

# Enrico Frigeri



Mestrando em Zoologia pela Universidade de São Paulo, Departamento de Zoologia. Trabalho com a invasão de cachorros domésticos em paisagens da Mata Atlântica brasileira. Minha tese possui o título: "INVASÃO DE REMANESCENTES FLORESTAIS POR CACHORROS DOMÉSTICOS (Canis lupus familiaris) EM PAISAGENS RURAIS DA MATA ATLÂNTICA", e orientação de Renata Pardini.

nome	projeto	email
Enrico Frigeri	Invasão de cachorros domésticos na Mata Atlântica	e_frigeri@hotmail.com

## Exercícios

[exec](#)

## Trabalho Final

### Proposta A - Não Realizada

A correlação entre duas ou mais variáveis frequentemente é expressa por uma matriz de correlação. Tal matriz evidencia a correlação entre pares de variáveis, porém esta forma de apresentação dificulta que encontremos associações causadas por algum fator comum a muitas variáveis. A análise de componentes principais (PCA) é uma das alternativas para melhor explorarmos as correlações entre variáveis e permite detectarmos os principais fatores causadores das correlações.

Em meu projeto de mestrado, possuo diversas variáveis da paisagem de estudo (ex: quantidade de matas ao redor do ponto amostrado, distância do ponto de amostragem à estrada, e etc). Uma das primeiras análises que realizarei será verificar se tais variáveis são correlacionadas. Por isso pretendo fazer uma função que me retorne: 1) uma matriz de correlação entre as variáveis; 2) um PCA com todas as variáveis estudadas; 3) uma matriz de espalhogramas. Desta forma poderei comparar os diferentes resultados obtidos por estes diferentes meios, permitindo que eu explore da melhor forma possível as correlações entre as variáveis de meu estudo. Esta função não precisa ser utilizada apenas com meus dados, sendo bastante genérica, visto que retornará os resultados citados acima, para qualquer uma tabela que tenha o formato especificado. Por isso poderá ser utilizada por qualquer pessoa que queira testar a correlação entre variáveis de uma maneira mais eficiente.

Além do descrito acima, minha função ainda retornará gráficos exploratórios de cada uma das variáveis: boxplot, histogramas, densidade de kernel. Estes gráficos permitirão que eu realize uma boa análise exploratória dos dados, antes mesmo de eu fazer as correlações.

### Comentários

A função é totalmente factível, mas todas essas funções estão implementadas de uma maneira ou outra nos pacotes básicos e até no vegan. Eu acho que você poderia pensar em um "q" a mais para

ela: visto que você vai mexer com PCAs, eu sugeriria implementar testes sobre os PCAs. Sugestões plausíveis são testar a significancia dos autovalores dos PCs contra uma distribuição do tipo brokenstick, ou avaliar por permutações, bootstrap ou jackknife.

— *Fabio de A. Machado* 2011/04/06 17:22

Paulo: Concordo com o Fabio! Uma função começa a ter mais graça se dentro dela vc cria obejtos intermediários que são recebidos por outros comandos. A sua proposta pode ser implementada como uma concatenação de comandos independentes. Gosto da ideia de um teste dos eixos contra algum modelo nulo. Um modelo nulo simples é que não há correlação entre as variáveis, o que vc pode simular embaralhando os valores de cada variável.

## Proposta B - Realizada

Outro capitulo de minha tese irá estimar o tamanho populacional de cachorros domésticos dentro de áreas de vegetação nativa. Os cachorros foram identificados e individualizados através de registros fotográficos obtidos por armadilhas fotográficas. Após a individualização montou-se uma tabela com o histórico de captura dos animais. Esta tabela é comum em estudos de captura e recaptura de animais. Conhecidas como tabelas de histórico de captura, cada linha corresponde a um indivíduo e cada coluna corresponde a um espaço de tempo, pré determinado pelo pesquisador. O numero 1 representa que um determinado indivíduo foi capturado em determinado tempo, já o número zero representa que o individuo não foi capturado naquele periodo.

Uma fonte de confusão em estudos de captura e recaptura são os individuos que mesmo presentes na área de estudo, durante uma sessao de captura, não sao capturados. Por não serem capturados estes individuos são consideraos ausentes. Existem diversas técnicas para minimizar tal confusão.

