

Clara de Souza Corat



Doutoranda em Psicologia Experimental, Instituto de Psicologia - USP. Estudo cultura cumulativa e estratégias de aprendizagem social em macacos-prego selvagens.

exec

Proposta do Trabalho Final

Como eu estudo Etologia (comportamento animal), minhas propostas estão ligadas à essa área de estudos. Grande parte da pesquisa que realizamos utiliza um etograma, que contém as categorias comportamentais que desejamos observar nos animais. As minhas propostas se baseiam no método de amostragem de Varredura Focal (Scanning sample): varredura rápida de todo o grupo de indivíduos (ou de um mesmo indivíduo seguidamente por um tempo determinado, e posteriormente outro, até completar o grupo todo) em intervalos regulares, onde comportamento de cada indivíduo naquele instante é registrado.

Primeira proposta:

No fim das nossas coletas acabamos com um etograma muito grande, com observações diárias de diversos indivíduos de um grupo. Minha proposta é a de fazer uma função que inicialmente some todas as observações de um mesmo dia (no etograma sinalizamos a ocorrência ou não de um comportamento com 1 ou 0, respectivamente), para um mesmo indivíduo ou para o grupo, para cada categoria comportamental. A partir deste resultado diário, forneça um gráfico (acredito que um boxplot) que mostre a média, máximos e mínimos da ocorrência de cada categoria comportamental, durante o tempo de coleta. E, por fim, que forneça a porcentagem de cada categoria comportamental, com relação ao número de observações realizadas (para o indivíduo ou grupo). Estas informações podem ser utilizadas para inferir as atividades mais realizadas por um espécie (ou grupos), o tempo gasto com cada atividade, a importância das atividades para a sobrevivência do grupo (que pode ser relacionada com o ambiente, se grupos de ambientes diferentes forem comparados), etc.

Segunda proposta:

Para estudos de enriquecimento ambiental (um princípio no trato de animais que visa melhorar seu bem-estar em cativeiro ao identificar e fornecer estímulos necessários para o bem-estar fisiológico e psicológico) também utilizamos um etograma. O estudo de enriquecimento ambiental se divide basicamente em 3 etapas: antes do enriquecimento, durante o enriquecimento e depois do enriquecimento. Para avaliar se o uso do enriquecimento de fato modificou e melhorou o bem-estar do animal, comparamos a primeira e a última etapa. Uma das maneiras de comparar as etapas é observar se o tempo que o animal ficou ativo aumentou significativamente. Outra é observar se houve diminuição dos comportamentos estereotipados. Desta maneira, minha proposta é a de fazer uma função que inicialmente some todas as observações de um mesmo dia (como na proposta anterior), separando as etapas, para a categoria comportamental que quer ser comparada, tanto para o grupo ou para um indivíduo. A partir dos resultados, comparar se houve mudança significativa entre a ocorrência da atividade antes e depois (mas ainda não sei que teste utilizar). Um dos problemas deste tipo de pesquisa é o pequeno número de indivíduos que são observados, acredito que seja necessário fazer uma randomização dos dados para algum teste estatístico poder ser utilizado.

Clara , a sua primeira proposta me parece demasiadamente simples, pois , pelo que pude entender, a sua função estaria basicamente tirando médias por colunas /linhas e criando um boxplot com os resultados. Já na segunda proposta você acrescentou um grau de dificuldade a mais , que é o teste de significância. Eu sugiro incrementar um pouco mais essa sua função. Eu não estou muito familiarizada com o a sua área de trabalho, mas será que não existem mais demandas que possam ser atendidas com a criação da sua função?.

— *Cristiane*

Mais Comentários

Comentários Concordo com os comentários da Cris, acrescentaria que DEVE descrever melhor o formato dos dados de entrada, quais argumentos terá e o tipo e estrutura do objeto de saída. Além disso, não fala como fará o teste de hipótese.

