

y R que R

Enero 2015, U. Politécnica de Gijón © <http://univgim.hol.es/wordpress/>



Elaboración de mapas territoriales con R

José Enrique Martín García (Doctor Ingeniero Industrial)

Sinopsis

En este artículo se muestran algunas de las posibilidades que ofrece R para crear mapas y representar sobre ellos información estadística.

R es usado en diversos campos para la realización cálculos estadísticos y gráficos por ordenador pero que también es magnífico a la hora de procesar datos geográficos. Las capacidades geoespaciales de R están disponibles a través de paquetes que proporcionan acceso a un elevado número de algoritmos. R y sus paquetes puede procesar puntos, líneas, polígonos o matrices de datos.

La fuente de los datos van a ser GADM database of Global Administrative Areas, una base de datos de zonas administrativas a nivel mundial. Pueden bajarse los de España en una variedad de formatos, incluido el que acepta el paquete sp de R. El INE proporciona mapas administrativos oficiales de España y permite su descarga. El centro de descargas Instituto Geográfico Nacional permite la descarga y distribución de datos para fines no comerciales. Otra opción es utilizar OpenStreetMap que se ha creado por la comunidad de usuarios. Ha crecido hasta convertirse en una de las fuentes de datos cartográficos a escala local más detallados que existen. Estos datos cartográficos son creados y mantenidos por miles de voluntarios de todo el mundo.

1. Tipos de imagen

Una imagen en mapa de bits o **imagen ráster** es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada matriz, que se puede visualizar en un monitor, papel u otro dispositivo de representación.

A las imágenes en mapa de bits se las suele definir por su altura y anchura (en píxeles) y por su profundidad de color (en bits por píxel), que determina el número de colores distintos que se pueden almacenar en cada punto individual, y por lo tanto, en gran medida, la calidad del color de la imagen.

Una **imagen vectorial** es una imagen digital formada por objetos geométricos independientes (segmentos, polígonos, arcos, etc.), cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc. Por ejemplo un círculo de color rojo quedaría definido por la posición de su centro, su radio, el grosor de línea y su color.

Este formato de imagen es completamente distinto al formato de las imágenes de mapa de bits. El interés principal de los gráficos vectoriales es poder ampliar el tamaño de una imagen a voluntad sin sufrir la pérdida de calidad que sufren los mapas de bits. De la misma forma, permiten mover, estirar y retorcer imágenes de manera relativamente sencilla. Su uso también está muy extendido en la generación de imágenes en tres dimensiones tanto dinámicas como estáticas.

Todos los ordenadores actuales traducen los gráficos vectoriales a mapas de bits para poder representarlos en pantalla al estar ésta constituida físicamente por píxeles.

El shapefile es un formato para sistemas de información geográfica. Se trata de un formato vectorial de almacenamiento digital donde se guarda la localización de los elementos geográficos y los atributos asociados a ellos. Un shapefile puede almacenar una cantidad de información descriptiva de un mapa impresionante. Existe un shapefile disponible para cualquier región y es posible encontrar varios sitios en la web que permiten acceder gratuitamente a sus shapefiles.

2. Páginas web que ofrecen datos GIS gratis

A continuación se muestra un listado de páginas web que nos ofrecen datos GIS gratis.

GADM es una base de datos espacial con la ubicación de las áreas y límites administrativos del mundo. Las áreas administrativas de esta base de datos son los países y las menores subdivisiones de nivel tales como comunidades autónomas, provincias, comarcas, municipios, departamentos. GADM describe dónde están estas áreas administrativas (los “elementos espaciales”), y para cada área proporciona algunos atributos, como el nombre y las variantes de los nombres.

Natural Earth. Proporciona una solución para la creación de mapas a escalas pequeñas: disponible a escalas 1:10m, 1:50m y 1:110m sirve para hacer mapas mundiales, regionales y de país. Se incluyen capas de información tanto políticas (fronteras administrativas), como físicas (naturales).



Características principales:

- Las capas vectoriales incluyen nombres y otros atributos.
- Los polígonos grandes están divididos para un manejo de los datos más eficiente, tales como capas de batimetría.
- Los datos raster incluyen capas de relieve sombreado en escala de grises y tintas hipsométricas
- Sin licencia: Los datos son públicos para realizar cualquier proyecto.

OpenStreetMap es un mapa del mundo creado por la comunidad de usuarios. Ha crecido hasta convertirse en una de las fuentes de datos cartográficos a escala local más detallados que existen. Estos datos cartográficos son creados y mantenidos por miles de voluntarios de todo el mundo, de manera similar a cómo se mantiene la enciclopedia en línea, wikipedia.



El aspecto más visible de OSM es la interfaz web basada en teselas, accesible a través de <http://osm.org>. Los mapas también se pueden visualizar, importar o editar en otras aplicaciones como Quantum GIS, OpenLayers, ArcGIS o aplicaciones específicas de OSM.

