

# **Introducción al Análisis Demográfico con R**

**XV Reunión Demográfica - SOMEDE 2025**

Ana Escoto

2025-06-17

# Table of contents

<b>Sobre el curso</b>	<b>4</b>
Docente . . . . .	4
Descripción del curso . . . . .	4
Contenidos temáticos . . . . .	4
1. Revisión de fuentes de datos demográficos y su importación en R . . . . .	4
2. Construcción y visualización de pirámides poblacionales . . . . .	4
3. Diagramas de Lexis para el análisis de eventos demográficos . . . . .	4
4. Cálculo de tasas de fecundidad, mortalidad y crecimiento poblacional . . . . .	4
5. Visualización de flujos migratorios con {migest} . . . . .	4
<b>Descargas</b>	<b>5</b>
<b>Bibliografía</b>	<b>6</b>
<b>1 1. Revisión de fuentes de datos demográficos y su importación en R</b>	<b>8</b>
1.1 Instalación de paquetes en desarrollo . . . . .	8
1.2 Dataframes con el paquete {WDI} . . . . .	9
1.3 Importación de datos . . . . .	31
1.3.1 Desde Excel . . . . .	31
1.3.2 Desde archivos de texto y de una url . . . . .	31
1.3.3 {wpp2024} . . . . .	32
1.4 De IPUMS . . . . .	34
1.5 {fmsb} Atracción digital . . . . .	35
1.5.1 Momento de práctica . . . . .	39
<b>2 Evaluación de información y pirámides</b>	<b>40</b>
2.1 Paquetes . . . . .	40
2.2 Pirámides . . . . .	40
2.2.1 Con grupos quinquenales . . . . .	40
2.2.2 Momento de práctica . . . . .	42
2.2.3 Pirámide en {ggplot2} . . . . .	42
2.2.4 Paquete {apyramid} . . . . .	49
2.2.5 Opcional . . . . .	51
2.2.6 Momento de práctica . . . . .	52

<b>3</b>	<b>Lexis</b>	<b>53</b>
3.1	Instalación local de los paquetes . . . . .	53
3.2	Diagrama de Lexis . . . . .	53
3.2.1	Dibujar una cuadrícula . . . . .	53
3.2.2	Sombreados en el diagrama . . . . .	57
3.2.3	Momento de práctica . . . . .	72
<b>4</b>	<b>Tasas</b>	<b>73</b>
4.1	Paquetes . . . . .	73
4.2	Datos . . . . .	73
4.3	Tasas de crecimiento . . . . .	74
4.3.1	Fechas censales y tasas de crecimiento . . . . .	74
4.3.2	Ritmo . . . . .	74
4.3.3	Crecimiento . . . . .	75
4.3.4	Crecimiento aritmético . . . . .	76
4.3.5	Crecimiento geométrico . . . . .	76
4.3.6	Crecimiento exponencial . . . . .	77
4.3.7	Proyeccion . . . . .	79
4.3.8	Tiempo . . . . .	81
4.4	Reconstrucción de las tasas de fecundidad . . . . .	81
4.4.1	Fusionando . . . . .	83
4.5	Nacimientos . . . . .	86
4.6	Mortalidad . . . . .	88
4.7	Discrepancias . . . . .	94
4.7.1	Momento de práctica . . . . .	95
<b>5</b>	<b>Migración y flujos</b>	<b>96</b>
5.1	Paquetes . . . . .	96
5.2	Datos . . . . .	96
5.3	Migración . . . . .	97
5.3.1	{wpp2024} . . . . .	97
5.4	Flujos bilaterales . . . . .	97

# Sobre el curso

## 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.

## Descripción del curso

La demografía utiliza diferentes fuentes de información para el análisis de la dinámica poblacional. Desde el software *R*, la comunidad académica ha desarrollado paquetes y códigos accesibles y replicables que permiten aplicar conceptos demográficos y estadísticos a fuentes de información actuales. Este taller tiene como objetivo que las personas participantes sean capaces de explorar, evaluar y analizar información poblacional de América Latina, con énfasis en México, a partir de fuentes censales, encuestas y bases internacionales, empleando herramientas del lenguaje *R*.

## Contenidos temáticos

1. Revisión de fuentes de datos demográficos y su importación en *R*
2. Construcción y visualización de pirámides poblacionales
3. Diagramas de Lexis para el análisis de eventos demográficos
4. Cálculo de tasas de fecundidad, mortalidad y crecimiento poblacional
5. Visualización de flujos migratorios con `{migest}`

# Descargas

Descarga los datos a usar [aquí](#)

# 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](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](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/wpp2024”. (2025) 2024. R. Probabilistic Projections Group. <https://github.com/PPgp/wpp2024>

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>.

Rodríguez, Germán. s/f. “Demographic Methods”.<https://grodriguez.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 Golemund, 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 Golemund. 2016. *R for data science: import, tidy, transform, visualize, and model data*. O'Reilly Media, Inc.

# 1. Revisión de fuentes de datos demográficos y su importación en R

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

Cargando paquete requerido: pacman

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

## 1.1 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 {wpp2024}

```
devtools::install_github("PPgp/wpp2024")
```

Skipping install of 'wpp2024' from a github remote, the SHA1 (bac89293) has not changed since  
Use `force = TRUE` to force installation

```
library(wpp2024)
```

Cargando paquete requerido: data.table

Warning: package 'data.table' was built under R version 4.4.1



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

## 1.2 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.CON
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.COGE.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  
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  
18473  
18480  
18486  
18493  
18500  
18504  
18517

UI  
UI  
U  
UIS: I  
UIS: Perc  
UIS: Pe

18525 Percentage of students in lower secondary  
 18528 Percentage of students in lower secondary  
 18532 Percentage of students in lower secondary education showing adequate understanding of  
 18536 Percentage of students in lower secondary education showing adequate understanding of  
 18540  
 18550 Percentage of students in lower secondary education showing adequate understanding of  
 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 showing adequate understanding of  
 18670 Percentage of students in lower secondary education showing adequate understanding of  
 18674 Percentage of students in lower secondary education showing adequate understanding of  
 18676 Percentage of students in lower secondary education showing adequate understanding of  
 18677 Percentage of students in lower secondary education showing adequate understanding of  
 18678 Percentage of students in lower secondary education showing adequate understanding of  
 18679 Percentage of students in lower secondary education showing adequate understanding of  
 18682 Percentage of students in lower secondary education showing adequate understanding of  
 18686 Percentage of students in lower secondary education showing adequate understanding of  
 18690 Percentage of students in lower secondary education showing adequate understanding of  
 18694 Percentage of students in lower secondary education showing adequate understanding of  
 18698 Percentage of students in lower secondary education showing adequate understanding of  
 18702  
 18706  
 18708  
 18711  
 18717  
 18719  
 18720

18723  
18725  
18726  
18742  
18746  
18750  
18754  
18758  
18762  
18766  
18770  
18774  
18793  
18798  
18802  
18805  
18810  
18814  
18819  
18825  
18829  
18835  
18840  
18844  
18848  
18864  
18880  
18907  
18915  
18922  
18929  
18936  
18943  
18950  
18955  
18959  
18963  
18967  
18971  
18977  
18982  
18986  
18990

Proportion

Proportion

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  
19321  
19346  
19350  
19363  
19378  
19392  
19396  
19400  
19404  
19408  
19463  
19479  
19495  
19553  
19557  
19561  
19565  
19569  
19573  
19577  
19583  
19591  
19598  
19605  
19612

Percentage of pupils  
Proportion of class

Particular

Proportion

Out-of-school

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	(
19697	
19704	
19711	
19718	(
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	On
19833	(
19837	(
19841	(
19845	On

19851  
19856  
19860  
19864  
19868  
19872  
19925  
19929  
19936  
19938  
19939  
19942  
19946  
19950  
19954  
19956  
19957  
19979  
19982  
19986  
19991  
20000  
20014  
20016  
20019  
20021  
20024  
20143  
20153

```
WDI::WDI(country = "MX",  
  indicator = "SP.POP.TOTL",  
  start = 2000,  
  end = 2024,  
  extra = FALSE,  
  cache = NULL)
```

	country	iso2c	iso3c	year	SP.POP.TOTL
1	Mexico	MX	MEX	2024	NA
2	Mexico	MX	MEX	2023	129739759
3	Mexico	MX	MEX	2022	128613117

4	Mexico	MX	MEX	2021	127648148
5	Mexico	MX	MEX	2020	126799054
6	Mexico	MX	MEX	2019	125762982
7	Mexico	MX	MEX	2018	124573711
8	Mexico	MX	MEX	2017	123400057
9	Mexico	MX	MEX	2016	122251351
10	Mexico	MX	MEX	2015	121072306
11	Mexico	MX	MEX	2014	119784261
12	Mexico	MX	MEX	2013	118343573
13	Mexico	MX	MEX	2012	116818208
14	Mexico	MX	MEX	2011	115243504
15	Mexico	MX	MEX	2010	113623895
16	Mexico	MX	MEX	2009	111999721
17	Mexico	MX	MEX	2008	110374288
18	Mexico	MX	MEX	2007	108774360
19	Mexico	MX	MEX	2006	107253666
20	Mexico	MX	MEX	2005	105811504
21	Mexico	MX	MEX	2004	104394131
22	Mexico	MX	MEX	2003	102978515
23	Mexico	MX	MEX	2002	101548624
24	Mexico	MX	MEX	2001	100099099
25	Mexico	MX	MEX	2000	98625552

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

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

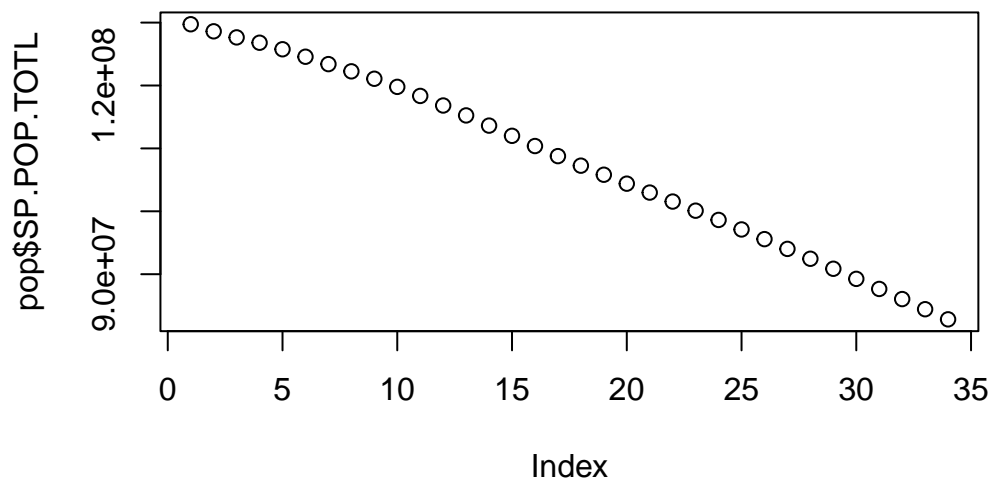
Vamos a revisar nuestro objeto:

```
class(pop)
```

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

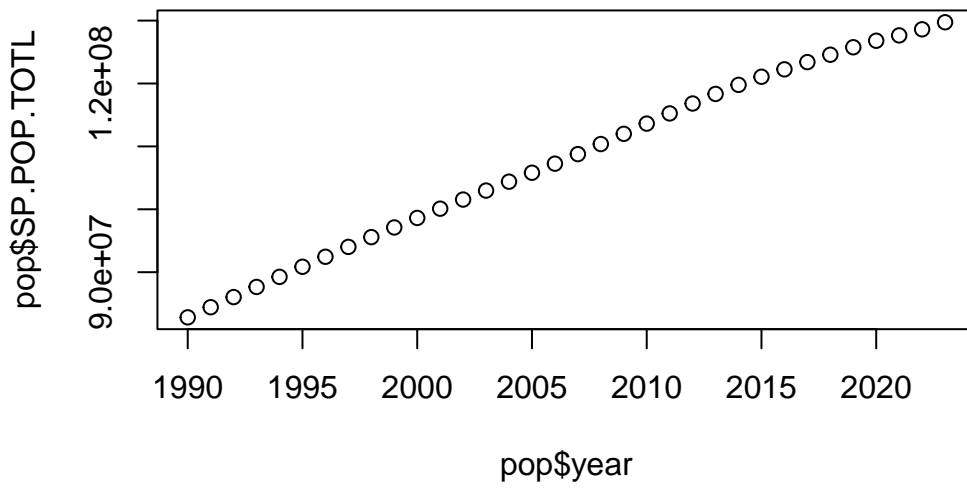
Veamos y conozcamos la función `plot()`

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



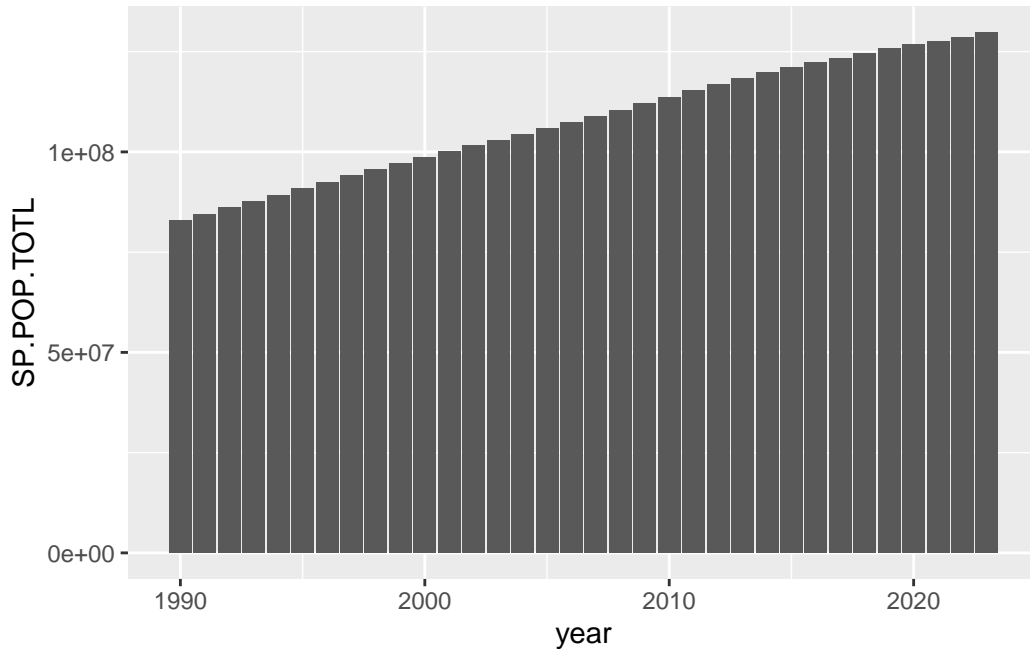
Este no es el mejor gráfico.

