



# **Dia 2. Introducción al R básico**

Acceso a datos en tablas.

Funciones básicas. Estructuras de control.

**Visualización de datos geográficos en R.**

---

Universidad de Guadalajara, CUCSH–CUCBA

Viacheslav Shalisko

16–24.07.2019

# Visualización de datos geográficos

Maunga Whau (Mt Eden) es uno de aproximadamente 50 volcanos en el campo volcánico Auckland

Un modelo topográfico de esta montaña con resolución de 10 m estan en los *datasets* incluidos en R

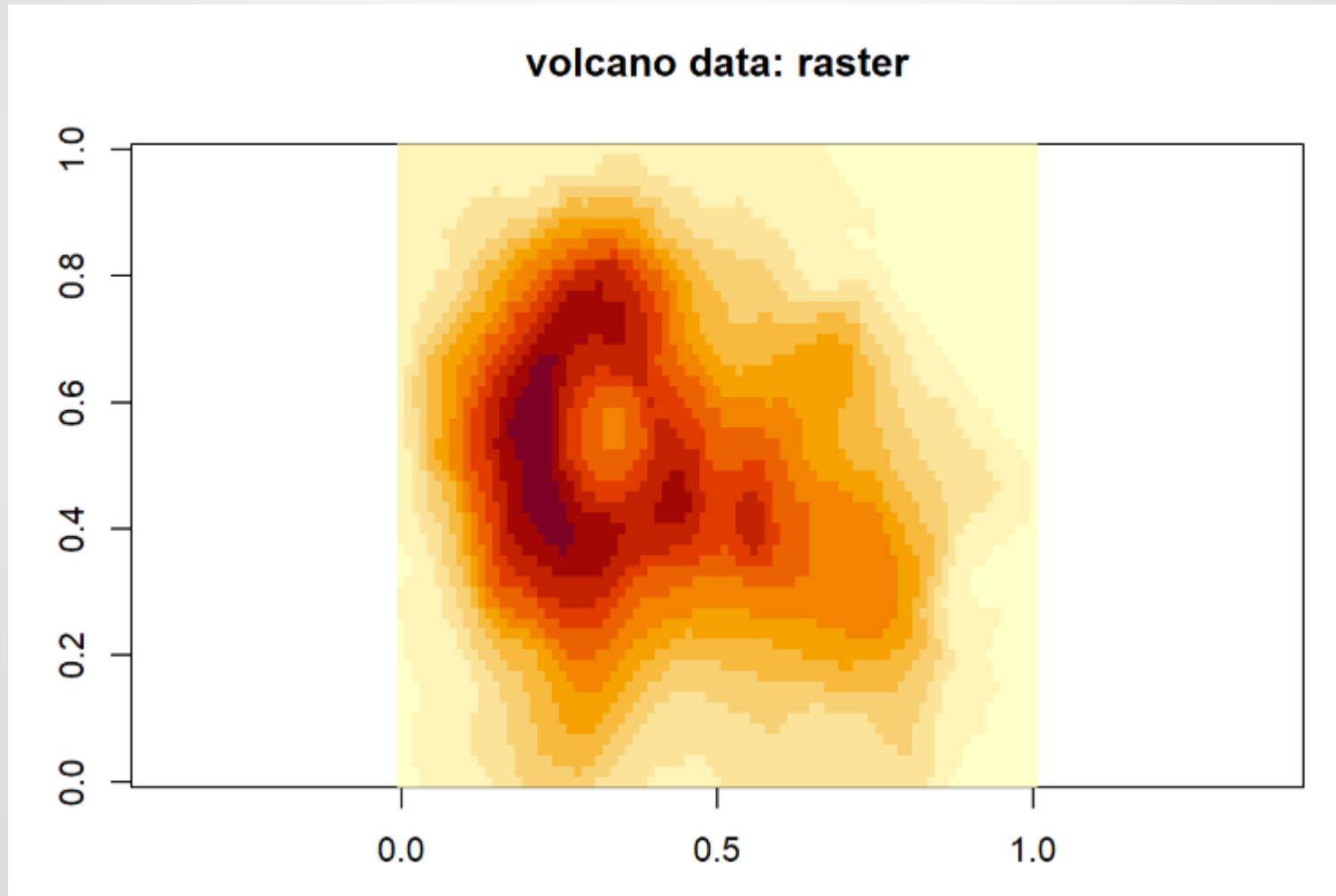
```
## volcano es un matriz números enteros  
str(volcano)
```

```
##  num [1:87, 1:61] 100 101 102 103 104 105 105 106 107 108 ...
```

```
## se puede visualizar por medio de la función image  
image(volcano, asp = 1)  
title(main = "volcano data: raster")
```

