

Função "pairdiff"

A função pode ser utilizada para avaliar se as diferenças entre atributos em pares de espécies que interagem em uma comunidade são maiores ou menores do que as esperadas pelo acaso e essas diferenças poderiam explicar quando a interação é mantida ou perdida ao longo do tempo. O mesmo pode ser aplicado para avaliar interações entre pessoas a partir de dados quantitativos de suas preferências ou características físicas e sociais.

Página de ajuda

pairdiff

package:-

R Documentation

Cálculo da diferença média de atributos entre pares de elementos que interagem e de sua probabilidade de ocorrência.

Description:

Para dois conjuntos de elementos que interagem em pares e que podem manter ou perder a interação ao longo do tempo, a função deve calcular a diferença média entre atributos quantitativos dos elementos pareados associada à manutenção e à perda da interação.

Além disso, a função deve calcular a probabilidade de as diferenças médias observadas nos dois casos ocorrerem ao acaso. Para cada atributo, um histograma com a distribuição simulada de diferenças médias, contendo a indicação das diferenças observadas para interações do tipo 01 e do tipo 02, será gerado.

Usage:

```
pairdiff(i, a, nsim = 1000, teste = "menor", graf = c(1,1), col = c("red", "blue"))
```

Arguments:

i Matriz de interação cujas linhas correspondem ao primeiro conjunto de elementos e as colunas ao segundo conjunto. Quando não há interação entre os elementos dos dois conjuntos, a célula da matriz referente à linha e à

coluna dos elementos em questão deve estar preenchida com o valor 0. Quando há interação e esta não foi mantida ao final do intervalo considerado, a célula deve ser preenchida com 1 (interação tipo 01). Quando há interação e esta foi mantida, a célula deve ser preenchida com 2 (interação tipo 02).

a Matriz ou data frame no qual as linhas correspondem aos elementos dos dois conjuntos e as colunas correspondem a cada um dos atributos considerados. As células devem ser preenchidas com o valor de cada atributo para o elemento da linha correspondente.

nsim Número de ciclos de simulação, sendo que em cada ciclo é gerado um valor de diferença média entre os elementos de um par para cada atributo. Em cada ciclo, os valores de cada atributo são embaralhados entre os elementos e as diferenças médias entre os valores de atributo dos elementos de um par interagente são calculadas. Default é de 1000 ciclos de simulação.

teste Teste lógico usado no cálculo da probabilidade de as diferenças médias observadas entre elementos pareados serem geradas ao acaso. Se teste = "menor" (default), calcula-se a probabilidade de: a) diferenças médias IGUAIS OU MENORES que as observadas para interações do tipo 01 ocorrerem em uma distribuição ao acaso e b) diferenças médias MAIORES que as observadas para interações do tipo 02 ocorrerem em uma distribuição ao acaso. Se teste = "maior", calcula-se a probabilidade de: a) diferenças médias IGUAIS OU MAIORES que as observadas para interações do tipo 01 ocorrerem em uma distribuição ao acaso e b) diferenças médias MENORES que as observadas para interações do tipo 02 ocorrerem em uma distribuição ao acaso.

graf Vetor numérico cujas posições determinam, respectivamente, o número de linhas e colunas da janela gráfica na qual os histogramas serão gerados. Se graf =

`c(1,1)` (default), uma janela gráfica será aberta para cada histograma. Caso contrário, o usuário deve fornecer um conjunto que permita a construção de todos os histogramas em APENAS UMA janela.

`col` Vetor da classe "character" cujas posições determinam, respectivamente, a cor de indicação das diferenças observadas para interações do tipo 01 e do tipo 02.

Details:

Os objetos "i" e "a" e os argumentos "nsim" e "teste" devem ser obrigatoriamente fornecidos para a execução da função.

Value:

A função retorna:

Um histograma para cada atributo, com a distribuição simulada de diferenças médias e a indicação das diferenças observadas para interações do tipo 01 e do tipo 02. (Ver "Arguments": 'graf' para disposição dos histogramas).

Um lista contendo:

`comp1:` Data frame com informações gerais sobre a matriz de interações, com:

elementos do conjunto 1, II) número de elementos do conjunto 2, III) número de pares de interação do tipo 01, IV) número de pares de interação do tipo 02.

