



FACULTAD DE CIENCIAS SOCIALES E HISTORIA

Clase reproducibilidad y replicabilidad.
Informes en .pdf

Amaru Simón Agüero Jiménez

2024-12-10

Tabla de contenidos

1 Paquetes R y LaTeX	2
1.1 Paquetes para Tablas y Visualización de Datos	3
1.2 Paquetes para Análisis y Manejo de Datos	3
1.3 Paquetes para Visualización Gráfica y Mapas	3
1.4 Paquete para Expresiones Matemáticas	3
2 Introducción a Quarto	4
2.1 Visualización en HTML	4
2.2 YAML	4
2.3 Opciones en el Chunk	4
3 Texto en Rmarkdown	5
3.1 Encabezados	5
3.1.1 Encabezado de nivel 3	5
3.2 Énfasis de Texto	5
3.3 Listas	5
3.3.1 Listas No Ordenadas	5
3.4 Listas Ordenadas	6
3.5 Tablas	6

3.6	Citas	6
3.6.1	Citas en bloque	6
3.6.2	Citas en Bibtex	6
3.7	Nota al pie	6
3.8	Texto en linea	7
3.9	Subíndice y Superíndice	7
3.10	LaTeX	7
3.10.1	Símbolos Matemáticos y Operadores	7
3.10.2	Matrices y Sistemas de Ecuaciones	7
3.10.3	Funciones Matematicas	7
3.10.4	Funciones Alineadas	8
4	Imágenes	8
4.1	Imágenes desde la web	8
4.2	Imágenes desde R	8
4.2.1	Plot	8
4.2.2	Ggplot	10
4.2.3	Ggplot y LaTeX	11
4.2.4	Mapas con maps	12
4.2.5	Mapas con sf	13
4.3	Imágenes desde Python	15
4.3.1	Citar figuras en el texto	17
5	Tablas	18
5.1	Table1	18
5.2	Kable	19

1 Paquetes R y LaTeX

```
install_and_load <- function(package) {
  if (!require(package, character.only = TRUE)) {
    utils::install.packages(package)
    library(package, character.only = TRUE)
  }
}

# List of packages to be installed and loaded
packages <- c("knitr",
              "kableExtra",
              "DT",
              "table1",
```

```
"tinytex",
"reticulate",
"gapminder",
"latex2exp",
"tidyverse",
"maps",
"sf",
"tmap")

# Apply the function to each package
invisible(capture.output(sapply(packages, install_and_load)))

# Install LaTeX
# tinytex::install_tinytex()
# tinytex::tlmgr_install(c("booktabs", "xcolor", "array"))
```

A continuación, se explica la funcionalidad de cada paquete contenido en la lista.

1.1 Paquetes para Tablas y Visualización de Datos

- **kableExtra**: Extiende la funcionalidad de `knitr::kable()`, permitiendo crear tablas avanzadas y personalizadas para documentos HTML, PDF y LaTeX.
- **DT**: Proporciona una interfaz para crear tablas interactivas en R Markdown, con opciones de búsqueda, clasificación y exportación.
- **table1**: Genera tablas de resumen descriptivo para reportar estadísticas de manera clara y profesional en investigaciones científicas.

1.2 Paquetes para Análisis y Manejo de Datos

- **reticulate**: Permite la integración entre R y Python, facilitando la ejecución de scripts y funciones de Python dentro de un entorno R.
- **gapminder**: Contiene un conjunto de datos históricos de desarrollo humano, ideal para ejercicios de análisis y visualización de datos.
- **tidyverse**: Un conjunto de paquetes esenciales para la manipulación, limpieza y visualización de datos en R, incluyendo `dplyr`, `ggplot2`, `tidyr`, `readr`, entre otros.

1.3 Paquetes para Visualización Gráfica y Mapas

- **maps**: Proporciona mapas base para crear visualizaciones geográficas simples en R.
- **sf**: Maneja datos espaciales, permitiendo operaciones GIS como transformación, consulta y análisis espacial.
- **tmap**: Crea mapas temáticos de alta calidad y visualizaciones interactivas basadas en datos espaciales.

1.4 Paquete para Expresiones Matemáticas

- **latex2exp**: Permite incluir expresiones matemáticas en gráficos de R utilizando sintaxis de LaTeX.

2 Introducción a Quarto

Quarto es una herramienta de publicación y creación de documentos científicos, informes y presentaciones interactivas. Combina la escritura en formato Markdown con lenguajes de programación como R, Python y Julia para crear documentos reproducibles y dinámicos. Es similar a R Markdown, pero más flexible y adaptable para proyectos multilingües. Una ayuda memoria se encuentra [aquí](#).

Quarto y HTML están estrechamente relacionados, ya que Quarto puede generar documentos HTML interactivos y personalizables a partir de archivos `.qmd` escritos en Markdown y combinados con lenguajes de programación como R, Python, Julia y JavaScript.

Por último Quarto le permite entrelazar contenido y código ejecutable en un documento terminado. Para obtener más información sobre Quarto, consulte [esto](#).

2.1 Visualización en HTML

Un archivo CSS (Cascading Style Sheets o Hojas de Estilo en Cascada) es un componente esencial en el desarrollo web que define la presentación y el diseño visual de una página o sitio web. Mientras que HTML estructura el contenido, CSS se encarga de su apariencia, controlando aspectos como colores, tipografías, márgenes y disposición de los elementos.

Aparte Quarto ofrece una variedad de temas predefinidos para personalizar la apariencia de tus documentos HTML. Por defecto, utiliza Bootstrap 5, pero puedes seleccionar entre 25 temas adicionales del proyecto Bootswatch. Algunos de los temas [disponibles](#).

