

Analisis gráfico con R

Gráficos con R base

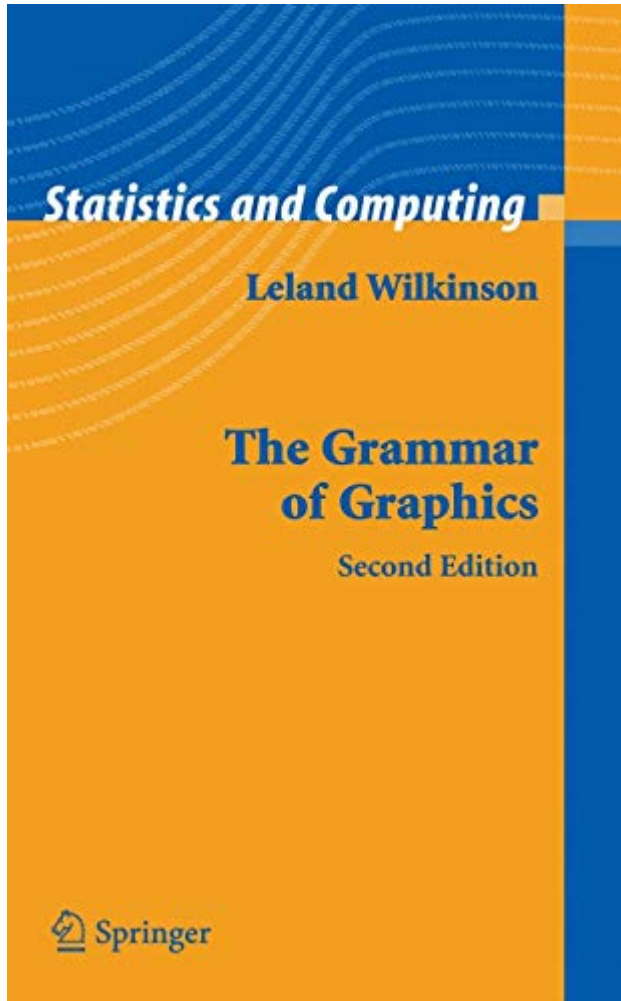
Jaime Pinilla - C. González-Martel
27/10/2022

Análisis gráfico con R

1. Gráficos interactivos con R.
2. Generación de informes.

Gráficos interactivos con R.

Ggplot es una librería de R integrada dentro de tidyverse para la generación de gráficos estáticos complejos utilizando los principios de la gramática de gráficos de *Leland Wilkinson*.

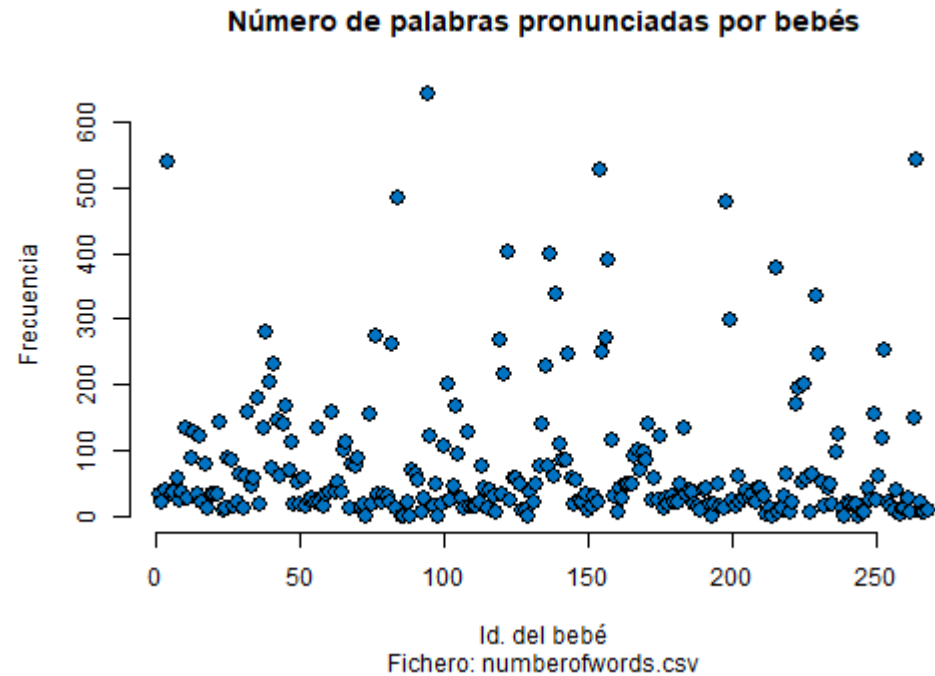


Plotly es una librería de código abierto en R y se basa en el *framework* de JavaScript con el mismo nombre, plotly.js. El paquete Plotly ayuda a crear gráficos interactivos e intuitivos. También brinda la posibilidad de incrustar estos gráficos en páginas web guardarlos en sus computadoras. Se usa ampliamente junto con el paquete ggplot para hacer visualizaciones complejas, atractivas e interactivas para el usuario.

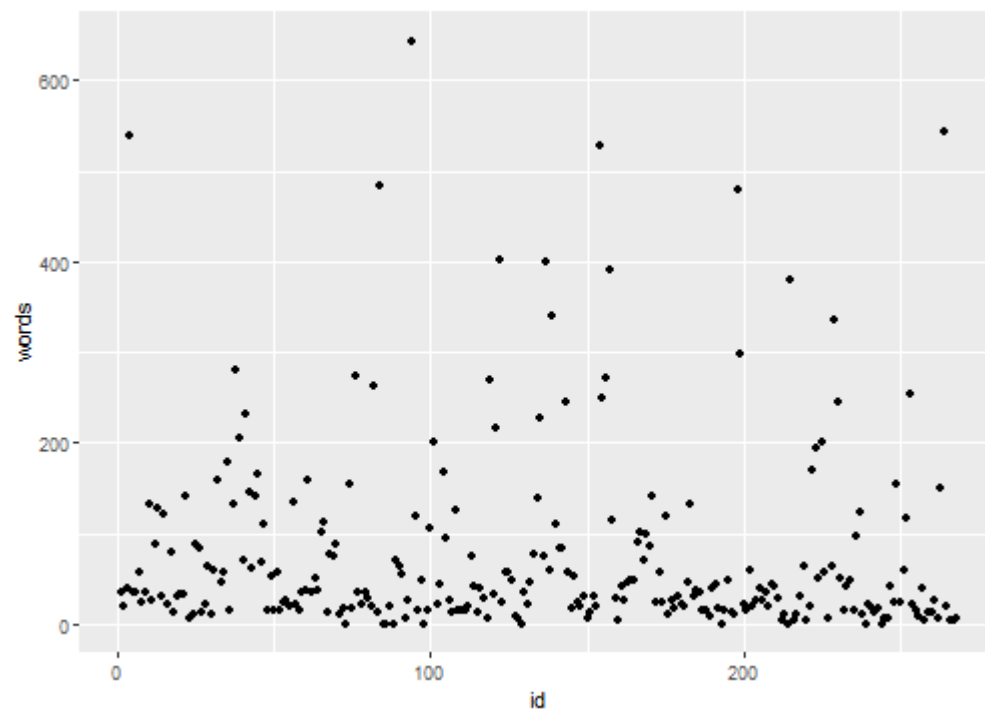


Base

```
numberofwords <- readr::read_csv("../Modulo 2/numberofwords.csv")  
plot(numberofwords$words, ylab = "Frecuencia", xlab = "Id. del bebé", main = "Número de palabras pronunciadas por
```



```
library(dplyr)
library(ggplot2)
numberofwords %>%
  ggplot(aes(x = id, y = words)) +
  geom_point()
```



