powered by *R programming*

Introducción general al uso de R - Clase 04 -

minolicnp@gmail.com

Ignacio Minoli *pwp*

Observatorio de Biodiversidad del Bosque Atlántico (OBBA)
Instituto de Biología Subtropical (IBS) - CONICET - UNaM.

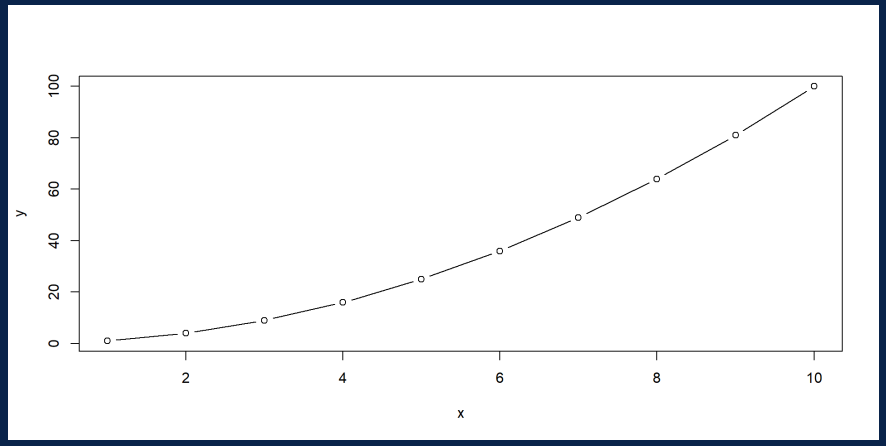
Gráficos o plots

- Los plots son parte escencial en la interpretación y comunicación de resultados obtenidos.
- Según los tipos de datos, formatos, cantidades y clases, se selecciona cual es el plot más adecuado.
- Es necesario saber explorar y conocer la estructura de los objetos para poder seleccionar que plotear.
- R base posee muchas altenativas a los plots más comunes con menos opciones de personalización.
- En la familia tidyverse y fuera de ella, hay una enorme variedad de paquetes que extienden las personalizaciones de los gráficos a un nivel superior que el R base.
- Con base a la complejidad de los objetos de salida de algunas funciones de algunos paquetes, crean sus propias funciones gráficas específicas.

Plots genéricos

- Utilizando **plot().**

```
# Creo dos vectores
x <- 1:10; y <-x*x

# Hago el plot
plot(x, y, type="b")
```

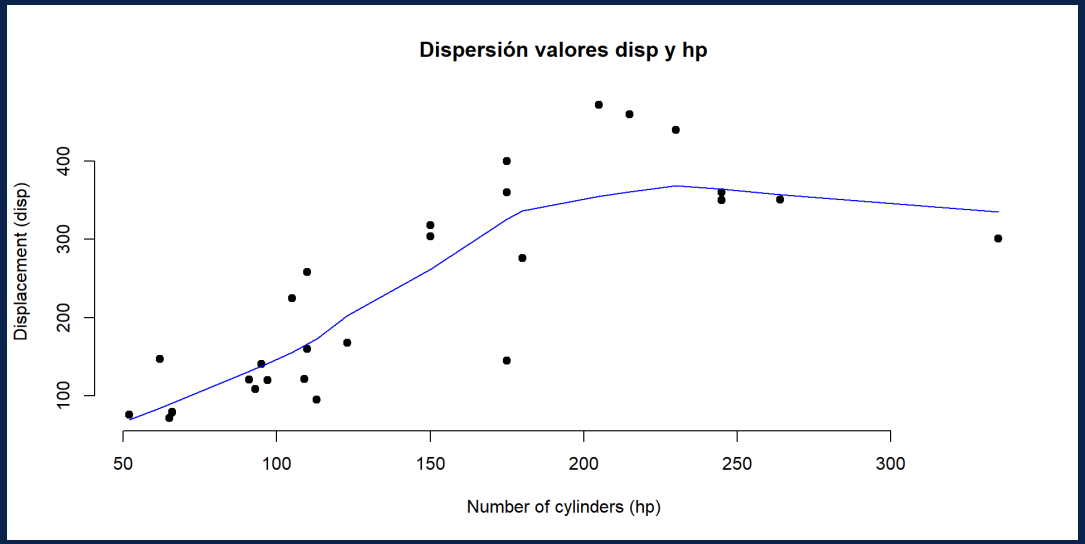



```
Argumentos de la función:
x , y: coordenadas de los puntos
type: es el tipo de gráfico a crear.
Pueden ser:
type = "p": plot de puntos (default)
type = "l": plot de lineas
type = "b": plot de puntos + lineas
type = "h": plot para histograma
type = "s": plot de escalones
type = "n": para no plotear
```

Scatterplots R base

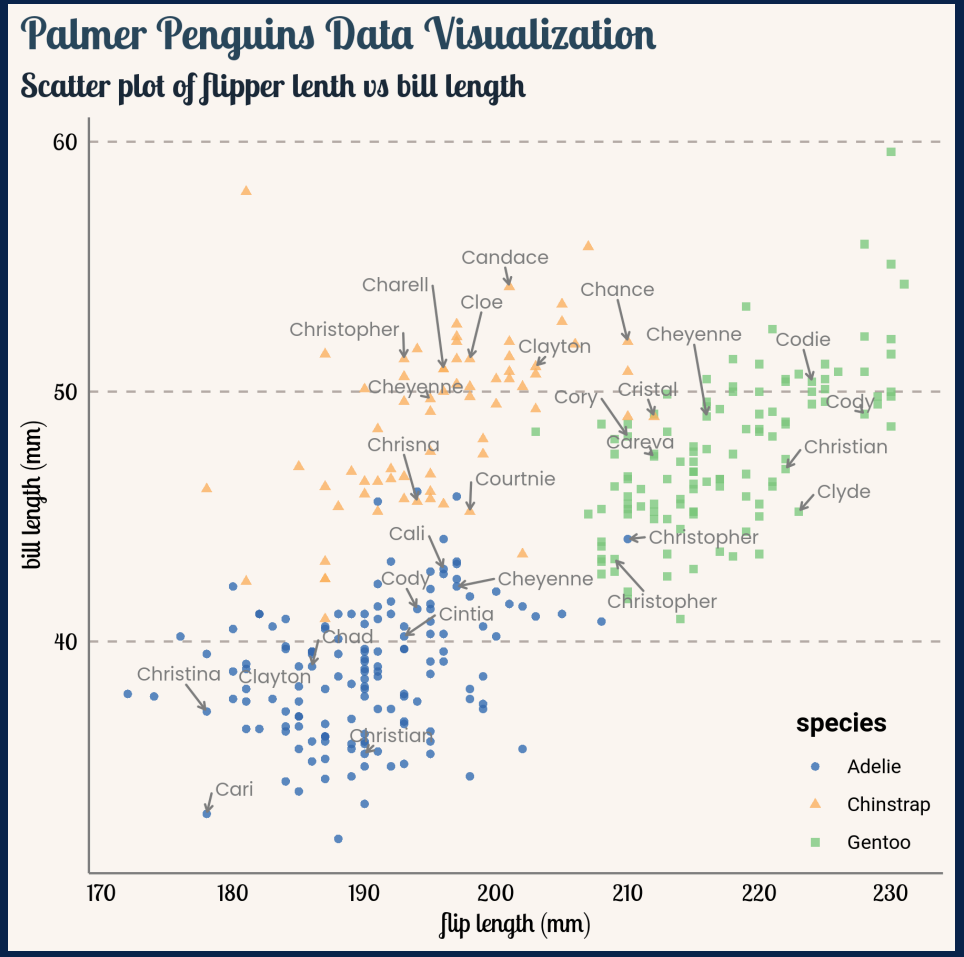
- Son los gráficos más comunes de dispersión de valores a lo largo de los ejes considerados R base **plot()**.

```r
# Uso dos vectores de data(mtcars)
x <- mtcars$hp # Number of cylinders
y <- mtcars$disp

# Hago el scatterplot
fig2 <- plot(x, y,
             main = "Dispersión valores
             xlab = "Number of cylinders
             ylab = "Displacement (disp)
             pch = 19,
             frame = FALSE
             )
lines(lowess(x, y), col = "blue")
#
# Argumentos de la función:
# x , y: coordenadas de los puntos
# main: título del plot
# xlab: label del eje x
# ylab: label del eje y
# pch: tipo de símbolo
# frame: con o sin marco
```