Características principales:

- Los datos vectoriales están etiquetados con nombres y otros atributos.
- Es posible extraer subconjuntos locales de los datos.
- Los datos son almacenados en el sistema de referencia espacial WGS84 (latitud-longitud).
- Etiquetado enriquecido, normalmente mucho más detallado que en cualquier otra fuente de datos.
- Licencia Creative Commons CC-By-SA.

Geofabrik. Este servidor tiene extractos de datos del proyecto OpenStreetMap que normalmente se actualiza todos los días. Se selecciona el continente y luego el país de la lista. Este servicio de descarga se ofrece de forma gratuita por Geofabrik GmbH.

El acceso a la página es <http://download.geofabrik.de/europe/spain.html>

DIVA-GIS. Se pueden descargar gratis datos geográficos de cualquier país del mundo. Incluye áreas administrativas, aguas continentales, carreteras y vías férreas, elevación, cobertura de la tierra, población y datos climáticos.

Probablemente se trata del sitio donde es más fácil obtener un conjunto de datos de un país específico.

Centro de Descargas del IGN es la página **web española del CNIG** desde donde se pueden descargar toda la información geográfica digital generada por el Instituto Geográfico Nacional. La página web es la siguiente: <http://centrodedescargas.cnig.es/CentroDescargas/>

El catálogo de productos disponibles es muy amplio:

El Equipamiento Geográfico de Referencia Nacional tiene carácter libre y gratuito (incluidos usos comerciales), siempre que se mencione el origen y propiedad de los datos. El EGRN incluye:

- * Vértices Geodésicos y Ptos. Nivelación
- * División Administrativa
- * Nomenclátor Geográfico Básico
- * Referencias Geográficas Entidades Locales

Para el resto de datos se permite el uso no comercial, mencionando origen y propiedad (© Instituto Geográfico Nacional de España). Para usos comerciales, se ha de firmar una licencia de uso ad hoc con el IGN.

UNEP GEOData. Una amplia gama de datos del Programa de Medio Ambiente de las Naciones Unidas, incluyendo la cobertura forestal mundial, la evapotranspiración potencial global, las temperaturas globales mensuales, presas, límites de cuenca y mucho más. La página de acceso es la siguiente: <http://geodata.grid.unep.ch/index.php>

Para obtener los datos, selecciona “More” y en el desplegable superior “Geospatial Data Sets”.

Global Map. Un conjunto de capas GIS que cubren todo el globo con una resolución de 1 kilómetro. Incluye: transporte, elevación, drenaje, vegetación, límites administrativos, cobertura del suelo, uso del suelo y los centros de población. Producido por el Comité Directivo Internacional de Cartografía Mundial. La página de acceso es la siguiente: <http://www.iscgm.org/>

MapCruzin. En esta página se pueden descargar mapas, shapefiles y datos geospaciales. MapCruzin es un intermediario de los datos, no crean los datos, sino que utilizan fuentes privadas u organismos. Incluyen datos sobre transporte, usos del suelo, climatología, desastres naturales. La cartografía política utiliza los datos de CloudMade. La página web de acceso es la siguiente: <http://www.mapcruzin.com/>

Aunque existen datos a escala global se centra en los Estados Unidos.

3. Paquetes de R para trabajar con datos espaciales

Caben destacar los siguientes paquetes:

sp Trabaja con datos vectoriales (puntos, líneas y polígonos).

raster trabaja con datos raster (matrices georeferenciadas, típicas en imágenes de satélite y similares). Define funciones para crear, leer, manipular y escribir este tipo de datos.

gstat es el paquete por excelencia para hacer análisis geoestadístico en R. Permite hacer modelado de variogramas, interpolación IDW, kriging ordinario, y kriging con regresión.

rgdal es una interface para GDAL. Entre otras cosas, permite transformar proyecciones. Complementa a sp y raster.

mapproj proporciona funciones para leer y manipular datos tipo “shapefile”. Es un buen complemento para sp.

rasterVis complementa al paquete raster, para visualización de datos raster.

4. Datos de GADM Global Administrative Areas

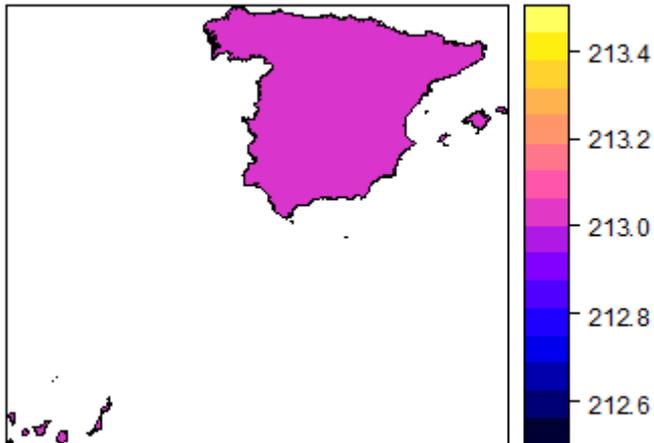
GADM contiene información sobre divisiones administrativas a diferentes niveles. Permite descargar información en diferentes formatos: shapefile, ESRI geodatabase, Google Earth .kmz y Rdata. Este conjunto de datos está disponible gratuitamente para usos académicos y otros usos no comerciales. La redistribución, o el uso comercial, no está permitido sin autorización previa.

