

# Escuela ALAP: Familia, parentesco y hogares en América Latina y el Caribe

Juan Galeano | jgaleano@ced.uab.es

2024-12-04

## Contents

<b>Introducción</b>	<b>3</b>
<b>Ejercicio 1: Tamaño medio de hogares por país</b>	<b>3</b>
Librerías y Configuración . . . . .	3
Leer la base de datos CORESIDENCE . . . . .	4
Filtrar casos de Latinoamérica . . . . .	4
Recuperar Datos Espaciales de Gisco . . . . .	4
¿Qué es giscoR? . . . . .	4
Características Clave de giscoR . . . . .	4
Aplicaciones Típicas . . . . .	5
Recursos y Documentación . . . . .	5
Unir datos espaciales y estadísticos . . . . .	5
Calcular deciles . . . . .	5
Categorizar datos por deciles . . . . .	5
Crear una paleta de colores . . . . .	7
Visualice los datos en un mapa . . . . .	7
Guardar el mapa como un archivo de imagen . . . . .	8
<b>Ejercicio 2: Tamaño medio de hogares por región, Latinoamérica</b>	<b>8</b>
Cargar Datos Espaciales . . . . .	9
Extraer el conjunto de datos armonizados de la base de datos CORESIDENCE . . . . .	9
Categorizar datos por deciles . . . . .	9
Unir datos estadísticos y espaciales . . . . .	10
Visualice los datos en un mapa . . . . .	10
Guarde el mapa como un archivo de imagen . . . . .	11

<b>Ejercicio 3: Diagrama de Caja para el tamaño medio de los hogares por regiones en cada país de latinoamérica</b>	<b>12</b>
Crear un tema para los gráficos . . . . .	12
Ordenar los países respecto a la mediana y calcular el tamaño medio de los hogares en cada país . . . . .	13
Calcular deciles sobre la media . . . . .	13
Adjuntar colores al factor . . . . .	14
Visualice los datos en un diagrama de caja . . . . .	14
Guarde el diagrama de caja como un archivo de imagen . . . . .	15
<b>Ejercicio 3.1: Organizar dos Gráficos en un <i>grid</i></b>	<b>15</b>
Crear un <i>grid</i> de objetos <i>ggplot</i> . . . . .	16
Organice una cuadrícula con los dos objetos . . . . .	17
Guarde la imagen de la cuadrícula ordenada como un archivo de imagen . . . . .	18
<b>Ejercicio 4: Hogares unipersonales en México</b>	<b>18</b>
Crear un objeto espacial de México por regiones . . . . .	19
Crear un marco de datos con datos mexicanos . . . . .	19
Calcular <i>pretty_breaks</i> . . . . .	19
Unir datos espaciales y estadísticos . . . . .	20
Crea una paleta de colores usando RColorBrewer . . . . .	20
Trazar un mapa coroplético facetado . . . . .	20
Guardar el mapa facetado como un archivo de imagen . . . . .	21
<b>Ejercicio 4.1: Hogares unipersonales en México, geofacet</b>	<b>21</b>
¿Qué es <i>geofacet</i> ? . . . . .	21
Trazar un gráfico de líneas por estados con <i>geogacet</i> . . . . .	22
Guardar el mapa facetado como un archivo de imagen . . . . .	23
<b>Ejercicio 5: Configuraciones de convivencia los hogares, Uruguay 1963-2011</b>	<b>23</b>
Cargar y preparar los datos . . . . .	23
Extrar los datos disponibles de Uruguay . . . . .	24
Dividir el marco de datos en una lista por muestras . . . . .	24
Calcular la distribución relativa de los arreglos de vida por sexo y edades individuales para cada uno de los elementos de la lista . . . . .	24
Combine los elementos de lista en un único marco de datos . . . . .	24
Leer archivo excel con la clasificación de los arreglos de vida . . . . .	24
Crear un marco de datos con años, edades individuales y sexo . . . . .	24
Recodificar y renivelar variables . . . . .	25
Crea una paleta de colores personalizada . . . . .	26

Crear una gráfico de area por edades simples y sexo . . . . .	27
Guardar el mapa facetado como un archivo de imagen . . . . .	27

## Introducción

El documento titulado “Escuela ALAP: Familia, parentesco y hogares en América Latina y el Caribe” presenta una serie de análisis diseñados para explorar aspectos demográficos de los hogares en América Latina, centrándose en su estructura, tamaño y configuraciones de convivencia. A través de ejercicios prácticos, se combinan datos estadísticos y espaciales para generar visualizaciones descriptivas y análisis detallados. Se incluyen ejercicios como la creación de mapas de coropletas para mostrar el tamaño promedio de los hogares en América Latina, diagramas de caja para comparar tamaños medianos por país, análisis de hogares unipersonales en México con visualizaciones temporales y espaciales, y una exploración de las configuraciones de convivencia en Uruguay entre 1963 y 2011 mediante gráficos de áreas que destacan diferencias de género y cambios a lo largo del ciclo de vida. El análisis utiliza técnicas avanzadas de manipulación y visualización de datos con herramientas como dplyr, ggplot2 y sf, junto con personalización de colores y diseño de gráficos para mejorar la interpretación visual. El documento está diseñado como un recurso educativo y analítico para investigadores y estudiantes interesados en demografía y análisis de la composición de los hogares.

---

## Ejercicio 1: Tamaño medio de hogares por país

Este script de R genera un mapa que visualiza el tamaño medio de los hogares en países de América Latina partiendo la distribución en deciles. Comienza cargando la base de datos CORESIDENCE y extrayendo datos a nivel nacional (**NATIONAL\_DB**). El conjunto de datos se filtra para incluir registros que representen países de América Latina, los últimos datos disponibles en CodB (**T3 == 1**) y asegurando que todos los registros son posteriores al año 1999 (**T1 >= 2000**). Las variables seleccionadas incluyen indicadores demográficos y de tamaño medio de hogar.

Los datos espaciales de los países del mundo se obtienen utilizando el paquete **giscoR**, asegurando compatibilidad con los datos estadísticos al alinear los nombres de las columnas (**C1**). Los datos filtrados de América Latina se combinan con el conjunto de datos espaciales mediante **left\_join**, manteniendo solo las filas correspondientes a países de América Latina. Los indicadores de tamaño de hogar se clasifican en deciles utilizando la función **quantile**, creando categorías para la visualización. Las etiquetas que representan los rangos de los deciles se generan dinámicamente y se reordenan para mayor claridad.

Se define una paleta de colores “Spectral” invertida utilizando **RColorBrewer** para una mejor diferenciación visual. Con **ggplot2**, el script traza un mapa de coropletas donde **geom\_sf** representa los datos espaciales y los colores de relleno indican las categorías de los deciles. Características adicionales, como una leyenda, anotaciones y un tema minimalista (**theme\_void**), mejoran la legibilidad del mapa. El mapa final se guarda como un archivo PNG de alta resolución. Este flujo de trabajo integra datos estadísticos y espaciales para proporcionar una representación visual clara de las distribuciones del tamaño medio de los hogares en América Latina.

---

## Librerías y Configuración

Cargue las librerías necesarias para la manipulación de datos, análisis espacial y visualización:

```

library(tidyverse)      # Data manipulation and visualization
library(sf)             # Spatial data manipulation
library(giscoR)         # Retrieve GISCO data
library(RColorBrewer)   # Color palettes for the map
library(readxl)          # Read excel files into R

```

## Leer la base de datos CORESIDENCE

```

load("CORESIDENCE_DATABASE_2024.RData")

# Extraer el conjunto de datos nacionales de la lista
NATIONAL_DB <- CORESIDENCE_DB[["NATIONAL"]]

```

## Filtrar casos de Latinoamérica

Filtrar el conjunto de datos para incluir sólo países de América Latina y seleccionar variables relevantes (HS17: Tamaño promedio de los hogares).

```

LA <- NATIONAL_DB |>
  filter(T3 == 1, C3 == "LATIN-AMERICA") |>
  select(1:12, "HS17") |>
  filter(T1 >= 2000)

```

## Recuperar Datos Espaciales de Gisco

### ¿Qué es giscoR?

**giscoR** es un paquete de R diseñado para trabajar con datos geoespaciales, específicamente para descargar y procesar datos geográficos de la base de datos GISCO (Geographical Information System of the Commission) de Eurostat. GISCO proporciona conjuntos de datos espaciales para Europa y otras regiones, utilizados en análisis y visualizaciones en campos como la demografía, la economía y los estudios urbanos.

### Características Clave de giscoR

- Acceso a Datos GISCO:** Permite acceder directamente a shapefiles (datos espaciales) de GISCO en R sin necesidad de descargar o procesar manualmente los datos.
- Herramientas de Preprocesamiento:** Simplifica el proceso de transformar y manipular datos geográficos.
- Mapas de Alta Calidad:** Facilita la creación de visualizaciones geográficas de alta calidad al integrar datos GISCO con potentes paquetes de R como **ggplot2** o **tmap**.
- Personalizable:** Ofrece la posibilidad de especificar niveles de detalle geográfico, como países, regiones o municipios, y definir el año de interés.
- Compatibilidad CRS:** Funciona con sistemas de referencia de coordenadas (CRS) y permite re-proyectar datos espaciales según sea necesario.

