



**Cursos Integrados  
em Vigilância em Saúde**

*Curso* —

**Estatística espacial aplicada  
à vigilância em saúde**

## **UNIVERSIDADE FEDERAL DE SANTA CATARINA**

Reitor Irineu Manoel de Souza

Vice-Reitora Joana Célia dos Passos

Pró-Reitora de Pós-graduação Werner Kraus

Pró-Reitor de Pesquisa e Inovação Jacques Mick

Pró-Reitor de Extensão Olga Regina Zigelli Garcia

## **CENTRO DE CIÊNCIAS DA SAÚDE**

Diretor Fabrício de Souza Neves

Vice-Diretora Rodrigo Otávio Moretti Pires

## **DEPARTAMENTO DE SAÚDE PÚBLICA**

Chefe do Departamento Sheila Rúbia Lindner

Subchefe do Departamento Maria Cristina Marino Calvo

## **INSTITUTO TODOS PELA SAÚDE (ITPS)**

Diretor-presidente Gerson Oliveira Penna

## **ASSOCIAÇÃO BRASILEIRA DE SAÚDE COLETIVA (ABRASCO)**

Presidente Rômulo Paes de Sousa

## **EQUIPE DE PRODUÇÃO**

Denis de Oliveira Rodrigues

Wagner de Souza Tassinari

João Henrique de Araujo Morais

Laís Picinini Freitas

Oswaldo Gonçalves Cruz



Cursos Integrados  
em Vigilância em Saúde

Estatística espacial aplicada  
à vigilância em saúde



*Curso* \_\_\_\_\_

**Estatística espacial aplicada  
à vigilância em saúde**



UFPEL



**Dados Internacionais de Catalogação na Publicação (CIP)  
(Câmara Brasileira do Livro, SP, Brasil)**

Estatística espacial aplicada à vigilância em saúde [livro eletrônico] / Denis de Oliveira Rodrigues ... [et al.] ; organização Alexandra Crispim da Silva Boing, Antonio Fernando Boing. -- 1. ed. -- Florianópolis, SC : Associação Brasileira de Saúde Coletiva, 2025. PDF

Outros autores: Wagner de Souza Tassinari, João Henrique de Araujo Morais, Lais Picinini Freitas, Oswaldo Gonçalves Cruz  
Bibliografia  
ISBN 978-85-85740-13-9

1. Estatística - Métodos 2. Saúde pública  
3. Vigilância epidemiológica 4. Vigilância em saúde  
I. Rodrigues, Denis de Oliveira. II. Tassinari,  
Wagner de Souza. III. Morais, João Henrique de  
Araujo.  
IV. Freitas, Lais Picinini. V. Cruz, Oswaldo  
Gonçalves. VI. Boing, Alexandra Crispim da Silva.  
VII. Boing, Antonio Fernando.

25-308623.0

CDD-362.1068

**Índices para catálogo sistemático:**

1. Vigilância em saúde pública : Bem-estar social  
362.1068

Maria Alice Ferreira - Bibliotecária - CRB-8/7964

# Sumário

Estatística espacial aplicada à vigilância em saúde .....	7
<b>Módulo 1: Introdução a estatística espacial.....</b>	<b>8</b>
Origem da análise estatística espacial.....	9
Quando usar métodos de análise espacial?.....	13
Principais conceitos e aplicações em Saúde.....	15
Tipologia dos dados espaciais .....	16
Dependência ou autocorrelação espacial.....	21
Estacionariedade espacial .....	22
Isotropia e anisotropia .....	23
<b>Módulo 2: Padrões Pontuais .....</b>	<b>25</b>
O que são padrões pontuais? .....	26
Por que é importante? .....	28
Padrões de distribuição pontual.....	30
Processos pontuais .....	31
Simulação de padrões pontuais e contagem em quadrantes .....	33
Estimativa de densidade de Kernel (mapa de calor) .....	36
Escolhendo a largura de banda.....	39
Estimativa de Kernel com correção por bordas .....	45
Diferenças entre as funções de Kernel.....	49
Kernel por atributo (kernel ponderado). ....	53
Razão de Kernel.....	54
Análise de um processo pontual de segunda ordem (funções G e K) .....	58
Função G - Distância do vizinho mais próximo.....	58
Função K de Ripley (ou apenas função K) .....	61
Padrões Espaciais Detectados pela Função.....	63
Detecção de cluster .....	64
Testes genéricos de detecção de clusters.....	65
Testes focados de detecção de clusters .....	66
Pratica em R .....	66
Baixando e preparando os dados.....	67
Testando a Completa Aleatoriedade Espacial (CSR) .....	81
Processo de primeira ordem: Gerando a estimativa de densidade de Kernel (mapa de calor) .....	84
Processo de segunda ordem: Funções G e K.....	90
<b>Módulo 3: Dados de Área .....</b>	<b>94</b>
Criando mapas.....	94
Pontos de corte em mapas .....	95
Definindo pontos de corte .....	97
Quebras arbitrárias .....	98
Quebras regulares .....	99
Quebras quantílicas.....	100

Quebras naturais, ou quebras de Jenks .....	101
Comparando mapas no tempo .....	102
Problemas conhecidos ao lidar com dados de área .....	105
Tamanho heterogêneo das áreas.....	105
Problema da área modificável (MAUP) .....	108
Gerrymandering .....	110
Prática em R: Criação de mapas temáticos .....	111
Malhas geográficas com geobr .....	114
Autocorrelação espacial .....	125
Efeitos de primeira ordem - tendência .....	126
Efeitos de segunda ordem - dependência local .....	127
Medidas de proximidade em dados de área.....	130
Matriz de vizinhança.....	131
Por contiguidade .....	133
Por vizinhos mais próximos .....	141
Por distância.....	143
Testando a autocorrelação espacial global .....	151
Índice I de Moran.....	152
Índice C de Geary.....	155
Testando a autocorrelação espacial local.....	156
Indicadores Locais de Associação Espacial .....	157
Suavização espacial .....	163
Kernel de área.....	164
Método Bayesiano Empírico .....	176
Principais modelos de regressão espacial para dados de área.....	181
<b>Módulo 4: Geoestatística .....</b>	<b>184</b>
Conceitos e objetivos.....	185
Padrões espaciais: efeitos de primeira e segunda ordem.....	188
Efeito de primeira ordem .....	188
Efeito de segunda ordem .....	189
Análise exploratória do efeito de primeira ordem .....	190
Análise exploratória do efeito de segunda ordem .....	193
Variograma, covariograma e correlograma .....	195
Estrutura do variograma .....	199
Variogramas para modelos isotrópicos .....	200
Algumas aplicações da geoestatística .....	203
Aplicação 1 .....	203
Aplicação 2 .....	205
Modelagem em geoestatística .....	206
Krigagem .....	207
Krigagem Universal .....	208
Considerações gerais sobre a Krigagem: .....	209
Exemplo de aplicação.....	210
Prática em R .....	212
<b>Módulo 5: Dados espaço-temporais .....</b>	<b>223</b>
Análise exploratória espaço-temporal .....	224
Detecção de clusters espaço-temporais de doenças .....	228
A estatística scan (SaTScan) .....	229
Exemplos de aplicação.....	231
Modelagem estatística espaço-temporal.....	238

# Estatística espacial aplicada à vigilância em saúde

## Introdução ao curso

Nesse **curso** você vai aprender a realizar análises espaciais com aplicações para a vigilância em saúde usando R. Em epidemiologia, a análise descritiva dos dados frequentemente se estrutura em torno de três elementos fundamentais: pessoa, tempo e lugar. O elemento “pessoa” diz respeito às características individuais dos afetados pelo evento de saúde, incluindo fatores como idade, sexo, raça/cor, entre outras. O elemento “tempo” considera o período em que o evento de saúde ocorre, já o elemento “lugar” refere-se à localização geográfica, como o surgimento de um caso de doença em uma área específica.

Compreender a distribuição espacial desses eventos é crucial, pois permite uma melhor compreensão dos fenômenos que se manifestam de maneiras distintas em diferentes locais. A análise espacial pode variar desde simples visualizações em mapas, que auxiliam na identificação de padrões geográficos, até a aplicação de métodos estatísticos mais avançados, que consideram a localização como um fator essencial na interpretação dos dados.

Ao longo de cinco módulos, você vai desde conhecer os conceitos-chave da análise espacial e sua origem, até ser capaz de realizar análises para os diferentes tipos de dados espaciais. Vamos começar?

### Ao final deste curso, você será capaz de:

1. Compreender os conceitos-chave da análise espacial e sua origem histórica.
2. Identificar quando utilizar métodos de análise espacial em saúde.
3. Diferenciar os tipos de dados espaciais e suas aplicações.
4. Entender e aplicar conceitos de dependência e autocorrelação espacial.
5. Reconhecer a importância da estacionariedade, isotropia e anisotropia na análise espacial.
6. Aplicar métodos de análise espacial para cada tipo de dado: pontuais, de área e de geoestatística.

### Atenção!



Para acompanhar este curso de maneira mais fluida e proveitosa, é essencial que você tenha familiaridade com as ferramentas básicas da linguagem R e o RStudio, além de conhecimentos prévios sobre rotinas de análise e visualização de dados.

Se ainda não teve contato com esses tópicos, recomendamos que realize o curso “Análise de dados para a Vigilância em Saúde”. O material pode ser acessado clicando [neste link](#). Nele, você encontrará códigos e orientações que facilitarão a construção de seus mapas e demais análises espaciais.

Essa leitura prévia ajudará a contextualizar os conceitos que abordaremos e permitirá um melhor aproveitamento das discussões e aplicações práticas.

## Módulo 1: Introdução a estatística espacial

A “análise estatística espacial” é definida quando os dados estão espacialmente localizados e se considera explicitamente a importância de seu arranjo espacial na análise ou interpretação dos resultados (Bailey & Gatrell, 1995). Isso significa que, ao analisar dados como incidências de doenças, contaminações ambientais ou qualquer outro evento de saúde pública, deve-se levar em conta o local onde esses eventos ocorrem para identificar padrões, verificar a existência de agrupamentos e orientar intervenções mais eficazes.

A principal característica da análise estatística espacial é que **a localização geográfica não é apenas um detalhe, mas um componente central da análise**. Em outras palavras, a geografia dos dados é utilizada explicitamente na análise e na interpretação, oferecendo uma abordagem que os métodos estatísticos tradicionais não contemplam. Portanto, os profissionais que atuam na Vigilância em Saúde precisam avaliar se os eventos observados poderiam apresentar desfechos diferentes dependendo da localização em que ocorrem.



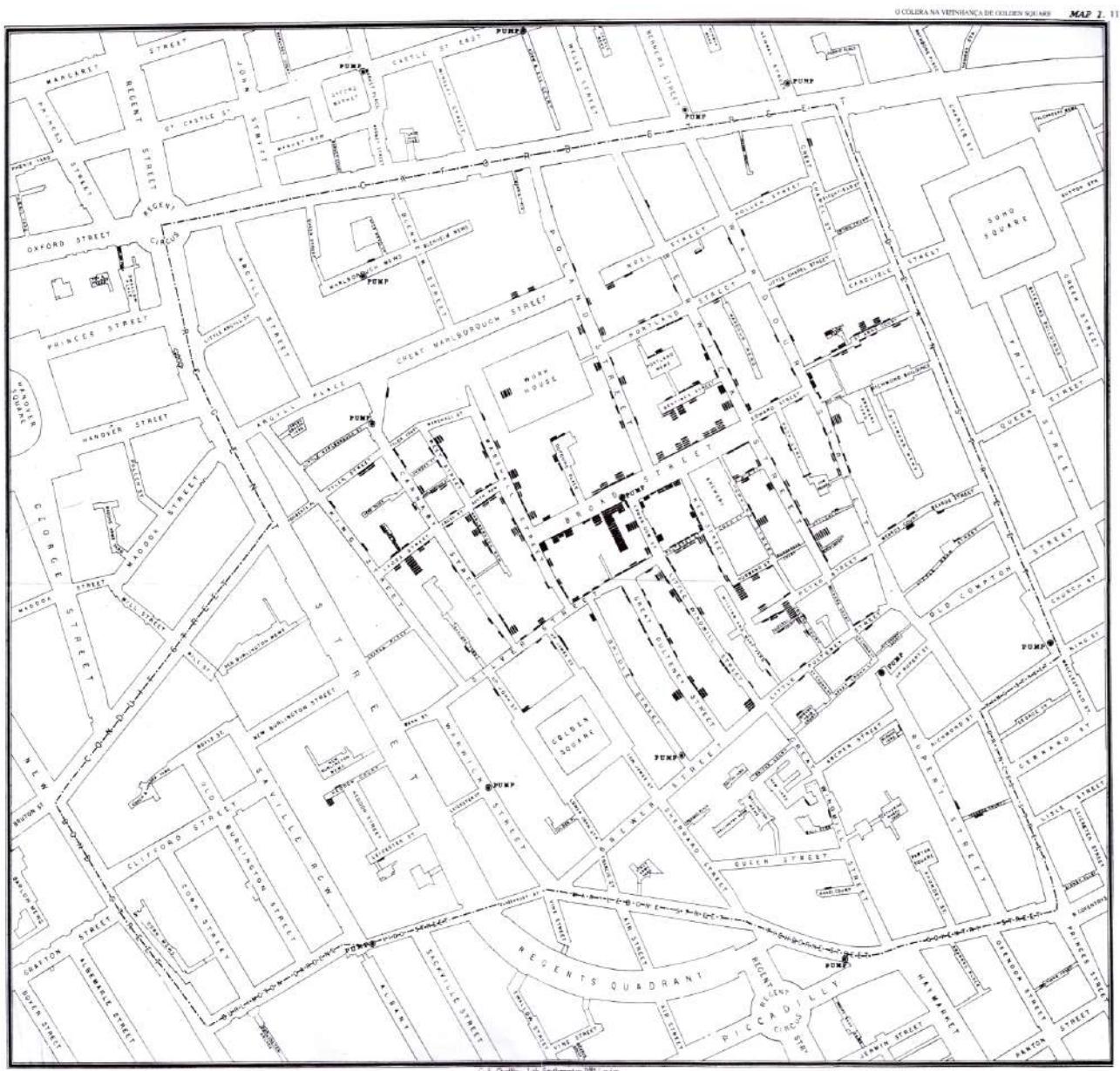
Neste curso, serão abordadas essencialmente as técnicas estatísticas de análise espacial. Quando nos referimos ao espaço, estamos tratando da representação digital de fenômenos que ocorrem em locais específicos, e não do espaço enquanto conceito abstrato da geografia. Além disso, diversas operações realizadas utilizando Sistemas de Informação Geográfica (SIG), também chamadas de análise espacial, não serão abordadas neste material.

## *Origem da análise estatística espacial*

O uso de dados espaciais para o mapeamento de doenças teve um grande marco na era moderna com o trabalho realizado por John Snow em 1854, no Reino Unido. Naquela época, Londres enfrentava um grave surto de cólera que estava causando muitas mortes. A teoria dominante sugeria que a doença se espalhava pelo ar contaminado, conhecida como teoria miasmática.

No entanto, John Snow suspeitava que a cólera era transmitida pela água contaminada. Para testar sua hipótese, ele mapeou cuidadosamente as residências das pessoas que faleceram devido à doença no bairro do Soho (Figura 1). Ao plotar os casos em um mapa, ele percebeu um padrão claro: a maioria das mortes estava concentrada ao redor da bomba de água localizada na Broad Street.

**Figura 1: Mapeamento dos casos de cólera por John Snow em Londres, Reino Unido.**



Fonte: SNOW, J. (1854). On the mode of communication of cholera. John Churchill.

Observando que as mortes diminuíam à medida que a distância das residências em relação à bomba aumentava, Snow concluiu que a água daquela bomba era a fonte de contaminação. Sua investigação levou as autoridades a removerem a alavanca da bomba, o que resultou em uma drástica redução nos casos de cólera na região (SNOW, 1854).

O trabalho de John Snow foi pioneiro ao utilizar a análise espacial para identificar a fonte de uma epidemia. Sua abordagem inovadora não só salvou vidas naquela ocasião, mas também estabeleceu fundamentos importantes para a epidemiologia moderna e para as técnicas de análise espacial utilizadas hoje na Vigilância em Saúde. Ele demonstrou que a localização geográfica dos casos pode revelar padrões cruciais para a compreensão e o controle de doenças.



Você sabia que o pacote `cholera` do R foi inspirado no surto de cólera em Londres em 1854? Este pacote oferece um conjunto de dados e funções que permitem:

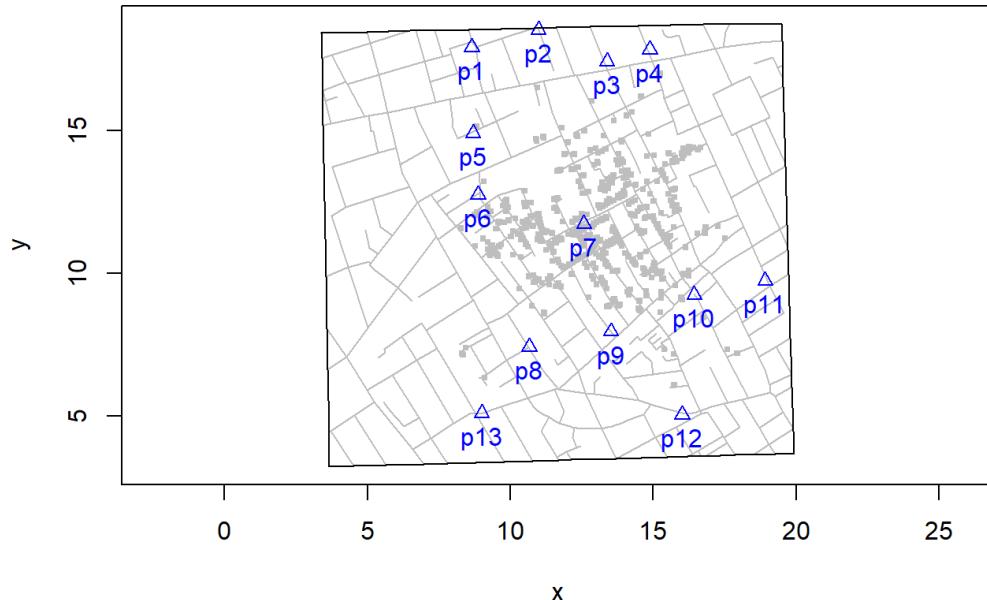
- Visualizar os dados históricos;
- Analisar os padrões espaciais da hipótese do John Snow;
- Ter uma imersão utilizando dados de um estudo clássico.

```
# Para instalar o pacote no seu computador
install.packages("cholera")

# Para carregar o pacote
library(cholera)

# Para visualizar os dados
# Os comandos abaixo adicionam os dados ao ambiente de trabalho
head(fatalities)
head(pumps)

# Para visualizar o mapa
snowMap()
```



Este mapa exibe a distribuição das mortes por cólera (pequenos pontos cinzas) e a localização das bombas de água (triângulos azuis numerados). A visualização é baseada na análise de *John Snow*, que identificou a contaminação da água como a causa do surto. O comando `snowMap()` plota o mapa sobre uma malha de ruas da época.

Para saber mais sobre o pacote `cholera`, acesse o [link](#).

## *Quando usar métodos de análise espacial?*

Como mencionado anteriormente, a localização de onde ocorrem os fenômenos de saúde é de grande importância para a Vigilância e, dessa forma, tem implicações práticas. Para o desenvolvimento adequado das ações de vigilância, uma questão fundamental se apresenta, antes da aplicação de qualquer método estatístico:

**A distribuição dos dados apresenta um algum tipo de padrão ou esses dados estão distribuídos de forma regular no espaço?**

Vamos pensar em situações que envolvem como o nosso território expressa as desigualdades sociais, econômicas e ambientais e como a exposição a fatores de forma diferente agem sobre a saúde da população. A distribuição de recursos, a acessibilidade aos serviços de saúde, a infraestrutura urbana, entre outros fatores, são exemplos de como a localização geográfica pode influenciar a saúde da população. Portanto, a análise espacial é uma ferramenta essencial para identificar padrões de distribuição espacial, detectar áreas de risco e orientar ações de prevenção e controle de doenças.

Neste sentido, um conceito importante a ser considerado é a **heterogeneidade espacial**, que se refere ao fato de que a magnitude e a direção de um fenômeno de interesse podem variar no espaço. Em outras palavras, os eventos geralmente não estão distribuídos de forma uniforme e essa variabilidade espacial pode influenciar diretamente os resultados de uma análise.

Considere o seguinte: ao realizar uma análise espacial, você pode perceber que alguns casos podem estar relacionados. Por exemplo, se uma pessoa fica doente em um bairro, é possível que outras pessoas próximas também fiquem. Isso significa que os casos não são independentes entre si. Estamos, então, lidando com uma **violação da suposição de independência** ao não considerar o espaço nas análises.

Este entendimento da heterogeneidade e interdependência dos eventos é crucial para a eficácia das ações de saúde pública. No próximo tópico, serão apresentadas algumas das principais aplicações em saúde, destacando como essas ferramentas contribuem para o monitoramento e controle de doenças, além de outras áreas relevantes.

## Tecnologias de Geoprocessamento



A utilização de técnicas de geoprocessamento é fundamental na epidemiologia, uma vez que permite uma visão abrangente da saúde dos indivíduos no contexto social, histórico, político, cultural e ambiental em que estão inseridos. Atualmente, existem vários *softwares* que apoiam as análises espaciais além do R, como o TabWin, o ArcGIS, o QGIS, o Geoda, o Google Maps e o Google Earth. Na prática da análise espacial, a escolha da ferramenta ideal vai depender do seu objetivo e da sua familiaridade com cada **software**.

Cada ferramenta oferece recursos únicos para apoiar as análises em saúde. O R, por exemplo, se destaca por sua flexibilidade e capacidade de integração com diversas bibliotecas específicas para análise espacial e, dessa forma, pode ser um auxílio para diversos softwares. Se você já tem afinidade com o R, pode aproveitar suas funcionalidades para complementar e agregar valor às suas análises.

O importante é usar a ferramenta que melhor se adapta ao seu perfil e às demandas da análise, garantindo eficiência e precisão no trabalho.

## *Principais conceitos e aplicações em Saúde*

A análise espacial é uma ferramenta poderosa na Saúde Pública, especialmente na Vigilância em Saúde. Ela permite entender como eventos de saúde se distribuem geograficamente, possibilitando a identificação de áreas de risco e o planejamento de ações de prevenção e controle. Confira algumas aplicações:

- **Mapeamento de doenças:** Avaliar a variação geográfica na ocorrência de doenças e identificar áreas com maior incidência, contaminações ambientais ou outros eventos de saúde. O mapeamento ajuda a direcionar recursos e implementar medidas preventivas e de controle de forma mais eficaz.
- **Detecção de clusters:** Identificar agrupamentos de eventos em determinadas áreas e determinar a significância de um risco adicional nessas regiões. Esses clusters podem revelar a presença de fatores de risco locais, como agentes infeciosos, contaminação ambiental localizada ou efeitos colaterais de tratamentos.
- **Estudos ecológicos:** Consistem basicamente em modelos de regressão que buscam explicar a variação na incidência de uma doença com base em outras variáveis, integrando aspectos ambientais, sociodemográficos e comportamentais.
- **Monitoramento ambiental:** Estimar e acompanhar a distribuição espacial de fatores ambientais relevantes para a saúde, como poluentes químicos, insolação, vegetação e clima. Esses fatores podem influenciar a ocorrência de doenças, sendo fundamentais na avaliação dos riscos à saúde.
- **Planejamento de ações de saúde:** Identificar padrões de distribuição espacial de equipamentos de saúde e orientar ações de prevenção e controle. Por exemplo, a localização de unidades de saúde, a distribuição de vacinas, e a implementação de campanhas de prevenção podem ser otimizadas com base nesses dados.

Para cada uma dessas aplicações, é fundamental utilizar dados que considerem a localização geográfica dos eventos, pois isso é essencial para a correta interpretação e análise dos resultados. A seguir, apresentaremos os principais tipos de dados espaciais.

## *Tipologia dos dados espaciais*

Ao realizar uma análise espacial, o primeiro passo é compreender o tipo de dado com o qual estamos trabalhando. Os dados espaciais estão sempre associados a uma localização geográfica específica, identificada, por exemplo, por coordenadas de latitude e longitude. Dependendo do evento que se deseja analisar, esses dados podem ser de tipos diferentes.

Existem três tipos principais de dados espaciais, cada um adequado para situações específicas:

- **Dados de processos pontuais:** Utilizados para analisar eventos que ocorrem em pontos específicos do espaço, como a localização exata de casos de uma doença em uma cidade ou pontos de depósito irregular de contaminantes. Essa localização é dada por coordenadas geográficas (latitude e longitude) ou coordenadas planas (x e y). É a forma mais simples de dado espacial e é frequentemente usada em estudos de saúde. São ideais para identificar padrões de distribuição, detectar agrupamentos e avaliar a dependência espacial.
- **Dados de área:** Usados quando o espaço é dividido em regiões ou áreas (bairros, municípios, estados) e os dados são agregados a esses níveis. Exemplos incluem a incidência de doenças por região, densidade populacional em municípios e cobertura vegetal em biomas. Esses dados ajudam a mapear a distribuição espacial de fenômenos, identificar áreas de risco e analisar a associação entre variáveis.
- **Dados de geoestatística:** Para situações onde os fenômenos são contínuos em um espaço, como a temperatura ou a poluição do ar. Nesse caso, as medições são realizadas em pontos amostrais e depois estimadas para áreas onde não houve coleta, sendo amplamente usados em modelos estatísticos de distribuição espacial.



O analista da vigilância em saúde se depara o tempo todo com a disponibilidade de dados para esses eventos. O acesso aos dados produzidos no nível local se dá às equipes de analistas deste nível. Mas, frequentemente, há a necessidade de integrar dados de outros setores (como os dados sobre internações). Fortalecer parcerias com outros setores e instituições é fundamental para a obtenção de dados de qualidade e para a realização de análises mais robustas.

Para cada tipo de dado, há métodos estatísticos diferentes para descrever ou analisar a distribuição espacial dos eventos. A seguir, o Quadro 1 apresenta exemplos de cada um desses tipos de dados e as técnicas de análise mais comuns para cada um deles.

