Procesamiento de microdatos en lenguaje R

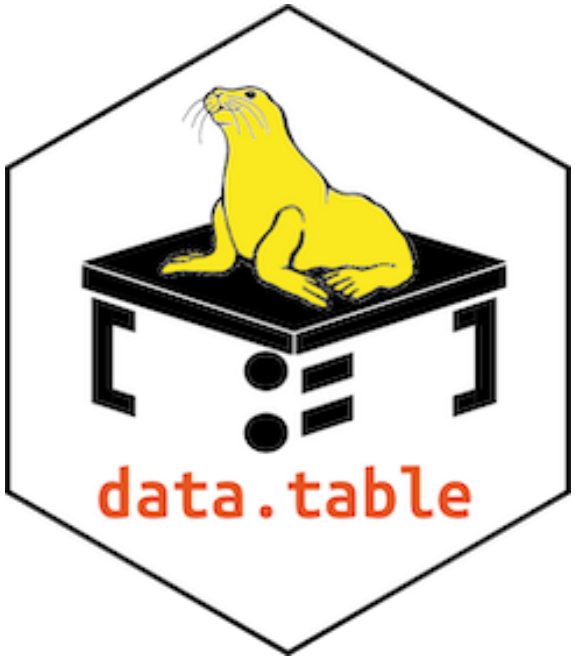
Una introducción al uso en demografía

César Andrés Cristancho-Fajardo.
Docente Universidad Santo Tomás. Experto DANE.

2022-08-18

Introducción ¿Qué es data.table?

- Es un paquete de R para trabajar con datos tabulares -Un paquete es una colección de funciones y conjuntos de datos desarrollados por la comunidad-.
- Es popular por su velocidad de ejecución para grandes bases de datos.
- La sintaxis de programación es más concisa que tidyverse.



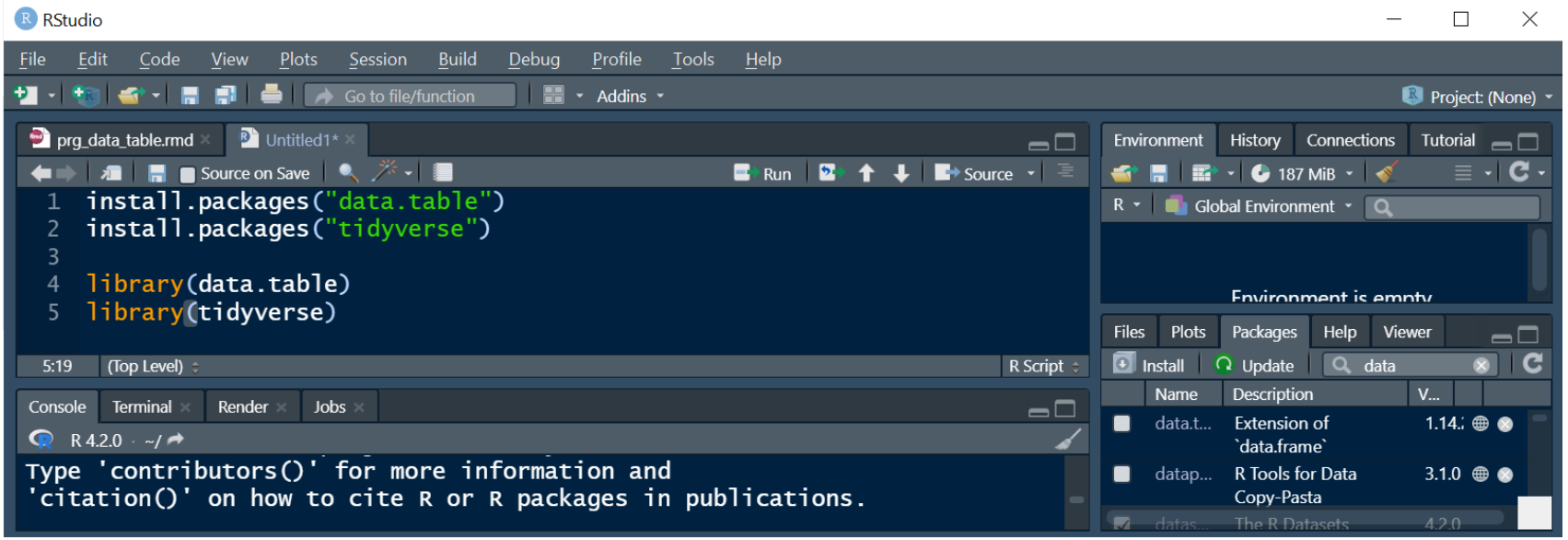
Introducción ¿Qué es tidyverse?

- Es una colección de paquetes de R diseñados para data science.
- Incluye los paquetes ggplot2 (gráficos), dplyr (procesamiento), tidyr (reestructuración de bases), readr (lectura de bases), purrr (programación funcional), tibble (data.frames optimizados), stringr (cadenas de caracteres), forcats (datos categoricos).

Instalación y carga de paquetes

```
# install.packages("data.table")
# install.packages("tidyverse")
```

```
library(data.table)
library(tidyverse)
```



Importación de datos desde un formato csv

Se debe configurar la dirección de la carpeta y en ella debe estar el archivo de trabajo.

```
setwd("D:/santo_tomas/clase sem 3 datatable")
bd <- fread('pob_sex_eds_mun_anio.csv')
glimpse(bd)
```

```
## Rows: 506,568
## Columns: 6
## $ anio     <int> 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2…
## $ cod_area <int> 5001, 5001, 5001, 5001, 5001, 5001, 5001, 5001, 5001, 5001, 5…
## $ area     <chr> "Medellín", "Medellín", "Medellín", "Medellín", "Medellín", "…
## $ edad     <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ h        <int> 18797, 16265, 16480, 16765, 17203, 18174, 18465, 18016, 18806…
## $ m        <int> 17809, 15711, 15798, 15948, 16728, 17312, 17516, 17615, 18453…
```

División político administrativa de Colombia en formato excel

Se puede descargar en un archivo formato excel de la siguiente dirección web: https://geoportal.dane.gov.co/geovisores/territorio/consulta-divipola-division-politico-administrativa-de-colombia/

Importación de un archivo de excel

```
 dpola <- readxl::read_excel('DIVIPOLA_Municipios.xlsx')
head(dpola)
```

```
## # A tibble: 6 × 7
##   `Codificación de la División Político Ad…` ...2  ...3  ...4  ...5  ...6  ...7
##   <chr>                                      <chr> <chr> <chr> <chr> <chr> <chr>
## 1 <NA>                                       <NA>  <NA>  <NA>   <NA> <NA>  <NA>
## 2 Municipios                                 <NA>  <NA>  <NA>   <NA> <NA>  <NA>
## 3 <NA>                                       <NA>  <NA>  <NA>   <NA> <NA>  <NA>
## 4 Departamento                               <NA>  Muni… <NA>  "Tip… <NA>  <NA>
## 5 Código                                     Nomb… Códi… Nomb…  <NA> LATI… LONG…
## 6 05                                         ANTI… 05001 MEDE… "Mun… 6.25… -75.…
```

Al ver el contenido del archivo se ve que se trata de datos no estructurados.