Lo que hay que hacer es disponer de un objeto de R con los datos del mapa. Estos datos los vamos a obtener de <http://www.gadm.org/country> Seleccionamos el país y el formato que deseamos descargar; en nuestro caso con formato Rdata. Para ilustrar vamos a descargar los mapas de España para tres niveles. Los hemos descargado a nuestro equipo (también se podría acceder desde la web). Una vez tengamos los archivos con su formato, su extensión y demás ya podemos trabajar con ellos. En nuestro caso vamos a trabajar con el paquete *sp* “spatial data” de R:

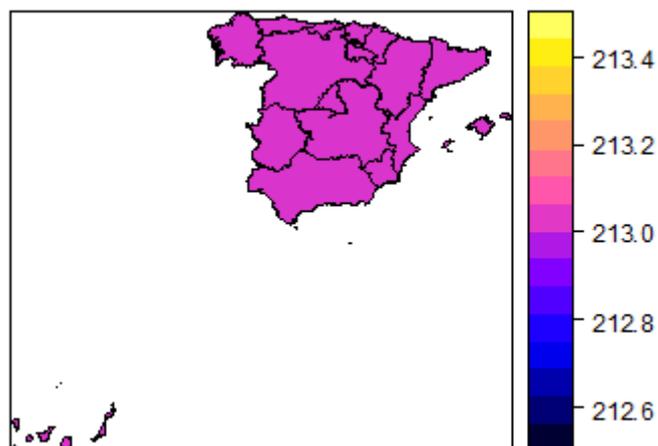
```
## Si no tenemos instalado el paquete "sp" lo instalamos
```

```
## install.packages("sp")
library(sp)
```

```
## Representacion de España a nivel Nacional
## nos bajamos el fichero "ESP_adm2.RData" a un directorio local
ub_espana="C:/r/proyectos/espacial/ESP_adm0.RData"
## Creamos los objetos de R
load(ub_espana)
espana=gadm
#Seleccionamos un objeto de mapa
mapa = espana[1]
splot(mapa)
```



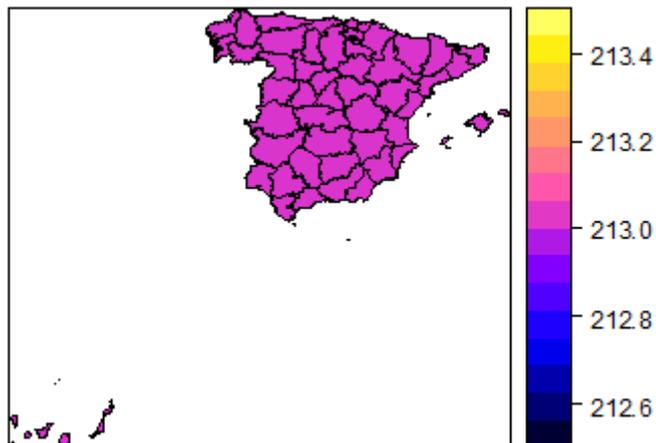
```
## Representacion de España a nivel Comunidad Autónoma
ub_espana="C:/r/proyectos/espacial/ESP_adm1.RData"
## Creamos los objetos de R
load(ub_espana)
espana=gadm
## Para conocer el Nombre de las CCAA
gadm$NAME_1
splot(mapa)
```



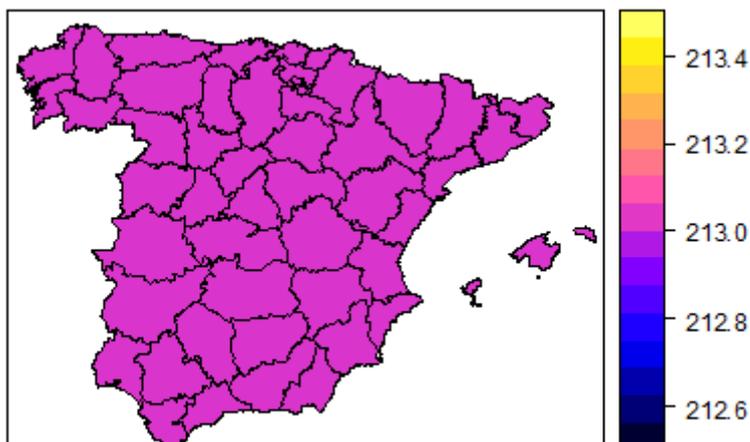
```
## Ponemos unos límites
splot(mapa,xlim=c(-9.5, 4.5), ylim=c(36,44))
```