Scatterplots de otros paquetes

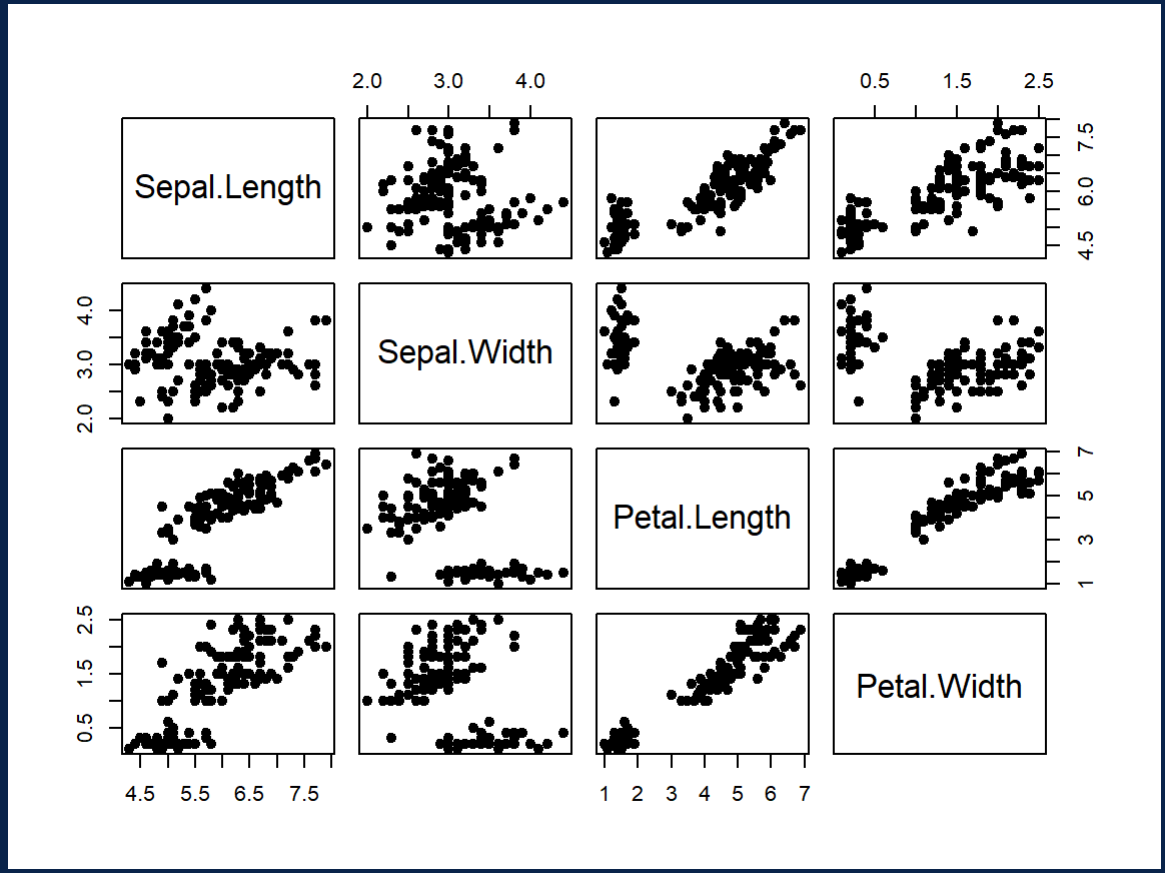
- Son muchas las opciones y paquetes que permiten hacer scatterplots. Por ej, con el paquete **ggplot2**, **geom_point()**.



Scatterplots completos

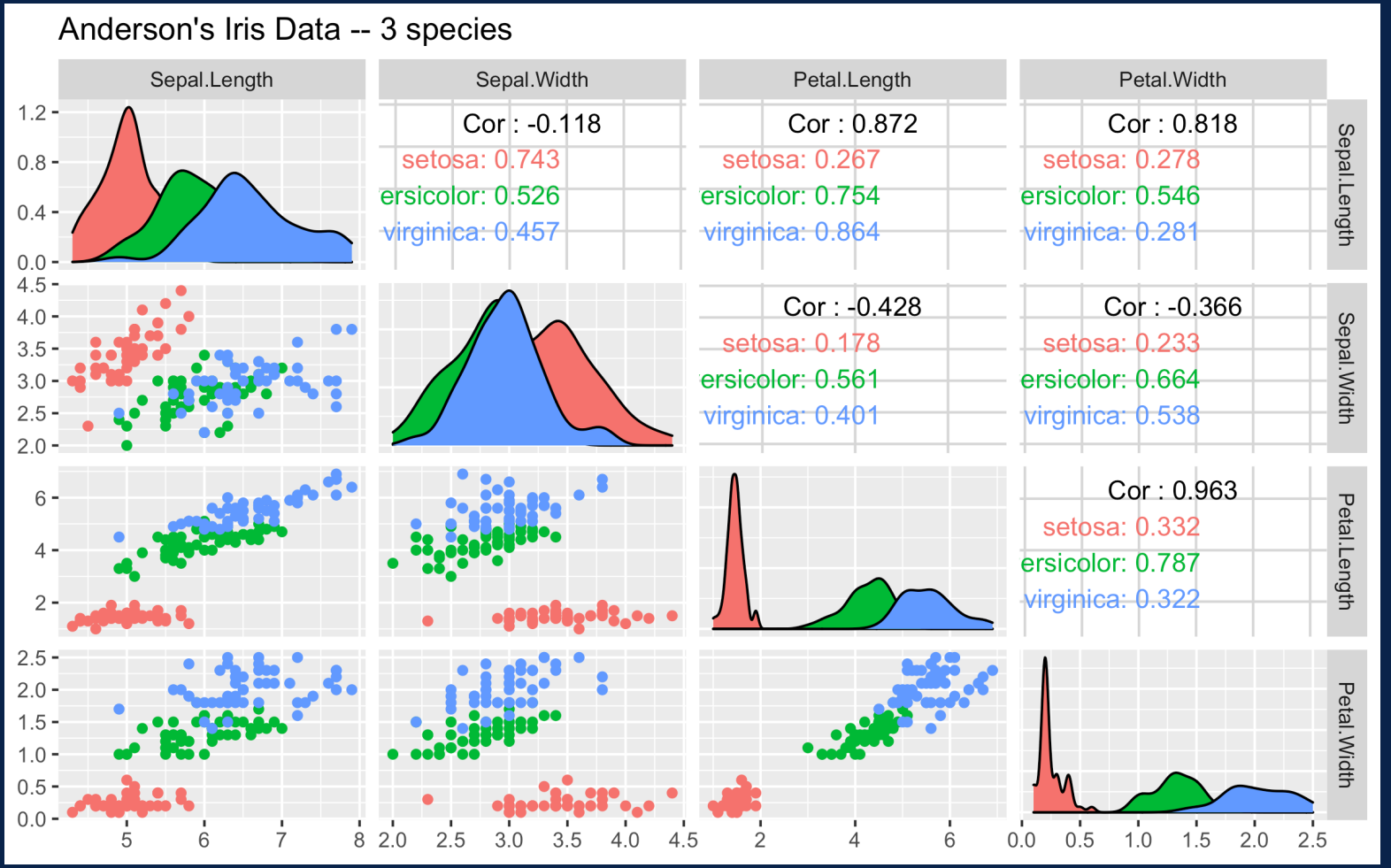
- Son los gráficos que se utilizan para ver globalmente un set de datos y como interactuar entre sí las variables que contiene. R base **pairs()**

```
pairs(iris[,1:4], pch = 19)
```



Scatterplots de otros paquetes

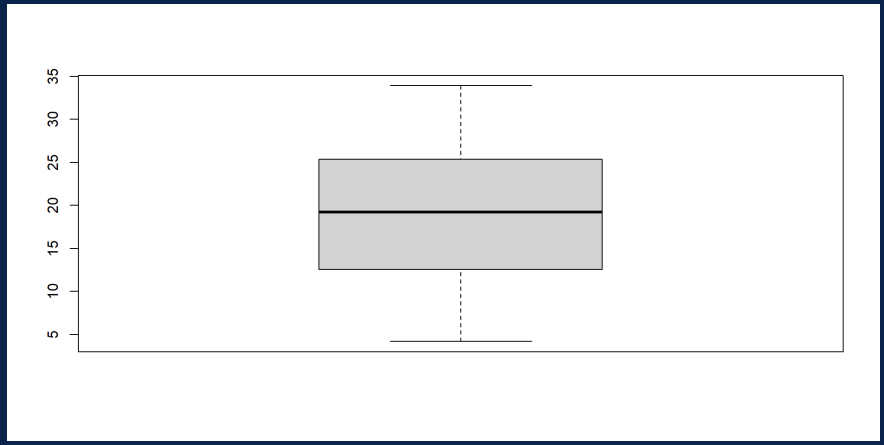
- Son muchas las opciones y paquetes que permiten hacer scatterplots. Por ej, con el paquete **GGally**, **ggpairs()**.

