Universidade de Brasília

Instituto de Ciências Exatas

Departamento de Estatística

Dissertação de Mestrado

Uma Estatística de Varredura Espacial para Dados de Contagem com Censura

por

Roberto Lazarte Kaqui

Orientador: Prof. Dr. André Luiz Fernandes Cançado

Dezembro de 2016

Roberto Lazarte Kaqui

Uma Estatística de Varredura Espacial para Dados de Contagem com Censura

Dissertação apresentada ao Departamento de Estatística do Instituto de Ciências Exatas da Universidadede de Brasília como requisito parcial à obtenção do título de Mestre em Estatística.

Universidade de Brasília

Brasília, Dezembro de 2016

Termo de Aprovação

Roberto Lazarte Kaqui

Uma Estatística de Varredura Espacial para Dados de Contagem com Censura

Dissertação apresentada ao Departamento de Estatística do Instituto de Ciências Exatas da Universidadede de Brasília como requisito parcial à obtenção do título de Mestre em Estatística.

Data da defesa: 19 de Dezembro de 2016

Orientador:

Prof. Dr. André Luiz Fernandes Cançado
Departamento de Estatística, UnB

Comissão Examinadora:

Prof[a]. Dra. Cibele Queiroz da Silva
Departamento de Estatística, UnB

Prof. Dr. Fernando Luiz Pereira de Oliveira
Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto

Brasília, Dezembro de 2016

Ficha Catalográfica

KAQUI, ROBERTO LAZARTE
Uma Estatística de Varredura Espacial
para Dados de Contagem com Censura, (UnB - IE, Mestre em Estatística, 2016).

Dissertação de Mestrado - Universidade de Brasília. Departamento de Estatística - Instituto de Ciências Exatas.

1. Scan circular 2. Cluster 3. Dados de contagem 4. Dados censurados
5. *Software R* 6. Homicídios

Agradecimentos

Agradeço primeiramente à Universidade de Brasília (UnB), principal espaço responsável pela minha formação profissional.

Agradeço ao IPEA, por ter me permitido conciliar o mestrado com minhas obrigações no trabalho, e agradeço a todos os técnicos que me incentiveram a concluir o mestrado.

Agradeço a todos os professores e professoras que contribuíram direta e indiretamente na minha formação, tanto no período da graduação como agora no mestrado. Em especial, agradeço ao meu orientador, professor André Cançado, pela paciência e por toda a sabedoria transmitida. Desde a época da graduação, sempre esteve solícito para me ajudar e orientar, e agora, no mestrado, foi muito importante para a conclusão deste trabalho.

Agradeço aos professores membros da banca examinadora, Cibele Queiroz da Silva e Fernando Luiz Pereira de Oliveira, por aceitarem o convite para participar da defesa, e ao professor Antônio Eduardo Gomes, por participar da banca examinadora na qualificação deste trabalho.

Agradeço aos colegas do mestrado, pela boa convivência no período em que tivemos aulas juntos e pelas dicas de matérias e provas.

Agradeço ao Gabriel e a Karine, que são minha família aqui em Brasília. Mesmo não estando tão presente neste último ano, eles se mantiveram presentes em minha vida e ao meu lado, me fazendo sentir acolhido em todos os momentos em que estivemos juntos.

Agradeço às minhas amigas do IPEA, Raquel, Akina e Patrícia, por todos os bons momentos que tivemos em almoços e lanches, por tornarem meu dia a dia no trabalho muito mais divertido e alegre, e por toda a ajuda e conselhos que me deram

nesse período do mestrado.

Por último e com muito carinho, agradeço imensamente à minha família. O amor incondicional deles, mesmo estando distantes, é a maior fonte de incentivo e de coragem para que consiga alcançar meus objetivos. Poder ver o orgulho deles a cada momento que concluo mais uma etapa na minha vida será sempre um dos maiores motivos de minha alegria nesta vida.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

O presente trabalho tem como objetivo propor uma estatística de varredura espacial para dados de contagem com censura. Essa extensão consiste em adaptações no método de varredura tradicional, também conhecido como método Scan, que permitem incorporar a informação da censura no processo de estimação da estatística razão de verossimilhança e no procedimento de verificação da significância do cluster detectado. Este método foi proposto com o intuito de melhorar a performance da estatística Scan na identificação de conglomerados em dados com a presença de censura. Os resultados mostraram que a extensão proposta é mais eficiente que a estatística Scan usual, uma vez que apresentou maior poder de detecção e maior precisão no processo de identificação do cluster.

Palavras Chave: *Scan circular, Cluster, Dados de contagem, Dados censurados, Software R, Homicídios.*

Abstract

This paper aims to propose an extension of the Spatial Scan Statistic for censored counting data. This extension consists of adaptations in the traditional Scan method that allows to incorporate the censored data into the estimation process of the likelihood ratio statistic and in the significance test of the detected cluster. This method was proposed with the aim of improving the performance of the Scan statistic in cluster detection in data with the presence of censoring. The results showed that the proposed extension is more efficient than the usual Scan statistic, since it presented greater detection power and greater precision in the cluster identification process.

Key words: *Spatial Scan, Cluster, Counting data, Censored data, R, Homicide.*

Capítulo 1

Introdução

Hoje em dia, cada vez mais se tem interesse em explicar ou descrever fenômenos espaciais, assim como avaliar sua interação. O aumento na disponibilidade de dados georreferenciados torna esse tipo de avaliação mais comum e acessível. Segundo Câmara et al. (2002), compreender a distribuição espacial de dados oriundos de fenômenos ocorridos no espaço constitui hoje um grande desafio para a elucidação de questões centrais em diversas áreas do conhecimento. De maneira geral, estudos que envolvem a posição geográfica do elemento observado buscam identificar se existe ou não algum tipo de regularidade espacial. De forma mais particular, dado um sistema de coordenadas geográficas, busca-se identificar alguma coleção ou conglomerado de regiões contido nesse sistema onde um particular evento de interesse ocorre com mais frequência que nos demais.

Atualmente, o método mais popular para detectar conglomerados espaciais é a estatística espacial Scan de Kulldorff (Kulldorff, 1997), também conhecida como estatística de varredura espacial. Esse método tem sido amplamente utilizado em virtude da sua fácil implementação computacional, do seu poder de detecção e da sua capacidade de atribuir um nível de significância à estatística de teste via simulação de Monte Carlo. Sua aplicação é possível em diversas áreas, como em estudos para detecção de conglomerados de doenças respiratórias infecciosas (Bakker et al., 2004), doenças do fígado (Ala et al., 2006), diabetes (Green et al., 2003), vigilância sindrômica (Kleinman et al., 2005), criminologia (Minamisava et al., 2009), doenças sexualmente transmitíveis (Jennings et al., 2005), entre outros.

Uma situação que tem ocorrido com frequência cada vez maior em estudos observacionais é a presença de dados censurados, muitas vezes gerados por limitações dos métodos analíticos de mensuração. Segundo Christofaroa e Leão (2014), a censura de dados interfere diretamente em quase todos os tipos de análises estatísticas. Apesar dessa interferência, os dados censurados não devem ser eliminados da série estudada, pois nessas situações, distorções ainda maiores podem ser geradas. Assim, uma vez que a presença de dados censurados prejudica a utilização dos testes estatísticos, técnicas específicas devem ser elaboradas para minimizar a interferência negativa das observações censuradas. Nesse sentido, Huang et al. (2007) apresentam uma extensão da estatística Scan utilizando a distribuição exponencial para ajustar dados censurados. Entretanto, essa extensão não é aplicável para o caso de dados discretos.

Nesta dissertação, apresentaremos uma estatística de varredura espacial para dados de contagem com censura. Essa extensão consistirá em uma adaptação no cálculo das funções de verossimilhança, de modo que a informação da censura será incorporada na estimação da estatística do teste e no processo de verificação da significância do cluster. Com essa nova formulação, esperamos obter melhores resultados do que os observados pelos modelos usuais da estatística Scan quando aplicados a dados de contagem com censura, uma vez que, estaremos aplicando ao dado censurado uma metodologia mais adequada à sua característica.

Para implementação do método proposto, utilizaremos o *software R* (R Core Team, 2015), e a aplicação será feita em diversos cenários simulados com o objetivo de comparar o desempenho da técnica Scan Circular tradicional e da extensão proposta em diferentes situações de tamanho, formato e número de regiões com censura dentro dos clusters. Além disso, iremos também, aplicar a metodologia sugerida neste dissertação a um banco de dados real com casos de homicídios registrados nos municípios do Rio de Janeiro para o ano de 2014, onde serão criados casos de censura artificias.

Este trabalho está organizado da seguinte forma: Nos capítulos 2, 3 e 4 estão apresentados os conceitos teóricos utilizados neste trabalho, em que o capítulo 2 corresponde aos conceitos da estatística Scan Circular de Kulldorff, o capítulo 3

corresponde ao tema de dados censurados com enfâse na área de *Análise de Sobrevivência* e o capítulo 4 corresponde aos fundamentos da estimação via métodos de otimização. No capítulo 5, é apresentada a metodologia deste trabalho, em que detalha como a análise foi realizada, tanto para os dados simulados, como para os dados reais. Em seguida, no capítulo 6, estão expostos as análises e resultados com relação aos dados simulados e aos dados reais. Por fim, no capítulo 7, é apresentada a conclusão deste trabalho, juntamente com as sugestões de trabalhos futuros.

Capítulo 2

Scan Circular de Kulldorff

2.1 Contexto Histórico

Nos últimos anos, estudos para detecção de conglomerados espaciais ganharam espaço na literatura e, assim, vários métodos foram propostos com esse objetivo. O primeiro método criado para detecção de conglomerados espaciais foi baseado em "quadrats", proposto por Choynowski (1959). O objetivo estava em estudar a distribuição espacial de casos de tumores no cérebro de uma certa região da Polônia, dividida em municípios. Este método fornece uma probabilidade para cada área onde foi aplicado, não fornecendo uma informação de significância global em relação a presença de cluster (Moura, 2006). Testando cada quadrante separadamente, surge o problema de múltiplos testes. Outro problema deste método era a incapacidade de detectar clusters que não seguissem as delimitações geográficas dos municípios da região em estudo.

Em 1965, seguindo as idéias apresentadas por Choynowski (1959), que baseia a modelagem através da probabilidade, Naus desenvolveu estudos para detecção de cluster em processos pontuais unidimensionais (Naus, 1965b), entretanto, em função do cálculo da probabilidade ser realizado em apenas uma dimensão, a aplicabilidade da técnica no contexto espacial fica comprometida. Ainda no mesmo ano, Naus faz a extensão do seu estudo para processos bidimensionais (Naus, 1965a). Dentre as limitações da técnica temos, além da ausência do cálculo exato da probabilidade, a

necessidade de que as duas dimensões consideradas na aplicação sejam paralelas aos eixos x e y e que tenham o formato fixo de um retângulo.

No artigo de Whittemore et al. (1987) é proposto um método para responder à questão sobre a existência de um cluster espacial, ao invés de estabelecer uma probabilidade para cada área como em Choynowski (1959). A estatística de Whittemore baseia-se na distância média entre todos os pares de casos, entretanto, sua formulação não permite a localização exata do cluster, além de mostrar sensibilidade para as variações na densidade populacional em diferentes localizações.

Já o método iterativo GAM ($GeographicalAnalysisMachine$), desenvolvido por Openshaw et al. (1988), também toma como base as idéias apresentadas por Choynowski (1959). O método faz uso de múltiplos círculos de raio R sobrepostos, permitindo que os conglomerados possam ter formas diferentes daquelas impostas pelas delimitações geográficas dos municípios da região em estudo. Apesar do método GAM ser de fácil compreensão e apresentar grande apelo visual, a técnica possui algumas desvantagens em relação a outros métodos, como o fato de ser um método exploratório e não inferencial.

Sob a hipótese de ausência de um cluster espacial, duas regiões distintas com a mesma quantidade de pessoas esperariam observar a mesma quantidade de ocorrências de determinada doença. Baseando-se nessa premissa, Turnbull et al. (1990) desenvolveram o CEPP − Cluster Evaluation Permutation Procedure, que também usa zonas circulares sobrepostas, de maneira que cada conjunto tenha população P constante. O estudo também compara o método CEPP com os propostos por Whittemore et al. (1987) e Openshaw et al. (1988) na detecção de gonglomerados em casos de leucemia ocorridos em parte do norte do estado de Nova Iorque.

Besag e Newell (1991) desenvolveram o teste TBN, em que o número de casos K é fixado previamente como o tamanho do conglomerado a ser identificado. Este método consiste em, fixado o tamanho K, centrar o círculo em um ponto na região, ir aumentando o seu raio e agregando os centróides vizinhos até que o círculo tenha agregado o menor número de centróides necessários para que o número de casos dentro do círculo tenha no mínimo K casos (Moura, 2006).

Tango (1995) desenvolveu o teste C_λ, onde o tamanho do conglomerado é deter-

minado por λ que, neste caso, é o parâmetro de escala de alguma função $g(\lambda)$ que mede a proximidade entre as áreas contidas no conglomerado.

Contudo, Moura (2006) indica que muitos desses métodos de detecção apresentam um problema comum, que é a definição à priori do parâmetro que caracteriza o tamanho do cluster: no GAM, o raio R do conglomerado, no CEPP, o tamanho populacional P, no TBN, o número K de casos, e no C_λ , o parâmetro λ. Diante dessa característica, os testes são repetidos usando valores diferentes para os parâmetros uma vez que as características do conglomerado em questão são desconhecidas. Consequentemente, além do vício de pré-seleção, os vários testes simultâneos resultam no problema de ajustes de múltiplos testes, que aumenta consideravelmente o nível de significância dos testes.

Com o intuito de contornar esses problemas e permitir uma avaliação global dos resultados, Kulldorff e Nagarwalla (1995) introduziram, a partir dos conceitos dos métodos GAM e CEPP, uma estatística para detectar áreas com elevada taxa de incidência para um evento de interesse. Esse método se baseia na razão de verossimilhança e utiliza uma estatística de varredura multidimensional, apoiando-se em três propriedades básicas: geometria da área sendo varrida, a distribuição de probabilidade que gera os casos sob a hipótese de completa aleatoriedade espacial e o tamanho e forma da área de varredura. Por fim, Kulldorff (1997) apresenta um método de varredura já amadurecida quando comparado com a de Kulldorff e Nagarwalla (1995), estabelecendo dessa forma a estatísca Scan de Kulldorff que seria amplamente utilizada nos anos posteriores. Os detalhes desse método serão apresentados de forma mais específica ao longo do capítulo 2.

Para maiores detalhes acerca de cada um dos métodos para detecção de conglomerados, consultar Araújo (2013).

2.2 Introdução

O metódo Scan Circular proposto por Kulldorff (1997) consiste em uma técnica para identificação e inferência de conglomerados espaciais. Por conglomerado espacial entende-se um subconjunto de regiões de um mapa em que a incidência de casos

para um fenômeno de interesse é discrepante do restantante do mapa (Figueiredo, 2010). Para sua aplicação, é necessário serem conhecidas três informações básicas para cada região em estudo:

- a população sob risco de ocorrência do evento estudado;
- o número de casos observados do evento;
- as coordenadas geográficas de um ponto definido arbitrariamente no interior da região, geralmente o centróide (centro geométrico).

Através de um processo iterativo, o método faz a varredura do mapa com áreas circulares em busca de conglomerados. As áreas circulares são limitadas por um critério preestabelecido, no caso, uma proporção máxima da população que poderá estar dentro delas. Esse processo tem como objetivo identificar um conglomerado onde o número de casos do evento de interesse seja significativamente maior (ou menor) que o esperado dentro da área circular estabelecida.

Uma vez identificado um potencial conglomerado, é realizado um teste de razão de verossimilhança para avaliar se o mesmo é significativo ou não do ponto de vista estatístico. Para auxiliar a decisão dos testes, são utilizadas simulações de Monte Carlo.

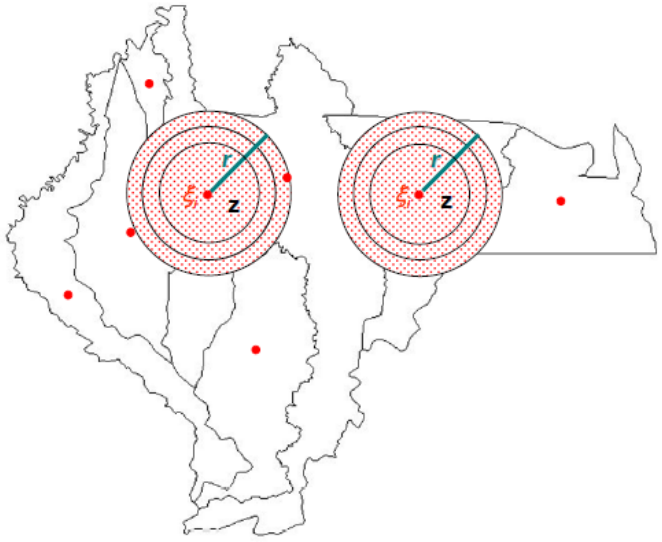


Figura 2.1: Exemplo de aplicação da varredura espacial da estatística scan.
Fonte: Adaptado de Balieiro (2008).

2.3 Notações básicas

Considere um mapa dividido em m regiões em que cada região i possui uma população em risco de tamanho n_i e um número de casos x_i que representa o número de ocorrências para um evento de interesse, como por exemplo, o número de casos de uma determinada doença. Define-se $N = \sum_{i=1}^{m} n_i$ e $C = \sum_{i=1}^{m} x_i$ como sendo, respectivamente, a população total em risco e o número total de casos do evento de interesse.

Considere uma zona z como sendo um subconjunto de regiões conexas do mapa e Z o conjunto dessas zonas . Um cluster é definido como sendo uma zona específica z para a qual a probabilidade θ_z de um indivíduo ser um caso do evento de interesse é maior que nas demais regiões fora da zona z.

Seja $x_z = \sum_{i \in z} x_i$ o número de casos na zona z, $x_{\bar{z}} = \sum_{i \notin z} x_i$ o número de casos fora de z, e $n_z = \sum_{i \in z} n_i$ e $n_{\bar{z}} = \sum_{i \notin z} n_i$ os tamanhos populacionais dentro e fora de z, respectivamente.

Dessa forma, a estatística Scan de Kulldorff é definida a partir de um teste de razão de verossimilhança, formulada sob as seguintes hipóteses:

$$\begin{cases} H_0 : \theta_z = \theta_0, \text{ para todo } z \in Z \\ H_1 : \text{Existe uma zona } z \text{ tal que } \theta_z > \theta_0, \end{cases} \tag{2.1}$$

onde θ_z é a probabilidade de ocorrer um caso na zona z e θ_0 a probabilidade de ocorrer um caso fora dela. A zona z será considerada um cluster caso o teste rejeite a hipótese nula de que não existe diferença significativa entre essas duas probabilidades.

Em situações onde os dados analisados são discretos, o número de casos x_i de uma regiao i é comumente modelado a partir das distribuições Binomial e Poisson, dependendo do tipo de contagem adotado. Segundo Silva (2012), quando o número de total de casos C é pequeno se comparado com N, os dois modelos se aproximam. Caso contrário, o modelo Binomial é mais adequado quando a contagem é resultante de uma contagem binária, enquanto que o modelo Poisson deve ser usado quando a contagem está relacionada a algum fator de risco contínuo. Para casos em que se trabalha com dados contínuos, Huang et al. (2007) e Kulldorff et al. (2009) apre-

sentam abordagens utilizando as distribuições Exponencial e Normal. Também já existem extensões para análises multivariadas (Kulldorff et al., 2007) e para estudos com dados ordinais (Jung et al., 2007). Neste estudo serão aprofundados apenas as abordagens para os casos discretos.

2.4 O modelo Poisson

Quando se analisa dados do tipo contagem, muito comum em estudos epidemiológicos, uma forma usual de modelar o número de casos x_i é assumindo a distribuição Poisson:

$$x_i \sim Poisson(n_i\theta_i).$$

Sob H_0, temos $\theta_i = \theta_0$, como definido por 2.1. Sendo assim, a verossimilhança assume a forma

$$\begin{aligned}\mathcal{L}_0(\mathrm{x}, \theta_0) = \prod_{i=1}^{m} \frac{e^{-n_i\theta_0}\left(n_i\theta_0\right)^{x_i}}{x_i!} &= \frac{e^{-\sum_i n_i\theta_0}\prod_i n_i^{x_i}\theta_0^{\sum_i x_i}}{\prod_i (x_i!)} \\ &= \frac{e^{-N\theta_0}\prod_i n_i^{x_i}\theta_0^{C}}{\prod_i x_i!}.\end{aligned} \tag{2.2}$$

Para a log-verossimilhança, tem-se que

$$\ell_0(\mathrm{x}, \theta_0) = \log(\mathcal{L}_0(\mathrm{x}, \theta_0)) = -N\theta_0 + \sum_i x_i \log n_i + C\log\theta_0 - \sum_i \log x_i!. \tag{2.3}$$

Derivando em relação a θ_0 e igualando a zero, temos

$$\frac{\partial \ell_0}{\partial \theta_0} = -N + \frac{C}{\theta_0} = 0. \tag{2.4}$$

Solucionando (2.4), conclui-se que $\hat{\theta}_0 = \frac{C}{N}$. De maneira análoga, sob H_1, temos

$$\begin{cases} H_0 : \theta_i = \theta_z \text{ , se } i \in z \\ H_1 : \theta_i = \theta_0 \text{ , se } i \notin z. \end{cases} \tag{2.5}$$

Nesse caso, a verossimilhança será escrita como

$$\begin{aligned}\mathcal{L}(\mathrm{z},\mathrm{x},\theta_0,\theta_z) &= \prod_{i\in z}\frac{e^{-n_i\theta_z}\left(n_i\theta_z\right)^{x_i}}{x_i!}\times\prod_{i\notin z}\frac{e^{-n_i\theta_0}\left(n_i\theta_0\right)^{x_i}}{x_i!}\\ &= \frac{e^{-\sum_{i\in z}n_i\theta_z}\prod_{i\in z}n_i^{x_i}\theta_z^{\sum_{i\in z}x_i}}{\prod_{i\in z}x_i!}\times\frac{e^{-\sum_{i\notin z}n_i\theta_0}\prod_{i\notin z}n_i^{x_i}\theta_0^{\sum_{i\notin z}x_i}}{\prod_{i\notin z}x_i!}.\end{aligned} \tag{2.6}$$

Aplicando o logaritmo para obter a log-verossimilhança, temos

$$\begin{aligned}\ell(\mathrm{z},\mathrm{x},\theta_0,\theta_z) &= \log(\mathcal{L}(\mathrm{z},\mathrm{x},\theta_0,\theta_z))\\ &= -n_z\theta_z+\sum_{i\in z}x_i\log n_i + x_z\log\theta_z-\sum_{i\in z}\log x_i!\\ &- n_{\bar{z}}\theta_0+\sum_{i\notin z}x_i\log n_i + x_{\bar{z}}\log\theta_0-\sum_{i\notin z}\log x_i!.\end{aligned} \tag{2.7}$$

Maximizando (2.7) em relação a θ_0 e θ_z,

$$\frac{\partial\ell}{\partial\theta_0}=0\Rightarrow\hat{\theta}_0=\frac{x_{\bar{z}}}{n_{\bar{z}}}, \tag{2.8}$$

$$\frac{\partial\ell}{\partial\theta_z}=0\Rightarrow\hat{\theta}_z=\frac{x_z}{n_z}. \tag{2.9}$$

Considerando (2.2) e (2.6), a razão de verossimilhança para o modelo Poisson é definida como

$$\begin{aligned}\lambda=\frac{\mathcal{L}}{\mathcal{L}_0} &= \frac{e^{-\sum_{i\in z}n_i\theta_z}\prod_{i\in z}n_i^{x_i}\theta_z^{\sum_{i\in z}x_i}}{\prod_{i\in z}x_i!}\times\frac{e^{-\sum_{i\notin z}n_i\theta_0}\prod_{i\notin z}n_i^{x_i}\theta_0^{\sum_{i\notin z}x_i}}{\prod_{i\notin z}x_i!}\\ &\times\left(\frac{e^{-N\theta}\prod_i n_i^{x_i}\theta^C}{\prod_i x_i!}\right)^{-1}=\frac{e^{-(n_z\theta_z+n_{\bar{z}}\theta_0-N\theta_0)}\theta_z^{x_z}\theta_0^{x_{\bar{z}}}}{\theta^C}.\end{aligned} \tag{2.10}$$

2.5 O modelo Binomial

Para estudos tipo caso-controle, onde um determinado indivíduo pode estar em um de dois estados possíveis, a distribuição usualmente utilizada para modelar o número de casos x_i é a distribuição Binomial:

$$x_i\sim Bin(n_i,\theta_i).$$

De maneira similar ao modelo Poisson, sob H_0, tem-se que $\theta_i = \theta_0$, obtendo assim a verossimilhança na forma

$$\mathcal{L}_0(\mathrm{x}, \theta_0) = \left[\prod_{i=1}^{m} \binom{n_i}{x_i}\right] \theta_0^C \left(1-\theta_0\right)^{N-C}. \tag{2.11}$$

Para a log-verossimilhança, tem-se que

$$\ell_0(\mathrm{x}, \theta_0) = \log(\mathcal{L}_0(\mathrm{x}, \theta_0)) = \sum_i \log \binom{n_i}{x_i} + C\log\theta_0 + (N-C)\log\left(1-\theta_0\right). \tag{2.12}$$

Derivando (2.12) em relação a θ_0 e igualando a zero,

$$\frac{\partial \ell_0}{\partial \theta_0} = \frac{C}{\theta_0} - \frac{N-C}{1-\theta_0} = 0. \tag{2.13}$$

Logo, tem-se $\hat{\theta}_0 = \frac{C}{N}$, assim como na Poisson.

