

Natane Sibon Purgato



Bacharela e Licenciada em Ciências Biológicas (UNICAMP). Mestra em Biologia Animal (UNICAMP). Atualmente, doutoranda em Biodiversidade Animal, Instituto de Biologia, UNICAMP.

Minha tese se intitula: “LEVANTAMENTO E DIVERSIDADE DE ESPÉCIES DE SARCOPHAGIDAE E CALLIPHORIDAE (DIPTERA) DE QUATRO FRAGMENTOS DO BIOMA MATA ATLÂNTICA, SP”. Sou orientada pelo Professor Doutor Arício Xavier Linhares.

[Meus Exercícios F F F F F F F](#)

Currículo Lattes: [Natane Sibon](#)

****TRABALHO FINAL****

Proposta: **Aromaterapia e a Saboaria Artesanal**

Contextualização

A “Aromaterapia” é um ramo da fitoterapia que se utiliza de óleos 100% puros extraídos de folhas, sementes, raízes, caules, frutos e flores de diversas plantas. Esses óleos podem ser utilizados tanto para o tratamento de enfermidades físicas (como um auxílio aos tratamentos médicos convencionais), como em enfermidades emocionais (depressão, ansiedade, entre outros). Os óleos essenciais podem ser combinados de inúmeras maneiras mas, uma das mais tradicionais é a utilização destes óleos, somados a outros componentes para elaborar sabões artesanais. A técnica da saboaria artesanal é antiga e apresenta etapas elaboradas, consistindo basicamente na reação de neutralização de um ou mais óleos com hidróxido de sódio (NaOH), uma base forte. Para que o processo de neutralização seja eficaz, é necessário conhecimento do índice de saponificação de cada óleo. Esse índice é definido como a quantidade de base necessária para que ocorra a saponificação completa de um grama do óleo. Cada gordura e óleo apresenta propriedades de hidratação, limpeza e poder de espuma diferenciados. Por isto, um sabão nunca é fabricado apenas com um tipo de gordura mas, suas combinações. Dessa maneira, é preciso conhecer os índices de saponificação distintos para assim chegar na quantidade exata de NaOH que será utilizado na reação. Nessa função o usuário entraria com o tipo de problema que foi subdividido em doze grupos: Dor, Infecção, Inflamação, Problema de pele, Problema respiratório, Feridas, Problema circulatório, Depressão, Problema digestivo, Problema capilar e Problema urinário. A função retornaria uma lista com três dos óleos mais indicados para o problema que o usuário reportar. Além disso, a função tem a opção de que o usuário escolha até duas gorduras para produção de sabão artesanal, retornando seus índices de saponificação (a soma do índice dos dois óleos).

****Planejamento da função****

Entrada: aromasoap (peso1, peso2=NULL)

peso1, peso2- variável de peso. É o índice de saponificação de cada óleo.

Nesta função o usuário poderá escolher óleos essenciais de acordo com doze categorias de enfermidade. Após escolher as categorias, o usuário também poderá escolher até dois óleos para saponificar, dentre dez opções e lhe daremos os seus índices de saponificação somados (quantidade necessária de hidróxido de sódio para neutralizar um grama do óleo escolhido).

Saída: índice de Saponificação dos óleos (somados) e lista com os óleos mais adequados para o tratamento da enfermidade reportada pelo usuário

Página de Ajuda

aromasoap package:unknown R Documentation

Escolha o óleo essencial de acordo com seu problema e produza um sabonete a partir das suas escolhas.

Description:

A função aromasoap é uma função interativa, dividida em duas partes. Na primeira parte, a função apresenta uma lista dos principais óleos essenciais e seus usos mais frequentes, divididos em doze categorias. O usuário deve responder qual seu problema de saúde e a função sorteia três óleos que podem ser utilizados para auxiliar no tratamento. Após isso, esses três óleos escolhidos podem ser utilizados para confecção de um sabão artesanal. Na segunda parte da função, o usuário recebe uma lista de gorduras e escolhe as gorduras para fazer o sabonete. A função retorna a soma dos índices de saponificação e até duas gorduras/óleos escolhidos para a fabricação do sabonete.

