Filipa Abreu

Doutoranda no Programa de Etnobiologia e Conservação da Natureza - UFRPE

Desenvolve trabalhos na área cognição de primatas não-humanos, particularmente no uso da memória espacial durante o forrageio e nos fatores ecológicos que influenciam essa capacidade. Integrante do Laboratório de Etologia Teórica e Aplicada. LETA

Exerc.

Proposta A

Contextualização: Em estudos de comportamento animal é comum avaliar o tempo que os animais demoram para aprender determinado comportamento ou atividade. Sendo que por vezes é necessário fazer uma comparação desse processo em diferentes contextos (por exemplo, em situações de alta ou baixa disponibilidade de alimento ou na presença de animais com diferentes graus de parentesco). Este cenário pode também ser observado em outras áreas, como por exemplo, na comparação do tempo de crescimento de organismos (vegetais ou animais) em diferentes condições.

Tarefa da função: A função terá como objetivo comparar a taxa de variação média entre dois pontos em duas curvas e medir a probabilidade de estas curvas serem iguais.

Entrada dos dados:

- Data frame com uma coluna categórica e duas colunas numéricas. A coluna categórica conterá um fator com dois níveis (as condições ou cenários que se deseja comparar), e as colunas numéricas terão uma variável medida em tempo e a variável de interesse que será em função da primeira variável numérica.
- Pontos inicial e final em que se pretende comparar a taxa de variação média que devem ser escolhidos pelo utilizador. Os pontos escolhidos terão de ser os mesmos nas duas curvas.

Saída dos dados:

- Gráfico plot com duas curvas plotadas e suas retas secantes, mostrando os ponto final e inicial da taxa de variação média escolhidos pelo utilizador.
- Valor da probabilidade.

```
Obrigada pela ajuda. Isso, os pontos entre os quais o utilizador pretende
calcular a variação média serão informados pelo utilizador em um
argumento. Optei por colocar isso no argumento, uma vez que como as curvas
```

podem ser tanto crescentes como decrescentes os pontos seriam diferentes e colocar pontos como default iria ficar confuso. Assim, quando o usuário entra com os dados ele terá que ter uma noção entre quais pontos ele quer que a comparação seja realizada. Com relação ao cálculo de valor da probabilidade, poderia ser feito com um teste de Monte Carlo (com 2000 simulações), isto porque na área em que trabalho é normal os dados não atingirem os pressupostos de normalidade e homogeneidade.

Comentários Julia

Oi Filipa

Me parece bastante interessante e de ampla ut}}ilização esta função. Mas não tenho certeza de que compreendi bem. A taxa de variação média escolhida pelo utilizador seria um dado em um argumento ?

Se achar que pode ser valido tente incluir um esboço como exemplo dessas curvas apenas para nos auxiliar a compreender melhor e pensar como podemos aprimorar a proposta

O cálculo da probabilidade seria feito com base em um teste(s) específico (s) ?

Proposta B

Contextualização: Em estudos sobre lateralidade em primatas humanos e não humanos, é comum esta lateralidade ser avaliada através de um uso preferencial de uma mão. Esta avaliação é realizada por meio de um índice simples mas que demanda tempo e é realizada em vários passos, visto que é feito por animal e por cada mão escolhida para realizar determinada ação. Esse índice de lateralidade é calculado através da formula (HI = (D-E) / (D+E)), em que D e E representam a frequência no uso da mão direita e esquerda, respectivamente) e varia de -1 (uso exclusivo da mão esquerda) a +1 (uso exclusivo da mão direita).

Tarefa da função: A função proposta terá como objetivo calcular o índice de lateralidade e avaliar a probabilidade de um dado animal ser destro, canhoto ou ambidestro e a probabilidade da lateralidade observada na população ser atribuída ao acaso (isto é, ter uma preferência geral por uma das mãos).

Entrada dos dados:

- Data frame contendo nas linhas as observações do uso da mão por ID do animal (fator que conterá x níveis de acordo com o número de animais introduzidos) e na coluna a mão usada em cada acção (A – ambidestro, D – direita, E – esquerda).

Saída dos dados:

- Uma lista com os seguintes parametros para cada animal: Frequência do uso de cada mão (A,

D, E), resultado do índice de lateralidade, classificação em categorias de acordo com esse índice e o valor da probabilidade.
- Valor da probabilidade para a população.

