BÁRBARA P. H. RIGHETTI

Mestranda do programa de Bioquímica da Universidade Federal de Santa Catarina PPG BQA- UFSC

Currículo lattes

MEUS EXERCÍCIOS

exercícios

Trabalho final

Proposta 1

A proposta visa realizar múltiplos testes de regressão linear, par a par, para diversas variáveis contínuas, preditoras e respostas, dando ao usuário a opção de “dados normais” ou “dados não normais”.

O input da função deve ser um data.frame com as observação nas linhas, e variáveis preditoras e respostas nas colunas. E o usuário deve informar quais colunas referem-se a variáveis preditoras e quais referem-se a variáveis respostas.

Para “dados normais”, a função realizará a avaliação da normalidade e homocedasticidade dos dados, retornando uma janela gráfica com histograma e curva de distribuição normal, gráfico Q-Q plot além do resultado para Shapiro-Wilk test.

A janela deve perguntar ao usuário se, com base nos resultados acima, deseja seguir com o argumento “dados normais” ou se deseja modificar para “dados-não normais”.

Para “dados normais” a função seguirá realizando regressões lineares simples, estabelecendo o melhor modelo linear que explique aquela distribuição de dados.

Para dados não normais a função teste realizará regressões lineares para simulações de distribuição aleatória dos dados observados, comparando a ocorrência destes valores ao valor obtido a partir dos dados observados a fim de determinar se a o valor de regressão obtido é significativo.

De ambas os tipos de avaliação surgirão os modelos lineares para cada relação variável preditora/resposta. Os modelos que tratem da mesma variável resposta serão comparados sucessivamente via ANOVA para que seja encontrado a variável preditora que melhor explica a distribuição de dados observada.

O output final da função será uma lista contendo: um data.frame com o par de variáves avaliado na primeira coluna e o valor de r^2 para cada relação na segunda coluna, um data.frame com os modelos mais relevantes para cada variável resposta, com seu respectivo r^2 , o “call” do modelo explicitando qual variável preditora está envolvida no modelo, além dos coeficientes associados ao modelo, e um plot de todos os gráficos referentes a estes melhores modelos (1 para cada variável resposta).

> Oi Bárbara,

> Acho que a proposta tem futuro mas ela ainda está muito solta. Todas essas coisas são interessantes de serem feitas (menos testes estatísticos de normalidade, esses não servem pra nada. Veja esse post: http://allendowney.blogspot.nl/2013/08/are-my-data-normal.html.
>
> A ideia de uma função não é ser um script completo que automatiza completamente uma analise, mas uma peça identificável da analise, que eventualmente possa ser usada em vários contextos, ou no minimo várias vezes dentro de uma mesma analise. Testar normalidade, fazer gráficos diagnósticos, escolher um modelo, rodar o modelo, diagnosticar o modelo e comparar modelos é coisa demais pra uma função só. Sugiro que vc foque num aspecto só do processo de modelagem, e não tente automatizar tudo. Frequentemente tem muita subjetividade na escolha de um modelo, e principalmente muita dependência na pergunta que vc está interessada em responder. É muito difícil automatizar isso.
>
> Se vc quer uma função que te ajude a diagnosticar problemas num conjunto de dados (por exemplo não normalidade, outliers, caudas longas...) e que depois vai te ajudar a escolher o modelo apropriado, acho bacana. Se vc quer fazer uma função que ajusta um modelo linear diferente das funções padrão, fazendo uma analise de significância não paramétrica, acho legal tb.
>
> Agora, juntar tudo numa coisa só eu acho demais.
>
> Também não ficou claro se vc espera receber um conjunto de variáveis preditoras e resposta e depois rodar todos os modelos possíveis em cima disso, ou pedir para o usuário especificar o modelo. Se for o primeiro caso, totalmente automático, eu sou contra. Acho que não é assim que devemos responder perguntas. Se for o segundo, ótimo, acho que um modelo só deveria ser ajustado depois que ele é entendido.
>
> Tente reformular a proposta de forma mais pontual, e qq coisa pode perguntar no forum ou aqui.
>
> Valeu!
>
> —-*Ogro*