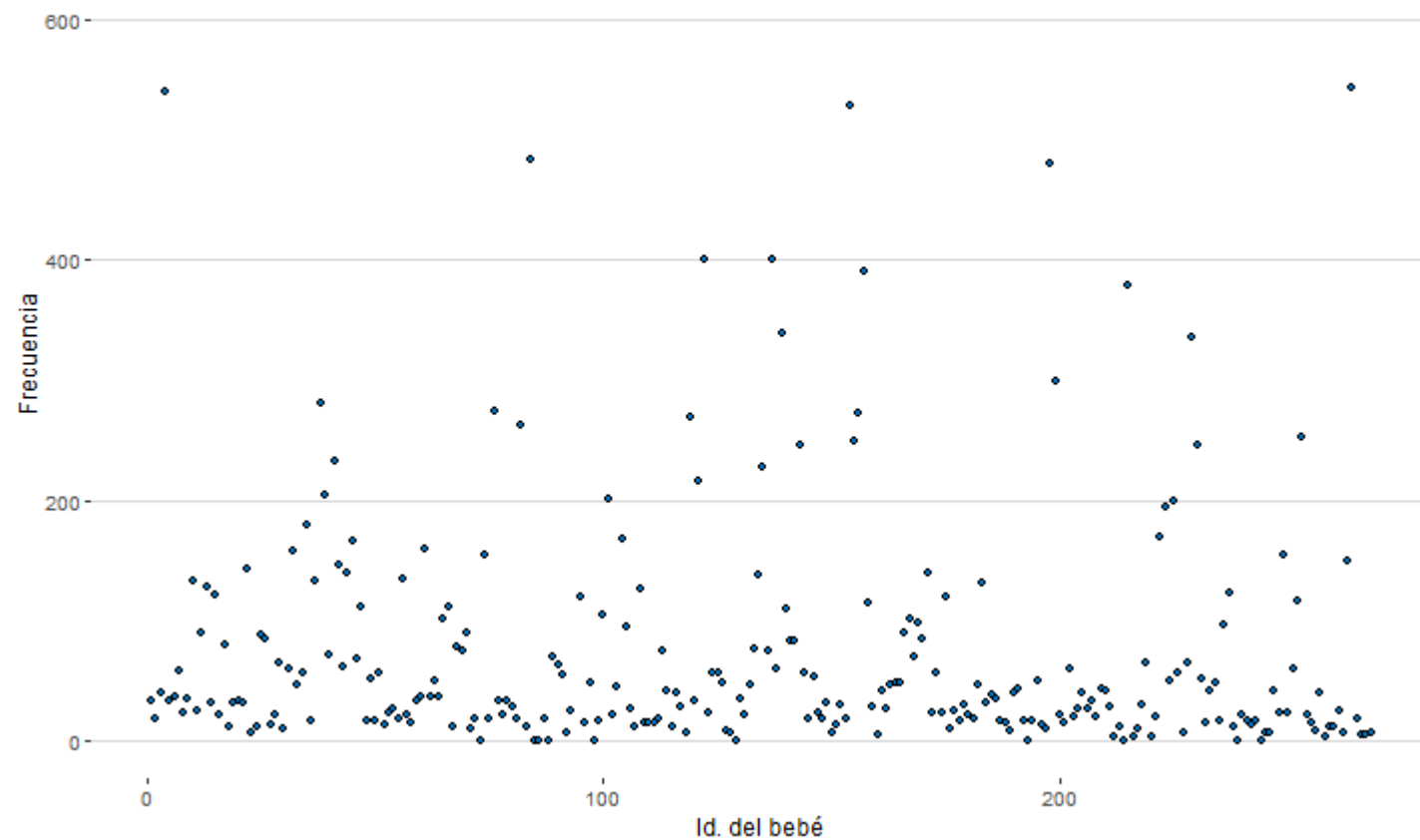
Enlaces

- [geom_point](#)
- [labs](#)
- [themes](#)
- [ggthemes](#)


```
library(ggthemes)
numberofwords %>%
  ggplot(aes(x = id, y = words)) +
  geom_point(colour = "black", fill = "#0073C2FF", size = 1.5, shape = 21) +
  labs(title = "Número de palabras pronunciadas por bebés",
        subtitle = "Fichero: numberofwords.csv",
        x = "Id. del bebé",
        y = "Frecuencia") +
  theme_hc()
```