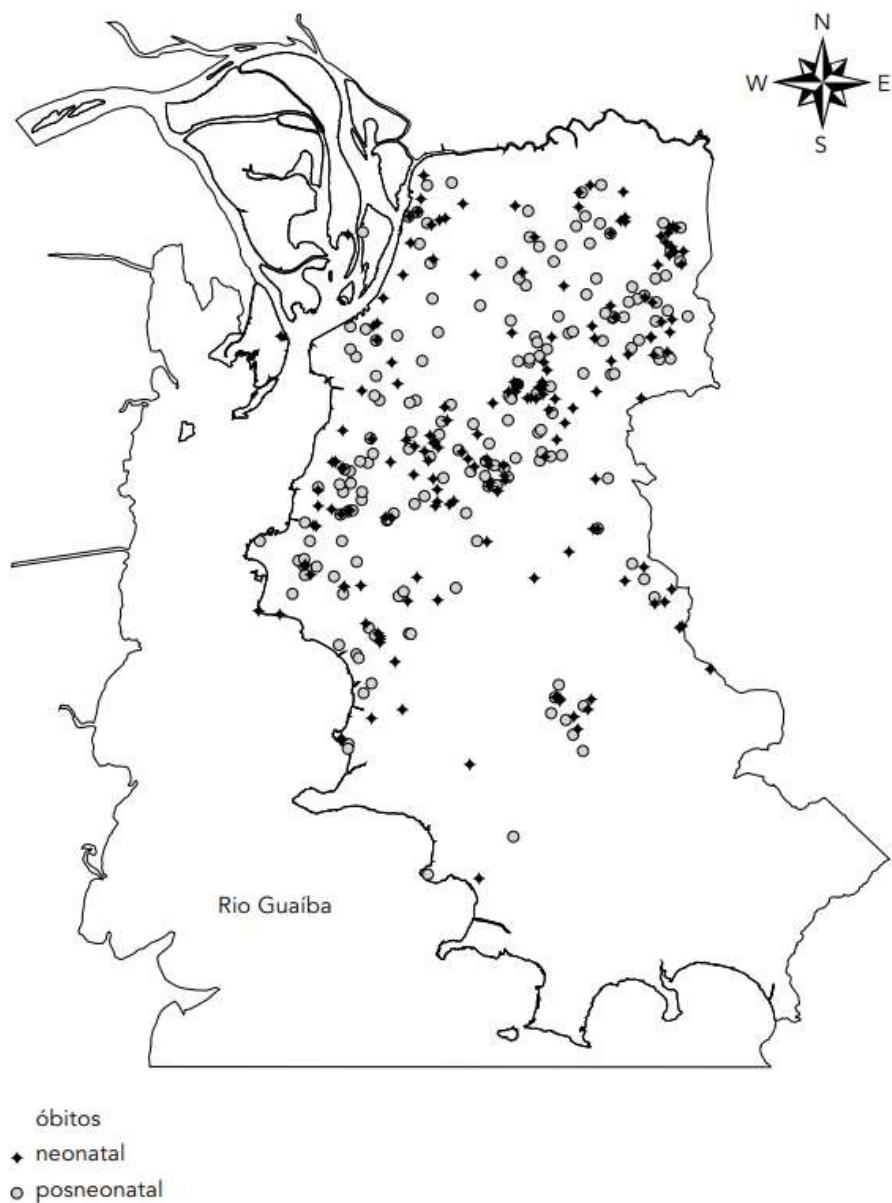
#### **Quadro 1. Exemplos de tipos de dados espaciais e técnicas de análise.**

<b>Tipo de Dado</b>	<b>Exemplos</b>	<b>Técnicas</b>
Pontos	Eventos localizados, como ocorrências de doenças. Os pontos podem representar a localização da residência de casos de uma doença.	A análise busca identificar padrões de cluster e dependência espacial.
Área	Dados agregados a regiões, como dados censitários (população por setor censitário, percentual de casas com esgotamento adequado, etc).	São usados métodos que exploram a correlação espacial e modelos de regressão para analisar a distribuição de fenômenos em diferentes polígonos.
Geoestatística (amostras)	Dados de estações meteorológicas, como chuva e temperatura.	Técnicas como a interpolação de superfícies são empregadas para estimar valores em locais não amostrados.

Agora vamos ilustrar os três principais tipos de dados espaciais que podem ser empregados na rotina da vigilância.

A Figura 2 apresenta a localização espacial de óbitos infantis em Porto Alegre, Rio Grande do Sul, em 1998. Perceba que são utilizados dados pontuais, mapeando individualmente os eventos e permitindo a visualização da distribuição geográfica dos óbitos infantis.

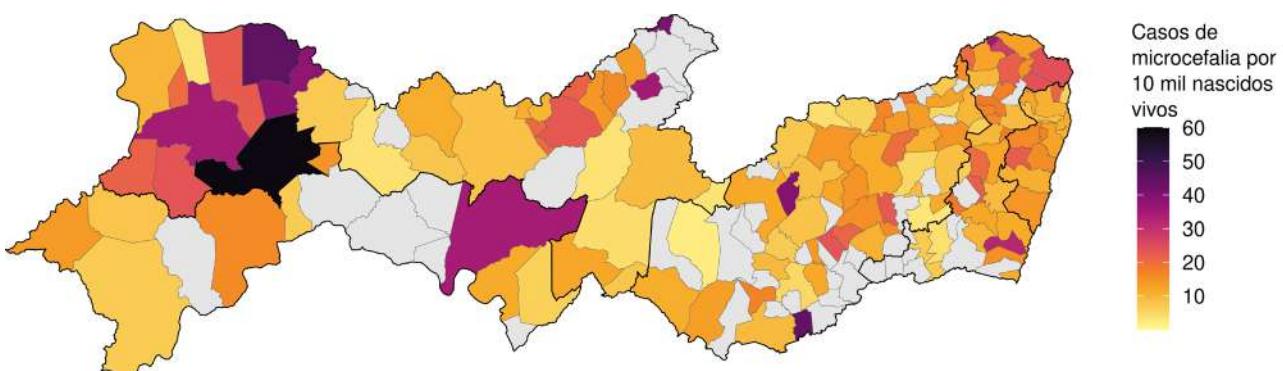
**Figura 2: Exemplo de dados de pontos - Distribuição espacial de nascidos vivos e óbitos infantis em Porto Alegre, 1998.**



Fonte: CARVALHO, M. S.; SOUZA-SANTOS, R. Cadernos de Saúde Pública, 21(2):361-378, 2005.

A Figura 3 apresenta um exemplo de dados de área com o número de casos de microcefalia por 10 mil nascidos vivos por município no estado de Pernambuco de 2015 a 2017.

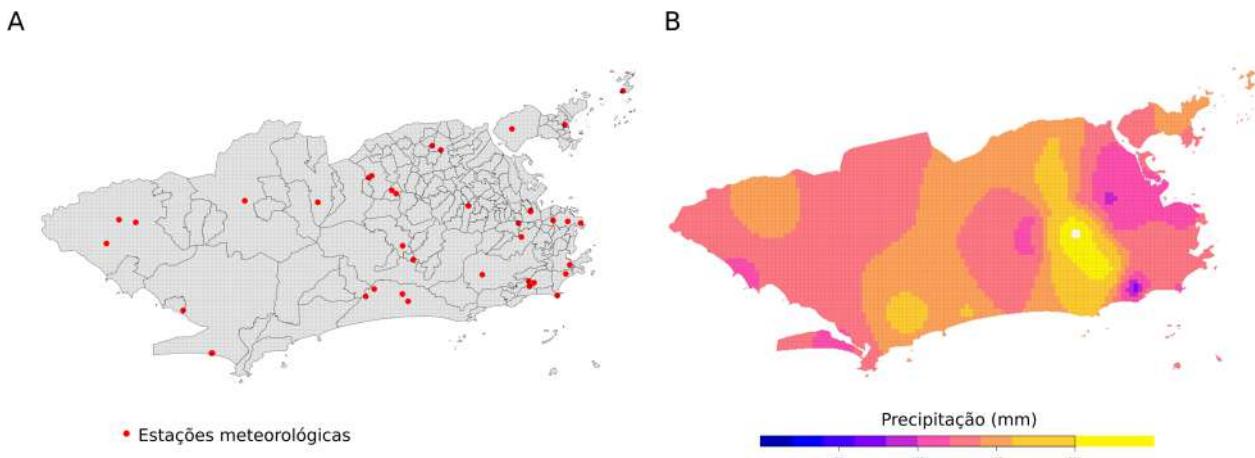
**Figura 3: Exemplo de dados de área - Casos de microcefalia por 10 mil nascidos vivos por município, 2015-2017, Pernambuco.**



Fonte: FREITAS, L. P. et al. Trans R Soc Trop Med Hyg, 117(3):189-196, 2023.

Por fim, a Figura 4 ilustra o uso de dados de amostragem para geoestatística. Neste exemplo, os dados de chuva obtidos em estações meteorológicas no município do Rio de Janeiro (painel A) foram utilizados para estimar a precipitação de forma contínua em todo o território (painel B). No módulo 2 do curso vamos aprender mais sobre o método que aqui foi utilizado, a krigagem.

**Figura 4: Exemplo de dados de geoestatística - Localização de estações meteorológicas onde os dados amostrais são coletados (A) e estimativa de precipitação de forma contínua no território (B), município do Rio de Janeiro.**



Esses exemplos demonstram a riqueza dos dados espaciais. A análise desses tipos de dados é essencial para a compreensão da distribuição das doenças e na formulação de políticas públicas mais eficazes para a prevenção e o controle de agravos à saúde.

Um artigo na área da saúde pública que explora e distingue claramente os três tipos principais de dados espaciais (dados de processos pontuais, dados de área e dados de geoestatística) é o artigo:

“CARVALHO, M. S; SOUZA-SANTOS, R. Análise de dados espaciais em saúde pública: métodos, problemas, perspectivas. Cadernos de Saúde Pública, 21(2): 361-378, 2005”. Disponível no [link](#).

Este material pode ser uma excelente leitura complementar para quem deseja aprofundar-se na análise espacial aplicada à saúde pública.



### OBSERVAÇÃO!

Existem ainda outros tipos de dados espaciais, como, por exemplo, imagens de satélites (raster), redes e fluxos. Estes não serão abordados nesse curso.

Eventualmente misturas de diferentes tipos de dados estão presentes em uma mesma análise. Em algumas situações pode-se converter o dado de um tipo para outro (troca de suporte).

## *Dependência ou autocorrelação espacial*

A dependência espacial (ou autocorrelação espacial) é um conceito central na análise espacial, indicando que eventos ou observações em locais próximos tendem a ser mais semelhantes do que aqueles que estão distantes. Em outras palavras, a presença ou intensidade de um fenômeno em uma área pode influenciar a ocorrência do mesmo fenômeno em locais vizinhos.

Um marco para essa ideia foi a “Primeira Lei de Tobler”, formulada por Waldo Tobler (1970), que afirma: “Tudo está relacionado a tudo, mas coisas próximas estão mais relacionadas do que coisas distantes”. Essa lei destaca que os fenômenos geográficos raramente ocorrem de forma isolada; eles se interligam, e essa interconexão precisa ser considerada em análises e modelos estatísticos.

Como ressaltado por Cressie (1991), embora a hipótese de independência seja conveniente para a teoria estatística, os dados espaciais normalmente apresentam dependência em todas as direções, com a intensidade dessa relação diminuindo à medida que aumenta a distância entre as observações. Essa característica torna os modelos que incorporam dependência estatística mais realistas, embora também mais complexos.

A autocorrelação espacial pode ser avaliada de duas maneiras principais:

- **Global:** Analisa a distribuição geral dos valores de uma variável em toda a área estudada para verificar se há um padrão de associação espacial ou se os dados estão distribuídos aleatoriamente. Índices como o de Moran global são frequentemente usados nessa abordagem.
- **Local:** Foca em identificar padrões específicos em pequenas áreas, permitindo detectar agrupamentos ou clusters onde os fenômenos se concentram. Ferramentas como o índice de Moran local auxiliam na identificação dessas áreas críticas.

Uma das ferramentas para visualizar a autocorrelação espacial são os correlogramas, gráficos que mostram como a correlação varia em função da distância (ou lag) entre os pontos. Esses gráficos são bastante versáteis e podem ser aplicados a diversos tipos de dados, como contagens, presença/ausência, proporções, distâncias, direções e até séries temporais.

Compreender a dependência espacial é fundamental para interpretar corretamente os dados em saúde, pois ela revela não apenas onde os riscos estão concentrados, mas também como esses riscos se espalham e interagem no espaço geográfico. Agora, vamos aprofundar essa discussão explorando a autocorrelação espacial, que nos permite quantificar o grau de associação entre os valores de uma variável em diferentes localidades e identificar padrões que podem não ser evidentes apenas pela observação visual.

## *Estacionariedade espacial*

A estacionariedade é um conceito fundamental na análise espacial pois permite inferir valores em locais não amostrados com base na variabilidade observada nas amostras disponíveis do local a ser analisado. Ou seja, **um processo espacial é considerado estacionário quando suas propriedades estatísticas permanecem constantes ao longo do espaço**. Isso significa que a média dos valores é uniforme em toda a área estudada e que a covariância entre quaisquer dois pontos depende apenas da distância que os separa, e não de sua localização específica.

A estacionariedade é uma hipótese importante para muitas técnicas de interpolação e modelagem espacial, como a krigagem, pois garante que as propriedades estatísticas do processo sejam uniformes em toda a área estudada.

## *Isotropia e anisotropia*

A isotropia é uma condição mais restritiva que a estacionariedade. Em um processo isotrópico, além da média constante, a covariância depende exclusivamente da distância entre os pontos, independentemente da direção. Isso implica que os padrões espaciais se repetem de maneira uniforme em todas as direções.

Em contraste, um processo é considerado anisotrópico quando a covariância varia não apenas com a distância, mas também com a direção. Fatores ambientais, topográficos ou influências de processos naturais (como ventos predominantes ou correntes de água) podem gerar essa variação direcional, resultando em padrões que diferem conforme o ângulo considerado.

Esses conceitos, embora mais técnicos, são fundamentais para a interpretação de padrões espaciais e a escolha de métodos de análise adequados. A estacionariedade, a isotropia e a anisotropia são pressupostos importantes em muitas técnicas de interpolação e modelagem espacial, e sua consideração é essencial para garantir a validade dos resultados obtidos.

Mas não se preocupe, estamos apresentando os conceitos que vamos aplicar de forma mais aprofundada em cada módulo do nosso curso.

## *Considerações finais*

Em resumo, as análises espaciais aplicadas aos dados de vigilância em saúde são poderosas para identificar problemas em saúde pública e direcionar medidas de intervenção. Vimos que a estatística espacial teve seu início com John Snow durante uma epidemia de cólera em Londres. Também aprendemos sobre os tipos de dados espaciais e principais conceitos em análise espacial.

Nos próximos módulos, entraremos a fundo a cada um dos tipos de dados e as análises que podem ser realizadas para cada objetivo. Vamos lá?

## Referências

- BAILEY, T. C. Interactive spatial data analysis. Harlow Essex, 1995.
- BIVAND, R. S.; PEBESMA, E. J.; GÓMEZ-RUBIO, V. Applied spatial data analysis with R. Springer Science & Business Media, 2013.
- CARVALHO, M. S.; SOUZA-SANTOS, R. Análise de dados espaciais em saúde pública: métodos, problemas, perspectivas. Cadernos de Saúde Pública, Rio de Janeiro, v. 21, n. 2, p. 361-378, mar./abr.
- CRESSIE, N. Statistics for spatial data. Wiley, 1991.
- FORTES, B. P. M. D.; VALENCIA, L. I. O.; RIBEIRO, S. V.; MEDRONHO, R. A. Modelagem geoestatística da infecção por Ascaris lumbricoides. Cadernos de Saúde Pública, Rio de Janeiro, v. 20, n. 3, p. 727-734, maio/jun.
- GETIS, A.; ORD, J. K. The analysis of spatial association by use of distance statistics. Geographical analysis, v. 24, n. 3, p. 189-206, 1992.
- MENDES, M. S.; OLIVEIRA, A. L. S.; PIMENTEL, L. M. L. M.; FIGUEIREDO, T. M. R. M.; SCHINDLER, H. C. Análise espacial da tuberculose em menores de 15 anos de idade e risco socioeconômico: um estudo ecológico na Paraíba, 2007-2016. Epidemiologia e Serviços de Saúde, Brasília, v. 30, n. 3, e20201038, 2021.
- SNOW, J. (1854). On the mode of communication of cholera. John Churchill.
- TOBLER, Waldo R. A computer movie simulating urban growth in the Detroit region. Economic geography, v. 46, n. sup1, p. 234-240, 1970.

## Módulo 2: Padrões Pontuais

A compreensão do lugar na vigilância em saúde é essencial para entender a distribuição e a dinâmica de doenças e outros fenômenos de saúde. Na análise de padrões pontuais — isto é, conjuntos de casos representados por pontos no mapa — eles identificam rapidamente localidades de risco elevado, passíveis de intervenção. A simples sobreposição desses pontos revela se os eventos se agregam, se dispersam ou se distribuem ao acaso; quando surgem aglomerações inesperadas, eles acionam investigações de campo e direcionam equipes e insumos para as áreas críticas. Esse olhar do espaço permite que a Vigilância em Saúde seja mais proativa, prevendo surtos e epidemias, otimizando recursos e melhorando a resposta a emergências de saúde pública.

No módulo 1 deste curso você aprendeu sobre os dados espaciais e como eles podem ser representados em um mapa. Agora, vamos nos aprofundar na análise de **padrões pontuais**. Vamos explorar como esses dados podem ser utilizados para entender a distribuição de eventos no espaço e como isso pode ser aplicado na vigilância em saúde.

Vamos lá?

## O que são padrões pontuais?

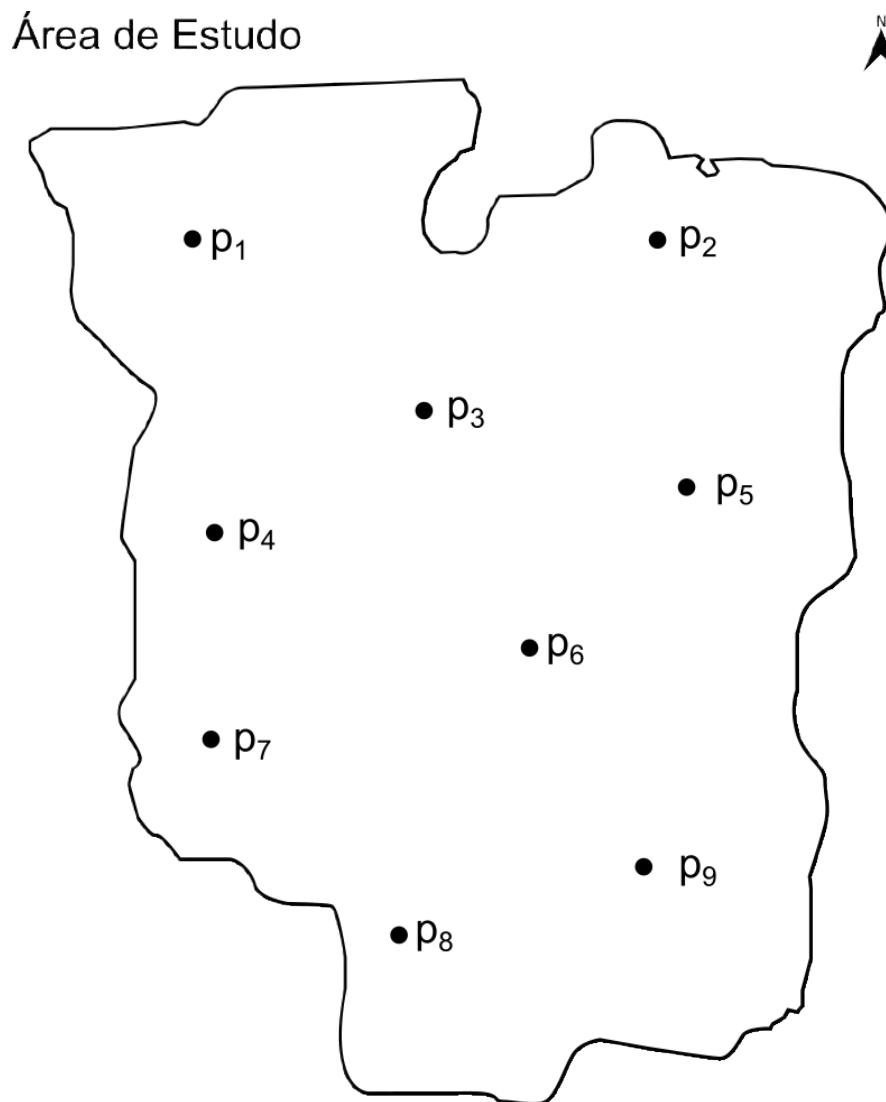
Como vimos no módulo anterior, o ponto é o tipo mais simples de dado espacial e possui um par de coordenadas ( $X, Y$ ) em um mapa ou plano. Vejamos abaixo um exemplo simples de tabela contendo dados de pontos:

Evento	Coordenada X	Coordenada Y
1	4.30	2.45
2	5.39	3.35
3	4.10	3.50

Ao analisar a localização de pontos no espaço, é possível identificar padrões que revelam informações valiosas sobre os fenômenos em estudo. Na tabela anterior, cada linha representa um evento e sua localização dada pelas coordenadas X e Y. Na rotina da vigilância em saúde, essas coordenadas podem indicar casos de doenças como, por exemplo, chikungunya. Um agrupamento de casos de chikungunya pode sugerir uma abundância de criadouros de mosquitos nas proximidades, o que, por consequência, aciona ações de campo e direciona equipes para áreas críticas. A análise de padrões de pontos é, portanto, uma ferramenta essencial para monitorar riscos, avaliar a eficácia das intervenções e compreender a dinâmica espacial dos eventos de saúde.

Chamamos a análise de dados de pontos de análise de **Padrão de Pontos** (ou **Processos Pontuais**). Esta técnica é fundamental para estudar a distribuição espacial de eventos em uma determinada área de estudo. A análise de processos pontuais se inicia com a visualização dos pontos na área de estudo, conforme esquematizado na Figura 5. Nesse exemplo, temos uma área de estudo fictícia com várias localizações pontuais ( $p_1, p_2, \dots, p_n$ ).

**Figura 5: Área de estudo fictícia com o conjunto de dados consistindo de uma série de localizações pontuais.**



Neste momento, não precisamos nos preocupar com o que cada ponto representa, mas sim com a sua localização e a forma com que estão distribuídos na área de estudo. A Figura acima permite, portanto, visualizar a distribuição espacial dos pontos e avaliar se há padrões. Dessa forma, a partir da visualização dos pontos, podemos começar a investigar o padrão de distribuição dos eventos. No próximo item, exploraremos detalhadamente a importância dessa análise na vigilância em saúde.

## *Por que é importante?*

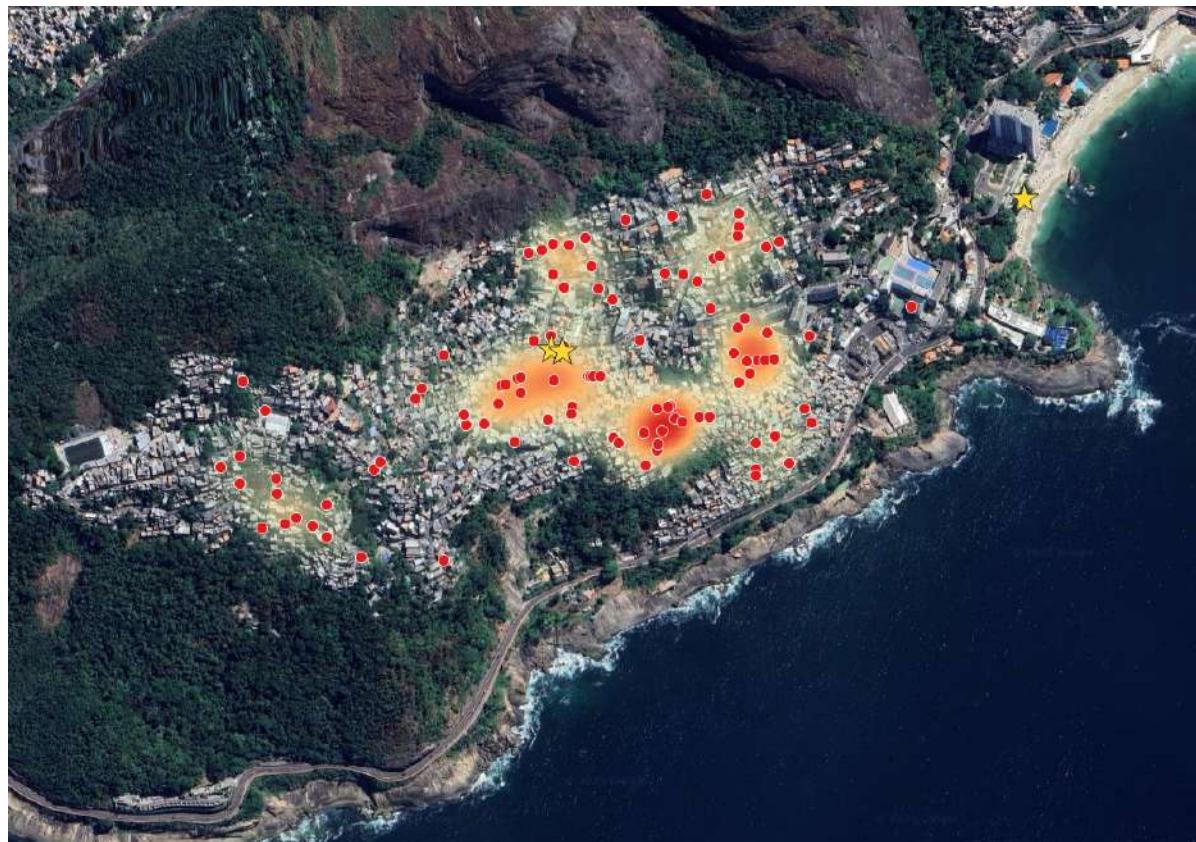
A análise de padrões pontuais aplica-se a diversos agravos de interesse da vigilância em saúde. Quando examinam a distribuição de casos de tuberculose como pontos em um mapa, por exemplo, conseguem localizar focos prováveis de transmissão e priorizar ações de busca ativa. Do mesmo modo, ao mapearem acidentes de trânsito, identificam trechos viários que demandam maior oferta de ambulâncias e ajustes na sinalização.

Depois de revelarem esses agrupamentos, as equipes conectam a distribuição espacial dos eventos a fatores ambientais, sociais ou econômicos e, a partir daí, conseguem:

- Relacionar a distribuição dos eventos com fatores ambientais, sociais ou econômicos.
- Fazer previsões da ocorrência de eventos futuros em áreas específicas.
- Otimizar recursos de forma mais eficiente, como serviços públicos ou campanhas de saúde.
- Tomar decisões através de políticas públicas e estratégias de planejamento.

Um exemplo concreto ilustra o potencial dessa abordagem. No verão de 2017-2018, registrou-se um surto de hepatite A no município do Rio de Janeiro. Na Figura 6, cada ponto marca um caso suspeito investigado e as estrelas amarelas assinalam os locais onde o vírus da hepatite A foi detectado em amostras de água pelos analistas da Vigilância Epidemiológica. Ao sobrepor esses dados, os analistas visualizaram rapidamente a coincidência entre casos humanos e pontos de contaminação hídrica, reforçando a necessidade de intervenções sanitárias e de comunicação de risco junto às populações mais expostas.

