

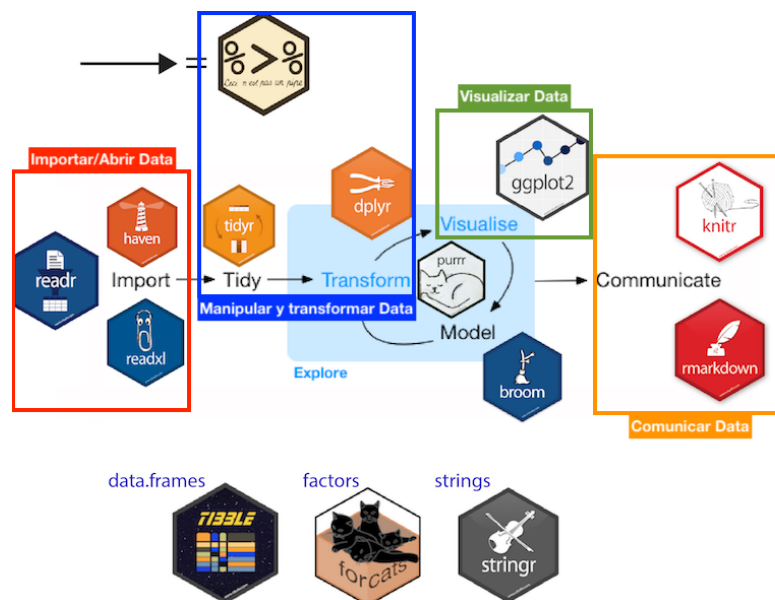
# Herramientas para el reporte de datos en RMarkdown\*

SOC00001 - BIG DATA  
Departamento de Sociología UDP

22 de marzo, 2022

## Introducción

R es un lenguaje de programación que proporciona un amplio abanico de herramientas para la computación estadística (modelos lineales, no lineales, test estadísticos, series de tiempo, algoritmos de clasificación y agrupación, ecuaciones estructurales, entre muchos otros<sup>1</sup>) y la visualización de datos<sup>2</sup>. Además, R junto a su interfaz RStudio conforman un entorno de desarrollo integrado (IDE) que nos permite realizar múltiples tareas<sup>3</sup>:



En cursos anteriores se ha trabajado en importar, manipular, transformar y visualizar datos. Sin embargo, no importa lo bien que estén hechos estos análisis si es que no pueden ser explicados de manera sencilla a otros. Así, el objetivo de este documento es **comprender de manera práctica** los principales elementos para la reporteria de datos en Rmarkdown. En síntesis veremos los siguientes temas:

- R, Markdown y compiladores.
- Flujo de trabajo en Rmarkdown.
- Escritura: títulos, texto, numeración.
- Ecuaciones y símbolos matemáticos.
- Incrustar código: **chunks**.
- Reporte de tablas, gráficos, imágenes.
- Bibliografía.

\***Autor:** José Daniel Conejeros (jdconejeros@uc.cl).

<sup>1</sup>Consulte aquí: <https://www.r-project.org/>

<sup>2</sup>Galería de gráficos: <https://www.r-graph-gallery.com/>

<sup>3</sup>Consulte aquí: <https://rstudio.com/products/rstudio/>

Antes de trabajar se debe tener instalados<sup>4</sup>:

- R. Sino lo tiene instalado puede revisar este tutorial: <https://vimeo.com/203516510>
- RStudio. Sino lo tiene instalado puede revisar este tutorial: <https://vimeo.com/203516968>

## 1. RMarkdown y compiladores

Rmarkdown es un ambiente para elaborar documentos interactivos con un código sencillo de utilizar. Esto lo hace un marco unificado para el desarrollo de análisis cuantitativos combinando texto, códigos y resultados en un solo espacio de trabajo. Estos documentos son completamente reproducibles y soportan variados formatos de salida.

Los archivos de Rmarkdown o .Rmd están diseñados para usarse con el paquete de `rmarkdown` que viene instalado por defecto en Rstudio, sino lo tiene disponible puede instalarlo directamente:

```
install.packages("rmarkdown")
```

Además, cuando estamos produciendo documentos en PDF se requiere de un *compilador* de L<sup>A</sup>T<sub>E</sub>X. En otras palabras, necesitamos un programa que tome el código del documento y genere una salida en PDF. Para esto se recomienda bastante descargar/installar MikTex que cuenta con soporte para Windows, Mac y Linux<sup>5</sup>.

