Visualización de información georeferenciada en ggplot2

Julio César Alonso

Sebastián Montenegro

No. 35

Septiembre de 2012

Apuntes de Economía

ISSN 1794-029X

No. 35

Editor

Julio César Alonso

jcalonso@icesi.edu.co

Asistente editorial

José Wilmar Quintero P.

jwquintero@icesi.edu.co

Gestión Editorial

Departamento de Economía - Universidad Icesi

www.icesi.edu.co

Tel: 5552334 ext: 8398. Fax: 5551441

Calle 18 # 122-135 Cali, Valle del Cauca, Colombia

Visualización de información georeferenciada en ggplot2

Julio César Alonso[*]
Sebastián Montenegro [**]
CIENFI - Departamento de Economía - Universidad Icesi
Cali - Colombia

19 de agosto de 2013

Resumen

Este documento presenta una breve introducción a la presentación de información georeferenciada en gráficos. Está dirigido a personas interesadas en la elaboración de herramientas visuales que permitan comparar diferentes zonas de un territorio con una misma variable usando ggplot en R. Se supone un conocimiento previo en el manejo del programa.
Palabras clave: Información georeferenciada, polígonos, R, mapas.

Abstract

This document presents an introdution to graphical tools for plotting georeferenced data. The document is appropiate for readers interested in presenting information to compare between territories using a variable, employing R and ggplot2 llibrary. We supose a previusly knowledge of R enviroment.
Key Words: Georeferenced data, ggplot2, R-project, polygons.

[*]Director del centro de investigación en economía y finanzas (CIENFI) y director académico de la maestría en Economía de la Universidad Icesi

[**]Estudiante en práctica del CIENFI y estudiante de último semestre de Economía en la Universidad Icesi.

Objetivos de Aprendizaje

Al finalizar la lectura de este documento se espera que el lector esté en capacidad de:

- Manipular y organizar los archivos y bases de datos necesarias para graficar información georeferenciada en R.
- Graficar información georeferenciada en R usando ggplot2.

