

Neri N. H. V.



Meu interesse é trabalhar em **ecologia, medio ambiente, biodiversidade e mudanças climáticas**. O alvo de meu trabalho de pesquisa é aprender a trabalhar com modelos usando só base de dados disponíveis de pesquisas anteriores.

Em razão do exposto, escolhi como linha de pesquisa trabalhar analisando como as *mudanças na estrutura da paisagem e outros fatores podem influenciar a presença de um evento determinado em um determinado tempo*, no caso de minha tese: casos de *Leishmanioses no estado de São Paulo* durante os últimos 15 anos. Aprender a mexer muito bem em todas as ferramentas informáticas que possam me ajudar a responder as hipóteses de meu trabalho é a razão pela qual escolhi esta disciplina.

Meus Exercícios

[Script Exercícios](#)

[Proposta de trabalho final](#)

Trabalho final

Página de ajuda

Data Input

Descrição

“reordenar” é uma função que lê uma tabela de um arquivo cujo formato tem cabeçalhos em todas as colunas e nomes nas linhas, e os dados numéricos no interior da tabela. Essa função, transforma essa tabela em um data.frame com os valores reestruturados.

Uso

```
reordenar(documento, sep, transpor= FALSE,colunomes,docfinal, resumo=FALSE, histograma=FALSE)
```

Argumentos

`documento` nome do arquivo que vai ser lido do diretório. Ele deve ser uma cadeia de caracteres (do tipo `string`)(ver `file`).

`sep` vetor de dois elementos da classe `caractere` que é a cadeia de campo separador. Os valores dentro de cada linha do documento estão

separados por esta cadeia.

`transpor` FALSE por padrão, caso seja TRUE a tabela vai ser transposta antes de reordenar os valores.

`colunomes` vetor da classe caractere com os nomes das três colunas que vai ter a nova tabela. A primeira coluna tem os valores numéricos por padrão.

`docfinal` nome do arquivo, cadeia de caracteres que vai ser salvo no console segundo o diretório de trabalho.

`resumo` FALSE por padrão, caso seja TRUE além da tabela ele gera uma lista na tela com o total de valores por cada fator, os três valores máximos numéricos com seus fatores associados, três valores mínimos diferentes de zero, número de zeros presentes nos dados, e quantidade de NAs.

`histograma` FALSE por padrão, caso seja TRUE será feito um histograma dos valores numéricos totais.

Detalhes

A função muda a ordem da tabela, após ler o arquivo e dependendo se o usuário escolheu a opção `transpor` a tabela ou não previamente.

Os dados numéricos ficarão em uma coluna só, que vai ser a primeira coluna do `data.frame`.

Os cabeçalhos das linhas e colunas serão reconhecidos como fatores e inseridos como a segunda e terceira coluna do `data.frame` respectivamente.

O resultado da função mostra na tela o `data.frame` reestruturado e também salvará o arquivo segundo o formato que o usuário deseje (`.txt`, `.csv`, ver `write.table`).

Caso o usuário deseje, ele também tem a opção de pedir um resumo com a somatória dos dados por fator e também terá a opção de gerar um histograma dos dados numéricos.

Valores

A tabela gerada é um `data.frame` de um formato que permite ser reconhecido para análises posteriores.

Caso usuário erre a entrada dos argumentos, a função gera uma mensagem de advertência.

Autor(es)

Nerida Nadia H. Valero
neridanadia@gmail.com

Referências

Crawley, M.J., 2012. The R book. John Wiley & Sons.

```
#####
##Exemplos

G=paste("T",sep="", seq(1,4))
A= c(5,4,0,2)
B= c(9,0,0,2)
C= c(6,1,NA,0)
D= c(2,0,2,2)
E= c(5,18,2,5)

prova<-data.frame(G=G, A=A, B=B, C=C, D=D, E=E)
prova

write.table(prova,file="prova.csv", sep=";", row.names = FALSE, na="NA",
dec=".", col.names = TRUE)
write.table(prova,file="prova2.csv", sep="," , row.names = FALSE, na="NA",
dec=".", col.names = TRUE)
write.table(prova,file="prova.txt", sep="\t", row.names = FALSE, na="NA",
dec=".", col.names = TRUE)

reordenar("prova.csv", sep=c(";", "\t"), transpor=
FALSE, colunomes=c("c1", "m1", "a1"), "pru1.txt")
reordenar("prova2.csv", sep= c(", ", " "), transpor=
FALSE, colunomes=c("c1", "m1", "a1"), "pru1.txt", resumo=TRUE, histograma=
TRUE)
reordenar("prova.txt", sep=c("\t", "\t"), transpor=
TRUE, colunomes=c("c1", "m1", "a1"), "pru2.txt")
```

