

# 01 Datos Vectoriales

Dr. Victor Augusto Lizcano S.

2023-11-29

## Datos espaciales

Cualquier fenómeno espacial puede ser representado a través de **objetos** discretos (cuyos límites son bien definidos como un bosque, un río o un asentamiento humano) o continuos (cuyos límites no suelen ser claros como la lluvia o la temperatura).

Espacialmente, estos **objetos** se suelen representar mediante datos **vectoriales**. Este tipo de datos constan de una descripción geométrica y forma (como puntos, líneas o polígonos). Por ejemplo las fronteras de los países, estados, provincias, prefecturas, departamentos, municipios o condados están delimitadas por polígonos.

Miremos el siguiente ejemplo en el que representamos en forma de puntos la localización de algunas estaciones climáticas y la magnitud de la variable precipitación:

```
nombreEstaciones <- LETTERS[1:10] #Las nombraremos con letras para este ejemplo
longitud <- c(-116.7, -120.4, -116.7, -113.5, -115.5,
             -120.8, -119.5, -113.7, -113.7, -110.7)
latitud <- c(45.3, 42.6, 38.9, 42.1, 35.7, 38.9,
            36.2, 39, 41.6, 36.9)
estaciones <- cbind(longitud, latitud)

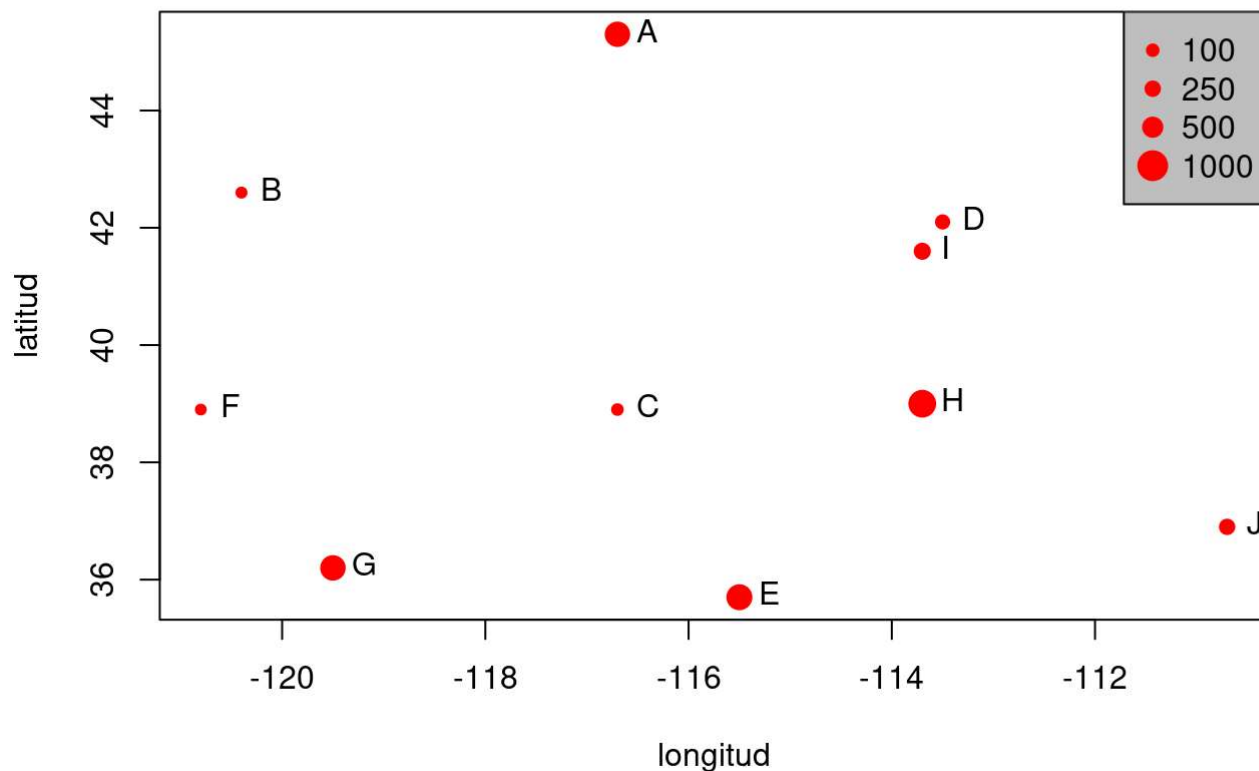
# Simularemos datos de precipitación para este ejemplo
set.seed(0)
precip <- round((runif(length(latitud))*10)^3) #Generamos datos aleatorios

psize <- 1 + precip/500
plot(estaciones, cex=psize, pch=20, col='red', main='Precipitación')

# Agregamos nombres al gráfico
text(estaciones, nombreEstaciones, pos=4)

# Agregamos Leyenda
breaks <- c(100, 250, 500, 1000)
legend.psize <- 1+breaks/500
legend("topright", legend=breaks, pch=20, pt.cex=legend.psize, col='red', bg='gray')
```

