Código da função

```
prev <- function(x,y, teste="rrt", probs=c(0.25, 0.5833), nsim=1000)    # função prev com argumentos: x (dados de respostas randomizadas), y (dados de questionamento direto), teste (tipos de teste desejado: rrt, qd ou ambos), probs (probabilidades de "sim" forçado e "diga a verdade", ver página de ajuda), nsim (número de simulações)
    {
    if(teste=="rrt")    # se usuário optou pela análise de dados de respostas randomizadas
        {
        rrt <- function(x) ((sum(na.omit(x))/length(na.omit(x)))-probs[1])/probs[2]    # fórmula de Hox & Lensvelt-Mulders (2004) para estimar prevalência de comportamento com base em dados RRT
        a <- round(rrt(x), 4)    # criando objeto a com estimativa de prevalência do comportamento para os dados de respostas randomizadas e arredondando
        rrt.boot <- rep(NA, nsim)    # criando objeto rrt.boot para receber valores de simulação a seguir
        for(i in 1:nsim)    # criando contador
            {
            sample.x <- sample(x, replace=TRUE)    # reamostrando vetor de dados de respostas randomizadas (x), com substituição
            rrt.boot[i] <- rrt(sample.x)    # preenchendo objeto rrt.boot com estimativas de prevalência para valores simulados
            }
        b <- round(quantile(rrt.boot, probs=c(0.025, 0.975), na.rm=T), 4) # criando objeto b com quantis referentes ao intervalo de confiança de 95% dos valores simulados e arredondando
        n.NAx <- (length(x))-(length(na.omit(x)))    # calcula numero de NAs no vetor de dados x
        cat("\n\t", n.NAx," valores NA omitidos de x\n\n")    # retorna texto informando número de NAs omitidos de x
        boxplot(rrt.boot, xlab="RRT", ylab="Prevalência")    # boxplot dos valores de prevalência reamostrados por bootstrap (objeto rrt.boot)
        m1 <- data.frame(matrix(c(a, b), ncol = 3, nrow = 1, byrow = TRUE)) # criando objeto m1 com resultados da analise
        rownames(m1) <- "RRT"    # modificando nomes de linhas de m1
        colnames(m1) <- c("prevalencia", "Q 2.5%", "Q 97.5%")    # modificando nomes de colunas de m1
        return(m1)    # retorna objeto m1 com resultados da análise
        }
    if(teste=="qd")    # se usuário optou pela análise de dados de questionamento direto
        {
        qd <- function(y) sum(na.omit(y))/length(na.omit(y))    # fórmula para estimar prevalência de comportamento com base em dados de questionamento direto
        a2 <- round(qd(y), 4)    # criando objeto a2 com estimativa de prevalência para os dados de questionamento direto e arredondando
```

```
        qd.boot <- rep(NA, nsim)    # criando objeto qd.boot para receber
valores de simulação a seguir
        for(i in 1:nsim)    # criando contador
            {
            sample.y <- sample(y, replace=TRUE)    # reamostrando vetor de
dados de questionamento direto (y), com substituição
            qd.boot[i] <- qd(sample.y)    # preenchendo objeto qd.boot com
estimativas de prevalência para valores reamostrados
            }
        b2 <- round(quantile(qd.boot, probs=c(0.025, 0.975), na.rm=T), 4)
# criando objeto b2 com quantis referentes ao intervalo de confiança de 95%
dos valores simulados e arredondando
        n.NAy <- (length(y))-(length(na.omit(y)))    # calcula numero de NAs
no vetor de dados y
        cat("\n\t", n.NAy," valores NA omitidos de y\n\n")    # retorna
texto informando número de NAs omitidos de y
        boxplot(qd.boot, xlab="QD", ylab="Prevalência")    # boxplot dos
valores de prevalência reamostrados por bootstrap (objeto qd.boot)
        m2 <- data.frame(matrix(c(a2, b2), ncol = 3, nrow = 1, byrow =
TRUE))    # criando objeto m2 com resultados da analise
        rownames(m2) <- "QD"    # modificando nomes de linhas de m2
        colnames(m2) <- c("prevalencia", "Q 2.5%", "Q 97.5%")    #
modificando nomes de colunas de m2
        return(m2)    # retorna objeto m2 com resultados da análise
        }
    if(teste=="rrt.qd")    # se usuário optou por análise simultânea de
dados de respostas randomizadas e questionamento direto
        {
        rrt <- function(x) ((sum(na.omit(x))/length(na.omit(x)))-
probs[1])/probs[2]    # fórmula de Hox & Lensvelt-Mulders (2004) para
estimar prevalência de comportamento com base em dados RRT
        a <- round(rrt(x), 4)    # criando objeto a com estimativa de
prevalência do comportamento para os dados de respostas randomizadas e
arredondando
        rrt.boot <- rep(NA, nsim)    # criando objeto rrt.boot para receber
valores de simulação
        for(i in 1:nsim)    # criando contador
            {
            sample.x <- sample(x, replace=TRUE)    # reamostrando vetor de
dados de respostas randomizadas (x), com substituição
            rrt.boot[i] <- rrt(sample.x)    # preenchendo objeto rrt.boot
com estimativa de prevalência para valores reamostrados
            }
        b <- round(quantile(rrt.boot, probs=c(0.025, 0.975), na.rm=T),4)
# criando objeto b com quantis referentes ao intervalo de confiança de 95%
dos valores simulados e arredondando
        qd <- function(y) sum(na.omit(y))/length(na.omit(y))    # fórmula
para estimar prevalência de comportamento com base em dados de
questionamento direto
        a2 <- round(qd(y),4)    # criando objeto a2 com estimativa de
```

```
prevalência para os dados de questionamento direto
        qd.boot <- rep(NA, nsim)    # criando objeto qd.boot para receber
valores de simulação a seguir
        for(i in 1:nsim)    # criando contador
            {
            sample.y <- sample(y, replace=TRUE)    # reamostrando vetor de
dados de questionamento direto (y), com substituição
            qd.boot[i] <- qd(sample.y)    # preenchendo objeto qd.boot com
estimativa de prevalência para valores reamostrados
            }
        b2 <- round(quantile(qd.boot, probs=c(0.025, 0.975), na.rm=T),4)
# criando objeto b2com quantis referentes ao intervalo de confiança de 95%
dos valores simulados e arredondando
        boxplot(rrt.boot, qd.boot, xlab="Metodo de amostragem",
ylab="Prevalência", xaxt="n")    # boxplot das estimativas de prevalência
obtidos pelos dois métodos
        axis(side=1, at=(1:2), labels=c("RRT", "QD"))    # adicionando
labels ao boxplot criado acima
        dif.obs <- a-a2    # criando objeto dif.obs com diferença observada
entre estimativas RRT e QD
        dif <- rep(NA, nsim)    # criando objeto dif que será preenchido por
simulação a seguir
        for(i in 1:nsim)    # criando contador
        {
        dif[i] <- (((sum(na.omit(sample(c(x,y), size=length(x),
replace=TRUE)))/length(na.omit(x)))-probs[1])/probs[2]) -
(sum(na.omit(sample(c(x,y), size=length(x),
replace=TRUE)))/length(na.omit(y)))    # preenche objeto dif com diferenças
entre RRT e QD produzidas por bootstrap
        }
        n <- NROW(dif[dif >= dif.obs | dif <= -dif.obs])    # calculando
quantas vezes o módulo da diferença observada foi obtido nas simulações de
boostrap
        prob <- (n)/nsim    # calculando a probabilidade de o valor
observado ter sido obtido ao acaso
        x11()    # abrindo novo graphic device
        hist(dif, ylab="Frequência", xlab="RRT-QD", main="Distribuição das
diferenças entre as \n estimativas obtidas por RRT e QD")    # plota
histograma exibindo distribuição da diferença entre estimativas de
prevalência obtidas pelos dois métodos nas simulações de boostrap
        legend("topleft", legend=paste("diferença observada =", dif.obs, ",
p = ", prob), bty="n", text.col="red3")    # insere legenda informando a
diferença observada e a probabilidade de amesma ter sido obtida ao acaso
        n.NAx <- (length(x))-(length(na.omit(x)))    # calcula numero de NAs
no vetor de dados x
        cat("\n\t", n.NAx," valores NA omitidos de x\n")    # insere texto
informando numero de valores de NA omitidos de x
        n.NAy <- (length(y))-(length(na.omit(y)))    # calcula numero de NAs
no vetor de dados y
        cat("\n\t", n.NAy," valores NA omitidos de y\n\n")    # insere texto
informando numero de valores de NA omitidos de y
```

```
        m <- data.frame(matrix(c(a, b, a2, b2), ncol = 3, nrow = 2, byrow =
TRUE))    # cria objeto m com resultados da análise
        rownames(m) <- c("RRT", "QD")    # modificando nomes de linhas de m
        colnames(m) <- c("prevalencia", "Q 2.5%", "Q 97.5%")    #
modificando nomes de colunas de m1
        ret.rrt.qd <- list(m, "Probabilidade de diferença entre estimativas
RRT e QD ter sido obtida ao acaso"=prob)    # cria objeto com lista de itens
a serem exibidos no return, incluindo m e prob
        return(ret.rrt.qd)    # retorna os objetos listados acima
(resultados da análise)
        }
    }
```