## Aplicaciones Típicas

- **Estudios Demográficos:** Mapear la distribución de población, flujos migratorios o características de los hogares.
- **Análisis Económicos:** Visualizar indicadores económicos a través de regiones o países.
- **Investigación Ambiental:** Mapear recursos naturales, áreas protegidas o datos climáticos.
- **Planificación Urbana:** Analizar el crecimiento urbano, redes de transporte o infraestructura.

## Recursos y Documentación

- **Documentación en CRAN:** giscoR en CRAN
- **Repositorio en GitHub:** giscoR en GitHub
- **Datos de GISCO de Eurostat:** Base de Datos GISCO

```
# Retrieve spatial data for countries of the world
world <- gisco_get_countries(resolution = "03", epsg = 4326)

# Rename column to match with statistical data
colnames(world)[3] <- "C1"
```

## Unir datos espaciales y estadísticos

```
world <- world |>
  left_join(LA, by = "C1") |>
  filter(C3 == "LATIN-AMERICA")

# Cambie el nombre de la columna correspondiente para mayor claridad
colnames(world)[16] <- "indicator"
```

## Calcular deciles

Calcule los deciles de la variable *indicator* para clasificar los datos en categorías.

```
deciles <- quantile(world$indicator, probs = seq(0, 1, by = 0.1), na.rm = TRUE)
```

## Categorizar datos por deciles

Asignar etiquetas a los datos en función de los rangos de deciles.

```
world <- world |>
  mutate(indicator2 = cut(indicator,
                          breaks = quantile(indicator, probs = seq(0, 1, 0.1), na.rm = TRUE),
                          include.lowest = TRUE,
                          labels = FALSE),
         labels = case_when(
           indicator2 == 1 ~ paste("<=", round(deciles[2], 2), sep = ""),
           indicator2 == 2 ~ paste("(", round(deciles[2], 2), "-", round(deciles[3], 2), "]", sep = "")),
```

```

indicator2 == 3 ~ paste("(", round(deciles[3], 2), "-",
                      round(deciles[4], 2), "]", sep = ""),
indicator2 == 4 ~ paste("(", round(deciles[4], 2), "-",
                      round(deciles[5], 2), "]", sep = ""),
indicator2 == 5 ~ paste("(", round(deciles[5], 2), "-",
                      round(deciles[6], 2), "]", sep = ""),
indicator2 == 6 ~ paste("(", round(deciles[6], 2), "-",
                      round(deciles[7], 2), "]", sep = ""),
indicator2 == 7 ~ paste("(", round(deciles[7], 2), "-",
                      round(deciles[8], 2), "]", sep = ""),
indicator2 == 8 ~ paste("(", round(deciles[8], 2), "-",
                      round(deciles[9], 2), "]", sep = ""),
indicator2 == 9 ~ paste("(", round(deciles[9], 2), "-",
                      round(deciles[10], 2), "]", sep = ""),
indicator2 == 10 ~ paste(">=", round(deciles[10], 2), sep = "")),
labels = fct_relevel(labels,
                      paste("<=", round(deciles[2], 2), sep = ""),
                      paste("(", round(deciles[2], 2), "-",
                            round(deciles[3], 2), "]", sep = ""),
                      paste("(", round(deciles[3], 2), "-",
                            round(deciles[4], 2), "]", sep = ""),
                      paste("(", round(deciles[4], 2), "-",
                            round(deciles[5], 2), "]", sep = ""),
                      paste("(", round(deciles[5], 2), "-",
                            round(deciles[6], 2), "]", sep = ""),
                      paste("(", round(deciles[6], 2), "-",
                            round(deciles[7], 2), "]", sep = ""),
                      paste("(", round(deciles[7], 2), "-",
                            round(deciles[8], 2), "]", sep = ""),
                      paste("(", round(deciles[8], 2), "-",
                            round(deciles[9], 2), "]", sep = ""),
                      paste("(", round(deciles[9], 2), "-",
                            round(deciles[10], 2), "]", sep = ""),
                      paste(">=", round(deciles[10], 2), sep = ""))

```

**mutate:** Añade o modifica columnas en un data frame. Aquí se crea una nueva columna llamada indicator2.

**cut:** Divide el rango de indicator en intervalos basados en los deciles.

**quantile(indicator, probs = seq(0, 1, 0.1), na.rm = TRUE):** Calcula los valores de los deciles del vector indicator, dividiendo su rango en 10 partes iguales.

**seq(0, 1, 0.1)** genera los valores de probabilidad para 0%, 10%, ..., 100%.

**na.rm = TRUE** ignora valores faltantes (NA).

**include.lowest = TRUE:** Incluye el valor más bajo en el primer intervalo.

**labels = FALSE:** Los intervalos no se etiquetan directamente; en su lugar, se asignan valores numéricos (1, 2, ..., 10) según el decil al que pertenece cada observación.

**case\_when:** Crea valores condicionales basados en indicator2. Para cada nivel de indicator2 (1, 2, ..., 10), se asigna una etiqueta que representa el rango correspondiente del indicador. Las etiquetas se construyen con paste, concatenando texto como “(”, valores redondeados de los deciles (round(deciles[n], 2)), y delimitadores (“-” o “]”).

Ejemplo:

Si indicator2 == 1, la etiqueta es “<=” seguida del valor del segundo decil. Si indicator2 == 2, la etiqueta indica un intervalo entre los deciles 2 y 3.

## Crear una paleta de colores

Defina una paleta de colores para la visualización del mapa.

```
myColors <- rev(brewer.pal(10, "Spectral"))

# Visualice su paleta de colores y obtenga los nombres hexadecimales de ellos.
barplot(rep(length(myColors),length(myColors)),col=c(myColors));myColors
```

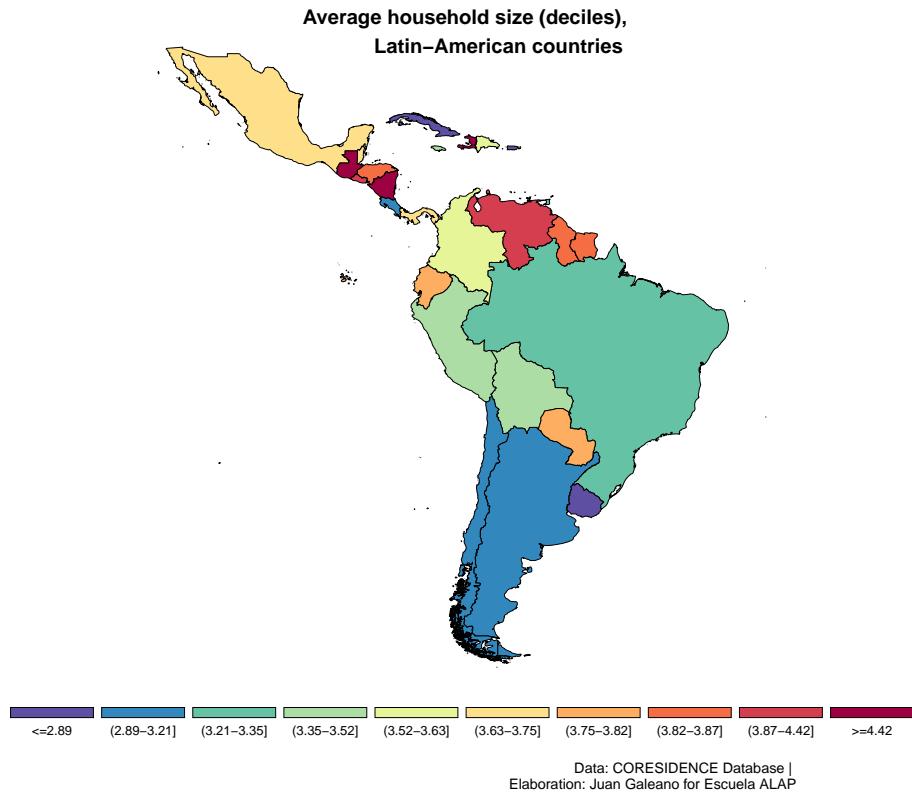


```
## [1] "#5E4FA2" "#3288BD" "#66C2A5" "#ABDDA4" "#E6F598" "#FEE08B" "#FDAE61"
## [8] "#F46D43" "#D53E4F" "#9E0142"
```