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



Un gráfico con {ggplot2}

```
pop %>%  
  ggplot2::ggplot() +  
  aes(x=year, y=SP.POP.TOTL) +  
  geom_col()
```



## 1.3 Importación de datos

### 1.3.1 Desde Excel

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

```
ejemploxl <- readxl::read_excel("datos/0_Pob_Mitad_1950_2070.xlsx")
```

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")
```

### 1.3.2 Desde archivos de texto y de una url

Desde el portal [https://conapo.segob.gob.mx/es/CONAPO/Datos\\_abiertos](https://conapo.segob.gob.mx/es/CONAPO/Datos_abiertos) 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 las proyecciones de población, pero hoy directamente desde este sitio y con formato de texto

```
pry23 <- read.csv("https://conapo.segob.gob.mx/work/models/CONAPO/Datos_Abiertos/pry23/00_
names(pry23)
```

```
[1] "REGLON" "AÑO" "ENTIDAD" "CVE_GEO" "EDAD" "SEXO"
[7] "POBLACION"
```

```
pry23 <- readr::read_csv("https://conapo.segob.gob.mx/work/models/CONAPO/Datos_Abiertos/pr
```

```
Rows: 737660 Columns: 7
```

```
-- Column specification -----
```

```
Delimiter: ","
```

```
chr (2): ENTIDAD, SEXO
```

```
dbl (5): REGLON, AÑO, CVE_GEO, EDAD, POBLACION
```

```
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(pry23)
```

```
[1] "REGLON" "AÑO" "ENTIDAD" "CVE_GEO" "EDAD" "SEXO"
[7] "POBLACION"
```

### 1.3.3 {wpp2024}

Vamos a utilizar datos del paquete {wpp2024}. Revisemos la viñeta del paquete que está [aquí](#)

Muy importante

Caution: All annual population datasets are considered to depict population to December 31 (midnight) of each year. This is different from the official WPP release, which treats population at January 1 (0h). Thus, the population numbers in this package are shifted by one year when compared to the official UN data. E.g., population in year 2050 in the R package corresponds to 2051 in the UN data. Vital rates and counts at time  $t$  refer to the calendar year  $t$ , so that they yield population at time  $t$  in this R package.



The 5-year datasets are created in legacy WPP 5-year periods so that vital rates observed/projected from July 1 of year  $t$  to June 30 of year  $t+5$  correspond to population at July 1 of year  $t+5$ . E.g. population in 2055 in a 5x5 dataset corresponds to vital rates/counts aggregated over 2050.5-2054.49. Since the 5x5 population is considered to July 1st, it corresponds to the interpolated values between year  $t$  and year  $t+1$  in the annual 1x1 dataset.

Todas las *data.frames* están en el paquete y si lo tenemos cargado podemos consultarlo con el comando `data()`

```
data("popAge5dt")
data("popprojAge5dt")
data("popAge1dt")
```

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
---		
293:	882	Samoa
294:	772	Tokelau
295:	776	Tonga
296:	798	Tuvalu
297:	876	Wallis and Futuna Islands

Podemos hacer búsquedas:

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

```

      country_code  name
      <int> <char>
1:          484 Mexico
2:          484 Mexico
3:          484 Mexico
4:          484 Mexico
5:          484 Mexico
---
311:         484 Mexico
312:         484 Mexico
313:         484 Mexico
314:         484 Mexico
315:         484 Mexico

```

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

```

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

popAge1dt<- popAge1dt %>%
  filter(country_code%in%c(484,1830))

```

También, tengo datos de algunos censos, descargados de ipums

## 1.4 De IPUMS

```

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()`

```
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
```

## 1.5 {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

```
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=total) %>%
  with(
    fmsb::WhipplesIndex(n) # se llama n por la segunda columna de la tabla anterior
  )
```

\$WI

```
[1] 102.4858
```

```
$JUDGE
```

```
[1] "highly accurate"
```

### **1.5.1 Momento de práctica**

Importa cualquier otro censo y encuentra el índice de Whipple

## 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,  
               wpp2024,  
               apyramid,  
               fmsb)
```

### 2.2 Pirámides

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

#### 2.2.1 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:



```
data("popAge1dt")

pob_mx<- popAge1dt %>%
  dplyr::filter(name=="Mexico") %>%
  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 intervalo
                        right=F)) # indica si el intervalo irá abierto en la derecha, ponemos un
```

Veamos esta variable:

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

	eda5	n
	<fctr>	<num>
1:	[0,5)	745149.938
2:	[5,10)	698422.625
3:	[10,15)	651597.722
4:	[15,20)	593967.895
5:	[20,25)	527228.518
6:	[25,30)	461932.892
7:	[30,35)	403449.304
8:	[35,40)	351074.518
9:	[40,45)	303107.968
10:	[45,50)	258285.820
11:	[50,55)	215494.976
12:	[55,60)	175213.095
13:	[60,65)	138112.572
14:	[65,70)	104856.354
15:	[70,75)	75892.530
16:	[75,80)	51216.662
17:	[80,85)	30912.693
18:	[85,90)	15501.847
19:	[90,95)	6079.788
20:	[95,100)	1783.451
21:	[100,105)	308.868

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

# A tibble: 22,725 x 7

	country_code	name	year	age	eda5	sexo	poblacion
	<int>	<chr>	<int>	<int>	<fct>	<chr>	<dbl>
1	484	Mexico	1949	0	[0,5)	popM	607.
2	484	Mexico	1949	0	[0,5)	popF	589.
3	484	Mexico	1949	0	[0,5)	pop	1196.
4	484	Mexico	1949	1	[0,5)	popM	546.
5	484	Mexico	1949	1	[0,5)	popF	534.
6	484	Mexico	1949	1	[0,5)	pop	1080.
7	484	Mexico	1949	2	[0,5)	popM	497.
8	484	Mexico	1949	2	[0,5)	popF	488.
9	484	Mexico	1949	2	[0,5)	pop	984.
10	484	Mexico	1949	3	[0,5)	popM	461.

# i 22,715 more rows

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

## 2.2.2 Momento de práctica

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

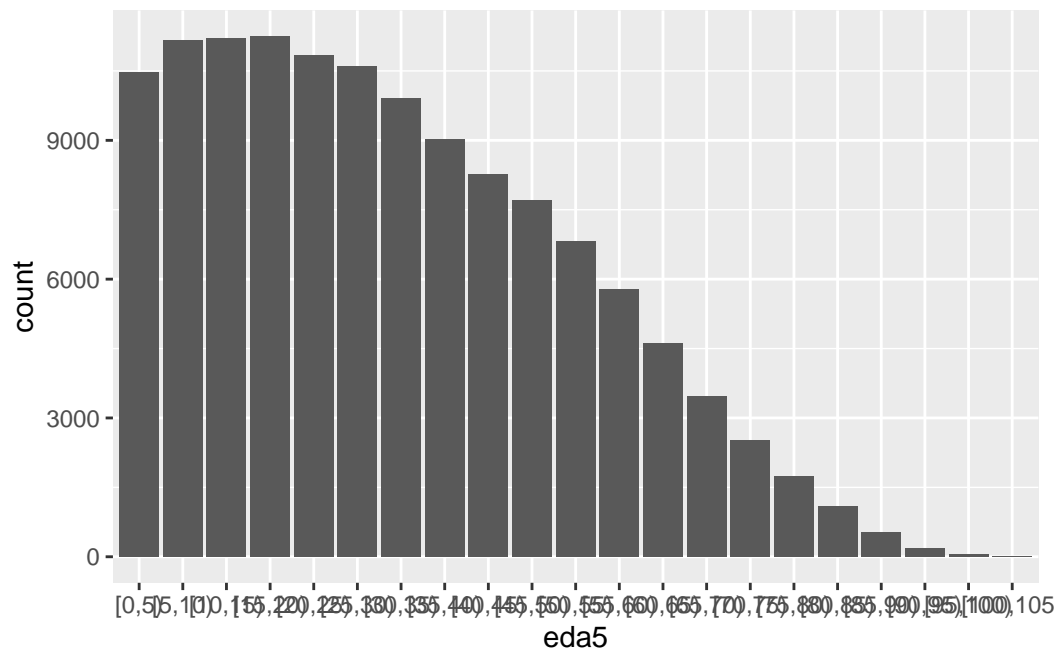
## 2.2.3 Pirámide en {ggplot2}

```
### gráfico de barras de edades quinquenales
pob_mx_long %>%
  dplyr::filter(year==2020) %>%
  ggplot2::ggplot() +
```

```

aes(x=eda5, weight=poblacion) +
geom_bar() # dibuja la geometría de barra

```

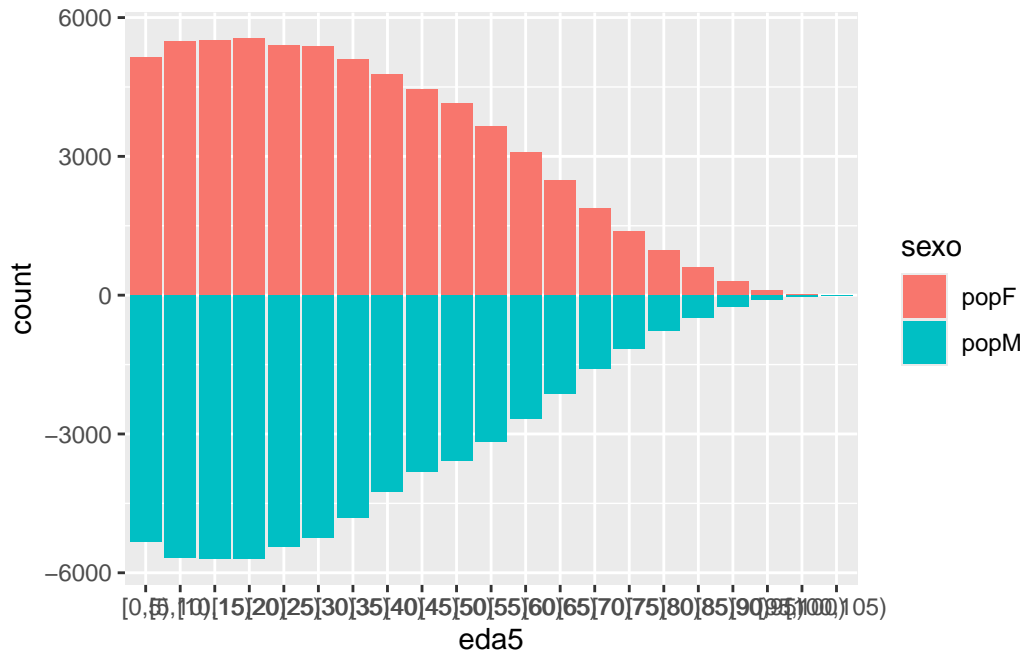


Una pirámide es un doble histograma por **sexo**, donde el valor de los hombres es negativo:

