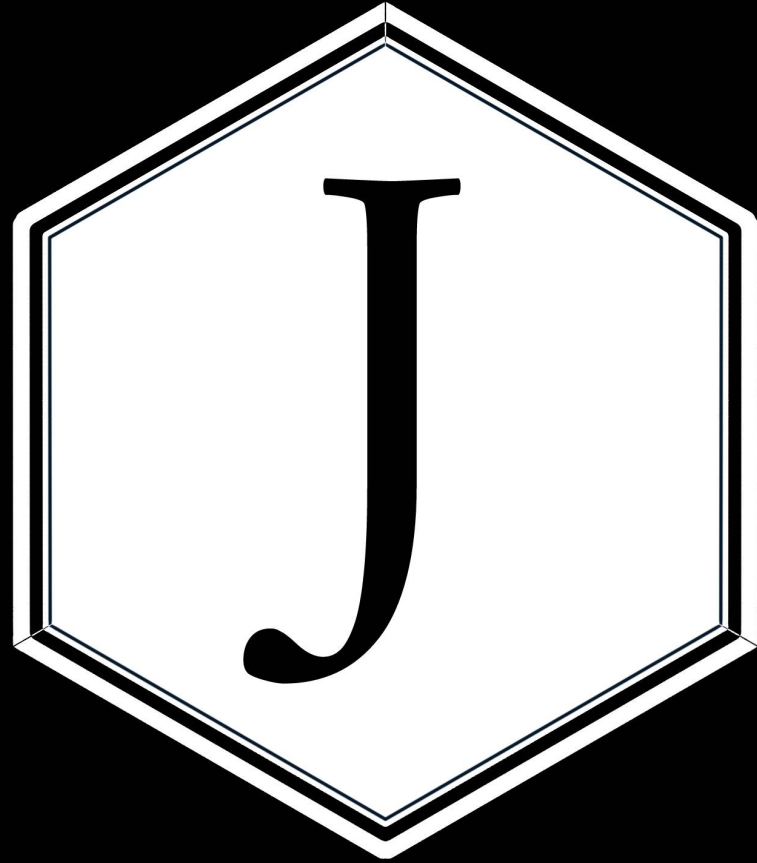


# Tidyverse

## GGplot2

Jhon

Fecha: 2020-07-27



# GGplot2

## (Grammar of Graphics)plot2

```
library(ggplot2)
```

## Elementos principales

Elemento	Descripcion
Data	La base de datos a graficar
aesthetics	Elementos presente en la grafica
geommetries	Elemtnos visuales de la grafica

## Todos los Elementos

Elemento	Descripcion
Data	La base de datos a graficar
aesthetics	Elementos presente en la grafica
geommetries	Elemtnos visuales de la grafica
facets	Pequenos graficos
statistics	Estadisticos
coordinates	Espacio donde se grafica
themes	Estetica de las figuras

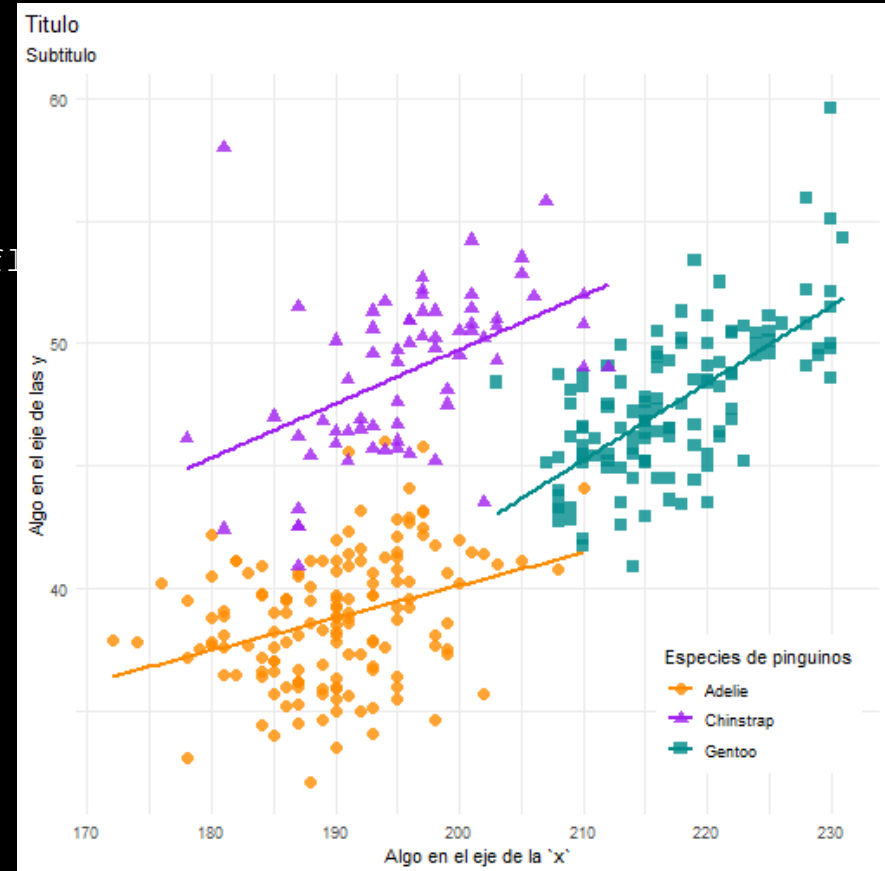
Todos estos elementos pueden ser agregados con `+`, y algunos de ellos pueden estar dentro de otros elementos, el formato basico se puede describir de la siguiente manera y puede ser reasignado como un objeto