Número de palabras pronunciadas por bebés

Fichero: numberofwords.csv



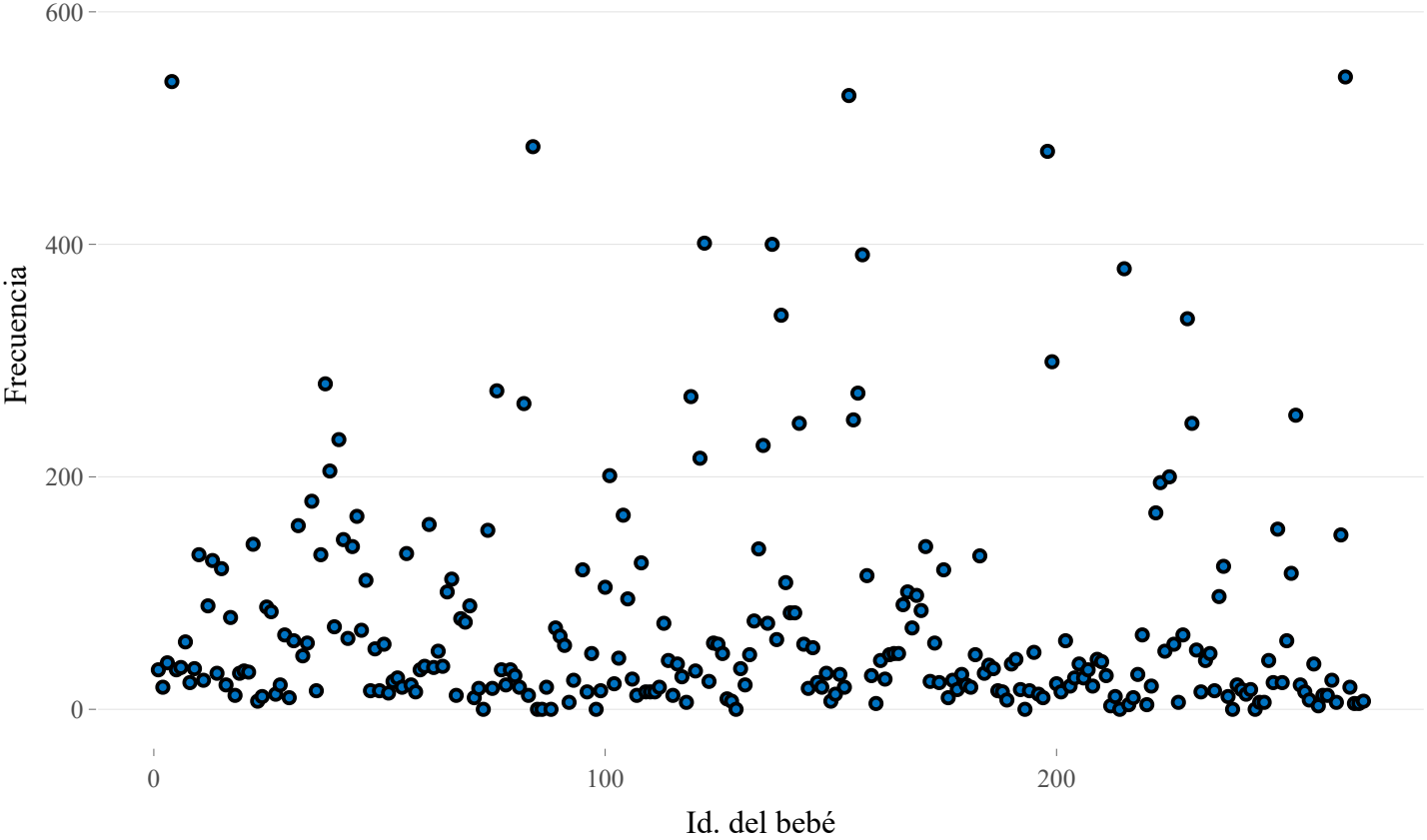
Plotly

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

migrafico <- numberofwords %>%
  ggplot(aes(x = id, y = words)) +
  geom_point(colour = "black", fill = "#0073C2FF", size = 1.5, shape = 21) +
  labs(title = "Número de palabras pronunciadas por bebés",
        subtitle = "Fichero: numberofwords.csv",
        x = "Id. del bebé",
        y = "Frecuencia") +
  theme_hc()

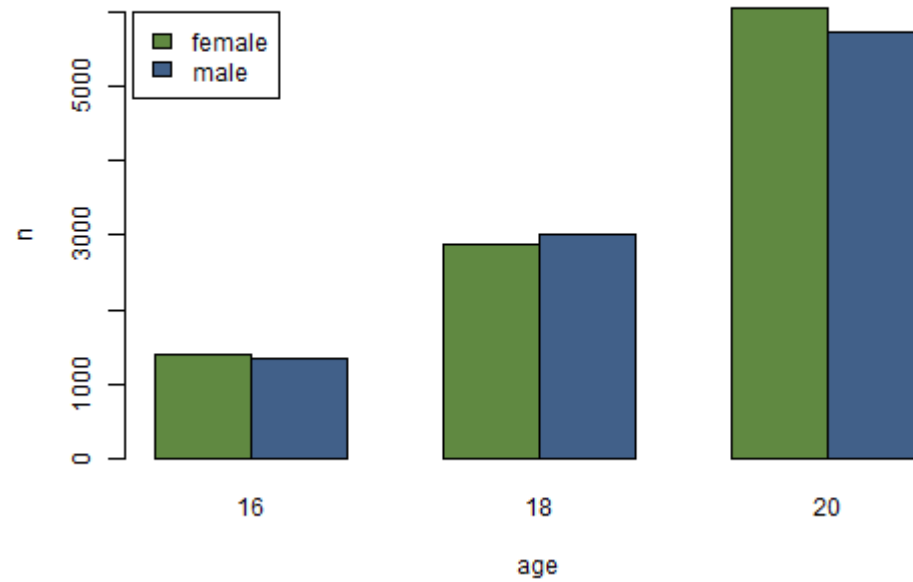
ggplotly(migrafico)
```

Número de palabras pronunciadas por bebés



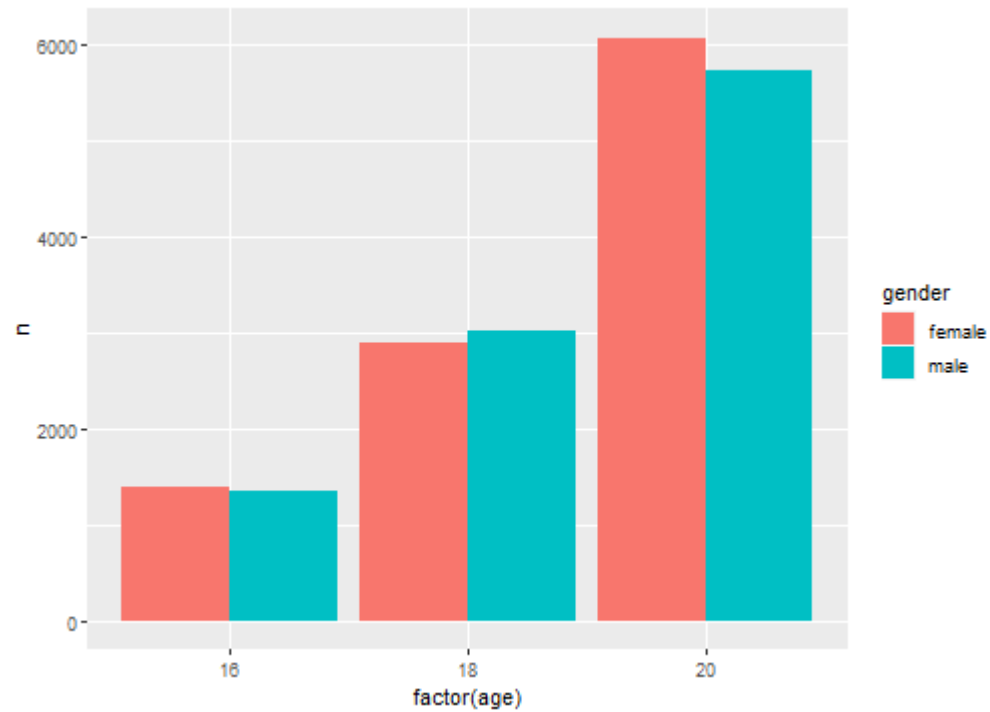
Base

```
freq <- numberofwords %>% group_by(age, gender) %>% summarise(n = sum(words))  
barplot(n ~ gender + age, data = freq, beside = T, col = c("#608941", "#416089"), legend = levels(factor(freq$gender)))
```



Ggplot