Veja algumas dicas de como estruturar a proposta que incluímos no wiki:

http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01_curso_atual:alunos:trabalho_final:start#a_proposta

Aguardamos suas correções! Avise a Cris quando terminar via email acima para ela ou eu darmos uma olhada. —

Alexandre Adalardo de Oliveira 2014/04/25 17:02

Correção da proposta 2

Minha proposta irá utilizar dados de dois data frames, que representam etogramas, para comparar os comportamentos de indivíduos, ou grupos, em dois momentos diferentes de tempo. O que poderia ser utilizado tanto para comparar dados de enriquecimento ambiental, quanto, por exemplo, diferença do comportamento entre inverno e verão, ou comportamentos antes e depois do uso de drogas, etc. Os data frames seriam como este (para o grupo ou um indivíduo):



Ou este:



Onde as colunas são os comportamentos de interesse e as linhas são o número de observações. No primeiro exemplo, o número de comportamentos realizados por dia já foi somado, na segunda não (esta é a tabela bruta que sai de qualquer etograma). Não faz diferença, já será utilizada a média com relação aos dias (soma do total de observações de cada comportamento por dia, dividida pelo número de dias). Dentro da função haverá a identificação da existência de números maiores que 1 no data frame, caso existam, a função não precisará somar os dados de cada dia para cada comportamento.

As linhas que contêm dados faltantes serão eliminadas, e a função avisará o usuário disso.

Os parâmetros iniciais seriam:

`x = data frame do período inicial`

`y = data frame do período final`

`grupo = F`, o usuário terá que indicar se a função estará lidando com grupos ou um indivíduo. Se esse parâmetro for apontado como `TRUE`, a função dará um aviso de que está partindo do princípio que o número de indivíduos e os indivíduos, são os mesmos; ou seja, é um teste dentro de participantes. Não sei se o usuário precisa indicar o número de indivíduos do grupo, já que a função iria tratar o grupo como uma única unidade.

`grafico = boxplot` ou `plot`, a função irá gerar o gráfico indicado pelo usuário. Caso seja escolhido o `boxplot`, serão gerados dois gráficos, um do período inicial e um do final, com todos os comportamentos neles. Se o usuário indicar o `plot`, serão gerados gráficos para cada um dos comportamentos, onde cada dia de observação (soma das observações) aparecerá no gráfico e os dois períodos estarão com cores diferentes em cada gráfico.

`gráfico.2 = T`, caso o usuário queira os dois tipos de gráfico.

`porc.total = T`, a função irá gerar a porcentagem de ocorrência do comportamento em cada período, caso o usuário queira, além da porcentagem total, a porcentagem por dia, terá que colocar `FALSE` neste argumento.

A função não irá rodar caso o período inicial e final sejam diferentes, ou seja, o número de dias de observação tem que ser o mesmo.

A função irá realizar uma análise de modelo linear, para comparar, para cada comportamento, os dois períodos. De modo a testar a hipótese de que os dois períodos são diferentes. Caso o data frame tenha menos de 30 dias de observação, a função irá realizar uma randomização com os dados. Para que o teste estatístico possa ser realizado.

A função retornará os gráficos escolhidos pelo usuário, uma saída que indique a porcentagem de ocorrência do comportamento em cada período (ou/ou um data frame que indique essa porcentagem por dia do período), e o valor de p , para cada comportamento.

Comentários da Reformulação

Comentários A proposta agora está mais CLARA! Bom, entendi que a função faz um modelo linear da comparação

entre os dados antes e depois. O problema está agora na definição desse modelo. Imagino que vc. já fez alguma análise como esta, eu não! Não tenho nem ideia do que vc. está sugerindo como modelo linear. Para constar na função vc. deve definir exatamente que teste é esse e o que está testando.

Mesmo não conhecendo, imagino que um teste de hipótese mais geral seria - "Há diferença entre o comportamento antes e depois". Que nos dados poderia ser definido de duas formas: total de eventos (atividade maior ou menor) e pela distribuição relativa dos eventos (mudou a hierarquia de importância). Ambos os testes, apesar de parecer simples, tem complicações e não são testes de modelos lineares simples.

Focando só em um, para simplificar, por exemplo "se o bicho ou grupo foi mais ativos depois que ante", sugiro:

- defina apenas um tipo de dado de entrada (dados de contagem de cada comportamento por dia - primeiro exemplo)
- calcule a proporção entre as soma de eventos de atividade e não atividade (parado) por dia
- faça um teste se as diferenças entre as proporções médias por dia foram diferentes entre esse dois grupos (usando o procedimento que fizemos para o chagal dourado no tutorial 8 [o chagal dourado no tutorial 8](#))
- apresente o resultado do teste de permutação e o histograma dos valores medios por dia nos dois grupos.