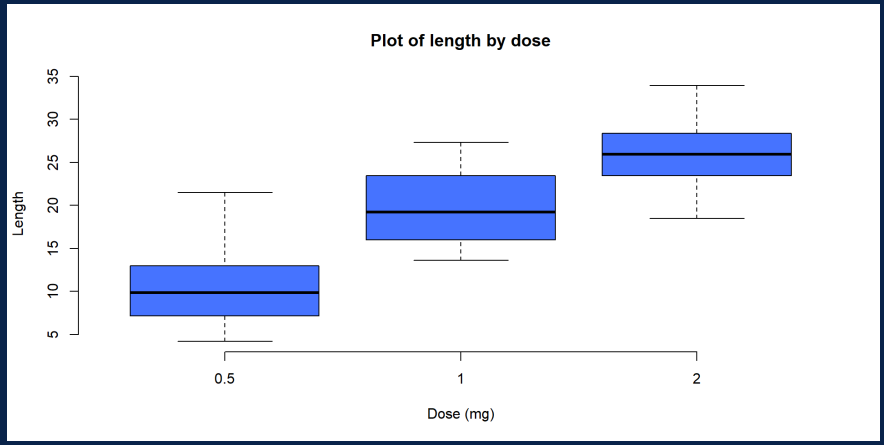


Boxplot

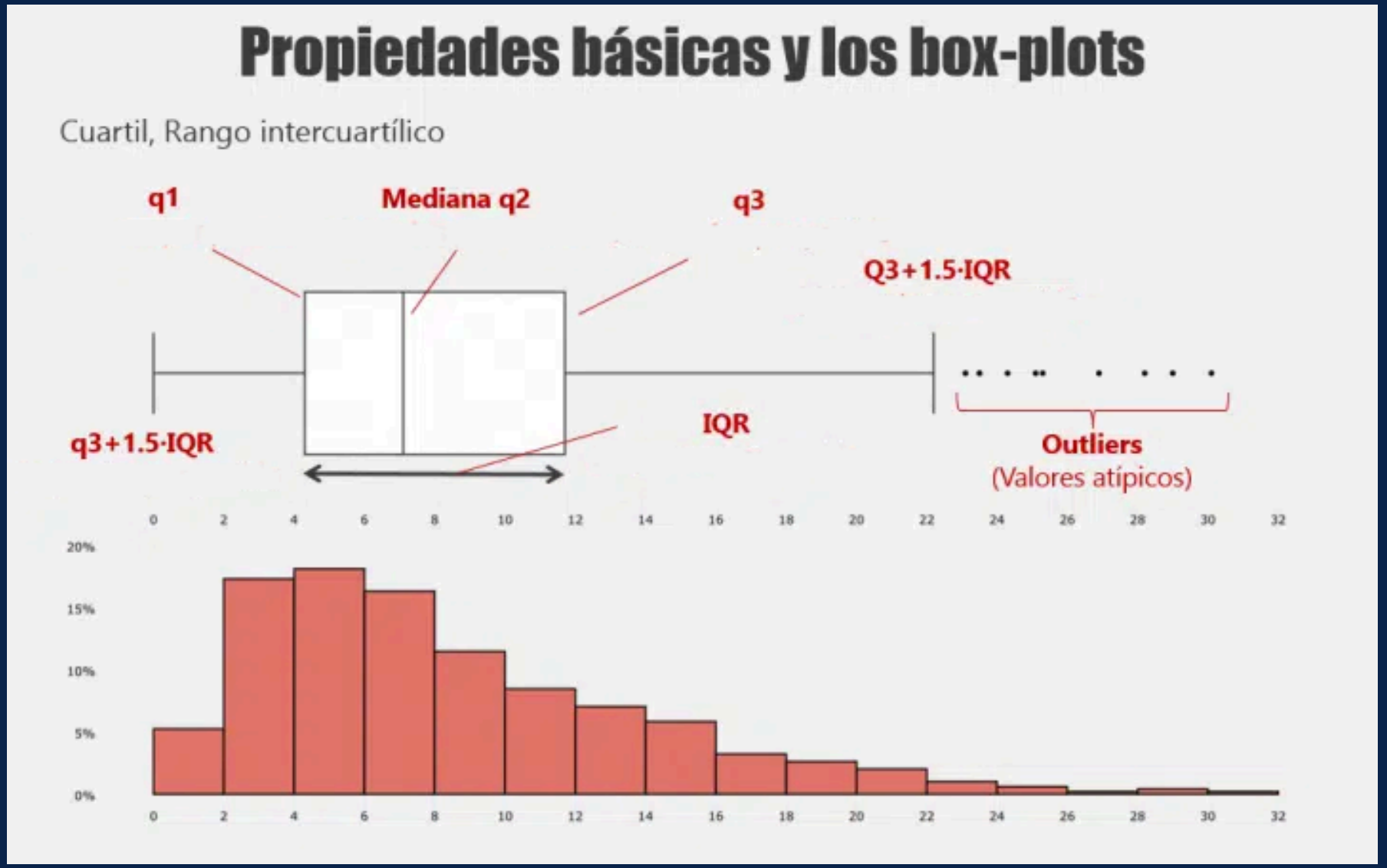
Son los gráficos que se utilizan para observar aspectos sobre la distribución de las observaciones en una o más series de datos cuantitativos. R base **boxplot()**

```r
# Boxplot de una sola variable
boxplot(ToothGrowth$len)

# Boxplot de dos variables
boxplot(len ~ dose,
        data = ToothGrowth,
        main = 'Plot of length by dose',
        xlab = 'Dose (mg)',
        ylab = 'Length',
        col = 'royalblue1',
        frame = FALSE
        )

# Argumentos:
# len: eje y
# dose: eje x
# data: data frame
# main: título del plot
# xlab: label del eje x
# ylab: label del eje y
# col: color
# frame: con o sin marco
```



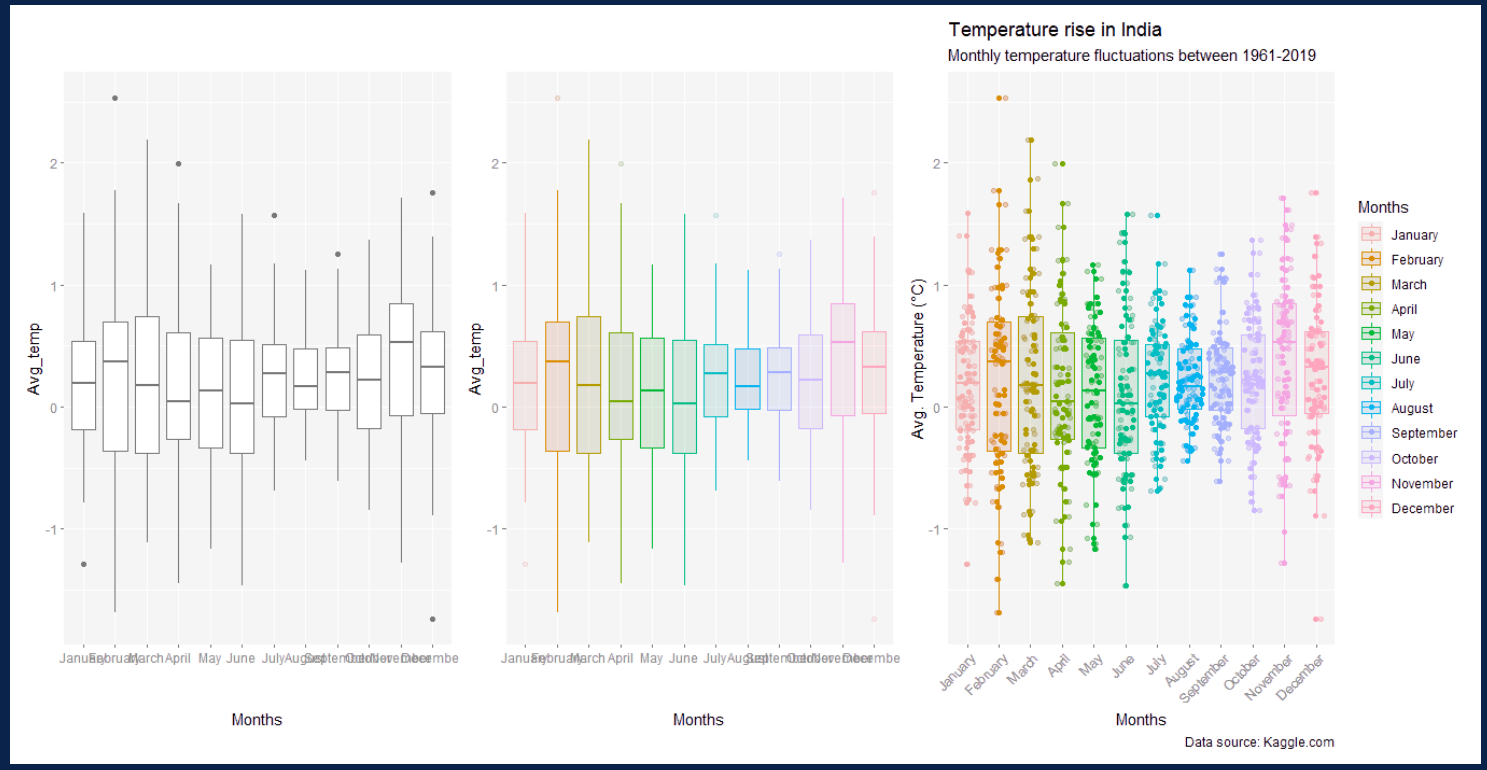


Boxplot y sus partes



Boxplot de otros paquetes

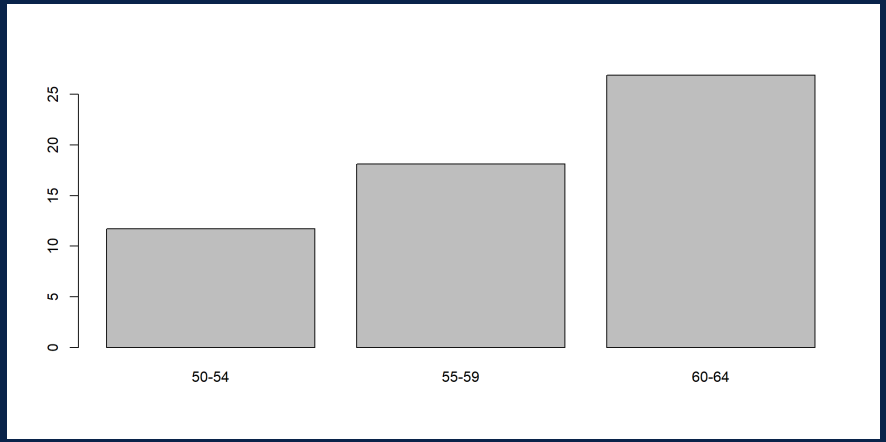
- Son muchas las opciones y paquetes que permiten hacer boxplots. Incluso con distintas formas de cajas, etiquetas de observaciones, detectar outliers, horizontales, verticales, mostrando valores media, mediana. Por ej, con el paquete **ggplot2**, **geom_boxplot()**.