Lectura mejorada desde excel

Las opciones nos permiten configurar que se importe solo desde una cierta fila -skip- y un número determinado de filas -n_max-.

```
dpolab <- readxl::read_excel('DIVIPOLA_Municipios.xlsx', skip = 10, n_max = 1121)
glimpse(dpolab)
```

```
## Rows: 1,121
## Columns: 7
## $ Código...1 <chr> "05", "05", "05", "05", "05", "05", "05", "05", "05", "05",…
## $ Nombre...2 <chr> "ANTIOQUIA", "ANTIOQUIA", "ANTIOQUIA", "ANTIOQUIA", "ANTIOQ…
## $ Código...3 <chr> "05001", "05002", "05004", "05021", "05030", "05031", "0503…
## $ Nombre...4 <chr> "MEDELLÍN", "ABEJORRAL", "ABRIAQUÍ", "ALEJANDRÍA", "AMAGÁ",…
## $ ...5       <chr> "Municipio", "Municipio", "Municipio", "Municipio", "Munici…
## $ LATITUD    <dbl> 6.257590, 5.803728, 6.627569, 6.365534, 6.032922, 6.977789,…
## $ LONGITUD   <dbl> -75.61103, -75.43847, -76.08598, -75.09060, -75.70800, -74.…
```

Filtrado de subconjuntos de filas

Se debe reemplazar 52001 por el código DIVIPOLA de su municipio de interés. El código 52001 corresponde a Pasto.

```
bds <- bd[cod_area == 52001]
glimpse(bds)
```

```
## Rows: 607
## Columns: 6
## $ anio     <int> 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2…
## $ cod_area <int> 52001, 52001, 52001, 52001, 52001, 52001, 52001, 52001, 52001…
## $ area     <chr> "Pasto", "Pasto", "Pasto", "Pasto", "Pasto", "Pasto", "Pasto"…
## $ edad     <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ h        <int> 3254, 3164, 3259, 3325, 3447, 3671, 3675, 3746, 3734, 3835, 3…
## $ m        <int> 3125, 3060, 3167, 3117, 3364, 3618, 3532, 3624, 3650, 3756, 3…
```

Filtrado de subconjuntos de filas 2

Por ejemplo el código 52835 corresponde a Tumaco.

```
bdj <- bd[cod_area == 52835]
glimpse(bdj)
```

```
## Rows: 509
## Columns: 6
## $ anio     <int> 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2…
## $ cod_area <int> 52835, 52835, 52835, 52835, 52835, 52835, 52835, 52835, 52835…
## $ area     <chr> "San Andres De Tumaco", "San Andres De Tumaco", "San Andres D…
## $ edad     <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ h        <int> 2799, 2249, 2325, 2173, 2051, 2314, 2159, 1995, 2154, 2052, 2…
## $ m        <int> 2515, 2133, 2177, 2087, 2045, 2187, 2111, 1900, 2036, 1907, 1…
```

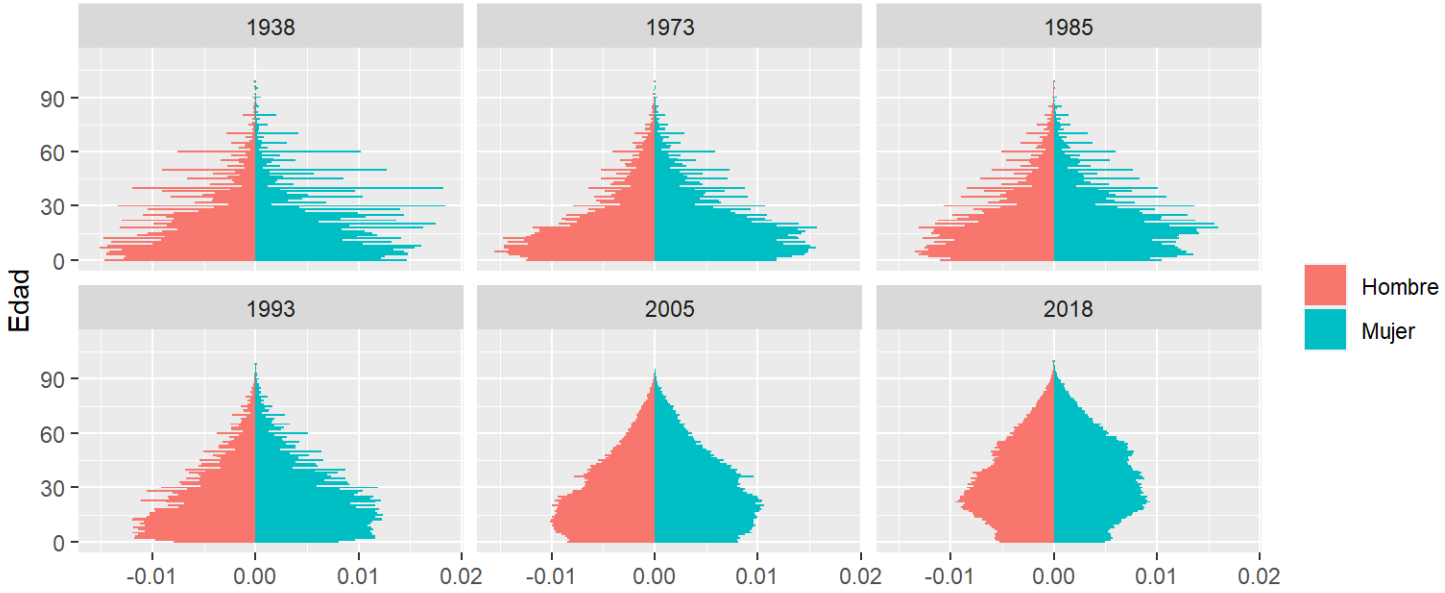
Cálculos de indicadores para pirámides

```
bdsb <- bds %>%
  .[, total := sum(h, na.rm = TRUE) + sum(m, na.rm = TRUE), keyby = .(anio)] %>%
  .[, `:=`(pct_h =  h / total, pct_m =  m / total ) ]
glimpse(bds)
```

```
## Rows: 607
## Columns: 9
## $ anio     <int> 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1…
## $ cod_area <int> 52001, 52001, 52001, 52001, 52001, 52001, 52001, 52001, 52001…
## $ area     <chr> "Pasto", "Pasto", "Pasto", "Pasto", "Pasto", "Pasto", "Pasto"…
## $ edad     <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ h        <int> 726, 634, 625, 716, 717, 706, 642, 750, 715, 494, 696, 453, 7…
## $ m        <int> 728, 606, 623, 731, 734, 715, 664, 767, 796, 517, 657, 419, 7…
## $ total    <int> 49644, 49644, 49644, 49644, 49644, 49644, 49644, 49644, 49644…
## $ pct_h    <dbl> 0.014624124, 0.012770929, 0.012589638, 0.014422690, 0.0144428…
## $ pct_m    <dbl> 0.014664411, 0.012206913, 0.012549351, 0.014724841, 0.0147852…
```

