Danilo Pereira Mori

Mestrando do programa de pós graduação do Depart. de Ecologia IB-USP sob orientação do professor Paulo Inácio e do doutor Renato A.F. Lima. Venho trabalhando com a equipe há mais de 1 ano no projeto Neotropical Tree Communities database (TreeCo) http://labtrop.ib.usp.br/doku.php?id=projetos:treeco:start. Cursando o primeiro ano de mestrado, projeto em construção.

Assuntos de interesse: ecologia de comunidade, teoria de nicho, teoria neutra, distribuição de abundância de espécies (SAD), abundância numérica, biomassa, simulação, banco de dados, fitossociologia.

Meus Exercícios

Caminho para os exercícios que realizei durante a disciplina exec

Minhas propostas

Proposta 1 - função 'gremlims'

Uma das maneiras de comparar modelos de regressão linear concorrentes par a par é utilizar a função ‘anova’. Essa tarefa é simples quando se tem poucas variáveis preditoras (VP) para um única variável resposta (VR). No entanto, quando há muitas VP a tarefa se torna enfadonha e por vezes não há a comparação de todas as possibilidades de combinações. Irei utilizar combinações, em detrimento de arranjo, pensando que a ordem das VP dentro do modelo não importa, ou seja pressuponho que a colinearidade entre elas é desprezível, afim de não inflar o tempo de execução da função.

A função irá gerar todas as combinações de modelos de regressão linear dado uma VR e um conjunto de VP; as interações entre as variáveis também serão consideradas. O input da função será um data frame ou uma matriz. O usuário deverá indicar qual é a VP e quais as VR, utilizando a posição das colunas. A função realizará as comparações par a par utilizando a função ‘anova’ e retornará um objeto do tipo lista contendo a matriz simétrica com os valores de cada comparação e o modelo mais simples com maior poder preditivo.

Proposta 2 - função 'bandex'

Todos os dias milhares de estudantes precisam se confrontar com decisões que tem grande impacto em sua vida: em qual bandejão terão sua revigorante refeição. Quem frequenta os bandejões sabe que a mão que afaga é a mesma que apedreja (e.g. hamburguer ao molho americano - a.k.a. azia em disco ao molho tristeza). Assim, para facilitar a vida daqueles que não tem tempo para decidir em qual restaurante irão comer, ou se está na hora de usar aquela reserva de dinheiro para comer fora

do bandejão, pretendo criar a função ‘bandex’. A ideia geral é comparar as preferências alimentares do usuário com os itens alimentares disponíveis no bandejão sugerindo onde o usuário terá maior satisfação em cada uma de suas refeições durante a semana.

A função irá comparar duas matrizes: i) das preferências alimentares do usuário e ii) itens alimentares por bandejão(ões) selecionado(s) por período. A primeira matriz será composto de um vetor com todos os itens alimentares que costumam aparecer no bandejão atribuídos com o valor padrão zero; o usuário poderá atribuir a cada um dos itens um valor entre -1 e 1, ele poderá escolher se irá fazer a atribuição a todos os itens alimentares ou apenas a certos grupos (e.g. prato principal, guarnição, etc). A segunda matriz será obtido pela leitura do site do cardápio (no caso da USP, o site do coseas, mas qualquer lugar que tenha um sistema similar poderá ter uma versão da função bandex), os itens alimentares serão computados por palavras-chave, por exemplo, ‘legumes’ ao invés de legumes à jardineira.

A função retorna uma série de histogramas por período (almoço ou janta), no eixo x há os bandejões selecionados, no eixo y o “Grau de Felicidade Potencial”. Para fazer o cálculo do GFP irei combinar as duas matrizes, da primeira irei utilizar os valores da segunda os itens alimentares, a soma dos valores dessa matriz combinada é o valor da GFP. No título de cada histograma haverá o nome do bandejão com melhor GFP assim como o item alimentar com melhor cotação - o grande motivo de ir ao bandejão. Se nenhum bandeijão obter um GFP>0 então o título irá indicar que talvez seja melhor não bandejar. As preferências alimentares também poderão ser salvas em um vetor para uso posterior.

