

# Clase 9

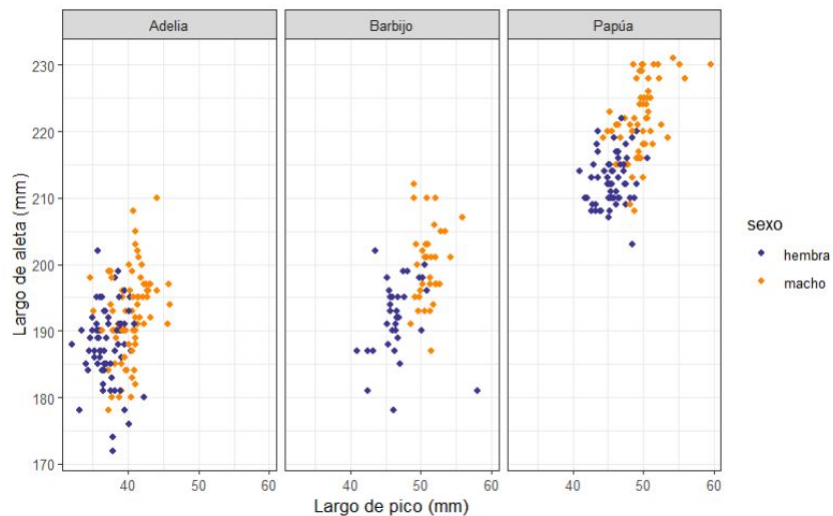
**unab**

VISUALIZACIÓN DE LA INFORMACIÓN 2024

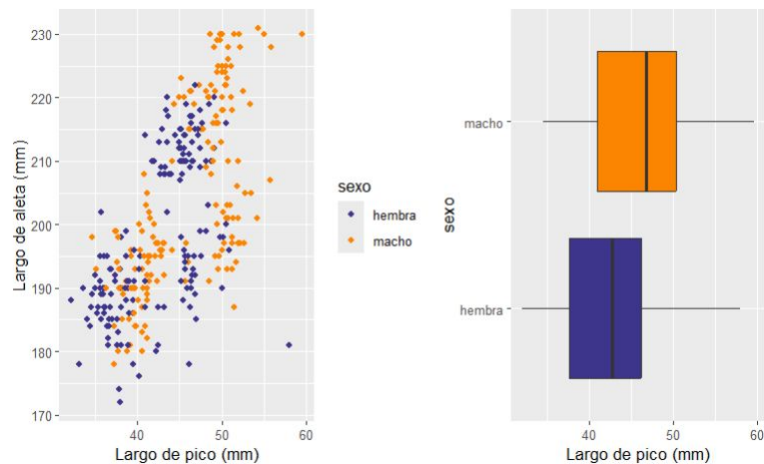
# Múltiples gráficos (compuestos)

# Múltiples gráficos: dos escenarios comunes

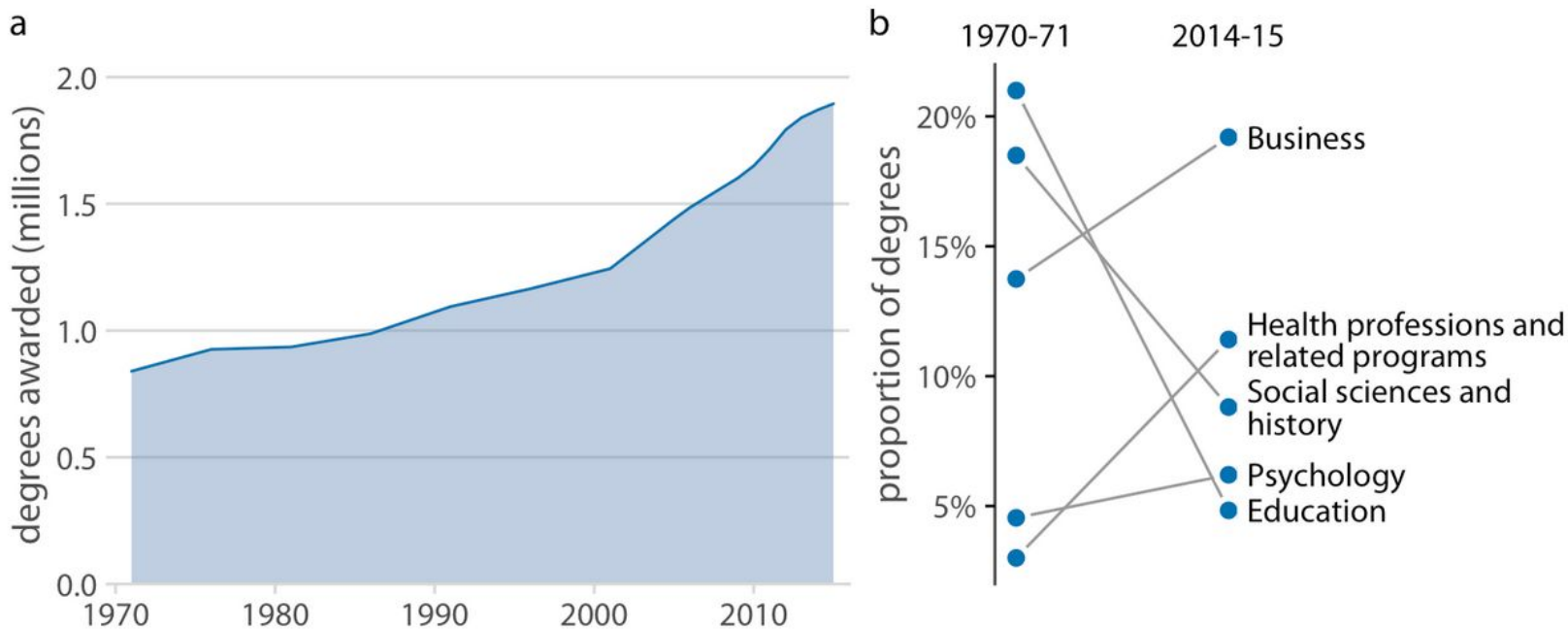
1. El mismo tipo de gráfico se repite muchas veces.