## Visualice los datos en un mapa

```
worldmap <- ggplot() +
  geom_sf(data = world, aes(fill = labels), colour = "black", linewidth = 0.075) +
  scale_fill_manual(values = c(myColors),
    guide = guide_legend(direction = "horizontal",
      nrow = 1,
      keyheight = 0.5,
      keywidth = 4,
      label.position = "bottom")) +
  theme_void() +
  annotate("text", x = -73, y = 35, label = "Average household size (deciles),
    Latin-American countries", fontface = "bold", size = 5) +
  labs(caption = "\nData: CORESIDENCE Database |
    Elaboration: Juan Galeano for Escuela ALAP") +
  theme(legend.position = "bottom",
    legend.title = element_blank(),
    plot.caption = element_text(vjust = 0.5, size = 10, colour = "black"),
    plot.margin = unit(c(0.5, 0, 0.5, 0), "cm"),
    panel.background = element_rect(fill = "white", color = "white"),
    plot.background = element_rect(fill = "white", color = "white"),
    panel.grid.major = element_line(color = "white", linewidth = 0.15, linetype = 2),
    panel.grid.minor = element_line(color = "white", linewidth = 0.1, linetype = 1))

worldmap
```



## Guardar el mapa como un archivo de imagen

```
ggsave("map_1.png",
       scale = 1,
       dpi = 300,
       height =7.5,
       width = 17)
```

---

## Ejercicio 2: Tamaño medio de hogares por región, Latinoamérica

Este script de R genera un mapa que representa el tamaño medio de los hogares para regiones subnacionales en América Latina partiendo la distribución en deciles. El análisis comienza cargando un shapefile armonizado que contiene datos espaciales (WORLDCO) y renombrando columnas para garantizar compatibilidad. Los datos armonizados a nivel subnacional (SUBNATHA\_DB) se extraen de la base de datos CORESIDIENCE y se filtran para incluir únicamente países de América Latina (C4 == "LATIN-AMERICA"), los datos más recientes disponibles (T3 == 1) y registros a partir del año 2000. Una variable indicadora relevante (HS17) se selecciona y renombra para el análisis.

Los deciles de esta variable indicadora se calculan utilizando `quantile`, y las etiquetas correspondientes a cada rango de decil se generan dinámicamente con `mutate`. Los datos estadísticos filtrados y categorizados se combinan con los datos espaciales (WORLDCO) mediante `left_join`, alineándose con base en una columna común (C5). El conjunto de datos fusionado se filtra para retener únicamente las regiones de América Latina.

Un mapa de coropletas se crea utilizando ggplot2, donde `geom_sf` traza las regiones y una paleta de colores “Spectral” invertida representa los deciles. Se agregan anotaciones adicionales, como etiquetas y leyendas, para mejorar la interpretación. El script aplica un tema minimalista (`theme_void`) que personaliza la posición de la leyenda y los estilos de la cuadrícula. El mapa final se guarda como un archivo PNG de alta resolución (`map_2.png`) para uso posterior. Este flujo de trabajo combina datos espaciales y estadísticos para visualizar las distribuciones del tamaño de los hogares de manera efectiva a nivel regional.

---

## Cargar Datos Espaciales

Lea el shapefile armonizado de la base de datos CORESIDENCE:

```
WORLDSCO <- st_read(".", "A1_WORLD_SUBNATIONAL_CORESIDENCE_SIMP")

## Reading layer 'A1_WORLD_SUBNATIONAL_CORESIDENCE_SIMP' from data source
##   'G:\Shared drives\CORESIDENCE\WP4_RESEARCH OUTPUT\4_5_Workshops\ALAP_2024\JUAN'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 1580 features and 7 fields (with 14 geometries empty)
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:  xmin: -180 ymin: -56.52419 xmax: 180 ymax: 83.2104
## Geodetic CRS:  WGS 84

colnames(WORLDSCO)[4] <- "C5"
```

## Extraer el conjunto de datos armonizados de la base de datos CORESIDENCE

Filtrar el conjunto de datos para incluir solo los países de América Latina y seleccionar las variables relevantes.

```
SUBNATHA_DB <- CORESIDENCE_DB[["SUBNATIONAL_HARMONIZED"]]

data <- SUBNATHA_DB |>
  filter(T3 == 1, C4 == "LATIN-AMERICA") |>
  select(1:14, 32) |>
  filter(T1 >= 2000)

colnames(data)[15] <- "indicator"
```

## Categorizar datos por deciles

Asignar etiquetas a los datos en función de los rangos de deciles.

```
deciles <- quantile(data$indicator, probs = seq(0, 1, by = .1), na.rm = TRUE)

data <- data |> mutate(
  indicator2 = cut(indicator,
    breaks = quantile(indicator, probs = seq(0, 1, 0.1), na.rm = TRUE),
    include.lowest = TRUE,
    labels = FALSE),
```

```

labels = case_when(
  indicator2 == 1 ~ paste("<=", round(deciles[2], 2), sep = ""),
  indicator2 == 2 ~ paste("(", round(deciles[2], 2), "-", round(deciles[3], 2), "]", sep = ""),
  indicator2 == 3 ~ paste("(", round(deciles[3], 2), "-", round(deciles[4], 2), "]", sep = ""),
  indicator2 == 4 ~ paste("(", round(deciles[4], 2), "-", round(deciles[5], 2), "]", sep = ""),
  indicator2 == 5 ~ paste("(", round(deciles[5], 2), "-", round(deciles[6], 2), "]", sep = ""),
  indicator2 == 6 ~ paste("(", round(deciles[6], 2), "-", round(deciles[7], 2), "]", sep = ""),
  indicator2 == 7 ~ paste("(", round(deciles[7], 2), "-", round(deciles[8], 2), "]", sep = ""),
  indicator2 == 8 ~ paste("(", round(deciles[8], 2), "-", round(deciles[9], 2), "]", sep = ""),
  indicator2 == 9 ~ paste("(", round(deciles[9], 2), "-", round(deciles[10], 2), "]", sep = ""),
  indicator2 == 10 ~ paste(">=", round(deciles[10], 2), sep = "")
),
labels = fct_relevel(labels,
  paste("<=", round(deciles[2], 2), sep = ""),
  paste("(", round(deciles[2], 2), "-", round(deciles[3], 2), "]", sep = ""),
  paste("(", round(deciles[3], 2), "-", round(deciles[4], 2), "]", sep = ""),
  paste("(", round(deciles[4], 2), "-", round(deciles[5], 2), "]", sep = ""),
  paste("(", round(deciles[5], 2), "-", round(deciles[6], 2), "]", sep = ""),
  paste("(", round(deciles[6], 2), "-", round(deciles[7], 2), "]", sep = ""),
  paste("(", round(deciles[7], 2), "-", round(deciles[8], 2), "]", sep = ""),
  paste("(", round(deciles[8], 2), "-", round(deciles[9], 2), "]", sep = ""),
  paste("(", round(deciles[9], 2), "-", round(deciles[10], 2), "]", sep = ""),
  paste(">=", round(deciles[10], 2), sep = ""))
)
)

```

## Unir datos estadísticos y espaciales

Fusionar datos espaciales con datos estadísticos para América Latina.

```

WORLD01 <- left_join(WORLD00, data, by = "C5") |>
  filter(C4 == "LATIN-AMERICA")

```

## Visualice los datos en un mapa

```

worldmap <- ggplot() +
  geom_sf(data = WORLD01, aes(fill = labels), colour = "black", linewidth = .075) +
  scale_fill_manual(values = rev(brewer.pal(10, "Spectral")),
    guide = guide_legend(direction = "horizontal",
      nrow = 1,
      keyheight = .5,
      keywidth = 4,
      label.position = "bottom")) +
  theme_void() +
  annotate("text", x = -73, y = 35, label = "Average household size (deciles),
    Latin-American regions", fontface = "bold", size = 5) +
  labs(caption = "\nData: CORESIDENTE Database |
    Elaboration: Juan Galeano for Escuela ALAP") +
  theme(legend.position = "bottom",
    legend.title = element_blank(),

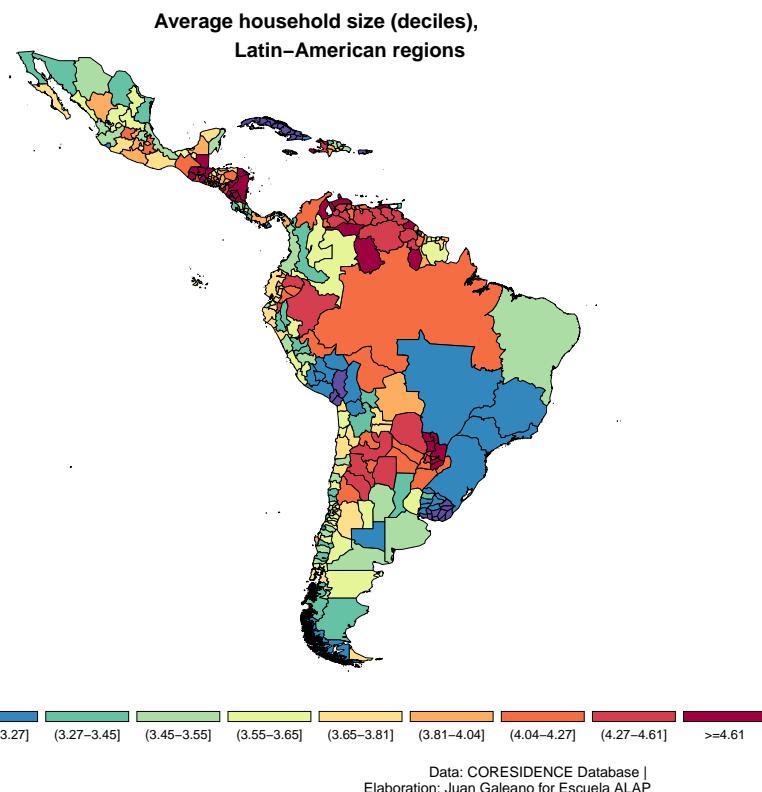
```

```

plot.caption = element_text(vjust = 0.5, size = 10, colour = "black"),
plot.margin = unit(c(0.5, 0, 0.5, 0), "cm"),
panel.background = element_rect(fill = "white", color = "white"),
plot.background = element_rect(fill = "white", color = "white"),
panel.grid.major = element_line(color = "white", linewidth = 0.15, linetype = 2),
panel.grid.minor = element_line(color = "white", linewidth = 0.1, linetype = 1))