O **MNKA** é um dos estimadores do tamanho populacional que tentam corrigir a confusão citada acima. A premissa deste estimador é que a população seja fechada (ou seja, nao ocorra nem entrada [nascimento e imigração] e nem saída [morte e emigração] de animais durante o período de estudo). Tal estimador assume que se um individuo nao foi capturado na sessao de captura  $t$ , mas foi capturado na sessao de captura  $t-1$  e na sessao de captura  $t+1$ , estes indivíduo estava presente e apenas não foi capturado no tempo  $t$ . O que, sendo a população realmente fechada, é bastante plausível... Apesar do **MNKA** receber criticas na literatura, visto que não considera diversos fatores, possui a vantagem de ser extremamente intuitivo, facilmente aplicado e fornecer uma visão inicial interessante dos dados.

Desta forma criarei uma função que calcula o **MNKA** a partir de uma tabela de historico de captura. Além disto, esta função resumiria o historico de captura de uma sessão em uma unica coluna. Esta coluna resumo teria 1 para indicar que determinado individuo foi capturado ao menos uma vez durante a sessao de captura e 0 para indicar se não houve nenhuma captura do indivíduo. A importância de uma coluna resumo esta no fato que ele pode ser unida à outras colunas resumo (provenientes de outras sessoes de captura) formando assim uma tabela de historico de captura de todas as sessões. Nesta noval tabela também poderei aplicar o estimador **MNKA** com o intuito de obter o tamanho minimo da população ao longo de todo o meu estudo. Tal função poderá ser utilizada por qualquer pessoa que tenha uma tabela de historico de captura no formato de entrada da função

## Comentários

Parece uma boa proposta. Não entendi exatamente como é computado o MNKA, e não consegui avaliar se ele já está implementado no R. De qualquer forma, me parece bem geral e aplicável.

— [Fabio de A. Machado](#) 2011/04/06 17:33

## Comentários Ale GERAL

Acho que ambas as propostas são factíveis e grandes desafios. A principio gosto mais do Plano B, pois implementa uma estimativa da população. Acho que não importa se há ou não a função para o calculo do MNKA. Isso é irrelevante desde que vc. faça seu algoritmo. Na primeira parte, poderia pensar em uma função para fazer o PCA... um desafio bom.. um PCA na unha feito por vc...Só a construção dos gráficos, concordo com o Fabio, pode ser meio banal!

# Função Final

## Comentários sobre a função

Certo. Concorde com os comentários acima. Por isso, fiz a função B.