**Figura 6: Distribuição espacial de casos suspeitos em uma investigação de surto de hepatite A no Rio de Janeiro no verão de 2017/2018.**



A partir dessa visualização, pode-se levantar três questões centrais para orientar as próximas análises:

- 1. Existe algum padrão na distribuição dos casos?**
- 2. O risco é maior próximo às fontes de contaminação?**
- 3. Qual é o provável local de exposição ao vírus?**

Neste momento, vamos concentrar primeiro na primeira pergunta. Em uma inspeção visual rápida, podemos identificar que o padrão na distribuição dos casos na Figura 6 difere daquele observado na Figura 5. Na Figura 5, os pontos parecem mais espalhados em toda a área de estudo, enquanto que, na Figura 6, existem alguns agrupamentos de pontos. A partir dessa diferença, podemos discutir agora quais tipos de padrões de distribuição pontual podem ocorrer, de modo a classificar corretamente o cenário observado e selecionar as ferramentas estatísticas adequadas nas próximas etapas.

Vamos, então, conhecer quais são os tipos de padrões de distribuição pontual?

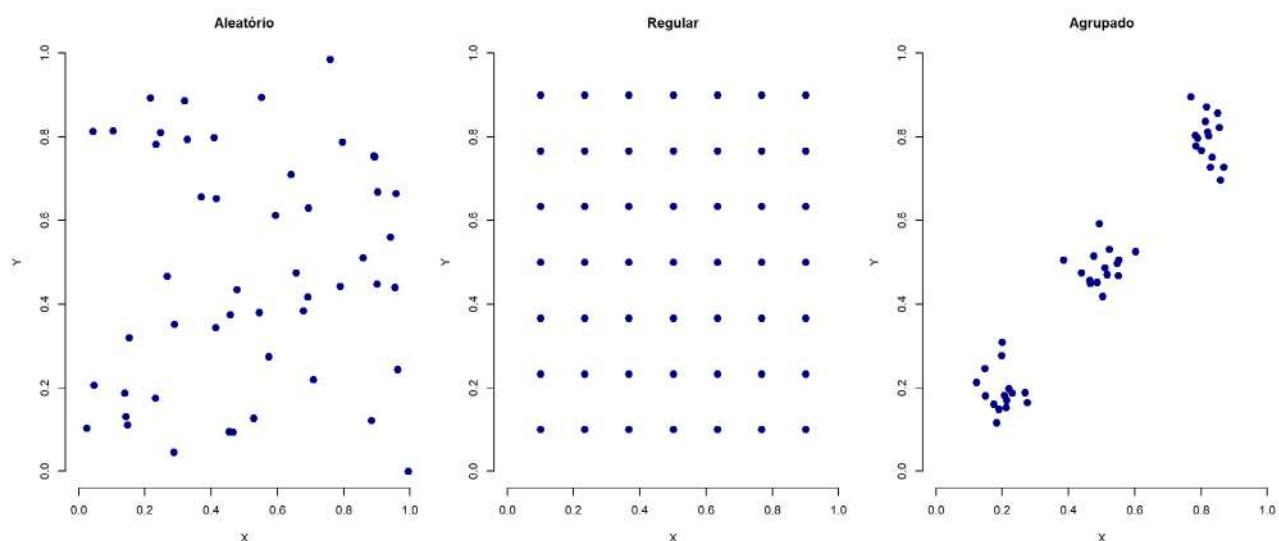
## Padrões de distribuição pontual

A disposição espacial dos pontos pode ser classificada em três categorias principais:

- **Aleatório:** Cada ponto tem a mesma probabilidade de ocorrer em qualquer local, e sua posição não é influenciada pela localização de outros pontos.
- **Regular (ou uniforme):** Os pontos estão distribuídos de forma que mantêm, em geral, uma distância aproximadamente uniforme entre si, criando uma disposição quase equidistante.
- **Agrupado (ou cluster):** Os pontos estão concentrados em determinadas áreas, formando grupos densos, enquanto outras regiões podem conter poucos ou nenhum ponto.

Veja abaixo na Figura 7 como os diferentes padrões de pontos são representados graficamente nos eixos  $X$  e  $Y$ :

**Figura 7: Padrões espaciais de pontos, a partir de suas coordenadas.**



Compreendidos os padrões da disposição dos pontos, o próximo passo é investigar como esses padrões se formam. Para isso, vamos conhecer os chamados **processos pontuais**.

## *Processos pontuais*

Agora, vamos compreender melhor como analisar os dados pontuais. Para interpretar esses padrões, podemos definir dois tipos básicos de processos pontuais:

- i. **Processos de primeira ordem:** Esses processos referem-se a fenômenos globais ou de grande escala, relacionados às variações no valor médio do processo no espaço. Esse processo é geralmente representado por variações na intensidade ou densidade de pontos no espaço. Para exemplificar um processo de primeira ordem, podemos citar o aumento de casos de dengue em áreas com maior densidade populacional e áreas mais urbanizadas. Para analisar esse fenômeno, utiliza-se frequentemente a estimativa de Kernel para avaliar a média dos eventos e, dessa forma, visualizar regiões com maior ou menor concentração de casos.
- ii. **Processos de segunda ordem:** Esses processos referem-se a fenômenos locais ou de pequena escala, investigam interações entre eventos próximos, avaliando a existência de dependência espacial em uma escala menor, local. O interesse aqui é determinar se os pontos interagem entre si, influenciando a localização uns dos outros. Por exemplo, casos de tuberculose podem ocorrer mais próximos uns dos outros devido à transmissão pessoa a pessoa, gerando pequenas concentrações locais ou clusters. Para investigar esses padrões locais, métodos como a análise dos vizinhos mais próximos ou a Função K são utilizados para verificar se há uma distância típica entre casos próximos, indicando possível interação ou agrupamento.

Além de entender como os pontos se formam, também é preciso avaliar estatisticamente se esses padrões ocorreram ao acaso ou se eles se distribuem de forma regular ou agrupada por influência de algum fator externo. É aqui que entra a análise da **completa aleatoriedade espacial (CSR - Complete Spatial Randomness)**. Em outras palavras, cada evento tem a mesma probabilidade de ocorrer em qualquer lugar dentro da área analisada, sem formar padrões específicos (nem agrupados, nem regulares).

Testar a CSR é justamente avaliar se a distribuição observada é fruto do acaso ou se existe alguma estrutura espacial determinando o padrão encontrado (por exemplo, fontes de contaminação, transmissão interpessoal, características ambientais, entre outras).

Na prática, isso é feito testando-se duas hipóteses estatísticas:

- **Hipótese nula ( $H_0$ ):** Os pontos (casos de doenças, acidentes ou outros agravos) estão distribuídos aleatoriamente no espaço, não havendo nenhum fator influenciando sua localização.
- **Hipótese alternativa ( $H_1$ ):** Os pontos formam agrupamentos (clusters) ou apresentam-se dispersos de forma regular, sugerindo fatores externos que influenciam sua localização.

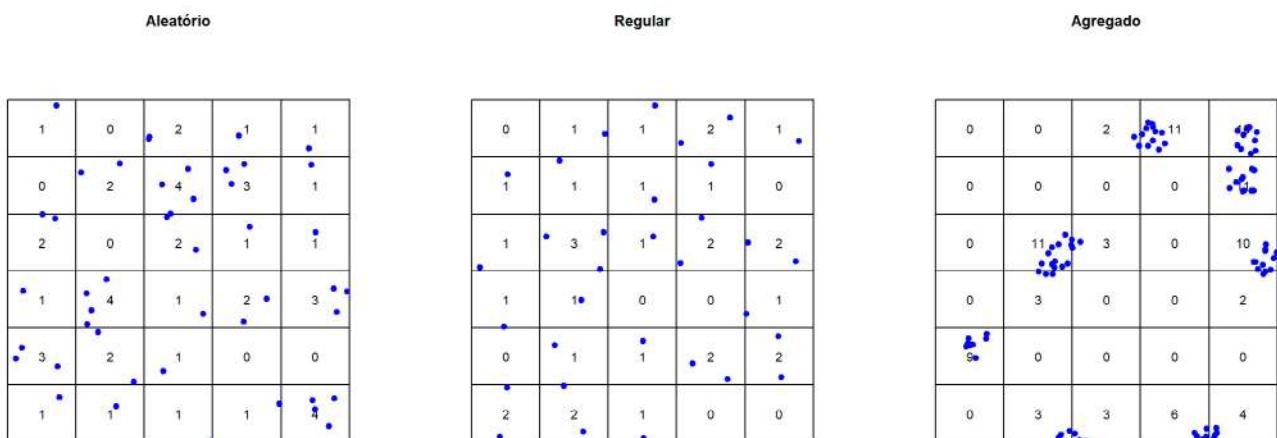
A suposição de CSR implica um processo homogêneo de Poisson, significando que todos os locais têm a mesma probabilidade de ocorrência dos eventos.

Por exemplo, se quisermos avaliar se casos de hepatite A estão relacionados a locais específicos de contaminação, primeiro precisamos testar se os casos estão distribuídos aleatoriamente ou não. Se rejeitarmos a hipótese de completa aleatoriedade, isso sugere fortemente que existe algum fator influenciando o padrão observado, reforçando a necessidade de investigação detalhada sobre locais ou fatores ambientais envolvidos.

## *Simulação de padrões pontuais e contagem em quadrantes*

Agora, vamos aprender a simular e testar esses padrões pontuais usando a técnica de distribuição por quadrantes, para comprovar se a distribuição dos eventos é de fato aleatória ou se há evidências de agrupamento ou regularidade.

**Figura 8: Disposição espacial dos pontos, levando em consideração três padrões espaciais de distribuição de pontos, destacados em grades com contagens de ocorrências em cada célula.**



A Figura 8 apresenta os três padrões espaciais de distribuição de pontos, vistos anteriormente. Além disso, os pontos estão destacados em grades com contagens de ocorrências em cada célula.

- **Aleatório:** A distribuição dos pontos parece desordenada, sem uma organização clara ou padrão definido. As contagens por célula variam de forma aleatória, sugerindo uma distribuição espacial sem dependência ou interação entre os pontos.
- **Regular:** Os pontos estão distribuídos de maneira uniforme em toda a grade, com contagens similares e bem distribuídas entre as células. Isso indica uma estrutura um pouco mais organizada e também é possível observar um espaçamento quase equidistante entre os pontos.
- **Agregado:** Os pontos estão concentrados em áreas específicas, formando clusters com células de contagens mais altas. Algumas regiões da grade possuem alta densidade, enquanto outras apresentam poucas ou nenhuma ocorrência, evidenciando um padrão de agrupamento.

A seguir, vamos aplicar o teste de Completa Aleatoriedade Espacial (CSR) nesses padrões de pontos. Neste momento, apresentaremos apenas os comandos e as interpretações básicas dos resultados. Não se preocupe, pois na seção prática com R detalharemos melhor o funcionamento deste teste.

```
# Padrão Aleatório
quadrat.test(aleatorio_qc)
```

```
Chi-squared test of CSR using quadrat counts
```

data:

X2 = 27.043, df = 29, p-value = 0.8613 alternative hypothesis: two.sided

Quadrats: 5 by 6 grid of tiles

O comando `quadrat.test()` realiza um teste da Completa Aleatoriedade Espacial (CSR) para um padrão de pontos baseado nas contagens em quadrantes. Por padrão, utiliza o teste qui-quadrado, mas pode realizar testes baseados em Monte Carlo. É usado para verificar se a distribuição dos pontos é uniforme ou segue o modelo esperado.

Para esse padrão dos dados pontuais, o  $p - valor = 0,8613$ , ou seja, o  $p - valor < \alpha$  supondo um  $\alpha = 0,05$ , indicando que não há evidências suficientes para rejeitar a hipótese nula de que o padrão de pontos segue uma distribuição aleatória completa.

```
# Padrão Regular
quadrat.test(regular_qc)
```

Chi-squared test of CSR using quadrat counts

**data:**

X2 = 16.75, df = 29, p-value = 0.06815 alternative hypothesis: two.sided

**Quadrats:** 5 by 6 grid of tiles

Para esse cenário, o  $p - valor = 0,8615$ , ou seja, também o  $p - valor < \alpha$  supondo um  $\alpha = 0,05$ , indicando que não há evidências suficientes para rejeitar a hipótese nula de que o padrão de pontos segue uma distribuição aleatória completa.

```
# Padrão agregado ou cluster
quadrat.test(agregado_qc)
```

Chi-squared test of CSR using quadrat counts

**data:**

X2 = 164.27, df = 29, p-value < 2.2e-16 alternative hypothesis: two.sided

**Quadrats:** 5 by 6 grid of tiles

Já para esse cenário, o  $p - valor = 2,2 \times 10^{-16}$ , ou seja, também o  $p - valor < \alpha$  supondo um  $\alpha = 0,05$ , indicando que há evidências fortes para rejeitar a hipótese nula de que o padrão de pontos segue uma distribuição aleatória completa.

Na prática da vigilância, um passo possível depois de plotar os casos em um mapa é transformá-los em um produto que revele a intensidade do evento. Uma técnica muito usada para isso é a Estimativa de Densidade por Kernel, também conhecida como **mapa de calor**, que veremos a seguir.

## *Estimativa de densidade de Kernel (mapa de calor)*

A **estimativa de densidade de Kernel** é uma ferramenta exploratória utilizada para analisar processos pontuais, permitindo calcular a intensidade das ocorrências em uma determinada região. Esse método gera uma superfície contínua, onde cada valor representa a intensidade de eventos por unidade de área. Essa técnica também é amplamente conhecida por **mapa de calor**, por representar áreas de concentração de eventos nas quais são utilizadas simbologias de cores quentes para áreas de alta concentração (por exemplo, vermelho) e cores frias para áreas de baixa concentração (azul ou verde).

O método é relativamente simples: essa técnica utiliza uma janela móvel que percorre toda a área de estudo e aplica pesos variáveis a cada ponto, considerando a distância dos pontos dentro da janela ao ponto avaliado: mais perto, peso maior; mais longe, peso menor. Como resultado, cria-se um mapa de calor que revela, de forma suavizada, áreas de alta e baixa intensidade e destacando padrões de concentração ou dispersão. Esse resultado é o mapa de calor que a vigilância utiliza para priorizar ações.

Dessa forma, mesmo locais onde nenhum caso foi registrado recebem uma estimativa, permitindo antecipar riscos, inferir rotas prováveis de propagação e até planejar o uso de recursos.

### **Por que isso importa para a vigilância?**

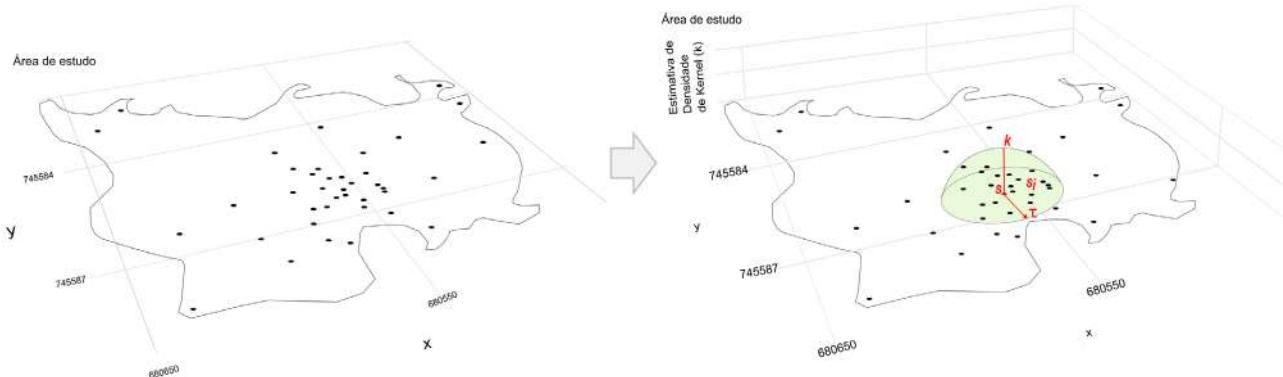
No Quadro 2 são apresentados alguns exemplos de como a estimativa de densidade de Kernel pode ajudar na análise espacial exploratória na vigilância em saúde. Esses exemplos ilustram como a Estimativa de Densidade de Kernel pode ser aplicada para identificar padrões e direcionar ações de saúde pública.

### **Quadro 2. Exemplos de usos da estimativa de densidade de kernel.**

Exemplo prático	Utilidade da estimativa de densidade de Kernel
Focos de <i>Aedes aegypti</i> ao longo do verão	Destaca bairros de maior intensidade para direcionar equipes de controle vetorial.
Acidentes de trânsito com vítimas graves	Evidencia "hot spots" viários para reforçar sinalização ou instalar lombadas eletrônicas.
Mordeduras de escorpião em zona rural	Permite localizar criadouros prováveis e orientar campanhas educativas.

Agora, vamos ilustrar o conceito de estimativa de densidade de Kernel aplicada em um processo pontual em uma área de estudo fictícia (Figura 9). Considere que nessa área há uma distribuição pontual de casos de um fenômeno de saúde (imagem da esquerda). Na imagem da direita, temos a representação tridimensional da função Kernel aplicada a um ponto específico  $s$  (ponto vermelho). Perceba que em ambas imagens a área de estudo está em perspectiva. O parâmetro  $\tau$ , geralmente chamado de *largura de banda ou bandwidth*, regula o raio de influência da função Kernel sobre os pontos vizinhos  $s_i$ . O vetor vermelho  $k$  indica a densidade estimada em  $s_i$ , enquanto a superfície curva representa o kernel em si, mostrando a ponderação dos eventos de acordo com a distância ao ponto  $s$ .

**Figura 9: Conceito de estimativa de densidade de Kernel aplicada em um processo pontual.**



Neste sentido, a densidade da intensidade estimada ( $\hat{\lambda}(s)$ ) em uma localização  $s$  é calculada pela fórmula:

$$\hat{\lambda}(s) = \sum_{i=1}^n \frac{1}{\tau^2} k\left(\frac{s - s_i}{\tau}\right)$$

para a qual:

- $\hat{\lambda}(s)$  é a estimativa da intensidade no ponto (local de interesse).
- $k(\cdot)$  é a função Kernel, escolhida para suavizar os dados. Pode ser, por exemplo, uma função Gaussiana ou uniforme.
- $\tau$  é o parâmetro de suavização (*largura de banda*), que controla o grau de influência dos pontos vizinhos na estimativa.
- $s - s_i$  é a diferença entre o ponto e os pontos observados  $s_i$ .
- $n$  é o número total de pontos ou eventos observados.

A função Kernel  $k(\cdot)$  calcula a influência dos pontos  $s_i$  ao redor de uma localização  $s$ . O parâmetro  $\tau$  determina a suavização de forma que:

Valores altos de  $\tau$  geram uma superfície mais uniforme, representando padrões globais. Isso pode mascarar áreas críticas ou surtos incipientes por “esconder” os detalhes e ocultar padrões importantes.

Valores baixos de  $\tau$  destacam padrões locais, com menor suavização. Isso pode confundir o que seria apenas um “ruído”, com um agrupamento real e, assim, resultar em um mapa de calor com muitos picos e vales.

A escolha do valor de  $\tau$  é fundamental, mas o melhor valor dependerá do tamanho da área, da densidade de casos e, sobretudo, da pergunta de saúde pública a ser respondida com a análise, ou seja, o objetivo da análise.

Muito interessante, não é mesmo? Nos próximos itens, entraremos em detalhes que, à primeira vista, parecem mais complicados, mas fazem toda a diferença para não criar artefatos enganosos na rotina da vigilância.

Primeiro, veremos como explorar as larguras de banda do kernel. Em seguida, explicaremos a correção de bordas, ajuste simples que impede que áreas próximas ao limite do mapa sejam subestimadas só porque têm vizinhos “faltando” fora da cena. Depois falaremos sobre as diferenças entre os tipos de kernel e algumas técnicas para cálculo do estimador Kernel.

Fique tranquilo: mostraremos exemplos fáceis, de modo que cada conceito se conecte às decisões tomadas no dia a dia pelos serviços de vigilância em saúde.

## *Escolhendo a largura de banda*

Vamos agora explorar, na prática, como diferentes **larguras de banda** na Estimativa de Densidade de Kernel afetam a suavização quando nossos dados estão espacialmente clusterizados. No R, podemos testar vários valores com a função `density()`. Essa função possui alguns parâmetros do Kernel como:

- **Largura de banda (sigma)**: Define o alcance de suavização do kernel.
- **Tipo de kernel (kernel)**: Permite escolher o formato do kernel.
- **Correção de bordas (diggle)**: Ajusta a densidade para considerar bordas da janela de observação.
- **Vetor de pesos (weights)**: Define pesos diferentes para os pontos.

Neste caso, vamos alterar o valor de `sigma` na função `density()` utilizando o objeto `agregado` que foi criado para os exemplos anteriores. Acompanhe o código abaixo:

```
kernel_cluster1 ← density(agregado, sigma = 0.03)
kernel_cluster2 ← density(agregado, sigma = 0.08)
kernel_cluster3 ← density(agregado, sigma = 0.15)

# vamos apresentar agora os gráficos em ggplot
library(ggplot2)
# vamos carregar a biblioteca patchwork para
# plotar os gráficos juntos
library(patchwork)

g_den1 ←
  ggplot(as_tibble(kernel_cluster1), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(agregado),
             color = "white") +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Largura de banda = 0,03") +
  guides(fill = "none") +
  theme_void()
```

```

g_den2 ←
  ggplot(as_tibble(kernel_cluster2), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(agregado),
             color = "white") +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Largura de banda = 0,08") +
  guides(fill = "none") +
  theme_void()

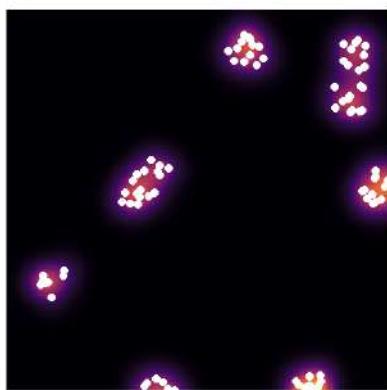
g_den3 ←
  ggplot(as_tibble(kernel_cluster3), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(agregado),
             color = "white") +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Largura de banda = 0,15") +
  guides(fill = "none") +
  theme_void()

(g_den1 | g_den2 | g_den3)

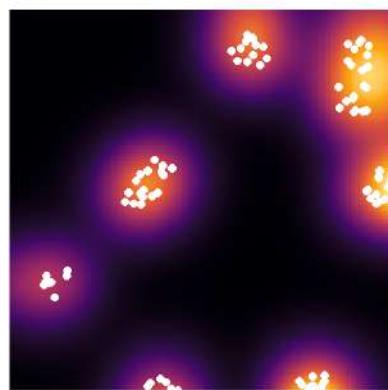
```

**Figura 10:** Exemplos da aplicação da função de kernel utilizando o padrão de pontos agregado variando as larguras de banda.

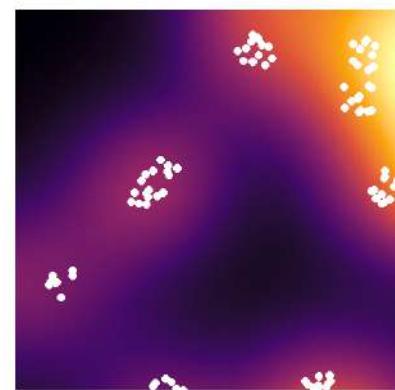
Largura de banda = 0,03



Largura de banda = 0,08



Largura de banda = 0,15





1. `kernel_cluster1 ← density(agregado, sigma = 0.03)`: Calcula a densidade de Kernel para os dados agregado com largura de banda ( $\sigma$ ) igual a 0.03, resultando em maior detalhamento (menos suavização).
2. `kernel_cluster2 ← density(agregado, sigma = 0.08)`: Calcula a densidade de Kernel para os mesmos dados, mas com largura de banda ( $\sigma$ ) igual a 0.08, resultando em uma suavização intermediária.
3. `kernel_cluster3 ← density(agregado, sigma = 0.15)`: Calcula a densidade de Kernel para os dados com  $\sigma = 0.15$ , o que gera uma superfície mais suavizada.
4. Os gráficos são gerados através dos seguintes argumentos da biblioteca `ggplot`:
  - `as_tibble(kernel_cluster1)`: Converte os dados de densidade calculados para um formato `tibble`, compatível com o `ggplot2`.
  - `aes(x, y)`: Define os eixos X e Y para o gráfico.
  - `geom_tile(aes(fill = value))`: Cria uma camada de azulejos coloridos (`tiles`) para representar a densidade (`value`).
  - `geom_point`: Adiciona os pontos originais da variável agregada, exibidos em branco para destacar sua posição no mapa de calor.
  - `scale_fill_viridis_c(option = "B")`: Define uma escala de cor contínua usando a paleta `viridis` (opção "B").
  - `labs(title)`: Adiciona título ao gráfico, indicando o valor da largura de banda (parâmetro  $\sigma$ ).
  - `guides(fill = "none")`: Remove a legenda da escala de cores.
  - `theme_void()`: Remove elementos gráficos como eixos e grades para focar apenas na densidade.

A Figura 10 mostra três mapas de calor gerados pelo método de estimativa de densidade de Kernel, aplicados a diferentes larguras de banda ( $\tau$ ) representadas pelos valores 0,03, 0,08 e 0,15. Cada painel corresponde a uma largura de banda específica:

- Largura de banda = 0,03: O mapa apresenta alta sensibilidade a variações locais, com áreas de concentração muito definidas e detalhadas. A suavização é mínima, evidenciando pequenos agrupamentos ou hotspots de alta densidade.
- Largura de banda = 0,08: O aumento da largura de banda reduz a granularidade da análise, resultando em uma distribuição mais suavizada. Os padrões locais começam a se mesclar, destacando áreas de concentração maiores, mas com menos detalhes.
- Largura de banda = 0,15: Com a maior largura de banda, a suavização é intensa, e os agrupamentos menores se fundem em grandes áreas de alta densidade. Este nível de suavização destaca padrões globais, mas pode ocultar variações locais.

Resumindo, a Figura 10 demonstra como a largura de banda no método Kernel afeta a suavização espacial. Larguras menores enfatizam detalhes locais, enquanto larguras maiores evidenciam padrões globais. A escolha da largura de banda deve equilibrar a necessidade de “capturar” detalhes locais sem perder a “visão geral” da distribuição dos eventos.

