

Milena de Godoy-Veiga



Doutoranda em Botânica no Instituto de Biociências da USP

Título da tese: Reconstrução das mudanças climáticas nos últimos dois séculos por meio de estudos dendrocronológicos e isotópicos de árvores na região do Parque Nacional Cavernas do Peruaçu. Orientada pelo Prof. Dr. Gregório Ceccantini.

Link para meus exercícios: [Exercícios](#)

PROPOSTA A

`tree.gRowth (data, year, hm, histogram, lyr = TRUE, return = FALSE)`

Descrição:

Uma função para análise exploratória de medidas de anéis de crescimento. A partir de data frames com as amostras distribuídas por colunas, a função entrega análises básicas da estrutura etária da população (histograma de idades e boxplot), taxa de crescimento (gráfico de crescimento acumulado) e padrão de crescimento (scatterplot de crescimento por idade).

Entrada:

`data` = nome do documento .txt ou .csv em que os dados de crescimento estão. Aceita data frames que contenham uma amostra por coluna. Primeira linha contendo o nome das amostras. A segunda linha deve conter a medida do último anel formado (`lyr = TRUE`). Se os valores estiverem invertidos, `lyr = FALSE`.

`year` = um vetor com dois valores, de mês e ano em que a coleta foi feita (ex. `c(6, 2018)`)

`hm` = em qual hemisfério a coleta foi feita ("south" ou "north")

`histogram` = um vetor com a idade inicial, final e o intervalo desejado no histograma (ex. `c(0, 300, 25)`)

`lyr = TRUE` indica que a segunda linha contém a medida do último anel formado. Se `FALSE` a primeira linha contém apenas as medidas da(s) árvore(s) mais velha(s).

`return = TRUE` retorna os data.frames gerados

Testando entradas:

`data` é um arquivo .txt ou .csv igual ao especificado (ver se primeira linha tem nomes e segunda números)? Se não, retorna "Check your data. Should be something like this: escrever um exemplo de data frame"

`year` é um vetor com duas posições, e onde a primeira é um número entre 1 e 12? Se não, retorna "Check your year. Should be something like this: `my.year ← c (4,2018)`"

`hm` é igual a "sul" ou "norte"? Se não, retorna "Check parameter hm. Should be south or north"

`histogram` é um vetor com três valores numéricos e a divisão do segundo pelo terceiro é um número inteiro? Se não, retorna "Check parameter histogram. Should be something like this:

```
my.histogram ← c(0, 300, 25)"
```

Pseudo-código:

#Arrumando as planilhas

1. A partir de data, criar um data.frame "growth"
 1. If lty = FALSE inverter os valores das colunas para que a segunda linha contenha os valores dos anéis formados no último ano.
 2. Onde não existem valores (final da coluna, já que as árvores mais velhas terão mais valores do que as mais novas), existem NA ou 0? Preencher tudo com NA
 3. if (hm == "sul" e year [1] > 7) #se estivermos no hemisfério sul e a coleta for realizada depois do mês de julho, temos que subtrair um ano já que a estação de crescimento nesse hemisfério começa no meio do ano.
 1. Adicionar uma coluna "Years" com valores decrescentes, começando com year [2] - 1
 4. else
 5. Adicionar uma coluna "Years" com valores decrescentes, começando com year [2]

#Análises

1. for. Para cada coluna de "growth" será calculado:
 1. Comprimento (length). Guardar esse valor em um vetor numérico "ages" #Esse comprimento é igual a idade
 2. Crescimento acumulado, que será o valor da última posição de cada coluna (ano 1 da árvore) somado a seguinte (ano 2), até o primeiro valor. Guardar estes valores em um data frame "accumulated.growth", onde a primeira coluna será igual a primeira coluna de "growth".
2. Criar um segundo data frame "lifetime.growth.pattern" ← "growth"
 1. A primeira coluna será "cambial age", começando do 0 e irá até a idade da árvore mais velha.
 2. Inverter todos os valores, retira os Nas # todos os anos 0 das árvores deverão estar na segunda linha agora.
 3. Preencher com NA o final das colunas

#Resultados

1. Construir um histograma da frequência das idades, calculado a partir de "ages", pelo intervalo indicado em histogram
2. Construir um boxplot com "ages"
3. Criar um gráfico de crescimento acumulado de "accumulated growth" mostrando todas as amostras como linhas. Adicionar uma linha de crescimento médio e o valor da inclinação.
4. Construir um scatterplot com os valores de "lifetime.growth.pattern" (crescimento por "cambial age") Adicionar uma linha de média suavizada (LOESS).

Saída:

mostrar em um painel (2,2)

1. Histograma com distribuição de idades
2. Boxplot das idades
3. Gráfico de crescimento acumulado de todas as amostras


4. Scatterplot do padrão de crescimento de todas as amostras
5. Se `return = TRUE`, retorna os `data.frames` *"accumulated growth"* e *"lifetime.growth.pattern"*

PROPOSTA B

`kit.inter(kitC0, kitA, lucro, alunos)`

Descrição:

Uma função para ajudar Atléticas a calcular o preço cobrado nos kits para eventos esportivos

universitários (inters ). É possível entrar com os valores que foram gastos, as margens de lucro desejadas e a quantidades de alunos que se deseja simular. A função retornará os diferentes cenários escolhidos (de lucro e quantidade de alunos) e os valores que devem ser cobrados nos kits em cada cenário.

Entrada:

`kitC0` = valor que a Comissão Organizadora do inter cobra por kit. Número maior que 0.

`kitA` = valor do kit que cada atlética monta. Número maior que 0.