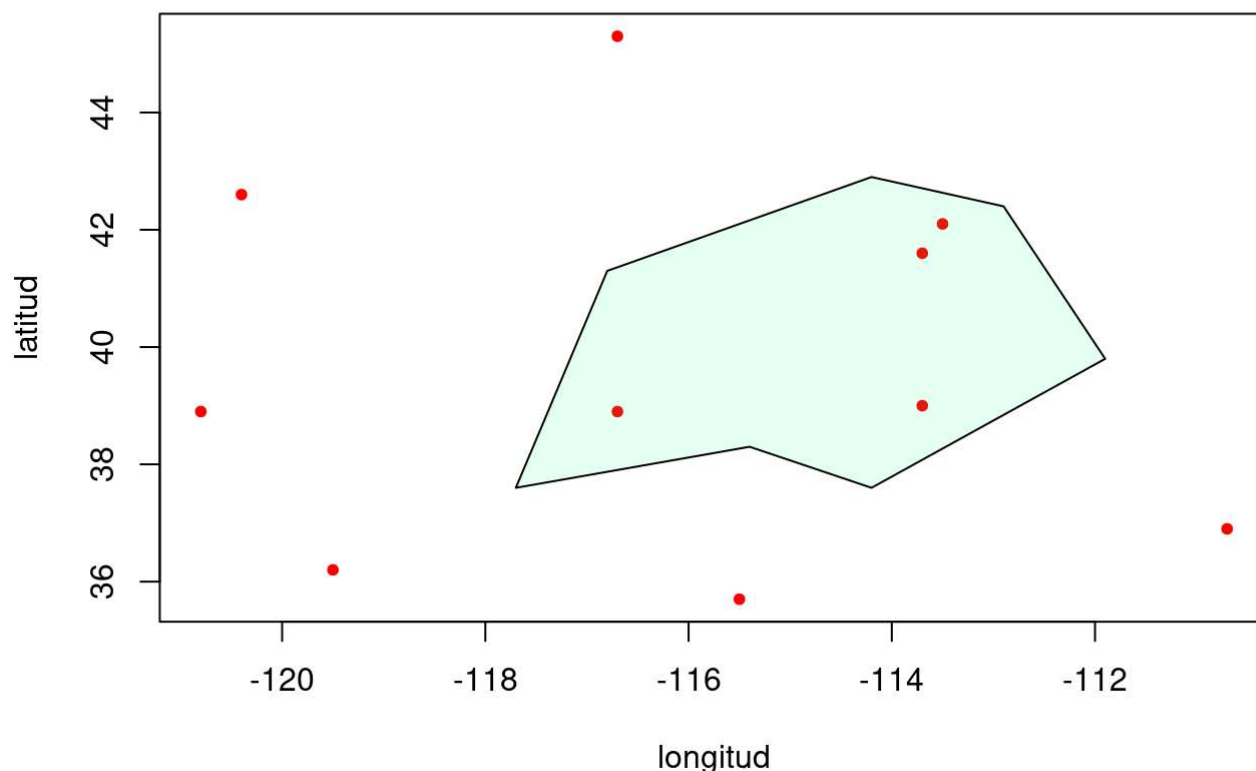
## Precipitación



A parte de los datos de tipo **vectorial**, también encontramos los datos tipo **raster**. Estos datos comunmente se emplean para representar espacialmente objetos continuos como la elevación de un terreno. Los datos **raster** son imágenes, las cuales almacenan sus datos en pixeles y conservan un sistema de referencia de coordenadas al igual que los datos vectoriales.

```
lon <- c(-116.8, -114.2, -112.9, -111.9, -114.2, -115.4, -117.7)
lat <- c(41.3, 42.9, 42.4, 39.8, 37.6, 38.3, 37.6)
esquinasPoligono = cbind(lon,lat)
plot(estaciones, main="Poligono que delimita el área de 4 estaciones climáticas", pch=20, col="red")
polygon(esquinasPoligono, col= rgb(0, 1, 0.5, 0.1), border = "black")
```