Estos temas se pueden ajustar o desde un archivo `.css` o `.scss` o desde un chunk css, ejemplo:

2.2 YAML

YAML (YAML Ain't Markup Language) es un formato de configuración utilizado en Quarto y R Markdown para definir metadatos y opciones del documento. El bloque YAML aparece al comienzo del archivo y permite configurar el título, autor, fecha, formato de salida y muchas otras opciones.

2.3 Opciones en el Chunk

Un chunk es un bloque de código que se incluye dentro de un documento de Quarto (`.qmd`) o R Markdown (`.Rmd`). Estos bloques permiten escribir y ejecutar código en lenguajes como R, Python, Julia o JavaScript. Los resultados del código, como texto, gráficos y tablas, se incluyen automáticamente en el documento renderizado.

Puede agregar opciones al YAML o al código chunk como este

```
2 * 2
```

[1] 4

- `eval`: Determina si el código del *chunk* se evalúa.
Valores: `true` (por defecto) o `false`.
Si se establece en `false`, el código no se ejecuta, pero puede mostrarse si `echo` es `true`.
- `echo`: Controla la visualización del código fuente en el documento final.
Valores: `true` (por defecto), `false` o `"fenced"`.
Con `echo: fenced`, el código se muestra con delimitadores de bloque, útil para tutoriales.

- **include:** Decide si se incluye alguna salida del *chunk* en el documento.
Valores: true (por defecto) o false.
Si es false, ni el código ni los resultados se muestran, aunque el código se ejecute.
- **warning:** Indica si las advertencias generadas durante la ejecución se muestran.
Valores: true (por defecto) o false.
- **error:** Especifica si los errores durante la ejecución se incluyen en la salida.
Valores: true o false (por defecto).
Si es true, los errores no detienen el procesamiento del documento.
- **message:** Controla la visualización de mensajes generados durante la ejecución.
Valores: true (por defecto) o false.
- **fig-width y fig-height:** Definen el ancho y alto de las figuras generadas, en pulgadas.
Por defecto: 7x5 para documentos HTML.
- **fig-cap:** Establece el título o leyenda de la figura generada.
- **fig-alt:** Proporciona texto alternativo para figuras, mejorando la accesibilidad.
- **output:** Controla la inclusión de resultados de la ejecución.
Valores: true (por defecto), false o asis (para incluir salida en bruto sin envolturas adicionales).

3 Texto en Rmarkdown

3.1 Encabezados

Los encabezados se crean usando el símbolo #. Cuantos más # uses, menor será el nivel del encabezado.

3.1.1 Encabezado de nivel 3

3.1.1.1 Encabezado de nivel 4

3.1.1.1.1 Encabezado de nivel 5

Encabezado de nivel 6

3.2 Énfasis de Texto

Negrita → **Texto en negrita**

Itálica → *Texto en itálica*

Negrita e Itálica → ***Texto en negrita e itálica***

3.3 Listas

3.3.1 Listas No Ordenadas

- Elemento 1
- Elemento 2
 - Sub-elemento 2.1
 - Sub-elemento 2.2

3.4 Listas Ordenadas

1. Primer elemento
2. Segundo elemento
 1. Sub-elemento 2.1
 2. Sub-elemento 2.2

3.5 Tablas

Columna 1	Columna 2	Columna 3
Valor 1	Valor 2	Valor 3
Valor A	Valor B	Valor C

3.6 Citas

3.6.1 Citas en bloque

Esta es una cita en bloque. Puede abarcar varias líneas.

3.6.2 Citas en Bibtex

En R Markdown, puedes citar referencias usando BibTeX en combinación con un archivo de bibliografía (.bib). para otros formatos de la referencia hay que descargar un archivo (.csl) del siguiente [link](#).

Ejemplo:

3.6.2.1 Direct standardization rates (DSR).

For the purpose of calculating the hospitalization rate for each C51-C58 ICD code, the direct standardization by age group (stratum) method was employed^{1 2}. The total population of Chilean females by year (2010-2023) is obtained from World Bank Open Data.³. This method utilizes a standard population from the 2017 Chilean Census as a reference⁴. The formula used to calculate DSR is as follows:

$$DSR = \frac{\sum_{i=1}^k N_i \frac{y_i}{n_i}}{\sum_{i=1}^k N_i}$$

where

N_i is the reference population size for the i th stratum.

y_i is the number of events observed in population in the i th stratum.

n_i is the person-time for the i th stratum of the study population.

3.7 Nota al pie

Este es un ejemplo de texto con una nota al pie ¹.

¹Aquí se incluye la nota al pie.

3.8 Texto en linea

El resultado de $\sqrt{4}$ es `rsqrt(16)`.

```
# Definimos una función de ejemplo
mi_funcion <- function(x, y) {
  suma <- x + y
  return(suma)
}

# Ejecutamos la función
mi_funcion(3, 5)
```

[1] 8

La suma de 3 y 5 es 8.

3.9 Subíndice y Superíndice

Texto con subíndice H₂O y superíndice E².

3.10 LaTeX

En R Markdown, puedes usar LaTeX para incluir ecuaciones matemáticas y símbolos especiales, lo que te permite agregar fórmulas complejas en documentos científicos, informes y presentaciones.