2. Varios gráficos diferentes se combinan en una imagen



# Combinar figuras diferentes en una sola vista

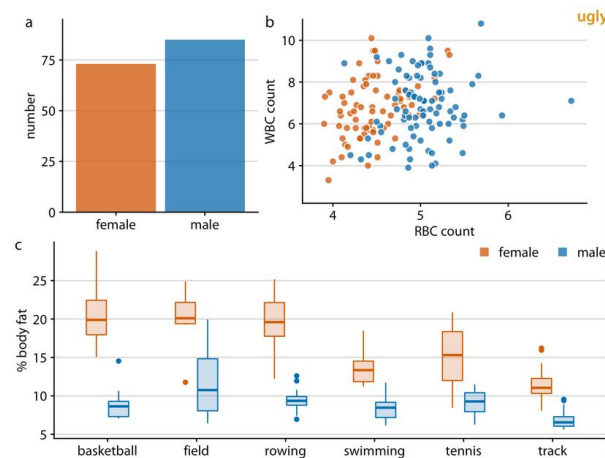
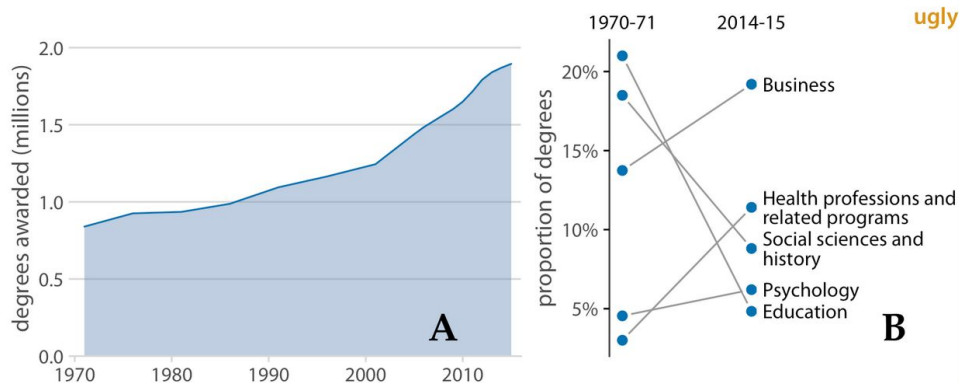


Note: Ahora tenemos dos etiquetas ("a", "b") para identificar cada gráfico.

# Algunas recomendaciones

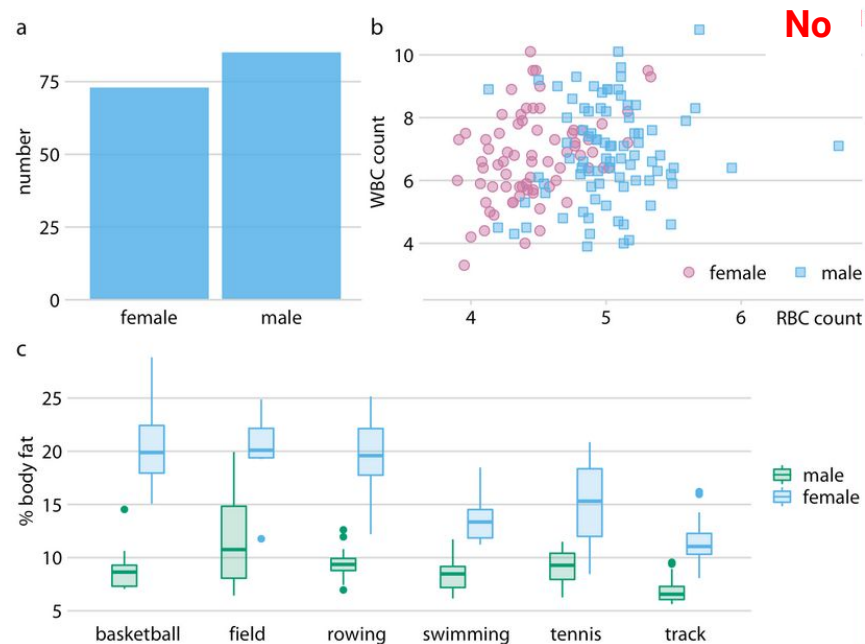
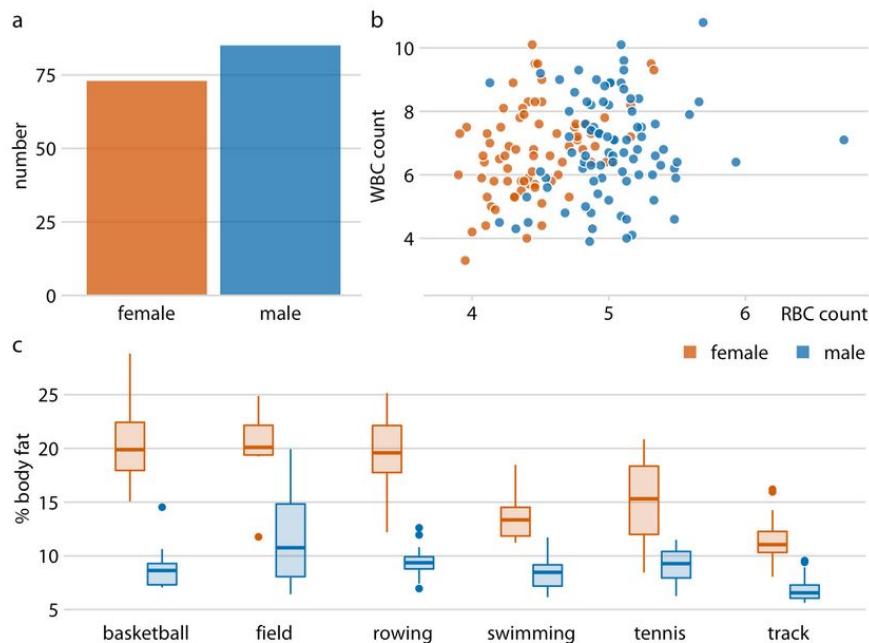
No usar etiquetas demasiado grandes o que destaquen de otra manera

Preste atención a la alineación de los subgráficos

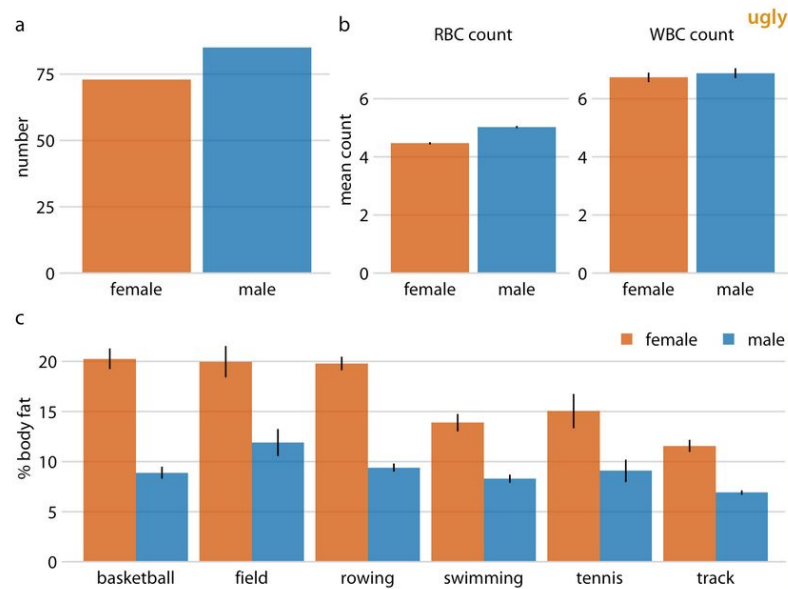
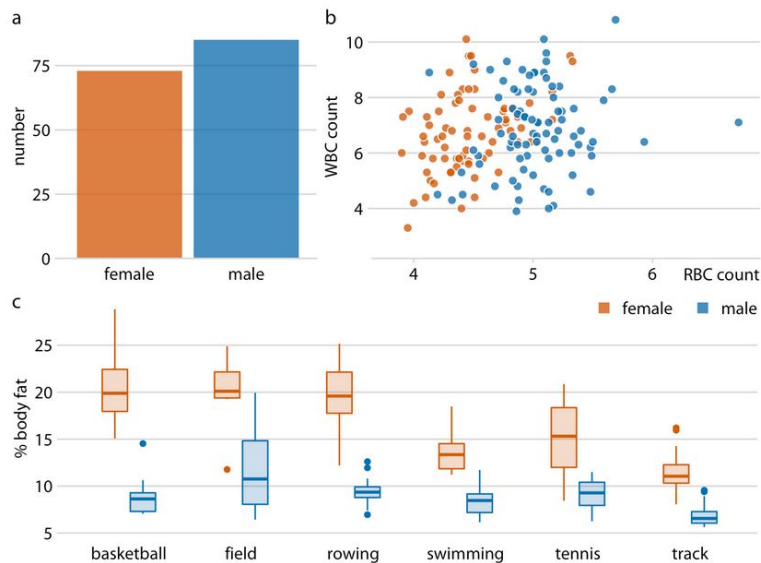