## Polígono que delimita el área de 4 estaciones climáticas



## Representación de vectores de tipo punto con R

Veamos el siguiente ejemplo en el cual representamos un conjunto de datos espaciales como vectores tipo punto:

```
#Vamos a emplear la librería sp para este caso
library(sp)

#Vamos a crear nuestro conjunto de datos de longitud y latitud

lonx = c(-76.28, -75.17, -73.03, -72.51)
latx = c(3.29, 4.27, 7.10, 11.34)
dengueCasos = c(79, 56, 49, 36)
coordenadas = cbind(lonx, latx)

puntos = SpatialPoints(coordenadas) #Empleamos la función espacial points de la librería sp para
crear nuestros datos vectoriales tipo punto

class(puntos) #Visualizamos la clase de archivo que generamos
```

```
## [1] "SpatialPoints"
## attr(,"package")
## [1] "sp"
```

```
showDefault(puntos) #Mostramos la información de nuestro archivo vectorial
```

```
## An object of class "SpatialPoints"
## Slot "coords":
##      lonx  latx
## [1,] -76.28  3.29
## [2,] -75.17  4.27
## [3,] -73.03  7.10
## [4,] -72.51 11.34
##
## Slot "bbox":
##      min      max
## lonx -76.28 -72.51
## latx   3.29  11.34
##
## Slot "proj4string":
## Coordinate Reference System:
## Deprecated Proj.4 representation: NA
```

```
# Vemos que aun no hemos definido un sistema de referencia de coordenadas
```

```
srcoord <- CRS('+proj=coordenadas +datum=WGS84') #Nuestro src es WGS84
```

```
# Ahora reasignamos a nuestra variable el src
```

```
puntos = SpatialPoints(coordenadas, proj4string=srcoord)
```

```
showDefault(puntos) #Nuevamente verificamos
```

```
## An object of class "SpatialPoints"
## Slot "coords":
##      lonx  latx
## [1,] -76.28  3.29
## [2,] -75.17  4.27
## [3,] -73.03  7.10
## [4,] -72.51 11.34
##
## Slot "bbox":
##      min      max
## lonx -76.28 -72.51
## latx   3.29  11.34
##
## Slot "proj4string":
## Coordinate Reference System:
## Deprecated Proj.4 representation: +proj=coordenadas +datum=WGS84
```

```
#A parte de La longitud y latitud (también el src) nuestra base datos aun carece de atributos.
#Vamos asignar datos de casos de dengue como eje z (o atributo)
#Para ello vamos a emplear la librería raster
```

```
puntos
```

```
## SpatialPoints:
##      lonx  latx
## [1,] -76.28  3.29
## [2,] -75.17  4.27
## [3,] -73.03  7.10
## [4,] -72.51 11.34
## Coordinate Reference System (CRS) arguments: +proj=coordenadas
## +datum=WGS84
```

```
library(raster)
```

```
puntos = SpatialPointsDataFrame(puntos, data=data.frame(dengueCasos))
puntos
```

```
## class      : SpatialPointsDataFrame
## features    : 4
## extent      : -76.28, -72.51, 3.29, 11.34 (xmin, xmax, ymin, ymax)
## crs         : +proj=coordenadas +datum=WGS84
## variables   : 1
## names       : dengueCasos
## min values   :      36
## max values   :      79
```

```
#Para ver con más detalle
str(puntos)
```