`comp2:` Data frame com informações gerais sobre as diferenças médias entre elementos pareados. Cada coluna corresponde a um atributo e as linhas correspondem a:

I) diferença média observada para pares de interação do tipo 01, II) diferença média observada para pares de interação do tipo 02, III) média das diferenças médias simuladas, IV) desvio padrão das diferenças médias simuladas, V) probabilidade de a diferença média

observada para interação do tipo 01 ocorrer na distribuição
ao acaso,
e VI) probabilidade de a diferença média observada para
interação do tipo 02 ocorrer na distribuição ao acaso.

comp3: Ocorrência de NA no objeto "a", indicando a linha e a coluna
de ocorrência. Esse componente só é retornado se houver NA no
objeto.

comp4: Ocorrência de NA no objeto "i", indicando a linha e a coluna
de ocorrência. Esse componente só é retornado se houver NA no
objeto.

Warnings:

A função é interrompida e retorna mensagens de erro se o objeto "i" não for
matriz, o
objeto "a" não for um data frame ou uma matriz, se o argumento "teste" não
for "menor"
ou "maior" e se os argumentos "graf" e/ou "col"
não tiverem o comprimento correto.

Notes:

Para melhor visualização dos histogramas, é recomendado abrir uma janela
gráfica. No
Windows, use o comando `x11()`, por exemplo. Para entender os parâmetros da
janela
gráfica, veja: "Arguments: 'graf'".

A função aceita objetos de entrada que contenham valores NA. No entanto,
pares em que o
valor de um dado atributo seja NA para ao menos um dos elementos serão
excluídos do
cálculo da diferença média para o atributo em questão.

Author(s):

Luanne Caires
lcaires@usp.br

São Paulo, 14 de maio de 2016.

References:

Valiente-Banuet et al. 2008. Temporal shifts from facilitation to competition occur between closely related taxa. *Journal of Ecology* 96: 489 - 494. (para uma abordagem semelhante à da função, mas utilizando distâncias filogenéticas como diferenças entre pares de espécies de plantas).

See Also:

Funções: `array()`, `for()`, `which()`, `mapply()`, `hist()`

Acknowledgments:

A Rodolfo Liporoni, Vinicius Biffi e Lucas Teixeira, pelas dicas de comandos e argumentos.

Examples:

```
# Criando a matriz de interacao (objeto i)
matriz.int = matrix(sample(rep(c(0,1,2), times = 6)),nrow=9,ncol=6)
colnames(matriz.int) = paste("N", 1:6)
rownames(matriz.int) = paste("F", 1:9)

# Criando a matriz de interacao com NA (objeto i)
matriz.int2 = matriz.int
matriz.int2[5, 1] = NA

# Criando o data frame de atributos (objeto a)
atributos = as.data.frame(matrix(round(runif(45,0.3,3.2),2),nrow=15,ncol=3))
rownames(atributos) = c(paste("N", 1:6), paste("F", 1:9))
colnames(atributos) = c(paste("Atributo",1:3))

# Criando o data frame de atributos com NA (objeto a)
atributos2 = atributos
atributos2[3,2] = NA

pairdiff(matriz.int, atributos)

pairdiff(matriz.int, atributos, 2000, "menor", c(2,2), c("green", "orange"))

pairdiff(matriz.int2, atributos2, 2000, "maior")
```