Pirámides poblacionales con ggplot - básica

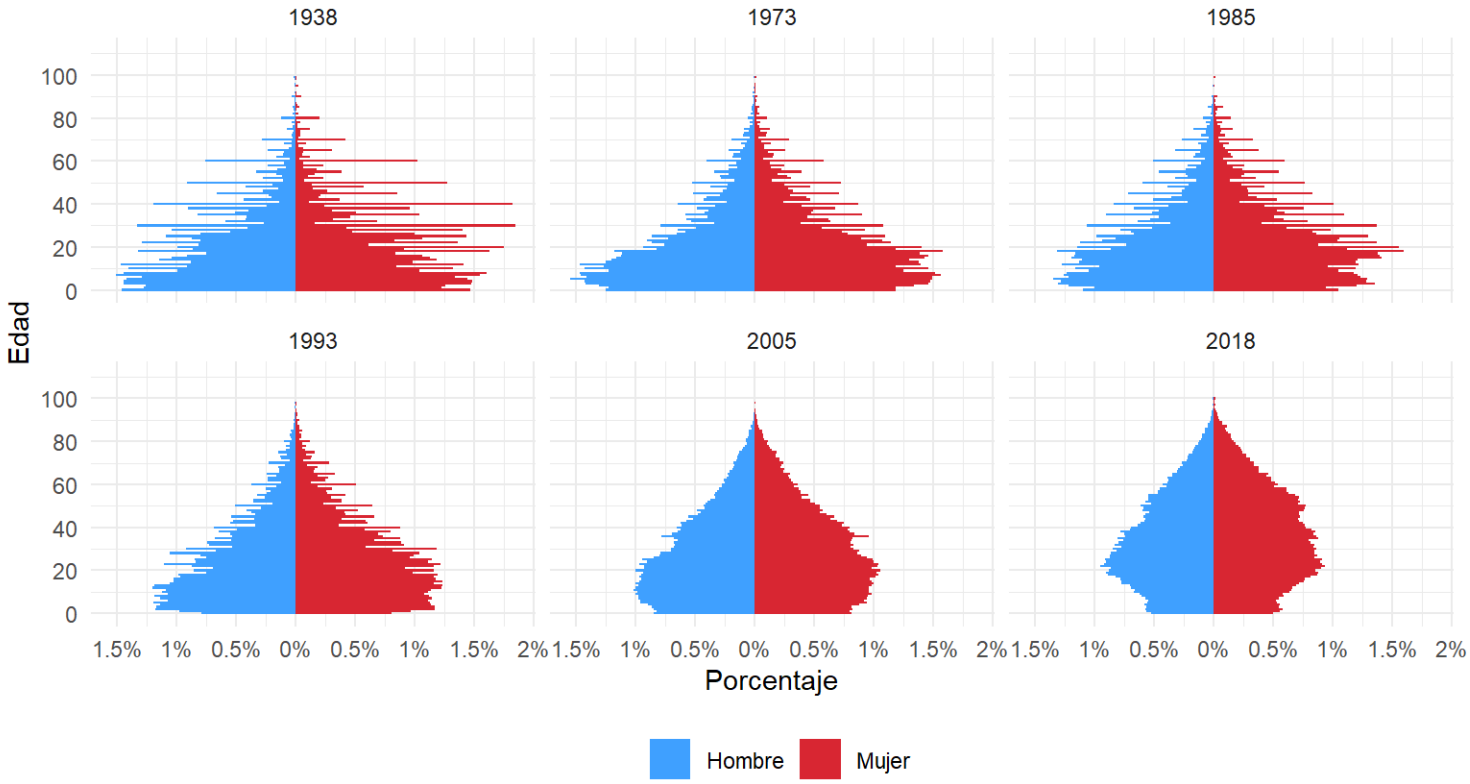
```
ggplot(bdsb) + facet_wrap(~anio) +
  geom_bar( aes(edad, -pct_h, fill = 'Hombre'), stat = 'identity', width = 1 ) +
  geom_bar( aes(edad,  pct_m, fill = 'Mujer'), stat = 'identity', width = 1 ) +
  coord_flip() + scale_fill_discrete(name = NULL) + xlab('Edad') + ylab('')
```



Pirámides poblacionales con ggplot - código

```
ggplot(bdsb) + facet_wrap(~anio) +
  geom_bar( aes(edad, -pct_h, fill = 'Hombre'), stat = 'identity', width = 1 ) +
  geom_bar( aes(edad,  pct_m, fill = 'Mujer'), stat = 'identity', width = 1 ) +
  coord_flip() +
  scale_fill_manual(name = NULL, values = c('#3FA0FF', '#D82632') ) +
  scale_y_continuous(name = 'Porcentaje', breaks = seq(-.02,.02, .005) ,
                     labels = paste0( abs(seq(-.02,.02, .005))*100 , '%') )+
  scale_x_continuous(name = 'Edad', breaks = seq(0,100,20) ) +
  theme_minimal()  + theme(legend.position = 'bottom')
```

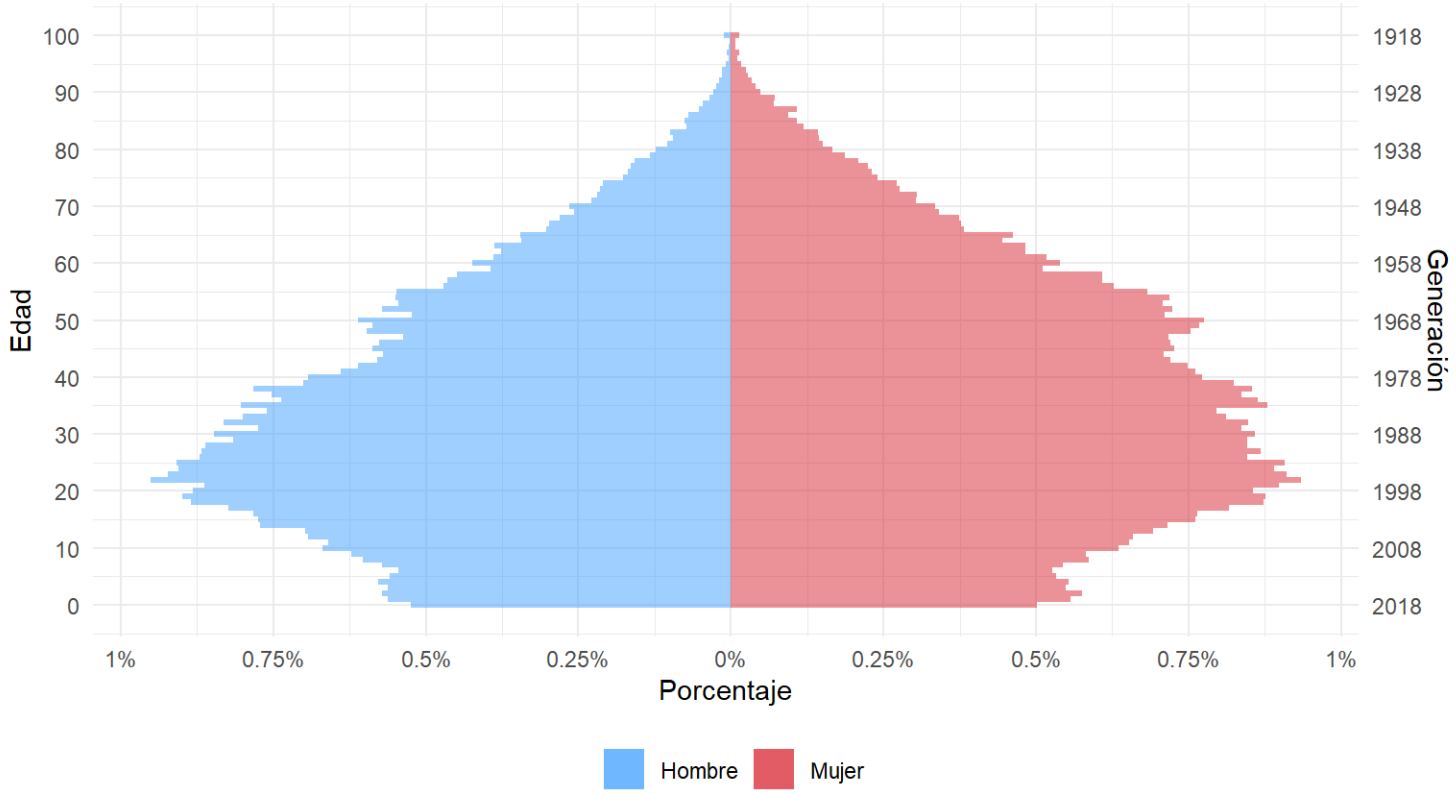
Pirámides poblacionales con ggplot - gráfica