3.10.1 Símbolos Matemáticos y Operadores

- Fracción: $\frac{a}{b}$
- Suma: $\sum_{i=1}^n i$
- Producto: $\prod_{i=1}^n i$
- Raíz Cuadrada: $\sqrt{x+1}$
- Potencia: x^2 o x^{n+1}
- Subíndice: x_1, y_2

3.10.2 Matrices y Sistemas de Ecuaciones

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

3.10.3 Funciones Matematicas

- Logaritmo: $\log(x)$
- Seno y Coseno: $\sin(\theta), \cos(\theta)$
- Límite: $\lim_{x \rightarrow \infty} f(x)$
- Derivada: $\frac{d}{dx} f(x)$
- Integral: $\int_0^{\infty} e^{-x} dx$

3.10.4 Funciones Alineadas

$$\begin{aligned}
 \int_{-\infty}^{\infty} f(x) dx &= (2\pi)^{-1/2} \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}x^2\right) dx \\
 &= (2\pi)^{-1/2} 2 \int_0^{\infty} \exp\left(-\frac{1}{2}x^2\right) dx \quad (\text{since the integrand is even}) \\
 &= (2\pi)^{-1/2} \left(\int_0^{\infty} \exp\left(-\frac{1}{2}x^2\right) dx \int_0^{\infty} \exp\left(-\frac{1}{2}y^2\right) dy \right)^{1/2} \\
 &= (2\pi)^{-1/2} \left(\int_0^{\infty} \int_0^{\infty} \exp\left(-\frac{1}{2}(x^2 + y^2)\right) dx dy \right)^{1/2} \\
 &= (2\pi)^{-1/2} \left(\int_0^{\infty} \int_0^{2\pi} \exp\left(-\frac{1}{2}r^2\right) r d\theta dr \right)^{1/2} \\
 &= (2\pi)^{-1/2} \left(2\pi \int_0^{\infty} \exp\left(-\frac{1}{2}r^2\right) r dr \right)^{1/2} \\
 &= (2\pi)^{-1/2} \left(\int_0^{\infty} \exp\left(-\frac{1}{2}r^2\right) r dr \right)^{1/2} \\
 &= (2\pi)^{-1/2} \left(\int_0^{\infty} \frac{1}{1+s^2} \exp\left(-\frac{1}{2}(1+s^2)\right) ds \right)^{1/2} \\
 &= (2\pi)^{-1/2} \left(\int_0^{\infty} \frac{1}{1+s^2} ds \right)^{1/2} \\
 &= (2\pi)^{-1/2} \left(2 \arctan(s) \Big|_0^{\infty} \right)^{1/2} \\
 &= (2\pi)^{-1/2} \left(2 \cdot \frac{\pi}{2} \right)^{1/2} \\
 &= 2^{-1/2} \pi^{-1/2} \cdot \sqrt{2\pi} \\
 &= 1.
 \end{aligned}$$

4 Imágenes

4.1 Imágenes desde la web

4.2 Imágenes desde R

4.2.1 Plot

```
# Filtrar los datos para Chile y Argentina
datos_filtrados <- gapminder %>%
  filter(country %in% c("Chile", "Argentina"))
```

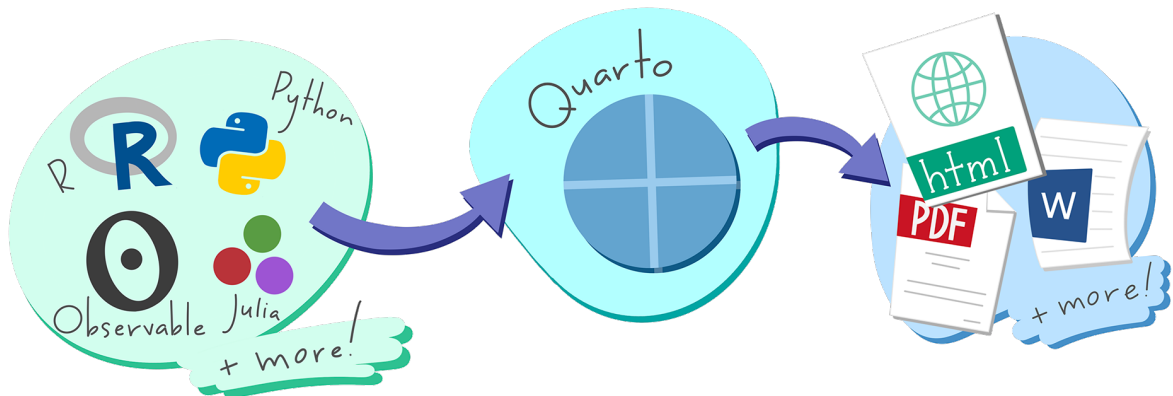



Figura 1: Logo de Quarto

```
# Definir el rango de años y la esperanza de vida para el eje Y
years <- unique(datos_filtrados$year)
ylim <- range(datos_filtrados$lifeExp)

# Crear el gráfico vacío
plot(NULL, xlim = range(years), ylim = ylim,
      xlab = "Año", ylab = "Esperanza de Vida (años)",
      main = "Evolución de la Esperanza de Vida en Chile y Argentina")

# Añadir líneas para cada país
countries <- c("Chile", "Argentina")
colors <- c("blue", "red")

for (i in 1:2) {
  country_data <- datos_filtrados %>% filter(country == countries[i])
  lines(country_data$year, country_data$lifeExp, type = "o", col = colors[i], pch = 16)
}

# Añadir una leyenda
legend("bottomright", legend = countries, col = colors, pch = 16, lty = 1)
```