```
freq %>%  
ggplot(aes(x = factor(age), y = n, fill = gender)) +  
  geom_col(position = "dodge")
```

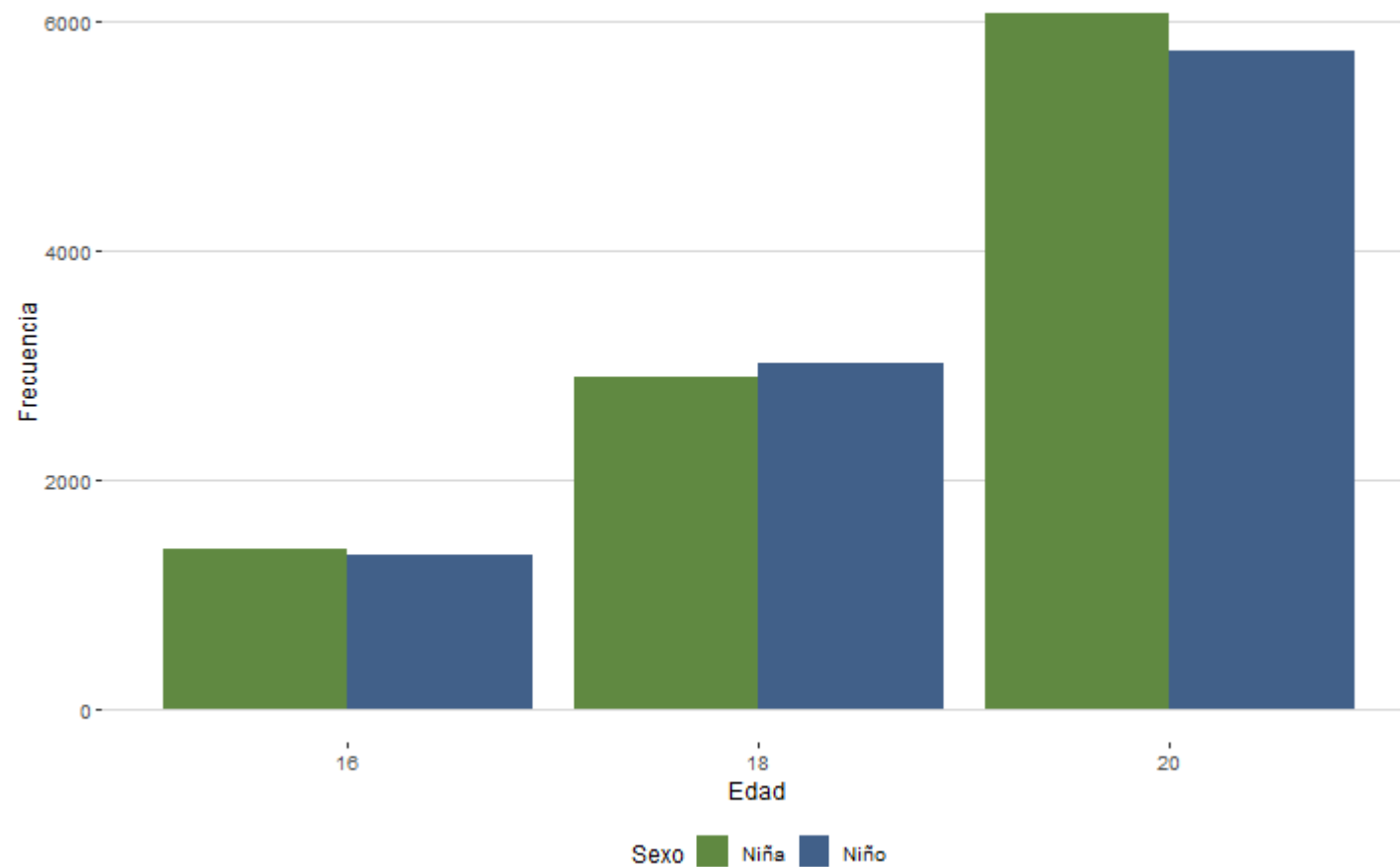


Enlaces

- [scale_fill_manual](#)

```
freq %>%  
ggplot(aes(x = factor(age), y = n, fill = gender)) +  
  geom_col(position = "dodge") +  
  scale_fill_manual(values = c("#608941", "#416089"),  
                    labels = c("Niña", "Niño"),  
                    name = "Sexo") +  
  labs(title = "Número de palabras pronunciadas por bebés",  
        subtitle = "según edad (meses) y sexo",  
        x = "Edad",  
        y = "Frecuencia") +  
  theme_hc()
```

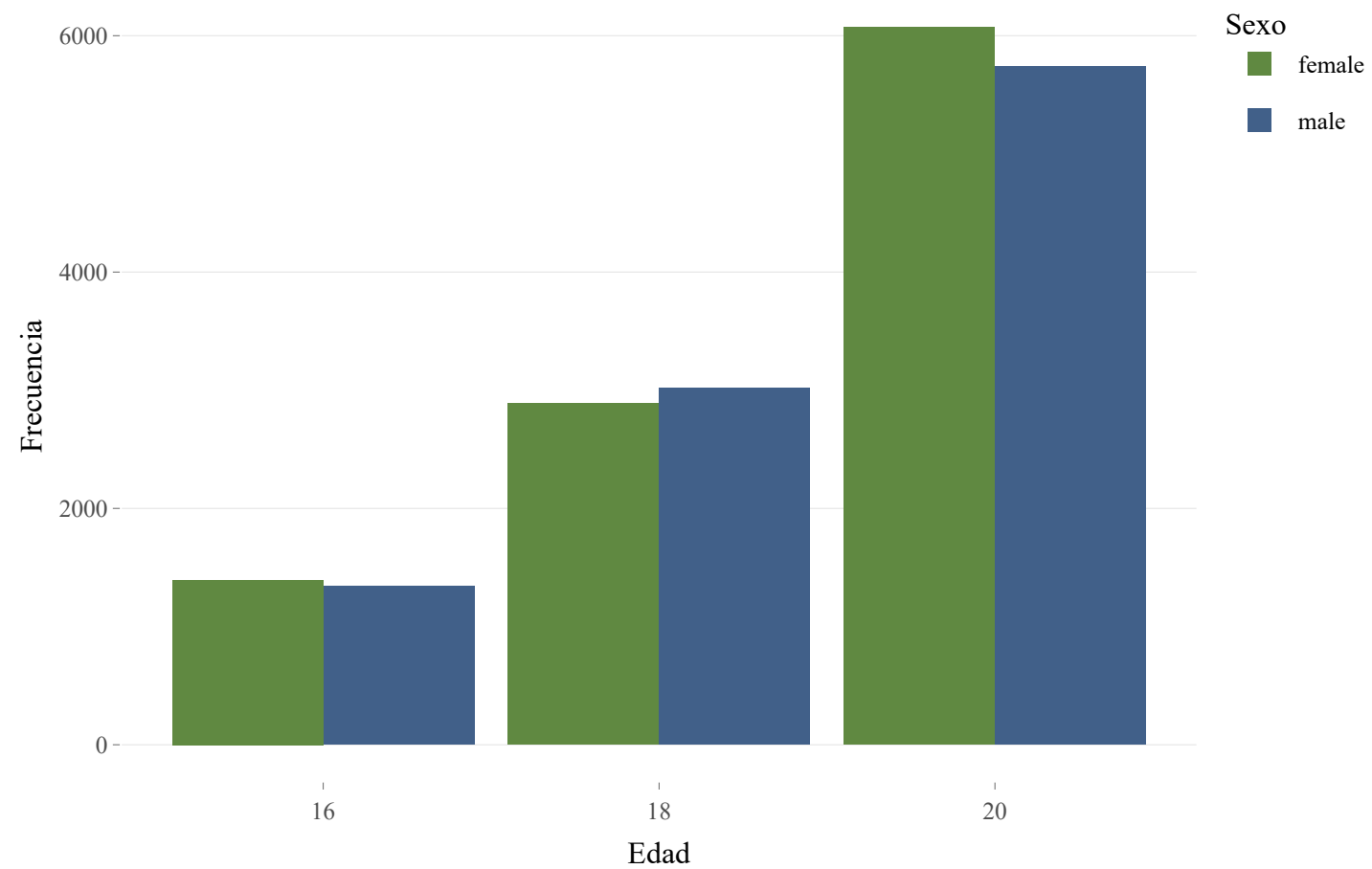

Número de palabras pronunciadas por bebés
según edad (meses) y sexo



Plotly