# Los datos

```
# install.packages("palmerpenguins")  
library(palmerpenguins) # Cargar la libreria  
head(penguins)
```

```
## # A tibble: 6 x 8  
##   species island bill_length_mm bill_depth_mm flipper_length_mm  
##   <fct>    <fct>          <dbl>         <dbl>         <dbl>  
## 1 Adelie   Torge~           39.1           18.7           181  
## 2 Adelie   Torge~           39.5           17.4           181  
## 3 Adelie   Torge~           40.3           18           181  
## 4 Adelie   Torge~           NA            NA            NA  
## 5 Adelie   Torge~           36.7           19.3           181  
## 6 Adelie   Torge~           39.3           20.6           181  
## # ... with 1 more variable: year <int>
```

# Lo que se quiere



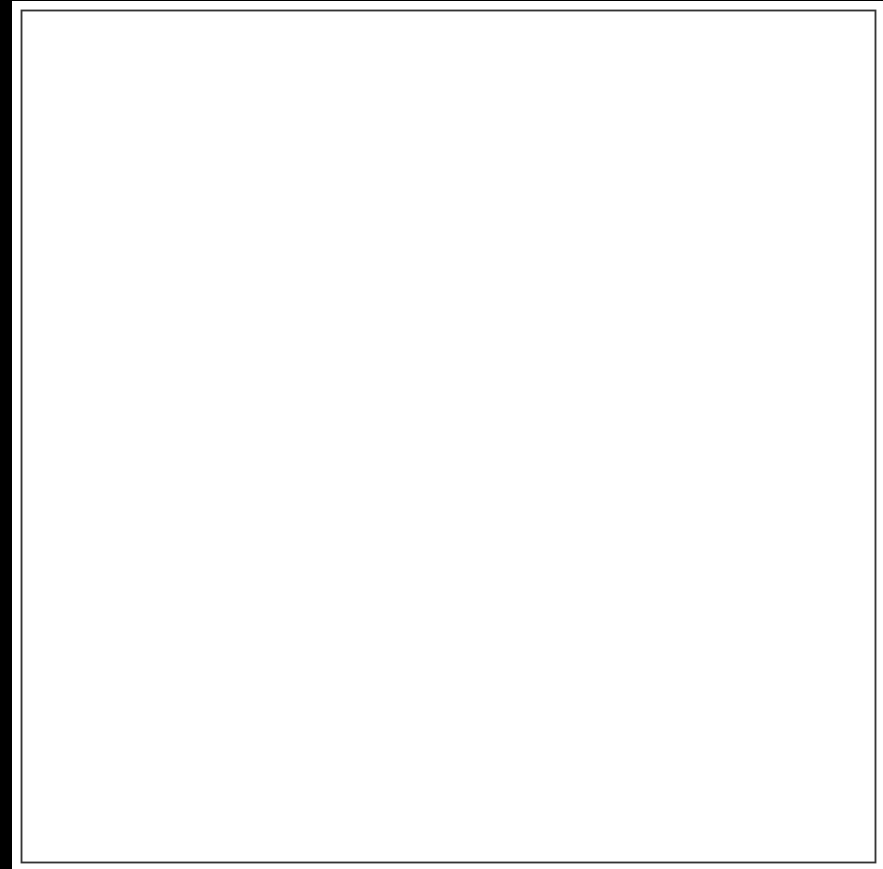
Exploremos las partes mencionadas

# Codigo generador

Primer elemento, la base de datos

```
ggplot(data = penguins)
```

# Resultado

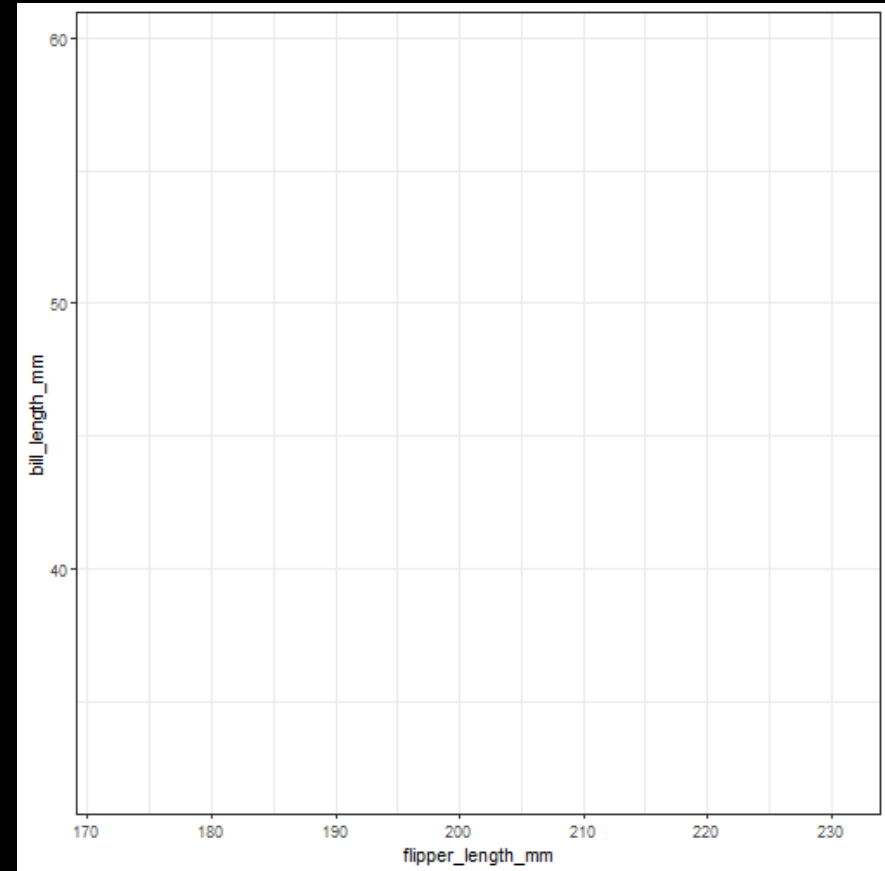


# Codigo generador

## Estetica

```
ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm)
```

# Resultado

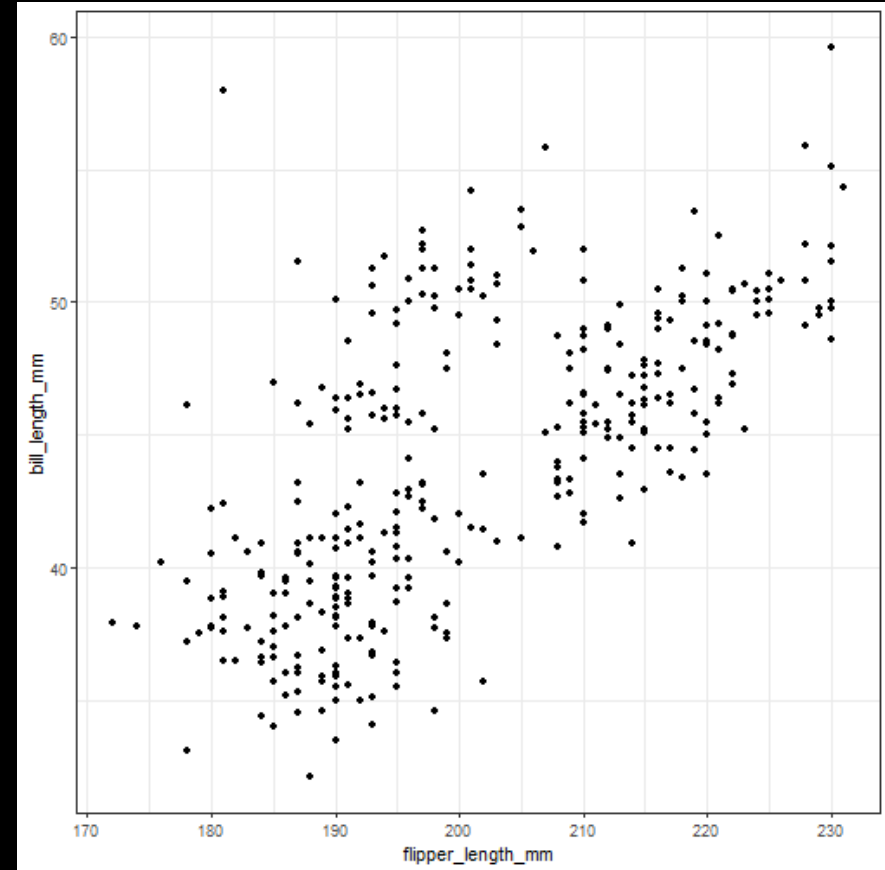


# Codigo generador

Grafico de puntos!

```
ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm) +  
  geom_point()
```

