

Hector Benatti



Médico Veterinário formado pela FMVZ - USP

Doutorando no Laboratório Doenças Parasitárias, Departamento de Medicina Veterinária Preventiva e Saúde Animal da Faculdade de Medicina Veterinária e Zootecnia da USP.

Tem como proposta para a tese definir o quanto a ação antrópica no ambiente no estado de SP influencia na saúde das capivaras e como isso tem influência na disseminação da Febre Maculosa Brasileira.

Meus Exercícios

link para [exec](#)

Proposta de Trabalho Final

Plano A

FUNÇÃO: tick.genus

CONTEXTO: carrapatos são ectoparasitas de distribuição mundial. Parasitam mamíferos (incluindo humanos), aves, répteis e anfíbios. Possuem grande importância na epidemiologia de doenças humanas e em animais. Entretanto, existe uma grande dificuldade na identificação de gêneros e espécies pelos profissionais da saúde. Sendo assim, a função em questão criará uma chave simplificada para a identificação dos três principais gêneros de carrapatos de importância médica e médica veterinária no Brasil. Basta o usuário analisar estruturas específicas e de fácil visualização do carrapato e, sem o uso de uma chave pictórica, mas apenas com a função, e ele chegará facilmente ao gênero do ectoparasita. Com o uso de uma lupa simples é fácil de analisar as estruturas nos indivíduos adultos que são utilizadas para o desenho da chave de gênero. Conhecendo as estruturas, torna-se simples fazer uso da função.

TIPO DE FUNÇÃO: esta função analisará logicamente cinco características que vão possibilitar que o usuário chegue de forma rápida no gênero de carrapato coletado, podendo assim tomar as devidas medidas profiláticas para o paciente do qual o parasita foi retirado. A função seguirá o seguinte desenho esquemático:



Analisando as características, o usuário chegará a um dos três gêneros de maior importância no Brasil (IXODES, RHIPICEPHALUS, AMBLYOMMA), caso seja de outro gênero de menor importância, devolverá OTHER. o usuário não precisa possuir maiores conhecimentos sobre a chave pictórica de gêneros.

ARGUMENTOS: os argumentos da função serão justamente as características da chave (modificada) de Harry Pratt (1961), que serão julgadas como TRUE ou FALSE:

Sulco anal na frente do ânus → groove

Segundo segmento do palpo sem projeção → palp.proj

Hipostômio longo → hipostomy

Base do capítulo hexagonal → hexag

Escudo com olhos → eyes

DADOS DE ENTRADA: informações serão vetores lógicos para cada um dos argumentos da função. Os argumentos que não receberem o julgamento, pelo default da função, serão considerados FALSE. A falta de informação levará o usuário ao OTHER, por não possuir acurácia na análise.

DADOS DE SAÍDA: ao final, a função devolverá um vetor atômico com o gênero do carrapato.

HELP: o help da função auxiliará o usuário na identificação das estruturas a serem observadas nos carrapatos e julgadas para que se tenha maior acurácia na identificação do gênero.

REFERÊNCIA: Manual de Vigilância Acarológica do Estado de SP, 2002

Oi Hector,

Eu entendi o apelo para a criação da função, mas não consigo ver a necessidade de haver uma função pra isso. Você resolve o problema com a figura da proposta, não?

Agora pensando na estrutura da função, acho que precisamos de mais alguma coisa nela. Consigo prever muitos "if" "else", mas só. Vamos repensar a estrutura dela?

* Se eu não tenho acesso a todas as características do bicho, há uma forma de calcular a probabilidade dele ser do gênero A, B ou C?

* Se eu te der informações de localidade, há uma probabilidade/certeza de que ele será do gênero A?

* Faz diferença saber se é larva, ninfa ou adulto? Tem como identificar bichos jovens?