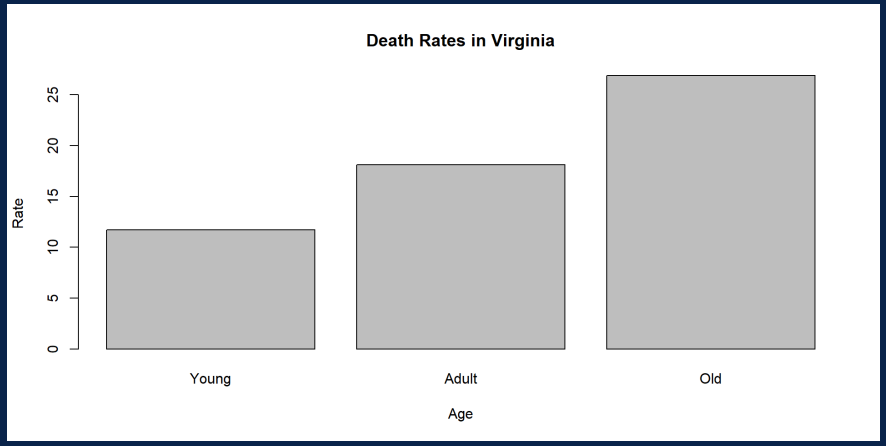


Barplots

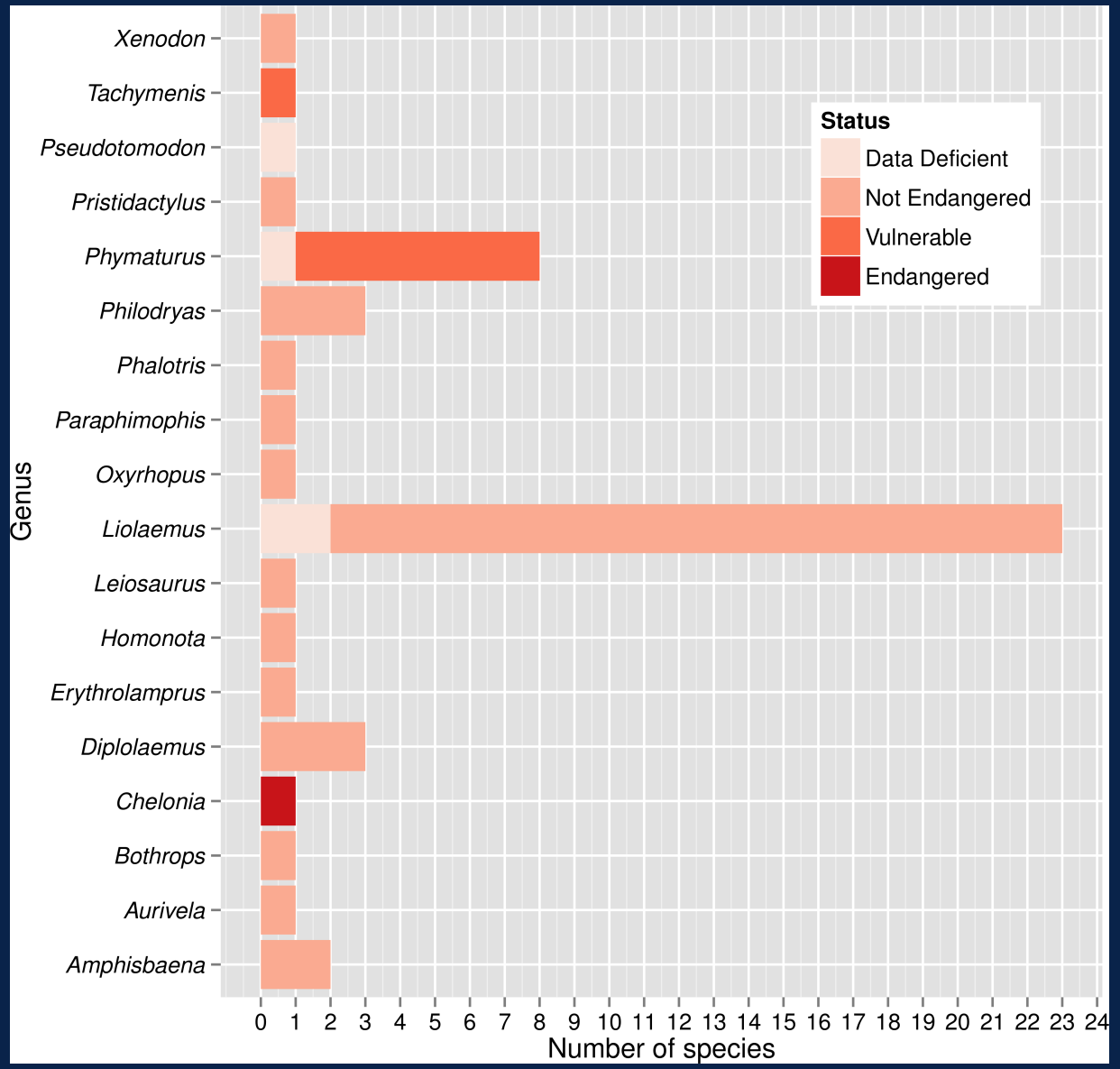
- Se utiliza cuando quiero visualizar la relación entre una variable numérica y una categórica. Puedo utilizar gráficos de barras apiladas o “stacked” cuando tengo un factor con más de un nivel a mostrar. R base **barplot()**

```r
x <- VADeaths[1:3, "Rural Male"]
x

# Subset
x <- VADeaths[1:3, "Rural Male"]

# Barplot de una variable
barplot(x)

# Barplot de 2 variables
barplot(x,
        main = "Death Rates in Virginia",
        names.arg = c("Young", "Adult", "Old"),
        xlab = "Age",
        ylab = "Rate"
        )
```





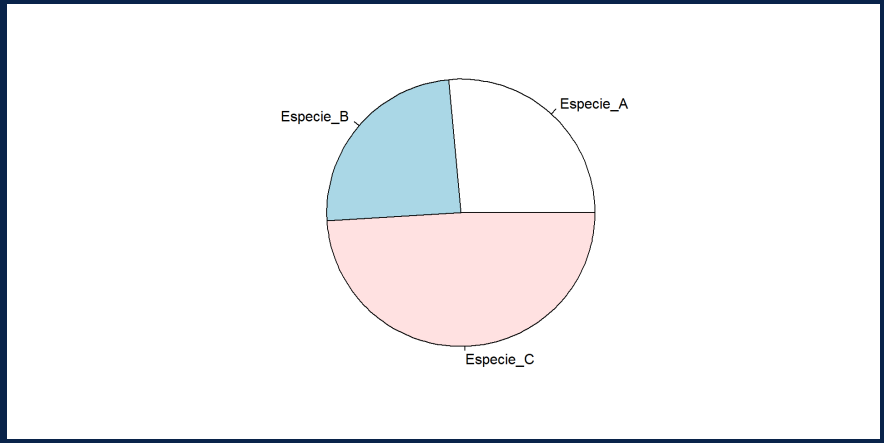
Barplots - otros paquetes

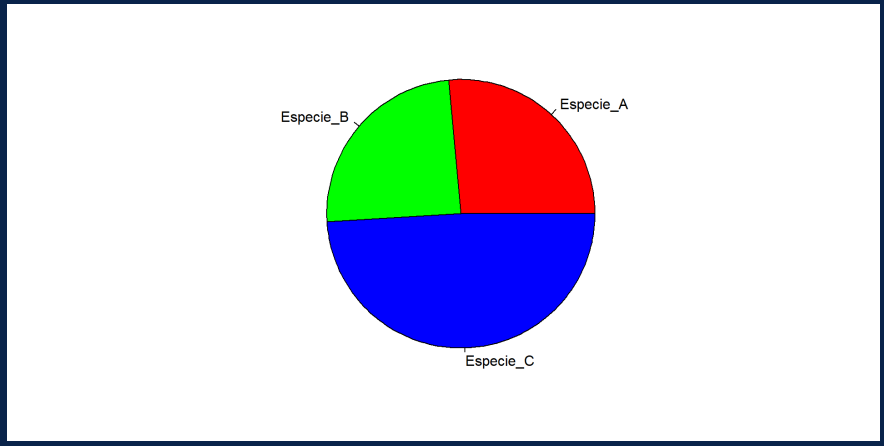


Pie charts

- Los gráficos de torta no están recomendados en la documentación de R CRAN. Se utilizan para representar porporciones numéricas de manera relativa al número de datos totales en el data frame. R base **pie()**

