Elaborar una Pirámide de Población

Alin Castillo (alincastillo1995@gmail.com)

Contents

Carga de paquetes

```
library(readxl)
library(dplyr)
library(tidyr)
library(pyramid)
library(ggplot2)
```

Carga de funciones locales

```
source(file = "function-pyramid-modified.R")
source(file = "function-pyramid-own.R")
```

Obtención de los datos

Los datos se obtuvieron del INEI y pueden ser descargados de forma directa en el siguiente link:
Población total estimada y proyectada al 30 de junio, por año quinquenal, según sexo y grupo de edad

Carga de los datos

Al ser un archivo excel por bloques, realizamos la carga de la población de hombres y mujeres por separado.

```
# Creamos una variable que contenga los nombres de las variables
variables_names <- c("Edad", "Pobl_2000", "Pobl_2005", "Pobl_2010",
                     "Pobl_2015", "Pobl_2020", "Pobl_2021", "Sexo")
```

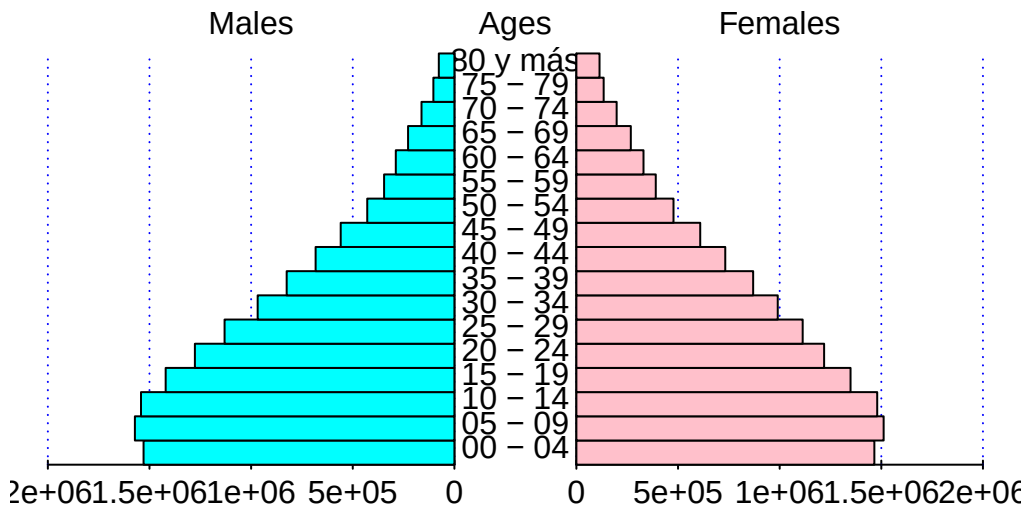
```r
# Cargamos los datos de población de los hombres
dataH <- read_excel(path = "../data/proy_03.xlsx",
                    range = "A27:H43",
                    col_names = F) %>%
  select(-2) %>%
  mutate(sexo = "H")

names(dataH) <- variables_names

# Cargamos los datos de población de las mujeres
dataM <- read_excel(path = "../data/proy_03.xlsx",
                    range = "A46:H62",
                    col_names = F) %>%
  select(-2) %>%
  mutate(sexo = "M")

names(dataM) <- variables_names

# Unimos los datos
data <- rbind(dataH, dataM)
```

Manipulación de datos

```r
data <- data %>%
  select(Edad, Sexo, Pobl_2000) %>%
  spread(key = Sexo, value = Pobl_2000) %>%
  select(H, M, Edad) %>%
  mutate(
    Edad = case_when(
      Edad == "5 - 9" ~ "05 - 09",
      Edad == "0 - 4" ~ "00 - 04",
      TRUE ~ Edad),
    H = as.integer(H),
    M = as.integer(M)
  ) %>%
  arrange(Edad) %>%
  as.data.frame()
```