* Se eu tenho uma infestação, ela será de uma única espécie/gênero? Poderíamos pedir pro usuário entrar com uma amostra coletada e jogamos pra ele quais gêneros estão lá? Ou seja, o cara entra com um vetor com as características de 10 bichos, vc identifica os 10 de uma vez e fala pra ele se é uma infestação de um só gênero ou de mais de um gênero. (Não sei se faz sentido, pq não sou dessa área, veja se falei algo coerente).

Vc precisaria pensar bastante sobre ela e me fazer uma contra-proposta, pois como ela está eu não fiquei convencida, ok?

Bjs —[Rena](#)

Continuando nossa conversa, pra deixar tudo registrado na wiki.

Entendo a dificuldade com números. Acho que há muitos exemplos de funções que não

estão trabalhando números ou estatísticas e podem servir de inspiração. A da [Patrícia, a minha, a do bandeirão do Guilherme...](#)

Sobre eu comentário: “Não sei se ainda é o necessário para que a função fique ao gosto dos avaliadores, mas ainda acredito que terá grande valia.” Seria legal se sua função não fosse composta apenas por if e else encadeados. Por isso minha sugestão de tentar calcular a probabilidade de ser o gênero a ou b a partir de menos características, por exemplo.

Gosto das suas ideias do email: “Colocar a possibilidade (em escala, pois não existem dados numéricos exatos sobre as probabilidades) do gênero pela espécie de animal doméstico (equinos, bovinos, cães ou gatos) de onde o carrapato foi retirado. Definir a possível espécie de acordo com o hospedeiro.”

E, de novo, sugestões de quem não é da área:

Conforme a espécie de carrapato ou do hospedeiro, o tratamento é diferente? O peso do hospedeiro influencia o tratamento realizado? Podemos realizar uma conta simples como: o usuário indica as características do carrapato, do hospedeiro e o peso do hospedeiro e com base nisso vc sugere um remédio e uma dosagem? Ou uma linha de tratamento? Caso eu te dê o tamanho da infestação, vc calcula o tipo/duração do tratamento?

Vou pedir pra alguém ajudar a gente. Eu investiria nessa proposta, esquece a outra. Mas ainda precisamos deixa-la mais redondinha, ok?! Beijo!

Respostas Hector:

Concordo que seria extremamente simples apenas fazer uma função de 10 ou 20 linhas com IFs e ELSEs apenas para definir gênero e mais nada. Tenho pensado bastante em outras possibilidades de melhoria, e acho que vamos chegar lá!

Vou tentar me fazer claro, mas não sei se vai dar pra entender. Na minha cabeça as ideias estão claras, mas não sei se vou conseguir transmitir, mas lá vai:

- Primeiramente seguir a chave pictórica da proposta inicial tendo também a espécie do animal. Tendo consciência da espécie da qual o carrapato foi retirado, já da pra ter uma breve noção de um ou dois gêneros de carrapato que podem ser. No caso de faltar alguma informação para definir o gênero com mais precisão, a função terá um STOP e um PRINT: “falta informação” (sei lá... algo assim)
- Se o usuário não tiver a informação faltante, a função segue para outra abordagem: tendo ciência da espécie do hospedeiro, a função “perguntará” ao usuário características bem generalistas das prováveis espécies que acometeriam aquele animal (o corpo é arredondado? É alongado? O carrapato é avermelhado? É preto? Blá blá blá), para nos aproximarmos ainda mais do carrapato em questão. (aqui eu não sei exatamente como eu faria para a função devolver as perguntas para que o usuário insira novos argumentos. Acredito que tenha como fazer, só não sei como)
- Tendo a espécie do hospedeiro, o peso do animal, a “intensidade da infestação”, a possível espécie de carrapato, a função devolve ao usuário (na forma de um texto conciso):

A possível espécie de carrapato (com a possibilidade que falamos antes – baixa, média, alta).

O princípio ativo a ser usado no tratamento (não posso usar nome comercial).

As formas de tratamento (como usar o medicamento e em que épocas do ano).

Prevenção de novas infestações.

