Daniela Coelho

Doutoranda em Ecologia pela USP. Minha pesquisa busca compreender as interações tróficas entre comunidades de serpentes neotropicais além da evolução do mimetismo Batesiano em serpentes camufladas. Sou orientada pelo Paulo Guimarães Jr. (Miúdo).

Exercícios

Trabalho Final

Proposta I: Organização do tempo

Contextualização

Aqueles que desfrutam da vida acadêmica precisam lidar com tarefas variadas, além de terem prazos apertado para entrega de resultados. Com o passar do tempo e do acúmulo das atividades, a tendência é a geração de ansiedade, especialmente porque muitas pessoas não conseguem organizar seu tempo de forma eficiente, o que gera investimento desnecessário em determinadas atividades e menos tempo dedicado a atividades de lazer, necessárias para manutenção da saúde mental. Para otimizar a realização das atividades acadêmicas, o principal objetivo dessa função é gerar uma planilha de organização de tempo semanal com base nas atividades essenciais de cada pessoa. É necessário pensar sobre a importância e o grau de dificildade de cada uma das atividades previstas para aquela semana e estabelecer, com base na sua experiência pessoal, quantas horas são necessárias para executar aquela tarefa. Existem alguns métodos que auxiliam no aumento da eficiência do rendimento, tais como o Pomodoro (https://pt.wikipedia.org/wiki/Técnica_pomodoro). Também é importante estabelecer o tempo mínimo dedicado a cada tarefa, assim como informar, na hora de construir a função, quais atividades são fixas (p. ex. reuniões e aulas), pois o horário referente a estas atividades não poderá ser ocupado na planilha. Da mesma forma, é preciso estabelecer qual o horário de início e qual o horário estabelecido para encerrar as atividades diárias.

Planejamento da função

```
Oi Dani,
Eu fiquei com algumas dúvidas com relação à função. Me ajuda a entender?
Fico com receio de que você esteja pedindo muitas informações ao usuário e
que, depois que ele pensar em tudo o que ele precisa pensar pra fornecer à
função, é mais fácil ele mesmo colocar na agenda.
Isso pode ter acontecido por que eu não tenho certeza se entendi seus
objetos de entrada. O que seria o intervalo entre cada atividade? O que é o
peso de um atividade e como ele vira um índice?
Será que poderíamos colocar os intervalos de atividade seguindo períodos de
descanso tipo o do pomodoro?
Me conta e vamos decidir sobre essas funções.

Beijo
Rena
```

Objetos e argumentos de entrada

- O usuário deve fornecer:

a) O horário de início e fim das atividades diárias e o intervalo entre cada atividade (linhas); O período de duração das atividades (número de colunas);

b) Fornecer a lista das atividades necessárias para aquele período e o peso de cada uma delas (equivalente ao tempo necessário de dedicação para cada tarefa);

- O usuário deve especificar

c) Quais horários não podem ser preenchidos, pois já possuem atividades fixas (ex: horário de almoço);

Saída

A partir da quantidade de horas livres para o período de tempo proposto, a função deverá retornar um data frame com a quantidade de horas necessárias para se dedicar a cada tarefa e em qual período do dia executar cada uma das atividades.

Passos a executar

1) Criar um data frame onde as linhas serão as horas de duração de cada tarefa e as colunas serão equivalentes aos dias da semana;

2) Criar um objeto com a lista de todas as atividades e os pesos atribuídos a cada uma delas;

3) Criar um objeto com as atividades fixas;

4) Gerar um data frame com mais de uma possibilidade de combinação de tempo/atividades para o período estabelecido.

Proposta II : Rotação de gado em piquetes

Contextualização

Um problema comum de criadores de gado é relativo ao rodízio dos animais nos diferentes piquetes da propriedade. É preciso compreender qual a melhor forma de executar o rodízio de maneira a otimizar o uso dos recursos sem exaurí-los. A rotação é realizada mantendo os animais dentro dos piquetes por um intervalo de tempo (intervalo que o gado irá consumir uma fração do capim) e retirando-os após o consumo previsto, para descanso do piquete e rebrotamento do capim até o tamanho desejado para reiniciar o rodízio. É necessário um conhecimento prévio sobre as taxas de crescimento (unidade de massa/unidade de área/unidade de tempo) da espécie de capim utilizada nos piquetes nas diferentes estações climáticas (seca e chuvosa) do ano, assim como da quantidade média de consumo (unidade de massa/tempo) de capim por indivíduo da espécie de gado utilizada na criação. Com tais informações, somadas a quantidade de cabeças de gado e ao tamanho de cada piquete da propriedade, poderemos prever qual a melhor forma de rotacionar os animais nas diferentes estações climáticas do ano.