```

worldmap



Guarde el mapa como un archivo de imagen

```

# Save the map
ggsave("map_2.png",
       scale = 1,
       dpi = 300,
       height = 7.5,
       width = 17)

```

## Ejercicio 3: Diagrama de Caja para el tamaño medio de los hogares por regiones en cada país de latinoamérica

Este script de R genera un diagrama de caja que visualiza el tamaño promedio de los hogares por países, ordenados según su tamaño mediano de hogar. Primero, se define un tema personalizado para los gráficos (`theme_alap`) que estandariza la apariencia de la visualización, especificando tamaños de fuente, formato de ejes, estilos de leyendas y elementos de fondo para un diseño limpio y consistente.

Los datos se procesan reordenando los países (`C2`) en función de su tamaño mediano de hogar utilizando `fct_reorder`. Luego, se calcula el tamaño medio de hogar para cada país y se asigna a una nueva columna. Posteriormente, los deciles del tamaño medio de hogar se calculan con `quantile`, y los datos se categorizan en estos rangos de deciles. Las etiquetas de cada decil se generan dinámicamente y se ordenan con `case_when` y `fct_relevel`, asegurando que la leyenda refleje una progresión clara de los valores del tamaño de hogar.

Se asignan colores a las categorías de los deciles utilizando una paleta “Spectral” invertida, y `myColors` mapea estos colores a los niveles correspondientes de los factores de las etiquetas de los deciles. El diagrama de caja se crea con `ggplot2`, representando a los países en el eje y y el tamaño de los hogares en el eje x. Las categorías de los deciles se visualizan mediante colores de relleno, y los valores atípicos se marcan en rojo para enfatizarlos. El gráfico se mejora con el tema `theme_alap` previamente definido y coordenadas invertidas (`coord_flip`) para facilitar la lectura de las etiquetas de los países. La leyenda se personaliza para mostrar las categorías de los deciles horizontalmente, pero se oculta del gráfico principal para simplificarlo.

Finalmente, el diagrama de caja se guarda como un archivo PNG de alta resolución (`fig_1.png`) para su uso posterior. Esta visualización destaca la distribución de tamaños de hogares en los países, permitiendo identificar fácilmente los valores medianos y la variación dentro de cada país.

---

### Crear un tema para los gráficos

```
theme_alap<-list(theme(plot.title = element_text(lineheight=1, size=15, face="bold"),
                        plot.subtitle = element_text(lineheight=1, size=12, face="bold"),
                        plot.caption = element_text(lineheight=1, size=13, hjust=1),
                        legend.title = element_blank(),
                        legend.text = element_text(colour="black", size = 15),
                        legend.position="bottom",
                        legend.background = element_rect(fill=NA, colour = NA),
                        legend.key.size = unit(1.5, 'lines'),
                        legend.key = element_rect(colour = NA, fill = NA),
                        axis.title.x = element_blank(),
                        axis.text.x = element_text(angle = 0,vjust=0.5, size=15,colour="black"),
                        axis.title.y = element_text(vjust=0.5, size=15,colour="black"),
                        axis.text.y = element_text(vjust=0.5, size=15,colour="black"),
                        strip.text = element_text(size=15, face="bold"),
                        plot.background = element_rect(fill = "white"),
                        panel.grid.major=element_line(colour="#E6E6E6", linewidth=.5),
                        panel.grid.minor=element_line(colour="#E6E6E6", linewidth=.15),
                        panel.border = element_rect(colour = "#585858", fill=NA, linewidth=.75),
                        panel.background =element_rect(fill ="#FFFFFF", colour = "#FFFFFF")))
```

Ordenar los países respecto a la mediana y calcular el tamaño medio de los hogares en cada país

```
data <- data |>
  mutate(C2 = fct_reorder(C2, indicator, .fun = 'median')) |>
  mutate(mean = mean(indicator), .by = C2)
```

Calcular deciles sobre la media

```
deciles <- quantile(data$mean, probs = seq(0, 1, by = .1), na.rm = TRUE)

data<-data|>mutate(indicator3=cut(mean,
                                      breaks = quantile(mean, probs = seq(0, 1, 0.1), na.rm = TRUE),
                                      include.lowest = TRUE,
                                      labels = FALSE),
                      labels_mean= case_when(
                        indicator3 == 1 ~ paste("<=",round(deciles[2],2),sep=""),
                        indicator3 == 2 ~ paste("(,round(deciles[2],2),"-",
                                              round(deciles[3],2),"]",sep=""),
                        indicator3 == 3 ~ paste("(,round(deciles[3],2),"-",
                                              round(deciles[4],2),"]",sep=""),
                        indicator3 == 4 ~ paste("(,round(deciles[4],2),"-",
                                              round(deciles[5],2),"]",sep=""),
                        indicator3 == 5 ~ paste("(,round(deciles[5],2),"-",
                                              round(deciles[6],2),"]",sep=""),
                        indicator3 == 6 ~ paste("(,round(deciles[6],2),"-",
                                              round(deciles[7],2),"]",sep=""),
                        indicator3 == 7 ~ paste("(,round(deciles[7],2),"-",
                                              round(deciles[8],2),"]",sep=""),
                        indicator3 == 8 ~ paste("(,round(deciles[8],2),"-",
                                              round(deciles[9],2),"]",sep=""),
                        indicator3 == 9 ~ paste("(,round(deciles[9],2),"-",
                                              round(deciles[10],2),"]",sep=""),
                        indicator3 == 10 ~ paste(">=",round(deciles[10],2),sep="")),
                      labels_mean= fct_relevel(labels_mean,
                                              paste("<=",round(deciles[2],2),sep=""),
                                              paste("(,round(deciles[2],2),"-",
                                                    round(deciles[3],2),"]",sep=""),
                                              paste("(,round(deciles[3],2),"-",
                                                    round(deciles[4],2),"]",sep=""),
                                              paste("(,round(deciles[4],2),"-",
                                                    round(deciles[5],2),"]",sep=""),
                                              paste("(,round(deciles[5],2),"-",
                                                    round(deciles[6],2),"]",sep=""),
                                              paste("(,round(deciles[6],2),"-",
                                                    round(deciles[7],2),"]",sep=""),
                                              paste("(,round(deciles[7],2),"-",
                                                    round(deciles[8],2),"]",sep=""),
                                              paste("(,round(deciles[8],2),"-",
                                                    round(deciles[9],2),"]",sep="),
                                              paste("(,round(deciles[9],2),"-",
```

