

```
#ROXANA BEATRIZ RONQUILLO UMAA
```

```
#CARTOGRAFIA EN R
```

```
#EJEMPLO 1: Se representa con un punto rojo la capital de Francia.
```

```
#Cargamos la libreria
```

```
library(maptools)
```

```
## Warning: package 'maptools' was built under R version 3.2.2
```

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

```
## Warning: package 'sp' was built under R version 3.2.2
```

```
## Checking rgeos availability: FALSE
```

```
## Note: when rgeos is not available, polygon geometry computations  
in maptools depend on gpclib,
```

```
## which has a restricted licence. It is disabled by default;
```

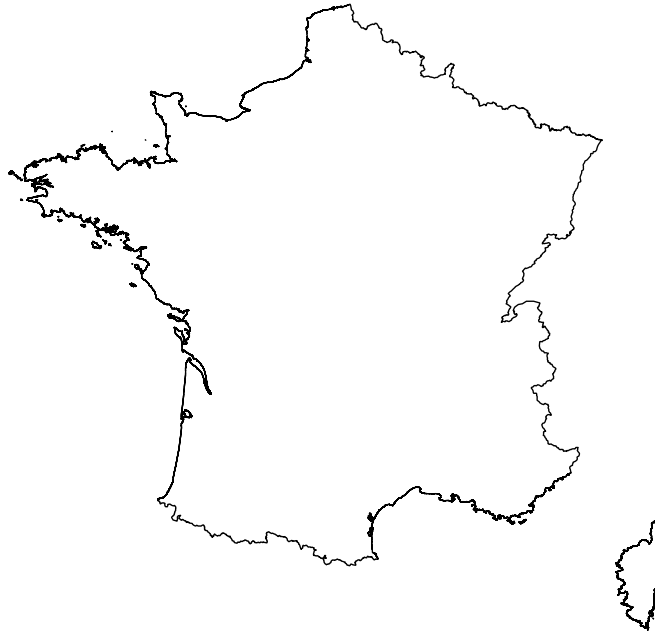
```
## to enable gpclib, type gpclibPermit()
```

```
#Se cargan los datos obtenidos de www.gadm.org
```

```
france<-readShapeSpatial("FRA_adm0.shp",proj4string=CRS("+proj=longlat"))
```

```
#mostramos el grafico
```

```
plot(france)
```



```
#Se grafica el punto correspondiente a la capital de Francia (Paris)  
plot(france,xlim=c(1,4),ylim=c(41.5,51))  
points(2.349,48.853,pch=20,col="red",cex=2)
```



```
#EJEMPLO 2: Zona de los andes.  
  
# cargamos libreria  
library("XML")  
  
## Warning: package 'XML' was built under R version 3.2.2  
  
# primer data frame  
theurl <- "http://www.peaklist.org/WWlists/ultras/andes1.html"  
tables <- readHTMLTable(theurl)  
n.rows <- unlist(lapply(tables, function(t) dim(t)[1]))  
df<-tables[[which.max(n.rows)]]  
df<-df[3:nrow(df),c(2,3,4,7,8)]  
names(df)=c("cumbre", "pais", "altura_m", "lat_geo", "lng_geo")  
df1<-na.omit(df)  
# segundo data frame
```

```

theurl <- "http://www.peaklist.org/WWlists/ultras/andes2.html"