Usage:

```
aromasoap(peso1, peso2=NULL)
```

Arguments:

peso1= peso (em gramas do primeiro óleo que será utilizado para a saponificação)

peso2 =peso (em gramas do segundo óleo que será utilizado para a saponificação)

Value:

A função retorna a quantidade de NaOH (EM GRAMAS) que deverá ser utilizada para saponificar os óleos escolhidos pelo usuário, bem como uma lista com os três óleos essenciais que podem ser utilizados para o tratamento da enfermidade reportada no início da função.

Note:

Dentro da função, insira sempre o peso dos óleos que quer saponificar (peso1, peso2) em GRAMAS.

Author(s):

Natane de C. S Purgato (natane.sibon@gmail.com)

References:

Roberto Akira: <http://www.japudo.com.br/>
Hall, F.G - A bíblia da Aromaterapia. Editora Pensamento.
Laszlo Aromaterapia & Aromatologia:<http://laszlo.ind.br/default.asp>

Examples:

```
##----Exemplo!! ----
##-- aromasoap(110)# peso (em gramas) de um óleo para saponificação
##-- aromasoap(120, 60)# peso (em gramas) para a saponificação de dois
óleos
```

===== Código da Função =====

```
aromasoap<- function(peso1, peso2=NULL) #peso2=NULL indica que caso ele
esteja ausente, a função vai entender como nulo.
{

analgésico <- c("Jasmin", "Grapefruit", "Alecrim", "Gengibre", "Canela",
"Camomila", "Eucalipto", "Erva-doce", "Hortela", "Melaleuca", "Menta",
"Limão", "Cravo", "Manjerona", "Olibano")#Cria um vetor com as opções de
óleos com a função analgésico

bactericida<- c("Alecrim", "Camomila", "Canela", "Eucalipto", "Erva-doce",
"Cravo", "Tomilho", "Hortela", "Lavanda", "Laranja", "Limão", "Melaleuca",
"Menta", "Rosa")#Cria um vetor com as opções de óleos com a função
bactericida

antiinflamatório <- c("Alecrim", "Camomila", "Hortela", "Menta",
"Melaleuca")#Cria um vetor com as opções de óleos com a função
antiinflamatório

cicatrizante<- c("Camomila", "Lavanda", "Melaleuca", "Menta")#Cria um vetor
com as opções de óleos com a função cicatrizante
```

```
descongestionante<- c("Pinho", "Tomilho", "Limao", "Canela", "Cipreste",
"Olibano", "Eucalipto", "Hortela") #Cria um vetor com as opções de óleos
com a função descongestionante

dermatologico <- c("Camomila", "Eucalipto", "Hortela", "Lavanda", "Rosa",
"Melaleuca", "Bergamota", "Capim-limao", "Cedro", "Mandarina",
"Palmarosa")#Cria um vetor com as opções de óleos com a função
dermatologico

circulatorio <- c("Geranio", "Alecrim", "Limao", "Rosa", "Cipreste" )#Cria
um vetor com as opções de óleos com a função circulatorio

antidepressivo <- c("Hortela", "Alecrim", "Lavanda", "Rosa", "Ylang-Ylang",
"Jasmin", "Laranja", "Manjerona", "Menta", "Palmarosa", "Salvia")#Cria um
vetor com as opções de óleos com a função antidepressivo

digestivo <- c("Hortela", "Alecrim", "Lavanda", "Limao", "Ylang-Ylang",
"Bergamota", "Capim-limao", "Cravo", "Erva-doce", "Laranja", "Mandarina",
"Petigrain")#Cria um vetor com as opções de óleos com a função digestivo

diuretico <- c("Geranio", "Capim-limao", "Grapefruit")#Cria um vetor com as
opções de óleos com a função diuretico

capilar <- c("Limao", "Alecrim", "Ylang-Ylang")#Cria um vetor com as opções
de óleos com a função capilar

antistress <- c("Bergamota", "Palmarosa", "Salvia")#Cria um vetor com as
opções de óleos com a função antistress

#Primeira interação com o usuário
cat(">>>>Vamos montar um tratamento para você<<<<<") #começando a interação
com uma afirmação na tela
pergunta<- readline(">>>>Qual é o seu problema? \n 1 - Dor \n 2 - Infecção
