

# TRABALHO FINAL

## Zoobodymass()

```
Zoobodymass=function(x, level.abundance = "", level.length = "", level.order = "", graphic = TRUE) #função com os argumentos.  
#argumentos: x = dados de natureza data.frame; level.abund = nome da coluna com os valores de abundância; level.a = nome da coluna com os valores de comprimento; order = nome da coluna com a ordem zooplancônica (cladocera ou copepoda); graphic = plot da relação entre abundância e o comprimento.  
{  
  if(class(x)!="data.frame") #condição para entrada de dados de natureza data.frame. ("Se a natureza do objeto for diferente de um data.frame")  
  {  
    stop("OPS... o objeto não é um data.frame!") #caso não seja um data.frame, a função não irá rodar e irá aparecer essa mensagem no console.  
  }  
  if(any(is.na (x[,c(level.abundance, level.length, level.order)])))  
  #condição para remover as linhas com dados ausentes (NA) das colunas "abundance", "length" e "order". ("Se existir alguma linha das colunas de abundância, comprimento e ordem com valores faltantes")  
  {  
    x=na.omit(x) #removendo as linhas com NA das colunas "abundance", "length" e "order".  
    print("Foi removido linhas com NA.") #para aparecer uma mensagem, caso tenha encontrado linhas com dados ausentes e tenha removido.  
  }  
  if(any(x[, level.abundance] ==0)) #condição para remover as linhas com zero da coluna "abundance". ("Se existir alguma linha da coluna de abundância com valores iguais a zero")  
  {  
    x=x[x[, level.abundance] !=0, ] #removendo as linhas com zero da coluna "abundance".  
    print("Foi removido linhas com valores de abundância igual a zero.")  
    #para aparecer uma mensagem, caso tenha encontrado linhas com valores igual a zero e tenha removido.  
  }  
  if(any(x[, level.length] ==0)) #condição para remover as linhas com zero da coluna "length". ("Se existir alguma linha da coluna de comprimento com valores iguais a zero")  
  {  
    x=x[x[, level.length] !=0, ]#removendo as linhas com zero da coluna "length".  
    print("Foi removido linhas com valores de comprimento igual a zero.")  
    #para aparecer uma mensagem, caso tenha encontrado linhas com zero e tenha removido.  
  }  
  if(any(!x[, level.order] %in% c("copepoda", "cladocera"))) #condição para
```

```
aceitar apenas se a coluna "order" tiver os gêneros "copepoda" e
"cladocera". ("Se existir alguma linha da coluna de ordem sem a informação
cladocera ou copepoda)
{
  stop("OPS...você não definiu os gêneros cladocera e/ou copepoda
corretamente ou então não possui essa informação em sua planilha!") #caso o
usuário não tenha definido os gêneros corretamente de todas as linhas do
data.frame, a função não irá rodar e irá aparecer essa mensagem no console.
}
ps=vector() #criando um objeto para guardar os valores de peso seco.
l=vector() #criando um objeto para guardar os valores de comprimento em
milímetro.
for(i in 1:nrow(x)) #aplicando o loop para que ele possa olhar linha por
linha e fazer as operações matemáticas de acordo com a ordem.
{
  if(x[i, level.order]=="cladocera") #condiçao caso a linha da coluna
"order" for "cladocera".
  {
    lna=1.9526 #definindo o valor de "lna" para "cladocera"
    b=2.399 #definindo o valor de "b" para "cladocera"
    l[i]=x[i, level.length]*0.001 #calculando os valores de comprimento em
milímetro para guardar no objeto "l".
    lnw=lna+b*log(l[i]) #caclculando lnw a partir dos valores de "lna",
"b" e "l".
    ps[i]=exp(lnw) #calculando os valores de peso seco para guardar no
objeto "ps".
  }
  if(x[i, level.order]=="copepoda") #condiçao caso a linha da coluna
"order" for "copepoda".
  {
    lna=1.7512 #definindo o valor de "lna" para "copepoda"
    b=2.653 #definindo o valor de "b" para "cladocera"
    l[i]=x[i, level.length]*0.001 #criando os valores de comprimento em
milímetro para guardar no objeto "l".
    lnw=lna+b*log(l[i]) #caclculando lnw a partir dos valores de "lna",
"b" e "l".
    ps[i]=exp(lnw) #calculando os valores de peso seco para guardar no
objeto "ps".
  }
}
sumario.l = summary(l) #fazendo um summary da variável comprimento (mm) e
guardando esses valores no objeto chamado "sumario.l".
sumario.ps = summary(ps) ##fazendo um summary da variável peso seco e
guardando esses valores no objeto chamado "sumario.ps".
body.mass = ps*x[, level.abundance] #calculando a biomassa a partir do
valor de peso seco e guardando esses valores no objeto chamado "body.mass".
sumario.body.mass = summary(body.mass) ##fazendo um summary da variável
biomassa e guardando esses valores no objeto chamado "sumario.body.mass".
graphic =
{
```

```
    par(mfrow=c(1, 2), bty="l") #alterando o layout do plot para mostrar
os dois gráficos (plot da relação biomassa~comprimento por "order" e boxplot
da biomassa de cladocera e copepoda).
    plot(body.mass~l, pch=20, ylab="Biomassa", xlab="Comprimento(mm)")
#fazendo um plot da biomassa e do comprimento (mm) por "order" e guardando
esse plot no objeto chamado "graphic".
    boxplot(body.mass~x[, level.order], col=c("orange", "purple"),
ylab="Biomassa", xlab="Comprimento(mm)") #fazendo um boxplot da biomassa por
"order" e guardando esse plot no objeto chamado "graphic".
    par(mfrow=c(1,1)) #retornando a configuração do layout.
  }
  resultado=list(l, sumario.l, ps, sumario.ps, body.mass, sumario.body.mass,
graphic) #criando um objeto da natureza lista chamado "resultado" com os
valores das variáveis comprimento, peso seco e biomassa, o summary dessas
variáveis e o plot da relação abundância~comprimento (mm).
  names(resultado)=c("comprimento (mm)", "sumário do comprimento", "peso
seco", "sumário do peso seco", "biomassa", "sumário da biomassa", "plot")
#nomeando cada objeto dentro do objeto resultado.
  return(resultado) #retornando o resultado da função.
}
```

