8 - Dados de contagens (Mapeamento de doenças - Disease Mapping).

Prof. Vinícius D. Mayrink
EST171 - Estatística Espacial

Sala: 4078 Email: vdm@est.ufmg.br

1^o semestre de 2024

A estatística espacial tem sido amplamente aplicada em epidemiologia para estudar a distribuição de casos de doenças.

Deseja-se identificar áreas onde uma doença é prevalente, o que pode levar à detecção de fatores de risco previamente desconhecidos.

Este tipo de estudo também é chamado de Epidemiologia Espacial.

O conjunto de dados SIDS (Sudden Infant Deaths, Carolina do Norte, EUA) será usado nos exemplos desta parte do curso. A variável resposta é o n^o de mortes infantis nos condados do estado (períodos 1974-1978 e 1979-1984) e outras informações adicionais.

Dados disponíveis em: `nc.sids` (pacote `spdep`).

Objetivo do mapeamento de doenças: representar a distribuição espacial do risco de uma doença na região de estudo (dividida em várias sub-regiões).

O risco pode refletir n^o de mortes devido à doença (mortalidade) ou, se não for fatal, n^o de pessoas infectadas (morbidade) em certo período do tempo para a população sob risco.

Elementos básicos dos dados:
tamanho populacional sob risco e n^o de casos em cada área.

Os dados são geralmente separados em uma quantidade de grupos ou estratos de acordo com diferentes variáveis (exemplo: sexo, idade, etc).

Dependendo do tipo de estudo, a população sob risco ($n_i = n^o$ de expostos) pode ser um subconjunto da população total. Nos dados SIDS, ela seria o n^o de crianças nascidas durante o período do estudo.

Denote:
P_{ij} = população da região i e estrato j,
$O_{ij} = n^o$ observado de casos na região i e estrato j.

$$P_i = \sum_j P_{ij} \qquad O_i = \sum_j O_{ij} \qquad P_+ = \sum_i \sum_j P_{ij} \qquad O_+ = \sum_i \sum_j O_{ij}$$

O n^o observado de casos sozinho não informa sobre o risco da doença, visto que a distribuição dos casos estará muito ligada à distribuição da população.

Para obter uma estimativa do risco, o n^o observado de casos deve ser comparado ao n^o esperado de casos.

Se P_i e O_i estão disponíveis, situação mais simples, o n^o esperado de casos na região i pode ser calculado como $E_i = P_i r_+$, sendo $r_+ = O_+/P_+$ a taxa global de incidência. Este é um exemplo do uso da *padronização indireta*.

Para dados agrupados em estratos, um procedimento similar pode ser usado. Em vez de calcular a taxa global $r_+ = O_+/P_+$ para todas as regiões, utilize uma taxa para cada estrato dada por

$$r_j = \frac{\sum_i O_{ij}}{\sum_i P_{ij}}.$$

Ou seja, calcule a razão entre a soma de todos os casos no estrato j e a população do estrato j.

Nesta situação, o n^o esperado de casos na região i é dado por $E_i = \sum_j P_{ij} r_j$.

Esta padronização também é chamada de *padronização interna*.

Comandos para ler os dados SIDS, as fronteiras da Carolina do Norte e a estrutura de adjacência dos municípios:

```
library(maptools)
library(spdep)

nc_file <- system.file("shapes/sids.shp",package = "maptools")[1]
llCRS <- CRS("+proj=longlat +datum=NAD27")
nc <- readShapePoly(nc_file, ID = "FIPSNO", proj4string = llCRS)
rn <- sapply(slot(nc,"polygons"), function(x) slot(x,"ID"))

gal_file <- system.file("etc/weights/ncCR85.gal", package = "spdep")[1]
ncCR85 <- read.gal(gal_file, region.id = rn)
```

Usando o argumento `region.id` nos certificamos de que a ordem da lista de vizinhos em `ncCR85` é a mesma das áreas definidas no objeto `nc` que é do tipo `SpatialPolygonDataFrame`.

Uma suposição comum para modelar o n^o observado de casos na região i e estrato j é que a contagem segue a Poisson com média $\theta_i E_{ij}$, sendo θ_i o risco relativo.

Se $\theta_i = 1$, então o risco é a média da região de referência (de onde r_j's são obtidos). Há interesse em localizar regiões nas quais o risco relativo é significantemente maior que 1.

Assuma que não há interações entre o risco e os estratos da população, isto é, o risco relativo θ_i depende apenas da região.

$SMR_i = O_i/E_i$ é um estimador básico do risco em uma região (denominado Taxa de Mortalidade Padronizada).

Sendo assim, os dados envolvendo os casos são frequentemente referidos como "numerador". Os dados populacionais são ditos "denominador".

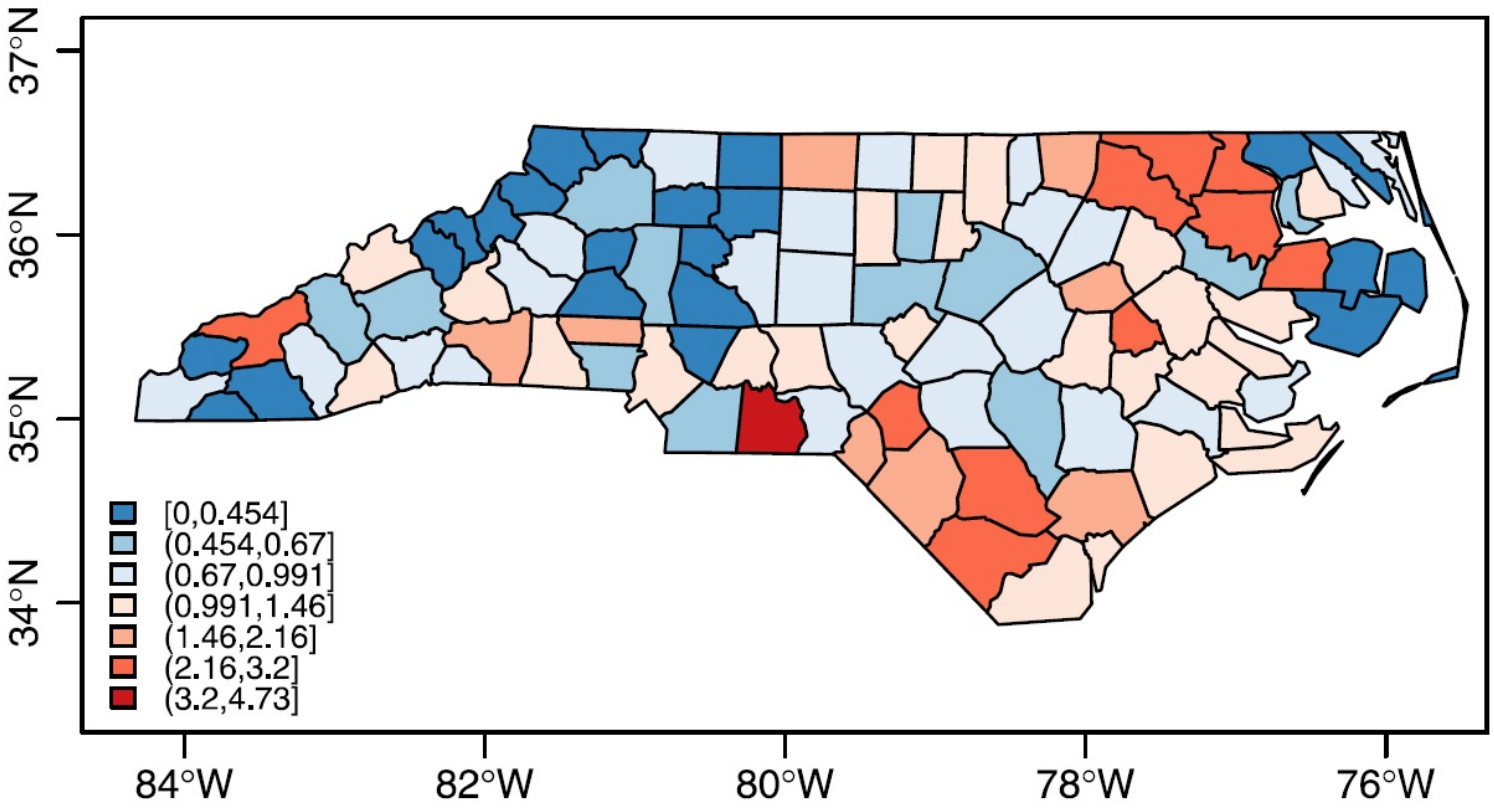


Fig.1 : Taxas de Mortalidade Padronizada (SMRs) para os dados SIDS (1974-1978).

A população sob risco é dada pelo n^o de nascimentos. Partindo disso, calcule as Taxas de Mortalidade Padronizadas (SMR_i).

```
observed <- nc.sids$SID74
population <- nc.sids$BIR74

r <- sum(observed)/sum(population)

expected <- population * r

SMR <- observed/expected
```

Visto que $O_i \sim$ Poisson, podemos obter um intervalo de confiança para cada SMR_i. Utilize a função `pois.exact` (pacote `epitools`).

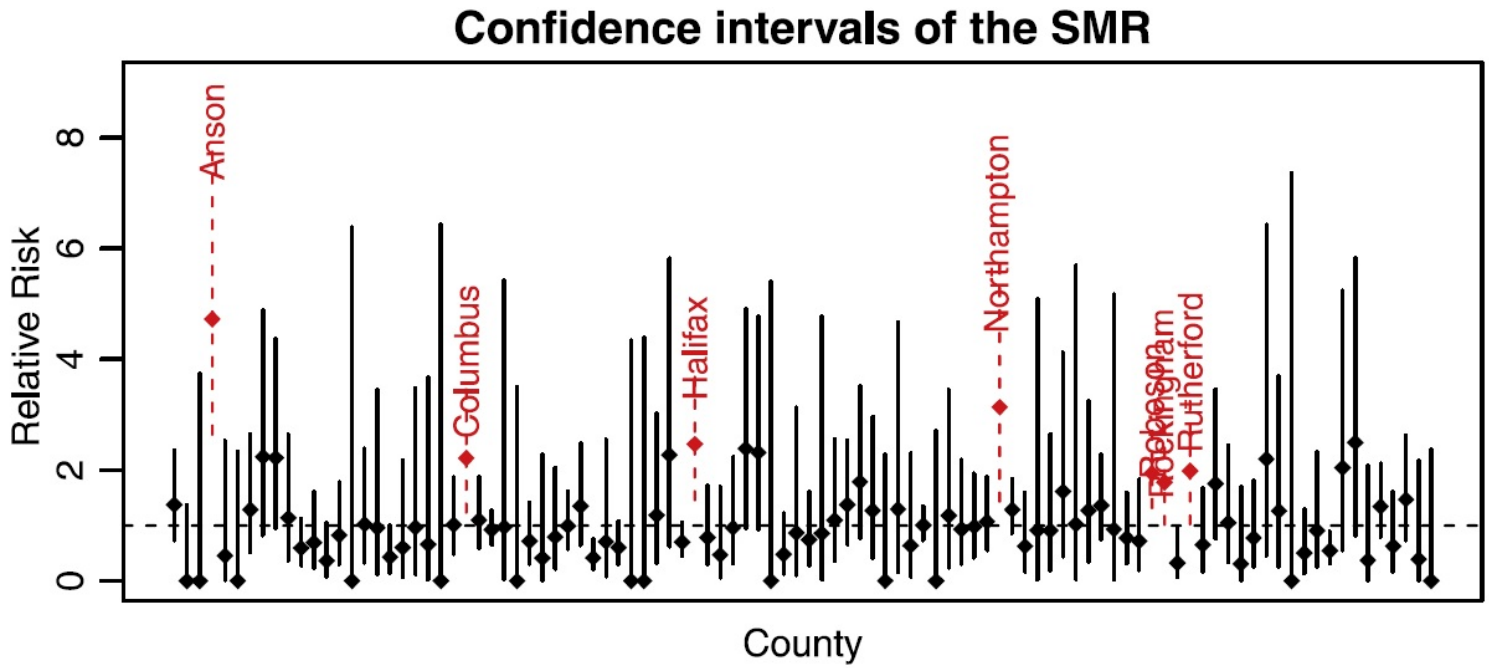


Fig.2 : Intervalos de confiança (95%) para o SMR obtido com uma fórmula exata. Pontos representam o SMR de cada área. Intervalos vermelhos tracejados são significativamente maiores que 1 (risco alto).

