

Factores

Daniel Eduardo Macias Estrada

9/1/2021

Factor

Es como un vector, pero con una estructura interna más rica que permite usarlo para clasificar observaciones.

- `levels`: atributo del factor. Cada elemento del factor es igual a un nivel. Los niveles clasifican las entradas del factor. Se ordenan por orden alfabético.
- Para definir un factor, primero hemos de definir un vector y transformarlo por medio de una de las siguientes funciones `factor()` o `as.factor()`

Ejemplo

```
nombres = c("Juan", "Antonio", "Ricardo", "Juan", "Juan", "Maria", "Maria")
nombres
```

```
## [1] "Juan"      "Antonio" "Ricardo" "Juan"      "Juan"      "Maria"      "Maria"
```

```
nombres.factor = factor(nombres)
nombres.factor
```

```
## [1] Juan      Antonio Ricardo Juan      Juan      Maria      Maria
## Levels: Antonio Juan Maria Ricardo
```

La otra función a usar era `as.factor()`. La diferencia entre ambas es que `as.factor()`, convierte el vector que ya se tenía en factores, y toma como niveles los que aparecen dentro del factor. La función `factor()` define un factor a partir del vector, pero se le permite definir los niveles del factor

```
gender = c("M", "H", "H", "M", "M", "H", "H", "H", "M");
gender.fact = factor(gender, levels = c("M", "H", "B"))
gender.fact
```

```
## [1] M H H M M H H M
## Levels: M H B
```

```
gender.fact2 = as.factor(gender)
gender.fact2
```

```
## [1] M H H M M H H H M
## Levels: H M
```

```
#Usando labels
gender.fact3 = factor(gender, levels = c("M", "H", "B"), labels = c("Mujer", "Hombre", "Bi Sexo"))
gender.fact3
```

```
## [1] Mujer  Hombre Hombre Mujer  Mujer  Hombre Hombre Hombre Mujer
## Levels: Mujer Hombre Bi Sexo
```

Conocer los levels de un factor

Con la función `levels()` puedes conocer los niveles que pertenecen a un factor

```
levels(gender.fact)
```

```
## [1] "M" "H" "B"
```

```
levels(gender.fact3)
```

```
## [1] "Mujer" "Hombre" "Bi Sexo"
```

Con esta función a su vez, es posible modificar los levels de un factor

```
levels(gender.fact3) = c("Femenino", "Masculino", "Híbrido")
gender.fact3
```

```
## [1] Femenino Masculino Masculino Femenino Femenino Masculino Masculino
## [8] Masculino Femenino
## Levels: Femenino Masculino Híbrido
```

Otro ejemplo sería

```
notas = c(1,3,4,1,3,2,2,3,2,4,3,1)
notas.factor = factor(notas)
levels(notas.factor)
```

```
## [1] "1" "2" "3" "4"
```

```
levels(notas.factor) = c("Suspendido", "Suficiente", "Notable", "Excelente")
notas.factor
```

```
## [1] Suspendido Notable Excelente Suspendido Notable Suficiente
## [7] Suficiente Notable Suficiente Excelente Notable Suspendido
## Levels: Suspendido Suficiente Notable Excelente
```

Incluso es posible modificar la estructura y el número de levels

```
levels(notas.factor) = c("Suspendido", "Aprobado", "Aprobado", "Aprobado")
notas.factor
```

```
## [1] Suspendido Aprobado Aprobado Suspendido Aprobado Aprobado
## [7] Aprobado Aprobado Aprobado Aprobado Aprobado Suspendido
## Levels: Suspendido Aprobado
```

Factor ordenado

Es un factor donde los niveles siguen un orden

- **ordered(vector, levels=...)**: función que define un factor ordenado y tiene los mismos parámetros que factor

```
fac = factor(c(1,3,4,1,3,2,2,3,2,4,3,1), levels = c(1,2,3,4), labels = c("Sus","Apr","Not","Exc"))
fac
```

```
## [1] Sus Not Exc Sus Not Apr Apr Not Apr Exc Not Sus
## Levels: Sus Apr Not Exc
```

```
facOrd = ordered(c(1,3,4,1,3,2,2,3,2,4,3,1), levels = c(1,2,3,4), labels = c("Sus","Apr","Not","Exc"))
facOrd
```

```
## [1] Sus Not Exc Sus Not Apr Apr Not Apr Exc Not Sus
## Levels: Sus < Apr < Not < Exc
```