Sob H_1, temos

$$\begin{cases} H_0 : \theta_i = \theta_z \text{ , se } i \in z \\ H_1 : \theta_i = \theta_0 \text{ , se } i \notin z, \end{cases} \tag{2.14}$$

de forma que a verossimilhança é dada por

$$\begin{aligned} \mathcal{L}(\mathrm{z}, \mathrm{x}, \theta_0, \theta_z) = &\left[\prod_{i\in z} \binom{n_i}{x_i}\right] \theta_z^{\sum_{i\in z} x_i} \left(1-\theta_z\right)^{\sum_{i\in z} n_i - \sum_{i\in z} x_i} \\ &\times \left[\prod_{i\notin z} \binom{n_i}{x_i}\right] \theta_0^{\sum_{i\notin z} x_i} \left(1-\theta_0\right)^{\sum_{i\notin z} n_i - \sum_{i\notin z} x_i}. \end{aligned} \tag{2.15}$$

Aplicando o logaritmo em (2.15), obtém-se

$$\begin{aligned} \ell(\mathrm{z}, \mathrm{x}, \theta_0, \theta_z) = \log(\mathcal{L}) = &\sum_{i\in z} \log \binom{n_i}{x_i} + x_z \log(\theta_z) + (n_z - x_z)\log\left(1-\theta_z\right) \\ &+ \sum_{i\notin z} \log \binom{n_i}{x_i} + x_{\bar{z}} \log(\theta_0) + (n_{\bar{z}} - x_{\bar{z}})\log\left(1-\theta_0\right). \end{aligned} \tag{2.16}$$

Maximizando-se (2.16), tem-se que

$$\frac{\partial \ell}{\partial \theta_0} = 0 \Rightarrow \hat{\theta_0} = \frac{x_{\bar{z}}}{n_{\bar{z}}}, \tag{2.17}$$

$$\frac{\partial \ell}{\partial \theta_z} = 0 \Rightarrow \hat{\theta_z} = \frac{x_z}{n_z}. \tag{2.18}$$

Análogo ao modelo Poisson, a razão de verossimilhança é definida como

$$\begin{aligned}\lambda = \frac{\mathcal{L}}{\mathcal{L}_0} &= \left[\prod_{i\in z}\binom{n_i}{x_i}\right]\theta_z^{\sum_{i\in z}x_i}\left(1-\theta_z\right)^{\sum_{i\in z}n_i-\sum_{i\in z}x_i} \\ &\times\left[\prod_{i\notin z}\binom{n_i}{x_i}\right]\theta_0^{\sum_{i\notin z}x_i}\left(1-\theta_0\right)^{\sum_{i\notin z}n_i-\sum_{i\notin z}x_i}\times\frac{1}{\left[\prod_{i=1}^{m}\binom{n_i}{x_i}\right]\theta^C\left(1-\theta\right)^{N-C}} \\ &= \frac{\theta_z^{x_z}\left(1-\theta_z\right)^{n_z-x_z}\theta_0^{x_{\bar{z}}}\left(1-\theta_0\right)^{n_{\bar{z}}-x_{\bar{z}}}}{\theta^C\left(1-\theta\right)^{N-C}}.\end{aligned} \tag{2.19}$$

2.6 Identificando o Cluster mais verossímil

Após a definição da distribuição mais adequada para modelar o número de casos x_i, o próximo passo é a identificação de um candidato a cluster dentre todas as zonas z existentes no mapa. Para os modelos Binomial e Poisson, tem-se, para uma zona z:

$$\lambda_z = \frac{\sup_{\theta_z>\theta_0}\mathcal{L}(\mathrm{z},\mathrm{x},\theta_0,\theta_z)}{\sup_{\theta_z=\theta_0}\mathcal{L}(\mathrm{z},\mathrm{x},\theta_0,\theta_z)} = \left(\frac{x_z/n_z}{C/N}\right)^{x_z}\times\left(\frac{x_{\bar{z}}/n_{\bar{z}}}{C/N}\right)^{x_{\bar{z}}}\times\times I\left(x_z/n_z > x_{\bar{z}}/n_{\bar{z}}\right) \quad , \tag{2.20}$$

onde $I()$ é a função indicadora.

A estatística Scan é definida por

$$T = \sup_z \lambda_z = \frac{\sup_{\theta_z>\theta_0}\mathcal{L}(\mathrm{z},\mathrm{x},\theta_0,\theta_z)}{\sup_{\theta_z=\theta_0}\mathcal{L}(\mathrm{z},\mathrm{x},\theta_0,\theta_z)}, \tag{2.21}$$

isto é, o cluster mais verossímil será a zona com maior valor de λ_z dentre todas as zonas candidatas a cluster. Como o valor dessa razão tende a crescer muito rápido, pode-se usar o seu logaritmo. Como o logaritmo é estritamente crescente, então o

valor que maximiza λ_z também maximiza $\log(\lambda_z)$.

Para encontrar o cluster mais verossímil, define-se o conjunto de zonas candidatas Z e, para o elemento $z \in Z$, calcula-se λ_z. Uma forma simples e eficiente que é comumente utilizada para definir-se um conjunto Z razoável é através de áreas circulares com diferentes centros e raios (Fernandes, 2015). A obtenção dessas áreas e das respectivas zonas candidatas é descrita nas subseções 2.6.1 e 2.6.2.

2.6.1 Matriz de distâncias

O primeiro passo no processo iterativo que vai varrer o mapa é a obtenção da matriz de distâncias. Para cada uma das m regiões do mapa, considere um centroíde com cordenadas (x_i, y_i), $i = 1, ..., m$. A distância entre duas regiões é dada pela distância entre seus centroídes. Então, para duas regiões i e j quaisquer, a distância Euclidiana é dada por

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{2.22}$$

A Figura 2.2 apresenta um esquema em duas dimensões que ilustra o cálculo dessa distância entre centróides.

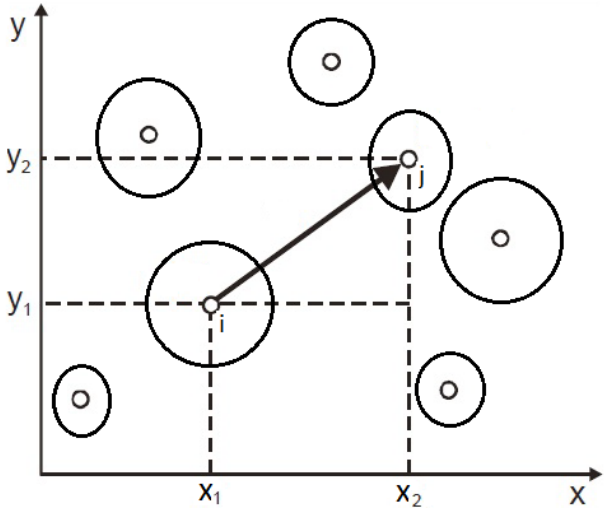


Figura 2.2: Esquema em duas dimensões para cálculo da distância entre centróides. Fonte: Adaptado de Barreto (2011).

A matriz quadrada de distâncias entre os centroídes é simétrica com m linhas e m colunas em que cada elemento da matriz representa a distância entre duas regiões. Sendo assim, a matriz tem a forma

$$D = \begin{bmatrix} 0 & d_{1,2} & \cdots & d_{1,j} & \cdots & d_{1,m} \\ d_{2,1} & 0 & \cdots & d_{2,j} & \cdots & d_{2,m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{i,1} & d_{i,2} & \cdots & 0 & \cdots & d_{i,m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{m,1} & d_{m,2} & \cdots & d_{m,j} & \cdots & 0 \end{bmatrix}. \tag{2.23}$$

O próximo passo será encontrar as zonas candidatas a clusters, calculando a estatística T para cada uma delas.

2.6.2 Identificando candidatos a Clusters

Este processo inicia-se pela construção de zonas, caracterizadas como a aglomeração de uma ou mais regiões próximas. Comumente, são utilizadas áreas circulares para realizar este procedimento. Por exemplo, iniciando-se pela região 1, tem-se o seguinte vetor coluna correspondente às distâncias para as outras regiões

$$D = \begin{bmatrix} 0 \\ d_{2,1} \\ d_{3,1} \\ d_{4,1} \\ \vdots \\ d_{m,1} \end{bmatrix}. \tag{2.24}$$

Seja $d_{(j),i}$ a distância da j-ésima região mais próxima da região i, em que $d_{(n),i} > d_{(n-1),i} > d_{(n-2),i} > \ldots > d_{(3),i} > d_{(2),i}$. O vetor coluna anterior ordenado em ordem crescente de distâncias tem a seguinte forma

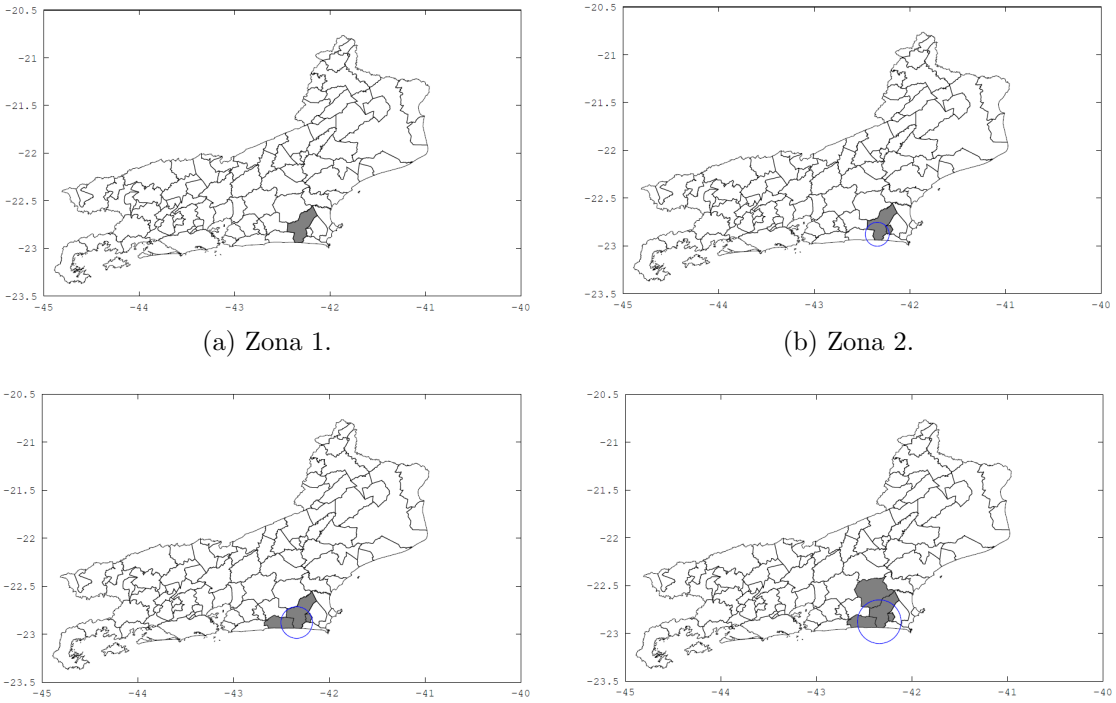
$$D = \begin{bmatrix} 0 \\ d_{(2),1} \\ d_{(3),1} \\ d_{(4),1} \\ \vdots \\ d_{(m),1} \end{bmatrix}. \tag{2.25}$$

A primeira zona selecionada será formada somente pela região 1, ou seja, $z_1 = 1$. Calcula-se, então, o valor λ_{z_1} , conforme 2.21. A segunda zona é formada por 1 e também pela região mais próxima, isto é, a região correspondente à distância $d_{(2),i}$. Esta zona será representada por $z_2 = 1,(2)$. Com os dados da zona z_2, calcula-se λ_{z_2}. O processo iterativo se repete agregando, em cada passo, uma região segundo a distância, até atingir um tamanho máximo de população previamente definido, por exemplo 50% da população total. O cluster mais verossímil será a zona correspondente ao maior valor de λ_{z_i}, obedecendo o limite de populacão máxima, e esta será a estatística T procurada.

A Figura 2.3 mostra um exemplo da construção de zonas por meio de áreas circulares apresentado por Fernandes (2015), utilizando os municípios do estado do Rio de Janeiro.

2.7 Verificação de significância do Cluster

Após a execução do algoritmo para a detecção do cluster mais verossímil, é necessário testar sua significância. Segundo Assunção (2001), a distribuição da estatística T depende da distribuição da população e é virtualmente impossível de ser obtida analiticamente. A aproximação comumente utilizada, com a distribuição quiquadrado da transformação $-2 \log T$, não pode ser feita, uma vez que as condições de regularidade não são atendidas para este caso. A solução apresentada por Kulldorff e Nagarwalla (1995) para contornar essa limitação é obter a distribuição empírica de T e seu *p-valor* via simulação de Monte Carlo. Esse procedimento, introduzido por Dwass (1957), baseia-se na geração de réplicas do mapa original, distribuindo o



(a) Zona 1. (b) Zona 2.

(c) Zona 3. (d) Zona 4.

Figura 2.3: Construção de zonas por áreas circulares.
Fonte: Fernandes (2015)

número total de casos C sob a hipótese nula, de completa aleatoriedade espacial.

Seja J o número de réplicas a serem executadas. A simulação via Monte Carlo segue os seguintes passos:

1. Distribuir C casos aleatoriamente de acordo com uma distribuição Multinomial proporcional à população de cada região.

2. Calcular T conforme 2.21 com o novo banco de dados gerado. Uma vez calculado, armazenar o valor dessa estatística.

3. Repetir os passos 1 e 2 um número J de vezes obtendo a distribuição empírica de T, sob H_0.

4. Rejeitar, com nível de significância de 5%, a hipótese H_0 de ausência de clusters se $T > P_{95}$, onde T é o valor da estatística de teste obtido para os dados reais, e P_{95} é o percentil 95 da distribuição empírica de T sob H_0. A Figura 2.4 apresenta um exemplo de histograma para a distribuição empírica de T obtida

via simulação de Monte de Carlo.

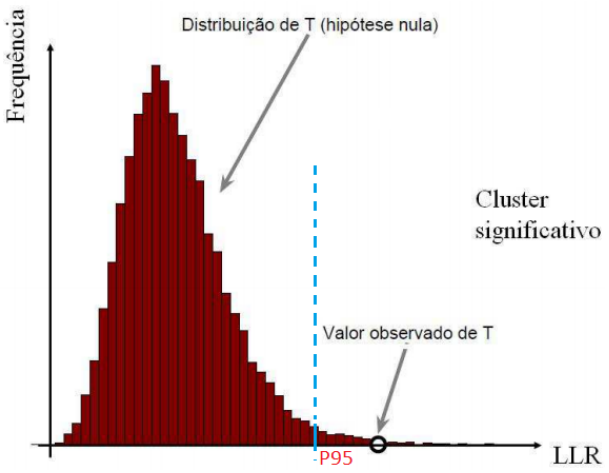


Figura 2.4: Histograma da distribuição empírica de T obtida via simulação de Monte de Carlo.
Fonte: Adaptado de Figueiredo (2010).

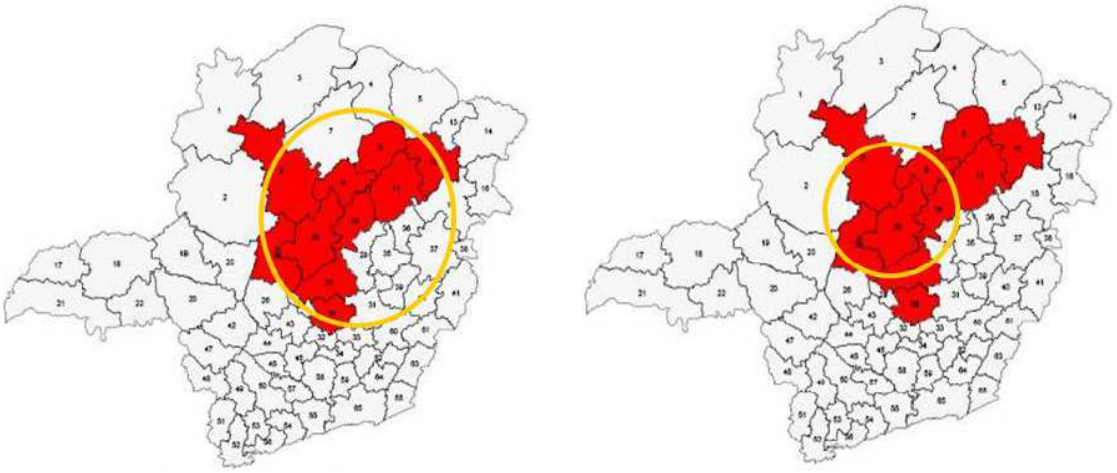
Estudos elaborados por Abrams et al. (2010) revelam que, sob a hipótese nula, a distribuição empírica dos valores da estatística scan de Kulldorff é aproximada pela distribuição Gumbel. Esses resultados não serão aprofundados neste estudo.

2.8 Propriedados da Estatística Scan

A principal vantagem da estatística Scan de Kulldorff, em relação aos demais métodos de detecção de cluster espacial existentes na literatura, é a possibilidade de se realizar tanto a detecção do cluster mais provável quanto o teste de significância através de um só procedimento. Uma das propriedades dessa estatística, que é provada em Kulldorff (1997), é a de que se a hipótese nula for rejeitada, de acordo com a distribuição atual dos pontos, mesmo fixando os pontos presentes dentro do cluster mais provável e aplicando qualquer disposição espacial para os pontos fora desse cluster, a hipótese nula continuará sendo rejeitada (Araújo, 2013). Kulldorff ressalta que, embora isso pareça evidente, muitas estatísticas não têm essa propriedade.

Outra questão de suma importância diz respeito ao poder de teste da estatística Scan. A estatística Scan Espacial de Kulldorff é mais apropriada para detecção de um cluster único bem definido, pois baseia-se no teste uniformemente mais poderoso para

detecção de clusters, como mostra Kulldorff (1997). Para situações em que o mapa apresenta mais de um cluster ou cluster de formato muito irregular, Kulldorff et al. (2003) e Duczmal et al. (2008) indicam que o poder do teste diminui. Essa redução quase sempre está associada a superestimação (situação onde o cluster detectado é maior do que o cluster real), ou subestimação do cluster (situação onde o cluster detectado é menor do que o cluster real), como pode ser visto na Figura 2.5.



(a) Superestimação na detecção do cluster. (b) Subestimação na detecção do cluster.

Figura 2.5: Efeito da redução do poder do teste na detecção do cluster.
Fonte: Figueiredo (2010).

2.9 Resumo do algoritmo Scan

O algoritmo Scan Circular pode ser resumido nos seguintes passos, cada um deles explicado de maneira detalhada nas seções do capítulo 2:

1. Definir os centróides das regiões que compõe o mapa em estudo;
2. Fazer a construção da matriz de distâncias, de acordo com a subseção 2.6.1;
3. Partindo da i-ésima região, ordenar as regiões segundo a distância de forma crescente até a região r_m;
4. Encontrar as zonas candidatas e calcular a estatística T de acordo com a subseção 2.6, registrando a zona com o maior valor de T até o momento e

respeitando o limite máximo de proporção da população que pode estar dentro da zona;

5. Repetir os passos 3 e 4 para todas as regiões do mapa, isto é, para $i = 1, 2, \ldots, m$;

6. Utilizar a simulação de Monte Carlo para avaliar a significância do teste, como descrito na seção 2.7;

7. Rejeitando a hipótese nula no passo 6, deve-se armazenar essa zona, que será considerada um cluster.

O algoritmo Scan Circular dará como resultado o cluster mais verossímil. Esse cluster é classificado como cluster primário. Além do cluster primário, as demais zonas que apresentarem os maiores valores da estatística T também podem ser úteis para a análise do mapa. O segundo maior valor que maximiza a estatística T será classificado como cluster secundário. Da mesma forma, para o terceiro e quarto valores, temos os clusters terciário e quaternário, respectivamente, e assim por diante. Um trabalho que compara esses diversos clusters pode ser visto em Lima (2004).

Capítulo 3

Dados Censurados

O conceito de dados censurados é aplicado em diversas áreas de conhecimento, dentre as quais pode-se destacar a *Análise de Sobrevivência*. A *Análise de Sobrevivência* consiste na reunião de técnicas e métodos para o estudo de dados relacionados ao tempo até a ocorrência de um determinado evento de interesse, ou tempo de falha (Colosimo e Giolo, 2006).

Um problema inerente a esse tipo de estudo relaciona-se ao fato da variável de interesse, tempo de falha, ser temporal e, consequentemente, não ser medida instantaneamente e independentemente do tamanho da resposta. Para casos onde o tempo de falha caracteristicamente ocorre após longo período de tempo, é necessário mais tempo de acompanhamento e persistência para observar o fenômeno de estudo. Em situações extremas, este fato pode comprometer a observação do valor da variável para algumas unidades da amostra, uma vez que o evento de interesse pode não ocorrer até o tempo final do estudo, gerando dessa forma dados censurados.

A censura ocorre quando há perda de informação decorrente da não observação do tempo de falha para alguma unidade da amostra. Em várias situações, a censura acontece por questões de limitação dos equipamentos de medição ou do projeto experimental, resultando em informações parciais ou incompletas sobre o evento de interesse. Segundo Colosimo e Giolo (2006), existem três tipos de censura:

- **Censura à direita**: Ocorre quando existem unidades da amostra que não falharam até o final do estudo, portanto o tempo de falha está à direita do tempo registrado.

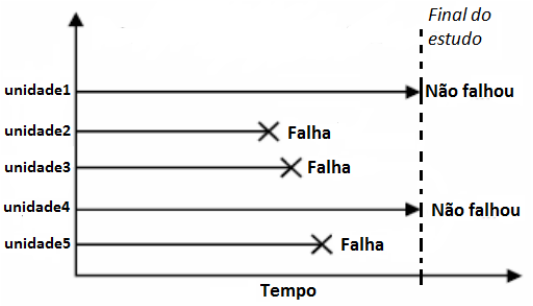


Figura 3.1: Esquema de censura à direita.
Fonte: Adaptado de de Matos e Marazotti (2010).

- **Censura à esquerda**: Neste caso, o tempo registrado para uma determinada unidade é maior do que o tempo de falha. O evento de interesse já ocorreu quando a unidade foi observada, portanto o tempo de falha está à esquerda do tempo registrado.

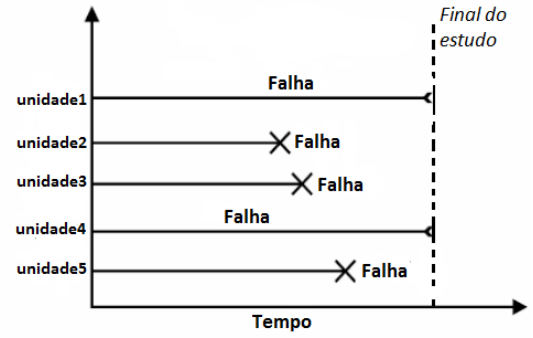


Figura 3.2: Esquema de censura à esquerda.
Fonte: Adaptado de de Matos e Marazotti (2010).

- **Censura intervalar**: Ocorre em estudos em que as unidades experimentais tem acompanhamento periódico. Neste caso, sabe-se apenas que o evento de interesse aconteceu em um dado intervalo de tempo.

