

forcats

Entornos de Análisis de Datos: R

Alberto Torres Barrán

2021-01-07

Factores

- Representan variables categóricas, con un número posible de valores:
 - fijo
 - *pequeño*
 - conocido
- Ejemplos: meses del año, provincias, países de la Unión Europea, etc.
- Muchas funciones de R convierten los vectores de cadenas de caracteres a factores automáticamente
- La librería `forcats` implementa funciones para manipular factores

Creando factores

- Función `factor()`
- Necesita dos argumentos:
 - Datos a convertir (típicamente vector de cadenas)
 - Opcionalmente también los valores posibles (**niveles**)

```
x <- c("Primavera", "Verano", "Verano", "Verano", "Otoño")
f <- factor(x, levels = c("Primavera", "Verano", "Otoño", "Invierno"))
f
## [1] Primavera Verano Verano Verano Otoño
## Levels: Primavera Verano Otoño Invierno
```

- Los niveles se pueden ver con la función `levels`

```
levels(f)
## [1] "Primavera" "Verano" "Otoño" "Invierno"
```

- Si un valor no está en los niveles se convierte a NA

```
x <- c("Primavera", "Verano", "Verano", "Verano", "Otono")
f1 <- factor(x, levels = c("Primavera", "Verano", "Otoño", "Invierno"))
f1
## [1] Primavera Verano Verano Verano <NA>
## Levels: Primavera Verano Otoño Invierno
```

- Si no se indican los niveles, se toman como los valores únicos del vector

```
unique(x)
## [1] "Primavera" "Verano" "Otono"
```

```
factor(x)
## [1] Primavera Verano Verano Verano Otono
## Levels: Otono Primavera Verano
```

Orden de los niveles

- Por defecto los niveles están en orden creciente:

```
sort(unique(x))  
## [1] "otoño"      "Primavera" "Verano"
```

- En la mayoría de los casos el orden de los factores es irrelevante
- Dos ejemplos donde importa el orden:
 1. realizar gráficos, el orden de las escalas es el orden de los niveles del factor
 2. realizar modelos, donde el primer nivel se considera el nivel de referencia

Función cut

- Crea un factor a partir de una variable numérica

```
edad <- c(10, 15, 20, 25, 38)
cut(edad, breaks = c(10, 20, 30, 40))
## [1] <NA>      (10,20] (10,20] (20,30] (30,40]
## Levels: (10,20] (20,30] (30,40]
```

- Incluir el valor de la izquierda del primer intervalo

```
cut(edad, breaks = c(10, 20, 30, 40), include.lowest = TRUE)
## [1] [10,20] [10,20] [10,20] (20,30] (30,40]
## Levels: [10,20] (20,30] (30,40]
```

- Cerrar los intervalos por la izquierda, no la derecha

```
cut(edad, breaks = c(10, 20, 30, 40), right = FALSE)
## [1] [10,20) [10,20) [20,30) [20,30) [30,40)
## Levels: [10,20) [20,30) [30,40)
```

- Cambiar nombre de los niveles

```
cut(edad, breaks = c(10, 20, 30, 40), labels = c("10-20", "20-30", "30-40"))  
## [1] <NA> 10-20 10-20 20-30 30-40  
## Levels: 10-20 20-30 30-40
```

Creación factor de ejemplo

- Creamos un factor de ejemplo con los tipos de coche de `mpg`

```
f <- factor(mpg$class)
length(f)
## [1] 234
```

- Niveles:

```
levels(f)
## [1] "2seater"      "compact"      "midsize"      "minivan"      "pickup"
## [6] "subcompact"   "suv"
```

- Frecuencia de cada uno de los niveles

```
fct_count(f) %>% arrange(desc(n))
## # A tibble: 7 x 2
##   f           n
##   <fct>     <int>
## 1 suv         62
## 2 compact    47
## 3 midsize    41
## 4 subcompact 35
## 5 pickup     33
## 6 minivan    11
## 7 2seater     5
```


Reordenar niveles de un factor

- `fct_relevel()` , para reordenar los niveles a mano (se mueven al principio)

```
f1 <- fct_relevel(f, "minivan", "compact")
levels(f1)
## [1] "minivan"      "compact"      "2seater"      "midsize"      "pickup"
## [6] "subcompact"   "suv"
```

- `fct_infreq()` , para reordenar los niveles de acuerdo a su frecuencia

```
f2 <- fct_infreq(f)
levels(f2)
## [1] "suv"          "compact"      "midsize"      "subcompact"   "pickup"
## [6] "minivan"      "2seater"
```

