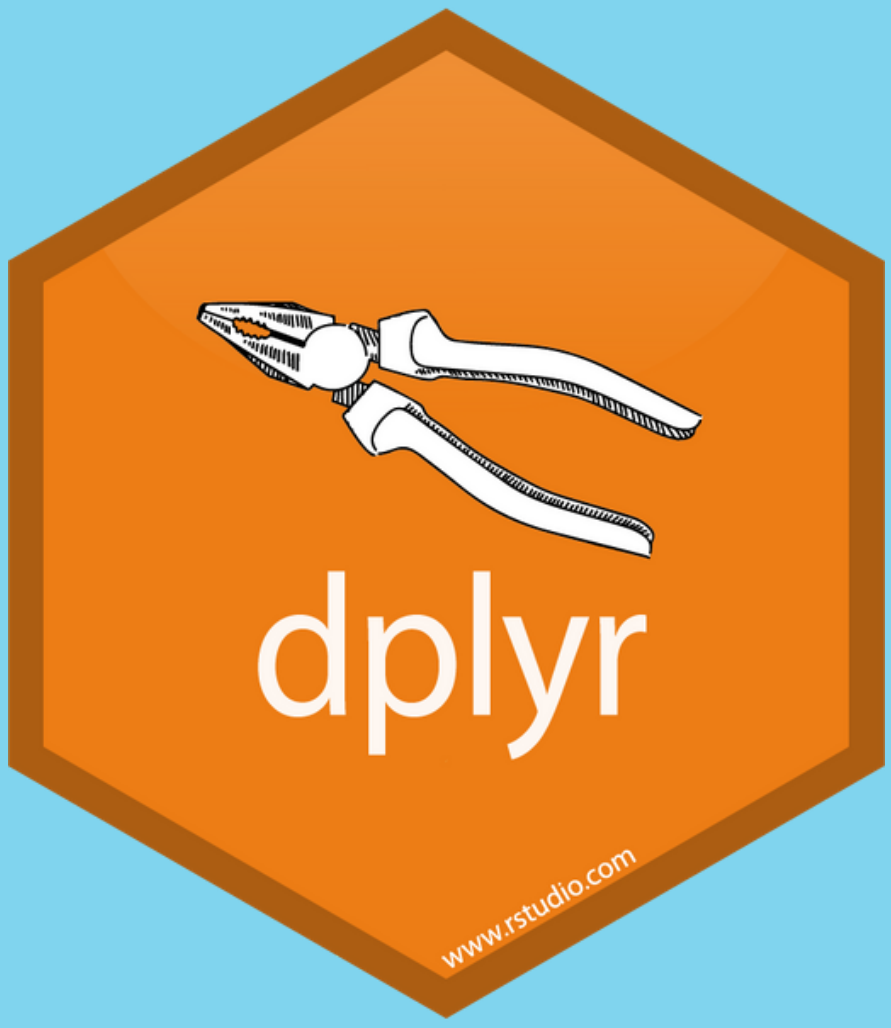
6

DATA SCIENCE

COLECCIONABLES

MAXIMA formación

GUÍA DE INTRODUCCIÓN



APRENDE R SOFTWARE

AGOSTO 2019

www.maximaformacion.es

Qué es dplyr

La "gramática" para la manipulación de datos con R Software

DPLYR

Es una librería de R orientada a la manipulación de datos.

Cada función de esta librería realiza una sola tarea y sus nombres son verbos lo cual hace más sencillo comprender y recordar lo que hace cada función.

A su vez r**esulta más simple desmenuzar en pequeños pasos las operaciones** que queremos realizar.

Además esta es parte del **tidyverse** y se integra perfectamente con las librerías que forman parte de este, como pueden ser **tidyr** o **ggplot2**.

En esta guía introductoria vamos a explicar las siguientes **funciones**:

- **filter()**
- **arrange()**
- **select()**
- **mutate()**
- **summarise()**
- **%>%**

Conociéndolas podremos enfrentarnos a las selecciones de datos y resúmenes estadísticos más comunes.

"El paquete dplyr fue desarrollado por Hadley Wickham de RStudio y es un versión mejorada de su paquete plyr."

Paso previo al estudio de funciones

Vamos a estudiar estas funciones aplicándolas al conjunto de datos ***mtcars***, al ser muy sencillo y conocido nos permitirá centrarnos en el funcionamiento de las funciones y no en las distintas peculiaridades de estas.

Para cargarlo debemos ejecutar:

```
data(mtcars)
```

Observamos sus primeras filas mediante:

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Si quieres obtener información sobre lo que representa cada columna puedes hacer uso de la ayuda de R:

```
?mtcars
```

La ejecución del anterior comando abrirá una ventana con información sobre el conjunto de datos.