# Algunas recomendaciones

Utilizar un set de colores coherente entre los subgráficos



# Combinar gráficos de diferentes tipos



# Paquete patchwork - uno al lado del otro



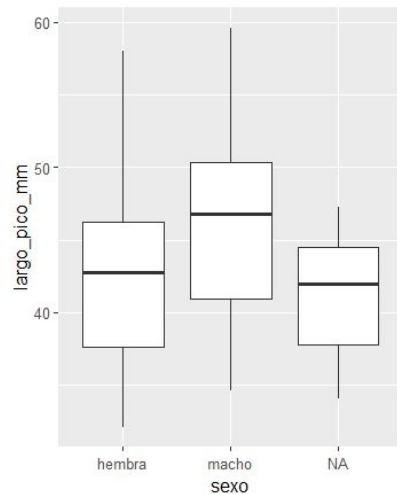
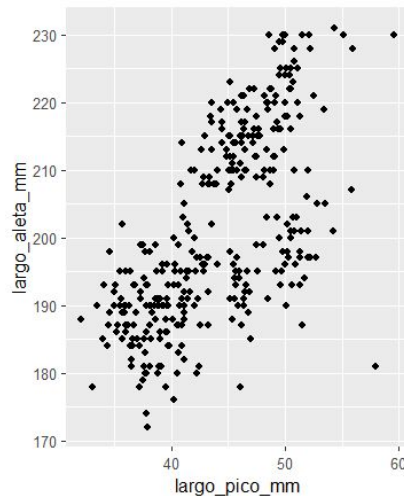
```
g1 <- ggplot(pinguinos) +
```

```
  geom_point(aes(largo_pico_mm,  
                 largo_aleta_mm))
```

```
g2 <- ggplot(pinguinos) +
```

```
  geom_boxplot(aes(sexo,  
                  largo_pico_mm))
```

```
g1 | g2
```

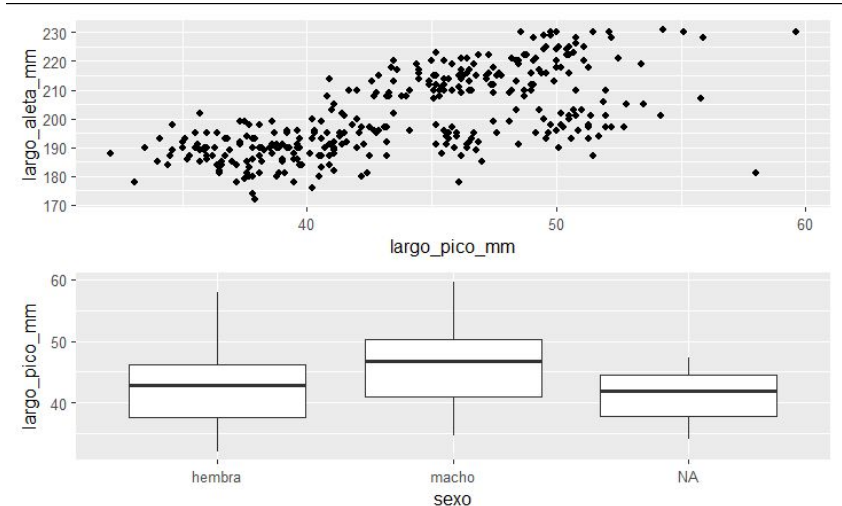




# Paquete patchwork - uno abajo del otro

```
g1 <- ggplot(pinguinos) +  
  
  geom_point(aes(largo_pico_mm,  
    largo_aleta_mm))  
  
g2 <- ggplot(pinguinos) +  
  
  geom_boxplot(aes(sexo,  
    largo_pico_mm))
```

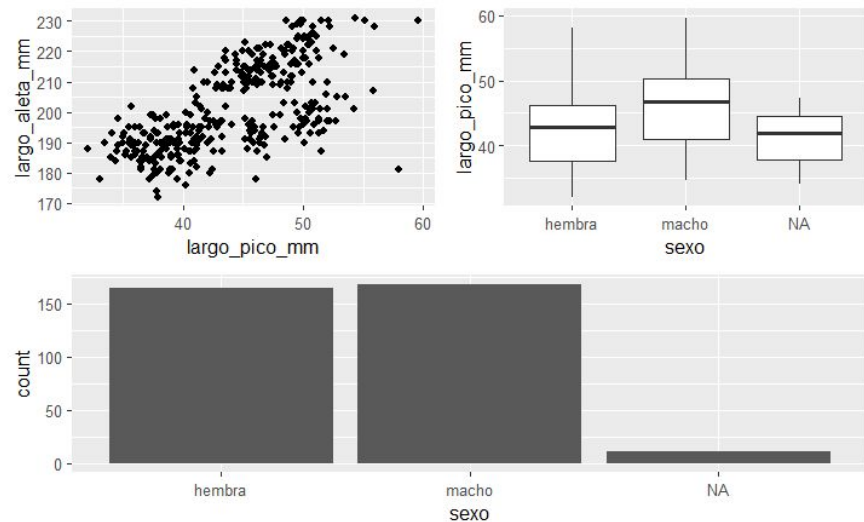
```
g1 / g2
```



# Paquete patchwork - más complejos

```
g1 <- ggplot(pinguinos) +  
  geom_point(aes(largo_pico_mm,  
                largo_aleta_mm))  
  
g2 <- ggplot(pinguinos) +  
  geom_boxplot(aes(sexo, largo_pico_mm))  
  
g3 <- ggplot(pinguinos) +  
  geom_bar(aes(sexo))
```