tables <- readHTMLTable(theurl)
n.rows <- unlist(lapply(tables, function(t) dim(t)[1]))
df<-tables[[which.max(n.rows)]]
df<-df[3:nrow(df),c(2,3,4,7,8)]
names(df)=c("cumbre", "pais", "altura_m", "lat_geo", "lng_geo")
df2<-na.omit(df)

# tercer data frame
theurl <- "http://www.peaklist.org/WWlists/ultras/andes3.html"
tables <- readHTMLTable(theurl)
n.rows <- unlist(lapply(tables, function(t) dim(t)[1]))
df<-tables[[which.max(n.rows)]]
df<-df[3:nrow(df),c(2,3,4,7,8)]
names(df)=c("cumbre", "pais", "altura_m", "lat_geo", "lng_geo")
df3<-na.omit(df)

# unimos los tres data frame en uno y convertimos las columnas de altura a entero y las coord
df<-rbind(df1,df2,df3)
df$altura_m<-as.integer(as.character(df$altura_m))
df$lat_geo<-as.character(df$lat_geo)
df$lng_geo<-as.character(df$lng_geo)

# creamos esta funcin para transformar las coordenadas geogrficas a decimal
geo2dec<-function(c) {
  z<-sapply( strsplit(c, "[\\'\\"]"), as.character )
  dec<- as.numeric(z[1, ])+ as.numeric(z[3, ])/60 + as.numeric(z[4, ])/3600
  if (z[1, ]=="N"||z[1, ]=="E") dec else -dec
}

# agregamos dos columnas con la latitud y longitud en coordenadas decimales
df$latitude<-geo2dec(df$lat_geo)
df$longitude<-geo2dec(df$lng_geo)

# cargamos libreria
library("ggmap")

## Warning: package 'ggmap' was built under R version 3.2.2
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.2.2

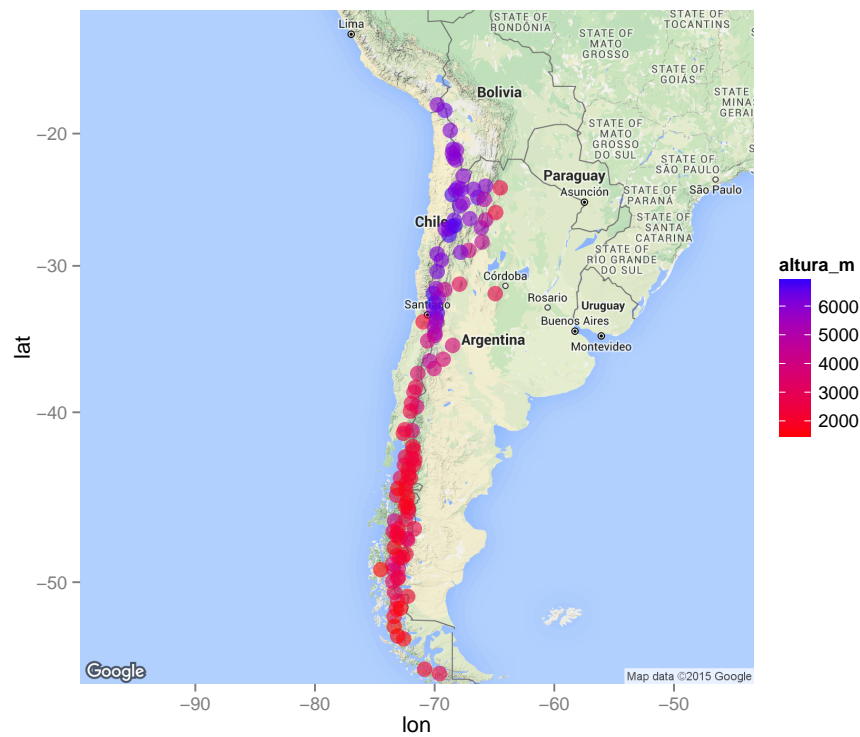
# descargamos el mapa de Chile desde Google
map <- get_map(location = 'Chile', zoom = 4, maptype = "terrain")

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Chile&zoom=4&size=640x480
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Chile&sensor=false