`lucro` = valor(es) indicando a(s) porcentagem(ns) de lucro que a Atlética deseja. Vetor com um ou mais valores maiores que 0 (ex. `c(5, 7.5)`)

`alunos` = valor(es) indicando a(s) quantidade(s) que a Atlética espera de alunos comprando o kit final. Vetor com um ou mais valores maiores que 0 (ex. `c(170, 190, 200)`).

Testando entradas:

`kitC0` é um número maior que 0? Se não, retorna *"kitC0 precisa ser um número maior que zero, como 110"*. `kitA` é um número maior que 0? Se não, retorna *"kitA precisa ser um número maior que zero, como 15"*. `lucro` é um vetor com número(s) maior(es) que 0? Se não, retorna *"lucro precisa ser um vetor com número(s) maior que zero, como lucro ← c(5, 7.5)"*.

`alunos` é um vetor com número(s) maior(es) que 0? Se não, retorna *"alunos precisa ser um vetor com número(s) maior que zero, como alunos ← c(170, 190, 200)"*.

Pseudo-código:

#Calculando gastos

1. para cada posição de `alunos`
 1. `Gastos ← alunos [] * (kitC0 + kitA)`
 2. Salvar o valor em um vetor *"gastos"*

#Análises

1. para cada posição de `lucro`
 1. Salvar em um `data.frame KIT` os resultados de $= ((\text{lucro}/100 + 1) * \text{"gastos"})/\text{alunos}$

#Resultados

1. `Data.frame KIT`
 1. Nome das linhas sendo os valores de `alunos` e nome das colunas sendo os valores de `lucro`

Saída:

-Um data.frame com as diferentes categorias de quantidades de alunos (nome das linhas) e lucros (nome das colunas) e os valores que os kits devem ser cobrados em cada categoria.

Comentários Julia

PROPOSTA A

Oi Milena,

Achei a proposta A bastante clara e factível. Duas sugestões:

1. Caso o argumento `return=TRUE` ele poderia retornar os data frames como arquivos salvos diretório de trabalho da pessoa. Você pode inclusive criar um argumento para indicar o nome que dará a esse data frame. Sugiro isso, pois muitas vezes não é útil que o R retorne uma tabela enorme em nosso console, mas sim que forneça essa tabela para que possamos acessá-la quando e como quisermos.
2. Poderia incluir na sua função, um ciclo para multiplas fontes de dados: Um exemplo possível: Imagine um banco de dados em que temos vários data frames como este que você descreveu. Seria um arquivo csv para cada localidade de coleta. Todos salvos em uma pasta X. A função realizaria este trabalho para todos os data frames.

Neste caso para te ajudar:

- Pode fazer a função ler todos os arquivos csv dentro daquela pasta X (o usuário irá dizer qual é a pasta X através de um argumento)
- A função realiza as correções e cálculos e gráficos para todos
- Salva um(s) data frame(s) de saída (se `return=TRUE`) separado(s) para cada uma, com um nome de arquivo compatível (é possível copiar o mesmo nome de arquivo do data frame de origem dos dados :: `file_path_sans_ext`)
- Além disso, caso a saída para cada dataframe de entrada seja mais de um dataframe gerado com os calculos, então pode também criar uma pasta para cada dataframe de entrada e nela salvar seus resultados.

Ex: Sua pasta Pasta X, contém dados1.csv, dados2.csv, dados3.csv No final você teria dentro da Pasta X: pasta1, pasta2, pasta3, e dentro de cada uma delas os dataframes resultados da função para respectivamente os data.frame1, data.frame2, data.frame3

PROPOSTA B

A proposta B me parece que ainda precisa de bastante lapidação. A) Poderia haver mais flexibilidade para o usuário através de outros argumentos B) Ainda não está com cara de função. Do modo colocado a função iria resolver um cálculo matemático, mas não uma função Na verdade na própria descrição que você apresentou já indica o código quase completo: $Gastos ? alunos [] * (kitCO + kitA) (...) = ((lucro/100 + 1) * "gastos")/alunos$

É possível seguir nesta ideia mas teria que dar maior complexidade para se tornar uma função e não expressão matemática em duas linhas. Coisas que pensei:

1. O usuário indica em que mês estamos ou a função acessa a data do computador para saber em que época do ano estamos. função Sys.Date() A partir disso a função distingue entre primavera/verão e outono/inverno.
2. Um argumento seria adicionado distinguindo o valor kitCO em duas possibilidades preço do kit simples (vem somente bata) e preço do kit completo (vem bata e moletom). (Sabemos que na realidade o kit do inter não tem moletom :D . Pode ser galocha! Já que no inter é comum rolar bastante barro,rs)
3. Na hora de realizar os cálculos ele realiza diferentes combinações de quantidades de kit simples X kit completo que atinjam o determinado lucro. Mas para limitar estas combinações possíveis a função leva em consideração se é primavera/verão ou outono/inverno. No primeiro caso, espera-se o dobro de vendas do kit simples, e no segundo caso o contrário.

Esses valores de dobro, de tipo de kit e etc são apenas exemplos. A ideia foi fornecer algo para alimentar a criatividade, você pode tornar a função mais complexa de muitas outras maneiras.

Comentários finais:

A proposta A está bem redijida e possui um grau de complexidade intermediário. Pode torná-la mais sofisticada um pouco mas está bem clara e adequada, em um bom caminho. A proposta B está bem clara, mas sem complexidade para que justifique uma função.

Sugiro seguir com a proposta A pensando em incorporar sugestões à ela.

Ao fazer alterações na sua proposta, inclua aqui a nova proposta atualizada. (Não apague a antiga).

Estou à disposição caso fique com dúvidas :)

Beijos,

Julia

Link para arquivos do trabalho final: [Trabalho final](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:milena.veiga:start

Last update: 2020/08/12 09:04