Os casos de censura à esquerda e censura intervalar são descritos em maiores detalhes em Lawless (2011). Para dados com censura à direita, Colosimo e Giolo (2006) apresentam três tipos principais: censura do **tipo I**, censura do **tipo II** e a censura **aleatória**. A censura do **tipo I** ocorre quando a unidade é censurada no final do estudo, sendo que este tempo deve ser determinado antes do início do estudo. Na censura do **tipo II**, determina-se antes do início do estudo o número de falhas, ou seja, o número de unidades que irão apresentar o evento de interesse.

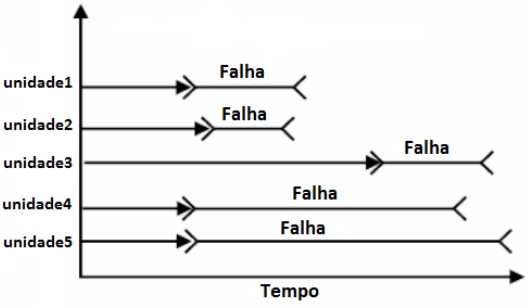


Figura 3.3: Esquema de censura intervalar.
Fonte: Adaptado de de Matos e Marazotti (2010).

A censura **aleatória** ocorre nos casos em que as observações não experimentam o evento de interesse por motivos não controláveis.

3.1 Conceitos básicos

Na presença de observações censuradas na amostra, a verossimilhança é composta tanto pela contribuição da informação do tempo de falha como do tempo de censura. Para essa abordagem, é necessário apresentar algumas definições básicas.

Conforme Colosimo e Giolo (2006), os dados em análise de sobrevivência são representados pela dupla básica (t_i, δ_i), sendo que, $i = 1, \ldots, n$ indica o número de indivíduos ou unidades na amostra, t_i é o tempo de falha ou censura e δ_i é a variável indicadora de falha ou censura, definida por

$$\delta_i = \begin{cases} 1, & \text{se } t_i \text{ é tempo de falha} \\ 0, & \text{se } t_i \text{ é tempo de censura.} \end{cases} \tag{3.1}$$

A variável aleatória não-negativa, geralmente contínua, T, representa o tempo de vida, ou seja, tempo de sobrevivência de um indivíduo ou unidade. Essa variável pode ser representada pela função densidade de probabilidade, $f(t)$; função de sobrevivência, $S(t)$; função risco, $h(t)$; e por relações existentes entre essas três funções.

A função de sobrevivência, denotada por $S(t)$, é definida como a probabilidade de um indivíduo sobreviver a um tempo t, ou seja, a probabilidade de um indivíduo

não falhar até um certo tempo t. Ela é expressa por

$$S(t) = P\left(T \geq t\right) = \int_t^{\infty} f(x)dx, \tag{3.2}$$

sendo que $S(t)$ é uma função monótona decrescente e contínua (Lawless, 2011). Em consequência dessa formulação, a função de distribuição acumulada é definida como a probabilidade de uma observação não sobreviver ao tempo t, isto é, $F(t) = 1-S(t)$.

De acordo com Lawless (2011), a função de risco é definida como o limite da probabilidade de um indivíduo falhar no intervalo de tempo $[t; \Delta t)$, assumindo que este mesmo indivíduo sobreviveu até o tempo t, dividida pelo comprimento do intervalo e é representada por

$$h(t) = \lim_{\Delta t \to 0} \frac{P\left(t \leq T < t + \Delta t | T \geq t\right)}{\Delta t}. \tag{3.3}$$

A função de risco descreve como a probabilidade instantânea de falha (taxa de falha) se modifica com o passar do tempo. É conhecida como taxa de falha instantânea, força de mortalidade ou taxa de mortalidade condicional (Cox e Oakes, 1984). Segundo Colosimo e Giolo (2006), a função de risco é mais informativa do que a função de sobrevivência. Diferentes funções de sobrevivência podem ter formas semelhantes, enquanto que suas respectivas funções de risco podem diferir drasticamente. Dessa forma, a modelagem da função taxa de falha é um importante método para dados de sobrevivência, pois pode apresentar forma crescente, decrescente, constante ou não monótona.

Pode-se definir a função de risco também em termos da função densidade de probabilidade e da função de sobrevivência, isto é

$$h(t) = \frac{f(t)}{S(t)}. \tag{3.4}$$

Após a definição das funções básicas utilizadas em análises que envolvem o tempo de sobrevivência de um indivíduo ou de uma unidade de estudo, podemos introduzir os conceitos aplicados na formulação da função de verossimilhança na presença de dados censurados.

3.2 Função de verossimilhança na presença de dados censurados

De forma geral, supondo uma amostra de observações $t_1, t_2, \ldots, t_n$ de uma certa população de interesse em que todas as observações são não censuradas, onde essa população é caracterizada pela sua função densidade $f(t_i, \boldsymbol{\theta})$, em que $\boldsymbol{\theta}$ é o vetor de parâmetros, temos que a função de verossimilhança é expressa por

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(t_i, \boldsymbol{\theta}). \tag{3.5}$$

Com a presença de censura, ou seja, supondo uma amostra $t_1, t_2, \ldots, t_n$ de tempos de sobrevivência e a variável indicadora δ_i, que segue a definição 3.1, Lawless (2011) define que a contribuição de cada elemento da amostra para a função de verossimilhança é dada por

$$\begin{cases} f(t_i, \boldsymbol{\theta}), \text{ se } t_i \text{ é tempo de falha} \\ S(t_i, \boldsymbol{\theta}), \text{ se } t_i \text{ é tempo de censura,} \end{cases} \tag{3.6}$$

onde $f(t_i, \boldsymbol{\theta})$ e $S(t_i, \boldsymbol{\theta})$ são as funções densidade e de sobrevivência, respectivamente, que caracterizam essa população.

Considerando essa amostra, temos que as observações, quando ordenadas segundo a variável δ_i, podem ser divididas em dois conjuntos:

- as r primeiras observações são as não censuradas;
- e as $n - r$ seguintes são as censuradas.

Dessa forma, Lawless (2011) apresenta a formulação da função de verossimilhança considerando os três tipos de censura à direita, descritas como:

1. **Censurado do Tipo I**: Neste caso, tem-se r falhas e $n-r$ censuras observadas no término do experimento. Desta maneira, a função de verossimilhança é expressa por

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{r} f(t_i, \boldsymbol{\theta}) \prod_{i=r+1}^{n} S(t_i, \boldsymbol{\theta}). \tag{3.7}$$

2. **Censura do Tipo II**: Neste caso, r é fixo, de modo que os r menores tempos são observados, assim como as $n - r$ censuras. Aplicando o resultado das estatísticas de ordem, temos que a função de sobrevivência é expressa por

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{n!}{(n-r)!} \prod_{i=1}^{r} f(t_i, \boldsymbol{\theta}) \prod_{i=r+1}^{n} S(t_i, \boldsymbol{\theta}). \tag{3.8}$$

   Observa-se que $\frac{n!}{(n-r)!}$ é uma constante e, desse modo, pode ser desprezada, uma vez que não envolve qualquer parâmetro de interesse. Assim, a função de verossimilhança pode ser expressa proporcionalmente como

$$\mathcal{L}(\boldsymbol{\theta}) \propto \prod_{i=1}^{r} f(t_i, \boldsymbol{\theta}) \prod_{i=r+1}^{n} S(t_i, \boldsymbol{\theta}). \tag{3.9}$$

3. **Censura do Tipo Aleatória**: Neste caso, temos a definição das variáveis aleatória T, tempo de falha, e C, tempo de censura, que são independentes entre si. Para $i = 1, \ldots, , n$ temos que as observações da amostra são representadas pelo par (t_i, δ_i), em que $t_i = min\,(T_i, C_i)$, e $\delta_i = 1$, se $T_i \leq C_i$, ou $\delta_i = 0$, se $T_i > C_i$.

   Levando em conta que T e C são independentes, e supondo que T_i é caracterizada pela sua função densidade $f(t_i, \boldsymbol{\theta})$ e pela sua função de sobrevivência $S(t_i, \boldsymbol{\theta})$, assim como C_i, que é caracterizada pela sua função densidade $g(t_i)$ e pela sua função de sobrevivência $G(t_i)$, temos que:

   - Se para o $i-$ésimo indivíduo for observada uma censura, tem-se $P\,[t_i = t, \delta_i = 0] = P\,[C_i = t, T_i > C_i] = P\,[C_i = t, T_i > t] = P\,[C_i = t]\,P\,[T_i > t] = g(t)S(t, \boldsymbol{\theta})$.
   - Se para o $i-$ésimo indivíduo for observada uma falha, tem-se $P\,[t_i = t, \delta_i = 1] = P\,[T_i = t, T_i \leq C_i] = P\,[T_i = t, C_i \geq t] = P\,[T_i = t]\,P\,[C_i > t] = f(t, \boldsymbol{\theta})G(t)$.

Dessa forma, a função de sobrevivência na presença de censura do tipo aleatória é expressa por

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{r} f(t_i, \boldsymbol{\theta}) G(t_i) \prod_{i=r+1}^{n} g(t_i) S(t_i, \boldsymbol{\theta}). \tag{3.10}$$

Partindo da suposição de que o mecanismo de censura é não informativo, ou seja, não carrega informações sobre os parâmetros, os termos $G(t_i)$ e $g(t_i)$ podem ser desprezados e, assim, a função de verossimilhança pode ser expressa proporcionalmente como

$$\mathcal{L}(\boldsymbol{\theta}) \propto \prod_{i=1}^{r} f(t_i, \boldsymbol{\theta}) \prod_{i=r+1}^{n} S(t_i, \boldsymbol{\theta}). \tag{3.11}$$

Portanto, considerando todos os mecanismos de censura à direita, a verossimilhança é dada pela fórmula geral

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{n} \left[f(t_i, \boldsymbol{\theta})\right]^{\delta_i} \left[S(t_i, \boldsymbol{\theta})\right]^{1-\delta_i} = \prod_{i=1}^{n} \left[h(t_i, \boldsymbol{\theta})\right]^{\delta_i} \left[S(t_i, \boldsymbol{\theta})\right], \tag{3.12}$$

em que δ_i é a variável indicadora de falha.

Os estimadores de máxima verossimilhança serão os valores de $\boldsymbol{\theta}$ que maximizam $\mathcal{L}(\boldsymbol{\theta})$ ou equivalentemente o seu logaritmo, ou seja

$$\log(\mathcal{L}(\boldsymbol{\theta})) = l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \{\delta_i \log[f(t_i, \boldsymbol{\theta})] + (1 - \delta_i) \log[S(t_i, \boldsymbol{\theta})]\}. \tag{3.13}$$

Os estimadores são encontrados resolvendo-se o sistema de equações

$$U(\boldsymbol{\theta}) = \frac{\partial \log(\mathcal{L}(\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}}. \tag{3.14}$$

As formulações apresentadas por Lawless (2011) para estudos com o tempo de sobrevivência serão utilizadas neste trabalho como modelos para a elaboração da extensão da estatística Scan, uma vez que são métodos que já conseguem incorporar a informação censurada de maneira explícita no cálculo da verossimilhança. De certa forma, Huang et al. (2007) já propuseram uma extensão que busca incorpo-

rar as técnicas da *Análise deSobrevivência* no processo de identificação de clusters, entretanto, como já exposto anteriormente, a aplicação dessa extensão é voltada para dados contínuos. Neste estudo, faremos uma extensão voltada para dados de contagem com censura.

Capítulo 4

Estimação via Métodos de otimização

Segundo Ruggiero e Lopes (1997), para algumas equações, como as equações polinomiais de segundo grau, existem fórmulas explícitas que permitem a identificação das raízes em função dos coeficientes. No entanto, no caso de polinômios de grau mais alto e nos casos de funções que não possuem forma simples, é praticamente impossível se achar exatamente os zeros da função. Da mesma forma, a estimação de parâmetros em modelos que não possuem forma fechada ou que não sejam facilmente diferenciáveis, torna-se intratável do ponto de vista dos métodos diretos de estimação, como a estimação por máxima verossimilhança. Nesse sentido, métodos de otimização apresentam-se como opções viáveis no processo de estimação para esses casos.

Segundo Saramago e Steffen Jr (2008), otimização consiste em encontrar uma solução ou um conjunto de soluções ótimas para uma determinada função ou conjunto de funções. O conceito de solução ótima é inerente ao problema que se deseja otimizar. Por exemplo, em uma situação A, representada por uma única função F_A, há a necessidade de se determinar um valor (valor ótimo) tal que F_A seja mínimo, ou ainda, uma situação B, cujo modelo seja expresso por n funções F_{B_n} $(n = 1, 2, 3, \ldots, n)$, em que, pretende-se maximizar apenas algumas funções e minimizar as demais. Nestes casos, o problema pode apresentar uma única solução, um conjunto de soluções ou ainda não haver solução que satisfaça todas as funções. À medida que o número de funções e o número de variáveis aumenta, a dificuldade em se determinar o conjunto de soluções ótimas também aumenta.

4.1 Formulação do problema de otimização

A forma genérica dos problemas de otimização busca minimizar $f(x)$ sujeito a $X \in S$, em que $f : \mathbb{R}^n \to \mathbb{R}$ e $S \subset \mathbb{R}^n$. Nesse contexto, S é chamado de *conjunto factível*. O vetor $X = \{x_1, x_2, \ldots, x_n\}$ é composto pelas variáveis do conjunto analisado.

Segundo Saramago e Steffen Jr (2008), consideram-se dois tipos de soluções deste problema:

Definição 4.1.1. *"Um ponto $X^* \in S$ é um minimizador local de f em S se e somente se existe $\varepsilon > 0$ tal que $f(X) \geq f(X^*)$ para todo $X \in S$ tal que $\| X - X^* \| < \varepsilon$. Se $f(X) > f(X^*)$ para todo $X \in S$ tal que $X \neq X^*$ e $\| X - X^* \| < \varepsilon$, considera-se que se trata de um minimizador local estrito em S."*

Definição 4.1.2. *"Um ponto $X^* \in S$ é um minimizador global de f em S se e somente se $f(X) \geq f(X^*)$ para todo $X \in S$. Se $f(X) > f(X^*)$ para todo $X \in S$ tal que $X \neq X^*$, considera-se que se trata de um minimizador global estrito em S."*

De maneira análoga, defini-se maxizadores locais e globais. Observa-se que "Maximizar f" é equivalente a "Minimizar -f", razão pelo qual se pode, sem perda de generalidade, considerar apenas o termo "Minimização". O Teorema de Bolzano-Weierstrass apresenta um resultado fundamental relacionado com o problema de otimização:

Teorema 4.1.1. *"Uma função real contínua f, definida em um conjunto fechado e limitado $S \subset \mathbb{R}^n$, admite um minimizador global em S."*

Dentre os diversos métodos para tratar o problema de minimização, os mais aplicados são baseados no cálculo de derivadas de $f(x)$. Entre esses métodos, podemos citar o método de Newton e os métodos Quase-Newton. Segundo Pedroso e Diniz-Ehrhardt (2005), apesar da indiscutível eficiência de tais algoritmos, muitas aplicações de otimização trabalham com métodos que não usam gradientes ou mesmo aproximaçõe locais para a função f, conhecidos como métodos de *busca direta*. O termo *busca direta* refere-se a um método iterativo em que um conjunto de pontos

é testado a cada iteração, associado a uma estratégia que usa somente avaliações da função f para definir a aproximação seguinte para o minimizador.

Considerando todos os métodos que não usam derivadas, o desenvolvido por Nelder e Mead (1965) está entre os mais utilizados. Essa grande popularidade reside nos bons resultados práticos, especialmente se levada em conta a simplicidade do algoritmo. A partir de um simplex, que será definido mais adiante, no $\mathbb{R}^n$, a idéia do método é que, a cada iteração, substitui-se o vértice com o valor menos desejado da função objetivo ou, quando isso não for possível, reduzir as dimensões do simplex.

Já para problemas de minimização de funções unidimensionais, o método proposto por Brent (1973) destaca-se como um dos mais utilizados pela sua combinação simples de diversos métodos computacionais para busca de raízes de função sem a utilização de derivadas. Este método, embora não apresente nenhuma novidade em termos matematicamente formais, é um algoritmo bastante usado e está presente em diversas bibliotecas e softwares como função chave na aproximação de raízes.

Neste trabalho, faremos uso dos métodos de Brent e de Nelder-Mead no desenvolvimento da extensão proposta para a estatística Scan circular, sendo estes apresentados com mais detalhes nas seções 4.2 e 4.3 a seguir.

4.2 Método de Brent

O Método de Brent é um algoritmo para minimização unidimensional sem a utilização de derivadas, de modo que busca solucionar o problema $f(x) = 0$, onde $f(x) : \mathbb{R} \to \mathbb{R}$. O método é uma combinação simples de diversos métodos computacionais para busca de raízes de função. Os métodos usados no algoritmo de Brent são: Método da Bissecção, da Secante e Interpolação Quadrática Inversa.

A ideia por trás deste algoritmo está em aproveitar as vantagens de cada uma das técnicas combinadas, procurando minimizar suas desvantagens. Desta forma, utiliza a Interpolação Quadrática Inversa sempre que as aproximações intermediárias não se sobrepõem (o que acarretaria em uma divisão por zero). Quando há a sobreposição, tenta fazer a próxima aproximação utilizando o Método da Secante. Todavia, estas duas últimas técnicas são suscetíveis à divergir ao longo da execução.

Caso este comportamento seja detectado, o algoritmo "ativa" o Método da Bissecção a fim de reconverter a aproximação ao zero da função. Nesta seção faremos uma exposição resumida sobre o funcionamente do algoritmo. Maiores detalhes acerca da formulação do método podem ser vistos em Brent (1973).

Uma visão geral do método pode ser descrito da seguinte forma:

- O método se inicia com:
  - uma tolerância de parada $\delta > 0$;
  - pontos **a** e **b** de tal modo que $f(a)f(b) < 0$

  Se necessário, **a** e **b** são trocados de modo que $|f(b)| \leq |f(a)|$; Dessa forma, o ponto **b** é considerado como a melhor solução aproximada. Um terceiro ponto **c** é inicializado, definido como **c**=**a**.

- Em cada iteração, o método mantém **a**, **b** e **c**, de tal modo que **b**$\neq$**c** e
  (a) $f(b)f(c) < 0$, de modo que uma solução se encontra entre **b** e **c**, caso f seja contínuo;
  (b) $|f(b)| \leq |f(c)|$ de modo que **b** pode ser considerado como a melhor solução aproximada na iteração atual;
  (c) **a** é diferente de **b** e **c**, ou **a**=**c**, nesse caso representando o valor imediatamente passado de **b**.

Cada iteração ocorre segundo os passo abaixo:

1. Se $|b - c| < \delta$, então o método retorna o ponto **b** como a raíz aproximada.
2. Caso contrário, o método determina o cálculo de um valor $\hat{\boldsymbol{b}}$ da seguinte forma:
   (a) Se **a**=**c**, então $\hat{\boldsymbol{b}}$ é determinado pelo método da Secante: $\hat{\boldsymbol{b}} = \frac{af(b)-bf(a)}{f(b)-f(a)}$.
   (b) Caso contrário, **a**, **b** e **c** são distintos, e $\hat{\boldsymbol{b}}$ é determinado através do método de Interpolação Quadrática Inversa:
     - Determinar α, β e γ de tal forma que $p(y) = \alpha y^2 + \beta y + \gamma$ satisfaça $p(f(a)) = a$, $p(f(b)) = b$ e $p(f(c)) = c$.

- Definir $\hat{\boldsymbol{b}} = \gamma$.

3. Se necessário, $\hat{\boldsymbol{b}}$ é ajustado ou substituído por um ponto estimado pelo método da Bissecção.

4. Uma vez que $\hat{\boldsymbol{b}}$ é finalizado, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ e $\hat{\boldsymbol{b}}$ são utilizados para determinar novos valores de $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$.

Sumarizando, o método de Brent apresenta as seguintes vantagens: tende a convergir rapidamente, não necessita de derivadas e sempre converge, caso a raiz esteja delimitada pelo intervalo necessário ao método da Bissecção. Por outro lado, possui a desvantagem de exigir que um intervalo contendo a raiz seja fornecido, o que reduz a generalidade da aplicação do método em alguns casos.

4.3 Método de Nelder-Mead

O método de Nelder-Mead, publicado em Nelder e Mead (1965), é provavelmente o método mais utilizado de busca direta. Sua formulação é baseada na construção e manipulação de um simplex, o qual é definido como:

Definição 4.3.1. ***Simplex*** *é um conjunto de vetores (pontos) em um espaço M-dimensional.*

A contribuição do algoritmo Nelder-Mead à classe dos métodos simplex reside na possibilidade de contração ou expansão do simplex, além da reflexão, como mostra a Figura 4.1.

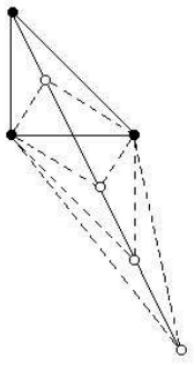


Figura 4.1: Possibilidades de atualização do simplex no método de Nelder−Mead. Fonte: Pedroso e Diniz-Ehrhardt (2005).

São necessários quatro coeficientes escalares no algoritmo de Nelder-Mead. São eles:

- Coeficiente de reflexão: $\rho > 0$;
- Coeficiente de expansão: $\chi > 1$ com $\chi > \rho$;
- Coeficiente de contração: $0 < \gamma < 1$ e
- Coeficiente de redução: $0 < \sigma < 1$.

Para uma função $f(x) : \mathbb{R}^n \to \mathbb{R}$ e um simplex no $\mathbb{R}^n$, a iteração de Nelder-Mead segue o seguinte algoritmo:

1. **Ordenação**: Ordenar os vértices do simplex de maneira que $f(x_1) \leq f(x_2) \leq \ldots \leq f(x_{n+1})$. Calcular o centróide dos n melhores pontos, $\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$. Este ponto é usado como base para definir as demais operações do método de Nelder-Mead. As seguintes operações modificam a forma do simplex adaptando-o às características da função. Cada operação visa gerar o novo vértice do simplex, que substituirá o pior vértice.

2. **Reflexão**: Calcular o ponto de reflexão $x_r = (1 + \rho)\bar{x} - \rho x_{n+1}$. Se $f(x_1) \leq f(x_r) < f(x_n)$, aceitar o ponto x_r e terminar a iteração.

3. **Expansão**: Se $f(x_r) < f(x_1)$, calcular o ponto de expansão $x_e = (1 + \rho\chi)\bar{x} - \rho\chi x_{n+1}$. Se $f(x_e) < f(x_r)$, aceitar x_e e terminar a iteração, caso contrário $(f(x_e) \geq f(x_r))$ aceitar x_r e terminar a iteração.

4. **Contração**: Se $f(x_r) \geq f(x_n)$, fazer uma contração:

   (a) **Contração externa**: Se $f(x_n) \leq f(x_r) < f(x_{n+1})$, calcular $x_c = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1}$. Se $f(x_c) \leq f(x_r)$, aceitar x_c e terminar a iteração. Caso contrário, ir para o passo 5.

   (b) **Contração interna**: Se $f(x_r) \geq f(x_{n+1})$, calcular $x_c = (1-\gamma)\bar{x}+\gamma x_{n+1}$. Se $f(x_c) < f(x_{n+1})$, aceitar x_c e terminar a iteração. Caso contrário, ir para o passo 5.

5. **Redução**: Calcular os vetores $v_i = x_1 + \sigma(x_i - x_1)$, $i = 2, \ldots, n+1$. Os vértices (ainda fora de ordem) para a próxima iteração são $x_1, v_2, \ldots, v_{n+1}$.

No processo de se ordenar os vértices do simplex no início da iteração, podem ocorrer empates no valor da função em dois ou mais pontos. No artigo original de Nelder e Mead (1965), não há menção de como proceder em tal situação. Entretanto, segundo Pedroso e Diniz-Ehrhardt (2005), uma regra de desempate é sugerida em Lagarias et al. (1998):

- **Iteração sem redução do simplex**: Quando não há redução do simplex, apenas o vértice de índice $n + 1$ é descartado e substituído por um vetor v. O vetor v toma a posição $j + 1$ no simplex da iteração seguinte, onde $j = max_{0 \leq m \leq n}\{m | f(v) < f(x_{m+1}|)\}$. Os outros vértices continuam na ordem que foi estabelecida antes de se iniciar a iteração.

- **Iteração com redução do simplex**: Nesse caso, apenas o vértice x_1 é aproveitado na iteração seguinte. Somente uma regra de desempate é necessária, que ocorre quando o vértice x_1 empata com um ou mais vetores como sendo o ponto de melhor valor da função no novo simplex. Caso isso ocorra, defini-se $x_1^{k+1} = x_i^k$. Em qualquer outra situação, pode-se fazer uso de qualquer outra regra de desempate depois de uma redução.