O pacote `ggplot2` também possui funções próprias que permitem a visualização de estimativas de densidade, incluindo o uso de Kernel espacial. Uma dessas funções é a `stat_density2d()`, que ajusta o Kernel diretamente na visualização e adiciona automaticamente as linhas de contorno, facilitando a interpretação das áreas de maior ou menor densidade. A largura de banda mais adequada para a estimação do Kernel é determinada automaticamente, através do pacote `MASS`. Abaixo, vamos comparar a largura de banda “ótima” com uma largura de banda arbitrária, escolhida sem muitos critérios. Acompanhe abaixo:

```

g_contour <-
  ggplot(as_tibble(agregado), aes(x, y)) +
  stat_density2d_filled(h = c(1, 1)) +
  stat_density2d(h = c(1, 1), n = 200, contour_var = "count") +
  geom_point(color = "white") +
  labs(title = "Largura de banda arbitrária (0,1)") +
  guides(fill = "none") +
  theme_void()

# vamos inserir linhas de contorno também

opt_bw <- c(MASS::bandwidth.nrd(agregado$x), MASS::bandwidth.nrd(agregado$y))

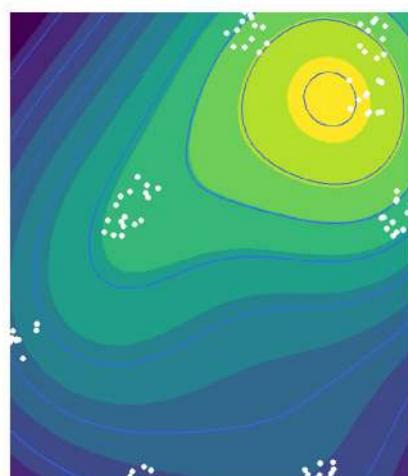
g_contour_opt <-
  ggplot(as_tibble(agregado), aes(x, y)) +
  stat_density2d_filled(h = opt_bw) +
  stat_density2d(h = opt_bw, n = 200, contour_var = "count") +
  geom_point(color = "white") +
  labs(title = "Largura de banda ótima") +
  guides(fill = "none") +
  theme_void()

(g_contour | g_contour_opt)

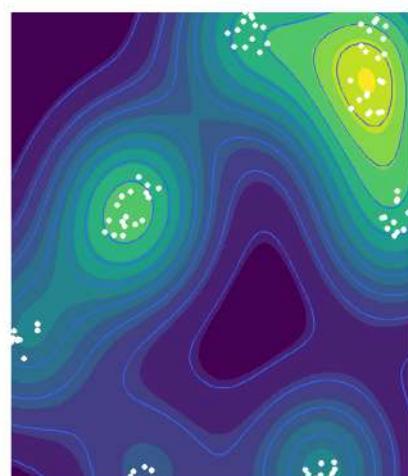
```

**Figura 11: Exemplos da aplicação da função de Kernel utilizando a largura de banda arbitrária e a ótima estimada.**

Largura de banda arbitrária (0,1)



Largura de banda ótima



Este código compara visualmente a Estimativa de Densidade de Kernel com largura de banda arbitrária e largura de banda otimizada usando `ggplot2`.

### 1. Gráfico 1: `g_contour`

Largura de banda arbitrária:  $h = c(1, 1)$ .

- `stat_density2d_filled()`: Preenche áreas de densidade.
- `stat_density2d()`: Adiciona linhas de contorno baseadas na densidade.
- `geom_point()`: Destaca os pontos originais em branco.

### 1. Gráfico 2: `g_contour_opt`

- Largura de banda otimizada: Calculada previamente com a função `bandwidth.nrd()` do pacote `MASS` e armazenada em `opt_bw`.

Na Figura 11 podemos observar que a largura de banda otimizada reflete a distribuição real dos dados com maior precisão, enquanto a largura arbitrária pode gerar resultados mais espúrios.

Mesmo com uma largura de banda bem escolhida, suas estimativas podem ser subestimadas nas periferias do mapa. Por isso, no item seguinte veremos correções de borda, muito útil quando a área de estudo é limitada (por exemplo, um bairro cercado por barreiras geográficas).

## *Estimativa de Kernel com correção por bordas*

Como citado anteriormente, o objetivo da correção por bordas na estimativa de densidade de Kernel é lidar com o viés introduzido nas extremidades da região de estudo. Ou seja, em áreas próximas às bordas, o interpolador Kernel pode saltar para fora da região de interesse, subestimando a densidade real, pois há menos pontos dentro do raio de influência do Kernel comparado ao que ocorre no interior da região.

A correção por bordas lida com esse problema ao considerar apenas a contribuição efetiva do Kernel dentro da região de estudo. Isso é feito normalizando o volume do Kernel sobre a área que está realmente dentro da região.

Para calcular essa estimativa, primeiramente, calculamos o volume sob o Kernel que está efetivamente dentro da região de estudo, representado por:

$$\delta_\tau(s) = \int_R \frac{1}{\tau^2} k\left(\frac{s-u}{\tau}\right) du$$

Onde:

- $\delta_\tau(s)$  é o volume sob o Kernel ajustado pela região de interesse.
- $k(\cdot)$  é a função Kernel escolhida.
- $s - u$  é a diferença entre a localização de interesse ( $s$ ) e os pontos dentro da região ( $u$ ).
- $\tau$  é o parâmetro de largura de banda que controla a suavização.
- $R$  é a região de estudo.

Após calcular essa correção, o estimador de densidade ajustado é dado por:

$$\hat{\lambda}(s) = \frac{1}{\delta_\tau(s)} \sum_{i=1}^n \frac{1}{\tau^2} k\left(\frac{s - s_i}{\tau}\right)$$

Onde:

- $\hat{\lambda}(s)$  é a estimativa da densidade corrigida.
- $\delta_\tau(s)$  é o fator de correção do Kernel calculado anteriormente.
- $n$  é o número total de pontos observados.
- $k(\cdot)$  é a função Kernel.
- $s - s_i$  é a diferença entre a localização de interesse ( $s$ ) e os pontos observados ( $s_i$ ).
- $\tau$  é o parâmetro de largura de banda.

Quando falamos em correção por bordas e diversas fórmulas, pode parecer complicado, mas não se preocupe! A correção por bordas é feita acrescentando o parâmetro `diggle = TRUE` na função `density()`, que já vimos. Agora, vamos comparar a função de Kernel utilizando o padrão de pontos agregado e a mesma largura de banda com e sem a correção por bordas. Veja o código abaixo:

```
kernel_cluster1 ← density(agregado, sigma = 0.08)
kernel_cluster2 ← density(agregado, sigma = 0.08, diggle = TRUE)

# vamos apresentar agora os gráficos em ggplot

g_den1 ←
  ggplot(as_tibble(kernel_cluster1), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(agregado),
             color = "white") +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Sem correção por bordas",
       subtitle = "Largura de banda = 0,08") +
  guides(fill = "none") +
  theme_void()
```

```

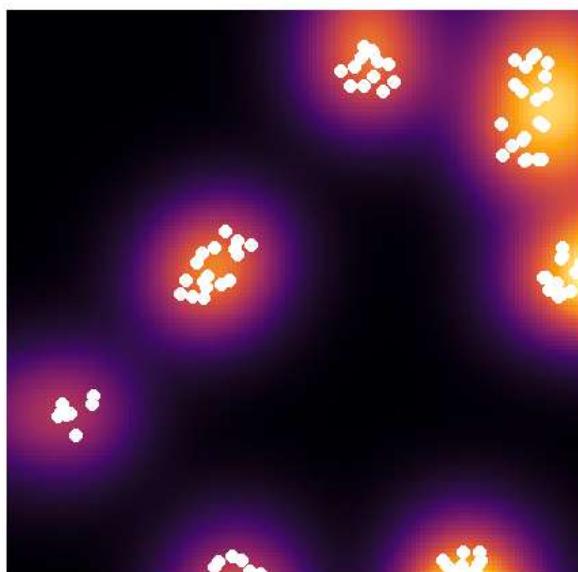
g_den2 <-
  ggplot(as_tibble(kernel_cluster2), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(agregado),
             color = "white") +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Com correção por bordas",
       subtitle = "Largura de banda = 0,08") +
  guides(fill = "none") +
  theme_void()

(g_den1 | g_den2)

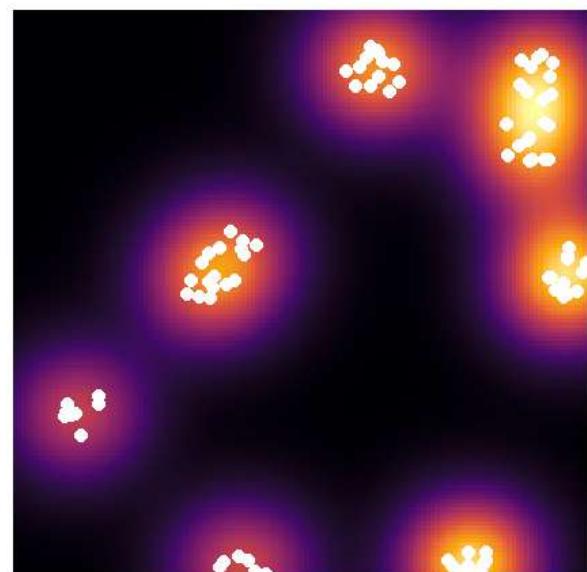
```

**Figura 12:** Exemplos da aplicação da função de Kernel com e sem a correção por bordas.

Sem correção por bordas  
Largura de banda = 0,08



Com correção por bordas  
Largura de banda = 0,08



Este código compara a estimativa de densidade de Kernel com e sem correção por bordas, utilizando o mesmo conjunto de dados e largura de banda ( $\sigma = 0.08$ ).

- `kernel_cluster1 ← density(agregado, sigma = 0.08)`: Calcula a densidade de Kernel para os dados agregado sem aplicar a correção por bordas.
- `kernel_cluster2 ← density(agregado, sigma = 0.08, diggle = TRUE)`: Calcula a densidade de Kernel para os mesmos dados, mas agora com a correção por bordas ativada (`diggle = TRUE`).

Observe na Figura 12 que a correção por bordas aumentou a intensidade nos pontos na borda inferior e lateral direita. A correção por bordas tende a ajustar a densidade para compensar o viés nas bordas da área de estudo, resultando em estimativas mais precisas para regiões próximas aos limites.

Viu como foi fácil? Em seguida, vamos explorar as diferenças entre tipos de interpolador Kernel e como isso pode impactar a análise.



## *Diferenças entre as funções de Kernel*

A função Kernel é um componente essencial na estimativa de densidade de Kernel, pois define como os pontos vizinhos influenciam a estimativa de densidade em um ponto específico. Mas, diferentes funções de Kernel podem resultar em superfícies de densidade distintas, mesmo com a mesma largura de banda. Muitos softwares (como o QGIS, por exemplo), padronizam o uso de uma só função, apesar que disponibilizar outras opções. Neste curso, veremos três tipos de Kernel: **Gaussiano**, **Quártico** e o **Disc**. É importante entender suas diferenças e o impacto na estimativa de densidade, mesmo com a mesma largura de banda.

Para visualizarmos as diferenças entre as funções de Kernel, vamos aplicar três tipos diferentes de Kernel ao mesmo conjunto de dados (o padrão agregado) e com a mesma largura de banda. Acompanhe o código abaixo e, em seguida, vamos discutir os resultados.

```
kernel_cluster1 ← density(agregado, kernel = "gaussian", sigma = 0.08)
kernel_cluster2 ← density(agregado, kernel = "quartic", sigma = 0.08)
kernel_cluster3 ← density(agregado, kernel = "disc", sigma = 0.08)

# vamos apresentar agora os gráficos em ggplot

g_den1 ←
  ggplot(as_tibble(kernel_cluster1), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(agregado),
             color = "white") +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Kernel Gaussiano",
       subtitle = "Largura de banda = 0,08") +
  guides(fill = "none") +
  theme_void()
```

```

g_den2 ←
  ggplot(as_tibble(kernel_cluster2), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(agregado),
             color = "white") +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Kernel Quártico",
       subtitle = "Largura de banda = 0,08") +
  guides(fill = "none") +
  theme_void()

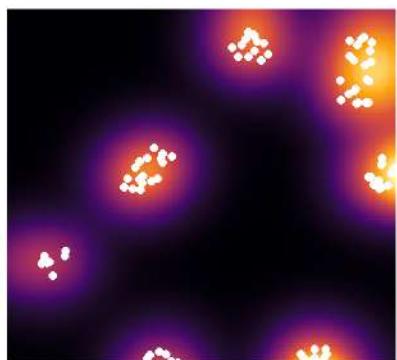
g_den3 ←
  ggplot(as_tibble(kernel_cluster3), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(agregado),
             color = "white") +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Kernel Disc",
       subtitle = "Largura de banda = 0,08") +
  guides(fill = "none") +
  theme_void()

(g_den1 | g_den2 | g_den3)

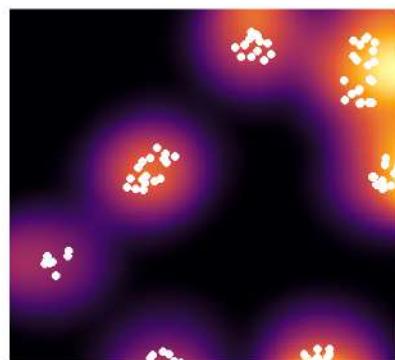
```

**Figura 13: Exemplos da aplicação de diferentes funções de Kernel para o mesmo conjunto de dados e com a mesma largura de banda.**

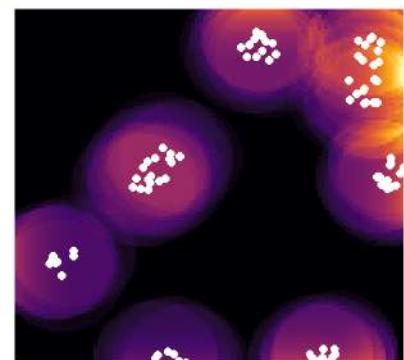
Kernel Gaussiano  
Largura de banda = 0,08



Kernal Quártico  
Largura de banda = 0,08



Kernel Disc  
Largura de banda = 0,08



### Cálculo das Densidades:

1. `kernel_cluster1 ← density(agregado, kernel = "gaussian", sigma = 0.08):`  
Calcula a densidade de Kernel com a função Gaussiana (a padrão) para os dados agregado, usando largura de banda  $\sigma = 0.08$ .
2. `kernel_cluster2 ← density(agregado, kernel = "quartic", sigma = 0.08):`  
Calcula a densidade de Kernel usando a função Quartic (também conhecida como *Epanechnikov*), com a mesma largura de banda.
3. `kernel_cluster3 ← density(agregado, kernel = "disc", sigma = 0.08):` Calcula a densidade de Kernel usando a função Disc, que aplica uma influência uniforme dentro de um raio.

Observando a Figura 13, podemos notar diferenças importantes entre os tipos de Kernel. Vamos analisar cada um deles:

- O **Gaussiano** utiliza uma distribuição normal, o que significa que os pontos próximos ao centro do Kernel recebem pesos maiores, mas o decrescimento do peso ocorre de forma contínua e suave à medida que a distância aumenta. As transições de cores são suaves e graduais, refletindo a suavização contínua típica do Kernel Gaussiano. Isso resulta em áreas de maior densidade (em laranja/vermelho) são bem definidas, mas as bordas das áreas têm transições gradativas, o que cria uma aparência mais difusa.
- O **Quártico** (também conhecida como *Epanechnikov*) dá mais peso a pontos próximos do que a pontos distantes, mas o decrescimento do peso é gradual e termina abruptamente em um limite definido (como uma “bolha”). As áreas de alta densidade permanecem bem definidas, mas as transições para áreas de menor densidade são ligeiramente menos difusas que no Kernel Gaussiano. Cria uma suavização intermediária, que destaca os agrupamentos sem perder muita precisão local.
- O **Disc** dá peso igual a todos os pontos dentro de um raio fixo e ignora pontos fora desse raio. Isso cria áreas de densidade com contornos abruptos e sem transições suaves. As áreas de alta densidade aparecem como “manchas” com bordas bem definidas, sem transições graduais para áreas de menor densidade. Esse Kernel é ideal para destacar claramente os agrupamentos locais, mas sacrifica a suavidade dos resultados.

Mas, no dia a dia da vigilância em saúde, qual função Kernel devemos escolher? **Depende do objetivo!** Aqui estão algumas orientações:

- Gaussiano: se o objetivo for explorar padrões gerais de distribuição dos pontos, o Kernel Gaussiano pode ser mais adequado, pois destaca padrões gerais em grandes áreas.
- Quártico: se o objetivo for identificar agrupamentos locais, com dados clusterizados, o Kernel Quártico pode ser mais apropriado, pois mantém detalhes locais e balanceia a suavização nas áreas mais distantes.
- Disc: se o objetivo for detectar áreas “hotspots”, com agrupamentos nítidos, o Kernel Disc pode ser mais apropriado, pois destaca claramente os pontos de alta densidade.

Essas escolhas permitem ajustar a suavização aos objetivos específicos da análise espacial. Mas, a análise utilizando estimadores de Kernel não se limita apenas a suavizar a distribuição dos eventos. Também é possível incorporar atributos e calcular razões entre diferentes eventos, o que pode enriquecer ainda mais a análise espacial. Vamos ver como isso funciona na prática nos próximos itens.

## *Kernel por atributo (kernel ponderado)*

Na rotina da vigilância, nem sempre basta saber onde ocorrem os casos; é crucial relacioná-los a um denominador: por exemplo, população exposta, leitos disponíveis, número de criadouros, entre outros. A Estimativa de Densidade de Kernel resolve isso facilmente: basta atribuir um peso a cada ponto, representando o valor do atributo de interesse. Assim, a estimativa de densidade ponderada é calculada como uma média ponderada dos pontos vizinhos, levando em conta o valor do atributo definido.

Neste sentido, o método de suavização utilizando o Kernel permite incorporar uma covariável (atributo) para refinar a estimativa. Por exemplo, é possível estimar a população por unidade de área ou calcular a razão entre dois Kernels, gerando uma estimativa suavizada de eventos por população.

A fórmula para a estimativa por atributo ( $y_i$ ) é:

$$\hat{\lambda}(s) = \sum_{i=1}^n \frac{1}{\tau^2} k \left( \frac{s - s'_j}{\tau} \right) y_i$$

Onde:

- $\hat{\lambda}(s)$  é a estimativa do atributo para unidade de área.
- $\tau$  é a largura de banda.
- $y_i$  é o valor do atributo associado a cada ponto  $i$ .
- $k$  é a Função Kernel, que define como os valores dos pontos vizinhos contribuem para a estimativa.

Este tipo de análise permite representar espacialmente tanto a distribuição do atributo (ex.: população) quanto a intensidade de eventos proporcional ao atributo, fornecendo uma visão mais detalhada e ajustada à realidade.



### Quadro 3: Exemplo de aplicações de Kernel por atributo.

Atributo	Como	Pergunta	Aplicação
População do setor censitário (habitantes)	Atribuindo o número de habitantes ao centroide ou ao centro populacional da área	Onde se concentram mais pessoas?	Planejar vacinação, larvicida
Eventos / População	Calculando a razão de eventos (ex.: casos de doença) em relação à densidade populacional.	Qual é o risco por 1.000 hab.?	Comparar bairros ajustando pelo tamanho populacional

Mas, como isso funciona na prática? O próximo passo agora é combinar duas dessas superfícies: dividir o kernel de casos pelo kernel de população (ou por outro denominador relevante) para obter uma **razão de kernels**, isto é, um mapa suavizado de risco ou taxa. No próximo item vamos ver, na prática, como gerar essas duas camadas no R e calcular a razão de forma rápida, usando as mesmas funções que acabamos de comentar.

## Razão de Kernel

Até aqui, vimos como gerar uma superfície suavizada para qualquer conjunto de pontos. Na rotina da vigilância, porém, o que mais interessa em determinados momentos é a taxa. Por exemplo, quantos eventos ocorrem para cada habitante (ou para cada leito, criadouro, domicílio visitado, entre outros).

A razão de Kernel faz exatamente isso. É uma técnica utilizada para calcular uma taxa suavizada a partir da divisão de duas estimativas de Kernel. Esse método é especialmente útil em análises espaciais que relacionam eventos com outros eventos (por exemplo, casos e controles) e também relacionar eventos e atributos (por exemplo, a densidade de eventos por população).

A razão de Kernel é a divisão da estimativa de densidade de kernel pela estimativa de densidade de kernel ponderada por um atributo. A fórmula é a seguinte:

$$\hat{\lambda}(s) = \frac{\sum_{i=1}^n \frac{1}{\tau^2} k\left(\frac{s-s_i}{\tau}\right)}{\sum_{i=1}^n \frac{1}{\tau^2} k\left(\frac{s-s'_i}{\tau}\right) y_i}$$

Dessa forma, temos que:

1. Numerador: Representa o alisamento dos eventos por unidade de área, considerando a função Kernel  $k$  para distribuir a densidade ao redor dos pontos de interesse  $s_i$ ;
2. Denominador: Representa o alisamento da covariável  $y_i$  (por exemplo, população) por unidade de área, utilizando a mesma largura de banda  $\tau$  e função Kernel  $k$ ;
3.  $\tau$ : Parâmetro de suavização (largura de banda) que controla o grau de influência dos pontos vizinhos na estimativa.
4.  $y_i$ : Atributo associado a cada ponto  $i$ , como o número de habitantes ou outro fator relevante.
5.  $s_i$ : Localização dos eventos de interesse.



**Observação:** Para a razão de Kernel também é possível utilizar diferentes larguras de banda no numerador e denominador (em geral maior no denominador para estabilizar mais) e também outro evento pontual como “estimador da população a risco”.

Vamos praticar no R? Nossa pequena tarefa, será criar uma “taxa suavizada”, produzir duas superfícies suavizadas e dividi-las, ponto a ponto. Para fixar o conceito, vamos montar dois conjuntos de eventos no R:

- casos: pontos gerados de forma clusterizada, simulando uma doença realmente concentrada em alguns focos;
- controles: pontos distribuídos quase ao acaso, servindo apenas de “fundo” populacional.

O resultado é a razão de Kernel, que reflete diferenças relativas entre os dois padrões:

- Valores maiores que 1 indicam maior densidade relativa no padrão agregado.
- Valores menores que 1 indicam menor densidade no padrão agregado em relação ao aleatório.

Acompanhe o código abaixo:

```
bandwidth <- 0.10
kernel_random <- density(aleatorio, sigma = bandwidth)
kernel_cluster <- density(agregado, sigma = bandwidth)

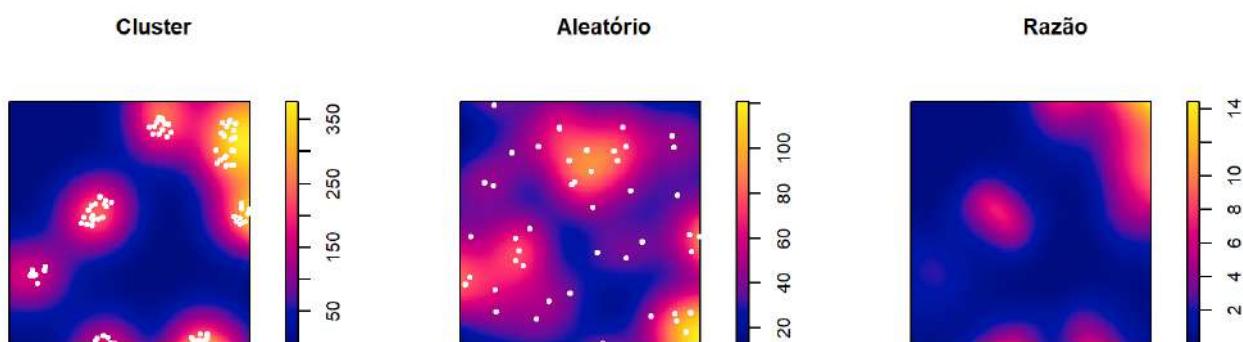
clu_alea_ratio <- kernel_cluster/kernel_random

par(mfrow = c(1, 3))
plot(kernel_cluster, main = "Cluster")
plot(agregado, add = T, col = "white", pch = 20)

plot(kernel_random, main = "Aleatório")
plot(aleatorio, add = T, col = "white", pch = 20)

plot(clu_alea_ratio, main = "Razão")
```

**Figura 14: Exemplos da aplicação da função da razão de Kernel utilizando casos clusterizados VS casos aleatórios.**



Utilizando a largura de banda de 0.10 (`bandwidth ← 0.10`) este código utiliza estimativas de densidade de Kernel para calcular e visualizar:

1. `kernel_random ← density(aleatorio, sigma = bandwidth)`: A densidade de eventos em um padrão aleatório.
2. `kernel_cluster ← density(agregado, sigma = bandwidth)`: A densidade em um padrão agregado.
3. `clu_alea_ratio ← kernel_cluster / kernel_random`: A razão entre as densidades (agregado/aleatório), que resulta em uma análise ajustada ou comparativa.

Na Figura 14, note que o mapa da razão de Kernel (imagem da direita) conserva, em amarelo e vermelho, as regiões em destaque que apareciam como “áreas quentes” no kernel de casos (imagem da esquerda). Isso indica que esses agrupamentos permanecem evidentes mesmo depois da padronização. Em outras palavras, a razão de Kernel permite identificar padrões relativos e destacar áreas de concentração que seriam menos visíveis apenas com as densidades absolutas.

Agora, vamos conhecer os processos pontuais de segunda ordem, que investigam a interação entre os eventos como, por exemplo, a Função K e G. Vamos lá?

## *Análise de um processo pontual de segunda ordem (funções G e K)*

Até agora exploramos técnicas que analisam **onde** os eventos ocorrem com maior ou menor intensidade, usando estimativas como a densidade por Kernel. Contudo, muitas vezes, na vigilância em saúde, não basta saber se há muitos casos em uma região. Cabe ao analista e ao técnico da vigilância o entendimento se os casos interagem espacialmente uns com os outros, formando agrupamentos reais ou mantendo-se dispersos por algum motivo.

Para responder a esse tipo de indagação, utilizamos métodos chamados de **análise de processos pontuais de segunda ordem**. Essas abordagens avaliam diretamente as distâncias e relações entre eventos, investigando se há interação espacial significativa. Para isso, empregamos duas ferramentas clássicas e muito importantes, conhecidas como funções G e K.

Essas funções têm sido amplamente utilizadas na vigilância em saúde para exploração inicial de padrões de agregação ou dispersão espacial em uma área. Dessa forma, ajudam a identificar áreas de risco e a entender a dinâmica de doenças e outros fenômenos espaciais.

Nos próximos itens, vamos, primeiramente, conhecer os conceitos. E, mais adiante, vamos aplicá-las na prática de R, utilizando exemplo de dados espaciais, sempre focando no seu uso nas análises da rotina da vigilância em saúde.

## Função G - Distância do vizinho mais próximo

A função G é uma ferramenta estatística que mede a distância do vizinho mais próximo (ou seja, o evento mais próximo) em relação a um ponto específico. É útil na vigilância em saúde por ser aplicada na detecção de padrões espaciais, como agrupamentos ou dispersões.

