

Group By en R

Jose R. Guignan

2023-08-14

```
library(magrittr)
library(dplyr)#tiene conflicto con otros paquetes
```

```
##
## 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(dplyr) #produce un conflicto porque varios paquetes tienen los mismos nombre para sus funciones
library(ggplot2)
```

```
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
```

```
#cuenta por la clasificacion que tiene cut
#metdo 1
diamonds %>% group_by(cut) %>% summarize(conteo = n())
```

```
## # A tibble: 5 x 2
##   cut      conteo
##   <ord>    <int>
## 1 Fair      1610
```

```
## 2 Good      4906
## 3 Very Good 12082
## 4 Premium   13791
## 5 Ideal     21551
```

```
#metodo 2
diamonds %>% group_by(cut) %>% tally()
```

```
## # A tibble: 5 x 2
##   cut      n
##   <ord>   <int>
## 1 Fair    1610
## 2 Good    4906
## 3 Very Good 12082
## 4 Premium 13791
## 5 Ideal   21551
```

```
#cuenta por la clasificacion de color de manera ascendente
diamonds %>% group_by(color) %>% tally() %>% arrange(desc(n))
```

```
## # A tibble: 7 x 2
##   color      n
##   <ord> <int>
## 1 G     11292
## 2 E     9797
## 3 F     9542
## 4 H     8304
## 5 D     6775
## 6 I     5422
## 7 J     2808
```

```
#Ordema de forma decreciente el promedio de la clasificacion de cut
diamonds %>% group_by(cut) %>% summarize(promedio=mean(price)) %>% arrange(desc(promedio))
```

```
## # A tibble: 5 x 2
##   cut      promedio
##   <ord>   <dbl>
## 1 Premium 4584.
## 2 Fair    4359.
## 3 Very Good 3982.
## 4 Good    3929.
## 5 Ideal   3458.
```

```
#valor relativo de la clasificacion de cut
diamonds %>% group_by(cut) %>% summarise(n=n()/nrow(.))
```

```
## # A tibble: 5 x 2
##   cut      n
##   <ord>   <dbl>
## 1 Fair    0.0298
## 2 Good    0.0910
```

```
## 3 Very Good 0.224
## 4 Premium   0.256
## 5 Ideal     0.400
```

```
#suma el valor relativo de la clasificacion de cut
diamonds %>% group_by(cut) %>% summarise(n=n()/nrow()) %>% select(n) %>% sum()
```

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

```
#cuenta por las clasificacion de cut y clarity
diamonds %>% group_by(cut,clarity) %>% tally()
```

```
## # A tibble: 40 x 3
## # Groups:   cut [5]
##   cut    clarity    n
##   <ord> <ord>   <int>
## 1 Fair   I1       210
## 2 Fair   SI2      466
## 3 Fair   SI1      408
## 4 Fair   VS2      261
## 5 Fair   VS1      170
## 6 Fair   VVS2       69
## 7 Fair   VVS1       17
## 8 Fair   IF         9
## 9 Good   I1        96
## 10 Good  SI2     1081
## # i 30 more rows
```

```
#saca el valor minimo y maximo de la clasificacion de cut
diamonds %>% group_by(cut) %>% summarize(Min_precio=min(price), Max_precio=max(price))
```

```
## # A tibble: 5 x 3
##   cut      Min_precio Max_precio
##   <ord>         <int>         <int>
## 1 Fair           337         18574
## 2 Good           327         18788
## 3 Very Good      336         18818
## 4 Premium        326         18823
## 5 Ideal          326         18806
```