Apesar de ser largamente empregado e poder ser aplicado em qualquer função $f(x)$, que não precisa ser contínua ou diferenciável, o método de Nelder-Mead não tem garantias de convergência para dimensões superiores a 2. Outro aspecto negativo

é a razão de convergência lenta, já que o método não usa derivadas e apenas busca o mapeamento da topografia da função $f(x)$. Por outro lado, a aproximação inicial do algoritmo não precisa estar próximo do ponto de mínimo e sua convergência é garantida para dimensões menores ou iguais a 2 para pelo menos um mínimo local.

Capítulo 5

Estatística de Varredura Espacial para Dados de Contagem com Censura

Diversas pesquisas que envolvem informações georreferenciadas apresentam dados caracteristicamente discretos, como estudos epidemiológicos, estudos de fraudes bancárias, registros de crimes, entre outros. Muitas vezes, limitações dos métodos de mensuração ou erros no processo de coleta dos dados criam situações em que não se conhece o valor exato da unidade observada, por exemplo, quando registra-se apenas um limite inferior, nesse caso o valor real é menor do que o registrado, ou um limite superior, nesse caso o valor real é maior do que o registrado, caracterizando assim a presença de censura à esquerda e censura à direita, respectivamente. A aplicação da estatística Scan de Kulldorff em dados censurados já foi proposta por Huang et al. (2007). Entretanto, a formulação apresentada é voltada para dados contínuos, uma vez que a distribuição utilizada na modelagem do fenômeno analisado foi a exponencial.

Neste estudo, apresentamos uma extensão da estatística Scan de Kulldorff para dados de contagem com presença de censura através de uma adaptação na fórmula da verossimilhança, de modo que a informação da censura foi incorporada no cálculo da estatística Scan e, consequentemente, na elaboração do teste de hipótese. O modelo de distribuição utilizado nesta aplicação foi o Binomial. Para avaliar o método apresentado e comparar seu desempenho com a técnica tradicionional do Scan circular, fizemos a implementação dessas duas metodologias em linguagem R

(R Core Team, 2015). Esses dois algoritmos foram aplicados em dois conjuntos de cenários artificiais simulados, um deles sem a presença de censura, para calibração e verificação do algoritmo implementado para a técnica Scan usual, e outro com a presença de censura, para comparação entre a performance do método proposto e da técnica tradicional do Scan circular na conjuntura de interesse do estudo. Por fim, esses dois algoritmos foram aplicados em um banco de dados real de homicídios nos municípios do Rio de Janeiro para o ano de 2014, onde casos de censura foram gerados de forma artificial. Todos esses procedimentos serão detalhados ao longo deste capítulo.

5.1 Extensão da estatística Scan

A extensão da estatística Scan para dados censurados de contagem tem como objetivo principal incorporar a informação da censura no processo de estimação da estatística do teste e possibilitar a utilização do dado censurado respeitando sua real característica. Para formulação dessa extensão, partimos inicialmente do pressuposto de que as regiões com presença de informação censurada eram conhecidas previamente. Essa informação foi incorporada na representação do mapa através da variável indicadora δ_i, identificando quais regiões apresentam dados censurados ou não, segundo a regra

$$\delta_i = \begin{cases} 0, \text{ se a região } i \text{ não possui informação censurada} \\ 1, \text{ se a região } i \text{ possui informação censurada.} \end{cases} \tag{5.1}$$

A representação do mapa com a presença dessa variável indicadora δ_i é exemplificada na Figura 5.1, que mostra as regiões administrativas do Distrito Federal em relação a uma determinada variável de interesse que apresenta regiões onde a informação observada é censurada.

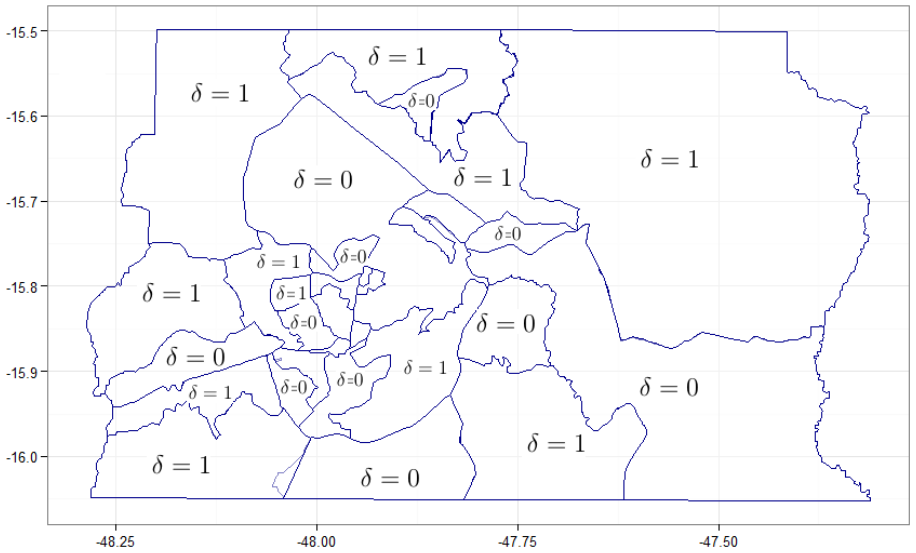


Figura 5.1: Esquema de censura na representação do mapa do DF.
Fonte: Elaborado pelo autor.

Com a identificação das regiões com presença de censura, adaptamos a formulação da verossimilhança seguindo as idéias básicas da área de *Análise de sobrevivência*, apresentadas no capítulo 3. Nos modelos usuais da estatística Scan de Kulldorff, apresentados nas seções 2.4 e 2.5, a verossimilhança é determinada de forma geral, sob H_0, através de

$$\mathcal{L}_0(\mathrm{x}, \theta_0) = \prod_{i=1}^{m} P(\mathrm{X}_i = \mathrm{x}_i|\theta_0) \tag{5.2}$$

e sob H_1

$$\mathcal{L}(\mathrm{z}, \mathrm{x}, \theta_0, \theta_z) = \prod_{i \in z} P(\mathrm{X}_i = \mathrm{x}_i|\theta_z) \times \prod_{i \notin z} P(\mathrm{X}_i = \mathrm{x}_i|\theta_0). \tag{5.3}$$

Considerando a presença de censura à direita, o ajuste realizado na formulação da verossimilhança é análogo a 3.12, isto é, sob H_0

$$\mathcal{L}_0(\mathrm{x}, \theta_0) = \prod_{i=1}^{m} \left[P(\mathrm{X}_i = \mathrm{x}_i|\theta_0)\right]^{1-\delta_i} \times \left[P(\mathrm{X}_i \geq \mathrm{x}_i|\theta_0)\right]^{\delta_i} \tag{5.4}$$

e sob H_1

$$\begin{aligned} \mathcal{L}(\mathrm{z}, \mathrm{x}, \theta_0, \theta_z) = &\prod_{i \in z} \left[P(\mathrm{X}_i = \mathrm{x}_i|\theta_z)\right]^{1-\delta_i} \times \left[P(\mathrm{X}_i \geq \mathrm{x}_i|\theta_z)\right]^{\delta_i} \\ &\times \prod_{i \notin z} \left[P(\mathrm{X}_i = \mathrm{x}_i|\theta_0)\right]^{1-\delta_i} \times \left[P(\mathrm{X}_i \geq \mathrm{x}_i|\theta_0)\right]^{\delta_i}, \end{aligned} \tag{5.5}$$

onde δ_i indica quais regiões possuem dados censurados ou não.

O seguinte passo foi estimar θ_0 e θ_z para obter o valor que maximiza a razão de verossimilhança λ_z. Ao contrário do que ocorre no método clássico, onde λ_z pode ser estimado de maneira análiticа, tanto 5.4 como 5.5 não possuem forma fechada, e trabalhar com suas derivadas não é trivial, o que dificulta a estimação de θ_0 e θ_z e, consequentemente, da razão de verossimilhança. Dessa forma, a estimação desses parâmetros foi realizada através de métodos de otimização. Como a verossimilhança apresentada em 5.4 é uma função unidimensional, aplicamos o método de Brent (Brent, 1973) para obter $\hat{\theta_0}$. Já para a verossimilhança em 5.5, aplicamos o método de Nelder-Mead (Nelder e Mead, 1965) para obter $\hat{\theta_0}$ e $\hat{\theta_z}$, já que este método é mais indicado para problemas multidimensionais. Com a obtenção desses estimadores, calculamos a estatística Scan segundo 2.21, e, dessa forma, identificamos o cluster mais verossímel.

A seguinte adaptação foi realizada na simulação via Monte Carlo para avaliar a significância do cluster mais verossímel, mais especificamente no processo de geração de réplicas do mapa original para obter a distribuição empírica de T. Na estatística Scan tradicional, cada uma das J réplicas é obtida distribuindo C casos aleatoriamente de acordo com uma distribuição Multinomial proporcional à população de cada região. Entretanto, na presença de regiões com informação censurada, o valor real do total de casos no mapa não é conhecido. Outro aspecto que precisava ser considerado é que as regiões que possuem informação censurada devem ser as mesmas em cada uma das réplicas geradas, de forma que os valores de δ_i em cada simulação sejam iguais aos observados no mapa original. Diante disso, considerando um mapa com k regiões censuradas, onde $C_k = \sum_{i=1}^{m} x_i\delta_i$ é o número de casos registrados nas regiões censuradas, e J o número de réplicas a serem executadas, adaptamos a simulação via Monte Carlo na implementação da extensão proposta através dos seguintes passos:

1. Separamos as k regiões censuradas do mapa, de modo que as $m-k$ regiões que não apresentam censura formam um subconjunto do mapa original, composto apenas por regiões sem censura.

2. Distribuímos $C - C_k$ casos aleatoriamente de acordo com uma distribuição Multinomial proporcional à população de cada região que não apresenta censura, desconsiderando as k regiões censuradas.

3. Ao mapa simulado, juntamos novamente as k regiões censuradas, que apresentam o mesmo número C_k de casos registrados no mapa original, de modo que o total de casos no mapa simulado seja C.

4. Calculamos T conforme 2.21 e seguindo as adaptações 5.4 e 5.5 com o novo banco de dados gerado. Uma vez calculado, armazenamos o valor dessa estatística.

5. Repetimos os passos de 1 a 4 um número J de vezes obtendo a distribuição empírica de T, sob H_0.

6. Rejeitamos, com nível de significância de 5%, a hipótese H_0 de ausência de clusters se $T > P_{95}$, onde T é o valor da estatística de teste obtido para os dados reais com censura, e P_{95} é o percentil 95 da distribuição empírica de T sob H_0.

A Figura 5.2 ilustra o processo de simulação do mapa original explicado nos itens acima.

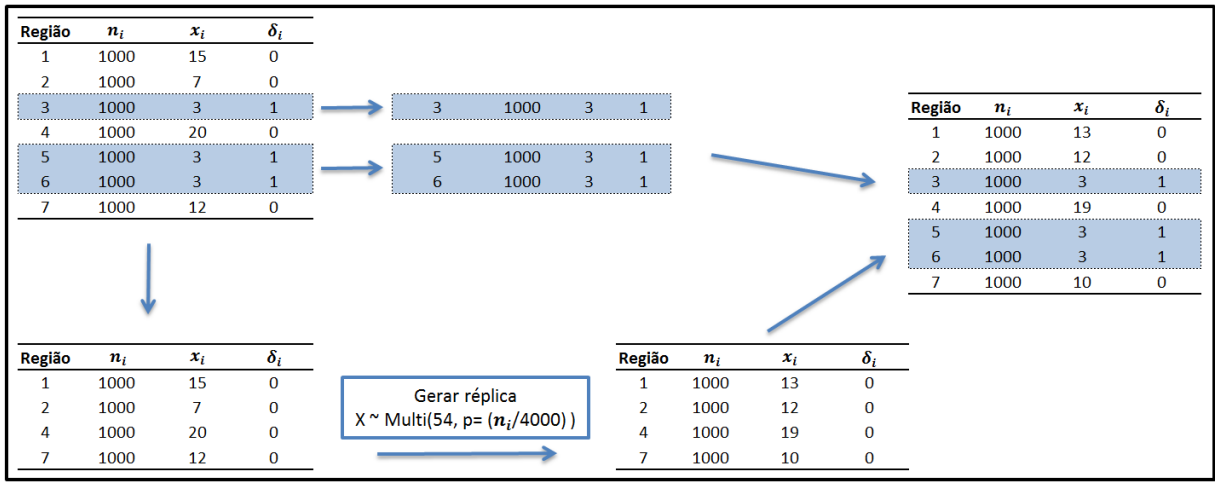


Figura 5.2: Processo de simulação em cada réplica do mapa original.
Fonte: Elaborado pelo autor.

Com essas modificações, adequamos a estatística Scan para tratar a informação censurada tanto no processo de identificação do cluster mais verossímil como no processo de avaliação de significância desse cluster. Os demais procedimentos do algoritmo Scan, como cálculo da matriz de distância e identificação dos candidatos a cluster através de janelas circulares foram realizados da mesma forma como no método Scan tradicional apresentado no capítulo 2.

5.2 Implementação computacional

Com o intuito de comparar o desempenho do método proposto com a técnica tradicional do Scan circular, fizemos a implementação computacional dos dois algoritmos em linguagem R, utilizando técnicas computacionais descritas em Chambers (2008). A implementação do método usual da estatística Scan seguiu o algoritmo já estabelecido na literatura e que foi apresentado com detalhes no capítulo 2. Já para a implementação da extensão proposta, utilizamos o algoritmo implementado para o método clássico como base e realizamos as adaptações apresentadas na seção 5.1. Nos dois casos, trabalhamos com a distribuição Binomial para modelar o número de casos. Na implementação, consideramos novamente $N = \sum_{i=1}^{m} n_i$ e $C = \sum_{i=1}^{m} x_i$, como a população total e o número total de casos estudado no mapa, respectivamente.

Primeiramente, fizemos a implementação do método tradicional da estatística Scan de Kulldorff, identificado a partir deste momento como *Scan-Binomial*. Baseado no resumo apresentado na seção 2.9, o algoritmo foi implementado da seguinte forma:

1. Construímos uma matriz com as informações de população (*pop*), número de casos (*cases*), longitude e latitude do centróide (x e y). Uma variável ID também é criada para identificação de cada região do mapa, seguindo a ordem de entrada dos dados. Neste passo, algumas variáveis que serão utilizadas ao longo do processo são calculadas

```
coords <- cbind(x,y)
id <- seq(1:nrow(coords))
```

```
data <- cbind(id,pop,cases,coords)

N <- sum(data[,2])
C <- sum(data[,3])
theta <- C/N
n_regions <- nrow(data)
```

2. Calculamos a matriz de distâncias conforme 2.6.1.

```
dist_matrix <- as.matrix(dist(coords , diag =TRUE , upper =TRUE ))
dist_matrix <- cbind(dist_matrix,id)
```

3. Fizemos a construção das zonas com base na distância entre os centróides das regiões do mapa. Após a construção das zonas, identificamos quais são candidatas a cluster. Foram consideradas candidatas aquelas zonas para as quais $\sum_{i \in Z} x_i > \frac{\sum_{i \in Z} n_i C}{N}$ e $\sum_{i \in Z} n_i < n_{max}$, isto é, o número observado de casos é maior que o esperado e a população da zona é menor que um valor previamente estipulado. Nesse caso, foi utilizado $n_{max} = 0,25N$. Este passo é realizado através de um processo iterativo que apresenta número de iterações igual ao número de regiões no mapa.

4. Calculamos a estatística razão de verossimilhança (λ_z) para as zonas candidatas a cluster segundo a equação 2.19, apresentada na seção 2.5. Este passo é realizado através de um processo iterativo aninhado ao passo 3, ou seja, ele repete em cada iteração do passo 3. O número de iterações será igual ao número de zonas candidatas a cluster, identificadas no passo anterior.

```
zones <- lapply(1:n_regions,
 function(z){
  dist_matrix <- dist_matrix[order(dist_matrix[,z]),]
  valid_zones <- sapply(1:n_regions, function(j){
  n_z <- sum( data[ dist_matrix[1:j,n_regions+1], 2] )
  return(n_z < 0.25*N) })

  partial_zones <- lapply(1:sum(valid_zones),
   function(y){
        n_z <- sum( data[ dist_matrix[1:y,n_regions+1],2] )
        x_z <- sum( data[ dist_matrix[1:y,n_regions+1],3] )
    x_z_bar <- C - x_z
        n_z_bar <- N - n_z
```

```
        theta_0 <- x_z_bar/n_z_bar
        theta_1 <- x_z/n_z
        ifelse(x_z > n_z*(C/N),
         llr <- x_z*log(theta_1)+(n_z - x_z)*log(1- theta_1)+
         x_z_bar*log(theta_0)+(n_z_bar - x_z_bar)*log(1- theta_0)-
         C*log(theta)-(N-C)*log(1 - theta),
         llr <- 0)
        return(data.frame(zone= paste(sort(as.numeric(dist_matrix[1:y,n_regions+1])),
             collapse=", "),
        llr=round(llr,5),
        nzone=n_z,
        xzone=x_z,
        expected_cases=n_z*C/N))
        }
  )
 partial_zones <- rbind.fill(partial_zones)
 return(partial_zones)
 })
```

5. Identificamos o cluster mais verossímil ordenando todos os valores de λ_z obtidos no nos passos 3 e 4.

6. Para fazer a verificação da significância do cluster encontrado, geramos *nsim* réplicas do mapa original, conforme apresentado na seção 2.7. A quantidade de réplicas *nsim* é definida pelo usuário.

```
sim_cases <- rmultinom(nsim, size = C, prob = data[,2]/N)
sim_cases <- cbind(id,sim_cases)
```

7. Repetimos os passos 4 e 5 para cada uma das réplicas geradas no passo 6, de forma a obter a distribuição empírica de T sob H_0. Não é necessário repetir o passo 3, uma vez que as zonas candidatas a cluster para os dados simulados serão as mesmas já identificadas nos dados originais.

8. Calculamos o percentil 95 da distribuição empírica de T sob H_0 e avaliamos se o cluster mais verossímel é significativo ou não.

Para a implementação da extensão da estatistica Scan para dados de contagem com censura, identificada a partir deste momento como $Scan\text{-}Binomial_{censored}$, nos baseamos no algoritmo *Scan-Binomial*, realizando as adaptações apresentadas na seção 5.1. As modificações estão listadas a seguir:

- Na construção da matriz inicial, além das informações de população, número de casos, coordenadas do centróide e a variável identificadora ID, também consideramos a informação da indicadora de censura (*cens*). A codificação dessa variável segue a definição 5.1.

```
coords <- cbind(x,y)
id <- seq(1:nrow(coords))
data <- cbind(id,pop,cases,coords,cens)
```

- Para realizar a estimação dos parâmetros θ, definimos as verossimilhanças apresentadas em 5.4 e 5.5.

```
## Definindo a log-verossimilhança sob H0
loglikehood <- function(p,n,xi,censor) {
-( sum((1-censor)*dbinom(xi,prob=p,size=n,log=TRUE) )+
sum(censor*pbinom(xi, prob=p, size=n, log=TRUE,lower.tail=FALSE)) )
}
## Definindo a log-verossimilhança sob H0 para cálculo da razão de verossimilhança
l_0 <- function(p,n,xi,censor) {
 l0 = sum( (1-censor)*dbinom(xi,prob=p,size=n,log=TRUE) ) +
 sum( censor*pbinom(xi, prob=p, size=n, log=TRUE,lower.tail=FALSE) )
 return(l0)
}
## Definindo a log-verossimilhança sob Ha
loglikehood_ha <- function(p, n_zbar, xi_zbar, n_z, xi_z, censor_zbar, censor_z) {
-(
sum( (1-censor_zbar)*dbinom(xi_zbar,prob=p[1],size=n_zbar,log=TRUE) ) + sum(
     censor_zbar*pbinom(xi_zbar, prob=p[1], size=n_zbar, log=TRUE,lower.tail=FALSE) )+
sum( (1-censor_z)*dbinom(xi_z,prob=p[2],size=n_z,log=TRUE) ) + sum( censor_z*pbinom(xi_z,
     prob=p[2], size=n_z, log=TRUE,lower.tail=FALSE) )
)
}
## Definindo a log-verossimilhança sob Ha cálculo da razão de verossimilhança
l_a <- function(p0,pz,n_zbar,xi_zbar, n_z, xi_z, censor_zbar, censor_z) {
 la = sum( (1-censor_zbar)*dbinom(xi_zbar,prob=p0,size=n_zbar,log=TRUE) ) +
         sum( censor_zbar*pbinom(xi_zbar, prob=p0, size=n_zbar, log=TRUE,lower.tail=FALSE) )+
      sum( (1-censor_z)*dbinom(xi_z,prob=pz,size=n_z,log=TRUE) ) +
         sum( censor_z*pbinom(xi_z, prob=pz, size=n_z, log=TRUE,lower.tail=FALSE) )
 return(la)
}
```

- A estimação dos parâmetros utilizados no cálculo da estatística razão de verossimilhança para as zonas candidatas a cluster, foi realizada através dos

métodos de Brent e Nelder-Mead, disponíveis no R atráves da função *optim*. Essa adaptação é incorporada no passo 4 do algoritmo *Scan-Binomial*.

```
## Estimando theta via método de Brent
maxi <- try( optim(p=theta,loglikehood,n=data[,2],xi=data[,3], censor=cens, method='Brent',
     lower=0.001,upper=0.999), silent=TRUE )
if ( inherits(maxi ,"try-error") ){ maxi <- try(
     optim(p=0.001,loglikehood,n=data[,2],xi=data[,3], censor=cens, method='Brent',
     lower=0.001,upper=0.999), silent=TRUE ) }
theta_maxi <- maxi$par ##theta sob H0 obtido via maximização

zones <- lapply(1:n_regions,
 function(z){
  dist_matrix <- dist_matrix[order(dist_matrix[,z]),]
  valid_zones <- sapply(1:n_regions, function(j){
  n_z <- sum( data[ dist_matrix[1:j,n_regions+1], 2] )
  return(n_z < 0.25*N) })

  partial_zones <- lapply(1:sum(valid_zones),
   function(y){
        n_z <- sum( data[ dist_matrix[1:y,n_regions+1],2] )
        x_z <- sum( data[ dist_matrix[1:y,n_regions+1],3] )
        x_z_bar <- C - x_z
        n_z_bar <- N - n_z
        theta_0 <- x_z_bar/n_z_bar
        theta_1 <- x_z/n_z
        ## Estimando thetaZ via método de Nelder-Mead
        maxi_ha <- try( optim(p=c(theta_0,theta_1),loglikehood_ha,
         n_zbar=data[ dist_matrix[-(1:y),n_regions+1],2],
         xi_zbar=data[ dist_matrix[-(1:y),n_regions+1],3],
         censor_zbar=cens[dist_matrix[-(1:y),n_regions+1]],
         n_z=data[ dist_matrix[1:y,n_regions+1],2],
         xi_z=data[ dist_matrix[1:y,n_regions+1],3],
         censor_z=cens[dist_matrix[1:y,n_regions+1]]
         ), silent=TRUE)

        if ( !inherits(maxi_ha ,"try-error") ) {
         theta0_maxi <- maxi_ha$par[1]
         theta1_maxi <- maxi_ha$par[2]
        }else{
         maxi_ha <- try( optim(p=c(0.001,0.001),loglikehood_ha,
          n_zbar=data[ dist_matrix[-(1:y),n_regions+1],2],
          xi_zbar=data[ dist_matrix[-(1:y),n_regions+1],3],
          censor_zbar=cens[dist_matrix[-(1:y),n_regions+1]],
          n_z=data[ dist_matrix[1:y,n_regions+1],2],
          xi_z=data[ dist_matrix[1:y,n_regions+1],3],
          censor_z=cens[dist_matrix[1:y,n_regions+1]]
```

```
        ), silent=TRUE)
      if ( !inherits(maxi_ha ,"try-error") ){
       theta0_maxi <- maxi_ha$par[1]
       theta1_maxi <- maxi_ha$par[2]
      }else{
       theta0_maxi <- NA
       theta1_maxi <- NA
      }
      }

      ifelse( theta1_maxi > theta_maxi & theta1_maxi>theta_1,
       llr <- l_a(p0=theta0_maxi,pz=theta1_maxi,
        n_zbar=data[ dist_matrix[-(1:y),n_regions+1],2],
        xi_zbar=data[ dist_matrix[-(1:y),n_regions+1],3],
        censor_zbar=cens[dist_matrix[-(1:y),n_regions+1]],
        n_z=data[ dist_matrix[1:y,n_regions+1],2],
        xi_z=data[ dist_matrix[1:y,n_regions+1],3],
        censor_z=cens[dist_matrix[1:y,n_regions+1]]) -
            l_0(p=theta_maxi,n=data[,2],xi=data[,3],censor=cens),
       llr <- 0)

      return(data.frame(zone= paste(sort(as.numeric(dist_matrix[1:y,n_regions+1])),
          collapse=", "),
      llr=round(llr,5),
      nzone=n_z,
      xzone = x_z,
      xzone_estimated=n_z*theta1_maxi,
      expected_cases=n_z*theta_maxi,
      ))
 }
 )
 partial_zones <- rbind.fill(partial_zones)
 return(partial_zones)
})
```

- No processo de verificação da significância do cluster encontrado, adaptamos o procedimento referente a simulação das *nsim* réplicas do mapa original conforme proposto na seção 5.1.

```
sim_cases <- cbind(id,matrix(0,ncol=nsim,nrow=n_regions))
censor_index <- which(data[,6]==1)
N_noCensor <- sum(data[-censor_index,2])
C_noCensor <- sum(data[-censor_index,3])

for(k in 1:nsim){
sim_cases[ censor_index, k+1] <- data[censor_index,3]
```

```
sim_cases[ -censor_index, k+1] <- rmultinom(1, size = C_noCensor, prob =
    data[-censor_index,2]/N_noCensor)
}
```

Os demais procedimentos do algoritmo $Scan\text{-}Binomial_{censored}$ foram iguais aos do $Scan\text{-}Binomial$. A maneira como foi realizada a implementação dos dois algoritmos permite a fácil obtenção de clusters secundários, caso haja interesse.