Resposta Rena:

Oi Hector,

Vamos fechar esse trem pra você conseguir ter tempo de executar?

Acho que tá legal, só vou pedir pra você indicar direitinho:

Os objetos de entrada da função;

Os argumentos possíveis;

O caminho imaginário que a função faz nos diferentes casos (todos os dados morfológicos, alguns dados morfológicos + localidade, alguns dados morfológicos + hospedeiro, hospedeiro + localidade). Acho que podemos fechar nesses casos iniciais e se tiver fôlego, incluir mais opções.

Não acho que vale a pena você tentar fazer uma função interativa, pois pode ser um desafio grande para você. Tem como fazer (até pq a pergunta é sempre “como fazer” e não “se é possível fazer no R”), você vai encontrar exemplos na internet com facilidade. Caso consiga fazer, vai valer estrelinha ok? ;)

Os objetos de saída: Entendo que sejam estes - A possível espécie de carrapato (com a probabilidade associada - baixa, média, alta, certeza). O princípio ativo a ser usado no tratamento. As formas de tratamento. Prevenção de novas infestações.

Qq coisa, to aqui.

Proposta B

Contexto: comumente na pesquisa de campo, seja coleta de dados à campo, seja na coleta de dados via preenchimento de questionário (prática comum em algumas pesquisas da área de Medicina Veterinária), alguns dados de determinados indivíduos não são colhidos: por falta de estrutura material ou pessoal na hora da coleta, ou, no caso de questionários, por imperícia do público a quem as perguntas são aplicadas. Tendo consciência da carência de algumas informações, nem sempre podemos dizer que nossos dados representam a verdade, desta forma, pensei em uma função que realize testes de reamostragem a partir dos dados coletados para que possamos inferir sobre a precisão de nossas estimativas.

Ideia: uma função que faça análises de intervalos de confiança e gráficos de normalidade em vetores com NAs inferindo sobre os valores faltantes.

Objeto de entrada: um vetor 'integer', a coluna de um dataframe (df\$coluna) ou a coluna de uma

matriz. Ex: vetor.entrada ← sample(c(1:10, NA), size = 20, replace = T)

```
vetor.entrada
```

```
[1] 6 9 4 3 2 8 3 1 4 9 1 3 2 3 5 5 3 2 NA 10
```

Estrutura: - detecta quantos NAs tem no vetor.

- Faz um IF para ver qual método o usuário escolheu:

```
Se média, substitui os NAs pela média do vetor;  
Se mediana, idem  
Se normal, substitui os NAs por valores arbitrários baseados em rnorm com  
MEAN e SD relativos aos valores do vetor
```

Como funciona (mede o impacto de usar as estimativas de substituição dos NA):

- calcula o intervalo de confiança do vetor original e do vetor final.

- calcula o intervalo de confiança baseado no bootstrap do vetor original.

- mesma coisa pro vetor final. O tamanho da amostra no bootstrap tem que ser igual ao vetor final (já com NA substituído)

Return:

- Um dataframe com o intervalo de confiança do vetor original e com o intervalo de confiança do vetor final (aqui não consigo lançar mão de exemplos)

- dois gráficos (ainda não consegui definir qual) para que o usuário compare o original do final graficamente

Oi de novo,

Veja se entendi. Vc tem uma amostra de altura de x e há dados faltantes. Com base na média calculada para os dados existentes, vc vai preencher o valor faltante? Pode isso Arnaldo? Se puder, me pareceu uma proposta mais elaborada. Vc precisa explicar direito os objetos de entrada e saída para que fique mais fácil de julgar a viabilidade e o tamanho do desafio da função. Vamos conversando. Bjs

12/06

Oi Hector, tudo bom?

Agora que conseguimos a ajuda dos demais monitores, acho que estamos no caminho certo.

Bora começar a fazer essa função e definir quais gráficos seriam os melhores.

Qq coisa, o time todo está aqui pra te ajudar.

