

# Biblioteca dplyr

Jose R. Guignan

2023-08-11

```
#library(magrittr) #tiene conflicto con otros paquetes  
library(dplyr) #produce un conflicto porque varios paquetes tienen los mismos nombre para sus funciones
```

```
##  
## 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
```

```
library(ggplot2)  
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v forcats   1.0.0      v stringr   1.5.0  
## v lubridate 1.9.2      v tibble   3.2.1  
## v purrr     1.0.2      v tidyr    1.3.0  
## v readr     2.1.4  
  
## -- 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("diamonds", package = "ggplot2" )  
head(diamonds)
```

```
## # A tibble: 6 x 10  
##   carat cut      color clarity depth table price      x      y      z  
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
## 1  0.23 Ideal     E      SI2     61.5    55   326   3.95   3.98   2.43  
## 2  0.21 Premium  E      SI1     59.8    61   326   3.89   3.84   2.31  
## 3  0.23 Good     E      VS1     56.9    65   327   4.05   4.07   2.31  
## 4  0.29 Premium  I      VS2     62.4    58   334   4.2    4.23   2.63  
## 5  0.31 Good     J      SI2     63.3    58   335   4.34   4.35   2.75  
## 6  0.24 Very Good J      VVS2     62.8    57   336   3.94   3.96   2.48
```

```
#select
#Quitar columnas
diamonds %>% select(-cut,-color) %>% head()
```

```
## # A tibble: 6 x 8
##   carat clarity depth table price     x     y     z
##   <dbl> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 SI2     61.5   55   326  3.95  3.98  2.43
## 2  0.21 SI1     59.8   61   326  3.89  3.84  2.31
## 3  0.23 VS1     56.9   65   327  4.05  4.07  2.31
## 4  0.29 VS2     62.4   58   334  4.2   4.23  2.63
## 5  0.31 SI2     63.3   58   335  4.34  4.35  2.75
## 6  0.24 VVS2    62.8   57   336  3.94  3.96  2.48
```

```
#Agarra la columna 2 y 7
diamonds %>% select(2,7) %>% head()
```

```
## # A tibble: 6 x 2
##   cut      price
##   <ord>    <int>
## 1 Ideal      326
## 2 Premium    326
## 3 Good       327
## 4 Premium    334
## 5 Good       335
## 6 Very Good  336
```

```
#Selecciona las columnas que comienzan con t
diamonds %>% select(starts_with("t")) %>% head()
```

```
## # A tibble: 6 x 1
##   table
##   <dbl>
## 1    55
## 2    61
## 3    65
## 4    58
## 5    58
## 6    57
```

```
#Selecciona las columnas que termine con h
diamonds %>% select(ends_with("h")) %>% head()
```

```
## # A tibble: 6 x 1
##   depth
##   <dbl>
## 1  61.5
## 2  59.8
## 3  56.9
## 4  62.4
## 5  63.3
## 6  62.8
```

```
#Selecciona las columnas que contenga a bl
diamonds %>% select(contains("bl")) %>% head()
```

```
## # A tibble: 6 x 1
##   table
##   <dbl>
## 1     55
## 2     61
## 3     65
## 4     58
## 5     58
## 6     57
```

```
#filter
```

```
#filtra los valores mayores de 500 en price
diamonds %>% filter(price > 500) %>% head()
```

```
## # A tibble: 6 x 10
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.35 Ideal  I     VS1     60.9   57   552  4.54  4.59  2.78
## 2  0.3  Premium D     SI1     62.6   59   552  4.23  4.27  2.66
## 3  0.3  Ideal  D     SI1     62.5   57   552  4.29  4.32  2.69
## 4  0.3  Ideal  D     SI1     62.1   56   552  4.3   4.33  2.68
## 5  0.42 Premium I     SI2     61.5   59   552  4.78  4.84  2.96
## 6  0.28 Ideal  G     VVS2     61.4   56   553  4.19  4.22  2.58
```

```
#filtra los valores mayores de 500 en price y cut ideal
diamonds %>% filter(price > 500 & cut=="Ideal") %>% head()
```

```
## # A tibble: 6 x 10
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.35 Ideal I     VS1     60.9   57   552  4.54  4.59  2.78
## 2  0.3  Ideal D     SI1     62.5   57   552  4.29  4.32  2.69
## 3  0.3  Ideal D     SI1     62.1   56   552  4.3   4.33  2.68
## 4  0.28 Ideal G     VVS2     61.4   56   553  4.19  4.22  2.58
## 5  0.32 Ideal I     VVS1     62    55.3  553  4.39  4.42  2.73
## 6  0.26 Ideal E     VVS2     62.9   58   554  4.02  4.06  2.54
```