```

pob_mx_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

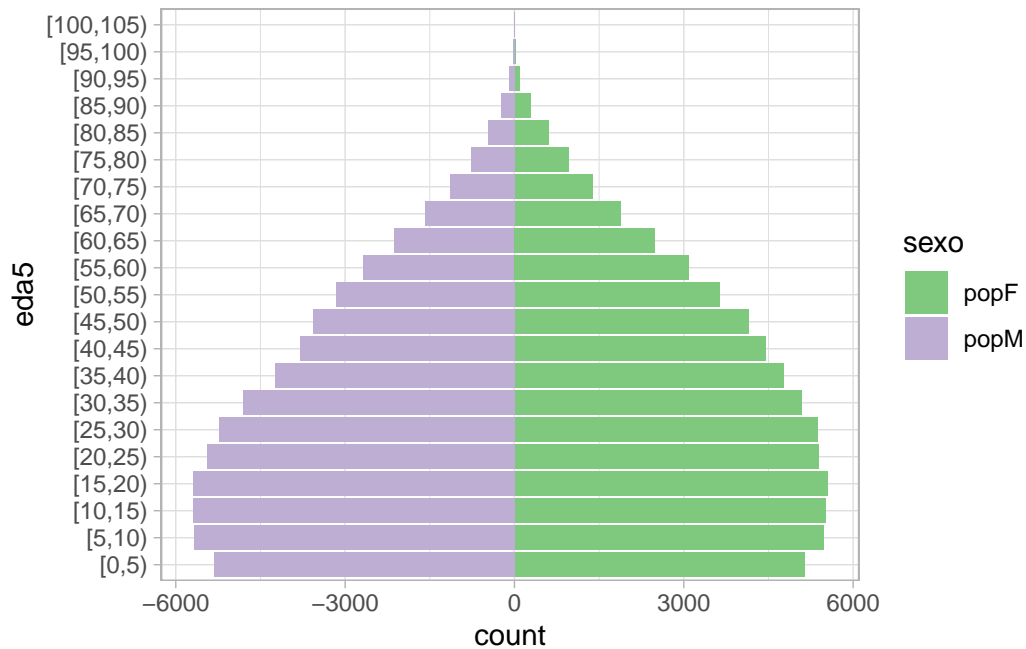
```



Podemos darle la vuelta y cambiarle los colores

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

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

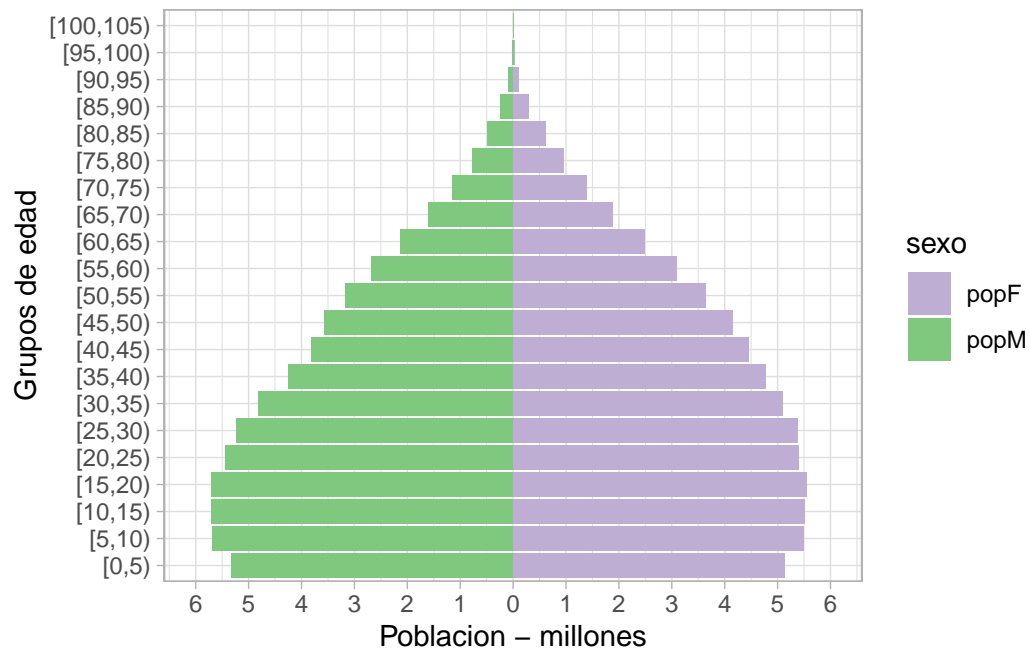
```
# A tibble: 1 x 2
  max    min
<dbl> <dbl>
1 5545. -5700.
```

```
pob_mx_long %>%
  filter(year==2020) %>%
  ggplot() +
  aes(eda5, fill=sexo, weight=poblacion2)+
  geom_bar() + coord_flip() +
  scale_y_continuous(breaks = seq(-6000, 6000, by=1000), # cuántos
                    limits = c(-6000, 6000),
```

```

labels = paste0(
  as.character(c(6:0,# sustituye negativos
  1:6) # Para lo positivo
)
)
)+
labs(y="Poblacion - millones", 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_mx_long<- pob_mx_long %>%
  mutate(p_edo=sum(poblacion), .by = year)

head(pob_mx_long)

```

# A tibble: 6 x 9

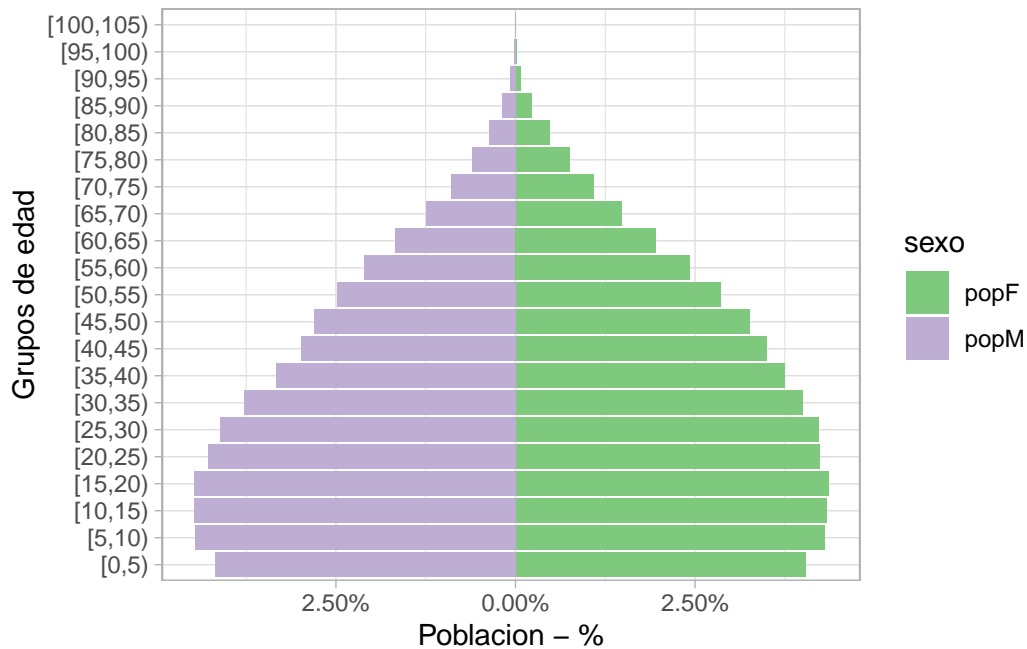
	country_code	name	year	age	eda5	sexo	poblacion	poblacion2	p_edo
	<int>	<chr>	<int>	<int>	<fct>	<chr>	<dbl>	<dbl>	<dbl>
1	484	Mexico	1949	0	[0,5)	popM	607.	-607.	27232.
2	484	Mexico	1949	0	[0,5)	popF	589.	589.	27232.
3	484	Mexico	1949	1	[0,5)	popM	546.	-546.	27232.
4	484	Mexico	1949	1	[0,5)	popF	534.	534.	27232.
5	484	Mexico	1949	2	[0,5)	popM	497.	-497.	27232.
6	484	Mexico	1949	2	[0,5)	popF	488.	488.	27232.

Hoy sí haremos lo mismo pero para las proporciones:

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

Una vez que ya tenemos nuestra variable proporcional:

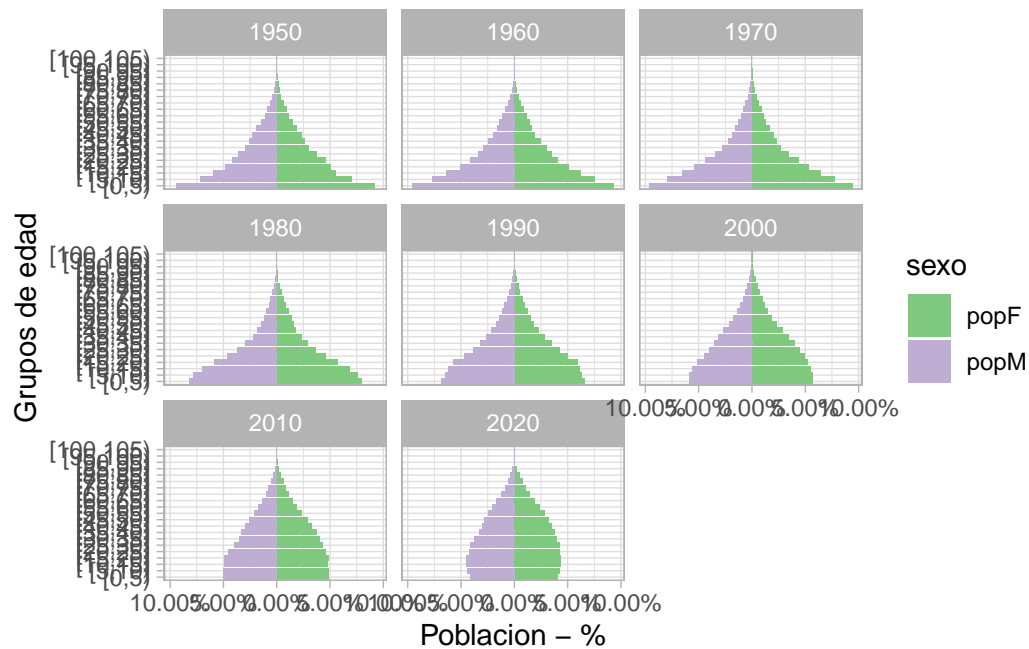
```
pob_mx_long%>%
  filter(year==2020) %>%
  ggplot(aes(eda5, fill=sexo, weight=poblacion3))+
  geom_bar() +
  coord_flip() +
  scale_y_continuous(labels = function(x) scales::percent(abs(x), accuracy = 0.01))+
  labs(y="Poblacion - %", x="Grupos de edad") +
  scale_fill_brewer(palette = "Accent") +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



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

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





## 2.2.4 Paquete {apyramid}

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

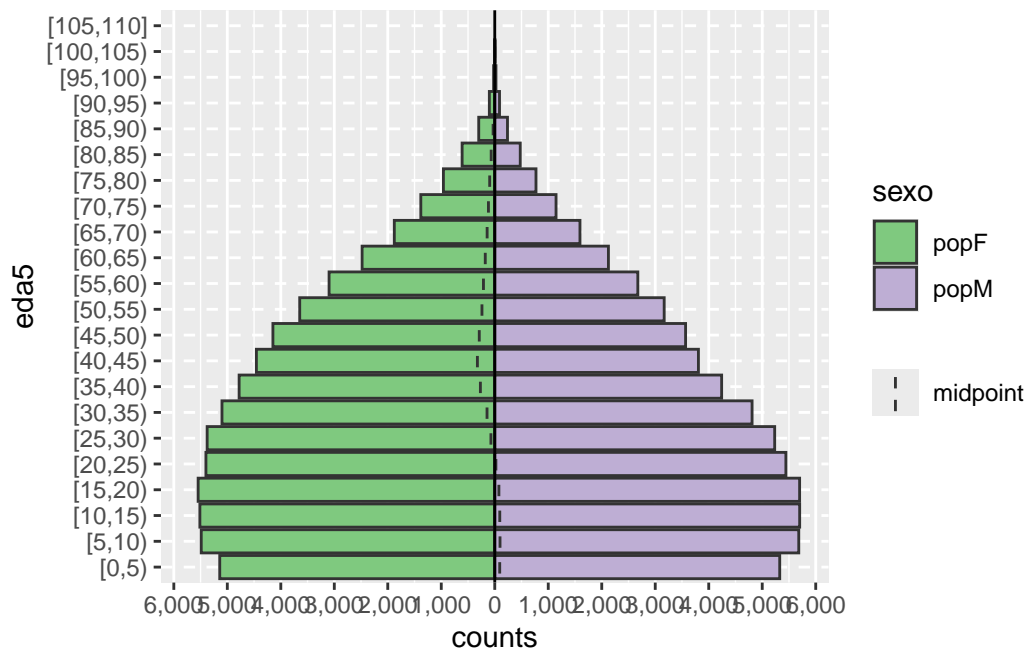
```
pob_mx_long %>%
  filter(year==2020) %>%
  count(eda5, sexo, wt=poblacion)
```

```
# A tibble: 42 x 3
  eda5      sexo      n
  <fct>    <chr> <dbl>
1 [0,5)   popF    5142.
2 [0,5)   popM    5327.
3 [5,10)  popF    5487.
4 [5,10)  popM    5683.
5 [10,15) popF    5513.
6 [10,15) popM    5700.
7 [15,20) popF    5545.
8 [15,20) popM    5699.
9 [20,25) popF    5400.
```

```
10 [20,25) popM 5442.
# i 32 more rows
```

```
pob_mx_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

#### 2.2.4.1 Momento de práctica

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

## 2.2.5 Opcional

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

```
data("popAge5dt")

#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)
}

```

### 2.2.6 Momento de práctica

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

## 3 Lexis

### 3.1 Instalación local de los paquetes

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

Cargando paquete requerido: pacman

```
pacman::p_load(tidyverse,  
               readxl,  
               writexl,  
               janitor,  
               remotes,  
               apyramid,  
               magrittr,  
               LexisPlotR)
```

### 3.2 Diagrama de Lexis

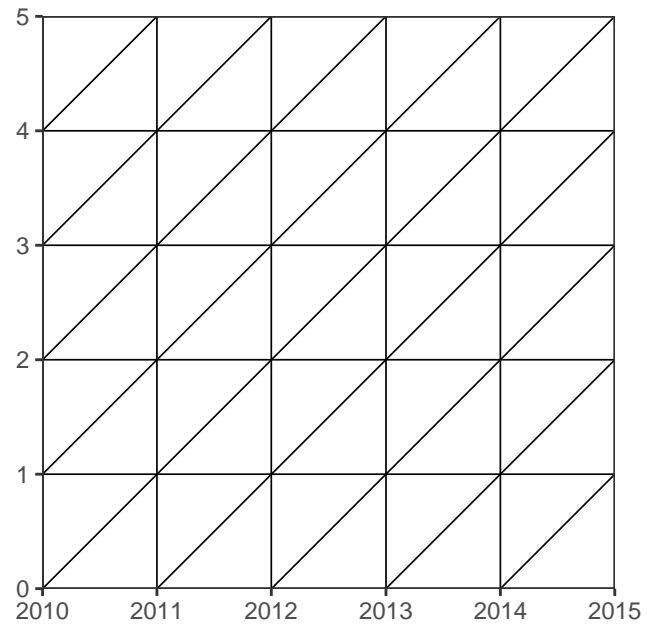
El paquete fue creado Philipp Ottolinger, este ejercicio es una versión en español (con algunos comentarios) de su ejemplo <https://github.com/ottlgr/LexisPlotR>

#### 3.2.1 Dibujar una cuadrícula

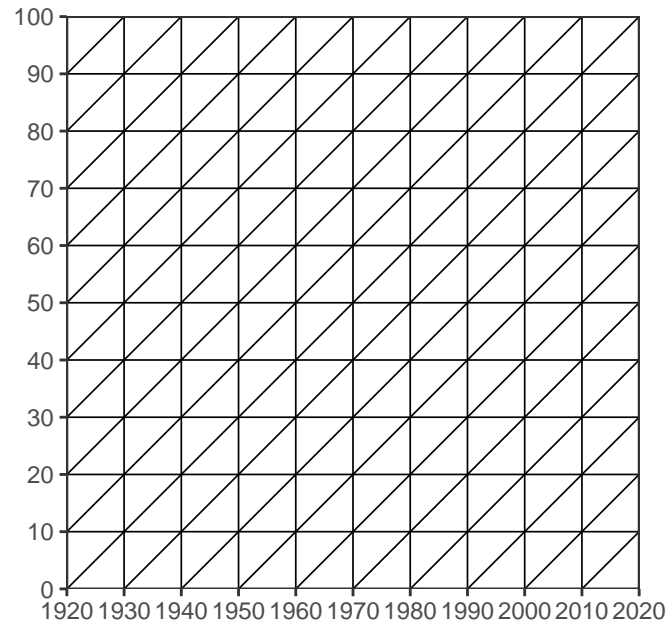
Este paquete nos puede ayudar a hacer nuestras cuadrículas. Ponemos los años de inicio y de final; así como las edades de inicio y de final. Recuerda que un diagrama de Lexis debe tener una misma escala en los ejes.