1. Ir a la web de Miktex y descargar: <https://miktex.org/download>.
2. Ejecutar el instalador (apropiado según el sistema operativo de su computador).
3. Esperar que termine la descarga de los paquetes (esto tardará algunos minutos).
4. Una vez completado el instalador se debe reiniciar el equipo.

Una vez realizado estos pasos se puede comenzar a trabajar sin problemas. Los usuarios de Mac también tiene la opción de descargar/installar MacTex. Además, es bastante recomendable trabajar en un Rproject ya que nos permite centralizar la ruta de trabajo con todos los insumos necesarios para el documento<sup>6</sup>.

## 2. Flujo de trabajo en Rmarkdown

Los archivos con extensión .Rmd son el código fuente que tras ser “tejidos” y luego “compilados” se transforman en documentos enriquecidos y reproducibles. El primer paso de **tejer** (o **knitr**) permite ejecutar cada fragmento de código R que esté en el documento (luego se decidirá si el código o el resultado serán visibles).

La principal ventaja de estos documentos es su automatización. Por ejemplo, cuando se produce un gráfico o una tabla y luego aplicamos el clásico *copy-paste* sobre el reporte, vamos a estar realizando un procedimiento manual que tiene dos desventajas: (1) si los datos cambian, será necesario repetir todo el proceso para actualizar y (2) se pueden producir muchos errores involuntarios al *tipear* información. En cambio, gracias a esta herramienta, basta con solo actualizar el código-fuente, volver e tejer el documento y el insumo se actualizará con los cambios.

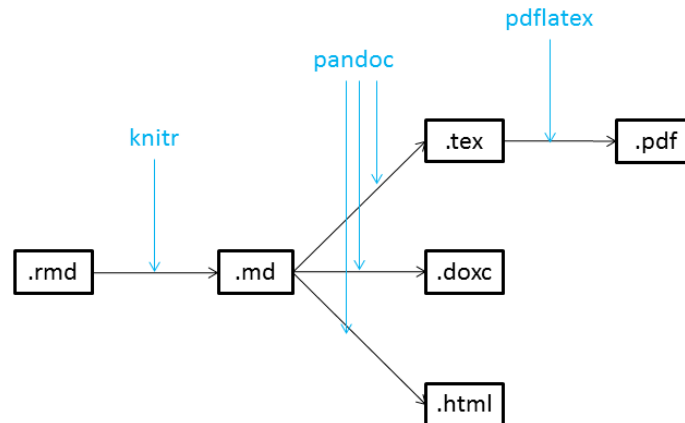


<sup>4</sup>Puedes seguir otros tutoriales en: [tc.instructure.com](http://tc.instructure.com), [www.datacamp.com](http://www.datacamp.com) o [www.dataquest.io](http://www.dataquest.io). Cualquier duda contactar al correo [jdconejeros@uc.cl](mailto:jdconejeros@uc.cl)

<sup>5</sup>No olvide revisar los requisitos computacionales: <https://miktex.org/kb/prerequisites>

<sup>6</sup>Puede consultar aquí: <https://es.r4ds.hadley.nz/flujo-de-trabajo-proyectos.html>

El segundo paso es **compilar/convertir** este documento a través de **pandoc** (un conversor). Esto nos permite transformar el código del archivo en un documento con distintos formatos. Por ejemplo podemos convertir nuestro archivo **.Rmd** en un archivo de salida HTML, PDF o Word e incluso en una presentación de diapositivas en HTML5 o PDF<sup>7</sup>. Gracias a Rstudio podemos realizar estos dos pasos al mismo tiempo:



### 3. Escribir en Rmarkdown

Para crear un informe de R Markdown podría:

- Abrir un archivo de texto sin formato y guardar con la extensión **.Rmd** ó
- Abrir un archivo de texto sin formato en su editor de scripts haciendo clic en:

Archivo > Nuevo archivo > R Markdown:

```

1 ---
2 title: "Mi primer documento"
3 author: "José Daniel Conejeros"
4 date: "3/3/2021"
5 output: pdf_document
6 ---
7
8 {r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.
15 For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
16
17 When you click the **Knit** button a document will be generated that includes both content as well as the output of any
18 embedded R code chunks within the document. You can embed an R code chunk like this:
19
20 {r cars}
21 summary(cars)
22
23 ## Including Plots
24
25 You can also embed plots, for example:
26
27 {r pressure, echo=FALSE}
28 plot(pressure)
29
30 Note that the 'echo = FALSE' parameter was added to the code chunk to prevent printing of the R code that generated the
31 plot.
  