Código da função

```
##### FUNCAO "pairdiff"
#####

pairdiff = function(i, a, nsim = 1000, teste="menor", graf = c(1,1), col =
c("red", "blue")){

### RENOMEANDO OS OBJETOS DE ENTRADA DE ACORDO COM OS NOMES A SEREM USADOS
NA EXECUCAO DA FUNCAO
#####
  m.int = i
  atrib = a
#####
  ####
  ### CONFERINDO SE OS DADOS DE ENTRADA FORAM FORNECIDOS E ESTAO CORRETOS
  #####
  ### Conferindo se os objetos de entrada sao das classes
  permitidas#####
  if(class(m.int) != "matrix"){ #se a matriz de interacao (objeto i) nao for
da classe matriz
    stop("0 objeto 'i' nao e uma matriz")} #para a funcao e retorna uma
mensagem de erro
  if(class(atrib) != "data.frame" & class(atrib) != "matrix"){ #se o objeto
com as informacoes dos atributos (a) nao for um data.frame ou uma matriz
    stop("0 objeto 'a' nao e um data.frame ou uma matriz")} #para a funcao e
retorna uma mensagem de erro

### Conferindo se os argumentos foram corretamente fornecidos
#####
  if(teste != "menor" & teste != "maior"){ #se o argumento "teste" nao
constar das opcoes aplicaveis a funcao
    stop("0 argumento 'teste' nao foi fornecido corretamente")} #para a
funcao e retorna uma mensagem de erro
  if(length(graf) != 2 | length(col) != 2){ #se os argumentos nao tiverem
duas posicoes
    stop("0 argumento 'graf' e/ou 'col' nao tem o comprimento adequado")}
#para a funcao e retorna uma mensagem de erro
#####
  #####
  ### TRANSFORMANDO "NA" NA MATRIZ DE INTERACAO EM 9 (VALOR SEM SIGNIFICADO)
  #####
  m.int[is.na(m.int)] = 9
#####
  #####
  ### REORDENANDO OS ELEMENTOS NO DATA FRAME DE ATRIBUTOS
  #####
```

```

  atrib = atrib[c(rownames(m.int), colnames(m.int)),] # reordena os
  elementos do data frame de atributos de acordo com sua ordem nos rotulos da
  matriz de interacoes (m.int). Esse passo sera importante no calculo das
  diferencas entre os valores de atributos a seguir
#####
####
### CRIANDO UM ARRAY NO QUAL CADA CAMADA (EIXO Z) CONTERA A DIFERENCA ENTRE
  PARES DE ELEMENTOS PARA UM ATRIBUTO
#####
  a.dif = array(m.int, dim = c(dim(m.int)[1],
  dim(m.int)[2],(dim(atrib)[2]+1)))
#####
####
### CALCULANDO AS DIFERENCAS ENTRE OS VALORES DE ATRIBUTOS DOS PARES
#####
  for(i in 1:dim(atrib)[2]){ #cria um ciclo de calculo para cada atributo
    a1 = atrib[1:dim(m.int)[1],i] # cria um vetor com valores de atributos
    para elementos do conjunto 1 (linhas da matriz de interacao)
    a2 = atrib[(dim(m.int)[1] + 1):(dim(m.int)[1] + dim(m.int)[2]),i] # cria
    um vetor com valores de atributos para elementos do conjunto 2 (colunas da
    matriz de interacao)
    rep.a2 = rep(a2,each=dim(m.int)[1]) # replica o vetor a2 para
    preenchimento do array por linhas, nao por colunas. Nas operacoes
    matematicas a seguir, o default do R soma, para cada coluna, as linhas da
    matriz com os valores do vetor. Com esse passo, cada linha de cada coluna
    serã somada com o mesmo valor de atributo
    int.a2 = m.int + rep.a2 #cria uma matriz na qual,para cada par, o valor
    da interacao (0,1 ou 2) ã somado ao valor do atributo para o conjunto 2
    int.dif = int.a2 - a1 #para cada par, desconta do objeto anterior a
    diferenca do valor do atributo para o conjunto 1
    dif = abs(int.dif - m.int) #para cada par, desconta do objeto anterior o
    valor da interacao, restando apenas uma matriz com o valor absoluto das
    diferencas entre os pares
    a.dif[ , ,i+1] = dif #adiciona a matriz dif (com os valores das
    diferencas entre os pares) ao array de diferencas. Nesse array, a primeira
    camada (eixo z) corresponde a matriz de interacao e cada camada seguinte
    corresponde a um atributo
  }
#####
####

### CALCULANDO A DIFERENCA MEDIA PARA OS PARES DE INTERACAO DO TIPO 01
#####
  int1 = which(m.int==1,arr.ind=T) #identifica os pares de interacao do tipo
  01
  dif1 = a.dif[ , ,2:(dim(atrib)[2]+1)][a.dif[ , ,1]==1] #cria um vetor com os
  valores das diferencas de cada par de todos os atributos
  m.dif1 = matrix(dif1, nrow=dim(int1)[1], ncol=dim(atrib)[2]) #cria uma
  matriz em que cada linha contem a diferenca entre um par de interacao do
  tipo 01 e as colunas correspondem aos atributos
  dm1 = mapply(mean, as.data.frame(m.dif1), na.rm = TRUE) #cria um vetor em