```
# 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)
```



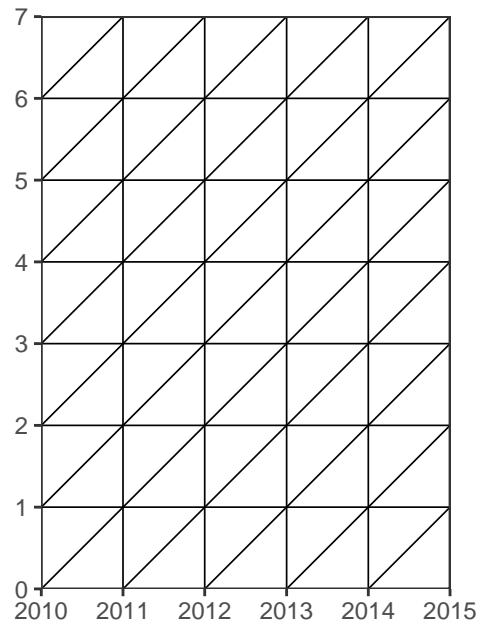
```
LexisPlotR::lexis_grid(year_start = 1920,  
  year_end = 2020,  
  age_start = 0,  
  age_end = 100,  
  delta = 10)
```



Aunque no necesariamente podemos dibujar sólo cuadrados

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

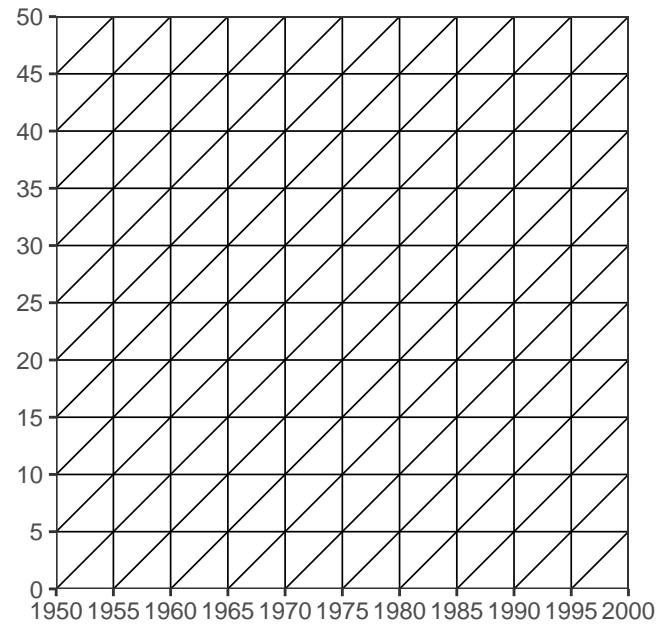
```
lexis_grid(year_start = 2010, year_end = 2015, age_start = 0, age_end = 7)
```



Si no ponemos nada específico 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)
```



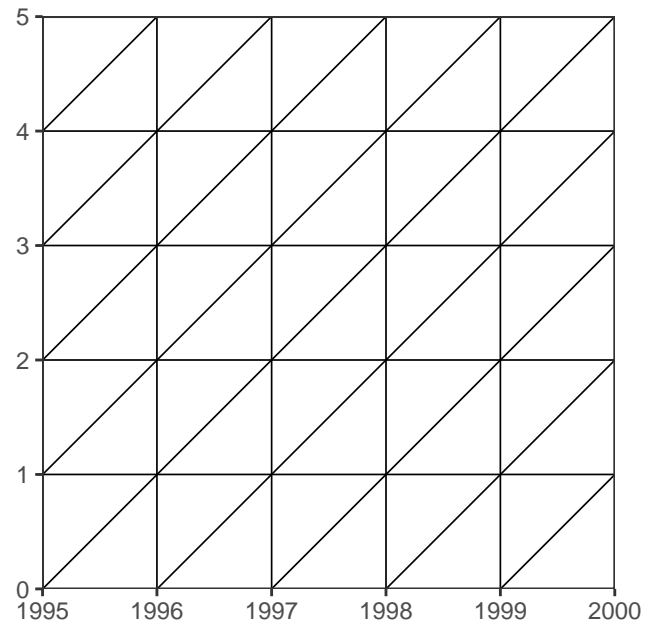


### 3.2.2 Sombreados en el diagrama

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)
```

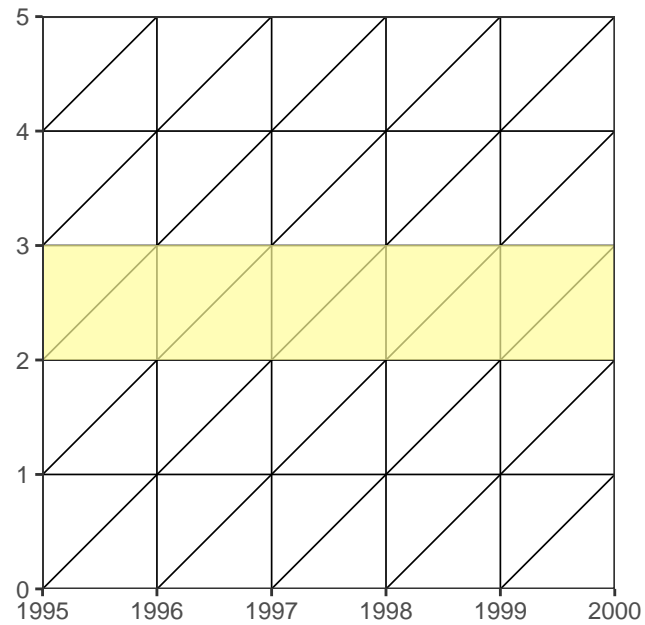
```
mi_diagrama
```



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

### 3.2.2.1 Edad

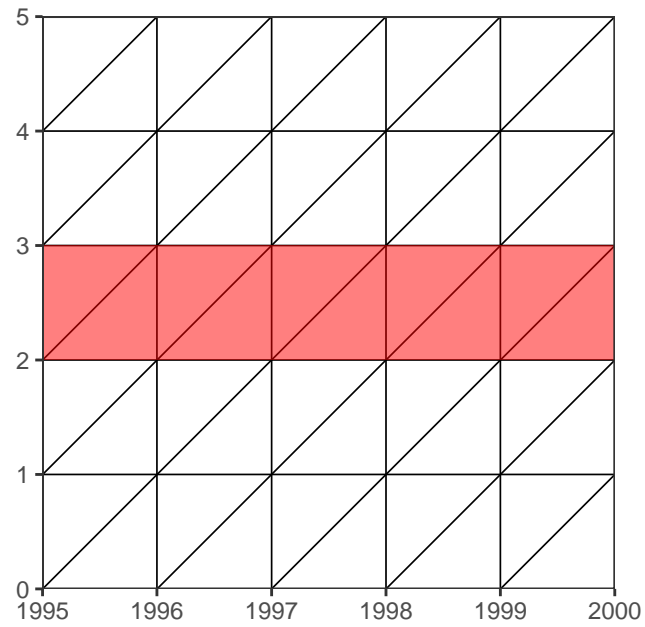
```
# Destacar todos los puntos que pertenecen a la edad de 2 años
mi_diagrama %>%
  lexis_age( age = 2)
```



¿Qué tipo de observación o estudio sería este?

Para cambiar el color:

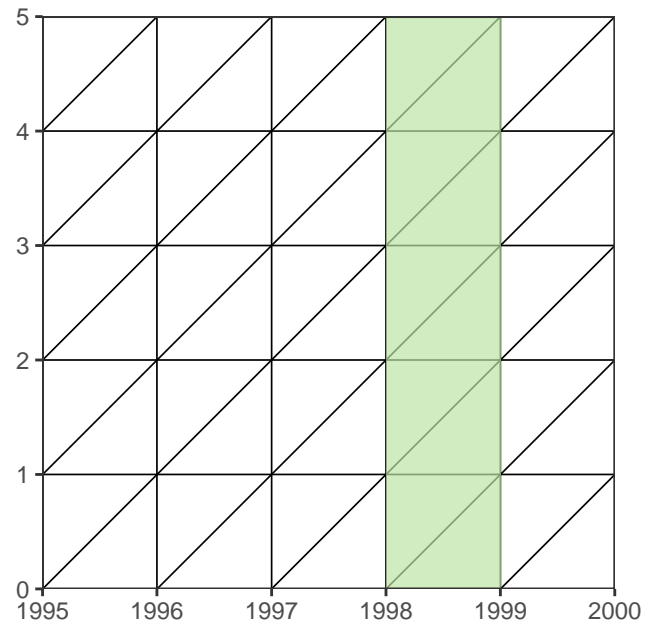
```
mi_diagrama %>%
  lexis_age(age = 2, fill = "red", alpha = 0.5)
```



### 3.2.2.2 Periodo

También podemos sombrear períodos

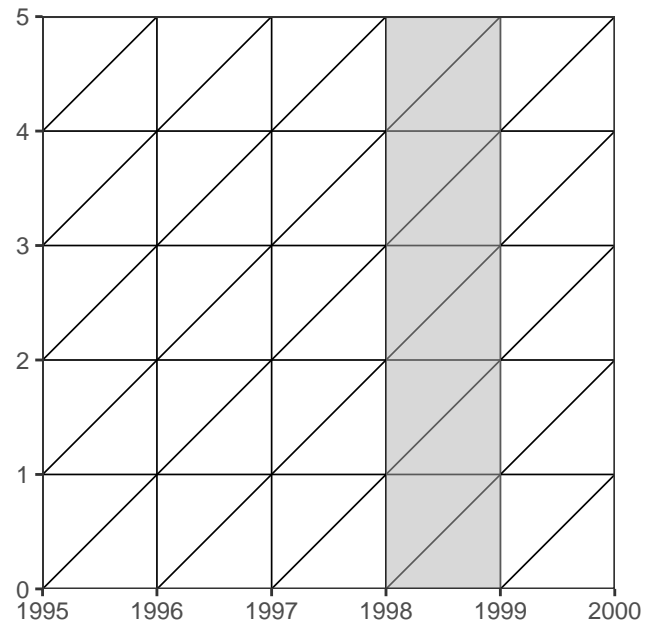
```
mi_diagrama %>%
  lexis_year(year=1998)
```



¿Qué tipo de observación o estudio sería este?

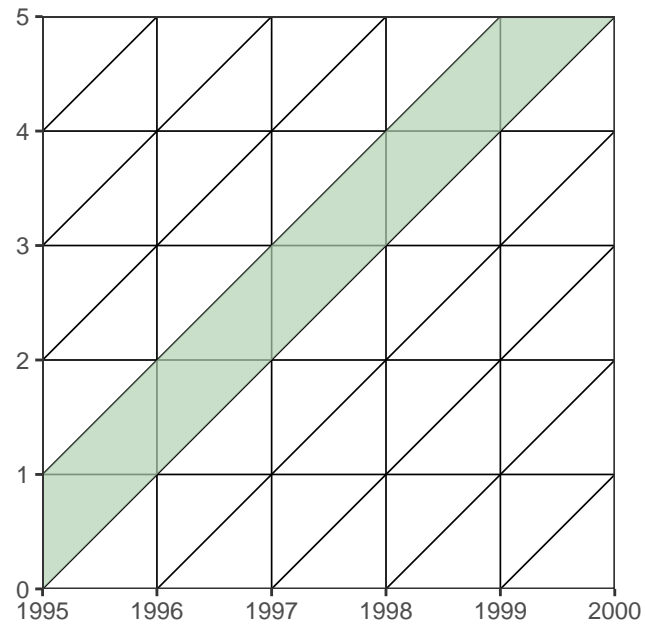
Para cambiar el color: [Más info del color](#)

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



### 3.2.2.3 Cohorte

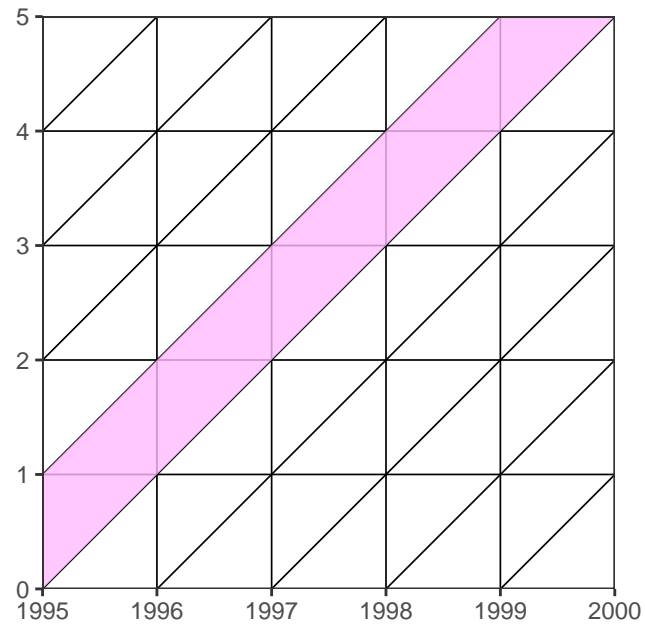
```
lexis_cohort(lg = mi_diagrama, cohort=1994)
```



¿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)
```