Generamos la pirámide poblacional para el año 2000

```r
pyramid(data)
```



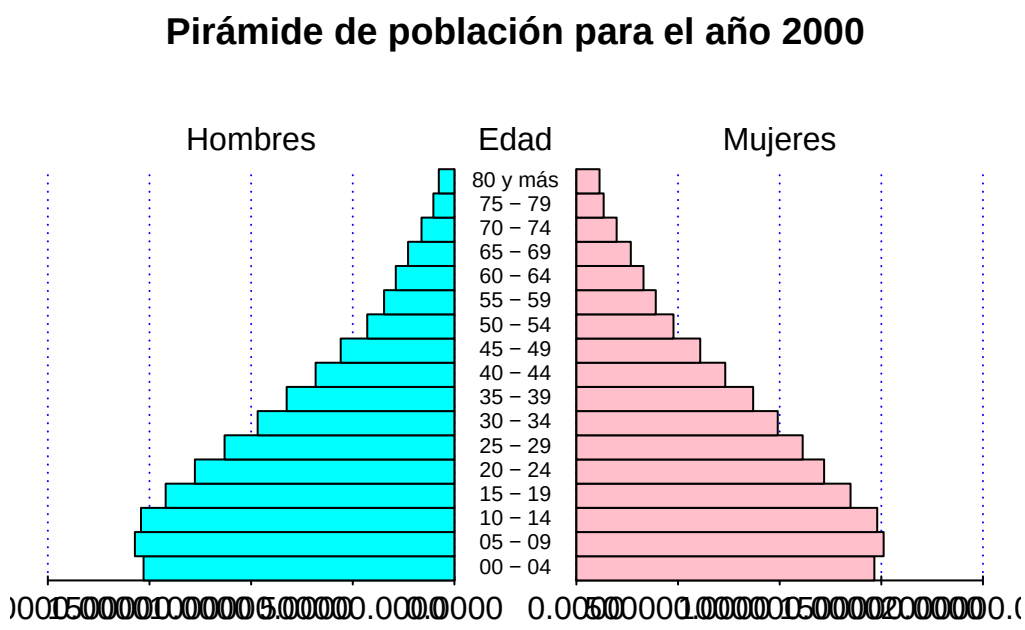
Maquillamos el gráfico de la pirámide

Para esto vamos a revisar los argumentos de la función

- *data*: un marco de datos
- *Laxis*: un vector para los ejes de la izquierda
- *Raxis*: un vector para los ejes de la derecha
- *AxisFM*: argumento format de la función formatC
- *AxisBM*: argumento big.mark de la función formatC
- *AxisBI*: argumento big.interval de la función formatC
- *Cgap*: Ancho de la parte central
- *Cstep*: Intervalo para escribir las etiquetas de la edad
- *Csize*: Tamaño de fuente para escribir las etiquetas de la edad
- *Cadj*: Ajuste vertical para las etiquetas de la clase edad
- *Llab*: Etiqueta de la pirámide izquierda
- *Rlab*: Etiqueta de la pirámide derecha
- *Clab*: Etiqueta de la pirámide central
- *GL*: Dibujar las lineas verticales
- *Lcol*: Color de la pirámide izquierda
- *Ldens*: Densidad de las lineas de sombreado de la pirámide izquierda
- *Rcol*: Color de la pirámide derecha
- *Rdens*: Densidad de las lineas de sombreado de la pirámide derecha
- *main*: Título de la pirámide

```
pyramid(data, AxisFM = "f",
        Csize = 0.7,
        Llab = "Hombres",
```

```
        Rlab = "Mujeres",
        Clab = "Edad",
        main = "Pirámide de población para el año 2000")
```



