

Riguel F Contente

✖ Doutorando em Oceanografia Biológica - IOUSP

Proposta principal

Mudanças alométricas relacionadas ao tamanho corpóreo podem afetar comparações de determinadas grandezas morfométricas entre populações. Lleonart et al. 2000 desenvolveram uma metodologia de remover esse efeito alométrico, a qual consiste na seguinte normalização: $Z=Y(Tlo/Tl)^b$, onde Y =valor da variável (e.g. tamanho da mandíbula), Tl = comprimento máximo do exemplar (em peixes, por exemplo, comprimento total), Tlo =representa um valor de referência (e.g. média ou mediana do comprimento total) através do qual todos os indivíduos são reduzidos. De um modo geral, o método remove do modelo o efeito do coeficiente 'a' ao qual se associa as informações de alometria. Pretendo realizar uma função que efetue essa normalização, produza gráficos de comparações de grandezas entre duas amostras (ex. boxplot, médias) e realize um teste de hipótese (teste-t ou anova) para avaliar possíveis diferenças significativas.

Referência

LLEONART J.; SALAT J.; TORRES G.J. Removing Allometric Effects of Body Size in Morphological Analysis Journal of Theoretical Biology, Volume 205, Number 1, 7 July 2000, pp. 85-93(9)

Comentários

Paulo

1. Faça como o Fulano: formate a página no padrão que indicamos, po favor!
2. Legal a proposta, e bem dimensionada. Apenas lembre que um teste t ou ANOVA tm várias premissas, e seria legal pelo indicar ao usuário se estas premissas parecem se cumprir com os dados.

Página de Ajuda

RemAllom

package:nenhum

R Documentation

Teste de alometria e comparação de grandezas morfométricas com remoção do efeito da alometria

Description:

Realiza teste-t para avaliar possíveis desvios do crescimento isométrico, realiza ANOVA para testar se há relação entre variável resposta e preditora,

remove potenciais efeitos de alometria através de uma normalização e, utilizando ANOVA, compara simultaneamente médias das estruturas morfométricas entre duas amostras/populações/espécies. Produz gráficos dos modelos alométricos e box-plots de comparação entre as amostras/populações/espécies. Exporta as matrizes com dados normalizados para cada população/espécie para uso em posteriores análises multivariadas.

Usage:

```
RemAllom<-  
function(A,B,coef.A,coef.B,nomes,diag.allom=TRUE,TL0media=TRUE,diag.norm=TRUE,WT=TRUE)
```

Arguments:

A: data.frame ou matriz da pop/espécie A. Estruturas morfométricas nas colunas e indivíduos nas linhas
B: data.frame ou matriz da pop/espécie B. Mesmo formato
coef.A: vetor numérico contendo os coeficientes b esperados para cada variável em A
coef.B: vetor numérico contendo os coeficientes b esperados para cada variável em B
nomes: objeto contendo a denominação para cada amostra/população/espécie
diag.allom: lógico. Plotar gráficos diagnósticos para avaliação das premissas de regressão linear?
TL0media: lógico. Reduzir os valores pela média? Se FALSE, será pela mediana
diag.norm: lógico. Plotar gráficos exploratórios para avaliação de normalidade e homogeneidade das variâncias?
WT: lógico. Exportar em .txt matrizes com valores normalizados?

Details:

Através do test-t, testa-se a hipótese nula de que o coeficiente b do modelo alométrico log-transformado (ln) é igual ao valor esperado de acordo com a dimensão da variável morfométrica em questão. Para comparação das variáveis entre duas populações/espécies, os dados são normalizados pela média ou mediana do comprimento do tamanho corpóreo, segundo a expressão: $Z=Y(Tl_0/Tl)^b$, onde Y=valor da variável, Tl= comprimento corpóreo máximo do exemplar (ou outra medida de referência), Tl₀= valor de referência, média ou mediana do comprimento total. O efeito da espécie/população sobre o tamanho médio de cada variável é posteriormente testado pela análise de variância.

Value:

Gráficos das regressões e box-plots de comparações serão gerados. Um data frame é gerado apresentando o resultado do teste-t para alometria e da análise de variância para avaliação de associação entre variável resposta e independente. Outro data frame exibe a tabela ANOVA de comparação entre pop/espécies. Dois arquivos .txt serão gerados no diretório de trabalho.