```
miplotly <- freq %>%  
  ggplot(aes(x = factor(age), y = n, fill = gender)) +  
    geom_col(position = "dodge") +  
    scale_fill_manual(values = c("#608941", "#416089"),  
                      labels = c("Niña", "Niño"),  
                      name = "Sexo") +  
    labs(title = "Número de palabras pronunciadas por bebés",  
         subtitle = "según edad (meses) y sexo",  
         x = "Edad",  
         y = "Frecuencia") +  
    theme_hc()  
  
ggplotly(miplotly)
```

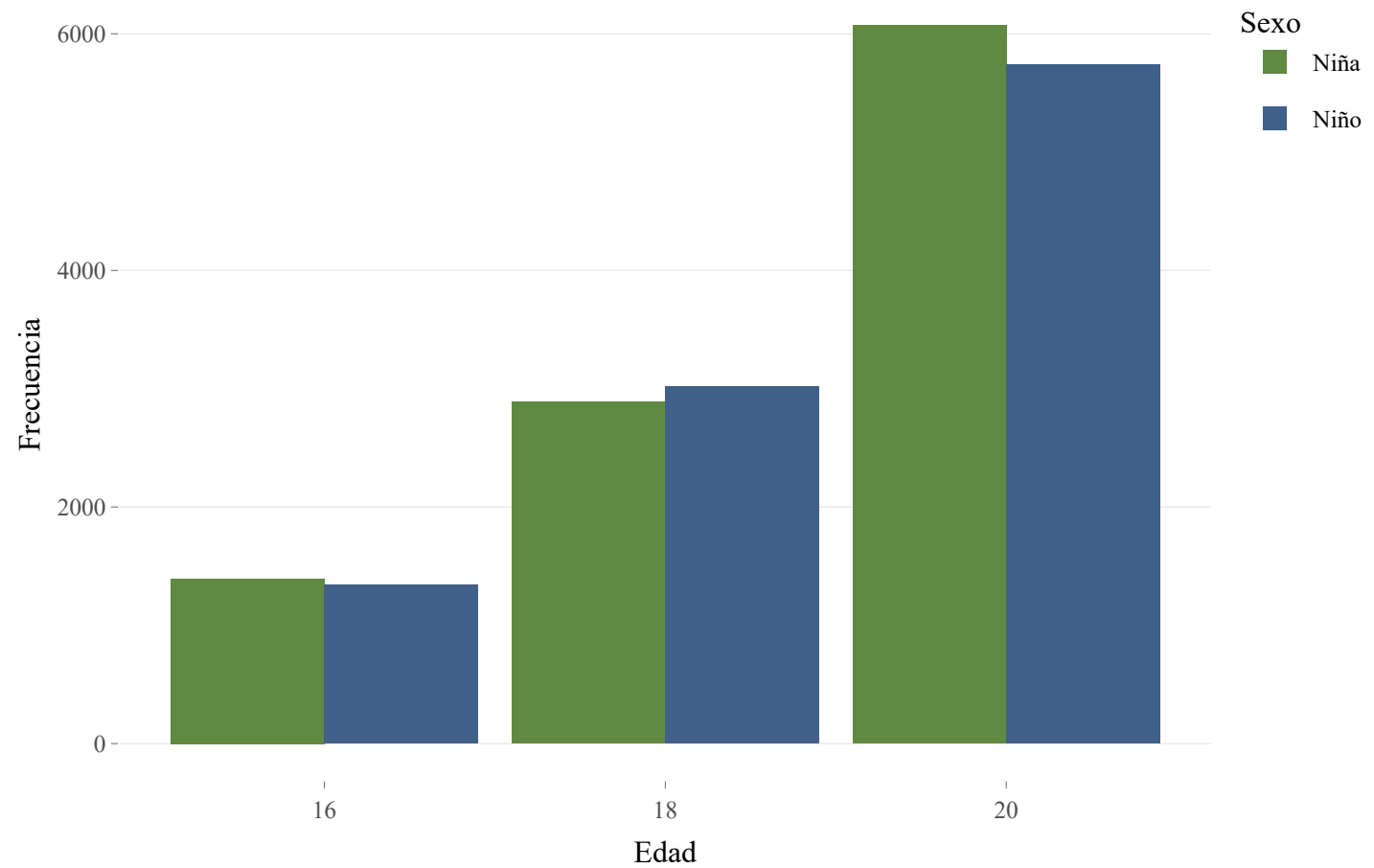
Número de palabras pronunciadas por bebés



Modificando la base de datos

```
miplotly_mod <- freq %>%  
  ungroup() %>%  
  mutate(Sexo = factor(gender, labels = c("Niña", "Niño")),  
         Edad = factor(age)) %>%  
  select(Edad, Sexo, n) %>%  
  ggplot(aes(x = Edad, y = n, fill = Sexo)) +  
    geom_col(position = "dodge") +  
    scale_fill_manual(values = c("#608941", "#416089")) +  
    labs(title = "Número de palabras pronunciadas por bebés",  
         subtitle = "según edad (meses) y sexo",  
         x = "Edad",  
         y = "Frecuencia") +  
    theme_hc()  
  