Concordo com a questão da normalidade, sim, retirarei ela da proposta. Quanto aos modelos, a idéia seria sim receber um conjunto de variáveis preditoras e outras respostas e testá-las par a par, estabelecendo modelos lineares simples (uma variável preditora apenas), e não pré-estabelecer os modelos. Na verdade, acho que não entendi em que consistiria pré-estabelecer os modelos, considerando que não seriam regressões múltiplas. Seriam vários testes de correlação, por assim dizer. Você acha possível?

> Pré-estabelecer o modelo é escolher o modelo de acordo com a pergunta que vc quer fazer. Dependendo do objetivo da modelagem estatística o modelo vai mudar.

> Se o objetivo é quantificar o efeito de determinada variável ou conjunto de variáveis numa variável resposta, essas variáveis deveriam ser estabelecidas e medidas antes de se iniciar o experimento, e expectativas e previsões claras deveriam ser feitas antes de fazer a modelagem. Sem isso, vira uma festa da caça à significância. Sempre vai ter alguma coisa < 0.05 pra se pescar num conjunto de dados, e sem uma perguntar e hipótese clara a priori isso não significa nada. algumas referencias sobre o tema: http://pss.sagepub.com/content/22/11/1359 e http://www.stat.columbia.edu/~gelman/research/unpublished/p_hacking.pdf
>
> Se o objetivo é preditivo, um modelo de regressão múltipla quase sempre é mais apropriado, e deveria ser validado num conjunto de dados diferente do que foi usado para ajustar o modelo.
>
> É por isso que eu acho a ideia da função que testa tudo par a par usando modelos lineares um pouco fraca, é uma péssima prática estatística.
>
> Mas isso é com vc, se vc achar interessante, acho que dá pra pensar numa função interessante com essa temática. Talvez uma função que calcule uma matriz de correlação, ou outra medida de associação qualquer, e faça uma visualização bacana dessas relações? Scatterplots e um heatmap das correlações? Ferramentas de visualização desse tipo são sempre muito úteis.
>
> Valeu!
>
> —-*Ogro*

PROPOSTA 1 A proposta visa realizar múltiplos testes de correlação, par a par, para diversas variáveis contínuas, preditoras e respostas. O input da função deverá ser um data.frame com as observação nas linhas, e variáveis preditoras e respostas nas colunas. O usuário deve informar quais colunas referem-se a variáveis preditoras e quais referem-se a variáveis respostas. A função realizará testes de correlação sucessivos entre as variáveis preditoras e variáveis respostas, gerando como output uma matriz de correlação e um heat map associado, a fim de visualizar os valores de correlação mais significativos.