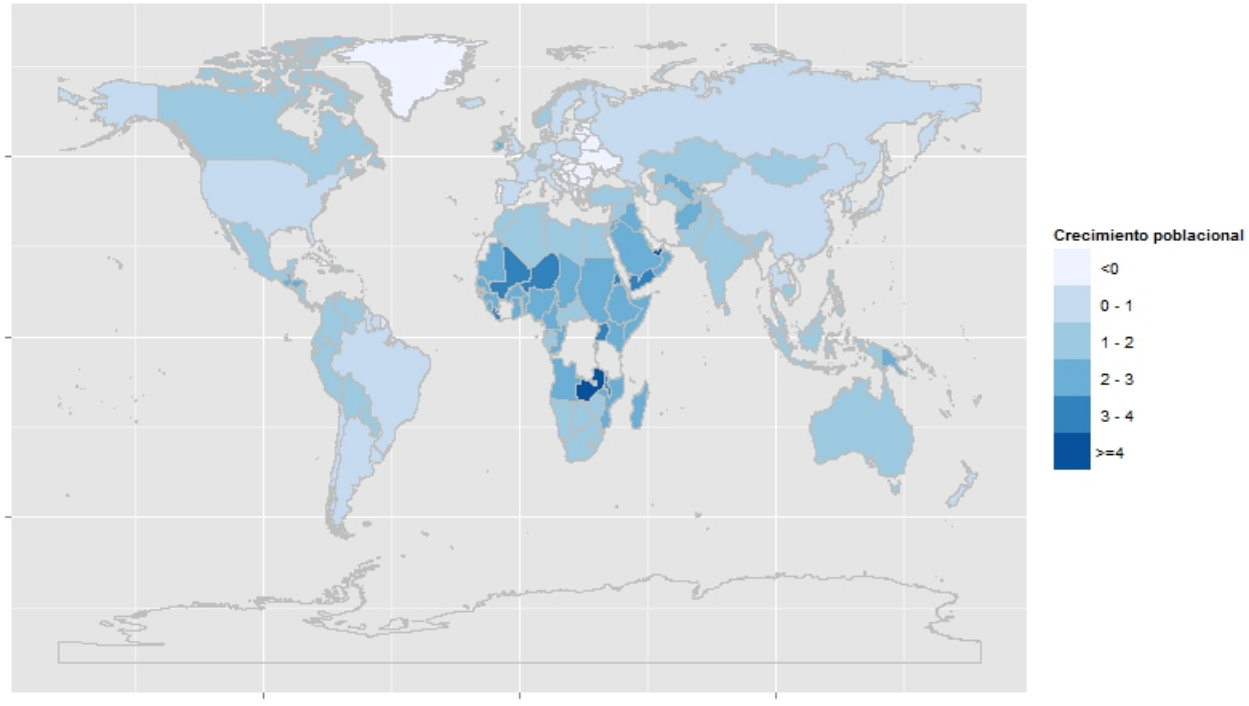
1. Introducción

Cada día es más común emplear gráficos para presentar la información, en parte debido a que es más fácil entender un problema teniendo una adecuada visualización del mismo (Zachry y Thralls (2004)). Con anterioridad, Alonso y González (2012) presentaron un documento para elaborar gráficos usando Ggplot2 en R. Este documento pretende complementar los anteriores autores, mostrando cómo se puede emplear dicha herramienta para construir mapas que permitan presentar información georeferenciada.
R es una herramienta gratutita útil para procesar información, la cual es usada en las diferentes áreas del conocimiento. Para descargarlo, puede acceder a la página del repositorio `http://cran.r-project.org/`. Por otro lado, para emplear R es recomendable usar la interfaz de R Studio para facilitar la comunicación del usuario con el programa; para descargarla puede acceder a:

`http://www.rstudio.com/`.

Por otro lado, ggplot es un paquete de R que permite hacer gráficas de diferentes tipos dependiendo de las necesidades del usuario. Una de esas gráficas son los mapas que incluyen información georeferenciada, los cuales permiten procesar la información y presentarla de tal forma que se pueden comparar diferentes zonas con respecto a la variable en referencia. Es decir, los mapas permiten observar cómo una variable cambia entre países, departamentos, ciudades o zonas.
En la figura 1 se presenta un ejemplo de mapa que permite comparar la tasa de crecimiento poblacional en 2011 entre los países del mundo.

Figura 1: Tasa de crecimiento poblacional. 2011.



Fuente: Banco Mundial

Como se puede observar, estos mapas facilitan el análisis de la información georeferenciándola. Esto resume la información en un sólo gráfico permitiendo responder a preguntas como ¿cuáles países tienen la población que crece más rápidamente? ¿cuáles países tienen la tasa de crecimiento más pequeña? De esta manera, es posible tomar decisiones y posibles conclusiones en problemas que requieran de análisis de datos espaciales.

Este documento presenta una breve introducción para entender el código de programación necesario para grafica mapas en ggplot. En este orden de ideas, se procederá a explicar dicha codificación con aplicaciones para el mundo, Colombia, Valle del Cauca y Cali, que podrán ser replicadas por el lector.

2. Mapas del mundo

Lo primero que se debe hacer es decargar los polígonos que le permiten a R marcar la forma de los países y sus límetes en forma de coordenadas. Es decir, se requiere de un archivo que contenga la información cartográfica y divisiones políticas del mundo.
Existen varias páginas que ponen a disposición los archivos de polígonos, lo cuales generalmente vienen en un archivo .zip. Para descargarlo puede ingresar a `http://www.mappinghacks.com/data/` y hacer click en el link correspondiente al archivo TMWORLDBORDERS-0.2.zip. Descargue el archivo y guárdelo. Lo más recomendable es ubicar los polígonos en la misma carpeta que empleará como directorio de trabajo (working directory) en R. Una vez descargada, puede proceder a descomprimirla. Obtendrá cuatro archivos con el mismo nombre pero con diferente extensión: `.dbf`, `.prj`, `.shp` y `.shx`. Ahora podemos comenzar a programar el código en R para graficar información en el mapa.
Los paquetes que requerirá para cargar los polígonos y hacer el mapa serán: sp, maps, rgeos, ggplot2, rgdal, gpclib y maptools. Para instalarlos puede hacer click en la pestaña de Rstudio "`Packages`" luego en "`Install Packages`" escribir el nombre de los paquetes que descargará uno por uno. Tambien puede descargarlos escribiendo el comando:

```
>install.packages("nombre del paquete")
```