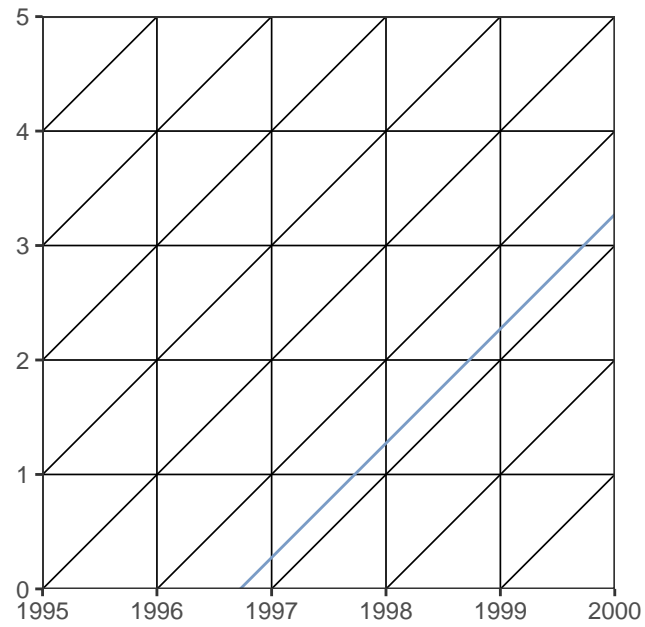


#### 3.2.2.4 Líneas de vida

Alguien entra

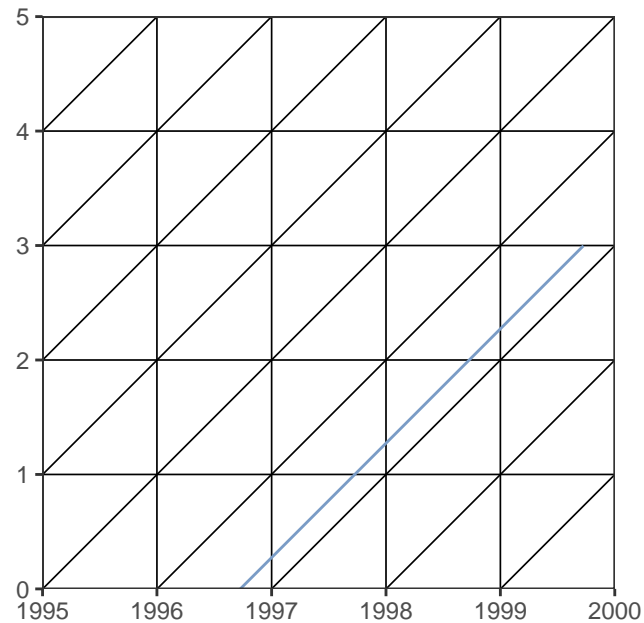
```
lexis_lifeline(lg = mi_diagrama, birth = "1996-09-23")
```





Alguien entra y sale

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

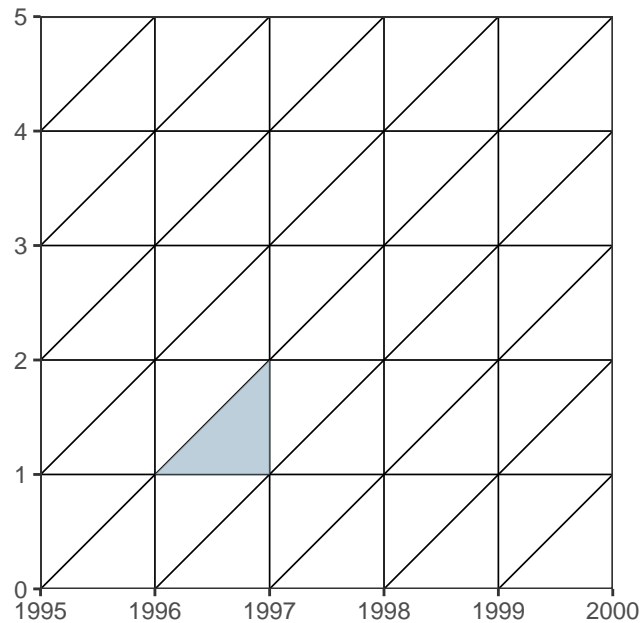


### 3.2.2.5 Polígonos

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)
```



chea 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:

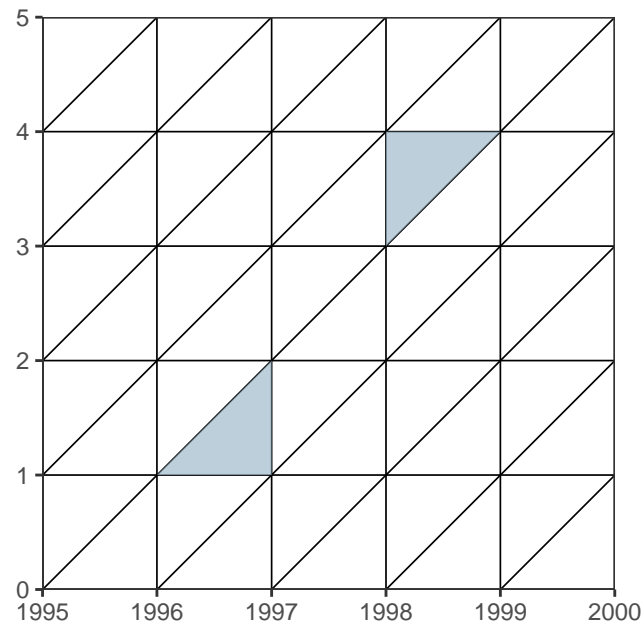
```
polygons <- data.frame(group = c(1,
                                1, # es un triángulo
                                2,
                                2,
                                2), # es otro triángulo
                       x = c("1996-01-01",
                             "1997-01-01",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

```



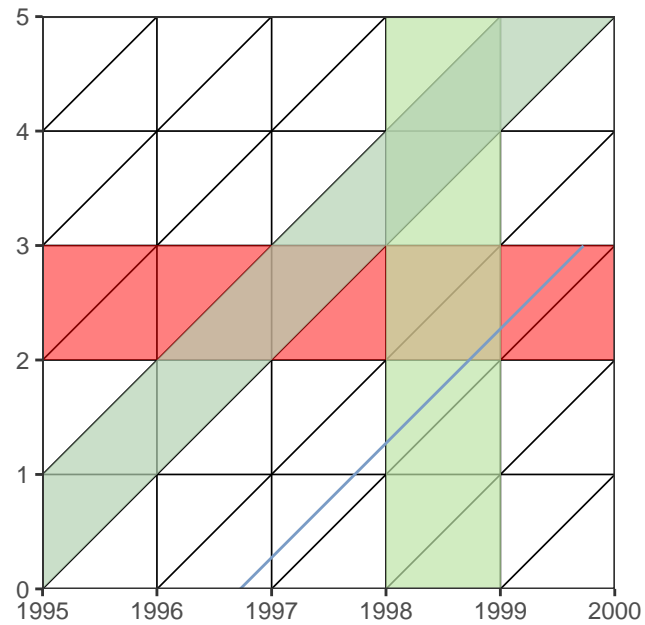
### 3.2.2.6 Todo en uno

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")

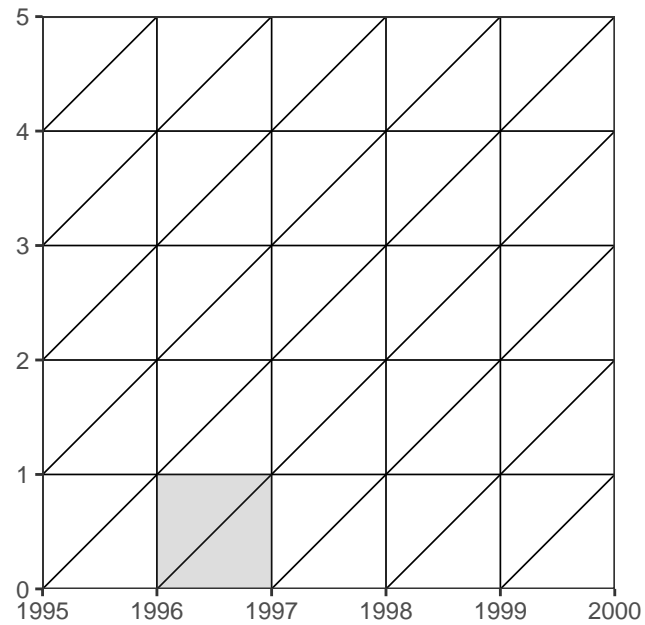
```



### 3.2.2.7 Anotación manual

Para hacer cuadrados

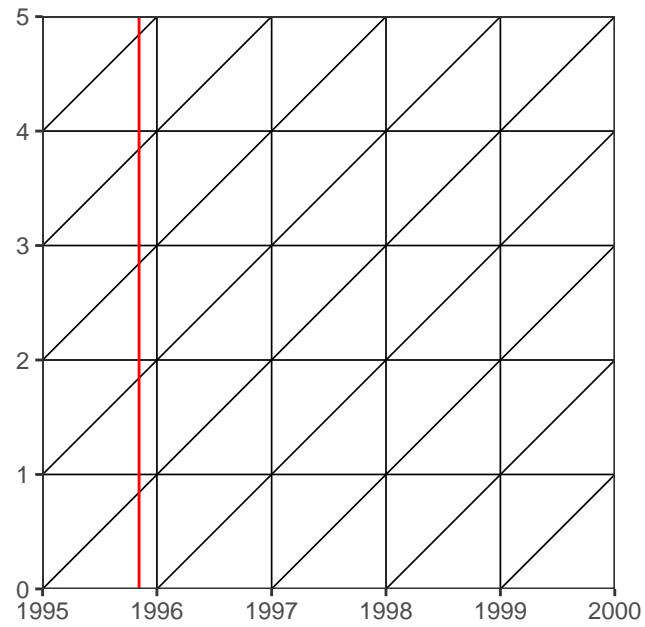
```
mi_diagrama + # ojo este siguiente comando es ggplot
  annotate("rect",
    xmin = as.Date("1996-01-01"),
    xmax = as.Date("1997-01-01"),
    ymin = 0, ymax = 1, alpha = .2)
```



¿Qué tipo de observación es esta?