\n 3 - Inflamação \n 4 - Problema de pele \n 5 - Respiratório \n 6 - Feridas
\n 7 - Circulação \n 8 - Depressão \n 9 - Digestão \n 10 - Problema capilar
\n 11 - Problema urinário<<<<<")# cria uma pergunta interativa ao usuário

dor <- sample(analgésico, size=3, replace=F)#cria um vetor que retorna três
opções de óleos quando o usuário escolher a opção dor

infeccao <- sample(bactericida, size=3, replace=F)#cria um vetor que retorna
três opções de óleos quando o usuário escolher a opção infecção

inflamacao <- sample(antiinflamatorio, size=3, replace=F)#cria um vetor que
retorna três opções de óleos quando o usuário escolher a opção inflamação

problema_pele <- sample(dermatologico, size=3, replace=F)#cria um vetor que
retorna três opções de óleos quando o usuário escolher a opção problema de
pele
```

```
respiratorio <- sample(descongestionante, size=3, replace=F)#cria um vetor
que retorna três opções de óleos quando o usuário escolher a opção
respiratorio

feridas <- sample(cicatrizante, size=3, replace=F)#cria um vetor que retorna
três opções de óleos quando o usuário escolher a opção feridas

circulacao <- sample(circulatorio, size=3, replace=F)#cria um vetor que
retorna três opções de óleos quando o usuário escolher a opção circulação

depressao <- sample(antidepressivo, size=3, replace=F)#cria um vetor que
retorna três opções de óleos quando o usuário escolher a opção depressão

digestao <- sample(digestivo, size=3, replace=F)#cria um vetor que retorna
três opções de óleos quando o usuário escolher a opção digestão

problema_capilar<- sample(capilar, size=3, replace=F)#cria um vetor que
retorna três opções de óleos quando o usuário escolher a opção problema
capilar

problema_urinario<- sample(diuretico, size=3, replace=F)#cria um vetor que
retorna três opções de óleos quando o usuário escolher a opção problema
urinário

if(pergunta==1){indicacao=sample(analgesico, size=3, replace=F)}#caso o
usuário escolha a opção número 1, retorna os óleos do vetor analgesico
if(pergunta==2){indicacao=sample(infeccao, size=3, replace=F)}#caso o
usuário escolha a opção número 2, retorna os óleos do vetor infecção
if(pergunta==3){indicacao=sample(inflamacao, size=3, replace=F)}#caso o
usuário escolha a opção número 3, retorna os óleos do vetor inflamação
if(pergunta==4){indicacao=sample(problema_pele, size=3, replace=F)}#caso o
usuário escolha a opção número 1, retorna os óleos do vetor problema_pele
if(pergunta==5){indicacao=sample(respiratorio, size=3, replace=F)}#caso o
usuário escolha a opção número 1, retorna os óleos do vetor respiratorio
if(pergunta==6){indicacao=sample(feridas, size=3, replace=F)}#caso o usuário
escolha a opção número 1, retorna os óleos do vetor feridas
if(pergunta==7){indicacao=sample(circulacao, size=3, replace=F)}#caso o
usuário escolha a opção número 1, retorna os óleos do vetor circulação
if(pergunta==8){indicacao=sample(depressao, size=3, replace=F)}#caso o
usuário escolha a opção número 1, retorna os óleos do vetor depressão
if(pergunta==9){indicacao=sample(digestao, size=3, replace=F)}#caso o
usuário escolha a opção número 1, retorna os óleos do vetor digestão
if(pergunta==10){indicacao=sample(problema_capilar, size=3, replace=F)}#caso
o usuário escolha a opção número 1, retorna os óleos do vetor problema
capilar
if(pergunta==11){indicacao=sample(problema_urinario, size=3,