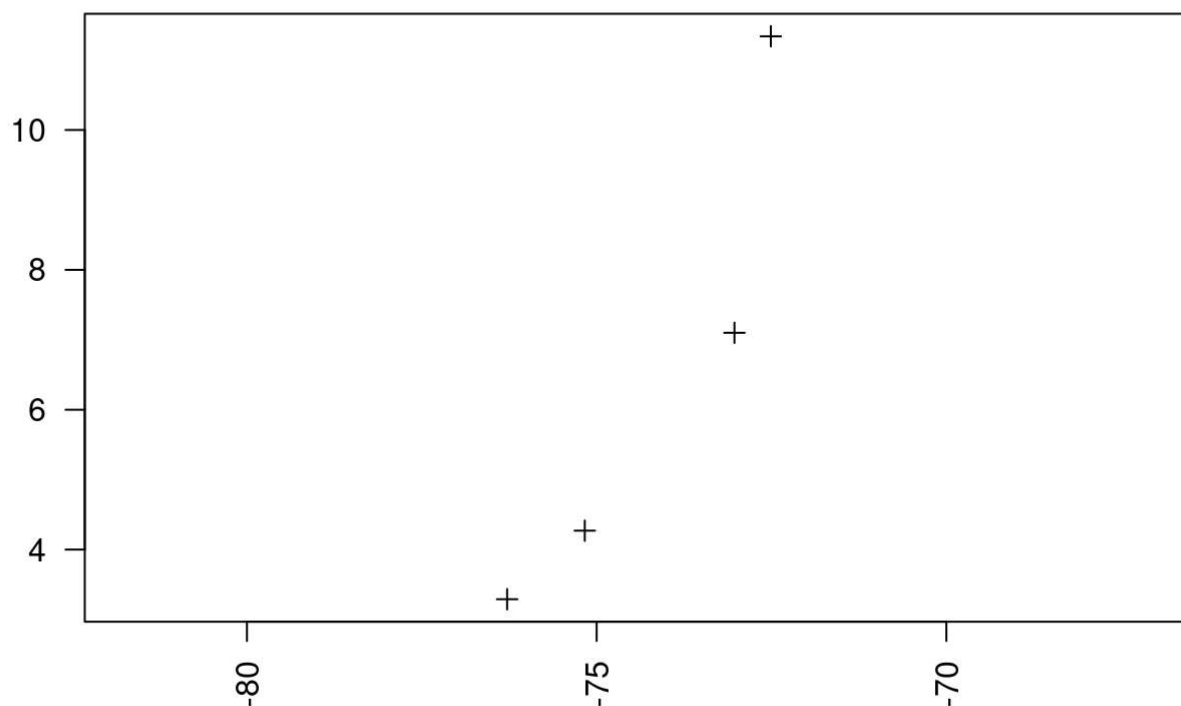
```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
## ..@ data      :'data.frame': 4 obs. of 1 variable:
## .. ..$ dengueCasos: num [1:4] 79 56 49 36
## ..@ coords.nrs : num(0)
## ..@ coords     : num [1:4, 1:2] -76.28 -75.17 -73.03 -72.51 3.29 ...
## .. ..- attr(*, "dimnames")=List of 2
## .. .. ..$ : NULL
## .. .. ..$ : chr [1:2] "lonx" "latx"
## ..@ bbox       : num [1:2, 1:2] -76.28 3.29 -72.51 11.34
## .. ..- attr(*, "dimnames")=List of 2
## .. .. ..$ : chr [1:2] "lonx" "latx"
## .. .. ..$ : chr [1:2] "min" "max"
## ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
## .. .. ..@ projargs: chr "+proj=coordenadas +datum=WGS84"
```

```
# 0
```

```
showDefault(puntos)
```

```
## An object of class "SpatialPointsDataFrame"
## Slot "data":
##   dengueCasos
## 1           79
## 2           56
## 3           49
## 4           36
##
## Slot "coords.nrs":
## numeric(0)
##
## Slot "coords":
##      lonx  latx
## [1,] -76.28  3.29
## [2,] -75.17  4.27
## [3,] -73.03  7.10
## [4,] -72.51 11.34
##
## Slot "bbox":
##      min    max
## lonx -76.28 -72.51
## latx   3.29  11.34
##
## Slot "proj4string":
## Coordinate Reference System:
## Deprecated Proj.4 representation: +proj=coordenadas +datum=WGS84
```

```
#Ahora observemos nuestros datos en un gráfico
plot(puntos, axes=TRUE, las=2)
```



## Representación de vectores de tipo línea con R

Para realizar vectores de líneas en R empleamos la función “spLines” (similar al “SpatialPoints” que empleamos anteriormente).

```
coordenadas
```