Modelo Poisson-Gama

A distribuição Poisson implica em suposições nem sempre válidas.

$O_i \sim$ Poisson $\Rightarrow$ média e a variância de O_i são iguais.
Dados podem ter sobredispersão, significando "variância maior que a média".
Neste caso, o modelo estatístico precisa ser melhorado.

Para permitir "variância $>$ média", troque a Poisson pela Binomial Negativa.

Seja $X = n^o$ de sucessos registrados até que $r > 0$ falhas ocorram.

Considere $p =$ probabilidade de sucesso.

Temos que $X \sim BNegativa(r, p)$ com:

$$P(X = x) = \binom{x + r - 1}{x} p^x (1 - p)^r \quad \text{para} \quad x = 0, 1, 2, \cdots$$

Então: $E(X) = rp/(1 - p)$ e $Var(X) = rp/(1 - p)^2$.

A Binomial Negativa pode ser vista como um modelo de mistura no qual um efeito aleatório com distribuição Gama é atribuído para cada região. Esta opção é conhecida como modelo Poisson-Gama, sendo estruturado em 2 níveis.

$$\begin{aligned}(O_i|\theta_i, E_i) &\sim Pois(\theta_i E_i),\\ \theta_i &\sim Ga(\nu, \alpha).\end{aligned}$$

Aqui, temos também o risco relativo θ_i sendo uma variável com distribuição Gama (média $= \nu/\alpha$ e variância $= \nu/\alpha^2$).

Veja que a distribuição de O_i está condicionada ao valor de θ_i.

A distribuição de cada O_i (sem condicionar em θ_i) é fácil de obter e segue uma Binomial Negativa parametrizada com ν e a probabilidade $\alpha/(\alpha + E_i)$.

A distribuição *a posteriori* de θ_i (isto é, θ_i dado as observações $\{O_i\}_{i=1}^n$) será a $Ga(\nu + O_i, \alpha + E_i)$. A esperança *a posteriori* será:

$$E[\theta_i|O_i, E_i] = \frac{\nu + O_i}{\alpha + E_i}.$$

Este resultado pode ser expresso como uma combinação linear entre a média *a priori* dos riscos relativos e a taxa SMR_i. Teremos o seguinte estimador:

$$E[\theta_i|O_i, E_i] = \frac{E_i}{\alpha + E_i} SMR_i + \left(1 - \frac{E_i}{\alpha + E_i}\right) \frac{\nu}{\alpha}.$$

Note que: $SMR_i = O_i/E_i$ e $\frac{E_i}{\alpha+E_i} \frac{O_i}{E_i} = \frac{O_i}{\alpha+E_i}$.

$$\frac{\nu}{\alpha} - \frac{\nu E_i}{\alpha(\alpha+E_i)} = \frac{\nu(\alpha+E_i)-\nu E_i}{\alpha(\alpha+E_i)} = \frac{\alpha\nu}{\alpha(\alpha+E_i)} = \frac{\nu}{(\alpha+E_i)}$$

No estimador (slide anterior), quando E_i é pequeno (comum para áreas com população pequena) uma variação pequena em O_i pode produzir mudanças drásticas no SMR_i. Assim, o SMR_i terá um peso baixo se comparado ao peso da média *a priori*.

Neste estimador, informações de todas as áreas são usadas para construir as estimativas *a posteriori*, visto que ν e α são os mesmos para todas as regiões.

ν e α são desconhecidos e devem ser estimados. Isto pode ser feito via estimativa Bayesiana Empírica (EB - Empirical Bayes), disponível no pacote `DCluster`.

```
library(DCluster)
eb <- empbaysmooth(observed, expected)
EBPG <- eb$smthrr
eb$nu
eb$alpha
```

O resultado é uma média *a priori* do risco relativo igual a $\nu/\alpha = 4.6307/4.3956 = 1.0535$ (bastante próxima de 1).

Mapas de probabilidades são uma forma de representar a significância das observações. Eles mostram a probabilidade de um valor ser mais alto que o observado conforme as suposições do modelo.

O próximo slide exibe 2 mapas de probabilidade (modelos Poisson e Poisson-Gama). O Poisson-Gama parece mais apropriado visto que há sobredispersão nos dados. Inferência deve ser baseada neste modelo.

Conforme esperado, as probabilidades de observações grandes no Poisson-Gama são mais altas (vermelho mais claro) visto que ele considera o contexto de maior variabilidade ou sobredispersão (*overdispersion*).

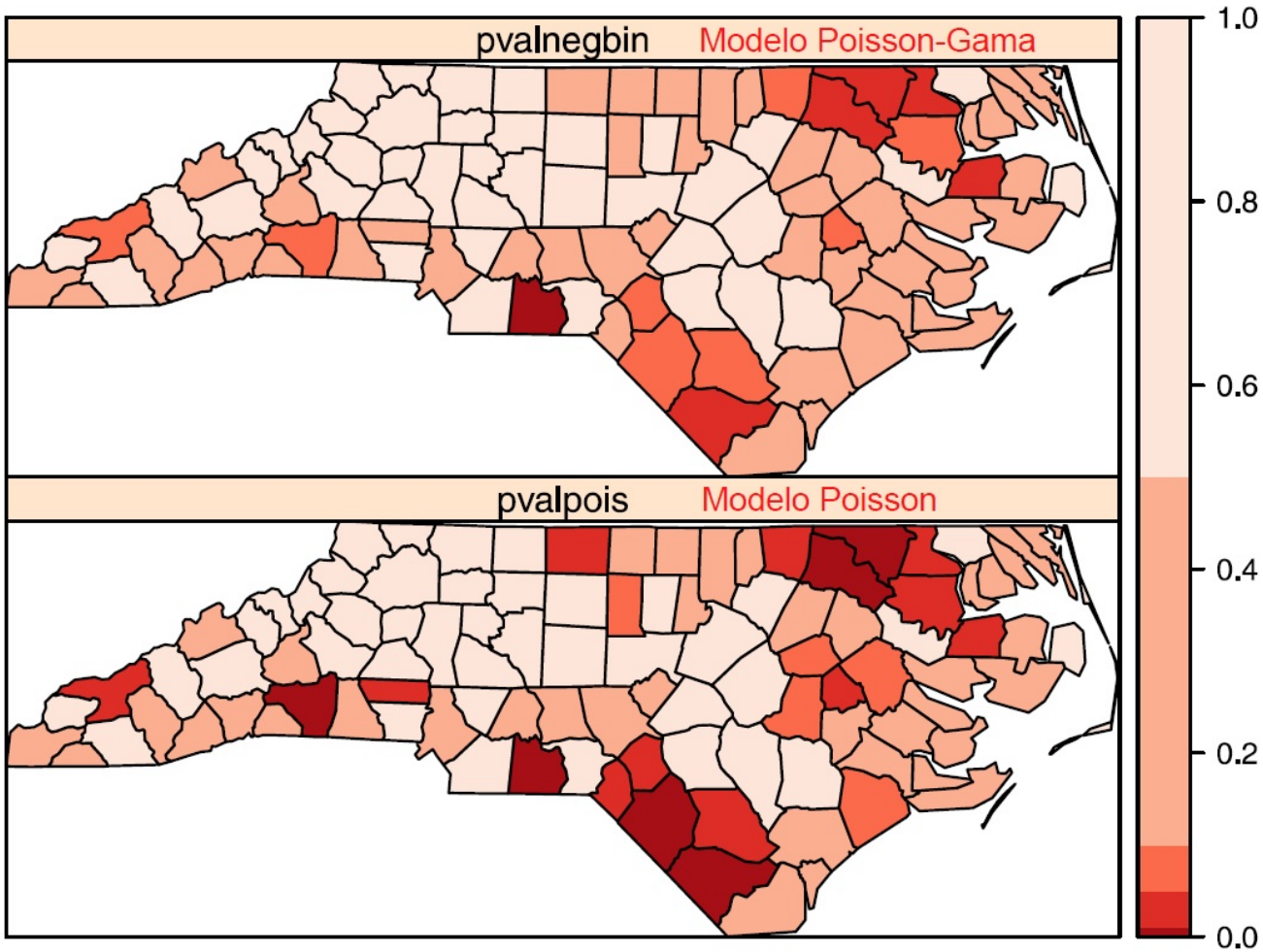


Fig.3 : Mapas de probabilidades para os modelos Poisson e Binomial Negativo (Poisson-Gama).

Modelo Log-Normal

Outro estimador do risco tomando como base o log do risco relativo é $\beta_i = \log(\theta_i)$, o qual segue uma Normal com média ϕ e variância σ^2 comuns a todo i.

$$\beta = (\beta_1, \beta_2, \cdots, \beta_n)' \sim N_n[\phi \mathbf{1}_n, \sigma^2 I_n].$$

$(O_i|\theta_i, E_i) \sim Pois(\theta_i E_i)$ então $\beta_i = \log(\theta_i) \sim N[\phi, \sigma^2]$ (i.e. $\theta_i \sim LogNormal$).

Neste caso, a estimativa do log-risco relativo não é obtida por $\log(O_i/E_i)$, mas sim por $\log[(O_i + 0.5)/E_i]$. Veja que a 1^a opção não está definida para $O_i = 0$.

Algorítmo EM é usado para estimar ϕ e σ^2. Estas estimativas (ver próximo slide) são inseridas no estimador empírico Bayesiano (EB) de β_i:

$$\hat{\beta}_i = b_i = \frac{\hat{\phi} + (O_i + 0.5)\hat{\sigma}^2 \log[(O_i + 0.5)/E_i] - \hat{\sigma}^2/2}{1 + (O_i + 0.5)\,\hat{\sigma}^2}$$

$$\hat{\phi} = \frac{1}{n}\sum_{i=1}^{n} b_i = \bar{b} \quad \text{e} \quad \hat{\sigma}^2 = \frac{1}{n}\left\{\hat{\sigma}^2 \sum_{i=1}^{n}[1 + \hat{\sigma}^2(O_i + 0.5)]^{-1} + \sum_{i=1}^{n}(b_i - \hat{\phi})^2\right\}$$

O b_i é atualizado sucessivamente (pelas expressões acima) até uma convergência.

Estimador de θ_i: $\hat{\theta}_i = \exp\{\hat{\beta}_i\}$.

Note que agora, informação é tomada emprestada ao estimarmos os parâmetros comuns ϕ e σ^2. As estimativas resultantes são uma combinação da estimativa local de ϕ e do log-risco relativo β_i.

