

Cecili Barrozo Mendes



Bióloga, Mestre em Ciências Marinhas Tropicais e Doutoranda em Genética e Biologia Evolutiva - IBUSP

Trabalho atualmente com taxonomia e genética de populações de invertebrados marinhos, especialmente nemertíneos. O título do meu projeto de doutorado é *Nemertopsis bivittata* (NEMERTEA) e *Perinereis ponteni* (POLYCHAETA:ANNELIDA): VALIDADE DAS ESPÉCIES, FLUXO GÊNICO E DIVERSIDADE GENÉTICA NO LITORAL BRASILEIRO, orientado pela Profa. Dra. Sônia Andrade

Meus Exercícios

Link para página com meus exercícios resolvidos [exercícios](#)

Trabalho Final

Proposta A

Função mares

Contextualização

Maré é o nome dado a mudança rítmica no nível do mar, sendo mais perceptível na costa (Soares-Gomes & Figueredo, 2009). Essa variação ocorre em ciclos diários, podendo ser observadas pelo menos uma maré baixa e uma alta a cada 12 horas. As marés são influenciadas principalmente pelos movimentos gravitacionais da lua e do sol, além dos movimentos rotacionais da lua, do sol e da terra. Além dos ciclos diários, as marés também possuem ciclos mensais, onde marés de maiores amplitudes (marés de sizígia) são observadas a cada quinze dias, em média. Isto ocorre quando há conjunção das forças gravitacionais do sol e da lua, durante os períodos de lua cheia e lua nova. Em períodos de lua minguante e crescente a posição da lua fica em 90 graus em relação ao sol, cancelando parcialmente os efeitos gravitacionais de ambos e dando origem a marés de menor amplitude (marés de quadratura). As marés de sizígia são de grande importância para biólogos marinhos e pescadores, assim como para outros profissionais que lidam com o mar, pois durante esse período maiores porções da costa ficam expostas por mais tempo durante a maré baixa. A alta da maré também é importante, pois durante esse período peixes de maior porte conseguem se aproximar mais da costa, facilitando sua pesca. Os dias e horários específicos das marés são calculados anualmente pela marinha do Brasil e disponibilizados em tábuas de marés para pontos específicos da costa brasileira, podendo ser acessados através do website <https://www.marinha.mil.br/chm/dados-do-segnav/dados-de-mare-mapa>. Esse acesso, entretanto, pode ser demorado caso o usuário necessite dos dados de maré para diferentes localidades, pois é necessário navegar por um mapa e clicar para realizar o download da tábua de marés de cada uma

das localidades. Assim, a função `mares` tem como objetivo otimizar a busca por dados de marés para diferentes localidades em um mesmo período, bem como a busca por alturas de maré específicas em cada período mensal, trimestral ou semestral.

Planejamento da função

Argumentos de entrada (`local`, `periodo`, `mare`)

`local` = Nomes das localidades ou localidade para onde se deseja consultar a tábua de marés, de acordo com os locais disponíveis que serão especificados no help da função

`periodo` = Mês, trimestre ou semestre para o qual deseja a previsão de marés, de acordo com o especificado no help da função

`mare` = Altura da maré desejada. Esse argumento é opcional, caso deseje os valores de previsão para marés baixas, o usuário deve colocar o valor máximo de maré desejado (p.ex. `max 0.5`). Caso deseje valores para marés altas, o usuário deve colocar o valor mínimo desejado (p.ex. `min 1.4`). Caso o argumento não seja utilizado, a função usa o padrão de busca para maré baixa máxima de 0.3, valor geralmente utilizado para planejamento de expedições de coleta marinhas.