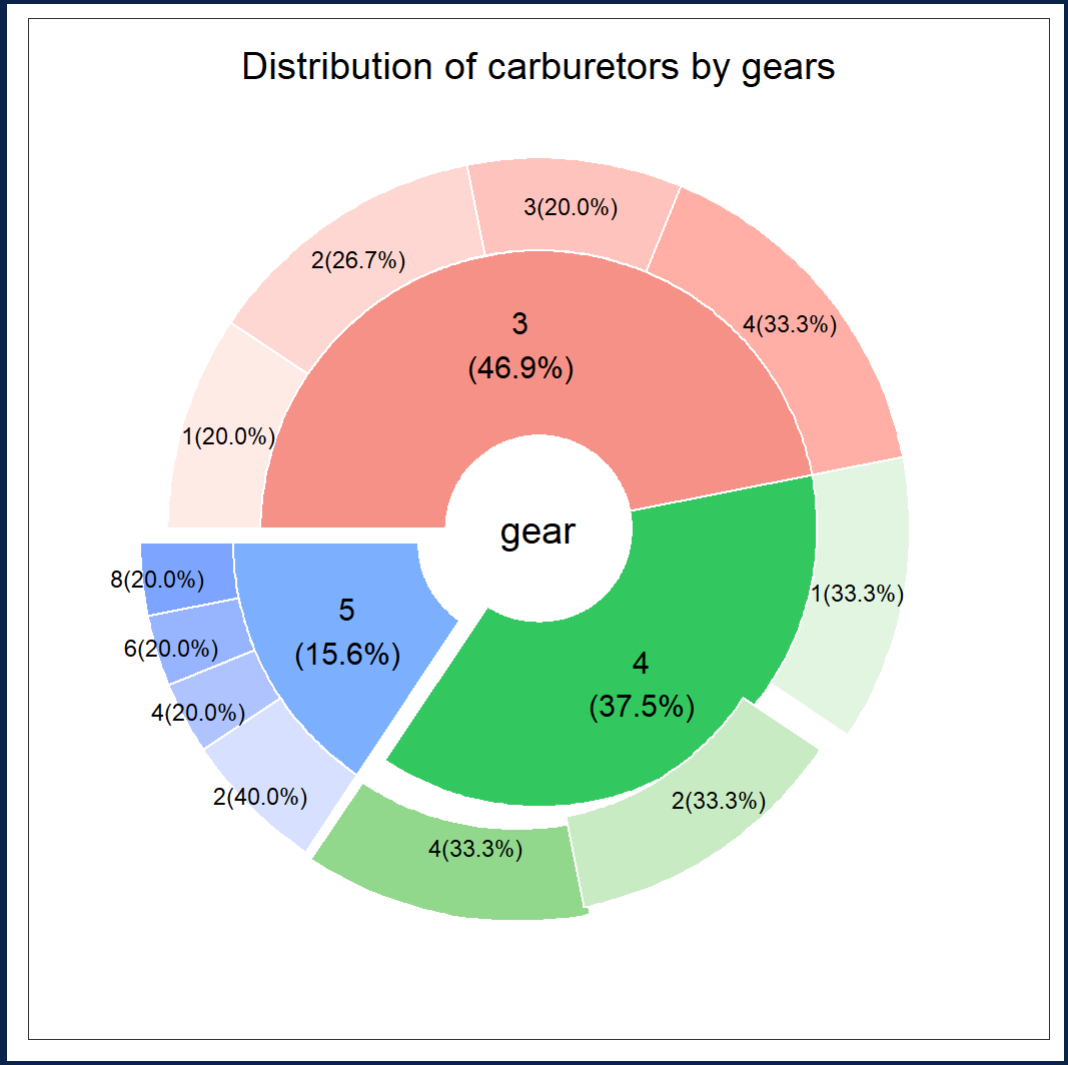
```
df <- data.frame(
  especies = c("sp_A",
               "sp_B",
               "sp_C"
               ),
  altura = c(27, 25, 50)
)
  especies altura
1     sp_A     27
2     sp_B     25
3     sp_C     50

pie(df$altura,
    labels=df$especies,
    radius = 1.05
    )

pie(df$altura,
    labels = df$especies,
    radius = 1.05,
    col = c("red", "green", "blue")
    )
```





Pie charts - otros paquetes

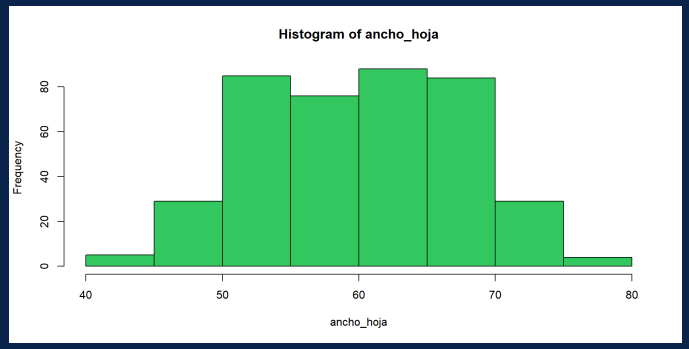
```
webr::PieDonut(mtcars, ggplot2::aes(gear, carb), explode=3,r1=0.9, explodeDonut=TRUE,
               title="Distribution of carburetors by gears", star=3*pi/2,labelposition=0)
```

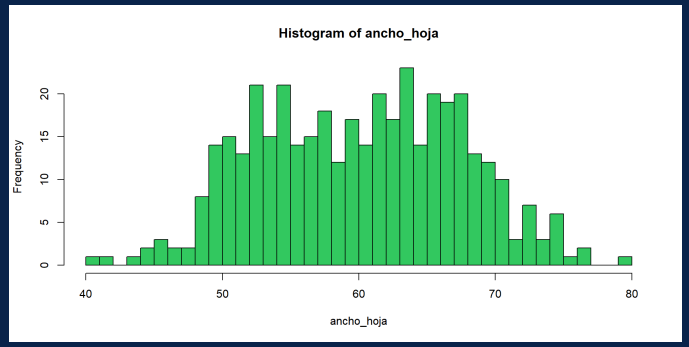


Histogramas

- Los histogramas se utilizan para representar la frecuencia numérica de variables usando un gráfico de barras. La altura en las barras en el eje y representa la frecuencia de distribución de una variable (la cantidad o que tan seguido se registra). El ancho de las barras (eje x) representa el valor de la variable (e.g., minutos, años, edades). Es bi dimensional. R base **hist()**.

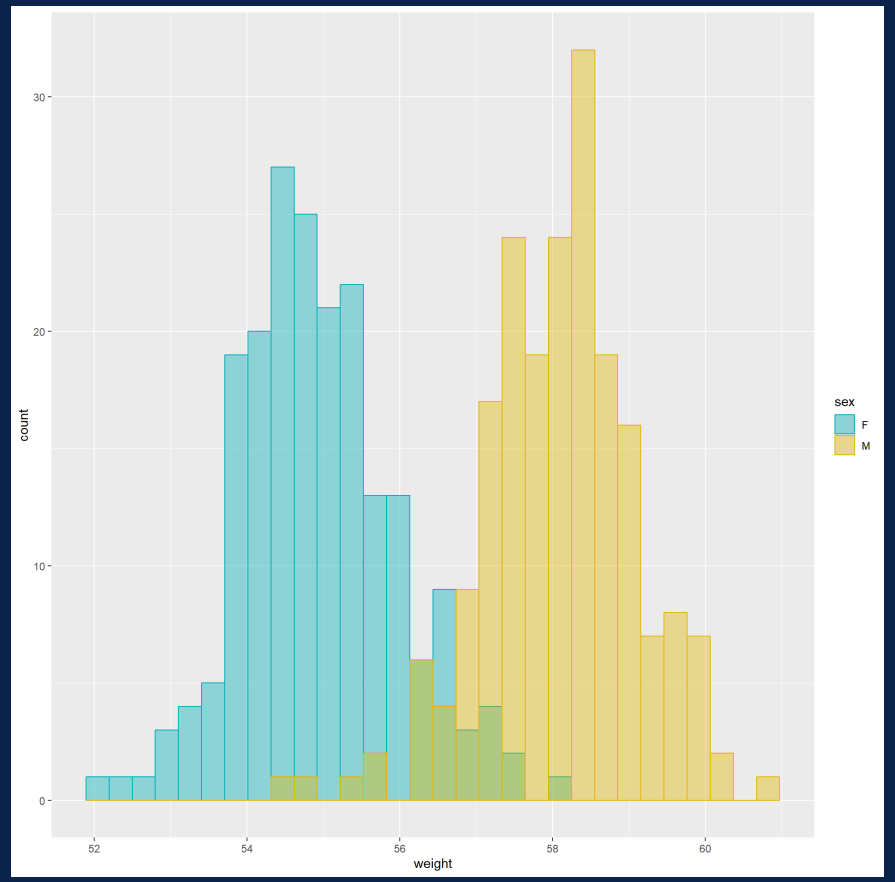
```
set.seed(1234)
ancho_hoja <- c(rnorm(200, mean=55, sd=5
rnorm(200, mean=65, sd=5))
head(ancho_hoja, 2)

48.96467 56.38715

hist(ancho_hoja,
      col = "#33c860"
      )

hist(ancho_hoja,
     col = "#33c860",
     breaks = 30
     )
```