Para maiores informações a cerca da função *optim* ou de alguma outra função do *software* R, consultar o sítio `https://cran.r-project.org/`. Os códigos completos estão apresentados no Apêndice A deste trabalho.

5.3 Dados Simulados

A aplicação inicial dos algoritmos implementados foi realizada em dois conjuntos de cenários artificiais. O uso de cenários artificiais controlados teve como objetivo, primeiramente, validar o algoritmo *Scan-Binomial*, atestando seu desempenho no processo de detecção de clusters espaciais em situações sem a presença de dados censurados e, em seguida, comparar os resultados dos algoritmos *Scan-Binomial* e $Scan\text{-}Binomial_{censored}$ na situação de interesse do estudo, para avaliar a hipótese de que o método proposto é mais adequado do que o método Scan circular tradicional para casos com dados censurados .

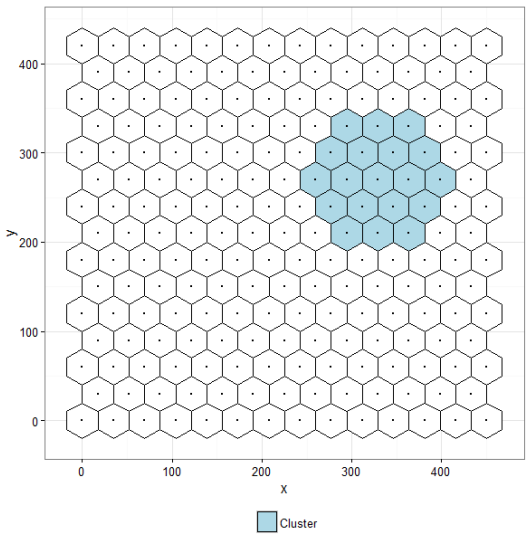
O primeiro conjunto é formado por 4 cenários artificiais (A_1,B_1,C_1 e D_1), onde cada um deles é formado por 203 regiões em formato de hexágono. Cada região possui uma população de $n_i = 1000$. Sendo assim, $N = \sum_i n_i = 203.000$, para todos os cenários. O total de casos C também foi fixado para todos os cenários, igual a $C = \sum_i x_i = 406$, ou seja, apenas $0,2\%$ da população total do cenário apresenta casos do evento simulado. Os cenários são descritos abaixo e estão ilustrados na Figura 5.3.

- O cenário A_1 possui um cluster em formato de hexágono, formado por 19 regiões. Devido ao seu formato circular mais bem definido, espera-se que o algoritmo *Scan-Binomial* consiga identificá-lo com mais facilidade;

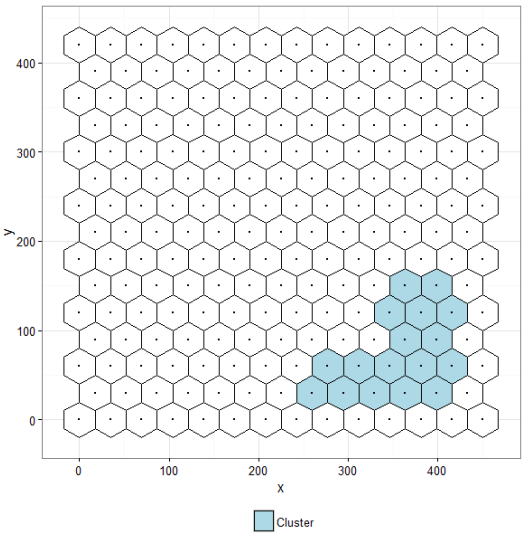
- O cenário B_1 possui um cluster em formato de "L" invertido, abrangendo 17 regiões. O formato irregular desse cluster deve afetar o desempenho do algoritmo, dificultando sua identificação;

- O cenário C_1 possui um cluster de formato diagonal alongado, composto pela sequência de 8 regiões. Esse cluster também apresenta formato irregular; e

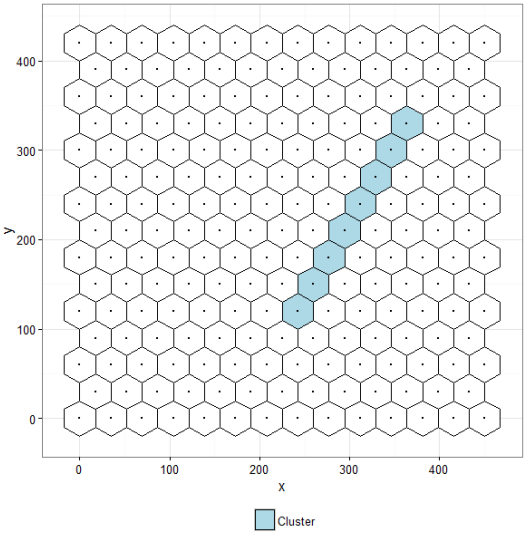
- O cenário D_1 possui um cluster semelhante ao observado em C_1. Esse cluster é formado por duas fileiras diagonais, composta por 16 regiões hexagonais.



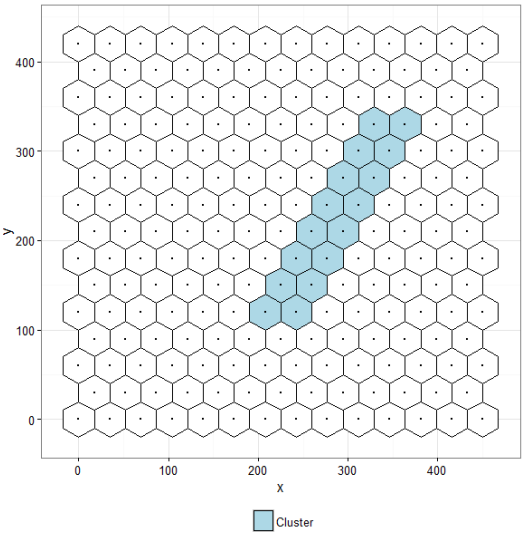
(a) Cenário A_1.



(b) Cenário B_1.



(c) Cenário C_1.



(d) Cenário D_1.

Figura 5.3: Cenários artificiais simulados sem a presença de censura. Regiões em azul representam o cluster.
Fonte: Elaborado pelo autor.

O segundo conjunto é formado por 4 cenários artificiais (A_2,B_2,C_2 e D_2), onde cada um deles também é formado por 203 regiões em formato de hexágono. Assim como no primeiro conjunto, cada região possui uma população de $n_i = 1000$, logo, $N = \sum_i n_i = 203.000$, para todos os cenários. Das 203 regiões, 15 foram selecionadas para apresentar informação censurada, também fixadas para os quatro cenários. Nessas 15 regiões, o número de casos foi determinado como sendo $x_i = 3$, de forma a simular uma situação de censura à direita, ou seja, o número de casos registrados

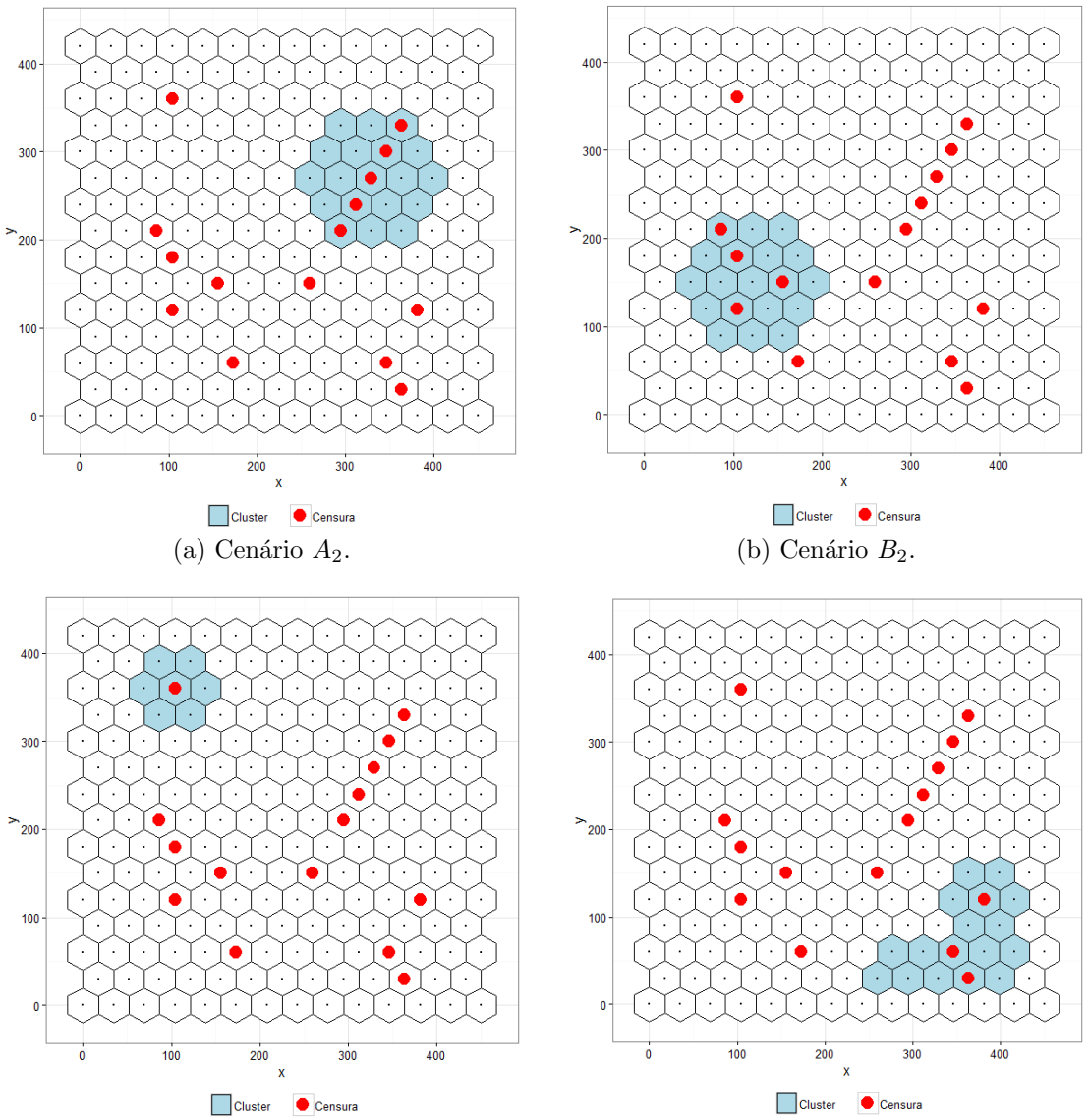
nessas 15 regiões é menor do que o número de casos real. Para os 4 cenários, o total de casos C é igual a 2030, o que representa 1% da população total nos mapas simulados. Os cenários são descritos abaixo e estão ilustrados na Figura 5.4.

- O cenário A_2 possui um cluster em formato circular, abrangendo 19 regiões. As regiões em sua diagonal apresentam censura e funcionam como um divisor do cluster, isto é, se removidas, separam o cluster em zonas desconexas;
- O cenário B_2 também possui um cluster em formato circular formado por 19 regiões. As regiões que apresentam censura dentro do cluster estão distribuídas de maneira aleatória;
- O cenário C_2 possui um pequeno cluster circular, composto por 7 regiões, sendo que a região do meio apresenta censura; e
- O cenário D_2 possui um cluster em formato de "L" invertido, formado por 17 regiões. Esse cluster possui um região censurada no meio da parte superior e duas regiões censuradas separando a parte inferior esquerda.

Os dois conjuntos de cenários foram definidos de forma similar aos que foram descritos em Cançado et al. (2014), de modo a comparar os métodos avaliados em diferentes situações de tamanho, formato e número de regiões com censura dentro dos clusters. O processo de simulação dos dados para os 8 cenários apresentados é detalhado a seguir.

5.3.1 Simulação dos dados

Para cada cenário, foram geradas 1000 simulações de banco de dados, onde os C casos fixados em cada cenário foram distribuídos aleatoriamente utilizando uma distribuição multinomial com probabilidades proporcionais aos riscos relativos de cada região. Os riscos relativos são calculados de acordo com Kulldorff et al. (2003), de maneira que as regiões dentro do cluster possuem alto risco relativo, enquanto que nas regiões fora dele o risco será baixo. Os riscos relativos para as regiões dentro do cluster devem ser altos o suficiente para que um teste Binomial simples rejeite com



(a) Cenário A_2.

(b) Cenário B_2.

(c) Cenário C_2.

(d) Cenário D_2.

Figura 5.4: Cenários artificiais simulados com a presença de censura. Regiões em azul representam o cluster e pontos em vermelho indicam quais regiões apresentam censura.
Fonte: Elaborado pelo autor.

probabilidade de $0,999$ a hipótese nula de ausência de cluster. Seja n_z a população observada dentro do cluster, N o total da população considerando todas as regiões, C o total de casos e H_0 a hipótese nula de ausência de cluster. Condicional a C, o número de casos sob H_0 para as regiões dentro do cluster segue uma distribução Binomial com media $m_0 = Cn_z/N$ e variância $v_0 = m_0(N - n_z)/N$. Aproximando a distribuição Binomial pela Normal, tem-se que o número de casos k necessário

para que o teste unilateral rejeite a hipotese nula com 5% de significância é tal que $(k - m_0)/\sqrt{v_0} = 1,645$.

Sob a hipótese alternativa, as regiões dentro do cluster possuem risco relativo r e os casos tem distribuição Binomial com media $m_1 = (Cn_z r)/(N - n_z + n_z r)$ e variancia $v_1 = m_1(N - n_z)/(N - n_z + n_z r)$. Sendo assim, aproximando novamente pela normal, calcula-se o risco relativo r de tal forma que $(k - m_1)/\sqrt{v_1} = 3,09$. Essa escolha é feita para que a hipótese nula seja rejeitada com probabilidade de $0,999$ quando realizado um teste Binomial simples. Para as regiões fora do cluster, o risco relativo é fixado em $r = 1$.

Uma região com censura dentro do cluster pode ser interpretada como uma área de alto risco em que a contagem de casos não é conhecida. Neste caso, apesar do alto risco relativo, a distribuição de casos dentro do cluster foi realizada somente nas regiões que não possuem censura. Um caso atribuído para uma região com censura foi rejeitado e outro caso foi gerado. Este procedimento garantiu que o número de casos em regiões com censura seja sempre nulo, para que posteriormente seja fixado com o valor igual a 3.

5.3.2 Análise de desempenho

Para avaliar os resultados obtidos pelos métodos propostos na aplicação em cenários simulados, utilizamos três medidas de desempenho que permitiram comparar os dois algoritmos implementados. Para cada cenário, em cada uma das M simulações, calculamos a estatística de teste λ correspondente e comparamos a mesma com um valor crítico λ^*, obtido através das simulações de Monte Carlo, sob H_0, conforme visto em 2.7, no caso de aplicação do algortimo *Scan-Binomial*, e em 5.1, no caso de aplicação do algoritmo $Scan\text{-}Binomial_{censored}$. Define-se, então, o poder como

$$Poder = \frac{\sum_i^M I(\lambda > \lambda^*)}{M}, \tag{5.6}$$

onde $I(\lambda_i > \lambda^*) = 1$, caso $\lambda_i > \lambda^*$, e 0, caso contrário. Utilizamos também as medidas de Sensibilidade e Valor Preditivo Positivo (VPP), conforme apresentado por Kulldorff et al. (2009). Tais medidas são baseadas na comparação das populações

do cluster detectado e do cluster verdadeiro.

$$Sensibilidade = \frac{Pop\,(ClusterDetectado \cap ClusterVerdadeiro)}{Pop\,(ClusterVerdadeiro)}. \tag{5.7}$$

$$VPP = \frac{Pop\,(ClusterDetectado \cap ClusterVerdadeiro)}{Pop\,(ClusterDetectado)}. \tag{5.8}$$

A Sensibilidade pode ser interpretada como o quanto do cluster verdadeiro é detectado, ou seja, a probabilidade de acerto do cluster detectado dado o conhecimento do cluster verdadeiro. Já o VPP representa o quanto do cluster detectado pertence ao verdadeiro, ou seja, a acurácia na identitifição do cluster verdadeiro dado o cluster detectado pelo algoritmo. Dessa forma, em casos de estudos envolvendo detecção de clusters de doenças, por exemplo, a utilização de um método com maior Sensibilidade é benéfico (Fernandes, 2015). O resultado ideal é que essas duas medidas estejam o mais próximo possível do valor 1, o que indicaria uma grande eficiência por parte do algoritmo aplicado no processo de detecção do verdadeito cluster.

5.4 Dados reais

Com o intuito de aplicarmos também a metodologia proposta considerando uma situação real, além do estudo de simulação, analisamos um banco de dados real com o número de homicídios nos municípios do estado do Rio de Janeiro para o ano de 2014. Esses dados foram extraídos das bases do Subsistema de Informação sobre Mortalidade (SIM), que atualmente estão disponibilizadas pelo Departamento de Informática do SUS (Datasus), no sítio `http://www2.datasus.gov.br/DATASUS/`.

Um aspecto importante na elaboração dessa base de dados foi a identificação da causa da morte para delimitar apenas os casos de homicídios. Todos os países do mundo, incluindo o Brasil, utilizam o sistema classificatório de morbidade e mortalidade desenvolvido pela Organização Mundial da Saúde (OMS). Até 1995, tais causas eram classificadas seguindo os capítulos da nona revisão da Classificação Internacional de Doenças (CID-9). A partir daquela data, o Ministério da Saúde adotou a décima revisão (CID-10). A CID-10 fornece códigos relativos à classifica-

ção de doenças e de uma grande variedade de sinais, sintomas, aspectos anormais, queixas, circunstâncias sociais e causas externas para ferimentos ou doenças, e visa padronizar a codificação de doenças e outros problemas relacionados à saúde.

Segundo o Mapa da Violência de 2014, divulgado pela Secretaria-Geral da Presidênca da República e Secretaria Nacional da Juventude, homicídios indicam, por excelência, formas conflitivas de relacionamento interpessoal que acabam com a morte de algum dos antagonistas. Corresponde ao somatório das categorias $X85$ a $Y09$ do CID-10, recebendo o título genérico de agressões. De maneira geral, corresponde a uma agressão intencional de terceiros, que utilizam qualquer meio para provocar danos, lesões que levam à morte da vítima.

Dessa forma, selecionamos todos os casos de óbitos que possuíam classificação de $X85$ a $Y09$, para o ano de 2014, nos 92 municipíos que formam o estado do Rio de Janeiro. As informações de população, importantes para o cálculo das taxas de homicídios, foram obtidas através das projeções populacionais utilizadas pelo Tribunal de Contas da União, que também estão disponibilizadas no DATASUS.

Como a base selecionada não apresentava dados censurados, criamos casos de censura de maneira artificial para aplicação do algoritmo $Scan\text{-}Binomial_{censored}$. Selecionamos 17 municípios no mapa e censuramos a informação do número de homicídios. Para esses municípios, o número de casos observados foi fixado em 5, de maneira a simular cenários de censura à direita, ou seja, o número de homicídios registrados é menor do que o número real de casos no município. A Figura 5.5 mostra em quais municípios foram criadas as censuras artificiais. Nota-se que há uma maior concentração de municípios censurados na região metropolitana do estado. Essa maior concentração ocorre devido ao fato de que essa região apresenta municípios com taxas de homicídios mais elevadas, o que torna mais pertinente para avaliação do método proposto a presença de censuras nesses municípios, já que a perda de informação será grande nesses casos. Esses resultados serão detalhados mais adiante no capítulo 6.

Os dados completos de número de homicídios e população dos 92 municípios analisados, assim como a descrição de todos os códigos do CID-10 considerados na elaboração dessa base de dados, podem ser vistos no Apêndice B.

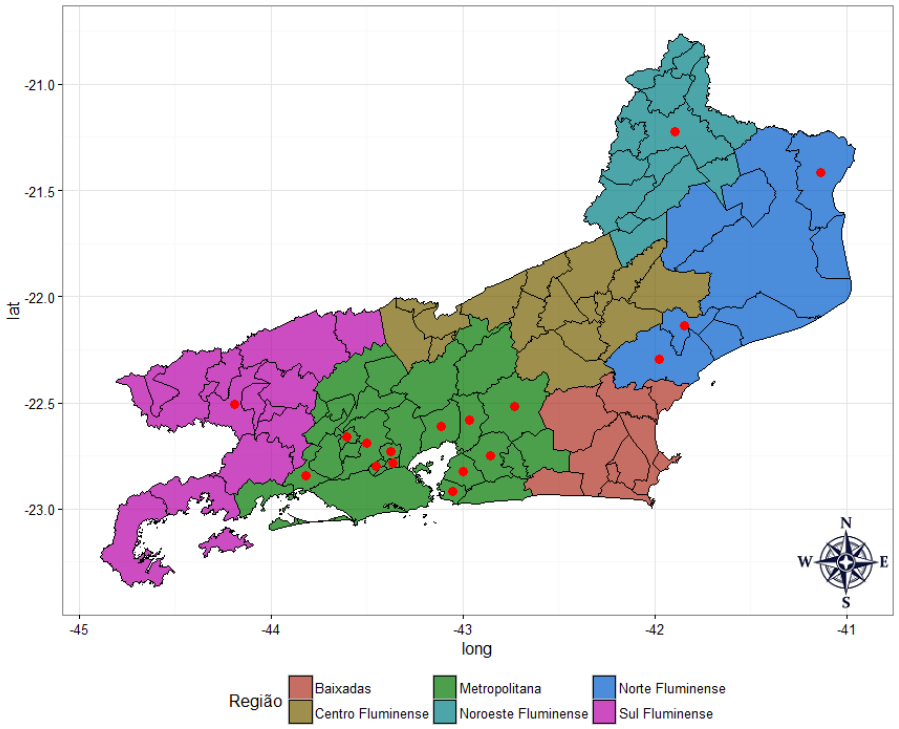


Figura 5.5: Municípios do Rio de Janeiro. Os pontos em vermelho indicam quais municípios apresentam informação censurada.
Fonte: Elaborado pelo autor.

Capítulo 6

Resultados

Neste capítulo, serão apresentados os resultados obtidos a partir das análises dos dados simulados e do banco de dados de homicídios extraído do DATASUS, conforme descrito nas seções 5.3 e 5.4, respectivamente. Nas análises, utilizamos os algoritmos *Scan-Binomial* e $Scan\text{-}Binomial_{censored}$, implementados em linguagem *R* apresentados na seção 5.2. Na aplicação em dados simulados, o desempenho dos algoritmos foi avaliado e comparado através das medidas apresentadas em 5.3.2. Na aplicação em dados reais, identificamos os conglomerados espaciais detectados pelo *Scan-Binomial*, quando aplicado nos dados originais, e comparamos esse resultado com os conglomerados espaciais identificados pelo mesmo *Scan-Binomial* e pelo $Scan\text{-}Binomial_{censored}$ quando aplicados nos dados censurados artificialmente.

