subvectores

MARK

2024-05-23

# subvector

indices enmpiezan en 1

vector[i] donde i es un vector extrae esas posiciones

vector[-i] todas menos la intrada i vector[-x] todas menos las posiciones del vector dado

x=1:20  
  
x[-1]

## [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

x[-c((1:10),12)]

## [1] 11 13 14 15 16 17 18 19 20

x[13:1]

## [1] 13 12 11 10 9 8 7 6 5 4 3 2 1

ir a los ultimos

x[length(x)] penultimo x[length(x)-1] antepenultimo

rescatar datos en orden inverso x[b:a] si v>a entonces va en sentido inverso

de hecho se le pueden dar en cualquier orden

x[c(5,7,2,3)]

## [1] 5 7 2 3

#filtros

elemtnos que buscan ciertas condiciones logicas

recordar que != :significa distintos de | : significa o >= <=

x[x>10]

## [1] 11 12 13 14 15 16 17 18 19 20

# filtrador de posiciones

which(“x cumpe condicion”) inidices de TODAS las entradas which.min() la primera que cumple el minimo indice which.max() la primera que cumple el maximo indice which(x==min(x)) todas las que son iguales al minimo indices which(x==max(x)) todas las que son iguales al minimo indices

y= c(1:10,1,1,1,10,10)  
which.min(y)

## [1] 1

which.max(y)

## [1] 10

which(y==min(y))

## [1] 1 11 12 13

which(y==max(y))

## [1] 10 14 15

# como funciona, obteniendo vector false true, parece a un mapeo

x%%2==0

## [1] FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE  
## [13] FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE

#si aplicamos x[x%%2==0] se queda con los TRUE

# empleando dos vectores para filtar logicamente

e1=1:10  
e1

## [1] 1 2 3 4 5 6 7 8 9 10

e2=seq(2,20,by=2)  
e2

## [1] 2 4 6 8 10 12 14 16 18 20

e1[e2%%2==1]

## integer(0)

e1[e2%%e1==0]

## [1] 1 2 3 4 5 6 7 8 9 10