Recuerde que cada vez que cierre el programa deberá cargar el paquete previamente instalado; para cargar un paquete se puede emplear el siguiente código:

```
>library(nombre del paquete)
```

Normalmente el directorio de trabajo queda programado en la misma carpeta que se seleccionó en la configuración de R. Si emplea R Studio, cada vez que abra el programa haciendo click en el archivo de R, el programa empleará la ubicación del archivo como directorio de trabajo. Sin embargo, si no es así, puede configurarlo en el botón superior de "`Session`". O emplear la siguiente línea de código:

```
>setwd("C:/Users/nombre de usuario/carpeta del directorio de trabajo")
```

En el caso de estar usando Mac OS o Linux, dichas direcciones pueden variar y típicamente se emplea el caracter "\" en vez de "/".

Ahora puede procederse a cargar el archivo de polígonos como un objeto de R para trazar el mapa. El siguente código crea un objeto llamado *world.map* que contiene los polígonos que se encuentran en los tres archivos que están en el directorio de trabajo.

```
>world.map <- readOGR(dsn="C:/Users/nombre de usuario/carpeta",layer="TM_WORLD_BOR
DERS-0.3")
```

El archivo de los polígonos es un poco grande y puede tardar en cargarse unos segundos. Una vez esté completo el proceso, deberá aparecerle el nuevo objeto en el workspace.
Ahora, se debe proceder a transformar la información de los polígonos a un dataframe. Para hacerlo, puede emplear el siguiente código:

```
>world.ggmap <- fortify(world.map, region = "NAME")
```

La función fortify transforma un objeto de R (en este caso world.map) en un data frame con el fin de poder usarlo para graficar la información. Ahora bien, el argumento *region* le permite establecer el criterio con el cual quiere organizar la información (en este caso serán los nombres de los diferentes países que se encuentran en la variable "NAME"del objeto *world.map*).
Para verificar cuál es el nombre de la columna que contiene las regiones puede usar el siguiente código:

```
> head(world.map@data)
```

Y obtendrá lo siguiente:

	FIPS	ISO2	ISO3	UN	NAME	AREA	POP2005	REGION	SUBREGION	LON	LAT
0	AC	AG	ATG	28	Ant y Barb	44	83039	19	29	-61.783	17.078
1	AG	DZ	DZA	12	Algeria	238174	32854159	2	15	2.632	28.163
2	AJ	AZ	AZE	31	Azerbaijan	8260	8352021	142	145	47.395	40.430
3	AL	AL	ALB	8	Albania	2740	3153731	150	39	20.068	41.143
4	AM	AM	ARM	51	Armenia	2820	3017661	142	145	44.563	40.534
5	AO	AO	AGO	24	Angola	124670	16095214	2	17	17.544	-12.296

En este caso, como se puede observar, existe una columna denominada NAME que contiene los nombres de los diferentes países, la cual será el criterio para organizar el archivo de polígonos en el dataframe.

Ahora es importante cargar la base de datos que se va a graficar en el mapa. Tenga en cuenta que lo más común al graficar información en un mapa es emplear variables categóricas o discretas. En caso de tener una variable continua (como por ejemplo la tasa de crecimiento poblacional), esta tendrá que transformarse en categorías para que sea una variable discreta comprendida en diferentes rangos. Para esta aplicación se usará la tasa de crecimiento poblacional, en las mismas categorías que se muestra en la figura 1.
El archivo grwth.csv se obtuvo de la base de datos del Banco Mundial y se organizó de tal manera que los nombres de los países coincidieran con el de los archivos de polígonos. Además, se hizo la debida transformación de la variable para que ésta fuera categórica; es decir, que estuviera comprendida en rangos.
El siguiente código carga en el objeto TCP

```
>tcp <-  read.csv("grwth.csv", sep = ";", na.strings="NA")
```

Donde "grwth.csv " es el archivo en formato .csv que corresponde a la base de datos para el gráfico.

