

Thayná J. Mello

✘ Sou aluna de mestrado em Ecologia no IBUSP. Faço parte do LABTROP, e atualmente trabalho com invasão biológica. No meu Projeto de mestrado estou investigando a invasão da leguminosa *Leucaena leucocephala* na ilha de Fernando de Noronha.

Meus Exercícios

[exec](#)

Minha Proposta

Plano A

Monitoramento de plantas

Muitos trabalhos em Ecologia Vegetal envolvem o monitoramento de plantas ao longo do tempo. Entre outras variáveis, em geral são monitorados o crescimento e a sobrevivência das plantas submetidas a diferentes tratamentos. A cada nova entrada de dados é realizada uma análise exploratória dos dados, assim como gráficos que permitam acompanhar as tendências destas variáveis ao longo do tempo. A proposta é desenvolver uma função que tenha como entrada dois tipos de matrizes: 1) Matriz de dados com fatores correspondentes ao diferentes tratamentos e variáveis de medidas (por exemplo diâmetro, altura, número das folhas) de plantas ao longo do tempo. 2) Matriz com fatores correspondentes ao diferentes tratamentos e vetores lógicos (T- vivo e F- morto) indicando a situação de cada indivíduo em determinada data. No primeiro caso a saída seria um gráfico com uma linha para cada tratamento, sendo que cada ponto apresentaria o valor médio e os desvios para cada variável a cada instante do tempo. No segundo caso o gráfico apresentaria a sobrevivência percentual dos indivíduos em cada tratamento, a cada instante. Esta função seria útil não só para a análise exploratória dos dados coletados durante meu mestrado, mas também para qualquer trabalho envolvendo monitoramento de sobrevivência e crescimento de plantas.

Plano B

Parques Marinhos - Capacidade de suporte da atividade de mergulho

A atividade de mergulho autônomo pode danificar os organismos marinhos, como quebra de corais e suspensão de sedimentos prejudiciais aos organismos filtradores. Diversos estudos buscam determinar a sensibilidade de cada área, com levantamentos bióticos e abióticos (quantidade de corais ramificados, profundidade, correnteza, etc). Outros estudos buscam traçar o perfil do mergulhador, identificando aqueles que causam maior impacto. Neste caso, me refiro a impacto literalmente, já que a medida é feita contando quantos chutes cada mergulhador dá nos corais ao longo de um determinado tempo. Muitos parques marinhos no Brasil e pelo mundo afora mantêm um

monitor para acompanhar os grupos de mergulhadores e realizar esta contagem (confesso: já fiz isso)! Há também trabalhos que tentam estimar a capacidade de suporte da atividade de mergulho autônomo em áreas protegidas, ou seja, quantos mergulhadores determinado local pode receber ao longo de um ano sem que haja danos sérios à vida marinha, valor que fica em torno de 5000-6000 mergulhos por ano em áreas com recifes de coral (Zakai & Furman, 1997).

A proposta é elaborar uma função para estimar a capacidade de suporte de um determinado local, e que ajude o gestor da área protegida a saber quando a capacidade de suporte foi atingida. O valor "default" seria de 5000 mergulhos, e os argumentos da função seriam relativos ao perfil do mergulhador (sexo, nível de certificação, experiência, etc), com dados baseados no trabalho de Rouphael & Inglis, 2000 (por exemplo, mergulhadores usando câmeras fotográficas causam 1.6 vezes mais quebras em corais do que os sem câmera) e da operação de mergulho (visibilidade, presença de guias, fotógrafos, ancoragem de embarcação) Assim, poderia ser obtida uma estimativa relativa mais próxima da capacidade de suporte, adaptada ao uso real do local de mergulho.

A entrada de dados seria feita com uma matriz de contagens de número de pessoas pertencentes a cada perfil que mergulharam no local a cada dia, e a função calcularia o aumento ou diminuição relativa do impacto, de forma que o usuário poderia saber em que momento a capacidade de suporte é atingida. Em princípio estes valores seriam baseados na literatura sobre o assunto (números mágicos), mas a função poderia ser aperfeiçoada à medida em que mais conhecimento sobre cada local fosse obtido com pesquisas sobre a vida marinha e monitoramento da atividade em cada local. Seria uma forma interessante de sistematizar os muitos dados coletados sobre chutes e toques e quebras de corais, e torná-los em um produto mais útil ao manejo das áreas protegidas.

Literatura citada:

Rouphael, A. B. & Inglis, G.J. 2001. "Take only photographs and leave only footprints"?: An experimental study of the impacts of underwater photographers on coral reef sites. *Biological Conservation* 100:281-287.

Zakai, D; Chadwick-Furman, N.E. 2002. Impacts os intensive recreational diving on reef corals at Eilat, northern Red Sea. *Biological Conservation* 105:179-187.