Este estimador não pode ser escrito no formato de mistura.

```
ebln <- lognormalEB(observed, expected)
EBLN <- exp(ebln$smthrr)
```

Assumindo risco relativo θ_i, com média μ e variância σ^2 comuns *a priori* (sem especificar qualquer distribuição), através do Método de Momentos é possível obter o novo estimador:

$$\hat{\theta}_i = \hat{\mu} + C_i(SMR_i - \hat{\mu}) = (1 - C_i)\hat{\mu} + C_i\, SMR_i,$$

sendo $\hat{\mu} = \frac{\sum_{i=1}^{n} O_i}{\sum_{i=1}^{n} E_i}$ e $C_i = \frac{s^2 - \hat{\mu}/\bar{E}}{s^2 - (\hat{\mu}/\bar{E}) + (\hat{\mu}/E_i)}$.

Nestas formulações $\bar{E}$ = média dos E_i's e s^2 = estimador não viciado usual da variância dos SMR_i's.

Este estimador pode gerar estimativas negativas do risco relativo quando $s^2 < \hat{\mu}/\bar{E}$, neste caso faça $\hat{\theta}_i = \hat{\mu}$.

Este estimador do tipo mistura depende fortemente (também) do valor E_i.

- Se E_i é grande (significando SMR_i é um estimador confiável), C_i é ≈ 1 e o SMR_i tem mais peso na estimação.
- Se E_i é pequeno (SMR_i é pouco confiável e pega emprestado muita informação de outras áreas), a média *a priori* $\hat{\mu}$ tem mais peso na estimação.

```
library(spdep)
EBMarshall <- EBest(observed, expected)
EBMarshall <- EBMarshall[,2]
```

Na figura a seguir, todos os estimadores EB parecem produzir estimativas similares em todas as áreas. Comparando estes mapas com o de SMR, veja como valores extremos (baixos e altos) foram movidos em direção à média global. Ou seja, os valores foram suavizados ao considerar a informação global na estimação.

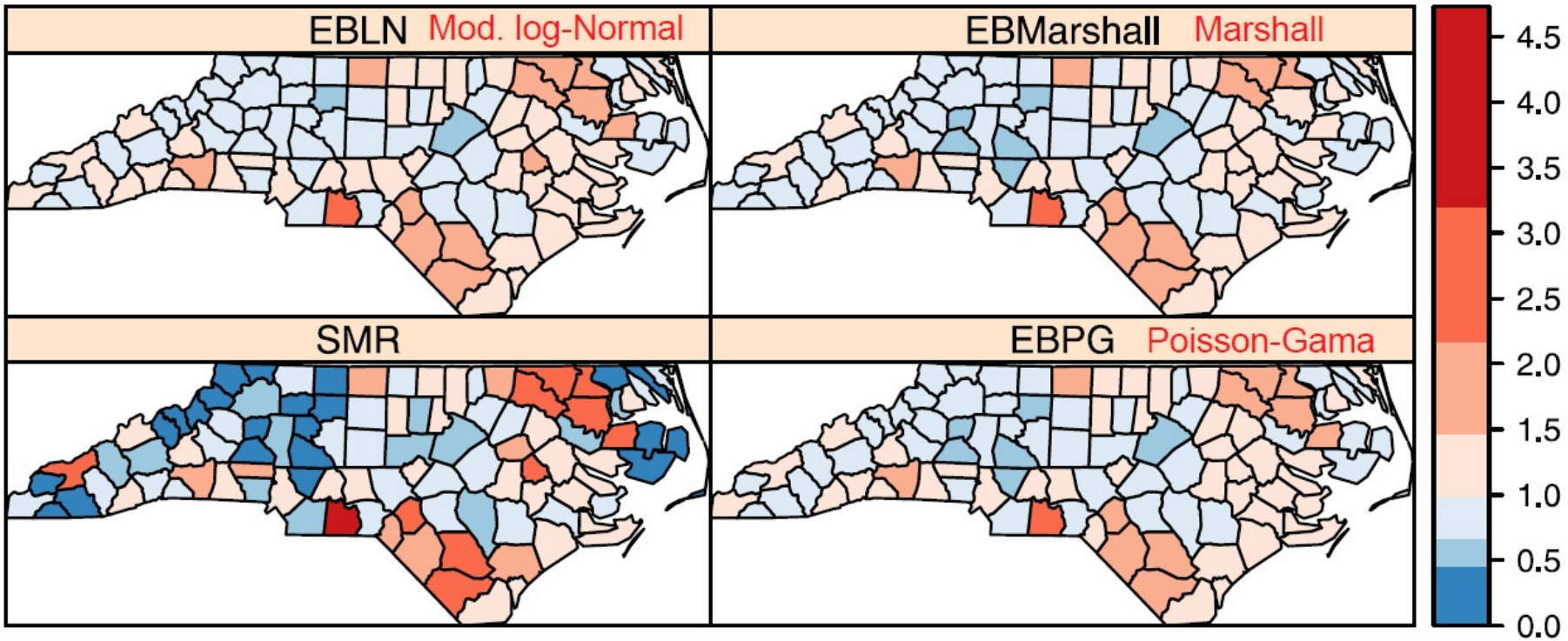


Fig.4 : Comparação de diferentes estimadores (EB) do risco: EBLN = Modelo log-normal, EBMarshal = Marshall e EBPG = Modelo Poisson-Gama. Gráfico SMR mostra a taxa de mortalidade padronizada.

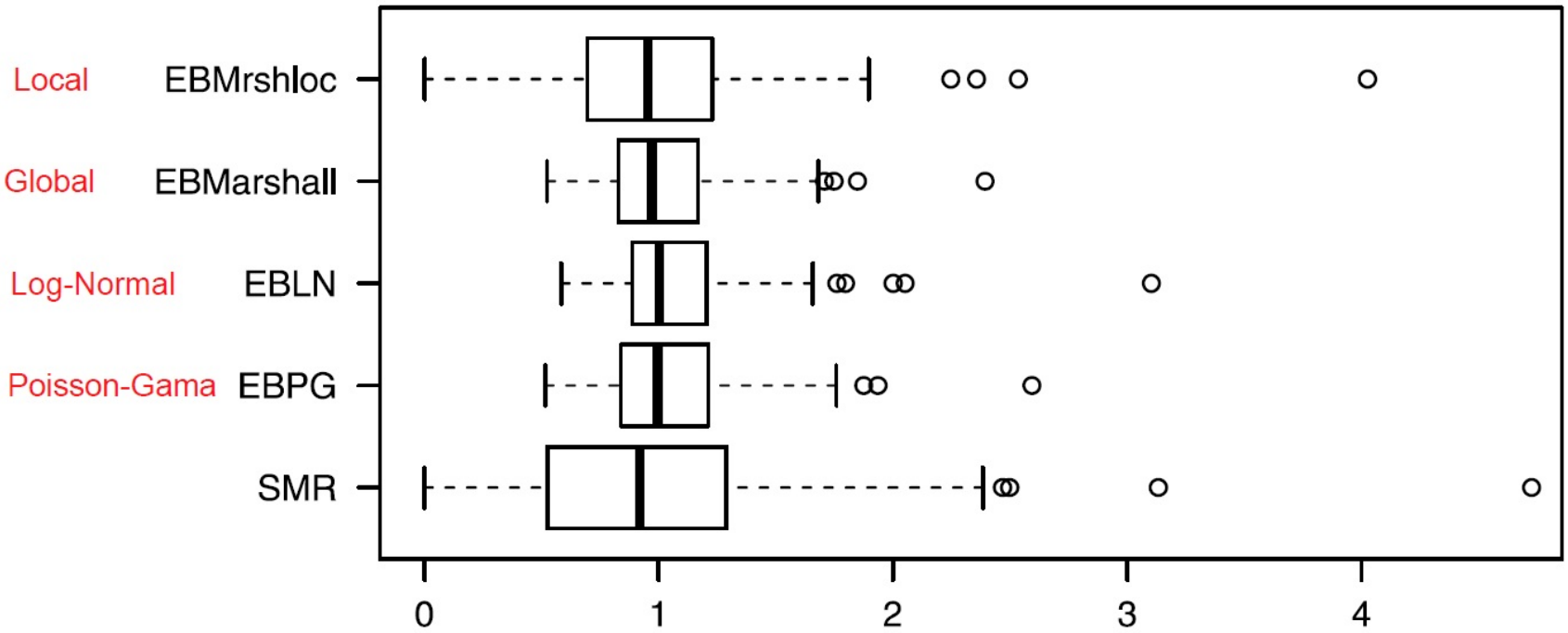


Fig.5 : Comparação do SMR e estimadores EB do risco relativo.

No slide anterior, veja que o SMR tem maior variabilidade e os demais estimadores foram comprimidos em torno da média global (≈ 1).

A estimação baseada no Poisson-Gama pode não convergir em algumas situações. Neste caso, outro estimador deve ser considerado.

O estimador proposto em Marshall (1991) também pode ser impraticável em circunstâncias similares.

O estimador EB baseado no modelo log-normal parece ser o mais estável e confiável computacionalmente.

Todos este estimadores EB geram estimativas suaves das taxas de risco tomando informação da área global. Entretanto (dependendo do tamanho da área total) pode ser razoável usar apenas um pequeno conjunto de áreas próximas (local) na estimação do risco (ex. áreas que dividem fronteira com a atual região).

Considere um conjunto reduzido de áreas das quais tomaremos informação emprestada. Uma opção é utilizar apenas as áreas vizinhas dentro de certa distância da área sob análise.

Marshall (1991) propôs outro estimador baseado apenas nas informações locais.

Para cada região, uma vizinhança é definida e médias, variâncias e fatores de mistura locais são obtidos de forma similar ao estimador global, mas usando apenas as áreas vizinhas. Isto produz uma mistura local para cada área, ao invés da mistura global do estimador anterior.

```
EBMrshloc <- EBlocal(observed, expected, ncCR85)$est
```

Os estimadores anteriores, não levaram em conta a forma em que as áreas estavam distribuídas na região de estudo. Se as áreas fossem permutadas aleatoriamente, as mesmas estimativas seriam obtidas. Com o novo estimador, o local exato das regiões é crucial (alterar a localizações forne estimativas distintas).

Embora vizinhança seja usualmente definida como "regiões que possuem fronteira comum", Cressie e Chan (1989) definem vizinhança quando a distância entre os centróides é pequena (< 30 milhas). Especificações distintas de vizinhança levam a resultados diferentes.