```

```
# pintamos los puntos de nuestro data frame en el mapa
andes <- ggmap(map) + geom_point(data=df, aes(x=longitude, y=latitude, colour=altura_m), alpha=0.5)

# agregamos una leyenda y escala coloreada
andes <- andes + scale_colour_continuous(low = "red", high = "blue", space = "Lab", guide = "colorbar")
# generamos el mapa
andes
```



```
#EJEMPLO 3: Provinvias de peru.

#Se carga la libreria
library(sp)
#Se descargan los datos de http://gadm.org/country
peru <- readRDS("PER_adm2.rds")
names(peru)
```

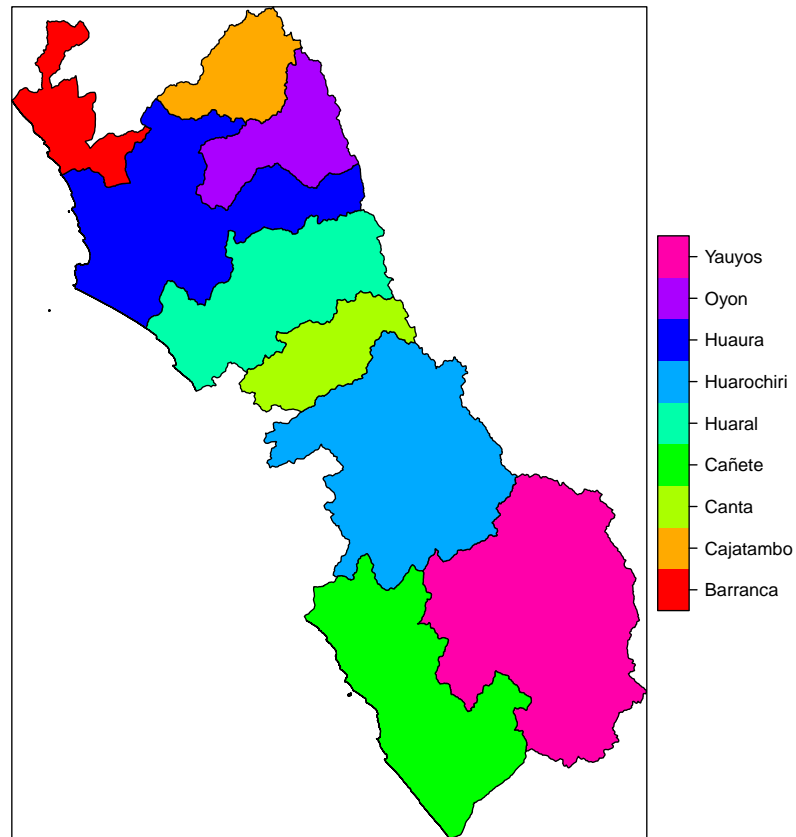
```
## [1] "OBJECTID" "ID_0" "ISO" "NAME_0" "ID_1"
## [6] "NAME_1" "ID_2" "NAME_2" "HASC_2" "CCN_2"
## [11] "CCA_2" "TYPE_2" "ENGTYPE_2" "NL_NAME_2" "VARNAME_2"

# El mapa y sus caractersticas estn cargados en gadm por provincias
mapalima <- peru[peru@data$NAME_1 == "Lima",]
# Seleccionamos solo los datos de la provincias
datalima <- data.frame(mapalima)
datalima[7];mapalima$NAME_2;

## ID_2
## 129 129
## 130 130
## 131 131
## 132 132
## 133 133
## 134 134
## 135 135
## 136 136
## 137 137
## [1] "Barranca" "Caete" "Cajatambo" "Canta" "Huaral"
## [6] "Huarochiri" "Huaura" "Oyon" "Yauyos"

# creamos el factor segn los nombres de las provincias elegidas
mapalima$provincias <- factor(mapalima$NAME_2)
# la funcin factor es para eliminar los niveles que no existen
col <- rainbow(length(levels(mapalima$provincias )))
# asignamos los colores segn los niveles
# mostramos el mapa
spplot(mapalima, "provincias", col.regions = col, main = "Lima Provincias")
```