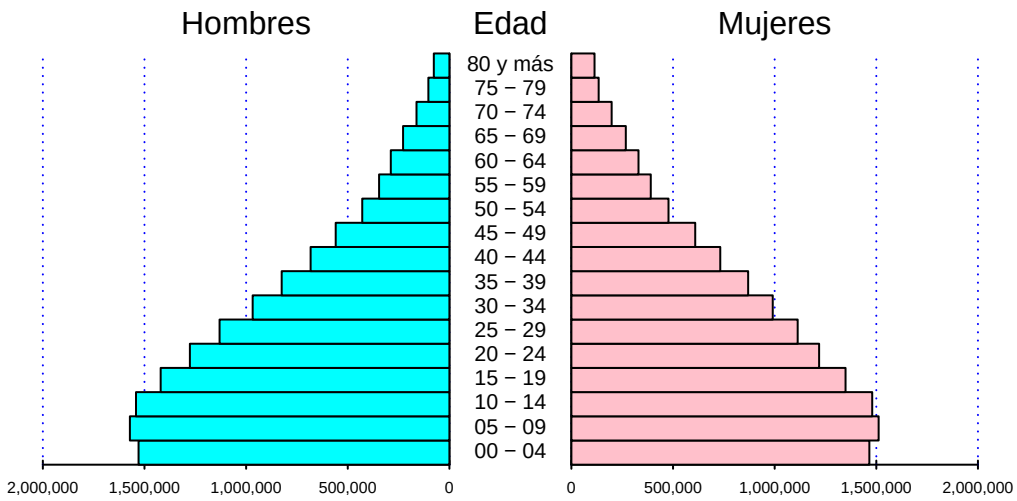
Modificando la funcíon *pyramid* para una mejor visualización

Como hemos visto en el gráfico anterior, tenemos problemas con la etiqueta de los ejes, esto se debe a que al ser números tan grandes no se lográn visualizar, con lo cual he agregado argumentos adicionales a la función para una mejor presentación.

- *Interval*: Intervalo personalizado
- *NDigits*: Número de decimales, por defecto 2
- *Und*: Unidad de medida
- *Nsize*: Tamaño de letra del eje x, por defecto 1
- *Tsize*: Tamaño de letra la descripción de las pirámides.

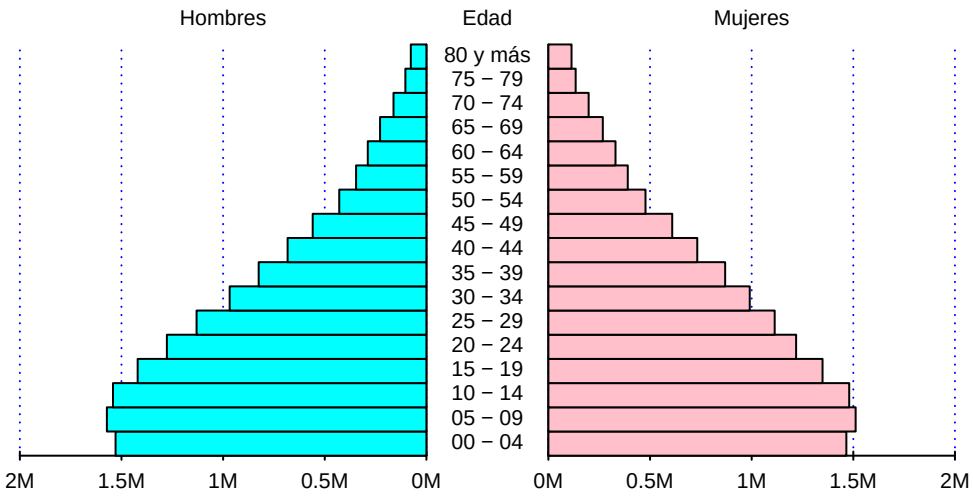
Primero intentemos con el argumentos *Nsize* y *NDigits* es decir, cambiando el tamaño y el número de decimales.

```
piramide(data, AxisFM = "f", AxisBM = ",",
         Csize = 0.7,
         Nsize = 0.5,
         NDigits = 0,
         Llab = "Hombres",
         Rlab = "Mujeres",
         Clab = "Edad",
         main = "Pirámide de población para el año 2000")
```