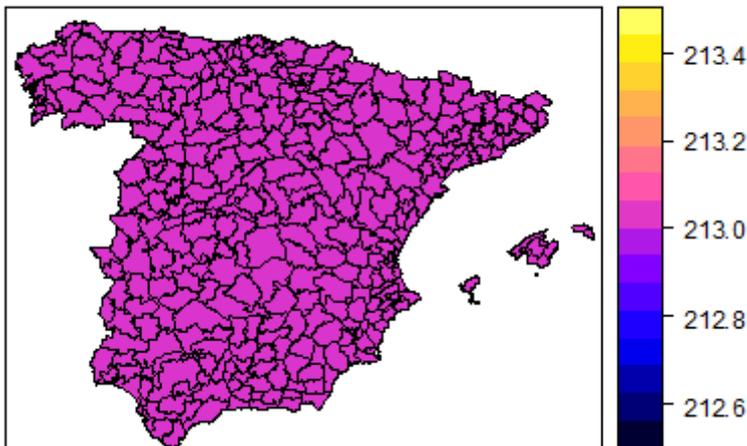
```
## Representacion de España a nivel Provincial
ub_espana="C:/r/proyectos/espacial/ESP_adm2.RData"
## Creamos los objetos de R
load(ub_espana)
espana=gadm
## Para conocer el Nombre de las CCAA de las provincias
gadm$NAME_1
## Para conocer el Nombre de las provincias
gadm$NAME_2
## Seleccionamos un objeto de mapa
mapa = espana[2]
splot(mapa)
```



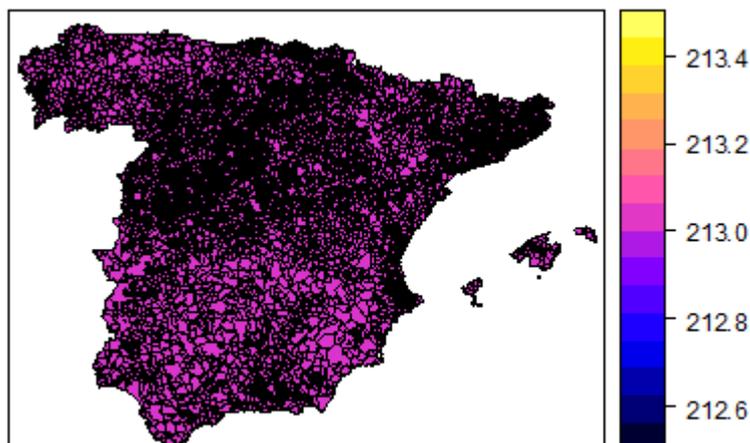
```
#Ponemos unos límites
splot(mapa,xlim=c(-9.5, 4.5), ylim=c(36,44))
```



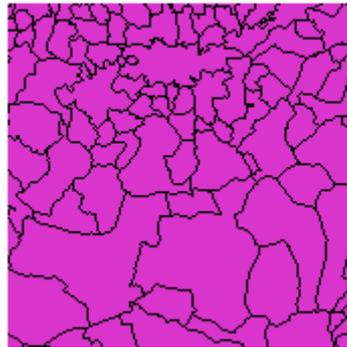
```
## Representacion de España a nivel comarcal
ub_espana="C:/r/proyectos/espacial/ESP_adm3.RData"
#Creamos los objetos de R
load(ub_espana)
espana=gadm
## Para conocer el Nombre de las CCAA de las comarcas
gadm$NAME_1
## Para conocer el Nombre de las provincias de las comarcas
gadm$NAME_2
## Para conocer el Nombre de las comarcas
gadm$NAME_3
#Seleccionamos un objeto de mapa
mapa = espana[2]
#Ponemos unos límites
splot(mapa,xlim=c(-9.5, 4.5), ylim=c(36,44))
```



```
## Representacion de España a nivel municipal
ub_espana="C:/r/proyectos/espacial/ESP_adm4.RData"
#Creamos los objetos de R
load(ub_espana)
espana=gadm
## Para conocer el Nombre de las CCAA de los municipios
gadm$NAME_1
## Para conocer el Nombre de las provincias de los municipios
gadm$NAME_2
## Para conocer el Nombre de las comarcas de los municipios
gadm$NAME_3
## Para conocer el Nombre de los municipios
gadm$NAME_4
#Seleccionamos un objeto de mapa
mapa = espana[2]
#Ponemos unos límites
splot(mapa,xlim=c(-9.5, 4.5), ylim=c(36,44))
```