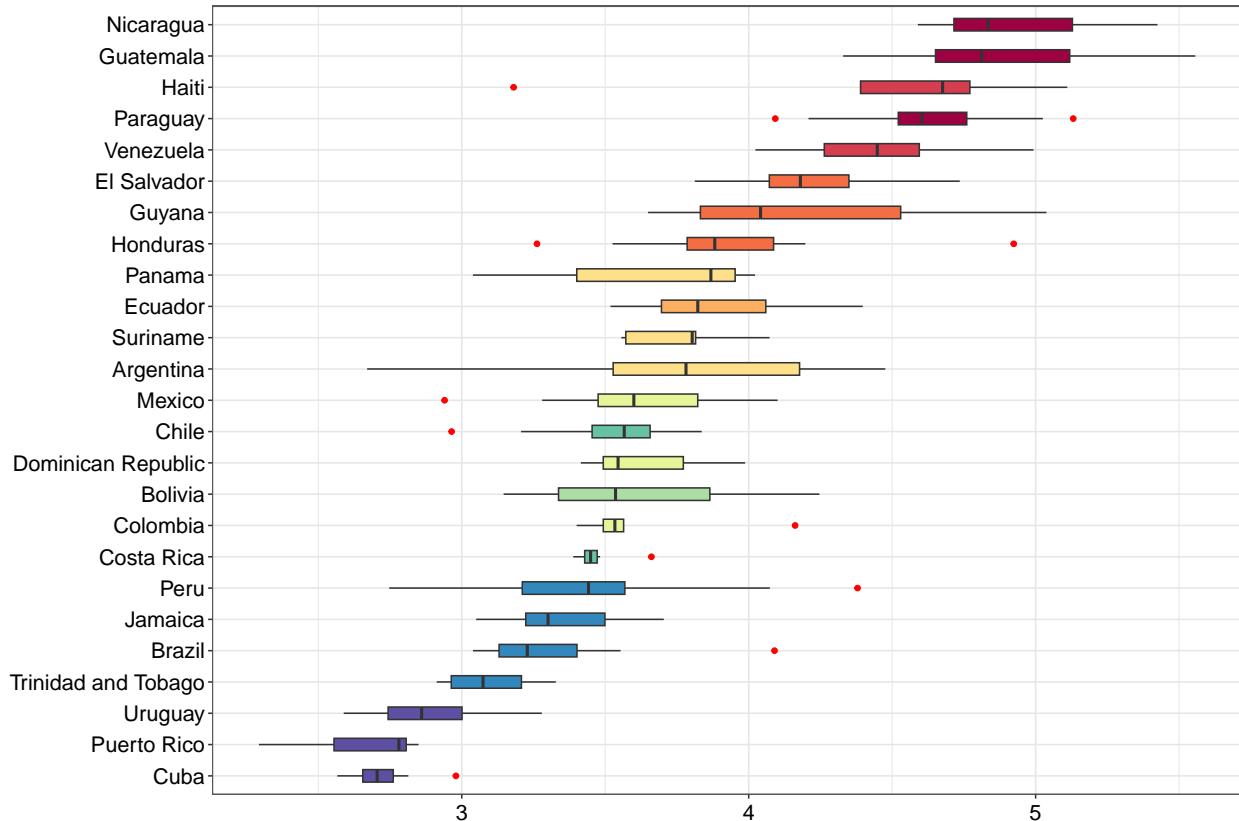
```
        round(deciles[10],2),"]",sep=""),
paste(">=",round(deciles[10],2),sep=""))
)
```

## Adjuntar colores al factor

```
names(myColors) <- levels(data$labels_mean)
```

## Visualice los datos en un diagrama de caja

```
boxplot<-ggplot(data,
  aes(C2, indicator, fill=labels_mean))+  
  geom_boxplot(alpha=1,  
  width=.4,  
  outlier.colour = "red") +  
  scale_fill_manual (name = "Average\nHH size",  
  values = c(myColors),  
  guide = guide_legend(direction = "horizontal",  
  nrow = 1,  
  keywidth=4.5,  
  label.position = "none")) +  
  theme_alap +  
  theme(legend.position = "none",  
  axis.title.y = element_blank()) +  
  coord_flip();boxplot
```



Guarde el diagrama de caja como un archivo de imagen

```
ggsave("fig_1.png",
  scale = 1,
  dpi = 300,
  height =12.5,
  width = 11.75)
```

### Ejercicio 3.1: Organizar dos Gráficos en un *grid*

Este script de R crea una imagen en CUADRÍCULA (grid) que combina un mapa de coropletas y un diagrama de caja, visualizando efectivamente los datos del tamaño de los hogares por países y regiones.

El primer componente, `worldmap2`, es un mapa refinado que muestra las regiones subnacionales de América Latina con colores de relleno que representan las categorías de deciles del tamaño de los hogares. El mapa utiliza una paleta “Spectral” invertida y un tema minimalista (`theme_void`) para garantizar una visualización clara, con la leyenda posicionada en la parte inferior.

El segundo componente es un diagrama de caja (`boxplot`) creado anteriormente, que visualiza la distribución del tamaño de los hogares entre países, ordenados según sus valores medianos. Utilizando el paquete `gridExtra`, estas dos visualizaciones se organizan una al lado de la otra en un diseño de rejilla con

`grid.arrange`, formando una comparación integral de los patrones de tamaño de los hogares a nivel regional y por país.

La imagen combinada se guarda como un archivo PNG de alta resolución (`fig_2.png`), con dimensiones optimizadas para su legibilidad en presentaciones o publicaciones. Este diseño facilita una vista integrada, conectando las tendencias regionales del mapa con las distribuciones específicas de cada país en el diagrama de caja, proporcionando una representación rica e interactiva de los datos de tamaño de los hogares.

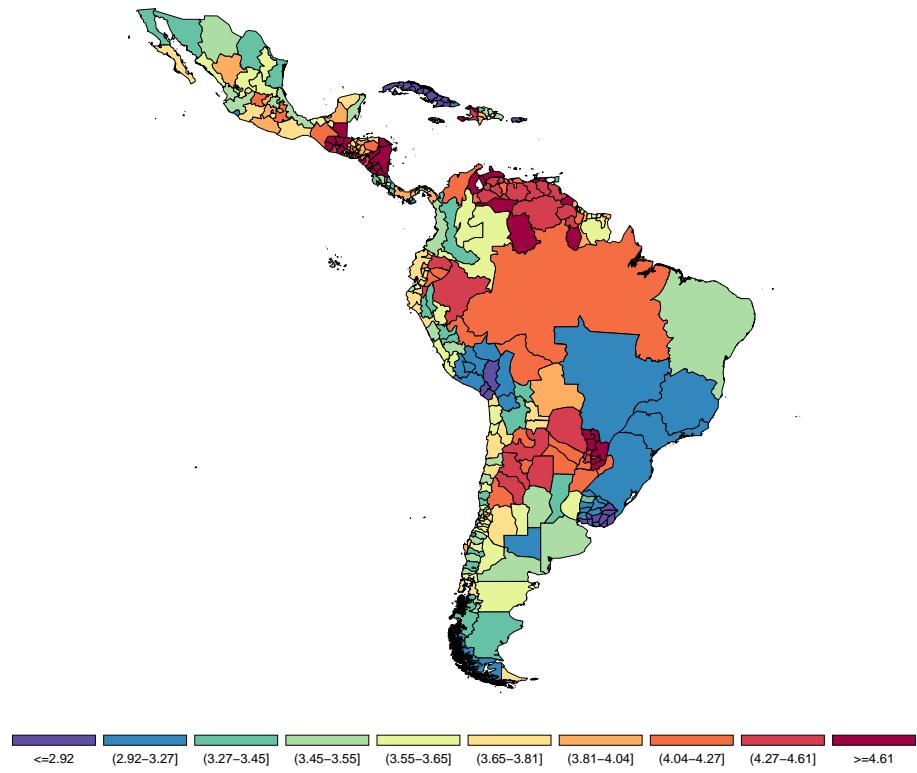
---

## Crear un *grid* de objetos *ggplot*

Crea un nuevo mapa sin título ni leyenda.