```

<sup>7</sup>Una ventaja adicional es la posibilidad de incrustar código de fuente L<sup>A</sup>T<sub>E</sub>X/ HTML a nuestros documentos independiente el formato de salida. Esto hace más atractivo la posibilidad de agregar variados detalles al documento.

En esta sección veremos cada uno de los elementos necesarios para la escritura de documentos:

#### a. YAML

El **YAML Header** es el bloque del archivo `.Rmd` en donde se especifican todas las configuraciones del documento: nombre, autor(a/es/as), fecha y su formato: tamaño de la fuente (`fonsize`), el interlineado (`linestretch`), márgenes (`geometry`), resumen (`abstract`) y la salida (`output`). Se pueden agregar otras especificaciones al texto con la opción `header-includes`<sup>8</sup> que nos permite cargar paquetes desde `LATEX`, o también se puede utilizar un archivo `.css` para agregar configuraciones en lenguaje HTML.

Veamos el **YAML** con las configuraciones por defecto:

```
---
title: "Mi primer documento"      #Título del documento
author: "José Daniel Conejeros"   #Autor
date: "3/3/2021"                  #Fecha
output: pdf_document              #Salida
---
```

Ahora veamos un **YAML** con especificaciones adicionales:

```
---
title: "Mi primer documento"      #Título del documento
author: "José Daniel Conejeros - jdconejeros@uc.cl" #Autor - Correo
date: '\r format(Sys.time(), "%d%B,%Y")`'          #Fecha automatizada
fontsize: 11pt                          #Tamalo de la fuente
linestretch: "1.0"                      #Interlineado
geometry: margin=0.78in                  #Márgenes
output: pdf_document                   #Documento de salida
header-includes:                        #Especificaciones en latex
- \usepackage[spanish,es-tabla,es-nodecimaldot]{babel} #"Tabla" y "Figura" en español.
- \usepackage[utf8]{inputenc}             #Codificación del texto
- \usepackage{booktabs}                  # Formato de tablas
abstract: "Aquí pueden agregar un resumen de lo que van a escribir (opcional)." #Resumen
---
```

---

<sup>8</sup>Puede complementar en el siguiente enlace: <https://cran.r-project.org/web/packages/ymlthis/vignettes/yaml-fieldguide.html>

Ahora veamos un YAML más complejo:

```
---
title: |
  | \vspace{-1cm} Herramientas para el reporte de datos en `RMarkdown`
subtitle: |
  | [SOC00001 - BIG DATA] (https://naimbro.github.io/programa\_udp\_2022.html)
  | [Departamento de Sociología UDP] (https://socialesehistoria.udp.cl/carreras/sociologia/)
date: |
  | \vspace{-0.5cm} `r Sys.setlocale("LC_TIME", "es_ES.UTF-8"); format(Sys.time(), "%d de %B, %Y")`
output:
  pdf_document:
    fig_caption: yes
    latex_engine: xelatex
    keep_tex: false
fontsize: 10pt
geometry: "left=2.2cm,right=2.2cm,top=2.5cm,bottom=2.5cm"
linkcolor: "blue"
header-includes:
  - \usepackage[spanish,es-tabla, es-nodecimaldot]{babel}
  - \usepackage[utf8]{inputenc}
  - \usepackage{leading}
  - \leading{13pt}
  - \usepackage[bottom]{footmisc}
  - \usepackage{multicol}
  - \usepackage{caption}
  - \usepackage{booktabs}
  - \usepackage{graphicx}
  - \usepackage{makecell}
  - \usepackage{fancyhdr}
  - \pagestyle{fancy}
  - \setlength\headheight{30pt}
  - \renewcommand{\headrulewidth}{0pt}
  - \fancypagestyle{plain}{\pagestyle{fancy}}
  - \fancyhead[L]{}
  - \fancyhead[R]{\includegraphics[width=5cm]{image/im0.png}}
  - \usepackage{floatrow}
  - \floatsetup[figure]{capposition=top}
  - \floatsetup[table]{capposition=top}
  - \newcommand{\bcenter}{\begin{center}}
  - \newcommand{\ecenter}{\end{center}}
  - \newcommand{\btwocol}{\begin{multicols}{2}}
  - \newcommand{\etwocol}{\end{multicols}}
thanks: |
  | **Autor:** José Daniel Conejeros (jdconejeros@uc.cl).
#abstract: "Aquí pueden agregar un resumen"
---
```

## b. Encabezados, texto y listas

b1. Los encabezados parten con el signo # y tienen un orden jerárquico:

# Encabezado de primer nivel (Título grande)      **Encabezado de primer nivel**

## Encabezado de segundo nivel (Título mediano) **Encabezado de segundo nivel**

### Encabezado de tercer nivel (Título pequeño) **Encabezado de tercer nivel**

(...) Puede seguir generando sub-encabezados.