2.1. Organizando los datos

Por ahora tenemos la información de la variable de interés en el objeto TCP para cada país y, por otro lado, en el objeto world.ggmap se encuentran los polígonos organizados en data frame. Ahora debemos combinar esta información para poderla graficar. Esto se puede hacer con el siguiente código:

```
>names(tcp) <- c("id",  "grwth")
```

Esto cambia el nombre de las variables de la base de datos denominada `tcp`. En este caso sólo se tienen dos variables: el nombre del país (que se renombra como *id*) y la tasa de crecimiento (*grwth*). Ahora tenemos que manipular los objetos `tcp` y `world.ggmap`, de tal manera que el nombre del país sea el mismo. Esto se puede realizar con las siguientes líneas de código:

```
>tcp$id <- tolower(tcp$id)
>world.ggmap$id <- tolower(world.ggmap$id)
```

Estos códigos hacen que los elemenos de la columna *id* estén en letras minúsculas para ambos objetos. Esto se hace con el fin de que los nombres de los países en la base de datos y en el archivo de polígonos coincidan exactamente. Si quiere verificar cuáles son los nombres que están registrados

en el archivo de polígonos, puede exportar un archivo .csv con el nombre de los países usando el siguiente código:

```
>write.csv(world.map@data$NAME,file="paises.csv")
```

El archivo debería quedar guardado en el directorio de trabajo.
Ahora bien, para unir los dos objetos de manera adecuada use el siguiente código:

```
>world.ggmape <- merge(world.ggmap, tcp, by = "id", all = TRUE)
```

Aquí se está indicando que los combine por el criterio *id*, que en este momento es el nombre de los países tanto en el data frame del mapa como en el objeto tcp. Por último, es necesario organizar los datos del objeto `world.ggmape` de menor a mayor según la columna "`order`". Lo cual se puede realizar con la siguiente línea de código:

```
>world.ggmape <- world.ggmape[order(world.ggmape$order), ]
```

Hecho esto, la base de datos ya está lista para graficarse en el mapa.

2.2. Construyendo el gráfico

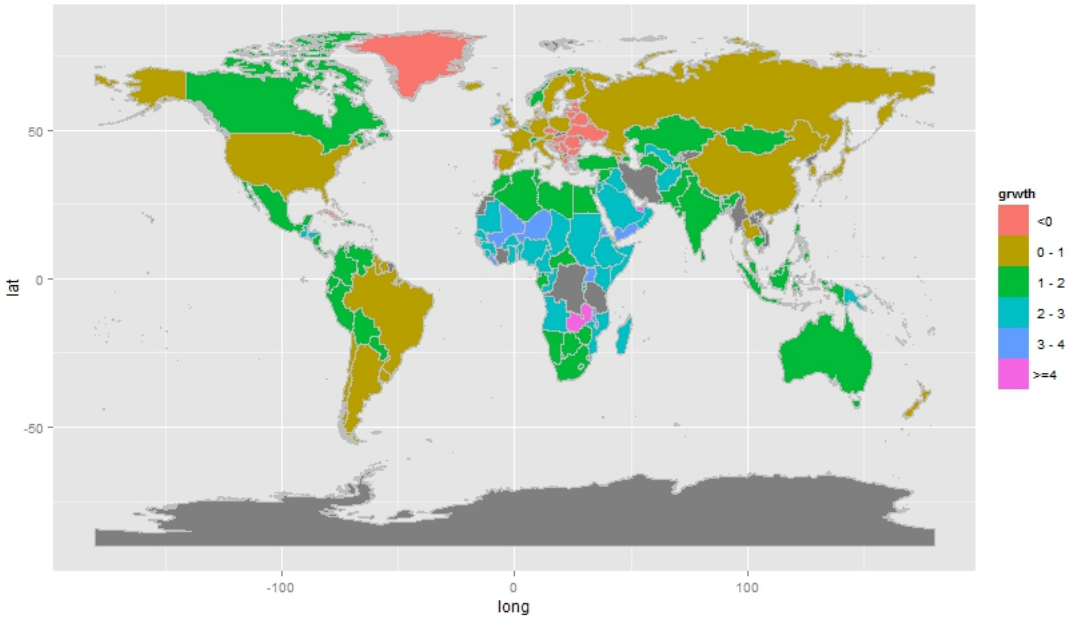
El código para armar el gráfico en ggplot2 es el siguiente:

```
>library(ggplot2)
>world.plot <- ggplot(data = world.ggmape, aes(x = long, y = lat, group = group))
+geom_polygon(aes(fill = grwth))
+geom_path(aes(x=long, y=lat, group=group), color="gray")
```