PROPOSTA 1 - FUNÇÂO multC

multC←function(x, pred, res, #x é data.frame; args. pred e res são indicam colunas de variáveis preditoras e respostas, devem estar no formato de sequência numérica normal=TRUE, #arg. normal determina se distribuição dos dados é normal ou não, default é “TRUE” pal=c(“blue”,“white”,“red”), #arg. pal permite criação de paletas de cores personalizadas, default é c(“blue”,“white”,“red”) main.h=NULL, #arg. main.h refere-se ao título para heat map xlab.h=NULL, #arg. xlab.h refere-se ao título do eixo 'x' para heat map ylab.h=NULL) #arg. ylab.h refere-se ao título do eixo 'y' para heat map {

```
result<-list()
if(is.null(pred))
stop("pred not specified")                                    #parar a
função se não houver especificação quanto à preditoras
if(is.null(res))
stop("res not specified")                                     #parar a
```

```
função se não houver especificação quanto à respostas
    na.method<-pmatch(normal, c("TRUE","T","FALSE","F"))
#estabelece possibilidades de escritas para arg. 'normal'
    if(is.na(na.method))                                          #comando
para parar a função caso nenhuma das possibilidades de escrita para o arg.
'normal' ocorra
        stop ("invalid 'normal' argument")
    xa<-na.omit(x)                                                #comando
para transformar data.frame dos dados, retirando os NAs
        if(normal=="TRUE"| normal=="T")                          #Estabelece
condição de arg. 'normal' ser "TRUE" ou "T"
        {
        tmp<-cor(xa[,pred], xa[,res], method="pearson")         #cria a
matriz de correlação com base nas colunas informadas como preditoras e
resposta, para condição de arg 'normal'=="TRUE" ou "T"
        }
    if(normal=="FALSE"|normal=="F")
#Estabelece condição de arg 'normal' ser "FALSE" ou "F"
        {
        tmp<-cor(xa[,pred], xa[,res], method="spearman")        #cria a
matriz de correlação com base, para arg. 'normal'=="FALSE" ou "F"
        }
mat<-tmp                                                          #cria o
objeto mat com base nas matrizes de correlação criadas acima
colors<-colorRampPalette(pal)                                     #cria a
paleta de cores personalizada, com base no vetor fornecido no arg. 'pal'
heat<-heatmap(tmp, margins=c(6,5),col=colors(100), scale="none",    #cria o
heat.map com base na matriz de correlação e paleta de cores, não escalonando
dados
          main=main.h, xlab=xlab.h, ylab=ylab.h)                       #e
utilizando args. para título principal e título de eixos

x11()                                              #abre janela gráfica nova
p<-par(mfrow=c(length(res), length(pred)), mar=c(1,1,1,0.5),        #cria
objeto para os scatterplots
    mgp=c(0.1,0.1,0), tcl=0, xaxt="n", yaxt="n")
        for(i in 1:length(x[,res]))                               #inicia o
loop para os scatterplots, que deve repetir, para cada variável resposta:
           {
           for(j in 1:length(x[,pred]))                        #um novo loop,
para cada variável preditora, repetindo
               {
               plot(x[,res[i]],x[,pred[j]],                #a criação de um
plot, que relacione cada preditora com cada resposta
                   xlab=colnames(x[pred[j]]),            #estabelece nome
do eixo x
                   ylab=colnames(x[res[i]]))             #estabelece nome
do eixo y
               abline(lm(x[,res[i]]~x[,pred[j]]),          #cria uma linha
de tendência linear de distribuição dos dados para cada correlação
```

```
                col="red", lwd=2)
                }
            }
return(list(mat,heat,p))                                                #retorna
uma lista com a matriz de correlação, um heat map associado e uma janela
gráfica com scatterplots
```

```
}
```

multC

HELP multC

```
multC {base}                                              R Documentation

Scatterplot and Heat Map for specified variables

Description:

Calculates a correlation matrix for various predicted and response variables
as specified by the user and provides an associated heat map and a
scatterplot matrix for all possible predicted and response variables
combinations.

Usage:
        multC<-function(x, pred, res, normal=TRUE,
    pal=c("blue","white","red"),main.h=NULL,
    xlab.h=NULL, ylab.h=NULL)

Arguments:
x       the data.frame from which the data are to be read from;         must
be arranged with cases in rows and variables in
        columns
pred        a sequence of integers corresponding to the columns of
the data.frame (x) to be used as predictor variables
res     a sequence of integers corresponding to the columns of          the
data.frame(x) to be used as response variables
normal      argument specifying if data distribution is to be
        considered normal (=”TRUE”) or not-normal (=”FALSE”).
Defaults to “TRUE”.
pal     character vector indicating colors to be used for heat
map. Defaults to c(“blue”,”white”,”red”).
main.h      character vector indicating main title for heat.map.
Defaults to NULL
xlab.h      character vector indicating title for x-axis of
        heat.map. Defaults to NULL
ylab.h      character vector indicating title for y-axis of
        heat.map. Defaults to NULL

Value:
     Returns a list with three components
```

```
  comp1 : correlation matrix of pred and res specified columns of x
  comp2 : heat map associated with correlation matrix from comp1
  comp3 : Scatterplot matrix for all possible combinations of predicted and
response variables

Author(s):
     Contributed by BSc. Barbara P. H. Righetti

References:
Crawley, M.J. The R Book. John Wiley and Sons, England. 2007

See Also:
colorRampPalette, for color palette specification
cor, for correlation test and requirements

Examples:
data<-data.frame(replicate(10,rnorm(20,0,1)))
multC(data, pred=1:5, res=6:10, normal=TRUE, pal=c("green", "blue", "grey"))
```