Podemos observar que mejora considerablemente, ahora provemos con los arguntos *Interval* y *Und* para mejorar aún mas la visualización. Si tiene problemas con la defición de intervalos puede usar la función local *intervals* y olvidarse del problema, para esto solo tiene que descomentar *intervals <- intervals(data)* para observar un resultado similar.

```
# Contruimos el vector con los intervalos
intervals <- c(0, 0.5, 1, 1.5, 2)
#intervals <- intervals(data)

piramide(data, AxisFM = "f",
         Csize = 0.7,
         Nsize = 0.7,
         Tsize = 0.7,
         Interval = intervals,
         Und = "M",
         Llab = "Hombres",
         Rlab = "Mujeres",
         Clab = "Edad",
         main = "Pirámide de población para el año 2000 en millones")
```



Función propia acPiramide() basado en ggplot2

Argumentos

- *data*: Un marco de datos
- *intervals*: Intervalos de los datos
- *digits*: Número de decimales, valor por defecto 0
- *labels*: Etiqueta para los intervalos de los datos
- *und*: Unidad de medida cuando se use el argumento *labels*
- *color*: Color basado en la metrica de la función *scale_fill_brewer*
- *title*: Título del gráfico
- *htitle*: Posción del título, valor por defecto 0.5 (Centrado)
- *sizetitle*: Tamaño de la letra del título, valor por defecto 20
- *subtitle*: Subtítulo del gráfico
- *hsubtitle*: Posción del subtítulo, valor por defecto 0.5 (Centrado)
- *sizesubtitle*: Tamaño de la letra del subtítulo, valor por defecto 10

A continuación mostramos la estructura que debe tener el marco de datos para poder usar la función.

Tipos de colores a utilizar en el argumento *color*

Puede visualizar la lista de colores ejecutando en su colonsola de R el siguiente comando:

```
RColorBrewer::display.brewer.all()
```

Puede encontrar una descripción detallada de la gamma de colores en el siguiente link:

ggplot2 Reference and Examples

En resumen el paquete **RcolorBrewer** nos ofrece 3 gamas de colores:

- Paletas Cualitativas
- Paletas Secuenciales
- paletas divergentes

Se detalla la paleta de colores cualitativa ya que es la de nuestro interés.

Paletas Cualitativas, Los grupos de colores disponibles son:

- Accent (8 niveles)
- Dark2 (8 niveles)
- Paired (12 niveles)
- Pastel1 (9 niveles)
- Pastel2 (8 niveles)
- Set1 (9 niveles)
- Set2 (8 niveles)
- Set3 (12 niveles)

Cabe mencionar que nosotros solo usamos dos niveles, solo tenemos dos categorías, con lo cual cualquiera de los 8 grupos de colores son suficientes.

Elaboración del marco de datos

```
# Creamos una variable que contenga los nombres de las variables
variables_names <- c("Edad", "Pobl_2000", "Pobl_2005", "Pobl_2010",
                     "Pobl_2015", "Pobl_2020", "Pobl_2021", "Sexo")

# Cargamos los datos de población de los hombres
dataH <- read_excel(path = "../data/proy_03.xlsx",
                    range = "A27:H43",
                    col_names = F) %>%
  select(-2) %>%
  mutate(sexo = "H")

names(dataH) <- variables_names

# Cargamos los datos de población de las mujeres
dataM <- read_excel(path = "../data/proy_03.xlsx",
                    range = "A46:H62",
                    col_names = F) %>%
  select(-2) %>%
  mutate(sexo = "M")

names(dataM) <- variables_names

# Unimos los datos
data <- rbind(dataH, dataM)

data <- data %>%
  select(Edad, Sexo, Pobl_2000) %>%
  mutate(Pobl_2000 = case_when(
    Sexo == "H" ~ Pobl_2000 * -1,
    TRUE ~ Pobl_2000
  ),
  Edad = as.factor(case_when(
```