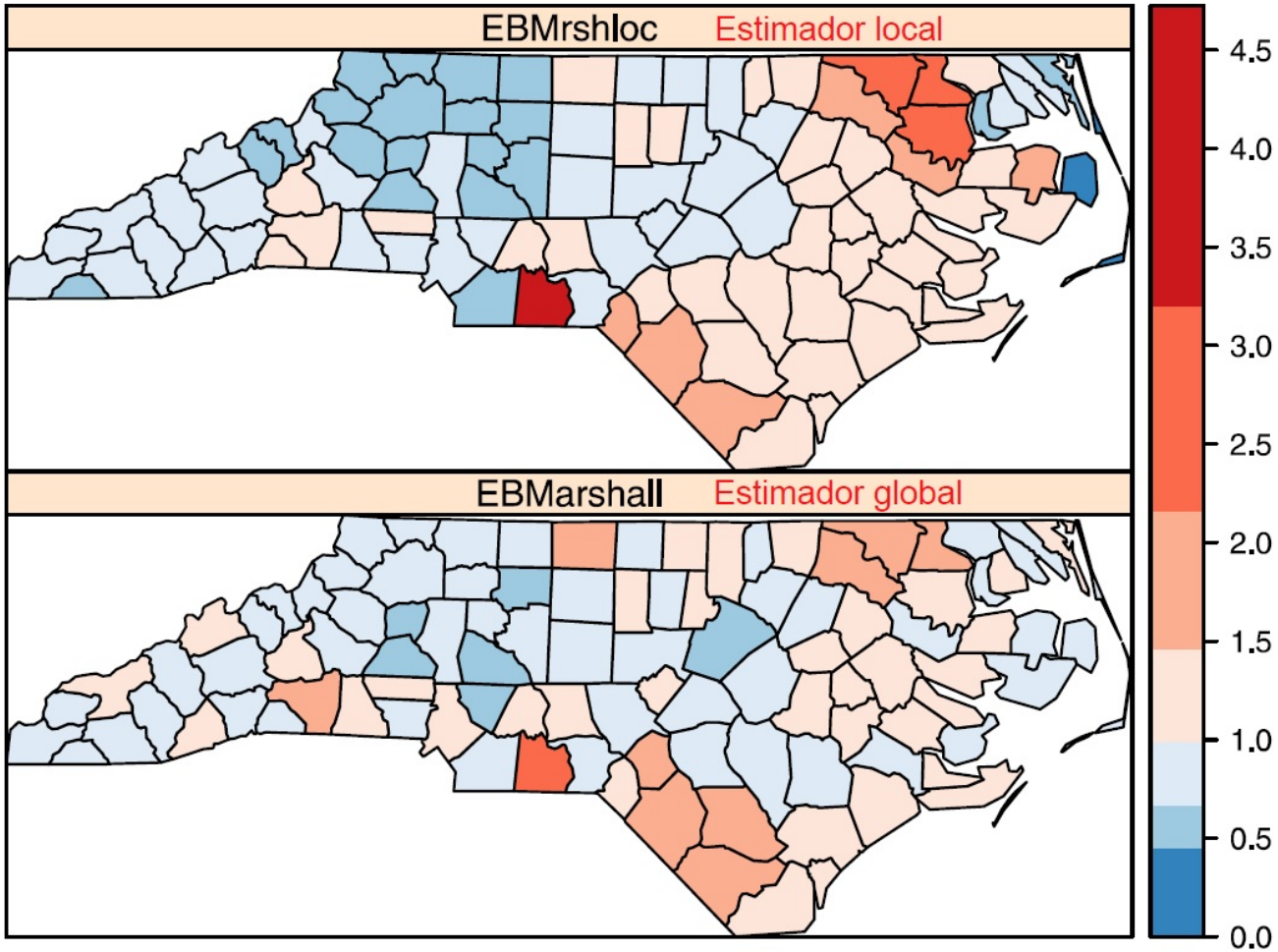


Fig.6 : Estimador EB de Marshall usando informação local (linha 1) e global (linha 2).

Os dois estimadores de Marshall são comparados na slide anterior.

A versão usando informação local produz estimativas suaves do risco relativo (mais concentradas em relação à média local e menos concentradas em relação à média global).

Reveja os boxplots da Figura 5 comparando os estimadores EB. O estimador local de Marshall também mostra uma redução de variabilidade em relação ao SMR, mas esta mudança é mais leve que nos demais estimadores.

Ajuste Bayesiano do modelo Poisson-Gama via Stan.

Script Stan (salvo em `scriptSTAN.stan`) definindo o modelo:

```
// data block
data{
  int<lower=1> n;
  int<lower=0> observed[n];
  vector[n] expected;
  real<lower=0> a_nu;
  real<lower=0> b_nu;
  real<lower=0> a_al;
  real<lower=0> b_al;
}

// parameters block
parameters{
  real<lower=0> nu;
  real<lower=0> alpha;
  vector[n] theta;
}

// Continua no próximo slide...
```

```
// Continuação do script Stan...

// transformed parameters block
transformed parameters{
  vector[n] mu;
  for(i in 1:n){
    mu[i] = theta[i]*expected[i];
  }
}

// model block
model{
  // verossimilhança
  for(i in 1:n){
    observed[i] ~ poisson(mu[i]);
    theta[i] ~ gamma(nu, alpha);
  }
  // distribuições a priori
  nu ~ gamma(a_nu, b_nu);
  alpha ~ gamma(a_al, b_al);
}
// deixe a linha final abaixo vazia (caso contrário o Stan reclama).
```

Fim do script em `scriptSTAN.stan`.

Carregando e organizando os dados em uma estrutura reconhecida pelo Stan:

```
# Dados:
require(spdep)
data(nc.sids)
observed = nc.sids$SID74
population = nc.sids$BIR74
n = length(observed)
r = sum(observed)/sum(population)
expected = population * r

# Especificações a priori
# nu ~ Ga(a_nu, b_nu) e alpha ~ Ga(a_al, b_al):
a_nu = 0.1;   b_nu = 0.1;   a_al = 0.1;   b_al = 0.1

# Organizando em lista todos os elementos que o Stan vai usar.
data = list(n=n, observed=observed, expected=expected,
            a_nu=a_nu, b_nu=b_nu, a_al=a_al, b_al=b_al)
```

```
# Nomes dos parametros a serem estimados
pars = c("mu","theta","nu","alpha")

# Chutes iniciais
initvals = list()
initvals[[1]] = list(theta=runif(n,0.5,1), nu=0.5, alpha=0.5)

# Aspectos relacionados ao MCMC
iter = 10000 # total de iterações (incluindo warm-up).
warmup = 5000
chains = 1

modelo = "scriptSTAN.stan"
Mstan = stan_model(modelo) # Pre-compilar o script Stan do modelo.

# Executar o MCMC via Stan.
output = sampling(Mstan, data = data,
                  iter = iter, warmup = warmup, chains = chains,
                  pars = pars, init = initvals, verbose = FALSE)
```

Após a executar o `Stan` teremos amostras da distribuição *a posteriori* de ν e α.
Através delas calcule estimativas pontuais e intervalos de credibilidade.

As médias *a posteriori* são iguais a $\hat{\nu} = 6.112$ e $\hat{\alpha} = 5.829$.
Estes valores são um pouco maiores do que aqueles obtidos via estimador EB (caso clássico).

Temos também estimativas *a posteriori* para os riscos relativos.

A próxima figura sumariza os intervalos de credibilidade de 95% dos riscos relativos para cada região.

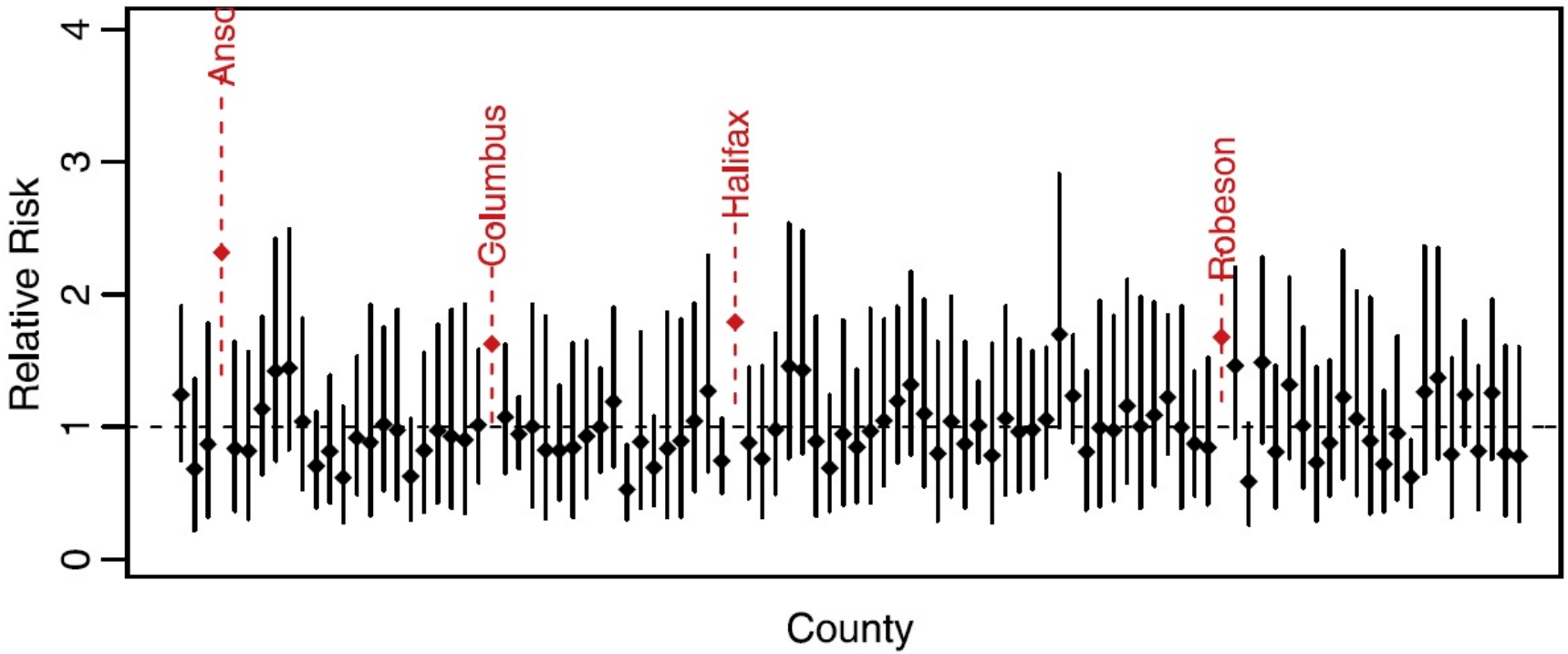


Fig.7 : Intervalos de credibilidade (95%) dos riscos relativos obtidos via ajuste Bayesiano para o modelo Poisson-Gama; intervalos acima de 1 em vermelho e ponto preto = mediana.

