

Tarea 1

Juan Cantero Jimenez

10/22/2021

1. Crear el vector `exp(1):1` y convertirlo a character, luego convertirlo en enteros. Construir otra secuencia `1:exp(1)` y convertirlo a character, luego convertirlo en enteros. Compara las dos secuencias finales (por ejemplo restando ambos vectores).

```
v1 <- exp(1):1
print(v1)
```

```
## [1] 2.718282 1.718282
```

```
v1c <- as.character(v1)
print(v1c)
```

```
## [1] "2.71828182845905" "1.71828182845905"
```

```
v1i <- as.integer(v1c)
print(v1i)
```

```
## [1] 2 1
```

```
v1 == v1i
```

```
## [1] FALSE FALSE
```

```
v1-v1i
```

```
## [1] 0.7182818 0.7182818
```

```
v2 <- 1:exp(1)
print(v2)
```

```
## [1] 1 2
```

```
v2c <- as.character(v2)
print(v2c)
```

```
## [1] "1" "2"
```

```
v2i <- as.integer(v2c)
print(v2i)
```

```
## [1] 1 2
```

```
v2 == v2i
```

```
## [1] TRUE TRUE
```

```
v2-v2i
```

```
## [1] 0 0
```

```
abs(v2-v1)
```

```
## [1] 1.7182818 0.2817182
```

```
v2 == v1
```

```
## [1] FALSE FALSE
```

2. En algunas ocasiones queremos discretizar una variable continua en categorías, para ello utilizaremos la función `cut()`. Por otra parte, para simular de una distribución t-Student veremos en el tema 3 que se utiliza la función `rt()`. Simular 100 valores de una t-Student con 10 grados de libertad y dividir los valores en 5 categorías. Utilizar `table()` para ver que hay cinco categorías.

```
set.seed(1)
```

```
sim <- function(){  
  v <- rt(100, 10)  
  return(table(cut(v,5)))  
}  
sim()
```

```
##  
## (-1.62,-0.785] (-0.785,0.0463] (0.0463,0.878] (0.878,1.71] (1.71,2.55]  
##          19          37          27          10          7
```

3. Utilizar las funciones `rep()` y `seq()` para producir un vector que contenga: `begin{enumerate}` item 1000 valores que indiquen paciente 'SANO' y paciente 'Enfermo', en concreto, los primeros 500 sanos y luego 500 enfermos. item 100 valores consecutivos entre 1 y 20 dos veces seguidas, esto es, 1, 2, . . ., 20, 1, 2, . . ., 20. item los valores: 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6. item la secuencia: 1, 2, 3, 4, 4, 3, 2, 1, 1, 1, 1, 1, 1. `end{enumerate}`

```
#### a  
bin <- rep(c("SANO","Enfermo"), c(500, 500))  
length(bin)
```

```
## [1] 1000
```

```
table(bin)
```

```
## bin  
## Enfermo    SANO  
##      500     500
```

```
#### b  
ji <- rep(seq(1,20,length.out=100) , 2)  
ji
```

```
## [1] 1.000000 1.191919 1.383838 1.575758 1.767677 1.959596 2.151515  
## [8] 2.343434 2.535354 2.727273 2.919192 3.111111 3.303030 3.494949  
## [15] 3.686869 3.878788 4.070707 4.262626 4.454545 4.646465 4.838384  
## [22] 5.030303 5.222222 5.414141 5.606061 5.797980 5.989899 6.181818  
## [29] 6.373737 6.565657 6.757576 6.949495 7.141414 7.333333 7.525253  
## [36] 7.717172 7.909091 8.101010 8.292929 8.484848 8.676768 8.868687  
## [43] 9.060606 9.252525 9.444444 9.636364 9.828283 10.020202 10.212121  
## [50] 10.404040 10.595960 10.787879 10.979798 11.171717 11.363636 11.555556  
## [57] 11.747475 11.939394 12.131313 12.323232 12.515152 12.707071 12.898990  
## [64] 13.090909 13.282828 13.474747 13.666667 13.858586 14.050505 14.242424  
## [71] 14.434343 14.626263 14.818182 15.010101 15.202020 15.393939 15.585859  
## [78] 15.777778 15.969697 16.161616 16.353535 16.545455 16.737374 16.929293  
## [85] 17.121212 17.313131 17.505051 17.696970 17.888889 18.080808 18.272727  
## [92] 18.464646 18.656566 18.848485 19.040404 19.232323 19.424242 19.616162
```

```
## [99] 19.808081 20.000000 1.000000 1.191919 1.383838 1.575758 1.767677
## [106] 1.959596 2.151515 2.343434 2.535354 2.727273 2.919192 3.111111
## [113] 3.303030 3.494949 3.686869 3.878788 4.070707 4.262626 4.454545
## [120] 4.646465 4.838384 5.030303 5.222222 5.414141 5.606061 5.797980
## [127] 5.989899 6.181818 6.373737 6.565657 6.757576 6.949495 7.141414
## [134] 7.333333 7.525253 7.717172 7.909091 8.101010 8.292929 8.484848
## [141] 8.676768 8.868687 9.060606 9.252525 9.444444 9.636364 9.828283
## [148] 10.020202 10.212121 10.404040 10.595960 10.787879 10.979798 11.171717
## [155] 11.363636 11.555556 11.747475 11.939394 12.131313 12.323232 12.515152
## [162] 12.707071 12.898990 13.090909 13.282828 13.474747 13.666667 13.858586
## [169] 14.050505 14.242424 14.434343 14.626263 14.818182 15.010101 15.202020
## [176] 15.393939 15.585859 15.777778 15.969697 16.161616 16.353535 16.545455
## [183] 16.737374 16.929293 17.121212 17.313131 17.505051 17.696970 17.888889
## [190] 18.080808 18.272727 18.464646 18.656566 18.848485 19.040404 19.232323
## [197] 19.424242 19.616162 19.808081 20.000000
```

```
#### c
```

```
ja <- rep(1:6, 1:6)
ja
```

```
## [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6
```

```
table(ja)
```

```
## ja
## 1 2 3 4 5 6
## 1 2 3 4 5 6
```

```
#### d
```

```
ju <- c(seq(1, 4), seq(4, 1) , rep(1, 5))
ju
```