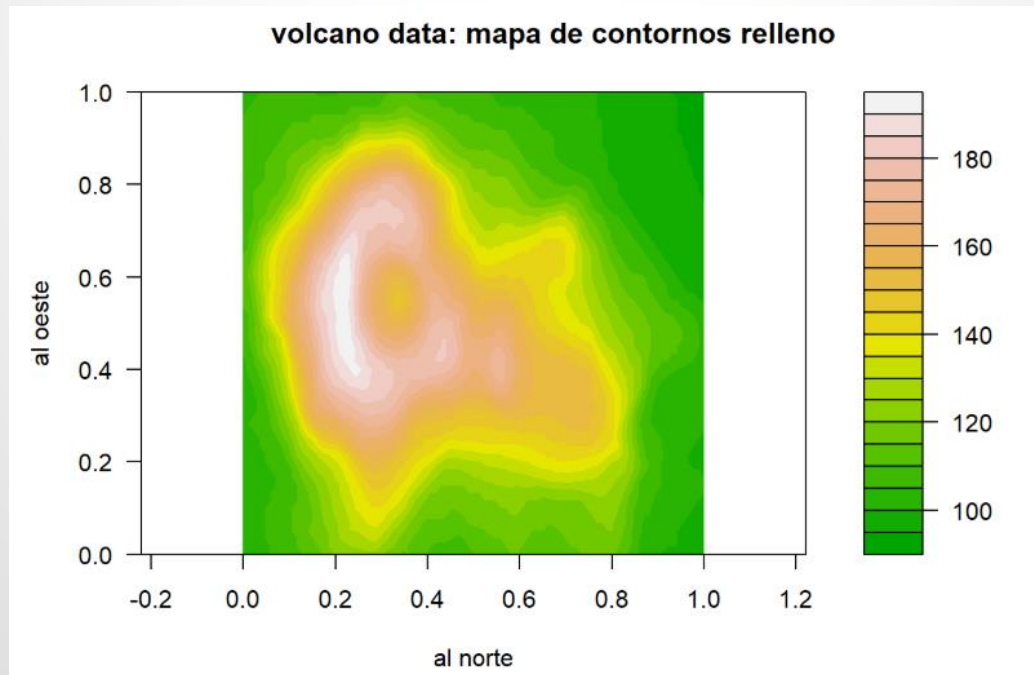


# Visualización de datos geográficos



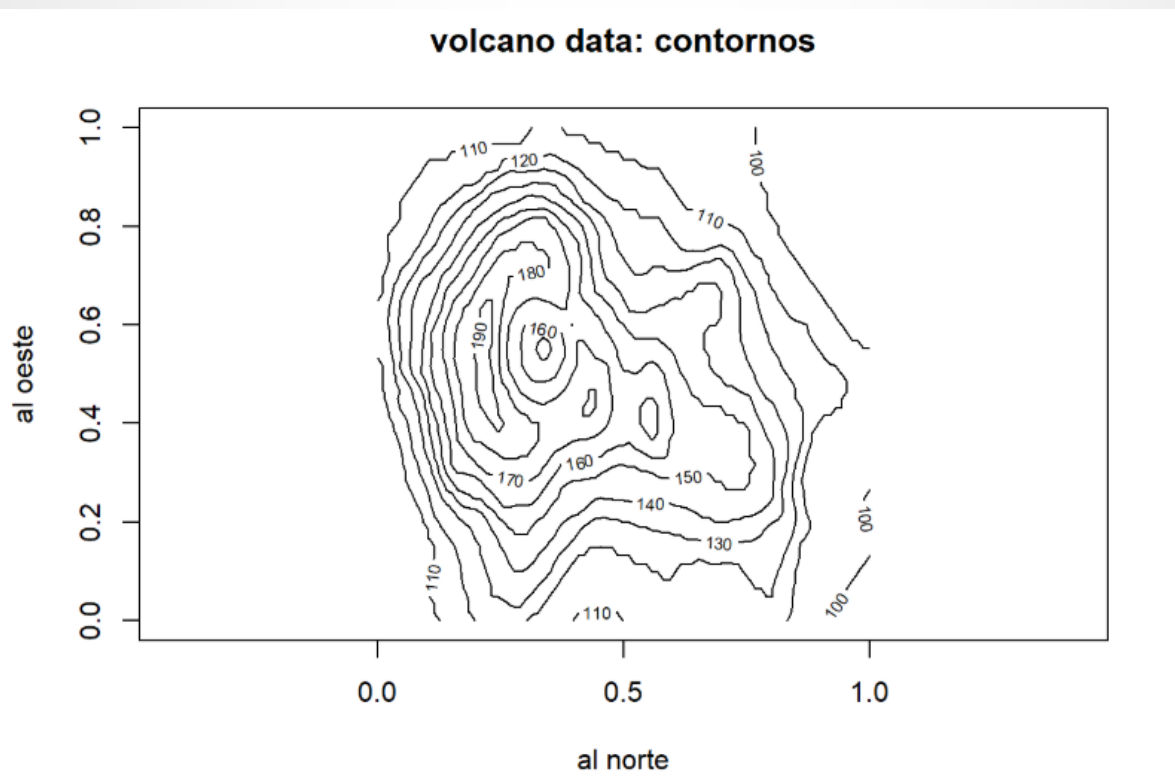
# Visualización de datos geográficos

```
## una alternativa de visualización es representarlo como contornos  
filled.contour(volcano, color.palette = terrain.colors, asp = 1, xlab = "al norte", y  
lab = "al oeste")  
title(main = "volcano data: mapa de contornos rellenos")
```



# Visualización de datos geográficos

```
## o simplemente como contornos  
contour(volcano, asp = 1, xlab = "al norte", ylab = "al oeste")  
title(main = "volcano data: contornos")
```



# Visualización de datos geográficos

Es posible convertir dataset *volcano* a una autentica capa de datos geográfica, para esto se requiere agregar georeferenciación

Módulos ***raster*** y ***sp*** son básicos para trabajo con datos geoespaciales

```
library(raster)
```

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

```
# se requiere rotar matriz antes de transformar al raster  
volcano.r <- raster(volcano[87:1,61:1])  
class(volcano)
```

```
## [1] "matrix"
```

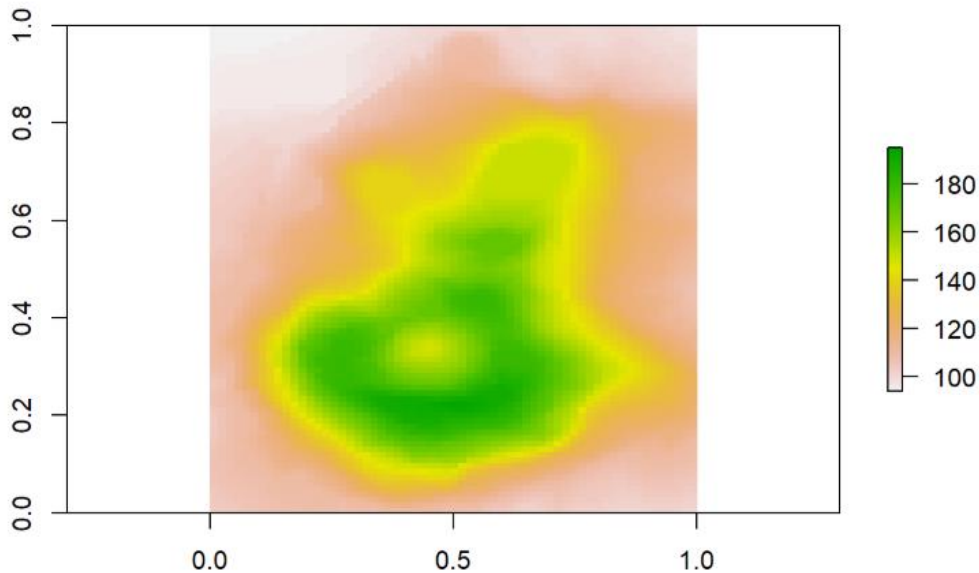
```
class(volcano.r)
```

```
## [1] "RasterLayer"  
## attr(,"package")  
## [1] "raster"
```

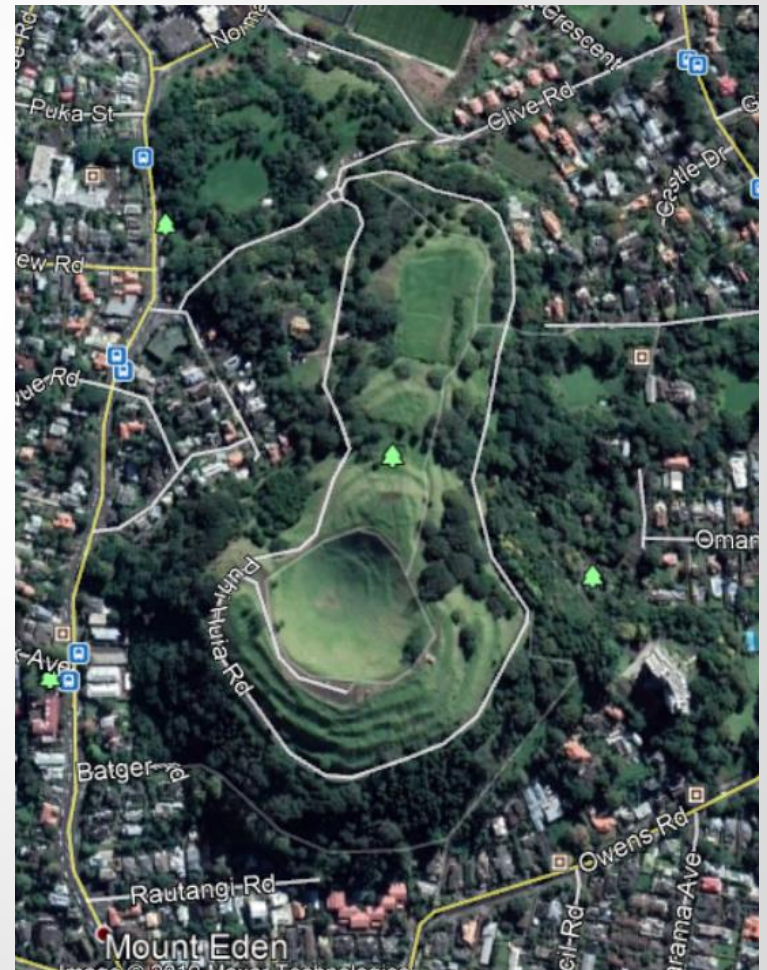


# Visualización de datos geográficos

```
plot(volcano.r)
```



Mt. Eden en GoogleEarth (norte arriba)



# Visualización de datos geográficos

```
y <- seq(from = 6478705, length.out = 87, by = 10)
x <- seq(from = 2667405, length.out = 61, by = 10)

volcano.r <- raster(volcano[87:1,61:1],
  xmn = min(x), xmx = max(x), ymn = min(y), ymx = max(y),
  crs = "+init=epsg:27200")
plot(volcano.r)
```

