

# Tarea 1 | STAT NT

Luis Gagnevin 5.153.261-6

4/29/2021

## 1 Ejercicio 1

### 1.1 Parte 1: Vectores

1.1.1 Dado los siguientes vectores, indica a que tipo coercionan

```
w <- c(29, 1L, FALSE, "HOLA")
x <- c("Celeste pelela!", 33, NA)
y <- c(seq(3:25), 10L)
z <- paste(seq(3:25), 10L)
```

Las Coerciones ocurren del mas rigido al mas flexible (Logic -> Integer -> Numeric -> Character) Y en caso de combinaciones toma al mas flexible como su tipo, teniendo esto en cuenta:

#### W es un vector del tipo Character

W Tiene Elementos clasificados como: Logic, Integer, Numeric y Character, por lo que tomara el mas flexible, osea Character.

#### X es un vector del tipo Character

X Tiene Elementos clasificados como: Logic, Numeric y Character, por lo que tomara el mas flexible, osea Character.

#### Y es un vector de ltipo Integer

Y tiene Elementos clasificados solo como Integer

#### Z es un vector del tipo character

Aunque Z sea practicamente igual a y dentro de los (), el uso de paste concatena los vectores luego de convertirlos en texto, por lo que genera un vector con solo elementos del tipo Character

1.1.2 ¿Cual es la diferencia entre c(4,3,2,1) y 4:1?

Si bien su resultado se ve igual, la clasificacion de estos es distinta. Ya que 4:1 es tomado como una cadena entera, mientras que c(4,3,2,1) se toma como un vector numerico.

## 1.2 Parte 2: factor

Dado el siguiente factor x:

```
x <-  
factor(  
c(  
"alto",  
"bajo",  
"medio",  
"alto",  
"muy alto",  
"bajo",  
"medio",  
"alto",  
"ALTO",  
"MEDIO",  
"BAJO",  
"MUY ALTO",  
"QUE LOCO",  
"QUE LOCO",  
"QUE LOCO",  
"A",  
"B",  
"C",  
"GUAU",  
"GOL",  
"MUY BAJO",  
"MUY BAJO",  
"MUY ALTO"  
)  
)
```

1.2.1 Genera un nuevo factor (llamalo xx) transformando el objeto x previamente generado de forma que quede como sigue:

[1] A B M A A B M A A M B A B B A Levels: B < M < A

Para reordenar y que quede de la misma forma, tomare los niveles que me sirven dejando como NA las palabras que no contienen ni “Alto, Medio o Bajo”.

Luego cambiare cada nivel por “alto”, “bajo” y “medio”

Luego elimino los NA y cambio los niveles por “A, B y M” dejandolo casi listo

Y para finalizar ordeno los niveles donde B sea el menor y A el mayor

```
xx <- factor(x, levels= c("alto", "ALTO", "bajo", "BAJO", "medio",  
                          "MEDIO", "muy alto", "MUY ALTO", "MUY BAJO"))  
  
levels(xx)[levels(xx)=="ALTO"]<- "alto"  
levels(xx)[levels(xx)=="MUY ALTO"]<- "alto"  
levels(xx)[levels(xx)=="muy alto"]<- "alto"  
levels(xx)[levels(xx)=="BAJO"]<- "bajo"  
levels(xx)[levels(xx)=="MUY BAJO"]<- "bajo"  
levels(xx)[levels(xx)=="muy bajo"]<- "bajo"
```

```

levels(xx)[levels(xx)=="MEDIO"]<- "medio"

xx<-xx[!is.na(xx)]
levels(xx)[levels(xx)=="alto"]<- "A"
levels(xx)[levels(xx)=="bajo"]<- "B"
levels(xx)[levels(xx)=="medio"]<- "M"

xx<- ordered(xx, levels=c("B","M","A"))
print(xx)

```

```

## [1] A B M A A B M A A M B A B B A
## Levels: B < M < A

```

1.2.2. Generará el siguiente data.frame() Para ello usó el vector xx que obtuviste en la parte anterior.

```

x2<-unique(x)
df<- data.frame(x2[1:15])
names(df)[1]<- "levels"

df2<-data.frame(levels(x))
df2$value<- c(1:15)
names(df2)[1]<- "levels"
df2[order(match(df2$levels, df$levels)),]

```

```

##      levels value
## 2      alto     2
## 5      bajo     5
## 10     medio    10
## 12 muy alto    12
## 3       ALTO     3
## 11     MEDIO    11
## 6       BAJO     6
## 13 MUY ALTO    13
## 15 QUE LOCO    15
## 1         A      1
## 4         B      4
## 7         C      7
## 9      GUAU     9
## 8       GOL     8
## 14 MUY BAJO    14

```