

6. Data visualization and decision making

Ana J. Alegre (jalegre@centrogeo.edu.mx), Cristian Silva (csilva@centrogeo.edu.mx)

11/4/2021

Introducción

La visualización de datos hace posible comunicar lo que significa la información en una manera clara a otras personas. Mediante el uso de contextos visuales, el aprendizaje a partir de los resultados se vuelve útil para llamar a la acción y hacer toma de decisiones basada en datos. Existen paquetes diferentes en R que nos permiten construir diferentes tipos de visualizaciones como gráficas, diagramas o mapas en una manera eficiente y simple.

Objetivo

En esta sección, construiremos algunas visualizaciones a partir de los datos preprocesados en las secciones previas y estos serán interpretados para entender el fenómeno de la violencia en México y proponer algunas posibles acciones generales para atender la problemática de los homicidios que establecimos como el objetivo principal al principio de este taller.

Preparación de datos

Primeramente, elimina todos los objetos del entorno:

```
rm(list = ls())
```

Carga los paquete que serán usados en esta sesión:

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5      v purrr  0.3.4
```

```
## v tibble  3.1.4      v dplyr  1.0.7
```

```
## v tidyr   1.1.3      v stringr 1.4.0
```

```
## v readr   1.4.0      v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()    masks stats::lag()
```

```
library(sf)
```

```
## Linking to GEOS 3.8.1, GDAL 3.2.1, PROJ 7.2.1
```

```
library(raster)
```

```
## Loading required package: sp
```

```
##
```

```
## Attaching package: 'raster'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      select
```

```
## The following object is masked from 'package:tidyr':
```

```
##
```

```
##      extract
```

```
library(DT)
```

```
library(tmap)
```

```
library(leaflet)
```

Lee el conjunto de datos que se preparó anteriormente con los homicidios por estado y por mes (homicidios_por_estado.csv):

```
homicidios_por_estado <-  
  read_csv("Data/murders_by_state.csv") %>%  
  glimpse()
```

```
##
```

```
## -- Column specification -----
```

```
## cols(
```

```
##   clave_ent = col_double(),
```

```
##   entidad = col_character(),
```

```
##   mes = col_date(format = ""),
```

```
##   homicidios = col_double()
```

```
## )
```

```
## Rows: 608
```

```
## Columns: 4
```

```
## $ clave_ent   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,~
```

```
## $ entidad     <chr> "Aguascalientes", "Aguascalientes", "Aguascalientes", "Aguas~
```

```
## $ mes         <date> 2018-12-01, 2019-01-01, 2019-02-01, 2019-03-01, 2019-04-01~
```

```
## $ homicidios  <dbl> 25, 28, 24, 22, 25, 29, 20, 15, 13, 26, 23, 21, 23, 15, 16,~
```

Línea de tendencia

Para visualizar la tendencia nacional de homicidios, agrupa el conjunto de datos `homicidios_por_estado` por mes y sumaliza todos los homicidios:

```
homicidios_por_mes <-
  homicidios_por_estado %>%
  group_by(mes) %>%
  summarize(homicidios_total = sum(homicidios)) %>%
  ungroup()
```

Visualiza la serie:

```
homicidios_por_mes
```

```
## # A tibble: 19 x 2
##   mes      homicidios_total
##   <date>          <dbl>
## 1 2018-12-01          3942
## 2 2019-01-01          3704
## 3 2019-02-01          3597
## 4 2019-03-01          3805
## 5 2019-04-01          3667
## 6 2019-05-01          3921
## 7 2019-06-01          3780
## 8 2019-07-01          3767
## 9 2019-08-01          3692
## 10 2019-09-01          3692
## 11 2019-10-01          3686
## 12 2019-11-01          3698
## 13 2019-12-01          3852
## 14 2020-01-01          3581
## 15 2020-02-01          3468
## 16 2020-03-01          4004
## 17 2020-04-01          3585
## 18 2020-05-01          3424
## 19 2020-06-01          3452
```

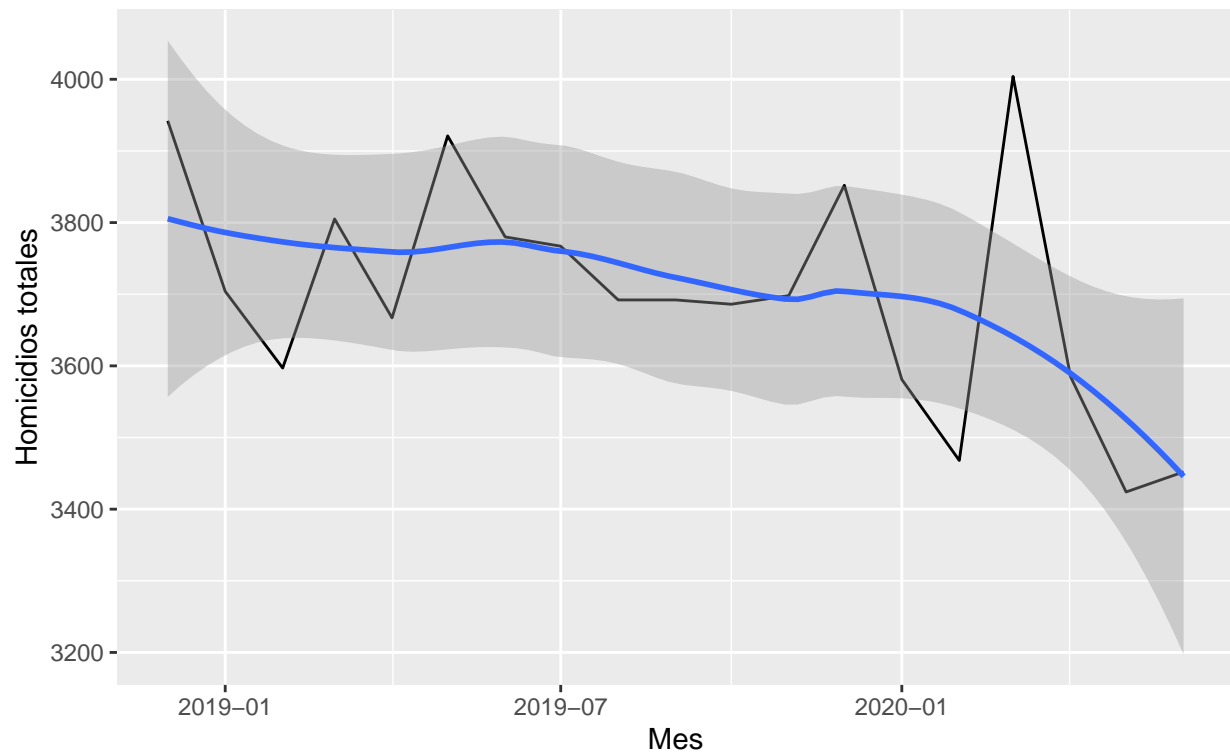
Construye una gráfica de línea con el conjunto de datos `homicidios_por_mes` usando `ggplot2`:

```
homicidios_por_mes %>%
  ggplot(aes(x = mes, y = homicidios_total)) + # Crea una gráfica vacía con el mes en x y los homicidios en y
  geom_line() + # Agrega una línea de tendencia
  geom_smooth() + # Agregar una línea de regresión
  labs(title = "Tendencia de homicidios en México",
        subtitle = "Diciembre 2018 a Junio 2021",
        x = "Mes",
        y = "Homicidios totales") # Agrega los títulos y etiquetas de ejes
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

Tendencia de homicidios en México

Diciembre 2018 a Junio 2021



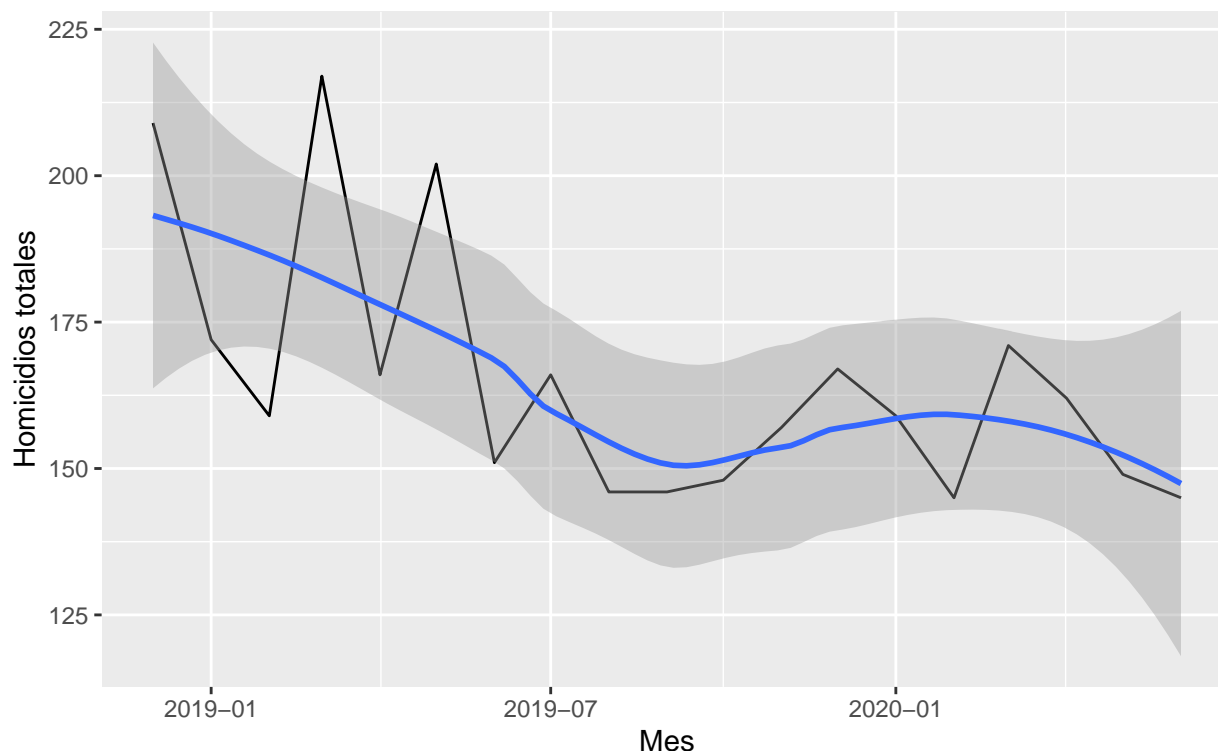
Repita los pasos anteriores para crear una línea de tendencia, usando ahora sólo los datos de la CDMX para compararlos con la tendencia nacional. Esta vez, debido al filtro, no se requiere agrupar los datos y es posible codificar todo el proceso en un sólo bloque usando *pipas*:

```
homicidios_por_estado %>%
  filter(entidad == "Ciudad de México") %>% # Filtra los datos para CDMX
  ggplot(aes(x = mes, y = homicidios)) + # Crea una gráfica vacía con el mes en x y los homicidios en y
  geom_line() + # Agrega una línea de tendencia
  geom_smooth() + # Agregar una línea de regresión
  labs(title = "Tendencia de homicidios en la CDMX",
        subtitle = "Diciembre 2018 a Junio 2021",
        x = "Mes",
        y = "Homicidios totales") # Add titles and axis labels
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

Tendencia de homicidios en la CDMX

Diciembre 2018 a Junio 2021



Como puede observarse en ambas gráficas, durante el periodo de diciembre 2018 a junio 2021, la tendencia de homicidios a nivel nacional ha ido en descenso durante los meses recientes. Igualmente en la CDMX, han descendido durante el periodo con algunos incrementos durante los primeros meses del año.

Mapa de coropletas

Los mapas de coropletas se usan para visualizar de manera sencilla las variaciones en cantidades al colorear polígonos. En este caso, este tipo de mapas pueden ser útiles para comparar las tasas de homicidios por cada 10,000 habitantes para cada estado en México. Para construir un mapa de coropletas, usa la capa de estados con datos de delitos asociados (`state_offenses.gpkg`), que se creó anteriormente en la sección 2 del taller.

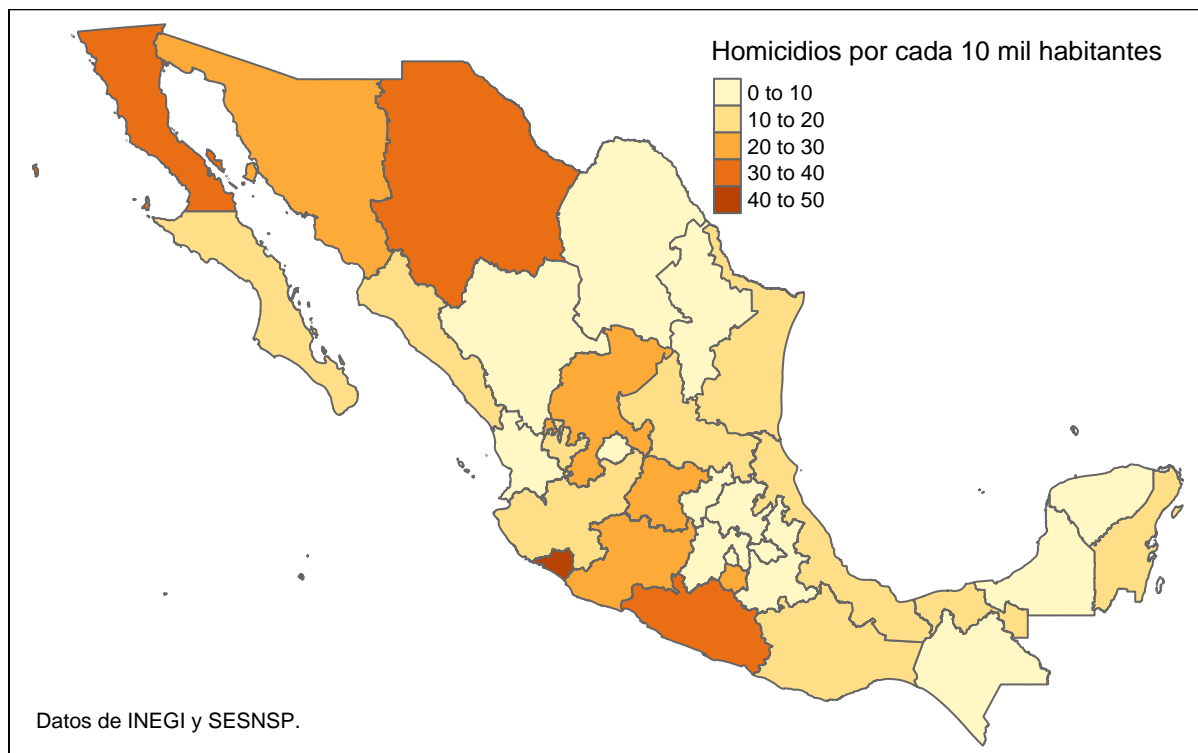
```
estados_delitos <-  
  st_read("Data/states_offenses.gpkg") %>%  
  glimpse()  
  
