Análisis Demográfico con R

Universidad de la República - Facultad de Ciencias Sociales - Doctorado en Ciencias Sociales

Ana Escoto

2024-07-29

Table of contents

Sobre el curso

1. Docente

Ana Ruth Escoto Castillo

Profesora de tiempo completo en la Facultad de Ciencias Políticas y Sociales, UNAM. Doctora en Estudios de Población por El Colegio de México y cuenta con nivel I en el Sistema Nacional de Investigadores.

2. Descripción del curso

La demografía utiliza diferentes fuentes de información para el análisis demográfico y los estudios de población. La consulta, la limpieza y la evaluación de los datos demográficos se realiza con distintos softwares, entre los cuales destaca R. Desde el software *R*, la comunidad de usuarios ha creado paquetes y códigos replicables y de fácil acceso que tienen un uso cada vez más extendido en la disciplina. En este curso se utilizarán estos insumos para el caso específico de América Latina y de Uruguay. Es decir, el objetivo general del curso es que el estudiantado sea capaz de aplicar conceptos demográficos y estadísticosm fuentes de información latinoamericana y mundiales, y sobre todo, actuales utilizando *R*.

Para ello, la mecánica del curso consistirá en lo siguiente:

1. La exposición de la facilitadora. Durante la primera parte de la sesión, se expondrán los comandos necesarios para trabajar cada tema. Se dará una introducción sobre la temática y se presentarán ejemplos concretos para facilitar el aprendizaje. Se espera que las personas asistentes expongan sus dudas o comentarios a lo largo de la explicación.

2. Realización de ejercicios prácticos. Al final de cada sesión, corresponderá al estudiantado realizar individualmente o en parejas un ejercicio relacionado con lo visto en la primera parte de la clase.

3. Consulta autónoma de material. Tanto la exposición como los ejercicios serán acompañados de material de consulta preparadopara el curso, de tal manera que el estudiantado pueda volver a los códigos y a las explicaciones posteriormente.

3. Carga horaria

15 horas

4. Créditos

3 (tres)

5. Estructura del curso

Día 1

1. Introducción a R y Rstudio (1 hora)

***Objetivo**: que el estudiantado se familiarice con la interfase de trabajo y la programación por objetos, y sea capaz de realizar tareas básicas como crear un script, un proyecto, objetos, ambientes e instalar paqueterías.*

2. Importación de información y primera revisión de fuentes demográficas (2 horas)

a. Importación de información a R en diferentes formatos

b. Importación de información de proyecciones de población utilizando {wppExplorer}

c. Consulta y descarga de información con paquetes como {IPUMSr}, {WDI} y otras API

***Objetivo:** que el estudiantado sea capaz de: importar información desde diferentes formatos (.txt, .csv, .xlsx, .dta, .dbf) a R, así como de exportar sus resultados en estos formatos; revisar de manera preliminar los objetos de tipo "data.frame", funciones "glimpse()", "skim() de {skimr}; manejar etiquetas; hacer subconjuntos de información, y consultas.*

Día 2

3. Evaluación de información (1.5 horas)

a. Tipo de de errores en las fuentes de información

b. Evaluación de la calidad de información en fuentes de stock

c. Suavizamiento de datos

***Objetivo:** Que el estudiantado pueda identificar los errores en el levantamiento de información y su naturaleza, adquiriendo capacidades para corregir y suavizar datos para el análisis estadístico con el paquete {DemoTools} y otras aplicaciones.*

4. Pirámides y diagramas de Lexis (1.5 horas)

a. Pirámides de población: crear una función

b. Hacer múltiples pirámides y automatización

c. Diagramas Lexis

Objetivo: *que el estudiantado sea capaz de crear y utilizar funciones específicas para el análisis demográfico, crear pirámides y la colocar eventos en el diagrama de Lexis*

Día 3

5. Crecimiento y tasas (3 horas)

a. Estandarización de tasas y gráficos de crecimiento, manejo de series de tiempo

b. Cálculos automatizados de población media

c. Cálculo de tasas de natalidad y mortalidad

d. Descomposición del cambio de tasas de natalidad y mortalidad según Kitagawa

Objetivo: *que el estudiantado sea capaz de calcular tasas brutas, tasas específicas y descomponerlas utilizando R.*

Día 4

6. Tasa de fecundidad con datos de encuestas (1.5 horas)

Objetivo: *que el estudiantado sea capaz de calcular tasas brutas y específicas de fecundidad con encuestas de hogares.*

7. Visualización de flujos migratorios (1.5 horas)

Objetivo: *que el estudiantado sea capaz de hacer gráficos de flujos con el paquete {migest} y gráficos aluviales.*

Día 5

8. Tablas de vida y esperanza de vida (3 horas)

a. Construcción de tabla de vida a "mano"

b. Construcción con {DemoTools}

Objetivo: *que el estudiantado sea capaz de calcular la tabla de vida con utilizando el paquete DemoTools*

6. Evaluación

· Entrega de un trabajo final que reúna lo trabajado en la instancia de práctica a lo largo de las cinco sesiones.

· La asistencia al 80% de las sesiones prácticas.

7. Bibliografía

El material guía construido por la facilitadora, que estará en este sitio web, será la bibliografía principal. Además se listan algunos insumos:

CEPAL, NU. 2014. "Los datos demográficos: alcances, limitaciones y métodos de evaluación".

Escoto, Ana. 2019. "Lexis en R". 2019.https://rstudio-pubs-static.s3.amazonaws.com/473169_a1348dd47070497a80fb2c0dc89e86e9.html.

Escoto Castillo, Ana Ruth. (2022) 2022. "aniuxa/paquetes_demogRaficos". R.https://github.com/aniuxa/paquetes_demogRaficos.

Moultrie, Tom, Rob Dorrington, Allan Hill, Kenneth Hill, Lan Timaeus, y Basia Zaba. 2013. *Tools for Demographic Estimation*. France: International Union for the Scientific Study of Population (IUSSP).

Poston, Dudley L., y Michael Micklin, eds. 2005. *Handbook of population*. Handbooks of sociology and social research. New York: Kluwer Academic/Plenum.

"PPgp/wpp2022". (2022) 2024. R. Probabilistic Projections Group.https://github.com/PPgp/wpp2022.

Pressat, Roland. 2000. *El análisis demográfico: métodos, resultados, aplicaciones*. Traducido por Tatiana Sule Hernández. México: Fondo de Cultura Económica.

Preston, Samuel H., Patrick Heuveline, y Michel Guillot. 2001. *Demography: measuring and modeling population processes*. Malden, MA: Blackwell Publishers.

Pujol, José Miguel. 1985. "Nuevas metodologías para evaluar y ajustar datos demográficos", diciembre.https://repositorio.cepal.org/handle/11362/12578.

Riffe, Tim. (2017) 2024. "timriffe/DemoTools". R.https://github.com/timriffe/DemoTools.

Rodríguez, Germán. s/f. "Demographic Methods".https://grodri.github.io/demography/.

Sevcikova, Hana, Adrian Raftery, y Thomas Buettner. 2023. "bayesPop: Probabilistic Population Projection".https://cran.r-project.org/web/packages/bayesPop/index.html.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the Tidyverse". *Journal of Open Source Software* 4 (43): 1686.https://doi.org/10.21105/joss.01686.

Wickham, Hadley, y Garrett Grolemund. 2016. *R for data science: import, tidy, transform, visualize, and model data.* O'Reilly Media, Inc.

Instalación de R y Rstudio

Introducción a R

https://youtu.be/YkN5urybh2A

Instalación en OS

1. Necesito que instalen la versión más nueva de R: Download R-4.4.0 of MAC. *The R-project for statistical computing.* https://cran.r-project.org/bin/macosx/

Elije la versión de acuerdo a tu procesador, intel o ARM.

2. Instalar también las herramientas Quartz, xcode y fortran

- https://www.xquartz.org/
- https://developer.apple.com/xcode/resources/
- https://mac.r-project.org/tools/gfortran-12.2-universal.pkg

3. Después de eso instalar el Rstudio, que hoy se encuentra alojado en el sitio posit, que vaya acorde con MAC

https://posit.co/download/rstudio-desktop/

Algunas indicaciones en video, pero son algo viejitas y pueden cambiar las versiones de R.

https://youtu.be/icWV8jzYOtA

Algunas indicaciones en video, pero son algo viejitas y pueden cambiar las versiones de R.

Instalación en PC

1. Necesito que instalen la versión más nueva de R: Download R-4.4.0 for Windows. *The R-project for statistical computing.* https://cran.r-project.org/bin/windows/base/
2. Instalar también la herramienta RTools https://cran.r-project.org/bin/windows/Rtools/rtools44/rtools.html
3. Después de eso instalar el Rstudio, que hoy se encuentra alojado en el sitio posit, que vaya acorde con Windows https://posit.co/download/rstudio-desktop/

Algunas indicaciones en video, pero son algo viejitas y pueden cambiar las versiones de R.

https://youtu.be/TNSQikMfgJI

Ojo

Desde octubre de 2022, RStudio se volvió **"Posit"**

1 Introducción a R y Rstudio

1.1 Primer acercamiento al uso del programa

Usaremos la IDE RStudio — pronto habrá *positron*

En RStudio de *posit* podemos tener varias ventanas que nos permiten tener más control de nuestro "ambiente", el historial, los *scripts¨ o códigos que escribimos y por supuesto, tenemos nuestra consola, que también tiene el símbolo `>`

Podemos pedir operaciones básicas

```
2+5
```

```
[1] 7
```

```
5*3
```

```
[1] 15
```

```
#Para escribir comentarios y que no los lea como operaciones ponemos el símbolo de gato
## Lo podemos hacer para un comentario en una línea o la par de una instrucción
1:5              ## Secuencia 1-5
```

```
[1] 1 2 3 4 5
```

```
seq(1, 10, 0.5)   ## Secuencia con incrementos diferentes a 1
```

```
 [1]  1.0  1.5  2.0  2.5  3.0  3.5  4.0  4.5  5.0  5.5  6.0  6.5  7.0  7.5  8.0
[16]  8.5  9.0  9.5 10.0
```

```
c('a','b','c')  ## Vector con caracteres
```

```
[1] "a" "b" "c"
```

```r
1:7              ## Entero
```

```
[1] 1 2 3 4 5 6 7
```

```r
40<80            ## Valor logico
```

```
[1] TRUE
```

```r
2+2 == 5         ## Valor logico
```

```
[1] FALSE
```

```r
T == TRUE        ## T expresion corta de verdadero
```

```
[1] TRUE
```

R es un lenguaje de programación por objetos. Por lo cual vamos a tener objetos a los que se les asigna su contenido. Si usamos una flechita `<-` o `->` le estamos asignando algo al objeto que apunta la flecha.

```r
x <- 24          ## Asignacion de valor 24 a la variable x para su uso posterior (OBJETO)
x/2              ## Uso posterior de variable u objeto x
```

```
[1] 12
```

```r
x                ## Imprime en pantalla el valor de la variable u objeto
```

```
[1] 24
```

```r
x <- TRUE        ## Asigna el valor logico TRUE a la variable x OJO: x toma el ultimo valor
x
```

```
[1] TRUE
```

1.1.1 Vectores

Los vectores son uno de los objetos más usados en R.

```r
y <- c( 2, 4, 6)      ## Vector numerico
y <- c('Primaria', 'Secundaria') ## Vector caracteres
```

Dado que poseen elementos, podemos también observar y hacer operaciones con sus elementos, usando `[ ]` para acceder a ellos

```r
y[2]               ## Acceder al segundo valor del vector y
```

```
[1] "Secundaria"
```

```r
y[3] <- 'Preparatoria y más' ## Asigna valor a la tercera componente del vector
sex <-1:2          ## Asigna a la variable sex los valores 1 y 2
names(sex) <- c("Femenino", "Masculino") ## Asigna nombres al vector de elementos sexo
sex[2]             ## Segundo elemento del vector sex
```

```
Masculino
        2
```

1.1.2 Funciones

Algunas funciones básicas son las siguientes. Vamos a ir viendo más funciones, pero para entender cómo *funcionan*, haremos unos ejemplos y cómo pedir ayuda sobre ellas.

```r
sum( 10, 20, 30)     ## Función suma
```

```
[1] 60
```

```r
rep( 'R',  times=3) ## Repite la letra R el numero de veces que se indica
```

```
[1] "R" "R" "R"
```

```r
sqrt(9)            ## Raiz cuadrada de 9
```

```
[1] 3
```

1.1.3 Indentación

En otros paquetes la indentación es muy importante (i.e. Python). En R no es necesario

```r
sum( 10, 20, 30)
```

```
[1] 60
```

```r
sum(10,
    20,
    30)
```

```
[1] 60
```

1.1.4 Ayuda

Pedir ayuda es indispensable para aprender a escribir nuestros códigos. A prueba y error, es el mejor sistema para aprender. Podemos usar la función help, example y ?

```r
help(sum)     ## Ayuda sobre función sum
?sum()        ## ídem
example(sum) ## Ejemplo de función sum
```

```
sum> ## Pass a vector to sum, and it will add the elements together.
sum> sum(1:5)
[1] 15

sum> ## Pass several numbers to sum, and it also adds the elements.
sum> sum(1, 2, 3, 4, 5)
[1] 15

sum> ## In fact, you can pass vectors into several arguments, and everything gets added.
sum> sum(1:2, 3:5)
[1] 15

sum> ## If there are missing values, the sum is unknown, i.e., also missing, ....
sum> sum(1:5, NA)
[1] NA
```

```
sum> ## ... unless  we exclude missing values explicitly:
sum> sum(1:5, NA, na.rm = TRUE)
[1] 15
```

1.1.5 Mi ambiente

Todos los objetos que hemos declarado hasta ahora son parte de nuestro “ambiente” (environment). Para saber qué está en nuestro ambiente usamos el comando

```
ls()
```

```
[1] "has_annotations" "pandoc_dir"      "quarto_bin_path" "sex"
[5] "x"               "y"
```

```
gc()           ## Garbage collection, reporta memoria en uso
```

```
         used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
Ncells  628442 33.6    1354192 72.4         NA  1354192 72.4
Vcells 1176971  9.0    8388608 64.0      16384  1962707 15.0
```

Para borrar todos nuestros objetos, usamos el siguiente comando, que equivale a usar la escobita de la venta de environment

```
rm(list=ls())  ## Borrar objetos actuales
```

1.2 Directorio de trabajo

Es muy útil saber dónde estamos trabajando y donde queremos trabajar. Por eso podemos utilizar los siguientes comandos para saberlo

Ojo, checa, si estás desdes una PC, cómo cambian las “ ” por”/” o por “\”

```
getwd()           # Directorio actual
```

```
[1] "/Users/anaescoto/Dropbox/2024/R_UY/r_demo_uy"
```

```
list.files()      # Lista de archivos en ese directorio
```

```
 [1] "LICENSE"                  "Mi_Exportación.xlsx"     "MiprimerAmbiente.RData"
 [4] "P1.html"                  "P1.qmd"                  "P1.rmarkdown"
 [7] "P1_files"                 "P2.qmd"                  "P3.qmd"
[10] "P4.qmd"                   "P5.qmd"                  "README.md"
[13] "_quarto.yml"              "códigos"                 "datos"
[16] "docs"                     "fecundidad.R"            "index.html"
[19] "index.qmd"                "instala.html"            "instala.qmd"
[22] "ipums.R"                  "ipumsi_00016.R"          "ipumsi_00016.dat.gz"
[25] "ipumsi_00016.xml"         "otros.qmd"               "paquetes"
[28] "pira1830.png"             "pira222.png"             "pira320.png"
[31] "pira340.png"              "pira484.png"             "pira858.png"
[34] "r_demo_uy.Rproj"          "r_demo_uy2"              "site_libs"
```

1.3 Proyectos

Pero… a veces preferimos trabajar en proyectos, sobre todo porque nos da más control.

Hay gente que lo dice mejor que yo, como Hadley Wickham: https://es.r4ds.hadley.nz/08-workflow-projects.html

Hagamos un proyecto. Este proyecto debe tener **adentro** una carpeta que se llame datos.

Descarga algunos de los datos que usaremos en el curso acá

1.4 Instalación de paquetes

Los paquetes son útiles para realizar funciones especiales. La especialización de paquetes es más rápida en R que en otros programas por ser un software libre.

```
#install.packages("foreign", dependencies = TRUE)
#install.packages("haven", dependencies = TRUE)
```

Este proceso no hay que hacerlo siempre. Si no sólo la primera vez. Una vez instalado un paquete, lo llamamos con el comando `library()`

```
library(foreign)
library(haven)
```

`{foreing}` nos permite leer archivos en formato de dBase, con extensión `.dbf`. Si bien no es un formato muy común para los investigadores, sí para los que generan la información, puesto que dBase es uno de los principales programas de administración de bases de datos.

He puesto un ejemplo de una base de datos mexicana en dbf, en este formato.

```
ejemplo_dbf<-foreign::read.dbf("datos/ejemplo_dbf.DBF") #checa cómo nos vamos adentro de n
```

1.5 Paquete {pacman}

En general, cuando hacemos nuestro código querremos verificar que nuestras librerías estén instaladas. Si actualizamos nuestro R y Rstudio es probable (sobre todo en MAC) que hayamos perdido alguno.

Este es un ejemplo de un código. Y vamos a introducir un paquete muy útil llamado "pacman"

```
if (!require("pacman")) install.packages("pacman") # instala pacman si se requiere
```

```
Cargando paquete requerido: pacman
```

```
pacman::p_load(tidyverse,
               readxl,
               writexl,
               haven,
               sjlabelled,
               foreign,
               WDI,
               remotes)
```

1.6 Instalación de paquetes en desarrollo

Además de los paquetes que están en CRAN, hay otros repositorios desde los cuáles podemos instalar el código. Un paquete que utilizaremos mucho, es el paquete `{wpp2022}`

```
remotes::install_github("PPgp/wpp2022")
```

```
Skipping install of 'wpp2022' from a github remote, the SHA1 (a45518ac) has not changed since
  Use `force = TRUE` to force installation
```

1.7 Dataframes con el paquete {WDI}

Instalamos anteriormente el paquete `{WDI}` que nos da acceso a un grupo amplio de bases de datos que nos ayudaran a revisar y analizar algunas técnicas sencillas.

> El Banco Mundial pone a disposición una gran cantidad de datos excelentes de los Indicadores de Desarrollo Mundial a través de su API web. El paquete WDI para R facilita la búsqueda y descarga de series de datos desde WDI".

Para saber un poco más de esta librería:

- https://cran.r-project.org/web/packages/WDI/WDI.pdf
- https://www.r-project.org/nosvn/pandoc/WDI.html
- https://databank.worldbank.org/reports.aspx?source=2&country=ARE

```
WDI::WDIsearch('gender')
```

```
                                     indicator
169                                2.3_GIR.GPI
172                                2.6_PCR.GPI
709                           5.51.01.07.gender
1573                       BI.EMP.PWRK.PB.FE.ZS
1575                       BI.EMP.PWRK.PB.MA.ZS
1587                       BI.EMP.TOTL.PB.FE.ZS
1589                       BI.EMP.TOTL.PB.MA.ZS
1712                          BI.WAG.PREM.PB.FE
1716                          BI.WAG.PREM.PB.FM
1717                       BI.WAG.PREM.PB.FM.ED
1718                       BI.WAG.PREM.PB.FM.HE
1719                       BI.WAG.PREM.PB.FM.PA
1723                          BI.WAG.PREM.PB.MA
1735                       BI.WAG.PREM.PV.FM.ED
1736                       BI.WAG.PREM.PV.FM.HE
1737                          BI.WAG.PRVS.ED.FM
1740                          BI.WAG.PRVS.HE.FM
1744                          BI.WAG.PUBS.ED.FM
1747                          BI.WAG.PUBS.HE.FM
1748                          BI.WAG.PUBS.PA.FM
2202                               CC.ESG.AGFE
2203                               CC.ESG.AGMA
2204                               CC.ESG.CMFE
```

2205	CC.ESG.CMMA
2206	CC.ESG.CNFE
2207	CC.ESG.CNMA
2208	CC.ESG.EUFE
2209	CC.ESG.EUMA
2210	CC.ESG.FBFE
2211	CC.ESG.FBMA
2212	CC.ESG.INFE
2213	CC.ESG.INMA
2214	CC.ESG.MAFE
2215	CC.ESG.MAMA
2216	CC.ESG.MIFE
2217	CC.ESG.MIMA
2218	CC.ESG.OSFE
2219	CC.ESG.OSMA
2220	CC.ESG.PAFE
2221	CC.ESG.PAMA
2222	CC.ESG.PSFE
2223	CC.ESG.PSMA
2224	CC.ESG.SEFE
2225	CC.ESG.SEMA
2226	CC.ESG.TCFE
2227	CC.ESG.TCMA
2296	CC.ISG.FFFE
2297	CC.ISG.FFMA
2298	CC.ISG.NAFE
2299	CC.ISG.NAMA
2300	CC.ISG.NBFE
2301	CC.ISG.NBMA
6251	FB.FCP.BREG.PR.DI.SC
8632	IC.REG.PRRT.LNDADM.GEN.XD.030.DB1719.DFRN
8904	IQ.CPA.GNDR.XQ
9755	JI.WAG.GNDR
9756	JI.WAG.GNDR.HE
9757	JI.WAG.GNDR.LE
9758	JI.WAG.GNDR.OL
9759	JI.WAG.GNDR.RU
9760	JI.WAG.GNDR.UR
9761	JI.WAG.GNDR.YG
14651	PRJ.MYS.15UP.GPI
14667	PRJ.MYS.25UP.GPI
15164	SE.ADT.1524.LT.FM.ZS
15175	SE.ENR.PRIM.FM.ZS

15177	SE.ENR.PRSC.FM.ZS
15178	SE.ENR.SECO.FM.ZS
15180	SE.ENR.TERT.FM.ZS
15998	SG.LAW.CRDD.GR
16011	SG.LAW.NODC.HR
16027	SG.NOD.CONS
17766	SPI.D3.5.GEND
18119	UIS.AIR.1.GLAST.GPIA
18120	UIS.AIR.2.GPV.GLAST.GPIA
18129	UIS.CR.1.GPIA
18137	UIS.CR.1.Q1.GPIA
18144	UIS.CR.1.Q2.GPIA
18151	UIS.CR.1.Q3.GPIA
18158	UIS.CR.1.Q4.GPIA
18165	UIS.CR.1.Q5.GPIA
18172	UIS.CR.1.RUR.GPIA
18177	UIS.CR.1.RUR.Q1.GPIA
18181	UIS.CR.1.RUR.Q2.GPIA
18185	UIS.CR.1.RUR.Q3.GPIA
18189	UIS.CR.1.RUR.Q4.GPIA
18193	UIS.CR.1.RUR.Q5.GPIA
18199	UIS.CR.1.URB.GPIA
18204	UIS.CR.1.URB.Q1.GPIA
18208	UIS.CR.1.URB.Q2.GPIA
18212	UIS.CR.1.URB.Q3.GPIA
18216	UIS.CR.1.URB.Q4.GPIA
18220	UIS.CR.1.URB.Q5.GPIA
18228	UIS.CR.2.GPIA
18236	UIS.CR.2.Q1.GPIA
18243	UIS.CR.2.Q2.GPIA
18250	UIS.CR.2.Q3.GPIA
18257	UIS.CR.2.Q4.GPIA
18264	UIS.CR.2.Q5.GPIA
18271	UIS.CR.2.RUR.GPIA
18276	UIS.CR.2.RUR.Q1.GPIA
18280	UIS.CR.2.RUR.Q2.GPIA
18284	UIS.CR.2.RUR.Q3.GPIA
18288	UIS.CR.2.RUR.Q4.GPIA
18292	UIS.CR.2.RUR.Q5.GPIA
18298	UIS.CR.2.URB.GPIA
18303	UIS.CR.2.URB.Q1.GPIA
18307	UIS.CR.2.URB.Q2.GPIA
18311	UIS.CR.2.URB.Q3.GPIA

18315	UIS.CR.2.URB.Q4.GPIA
18319	UIS.CR.2.URB.Q5.GPIA
18327	UIS.CR.3.GPIA
18335	UIS.CR.3.Q1.GPIA
18342	UIS.CR.3.Q2.GPIA
18349	UIS.CR.3.Q3.GPIA
18356	UIS.CR.3.Q4.GPIA
18363	UIS.CR.3.Q5.GPIA
18370	UIS.CR.3.RUR.GPIA
18375	UIS.CR.3.RUR.Q1.GPIA
18379	UIS.CR.3.RUR.Q2.GPIA
18383	UIS.CR.3.RUR.Q3.GPIA
18387	UIS.CR.3.RUR.Q4.GPIA
18391	UIS.CR.3.RUR.Q5.GPIA
18397	UIS.CR.3.URB.GPIA
18402	UIS.CR.3.URB.Q1.GPIA
18406	UIS.CR.3.URB.Q2.GPIA
18410	UIS.CR.3.URB.Q3.GPIA
18414	UIS.CR.3.URB.Q4.GPIA
18418	UIS.CR.3.URB.Q5.GPIA
18459	UIS.EA.1T8.AG25T99.GPIA
18466	UIS.EA.2T8.AG25T99.GPIA
18473	UIS.EA.3T8.AG25T99.GPIA
18480	UIS.EA.4T8.AG25T99.GPIA
18486	UIS.EA.5T8.AG25T99.GPIA
18493	UIS.EA.6T8.AG25T99.GPIA
18500	UIS.EA.7T8.AG25T99.GPIA
18504	UIS.EA.8.AG25T99.GPIA
18517	UIS.EA.S1T8.AG25T99.GPIA
18525	UIS.ESG.LOWERSEC.COGN.GPIA
18528	UIS.ESG.LOWERSEC.GPIA
18532	UIS.ESG.LOWERSEC.NCOG.CONF.GPI
18536	UIS.ESG.LOWERSEC.NCOG.ENJO.GPI
18540	UIS.EV1524P.2T5.V.GPIA
18550	UIS.FHLANGILP.1.GPIA
18569	UIS.GAR.5T8.GPIA
18577	UIS.GAR.5T8.Q1.GPIA
18584	UIS.GAR.5T8.Q2.GPIA
18591	UIS.GAR.5T8.Q3.GPIA
18598	UIS.GAR.5T8.Q4.GPIA
18605	UIS.GAR.5T8.Q5.GPIA
18612	UIS.GAR.5T8.RUR.GPIA
18617	UIS.GAR.5T8.RUR.Q1.GPIA

18621	UIS.GAR.5T8.RUR.Q2.GPIA
18625	UIS.GAR.5T8.RUR.Q3.GPIA
18629	UIS.GAR.5T8.RUR.Q4.GPIA
18633	UIS.GAR.5T8.RUR.Q5.GPIA
18639	UIS.GAR.5T8.URB.GPIA
18644	UIS.GAR.5T8.URB.Q1.GPIA
18648	UIS.GAR.5T8.URB.Q2.GPIA
18652	UIS.GAR.5T8.URB.Q3.GPIA
18656	UIS.GAR.5T8.URB.Q4.GPIA
18660	UIS.GAR.5T8.URB.Q5.GPIA
18667	UIS.GCS.LOWERSEC.COG.GPIA
18670	UIS.GCS.LOWERSEC.GPIA
18674	UIS.GCS.LOWERSEC.NCOG.FREE.GPI
18676	UIS.GCS.LOWERSEC.NCOG.GEQU
18677	UIS.GCS.LOWERSEC.NCOG.GEQU.F
18678	UIS.GCS.LOWERSEC.NCOG.GEQU.GPI
18679	UIS.GCS.LOWERSEC.NCOG.GEQU.M
18682	UIS.GCS.LOWERSEC.NCOG.GLOC.GPI
18686	UIS.GCS.LOWERSEC.NCOG.MULT.GPI
18690	UIS.GCS.LOWERSEC.NCOG.PEAC.GPI
18694	UIS.GCS.LOWERSEC.NCOG.SDEV.GPI
18698	UIS.GCS.LOWERSEC.NCOG.SJUS.GPI
18702	UIS.GER.0.GPIA
18706	UIS.GER.01.GPIA
18708	UIS.GER.02.GPIA
18711	UIS.GER.12.GPI
18717	UIS.GER.1T6.GPI
18719	UIS.GER.2.GPI
18720	UIS.GER.3.GPI
18723	UIS.GER.4.GPI
18725	UIS.GER.5T8.GPIA
18726	UIS.GGR.5.A.GPI
18742	UIS.ICTSKILLATTACH.GPIA
18746	UIS.ICTSKILLCONNEC.GPIA
18750	UIS.ICTSKILLCOPI.GPIA
18754	UIS.ICTSKILLCREAT.GPIA
18758	UIS.ICTSKILLDUPLIC.GPIA
18762	UIS.ICTSKILLFORMULA.GPIA
18766	UIS.ICTSKILLPROGLANG.GPIA
18770	UIS.ICTSKILLSOFTWARE.GPIA
18774	UIS.ICTSKILLTRANSFERFILE.GPIA
18793	UIS.LR.AG15T24.GPIA
18798	UIS.LR.AG15T24.RUR.GPIA

18802	UIS.LR.AG15T24.URB.GPIA
18805	UIS.LR.AG15T99.GPIA
18810	UIS.LR.AG15T99.RUR.GPIA
18814	UIS.LR.AG15T99.URB.GPIA
18819	UIS.LR.AG25T64.GPIA
18825	UIS.LR.AG25T64.RUR.GPIA
18829	UIS.LR.AG25T64.URB.GPIA
18835	UIS.LR.AG65T99.GPIA
18840	UIS.LR.AG65T99.RUR.GPIA
18844	UIS.LR.AG65T99.URB.GPIA
18848	UIS.MATH.G2T3.GPIA
18864	UIS.MATH.LOWERSEC.GPIA
18880	UIS.MATH.PRIMARY.GPIA
18907	UIS.NARA.AGM1.GPIA
18915	UIS.NARA.AGM1.Q1.GPIA
18922	UIS.NARA.AGM1.Q2.GPIA
18929	UIS.NARA.AGM1.Q3.GPIA
18936	UIS.NARA.AGM1.Q4.GPIA
18943	UIS.NARA.AGM1.Q5.GPIA
18950	UIS.NARA.AGM1.RUR.GPIA
18955	UIS.NARA.AGM1.RUR.Q1.GPIA
18959	UIS.NARA.AGM1.RUR.Q2.GPIA
18963	UIS.NARA.AGM1.RUR.Q3.GPIA
18967	UIS.NARA.AGM1.RUR.Q4.GPIA
18971	UIS.NARA.AGM1.RUR.Q5.GPIA
18977	UIS.NARA.AGM1.URB.GPIA
18982	UIS.NARA.AGM1.URB.Q1.GPIA
18986	UIS.NARA.AGM1.URB.Q2.GPIA
18990	UIS.NARA.AGM1.URB.Q3.GPIA
18994	UIS.NARA.AGM1.URB.Q4.GPIA
18998	UIS.NARA.AGM1.URB.Q5.GPIA
19006	UIS.NART.1.GPIA
19014	UIS.NART.1.Q1.GPIA
19021	UIS.NART.1.Q2.GPIA
19028	UIS.NART.1.Q3.GPIA
19035	UIS.NART.1.Q4.GPIA
19042	UIS.NART.1.Q5.GPIA
19049	UIS.NART.1.RUR.GPIA
19054	UIS.NART.1.RUR.Q1.GPIA
19058	UIS.NART.1.RUR.Q2.GPIA
19062	UIS.NART.1.RUR.Q3.GPIA
19066	UIS.NART.1.RUR.Q4.GPIA
19070	UIS.NART.1.RUR.Q5.GPIA

19076	UIS.NART.1.URB.GPIA
19081	UIS.NART.1.URB.Q1.GPIA
19085	UIS.NART.1.URB.Q2.GPIA
19089	UIS.NART.1.URB.Q3.GPIA
19093	UIS.NART.1.URB.Q4.GPIA
19097	UIS.NART.1.URB.Q5.GPIA
19105	UIS.NART.2.GPIA
19113	UIS.NART.2.Q1.GPIA
19120	UIS.NART.2.Q2.GPIA
19127	UIS.NART.2.Q3.GPIA
19134	UIS.NART.2.Q4.GPIA
19141	UIS.NART.2.Q5.GPIA
19148	UIS.NART.2.RUR.GPIA
19153	UIS.NART.2.RUR.Q1.GPIA
19157	UIS.NART.2.RUR.Q2.GPIA
19161	UIS.NART.2.RUR.Q3.GPIA
19165	UIS.NART.2.RUR.Q4.GPIA
19169	UIS.NART.2.RUR.Q5.GPIA
19175	UIS.NART.2.URB.GPIA
19180	UIS.NART.2.URB.Q1.GPIA
19184	UIS.NART.2.URB.Q2.GPIA
19188	UIS.NART.2.URB.Q3.GPIA
19192	UIS.NART.2.URB.Q4.GPIA
19196	UIS.NART.2.URB.Q5.GPIA
19204	UIS.NART.3.GPIA
19212	UIS.NART.3.Q1.GPIA
19219	UIS.NART.3.Q2.GPIA
19226	UIS.NART.3.Q3.GPIA
19233	UIS.NART.3.Q4.GPIA
19240	UIS.NART.3.Q5.GPIA
19247	UIS.NART.3.RUR.GPIA
19252	UIS.NART.3.RUR.Q1.GPIA
19256	UIS.NART.3.RUR.Q2.GPIA
19260	UIS.NART.3.RUR.Q3.GPIA
19264	UIS.NART.3.RUR.Q4.GPIA
19268	UIS.NART.3.RUR.Q5.GPIA
19274	UIS.NART.3.URB.GPIA
19279	UIS.NART.3.URB.Q1.GPIA
19283	UIS.NART.3.URB.Q2.GPIA
19287	UIS.NART.3.URB.Q3.GPIA
19291	UIS.NART.3.URB.Q4.GPIA
19295	UIS.NART.3.URB.Q5.GPIA
19301	UIS.NERA.AGM1.GPIA.CP

19305	UIS.NERT.1.GPI
19309	UIS.NERT.2.GPI
19313	UIS.NERT.3.GPI
19317	UIS.OAEPG.1.GPIA
19321	UIS.OAEPG.2.GPV.GPIA
19346	UIS.ONTRACK.THREE.DOMAINS.GPIA
19350	UIS.PER.11T15.BULLIED.GPIA
19363	UIS.POSTIMUENV.GPIA
19378	UIS.PRYA.12MO.GPI
19392	UIS.QUTP.02.GPIA
19396	UIS.QUTP.1.GPIA
19400	UIS.QUTP.2.GPIA
19404	UIS.QUTP.2T3.GPIA
19408	UIS.QUTP.3.GPIA
19463	UIS.READ.G2T3.GPIA
19479	UIS.READ.LOWERSEC.GPIA
19495	UIS.READ.PRIMARY.GPIA
19553	UIS.ROFST.1.GPIA.CP
19557	UIS.ROFST.1T2.GPIA.CP
19561	UIS.ROFST.1T3.GPIA.CP
19565	UIS.ROFST.2.GPIA.CP
19569	UIS.ROFST.2T3.GPIA.CP
19573	UIS.ROFST.3.GPIA.CP
19577	UIS.ROFST.AGM1.GPIA.CP
19583	UIS.ROFST.H.1.GPIA
19591	UIS.ROFST.H.1.Q1.GPIA
19598	UIS.ROFST.H.1.Q2.GPIA
19605	UIS.ROFST.H.1.Q3.GPIA
19612	UIS.ROFST.H.1.Q4.GPIA
19619	UIS.ROFST.H.1.Q5.GPIA
19626	UIS.ROFST.H.1.RUR.GPIA
19631	UIS.ROFST.H.1.RUR.Q1.GPIA
19635	UIS.ROFST.H.1.RUR.Q2.GPIA
19639	UIS.ROFST.H.1.RUR.Q3.GPIA
19643	UIS.ROFST.H.1.RUR.Q4.GPIA
19647	UIS.ROFST.H.1.RUR.Q5.GPIA
19653	UIS.ROFST.H.1.URB.GPIA
19658	UIS.ROFST.H.1.URB.Q1.GPIA
19662	UIS.ROFST.H.1.URB.Q2.GPIA
19666	UIS.ROFST.H.1.URB.Q3.GPIA
19670	UIS.ROFST.H.1.URB.Q4.GPIA
19674	UIS.ROFST.H.1.URB.Q5.GPIA
19682	UIS.ROFST.H.2.GPIA

19690	UIS.ROFST.H.2.Q1.GPIA
19697	UIS.ROFST.H.2.Q2.GPIA
19704	UIS.ROFST.H.2.Q3.GPIA
19711	UIS.ROFST.H.2.Q4.GPIA
19718	UIS.ROFST.H.2.Q5.GPIA
19725	UIS.ROFST.H.2.RUR.GPIA
19730	UIS.ROFST.H.2.RUR.Q1.GPIA
19734	UIS.ROFST.H.2.RUR.Q2.GPIA
19738	UIS.ROFST.H.2.RUR.Q3.GPIA
19742	UIS.ROFST.H.2.RUR.Q4.GPIA
19746	UIS.ROFST.H.2.RUR.Q5.GPIA
19752	UIS.ROFST.H.2.URB.GPIA
19757	UIS.ROFST.H.2.URB.Q1.GPIA
19761	UIS.ROFST.H.2.URB.Q2.GPIA
19765	UIS.ROFST.H.2.URB.Q3.GPIA
19769	UIS.ROFST.H.2.URB.Q4.GPIA
19773	UIS.ROFST.H.2.URB.Q5.GPIA
19781	UIS.ROFST.H.3.GPIA
19789	UIS.ROFST.H.3.Q1.GPIA
19796	UIS.ROFST.H.3.Q2.GPIA
19803	UIS.ROFST.H.3.Q3.GPIA
19810	UIS.ROFST.H.3.Q4.GPIA
19817	UIS.ROFST.H.3.Q5.GPIA
19824	UIS.ROFST.H.3.RUR.GPIA
19829	UIS.ROFST.H.3.RUR.Q1.GPIA
19833	UIS.ROFST.H.3.RUR.Q2.GPIA
19837	UIS.ROFST.H.3.RUR.Q3.GPIA
19841	UIS.ROFST.H.3.RUR.Q4.GPIA
19845	UIS.ROFST.H.3.RUR.Q5.GPIA
19851	UIS.ROFST.H.3.URB.GPIA
19856	UIS.ROFST.H.3.URB.Q1.GPIA
19860	UIS.ROFST.H.3.URB.Q2.GPIA
19864	UIS.ROFST.H.3.URB.Q3.GPIA
19868	UIS.ROFST.H.3.URB.Q4.GPIA
19872	UIS.ROFST.H.3.URB.Q5.GPIA
19925	UIS.SLE.02.GPI
19929	UIS.SLE.1.GPI
19936	UIS.SLE.123.GPI
19938	UIS.SLE.1T2.GPI
19939	UIS.SLE.1T6.GPI
19942	UIS.SLE.23.GPI
19946	UIS.SLE.4.GPI
19950	UIS.SLE.56.GPI

19954	UIS.SR.1.G4.GPI
19956	UIS.SR.1.G5.GPI
19957	UIS.SR.1.GLAST.GPI
19979	UIS.TATTRR.02.GPIA
19982	UIS.TATTRR.1.GPIA
19986	UIS.TATTRR.2.GPIA
19991	UIS.TATTRR.2T3.GPIA
20000	UIS.TATTRR.3.GPIA
20014	UIS.TRTP.02.GPIA
20016	UIS.TRTP.1.GPIA
20019	UIS.TRTP.2.GPIA
20021	UIS.TRTP.2T3.GPIA
20024	UIS.TRTP.3.GPIA
20143	UIS.YADULT.PROFILITERACY.GPIA
20153	UIS.YADULT.PROFINUMERACY.GPIA

169
172
709
1573
1575
1587
1589
1712
1716
1717
1718
1719
1723
1735
1736
1737
1740
1744
1747
1748
2202
2203
2204
2205
2206
2207
2208

2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2296
2297
2298
2299
2300
2301
6251 589_Do laws and
8632
8904
9755
9756
9757
9758
9759
9760
9761
14651
14667
15164
15175
15177
15178
15180
15998

16011
16027
17766
18119
18120
18129
18137
18144
18151
18158
18165
18172
18177
18181
18185
18189
18193
18199
18204
18208
18212
18216
18220
18228
18236
18243
18250
18257
18264
18271
18276
18280
18284
18288
18292
18298
18303
18307
18311
18315
18319
18327
18335

18342
18349
18356
18363
18370
18375
18379
18383
18387
18391
18397
18402
18406
18410
18414
18418
18459
18466 UI
18473 UI
18480 U
18486 UIS: I
18493 UIS: Perc
18500 UIS: Pe
18504
18517
18525 Percentage of students in lower seconda
18528 Percentage of stude
18532 Percentage of students in lower secondary education sho
18536 Percentage of students in lower secondary education sh
18540
18550 Pe
18569
18577
18584
18591
18598
18605
18612
18617
18621
18625
18629
18633

18639
18644
18648
18652
18656
18660
18667 Percentage of students in lower secondary education showir
18670 Percentage of students in lower secor
18674 Percentage of students in lower secondary education showing adequate ur
18676 Percentage of students in lower secondary education showir
18677 Percentage of students in lower secondary education sh
18678 Percentage of students in lower secondary education showing adequate understanc
18679 Percentage of students in lower secondary education
18682 Percentage of students in lower secondary education showing adequate understanding of
18686 Percentage of students in lower secondary education showing adequate understandi
18690 Percentage of students in lower secondary education showing adequate
18694 Percentage of students in lower secondary education showing adequate understanding of i
18698 Percentage of students in lower secondary education showing adequate understar
18702
18706
18708
18711
18717
18719
18720
18723
18725
18726
18742
18746
18750
18754
18758 Proportion
18762
18766
18770
18774
18793
18798
18802
18805
18810
18814

18819
18825
18829
18835
18840
18844
18848
18864 Proporti
18880 I
18907
18915
18922
18929
18936
18943
18950
18955
18959
18963
18967
18971
18977
18982
18986
18990
18994
18998
19006
19014
19021
19028
19035
19042
19049
19054
19058
19062
19066
19070
19076
19081
19085
19089

19093
19097
19105
19113
19120
19127
19134
19141
19148
19153
19157
19161
19165
19169
19175
19180
19184
19188
19192
19196
19204
19212
19219
19226
19233
19240
19247
19252
19256
19260
19264
19268
19274
19279
19283
19287
19291
19295
19301
19305
19309
19313
19317 Pe

19321 Percentage of pupil
19346 Proportion of cl
19350
19363
19378 Partic
19392
19396
19400
19404
19408
19463
19479 Propo
19495
19553
19557
19561 Out-of-s
19565
19569
19573
19577
19583
19591
19598
19605
19612
19619
19626
19631 Out-of-school ra
19635 Out-of-school r
19639 Out-of-school r
19643 Out-of-school r
19647 Out-of-school ra
19653
19658 Out-of-school ra
19662 Out-of-school r
19666 Out-of-school r
19670 Out-of-school r
19674 Out-of-school ra
19682
19690 O
19697
19704
19711

19718	O
19725	
19730	Out-of-s
19734	Out-of-
19738	Out-of-
19742	Out-of-
19746	Out-of-s
19752	
19757	Out-of-s
19761	Out-of-
19765	Out-of-
19769	Out-of-
19773	Out-of-s
19781	
19789	
19796	
19803	
19810	
19817	
19824	
19829	Ou
19833	O
19837	O
19841	O
19845	Ou
19851	
19856	Ou
19860	O
19864	O
19868	O
19872	Ou
19925	
19929	
19936	
19938	
19939	
19942	
19946	
19950	
19954	
19956	
19957	
19979	

```
19982
19986
19991
20000
20014
20016
20019
20021
20024
20143
20153
```

```r
WDI::WDI(country = "UY",
    indicator = "SP.POP.TOTL",
    start = 2000,
    end = 2023,
    extra = FALSE,
    cache = NULL)
```

```
   country iso2c iso3c year SP.POP.TOTL
1  Uruguay    UY   URY 2023     3423108
2  Uruguay    UY   URY 2022     3422794
3  Uruguay    UY   URY 2021     3426260
4  Uruguay    UY   URY 2020     3429086
5  Uruguay    UY   URY 2019     3428409
6  Uruguay    UY   URY 2018     3427042
7  Uruguay    UY   URY 2017     3422200
8  Uruguay    UY   URY 2016     3413766
9  Uruguay    UY   URY 2015     3402818
10 Uruguay    UY   URY 2014     3391662
11 Uruguay    UY   URY 2013     3381180
12 Uruguay    UY   URY 2012     3371133
13 Uruguay    UY   URY 2011     3361637
14 Uruguay    UY   URY 2010     3352651
15 Uruguay    UY   URY 2009     3344156
16 Uruguay    UY   URY 2008     3336126
17 Uruguay    UY   URY 2007     3328651
18 Uruguay    UY   URY 2006     3322282
19 Uruguay    UY   URY 2005     3317665
20 Uruguay    UY   URY 2004     3313801
21 Uruguay    UY   URY 2003     3310202
```

```
22 Uruguay     UY    URY 2002      3306441
23 Uruguay     UY    URY 2001      3300939
24 Uruguay     UY    URY 2000      3292224
```

Esta información la podemos guardar en un objeto. En este caso mejor pediremos un solo país:

```
pop <- WDI::WDI(country = "UY",
               indicator = "SP.POP.TOTL",
               start = 1990,
               end = 2023)
```

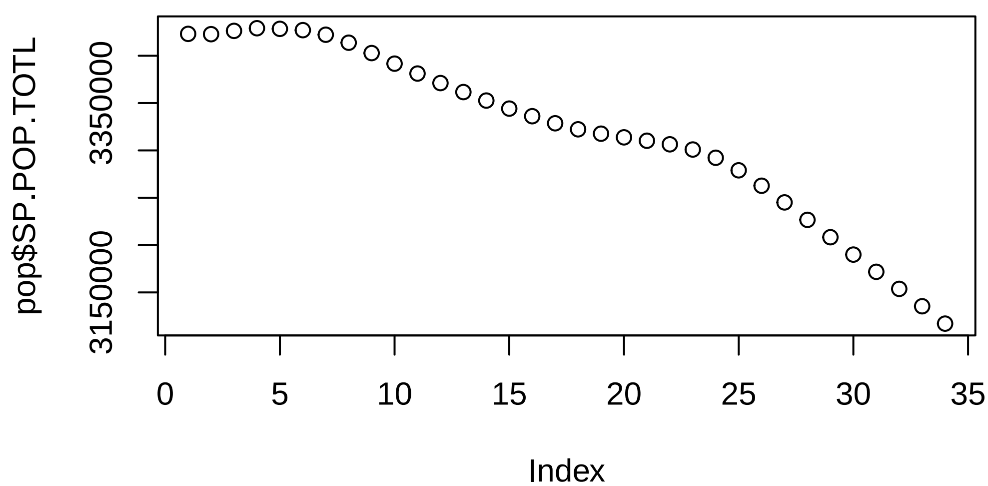
Vamos a revisar nuestro objeto:

```
class(pop)
```

```
[1] "data.frame"
```

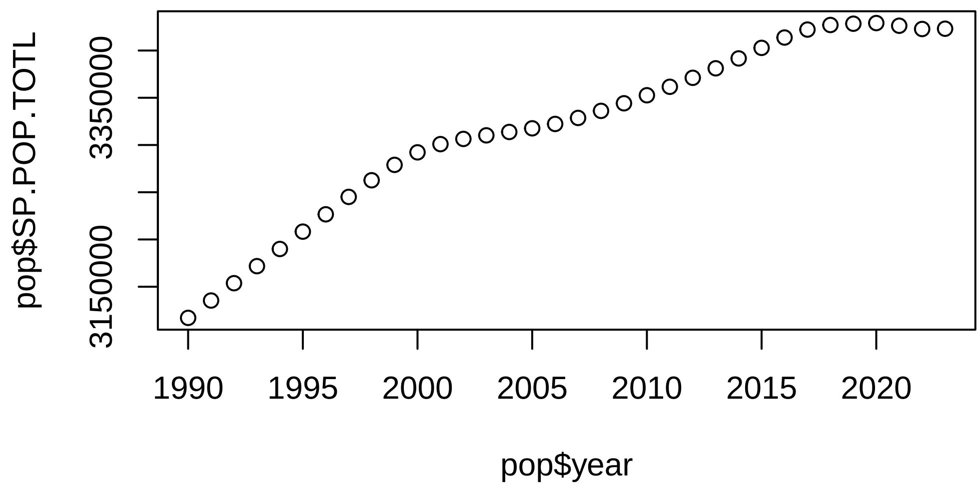
Veamos y conozcamos la función `plot()`

```
plot(pop$SP.POP.TOTL)
```



Este no es el mejor gráfico.

```
plot(pop$year, pop$SP.POP.TOTL)
```



Las matrices por lo general sólo almacenan un tipo de datos mientras que las data frames puede almacenar varios tipos de datos.