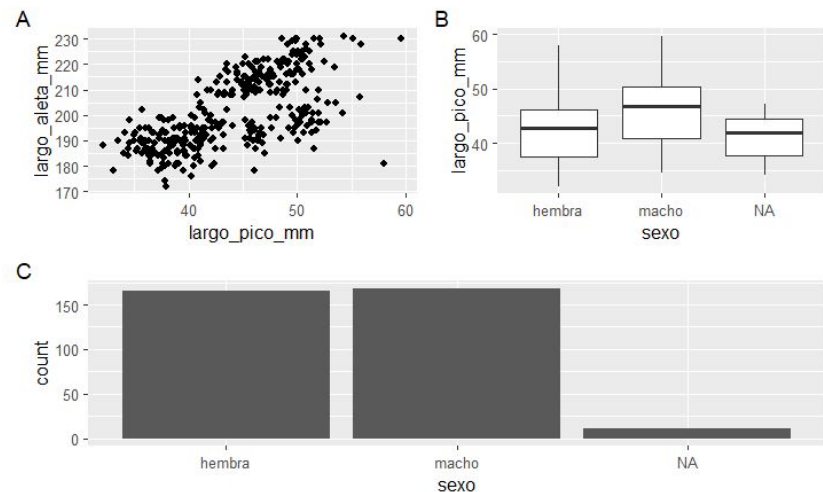
```
(g1 | g2) / g3
```



# Paquete patchwork - anotaciones

```
g1 <- ggplot(pinguinos) +  
  geom_point(aes(largo_pico_mm,  
                 largo_aleta_mm))  
  
g2 <- ggplot(pinguinos) +  
  geom_boxplot(aes(sexo, largo_pico_mm))  
  
g3 <- ggplot(pinguinos) +  
  geom_bar(aes(sexo))
```

```
(g1 | g2) / g3 +  
  plot_annotation(  
    tag_levels = "A"  
  )
```

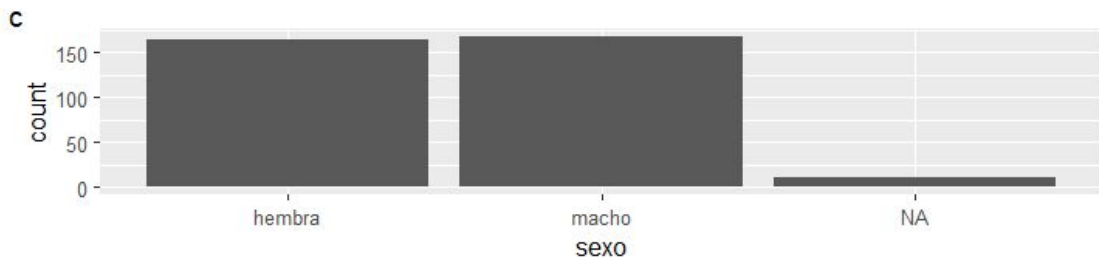
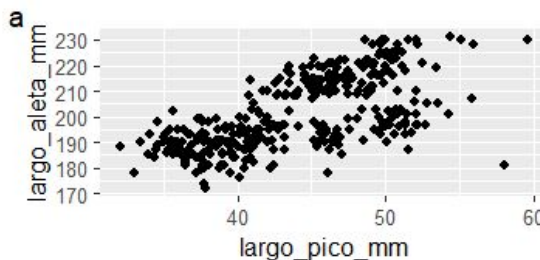


# Paquete patchwork - anotaciones

```
g1 <- ggplot(pinguinos) +  
  geom_point(aes(largo_pico_mm,  
                 largo_aleta_mm))  
  
g2 <- ggplot(pinguinos) +  
  geom_boxplot(aes(sexo, largo_pico_mm))  
  
g3 <- ggplot(pinguinos) +  
  geom_bar(aes(sexo))  
  
(g1 | g2) / g3 +  
  plot_annotation(  
    tag_levels = "a",  
    title = "Un gráfico de pingüinos",  
    subtitle = "Con un subtítulo...",  
    caption = "...y una nota al pié"  
  )
```

## Un gráfico de pingüinos

Con un subtítulo...

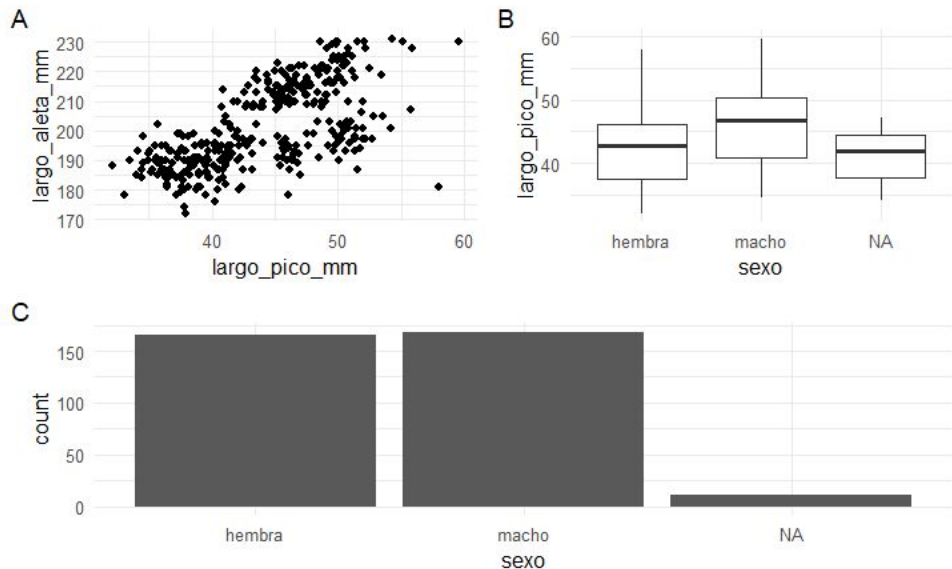


...y una nota al pié

# Paquete patchwork - mismo tema para todos

```
g1 <- ggplot(pinguinos) +  
  geom_point(aes(largo_pico_mm,  
                 largo_aleta_mm))  
  
g2 <- ggplot(pinguinos) +  
  geom_boxplot(aes(sexo, largo_pico_mm))  
  
g3 <- ggplot(pinguinos) +  
  geom_bar(aes(sexo))
```

```
(g1 | g2) / g3 +  
  plot_annotation(  
    tag_levels = "A"  
  ) &  
  theme_minimal()
```