Caso não queria seguir essa sugestão e deseje manter a proposta reformulada, defina exatamente o teste e diminua as possibilidades de ajustes que fornece ao usuário (tipo de grafico, tipo de dado de entrada, etc..) pq. apesar de desejáveis para tornar a função mais flexível, elas complicam o código.

— [Alexandre Adalardo de Oliveira](#) 2014/05/01 13:49

****Help da Função****

is.moment.dif

package:nenhum

R Documentation

Teste de permutação de dois momentos/situações diferentes.

Description:

Relacionada à área de Etologia (Estudo do Comportamento), esta função realiza um teste de permutação para comparação entre a quantidade de observações com a presença ou a ausência de um comportamento específico em dois Momentos no tempo ou duas Situações distintas (ex. forrageamento no verão e inverno; atividade antes e depois da ingestão de uma droga)

Usage:

```
is.moment.dif(x, y, n.per=1000, int.conf=0.95)
```

Arguments:

x Momento/Situação 1. Uma matrix com duas colunas. As linhas da matriz devem representar cada dia de observação.
A Primeira coluna deve conter os valores da somatória de observações do comportamento de interesse para cada dia. A Segunda coluna deve conter os valores da somatória de observações de comportamentos que não se referem ao comportamento de interesse da Primeira coluna, para cada dia.

y Momento/Situação 2. Idem x

n.per Número de permutações que a função irá realizar.

int.conf Valor do intervalo de confiança que será utilizado no teste de permutação. O valor padrão é 0.95.

Details:

A H_0 para esta função é de que os dois momentos/situações são iguais, estatisticamente.
Se a variação (diferença média na proporção das observações com e sem o comportamento de interesse) entre os momentos/situações encontrada é devido à variações não relacionadas à mudança ocorrida entre estes momentos/situações, é possível gerar essa diferença permutando os dados. Caso isso seja verdade encontraremos frequentemente diferenças iguais, maiores ou menores que a observada, e H_0 não poderá ser rejeitada.

Os valores da Primeira coluna são divididos pelos valores da Segunda coluna, para cada dia nos dois momento/situação (x e y), resultando na proporção, por dia, das observações com

e sem o comportamento de interesse. Estas proporções são permutadas (quantidade de permutações é definida pelo usuário, mas o padrão é 1000), então é calculada a diferença das médias das proporções de cada momento/situação. Utilizando o resultado das permutações, é realizado um teste bicaudal, que calcula a probabilidade (valor p) dos momentos/situação (x e y) serem iguais em relação à proporção média das observações com e sem o comportamento de interesse.

Os dados originais são mantidos como uma das amostras da permutação.

Se existirem valores faltantes (NAs) nos dados, a linha do dia com aqueles valores será eliminada antes da realização das análises.

Se os dados não estiverem em um objeto da classe "matrix" a função não irá rodar.

Se a matriz tiver um número de colunas diferente de 2, ou as colunas tirem número de linhas diferentes, a função não irá rodar.

Exemplo de matriz apropriada:

	Atividade	Não-Atividade
[1,]	84	315
[2,]	125	342
[3,]	106	278
[4,]	120	201
[5,]	95	339
[6,]	109	299
.		
.		
.		

Value:

Produz duas janelas gráficas, cada uma com dois gráficos. Uma apresenta dois gráficos de barras, um para cada Momento/Situação de interesse; em cada gráfico está representada a quantidade, por dia, de observações com presença e ausência do comportamento de interesse.

A outra apresenta:

1) Um gráfico de dispersão com as proporções da quantidade - por dia - de observações, com presença e ausência do comportamento de interesse (presença/ausência), para cada um dos momentos/situações;

no gráfico há também uma linha que representa a proporção média para cada momento/situação.

2) Um gráfico de densidade probabilística, que representa a diferença das médias das proporções geradas no teste de permutação; o gráfico apresenta duas linhas azuis, que indicam os limites do intervalo de confiança.