## b2. Formato al texto:

- **\*cursiva\*** o **\_cursiva\_** para escribir en *cursiva*
- **\*\*negrita\*\*** o **\_\_negrita\_\_** para escribir en **negrita**
- **superíndice^2** para agregar un superíndice<sup>2</sup>
- **subíndice~2~** para agregar un subíndice<sub>2</sub>
- Hipervínculos: [Link a google](www.google.cl), se obtiene como salida: [Link a google](http://www.google.cl).

## b3. Listas:

Permite enumerar elementos en el texto.

Numeradas:

1. Para el primer ítem
2. Para el segundo ítem
3. Para el tercer ítem

Letras:

- a. Para el primer elemento
- b. Para el segundo elemento
- c. Para el tercer elemento

Viñetas:

- \* Una viñeta cuadrada
- + Una viñeta circular
- Una viñeta circular

Por ejemplo:

1. Deportes:

a. Fútbol:

- \* 11 Jugadores

b. Tenis:

- \* 2 a 4 Jugadores

1. Deportes:

a. Fútbol:

- 11 Jugadores

b. Tenis:

- 2 a 4 Jugadores

## 4. Expresiones matemáticas

Ecuaciones y otras expresiones matemáticas se pueden escribir utilizando el operador `$` al inicio y al final. Esto permite insertar expresiones matemáticas en lenguaje latex de forma sencilla<sup>9</sup>:

- Un `$` permite agregar la expresión en el texto:

Escribir el promedio en Rmarkdown: `$\overline{x}=\frac{x_1+x_2+\cdots+x_n}{n}$`

Da la siguiente salida:  $\bar{x} = \frac{x_1+x_2+\cdots+x_n}{n}$

- Dos `$$` permite agregar la expresión en un bloque visualizado:

Escribir la desviación estándar en Rmarkdown:

````s=\sqrt{\frac{1}{N-1} \sum_{i=1}^n (x_i-\bar{x})^2}````

Da la siguiente salida:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

También se puede ejecutar código<sup>10</sup> de `LaTeX`:

```
$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix}$$
```

Entrega la siguiente salida:

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix}$$

Por último, se pueden extraer ecuaciones matemáticas con la librería `equationmatic`<sup>11</sup>.

<sup>9</sup>Puede consultar los símbolos matemáticos disponibles en: [https://oeis.org/wiki/List\\_of\\_LaTeX\\_mathematical\\_symbols](https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols) o revisar: [https://www.overleaf.com/learn/latex/List\\_of\\_Greek\\_letters\\_and\\_math\\_symbols](https://www.overleaf.com/learn/latex/List_of_Greek_letters_and_math_symbols)

<sup>10</sup>Puede consultar más detalles en el punto 2.5.3 de este bookdown: <https://bookdown.org/yihui/rmarkdown/markdown-syntax.html> o ver más expresiones en: <https://en.wikibooks.org/wiki/LaTeX/Mathematics>

<sup>11</sup>Más detalles en: <https://datalogorax.github.io/equationmatic/articles/intro-equationmatic.html#other-models-1>

## 5. Incrustar código: chunk

Para ejecutar un código dentro del documento se necesita incrustar un **bloque** o **chunk**. Para esto hay tres caminos:

1. Utilizar un atajo de teclado: `Cmd/Ctrl + option/Alt + I`.
2. Con el ícono **insert** en la barra de edición.
3. Tipeando de forma manual los delimitadores de bloque al inicio: `````{r}` y al final: ``````.

Por ejemplo, se puede explorar el siguiente **chunk**:

```
````{r echo=TRUE, message=FALSE, warning=FALSE}
A = 1+1
A
B = A/2
B
A != B
````
```

Se obtiene la siguiente salida:

```
A = 1+1
A
```

```
## [1] 2
```

```
B = A/2
B
```

```
## [1] 1
```

```
A != B
```

```
## [1] TRUE
```

La salida de los bloques se pueden personalizar con las **options** que son argumentos suministrados en el encabezado del bloque. **Knitr** cuenta con alrededor de 60 opciones que se pueden utilizar para personalizar los bloques de códigos<sup>12</sup>. Los más utilizados controlan si el bloque es ejecutado y las especificaciones de la salida (en general son opciones lógicas aunque pueden variar en otras formas):

- `eval = FALSE` evita que el código sea evaluado. Si el código no es ejecutado, entonces no se generaran resultados. Esto es útil para mostrar códigos de ejemplo, o para deshabilitar un gran bloque de código sin comentar cada línea.
- `include = FALSE` ejecuta un código, pero no lo muestra y tampoco presenta el resultado. Se usa para el código de configuración del reporte.
- `echo = FALSE` evita que se vea el código, pero sí muestra los resultados en el reporte.
- `message = FALSE` o `warning = FALSE` evita que aparezcan mensajes o advertencias.
- `results = 'hide'` oculta el output impreso; `results='asis'` el código creado por las funciones sea empleado en la compilación; `fig.show = 'hide'` oculta los gráficos.

Otro elemento muy útil es el **código en línea**, en otras palabras, se puede incrustar un resultado de **R** en un texto solo usando la expresión ``r``. Esto evita errores de digitación y facilita la actualización rápida en las cifras de reportes. Por ejemplo: El resultado de `1+1` es ``r` 1+1`` → **Salida:** El resultado de `1+1` es 2.

<sup>12</sup>Puede revisar todas las opciones en el siguiente enlace: <https://yihui.org/knitr/options/>



## 6. Reporte de tablas, gráficos e imágenes

### a. Tablas

Hay varias maneras de construir tablas en Rmarkdown una de las más sencillas (pero menos automatizadas) es elaborar la tabla de forma manual en lenguaje pandoc:

```
País	Gasto Social	Votación por la izquierda
Inglaterra	16.0	20
Alemania	29.7	37.0
Portugal	21.6	34.4
Dinamarca	34.4	41.9
```

Esto produce la siguiente salida:

| País       | Gasto Social | Votación por la izquierda |
|------------|--------------|---------------------------|
| Inglaterra | 16.0         | 20                        |
| Alemania   | 29.7         | 37.0                      |
| Portugal   | 21.6         | 34.4                      |
| Dinamarca  | 34.4         | 41.9                      |

A pesar de que esto es sencillo, toma tiempo elaborar tablas con mucha información o especificaciones más complejas. Además, es vulnerable a errores cuando los datos son actualizados. Dada estas limitaciones, hay dos librerías que facilitan la gestión y automatización de tablas<sup>13</sup>: `kable` y `kableExtra`<sup>14</sup>.

Un ejemplo:

```
#Crear una tabla con valores
tabla <- data.frame(pais=c("Inglaterra", "Alemania", "Portugal", "Dinamarca"),
                    gto_social=c(16, 29.7, 21.6, 34.4),
                    votacion_izq=c(20, 37, 34.4, 41.9))
#Ver la estructura de la tabla
str(tabla)
```

```
## 'data.frame':    4 obs. of  3 variables:
## $ pais          : chr  "Inglaterra" "Alemania" "Portugal" "Dinamarca"
## $ gto_social     : num  16 29.7 21.6 34.4
## $ votacion_izq   : num  20 37 34.4 41.9
```

```
tabla #Esta tabla no es de calidad
```

```
##      pais gto_social votacion_izq
## 1 Inglaterra    16.0         20.0
## 2  Alemania    29.7         37.0
## 3  Portugal    21.6         34.4
## 4  Dinamarca    34.4         41.9
```

<sup>13</sup>Otras librerías: `xtable` (<https://cran.r-project.org/web/packages/xtable/vignettes/xtableGallery.pdf>), `stargazer` (<https://cran.r-project.org/web/packages/stargazer/vignettes/stargazer.pdf>), `pander` (<http://rapporter.github.io/pander/>), `DT` (<https://rstudio.github.io/DT/>), `flextable` (<https://davidgohel.github.io/flextable/>), `gt` (<https://gt.rstudio.com/>), `huxtable` (<https://github.com/hughjonesd/huxtable>), etc.

<sup>14</sup>Todo el detalle de esta librería lo puede encontrar en: <https://haozhu233.github.io/kableExtra/> para trabajar en pdf se recomienda revisar esta guía: [https://haozhu233.github.io/kableExtra/awesome\\_table\\_in\\_pdf.pdf](https://haozhu233.github.io/kableExtra/awesome_table_in_pdf.pdf)

