ESTUDO PERFILHADA PARA ρ:

```
## theta1=(eta,nu1,nu2,phi)
monta.V.rho <-function(theta1, rho, dados.comp){
   y <- dados.comp[[2]][[1]]
   eta <- theta1[1]
   nu1 <- theta1[2]
   nu2 <- theta1[3]
   phi <- theta1[4]
   rho <- rho
   ## Calculando os elementos da matriz de correlação espacial
   distancia <- unname(as.matrix(dist(dados.comp[[3]],diag=TRUE,
                       upper=TRUE)))
   correla <- exp(-distancia/phi)
   ## Organizando a matriz de correlação espacial com eltos y1 e y2
   ## intercalados. Ou seja, calculando R1
   coluna1 <- rep(c(1, eta),length(y)/2)
   coluna2 <- rep(c(eta,eta^2),length(y)/2)
   seq1 <- seq(1,length(y),by=2)
   seq2 <- seq(2,length(y),by=2)
   ERRE1 <- matrix(ncol=length(y),nrow=length(y))
   ERRE1[,c(seq1)] <- coluna1
   ERRE1[,c(seq2)] <- coluna2
   ## Organizando a matriz de correlação espacial com eltos y1 e y2
   ## intercalados
   ERRE2 <- matrix(nrow=length(y),ncol=length(y))
   for(i in 1:length(y)/2){
     ERRE2[seq1[i],] <- rep(correla[i,],each=2)
     ERRE2[seq2[i],] <- rep(correla[i,],each=2)
   }
   ## Calculando a matriz R
   ERRE <- ERRE1*ERRE2
   ## Calculando a matriz de covariância composicional
   Ib1 <- matrix(c(nu1^2,nu1*nu2*rho,nu1*nu2*rho,nu2^2),nc=2)
   Ib <- kronecker(diag(length(y)/2),Ib1)
```

```
   ## Matriz de covariância espacial composicional
   V <- ERRE + Ib
   return(V)
}

## theta1=(eta,nu1,nu2,phi)
log.vero.rho <-function(theta1,rho,dados.comp,print.pars=F){
  if(theta1[1] <= 0)     return(.Machine$double.xmax^0.5)
  if(theta1[2] <= 0)     return(.Machine$double.xmax^0.5)
  if(theta1[3] <= 0)     return(.Machine$double.xmax^0.5)
  if(theta1[4] < 0)      return(.Machine$double.xmax^0.5)
  y <- dados.comp[[2]][[1]]
  X <- cbind(rep(1:0,length=length(y)),rep(0:1,length=length(y)))
  V <- monta.V.rho(theta1=theta1,rho=rho ,dados.comp = dados.comp)
  ldetV <- determinant(V,log=TRUE)$modulus[1]
  mu <- drop(solve(crossprod(X,solve(V,X))) %*%crossprod(X,solve(V,y)))
  Qe <- drop(crossprod(y,solve(V,y))-2*crossprod(y,solve(V,X%*%mu))+
             crossprod(mu,crossprod(X,solve(V,X%*%mu))))
  if(Qe < 0) return(Qe=.Machine$double.xmax^0.5)
  n <- length(y)
  s1 <- sqrt(Qe/n)
  ll <- drop(-0.5*(n*log(2*pi)+n*log(s1^2)+ ldetV + (1/(s1^2))*Qe))
  if(print.pars) print(c(theta1,ll))
  return(-ll)
}

## Obtendo valores iniciais para theta1 = (eta, nu1, nu2, phi)
perf.rho <- function(dados.comp,
                     tamanho,# Tamanho do vetor rho p/ perfilhar
                     min.delta, # Qto % menor que o mínimo dado pelo delta
                     max.delta, # Qto % maior que o máximo
                     alpha){
  var_y1 <- var(dados.comp[[1]][1])
  s1 <- var_y1/2
  tau1 <- s1
  var_y2 <- var(dados.comp[[1]][2])
```

```
s2 <- var_y2/2
tau2 <- s2
dim <- range(dist(dados[[3]]))
phi <- dim[1]+0.2*(dim[2]-dim[1])
eta <- s2/s1
nu1 <- tau1/s1
nu2 <- tau2/s1
theta1 <- c(eta,nu1,nu2,phi)
## Fazendo a maximização
source("monta.V.rho.R")
source("log.vero.rho.R")
estima <- mec(dados.comp=dados.comp)
## ic.inf.delta = limite inferior para rho pelo metodo delta
## ic.sup.delta = limite superior para rho pelo metodo delta
ic_i.delta.rho <- estima[[1]]$LI.Delta[8]
ic_s.delta.rho <- estima[[1]]$LS.Delta[8]