> *Danilo, a proposta 1 tem alguns problemas estatísticos. Para comparar modelos com diferentes variáveis preditoras, é necessário penalizar os modelos com mais parâmetros e a ANOVA não faz isso. Como você iria fazer isso? Além disso, a melhor abordagem para seleção de modelos é baseada em critérios de informação, p.ex. AIC. Conversei com o Danilo Muniz (que entende bastante sobre seleção de modelos) e ele indicou os seguintes artigos que falam sobre os problemas dessa abordagem: “Stepwise Model Fitting and Statistical Inference: Turning Noise into Signal Pollution” e “Mundry,R & Nunn, C. L. 2009. The American Naturalist 173: 119-123”. A proposta 2 parece interessante e viável. —* Juliana Vendrami 2015/03/25 16:24

> Além disso, a função 'anova' só pode ser usada para testar modelos aninhados. Ou seja, você não pode sair testando modelos a esmo com essa função. A melhor alternativa, como a Ju comentou, seria usar o AIC. — *Leo*

Página de Ajuda

```
bandex                  package:unknown                  R Documentation

Gráficos de Grau de Felicidade Potencial nos bandejões da Cid. Universitária
(CUASO)

Description:

     Produz gráficos de Grau de Felicidade Potencial (GFP) a partir da
comparação dos gostos pessoais do usuário com itens alimentares chave do
cardápio dos restaurantes universitários

Usage:

     bandex(bandex="A")
Arguments:

    bandex: carácter. Quando "A" ele compara os cardápios dos bandejões da
Química e do Central. Quando "C" ele mostra o GFP para cada dia da semana no
bandejão central, quando "Q" faz o mesmo para o bandejão da Química.

Details:

    A partir da atribuição dos itens alimentares pelo usuário, a função
compara esses itens àqueles presentes no cardapio de cada dia e período
(almoço ou janta).
O GFP é calculado somando o valor de cada iten alimentar presente no dia e
período.
Essa primeira versão, compara apenas para os dias da semana, deixando o
almoço do fim de semana de fora da análise.

Value:

    Um gráfico é gerado, o vetor de posições do gráfico também é gerado mas
não é mostrado.

Warning:

    Sempre preencham os valores dos itens alimentares com valores entre -1 e
1 para permitir a visualização dos gráficos

Note:

     Versão beta da função.

Author(s):
Danilo Pereira Mori
danilo.mori@usp.br
```

```
Examples:

bandex(bandex="C") #para o bandejão central
bandex(bandex="A") #para comparar bandejões
```

