Marina Puglisi

Pós-doutoranda em Ciências da Reabilitação pela FMUSP. Interesses de pesquisa: Linguagem e Cognição, Desenvolvimento de Linguagem, Distúrbios de Linguagem.

exec

Trabalho final

Principal

A análise de cluster é uma técnica multivariada interessante para explorar os dados e agrupar indivíduos em função de seu comportamento. Há uma série de representações gráficas usadas para ilustrar os modelos de clusters gerados ao final da análise, mas nenhum, até onde eu sei, sintetiza de forma clara todas as informações relevantes. Veja as principais vantagens e desvantagens de cada tipo de gráfico:

Dendrograma

- Vantagens: Demonstra de forma muito clara quais observações foram agrupadas para formar os clusters e a sequência de clusters formada a partir de cada observação; promove um índice de distância para cada cluster.
- Desvantagens: cada etapa de formação dos clusters é apresentada de forma muito confusa, sendo difícil detectar quais clusters foram formados em cada etapa.
- Ex. dendrograma

Vertical Icicle

- Vantagens: Todas as informações são apresentadas (quais cluster foram formados em cada etapa; quantas observações compõem cada cluster em cada etapa).
- Desvantagens: a relação dados/ tinta é baixa, pois o gráfico é muito poluído e necessita de explicação para que seja compreendido por um leigo.
- Ex. vertical icicle

Cluster membership

- Vantagens: define claramente quais observações compõem cada cluster em cada etapa.
- Desvantagens: é apresentado em forma de tabela, sendo de difícil visualização.
- Ex. cluster membership

Minha proposta

Criar um dendrograma interativo no qual seja possível representar:

- Como as observações foram agrupadas em clusters, em cada etapa;
- Qual o índice de distância padronizado em cada nova formação de cluster;
- Quais clusters foram formados em cada etapa, bem como quais observações compõem estes clusters (esta seria a função interativa). Abaixo do dendrograma, será criada uma legenda com o número de clusters desejado. Cada vez que clicamos em um número, visualizamos a respectiva formação de clusters no dendrograma separada por cores diferentes.

Comentários da Proposta Principal

A idéia é muito boa, mas creio que não é factível no tempo que você tem disponível. Como solução sugiro que você execute parte da função, como criar o dendrograma e talvez representar os índices de distância padronizados em cada nova formação de cluster. Até aí acho que já vai te dar um bom trabalho. Como a função será útil para você depois, conclua essa etapa para a disciplina e finalize a função posteriormente. Você já sabe que dados dará à função e o que ela te retorna, agora tente pensar passo-a-passo como o processo será realizado. De qualquer maneira, o plano B parece mais promissor pelo tempo que você tem.

Gabriel

===== Plano B ===== Utilizar a função “simula” para construir funções específicas que demonstrem como a violação de determinadas premissas pode afetar o resultado de um teste específico. Esta função foi idealizada para testar o efeito da utilização de um teste inapropriado para a amostra coletada (“a posteriori”). Os testes que entrarão na função serão: Teste-T, “ANOVA”, correlação de Pearson, regressão simples e múltipla. As premissas testadas serão: - Violação da normalidade. - Violação da homogeneidade de variância. - Violação da multicolinearidade (apenas para a regressão múltipla) Os argumentos da função exigirão a especificação do teste e os dados da amostra coletada (o objeto em formato data.frame). O output apresentará o parâmetro que foi violado e a comparação dos resultados do teste com uma amostra de mesma média e dp, que cumpra os pré-requisitos do teste em questão. == Comentários == Muito legal a idéia, é mais simples que sua proposta inicial e parece caber melhor no tempo que você tem disponível. A princípio eu seguiria com essa e faria para um teste apenas. Se te sobrar tempo, faça para o segundo, e depois para um terceiro. **Gabriel Após a entrega:** muito legal a função,e bem fundamentada num conceito essencial de análise de dados. Apenas um detalhe: por que reduzir o número de casas decimais dos quantis teóricos (uso da função round'')?

Minhas funções

Concordei com os comentários do Gabriel (obrigada, Gabriel!) e acabei optando por realizar o plano B. Meu foco principal foi testar os efeitos da violação do parâmetro de normalidade sobre a probabilidade do teste de hipótese (especificamente do teste t). Para isso achei necessário elaborar duas “subfunções”: a diagnos() e a simula.t().

diagnos()