Página de ajuda

```
prev                package:nenhum                R Documentation

Prevalência de comportamentos sensitivos

Description:

Função para estimar a prevalência de comportamentos sensitivos a partir de
dados obtidos pela técnica de respostas randomizadas (RRT) e questionamento
direto (QD).

Usage:

prev(x, y, teste=c("rrt", "qd", "rrt.qd"), probs=c(0.25, 0.5833), nsim=1000)

Arguments:

x   vetor numérico representando dados obtidos pela técnica de respostas
randomizadas.
y   vetor numérico representando dados obtidos por questionamento direto.
teste   argumento lógico que indica a opção de analise ("rrt" para tecnica
de respostas randomizadas, "qd" para questionamento direto, "rrt.qd" para
ambas)
probs   vetor representando probabilidades de "SIM" forçado e "DIGA A
VERDADE" associados à técnica de respostas randomizadas (Hox & Lensvelt-
Mulders 2004).
nsim    número de simulações.

Details:



Os dados de entrada devem ser binários, na forma de 0 e 1 representando
respostas NAO e SIM respectivamente. A fórmula para cálculo de prevalência
com base em respostas randomizadas corresponde ao método de "respostas
forçadas" (Lensvelt-Mulders et al. 2005), onde as probabilidades de "SIM
FORÇADO" e "DIGA A VERDADE" podem ser definidas pelo usuário (default 0.25 e
```