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

4. Búsqueda de un banco de datos en la web del INE que contenga una variable estudiada por año, por provincia (comunidad autónoma o municipio) y por alguna otra variable (género, sector industrial...). Una vez descargado se debe leer desde R (con código) y convertirlo en un data.frame con 4 variables, la estudiada, el año, la provincia (o similar) y la otra variable seleccionada. Da el formato correcto (numérico, factor, etc.) a cada columna del banco de datos (si no lo tiene).

```
data = read.csv("20173.csv", sep = ";")
colnames(data)
```

```
## [1] "Provincias" "Tipo.de.disolución.matrimonial"
## [3] "Periodo" "Total"
```

```
ncol(data)
```

```
## [1] 4
```

```
nrow(data)
```

```
## [1] 1696
```

```
## clase y tipo de los datos
```

```
for (x in colnames(data)){
  cat("La variable",x,"es de clase",class(data[[x]]),"\n")
  cat("La variable",x,"es de tipo",typeof(data[[x]]),"\n")
}
```

```

## La variable Provincias es de clase character
## La variable Provincias es de tipo character
## La variable Tipo.de.disolución.matrimonial es de clase character
## La variable Tipo.de.disolución.matrimonial es de tipo character
## La variable Periodo es de clase integer
## La variable Periodo es de tipo integer
## La variable Total es de clase character
## La variable Total es de tipo character

#Se ha considerado oportuno retirar la numeración que aparece antes de cada provincia
#Para esto se hace uso de la función gsub
data$Provincias <- as.factor(gsub("\\d{1}", "", data$Provincias))
data$Tipo.de.disolución.matrimonial <- as.factor(data$Tipo.de.disolución.matrimonial)
#Es necesario quitar el punto que en español indica unidades de millar pero que en inglés
#indica los decimales, esto se realiza con la función gsub

data$Total <- as.numeric(gsub(".", "", data$Total))
for (x in colnames(data)){
  cat("La variable", x, "es de clase", class(data[[x]]), "\n")
  cat("La variable", x, "es de tipo", typeof(data[[x]]), "\n")
}

## La variable Provincias es de clase factor
## La variable Provincias es de tipo integer
## La variable Tipo.de.disolución.matrimonial es de clase factor
## La variable Tipo.de.disolución.matrimonial es de tipo integer
## La variable Periodo es de clase integer
## La variable Periodo es de tipo integer
## La variable Total es de clase numeric
## La variable Total es de tipo double

```