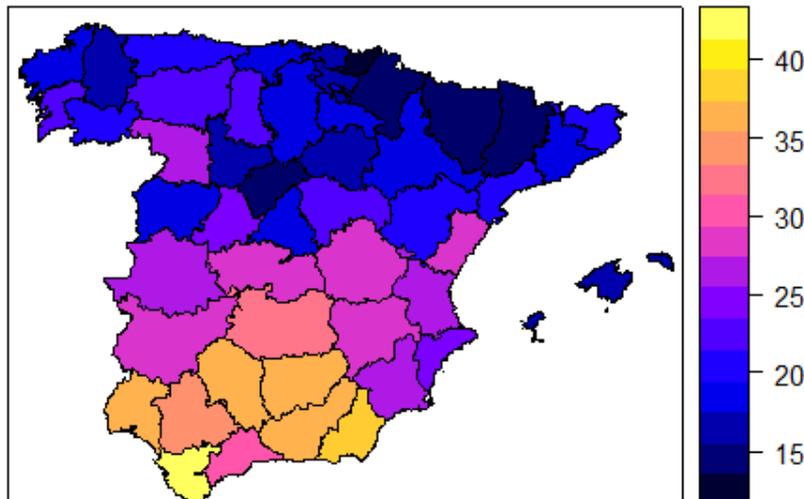
```
##Ponemos otros límites
splot(mapa,xlim=c(-4.5, -3.5), ylim=c(39,40))
```



4.1. Mapas temáticos territoriales

Hasta ahora nos hemos limitado a representar los límites geográficos de a nivel estatal, de comunidad autónoma, provincial, comarcal y municipal. La creación der mapas temáticos territoriales es muy sencilla ya que solamente hay que dar los valores a cada polígono identificado. Ahora bien hay que cerciorarnos de que los datos corresponden al territorio que queremos representar. Como ejemplo vamos a representar unos datos provinciales (en concreto la tasa de paro provincial española del tercer trimestre de 2014, datos obtenidos del Instituto Nacional de Estadística INE). Estos datos se han manipulado para que coincidan los índices provinciales de “ESP_adm2.RData” con los índices del INE; ya que són diferentes. El fichero “tasaparoprov.csv” contiene los datos del INE ordenados según vienen dados en “ESP_adm2.RData”.

```
## Representacion de espana a nivel Provincial
ub_espana="C:/r/proyectos/espacial/ESP_adm2.RData"
#Creamos los objetos de R
load(ub_espana)
espana=gadm
#Seleccionamos un objeto de mapa
mapa = espana[1]
#Añadimos los valores que deseamos representar
setwd("c:/R/DATA")
datos.csv <- read.csv("tasaparoprov.csv", sep=";",dec = ",",header = T)
paro<-datos.csv[,3]
mapa@data=data.frame(paro)
splot(mapa,c("paro"),xlim=c(-9.5, 4.5), ylim=c(36,44))
```



Se ha realizado este ejemplo para ilustrar las posibilidades de R para representar mapas temáticos territoriales. Como puede observarse las posibilidades que nos da R para representar datos territoriales son infinitas ya que podremos representar cualquier parámetro (tasa de paro, riqueza, nº habitantes, incidencia de cáncer, nº de animales, terreno cultivado etc, etc, etc) a cualquier nivel territorial (mundial, estatal, comunidad autónoma, provincial, comarcal, municipal, distrito etc).