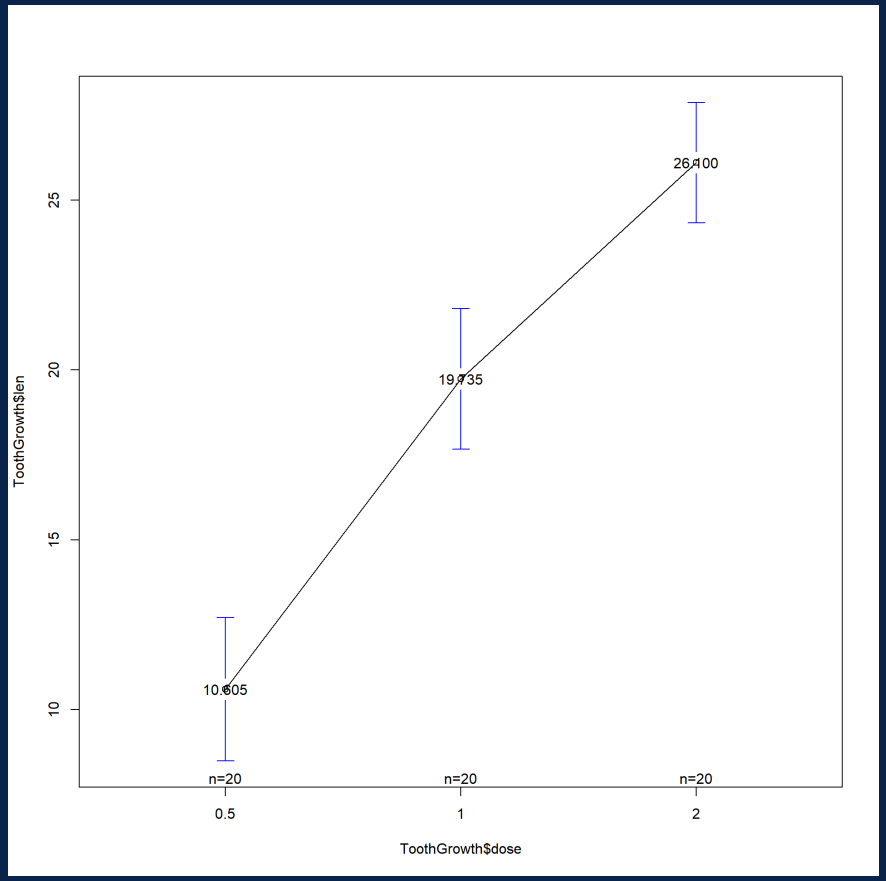
Histograma - otros paquetes

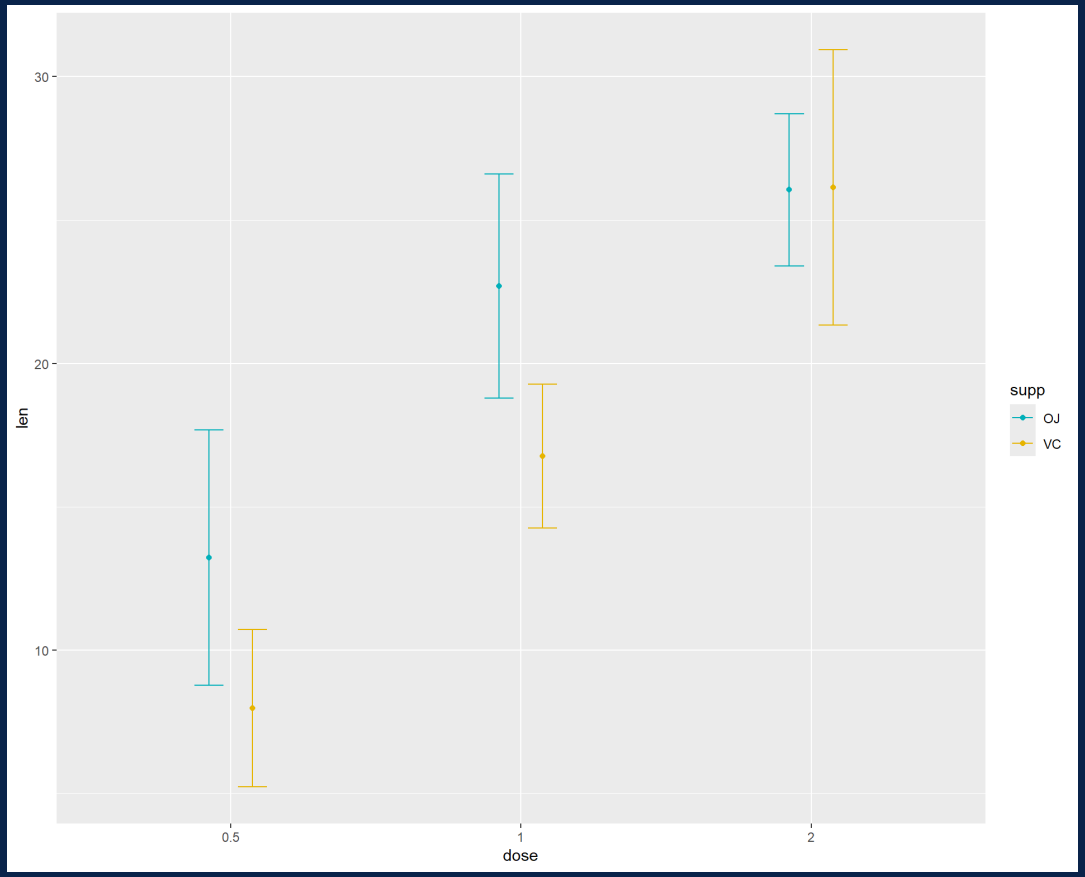
```r
# Mismos valores al azar
set.seed(1234)

# Creo el set de datos
wdata <- data.frame(
sex = factor(rep(c("F", "M"), each=200)),
weight = c(rnorm(200, 55), rnorm(200, 58))
)

# Veo las 4 primeras lineas
head(wdata, 4)
  sex    weight
1   F 53.79293
2   F 55.27743
3   F 56.08444
4   F 52.65430

# Llamo el paquete
library(ggplot2)

# Hago el histograma
ggplot(wdata, aes(x = weight)) +
  geom_histogram(aes(color = sex, fill = sex),
                 position = "identity",
                 bins = 30,
                 alpha = 0.4
```



Intervalos de confianza

- Un plot de intervalos de confianza se emplea para comparar grupos de manera similar a un boxplot o plot de puntos con datos continuos. Muestra el intervalo de confianza para la media de los datos.





Graphics devices

- Es todo aquello en el entorno de una PC que permite visualizar y/o almacenar un plot o gráfico.
- Puede ser:

  * Una ventana en la PC (screen device).
  * Un archivo PDF (file device).
  * Un archivo PNG, TIFF, JPEG, etc. (file device).
  * Un archivo escalable vectorial SVG (file device).

- Al hacer un gráfico en R debe ser "enviado" a un graphic device. Si solo se visualiza el gráfico, se usará un screen device según el sistema operativo:

  * En Mac: **quartz()**
  * En Windows **windows()**
  * En Unix/Linux **x11()**

Graphics devices

- La lista de dispositivos admitidos por su instalación de R se encuentra en ? Devices.
- Un file device puede ser de dos tipos: vector (pdf, svg, postscript) y bitmap devices (png, jpeg, tiff, bmp).
- Funciones como **plot()** en R base, **xyplot()** en el paquete lattice, o **qplot()** en ggplot2 enviarán por default un screen device según sea el sistema operativo.
- Es posible abrir múltiples graphic devices (screen, file, o ambos!). Incluso es posible muchos screen devices con una sola función ejecutada.
- Las opciones de colores y parámetros de los gráficos son un tema largo y complejo, por lo cual deben ser aprendidos de manera práctica. Los argumentos y las opciones pueden cambiar en paquetes de la familia tidyverse.

Graphics devices

- Los paquetes para gráficos avanzados en R son muchos y depende de las necesidades de los objetos-resultados producidos con nuestros anaĺisis.

- R como herramienta estadística y científica tuvo y tiene éxito principalmente con base en las capacidades y diversidad gráfica que posee para mostrar resultados.

- Algunos paquetes top para visualización más completos, versátiles, configurables, documentados y mantenidos son:

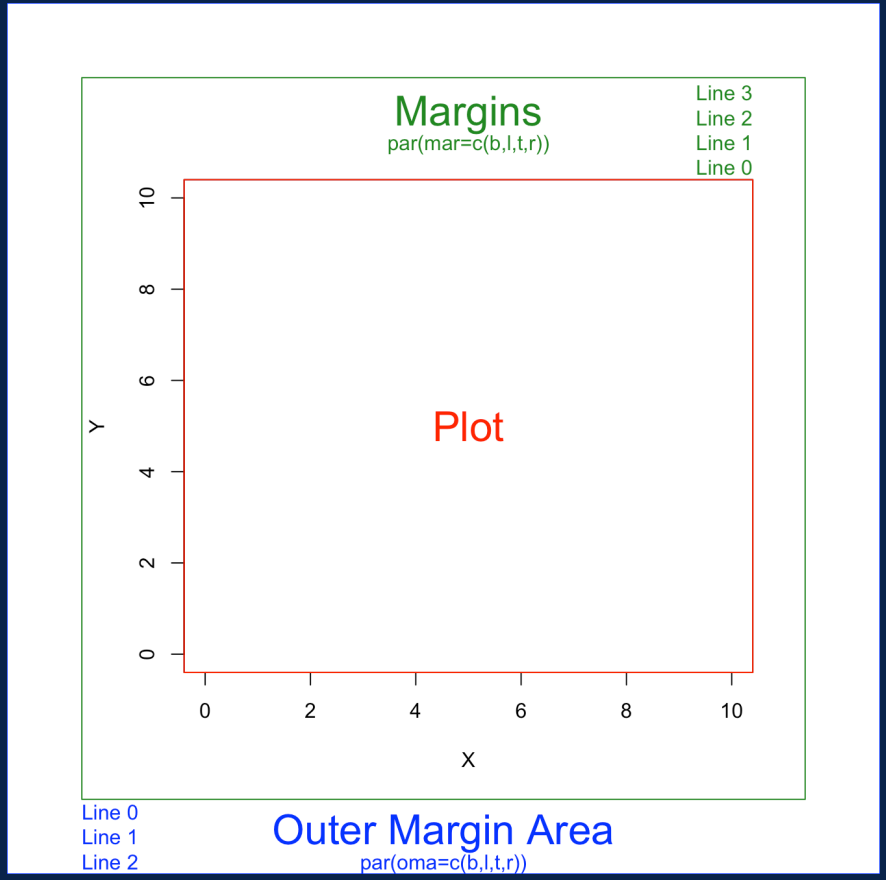
Paquetes para gráficos

- **1) ggplot2**: (Hadley Wickham/Tidyverse): multicapas | curva aprendizaje alta
- **2) lattice**: basado en el paquete grid (R base) incorporó paneles de plots
- **3) leaflet**: ofrece maneras ligeras y potentes para hacer mapas interactivos.
- **4) RColorBrewer**: provee 3 tipos de paletas de colores: secuencial, divergente y cualitativa.
- **5) plotly**: optimiza la visualización con herramientas interactivas personalizables (botones, controles deslizantes y menús desplegables para mostrar diferentes perspectivas de gráficos).
- **6) rgl**: basado en el paquete grid (R base) es para crear gráficos 3D interactivos (rotables en todos los sentidos).
- **7) kableExtra**: permite crear tablas con colores y estilos de fuente en HTML.