Código da Função

```
bandex<-function(bandex)
{
  webcentral<-"http://www.usp.br/coseas/cardapio.html"
  card.central<-readLines(webcentral)
  webquimica<-"http://www.usp.br/coseas/cardapioquimica.html"
  card.quim<-readLines(webquimica)
  dias.central<-grep("FEIRA",card.central)
  dias.quim<-grep("FEIRA",card.quim)
  segunda_almoco.q<-card.quim[dias.quim[1]:dias.quim[2]-1]
  segunda_janta.q<-card.quim[dias.quim[2]:dias.quim[3]-1]
  terca_almoco.q<-card.quim[dias.quim[3]:dias.quim[4]-1]
  terca_janta.q<-card.quim[dias.quim[4]:dias.quim[5]-1]
  quarta_almoco.q<-card.quim[dias.quim[5]:dias.quim[6]-1]
  quarta_janta.q<-card.quim[dias.quim[6]:dias.quim[7]-1]
  quinta_almoco.q<-card.quim[dias.quim[7]:dias.quim[8]-1]
  quinta_janta.q<-card.quim[dias.quim[8]:dias.quim[9]-1]
  sexta_almoco.q<-card.quim[dias.quim[9]:dias.quim[10]-1]
  sexta_janta.q<-card.quim[dias.quim[10]:length(card.quim)]
  dia.per.quim<-
list(segunda_almoco.q,segunda_janta.q,terca_almoco.q,terca_janta.q,quarta_al
moco.q,quarta_janta.q,quinta_almoco.q,quinta_janta.q,sexta_almoco.q,sexta_ja
nta.q)
  segunda_almoco.c<-card.central[dias.central[1]:dias.central[2]-1]
  segunda_janta.c<-card.central[dias.central[2]:dias.central[3]-1]
  terca_almoco.c<-card.central[dias.central[3]:dias.central[4]-1]
  terca_janta.c<-card.central[dias.central[4]:dias.central[5]-1]
  quarta_almoco.c<-card.central[dias.central[5]:dias.central[6]-1]
  quarta_janta.c<-card.central[dias.central[6]:dias.central[7]-1]
  quinta_almoco.c<-card.central[dias.central[7]:dias.central[8]-1]
  quinta_janta.c<-card.central[dias.central[8]:dias.central[9]-1]
  sexta_almoco.c<-card.central[dias.central[9]:dias.central[10]-1]
  sexta_janta.c<-card.central[dias.central[10]:length(card.central)]
  dia.per.central<-
list(segunda_almoco.c,segunda_janta.c,terca_almoco.c,terca_janta.c,quarta_al
moco.c,quarta_janta.c,quinta_almoco.c,quinta_janta.c,sexta_almoco.c,sexta_ja
nta.c)
  alimentos<-unique(sort(c("Linguiça",  "Virado de milho",  "acelga",
"Laranja",  "Lagarto",  "Abóbora",  "almeirão",  "Sorvete",  "Batata",
"cogumelos","Doce de abóbora",  "Frango",  "Polenta",  "Banana","Frango
assado",  "Berinjela",  "Maça",  "Lombo",  "Couve",  "beterraba",  "Caqui",
"Carne",  "escarola",  "Goiabinha",  "peixe",  "laranja",  "Chuchu",
```

```
"Banana",  "Legumes",  "moyashi",  "Brócolis",  "Pêssego","Carne",
"Repolho",  "beterraba",  "PVT",  "Cocada",  "Frango",  "Mandioca",
"alface",  "Melancia",  "Estrogonofe de frango",  "Batata palha",
"Estrogonofe PVT",  "Banana",  "Bife",  "Cenoura",  "almeirão",  "Sagu",
"Escarola",  "Gelatina",  "Isca de frango",  "Abobrinha",  "Melão",
"Quibebe",  "Pepino",  "Laranja",  "Farofa",  "Curau",  "Flan de chocolate",
"Filé de frango",  "Acelga",  "Maçã")))
  df.itens_alimentares<-data.frame(valor=rep(0,length(alimentos)),
                                   presenca=(rep(NA,length(alimentos)))
  )
  rownames(df.itens_alimentares)<-alimentos
  df.itens_alimentares<-edit(df.itens_alimentares)
  segunda_almoco.df.c<-df.itens_alimentares
  segunda_janta.df.c<-df.itens_alimentares
  terca_almoco.df.c<-df.itens_alimentares
  terca_janta.df.c<-df.itens_alimentares
  quarta_almoco.df.c<-df.itens_alimentares