Warning:

Atente se os dados exibem distribuição normal, e se as variâncias das variáveis em comparação são homogêneas. Efetue os testes diagnósticos para explorar os dados.

O designe da ANOVA é para amostras balanceadas, logo NAs (que serão automaticamente removidos) podem tornar a análise inválida a subsequentes interpretações.

Note que não há dimensão nos box-plot, pois Z é adimensional

Author(s):

Riguel Feltrin Contente

riguel@io.usp.br

References:

Slope of a regression line t-test - <http://en.wikipedia.org/wiki/T-test>

LLEONART J.; SALAT J.; TORRES G.J. Removing Allometric Effects of Body Size in Morphological Analysis. Journal of Theoretical Biology, Volume 205, Number 1, 7 July 2000, pp.

85-93(9)

Examples:

```
data(iris) ## Abrir "iris"
setosa=iris[iris$Species=="setosa",] ## Selecionar apenas os vetores com as
variáveis morfométricas da sépala e pétala da espécie 'setosa'
setosa<-setosa[,-5]
head(setosa)
virginical=iris[iris$Species=="virginica",] ## Selecionar apenas os vetores
com as variáveis morfométricas da sépala e pétala da espécie 'virginica'
virginica<-virginica[,-5]
head(virginica)
setosa.b<-c(1,1,1,1) ## Criar um vetor numérico com os coeficientes
esperados de acordo com a dimensão da variável
virginica.b<-c(1,1,1,1)
nomes<-c("Setosa","Virginica") ## Criar objeto com o nome das espécies
RemAllom(setosa,virginica,setosa.b,virginica.b,nomes,diag.allom=F,TL0media=T
,diag.norm=F,WT=T) ## Rodar a função na qual a normalização será efetuada
pela média da sépala
```

Código da Função

```
RemAllom<-  
function(A,B,coef.A,coef.B,nomes,diag.allom=T,TL0media=T,diag.norm=T,WT=T)  
{  
##Matriz A  
##Remoção das linhas com NA, transformação ln e calculo da media e mediana  
  x.1<-na.omit(A)  
  x<-log(x.1)  
  MM<- matrix(rep(NA,dim(x)[2]*2),nrow=2)  
  for(j in 1:dim(x)[2])  
  {  
    MM[1,j]=mean(x[,j])  
    MM[2,j]=median(x[,j])  
  }  
##Remoção das linhas com NA e calculo da media e mediana  
  x.3<-na.omit(A)  
  MM2<- matrix(rep(NA,dim(x.3)[2]*2),nrow=2)  
  for(j in 1:dim(x.3)[2])  
  {  
    MM2[1,j]=mean(x.3[,j])  
    MM2[2,j]=median(x.3[,j])  
  }  
  mm=MM2  
  med=MM  
##Teste de hipótese (test-t) para crescimento alométrico  
  Stat<- matrix(rep(NA,(dim(x)[2]*8)-8),nrow=8)  
  for(j in 2:dim(x)[2])  
  {  
    som.q<-sum((x[,j]-med[1,j])^2)  
    mod.l<-lm(x[,j]~x[,1],data=x)  
    gl<-length(x[,1])-2  
    test.coef<-((coef(mod.l)[2]-  
coef.A[j])*sqrt(gl))/(sqrt(sum(residuals(mod.l)^2)/som.q))  
    p.1<-2*pt(test.coef,df=gl,lower.tail=F)  
    Stat[1,c(j-1)]=round(test.coef,3)  
    Stat[2,c(j-1)]=round(p.1,5)  
    Stat[3,c(j-1)]=round(coef(mod.l)[2],3)  
    Stat[4,c(j-1)]=coef.A[j]  
    Stat[6,c(j-1)]=round(summary(mod.l)$r.squared,3)  
    Stat[7,c(j-1)]=round(anova(mod.l)$"F value"[1],3)  
    Stat[8,c(j-1)]=round(anova(mod.l)$"Pr(>F)"[1],7)  
    StatA<-data.frame(Stat)  
  }  
  for(i in 1:dim(StatA)[2])  
  {  
    if(StatA[2,i]>=0.05)
```

```

        {StatA[5,i]=c("isom")}
      else{
        if(StatA[2,i]<0.05&StatA[3,i]>StatA[4,i])
          {StatA[5,i]=c("alom+")}
        else{
          if(StatA[2,i]<0.05&StatA[3,i]<StatA[4,i])
            {StatA[5,i]=c("alom-")}
        }}}
    }
  for(k in 1:dim(StatA)[2])
  {
    if(StatA[2,k]>=0.05)
      {StatA[2,k]=c(">0.05")}
    else{
      if(StatA[2,k]<0.05&StatA[2,k]>0.0001)
        {StatA[2,k]=StatA[2,k]}
      else{
        if(StatA[2,k]<0.0001)
          {StatA[2,k]=c("<0.0001")}
      }}}
    if(StatA[8,k]>=0.05)
      {StatA[8,k]=c(">0.05")}
    else{
      if(StatA[8,k]<0.05&StatA[8,k]>0.0001)
        {StatA[8,k]=StatA[8,k]}
      else{
        if(StatA[8,k]<0.0001)
          {StatA[8,k]=c("<0.0001")}
      }}}
  }
  StatB<- matrix(rep(NA,(dim(x)[2]*5)-5),nrow=5)
  for(j in 2:dim(x)[2])
  {
    som.q<-sum((x[,1]-med[1,j])^2)
    mod.l<-lm(x[,j]~x[,1],data=x)
    gl<-length(x[,1])-2
    test.coef<-((coef(mod.l)[2]-
coef.A[j])*sqrt(gl))/(sqrt(sum(residuals(mod.l)^2)/som.q))
    p.l<-2*pt(test.coef,df=gl,lower.tail=F)
    StatB[1,c(j-1)]=round(test.coef,3)
    StatB[2,c(j-1)]=round(p.l,5)
    StatB[3,c(j-1)]=coef(mod.l)[2]
    StatB[4,c(j-1)]=coef.A[j]
  }
  ##Gráficos diagnósticos para o Modelo Gaussiano
  if(diag.allom==T)
  {
    for(j in 2:dim(x)[2])
    {
      win.graph(width = 9, height = 9)
      par(mfrow=c(2,2))