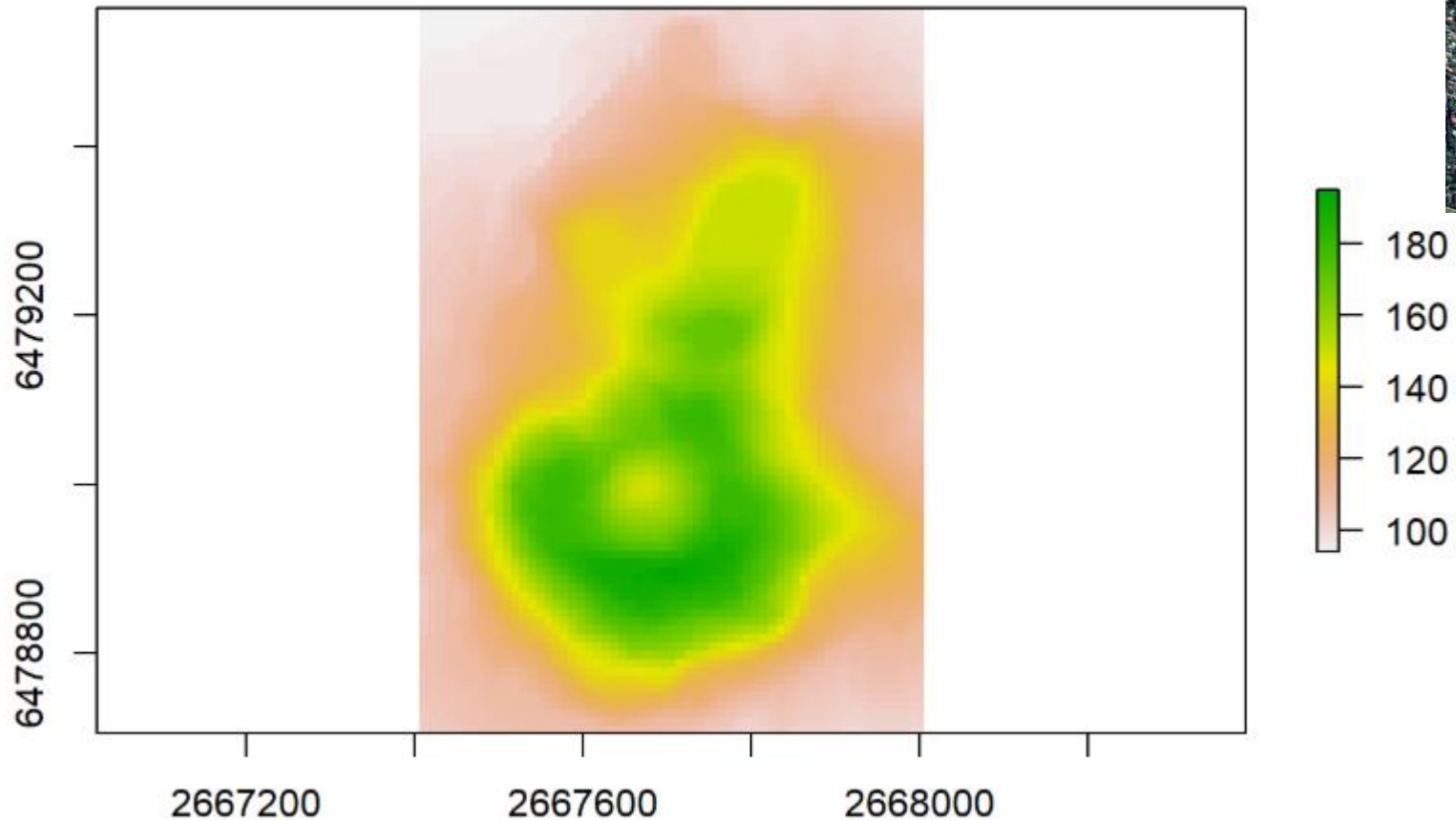
Clave de proyección según estándar EPSG  
Para más información consulta material de Melanie Frazier  
sobre las sistemas de referencias de coordenadas (CRS) en R

## EPSG:27200

NZGD49 / New Zealand Map Grid -- New Zealand Map  
Grid (NZMG)



# Visualización de datos geográficos



# Visualización de datos geográficos

volcano.r

```
## class      : RasterLayer
## dimensions : 87, 61, 5307  (nrow, ncol, ncell)
## resolution : 9.836066, 9.885057  (x, y)
## extent     : 2667405, 2668005, 6478705, 6479565  (xmin, xmax, ymin, ymax)
## crs        : +init=epsg:27200 +proj=nzmg +lat_0=-41 +lon_0=173 +x_0=2510000 +y_0=6
023150 +datum=nzgd49 +units=m +no_defs +ellps=intl +towgs84=59.47,-5.04,187.44,0.47,-
0.1,1.024,-4.5993
## source     : memory
## names      : layer
## values     : 94, 195  (min, max)
```

# Visualización de datos geográficos

El módulo **leaflet** permite incorporar los mapas interactivos al documento HTML/XHTML. Mapas funcionan en un navegador por medio de JavaScript.

Es posible incluir mapas OpenStreetMap o productos de otros proveedores que funcionan con la tecnología de “tiles” y cuentan con el API abierto y gratuito



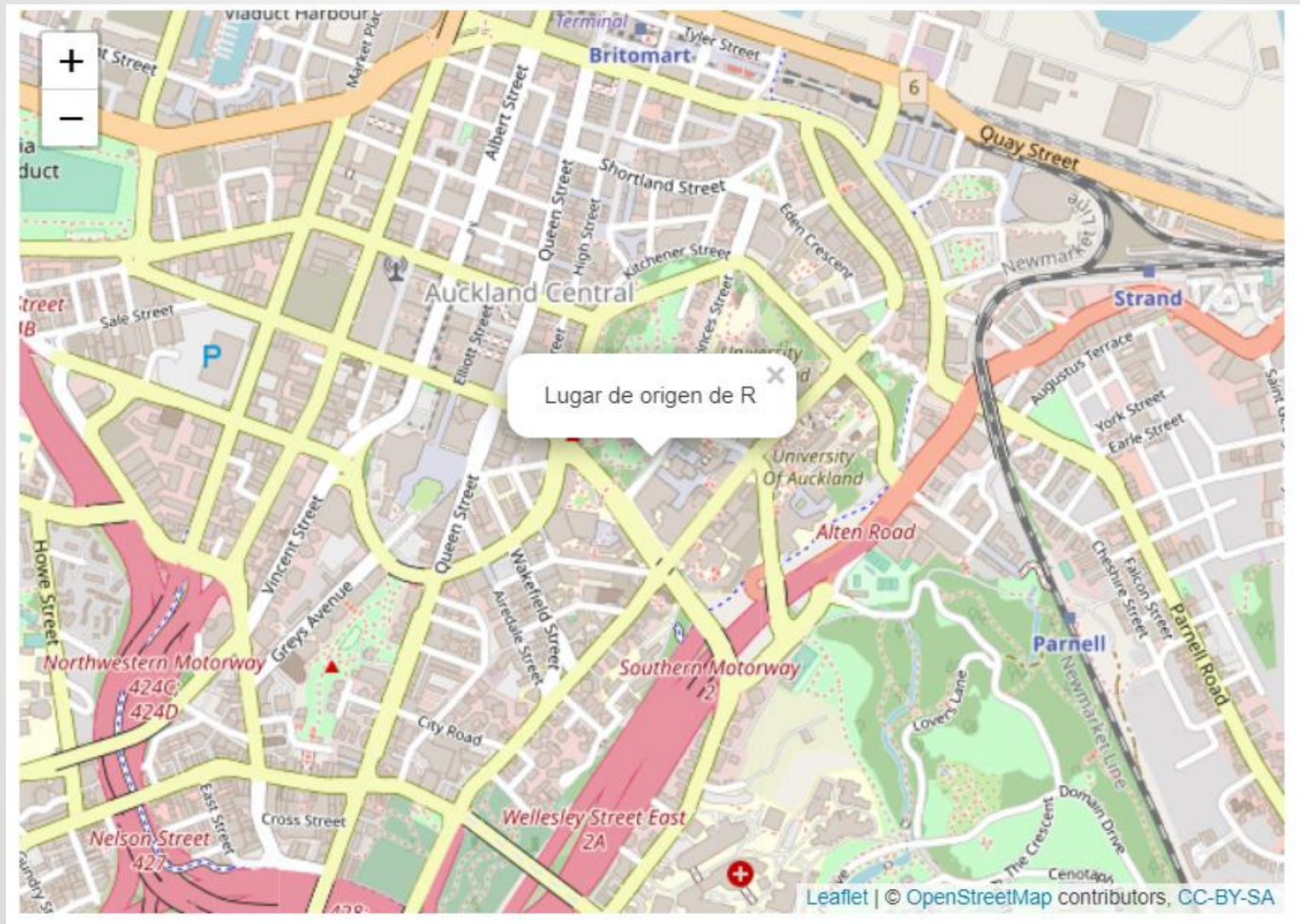
Lista de proveedores de datos para **leaflet**