# **Pausa**

10 minutos

**Guardar figuras**

# ¿Por qué guardar una figura?

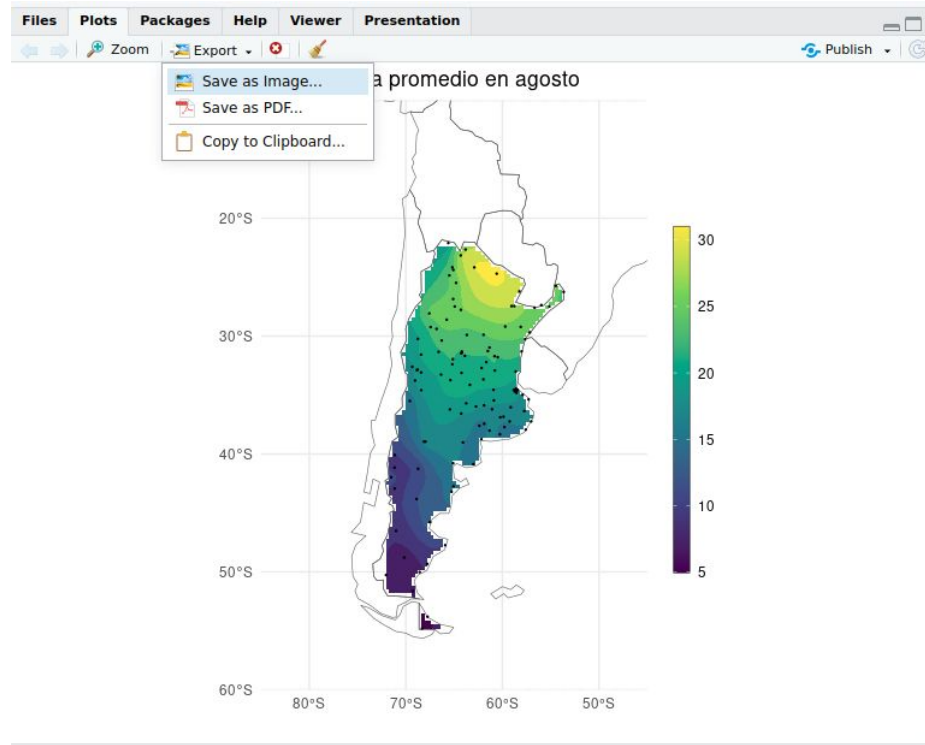
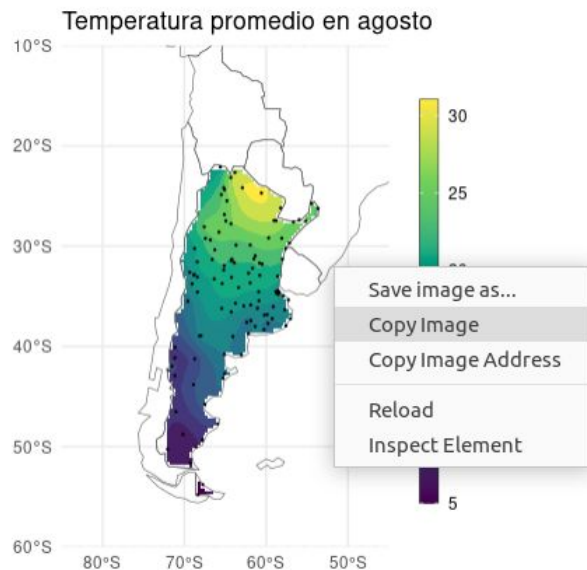
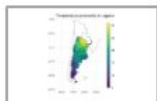
- ~~Para copiarla y pegarla en un documento de word~~
- Para compartir la visualización en redes sociales
- Para imprimir la visualización
- Para incorporar a un diseño más complejo
- Qué más?



# Que cosas NO hacer

```
theme_minimal() +  
theme()  
...
```

R Console



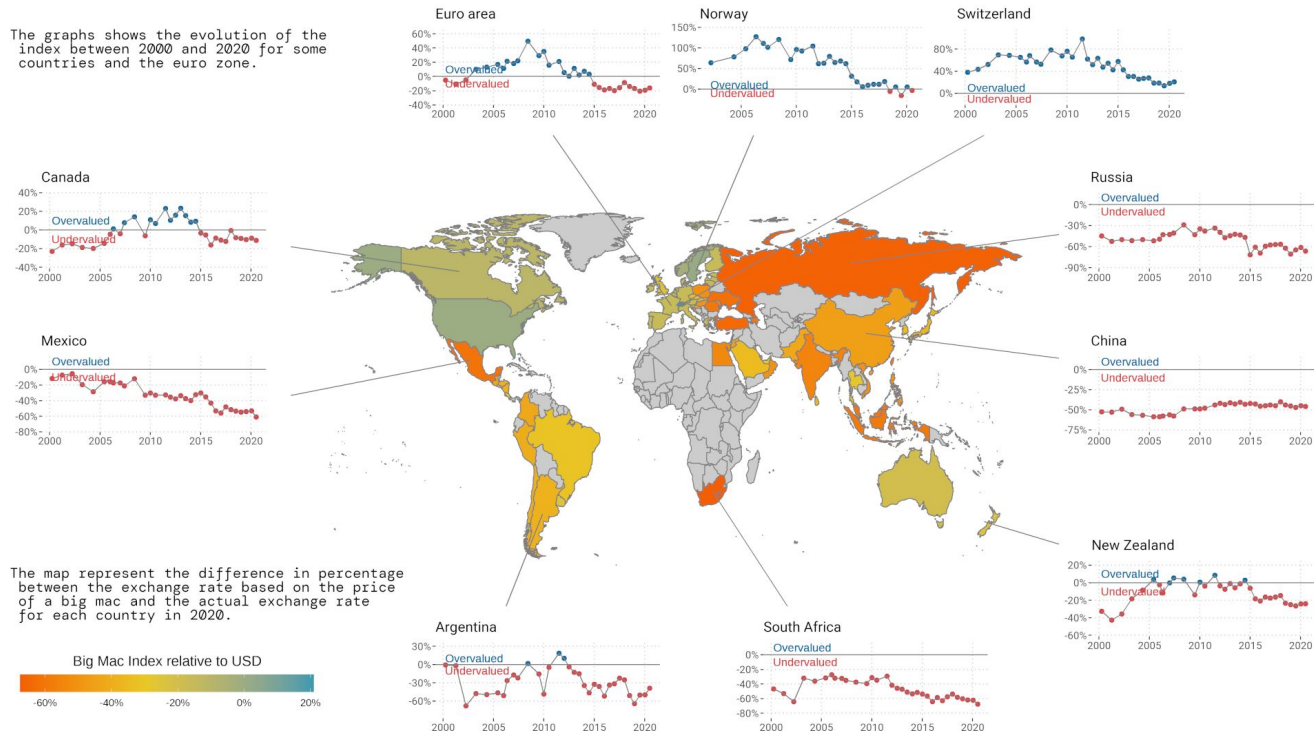
# ggsave()