# Resultado



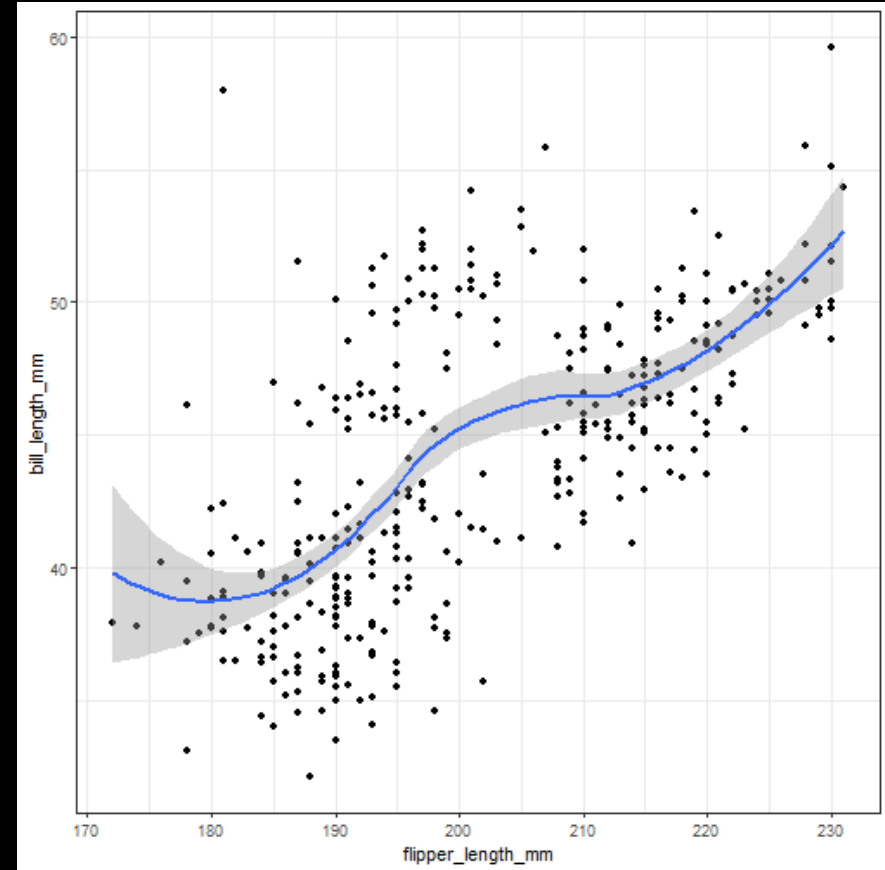


# Codigo generador

## Un modelo

```
ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm) +  
  geom_point() +  
  geom_smooth()
```

# Resultado

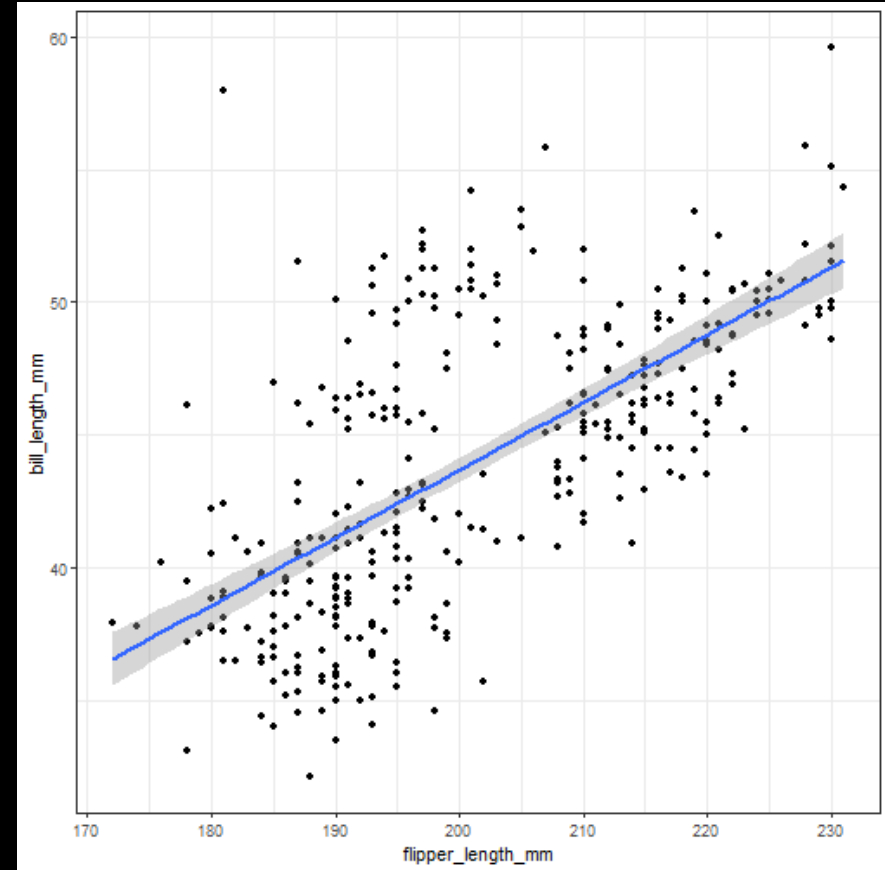


# Codigo generador

## Minimos cuadrados ordinarios

```
ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm) +  
  geom_point() +  
  geom_smooth(method= "lm")
```

# Resultado

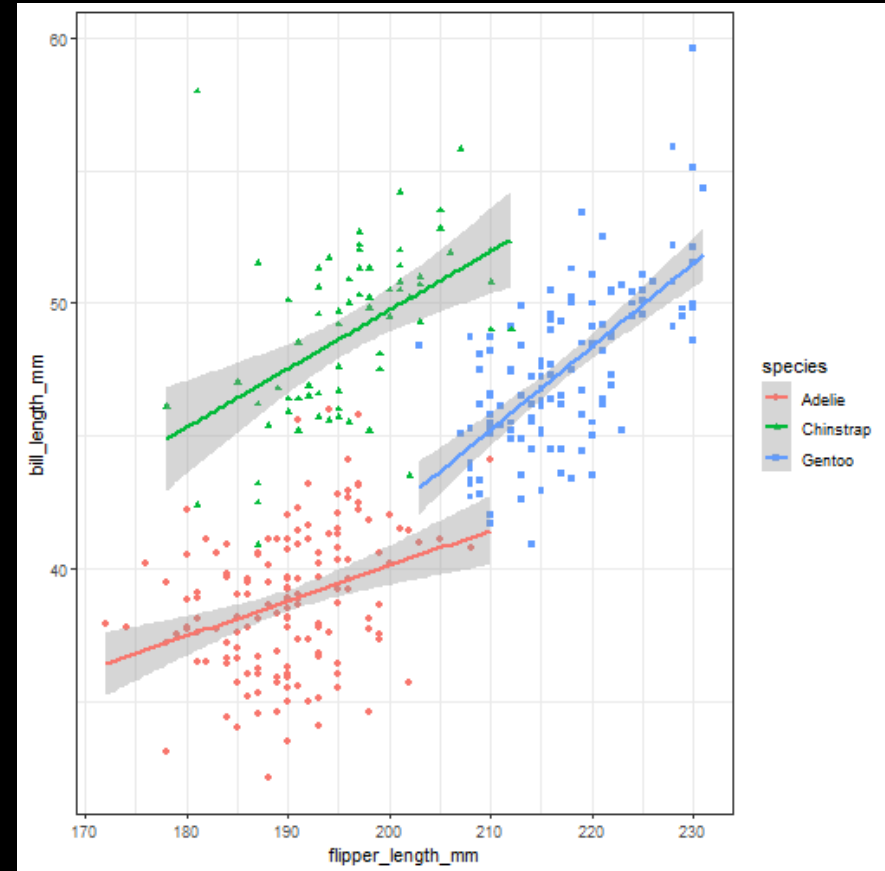


# Codigo generador

Un poco de color `color`, y formas `shape` diferentes para las especies

```
ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm,  
      color = species,  
      shape = species) +  
  geom_point() +  
  geom_smooth(method = "lm")
```