## Reading layer `states_offenses' from data source  
##   `/Users/cristiansilva/Documents/Centro Geo/Semana de divulgación 2021/taller-analisis-geo-r/Data/s  
##   using driver `GPKG'  
## Simple feature collection with 32 features and 10 fields  
## Geometry type: MULTIPOLYGON  
## Dimension:      XY  
## Bounding box:   xmin: -118.3651 ymin: 14.5321 xmax: -86.71468 ymax: 32.71865  
## Geodetic CRS:   WGS 84  
## Rows: 32
```

```
## Columns: 11
## $ clave_ent      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
## $ entidad       <chr> "Aguascalientes", "Baja California", "Baja California Su~
## $ poblacion     <dbl> 1425607, 3769020, 798447, 928363, 3146771, 731391, 55438~
## $ extorsion     <dbl> 637, 1171, 935, 138, 204, 366, 960, 96, 3743, 524, 166, ~
## $ homicidio     <dbl> 436, 13761, 1313, 464, 1443, 3474, 3232, 11257, 7210, 12~
## $ secuestro     <dbl> 48, 88, 17, 32, 91, 59, 109, 94, 689, 40, 55, 350, 143, ~
## $ ext_10khab    <dbl> 4.4682721, 3.1069084, 11.7102325, 1.4864875, 0.6482836, ~
## $ sec_10khab    <dbl> 0.33669868, 0.23348244, 0.21291332, 0.34469275, 0.289185~
## $ hom_10khab    <dbl> 3.058346, 36.510817, 16.444423, 4.998045, 4.585653, 47.4~
## $ nom_abreviado <chr> "AGS", "BC", "BCS", "CAMP", "COAH", "COL", "CHIS", "CHIH~
## $ geom          <MULTIPOLYGON [°]> MULTIPOLYGON (((-102.0756 2..., MULTIPOLYGO~
```

A continuación, construye el mapa de coropletas con los resultados y compara las tasas de homicidio entre estados, usando el paquete `tmap`:

```
tm_shape(estados_delitos) + # Crea un mapa vacío
  tm_polygons("hom_10khab",
    title = "Homicidios por cada 10 mil habitantes") + # Agrega la capa de polígonos de estad
tm_layout(main.title = "Estados más violentos en México, Dic, 2018 a Jun. 2021") + # Agrega un títul
tm_credits("Datos de INEGI y SESNSP.",
  position = c("left", "bottom")) # Agrega las fuentes de datos
```

Estados más violentos en México, Dic, 2018 a Jun. 2021



Mapa hexagonal de coropletas

Hagamos una variante del mapa de coropletas, usando ahora una cuadrícula hexagonal *hexgrid* para ayudar a comparar visualmente los estados usando hexágonos del mismo tamaño, y representando una versión simplificada del mapa.

Usaremos los paquetes `rmapshaper` y `geogrid` para construir el *hexgrid*, pero antes será necesario eliminar todas las islas de la capa de polígonos de los estados para asegurarnos de que sólo tendremos estados contiguos que se representarán como polígonos. Para eliminar las islas de la capa `estados_delitos`, usa la función `ms_filter_islands` para eliminar cada isla con menos de 1,500 vértices por ejemplo:

```
library(rmapshaper)

## Registered S3 method overwritten by 'geojsonlint':
##   method      from
##   print.location dplyr
```

```
library(geogrid)

estados_sin_islas <-
  estados_delitos %>%
  ms_filter_islands(min_vertices = 1500)

# Previsualiza la capa de estados sin islas:
qtm(estados_sin_islas)
```



A continuación, usa la función `calculate_grid` con algunos valores diferentes de *semilla* para generar algunos posibles arreglos hexagonales y visualízalos con la función `plot` para escoger el más apropiado:

```
# Define algunos parámetros para graficar las 10 pruebas en una cuadrícula de 2 filas por 5 columnas
par(mfrow = c(2, 5), mar = c(0, 0, 2, 0))
```

```
# Itera la semilla para construir los posibles hexgrids:
for (i in 1:10) {
  new_cells <- calculate_grid(shape = estados_sin_islas, grid_type = "hexagonal", seed = i)
  plot(new_cells, main = paste("Seed", i, sep = " "))
}
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

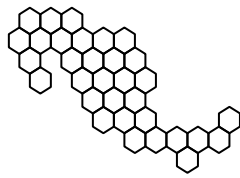
[illegible]


```
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
## Warning in proj4string(obj): CRS object has comment, which is lost in output
```

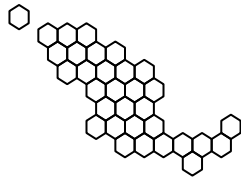
Seed 1



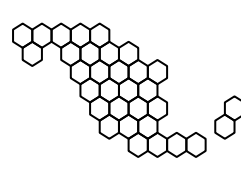
Seed 2



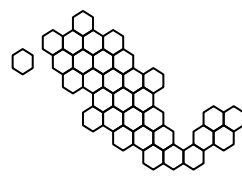
Seed 3



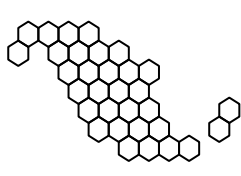
Seed 4



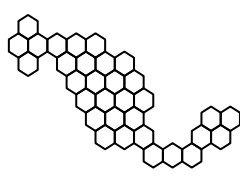
Seed 5



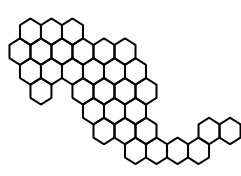
Seed 6



Seed 7



Seed 8



Seed 9



Seed 10



Aparecen algunas advertencias relacionadas al uso de una versión anterior de las librerías PROJ que usa el paquete, ignóralas y continúa.

Aparentemente, el hexgrid que mejor se ajusta a la forma de los estados en México es la que se construyó con la **semilla 7**. Calcula nuevamente el hexgrid con `seed = 7` y usa la función para combinarle los valores de `estados_delitos`, manteniendo solamente las columnas `clave_ent`, `nom_abreviado` y `hom_10khab`:

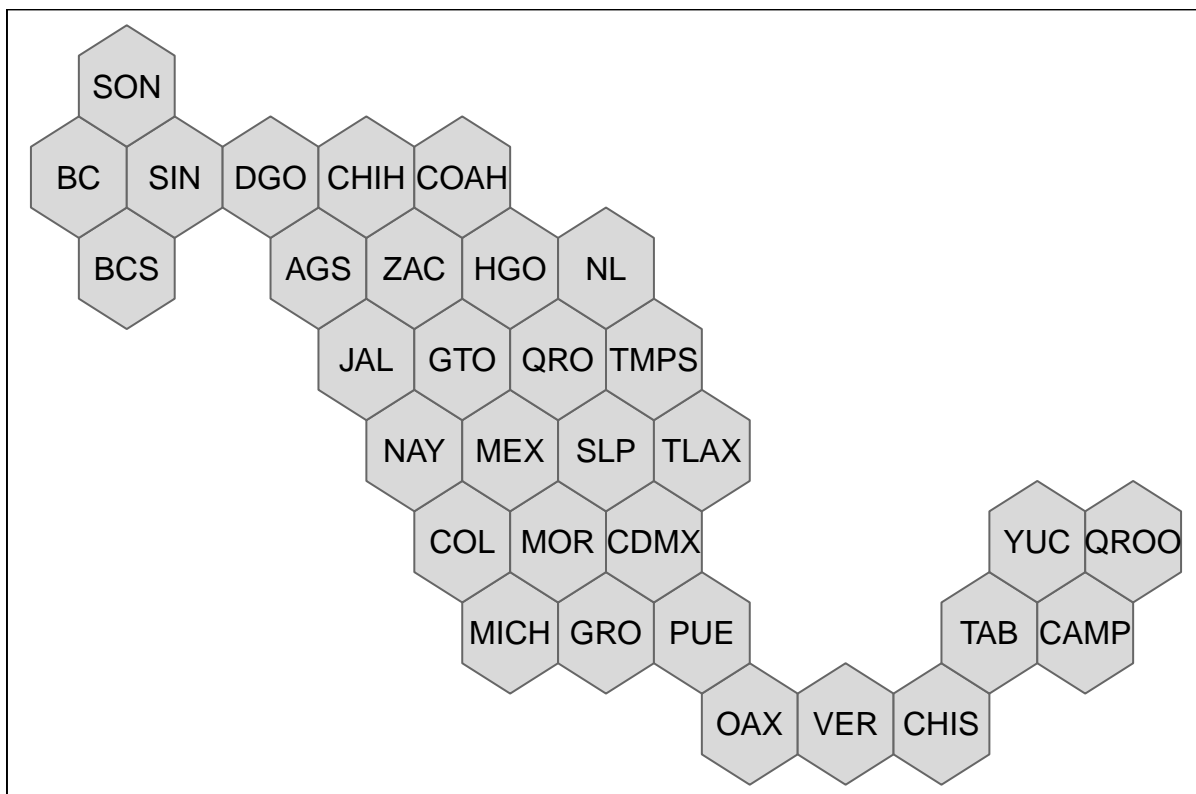
```
# Crea el hexgrid con la mejor semilla:
estados_hex_celdas <- calculate_grid(shape = estados_sin_islas, grid_type = "hexagonal", seed = 7)

# Assign original layer variables to it:
estados_hex <-
  assign_polygons(estados_sin_islas, estados_hex_celdas) %>%
  dplyr::select(clave_ent, nom_abreviado, hom_10khab) %>% # Warning: both raster and tidyverse have a '
  glimpse()
```

```
## Rows: 32
## Columns: 4
## $ clave_ent      <dbl> 20, 30, 7, 16, 12, 21, 27, 4, 6, 17, 9, 31, 23, 18, 15, ~
## $ nom_abreviado  <chr> "OAX", "VER", "CHIS", "MICH", "GRO", "PUE", "TAB", "CAMP~
## $ hom_10khab     <dbl> 12.136545, 10.319279, 5.829907, 20.057505, 34.589352, 8.~
## $ geometry      <POLYGON [°]> POLYGON ((-98.41321 17.4134..., POLYGON ((-95.87~
```