Lima Provincias



```
#EJEMPLO 4: Se muestra mapa de El Salvador y se selecciona
#el departamento de Santa Ana
library(sp)
library(maptools)
library(maps)
SLV<-readShapeSpatial("SLV_adm1.shp",proj4string=CRS("+proj=longlat"))
names(SLV)

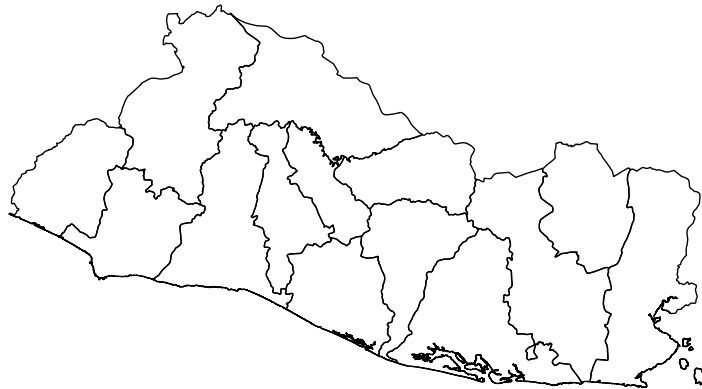
## [1] "ID_0"      "ISO"      "NAME_0"   "ID_1"     "NAME_1"
## [6] "HASC_1"   "CCN_1"    "CCA_1"    "TYPE_1"   "ENGTYPE_1"
## [11] "NL_NAME_1" "VARNAME_1"

SLV$NAME_1

## [1] Ahuachapn Cabaas Chalatenango Cuscatln La Libertad
## [6] La Paz La Unin Morazn San Miguel San Salvador
```

```
## [11] San Vicente Santa Ana Sonsonate Usulutn
## 14 Levels: Ahuachapn Cabaas Chalatenango Cuscatln ... Usulutn

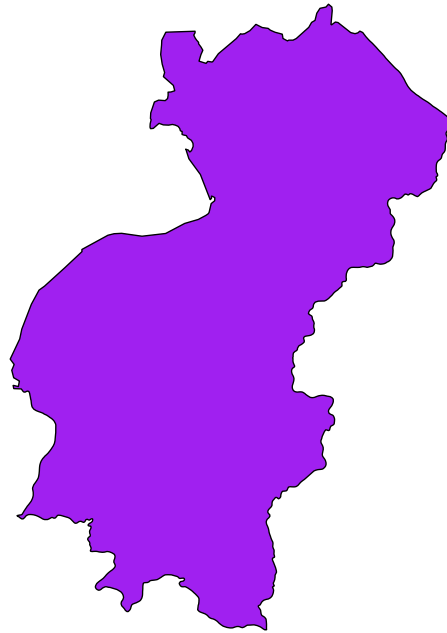
plot(SLV)
```



```
plot(SLV[SLV$NAME_1=="Santa Ana",],col="purple")
title("Santa Ana")
```



## Santa Ana



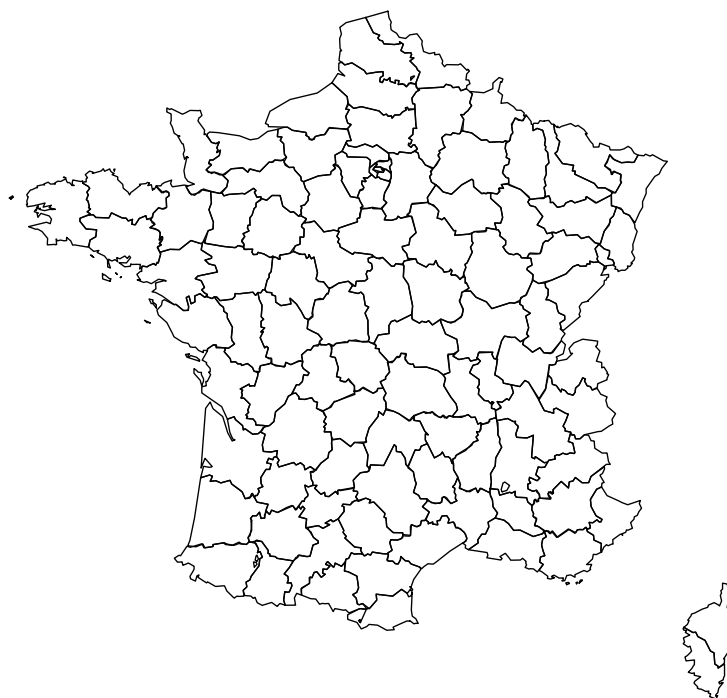
*#EJEMPLO 5: Departamentos de Francia.*

*#Cargamos la libreria maps.*

```
library(maps)
```

*#Se genera el mapa de francia*

```
france<-map(database="france")
```



```
#Pintamos de color morado "Ain", color azul "Marne",  
#color amarillo "Nord", color rojo "Charente".  
dpt2001<-c("Ain","Marne","Nord","Charente")  
col2001<-c(6,4,7,2)  
match <- match.map(france,dpt2001)  
color <- col2001[match]  
map(database="france", fill=TRUE, col=color)
```