# Resultado

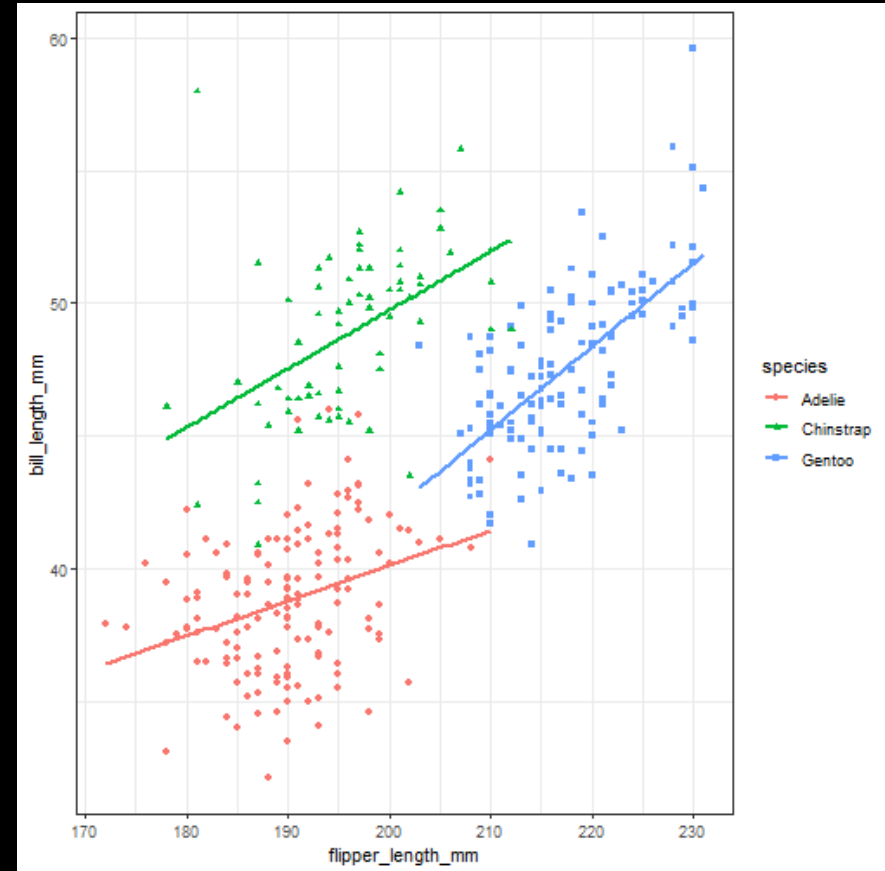


# Codigo generador

No me gusta los errores estandar `se = FALSE`

```
ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm,  
      color = species,  
      shape = species) +  
  geom_point() +  
  geom_smooth(method = "lm",  
              se = F)
```

# Resultado

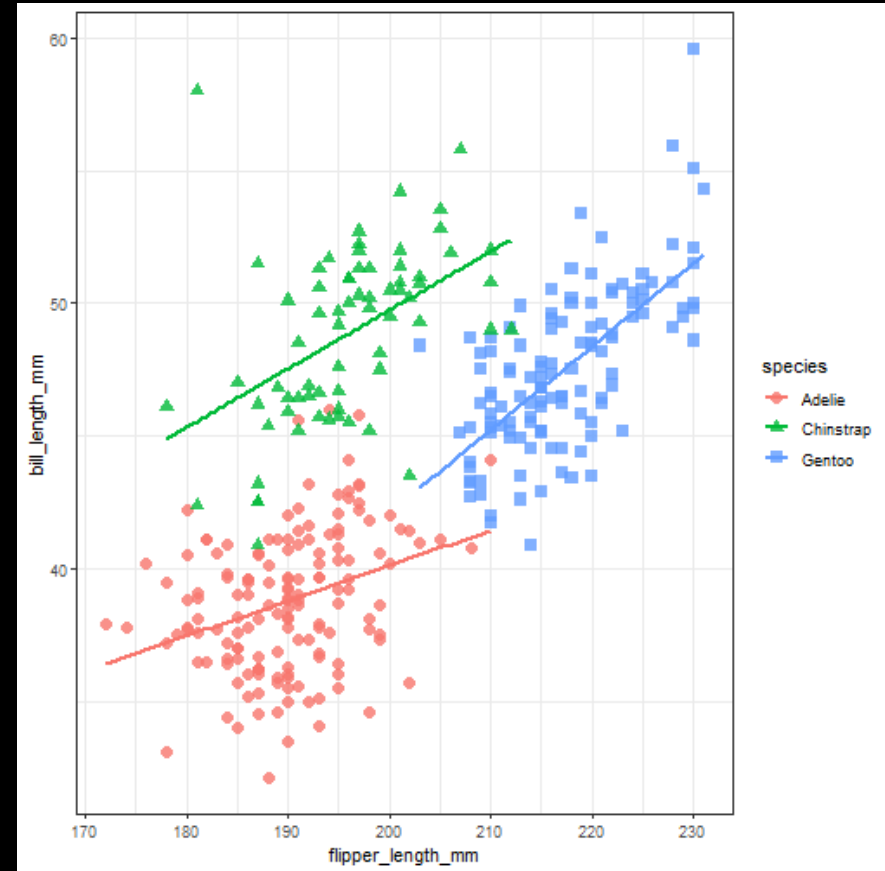


# Codigo generador

Muy pequeno!, no se notan las formas  
`size`, pero si se hacen mas grandes se  
sobreponen `alpha`

```
ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm,  
      color = species,  
      shape = species) +  
  geom_point(  
    size = 3,  
    alpha = 0.8  
  ) +  
  geom_smooth(method = "lm",  
             se = F)
```

