Lucia Neco

Mestranda em Comportamento Animal

Programa de Pós-graduação em Psicologia Experimental

Instituto de Psicologia

Universidade de São Paulo

Currículo Lattes

Exercícios

exec

Trabalho Final

Proposta A

A função aqui proposta deverá facilitar o teste da significância das associações de indivíduos em um grupo social. As associações são medidas a partir da identificação de grupos entre indivíduos em contextos sociais. A definição do que é cada grupo e em que contexto eles serão observados é feita pelo pesquisador.

Diversos artigos usam a medida de associação entre indivíduos para aplicar métricas de estrutura da rede social, como a coesão da população, então essa função será útil como ponto de partida para outras análises. Assim, mais especificamente, a função usará o teste de Monte Carlo (desenvolvida por Manly) adaptada por Bejder (1998) para verificar essa significância usando um dos três tipos de índices de associação a serem definidos pelo usuário, a partir de seus dados.

Argumentos da função:

- file: objeto do tipo matriz binária em que as linhas representam grupos identificados nas amostras e as colunas representam indivíduos do grupo social estudado (1 se presente neste grupo, 0 se não presentes).
- indice: escolha entre os três tipo de índice de associação presentes na literatura – “sri”(simple ratio index), “hwi” (half weight index), “twi”(twice weighted index) (Croft et al. 2007)

Objeto de saída:

Uma lista contendo:

- a matriz com os valores de associação para cada par de indivíduos (linhas e colunas são os indivíduos do objeto “file”—correspondentes às colunas)

- uma matriz de mesmo tamanho da anterior contendo a probabilidade de cada valor de associação estar incluído em uma distribuição randômica de interações
- um histograma com a distribuição da frequência das associações observadas.

Bejder, L., Fletcher, D., & Bräger, S. (1998). A method for testing association patterns of social animals. Animal Behaviour, 56(3), 719-725.

Croft, D.P., James, R. and Krause, J., 2008. Exploring animal social networks. Princeton University Press.

Comentários Renata Orofino

Lucia, já tinha ouvido um pouco da função. Acho legal, mas obviamente vou dar pitaco.
Seria interessante ver se os diferentes métodos dão uma resposta semelhante? Ou seja, ver se a associação de indivíduos naquele caso tem resultado consistente de acordo com os diferentes métodos?
Fora isso, acho joia, pode investir. Lembre que o help dessa sua função vai ter que ser bem bonitinho pra ajudar seu usuário, ok?

Ótima proposta, Lucia. Reforçando o comentário da Rena, lembre de descrever no help da função como funciona cada índice e colocar as referências. — *Sara*

Proposta B

A função proposta realizará os testes de validação para experimentos de personalidade de acordo com a metodologia de Carter et al. (2013).

Neste artigo propõe-se que, para relacionar uma dimensão da personalidade (por exemplo, agressividade) a um comportamento específico, devemos verificar se a dimensão da personalidade realmente está presente nos objetos de estudo. Essa verificação é feita através do teste da correlação de resultados de dois experimentos para a mesma dimensão da personalidade (validação positiva) e de um experimento para outra dimensão da personalidade.

Usando a função rrcor() do pacote Hmisc podemos obter as correlações entre os dados e o nível de significância de cada correlação.

Argumentos da função:

- file: Uma matriz (como requerido pela função rrcor()) na qual as linhas são os indivíduos testados e as colunas são os diferentes testes realizados para a validação.
- test: O usuário pode escolher de acordo com seus dados qual tipo de correlação deverá ser realizada: “pearson”, “spearman”.

Objeto de saída:

Uma lista contendo:

- Plot das correlações entre os dados (o número de plots vai depender do número de testes realizados)
- Sumário dos resultados das correlações e suas significâncias

Carter, A. J., Feeney, W. E., Marshall, H. H., Cowlishaw, G., & Heinsohn, R. (2013). Animal personality: what are behavioural ecologists measuring?. Biological Reviews, 88(2), 465-475.

Comentários Renata Orofino

É interessante também, mas acho que a ideia A é mais bacana e vai ser mais útil pra vc, pelo que conversamos. Se tiver dúvidas, me avise: renata.orofino@gmail.com

Acho também a 1 mais interessante e que você dedicou mais tempo pensando nela. Manda bala! — *Sara*

Trabalho final - Minha função

funcao

Função assoc

Autora: Lucia Neco

Obs: O código completo não está funcionando no meu computador como deveria. Como não consegui achar o erro até o final do tempo para postar, coloco o código com opções alternativas.