```

```
que cada posicao corresponde a diferenca media entre os pares para um
atributo. As medias sao calculadas excluindo-se as diferencas de pares em
que ao menos um dos elementos tenha valor faltante do atributo em questao
#####
#####
### CALCULANDO A DIFERENCA MEDIA PARA OS PARES DE INTERACAO DO TIPO 02
#####
  int2 = which(m.int==2,arr.ind=T) #identifica os pares de interacao do tipo
2
  dif2 = a.dif[ , 2:(dim(atrib)[2]+1)][a.dif[,1]==2] #cria um vetor com os
valores das diferencas de cada par de todos os atributos
  m.dif2 = matrix(dif2, nrow=dim(int2)[1], ncol=dim(atrib)[2]) #cria uma
matriz em que cada linha contem a diferenca entre um par de interacao do
tipo 02 e as colunas correspondem aos atributos
  dm2 = mapply(mean, as.data.frame(m.dif2), na.rm = TRUE) #cria um vetor em
que cada posicao corresponde a diferenca media entre os pares para um
atributo. As medias sao calculadas excluindo-se as diferencas de pares em
que ao menos um dos elementos tenha valor faltante do atributo em questao
#####
#####
### SIMULANDO DIFERENCAS ALEATORIAS ENTRE OS PARES DE INTERACAO
#####
  int12 = which(m.int==1 | m.int==2, arr.ind=T) #identifica os pares de
interaco do tipo 1 e 2
  a.difsim = array(m.int, dim = c(dim(m.int)[1],
dim(m.int)[2],(dim(atrib)[2]+1))) #cria um array no qual cada camada (eixo
z) contera a diferenca simulada entre os pares de especies para um atributo
  dmsim = matrix(data = NA, nrow = nsim, ncol = dim(atrib)[2]) #cria matriz
para guardar as diferencas simuladas entre os pares
  for(s in 1:nsim){ #cria "nsim" ciclos de simulacao
    atbmix = apply(atrib, 2, sample) #cria um data.frame de atributos com os
valores de cada atributo embaralhados entre os elementos
    for(i in 1:dim(atrib)[2]){ #cria um ciclo de simulacao para cada
atributo
      a1 = atbmix[1:dim(m.int)[1],i] #cria um vetor com os valores
embaralhados de atributos para elementos do conjunto 1 (linhas da matriz de
interacao)
      a2 = atbmix[(dim(m.int)[1] + 1):(dim(m.int)[1] + dim(m.int)[2]),i]
#cria um vetor com os valores embaralhados de atributos para elementos do
conjunto 2 (colunas da matriz de interacao)
      rep.a2 = rep(a2,each=dim(m.int)[1]) #cria um vetor para preenchimento
do array por linhas, nao por colunas
      int.a2 = m.int + rep.a2 #cria uma matriz na qual o valor da interacao
(0,1 ou 2) soma-se ao valor do atributo para o conjunto 2
      int.dif = int.a2 - a1 #desconta do objeto anterior a diferenca do
valor do atributo para o conjunto 1
      dif = abs(int.dif - m.int) #desconta do objeto anterior o valor da
interacao, restando apenas o valor absoluto das diferencas simuladas entre
os pares
      a.difsim[ , ,i+1] = dif #adiciona a matriz dif (com os valores das
```


