

Código Smart.sample

```
#Funcao ''smart.sample'' para amostragem e sorteio de elementos.

#Doenças infecciosas de importância para saúde pública e animal são
contempladas, geralmente, com sistemas de vigilância epidemiológica que
visam estratégias de prevenção, rápida detecção e combate a focos e
disseminação.
#Para isso, além de componentes de vigilância passiva, como as notificações
mandatórias respaldadas por legislação específica (dependendo do agravo em
questão), são necessárias medidas ativas, ou seja, busca de potenciais casos
de doença numa população suscetível.
#Deste modo, componentes ativos de busca de doenças podem ser realizados a
partir da coleta de espécimes biológicos para diagnóstico, uma vez que o
processo de investigação seja desencadeado.
#Para tal, o processo de amostragem é essencial, dado que a coleta
censitária seria extremamente onerosa e ineficaz.
#Esse processo de amostragem é geralmente realizado em dois níveis: agrgado
(representando fazendas, abatedouros, granjas, por exemplo) ou individual
(animais ou carcaças).

#A ideia desta função (smart.sample) é definir o número necessário de
fazendas e/ou de animais que devem ser amostrados. Além disso, quando
necessário, o usuário poderá de imediato saber quais animais/fazendas devem
ser coletados para avaliar a prevalência aparente de determinado agravo.

#Entrada:

smart.sample = function(data = FALSE, population = FALSE, N, confidence =
c(0.90, 0.95, 0.99), precision, position = FALSE, P )

{
  conf <- c(0.90, 0.95, 0.99) #determinando os valores de confiança que a
funcao suporta
  if(class(confidence) != "numeric" | !confidence %in% conf) #determinando
classe do argumento e definindo mensagem caso nao seja acatada
  {stop("os niveis de confianca suportados por esta funcao sao de 0.90, 0.95
ou 0.99")}
  if(class(P) != "numeric" & P <= 1 & P > 0 ) #determinando classe do
argumento e definindo mensagem caso nao seja acatada
  { stop("P deve ser numeric e estar entre 0 e 1") }
  if(class(precision) != "numeric" & precision > 0) #determinando classe do
argumento e definindo mensagem caso nao seja acatada
  { stop("precision deve ser numeric e maior que 0") }
  if(is.data.frame(data) == TRUE & position == TRUE) #determinando vetores
logicos para o fluxo da funcao, dadas as condicoes
  {
```

```
    if( population == TRUE & confidence == 0.95) #formula para o calculo amostral e formula de ajuste para populacoes finitas
    {n = 1.96^2 * (P * (1 - P)) / precision^2
    n.aj = n / 1 + (n / N)
    }
    if( population == TRUE & confidence == 0.90) #formula para o calculo amostral e formula de ajuste para populacoes finitas
    { n = 1.645^2 * (P * (1 - P)) / precision^2
    n.aj = n / 1 + (n / N)
    }
    if( population == TRUE & confidence == 0.99) #formula para o calculo amostral e formula de ajuste para populacoes finitas
    { n = 2.58^2 * (P * (1 - P)) / precision^2
    n.aj = n / 1 + (n / N)
    }
    cat(as.vector(round(n.aj, digits = 0))) #arrendonda o valor e imprime no console
    aggregate = sample(data$agr, size = round(n.aj, digits = 0), replace = FALSE) #cria objeto para determinar quais elemento foram sorteados a partir de um data.frame
    print(as.vector(aggregate)) #mostra os elementos sorteados no console
  }

else{
  if(population == FALSE & confidence == 0.95) #formula para o calculo amostral e formula de ajuste para populacoes infinitas
  { n = 1.96^2 * (P * (1 - P)) / precision^2 }
  if( population == FALSE & confidence == 0.90) #formula para o calculo amostral e formula de ajuste para populacoes infinitas
  { n = 1.645^2 * (P * (1 - P)) / precision^2 }
  if( population == FALSE & confidence == 0.99) #formula para o calculo amostral e formula de ajuste para populacoes infinitas
  { n = 2.58^2 * (P * (1 - P)) / precision^2 }
  warning("0 numero amostral foi calculado para uma populacao infinita. Deste modo, nao ha identificacao dos elementos calculados. O sorteio dos elementos sera realizado se for inserido um data.frame, se 'position' = TRUE e se 'population' = TRUE") #mensagem de aviso para reforcar qual operacao foi estabelecida
  return(round(n, digits = 0)) #retorna o valor de n no console. fim da funcao
}
}
```

From:

<http://labtrop.ib.usp.br/> - **Laboratório de Ecologia de Florestas Tropicais**

Permanent link:

http://labtrop.ib.usp.br/doku.php?id=cursos:ecor:05_curso_antigo:r2019:alunos:trabalho_final:joacallefe:codigo 

Last update: **2020/07/27 18:48**