

# Untitled

Victor Lopez

2023-01-21

## DATOS ORDINALES

Se suelen trabajar con factores ordenados. Para crear una factor ordenado se utiliza:

```
notas = ordered(c("S", "A", "N", "Ex", "S", "S", "Ex", "Ex", "N", "A", "A", "A", "A", "N", "S"),  
               levels = c("S", "A", "N", "Ex"))
```

Con los datos ordinales podemos hacer exactamente lo mismo que los cualitativos, lo unico que cambia, son las frecuencias absolutas y relativas acumuladas

### Frecuencia absoluta acumulada (unidimensional)

```
cumsum(table(notas))
```

```
##  S  A  N Ex  
##  4  9 12 15
```

### Frecuencia relativa acumulada (unidimensional)

```
cumsum(prop.table(table(notas)))
```

```
##           S           A           N           Ex  
## 0.2666667 0.6000000 0.8000000 1.0000000
```

### cumsum() en tablas multidimensionales

```
tam = sample(c("Muy.corto", "Corto", "Normal", "Largo", "Muy.largo"), size = 100, replace = TRUE)  
zones = rep(c("A", "B", "C", "D"), c(30, 25, 35, 10))
```

```
jirafas = data.frame(zones, tam)

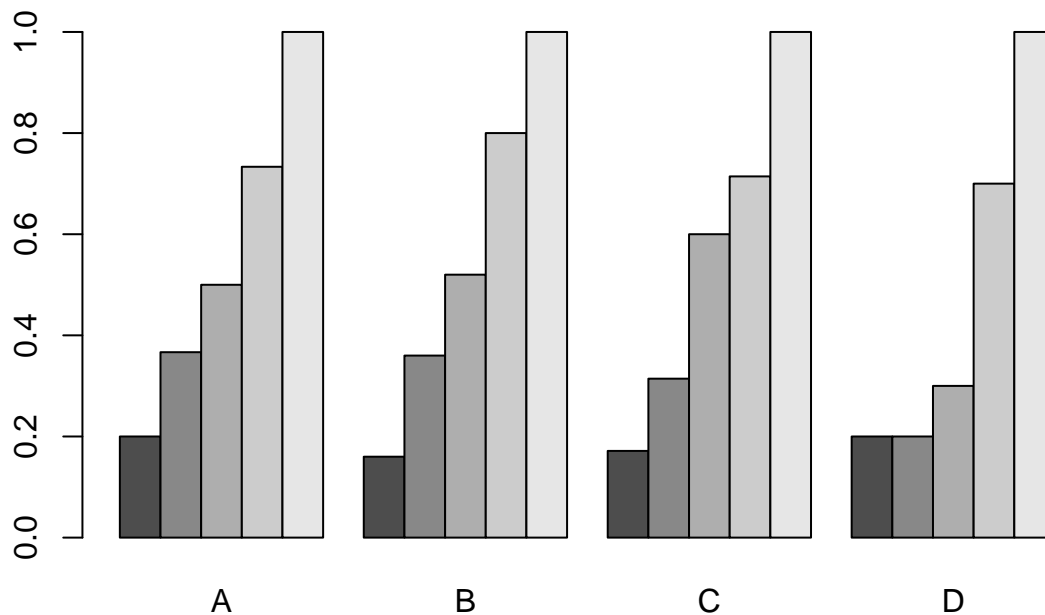
t = prop.table(table(jirafas), margin = 1)

t(apply(t, MARGIN = 1, FUN = cumsum))
```

```
##      tam
## zones   Corto   Largo Muy.corto Muy.largo Normal
##   A 0.2000000 0.3666667    0.50 0.7333333      1
##   B 0.1600000 0.3600000    0.52 0.8000000      1
##   C 0.1714286 0.3142857    0.60 0.7142857      1
##   D 0.2000000 0.2000000    0.30 0.7000000      1
```

*# Se le aplicara el cumsum() en las filas pero al resultado final de la tabla le hara un t().  
# Por eso nosotros le hacemos otro t() para regresarlo a su forma original*

```
barplot(apply(t, MARGIN = 1, FUN = cumsum), beside = TRUE)
```



## Convertir datos cuantitativos en factores ordinales

```
crabs = read.table("../data/datacrab.txt", header = TRUE)
crabs = crabs[,-1] # Todas menos la primera columna
str(crabs)
```

```
## 'data.frame': 173 obs. of 5 variables:
## $ color : int 3 4 2 4 4 3 2 4 3 4 ...
## $ spine : int 3 3 1 3 3 3 1 2 1 3 ...
## $ width : num 28.3 22.5 26 24.8 26 23.8 26.5 24.7 23.7 25.6 ...
## $ satell: int 8 0 9 0 4 0 0 0 0 0 ...
## $ weight: int 3050 1550 2300 2100 2600 2100 2350 1900 1950 2150 ...
```

```
intervalos = cut(crabs$width, breaks = c(21, 25, 29, 33, Inf), right = FALSE,
                labels = c("21-25", "25-29", "29-33", "33-..."))
# Crea un vector pero con los datos agrupados en intervalos, cada dato sera cambiado por el
# valor del intervalo que le corresponde. Con right = FALSE, estamos diciendo que, es de 21 a 25
# sin tomarse en cuenta el 25 y asi sucesivamente

crabs$width.rank = ordered(intervalos)
```