Reordenar de acuerdo a otra variable

- `fct_reorder()` , para reordenar los niveles de acuerdo a otra variable

```
f3 <- fct_reorder(f, .x = mpg$cty, .fun = mean)
levels(f3)
## [1] "pickup"      "suv"          "2seater"      "minivan"      "midsize"
## [6] "compact"     "subcompact"
```

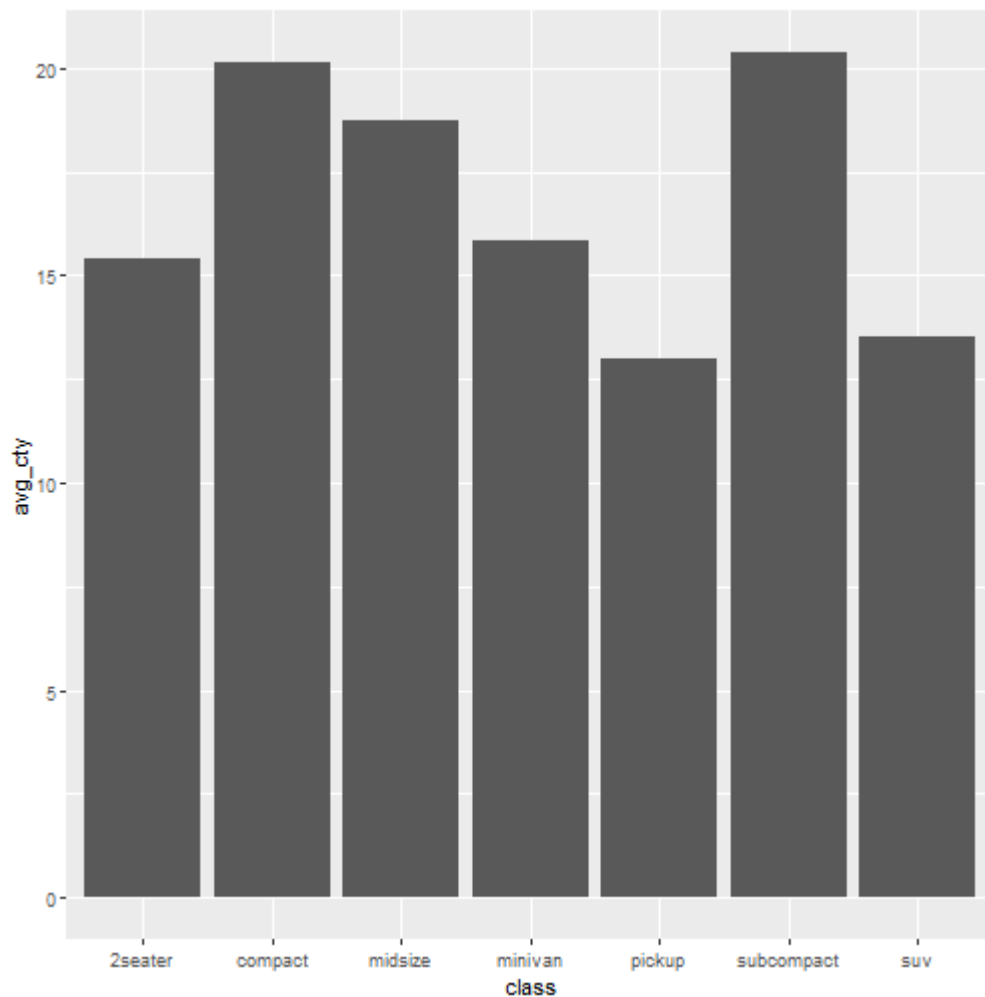
- El orden de los niveles es creciente en `.fun(.x)`

```
consumo_medio <-
  mpg %>%
  group_by(class) %>%
  summarize(avg_cty = mean(cty))

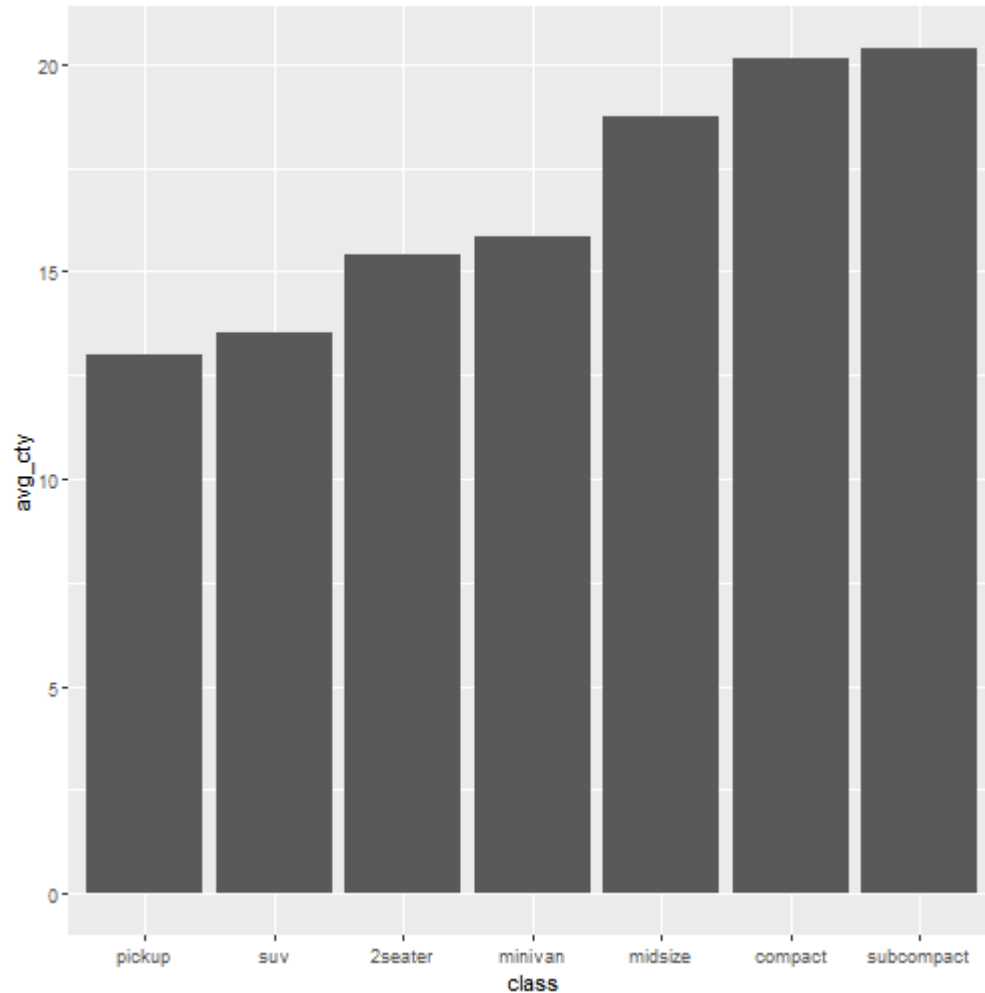
arrange(consumo_medio, avg_cty)
## # A tibble: 7 x 2
##   class      avg_cty
##   <chr>      <dbl>
## 1 pickup      13
## 2 suv        13.5
## 3 2seater    15.4
## 4 minivan    15.8
## 5 midsize    18.8
## 6 compact    20.1
## 7 subcompact 20.4
```