Previsualiza la capa resultante:

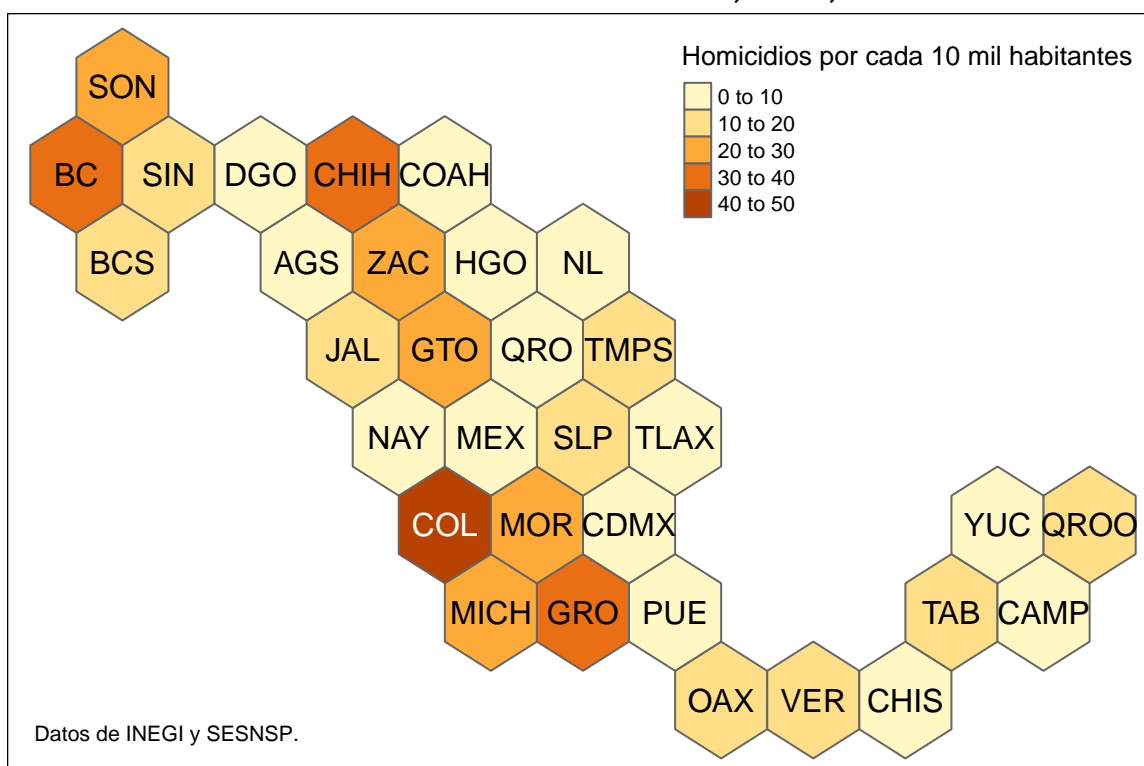
```
qtm(estados_hex) +
  tm_text("nom_abreviado") # Agrega una capa de texto para etiquetar con los valores de 'nom_abreviado'
```



El hexgrid parece aceptable para representar los estados, ahora construye el mapa de coropletas igual que el primero, pero cambiando la capa original para usar los hexgrid para identificar aquellos con las tasas más altas en el periodo de tiempo definido:

```
tm_shape(estados_hex) +
  tm_polygons("hom_10khab",
              title = "Homicidios por cada 10 mil habitantes") +
  tm_text("nom_abreviado") +
  tm_layout(main.title = "Estados más violentos en México, Dic, 2018 a Jun. 2021") +
  tm_credits("Datos de INEGI y SESNSP.",
            position = c("left", "bottom"))
```

Estados más violentos en México, Dic, 2018 a Jun. 2021



Este tipo de mapas son más legibles y hacen más fácil la comparación de las tasas de homicidios sólo mediante el color y no por el tamaño del polígono, haciéndolo una versión simplificada el mapa de coropletas original.

Mapa de burbujas combinado con coropletas

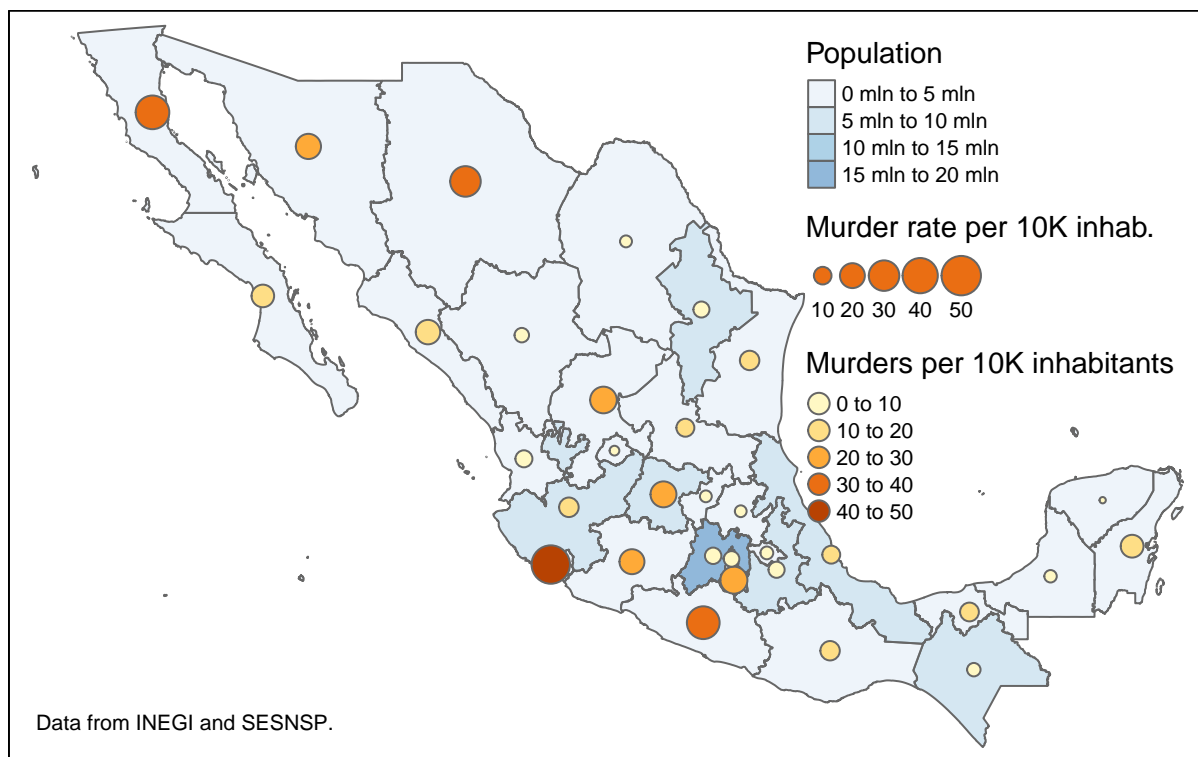
Otra manera de visualizar las proporciones de una variable en un mapa es usando un *mapa de burbujas*. Este tipo de mapas son útiles para comparar entre cantidades usando círculos, donde su tamaño y color representan valores altos y bajos. En este ejemplo visualizaremos los estados con las mayores tasas de homicidios usando burbujas y adicionalmente usaremos coropletas para visualizar los estados con mayor población para encontrar relaciones entre estas variables.

```
tmap_mode("plot") # Cambia el modo de mapa a interactivo
```

```
## tmap mode set to plotting
```

```
tm_shape(estados_delitos) +
  tm_polygons("poblacion",
    palette = "Blues",
    alpha = 0.5,
    title = "Population") +
  tm_bubbles("hom_10khab",
    col = "hom_10khab",
    title.size = "Murder rate per 10K inhab.",
    title.col = "Murders per 10K inhabitants") +
  tm_layout(main.title = "Most violent states in Mexico, Dec. 2018 to Jun. 2021") +
  tm_credits("Data from INEGI and SESNSP.",
    position = c("left", "bottom"))
```