O método do vizinho mais próximo é utilizado para estimar a função de distribuição cumulativa  $\hat{G}(r)$ , que se baseia nas distâncias  $r$  entre eventos dentro de uma região de análise.

A função de distribuição cumulativa pode ser estimada empiricamente pela fórmula:

$$\hat{G}(r) = \frac{\#(d(u_i, u_j) \leq r)}{n}$$

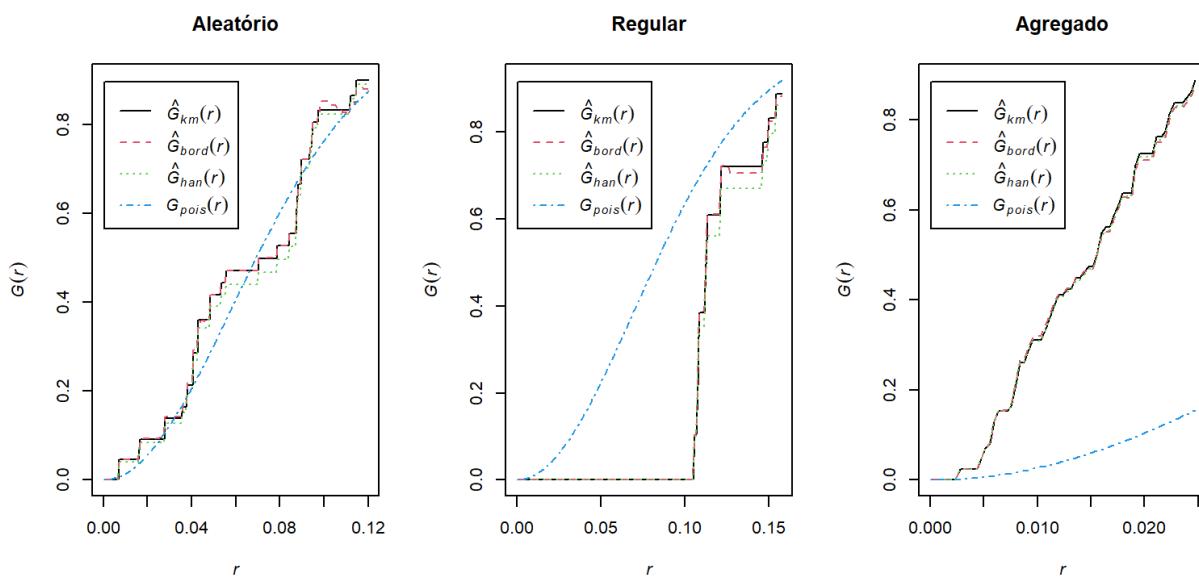
Para a qual:

- $d(u_i, u_j)$  é a distância entre dois eventos  $u_i$  e  $u_j$ .
- $r$  é o limite de distância considerado.
- $n$  é o número total de eventos na região de análise.

A análise gráfica de  $\hat{G}(r)$  serve como uma ferramenta exploratória para identificar a existência de interações espaciais entre eventos. Os seguintes padrões podem ser observados:

- Um crescimento rápido da função para pequenas distâncias  $r$  pode indicar interações entre eventos, sugerindo agrupamento ou padrões de aglomeração em escalas menores.
- Valores baixos de  $\hat{G}(r)$  para distâncias pequenas, seguidos de crescimento lento, podem sugerir uma distribuição mais regular e menos interativa entre os eventos.

**Figura 15: Estimativas da função G para três diferentes padrões de pontos: Aleatório, Regular e Agregado.**



Os gráficos da Figura 15 apresentam as estimativas da função de distribuição  $\hat{G}(r)$  para os seguintes padrões espaciais: Aleatório, Regular e Agregado. Cada gráfico compara o comportamento empírico de  $\hat{G}(r)$  com a função teórica  $G_{pois}(r)$ , que representa um padrão de distribuição aleatório (processo de Poisson). Abaixo está a interpretação de cada caso:

- 1. Aleatório (Primeiro Gráfico):** O comportamento de  $\hat{G}(r)$  é muito próximo da função  $G_{pois}(r)$ , sugerindo que os eventos não apresentam interação espacial e estão distribuídos de forma aleatória. Pequenas variações são esperadas devido a flutuações empíricas.
- 2. Regular (Segundo Gráfico):** A curva empírica  $\hat{G}(r)$  cresce mais lentamente em comparação com  $G_{pois}(r)$ , indicando uma distribuição mais espaçada dos eventos. Esse padrão é característico de processos regulares, nos quais os eventos mantêm uma distância mínima entre si, sugerindo repulsão.
- 3. Agregado (Terceiro Gráfico):** A curva  $\hat{G}(r)$  cresce mais rapidamente para valores baixos de  $r$  em relação a  $G_{pois}(r)$ , indicando que os eventos estão agrupados ou formam aglomerações em pequenas distâncias. Este padrão é típico de processos que apresentam interações positivas ou atração entre os eventos.

Embora o método do vizinho mais próximo seja útil para fornecer uma indicação inicial sobre a distribuição espacial dos eventos, ele é limitado a pequenas escalas espaciais, o que pode restringir a análise de padrões em distâncias maiores. Para obter uma visão mais abrangente e efetiva do padrão espacial em escalas maiores, recomenda-se o uso da função  $G$ , que permite uma análise mais detalhada ao considerar uma gama mais ampla de distâncias e a interação entre os eventos em diferentes escalas.

## *Função K de Ripley (ou apenas função K)*

A função  $K$ , também conhecida como medida de momento de segunda ordem reduzida, é uma ferramenta amplamente utilizada na análise de padrões espaciais. Ela é definida para um processo univariado como:

$$\lambda K(h) = E(\# \text{ de eventos contidos até uma distância } h \text{ de um evento arbitrário}),$$

Onde:

- $\lambda$  é a intensidade ou o número médio de eventos por unidade de área, assumido constante na região de estudo.
- $E()$  é o operador de expectativa.
- $h$  é a distância considerada.

A fórmula estimada para  $K(h)$  é dada por:

$$\hat{K}(h) = \frac{A}{n^2} \sum_{i=1}^n \sum_{j \neq i} w_{ij}^{-1} I[x_j : d(x_i, x_j)],$$

Onde:

- $A$  é a área da região de análise.
- $n$  é o número total de eventos.
- $w_{ij}$  é o fator de correção de borda.
- $d(x_i, x_j)$  é a distância entre os pontos  $x_i$  e  $x_j$ .

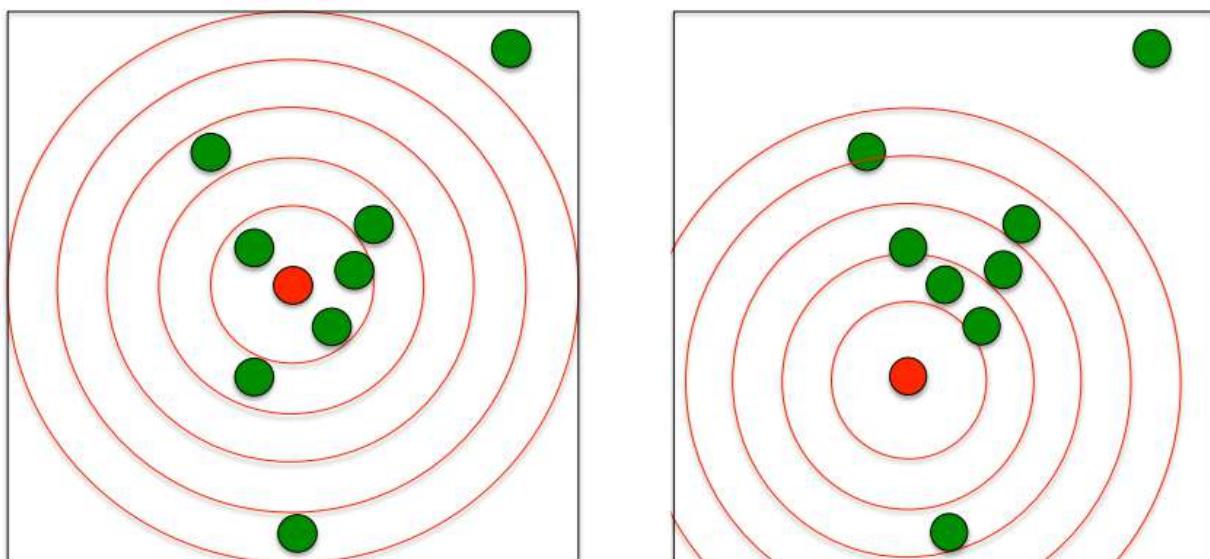
A função  $K$  mensura quantos eventos estão distribuídos em círculos concêntricos ao redor de um ponto de referência (chamado de ponto focal). Esses círculos partem de um raio inicial igual a zero e aumentam gradativamente até cobrir toda a área de estudo.

### Principais características:

- A função  $K$  é cumulativa, representando o número esperado de eventos em círculos de raio  $t$ , centrados em cada ponto da região de estudo.
- A intensidade  $\lambda$  normaliza os resultados, permitindo uma comparação entre diferentes escalas ou regiões.

A Figura 16 ilustra círculos concêntricos ao redor de pontos focais, demonstrando como a função  $K$  considera diferentes distâncias para analisar padrões espaciais. Essa análise permite identificar se os eventos seguem um padrão aleatório, regular ou agregado. A distribuição é cumulativa, representando o número esperado de vizinhos em um círculo de raio  $t$  centrado em um ponto arbitrário, normalizado pela intensidade  $\lambda$  do padrão de pontos na área de estudo, proporcionando uma ferramenta poderosa para análise em diferentes escalas espaciais.

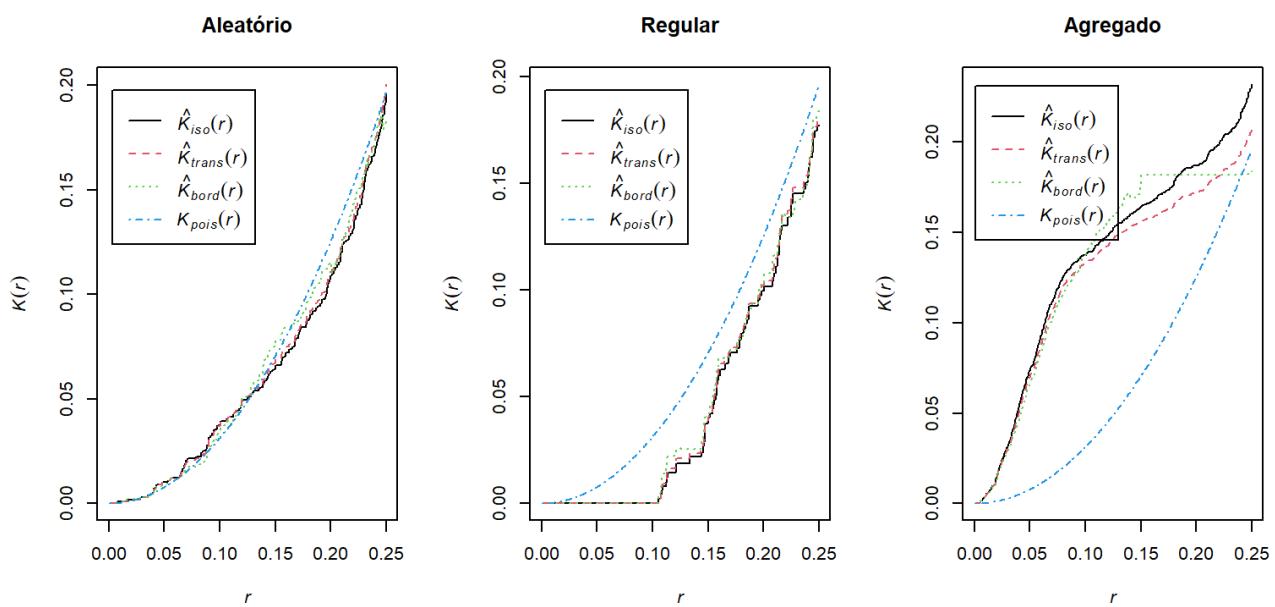
**Figura 16: Representação de Círculos Concêntricos para Análise da Função K.**



## *Padrões Espaciais Detectados pela Função*

A Figura 17 apresenta as curvas da função  $K(r)$  para três padrões espaciais: Aleatório, Regular e Agregado. Cada gráfico compara as diferentes estimativas empíricas ( $\hat{K}_{iso}(r)$ ,  $\hat{K}_{trans}(r)$ ,  $\hat{K}_{bord}(r)$ ) com a função teórica  $K_{pois}(r)$ , que corresponde a um padrão de completa aleatoriedade espacial (CSR). A comparação entre as estimativas empíricas e a curva teórica permite diferenciar padrões espaciais. O desvio das curvas empíricas em relação a  $K_{pois}(r)$  revela a natureza dos eventos: aleatórios, regulares ou agregados.

**Figura 17: Estimativas da função K para três diferentes padrões espaciais: Aleatório, Regular e Agregado.**



- **Padrão Aleatório (Gráfico à Esquerda):** As curvas empíricas estão próximas de  $K_{pois}(r)$ , indicando que os eventos seguem um padrão espacial aleatório, sem tendência de agregação ou repulsão. Isto ocorre pois, se os eventos são distribuídos de forma aleatória,  $\hat{K}(r)$  permanece próximo ao valor de referência teórico  $\pi r^2$ .
- **Padrão Regular (Gráfico Central):** As curvas empíricas estão abaixo de  $K_{pois}(r)$ , sugerindo repulsão entre os eventos  $\hat{K}(r) < K_{pois}(r)$ . Isso reflete uma distribuição regular, na qual os eventos mantêm uma distância mínima entre si.
- **Padrão Agregado (Gráfico à Direita):** Quando os eventos estão agrupados ou aglomerados, o número de eventos dentro de pequenas distâncias é maior que o esperado para uma distribuição aleatória. Nesse caso,  $\hat{K}(r) > K_{pois}(r)$ , e a curva da função  $\hat{K}$  está acima da linha de referência, evidenciando a interação positiva entre os eventos.

## *Detecção de cluster*

Entre as técnicas de análise espacial utilizadas em epidemiologia e vigilância em saúde, a detecção de clusters ocupa uma posição especialmente relevante. Essa abordagem envolve a identificação de padrões de aglomeração de eventos pontuais, como os vistos anteriormente, ou regiões onde há um excesso significativo de risco. Esses padrões incluem surtos de doenças, acidentes e outros eventos que demandam atenção especial e ações direcionadas.

Para identificar esses padrões de forma objetiva, é essencial utilizar métodos estatísticos específicos que permitem detectar e quantificar clusters espaciais. Neste curso, veremos dois tipos principais de testes para a detecção de clusters: **genéricos** e **focados**.

## *Testes genéricos de detecção de clusters*

Os testes genéricos são aplicados quando não existe uma hipótese prévia sobre a localização de um possível cluster. Esses testes analisam todo o espaço de estudo à procura de áreas com concentração incomum de eventos. Entre os testes genéricos mais utilizados, destacamos:

### i) Estatística de varredura de Kulldorff (SaTScan)

Este método utiliza janelas móveis de diferentes tamanhos para identificar regiões com alta incidência de eventos. Pode ser aplicado a modelos estatísticos diversos, como Poisson, Bernoulli ou Normal.

Exemplo prático: detectar áreas onde ocorrem surtos inesperados de doenças, sem que haja uma suposição prévia sobre onde esses clusters possam estar.

### ii) Índice I de Moran

O Índice I de Moran mede a autocorrelação espacial global, verificando se há padrões significativos de agrupamento espacial no conjunto de dados. Embora não indique exatamente a localização dos clusters, ele permite confirmar se existe dependência espacial.

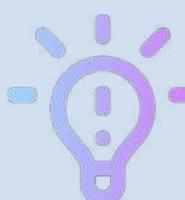
Exemplo prático: identificar padrões globais de altas taxas de mortalidade em uma cidade. Voltaremos a discutir esse método mais adiante no curso.

### iii) Teste de Knox

O teste de Knox é especialmente útil para detectar clusters espaço-temporais. Ele avalia se eventos próximos no espaço também ocorrem próximos no tempo, indicando padrões que não são resultado do acaso, como epidemias ou surtos.

Esse teste parte da hipótese nula de que as ocorrências dos eventos são independentes tanto no espaço quanto no tempo. Caso contrário, se existir associação espacial e temporal significativa, isso sugere fortemente a presença de um cluster espaço-temporal.

Exemplo prático: identificar se casos de uma epidemia apresentam uma concentração significativa tanto no espaço quanto no tempo.



**Observação:** Técnicas vistas anteriormente para análise de processos pontuais de segunda ordem, como as funções G e K, também são frequentemente utilizadas em conjunto com essas abordagens para apoiar a detecção de clusters.

## *Testes focados de detecção de clusters*

Ao contrário dos testes genéricos, os testes focados são aplicados quando já existe um local específico de interesse e pretende-se verificar se há um cluster ao redor dele. Esse tipo de análise é indicado especialmente quando existe uma hipótese prévia sobre onde o cluster pode ocorrer. Entre os testes focados, destacamos:

### i) Estatística de varredura focada de Kulldorff (SaTScan Focado)

Semelhante à versão genérica, esse método testa especificamente a presença de clusters em torno de pontos pré-determinados.

Exemplo prático: investigar se há aumento significativo de casos de câncer ao redor de uma usina nuclear.

## *Pratica em R*

Até aqui, você conheceu diferentes abordagens teóricas sobre processos pontuais. Agora, chegou a hora de aplicarmos esses conceitos com dados reais, utilizando o R.

Nesta parte prática, trabalharemos com dados referentes à ocorrência de homicídios, suicídios e acidentes de trânsito registrados na cidade de Porto Alegre, Rio Grande do Sul. Ao longo deste exercício, vamos explorar como:

- Preparar os dados espaciais;
- Aplicar métodos estatísticos para testar a Completa Aleatoriedade Espacial (CSR);
- Analisar processos pontuais de primeiro e segunda ordem
- Interpretar resultados em um contexto de saúde pública.

Abra o R e siga atentamente as orientações abaixo.

## Baixando e preparando os dados

**library**(tidyverse)

**library**(spatstat)

**library**(splancs)

Chamando algumas bibliotecas para as análises:

**tidyverse**: Um conjunto de pacotes no R (como ggplot2, dplyr, tidyr, entre outros) para manipulação, visualização, e análise de dados. É usado para transformar, organizar e visualizar dados de forma eficiente e intuitiva.

**spatstat**: Pacote para análise de padrões espaciais, especialmente voltado para processar e modelar pontos georreferenciados, como análise de densidade e estatísticas de pontos em superfícies bidimensionais.

**splancs**: Pacote para análise de dados espaciais em R, com foco em cálculos de geometria (como polígonos e coordenadas), análise de pontos no espaço e medidas espaciais básicas.

```
# Lendo os bancos que estão no repositório github
local <- "https://raw.githubusercontent.com/ogcruz/dados_eco_2023/main/dados/"

homic <- read.table(paste0(local, "homic.dat"), col.names = c("x", "y"))
suic <- read.table(paste0(local, "suic.dat"), col.names = c("x", "y"))
acid <- read.table(paste0(local, "acid.dat"), col.names = c("x", "y"))

# Plotando os casos de homicídios em um plano cartesiano
g <- ggplot(homic) +
  geom_point(aes(x, y, color = "Homicídios"), shape = 1) +
  labs(color = "")
```



## Distribuição Espacial dos Homicídios em Porto Alegre (RS) no plano cartesiano.



Este código está realizando as seguintes etapas:

1. Definindo o local dos dados: A variável local armazena a URL base para acessar arquivos do repositório GitHub.
2. Lendo os dados: Os arquivos `homic.dat`, `suic.dat`, e `acid.dat` são lidos diretamente da URL. Cada arquivo é tratado como uma tabela e recebe os nomes de colunas `x` e `y`, que presumivelmente representam coordenadas espaciais.
3. Visualização dos homicídios: Criando um gráfico usando o `ggplot2`, onde os pontos do dataset `homic` são plotados em um plano cartesiano (`x` e `y`) com uma legenda indicando “Homicídios”. Os pontos são representados por um símbolo circular vazio (`shape = 1`), e a legenda é personalizada removendo o título (`labs(color = "")`).

Porto Alegre é uma cidade disposta ao longo do eixo norte/sul. O gráfico perdeu a estrutura espacial, ajustando para o tamanho e forma da janela. Por isso é necessário informar ao programa que este tipo de objeto é um objeto espacial e tem uma escala, que deve ser preservada. Vamos fazer isso com transformando-o em um objeto espacial do pacote `sf`.

#### Library(sf)

A biblioteca `sf` (*simple features*) é usada para manipular e analisar dados geoespaciais no `R`. Ela permite trabalhar com objetos espaciais (como pontos, linhas e polígonos) em formatos padronizados, facilita a integração com bancos de dados geoespaciais e oferece suporte a operações como transformação de projeções, manipulação de geometrias e análise espacial.

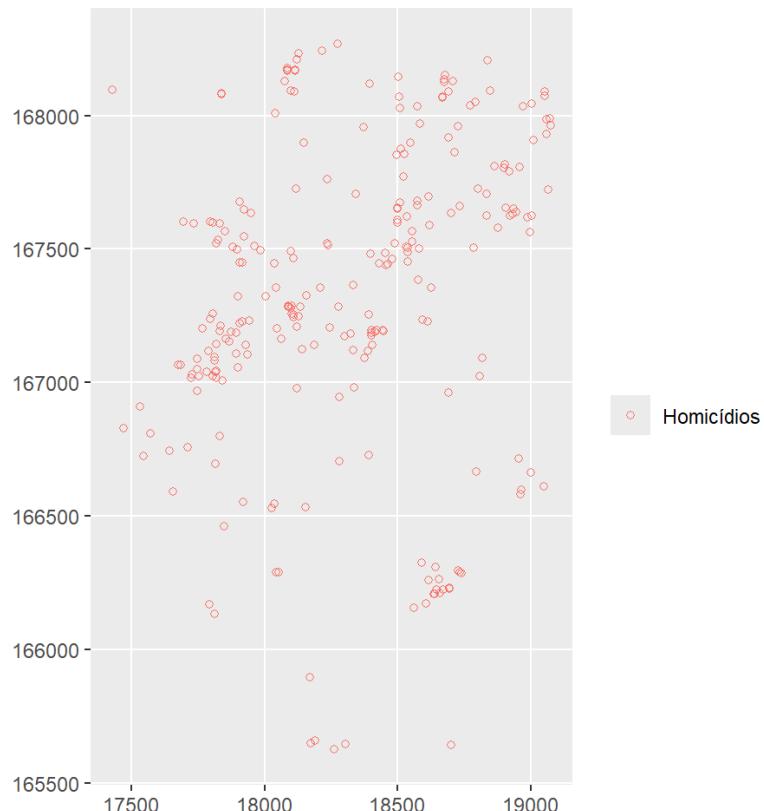
```
# Transformando os pontos em geometria espacial de pontos

homic_sf ← homic ▷
  st_as_sf(coords = c("x", "y"))
suic_sf ← suic ▷
  st_as_sf(coords = c("x", "y"))
acid_sf ← acid ▷
  st_as_sf(coords = c("x", "y"))

# Repetindo o mesmo gráfico. Repare que agora usamos a função `geom_sf()``:
g ← ggplot() +
  geom_sf(data = homic_sf,
          aes(geometry = geometry, color = "Homicídios"),
          shape = 1) +
  labs(color = "")  
g
```



## Distribuição Espacial dos Homicídios em Porto Alegre (RS) utilizando a geometria espacial.



O código está realizando as seguintes ações:

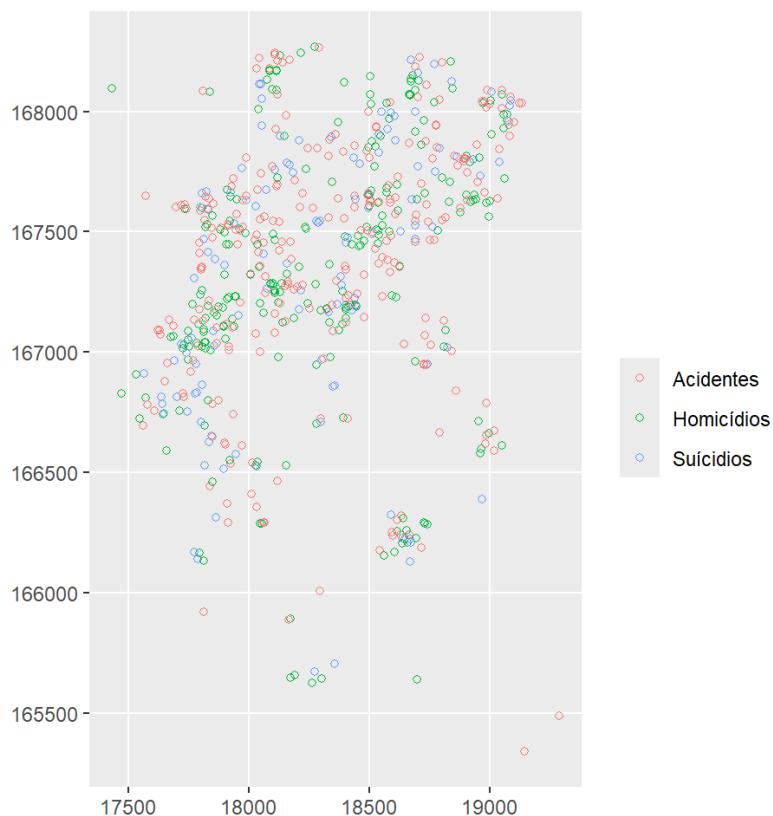
1. Transformação de dados em objetos espaciais: As tabelas de dados `homic`, `suic`, e `acid` são convertidas em objetos de geometria espacial (pontos) usando a função `st_as_sf()` do pacote `sf`. As colunas `x` e `y` são usadas como coordenadas para essa conversão. Isso é feito para `homic_sf`, `suic_sf`, e `acid_sf`.
2. Criação de um gráfico espacial: Um gráfico é criado usando `ggplot()` e a camada `geom_sf()` para plotar os dados espaciais de `homic_sf`. Os pontos são coloridos e rotulados como “Homicídios” no gráfico. O argumento `shape = 1` especifica que os pontos devem ser representados por círculos não preenchidos.

O gráfico gerado mostra os pontos espaciais de `homic_sf` (homicídios) em um mapa ou plano de coordenadas, com a legenda indicando “Homicídios”.

Agora vamos adicionar os pontos referentes a suicídios e acidentes:

```
g ← g +
  geom_sf(data = suic_sf,
           aes(geometry = geometry, color = "Suicídios"),
           shape = 1) +
  geom_sf(data = acid_sf,
           aes(geometry = geometry, color = "Acidentes"),
           shape = 1)
g
```

### Distribuição Espacial dos Acidentes, Homicídios e Suicídios em Porto Alegre (RS) utilizando a geometria espacial.



Este trecho de código adiciona camadas ao gráfico `g`, que já havia sido criado anteriormente, para incluir dados espaciais de suicídios e acidentes:

`suic_sf`: Os dados de suicídios (convertidos em geometria espacial anteriormente) são adicionados ao gráfico como uma nova camada usando `geom_sf()`. Os pontos dessa camada são coloridos e rotulados como “Suicídios”, com a mesma forma (círculos vazados, `shape = 1`).

