

ProgM Lab02

Javier Ortín

2026-01-28

Repaso de R

Manejo de datos

Simulamos una base de datos...

```
datos = rbind(c(4,15,101),  
              c(11,28,132)); datos
```

```
##      [,1] [,2] [,3]  
## [1,]    4   15  101  
## [2,]   11   28  132
```

```
colnames(datos) = c("Edad", "Peso", "Altura"); datos
```

```
##      Edad Peso Altura  
## [1,]    4   15   101  
## [2,]   11   28   132
```

```
datos[,1]
```

```
## [1]  4 11
```

```
mean(datos[, "Edad"])
```

```
## [1] 7.5
```

#Añadimos una nueva columna

```
datos2 = cbind(datos, Colegio=c("Colegio1", "Colegio2")); datos2
```

```
##      Edad Peso Altura Colegio  
## [1,] "4"   "15" "101"  "Colegio1"  
## [2,] "11"  "28" "132"  "Colegio2"
```

```
mean(datos2[, "Edad"])
```

```
## Warning in mean.default(datos2[, "Edad"]): argument is not numeric or logical:  
## returning NA
```

```
## [1] NA
```

La penúltima línea modifica los tipos de datos a string. No es un comportamiento adecuado. Como las matrices no son heterogéneas, `\texttt{datos2}` convierte sus elementos a `\texttt{chr}` para mantener un único tipo. Es por ello que la última línea produce un comportamiento inesperado. Es preferible trabajar con `\texttt{dataframe}` para este manejo de datos.

```
datos = as.data.frame(datos) #a partir de una matriz  
#desde cero  
datos = data.frame(Edad=c(4,11),
```



```

        Peso=c(15,18),
        Altura=c(101,132))
View(datos)

```

La última línea abre una pestaña en RStudio para examinar los contenidos de `\texttt{datos}`. Podemos seguir operando con el `\texttt{dataframe}` de manera similar a cómo lo hacíamos con las matrices.

```

datos[,1]

## [1] 4 11
datos[, "Edad"]

```

```

## [1] 4 11
datos$Altura

## [1] 101 132

```

En este caso sí se puede añadir una columna nueva por asignación directa. Sigue siendo posible usar `\texttt{cbind}`, aunque puede dar lugar a duplicados. Para remediar esto, podemos asignarle un valor nulo: por defecto elimina solo una ocurrencia.

```

datos$Colegio = c("Colegio1", "Colegio2")
datos = cbind(datos, Colegio=c("Colegio1", "Colegio2"))
datos #Columna duplicada

```

```

##   Edad Peso Altura  Colegio  Colegio
## 1    4   15   101 Colegio1 Colegio1
## 2   11   18   132 Colegio2 Colegio2

```

```

datos$Colegio = NULL
datos #sin duplicado

```

```

##   Edad Peso Altura  Colegio
## 1    4   15   101 Colegio1
## 2   11   18   132 Colegio2

```

Si volviésemos a asignar `\texttt{NULL}` a `\texttt{datos$Colegio}` nos quedaríamos sin esta columna. Podemos acceder a individuos indexando como con matrices:

```

datos[1,]

```

```

##   Edad Peso Altura  Colegio
## 1    4   15   101 Colegio1

```

```

datos[which.min(datos$Edad)]

```

```

##   Edad
## 1    4
## 2   11

```

```

datos[which(datos$Edad > 10)]

```

```

##   Peso
## 1   15
## 2   18

```

```

subset(datos, Edad>10 | Altura < 130) #filtrado con "or"

```

```

##   Edad Peso Altura  Colegio
## 1    4   15   101 Colegio1

```



```
## 2 11 18 132 Colegio2
```

Podemos añadir una nueva observación tanto con asignación directa como con `rbind`:

```
datos2 = rbind(datos, c(12,29,133,"Colegio1")); datos2
```

```
## Edad Peso Altura Colegio
## 1 4 15 101 Colegio1
## 2 11 18 132 Colegio2
## 3 12 29 133 Colegio1
```

Todos los campos de `datos2` son del tipo `chr`, pues aunque el `dataframe` puede tener tipos distintos, el vector que se le une no. Aunque se pueden añadir datos con `rbind`, es preferible hacerlo con otro `dataframe`:

```
datos2 = rbind(datos, data.frame(
  Edad=12, Peso=29, Altura=133, Colegio="Colegio1")); datos2
```

```
## Edad Peso Altura Colegio
## 1 4 15 101 Colegio1
## 2 11 18 132 Colegio2
## 3 12 29 133 Colegio1
```

En este caso sí se mantiene la integridad de los tipos.