  quarta_janta.df.c<-df.itens_alimentares
  quinta_almoco.df.c<-df.itens_alimentares
  quinta_janta.df.c<-df.itens_alimentares
  sexta_almoco.df.c<-df.itens_alimentares
  sexta_janta.df.c<-df.itens_alimentares
  list.atribuicao.c<-list("segunda almoco"=segunda_almoco.df.c,"segunda
janta"=segunda_janta.df.c,"terca almoco"=terca_almoco.df.c,"terca
janta"=terca_janta.df.c,"quarta almoco"=quarta_almoco.df.c,"quarta
janta"=quarta_janta.df.c,"quinta almoco"=quinta_almoco.df.c,"quinta
janta"=quinta_janta.df.c,"sexta almoco"=sexta_almoco.df.c,"sexta
janta"=sexta_janta.df.c)
  segunda_almoco.df.q<-df.itens_alimentares
  segunda_janta.df.q<-df.itens_alimentares
  terca_almoco.df.q<-df.itens_alimentares
  terca_janta.df.q<-df.itens_alimentares
  quarta_almoco.df.q<-df.itens_alimentares
  quarta_janta.df.q<-df.itens_alimentares
  quinta_almoco.df.q<-df.itens_alimentares
  quinta_janta.df.q<-df.itens_alimentares
  sexta_almoco.df.q<-df.itens_alimentares
  sexta_janta.df.q<-df.itens_alimentares
  list.atribuicao.q<-list("segunda almoco"=segunda_almoco.df.q,"segunda
janta"=segunda_janta.df.q,"terca almoco"=terca_almoco.df.q,"terca
janta"=terca_janta.df.q,"quarta almoco"=quarta_almoco.df.q,"quarta
janta"=quarta_janta.df.q,"quinta almoco"=quinta_almoco.df.q,"quinta
janta"=quinta_janta.df.q,"sexta almoco"=sexta_almoco.df.q,"sexta
janta"=sexta_janta.df.q)

  a<-data.frame("GFP"=rep(NA,10),
                "melhor_comida"=rep(NA,10),
                "pior_comida"=rep(NA,10),
                row.names=names(list.atribuicao.c))
  tabela_graf.c<-a
  tabela_graf.q<-a
```

```
  if(bandex=="A"){
    for(l in 1:length(list.atribuicao.c))
    {
      for(i in 1:length(df.itens_alimentares[,1]))
      {
        list.atribuicao.c[[l]][,2][i]<-unique(
          sort(
            grepl(rownames(df.itens_alimentares)[i],dia.per.central[[l]])
          )
        )[2]
      }
      list.atribuicao.c[[l]]<-
list.atribuicao.c[[l]][!is.na(list.atribuicao.c[[l]][,2]),]
      list.atribuicao.c[[l]][,2]<-NULL
      tabela_graf.c$melhor_comida[l]<-
rownames(list.atribuicao.c[[l]])[list.atribuicao.c[[l]]==max(list.atribuicao
.c[[l]])]
      tabela_graf.c$pior_comida[l]<-
rownames(list.atribuicao.c[[l]])[list.atribuicao.c[[l]]==min(list.atribuicao
.c[[l]])]
      tabela_graf.c$GFP[l]<-sum(list.atribuicao.c[[l]][,1])
    }
    for(l in 1:length(list.atribuicao.q))
    {
      for(i in 1:length(df.itens_alimentares[,1]))
      {
        list.atribuicao.q[[l]][,2][i]<-unique(
          sort(
            grepl(rownames(df.itens_alimentares)[i],dia.per.quim[[l]])
          )
        )[2]
      }
      list.atribuicao.q[[l]]<-
list.atribuicao.q[[l]][!is.na(list.atribuicao.q[[l]][,2]),]
      list.atribuicao.q[[l]][,2]<-NULL
      tabela_graf.q$melhor_comida[l]<-
rownames(list.atribuicao.q[[l]])[list.atribuicao.q[[l]]==max(list.atribuicao
.q[[l]])]
      tabela_graf.q$pior_comida[l]<-
rownames(list.atribuicao.q[[l]])[list.atribuicao.q[[l]]==min(list.atribuicao
.q[[l]])]
      tabela_graf.q$GFP[l]<-sum(list.atribuicao.q[[l]][,1])
    }
    x11()
    par(mfrow=c(3,4),mar=c(4,5,4,2))
    for(i in 1:length(tabela_graf.c$GFP)){
      graf<-barplot(c(tabela_graf.c$GFP[i],tabela_graf.q$GFP[i]),