## Código da Função

```
calc.mnka = function(dados,contagem=TRUE) #a função tem dois argumentos, o
# dado, e contagem. Se contagem=true, ele #tambem calcula a contagem de
# individuos por sessao. Se for false, nao calcula
{
  dados$ind=factor(dados$ind) #transformando a variavel individuo em um fator
  dias=dim(dados)[2] #obtendo o numero de colunas
  dados$soma=apply(dados[,3:dias],1,sum,na.rm=TRUE) #somando os registros de
  # cada dia,removendo os NA
  dados$soma[dados$soma>0]=1 #valores maiores que 1 ficam sendo 1
  sessao = unique(dados[,1]) #obtendo o numero de sessao do estudo
  numerosessao=length(sessao)
  individuos= unique(dados[,2]) #obtendo o numero de individuos do estudo
  numerosind=length(individuos)
  resultados=matrix(NA,nrow=numerosind,ncol=length(sessao)+1) #criando uma
  # matriz para colocar os resultados
  resultados[,1]=unique(dados[,2]) #colocando na primeira coluna da matriz os
  # individuos
  colnames(resultados)= c("individuos",paste("sessão",1:numerosessao,sep =
  "")) #nomeando as colunas
  rownames(resultados)= c(rep("",each=numerosind))
  for(i in 1:numerosessao) ##ciclo para obter apenas a coluna com a soma de
```

```
cada uma das sessoes
{m=i+1
abc=dados[dados$sessao==i,]
resultados[,m]=abc$soma
}
resultadosdf=data.frame(resultados) #criando um data frame com o resultado
write.csv(resultados, "Matriz com o total de captura por sessao.csv")
write.csv(resultadosdf, "Dataframe com o total de captura por sessao.csv")
if(contagem==TRUE) #se quero calcular a contagem a função entra neste
ciclo
{
colunassessao=numerosessao+1
cont=apply(resultadosdf[,2:colunassessao],2,sum) #calculando o
indivíduos capturados por sessao
x11()
par(mfrow=c(1,2),family="serif", mar=c(5,5,3,2),cex=1.2)
plot(c(1:numerosessao), cont, bty="l", tcl=0.2, main="Contagem do
número de \n indivíduos por sessão", ylab="Número de indivíduos
capturados",xlab="Sessões de captura",pch=19,xlim=c(1,numerosessao),
xaxt="n",col="red", ylim=c(min(cont)-1,max(cont)+2)) #grafico com os
indivíduos capturados por sessao
axis(1, 1:numerosessao, c(1:numerosessao),tcl=0.2) #eixo do grafico
text(c(1:numerosessao), cont, labels=cont, pos=1,
offset=-1,cex=0.9)#adicionando os numeros em cima #dos pontos
resultados spp=t(resultados) #criando uma matriz transposta da
matriz dos resultados
resultados spp=resultados spp[-1,] #removendo a linha dos indivíduos e
ficando só com as sessoes
curvadeespecies=specaccum(resultados spp, method = "collector")
#calculando a curva de acúmulo de #especies
plot(curvadeespecies,bty="l", tcl=0.2, main="Curva de acúmulo de \n
indivíduos por sessão",ylim=c(0,numerosind+2), ylab="Número acumulado de
indivíduos",xlab="Sessões de captura",pch=19,xlim=c(1,numerosessao), yaxt="n",
col="blue", lwd=3,ci.type="poly", ci.lty=0, ci.col="lightblue") #plotando a
curva
axis(1, 1:numerosessao, c(1:numerosessao),tcl=0.2) #mechendo nos
eixos
resultadosmatriz=resultados[,,-1] #criando outro objeto para
calcular o mnka
resultadosmnka=matrix(NA,nrow=numerosind,ncol=numerosessao,dimnames=list(c(1
:numerosind),paste("sessão",1:numerosessao,sep = ""))) #criando uma matriz
para colocar os resultados do mnka
for(f in 1:numerosind) #ciclo de calculo do MNKA
{
for (n in 1:numerosessao)
{
if
((resultadosmatriz[f,n]==0 &
(sum(resultadosmatriz[f,(1:n)])==0))==TRUE) #se na matriz inicial é 0 e nao
tem nada antes , na nova matriz fica sendo 0
```

```

        {
            resultadosmnka[f,n]=0
        }
        if
            ((resultadosmatriz[f,n]==0 &
(sum(resultadosmatriz[f,(n:numerosessao)]))==0))==TRUE) #se na matriz inicial
é 0 e nao tem nada depois , na nova matriz fica sendo 0
        {
            resultadosmnka[f,n]=0
        }
        if ((resultadosmatriz[f,n]==1)==TRUE) #se na matriz
inicial é 1, na #nova matriz fica sendo 1
        {
            resultadosmnka[f,n]=1
        }
        if
            ((resultadosmatriz[f,n]==0 &
(sum(resultadosmatriz[f,(1:n)]))>=1) &
(sum(resultadosmatriz[f,(n:numerosessao)]))>=1))==TRUE) #se na matriz
inicial é 0 e tem algum 1 antes e depois , na nova #matriz fica sendo 1
        {
            resultadosmnka[f,n]=1
        }
    }
}

resultadosmnkadf=data.frame(resultadosmnka) #criando o data frame
com os resultados
resultadosmnkadf$ind=unique(dados[,2]) #acrescentando a coluna de
individuos no data frame
resultadosfinal=apply(resultadosmnka,2,sum) #somando o mnka por
sessao
write.csv(resultadosmnkadf, "Data.frame modificado através do método
do MNKA.csv")
resultadosfim=list("Número de individuos capturados por
sessão"=cont, "MNKA por sessão"=resultadosfinal)
cat("\n \t Foi salvo em seu diretório de trabalho os seguintes
arquivos: \n \t \t Matriz com o total de captura por sessao.csv \n \t \t
Dataframe com o total de captura por sessao \n \t \t Data.frame modificado
através do método do MNKA.csv \n\n\n")
return(resultadosfim)
par(mfrow=c(1,1))
}

if(contagem==FALSE) #se não quero calcular a contagem a função entra
neste ciclo
{
x11()
par(mfrow=c(1,1),mar=c(5,5,3,2),family="serif",cex=1.2 )
resultados spp=t(resultados)
resultados spp=resultados spp[-1,]
curvadeespecies=specaccum(resultados spp, method = "collector")
plot(curvadeespecies,bty="l",ylim=c(0,numerosind+2), tcl=0.2,main="Curva