Verificando os parâmetros

- Item de lista não ordenada Os nomes das localidades estão corretos? Se não constarem na lista de localidades disponíveis retorna “Revise as localidades, uma ou mais não correspondem a localidades disponíveis para busca”
- Item de lista não ordenada O período especificado está dentro esperado? Caso seja diferente do esperado, retorna “Revise o período, não se encaixa no esperado”
- Item de lista não ordenada O argumento `mare` foi utilizado? Se não, retorna “Nenhuma altura de `mare` foi especificada, buscando mares iguais ou menores que 0.3”
- Item de lista não ordenada O argumento `mare` foi utilizado, mas < -0.1 ou > 1.5 , retorna “Altura de `mare` além das registradas para o Brasil, revise”

Pseudocódigo

1. Modificar a URL base recursivamente para baixar o pdf de cada localidade solicitada
2. Converter os arquivos .pdf baixados para .txt para facilitar a busca de padrões
3. Fazer o subset dos arquivos .txt de acordo com o período solicitado

```
3.1 Se solicitado semestre, fazer o subset para o semestre escolhido e guarda em um novo objeto
```

```
3.2 Se solicitado trimestre, fazer o subset para o trimestre escolhido e guarda em um novo objeto
```

```
3.3 Se solicitado mês, fazer o subset para o mês escolhido e guarda em um novo objeto
```

4. Dentro do subset escolhido, fazer o subset para datas com marés com altura solicitada

```
4.1 Se não houve maré especificada, buscar marés  $\leq 0.3$  e guarda esse subset em um data.frame
```

4.2 Se a maré especificada for max ..., buscar marés $\leq$ o valor especificado e guarda esse subset em um data.frame

4.3 Se a maré especificada for min ..., buscar marés $\geq$ o valor especificado e guarda esse subset em um data.frame

Saída

Retorna o data.frame com as datas de maré separadas por localidade

Referências

Soares-Gomes, A. & Figueredo, A.G. O ambiente marinho. In: Pereira, R.C & Soares-Gomes (org.). Biologia Marinha. Rio de Janeiro: Interciência, 2009. p. 1-33.

— [Alexandre Adalardo de Oliveira](#) 2018/05/24 09:15

Ajustamos a propostas para apenas uma localidade devido aos problemas em acessar dados de pdf. Vamos considerar a propostas dessa forma.

Oi Cecili,

Gostei da proposta, acho que você vai ter um trabalhinho considerável em extrair essas infos e acho que há elementos de ciclos e escolhas interessantes para a sua função. Bota ela pra rodar e procura a gente se tiver dúvidas, ok?

Beijo

Rena

renata.orofino@gmail.com

Proposta B

Função pokecandies

Contextualização

Pokémon go é um jogo de realidade aumentada desenvolvido para smartphones. O jogo é baseado na franquia Pokémon, originária do Japão, que é centrada na ideia de um mundo onde existem criaturas, pokémons, que são capturadas por humanos, os treinadores, para serem treinados e batalharem em ginásios. Alguns pokémons podem evoluir para outras formas em determinadas condições, dependendo do jogo da franquia. Em todos os jogos, entretanto, a evolução depende de uma determinada quantidade de doces que são específicos para cada tipo de Pokémon, os pokémons que evoluem com 12, 25, 50, 100 ou 400 doces. No jogo Pokémon go existem três formas de conseguir os doces necessários para evolução de cada pokémon, ao chocar um ovo com o pokémon, ao capturar o pokémon no campo ou ao caminhar com o pokémon como companheiro. A quantidade de doces recebidos ao caminhar é relativa ao tipo de pokémon, já a quantidade de doces de captura é fixa (3 doces), podendo ser aumentada com o uso de doces caxi (6 doces). Assim, a função pokecandies tem

como objetivo calcular a quantidade de pokémons precisam ser capturados, com ou sem doces caxis, para se atingir a quantidade de doces necessária para evoluir o pokémon para o próximo estágio. Além disso, calcula quantos quilômetros devem ser caminhados para conseguir os doces necessários. Ainda, caso o jogador inclua a quantidade de doces que já possui, a função desconta esse valor.

Planejamento da função

Argumentos de entrada: (tabela,...)

tabela = data.frame que pode ter uma ou duas colunas. Na primeira coluna terá a lista dos pokémons para os quais o jogador deseja calcular a quantidade de doces. Na segunda coluna terá a quantidade dos doces já possuídos.