diferencas simuladas entre os pares) ao array de diferencas. Nesse array, a primeira camada (eixo z) corresponde a matriz de interacao e cada camada seguinte corresponde a um atributo

```
difl2 = a.difsim[ , ,2:(dim(atrib)[2]+1)][a.difsim[ , ,1]==1|a.difsim[ , ,1]==2] #cria um vetor com os valores das diferencas simuladas de cada par de interacao dos tipos 01 e 02 para todos os atributos
```

```
m.difl2 = matrix(difl2, nrow=dim(intl2)[1], ncol=dim(atrib)[2]) ##cria uma matriz em que cada linha contem a diferenca simulada de um par de interacao (tipo 1 ou 2) e as colunas correspondem aos atributos
```

```
dm12 = mapply(mean, as.data.frame(m.difl2), na.rm = TRUE) #cria um vetor em que cada posicao corresponde a diferenca media simulada dos pares para um atributo. As medias sao calculadas excluindo-se as diferencas de pares em que ao menos um dos elementos tenha valor faltante do atributo em questao
```

```
dmsim[s, ] = dm12 #adiciona os valores das diferencas medias simuladas dos pares na matriz criada antes do ciclo. As linhas correspondem aos ciclos de simulacao e as colunas aos atributos
```

```
}}
```

```
#####  
#####
```

```
### CALCULANDO A PROBABILIDADE DE AS DIFERENCAS MEDIAS OBSERVADAS SEREM  
OBTIDAS AO ACASO
```

```
#####
```

```
dmsim[1,] = dm1 #insere as diferencas medias observadas para interacao do tipo 01 na primeira posicao da matriz de medias simuladas
```

```
dmsim[2,] = dm2 #insere as diferencas medias observadas para interacao do tipo 02 na segunda posicao da matriz de medias simuladas
```

```
rep.dm1 = rep(dm1, each = nsim) #replica o vetor de diferencas medias observadas para interacao tipo 01 de modo que a media para cada atributo seja repetida de acordo com o numero de simulacoes. Esse vetor replicado sera usado no teste logico
```

```
rep.dm2 = rep(dm2, each = nsim) #replica o vetor de diferencas medias observadas para interacao tipo 02, de modo que a media para cada atributo seja repetida de acordo com o numero de simulacoes. Esse vetor replicado sera usado no teste logico
```

```
if(teste=="menor"){ #determina o tipo de teste logico
```

```
menor1 = dmsim <= rep.dm1 #teste logico para valores ao acaso iguais ou menores que os observados para interacao tipo 01. Para cada atributo (coluna de dmsim), a diferenca media simulada (cada linha de dmsim) é testada contra a diferenca media observada para aquele atributo (cada valor distinto de rep.dm1)
```

```
sum1 = apply(menor1, 2, sum) #vetor em que cada posicao e a quantidade de valores que atendem ao teste do passo anterior para o atributo em questao  
prob1 = sum1 / nsim #probabilidade de valores iguais ou menores aos das diferencas medias observadas para interacao tipo 01 ocorrerem em uma distribuicao ao acaso
```

```
maior2 = dmsim >= rep.dm2 #teste logico para valores ao acaso maiores que os observados para interacao tipo 2
```

```
sum2 = apply(maior2, 2, sum) #vetor em que cada posicao e a quantidade de valores que atendem ao teste do passo anterior para o atributo em questao  
prob2 = sum2 / nsim #probabilidade de valores maiores aos das diferencas
```

```
medias observadas para interacao tipo 02 ocorrerem em uma distribuicao ao
acaso
}
if(teste=="maior"){ #determina o tipo de teste logico
  maior1 = dmsim >= rep.dm1 #teste logico para valores ao acaso maiores
que os observados para interacao tipo 01
  sum1 = apply(maior1, 2, sum) #quantidade de valores que atendem ao teste
do passo anterior
  prob1 = sum1 / nsim #probabilidade de os valores ocorrerem ao acaso
  menor2 = dmsim <= rep.dm2 #teste logico para valores ao acaso menores
que os observados para interacao tipo 02
  sum2 = apply(menor2, 2, sum) #quantidade de valores que atendem ao teste
do passo anterior
  prob2 = sum2 / nsim #probabilidade de os valores ocorrerem ao acaso para
cada atributo
}
#####
####
### CONSTRUINDO GRAFICOS DA DISTRIBUICAO DE DIFERENCAS MEDIAS PARA CADA
ATRIBUTO
  if(graf[1] != 1 & graf[2] != 1){ #se a janela grafica contiver mais de um
grafico
    par(mfrow = graf) #com os parametros determinados pelo usuario
    for(i in 1:dim(dmsim)[2]){ #cria um ciclo de plotagem grafica
      hdados = hist(dmsim[,i], plot = FALSE) #cria um objeto a partir do