Obrigada pela ajuda. Realmente acho que acabei me expressando de forma confusa. Pensando melhor a função poderia incluir mais argumentos. A função pode permitir que o data frame de entrada inclua um fator que pode conter vários níveis, cuja influência sobre a lateralidade o usuário tivesse interesse de analisar (isto é, comparar o índice entre dois tipos de alimento, entre quantidades diferentes de alimento ou tarefas de diferentes níveis de dificuldade).

Outro argumento que pode ser incluído é a escolha do usuário sobre a comparação dos diferentes níveis do fator. Assim, o argumento “comp” teria como default FALSE. Nesse caso não haveria nenhuma análise entre os diferentes níveis do fator. Ou caso o usuário quisesse essa comparação, o argumento “comp” teria de ser preenchido com o nome da coluna do fator apresentada no “head” do data frame.

Dessa forma, por padrão, a função irá calcular a frequência do uso de cada mão para cada indivíduo, o índice de lateralidade (através da formula apresentada), a sua classificação de acordo com o índice, e a probabilidade do índice de cada individuo e da população serem observadas ao acaso através simulações de Monte Carlo. Ainda calculará a probabilidade da diferença entre os grupos ser significativa através do mesmo método, se o argumento “comp” for preenchido com o nome do fator a ser comparado.

Comentários Julia

Filipa, Acho que aqui também faltou elaborar quais seriam os argumentos possíveis. A principio a função parece um pouco simples demais mas os argumentos poderiam conferir maior importância a ela.

Assim como coloquei para a proposta A, é importante que ao falar de probabilidades fique um pouco mais claro como este cálculo é realizado. Quando você diz a probabilidade de que a lateralidade da população seja atribuída ao acaso, seria na verdade avaliar se existe uma lateralidade prevalente significativamente diferente da ocorrência das demais ?

Acho que ambas as propostas podem ser viáveis mas precisam de mais um pouco de detalhamento talvez e elaboração dos argumentos para poder seguir.

Comentários Julia 09/junho

Oi Filipa,

Ficaram mais claras as propostas após a edição e com seus comentários. Acredito que ambas sejam viáveis.

Dois comentários:

- Você está segura de como irá realizar os cálculos probabilísticos ?

- Para a proposta B concordo que seja melhor não ter um padrão determinado que a função irá buscar, e sim ler quais tipos de

classificação existem no dado de entrada (no exemplo que deu seria : D, E e A) e trabalhar com eles. Acho que a proposta B talvez ainda esteja um pouco simples.

- Para a proposta A: Ainda não acho que eu compreendi perfeitamente como serão feitos os cálculos mas entendi o objetivo. Algum outro monitor ou o Ale ainda devem passar aqui para comentar. Talvez essa proposta seja mais interessante analiticamente e em código também, focaria nela. De qualquer modo seria util pensar que tipo de possibilidades em argumentos, seria interessante o usuário ter. Por exemplo: é o mesmo tipo de cálculo que você utiliza para uma curva bimodal, curva exponencial, etc... ? Talvez possa ser esta uma flexibilidade da função para aumentar a sua aplicabilidade. Ou outros argumentos que você imagina que o usuário (voce por exemplo) acharia útil.

Qualquer coisa escreva, busque ajuda no forum ou no stackoverflow ;)