ggplotly(miplotly_mod)
```

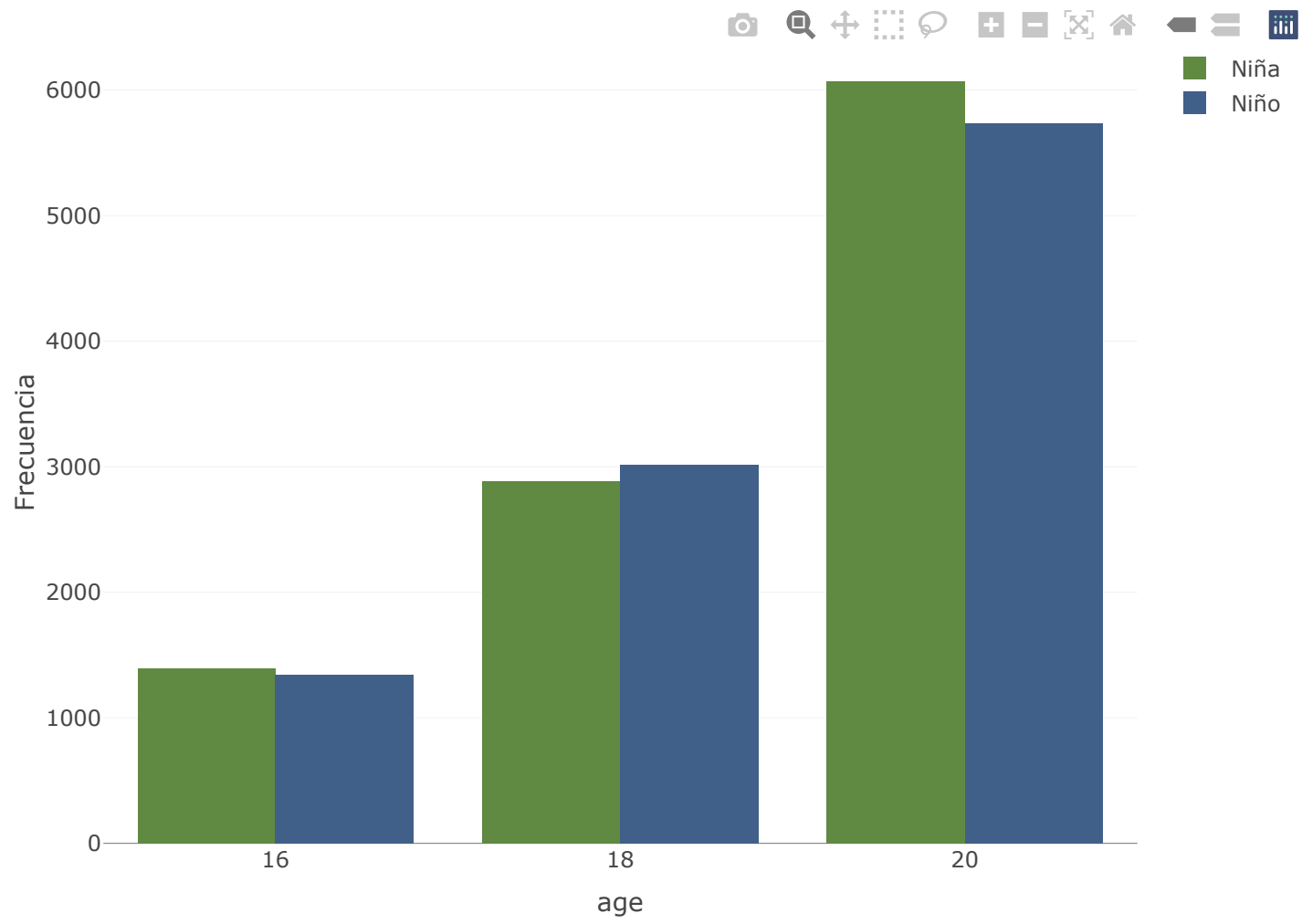
Número de palabras pronunciadas por bebés



Mediante comandos Plotly

```
library(tidyr)
freq_plotly <- freq %>%
  pivot_wider(names_from = "gender", values_from = "n")

fig <- plot_ly(freq_plotly, x = ~ age, y = ~female, type = 'bar', name = 'Niña')
fig <- fig %>% add_trace(y = ~male, name = 'Niño')
fig <- fig %>% layout(yaxis = list(title = 'Frecuencia'), barmode = 'group')
```



Guardar los gráficos plotly


```
# install.packages("htmlwidgets")  
htmlwidgets::saveWidget(fig, "img/grafico_plotly.html")
```





Generación de informes

Rmarkdown

La librería **rmarkdown** es una herramienta para la creación de documentos dinámicos y reproducibles que combinan código, resultados renderizados (como figuras) y texto.



Outputs Rmarkdown

Los documentos de R Markdown se pueden tener muchos formatos de salida, incluidos documentos HTML , PDF , Word , presentaciones de diapositivas  entre otros formatos. Esto permite centrarse en el contenido mientras R Markdown se encarga de su formato.

¿Cómo funciona Rmarkdown?

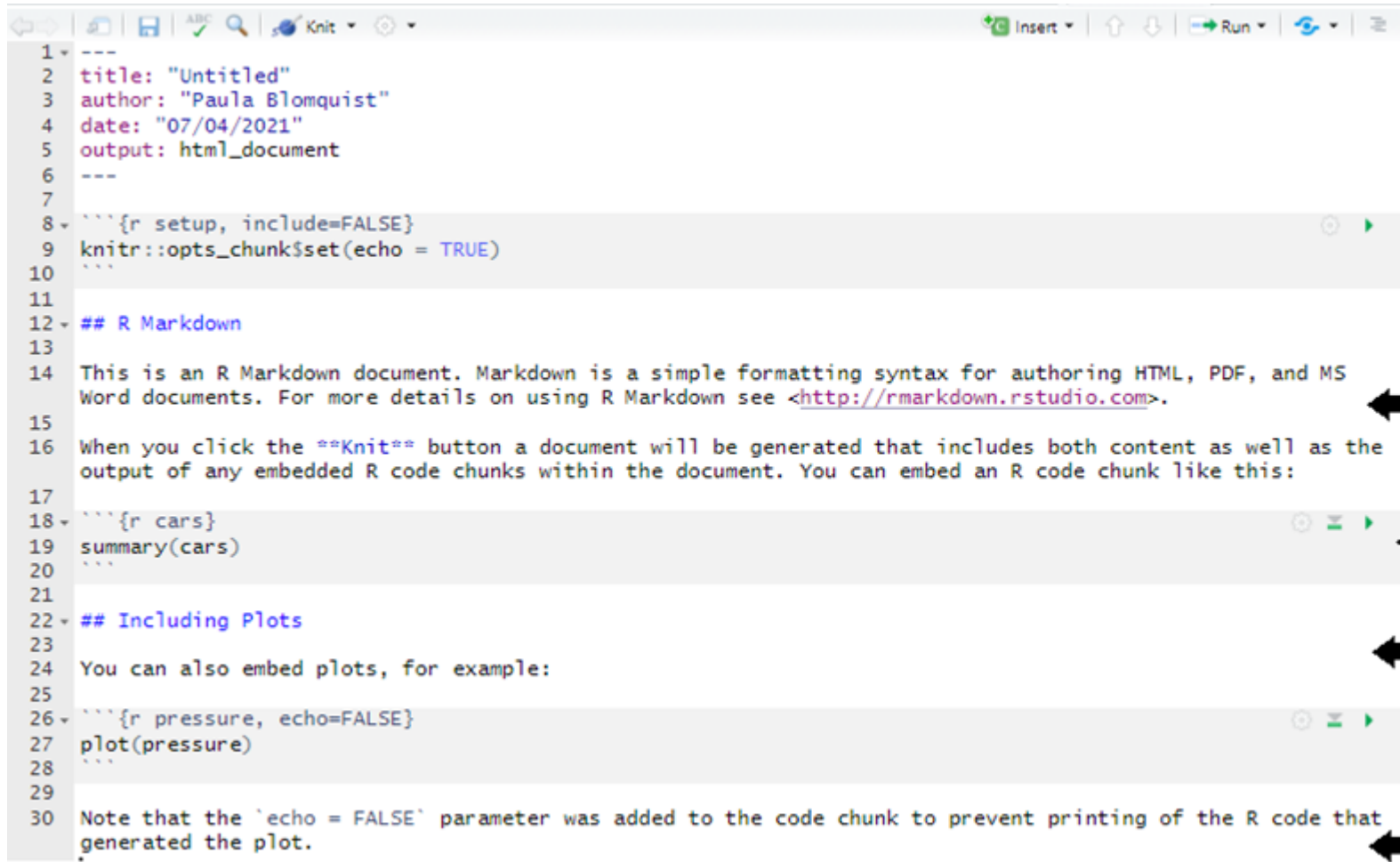
- **Markdown**: Es el lenguaje que permite escribir documentos en texto plano y que posteriormente será convertidos a otros formatos. No es exclusivo de R. Los ficheros *markdown* tienen extensión **.md**.
- **Rmarkdown**: Es una variación del anterior exclusivo para R. Permite el *incrustado* de código R. Los ficheros *rmarkdown* tienen extensión ****.Rmd**
- **La librería rmarkdown**: Contiene el código para convetir (renderizar) el fichero .Rmd en el formato deseado.
- **Knitr**: Es la librería encargada de pasar el fichero .Rmd en un fichero .md. Ejecuta los códigos.
- **Pandoc**: Es el software encargado de '*traducir*' el fichero .md en el documento con el formato requerido: Word, PDF, HTML, etc...



Creación de un fichero Rmarkdown

1. A través de los menús *File > New File > R Markdown*
2. En la ventana de configuración del fichero se le añadiría:
 - Título del documento (*opcional pero recomendado*).
 - Autor del documento (*opcional pero recomendado*).
 - Formato de salida.

Componentes fichero Rmarkdown



The image shows a screenshot of an R Markdown file in a code editor. The file content is as follows:

```
1 ---
2 title: "Untitled"
3 author: "Paula Blomquist"
4 date: "07/04/2021"
5 output: html_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
15 Word documents. 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
18 output of any 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
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
31
32 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that
33 generated the plot.
```

Annotations on the right side of the image point to specific parts of the code:

- YAML**: Points to the YAML front matter (lines 1-6).
- Code chunk**: Points to the first R code chunk (lines 8-10).
- markdown text**: Points to the main body text (lines 12-17).
- Code chunk**: Points to the second R code chunk (lines 18-22).
- markdown text**: Points to the text following the second code chunk (lines 24-26).
- Code chunk**: Points to the third R code chunk (lines 27-30).
- markdown text**: Points to the final text block (lines 32-33).

R Markdown script

```
1 ---  
2 title: "Outbreak Situation Report"  
3 date: "4/24/2021"  
4 output: word_document  
5 ---  
6  
7 ```{r setup, echo=FALSE}  
8 pacman::p_load(rio, here, tidyverse, janitor, incidence2, flextable)  
9 linelist <- rio::import(here::here("data", "case_linelists", "linelist_cleaned.rds"))  
10 ```  
11  
12 This report is for the Incident Command team of the fictional outbreak of Ebola cases.  
13 **As of `r format(max(linelist$date_hospitalisation, na.rm=T), "%d %B")` there have  
14 been `r nrow(linelist)` cases reported as hospitalized.**  
15  
16 ## Summary table of cases by hospital  
17  
18 ```{r, echo=F, out.height="75%"  
19 linelist %>%  
20 filter(!is.na(hospital)) %>%  
21 group_by(hospital) %>%  
22 summarise(cases = n(),  
23           deaths = sum(outcome == "Death", na.rm=T),  
24           recovered = sum(outcome == "Recover", na.rm=T)) %>%  
25 adorn_totals() %>%  
26 qflextable()  
27 ```  
28  
29 ## Epidemic curve by age  
30  
31 ```{r, echo=F, warning=F, message=F, out.height = "75%", out.width="100%"  
32 # create epicurve  
33 age_outbreak <- incidence(  
34   linelist,  
35   date_index = date_onset, # date of onset for x-axis  
36   interval = "week", # weekly aggregation of cases  
37   groups = age_cat)  
38  
39 # plot  
40 plot(age_outbreak, n_breaks = 3, fill = age_cat, col_pal = muted, title = "Epidemic  
curve by age group")  
41 ```
```

YAML sets title, date,
and output type

Code chunk loads
packages and data

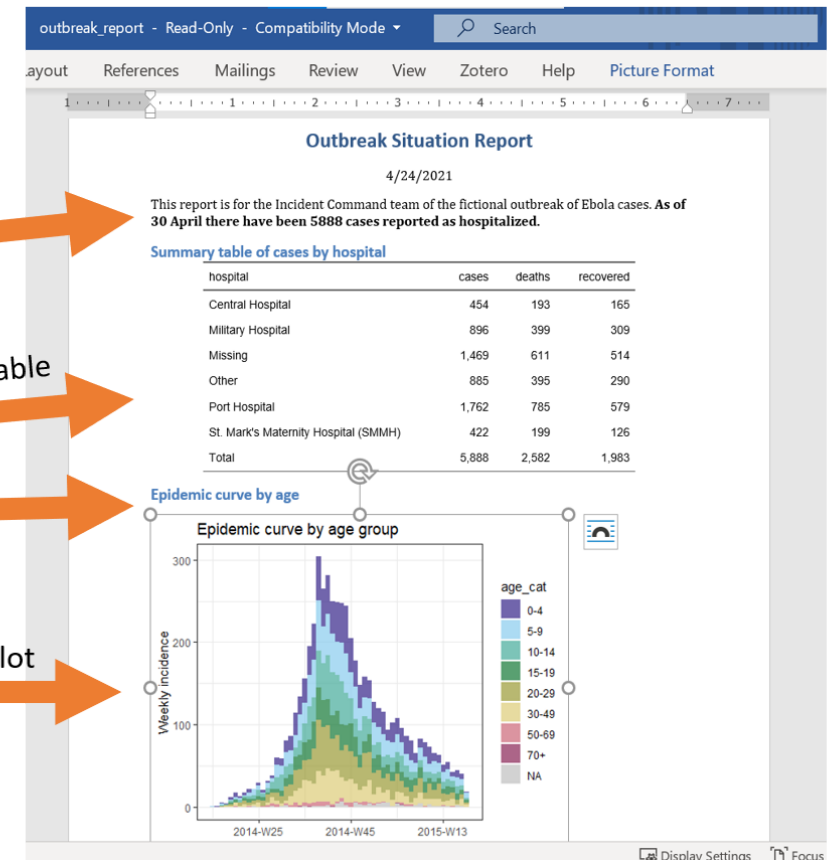
Text and in-line code

Code chunk makes table

Headings

Code chunk makes plot

Output (e.g. Word document)



Fuente: The Epidemiologist R Handbook

Markdown

Encabezados

Una almohadilla # antes del texto indica que se trata de un encabezado. Los niveles del encabezado se determinan por el número de almohadillas.

```
# Encabezado de nivel 1
```

```
## Encabezado de nivel 2
```

```
### Encabezado de nivel 3
```

Encabezado de nivel 1

Encabezado de nivel 2

Encabezado de nivel 3

Markdown

Listas no ordenadas

En este caso se precede del texto con *, + o -. Si queremos un subnivel debemos introducir 4 espacios o un tabulador antes del *, +, o -.

-
- Lista no ordenada
 - Segundo elemento de la lista
 - Subnivel del segundo elemento
-

- Lista no ordenada
- Segundo elemento de la lista
 - Subnivel del segundo elemento

Markdown

Listas ordenadas

En este caso se precede del texto con un número y . o). Si queremos un subnivel debemos introducir 4 espacios o un tabulador.

-
1. Lista ordenada
 2. Segundo elemento de la lista
 - 2.1. Subnivel del segundo elemento
-

1. Lista no ordenada
2. Segundo elemento de la lista
 - 2.1. Subnivel del segundo elemento

Markdown

Enlaces

[texto](enlace)

[Escuela de Servicios Sanitarios y Sociales de Canarias (ESSSCAN)](https://www.essscan.es)

[Escuela de Servicios Sanitarios y Sociales de Canarias \(ESSSCAN\)](https://www.essscan.es)

Markdown

Imágenes

`![texto sustitutivo](enlace)`

`![Escuela de Servicios Sanitarios y Sociales de Canarias (ESSSCAN)](https://www.essscan.es/php/wp-content/uploads/Logo_ESSSCAN_opt.jpg)`



Markdown

Fomato de texto


- *Itálica*. *Itálica* O _Itálica_
- **Negrita**. **Negrita**
- ***Negrita e itálica***. ***Negrita e itálica***

RMarkdown

Código R

- Bloque de código