Most violent states in Mexico, Dec. 2018 to Jun. 2021



El mapa anterior es un ejemplo de cómo dos visualizaciones diferentes pueden combinarse en el mismo mapa para relacionar diferentes tipos de datos.

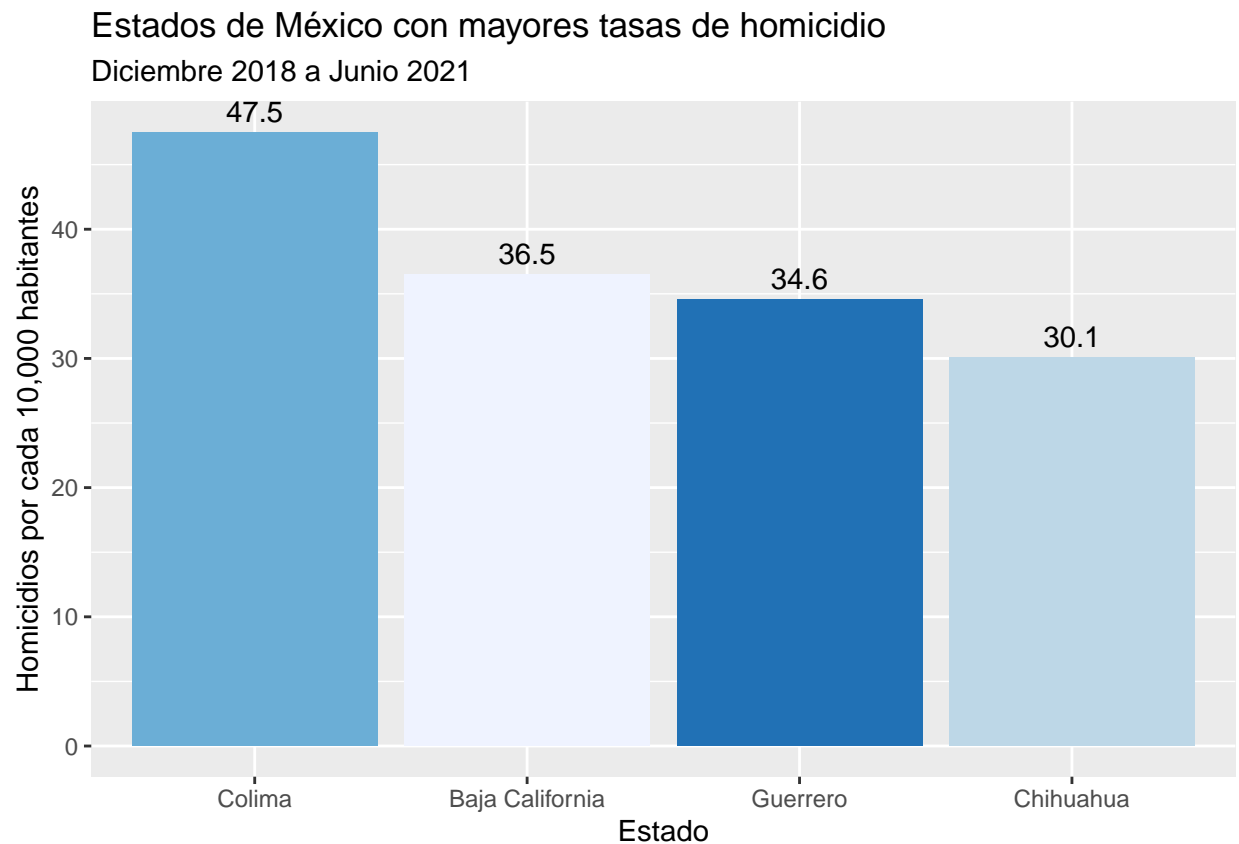
Gráfico de columnas

Para complementar los mapas de coropletas, crearemos una gráfica de columnas con los estados con las tasas más altas de homicidios por cada 10,000 habitantes durante el periodo, que están representados en el mapa por los colores naranja y rojo (tasas de homicidios mayores a 30). Para esto, filtra los estados con `hom_10khab >= 30` y ordénalos en orden descendente:


```

estados_delitos %>%
  as_tibble() %>% # Convierte de sf a tibble para eliminar las geometrías
  dplyr::select(entidad, hom_10khab) %>% # Selecciona las variables que se van a mostrar
  filter(hom_10khab >= 30) %>% # Conserva los estados con una tasa mayor a 30
  ggplot(aes(x = fct_reorder(entidad, desc(hom_10khab)), y = hom_10khab, fill = entidad)) + # Crea una
  geom_col(show.legend = F) + # Agrega las columnas, sin mostrar la leyenda
  scale_fill_brewer(palette = "Blues") +
  geom_text(aes(label = round(hom_10khab, digits = 1)), vjust = -0.5) + # Agrega las etiquetas encima d
  labs(title = "Estados de México con mayores tasas de homicidio",
        subtitle = "Diciembre 2018 a Junio 2021",
        x = "Estado",
        y = "Homicidios por cada 10,000 habitantes") # Agrega los nombre de títulos y ejes

