Tema 5 - Vectores con R, Python y Octave

Ramon Ceballos

2/3/2021

Vectores con R

1. Definir vectores en R

Para definir vectores con R, simplemente utilizamos la función **c()**.

```
x = c(1,2,3,4,5,6,7,8,9,0)
x
```

```
##  [1] 1 2 3 4 5 6 7 8 9 0
```

Para saber la longitud de un vector (la dimensión), utilizamos la función **length()**.

```
length(x)
```

```
## [1] 10
```

2. Operaciones básicas con vectores en R

Sumar y restar dos o más vectores es muy simple, siempre y cuando sean de la misma dimensión:

```
x = c(0,3,-1,3,5)
y = c(1,2,3,-1,0)
x+y
```

```
## [1] 1 5 2 2 5
```

```
x-y
```

```
## [1] -1  1 -4  4  5
```

El **producto por un escalar** también funciona de forma sencilla:

```r
x = c(1,0,-1,0,2,0,-2)
2*x
```

```
## [1]  2  0 -2  0  4  0 -4
```

```r
-2*x
```

```
## [1] -2  0  2  0 -4  0  4
```

```r
5*x
```

```
## [1]   5   0  -5   0  10   0 -10
```

3. Producto escalar en R

El **producto escalar** no está definido en R, pero es sencillo crear una función que nos lo calcule.

```r
productoEscalar = function(x,y){
  if (length(x) == length(y)){
    sum(x*y)
  }else{
    print("ERROR: No se puede calcular el producto escalar de estos dos vectores porque
          no son de la misma dimensión")
  }
}
```

Simplemente, lo que hace la función anterior es calcular el producto escalar de dos vectores, siempre y cuando estos tengan la misma dimensión.

Si no, salta un mensaje explicando el error cometido.

```r
x = c(0,3,-1,3,5)
y = c(1,2,3,-1,0)
productoEscalar(x,y)
```

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

4. Norma Euclídea con R

Se requiere del empleo de la librería **pracma**.

```r
library (pracma)
```

Para calcular la **norma euclídea** de un vector, utilizamos la función `Norm()`.

```r
x = c(1,2,0,3,-1,1)
Norm(x)
```

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

5. Calcular Distancia Euclídea entre dos puntos con R

La distancia entre dos puntos x, y, se define como la norma del vector $\vec{xy}$, es decir $\mathrm{d}(x, y) = ||x - y||$. Para calcularla, definimos la función **distancia()** del siguiente modo:

```
distancia = function(x,y){
  if (length(x) == length(y)){
    Norm(x-y)
  }else{
    print("ERROR: No se puede calcular la distancia entre estos dos puntos porque
          no son de la misma dimensión")
  }
}
```

Simplemente, lo que hace la función anterior es calcular la distancia Euclídea entre dos puntos, siempre y cuando estos tengan la misma dimensión.

Si no, salta un mensaje explicando el error cometido.

```
x = c(0,3,-1,3,5)
y = c(1,2,3,-1,0)
distancia(x,y)
```

```
## [1] 7.681146
```

6. Ángulo entre dos vectores en R

Si queremos calcular el ángulo entre dos vectores, debemos definir nosotros mismos la función, atendiendo al **teorema del coseno**.

```
angleRad = function(x,y){
  if (length(x) == length(y)){
    acos(productoEscalar(x,y)/(Norm(x)*Norm(y)))
  }else{
    print("ERROR: No se puede calcular el ángulo entre estos dos vectores porque
          no son de la misma dimensión")
  }
}
angleRad(x,y)
```

```
## [1] 1.570796
```

Nuestra función nos devuelve el resultado en Radianes. En caso de querer el resultado en grados, lo que podemos hacer es una pequeña conversión:

```
angleGrad = function(x,y){
  if (length(x) == length(y)){
    acos(productoEscalar(x,y)/(Norm(x)*Norm(y)))*360/(2*pi)
  }else{
    print("ERROR: No se puede calcular el ángulo entre estos dos vectores porque
```

```
          no son de la misma dimensión")
  }
}
angleGrad(x,y)
```

```
## [1] 90
```

7. Calcular la proyección ortogonal de $\vec{v}$ sobre $\vec{u}$ con R

También definimos una función que calcule la proyección ortoganol de un vector **v** sobre el vector **u**.

$$P_{\vec{v}}(\vec{u}) = \vec{v}_1 = \lambda\vec{u} = \frac{\langle\vec{u},\vec{v}\rangle}{||\vec{u}||^2}\vec{u}$$

```
proyeccionOrt = function(u,v){
  if (length(u) == length(v)){
    productoEscalar(u,v)/Norm(u)^2*u
  }else{
    print("ERROR: No se puede calcular la proyección ortogonal de v sobre
          u porque no son de la misma dimensión")
  }
}
u = c(3,1)
v = c(1,2)
proyeccionOrt(u,v)
```

```
## [1] 1.5 0.5
```

8. Producto vectorial con R

Desarrollamos la función del producto vectorial como hemos visto en clase.

```
productoVectorial= function(x,y){
  if (length(x) == length(y) & length(x) == 3){
    c(x[2]*y[3] - x[3]*y[2], -(x[1]*y[3]-x[3]*y[1]), x[1]*y[2]-x[2]*y[1])
  }else{
    print("No se cumplen las condiciones necesarias para calcular el
          producto vectorial de estos dos vectores")
  }
}
```

```
x = c(1,2,3)
y = c(0,-1,1)
productoVectorial(x,y)
```

```
## [1]  5 -1 -1
```

9. Producto mixto con R

Desarrollamos la función del producto mixto como hemos visto en clase.

```
productoMixto = function(x,y,z){
  if (length(x) == length(y) & length(x) == length(z) & length(x) == 3){
    det(rbind(x,y,z))
  }else{
    print("No se cumplen las condiciones necesarias para calcular el
          producto mixto de estos tres vectores")
  }
}
```

```
x = c(1,2,3)
y = c(0,-1,1)
z = c(2,0,-3)
productoMixto(x,y,z)
```

```
## [1] 13
```