

SS().R

```

SS <- function(m=NULL){
  MAD <- function(x){
    2*median(abs(x-median(x))) #Fórmula para calculo do limite
superior para valores considerados naturais
  }
  MAD15 <- function(y){ #Formula de limite inferior para calculo dos
valores considerados naturais.
    q15=quantile(y,c(.15)) #Precisamos calcular o quantil de 15% para
isso
    vars=sort(y) #Aqui ordenei o vetor de forma crescente
    pos=length(which(vars<q15)) #aqui chamamos de pos o numero de
elementos do vetor ordenado onde temos valores menores que o quantil de
15%
    posF=pos #Criei o vetor posF igual ao pos, ele só será substituido
se entrarmos no IF
    if (pos!=0){ #essa linha para inserir o calculo para quando o valor
de pos é maior que 0.
      if (abs(q15-vars[pos]) >= (abs(q15-vars[pos+1]))){ #Outro IF para
um teste lógico para podermos atribuir novo valor posF caso o quantil
15% seja maior ou igual ao vars na posição do pos
        posF=pos+1 #estabelecemos novo valor para o posF
      }
    }
    medianal5=median(sort(y)[1:posF]) #fórmula para o cálculo da
mediana dos valores 15% menores.
    mad15 <- 2*median(sort(y)[1:posF]-mediana15) #Calculo do limite
inferior,
  }
  medianal5 <- function(y){ #calculo da mediana de 15%
    q15=quantile(y,c(.15)) #QUantil de 15%
    vars=sort(y) #vetor ordenado de forma crescente
    pos=length(which(vars<q15)) #aqui chamamos de pos o numero de
elementos do vetor ordenado onde temos valores menores que o quantil de
15%
    posF=pos #Criei o vetor posF igual ao pos, ele só será substituido
se entrarmos no IF
    if (pos!=0){ #inserir o calculo quando pos é maior que 0
      if (abs(q15-vars[pos]) >= (abs(q15-vars[pos+1]))){ #Teste lógico
para podermos atribuir novo valor posF caso o quantil 15% seja maior ou
igual ao vars na posição do pos
        posF=pos+1 #Novo valor para o posF
      }
    }
    median(sort(y)[1:posF]) #calculo dos valores ordenados ate a
posição de posF
  }
  cv <- function(j){ #Formula para o calculo de coeficiente de variação
na.omit(j)
    sd(j)/mean(j)*100 #desvio padrao de cada elemento do vetor dividido

```

```
pela média do vetor
}
if(is.null(m))m <- abs(matrix(runif(540,0,5),90,6)) #RESGATADO DA
AJUDA DO FORUM, CONVERSA DA PATRICIA, DIOGRO E DANILO MUNIZ, para
funcionamento da funcao mesmo que o usuario nao adicione dados
if(class(m)!="matrix"){
  stop("Vacilou, o vetor precisa ser uma matriz")
}
N <- apply(m,2,length) #número de elementos do vetor
Media <- apply(m,2,mean) #média de cada coluna do vetor
Desvio_Padrao <- apply(m,2,sd) #desvio padrao para cada coluna do
vetor
Minimo <- apply(m,2,min) #Valor minimo para cada coluna do vetor
Maximo <- apply(m,2,max) #Valor maximo para cada coluna do vetor
Mediana <- apply(m,2,median) #Mediana para cada variavel
MeMAD <- Mediana + apply(m,2,MAD) # valor superior do limite
considerado natural
Mediana15 <- apply(m,2,median15) #Mediana dos valores menores que
o quantil de 15% para cada variável
Me15MAD15 <- Mediana15 - apply(m,2,MAD15) #Valor do limite inferior
considerado natural para cada variavel
Coeficiente_Variacao <- apply(m,2,cv) #Coeficiente de variacao de
cada variavel
resultados <-
data.frame(N,Minimo,Media,Mediana,Mediana15,Desvio_Padrao,Maximo,MeMAD,
Me15MAD15,Coeficiente_Variacao) #vetor dos resultados
return(round(resultados,2)) #Saída da funcao arredondado para 2
casas.
#return(list(n=n,media=media,
Minimos=Minimos,Maximos=Maximos,Mediana=Mediana,MAD=MAD,MAD15=MAD15,Coe
ficiente_Variacao=Coeficiente_Variacao, Quantil=Quantil))
}
```

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:mcristine:ss

Last update: **2020/08/12 06:04**