Obs2: O problema foi resolvido e adicionei uma versão atualizada do código

Arquivos da função

Help da função assoc .txt

Help da função assoc .docx

Função assoc

Função assoc com problemas

Função assoc - Retorno indices de associação e histograma apenas

Função assoc - Retorno resultado do teste de significância e histograma apenas

Help da função

assoc {Disciplina ecoR} Lucia Carvalho Neco

Teste de significância da associação de indivíduos

Descrição

A função “assoc” executa o teste da significância das associações de indivíduos em um grupo social. A função utiliza o teste de Monte Carlo (desenvolvida por Manly) adaptada por Bejder (1998) para verificar essa significância usando um dos três tipos de índices de associação: simple ratio index, half weight index e twice weight index.

Uso

assoc(file, INDEX)

Argumentos

file objeto do tipo dataframe em que as linhas representam grupos identificados nas amostras e as colunas representam indivíduos do grupo social estudado (1 se presente neste grupo, 0 se não presentes). A primeira coluna do dataframe deve conter as repetições/observações co respondente aos dados.

INDEX escolha entre os três tipo de índice de associação presentes na literatura – “sri”(simple ratio index), “hwi” (half weight index), “twi”(twice weighted index): cheque “Detalhes”.

Detalhes

A função exclui Nas dos dados antes de começar as análises. Use “sri” em INDEX para associações em que o observador é capaz de localizar todos os indivíduos em cada repetição/observação. Use “hwi” em INDEX se existe algum viés em localizar indivíduos em grupos diferentes. Use “twi” em INDEX se existe algum viés em localizar indivíduos quando estão juntos formando um grupo.

Valor

A função retorna um objeto de saída do tipo lista contendo: uma matriz com os valores dos índices de associação para cada par de indivíduos presentes no dataframe; uma matriz de mesmo tamanho da anterior contendo a probabilidade de cada valor de associação estar incluído em uma distribuição randômica de interações; um histograma com a distribuição da frequência das associações observadas;

Referências

Bejder, L., Fletcher, D., & Bräger, S. (1998). A method for testing association patterns of social animals. Animal Behaviour, 56(3), 719-725.

Croft, D.P., James, R. and Krause, J., 2008. Exploring animal social networks. Princeton University Press.

Exemplo

##Dados de associação de indivíduos com 5 observações, 7 indíviduos e 10 grupos no total

file=matrix(c(1,1,0,0,0,1,1,0,0,1,1,1,0,0,0,0,1,0,0,0,1,0,0,1,0,1,1,0,0,1,1,0,0,0,0,1,0,1,0,1,1,0,0,1,0,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,1,1,1,0,1,0,0,0,0,0,1,0,1,0,1), ncol=7,nrow= 10)

file = cbind(c(1,1,2,2,2,3,3,4,5,5), file)

file = data.frame(file)

assoc(file = file , INDEX = ”sri”)