1.8 Importación de datos

1.8.1 Desde Excel

El paquete más compatible con RStudio es `{readxl}`. Como su nombre dice "lee" los archivos de excel

```
ejemploxl <- readxl::read_excel("datos/ejemplo_xlsx.xlsx")
```

https://catalogodatos.gub.uy/dataset/mides-indicador-10829/resource/3f5356a2-b6dc-4827-8a8e-e34285ef54ba

Como el nombre de paquete lo indica, sólo lee. Para "escribir" en este formato, recomiendo el paquete `{writexl}`. Lo instalamos anteriormente.

Si quisiéramos exportar un objeto a Excel, se hace de la siguiente forma:

```
writexl::write_xlsx(ejemploxl, path = "Mi_Exportación.xlsx")
```

1.8.2 Desde STATA y SPSS

Si bien también se puede realizar desde el paquete `{foreign}` Pero este no importa algunas características como las etiquetas y tampoco funciona con las versiones más nuevas de STATA. Vamos a instalar otro paquete, compatible con el mundo `{tidyverse}`.

Recuerda que no hay que instalarlo (viene adentro de `{tidyverse}`).

```
encuesta_generacion <- haven::read_dta("datos/GGSII_Wave1_UY_V_1_3.dta")
```

!Importante, a R no le gustan los objetos con nombres que empiezan en números

El paquete haven sí exporta información.

```
haven::write_dta(encuesta_generacion,
                 "datos/mi_exportación.dta",
                 version = 12)
```

Con SSPS es muy parecido. Dentro de `{haven}` hay una función específica para ello.

Checa que en todas las exportaciones en los nombres hay que incluir la extensión del programa. Si quieres guardar en un lugar diferente al directorio del trabajo, hay que escribir toda la ruta dentro de la computadora.

1.8.3 Desde archivos de texto y de una url

Desde el portal https://catalogodatos.gub.uy/ tenemos acceso a directo a varias fuentes de información, al ser datos abiertos, los archivos de texto son muy comunes.

Leeremos parte de esa información, específicamente de la actividad docente

```
docente2019 <- read.csv("https://catalogodatos.gub.uy/dataset/e5b78d49-1707-4f50-9b3b-f2db
names(docente2019)
```

```
[1] "Id.persona"
[2] "Sexo"
[3] "Rol"
[4] "Departamento"
```

```
 [5] "Subsistema"
 [6] "Año.lectivo"
 [7] "Cantidad.de.días.ingreso.a.CREA"
 [8] "Cantidad.de.Comentarios.posteados"
 [9] "Cantidad.de.Acciones.totales"
[10] "Cantidad.de.días.de.ingreso.a.Biblioteca"
[11] "Cantidad.de.préstamos.en.biblioteca"
```

```
docente2019 <- readr::read_csv("https://catalogodatos.gub.uy/dataset/e5b78d49-1707-4f50-9b
```

```
Rows: 51370 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (4): Sexo, Rol, Departamento, Subsistema
dbl (7): Id persona, Año lectivo, Cantidad de días ingreso a CREA, Cantidad ...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
names(docente2019)
```

```
 [1] "Id persona"
 [2] "Sexo"
 [3] "Rol"
 [4] "Departamento"
 [5] "Subsistema"
 [6] "Año lectivo"
 [7] "Cantidad de días ingreso a CREA"
 [8] "Cantidad de Comentarios posteados"
 [9] "Cantidad de Acciones totales"
[10] "Cantidad de días de ingreso a Biblioteca"
[11] "Cantidad de préstamos en biblioteca"
```

1.9 Revisión de nuestro conjunto de datos

1.9.1 con base

Vamos a revisar la base, brevemente la base

```r
class(encuesta_generacion) # tipo de objeto
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

```r
names(encuesta_generacion) # lista las variables
```

```
  [1] "country"          "region"           "respid"          
  [4] "intid"            "mode"             "weight"          
  [7] "instrument"       "intdatem"         "intdatey"        
 [10] "dem01"            "dem02m"           "dem02y"          
 [13] "dem03"            "dem04a"           "dem04biso"       
 [16] "dem05m"           "dem05y"           "dem06"           
 [19] "dem07"            "dem07isced"       "dem08m"          
 [22] "dem08y"           "dem09"            "dem10m"          
 [25] "dem10y"           "dem11"            "dem12"           
 [28] "dem14"            "dem15"            "dem17"           
 [31] "dem18"            "dem19"            "dem20"           
 [34] "dem21"            "dem22a"           "dem22m"          
 [37] "dem22y"           "dem23"            "dem24a"          
 [40] "dem24biso"        "dem24em"          "dem24ey"         
 [43] "dem25"            "dem25isced"       "dem26"           
 [46] "dem27"            "dem28a"           "dem28bm"         
 [49] "dem28by"          "dem28c"           "dem30a"          
 [52] "dem30bm"          "dem30by"          "dem30c"          
 [55] "dem30d"           "dem31m"           "dem31y"          
 [58] "dem32a"           "dem32b"           "dem32c"          
 [61] "dem32d"           "dem33"            "dem33am"         
 [64] "dem33ay"          "dem34m"           "dem34y"          
 [67] "dem35"            "dem36a"           "dem36au"         
 [70] "dem36b"           "dem36bu"          "dem37"           
 [73] "dem38a"           "dem38b"           "dem38c"          
 [76] "dem38d"           "dem38e"           "dem38f"          
 [79] "dem38g"           "dem39a"           "dem39b"          
 [82] "dem39c"           "dem39d"           "dem40"           
 [85] "dem41"            "dem42"            "dem43"           
 [88] "dem44"            "dem45"            "dem46"           
 [91] "lhi01"            "lhi02"            "lhi04_m1"        
 [94] "lhi04_m2"         "lhi04_m3"         "lhi04_m4"        
 [97] "lhi04_m5"         "lhi04_m6"         "lhi04_m7"        
[100] "lhi04_m8"         "lhi04_m9"         "lhi04_m10"       
```

```
[103] "lhi04_m11"           "lhi04_m12"           "lhi04_m13"
[106] "lhi04_m14"           "lhi04_m15"           "lhi04_m16"
[109] "lhi04_m17"           "lhi04_m18"           "lhi04_m19"
[112] "lhi04_m20"           "lhi04_y1"            "lhi04_y2"
[115] "lhi04_y3"            "lhi04_y4"            "lhi04_y5"
[118] "lhi04_y6"            "lhi04_y7"            "lhi04_y8"
[121] "lhi04_y9"            "lhi04_y10"           "lhi04_y11"
[124] "lhi04_y12"           "lhi04_y13"           "lhi04_y14"
[127] "lhi04_y15"           "lhi04_y16"           "lhi04_y17"
[130] "lhi04_y18"           "lhi04_y19"           "lhi04_y20"
[133] "lhi04a_1"            "lhi04a_2"            "lhi04a_3"
[136] "lhi04a_4"            "lhi04a_5"            "lhi04a_6"
[139] "lhi04a_7"            "lhi04a_8"            "lhi04a_9"
[142] "lhi04a_10"           "lhi04a_11"           "lhi04a_12"
[145] "lhi04a_13"           "lhi04a_14"           "lhi04a_15"
[148] "lhi04a_16"           "lhi04a_17"           "lhi04a_18"
[151] "lhi04a_19"           "lhi04a_20"           "lhi05a_1"
[154] "lhi05a_2"            "lhi05a_3"            "lhi05a_4"
[157] "lhi05a_5"            "lhi05a_6"            "lhi05a_7"
[160] "lhi05a_8"            "lhi05a_9"            "lhi05a_10"
[163] "lhi05a_11"           "lhi05a_12"           "lhi05a_13"
[166] "lhi05a_14"           "lhi05a_15"           "lhi05a_16"
[169] "lhi05a_17"           "lhi05a_18"           "lhi05a_19"
[172] "lhi05a_20"           "lhi05b_m1"           "lhi05b_m2"
[175] "lhi05b_m3"           "lhi05b_m4"           "lhi05b_m5"
[178] "lhi05b_m6"           "lhi05b_m7"           "lhi05b_m8"
[181] "lhi05b_m9"           "lhi05b_m10"          "lhi05b_m11"
[184] "lhi05b_m12"          "lhi05b_m13"          "lhi05b_m14"
[187] "lhi05b_m15"          "lhi05b_m16"          "lhi05b_m17"
[190] "lhi05b_m18"          "lhi05b_m19"          "lhi05b_m20"
[193] "lhi05b_y1"           "lhi05b_y2"           "lhi05b_y3"
[196] "lhi05b_y4"           "lhi05b_y5"           "lhi05b_y6"
[199] "lhi05b_y7"           "lhi05b_y8"           "lhi05b_y9"
[202] "lhi05b_y10"          "lhi05b_y11"          "lhi05b_y12"
[205] "lhi05b_y13"          "lhi05b_y14"          "lhi05b_y15"
[208] "lhi05b_y16"          "lhi05b_y17"          "lhi05b_y18"
[211] "lhi05b_y19"          "lhi05b_y20"          "lhi06_m1"
[214] "lhi06_m2"            "lhi06_m3"            "lhi06_m4"
[217] "lhi06_m5"            "lhi06_m6"            "lhi06_m7"
[220] "lhi06_m8"            "lhi06_m9"            "lhi06_m10"
[223] "lhi06_m11"           "lhi06_m12"           "lhi06_m13"
[226] "lhi06_m14"           "lhi06_m15"           "lhi06_m16"
[229] "lhi06_m17"           "lhi06_m18"           "lhi06_m19"
```

```
[232] "lhi06_m20"          "lhi06_y1"           "lhi06_y2"
[235] "lhi06_y3"           "lhi06_y4"           "lhi06_y5"
[238] "lhi06_y6"           "lhi06_y7"           "lhi06_y8"
[241] "lhi06_y9"           "lhi06_y10"          "lhi06_y11"
[244] "lhi06_y12"          "lhi06_y13"          "lhi06_y14"
[247] "lhi06_y15"          "lhi06_y16"          "lhi06_y17"
[250] "lhi06_y18"          "lhi06_y19"          "lhi06_y20"
[253] "lhi07_1"            "lhi07_2"            "lhi07_3"
[256] "lhi07_4"            "lhi07_5"            "lhi07_6"
[259] "lhi07_7"            "lhi07_8"            "lhi07_9"
[262] "lhi07_10"           "lhi07_11"           "lhi07_12"
[265] "lhi07_13"           "lhi07_14"           "lhi07_15"
[268] "lhi07_16"           "lhi07_17"           "lhi07_18"
[271] "lhi07_19"           "lhi07_20"           "lhi08_1"
[274] "lhi08_2"            "lhi08_3"            "lhi08_4"
[277] "lhi08_5"            "lhi08_6"            "lhi08_7"
[280] "lhi08_8"            "lhi08_9"            "lhi08_10"
[283] "lhi08_11"           "lhi08_12"           "lhi08_13"
[286] "lhi08_14"           "lhi08_15"           "lhi08_16"
[289] "lhi08_17"           "lhi08_18"           "lhi08_19"
[292] "lhi08_20"           "lhi09_1"            "lhi09_2"
[295] "lhi09_3"            "lhi09_4"            "lhi09_5"
[298] "lhi09_6"            "lhi09_7"            "lhi09_8"
[301] "lhi09_9"            "lhi09_10"           "lhi09_11"
[304] "lhi09_12"           "lhi09_13"           "lhi09_14"
[307] "lhi09_15"           "lhi09_16"           "lhi09_17"
[310] "lhi09_18"           "lhi09_19"           "lhi09_20"
[313] "lhi10_1"            "lhi10_2"            "lhi10_3"
[316] "lhi10_4"            "lhi10_5"            "lhi10_6"
[319] "lhi10_7"            "lhi10_8"            "lhi10_9"
[322] "lhi10_10"           "lhi10_11"           "lhi10_12"
[325] "lhi10_13"           "lhi10_14"           "lhi10_15"
[328] "lhi10_16"           "lhi10_17"           "lhi10_18"
[331] "lhi10_19"           "lhi10_20"           "lhi11_1"
[334] "lhi11_2"            "lhi11_3"            "lhi11_4"
[337] "lhi11_5"            "lhi11_6"            "lhi11_7"
[340] "lhi11_8"            "lhi11_9"            "lhi11_10"
[343] "lhi11_11"           "lhi11_12"           "lhi11_13"
[346] "lhi11_14"           "lhi11_15"           "lhi11_16"
[349] "lhi11_17"           "lhi11_18"           "lhi11_19"
[352] "lhi11_20"           "lhi12_1"            "lhi12_2"
[355] "lhi12_3"            "lhi12_4"            "lhi12_5"
[358] "lhi12_6"            "lhi12_7"            "lhi12_8"
```

```
[361] "lhi12_9"         "lhi12_10"        "lhi12_11"
[364] "lhi12_12"        "lhi12_13"        "lhi12_14"
[367] "lhi12_15"        "lhi12_16"        "lhi12_17"
[370] "lhi12_18"        "lhi12_19"        "lhi12_20"
[373] "lhi13_1"         "lhi13_2"         "lhi13_3"
[376] "lhi13_4"         "lhi13_5"         "lhi13_6"
[379] "lhi13_7"         "lhi13_8"         "lhi13_9"
[382] "lhi13_10"        "lhi13_11"        "lhi13_12"
[385] "lhi13_13"        "lhi13_14"        "lhi13_15"
[388] "lhi13_16"        "lhi13_17"        "lhi13_18"
[391] "lhi13_19"        "lhi13_20"        "lhi14_m1"
[394] "lhi14_m2"        "lhi14_m3"        "lhi14_m4"
[397] "lhi14_m5"        "lhi14_m6"        "lhi14_m7"
[400] "lhi14_m8"        "lhi14_m9"        "lhi14_m10"
[403] "lhi14_m11"       "lhi14_m12"       "lhi14_m13"
[406] "lhi14_m14"       "lhi14_m15"       "lhi14_m16"
[409] "lhi14_m17"       "lhi14_m18"       "lhi14_m19"
[412] "lhi14_m20"       "lhi14_y1"        "lhi14_y2"
[415] "lhi14_y3"        "lhi14_y4"        "lhi14_y5"
[418] "lhi14_y6"        "lhi14_y7"        "lhi14_y8"
[421] "lhi14_y9"        "lhi14_y10"       "lhi14_y11"
[424] "lhi14_y12"       "lhi14_y13"       "lhi14_y14"
[427] "lhi14_y15"       "lhi14_y16"       "lhi14_y17"
[430] "lhi14_y18"       "lhi14_y19"       "lhi14_y20"
[433] "lhi15a_1"        "lhi15a_2"        "lhi15a_3"
[436] "lhi15a_4"        "lhi15a_5"        "lhi15a_6"
[439] "lhi15a_7"        "lhi15a_8"        "lhi15a_9"
[442] "lhi15a_10"       "lhi15a_11"       "lhi15a_12"
[445] "lhi15a_13"       "lhi15a_14"       "lhi15a_15"
[448] "lhi15a_16"       "lhi15a_17"       "lhi15a_18"
[451] "lhi15a_19"       "lhi15a_20"       "lhi15b_m1"
[454] "lhi15b_m2"       "lhi15b_m3"       "lhi15b_m4"
[457] "lhi15b_m5"       "lhi15b_m6"       "lhi15b_m7"
[460] "lhi15b_m8"       "lhi15b_m9"       "lhi15b_m10"
[463] "lhi15b_m11"      "lhi15b_m12"      "lhi15b_m13"
[466] "lhi15b_m14"      "lhi15b_m15"      "lhi15b_m16"
[469] "lhi15b_m17"      "lhi15b_m18"      "lhi15b_m19"
[472] "lhi15b_m20"      "lhi15b_y1"       "lhi15b_y2"
[475] "lhi15b_y3"       "lhi15b_y4"       "lhi15b_y5"
[478] "lhi15b_y6"       "lhi15b_y7"       "lhi15b_y8"
[481] "lhi15b_y9"       "lhi15b_y10"      "lhi15b_y11"
[484] "lhi15b_y12"      "lhi15b_y13"      "lhi15b_y14"
[487] "lhi15b_y15"      "lhi15b_y16"      "lhi15b_y17"
```

```
[490] "lhi15b_y18"         "lhi15b_y19"         "lhi15b_y20"
[493] "lhi16_1"            "lhi16_2"            "lhi16_3"
[496] "lhi16_4"            "lhi16_5"            "lhi16_6"
[499] "lhi16_7"            "lhi16_8"            "lhi16_9"
[502] "lhi16_10"           "lhi16_11"           "lhi16_12"
[505] "lhi16_13"           "lhi16_14"           "lhi16_15"
[508] "lhi16_16"           "lhi16_17"           "lhi16_18"
[511] "lhi16_19"           "lhi16_20"           "lhi17_1"
[514] "lhi17_2"            "lhi17_3"            "lhi17_4"
[517] "lhi17_5"            "lhi17_6"            "lhi17_7"
[520] "lhi17_8"            "lhi17_9"            "lhi17_10"
[523] "lhi17_11"           "lhi17_12"           "lhi17_13"
[526] "lhi17_14"           "lhi17_15"           "lhi17_16"
[529] "lhi17_17"           "lhi17_18"           "lhi17_19"
[532] "lhi17_20"           "lhi18"              "lhi19"
[535] "lhi20"              "lhi21"              "lhi22"
[538] "lhi23"              "lhi25_1"            "lhi25_2"
[541] "lhi25_3"            "lhi25_4"            "lhi25_5"
[544] "lhi25_6"            "lhi25_7"            "lhi25_8"
[547] "lhi25_9"            "lhi25_10"           "lhi25_11"
[550] "lhi25_12"           "lhi25_13"           "lhi25_14"
[553] "lhi25_15"           "lhi25_16"           "lhi25_17"
[556] "lhi25_18"           "lhi25_19"           "lhi25_20"
[559] "lhi26_1"            "lhi26_2"            "lhi26_3"
[562] "lhi26_4"            "lhi26_5"            "lhi26_6"
[565] "lhi26_7"            "lhi26_8"            "lhi26_9"
[568] "lhi26_10"           "lhi26_11"           "lhi26_12"
[571] "lhi26_13"           "lhi26_14"           "lhi26_15"
[574] "lhi26_16"           "lhi26_17"           "lhi26_18"
[577] "lhi26_19"           "lhi26_20"           "lhi27_1"
[580] "lhi27_2"            "lhi27_3"            "lhi27_4"
[583] "lhi27_5"            "lhi27_6"            "lhi27_7"
[586] "lhi27_8"            "lhi27_9"            "lhi27_10"
[589] "lhi27_11"           "lhi27_12"           "lhi27_13"
[592] "lhi27_14"           "lhi27_15"           "lhi27_16"
[595] "lhi27_17"           "lhi27_18"           "lhi27_19"
[598] "lhi27_20"           "lhi28_1"            "lhi28_2"
[601] "lhi28_3"            "lhi28_4"            "lhi28_5"
[604] "lhi28_6"            "lhi28_7"            "lhi28_8"
[607] "lhi28_9"            "lhi28_10"           "lhi28_11"
[610] "lhi28_12"           "lhi28_13"           "lhi28_14"
[613] "lhi28_15"           "lhi28_16"           "lhi28_17"
[616] "lhi28_18"           "lhi28_19"           "lhi28_20"
```

```
[619] "lhi29_m1"            "lhi29_m2"            "lhi29_m3"
[622] "lhi29_m4"            "lhi29_m5"            "lhi29_m6"
[625] "lhi29_m7"            "lhi29_m8"            "lhi29_m9"
[628] "lhi29_m10"           "lhi29_m11"           "lhi29_m12"
[631] "lhi29_m13"           "lhi29_m14"           "lhi29_m15"
[634] "lhi29_m16"           "lhi29_m17"           "lhi29_m18"
[637] "lhi29_m19"           "lhi29_m20"           "lhi29_y1"
[640] "lhi29_y2"            "lhi29_y3"            "lhi29_y4"
[643] "lhi29_y5"            "lhi29_y6"            "lhi29_y7"
[646] "lhi29_y8"            "lhi29_y9"            "lhi29_y10"
[649] "lhi29_y11"           "lhi29_y12"           "lhi29_y13"
[652] "lhi29_y14"           "lhi29_y15"           "lhi29_y16"
[655] "lhi29_y17"           "lhi29_y18"           "lhi29_y19"
[658] "lhi29_y20"           "lhi30_m1"            "lhi30_m2"
[661] "lhi30_m3"            "lhi30_m4"            "lhi30_m5"
[664] "lhi30_m6"            "lhi30_m7"            "lhi30_m8"
[667] "lhi30_m9"            "lhi30_m10"           "lhi30_m11"
[670] "lhi30_m12"           "lhi30_m13"           "lhi30_m14"
[673] "lhi30_m15"           "lhi30_m16"           "lhi30_m17"
[676] "lhi30_m18"           "lhi30_m19"           "lhi30_m20"
[679] "lhi30_y1"            "lhi30_y2"            "lhi30_y3"
[682] "lhi30_y4"            "lhi30_y5"            "lhi30_y6"
[685] "lhi30_y7"            "lhi30_y8"            "lhi30_y9"
[688] "lhi30_y10"           "lhi30_y11"           "lhi30_y12"
[691] "lhi30_y13"           "lhi30_y14"           "lhi30_y15"
[694] "lhi30_y16"           "lhi30_y17"           "lhi30_y18"
[697] "lhi30_y19"           "lhi30_y20"           "lhi31_1"
[700] "lhi31_2"             "lhi31_3"             "lhi31_4"
[703] "lhi31_5"             "lhi31_6"             "lhi31_7"
[706] "lhi31_8"             "lhi31_9"             "lhi31_10"
[709] "lhi31_11"            "lhi31_12"            "lhi31_13"
[712] "lhi31_14"            "lhi31_15"            "lhi31_16"
[715] "lhi31_17"            "lhi31_18"            "lhi31_19"
[718] "lhi31_20"            "lhi32_1"             "lhi32_2"
[721] "lhi32_3"             "lhi32_4"             "lhi32_5"
[724] "lhi32_6"             "lhi32_7"             "lhi32_8"
[727] "lhi32_9"             "lhi32_10"            "lhi32_11"
[730] "lhi32_12"            "lhi32_13"            "lhi32_14"
[733] "lhi32_15"            "lhi32_16"            "lhi32_17"
[736] "lhi32_18"            "lhi32_19"            "lhi32_20"
[739] "lhi33_1"             "lhi33_2"             "lhi33_3"
[742] "lhi33_4"             "lhi33_5"             "lhi33_6"
[745] "lhi33_7"             "lhi33_8"             "lhi33_9"
```

```
[748] "lhi33_10"           "lhi33_11"           "lhi33_12"
[751] "lhi33_13"           "lhi33_14"           "lhi33_15"
[754] "lhi33_16"           "lhi33_17"           "lhi33_18"
[757] "lhi33_19"           "lhi33_20"           "lhi33u_1"
[760] "lhi33u_2"           "lhi33u_3"           "lhi33u_4"
[763] "lhi33u_5"           "lhi33u_6"           "lhi33u_7"
[766] "lhi33u_8"           "lhi33u_9"           "lhi33u_10"
[769] "lhi33u_11"          "lhi33u_12"          "lhi33u_13"
[772] "lhi33u_14"          "lhi33u_15"          "lhi33u_16"
[775] "lhi33u_17"          "lhi33u_18"          "lhi33u_19"
[778] "lhi33u_20"          "lhi34_1"            "lhi34_2"
[781] "lhi34_3"            "lhi34_4"            "lhi34_5"
[784] "lhi34_6"            "lhi34_7"            "lhi34_8"
[787] "lhi34_9"            "lhi34_10"           "lhi34_11"
[790] "lhi34_12"           "lhi34_13"           "lhi34_14"
[793] "lhi34_15"           "lhi34_16"           "lhi34_17"
[796] "lhi34_18"           "lhi34_19"           "lhi34_20"
[799] "lhi35_1"            "lhi35_2"            "lhi35_3"
[802] "lhi35_4"            "lhi35_5"            "lhi35_6"
[805] "lhi35_7"            "lhi35_8"            "lhi35_9"
[808] "lhi35_10"           "lhi35_11"           "lhi35_12"
[811] "lhi35_13"           "lhi35_14"           "lhi35_15"
[814] "lhi35_16"           "lhi35_17"           "lhi35_18"
[817] "lhi35_19"           "lhi35_20"           "lhi36_1"
[820] "lhi36_2"            "lhi36_3"            "lhi36_4"
[823] "lhi36_5"            "lhi36_6"            "lhi36_7"
[826] "lhi36_8"            "lhi36_9"            "lhi36_10"
[829] "lhi36_11"           "lhi36_12"           "lhi36_13"
[832] "lhi36_14"           "lhi36_15"           "lhi36_16"
[835] "lhi36_17"           "lhi36_18"           "lhi36_19"
[838] "lhi36_20"           "lhi37_1"            "lhi37_2"
[841] "lhi37_3"            "lhi37_4"            "lhi37_5"
[844] "lhi37_6"            "lhi37_7"            "lhi37_8"
[847] "lhi37_9"            "lhi37_10"           "lhi37_11"
[850] "lhi37_12"           "lhi37_13"           "lhi37_14"
[853] "lhi37_15"           "lhi37_16"           "lhi37_17"
[856] "lhi37_18"           "lhi37_19"           "lhi37_20"
[859] "lhi38_1"            "lhi38_2"            "lhi38_3"
[862] "lhi38_4"            "lhi38_5"            "lhi38_6"
[865] "lhi38_7"            "lhi38_8"            "lhi38_9"
[868] "lhi38_10"           "lhi38_11"           "lhi38_12"
[871] "lhi38_13"           "lhi38_14"           "lhi38_15"
[874] "lhi38_16"           "lhi38_17"           "lhi38_18"
```

```
 [877] "lhi38_19"            "lhi38_20"            "lhi39a_1"
 [880] "lhi39a_2"            "lhi39a_3"            "lhi39a_4"
 [883] "lhi39a_5"            "lhi39a_6"            "lhi39a_7"
 [886] "lhi39a_8"            "lhi39a_9"            "lhi39a_10"
 [889] "lhi39a_11"           "lhi39a_12"           "lhi39a_13"
 [892] "lhi39a_14"           "lhi39a_15"           "lhi39a_16"
 [895] "lhi39a_17"           "lhi39a_18"           "lhi39a_19"
 [898] "lhi39a_20"           "lhi39au_1"           "lhi39au_2"
 [901] "lhi39au_3"           "lhi39au_4"           "lhi39au_5"
 [904] "lhi39au_6"           "lhi39au_7"           "lhi39au_8"
 [907] "lhi39au_9"           "lhi39au_10"          "lhi39au_11"
 [910] "lhi39au_12"          "lhi39au_13"          "lhi39au_14"
 [913] "lhi39au_15"          "lhi39au_16"          "lhi39au_17"
 [916] "lhi39au_18"          "lhi39au_19"          "lhi39au_20"
 [919] "lhi39b_1"            "lhi39b_2"            "lhi39b_3"
 [922] "lhi39b_4"            "lhi39b_5"            "lhi39b_6"
 [925] "lhi39b_7"            "lhi39b_8"            "lhi39b_9"
 [928] "lhi39b_10"           "lhi39b_11"           "lhi39b_12"
 [931] "lhi39b_13"           "lhi39b_14"           "lhi39b_15"
 [934] "lhi39b_16"           "lhi39b_17"           "lhi39b_18"
 [937] "lhi39b_19"           "lhi39b_20"           "lhi39bu_1"
 [940] "lhi39bu_2"           "lhi39bu_3"           "lhi39bu_4"
 [943] "lhi39bu_5"           "lhi39bu_6"           "lhi39bu_7"
 [946] "lhi39bu_8"           "lhi39bu_9"           "lhi39bu_10"
 [949] "lhi39bu_11"          "lhi39bu_12"          "lhi39bu_13"
 [952] "lhi39bu_14"          "lhi39bu_15"          "lhi39bu_16"
 [955] "lhi39bu_17"          "lhi39bu_18"          "lhi39bu_19"
 [958] "lhi39bu_20"          "lhi40_1"             "lhi40_2"
 [961] "lhi40_3"             "lhi40_4"             "lhi40_5"
 [964] "lhi40_6"             "lhi40_7"             "lhi40_8"
 [967] "lhi40_9"             "lhi40_10"            "lhi40_11"
 [970] "lhi40_12"            "lhi40_13"            "lhi40_14"
 [973] "lhi40_15"            "lhi40_16"            "lhi40_17"
 [976] "lhi40_18"            "lhi40_19"            "lhi40_20"
 [979] "lhi41_1"             "lhi41_2"             "lhi41_3"
 [982] "lhi41_4"             "lhi41_5"             "lhi41_6"
 [985] "lhi41_7"             "lhi41_8"             "lhi41_9"
 [988] "lhi41_10"            "lhi41_11"            "lhi41_12"
 [991] "lhi41_13"            "lhi41_14"            "lhi41_15"
 [994] "lhi41_16"            "lhi41_17"            "lhi41_18"
 [997] "lhi41_19"            "lhi41_20"            "fer01a"
[1000] "fer01b"              "fer01c"              "fer02m"
[1003] "fer02y"              "fer03"               "fer04"
```

```
[1006] "fer04b"      "fer04c"      "fer04d"
[1009] "fer04e"      "fer05"       "fer06"
[1012] "fer07_1"     "fer07_2"     "fer07_3"
[1015] "fer07_4"     "fer07_5"     "fer07_6"
[1018] "fer07_7"     "fer07_8"     "fer07_9"
[1021] "fer07_10"    "fer08"       "fer09"
[1024] "fer10a"      "fer10bm"     "fer10by"
[1027] "fer11_1"     "fer11_2"     "fer11_3"
[1030] "fer11_4"     "fer11_5"     "fer11_6"
[1033] "fer11_7"     "fer11_8"     "fer12_1"
[1036] "fer12_2"     "fer12_3"     "fer12_4"
[1039] "fer12_5"     "fer12_6"     "fer12_7"
[1042] "fer12_8"     "fer12_9"     "fer12_10"
[1045] "fer12_11"    "fer12_12"    "fer12_13"
[1048] "fer12_14"    "fer13"       "fer14"
[1051] "fer15"       "fer16a"      "fer16b"
[1054] "fer16c"      "fer17"       "fer21"
[1057] "fer22"       "fer23"       "fer24"
[1060] "fer25a"      "fer25b"      "fer25c"
[1063] "fer25d"      "fer25e"      "fer25f"
[1066] "fer26a"      "fer26b"      "fer26e"
[1069] "fer26f"      "fer26h"      "fer27a"
[1072] "fer27b"      "fer27c"      "fer28"
[1075] "fer29"       "hhd01a"      "hhd01b"
[1078] "hhd03_1"     "hhd03_2"     "hhd03_3"
[1081] "hhd03_4"     "hhd03_5"     "hhd03_6"
[1084] "hhd03_7"     "hhd03_8"     "hhd03_9"
[1087] "hhd03_10"    "hhd03_11"    "hhd03_12"
[1090] "hhd03_13"    "hhd03_14"    "hhd03_15"
[1093] "hhd03_16"    "hhd03_17"    "hhd03_18"
[1096] "hhd03_19"    "hhd03_20"    "hhd04_1"
[1099] "hhd04_2"     "hhd04_3"     "hhd04_4"
[1102] "hhd04_5"     "hhd04_6"     "hhd04_7"
[1105] "hhd04_8"     "hhd04_9"     "hhd04_10"
[1108] "hhd04_11"    "hhd04_12"    "hhd04_13"
[1111] "hhd04_14"    "hhd04_15"    "hhd04_16"
[1114] "hhd04_17"    "hhd04_18"    "hhd04_19"
[1117] "hhd04_20"    "hhd05_1"     "hhd05_2"
[1120] "hhd05_3"     "hhd05_4"     "hhd05_5"
[1123] "hhd05_6"     "hhd05_7"     "hhd05_8"
[1126] "hhd05_9"     "hhd05_10"    "hhd05_11"
[1129] "hhd05_12"    "hhd05_13"    "hhd05_14"
[1132] "hhd05_15"    "hhd05_16"    "hhd05_17"
```

```
[1135] "hhd05_18"         "hhd05_19"         "hhd05_20"
[1138] "hhd06_m1"         "hhd06_m2"         "hhd06_m3"
[1141] "hhd06_m4"         "hhd06_m5"         "hhd06_m6"
[1144] "hhd06_m7"         "hhd06_m8"         "hhd06_m9"
[1147] "hhd06_m10"        "hhd06_m11"        "hhd06_m12"
[1150] "hhd06_m13"        "hhd06_m14"        "hhd06_m15"
[1153] "hhd06_m16"        "hhd06_m17"        "hhd06_m18"
[1156] "hhd06_m19"        "hhd06_m20"        "hhd06_y1"
[1159] "hhd06_y2"         "hhd06_y3"         "hhd06_y4"
[1162] "hhd06_y5"         "hhd06_y6"         "hhd06_y7"
[1165] "hhd06_y8"         "hhd06_y9"         "hhd06_y10"
[1168] "hhd06_y11"        "hhd06_y12"        "hhd06_y13"
[1171] "hhd06_y14"        "hhd06_y15"        "hhd06_y16"
[1174] "hhd06_y17"        "hhd06_y18"        "hhd06_y19"
[1177] "hhd06_y20"        "hhd07_1"          "hhd07_2"
[1180] "hhd07_3"          "hhd07_4"          "hhd07_5"
[1183] "hhd07_6"          "hhd07_7"          "hhd07_8"
[1186] "hhd07_9"          "hhd07_10"         "hhd07_11"
[1189] "hhd07_12"         "hhd07_13"         "hhd07_14"
[1192] "hhd07_15"         "hhd07_16"         "hhd07_17"
[1195] "hhd07_18"         "hhd07_19"         "hhd07_20"
[1198] "hhd08_1"          "hhd08_2"          "hhd08_3"
[1201] "hhd08_4"          "hhd08_5"          "hhd08_6"
[1204] "hhd08_7"          "hhd08_8"          "hhd08_9"
[1207] "hhd08_10"         "hhd08_11"         "hhd08_12"
[1210] "hhd08_13"         "hhd08_14"         "hhd08_15"
[1213] "hhd08_16"         "hhd08_17"         "hhd08_18"
[1216] "hhd08_19"         "hhd08_20"         "hhd09_1"
[1219] "hhd09_2"          "hhd09_3"          "hhd09_4"
[1222] "hhd09_5"          "hhd09_6"          "hhd09_7"
[1225] "hhd09_8"          "hhd09_9"          "hhd09_10"
[1228] "hhd09_11"         "hhd09_12"         "hhd09_13"
[1231] "hhd09_14"         "hhd09_15"         "hhd09_16"
[1234] "hhd09_17"         "hhd09_18"         "hhd09_19"
[1237] "hhd09_20"         "hhd11a"           "hhd11b"
[1240] "hhd11c"           "hhd11d"           "hhd11e"
[1243] "hhd11f"           "hhd12"            "hhd13a"
[1246] "hhd13b"           "hhd13c"           "hhd13d"
[1249] "hhd13e"           "hhd14"            "hhd15a"
[1252] "hhd15b"           "hhd15c"           "hhd15d"
[1255] "hhd16"            "hhd17"            "hhd18"
[1258] "hhd19_1"          "hhd19_2"          "hhd19_3"
[1261] "hhd19_4"          "hhd19_5"          "hhd19_6"
```

```
[1264] "hhd19_7"      "hhd19_8"      "hhd19_9"
[1267] "hhd19_10"     "hhd19_11"     "hhd19_12"
[1270] "hhd19_13"     "hhd19_14"     "hhd19_15"
[1273] "hhd19_16"     "hhd19_17"     "hhd19_18"
[1276] "hhd19_19"     "hhd19_20"     "hhd19_21"
[1279] "hhd19_22"     "hhd20"        "hhd20u"
[1282] "hhd21"        "hhd22"        "hhd23_1"
[1285] "hhd23_2"      "hhd23_3"      "hhd23_4"
[1288] "hhd23_5"      "hhd23_6"      "hhd24"
[1291] "hhd24a"       "hhd24u"       "hhd25"
[1294] "hhd26_1"      "hhd26_2"      "hhd26_3"
[1297] "hhd26_4"      "hhd26_5"      "hhd26_6"
[1300] "hhd26_7"      "hhd26_8"      "hhd26_9"
[1303] "hhd26_10"     "hhd26_11"     "hhd26_12"
[1306] "hhd26_13"     "hhd26_14"     "hhd26_15"
[1309] "hhd26_16"     "hhd26_17"     "hhd26_18"
[1312] "hhd26_19"     "hhd26_20"     "hhd26_21"
[1315] "hhd26_22"     "hhd27"        "hhd27u"
[1318] "hhd28"        "hhd29_1"      "hhd29_2"
[1321] "hhd29_3"      "hhd29_4"      "hhd29_5"
[1324] "hhd29_6"      "hhd29_7"      "hhd29_8"
[1327] "hhd29_9"      "hhd29_10"     "hhd29_11"
[1330] "hhd29_12"     "hhd29_13"     "hhd29_14"
[1333] "hhd29_15"     "hhd29_16"     "hhd29_17"
[1336] "hhd29_18"     "hhd29_19"     "hhd29_20"
[1339] "hhd29_21"     "hhd29_22"     "hhd30"
[1342] "hhd30u"       "hhd31"        "hhd35"
[1345] "hhd36_1"      "hhd36_2"      "hhd36_3"
[1348] "hhd36_4"      "hhd36_5"      "hhd36_6"
[1351] "hhd36_7"      "hhd36_8"      "hhd36_9"
[1354] "hhd36_10"     "hhd36_11"     "hhd36_12"
[1357] "hhd36_13"     "hhd36_14"     "hhd36_15"
[1360] "hhd36_16"     "hhd36_17"     "hhd36_18"
[1363] "hhd36_19"     "hhd36_20"     "hhd36_21"
[1366] "hhd36_22"     "gen01"        "gen02"
[1369] "gen03"        "gen09m"       "gen09y"
[1372] "gen10m"       "gen10y"       "gen11"
[1375] "gen12iso"     "gen15a"       "gen15au"
[1378] "gen15b"       "gen15bu"      "gen16"
[1381] "gen23m"       "gen23y"       "gen24m"
[1384] "gen24y"       "gen25"        "gen26iso"
[1387] "gen29a"       "gen29au"      "gen29b"
[1390] "gen29bu"      "gen30"        "gen37a"
```