Una vez que hemos cargado y conocemos nuestro conjunto de datos, vamos a proceder a explicar la instalación y funcionamiento de *dplyr*.

Siguiente paso: instalar dplyr

Instala y carga dplyr

La librería dplyr se encuentra en los repositorios por defecto de *R CRAN* por lo que su instalación es muy sencilla de realizar.

Solo tenemos que ejecutar:

```
install.packages("dplyr")
```

A continuación siempre que queramos utilizar *dplyr* solo tendremos que cargarla mediante:

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

El mensaje de salida que muestra la carga de la librería es solo un aviso de que hay funciones de *dplyr* cuyos nombres coinciden con las funciones de otras librerías y ahora su funcionamiento irá conforme a esta librería.

Bien, una vez instalada y cargada nuestra librería vamos a estudiar función a función su funcionamiento.

Siguiente paso: función *filter*

Función filter

• filter()

La función filter nos permite **seleccionar las filas que cumplan con las condiciones que indicamos a la función.**

Por ejemplo:

```
filter(mtcars, hp == 110, gear == 4)

##    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## 1  21   6  160 110  3.9 2.620 16.46  0  1    4    4
## 2  21   6  160 110  3.9 2.875 17.02  0  1    4    4
```

De esta manera hemos obtenido las filas que contienen valores para los coches:

- Con 110 caballos de potencia,
- y 4 marchas.

Siguiente paso: función *arrange*

Función arrange

- **arrange()**

La función *arrange* nos permite **ordenar los datos en función a los valores de una variable.**

Si queremos mostrar ordenados de menor a mayor los valores de nuestro conjunto de datos en función a los caballos de potencia ejecutaremos lo siguiente:

```
arrange(mtcars, hp)

##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1  30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## 2  24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## 3  33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## 4  32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## 5  27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## 6  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## 7  22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 8  22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## 9  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## 10 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## 11 21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## 12 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 13 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 14 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 15 30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## 16 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 17 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## 18 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## 19 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## 20 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 21 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## 22 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## 23 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## 24 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## 25 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## 26 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## 27 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## 28 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## 29 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## 30 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## 31 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## 32 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
```

Si queremos **obtener los resultados de mayor a menor** utilizaremos la función dentro de *arrange*.

En este caso no mostraremos la salida de la función para ahorrar un poco de espacio, te a tí ver el resultado de la función.

```
arrange(mtcars, desc(hp))
```

Siguiente paso: función *select*

Función select

- **select()**

La función *select* nos **permite seleccionar columnas.**

Por ejemplo, si queremos seleccionar las columnas millas por galón (mpg), cilindrada (cyl) y caballos de potencia (hp):

```
select(mtcars, mpg, cyl, hp)

##                      mpg cyl  hp
## Mazda RX4           21.0   6 110
## Mazda RX4 Wag       21.0   6 110
## Datsun 710          22.8   4  93
## Hornet 4 Drive      21.4   6 110
## Hornet Sportabout   18.7   8 175
## Valiant             18.1   6 105
## Duster 360          14.3   8 245
## Merc 240D           24.4   4  62
## Merc 230            22.8   4  95
## Merc 280            19.2   6 123
## Merc 280C           17.8   6 123
## Merc 450SE          16.4   8 180
## Merc 450SL          17.3   8 180
## Merc 450SLC         15.2   8 180
## Cadillac Fleetwood  10.4   8 205
## Lincoln Continental 10.4   8 215
## Chrysler Imperial   14.7   8 230
## Fiat 128            32.4   4  66
## Honda Civic         30.4   4  52
## Toyota Corolla      33.9   4  65
## Toyota Corona       21.5   4  97
## Dodge Challenger    15.5   8 150
## AMC Javelin         15.2   8 150
## Camaro Z28          13.3   8 245
## Pontiac Firebird    19.2   8 175
## Fiat X1-9           27.3   4  66
## Porsche 914-2       26.0   4  91
## Lotus Europa        30.4   4 113
## Ford Pantera L      15.8   8 264
## Ferrari Dino        19.7   6 175
## Maserati Bora       15.0   8 335
## Volvo 142E          21.4   4 109
```

Si queremos seleccionar desde la columna millas por galón (mpg) hasta la columna caballos de potencia (hp) ejecutaremos lo siguiente:

```
select(mtcars, mpg:hp)
```

Si queremos obtener todas las columnas menos desde millas por galón (mpg) hasta la columna caballos de potencia (hp) ejecutaremos lo siguiente.

```
select(mtcars, -(mpg:hp))
```

Siguiente paso: función *mutate*

Función mutate

- **mutate()**

La función *mutate* nos permite **crear nuevas columnas que contengan cálculos a partir de las que ya tenemos.**

Por ejemplo sabemos que la variable peso (wt) está representada el libras divididas por 1000 y queremos obtener esta variable en kilogramos.
Y además queremos otra columna que represente el peso en kilogramos entre los caballos de potencia.