```

```
        mlw<-lm(x[,j]~x[,1],data=x)
        plot(mlw,main=paste("ln",names(x[j])),nomes[1]))
    }
    par(mfrow=c(1,1))
}
##Regressões
g=1
for(i in 1:dim(x)[2])
{
    g=g+1
    if(g<=dim(x)[2]){
        x11()
        par(mfrow=c(2,2))
        par(tcl=0.3)
        par(mgp=c(1.8,0.5,0))
        mlw<-lm(x[,g]~x[,1],data=x)
        gra<-
plot(x[,g]~x[,1],data=x,main=nomes[1],ylab=paste("ln",names(x[g])),xlab=paste("ln",names(x[1])))
        mtext(StatA[5,g-1],side=3,cex=0.8,line=0.75,las=0,font=2,adj=0)
        abline(mlw,col=2)
        g=g+1
        if(g<=dim(x)[2]){
            mlw<-lm(x[,g]~x[,1],data=x)
            gra<-
plot(x[,g]~x[,1],data=x,main=nomes[1],ylab=paste("ln",names(x[g])),xlab=paste("ln",names(x[1])))
            mtext(StatA[5,g-1],side=3,cex=0.8,line=0.75,las=0,font=2,adj=0)
            abline(mlw,col=2)
            g = g+1
            if(g<=dim(x)[2]){
                mlw<-lm(x[,g]~x[,1],data=x)
                gra<-
plot(x[,g]~x[,1],data=x,main=nomes[1],ylab=paste("ln",names(x[g])),xlab=paste("ln",names(x[1])))
                mtext(StatA[5,g-1],side=3,cex=0.8,line=0.75,las=0,font=2,adj=0)
                abline(mlw,col=2)
                g = g+1
                if(g<=dim(x)[2]){
                    mlw<-lm(x[,g]~x[,1],data=x)
                    gra<-
plot(x[,g]~x[,1],data=x,main=nomes[1],ylab=paste("ln",names(x[g])),xlab=paste("ln",names(x[1])))
                    mtext(StatA[5,g-1],side=3,cex=0.8,line=0.75,las=0,font=2,adj=0)
                    abline(mlw,col=2)
                }
            }
        }
    }
}
##Dados normalizados pela média ou mediana
R.all<-
matrix(rep(NA,dim(x.3)[1]*dim(x.3)[2]),nrow=dim(x.3)[1],ncol=dim(x.3)[2])
```

```

for(l in 1:dim(x.3)[1])
{
  for(j in 2:dim(x.3)[2])
  {
    if(TL0media==T)
      {raz<-x.3[l,j]*((mm[1,1]/x.3[l,1])^StatB[3,j-1])
      R.all[l,j-1]<-raz}
    else{
      {raz<-x.3[l,j]*((mm[2,1]/x.3[l,1])^StatB[3,j-1])
      R.all[l,j-1]<-raz}}
  }
}
##Resultados - dados matriz A
nomes.0<-c(nomes[1],"","","","","","","")
nomes.1<-c("-->TESTE-T","p/ avaliar","alometria","","","-->ANOVA","p/
avaliar","sig. regressão")
nomes.2<-c("t","Pr(>|t|)","b obs","b exp","cresc","r2","F","Pr(>F)")
StatC<-data.frame(nomes.0,nomes.1,nomes.2,StatA)
nomes.componentes<-c("sp/pop.","testes","comp.")
nomes.col<-matrix(rep(NA,(dim(x)[2])-1),nrow=1)
for(j in 2:dim(x)[2])
{
  nomes.col[1,j-1]<-names(x[j])
}
MF<-R.all[,-c(dim(R.all))]
colnames(MF)<-nomes.col
names(StatC)<-c(nomes.componentes,nomes.col)

##Teste de normalidade- dados matriz A
if(diag.norm==T)
{
  x11()
  par(tcl=0.3)
  par(mfrow=c(2,2),pty="s")
  devAskNewPage(TRUE)
  for(m in 1:dim(MF)[2])
  {
    qqnorm(MF[,m],main=paste(nomes[1],nomes.col[1,m]))
    qqline(MF[,m],col=2)
    hist(MF[,m],prob=T,main="",xlab=nomes.col[1,m])
    par(new=T)
  }
  curve(expr=dnorm(x,mean(MF[,m]),sd(MF[,m])),main="",col=2,ann=F,axes=F)
  par(mfrow=c(1,1))
}
devAskNewPage(F)

##Matriz B
##Remoção das linhas com NA, transformação ln e calculo da media e mediana
y.1<-na.omit(B)
y<-log(y.1)
MMY<- matrix(rep(NA,dim(y)[2]*2),nrow=2)