0.5833 para "SIM FORÇADO" e "DIGA A VERDADE" respectivamente). A fórmula para cálculo de prevalência com base em questionamento direto corresponde à proporção de respostas SIM em relação ao tamanho da amostra (Lee 1993). Na opção de teste "rrt.qd", a função executa um teste de permutação comparando a diferença observada entre estimativas de prevalência obtidas por respostas randomizadas e questionametno direto com diferenças obtidas por meio de reamostragem dos dados originais. O número de reamostragens é definido pelo usuário com o argumento nsim.

Value:

A função retorna um dataframe com a estimativa de prevalência do comportamento de acordo com o teste selecionado e com os limites inferior e superior do intervalo de confiança de 95% da estimativa, gerados por bootstrap. A função também retorna um boxplot representando estimativas de prevalência geradas por bootstrap. Caso o usuário opte pelo teste "rrt.qd", a função retorna ainda o resultado de um teste de permutação bicauldal comparando as estimativas obtidas pelos dois métodos, e um histograma com a distribuição das diferenças entre as estimativas obtidas pelos dois métodos.

Warnings:

A função roda mesmo se houverem NAs e retorna um aviso informando o número de NAs omitidos para cada vetor. Estimativas de prevalência com valor negativo podem ser obtidas por simulação.

Note:

....

Author(s):

Elildo Alves Ribeiro de Carvalho Jr (CENAP/ICMBio)
contato: elildojr@gmail.com, elildo.carvalho-junior@icmbio.gov.br

References:

Hox, J. & Lensvelt-Mulders, G. 2004. Randomized response analysis in Mplus. Struct. Equ. Model. 11,615–620. (doi:10.1207/s15328007sem1104_6)
Lee, R.M. 1993. Doing research on sensitive topics. Sage Publications, London.
Lensvelt-Mulders, G.J.L.M., Hox, J.J., van der Heijden, P.G.M. 2005. How to improve the efficiency of randomised response designs. Qual. Quantity 39, 253–265. (doi:10.1007/s11135-004-0432-3)

See Also:

```
     ~~objects to See Also as 'help', ~~~


Examples:

x <- rbinom(n=200, size=1, prob=0.5)    # simula conjunto de dados x com 50%
de respostas SIM
y <- rbinom(n=200, size=1, prob=0.2)    # simula conjunto de dados y com 20%
de respostas SIM

prev(x, teste="rrt")    # estima prevalência por respostas randomizadas
prev(,y, teste="qd")    # estima prevalência por questionamento direto
prev(x, y, teste="rrt.qd")    # estima prevalência pelos dois métodos e
executa teste de permutação comparando estimativas
prev(x, y, teste="rrt.qd", nsim=2000)    # idem, mas modificando número de
simulações
```

Arquivo da função prev.r

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:elildojr:trabalho_final_-_codigo_e_help

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