Comentarios

Thayna,

Achei a proposta A um pouco confusa, mas acho que deve ficar com ela. Poderia fazer, ao inves desvio, o intervalo de confiança bootstrap para cada variavel, inclusive para a porcentagem de mortalidade, ao longo do tempo (esqueça a linha!), acho que fica um desafio mais interessante. O de mergulho fica para outro momento... meio polemico!

— [Alexandre Adalardo de Oliveira](#) 2013/03/25 11:11

Minha função

```
#####  
##FUNÇÃO IC.CENSUS##
```

```
#####
```

```
ic.census<-function(dados, tempo, replica, variavel, nsim=100, add=FALSE,
ylab="")

{
#####PARTE 1:
VERIFICAÇÕES#####
#verificando se as colunas do data.frame sao das classes requeridas pela
função:
if(class(dados)!="data.frame")
{stop("dados deve ser um data.frame")}
if(class(dados[,tempo])!="factor")
{stop("tempo deve ser da classe 'factor'")}
if(class(dados[,replica])!="factor")
{stop("replica deve ser da classe 'factor'")}
if(class(dados[,variavel])!="integer"& class(dados[,variavel])!="numeric")
{stop("A coluna 'variavel' deve ser das classes 'integer' ou 'numeric'")}
#verificando se o pacote plyer está instalado
if (sum(installed.packages()[,1]!="plyr")==1)
{stop("Lembre-se que é necessário instalar o pacote plyr para rodar a
função!")}
}
#####PARTE 2: MANIPULANDO OS
DADOS#####
#o número de censos será usado em diversas indexações na função,
#então aqui criei um objeto para ele
ncensus<-length(levels(dados[,tempo]))

#criando uma matriz contendo o valor da variavel de interesse em cada
censo
#Matriz para armazenar os dados:cada linha irá conter a informação
referente a um censo
names(dados)[variavel]<-"variavel"
matriz<-matrix(NA, nrow=ncensus, ncol=length(levels(dados[,replica])))
for (i in 1:ncensus)
{
matriz[i,<-dados$variavel[dados[,tempo]==levels(dados[,tempo])[i]]
}
rownames(matriz)<-paste("censo", 1:ncensus, sep=" ")
colnames(matriz)<-1:length(levels(dados[,replica]))
#calculando a media de sobreviventes por censo
media.surv<-rep(NA, ncensus)
for(i in 1:ncensus)
{
media.surv[i]<-mean(matriz[i,])
}
media.surv #vetor com a média de cada linha da matriz, ou seja, de cada
censo
names(media.surv)<-1:ncensus
```

```
#####PARTE 3: REAMOSTRAGEM E CÁLCULO DO  
IC#####  
  resulta=matrix(NA,nrow=ncensus, ncol=nsim)  
  for(i in 1:nsim)  
  {  
    for (j in 1:ncensus)  
    {  
      resulta[j,i]<-mean(sample(matriz[j,], replace=TRUE))  
    }  
  }  
  #carregando o pacote "plyr", que tem a função "aapply", para aplicar uma  
função  
  #a cada margin da matrix.  
  require("plyr")  
  #Ordenando cada linha da matriz "resulta" independentemente  
ordenado<-aapply(resulta,1,sort)  
  #selecionando os valores referentes a 5% e 95% dos dados  
  ic05<-ordenado[,nsim*(6/100)]  
  ic95<-ordenado[,nsim*(95/100)]  
#####PARTE 4:  
GRÁFICO#####  
  if(add==FALSE)  
  {  
    par(bty="l", mar=c(5,4,2,3))  
    #plotando as médias ao longo dos censos  
    plot(media.surv, xlim=c(1,ncensus),xlab="censo",ylab=ylab,  
         xaxt="n", ylim=c(min(ordenado),max(ordenado)))  
    axis(1, at=c(1:ncensus))  
    #plotando os intervalos de confiança no gráfico  
    segments(x0=1:ncensus,y0=ic05,x1=1:ncensus,y1=ic95,lwd=1, col="black")  
    points(x=c(1:ncensus,1:ncensus), y=c(ic05,ic95), pch="-")  
  }  
  if(add==TRUE)  
  {  
    color<-sample(rainbow(30,s=1,v=1,alpha=1, start=.7, end=.1))[1]  
    #adicionando pontos a um gráfico existente  
    points(media.surv, col=color)  
    #plotando os intervalos de confiança no gráfico  
    segments(x0=1:ncensus,y0=ic05,x1=1:ncensus,y1=ic95,lwd=1, col=color)  
    points(x=c(1:ncensus,1:ncensus), y=c(ic05,ic95), pch="-", col=color)  
  }  
#####PARTE 5: FINALIZANDO  
#####  
  #criando uma lista com os principais objetos criados dentro da função, que  
podem ser  
  #úteis para o usuário  
  lista<-list(matriz, media.surv, ic05, ic95)  
  names(lista)<-c("variavel de interesse por censo","média da variavel de  
interesse por censo", "IC 5% por bootstrap", "IC 95% por bootstrap" )
```

```
#Fim
return(lista)
}
```

Help da função

ic.census package:nenhum R Documentation

Intervalo de confiança por bootstrap de uma variável ao longo do tempo.