Apresenta em quantas permutações a diferença das médias de proporção (da quantidade de observações, com presença e ausência do comportamento de interesse) dos dois momentos/situações foi igual, maior ou menor que a diferença das médias de proporção dos dados originais.

Apresenta o valor p do teste de permutação, indicando se a diferença entre os momentos/situações é significativa ou não, levando em consideração o intervalo de confiança indicado pelo usuário.

Autor(s):

Clara de Souza Corat

clacorat@gmail.com

Example:

```
x=matrix(c(round(rnorm(30,mean=100,sd=40)),round(rnorm(30,mean=300,sd=60))),
ncol=2)
colnames(x)= c('Ativo','Inativo')
rownames(y)= paste("Dia",1:30)

y=matrix(c(round(rnorm(30,mean=300,sd=90)),round(rnorm(30,mean=100,sd=20))),
ncol=2)
colnames(y)= c('Ativo','Inativo')
rownames(y)= paste("Dia",1:30)

is.moment.dif(x,y,nper=1500,int.conf=0.99)
```

****Código da Função****

```
is.moment.dif=function(x,y,nper,int.conf=0.95)
{
  # Para a função caso x ou y não sejam da classe matriz, dando um aviso
  disso
  if(class(x)!="matrix"|class(y)!="matrix")
  {
    stop ("Os dados (x e y) precisam estar em um objeto da classe matrix")
  }
}
```

```
# Para a função caso x e y não tenham o mesmo número de linhas (dias de
observação),
# dando um aviso disso
if(nrow(x)!=nrow(y))
{
  stop("O número de observações nos dois momentos precisa ser o mesmo")
}
# Para a função caso x ou y não tenham duas colunas (presença e ausência
do comportamento),
# dando um aviso disso
if(ncol(x)!=2 | ncol(y)!=2)
{
  stop("Esta função trabalha apenas com dois momentos de tempo/situação
cada arquivo de dado precisa ter duas colunas")
}
# Se forem identificados NAs nos dados
if(sum(is.na(x))>1 | sum(is.na(y))>1) # Soma o número de NAs e verifique
se o valor é maior que 1
{
  # Produz uma na matriz, excluindo as linhas com NAs
  x=x[complete.cases(x),]
  y=y[complete.cases(x),]
  # Aviso que as linhas com NAs foram excluídas
  warning("As linhas com NAs foram removidas")
}
#Função para calcular a proporção entre as soma
#de observações com presença e ausência do comportamento por dia
divi=function(z) {z[,1]/z[,2]} #Função que divide as linhas da primeira
coluna da matriz, pelas linhas da segunda coluna.
# Cálculo da proporção de cada momento
prop.momento1=divi(x) #Aplica a função acima em x
prop.momento2=divi(y) #Aplica a função acima em y
##### Gráficos #####
# Para que os gráficos possam ser comparados visualmente,
# estabeleço que o limite do eixo y de ambos é o valor máximo
# de observações encontrado nos dados
if(max(x)>max(y)) # Compara os valores máximos das matrizes. Se o valor
máximo de x é maior, segue o abaixo.