```

```
for(j in 1:dim(y)[2])
{
  MMy[1,j]=mean(y[,j])
  MMy[2,j]=median(y[,j])
}
##Remoção das linhas com NA e calculo da media e mediana
y.3<-na.omit(B)
MM2y<- matrix(rep(NA,dim(y.3)[2]*2),nrow=2)
for(j in 1:dim(y.3)[2])
{
  MM2y[1,j]=mean(y.3[,j])
  MM2y[2,j]=median(y.3[,j])
}
##Teste de hipótese (test-t) para crescimento alométrico
mmy=MM2y
medy=MMy
Staty<- matrix(rep(NA,(dim(y)[2]*8)-8),nrow=8)
for(j in 2:dim(y)[2])
{
  som.qy<-sum((y[,j]-medy[1,j])^2)
  mod.ly<-lm(y[,j]~y[,1],data=y)
  gly<-length(y[,j])-2
  test.coefy<-((coef(mod.ly)[2]-
coef.B[j])*sqrt(gly))/(sqrt(sum(residuals(mod.ly)^2)/som.qy))
  p.ly<-2*pt(test.coefy,df=gly,lower.tail=F)
  Staty[1,c(j-1)]=round(test.coefy,3)
  Staty[2,c(j-1)]=round(p.ly,5)
  Staty[3,c(j-1)]=round(coef(mod.ly)[2],3)
  Staty[4,c(j-1)]=coef.B[j]
  Staty[6,c(j-1)]=round(summary(mod.ly)$r.squared,3)
  Staty[7,c(j-1)]=round(anova(mod.ly)$"F value"[1],3)
  Staty[8,c(j-1)]=round(anova(mod.ly)$"Pr(>F)"[1],7)
  StatAy<-data.frame(Staty)
}
for(i in 1:dim(StatAy)[2])
{
  if(StatAy[2,i]>=0.05)
    {StatAy[5,i]=c("isom")}
  else{
    if(StatAy[2,i]<0.05&StatAy[3,i]>StatAy[4,i])
      {StatAy[5,i]=c("alom+")}
    else{
      if(StatAy[2,i]<0.05&StatAy[3,i]<StatAy[4,i])
        {StatAy[5,i]=c("alom-")}
    }
  }
}
for(k in 1:dim(StatAy)[2])
{
  if(StatAy[2,k]>=0.05)
    {StatAy[2,k]=c(">0.05")}
```

```

    else{
      if(StatAy[2,k]<0.0499999&StatA[2,k]>0.0001)
        {StatAy[2,k]=StatAy[2,k]}
      else{
        if(StatAy[2,k]<0.0001)
          {StatAy[2,k]=c("<0.0001")}
      }}
    if(StatAy[8,k]>=0.05)
      {StatAy[8,k]=c(">0.05")}
    else{
      if(StatAy[8,k]<0.0499999&StatA[8,k]>0.0001)
        {StatAy[8,k]=StatAy[8,k]}
      else{
        if(StatAy[8,k]<0.0001)
          {StatAy[8,k]=c("<0.0001")}
      }}
  }}}
}
StatBy<- matrix(rep(NA,(dim(y)[2]*5)-5),nrow=5)
for(j in 2:dim(y)[2])
{
  som.qy<-sum((y[,1]-medy[1,j])^2)
  mod.ly<-lm(y[,j]~y[,1],data=y)
  gly<-length(y[,1])-2
  test.coefy<-((coef(mod.ly)[2]-
coef.B[j])*sqrt(gly))/(sqrt(sum(residuals(mod.ly)^2)/som.qy))
  p.ly<-2*pt(test.coefy,df=gly,lower.tail=F)
  StatBy[1,c(j-1)]=round(test.coefy,3)
  StatBy[2,c(j-1)]=round(p.ly,5)
  StatBy[3,c(j-1)]=coef(mod.ly)[2]
  StatBy[4,c(j-1)]=coef.B[j]
}
##Gráficos diagnósticos para o Modelo Gaussiano
if(diag.allom==T)
{
  for(j in 2:dim(y)[2])
  {
    win.graph(width=9,height=9)
    par(mfrow=c(2,2))
    mlwy<-lm(y[,j]~y[,1],data=y)
    plot(mlwy,main=paste("ln",names(x[j]),nomes[2]))
  }
  par(mfrow=c(1,1))
}
##Regressões
g=1
for(i in 1:dim(y)[2])
{
  g=g+1
  if(g<=dim(y)[2]){
    x11()
    par(mfrow=c(2,2))
  }
}