### Arquivo da função Zoobodymass

[zoobodymass.r](#)

### Help da função Zoobodymass

Zoobodymass  
Documentation

R

Cálcula a biomassa de microcrustáceos zooplânctônicos

Description:

A função gera dados de biomassa em  $\mu\text{g.mL}$  a partir de dados de entrada de abundância e tamanho corporal. A biomassa é calculada de acordo com as equações de peso seco e comprimento de acordo com Bottrell et al. (1976) para cladocera e Copepoda. A função também gera um sumário das principais variáveis de interesse (comprimento em milímetros, peso seco e biomassa) e oferece a opção de gerar gráficos.

Usage:

```
Zoobodymass(x, level.abundance = "", level.length = "", level.order = "",
graphic = TRUE)
```

Arguments:

x: dataframe

level.abundance: define o nome da coluna do dataframe que contém os valores

de abundância

`level.length`: define o nome da coluna do dataframe que contém os valores de comprimento

`level.order`: define o nome da coluna que contém a ordem zooplanctônica (cladocera e/ou copepoda).

`graphic`: é um argumento opcional para plotar um gráfico da relação entre a biomassa e o comprimento dos organismos e um boxplot da biomassa por "`level.order`" ("`cladocera`", "`copepoda`").

Details:

- Os dados de entrada obrigatoriamente precisam estar na forma de `data.frame`. Caso o usuário entre com dados de outra natureza, a função irá acusar uma mensagem de erro e não irá rodar.