Obtendremos esto ejecutando:

```
mutate(mtcars, wt_kg = (wt / 2.2046) * 1000, wt_kg_caballos = wt_kg/hp)
```

Siguiente paso: función *summarise*

Función summarise

summarise()

La función *summarise* nos permite **crear resúmenes estadísticos para nuestros datos.**

Por ejemplo vamos a calcular la media y la mediana para las millas por galón recorridas (mpg):

```
summarise(mtcars, media_mpg = mean(mpg), mediana_mpg = median(mpg))

##   media_mpg mediana_mpg
## 1  20.09062        19.2
```

Esta función es muy útil combinarla con la *group_by* para obtener nuestros estadísticos de resumen en función a grupos contenidos en una variable.

Por ejemplo vamos a calcular la media y la mediana para las millas por galón recorridas (mpg) en función al número de cilindros de los coches (cyl):

```
summarise(group_by(mtcars, cyl), media_mpg = mean(mpg), mediana_mpg =
median(mpg))

## # A tibble: 3 x 3
##     cyl media_mpg mediana_mpg
##   <dbl>     <dbl>       <dbl>
## 1     4      26.7        26
## 2     6      19.7        19.7
## 3     8      15.1        15.2
```

Siguiente paso: función *pipes* %>%

Función pipes %>%

- **pipes %>%()**

La función %>% conocida como *pipe* nos **permite encadenar funciones** sin tener que ir creando variables para un uso temporal o sin tener que anidar las funciones.

También nos ayuda a pensar paso a paso las transformaciones que queremos aplicar.

```
# Paso a paso, creando variables
tmp <- group_by(mtcars, cyl, am)
resumen <- summarise(tmp,
                     media_mpg = mean(mpg),
                     desv_est = sd(mpg),
                     mediana_mpg = median(mpg),
                     iqr = IQR(mpg))
resumen
```

```
# Anidando funciones
summarise(group_by(mtcars, cyl, am),
          media_mpg = mean(mpg),
          desv_est = sd(mpg),
          mediana_mpg = median(mpg),
          iqr = IQR(mpg))
```

Mediante *pipes* lo anterior se haría de la siguiente manera:

```
mtcars %>%
  group_by(cyl, am) %>%
  summarise(media_mpg = mean(mpg),
            desv_est_mpg = sd(mpg),
            mediana_mpg = median(mpg),
            iqr_mpg = IQR(mpg))
```

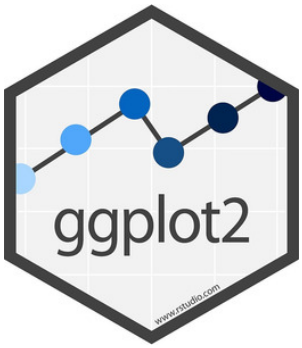
```
## # A tibble: 6 x 6
## # Groups:   cyl [3]
##     cyl    am media_mpg desv_est_mpg mediana_mpg iqr_mpg
##   <dbl> <dbl>     <dbl>        <dbl>       <dbl>   <dbl>
## 1     4     0      22.9        1.45         22.8   1.45
## 2     4     1      28.1        4.48         28.8   5.70
## 3     6     0      19.1        1.63         18.6   1.72
## 4     6     1      20.6        0.751        21     0.650
## 5     8     0      15.0        2.77         15.2   2.57
## 6     8     1      15.4        0.566        15.4   0.4
```

Como podéis observar, simplifica el pensar las operaciones y los pasos a seguir en la manipulación de datos.

De esta forma le decimos, toma *mtcars* y a continuación agrupa sus variables en función al número de cilindros (cyl) y del tipo de cambio (am) y finalmente haz el siguiente resumen estadístico.