Evolución de la Esperanza de Vida en Chile y Argentina

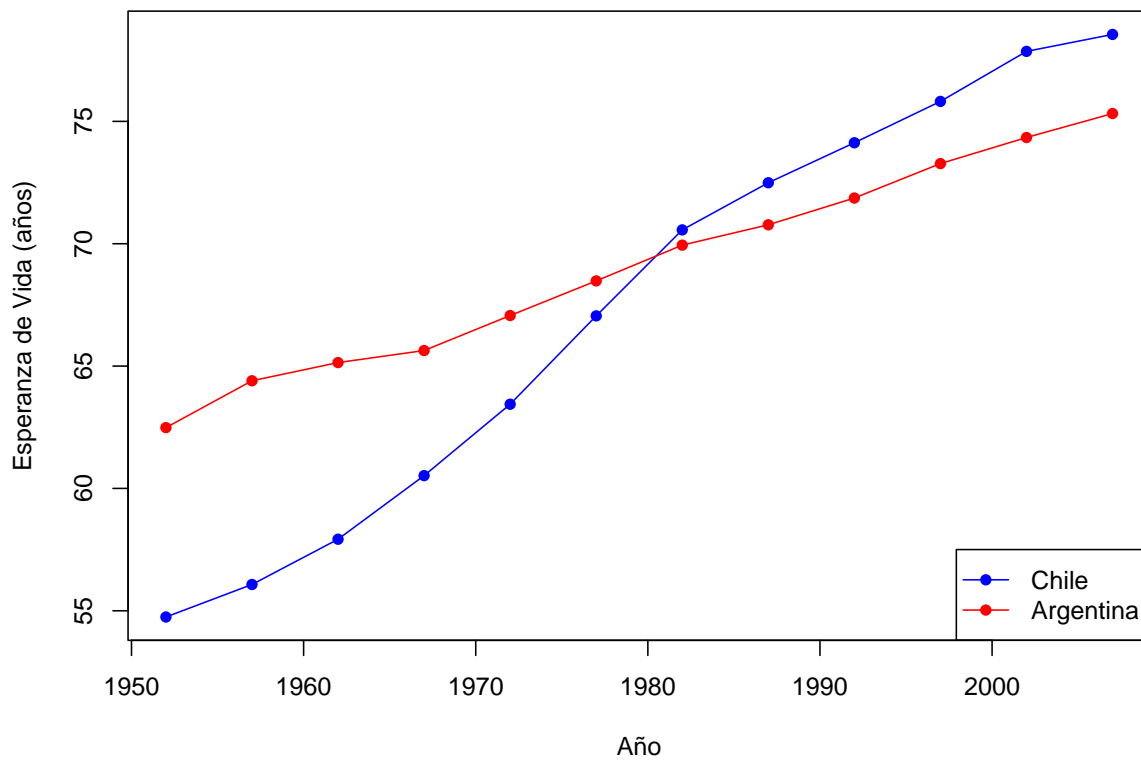


Figura 2: Gráfico de líneas en plot

4.2.2 Ggplot

```
# Crear el gráfico de líneas
ggplot(datos_filtrados, aes(x = year, y = lifeExp, color = country)) +
  geom_line(size = 1) +
  labs(
    title = "Evolución de la Esperanza de Vida en Chile y Argentina",
    x = "Año",
    y = "Esperanza de Vida (años)",
    color = "País"
  ) +
  theme_minimal()
```