# Resultado

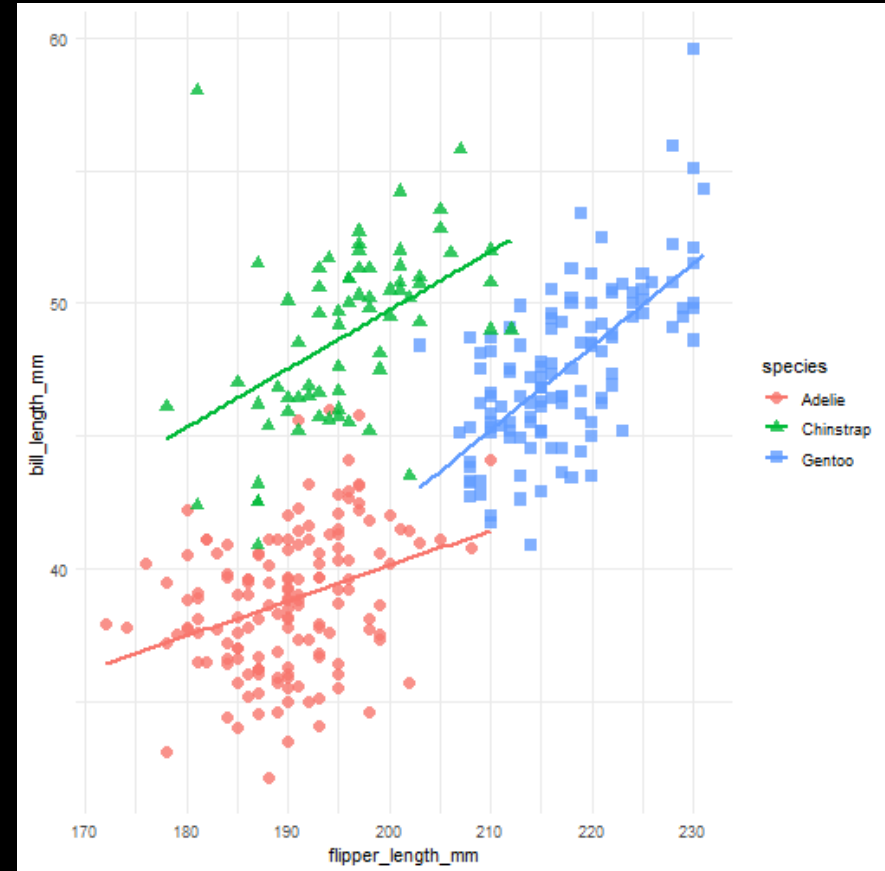


# Codigo generador

El fondo no me gusta `theme_`

```
ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm,  
      color = species,  
      shape = species) +  
  geom_point(size = 3, alpha = 0.8) +  
  geom_smooth(method = "lm",  
              se = F) +  
  theme_minimal()
```

# Resultado



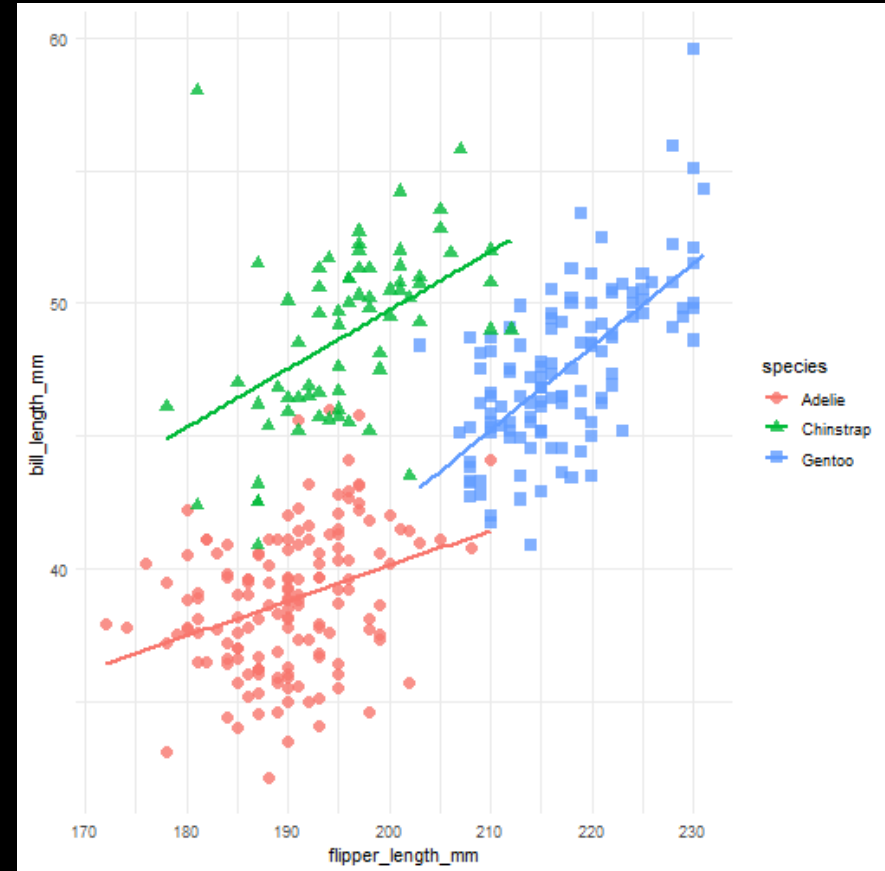
# Codigo generador

Mucho codigo ya cansa!, mejor es un objeto

```
mi_grafico <- ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm,  
      color = species,  
      shape = species) +  
  geom_point(size = 3, alpha = 0.8) +  
  geom_smooth(method = "lm",  
              se = F) +  
  theme_minimal()
```

```
mi_grafico
```