Beijos

Função Curly

```
curly<- function (dados, choosepoints=FALSE, xtype="name", ytype="name",
...)
{
  if (!(is.factor(dados[,1]) & is.numeric(dados[,2]) &
is.numeric(dados[,3]))) #vê se a primeira coluna do data frame é um fator e
a segunda e terceira um vector numérico
  {
    stop("WARNING: The first column has to be a factor and second and third
columns numeric vectors") #a função para e retorna um aviso em vermelho
  }
  else (is.factor(dados[,1]) & is.numeric(dados[,2]) &
is.numeric(dados[,3])) #vê se a primeira coluna do data frame é um fator e a
segunda e terceira um vector numérico
  {
    dados1 <- na.omit(dados) #remove as linhas do data frame que contenham
NA's
    cat("Lines in the data frame with NA's were excluded\n") #mensagem no
console dizendo o que foi feito essa exclusão.
    nivel <- levels(dados1[,1]) #mostrar quantos níveis tem o fator da
primeira coluna
    cat1 <- subset(dados1, dados1[,1]==nivel[1]) #fazer um conjunto de dados
em função do 1º nível do fator
    cat2 <- subset(dados1, dados1[,1]==nivel[2]) #fazer um conjunto de dados
em função 2º nível do fator
```

```
    #Neste passo, a ideia é criar um gráfico que possa ser utilizado pelo
usuário em qualquer local.
    #O uso do ..., faz com que o utilizador possa modificar os paramêtros do
gráfico a seu gosto
    pdf("grafico curva.pdf") #guarda o gráfico em PDF
    par(tck=0.01, cex.axis=1.3, family="serif") #alterando alguns parâmetros
do grafico
    plot(cat1[,3]~cat1[,2], xlab=xtype, ylab=ytype, ...) #fazer gráfico de
um dos conjuntos de dados e mudar os parâmetros do grafico nos argumentos da
função
    lines(lowess(cat1[,2], cat1[,3]), lwd=3) #inserir a linha para fazer a
curva de ajuste aos pontos
    par(new=TRUE) #incluir um outro gráfico (a ser realizado posteriormente)
na mesma figura
    plot(cat2[,3]~cat2[,2], xlab="", ylab="", xaxt="n", yaxt="n", ...)
#fazer gráfico de um dos conjuntos de dados e mudar os parâmetros do grafico
nos argumentos da função
    lines(lowess(cat2[,2], cat2[,3]), lty=2, lwd=3) #inserir a linha para
fazer a curva de ajuste aos pontos
    graphics.off() #apaga todos os gráficos abertos de forma a guardar em
pdf o gráfico acima
    #Neste caso, a ideia é plotar o grafico apenas para visualização do
próprio utilizador para que possa avaliar seus dados
    #e escolher depois os pontos de interesse para fazer a taxa de variação
média. Por essa razão não foi incluído ... aqui
    par(tck=0.01, cex.axis=1.3, family="serif") #alterando alguns parâmetros
do grafico
    graphic1 <- plot(cat1[,3]~cat1[,2], pch=20, cex=0.7, xlab=xtype,
ylab=ytype, cex.lab=1.5) #fazer gráfico de um dos conjuntos de dados
    lines(lowess(cat1[,2], cat1[,3]), lwd=3) #inserir a linha para fazer a
curva de ajuste aos pontos
    par(new=TRUE) #incluir um outro gráfico (a ser realizado posteriormente)
na mesma figura
    graphic2 <- plot(cat2[,3]~cat2[,2],  cex=0.7, xlab="", ylab="",
xaxt="n", yaxt="n") #fazer gráfico de um dos conjuntos de dados
    lines(lowess(cat2[,2], cat2[,3]), lty=2, lwd=3) #inserir a linha para
fazer a curva de ajuste aos pontos
  }
  if (choosepoints==FALSE) #caso o argumento seja igual a FALSE a função irá
fazer um teste qui-quadrado com todos os dados inseridos
  {
    test1 <- chisq.test(dados1[,2], dados1[,3]) #teste qui-quadrado
realizado em todos os dados de cada conjunto
    exit1 <- list(cat1, cat2, test1) #lista dos componentes que a função irá
retornar
    names(exit1) <- paste(list("Conjunto de dados do primeiro nível do
fator", "Conjunto de dados do segundo nível do fator", "Chi-Squared Test"))
#para atribuir um nome a cada componente da lista
    print("ATTENTION: The function compared only the first two levels of
your factor!!") #aviso para o utilizador
    return(exit1)   #retorna a lista acima
```

```
  }
