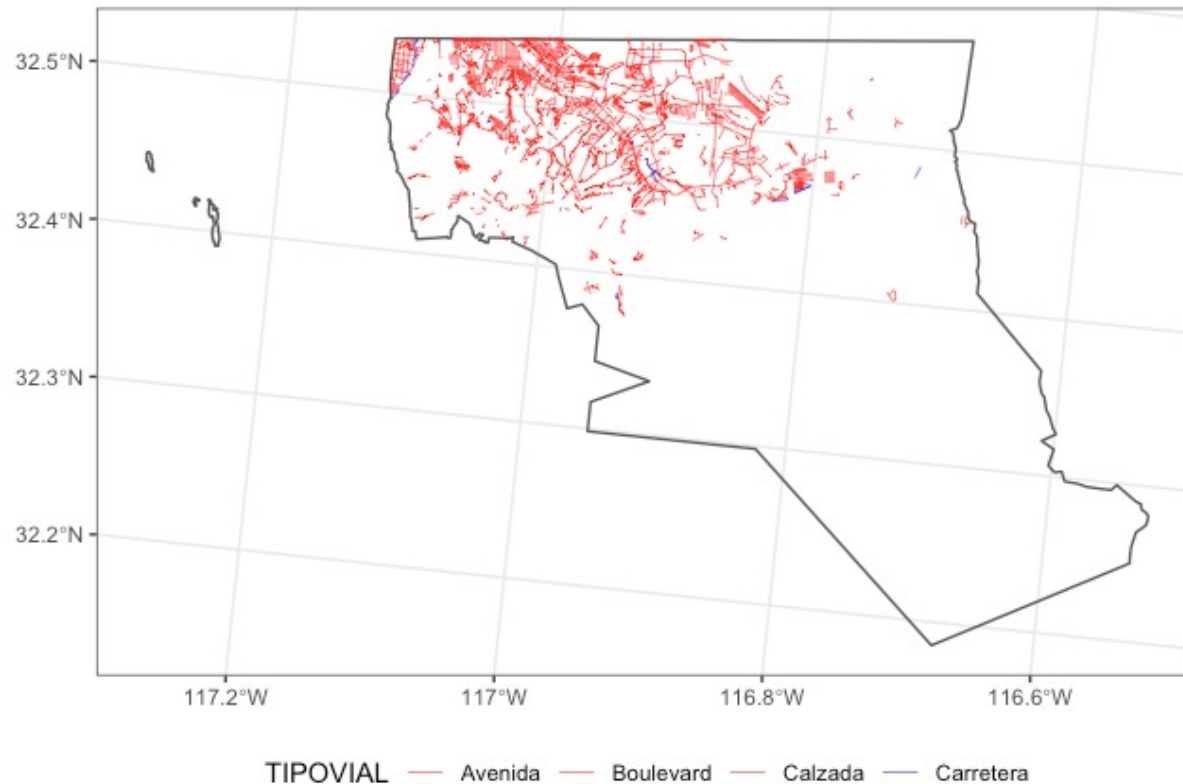
Boulevard, Calzada y Avenida son casi lo mismo, deberían ser un solo color que las diferencie de Carretera.



```
ggplot() +  
  geom_sf(data = tij_mun, fill = "transparent") +  
  geom_sf(data = tij_vial, aes(colour = tipovial), size = 0.2) +  
  scale_linetype_manual(values = c("solid", "solid", "solid", "dashed")) +  
  scale_color_manual(values = c('red', 'red', 'red', 'blue')) +  
  theme_bw() +  
  theme(legend.position = "bottom")
```



```
ggplot() +  
  geom_sf(data = tij_mun, fill = "transparent") +  
  geom_sf(data = tij_vial, aes(colour = tipovial), show.legend = "line", size = 0.2) +  
  scale_linetype_manual(values = c("solid", "solid", "solid", "dashed")) +  
  scale_color_manual(values = c('red', 'red', 'red', 'blue')) +  
  theme_bw() +  
  theme(legend.position = "bottom")
```





puntos

# Su turno

- Cargue la capa
  - "02sip.shp" (bc\_serv\_punt)
- Capas municipio, puntual y eje vial
  - Filtre sólo los *features* del municipio de Mexicali
  - Nombre los objetos nuevos como: mxc\_mun, mxc\_serv\_punt, mxc\_vial

bc\_mun

```
mxm_mun <- bc_mun %>% filter(cve_mun == "002") %>% print()
```

```
plot(st_geometry(mxm_mun))
```