Esta função foi realizada para ajudar o pesquisador a visualizar qual distribuição melhor se ajusta aos dados de sua amostra. É uma função simples que exige um único argumento (um vetor que contenha os dados amostrais) e retorna quatro Q-Q plots (para quatro diferentes distribuições).

```
diagnos <- function(x)
    {
    ### recriando x a partir dos dados inseridos pelo usuário
    x <- sort(round(x,2))
    ### criando os percentis para uma amostra x de length(x)
    px <- (1:length(x))/length(x)
    ### criando os quantiles esperados (teóricos) para cada distribuição
    q.normal <- round(qnorm(px, mean=mean(x), sd=sd(x)),2)
    q.uniforme <- round(qunif(px, min=min(x), max=max(x)),2)
    q.agregada <- round(qnbinom(px, size=max(x), mu=mean(x)),2)
    q.poisson <- round(qpois(px, lambda=mean(x)),2)
    qqplot.x <- data.frame(perc=px, q.obs=x, q.norm=q.normal,
q.unif=q.uniforme,
        q.agreg=q.agregada, q.pois=q.poisson)
    ### criando os Q-Q plots
    x11()
    par(mfrow=c(2,2))
    plot(q.obs~q.norm, data=qqplot.x, xlab="Quantis Esperados",
ylab="Quantis Observados",
    main="Q-Q Normal")
    abline(0,1, col="red")
    plot(q.obs~q.unif, data=qqplot.x, xlab="Quantis Esperados",
ylab="Quantis Observados",
    main="Q-Q Uniforme")
    abline(0,1, col="red")
    plot(q.obs~q.agreg, data=qqplot.x, xlab="Quantis Esperados",
ylab="Quantis Observados",
    main="Q-Q Agregada")
    abline(0,1, col="red")
    plot(q.obs~q.pois, data=qqplot.x, xlab="Quantis Esperados",
ylab="Quantis Observados",
    main="Q-Q Poisson")
    abline(0,1, col="red")
    }
```

simula.t()

Sabendo qual a distribuição mais adequada para os seus dados, o pesquisador pode então inserir esta informação na função “simula.t”. Esta função permite que se verifique o efeito da violação do parâmetro de normalidade na probabilidade t. A função exige 3 argumentos (VD, VI, distribuição dos dados) e retorna a probabilidade t e a probabilidade do teste alternativo, gerado a partir de simulações com a mesma distribuição da amostra. Caso nenhuma das quatro distribuições seja adequada para os dados (dist=“none”), o teste alternativo consiste em simulações com permutação. Todos os testes são bicaudais.

```
simula.t <- function(x, y, dist=c("norm", "unif", "aggreg", "pois", "none"))
    {
    m.grupos <- aggregate(x~y, FUN=mean)
    n.grupos <- aggregate(x~y, FUN=length)
    v.grupos <- aggregate(x~y, FUN=var)
    dif.med <- abs(m.grupos$x[2] - m.grupos$x[1])
    var.grupos <-
sqrt((v.grupos$x[1]/n.grupos$x[1])+(v.grupos$x[2]/n.grupos$x[2]))
    t.x <- dif.med/var.grupos
    p.ttest <- (pt(t.x, df=length(x)-2, lower.tail=F))*2
    ## p.ttest é a própria probabilidade gerada pelo teste t
    sim.dif=rep(NA,1000)
    if(dist=="norm")
        {
        for (i in 1:1000)
            {
sim.dif[i]=round(abs(mean(rnorm(n.grupos$x[1],mean=mean(x),sd=sd(x)))-
            mean(rnorm(n.grupos$x[2],mean=mean(x),sd=sd(x)))),1)
            }
        }
    if(dist=="unif")
        {
        for (i in 1:1000)
            {
sim.dif[i]=round(abs(mean(runif(n.grupos$x[1],max=max(x),min=min(x)))-
            mean(runif(n.grupos$x[2],max=max(x),min=min(x)))),1)
            }
        }
    if(dist=="aggreg")
        {
        for (i in 1:1000)
            {
sim.dif[i]=round(abs(mean(rnbinom(n.grupos$x[1],size=max(x),mu=mean(x)))-
            mean(rnbinom(n.grupos$x[2],size=max(x),mu=mean(x)))),1)
            }
        }
    if(dist=="pois")
        {
```

```
        for (i in 1:1000)
            {
            sim.dif[i]=round(abs(mean(rpois(n.grupos$x[1],lambda=mean(x)))-
            mean(rpois(n.grupos$x[1],lambda=mean(x)))),1)
            }
        }
    if(dist=="none")
        {
        for(i in 1:1000)
            {
            sim.dif[i]=abs(diff(tapply(sample(x),y,mean)))
            }
        }
    p.simtest <- (sum(sim.dif>=dif.med))/length(sim.dif)
    ## p.simtest é a probabilidade gerada pelo teste alternativo, que leva
em conta a
    ## distribuição empírica
    cat("ambos os testes são bicaudais\n")
    return(list("p teste t"=p.ttest, "p teste alternativo"=p.simtest))
    }
```