```
[1393] "gen37m"      "gen37y"      "gen38a"
[1396] "gen38bm"     "gen38by"     "gen39a"
[1399] "gen39b"      "gen40"       "gen41a"
[1402] "gen41a_4001" "gen41b"      "gen41b_4001"
[1405] "gen42"       "gen43"       "gen44aiso"
[1408] "gen44b"      "gen45"       "gen46"
[1411] "gen47"       "gen48"       "gen48isco"
[1414] "gen49"       "gen49isced"  "gen50"
[1417] "gen50isco"   "gen51"       "gen51isced"
[1420] "gen52"       "gen52am"     "gen52ay"
[1423] "gen53"       "gen54"       "gen55"
[1426] "gen56"       "gen57m"      "gen57y"
[1429] "gen58"       "gen59"       "gen60_1"
[1432] "gen60_2"     "gen60_3"     "gen60_4"
[1435] "gen60_5"     "gen60_6"     "gen60_7"
[1438] "gen60_8"     "gen60_9"     "gen60_10"
[1441] "gen60_11"    "gen60_12"    "gen60_13"
[1444] "gen60_14"    "gen60_15"    "gen60_16"
[1447] "gen60_17"    "gen60_18"    "gen60_19"
[1450] "gen60_20"    "gen60_21"    "gen60_22"
[1453] "gen63"       "gen66"       "gen67_1"
[1456] "gen67_2"     "gen67_3"     "gen67_4"
[1459] "gen67_5"     "gen67_6"     "gen67_7"
[1462] "gen67_8"     "gen67_9"     "gen67_10"
[1465] "gen67_11"    "gen67_12"    "gen67_13"
[1468] "gen67_14"    "gen67_15"    "gen67_16"
[1471] "gen67_17"    "gen67_18"    "gen67_19"
[1474] "gen67_20"    "gen67_21"    "gen67_22"
[1477] "gen68"       "gen69_1"     "gen69_2"
[1480] "gen69_3"     "gen69_4"     "gen69_5"
[1483] "gen69_6"     "gen69_7"     "gen69_8"
[1486] "gen69_9"     "gen69_10"    "gen69_11"
[1489] "gen69_12"    "gen69_13"    "gen69_14"
[1492] "gen69_15"    "gen69_16"    "gen69_17"
[1495] "gen69_18"    "gen69_19"    "gen69_20"
[1498] "gen69_21"    "gen69_22"    "gen70"
[1501] "gen71_1"     "gen71_2"     "gen71_3"
[1504] "gen71_4"     "gen71_5"     "gen71_6"
[1507] "gen71_7"     "gen71_8"     "gen71_9"
[1510] "gen71_10"    "gen71_11"    "gen71_12"
[1513] "gen71_13"    "gen71_14"    "gen71_15"
[1516] "gen71_16"    "gen71_17"    "gen71_18"
[1519] "gen71_19"    "gen71_20"    "gen71_21"
```

```
[1522] "gen71_22"         "wel01"            "wel02"
[1525] "wel02a"           "wel03_1"          "wel03_2"
[1528] "wel03_3"          "wel03_4"          "wel03_5"
[1531] "wel03_6"          "wel03_7"          "wel03_8"
[1534] "wel03_9"          "wel03_10"         "wel03_11"
[1537] "wel03_12"         "wel03_13"         "wel03_14"
[1540] "wel03_15"         "wel03_16"         "wel03_17"
[1543] "wel03_18"         "wel03_19"         "wel03_20"
[1546] "wel04"            "wel05"            "wel06"
[1549] "wel07"            "wel08"            "wel09a"
[1552] "wel09b"           "wel09c"           "wel09d"
[1555] "wel09e"           "wel09f"           "wel10_1"
[1558] "wel10_2"          "wel10_3"          "wel10_4"
[1561] "wel10_5"          "wel10_6"          "wel10_7"
[1564] "wel10_8"          "wel10_9"          "wel10_10"
[1567] "wel10_11"         "wel10_12"         "wel10_13"
[1570] "wel10_14"         "wel10_15"         "wel10_16"
[1573] "wel10_17"         "wel10_18"         "wel10_19"
[1576] "wel10_20"         "wel10_21"         "wel10_22"
[1579] "wel11a"           "wel11b"           "wel11c"
[1582] "wel11d"           "wel11e"           "wel14a_4001"
[1585] "wel14b_4001"      "wel14c_4001"      "wel14d_4001"
[1588] "wel14e_4001"      "wel14f_4001"      "wel14g_4001"
[1591] "wel16a_4001"      "wel16b_4001"      "wel16c_1_4001"
[1594] "wel16c_2_4001"    "wel16c_3_4001"    "wel16c_4_4001"
[1597] "wel16c_5_4001"    "wel16c_6_4001"    "wel16c_7_4001"
[1600] "wel16c_8_4001"    "wel16c_9_4001"    "wel16c_10_4001"
[1603] "wel16c_11_4001"   "wel16c_12_4001"   "wel16c_13_4001"
[1606] "wel16c_14_4001"   "wel16c_15_4001"   "wel16c_16_4001"
[1609] "wel16c_17_4001"   "wel16c_18_4001"   "wel16c_19_4001"
[1612] "wel16c_20_4001"   "wel16d_m1_4001"   "wel16d_m2_4001"
[1615] "wel16d_m3_4001"   "wel16d_m4_4001"   "wel16d_m5_4001"
[1618] "wel16d_m6_4001"   "wel16d_m7_4001"   "wel16d_m8_4001"
[1621] "wel16d_m9_4001"   "wel16d_m10_4001"  "wel16d_m11_4001"
[1624] "wel16d_m12_4001"  "wel16d_m13_4001"  "wel16d_m14_4001"
[1627] "wel16d_m15_4001"  "wel16d_m16_4001"  "wel16d_m17_4001"
[1630] "wel16d_m18_4001"  "wel16d_m19_4001"  "wel16d_m20_4001"
[1633] "wel16d_y1_4001"   "wel16d_y2_4001"   "wel16d_y3_4001"
[1636] "wel16d_y4_4001"   "wel16d_y5_4001"   "wel16d_y6_4001"
[1639] "wel16d_y7_4001"   "wel16d_y8_4001"   "wel16d_y9_4001"
[1642] "wel16d_y10_4001"  "wel16d_y11_4001"  "wel16d_y12_4001"
[1645] "wel16d_y13_4001"  "wel16d_y14_4001"  "wel16d_y15_4001"
[1648] "wel16d_y16_4001"  "wel16d_y17_4001"  "wel16d_y18_4001"
```

```
[1651] "wel16d_y19_4001"   "wel16d_y20_4001"   "wel16a_1_4002"
[1654] "wel16a_2_4002"     "wel16a_3_4002"     "wel16a_4_4002"
[1657] "wel16a_5_4002"     "wel16a_6_4002"     "wel16a_7_4002"
[1660] "wel16a_8_4002"     "wel16a_9_4002"     "wel16a_10_4002"
[1663] "wel16a_11_4002"    "wel16a_12_4002"    "wel16a_13_4002"
[1666] "wel16a_14_4002"    "wel16a_15_4002"    "wel16a_16_4002"
[1669] "wel16a_17_4002"    "wel16a_18_4002"    "wel16a_19_4002"
[1672] "wel16a_20_4002"    "wel16b_1_4002"     "wel16b_2_4002"
[1675] "wel16b_3_4002"     "wel16b_4_4002"     "wel16b_5_4002"
[1678] "wel16b_6_4002"     "wel16b_7_4002"     "wel16b_8_4002"
[1681] "wel16b_9_4002"     "wel16b_10_4002"    "wel16b_11_4002"
[1684] "wel16b_12_4002"    "wel16b_13_4002"    "wel16b_14_4002"
[1687] "wel16b_15_4002"    "wel16b_16_4002"    "wel16b_17_4002"
[1690] "wel16b_18_4002"    "wel16b_19_4002"    "wel16b_20_4002"
[1693] "wel16c_1_4002"     "wel16c_2_4002"     "wel16c_3_4002"
[1696] "wel16c_4_4002"     "wel16c_5_4002"     "wel16c_6_4002"
[1699] "wel16c_7_4002"     "wel16c_8_4002"     "wel16c_9_4002"
[1702] "wel16c_10_4002"    "wel16c_11_4002"    "wel16c_12_4002"
[1705] "wel16c_13_4002"    "wel16c_14_4002"    "wel16c_15_4002"
[1708] "wel16c_16_4002"    "wel16c_17_4002"    "wel16c_18_4002"
[1711] "wel16c_19_4002"    "wel16c_20_4002"    "wel16d_1_4002"
[1714] "wel16d_2_4002"     "wel16d_3_4002"     "wel16d_4_4002"
[1717] "wel16d_5_4002"     "wel16d_6_4002"     "wel16d_7_4002"
[1720] "wel16d_8_4002"     "wel16d_9_4002"     "wel16d_10_4002"
[1723] "wel16d_11_4002"    "wel16d_12_4002"    "wel16d_13_4002"
[1726] "wel16d_14_4002"    "wel16d_15_4002"    "wel16d_16_4002"
[1729] "wel16d_17_4002"    "wel16d_18_4002"    "wel16d_19_4002"
[1732] "wel16d_20_4002"    "wel16e_1_4002"     "wel16e_2_4002"
[1735] "wel16e_3_4002"     "wel16e_4_4002"     "wel16e_5_4002"
[1738] "wel16e_6_4002"     "wel16e_7_4002"     "wel16e_8_4002"
[1741] "wel16e_9_4002"     "wel16e_10_4002"    "wel16e_11_4002"
[1744] "wel16e_12_4002"    "wel16e_13_4002"    "wel16e_14_4002"
[1747] "wel16e_15_4002"    "wel16e_16_4002"    "wel16e_17_4002"
[1750] "wel16e_18_4002"    "wel16e_19_4002"    "wel16e_20_4002"
[1753] "wel16f_1_4002"     "wel16f_2_4002"     "wel16f_3_4002"
[1756] "wel16f_4_4002"     "wel16f_5_4002"     "wel16f_6_4002"
[1759] "wel16f_7_4002"     "wel16f_8_4002"     "wel16f_9_4002"
[1762] "wel16f_10_4002"    "wel16f_11_4002"    "wel16f_12_4002"
[1765] "wel16f_13_4002"    "wel16f_14_4002"    "wel16f_15_4002"
[1768] "wel16f_16_4002"    "wel16f_17_4002"    "wel16f_18_4002"
[1771] "wel16f_19_4002"    "wel16f_20_4002"    "wel16g_1_4002"
[1774] "wel16g_2_4002"     "wel16g_3_4002"     "wel16g_4_4002"
[1777] "wel16g_5_4002"     "wel16g_6_4002"     "wel16g_7_4002"
```

```
[1780] "wel16g_8_4002"     "wel16g_9_4002"     "wel16g_10_4002"
[1783] "wel16g_11_4002"    "wel16g_12_4002"    "wel16g_13_4002"
[1786] "wel16g_14_4002"    "wel16g_15_4002"    "wel16g_16_4002"
[1789] "wel16g_17_4002"    "wel16g_18_4002"    "wel16g_19_4002"
[1792] "wel16g_20_4002"    "wrk01"             "wrk02"
[1795] "wrk03m"            "wrk03y"            "wrk04"
[1798] "wrk04isco"         "wrk06"             "wrk07"
[1801] "wrk08"             "wrk09"             "wrk10"
[1804] "wrk11"             "wrk12"             "wrk13"
[1807] "wrk14"             "wrk15a"            "wrk15b"
[1810] "wrk15c"            "wrk15d"            "wrk16a"
[1813] "wrk16b"            "wrk17"             "wrk18"
[1816] "wrk20"             "wrk21"             "wrk22"
[1819] "wrk23"             "wrk24"             "wrk25"
[1822] "wrk26"             "wrk27"             "wrk27isco"
[1825] "wrk28"             "wrk30"             "wrk30am"
[1828] "wrk30ay"           "wrk31"             "wrk32"
[1831] "wrk34"             "wrk34isco"         "wrk35"
[1834] "wrk36"             "wrk37"             "wrk38"
[1837] "wrk39"             "wrk40"             "wrk41"
[1840] "wrk42"             "wrk43"             "wrk44"
[1843] "wrk46"             "wrk47"             "wrk48"
[1846] "wrk49"             "wrk50"             "wrk51_4001"
[1849] "wrk51_4002"        "wrk51_4003"        "wrk51_4004"
[1852] "wrk51a_4005"       "wrk51b_4005"       "wrk51_4006"
[1855] "wrk51a_4007"       "wrk51b_4007"       "wrk51a_4008"
[1858] "wrk51b_4008"       "wrk51_4009"        "wrk51_4010"
[1861] "wrk51a_4011"       "wrk51b_4011"       "wrk51_4012"
[1864] "wrk51_4013"        "wrk51_4014"        "wrk51_4015"
[1867] "inc01"             "inc03"             "inc05"
[1870] "inc06"             "inc08_1"           "inc08_2"
[1873] "inc08_3"           "inc08_4"           "inc08_5"
[1876] "inc08_6"           "inc08_7"           "inc08_8"
[1879] "inc08_9"           "inc08_10"          "inc08_11"
[1882] "inc08_12"          "inc09_1"           "inc09_2"
[1885] "inc09_3"           "inc09_4"           "inc09_5"
[1888] "inc09_6"           "inc09_7"           "inc09_8"
[1891] "inc09_9"           "inc09_10"          "inc09_11"
[1894] "inc11_1"           "inc11_2"           "inc11_3"
[1897] "inc11_4"           "inc11_5"           "inc11_6"
[1900] "inc11_7"           "inc11_8"           "inc11_9"
[1903] "inc11_10"          "inc11_11"          "inc12"
[1906] "inc13"             "inc14_1"           "inc14_2"
```

```
[1909] "inc14_3"          "inc14_4"          "inc14_5"
[1912] "inc14_6"          "inc14_7"          "inc14_8"
[1915] "inc14_9"          "inc14_10"         "inc14_11"
[1918] "inc14_12"         "inc14_13"         "inc14_14"
[1921] "inc14_15"         "inc14_16"         "inc14_17"
[1924] "inc14_18"         "inc14_19"         "inc14_20"
[1927] "inc14_21"         "inc14_22"         "inc15"
[1930] "att01"            "att02"            "att03a"
[1933] "att03b"           "att03d"           "att03e"
[1936] "att03g"           "att03h"           "att03i"
[1939] "att03j"           "att05b"           "att06a"
[1942] "att06b"           "att07a"           "att07b"
[1945] "att07c"           "att07d"           "att07g"
[1948] "att08"            "att09"            "att09u"
[1951] "att10"            "att11b"           "att11d"
[1954] "att13a_4001"      "att13b_4001"      "att13c_4001"
[1957] "att13d_4001"      "att13e_4001"      "att13f_4001"
[1960] "att13g_4001"      "att13h_4001"      "att13_4002"
[1963] "att13_4003"       "att13_4004"       "att13_4005"
[1966] "att13_1_4006"     "att13_2_4006"     "att13_3_4006"
[1969] "att13_4_4006"     "att13_5_4006"     "att13_6_4006"
[1972] "att13_7_4006"     "att13_8_4006"     "att13_9_4006"
[1975] "att13_4007"       "att19a_4001"      "att19b_4001"
[1978] "att19c_4001"      "rep01"            "rep02"
[1981] "rep03_1"          "rep03_2"          "rep03_3"
[1984] "rep03_4"          "rep04"            "rep05"
[1987] "rep06"            "flag1"            "localitysize_4001"
[1990] "department_4001"  "city_4001"
```

```r
head(encuesta_generacion) # muestra las primeras 6 líneas
```

```
# A tibble: 6 x 1,991
  country    region     respid intid mode    weight instrument intdatem intdatey
  <dbl+lbl>  <dbl+lbl>  <chr>  <chr> <dbl+l>  <dbl> <chr>      <dbl+lb>    <dbl>
1 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  1.37  GGP UY     11 [Nov~     2021
2 40 [Urugu~   NA       URAA0~ ""    2 [Web] NA     GGP UY     12 [Dec~     2021
3 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.522 GGP UY     12 [Dec~     2021
4 40 [Urugu~ 4001 [Mon~ URAA0~ ""    2 [Web]  1.17  GGP UY     12 [Dec~     2021
5 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.636 GGP UY      2 [Feb~     2022
6 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.200 GGP UY      2 [Feb~     2022
# i 1,982 more variables: dem01 <dbl+lbl>, dem02m <dbl+lbl>, dem02y <dbl+lbl>,
```

```
#   dem03 <dbl+lbl>, dem04a <dbl+lbl>, dem04biso <dbl+lbl>, dem05m <dbl+lbl>,
#   dem05y <dbl+lbl>, dem06 <dbl+lbl>, dem07 <dbl+lbl>, dem07isced <dbl+lbl>,
#   dem08m <dbl+lbl>, dem08y <dbl+lbl>, dem09 <dbl+lbl>, dem10m <dbl+lbl>,
#   dem10y <dbl+lbl>, dem11 <dbl+lbl>, dem12 <dbl+lbl>, dem14 <dbl+lbl>,
#   dem15 <dbl+lbl>, dem17 <dbl+lbl>, dem18 <dbl+lbl>, dem19 <dbl+lbl>,
#   dem20 <dbl+lbl>, dem21 <dbl+lbl>, dem22a <dbl+lbl>, dem22m <dbl+lbl>, ...
```

```
table(encuesta_generacion$dem01) # un tabulado simple
```

```

   1    2 4001 
2608 4575    9 
```

1.9.2 Revisión con {skimr}

Esto se puede tardar un poquito

```
skimr::skim(encuesta_generacion[, 1:20])
```

Table 1.1: Data summary

Name	encuesta_generacion[, 1:2...
Number of rows	7192
Number of columns	20
Column type frequency:	
character	3
numeric	17
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
respid	0	1	9	9	0	7192	0
intid	0	1	0	9	981	593	0
instrument	0	1	6	6	0	1	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
country	0	1.00	40.00	0.00	40.0	40.00	40.00	40.00	40	
region	99	0.99	4002.37	2.04	4001.0	4001.00	4001.00	4003.00	4007	
mode	0	1.00	1.14	0.34	1.0	1.00	1.00	1.00	2	
weight	174	0.98	1.00	0.80	0.2	0.45	0.79	1.27	5	
intdatem	0	1.00	8.15	2.94	1.0	6.00	8.00	11.00	12	
intdatey	0	1.00	2021.80	0.40	2021.0	2022.00	2022.00	2022.00	2022	
dem01	0	1.00	6.64	141.40	1.0	1.00	2.00	2.00	4001	
dem02m	21	1.00	6.57	3.39	1.0	4.00	7.00	9.00	12	
dem02y	0	1.00	1972.60	16.83	1942.0	1958.00	1972.00	1987.00	2004	
dem03	0	1.00	1.03	0.18	1.0	1.00	1.00	1.00	2	
dem04a	245	0.97	4009.81	4.08	4001.0	4009.00	4010.00	4011.00	4019	
dem04biso	6953	0.03	289.11	323.40	32.0	32.00	76.00	600.00	862	
dem05m	7006	0.03	6.03	3.69	1.0	3.00	6.00	9.00	12	
dem05y	6956	0.03	1994.95	23.30	1921.0	1980.75	1998.50	2017.00	2022	
dem06	10	1.00	4.04	2.47	1.0	2.00	3.00	6.00	12	
dem07	24	1.00	2.90	1.77	0.0	2.00	3.00	4.00	8	
dem07isced	24	1.00	2.90	1.77	0.0	2.00	3.00	4.00	8	

1.10 Un poquito de {dplyr} y limpieza

1.10.1 Primero, los pipes

R utiliza dos pipes el nativo `|>` y el pipe que está en `{dplyr}` `%>%`. Algunas de las diferencias las puedes checar acá https://eliocamp.github.io/codigo-r/2021/05/r-pipa-nativa/

Aquí hay un *tuit*, o *post de x.com* que lo explica bien.

https://x.com/ArthurWelle/status/1535429654760284161

En estas prácticas utilizaremos el segundo, son muy parecidos y así esta instructora pueda reciclar algunos de sus códigos viejos. Pero funcionan igual:

```
encuesta_generacion|> #pipe nativo, no necesita instalación
  head()
```

```
# A tibble: 6 x 1,991
  country     region      respid intid mode     weight instrument intdatem intdatey
  <dbl+lbl>   <dbl+lbl>   <chr>  <chr> <dbl+l>  <dbl>  <chr>      <dbl+lb>    <dbl>
```

```
1 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  1.37  GGP UY     11 [Nov~     2021
2 40 [Urugu~   NA       URAA0~ ""    2 [Web] NA     GGP UY     12 [Dec~     2021
3 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.522 GGP UY     12 [Dec~     2021
4 40 [Urugu~ 4001 [Mon~ URAA0~ ""    2 [Web]  1.17  GGP UY     12 [Dec~     2021
5 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.636 GGP UY      2 [Feb~     2022
6 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.200 GGP UY      2 [Feb~     2022
# i 1,982 more variables: dem01 <dbl+lbl>, dem02m <dbl+lbl>, dem02y <dbl+lbl>,
#   dem03 <dbl+lbl>, dem04a <dbl+lbl>, dem04biso <dbl+lbl>, dem05m <dbl+lbl>,
#   dem05y <dbl+lbl>, dem06 <dbl+lbl>, dem07 <dbl+lbl>, dem07isced <dbl+lbl>,
#   dem08m <dbl+lbl>, dem08y <dbl+lbl>, dem09 <dbl+lbl>, dem10m <dbl+lbl>,
#   dem10y <dbl+lbl>, dem11 <dbl+lbl>, dem12 <dbl+lbl>, dem14 <dbl+lbl>,
#   dem15 <dbl+lbl>, dem17 <dbl+lbl>, dem18 <dbl+lbl>, dem19 <dbl+lbl>,
#   dem20 <dbl+lbl>, dem21 <dbl+lbl>, dem22a <dbl+lbl>, dem22m <dbl+lbl>, ...
```

```
encuesta_generacion %>%  #pipe de dplyr, necesita instalación de dplyr en tidyverse
  head()
```

```
# A tibble: 6 x 1,991
  country    region     respid intid mode    weight instrument intdatem intdatey
  <dbl+lbl>  <dbl+lbl>  <chr>  <chr> <dbl+l>  <dbl> <chr>      <dbl+lb>    <dbl>
1 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  1.37  GGP UY     11 [Nov~     2021
2 40 [Urugu~   NA       URAA0~ ""    2 [Web] NA     GGP UY     12 [Dec~     2021
3 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.522 GGP UY     12 [Dec~     2021
4 40 [Urugu~ 4001 [Mon~ URAA0~ ""    2 [Web]  1.17  GGP UY     12 [Dec~     2021
5 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.636 GGP UY      2 [Feb~     2022
6 40 [Urugu~ 4001 [Mon~ URAA0~ "URU~ 1 [Fac~  0.200 GGP UY      2 [Feb~     2022
# i 1,982 more variables: dem01 <dbl+lbl>, dem02m <dbl+lbl>, dem02y <dbl+lbl>,
#   dem03 <dbl+lbl>, dem04a <dbl+lbl>, dem04biso <dbl+lbl>, dem05m <dbl+lbl>,
#   dem05y <dbl+lbl>, dem06 <dbl+lbl>, dem07 <dbl+lbl>, dem07isced <dbl+lbl>,
#   dem08m <dbl+lbl>, dem08y <dbl+lbl>, dem09 <dbl+lbl>, dem10m <dbl+lbl>,
#   dem10y <dbl+lbl>, dem11 <dbl+lbl>, dem12 <dbl+lbl>, dem14 <dbl+lbl>,
#   dem15 <dbl+lbl>, dem17 <dbl+lbl>, dem18 <dbl+lbl>, dem19 <dbl+lbl>,
#   dem20 <dbl+lbl>, dem21 <dbl+lbl>, dem22a <dbl+lbl>, dem22m <dbl+lbl>, ...
```

1.10.2 Limpieza de nombres con `{janitor}`

Este paso también nos permitirá enseñar otro *pipe* que está en el paquete `{magrittr}`.

Los nombres de una base de datos son los nombres de las columnas.

```
names(encuesta_generacion)
```

```
  [1] "country"           "region"            "respid"
  [4] "intid"             "mode"              "weight"
  [7] "instrument"        "intdatem"          "intdatey"
 [10] "dem01"             "dem02m"            "dem02y"
 [13] "dem03"             "dem04a"            "dem04biso"
 [16] "dem05m"            "dem05y"            "dem06"
 [19] "dem07"             "dem07isced"        "dem08m"
 [22] "dem08y"            "dem09"             "dem10m"
 [25] "dem10y"            "dem11"             "dem12"
 [28] "dem14"             "dem15"             "dem17"
 [31] "dem18"             "dem19"             "dem20"
 [34] "dem21"             "dem22a"            "dem22m"
 [37] "dem22y"            "dem23"             "dem24a"
 [40] "dem24biso"         "dem24em"           "dem24ey"
 [43] "dem25"             "dem25isced"        "dem26"
 [46] "dem27"             "dem28a"            "dem28bm"
 [49] "dem28by"           "dem28c"            "dem30a"
 [52] "dem30bm"           "dem30by"           "dem30c"
 [55] "dem30d"            "dem31m"            "dem31y"
 [58] "dem32a"            "dem32b"            "dem32c"
 [61] "dem32d"            "dem33"             "dem33am"
 [64] "dem33ay"           "dem34m"            "dem34y"
 [67] "dem35"             "dem36a"            "dem36au"
 [70] "dem36b"            "dem36bu"           "dem37"
 [73] "dem38a"            "dem38b"            "dem38c"
 [76] "dem38d"            "dem38e"            "dem38f"
 [79] "dem38g"            "dem39a"            "dem39b"
 [82] "dem39c"            "dem39d"            "dem40"
 [85] "dem41"             "dem42"             "dem43"
 [88] "dem44"             "dem45"             "dem46"
 [91] "lhi01"             "lhi02"             "lhi04_m1"
 [94] "lhi04_m2"          "lhi04_m3"          "lhi04_m4"
 [97] "lhi04_m5"          "lhi04_m6"          "lhi04_m7"
[100] "lhi04_m8"          "lhi04_m9"          "lhi04_m10"
[103] "lhi04_m11"         "lhi04_m12"         "lhi04_m13"
[106] "lhi04_m14"         "lhi04_m15"         "lhi04_m16"
[109] "lhi04_m17"         "lhi04_m18"         "lhi04_m19"
[112] "lhi04_m20"         "lhi04_y1"          "lhi04_y2"
[115] "lhi04_y3"          "lhi04_y4"          "lhi04_y5"
[118] "lhi04_y6"          "lhi04_y7"          "lhi04_y8"
[121] "lhi04_y9"          "lhi04_y10"         "lhi04_y11"
[124] "lhi04_y12"         "lhi04_y13"         "lhi04_y14"
[127] "lhi04_y15"         "lhi04_y16"         "lhi04_y17"
```

```
[130] "lhi04_y18"          "lhi04_y19"          "lhi04_y20"
[133] "lhi04a_1"           "lhi04a_2"           "lhi04a_3"
[136] "lhi04a_4"           "lhi04a_5"           "lhi04a_6"
[139] "lhi04a_7"           "lhi04a_8"           "lhi04a_9"
[142] "lhi04a_10"          "lhi04a_11"          "lhi04a_12"
[145] "lhi04a_13"          "lhi04a_14"          "lhi04a_15"
[148] "lhi04a_16"          "lhi04a_17"          "lhi04a_18"
[151] "lhi04a_19"          "lhi04a_20"          "lhi05a_1"
[154] "lhi05a_2"           "lhi05a_3"           "lhi05a_4"
[157] "lhi05a_5"           "lhi05a_6"           "lhi05a_7"
[160] "lhi05a_8"           "lhi05a_9"           "lhi05a_10"
[163] "lhi05a_11"          "lhi05a_12"          "lhi05a_13"
[166] "lhi05a_14"          "lhi05a_15"          "lhi05a_16"
[169] "lhi05a_17"          "lhi05a_18"          "lhi05a_19"
[172] "lhi05a_20"          "lhi05b_m1"          "lhi05b_m2"
[175] "lhi05b_m3"          "lhi05b_m4"          "lhi05b_m5"
[178] "lhi05b_m6"          "lhi05b_m7"          "lhi05b_m8"
[181] "lhi05b_m9"          "lhi05b_m10"         "lhi05b_m11"
[184] "lhi05b_m12"         "lhi05b_m13"         "lhi05b_m14"
[187] "lhi05b_m15"         "lhi05b_m16"         "lhi05b_m17"
[190] "lhi05b_m18"         "lhi05b_m19"         "lhi05b_m20"
[193] "lhi05b_y1"          "lhi05b_y2"          "lhi05b_y3"
[196] "lhi05b_y4"          "lhi05b_y5"          "lhi05b_y6"
[199] "lhi05b_y7"          "lhi05b_y8"          "lhi05b_y9"
[202] "lhi05b_y10"         "lhi05b_y11"         "lhi05b_y12"
[205] "lhi05b_y13"         "lhi05b_y14"         "lhi05b_y15"
[208] "lhi05b_y16"         "lhi05b_y17"         "lhi05b_y18"
[211] "lhi05b_y19"         "lhi05b_y20"         "lhi06_m1"
[214] "lhi06_m2"           "lhi06_m3"           "lhi06_m4"
[217] "lhi06_m5"           "lhi06_m6"           "lhi06_m7"
[220] "lhi06_m8"           "lhi06_m9"           "lhi06_m10"
[223] "lhi06_m11"          "lhi06_m12"          "lhi06_m13"
[226] "lhi06_m14"          "lhi06_m15"          "lhi06_m16"
[229] "lhi06_m17"          "lhi06_m18"          "lhi06_m19"
[232] "lhi06_m20"          "lhi06_y1"           "lhi06_y2"
[235] "lhi06_y3"           "lhi06_y4"           "lhi06_y5"
[238] "lhi06_y6"           "lhi06_y7"           "lhi06_y8"
[241] "lhi06_y9"           "lhi06_y10"          "lhi06_y11"
[244] "lhi06_y12"          "lhi06_y13"          "lhi06_y14"
[247] "lhi06_y15"          "lhi06_y16"          "lhi06_y17"
[250] "lhi06_y18"          "lhi06_y19"          "lhi06_y20"
[253] "lhi07_1"            "lhi07_2"            "lhi07_3"
[256] "lhi07_4"            "lhi07_5"            "lhi07_6"
```

```
[259] "lhi07_7"            "lhi07_8"            "lhi07_9"
[262] "lhi07_10"           "lhi07_11"           "lhi07_12"
[265] "lhi07_13"           "lhi07_14"           "lhi07_15"
[268] "lhi07_16"           "lhi07_17"           "lhi07_18"
[271] "lhi07_19"           "lhi07_20"           "lhi08_1"
[274] "lhi08_2"            "lhi08_3"            "lhi08_4"
[277] "lhi08_5"            "lhi08_6"            "lhi08_7"
[280] "lhi08_8"            "lhi08_9"            "lhi08_10"
[283] "lhi08_11"           "lhi08_12"           "lhi08_13"
[286] "lhi08_14"           "lhi08_15"           "lhi08_16"
[289] "lhi08_17"           "lhi08_18"           "lhi08_19"
[292] "lhi08_20"           "lhi09_1"            "lhi09_2"
[295] "lhi09_3"            "lhi09_4"            "lhi09_5"
[298] "lhi09_6"            "lhi09_7"            "lhi09_8"
[301] "lhi09_9"            "lhi09_10"           "lhi09_11"
[304] "lhi09_12"           "lhi09_13"           "lhi09_14"
[307] "lhi09_15"           "lhi09_16"           "lhi09_17"
[310] "lhi09_18"           "lhi09_19"           "lhi09_20"
[313] "lhi10_1"            "lhi10_2"            "lhi10_3"
[316] "lhi10_4"            "lhi10_5"            "lhi10_6"
[319] "lhi10_7"            "lhi10_8"            "lhi10_9"
[322] "lhi10_10"           "lhi10_11"           "lhi10_12"
[325] "lhi10_13"           "lhi10_14"           "lhi10_15"
[328] "lhi10_16"           "lhi10_17"           "lhi10_18"
[331] "lhi10_19"           "lhi10_20"           "lhi11_1"
[334] "lhi11_2"            "lhi11_3"            "lhi11_4"
[337] "lhi11_5"            "lhi11_6"            "lhi11_7"
[340] "lhi11_8"            "lhi11_9"            "lhi11_10"
[343] "lhi11_11"           "lhi11_12"           "lhi11_13"
[346] "lhi11_14"           "lhi11_15"           "lhi11_16"
[349] "lhi11_17"           "lhi11_18"           "lhi11_19"
[352] "lhi11_20"           "lhi12_1"            "lhi12_2"
[355] "lhi12_3"            "lhi12_4"            "lhi12_5"
[358] "lhi12_6"            "lhi12_7"            "lhi12_8"
[361] "lhi12_9"            "lhi12_10"           "lhi12_11"
[364] "lhi12_12"           "lhi12_13"           "lhi12_14"
[367] "lhi12_15"           "lhi12_16"           "lhi12_17"
[370] "lhi12_18"           "lhi12_19"           "lhi12_20"
[373] "lhi13_1"            "lhi13_2"            "lhi13_3"
[376] "lhi13_4"            "lhi13_5"            "lhi13_6"
[379] "lhi13_7"            "lhi13_8"            "lhi13_9"
[382] "lhi13_10"           "lhi13_11"           "lhi13_12"
[385] "lhi13_13"           "lhi13_14"           "lhi13_15"
```

```
[388] "lhi13_16"       "lhi13_17"       "lhi13_18"
[391] "lhi13_19"       "lhi13_20"       "lhi14_m1"
[394] "lhi14_m2"       "lhi14_m3"       "lhi14_m4"
[397] "lhi14_m5"       "lhi14_m6"       "lhi14_m7"
[400] "lhi14_m8"       "lhi14_m9"       "lhi14_m10"
[403] "lhi14_m11"      "lhi14_m12"      "lhi14_m13"
[406] "lhi14_m14"      "lhi14_m15"      "lhi14_m16"
[409] "lhi14_m17"      "lhi14_m18"      "lhi14_m19"
[412] "lhi14_m20"      "lhi14_y1"       "lhi14_y2"
[415] "lhi14_y3"       "lhi14_y4"       "lhi14_y5"
[418] "lhi14_y6"       "lhi14_y7"       "lhi14_y8"
[421] "lhi14_y9"       "lhi14_y10"      "lhi14_y11"
[424] "lhi14_y12"      "lhi14_y13"      "lhi14_y14"
[427] "lhi14_y15"      "lhi14_y16"      "lhi14_y17"
[430] "lhi14_y18"      "lhi14_y19"      "lhi14_y20"
[433] "lhi15a_1"       "lhi15a_2"       "lhi15a_3"
[436] "lhi15a_4"       "lhi15a_5"       "lhi15a_6"
[439] "lhi15a_7"       "lhi15a_8"       "lhi15a_9"
[442] "lhi15a_10"      "lhi15a_11"      "lhi15a_12"
[445] "lhi15a_13"      "lhi15a_14"      "lhi15a_15"
[448] "lhi15a_16"      "lhi15a_17"      "lhi15a_18"
[451] "lhi15a_19"      "lhi15a_20"      "lhi15b_m1"
[454] "lhi15b_m2"      "lhi15b_m3"      "lhi15b_m4"
[457] "lhi15b_m5"      "lhi15b_m6"      "lhi15b_m7"
[460] "lhi15b_m8"      "lhi15b_m9"      "lhi15b_m10"
[463] "lhi15b_m11"     "lhi15b_m12"     "lhi15b_m13"
[466] "lhi15b_m14"     "lhi15b_m15"     "lhi15b_m16"
[469] "lhi15b_m17"     "lhi15b_m18"     "lhi15b_m19"
[472] "lhi15b_m20"     "lhi15b_y1"      "lhi15b_y2"
[475] "lhi15b_y3"      "lhi15b_y4"      "lhi15b_y5"
[478] "lhi15b_y6"      "lhi15b_y7"      "lhi15b_y8"
[481] "lhi15b_y9"      "lhi15b_y10"     "lhi15b_y11"
[484] "lhi15b_y12"     "lhi15b_y13"     "lhi15b_y14"
[487] "lhi15b_y15"     "lhi15b_y16"     "lhi15b_y17"
[490] "lhi15b_y18"     "lhi15b_y19"     "lhi15b_y20"
[493] "lhi16_1"        "lhi16_2"        "lhi16_3"
[496] "lhi16_4"        "lhi16_5"        "lhi16_6"
[499] "lhi16_7"        "lhi16_8"        "lhi16_9"
[502] "lhi16_10"       "lhi16_11"       "lhi16_12"
[505] "lhi16_13"       "lhi16_14"       "lhi16_15"
[508] "lhi16_16"       "lhi16_17"       "lhi16_18"
[511] "lhi16_19"       "lhi16_20"       "lhi17_1"
[514] "lhi17_2"        "lhi17_3"        "lhi17_4"
```

```
[517] "lhi17_5"      "lhi17_6"      "lhi17_7"
[520] "lhi17_8"      "lhi17_9"      "lhi17_10"
[523] "lhi17_11"     "lhi17_12"     "lhi17_13"
[526] "lhi17_14"     "lhi17_15"     "lhi17_16"
[529] "lhi17_17"     "lhi17_18"     "lhi17_19"
[532] "lhi17_20"     "lhi18"        "lhi19"
[535] "lhi20"        "lhi21"        "lhi22"
[538] "lhi23"        "lhi25_1"      "lhi25_2"
[541] "lhi25_3"      "lhi25_4"      "lhi25_5"
[544] "lhi25_6"      "lhi25_7"      "lhi25_8"
[547] "lhi25_9"      "lhi25_10"     "lhi25_11"
[550] "lhi25_12"     "lhi25_13"     "lhi25_14"
[553] "lhi25_15"     "lhi25_16"     "lhi25_17"
[556] "lhi25_18"     "lhi25_19"     "lhi25_20"
[559] "lhi26_1"      "lhi26_2"      "lhi26_3"
[562] "lhi26_4"      "lhi26_5"      "lhi26_6"
[565] "lhi26_7"      "lhi26_8"      "lhi26_9"
[568] "lhi26_10"     "lhi26_11"     "lhi26_12"
[571] "lhi26_13"     "lhi26_14"     "lhi26_15"
[574] "lhi26_16"     "lhi26_17"     "lhi26_18"
[577] "lhi26_19"     "lhi26_20"     "lhi27_1"
[580] "lhi27_2"      "lhi27_3"      "lhi27_4"
[583] "lhi27_5"      "lhi27_6"      "lhi27_7"
[586] "lhi27_8"      "lhi27_9"      "lhi27_10"
[589] "lhi27_11"     "lhi27_12"     "lhi27_13"
[592] "lhi27_14"     "lhi27_15"     "lhi27_16"
[595] "lhi27_17"     "lhi27_18"     "lhi27_19"
[598] "lhi27_20"     "lhi28_1"      "lhi28_2"
[601] "lhi28_3"      "lhi28_4"      "lhi28_5"
[604] "lhi28_6"      "lhi28_7"      "lhi28_8"
[607] "lhi28_9"      "lhi28_10"     "lhi28_11"
[610] "lhi28_12"     "lhi28_13"     "lhi28_14"
[613] "lhi28_15"     "lhi28_16"     "lhi28_17"
[616] "lhi28_18"     "lhi28_19"     "lhi28_20"
[619] "lhi29_m1"     "lhi29_m2"     "lhi29_m3"
[622] "lhi29_m4"     "lhi29_m5"     "lhi29_m6"
[625] "lhi29_m7"     "lhi29_m8"     "lhi29_m9"
[628] "lhi29_m10"    "lhi29_m11"    "lhi29_m12"
[631] "lhi29_m13"    "lhi29_m14"    "lhi29_m15"
[634] "lhi29_m16"    "lhi29_m17"    "lhi29_m18"
[637] "lhi29_m19"    "lhi29_m20"    "lhi29_y1"
[640] "lhi29_y2"     "lhi29_y3"     "lhi29_y4"
[643] "lhi29_y5"     "lhi29_y6"     "lhi29_y7"
```

```
[646] "lhi29_y8"         "lhi29_y9"         "lhi29_y10"
[649] "lhi29_y11"        "lhi29_y12"        "lhi29_y13"
[652] "lhi29_y14"        "lhi29_y15"        "lhi29_y16"
[655] "lhi29_y17"        "lhi29_y18"        "lhi29_y19"
[658] "lhi29_y20"        "lhi30_m1"         "lhi30_m2"
[661] "lhi30_m3"         "lhi30_m4"         "lhi30_m5"
[664] "lhi30_m6"         "lhi30_m7"         "lhi30_m8"
[667] "lhi30_m9"         "lhi30_m10"        "lhi30_m11"
[670] "lhi30_m12"        "lhi30_m13"        "lhi30_m14"
[673] "lhi30_m15"        "lhi30_m16"        "lhi30_m17"
[676] "lhi30_m18"        "lhi30_m19"        "lhi30_m20"
[679] "lhi30_y1"         "lhi30_y2"         "lhi30_y3"
[682] "lhi30_y4"         "lhi30_y5"         "lhi30_y6"
[685] "lhi30_y7"         "lhi30_y8"         "lhi30_y9"
[688] "lhi30_y10"        "lhi30_y11"        "lhi30_y12"
[691] "lhi30_y13"        "lhi30_y14"        "lhi30_y15"
[694] "lhi30_y16"        "lhi30_y17"        "lhi30_y18"
[697] "lhi30_y19"        "lhi30_y20"        "lhi31_1"
[700] "lhi31_2"          "lhi31_3"          "lhi31_4"
[703] "lhi31_5"          "lhi31_6"          "lhi31_7"
[706] "lhi31_8"          "lhi31_9"          "lhi31_10"
[709] "lhi31_11"         "lhi31_12"         "lhi31_13"
[712] "lhi31_14"         "lhi31_15"         "lhi31_16"
[715] "lhi31_17"         "lhi31_18"         "lhi31_19"
[718] "lhi31_20"         "lhi32_1"          "lhi32_2"
[721] "lhi32_3"          "lhi32_4"          "lhi32_5"
[724] "lhi32_6"          "lhi32_7"          "lhi32_8"
[727] "lhi32_9"          "lhi32_10"         "lhi32_11"
[730] "lhi32_12"         "lhi32_13"         "lhi32_14"
[733] "lhi32_15"         "lhi32_16"         "lhi32_17"
[736] "lhi32_18"         "lhi32_19"         "lhi32_20"
[739] "lhi33_1"          "lhi33_2"          "lhi33_3"
[742] "lhi33_4"          "lhi33_5"          "lhi33_6"
[745] "lhi33_7"          "lhi33_8"          "lhi33_9"
[748] "lhi33_10"         "lhi33_11"         "lhi33_12"
[751] "lhi33_13"         "lhi33_14"         "lhi33_15"
[754] "lhi33_16"         "lhi33_17"         "lhi33_18"
[757] "lhi33_19"         "lhi33_20"         "lhi33u_1"
[760] "lhi33u_2"         "lhi33u_3"         "lhi33u_4"
[763] "lhi33u_5"         "lhi33u_6"         "lhi33u_7"
[766] "lhi33u_8"         "lhi33u_9"         "lhi33u_10"
[769] "lhi33u_11"        "lhi33u_12"        "lhi33u_13"
[772] "lhi33u_14"        "lhi33u_15"        "lhi33u_16"
```

```
[775] "lhi33u_17"          "lhi33u_18"          "lhi33u_19"
[778] "lhi33u_20"          "lhi34_1"            "lhi34_2"
[781] "lhi34_3"            "lhi34_4"            "lhi34_5"
[784] "lhi34_6"            "lhi34_7"            "lhi34_8"
[787] "lhi34_9"            "lhi34_10"           "lhi34_11"
[790] "lhi34_12"           "lhi34_13"           "lhi34_14"
[793] "lhi34_15"           "lhi34_16"           "lhi34_17"
[796] "lhi34_18"           "lhi34_19"           "lhi34_20"
[799] "lhi35_1"            "lhi35_2"            "lhi35_3"
[802] "lhi35_4"            "lhi35_5"            "lhi35_6"
[805] "lhi35_7"            "lhi35_8"            "lhi35_9"
[808] "lhi35_10"           "lhi35_11"           "lhi35_12"
[811] "lhi35_13"           "lhi35_14"           "lhi35_15"
[814] "lhi35_16"           "lhi35_17"           "lhi35_18"
[817] "lhi35_19"           "lhi35_20"           "lhi36_1"
[820] "lhi36_2"            "lhi36_3"            "lhi36_4"
[823] "lhi36_5"            "lhi36_6"            "lhi36_7"
[826] "lhi36_8"            "lhi36_9"            "lhi36_10"
[829] "lhi36_11"           "lhi36_12"           "lhi36_13"
[832] "lhi36_14"           "lhi36_15"           "lhi36_16"
[835] "lhi36_17"           "lhi36_18"           "lhi36_19"
[838] "lhi36_20"           "lhi37_1"            "lhi37_2"
[841] "lhi37_3"            "lhi37_4"            "lhi37_5"
[844] "lhi37_6"            "lhi37_7"            "lhi37_8"
[847] "lhi37_9"            "lhi37_10"           "lhi37_11"
[850] "lhi37_12"           "lhi37_13"           "lhi37_14"
[853] "lhi37_15"           "lhi37_16"           "lhi37_17"
[856] "lhi37_18"           "lhi37_19"           "lhi37_20"
[859] "lhi38_1"            "lhi38_2"            "lhi38_3"
[862] "lhi38_4"            "lhi38_5"            "lhi38_6"
[865] "lhi38_7"            "lhi38_8"            "lhi38_9"
[868] "lhi38_10"           "lhi38_11"           "lhi38_12"
[871] "lhi38_13"           "lhi38_14"           "lhi38_15"
[874] "lhi38_16"           "lhi38_17"           "lhi38_18"
[877] "lhi38_19"           "lhi38_20"           "lhi39a_1"
[880] "lhi39a_2"           "lhi39a_3"           "lhi39a_4"
[883] "lhi39a_5"           "lhi39a_6"           "lhi39a_7"
[886] "lhi39a_8"           "lhi39a_9"           "lhi39a_10"
[889] "lhi39a_11"          "lhi39a_12"          "lhi39a_13"
[892] "lhi39a_14"          "lhi39a_15"          "lhi39a_16"
[895] "lhi39a_17"          "lhi39a_18"          "lhi39a_19"
[898] "lhi39a_20"          "lhi39au_1"          "lhi39au_2"
[901] "lhi39au_3"          "lhi39au_4"          "lhi39au_5"
```