Podemos hacer transformaciones de nuestros datos:

```
datos2 = transform(datos, Var=Edad*Altura); datos2
```

```
## Edad Peso Altura Colegio Var
## 1 4 15 101 Colegio1 404
## 2 11 18 132 Colegio2 1452
```

##Tidyverse y tibbles Los tibbles son un tipo de dataframe.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
## Warning: package 'ggplot2' was built under R version 4.4.3
## Warning: package 'tidyr' was built under R version 4.4.3
## Warning: package 'readr' was built under R version 4.4.3
## Warning: package 'purrr' was built under R version 4.4.3
## Warning: package 'dplyr' was built under R version 4.4.3
## Warning: package 'stringr' was built under R version 4.4.3
## Warning: package 'forcats' was built under R version 4.4.3
## Warning: package 'lubridate' was built under R version 4.4.3

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.6
## v forcats    1.0.1      v stringr    1.6.0
## v ggplot2    4.0.1      v tibble     3.2.1
## v lubridate  1.9.4      v tidyr      1.3.1
## v purrr      1.2.0
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```



```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
data("starwars")
?dplyr #recomendado por la profesora

## starting httpd help server ... done
?magrittr #>%
```

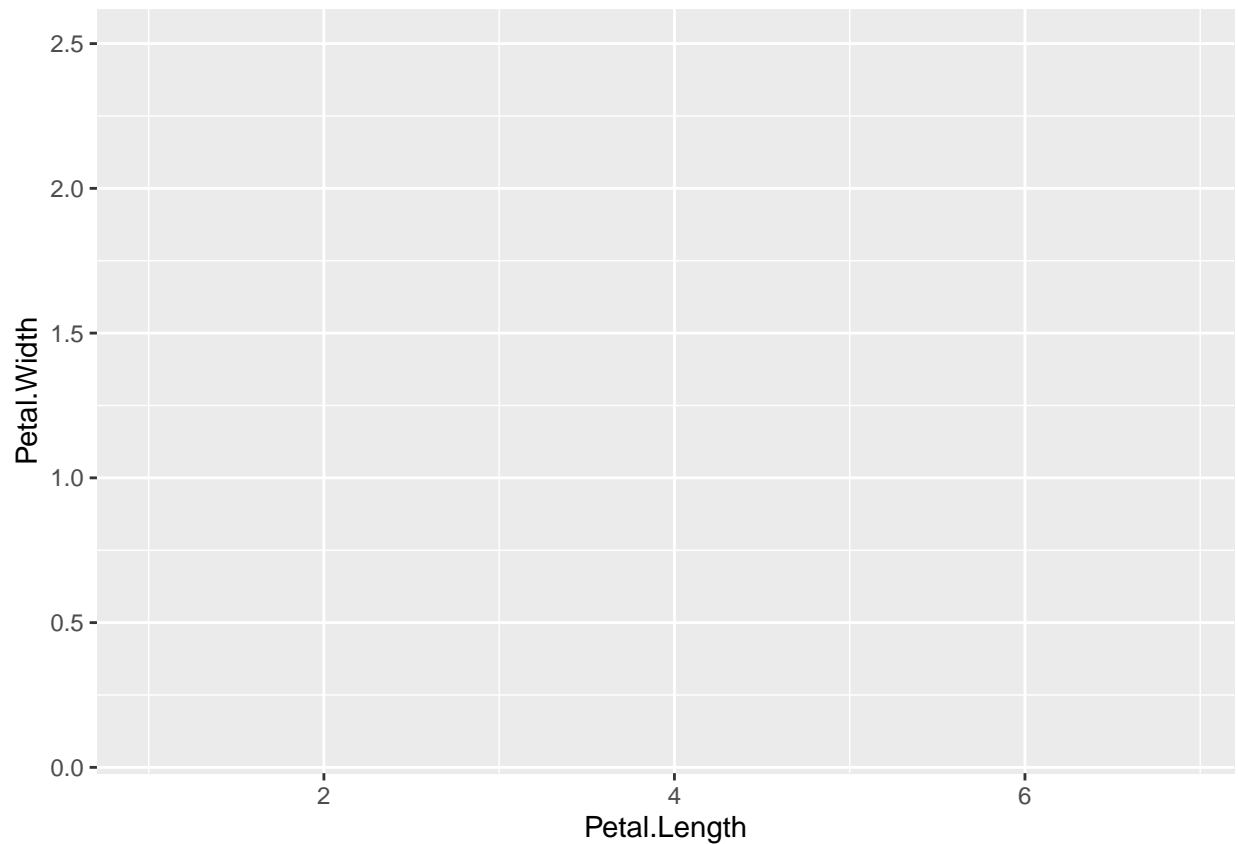
Gráficos en R

Al haber cargado \texttt{tidyverse}, ya tenemos activo \texttt{ggplot2}.