5. Datos de DIVA-GIS con nombres provinciales

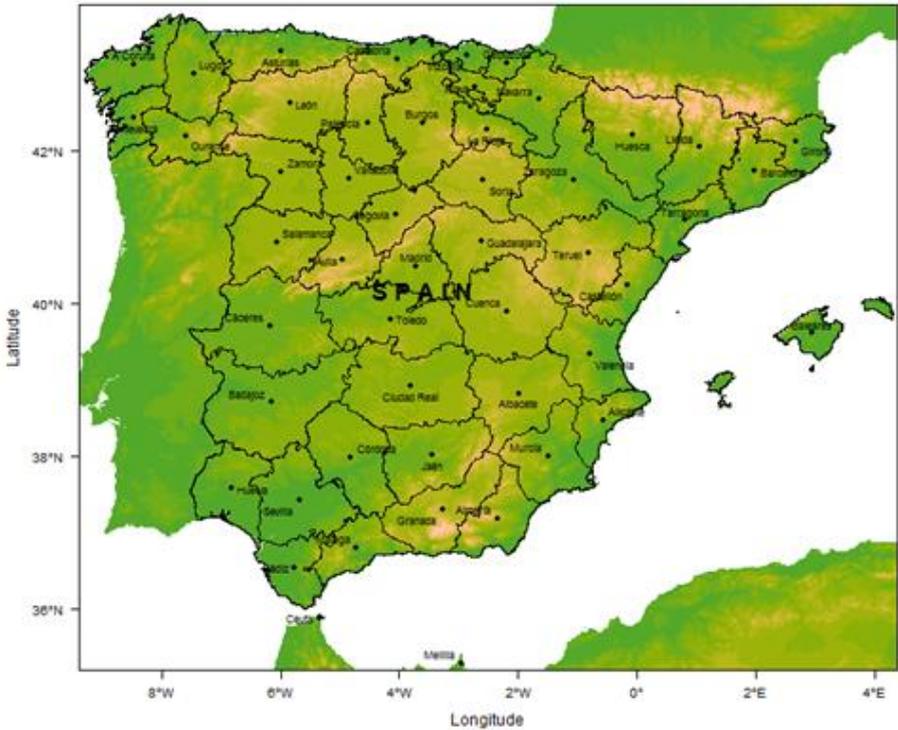
DIVA-GIS contiene información sobre divisiones administrativas a diferentes niveles. Incluye además aguas continentales, carreteras y vías férreas, elevación, cobertura de la tierra, población y datos climáticos.

```
#####
## mapas Físicos
#####
library(raster)
library(rasterVis)
library(maptools)
library(rgeos)
library(latticeExtra)
library(Colorspace)
## proyeccion Longitud-Latitud
proj <- CRS('+proj=longlat +ellps=WGS84')
#####
## Retrieving data from DIVA-GIS, GADM and Natural Earth Data
#####
setwd("c:/R/DATA/ESP_adm")
EspAdm <- readShapePoly('ESP_adm2.shp', proj4string=proj)
Encoding(levels(EspAdm$NAME_2)) <- 'latin1'
spplot(EspAdm[1], xlim=c(-9.5, 4.5), ylim=c(36,44))
EspDEM <- raster('ESP_alt')
spplot(EspDEM, xlim=c(-9.5, 4.5), ylim=c(36,44))
#####
## Intersection of shapefiles and elevation model
#####
## Define the extent of Spain as an SpatialPolygons
extEsp <- as(extent(EspDEM), 'SpatialPolygons')
proj4string(extEsp) <- proj
#####
## Etiquetas o nombres
#####

## Localización de las etiquetas de cada polígono
centroids <- coordinates(EspAdm)
xySpain <- apply(centroids, 2, mean)
admNames <- strsplit(as.character(EspAdm$NAME_2), ' ')
admNames <- sapply(admNames,
  FUN=function(s){
    sep=if (length(s)>2) '\n' else ' '
    paste(s, collapse=sep)
  })
#####
## Overlaying layers of information
#####
terrainTheme <- rasterTheme(region=terrain_hcl(15))
altPlot <- levelplot(EspDEM, par.settings=terrainTheme,
  maxpixels=1e6, panel=panel.levelplot.raster,
  margin=FALSE, colorkey=FALSE)

altPlot + layer({
  ## Areas administrativas
  sp.polygons(EspAdm, col='black', lwd=0.2)
  ## Centroides de áreas administrativas
  panel.points(centroids, col='black')
```

```
## ... con sus etiquetas
panel.pointLabel(centroids, labels=admNames,
                 cex=0.7, fontfamily='Palatino', lineheight=.8)
## Nombre del país
panel.text(xySpain[1], xySpain[2], labels='S P A I N',
          cex=1.5, fontfamily = 'Palatino', fontface=2)
})
```



```
library(raster)
library(rasterVis)
library(maps)
library(mapdata)
library(maptools)

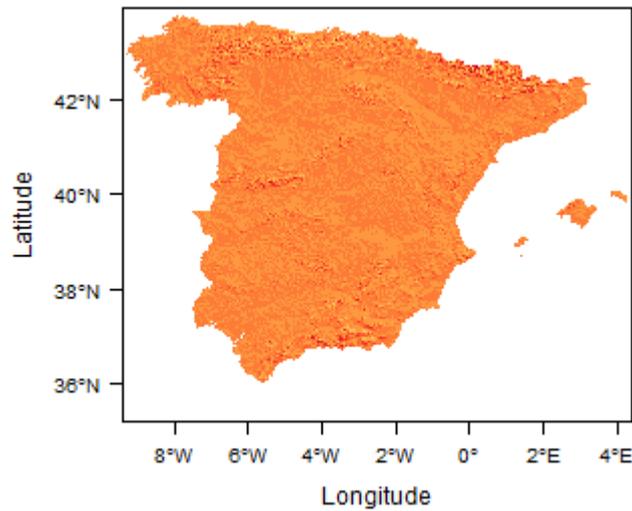
#####
## Nos descargamos el fichero "ESP_msk_alt.zip" de
## "http://www.diva-gis.org/gdata" Country "SPAIN"
## country "Elevation (country mask)" en nuestro directorio local
## y lo descomprimimos
#####

#####
## vamos al directorio local
#####
setwd("c:/R/DATA")

DEM <- raster('ESP_msk_alt')

slope <- terrain(DEM, 'slope')
aspect <- terrain(DEM, 'aspect')
hs <- hillshade(slope=slope, aspect=aspect,
               angle=20, direction=30)
hsTheme <- modifyList(GrTheme(), list(regions=list(alpha=0.6)))
levelplot(hs, par.settings=hsTheme, maxpixels=1e6, margin=FALSE,
          colorkey=FALSE)
```

```
## hillshade theme:
levelplot(hs,col=rainbow(10, alpha=0.25), maxpixels=1e6,colorkey =
FALSE,margin=FALSE)
```



```
plot3D(DEM, maxpixels=5e4,colorkey = FALSE,margin=FALSE)
## dev.off()
```