ylim=c(min(c(tabela_graf.c$GFP,tabela_graf.q$GFP)),max(c(tabela_graf.c$GFP,t
abela_graf.q$GFP))),
                    xlim=c(0,3),
```

```
                    ylab="GFP",
                    names.arg=c("Central","Química"),
                    main=rownames(tabela_graf.c)[i]
      )
      if(tabela_graf.c$GFP[i]<0 & tabela_graf.q$GFP[i]<0){
        text(x=graf[1],y=2,paste0(sample(c("FUJA!\n","Ai, ai, ai\n","Cade
meu\nomeprazol?\n"),1),tabela_graf.c$pior_comida[i],"!"))
        text(x=graf[2],y=2,paste0(sample(c("FUJA!\n","Ai, ai, ai\n","Cade
meu\nomeprazol?\n"),1),tabela_graf.q$pior_comida[i],"!"))
      }else if(tabela_graf.c$GFP[i]>0 & tabela_graf.q$GFP[i]>0){
        text(x=graf[1],y=2,paste0(sample(c("Ai sim!\n","A larica\nta
pegando\n","Bora come\nlogo,\n"),1),tabela_graf.c$melhor_comida[i],"!"))
        text(x=graf[2],y=2,paste0(sample(c("Ai sim!\n","A larica\nta
pegando\n","Bora come\nlogo,\n"),1),tabela_graf.q$melhor_comida[i],"!"))
      }else if(tabela_graf.c$GFP[i]<0 & tabela_graf.q$GFP[i]>0){
        text(x=graf[1],y=2,paste0(sample(c("FUJA!\n","Ai, ai, ai\n","Cade
meu\nomeprazol?\n"),1),tabela_graf.c$pior_comida[i],"!"))
        text(x=graf[2],y=2,paste0(sample(c("Ai sim!\n","A larica\nta
pegando\n","Bora come\nlogo,\n"),1),tabela_graf.q$melhor_comida[i],"!"))
      }else if(tabela_graf.c$GFP[i]>0 & tabela_graf.q$GFP[i]<0){
        text(x=graf[1],y=2,paste0(sample(c("Ai sim!\n","A larica\nta
pegando\n","Bora come\nlogo,\n"),1),tabela_graf.c$melhor_comida[i],"!"))
        text(x=graf[2],y=2,paste0(sample(c("FUJA!\n","Ai, ai, ai\n","Cade
meu\nomeprazol?\n"),1),tabela_graf.q$pior_comida[i],"!"))
      }else if(tabela_graf.c$GFP[i]==0 & tabela_graf.q$GFP[i]<0){
        text(x=graf[1],y=2,"Ok")
        text(x=graf[2],y=2,paste0(sample(c("FUJA!\n","Ai, ai, ai\n","Cade
meu\nomeprazol?\n"),1),tabela_graf.q$pior_comida[i],"!"))
      }else if(tabela_graf.c$GFP[i]==0 & tabela_graf.q$GFP[i]>0){
        text(x=graf[1],y=2,"Ok")
        text(x=graf[2],y=2,paste0(sample(c("Ai sim!\n","A larica\nta
pegando\n","Bora come\nlogo,\n"),1),tabela_graf.q$melhor_comida[i],"!"))
      }else if(tabela_graf.c$GFP[i]==0 & tabela_graf.q$GFP[i]==0){
        text(x=graf[1],y=2,"Ok")
        text(x=graf[2],y=2,"Ok")
      }else if(tabela_graf.c$GFP[i]>0 & tabela_graf.q$GFP[i]==0){
        text(x=graf[1],y=2,paste0(sample(c("Ai sim!\n","A larica\nta
pegando\n","Bora come\nlogo,\n"),1),tabela_graf.c$melhor_comida[i],"!"))
        text(x=graf[2],y=2,"Ok")
      }else if(tabela_graf.c$GFP[i]<0 & tabela_graf.q$GFP[i]==0){
        text(x=graf[1],y=2,paste0(sample(c("FUJA!\n","Ai, ai, ai\n","Cade
meu\nomeprazol?\n"),1),tabela_graf.c$pior_comida[i],"!"))
        text(x=graf[2],y=2,"Ok")
      }
    }
    invisible(graf)
    par(mfrow=c(1,1))
  }else if(bandex=="C"){
    for(l in 1:length(list.atribuicao.c))
    {
      for(i in 1:length(df.itens_alimentares[,1]))
```

```
      {
        list.atribuicao.c[[l]][,2][i]<-unique(
          sort(
            grepl(rownames(df.itens_alimentares)[i],dia.per.central[[l]])
          )
        )[2]
      }
      list.atribuicao.c[[l]]<-
list.atribuicao.c[[l]][!is.na(list.atribuicao.c[[l]][,2]),]