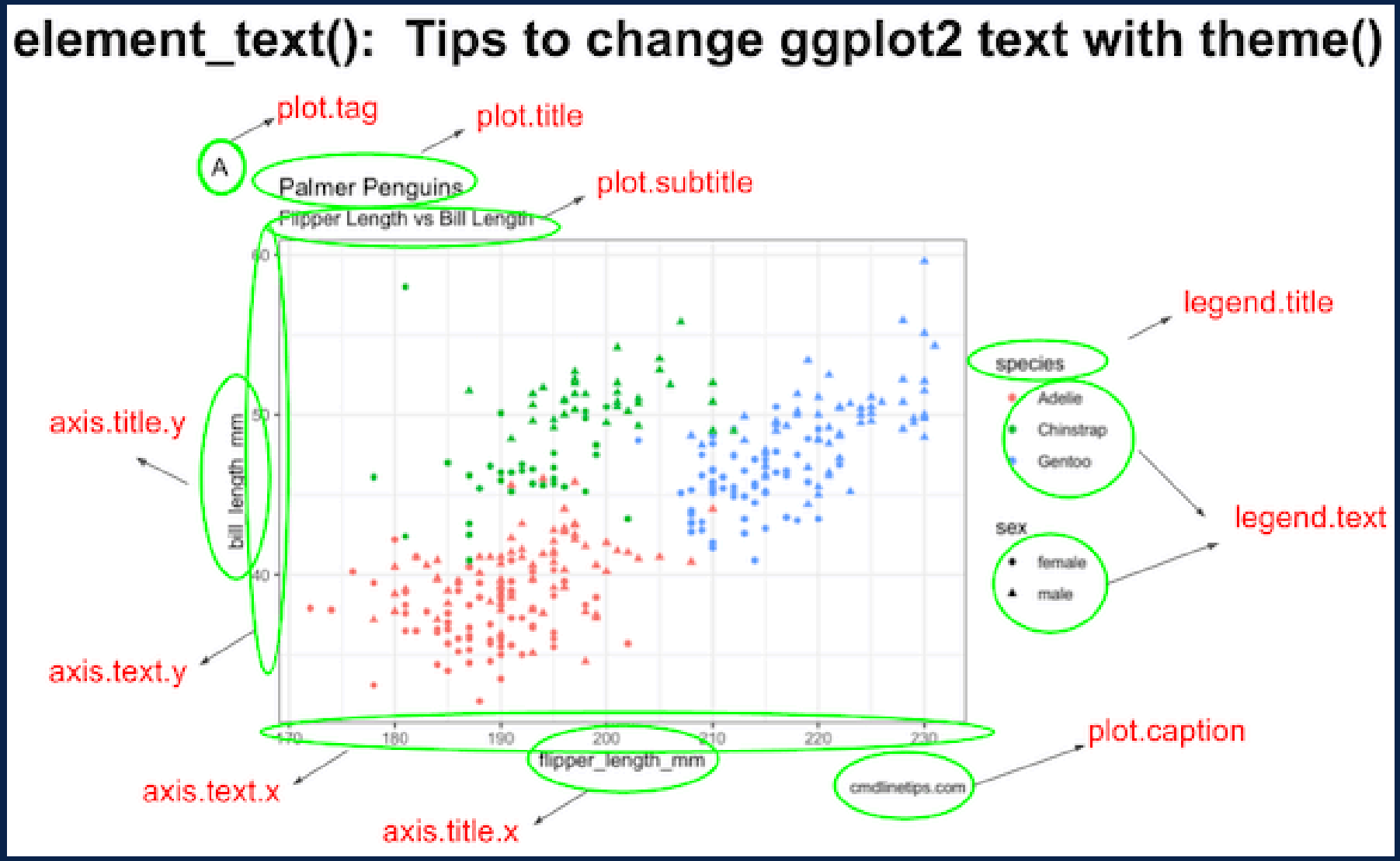
Componentes gráficos

- Hay componentes comunes entre los plots del R base y la familia tidyverse, pero se personalizan de diferente manera:
- Datos:

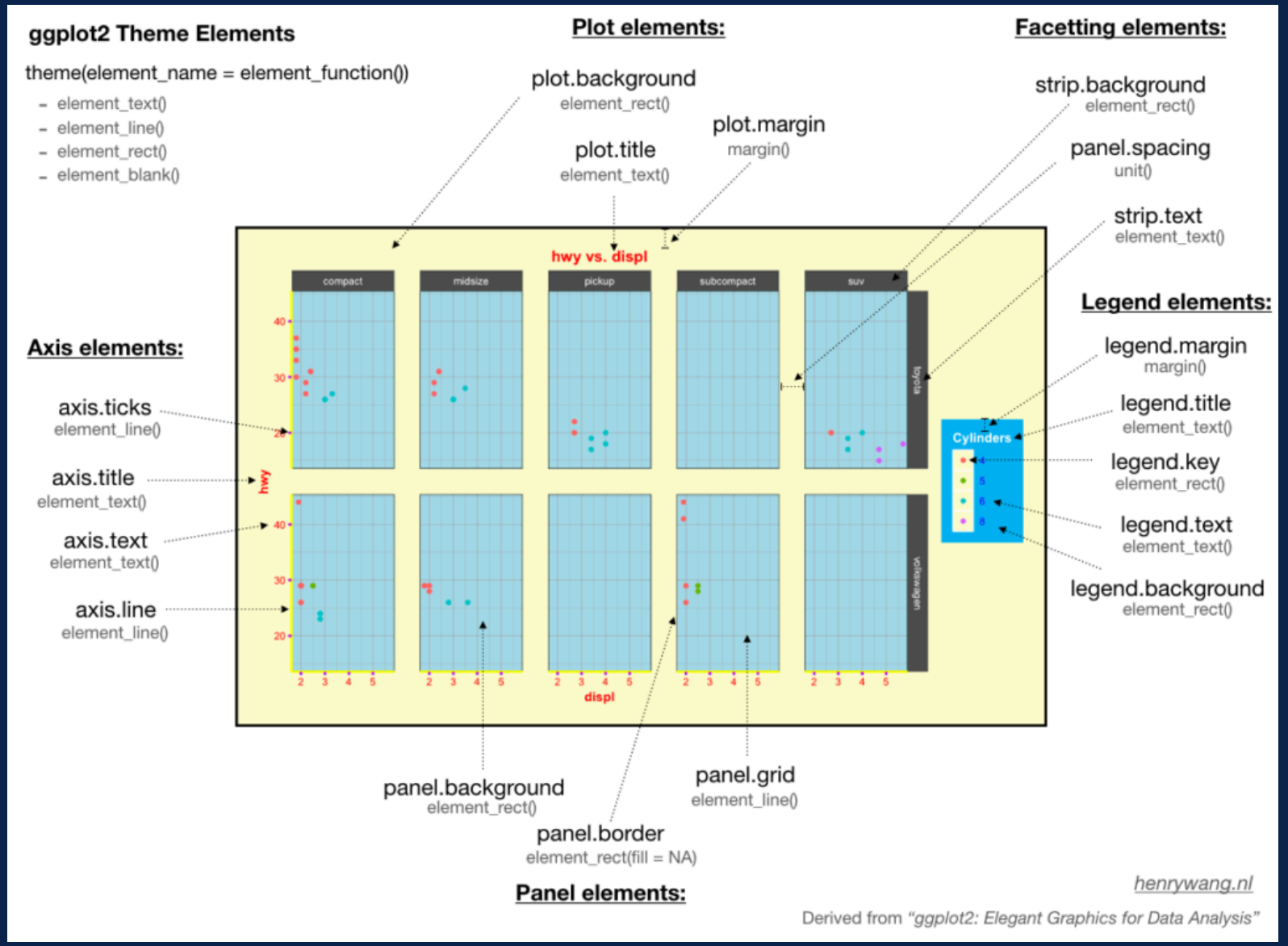
  * Colores
  * Símbolos - lineas - barras
  * Título
  * Leyenda
  * Ticks - escalas - ejes
  * Etiquetas ejes
  * Márgenes