Una pirámide con doble eje - código

```
ggplot(bdsb[anio == 2018]) +
  geom_bar( aes(edad, -pct_h, fill = 'Hombre'), stat = 'identity',
            width = 1, alpha = .5 ) +
  geom_bar( aes(edad,  pct_m, fill = 'Mujer'),  stat = 'identity',
            width = 1, alpha = .5 ) +
  coord_flip() +
  scale_fill_manual(name = NULL, values = c('#3FA0FF', '#D82632') ) +
  scale_y_continuous(name = 'Porcentaje', breaks = seq(-.02,.02, .0025) ,
                     labels = paste0( abs(seq(-.02,.02, .0025))*100 , '%') )+
  scale_x_continuous(name = 'Edad', breaks = seq(0, 100, 10),
                     sec.axis = sec_axis(name = 'Generación', ~ 2018 - . ,
                                         breaks = seq(2018, 1918, -10) )  ) +
  theme_minimal()  + theme(legend.position = 'bottom')
```

Una pirámide con doble eje - gráfico



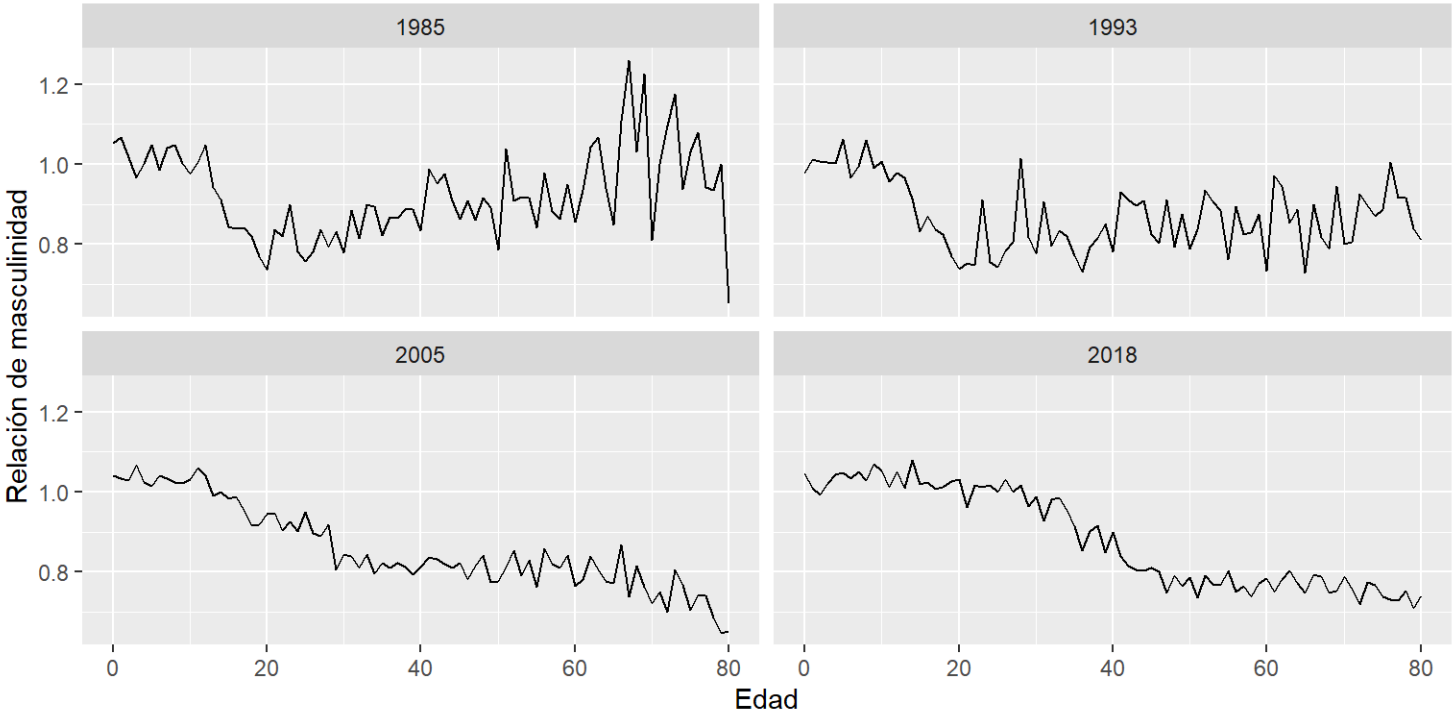
Relaciones de masculinidad por edad - cálculo

```
bdsc <- bdsb[, rm := h / m ]
glimpse(bdsc)
```

```
## Rows: 607
## Columns: 10
## $ anio     <int> 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1…
## $ cod_area <int> 52001, 52001, 52001, 52001, 52001, 52001, 52001, 52001, 52001…
## $ area     <chr> "Pasto", "Pasto", "Pasto", "Pasto", "Pasto", "Pasto", "Pasto"…
## $ edad     <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ h        <int> 726, 634, 625, 716, 717, 706, 642, 750, 715, 494, 696, 453, 7…
## $ m        <int> 728, 606, 623, 731, 734, 715, 664, 767, 796, 517, 657, 419, 7…
## $ total    <int> 49644, 49644, 49644, 49644, 49644, 49644, 49644, 49644, 49644…
## $ pct_h    <dbl> 0.014624124, 0.012770929, 0.012589638, 0.014422690, 0.0144428…
## $ pct_m    <dbl> 0.014664411, 0.012206913, 0.012549351, 0.014724841, 0.0147852…
## $ rm       <dbl> 0.9972527, 1.0462046, 1.0032103, 0.9794802, 0.9768392, 0.9874…
```