```

```
      par(tcl=0.3)
      par(mgp=c(1.8,0.5,0))
      mlwy<-lm(y[,g]~y[,1],data=y)
      grayy<-
plot(y[,g]~y[,1],data=y,main=nomes[2],ylab=paste("ln",names(y[g])),xlab=paste("ln",names(y[1])))
      mtext(StatAy[5,g-1],side=3,cex=0.8,line=0.75,las=0,font=2,adj=0)
      abline(mlwy,col=2)
      g=g+1
      if(g<=dim(y)[2]){
        mlwy<-lm(y[,g]~y[,1],data=y)
        grayy<-
plot(y[,g]~y[,1],data=y,main=nomes[2],ylab=paste("ln",names(y[g])),xlab=paste("ln",names(y[1])))
      mtext(StatAy[5,g-1],side=3,cex=0.8,line=0.75,las=0,font=2,adj=0)
      abline(mlwy,col=2)
      g = g+1
      if(g<=dim(y)[2]){
        mlwy<-lm(y[,g]~y[,1],data=y)
        grayy<-
plot(y[,g]~y[,1],data=y,main=nomes[2],ylab=paste("ln",names(y[g])),xlab=paste("ln",names(y[1])))
      mtext(StatAy[5,g-1],side=3,cex=0.8,line=0.75,las=0,font=2,adj=0)
      abline(mlwy,col=2)
      g = g+1
      if(g<=dim(y)[2]){
        mlwy<-lm(y[,g]~y[,1],data=y)
        grayy<-
plot(y[,g]~y[,1],data=y,main=nomes[2],ylab=paste("ln",names(y[g])),xlab=paste("ln",names(y[1])))
      mtext(StatAy[5,g-1],side=3,cex=0.8,line=0.75,las=0,font=2,adj=0)
      abline(mlwy,col=2)
    }}}}
  }
##Dados normalizados pela média ou mediana
  R.ally<-
matrix(rep(NA,dim(y.3)[1]*dim(y.3)[2]),nrow=dim(y.3)[1],ncol=dim(y.3)[2])
  for(l in 1:dim(y.3)[1])
  {
    for(j in 2:dim(y.3)[2])
    {
      if(TL0media==T)
        {razy<-y.3[l,j]*((mmy[1,1]/y.3[l,1])^StatBy[3,j-1])
        R.ally[l,j-1]<-razy}
      else{
        {razy<-y.3[l,j]*((mmy[2,1]/y.3[l,1])^StatBy[3,j-1])
        R.ally[l,j-1]<-razy}}
    }
  }
  nomes.0y<-c(nomes[2], "", "", "", "", "", "", "")
```

```

    nomes.1y<-c("-->TESTE-T","p/ avaliar","alometria","","","-->ANOVA","p/
avaliar","sig. regressão")
    nomes.2y<-c("t","Pr(>|t|)","b obs","b exp","cresc","r2","F","Pr(>F)")
    StatCy<-data.frame(nomes.0y,nomes.1y,nomes.2y,StatAy)
    nomes.componentesy<-c("sp/pop.","testes","comp.")
    nomes.coly<-matrix(rep(NA,(dim(y)[2])-1),nrow=1)
    for(j in 2:dim(y)[2])
    {
        nomes.coly[1,j-1]<-names(y[j])
    }
    MFy<-R.ally[, -c(dim(R.ally))]
    colnames(MFy)<-nomes.coly
    names(StatCy)<-c(nomes.componentesy,nomes.coly)

##Teste de normalidade- dados matriz B
    if(diag.norm==T)
        {x11()
        par(tcl=0.3)
        par(mfrow=c(2,2),pty="s")
        devAskNewPage(TRUE)
        for(m in 1:dim(MFy)[2])
        {
            qqnorm(MFy[,m],main=paste(nomes[2],nomes.coly[1,m]))
            qqline(MFy[,m],col=2)
            hist(MFy[,m],prob=T,main="",xlab=nomes.coly[1,m])
            par(new=T)
        }
        curve(expr=dnorm(x,mean(MFy[,m]),sd(MFy[,m])),main="",col=2,ann=F,axes=F)
        }
        par(mfrow=c(1,1))
        devAskNewPage(F)

##Geração dos dados normalizados e análises comparativas
##Geração do fator
MFA=matrix(rep(NA,(dim(MF)[2])*(dim(MF)[1]*2)),nrow=dim(MF)[1]*2,ncol=dim(MF)
)[2])
    for(i in 1:dim(MF)[1])
    {
        for(j in 1:dim(MF)[2])
        {
            MFA[i,j]=nomes[1]
        }
    }
    for(i in 1:dim(MFy)[1])
    {
        for(j in 1:dim(MFy)[2])
        {
            MFA[i+dim(MF)[1],j]=nomes[2]
            as.factor(MFA)
            MFB<-MFA[,1]
        }
    }
##Matriz p/ as análises
mtx=matrix(rep(NA,(dim(MF)[2])*(dim(MF)[1]*2)),nrow=dim(MF)[1]*2,ncol=dim(MF)