<http://leaflet-extras.github.io/leaflet-providers/preview/>

```
m1 <- leaflet()
m1 <- addTiles(m1)
m1 <- addPopups(m1, lng=174.768, lat=-36.852, popup="Lugar de origen de R")
m1
```



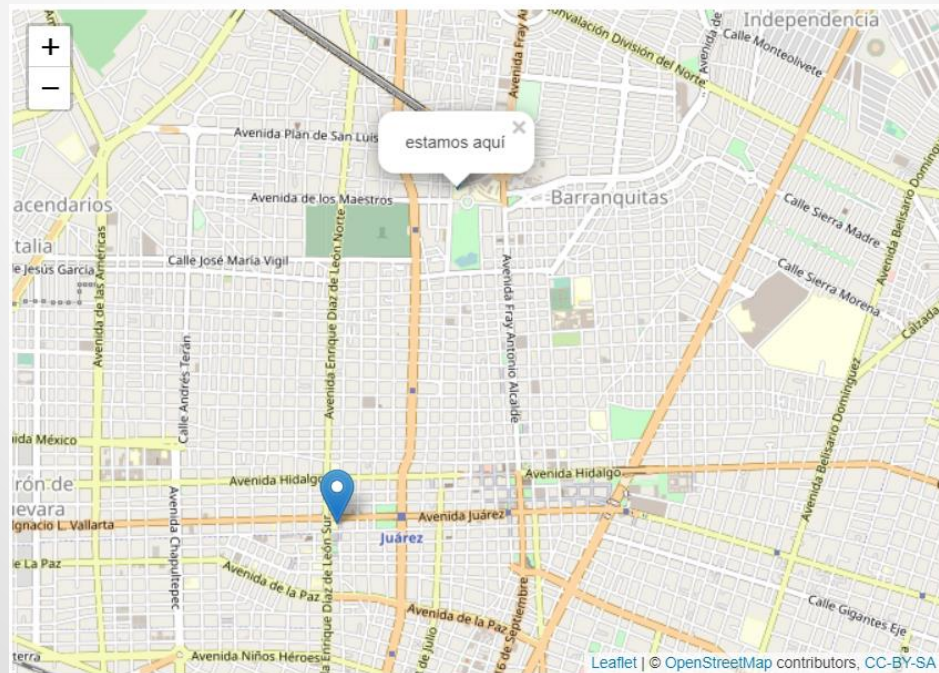
# Visualización de datos geográficos





# Visualización de datos geográficos

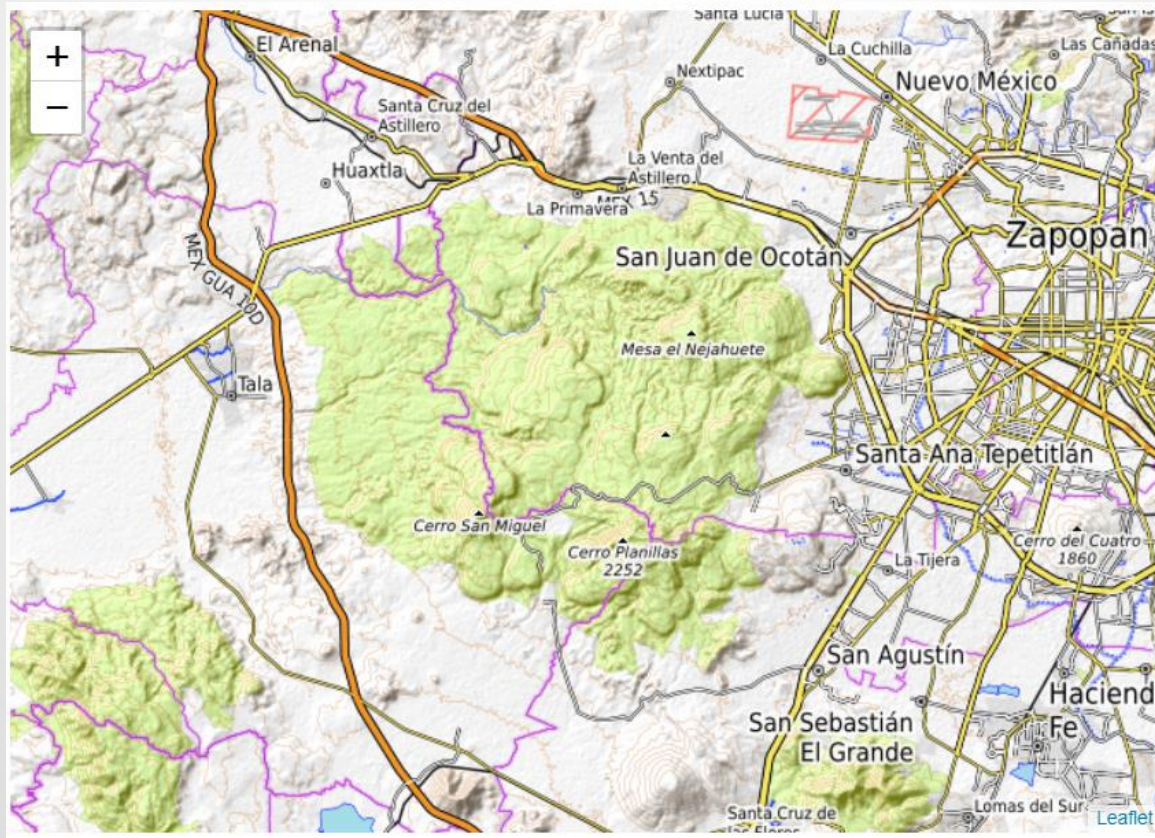
```
m2 <- leaflet()  
m2 <- addTiles(m2)  
m2 <- setView(m2, lng=-103.35, lat=20.685, zoom = 14)  
m2 <- addMarkers(m2, lng=-103.3588, lat=20.6744, popup="Rectoría")  
m2 <- addMarkers(m2, lng=-103.3513, lat=20.6937, popup="CUCSH, Geografía")  
m2 <- addPopups(m2, lng=-103.3513, lat=20.6937, popup="estamos aquí")  
m2
```





# Visualización de datos geográficos

```
m3 <- leaflet()  
m3 <- addTiles(m3, urlTemplate = 'https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png')  
m3 <- setView(m3, lng=-103.56, lat=20.64, zoom = 11)  
m3
```

