

# Gustavo de Oliveira



Mestrando em Ecologia de Sistemas Aquáticos e Terrestres - IB/USP

No meu trabalho de mestrado investigo a importância relativa da estrutura e dinâmica da paisagem e das atividades e atitudes da população humana sobre a distribuição de mamíferos de maior porte em uma paisagem rural do Planalto Atlântico de São Paulo.

[exec](#)

## Trabalho Final

### Plano A

Em estudos que utilizam armadilhas fotográficas para registrar mamíferos de maior porte é necessário determinar o intervalo de tempo mais adequado para a quantificação de registros fotográficos independentes, e isto pode ser feito através da identificação do intervalo de tempo em que ocorre uma forte redução no número de fotos seqüenciais de uma mesma espécie em uma mesma câmera, para isto é necessário quantificar a distribuição de freqüências dos intervalos de tempos decorridos entre fotografias seqüenciais da mesma espécie na mesma câmera. Em meu trabalho final pretendo fazer uma função que crie: histogramas da distribuição de freqüências dos intervalos entre fotos de uma mesma espécie, e gráficos da proporção entre o número de registros independentes e o número total de fotos de cada espécie para diferentes intervalos de corte que poderiam ser utilizados para definir registros independentes (15 e 30 min e 1 hora). Esta função será muito útil, pois irá possibilitar uma exploração gráfica dos dados e irá poupar muito tempo a outras pessoas que venham a trabalhar com armadilhas fotográficas e que queiram estabelecer um intervalo de corte entre duas fotografias seqüenciais de uma mesma espécie em uma mesma câmera como registros independentes.

### Comentários PI

A idéia é legal e viável como trabalho final. Como conversamos, o crítico é definir o critério de identificação de registros independentes. Correlação temporal é uma possibilidade, mas deve haver outras tb. Tenha este critério bem claro antes de seguir adiante.

## Página de Ajuda

freq

package:unknown

R Documentation

Identifica a unidade de tempo para considerar duas fotografias seqüenciais da mesma espécie como registros independentes.

## Description:

Quantifica a distribuição de freqüências dos intervalos de tempos decorridos entre fotografias seqüenciais de uma mesma espécie no mesmo fragmento, data e câmera em classes de 10 minutos. Gera uma matriz com o número total e a proporção de registros independentes para cada intervalo de corte (0, 15, 30, 45, 60 e 90 min). Constrói um histograma de distribuição de freqüências dos intervalos entre fotos em classes de 10 minutos, e um gráfico para o número total e da proporção entre o número de registros independentes para os diferentes intervalos de corte (0, 15, 30, 45, 60, e 90min). Gera valores de correlação das freqüências dos registros fotográficos entre as classes de intervalos de 10 minutos. É possível fazer exploração gráfica e comparar com os valores de correlação para definir o intervalo de corte para considerar duas fotografias seqüenciais da mesma espécie como registros independentes.

## Usage:

`freq(x)`

## Arguments:

x: data.frame onde a primeira coluna seja Fragmento, a segunda coluna Data , a terceira coluna Camera e quarta Horário.

## Details:

A tabela que entrará na função deve estar no formato (.csv) e o nome da primeira coluna deve ser Fragmento, da segunda coluna Data , da terceira coluna Camera e da quarta coluna Horário. Antes de iniciar a função a coluna Data deve ser transformada para a classe Date através da função `as.Date` e a coluna Horário deve ser transformada para a classe `times` através da função `times` que está dentro do pacote `chron`.

Observação esta tabela de entrada não pode conter `Nas`.

## Value:

Tabela1 = Retorna uma matriz com os valores totais e a proporção de fotos para os intervalos de corte de (0, 15, 30, 45, 60 e 90 min).

Tabela2 = Retorna um array com os valores das distribuições das freqüências dos intervalos de tempos de 10 minutos até 300 minutos entre fotografias seqüenciais separadas em classes de 10 minutos .

CorSpearman = Retorna um vetor com os valores das correlações das distribuições das freqüências dos intervalos de tempos de 10 minutos até 300 minutos, onde pode ser observada a posição onde o valor de p não é mais

significativo. Essa posição é correspondente a posição do array, assim é possível saber em qual classe deve ser feito o corte.

Retorna um histograma de frequência dos intervalos de tempos em classes de 10 minutos.