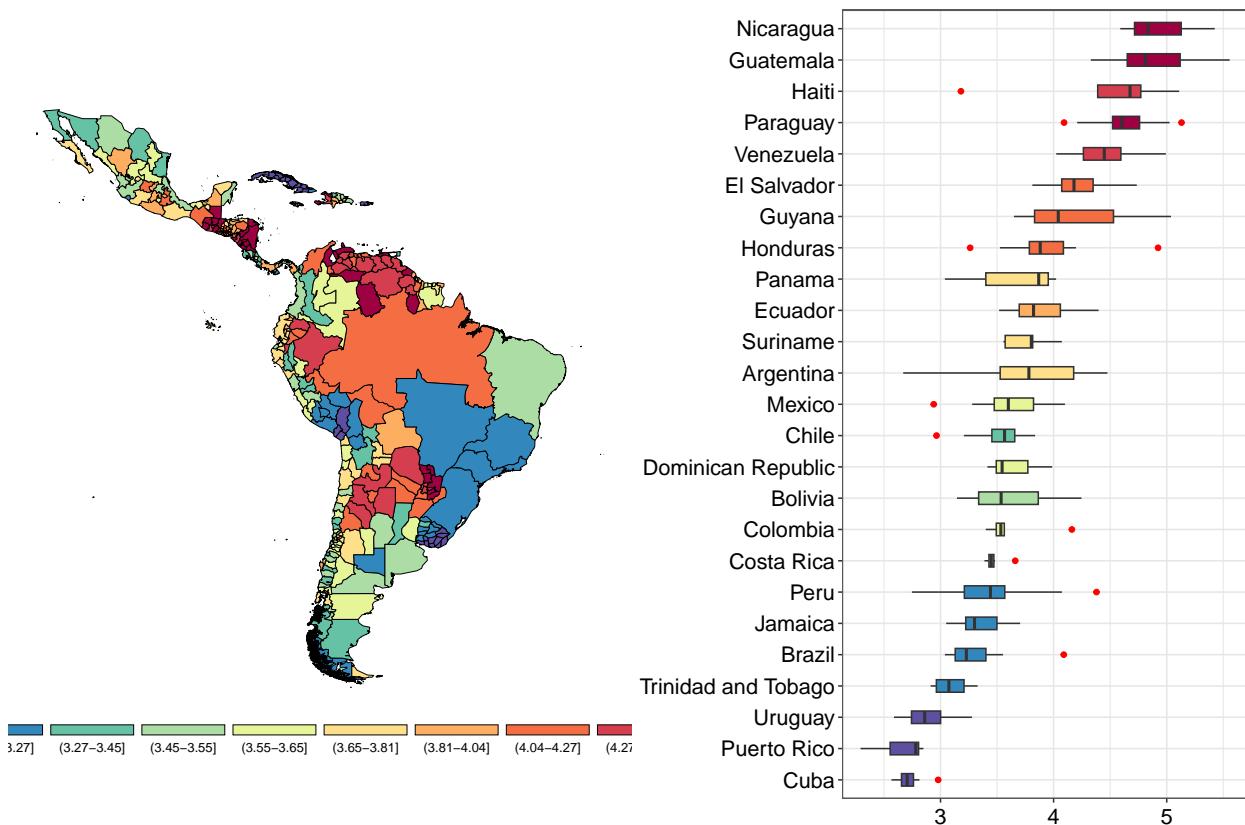
```
worldmap2 <- ggplot() +
  geom_sf(data = WORLD_C01, aes(fill = labels), colour = "black", linewidth = .075) +
  scale_fill_manual(values = rev(brewer.pal(10, "Spectral"))),
  guide = guide_legend(direction = "horizontal",
    nrow = 1,
    keyheight = .5,
    keywidth = 4,
    label.position = "bottom")) +
  theme_void() +
  theme(legend.position = "bottom",
    legend.title = element_blank(),
    plot.caption = element_text(vjust = 0.5, size = 10, colour = "black"),
    plot.margin = unit(c(0.5, 0, 0.5, 0), "cm"),
    panel.background = element_rect(fill = "white", color = "white"),
    plot.background = element_rect(fill = "white", color = "white"),
    panel.grid.major = element_line(color = "white", linewidth = 0.15, linetype = 2),
    panel.grid.minor = element_line(color = "white", linewidth = 0.1, linetype = 1))

worldmap2
```



Organice una cuadrícula con los dos objetos

```
library(gridExtra)
map_box<-grid.arrange(worldmap2, boxplot,
                      ncol = 2, nrow = 1)
```



```
map_box
```

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z    cells    name      grob
## 1 1 (1-1,1-1) arrange_gtable[layout]
## 2 2 (1-1,2-2) arrange_gtable[layout]
```

**Guarde la imagen de la cuadrícula ordenada como un archivo de imagen**

```
ggsave("fig_2.png",
       plot=map_box,
       scale = 1,
       dpi = 300,
       height =12.5,
       width = 22.75)
```

## Ejercicio 4: Hogares unipersonales en México

Este script de R genera un mapa de coropletas facetado que ilustra la prevalencia de hogares unipersonales en México por región a lo largo de múltiples años. El proceso comienza creando un objeto espacial (**sf**) de

México (`mex_shp`) filtrado del shapefile global (`WORLDC0`). Se extraen datos estadísticos específicos de México desde la base de datos CORESIDENCE (`SUBNATHA_DB`), seleccionando las variables relevantes, incluyendo `HS01`, que representa la proporción de hogares unipersonales.

Para mejorar la interpretación, la variable `HS01` se categoriza en cinco rangos porcentuales (<3%, 3–6%, 6–9%, 9–12%, 12–16%) utilizando `mutate` y `if_else`. Las categorías resultantes se reorganizan con `fct_relevel` para una visualización consistente. Los datos espaciales y estadísticos se combinan con `left_join`, y la variable del año (`T1`) se convierte en un factor para crear los facetos.

Se define una paleta de colores personalizada de la gama “YlOrRd” utilizando `RColorBrewer` para representar las categorías porcentuales. Usando `ggplot2`, el script superpone los datos regionales de México (`mexico_shp2`) sobre un mapa base de los países del mundo (`world`). El mapa incluye facetas para cada año, mostrando la distribución espacial de los hogares unipersonales a lo largo del tiempo. La leyenda de colores se personaliza y se posiciona dentro del área del gráfico para mayor claridad, mientras que elementos visuales como las etiquetas de los ejes y las marcas se ocultan para enfatizar el contenido del mapa.

El mapa facetado final se guarda como un archivo PNG de alta resolución (`map_3.png`), adecuado para análisis y presentaciones. Esta visualización destaca las tendencias regionales y temporales en la prevalencia de hogares unipersonales, apoyando análisis demográficos y espaciales en México.

---

## Crear un objeto espacial de México por regiones

```
mex_shp <- WORLDC0 |>
  filter(CNTRY_NAME == "Mexico")
```

## Crear un marco de datos con datos mexicanos

El indicador `HS01` se refiere a la proporción de hogares unipersonales.

```
mex_data <- SUBNATHA_DB |>
  filter(C2 == "Mexico") |>
  select(1:14, "HS01")
```

## Calcular *pretty\_breaks*

Calcular cortes y ordenar categorías para la leyenda del mapa.

```
mex_data<-mex_data|>
  mutate(HS01_c=if_else(HS01<0.03, "<3%",
                        if_else(HS01<0.06, "3-6%",
                               if_else(HS01<0.09, "6-9%",
                                      if_else(HS01<0.12, "9-12%", "12-16%")))),
  HS01_c=fct_relevel(HS01_c, "<3%",
                      "3-6%",
                      "6-9%",
                      "9-12%",
                      "12-16%"))
```

## Unir datos espaciales y estadísticos

```
mexico_shp2 <- mex_shp |>
  left_join(mex_data, by = "C5") |>
  mutate(T1 = as.factor(T1))
```

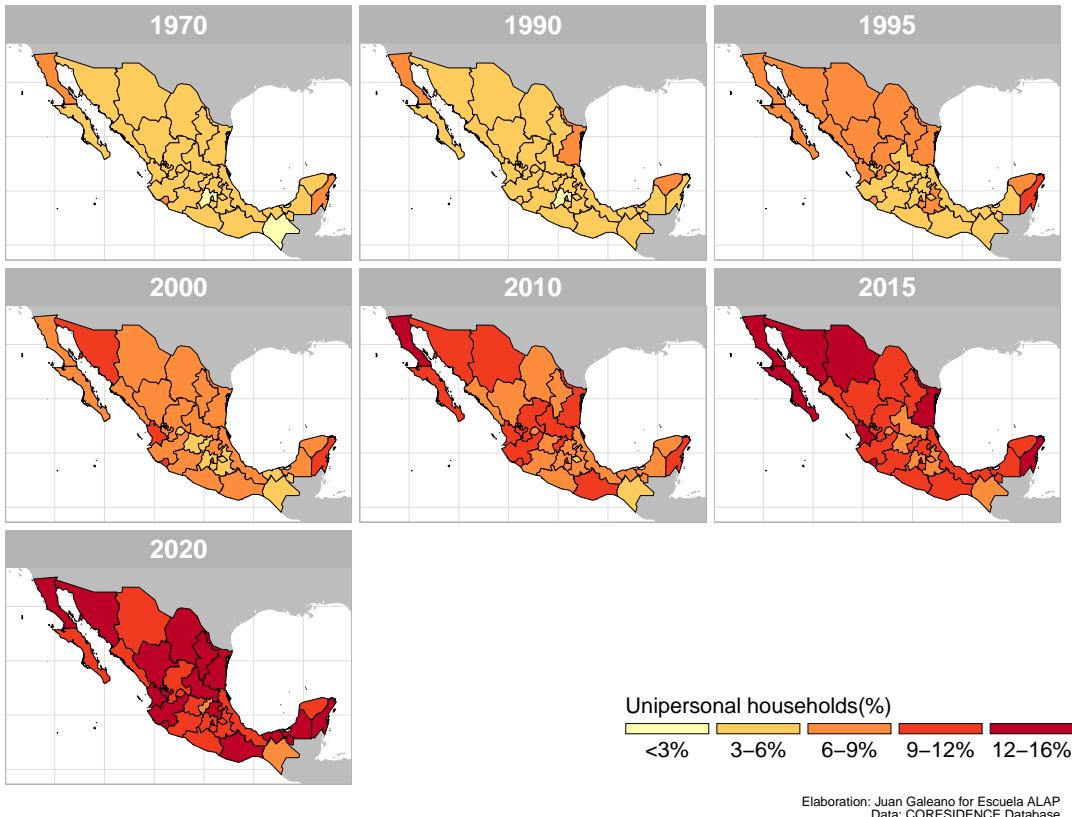
## Crea una paleta de colores usando RColorBrewer