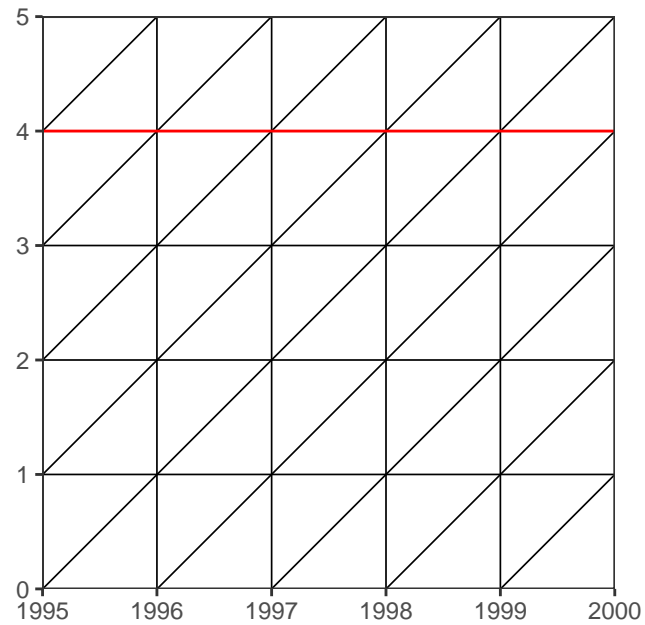
Si quisiéramos graficar la fecha de un censo:

```
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")
```



### 3.2.3 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,  
               wpp2024)
```

### 4.2 Datos

```
censos <- readxl::read_excel("datos/censos.xlsx",  
                             col_types = c("numeric", "date", "numeric"))  
  
data("tfr1dt")  
data("percentASFR1dt")  
data("misc1dt")  
data("mx1dt")  
data("pop1dt")  
data("popAge1dt")
```

## 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: 13 x 5
```

	ano	fecha		poblacion	dias	n
	<dbl>	<dtm>		<dbl>	<dbl>	<dbl>
1	1900	1900-10-28 00:00:00		13607272	NA	NA
2	1910	1910-10-27 00:00:00		15160369	3651	10.0
3	1921	1921-10-20 00:00:00		14334780	4011	11.0
4	1930	1930-05-15 00:00:00		16552722	3129	8.57
5	1940	1940-03-06 00:00:00		19653552	3583	9.82
6	1950	1950-06-06 00:00:00		25791017	3744	10.3
7	1960	1960-06-08 00:00:00		34923129	3655	10.0
8	1970	1970-01-28 00:00:00		48225238	3521	9.65
9	1980	1980-06-04 00:00:00		66846833	3780	10.4
10	1990	1990-03-12 00:00:00		81249645	3568	9.78
11	2000	2000-02-14 00:00:00		97483412	3626	9.93
12	2010	2010-06-12 00:00:00		112336538	3771	10.3
13	2020	2020-03-15 00:00:00		126014024	3564	9.76

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

### 4.3.2 Ritmo

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

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

# A tibble: 13 x 6

	ano	fecha		poblacion	dias	n	ritmo
	<dbl>	<dtm>		<dbl>	<dbl>	<dbl>	<dbl>
1	1900	1900-10-28 00:00:00		13607272	NA	NA	NA
2	1910	1910-10-27 00:00:00		15160369	3651	10.0	1.11
3	1921	1921-10-20 00:00:00		14334780	4011	11.0	0.946
4	1930	1930-05-15 00:00:00		16552722	3129	8.57	1.15
5	1940	1940-03-06 00:00:00		19653552	3583	9.82	1.19
6	1950	1950-06-06 00:00:00		25791017	3744	10.3	1.31
7	1960	1960-06-08 00:00:00		34923129	3655	10.0	1.35
8	1970	1970-01-28 00:00:00		48225238	3521	9.65	1.38
9	1980	1980-06-04 00:00:00		66846833	3780	10.4	1.39
10	1990	1990-03-12 00:00:00		81249645	3568	9.78	1.22
11	2000	2000-02-14 00:00:00		97483412	3626	9.93	1.20
12	2010	2010-06-12 00:00:00		112336538	3771	10.3	1.15
13	2020	2020-03-15 00:00:00		126014024	3564	9.76	1.12

### 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

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

# A tibble: 13 x 7

	ano	fecha		poblacion	dias	n	ritmo	c
	<dbl>	<dtm>		<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1900	1900-10-28 00:00:00		13607272	NA	NA	NA	NA
2	1910	1910-10-27 00:00:00		15160369	3651	10.0	1.11	0.114
3	1921	1921-10-20 00:00:00		14334780	4011	11.0	0.946	-0.0545
4	1930	1930-05-15 00:00:00		16552722	3129	8.57	1.15	0.155
5	1940	1940-03-06 00:00:00		19653552	3583	9.82	1.19	0.187
6	1950	1950-06-06 00:00:00		25791017	3744	10.3	1.31	0.312
7	1960	1960-06-08 00:00:00		34923129	3655	10.0	1.35	0.354
8	1970	1970-01-28 00:00:00		48225238	3521	9.65	1.38	0.381
9	1980	1980-06-04 00:00:00		66846833	3780	10.4	1.39	0.386
10	1990	1990-03-12 00:00:00		81249645	3568	9.78	1.22	0.215
11	2000	2000-02-14 00:00:00		97483412	3626	9.93	1.20	0.200

12	2010	2010-06-12 00:00:00	112336538	3771	10.3	1.15	0.152
13	2020	2020-03-15 00:00:00	126014024	3564	9.76	1.12	0.122

#### 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: 13 x 8

	ano	fecha	poblacion	dias	n	ritmo	c	ra
	<dbl>	<dtm>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1900	1900-10-28 00:00:00	13607272	NA	NA	NA	NA	NA
2	1910	1910-10-27 00:00:00	15160369	3651	10.0	1.11	0.114	0.0114
3	1921	1921-10-20 00:00:00	14334780	4011	11.0	0.946	-0.0545	-0.00496
4	1930	1930-05-15 00:00:00	16552722	3129	8.57	1.15	0.155	0.0180
5	1940	1940-03-06 00:00:00	19653552	3583	9.82	1.19	0.187	0.0191
6	1950	1950-06-06 00:00:00	25791017	3744	10.3	1.31	0.312	0.0304
7	1960	1960-06-08 00:00:00	34923129	3655	10.0	1.35	0.354	0.0354
8	1970	1970-01-28 00:00:00	48225238	3521	9.65	1.38	0.381	0.0395
9	1980	1980-06-04 00:00:00	66846833	3780	10.4	1.39	0.386	0.0373
10	1990	1990-03-12 00:00:00	81249645	3568	9.78	1.22	0.215	0.0220
11	2000	2000-02-14 00:00:00	97483412	3626	9.93	1.20	0.200	0.0201
12	2010	2010-06-12 00:00:00	112336538	3771	10.3	1.15	0.152	0.0147
13	2020	2020-03-15 00:00:00	126014024	3564	9.76	1.12	0.122	0.0125

#### 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: 13 x 9
```

	ano	fecha	poblacion	dias	n	ritmo	c	ra
	<dbl>	<dtm>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1900	1900-10-28 00:00:00	13607272	NA	NA	NA	NA	NA
2	1910	1910-10-27 00:00:00	15160369	3651	10.0	1.11	0.114	0.0114
3	1921	1921-10-20 00:00:00	14334780	4011	11.0	0.946	-0.0545	-0.00496
4	1930	1930-05-15 00:00:00	16552722	3129	8.57	1.15	0.155	0.0180
5	1940	1940-03-06 00:00:00	19653552	3583	9.82	1.19	0.187	0.0191
6	1950	1950-06-06 00:00:00	25791017	3744	10.3	1.31	0.312	0.0304
7	1960	1960-06-08 00:00:00	34923129	3655	10.0	1.35	0.354	0.0354
8	1970	1970-01-28 00:00:00	48225238	3521	9.65	1.38	0.381	0.0395
9	1980	1980-06-04 00:00:00	66846833	3780	10.4	1.39	0.386	0.0373
10	1990	1990-03-12 00:00:00	81249645	3568	9.78	1.22	0.215	0.0220
11	2000	2000-02-14 00:00:00	97483412	3626	9.93	1.20	0.200	0.0201
12	2010	2010-06-12 00:00:00	112336538	3771	10.3	1.15	0.152	0.0147
13	2020	2020-03-15 00:00:00	126014024	3564	9.76	1.12	0.122	0.0125

```
# 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: 13 x 10
```

	ano	fecha	poblacion	dias	n	ritmo	c	ra
	<dbl>	<dtm>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1900	1900-10-28 00:00:00	13607272	NA	NA	NA	NA	NA
2	1910	1910-10-27 00:00:00	15160369	3651	10.0	1.11	0.114	0.0114
3	1921	1921-10-20 00:00:00	14334780	4011	11.0	0.946	-0.0545	-0.00496
4	1930	1930-05-15 00:00:00	16552722	3129	8.57	1.15	0.155	0.0180
5	1940	1940-03-06 00:00:00	19653552	3583	9.82	1.19	0.187	0.0191
6	1950	1950-06-06 00:00:00	25791017	3744	10.3	1.31	0.312	0.0304
7	1960	1960-06-08 00:00:00	34923129	3655	10.0	1.35	0.354	0.0354
8	1970	1970-01-28 00:00:00	48225238	3521	9.65	1.38	0.381	0.0395
9	1980	1980-06-04 00:00:00	66846833	3780	10.4	1.39	0.386	0.0373

```

10 1990 1990-03-12 00:00:00 81249645 3568 9.78 1.22 0.215 0.0220
11 2000 2000-02-14 00:00:00 97483412 3626 9.93 1.20 0.200 0.0201
12 2010 2010-06-12 00:00:00 112336538 3771 10.3 1.15 0.152 0.0147
13 2020 2020-03-15 00:00:00 126014024 3564 9.76 1.12 0.122 0.0125
# 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: 39 x 3
   ano tipo crecimiento
  <dbl> <chr>         <dbl>
1  1900 ra           NA
2  1900 rg           NA
3  1900 r            NA
4  1910 ra          0.0114
5  1910 rg          0.0109
6  1910 r           0.0108
7  1921 ra        -0.00496
8  1921 rg        -0.00508
9  1921 r        -0.00510
10 1930 ra          0.0180
# i 29 more rows

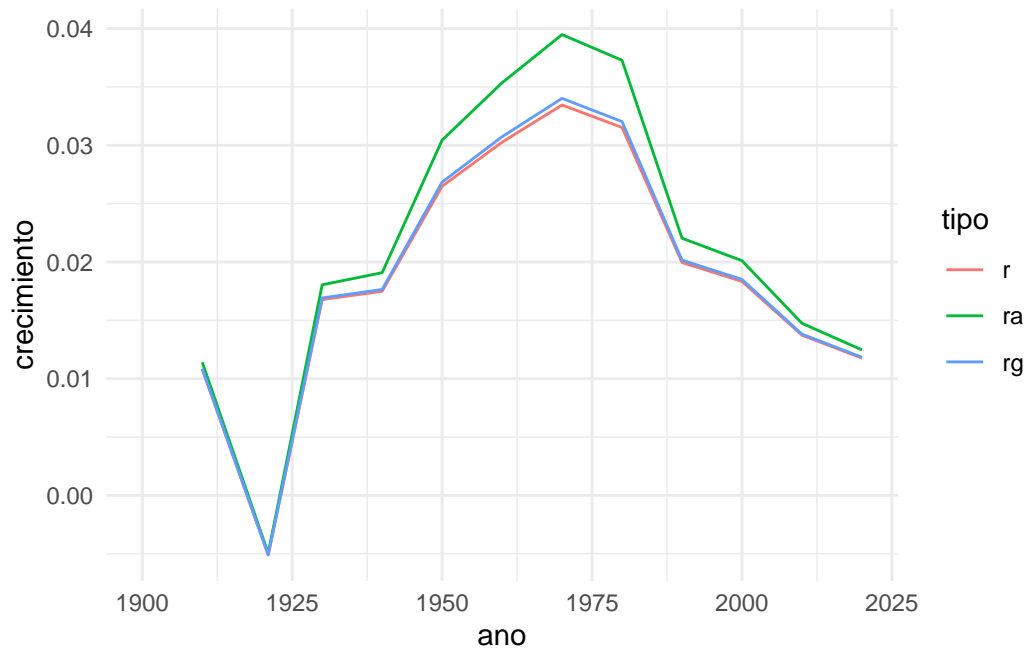
```

```

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$ano==2000,]$fecha)))
n
```

Time difference of 3425 days

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

```
[1] 9.383562
```

```
censos[censos$ano==2010, "poblacion"]
```

```
# A tibble: 1 x 1
  poblacion
    <dbl>
1 112336538
```

```
censos[censos$ano==2010, "r"]
```

```
# A tibble: 1 x 1
      r
    <dbl>
1 0.0137
```

```
ptn<- censos[censos$ano==2010,"poblacion"]*exp(n*censos[censos$ano==2010, "r"])

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

```
[1] "127779228.863837"
```

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

}
```

```
pob_estim(fecha ="2008-01-07",
          ano = 2010)
```

```
poblacion
1 108651068
```



### 4.3.8 Tiempo

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

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

```
n_calc<-log(2*censos[censos$ano==2020,"poblacion"]/censos[censos$ano==2020,"poblacion"])/c
n_calc
```

```
poblacion
1 58.9078
```

#### 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 :)

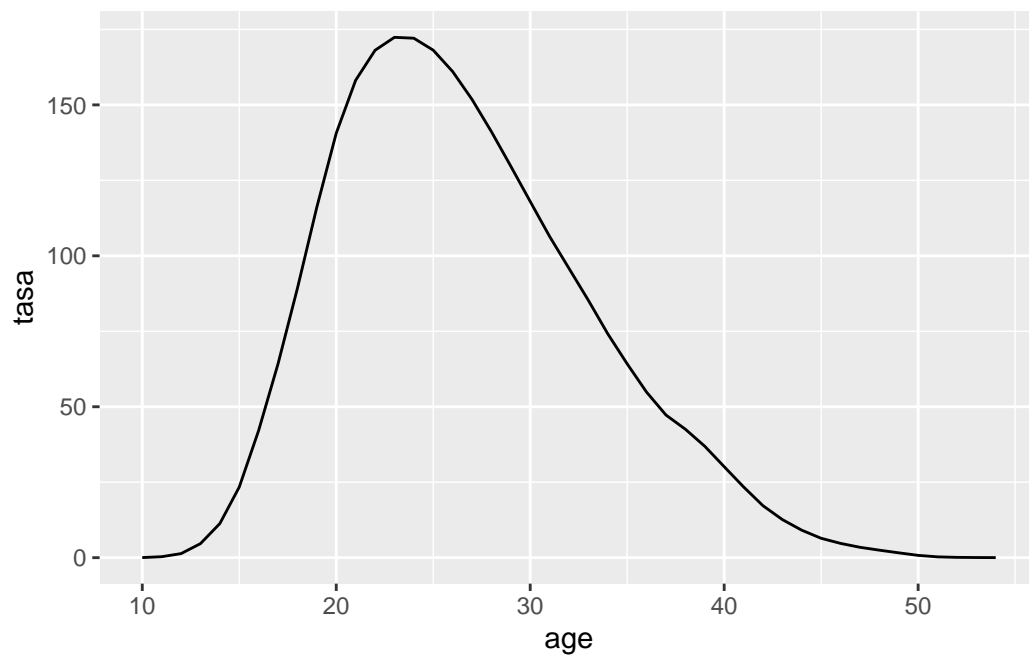
```
pasfr %<>%  
  mutate(tasa=pasfr*tfr$tfr/100*1000)
```

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

```
[1] 2.753664
```

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.

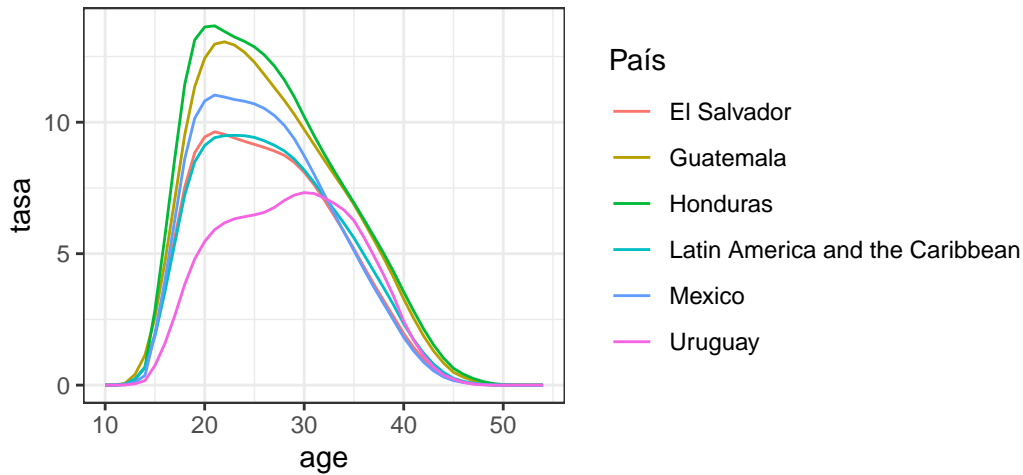
```
tasa_fec1dt<-percentASFR1dt %>%  
  dplyr::left_join(tfr1dt) %>%  
  mutate(tasa = pasfr * tfr)
```

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

Hoy podemos comparar intensidades y calendarios históricos!

