Ejercicios Factores

David Criado Ramón

9/11/2019

Introducción a R (3)

1. Factors

Dado x = c(1, 2, 3, 3, 5, 3, 2, 4, NA), ¿cuáles son los levels de factor(x)?

```
x <- c(1, 2, 3, 3, 5, 3, 2, 4, NA)
factor(x)
```

```
## [1] 1    2    3    3    5    3    2    4    <NA>
## Levels: 1 2 3 4 5
```

Los niveles del factor son los valores únicos para el mismo, ignorando los "missing values" (NA), por lo que son 1, 2, 3, 4 y 5.

Dado x = c(11, 22, 47, 47, 11, 47, 11) y la ejecución de la sentencia factor(x, levels=c(11, 22, 47), ordered=TRUE), ¿cuál es el cuarto elemento de salida?

```
x <- c(11, 22, 47, 47, 11, 47, 11)
factor(x, levels=c(11,22,47), ordered=TRUE)
```

```
## [1] 11 22 47 47 11 47 11
## Levels: 11 < 22 < 47
```

El cuarto elemento de salida es el factor asociado al valor que ocupa la cuarta posición en el vector x, por lo que el cuarto elemento de salida es el factor 47.

Para el factor z <- c("p", "a", "g", "t", "b"), reemplazar el tercer elemento de z por "b"

```
z <- c("p", "a", "g", "t", "b")
z[3] <- "b"
z
```

```
## [1] "p" "a" "b" "t" "b"
```

Z es un vector de caracteres por lo que hemos de utilizar z[3] para poder modificar la categoría asociada a esa posición.

Si z fuese un factor se haría lo mismo, aunque en los diferentes valores para el factor seguiría quedando "g" como un valor posible.

```
z <- factor(c("p", "a", "g", "t", "b"))
z[3] <- "b"
z
```

```
## [1] p a b t b
## Levels: a b g p t
```

Dado z <- factor(c("p", "q", "p", "r", "q")), escribe una expresión de R, que cambie el level "p" a "w"

```
z <- factor(c("p", "q", "p", "r", "q"))
levels(z) <- c("w", "q", "r")
z
```

```
## [1] w q w r q
## Levels: w q r
```

Usa el dataset "iris":

- Escribe la expresión necesaria para convertir la variable "Sepal.Length" en un valor con cinco niveles (levels). Pista: mira la función table() y la función cut().

```
x <- iris
x$Sepal.Length <- cut(x$Sepal.Length, 5)
head(x$Sepal.Length)
```

```
## [1] (5.02,5.74] (4.3,5.02]  (4.3,5.02]  (4.3,5.02]  (4.3,5.02]  (5.02,5.74]
## Levels: (4.3,5.02] (5.02,5.74] (5.74,6.46] (6.46,7.18] (7.18,7.9]
```

- Escribe la expresión necesaria para generar una tabla de frecuencias con dos filas y tres columnas. Las filas deben referirse a si la variable "Sepal.Length" es menor que 5 y las columnas a las diferentes especies.

```
table(iris$Sepal.Length < 5, iris$Species)
```

```
##
##         setosa versicolor virginica
##   FALSE     30         49        49
##   TRUE      20          1         1
```

El factor responses se define como responses <- factor(c("Agree", "Agree", "Strongly Agree", "Disagree", "Agree")), sin embargo nos damos cuenta que tiene un nuevo nivel, "Strongly Disagree". Añade el nuevo nivel al factor y conviértelo en un factor ordenado. Debe quedar de la siguiente manera:

Levels: Strongly Agree < Agree < Disagree < Strongly Disagree

```
responses <- factor(c("Agree", "Agree", "Strongly Agree", "Disagree", "Agree"))
responses <- factor(responses, levels = c("Strongly Agree", "Agree", "Disagree",
                                          "Strongly Disagree"), order=T)
responses
```

```
## [1] Agree          Agree          Strongly Agree Disagree
## [5] Agree
## Levels: Strongly Agree < Agree < Disagree < Strongly Disagree
```

Dado el factor x <- factor(c("high", "low", "medium", "high", "high", "low", "medium")) escribe expresión en R que permita dar valores numéricos únicos para los distintos niveles (levels) de x según el siguiente esquema: (Pista: investiga la función unique() y los parámetros de data.frame())

level high => value 1 level low => value 2 level medium => value 3

```
x <- factor(c("high", "low", "medium", "high", "high", "low", "medium"))
df <- data.frame(x, value=as.numeric(x))
df
```

```
##        x value
## 1   high     1
## 2    low     2
## 3 medium     3
## 4   high     1
## 5   high     1
## 6    low     2
## 7 medium     3
```

Para que cada categoría del factor tenga asociada un valor numérico simplemente hemos de crear un dataframe e indicar el parámetro value como una versión numérica de x.