Relaciones de masculinidad por edad - gráfica

```
ggplot(bdsc[edad %in% 0:80 & anio %in% 1985:2018]) +
  geom_line(aes(edad,rm)) + facet_wrap(~anio) +
  ylab('Relación de masculinidad') + xlab('Edad')
```



Relaciones de masculinidad por grupos de edad - cálculo

```
bdscgre <- bdsb %>%
  .[, Edadgr5 := cut( edad ,  c( 0, seq(4, 90, by = 5), Inf) ,
                      labels = c('0 a 4', '5 a 9', '10 a 14', '15 a 19',
                                 '20 a 24', '25 a 29',  '30 a 34', '35 a 39',
                                 '40 a 44', '45 a 49','50 a 54', '55 a 59',
                                 '60 a 64', '65 a 69', '70 a 74', '75 a 79',
                                 '80 a 84', '85 a 89', '90 y más') ,
                      include.lowest = TRUE )  ]  %>%
  .[, .(h = sum(h,na.rm = TRUE), m = sum(m,na.rm = TRUE) ) ,
    keyby = .(anio, Edadgr5)] %>%
  .[, rm := h / m ]
```

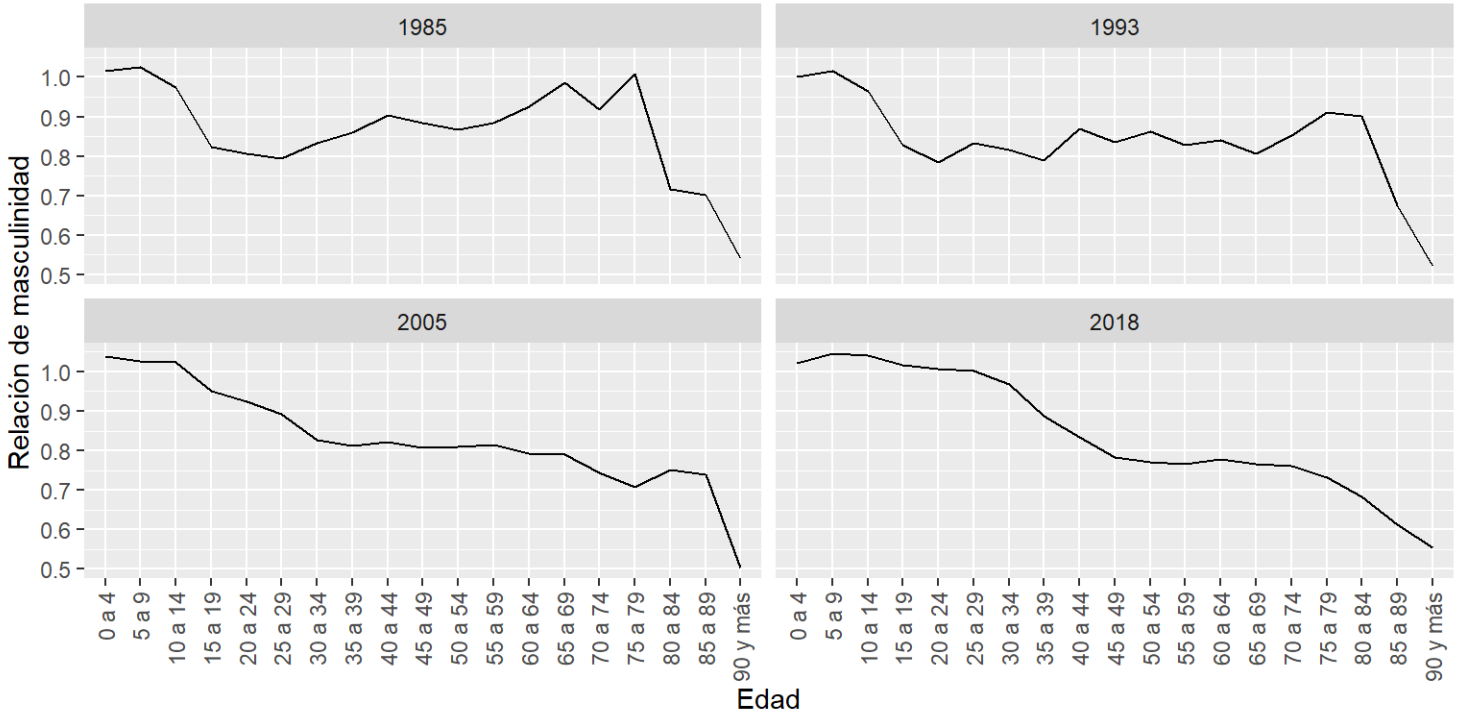
Relaciones de masculinidad por grupos de edad - resultados

```
glimpse(bdscgre)
```

```
## Rows: 114
## Columns: 5
## $ anio    <int> 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 19…
## $ Edadgr5 <fct> 0 a 4, 5 a 9, 10 a 14, 15 a 19, 20 a 24, 25 a 29, 30 a 34, 35 …
## $ h       <int> 3418, 3307, 2885, 2424, 2365, 1927, 1498, 1429, 1092, 844, 777…
## $ m       <int> 3422, 3459, 2850, 2761, 2789, 2349, 1716, 1513, 1341, 972, 914…
## $ rm      <dbl> 0.9988311, 0.9560567, 1.0122807, 0.8779428, 0.8479742, 0.82034…
```

