

# Tareas varias voluntarias

David Cabezas Berrido

## 1 Resolución de sistemas de ecuaciones en R

La función `solve` de R, sirve para resolver sistemas de ecuaciones lineales. En concreto, ecuaciones del tipo  $ax = b$ , donde  $a$  es una matriz cuadrada real o compleja (la matriz de coeficientes del sistema) y  $b$  una matriz o un vector real o complejo. La función `solve(a,b)` devuelve la matriz o vector  $x$  que resuelve el sistema. Si se omite el parámetro  $b$ , se toma la matriz identidad, por lo que la función devuelve la inversa de  $a$ .

```
> a=matrix(c(1+1i, -1i, 2, 3i), nrow = 2)
> a
      [,1] [,2]
[1,] 1+1i 2+0i
[2,] 0-1i 0+3i
> b=matrix(c(0, -1+1i), nrow=2)
> b
      [,1]
[1,] 0+0i
[2,] -1+1i
> x=solve(a,b)
> x
      [,1]
[1,] -0.4705882-0.1176471i
[2,] 0.1764706+0.2941176i
> a%%x
      [,1]
[1,] 5.551115e-17+0e+00i
[2,] -1.000000e+00+1e+00i
```

Figure 1: Ejemplo de uso de `solve` para resolver un sistema lineal complejo de dos ecuaciones.

En este ejemplo definimos una matriz compleja  $2 \times 2$   $a$ , un vector complejo  $b$  con 2 componentes, llamamos a la función `solve` para resolver el sistema y asignamos el resultado a la variable  $x$ . Después comprobamos que efectivamente  $x$  satisface la ecuación  $ax = b$ .

## 2 El paquete gráfico ggplot2

`ggplot2` es un paquete gráfico de R. Según <https://ggplot2.tidyverse.org/index.html>, está basado en el libro *The Grammar of Graphics*, de Leland Wilkinson. En la página <https://www.rdocumentation.org/packages/ggplot2/versions/3.3.2> se encuentra una guía de instalación, una Cheatsheet, una descripción de las funciones del paquete y algunos tutoriales para dominarlo.

Probamos y explicamos un pequeño ejemplo encontrado en:

<https://www.datanovia.com/en/lessons/introduction-to-ggplot2/>, en el que representamos en un diagrama de dispersión (esto lo hacemos con `aes(x,y)`) en el que representamos la longitud (eje X) y anchura (eje Y) de los pétalos de las distintas flores. Elegimos un color y una forma diferentes para cada clase. Seleccionamos manualmente los colores en formato RGB hexadecimal: `#FF0000` sería el rojo más puro e intenso posible, `#00FF00` el verde más puro y `#0000FF` el azul puro.

```
1 library(ggplot2) # Cargamos el paquete
2
3 # Elegimos las variables a representar en cada eje
4 ggplot(iris, aes(x = Petal.Length, y = Petal.Width))+
5   # Para cada clase (especie) utilizamos un color y una forma
6   geom_point(aes(color = Species, shape = Species))+
7   # Elegimos los colores manualmente: Rojo, Verde y Azul
8   scale_color_manual(values = c("#FF0000", "#00FF00", "#0000FF"))
```

Figure 2: Ejemplo de gráfico con ggplot2.

Este es el gráfico que generamos:

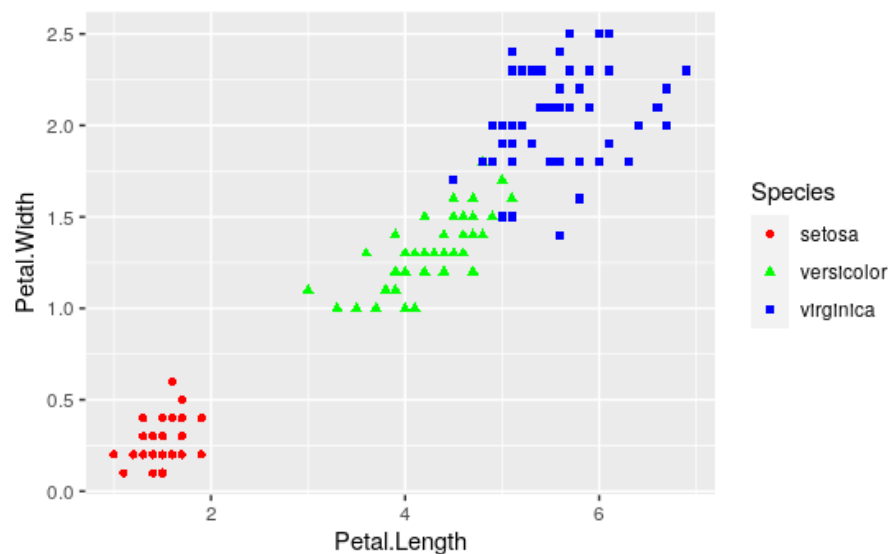


Figure 3: Gráfico generado