replace=F)}#caso o usuário escolha a opção número 1, retorna os óleos do
vetor problema urinario

#saboaria
```

```
gorduras <- c( "Amendoa", "Cacau", "Canola", "Coco", "Gergelim",
"Girassol", "Oliva", "Palma", "Soja") #cria vetor com as gorduras e óleos
indicados

hidroxido <- c( 0.139, 0.1287, 0.1234, 0.1836, 0.1365, 0.1362, 0.1355,
0.1408, 0.1355) #cria vetor com o índice de saponificação dos óleos

saponifica<- data.frame( gorduras, hidroxido) #cria data.frame
pergunta2<- readline(">>>>Gostaria de saber quais óleos temos? (s ou n)
Resposta:") #pergunta ao usuário se ele gostaria de ser informado dos óleos
disponíveis para saponificação

if(substr(pergunta2, 1, 1)=="s") #para que o usuário possa receber uma tabela
com os óleos disponíveis
{ print(saponifica$gorduras) #mostra as opções de óleos
}

sabao <- readline(">>>>Qual(is) gordura(s) deseja adicionar em seu sabonete?
Vamos dar seus indices de saponificação somados \n (Digite até duas,sem
vírgulas. ex: Amendoa Cacau.") #cria uma pergunta para que o usuário
escolhas quais gorduras deseja saponificar

#quando apenas um óleo for requisitado

if(sabao=="Amendoa") {saponificacao<-(peso1*hidroxido[1])} #chama o valor da
coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
usuário.
if(sabao=="Cacau") {saponificacao<-(peso1*hidroxido[2])}#chama o valor da
coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
usuário.
if(sabao=="Canola") {saponificacao<-(peso1*hidroxido[3])}#chama o valor da
coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
usuário.
if(sabao=="Coco") {saponificacao<-(peso1*hidroxido[4])}#chama o valor da
coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
usuário.
if(sabao=="Gergelim") {saponificacao<-(peso1*hidroxido[5])}#chama o valor da
coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
usuário.
if(sabao=="Girassol") {saponificacao<-(peso1*hidroxido [6])}#chama o valor
da coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
usuário.
if(sabao=="Oliva") {saponificacao<-(peso1*hidroxido[7])}#chama o valor da
coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
usuário.
if(sabao=="Palma") {saponificacao<-(peso1*hidroxido[8])}#chama o valor da
coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
usuário.
if(sabao=="Soja") {saponificacao<-(peso1*hidroxido[9])}#chama o valor da
coluna hidróxido e multiplica pelo peso (em gramas) especificado pelo
```

usuário.

```
#para dois tipos de óleos
```

```
a=strsplit(sabao, split=" ")[[1]] ## Linha que separa a resposta do usuário em um vetor com dois objetos
```

```
saponificacao=sum((saponifica[gorduras==a[1],2]*peso1),  
(saponifica[gorduras==a[2],2]*peso2)) #chama o valor da coluna de hidróxido (saponifica) que é referente à resposta que o usuário escreveu
```

```
indicacao2=do.call(paste, c(as.list(indicacao), sep = ", ")) #Transforma as respostas eu um só vetor, separando as palavras por vírgulas
```

```
resposta=list(saponificacao, paste("Para seu tratamento os óleos  
medicinais",indicacao2, "são indicados.## Retorna a lista de opções que o usuário requisitou.
```

```
return(resposta)
```

```
### Função retorna o valor de saponificação de acordo com as gorduras escolhidas e o peso e quais os óleos indicados para o problema da pessoa.  
}
```

Arquivos

[aromasoap_function_final.r](#)

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:natane.sibon:start&rev=1597223093

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