```

```

)[2])
  for(i in 1:dim(MF)[1])
  {
    for(j in 1:dim(MF)[2])
    {
      mtx[i,j]=MF[i,j]
    }
  }
  for(i in 1:dim(MFy)[1])
  {
    for(j in 1:dim(MFy)[2])
    {
      mtx[i+dim(MF)[1],j]=MFy[i,j]
    }
  }
##Exportação das matrizes com dados normalizados
if(WT==T)
  {write.table(MF,file="MFA.txt",row.names=T,col.names=T,sep=",")
  write.table(MFy,file="MFB.txt",row.names=T,col.names=T,sep=",")
  }
##Box-plots
g=0
for(i in 1:dim(mtx)[2])
{
  g=g+1
  if(g<=dim(mtx)[2]){
    x11()
    par(mfrow=c(2,2))
    par(tcl=0.3)
    par(mgp=c(1.8,0.5,0))
    boxplot(mtx[,g]~MFB,main=nomes.col[g],ylab=names(mtx[,g]))
    g=g+1
    if(g<=dim(mtx)[2]){
      boxplot(mtx[,g]~MFB,main=nomes.col[g],ylab=names(mtx[,g]))
      g = g+1
      if(g<=dim(mtx)[2]){
boxplot(mtx[,g]~MFB,main=nomes.col[g],ylab=names(mtx[,g]))
      g = g+1
      if(g<=dim(mtx)[2]){
boxplot(mtx[,g]~MFB,main=nomes.col[g],ylab=names(mtx[,g]))
      }}}}
  }

##ANOVA
##SS Total
media.geral=matrix(rep(NA,dim(MF)[2]),nrow=1)
for(i in 1:dim(MF)[2])
{
  media.geral[1,i]=mean(c(MF[,i],MFy[,i]))
}
mtx1=matrix(rep(NA,(dim(MF)[2])*(dim(MF)[1]*2)),nrow=dim(MF)[1]*2,ncol=dim(MF)[2])