```



Visualización como tabla de datos

La CDMX no aparece en la gráfica anterior, así que será necesario saber en qué lugar de tasa de homicidio por cada 10,000 habitantes se encuentra en el país. Para ello, es posible construir una tabla interactiva usando el paquete DT que nos permite formatear las filas y columnas para enfatizar los hallazgos importantes y que nos permite reordenar los datos usando cualquiera de las variables disponibles.

```

estados_delitos %>%
  as_tibble() %>% # Convierte de sf a tibble, esto elimina las geometrías
  mutate(participacion = homicidio/sum(homicidio),
         ranking = dense_rank(desc(hom_10khab))) %>%

```

```

arrange(ranking) %>%
dplyr::select(ranking, entidad, homicidio, hom_10khab, participacion) %>%
datatable(colnames = c("Ranking", "State", "Murders total", "Murders rate per 10K inhabitants", "Perce
rownames = FALSE) %>%
formatRound(columns = "homicidio",
digits = 0) %>%
formatRound(columns = "hom_10khab",
digits = 2) %>%
formatPercentage(columns = "participacion",
digits = 1) %>%
formatStyle(columns = "participacion",
background = styleColorBar(data = 0:1, color = "steelblue")) %>%
formatStyle(columns = "ranking",
target = "row",
fontWeight = styleEqual(levels = 1:5, values = rep("bold", 5))) %>%
formatStyle(columns = "entidad",
target = "row",
fontWeight = styleEqual(levels = "Ciudad de México", values = "bold"),
backgroundColor = styleEqual(levels = "Ciudad de México", values = "lightblue"))

```

Usando la tabla interactiva anterior es posible ordenar los datos usando cualquier variable y buscar un estado específico, en este caso podemos observar que los primeros 5 estados con las tasas más altas de homicidio por cada 10,000 habitantes son Colima, Baja California, Quintana Roo, Guanajuato and Chihuahua, mientras que la CDMX se ubica en el sitio #21, pero si los datos se ordenan por homicidios totales los primeros lugares son México, Baja California and Guanajuato, con la CDMX dentro de los primeros 10 lugares.

Clustering de puntos simple

Ahora que aprendimos sobre los homicidios a nivel nacional, será necesario ir a detalle con los datos para la CDMX. Para encontrar patrones que puedan ser útiles para aprender más acerca del problema de los homicidios en la ciudad, se pueden construir mapas con mayor detalle gracias a que los registros están georeferenciados.

Carga la capa `murders_cdmx.gpkg` que se procesó anteriormente y que está en la carpeta `Data`:

```

homicidios_cdmx <-
  st_read("Data/murders_cdmx.gpkg") %>%
  glimpse()

## Reading layer `murders_cdmx' from data source
##   `/Users/cristiansilva/Documents/Centro Geo/Semana de divulgación 2021/taller-analisis-geo-r/Data/m
##   using driver `GPKG'
## Simple feature collection with 2257 features and 20 fields (with 264 geometries empty)
## Geometry type: POINT
## Dimension:      XY
## Bounding box:   xmin: -99.33316 ymin: 19.13077 xmax: -98.95981 ymax: 19.57635
## Geodetic CRS:   WGS 84
## Rows: 2,257
## Columns: 21
## $ ao_hechos      <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2~
## $ mes_hechos     <chr> "Mayo", "Mayo", "Mayo", "Mayo", "Mayo", "Mayo", "Mayo", "~
## $ fecha_hechos   <dtm> 2019-05-17 19:00:00, 2019-05-17 19:30:00, 2019-0~

```

```
## $ ao_inicio      <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2~
## $ mes_inicio     <chr> "Mayo", "Mayo", "Mayo", "Mayo", "Mayo", "Mayo", "~
## $ fecha_inicio   <dtm> 2019-05-17 20:26:53, 2019-05-17 23:40:11, 2019-0~
## $ delito         <chr> "HOMICIDIO POR ARMA BLANCA", "HOMICIDIOS INTENCIO~
## $ fiscalia       <chr> "INVESTIGACIÓN EN GUSTAVO A. MADERO", "INVESTIGAC~
## $ agencia        <chr> "GAM-4", "VC-3", "IZC-2", "AO-1", "AO-1", "COY-2"~
## $ unidad_investigacion <chr> "UI-2CD", "UI-1SD", "UI-3SD", "UI-3SD", "UI-3SD",~
## $ categoria_delito <chr> "HOMICIDIO DOLOSO", "HOMICIDIO DOLOSO", "HOMICIDI~
## $ calle_hechos    <chr> "PROLONGACION MISTERIOS ESQUINA CON AVENIDA ACU~
## $ calle_hechos2   <chr> "AVENIDA INSURGENTES NORTE", NA, "TALLERES GRAFIC~
## $ colonia_hechos  <chr> "TEPEYAC INSURGENTES", "MERCED BALBUENA", "AGRÍCO~
## $ alcaldia_hechos <chr> "GUSTAVO A MADERO", "VENUSTIANO CARRANZA", "IZTAC~
## $ competencia    <lg1> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ longitud        <dbl> -99.11739, -99.12231, -99.06589, -99.19430, -99.2~
## $ latitud         <dbl> 19.48924, 19.42010, 19.42068, 19.38319, 19.37849,~
## $ tempo          <lg1> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ fecha          <date> 2019-05-18, 2019-05-18, 2019-05-18, 2019-05-18, ~
## $ geom           <POINT [°]> POINT (-99.11739 19.48924), POINT (-99.1223~
```

Visualiza los puntos geográficos en un mapa:

```
tmap_mode("view") # Configura el mapa como interactivo
```

```
## tmap mode set to interactive viewing
```

```
qtm(homicidios_cdmx) # Mapa rápido con puntos de homicidios
```

```
## Warning: The shape homicidios_cdmx contains empty units.
```

Leaflet es una librería de javascript que se usa para construir mapas web interactivos que también puede usarse en R, de hecho, los mapas interactivos que el paquete `tmap` construye la utilizan. *Leaflet* nos ofrece otra alternativa para construir mapas directamente en R usando el paquete `leaflet`.