<code>ggsave(</code>	
<code>  filename, _____</code>	nombre del archivo (+ ruta al archivo)
<code>  plot = last_plot(),</code>	por defecto guarda el último gráfico generado
<code>  device = NULL, _____</code>	png? jpg?
<code>  path = NULL,</code>	opcionalmente, la ruta del archivo puede ir acá
<code>  scale = 1, _____</code>	scale = 1,
<code>  width = NA,</code>	ancho
<code>  height = NA, _____</code>	alto
<code>  units = c("in", "cm", "mm", "px"),</code>	unidades para ancho y alto, usa in por defecto
<code>  dpi = 300, _____</code>	dpi es “dots per inch”, la resolución
<code>  limitsize = TRUE,</code>	Si es TRUE, máximo 50x50 inches
<code>  bg = NULL, _____</code>	fondo de la imagen, ej para png es transparente
<code>  ...</code>	...
<code>)</code>	

# A veces las cosas se ven bien...

## The Big Mac Index

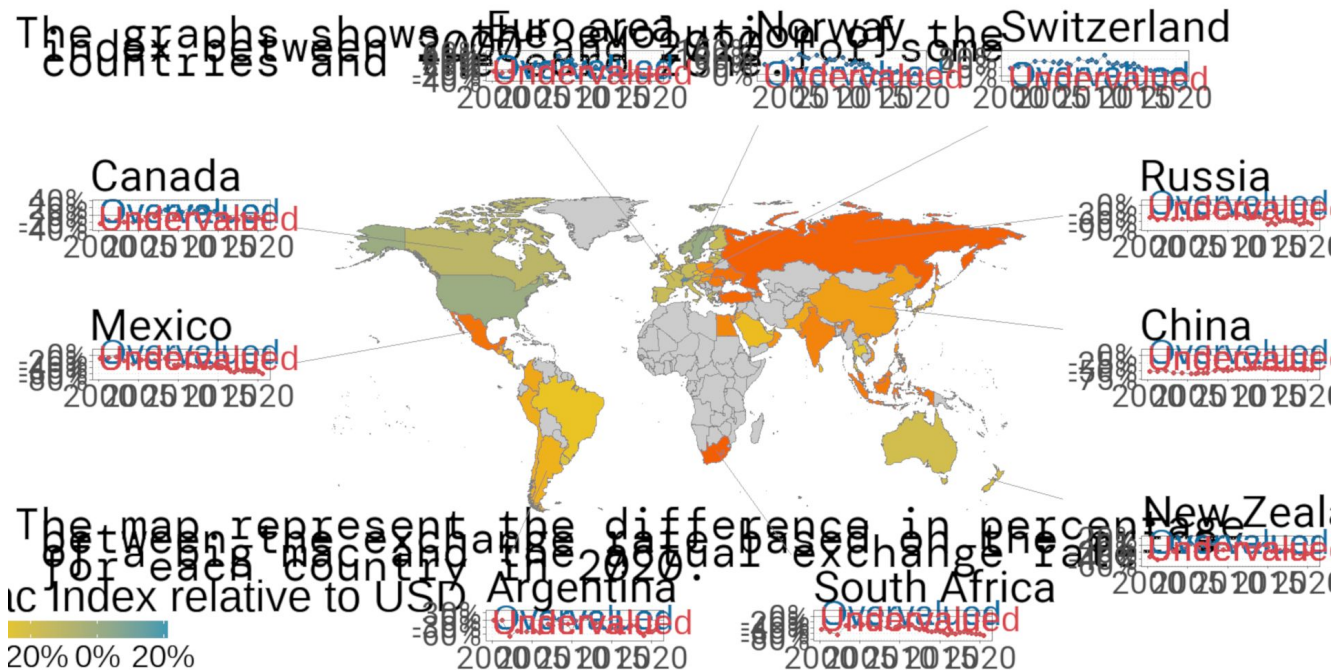
The graphs shows the evolution of the index between 2000 and 2020 for some countries and the euro zone.



Otras... no tanto

## The Big Mac Index

El tamaño que le damos al gráfico y el tamaño del texto que agregamos (`geom_text()`, `geom_label()`, etc) **no** van de la mano



Visualization: Christophe Nicault | Data: The Economist

# Algunas definiciones

**Tamaño absoluto (Absolute size):** Dimensiones físicas de la figura, en cm, in, etc.

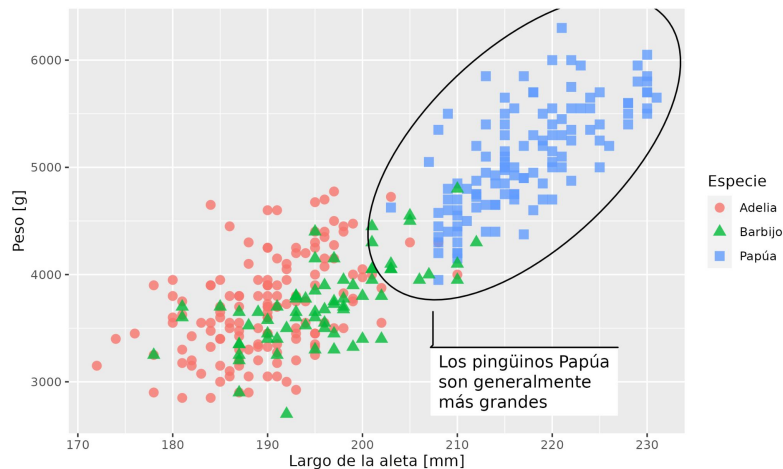
**Tamaño en pixeles (Pixel size):** para formatos raster (png, jpg), la figura es una matriz de pixeles, cada uno de un color. El tamaño en pixeles es el número de pixeles en ancho y alto. No representa un valor físico.

**Resolución (Resolution):** Combina el tamaño real con la cantidad de pixeles. Se describe en ppi (pixels per inch), o dpi (dots per inch). Una resolución de 72 ppi quiere decir que en una pulgada (inch) entran 72 pixeles.