Código da função

```
##Funcao reordenar
reordenar<-function(documento, sep, transpor= FALSE,colunomes,docfinal,
resumo=FALSE, histograma=FALSE)#entrada
{
  sepa= c(NA,NA)## vetor vazio que salva os separadores inicial e final
  sepa=sep## recebe os separadores
  if(is.character(documento) == FALSE )##se o nome do arquivo não esta entre
  aspas
  {
    stop("o argumento tem que ser um arquivo externo, ponha a ruta entre
    aspas")## mensagem de erro que alerta a forma em que deve ser escrito o
    input
  }
  if(missingArg(colunomes) )## se os nomes das novas colunas não são
  introduzidos
  {
    stop("tem que escrever os cabecalhos das colunas")
  }
  if(missing(docfinal))#se o formato do arquivo não é especificado
```

```
{
  stop("tem que especificar o nome e formato do arquivo final: .txt,
.csv")
}
if (transpor == FALSE)##indica que o usuario nao quer transpor a tabela.
{
  documento<-read.csv(documento, sep=sepa[1], header =TRUE, na.strings =
"NA")##lê a tabela, separador posição [1], a tabela tem cabeçalhos,
considera os NA
  b<- (documento[,-1])##tira a primeira coluna de caracteres que logo
serão o fator 1.
  c<-as.matrix(b)##mudo a matriz para poder colocar os dados em uma
coluna.
  d<- matrix(c, nrow=(nrow(c)*ncol(c)), ncol=1)##crio uma matriz de uma
coluna
  d<-as.data.frame(d)##mudo para dataframe para poder adicionar colunas.
  d$f1<-rep(documento[,1], times=ncol(c))##agrego uma nova coluna com o
fator1
  d$f2<-rep(names(b), each=nrow(c))##agrego uma nova coluna com o fator2
  fn<-c()##vetor vazio para receber os nomes das colunas
  fn<-colunomes##ingresso dos nomes no vetor
  colnames(d)<-fn##coloco os nomes
  resulta=d## objeto resultado
}
if (transpor == TRUE)##indica que o usuario quer transpor a tabela
{
  documento<-read.csv(documento, sep=sepa[1],header=TRUE)##le a tabela,
sep posição [1]
  b<-t(documento)##transpoe a tabela
  c<-b[-1,]##tira a primeira coluna de caracteres que logo sera o fator 1.
  d<-matrix(c, nrow= (nrow(c)*ncol(c)), ncol=1)##crio diretamente a matriz
de uma coluna, nao precisa de mudar previamente pois o resultado da funÃ§Ã£o
t() Ã© uma matriz
  d<-as.data.frame(d)## mudo para dataframe para adicionar as colunas
  d$f1<-rep(documento[,1], each=nrow(c))##agrego a coluna com o fator1
  d$f2<-rep(rownames(c), times=ncol(c))##agrego a coluna com o fator2
  fn<-c()##vetor vazio para receber os nomes das colunas
  fn<-colunomes##ingresso os nomes das colunas
  colnames(d)<-fn##coloco os nomes
  resulta=d## objeto resultado
}
if(resumo==TRUE)##indica que o usuario deseja um resumo dos totais e
maximos da tabela (so na tela)
{
  t1=tapply(d[,1],d[,2],sum)##total por fator 1
  t2=tapply(d[,1],d[,3],sum)##total por fator 2
  m<-head(d[order(d[,1], decreasing = TRUE),], 3)## os três valores
máximos
  z<-nrow(d[d==0,])##quantos zeros têm os dados
}
```

```
nz<-(d[d!=0,])##valores mínimos diferentes de 0
min<-head(nz[order(nz[,1], decreasing = FALSE),], 3)
n<-length(d[is.na(d)])## quantos NA têm os dados
resulta=list(tabela=d, f1=t1, f2=t2, maximos=m, minimos=min, zero=z,
na=n)## objeto resultado com a tabela e a lista do resumo
}
if(histograma == TRUE)## se o argumento é verdadeiro
{
  hist(d[,1], xlab = colunomes[1], ylab = "frequência", main=
paste("Histograma de", colunomes[1]))## gráfico de histograma
}
write.table(d,file=docfinal,sep = sepa[2],row.names=FALSE)##salvo a tabela
em um arquivo, separador posição[2]
return(resulta)## mostra na tela a tabela modificada
}
```

Arquivo da função

codigo [reordenar.r](#) exemplos de input [prova.txt](#) [prova.csv](#) [prova2.csv](#)[prova2.csv](#) exemplos de output [pru1.txt](#) [pru2.txt](#)

Urls

<http://127.0.0.1:13620/library/base/html/connections.html>

<http://127.0.0.1:13620/library/utils/html/write.table.html>

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:nerina:start



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