  else (choosepoints==TRUE) #caso o argumento seja igual a TRUE a função
aguarda para que sejam escolhidos os pontos no gráfico
  {
    message("Please choose two points on the x-axis for calculation of
median variation rate") #aviso para o utilizador
    xi <- locator(1) #localizar o ponto inicial do eixo x para o calculo da
taxa de variação
    xf <- locator(1) #localizar o ponto final do eixo x para o calculo da
taxa de variação
    xi <- round(xi[[1]]) #arredondar o valor de x inicial escolhido pelo
utilizador
    xf <- round(xf[[1]]) #arredondar o valor de x final escolhido pelo
utilizador
    secante1 <- lowess(cat1[,3]) #saber quais os valores da curva ajustada
do primeiro conjunto de dados
    seci1 <- secante1$y[[xi]] #acessar valor do y correspondente ao valor de
x inicial escolhido pelo utilizador na curva ajustada (primeiro conjunto de
dados)
    secf1 <- secante1$y[[xf]] #acessar valor do y correspondente ao valor de
x final escolhido pelo utilizador na curva ajustada (primeiro conjunto de
dados)
    secante2 <- lowess(cat2[,3]) #saber quais os valores da curva ajustada
do segundo conjunto de dados
    seci2 <- secante2$y[[xi]] #acessar valor do y correspondente ao valor de
x inicial escolhido pelo utilizador na curva ajustada (segundo conjunto de
dados)
    secf2 <- secante2$y[[xf]] #acessar valor do y correspondente ao valor de
x final escolhido pelo utilizador na curva ajustada (segundo conjunto de
dados)
    lines(c(xi,xf), c(seci1, secf1), lwd=2, col="red") #traçar recta secante
do primeiro conjunto de dados
    lines(c(xi,xf), c(seci2, secf2), lwd=2, col="blue", lty=2) #traçar recta
secante do segundo conjunto de dados
    y1i <- cat1[cat1[,2]==round(xi),] #seleccionar as linhas do dataframe do
primeiro conjunto de dados que correspondem ao valor de x inicial para pegar
o valor de y correspondente
    y1f <- cat1[cat1[,2]==round(xf),] #seleccionar as linhas do dataframe do
primeiro conjunto de dados que correspondem ao valor de x final para pegar o
valor de y correspondente
    y2i <- cat2[cat2[,2]==round(xi),] #seleccionar as linhas do dataframe do
segundo conjunto de dados que correspondem ao valor de x inicial para pegar
o valor de y correspondente
    y2f <- cat2[cat2[,2]==round(xf),] #seleccionar as linhas do dataframe do
segundo conjunto de dados que correspondem ao valor de x final para pegar o
valor de y correspondente
    Ax1 <- (y1f[[3]]-y1i[[3]])/(xf-xi) #cálculo da taxa de variação média no
primeiro conjunto de dados
    Ax2 <- (y2f[[3]]-y2i[[3]])/(xf-xi) #cálculo da taxa de variação média no
segundo conjunto de dados
```

```
    aa1 <- cat1[cat1[,2]>=xi & cat1[,2]<=xf,] #para ter acesso ao data frame
dos valores do primeiro conjunto seleccionados pelo usuário
    aa2 <- cat2[cat2[,2]>=xi & cat2[,2]<=xf,] #para ter acesso ao data frame
dos valores do primeiro conjunto seleccionados pelo usuário
    intervalo1 <-aa1[[3]] #valores de y do primeiro conjunto (apenas no
intervalo seleccionado pelo usuário)
    intervalo2 <- aa2[[3]] #valores de y do segundo conjunto (apenas no
intervalo seleccionado pelo usuário)
    test <- chisq.test(intervalo1, intervalo2) #faz um teste qui-quadrado de
forma a comparar os valores das variváveis dependentes no intervalo de
tempo(x) escolhido pelo utilizador
    exit <- list(aa1, aa2, Ax1, Ax2, test) #lista dos componentes que a
função irá retornar
    names(exit) <- paste(list("Conjunto de dados do primeiro nível do fator
no intervalo escolhido", "Conjunto de dados do segundo nível do fator no
intervalo escolhido", "Taxa de variação média do primeiro conjunto de
dados", "Taxa de variação do segundo conjunto de dados", "Chi-Squared
Test"))
    print("ATTENTION: The function compared only the first two levels of
your factor!!") #aviso para o utilizador
    return(exit) #retorna a lista acima
  }
}
```