6.1 Dados simulados

Inicialmente, aplicamos o algoritmo *Scan-Binomial* para o primeiro conjunto de cenários artificiais simulados. Essa aplicação teve intuito de verificar o algoritmo implementado para a estatística Scan tradicional e atestar seu desempenho quando aplicado em dados que não apresentam censura. A Tabela 6.1 apresenta os resultados observados na análise das 1.000 simulações dos quatro cenários analisados. Os valores de Sensibilidade e VPP apresentados são as médias dessas medidas para todos os casos simulados que retornaram clusters significativos.

Tabela 6.1: Resultados do algoritmo *Scan-Binomial* no primeiro conjunto de dados simulados.

Cenários	**Poder**	**Sensibilidade**	**VPP**
A_1	0,9390	0,8681	0,8732
B_1	0,8840	0,7189	0,6743
C_1	0,6020	0,4113	0,4848
D_1	0,7480	0,4550	0,6041

Observa-se que o algoritmo *Scan-Binomial* apresentou melhores resultados na análise do cenário A_1, com um poder de $0,9390$, que indica uma alta capacidade de detecção do cluster espacial. A sensibilidade média foi de $0,8681$, valor próximo ao VPP médio, que foi igual a $0,8732$. Esses valores acima de $0,8$ para essas duas medidas indicam que o algoritmo tem bom desempenho no processo de identificação do verdadeiro cluster. Analisando os demais cenários, nota-se que o algoritmo vai perdendo performance à medida que o cluster verdadeiro assume formas mais irregulares. No cenário B_1, onde o cluster verdadeiro possui formato de "L" invertido, o poder de detecção do algoritmo ainda foi alto, aproximadamente $0,9$. Entretanto, sua capacidade de detectar o verdadeiro cluster não é tão elevada, já que sua sensibilidade média foi de $0,7189$ e seu VPP médio foi de $0,6743$. Já nos cenários C_1 e D_1, onde os clusters verdadeiros assumem formas mais alongadas, o poder de detecção reduziu bastante, com valores iguais a $0,6020$ e $0,7480$, respectivamente. Nesses dois cenários, a sensibilidade média do *Scan-Binomial* esteve abaixo de $0,5$, o que mostra uma baixa capacidade de identificação da composição real do verdadeiro cluster.

O bom funcionamento do algoritmo *Scan-Binomial* no cenário A_1 e sua piora de desempenho à medida que o cluster verdadeiro assumia formas mais irregulares eram resultados esperados, uma vez que já se sabe da literatura que a estatística Scan Espacial de Kulldorff é mais apropriada para detecção de um cluster circular único bem definido do que para situações em que o mapa apresenta mais de um cluster ou um cluster de formato muito irregular. Dessa forma, o algoritmo *Scan-Binomial* teve resultados dentro do esperado, o que atesta sua implementação.

Após a verificação do *Scan-Binomial* para o conjunto de dados simulados sem a presença de censuras, o passo seguinte foi aplicar os algoritmos *Scan-Binomial*

e *Scan-Binomial*$_{censored}$ para o segundo conjunto de cenários simulados, agora com a presença de regiões com censura, e em seguida, comparar seus desempenhos. A Tabela 6.2 apresenta os resultados de cada método para as 1000 simulações dos quatro cenários analisados.

Tabela 6.2: Resultados dos algoritmos *Scan-Binomial* e *Scan-Binomial*$_{censored}$ no conjunto de dados simulados com a presença de censura.

	Scan-Binomial			*Scan-Binomial*$_{censored}$		
Cenários	**Poder**	**Sensibilidade**	**VPP**	**Poder**	**Sensibilidade**	**VPP**
A_2	0,4330	0,2668	0,5328	0,6360	0,7947	0,8050
B_2	0,3660	0,4170	0,5489	0,6770	0,8030	0,8069
C_2	0,6480	0,7685	0,4881	0,6670	0,8721	0,7899
D_2	0,4730	0,3993	0,5499	0,5350	0,6564	0,6497

Nota-se que, de maneira geral, a presença de regiões com censura no mapa gera uma diminuição no poder de detecção do cluster. Nos quatro cenários analisados, o *Scan-Binomial* teve piores resultados em relação ao *Scan-Binomial*$_{censored}$ para todas as medidas de desempenho. No cenário A_2, o poder de detecção do algoritmo *Scan-Binomial*$_{censored}$ foi aproximadamente 1,5 vezes maior do que o poder do *Scan-Binomial*, enquanto que no cenário B_2, o *Scan-Binomial*$_{censored}$ teve um poder aproximadamente 1,8 vezes maior. Nos cenários C_2 e D_2 os poderes estiveram mais próximos, mas o algoritmo *Scan-Binomial*$_{censored}$ obteve melhor desempenho em ambos os casos.

Analisando as medidas de sensibilidade e o VPP, observamos com mais evidência a melhora no desempenho da extensão proposta. No cenário A_2, a sensibilidade média do *Scan-Binomial*$_{censored}$ foi aproximadamente três vezes maior do que a sensibilidade média do *Scan-Binomial*, enquanto que no cenário B_2, a sensibilidade média do *Scan-Binomial*$_{censored}$ foi aproximadamente duas vezes maior. No cenário C_2, que apresenta apenas uma região com censura no cluster real, há uma melhora expressiva no VPP médio, que foi de 0,48 no *Scan-Binomial* para 0,78 no *Scan-Binomial*$_{censored}$. No cenário D_2, que apresenta o cluster com o formato mais irregular dentre os quatro cenários analisados, as medidas de sensibilidade e VPP também foram melhores no *Scan-Binomial*$_{censored}$.

A partir desses resultados, foi possível perceber que a presença de regiões com

informação censurada causam um impacto negativo significativo no desempenho da estatística Scan tradicional. Por outro lado, o método proposto apresentou uma grande melhora no processo de identificação do cluster com a presença de censura, com melhores resultados em todas as medidas avaliadas, para todos os cenários analisados , confirmando assim o êxito de nossa proposta. Na próxima seção serão apresentados os resultados para os dados reais.

6.2 Dados reais

Para aplicação da metodologia proposta em dados reais, foram utilizados dados de homicídios nos municípios do Rio de Janeiro para o ano de 2014. De acordo com o que foi apresentado na seção 5.4, esse banco de dados foi extraído do DATASUS, onde selecionamos das bases de dados do SIM somente os óbitos que tiveram causas relacionadas à homicídios. Os dados de população foram obtidos através das projeções populacionais realizadas pelo TCU, que também estão disponibilizadas no DATASUS.

A questão da violência cotidiana é um tema bastante presente na realidade do Brasil. Seu contínuo crescimento configura um aspecto bastante representativo e problemático na atual organização da vida social, de modo que esse tema, assim como a segurança do cidadão, é uma das principais preocupações no Brasil. Segundo o Mapa da Violência de 2014, elaborado por Waiselfisz (2014, p.33):

> "*No Brasil - país sem disputas territoriais, movimentos emancipatórios, guerras civis, enfrentamentos religiosos, raciais ou étnicos, conflitos de fronteira ou atos terroristas -, foram contabilizados, nos últimos quatro anos disponíveis, de* 2008 *a* 2011*, um total de* 206.005 *vítimas de homicídios, número bem superior quando comparado aos números dos* ***12 maiores conflitos armados acontecidos no mundo entre 2004 e 2007****. E ainda, esse número de homicídios brasileiro resulta quase idêntico ao total de mortes diretas* ***nos 62 conflitos armados desse período, que foi de 208.349***". .

Nesse sentido, estudos que abordam essa temática mostram relevância, pois são fontes de informações importantes para o entendimento do panorama da violência e para a elaboração de políticas públicas. Neste estudo, focamos nossa atenção para a questão da violência no Rio de Janeiro.

O estado do Rio de Janeiro possui 92 municípios, ilustrado na Figura 6.1, e apresentou, em 2014, 5.412 casos de homicídios. Os municípios que registraram maior número de homicídios foram Rio de Janeiro, com 1.386 homicídios, Duque de Caxias e Nova Iguaçu, com 576 e 560 homicídios, respectivamente. Nota-se que esses três municípios são da região Metropolitana do estado. Dez municípios não registraram nenhum caso de homicídio no ano de 2014, sendo que quatro deles estão na região Noroeste e apenas um deles está na região Metropolitana.

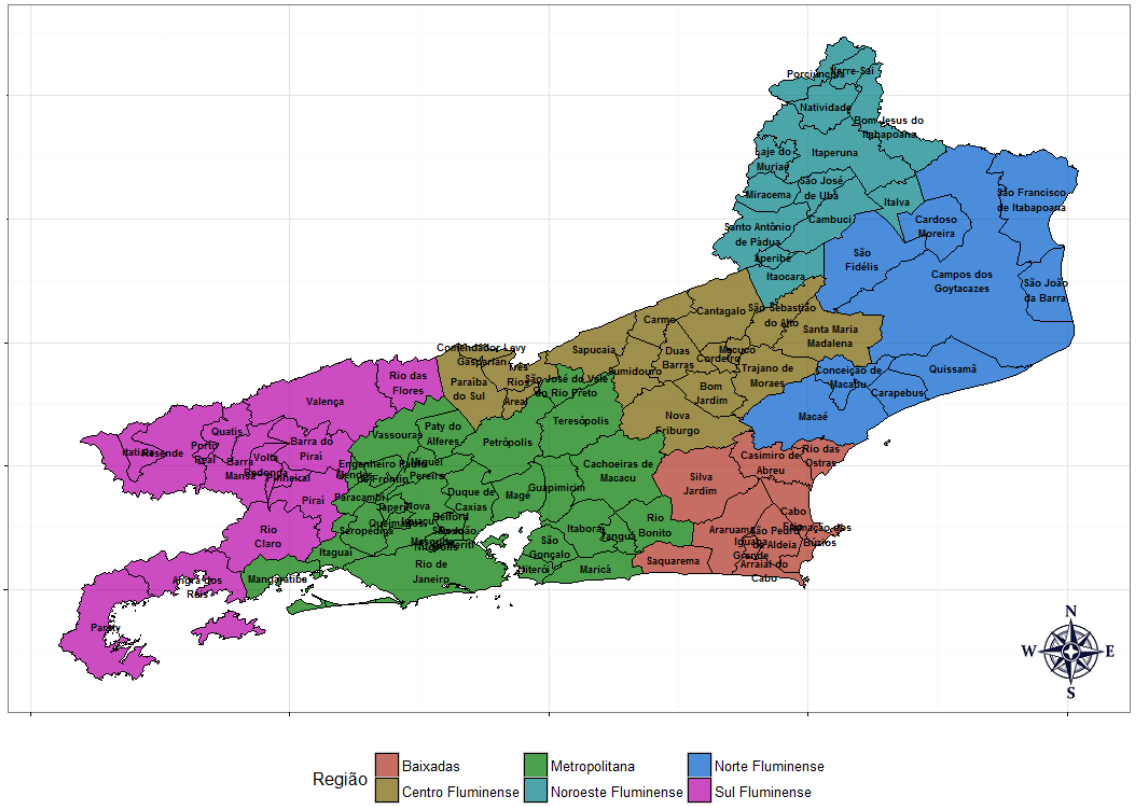


Figura 6.1: Municípios do Rio de Janeiro segundo região.
Fonte: Elaborado pelo autor.

A Tabela 6.3 apresenta o número de homicídios por região. A região Metropolitana registrou o maior número de casos no ano de 2014, com 4.186 homicídios, o que representa mais de 75% do total registrado no estado. As regiões Norte e das

Baixadas registraram números próximos, 412 e 389, respectivamente. O número de homicídios na região Sul representa, aproximadamente, 5,7% do total de casos, enquanto que os casos registrados nas regiãos Centro e Noroeste, somadas, representam um pouco mais de 2% do total de homicídios que ocorreram em 2014.

Tabela 6.3: Número de homicídios por região do RJ - 2014.

Região	Número de homicídios	Frequência Relativa
Metropolitana	4.186	77,35%
Norte Fluminense	412	7,61%
Baixadas	389	7,19%
Sul Fluminense	308	5,69%
Centro Fluminense	69	1,27%
Noroeste Fluminense	48	0,89%

Analisando a Tabela 6.4, que apresenta os cinco municípios com maiores taxas de homicídio por 100.000 habitantes, além da capital, nota-se que não são necessariamente os mesmos que apresentaram os maiores números de casos. Cabo Frio apresentou a maior taxa, aproximadamente 85 homicídios a cada 100 mil habitantes. Paraty, município da região Sul, teve uma taxa de homicídios de $75,17$, enquanto que Seropédica e Nova Iguaçu, cidades da região Metropolitana, tiveram taxas de 71,87 e 69,46 homicídios a cada 100 mil habitantes, respectivamente. Observa-se que as taxas nesses cinco municípios foram bem maiores do que a observada na capital Rio de Janeiro, e também estão bem acima da taxa de homicídio do Brasil[1] no mesmo ano de 2014, que foi aproximadamente de 29 homicídios a cada 100 mil habitantes.

Tabela 6.4: Cinco municípios com maiores taxas de homicídio no RJ - 2014.

Região	Município	Taxa de Homicídio (100.000 habitantes)
Baixadas	Cabo Frio	85,09
Sul Fluminense	Paraty	75,07
Metropolitana	Seropédica	71,87
Metropolotina	Nova Iguaçu	69,46
Baixadas	Armação dos Búzios	65,71
Metropolitana	**Rio de Janeiro**	**21,48**

[1]Fonte: `https://www12.senado.leg.br/institucional/omv/entenda-a-violencia/pdfs/atlas-da-violencia-2016`

Para melhor visualização dos dados ao longo do mapa, elaboramos alguns mapas de quartis para análise das informações coletadas. Observa-se na Figura 6.2, a qual mostra o mapa de quartis da população, que os municípios da região da região Metropolitana apresentam as maiores populações no estado. Na região Norte do estado, destaca-se o município de Campos de Goytacazes, e na região Sul destacam-se os municípios de Angra dos Reis, Volta Redonda e Barra Mansa. As menores populações se concentram em cidades das regiões Noroeste e Central.

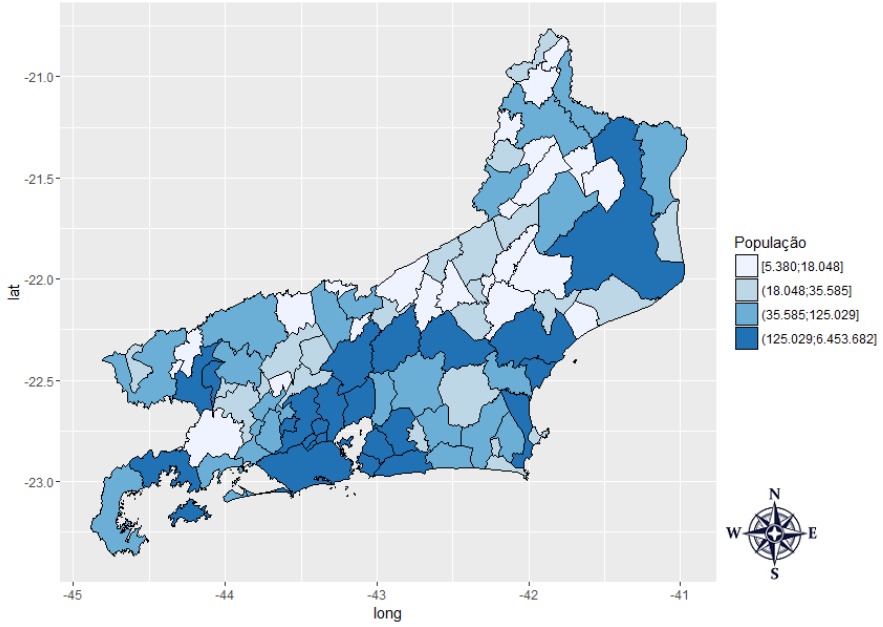


Figura 6.2: Mapa de quartis da população para os municípios do RJ - 2014.
Fonte: Elaborado pelo autor.

A Figura 6.3 corresponde ao mapa de quartis do número de homicídios. Nota-se que os municípios que registraram mais homicídios em 2014 se concentram na região Metropolitana. Na região da Baixadas, destacam-se as cidades de Cabo Frio e São Pedro da Aldeia. Na região Sul, o município de Angra dos Reis se sobressai em relação aos demais com um número maior de homicídios. Já na região Norte, destacam-se os municípios de Macaé e Campo de Goytacazes. As regiões Noroeste e Centro apresentam números de homicídios mais baixos em relação as demais cidades do estado.

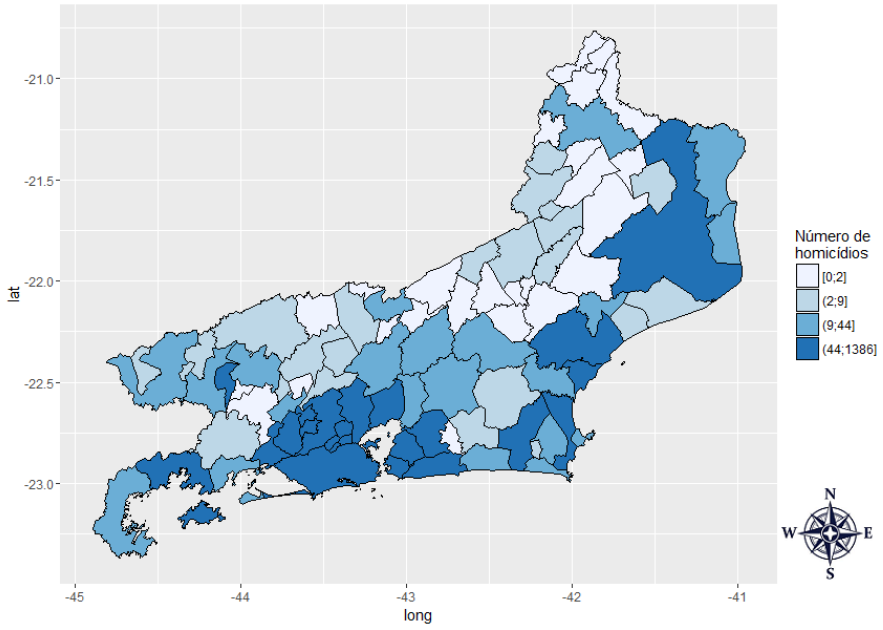


Figura 6.3: Mapa de quartis do número de homicídios para os municípios do RJ - 2014.
Fonte: Elaborado pelo autor.

Por fim, a Figura 6.4 apresenta o mapa de quartis da taxa de homicídio. Observa-se que as maiores concentrações de municípios com altas taxas de homicídio estão nas regiões Sul, Norte, Baixadas e principalmente na região Metropolitana. Podemos verificar que as cidades de Angra dos Reis, Campo de Goytacazes e Cabo Frio apresentaram altas taxas de homicídio, assim como municípios limítrofes a eles. Por outro lado, é interessante notar que o município do Rio de Janeiro não apresentou uma taxa alta, mesmo tendo registrado o maior número de homicídios no ano. Esse resultado ocorre devido a grande população que reside na capital, o que de certa forma padroniza o alto número de homicídios nesse caso.

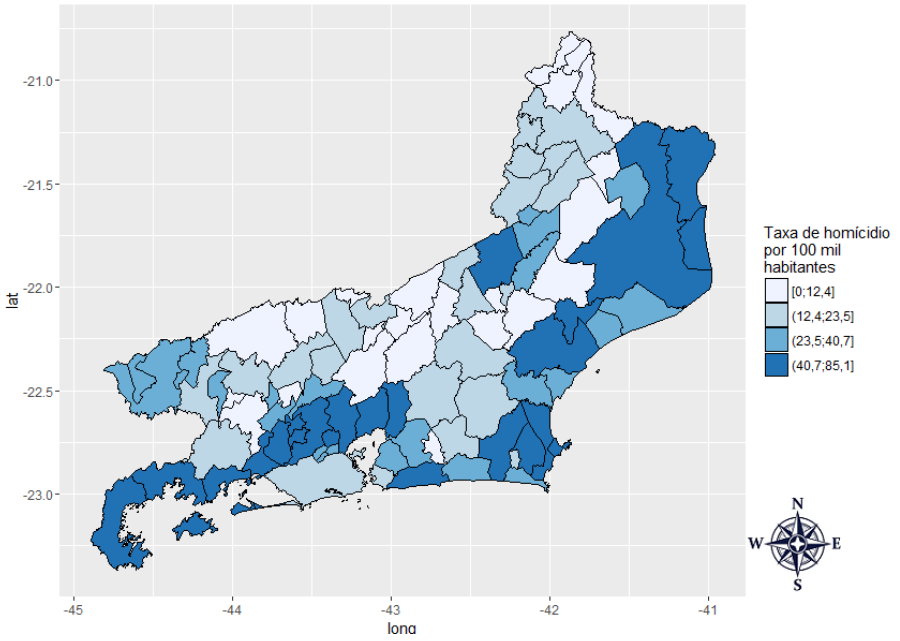


Figura 6.4: Mapa de quartis da taxa de homicídio por 100 mil habitantes para os municípios do RJ - 2014.
Fonte: Elaborado pelo autor.

Após a análise inicial, mostrada nas Tabelas 6.3 e 6.4 e nas Figuras 6.2, 6.3 e 6.4, aplicamos a estatística Scan circular para identificar possíveis conglomerados onde o número de homicídios seja maior do que o esperado. Inicialmente, aplicamos o algoritmo *Scan-Binomial* para realizar a detecção de possíveis clusters.

A Figura 6.5 apresenta os resultados obtidos pelo *Scan-Binomial*. O algoritmo identificou 6 clusters, sendo que três deles se localizam na região Metropolitana. Além disso, observou-se um cluster para cada uma das regiões Norte, Sul e das Baixadas. O cluster mais significativo (primário) foi identificado na região Metropolitana, composto por cidades próximas à capital Rio de Janeiro. Nota-se que quatro dos seis clusters identificados fazem fronteira e estão conectados, formando assim um grande conglomerado de municípios onde a taxa de homicídios foi acima do esperado.

Os resultados obtidos pelo *Scan-Binomial* corroboram com as análises iniciais feitas através dos mapas de quartis, onde já havíamos observado que a região Metropolitana concentrava um grande número de municípios com altas taxas de homicídio, assim como focos específicos verificados nas regiões Norte e das Baixadas, onde as taxas de homicídio também estava elevadas, mais especificicamente em torno de Cabo Frio e Campo dos Goytacazes.

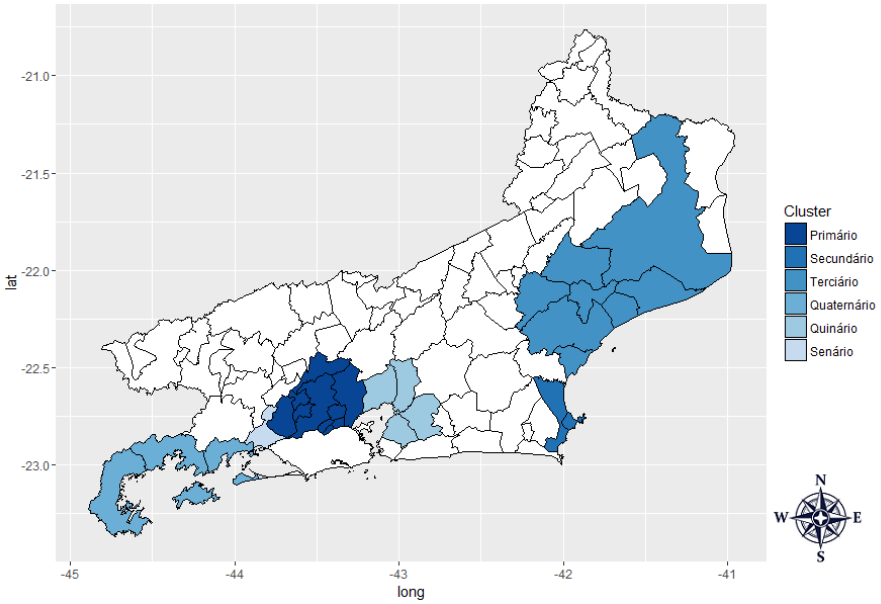


Figura 6.5: Clusters detectados pelo algoritmo *Scan-Binomial* - dados originais.
Fonte: Elaborado pelo autor.

A Tabela 6.5 apresenta com detalhes as informações dos seis clusters detectados pelo *Scan-Binomial*. O cluster primário é formado por 10 municípios, que registraram, no total, 1.889 homicídios, o que representa aproximadamente 35% do total

de casos no estado, para uma população total de 3.302.217 pessoas. Nesse cluster, observou-se aproximadamente 800 casos a mais de homicídios do que o seu valor esperado. Entre os municípios que o compõe, podemos destacar Duque de Caxias, Nova Iguaçu e Seropédica, cidades que já haviam se sobressaído nas análises descritivas como locais onde o número de casos era mais elevado.