{
  #Abrir uma janela gráfica
  x11()
  #Dividir a Janela gráfica em duas linhas e uma coluna
  par(mfrow=c(2,1))
  # Deixar a "caixa" dos gráficos com formato de L
  par(bty="l")
  # Barplots da presença e ausência do comportamento no Momento/Situação 1
  barplot(x, beside=T, density=0,col="", main="Momento/Situação 1",
          ylab="Frequência de observações",ylim=c(1,max(x)))
  mtext(c("Dias de observação para cada comportamento"), side=1, line=2.5,
        cex=1) #Insere, no gráfico, a legenda do eixo x
```

```

# Barplots da presença e ausência do comportamento no Momento/Situação 2
barplot(y, beside=T, density=0,col="", main="Momento/Situação 2",
        ylab="Frequência de observações",ylim=c(1,max(x)))
mtext(c("Dias de observação para cada comportamento"), side=1, line=2.5,
cex=1) #Insere, no gráfico, a legenda do eixo x
}
if(max(y)>max(x)) # Compara os valores máximos das matrizes. Se o valor
máximo de y é maior, segue o abaixo.
{
  #Abrir uma janela gráfica
  x11()
  #Dividir a Janela gráfica em duas linhas e uma coluna
  par(mfrow=c(2,1))
  # Deixar a "caixa" dos gráficos com formato de L
  par(bty="l")
  # Barplots da presença e ausência do comportamento no Momento/Situação
1
  barplot(x, beside=T, density=0,col="", main="Momento/Situação 1",
        ylab="Frequência de observações",ylim=c(1,max(y)))
  mtext(c("Dias de observação para cada comportamento"), side=1,
line=2.5, cex=1) #Insere, no gráfico, a legenda do eixo x
  # Barplots da presença e ausência do comportamento no Momento/Situação
2
  barplot(y, beside=T, density=0,col="", main="Momento/Situação 2",
        ylab="Frequência de observações",ylim=c(1,max(y)))
  mtext(c("Dias de observação para cada comportamento"), side=1,
line=2.5, cex=1) #Insere, no gráfico, a legenda do eixo x
}
# Gráfico de dispersão da proporção entre as soma de observações
# com presença e ausência do comportamento por dia
# Estabeleço que o limite do eixo y é o maior valor encontrado
# nas proporções medidas
if(max(prop.momento1)>max(prop.momento2)) # Compara os valores máximos das
proporções calculadas. Se o valor máximo do momento/situação 1 (x) é maior,
segue o abaixo.
{
  # Abrir outra janela gráfica
  x11()
  #Dividir a Janela gráfica em duas linhas e uma coluna
  par(mfrow=c(2,1))
  # Deixar a "caixa" dos gráficos com formato de L
  par(bty="l")
  # Estabelecer as margens do gráfico, para que as legendas caibam no
mesmo
  par(mar=c(3,4,4,1))
  # Estabelecer o tamanho dos números dos eixos no gráfico e o tamanho do
"tick"
  par(cex.axis=0.7, tcl=0.3)
  # Construir o gráfico
  plot(prop.momento1,ylim=c(0,max(prop.momento1)+2), ann=F,
col="red",pch=15)