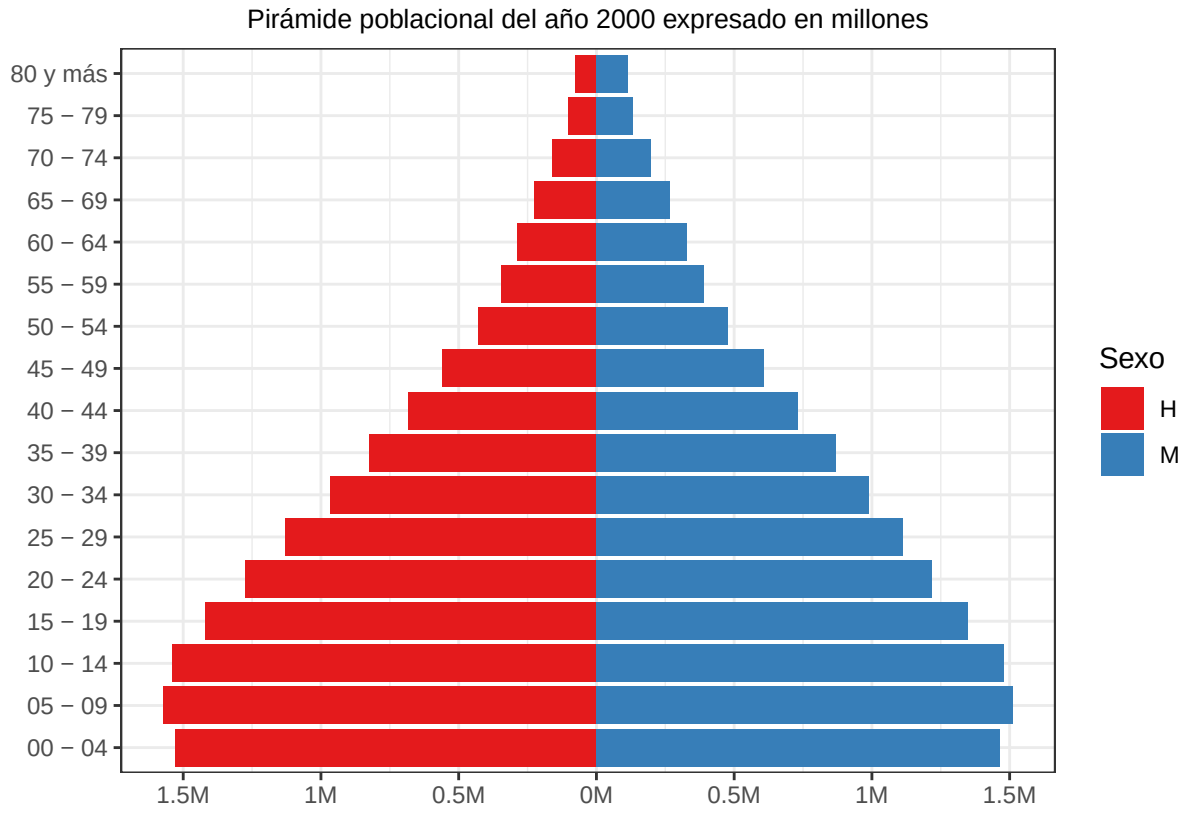
```
    Edad == "5 - 9" ~ "05 - 09",
    Edad == "0 - 4" ~ "00 - 04",
    TRUE ~ Edad)))

data %>% head(5)
```

```
## # A tibble: 5 x 3
##   Edad    Sexo  Pobl_2000
##   <fct>   <chr>     <dbl>
## 1 00 - 04 H      -1529128
## 2 05 - 09 H      -1571815
## 3 10 - 14 H      -1541751
## 4 15 - 19 H      -1420347
## 5 20 - 24 H      -1276691
```