```

```
de acúmulo de \n indivíduos por sessão", ylab="Número acumulado de
indivíduos",xlab="Sessões de captura",pch=19,xlim=c(1,numerosessao),
xaxt="n", col="blue", lwd=3,ci.type="poly", ci.lty=0, ci.col="lightblue")
  axis(1, 1:numerosessao, c(1:numerosessao),tcl=0.2)
  resultadosmatriz=resultados[, -1]
resultadosmnka=matrix(NA,nrow=numerosind,ncol=numerosessao,dimnames=list(c(1
:numerosind),paste("sessão",1:numerosessao,sep = ""))) #criando uma matriz
para colocar os resultados do mnka
  for(f in 1:numerosind)
  {
    for (n in 1:numerosessao)
    {
      if
      ((resultadosmatriz[f,n]==0 &
(sum(resultadosmatriz[f,(1:n)])==0))==TRUE)
      {
        resultadosmnka[f,n]=0
      }
      if
      ((resultadosmatriz[f,n]==0 &
(sum(resultadosmatriz[f,(n:numerosessao)])==0))==TRUE)
      {
        resultadosmnka[f,n]=0
      }
      if ((resultadosmatriz[f,n]==1)==TRUE)
      {
        resultadosmnka[f,n]=1
      }
      if
      ((resultadosmatriz[f,n]==0 &
(sum(resultadosmatriz[f,(1:n)])>=1) &
(sum(resultadosmatriz[f,(n:numerosessao)])>=1))==TRUE)
      {
        resultadosmnka[f,n]=1
      }
    }
  }
  resultadosmnkadf=data.frame(resultadosmnka)
  resultadosmnkadf$ind=unique(dados[,2])
  resultadosfinal=apply(resultadosmnka,2,sum)
  resultadosfim=list("MNKA por sessão"=resultadosfinal)
  write.csv(resultadosmnkadf, "Data.frame modificado através do método do
MNKA.csv")
  cat("\n \t Foi salvo em seu diretório de trabalho os seguintes arquivos:
\n \t \t Matriz com o total de captura por sessao.csv \n \t \t Dataframe com
o total de captura por sessao \n \t \t Data.frame modificado através do
método do MNKA.csv \n\n\n")
  return(resultadosfim)
}
}
```

## Help da Função

calc.mnka

package: -

R Documentation

Calcula MNKA e Contagem Populacional

### Description:

Utiliza um data.frame que nas linhas possui todos os registros dos indivíduos capturados; e que nas colunas possui as sessões de captura, a identificação do animal e os dias de captura. Calcula o MNKA - Minimum Number Known to be Alive(Krebs, 1966)- por sessão de captura. Além deste calculo, pode calcular a contagem simples dos indivíduos por sessão de captura. Gera uma matriz e um data.frame que resumem cada sessão de captura à apenas uma coluna, facilitando a comparação entre diferentes sessões e facilitando a análise dos dados. Pode produzir dois gráficos, um com a quantidade de indivíduos capturados em cada sessão e outro com a curva de acumulação de indivíduos ao longo das sessões de captura. Por fim, gera um data.frame com os dados já transformados pelo método MNKA. Desta forma, fornece uma análise inicial à tabelas de captura e recaptura e ainda resume tais dados, facilitando a comparação e futuras análises.

### Usage:

```
calc.mnka(dados,contagem=TRUE)
```

Para a função funcionar é necessário carregar o pacote vegan.

### Arguments:

**dados:** data.frame onde a primeira coluna seja a sessão de captura, a segunda coluna o individuo e da terceira coluna à ultima os dias dentro da sessão de captura.

**contagem:** Argumento Lógico. O padrão é TRUE. Assim, a função retornará além do cálculo do MNKA por sessão, do data.frame com o resumo de cada sessão de captura, da matriz com o resumo de cada sessão de captura, do gráfico com o número acumulado de indivíduos por sessão e do data.frame com as modificações decorrentes do método do MNKA, o gráfico com o número de indivíduos capturados em cada sessão bem como a contagem simples dos indivíduos capturados. Caso o argumento seja FALSE, a função retornará o cálculo do MNKA por sessão, o data.frame com o resumo de cada sessão de captura, a matriz com o resumo de cada sessão de captura, o gráfico com o número acumulado de indivíduos por sessão e o data.frame com as modificações decorrentes do método do MNKA.