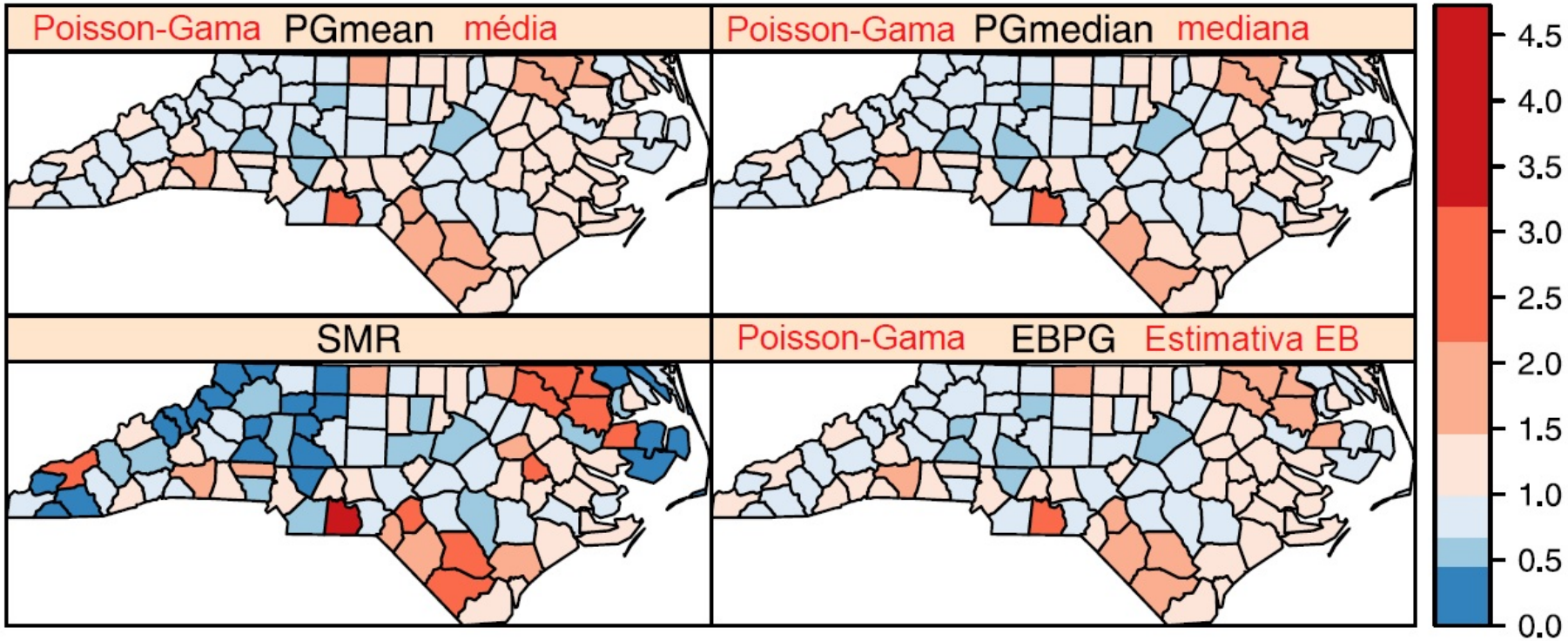


Fig.8 : Comparação de estimativas do risco relativo obtidas via EB e via abordagem Bayesiana completa (modelo Poisson-Gama).

Modelagem espacial.

Estrutura espacial pode ser incluída considerando o CAR. Covariáveis podem explicar parte da variabilidade do risco relativo. Cressie e Chan (1989) usam a "proporção de nascimentos não-brancos" como covariável.

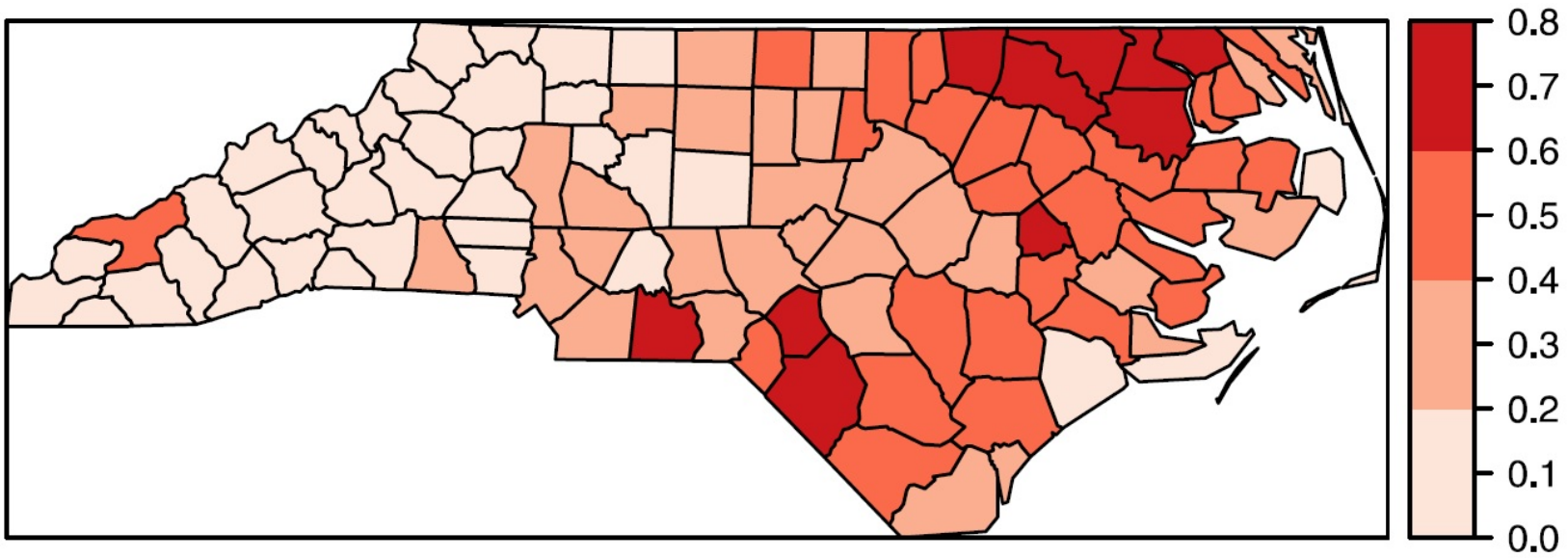


Fig.9 : Proporção de nascimentos não-brancos na Carolina do Norte (1974-1978). Padrão similar ao das estimativas do risco relativo.

Especificação CAR para um conjunto de variáveis aleatórias $\{v_i\}_{i=1}^n$:

$$(v_i|v_{-i}) \sim N\left(\rho \sum_{j\sim i} \frac{w_{ij} v_j}{\sum_j w_{ij}}, \frac{\tau_v}{\sum_j w_{ij}}\right).$$

Em que:

$\sum_j w_{ij} = w_{i+}$.

$w_{ij} = 1$ se $i \sim j$ (0, caso contrário),

$\tau_v =$ variância condicional do modelo CAR,

$\rho =$ parâmetro que torna a conjunta de $(v_1, \cdots, v_n)$ uma distribuição própria, a qual será Normal Multivariada.

O CAR é muito usado como distribuição *a priori* para efeitos aleatórios espaciais.

Dada a estrutura do CAR, é necessário conhecer os vizinhos de cada região. Eles podem ser definidos de formas diferentes, dependendo do tipo de relacionamento entre as áreas.

Em nosso exemplo, dois condados são vizinhos se a distância Euclidiana entre os centróides for menor que 0.4. Este limiar estabelece de 2 a 16 vizinhos por condado (Obs.: evite escolher um limiar tal que alguma região fique sem vizinho).

Para verificar significância da covariável, use o intervalo de credibilidade do coeficiente. Se o intervalo de 95% não incluir o 0, assuma que o coeficiente correspondente é significativo.

Se a estimativa desse coeficiente for positiva, o aumento da covariável implica em aumento do risco.

Nos dados SIDS, temos disponível o n^o de nascimentos de não-brancos em cada condado da Carolina do Norte.

A variável etnia é usada nos EUA como uma fonte de informação indireta do índice de nível social. Esta variável em nosso modelo pode ajudar a explicar parte da variabilidade espacial do risco de SIDS.

Calcule a "proporção de nascimentos não-brancos" (por região) como segue.

```
prop <- nc.sids$NWBIR74 / nc.sids$BIR74
```

O gráfico mostrando a variação da proporção de nascimentos de não-brancos foi apresentado anteriormente. Note que existe um padrão similar àquele da distribuição espacial do SMR e das diferentes estimativas EB.

O script `Stan` do modelo Bayesiano é mostrado a seguir.

Script do modelo salvo no arquivo `scriptSTAN.stan`.

```
// data block
data{
  int<lower=1> n;
  int<lower=0> observed[n];
  vector[n] expected;
  vector[n] prop;
  matrix[n,n] Sig;
  real m0;
  real<lower=0> v0;
  real m1;
  real<lower=0> v1;
  real<lower=0> a_u;
  real<lower=0> b_u;
  real<lower=0> a_v;
  real<lower=0> b_v;
}

// Script continua a seguir...
```

```
// Continuando script...

// parameters block
parameters{
  real beta0;
  real beta1;
  real<lower=0> tau_u;
  real<lower=0> tau_v;
  vector[n] u;
  vector[n] v;
}

// transformed parameters block
transformed parameters{
  vector[n] theta;
  for(i in 1:n){
    theta[i] = exp(beta0 + beta1*prop[i] + u[i] + v[i]);
  }
}

// Script continua a seguir...
```

```
// Continuando script...

// model block
model{
  // verossimilhança
  for(i in 1:n){
    observed[i] ~ poisson(theta[i]*expected[i]);
    u[i] ~ normal(0, sqrt(tau_u));
  }
  // distribuições a priori
  v ~ multi_normal(rep_vector(0,n), tau_v*Sig);
  tau_u ~ gamma(a_u,b_u);
  tau_v ~ gamma(a_v,b_v);
  beta0 ~ normal(m0,sqrt(v0));
  beta1 ~ normal(m1,sqrt(v1));
}
// deixe a linha final abaixo vazia (caso contrário o Stan reclama).
```

Os passos para executar o MCMC via `Stan` são semelhantes àqueles explicados na aula sobre o ajuste do modelo Normal via `Stan`.

Parâmetros a serem avaliados: risco relativo θ_i, intercepto β_0, coeficiente β_1 e os efeitos aleatórios u_i (não espacial) e v_i (espacial).

Cuidado! Para fazer o mapa da Carolina do Norte exibindo os condados e suas estimativas (em tonalidades de cores), esteja atento à ordenação dos condados na `data.frame` do `R` e à ordenação dos condados no `shape file` do mapa. Estas duas odenações devem ser compatíveis para evitar um mapa embaralhado.

Convergência da cadeia de Markov deve ser verificada antes de realizarmos inferência. O pacote `coda` inclui algumas opções de testes de convergência.

O critério de Geweke (Geweke, 1992) é um teste que compara as médias da 1^a e última parte da cadeia (10% inicial e 50% final).