# Resultado

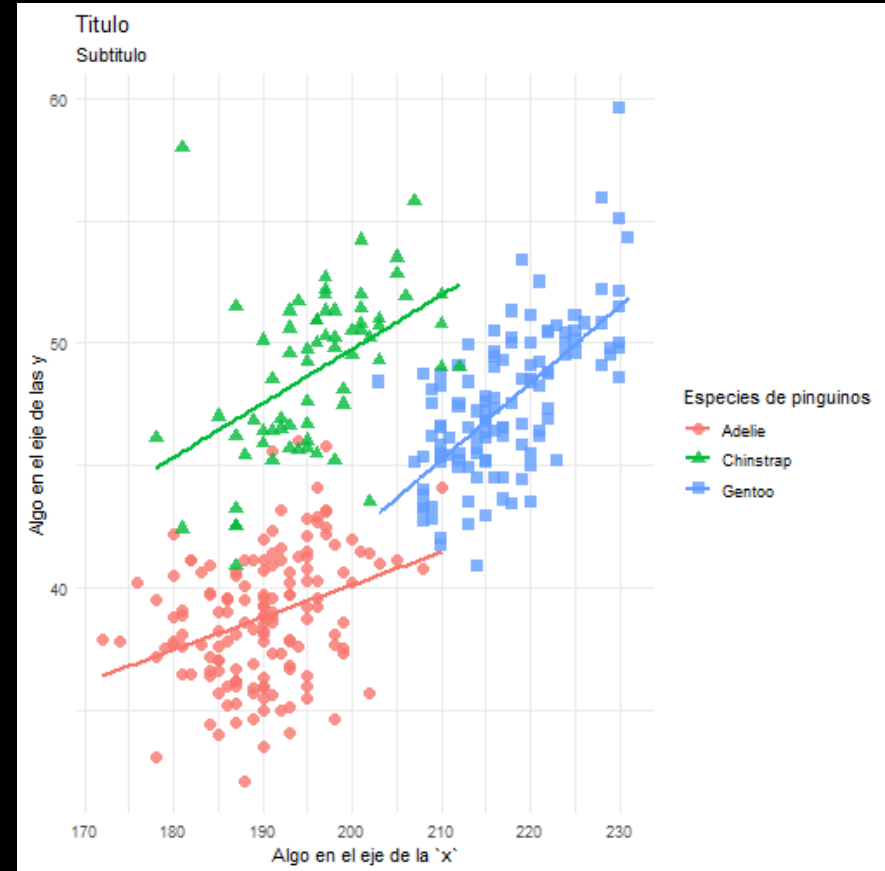


# Codigo generador

Podemos seguir agregando con total normalidad, ahora para no perdernos, definamos los titulos, subtítulos..etc. Todo esto en `labs`, todos sus elementos son intuitivos si lo traducimos al español.

```
lab_grafico <- mi_grafico +  
  labs(  
    title = "Titulo",  
    subtitle = "Subtitulo",  
    x = "Algo en el eje de la `x`",  
    y = "Algo en el eje de las `y`",  
    color = "Especies de pinguinos",  
    shape = "Especies de pinguinos")  
lab_grafico
```

# Resultado



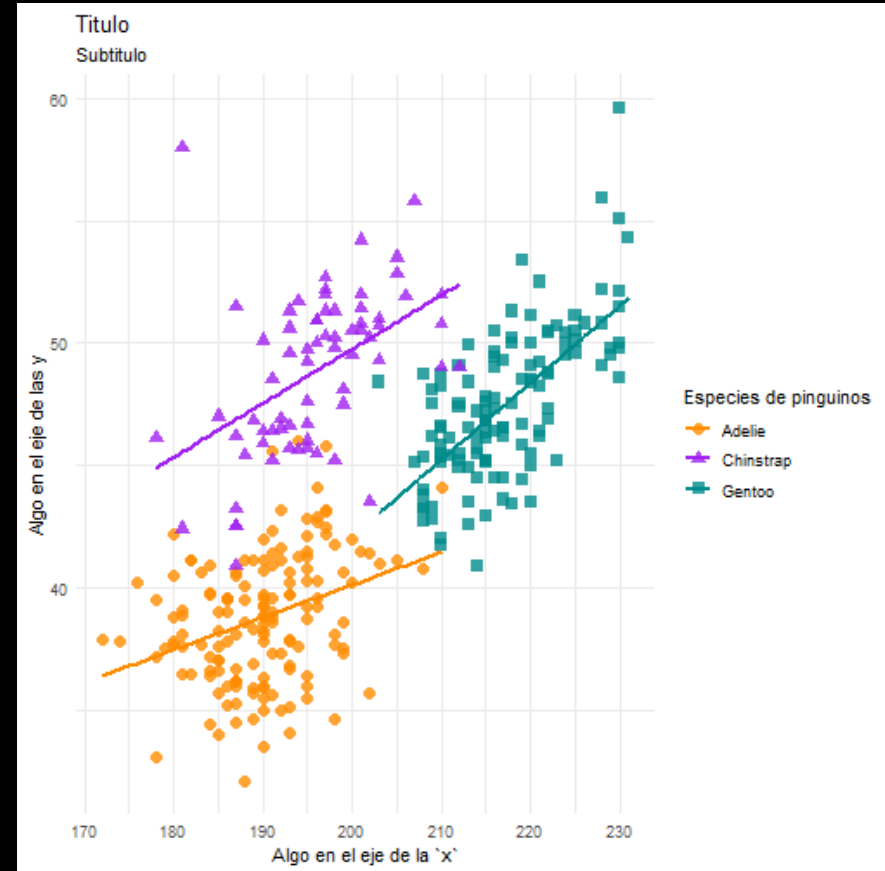


# Codigo generador

Ahora colores personalizados.

```
lab_grafico +  
  scale_color_manual(  
    values = c(  
      "darkorange",  
      "purple",  
      "cyan4") )
```