# Su turno – Mapeando

- Genere un mapa con las capas de Mexicali
  - Vialidades todas de color rojo y tamaño 0.2
  - Puntos de color "dodgerblue4" y tamaño 0.3
  - Tema: theme\_bw

ggplot() +

geom\_sf(data = mxc\_mun, fill = "transparent") +

geom\_sf(data = mxc\_vial, colour = "red", show.legend = "line", size = 0.2) +

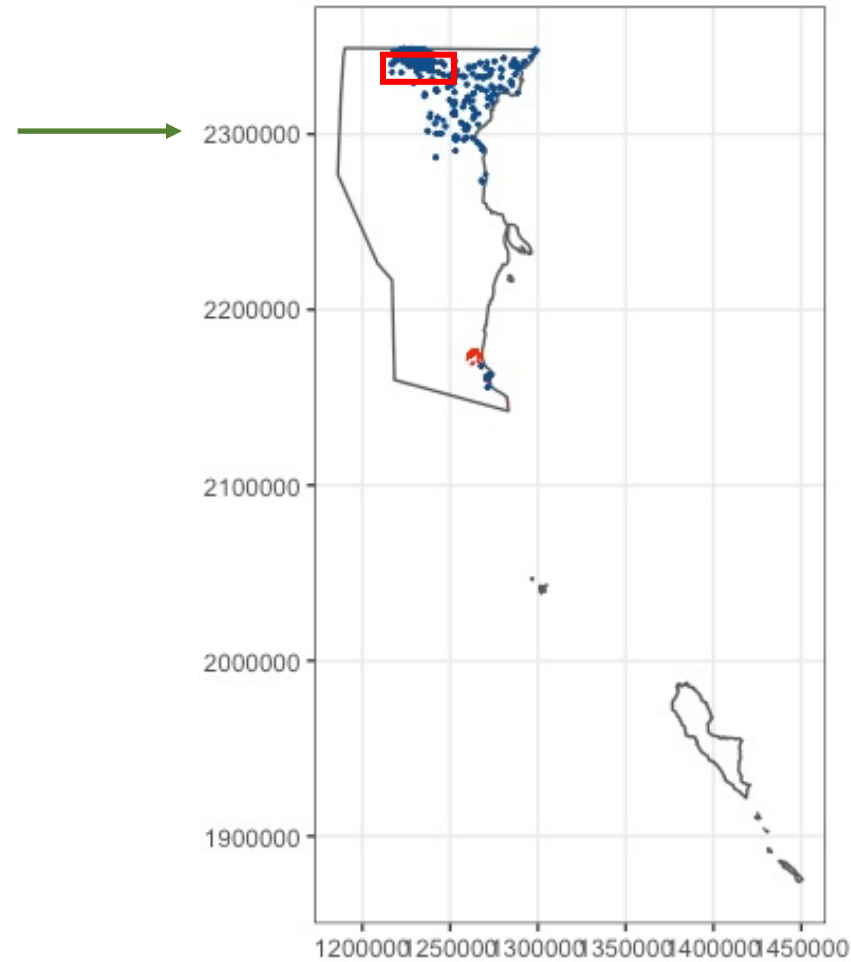
geom\_sf(data = mxc\_serv\_punt, size = 0.3, color = "dodgerblue4") +

theme\_bw()