- Dentro do argumento "`level.order`", o usuário terá que apresentar a informação da seguinte forma: para os cladóceros o nome da ordem é "`cladocera`" e para os copepodes o nome da ordem é "`copepoda`". Caso o usuário não forneça essa informação a função irá acusar uma mensagem de erro e não irá rodar.

- A função oferece a opção de gerar dois gráficos: um plot da relação entre a biomassa ( $\mu\text{g.mL}$ ) e do comprimento (mm) dos microcrustáceos e um boxplot da biomassa para cada ordem, cladocera e copepoda.

Value:

A função irá retornar um objeto com natureza de lista, com os seguintes componentes:

`comprimento`: um vetor com os valores de tamanho corporal dos organismos em milímetros

`sumário do comprimento`: um sumário dos valores de tamanho corporal dos organismos

`peso seco`: um vetor com os valores de peso seco dos organismos em  $\mu\text{g.mL}$

`sumário do peso seco`: um sumário dos valores de peso seco dos organismos.

`biomassa`: um vetor com os valores de biomassa dos organismos em  $\mu\text{g.mL}$

`sumário da biomassa`: um sumário dos valores de biomassa dos organismos

`plot`: um plot da relação entre a biomassa e o tamanho corporal dos microcrustáceos e um boxplot da biomassa dos cladóceros e copepodes.

Warning:

As mensagens de erro que aparecem na função são:

```
{Warning }{"OPS... o objeto não é um data.frame!"}
```

```
{Warning }{"Foi removido linhas com NA."}
```

```
{Warning }{"Foi removido linhas com valores de abundância igual a zero."}
```

```
{Warning }{"Foi removido linhas com valores de comprimento igual a zero."}
```

```
{Warning }{"OPS...você não definiu os gêneros cladocera e/ou copepoda"}
```

```
corretamente ou então não possui essa informação em sua planilha!"}  
Author(s):
```

Lorena Pinheiro Silva

#### References:

Blettler, M. C. M., Bonecker, C. C. Avaliação da biomassa de microcrustáceos em ambientes aquáticos continentais. Caracas: Interciencia. v. 21, n. 8, p. 591-597, 2006.

Bottrell, H. H., Duncan, A., Gliwics, Z. M., Grygierek, E., Herzig, A., Hillbricht-Ilkowska, A., Kurasawa, H., Larsson, P., Weglenska, T. A review of some problems in zooplankton production studies. Oslo: Nor. J. Zool. v. 24, p. 419-456, 1976.

#### Examples:

```
#Criando um vetor com os dados de abundância  
abundance= c(0, 153256, 65280, 36960, 31664, 22772, 35456, 430, 0, 32404,  
13589, 11196, 211152, 1169, 11814, 17344, 12596, 13080, 11908, 17696, 13180,  
1043, 11237, 15180, 11960, 21720, 21712, 15576, 32080, 21664, 21632, 21376,  
1543, 32186, 11543, 133357)
```

```
#Criando um vetor com os dados de comprimento  
length= c(234, 133, 157, 149, 138, 230, 195, 409, 234, 233, 207, 201, 145,  
433, 207, 249, 238, 230, 195, 209, 213, 546, 133, 233, 207, 249, 238, 230,  
195, 164, 234, 123, 543, 134, 321, 133)
```

```
#Criando as cordões "cladocera" e "copepoda"  
order= sort(rep(c("cladocera", "copepoda"), 18))
```

```
#Criando um data.frame com todas as variáveis  
teste=data.frame(abundance, length, order)
```

```
# Aplicando a função  
Zoobodymass(teste, level.abundance = "abundance", level.length = "length",  
level.order="order", graphic = TRUE)
```

#### Arquivo do help da função Zoobodymass

[zoobodymass.txt](#)

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

Permanent link:  
[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2017:alunos:trabalho\\_final:lorena.biouni:syntax](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:lorena.biouni:syntax)

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