`acid_sf`: Os dados de acidentes são adicionados da mesma forma, com os pontos coloridos e rotulados como “Acidentes”.

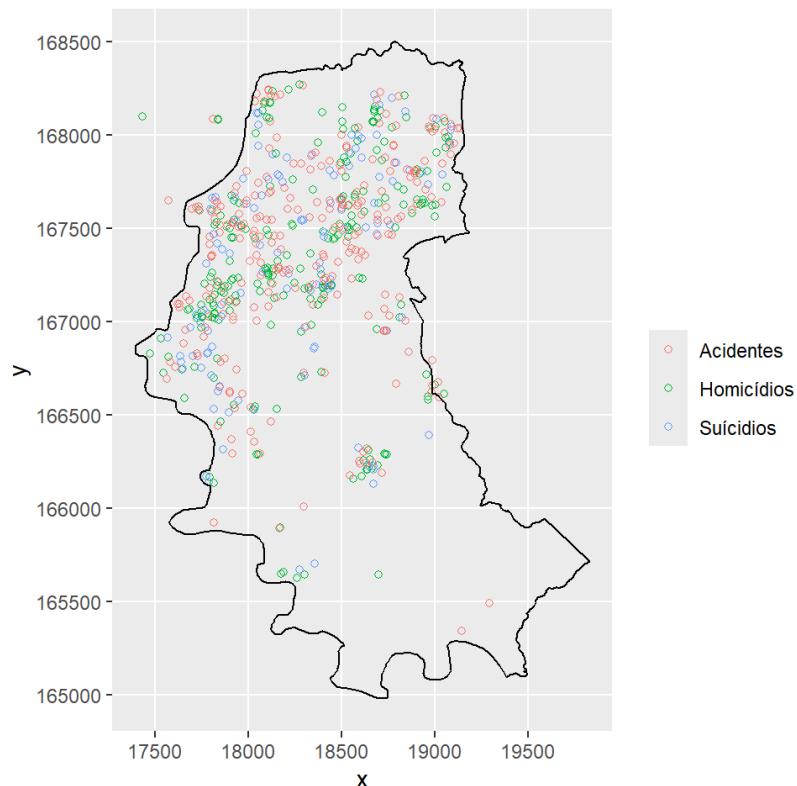
O gráfico final (`g`) agora inclui três conjuntos de pontos espaciais (homicídios, suicídios e acidentes), cada um com uma cor diferente e identificado na legenda.

Agora vamos importar o polígono que corresponde ao contorno de Porto Alegre:

```
# Contorno de Porto Alegre
contorno.poa <- read.table(paste0(local, "/contpoa.dat"),
                           col.names = c("x", "y"))

# Plotando com a função `geom_polygon`:
g +
  geom_polygon(data = contorno.poa, aes(x, y),
                fill = "transparent", color = "black")
```

## Distribuição Espacial dos Acidentes, Homicídios e Suicídios em Porto Alegre (RS) utilizando a geometria espacial com o contorno.



Este código lê um arquivo contendo as coordenadas do contorno de Porto Alegre, armazena os dados em `contorno.poa` e, em seguida, utiliza a função `geom_polygon` para traçar o contorno no gráfico `g`, com preenchimento transparente e borda na cor preta.

Os pontos fora do contorno são das ilhas, não devem ser incorporados a análise. Vamos transformar o contorno também em um objeto espacial (`st_polygon`) para identificar os pontos fora do contorno:

```
# Para transformar em polígono espacial,
# transformamos primeiro em matriz e em seguida
# em uma lista:

poa_sf ← contorno.poa ▷
as.matrix() ▷
list() ▷
st_polygon()
```

Este código transforma o objeto `contorno.poa` em um polígono espacial no formato `sf`.

- `as.matrix()`: Converte os dados em uma matriz de coordenadas.
- `list()`: Envolve a matriz em uma lista (requisito para criar polígonos com múltiplos anéis).
- `st_polygon()`: Constrói o polígono espacial a partir da lista de coordenadas.

O resultado é um objeto `sf` representando o contorno como um polígono espacial.

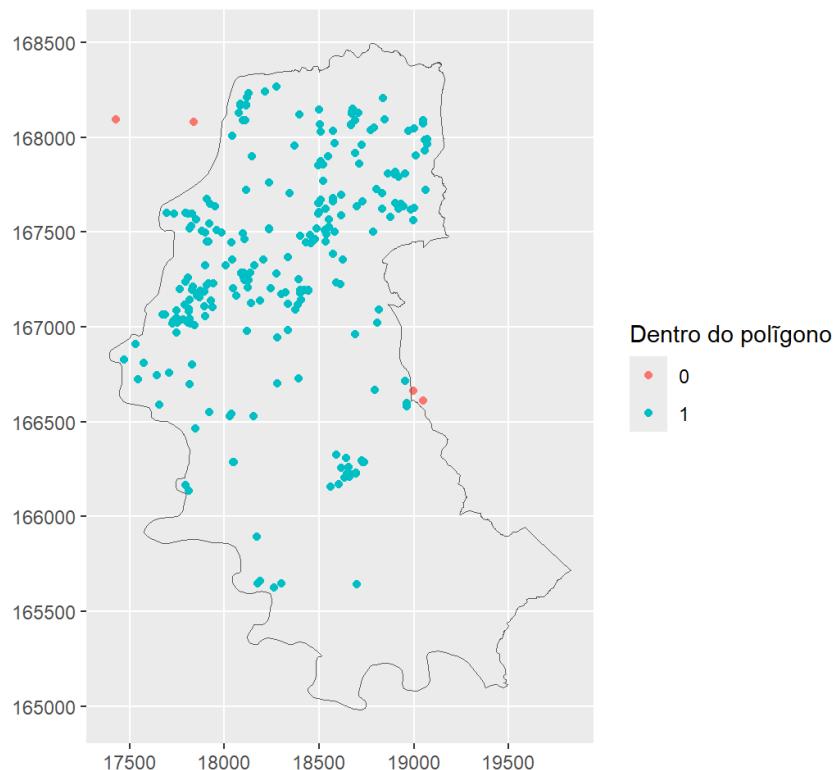
Agora sim, podemos utilizar a função `st_within()` para identificar os pontos fora do contorno:

```
# A função st_within() tem essa funcionalidade
homic_sf ← homic_sf |>
  mutate(dentro = lengths(st_within(homic_sf, poa_sf)))

ggplot(homic_sf, aes(geometry = geometry)) +
  geom_sf(aes(color = as.factor(dentro))) +
  geom_sf(data = poa_sf, fill = "transparent") +
  labs(color = "Dentro do polígono")
```



## Distribuição Espacial dos Homicídios em Porto Alegre (RS) utilizando a geometria espacial com o contorno.



Este código realiza o seguinte:

### 1. Adicionando uma coluna dentro ao dataset `homic_sf`:

- Calcula, para cada ponto em `homic_sf`, se ele está dentro do polígono `poa_sf` usando `st_within()`.
- A função `lengths()` conta quantos polígonos contêm cada ponto (1 se está dentro, 0 se não está).

### 2. Criando um gráfico com `ggplot`:

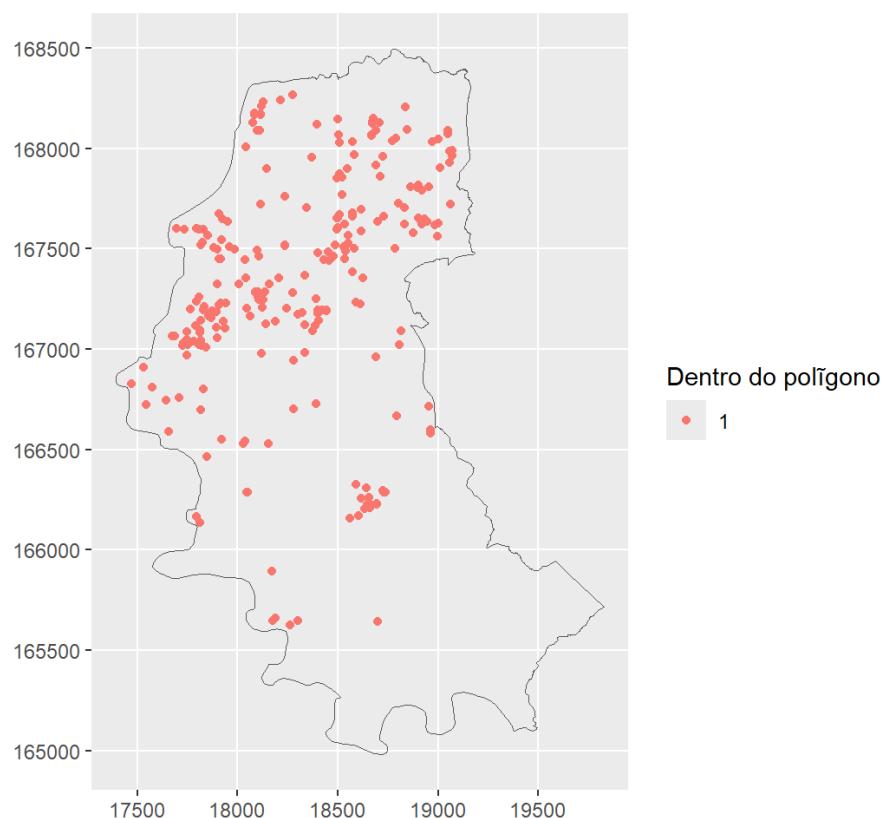
- Plota os pontos de `homic_sf` usando `geom_sf()`, colorindo-os com base na nova coluna dentro (convertida em fator).
- Adiciona o polígono `poa_sf` como uma camada transparente.
- Define a legenda para indicar “Dentro do polígono”.

Isso resulta em um mapa que visualiza os pontos classificados como dentro ou fora do polígono de referência.

```
# Filtra somente as observações dentro do polígono
homic_sf2 <- homic_sf %>
  dplyr::filter(dentro == 1)

ggplot(homic_sf2, aes(geometry = geometry)) +
  geom_sf(aes(color = as.factor(dentro))) +
  geom_sf(data = poa_sf, fill = "transparent") +
  labs(color = "Dentro do polígono")
```

### Distribuição Espacial dos Homicídios e em Porto Alegre (RS) utilizando a geometria espacial com o contorno.



Este comando filtra os pontos espaciais do objeto espacial `homic_sf` (que contém os pontos de homicídios) para incluir apenas as observações que têm o valor da coluna dentro igual a 1.

E aí construímos novamente o mapa utilizando o objeto `homic_sf2` contemplando apenas os pontos dentro do polígono.

Pronto, agora podemos começar a reproduzir algumas análises, que foram vistas na parte teórica a respeito dos padrões pontuais, utilizando o contexto casos de homicídios, suicídios e acidentes de carro em Porto Alegre/RS.

Primeiro, para organizar imagens e apresentá-las juntas, vamos utilizar o pacote `gridExtra`:

```
# se não estiver instalado, rodar:
install.packages("gridExtra")
library(gridExtra)
```

Em seguida, vamos simular alguns padrões dos dados de ponto da ocorrência de homicídios, utilizando como referência o polígono referente ao contorno de Porto Alegre/RS:

```
set.seed(123)

# Gerando um padrão aleatório
alea_sf ← st_sample(poa_sf, size = 100, type = "random")

g_alea ← ggplot(alea_sf, aes(geometry = geometry)) +
  geom_sf() +
  geom_sf(data = poa_sf, fill = "transparent") +
  ggtitle("Distribuição aleatória")

# Gerando um padrão regular
uni_sf ← st_sample(poa_sf, size = 100, type = "regular") %>%
  st_as_sf() %>%
  # a função jitter é bastante utilizada e
  # serve para adicionar uma flutuação (desvio)
  # aleatória aos pontos
  st_jitter(amount = 100) %>%
  dplyr::filter(lengths(st_within(., poa_sf)) = 1)

g_uni ← ggplot(uni_sf, aes(geometry = x)) + geom_sf() +
  geom_sf(data = poa_sf, aes(geometry = geometry),
         fill = "transparent") + ggtitle("Distribuição regular")
```

```

# Gerando um padrão de cluster
# 1. Criar pontos centrais para os clusters
clusters_centers ← st_sample(poa_sf, size = 5, type = "regular") %>%
  st_as_sf()

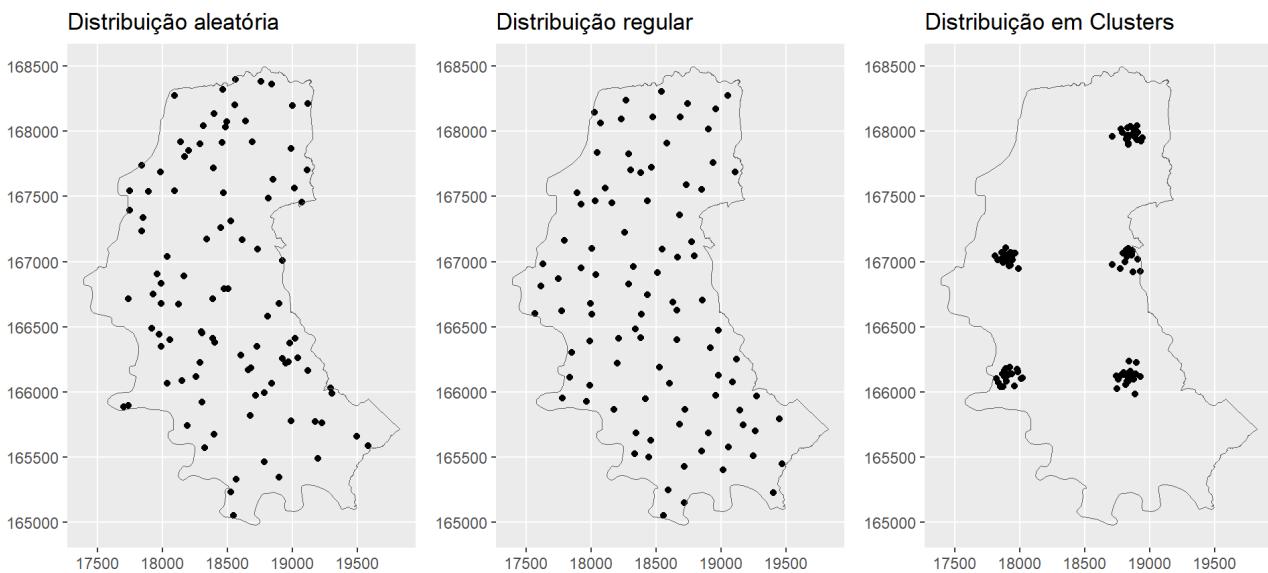
# 2. Adicionar pontos ao redor dos centros dos clusters
cluster_sf ← clusters_centers %>%
  st_coordinates() %>%
  as.data.frame() %>%
  rename(x = X, y = Y) %>%
  mutate(cluster_id = 1:nrow(.)) %>%
  # Adicionar 20 pontos ao redor de cada centro
  group_by(cluster_id) %>%
  do({
    center_x ← .$x
    center_y ← .$y
    n_points ← 20
    tibble(
      x = rnorm(n_points, mean = center_x, sd = 50),
      y = rnorm(n_points, mean = center_y, sd = 50)
    )
  }) %>%
  ungroup() %>%
  st_as_sf(coords = c("x", "y"), crs = st_crs(poa_sf)) %>%
  dplyr::filter(lengths(st_within(., poa_sf)) == 1) # Filtrar apenas pontos dentro do polígono

# Visualizar o padrão de cluster
g_cluster ← ggplot(cluster_sf, aes(geometry = geometry)) +
  geom_sf() +
  geom_sf(data = poa_sf, aes(geometry = geometry),
          fill = "transparent") +
  ggtitle("Distribuição em Clusters")

```

```
grid.arrange(g_alea, g_uni, g_cluster, ncol = 3)
```

## Distribuição espacial dos homicídios em Porto Alegre (RS) nos diferentes padrões de pontos.



Este código simula e visualiza diferentes padrões espaciais de dados de pontos (distribuição aleatória, regular e em clusters) dentro de um polígono representando o contorno de Porto Alegre/RS.

### 1. Padrão Aleatório

- `st_sample(poasf, size = 100, type = "random")`: Gera 100 pontos distribuídos aleatoriamente dentro do polígono `poasf`.
- `ggplot2`: visualiza os pontos sobrepostos ao contorno do polígono. Resultado: Gráfico com pontos aleatoriamente espalhados.

### 2. Padrão Regular

Gera uma grade regular de pontos dentro do polígono e aplica um pequeno desvio aleatório para simular uma regularidade imperfeita.

- `st_sample(poasf, size = 100, type = "regular")`: cria uma grade regular.
- `st_jitter()`: adiciona variação (deslocamento aleatório) aos pontos.
- `filter(lengths(st_within(...)) = 1)`: mantém apenas pontos dentro do polígono.



### 3. Padrão em Clusters

- Simula clusters de pontos ao redor de centros pré-definidos.
- `st_sample(poa_sf, size = 5, type = "regular")`: cria 5 centros regularmente distribuídos. Para cada centro, gera 20 pontos ao redor utilizando valores aleatórios normalmente distribuídos (`rnorm`). Converte coordenadas para o sistema espacial original e filtra pontos dentro do polígono.
- `ggplot2`: exibe os clusters com pontos concentrados ao redor de centros.

O código demonstra a criação de três padrões espaciais distintos (aleatório, regular e clusters) usando funções de manipulação e visualização espacial, permitindo análise visual e comparativa.

## *Testando a Completa Aletoriedade Espacial (CSR)*

```
# Convertendo para a class ppp
alea_ppp ← alea_sf ▷
  as.ppp()

uni_ppp ← uni_sf ▷
  as.ppp()

cluster_ppp ← cluster_sf ▷
  as.ppp()

# Construindo os quadrantes com as respectivas
# contagens
alea_qc ← quadratcount(alea_ppp, nx = 5, ny = 6)
uni_qc ← quadratcount(uni_ppp, nx = 5, ny = 6)
cluster_qc ← quadratcount(cluster_ppp, nx = 5, ny = 6)
```

```
quadrat.test(alea_qc)
```

```
Chi-squared test of CSR using quadrat counts
```

**data:**

X2 = 53, df = 29, p-value = 0.008416 alternative hypothesis: two.sided

Quadrats: 5 by 6 grid of tiles

```
quadrat.test(uni_qc)
```

```
Chi-squared test of CSR using quadrat counts
```



data:

X2 = 34.789, df = 29, p-value = 0.4231 alternative hypothesis: two.sided

Quadrats: 5 by 6 grid of tiles

```
quadrat.test(cluster_qc)
```

```
Chi-squared test of CSR using quadrat counts
```

data:

X2 = 381.39, df = 29, p-value < 2.2e-16 alternative hypothesis: two.sided

Quadrats: 5 by 6 grid of tiles

O objetivo deste código é realizar o teste de completa aleatoriedade espacial (CSR) para três padrões espaciais diferentes (aleatório, regular e em clusters), utilizando quadrantes para contagem de pontos e o teste de  $\chi^2$  (qui-quadrado) com o pacote `spatstat`.

- Os objetos espaciais (`alea_sf`, `uni_sf` e `cluster_sf`) são convertidos para a classe `ppp` (*point pattern dataset in the two-dimensional plane*) do pacote `spatstat`, que é necessária para análises espaciais detalhadas.
- `as.ppp()`: Realiza a conversão de objetos da classe `sf` (Simple Features) para `ppp`.
- O espaço de cada padrão é dividido em uma grade  $5 \times 6$  (30 células ao todo) para contar o número de pontos em cada quadrante.
- `quadratcount()`: `nx = 5`, `ny = 6`: Divide o espaço em 5 células no eixo  $x$  e 6 células no eixo  $y$ . Retorna as contagens de pontos em cada célula.
- O teste de  $\chi^2$  (qui-quadrado) avalia se os pontos estão distribuídos de forma aleatória no espaço.
- `quadrat.test()`: Realiza o teste de CSR para as contagens por quadrante.

Lembrando que:

- **Hipótese nula ( $H_0$ )**: Os pontos estão distribuídos aleatoriamente no espaço.
- **Hipótese alternativa ( $H_1$ )**: Os pontos formam agrupamentos (clusters) ou estão organizados de maneira dispersa no espaço.

Para padrão aleatório dos dados pontuais, o  $p - valor = 0,008416$ , ou seja, o  $p - valor > \alpha$ , supondo um  $\alpha = 0,05$ , indica que há evidências suficientes para rejeitar a hipótese nula de que o padrão de pontos segue uma distribuição aleatória completa (CSR).

Já supondo o padrão regular dos dados pontuais, o  $p - valor = 0,4231$ , ou seja, também o  $p - valor > \alpha$ , supondo um  $\alpha = 0,05$ , indica que não há evidências suficientes para rejeitar a hipótese nula de que o padrão de pontos segue uma distribuição aleatória completa (CSR).

Por fim, para esse cenário aglomerado dos dados pontuais, o  $p - valor = 2,2 \times 10^{-16}$ , ou seja, também o  $p - valor < \alpha$ , supondo um  $\alpha = 0,05$ , indica que há evidências fortes para rejeitar a hipótese nula de que o padrão de pontos segue uma distribuição aleatória completa (CSR).

## *Processo de primeira ordem: Gerando a estimativa de densidade de Kernel (mapa de calor)*

Agora iremos verificar o padrão espacial de primeira ordem dos casos dos casos de homicídios, suicídios e acidentes de carro em Porto Alegre/RS.

```

homic_ppp <- homic_sf2 |>
  as.ppp()
Window(homic_ppp) <- as.owin(poa_sf)

homic_den1 <- density(homic_ppp, sigma = 200, diggle = TRUE)

g_homic_den1 <- ggplot(as_tibble(homic_den1), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(homic_ppp),
             color = "white", shape = 1, size = 0.5) +
  geom_sf(data = poa_sf, aes(geometry = geometry),
          fill = "transparent", inherit.aes = FALSE) +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Homicídios", subtitle = "sigma = 200") +
  guides(fill = "none") +
  theme_void()

suic_ppp <- suic_sf |>
  as.ppp()
Window(suic_ppp) <- as.owin(poa_sf)

suic_den1 <- density(suic_ppp, sigma = 200, diggle = TRUE)
g_suic_den1 <- ggplot(as_tibble(suic_den1), aes(x,y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(suic_ppp),
             color = "white", shape = 1, size = 0.5) +
  geom_sf(data = poa_sf, aes(geometry = geometry),
          fill = "transparent", inherit.aes = FALSE) +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Suicídios", subtitle = "sigma = 200") +
  guides(fill = "none") +
  theme_void()
```

```

acid_ppp <- acid_sf |>
  as.ppp()
Window(acid_ppp) <- as.owin(poa_sf)

acid_den1 <- density(acid_ppp, sigma = 200, diggle = TRUE)

g_acid_den1 <- ggplot(as_tibble(acid_den1), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(acid_ppp),
             color = "white", shape = 1, size = 0.5) +
  geom_sf(data = poa_sf, aes(geometry = geometry),
          fill = "transparent", inherit.aes = FALSE) +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Acidentes", subtitle = "sigma = 200") +
  guides(fill = "none") +
  theme_void()

grid.arrange(g_homic_den1, g_suic_den1, g_acid_den1, ncol = 3)

```

### **Padrão de Densidade Espacial de Homicídios, Suicídios e Acidentes (Sigma = 200) em Porto Alegre/RS.**

Homicídios  
sigma = 200



Suicídios  
sigma = 200



Acidentes  
sigma = 200



Este código realiza a análise de densidade de pontos (*kernel density estimation*) para três conjuntos de dados espaciais (homicídios, suicídios e acidentes) em uma região delimitada, em nosso caso o polígono referente a Porto Alegre/RS, e visualiza os resultados em um painel com três mapas temáticos.

- Os objetos espaciais `homic_sf2`, `suic_sf` e `acid_sf` são convertidos em objetos de pontos (classe `ppp`) para análise espacial com o pacote `spatstat`.
- A região de análise é definida com a função `Window` usando os limites do objeto espacial `poa_sf`.
- A densidade de pontos é calculada para cada conjunto de dados utilizando a função `density` do pacote `spatstat`.
- O parâmetro `sigma = 200` define o raio de suavização, e o argumento `diggle = TRUE` aplica a correção de borda.
- Para cada conjunto de dados (homicídios, suicídios e acidentes). Os resultados da densidade são convertidos para o formato de objeto `tibble` para integração com o `ggplot2`.
- É criado um mapa de calor com `geom_tile`, sobreposto aos pontos originais (com `geom_point`) e aos limites geográficos da área (`geom_sf`).
- A escala de cores é definida com `scale_fill_viridis_c`, e os títulos e temas são ajustados.

Os três mapas temáticos são dispostos lado a lado em uma grade com a função `grid.arrange` do pacote `gridExtra`.

Agora iremos investigar verificar o padrão espacial de segunda ordem dos casos dos casos de homicídios em Porto Alegre/RS utilizando diferentes larguras de banda ( $\sigma = 100, 200$  e  $500$  metros).

```
homic_den2 ← density(homic_ppp, sigma = 100, diggle = TRUE)

g_homic_den2 ← ggplot(as_tibble(homic_den2), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(homic_ppp),
             color = "white", shape = 1, size = 0.5) +
  geom_sf(data = poa_sf, aes(geometry = geometry),
          fill = "transparent", inherit.aes = FALSE) +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Homicídios", subtitle = "sigma = 100") +
  guides(fill = "none") +
  theme_void()

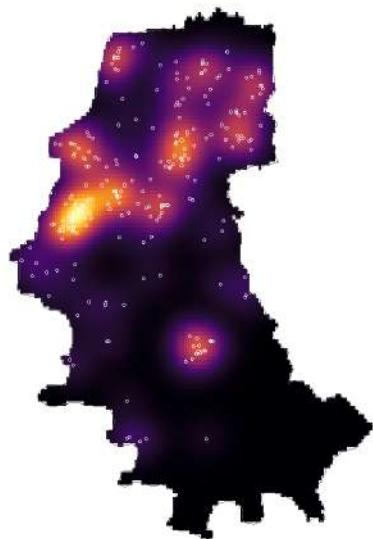
homic_den3 ← density(homic_ppp, sigma = 500, diggle = TRUE)

g_homic_den3 ← ggplot(as_tibble(homic_den3), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_point(data = as_tibble(homic_ppp),
             color = "white", shape = 1, size = 0.5) +
  geom_sf(data = poa_sf, aes(geometry = geometry),
          fill = "transparent", inherit.aes = FALSE) +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Homicídios", subtitle = "sigma = 500") +
  guides(fill = "none") +
  theme_void()

grid.arrange(g_homic_den2, g_homic_den1, g_homic_den3, ncol = 3)
```

## Mapas de Densidade Espacial de Homicídios com Diferentes Valores de Suavização ( $\sigma = 100, 200$ e $500\text{m}$ ) em Porto Alegre/RS.