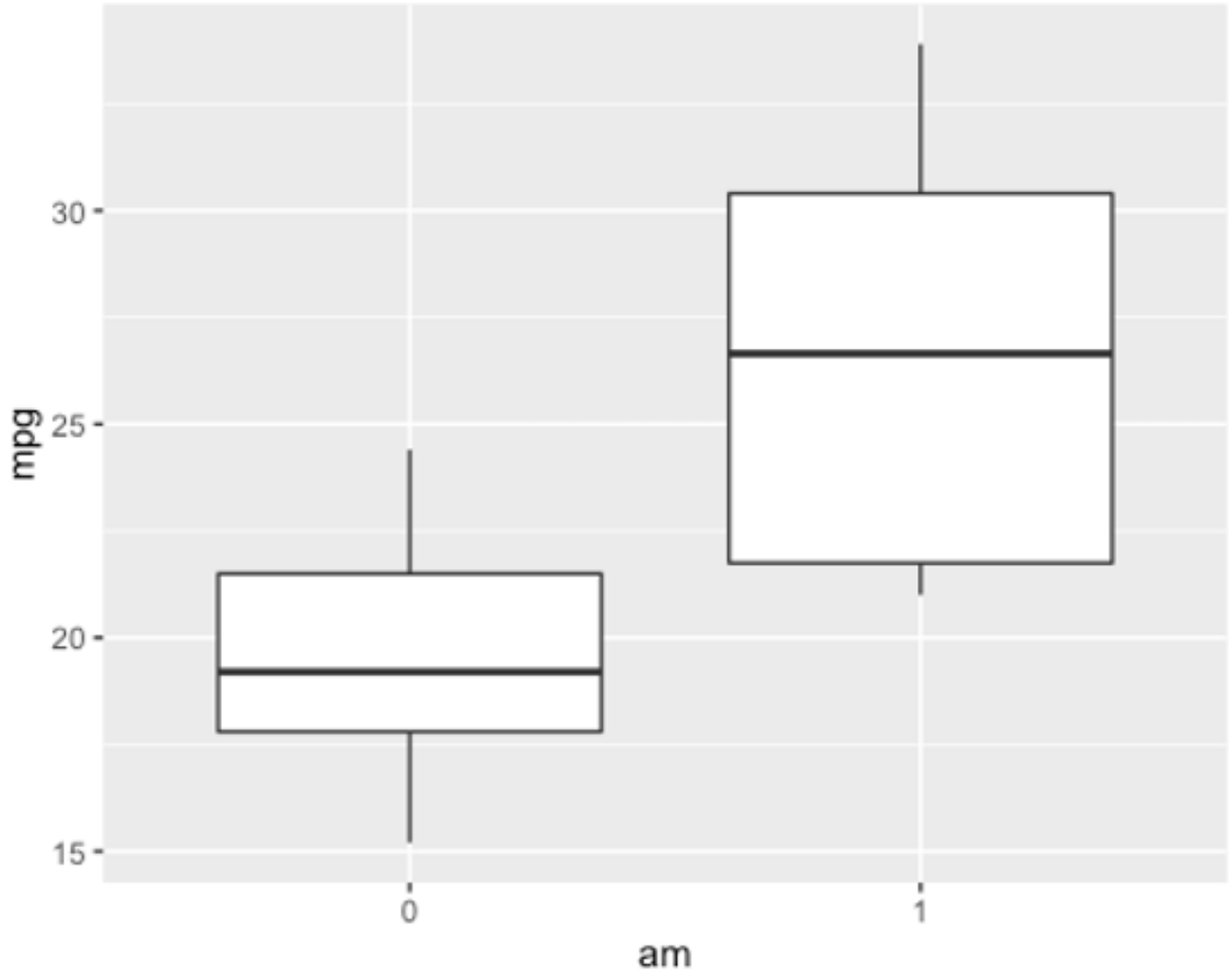
Veamos otro ejemplo, en este caso vamos a realizar un gráfico con *ggplot2* ayudándonos de *dplyr*.

Si no tenéis ggplot2 instalado podéis hacerlo ejecutando *install.packages("ggplot2")*.



```r
library(ggplot2) # Cargamos ggplot2 antes de realizar el gráfico.

mtcars %>%
  filter(hp <= 150) %>%
  mutate(am = as.factor(am)) %>%
  ggplot(aes(x = am, y = mpg)) + geom_boxplot()
```



En este caso le hemos indicado que tome *mtcars* y seleccione los datos de las filas que sean menores o iguales a 150 caballos (hp), que a continuación transforme en categórica la columna tipo de cambio (am) para que *ggplot* no la interprete como numérica y a continuación realiza un diagrama de cajas en función al tipo de cambio donde 0 es automático y 1 manual.

Si quieres más ejemplos introductorios del uso de esta librería , te recomiendo que eches un vistazo a los siguientes links:

- Introduction to *dplyr*
- Cap 4: Data Transformation en R for Data Science

MAXIMA formación

Guía didáctica

Máster de

Estadística Aplicada con R Software

Especialízate en Ciencia de Datos

IX edición

DATA SCIENCE

COLECCIONABLES

MÁXIMA formación

Con la colaboración de

MÁXIMA Consultoría