Relaciones de masculinidad por grupos de edad

```
ggplot(bdscgre[anio %in% 1985:2018]) +   geom_line(aes(Edadgr5,rm, group = 1)) +
  facet_wrap(~anio) +   ylab('Relación de masculinidad') + xlab('Edad') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



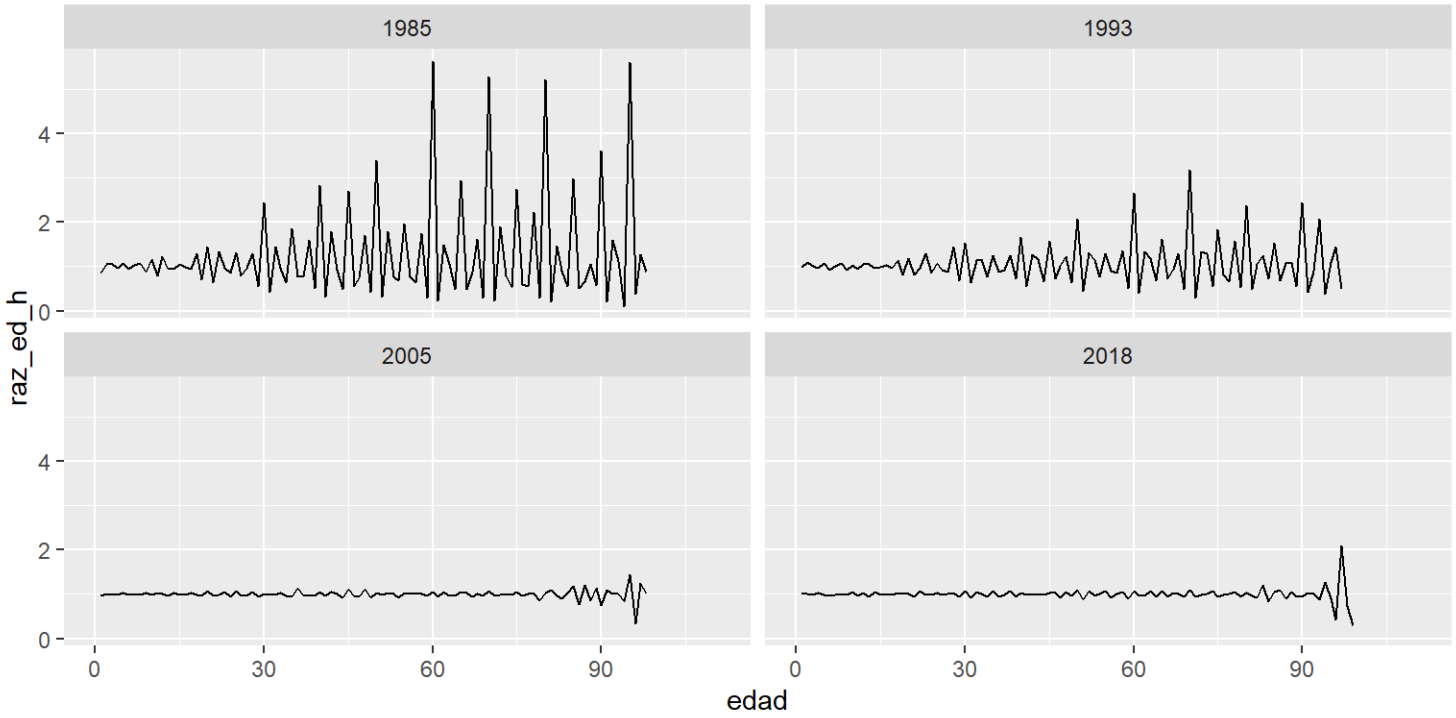
Relación de edades - cálculo

```
bdsd <- bdsc %>% .[ order(anio, edad), ] %>%
  .[, hlag  := shift(h, type = 'lag') , keyby =.(anio)  ] %>%
  .[, hlead := shift(h, type = 'lead') , keyby =.(anio)  ] %>%
  .[, mlag  := shift(m, type = 'lag') , keyby =.(anio)  ] %>%
  .[, mlead := shift(m, type = 'lead') , keyby =.(anio)  ] %>%
  .[, raz_ed_h := 2 * h / (hlag + hlead) ] %>%
  .[, raz_ed_m := 2 * m / (mlag + mlead) ] %>%
  .[,.(anio, edad, raz_ed_h, raz_ed_m)]
glimpse(bdsd)
```

```
## Rows: 607
## Columns: 4
## $ anio     <int> 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1938, 1…
## $ edad     <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ raz_ed_h <dbl> NA, 0.9385640, 0.9259259, 1.0670641, 1.0084388, 1.0389993, 0.…
## $ raz_ed_m <dbl> NA, 0.8971132, 0.9319372, 1.0773766, 1.0152144, 1.0228898, 0.…
```

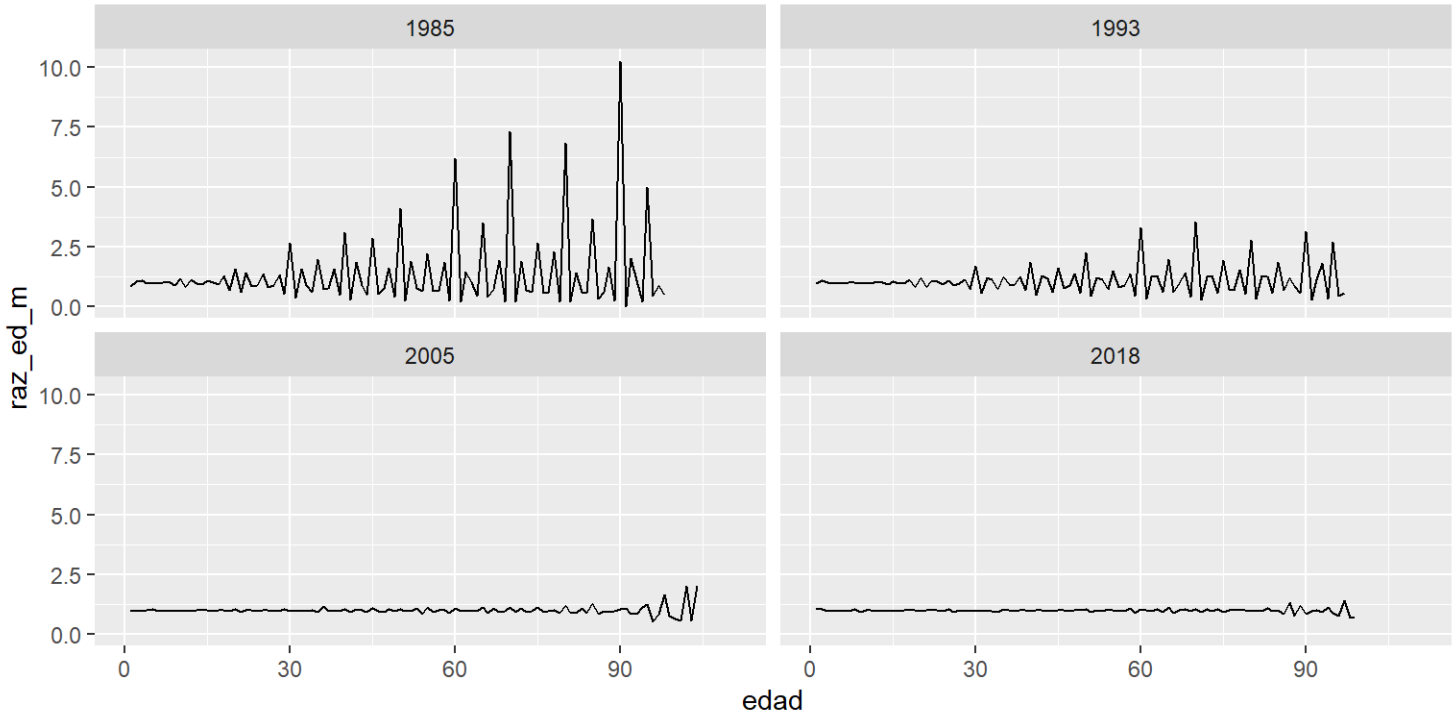
Relación de edades - gráfico hombres