La función `fct_reorder` es útil para ordenar los ejes en gráficos

```
ggplot(consumo_medio, aes(x = class, y = avg_pty)) + geom_col()
```



```
ggplot(consumo_medio, aes(x = fct_reorder(class, avg_pty), y = avg_pty)) +  
  geom_col() +  
  xlab("class")
```



Renombrar niveles

- `fct_recode()` cambia el nombre de los niveles o los elimina (asignando el valor `NULL`)

```
f4 <- fct_recode(f, todoterreno = "suv", `2plazas` = "2seater",  
                NULL = "pickup")  
levels(f4)  
## [1] "2plazas"      "compact"      "midsize"      "minivan"  
## [5] "subcompact"   "todoterreno"
```

- `fct_recode()` también puede colapsar niveles asignando el mismo nombre

```
f5 <- fct_recode(f, `4seater` = "compact", `4seater` = "subcompact",  
                `4seater` = "midsize")  
levels(f5)  
## [1] "2seater" "4seater" "minivan" "pickup"  "suv"
```

Otras funciones

- Otra alternativa es `fct_collapse`

```
f6 <- fct_collapse(f, `4seater` = c("compact", "subcompact", "midsize"),  
                     `2seater` = c("2seater", "pickup"))  
levels(f6)  
## [1] "2seater" "4seater" "minivan" "suv"
```

- O `fct_lump`, que colapsa los niveles menos frecuentes

```
fct_count(f) %>% arrange(desc(n))  
## # A tibble: 7 x 2  
##   f           n  
##   <fct>     <int>  
## 1 suv       62  
## 2 compact   47  
## 3 midsize   41  
## 4 subcompact 35  
## 5 pickup    33  
## 6 minivan   11  
## 7 2seater    5
```

```
f7 <- fct_lump(f, n = 3)  
levels(f7)  
## [1] "compact" "midsize" "suv"      "other"
```