```

```
# Colocar as legendas dos eixos
mtext("Dias de observação",side=1,line=2,cex=0.9) # Colocar legenda do
eixo x
mtext("Proporção Presença/Ausência \n do
Comportamento",side=2,line=2,cex=0.9) #Colocar legenda do eixo y
# Colocar o nome do gráfico
mtext("Gráfico de dispersão das proporções",side=3,line=1.5,cex=1)
# Introduzir outro conjunto de dados (gráfico) no mesmo gráfico
par(new=T)
# Construir o segundo gráfico, que será construído em cima do primeiro
plot(prop.momento2,ylim=c(0,max(prop.momento1)+2), ann=F,
axes=F,col="green",pch=16)
# Colocar as linhas que indicam o valor médio de cada conjunto de dados
abline(h=mean(prop.momento1), col="red") # Colocar, no gráfico, linha do
valor médio dos valores do cálculo das proporções de x
abline(h=mean(prop.momento2), col="green") # Colocar, no gráfico, linha
do valor médio dos valores do cálculo das proporções de y
# Colocar a legenda dos dados do gráfico
legend("topright",legend=c("Momento1","Momento2"), col=c("red","green"),
pch=c(15,16),cex=0.8,ncol=2)
# Colocar a legenda das linhas de média
legend("topleft",legend=c("Média Momento1","Média Momento2"),
col=c("red","green"), lty=c(1,1),cex=0.8,ncol=2)
}
if(max(prop.momento2)>max(prop.momento1)) # Compara os valores máximos das
proporções calculadas. Se o valor máximo do momento/situação 2 (y) é maior,
segue o abaixo.
{
#Abrir outra janela gráfica
x11()
#Dividir a Janela gráfica em duas linhas e uma coluna
par(mfrow=c(2,1))
# Deixar a "caixa" dos gráficos com formato de L
par(bty="l")
# Estabelecer as margens do gráfico, para que as legendas caibam no
mesmo
par(mar=c(3,4,4,1))
# Estabelecer o tamanho dos números dos eixos no gráfico e o tamanho do
"tick"
par(cex.axis=0.7, tcl=0.3)
# Construir o gráfico
plot(prop.momento1,ylim=c(0,max(prop.momento2)+2), ann=F,
col="red",pch=15)
# Colocar as legendas dos eixos
mtext("Dias de observação",side=1,line=2,cex=0.9) # Colocar legenda do
eixo x
mtext("Proporção Presença/Ausência \n do
Comportamento",side=2,line=2,cex=0.9) # Colocar legenda do eixo y
# Colocar o nome do gráfico
mtext(" Gráfico de dispersão das proporções", side=3,line=1.5,cex=1)
```

```
# Introduzir outro conjunto de dados (gráfico) no mesmo gráfico
par(new=T)
# Construir o segundo gráfico, que será construído em cima do primeiro
plot(prop.momento2,ylim=c(0,max(prop.momento2)+2), ann=F,
axes=F,col="green",pch=16)
# Colocar as linhas que indicam o valor médio de cada conjunto de dados
abline(h=mean(prop.momento1), col="red") # Colocar, no gráfico, linha
do valor médio dos valores do cálculo das proporções de x
abline(h=mean(prop.momento2), col="green") # Colocar, no gráfico, linha
do valor médio dos valores do cálculo das proporções de y
# Colocar a legenda dos dados do gráfico
legend("topright",legend=c("Momento1","Momento2"), col=c("red","green"),
pch=c(15,16),cex=0.8,ncol=2)
# Colocar a legenda das linhas de média
legend("topleft",legend=c("Média Momento1","Média Momento2"),
col=c("red","green"), lty=c(1,1),cex=0.8,ncol=2)
}
##### Análise Estatística
#####
# Como os dados podem não atender as premissas de um teste t,
# o substituímos por um teste de permutação ("embaralhamos" os elementos
de um ou mais vetores,
# cujos elementos são amostrados ao acaso sem reposição). Desta maneira,
será possível
# determinar se os grupos (Momento 1 e Momento 2) são de fato diferentes
# com relação à proporção de observações com presença e ausência do
comportamento.

# Preparação para a permutação (Para ter certeza que os momentos
# são de fato diferentes "embaralhamos" os elementos de um ou mais
# vetores, o que podemos fazer gerando um vetor de mesmo tamanho,
# cujos elementos são amostrados ao acaso sem reposição)

# Concatenar as proporções de cada momento em um novo objeto
prop.juntas=c(prop.momento1,prop.momento2)

# Criar um objeto que tenha o mesmo tamanho do objeto acima (prop.juntas),
# com os nomes referente aos dados do objeto acima
momento= rep(c("Momento1", "Momento2"), each=30)
# cria um vetor para armazenar os resultados (repleto de NA, que serão
substituídos
# abaixo, com o tamanho do número de permutações escolhido pelo usuário)
result=rep(NA,nper)
# Utilizar os valores originais como uma amostra da permutação
result[1]=diff(tapply(prop.juntas,momento,mean)) # Tira a média das
proporções dos dois momentos, para os dados originais, e verifica a
diferença entre estas médias
# Permutar os valores dos dados, calcular a diferença entre as médias das
# proporções dos momentos e armazenar no vetor "results".
# Repetir a operação mil vezes.
```

```
for(i in 2:nper)
{
  diferenca=diff(tapply(sample(prop.juntas),momento,mean)) # Realiza a
permutação (sample) dos dados originais, tira a média das proporções dos