      list.atribuicao.c[[l]][,2]<-NULL
      tabela_graf.c$melhor_comida[l]<-
rownames(list.atribuicao.c[[l]])[list.atribuicao.c[[l]]==max(list.atribuicao
.c[[l]])]
      tabela_graf.c$pior_comida[l]<-
rownames(list.atribuicao.c[[l]])[list.atribuicao.c[[l]]==min(list.atribuicao
.c[[l]])]
      tabela_graf.c$GFP[l]<-sum(list.atribuicao.c[[l]][,1])
    }
    x11()
    graf.c<-barplot(tabela_graf.c$GFP,
                    ylim=c(min(tabela_graf.c$GFP),max(tabela_graf.c$GFP)+1),
                    xlim=c(0,length(tabela_graf.c$GFP)+3),
                    ylab="GFP",
                    names.arg=c("seg\n almoco","seg\n janta","ter\n
almoco","ter\n janta","qua\n almoco","qua\n janta","qui\n almoco","qui\n
janta","sex\n almoco","sex\n janta"),
                    main="Bandejão Central"
    )
    for(i in 1:length(tabela_graf.c$GFP))
    {
      if(tabela_graf.c$GFP[i]<0){
        text(x=graf.c[i],y=max(tabela_graf.c$GFP)+0.5,
paste0(sample(c("FUJA!\n","Ai, ai, ai\n","Cade
meu\nomeprazol?\n"),1),tabela_graf.c$pior_comida[i],"!"))
      } else if(tabela_graf.c$GFP[i]==0){
        text(x=graf.c[i],y=max(tabela_graf.c$GFP)+0.5, "Ok\n")
      } else{
        text(x=graf.c[i],y=max(tabela_graf.c$GFP)+0.5, paste0(sample(c("Ai
sim!\n","A larica\nta pegando\n","Bora
come\nlogo,\n"),1),tabela_graf.c$melhor_comida[i],"!"))
      }
    }
    invisible(graf.c)
  }else if(bandex=="Q"){
    for(l in 1:length(list.atribuicao.q))
    {
      for(i in 1:length(df.itens_alimentares[,1]))
      {
        list.atribuicao.q[[l]][,2][i]<-unique(
          sort(
```

```
            grepl(rownames(df.itens_alimentares)[i],dia.per.quim[[l]])
          )
        )[2]
      }
      list.atribuicao.q[[l]]<-
list.atribuicao.q[[l]][!is.na(list.atribuicao.q[[l]][,2]),]
      list.atribuicao.q[[l]][,2]<-NULL
      tabela_graf.q$melhor_comida[l]<-
rownames(list.atribuicao.q[[l]])[list.atribuicao.q[[l]]==max(list.atribuicao
.q[[l]])]
      tabela_graf.q$pior_comida[l]<-
rownames(list.atribuicao.q[[l]])[list.atribuicao.q[[l]]==min(list.atribuicao
.q[[l]])]
      tabela_graf.q$GFP[l]<-sum(list.atribuicao.q[[l]][,1])
    }
    x11()
    graf.q<-barplot(tabela_graf.q$GFP,
                    ylim=c(min(tabela_graf.q$GFP),max(tabela_graf.q$GFP)+1),
                    xlim=c(0,length(tabela_graf.q$GFP)+3),
                    ylab="GFP",
                    names.arg=c("seg\n almoco","seg\n janta","ter\n
almoco","ter\n janta","qua\n almoco","qua\n janta","qui\n almoco","qui\n
janta","sex\n almoco","sex\n janta"),
                    main="Bandejão Química" #o título do gráfico
    )

    for(i in 1:length(tabela_graf.q$GFP))
    {
      if(tabela_graf.q$GFP[i]<0){
        text(x=graf.q[i],y=max(tabela_graf.q$GFP)+0.5,
paste0(sample(c("FUJA!\n","Ai, ai, ai\n","Cade
meu\nomeprazol?\n"),1),tabela_graf.q$pior_comida[i],"!"))
      } else if(tabela_graf.q$GFP[i]==0){
        text(x=graf.q[i],y=max(tabela_graf.q$GFP)+0.5, "Ok\n")
      } else{
        text(x=graf.q[i],y=max(tabela_graf.q$GFP)+0.5, paste0(sample(c("Ai
sim!\n","A larica\nta pegando\n","Bora
come\nlogo,\n"),1),tabela_graf.q$melhor_comida[i],"!"))
      }
    }
    invisible(graf.q)
  }
}
```

Arquivos

função com comentários:bandex.r