Description:

A partir de um `data.frame` contendo os valores de uma variável de interesse por réplica e censo, a função calcula a média da variável e o intervalo de confiança (5% e 95%) por bootstrap a cada censo, e plota os valores em um gráfico. A função pode ser usada para calcular intervalos de confiança de variáveis contínuas com medidas efetuadas ao longo do tempo, como por exemplo censo de sobrevivência de indivíduos, medidas de altura e circunferência de plantas em censos de parcelas permanentes, registros de ocorrência de espécies em programas de monitoramento. A função pode plotar vários grupos de pontos num mesmo gráfico, permitindo comparar os efeitos de diferentes tratamentos sobre a variável de interesse.

Usage:

```
ic.census(dados, tempo, replica, variavel, nsim=100, add=FALSE, ylab="")
```

Arguments:

`dados` objeto da classe `data.frame`, cujas colunas contenham o valor da variável de interesse por réplica por censo, a identificação do censo e a identificação da réplica.

`tempo` Número da coluna do `data.frame` que indica a qual censo o valor da variável de interesse pertence. A coluna deve ser da classe `factor`.

`replica` Número da coluna do `data.frame` que indica a qual réplica o valor da variável de interesse pertence. A coluna deve ser da classe `factor`.

`variavel` Número da coluna do `data.frame` que contém os valores da variável de interesse.

A coluna deve ser da classe numeric ou integer.

`nsim` Número de ciclos de reamostragem com reposição para cálculo do intervalo de confiança por bootstrap.

`add` Se FALSE os valores da média e intervalo de confiança serão plotados em um novo gráfico. Se TRUE os valores serão plotados em um gráfico existente.

`ylab` Legenda do eixo y do gráfico.

Details:

O input da função deve ser um data.frame cujas colunas contenham as réplicas (factor), os censos realizados (factor) e os valores da variável de interesse em cada réplica (integer ou numeric) a cada censo. É necessário ter instalado o pacote plyr para rodar a função. O pacote plyr será carregado durante a execução da função.

Value:

A função retorna uma lista com os seguintes componentes:

\$`variavel de interesse por censo`: Matriz em que cada linha contém os valores da variável de interesse referente a um censo.

\$`média da variavel de interesse por censo`: Vetor numérico com a média de cada linha da matriz `variavel de interesse por censo`, ou seja, a média de cada censo.

\$`IC 5% por bootstrap`: vetor numérico com o intervalo de confiança 5% de cada censo.

\$`IC 95% por bootstrap`: vetor numérico com o intervalo de confiança 95% de cada censo.

Author:

Thayná J. Mello - thayna@usp.br

References:

Davison, A.C. and Hinkley, D.V. (1997) Bootstrap Methods and Their Application. Cambridge University Press.

Examples:

```
###Growth of Loblolly pine trees###
```

```
require(datasets)
```

```
data(Loblolly)
```

```
growth<-data.frame(Loblolly)
```

```
growth$age<-as.factor(growth$age)
```

```
growth$seed<-as.factor(rep(1:14, each=6))
```

```
ic.census(dados=growth, tempo=2, replica=4, variavel=1, nsim=100,  
ylab="altura(pés)")
```

```
###Monthly Airline Passenger Numbers###
```

```
data(AirPassengers)
```

```
passageiros<-data.frame(AirPassengers)
```

```
censo<-as.factor(rep(1949:1960, each=12))
```

```
replica<-as.factor(rep(1:12, each=12))
```

```
data<-data.frame(replica,censo, passageiros)
```

```
data$AirPassengers<-as.numeric(data$AirPassengers)
```

```
ic.census(dados=data, tempo=2, replica=1,variavel=3,nsim=100, ylab="número  
de passageiros")
```

```
###Growth of Orange Trees###
```

```
data(Orange)
```

```
laranja<-data.frame(Orange)
```

```
laranja$tree<-as.factor(rep(1:5, each=7))
```

```
laranja$age<-as.factor(laranja$age)
```

```
ic.census(dados=laranja, tempo=2, variavel=3,replica=4, nsim=100,  
ylab="circunferência (mm)")
```

[funcao_bootstrap.r](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:thaynajm:start 

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