```
myColors <- c(brewer.pal(5, "YlOrRd"))
```

## Trazar un mapa coroplético facetado

```
world <- gisco_get_countries(resolution = "03", epsg = 4326)
Map1 <- ggplot() +
  geom_sf(data= world, fill = "#bdbdbd",
          colour = "#bdbdbd", linewidth=.05) +
  geom_sf(data=mexico_shp2,
          aes(fill = HS01_c), colour = "Black", linewidth=.05) +
  scale_fill_manual(name="Unipersonal households(%)",
                    values=myColors,
                    guide = guide_legend(direction = "horizontal",
                                          title.position = "top",
                                          nrow = 1,
                                          keyheight=.5,
                                          keywidth=4,
                                          label.position = "bottom")) +
  facet_wrap(~T1) +
  coord_sf(
    xlim = c(-118.37075, -86.70731),
    ylim = c(14.53455, 32.71878)) +
  labs(caption="Elaboration: Juan Galeano for Escuela ALAP\nData: CORESIDENCE Database") +
  theme_light() +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        strip.text = element_text(size=18, face="bold"),
        legend.position = "inside",
        legend.position.inside = c(.80, 0.075),
        legend.text = element_text(colour="black", size = 15),
        legend.title= element_text(colour="black", size = 15),
        axis.ticks = element_blank())
```

Map1



Guardar el mapa facetado como un archivo de imagen

```
ggsave("map_3.png",
  scale = 1,
  height = 8,
  width=12,
  dpi = 300)
```

## Ejercicio 4.1: Hogares unipersonales en México, geofacet

¿Qué es geofacet?

La librería geofacet en R permite crear facetas geográficas organizadas en una cuadrícula que respeta la disposición espacial aproximada de las regiones que se analizan (por ejemplo, estados, provincias o países). Esta herramienta es especialmente útil para visualizar datos distribuidos geográficamente, manteniendo la coherencia espacial en los gráficos facetados.

```
mex_data<-mex_data|>
  mutate(name_official=C6,
    name_official=ifelse(name_official=="Distrito Federal", "Ciudad de México", name_official))
```

```

library(geofacet)

GRID_MX<-mx_state_grid3

GRID_MX$name<-with(GRID_MX,if_else(name=="Ciudad de México", "CDMX",
                                     if_else(name=="Baja California Sur", "Baja California S",name)))

df <- merge(mex_data, GRID_MX, by = "name_official", all.x = TRUE)

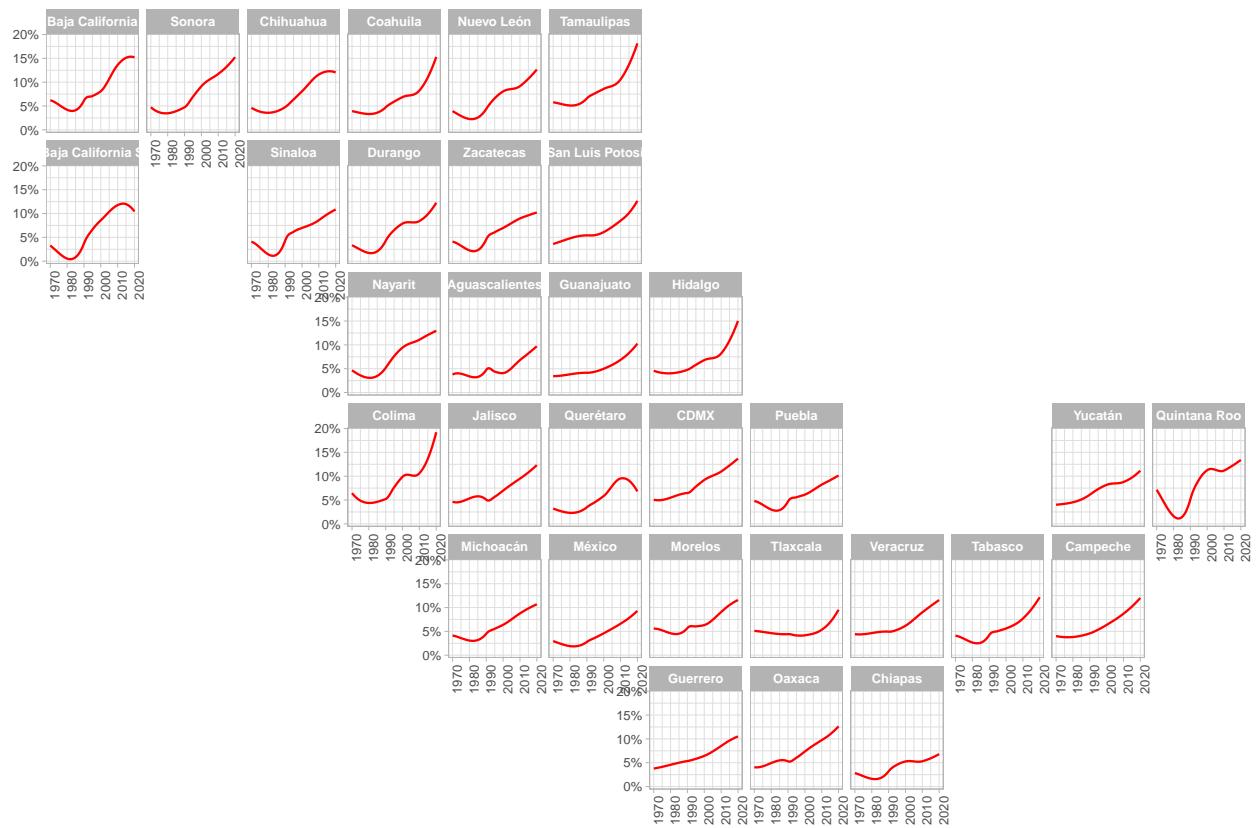
```

## Trazar un gráfico de líneas por estados con geogacet

```

df %>%
  ggplot() +
  geom_smooth(stat = "smooth",
              aes(T1, HS01),
              colour="red", se=FALSE, linewidth=.75) +
  # coord_flip() +
  scale_y_continuous(labels = scales::percent) +
  facet_geo(~ name, grid = GRID_MX) +
  theme_light() + theme(legend.position = "none",
                        axis.text.x = element_text(angle = 90 ),
                        axis.title.x = element_blank(),
                        axis.title.y = element_blank(),
                        strip.text = element_text(face="bold"))

```



## Guardar el mapa facetado como un archivo de imagen

```
ggsave("map_4.png",
       scale = 1,
       height = 8,
       width=13,
       dpi = 300)
```

## Ejercicio 5: Configuraciones de convivencia los hogares, Uruguay 1963-2011

Este script de R analiza la distribución relativa de las configuraciones de convivencia en los hogares según la edad y el sexo en Uruguay utilizando datos de IPUMS. El proceso comienza cargando la base de datos CORESIDENCE y extrayendo datos específicos de Uruguay. El conjunto de datos se divide en una lista por año de muestra para calcular las distribuciones relativas de las configuraciones de convivencia en los hogares (`n_rel`) por edades simples (`AGE2`) y sexo sobre cada uno de los elementos de la lista. Las agregaciones se realizan a nivel de las configuraciones de convivencia en los hogares (`LA`) y subclasificación (`LAQ`). Los datos faltantes se manejan asignando ceros a los valores vacíos en las variables clave.

Las clasificaciones de las configuraciones de convivencia en los hogares se importan desde un archivo Excel externo y se utilizan para crear combinaciones distintas de `LA` y `LAQ`. Estas se combinan con combinaciones únicas de años, edades y sexo mediante cruces (`cross-join`), asegurando un conjunto de datos completo para el análisis. Se crean factores personalizados para variables como `SEX` (recodificados como “Men” y “Women”) y `LAQ` (reordenados para reflejar categorías jerárquicas de arreglos de vivienda).

Se genera una paleta de colores personalizada utilizando `RColorBrewer` y se modifica para mejorar la diferenciación visual de las categorías de arreglos de vivienda. Además, se vuelve a utilizar nuestro tema predeterminado anteriormente (`theme_alap`) para estandarizar el diseño, incluidos tamaños de fuente, etiquetas de ejes y la posición de las leyendas.

La visualización se centra en el primer (1963) y último (2011) año de datos disponibles. Se crea un gráfico de áreas facetado para mostrar las distribuciones relativas de los arreglos de vivienda a través de las edades, con facetas por año y sexo. El gráfico utiliza suavizado `loess` para una visualización continua de áreas, con el eje y representando porcentajes y el eje x grupos de edad. El diseño apilado resalta eficazmente las diferencias proporcionales en los arreglos de vivienda a lo largo del tiempo.

El gráfico final está diseñado con etiquetas claras, una leyenda posicionada en la parte inferior y un fondo blanco limpio para la claridad en la presentación. Esta visualización proporciona una visión integral de los cambios en los arreglos de vivienda a lo largo del ciclo de vida, segmentada por sexo, ofreciendo perspectivas sobre los cambios sociales y demográficos en Uruguay.

---

## Cargar y preparar los datos