Help Curly

```
curly               package:unknown           R documentation

Teste de comparação de curvas de dois conjuntos de dados.


Description

Faz a comparação estatística entre as proporções ou porcentagens de dois
conjuntos de dados ao longo de um determinado período de tempo
e produz um gráfico de dispersão dos dados e suas curvas ajustadas.


Usage:

curly(dados, choosepoints=FALSE, xtype="name", ytype="name", ...)


Arguments:

dados - data frame com uma coluna categórica com 1 fator de vários níveis e
duas colunas numéricas.
A primeira coluna numérica deve conter a variável independente
```

```
e a segunda coluna numérica deve ser a variável dependente (em proporção ou
porcentagem).

choosepoints - se TRUE, devem ser escolhidos dois pontos para uma comparação
num determinado intervalo delimitado no eixo X.
Caso seja FALSE, a comparação será feita em toda a extensão do conjunto de
dados.  Default=FALSE.

xtype - colocar o nome do xlab nos dois gráficos plotados

ytype - colocar o nome do ylab nos dois gráficos plotados

... - argumentos da função plot, por favor veja função plot() e par().
Alguns argumentos que podem ser usados:

    pch - muda o tipo dos pontos do gráfico de dispersão
    col - muda as cores a serem colocadas no plot
    cex - aumenta do tamanho do texto do plot e dos símbolos
    main - muda o título do gráfico


Details:

A função é apenas aplicada para dados que sejam medidos em proporção ou
porcentagem.
O objetivo da função é verificar se duas curvas são iguais em função de uma
variável independente (como tempo, peso, etc).
Esta variável independente deve ter o mesmo intervalo nos dois conjuntos de
dados.
Se o argumento choosepoints for TRUE, a função irá produzir as linhas
secantes da taxa de variação média.
A função permite a entrada de um um fator com vários niveís no entanto
apenas os primeiros dois níveis serão usados para a função.


Value:

Terá uma lista como saída com os seguintes componentes:

Se, choosepoints=FALSE

    Tabela com as variáveis do primeiro conjunto de dados
    Tabela com as variáveis do segundo conjunto de dados
    Parâmetros do Teste Qui-quadrado e valor de p associado

Se, choosepoints=TRUE

    Tabela com as variáveis do primeiro conjunto de dados (mostrando apenas
as que estão no intervalo escolhido pelo utilizador)
    Tabela com as variáveis do segundo conjunto de dados (mostrando apenas
```

```
as que estão no intervalo escolhido pelo utilizador)
    Valor da taxa de variação média da primeira curva (entre os pontos
escolhidos pelo utilizador)
    Valor da taxa de variação média da segunda curva (entre os pontos
escolhidos pelo utilizador)
    Parâmetros do Teste Qui-quadrado e valor d p associado

Warning:

Aceita apenas data frame com uma coluna categórica com 1 fator de vários
níveis e duas colunas numéricas.
Se forem colocados mais de 2 níveis no factor a função irá apenas fazer a
comparação entre os dois primeiros níveis do data frame.

Author:

Filipa Alexandra de Abreu Paulos

Examples:

#Exemplo

x <- rep(c("Adequatefood","Inadequatefood"), each=15)
y <- rep.int(c(1:15), times=2)
z1 <- c(0.06666667, 0.33333333, 0.86666667, 0.60000000, 0.60000000,
0.66666667, 0.73333333, 0.80000000, 0.86666667, 0.80000000, 1.00000000,
1.00000000, 1.00000000, 0.66666667, 1.00000000)
z2 <- c(0.00000000, 0.66666667, 0.53333333, 0.53333333, 0.93333333,
0.80000000, 0.73333333, 0.93333333, 0.80000000, 0.93333333, 0.93333333,
0.93333333, 0.86666667, 0.86666667, 0.86666667)
z <- c(z1,z2)
dados <- data.frame(x,y,z)

curly(dados)

curly(dados, xtype="Proporção de escolhas correctas", ytype="Sessão",
pch=c(1,3))
#muda o nome dos eixos x e y e o formato dos pontos do gráfico.

curly(dados, choosepoints=TRUE, cex=2)
#faz com que a função peça para escolher dois pontos do eixo x para que se
faça a comparação apenas nesse intervalo. Aumento o tamanho de todos os
parâmetros do gráfico

##Exemplo com distribuições exponenciais

x <- rep(c("f","h"), each=60)
y <- rep.int(c(1:60), times=2)
zi <- pexp(1:500, 0.01)
z1 <- sort(sample(zi,60))
z2 <- sort(sample(zi,60))
```

```
z <- c(z1,z2)
da <- data.frame(x,y,z)

curly(da)

curly(da, choosepoints=TRUE)
#faz com que a função peça para escolher dois pontos do eixo x para que se faça a comparação apenas nesse intervalo.

curly(da, choosepoints=F, cex.lab=2, col="red")
#Muda alguns parâmetros do gráfico como o tamanho dos eixos x e y e a coloração dos pontos plotados.
```

Arquivos Curly

Função Curly

Help Curly

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:filipaabreu88:start

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