Retorna um gráfico para o número total e outro para a proporção entre o número de registros independentes para os diferentes intervalos de corte (0, 15, 30, 45, 60, e 90min)

Author(s):

Gustavo de Oliveira  
gunespe@yahoo.com.br

See Also:

Examples:

```
canis=read.table("canis.familiaris.csv",header=T,sep=" ",as.is=T)
canis
canis$Data<-as.Date(canis$Data, format="%d/%m/%Y")      ###Transforme a
coluna Data para a classe Date
canis$Data
library(chron)                                          ###baixo o pacote
chron
canis$Horário<-times(canis$Horário)                    ###Transformar sua
coluna Horário para a classe times
canis$Horário
str(canis)
head(canis)
summary(canis)                                         ###Confira a sua
tabela

freq(canis)
```

###Tabela para usar no exemplo. Não consegui fazer com que esta tabela ficasse disponível no exemplo da página de ajuda [canis.familiaris.csv](#)

## Código da Função

```
#####Antes de rodar a função faça essas transformações#####
x=read.table("x.csv",header=T,sep=" ",as.is=T)
x
x$Data<-as.Date(x$Data, format="%d/%m/%Y")
x$Data
```

```
library(chron)
x$Horário<-times(x$Horário)
x$Horário
str(x)
head(x)
summary(x)

freq=function(x)
{
  dif=rep(NA,dim(x)[1])
  for(i in 1:(dim(x)[1]-1))
  {
    if(x$Fragmento[i]==x$Fragmento[i+1]& x$Data[i]== x$Data[i+1]&
x$Camera[i]== x$Camera[i+1])
      {
        dif[i+1]= (x$Horário[i+1]- x$Horário[i])*24*60
###transformando a fração de dias em minutos
      }
  }
  b=sum(table(dif))
  c=sum(table(dif[dif>=15]))
  d=sum(table(dif[dif>=30]))
  e=sum(table(dif[dif>=45]))
  f=sum(table(dif[dif>=60]))
  k=sum(table(dif[dif>=90]))
  pb=100
  pc=sum(table(dif[dif>=15]))/b*100
  pd=sum(table(dif[dif>=30]))/b*100
  pe=sum(table(dif[dif>=45]))/b*100
  pf=sum(table(dif[dif>=60]))/b*100
  pk=sum(table(dif[dif>=90]))/b*100
  g=c(b,c,d,e,f,k)
  j=as.matrix(g)
  j1=cbind(j,c(pb,pc,pd,pe,pf,pk)) #####gerando a matriz com o número
total e a proporção de registros independentes em cada classe
  rownames(j1)=c("0", "15min", "30min", "45min", "60min", "90min")
  colnames(j1)=c("Número de Fotos", "Proporção do número de fotos")
  tempo=c(0,15,30,45,60,90)
  par(mfrow=c(2,2))
  hist(dif,main="Nome da Espécie",xlab="Intervalo entre fotos(min)",
ylab="Número de fotos",xlim=c(0,360),
ylim=c(0,200),xaxp=c(0,360,36),yaxp=c(0,200,20),breaks=100,col=2)
###gerando o histograma
  plot(j1[,1]~tempo, xlab="Tempo(min)",ylab="Número de
fotos",bty="l",pch=16, main="Nome da Espécie") #####gerando o gráfico do
número total de registros independentes em cada classes
  lines(tempo, j1[,1],col=2,lwd=2)
  plot(j1[,2]~tempo, xlab="Tempo(min)",ylab="Proporção do número de
fotos",ylim=c(0,120),bty="l",pch=16, main="Nome da Espécie")
#####gerando o gráfico da proporção dos registros independentes em cada
```

```

classe
  lines(tempo, j1[,2],col=2,lwd=2)
  z=table(round(dif,0))
  z.df= as.data.frame(z,stringsAsFactors=F)
  z.df=z.df[1:which(z.df$Var1==300), ]
  m = data.frame(c(1:300), rep(0, 300))
  c1=(merge(m, z.df, by.x="c.1.300.", by.y="Var1", all=T))
  c2=c1[order(c1$c.1.300.), ]
  c2$Freq[is.na(c2$Freq)==T]=0
  intervalo = rep(1:30,each =10)
  c3=data.frame(c2, intervalo)
  frio = tapply(c3$Freq, c3$intervalo, sum)
  c4 = as.data.frame(frio)
  c5 = c4$frio
  Tabela1=c5      #####gerando o array com as frequências dos intervalos de
tempos de 10minutos até 300 minutos entre fotografias.
  names(Tabela1)=seq(10,300,10)
  c6=rep(NA,dim(c5))
  Tabela2=j1
  for(i in 1:(dim(c5)-3))
  {
    c6[i]=cor.test(c5[(i+1):length(c5)],c5[1:(length(c5)-
i)],method="spearm")$p.value   ###gerando os valores de p para a correlação
das distribuições das frequências dos intervalos de tempos de 10minutos até
300 minutos
  }
  CorSpearm=c6
  return(Tabela1,Tabela2,CorSpearm)
}

```

## Teste da Função

[nasua.csv](#)

```

nasua=read.table("nasua.csv",header=T,sep=" ",as.is=T)
nasua
nasua$Data<-as.Date(nasua$Data, format="%d/%m/%Y")
nasua$Data
library(chron)
nasua$Horário<-times(nasua$Horário)
nasua$Horário
str(nasua)
head(nasua)
summary(nasua)

freq(nasua)

```

## Arquivo da Função

[scripttrabalhofinal.r](#)

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

Permanent link:  
[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2010:alunos:trabalho\\_final:gunespe:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2010:alunos:trabalho_final:gunespe:start)

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

