

1. Graficación en R
2. graphics
3. ggplot2
4. Live coding

# Graficación en R



Podemos crear graficas en R de dos distintas maneras:

1. Usando paquete **graphics** (precargado en R)

- Conveniente para gráficas sencillas
- Difícil de usar para gráficas más complicadas/personalizadas

2. Usando el paquete **ggplot2** de tidyverse

- Requiere un poco más de trabajo
- Podemos hacer gráficas más complicadas
- Fácil (cuando ya aprendimos bien) de personalizar

graphics



## Funciones más comunes y usadas

Function	Description
<code>plot(x)</code>	Plot of the values of $x$ (on the $y$ -axis) ordered on the $x$ -axis
<code>plot(x, y)</code>	Bivariate plot of $x$ (on the $x$ -axis) and $y$ (on the $y$ -axis)
<code>pie(x)</code>	Circular pie chart
<code>boxplot(x)</code>	“Box-and-whiskers” plot
<code>stripchart(x)</code>	Plot of the values of $x$ on a line (an alternative to <code>boxplot()</code> for small sample sizes)
<code>coplot(x~y z)</code>	Bivariate plot of $x$ and $y$ for each value (or interval of values) of $z$
<code>dotchart(x)</code>	If $x$ is a data frame, plots a Cleveland dot plot (stacked plots line-by-line and column-by-column)
<code>fourfoldplot(x)</code>	Visualizes, with quarters of circles, the association between two dichotomous variables for different populations ( $x$ must be an array with <code>dim=c(2, 2, k)</code> , or a matrix with <code>dim=c(2, 2)</code> if $k=1$ )
<code>assocplot(x)</code>	Cohen-Friendly graph showing the deviations from independence of rows and columns in a two dimensional contingency table
<code>mosaicplot(x)</code>	‘Mosaic’ graph of the residuals from a log-linear regression of a contingency table
<code>pairs(x)</code>	If $x$ is a matrix or a data frame, draws all possible bivariate plots between the columns of $x$
<code>hist(x)</code>	Histogram of the frequencies of $x$
<code>barplot(x)</code>	Histogram of the values of $x$
<code>qqnorm(x)</code>	Quantiles of $x$ with respect to the values expected under a normal law
<code>qqplot(x, y)</code>	Quantiles of $y$ with respect to the quantiles of $x$
<code>contour(x, y, z)</code>	Contour plot (data are interpolated to draw the curves), $x$ and $y$ must be vectors and $z$ must be a matrix so that <code>dim(z)=c(length(x), length(y))</code> ( $x$ and $y$ may be omitted)
<code>symbols(x, y, ...)</code>	Draws, at the coordinates given by $x$ and $y$ , symbols (circles, squares, rectangles, stars, thermometers or “boxplots”) which sizes, colors, etc, are specified by supplementary arguments
<code>termplot(mod.obj)</code>	Plot of the (partial) effects of a regression model ( <code>mod.obj</code> )

# Parámetros para personalizar funciones

Agregar título y nombre a los ejes

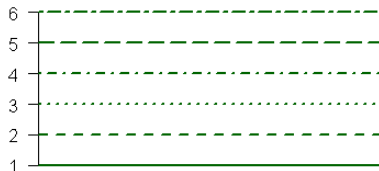
- `main` = "Título"
- `xlab` = "Nombre eje x"
- `ylab` = "Nombre eje y"

Personalizar

- `pch` = tipo de punto (número del 1 al 25)
- `lty` = tipo de línea (número del 1 al 6)
- `col` = "color"



**Line Types: lty=**



## R colors

[illegible]

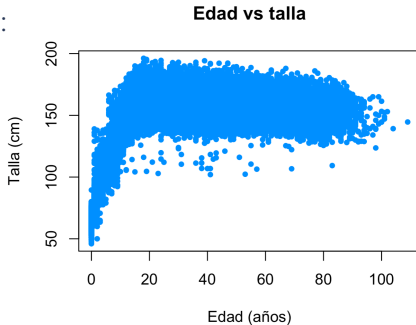
**Nota:** Podemos usar números, nombres o códigos hexadecimales

# Ejemplo

El código:

```
> plot(edad, talla,  
+      main = "Edad vs talla", xlab = "Edad (años)",  
+      ylab = "Talla (cm)", col = "dodgerblue",  
+      pch= 20)
```

Nos devuelve la gráfica:





ggplot2



# Paquete ggplot2

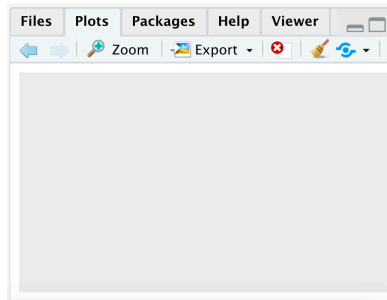
La idea principal es ir trabajando en capas:

1. Lienzo:

El código

```
> ggplot()
```

Nos devuelve

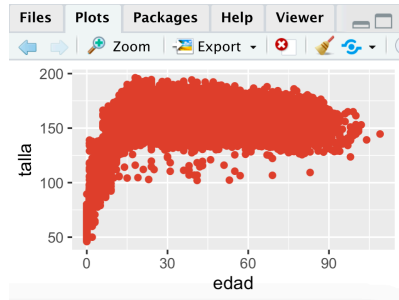


2. Agrego siguiente capa con un símbolo de +

El código

```
> ggplot() +  
+   geom_point(data = Antropometria,  
+             aes(x = edad, y = talla),  
+             color = "tomato3")  
_
```

Nos devuelve

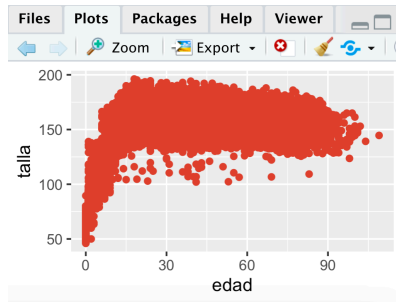


2. Agrego siguiente capa con un símbolo de +

El código

```
> ggplot() +  
+   geom_point(data = Antropometria,  
+             aes(x = edad, y = talla),  
+             color = "tomato3")  
_
```

Nos devuelve



# Template básico de una gráfica

En general, el template para ggplot siempre se ve así:

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```



En el ejemplo anterior, usamos la función *geom\_point*, pero podríamos usar por ejemplo líneas (*geom\_line*) o barras (*geom\_bar*)

Dentro de todas las (*geom\_functions*) debemos agregar la función (*aes()*) para indicar las variables que queremos graficar, y también las *aesthetics* que son propiedades visuales de los objetos en la gráfica. Estas propiedades incluyen cosas como el tamaño, la forma o el color de los puntos/lineas/barras etc.

En el ejemplo anterior:

```
> ggplot(data = Antropometria) +  
+   geom_point(aes(x = edad, y = talla),  
+               color = "tomato3")
```

Live coding

---

- P. Kuhnert & B. Venables, An Introduction to R: Software for Statistical Modeling & Computing
- Golemund, G., & Wickham, H. (2017). R for Data Science. O'Reilly Media.  
*<https://r4ds.had.co.nz/>*
- *<https://www.tidyverse.org/>*
- *<https://ggplot2.tidyverse.org/>*