### Details:

O data.frame de entrada da função deve estar no formato (.csv), com separador “;”, o nome da primeira coluna deve ser sessao, da segunda coluna ind , da terceira coluna “dia 1” (para o primeiro dia de captura e assim por diante). A tabela de entrada pode conter Na, os quais serão convertidos em 0, ou seja, ausência de captura. A coluna sessao deve conter apenas o número da sessão, a coluna ind deve conter o número de identificação do indivíduo, sendo que em todas as sessões deve conter uma linha para cada indivíduo, independente se o mesmo foi ou não capturado naquela sessão. Os indivíduos devem estar numerados em ordem crescente, e tal ordem deve ser mantida em todas as sessões. Nas colunas seguintes, deve-se entrar com a quantidade de registros de cada indivíduo no determinado dia da sessão. O zero representa que o indivíduo não foi capturado naquela data. Outros valores significam que o indivíduo foi capturado. Para o correto funcionamento da função é necessário que o pacote vegan esteja carregado e que a planilha de entrada de dados siga todos os pré-requisitos citados acima.

Value:

Matriz com o total de captura por sessão = Retorna uma matriz que resume se o indivíduo foi ou não capturado em tal sessão. Zero representa não captura e 1 representa que tal indivíduo foi capturado em tal sessão. Tal matriz é salva no diretório de trabalho.

Dataframe com o total de captura por sessão = Retorna um Data.frame que resume se o indivíduo foi ou não capturado em tal sessão. Zero representa não captura e 1 representa que tal indivíduo foi capturado em tal sessão. Tal data.frame é salvo no diretório de trabalho.

Data.frame modificado através do método do MNKA = Retorna um Data.frame com o método de MNKA aplicado. Tal data.frame é salvo no diretório de trabalho.

Retorna a contagem simples do número de indivíduos por sessão.

Retorna o valor de MNKA para cada sessão.

Retorna um gráfico com a quantidade de indivíduos capturados em cada sessão.

Retorna um gráfico com a curva de acúmulo de indivíduos ao longo das sessões.

References:

Crawley, M.J. (2007) The R Book. Ed: Wiley. 942 p.

Fernandez, F.A.S. (1995) Métodos para estimativas de parâmetros populacionais por captura, marcação e recaptura. Oecologia Brasiliensis (2)1-26.

Krebs, C. J. (1966) Demographic changes in fluctuating populations of Microtus californicus. Ecol. Monogr (36) 239 - 273.

Krebs, C. J. (1999) Ecological Methodology, 2 edição. Ed: Benjamin Cummings.



624 p.

Author:

Enrico Frigeri - e\_frigeri@hotmail.com

See Also:

specaccum() ---- Pacote vegan

apply()--- Pacote básico

sum() ---Pacote básico

Examples:

```
dados=read.table("planilha entrada de dados.csv" ,sep=";", header=TRUE,  
as.is=TRUE) #lendo a planilha que tem que #estar separada por ponto  
virgula
```

```
require(vegan)      ###carregando o pacote vegan
```

```
calc.mnka(dados, contagem=TRUE) #Caso queira a contagem dos indivíduos por  
sessão
```

#ou

```
calc.mnka(dados, contagem=FALSE) #Caso não queira a contagem dos indivíduos  
por sessão
```

## Testando a Função

```
dados=read.table("planilha_entrada_de_dados2.csv" ,sep=";", header=TRUE,  
as.is=TRUE) #lendo a planilha que tem que #estar separada por ponto  
virgula
```

```
head(dados)
```

```
summary(dados)
```

```
tail(dados)
```

```
dim(dados)
```

```
str(dados)
```

```
require(vegan)      ###carregando o pacote vegan
```

```
calc.mnka(dados, contagem=TRUE) #Caso queira a contagem dos indivíduos por  
sessão
```

#ou

```
calc.mnka(dados, contagem=FALSE) #Caso não queira a contagem dos indivíduos  
por sessão
```

## Arquivo da função

[planilha\\_entrada\\_de\\_dados2.csv](#)

From:

<http://lbtrop.ib.usp.br/> - **Laboratório de Ecologia de Florestas Tropicais**

Permanent link:

[http://lbtrop.ib.usp.br/doku.php?id=cursos:ecor:05\\_curso\\_antigo:r2011:alunos:trabalho\\_final:enrico:start](http://lbtrop.ib.usp.br/doku.php?id=cursos:ecor:05_curso_antigo:r2011:alunos:trabalho_final:enrico:start)



Last update: **2020/07/27 18:48**