

**ESTUDO PERFILHADA PARA  $\phi$ :**

```

## theta1=(eta,nu1,nu2,rho)
monta.V.phi <-function(theta1, phi, dados.comp){
  y <- dados.comp[[2]][[1]]
  eta <- theta1[1]
  nu1 <- theta1[2]
  nu2 <- theta1[3]
  rho <- theta1[4]
  phi <- phi
  ## 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,rho)
log.vero.phi <-function(theta1,phi,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$double.xmax^0.5)
  if(abs(theta1[4]) > 1) return(.Machine$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.phi(theta1=theta1,phi=phi ,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, rho)
perf.phi <- function(dados.comp,tamanho, # Tamanho vetor phi pra 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
eta <- s2/s1
nu1 <- tau1/s1
nu2 <- tau2/s1
rho <- cor(dados[[1]][1],dados[[1]][2])
theta1 <- c(eta,nu1,nu2,rho)
## Fazendo a maximização
source("monta.V.phi.R")
source("log.vero.phi.R")
estima <- mec(dados.comp=dados.comp)
## ic_i.delta.phi = limite inferior para phi pelo método delta
## ic_s.delta.phi = limite superior para phi pelo método delta
ic_i.delta.phi <- estima[[1]]$LI.Delta[7]
ic_s.delta.phi <- estima[[1]]$LS.Delta[7]
phi <- seq(ic_i.delta.phi*min.delta,ic_s.delta.phi*max.delta,l=tamanho)
lista.res <- list()
for(i in 1:length(phi)){
  teste <- optim(theta1, log.vero.phi, phi=phi[i],
                dados.comp=dados.comp, method="L-BFGS-B",
                lower=c(1e-32,1e-32,1e-32,-1),
                upper=c(Inf,Inf,Inf,1), hessian=TRUE)
  ## Todos os resultados para cada valor de phi
  lista.res[i][[1]] <- teste
}
perf.phi <- c()
for(i in 1:length(phi)){
  perf.phi[i] <- lista.res[[i]][2]$value
}
## Gráfico da perfilhada: phi versus log-verossimilhança
plot(phi,perf.phi,ylab='Log-verossimilhanca Perfilhada')
maximo <- -estima[[2]][2]
deviance.phi <- 2*(perf.phi-maximo)
plot(phi,deviance.phi,ylab='Deviance',type="l")
abline(h=qchisq(0.835,df=1))
val.phi.perf.dev <- data.frame(phi,perf.phi,deviance.phi)

```

```

intervalo.phi <- val.phi.perf.dev[which(val.phi.perf.dev$deviance.phi<
                                     qchisq(alpha,df=1)),]$phi
retorna <- list()
retorna[[1]] <- val.phi.perf.dev
retorna[[2]] <- intervalo.phi
return(retorna)
}

```

### Script para obtenção da perfilhada para phi

```

## 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 )
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 phi com alpha=0.566
source('perf.phi.R')
perfilhada.phi <-perf.phi(dados,tamanho=150,min.delta=0.8,max.delta=5,
                        alpha=0.566)
## Arquivo contendo valores de phi, perf.phi e deviance.phi
perfilhada.phi.1 <- perfilhada.phi[[1]]
write.table(perfilhada.phi.1,'dataframePerfphi0566p150.txt')
## Arquivo contendo todos os valores para os quais
## deviance<qchisq(alpha,df=1))

```

```
perfilhada.phi.2 <- perfilhada.phi[[2]]
write.table(perfilhada.phi.2 , 'IC.Perf.phi.0566p150.txt')

## Fazendo a perfilhada para phi com alpha=0.835
perfilhada.phi.3 <-perf.phi(dados,tamanho=150,min.delta=0.8,max.delta=5,
                           alpha=0.835)
## Arquivo contendo valores de phi, perf.phi e deviance.phi
perfilhada.phi.4 <- perfilhada.phi.3[[1]]
write.table(perfilhada.phi.4,'dataframePerfphi0835p150.txt')
## Arquivo contendo todos os valores para os quais
## deviance<qchisq(alpha,df=1))
perfilhada.phi.5 <- perfilhada.phi.3[[2]]
write.table(perfilhada.phi.5 , 'IC.Perf.phi.0835p150.txt')

## Lendo os resultados
dad.per.phi.0566 <- read.table('dataframePerfphi0566p150.txt')

dad.ic.phi.0566 <- read.table('IC.Perf.phi.0566p150.txt')
dad.ic.phi.0835 <- read.table('IC.Perf.phi.0835p150.txt')

## Organizando os intervalos
dad.ic.phi.0566 <- data.frame(dad.ic.phi.0566[1,],dad.ic.phi.0566[41,])
names(dad.ic.phi.0566) <- c('Li.phi.0566', 'Ls.phi.0566')
dad.ic.phi.0835 <- data.frame(dad.ic.phi.0835[1,],dad.ic.phi.0835[141,])
names(dad.ic.phi.0835) <- c('Li.phi.0835', 'Ls.phi.0835')

## Gráficos das perfilhadas: phi 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.phi.0566$phi,-dad.per.phi.0566$perf.phi,xlab=expression(phi),
      ylab="Log-verossimilhanca Perfilhada",type='l')

plot(dad.per.phi.0835$phi,-dad.per.phi.0835$perf.phi,xlab=expression(phi),
      ylab="Log-verossimilhanca Perfilhada",type='l')

## Gráficos das Deviances: phi 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.phi.0566$phi,dad.per.phi.0566$deviance.phi,xlab=expression(phi),
      ylab='Deviance',type="l")
segments(x0=dad.ic.phi.0566$Li.phi.0566,x1=dad.ic.phi.0566$Ls.phi.0566,
          y0=qchisq(0.566,df=1),y1=qchisq(0.566,df=1))
arrows(x0=dad.ic.phi.0566$Li.phi.0566,y0=qchisq(0.566,df=1),
        x1=dad.ic.phi.0566$Li.phi.0566,y1=0)
arrows(x0=dad.ic.phi.0566$Ls.phi.0566,y0=qchisq(0.566,df=1),
        x1=dad.ic.phi.0566$Ls.phi.0566,y1=0)
text(x=70,y=-0.1,label=as.character(round(dad.ic.phi.0566$Li.phi.0566,2)),
      cex=0.8)
text(x=dad.ic.phi.0566$Ls.phi.0566,y=-0.1,
      label=as.character(round(dad.ic.phi.0566$Ls.phi.0566,2)),cex=0.8)
segments(x0=dad.ic.phi.0835$Li.phi.0835,x1=dad.ic.phi.0835$Ls.phi.0835,
          y0=qchisq(0.835,df=1),y1=qchisq(0.835,df=1))
arrows(x0=dad.ic.phi.0835$Li.phi.0835,y0=qchisq(0.835,df=1),
        x1=dad.ic.phi.0835$Li.phi.0835,y1=0)
arrows(x0=dad.ic.phi.0835$Ls.phi.0835,y0=qchisq(0.835,df=1),
        x1=dad.ic.phi.0835$Ls.phi.0835,y1=0)
text(x=23,y=-0.1,label=as.character(round(dad.ic.phi.0835$Li.phi.0835,1)),
      cex=0.8)
text(x=dad.ic.phi.0835$Ls.phi.0835,y=-0.1,
      label=as.character(round(dad.ic.phi.0835$Ls.phi.0835,2)),cex=0.8)

```