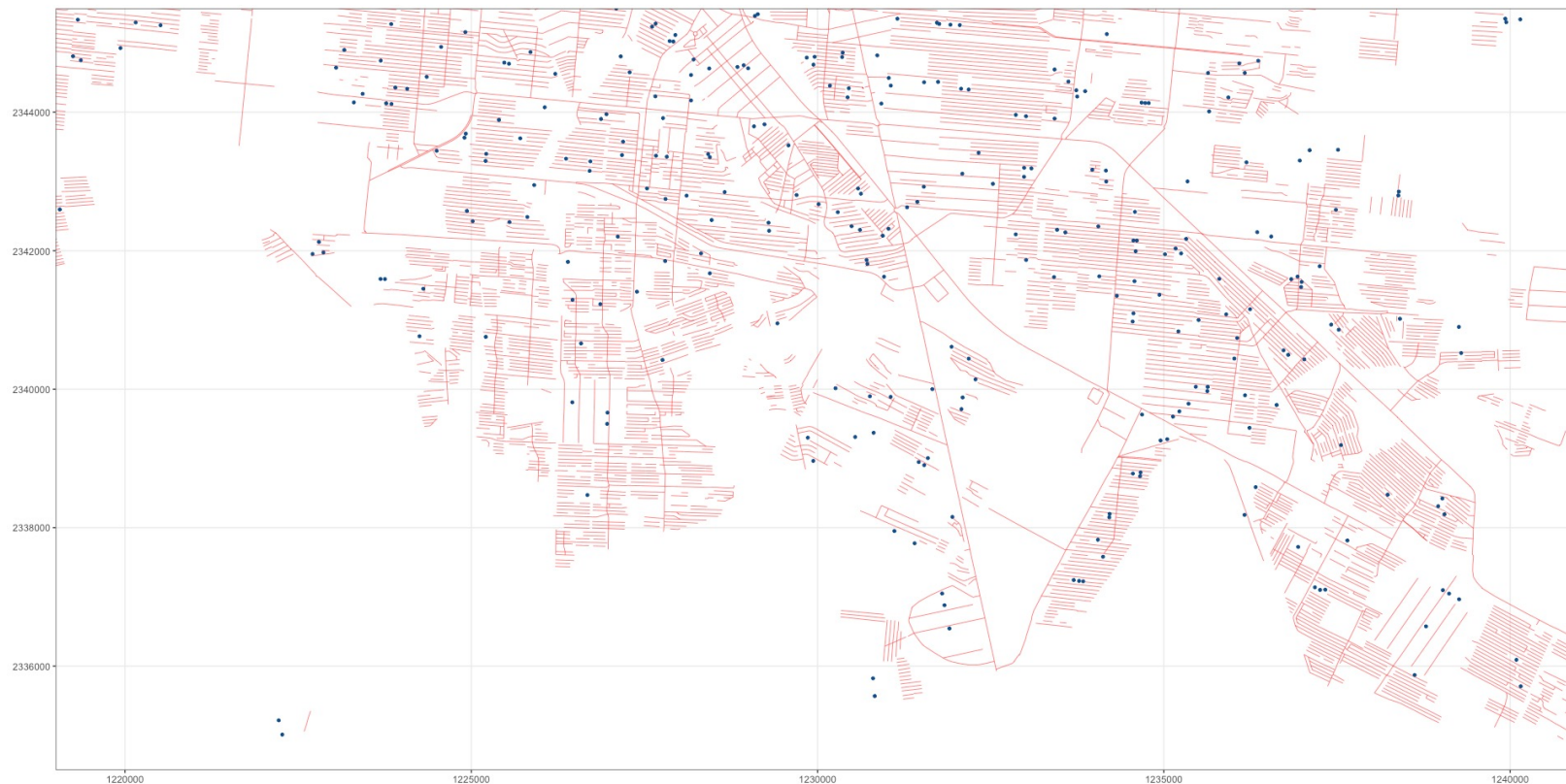
```
ggplot() +  
  geom_sf(data = mxc_mun, fill = "transparent") +  
  geom_sf(data = mxc_vial, colour = "red", size = 0.2) +  
  geom_sf(data = mxc_serv_punt, size = 0.3, color = "dodgerblue4") +  
  theme_bw()
```