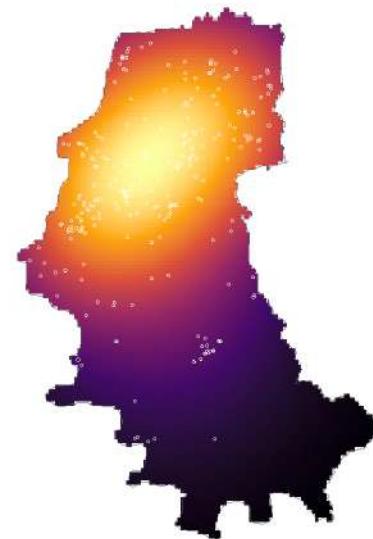
Homicídios  
 $\sigma = 100$



Homicídios  
 $\sigma = 200$



Homicídios  
 $\sigma = 500$



Este código irá usar a função `density` para calcular a densidade kernel espacial dos eventos de homicídio representados pelo objeto `homic_ppp` (um objeto espacial de pontos).

- O parâmetro `sigma` define a largura da suavização (100 no primeiro caso, 200 no segundo e 500 no terceiro).
- Converte os resultados de densidade kernel (`homic_den2` e `homic_den3`) em tibble para serem usados no `ggplot2`.
- Cria mapas de calor (`geom_tile`) das densidades calculadas, com a escala de cores viridis (opção "B").
- Adiciona os pontos originais (`homic_ppp`) para mostrar a localização dos eventos de homicídio.
- Sobrepõe os mapas com limites espaciais de um objeto `poa_sf` (representando um mapa de Porto Alegre), sem preenchimento (`fill = "transparent"`).
- Remove a legenda de preenchimento (`guides(fill = "none")`) e aplica um tema minimalista (`theme_void()`).

Vamos fazer a razão de kernel entre as causas de homicídio e suicídios de Porto alegre/RS:

```

suic_homic_ratio ← suic_den1/homic_den1

g_suic_homic ← ggplot(as_tibble(suic_homic_ratio), aes(x, y)) +
  geom_tile(aes(fill = value)) +
  geom_sf(data = poa_sf, aes(geometry = geometry),
          fill = "transparent", inherit.aes = FALSE) +
  scale_fill_viridis_c(option = "B") +
  labs(title = "Razão", subtitle = "Suicídios/Homicídios") +
  guides(fill = "none") +
  theme_void()

grid.arrange(g_homic_den1, g_suic_den1, g_suic_homic, ncol = 3)

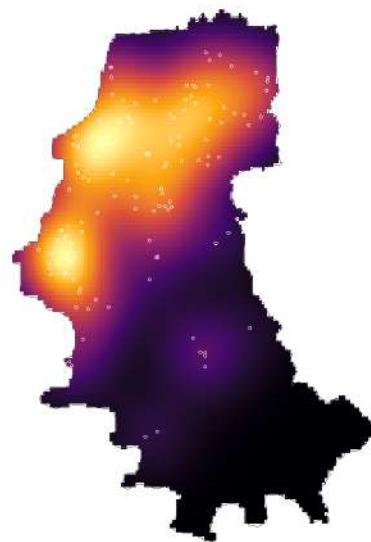
```

### Mapas de Densidade de Homicídios, Suicídios e Razão Suicídios/Homicídios ( $\sigma = 200m$ ) em Porto Alegre/RS.

Homicídios  
 $\sigma = 200$



Suicídios  
 $\sigma = 200$



Razão  
Suicídios/Homicídios



Este trecho de código realiza o cálculo da razão de kernel entre dois tipos distintos de pontos (`suic_den1` e `homic_den1`) armazenando o resultado na variável `suic_homic_ratio`.

Para a criação de um mapa da razão de kernel (`g_suic_homic`) estamos usando o pacote `ggplot2` que mostra a razão de suicídios por homicídios em uma representação espacial.

## *Processo de segunda ordem: Funções G e K*

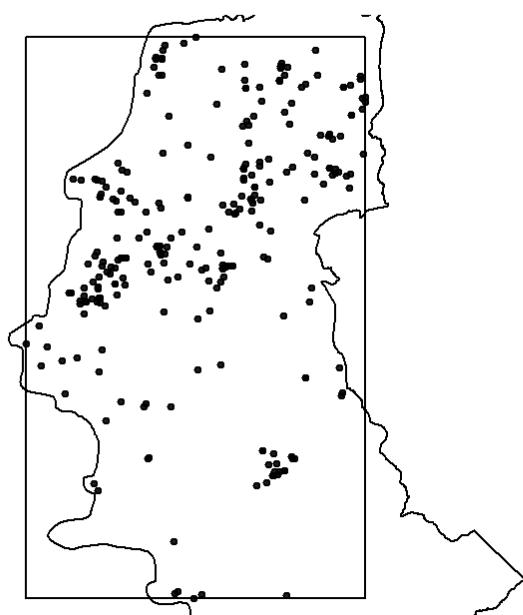
Agora, vamos verificar o padrão espacial de segunda ordem dos casos de homicídios em POA/RS.

```
# se não estiver instalado, rodar:  
install.packages("splancs")  
library(splancs)
```

```
homic_ppp2 ← as.ppp(homic_sf2$geometry)  
  
plot(homic_ppp2, pch = 19, cex = 0.5)  
polymap(contorno.poa, add = T)
```

**Distribuição de Pontos dos Homicídios em Porto Alegre/RS.**

**homic\_ppp2**



Neste trecho do código estaremos convertendo os dados em formato `sf` (dados espaciais) para o formato `ppp` (padrão pontual) e construindo o mapa com os eventos.

- `homic_sf$geometry`: Estamos extraindo apenas a coluna de geometria de um objeto espacial chamado `homic_sf`. Essa geometria representa os pontos (localizações dos homicídios).
- `as.ppp()`: Converte esses dados de geometria para o formato “padrão de pontos” (point pattern) usado na análise espacial. Esse formato é compatível com as funções do pacote `spatstat`, amplamente usado para análise de padrões espaciais em R.
- `plot(homic_ppp2, pch = 19, cex = 0.5)`: Plota o padrão espacial dos pontos (localizações dos homicídios). Com os pontos no formato círculo sólido (`pch = 19`) e tamanho de pontos `cex = 0.5`.
- `polymap(contorno.poa, add = T)`: Desenha o contorno no gráfico do objeto que representa o contorno de Porto Alegre, usado para delimitar visualmente a área de estudo, adicionando o contorno ao gráfico existente (em vez de criar um novo).

```
par(mfrow = c(1, 2))
plot(envelope(Y = homic_ppp2, fun = Gest, nsim = 9), main = "Funcao G")
```

Generating 9 simulations of CSR ... 1, 2, 3, 4, 5, 6, 7, 8, 9.

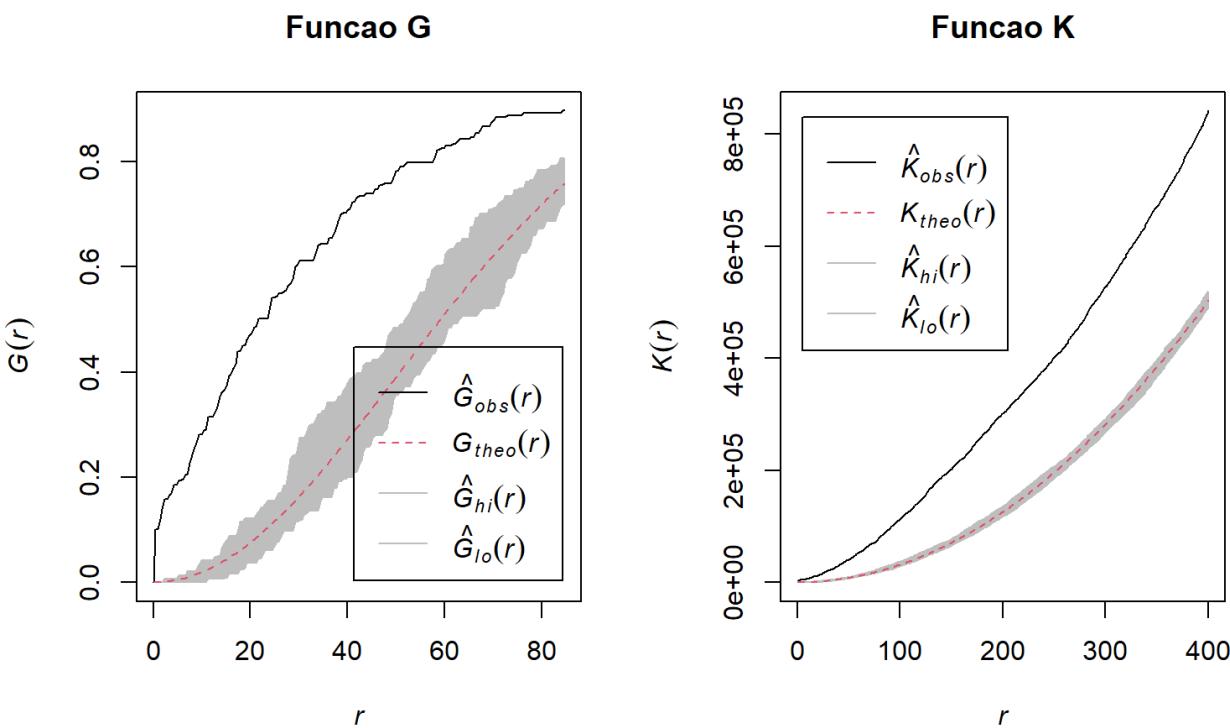
Done.

```
plot(envelope(Y = homic_ppp2, fun = Kest, nsim = 9), main = "Funcao K")
```

Generating 9 simulations of CSR ... 1, 2, 3, 4, 5, 6, 7, 8, 9.

Done.

## Análise do Padrão Espacial: Funções G e K para Homicídios em Porto Alegre/RS.



- `par(mfrow = c(1, 2))`: Ajusta a janela gráfica para exibir dois gráficos lado a lado (1 linha, 2 colunas).
- `plot(envelope(Y = homic_ppp2, fun = Gest, nsim = 9))`: Calcula e plota a função G com simulações de CSR gerando os envelopes simulados para comparar a função observada com um padrão aleatório do padrão de pontos referente aos homicídios.
- `plot(envelope(Y = homic_ppp2, fun = Kest, nsim = 9))`: Calcula e plota a função K com simulações de CSR gerando os envelopes simulados para comparar a função observada com um padrão aleatório do padrão de pontos referente aos homicídios.

Os dois gráficos (funções G e K) sugerem que os homicídios em Porto Alegre (RS) apresentam um padrão agrupado em vez de uma distribuição aleatória. Esse padrão indica que há concentrações de homicídios em determinadas áreas, o que pode estar associado a fatores socioeconômicos, geográficos ou outros. Para uma análise mais aprofundada, seria necessário investigar as causas do agrupamento.

## *Considerações finais*

Neste módulo, você aprendeu a manipular e analisar dados espaciais pontuais utilizando o R. A partir de exemplos práticos, exploramos técnicas essenciais para descrever visualmente e analisar estatisticamente esses eventos espaciais. Vimos como realizar estimativas de densidade espacial, avaliar padrões por meio da técnica de Kernel, e testar hipóteses sobre aleatoriedade ou dependência espacial, ferramentas que nos permitem identificar se os pontos estão agrupados ou dispersos no espaço.

A análise de padrões pontuais mostrou-se uma abordagem poderosa para explorar a distribuição espacial dos eventos, ajudando a vigilância em saúde a obter insights valiosos sobre fenômenos epidemiológicos e situações críticas como surtos de doenças, acidentes ou outros eventos importantes. Ao identificar áreas com concentrações significativas, podemos tomar decisões mais informadas, direcionar recursos e implementar estratégias de intervenção adequadas.

Em resumo, entender como eventos estão distribuídos no espaço, identificando padrões de agrupamento (clusters) ou confirmado sua completa aleatoriedade, é fundamental para compreender melhor os fenômenos em estudo, permitindo ações mais efetivas no contexto da vigilância em saúde.

## Referências

- BIVAND, Roger S. et al. *Applied spatial data analysis with R*. New York: Springer, 2013.
- DIGGLE, P. J. Overview of statistical methods for disease mapping and its relationship to cluster detection. In: ELLIOTT, P.; WAKEFIELD, Jon; BEST, Nicola; BRIGGS, David. *Spatial Epidemiology: Methods and Application*. Oxford: Oxford University Press, 2001. p. 87-103.
- GATRELL, Anthony C. et al. Spatial point pattern analysis and its application in geographical epidemiology. *Transactions of the Institute of British geographers*, p. 256-274, 1996.
- WAKEFIELD, Jon; KELSALL, J. E.; MORRIS, S. E. Clustering, cluster detection, and spatial variation in risk. In: ELLIOTT, P.; WAKEFIELD, Jon; BEST, Nicola; BRIGGS, David. *Spatial Epidemiology: Methods and Application*. Oxford: Oxford University Press, 2001. p. 128-152.



## Módulo 3: Dados de Área

Neste **módulo** vamos mostrar as principais técnicas de análise estatística espacial utilizando dados de área na vigilância em saúde.

### Ao final deste módulo, você será capaz de:

1. Importar e manipular dados de área no software R.
2. Construir mapas atentando-se a suas escalas, pontes de corte e legendas.
3. Entender o conceito de autocorrelação espacial, assim como identificá-la global e localmente dentro de um conjunto de dados espacial.

## Criando mapas

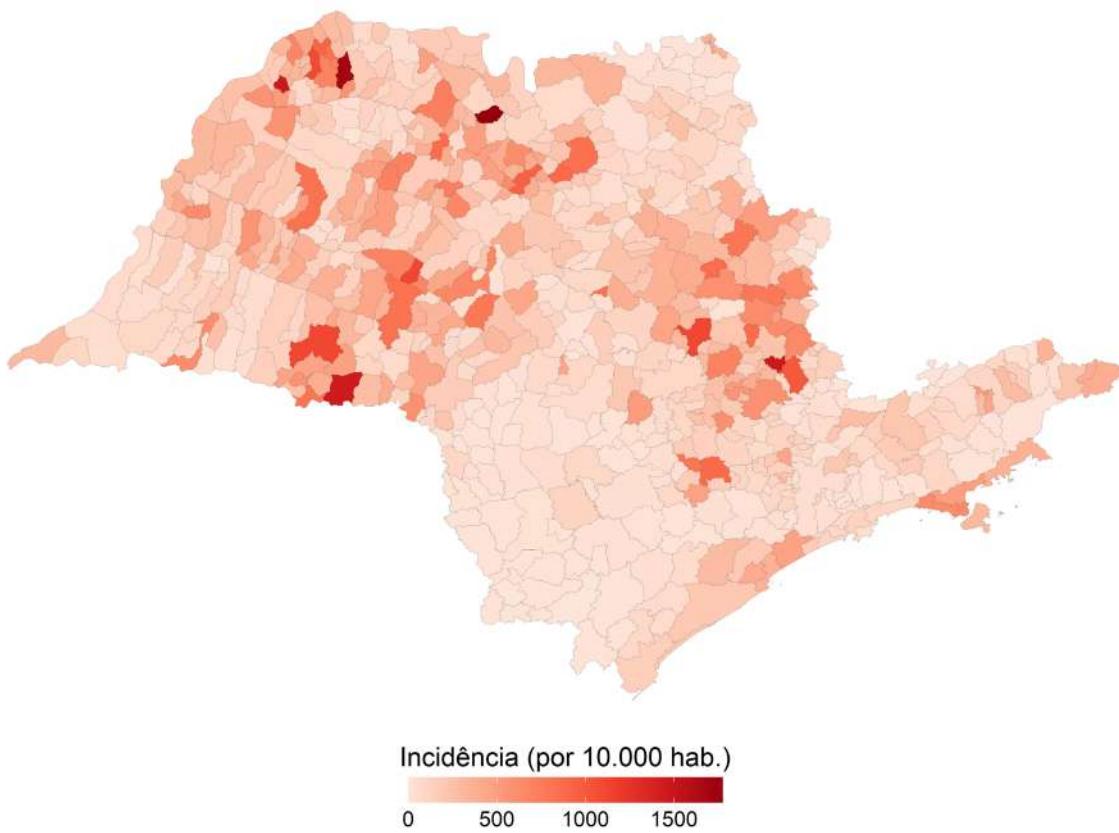
A construção de mapas com dados de área é muito útil na vigilância em saúde pública pois permite visualizar espacialmente a distribuição em áreas de doenças, recursos, fatores de risco, entre outros. Alguns cuidados são necessários para que tais mapas não sejam enviesados, ou seja, que a sua visualização não esteja induzindo a interpretações incorretas. A seguir, vamos mostrar como a definição dos pontos de corte de uma variável contínua (por exemplo, incidência de uma doença) pode levar a conclusões enviesadas, e quais são as boas práticas para evitá-las.

## Pontos de corte em mapas

Ao representar uma variável em mapa (exemplo: número de casos, ou incidência de casos de uma doença), nem sempre representá-la de forma contínua pode ser informativo. Em distribuições desbalanceadas e/ou que possuem valores aberrantes (valores próximos de zero e valores muito mais altos que os demais), a escala de cores para o mapa pode acabar chamando atenção somente para valores extremos.

Vamos tomar como exemplo a incidência de casos prováveis de dengue por município no estado de São Paulo no ano de 2015. A partir do número de casos notificados no SIANAN e da população estimada para o ano em questão, podemos calcular a incidência e representá-la, de forma contínua, em um mapa como na Figura 18.

**Figura 18: Incidência de casos prováveis de dengue por município no estado de São Paulo, 2015.**



No mapa, nota-se alguns municípios que chamam mais atenção pela cor mais escura - ou seja, maior incidência. Alguns outros pontos de concentração da doença são marcados no estado, mas, no geral, a maioria dos municípios é pintada em tom mais claro.

A visualização da variável em sua forma contínua acaba sendo útil em poucos casos, principalmente em situações ligadas à vigilância em saúde. Tradicionalmente, trabalhamos com dados de contagem de casos - e, na maioria dos cenários, a distribuição dessa variável é desbalanceada. Poucos municípios possuem muitos casos enquanto muitos municípios possuem poucos casos. Além disso, municípios com populações pequenas acabam tendo a incidência inflada por conta de um denominador pequeno. Isso faz com que poucos municípios se destaquem dos demais na escala de cores utilizando a variável de forma contínua, e torna difícil a distinção do cenário entre o restante.

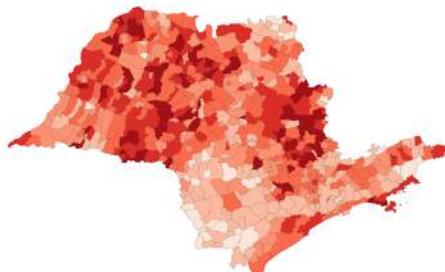
Dessa forma, torna-se importante a separação dessa variável em faixas - através da definição de **pontos de corte**. Contudo, a escolha da estratégia utilizada para definição desses pontos importa e pode impactar e muito no resultado da visualização obtida.

## *Definindo pontos de corte*

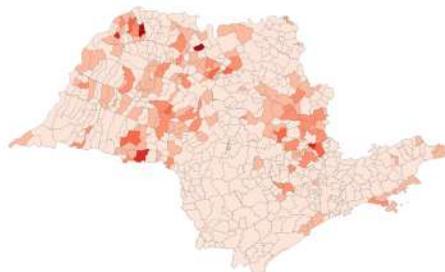
Observe os mapas abaixo na Figura 19. Eles representam a mesma informação - incidência de casos prováveis de dengue por município no estado de São Paulo, em 2015. O que há, portanto, de diferente entre eles?

**Figura 19: Formas de representação da incidência de casos prováveis de dengue no estado de São Paulo, 2015.**

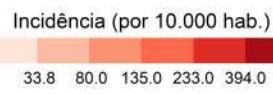
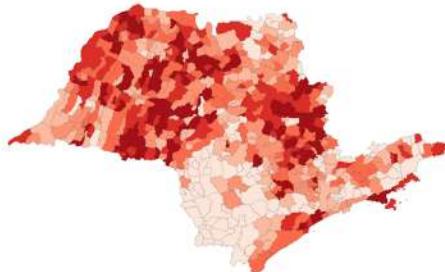
A. Quebras arbitrárias



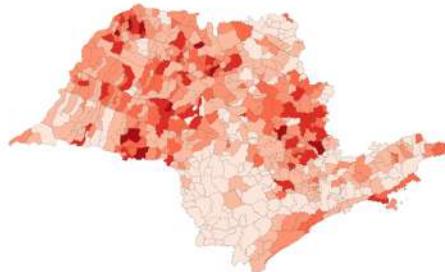
B. Quebras regulares



C. Quebras quantílicas



D. Quebras naturais (Jenks)



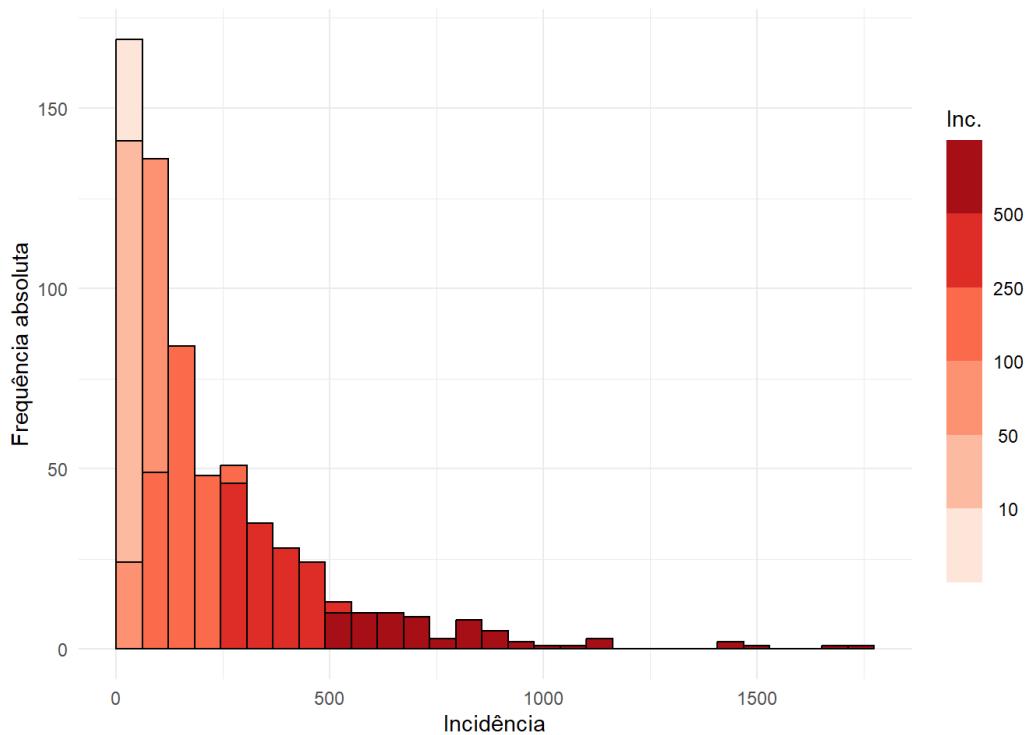
Podemos observar que, a depender da estratégia de definição dos pontos de corte utilizada, a interpretação que temos de cada mapa pode ser diferente. Em alguns mapas (A e C), a situação no estado parece pior, com mais municípios pintados em tons escuros, levando a pensar em uma maior incidência. No mapa B, no entanto, a impressão é exatamente a oposta - a maioria dos municípios está em tons claros, o que nos leva a interpretar que o estado em geral está com baixa incidência da doença. Vamos entender cada uma dessas situações.



## Quebras arbitrárias

O primeiro caso, representado na Figura 19A, não utiliza nenhum método específico para estabelecimento de pontos de corte da incidência. Os pontos são definidos arbitrariamente, e podem ou não ser baseados em uma referência existente sobre o tema. Em um departamento de vigilância, pode-se ter estabelecido que uma incidência de 50 casos por 10.000 habitantes é um patamar controlado, e acima de 100 casos por 100.000 tem-se um cenário de preocupação. Assim, o analista pode-se basear nesses pontos pré-definidos na rotina de vigilância para gerar seus mapas. Contudo, não é garantia de que a visualização seja adequada aos dados: pode ser que muitos municípios se concentrem em uma mesma faixa, por exemplo, trazendo dificuldades na distinção entre esses municípios (Figura 20).

**Figura 20: Distribuição dos municípios em cada faixa de incidência, definida por quebras arbitrárias.**





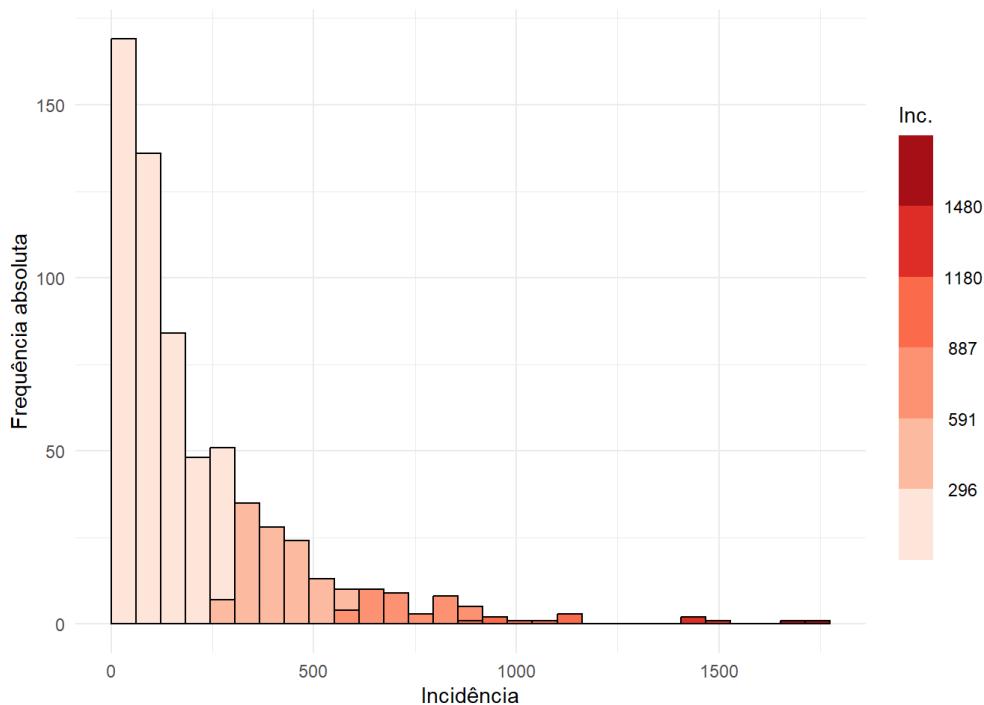
## Quebras regulares

A segunda estratégia de quebras utilizada é a das **quebras regulares**. Nela, considera-se o intervalo entre o **mínimo** e o **máximo** da distribuição, e divide-se esse intervalo em partes **iguais**. Dessa forma, cada faixa das quebras possui a mesma amplitude.