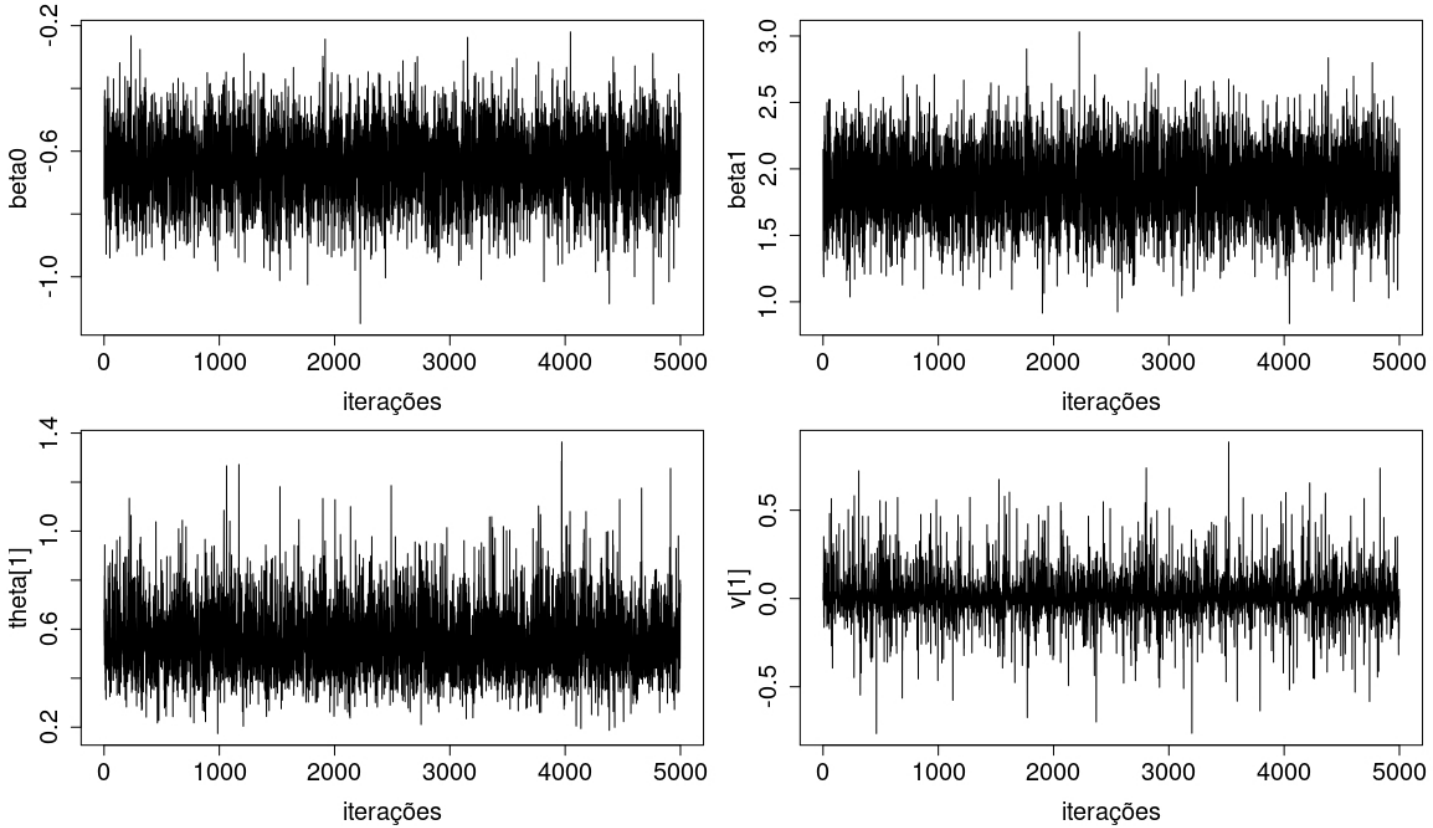
Se há convergência, as duas médias devem ser iguais. Estatística de teste entre -1.96 e 1.96 indica convergência. Estes são os escores $z_{0.025}$ e $z_{0.975}$ da N(0,1).

```
------------------------------------------------------------
samp = extract(output)
sbeta0 = samp$beta0; sbeta1 = samp$beta1; stheta = samp$theta
su = samp$u; sv = samp$v

require(coda)
aux = cbind(sbeta0, sbeta1, stheta[,1], sv[,1])
colnames(aux) = c("beta0", "beta1", "theta[1]", "v[1]")
aux = as.mcmc(aux); geweke.diag(aux)
------------------------------------------------------------
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

   beta0    beta1 theta[1]     v[1]
  0.0672  -1.0674   0.9463   1.4946
------------------------------------------------------------
```

As quatro estatísticas estão entre -1.96 e 1.96 (temos convergência).



- As cadeias parecem convergir.
- $\beta_1 > 0$ e seu HPD (95%) não irá conter 0, portanto, o risco é maior para regiões com maior proporção de nascimentos não-brancos.

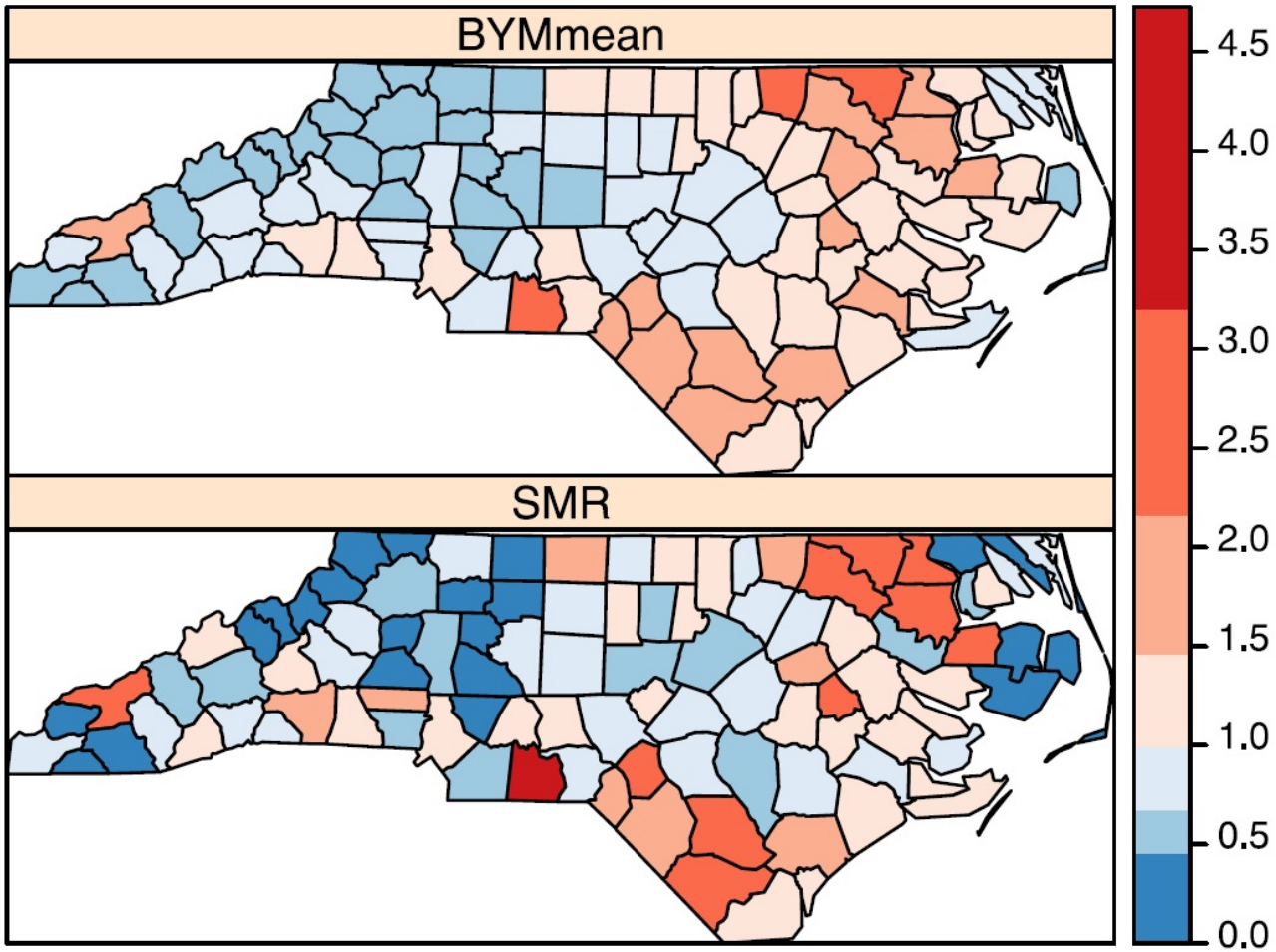


Fig.10 : Taxa de mortalidade padronizada e médias *a posteriori* dos riscos relativos obtidos com o modelo BYM. Comparação entre as estimativas suaves do BYM e do SMR.

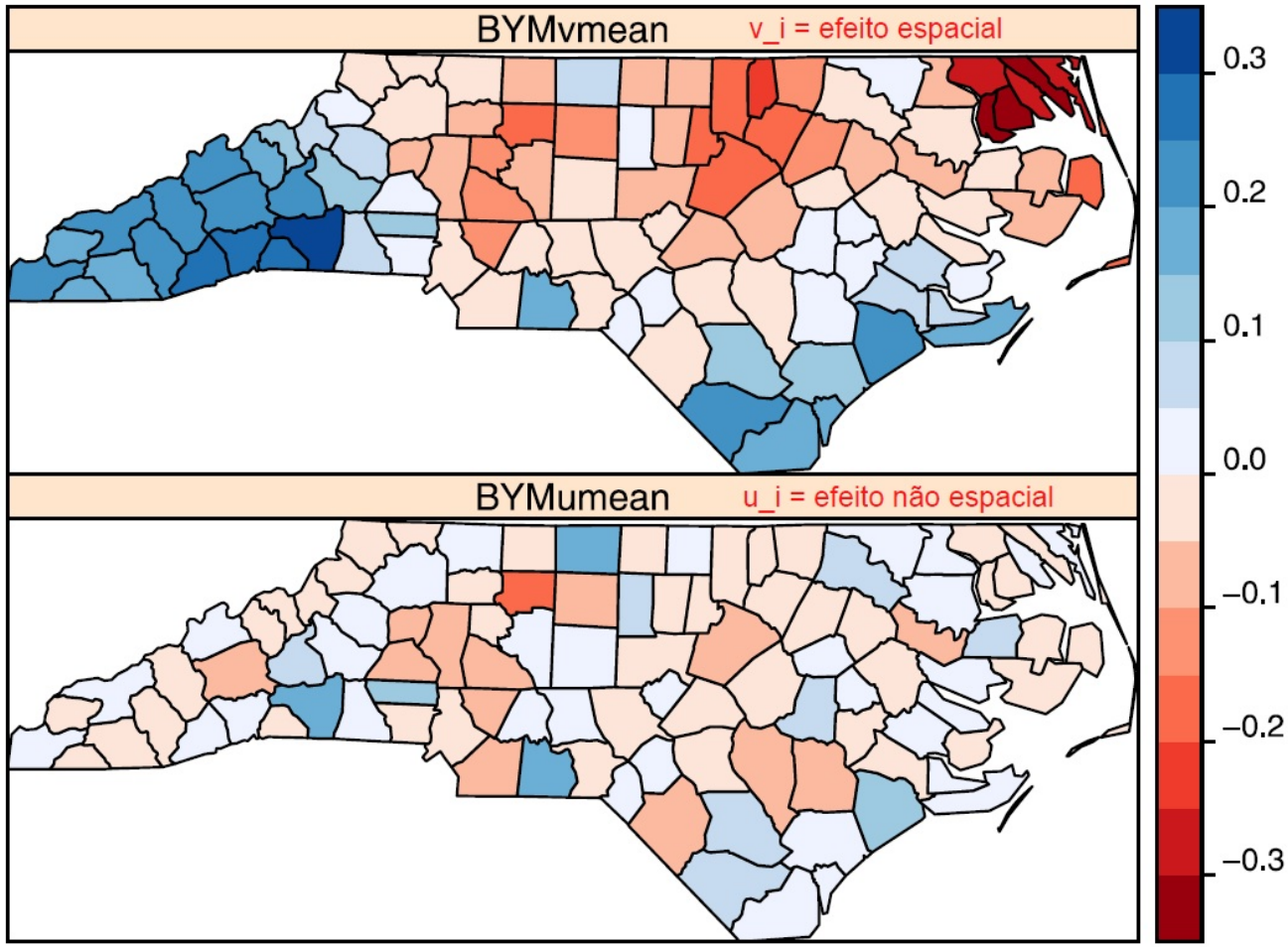


Fig.11 : Médias *a posteriori* dos efeitos aleatórios não-espacial (u_i) e espacial (v_i).

- Pouca variação (especialmente u_i). Isso se deve à escala log (escala original varia mais).
- Se o padrão espacial é fraco ou temos covariáveis explicando a estrutura espacial, os efeitos u_i e v_i ficariam não identificáveis (v_i seria um 2^o efeito de erro).