```

```

for(i in 1:dim(MF)[1])
{
  for(j in 1:dim(MF)[2])
  {
    mtx1[i,j]=(MF[i,j]-media.geral[1,j])^2
  }
}
for(i in 1:dim(MFy)[1])
{
  for(j in 1:dim(MFy)[2])
  {
    mtx1[i+dim(MF)[1],j]=(MFy[i,j]-media.geral[1,j])^2
  }
}
ss.total=(apply(mtx1,2,sum))

##Somatória Quadráticas Intra grupo
mediaA.q<-
matrix(rep(NA,(dim(MF)[1]*dim(MF)[2])),nrow=dim(MF)[1],ncol=dim(MF)[2])
for(i in 1:dim(MF)[1])
{
  for(j in 1:dim(MF)[2])
  {
    mediaA.q[i,j]<-(MF[i,j]-mean(c(MF[,j])))^2
  }
}
ss.mediaA.q=apply(mediaA.q,2,sum)
mediaB.q<-
matrix(rep(NA,(dim(MFy)[1]*dim(MFy)[2])),nrow=dim(MFy)[1],ncol=dim(MFy)[2])
for(i in 1:dim(MFy)[1])
{
  for(j in 1:dim(MFy)[2])
  {
    mediaB.q[i,j]<-(MFy[i,j]-mean(c(MFy[,j])))^2
  }
}
ss.mediaB.q=apply(mediaB.q,2,sum)
ss.intra=ss.mediaA.q+ss.mediaB.q
##Somatória Quadráticas Entre grupos
mediaA<-apply(MF,2,mean)
mediaB<-apply(MFy,2,mean)
ss.entre=dim(MF)[1]*((mediaA-media.geral)^2+(mediaB-media.geral)^2)
##Cálculo do F
GL1=1
GL2=(dim(mtx)[1]-2)
ms.entre=ss.entre/GL1
ms.intra=ss.intra/GL2
F.anova=ms.entre/ms.intra
format.pval(c(0.1, 0.001, 1e-27))
##Cálculo do P
pf.anova=matrix(rep(NA,dim(MF)[2]*1),ncol=dim(MF)[2],nrow=1)
for(i in 1:dim(MF)[2])
{
  pf.anova[1,i]=pf(F.anova[1,i],GL1,GL2,lower.tail=FALSE)
}

```

```
##Tabela ANOVA
  PPato=round(c(pf.anova),3)
  SSE=round(c(ms.entre),3)
  MSI=round(c(ms.intra),3)
  FI=round(c(F.anova),3)
  SSI=round(c(ss.intra),3)
  GL1m=rep(1,dim(MF)[2])
  GL2m=rep(GL2,dim(MF)[2])
  GL1.r=round(GL1m,1)
  GL2.r=round(GL2m,1)
  Tabela.ANOVA<-
matrix(c(SSE,SSI,GL1.r,GL2.r,SSE,MSI,FI,PPato),ncol=dim(MF)[2],nrow=8,byrow=
T)
  colnames(Tabela.ANOVA)=nomes.col
  rownames(Tabela.ANOVA)=c("Sq.entre","Sq.res","gl1","gl2","Mq.entre","Mq.res"
,"F","Pr(>F)")
##Resultado
  cat("\t\nResultado teste-t - avaliação de desvios do crescimento
isométrico \nResultado ANOVA - teste de associação entre as variáveis
resposta e independente \nTabela ANOVA\n")
  return(StatC,StatCy,Tabela.ANOVA)
}
```

Arquivo da Função

[RemAllom.r](#)

From:

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

Permanent link:

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



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