Proposta 2

A função proposta irá ranquear variáveis quanto sua estabilidade entre distintos grupos e/ou tratamentos, em diferentes tempos, da mais estável para a menos estável.

O input da função deverá ser um data frame com identificação das amostras, variáveis categóricas (grupo e tempo), e valores de variáveis contínuas a serem avaliadas quanto à estabilidade.

A função calculará a variação intragrupo e intergrupo das variáveis em questão, levando em consideração para o estabelecimento de grupos as combinações possíveis entre as variáveis categóricas.

Em seguida, será calculada uma medida de estabilidade para cada variável, com base nos valores de variação obtidos acima.

O output da função será um data frame com as variáveis avaliadas e seus respectivos valores de estabilidade, ranqueados do mais ao menos estável.

> Não entendi oq é estabilidade. É só a variância dentro de grupo?
>
> Faltam elementos pra eu conseguir avaliar essa função, mas a primeira me parece mais interessante no geral.
>
> —-*Ogro*

Seriam avaliadas as variâncias dentro do grupo e entre os grupos e tempos. A fim de obter variáveis que variem menos, sejam mais estáveis ou constantes dentre os dados. Acho esta mais interessante,

embora ainda esteja trabalhando em uma fórmula apropriada. Re-escrevo esta também? Ou invisto apenas na proposta 1, acima?

> Qual o objetivo disso? Achar variáveis que sejam identificadoras de cada grupo? Se for isso, talvez vc goste de tentar implementar uma analise de variáveis canônicas, que procura exatamente isso: variáveis que maximizem a diferença entre grupos pré-determinados. Talvez seja um pouco avançado nesse momento, maximizar uma coisa num lugar e minimizar em outro é um problema simples, mas vc precisa saber um pouco de algebra linear.
>
> —-*Ogro*

PROPOSTA 2 A função proposta irá ranquear variáveis quanto sua variabilidade entre distintos grupos e/ou tratamentos, em diferentes tempos, a fim de obter as variáveis mais constantes entre tratamentos, grupos e tempos. O input da função deverá ser um data frame com identificação das amostras nas linhas, variáveis categóricas (grupo e tempo), e valores de variáveis contínuas a serem avaliadas quanto nas colunas.

A função terá um argumento STD, que permitirá ao usuário, quando STD=TRUE, sinalizar a necessidade de normalização dos dados previamente aos demais cálculos da função. Se STD=TRUE a normalização será realizada através da centralização dos dados pela média da variável, seguida de um ajuste pelo desvio padrão da mesma variável, para todas as variáveis resposta. Se STD=FALSE, a função seguirá diretamente para o cálculo das variâncias.

Primeiramente será calculada a variância (w) de cada variável resposta dentro dos tratamentos existentes, classificados de acordo com as combinações possíveis de tratamento/tempo (Por exemplo, se houverem 3 tratamentos, sendo avaliados em 3 tempos distintos, serão formados 9 grupos). Como resultado, obteremos ‘x’ valores de ‘w’ para cada variável, onde ‘x’ é o número de grupos. Em seguida será calculada a variância (z) de cada variável resposta entre grupos, com base na média total da variável e as médias obtidas para cada grupo.

O valor de estabilidade (E) de cada variável será calculada através da inversa do resultado obtido para: média aritmética dos valores de ‘w’ multiplicado pelo valor de z. Serão consideradas mais estáveis as variáveis com maiores valores de estabilidade (E).

O output da função será um data frame com as variáveis avaliadas e seus respectivos valores de E, ranqueados do mais ao menos estável.