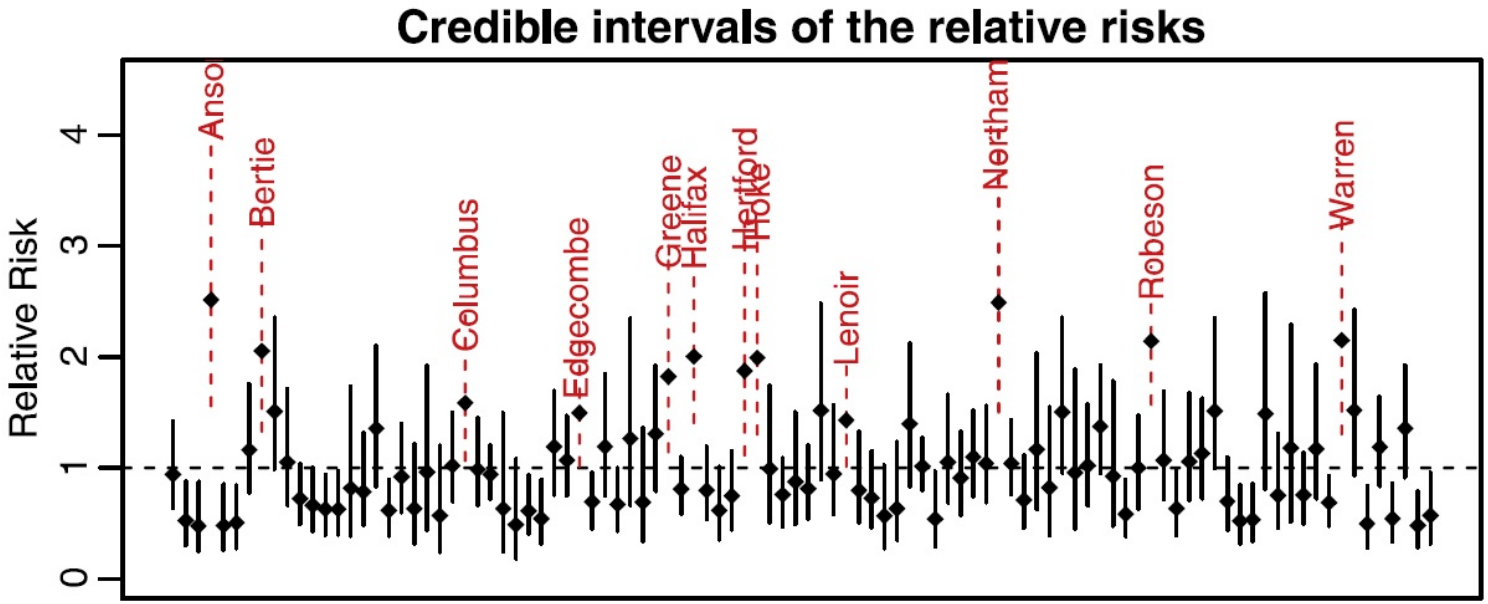


Fig.12 : Intervalos de credibilidade (95%) dos riscos relativos obtidos com o modelo BYM.

Os intervalos tracejados mostram condados para os quais o risco relativo é significativamente maior que 1.

Detecção de cluters.

Mapeamento de doenças fornece uma visão da distribuição espacial, entretanto, o interesse pode ser localizar zonas onde o risco é mais alto que o esperado.

Literatura: faz distinção entre métodos para aglomeração (identificar clusters) e para acessar o risco em torno de uma fonte causadora.

Alguns métodos estão no pacote `DCluster` (Gomez-Rubio et al., 2005).
Ele usa diferentes modelos e o *bootstrap* para computar a significância dos valores de estatísticas de teste.

- Simular diferentes conjuntos de dados (reamostrar o n^o de casos em cada área).
- Calcular a estatística de teste para cada conjunto simulado.
- Ranquear as estatísticas de testes calculadas para determinar um valor-p.

Sob a suposição $O_i \sim \text{Poisson}(\theta_i E_i)$, dado o n^o total de casos O_+, a distribuição de $(O_1, \cdots, O_n)$ é Multinomial com probabilidades $(E_1/E_+, \cdots, E_n/E_+)$.

Além da Multinomial, o `DCluster` oferece a possibilidade de amostrar via:

- *Bootstrap* não paramétrico.
- Poisson (sem condicionar em O_+).
- Binomial Negativa (levando em conta sobredispersão nos dados).

`DCluster` realiza um teste *bootstrap* baseado em:

- $O_i \sim \text{Poisson}(\theta_i E_i)$ e sem condicionar em O_+.
- $(O_1, \cdots, O_n) \sim \text{Mult}(O_+, [E_1/E_+, \cdots, E_n/E_+])$.
- $O_i \sim$ BNegativa ou $O_i \sim \text{Pois}(\theta_i E_i)$ com $\theta_i \sim$ Gama.

Testando a homogeneidade dos riscos relativos.

Antes de detectar clusters, investigue a heterogeneidade dos riscos relativos.

Podemos testar se há diferença entre distintos riscos relativos.

Possível razão desta diferença: presença de um fator de risco variando no espaço (ex. fonte poluidora aumentando o risco na redondeza).

Em cada área temos o n^o de casos observados e esperados, então um teste qui-quadrado pode ser aplicado para ver se há diferença (global) significativa entre estas duas quantidades.

Estatística de teste: $Q = \sum_{i=1}^{n} \frac{(O_i - \theta E_i)^2}{\theta E_i}$.

Assintoticamente, $Q \sim \chi^2_n$ (assuma: $\theta = \text{SMR} = \sum_i O_i / \sum_i E_i$).
Se a padronização interna foi usada para obter E_i, então $\theta = 1$ e $Q \sim \chi^2_{n-1}$, dada a restrição adicional $\sum_{i=1}^{n} O_i = \sum_{i=1}^{n} E_i$.

```
chtest <- achisq.test(observed~offset(log(expected)),
                      as(nc,"data.frame"),"multinom", 999)

Chi-square test for overdispersion

 Type of boots.: parametric
 Model used when sampling: Multinomial
 Number of simulations: 999
 Statistic:  225.5723
 p-value :  0.001
```

$H_0 : O_i \sim \text{Poisson}(E_i), \quad i = 1, \cdots, n.$
$H_1 : O_i \sim \text{Poisson}(\theta_i E_i), \quad$ ou seja, $\quad \theta_i \neq 1$ para pelo menos um i.

Gerar 999 conjuntos de dados (sob H_0) e compará-los ao caso real.

- $O_i \sim \text{Poisson}(\theta_i E_i)$. $E(O_i|\theta_i E_i) = \theta_i E_i$. Se $\theta_i = 1$, então $E(O_i|\theta_i E_i) = E_i$.
- Se $\theta_i = 1$, $\log(\theta_i E_i) = 1 \times \log(E_i)$,
  offset = covariável $\log(E_i)$ com coeficiente $= 1$.
- Se $\log(\theta_i E_i) - \log(E_i) = 0$, então $\log(\theta_i E_i / E_i) = 0$ e $\theta_i = 1$.

Reamostragem pode ser útil para n pequeno ou quando deseja-se um teste Monte Carlo usando a Binomial Negativa.

Um teste χ^2_{n-1} exato (sem reamostragem) pode ser feito. Versão unilateral:

```
1 - pchisq(chtest$t0, 100-1)
```

```
7.135514e-12
```

Potthoff e Whittinghill (1966) criaram um teste de homogeneidade das médias de diferentes variáveis Poisson. Ele pode ser usado para verificar a homogeneidade dos riscos relativos.

$H_0 : \theta_1 = \cdots = \theta_n = \lambda$.
$H_1 : \theta_i \sim \text{Ga}(\lambda^2/\sigma^2, \lambda/\sigma^2)$, ou seja, Gama com média λ e variância σ^2.

Estatística de teste: $PW = E_+ \sum_{i=1}^{n} \dfrac{O_i(O_i - 1)}{E_i}$.

H_1 também diz que $O_i \sim$ BNegativa (logo é um teste de sobredispersão).

Teste Monte Carlo:

```
pwtest <- pottwhitt.test(observed~offset(log(expected)),
                         as(nc, "data.frame"), "multinom", 999)
```

Notação: $O_+ = \sum_i \sum_j O_{ij}$ e $E_+ = \sum_i \sum_j E_{ij}$

Distribuição assintótica da estatística:
Normal com média $O_+(O_+ - 1)$ e variância $2\ n\ O_+(O_+ - 1)$.

Um teste exato unilateral pode ser feito como segue:

```
Oplus <- sum(observed)
1 - pnorm(pwtest$t0, Oplus*(Oplus-1), sqrt(2*100*Oplus*(Oplus-1)))
```

```
0
```

A avaliação inicial sobre clusters no mapa pode ser feita com base na autocorrelação. O teste χ^2 detecta se há diferenças claras entre os riscos relativos. Ele não avalia se há estrutura espacial nestas diferenças. Isto é feito no próximo teste.

Teste *I de Moran* para autocorrelação espacial.

Aplique o "I de Moran" ao SMR para avaliar a distribuição espacial populacional.

"I de Moran" aplicado ao O_i pode determinar autocorrelação espacial devido à distribuição espacial da população (maior população, maior n^o de casos).

Utilize o tamanho populacional ou o n^o de expostos como segue:

```
col.W <- nb2listw(ncCR85, zero.policy = TRUE)
moranI.test(observed~offset(log(expected)), as(nc,"data.frame"),
            "negbin", 999, listw = col.W, n = length(ncCR85),
            S0 = Szero(col.W))

Moran's I test of spatial autocorrelation

 Type of boots.: parametric
 Model used when sampling: Negative Binomial
 Number of simulations: 999
 Statistic:  0.2395172
 p-value : 0.001
```

No slide anterior, temos valor-p < 0.05 indicando "rejeitar H_0".
Conclusão: Existe autocorrelação espacial.

Etapas:

- Calcule o "I de Moran" para os dados originais.
- Simule O_i's usando o modelo Binomial Negativo (sobredispersão).
  O modelo Poisson-Gama é usado para amostragem no teste *bootstrap*.
- p-valor = razão dada por
  "frequência do *I de Moran* original" / n^o de simulações.

Teste de Tango para aglomeração geral.

Proposto em Tango (1995), este teste compara o n^o de casos observados e esperados de cada região.

Diferentes tipos de interações entre vizinhos podem ser consideradas.
O autor sugere uma medida de intensidade baseada em uma função decrescente para a distância ente regiões.

Estatística de teste: $T = (r - p)' \, A \, (r - p)$

sendo:

- $r' = (O_1/O_+, \cdots, O_n/O_+)$,
- $p' = (E_1/E_+, \cdots, E_n/E_+)$,
- $A =$ matriz de proximidade tal que $a_{ij} = \exp\{-d_{ij}/\phi\}$,
  $d_{ij} =$ distância entre os centróides de i e j,
  $\phi =$ constante positiva refletindo a força da dependência entre áreas e a escala em que a interações ocorrem.

A matriz de dependência A é construída usando algumas funções do R (pacote spdep). Assuma $\phi = 100$ (decrescimento suave) e use os comandos abaixo:

```
data(nc.sids)
idx <- match(nc.sids$NAME, rownames(nc.sids))
x <- nc.sids$x[idx]
y <- nc.sids$y[idx]
coords <- cbind(x, y)
dlist <- dnearneigh(coords, 0 ,Inf)
dlist <- include.self(dlist)
dlist.d <- nbdists(dlist, coords)
phi <- 100
col.W.tango <- nb2listw(dlist, glist = lapply(dlist.d,
               function(x, phi) { exp(-x/phi) }, phi=phi), style = "C")
```

O local aproximado do condado é obtido em `nc.sids` (colunas `x` e `y`), os quais estão no formato de projeção UTM (zona 18).