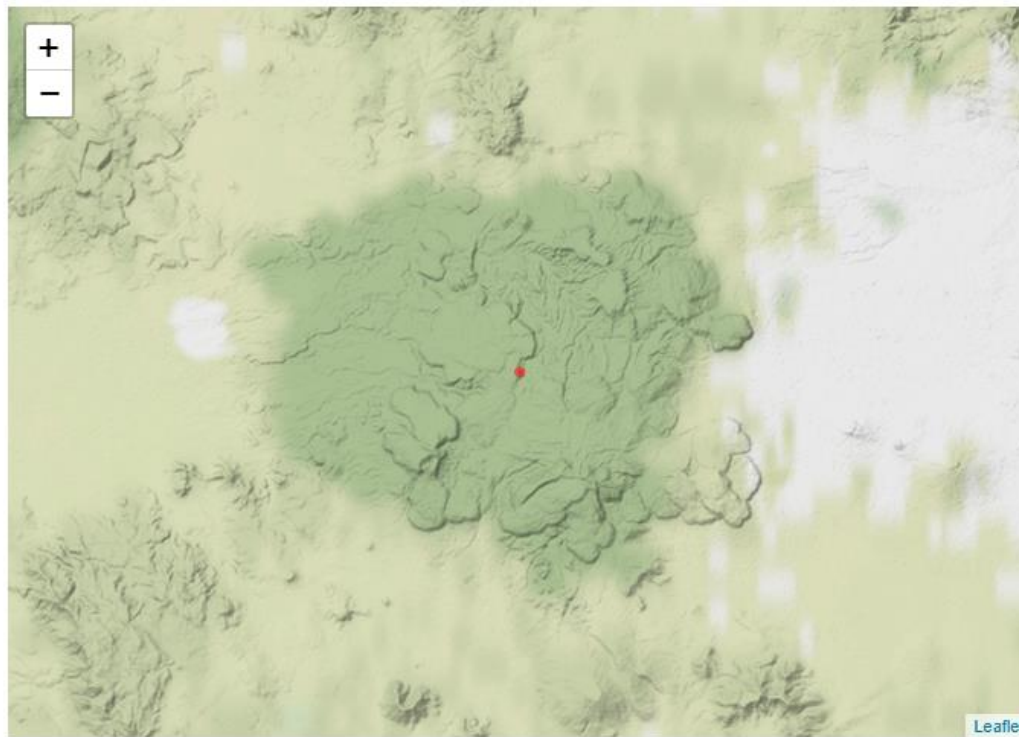




# Visualización de datos geográficos

```
mi_lat <- c(20.64)
mi_lon <- c(-103.56)

m4 <- leaflet()
m4 <- addTiles(m4, urlTemplate = 'https://stamen-tiles-{s}.a.ssl.fastly.net/terrain-background/{z}/{x}/{y}/{r}.png')
m4 <- setView(m4, lng=-103.56, lat=20.64, zoom = 11)
m4 <- addCircles(m4, lng = mi_lon, lat = mi_lat, weight = 5, radius=5,
                 color= "red", stroke = TRUE, fillOpacity = 0.8)
m4
```



# Visualización de datos geográficos

## Ejercicio 4A

Incrustar al documento Markdown HTML el mapa base de zona de Bahía de Banderas (México) utilizando el modulo *leaflet* y proveedor de datos de su elección.

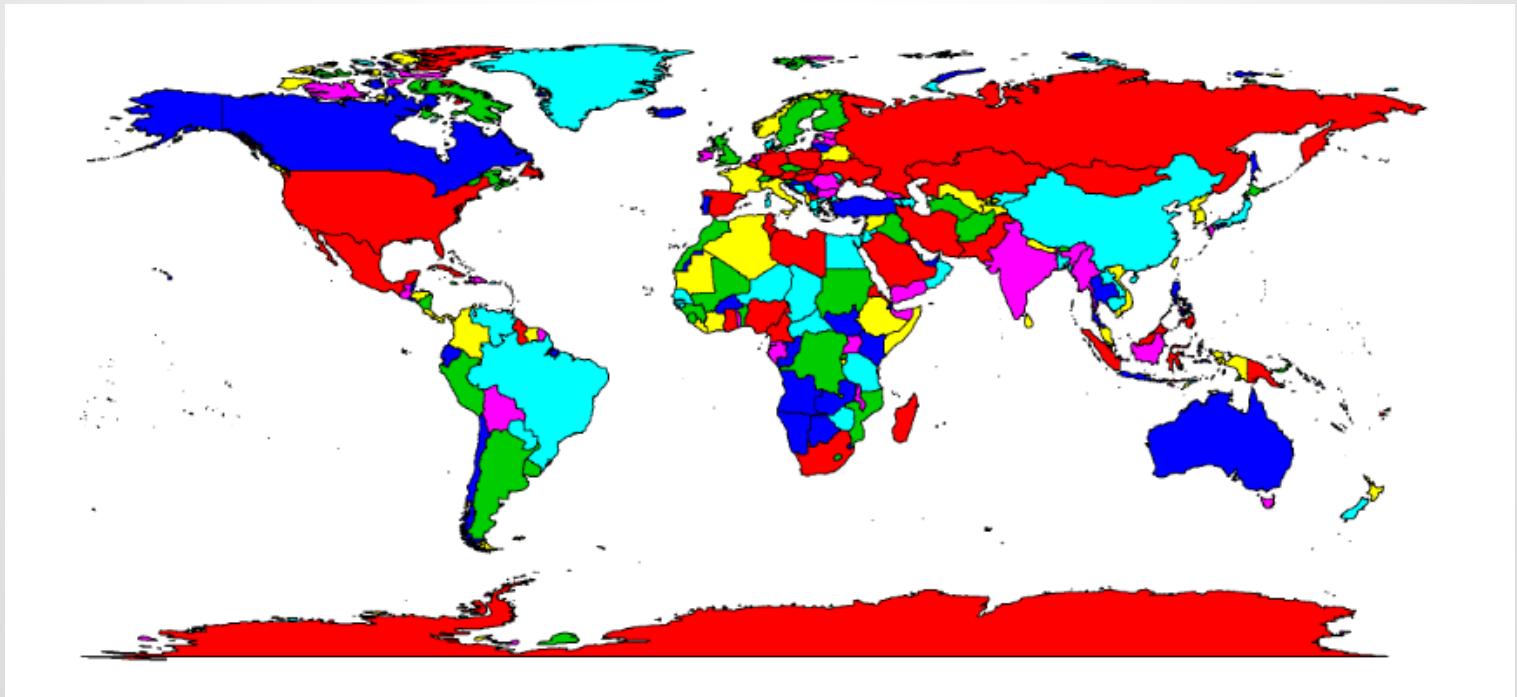
Agrega un marcador de Islas Marietas con coordenadas 20°41'56"N 105°35'06"O

Realiza exportación del resultado de trabajo en el ejercicio 4A en formato PDF utilizando función de navegador *Imprimir -> Guardar como PDF*.