```
p <- ggplot(bdsd[anio %in% 1985:2018 ]) + geom_line(aes(edad, raz_ed_h)) +
  facet_wrap(~anio)
p
```



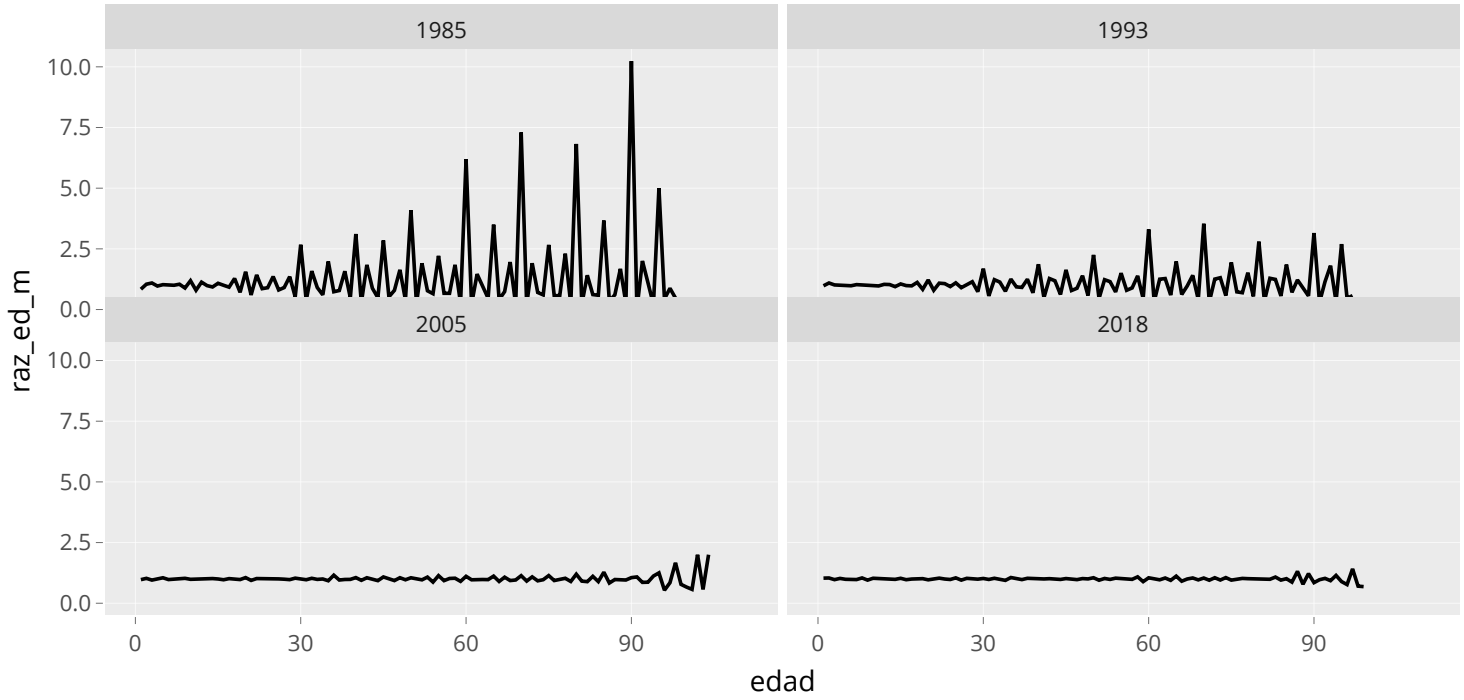
Relación de edades - gráfico mujeres

```
p <- ggplot(bdsd[anio %in% 1985:2018 ]) + geom_line(aes(edad, raz_ed_m)) +
  facet_wrap(~anio)
p
```



Un gráfico dinámico básico

```
# install.packages("plotly")
library(plotly)
ggplotly(p)
```



Diagramas triangulares

El paquete ggtern (www.ggtern.com) es adecuado para elaborar diagramas triangulares.

```
# install.packages('ggtern')
```

Como es usual se carga la libreria con la siguiente linea.

```
library('ggtern')
```

A continuación se debe organizar una matriz de datos para tres categorías de edad.

Diagrama triangular de edad - datos

```
bdgred <-  bds %>% .[, Edadgrg := cut( edad ,  c( 0, 14, 59, Inf) ,
                            labels = c('0 a 14', '15 a 59', '60 y más') ,
                            include.lowest = TRUE )  ] %>%
  .[, .(h = sum(h,na.rm = TRUE), m = sum(m,na.rm = TRUE) ), keyby = .(anio, Edadgrg)]
  glimpse(bdgred)
```

```
## Rows: 18
## Columns: 4
## $ anio    <int> 1938, 1938, 1938, 1973, 1973, 1973, 1985, 1985, 1985, 1993, 19…
## $ Edadgrg <fct> 0 a 14, 15 a 59, 60 y más, 0 a 14, 15 a 59, 60 y más, 0 a 14, …
## $ h       <int> 9610, 12804, 1224, 30061, 35158, 3729, 43159, 65439, 7144, 486…
## $ m       <int> 9731, 14811, 1464, 30517, 44080, 4234, 42902, 78239, 7815, 490…
```

Diagrama triangular de edad - porcentajes

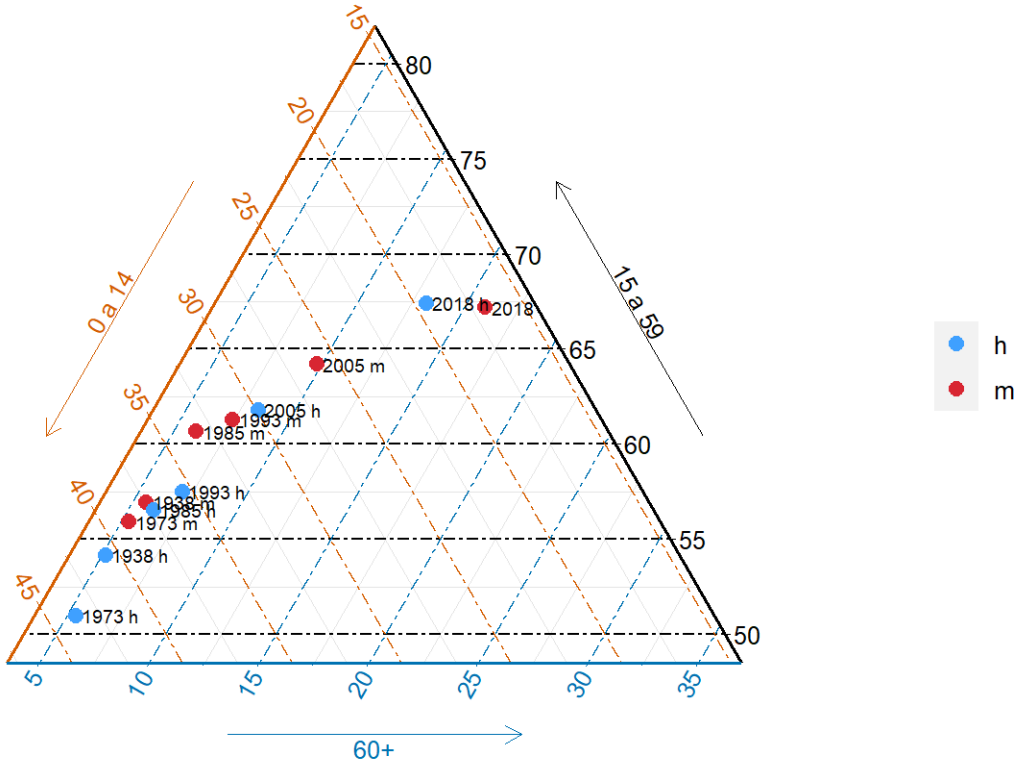
```
bdgrb <- melt(bdgred, id.vars = c('anio', 'Edadgrg') ) %>%
  .[, total := sum(value,na.rm = TRUE) , keyby = .(anio, variable)] %>%
  .[, pct := value / total ] %>%
  dcast(., anio + variable ~ Edadgrg, value.var = 'pct')
glimpse(bdgrb)
```