6. Poner las Islas Canarias junto con la península

Si tenemos dos áreas separadas geográficamente y las queremos poner juntas como es el caso de la península Ibérica y las islas Canarias lo que hay que hacer es cambiar de coordenadas los centroides de los polígonos que definen una de las áreas; en nuestro caso las islas canarias. Procedemos de la siguiente forma:

```
library(sp)
library(maptools)
MapaEsp <- readShapePoly(fn="c:/R/DATA/MUNI/esp_muni_0109")
plot(MapaEsp)
Encoding(levels(MapaEsp$NOMBRE)) <- "latin1"
provincias <- readShapePoly(fn="c:/R/DATA/MUNI/spain_provinces_ag_2")
plot(provincias)
```

```
## dissolve repeated polygons
espana <- unionSpatialPolygons(MapaEsp, MapaEsp$PROVMUN)
## Extract Canarias islands from the SpatialPolygons object
canarias <- sapply(espana@polygons, function(x)substr(x@ID, 1, 2) %in%
c("35", "38"))
peninsula <- espana[!canarias]
Islas <- espana[canarias]
## shift the island extent box to position them at the bottom right corner
dy <- bbox(peninsula)[2,1] - bbox(Islas)[2,1]
dx <- bbox(peninsula)[1,2] - bbox(Islas)[1,2]
Islas2 <- elide(Islas, shift=c(dx, dy))
bbIslands <- bbox(Islas2)
## Bind Peninsula (without islands) with shifted islands
espana <- rbind(peninsula, Islas2)
plot(espana)
```



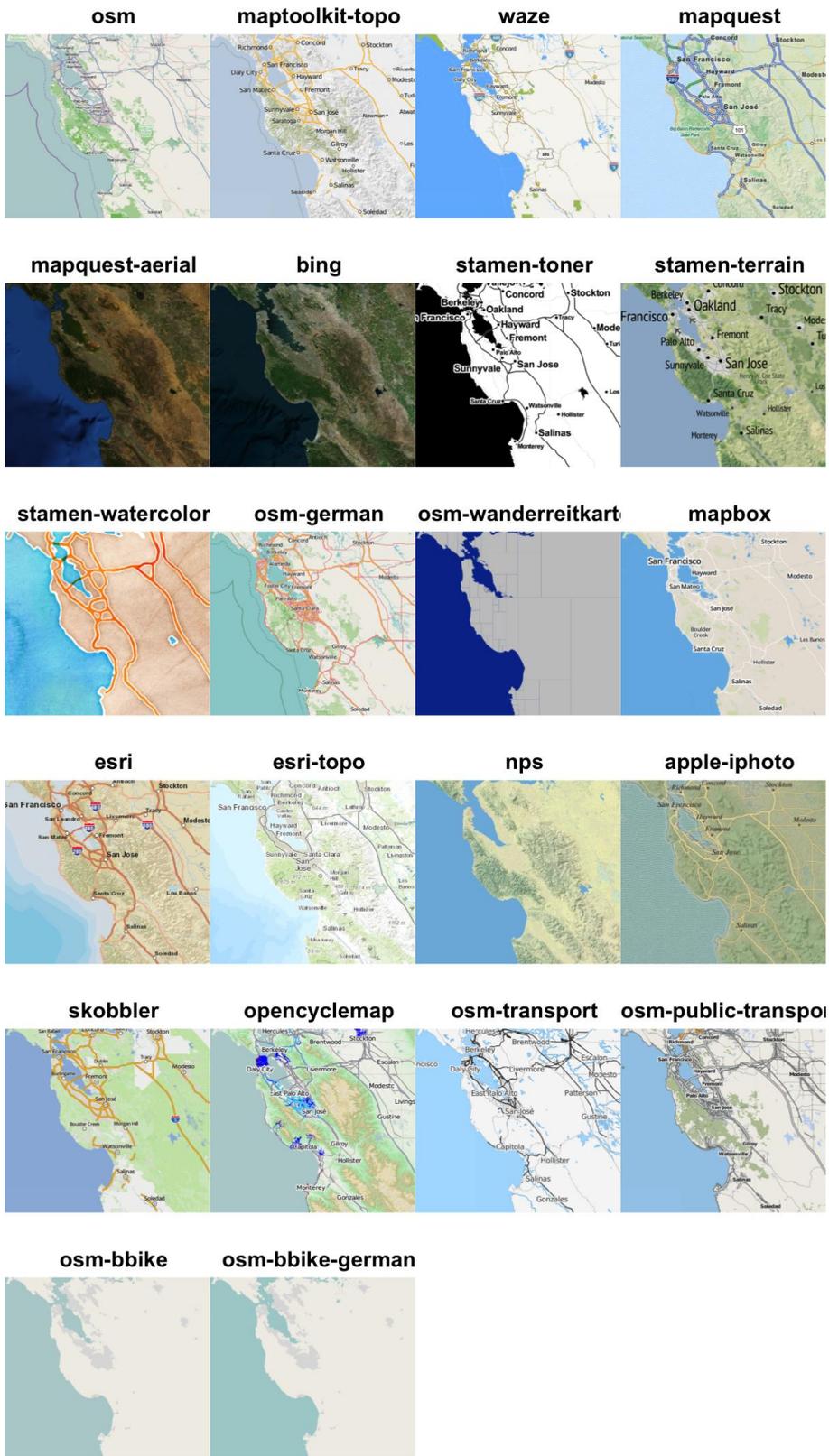
7. Mapas generados con OpenStreetMap

OpenStreetMap (también conocido como OSM) es un proyecto colaborativo para crear mapas libres y editables.