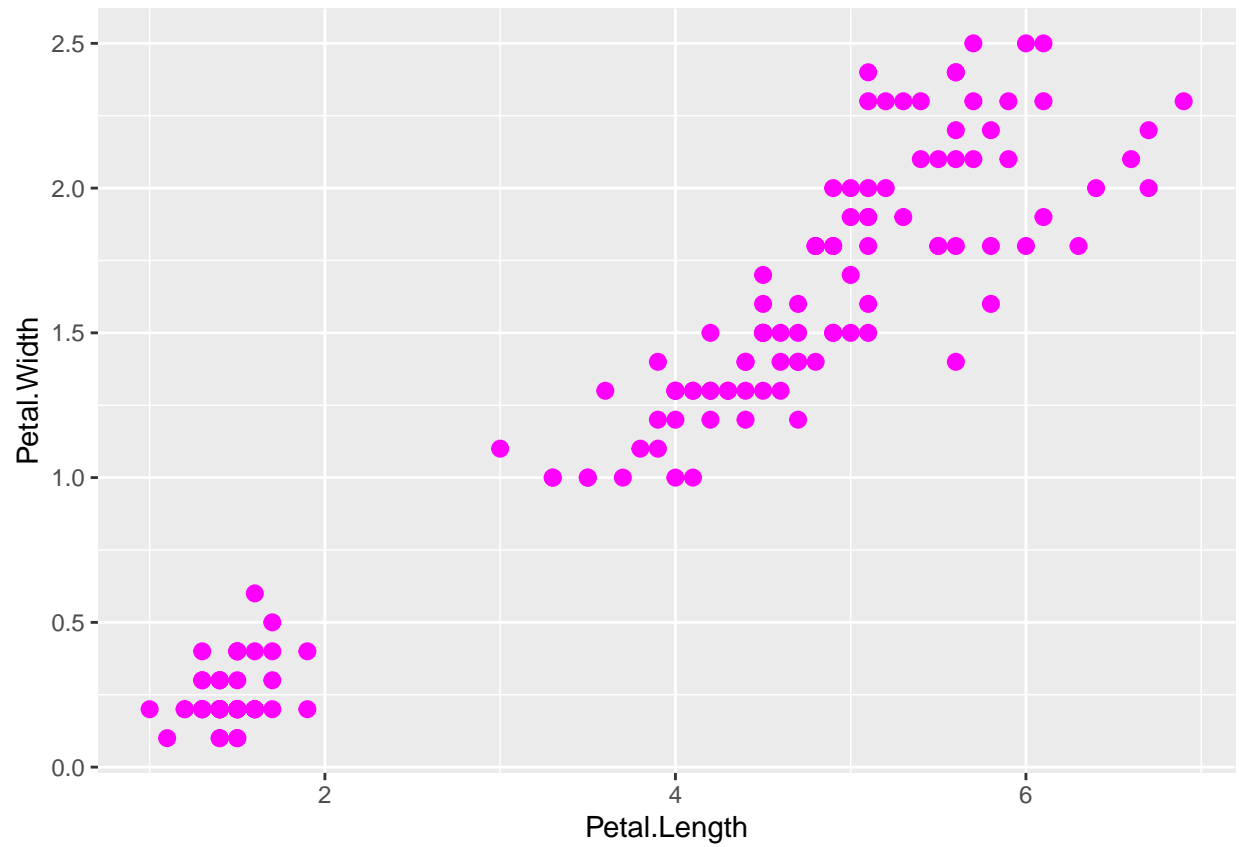
```
data(iris) # Conjunto de datos de ejemplo
?iris # Se usa para clasificación

p = ggplot(iris) # de primeras crea una lista de 11 variables