qual sera possivel obter informacoes sobre a construcao do histograma pelo
R, incluindo os breaks dos intervalos do eixo x e a frequencia maxima para
esses intervalos (max(hdados$counts))
      hist(dmsim[,i], xlab = colnames(atrib)[i], ylab = "Frequencia", main
= "Diferencas medias entre os pares", axes = FALSE) #constroi um histograma
com as diferencas medias simuladas para cada atributo. O histograma nao
contera os eixos x e y.
      y = round(seq(0,max(hdados$counts),length.out=10)) #constroi o eixo y,
que deve ter como valor maximo a frequencia maxima para os intervalos do
histograma
      x = round(seq(min(hdados$breaks), max(hdados$breaks), length.out =
length(hdados$breaks)), digits = 2) #constroi o eixo x de acordo com os
breaks do histograma
      axis(2,at = y, las = 1, cex.axis = 0.9, tcl = -0.2, mgp =
c(3,0.4,0.5)) #adiciona o eixo y
      axis(1,at = x, las = 2, cex.axis = 0.9, tcl = -0.2, mgp =
c(3,0.5,0.7)) #adiciona o eixo x
      abline(v = dm1[i], col = col[1]) #traca uma linha indicando as
diferencas medias observadas para interacoes do tipo 01
      abline(v = dm2[i], col = col[2]) #traca uma linha indicando as
diferencas medias observadas para interacoes do tipo 02
      legend("topright", legend = c("Int. 1", "Int. 2"), bty = "n", cex =
0.9, pch = 16, col = col, pt.cex = 1.2) #insere a legenda de cores
    }
  }
  if(graf[1] == 1 & graf[2] == 1){ #se a janela grafica contiver apenas um
```

```

grafico
  for(i in 1:dim(dmsim)[2]){ #cria um ciclo de plotagem grafica
    par(mfrow = graf) #com o graf = c(1,1)
    hdados = hist(dmsim[,i], plot = FALSE) #cria um objeto a partir do
    qual sera possivel obter informacoes sobre a construcao do histograma pelo
    R, incluindo os breaks dos intervalos do eixo x e a frequencia maxima para
    esses intervalos (max(hdados$counts))
    hist(dmsim[,i], xlab = colnames(atrib)[i], ylab = "Frequencia", main =
    "Diferencas medias entre os pares", axes = FALSE) #constroi um histograma
    com as diferencas medias simuladas para cada atributo. O histograma nao
    contera os eixos x e y.
    y = round(seq(0,max(hdados$counts),length.out=10)) #constroi o eixo y
    x = round(seq(min(hdados$breaks), max(hdados$breaks), length.out =
    length(hdados$breaks)), digits = 2) #constroi o eixo x
    axis(2,at = y, las = 1, cex.axis = 0.9, tcl = -0.2, mgp =
    c(3,0.4,0.5)) #adiciona o eixo y
    axis(1,at = x, las = 2, cex.axis = 0.9, tcl = -0.2, mgp =
    c(3,0.5,0.7)) #adiciona o eixo x
    abline(v = dm1[i], col = col[1]) #traca uma linha indicando as
    diferencas medias observadas para interacoes do tipo 01
    abline(v = dm2[i], col = col[2]) #traca uma linha indicando as
    diferencas medias observadas para interacoes do tipo 02
    legend("topright", legend = c("Int. 1", "Int. 2"), bty = "n", cex =
    0.9, pch = 16, col = col, pt.cex = 1.2) #adiciona a legenda de cores
  }
  par(mfrow = c(1,1)) #retorna a janela grafica as condicoes padroes
#####
####
### CRIANDO OS OBJETOS DE SAIDA COM AS INFORMACOES SOBRE OS DADOS E AS
SIMULACOES
## Objeto de saida 01: Informacoes sobre a matriz de interacao
n1 = dim(m.int)[1] #numero de elementos do conjunto 1
n2 = dim(m.int)[2] #numero de elementos do conjunto 2
npar1 = dim(int1)[1] #numero de pares de interacao do tipo 01
npar2 = dim(int2)[1] #numero de pares de interacao do tipo 02
tab1 = data.frame(n1, n2,npar1, npar2) #constroi o primeiro data frame de
saida com informacoes gerais sobre a matriz de interacoes
tab1 = t(tab1) #transpoe o data frame anterior para melhor visualizacao
dos dados
rownames(tab1) = c("Elementos do conjunto 01", "Elementos do conjunto 02",
"Interacoes tipo 01", "Interacoes tipo 02") #rotula as linhas do objeto de
acordo com a informacao que ela contem
colnames(tab1) = "n" #rotula a coluna do objeto
## Objeto de saida 02: Informacoes sobre os dados e as simulacoes
med.sim = apply(dmsim, 2, mean) #media das diferencas medias simuladas
para um par para cada atributo
sd.sim = apply(dmsim, 2, sd) #desvio padrao das diferencas medias
simuladas
tab2 = data.frame(dm1, dm2, med.sim, sd.sim, prob1, prob2) #constroi o
segundo data.frame de saida com informacoes sobre as diferencas medias para
cada atributo