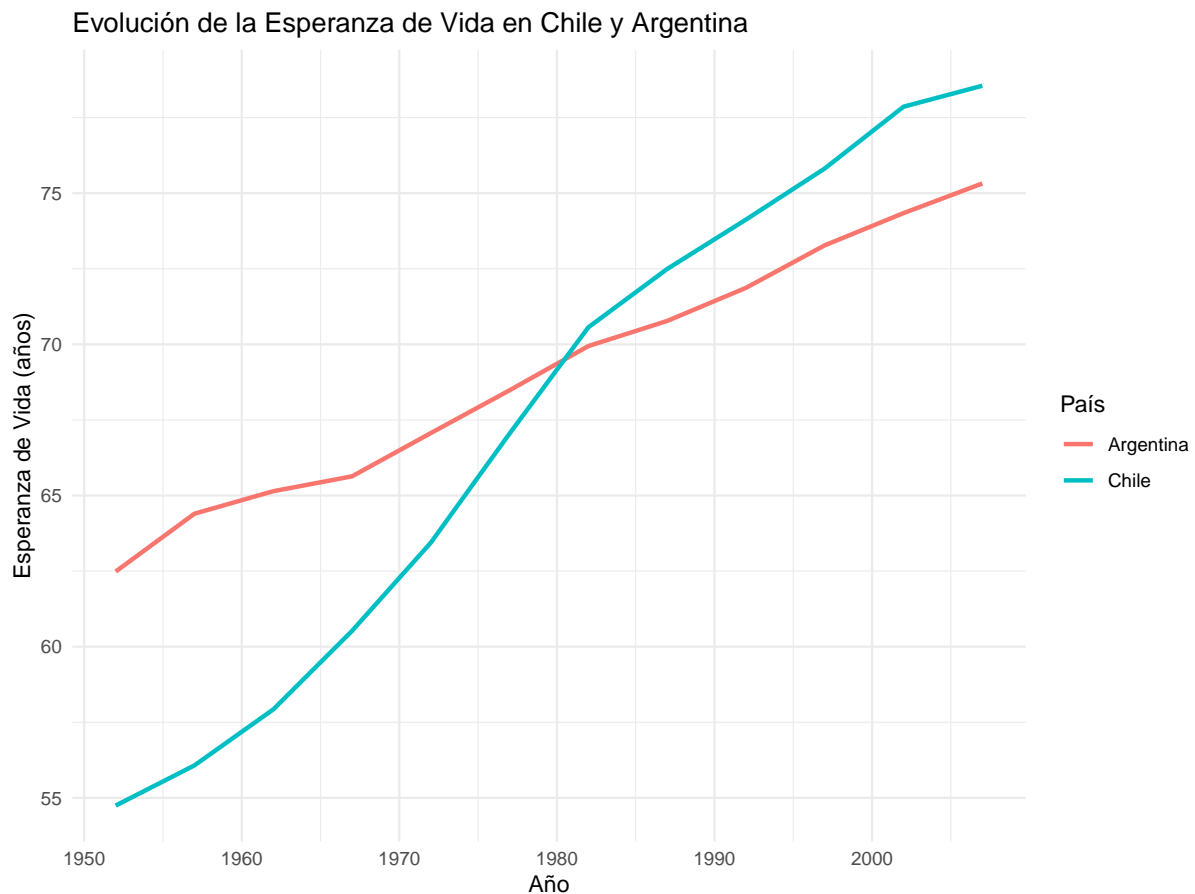


Figura 3: Gráfico de líneas en ggplot

4.2.3 Ggplot y LaTeX

```
#Distribucion Normal
x <- seq(from = -5, to = 5, by = 0.01)
norm_dat <- data.frame(x, pdf = dnorm(x,0,1))
quant <- c(-Inf, -1.96, 1.96, Inf)
norm_dat$quant <- factor(findInterval(norm_dat$x,quant))

#grafico de funcion de densidad de probabilidades de la normal
ggplot(norm_dat, aes(x,pdf)) + geom_line()+ geom_ribbon(aes(ymin=0, ymax=pdf, fill=quant), alpha=0.1)
  scale_x_continuous(breaks=seq(from = -5, to = 5, by = 1), expand = c(0, 0))+
  scale_y_continuous(limits = c(0,0.45), expand = c(0, 0))+
  geom_vline(xintercept=(1.96), size=1, color="red")+
  geom_vline(xintercept=(-1.96), size=1, color="red")+
  scale_fill_manual(values = c("red","grey","red"))+
  theme_bw() + theme(legend.position = "none")+
  ylab(latex2exp::TeX("$ = \\frac{1}{\\sigma\\sqrt{2\\pi}}\\exp\\left(-\\frac{1}{2}\\left(\\frac{x-\\mu}{\\sigma}\\right)^2\\right)$"))+
  xlab(latex2exp::TeX("$z_0$"))+
  annotate(geom = "text",
    label = "1,96",
    x = 1.96,
    y = 0.3,
```

```
angle = 90,
vjust = -1,
colour = "red",
size = 8)+
annotate(geom = "text",
label = "-1,96",
x = -1.96,
y = 0.3,
angle = 90,
vjust = -1,
colour = "red",
size = 8)
```

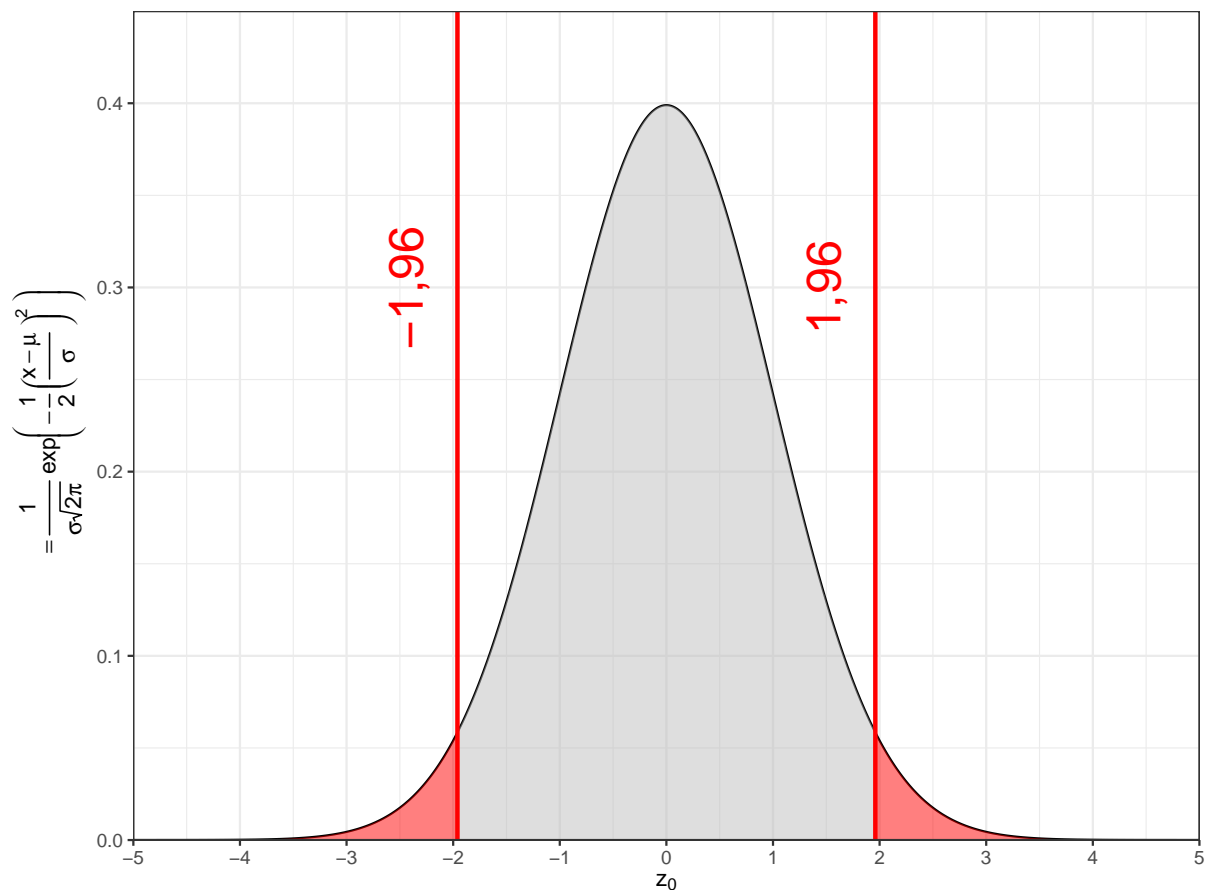


Figura 4: Distribución Normal

4.2.4 Mapas con maps

```
# Cargar el mapa de un país o región (ejemplo: Chile)
map_data <- map_data("world", region = "Chile")

# Crear el gráfico del mapa
ggplot(data = map_data) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = "lightblue", color = "black") +
```

```
coord_fixed(1.3) + # Mantiene las proporciones
labs(title = "Mapa de Chile") +
theme_minimal()
```