Muestra los puntos de homicidios en un mapa, usando ahora el paquete `leaflet`:

```
library(leaflet) # Carga el paquete leaflet
```

```
leaflet(homicidios_cdmx) %>% # Crea un mapa vacío con la capa de homicidios
  addTiles() %>% # Agrega un mapa base, por default usa OpenStreetMap
  addCircles() # Agrega una capa de puntos como círculos
```

```
## Warning in validateCoords(lng, lat, funcName): Data contains 264 rows with
## either missing or invalid lat/lon values and will be ignored
```

```
# De manera alternativa, se pueden usar marcadores en vez de círculos usando la función addMarkers, no
# leaflet(homicidios_cdmx) %>% # Crea un mapa vacío con la capa de homicidios
#   addTiles() %>% # Agrega un mapa base, por default usa OpenStreetMap
#   addMarkers() # Agrega una capa de puntos como círculos
```

Leaflet puede ser usado para agrupar puntos de manera sencilla, repite el ejemplo anterior pero usando ahora el parámetro `clusterOptions = markerClusterOptions()` de la función `addMarkers` en vez de la función `addCircle`:

```
leaflet(homicidios_cdmx) %>% # Crea un mapa vacío con la capa de homicidios
  addTiles() %>% # Agrega un mapa base, por default usa OpenStreetMap
  addMarkers(label = ~fecha_hechos, # Muestra la fecha al pasar el mouse por un punto
    popup = ~fecha_hechos, # Muestra la fecha al dar clic a un punto
    clusterOptions = markerClusterOptions()) # Construye los clusters de puntos
```

```
## Warning in validateCoords(lng, lat, funcName): Data contains 264 rows with
## either missing or invalid lat/lon values and will be ignored
```

Estos mapas de clusters interactivos son útiles para encontrar áreas específicas en la ciudad donde los homicidios ocurren más frecuentemente, los clusters de puntos son representados como círculos con el número total de puntos representados, coloreados desde los más bajos hasta los más altos y ubicados en el centroide común entre ellos. Es posible navegar y expandir los clusters al hacer acercamientos para identificar las calles y áreas con mayor incidencia de homicidios.

Mapa de calor (hotspots)

Los mapas de calor usan una escala de color para contrastar las zonas con valores más bajos (fríos) y altos (calientes). En este ejemplo usaremos el archivo raster con la estimación de densidad de Kernel (KDE) que se construyó en la sección 4 para construir un mapa para visualizar las áreas con mayor densidad de incidencia de homicidios en la ciudad.

Lee el archivo raster usando la función `raster`:

```
murders_raster <- raster("Data/murders_kde.tif")
```

```
## Warning in showSRID(SRS_string, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown_based_on_GRS80_ellipsoid in Proj4
## definition
```

```
# Previsualizar el contenido del raster:
murders_raster
```

```
## class      : RasterLayer
## dimensions : 128, 128, 16384  (nrow, ncol, ncell)
## resolution : 348.5856, 470.7479  (x, y)
## extent     : 461655.7, 506274.7, 2106166, 2166422  (xmin, xmax, ymin, ymax)
## crs        : +proj=utm +zone=14 +ellps=GRS80 +units=m +no_defs
## source     : murders_kde.tif
## names      : murders_kde
## values     : -5.02363e-22, 1.267477e-05  (min, max)
```

Visualizar la capa raster en un mapa usando el paquete `tmap`:

```
tm_shape(murders_raster) +
  tm_basemap() +
  tm_raster(palette = "YlOrRd",
    alpha = 0.5)
```

El mapa resultante muestra varias zonas de alta concentración en rojo. Los valores en la escala de colores representan la densidad de homicidios por unidad de área de la cuadrícula.

Conclusiones

1. Hay ocho estados con el mayor número de homicidios: Guerrero, Baja California, Michoacán, Jalisco, Guanajuato, Estado de México, Veracruz y Chihuahua.
2. De acuerdo con los delitos que pueden asociarse al crimen organizado (homicidios, extorsiones y secuestros), hay estados que tienen algunas similitudes, especialmente en la frontera norte, el Pacífico y algunos estados adyacentes al centro como San Luis Potosí y Puebla. Es posible que cada uno de los grupos de estados similares requieran una estrategia de seguridad particular.
3. Este tipo de estudios pueden extenderse incorporando más variables, y se puede sugerir una coordinación entre los estados que comparten fronteras para la implementación conjunta de acciones.
4. La CDMX presenta un patrón de acumulación de homicidios en el centro de la ciudad. La distancia del primer vecino más cercano entre homicidios es de 265 metros.
5. Las zonas este y norte de la ciudad también muestran patrones de gran acumulación, por lo que se puede sugerir un refuerzo en la seguridad de estas áreas en particular.
6. En estudios subsecuentes, se pueden realizar estudios de correlación y agendas de eventos enfocados en programas de prevención.

Referencias

- Tennekes, M., Nowosad, J. (2018). *tmap: Thematic Maps in R*. Journal of Statistical Software. Retrieved September 8, 2021, from https://www.researchgate.net/publication/324652152_tmap_Thematic_Maps_in_R/fulltext/5ad9e7eb0f7e9b28593cf867/tmap-Thematic-Maps-in-R.pdf
- Wickham, H. (2021). *ggplot2: Elegant Graphics for Data Analysis*. Recuperado el 8 de octubre, 2021, desde <https://ggplot2-book.org/index.html>
- Tennekes, M., Nowosad, J. (2021). *Elegant and informative maps with tmap*. Recuperado el 8 de Septiembre, 2021, desde <https://r-tmap.github.io/tmap-book/>
- *DT: An R interface to the DataTables library*. Recuperado el 8 de Septiembre, 2021, desde <https://rstudio.github.io/DT/>
- *Algorithmic tessellation with geogrid*. Recuperado el 15 de Octubre, 2021, desde <https://github.com/jbaileyh/geogrid>