```
txt <- c("casa","carro","bicicleta","moto")
#Da TRUE para la cadena de caracteres escogida
grepl(pattern = "mot", x=txt)
```

```
## [1] FALSE FALSE FALSE  TRUE
```

```
#filtra los valores que tienen w en la columna clarity
diamonds %>% filter(grepl(pattern = "VV", clarity)) %>% head()
```

```
## # A tibble: 6 x 10
##   carat cut      color clarity depth table price     x     y     z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.24 Very Good J      VVS2    62.8  57    336  3.94  3.96  2.48
## 2  0.24 Very Good I      VVS1    62.3  57    336  3.95  3.98  2.47
## 3  0.23 Very Good G      VVS2    60.4  58    354  3.97  4.01  2.41
## 4  0.28 Ideal    G      VVS2    61.4  56    553  4.19  4.22  2.58
## 5  0.32 Ideal    I      VVS1    62    55.3  553  4.39  4.42  2.73
## 6  0.24 Premium  E      VVS1    60.7  58    553  4.01  4.03  2.44
```

```
#rename
```

```
#cambia los nombre de las columnas
```

```
diamonds %>% rename(precio=price, corte=cut) %>% head()
```

```
## # A tibble: 6 x 10
##   carat corte      color clarity depth table precio     x     y     z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal    E      SI2    61.5  55    326  3.95  3.98  2.43
## 2  0.21 Premium  E      SI1    59.8  61    326  3.89  3.84  2.31
## 3  0.23 Good     E      VS1    56.9  65    327  4.05  4.07  2.31
## 4  0.29 Premium  I      VS2    62.4  58    334  4.2   4.23  2.63
## 5  0.31 Good     J      SI2    63.3  58    335  4.34  4.35  2.75
## 6  0.24 Very Good J      VVS2    62.8  57    336  3.94  3.96  2.48
```

```
#transform
```

```
#transforma los valores de una columna, las clasifica
```

```
diamonds %>% transform(price= ifelse(price>500, "Muy costoso", "Barato")) %>% head()
```

```
##   carat      cut color clarity depth table price     x     y     z
## 1  0.23    Ideal    E      SI2    61.5  55 Barato  3.95  3.98  2.43
## 2  0.21  Premium  E      SI1    59.8  61 Barato  3.89  3.84  2.31
## 3  0.23    Good    E      VS1    56.9  65 Barato  4.05  4.07  2.31
## 4  0.29  Premium  I      VS2    62.4  58 Barato  4.20  4.23  2.63
## 5  0.31    Good    J      SI2    63.3  58 Barato  4.34  4.35  2.75
## 6  0.24 Very Good  J      VVS2    62.8  57 Barato  3.94  3.96  2.48
```

```
#transforma los valores de una columna, las clasifica
```

```
diamonds %>% transform(price= case_when(
  price>500 & price<1000 ~ "Buen Precio",
  price<=500 ~ "Muy Barato",
  TRUE ~ "Muy Caro")) %>% head()
```

```
##   carat      cut color clarity depth table price     x     y     z
## 1  0.23    Ideal    E      SI2    61.5  55 Muy Barato  3.95  3.98  2.43
## 2  0.21  Premium  E      SI1    59.8  61 Muy Barato  3.89  3.84  2.31
## 3  0.23    Good    E      VS1    56.9  65 Muy Barato  4.05  4.07  2.31
## 4  0.29  Premium  I      VS2    62.4  58 Muy Barato  4.20  4.23  2.63
## 5  0.31    Good    J      SI2    63.3  58 Muy Barato  4.34  4.35  2.75
## 6  0.24 Very Good  J      VVS2    62.8  57 Muy Barato  3.94  3.96  2.48
```

```
#Da un resumen del dataset
glimpse(diamonds)
```

```
## Rows: 53,940
## Columns: 10
## $ carat    <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
## $ cut      <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
## $ color    <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
## $ clarity  <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
## $ depth    <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
## $ table    <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
## $ price    <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
## $ x        <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
## $ y        <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
## $ z        <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