```
tasa_fec1dt %>%  
  filter(country_code %in% c(484, 222, 320, 340, 858, 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>."  
  )
```

## Tasas de fecundidad en América Latina 2020



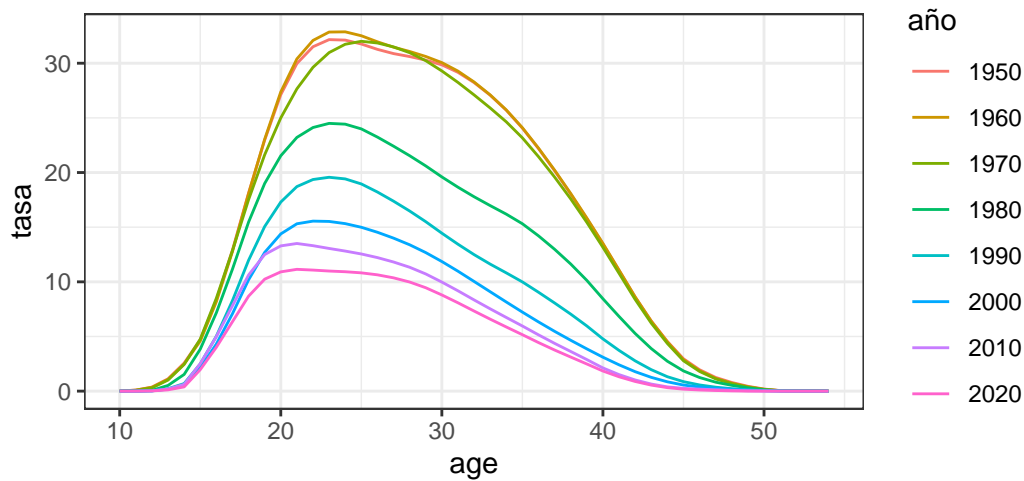
23). `_wpp2022`: World Population Prospects 2022\_. R package version 1.1-4, <<http://population.un.org/wpp>>.

Hoy comparamos tasas a lo largo del tiempo

```
tasa_fec1dt %>%
  filter(country_code==484) %>%
  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 = "México",
        color = "año",
        caption = "Division UNP (2024). _wpp2024: World Population Prospects 2024_. R package version 1.1-3, commit bac89293562767ff8510144974fd888e25bc7ee7, <https://github.com/PPgp/wpp2024>.")
```

## Tasas de fecundidad en América Latina

México

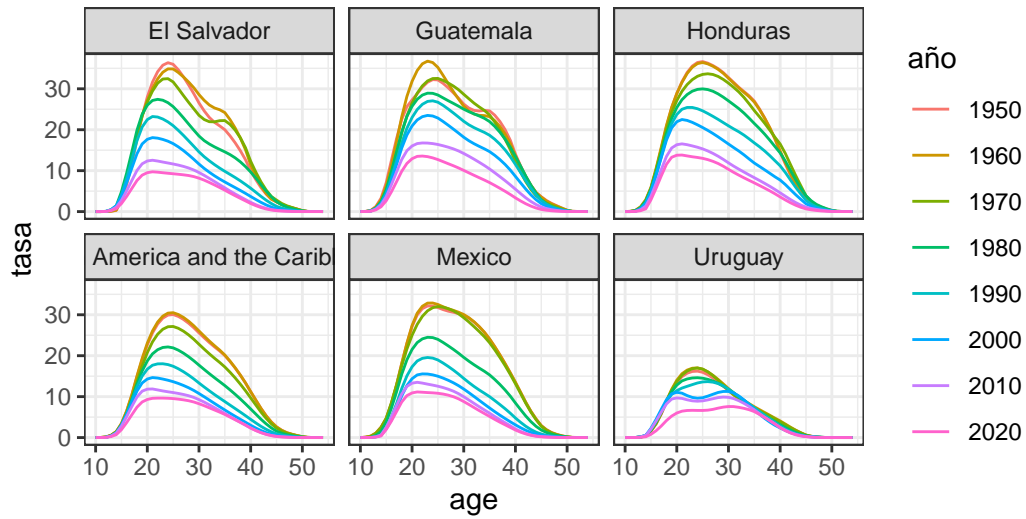


Division UNP (2024). `_wpp2024: World Population Prospects 2024_`. R package version 1.1-3, commit bac89293562767ff8510144974fd888e25bc7ee7, <<https://github.com/PPgp/wpp2024>>.

Un mix:

```
tasa_fec1dt %>%
  filter(country_code %in% c(484, 222, 320, 340, 858, 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() +
  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)
```

## Tasas de fecundidad en América Latina



Division UNP (2023). \_wpp2022: World Population Prospects 2022\_. R package version 1.1–4, <<http://population.un.org/wpp>>.

## 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.

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

Table 4.1: Data summary

Name	misc1dt
Number of rows	21978
Number of columns	11
Key	NULL
Column type frequency:	
character	1
numeric	10
Group variables	None

### Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
name	0	1	4	59	0	295	0

### Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
country_code	0	1	703.75	823.78	4.00	275.00	554.00	831.00	5505.00	
year	0	1	1986.50	21.36	1950.00	1968.00	1986.50	2005.00	2023.00	
births	0	1	4988.71	17524.29	0.03	13.27	145.98	843.95	146054.88	
cbr	0	1	28.84	13.13	3.52	16.93	27.80	40.94	58.63	
cdr	0	1	11.18	6.03	0.84	7.19	9.55	13.41	153.96	
deaths	0	1	1953.68	6599.80	0.01	4.06	62.49	329.23	69728.10	
PopChange	0	1	3009.97	10895.21	-	3.43	73.31	494.91	93667.18	
					6048.71					
growthrate	0	1	1.74	1.86	-	0.78	1.75	2.59	37.89	
					71.06					
cnmr	0	1	-0.27	15.63	-	-3.46	-0.31	2.04	357.14	
					709.90					
NatChangeRT	0	1	17.67	10.72	-	9.07	19.05	26.39	43.22	
					105.92					

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: 2,018,115

Columns: 18

\$ 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> 0.001309, 0.016938, 0.059008, 0.154829, 0.331357, 0.64295~
$ tfr       <dbl> 4.851944, 4.851944, 4.851944, 4.851944, 4.851944, 4.85194~
$ tasa      <dbl> 0.006351195, 0.082182227, 0.286303512, 0.751221638, 1.607~
$ popM      <dbl> 26927.18, 26448.37, 26657.29, 26352.70, 25849.18, 25185.7~
$ popF      <dbl> 25766.62, 25265.96, 25409.33, 25121.90, 24817.44, 24321.5~
$ pop       <dbl> 52693.80, 51714.33, 52066.62, 51474.60, 50666.63, 49507.2~
$ births    <dbl> 91823.94, 91823.94, 91823.94, 91823.94, 91823.94, 91823.9~
$ cbr       <dbl> 36.831, 36.831, 36.831, 36.831, 36.831, 36.831, 36.831, 3~
$ cdr       <dbl> 19.448, 19.448, 19.448, 19.448, 19.448, 19.448, 19.448, 1~
$ deaths    <dbl> 48486.89, 48486.89, 48486.89, 48486.89, 48486.89, 48486.8~
$ PopChange <dbl> 43337.69, 43337.69, 43337.69, 43337.69, 43337.69, 43337.6~
$ growthrate <dbl> 1.738, 1.738, 1.738, 1.738, 1.738, 1.738, 1.738, 1.738, 1~
$ cnmr      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ NatChangeRT <dbl> 17.383, 17.383, 17.383, 17.383, 17.383, 17.383, 17.383, 1~

```

Vamos a reconstruir los numeradores de nacimientos específicos

```

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.

```

mx1dt %>%
  glimpse()

```

Rows: 4,529,547

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> 0.16194393, 0.04901019, 0.02821689, 0.01817772, 0.0126645~

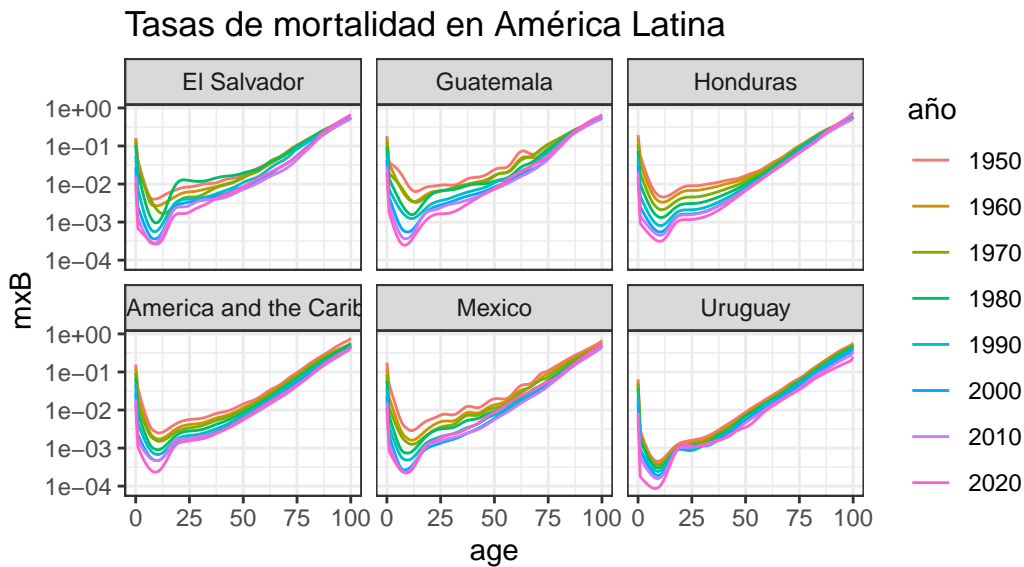
```



```
$ mxF      <dbl> 0.14354950, 0.04953383, 0.02897489, 0.01861074, 0.0129510~
$ mxB      <dbl> 0.15293607, 0.04926671, 0.02858799, 0.01838971, 0.0128048~
```

Grafiquemos:

```
mx1dt %>%
  filter(country_code %in% c(484, 222, 320, 340, 858, 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)
```



Division UNP (2023). \_wpp2022: World Population Prospects 2022\_. R  
package version 1.1-4, <http://population.un.org/wpp>.