Abs, Rena

Função: na2value (proposta B)

```
##### NA2VALUE
#####

na2value <- function(vetor.entrada, metodo){ #nomeando a função e colocando
os objetos de entrada
  #### Verificando os Argumentos ####
  if(!is.vector(vetor.entrada)){ #conferindo se 'vetor.entrada' é mesmo um
vetor
    stop("0 argumento 'vetor.entrada' só aceita vetores.") #parando a
função caso não seja
  }
  if((metodo %in% c('media', 'mediana', 'sorteio')) == F){ #conferindo se
'metodo' é uma das opções disponíveis
    stop("Selecione um método disponível para a alteração dos NAs")
#avisando o usuário caso o haja algum problema no método selecionado
  }
  #### Verificando parametros e definindo o novo vetor ####
  na.entrada <- sum(is.na(vetor.entrada)) #detecção da quantidade de NAs
no vetor de entrada
  if(na.entrada == 0){ #verificando se há NAs no vetor
    stop('Seu vetor não possui nenhum NA') #parando a função caso não
haja NAs
  }
  media.entrada <- mean(vetor.entrada, na.rm = T) #criação do objeto da
média do vetor original sem os NA

  mediana.entrada <- median(vetor.entrada, na.rm = T) #criação do objeto
da mediana do vetor original sem os NA
  dp.entrada <- sd(vetor.entrada, na.rm = T) #criação do objeto do desvio
padrão do vetor original sem os NA
  novo.vetor <- vetor.entrada #criação de um novo vetor para usar daqui
para frente. O original será mantido
  #### Redefinindo os NAs ####
  if(metodo == "media"){ #caso o usuário escolha a substituição de NA por
média
    novo.vetor[is.na(novo.vetor)] <- media.entrada #substitui NA por
média do vetor
  }else if(metodo == "mediana"){ #caso o usuário escolha a substituição de
NA por mediana
    novo.vetor[is.na(novo.vetor)] <- mediana.entrada #substitui NA
por mediana do vetor
  }else if(metodo == "sorteio"){ #caso o usuário escolha a
```

```

substituição de NA por valores aleatórios baseados na média e desvio padrão
do vetor original
      novo.vetor[is.na(novo.vetor)] <- round(
#escolherá valores aleatórios
      sample(      #de uma amostra de tamanho determinado
        rnorm(10000, mean = media.entrada, sd = dp.entrada),
#em que os valores serão de uma distribuição normal baseadas na média e
desvio padrão do vetor original
        na.entrada, replace = T),
#ocupando o lugar dos NA e com reposição
        3)      #aqui escolho o
número de casas decimais
      }
      ##### Calculando os intervalos de confiança e comparando os Vetores #####
      error.novo <- qnorm(0.975)*sd(novo.vetor/sqrt(length(novo.vetor)))
#cálculo do erro amostral para o novo vetor
      min.novo <- mean(novo.vetor) - error.novo #determinação do valor mínimo
do intervalo de confiança (média - erro amostral) do novo vetor
      max.novo <- mean(novo.vetor) + error.novo #determinação do valor máximo
do intervalo de confiança (média + erro amostral) do novo vetor
      ic.novo <- c(min.novo, mean(novo.vetor),max.novo) #determinação do
intervalo de confiança do novo vetor
      error.amostra <-
qnorm(0.975)*sd(vetor.entrada[!is.na(vetor.entrada)]/sqrt(length(vetor.entr
ada[!is.na(vetor.entrada)])) #cálculo do erro amostral para o vetor
original, descartando os NA
      min.amostra <- mean(vetor.entrada[!is.na(vetor.entrada)]) -
error.amostra #determinação do valor mínimo do intervalo de confiança (média
- erro amostral) do vetor original, descartando os NA
      max.amostra <- mean(vetor.entrada[!is.na(vetor.entrada)]) +
error.amostra #determinação do valor máximo do intervalo de confiança (média
+ erro amostral) do vetor original, descartando os NA
      ic.amostra <- c(min.amostra,
mean(vetor.entrada[!is.na(vetor.entrada)]),max.amostra) #determinação do
intervalo de confiança do vetor original, descartando os NA
      #confeccionando o data.frame com o qual o ggplot2 consegue lidar
      wide <- data.frame(valores = rbind(ic.novo, ic.amostra, deparse.level =
0),
      grupo = c('novo', 'amostra'))
      library('ggplot2', quietly = T) #carregando o ggplot2
      comparacao <- ggplot(wide)+ #inserindo o data.frame no qual vamos
trabalhar
      aes(x = grupo, y = valores.2)+ #definindo os eixos do gráfico
      geom_point()+ #plotando as médias
      geom_errorbar(aes(ymin = valores.1, ymax = valores.3), width = .1)
#plotando as barras de erro
      comparacao
      ##### Preparando a saída #####
      saida <- list(novo.vetor = novo.vetor, #criando a saída da função
      grafico = comparacao) # colocando na saída o novo vetor e
a comparação gráfica entre o antigo e o novo