Los mapas se crean utilizando información geográfica capturada con dispositivos GPS móviles, ortofotografías y otras fuentes libres. Esta cartografía, tanto las imágenes creadas como los datos vectoriales almacenados en su base de datos, se distribuye bajo licencia abierta Open Database License (ODbL).

En R se ha desarrollado el paquete “OpenStreetMap” para trabajar con este tipo de datos.

A continuación se muestra los tipos de mapas que podemos generar con dicho paquete.



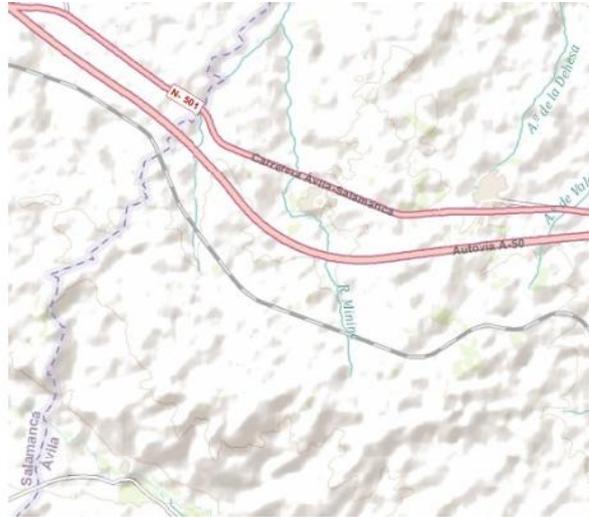
```
require("OpenStreetMap")
#####
#####
##
## Selecciona los límites del plano:
## - Izquierda superior
## - Derecha inferior
##
## Selecciona el tipo de mapa que quieres Representar:
##
## 1. "osm" OpenStreet map (típico)
## 2. "osm-bw" OpenStreet map (básico)
## 3. "maptoolkit-topo" topológico....
## .....
##

type.map=c("osm", "osm-bw", "maptoolkit-topo", "waze", "mapquest",
"mapquest-aerial", "bing", "stamen-toner", "stamen-terrain",
"stamen-watercolor", "osm-german", "osm-wanderreitkarte",
"mapbox", "esri", "esri-topo", "nps", "apple-iphoto",
"skobbler", "cloudmade-", "hillshade", "opencyclemap",
"osm-transport", "osm-public-transport", "osm-bbike",
"osm-bbike-german")

##
## Mapa de Madrid
##
map=openmap(c(40.5, -3.75), c(40.35, -3.58),type=type.map[1])
plot(map)
```



```
##
## Mapas de Gimialcón
##
map=openmap(c(40.90, -5.17), c(40.84, -5.08),type=type.map[15])
plot(map)
```



```
map=openmap(c(40.90, -5.17), c(40.84, -5.08),type=type.map[2])  
plot(map)
```



```
map=openmap(c(40.90, -5.17), c(40.84, -5.08),type=type.map[3])  
plot(map)
```



7.1. Añadir puntos y texto a un mapa

Una de las características de R es poder añadir puntos de una localización concreta y texto dentro del mapa previamente hay que hacer la transformación al elipsoide correspondiente.

```
map=openmap(c(40.90, -5.17), c(40.84, -5.08),type=type.map[1])
plot(map)
## Añadimos dos puntos al mapa
xy.1=c(-5.1,40.86)
xy.2=c(-5.13,40.88)
# Create a spatial object and define the coordinate reference system
xy=rbind(xy.1,xy.2)
xya=SpatialPoints(xy)
proj4string(xya)=CRS("+proj=longlat +ellps=WGS84")
xyb=spTransform(xya,CRS("+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0
+lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m
+nadgrids=@null +no_defs"))

## Dibujamos los dos puntos en el mapa
plot(map,raster=TRUE)
plot(xyb[1,],add=TRUE,col="darkred",lwd=4)
plot(xyb[2,],add=TRUE,col="darkgreen",lwd=2)

## Añadimos texto
xy=coordinates(xyb)
names.xy=c("punto 1","punto 2")
text(xy[1,1],xy[1,2],names.xy[1],adj=1,pos=3,lwd=1.5)
text(xy[2,1],xy[2,2],names.xy[2],adj=1,pos=2,lwd=1.5)
```