La primera línea de código se usa para cargar el paquete de ggplot2, el cual nos permite hacer gráficas en R.
La siguiente línea le dice a R que debe emplear la función ggplot del paquete con el mismo nombre para graficar el objeto previamente organizado con los datos a graficar y las coordenadas de cada uno de los países. Además, esta misma línea indica al programa que se debe emplear la longitud en el eje horizontal y la latitid en el eje vertical. Asimismo, `group` agrupa las zonas por país; es decir, da la orden a R para que identifique que dichas coordenadas pertenecen al mismo país.
La tercera línea hace que la zona demarcada para cada país se rellene con la variable deseada por medio de colores, en este caso la tasa de crecimiento poblacional.

El anterior código producirá la figura 2:

Figura 2: Crecimiento poblacional. 2011.



Fuente: Banco Mundial

Puede hacerle adiciones al anterior código para eliminar el título de los ejes y modificar los colores con los que se grafica. El código para obtener el mapa de la figura 1 es:

```
>world.plot <- ggplot(data = world.ggmape, aes(x = long, y = lat, group = group))
+geom_polygon(aes(fill = grwth))
+scale_fill_brewer(palette="Blues")
+geom_path(aes(x=long, y=lat, group=group), color="gray")
+labs(x="", y="",fill="Tasa de crecimiento poblacional (\%)")
+theme(axis.text.y=element_text(size=0, colour="snow"))
+theme(axis.text.x=element_text(size=0, colour="snow"))
```

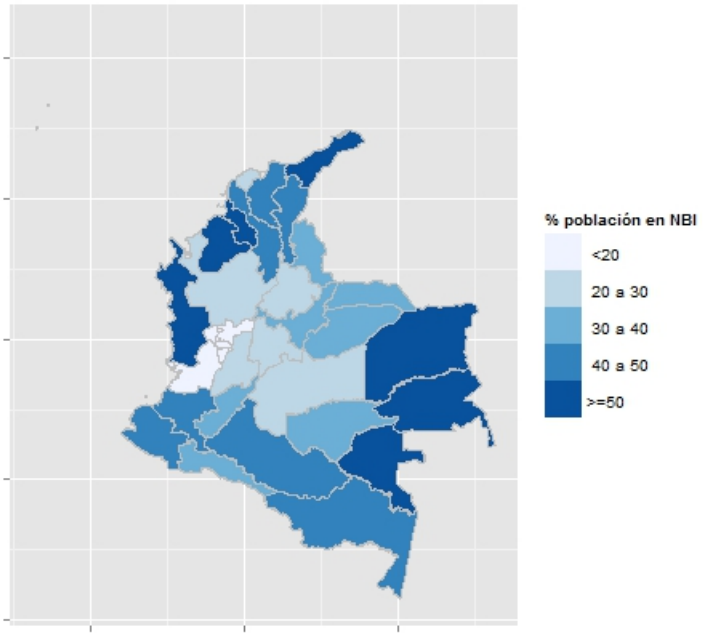
3. Colombia

La codificación en R para graficar los mapas de diferentes regiones es similar. Para el caso de Colombia, es necesario descargar el archivo de polígonos. Estos se pueden obtener en la página web `http://gadm.org/country`. Aquí encontrará una lista de todos los países, seleccione Colombia y descargue el *shapefile* que contiene los archivos de polígonos del mapa.

