

Estructura de datos

Pablo

2023-12-20

Básico

-`c()`: para definir un vector

-`scan()`: para definir un vector escaneandolo.

-`fix(x)`: para modificar visualmente el vector x

-`rep(a,n)`: para definir un vector que consiste en el dato a repetido n veces.

```
c(1, 2, 3)
```

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

```
rep("Lunes", 6)
```

```
## [1] "Lunes" "Lunes" "Lunes" "Lunes" "Lunes" "Lunes"
```

Una **progresión aritmética** es una sucesión de números tales que la diferencia d , de cualquier par de términos sucesivos de la secuencia es constante.

$$a_n = a_1 + (n - 1) \cdot d$$

-`seq(a,b,by=d)`: para generar una progresión aritmética de diferencia d que empieza en a hasta llegar a b .

-`seq(a,b, length.out=n)`: define progresión aritmética de longitud n que va de a a b con diferencia d . Por tanto $d = (b - a)/(n - 1)$.

-`seq(a,by=d, length.out=n)`: define la progresión aritmética de longitud n y diferencia d que empieza en a .

-`a:b`: define la secuencia de números enteros consecutivos entre dos números a y b .

Funciones

Cuando queremos aplicar una función a cada uno de los elementos de un vector de datos la función *sapply* nos ahorra tener que programar con bucles en *R*.

- `sapply(nombre_de_vector, FUN=nombre_de_funcion)`: para aplicar dicha función a todos los elementos de un vector.

`-sqrt(x)`: calcula un nuevo vector con las raíces cuadradas de cada uno de los elementos del vector x .

Dado un vector de datos x podemos calcular muchas medidas estadísticas del mismo:

`-length(x)`: calcula la longitud del vector x .

`-max(x)`: calcula el máximo del vector x .

`-min(x)`: calcula el mínimo del vector x .

`-sum(x)`: calcula la suma de las entradas del vector x .

`-prod(x)`: calcula el producto de las entradas del vector x .

`-mean(x)`: calcula la media aritmética de las entradas del vector x .

`-diff(x)`: calcula el vector formado por las diferencias sucesivas entre las entradas del vector original x .

`-cumsum(x)`: calcula el vector formado por las sumas acumuladas entre las entradas del vector original x :

- Permite definir sucesiones descritas mediante sumatorios.

- Cada entrada `cumsum(x)` es la suma de las entradas de x hasta su posición.

`-sort(x)`: ordena el vector en orden natural de los objetos que lo forman: el orden numérico creciente, orden alfabético... `-rev(x)`: invierte el orden de los elementos del vector x .

Ejercicio: Binomio de Newton

Producto notable

La fórmula del producto notable es:

$$(a + b)^2 = a^2 + 2ab + b^2$$

Función en R:

```
binomioNewton2 = function(a,b){  
  a^2+2*a*b+b^2  
}  
binomioNewton2(1,2)
```

```
## [1] 9
```

```
binomioNewton2(2,1)
```

```
## [1] 9
```

Binomio de Newton

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} \cdot a^{n-k} \cdot b^k$$

Funcion en R:

```
binomioNewton=function(a,b,n){  
  cumsum(choose(n,(0:n))*a^{n-(0:n)}*b^{(0:n)})[n+1]  
}  
binomioNewton(2,1,2)
```

```
## [1] 9
```