

# funciones

2024-09-07

Se distinguen dos casos

## intervalos del mismo tamaño

### tabla de frecuencias

```
tabla_frecuencias <- function(x , k , p , i.l = F){  
  
  # amplitud  
  z = diff(range(x))/k  
  y = ceiling (z * 10 ^ p)  
  y = y * 10 ^ (-p)  
  if (y==z){  
    A = z + ( 10 ^ (-p) )  
  } else {A = y}  
  A = y  
  # Puntos de corte  
  
  L1 = min(x) - 1/2 * 10 ^ (-p)  
  L = L1 + A * c(0:k)  
  
  # marcas de clase  
  
  X1 = (L[1]+L[2])/2  
  X = X1 + A * c(0:(k-1))  
  
  # establecemos como ordinal la variable cuantitativa  
  #aqui modificamos, tenemos que restar 1 para ser exactos, debido a que el punto se cuenta como char,  
  
  d = nchar(as.character(L[1]))  
  
  x_cut = cut(x, breaks = L, right = F , dig.lab = d, include.lowest = i.l)  
  
  #obtenemos los intervalos para df  
  intervals = levels(x_cut)  
  
  #tablas de frecuencias  
  Fr_abs = as.vector(table(x_cut))  
  Fr_rel = as.vector(round(table(x_cut)/length(x),5))  
  Fr_acum = cumsum(Fr_abs)  
  Fr_rel_acum = cumsum(Fr_rel)
```

```

    tabla = data.frame(intervalos = intervals, mc = X , Fr.abs = Fr_abs, Fr_acum, Fr_rel, Fr_rel_acum)

    return(tabla)
}

```

## Calculo de estadisticos

```

lista_frecuencias_asist <- function(x , k , p , i.l = F){

  # amplitud
  z = diff(range(x))/k
  y = ceiling (z * 10 ^ p)
  y = y * 10 ^ (-p)
  if (y==z){
    A = z + ( 10 ^ (-p) )
  } else {A = y}

  # Puntos de corte

  L1 = min(x) - 1/2 * 10 ^ (-p)
  L = L1 + A * c(0:k)

  # marcas de clase

  X1 = (L[1]+L[2])/2
  X = X1 + A * c(0:(k-1))

  # establecemos como ordinal la variable cuantitativa
  #aquí modificamos, tenemos que restar 1 para ser exactos, debido a que el punto se cuenta como char,

  d = nchar(as.character(L[1]))

  x_cut = cut(x, breaks = L, right = F , dig.lab = d , include.lowest = i.l)

  #obtenemos los intervalos para df
  intervals = levels(x_cut)

  #tablas de frecuencias
  Fr_abs = as.vector(table(x_cut))
  Fr_rel = as.vector(round(table(x_cut)/length(x),5))
  Fr_acum = cumsum(Fr_abs)
  Fr_rel_acum = cumsum(Fr_rel)

  #aquí conviene mas devolver una lista
  lista = list(intervalos = intervals, mc = X , Fr.abs = Fr_abs, Fr.acum = Fr_acum, Fr.rel = Fr_rel, Fr.rel.acum = Fr_rel_acum)

  return(lista)
}

```

```

notas = sample(0:10, 100, replace = T)
lista_frecuencias_asist(notas, k = 10, p= 1 , i.l = T)

```

```

## $intervalos
## [1] "[-0.05,1.05)" "[1.05,2.15)" "[2.15,3.25)" "[3.25,4.35)" "[4.35,5.45)"
## [6] "[5.45,6.55)" "[6.55,7.65)" "[7.65,8.75)" "[8.75,9.85)" "[9.85,10.95]"
##
## $mc
## [1] 0.5 1.6 2.7 3.8 4.9 6.0 7.1 8.2 9.3 10.4
##
## $Fr.abs
## [1] 14 12 7 8 15 13 13 3 7 8
##
## $Fr.acum
## [1] 14 26 33 41 56 69 82 85 92 100
##
## $Fr.rel
## [1] 0.14 0.12 0.07 0.08 0.15 0.13 0.13 0.03 0.07 0.08
##
## $Fr.rel.acum
## [1] 0.14 0.26 0.33 0.41 0.56 0.69 0.82 0.85 0.92 1.00
##
## $L
## [1] -0.05 1.05 2.15 3.25 4.35 5.45 6.55 7.65 8.75 9.85 10.95

```

*# necesaria para el rename pero no usamos aqui : library(dplyr)*

```

funcion_estadisticos<- function(x , k , p , q , i.d = F){

  lista = lista_frecuencias_asist(x,k,p, i.d)

  # calculo media
  n = length(x)
  media = sum(lista$mc * lista$Fr.abs) /n

  # varianza y dt
  var = sum(lista$mc ^ 2 * lista$Fr.abs) /n - media ^ 2
  dt = sqrt(var)

  # intervalo modal

  Modal = lista$intervalos[which(lista$Fr.abs == max(lista$Fr.abs))]

  # calculamos cuantil

  if(lista$Fr.rel.acum[1] >= q){
    LC = 1
    N_C = 0
  } else{
    LC = which( lista$Fr.rel.acum >= q)[1]
    N_C = lista$Fr.acum[LC - 1]
  }
}

```

```

L_C = lista$L[LC]
A_C = lista$L[LC+1] - lista$L[LC]
n_C = lista$Fr.abs[LC]
cuantil = L_C + A_C * (n * q - N_C) / n_C

# asignando nombre a variables de forma automatica, no funciona pero tenerlo en cuenta
# assign(paste("cuantil", q , sep = ""), mediana)

#lo siguiente funciona parcialmente
# names(cuantil) = paste("cuantil", q * 100 , sep = "_")

#otra forma
#df %>%
#  rename(paste("cuantil", q * 100 , sep = "_") = cuantil) %>%

df = data.frame(Media = media, Varianza = var, Desv.tip = dt, I.modal = Modal, cuantil)

# esta si funciona
name = paste("cuantil", q * 100 , sep = "_")
colnames(df)[grep("cuantil", colnames(df))] <- name

return(df)
}

```

```
funcion_estadisticos(notas, 10, p=0 , q = 0.59, i.d = T)
```

```
## Media Varianza Desv.tip I.modal cuantil_59
## 1 4.96 8.3084 2.88243 [5.5,7.5) 5.730769
```

```
tabla_frecuencias(notas, 10, p = 0)
```

```
## intervalos mc Fr.abs Fr_acum Fr_rel Fr_rel_acum
## 1 [-0.5,0.5) 0 5 5 0.05 0.05
## 2 [0.5,1.5) 1 9 14 0.09 0.14
## 3 [1.5,2.5) 2 12 26 0.12 0.26
## 4 [2.5,3.5) 3 7 33 0.07 0.33
## 5 [3.5,4.5) 4 8 41 0.08 0.41
## 6 [4.5,5.5) 5 15 56 0.15 0.56
## 7 [5.5,6.5) 6 13 69 0.13 0.69
## 8 [6.5,7.5) 7 13 82 0.13 0.82
## 9 [7.5,8.5) 8 3 85 0.03 0.85
## 10 [8.5,9.5) 9 7 92 0.07 0.92
```