```
 [904] "lhi39au_6"         "lhi39au_7"         "lhi39au_8"
 [907] "lhi39au_9"         "lhi39au_10"        "lhi39au_11"
 [910] "lhi39au_12"        "lhi39au_13"        "lhi39au_14"
 [913] "lhi39au_15"        "lhi39au_16"        "lhi39au_17"
 [916] "lhi39au_18"        "lhi39au_19"        "lhi39au_20"
 [919] "lhi39b_1"          "lhi39b_2"          "lhi39b_3"
 [922] "lhi39b_4"          "lhi39b_5"          "lhi39b_6"
 [925] "lhi39b_7"          "lhi39b_8"          "lhi39b_9"
 [928] "lhi39b_10"         "lhi39b_11"         "lhi39b_12"
 [931] "lhi39b_13"         "lhi39b_14"         "lhi39b_15"
 [934] "lhi39b_16"         "lhi39b_17"         "lhi39b_18"
 [937] "lhi39b_19"         "lhi39b_20"         "lhi39bu_1"
 [940] "lhi39bu_2"         "lhi39bu_3"         "lhi39bu_4"
 [943] "lhi39bu_5"         "lhi39bu_6"         "lhi39bu_7"
 [946] "lhi39bu_8"         "lhi39bu_9"         "lhi39bu_10"
 [949] "lhi39bu_11"        "lhi39bu_12"        "lhi39bu_13"
 [952] "lhi39bu_14"        "lhi39bu_15"        "lhi39bu_16"
 [955] "lhi39bu_17"        "lhi39bu_18"        "lhi39bu_19"
 [958] "lhi39bu_20"        "lhi40_1"           "lhi40_2"
 [961] "lhi40_3"           "lhi40_4"           "lhi40_5"
 [964] "lhi40_6"           "lhi40_7"           "lhi40_8"
 [967] "lhi40_9"           "lhi40_10"          "lhi40_11"
 [970] "lhi40_12"          "lhi40_13"          "lhi40_14"
 [973] "lhi40_15"          "lhi40_16"          "lhi40_17"
 [976] "lhi40_18"          "lhi40_19"          "lhi40_20"
 [979] "lhi41_1"           "lhi41_2"           "lhi41_3"
 [982] "lhi41_4"           "lhi41_5"           "lhi41_6"
 [985] "lhi41_7"           "lhi41_8"           "lhi41_9"
 [988] "lhi41_10"          "lhi41_11"          "lhi41_12"
 [991] "lhi41_13"          "lhi41_14"          "lhi41_15"
 [994] "lhi41_16"          "lhi41_17"          "lhi41_18"
 [997] "lhi41_19"          "lhi41_20"          "fer01a"
[1000] "fer01b"            "fer01c"            "fer02m"
[1003] "fer02y"            "fer03"             "fer04"
[1006] "fer04b"            "fer04c"            "fer04d"
[1009] "fer04e"            "fer05"             "fer06"
[1012] "fer07_1"           "fer07_2"           "fer07_3"
[1015] "fer07_4"           "fer07_5"           "fer07_6"
[1018] "fer07_7"           "fer07_8"           "fer07_9"
[1021] "fer07_10"          "fer08"             "fer09"
[1024] "fer10a"            "fer10bm"           "fer10by"
[1027] "fer11_1"           "fer11_2"           "fer11_3"
[1030] "fer11_4"           "fer11_5"           "fer11_6"
```

```
[1033] "fer11_7"          "fer11_8"          "fer12_1"
[1036] "fer12_2"          "fer12_3"          "fer12_4"
[1039] "fer12_5"          "fer12_6"          "fer12_7"
[1042] "fer12_8"          "fer12_9"          "fer12_10"
[1045] "fer12_11"         "fer12_12"         "fer12_13"
[1048] "fer12_14"         "fer13"            "fer14"
[1051] "fer15"            "fer16a"           "fer16b"
[1054] "fer16c"           "fer17"            "fer21"
[1057] "fer22"            "fer23"            "fer24"
[1060] "fer25a"           "fer25b"           "fer25c"
[1063] "fer25d"           "fer25e"           "fer25f"
[1066] "fer26a"           "fer26b"           "fer26e"
[1069] "fer26f"           "fer26h"           "fer27a"
[1072] "fer27b"           "fer27c"           "fer28"
[1075] "fer29"            "hhd01a"           "hhd01b"
[1078] "hhd03_1"          "hhd03_2"          "hhd03_3"
[1081] "hhd03_4"          "hhd03_5"          "hhd03_6"
[1084] "hhd03_7"          "hhd03_8"          "hhd03_9"
[1087] "hhd03_10"         "hhd03_11"         "hhd03_12"
[1090] "hhd03_13"         "hhd03_14"         "hhd03_15"
[1093] "hhd03_16"         "hhd03_17"         "hhd03_18"
[1096] "hhd03_19"         "hhd03_20"         "hhd04_1"
[1099] "hhd04_2"          "hhd04_3"          "hhd04_4"
[1102] "hhd04_5"          "hhd04_6"          "hhd04_7"
[1105] "hhd04_8"          "hhd04_9"          "hhd04_10"
[1108] "hhd04_11"         "hhd04_12"         "hhd04_13"
[1111] "hhd04_14"         "hhd04_15"         "hhd04_16"
[1114] "hhd04_17"         "hhd04_18"         "hhd04_19"
[1117] "hhd04_20"         "hhd05_1"          "hhd05_2"
[1120] "hhd05_3"          "hhd05_4"          "hhd05_5"
[1123] "hhd05_6"          "hhd05_7"          "hhd05_8"
[1126] "hhd05_9"          "hhd05_10"         "hhd05_11"
[1129] "hhd05_12"         "hhd05_13"         "hhd05_14"
[1132] "hhd05_15"         "hhd05_16"         "hhd05_17"
[1135] "hhd05_18"         "hhd05_19"         "hhd05_20"
[1138] "hhd06_m1"         "hhd06_m2"         "hhd06_m3"
[1141] "hhd06_m4"         "hhd06_m5"         "hhd06_m6"
[1144] "hhd06_m7"         "hhd06_m8"         "hhd06_m9"
[1147] "hhd06_m10"        "hhd06_m11"        "hhd06_m12"
[1150] "hhd06_m13"        "hhd06_m14"        "hhd06_m15"
[1153] "hhd06_m16"        "hhd06_m17"        "hhd06_m18"
[1156] "hhd06_m19"        "hhd06_m20"        "hhd06_y1"
[1159] "hhd06_y2"         "hhd06_y3"         "hhd06_y4"
```

```
[1162] "hhd06_y5"      "hhd06_y6"      "hhd06_y7"
[1165] "hhd06_y8"      "hhd06_y9"      "hhd06_y10"
[1168] "hhd06_y11"     "hhd06_y12"     "hhd06_y13"
[1171] "hhd06_y14"     "hhd06_y15"     "hhd06_y16"
[1174] "hhd06_y17"     "hhd06_y18"     "hhd06_y19"
[1177] "hhd06_y20"     "hhd07_1"       "hhd07_2"
[1180] "hhd07_3"       "hhd07_4"       "hhd07_5"
[1183] "hhd07_6"       "hhd07_7"       "hhd07_8"
[1186] "hhd07_9"       "hhd07_10"      "hhd07_11"
[1189] "hhd07_12"      "hhd07_13"      "hhd07_14"
[1192] "hhd07_15"      "hhd07_16"      "hhd07_17"
[1195] "hhd07_18"      "hhd07_19"      "hhd07_20"
[1198] "hhd08_1"       "hhd08_2"       "hhd08_3"
[1201] "hhd08_4"       "hhd08_5"       "hhd08_6"
[1204] "hhd08_7"       "hhd08_8"       "hhd08_9"
[1207] "hhd08_10"      "hhd08_11"      "hhd08_12"
[1210] "hhd08_13"      "hhd08_14"      "hhd08_15"
[1213] "hhd08_16"      "hhd08_17"      "hhd08_18"
[1216] "hhd08_19"      "hhd08_20"      "hhd09_1"
[1219] "hhd09_2"       "hhd09_3"       "hhd09_4"
[1222] "hhd09_5"       "hhd09_6"       "hhd09_7"
[1225] "hhd09_8"       "hhd09_9"       "hhd09_10"
[1228] "hhd09_11"      "hhd09_12"      "hhd09_13"
[1231] "hhd09_14"      "hhd09_15"      "hhd09_16"
[1234] "hhd09_17"      "hhd09_18"      "hhd09_19"
[1237] "hhd09_20"      "hhd11a"        "hhd11b"
[1240] "hhd11c"        "hhd11d"        "hhd11e"
[1243] "hhd11f"        "hhd12"         "hhd13a"
[1246] "hhd13b"        "hhd13c"        "hhd13d"
[1249] "hhd13e"        "hhd14"         "hhd15a"
[1252] "hhd15b"        "hhd15c"        "hhd15d"
[1255] "hhd16"         "hhd17"         "hhd18"
[1258] "hhd19_1"       "hhd19_2"       "hhd19_3"
[1261] "hhd19_4"       "hhd19_5"       "hhd19_6"
[1264] "hhd19_7"       "hhd19_8"       "hhd19_9"
[1267] "hhd19_10"      "hhd19_11"      "hhd19_12"
[1270] "hhd19_13"      "hhd19_14"      "hhd19_15"
[1273] "hhd19_16"      "hhd19_17"      "hhd19_18"
[1276] "hhd19_19"      "hhd19_20"      "hhd19_21"
[1279] "hhd19_22"      "hhd20"         "hhd20u"
[1282] "hhd21"         "hhd22"         "hhd23_1"
[1285] "hhd23_2"       "hhd23_3"       "hhd23_4"
[1288] "hhd23_5"       "hhd23_6"       "hhd24"
```

```
[1291] "hhd24a"       "hhd24u"       "hhd25"
[1294] "hhd26_1"      "hhd26_2"      "hhd26_3"
[1297] "hhd26_4"      "hhd26_5"      "hhd26_6"
[1300] "hhd26_7"      "hhd26_8"      "hhd26_9"
[1303] "hhd26_10"     "hhd26_11"     "hhd26_12"
[1306] "hhd26_13"     "hhd26_14"     "hhd26_15"
[1309] "hhd26_16"     "hhd26_17"     "hhd26_18"
[1312] "hhd26_19"     "hhd26_20"     "hhd26_21"
[1315] "hhd26_22"     "hhd27"        "hhd27u"
[1318] "hhd28"        "hhd29_1"      "hhd29_2"
[1321] "hhd29_3"      "hhd29_4"      "hhd29_5"
[1324] "hhd29_6"      "hhd29_7"      "hhd29_8"
[1327] "hhd29_9"      "hhd29_10"     "hhd29_11"
[1330] "hhd29_12"     "hhd29_13"     "hhd29_14"
[1333] "hhd29_15"     "hhd29_16"     "hhd29_17"
[1336] "hhd29_18"     "hhd29_19"     "hhd29_20"
[1339] "hhd29_21"     "hhd29_22"     "hhd30"
[1342] "hhd30u"       "hhd31"        "hhd35"
[1345] "hhd36_1"      "hhd36_2"      "hhd36_3"
[1348] "hhd36_4"      "hhd36_5"      "hhd36_6"
[1351] "hhd36_7"      "hhd36_8"      "hhd36_9"
[1354] "hhd36_10"     "hhd36_11"     "hhd36_12"
[1357] "hhd36_13"     "hhd36_14"     "hhd36_15"
[1360] "hhd36_16"     "hhd36_17"     "hhd36_18"
[1363] "hhd36_19"     "hhd36_20"     "hhd36_21"
[1366] "hhd36_22"     "gen01"        "gen02"
[1369] "gen03"        "gen09m"       "gen09y"
[1372] "gen10m"       "gen10y"       "gen11"
[1375] "gen12iso"     "gen15a"       "gen15au"
[1378] "gen15b"       "gen15bu"      "gen16"
[1381] "gen23m"       "gen23y"       "gen24m"
[1384] "gen24y"       "gen25"        "gen26iso"
[1387] "gen29a"       "gen29au"      "gen29b"
[1390] "gen29bu"      "gen30"        "gen37a"
[1393] "gen37m"       "gen37y"       "gen38a"
[1396] "gen38bm"      "gen38by"      "gen39a"
[1399] "gen39b"       "gen40"        "gen41a"
[1402] "gen41a_4001"  "gen41b"       "gen41b_4001"
[1405] "gen42"        "gen43"        "gen44aiso"
[1408] "gen44b"       "gen45"        "gen46"
[1411] "gen47"        "gen48"        "gen48isco"
[1414] "gen49"        "gen49isced"   "gen50"
[1417] "gen50isco"    "gen51"        "gen51isced"
```

```
[1420] "gen52"           "gen52am"         "gen52ay"
[1423] "gen53"           "gen54"           "gen55"
[1426] "gen56"           "gen57m"          "gen57y"
[1429] "gen58"           "gen59"           "gen60_1"
[1432] "gen60_2"         "gen60_3"         "gen60_4"
[1435] "gen60_5"         "gen60_6"         "gen60_7"
[1438] "gen60_8"         "gen60_9"         "gen60_10"
[1441] "gen60_11"        "gen60_12"        "gen60_13"
[1444] "gen60_14"        "gen60_15"        "gen60_16"
[1447] "gen60_17"        "gen60_18"        "gen60_19"
[1450] "gen60_20"        "gen60_21"        "gen60_22"
[1453] "gen63"           "gen66"           "gen67_1"
[1456] "gen67_2"         "gen67_3"         "gen67_4"
[1459] "gen67_5"         "gen67_6"         "gen67_7"
[1462] "gen67_8"         "gen67_9"         "gen67_10"
[1465] "gen67_11"        "gen67_12"        "gen67_13"
[1468] "gen67_14"        "gen67_15"        "gen67_16"
[1471] "gen67_17"        "gen67_18"        "gen67_19"
[1474] "gen67_20"        "gen67_21"        "gen67_22"
[1477] "gen68"           "gen69_1"         "gen69_2"
[1480] "gen69_3"         "gen69_4"         "gen69_5"
[1483] "gen69_6"         "gen69_7"         "gen69_8"
[1486] "gen69_9"         "gen69_10"        "gen69_11"
[1489] "gen69_12"        "gen69_13"        "gen69_14"
[1492] "gen69_15"        "gen69_16"        "gen69_17"
[1495] "gen69_18"        "gen69_19"        "gen69_20"
[1498] "gen69_21"        "gen69_22"        "gen70"
[1501] "gen71_1"         "gen71_2"         "gen71_3"
[1504] "gen71_4"         "gen71_5"         "gen71_6"
[1507] "gen71_7"         "gen71_8"         "gen71_9"
[1510] "gen71_10"        "gen71_11"        "gen71_12"
[1513] "gen71_13"        "gen71_14"        "gen71_15"
[1516] "gen71_16"        "gen71_17"        "gen71_18"
[1519] "gen71_19"        "gen71_20"        "gen71_21"
[1522] "gen71_22"        "wel01"           "wel02"
[1525] "wel02a"          "wel03_1"         "wel03_2"
[1528] "wel03_3"         "wel03_4"         "wel03_5"
[1531] "wel03_6"         "wel03_7"         "wel03_8"
[1534] "wel03_9"         "wel03_10"        "wel03_11"
[1537] "wel03_12"        "wel03_13"        "wel03_14"
[1540] "wel03_15"        "wel03_16"        "wel03_17"
[1543] "wel03_18"        "wel03_19"        "wel03_20"
[1546] "wel04"           "wel05"           "wel06"
```

```
[1549] "wel07"           "wel08"           "wel09a"
[1552] "wel09b"          "wel09c"          "wel09d"
[1555] "wel09e"          "wel09f"          "wel10_1"
[1558] "wel10_2"         "wel10_3"         "wel10_4"
[1561] "wel10_5"         "wel10_6"         "wel10_7"
[1564] "wel10_8"         "wel10_9"         "wel10_10"
[1567] "wel10_11"        "wel10_12"        "wel10_13"
[1570] "wel10_14"        "wel10_15"        "wel10_16"
[1573] "wel10_17"        "wel10_18"        "wel10_19"
[1576] "wel10_20"        "wel10_21"        "wel10_22"
[1579] "wel11a"          "wel11b"          "wel11c"
[1582] "wel11d"          "wel11e"          "wel14a_4001"
[1585] "wel14b_4001"     "wel14c_4001"     "wel14d_4001"
[1588] "wel14e_4001"     "wel14f_4001"     "wel14g_4001"
[1591] "wel16a_4001"     "wel16b_4001"     "wel16c_1_4001"
[1594] "wel16c_2_4001"   "wel16c_3_4001"   "wel16c_4_4001"
[1597] "wel16c_5_4001"   "wel16c_6_4001"   "wel16c_7_4001"
[1600] "wel16c_8_4001"   "wel16c_9_4001"   "wel16c_10_4001"
[1603] "wel16c_11_4001"  "wel16c_12_4001"  "wel16c_13_4001"
[1606] "wel16c_14_4001"  "wel16c_15_4001"  "wel16c_16_4001"
[1609] "wel16c_17_4001"  "wel16c_18_4001"  "wel16c_19_4001"
[1612] "wel16c_20_4001"  "wel16d_m1_4001"  "wel16d_m2_4001"
[1615] "wel16d_m3_4001"  "wel16d_m4_4001"  "wel16d_m5_4001"
[1618] "wel16d_m6_4001"  "wel16d_m7_4001"  "wel16d_m8_4001"
[1621] "wel16d_m9_4001"  "wel16d_m10_4001" "wel16d_m11_4001"
[1624] "wel16d_m12_4001" "wel16d_m13_4001" "wel16d_m14_4001"
[1627] "wel16d_m15_4001" "wel16d_m16_4001" "wel16d_m17_4001"
[1630] "wel16d_m18_4001" "wel16d_m19_4001" "wel16d_m20_4001"
[1633] "wel16d_y1_4001"  "wel16d_y2_4001"  "wel16d_y3_4001"
[1636] "wel16d_y4_4001"  "wel16d_y5_4001"  "wel16d_y6_4001"
[1639] "wel16d_y7_4001"  "wel16d_y8_4001"  "wel16d_y9_4001"
[1642] "wel16d_y10_4001" "wel16d_y11_4001" "wel16d_y12_4001"
[1645] "wel16d_y13_4001" "wel16d_y14_4001" "wel16d_y15_4001"
[1648] "wel16d_y16_4001" "wel16d_y17_4001" "wel16d_y18_4001"
[1651] "wel16d_y19_4001" "wel16d_y20_4001" "wel16a_1_4002"
[1654] "wel16a_2_4002"   "wel16a_3_4002"   "wel16a_4_4002"
[1657] "wel16a_5_4002"   "wel16a_6_4002"   "wel16a_7_4002"
[1660] "wel16a_8_4002"   "wel16a_9_4002"   "wel16a_10_4002"
[1663] "wel16a_11_4002"  "wel16a_12_4002"  "wel16a_13_4002"
[1666] "wel16a_14_4002"  "wel16a_15_4002"  "wel16a_16_4002"
[1669] "wel16a_17_4002"  "wel16a_18_4002"  "wel16a_19_4002"
[1672] "wel16a_20_4002"  "wel16b_1_4002"   "wel16b_2_4002"
[1675] "wel16b_3_4002"   "wel16b_4_4002"   "wel16b_5_4002"
```

```
[1678] "wel16b_6_4002"    "wel16b_7_4002"    "wel16b_8_4002"
[1681] "wel16b_9_4002"    "wel16b_10_4002"   "wel16b_11_4002"
[1684] "wel16b_12_4002"   "wel16b_13_4002"   "wel16b_14_4002"
[1687] "wel16b_15_4002"   "wel16b_16_4002"   "wel16b_17_4002"
[1690] "wel16b_18_4002"   "wel16b_19_4002"   "wel16b_20_4002"
[1693] "wel16c_1_4002"    "wel16c_2_4002"    "wel16c_3_4002"
[1696] "wel16c_4_4002"    "wel16c_5_4002"    "wel16c_6_4002"
[1699] "wel16c_7_4002"    "wel16c_8_4002"    "wel16c_9_4002"
[1702] "wel16c_10_4002"   "wel16c_11_4002"   "wel16c_12_4002"
[1705] "wel16c_13_4002"   "wel16c_14_4002"   "wel16c_15_4002"
[1708] "wel16c_16_4002"   "wel16c_17_4002"   "wel16c_18_4002"
[1711] "wel16c_19_4002"   "wel16c_20_4002"   "wel16d_1_4002"
[1714] "wel16d_2_4002"    "wel16d_3_4002"    "wel16d_4_4002"
[1717] "wel16d_5_4002"    "wel16d_6_4002"    "wel16d_7_4002"
[1720] "wel16d_8_4002"    "wel16d_9_4002"    "wel16d_10_4002"
[1723] "wel16d_11_4002"   "wel16d_12_4002"   "wel16d_13_4002"
[1726] "wel16d_14_4002"   "wel16d_15_4002"   "wel16d_16_4002"
[1729] "wel16d_17_4002"   "wel16d_18_4002"   "wel16d_19_4002"
[1732] "wel16d_20_4002"   "wel16e_1_4002"    "wel16e_2_4002"
[1735] "wel16e_3_4002"    "wel16e_4_4002"    "wel16e_5_4002"
[1738] "wel16e_6_4002"    "wel16e_7_4002"    "wel16e_8_4002"
[1741] "wel16e_9_4002"    "wel16e_10_4002"   "wel16e_11_4002"
[1744] "wel16e_12_4002"   "wel16e_13_4002"   "wel16e_14_4002"
[1747] "wel16e_15_4002"   "wel16e_16_4002"   "wel16e_17_4002"
[1750] "wel16e_18_4002"   "wel16e_19_4002"   "wel16e_20_4002"
[1753] "wel16f_1_4002"    "wel16f_2_4002"    "wel16f_3_4002"
[1756] "wel16f_4_4002"    "wel16f_5_4002"    "wel16f_6_4002"
[1759] "wel16f_7_4002"    "wel16f_8_4002"    "wel16f_9_4002"
[1762] "wel16f_10_4002"   "wel16f_11_4002"   "wel16f_12_4002"
[1765] "wel16f_13_4002"   "wel16f_14_4002"   "wel16f_15_4002"
[1768] "wel16f_16_4002"   "wel16f_17_4002"   "wel16f_18_4002"
[1771] "wel16f_19_4002"   "wel16f_20_4002"   "wel16g_1_4002"
[1774] "wel16g_2_4002"    "wel16g_3_4002"    "wel16g_4_4002"
[1777] "wel16g_5_4002"    "wel16g_6_4002"    "wel16g_7_4002"
[1780] "wel16g_8_4002"    "wel16g_9_4002"    "wel16g_10_4002"
[1783] "wel16g_11_4002"   "wel16g_12_4002"   "wel16g_13_4002"
[1786] "wel16g_14_4002"   "wel16g_15_4002"   "wel16g_16_4002"
[1789] "wel16g_17_4002"   "wel16g_18_4002"   "wel16g_19_4002"
[1792] "wel16g_20_4002"   "wrk01"            "wrk02"
[1795] "wrk03m"           "wrk03y"           "wrk04"
[1798] "wrk04isco"        "wrk06"            "wrk07"
[1801] "wrk08"            "wrk09"            "wrk10"
[1804] "wrk11"            "wrk12"            "wrk13"
```

```
[1807] "wrk14"          "wrk15a"          "wrk15b"
[1810] "wrk15c"         "wrk15d"          "wrk16a"
[1813] "wrk16b"         "wrk17"           "wrk18"
[1816] "wrk20"          "wrk21"           "wrk22"
[1819] "wrk23"          "wrk24"           "wrk25"
[1822] "wrk26"          "wrk27"           "wrk27isco"
[1825] "wrk28"          "wrk30"           "wrk30am"
[1828] "wrk30ay"        "wrk31"           "wrk32"
[1831] "wrk34"          "wrk34isco"       "wrk35"
[1834] "wrk36"          "wrk37"           "wrk38"
[1837] "wrk39"          "wrk40"           "wrk41"
[1840] "wrk42"          "wrk43"           "wrk44"
[1843] "wrk46"          "wrk47"           "wrk48"
[1846] "wrk49"          "wrk50"           "wrk51_4001"
[1849] "wrk51_4002"     "wrk51_4003"      "wrk51_4004"
[1852] "wrk51a_4005"    "wrk51b_4005"     "wrk51_4006"
[1855] "wrk51a_4007"    "wrk51b_4007"     "wrk51a_4008"
[1858] "wrk51b_4008"    "wrk51_4009"      "wrk51_4010"
[1861] "wrk51a_4011"    "wrk51b_4011"     "wrk51_4012"
[1864] "wrk51_4013"     "wrk51_4014"      "wrk51_4015"
[1867] "inc01"          "inc03"           "inc05"
[1870] "inc06"          "inc08_1"         "inc08_2"
[1873] "inc08_3"        "inc08_4"         "inc08_5"
[1876] "inc08_6"        "inc08_7"         "inc08_8"
[1879] "inc08_9"        "inc08_10"        "inc08_11"
[1882] "inc08_12"       "inc09_1"         "inc09_2"
[1885] "inc09_3"        "inc09_4"         "inc09_5"
[1888] "inc09_6"        "inc09_7"         "inc09_8"
[1891] "inc09_9"        "inc09_10"        "inc09_11"
[1894] "inc11_1"        "inc11_2"         "inc11_3"
[1897] "inc11_4"        "inc11_5"         "inc11_6"
[1900] "inc11_7"        "inc11_8"         "inc11_9"
[1903] "inc11_10"       "inc11_11"        "inc12"
[1906] "inc13"          "inc14_1"         "inc14_2"
[1909] "inc14_3"        "inc14_4"         "inc14_5"
[1912] "inc14_6"        "inc14_7"         "inc14_8"
[1915] "inc14_9"        "inc14_10"        "inc14_11"
[1918] "inc14_12"       "inc14_13"        "inc14_14"
[1921] "inc14_15"       "inc14_16"        "inc14_17"
[1924] "inc14_18"       "inc14_19"        "inc14_20"
[1927] "inc14_21"       "inc14_22"        "inc15"
[1930] "att01"          "att02"           "att03a"
[1933] "att03b"         "att03d"          "att03e"
```

```
[1936] "att03g"           "att03h"           "att03i"
[1939] "att03j"           "att05b"           "att06a"
[1942] "att06b"           "att07a"           "att07b"
[1945] "att07c"           "att07d"           "att07g"
[1948] "att08"            "att09"            "att09u"
[1951] "att10"            "att11b"           "att11d"
[1954] "att13a_4001"      "att13b_4001"      "att13c_4001"
[1957] "att13d_4001"      "att13e_4001"      "att13f_4001"
[1960] "att13g_4001"      "att13h_4001"      "att13_4002"
[1963] "att13_4003"       "att13_4004"       "att13_4005"
[1966] "att13_1_4006"     "att13_2_4006"     "att13_3_4006"
[1969] "att13_4_4006"     "att13_5_4006"     "att13_6_4006"
[1972] "att13_7_4006"     "att13_8_4006"     "att13_9_4006"
[1975] "att13_4007"       "att19a_4001"      "att19b_4001"
[1978] "att19c_4001"      "rep01"            "rep02"
[1981] "rep03_1"          "rep03_2"          "rep03_3"
[1984] "rep03_4"          "rep04"            "rep05"
[1987] "rep06"            "flag1"            "localitysize_4001"
[1990] "department_4001"  "city_4001"
```

```r
names(ejemploxl)
```

```
[1] "Causa" "año"   "valor"
```

Como vemos en las bases hay mayúsculas, caracteres especiales y demás. Esto lo podemos cambiar

```r
ejemploxl<-ejemploxl %>%
  janitor::clean_names()

names(ejemploxl)
```

```
[1] "causa" "ano"   "valor"
```

Si quisiéramos que la acción quedará en una sola operación, podemos usar un pipe diferente:

```r
pacman::p_load(magrittr)

encuesta_generacion %<>% # este es otro pipe
```

```r
  janitor::clean_names()

names(encuesta_generacion)
```

```
  [1] "country"         "region"          "respid"
  [4] "intid"           "mode"            "weight"
  [7] "instrument"      "intdatem"        "intdatey"
 [10] "dem01"           "dem02m"          "dem02y"
 [13] "dem03"           "dem04a"          "dem04biso"
 [16] "dem05m"          "dem05y"          "dem06"
 [19] "dem07"           "dem07isced"      "dem08m"
 [22] "dem08y"          "dem09"           "dem10m"
 [25] "dem10y"          "dem11"           "dem12"
 [28] "dem14"           "dem15"           "dem17"
 [31] "dem18"           "dem19"           "dem20"
 [34] "dem21"           "dem22a"          "dem22m"
 [37] "dem22y"          "dem23"           "dem24a"
 [40] "dem24biso"       "dem24em"         "dem24ey"
 [43] "dem25"           "dem25isced"      "dem26"
 [46] "dem27"           "dem28a"          "dem28bm"
 [49] "dem28by"         "dem28c"          "dem30a"
 [52] "dem30bm"         "dem30by"         "dem30c"
 [55] "dem30d"          "dem31m"          "dem31y"
 [58] "dem32a"          "dem32b"          "dem32c"
 [61] "dem32d"          "dem33"           "dem33am"
 [64] "dem33ay"         "dem34m"          "dem34y"
 [67] "dem35"           "dem36a"          "dem36au"
 [70] "dem36b"          "dem36bu"         "dem37"
 [73] "dem38a"          "dem38b"          "dem38c"
 [76] "dem38d"          "dem38e"          "dem38f"
 [79] "dem38g"          "dem39a"          "dem39b"
 [82] "dem39c"          "dem39d"          "dem40"
 [85] "dem41"           "dem42"           "dem43"
 [88] "dem44"           "dem45"           "dem46"
 [91] "lhi01"           "lhi02"           "lhi04_m1"
 [94] "lhi04_m2"        "lhi04_m3"        "lhi04_m4"
 [97] "lhi04_m5"        "lhi04_m6"        "lhi04_m7"
[100] "lhi04_m8"        "lhi04_m9"        "lhi04_m10"
[103] "lhi04_m11"       "lhi04_m12"       "lhi04_m13"
[106] "lhi04_m14"       "lhi04_m15"       "lhi04_m16"
[109] "lhi04_m17"       "lhi04_m18"       "lhi04_m19"
[112] "lhi04_m20"       "lhi04_y1"        "lhi04_y2"
```

```
[115] "lhi04_y3"        "lhi04_y4"        "lhi04_y5"
[118] "lhi04_y6"        "lhi04_y7"        "lhi04_y8"
[121] "lhi04_y9"        "lhi04_y10"       "lhi04_y11"
[124] "lhi04_y12"       "lhi04_y13"       "lhi04_y14"
[127] "lhi04_y15"       "lhi04_y16"       "lhi04_y17"
[130] "lhi04_y18"       "lhi04_y19"       "lhi04_y20"
[133] "lhi04a_1"        "lhi04a_2"        "lhi04a_3"
[136] "lhi04a_4"        "lhi04a_5"        "lhi04a_6"
[139] "lhi04a_7"        "lhi04a_8"        "lhi04a_9"
[142] "lhi04a_10"       "lhi04a_11"       "lhi04a_12"
[145] "lhi04a_13"       "lhi04a_14"       "lhi04a_15"
[148] "lhi04a_16"       "lhi04a_17"       "lhi04a_18"
[151] "lhi04a_19"       "lhi04a_20"       "lhi05a_1"
[154] "lhi05a_2"        "lhi05a_3"        "lhi05a_4"
[157] "lhi05a_5"        "lhi05a_6"        "lhi05a_7"
[160] "lhi05a_8"        "lhi05a_9"        "lhi05a_10"
[163] "lhi05a_11"       "lhi05a_12"       "lhi05a_13"
[166] "lhi05a_14"       "lhi05a_15"       "lhi05a_16"
[169] "lhi05a_17"       "lhi05a_18"       "lhi05a_19"
[172] "lhi05a_20"       "lhi05b_m1"       "lhi05b_m2"
[175] "lhi05b_m3"       "lhi05b_m4"       "lhi05b_m5"
[178] "lhi05b_m6"       "lhi05b_m7"       "lhi05b_m8"
[181] "lhi05b_m9"       "lhi05b_m10"      "lhi05b_m11"
[184] "lhi05b_m12"      "lhi05b_m13"      "lhi05b_m14"
[187] "lhi05b_m15"      "lhi05b_m16"      "lhi05b_m17"
[190] "lhi05b_m18"      "lhi05b_m19"      "lhi05b_m20"
[193] "lhi05b_y1"       "lhi05b_y2"       "lhi05b_y3"
[196] "lhi05b_y4"       "lhi05b_y5"       "lhi05b_y6"
[199] "lhi05b_y7"       "lhi05b_y8"       "lhi05b_y9"
[202] "lhi05b_y10"      "lhi05b_y11"      "lhi05b_y12"
[205] "lhi05b_y13"      "lhi05b_y14"      "lhi05b_y15"
[208] "lhi05b_y16"      "lhi05b_y17"      "lhi05b_y18"
[211] "lhi05b_y19"      "lhi05b_y20"      "lhi06_m1"
[214] "lhi06_m2"        "lhi06_m3"        "lhi06_m4"
[217] "lhi06_m5"        "lhi06_m6"        "lhi06_m7"
[220] "lhi06_m8"        "lhi06_m9"        "lhi06_m10"
[223] "lhi06_m11"       "lhi06_m12"       "lhi06_m13"
[226] "lhi06_m14"       "lhi06_m15"       "lhi06_m16"
[229] "lhi06_m17"       "lhi06_m18"       "lhi06_m19"
[232] "lhi06_m20"       "lhi06_y1"        "lhi06_y2"
[235] "lhi06_y3"        "lhi06_y4"        "lhi06_y5"
[238] "lhi06_y6"        "lhi06_y7"        "lhi06_y8"
[241] "lhi06_y9"        "lhi06_y10"       "lhi06_y11"
```

```
[244] "lhi06_y12"           "lhi06_y13"           "lhi06_y14"
[247] "lhi06_y15"           "lhi06_y16"           "lhi06_y17"
[250] "lhi06_y18"           "lhi06_y19"           "lhi06_y20"
[253] "lhi07_1"             "lhi07_2"             "lhi07_3"
[256] "lhi07_4"             "lhi07_5"             "lhi07_6"
[259] "lhi07_7"             "lhi07_8"             "lhi07_9"
[262] "lhi07_10"            "lhi07_11"            "lhi07_12"
[265] "lhi07_13"            "lhi07_14"            "lhi07_15"
[268] "lhi07_16"            "lhi07_17"            "lhi07_18"
[271] "lhi07_19"            "lhi07_20"            "lhi08_1"
[274] "lhi08_2"             "lhi08_3"             "lhi08_4"
[277] "lhi08_5"             "lhi08_6"             "lhi08_7"
[280] "lhi08_8"             "lhi08_9"             "lhi08_10"
[283] "lhi08_11"            "lhi08_12"            "lhi08_13"
[286] "lhi08_14"            "lhi08_15"            "lhi08_16"
[289] "lhi08_17"            "lhi08_18"            "lhi08_19"
[292] "lhi08_20"            "lhi09_1"             "lhi09_2"
[295] "lhi09_3"             "lhi09_4"             "lhi09_5"
[298] "lhi09_6"             "lhi09_7"             "lhi09_8"
[301] "lhi09_9"             "lhi09_10"            "lhi09_11"
[304] "lhi09_12"            "lhi09_13"            "lhi09_14"
[307] "lhi09_15"            "lhi09_16"            "lhi09_17"
[310] "lhi09_18"            "lhi09_19"            "lhi09_20"
[313] "lhi10_1"             "lhi10_2"             "lhi10_3"
[316] "lhi10_4"             "lhi10_5"             "lhi10_6"
[319] "lhi10_7"             "lhi10_8"             "lhi10_9"
[322] "lhi10_10"            "lhi10_11"            "lhi10_12"
[325] "lhi10_13"            "lhi10_14"            "lhi10_15"
[328] "lhi10_16"            "lhi10_17"            "lhi10_18"
[331] "lhi10_19"            "lhi10_20"            "lhi11_1"
[334] "lhi11_2"             "lhi11_3"             "lhi11_4"
[337] "lhi11_5"             "lhi11_6"             "lhi11_7"
[340] "lhi11_8"             "lhi11_9"             "lhi11_10"
[343] "lhi11_11"            "lhi11_12"            "lhi11_13"
[346] "lhi11_14"            "lhi11_15"            "lhi11_16"
[349] "lhi11_17"            "lhi11_18"            "lhi11_19"
[352] "lhi11_20"            "lhi12_1"             "lhi12_2"
[355] "lhi12_3"             "lhi12_4"             "lhi12_5"
[358] "lhi12_6"             "lhi12_7"             "lhi12_8"
[361] "lhi12_9"             "lhi12_10"            "lhi12_11"
[364] "lhi12_12"            "lhi12_13"            "lhi12_14"
[367] "lhi12_15"            "lhi12_16"            "lhi12_17"
[370] "lhi12_18"            "lhi12_19"            "lhi12_20"
```

```
[373] "lhi13_1"          "lhi13_2"          "lhi13_3"
[376] "lhi13_4"          "lhi13_5"          "lhi13_6"
[379] "lhi13_7"          "lhi13_8"          "lhi13_9"
[382] "lhi13_10"         "lhi13_11"         "lhi13_12"
[385] "lhi13_13"         "lhi13_14"         "lhi13_15"
[388] "lhi13_16"         "lhi13_17"         "lhi13_18"
[391] "lhi13_19"         "lhi13_20"         "lhi14_m1"
[394] "lhi14_m2"         "lhi14_m3"         "lhi14_m4"
[397] "lhi14_m5"         "lhi14_m6"         "lhi14_m7"
[400] "lhi14_m8"         "lhi14_m9"         "lhi14_m10"
[403] "lhi14_m11"        "lhi14_m12"        "lhi14_m13"
[406] "lhi14_m14"        "lhi14_m15"        "lhi14_m16"
[409] "lhi14_m17"        "lhi14_m18"        "lhi14_m19"
[412] "lhi14_m20"        "lhi14_y1"         "lhi14_y2"
[415] "lhi14_y3"         "lhi14_y4"         "lhi14_y5"
[418] "lhi14_y6"         "lhi14_y7"         "lhi14_y8"
[421] "lhi14_y9"         "lhi14_y10"        "lhi14_y11"
[424] "lhi14_y12"        "lhi14_y13"        "lhi14_y14"
[427] "lhi14_y15"        "lhi14_y16"        "lhi14_y17"
[430] "lhi14_y18"        "lhi14_y19"        "lhi14_y20"
[433] "lhi15a_1"         "lhi15a_2"         "lhi15a_3"
[436] "lhi15a_4"         "lhi15a_5"         "lhi15a_6"
[439] "lhi15a_7"         "lhi15a_8"         "lhi15a_9"
[442] "lhi15a_10"        "lhi15a_11"        "lhi15a_12"
[445] "lhi15a_13"        "lhi15a_14"        "lhi15a_15"
[448] "lhi15a_16"        "lhi15a_17"        "lhi15a_18"
[451] "lhi15a_19"        "lhi15a_20"        "lhi15b_m1"
[454] "lhi15b_m2"        "lhi15b_m3"        "lhi15b_m4"
[457] "lhi15b_m5"        "lhi15b_m6"        "lhi15b_m7"
[460] "lhi15b_m8"        "lhi15b_m9"        "lhi15b_m10"
[463] "lhi15b_m11"       "lhi15b_m12"       "lhi15b_m13"
[466] "lhi15b_m14"       "lhi15b_m15"       "lhi15b_m16"
[469] "lhi15b_m17"       "lhi15b_m18"       "lhi15b_m19"
[472] "lhi15b_m20"       "lhi15b_y1"        "lhi15b_y2"
[475] "lhi15b_y3"        "lhi15b_y4"        "lhi15b_y5"
[478] "lhi15b_y6"        "lhi15b_y7"        "lhi15b_y8"
[481] "lhi15b_y9"        "lhi15b_y10"       "lhi15b_y11"
[484] "lhi15b_y12"       "lhi15b_y13"       "lhi15b_y14"
[487] "lhi15b_y15"       "lhi15b_y16"       "lhi15b_y17"
[490] "lhi15b_y18"       "lhi15b_y19"       "lhi15b_y20"
[493] "lhi16_1"          "lhi16_2"          "lhi16_3"
[496] "lhi16_4"          "lhi16_5"          "lhi16_6"
[499] "lhi16_7"          "lhi16_8"          "lhi16_9"
```

```
[502] "lhi16_10"     "lhi16_11"     "lhi16_12"
[505] "lhi16_13"     "lhi16_14"     "lhi16_15"
[508] "lhi16_16"     "lhi16_17"     "lhi16_18"
[511] "lhi16_19"     "lhi16_20"     "lhi17_1"
[514] "lhi17_2"      "lhi17_3"      "lhi17_4"
[517] "lhi17_5"      "lhi17_6"      "lhi17_7"
[520] "lhi17_8"      "lhi17_9"      "lhi17_10"
[523] "lhi17_11"     "lhi17_12"     "lhi17_13"
[526] "lhi17_14"     "lhi17_15"     "lhi17_16"
[529] "lhi17_17"     "lhi17_18"     "lhi17_19"
[532] "lhi17_20"     "lhi18"        "lhi19"
[535] "lhi20"        "lhi21"        "lhi22"
[538] "lhi23"        "lhi25_1"      "lhi25_2"
[541] "lhi25_3"      "lhi25_4"      "lhi25_5"
[544] "lhi25_6"      "lhi25_7"      "lhi25_8"
[547] "lhi25_9"      "lhi25_10"     "lhi25_11"
[550] "lhi25_12"     "lhi25_13"     "lhi25_14"
[553] "lhi25_15"     "lhi25_16"     "lhi25_17"
[556] "lhi25_18"     "lhi25_19"     "lhi25_20"
[559] "lhi26_1"      "lhi26_2"      "lhi26_3"
[562] "lhi26_4"      "lhi26_5"      "lhi26_6"
[565] "lhi26_7"      "lhi26_8"      "lhi26_9"
[568] "lhi26_10"     "lhi26_11"     "lhi26_12"
[571] "lhi26_13"     "lhi26_14"     "lhi26_15"
[574] "lhi26_16"     "lhi26_17"     "lhi26_18"
[577] "lhi26_19"     "lhi26_20"     "lhi27_1"
[580] "lhi27_2"      "lhi27_3"      "lhi27_4"
[583] "lhi27_5"      "lhi27_6"      "lhi27_7"
[586] "lhi27_8"      "lhi27_9"      "lhi27_10"
[589] "lhi27_11"     "lhi27_12"     "lhi27_13"
[592] "lhi27_14"     "lhi27_15"     "lhi27_16"
[595] "lhi27_17"     "lhi27_18"     "lhi27_19"
[598] "lhi27_20"     "lhi28_1"      "lhi28_2"
[601] "lhi28_3"      "lhi28_4"      "lhi28_5"
[604] "lhi28_6"      "lhi28_7"      "lhi28_8"
[607] "lhi28_9"      "lhi28_10"     "lhi28_11"
[610] "lhi28_12"     "lhi28_13"     "lhi28_14"
[613] "lhi28_15"     "lhi28_16"     "lhi28_17"
[616] "lhi28_18"     "lhi28_19"     "lhi28_20"
[619] "lhi29_m1"     "lhi29_m2"     "lhi29_m3"
[622] "lhi29_m4"     "lhi29_m5"     "lhi29_m6"
[625] "lhi29_m7"     "lhi29_m8"     "lhi29_m9"
[628] "lhi29_m10"    "lhi29_m11"    "lhi29_m12"
```