# Visualización de datos geográficos

## Simple mapa del Mundo

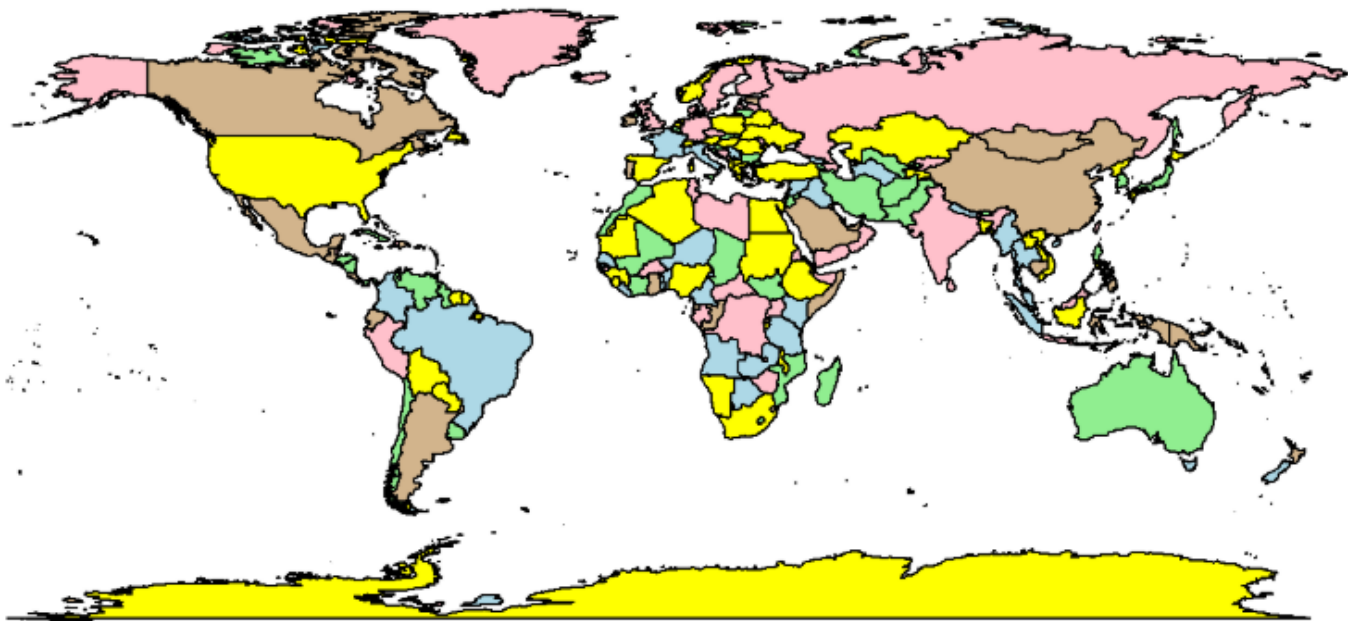
```
library(maps)  
map('world', fill = TRUE, col = 2:7)
```





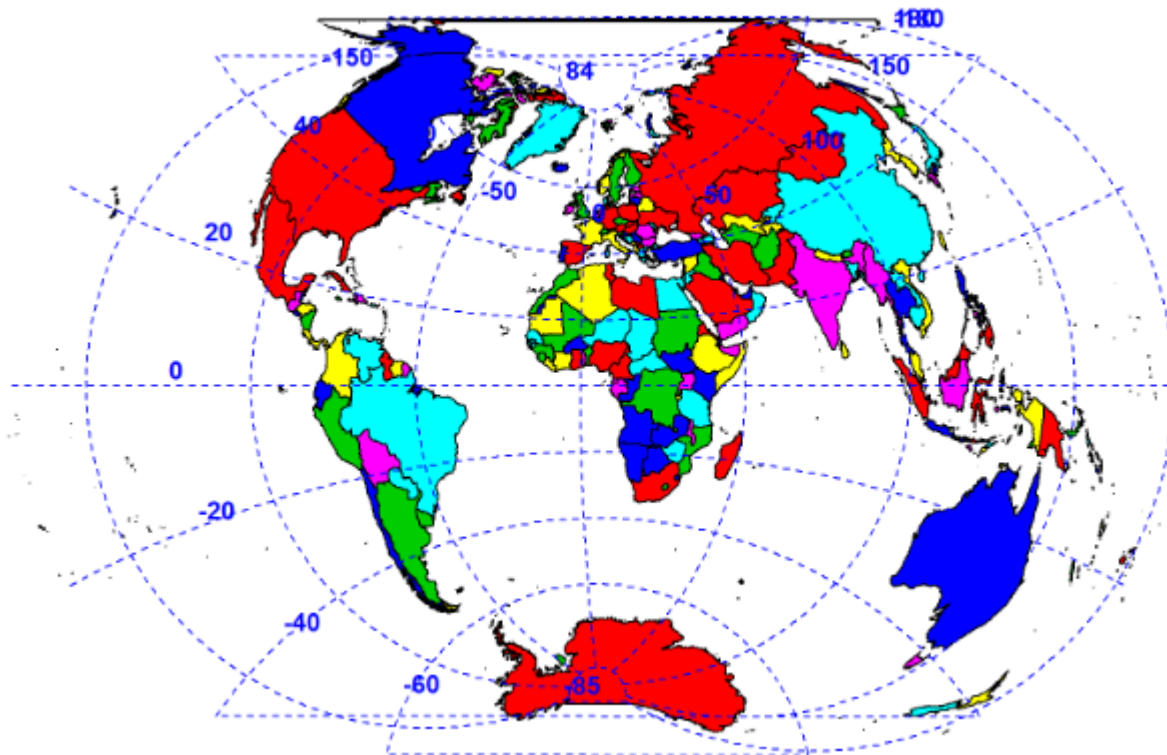
# Visualización de datos geográficos

```
mis_colores <- c("yellow", "lightgreen", "lightblue", "pink", "tan")  
  
map(database = "world", fill = TRUE, col = mis_colores)
```



# Visualización de datos geográficos

```
# otra proyección y líneas de cuadrantes  
library(mapproj)  
map('world', fill = TRUE, col = 2:7, projection = "globular")  
my_grid <- map('world', plot = FALSE)  
map.grid(my_grid)
```



# Visualización de datos geográficos

## Sur de México y Centroamérica

```
extent <- c(-107,-72,6,24)

my_map <- map("world", fill = TRUE, col = mis_colores,
             xlim = extent[1:2], ylim = extent[3:4])

map.axes()
map.scale()

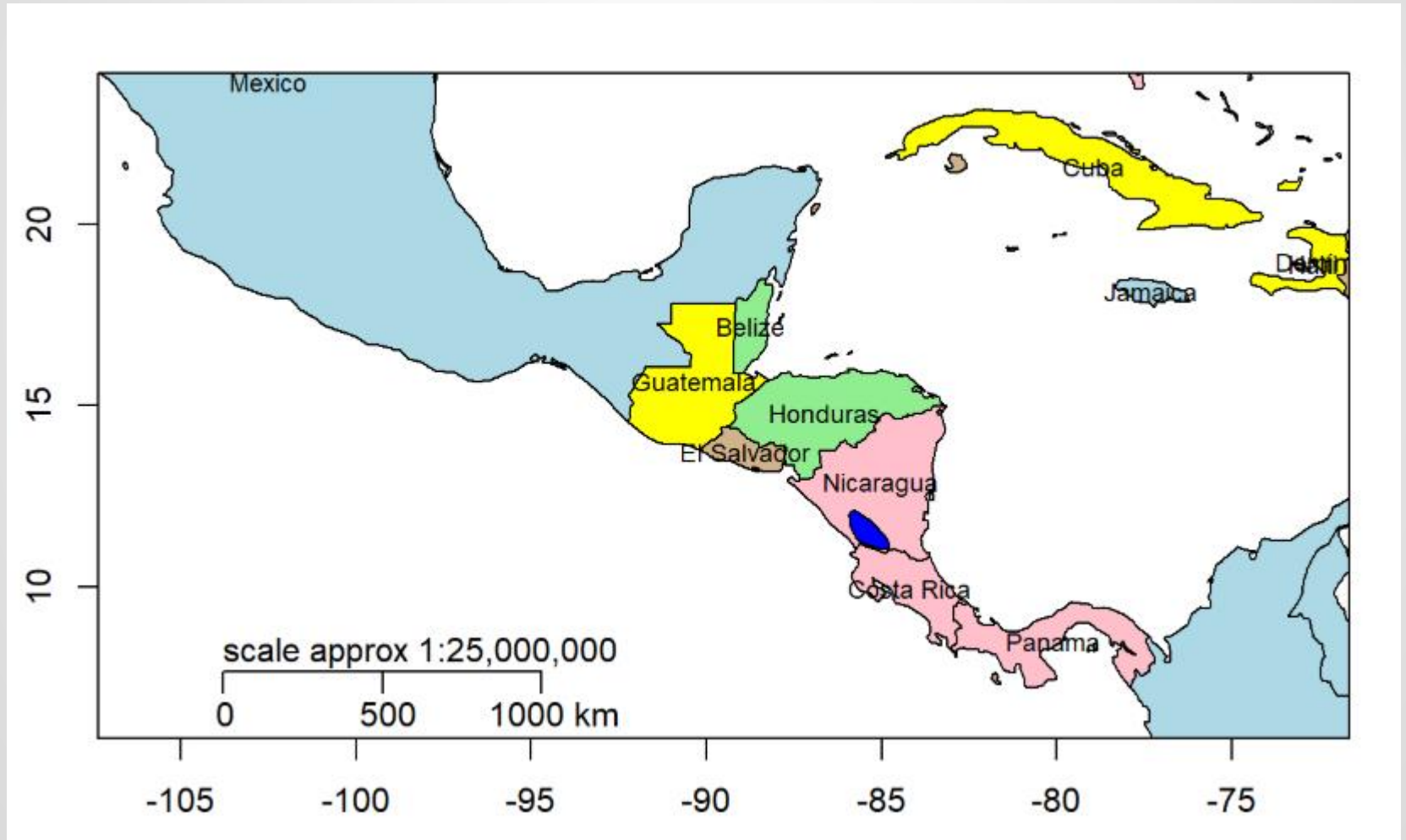
map("lakes", add = TRUE, fill = TRUE, col = "blue")

my_names <- my_map$names
my_good_names <- my_names[!grepl(":+", my_names)]

#my_good_names
map.text("world", regions = my_good_names, exact = TRUE, add = TRUE)
```