```
#Generar la tabla:
#install.packages("kableExtra")
library(kableExtra)
kable(tabla) # Generamos la tabla
```

| pais       | gto_social | votacion_izq |
|------------|------------|--------------|
| Inglaterra | 16.0       | 20.0         |
| Alemania   | 29.7       | 37.0         |
| Portugal   | 21.6       | 34.4         |
| Dinamarca  | 34.4       | 41.9         |

Se puede mejorar esta tabla agregando una serie de argumentos (especificaciones a la función):

```
#Tabla de calidad
#install.packages("dplyr")
library(dplyr) # Cargamos para usar el pipe (%>%) que encadena operaciones
kable(tabla, booktabs =TRUE, #Estilo de la tabla
      align = c("ccc"), #Alineación de columnas
      linesep = "", #Sin línea entre columnas
      caption = "Tabla de calidad", #Título
      col.names = c("País", "Gasto Social (%)",
                    "Votación de izquierda (%)")) %>% #Nombre de las columnas
kable_styling(latex_options = c("hold_position")
              , full_width = F) %>% #Posición de tabla
kableExtra::footnote(general = "Elaboración propia.", footnote_as_chunk = T,
                     general_title="Fuente:", fixed_small_size=T) #Nota al pie de la tabla
```

Tabla 2: Tabla de calidad

| País       | Gasto Social (%) | Votación de izquierda (%) |
|------------|------------------|---------------------------|
| Inglaterra | 16.0             | 20.0                      |
| Alemania   | 29.7             | 37.0                      |
| Portugal   | 21.6             | 34.4                      |
| Dinamarca  | 34.4             | 41.9                      |

*Fuente:* Elaboración propia.

Puede parecer un poco extenso el código, no obstante, en el mediano plazo esto es mejor por su reproducibilidad y su fácil edición. Ahora se va a representar el siguiente modelo de regresión OLS:

$$Gto\_Social = \beta_0 + \beta_1 * Vot\_Izquierda + \epsilon$$

```
modelo <- lm(gto_social ~ votacion_izq, data=tabla) #Estimar
modelo #Ver la salida en R
```

```
##
## Call:
## lm(formula = gto_social ~ votacion_izq, data = tabla)
##
## Coefficients:
## (Intercept) votacion_izq
## -1.4293 0.8058
```

Hay varios caminos para generar una tabla de calidad para modelos de regresión<sup>15</sup>:

- Librería `kable` + función `tidy` (de la librería `broom`):

```
#install.packages("broom")
library(broom)
mod <- tidy(modelo)
mod$term <- c("Intercepto", "Votación por la Izquierda (%)")
kable(mod, booktabs = TRUE, digits = 3, align = c("cccc"), caption = "Modelo OLS para Gasto Social",
      col.names = c(" ", "Beta", "Error estándar", "Valor t", "Valor p")) %>%
  kable_styling(latex_options = c("hold_position"), full_width = F) %>%
  kableExtra::footnote(general = "Elaboración propia.", footnote_as_chunk = T,
                      general_title = "Fuente:", fixed_small_size = T)
```

Tabla 3: Modelo OLS para Gasto Social

|                               | Beta   | Error estándar | Valor t | Valor p |
|-------------------------------|--------|----------------|---------|---------|
| Intercepto                    | -1.429 | 8.113          | -0.176  | 0.876   |
| Votación por la Izquierda (%) | 0.806  | 0.236          | 3.408   | 0.076   |

*Fuente:* Elaboración propia.

- Librería `texreg`<sup>16</sup>