Planejamento da função

```
Oi Dani,
```

```
Não sabia que você era dona de cabeças de gado... rs
Entendi o problema e acho que é algo necessário. Não tenho certeza de que
entendi se o trabalho que você vai ter com a função vale a pena ou não. A
solução me pareceu simples demais e não seria um desafio bacana.
Me conta mais ou vamos tentar pensar em outras informações importantes pra
calcular no contexto.

Beijo
Rena
```

Objetos e argumentos de entrada

- O usuário deve fornecer:

a) Tamanho de cada piquete;

b) Quantidade total de cabeças de gado e qual a espécie;

c) Biomassa da espécie de capim utilizada na estação seca e na estação chuvosa;

d) Quantidade média de consumo de capim por indivíduo da espécie de gado utilizada;

- O usuário deve especificar

e) Qual a estação climática, se a seca ou se a chuvosa, já que a biomassa de capim irá variar de acordo a estação.

Saída

A função deverá retornar o número de dias em que a quantidade de cabeças de gado deverá permanecer dentro e fora de cada piquete na estação climática de interesse.

Passos a executar

1) Cálculo da biomassa de capim para estação chuvosa;

2) Cálculo do tempo que o gado permanecerá dentro do piquete na estação chuvosa (biomassa do piquete / quantidade de cabeças de gado / consumo médio de capim por indivíduo por dia);

3) Cálculo do tempo que o gado permanecerá fora do piquete na estação chuvosa (taxa de crescimento médio da espécie de capim / unidade de área);

4) Cálculo da biomassa de capim para estação seca;

5) Cálculo do tempo que o gado permanecerá dentro do piquete na estação seca (biomassa do piquete / quantidade de cabeças de gado / consumo médio de capim por indivíduo por dia);

6) Cálculo do tempo que o gado permanecerá fora do piquete na estação seca (taxa de crescimento médio da espécie de capim / unidade de área);

Proposta III: Alimentação de serpentes cativas (Proposta selecionada)

Contextualização

A manutenção de serpentes em cativeiro é uma atividade que demanda muita organização e atenção com os animais, especialmente com sua alimentação, já que em animais silvestres cativos o estresse é um fator determinante pro sucesso da criação. A alimentação dos animais precisa seguir um rígido cronograma, pois além das serpentes se alimentarem em intervalos de tempo grandes (15 a 30 dias entre uma alimentação e outra), as presas (camundongos) normalmente são fornecidas por meio de doação por instituições de pesquisa (p.ex. Biotérios de Universidades, Fiocruz). Diante disso, faz-se necessário o envio de um cronograma prévio para tais instituições com o planejamento da quantidade de camundongos necessária em um determinado intervalo de tempo, que seja suficiente pra suprir as necessidades alimentares de todas as serpentes do plantel. A quantidade de camundongos fornecida a cada serpente irá variar de acordo ao seu próprio peso, sendo sugerido, pela literatura, cerca de 15 a 30% do peso da serpente em alimento. A frequência da alimentação também pode variar entre as diferentes espécies (peçonhentas e não peçonhentas) e entre os animais que estão doentes ou saudáveis. A alimentação das peçonhentas precisa seguir o cronograma de extração de veneno, onde a alimentação precisa ocorrer 15 dias após a extração, sendo portanto uma vez por mês. No caso das não peçonhentas (em sua maioria constritoras), a alimentação pode ser mensal ou quinzenal, a depender da demanda energética da espécie. Animais que estão doentes na quarentena precisam de maior demanda energética, se alimentando quinzenalmente, enquanto que os que estão saudáveis podem se alimentar apenas uma vez por mês. Assim, um rígido controle do cronograma de alimentação é fundamental para o sucesso da criação. O controle da alimentação poderá seguir a seguinte especificação:

Quarentena + peçonhenta = 10% de alimento a cada 15 dias;

Quarentena + não peçonhenta = 15% de alimento a cada 15 dias;

Serpentário + peçonhenta = 20% de alimento a cada 30 dias;

Serpentário + não peçonhenta = 30% de alimento a cada 30 dias;

Planejamento da função

- Objetos e argumentos de entrada

O usuário deve fornecer:

#Input

data.frame

Um data frame que conterá, nas linhas, cada espécime de serpente, e nas colunas, a) o peso do espécime de serpente; b) em qual local ela está (serpentário = SER ou quarentena = QUA); c) se ela tem veneno (S) ou não (N) e d) a frequência de alimentação (mensal = M ou quinzenal = Q). # É preciso saber previamente o peso médio (g) de cada camundongo oferecido na alimentação (pmc). Para tal, deverá ser criado um objeto chamado “pmc” contendo este valor.

#Output

#O mesmo data frame da entrada, porém acrescido de uma coluna que conterá o número de camundongos necessários para cada um dos espécimes de serpentes. A função calculará, para cada indivíduo, uma determinação alimentar específica (se 10, 15, 20 ou 30% do seu peso em alimento).

Pseudocódigo

```
Nome da funcao <- function (data.frame) #Aqui criaremos a funcao que eu
chamarei de “alimento.cobra“ com base no data frame chamado “cobra”.

#MENSAGEM DE ALERTA! Antes de iniciar, precisamos verificar se existe alguma
informacao faltante (NA). Caso haja, nao poderemos simplesmente eliminar a
linha, mas sim, devemos preenche-la com a informacao que esta faltando
(peso, por exemplo), pois nenhum animal pode ficar sem se alimentar.

  if (anyNA(cobra) == TRUE) #Indicando que caso haja alguma linha contendo
NA no data frame que foi inserido, a funcao sera interrompida e uma mensagem
aparecera com a informacao abaixo:
  {
    stop ("Preencha a informacao faltante, pois todos os animais precisam
ser alimentados!")
  }
pmc <- "peso medio do camundongo" #Preciso fornecer para a funcao um objeto
contendo um valor medio do peso dos camundongos (em gramas).

for (i in 1: nrows)     #Como precisamos fazer um cálculo individualizado
para cada espécime, será gerado um ciclo de repetições para cada linha
(espécime).

{

#Primeiramente precisamos saber onde o especime esta (quarentena ou
serpentario), para entao sabermos se a alimentacao sera mensal ou quinzenal.

  if (cobra$local[i] == "QUA" && cobra$veneno[i] == "S") #Se o especime
estiver na quarentena (alimentacao quinzenal) e se for peconhenta,
forneceremos apenas 10% de alimento.
  {
  result <- round((cobra$peso[i] * 0.10)/pmc)*2 #Para calcular quantos
camundongos o especime devera comer, iremos multiplicar o peso da serpente
pela porcentagem de alimentacao (10%), e dividiremos esse valor pelo peso
medio dos camundongos (pmc). O total sera multiplicado por 2, já que a
alimentação deverá ser quinzenal. Sera necessario usar o argumento “round“
para arredondar os valores equivalentes ao numero de camundongos. Iremos
criar o vetor ‘result’ para guardado o valor deste calculo.
}

  if (cobra$local[i] == "QUA" && cobra$veneno[i] == "N")  #Ja se o especime
estiver na quarentena e nao for peconhenta, forneceremos 15% de alimento.
  {
  result <- round((cobra$peso[i] * 0.15)/pmc)*2  #Vetor onde sera guardado o
valor do calculo.
}

  else #Caso o especime nao esteja na quarentena, então necessariamente ele
estara no serpentario.
  {
    if (cobra$local[i] == "SER" && cobra$veneno[i] == "S") #Se estiver no
```

```
serpentario e se for peconhenta,
    forneceremos 20% de alimento.
    {
      result <- round((cobra$peso[i] * 0.20)/pmc) #Aqui nao eh necessario
multiplicar por 2, ja que a alimentaçao, neste caso, sera mensal.
    }
    if (cobra$local[i] == "SER" && cobra$veneno[i] == "N") #Se estiver no
serpentario e nao for peconhenta, entao iremos fornecer 30% de alimento.
    {
      result <- round((cobra$peso[i] * 0.30)/pmc)  #Vetor onde sera guardado
o valor do calculo.
    }
  }
  cobra$quantidade[i] <- result #Vamos criar a coluna "quantidade" dentro do
data frame "cobra" que contera todos os resultados (previamente guardados no
vetor “result“) da quantidade de camungondos que devera ser forncecido a
cada especime de serpente.
}

  return (cobra)  #Aqui, a funcao ira me retornar o mesmo data frame que eu
inseri no input, porem agora com a coluna "quantidade", contendo o numero de
camundongos necessario para cada especime.
}
```