```

```
rownames(tab2) = paste("Atributo", 1:dim(atrib)[2]) #nomeia as linhas do
objeto de acordo os atributos a que se referem
colnames(tab2) = c("Dif.media para interacao tipo 01", "Dif.media para
interacao tipo 02", "Media das dif.medias simuladas", "sd das dif.medias
simuladas", "Prob. diferenca media para int. tipo 01", "Prob. diferenca
media para int. tipo 02") #nomeia as colunas do objeto de acordo as
informacoes a que se referem
tab2 = round(t(tab2), 4) #transpoe a tabela de informacoes, facilitando a
visualizacao da tabela.
#####
####
### READICIONANDO OS 'NA' NA MATRIZ DE INTERACAO
#####
m.int[m.int == 9] = NA #passo importante para o retorno da ocorrencia de
NA nos objetos de entrada

#####
####

### LOCALIZANDO OCORRENCIA DE NA NOS OBJETOS DE ENTRADA
#####
a.na = which(is.na(atrib), arr.ind=T) #cria um objeto com a posicao de NAs
no data frame de entrada dos atributos
i.na = which(is.na(m.int), arr.ind=T) #cria um objeto com a indicacao da
posicao de NAs na matriz de interacao
#####
####
### DEFININDO QUAIS OBJETOS DE SAIDA DEVEM SER RETORNADOS NA FORMA DE LISTA
#####
## Se houver NA nos dois objetos de entrada:
if(sum(is.na(atrib))>0 & sum(is.na(m.int))>0){
  sumario = list(tab1, tab2, a.na, i.na) #cria um objeto unico de saida em
forma de lista contendo todos os objetos a serem retornandos na funcao
  names(sumario) = c("Informacoes gerais sobre a matriz de interacoes",
"Informacoes gerais sobre as diferencas entre os pares", "Ocorrencia de NA
no objeto de atributos", "Ocorrencia de NA na matriz de interacao") #nomeia
cada item da lista de saida de acordo com as informacoes que contem
}
## Se houver NA apenas no objeto de entrada contendo os atributos:
if(sum(is.na(atrib))>0 & sum(is.na(m.int))==0){
  sumario = list(tab1, tab2, a.na) #cria um objeto unico de saida em forma
de lista contendo todos os objetos a serem retornandos na funcao
  names(sumario) = c("Informacoes gerais sobre a matriz de interacoes",
"Informacoes gerais sobre as diferencas entre os pares", "Ocorrencia de NA
no objeto de atributos") #nomeia cada item da lista de saida de acordo com
as informacoes que contem
}
## Se houver NA apenas na matriz de interacao:
if(sum(is.na(atrib))==0 & sum(is.na(m.int))>0){
  sumario = list(tab1, tab2, i.na) #cria um objeto unico de saida em forma
```

```
de lista contendo todos os objetos a serem retornados na funcao
  names(sumario) = c("Informacoes gerais sobre a matriz de interacoes",
"Informacoes gerais sobre as diferencas entre os pares", "Ocorrencia de NA
na matriz de interacao") #nomeia cada item da lista de saida de acordo com
as informacoes que contem
}
## Se não houver NA em nenhum dos objetos de entrada:
if(sum(is.na(atrib))==0 & sum(is.na(m.int))==0){
  sumario = list(tab1, tab2) #cria um objeto unico de saida em forma de
lista contendo todos os objetos a serem retornados na funcao
  names(sumario) = c("Informacoes gerais sobre a matriz de interacoes",
"Informacoes gerais sobre as diferencas entre os pares") #nomeia cada item
da lista de saida de acordo com as informacoes que contem
}
return(sumario) #retorna a lista como objeto unico de saida
}

##### FIM
#####
```

Arquivos da função

[Função pairdiff - Documentação](#)

[Função pairdiff - Código da função](#)

[Função pairdiff - Código dos exemplos](#)

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:lcaires:funcao

Last update: **2020/08/12 09:04**