#visualmente, genera una gráfica en blanco (en este punto)

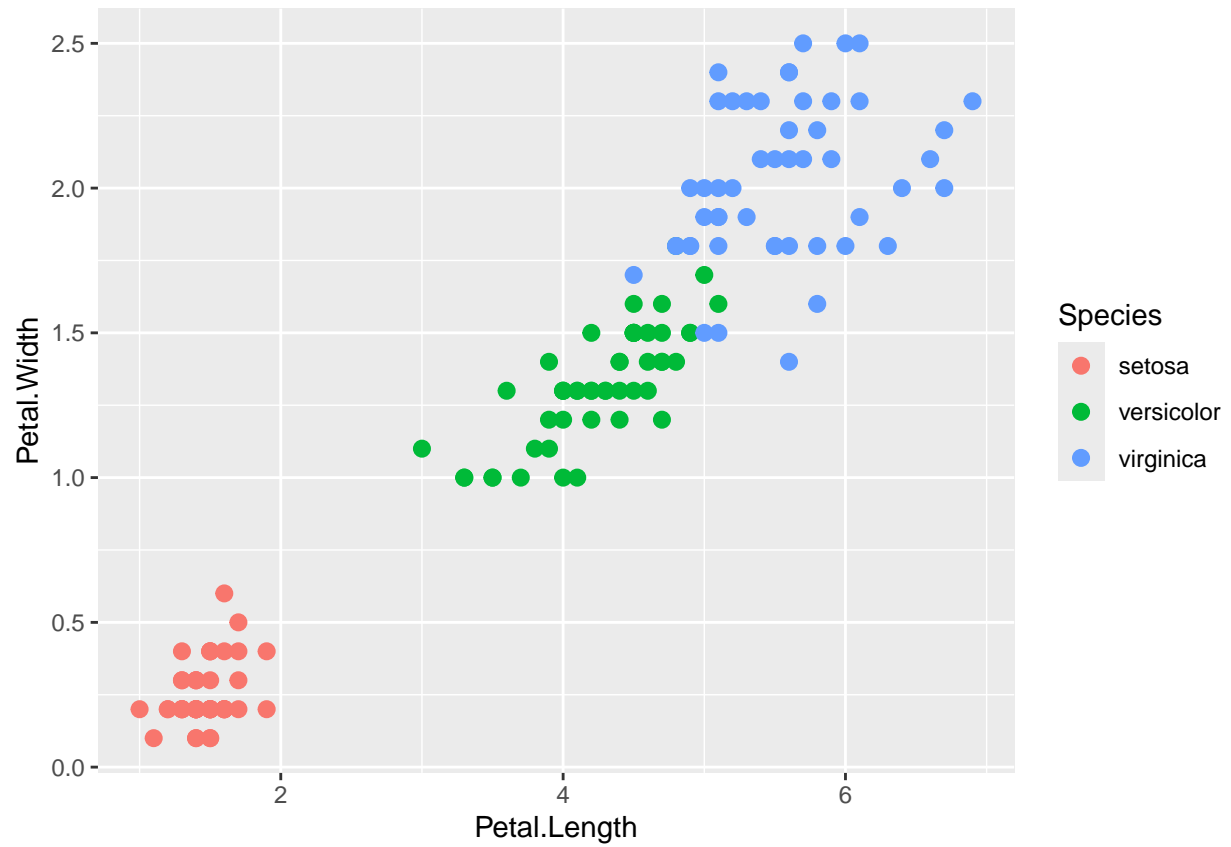
p = p + aes(x=Petal.Length, y=Petal.Width); p #ya genera la cuadrícula y los ejes etiquetados
```