Depois de calcular a matriz de adjacências, realize o teste de Tango sobre presença global de clusters.

```
tango.test(observed~offset(log(expected)), as(nc,"data.frame"),
           "negbin", 999, listw = col.W.tango, zero.policy = TRUE)

Tango's test of global clustering

 Type of boots.: parametric
 Model used when sampling: Negative Binomial
 Number of simulations: 999
 Statistic:  0.000483898
 p-value :   0.049
```

- H_0 : Sem associação espacial vs. H_1 : Existe associação espacial.
- $999 = n^o$ de simulações Monte Carlo.
- Estatística de teste próxima de 0 (indica pouca força).
- Valor-p é pouco menor que 0.05 (quase não rejeita H_0).

Detecção da localização de um cluster.

Até aqui, estudamos métodos que apenas acessam a heterogeneidade dos riscos em um mapa, fornecendo uma avaliação geral sobre a presença de clusters.

Para detectar a localização dos clusters, um grupo de métodos útil inclui as *estatísticas scan* (Hjalmars et al., 1996).

Estes métodos baseiam-se em uma "janela móvel" (cobrindo poucas áreas a cada instante). O teste de cluster é realizado localmente, usando a captura da janela. Isto permite detectar os locais dos clusters.

Métodos *scan* geralmente diferem: (*i*) forma em que a janela é definida, (*ii*) como ela se move no mapa e (*iii*) como o teste local de clusters é realizado.

Revisão: Statistics in Medicine (Lawson, Gangnon and Wartenburg, editors, 2006).

GAM - Máquina de Análise Geográfica.

Talvez o 1^o método "scan" (Openshaw et al., 1987).

Considera um grid regular de pontos $\{(x_k, y_k)\}_{k=1}^{p}$ sob o mapa, para o qual uma janela circular é definida.

O teste usa apenas as áreas com centróides dentro da janela.

Ele compara o n^o total de casos observados na janela O_{k+} com o total de casos esperados na janela E_{k+}. Objetivo: verificar se O_{k+} é significativamente maior.

Teste (unilateral) com p-valor calculado sob a suposição $O_{k+} \sim \text{Poisson}(E_{k+})$.

Se O_{k+} não seguir a Poisson, é possível obter o valor-p via simulação.

Se valor-p $< \alpha$, rejeite "H_0 : não há cluster".
Neste caso, o círculo (ou seu centróide) será demarcado no mapa.

```
sidsgam <- opgam(data = as(nc, "data.frame"), radius = 30,
                 step = 10, alpha = 0.002)
gampoints <- SpatialPoints(sidsgam[,c("x","y")] * 1000,
                           CRS("+proj=utm +zone=18 +datum=NAD27"))

library(rgdal)
ll <- CRS("+proj=longlat + datum=NAD27")
gampoints <- spTransform(gampoints, ll)
gam.layout <- list("sp.points", gampoints)
```

Cuidado com a manipulação de coordenadas: projetar os centróides de volta para longitude/latitude para ser capaz de inserí-los no mapa da Carolina do Norte.

Se o mapa inteiro é escaneado, vários locais com clusters sobrepostos serão encontrados (figura a seguir). O motivo é que realizam-se testes independentes detectando clusters similares (duas janelas podem envolver as mesmas áreas). O GAM é criticado por isso.

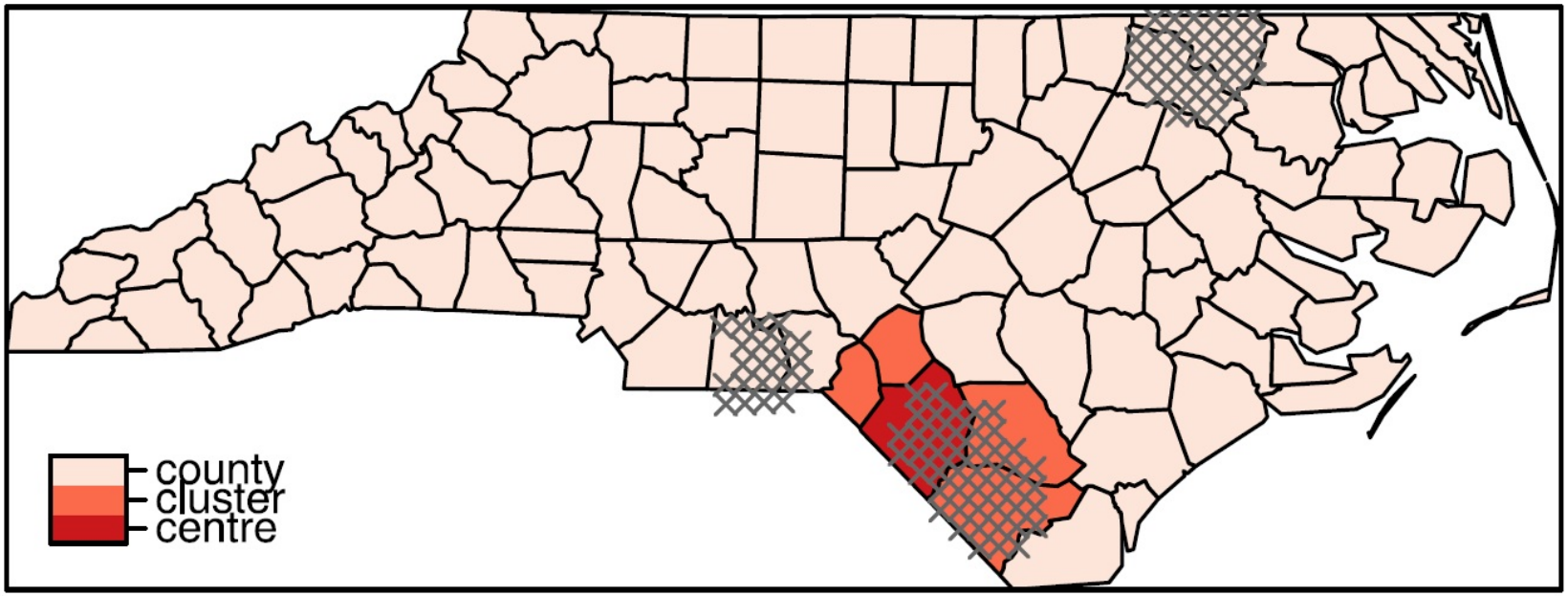


Fig.13 : Resultados dos testes GAM e Kulldorff (a seguir). Sinais "×" indicam os centros dos círculos obtidos via GAM. Cores mostram resultados do Kulldorff.

Estatística Kulldorff.

Detecção de clusters baseada em janelas de tamanhos variados.
Seleciona o cluster mais provável em uma região.

Janela tem formato circular.
Compara o risco relativo das regiões de dentro (D) com as de fora (F) da janela.

H_0 : Não há conglomerado, ou seja, riscos D = risco F.
H_1 : Existe conglomerado, ou seja, risco D > risco F.

Teste da razão de verossimilhanças (TRV). Vantagens:

- Cluster mais provável = janela com o menor valor da razão de verossimilhança.
- Não precisa corrigir o valor-p (simulações independentes para cada centro).

TRV: Seja L a verossimilhança, temos: $\lambda(X) = \sup_{\Theta_0}\{L\}/\sup_{\Theta}\{L\}$.

Rejeite H_0 se $\lambda(X) < c \in (0,1)$.

Assuma $O_i \sim \text{Poisson}(\theta_i E_i)$, com θ_i = risco relativo da região i.

H_0 : sem cluster $(\theta_D = \theta_F)$ vs. H_1 : com cluster $(\theta_D \neq \theta_F)$.

Para o modelo Poisson, a expressão da estatística de teste é dada por:

$$\max_{z \in Z_i} \left(\frac{O_z}{E_z} \right)^{O_z} \left(\frac{O_+ - O_z}{E_+ - E_z} \right)^{O_+ - O_z}$$

sendo z elemento de Z_i = conjunto dos círculos centrados na região i.

Círculos são definidos de forma a envolver até uma proporção da população total. O raio é aumentado até atingir a proporção desejada da população.

Mesmo selecionando o cluster mais provável em cada região, este pode não ser significativo. Entretanto, pode-se ter mais do que 1 cluster significativo em torno de 2 ou mais regiões diferentes (clusters podem estar sobrepostos).

Encontrando mais de 1 cluster, aquele com o menor valor-p é dito "primário" (mais importante). Clusters secundários (sem sobrepor o primário) podem ser considerados também.

```
mle <- calculate.mle(as(nc, "data.frame"), model = "negbin")
the grid <- as(nc, "data.frame")[, c("x","y")]
knresults <- opgam(data = as(nc, "data.frame"),
                   thegrid = thegrid, alpha = 0.05, iscluster = kn.iscluster,
                   fractpop = 0.15, R = 99, model = "negbin", mle = mle)
```

- `opgam`: escaneia uma área buscando clusters através do método indicado em `is.cluster` (default: `gam.iscluster`).
- `kn.iscluster`: para cada centróide, esta função determina se há ou não um cluster através do cálculo da estatística de kulldorff.
- `fractpop`: proporção da população total a ser atingida no ajuste do raio da janela.
- `calculate.mle` (pacote `DCluster`): fornece uma lista de estimativas dos parâmetros envolvidos no modelo (amostragem via *bootstrap*). No caso Binomial Negativo: temos n^o total de regiões n e "tamanho" $= n^o$ de falhas ou sucessos calculados após estimar v e a da Ga(v,a).

Teste de Stone para clusters locais.

Suponha a presença de uma fonte poluidora e deseja-se investigar se existe risco mais elevado ao seu redor.

Teste (Stone, 1988): "H_1 : há tendência decrescente ao redor da fonte poluidora".

Denote: $\theta_{(1)}, \cdots, \theta_{(n)}$ sendo
riscos relativos das regiões ordenadas pela distância até a fonte.

$H_0 : \theta_{(1)} = \cdots = \theta_{(n)} = \lambda \quad vs. \quad H_1 : \theta_{(1)} \leq \cdots \leq \theta_{(n)}.$

$\lambda =$ risco relativo global (pode ser 1 se usar padronização interna).

Estatística de teste avalia o risco acumulado máximo até uma área i:

$$\max_i \left\{ \frac{\sum_{j=1}^{i} O_j}{\sum_{j=1}^{i} E_j} \right\}.$$

Cuidado! temos vício se o teste for usado em regiões com elevado risco observado. Tendência de detectar um falso cluster. Ver próximo slide.

Exemplo: Verificar se há risco elevado ao redor do condado de Anson, o qual foi identificado como área de alto risco.

```
stone.stat(as(nc, "data.frame"), region = which(nc.sids$NAME == "Anson"))
```

```
            region
4.726392  1.000000
```

Estatística de teste calculada com $i = 1$ (rever fórmula).

Tamanho do cluster $= 1$ (apenas "Anson").

O cluster é significativo (valor-p dado abaixo).

```
stone.test(observed~offset(log(expected)),as(nc, "data.frame"),
                model = "negbin", 99, region = which(nc$NAME == "Anson"))
```

```
Stone's Test for raised incidence around locations

 Type of boots.: parametric
 Model used when sampling: Negative Binomial
 Number of simulations: 99
 Statistic:   4.726392
 p-values : 0.01
```