```
#transforma la columna de clarity a caracter
diamonds %>% transform(clarity = as.character(clarity)) %>% head()
```

```
##   carat      cut color clarity depth table price     x     y     z
## 1  0.23    Ideal     E    SI2   61.5     55   326  3.95  3.98  2.43
## 2  0.21  Premium     E    SI1   59.8     61   326  3.89  3.84  2.31
## 3  0.23     Good     E    VS1   56.9     65   327  4.05  4.07  2.31
## 4  0.29  Premium     I    VS2   62.4     58   334  4.20  4.23  2.63
## 5  0.31     Good     J    SI2   63.3     58   335  4.34  4.35  2.75
## 6  0.24 Very Good     J   VVS2   62.8     57   336  3.94  3.96  2.48
```

```
#matute
```

```
#inserta nueva columna
diamonds %>% mutate(nueva.columna = 1)
```

```
## # A tibble: 53,940 x 11
##   carat cut      color clarity depth table price     x     y     z nueva.columna
##   <dbl> <ord>   <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>         <dbl>
## 1  0.23 Ideal     E    SI2   61.5     55   326  3.95  3.98  2.43             1
## 2  0.21 Premium   E    SI1   59.8     61   326  3.89  3.84  2.31             1
## 3  0.23 Good      E    VS1   56.9     65   327  4.05  4.07  2.31             1
## 4  0.29 Premium   I    VS2   62.4     58   334  4.2  4.23  2.63             1
## 5  0.31 Good      J    SI2   63.3     58   335  4.34  4.35  2.75             1
## 6  0.24 Very G~ J    VVS2   62.8     57   336  3.94  3.96  2.48             1
## 7  0.24 Very G~ I    VVS1   62.3     57   336  3.95  3.98  2.47             1
## 8  0.26 Very G~ H    SI1   61.9     55   337  4.07  4.11  2.53             1
## 9  0.22 Fair      E    VS2   65.1     61   337  3.87  3.78  2.49             1
## 10 0.23 Very G~ H    VS1   59.4     61   338  4    4.05  2.39             1
## # i 53,930 more rows
```

```
#inserta una nueva columna con la clasificacion de precios
diamonds %>% mutate(categoria_precio = case_when(
  price>500 & price<1000 ~ "Buen Precio",
  price<=500 ~ "Muy Barato",
  TRUE ~ "Muy Caro"))
```

```
## # A tibble: 53,940 x 11
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2     61.5   55   326   3.95   3.98   2.43
## 2  0.21 Premium  E     SI1     59.8   61   326   3.89   3.84   2.31
## 3  0.23 Good     E     VS1     56.9   65   327   4.05   4.07   2.31
## 4  0.29 Premium  I     VS2     62.4   58   334   4.2    4.23   2.63
## 5  0.31 Good     J     SI2     63.3   58   335   4.34   4.35   2.75
## 6  0.24 Very Good J     VVS2     62.8   57   336   3.94   3.96   2.48
## 7  0.24 Very Good I     VVS1     62.3   57   336   3.95   3.98   2.47
## 8  0.26 Very Good H     SI1     61.9   55   337   4.07   4.11   2.53
## 9  0.22 Fair     E     VS2     65.1   61   337   3.87   3.78   2.49
## 10 0.23 Very Good H     VS1     59.4   61   338   4      4.05   2.39
## # i 53,930 more rows
## # i 1 more variable: categoria_precio <chr>
```

```
#separate
df <- data.frame(x = c("x:1", "x:2", "y:4", "z", NA))

#separa el dataframe en dos, separando por :
df %>% separate(x, c("clave", "valor"), ":")
```

```
## Warning: Expected 2 pieces. Missing pieces filled with 'NA' in 1 rows [4].
```

```
##   clave valor
## 1     x     1
## 2     x     2
## 3     y     4
## 4     z  <NA>
## 5  <NA>  <NA>
```