rho <- seq(ic_i.delta.rho*min.delta,ic_i.delta.rho*max.delta,l=tamanho)
rho[which(rho > 1*0.9995)] <- 1*0.9995
lista.res <- list()
for(i in 1:length(rho)){
  teste <- optim(theta1, log.vero.rho, rho=rho[i],
                 dados.comp=dados.comp, method="L-BFGS-B",
                 lower=c(1e-32,1e-32,1e-32,1e-32),
                 upper=c(20,20,20,200), hessian=TRUE)
   ## Todos os resultados para cada valor de rho
   lista.res[i][[1]] <- teste
}
perf.rho <- c()
for(i in 1:length(rho)){
  perf.rho[i] <- lista.res[[i]][2]$value
}
perf.rho
## Gráfico da perfilhada: rho versus log-verossimilhança
plot(rho,perf.rho,ylab="Log-verossimilhanca Perfilhada")
maximo <- -estima[[2]][2]
```

```
  deviance.rho <- 2*(perf.rho-maximo)
  plot(rho,deviance.rho,ylab='Deviance',type="l")
  abline(h=qchisq(0.95,df=1))
  val.rho.perf.dev <- data.frame(rho,perf.rho,deviance.rho)
  #intervalo.rho <- range(val.rho.perf.dev[which(val.rho.perf.dev$deviance<
                          qchisq(0.95,df=1)),]$rho)
  intervalo.rho <- val.rho.perf.dev[which(val.rho.perf.dev$deviance.rho<
                                          qchisq(alpha,df=1)),]$rho
  retorna <- list()
  retorna[[1]] <- val.rho.perf.dev
  retorna[[2]] <- intervalo.rho
  return(retorna)
}
```

Script para obtenção da perfilhada para rho

```
## Carregando o arquivo de dados do geoComp
source('pivo.R')
##data(pivo)
pivo
## Selecionando os componentes e as coordenadas
dados <- pivo[,c(6,7,8,1,2)]
## Convertendo o arquivo para o formato geoComp
## Transformacao ALR ( denominador=argila )
## dados[[1]]= Y1 Y2 ; dados[[2]]= Y ; dados[[3]]= Coord.X Coord.Y
source('as.geoComp.R')
dados <- as.geoComp(dados)
source('monta.V.R')
source('log.vero.R')
source('mec.R')
dados.comp <- dados
estima <- mec(dados)
## Fazendo a perfilhada para rho com alpha=095
source('perf.rho.R')
perfilhada.rho <- perf.rho(dados,tamanho=150,min.delta=0.85,max.delta=1.10,
```

```
                  alpha=0.95)
perfilhada.rho.1 <- perfilhada.rho[[1]]
write.table(perfilhada.rho.1,'dataframePerfrho095p150.txt')
perfilhada.rho.2 <- perfilhada.rho[[2]]
write.table(perfilhada.rho.2 ,'IC.Perf.rho.095p150.txt')
## Lendo os resultados
dad.per.rho.095 <- read.table('dataframePerfrho095p150.txt')
dad.ic.rho.095 <- read.table('IC.Perf.rho.095p150.txt')
## Organizando os intervalos
dad.ic.rho.095 <- data.frame(dad.ic.rho.095[1,],dad.ic.rho.095[90,])
names(dad.ic.rho.095) <- c('Li.rho.095','Ls.rho.095')
## Gráficos das perfilhadas: rho versus log-verossimilhanca
par(mfrow=c(1,1),mar=c(3,3,1.5,1.5),mgp=c(1.8,0.8,0))
plot(dad.per.rho.095$rho,-dad.per.rho.095$perf.rho,xlab=expression(rho),
     ylab="Log-verossimilhanca Perfilhada",type='l')
## Gráficos das deviances: rho versus Deviance
par(mfrow=c(1,1),mar=c(3,3,1.5,1.5),mgp=c(1.8,0.8,0))
plot(dad.per.rho.095$rho,dad.per.rho.095$deviance.rho,xlab=expression(rho),
     ylab='Deviance',type="l")
segments(x0=dad.ic.rho.095$Li.rho.095,x1=dad.ic.rho.095$Ls.rho.095,
         y0=qchisq(0.95,df=1),y1=qchisq(0.95,df=1))
arrows(x0=dad.ic.rho.095$Li.rho.095,y0=qchisq(0.95,df=1),
       x1=dad.ic.rho.095$Li.rho.095,y1=0)
text(x=dad.ic.rho.095$Li.rho.095,y=-0.2,
     label=as.character(round(dad.ic.rho.095$Li.rho.095,2)),cex=0.8)
```