# Visualización de datos geográficos



# Visualización de datos geográficos

## Cargar los archivos shapefile y visualizarlos

```
#library(maptools)  
#municipios <- readShapePoly("datos/Municipios.shp")    # obsoleto  
#estados <- readShapePoly("datos/Estados.shp")    # obsoleto
```

```
library(sp)  
library(rgdal)
```

```
municipios <- readOGR("datos/Municipios.shp")
```

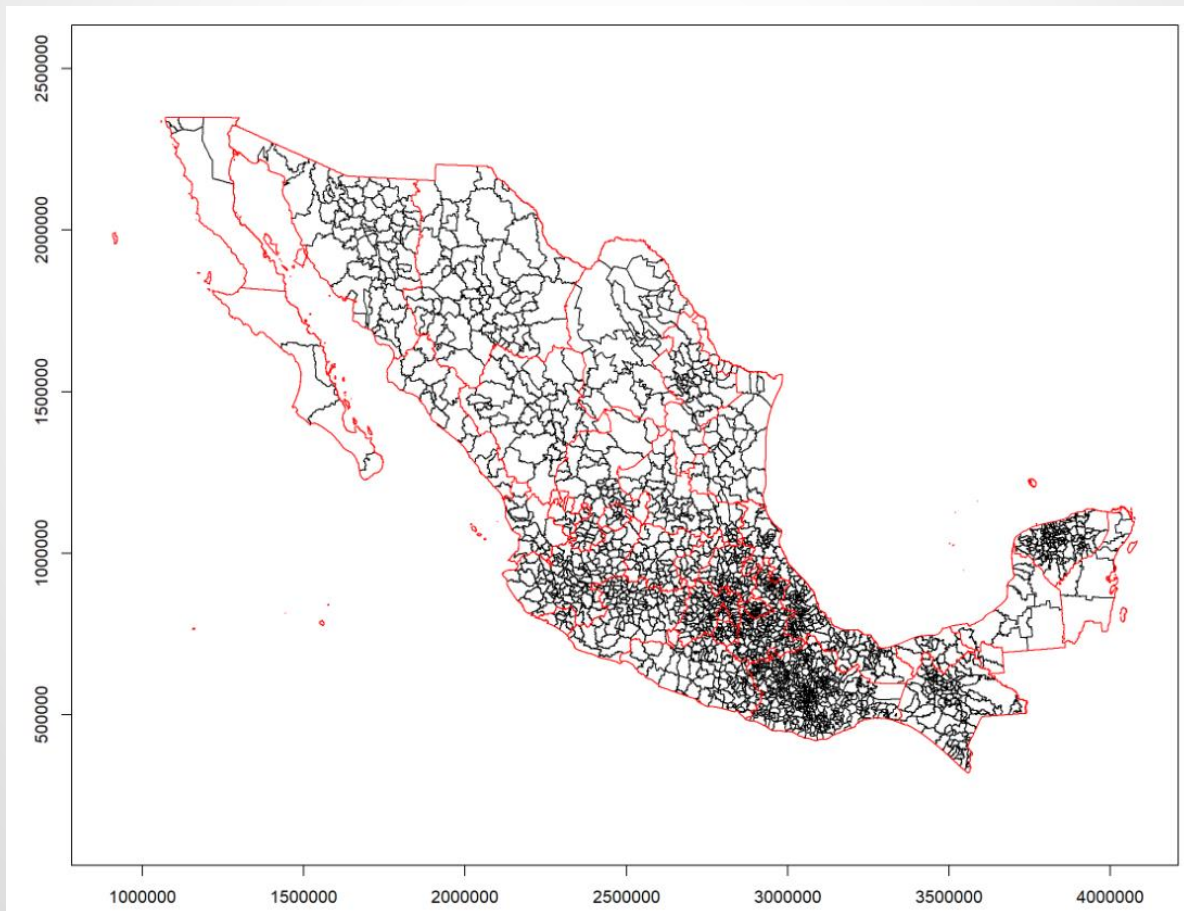
```
## OGR data source with driver: ESRI Shapefile  
## Source: "C:\Users\vshal\GD\UdeG_Docencia\CUCSH_Curso_R\sources\datos\Municipios.shp", layer: "Municipios"  
## with 2456 features  
## It has 4 fields
```

```
estados <- readOGR("datos/Estados.shp")
```

```
## OGR data source with driver: ESRI Shapefile  
## Source: "C:\Users\vshal\GD\UdeG_Docencia\CUCSH_Curso_R\sources\datos\Estados.shp", layer: "Estados"  
## with 32 features
```

# Visualización de datos geográficos

```
plot(municipios, axes = TRUE)  
plot(estados, border = "red", add = TRUE)
```





```
estados_utf <- readOGR("datos/Estados.shp", encoding = "UTF-8")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\vshal\GD\UdeG_Docencia\CUCSH_Curso_R\sources\datos\Estados.shp", layer: "Estados"
## with 32 features
## It has 2 fields
```

```
proj4string(estados_utf)
```

```
## [1] "+proj=lcc +lat_1=17.5 +lat_2=29.5 +lat_0=12 +lon_0=-102 +x_0=2500000 +y_0=0 +ellps=GRS80 +units=m +no_defs"
```

```
levels(estados_utf$NOM_ENT)
```

```
## [1] "Aguascalientes"      "Baja California"
## [3] "Baja California Sur"  "Campeche"
## [5] "Chiapas"              "Chihuahua"
## [7] "Coahuila de Zaragoza" "Colima"
## [9] "Distrito Federal"     "Durango"
## [11] "Guanajuato"           "Guerrero"
## [13] "Hidalgo"              "Jalisco"
## [15] "México"               "Michoacán de Ocampo"
## [17] "Morelos"              "Nayarit"
## [19] "Nuevo León"           "Oaxaca"
## [21] "Puebla"               "Querétaro"
## [23] "Quintana Roo"         "San Luis Potosí"
## [25] "Sinaloa"              "Sonora"
## [27] "Tabasco"              "Tamaulipas"
## [29] "Tlaxcala"             "Veracruz de Ignacio de la Llave"
## [31] "Yucatán"              "Zacatecas"
```

# Visualización de datos geográficos

Nota sobre proyección LCC usada por INEGI:

De acuerdo con la descripción en la [Guía para asignar sistemas de coordenadas](#), la proyección LCC de INEGI cuenta con siguientes parámetros:

Datum International Terrestrial Reference Frame 1992 (ITRF92 ) (epsg:6651)

Elipsoide GRS 1980 (epsg:7019)

Proyección Lambert Conformal Conic 2SP (epsg:9802)

Falso este 2500000

Falso norte 0

Paralelo estándar 1 17°30'0.00" N

Paralelo estándar 2 29°30'0.00" N

Meridiano central 102°00'0.00" W

Latitud origen 12° 00'0.00" W

Factor de escala 1

Unidad lineal Metros

Para propósitos prácticos se puede considerar que el Datum ITRF92 coincide con el datum WGS1984, ya que la diferencia entre estos datums en mayor parte de la Tierra no supera 10 cm (<ftp://itrf.ensg.ign.fr/pub/itrf/WGS84.TXT>).