```
[631] "lhi29_m13"          "lhi29_m14"          "lhi29_m15"
[634] "lhi29_m16"          "lhi29_m17"          "lhi29_m18"
[637] "lhi29_m19"          "lhi29_m20"          "lhi29_y1"
[640] "lhi29_y2"           "lhi29_y3"           "lhi29_y4"
[643] "lhi29_y5"           "lhi29_y6"           "lhi29_y7"
[646] "lhi29_y8"           "lhi29_y9"           "lhi29_y10"
[649] "lhi29_y11"          "lhi29_y12"          "lhi29_y13"
[652] "lhi29_y14"          "lhi29_y15"          "lhi29_y16"
[655] "lhi29_y17"          "lhi29_y18"          "lhi29_y19"
[658] "lhi29_y20"          "lhi30_m1"           "lhi30_m2"
[661] "lhi30_m3"           "lhi30_m4"           "lhi30_m5"
[664] "lhi30_m6"           "lhi30_m7"           "lhi30_m8"
[667] "lhi30_m9"           "lhi30_m10"          "lhi30_m11"
[670] "lhi30_m12"          "lhi30_m13"          "lhi30_m14"
[673] "lhi30_m15"          "lhi30_m16"          "lhi30_m17"
[676] "lhi30_m18"          "lhi30_m19"          "lhi30_m20"
[679] "lhi30_y1"           "lhi30_y2"           "lhi30_y3"
[682] "lhi30_y4"           "lhi30_y5"           "lhi30_y6"
[685] "lhi30_y7"           "lhi30_y8"           "lhi30_y9"
[688] "lhi30_y10"          "lhi30_y11"          "lhi30_y12"
[691] "lhi30_y13"          "lhi30_y14"          "lhi30_y15"
[694] "lhi30_y16"          "lhi30_y17"          "lhi30_y18"
[697] "lhi30_y19"          "lhi30_y20"          "lhi31_1"
[700] "lhi31_2"            "lhi31_3"            "lhi31_4"
[703] "lhi31_5"            "lhi31_6"            "lhi31_7"
[706] "lhi31_8"            "lhi31_9"            "lhi31_10"
[709] "lhi31_11"           "lhi31_12"           "lhi31_13"
[712] "lhi31_14"           "lhi31_15"           "lhi31_16"
[715] "lhi31_17"           "lhi31_18"           "lhi31_19"
[718] "lhi31_20"           "lhi32_1"            "lhi32_2"
[721] "lhi32_3"            "lhi32_4"            "lhi32_5"
[724] "lhi32_6"            "lhi32_7"            "lhi32_8"
[727] "lhi32_9"            "lhi32_10"           "lhi32_11"
[730] "lhi32_12"           "lhi32_13"           "lhi32_14"
[733] "lhi32_15"           "lhi32_16"           "lhi32_17"
[736] "lhi32_18"           "lhi32_19"           "lhi32_20"
[739] "lhi33_1"            "lhi33_2"            "lhi33_3"
[742] "lhi33_4"            "lhi33_5"            "lhi33_6"
[745] "lhi33_7"            "lhi33_8"            "lhi33_9"
[748] "lhi33_10"           "lhi33_11"           "lhi33_12"
[751] "lhi33_13"           "lhi33_14"           "lhi33_15"
[754] "lhi33_16"           "lhi33_17"           "lhi33_18"
[757] "lhi33_19"           "lhi33_20"           "lhi33u_1"
```

```
[760] "lhi33u_2"     "lhi33u_3"     "lhi33u_4"
[763] "lhi33u_5"     "lhi33u_6"     "lhi33u_7"
[766] "lhi33u_8"     "lhi33u_9"     "lhi33u_10"
[769] "lhi33u_11"    "lhi33u_12"    "lhi33u_13"
[772] "lhi33u_14"    "lhi33u_15"    "lhi33u_16"
[775] "lhi33u_17"    "lhi33u_18"    "lhi33u_19"
[778] "lhi33u_20"    "lhi34_1"      "lhi34_2"
[781] "lhi34_3"      "lhi34_4"      "lhi34_5"
[784] "lhi34_6"      "lhi34_7"      "lhi34_8"
[787] "lhi34_9"      "lhi34_10"     "lhi34_11"
[790] "lhi34_12"     "lhi34_13"     "lhi34_14"
[793] "lhi34_15"     "lhi34_16"     "lhi34_17"
[796] "lhi34_18"     "lhi34_19"     "lhi34_20"
[799] "lhi35_1"      "lhi35_2"      "lhi35_3"
[802] "lhi35_4"      "lhi35_5"      "lhi35_6"
[805] "lhi35_7"      "lhi35_8"      "lhi35_9"
[808] "lhi35_10"     "lhi35_11"     "lhi35_12"
[811] "lhi35_13"     "lhi35_14"     "lhi35_15"
[814] "lhi35_16"     "lhi35_17"     "lhi35_18"
[817] "lhi35_19"     "lhi35_20"     "lhi36_1"
[820] "lhi36_2"      "lhi36_3"      "lhi36_4"
[823] "lhi36_5"      "lhi36_6"      "lhi36_7"
[826] "lhi36_8"      "lhi36_9"      "lhi36_10"
[829] "lhi36_11"     "lhi36_12"     "lhi36_13"
[832] "lhi36_14"     "lhi36_15"     "lhi36_16"
[835] "lhi36_17"     "lhi36_18"     "lhi36_19"
[838] "lhi36_20"     "lhi37_1"      "lhi37_2"
[841] "lhi37_3"      "lhi37_4"      "lhi37_5"
[844] "lhi37_6"      "lhi37_7"      "lhi37_8"
[847] "lhi37_9"      "lhi37_10"     "lhi37_11"
[850] "lhi37_12"     "lhi37_13"     "lhi37_14"
[853] "lhi37_15"     "lhi37_16"     "lhi37_17"
[856] "lhi37_18"     "lhi37_19"     "lhi37_20"
[859] "lhi38_1"      "lhi38_2"      "lhi38_3"
[862] "lhi38_4"      "lhi38_5"      "lhi38_6"
[865] "lhi38_7"      "lhi38_8"      "lhi38_9"
[868] "lhi38_10"     "lhi38_11"     "lhi38_12"
[871] "lhi38_13"     "lhi38_14"     "lhi38_15"
[874] "lhi38_16"     "lhi38_17"     "lhi38_18"
[877] "lhi38_19"     "lhi38_20"     "lhi39a_1"
[880] "lhi39a_2"     "lhi39a_3"     "lhi39a_4"
[883] "lhi39a_5"     "lhi39a_6"     "lhi39a_7"
[886] "lhi39a_8"     "lhi39a_9"     "lhi39a_10"
```

```
 [889] "lhi39a_11"          "lhi39a_12"          "lhi39a_13"
 [892] "lhi39a_14"          "lhi39a_15"          "lhi39a_16"
 [895] "lhi39a_17"          "lhi39a_18"          "lhi39a_19"
 [898] "lhi39a_20"          "lhi39au_1"          "lhi39au_2"
 [901] "lhi39au_3"          "lhi39au_4"          "lhi39au_5"
 [904] "lhi39au_6"          "lhi39au_7"          "lhi39au_8"
 [907] "lhi39au_9"          "lhi39au_10"         "lhi39au_11"
 [910] "lhi39au_12"         "lhi39au_13"         "lhi39au_14"
 [913] "lhi39au_15"         "lhi39au_16"         "lhi39au_17"
 [916] "lhi39au_18"         "lhi39au_19"         "lhi39au_20"
 [919] "lhi39b_1"           "lhi39b_2"           "lhi39b_3"
 [922] "lhi39b_4"           "lhi39b_5"           "lhi39b_6"
 [925] "lhi39b_7"           "lhi39b_8"           "lhi39b_9"
 [928] "lhi39b_10"          "lhi39b_11"          "lhi39b_12"
 [931] "lhi39b_13"          "lhi39b_14"          "lhi39b_15"
 [934] "lhi39b_16"          "lhi39b_17"          "lhi39b_18"
 [937] "lhi39b_19"          "lhi39b_20"          "lhi39bu_1"
 [940] "lhi39bu_2"          "lhi39bu_3"          "lhi39bu_4"
 [943] "lhi39bu_5"          "lhi39bu_6"          "lhi39bu_7"
 [946] "lhi39bu_8"          "lhi39bu_9"          "lhi39bu_10"
 [949] "lhi39bu_11"         "lhi39bu_12"         "lhi39bu_13"
 [952] "lhi39bu_14"         "lhi39bu_15"         "lhi39bu_16"
 [955] "lhi39bu_17"         "lhi39bu_18"         "lhi39bu_19"
 [958] "lhi39bu_20"         "lhi40_1"            "lhi40_2"
 [961] "lhi40_3"            "lhi40_4"            "lhi40_5"
 [964] "lhi40_6"            "lhi40_7"            "lhi40_8"
 [967] "lhi40_9"            "lhi40_10"           "lhi40_11"
 [970] "lhi40_12"           "lhi40_13"           "lhi40_14"
 [973] "lhi40_15"           "lhi40_16"           "lhi40_17"
 [976] "lhi40_18"           "lhi40_19"           "lhi40_20"
 [979] "lhi41_1"            "lhi41_2"            "lhi41_3"
 [982] "lhi41_4"            "lhi41_5"            "lhi41_6"
 [985] "lhi41_7"            "lhi41_8"            "lhi41_9"
 [988] "lhi41_10"           "lhi41_11"           "lhi41_12"
 [991] "lhi41_13"           "lhi41_14"           "lhi41_15"
 [994] "lhi41_16"           "lhi41_17"           "lhi41_18"
 [997] "lhi41_19"           "lhi41_20"           "fer01a"
[1000] "fer01b"             "fer01c"             "fer02m"
[1003] "fer02y"             "fer03"              "fer04"
[1006] "fer04b"             "fer04c"             "fer04d"
[1009] "fer04e"             "fer05"              "fer06"
[1012] "fer07_1"            "fer07_2"            "fer07_3"
[1015] "fer07_4"            "fer07_5"            "fer07_6"
```

```
[1018] "fer07_7"          "fer07_8"          "fer07_9"
[1021] "fer07_10"         "fer08"            "fer09"
[1024] "fer10a"           "fer10bm"          "fer10by"
[1027] "fer11_1"          "fer11_2"          "fer11_3"
[1030] "fer11_4"          "fer11_5"          "fer11_6"
[1033] "fer11_7"          "fer11_8"          "fer12_1"
[1036] "fer12_2"          "fer12_3"          "fer12_4"
[1039] "fer12_5"          "fer12_6"          "fer12_7"
[1042] "fer12_8"          "fer12_9"          "fer12_10"
[1045] "fer12_11"         "fer12_12"         "fer12_13"
[1048] "fer12_14"         "fer13"            "fer14"
[1051] "fer15"            "fer16a"           "fer16b"
[1054] "fer16c"           "fer17"            "fer21"
[1057] "fer22"            "fer23"            "fer24"
[1060] "fer25a"           "fer25b"           "fer25c"
[1063] "fer25d"           "fer25e"           "fer25f"
[1066] "fer26a"           "fer26b"           "fer26e"
[1069] "fer26f"           "fer26h"           "fer27a"
[1072] "fer27b"           "fer27c"           "fer28"
[1075] "fer29"            "hhd01a"           "hhd01b"
[1078] "hhd03_1"          "hhd03_2"          "hhd03_3"
[1081] "hhd03_4"          "hhd03_5"          "hhd03_6"
[1084] "hhd03_7"          "hhd03_8"          "hhd03_9"
[1087] "hhd03_10"         "hhd03_11"         "hhd03_12"
[1090] "hhd03_13"         "hhd03_14"         "hhd03_15"
[1093] "hhd03_16"         "hhd03_17"         "hhd03_18"
[1096] "hhd03_19"         "hhd03_20"         "hhd04_1"
[1099] "hhd04_2"          "hhd04_3"          "hhd04_4"
[1102] "hhd04_5"          "hhd04_6"          "hhd04_7"
[1105] "hhd04_8"          "hhd04_9"          "hhd04_10"
[1108] "hhd04_11"         "hhd04_12"         "hhd04_13"
[1111] "hhd04_14"         "hhd04_15"         "hhd04_16"
[1114] "hhd04_17"         "hhd04_18"         "hhd04_19"
[1117] "hhd04_20"         "hhd05_1"          "hhd05_2"
[1120] "hhd05_3"          "hhd05_4"          "hhd05_5"
[1123] "hhd05_6"          "hhd05_7"          "hhd05_8"
[1126] "hhd05_9"          "hhd05_10"         "hhd05_11"
[1129] "hhd05_12"         "hhd05_13"         "hhd05_14"
[1132] "hhd05_15"         "hhd05_16"         "hhd05_17"
[1135] "hhd05_18"         "hhd05_19"         "hhd05_20"
[1138] "hhd06_m1"         "hhd06_m2"         "hhd06_m3"
[1141] "hhd06_m4"         "hhd06_m5"         "hhd06_m6"
[1144] "hhd06_m7"         "hhd06_m8"         "hhd06_m9"
```

```
[1147] "hhd06_m10"          "hhd06_m11"          "hhd06_m12"
[1150] "hhd06_m13"          "hhd06_m14"          "hhd06_m15"
[1153] "hhd06_m16"          "hhd06_m17"          "hhd06_m18"
[1156] "hhd06_m19"          "hhd06_m20"          "hhd06_y1"
[1159] "hhd06_y2"           "hhd06_y3"           "hhd06_y4"
[1162] "hhd06_y5"           "hhd06_y6"           "hhd06_y7"
[1165] "hhd06_y8"           "hhd06_y9"           "hhd06_y10"
[1168] "hhd06_y11"          "hhd06_y12"          "hhd06_y13"
[1171] "hhd06_y14"          "hhd06_y15"          "hhd06_y16"
[1174] "hhd06_y17"          "hhd06_y18"          "hhd06_y19"
[1177] "hhd06_y20"          "hhd07_1"            "hhd07_2"
[1180] "hhd07_3"            "hhd07_4"            "hhd07_5"
[1183] "hhd07_6"            "hhd07_7"            "hhd07_8"
[1186] "hhd07_9"            "hhd07_10"           "hhd07_11"
[1189] "hhd07_12"           "hhd07_13"           "hhd07_14"
[1192] "hhd07_15"           "hhd07_16"           "hhd07_17"
[1195] "hhd07_18"           "hhd07_19"           "hhd07_20"
[1198] "hhd08_1"            "hhd08_2"            "hhd08_3"
[1201] "hhd08_4"            "hhd08_5"            "hhd08_6"
[1204] "hhd08_7"            "hhd08_8"            "hhd08_9"
[1207] "hhd08_10"           "hhd08_11"           "hhd08_12"
[1210] "hhd08_13"           "hhd08_14"           "hhd08_15"
[1213] "hhd08_16"           "hhd08_17"           "hhd08_18"
[1216] "hhd08_19"           "hhd08_20"           "hhd09_1"
[1219] "hhd09_2"            "hhd09_3"            "hhd09_4"
[1222] "hhd09_5"            "hhd09_6"            "hhd09_7"
[1225] "hhd09_8"            "hhd09_9"            "hhd09_10"
[1228] "hhd09_11"           "hhd09_12"           "hhd09_13"
[1231] "hhd09_14"           "hhd09_15"           "hhd09_16"
[1234] "hhd09_17"           "hhd09_18"           "hhd09_19"
[1237] "hhd09_20"           "hhd11a"             "hhd11b"
[1240] "hhd11c"             "hhd11d"             "hhd11e"
[1243] "hhd11f"             "hhd12"              "hhd13a"
[1246] "hhd13b"             "hhd13c"             "hhd13d"
[1249] "hhd13e"             "hhd14"              "hhd15a"
[1252] "hhd15b"             "hhd15c"             "hhd15d"
[1255] "hhd16"              "hhd17"              "hhd18"
[1258] "hhd19_1"            "hhd19_2"            "hhd19_3"
[1261] "hhd19_4"            "hhd19_5"            "hhd19_6"
[1264] "hhd19_7"            "hhd19_8"            "hhd19_9"
[1267] "hhd19_10"           "hhd19_11"           "hhd19_12"
[1270] "hhd19_13"           "hhd19_14"           "hhd19_15"
[1273] "hhd19_16"           "hhd19_17"           "hhd19_18"
```

```
[1276] "hhd19_19"     "hhd19_20"     "hhd19_21"
[1279] "hhd19_22"     "hhd20"        "hhd20u"
[1282] "hhd21"        "hhd22"        "hhd23_1"
[1285] "hhd23_2"      "hhd23_3"      "hhd23_4"
[1288] "hhd23_5"      "hhd23_6"      "hhd24"
[1291] "hhd24a"       "hhd24u"       "hhd25"
[1294] "hhd26_1"      "hhd26_2"      "hhd26_3"
[1297] "hhd26_4"      "hhd26_5"      "hhd26_6"
[1300] "hhd26_7"      "hhd26_8"      "hhd26_9"
[1303] "hhd26_10"     "hhd26_11"     "hhd26_12"
[1306] "hhd26_13"     "hhd26_14"     "hhd26_15"
[1309] "hhd26_16"     "hhd26_17"     "hhd26_18"
[1312] "hhd26_19"     "hhd26_20"     "hhd26_21"
[1315] "hhd26_22"     "hhd27"        "hhd27u"
[1318] "hhd28"        "hhd29_1"      "hhd29_2"
[1321] "hhd29_3"      "hhd29_4"      "hhd29_5"
[1324] "hhd29_6"      "hhd29_7"      "hhd29_8"
[1327] "hhd29_9"      "hhd29_10"     "hhd29_11"
[1330] "hhd29_12"     "hhd29_13"     "hhd29_14"
[1333] "hhd29_15"     "hhd29_16"     "hhd29_17"
[1336] "hhd29_18"     "hhd29_19"     "hhd29_20"
[1339] "hhd29_21"     "hhd29_22"     "hhd30"
[1342] "hhd30u"       "hhd31"        "hhd35"
[1345] "hhd36_1"      "hhd36_2"      "hhd36_3"
[1348] "hhd36_4"      "hhd36_5"      "hhd36_6"
[1351] "hhd36_7"      "hhd36_8"      "hhd36_9"
[1354] "hhd36_10"     "hhd36_11"     "hhd36_12"
[1357] "hhd36_13"     "hhd36_14"     "hhd36_15"
[1360] "hhd36_16"     "hhd36_17"     "hhd36_18"
[1363] "hhd36_19"     "hhd36_20"     "hhd36_21"
[1366] "hhd36_22"     "gen01"        "gen02"
[1369] "gen03"        "gen09m"       "gen09y"
[1372] "gen10m"       "gen10y"       "gen11"
[1375] "gen12iso"     "gen15a"       "gen15au"
[1378] "gen15b"       "gen15bu"      "gen16"
[1381] "gen23m"       "gen23y"       "gen24m"
[1384] "gen24y"       "gen25"        "gen26iso"
[1387] "gen29a"       "gen29au"      "gen29b"
[1390] "gen29bu"      "gen30"        "gen37a"
[1393] "gen37m"       "gen37y"       "gen38a"
[1396] "gen38bm"      "gen38by"      "gen39a"
[1399] "gen39b"       "gen40"        "gen41a"
[1402] "gen41a_4001"  "gen41b"       "gen41b_4001"
```

```
[1405] "gen42"          "gen43"          "gen44aiso"
[1408] "gen44b"         "gen45"          "gen46"
[1411] "gen47"          "gen48"          "gen48isco"
[1414] "gen49"          "gen49isced"     "gen50"
[1417] "gen50isco"      "gen51"          "gen51isced"
[1420] "gen52"          "gen52am"        "gen52ay"
[1423] "gen53"          "gen54"          "gen55"
[1426] "gen56"          "gen57m"         "gen57y"
[1429] "gen58"          "gen59"          "gen60_1"
[1432] "gen60_2"        "gen60_3"        "gen60_4"
[1435] "gen60_5"        "gen60_6"        "gen60_7"
[1438] "gen60_8"        "gen60_9"        "gen60_10"
[1441] "gen60_11"       "gen60_12"       "gen60_13"
[1444] "gen60_14"       "gen60_15"       "gen60_16"
[1447] "gen60_17"       "gen60_18"       "gen60_19"
[1450] "gen60_20"       "gen60_21"       "gen60_22"
[1453] "gen63"          "gen66"          "gen67_1"
[1456] "gen67_2"        "gen67_3"        "gen67_4"
[1459] "gen67_5"        "gen67_6"        "gen67_7"
[1462] "gen67_8"        "gen67_9"        "gen67_10"
[1465] "gen67_11"       "gen67_12"       "gen67_13"
[1468] "gen67_14"       "gen67_15"       "gen67_16"
[1471] "gen67_17"       "gen67_18"       "gen67_19"
[1474] "gen67_20"       "gen67_21"       "gen67_22"
[1477] "gen68"          "gen69_1"        "gen69_2"
[1480] "gen69_3"        "gen69_4"        "gen69_5"
[1483] "gen69_6"        "gen69_7"        "gen69_8"
[1486] "gen69_9"        "gen69_10"       "gen69_11"
[1489] "gen69_12"       "gen69_13"       "gen69_14"
[1492] "gen69_15"       "gen69_16"       "gen69_17"
[1495] "gen69_18"       "gen69_19"       "gen69_20"
[1498] "gen69_21"       "gen69_22"       "gen70"
[1501] "gen71_1"        "gen71_2"        "gen71_3"
[1504] "gen71_4"        "gen71_5"        "gen71_6"
[1507] "gen71_7"        "gen71_8"        "gen71_9"
[1510] "gen71_10"       "gen71_11"       "gen71_12"
[1513] "gen71_13"       "gen71_14"       "gen71_15"
[1516] "gen71_16"       "gen71_17"       "gen71_18"
[1519] "gen71_19"       "gen71_20"       "gen71_21"
[1522] "gen71_22"       "wel01"          "wel02"
[1525] "wel02a"         "wel03_1"        "wel03_2"
[1528] "wel03_3"        "wel03_4"        "wel03_5"
[1531] "wel03_6"        "wel03_7"        "wel03_8"
```

```
[1534] "wel03_9"          "wel03_10"         "wel03_11"
[1537] "wel03_12"         "wel03_13"         "wel03_14"
[1540] "wel03_15"         "wel03_16"         "wel03_17"
[1543] "wel03_18"         "wel03_19"         "wel03_20"
[1546] "wel04"            "wel05"            "wel06"
[1549] "wel07"            "wel08"            "wel09a"
[1552] "wel09b"           "wel09c"           "wel09d"
[1555] "wel09e"           "wel09f"           "wel10_1"
[1558] "wel10_2"          "wel10_3"          "wel10_4"
[1561] "wel10_5"          "wel10_6"          "wel10_7"
[1564] "wel10_8"          "wel10_9"          "wel10_10"
[1567] "wel10_11"         "wel10_12"         "wel10_13"
[1570] "wel10_14"         "wel10_15"         "wel10_16"
[1573] "wel10_17"         "wel10_18"         "wel10_19"
[1576] "wel10_20"         "wel10_21"         "wel10_22"
[1579] "wel11a"           "wel11b"           "wel11c"
[1582] "wel11d"           "wel11e"           "wel14a_4001"
[1585] "wel14b_4001"      "wel14c_4001"      "wel14d_4001"
[1588] "wel14e_4001"      "wel14f_4001"      "wel14g_4001"
[1591] "wel16a_4001"      "wel16b_4001"      "wel16c_1_4001"
[1594] "wel16c_2_4001"    "wel16c_3_4001"    "wel16c_4_4001"
[1597] "wel16c_5_4001"    "wel16c_6_4001"    "wel16c_7_4001"
[1600] "wel16c_8_4001"    "wel16c_9_4001"    "wel16c_10_4001"
[1603] "wel16c_11_4001"   "wel16c_12_4001"   "wel16c_13_4001"
[1606] "wel16c_14_4001"   "wel16c_15_4001"   "wel16c_16_4001"
[1609] "wel16c_17_4001"   "wel16c_18_4001"   "wel16c_19_4001"
[1612] "wel16c_20_4001"   "wel16d_m1_4001"   "wel16d_m2_4001"
[1615] "wel16d_m3_4001"   "wel16d_m4_4001"   "wel16d_m5_4001"
[1618] "wel16d_m6_4001"   "wel16d_m7_4001"   "wel16d_m8_4001"
[1621] "wel16d_m9_4001"   "wel16d_m10_4001"  "wel16d_m11_4001"
[1624] "wel16d_m12_4001"  "wel16d_m13_4001"  "wel16d_m14_4001"
[1627] "wel16d_m15_4001"  "wel16d_m16_4001"  "wel16d_m17_4001"
[1630] "wel16d_m18_4001"  "wel16d_m19_4001"  "wel16d_m20_4001"
[1633] "wel16d_y1_4001"   "wel16d_y2_4001"   "wel16d_y3_4001"
[1636] "wel16d_y4_4001"   "wel16d_y5_4001"   "wel16d_y6_4001"
[1639] "wel16d_y7_4001"   "wel16d_y8_4001"   "wel16d_y9_4001"
[1642] "wel16d_y10_4001"  "wel16d_y11_4001"  "wel16d_y12_4001"
[1645] "wel16d_y13_4001"  "wel16d_y14_4001"  "wel16d_y15_4001"
[1648] "wel16d_y16_4001"  "wel16d_y17_4001"  "wel16d_y18_4001"
[1651] "wel16d_y19_4001"  "wel16d_y20_4001"  "wel16a_1_4002"
[1654] "wel16a_2_4002"    "wel16a_3_4002"    "wel16a_4_4002"
[1657] "wel16a_5_4002"    "wel16a_6_4002"    "wel16a_7_4002"
[1660] "wel16a_8_4002"    "wel16a_9_4002"    "wel16a_10_4002"
```

```
[1663] "wel16a_11_4002"    "wel16a_12_4002"    "wel16a_13_4002"
[1666] "wel16a_14_4002"    "wel16a_15_4002"    "wel16a_16_4002"
[1669] "wel16a_17_4002"    "wel16a_18_4002"    "wel16a_19_4002"
[1672] "wel16a_20_4002"    "wel16b_1_4002"     "wel16b_2_4002"
[1675] "wel16b_3_4002"     "wel16b_4_4002"     "wel16b_5_4002"
[1678] "wel16b_6_4002"     "wel16b_7_4002"     "wel16b_8_4002"
[1681] "wel16b_9_4002"     "wel16b_10_4002"    "wel16b_11_4002"
[1684] "wel16b_12_4002"    "wel16b_13_4002"    "wel16b_14_4002"
[1687] "wel16b_15_4002"    "wel16b_16_4002"    "wel16b_17_4002"
[1690] "wel16b_18_4002"    "wel16b_19_4002"    "wel16b_20_4002"
[1693] "wel16c_1_4002"     "wel16c_2_4002"     "wel16c_3_4002"
[1696] "wel16c_4_4002"     "wel16c_5_4002"     "wel16c_6_4002"
[1699] "wel16c_7_4002"     "wel16c_8_4002"     "wel16c_9_4002"
[1702] "wel16c_10_4002"    "wel16c_11_4002"    "wel16c_12_4002"
[1705] "wel16c_13_4002"    "wel16c_14_4002"    "wel16c_15_4002"
[1708] "wel16c_16_4002"    "wel16c_17_4002"    "wel16c_18_4002"
[1711] "wel16c_19_4002"    "wel16c_20_4002"    "wel16d_1_4002"
[1714] "wel16d_2_4002"     "wel16d_3_4002"     "wel16d_4_4002"
[1717] "wel16d_5_4002"     "wel16d_6_4002"     "wel16d_7_4002"
[1720] "wel16d_8_4002"     "wel16d_9_4002"     "wel16d_10_4002"
[1723] "wel16d_11_4002"    "wel16d_12_4002"    "wel16d_13_4002"
[1726] "wel16d_14_4002"    "wel16d_15_4002"    "wel16d_16_4002"
[1729] "wel16d_17_4002"    "wel16d_18_4002"    "wel16d_19_4002"
[1732] "wel16d_20_4002"    "wel16e_1_4002"     "wel16e_2_4002"
[1735] "wel16e_3_4002"     "wel16e_4_4002"     "wel16e_5_4002"
[1738] "wel16e_6_4002"     "wel16e_7_4002"     "wel16e_8_4002"
[1741] "wel16e_9_4002"     "wel16e_10_4002"    "wel16e_11_4002"
[1744] "wel16e_12_4002"    "wel16e_13_4002"    "wel16e_14_4002"
[1747] "wel16e_15_4002"    "wel16e_16_4002"    "wel16e_17_4002"
[1750] "wel16e_18_4002"    "wel16e_19_4002"    "wel16e_20_4002"
[1753] "wel16f_1_4002"     "wel16f_2_4002"     "wel16f_3_4002"
[1756] "wel16f_4_4002"     "wel16f_5_4002"     "wel16f_6_4002"
[1759] "wel16f_7_4002"     "wel16f_8_4002"     "wel16f_9_4002"
[1762] "wel16f_10_4002"    "wel16f_11_4002"    "wel16f_12_4002"
[1765] "wel16f_13_4002"    "wel16f_14_4002"    "wel16f_15_4002"
[1768] "wel16f_16_4002"    "wel16f_17_4002"    "wel16f_18_4002"
[1771] "wel16f_19_4002"    "wel16f_20_4002"    "wel16g_1_4002"
[1774] "wel16g_2_4002"     "wel16g_3_4002"     "wel16g_4_4002"
[1777] "wel16g_5_4002"     "wel16g_6_4002"     "wel16g_7_4002"
[1780] "wel16g_8_4002"     "wel16g_9_4002"     "wel16g_10_4002"
[1783] "wel16g_11_4002"    "wel16g_12_4002"    "wel16g_13_4002"
[1786] "wel16g_14_4002"    "wel16g_15_4002"    "wel16g_16_4002"
[1789] "wel16g_17_4002"    "wel16g_18_4002"    "wel16g_19_4002"
```

```
[1792] "wel16g_20_4002"     "wrk01"              "wrk02"
[1795] "wrk03m"             "wrk03y"             "wrk04"
[1798] "wrk04isco"          "wrk06"              "wrk07"
[1801] "wrk08"              "wrk09"              "wrk10"
[1804] "wrk11"              "wrk12"              "wrk13"
[1807] "wrk14"              "wrk15a"             "wrk15b"
[1810] "wrk15c"             "wrk15d"             "wrk16a"
[1813] "wrk16b"             "wrk17"              "wrk18"
[1816] "wrk20"              "wrk21"              "wrk22"
[1819] "wrk23"              "wrk24"              "wrk25"
[1822] "wrk26"              "wrk27"              "wrk27isco"
[1825] "wrk28"              "wrk30"              "wrk30am"
[1828] "wrk30ay"            "wrk31"              "wrk32"
[1831] "wrk34"              "wrk34isco"          "wrk35"
[1834] "wrk36"              "wrk37"              "wrk38"
[1837] "wrk39"              "wrk40"              "wrk41"
[1840] "wrk42"              "wrk43"              "wrk44"
[1843] "wrk46"              "wrk47"              "wrk48"
[1846] "wrk49"              "wrk50"              "wrk51_4001"
[1849] "wrk51_4002"         "wrk51_4003"         "wrk51_4004"
[1852] "wrk51a_4005"        "wrk51b_4005"        "wrk51_4006"
[1855] "wrk51a_4007"        "wrk51b_4007"        "wrk51a_4008"
[1858] "wrk51b_4008"        "wrk51_4009"         "wrk51_4010"
[1861] "wrk51a_4011"        "wrk51b_4011"        "wrk51_4012"
[1864] "wrk51_4013"         "wrk51_4014"         "wrk51_4015"
[1867] "inc01"              "inc03"              "inc05"
[1870] "inc06"              "inc08_1"            "inc08_2"
[1873] "inc08_3"            "inc08_4"            "inc08_5"
[1876] "inc08_6"            "inc08_7"            "inc08_8"
[1879] "inc08_9"            "inc08_10"           "inc08_11"
[1882] "inc08_12"           "inc09_1"            "inc09_2"
[1885] "inc09_3"            "inc09_4"            "inc09_5"
[1888] "inc09_6"            "inc09_7"            "inc09_8"
[1891] "inc09_9"            "inc09_10"           "inc09_11"
[1894] "inc11_1"            "inc11_2"            "inc11_3"
[1897] "inc11_4"            "inc11_5"            "inc11_6"
[1900] "inc11_7"            "inc11_8"            "inc11_9"
[1903] "inc11_10"           "inc11_11"           "inc12"
[1906] "inc13"              "inc14_1"            "inc14_2"
[1909] "inc14_3"            "inc14_4"            "inc14_5"
[1912] "inc14_6"            "inc14_7"            "inc14_8"
[1915] "inc14_9"            "inc14_10"           "inc14_11"
[1918] "inc14_12"           "inc14_13"           "inc14_14"
```

```
[1921] "inc14_15"         "inc14_16"         "inc14_17"
[1924] "inc14_18"         "inc14_19"         "inc14_20"
[1927] "inc14_21"         "inc14_22"         "inc15"
[1930] "att01"            "att02"            "att03a"
[1933] "att03b"           "att03d"           "att03e"
[1936] "att03g"           "att03h"           "att03i"
[1939] "att03j"           "att05b"           "att06a"
[1942] "att06b"           "att07a"           "att07b"
[1945] "att07c"           "att07d"           "att07g"
[1948] "att08"            "att09"            "att09u"
[1951] "att10"            "att11b"           "att11d"
[1954] "att13a_4001"      "att13b_4001"      "att13c_4001"
[1957] "att13d_4001"      "att13e_4001"      "att13f_4001"
[1960] "att13g_4001"      "att13h_4001"      "att13_4002"
[1963] "att13_4003"       "att13_4004"       "att13_4005"
[1966] "att13_1_4006"     "att13_2_4006"     "att13_3_4006"
[1969] "att13_4_4006"     "att13_5_4006"     "att13_6_4006"
[1972] "att13_7_4006"     "att13_8_4006"     "att13_9_4006"
[1975] "att13_4007"       "att19a_4001"      "att19b_4001"
[1978] "att19c_4001"      "rep01"            "rep02"
[1981] "rep03_1"          "rep03_2"          "rep03_3"
[1984] "rep03_4"          "rep04"            "rep05"
[1987] "rep06"            "flag1"            "localitysize_4001"
[1990] "department_4001"  "city_4001"
```

2 Evaluación de información y pirámides

2.1 Paquetes

```
if (!require("pacman")) install.packages("pacman") # instala pacman si se requiere
```

```
Cargando paquete requerido: pacman
```

```
pacman::p_load(tidyverse,
               readxl,
               writexl,
               haven,
               sjlabelled,
               foreign,
               janitor,
               remotes,
               wppExplorer,
               apyramid,
               fmsb)
```

2.2 Instalación de paquetes en desarrollo, reprisse

Esto puede tardar un ratito

```
install.packages("rstan", repos = c("https://mc-stan.org/r-packages/", getOption("repos"))
remotes::install_github("timriffe/DemoTools")

library("DemoTools")

remotes::install_github("PPgp/wpp2022")
```

- Si pide actualizar darle 1, de "All

- o 2 de "CRAN only"

```
Cargando paquete requerido: Rcpp

Cargando paquete requerido: data.table


Adjuntando el paquete: 'data.table'

The following objects are masked from 'package:lubridate':

    hour, isoweek, mday, minute, month, quarter, second, wday, week,
    yday, year

The following objects are masked from 'package:dplyr':

    between, first, last

The following object is masked from 'package:purrr':

    transpose
```

2.3 Datos

2.3.1 {wpp2022}

Vamos a utilizar datos del paquete `{wpp2022}` . Revisemos la viñeta del paquete que está aquí

Todas los *data.frames* están en el paquete y si lo tenemos cargados podemos consultarlo con el comando data()

```r
data("popAge5dt")
data("popprojAge5dt")
```

** En caso que no tengas disponible, puedes descargar la información de la carpeta datos

```r
load("datos/wpp2022.RData")
```

Aquí están todos los países, revisemos un poco

```
popAge5dt %>%
  dplyr::select(country_code, name) %>%
  unique()
```

```
     country_code                             name
            <int>                           <char>
  1:          900                            World
  2:         1834               Sub-Saharan Africa
  3:         1833 Northern Africa and Western Asia
  4:         1831        Central and Southern Asia
  5:         1832   Eastern and South-Eastern Asia
 ---
281:          882                            Samoa
282:          772                          Tokelau
283:          776                            Tonga
284:          798                           Tuvalu
285:          876        Wallis and Futuna Islands
```

Podemos hacer búsquedas:

```
popAge5dt %>%
  mutate(uy=stringr::str_detect(name, "Uruguay")) %>%
  filter(uy) %>%
  select(country_code, name)
```

```
     country_code    name
            <int>  <char>
  1:          858 Uruguay
  2:          858 Uruguay
  3:          858 Uruguay
  4:          858 Uruguay
  5:          858 Uruguay
 ---
311:          858 Uruguay
312:          858 Uruguay
313:          858 Uruguay
314:          858 Uruguay
315:          858 Uruguay
```

Vamos a hacer el ejercicio con Uruguay pero pueden buscar cualquier otro país y la región

```r
# Países:
# uy: 858
# sv: 222
# gt: 320
# hn: 340
# mx: 484
# CA: 916
# LAC: 1830

popAge1dt<- popAge1dt %>%
  filter(country_code%in%c(858,1830))
```

También, tengo datos de algunos censos, descargados de ipums

2.4 De IPUMS

```r
readxl::read_excel("datos/censos_p2.xlsx",
                   sheet = "El Salvador 1992" ) %>% #ojo con este argumento
  head() %>%
  janitor::clean_names() # checa qué hace
```

```
# A tibble: 6 x 4
  age    male female unknown
  <chr> <dbl>  <dbl>   <dbl>
1 0      6093   6113       0
2 1      6089   5795       0
3 2      6805   6737       0
4 3      7028   6699       0
5 4      7294   6965       0
6 5      6628   6408       0
```

Usaremos esta tabla de datos agregados para **crear variables**. Esto se hace con el comando `dplyr::mutate()`

```r
sv1992<-readxl::read_excel("datos/censos_p2.xlsx", sheet = "El Salvador 1992") %>%
  janitor::clean_names() %>% #
  dplyr::mutate(total=male + female) %>% # ojo
  dplyr::mutate(age=as.numeric(age)) #ojo
```

2.5 {fmsb} Atracción digital

Este paquete tiene cosas muy interesantes. Es un paquete no sólo para demografía pero permite ajustar algunas funciones demográficas

Limitantes: como que está en japonés :P

Un ejemplo con el índice de Whipple, que mide la atracción digital. Necesitamos datos en edades singulares:

Tenemos un archivo en datos con varios censos, para evaluar su información a través de la atracción digital. Revisemos los datos del censo de 1992.

Para ver los totales podemos agregar una fila muy simple con `janitor::adorn_totals(where="row")`

```
sv1992 %>%
  janitor::adorn_totals(where="row")
```

```
 age   male female unknown  total
   0   6093   6113       0  12206
   1   6089   5795       0  11884
   2   6805   6737       0  13542
   3   7028   6699       0  13727
   4   7294   6965       0  14259
   5   6628   6408       0  13036
   6   6906   6570       0  13476
   7   7012   6350       0  13362
   8   6444   6232       0  12676
   9   6086   5884       0  11970
  10   7012   6690       0  13702
  11   6394   6067       0  12461
  12   7955   7415       0  15370
  13   6482   6175       0  12657
  14   6654   6622       0  13276
  15   6617   6717       0  13334
  16   5872   5985       0  11857
  17   5891   6089       0  11980
  18   6073   6274       0  12347
  19   4356   4999       0   9355
  20   5205   5950       0  11155
  21   3812   4526       0   8338
  22   4997   5645       0  10642
  23   4177   4875       0   9052
```

24	4195	4875	0	9070
25	4045	4922	0	8967
26	3816	4453	0	8269
27	3598	4160	0	7758
28	3657	4254	0	7911
29	3023	3341	0	6364
30	4533	4886	0	9419
31	2139	2612	0	4751
32	3490	4025	0	7515
33	2599	3024	0	5623
34	2368	2765	0	5133
35	2996	3456	0	6452
36	2590	2859	0	5449
37	2325	2667	0	4992
38	2627	2936	0	5563
39	1985	2217	0	4202
40	3352	3548	0	6900
41	1433	1649	0	3082
42	2851	3003	0	5854
43	1691	1962	0	3653
44	1576	1763	0	3339
45	2329	2497	0	4826
46	1490	1760	0	3250
47	1716	1811	0	3527
48	1807	2105	0	3912
49	1276	1392	0	2668
50	2319	2597	0	4916
51	986	1151	0	2137
52	1901	2067	0	3968
53	1196	1470	0	2666
54	1198	1456	0	2654
55	1561	1781	0	3342
56	1215	1401	0	2616
57	1078	1238	0	2316
58	1110	1291	0	2401
59	824	943	0	1767
60	2020	2242	0	4262
61	601	779	0	1380
62	1358	1453	0	2811
63	876	1007	0	1883
64	860	952	0	1812
65	1211	1347	0	2558
66	799	931	0	1730

```
   67    755    937        0   1692
   68    779    905        0   1684
   69    514    554        0   1068
   70   1212   1276        0   2488
   71    371    438        0    809
   72    747    923        0   1670
   73    472    562        0   1034
   74    453    480        0    933
   75    649    777        0   1426
   76    385    472        0    857
   77    350    378        0    728
   78    410    533        0    943
   79    240    274        0    514
   80    501    657        0   1158
   81    178    239        0    417
   82    328    390        0    718
   83    164    225        0    389
   84    148    187        0    335
   85    210    294        0    504
   86    146    202        0    348
   87    110    152        0    262
   88    124    155        0    279
   89     93    111        0    204
   90    110    155        0    265
   91     47     53        0    100
   92     67     98        0    165
   93     23     41        0     64
   94     14     27        0     41
   95     17     44        0     61
   96     16     36        0     52
   97     10     19        0     29
   98     71    120        0    191
   99      0      0        0      0
  100      0      0        0      0
  999      0      0        0      0
Total 248216 262544        0 510760
```

```r
sv1992<-readxl::read_excel("datos/censos_p2.xlsx") %>%
  janitor::clean_names() %>% #
  dplyr::mutate(total= male + female) %>%
  dplyr::mutate(age=as.numeric(age))
```

El índice de Whipple

```
sv1992 %>%
  dplyr::filter(!age>64) %>% # Este filtro es importante
  dplyr::count(age, wt=total) %>%  # necesitamos siempre una tabla que se ve así
  head()
```

```
# A tibble: 6 x 2
    age     n
  <dbl> <dbl>
1     0 67733
2     1 66507
3     2 66977
4     3 64758
5     4 65412
6     5 67617
```

```
sv1992 %>%
  dplyr::filter(!age>64) %>% # Este filtro es importante
  dplyr::count(age, wt=male) %>%
 with(
    fmsb::WhipplesIndex(n) # se llama n por la segunda columa de la tabla anterior
    )
```

```
$WI
[1] 102.4858

$JUDGE
[1] "highly accurate"
```

2.5.1 Momento de práctica

** Importa cualquier otro censo y encuentra el índice de Whipple

2.6 Pirámides

as pirámides son parte esencial de lo que llamamos *Demografía estática*, nos cuentan un siglo de historia de las poblaciones

2.6.1 Con grupos quinquenales

Si queremos hacerlo como gráficos de barra, seguramente queremos cortar la variable de edad. Igual este paso es esencial en la vida demográfica:

```
pob_uy<- popAge1dt %>%
  dplyr::filter(name=="Uruguay") %>%
  dplyr::mutate(eda5=cut(age, # la variable a cortar
                breaks=seq(0,110, # El rango válido
                           by=5), # El ancho del intervalo
                include.lowest=T, # para que incluya el valor más bajo dentro del interv
                right=F)) # indica si el intervalo irá abierto en la derecha, ponemos un
```

Veamos esta variable:

```
pob_uy %>%
  count(eda5, wt=pop)
```

```
         eda5          n
       <fctr>      <num>
 1:     [0,5) 18515.960
 2:    [5,10) 18288.987
 3:   [10,15) 17935.920
 4:   [15,20) 17453.353
 5:   [20,25) 16764.673
 6:   [25,30) 15926.536
 7:   [30,35) 15076.952
 8:   [35,40) 14368.985
 9:   [40,45) 13648.200
10:   [45,50) 12754.619
11:   [50,55) 11801.696
12:   [55,60) 10726.383
13:   [60,65)  9461.974
14:   [65,70)  8026.193
15:   [70,75)  6481.370
16:   [75,80)  4815.778
17:   [80,85)  3121.340
18:   [85,90)  1673.401
19:   [90,95)   702.236
20:  [95,100)   215.386
21: [100,105)    52.330
         eda5          n
```

Para que funcione mejor, necesitamos que sexo sea una variable y una columna.

Vamos a utilizar `tidyr::pivot_longer()` para hacer "larga" nuestro data.frame

```
pob_uy %>%
  tidyr::pivot_longer(cols = popM:pop,
                      values_to = "poblacion",
                      names_to = "sexo")
```

```
# A tibble: 22,119 x 7
   country_code name     year   age eda5  sexo  poblacion
          <int> <chr>   <int> <int> <fct> <chr>     <dbl>
 1          858 Uruguay  1949     0 [0,5) popM       22.3
 2          858 Uruguay  1949     0 [0,5) popF       21.4
 3          858 Uruguay  1949     0 [0,5) pop        43.7
 4          858 Uruguay  1949     1 [0,5) popM       21.8
 5          858 Uruguay  1949     1 [0,5) popF       21.3
 6          858 Uruguay  1949     1 [0,5) pop        43.1
 7          858 Uruguay  1949     2 [0,5) popM       21.8
 8          858 Uruguay  1949     2 [0,5) popF       21.5
 9          858 Uruguay  1949     2 [0,5) pop        43.3
10          858 Uruguay  1949     3 [0,5) popM       21.3
# i 22,109 more rows
```

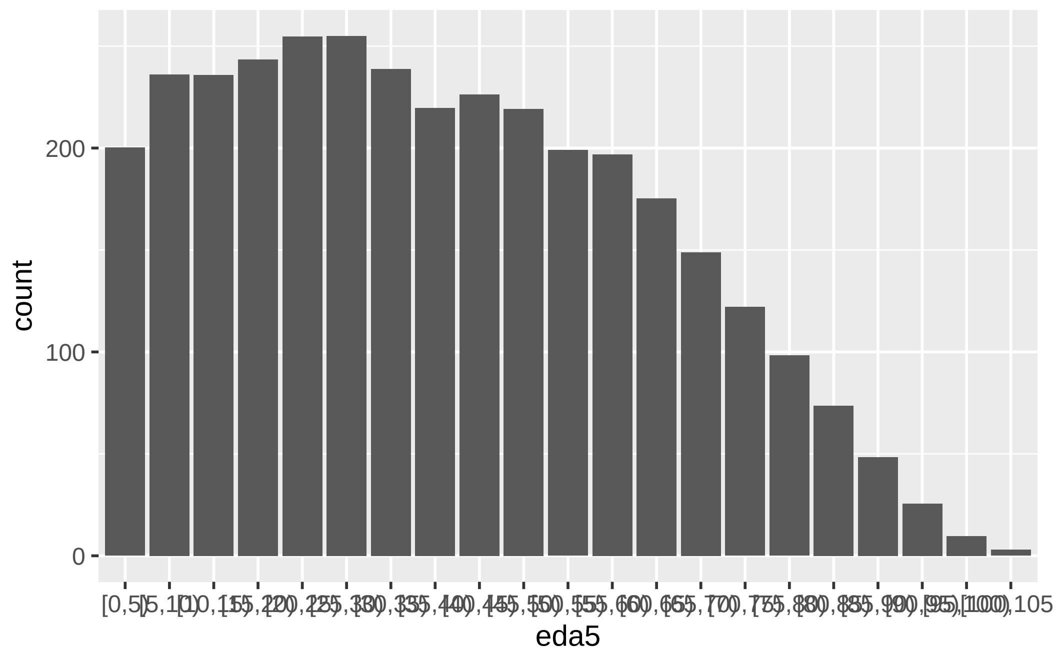
```
pob_uy_long<-pob_uy %>%
  tidyr::pivot_longer(cols = popM:popF,
                      values_to = "poblacion",
                      names_to = "sexo") %>%
  dplyr::select(-pop) # checa este tipo de "anti-selección"
```

2.6.2 Momento de práctica

popAge son los datos históricos, piensa cómo volverías *long* la base de proyecciones *popprojAge*. ¡Checa que hay tipos de proyecciones!

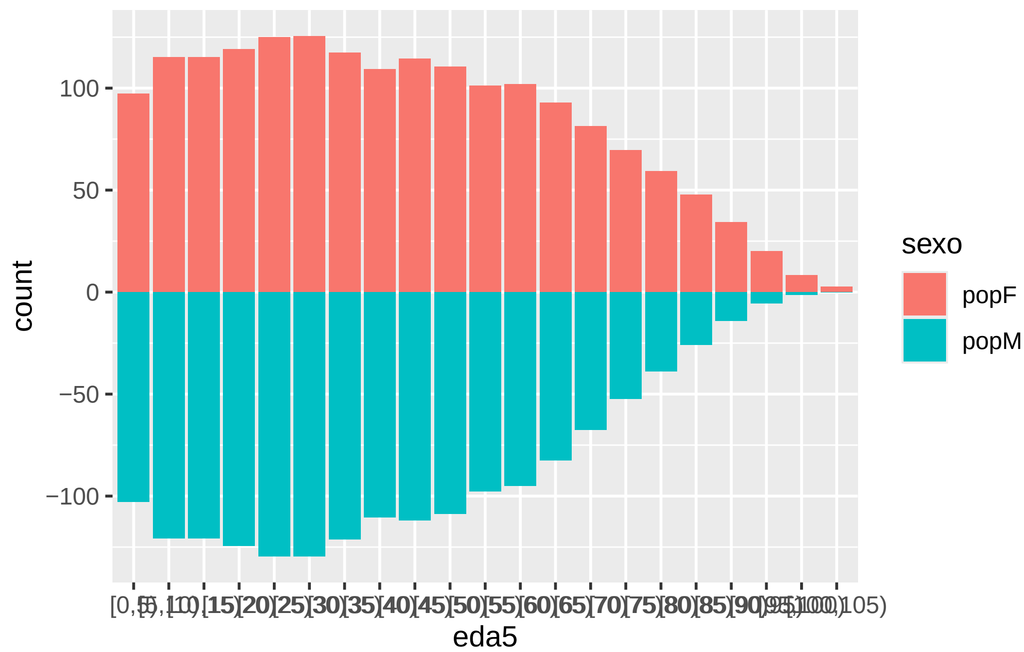
2.6.3 Pirámide en {ggplot2}

```
### gráfico de barras de edades quinquenales
pob_uy_long %>%
  dplyr::filter(year==2020) %>%
  ggplot2::ggplot() +
  aes(x=eda5, weight=poblacion) +
  geom_bar() # dibuja la geometría de barra
```



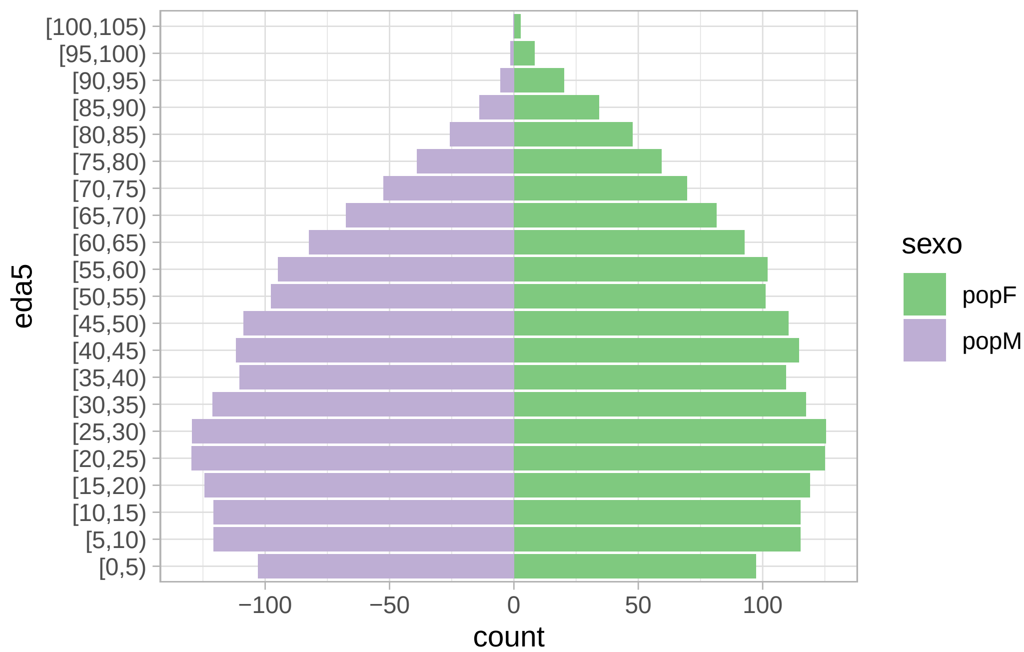
Una pirámide es un doble histograma por **sexo**, donde el valor de los hombres es negativo:

```
pob_uy_long %>%
  dplyr::filter(year==2020) %>%
  dplyr::mutate(poblacion2=if_else(sexo=="popM", -poblacion, poblacion)) %>%
  ggplot2::ggplot() +
  aes(eda5, fill=sexo, weight=poblacion2)+
  geom_bar() # dibuja la geometría de barra
```



Podemos darle la vuelta y cambiarle los colores

```
pob_uy_long <-  pob_uy_long %>%
  mutate(poblacion2=if_else(sexo=="popM", -poblacion, poblacion))


pob_uy_long %>%
  filter(year==2020) %>%
  ggplot(aes(eda5, fill=sexo, weight=poblacion2)) +
  geom_bar() + coord_flip() +
  scale_fill_brewer(palette = "Accent") +
  theme_light()
```



Como que las escalas tampoco están muy perfectas y no queremos las negativa.¡Los hombres no son personas negativas!

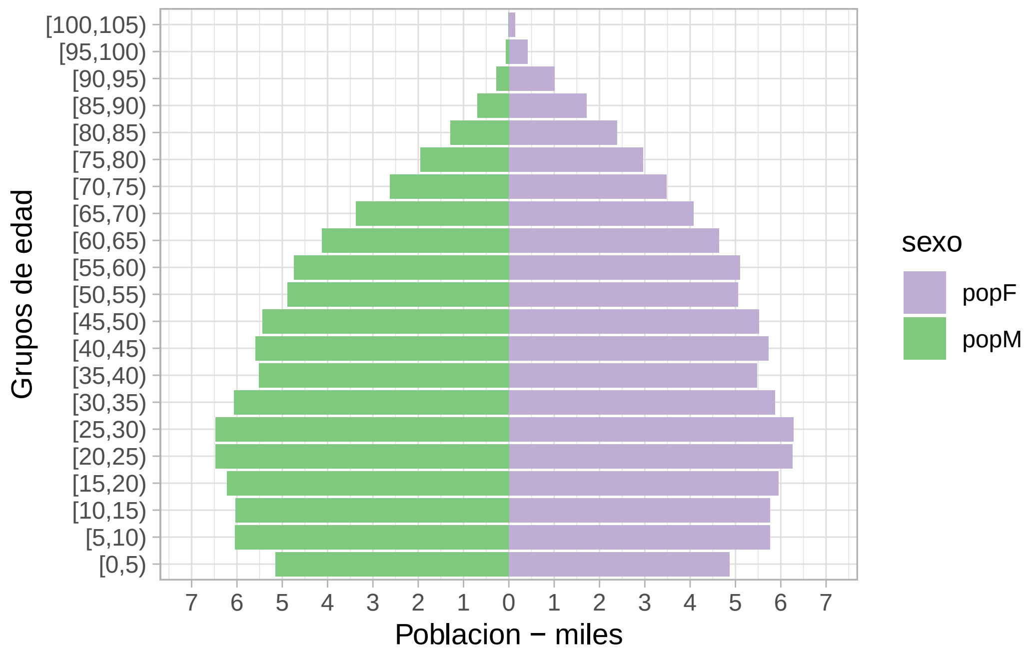
Veamos un poco cómo se comporta esa variable:

```
pob_uy_long %>%
  filter(year==2020) %>%
  count(eda5, sexo, wt=poblacion2) %>%
  summarise(max=max(n), min=min(n))
```

```
# A tibble: 1 x 2
    max   min
  <dbl> <dbl>
1  125. -130.
```

```
pob_uy_long %>%
  filter(year==2020) %>%
  ggplot() +
  aes(eda5, fill=sexo, weight=poblacion2)+
  geom_bar() + coord_flip() +
   scale_y_continuous(breaks = seq(-140, 140, by=20), # cuántos
                      limits = c(-140,140),
```

```
                         labels = paste0(
                          as.character(c(7:0,# sustituye negativos
                            1:7) # Para lo positivo
                            )
                          )
                         )+
  labs(y="Poblacion - miles", x="Grupos de edad") +
  scale_fill_brewer(palette = "Accent", direction = -1) +
  theme_light()
```



Esto es para el volumen de la población ¿Cómo podemos hacer una pirámide que sea en términos de proporciones?

Vamos a necesitar el total de la población:

```
pob_uy_long<- pob_uy_long %>%
  mutate(p_edo=sum(poblacion), .by = year)

head(pob_uy_long)
```

```
# A tibble: 6 x 9
```

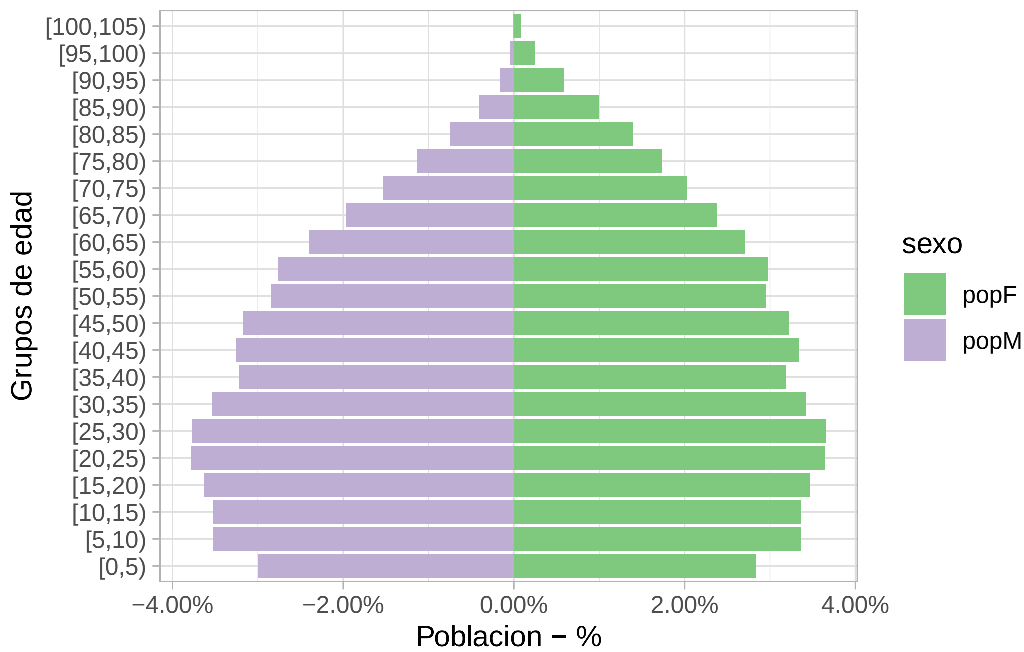
```
  country_code name     year   age eda5  sexo  poblacion poblacion2 p_edo
         <int> <chr>   <int> <int> <fct> <chr>     <dbl>      <dbl> <dbl>
1          858 Uruguay  1949     0 [0,5) popM       22.3      -22.3 2224.
2          858 Uruguay  1949     0 [0,5) popF       21.4       21.4 2224.
3          858 Uruguay  1949     1 [0,5) popM       21.8      -21.8 2224.
4          858 Uruguay  1949     1 [0,5) popF       21.3       21.3 2224.
5          858 Uruguay  1949     2 [0,5) popM       21.8      -21.8 2224.
6          858 Uruguay  1949     2 [0,5) popF       21.5       21.5 2224.
```

Hoy sí haremos lo mismo pero para las proporciones:

```r
pob_uy_long <- pob_uy_long %>%
  mutate(poblacion3=if_else(sexo=="popM",
                            -poblacion/p_edo,
                            poblacion/p_edo))
```

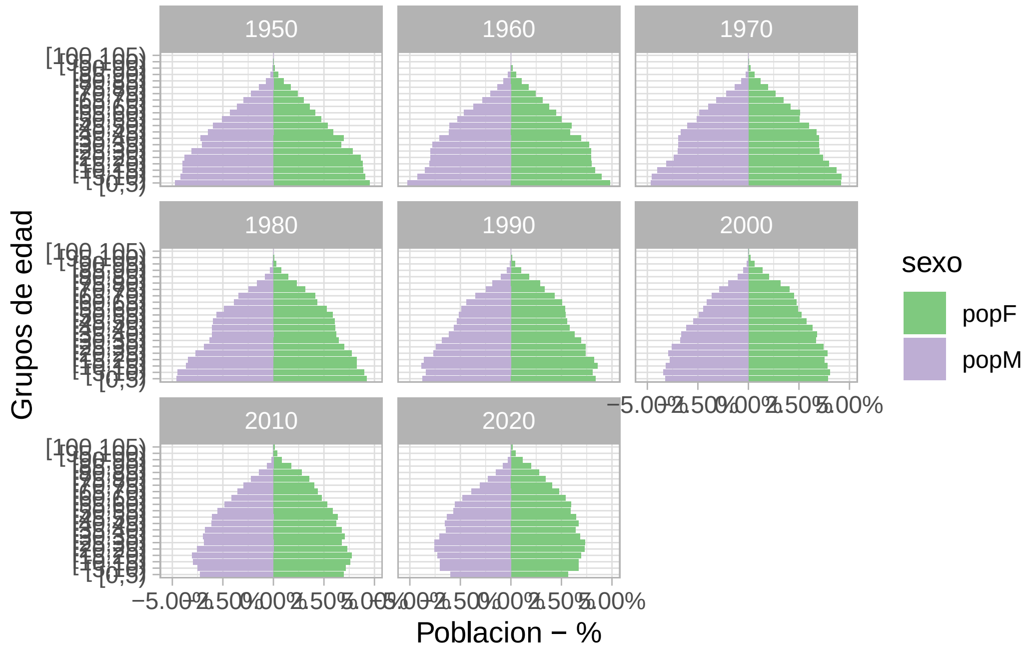
Una vez que ya tenemos nuestra variable proporcional:

```r
pob_uy_long%>%
  filter(year==2020) %>%
  ggplot(aes(eda5, fill=sexo, weight=poblacion3))+
    geom_bar() + coord_flip() +
   scale_y_continuous(labels = scales::percent_format(accuracy=0.01))+
  labs(y="Poblacion - %", x="Grupos de edad") +
  scale_fill_brewer(palette = "Accent") +
  theme_light()
```



Podemos hacer varias pirámides aplicando *facets* o *grids*:

```
pob_uy_long %>%
  filter(year %in% seq(1950,2020, by=10)) %>%
  ggplot() +
  aes(eda5, fill=sexo, weight=poblacion3)+
  geom_bar() + coord_flip() +
  scale_y_continuous(labels = scales::percent_format(accuracy=0.01)) +
  labs(y="Poblacion - %", x="Grupos de edad") +
  scale_fill_brewer(palette = "Accent") +
  theme_light() +
  facet_wrap(~year)
```


1950
1960
1970
1980
1990
2000
2010
2020
Grupos de edad
Poblacion − %
sexo
popF
popM

3 Cont. Pirámides y Lexis

3.1 Instalación local de los paquetes

Antes de empezar esta práctica descargá los siguientes archivos de la carpeta paquetes aquí.

Colocala en tu directorio del proyecto.

Descargá esté código y correlo.

> esto es un problema de IP que tenemos en el curso, si trabajás desde tu casa esto no será problema.

3.2 Paquetes

```
if (!require("pacman")) install.packages("pacman") # instala pacman si se requiere
```

```
Cargando paquete requerido: pacman
```

```
pacman::p_load(tidyverse,
               readxl,
               writexl,
               janitor,
               remotes,
               wppExplorer,
               apyramid,
               magrittr,
               fmsb,
               DemoTools,
               wpp2022,
               LexisPlotR)
```

3.3 Datos

Rehacemos un poco lo que teníamos del día de ayer:

```
sv1992<-readxl::read_excel("datos/censos_p2.xlsx", sheet = "El Salvador 1992") %>%
  janitor::clean_names() %>% #
  dplyr::mutate(total=male + female) %>% # ojo
  dplyr::mutate(age=as.numeric(age)) #ojo
```

Los datos para las pirámides

```
data("popAge5dt")
data("popprojAge5dt")
```

```
pob_uy_long<-popAge1dt %>%
  dplyr::filter(name=="Uruguay") %>%
  dplyr::mutate(eda5=cut(age, # la variable a cortar
                 breaks=seq(0,110, # El rango válido
                            by=5), # El ancho del intervalo
                 include.lowest=T, # para que incluya el valor más bajo dentro del interv
                 right=F)) %>%
  tidyr::pivot_longer(cols = popM:popF,
                      values_to = "poblacion",
                      names_to = "sexo") %>%
  dplyr::select(-pop) # checa este tipo de "anti-selección"
```

3.3.1 Paquete {apyramid}

- Necesita que tengamos los datos quinquenales.
- No acepta funciones en las variables edad y sexo

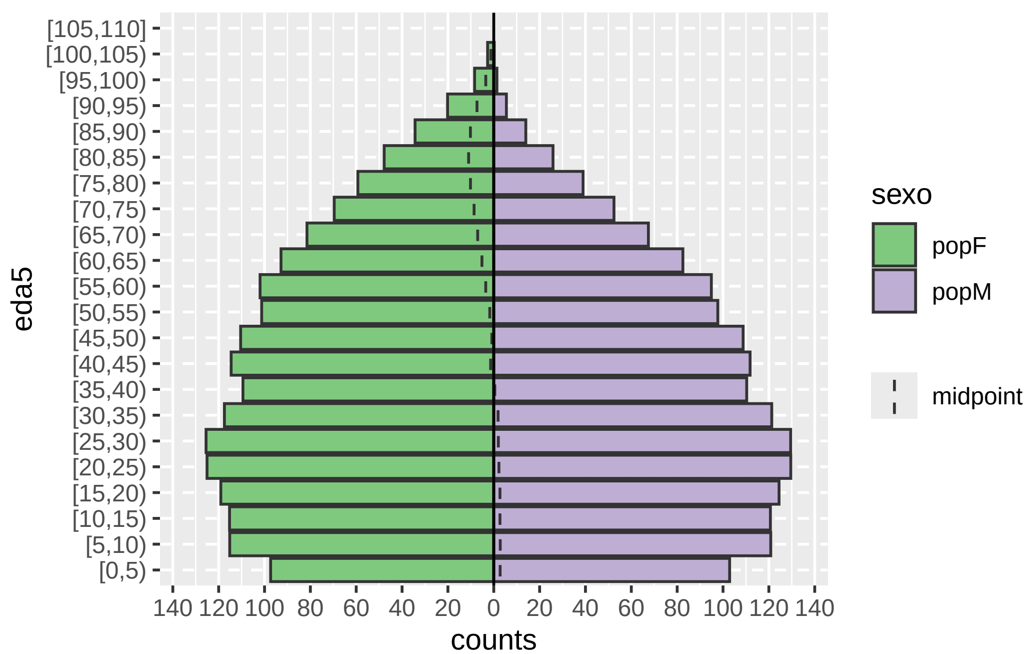
```
pob_uy_long %>%
  filter(year==2020) %>%
  count(eda5, sexo, wt=poblacion)
```

```
# A tibble: 42 x 3
   eda5    sexo      n
   <fct>   <chr> <dbl>
 1 [0,5)   popF   97.3
```

```
 2 [0,5)   popM  103.
 3 [5,10)  popF  115.
 4 [5,10)  popM  121.
 5 [10,15) popF  115.
 6 [10,15) popM  121.
 7 [15,20) popF  119.
 8 [15,20) popM  124.
 9 [20,25) popF  125.
10 [20,25) popM  130.
# i 32 more rows
```

```
pob_uy_long %>%
  filter(year==2020) %>%
  count(eda5, sexo, wt=poblacion) %>%
  apyramid::age_pyramid(age_group = eda5,
                        split_by = sexo,
                        count = n)
```

```
Warning: Removed 1 row containing missing values or values outside the scale
range.
```



Nos ahorra un par de pasos, pero siempre tenemos que solucionar algunos elementos

3.3.1.1 Momento de práctica

Haz una pirámide para otro país, para el año 2040 en escenario de alta fecundidad.

3.3.2 Opcional

Veamos como hacemos un *loop* para hacer varias pirámides, pero antes tenemos que arreglar un poco esa base que bajamos de WPP

```
#popAge5dt <- popAge5dt
popAge5dt  %<>% # checa este pipe
  mutate(edad=parse_number(age)) %>%
  mutate(edad_factor=as.factor(edad))



popAge5dt %<>%
  pivot_longer(cols=popM:pop,
               names_to = "sex",
               values_to = "poblacion") %>%
  mutate(sex=str_replace_all(sex,"popF", "Mujeres")) %>%
  mutate(sex=str_replace_all(sex,"popM", "Hombres")) %>%
  mutate(sex=str_replace_all(sex,"pop", "Total"))
```

El loop:

```
anios<-unique(popAge5dt$year)
pais<-c(858, 222, 320, 340, 484, 1830)
# uy: 858
# sv: 222
# gt: 320
# hn: 340
# mx: 484
# CA: 916
# LAC: 1830)

# Este es el loop donde reemplaza por i cada código de país
for (i in pais){

  popAge5dt %>%
    mutate(poblacion=poblacion/1000) %>%
```

```
    filter(country_code==i) %>%
    filter(!sex=="Total") %>%
    filter(year==2020) %>%
    age_pyramid(edad_factor, # edad
                split_by = sex,
                count=poblacion)+
    labs(x="edad",
         y="millones de personas",
         title = paste0(popAge5dt[popAge5dt$country_code==i,]$name),
         fill="Sexo")->g

   ggsave(plot=g,
          filename=paste0("pira",i,".png", sep=""),
          width=9,
          height=7)

  g
  assign(paste0("pira",i, sep=""), g)

}
```

3.3.3 Momento de práctica

Haz un loop para hacer las pirámides de las proyecciones de uruguay, una para cada año.

3.4 Diagrama de Lexis

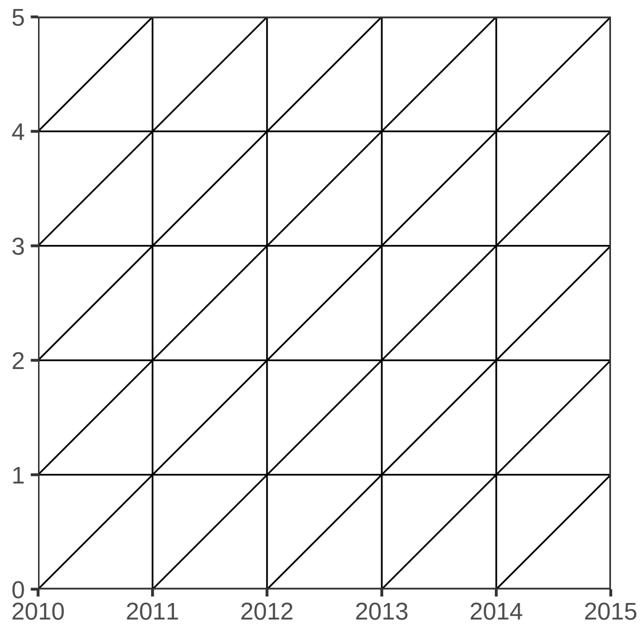
El paquete fue creado Philipp Ottolinger, este ejercicio es una versión en español (con algunos comentarios) de su ejemplo https://github.com/ottlngr/LexisPlotR

3.4.1 Dibujar una cuadrícula

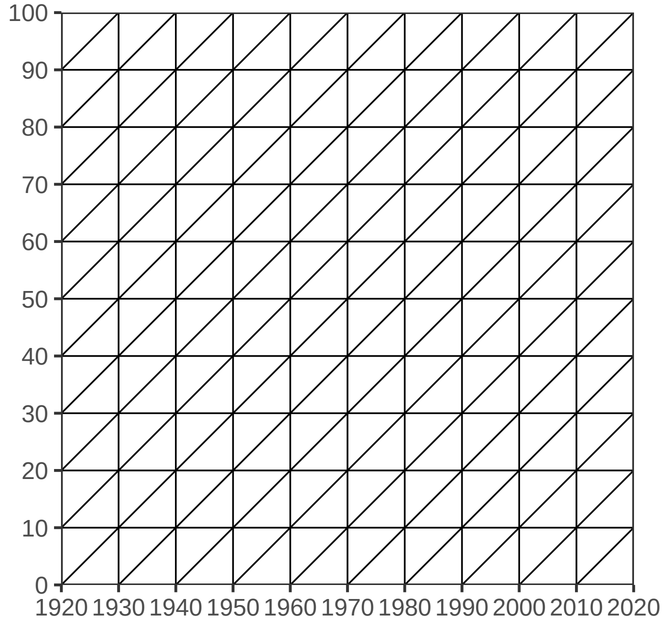
Este paquete nos puede ayudar a hacer nuestras cuadrículas. Ponemos los años de inicio y de final; así como las edades de inicio y de final. Recuerda que un diagrama de Lexis debe tener una misma escala en los ejes.

```
# Dibuje una cuadrícula de Lexis desde el año 2010 hasta el año 2015, que representa las e
LexisPlotR::lexis_grid(year_start = 2010,
                       year_end = 2015,
```

```
                                    age_start = 0,
                                    age_end=5)
```

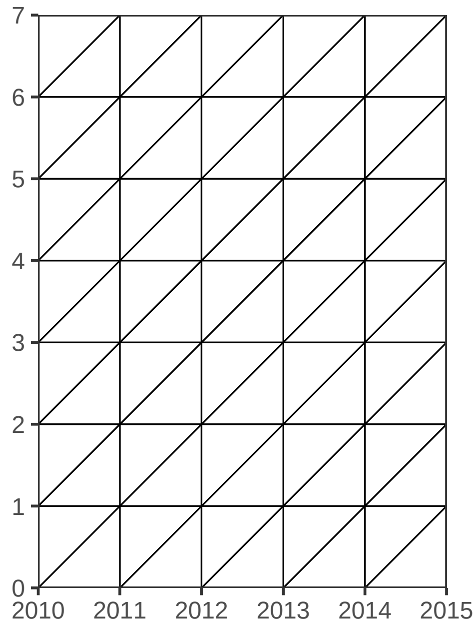



```
LexisPlotR::lexis_grid(year_start = 1920,
                       year_end = 2020,
                       age_start = 0,
                       age_end = 100,
                       delta = 10)
```



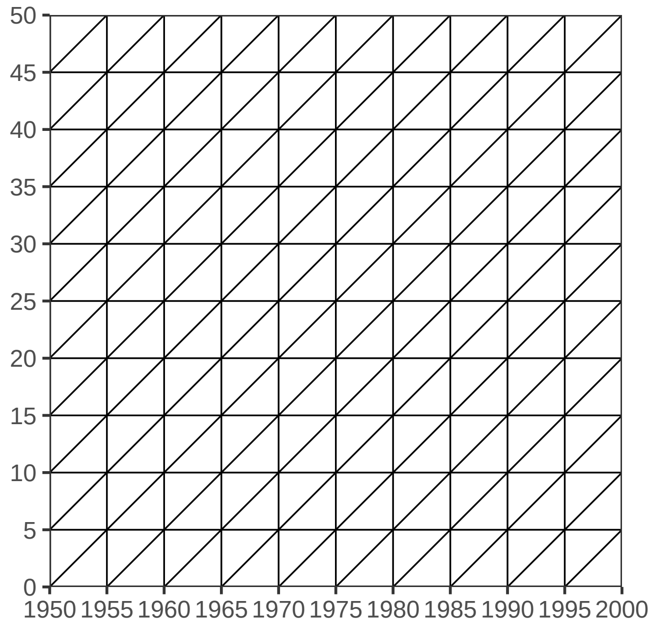
Aunque no necesariamente podemos dibujar sólo cuadrados

```
# Dibuje una cuadrícula de Lexis desde el año 2010 hasta el año 2015, que representa las e

lexis_grid(year_start = 2010, year_end = 2015, age_start = 0, age_end = 7)
```



Si no ponemos nada especifico en un argumento "d=", asume que los deltas son de un año. Pero lo podemos modificar

```
lexis_grid(year_start = 1950, year_end = 2000, age_start = 0, age_end = 50, delta = 5)
```

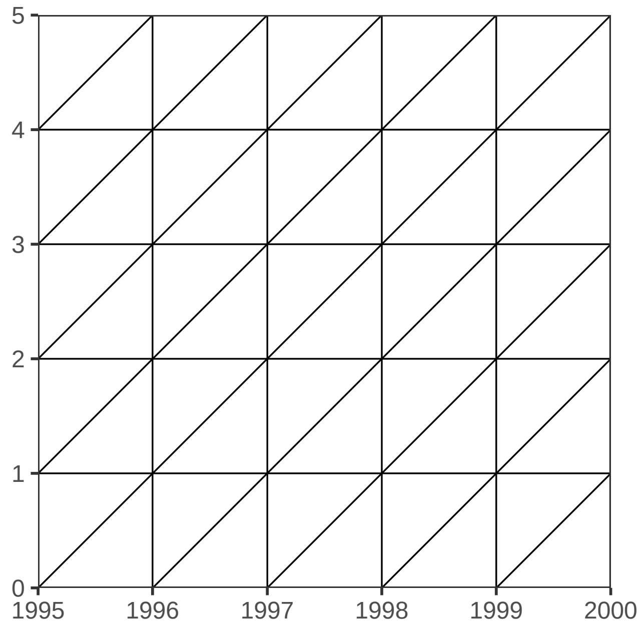


3.4.2 Sombreados en el diagrama

Lo más fácil es crear un objeto primero con nuestra cuadrícula sobre la cual graficaremos los elementos del Lexis

```
mi_diagrama <- lexis_grid(year_start = 1995, year_end = 2000, age_start = 0, age_end = 5)
```

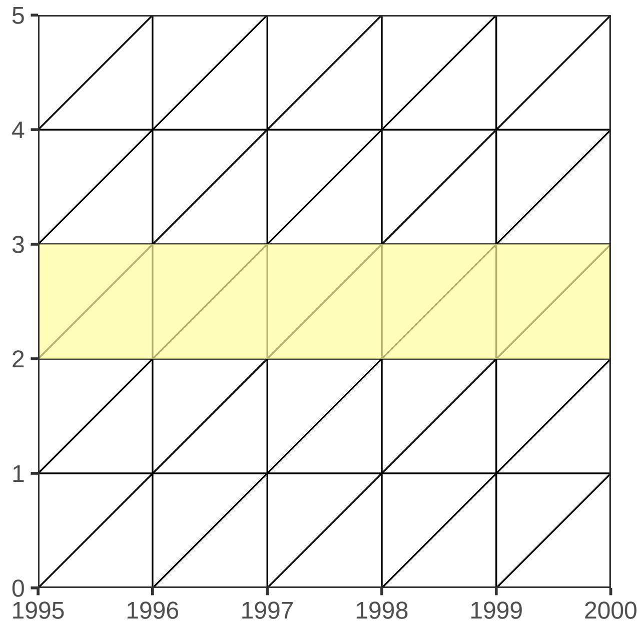
```
mi_diagrama
```



Para poder sombrear áreas con este paquete, debemos tener un diagrama ya guardado como objeto. Con distintas funciones vamos sombreando áreas.

3.4.2.1 Edad

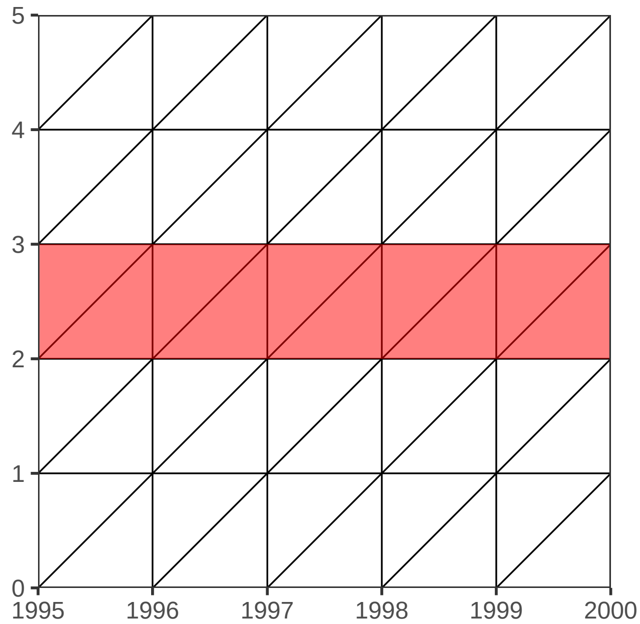
```
# Destacar todos los puntos que pertenecen a la edad de 2 años
mi_diagrama %>%
  lexis_age( age = 2)
```



¿Qué tipo de observación o estudio sería este?

Para cambiar el color:

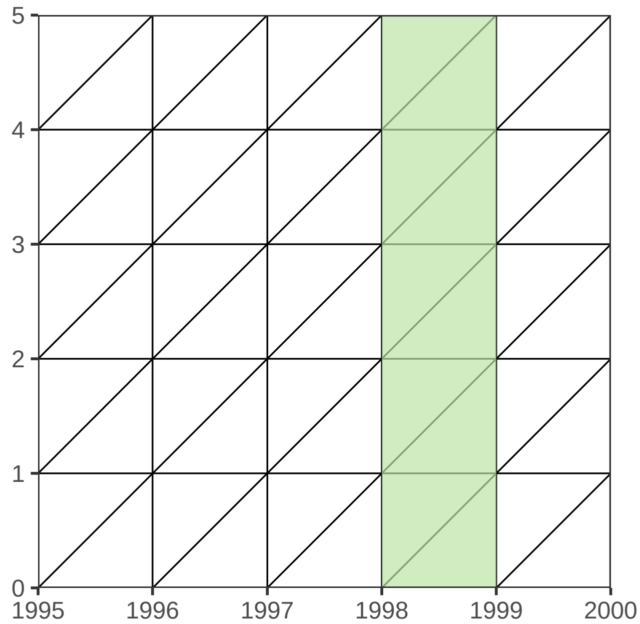
```
mi_diagrama %>%
  lexis_age(age = 2, fill = "red", alpha = 0.5)
```



3.4.2.2 Periodo

También podemos sombrear períodos

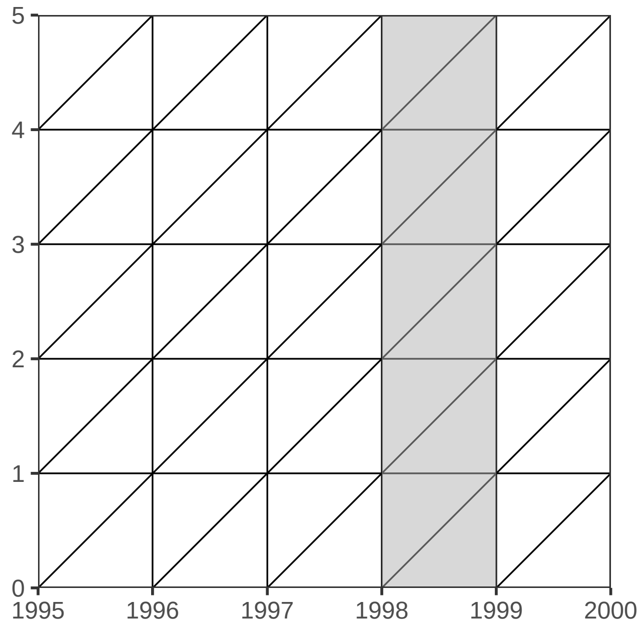
```
mi_diagrama %>%
  lexis_year(year=1998)
```



¿Qué tipo de observación o estudio sería este?

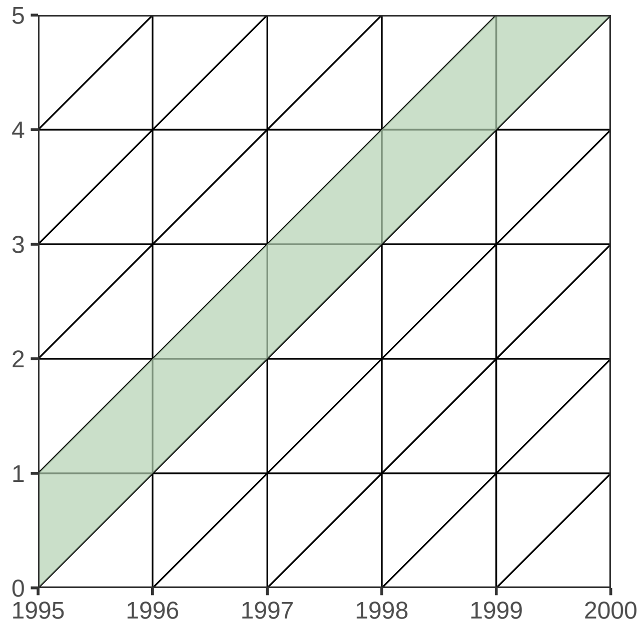
Para cambiar el color: Más info del color

```
mi_diagrama %>%
  lexis_year(year=1998, fill = "grey70", alpha = 0.5)
```



3.4.2.3 Cohorte

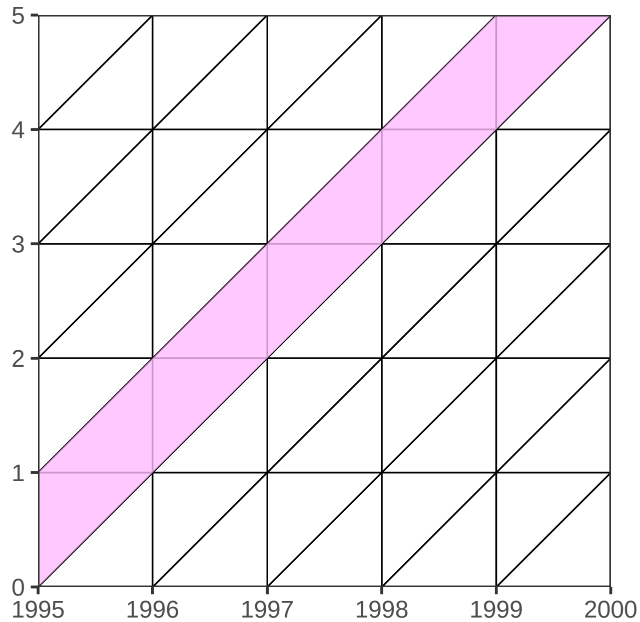
```
lexis_cohort(lg = mi_diagrama, cohort=1994)
```



¿Qué tipo de observación o estudio sería este?

También podemos cambiar el color y la transparencia:

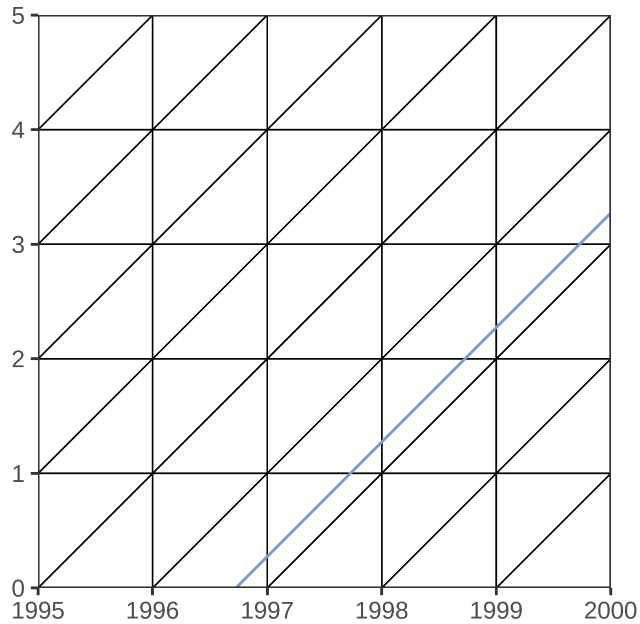
```
lexis_cohort(lg = mi_diagrama, cohort=1994, fill="plum1", alpha=0.8)
```



3.4.2.4 Líneas de vida

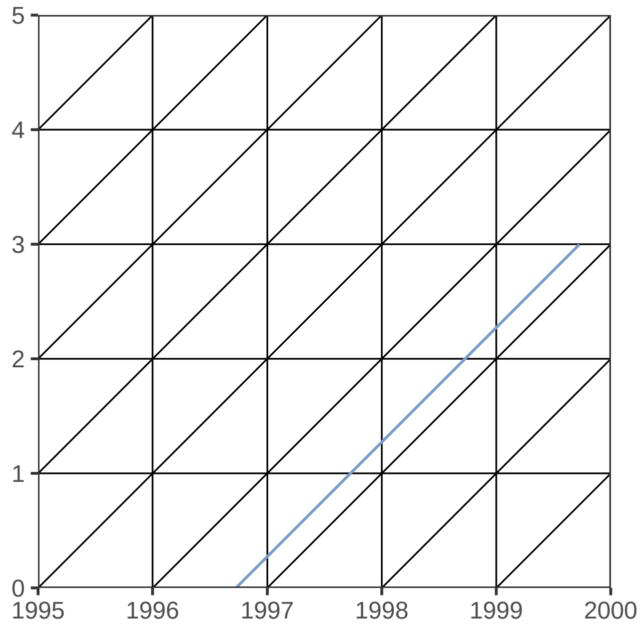
Alguien entra

```
lexis_lifeline(lg = mi_diagrama, birth = "1996-09-23")
```



Alguien entra y sale

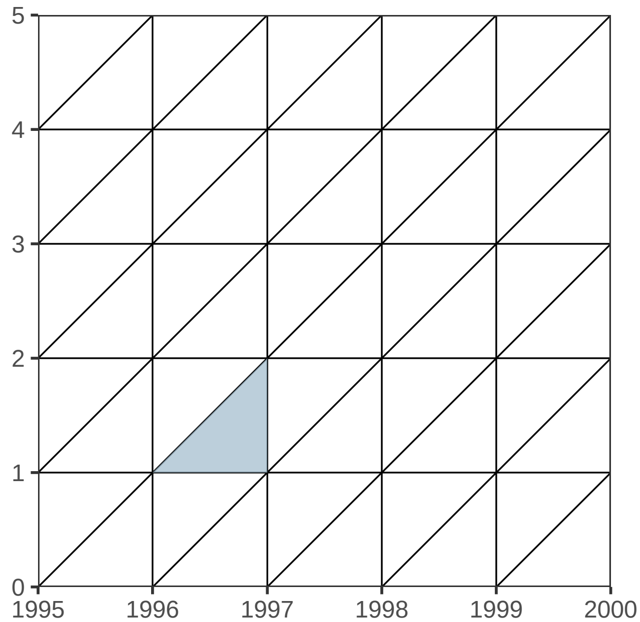
```
lexis_lifeline(lg = mi_diagrama, birth =  "1996-09-23", exit="1999-09-23")
```



3.4.2.5 Polígonos

No es tan sencillo, pero podemos dibujar un espacio “APC”, o varios.