Componentes gráficos ggplot2



Componentes gráficos ggplot2



Guardar gráficos

- Desde RStudio puedo hacerlo de manera visual desde la sección “Plots” con el botón “Export”. Incluso puedo hacer algunos ajustes post creación del plot.
- Desde las lineas de código, primero debo crear el archivo con la extensión del formato que quiero guardar (pdf, png, etc.). Argumentos posibles:

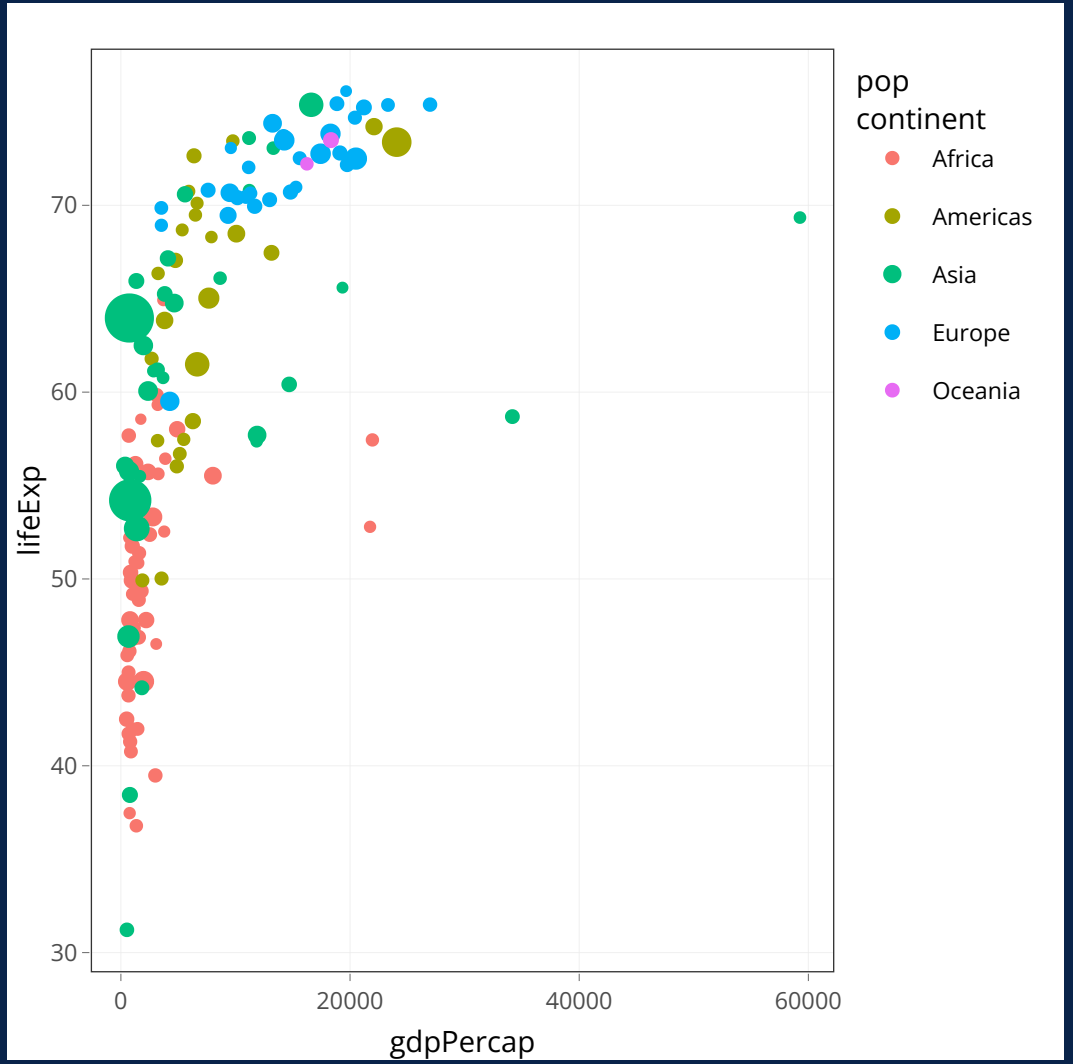
  * **width**: ancho (default es 480 pixeles)
  * **height**: alto (default es 480 pixeles)
  * **unit**: tamaño y unidad (“px”, “in”, “cm” y “mm”)
  * **pointsize**: tamaño del texto
  * **bg**: color de background inicial
  * **res**: resolución en pulgadas por píxel

- Ya creado el archivo en el directorio de trabajo, ejecuto los comandos que van a crear el plot.
- Finalmente, tengo que cerrar el graphic device. Se realiza con **dev.off()** o para todo tipo de gráficos **graphics.off()**.

Funciones gráficas interactivas

- En general se utilizan con data frames grandes.

```
# Cargo los paquetes
library(gplots)
library(ggplot2)
library(plotly)
library(gapminder)

# Genero el grafico
# Asigno a un objeto
p <- gapminder %>%
  filter(year==1977) %>%
  ggplot(aes(gdpPercap,
             lifeExp,
             size = pop,
             color=continent)
         ) +
  geom_point() +
  theme_bw()

# Plot del objeto
ggplotly(p)
```

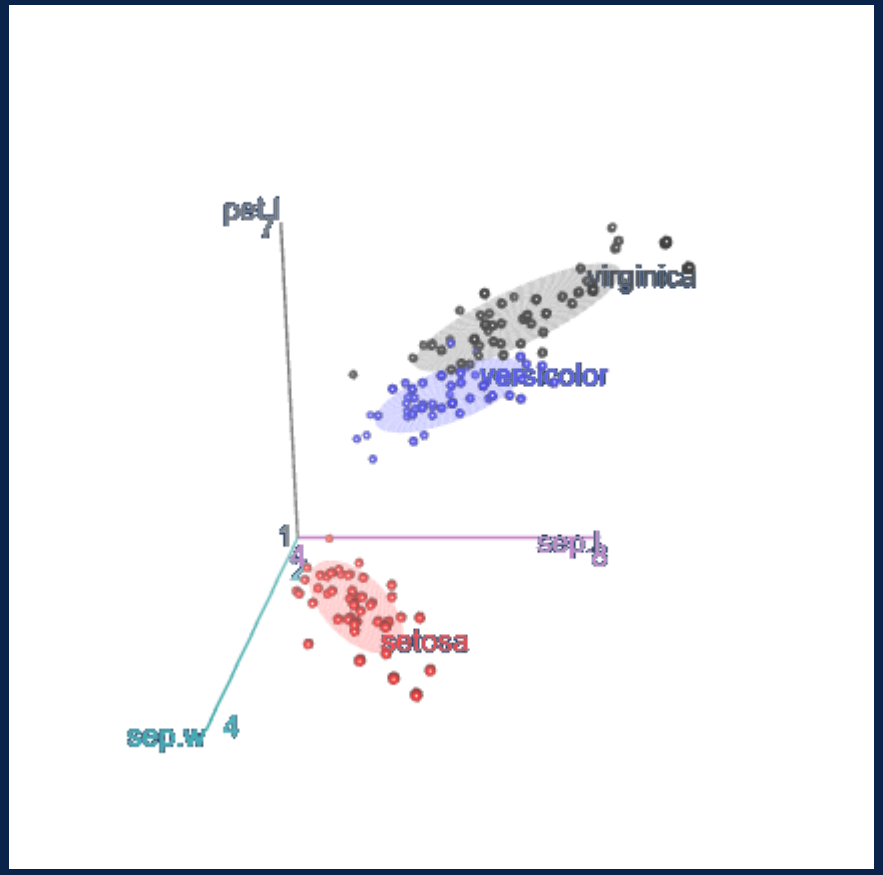


Funciones gráficas interactivas 3D

- En general se utilizan con la necesidad de respresentar más de 3 ejes, relieve, altura o profundidades.

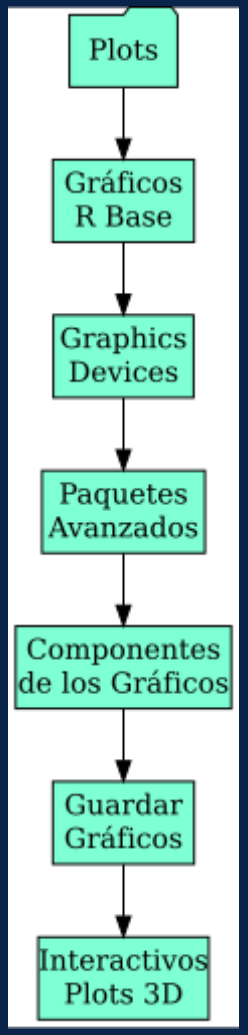
```
# Cargo los paquetes
library(car)
library(rgl)

# Uso 3 vectores
sep.l <- iris$Sepal.Length
sep.w <- iris$Sepal.Width
pet.l <- iris$Petal.Length

# Creo el grafico
p3d <- scatter3d(
  x = sep.l,
  y = pet.l,
  z = sep.w,
  groups = iris$Species,
  surface = FALSE,
  grid = FALSE,
  ellipsoid = TRUE,
  surface.col = c(
    "red", "blue", "black"
    )
  )

p3d
```

```
NULL
```



Resumen

Gráficos, Graphics devices, Exportación, Formatos, Argumentos, Base vs Tidyverse



QR - presentación

PDF Presentación: Clase 04 - Teoría

Fin. Clase 04 - Teoría

¡¡¡ Muchas gracias !!!

¿ Preguntas ? ... ¿ Consultas ?