```

```
return(saida) #retornando os resultados da função  
}
```

HELP: na2value

na2value package: unknown R
Documentation

Description:

na2value consiste na substituição dos NAs de um vetor inicial por valores pré-determinados para futuras análises estatísticas. A função substituirá os NAs de acordo com a escolha do usuário: pela média do vetor, pela mediana do vetor ou por valores aleatórios (rnorm) baseados na média e no desvio padrão do vetor original. O usuário poderá escolher qual dos métodos de substituição mais se adequa à necessidade a partir da comparação entre o intervalo de confiança do vetor original e após a substituição dos NAs, e da análise visual gráfica do IC do vetor original e do novo vetor.

Usage:

```
na2value(vetor.entrada, método)
```

Arguments:

vetor.entrada → um vetor integer, a coluna de um dataframe ou de uma matriz.
método → media, mediana, sorteio

media: os NAs do vetor são substituídos pela média do vetor original.

mediana: os NAs do vetor são substituídos pela mediana do vetor original.

sorteio: os NAs do vetor são substituídos por um valor aleatório baseado na distribuição normal de acordo com a média e desvio padrão do vetor original.

Value:

Retorna o comparativo entre o vetor original e o novo vetor com os NAs substituídos, além de dois gráficos de intervalo de confiança para comparação visual.

Warning:

No caso do usuário não escolher nenhum dos métodos disponíveis, a função dará erro.

Author:

Hector Ribeiro Benatti

References:

Nakagawa, Shinichi, and Robert P. Freckleton. "Missing inaction: the dangers of ignoring missing data." *Trends in Ecology & Evolution* 23.11 (2008): 592-596.

Examples:

```
# criando um vetor com NAs

vetor.teste <- sample(c(1:100, rep(NA, 10)), 1000, replace = T)
mean(vetor.teste, na.rm = T)
(sum(is.na(vetor.teste))/length(vetor.teste))*100

# Conferindo o funcionamento da função ...

#... usando a media
resultado1 <- na2value(vetor.entrada = vetor.teste, metodo = 'media')
sum(is.na(resultado1$vetor))
resultado1$grafico

#... usando a mediana
resultado2 <- na2value(vetor.entrada = vetor.teste, metodo = 'mediana')
sum(is.na(resultado2$vetor))
resultado2$grafico

# ... usando o sorteio
resultado3 <- na2value(vetor.entrada = vetor.teste, metodo = 'sorteio')
sum(is.na(resultado3$vetor))
resultado3$grafico

# Testando um vetor sem NAs
vetor.sem.na <- rnorm(100)
resultado4 <- na2value(vetor.sem.na, 'media')

# Testando o uso de um método não disponível
resultado5 <- na2value(vetor.teste, 'median')
```

Arquivos

[trabalho_final_hector.r](#)

[help_da_funcao.docx](#)

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

Permanent link: 
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:hector.benatti:start

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