```
polygons <- data.frame(group = c(1,
                                 1,
                                 1), # es un triángulo
                       x = c("1996-01-01",
                             "1997-01-01",
                             "1997-01-01"), # van en fechas
                       y = c(1,
                             1,
                             2)) # van en edades

mi_diagrama %>%
  lexis_polygon(x = polygons$x,
                y = polygons$y,
                group = polygons$group)
```



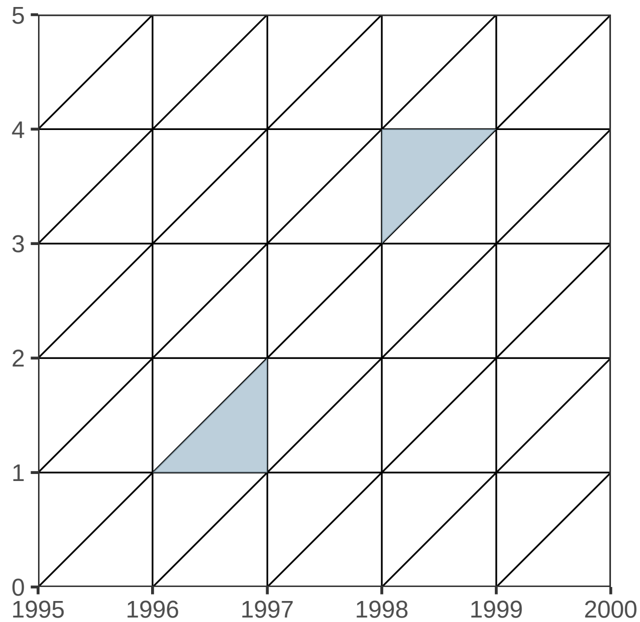
checa que básicamente se trata de colocar los puntos que dibujan el polígono. Son tres puntos:

1. Fecha: "1996-01-01", edad=1
2. Fecha: "1997-01-01", edad=1
3. Fecha: "1997-01-01", edad=2

Si queremos más triángulos, podemos agregarlos en el mismo objeto:

```
polygons <- data.frame(group = c(1,
                                 1,
                                 1, # es un triángulo
                                 2,
                                 2,
                                 2), # es otro triángulo
                       x = c("1996-01-01",
                             "1997-01-01",
                             "1997-01-01", # van en fechas 1
                             "1998-01-01",
                             "1998-01-01",
                             "1999-01-01"), # van en fechas 2
```

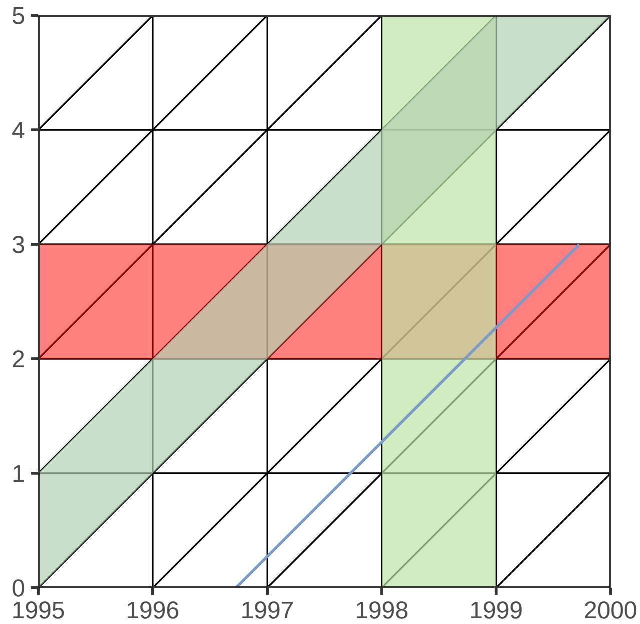
```
                        y = c(1,
                              1,
                              2, # van en edades 1
                              3,
                              4,
                              4))# van en edades 2

mi_diagrama %>%
  lexis_polygon( x = polygons$x,
                 y = polygons$y,
                 group = polygons$group)
```



3.4.2.6 Todo en uno

Podemos ir reescribiendo nuestro objeto

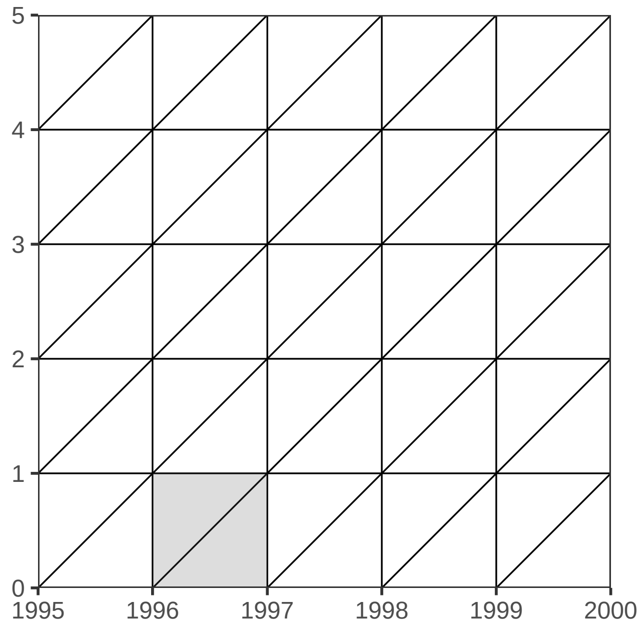
```
lexis_grid(year_start = 1995, year_end = 2000, age_start = 0, age_end = 5) %>%
  lexis_age(age = 2, fill = "red", alpha = 0.5) %>%
  lexis_year(year = 1998) %>%
  lexis_cohort(cohort=1994) %>%
  lexis_lifeline(birth = "1996-09-23", exit="1999-09-23")
```



3.4.2.7 Anotación manual

Para hacer cuadrados

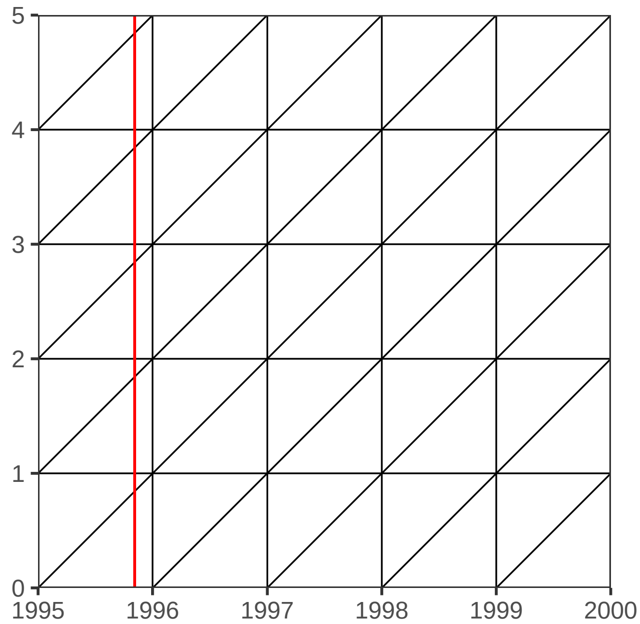
```
mi_diagrama  + # ojo este siguiente comando es ggplot
   annotate("rect",
            xmin = as.Date("1996-01-01"),
            xmax = as.Date("1997-01-01"),
            ymin = 0, ymax = 1, alpha = .2)
```



¿Qué tipo de observación es esta?

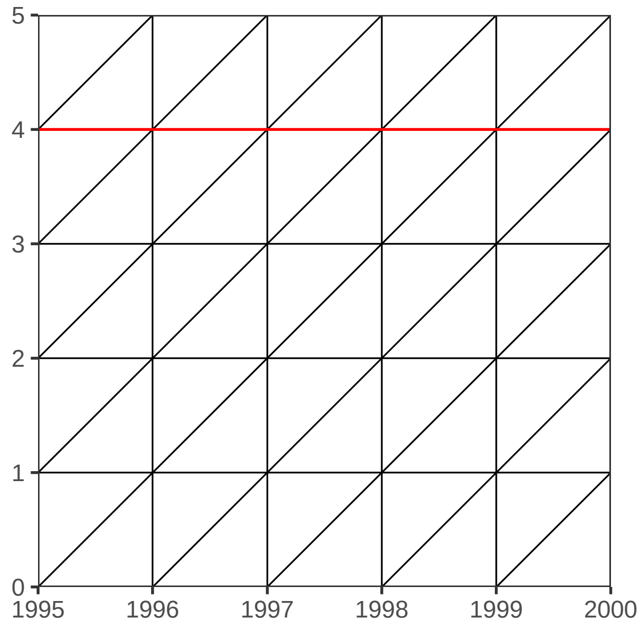
Si quisiéramos gráficar la fecha de un censo:

```
mi_diagrama  +
   geom_vline(xintercept = as.Date("1995-11-05"), colour = "red")
```



Si queremos poner una edad exacta:

```
mi_diagrama  +
   geom_hline(yintercept = 4, colour = "red")
```



3.4.3 Momento de práctica

Hacé un diagrama de Lexis decenal desde tu quinquenio de nacimiento hasta 2025, gráfica tu línea de vida y otros elementos o etapas importantes de tu vida

4 Tasas

4.1 Paquetes

```
if (!require("pacman")) install.packages("pacman") # instala pacman si se requiere
```

```
Cargando paquete requerido: pacman
```

```
pacman::p_load(tidyverse,
               readxl,
               writexl,
               janitor,
               remotes,
               wppExplorer,
               magrittr,
               collapse,
               fmsb,
               DemoTools,
               wpp2022)
```

4.2 Datos

```
censos <- readxl::read_excel("datos/censos.xlsx",
                     col_types = c("numeric","date", "numeric"))

sv1992<-readxl::read_excel("datos/censos_p2.xlsx", sheet = "El Salvador 1992") %>%
  janitor::clean_names() %>% #
  dplyr::mutate(total=male + female) %>% # ojo
  dplyr::mutate(age=as.numeric(age)) #ojo

# de wpp2022
```

```
data("tfr1dt")
data("percentASFR1dt")
data("misc1dt")
data("mx1dt")
data("pop1dt")
```

4.3 Tasas de crecimiento

4.3.1 Fechas censales y tasas de crecimiento

Un elemento fundamental es encontrar los periodos intercensales en años. Los censos tienen diferentes fechas.

```
censos %<>%
    dplyr::mutate(dias = c(NA, diff(fecha))) %>%
    dplyr::mutate(n=dias/365) %>%
  clean_names()

censos
```

```
# A tibble: 6 x 5
    ano fecha               poblacion  dias     n
  <dbl> <dttm>                  <dbl> <dbl> <dbl>
1  1963 1963-10-16 00:00:00   2561710    NA NA
2  1975 1975-10-23 00:00:00   2799940  4390 12.0
3  1985 1985-10-21 00:00:00   2959150  3651 10.0
4  1996 1996-05-22 00:00:00   3159200  3866 10.6
5  2006 2006-07-01 00:00:00   3065604  3692 10.1
6  2011 2011-09-01 00:00:00   3284250  1888  5.17
```

Con esta base ya podemos ir calculando diferentes tipos de crecimiento básicos.

4.3.2 Ritmo

$$ritmo = \frac{P_{t+n}}{P_t}$$

```r
censos<-censos %>%
   mutate(ritmo = poblacion/lag(poblacion))
censos
```

```
# A tibble: 6 x 6
    ano fecha               poblacion  dias     n  ritmo
  <dbl> <dttm>                  <dbl> <dbl> <dbl>  <dbl>
1  1963 1963-10-16 00:00:00   2561710    NA NA    NA
2  1975 1975-10-23 00:00:00   2799940  4390 12.0   1.09
3  1985 1985-10-21 00:00:00   2959150  3651 10.0   1.06
4  1996 1996-05-22 00:00:00   3159200  3866 10.6   1.07
5  2006 2006-07-01 00:00:00   3065604  3692 10.1   0.970
6  2011 2011-09-01 00:00:00   3284250  1888  5.17  1.07
```

4.3.3 Crecimiento

$$c = \frac{P_{t+n} - P_t}{P_t} = \frac{P_{t+n}}{P_t} - 1$$

Básicamente es el ritmo menos 1

```r
censos<-censos %>%
   mutate(c = ritmo-1)
censos
```

```
# A tibble: 6 x 7
    ano fecha               poblacion  dias     n  ritmo       c
  <dbl> <dttm>                  <dbl> <dbl> <dbl>  <dbl>   <dbl>
1  1963 1963-10-16 00:00:00   2561710    NA NA    NA     NA
2  1975 1975-10-23 00:00:00   2799940  4390 12.0   1.09   0.0930
3  1985 1985-10-21 00:00:00   2959150  3651 10.0   1.06   0.0569
4  1996 1996-05-22 00:00:00   3159200  3866 10.6   1.07   0.0676
5  2006 2006-07-01 00:00:00   3065604  3692 10.1   0.970 -0.0296
6  2011 2011-09-01 00:00:00   3284250  1888  5.17  1.07   0.0713
```

4.3.4 Crecimiento aritmético

$$r_a = \frac{P_{t+n} - P_t}{n * P_t} = \frac{c}{n}$$

Básicamente es el crecimiento entre el periodo intercensal.

```
censos<-censos %>%
   mutate(ra = c/n)
censos
```

```
# A tibble: 6 x 8
    ano fecha               poblacion  dias     n  ritmo       c        ra
  <dbl> <dttm>                  <dbl> <dbl> <dbl>  <dbl>   <dbl>     <dbl>
1  1963 1963-10-16 00:00:00   2561710    NA NA    NA     NA      NA
2  1975 1975-10-23 00:00:00   2799940  4390 12.0   1.09   0.0930  0.00773
3  1985 1985-10-21 00:00:00   2959150  3651 10.0   1.06   0.0569  0.00568
4  1996 1996-05-22 00:00:00   3159200  3866 10.6   1.07   0.0676  0.00638
5  2006 2006-07-01 00:00:00   3065604  3692 10.1   0.970 -0.0296 -0.00293
6  2011 2011-09-01 00:00:00   3284250  1888  5.17  1.07   0.0713  0.0138
```

4.3.5 Crecimiento geométrico

$$r_g = \sqrt[n]{\frac{P_{t+n}}{P_t}} - 1$$

Es la raíz n-ésima del ritmo menos 1

```
censos<-censos %>%
   mutate(rg = ritmo^(1/n)-1)
censos
```

```
# A tibble: 6 x 9
    ano fecha               poblacion  dias     n  ritmo       c        ra
  <dbl> <dttm>                  <dbl> <dbl> <dbl>  <dbl>   <dbl>     <dbl>
1  1963 1963-10-16 00:00:00   2561710    NA NA    NA     NA      NA
2  1975 1975-10-23 00:00:00   2799940  4390 12.0   1.09   0.0930  0.00773
3  1985 1985-10-21 00:00:00   2959150  3651 10.0   1.06   0.0569  0.00568
4  1996 1996-05-22 00:00:00   3159200  3866 10.6   1.07   0.0676  0.00638
5  2006 2006-07-01 00:00:00   3065604  3692 10.1   0.970 -0.0296 -0.00293
6  2011 2011-09-01 00:00:00   3284250  1888  5.17  1.07   0.0713  0.0138
# i 1 more variable: rg <dbl>
```

4.3.6 Crecimiento exponencial

$$r = \frac{ln \frac{P_{t+n}}{P_t}}{n}$$

Básicamente es logaritmo del ritmo entre n

```
censos<-censos %>%
   mutate(r = log(ritmo)/n)
censos
```

```
# A tibble: 6 x 10
    ano fecha               poblacion  dias     n  ritmo       c        ra
  <dbl> <dttm>                  <dbl> <dbl> <dbl>  <dbl>   <dbl>    <dbl>
1  1963 1963-10-16 00:00:00   2561710    NA NA    NA     NA       NA
2  1975 1975-10-23 00:00:00   2799940  4390 12.0   1.09   0.0930  0.00773
3  1985 1985-10-21 00:00:00   2959150  3651 10.0   1.06   0.0569  0.00568
4  1996 1996-05-22 00:00:00   3159200  3866 10.6   1.07   0.0676  0.00638
5  2006 2006-07-01 00:00:00   3065604  3692 10.1   0.970 -0.0296 -0.00293
6  2011 2011-09-01 00:00:00   3284250  1888  5.17  1.07   0.0713  0.0138
# i 2 more variables: rg <dbl>, r <dbl>
```

Este crecimiento es el más utilizado.

Podemos graficar los diferentes crecimientos, será más fácil si cambiamos el formato

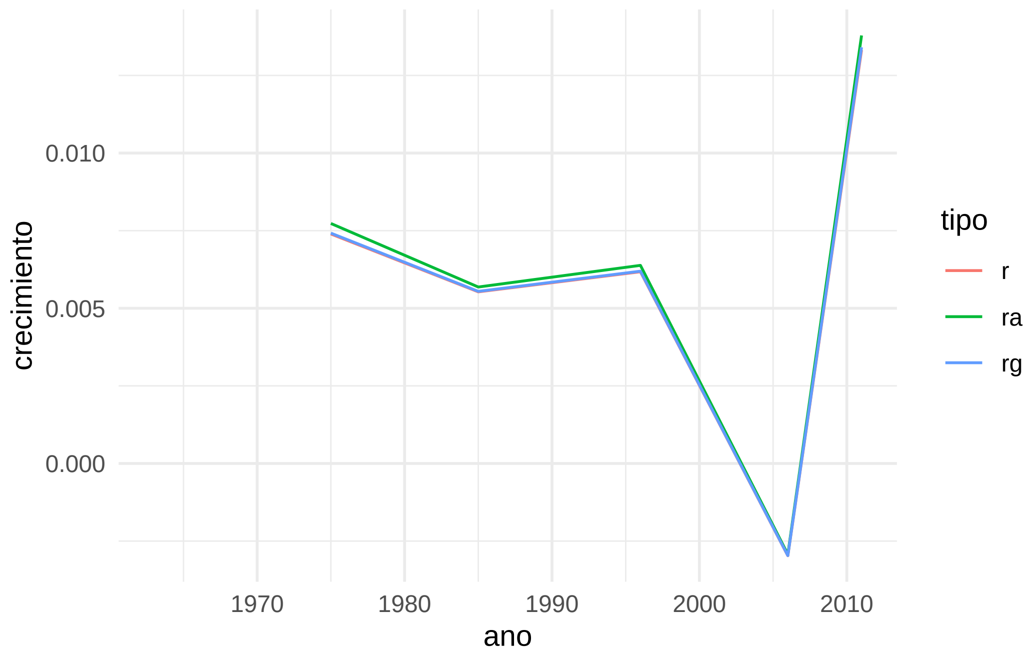
```
censos %>%
  select(c(ano, ra, rg, r)) %>%
  pivot_longer(-ano, names_to = "tipo", values_to = "crecimiento")
```

```
# A tibble: 18 x 3
     ano tipo  crecimiento
   <dbl> <chr>       <dbl>
 1  1963 ra       NA
 2  1963 rg       NA
 3  1963 r        NA
 4  1975 ra        0.00773
 5  1975 rg        0.00742
 6  1975 r         0.00739
 7  1985 ra        0.00568
 8  1985 rg        0.00554
 9  1985 r         0.00553
10  1996 ra        0.00638
11  1996 rg        0.00620
12  1996 r         0.00618
13  2006 ra       -0.00293
```

```
14  2006 rg       -0.00297
15  2006 r        -0.00297
16  2011 ra        0.0138
17  2011 rg        0.0134
18  2011 r         0.0133
```

```
censos %>%
  select(c(ano, ra, rg, r)) %>%
  pivot_longer(-ano, names_to = "tipo", values_to = "crecimiento") %>%
  ggplot(aes(ano,crecimiento, group=tipo, color=tipo)) +
  geom_line() + theme_minimal()
```

```
Warning: Removed 3 rows containing missing values or values outside the scale range
(`geom_line()`).
```



Con estas tasas de crecimiento también podemos hacer estimaciones de tiempo y de poblaciones en fechas específicas.

4.3.7 Proyeccion

$$P_{t+n} = P_t * e^{nr}$$

Vamos a proyectar la población al primero de julio de 2009

```
n<-difftime(as.Date("2009-07-1"),
        as.Date(paste(censos[censos$ano==2006,]$fecha)))
n
```

```
Time difference of 1096 days
```

```
n<-as.numeric(n/365)

n
```

```
[1] 3.00274
```

```
censos[censos$ano==2006, "poblacion"]
```

```
# A tibble: 1 x 1
  poblacion
      <dbl>
1   3065604
```

```
censos[censos$ano==2006, "r"]
```

```
# A tibble: 1 x 1
        r
     <dbl>
1 -0.00297
```

```
ptn<- censos[censos$ano==2006,"poblacion"]*exp(n*censos[censos$ano==2006, "r"])

paste(ptn) # para ver los decimales
```

```
[1] "3038356.78921973"
```

Con esto podemos crear una función: [debes tener los datos de censo con las estimaciones de r]

```r
pob_estim <- function(fecha,ano) {

    n<-difftime(as.Date(fecha),
            as.Date(paste(censos[censos$ano==ano,]$fecha)))
    n<-as.numeric(n/365)
    ptn<-censos[censos$ano==ano,"poblacion"]*exp(n*censos[censos$ano==ano, "r"])

  return(ptn)

}
```

```r
pob_estim(fecha ="2008-01-07",
          ano = 2006)
```

```
  poblacion
1   3051776
```

4.3.8 Tiempo

$$n = \frac{ln\frac{P_{t+n}}{P_t}}{r}$$

¿Cuánto tiempo tardaría en duplicarse la población del último censo?

```r
n_calc<-log(2*censos[censos$ano==2011,"poblacion"]/censos[censos$ano==2011,"poblacion"])/c
n_calc
```

```
  poblacion
1  52.04213
```

4.3.8.1 Momento de práctica

- Calcula, si la población creciera cómo lo hizo entre los primeros dos censos, en cuánto tiempo se duplicaría
- Calcula la población al 1 de julio de 1999, puedes usar la función.

4.4 Reconstrucción de las tasas de fecundidad

"tfr1dt" # esta es la base de las tasas de fecundidad "percentASFR1dt" ¿cómo se distribuye a lo largo de las edades de las mujeres?

La lógica la muestro con un ejemplo

```
tfr1dt %>%
  filter(name=="World") %>%
  filter(year==2000) %>%
  select(tfr) -> tfr # esto es un escalar
```

hoy queremos el vector de las edades y del porcentaje

```
percentASFR1dt %>%
  filter(name=="World") %>%
  filter(year==2000) %>%
  select(age, pasfr) -> pasfr # esto es un vector ordenado por las edades
```

Sumamos el vector y nos damos cuenta que no es la *age specific fertility rate*

```
sum(pasfr$pasfr)# suma el 100%
```

```
[1] 100
```

Vamos a prorratear la intensidad sobre el calendario :)

```
pasfr %<>%
  mutate(tasa=pasfr*tfr$tfr/100*1000)
```

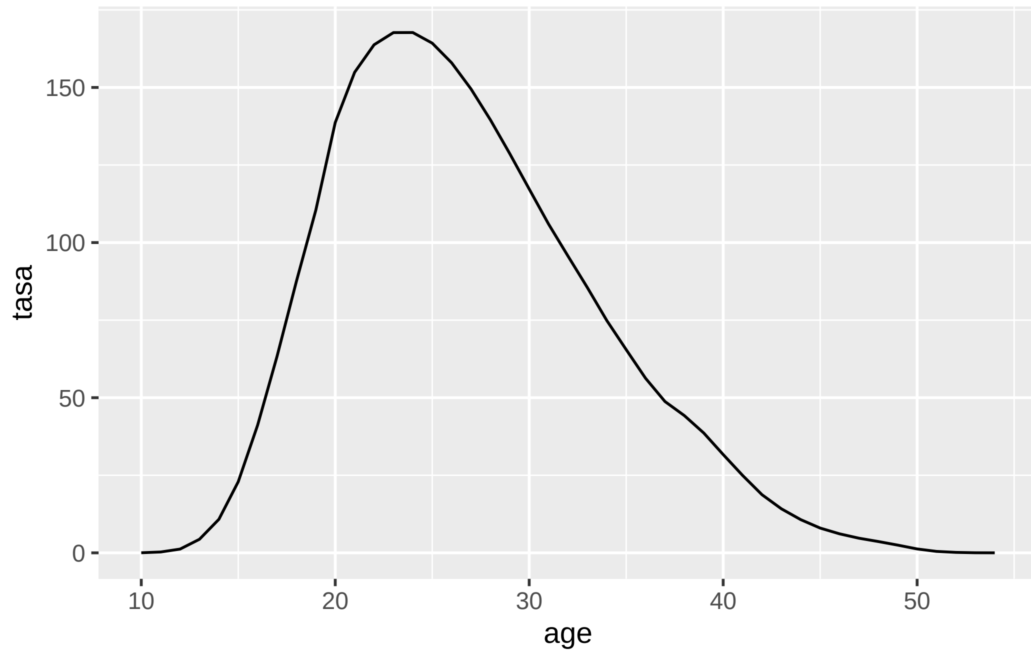
```
sum(pasfr$tasa/1000) # suma la tasa global
```

```
[1] 2.734497
```

Vamos a graficarla

```
pasfr %>%
  ggplot() +
  aes(x=age,
      y=tasa) +
```

```
  geom_line()
```



4.4.1 Fusionando

Podemos fusionar ambas tablas para hacer esto para todos.

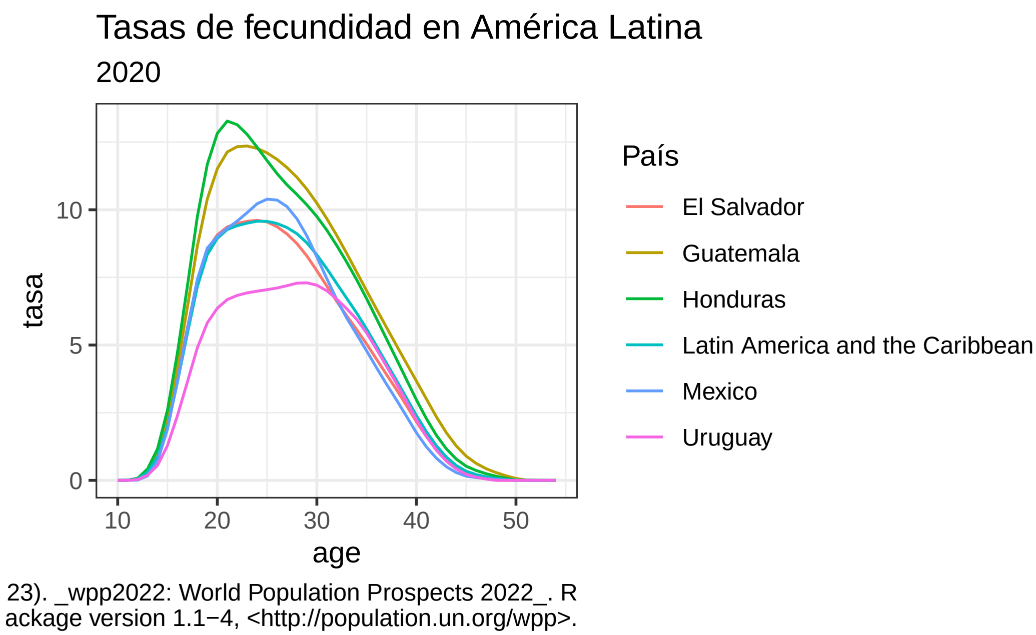
```
tasa_fec1dt<-percentASFR1dt %>%
  dplyr::left_join(tfr1dt) %>%
  mutate(tasa = pasfr * tfr)
```

```
Joining with `by = join_by(country_code, name, year)`
```

Hoy podemos comparar intensidades y calendarios históricos!

```
tasa_fec1dt %>%
  filter(country_code %in% c(858, 222, 320, 340, 484, 1830)) %>%
  filter(year==2021) %>%
  ggplot() +
  aes(x = age,
      y = tasa,
```

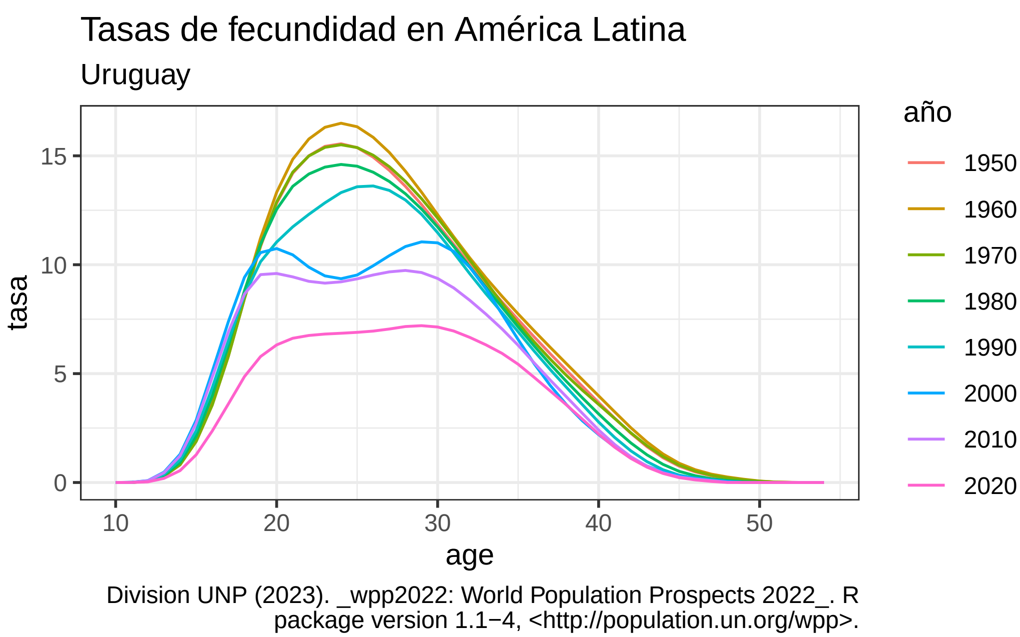
```
         color = name,
         group = name) +
  geom_line() +
  theme_bw() +
  labs(title = "Tasas de fecundidad en América Latina",
       subtitle = "2020",
       color = "País",
       caption = "Division UNP (2023). _wpp2022: World Population Prospects 2022_. R
  package version 1.1-4, <http://population.un.org/wpp>.
")
```



Hoy comparamos tasas a lo largo del tiempo

```
tasa_fec1dt %>%
  filter(country_code==858) %>%
  filter(year%in%seq(1950,2020, by=10)) %>%
  ggplot() +
  aes(x = age,
      y = tasa,
      color = as.factor(year),
      group = as.factor(year)) +
```

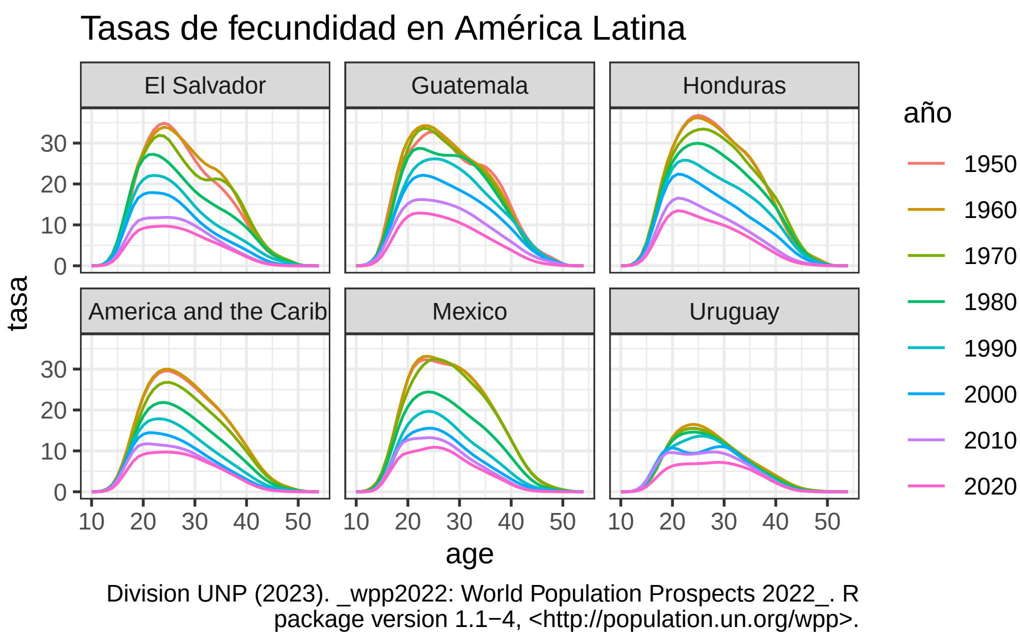
```
  geom_line() +
  theme_bw() +
  labs(title = "Tasas de fecundidad en América Latina",
       subtitle = "Uruguay",
       color = "año",
       caption = "Division UNP (2023). _wpp2022: World Population Prospects 2022_. R
  package version 1.1-4, <http://population.un.org/wpp>.
")
```



Un mix:

```
tasa_fec1dt %>%
  filter(country_code %in% c(858, 222, 320, 340, 484, 1830)) %>%
  filter(year%in%seq(1950,2020, by=10)) %>%
  ggplot() +
  aes(x = age,
      y = tasa,
      color = as.factor(year),
      group = as.factor(year)) +
  geom_line() +
  theme_bw() +
```

```r
  labs(title = "Tasas de fecundidad en América Latina",
       color = "año",
       caption = "Division UNP (2023). _wpp2022: World Population Prospects 2022_. R
  package version 1.1-4, <http://population.un.org/wpp>.
") +
  facet_wrap(~name)
```



4.5 Nacimientos

Es muy útil tener las tasa en edades específicas. Pero si quisiéramos las tasas en edades quinquenales o bien calcular la tasa general de fecundidad, tendríamos que tener la estructura de los nacimientos.

```r
skimr::skim(misc1dt)
```

Table 4.1: Data summary

Name	misc1dt
Number of rows	20520
Number of columns	8

Key	NULL
Column type frequency:	
character	1
numeric	7
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
name	0	1	4	59	0	283	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
country_code	0	1	598.28	566.41	4.00	266.00	531.00	792.00	5501.00	
year	0	1	1985.50	20.78	1950.00	1967.75	1985.50	2003.25	2021.00	
births	0	1	4182.98	15739.92	0.02	11.42	129.38	752.17	144194.06	
cbr	0	1	29.06	13.06	5.06	17.23	28.25	41.07	59.42	
cdr	0	1	11.24	6.00	0.80	7.11	9.60	13.59	103.53	
deaths	0	1	1630.65	5910.08	0.01	3.43	55.89	278.38	69248.15	
growthrate	0	1	1.76	1.82	-71.69	0.79	1.76	2.61	36.30	

La lógica es fusionar a nuestro archivos de tasas, las poblaciones medias y los nacimientos totales en el año, para reconstruir los numeradores.

```
tasa_fec1dt %<>%
  left_join(popAge1dt) %>%
  left_join(misc1dt)
```

```
Joining with `by = join_by(country_code, name, year, age)`
Joining with `by = join_by(country_code, name, year)`
```

```
dplyr::glimpse(tasa_fec1dt)
```

```
Rows: 1,936,575
Columns: 15
$ country_code <int> 900, 900, 900, 900, 900, 900, 900, 900, 900, 900, 900, 90~
$ name         <chr> "World", "World", "World", "World", "World", "World", "Wo~
$ year         <int> 1950, 1950, 1950, 1950, 1950, 1950, 1950, 1950, 1950, 195~
$ age          <int> 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2~
$ pasfr        <dbl> 0.000773, 0.009857, 0.037902, 0.118608, 0.305228, 0.63398~
$ tfr          <dbl> 4.859551, 4.859551, 4.859551, 4.859551, 4.859551, 4.85955~
$ tasa         <dbl> 0.003756433, 0.047900594, 0.184186702, 0.576381625, 1.483~
$ popM         <dbl> 26811.66, 26357.60, 26832.81, 26554.83, 26048.04, 25301.0~
$ popF         <dbl> 25528.05, 25111.84, 25478.13, 25273.50, 24924.47, 24432.2~
$ pop          <dbl> 52339.71, 51469.44, 52310.93, 51828.33, 50972.51, 49733.2~
$ births       <dbl> 92083.26, 92083.26, 92083.26, 92083.26, 92083.26, 92083.2~
$ cbr          <dbl> 36.837, 36.837, 36.837, 36.837, 36.837, 36.837, 36.837, 3~
$ cdr          <dbl> 19.518, 19.518, 19.518, 19.518, 19.518, 19.518, 19.518, 1~
$ deaths       <dbl> 48788.54, 48788.54, 48788.54, 48788.54, 48788.54, 48788.5~
$ growthrate   <dbl> 1.732, 1.732, 1.732, 1.732, 1.732, 1.732, 1.732, 1.732, 1~
```

Vamos a reconstruir los numeradores de nacimientos específicos

```r
tasa_fec1dt %<>%
  mutate(birth_age=tfr*popF)
```

Con esta base ya podemos hacer cálculos quinquenales y demás…

4.6 Mortalidad

Podemos hacer algo parecido con la mortalidad, sólo que acá las tasas de mortalidad ya están calculadas de manera específica y se incluyen los datos proyectados.

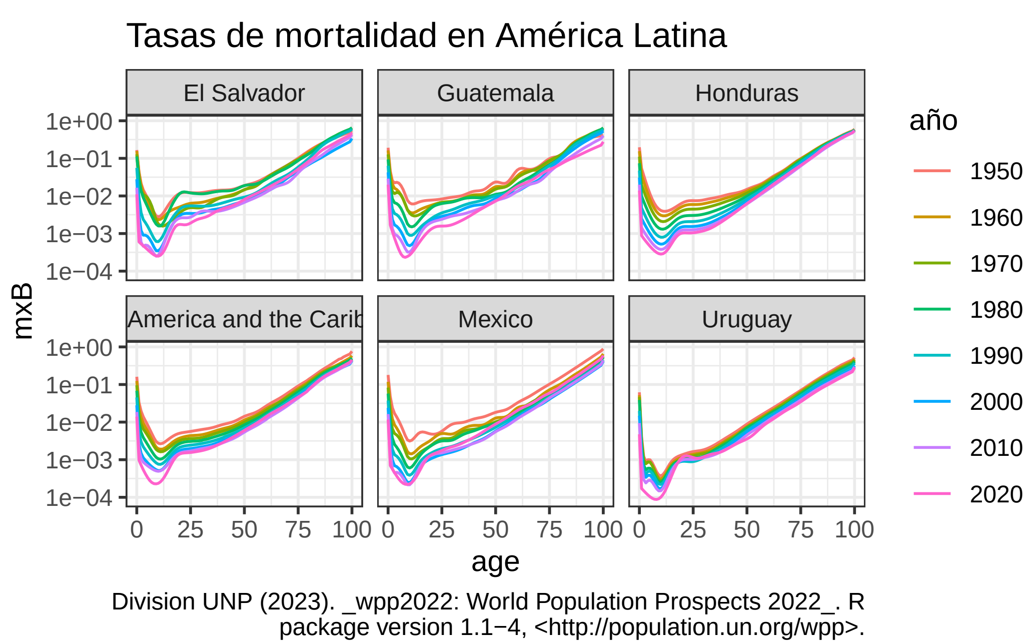
```r
mx1dt %>%
  glimpse()
```

```
Rows: 4,346,535
Columns: 7
$ country_code <int> 900, 900, 900, 900, 900, 900, 900, 900, 900, 900, 900, 90~
$ name         <chr> "World", "World", "World", "World", "World", "World", "Wo~
$ year         <int> 1950, 1950, 1950, 1950, 1950, 1950, 1950, 1950, 1950, 195~
$ age          <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
$ mxM          <dbl> 0.16379163, 0.04497134, 0.02621819, 0.01725326, 0.0122286~
```

```
$ mxF          <dbl> 0.15478432, 0.04220881, 0.02517510, 0.01706646, 0.0124719~
$ mxB          <dbl> 0.15939104, 0.04362024, 0.02570769, 0.01716187, 0.0123477~
```

Grafiquemos:

```
mx1dt %>%
  filter(country_code %in% c(858, 222, 320, 340, 484, 1830)) %>%
  filter(year%in%seq(1950,2020, by=10)) %>%
  ggplot() +
  aes(x = age,
      y = mxB,
      color = as.factor(year),
      group = as.factor(year)) +
  geom_line() +
  scale_y_continuous(trans = "log10") + # ojo
  theme_bw() +
  labs(title = "Tasas de mortalidad en América Latina",
       color = "año",
       caption = "Division UNP (2023). _wpp2022: World Population Prospects 2022_. R
  package version 1.1-4, <http://population.un.org/wpp>.
") +
  facet_wrap(~name)
```



Si queremos reconstruir muertes por años podemos pegar la información de misc1dt y de pop1dt

```
mx1dt %<>%
  left_join(misc1dt) %>%
  left_join(popAge1dt)
```

```
Joining with `by = join_by(country_code, name, year)`
Joining with `by = join_by(country_code, name, year, age)`
```

Vamos a calcular muertes por edad

```
mx1dt %<>%
  mutate(deaths_age=mxB*pop)

head(mx1dt)
```

```
   country_code   name  year   age        mxM        mxF        mxB   births
          <int> <char> <int> <int>      <num>      <num>      <num>    <num>
1:          900  World  1950     0 0.16379163 0.15478432 0.15939104 92083.26
2:          900  World  1950     1 0.04497134 0.04220881 0.04362024 92083.26
3:          900  World  1950     2 0.02621819 0.02517510 0.02570769 92083.26
4:          900  World  1950     3 0.01725326 0.01706646 0.01716187 92083.26
5:          900  World  1950     4 0.01222864 0.01247194 0.01234776 92083.26
6:          900  World  1950     5 0.00886063 0.00927862 0.00906522 92083.26
      cbr    cdr   deaths growthrate     popM     popF      pop deaths_age
    <num>  <num>    <num>      <num>    <num>    <num>    <num>      <num>
1: 36.837 19.518 48788.54      1.732 42252.26 40419.48 82671.74 13177.1346
2: 36.837 19.518 48788.54      1.732 38506.42 36803.01 75309.44  3285.0157
3: 36.837 19.518 48788.54      1.732 34593.81 33244.05 67837.86  1743.9547
4: 36.837 19.518 48788.54      1.732 32842.45 31439.53 64281.98  1103.1990
5: 36.837 19.518 48788.54      1.732 30648.62 29346.23 59994.85   740.8020
6: 36.837 19.518 48788.54      1.732 28410.58 27284.76 55695.34   504.8905
```

#Estandarización de tasas

Vamos a volver a nuestros grupos quinquenales, dejando la mortalidad infantil aparte:

```
est<-mx1dt %>%
  select(country_code:age, pop, deaths_age) %>%
  filter(name=="Uruguay") %>%
```

```r
  filter(year%in%c(1950,2020)) %>%
  mutate(eda5=cut(age, # la variable a cortar
                  breaks=c(0,1, # para dejar la mortalidad infantil
                           seq(5,110, # El rango válido
                             by=5)), # El ancho del intervalo
                  include.lowest=T, # para que incluya el valor más bajo dentro del interv
                  right=F)) # indica si el intervalo irá abierto en la derecha, ponemos un
```

Vamos a volverlo quinquenal:

```r
est %<>%
 group_by(country_code, year, eda5) %>%
  mutate(pop=sum(pop),
         deaths_age=sum(deaths_age)) %>%
  select(-age) %>%
  ungroup() %>%
  unique()

est
```

```
# A tibble: 44 x 6
   country_code name     year   pop deaths_age eda5
          <int> <chr>   <int> <dbl>      <dbl> <fct>
 1          858 Uruguay  1950  44.6      2.77  [0,1)
 2          858 Uruguay  1950 171.       0.312 [1,5)
 3          858 Uruguay  1950 205.       0.136 [5,10)
 4          858 Uruguay  1950 201.       0.105 [10,15)
 5          858 Uruguay  1950 200.       0.225 [15,20)
 6          858 Uruguay  1950 195.       0.290 [20,25)
 7          858 Uruguay  1950 178.       0.305 [25,30)
 8          858 Uruguay  1950 155.       0.317 [30,35)
 9          858 Uruguay  1950 159.       0.465 [35,40)
10          858 Uruguay  1950 139.       0.611 [40,45)
# i 34 more rows
```

Creamos las tasas por grupos

```r
est %<>%
  mutate(mx5=deaths_age/pop)
```

Necesitamos "c", es decir la estructura por edad de la población y de las . En este formato será más facil de calcular:

```r
est %<>%
  mutate(c=pop/sum(pop), .by = year)

est %>%
  summarise(suma= sum(c), .by = year)
```

```
# A tibble: 2 x 2
   year  suma
  <int> <dbl>
1  1950     1
2  2020     1
```

Si "recordamos":

$$TBM = \sum_{n=i}^{\omega} c_i * Mx_i$$

La suma de eso será la tasa:

```r
est %>%
 summarise(tbm=sum(mx5*c)*1000, .by = year)
```

```
# A tibble: 2 x 2
   year   tbm
  <int> <dbl>
1  1950 10.7
2  2020  9.50
```

Una estandarización sería usar la "c" de una de las poblaciones, como la de 1950

```r
c_1950 <- est %>%
  filter(year==1950) %>%
  select(country_code:year, eda5, c) %>%
  select(-year) %>%
  rename(c_1950=c)

est %<>%
  left_join(c_1950)
```

```
Joining with `by = join_by(country_code, name, eda5)`
```

```
est %>%
 summarise(tbm=sum(mx5*c_1950)*1000, .by = year)
```

```
# A tibble: 2 x 2
   year   tbm
  <int> <dbl>
1  1950 10.7
2  2020  4.77
```

Para estandarizar, cambiamos las "c", normalmente lo que se hace usar una $\bar{c}$, con

```
c_mean <- est %>%
  select(country_code:year, eda5, c) %>%
  mutate(c_mean=mean(c), .by = eda5) %>%
  select(-c(year, c)) %>%
  unique()

est %<>%
  left_join(c_mean)
```

```
Joining with `by = join_by(country_code, name, eda5)`
```

Vamos a sacar las tasas estandarizadas.