O cluster secundário é formado por dois municípios, Cabo Frio e Armação dos Búzios, e apresentou um total de 194 homicídios, mais do que o dobro do número de casos esperado. A população observada foi de 234.925 pessoas. No cluster terciário, composto por 8 municípios, o total de homicídios observado foi de 417 casos, sendo que o número esperado foi de de 301 casos. Neste cluster, podemos destacar a presença das cidades de Campos dos Goytacazes e Macaé. Já o cluster quaternário é composto por 3 municípios da região Sul: Paraty, Angra dos Reis e Mangaratiba. O número de casos foi de 143 homicídios, para uma população total de 264.913 pessoas.

No que diz respeito à significância desses conglomerados, verifica-se que os 4 primeiros clusters foram bastante significativos, com p-valores abaixo de $0,0001$. O cluster quinário teve um p-valor de $0,005$, enquanto que o cluster senário apresntou p-valor de $0,007$

Tabela 6.5: Descrição dos clusters de homicídios detectados pelo *Scan-Binomial*.

Cluster	Quantidade de Municípios	População no cluster	Casos observados	Casos esperados	λ_z	p−valor
Primário	10	3.302.217	1.889	1.085,6820	322,6866	<0,0001
Secundário	2	234.925	194	77,2372	63,2235	<0,0001
Terciário	8	917.024	417	301,4933	21,0654	<0,0001
Quaternário	3	264.913	143	87,0964	15,3004	<0,0001
Quinário	4	1.548.331	598	509,0504	8,1692	0,005
Senário	1	117.374	66	38,5895	8,0833	0,007

Após a identificação dos clusters considerando o banco de dados sem a presença de censuras, aplicamos os algortimos *Scan-Binomial* e $Scan\text{-}Binomial_{censored}$ no banco de dados com a presença de censuras artificias. Mesmo com a presença das censuras, o objetivo desta aplicação era obter resultados próximos aos observados na aplicação com os dados originais e, com o intuito de avaliar se o método proposto também apresentaria melhores resultados do que a estatística Scan tradicional, fizemos o processo de identificação dos conglomerados através dessas duas metodologias.

A Figura 6.6 apresenta os resultados obtidos pelo *Scan-Binomial*, agora aplicado ao banco de dados com censura. A primeira conclusão que se chega é que a presença de municípios com censura afetou significativamente o desempenho do *Scan-Binomial*, e consequentemente, seus resultados também foram influênciados. Dos seis clusters identificados na aplicação do mesmo *Scan-Binomial* com os dados completos, apresentados na Figura 6.5, apenas um deles também foi detectado na aplicação com os dados censurados. É interessante notar que esse cluster não apresenta nenhum caso de censura, logo, era esperado que ele fosse identificada nas duas ocasiões.

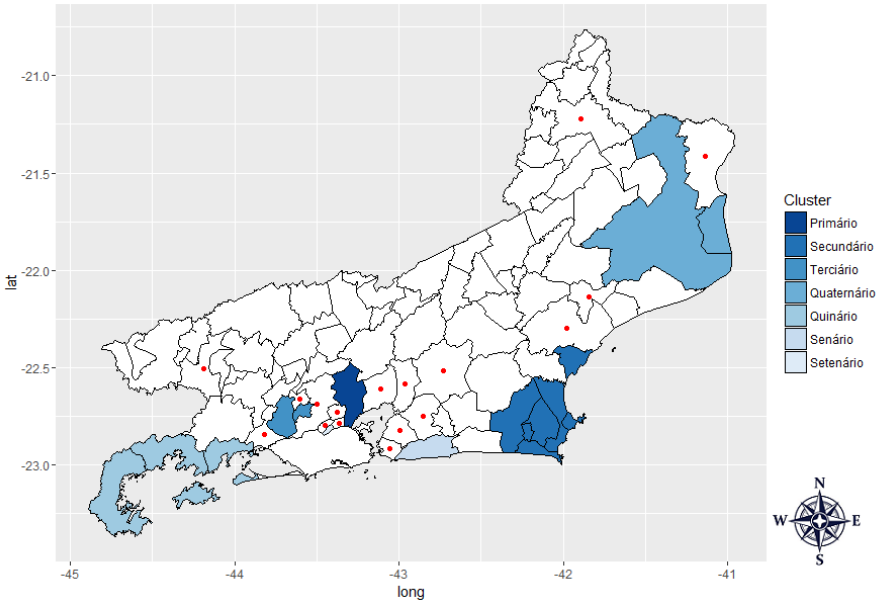


Figura 6.6: Clusters detectados pelo algoritmo *Scan-Binomial*. Os pontos em vermelho indicam quais municípios apresentam informação censurada.
Fonte: Elaborado pelo autor.

O *Scan-Binomial* apontou a presença de sete clusters significativos, sendo que nenhum deles teve em sua composição algum dos 17 municípios censurados. Esse resultado evidencia como o método Scan Circular tradicional tende a considerar regiões com presença de dado censurado como sendo menos verossímeis do que realmente são, o que gera grande impacto no processo de identificação do conglomerado. A maior influência se observa na região Metropolitana, que é a região que apresen-

tou a maior quantidade de municípios com altas taxas de homicídio, mas ao mesmo tempo, foi a região que recebeu o maior número de censuras. Na aplicação com os dados completos, o *Scan-Binomial* identificou três clusters significativos nessa região, formados por 15 municípios. Já com os dados censurados, o algoritmo identificou também três clusters, mas formados apenas por quatro municípios. Portanto, 11 municípios que formavam conglomerados significativos deixaram de ser captados por causa da censura.

A Tabela 6.6 apresenta com detalhes as informações dos sete clusters detectados pelo *Scan-Binomial* no caso de dados censurados. O cluster primário é composto por apenas um município, Guapimirim. Nele foram observados 576 homicídios, bem acima do seu valor esperado. O cluster secundário é formado por 7 municípios, onde se observou um total de 352 casos de homicídios, quase três vezes maior do que número de casos esperado.

Tabela 6.6: Descrição dos clusters de homicídios detectados pelo *Scan-Binomial* - dados censurados.

Cluster	Quantidade de Municípios	População no cluster	Casos observados	Casos esperados	λ_z	p−valor
Primário	1	878.402	576	183,8856	290,2233	<0,0001
Secundário	7	632.582	352	132,4254	132,0257	<0,0001
Terciário	2	224.799	151	47,0597	73,7351	<0,0001
Quaternário	2	514.921	242	107,7941	64,2557	<0,0001
Quinário	3	264.913	143	55,4572	49,0663	<0,0001
Senário	1	143.111	59	29,9590	11,0705	<0,0001
Setenário	1	158.299	61	33,1385	9,4759	0,004

Os clusters terciário e quaternário apresentaram, cada um, dois municípios. Já o cluster quinário, que é igual ao cluster quaternário da Tabela 6.5, é composto por três municípios. Nota-se para esse cluster, que seu valor esperado reduziu de 87 casos para 55 casos, o que mostra que a presença de dados censurados causou um impacto ao reduzir o número de casos esperados também para zonas que não apresentavam censura.

Esses resultados vão ao encontro das conclusões obtidas nas análises dos dados simulados, onde já tínhamos observado que o *Scan-Binomial* não é adequado para aplicações com dados censurados, já que seu poder de detecção e capacidade de iden-

tificação do cluster verdadeiro deteriora significativamente. Após a análise dos resultados obtidos pelo *Scan-Binomial*, fizemos a aplicação do $Scan\text{-}Binomial_{censored}$ nos dados censurados.

A Figura 6.7 apresenta os resultados obtidos pelo $Scan\text{-}Binomial_{censored}$. O algoritmo identificou 5 clusters, sendo que dois deles se localizam na região Metropolitana. Além disso, observou-se um cluster para cada uma das regiões Norte, Sul e das Baixadas. Ao contrário do que foi observado na Figura 6.6, o $Scan\text{-}Binomial_{censored}$ conseguiu tratar a questão da censura, já que em três dos cinco clusters identificados, existe a presença de municípios censurados.

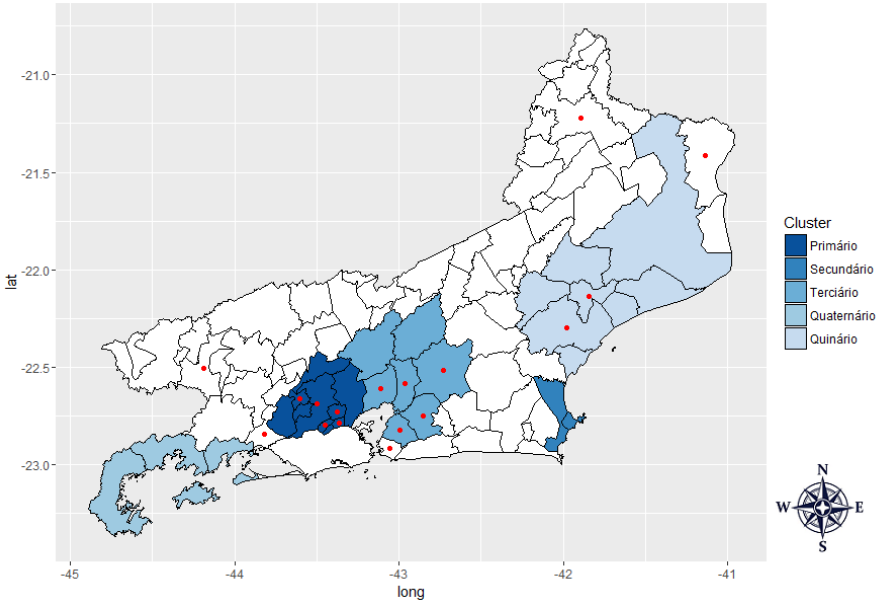


Figura 6.7: Clusters detectados pelo algoritmo $Scan\text{-}Binomial_{censored}$. Os pontos em vermelho indicam quais municípios apresentam informação censurada.
Fonte: Elaborado pelo autor.

Outro aspecto positivo, foi que dos cinco clusters detectados nos dados censurados, 4 deles também foram identificados na aplicação com os dados originais, o que mostra que o método proposto no algoritmo $Scan\text{-}Binomial_{censored}$ conseguiu trabalhar com a incerteza presente na informação censurada, e gerou resultados muito próximos aos obtidos com os dados completos.

Também podemos destacar que a adaptação realizada no $Scan\text{-}Binomial_{censored}$ não ocasionou uma perda de performance no aspecto de detecção de clusters sem a presença de censura. Dois dos cinco clusters identificados não apresentam municípios

com censura, e esses mesmos dois clusters também foram identificados na aplicação com os dados originais.

Por fim, ressaltamos que a relação de significância entre os clusters detectados não foi a mesma observada na análise com os dados originais. Entranto, o cluster primário selecionado pelo $Scan\text{-}Binomial_{censored}$ foi o mesmo selecionado pelo algoritmo *Scan-Binomial* na aplicação com dados originais, portanto, podemos assumir que a extensão proposta para dados de contagem com censura teve resultados bastante satisfatórios.

A Tabela 6.7 apresenta com detalhes as informações dos cinco clusters detectados pelo $Scan\text{-}Binomial_{censored}$ no caso de dados censurados. Nesta tabela, ao contrário das Tabelas 6.5 e 6.6, também temos a informação de número de casos estimados para cada cluster. Essa informação também é calculada, pois o número de casos observados não é necessariamente o número de casos real, por conta da presença de censuras. Dessa forma, além do número de casos esperado, também fizemos a estimação do número de casos através da múltiplicação entre a população dentro do cluster e a proporção de casos dentro do cluster estimada pelos algoritmos de otimização.

Tabela 6.7: Descrição dos clusters de homicídios detectados pelo $Scan\text{-}Binomial_{censored}$ - dados censurados.

Cluster	Qtd. de Munic.	População no cluster	Casos obs.	Casos estimados	Casos esperados	λ_z	p−valor
Primário	10	3.302.217	822	2045,9199	3302,2383	4333,7181	<0,0001
Secundário	2	234.925	194	194,0105	234,9265	4198,7703	<0,0001
Terciário	7	2.073.797	66	1305,9980	2073,8103	4163,5049	<0,0001
Quaternário	3	264.913	143	142,9832	264,9147	4142,1925	<0,0001
Quinário	8	917.024	296	394,9049	917,0299	4140,3424	<0,0001

Outro aspecto observado na Tabela 6.7 é que em todos os clusters listados, o número de casos estimado foi menor que o número de casos esperado, o que, a primeiro momento, pode ser interpretado como indicativo de que todos os clusters detectados são de baixa incidência. Entretanto, essa situação ocorreu devido a alta proporção de homicídios estimada considerando todos os municípios do mapa, que provavelmente pode ter sido acarretada pela presença de muitos municípios com censura. Desse modo, os valores de casos esperados também ficaram bastante ele-

vados. Após a análise de todos os clusters significativos detectados pelo algoritmo $Scan\text{-}Binomial_{censored}$, chegamos aos 5 clusters mais significativos de alta incidência apresentados na Figura 6.7 e Tabela 6.7.

Analisando os resultados, observamos que o cluster primário é composto por 10 municípios, sendo que 5 deles apresentaram censura. O número de homicídios observados foi de 822 casos, enquanto que o valor estimado foi de aproximadamente 2.045 casos. Sabemos atravé da Tabela 6.5 que o número real de casos observados nesse cluster é de 1.889, um pouco abaixo do valor estimado e bem acima do valor observado.

Nota-se para os clusters secudário e quaternário, que o número de casos observados e o número de casos estimados foram praticamente iguais. Esse resultado era esperado, uma vez que nesses clusters não há presença de municípios com censura. No cluster quinário, que é formado por 6 municípios sem censura e 2 com censura, o número de casos observados foi de 296, enquanto que o número estimado de casos foi de 394. Através da Tabela 6.5, sabemos que o número real de homicídios observados nesse cluster é de 417, bem próximo do valor estimado pelo algoritmo $Scan\text{-}Binomial_{censored}$.

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Conclusões

Nesta dissertação, apresentamos uma estatística de varredura espacial para dados de contagem com censura. Essa extensão consistiu em adaptações no método Scan tradicional que permitiram incorporar a informação da censura no processo de estimação da estatística razão de verossimilhança e no procedimento de verificação da siginificância do cluster detectado. Essa adaptações foram realizadas através de adequações nas fórmulas de verossimilhança e no método de estimação dos seus parâmetros, além de ajustes na simulação via Monte Carlo para obtenção da distribuição empírica da estatística do teste. Para comparar o método proposto com o método usual da estatística Scan, fizemos a implementação das duas técnicas em linguagem R e realizamos aplicações em dados simulados, através de cenários controlados, e em dados reais, referentes a casos de homicídios nos municípios do Rio de Janeiro, onde foram geradas censuras artificiais.

Os resultados obtidos na análise dos dados simulados mostraram que a presença de regiões com informação censurada causam um significativo impacto negativo no desempenho da estatística Scan tradicional, deteriorando sua capacidade de detecção e sua precisão na identificação do cluster real. Por outro lado, o método proposto apresentou uma grande melhora em relação ao método usual, já que em todos os cenários testados, o algortimo $Scan\text{-}Binomial_{censored}$ apresentou melhor poder de

detecção, assim como melhores resultados para as medidas de Sensibilidade média e VPP médio. Esses resultados indicam que o método proposto foi mais eficiente na identificação do cluster e foi mais preciso na caracterização desse cluster em relação ao método usual.

Com relação aos resultados obtidos na aplicação com dados reais, observamos novamente uma melhor performance por parte do $Scan\text{-}Binomial_{censored}$ em relação ao *Scan-Binomial*. Aplicando o método usual no conjunto de dados sem censura, identificamos seis clusters significativos. Entretanto, ao aplicarmos o mesmo método no conjunto de dados após a criação das censuras artificiais, apenas um desses seis clusters foi identificado novamente. Também verificamos que o método usual considerou municípios com a presença de censura como sendo menos verossímeis do que realmente são, já que 12 municípios que antes da criação da censura faziam parte de clusters significativos, com a presença da censura não foram mais considerados em nenhum cluster detectado.

Em contrapartida, quando aplicamos o $Scan\text{-}Binomial_{censored}$ aos dados censurados, identificamos 5 clusters significativos, sendo que 4 deles também foram identificados na aplicação com os dados completos. Notamos, também, que em três desses clusters existe a presença de municípios cuja variável de interesse apresenta censura, o que mostra boa capacidade do método proposto em tratar a informação censurada. No que diz respeito ao nível de significância, o $Scan\text{-}Binomial_{censored}$ identificou o mesmo cluster primário observado na aplicação com os dados originais. Por fim, observamos que o número de casos estimados para os clusters esteve bem próximo do número de casos real obtidos na análise dos dados sem a censura.

Dessa forma, podemos concluir a partir dos resultados observados nas análises dos dados simulados e dos dados reais que a extensão proposta é mais eficiente do que o método usual da estatística Scan na detecção de clusters espaciais quando há presença de censuras. O método proposto apresentou maior poder de detecção e maior precisão no processo de identificação do cluster, de modo que as adaptações realizadas no método usual para incorporar a informação censurada refletiram em uma melhora de performance, com a obtenção de melhores resultados.

7.2 Trabalhos Futuros

Apesar da extensão proposta para a estatística Scan espacial ter apresentado resultados bastante satisfatórios no processo de detecção de conglomerados espaciais na presença de dados censurados, principalmente quando comparados com os resultados obtidos pelo método tradicional no mesmo contexto, alguns aprofundamentos em relação ao método apresentado podem trazer melhorias na identificação de clusters. Entre as sugestões que podem gerar trabalhos futuros dentro do mesmo cenário abordado, podemos citar:

- Incorporar outros modelos de distribuição para dados discretos no algoritmo proposto, por exemplo, a distribuição Poisson. Dessa forma, a metodologia proposta seria mais abrangente, ampliando as opções de aplicação e alternativas de análises.

- Avaliação de outros métodos de otimização no processo de estimação dos parâmetros da verossimilhança.

- Na formulação da verossimilhança, incorporar um limite superior na probabilidade referente ao dado censurado, de forma que esse valor reflita mais fielmente a proporção de número de casos em relação à população total.

- Considerar a situação onde não se conhece quais regiões possuem censura, de forma que o processo de estimação deverá sofrer alguma adaptação.

- Desenvolver a extensão proposta para aplicações voltadas para detecção de conglomerados espaço-tempo.

- Variar a proporção de regiões com presença de censura dentro do mesmo cluster e avaliar o comportamento e a eficiência da extensão proposta no processo de identificação do cluster.

Referências Bibliográficas

Abrams, A. M., Kleinman, K., & Kulldorff, M. (2010). Gumbel based p-value approximations for spatial scan statistics. *International Journal of Health Geographics*, 9(1):1.

Ala, A., Stanca, C. M., Bu-Ghanim, M., Ahmado, I., Branch, A. D., Schiano, T. D., Odin, J. A., & Bach, N. (2006). Increased prevalence of primary biliary cirrhosis near superfund toxic waste sites. *Hepatology*, 43(3):525–531.

Araújo, T. C. (2013). Extensão da estatística scan para detecção de conglomerados espaço-temporais em dados com excesso de zeros. Master's thesis, Universidade de Brasília.

Assunção, R. M. (2001). Estatística espacial com aplicações em epidemiologia, economia e sociologia. *São Carlos: Associação Brasileira de Estatística*, 131.

Bailey, T. C. & Gatrell, A. C. (1995). *Interactive Spatial Data Analysis*. Prentice Hall.

Bakker, M. I., Hatta, M., Kwenang, A., Faber, W. R., van Beers, S. M., Klatser, P. R., & Oskam, L. (2004). Population survey to determine risk factors for mycobacterium leprae transmission and infection. *International Journal of Epidemiology*, 33(6):1329–1336.

Balieiro, A. A. d. S. (2008). *Detecção de conglomerados dos alertas de desmatamento no estado do Amazonas usando estatística de varredura espaço-temporal*. PhD thesis, Universidade Federal de Viçosa.

Barreto, L. T. (2011). Estimação da temperatura da região amazônica via interpoladores geoestatísticos. Technical report, Departamento de Estatística, Universidade de Brasília.

Besag, J. & Newell, J. (1991). The detection of clusters in rare diseases. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, pages 143–155.

Brent, R. P. (1973). *Algorithms for minimization without derivatives*. Prentice-Hall.

Cançado, A. L., da Silva, C. Q., & da Silva, M. F. (2014). A spatial scan statistic for zero-inflated poisson process. *Environmental and ecological statistics*, 21(4):627–650.

Chambers, J. (2008). *Software for data analysis: programming with R*. Springer Science & Business Media.

Choynowski, M. (1959). Maps based on probabilities. *Journal of the American Statistical Association*, 54(286):385–388.

Christofaroa, C. & Leão, M. M. D. (2014). Tratamento de dados censurados em estudos ambientais. *Quim. Nova*, 37(1):104–110.

Câmara, G., Monteiro, A. M., Fucks, S. D., & Carvalho, M. S. (2002). Análise espacial e geoprocessamento. *Análise espacial de dados geográficos*, 2.

Colosimo, E. & Giolo, S. (2006). Análise de sobrevivência aplicada. In: *ABE-Projeto Fisher*. Edgard Blücher.

Cox, D. R. & Oakes, D. (1984). *Analysis of survival data*, volume 21. CRC Press.

de Matos, P. Z. & Marazotti, D. (2010). Análise de confiabilidade aplicada à industria para estimações de falhas e provisionamento de custos. Technical report, Setor de Ciências Exatas, Universidade Federal do Paraná.

Duczmal, L., Cançado, A. L. F., & Takahashi, R. H. C. (2008). Delineation of irregularly shaped disease clusters through multiobjective optimization. *Journal of Computational and Graphical Statistics*, 17(1):243–262.

Dwass, M. (1957). Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, pages 181–187.

Fernandes, L. B. (2015). Uma estatística scan espacial bayesiana para dados com excesso de zeros. Master's thesis, Universidade de Brasília.

Fernandes, L. B. & Reis, S. D. d. S. (2012). Detecção, identificação e inferência de conglomerados espaciais de fraudes bancárias em uma instituição financeira no centro-oeste do brasil. Technical report, Departamento de Estatística, Universidade de Brasília.

Figueiredo, R. L. (2010). Detecção de clusters usando a estatística scan espacial circular em conjuntos seletivos e um fator de penalização: a ocupação circular. Master's thesis, Universidade Federal de Minas Gerais.

Green, C., Hoppa, R. D., Young, T. K., & Blanchard, J. (2003). Geographic analysis of diabetes prevalence in an urban area. *Social science & medicine*, 57(3):551–560.

Huang, L., Kulldorff, M., & Gregorio, D. (2007). A spatial scan statistic for survival data. *Biometrics*, 63(1):109–118.

Jennings, J. M., Curriero, F. C., Celentano, D., & Ellen, J. M. (2005). Geographic identification of high gonorrhea transmission areas in baltimore, maryland. *American Journal of Epidemiology*, 161(1):73–80.

Jung, I., Kulldorff, M., & Klassen, A. C. (2007). A spatial scan statistic for ordinal data. *Statistics in medicine*, 26(7):1594–1607.

Kleinman, K., Abrams, A., Kulldorff, M., & Platt, R. (2005). A model-adjusted space–time scan statistic with an application to syndromic surveillance. *Epidemiology and Infection*, 133(03):409–419.

Kulldorff, M. (1997). A spatial scan statistic. *Communications in Statistics - Theory and Methods*, 26(6):1481–1496.

Kulldorff, M., Huang, L., & Konty, K. (2009). A scan statistic for continuous data based on the normal probability model. *International Journal of Health Geographics*, 8(1):1.

Kulldorff, M., Mostashari, F., Duczmal, L., Katherine Yih, W., Kleinman, K., & Platt, R. (2007). Multivariate scan statistics for disease surveillance. *Statistics in medicine*, 26(8):1824–1833.

Kulldorff, M. & Nagarwalla, N. (1995). Spatial disease cluster: detection and inference. *Statistics in Medicine*, 14(8):799–810.

Kulldorff, M., Tango, T., & Park, P. J. (2003). Power comparisons for disease clustering tests. *Computational Statistics & Data Analysis*, 42(4):665–684.

Lagarias, J. C., Reeds, J. A., Wright, M. H., & Wright, P. E. (1998). Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147.

Lawless, J. F. (2011). *Statistical models and methods for lifetime data*, volume 362. John Wiley & Sons.

Lima, M. S. (2004). Avaliação do poder do teste da estatística scan para múltiplos clusters. Master's thesis, Universidade Federal de Minas Gerais.