Una vez tenga el archivo de polígonos y la base de datos, puede proceder a organizar los archivos de R. Recuerde que los nombres de los departamentos deben coincidir a los del archivo de polígonos. En este caso, el archivo .zip que se descarga contiene archivos con el nombre `COLadmX`, donde X es un número, el cuál indica el tipo del polígono del mapa, dado que en unos se incluyen ríos y en otros ciudades. El que se usará es el 1; es decir, los archivos `COLadm1` con sus diferentes extensiones mencionadas anteriormente. Este mapa contiene las divisiones correspondientes a los departamentos.

La figura 3 muestra el porcentaje de la población que tiene necesidades básicas insatisfechas, empleando el archivo nbi.csv.

Figura 3: Porcentaje de personas con NBI. Colombia. Diciembre de 2011.



Fuente: DANE

El código que genera el mapa de la figura 3 es el siguiente:

```
>col <- readOGR(dsn="C:/Users/nombre de usuario/carpeta",layer="COL_adm1")
>col.ggmap <- fortify(col, region= "NAME_1")
>mapa <-  read.csv("nbi.csv", sep = ";", header=TRUE)
>names(mapa) <- c("id",  "nbi_jun_2012")
>mapa$id <- tolower(mapa$id)
>col.ggmap$id <- tolower(col.ggmap$id)
>col.ggmape <- merge(col.ggmap, mapa, by = "id", all = TRUE)
>col.ggmape <- col.ggmape[order(col.ggmape$order), ]
>col.plot <- ggplot(data = col.ggmape, aes(x = long, y = lat, group = group))
+geom_polygon(aes(fill = nbi_jun_2012))
+scale_fill_brewer(palette="Blues")
+geom_path(aes(x=long, y=lat, group=group), color="gray")
+labs(x="", y="",fill="% población en NBI")
+theme(axis.text.y=element_text(size=0, colour="snow"))
+theme(axis.text.x=element_text(size=0, colour="snow"))
```

Hasta aquí hemos mostrado cómo graficar información georeferenciada. Para tal fin es importante contar con la información de los polígonos y una base de datos (la cual debería tener una variable discreta a graficar). El procedimiento para realizar el mapa es relativamente sencillo y no cambia mucho a la hora de graficar diferentes regiones.