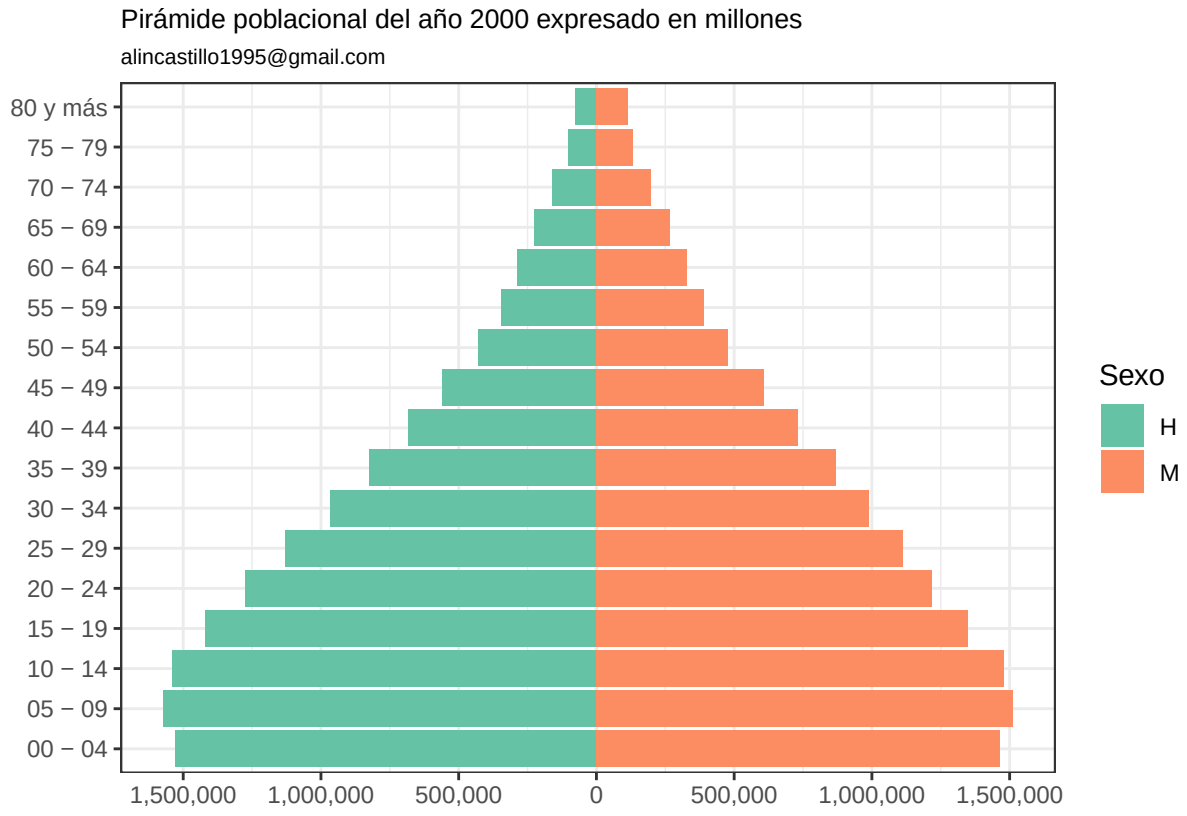
Pirámide poblacional

Se muestran dos ejemplos con distinto uso de los argumentos, si tiene dificultad para determinar los intervalos puede usar la función local *acIntervals* el cual devuelve una lista, con los intervalos y la etiqueta de los mismos.

```
acPiramide(data = data,
           intervals = acIntervals(data)$intervals,
           labels = acIntervals(data)$labels,
           und = "M",
           title = "Pirámide poblacional del año 2000 expresado en millones",
           sizetitle = 10,
           color = "Set1")
```




```
acPiramide(data = data,
           intervals = acIntervals(data)$intervals,
           digits = 0,
           color = "Set2",
           title = "Pirámide poblacional del año 2000 expresado en millones",
           htitle = 0,
           sizetitle = 10,
           subtitle = "alincastillo1995@gmail.com",
           hsubtitle = 0,
           sizesubtitle = 8)
```



Recursos

- pyramid: Population pyramid
- formatC: Formatting Using C-style Formats
- geom_bar: Bars, rectangles with bases on x-axis
- scale_colour_brewer: Sequential, diverging and qualitative colour scales from colorbrewer.org