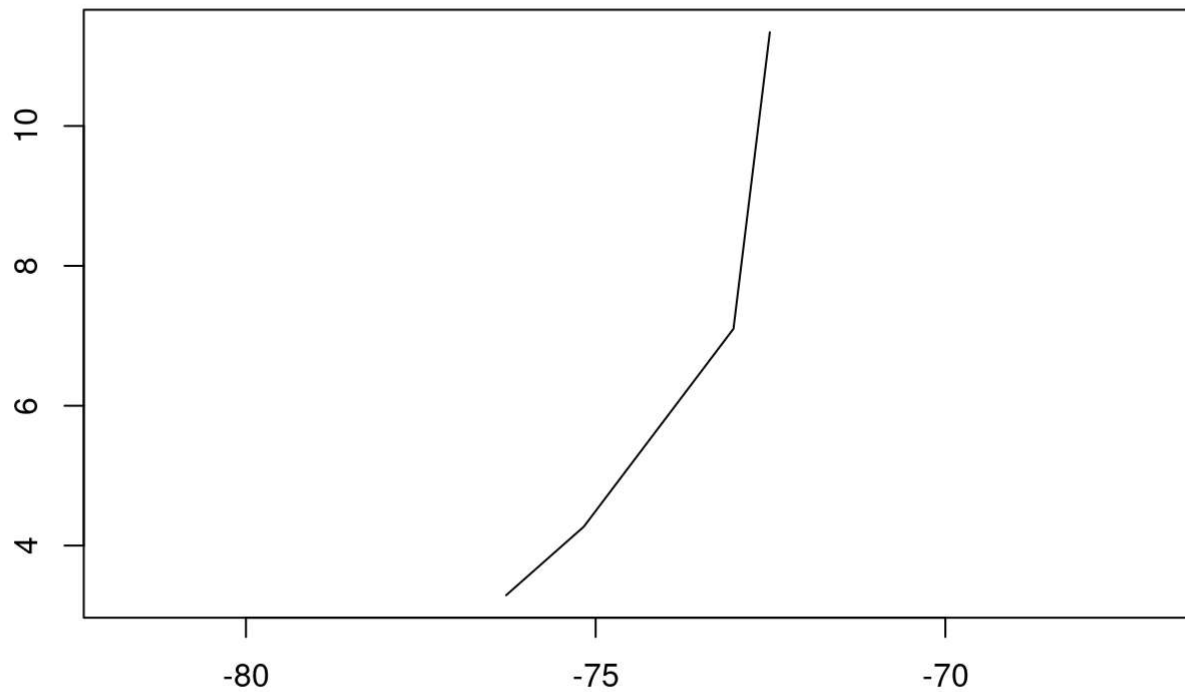
```
##      lonx  latx
## [1,] -76.28  3.29
## [2,] -75.17  4.27
## [3,] -73.03  7.10
## [4,] -72.51 11.34
```

```
lineas = splines(coordenadas, crs=srcoord)
showDefault(lineas)
```

```
## An object of class "SpatialLines"
## Slot "lines":
## [[1]]
## An object of class "Lines"
## Slot "Lines":
## [[1]]
## An object of class "Line"
## Slot "coords":
##      lonx  latx
## [1,] -76.28  3.29
## [2,] -75.17  4.27
## [3,] -73.03  7.10
## [4,] -72.51 11.34
##
##
##
## Slot "ID":
## [1] "1"
##
##
##
## Slot "bbox":
##      min    max
## x -76.28 -72.51
## y   3.29  11.34
##
## Slot "proj4string":
## Coordinate Reference System:
## Deprecated Proj.4 representation: +proj=coordenadas +datum=WGS84
```

```
plot(lineas, axes=TRUE)
```





## Representación de vectores de tipo poligono con R

Para realizar vectores de líneas en R empleamos la función “spPolygons” (similar al “SpatialPoints” que empleamos anteriormente).

```
poligonos = spPolygons(coordenadas, crs=srcoord)
showDefault(poligonos)
```

```
## An object of class "SpatialPolygons"
## Slot "polygons":
## [[1]]
## An object of class "Polygons"
## Slot "Polygons":
## [[1]]
## An object of class "Polygon"
## Slot "labpt":
## [1] -74.012084 7.051741
##
## Slot "area":
## [1] 6.42145
##
## Slot "hole":
## [1] FALSE
##
## Slot "ringDir":
## [1] 1
##
## Slot "coords":
##      [,1] [,2]
## [1,] -76.28 3.29
## [2,] -72.51 11.34
## [3,] -73.03 7.10
## [4,] -75.17 4.27
## [5,] -76.28 3.29
##
##
## Slot "plotOrder":
## [1] 1
##
## Slot "labpt":
## [1] -74.012084 7.051741
##
## Slot "ID":
## [1] "1"
##
## Slot "area":
## [1] 6.42145
##
##
## Slot "plotOrder":
## [1] 1
##
## Slot "bbox":
##      min    max
## x -76.28 -72.51
## y  3.29 11.34
##
## Slot "proj4string":
```

```
## Coordinate Reference System:
```

```
## Deprecated Proj.4 representation: +proj=coordenadas +datum=WGS84
```

```
plot(poligonos, axes=TRUE)
```

