

Funciones creadas para hallar la MODA en algún vector

Debido a que no existen muchos paquetes que ofrezcan entre sus funciones la moda, decidí crear una función. Sin embargo, en el transcurso de programar y probar mi función, noté ciertas circunstancias por las que podría modificar e implementar más la función inicial.

Para probar y notar las diferencias de cada función creada para la moda, utilizaremos los siguientes vectores:

Vector:	Descripción:
<code>z<-c(9,8,7,6,5,4,3,2,5,8,64,8,8,8,55,5,5)</code>	Vector numérico y unimodal. La moda es 8
<code>w<-c(12,12,11,11,7,8,9,6,3)</code>	Vector numérico y bimodal. Moda: 11 y 12
<code>b<-c("azul","azul","azul","rojo","amarillo","amarillo","amarillo","verde")</code>	Vector de caracteres. Bimodal. Moda: Azul y Amarillo.
<code>d<-c(7,8,9,6,5,4,1,2,3)</code>	Vector numérico. No tiene Moda.

Función Simple para la Moda:	Prueba con vector z
<pre>modaprim<-function(x){ q<-table(x) a<-which.max(q) print(a) }</pre>	<pre>> z<-c(9,8,7,6,5,4,3,2,5,8,64,8,8,8,55,5,5) > table(z) z 2 3 4 5 6 7 8 9 55 64 1 1 1 4 1 1 5 1 1 1 > modaprim(z) 8 7</pre>
Prueba con vector w	Prueba con vector b
<pre>> modaprim(w) 11 6</pre>	<pre>> modaprim(b) amarillo 1</pre>

‘modaprim’ hace uso de la función ‘table’, la cual nos da las frecuencias de cada cantidad en el vector; ‘which.max’ es utilizada sobre la salida de ‘table’ para dar a conocer la cantidad con mayor frecuencia (¡moda!)

Resultados con el vector z: aparecen 2 valores. El primero es la moda, el segundo se refiere a la posición de la moda en la salida de ‘table’. Así notamos que la moda es 8, valor que se encuentra 7mo en el vector de ‘table’.

Resultados con el vector w: Notamos que sólo da como moda el valor de 11, pese a que 12 también tiene la misma frecuencia. Esto nos indicaría que ‘modaprim’ es para vectores unimodales solamente. Observar también que toma el valor menor si es que hay más de un valor para la moda.

Resultados con el vector b: La función permite la moda para caracteres (debido a la función 'table'). Aquí también se aprecia que solo toma un elemento como moda ('amarillo'). Pese a que 'azul' aparece primero en el vector y tiene la misma frecuencia que 'amarillo', la función toma este último por orden alfabético.

¿Cómo solucionamos el hecho de que la función no tome más de un valor para la moda? Y, ¿qué pasa si el vector a analizar en realidad no tiene moda?

Función:	Resultados:
<pre>moda<-function(x) { m1=sort(table(x),decreasing=T) moda=names(m1[m1==m1[1]]) moda=as.numeric(moda) return(moda) }</pre>	<pre>> moda(z) [1] 8 > moda(w) [1] 11 12 > moda(b) [1] NA NA Warning message: In moda(b) : NAs introduced by coercion > moda(d) [1] 1 2 3 4 5 6 7 8 9</pre>
<pre>moda1<- function(x) {w<-sort(unique(x)) z<-table(x) if(max(z)==1)print("No existe moda") else{print(w[which.max(z)])}}</pre>	<pre>> moda1(z) [1] 8 > moda1(w) [1] 11 > moda1(b) [1] "amarillo" > moda1(d) [1] "No existe moda"</pre>
<pre>moda2<-function(x) { m1=sort(table(x),decreasing=T) moda=names(m1[m1==m1[1]]) moda=as.character(moda) return(moda) }</pre>	<pre>> moda2(z) [1] "8" > moda2(w) [1] "11" "12" > moda2(b) [1] "amarillo" "azul" > moda2(d) [1] "1" "2" "3" "4" "5" "6" "7" "8" "9"</pre>
<pre>moda3<- function(x) { z<- table(x) if(max(z)==1)print("Todos los datos tienen la misma frecuencia, No existe moda") return(as.numeric(names(z)[z == max(z)])) }</pre>	<pre>> moda3(z) [1] 8 > moda3(w) [1] 11 12 > moda3(b) Error in round(z, digits = n) : non-numeric argument to mathematical function > moda3(d) [1] "Todos los datos tienen la misma frecuencia, No existe moda"</pre>

	[1] 1 2 3 4 5 6 7 8 9
moda4<- function(x) { z<-table(x) if(max(z)==1)print("Todos los datos tienen la misma frecuencia, No existe moda") return(as.character(names(z)[z==max(z)])) }	> moda4(z) [1] "8" > moda4(w) [1] "11" "12" > moda4(b) [1] "amarillo" "azul" > moda4(d) [1] "Todos los datos tienen la misma frecuencia, No existe moda" [1] "1" "2" "3" "4" "5" "6" "7" "8" "9"

La función 'moda' usa la función 'sort' para ordenar las frecuencias de 'table' de manera decreciente (usando el argumento 'decreasing=T') para que la moda sea el primer valor. La función 'as.numeric' impide que pueda tomar vectores de caracteres. Si omitimos la sentencia 'moda=as.numeric(moda)', las salidas nos saldrán en comillas, sean números o caracteres, debido a la función 'names', la cual toma cualquier valor convirtiéndolo en cadena de caracteres. Las salidas de los vectores de prueba nos demuestran que se ha solucionado el problema para cuando haya más de una moda. Sin embargo, no nos indica cuando no existe moda (todos los valores con la misma frecuencia).

La función 'moda1' usa 'unique' para tomar los distintos valores del vector por única vez, evitando repeticiones del mismo valor. Se ordenará de forma ascendente con la función 'sort' para que coincida con el orden de recuentos de frecuencias en 'table', con la intención de que no salgan 2 valores, tal y como ocurre con 'modaprim'. Tratando de solucionar el tema de la no existencia de la moda, colocamos el condicional 'if' indicando que si el valor máximo de frecuencia es 1, entonces no existe moda.

La función 'moda2' es la modificación de la función 'moda', cambiando 'as.numeric' por 'as.character' para poder tomar cadenas de caracteres. Las funciones 'moda3' y 'moda4' son mejoras de 'moda' y 'moda2', respectivamente. Se les ha añadido el condicional 'if' con el mismo propósito que 'moda1'.

FUNCIÓN	Apto para caracteres	Indica que 'No existe moda'	Apto para bimodales
Modaprim	✓	X	X
moda	X	X	✓
moda1	✓	✓	X
moda2	✓	X	✓
moda3	X	✓	✓
moda4	✓	✓	✓

La última función sería la más completa, sin embargo, debemos notar que esta función solo identificará que no existe moda en un vector sólo si la frecuencia es 1. Si cada valor fuese repetido la misma cantidad de veces, 'moda4' tomaría como moda el primer valor.

Podríamos mejorar la función aún más, sea con más funciones, argumentos modificados, condicionales y otros comandos que puedan ser útiles y compatibles entre sí.

Trabajo realizado por: Flores Cáceres Fiorella Alexandra

Correo electrónico: 20101254@lamolina.edu.pe