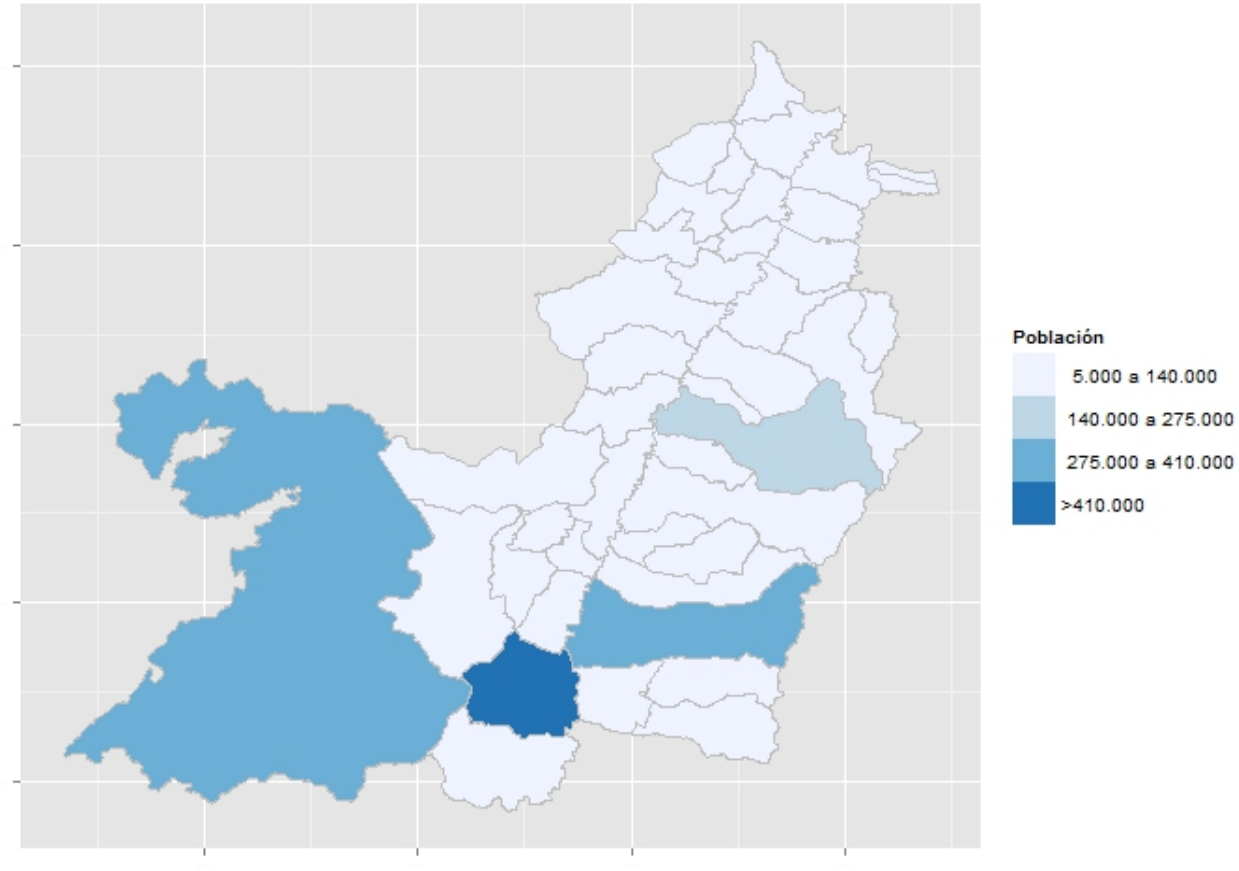
A continuación, se plantean dos ejemplos de mapas que se dejan como ejercicio para que el lector los pueda reproducir. Al final se muestra el código sugerido para replicarlos.

4. Ejercicios

4.1. Valle del Cauca

Usando los archivos `poligonosValle.zip` y `pob2012.csv`, grafique el número de habitantes de los diferentes municipios de Valle del Cauca; de tal forma que pueda reproducir la figura 4.

Figura 4: Población por municipios. Valle del Cauca. 2012.

