

Cristiane Follmann Jurinitz



Aluna de doutorado da Ecologia da USP.

Plano A

Estudos demográficos são essencialmente observacionais e baseados em medidas simples, mas que geram planilhas brutas enormes, a partir das quais é necessário extrair uma série de informações, que, quando não automatizadas numa função, podem nos dar muito trabalho e tomar muito tempo! (Bom, essa não é uma característica exclusiva desses estudos...)

Pensei em elaborar uma função para extrair informações básicas de um conjunto de dados demográficos de plantas (nesse caso árvores).

De cada indivíduo são tomadas medidas de tamanho em diferentes tempos. Como se trata de árvores, um indivíduo pode ter mais de uma medida (no caso de apresentar mais de um fuste). Por isso, nesse caso, inicialmente é necessário gerar um único valor que represente cada indivíduo. Essa seria a primeira tarefa que essa função executaria: calcular as áreas basais de cada fuste de determinado indivíduo, somá-las e gerar um valor final de diâmetro à altura do peito (como é usual, pois a somatória dos DAP's geraria super-estimativas).

Uma vez que cada indivíduo tem seu tamanho correspondente em determinado tempo, a tarefa seguinte seria a contagem do número de indivíduos em cada classe de tamanho (definidas previamente). Esta contagem do número de indivíduos por classe de tamanho seria feita ainda por localidade e por espécie, em cada tempo.

Mas, como produto final para o início da modelagem da demografia das espécies, é preciso saber a probabilidade de transição dos indivíduos pertencentes à determinada classe de tamanho, ou seja, integrar as contagens de indivíduos por classe ao longo do intervalo de tempo.

Nesta etapa, é preciso ponderar as contagens em cada classe com a mortalidade, a permanência dos indivíduos em determinada classe, ou o seu crescimento ou retração para outras classes.

Essa integração temporal dos dados de demografia seria uma terceira tarefa que a função executaria, mas que na verdade não tenho certeza se consigo incorporar (mas gostaria) tanto em função do tempo necessário para pensar isso quanto do grau de dificuldade...

Comentários

Alexandre

Acho muito legal! Não é trivial e vc. ficará craque na indexação do R! Vai tomar tempo e demandar muito de vc, mas ao final terá uma ferramenta muito útil para o seu trabalho. Sugiro que vc. deixe como argumento a possibilidade de definir as classes de tamanho. Comece pensando no formato de entrada e no de saída. Minha sugestão é usar uma planilha pequena (poucos registros) de uma espécie em uma localidade em um único tempo e fazer todo o processo no console do R. Depois juntar os comandos do script para iniciar a função. Caso consiga fazer tudo no R para uma espécie em um tempo e uma localidade a idéia geral do algoritmo já está estabelecido. Aí, o passo seguinte é

fazer os ciclos para incluir tempos distintos e localidades... Boa sorte!

Sobre a entrega da função

Olá professores, lamento não ter conseguido completar minha função no prazo.

Acabei ficando muito tempo para resolver a primeira parte, não tendo conseguido avançar muito além dela...

Em função disso, não consegui começar a página de help e mantive os comentários no código.

No arquivo da função, mantive todas as tentativas que fiz até agora.

Ver item acima: sobre a entrega da função...

Página de Ajuda

qqnorm.ci

package:nenhum

R Documentation

Gráfico dos quantis da normal com intervalo de confiança empírico

Description:

Produz um gráfico dos valores de um vetor em função dos quantis esperados pela distribuição normal. Adiciona ao gráfico uma linha de inclinação = 1, e também os limites do intervalo de confiança simulado.

Usage:

```
qqnorm.ci(x, alfa=0.9, nrep=1000, qqline=TRUE, ci=TRUE, ...)
```

Arguments:

x: Vetor numérico. Valores de uma amostra.

alfa: numérico. Probabilidade associada ao intervalo de confiança (nível de confiança).

nrep: número de randomizações para o cálculo do intervalo de confiança simulado (ver detalhes)

qqline: lógico. Acrescentar linha do esperado?

ci: lógico. Calcular o intervalo de confiança simulado?

...: qualquer argumento a repassar para a função plot.

Details:

Os valores da amostra (x) são ordenados e plotados contra os quantis esperados para uma distribuição normal com média e desvio padrão iguais aos da amostra. A linha esperada tem inclinação um e intercepto zero, e difere da obtida com a função 'qqline', que passa pelos segundo e terceiro quartis. Para o intervalo de confiança, nrep amostras de mesmo tamanho de x são simuladas de uma distribuição normal de mesma média e variância de x. Os valores em cada amostra simulada são ordenados e o intervalo de confiança é aquele que inclui a proporção dos valores simulados definida por alfa. Estes intervalos são obtidos aplicando-se a função 'quantile' aos nrep valores simulados do primeiro, segundo, terceiro, enésimo valor do vetor ordenado.