```
## Rows: 12
## Columns: 5
## $ anio       <int> 1938, 1938, 1973, 1973, 1985, 1985, 1993, 1993, 2005, 2005,…
## $ variable   <fct> h, m, h, m, h, m, h, m, h, m, h, m
## $ `0 a 14`   <dbl> 0.4065488, 0.3741829, 0.4359952, 0.3871193, 0.3728897, 0.33…
## $ `15 a 59`  <dbl> 0.5416702, 0.5695224, 0.5099205, 0.5591709, 0.5653868, 0.60…
## $ `60 y más` <dbl> 0.05178103, 0.05629470, 0.05408424, 0.05370983, 0.06172349,…
```

Diagrama triangular de edad - código

```
ggtern(bdgrb, aes(`0 a 14`, `15 a 59` , `60 y más` )) +
  theme_nomask() +
  theme_bvbw() +
  geom_point(aes(colour = variable) , size = 2.5) +
  geom_text(aes(label = paste(anio, variable)) , size = 2.5, hjust= -.1) +
  limit_tern(.82, .48, .37) +
  scale_colour_manual(name = '', values = c('#3FA0FF', '#D82632') ) +
  labs(x = '', y = '', z = '', xarrow = '0 a 14', yarrow = '15 a 59', zarrow = '60+')
```

Diagrama triangular de edad - gráfico

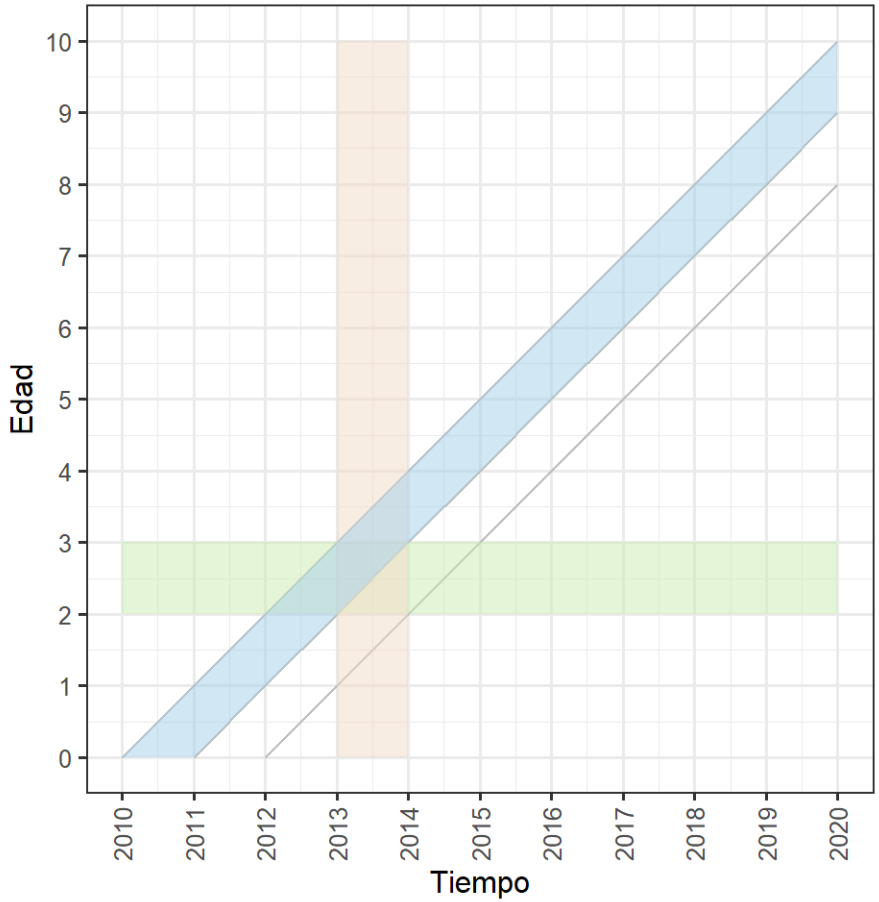


Un diagrama de lexis - código

```
Edad <- 0:10
Tiempo <- 2010:2020
pol <- data.table( y = c(0,0,10,9), x = c(2011,2010,2020, 2020), gr = rep(1,4)  )

ggplot() + geom_line( aes(x=Tiempo, y=Edad), colour = "gray")   +
  theme_bw() +  coord_fixed(ratio = 1) +
  scale_x_continuous(breaks = seq(2010,2020), limits =c(2010,2020)) +
  scale_y_continuous(breaks = seq(0,10), limits =c(0,10)) +
  geom_line( aes(x=Tiempo + 1, y=Edad), colour = "gray")  +
  geom_line( aes(x=Tiempo + 2, y=Edad), colour = "gray") +
  geom_rect(aes(xmin = 2010, xmax = 2020, ymin = 2, ymax = 3),
            fill = "#CCEDB1", alpha = .5 ) + # horizontal edad
  geom_rect(aes(xmin = 2013, xmax = 2014, ymin = 0, ymax = 10),
            fill = "#F2DBC8" , alpha = .5 ) + # vertical periodo +
  geom_polygon( aes(x = pol$x, y = pol$y, group = pol$gr),
                fill = '#A5CFE9', alpha = .5 ) +
   theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

Un diagrama de lexis - gráfico



Edad periodo y cohorte - código

```
Edad <- 0:10
Tiempo <- 2010:2020
edad_per <- data.table( y = c(4,4,5,5), x = c(2011, 2012, 2012, 2011), gr = rep(1,4)  )
per_coh <-  data.table( y = c(4,5,6,5), x = c(2014, 2015, 2015, 2014), gr = rep(1,4)  )
coh_edad <-  data.table( y = c(4,4,5,5), x = c(2017, 2018, 2019, 2018), gr = rep(1,4)  )

ggplot() +
  theme_bw() +  coord_fixed(ratio = 1) +
  scale_x_continuous(name = 'Periodo', breaks = seq(2010, 2020), limits =c(2010, 2020)) +
  scale_y_continuous(name = 'Edad', breaks = seq(0,10), limits =c(0,10)) +
  geom_polygon(aes(x = edad_per$x, y = edad_per$y, group = edad_per$gr),
            fill = "#CCEDB1", alpha = .5 ) +
  geom_polygon(aes(x = per_coh$x, y = per_coh$y, group = per_coh$gr),
            fill = "#F2DBC8", alpha = .5 ) +
  geom_polygon(aes(x = coh_edad$x, y = coh_edad$y, group = coh_edad$gr),
            fill = "#A5CFE9", alpha = .5 ) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

Edad periodo y cohorte - gráfico