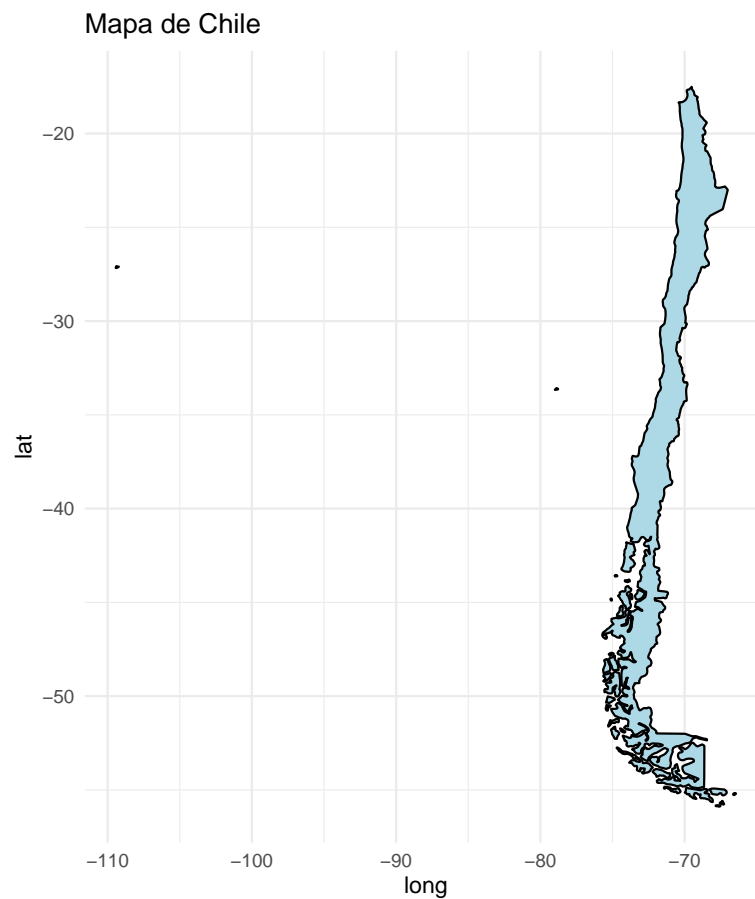


Figura 5: Chile

4.2.5 Mapas con sf

```
# Cargar datos de ejemplo del paquete sf
world <- st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)

# Graficar el mapa
ggplot(data = world) +
  geom_sf(fill = "lightblue", color = "black") +
  labs(title = "Mapa de Ejemplo (Datos de sf)") +
  theme_minimal()
```

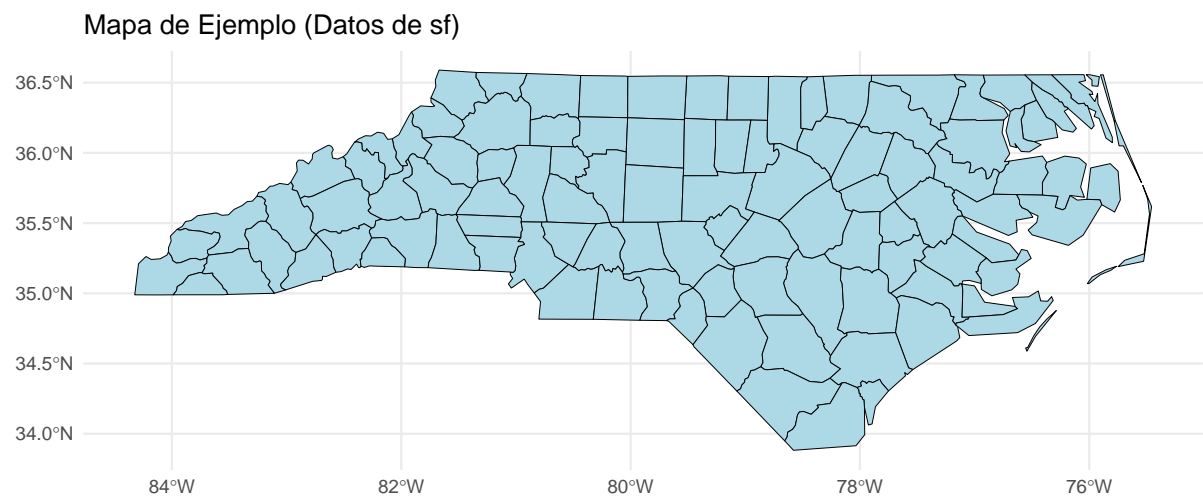


Figura 6: Carolina del Norte (Estados Unidos).

```
# Crear un mapa estático
data("World", package = "tmap")

# Cambiar a modo de mapa estático
tmap_mode("plot")

# Crear el mapa
map <- tm_shape(World) +
  tm_polygons("HPI", title = "Índice de Planeta Feliz (HPI)") +
  tm_layout(title = "Mapa Mundial")
map
```