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	0.16194393	0.14354950	0.15293607	91823.94
2:	900	World	1950	1	0.04901019	0.04953383	0.04926671	91823.94
3:	900	World	1950	2	0.02821689	0.02897489	0.02858799	91823.94
4:	900	World	1950	3	0.01817772	0.01861074	0.01838971	91823.94
5:	900	World	1950	4	0.01266450	0.01295105	0.01280488	91823.94
6:	900	World	1950	5	0.00908561	0.00929513	0.00918820	91823.94

	cbr	cdr	deaths	PopChange	growthrate	cnmr	NatChangeRT	popM
	<num>	<num>	<num>	<num>	<num>	<num>	<num>	<num>
1:	36.831	19.448	48486.89	43337.69	1.738	0	17.383	42423.73
2:	36.831	19.448	48486.89	43337.69	1.738	0	17.383	37788.70
3:	36.831	19.448	48486.89	43337.69	1.738	0	17.383	34783.84
4:	36.831	19.448	48486.89	43337.69	1.738	0	17.383	32924.84
5:	36.831	19.448	48486.89	43337.69	1.738	0	17.383	31297.27
6:	36.831	19.448	48486.89	43337.69	1.738	0	17.383	28363.97

	popF	pop	deaths_age
	<num>	<num>	<num>
1:	40762.38	83186.11	12722.1563
2:	36312.98	74101.68	3650.7460
3:	33382.79	68166.63	1948.7468
4:	31551.20	64476.04	1185.6957
5:	30028.38	61325.65	785.2676
6:	27249.36	55613.32	510.9863

#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=="Mexico") %>%
  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 intervalo
         right=F)) # indica si el intervalo irá abierto en la derecha, ponemos un
```

Vamos a volverlo quinquenal:

```
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	484	Mexico	1950	1209.	214.	[0,1)
2	484	Mexico	1950	3998.	118.	[1,5)
3	484	Mexico	1950	3970.	21.8	[5,10)
4	484	Mexico	1950	3206.	9.75	[10,15)
5	484	Mexico	1950	2775.	12.6	[15,20)
6	484	Mexico	1950	2462.	15.3	[20,25)
7	484	Mexico	1950	2077.	15.8	[25,30)
8	484	Mexico	1950	1652.	13.2	[30,35)
9	484	Mexico	1950	1470.	17.3	[35,40)
10	484	Mexico	1950	1296.	15.3	[40,45)

# i 34 more rows

Creamos las tasas por grupos

```
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 fácil de calcular:

```
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:

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

```
# A tibble: 2 x 2
  year  tbm
<int> <dbl>
1  1950 21.6
2  2020  8.46
```

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

```
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 21.6
2  2020  4.50

```

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 21.9
2  2020  6.48

```

¡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 disimilitud 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 0.00227 -0.0154
```

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

```
# A tibble: 2 x 2  
  year    tbm  
  <int> <dbl>  
1  1950 0.0216  
2  2020 0.00846
```

#### 4.7.1 Momento de práctica

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

## 5 Migración y flujos

### 5.1 Paquetes

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

Cargando paquete requerido: pacman

```
pacman::p_load(tidyverse,  
               readxl,  
               writexl,  
               janitor,  
               remotes,  
               magrittr,  
               collapse,  
               fmsb,  
               migest, # ojo  
               countrycode, # ojo  
               wpp2024)
```

### 5.2 Datos

```
# de wpp2024  
  
data("migration1dt")  
data("misc1dt")  
data("popAge1dt")
```

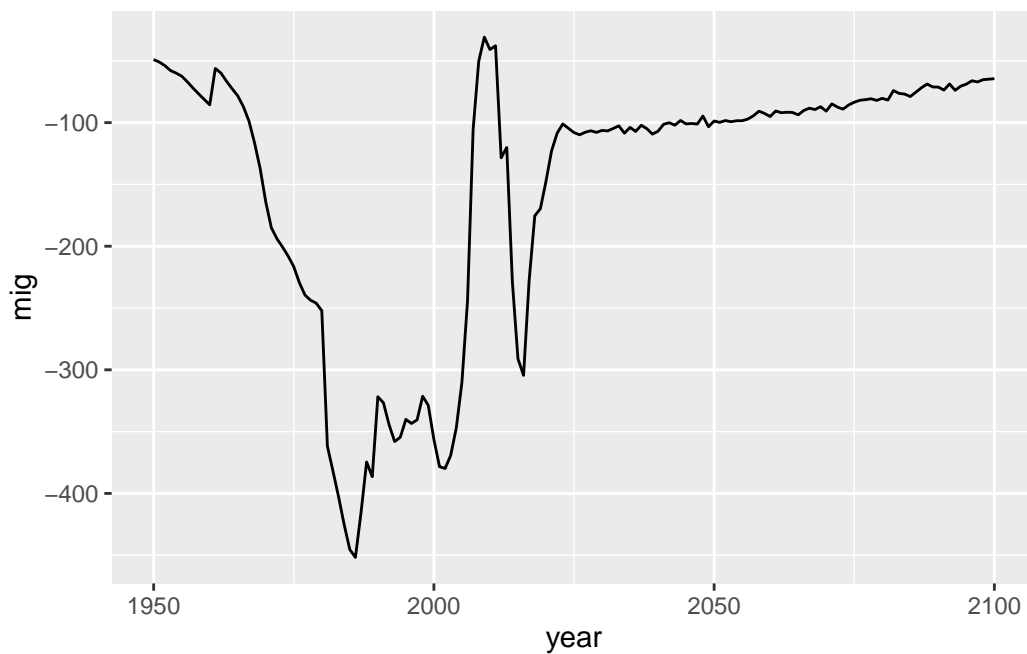


## 5.3 Migración

### 5.3.1 {wpp2024}

El conjunto de datos migration de wpp2024 nos da la migración neta

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



## 5.4 Flujos bilaterales

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 código
                                     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 código
                                     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()

```

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

```

pb <- d %>%
  filter(year0 == 2015) %>% # seleccionamos un año específico
  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

```

```

migest::mig_chord(x = pb) # objeto

```

```

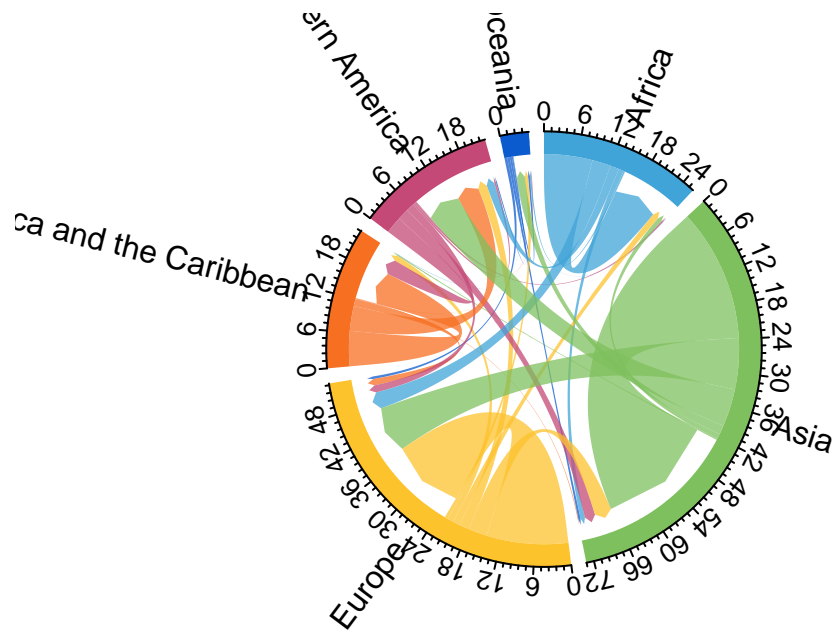
d %>%

```

```

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

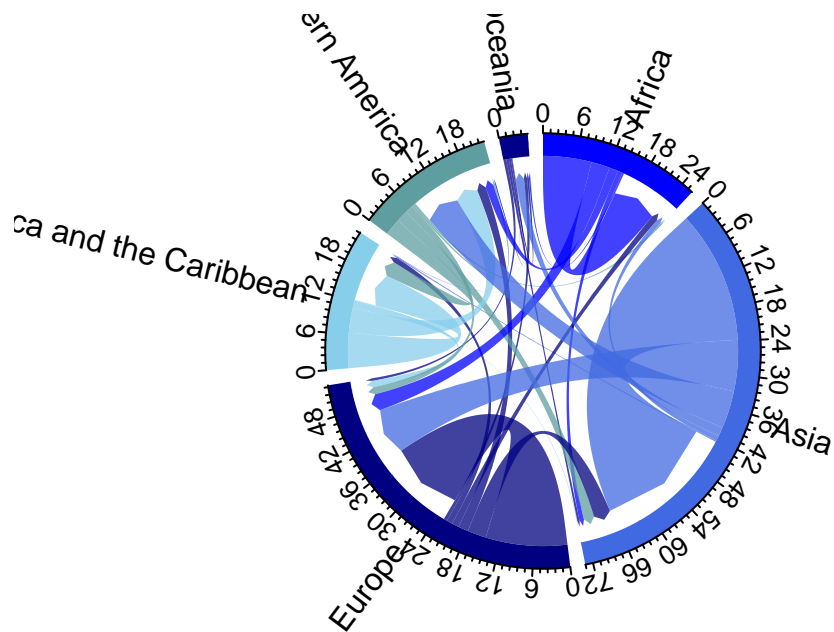
```



```

d %>%
  filter(year0 == 2015) %>% # seleccionamos un año específico
  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", "darkblue"),
)

```



Vamos a preparar los datos de los tabulados de los censos, para la migración interna hace 5 años. El archivo de tabulados lo descargas acá en formato desde INEGI [https://www.inegi.org.mx/contenidos/programas/ccpv/2020/tabulados/cpv2020\\_b\\_eum\\_04\\_migracion.xlsx](https://www.inegi.org.mx/contenidos/programas/ccpv/2020/tabulados/cpv2020_b_eum_04_migracion.xlsx)

Trabajaremos específicamente con la tabla 4 que nos brinda la información según residencia en marzo de 2015 y la población residencia actual

```
migra <- readxl::read_excel("datos/cpv2020_b_eum_04_migracion.xlsx", sheet = "04", skip =
  clean_names() %>% na.omit() %>%
  rename(dest0=x1) %>% # residencia actual es el destino
  rename(filtro=x2) %>%
  rename(orig0=x3) %>%
  rename(flujo=x4)
```

New names:

```
* `` -> `...1`
* `` -> `...2`
* `` -> `...3`
* `` -> `...4`
```

Para nombrar nuestro flujos, tenemos en este archivo los datos de las entidades con su código iso:

```
iso <- read_excel("datos/iso.xlsx", sheet = "Hoja1") %>% select(-entidad)
```

Vamos a construir flujos bilaterales, es decir, que solo hayan migrantes y en términos de entidades por lo que modificamos estos datos

```
migra %>%
  filter(!dest0=="Estados Unidos Mexicanos") %>%
  filter(filtro=="En otra entidad") %>%
  filter(!orig0=="Total")%>%
  mutate(cve_geo_orig=parse_number(orig0)) %>%
  mutate(cve_geo_dest=parse_number(dest0)) -> bilateral
```

Para graficarlos mejor en el diagrama, vamos utilizar los codigos iso de tres letras para los estados:

```
bilateral %<>%
  left_join(iso, join_by(cve_geo_dest==cve_geo)) %>%
  rename(dest=iso) %>%
  left_join(iso, join_by(cve_geo_orig==cve_geo)) %>%
  rename(orig=iso)
```

Vamos a crear un objeto sólo con tres variables

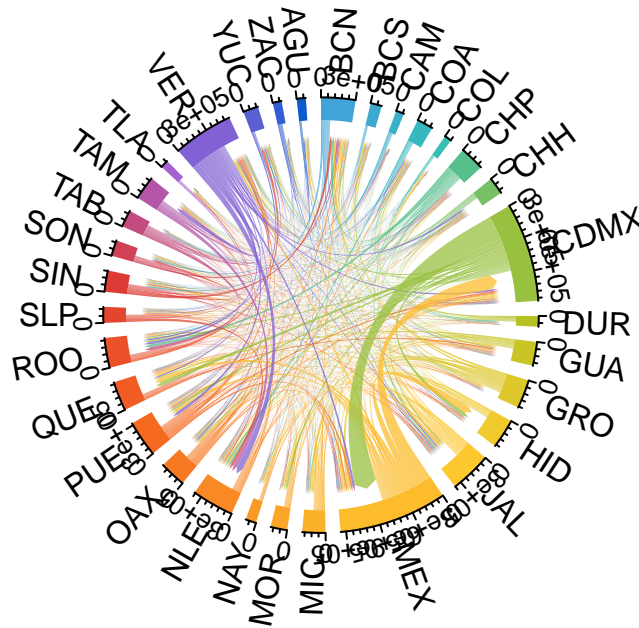
```
bilateral %>%
  select(orig, dest, flujo) # necesitamos un objeto con estas tres variables
```

```
# A tibble: 992 x 3
  orig dest flujo
  <chr> <chr> <dbl>
1 BCN AGU 806
2 BCS AGU 175
3 CAM AGU 138
4 COA AGU 960
5 COL AGU 418
6 CHP AGU 481
7 CHH AGU 999
8 CDMX AGU 6426
9 DUR AGU 796
10 GUA AGU 3245
# i 982 more rows
```

```

bilateral %>%
  mutate(flow=flujo/1000) %>% # para tener los flujos en miles
  select(orig, dest, flujo) %>%
  migest::mig_chord()

```

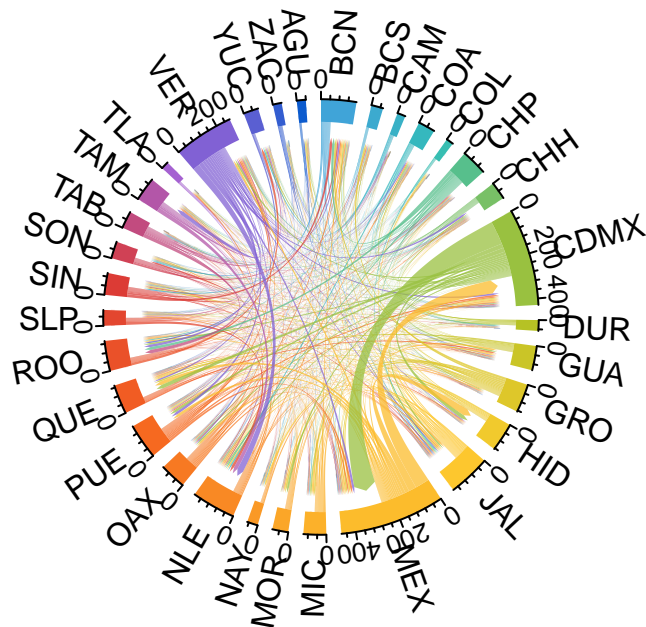


Veamos si hay diferencia entre sexos

```

bilateral %>%
  mutate(flow =hombres/1000) %>% # pasamos el flujo cerrado a miles
  select(orig, dest, flow) %>%
  mig_chord()

```



```

bilateral %>%
  mutate(flow =mujeres/1000) %>% # pasamos el flujo cerrado a miles
  select(orig, dest, flow) %>%
  mig_chord()

```