Minamisava, R., Nouer, S. S., de Morais Neto, O. L., Melo, L. K., & Andrade, A. L. S. (2009). Spatial clusters of violent deaths in a newly urbanized region of brazil: highlighting the social disparities. *International journal of health geographics*, 8(1):1.

Moura, F. d. R. (2006). Detecção de clusters espaciais via algoritmo scan multi-objetivo. Master's thesis, Universidade Federal de Minas Gerais.

Naus, J. I. (1965a). Clustering of random points in two dimensions. *Biometrika*, 52(1/2):263–267.

Naus, J. I. (1965b). The distribution of the size of the maximum cluster of points on a line. *Journal of the American Statistical Association*, 60(310):532–538.

Nelder, J. A. & Mead, R. (1965). A simplex method for function minimization. *The computer journal*, 7(4):308–313.

Openshaw, S., Charlton, M., Craft, A. W., & Birch, J. (1988). Investigation of leukaemia clusters by use of a geographical analysis machine. *The Lancet*, 331(8580):272–273.

Pedroso, L. G. & Diniz-Ehrhardt, M. A. (2005). Busca direta em minimização irrestrita. Technical report, UNICAMP, Campinas-SP.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.

Ruggiero, M. A. G. & Lopes, V. L. d. R. (1997). *Cálculo numérico: aspectos teóricos e computacionais*. Makron Books do Brasil.

Saramago, S. P. & Steffen Jr, V. (2008). Introdução às técnicas de otimização em engenharia. *Horizonte científico*, 2(2).

Silva, W. d. J. (2012). Estimação de incertezas no delineamento de clusters espaciais com dados pontuais. Master's thesis, Universidade de Brasília.

Tango, T. (1995). A class of tests for detecting general and focused clustering of rare diseases. *Statistics in Medicine*, 14(21-22):2323–2334.

Turnbull, B. W., Iwano, E. J., Burnett, W. S., HOWE, H. L., & CLARK, L. C. (1990). Monitoring for clusters of disease: application to leukemia incidence in upstate new york. *American Journal of Epidemiology*, 132(supp1):136–143.

Waiselfisz, J. J. (2014). Mapa da violência 2014: homicídios e juventude no brasil: atualização de 15 a 29 anos. *Secretaria Nacional da Juventude*.

Whittemore, A. S., Friend, N., Brown, B. W., & Holly, E. A. (1987). A test to detect clusters of disease. *Biometrika*, 74(3):631–635.

Apêndice

Apêndice A

Programações

A.1 Scan-binomial

```
##################################
## pop = População              ##
## cases = Número de casos      ##
## x = longitude                ##
## y = latitude                 ##
## nsim = número de simulações  ##
##################################
library(plyr)

scan_bin <- function(pop, cases, x, y, nsim=999){

## Cria uma matrix contendo as coordenadas e dados de população e número de casos.
## É criado uma variável ID para questões de ordenação
coords <- cbind(x,y)
id <- seq(1:nrow(coords))
data <- cbind(id,pop,cases,coords)

## Cria a matriz de distâncias
dist_matrix <- as.matrix(dist(coords , diag =TRUE , upper =TRUE ))
dist_matrix <- cbind(dist_matrix,id)

## Cria variáveis auxiliares que serão utilizadas ao longo do processos.
## N= população total
## C= total de casos
## theta = proporção de casos para todo o mapa
## n_regions = número de regiões no mapa
N <- sum(data[,2])
C <- sum(data[,3])
theta <- C/N
n_regions <- nrow(data)

## Neste passo é feito o processo de criação das zonas candidatas para
## encontrar o cluster mais verossímil.
zones <- lapply(1:n_regions,
 function(z){
  dist_matrix <- dist_matrix[order(dist_matrix[,z]),]
```

```
  valid_zones <- sapply(1:n_regions, function(j){
  n_z <- sum( data[ dist_matrix[1:j,n_regions+1], 2] )
  return(n_z < 0.25*N) })

  partial_zones <- lapply(1:sum(valid_zones),
   function(y){
        n_z <- sum( data[ dist_matrix[1:y,n_regions+1],2] )
        x_z <- sum( data[ dist_matrix[1:y,n_regions+1],3] )
    x_z_bar <- C - x_z
        n_z_bar <- N - n_z
        theta_0 <- x_z_bar/n_z_bar
        theta_1 <- x_z/n_z
        ifelse(x_z > n_z*(C/N),
         llr <- x_z*log(theta_1)+(n_z - x_z)*log(1- theta_1)+
         x_z_bar*log(theta_0)+(n_z_bar - x_z_bar)*log(1- theta_0)-
         C*log(theta)-(N-C)*log(1 - theta),
         llr <- 0)
        return(data.frame(zone= paste(sort(as.numeric(dist_matrix[1:y,n_regions+1])), collapse=",
            "),
        llr=round(llr,5),
        nzone=n_z,
        xzone=x_z,
        expected_cases=n_z*C/N))
        }
   )
  partial_zones <- rbind.fill(partial_zones)
  return(partial_zones)
  })

zones <- rbind.fill(zones)
zones <- unique(zones[order(zones[,2],decreasing=TRUE),])
row.names(zones) <- NULL
clustera <- as.numeric(unlist(strsplit(as.character(zones[1,1]), ",")))

##### Simulação de Monte Carlo para realização do teste de hipósteses.

## Simulando o mapa segundo uma multinomial com parâmatros C e
## p=proporção da população total em cada área.
sim_cases <- rmultinom(nsim, size = C, prob = data[,2]/N)
sim_cases <- cbind(id,sim_cases)

## O processo de cálculo da razão de verossimilhança para todas as zonas candidatas (zonas que
     atendem o critério da população) e identificação
## do cluster mais verissímil é repetido para cada simulação do mapa feito, ou seja, o processo é
     repetido n_sim vezes.
## O resultado é a obtenção de um vetor com a distribuição empírica da estatística do teste T.
empirical_t <- sapply(1:nsim, function(p){

 partial_t <- sapply(1:nrow(zones), function(w){
  index <- as.numeric(unlist(strsplit(as.character(zones[w,1]), ",")))
  n_z <- sum( data[ index,2] )
  x_z <- sum( sim_cases[ index,p+1] )
  x_z_bar <- C - x_z
  n_z_bar <- N - n_z
  theta_0 <- x_z_bar/n_z_bar
  theta_1 <- x_z/n_z
```

```
       ifelse(x_z > n_z*(C/N),
        llr <- x_z*log(theta_1)+(n_z - x_z)*log(1- theta_1)+
        x_z_bar*log(theta_0)+(n_z_bar - x_z_bar)*log(1- theta_0)-
        C*log(theta)-(N-C)*log(1 - theta),
        llr <- 0)
       return(llr)
   })
max_t <- max(partial_t, na.rm=TRUE)
return(max_t)
})

z_95 <- quantile (empirical_t ,0.95)
if( zones$llr[1] > z_95){ signif <-" TRUE "} else { signif <-" FALSE "}
output <-list (t=zones$llr[1], cluster_population = zones$nzone[1] ,
cluster_size = length ( clustera ),
expected_cases = zones$expected_cases[1] , observed_cases = zones$xzone[1] ,
Significance =signif , cluster = clustera)

plot1 <-plot(x,y, main =" Scan - Binomial ")
pt <- points (x=x[clustera],y=y[clustera], col = "red", pch =21 , cex = 2)
return ( list (output ,plot1 , pt ))
}
```

A.2 Scan-binomial$_{\text{censored}}$

```
####################################
## pop = População              ##
## cases = Número de casos      ##
## x = longitude                ##
## y = latitude                 ##
## nsim = número de simulações ##
## cens = censura                          ##
####################################
library(plyr)

scan_bin_censored <- function(pop, cases, x, y, nsim=999,cens){

## Cria uma matrix contendo as coordenadas e dados de população e número de casos.
## È criado uma variável ID para questões de ordenação
coords <- cbind(x,y)
id <- seq(1:nrow(coords))
data <- cbind(id,pop,cases,coords,cens)

## Cria a matriz de distâncias
dist_matrix <- as.matrix(dist(coords , diag =TRUE , upper =TRUE ))
dist_matrix <- cbind(dist_matrix,id)

## Cria variáveis auxiliares que serão utilizadas ao longo do processos.
## N= população total
## C= total de casos
## theta = proporção de casos para todo o mapa
## n_regions = número de regiões no mapa
N <- sum(data[,2])
C <- sum(data[,3])
```

```
theta <- C/N
n_regions <- nrow(data)


## Definindo a log-verossimilhança sob H0 para ser maximizado (a função é definida negativa pois o
    optim minimiza)
loglikehood <- function(p,n,xi,censor) {
-( sum((1-censor)*dbinom(xi,prob=p,size=n,log=TRUE) )+
sum(censor*pbinom(xi, prob=p, size=n, log=TRUE,lower.tail=FALSE)) )
}
## Estimando theta via método de Brent (proporção de casos para todo o mapa)
maxi <- try( optim(p=theta,loglikehood,n=data[,2],xi=data[,3], censor=cens, method='Brent',
    lower=0.001,upper=0.999), silent=TRUE )
if ( inherits(maxi ,"try-error") ){ maxi <- try( optim(p=0.001,loglikehood,n=data[,2],xi=data[,3],
    censor=cens, method='Brent', lower=0.001,upper=0.999), silent=TRUE ) }
theta_maxi <- maxi$par ##theta sob H0 obtido via maximização

## Definindo a log-verossimilhança sob H0 para depois ser utilizada no cálculo da razão de
    verossimilhança
l_0 <- function(p,n,xi,censor) {
 l0 = sum( (1-censor)*dbinom(xi,prob=p,size=n,log=TRUE) ) +
 sum( censor*pbinom(xi, prob=p, size=n, log=TRUE,lower.tail=FALSE) )
 return(l0)
}

## Definindo a log-verossimilhança sob Ha para ser maximizado
loglikehood_ha <- function(p, n_zbar, xi_zbar, n_z, xi_z, censor_zbar, censor_z) {
-(
sum( (1-censor_zbar)*dbinom(xi_zbar,prob=p[1],size=n_zbar,log=TRUE) ) + sum(
    censor_zbar*pbinom(xi_zbar, prob=p[1], size=n_zbar, log=TRUE,lower.tail=FALSE) )+
sum( (1-censor_z)*dbinom(xi_z,prob=p[2],size=n_z,log=TRUE) ) + sum( censor_z*pbinom(xi_z,
    prob=p[2], size=n_z, log=TRUE,lower.tail=FALSE) )
)
}

## Definindo a log-verossimilhança sob Ha para depois ser utilizada no cálculo da razão de
    verossimilhança
l_a <- function(p0,pz,n_zbar,xi_zbar, n_z, xi_z, censor_zbar, censor_z) {
 la = sum( (1-censor_zbar)*dbinom(xi_zbar,prob=p0,size=n_zbar,log=TRUE) ) +
         sum( censor_zbar*pbinom(xi_zbar, prob=p0, size=n_zbar, log=TRUE,lower.tail=FALSE) )+
      sum( (1-censor_z)*dbinom(xi_z,prob=pz,size=n_z,log=TRUE) ) +
         sum( censor_z*pbinom(xi_z, prob=pz, size=n_z, log=TRUE,lower.tail=FALSE) )
 return(la)
}

## Neste passo é feito o processo de criação das zonas candidatas para encontrar o cluster mais
    verossímil.
zones <- lapply(1:n_regions,
 function(z){
  dist_matrix <- dist_matrix[order(dist_matrix[,z]),]
  valid_zones <- sapply(1:n_regions, function(j){
  n_z <- sum( data[ dist_matrix[1:j,n_regions+1], 2] )
  return(n_z < 0.25*N) })

  partial_zones <- lapply(1:sum(valid_zones),
   function(y){
```

```
      n_z <- sum( data[ dist_matrix[1:y,n_regions+1],2] )
      x_z <- sum( data[ dist_matrix[1:y,n_regions+1],3] )
      x_z_bar <- C - x_z
      n_z_bar <- N - n_z
      theta_0 <- x_z_bar/n_z_bar
      theta_1 <- x_z/n_z

      maxi_ha <- try( optim(p=c(theta_0,theta_1),loglikehood_ha,
       n_zbar=data[ dist_matrix[-(1:y),n_regions+1],2],
       xi_zbar=data[ dist_matrix[-(1:y),n_regions+1],3],
       censor_zbar=cens[dist_matrix[-(1:y),n_regions+1]],
       n_z=data[ dist_matrix[1:y,n_regions+1],2],
       xi_z=data[ dist_matrix[1:y,n_regions+1],3],
       censor_z=cens[dist_matrix[1:y,n_regions+1]]
       ), silent=TRUE)

      if ( !inherits(maxi_ha ,"try-error") ) {
       theta0_maxi <- maxi_ha$par[1]
       theta1_maxi <- maxi_ha$par[2]
      }else{
       maxi_ha <- try( optim(p=c(0.001,0.001),loglikehood_ha,
        n_zbar=data[ dist_matrix[-(1:y),n_regions+1],2],
        xi_zbar=data[ dist_matrix[-(1:y),n_regions+1],3],
        censor_zbar=cens[dist_matrix[-(1:y),n_regions+1]],
        n_z=data[ dist_matrix[1:y,n_regions+1],2],
        xi_z=data[ dist_matrix[1:y,n_regions+1],3],
        censor_z=cens[dist_matrix[1:y,n_regions+1]]
        ), silent=TRUE)
      if ( !inherits(maxi_ha ,"try-error") ){
       theta0_maxi <- maxi_ha$par[1]
       theta1_maxi <- maxi_ha$par[2]
      }else{
       theta0_maxi <- NA
       theta1_maxi <- NA
      }
      }

      ifelse( theta1_maxi > theta_maxi & theta1_maxi>theta_1,
       llr <- l_a(p0=theta0_maxi,pz=theta1_maxi,
        n_zbar=data[ dist_matrix[-(1:y),n_regions+1],2],
        xi_zbar=data[ dist_matrix[-(1:y),n_regions+1],3],
        censor_zbar=cens[dist_matrix[-(1:y),n_regions+1]],
        n_z=data[ dist_matrix[1:y,n_regions+1],2],
        xi_z=data[ dist_matrix[1:y,n_regions+1],3],
        censor_z=cens[dist_matrix[1:y,n_regions+1]]) -
            l_0(p=theta_maxi,n=data[,2],xi=data[,3],censor=cens),
       llr <- 0)

      return(data.frame(zone= paste(sort(as.numeric(dist_matrix[1:y,n_regions+1])), collapse=", "),
      llr=round(llr,5),
      nzone=n_z,
      xzone = x_z,
      xzone_estimated=n_z*theta1_maxi,
      expected_cases=n_z*theta_maxi,
      ))
}
```

```
  )
  partial_zones <- rbind.fill(partial_zones)
  return(partial_zones)
 })

zones <- rbind.fill(zones)
zones <- zones[order(zones[,2],decreasing=TRUE),]
zones <- zones[!duplicated(zones[,1]),]
zones <- zones[order(zones[,2],decreasing=TRUE),]
row.names(zones) <- NULL
clustera <- as.numeric(unlist(strsplit(as.character(zones[1,1]), ",")))

##### Simulação de Monte Carlo para realização do teste de hipósteses.

sim_cases <- cbind(id,matrix(0,ncol=nsim,nrow=n_regions))
censor_index <- which(data[,6]==1)
N_noCensor <- sum(data[-censor_index,2])
C_noCensor <- sum(data[-censor_index,3])

for(k in 1:nsim){
sim_cases[ censor_index, k+1] <- data[censor_index,3]
sim_cases[ -censor_index, k+1] <- rmultinom(1, size = C_noCensor, prob =
     data[-censor_index,2]/N_noCensor)
}

## O processo de cálculo da razão de verossimilhança para todas as zonas candidatas (zonas que
     atendem o critério da população) e identificação
## do cluster mais verissímil é repetido para cada simulação do mapa feito, ou seja, o processo é
     repetido n_sim vezes.
## O resultado é a obtenção de um vetor com a distribuição empírica da estatística do teste T.
empirical_t <- sapply(1:nsim, function(p){

 partial_t <- sapply(1:nrow(zones), function(w){
  index <- as.numeric(unlist(strsplit(as.character(zones[w,1]), ",")))
  n_z <- sum( data[ index,2] )
  x_z <- sum( sim_cases[ index,p+1] )
  x_z_bar <- C - x_z
  n_z_bar <- N - n_z
  theta_0 <- x_z_bar/n_z_bar
  theta_1 <- x_z/n_z

  maxi_ha <- try( optim(p=c(theta_0,theta_1),loglikehood_ha,
        n_zbar=data[ -index,2 ],
        xi_zbar=sim_cases[ -index,p+1],
        censor_zbar=cens[-index],
        n_z=data[ index,2] ,
        xi_z=sim_cases[ index,p+1],
        censor_z=cens[index]
        ), silent=TRUE )

  if ( !inherits(maxi_ha ,"try-error") ) {
        theta0_maxi <- maxi_ha$par[1]
        theta1_maxi <- maxi_ha$par[2]
  }else{
        maxi_ha <- try( optim(p=c(0.001,0.001),loglikehood_ha,
               n_zbar=data[ -index,2 ],
```

```
              xi_zbar=sim_cases[ -index,p+1],
              censor_zbar=cens[-index],
              n_z=data[ index,2] ,
              xi_z=sim_cases[ index,p+1],
              censor_z=cens[index]
              ), silent=TRUE)
              if ( !inherits(maxi_ha ,"try-error") ){
               theta0_maxi <- maxi_ha$par[1]
               theta1_maxi <- maxi_ha$par[2]
              }else{
               theta0_maxi <- NA
               theta1_maxi <- NA
              }
 }

 ifelse( theta1_maxi > theta_maxi & theta1_maxi>theta_1,
       llr <- l_a(p0=theta0_maxi,pz=theta1_maxi,
        n_zbar=data[ -index,2 ],
        xi_zbar=sim_cases[ -index,p+1],
        censor_zbar=cens[-index],
        n_z=data[ index,2],
        xi_z=sim_cases[ index,p+1],
        censor_z=cens[index]) - l_0(p=theta_maxi,n=data[,2],xi=sim_cases[,p+1],censor=cens),
       llr <- 0)
 return(llr)
})

 max_t <- max(partial_t, na.rm=TRUE)
 return(max_t)
})

z_95 <- quantile (empirical_t ,0.95)
if( zones$llr[1] > z_95){ signif <-" TRUE "} else { signif <-" FALSE "}
output <-list (t=zones$llr[1], cluster_population = zones$nzone[1] ,
cluster_size = length ( clustera ),
expected_cases = zones$expected_cases[1] , observed_cases = zones$xzone[1] ,
estimated_cases= zones$xzone_estimated[1]
Significance =signif , cluster = clustera)

plot1 <-plot(x,y, main =" Scan - Binomial(censored) ")
pt <- points (x=x[clustera],y=y[clustera], col = "red", pch =21 , cex = 2)
return ( list (output ,plot1 , pt ))
}
```

Apêndice B

Tabelas com informações do banco de dados

Tabela B.1: Códigos do CID-10 selecionados na construção da base de homicídios.

Categoria	Descrição
X85	Agressão por meio de drogas, medicamentos e substâncias biológicas
X86	Agressão por meio de substâncias corrosivas
X87	Agressão por pesticidas
X88	Agressão por meio de gases e vapores
X89	Agressão por meio de outros produtos químicos e substâncias nocivas especificados
X90	Agressão por meio de produtos químicos e substâncias nocivas não especificados
X91	Agressão por meio de enforcamento, estrangulamento e sufocação
X92	Agressão por meio de afogamento e submersão
X93	Agressão por meio de disparo de arma de fogo de mão
X94	Agressão por meio de disparo de espingarda, carabina ou arma de fogo de maior calibre
X95	Agressão por meio de disparo de outra arma de fogo ou de arma não especificada
X96	Agressão por meio de material explosivo
X97	Agressão por meio de fumaça, fogo e chamas
X98	Agressão por meio de vapor de água, gases ou objetos quentes
X99	Agressão por meio de objeto cortante ou penetrante
Y00	Agressão por meio de um objeto contundente
Y01	Agressão por meio de projeção de um lugar elevado
Y02	Agressão por meio de projeção ou colocação da vítima diante de um objeto em movimento
Y03	Agressão por meio de impacto de um veículo a motor
Y04	Agressão por meio de força corporal
Y05	Agressão sexual por meio de força física
Y06	Negligência e abandono
Y07	Outras síndromes de maus tratos
Y08	Agressão por outros meios especificados
Y09	Agressão por meios não especificados

Tabela B.2: Banco de dados de homicídios para os municípios do RJ - 2014.

Município	População	Homicídios	δ_i	Homicídios censurado
Rio de Janeiro	6453682	1386	0	1386
Duque de Caxias	878402	576	0	576
Nova Iguaçu	806177	560	1	5
São Gonçalo	1031903	382	1	5
Belford Roxo	479386	237	1	5
Campos dos Goytacazes	480648	225	0	225
São João de Meriti	460711	187	1	5

Continua na próxima página

Tabela B.2 – *Continuação da página anterior*

Município	População	Homicídios	δ_i	Homicídios censurado
Cabo Frio	204486	174	0	174
Macaé	229624	120	1	5
Magé	233634	106	1	5
Queimados	142709	92	0	92
Angra dos Reis	184940	89	0	89
Itaboraí	227168	85	1	5
Niterói	495470	71	1	5
Itaguaí	117374	66	1	5
Volta Redonda	262259	63	0	63
Nilópolis	158299	61	0	61
Seropédica	82090	59	0	59
Maricá	143111	59	0	59
Mesquita	170473	55	1	5
Japeri	99141	53	1	5
Araruama	120948	51	0	51
Rio das Ostras	127171	49	0	49
São Pedro da Aldeia	95318	43	0	43
Barra Mansa	179697	38	1	5
Resende	124316	33	0	33
Paraty	39965	30	0	30
Guapimirim	55626	25	1	5
Mangaratiba	40008	24	0	24
Petrópolis	298017	23	0	23
Nova Friburgo	184460	23	0	23
São Francisco de Itabapoana	41343	23	1	5
Barra do Piraí	96568	22	0	22
Saquarema	80915	22	0	22
Armação dos Búzios	30439	20	0	20
Itaperuna	98521	19	1	5
Teresópolis	171482	18	0	18
São João da Barra	34273	17	0	17
Paracambi	49120	14	0	14
Cachoeiras de Macacu	55967	13	1	5
Três Rios	78998	12	0	12
Casimiro de Abreu	39414	12	0	12
Conceição de Macabu	22006	11	1	5
Arraial do Cabo	28866	10	0	10
Valença	73445	9	0	9
Miguel Pereira	24829	9	0	9
Rio Bonito	57284	9	0	9
Cantagalo	19792	9	0	9
Santo Antônio de Pádua	41108	9	0	9
Itatiaia	29996	8	0	8
Itaocara	22824	8	0	8
Paraíba do Sul	42159	7	0	7
Vassouras	35275	6	0	6
Quissamã	22261	6	0	6
Porto Real	17970	5	0	5
Quatis	13415	5	0	5
Mendes	18086	5	0	5
Iguaba Grande	25354	5	0	5
Carapebus	14713	5	0	5
Paty do Alferes	26758	4	0	4
Carmo	18074	4	0	4

Continua na próxima página

Tabela B.2 – *Continuação da página anterior*

Município	População	Homicídios	δ_i	Homicídios censurado
Miracema	26724	4	0	4
Rio Claro	17768	3	0	3
Silva Jardim	21336	3	0	3
Aperibé	10882	3	0	3
São Sebastião do Alto	9033	3	0	3
Cardoso Moreira	12578	3	0	3
Pinheiral	23691	2	0	2
Sapucaia	17608	2	0	2
Duas Barras	11096	2	0	2
Bom Jardim	26126	2	0	2
Cordeiro	20965	2	0	2
Cambuci	14849	2	0	2
São Fidélis	37710	2	0	2
Rio das Flores	8838	1	0	1
São José do Vale do Rio Preto	20812	1	0	1
Sumidouro	15099	1	0	1
Macuco	5380	1	0	1
Trajano de Moraes	10348	1	0	1
Laje do Muriaé	7341	1	0	1
São José de Ubá	7175	1	0	1
Bom Jesus do Itabapoana	35896	1	0	1
Piraí	27579	0	0	0
Engenheiro Paulo de Frontin	13566	0	0	0
Comendador Levy Gasparian	8245	0	0	0
Areal	11879	0	0	0
Tanguá	32140	0	0	0
Porciúncula	18293	0	0	0
Natividade	15040	0	0	0
Santa Maria Madalena	10253	0	0	0
Varre-Sai	9966	0	0	0
Italva	14489	0	0	0