Help das minhas funções

diagnos()

```
Diagnosis{not availabe package}
                Distribution Diagnosis
Description:
    This function generates Q-Q plots that compare the empirical data with
theoretical samples that come from populations with different distributions
(normal, uniform, aggregate and poisson).
Usage:
    diagnos(x)
Arguments:
    x    a vector that represents the dependent variable
Details:
    Default estimate parameters for generating the theoretical quantiles
were:
    Aggregate distribution: size=max(x), mu=mean(x)
    Poisson distribution: lambda=mean(x)
Author(s):
    Marina Puglisi
    marinapuglisi@usp.br
References:
    Vito Ricci. Fitting distributions with R. Release 0.4-21 February 2005
    Documentation License, Version 1.2
    http://www.fsf.org/licenses/licenses.html#FDL
```

```
    Chi Yau. R Tutorial: An R Introduction to Statistics.
    http://www.r-tutor.com/
See also:
    qqplot for creating different kinds of Q-Q plots.
Examples:
    x.norm <- rnorm(200,30,5)
    x.unif <- runif(200,11.2,49.7)
    x.agreg <- rnbinom(200,size=50,mu=30)
    x.contag <- rpois(200,lambda=30)
    diagnos(x.norm)  # simulating a sample with normal distribution
    diagnos(x.unif)  # simulating a sample with uniform distribution
    diagnos(x.agreg)  # simulating a sample with aggregated distribution
    diagnos(x.contag)  # simulating a sample with poisson distribution
```

simula.t()

```
Simula.t{not availabe package}
                Simultation of the violation of normal distribution on t
probability
Description:
    This function simulates the violation of the assumption of normal
distribution.
    It provides the probabilities of finding differences between two groups
based on two theoretical simulations (one from the t-distribution and one
from the same distribution of the empirical sample).
Usage:
    simula.t(x,y,dist="norm")
Arguments:
    x    a vector that represents the dependent variable
    y    a vector or a factor that represents the independent variable
    dist    the distribution from wich the sample might come from
("norm","unif","aggreg","pois","none")
Details:
    Default estimate parameters for generating the theoretical samples were:
    Aggregate distribution: size=max(x), mu=mean(x)
    Poisson distribution: lambda=mean(x)
    The t probability and the alternative probability are considered for
two-tailed tests.
Author(s):
    Marina Puglisi
    marinapuglisi@usp.br
References:
    Everitt BS, Hothorn T. A Handbook of Statistical Analyses Using R. Ed:
Chapman and Hall/CRC, 2 edition, 2009.
See also:
    simula.r for simulating normal samples
Examples
    x.norm <- rnorm(200,30,5) # teste1: para a simulação de uma amostra com
```

```
dist. normal
    x.unif <- runif(200,11.2,49.7) # teste2: para a simulação de uma amostra
com dist.uniforme
    x.agreg <- rnbinom(200,size=50,mu=30) # teste3: para a simulação de uma
amostra com dist.agregada
    x.contag <- rpois(200,lambda=30) # teste4: para a simulação de uma
amostra com dist.poisson
    x.exp <- rexp(200,rate=1/30) # teste5: para a simulação de uma amostra
com outra distribuição (exponencial)
    y <- c(rep("grupo1", each=100), rep("grupo2", each=100))
    simula.t(x.norm,y,dist="norm")
    simula.t(x.unif,y,dist="unif")
    simula.t(x.agreg,y,dist="aggreg")
    simula.t(x.contag,y,dist="pois")
    simula.t(x.exp,y,dist="none")
```

Arquivos .r e .txt

diagnos Help diagnos simula.t Help simula.t

Comentários finais

Tive algumas dúvidas ao elaborar a função e tentei resolver da melhor forma possível, mas continuo sem resposta para algumas questões... agradeceria se puder ter um feedback da função, assim como das dúvidas específicas :)

1. Parâmetros estimados: assumi size=max(x), mu=mean(x) na d. agregada; e lambda=mean(x) na d. poisson. Achei que para uma simulação genérica, esses seriam os melhores parâmetros. Concordam?
2. Os Q-Q plots das dist. agregada e poisson nunca ficam muito bom, mesmo quando as amostras experimentais vêm destas distribuições...
3. Fiz a simulação (usando simula.t) com várias amostas diferentes (além das que deixei no exemplo do help). O resultado final é que normalmente não há diferenças entre o resultado do teste t e do teste alternativo, ou seja, o teste t parece ser relativamente robusto à violação da distribuição normal. Quando as diferenças entre os grupos são muito sutis, o teste alternativo é mais sensível, e o teste t leva a maiores chances de cometer erros tipo II. Eu pensei nesta função porque me preocupa muito esta idéia de usar um teste estatístico de maneira inapropriada, mas a moral da história é que no fundo o problema nem é tão grave assim... será???

Obrigada a toda a equipe da disciplina!!