Código da função:

```
alimento.cobra <- function (cobra)

{

  if (anyNA(cobra) == TRUE)
  {
    stop ("Preencha a informacao faltante, pois todos os animais precisam
ser alimentados!")
  }
  pmc <- 300
  for (i in 1:10)
  {
    if (cobra$local[i] == "QUA" && cobra$veneno[i] == "S")
    {
      result <- round((cobra$peso[i] * 0.10)/pmc)*2
    }
    if (cobra$local[i] == "QUA" && cobra$veneno[i] == "N")
    {
      result <- round((cobra$peso[i] * 0.15)/pmc)*2
    }
    else
    {
      if (cobra$local[i] == "SER" && cobra$veneno[i] == "S")
      {
```

```
        result <- round((cobra$peso[i] * 0.20)/pmc)
      }
      if (cobra$local[i] == "SER" && cobra$veneno[i] == "N")
      {
        result <- round((cobra$peso[i] * 0.30)/pmc)
      }
    }
    cobra$quantidade[i] <- result
  }
  return (cobra)
}
```

Help

```
alimento.cobra                                    package: -
R Documentation

ALIMENTAÇÃO DE SERPENTES DE CATIVEIRO

Description:

Função para calcular a quantidade individual de alimento (camundongos)
necessário para diferentes espécies de serpentes que encontram-se em
ambiente cativo. São usados, como base para o cálculo, o peso médio da
espécie de camundongo oferecida como alimento e o peso individual (g) de
cada serpente. A função retorna o mesmo data frame fornecido inicialmente,
porém, acrescido de uma coluna que indicará a quantidade de camundongos
necessária para cada espécime de serpente, de acordo com sua frequência de
alimentação.

Usage:

alimento.cobra (data.frame, pmc)

Arguments:

Data frame: contém as informações provenientes de cada espécime de serpente
que deverá ser alimentada. Esse argumento de entrada precisará conter
obrigatoriamente as colunas espécime (coluna que conterá todos os indivíduos
que precisarão receber alimento), local (coluna que deverá conter o local em
que cada espécime de serpente estiver alocada - quarentena ou serpentário),
veneno (coluna que deverá conter a informação sobre a presença (S) ou não
(N) de peçonha em cada espécime de serpente) e peso (coluna com número
inteiro que define o peso de cada espécime de serpente, em gramas).

pmc: número inteiro que define o peso médio da espécie de camundongo
fornecida na alimentação, em gramas.

Details:

O objeto ”pmc” deverá ser obrigatoriamente fornecido para a execução da
```

```
função.

A função não poderá conter NAs (caso haja, eles precisam ser substituídos
pela informação correta), pois nenhuma serpente poderá ficar sem se
alimentar.

Value:

O mesmo data.frame de entrada, porém acrescido da coluna (“quantidade”)
contendo a quantidade necessária de camundongos para cada espécime de
serpente. Os resultados contidos nessa coluna serão proveniente do vetor
“result” (vetor criado para armazenar o resultado do cálculo da quantidade
de alimento necessário para cada indivíduo).

Warning:

Se houver algum NA no data frame de entrada a função não será executada.

Autor(s):

Daniela Pinto Coelho

danipcoelho@gmail.com

Examples:
Para rodar a função é necessário baixar a tabela (data frame) "cobra" que
contém as informações sobre os espécimes de serpente.
cobra <- read.table("cobra.csv", header=T, sep=";")
alimento.cobra (cobra, pmc=300)
```

Banco de dados fornecido para função cobra.csv

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:danipcoelho:start

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