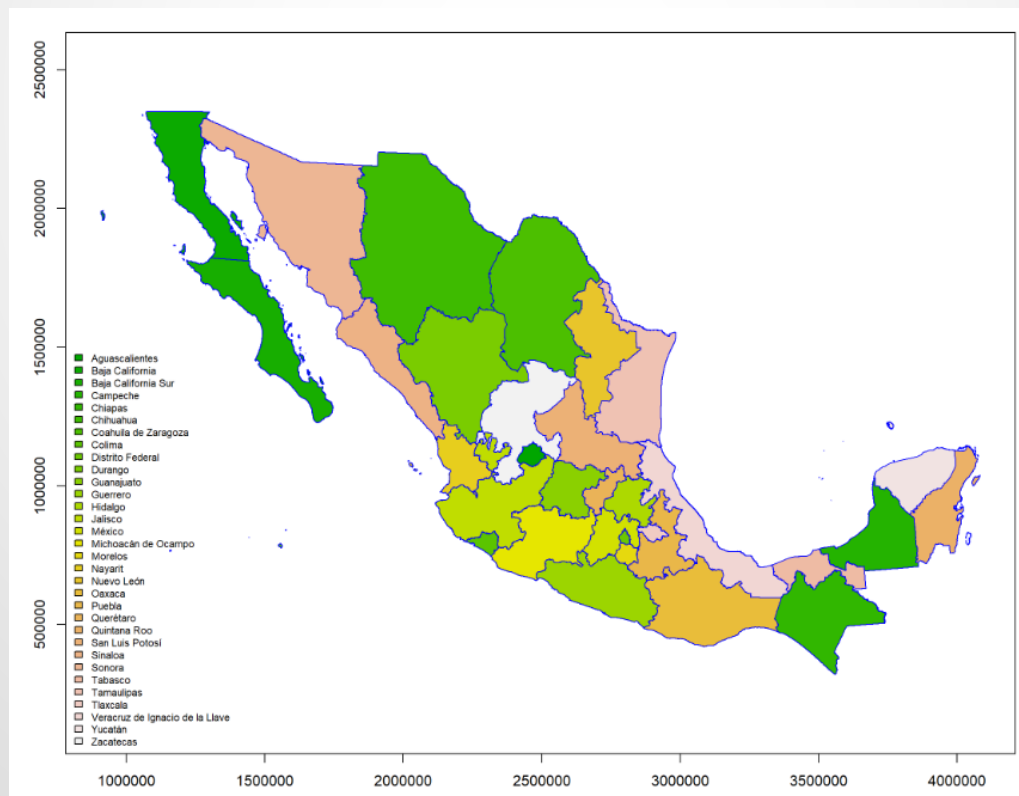
La cadena que empleamos para definir la proyección LCC de INEGI en este caso es

```
"+proj=lcc +lat_1=17.5 +lat_2=29.5 +lat_0=12 +lon_0=-102 +x_0=2500000 +y_0=0 +units=m +ellps=GRS80 +datum=WGS84" .
```



# Visualización de datos geográficos

```
palette(terrain.colors(32))  
plot(estados_utf, axes = TRUE, border = "blue", col = estados_utf$NOM_ENT)  
legend("bottomleft", legend = levels(estados_utf$NOM_ENT),  
      fill = palette(terrain.colors(32)), bty = "n", cex = 0.7)
```



```
library(rgdal)
library(sp)
library(maps)
library(GISTools)
```

```
## Warning: package 'GISTools' was built under R version 3.6.1
```

```
## Loading required package: maptools
```

```
## Checking rgeos availability: TRUE
```

```
## Loading required package: RColorBrewer
```

```
## Loading required package: MASS
```

```
## Loading required package: rgeos
```

```
## rgeos version: 0.4-3, (SVN revision 595)
## GEOS runtime version: 3.6.1-CAPI-1.10.1
## Linking to sp version: 1.3-1
## Polygon checking: TRUE
```

```
##
## Attaching package: 'GISTools'
```

```
## The following object is masked from 'package:maps':
##
##     map.scale
```

# Visualización de datos geográficos

## Cuadro reproyectado a coordenadas geográficas con puntos de Guadalajara y Ciudad de México

```
mi_crs2 <- CRS("+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0")

estados_geo <- spTransform(estados_utf, mi_crs2)

lista_lat <- c(20.65,19.41)
lista_lon <- c(-103.35,-99.13)
puntos <- data.frame(lista_lon,lista_lat)
names(puntos) <- c("Longitud","Latitud")

puntos_sp <- SpatialPoints(puntos, proj4string = mi_crs2)
puntos_sp
```

```
## SpatialPoints:
##      Longitud Latitud
## [1,]  -103.35   20.65
## [2,]   -99.13   19.41
## Coordinate Reference System (CRS) arguments: +proj=longlat
## +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
```



# Visualización de datos geográficos

```
plot(estados_geo, axes = TRUE, border = "blue", col = estados_geo$NOM_ENT)
points(puntos_sp$Longitud, puntos_sp$Latitud, pch = 19, col = "red", cex = 2)
legend("bottomleft", legend = levels(estados_geo$NOM_ENT),
      fill = palette(terrain.colors(32)), bty = "n", cex = 0.8)

maps::map.scale(-95, 13.5, relwidth = 0.1, ratio = FALSE)
north.arrow(-89, 13.2, 0.2, "N")
text(-102.5, 34, "Estados de México", cex = 1.8, adj = c(0.5,0))
```

# Visualización de datos geográficos



# Visualización de datos geográficos

## Ejercicio 4B

Elabora el mapa de México con límites de estados utilizando el cuadro con coordenadas geográficas.

Agrega leyenda con nombres de estados, escala, flecha del norte (similar al ejemplo).

Agrega los puntos de 4 ciudades incluidos en la tabla ciudades.df

\*Opcionalmente agrega los nombres de 4 ciudades.



# PROFACAD: Producto 4 para el portafolio

## **Producto para el portafolio: Producto 5. Ejercicios 4A y 4B**

Fecha de entrega del producto: 26 de julio 2019

Actividades: En forma individual realizar los ejercicios 4A y 4B.

Para lograrlo escribir el código en R que permite visualizar los datos geográficos en el formato vectorial y ráster, realizar la definición de las proyecciones geográficas, re-proyección entre ellos, combinación de capas de datos y generación de un mapa base en el formato que cumpla con algunos de los criterios del producto cartográfico.

Producto para el portafolio: Reporte de trabajo con capas de datos geográficas y generación de un mapa base en el formato del documento R Markdown HTML y PDF, que cumpla con las condiciones de los ejercicios 4A y 4B.

# Estructuras de datos en R

## Bibliografía complementaria

Engel, C. (2019). *Using Spatial Data with R*. Libro electrónico recuperado de <https://cengel.github.io/R-spatial/>

RStudio, Inc. (2016). *Leaflet for R*. Manual electrónico recuperado de <https://rstudio.github.io/leaflet/>

Mas, J.-F. (2013). *Análisis espacial con R: Usa R como un sistema de información geográfica*. European Scientific Institute Publishing.  
<http://eujournal.org/files/journals/1/books/JeanFrancoisMas.pdf>