```
```${r}  
Bloque de código
```

- Ctrl + Alt + i
- Mediante el icono  en la barra del editor

```

title: "Apuntes del curso de R"
author: "Christian González & Jaime Pinilla"
date: "2022-11-02"
output: word_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

## Día 1

* Introducción
* Principios básicos de R **Importante:** Crear pro
  * Instalación
  * Vectores
  * Etc...

## Día 2
* Importar

```{r}
numberofwords <- readr::read_csv("../Modulo 2/numbe
head(numberofwords)

```

```

Día 3
* Graficar

```{r}
library(ggplot2)
library(dplyr)
library(ggthemes)
numberofwords %>%
  ggplot(aes(x = id, y = words)) +
  geom_point(colour = "black", fill = "#0073C2FF",
    labs(title = "Número de palabras pronunciadas por
      subtitle = "Fichero: numberofwords.csv",
        x = "Id. del bebé",
        y = "Frecuencia") +
  theme_hc()

```

```

[Transparencias](https://github.com/chrglez/slidesc



# Parámetros en los bloques de código

- El primer parámetro es el lenguaje del código. En nuestro caso `r`
- El segundo es opcional y es el nombre del bloque
- `eval` = `FALSE` no ejecuta el código
- `echo` = `FALSE` no muestra el código
- `result` = `FALSE` no muestra el resultado
- `warning` = `FALSE` no muestra los warnings que se produzcan al ejecutar el código
- `message` = `FALSE` no muestra los mensajes que se produzcan al ejecutar el código
- `include` = `FALSE` equivale a `echo` = `FALSE`, `result` = `FALSE`, `warning` = `FALSE`, `message` = `FALSE`
- `fig.height`, `fig.width` alto y ancho de las figuras en caso de que el código las genere

# Tablas

La salidas de las tablas proporcionadas por R para el documento se producen sin formato, como si fuera a consola. Son poco amigables. Para dar formato existen diefrerentes librerías.

- [xtable](#)
- [stargazer](#)
- [pander](#)
- [tables](#)
- [knitr](#) con su complemento [kableExtra](#)

---

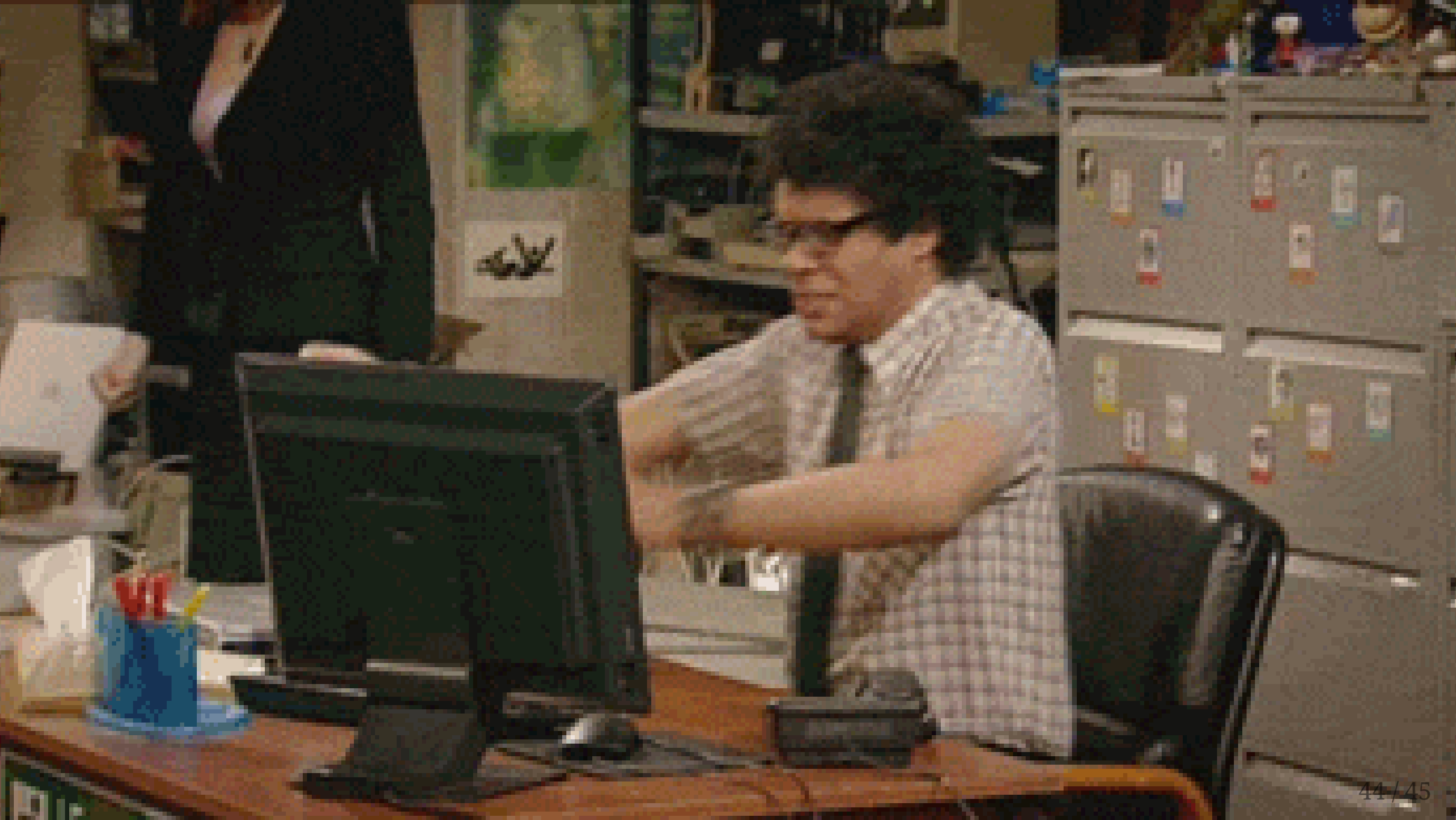
```
```{r}
knitr::kable(head(numberofwords))
```
```

# Trabajar con PDF

- Se necesita trabajar con un compilador [LaTeX](#).
- La forma más sencilla de instalar ese compilador es a través de la librería [tinytex](#)

```
install.packages('tinytex')
tinytex::install_tinytex()
```

- Ya se podría usar el formato de salida PDF cambiando `output: pdf_document` en el YALM.



A close-up, low-angle shot of Din Djarin in his Mandalorian armor, looking slightly to the right. The armor is dark and metallic, with a helmet that has a silver-colored faceplate. The background is dark and out of focus, suggesting an indoor setting with some light sources.

THIS IS THE WAY