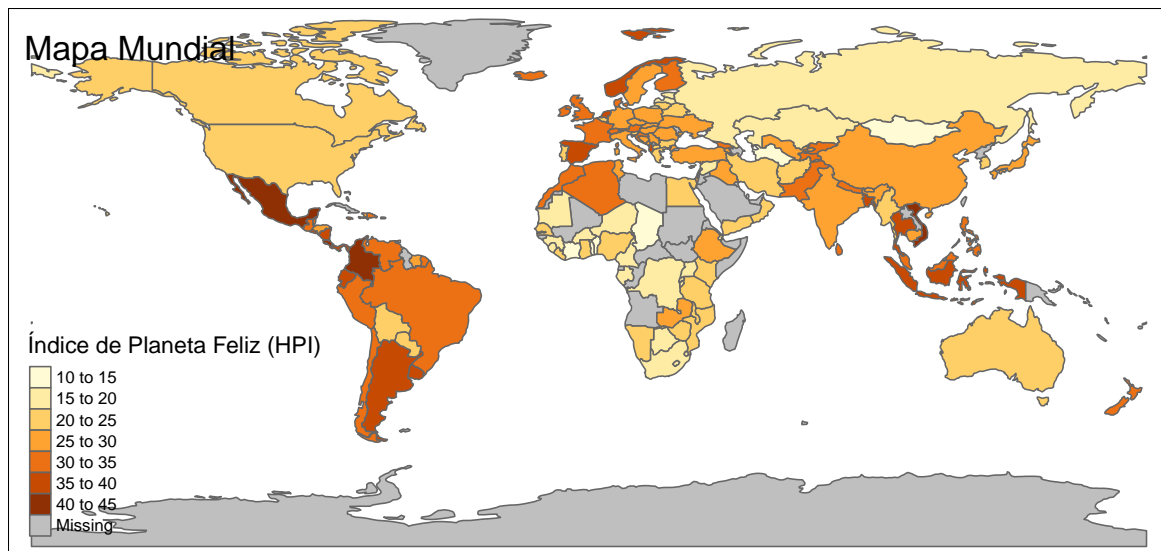


Figura 7: Mapa Mundial.

4.3 Imagenes desde Python

```
# Crea el entorno virtual
reticulate::virtualenv_create("mi_entorno_python")
```

```
virtualenv: mi_entorno_python
```

```
# Activa el entorno virtual
reticulate::use_virtualenv("mi_entorno_python", required = TRUE)

# Instala paquetes de Python en el entorno virtual
reticulate::virtualenv_install("mi_entorno_python",
                               packages = c("seaborn", "matplotlib",
                                             "scikit-learn", "pandas", "plotly"))
```

```
Using virtual environment "mi_entorno_python" ...
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
# Crear datos
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Generar la figura
plt.figure(figsize=(8, 6))
plt.plot(x, y, label="Seno(x)", color="blue")
plt.title("Gráfico generado con Python en Quarto")
plt.xlabel("Eje X")
plt.ylabel("Eje Y")
plt.legend()
plt.grid(True)

# Mostrar la figura
plt.show()
```

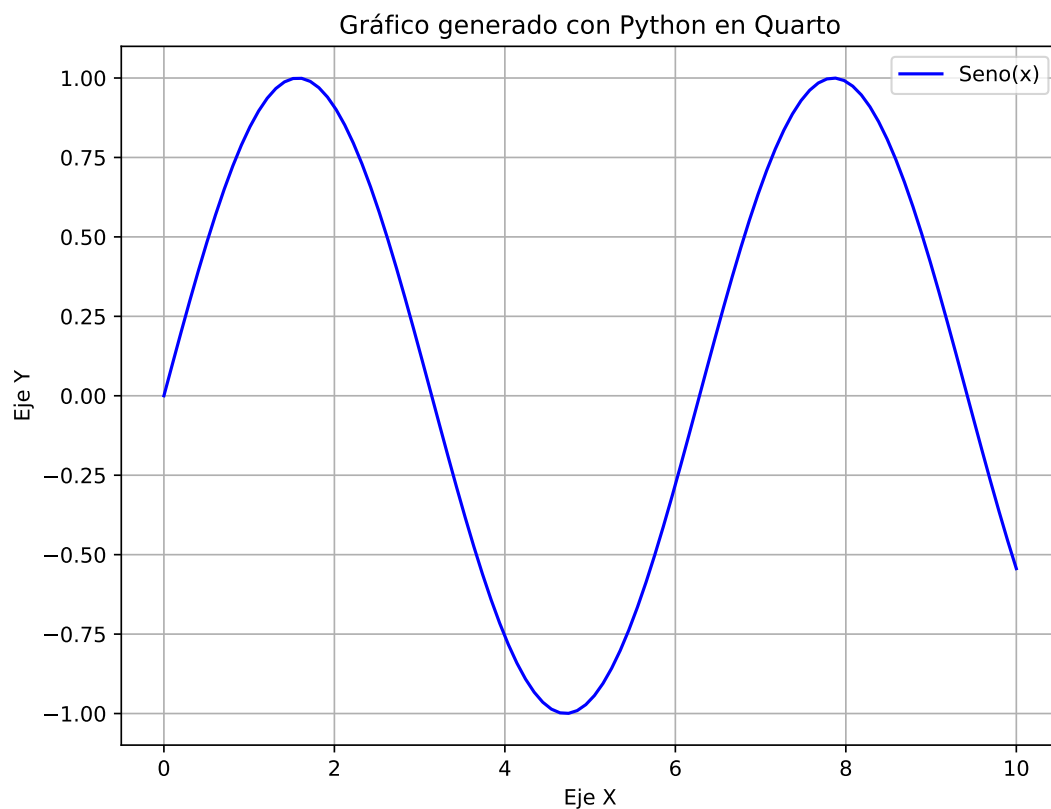


Figura 8: Grafico Python 1

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
import pandas as pd

# Cargar el dataset de ejemplo
```



```
data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['species'] = data.target

# Crear un gráfico con seaborn
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='species', y='sepal length (cm)', palette='viridis')

# Configurar el título
plt.title("Distribución de la longitud del sépalo por especie", fontsize=14)
plt.xlabel("Especie", fontsize=12)
plt.ylabel("Longitud del Sépalo (cm)", fontsize=12)