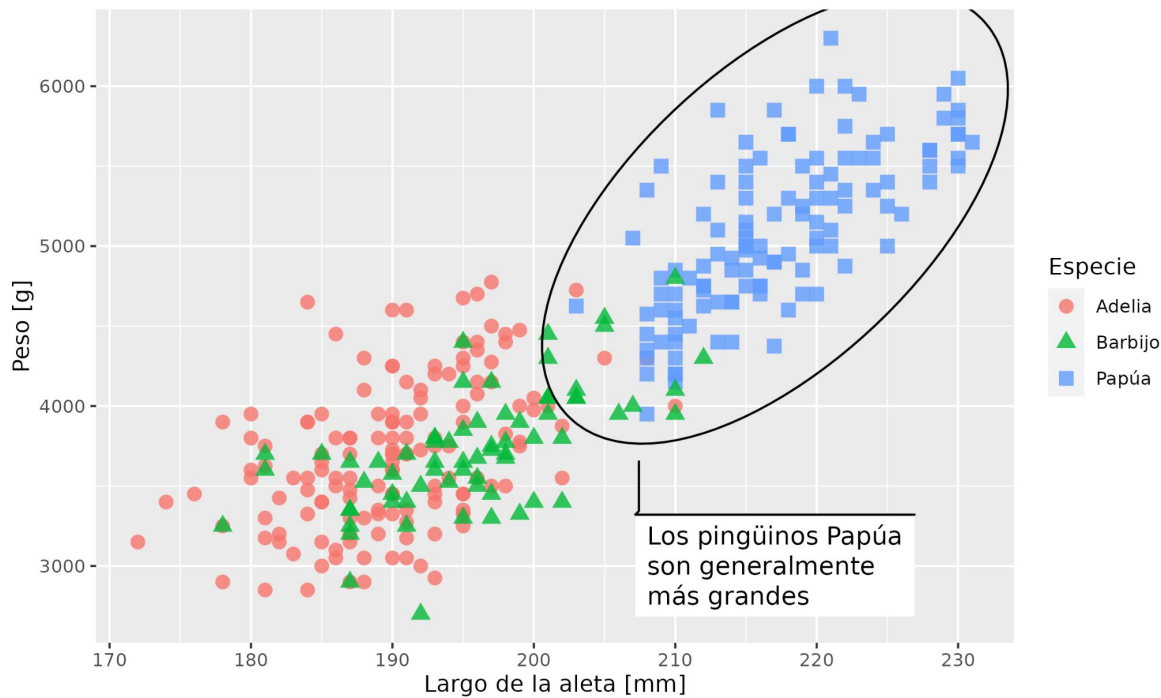
**Tamaño del punto (Pointsize):** Es la medida que representa el tamaño del texto. Por ejemplo cuando decimos Arial tamaño 12, estamos hablando de 12 puntos. En general, y en R, 1 punto =  $1/72$  inch.

```
p <- ggplot(na.omit(pinguinos), aes(x = largo_aleta_mm, y = masa_corporal_g)) +
  geom_point(
    aes(color = especie, shape = especie),
    size = 3,
    alpha = 0.8
  ) +
  ggforce::geom_mark_ellipse(
    aes(filter = especie == "Papúa",
        description = "Los pingüinos Papúa son generalmente más grandes")
  ) +
  labs(x = "Largo de la aleta [mm]",
       y = "Peso [g]",
       colour = "Especie",
       shape = "Especie")

ggsave("base.png", width = 20, height = 12,
       units = "cm", dpi = 300)
```



# El problema



Necesitamos guardar este gráfico para incluirlo en un poster (normalmente del tamaño de un afiche)

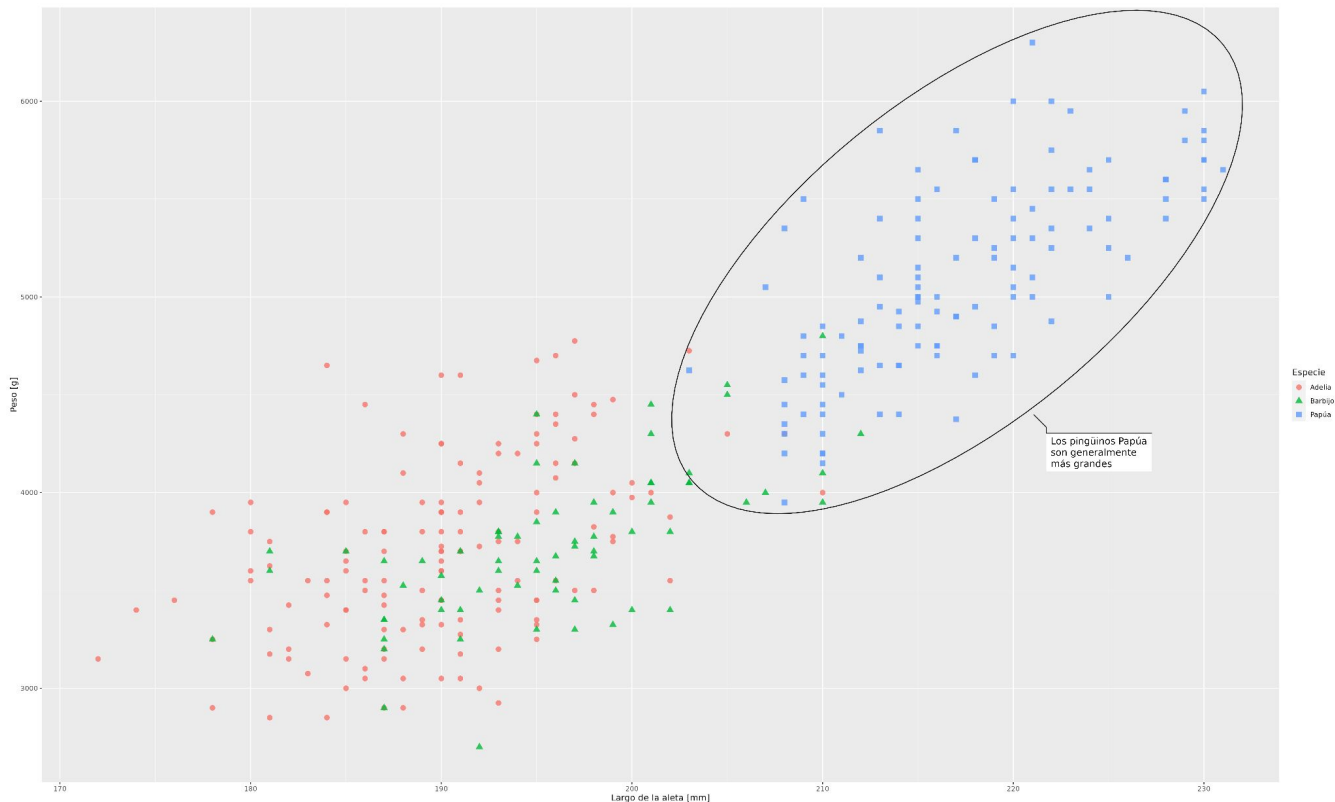
Se debe poder ver a 1,5 metros de distancia.

→ **Tiene que ser 3 veces más grande**

```
ggsave("base.png", width = 20, height = 12, units = "cm", dpi = 300)
```

```
ggsave("base.png", width = 20, height = 12, units = "cm", dpi = 300)
```

```
ggsave("base_grande.png", width = 60, height = 36, units = "cm", dpi = 300)
```



Algunos elementos del gráfico están asociados a tamaños **absolutos** y no cambian cuando cambiamos el tamaño de la figura al guardarla.

**Son aquellos expresados en puntos.**



# Soluciones

1. Usar gráficos de vectores, por ejemplo formato pdf

Pros: se puede agrandar o achicar la imagen infinitamente

Cons:

- La pinta del gráfico puede depender del programa con el que se lo mira.
- Si el gráfico es muy complejo, el archivo puede ser muy pesado.