```
load("CORESIDENCE_DATABASE_LA_2024.Rda")

# Extraer los datos de edades individuales de IPUMS de la base de datos GLAD
ipums <- CODBLA[["IPUMS"]]
```

## Extrar los datos disponibles de Uruguay

```
ury <- ipums |> filter(CNTRY == "Uruguay") |> select(1:14)
```

## Dividir el marco de datos en una lista por muestras

```
ury_list <- split(ury, f = ury$SAMPLE)
```

Calcular la distribución relativa de los arreglos de vida por sexo y edades individuales para cada uno de los elementos de la lista

```
list1 <- lapply(ury_list, function(df) {
  df <- df |>
    mutate(SEX = unclass(SEX),
           AGE2 = if_else(AGE < 80, AGE, 80)) |>
    filter(SEX != 9) |>
    group_by(SAMPLE, SEX, AGE2, LA, LAQ) |>
    summarise(pop = sum(POPW)) |>
    ungroup()

  df <- df |>
    mutate(n_rel = pop / sum(pop), .by = c(SEX, AGE2))
})
```

## Combine los elementos de lista en un único marco de datos

```
final <- data.table::rbindlist(list1)

# Agregar información del año desde la columna SAMPLE
final <- final |> mutate(YEAR = as.numeric(substr(SAMPLE, 5, 8)))
```

## Leer archivo excel con la clasificación de los arreglos de vida

```
GELAI_AGRUPACION <- read_excel("GELAI_AGRUPACION.xlsx",
                                sheet = "GELAI_TO_LA")

# Crea un marco de datos con distintas combinaciones de LA y LAQ
LAQ <- GELAI_AGRUPACION |> distinct(LA, LAQ)
```

## Crear un marco de datos con años, edades individuales y sexo

```

unique_ages <- data.frame(
  YEAR = rep(unique(final$YEAR), each = 81 * 2),
  AGE2 = rep(0:80, length(unique(final$YEAR)) * 2),
  SEX = rep(rep(c(1, 2), each = 81), length(unique(final$YEAR)))
)

# Realizar una unión cruzada entre edades únicas y LAQ
cross_join <- unique_ages %>%
  mutate(key = 1) %>%
  inner_join(LAQ %>% mutate(key = 1), by = "key") %>%
  select(-key)

# Fusionar datos unidos de forma cruzada con el conjunto de datos final
final <- cross_join %>%
  left_join(final, by = c("YEAR", "SEX", "AGE2", "LA", "LAQ"))

# Reemplazar NA en variables cruciales
final$SAMPLE <- paste("URY_", final$YEAR, "_IPUMS", sep = "")
final$pop[is.na(final$pop)] <- 0
final$n_rel[is.na(final$n_rel)] <- 0

```

## Recodificar y renivelar variables

```

final <- final |>
  mutate(SEX = as.factor(SEX),
         SEX = fct_recode(SEX, Men = "1", Women = "2"),
         LAQ = as.factor(LAQ),
         LAQ = fct_relevel(LAQ,
                           "Alone",
                           "With single parent",
                           "With single parent extended",
                           "With single parent extended composite",
                           "With parents",
                           "With parents extended",
                           "With parents extended composite",
                           "With partner",
                           "With partner extended",
                           "With partner extended composite",
                           "With partner and children",
                           "With partner and children extended",
                           "With partner and children extended composite",
                           "With children",
                           "With children extended",
                           "With children extended composite",
                           "Extended",
                           "Extended composite",
                           "Non-relative"))

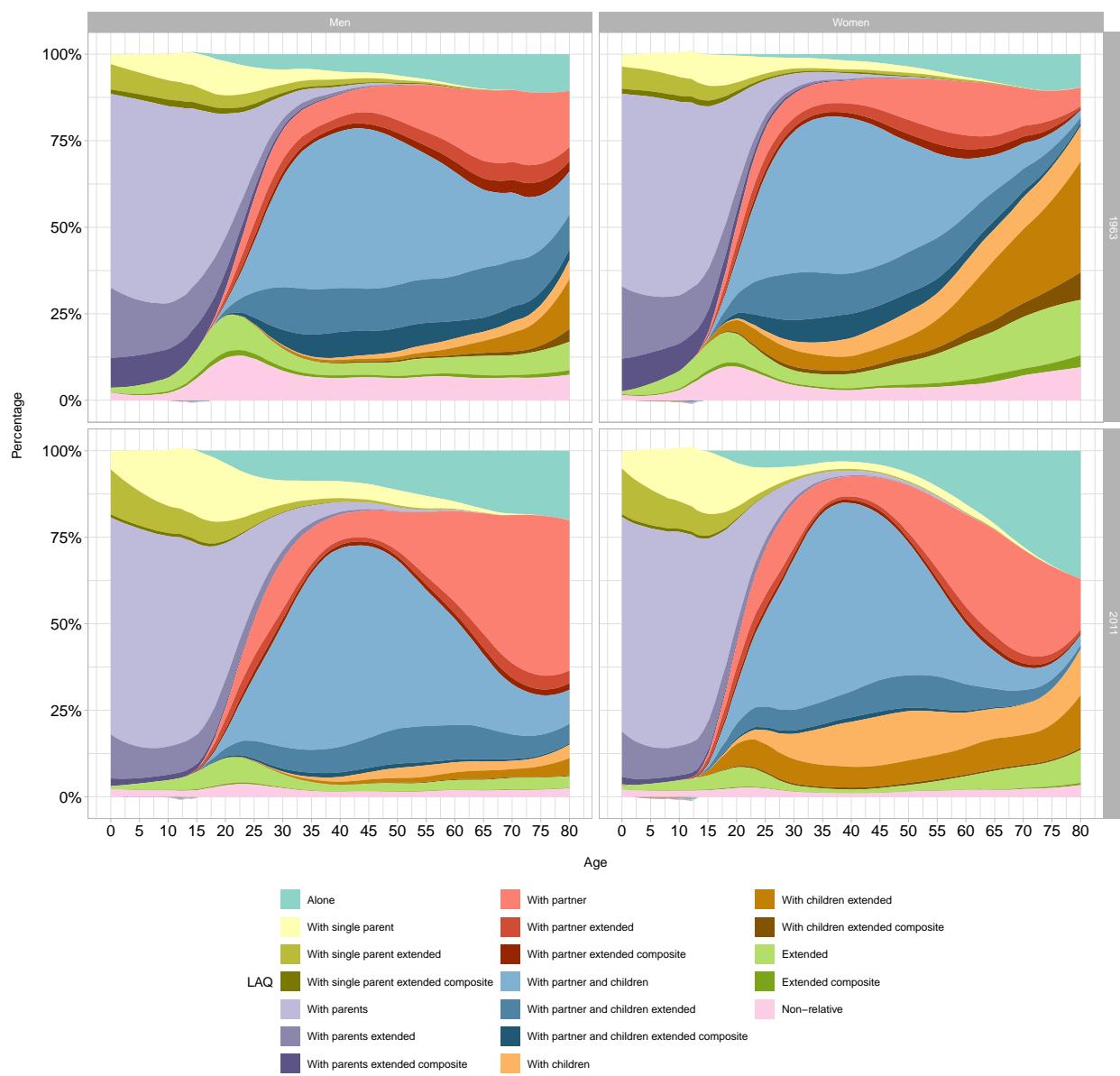
```

## Crea una paleta de colores personalizada

```
library(colorspace)
library(RColorBrewer)

pal <- "Set3"
my_colors <- c(
  brewer.pal(8, pal)[1],
  brewer.pal(8, pal)[2], darken(brewer.pal(8, pal)[2], 0.25), darken(brewer.pal(8, pal)[2], 0.5),
  brewer.pal(8, pal)[3], darken(brewer.pal(8, pal)[3], 0.25), darken(brewer.pal(8, pal)[3], 0.5),
  brewer.pal(8, pal)[4], darken(brewer.pal(8, pal)[4], 0.25), darken(brewer.pal(8, pal)[4], 0.5),
  brewer.pal(8, pal)[5], darken(brewer.pal(8, pal)[5], 0.25), darken(brewer.pal(8, pal)[5], 0.5),
  brewer.pal(8, pal)[6], darken(brewer.pal(8, pal)[6], 0.25), darken(brewer.pal(8, pal)[6], 0.5),
  brewer.pal(8, pal)[7], darken(brewer.pal(8, pal)[7], 0.25),
  brewer.pal(8, pal)[8]
)
```

## Crear una gráfico de area por edades simples y sexo



## Guardar el mapa facetado como un archivo de imagen

```
ggsave("fig_3.png",
       scale = 1,
       height = 10,
       width=12,
       dpi = 300)
```