# Mostrar la figura
plt.show()
```

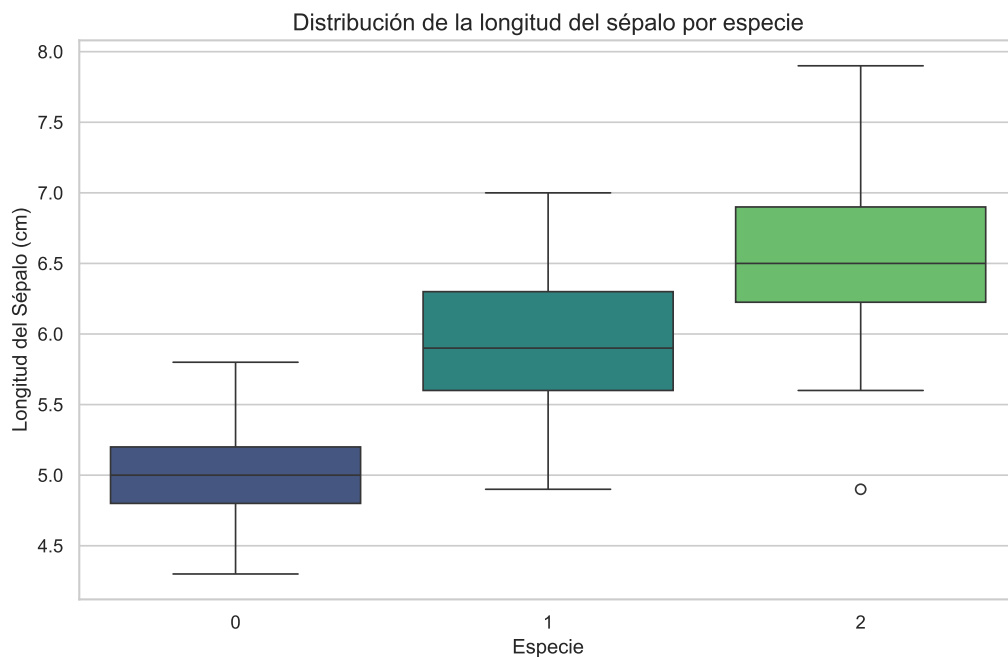


Figura 9: Grafico Python 2

4.3.1 Citar figuras en el texto

La Figura 2, Figura 3, la Figura 6 son...

5 Tablas

5.1 Table1

```
# Filtrar datos de Gapminder para 2007
data_gap <- gapminder %>% filter(year == 2007)

# Etiquetas descriptivas
label(data_gap$lifeExp) <- "Esperanza de Vida (años)"
label(data_gap$gdpPercap) <- "PIB per cápita (USD)"
label(data_gap$continent) <- "Continente"
label(data_gap$pop) <- "Población Total"
```

Tabla 2: Resumen de Indicadores Sociodemográficos por Continente (2007)

	Africa (N=52)	Americas (N=25)	Asia (N=33)	Europe (N=30)	Oceania (N=2)	Overall (N=142)
Esperanza de Vida (años)						
Mean (SD)	54.8 (9.63)	73.6 (4.44)	70.7 (7.96)	77.6 (2.98)	80.7 (0.729)	67.0 (12.1)
Median [Min, Max]	52.9 [39.6, 76.4]	72.9 [60.9, 80.7]	72.4 [43.8, 82.6]	78.6 [71.8, 81.8]	80.7 [80.2, 81.2]	71.9 [39.6, 82.6]
PIB per cápita (USD)						
Mean (SD)	3090 (3620)	11000 (9710)	12500 (14200)	25100 (11800)	29800 (6540)	11700 (12900)
Median [Min, Max]	1450 [278, 13200]	8950 [1200, 43000]	4470 [944, 47300]	28100 [5940, 49400]	29800 [25200, 34400]	6120 [278, 49400]
Población Total						
Mean (SD)	17900000 (24900000)	36000000 (68800000)	116000000 (290000000)	19500000 (23600000)	12300000 (11500000)	44000000 (148000000)
Median [Min, Max]	10100000 [200000, 135000000]	9320000 [1060000, 301000000]	24800000 [709000, 132000000]	9490000 [362000, 82400000]	12300000 [4120000, 20400000]	10500000 [200000, 132000000]

Opciones table1.

5.2 Kable

```
# Crear un resumen de datos por continente
tabla_gap <- data_gap %>%
  group_by(continent) %>%
  summarise(
    `Esperanza de Vida Promedio` = round(mean(lifeExp), 1),
    `PIB per cápita Promedio (USD)` = round(mean(gdpPercap), 2),
    `Población Total` = format(sum(pop), big.mark = ",")
  )

# Crear tabla bien formateada para PDF
tabla_gap %>%
  kbl(caption = "Resumen de Indicadores Sociodemográficos por Continente (2007)",
      booktabs = TRUE) %>%
  kable_styling(
    latex_options = c("striped", "hold_position"),
    font_size = 10
  )
```

Tabla 3: Resumen de Indicadores Sociodemográficos por Continente (2007)

continent	Esperanza de Vida Promedio	PIB per cápita Promedio (USD)	Población Total
Africa	54.8	3089.03	929,539,692
Americas	73.6	11003.03	898,871,184
Asia	70.7	12473.03	3,811,953,827
Europe	77.6	25054.48	586,098,529
Oceania	80.7	29810.19	24,549,947

Opciones kableExtra.

1. Chiang, C. L. (1984). *The Life Table and Its Applications*. R.E. Krieger Publishing Company. https://books.google.cl/books?id=T_PrAAAAIAAJ
2. Dobson, A. J., Kuulasmaa, K., Eberle, E., & Scherer, J. (1991). Confidence intervals for weighted sums of poisson parameters. *Statistics in Medicine*, 10(3), 457-462. <https://doi.org/10.1002/sim.4780100317>
3. World Bank. (2023). *World Bank Open Data: Female Population in Chile*. <https://data.worldbank.org/indicator/SP.POP.TOTL.FE.IN?locations=CL>.
4. Vargas, M. (2023). *censo2017: Base de Datos de Facil Acceso del Censo 2017 de Chile (2017 Chilean Census Easy Access Database)*. <https://CRAN.R-project.org/package=censo2017>