... = argumentos da função read.table

Carregar a tabela

Carrega a tabela com a função read.table, caso não sejam utilizados parâmetros da função read.table, carrega com as.is=T e os outros parâmetros padrão da função.

Verificando os parâmetros

- Item de lista não ordenada Verificar se o data.frame possui alguma linha com string que não bate com os pokemons suportados, caso tenha retornar “Sua tabela possui X pokemons desconhecidos, calculando para o restante” e faz um subset desse dataframe apenas com os pokemons esperados
- Item de lista não ordenada Verificar se o data.frame possui uma ou duas colunas. Se possuir apenas uma coluna, retorna “Você não forneceu a quantidade de doces inicial, calculando com quantidade máxima necessária”

Pseudocódigo

1. Comparar as tabelas com os nomes concatenados dos pokémons que evoluem com cada quantidade de doces

1.1 Comparar a tabela com os nomes concatenados dos pokémons que evoluem com 12 doces, se encontrar faz um subset desses pokes, atribui o valor 12 em uma nova coluna do data.frame

1.2 Compara novamente o data.frame, com os pokémons que evoluem com 25 doces, se encontrar faz um subset desses pokes, atribui o valor 25 na nova coluna

1.3 Compara novamente o data.frame, com os pokémons que evoluem com 50 doces, se encontrar faz um subset desses pokes, atribui o valor 50 na nova coluna

1.4 Compara novamente o data.frame, com os pokémons que evoluem com 100 doces, se encontrar faz um subset desses pokes, atribui o valor 100 na nova coluna

1.5 Compara novamente o data.frame, com os pokémons que evoluem com 400 doces, se encontrar faz um subset desses pokes, atribui o valor 400 na nova coluna

2. Se o data.frame possuir duas colunas, faz a subtração dos valores da terceira coluna pela terceira coluna e guarda esses valores na quarta coluna

2.1 Dentro deste controlador de fluxo, calcula a partir da quarta coluna do data.frame o valor de cada linha dividido por três e guarda em uma nova coluna nomeada como "Pokes a capturar"

2.2 Após isso, calcula a partir da quarta coluna o valor de cada linha dividido por seis e guarda esse valor em uma nova coluna nomeada "Pokes a capturar com caxi"

2.3 Faz o subset dos pokémons que ganham doce a cada 1km e repete o valor da quarta coluna em nova coluna nomeada "Km a percorrer"

2.4 Faz o subset dos pokémons que ganham doce a cada 3km e multiplica o valor da quarta coluna por 3 e coloca o resultado na coluna nomeada "Km a percorrer"

2.5 Faz o subset dos pokémons que ganham doce a cada 5km e multiplica o valor da quarta coluna por 5 e coloca o resultado na coluna nomeada "Km a percorrer"

3. Se o data.frame original possui apenas uma coluna, faz os mesmos cálculos do passo anterior, porém com base na segunda coluna, criada a partir do passo 1

Saída

Retorna o data.frame final

Oi Cecili,

Não entendo lhufas de pokémon e entendi a função, então, parabéns pela proposta.

Porém, não acho ela tão bacana, pois tem os passos muito mais simples que sua outra proposta.

Beijo
Rena

Oi Rena, vou seguir com a proposta A então e qualquer coisa peço socorro. Brigada!

Bjs ;)

Links para o trabalho final

Para o trabalho final segui com o plano A, conforme combinado. A função se mostrou mais desafiadora que o previsto, devido a falta de padrão na organização dos pdfs disponibilizados pela Marinha do Brasil, e por isso tive de realizar algumas modificações na proposta (justificativa para as mudanças encontra-se no link). A função final realiza a busca de datas de marés com altura, localidade e período do ano determinados pelo usuário e retorna uma tabela com as datas e horários onde a altura de maré está dentro dos parâmetros buscados. Essa tabela também é salva no diretório corrente do usuário.

Link para a página da função: [Funcao Mares](#)

Link para o help : [Help](#)

Link para justificativa : [Justificativa para mudança na proposta inicial](#)

From:

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

Permanent link:

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

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