Value:

Um gráfico é gerado. Um data frame é retornado mas não é exibido na tela, com os seguintes vetores:

Obs : valores da amostra em ordem crescente

Percentis: os percentis correspondentes a cada elemento de x

Quantis: os quantis esperados pela distribuição normal

Lower, Upper: se o argumento ci é verdadeiro, os limites inferior e superior do intervalo de confiança simulado.

Warning:

O intervalo simulado não é uma solução analítica e serve apenas para fins exploratórios.

Author(s):

Fulano de Tal

fulano@fulanonet.com.br

References:

Cleveland, W.S. (1994) The Elements of Graphing Data, Hobart Press ISBN 0-9634884-1-4

Wilk, M.B., Gnanadesikan, R. (1968) Probability plotting methods for the analysis of data.

Normal probability plot.
http://en.wikipedia.org/wiki/Normal_probability_plot

See Also:

'qqnorm' e 'qqline' do pacote base, para o gráfico de quantil padrão do R, e a função 'ppoints', com a qual os percentis são calculados.

Para uma função similar a esta mas bem mais completa, veja 'qq.plot', do pacote car.

Examples:

```
qqnorm.ci(rnorm(100)) ## Uma amostra de uma normal padronizada.  
qqnorm.ci(rexp(100, rate=10)) ## Amostra de uma distribuição exponencial  
com média =10  
valores <- qqnorm(rpois(30,5)) ## Cria um objeto com os valores do  
gráfico
```

Código da Função

```
#####TRABALHO  
FINAL#####  
#####CRISTIANE FOLLMANN  
JURINITZ#####  
  
###Produzir uma função que extrai informações básicas de uma planilha de  
dados de demografia de árvores###
```

```
###Tentativa de Primeira Versão a partir de 28/03/2009###
```

```
#Esta função está sendo pensada para uma planilha típica de dados de
demografia de árvores, onde as linhas são os indivíduos e as colunas as
medidas tomadas, sendo que repetições temporais da mesma medida representam
colunas diferentes.
```

```
#0 indivíduo pode ter uma das seguintes medidas: DAP(diâmetro à altura do
peito), para plantas com DAP até 70mm (com paquímetro), PAP (perímetro à
altura do peito - fita métrica) ou DAS (diâmetro à altura do solo), quando à
altura do peito ele apresenta DAP<10mm ou tem altura inferior a 1,3m.
```

```
#As medidas em cada tempo estão identificadas por um número: 0, 1 e 2
```

```
#####CÁLCULO DA ÁREA
```

```
BASAL#####
```

```
ab <- function (dados)
{
  dados.dap0=dados$dap0
  dados$ab.dap0=(pi*(dados.dap0^2))/4
  dados.das0=dados$das0
  dados$ab.das0=(pi*(dados.das0^2))/4
  dados.pap0=dados$pap0
  dados$ab.pap0=(dados.pap0^2)/(4*pi)
  return(dados)
}
```

```
#DEU CERTO!!!!!!!!!!!!!!!!EBA!!!!!!!!!!!!!!!!!!!!!!
```

```
#####
#####
```

```
#Daqui em diante, deixei os comentários, pois não consegui concluir por
conta de alguns impasses...
```

```
#####CÁLCULO DO DIÂMETRO
```

```
FINAL#####
```

```
diametro <- function (dados)
{
  dados$soma.ab=apply(cbind(dados$dap0,dados$das0,dados$pap0),1,
FUN=sum(na.rm=T)) #tentativa de fazer uma somatória sem os NA's, mas não
funcionou por causa do na.rm=T, dá a seguinte mensagem de erro: 'sum(na.rm =
T)' is not a function, character or symbol
```

```
#e depois somá-las para cada placa, que é a identidade do indivíduo
```

```
diam.ind=aggregate(dados$soma.ab, list(dados$placa), FUN=sum)
```

```
#para finalmente gerar um valor final de diâmetro que servirá para encaixar
o indivíduo numa determinada classe de tamanho
```

```
diam.final=sqrt((4*diam.ind)/pi)
```

#agora, como número de linhas será igual ao número de indivíduos e não mais ao número de fustes, não posso mais pedir para ele retornar o 'dados';
preciso criar um arquivo para armazenar os resultados?

```
    return(?)  
}
```

#####ENCAIXE NAS CLASSES DE
TAMANHO#####

#Esta seria a próxima etapa, mas não testei ainda, tenho um "palpite" a partir da dica do Alexandre de que as classes seriam argumentos:

```
classes <- function (dados, class.1= <= 10, class.2= > 10 <= 20, class.3= >  
20 <= 30, class.4 = > 30 <= 40, class.5= > 40 <= 50, ...)  
{  
  if(dados$diam.final <= 10) ... #não consigo pensar em como "dizer" ao R  
  para retornar uma classe!  
  ...  
}
```

Arquivo da Função

[funcao_cristiane.r](#)

From:

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

Permanent link:

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



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