```
est %>%
 summarise(tbm=sum(mx5*c_mean)*1000, .by = year)
```

```
# A tibble: 2 x 2
   year   tbm
  <int> <dbl>
1  1950 15.2
2  2020  7.14
```

¡Es muy importante estandarizar!

4.7 Discrepancias

¿Qué parte de la diferencia de las tasas se debe al cambio etario y cuál al cambio de los riesgos de morir? Siguiendo a Kitagawa en Partida(2013), tenemos:

$$d^{2020} - d^{1950} = \sum_{n=i}^{\omega}(c_i^{2020} - c_i^{1950})\frac{(Mx_i^{2020} + Mx_i^{1950})}{2} + \sum_{n=i}^{\omega}(Mx_i^{2020} - Mx_i^{1950})\frac{(c_i^{2020} + c_i^{1950})}{2}$$

> El primer sumando en el lado derecho, esto es, la diferencia de las estructuras etarias ponderada por el promedio de las tasas específicas, mide el efecto de la disimiltud en las composiciones por edad; y el segundo, la diferencia de las pautas etarias de las tasas específicas ponderada por el promedio de las composiciones por edad de la población, da cuenta de la diferencia en el riesgo medio de morir (Partida,2013:p.63).

Para esto sería más fácil tener nuestro formato wide

```
est_wide<-est%>%
  select(-c_1950 ) %>%
  pivot_wider(
    names_from = year,
    values_from = pop:c,
    names_vary = "slowest"
  )
```

Caculemos el primer elemento

```
est_wide %>%
  summarise(e1=sum((c_2020-c_1950)    * (mx5_2020 +mx5_1950)/2),
            e2=sum((mx5_2020-mx5_1950) *(c_2020 +c_1950)/2) )
```

```
# A tibble: 1 x 2
       e1       e2
    <dbl>    <dbl>
1 0.00688 -0.00810
```

```
est %>%
 summarise(tbm=sum(mx5*c), .by = year)
```

```
# A tibble: 2 x 2
   year     tbm
  <int>   <dbl>
1  1950 0.0107
2  2020 0.00950
```

4.7.1 Momento de práctica

Compara las tasas brutas de mortalidad de 1990 con respecto a 2000

5 Migración y tablas de vida

5.1 Paquetes

5.1.1 Instalación local de paquetes si no hay internet

Carpeta de paquetes:

```
remotes::install_local(path = "paquetes/migest-master.zip",
                       upgrade = "never") # migest-master.zip

remotes::install_local(path = "paquetes/countrycode-main.zip",
                       upgrade = 'never') # fertestr-master.zip
```

5.1.2 paquetes

```
if (!require("pacman")) install.packages("pacman") # instala pacman si se requiere
```

```
Cargando paquete requerido: pacman
```

```
pacman::p_load(tidyverse,
               readxl,
               writexl,
               janitor,
               remotes,
               wppExplorer,
               magrittr,
               collapse,
               fmsb,
               migest, # ojo
               countrycode, # ojo
               DemoTools,
               wpp2022)
```

5.2 Datos

```
sv1992<-readxl::read_excel("datos/censos_p2.xlsx", sheet = "El Salvador 1992") %>%
  janitor::clean_names() %>% #
  dplyr::mutate(total=male + female) %>% # ojo
  dplyr::mutate(age=as.numeric(age)) #ojo

# de wpp2022

data("migration1dt")
data("mx1dt")
data("misc1dt")
data("popAge1dt")

mx1dt %<>%
  left_join(misc1dt) %>%
  left_join(popAge1dt)
```

```
Joining with `by = join_by(country_code, name, year)`
Joining with `by = join_by(country_code, name, year, age)`
```

```
data("e01dt")
```

Vamos a trabajar con datos ya calculados para flujos

```
# para flujos datos del autor Abel and Cohen (2019) estimates

# Estos cuando tengas internet se descargan así:
# f <- read_csv("https://ndownloader.figshare.com/files/38016762", show_col_types = FALSE)


f <- readRDS("datos/f.rds")

head(f)
```

```
# A tibble: 6 x 9
  year0 orig  dest  sd_drop_neg sd_rev_neg mig_rate da_min_open da_min_closed
  <dbl> <chr> <chr>       <dbl>      <dbl>    <dbl>       <dbl>         <dbl>
1  1990 BDI   BDI             0          0        0           0             0
2  1990 COM   BDI             0          0        0           0             0
```

```
3  1990 DJI   BDI             0           0      0           0             0
4  1990 ERI   BDI             0           0      0           0             0
5  1990 ETH   BDI             0           0      0           0             0
6  1990 KEN   BDI            30          30     75.7        51.3         207.
# i 1 more variable: da_pb_closed <dbl>
```

```
# Abel usa el paquete countrycode para generar regiones

f %>%
  mutate(
    orig = countrycode::countrycode(sourcevar = orig, # la variable de origen que tiene có
                                    custom_dict = dict_ims, # el diccionario a usar
                                    origin = "iso3c", # el tipo de abreviatura
                                    destination = "region"))  # que nos va a regresar
```

```
# A tibble: 307,833 x 9
   year0 orig   dest  sd_drop_neg sd_rev_neg mig_rate da_min_open da_min_closed
   <dbl> <chr>  <chr>       <dbl>      <dbl>    <dbl>       <dbl>         <dbl>
 1  1990 Africa BDI             0          0      0          0                0
 2  1990 Africa BDI             0          0      0          0                0
 3  1990 Africa BDI             0          0      0          0                0
 4  1990 Africa BDI             0          0      0          0                0
 5  1990 Africa BDI             0          0      0          0                0
 6  1990 Africa BDI            30         30     75.7       51.3            207.
 7  1990 Africa BDI             0          0      0          0.03             0
 8  1990 Africa BDI             0          0      0          0                0
 9  1990 Africa BDI             0          0      0          0.06             0
10  1990 Africa BDI             0          0      0          0                0
# i 307,823 more rows
# i 1 more variable: da_pb_closed <dbl>
```

```
d <- f %>%
  mutate(
    orig = countrycode::countrycode(sourcevar = orig, # la variable de origen que tiene có
                                    custom_dict = dict_ims, # el diccionario a usar
                                    origin = "iso3c", # el tipo de abreviatura
                                    destination = "region"),  # que nos va a regresar
    dest = countrycode::countrycode(sourcevar = dest,
                                    custom_dict = dict_ims,
                                    origin = "iso3c",
```

```
                                        destination = "region")
  ) %>%
  group_by(year0, orig, dest) %>%
  summarise_all(sum) %>%
  ungroup()
d
```

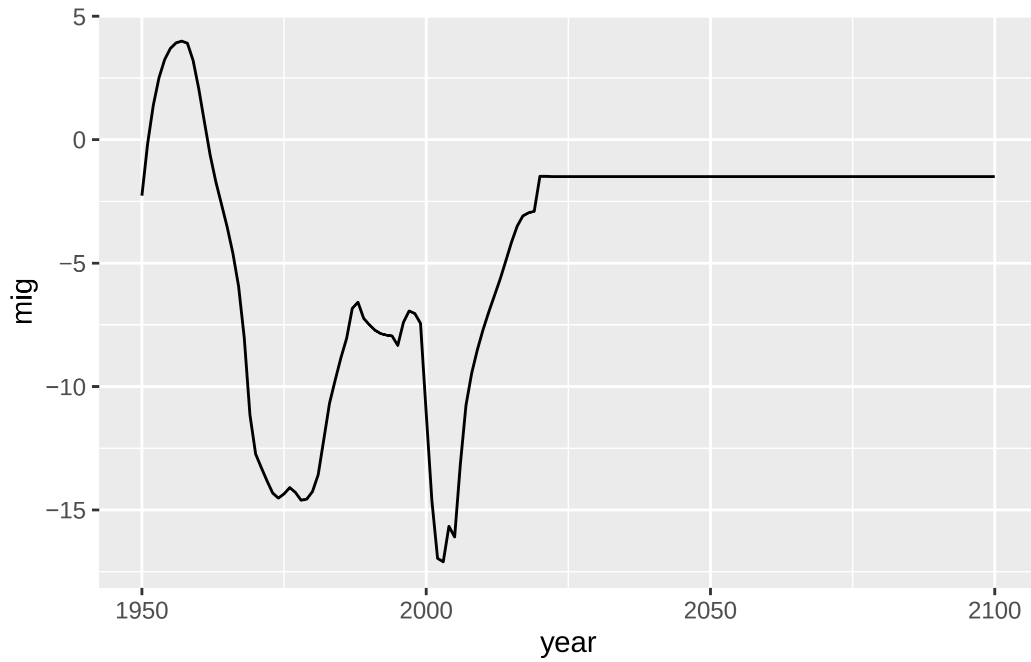
```
# A tibble: 216 x 9
   year0 orig   dest   sd_drop_neg sd_rev_neg mig_rate da_min_open da_min_closed
   <dbl> <chr>  <chr>        <dbl>      <dbl>    <dbl>       <dbl>         <dbl>
 1  1990 Africa Africa     4297155    7845806   5.47e6    6872677.      7728373.
 2  1990 Africa Asia        240464     258816   7.24e5     283708.       554047.
 3  1990 Africa Europe      555826     664496   1.91e6     830461.      2190967.
 4  1990 Africa Latin~        1505       2709   7.81e3       9043.        56747.
 5  1990 Africa North~      289058     301706   2.23e5     321650.       783334.
 6  1990 Africa Ocean~       21550      23570   6.59e4      30186.       165598.
 7  1990 Asia   Africa       94088     158903   2.00e5     102036.        93577.
 8  1990 Asia   Asia       3616112    8617460   1.44e7    6969956.     10337980.
 9  1990 Asia   Europe     1496141    2322839   5.48e6    2851352.      4214903.
10  1990 Asia   Latin~       14316      14343   1.07e5      20177.       136270.
# i 206 more rows
# i 1 more variable: da_pb_closed <dbl>
```

5.3 Migración

5.3.1 wpp2022

El conjunto de datos migration de wpp2022 nos da la migración neta

```
migration1dt %>%
  filter(name=="Uruguay") %>%
  ggplot() +
  aes(x=year, y=mig) +
  geom_line()
```



5.3.2 Flujos bilaterales

5.3.2.1 Gráficos circulaes

5.4 Diagramas

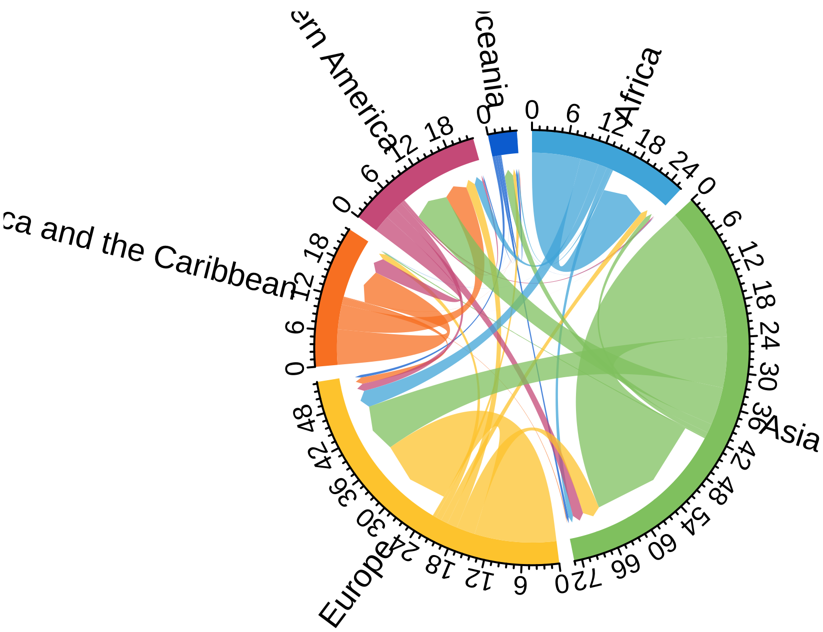
Vamos a usar "2015-2020 pseudo-Bayesian estimates for plotting" de la base de Guy Abel

```
pb <- d %>%
  filter(year0 == 2015) %>% # seleccionamos un año especifico
  mutate(flow = da_pb_closed/1e6) %>% # pasamos el flujo cerrado a miles
  select(orig, dest, flow) # necesitamos un objeto con estas tres variables
pb
```

```
# A tibble: 36 x 3
   orig   dest                              flow
   <chr>  <chr>                            <dbl>
 1 Africa Africa                          8.69
 2 Africa Asia                            0.896
 3 Africa Europe                          3.31
 4 Africa Latin America and the Caribbean  0.0361
```

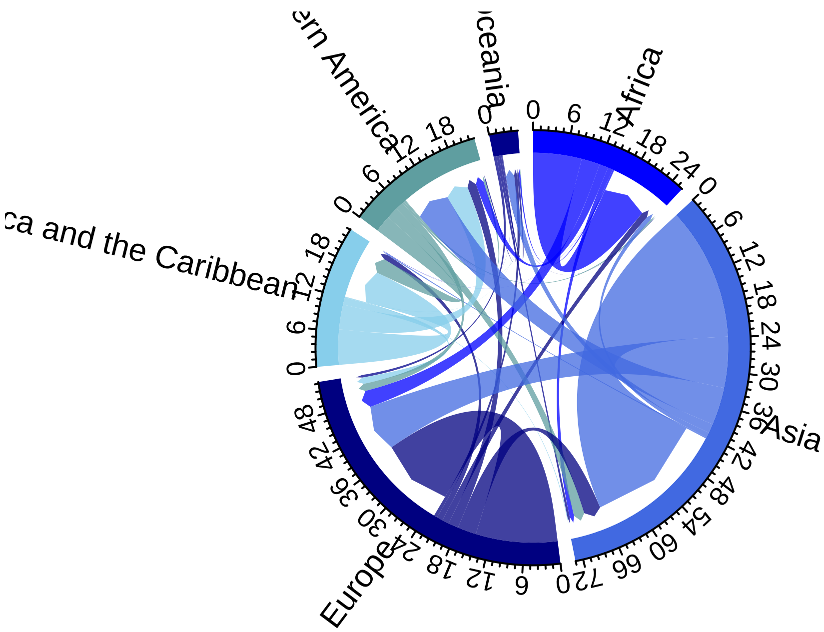
```
 5 Africa Northern America                 1.59 
 6 Africa Oceania                          0.264
 7 Asia   Africa                           0.907
 8 Asia   Asia                            23.8  
 9 Asia   Europe                           9.14 
10 Asia   Latin America and the Caribbean  0.233
# i 26 more rows
```

```r
migest::mig_chord(x = pb) # objeto

d %>%
  filter(year0 == 2015) %>% # seleccionamos un año especifico
  mutate(flow = da_pb_closed/1e6) %>% # pasamos el flujo cerrado a miles
  select(orig, dest, flow) %>%
  mig_chord()
```




```r
d %>%
  filter(year0 == 2015) %>% # seleccionamos un año especifico
  mutate(flow = da_pb_closed/1e6) %>% # pasamos el flujo cerrado a miles
  select(orig, dest, flow) %>%
  mig_chord(grid.col = c("blue", "royalblue", "navyblue", "skyblue", "cadetblue", "darkblu
```

```
)
```



5.4.1 Momento de práctica

5.5 Tabla de vida con `{DemoTools}`

Este paquete nos da la oportunidad de construir las tablas de vida con diferentes insumos, con **cualquiera** de las siguientes opciones:

- Vector de muertes y vector de Población media
- Vector de tasas de Mortalidad (nMx)
- Vector de cocientes de mortalidad (nqx)
- Vector de efectivos a edad exacta (lx)

5.5.1 Input: nMx

Datos de México 2000

```r
nMx <- c(0.025429618,
         0.000895531,
         0.000364678,
         0.000480071,
         0.000979976,
         0.001661119,
         0.002167313,
         0.002549786,
         0.00307099,
         0.003970018,
         0.005461053,
         0.007799417,
         0.011317907,
         0.016516166,
         0.024145341,
         0.035168272,
         0.051143602,
         0.074042144,
         0.136811785)
```

Nuestros grupos de edad

```r
grupo_eda<-c(0,1,seq(5,85,by=5))
AgeInt <- inferAgeIntAbr(vec = nMx)
```

```r
mx_lifetable2000 <- lt_abridged(nMx = nMx,
                                Age = grupo_eda,
                                AgeInt = AgeInt,
                                axmethod = "un",
                                Sex = "m",   mod = FALSE)
```

```r
nMx2010 <- c(0.018082902,
         0.000680864,
         0.000328649,
         0.000495605,
         0.001179152,
         0.002071347,
         0.002659697,
         0.002986375,
         0.003396466,
         0.004205501,
```

```
          0.00566518,
          0.008027968,
          0.011670117,
          0.017170344,
          0.025320726,
          0.037292527,
          0.054805372,
          0.079982834,
          0.148194498)


mx_lifetable2010 <- lt_abridged(nMx = nMx2010,
                                Age = grupo_eda,
                                AgeInt = AgeInt,
                                axmethod = "un",
                                Sex = "m",
                                mod = FALSE)
```

5.5.2 Desplegando hasta grupos más allá de 85+

Cerrar la tabla en ω=100

```
mx2020_100<-lt_abridged(nMx = nMx,
            Age = grupo_eda,
            AgeInt = AgeInt,
            axmethod = "un",
            Sex = "m",
            mod = FALSE,
            OAnew = 100)
```

5.5.3 Usando información de wpp2022

```
mx1dt %>%
  filter(country_code==858) %>%
  filter(year == 2020) -> data_uy

lt_uy_single<- lt_single_mx(nMx = data_uy$mxM,
               Age = data_uy$age,
               AgeInt =  inferAgeIntAbr(vec = data_uy$mxM),
```

```
                axmethod = "un",
                Sex = "m",
                mod = FALSE,
                OAnew = 100)

lt_uy_single
```

```
   Age AgeInt        nMx       nAx          nqx         lx        ndx
0    0      1 0.00532047 0.1386733 5.296199e-03 100000.000 529.619930
1    1      1 0.00013724 0.5000000 1.372306e-04  99470.380  13.650378
2    2      1 0.00011861 0.5000000 1.186030e-04  99456.730  11.795863
3    3      1 0.00010330 0.5000000 1.032947e-04  99444.934  10.272131
4    4      1 0.00009146 0.5000000 9.145582e-05  99434.662   9.093878
5    5      1 0.00008317 0.5000000 8.316654e-05  99425.568   8.268881
6    6      1 0.00007860 0.5000000 7.859691e-05  99417.299   7.813893
7    7      1 0.00007802 0.5000000 7.801696e-05  99409.485   7.755625
8    8      1 0.00008188 0.5000000 8.187665e-05  99401.729   8.138680
9    9      1 0.00009124 0.5000000 9.123584e-05  99393.591   9.068258
10  10      1 0.00010786 0.5000000 1.078542e-04  99384.522  10.719037
11  11      1 0.00013486 0.5000000 1.348509e-04  99373.803  13.400648
12  12      1 0.00017708 0.5000000 1.770643e-04  99360.403  17.593182
13  13      1 0.00024199 0.5000000 2.419607e-04  99342.810  24.037058
14  14      1 0.00033959 0.5000000 3.395323e-04  99318.773  33.721936
15  15      1 0.00048190 0.5000000 4.817839e-04  99285.051  47.833940
16  16      1 0.00067885 0.5000000 6.786197e-04  99237.217  67.344326
17  17      1 0.00093157 0.5000000 9.311363e-04  99169.872  92.340667
18  18      1 0.00122604 0.5000000 1.225289e-03  99077.532 121.398597
19  19      1 0.00152688 0.5000000 1.525715e-03  98956.133 150.978877
20  20      1 0.00178716 0.5000000 1.785564e-03  98805.154 176.422971
21  21      1 0.00195968 0.5000000 1.957762e-03  98628.731 193.091553
22  22      1 0.00202271 0.5000000 2.020666e-03  98435.640 198.905589
23  23      1 0.00198301 0.5000000 1.981046e-03  98236.734 194.611468
24  24      1 0.00187777 0.5000000 1.876009e-03  98042.123 183.927869
25  25      1 0.00175245 0.5000000 1.750916e-03  97858.195 171.341460
26  26      1 0.00164337 0.5000000 1.642021e-03  97686.853 160.403843
27  27      1 0.00157630 0.5000000 1.575059e-03  97526.449 153.609875
28  28      1 0.00155704 0.5000000 1.555829e-03  97372.840 151.495464
29  29      1 0.00158371 0.5000000 1.582457e-03  97221.344 153.848589
30  30      1 0.00164460 0.5000000 1.643249e-03  97067.496 159.506041
31  31      1 0.00172027 0.5000000 1.718792e-03  96907.989 166.564639
32  32      1 0.00179406 0.5000000 1.792452e-03  96741.425 173.404372
33  33      1 0.00185158 0.5000000 1.849867e-03  96568.020 178.638034
```

34	34	1	0.00189091	0.5000000	1.889124e-03	96389.382	182.091488
35	35	1	0.00192156	0.5000000	1.919716e-03	96207.291	184.690635
36	36	1	0.00195720	0.5000000	1.955287e-03	96022.600	187.751700
37	37	1	0.00201470	0.5000000	2.012673e-03	95834.849	192.884168
38	38	1	0.00210370	0.5000000	2.101490e-03	95641.964	200.990589
39	39	1	0.00222633	0.5000000	2.223854e-03	95440.974	212.246838
40	40	1	0.00237750	0.5000000	2.374677e-03	95228.727	226.137478
41	41	1	0.00254184	0.5000000	2.538614e-03	95002.590	241.174868
42	42	1	0.00270427	0.5000000	2.700618e-03	94761.415	255.914420
43	43	1	0.00285284	0.5000000	2.848776e-03	94505.500	269.225043
44	44	1	0.00298420	0.5000000	2.979754e-03	94236.275	280.800909
45	45	1	0.00310921	0.5000000	3.104384e-03	93955.474	291.673863
46	46	1	0.00324410	0.5000000	3.238846e-03	93663.800	303.362666
47	47	1	0.00341107	0.5000000	3.405262e-03	93360.438	317.916770
48	48	1	0.00363124	0.5000000	3.624659e-03	93042.521	337.247411
49	49	1	0.00392097	0.5000000	3.913298e-03	92705.274	362.783365
50	50	1	0.00429536	0.5000000	4.286155e-03	92342.490	395.794200
51	51	1	0.00476301	0.5000000	4.751694e-03	91946.696	436.902547
52	52	1	0.00533136	0.5000000	5.317186e-03	91509.793	486.574600
53	53	1	0.00600444	0.5000000	5.986467e-03	91023.219	544.907524
54	54	1	0.00678261	0.5000000	6.759686e-03	90478.311	611.604960
55	55	1	0.00766125	0.5000000	7.632015e-03	89866.706	685.864017
56	56	1	0.00863017	0.5000000	8.593090e-03	89180.842	766.339013
57	57	1	0.00967729	0.5000000	9.630691e-03	88414.503	851.492718
58	58	1	0.01078898	0.5000000	1.073109e-02	87563.011	939.646656
59	59	1	0.01196200	0.5000000	1.189088e-02	86623.364	1030.028082
60	60	1	0.01319940	0.5000000	1.311286e-02	85593.336	1122.373351
61	61	1	0.01451868	0.5000000	1.441404e-02	84470.963	1217.568134
62	62	1	0.01594341	0.5000000	1.581732e-02	83253.394	1316.845498
63	63	1	0.01749014	0.5000000	1.733851e-02	81936.549	1420.657959
64	64	1	0.01917061	0.5000000	1.898860e-02	80515.891	1528.883926
65	65	1	0.02097513	0.5000000	2.075744e-02	78987.007	1639.567669
66	66	1	0.02288317	0.5000000	2.262431e-02	77347.439	1749.932602
67	67	1	0.02487522	0.5000000	2.456963e-02	75597.507	1857.402959
68	68	1	0.02693509	0.5000000	2.657716e-02	73740.104	1959.802603
69	69	1	0.02908138	0.5000000	2.866458e-02	71780.301	2057.551991
70	70	1	0.03135622	0.5000000	3.087220e-02	69722.749	2152.494813
71	71	1	0.03383191	0.5000000	3.326913e-02	67570.254	2248.003638
72	72	1	0.03660293	0.5000000	3.594508e-02	65322.251	2348.013682
73	73	1	0.03975116	0.5000000	3.897648e-02	62974.237	2454.514084
74	74	1	0.04335636	0.5000000	4.243642e-02	60519.723	2568.240126
75	75	1	0.04746866	0.5000000	4.636814e-02	57951.483	2687.102656
76	76	1	0.05199502	0.5000000	5.067753e-02	55264.380	2800.662309

77	77	1	0.05703439	0.5000000	5.545303e-02	52463.718	2909.271875
78	78	1	0.06257633	0.5000000	6.067783e-02	49554.446	3006.856351
79	79	1	0.06861040	0.5000000	6.633477e-02	46547.590	3087.723767
80	80	1	0.07512615	0.5000000	7.240635e-02	43459.866	3146.770047
81	81	1	0.08211315	0.5000000	7.887482e-02	40313.096	3179.688183
82	82	1	0.08956096	0.5000000	8.572228e-02	37133.408	3183.160200
83	83	1	0.09745914	0.5000000	9.293067e-02	33950.247	3155.019195
84	84	1	0.10579725	0.5000000	1.004819e-01	30795.228	3094.362924
85	85	1	0.11456485	0.5000000	1.083578e-01	27700.865	3001.606205
86	86	1	0.12327489	0.5000000	1.161177e-01	24699.259	2868.020973
87	87	1	0.13261295	0.5000000	1.243666e-01	21831.238	2715.077670
88	88	1	0.14282711	0.5000000	1.333072e-01	19116.161	2548.321281
89	89	1	0.15405178	0.5000000	1.430344e-01	16567.839	2369.771378
90	90	1	0.16626274	0.5000000	1.535019e-01	14198.068	2179.430614
91	91	1	0.17936031	0.5000000	1.645990e-01	12018.637	1978.256183
92	92	1	0.19315316	0.5000000	1.761420e-01	10040.381	1768.532511
93	93	1	0.20746129	0.5000000	1.879637e-01	8271.849	1554.807218
94	94	1	0.22215733	0.5000000	1.999474e-01	6717.041	1343.055193
95	95	1	0.23734087	0.5000000	2.121634e-01	5373.986	1140.162919
96	96	1	0.25371726	0.5000000	2.251545e-01	4233.823	953.264235
97	97	1	0.27082236	0.5000000	2.385236e-01	3280.559	782.490736
98	98	1	0.28863464	0.5000000	2.522330e-01	2498.068	630.095360
99	99	1	0.30712501	0.5000000	2.662405e-01	1867.973	497.329962
100	100	NA	0.36629903	2.7129654	1.000000e+00	1370.643	1370.642941

	nLx	Sx	Tx	ex
0	99543.824	0.9954382	7468800.876	74.688009
1	99463.555	0.9991936	7369257.052	74.084939
2	99450.832	0.9998721	7269793.497	73.095039
3	99439.798	0.9998891	7170342.665	72.103650
4	99430.115	0.9999026	7070902.867	71.111047
5	99421.433	0.9999127	6971472.753	70.117505
6	99413.392	0.9999191	6872051.319	69.123295
7	99405.607	0.9999217	6772637.927	68.128689
8	99397.660	0.9999201	6673232.320	67.133966
9	99389.057	0.9999134	6573834.660	66.139422
10	99379.163	0.9999005	6474445.603	65.145411
11	99367.103	0.9998786	6375066.440	64.152384
12	99351.606	0.9998440	6275699.337	63.160969
13	99330.791	0.9997905	6176347.731	62.172066
14	99301.912	0.9997093	6077016.940	61.186992
15	99261.134	0.9995894	5977715.028	60.207604
16	99203.545	0.9994198	5878453.895	59.236384
17	99123.702	0.9991952	5779250.350	58.276271

18	99016.832	0.9989219	5680126.648	57.330119
19	98880.644	0.9986246	5581109.816	56.399837
20	98716.943	0.9983445	5482229.172	55.485255
21	98532.185	0.9981284	5383512.229	54.583610
22	98336.187	0.9980108	5284980.044	53.689701
23	98139.428	0.9979991	5186643.857	52.797397
24	97950.159	0.9980714	5088504.429	51.901206
25	97772.524	0.9981865	4990554.270	50.997817
26	97606.651	0.9983035	4892781.746	50.086389
27	97449.645	0.9983914	4795175.094	49.167945
28	97297.092	0.9984345	4697725.450	48.244721
29	97144.420	0.9984309	4600428.358	47.319119
30	96987.743	0.9983872	4503283.938	46.393326
31	96824.707	0.9983190	4406296.196	45.468864
32	96654.723	0.9982444	4309471.489	44.546289
33	96478.701	0.9981789	4212816.766	43.625382
34	96298.337	0.9981305	4116338.064	42.705306
35	96114.946	0.9980956	4020039.728	41.785188
36	95928.724	0.9980625	3923924.782	40.864596
37	95738.407	0.9980160	3827996.058	39.943675
38	95541.469	0.9979430	3732257.651	39.023222
39	95334.850	0.9978374	3636716.182	38.104349
40	95115.658	0.9977008	3541381.331	37.188162
41	94882.002	0.9975435	3446265.673	36.275492
42	94633.457	0.9973805	3351383.671	35.366543
43	94370.888	0.9972254	3256750.214	34.460959
44	94095.875	0.9970858	3162379.326	33.557983
45	93809.637	0.9969580	3068283.451	32.656782
46	93512.119	0.9968285	2974473.814	31.756920
47	93201.479	0.9966781	2880961.695	30.858485
48	92873.897	0.9964852	2787760.215	29.962217
49	92523.882	0.9962313	2694886.318	29.069396
50	92144.593	0.9959006	2602362.436	28.181636
51	91728.245	0.9954816	2510217.843	27.300794
52	91266.506	0.9949662	2418489.598	26.428752
53	90750.765	0.9943491	2327223.092	25.567357
54	90172.509	0.9936281	2236472.327	24.718325
55	89523.774	0.9928056	2146299.818	23.883148
56	88797.673	0.9918893	2056776.043	23.062981
57	87988.757	0.9908903	1967978.371	22.258547
58	87093.187	0.9898218	1879989.614	21.470134
59	86108.350	0.9886921	1792896.426	20.697608
60	85032.149	0.9875018	1706788.076	19.940665

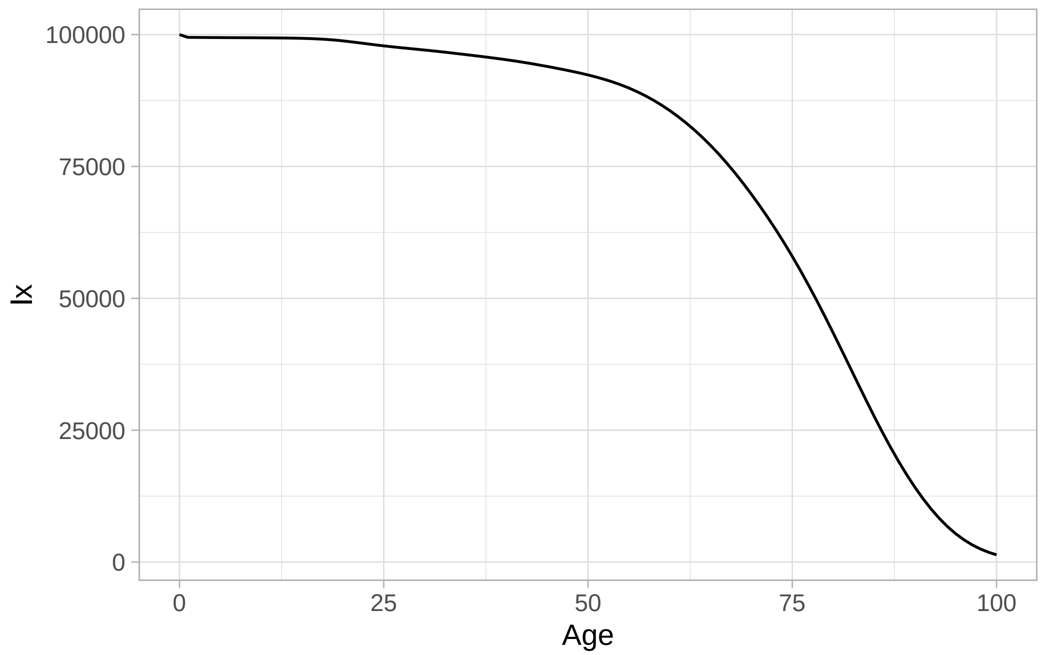
61	83862.179	0.9862408	1621755.927	19.198975
62	82594.972	0.9848894	1537893.749	18.472445
63	81226.220	0.9834281	1455298.777	17.761290
64	79751.449	0.9818437	1374072.557	17.065855
65	78167.223	0.9801355	1294321.108	16.386507
66	76472.473	0.9783189	1216153.885	15.723260
67	74668.805	0.9764142	1139681.412	15.075648
68	72760.203	0.9744391	1065012.606	14.442787
69	70751.525	0.9723932	992252.404	13.823464
70	68646.502	0.9702477	921500.879	13.216646
71	66446.253	0.9679481	852854.377	12.621743
72	64148.244	0.9654155	786408.124	12.038901
73	61746.980	0.9625670	722259.880	11.469133
74	59235.603	0.9593279	660512.900	10.914011
75	56607.932	0.9556403	601277.297	10.375529
76	53864.049	0.9515283	544669.366	9.855704
77	51009.082	0.9469968	490805.317	9.355138
78	48051.018	0.9420091	439796.235	8.875011
79	45003.728	0.9365822	391745.217	8.416015
80	41886.481	0.9307336	346741.489	7.978430
81	38723.252	0.9244809	304855.008	7.562183
82	35541.828	0.9178420	266131.756	7.166909
83	32372.738	0.9108349	230589.929	6.791996
84	29248.047	0.9034777	198217.191	6.436620
85	26200.062	0.8957884	168969.144	6.099779
86	23265.249	0.8879845	142769.082	5.780298
87	20473.699	0.8800121	119503.833	5.473983
88	17842.000	0.8714595	99030.133	5.180441
89	15382.954	0.8621765	81188.134	4.900345
90	13108.353	0.8521350	65805.180	4.634798
91	11029.509	0.8414108	52696.827	4.384593
92	9156.115	0.8301471	41667.318	4.149974
93	7494.445	0.8185180	32511.203	3.930343
94	6045.514	0.8066660	25016.759	3.724372
95	4803.905	0.7946231	18971.245	3.530200
96	3757.191	0.7821119	14167.340	3.346229
97	2889.314	0.7690090	10410.149	3.173285
98	2183.021	0.7555499	7520.835	3.010660
99	1619.308	0.7417740	5337.815	2.857544
100	3718.507	0.6966347	3718.507	2.712965

Comparemos:

```r
e01dt %>%
  filter(name == "Uruguay") %>%
  filter(year == 2020)
```

```
   country_code    name  year     e0M    e0F     e0B
          <int>  <char> <int>   <num>  <num>   <num>
1:          858 Uruguay  2020 74.6882 81.911 78.4298
```

```r
lt_uy_single %>%
  ggplot()+
  aes(x=Age,
      y=lx) +
  geom_line()+
  theme_light()
```



5.5.3.1 De una tabla quinquenal a una de edades simples

```
mx_single2000 <- lt_abridged2single(nMx = nMx,
                              Age = grupo_eda,
                              axmethod = "un",
                              Sex = "m",
                              mod = T,
                              OAnew = 100)
```

5.6 Pendientes

Con este paquete también podemos hacer evaluaciones, pero podemos hacer índices más complejos.

Trabaja con vectores individuales.

5.6.0.1 Whipple

```
check_heaping_whipple(Value=sv1992$total,
                      Age= sv1992$age,
                      ageMin = 25,
                      ageMax = 60,
                      digit = c(0, 5))
```

```
[1] 1.260264
```

5.6.0.2 Noumbissi

```
check_heaping_noumbissi(sv1992$male,
                        Age=sv1992$age,
                        ageMin = 30,
                        ageMax = 60,
                        digit = 0)
```

```
[1] 1.411808
```

Mayor a 1, el dígito atrae; menor que 1, el índice "repele"

Vamos a hacer un "loop"

```
# Para todos los dígitos

for(i in 0:2){
  Ni<-check_heaping_noumbissi(sv1992$total,
                              sv1992$age,
                              ageMin = 30+i, # ojo
                              ageMax = 60+i,
                              digit = i)
  names(Ni)<-i
  print(Ni)
}
```

```
       0
1.388983
        1
0.6612601
       2
1.202751
```

```
for(i in 3:9) {
  Ni<-check_heaping_noumbissi(sv1992$total,
                              sv1992$age,
                              ageMin = 20+i, #ojo
                              ageMax = 50+i,
                              digit = i)
  names(Ni)<-i
  print(Ni)
}
```

```
        3
0.9451307
        4
0.8988784
       5
1.145512
        6
0.9623871
```

```
       7
0.962849
       8
1.004804
       9
0.831283
```

5.6.0.3 Spoorrenberg

```r
check_heaping_spoorenberg(sv1992$total,
                          sv1992$age,
                          ageMin = 23,
                          ageMax = 62)
```

```
[1] 1.480261
```

5.6.0.4 Indice de Myers

```r
check_heaping_myers(Value = sv1992$male,
                    Age = sv1992$age,
                    ageMin = 10,
                    ageMax = 89,
                    method = "pasex")
```

```
[1] 6.906285
```

5.6.1 Opcional

sprague_male <- graduate(males$n, age, AgeInt = c(rep(5, times=17), 15), method = “sprague”)

single.age <- names2age(sprague_male)

ggplot() + aes(x=single.age, y=sprague_male) + geom_line()

sprague_female <- graduate(females$n, age, AgeInt = c(rep(5, times=17), 15), method = “sprague”) single.age <- names2age(sprague_female)

ggplot() + aes(x=single.age, y=sprague_female) + geom_line()

6 El grupo de 85 y más queda abierto...

ggplot() + geom_bar(aes(x=single.age, y=sprague_female, fill="Mujeres"), stat="identity") + geom_bar(aes(x=single.age, y=-sprague_male, fill="Hombres"), stat="identity")+ coord_flip() + labs(y= "Población desagregada")

7 Hay otros métodos pero este es el más importante.

Evaluación, ligas y más

Trabajo final

El trabajo consiste en un reporte que contenga en un documento lo siguiente

1. Una pirámide de alguna fuente de información, pueden ser las utilizadas en clases o las que estés trabajando en algún interés particular
2. Un diagrama de Lexis. Ya sea el tuyo propio (de tu vida) o de alguna investigación.
3. Un aplicación de cualquiera de las siguientes opciones:
    1. Mortalidad (una tabla de vida, tasas o estimaciones y descomposición)
    2. Fecundidad (tasas para una región o país durante varios años
    3. Migración (graficos de migración neta o gráficos circulares.
4. Adjuntar el código de R

Entrega: a más tardar 16 de agosto de 2024 Enviar a ana.escoto(at)gmail.com con el asunto [r_demo_uy] Nombres _ Apellidos

Códigos, proyecto, datos...

Si necesitas descargar el proyecto, el proyecto *vivo* está acá

Los códigos de las sesiones están en esta carpeta

La carpeta de datos

Otros cursos de la misma profe

Estos tienen videos, además.

- R para el análisis estadístico de datos
- Inferencia e introducción a los modelos estadísticos con R