```
#install.packages("kableExtra")
library(texreg) #Especificar en el chunk la opción {results='asis'} para visualizar.
texreg(l=list(modelo), #Modelo a utilizar
       caption="Modelo OLS para Gasto Social", #Título
       custom.coef.names=c("Intercepto", "Votación por la Izquierda (%)", #Nombre de variables
                           float.pos="h" , digits = 3, #Posición y dígitos
                           single.row = FALSE, #Sin línea de fila
                           fsingle.row = T,
                           caption.above = TRUE, #Título sobre la tabla
                           include.ci = FALSE, #Mostrar el intervalo
                           stars = c(0.01, 0.05, 0.1)) #Mostrar el nivel de significancia
```

Tabla 4: Modelo OLS para Gasto Social

|                               | Model 1           |
|-------------------------------|-------------------|
| Intercepto                    | -1.429<br>(8.113) |
| Votación por la Izquierda (%) | 0.806*<br>(0.236) |
| R <sup>2</sup>                | 0.853             |
| Adj. R <sup>2</sup>           | 0.780             |
| Num. obs.                     | 4                 |

\*\*\* $p < 0.01$ ; \*\* $p < 0.05$ ; \* $p < 0.1$

<sup>15</sup>Puede ver otros caminos en: <https://stackoverflow.com>

<sup>16</sup>Puede profundizar en: <https://cran.r-project.org/web/packages/texreg/vignettes/texreg.pdf>

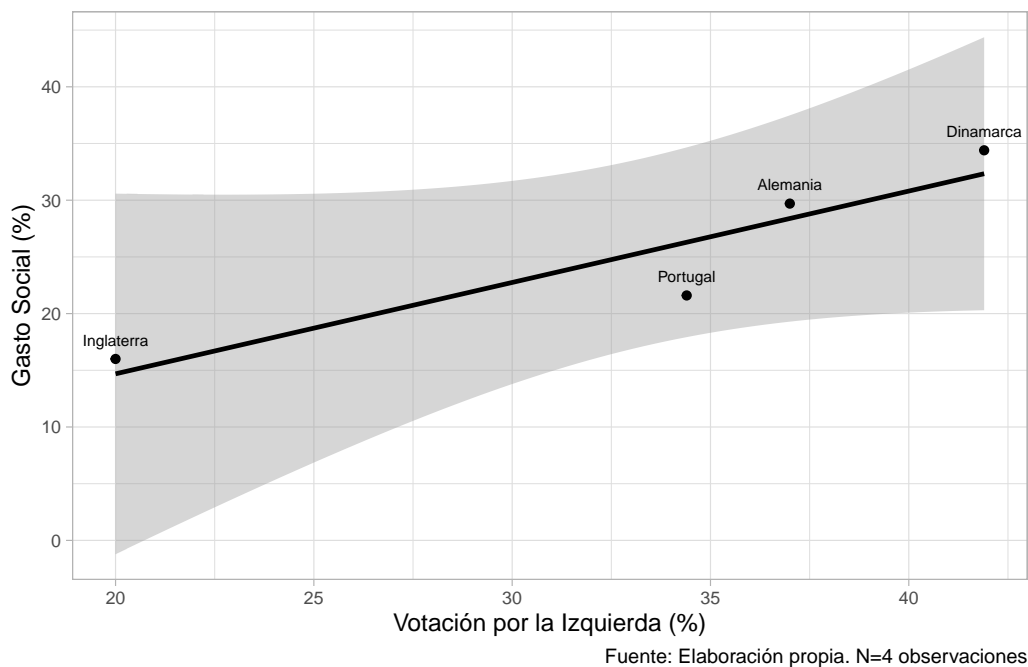
## b. Gráficos

Para insertar un gráfico se puede ocupar la librería `ggplot2` y realizar las siguientes especificaciones adicionales en el chunk:

- `fig.align = 'center'`: Centrar la figura
- `fig.cap = 'Asociación de variables'`: Nombre de la figura
- `out.height = "80%"`: Alto de la figura
- `out.width = "80%"`: Ancho de la figura

```
#Gráfico en ggplot
#install.packages("ggplot2")
library(ggplot2)
g1 <- ggplot(tabla, aes(y=gto_social, x=votacion_izq)) +
  geom_smooth(method = "lm", colour="black", se=T, formula=y ~ x) +
  geom_point() +
  geom_text(aes(label=pais), hjust=0.5, vjust=-1.2, size=2.5) +
  labs(y = "Gasto Social (%)", x = "Votación por la Izquierda (%)",
       title="", caption = "Fuente: Elaboración propia. N=4 observaciones") +
  theme_light() +
  theme(axis.text.y = element_text(size=8),
        axis.text.x = element_text(size=8))
g1 #Ver el gráfico
```

Figura 1: Asociación de variables



```
ggsave("graficos/graph1.png", plot = g1) #Guardar el gráfico
```

### c. Imágenes

Hay dos caminos sencillos para insertar imágenes:

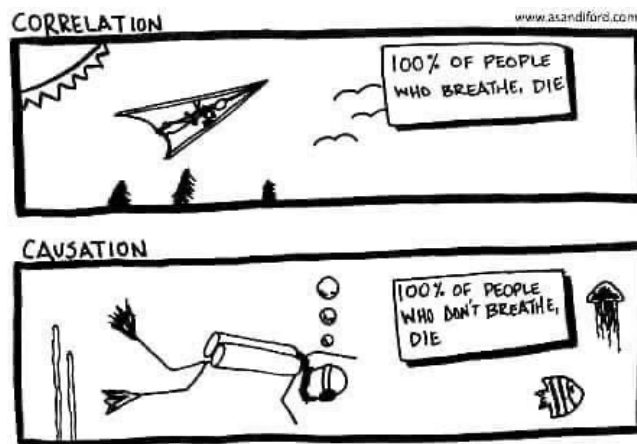
- Desde el lenguaje base de Rmarkdown:

```
! [Figura] (/Users/josedanielconejeros/Dropbox/Taller I-2021/Ayudantia/Ayud1_Intro/image/f5.png)
```

Sin embargo, este camino depende bastante de las rutas de trabajo, por lo que no es muy recomendado ya que dificulta la replicación (produce errores en documentos complejos). Lo ideal es trabajar con un Rproject y las operaciones a continuación.

- Función `include_graphics` del paquete `knitr`:

```
knitr::include_graphics("image/f5.png")
```



Se pueden ajustar estos parámetros<sup>17</sup> del chunk:

- `fig.align` = 'center': Alineación de la figura en el centro.
- `fig.height` = 1: Altura.
- `out.height` = "50%": Proporción del alto de la salida.
- `out.width` = "50%": Proporción del ancho de la salida.

<sup>17</sup>Para otros parámetros puede revisar: <https://yihui.org/knitr/options/#plots>

## 7. Agregar Bibliografía

Para el trabajo con referencias bibliográficas se requiere de un archivo de bibliografía usando la opción `bibliography` en los metadatos del YAML. Por ejemplo:

```
---
title: "Mi primer documento"
author: "José Daniel Conejeros"
date: "3/3/2021"
output: pdf_document
bibliography: references.bib
---
```

Un documento de extensión `.bib`<sup>18</sup> es un archivo de texto plano BibTeX que consta de entradas bibliográficas tales como:

```
@Manual{R-base,
  title = {R: A Language and Environment for Statistical
           Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2019},
  url = {https://www.R-project.org},
}
```

Se puede citar directamente en el documento utilizando el código `@key` donde el `key` es la primera línea de la entrada anterior: `@R-base`. Para dejar las citas entre paréntesis, utilice `[@R-base]`<sup>19</sup>.

## Nota Final

Es recomendable practicar cada una de las herramientas revisadas. La curva de aprendizaje es empinada al inicio, pero se pueden obtener muchos beneficios en el mediano plazo. Al igual que aprender un idioma, R y RMarkdown requieren de mucha practica, ensayo/error y paciencia. Suerte en esta hazaña de construir documentos en RMarkdown.

<sup>18</sup>Se puede usar otros archivos de extensión distintos al `.bib`, puede consultar en las referencias de este documento.

<sup>19</sup>Puede profundizar en <https://bookdown.org/yihui/bookdown/citations.html>, <https://bookdown.org/yihui/rmarkdown-cookbook/bibliography.html> y [https://rmarkdown.rstudio.com/authoring\\_bibliographies\\_and\\_citations.html](https://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html)

## Referencias

Hay múltiples referencias, paquetes, herramientas y formas de hacer las cosas en **RMarkdown**. Para no intimidarse con tal magnitud de información puede acceder a las siguientes referencias:

- [R Markdown: The Definitive Guide](#): Una referencia obligada para aprender **Rmarkdown** desarrollada por Yihui Xie, J. J. Allaire & Garrett Grolemond en el 2020. (Yihui Xie es uno de los impulsores de **Rmarkdown**).
- [R para Ciencia de Datos](#): Libro base para el uso de R, aquí podrán ver que la plataforma de Rstudio no es solo para el análisis estadístico, sino que de procesamiento de datos y reporte.
- [Funciones de RMarkdown](#): Una referencia compacta de Yongfu Liao (2019).
- [Conceptos básicos de RMarkdown](#): Una referencia más detallada de Andy Lin de UCLA.
- [Tutorial en RStudio](#): tutorial rápido con opciones básicas.
- [RMarkdown Cheat Sheets](#): Recurso visual en **inglés**.
- [Guía de referencia RMarkdown](#): Recurso visual en **inglés**.
- [Latex Cheat Sheets](#): Recurso para tener una idea base de  $\text{\LaTeX}$ .

## Foros

Para los que alguna vez fuimos nuevos en **Rmarkdown** sirve bastante ver las preguntas/respuestas de otras personas en las comunidades de R (¡son muy activas!). De hecho, casi todas nuestras preguntas ya fueron respondidas por personas en todo el mundo. No olvidar que la mayoría de estos foros están en inglés:

- [RStudio Community](#)
- [Stackoverflow](#)