Código da função

```
assoc = function (file, INDEX) # Cria a função
{

#### INDICES DE ASSOCIACAO OBSERVADOS
file = na.omit(file) #Retira quaisquer NAs dos dados
file2 = file[,1] #Transforma a primeira coluna do dataframe em um objeto
separado
file = file[,-1] #Retira a primeira columa do dataframe, deixando apenas as
observa??es de presen?a e ausencia dos individuos

indices=matrix(NA, ncol=ncol(file), nrow=ncol(file)) # Criar uma matriz para
guardar os indices de associacao observados

for(a in 1:(ncol(file)-1))# Repeticoes para cada par de individuos
{
  for(b in (a+1):ncol(file))#  Repeticoes para cada par de individuos
  {
    x = length(file[,a][file[,a]==1 & file[,b]==1]) #numero de vezes a e b
estavam no mesmo grupo
    x
    indjuntos = aggregate(file, list(c(file2)), sum) # Juntando as
observacoes de cada repeti??o para averiguar depois os indiv?duos observados
na mesma repeticao
    yab =  (length(indjuntos[,a][indjuntos[,a]>=1 | indjuntos[,b]>=1])) - x
#numero de vezes que os dois indiv?duos foram observados, por?m em grupos
diferentes
    yab

    ya = length(file[,a][file[,a]==1 & file[,b]==0]) #numero de vezes em que
o individuo a foi observado em um grupo, mas nao b
    ya

    yb = length(file[,a][file[,b]==1 & file[,a]==0])#numero de vezes em que
o individuo b foi observado em um grupo, mas nao a
  yb


  if (INDEX == "sri") #escolha do usuario
```

```
  {
index = x/(x + yab + ya + yb) #simple ratio index
  }
  if (INDEX == "hwi") #escolha do usuario
  {
index = x/(x + yab + 1/2*(ya + yb)) #half weight index
  }
  if (INDEX == "twi") #escolha do usuario
  {
index = x/(x + (2*yab) + ya + yb) #twice weight index
  }
  if(INDEX!="sri"&INDEX!="hwi"& INDEX!="twi") #Parar a funcao se o usuario
nao escolheu um indice
  {
    stop("Nenhum indice de associacao escolhido") # Aviso de erro
  }
indices[a,b] = index #Salvar os resultados em uma matriz
  }
}

indices



##### HISTOGRAMA DE FREQUENCIAS OBSERVADAS

hist(indices, main="Histograma dos Indices de Associacao \n Observados",
xlab="Valor do Indice de Associacao",ylab="Frequencia", col = "purple")
#Cria histograma dos valores do indice de associacao observados




##### TESTE DE SIGNIFIC?NCIA - PERMUTACAO


pmontecarlo=matrix(NA, ncol=ncol(file), nrow=ncol(file)) # Criar uma matriz
para guardar os resultados de p depois do teste de significancia

for(i in 1:(ncol(file)-1)) # Repeticoes para cada par de individuos
{
  for(j in (i+1):ncol(file)) # Repeticoes para cada par de individuos
  {

  simulaab = rep(NA, 1000) #Vetor para guardar os valores de cada simula??o.

  simulaab[1] = indices[i,j] #Valor observado na primeira posicao do vetor
da simulacao
```

```
  for (v in 2:1000) #Ciclo com contador (v) que vai de 2 a 1000 para as
simulacoes
  {
    filemc = apply(file, 1, sample) # Aqui os dados sao embaralhados,
mantendo o mesmo numero de individuos em cada grupo
    filemc = data.frame(t(filemc)) #Transpor o resultado do apply com o
sample que inverte a posicao de linhas e coluna dos dados

    indicesmc=matrix(NA, ncol=ncol(file), nrow=ncol(file)) # Matriz para
guardar os indices gerados aleatoriamente
    for(c in 1:(ncol(file)-1)) # Repeticoes para cada par de indiv?duos
    {
      for(d in (c+1):ncol(file)) # Repeticoes para cada par de indiv?duos
      {
        x = length(filemc[,c][filemc[,c]==1 & filemc[,d]==1]) #numero de
vezes a e b estavam no mesmo grupo
        x
        indjuntos = aggregate(filemc, list(c(file2)), sum) # Juntando as
observacoes de cada repeticao para averiguar depois os individuos observados
na mesma repeticao
        yab =  (length(indjuntos[,c][indjuntos[,c]>=1 | indjuntos[,d]>=1]))
- x #numero de vezes que os dois individuos foram observados, porem em
grupos diferentes
        yab
        ya = length(filemc[,c][filemc[,c]==1 & filemc[,d]==0]) #numero de
vezes em que o individuo a foi observado em um grupo, mas nao b
        ya
        yb = length(filemc[,c][filemc[,c]==1 & filemc[,d]==0])#numero de
vezes em que o indiv?duo b foi observado em um grupo, mas n?o a
        yb
        if (INDEX == "sri") #escolha do usuario
        {
          indexmc = x/(x + yab + ya + yb) #simple ratio index
        }
        if (INDEX == "hwi")#escolha do usuario
        {
          indexmc = x/(x + yab + 1/2*(ya + yb)) #half weight index
        }
        if (INDEX == "twi")#escolha do usuario
        {
          indexmc = x/(x + (2*yab) + ya + yb) #twice weight index
        }
        indicesmc[c,d] = indexmc #Salvar os resultados em uma matriz
      }
    }
      simulaab[v] = indicesmc[i,j] #Salvar os resultados em um vetor
    }
  obsXesp=sum(simulaab>=simulaab[1]) #Separar os indices formados
aleatoriamente que est?o acima do valor observado
  p=obsXesp/length(simulaab) #Qual a proporcaoo de valores aleatorios sao
maiores do que o valor observado
```

```
  pmontecarlo [i,j] = p # Guardar esse valor em uma matriz
  }
}
indices = data.frame(indices) #Transformar em data.frame
pmontecarlo = data.frame(pmontecarlo) #Transformar em data.frame

listaassoc = list(indices,pmontecarlo) # Criando lista com o objeto indices
e o objeto pmontecarlo

return(listaassoc) # O que a função vai retornar para o usuario

}# Fechar a funcao
```

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:lcneco:start

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