# Resultado

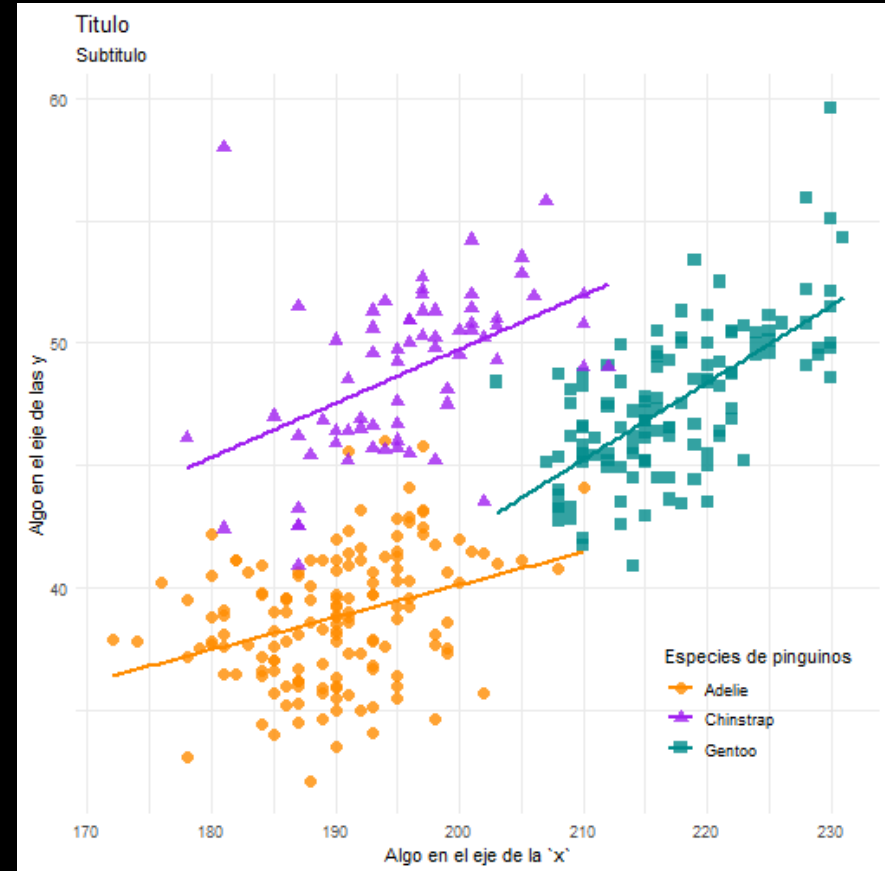


# Codigo generador

Ahora la posicion de la leyenda

```
lab_grafico +  
  scale_color_manual(  
    values = c(  
      "darkorange",  
      "purple",  
      "cyan4")) +  
  theme(legend.position = c(0.85, 0.15))
```

# Resultado



Uff

Que cansado fue eso, pero puede que sea la ultima vez que se tenga que repetir todos esos paso

## Primero definamos los elementos principales que se quiere presentar

```
mi_grafico <- ggplot(data = penguins) +  
  aes(x = flipper_length_mm,  
      y = bill_length_mm,  
      color = species,  
      shape = species) +  
  geom_point(size = 3, alpha = 0.8) +  
  geom_smooth(method = "lm",  
              se = F) +  
  theme_minimal()
```

## Luego elementos de estetica

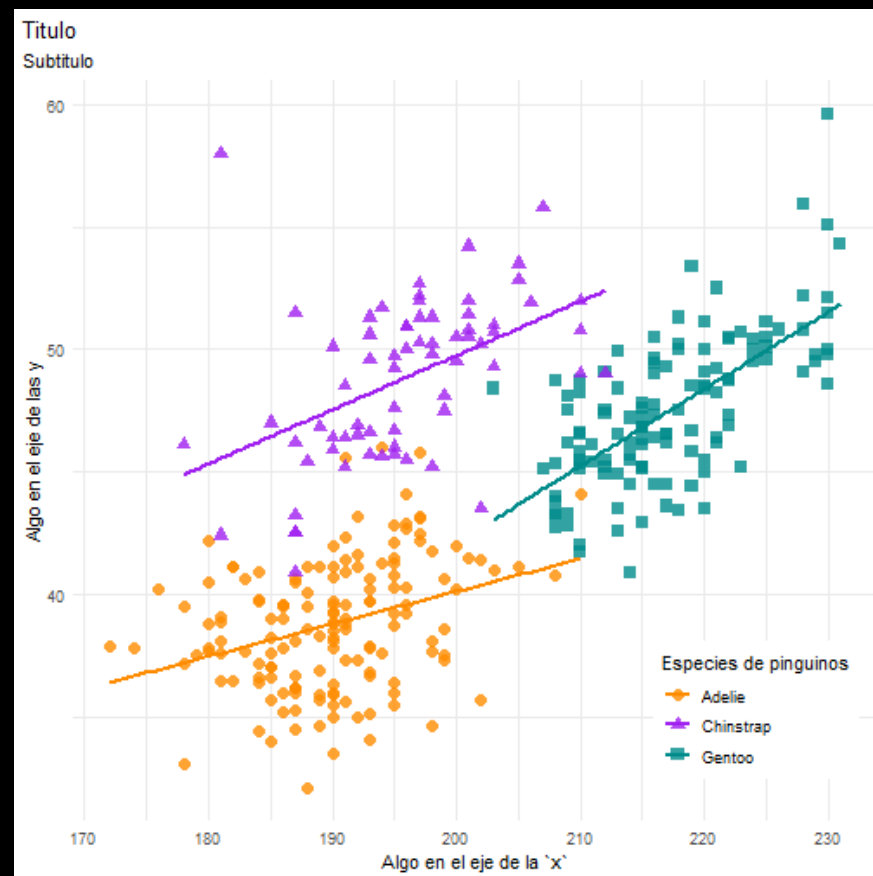
```
estetica <- labs(  
  title = "Titulo",  
  subtitle = "Subtitulo",  
  x = "Algo en el eje de la `x`",  
  y = "Algo en el eje de las y",  
  color = "Especies de pinguinos",  
  shape = "Especies de pinguinos") +  
  scale_color_manual(  
    values = c(  
      "darkorange",  
      "purple",  
      "cyan4") +  
    theme(legend.position = c(0.85, 0.15))
```

# Facil

```
final <- mi_grafico + estetica
```

```
final
```

# Final



# No se olvida a los autores de la Data

## Adélie penguins:

- Palmer Station Antarctica LTER and K. Gorman, 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Adélie penguins (*Pygoscelis adeliae*) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 5. Environmental Data Initiative. <https://doi.org/10.6073/pasta/98b16d7d563f265cb52372c8ca99e60f> (Accessed 2020-06-08).

## Gentoo penguins:

- Palmer Station Antarctica LTER and K. Gorman, 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Gentoo penguin (*Pygoscelis papua*) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 5. Environmental Data Initiative. <https://doi.org/10.6073/pasta/7fca67fb28d56ee2ffa3d9370ebda689> (Accessed 2020-06-08).

# Gracias!

Diapositivas creadas mediante [xaringan](#).

## Para descargar los actuales y futuros materiales

Ejecutar el siguiente comando via `git bash`

 `git clone https://github.com/TJhon/R-curso.git`

Las diapositivas estan en el siguiente link <https://github.com/TJhon/R-curso>