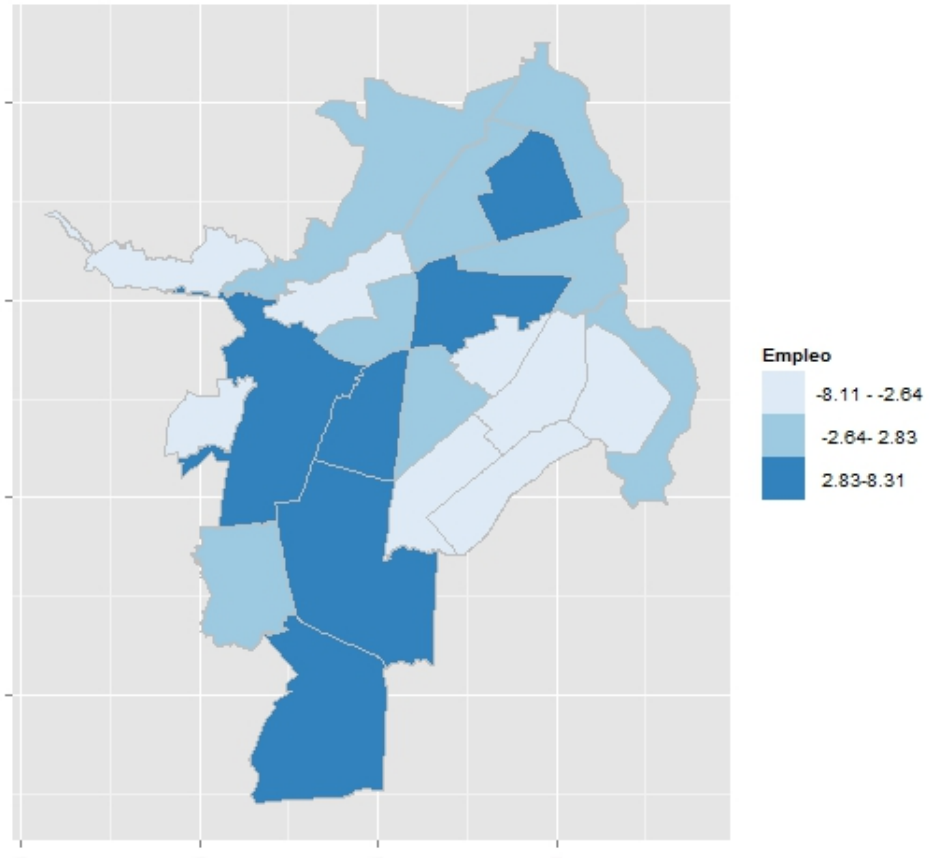


Fuente: DANE

4.2. Cali

Replique la figura 5 usando como variable georeferenciada la estimación de la calidad del empleo para las diferentes comunas de la ciudad. Emplee los archivos `cali.zip` y `empleo.csv` para hacerlo.

Figura 5: Estimación de la calidad del empleo. Cali. 2012.



Fuente: Indicadores de calidad de empleo, Pérez (2012)

5. Respuestas

5.1. El código para obtener el mapa del Valle del Cauca (figura 4) fue:

```
>valle <- readOGR(dsn="C:/Users/nombre de usuario/carpeta",layer="Valle_del__Cauca")
>valle.ggmap <- fortify(valle, region= "nom_munici")
>mapa <-  read.csv("pob_2012.csv", sep = ";", header=TRUE)
>names(mapa) <- c("id",  "pob", "categ")
>mapa$id <- tolower(mapa$id)
>valle.ggmap$id <- tolower(valle.ggmap$id)
```

```
>valle.ggmape <- merge(valle.ggmap, mapa, by = "id", all = TRUE)
>valle.ggmape <- valle.ggmape[order(valle.ggmape$order), ]
>valle.plot <- ggplot(data = valle.ggmape, aes(x = long, y = lat, group = group))
+geom_polygon(aes(fill = categ))
+scale_fill_brewer(palette="Blues")
+geom_path(aes(x=long, y=lat, group=group), color="gray")
+labs(x="", y="",fill="Población")
+theme(axis.text.y=element_text(size=0, colour="snow"))
+theme(axis.text.x=element_text(size=0, colour="snow"))
```

5.2. El código en R usado para obtener el mapa del Cali (figura 5) fue:

```
>cali <- readOGR(dsn="C:/Users/nombre de usuario/carpeta",layer="COMUNAS5")
>cali.ggmap <- fortify(cali, region= "COMUNAS5_I")
>mapa <-  read.csv("empleo.csv", sep = ";", header=TRUE)
>names(mapa) <- c("id",  "emp")
>cali.ggmape <- merge(cali.ggmap, mapa, by = "id", all = TRUE)
>cali.ggmape <- cali.ggmape[order(cali.ggmape$order), ]
>cali.plot <- ggplot(data = cali.ggmape, aes(x = long, y = lat, group = group))
+geom_polygon(aes(fill = emp))
+scale_fill_brewer(palette="Blues")
+geom_path(aes(x=long, y=lat, group=group), color="gray")
+labs(x="", y="",fill="Empleo")
+theme(axis.text.y=element_text(size=0, colour="snow"))
+theme(axis.text.x=element_text(size=0, colour="snow"))
```

Referencias

J. C. Alonso y A. González. *Apuntes de Economía*. Number 33. Universidad Icesi, 2012.

M. Zachry y C. Thralls. An interview with edward r. tufte. *Technical Communication Quarterly*, 13 (4):447–462, 2004.