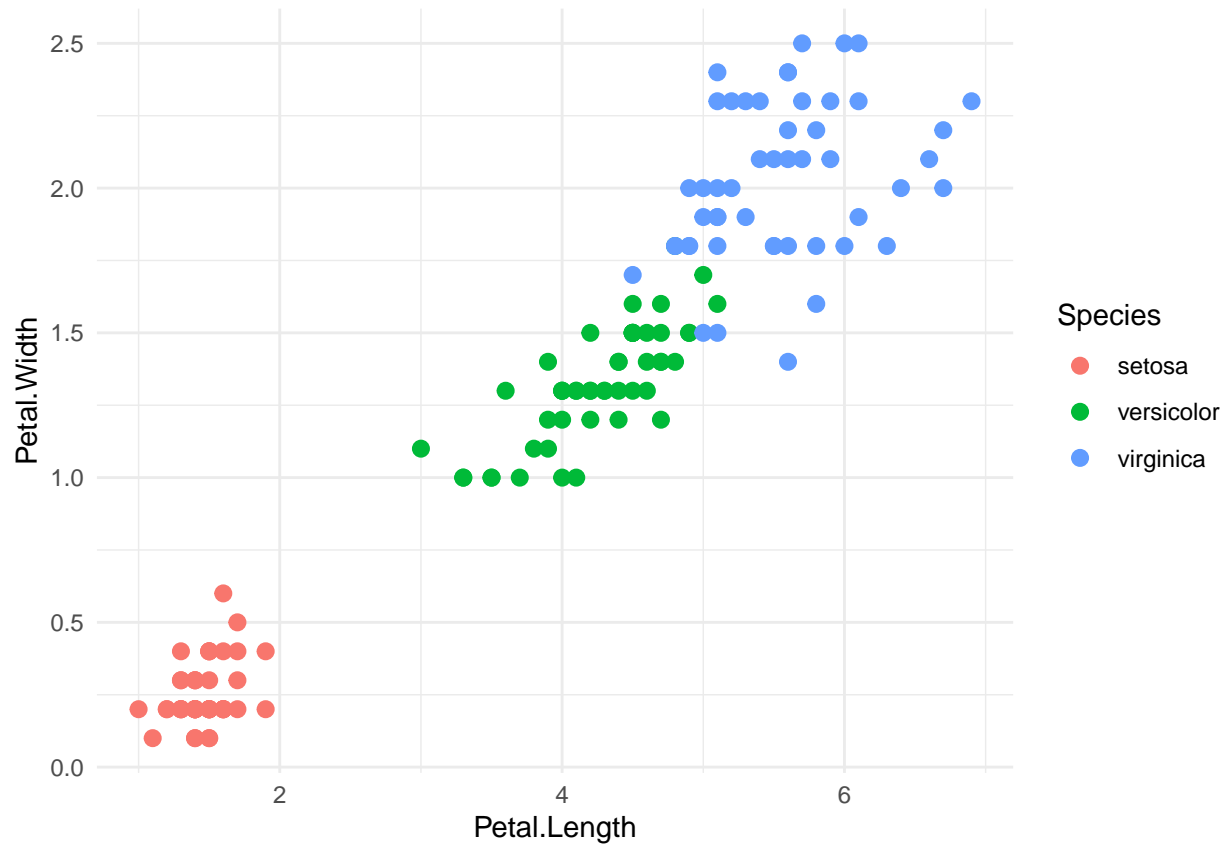
```
p + geom_point(size=2.5, col="magenta"); #ya tenemos cómo se relacionan las variables
```

```
# ¿Cómo podemos hacer que se coloreen los puntos según las especies de plantas?  
p = p + aes(col=Species) + geom_point(size=2.5); p
```

```
#Podemos cambiar el tema  
p + theme_minimal() # theme_bw() es otra opción
```

Addins de R

Se instalan como una librería de R, pero no se tienen que cargar. Una vez instalados, aparecen en el menú siempre. Podemos cargar, por ejemplo, un asistente de `\texttt{ggplot}`.

```
# ejecutar si no se tiene:  
#install.packages("ggThemeAssist")
```

Aparece una interfaz visual para configurar el tema de los gráficos.