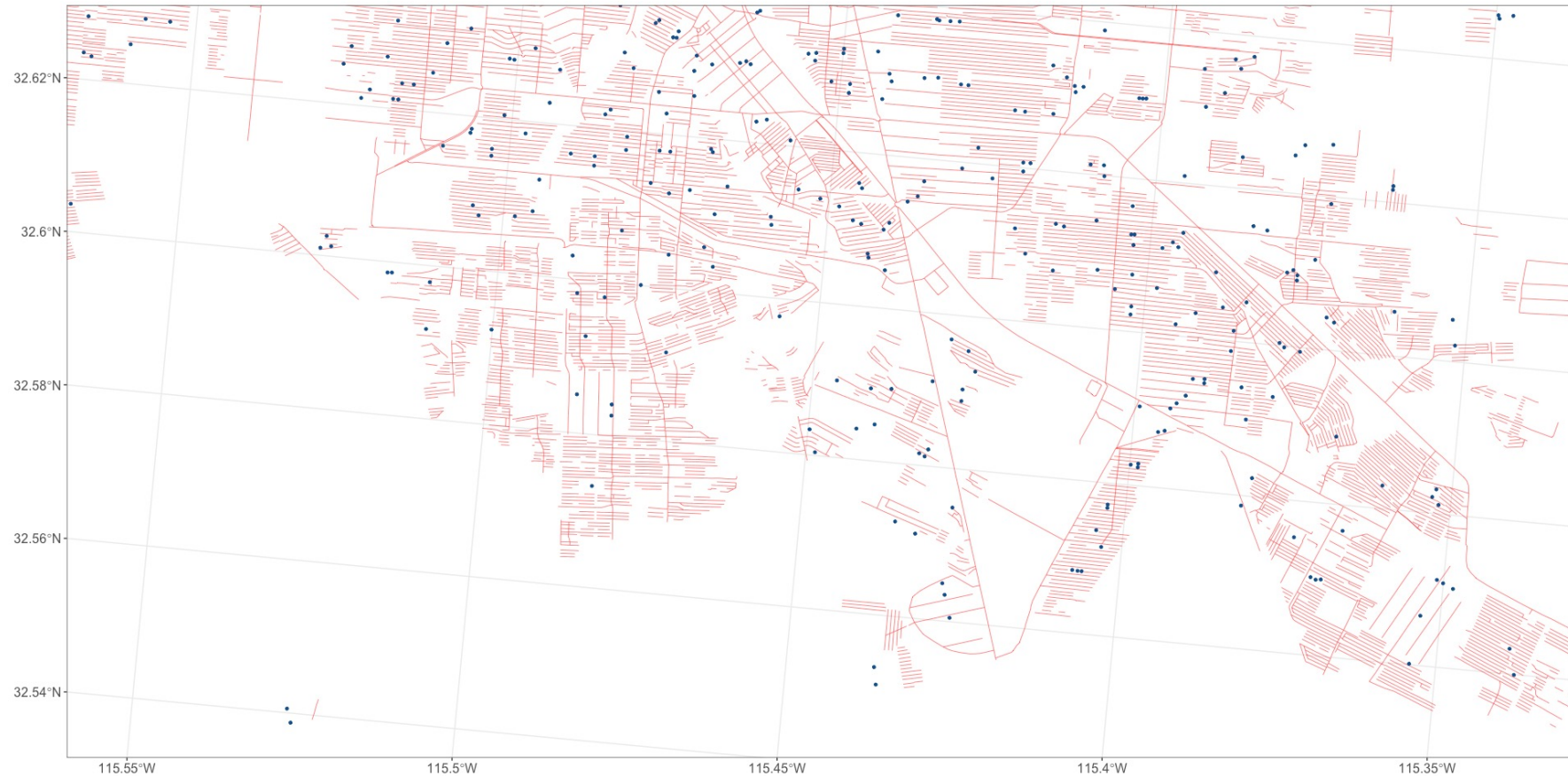
```
ggplot() +  
  geom_sf(data = mxc_mun, fill = "transparent") +  
  geom_sf(data = mxc_eje_vial, colour = "red", size = 0.2) +  
  geom_sf(data = mxc_servicios_puntual, size = 0.3, color = "dodgerblue4") +  
  coord_sf(datum = st_crs(mxc_mun)) +  
  theme_bw()
```



```
ggplot() +  
  geom_sf(data = mxc_mun, fill = "transparent") +  
  geom_sf(data = mxc_vial, colour = "red", size = 0.2) +  
  geom_sf(data = mxc_serv_punt, size = 0.3, color = "dodgerblue4")  
  coord_sf(datum = st_crs(mxc_mun),  
    xlim = c(1220000, 1240000),  
    ylim = c(2335000, 2345000)) +  
  theme_bw()
```



```
ggplot() +  
  geom_sf(data = mxc_mun, fill = "transparent") +  
  geom_sf(data = mxc_eje_vial, colour = "red", size = 0.2) +  
  geom_sf(data = mxc_servicios_puntual, size = 0.3, color = "dodgerblue4")  
  coord_sf(xlim = c(1220000, 1240000),  
            ylim = c(2335000, 2345000)) +  
  theme_bw()
```



# Exportar *shapefile*

```
st_write(objeto, "ruta/nombre_archivo.shp")
```

```
st_write(tij_mun, "./capas/tij_mun.shp")
```

Exporte la capa de ageb de tijuana "tij\_ageb"