# Soluciones

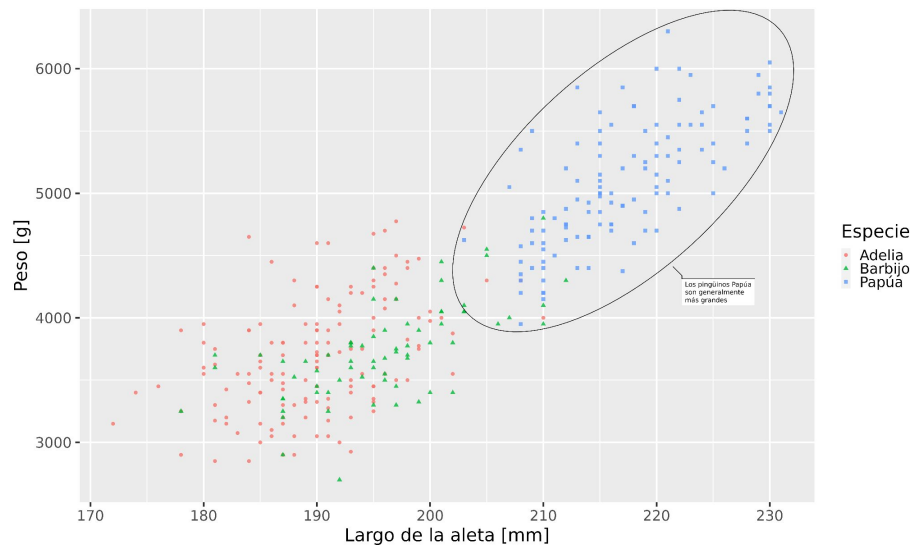
## 2. Cambiar el aspecto del gráfico con theme()

Pros: es simple

Cons: solo cambia el tamaño del texto intrínseco al gráfico

Se podría cambiar el tamaño de los puntos en `geom_point()` y el tamaño de la anotación en `geom_mark_ellipse()`

Pero si usamos más anotaciones, implica más cambios a mano, se vuelve un problema de nuevo.

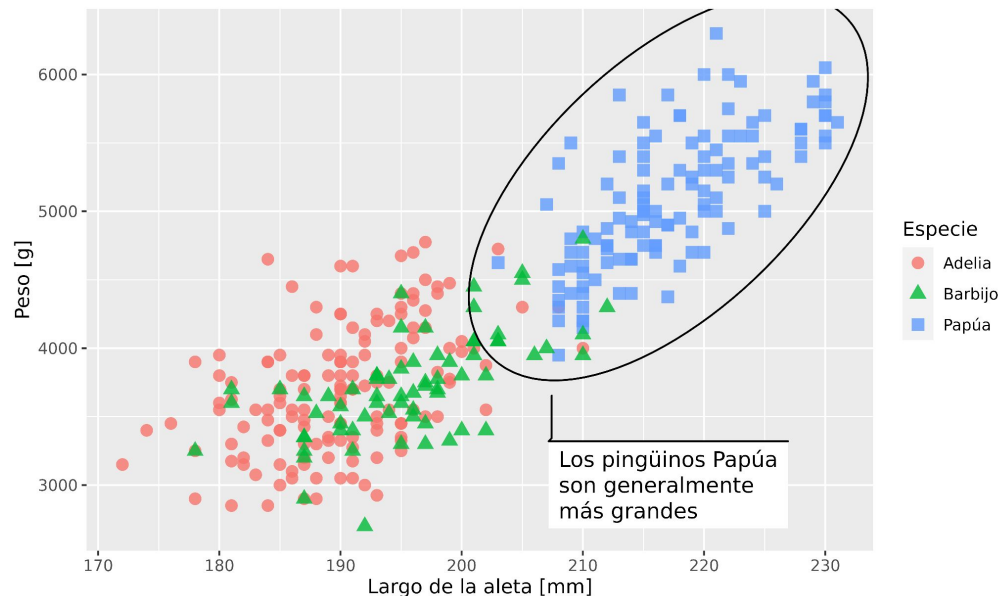


# Soluciones

## 3. Escalar la resolución usando siempre píxeles

Pros: resuelve el problema

Cons: hay que pasar todo a pixeles



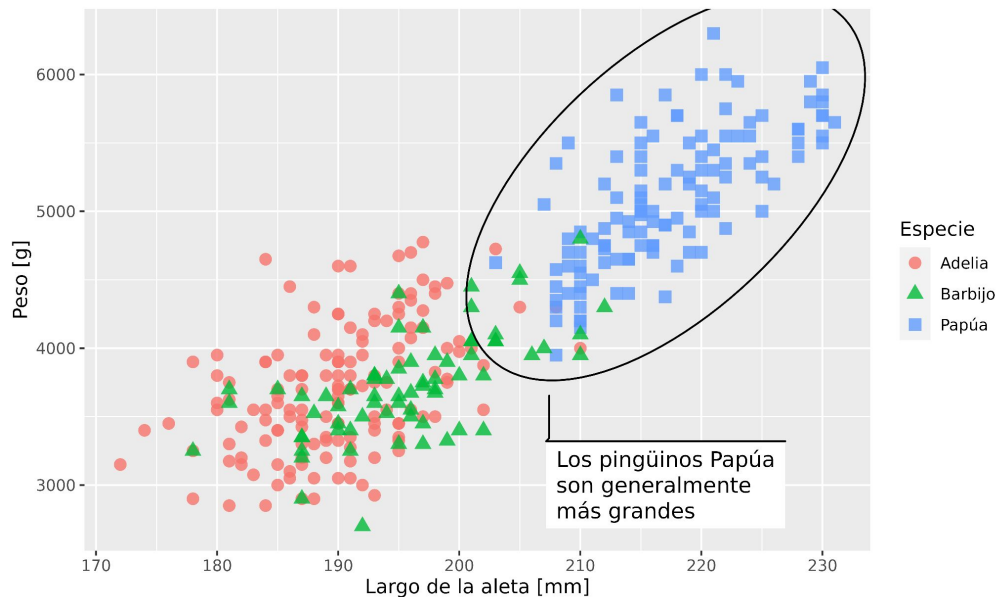
```
ggsave("base_grande_pixeles.png", width = 7087, height = 4252, units = "px", dpi = 300)
```

# Soluciones

4. usar el paquete ragg

Pros: resuelve el problema

Cons: más código involucrado



```
ragg::agg_png("base_ragg.png", width = 60, height = 36, units = "cm", res = 300, scaling = 3)
plot(p)
invisible(dev.off())
```

## En resumen...

- Hay que pensar donde va a vivir la figura
- Cual es el tamaño físico real
- Imagen tipo raster o vectorial

# Encuesta Feedback

No te olvides de completar la encuesta si tenés algún comentario :)

<https://forms.gle/Nq7SZGLJjCyzH5rD6>