dois momentos e verifica a diferença entre estas médias, para cada
permutação
  result[i]=diferença # Guarda no objeto criado acima o resultado da
diferença de médias de cada permutação
}

# Mostrar para o usuário quantas permutações têm médias iguais, maiores
# ou menores do que a observada nos dados originais
perm.obs.maior=sum(result>=result[1]) # Soma, e guarda em um objeto o
valor, a quantidade de resultados, encontrados acima (das permutações), que
é maior ou igual ao resultado da diferença de médias das proporções dos
dados originais
perm.obs.menor=sum(result<=(result[1]*-1)) # Soma, e guarda em um objeto o
valor, a quantidade de resultados, encontrados acima (das permutações), que
é menor ou igual ao resultado da diferença de médias das proporções dos
dados originais
cat("Em\t", perm.obs.maior, " das\t", nper, "permutações a diferença das
médias foi igual ou maior do que a observada.\n") # Retorna ao usuário esta
mensagem com o resultado da soma dos valor iguais ou maiores ao resultado da
diferença de médias das proporções dos dados originais
cat("Em\t", perm.obs.menor, " das\t", nper, "permutações a diferença das
médias foi igual ou menor do que a observada.\n
Observe o Gráfico de Densidade Probabilística\n\n\n") # Retorna ao
usuário esta mensagem com o resultado da soma dos valor iguais ou maiores ao
resultado da diferença de médias das proporções dos dados originais

# Gráfico de densidade probabilística dos valores das permutações, para
demonstrar
# se a H0 (as médias são iguais, ou seja, os grupos não diferem) pode ser
rejeitada
# nos dados originais
# Determinar as margens do gráfico
par=(mar=c(4,3,2,1))
# Estabelecer o tamanho dos números dos eixos no gráfico
par(cex.axis=0.7, tcl=0.3)

# Construir o gráfico
plot(density(result),xlab="Diferença das Médias",ylab="Frequência
Relativa", ann=F)

# Colocar nome do gráfico
mtext("Gráfico de densidade probabilística",cex=1,side=3,line=1)

# Colocar legendas dos eixos no gráfico
mtext("Diferença das Médias",cex=0.9,side=1,line=2) # Colocar legenda do
eixo x
```

```
mtext("Frequência Relativa",cex=0.9,side=2,line=2) # Colocar legenda do
eixo y
# Inserir duas linhas, no gráfico, que mostra a diferença das
# médias das proporções de cada momento, sem permutação (dados originais)
abline(v = result[1], col="blue") # Inserir a linha que representa a
diferença dos dados originais, que limita os maiores valores do intervalo de
confiança
abline(v = (result[1]*-1), col="blue") # Inserir a linha que representa a
diferença dos dados originais, que limita os menores do intervalo de
confiança

# Calcular a quantidade de resultados que são maiores ou menores que
# a diferença da proporção média dos dados originais
bicaudal=sum(result>=result[1]| result<=(result[1]*-1)) # Calcular a
quantidade de resultados da permutação que estão fora do intervalo de
confiança
# calcula a probabilidade (valor p) dos momentos/situação (x e y) serem
# iguais em relação à proporção média das observações com a presença e
ausência do comportamento
# de interesse, ao dividir o número de resultados fora do intervalo de
confiança pelo número total de resultados da permutação.
p.bi=bicaudal/length(result)

# Apresentar a o valor p do teste de permutação para o usuário

# Mudança de mensagem que apresenta o resultado, dependendo do valor p e
do intervalo
# de confiança escolhido.

# Mensagem quando o teste é significativo
if(p.bi<=(1-int.conf)) # Apresentar a mensagem abaixo se o valor p menor
ou igual ao valor de intervalo de confiança estipulado pelo usuário
{
  return(cat("O valor p do Teste de permutação foi de\t",p.bi,"
como intervalo de confiança estabelecido foi\t",int.conf,"
a diferença entre os Momentos/Situações é significativa,
a H0 (os dois Momentos são iguais) pode ser rejeitada,
ou seja,os dois Momentos/Situações de interesse são diferentes
em relação à proporção da presença e ausência do comportamento\n"))
}
# Mensagem quando o teste não é significativo
if(p.bi>(1-int.conf)) # Apresentar a mensagem abaixo se o valor p menor ou
igual ao valor de intervalo de confiança estipulado pelo usuário
{
  return(cat("O valor p do Teste de permutação foi de\t",p.bi,"
como intervalo de confiança estabelecido foi\t",int.conf,"
a diferença entre os Momentos/Situações não é significativa,
a H0 (os dois Momentos são iguais) não pode ser rejeitada,
ou seja,os dois Momentos/Situações de interesse não são diferentes
em relação à proporção da presença e ausência do comportamento de
interesse\n"))
}
```

```
}
```

```
}
```

[funcao_clara.r](#)

[help.r](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2014:alunos:trabalho_final:clacorat:start 

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