Repare, na Figura 19B, que a distância entre um ponto de corte e outro é de aproximadamente 296 para todos os intervalos.

Essa estratégia facilita a comunicação mas é particularmente útil quando a distribuição dos dados se aproxima de uma distribuição uniforme. No cenário da vigilância em saúde, esse é raramente o caso - há muitos valores concentrados em uma faixa da distribuição e poucos valores distribuídos entre outras. Isso causa o efeito de que um número excessivo de municípios pode cair na mesma categoria, enquanto outras categorias podem possuir um ou dois municípios (Figura 21). Pode ser, até, que faixas de incidência não possuam nenhum município dentro dela.

**Figura 21: Distribuição dos municípios em cada faixa de incidência, definida por quebras regulares.**

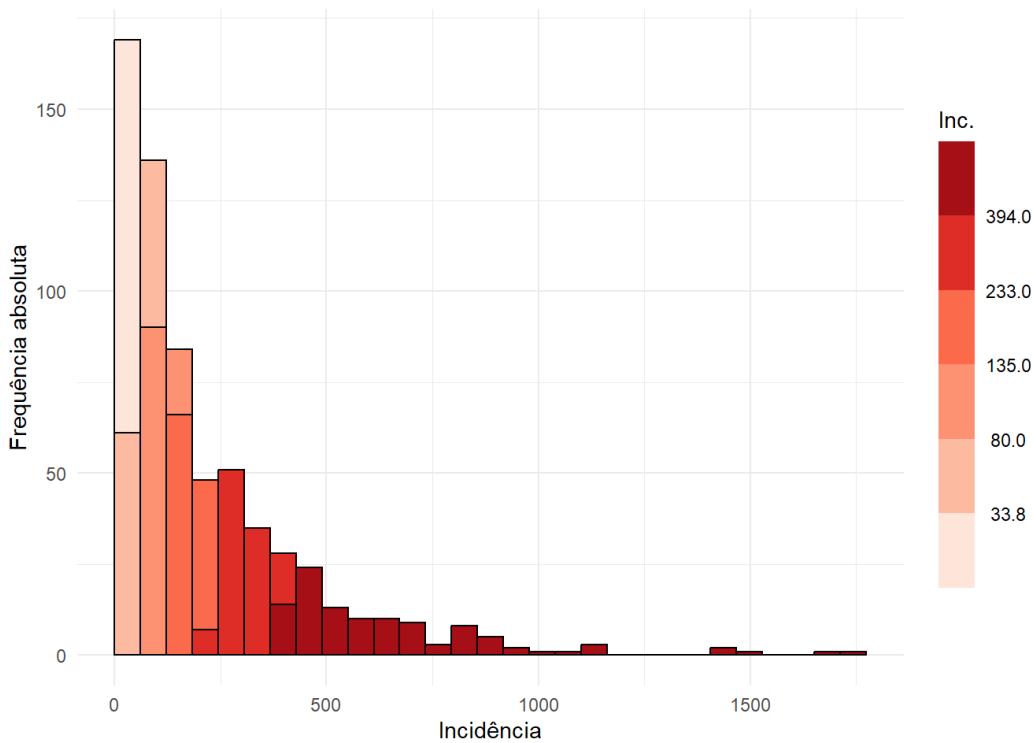


Vê-se agora que a distância entre a mudança de cores é sempre a mesma, o que é agradável visualmente. Contudo, a vasta maioria dos municípios se concentra dentro de uma mesma cor. Dessa forma, a estratégia falha em distinguir esses municípios com incidência mais baixa entre eles.

## *Quebras quantílicas*

Outra estratégia consiste em determinar pontos de quebra a partir de quantis da distribuição. As quebras quantílicas tentam dividir os municípios em grupos de forma com que cada grupo contenha aproximadamente a mesma quantidade de municípios. Por exemplo, ao dividir em 5 grupos, teríamos cada grupo com aproximadamente 20% do total de observações.

**Figura 22: Distribuição dos municípios em cada faixa de incidência, definida por quebras quantílicas**

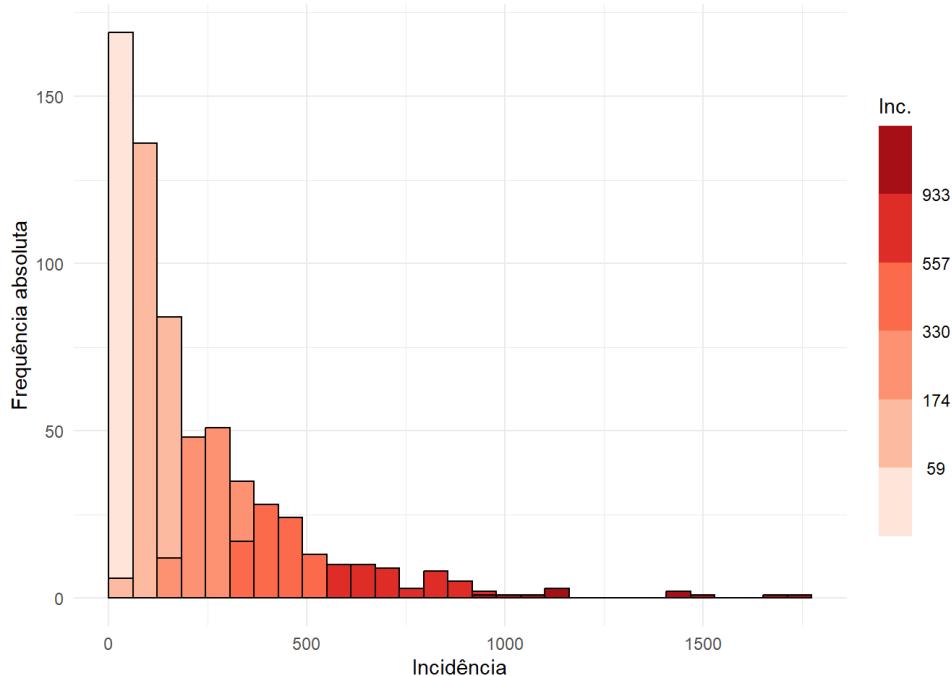


Dessa forma, “forçamos” os grupos a conterem aproximadamente a mesma quantidade de municípios. Essa técnica garante que os mapas terão representação visível de todos os grupos; contudo, vemos que mesmo municípios com incidências baixas (Figura 22, primeira barra) já estão sendo categorizados para o segundo grupo, enquanto o ponto de corte para o grupo de incidência mais alta fica muito mais baixo. Isso pode levar a falsas impressões nos mapas, já que fazemos com que necessariamente uma parcela igual esteja contida nos níveis mais altos e mais baixos de incidência.

## *Quebras naturais, ou quebras de Jenks*

O método de classificação por quebras naturais (ou quebras de Jenks) parte da ideia de obter pontos de corte que tentam, ao máximo, **minimizar** a variação **intra-grupo** e **maximizar** a variância **entre os grupos**. Ou seja, busca-se dividir o conjunto de dados em grupos semelhantes entre si e diferentes dos demais.

**Figura 23: Distribuição dos municípios em cada faixa de incidência, definida por quebras naturais de Jenks**



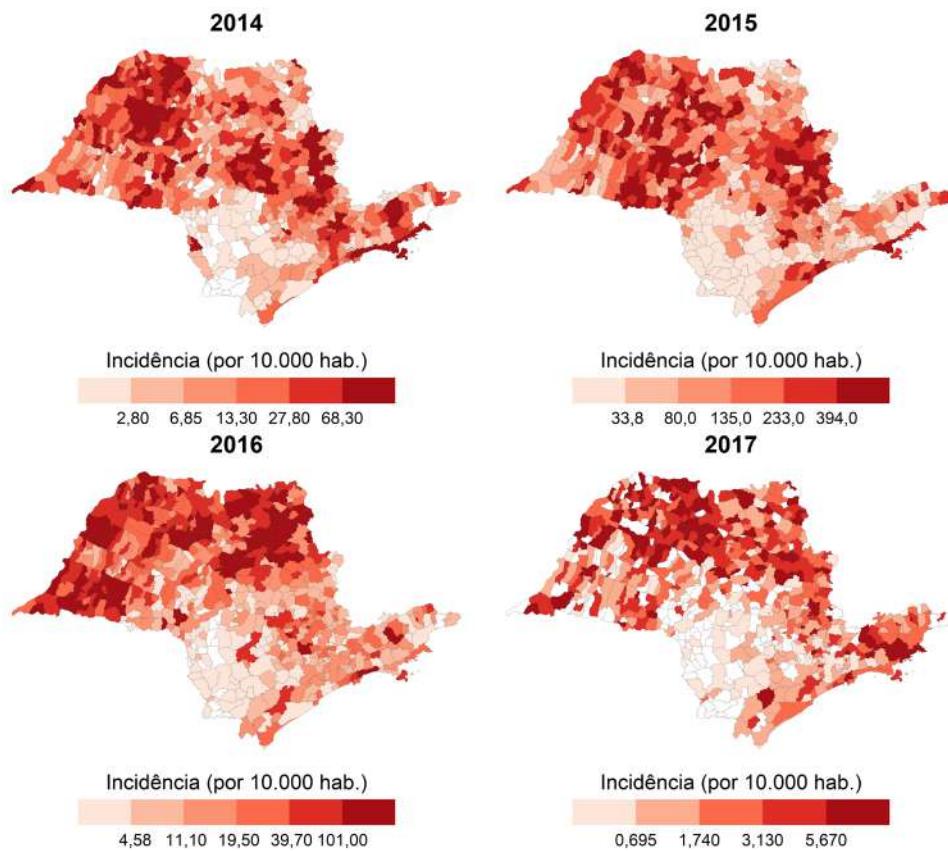
Para este exemplo, as quebras naturais classificaram os municípios de forma equilibrada em relação às demais (Figura 23) - não houve uma hiper concentração dos municípios em uma só classe (como as quebras regulares), nem a definição de quebras que abrangiam muitos municípios diferentes dentro de si (como nas quebras quantílicas). Percebe-se que o primeiro grupo consta majoritariamente de municípios com baixa incidência (primeira barra no histograma), e o grupo mais alto inclui poucos municípios, mas que realmente apresentaram valores de incidência mais extremos na distribuição.

## *Comparando mapas no tempo*

Outro ponto importante ao definir escalas e pontos de corte em mapas é a padronização dessas escalas ao comparar mapas em diferentes pontos no tempo.

Por exemplo, suponha que queremos comparar a incidência de dengue nos municípios do estado de São Paulo no período 2014-2017, utilizando as quebras quantílicas (Figura 24).

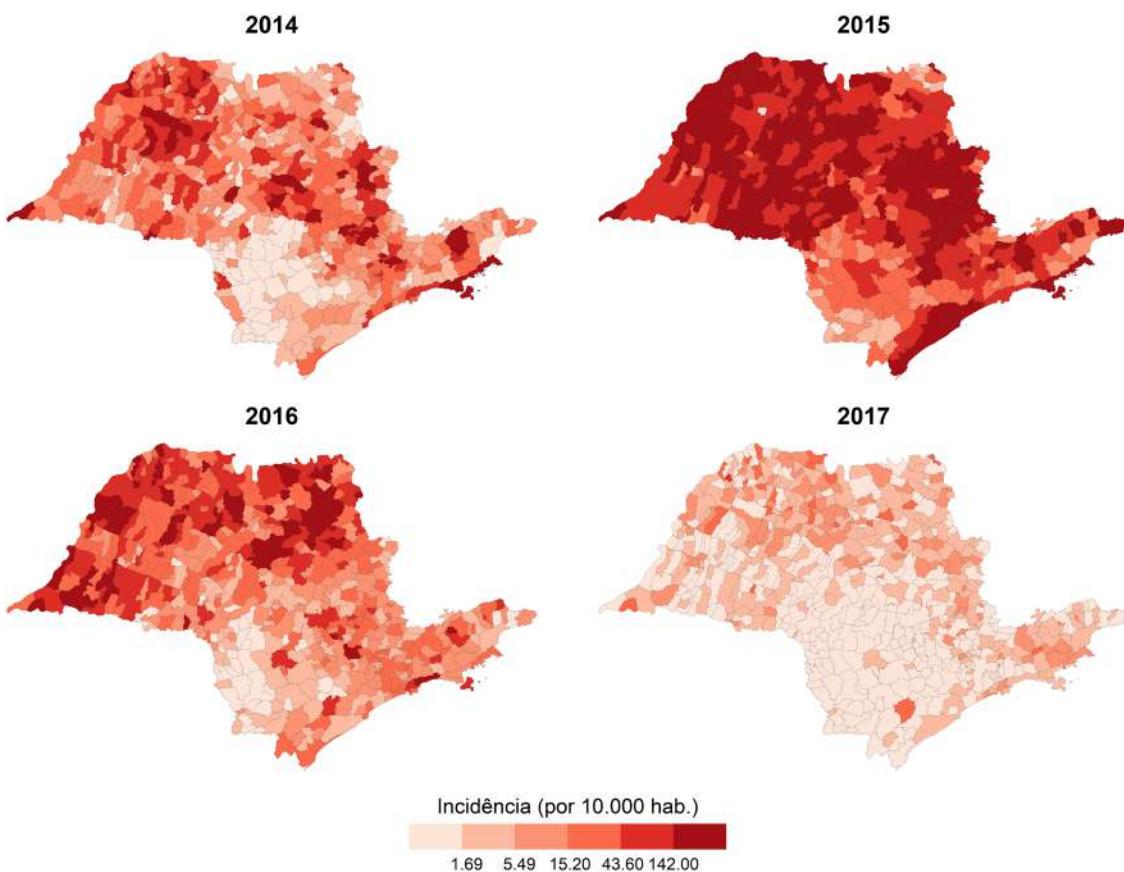
**Figura 24: Incidência de dengue nos municípios São Paulo, ao longo de 2014-2017**



Os mapas gerados nos levam a interpretações sobre as áreas onde a doença se concentrou em cada ano. Podemos dizer que há certa similaridade entre os padrões espaciais, onde certas regiões do Norte e Noroeste paulista aparecendo repetidamente nas faixas de maior incidência. A impressão geral que temos é que, a distribuição da doença se deu de forma parecida ao longo dos anos. Contudo, vemos que os pontos de corte em cada mapa são criticamente diferentes: enquanto o tom mais escuro no mapa relativo à 2017 representa uma incidência maior que 5,67, o mesmo tom no mapa de 2015 representa incidência superior a 393,97.

Assim, ao comparar mapas no tempo, faz-se necessária a utilização de uma escala única para permitir a comparação das taxas não somente no espaço, mas entre os anos. Veja:

**Figura 25: Incidência de dengue nos municípios São Paulo, ao longo de 2014-2017, utilizando escalas de cor independentes a cada ano.**



Agora utilizamos a mesma técnica de classificação a partir de quebras quantílicas, mas para todo o período de estudo para determinação das quebras. Isso permite a comparação dos cenários ao longo do tempo: além de identificarmos certos padrões espaciais, vemos que o ano de 2015 foi muito mais crítico em termos de incidência de Dengue do que os demais anos - principalmente 2017, quando a maioria dos municípios ficou no primeiro grupo da escala de cores.

## *Problemas conhecidos ao lidar com dados de área*

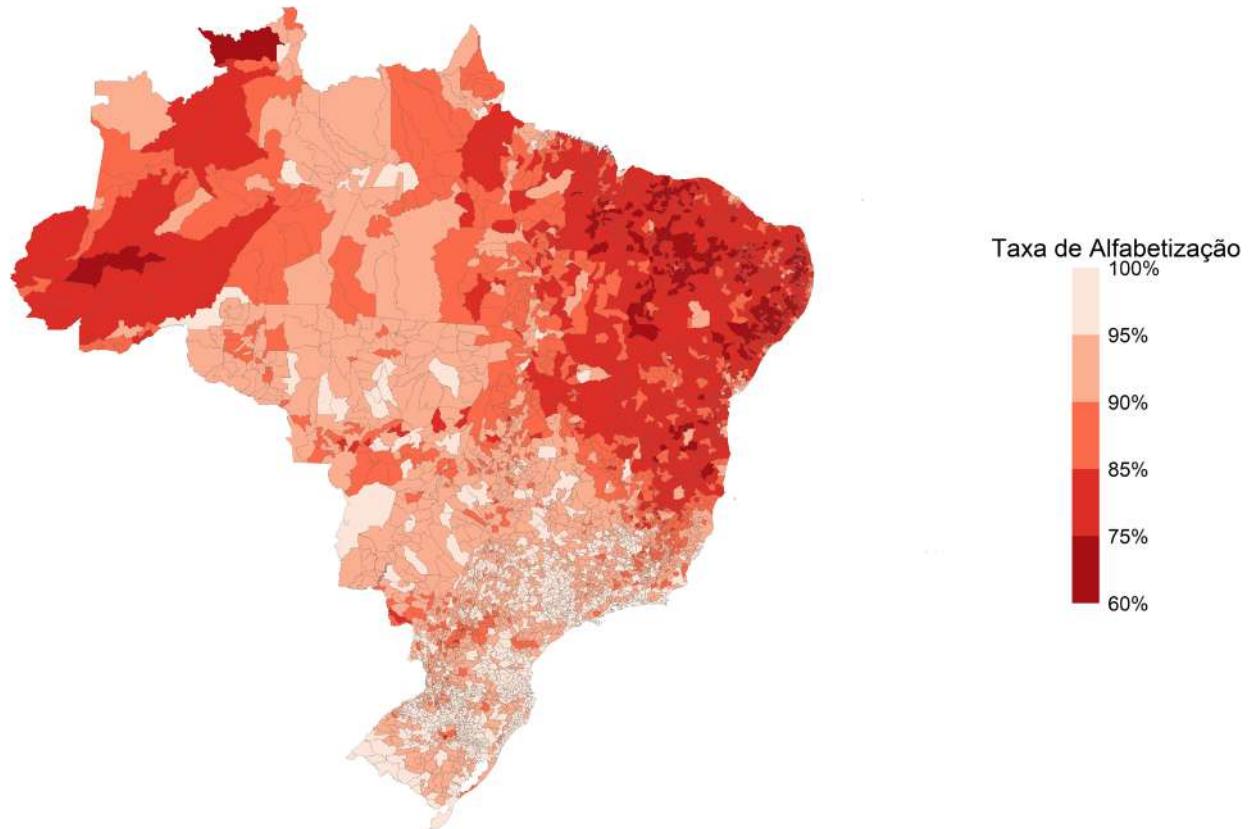
Ao trabalhar com dados agregados a nível de área, estamos sujeitos à dependência das definições e delimitações impostas na divisão do território, sobre as quais muitas vezes não temos controle. Aqui tratamos de alguns efeitos conhecidos que podem acontecer ao lidar com esse tipo de dado, e que devem sempre ser considerados ao interpretar os resultados de análises com dados de área.

### *Tamanho heterogêneo das áreas*

Como mencionado, as delimitações no território frequentemente são derivadas de divisões políticas e administrativas. Entre outros efeitos, isso pode levar à geração de áreas pequenas em extensão territorial, porém com alta densidade populacional; enquanto áreas extensas concentram pouca população. Isso pode afetar nossas visualizações: as áreas maiores, além de chamarem maior atenção aos nossos olhos, estão sujeitas a uma inflação das taxas devido ao baixo denominador (população). Vamos conferir, por exemplo, a análise da taxa de alfabetização dos municípios brasileiros, segundo o Censo 2022 (Figura 26).

As áreas de principal destaque no mapa são áreas consideravelmente grandes, e que apresentam piores indicadores.

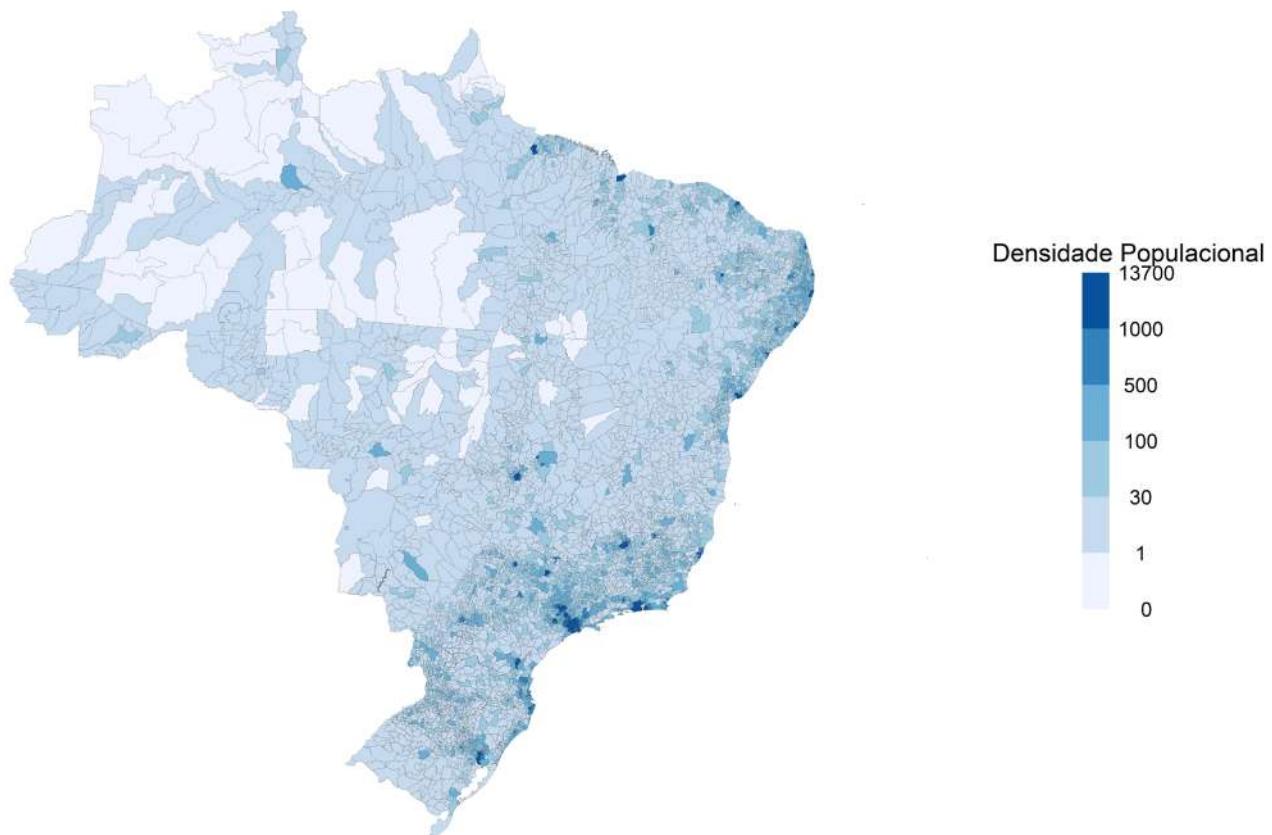
**Figura 26: Taxa de Alfabetização (%) nos municípios brasileiros, segundo o Censo 2022.**



Vamos comparar esse padrão espacial com a densidade populacional nesse municípios (Figura 27).



**Figura 27: Densidade populacional (habitantes/km<sup>2</sup>)  
nos municípios brasileiros, segundo o Censo 2022.**



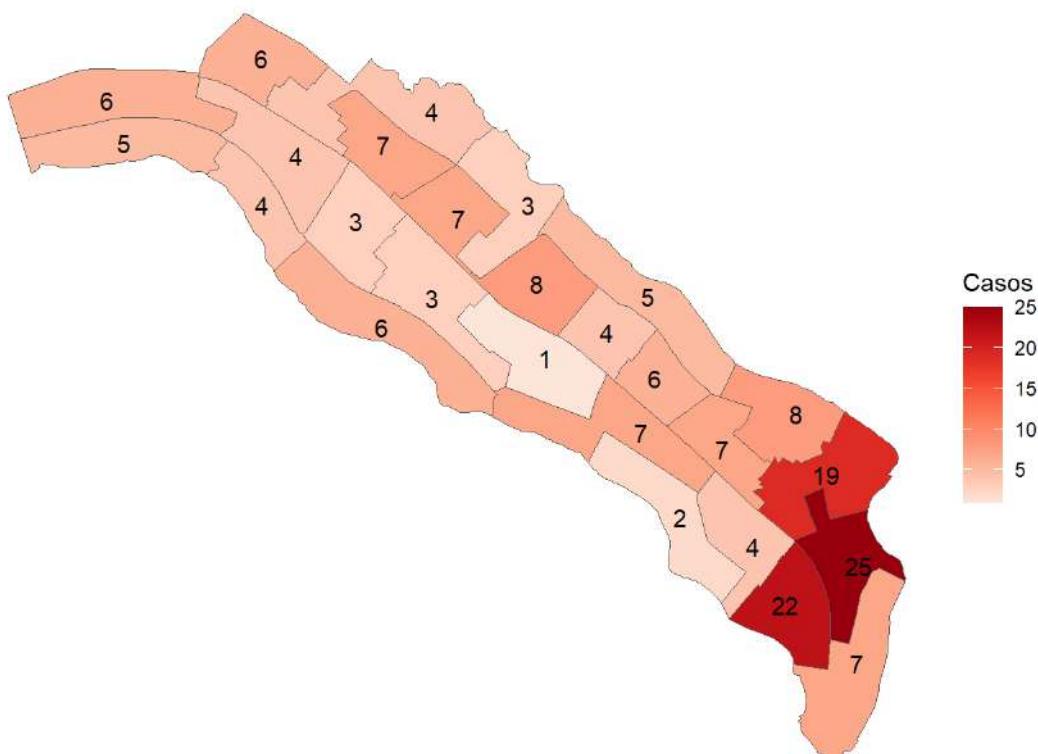
Vemos que além de grandes, as áreas de destaque no primeiro mapa possuem baixa densidade populacional. Como a população é frequentemente o denominador nos indicadores que analisamos, quando a população é pequena essas taxas tendem a ser instáveis. Portanto, municípios de população pequena (e que, como vimos, podem ser grandes em termos de extensão territorial) estão sujeitos a uma elevação artificial nas taxas devido a essa baixa população, e acabam mascarando demais padrões espaciais quando olhamos o mapa completo.

## *Problema da área modificável (MAUP)*

Outro problema comum é que as taxas e indicadores dependem da forma com que as áreas são delimitadas e agregadas. E se, de repente, essas delimitações são alteradas, resultados diferentes podem ser obtidos. Esse problema é conhecido como Problema da Área Modificável (*Modifiable Areal Unit Problem - MAUP*).

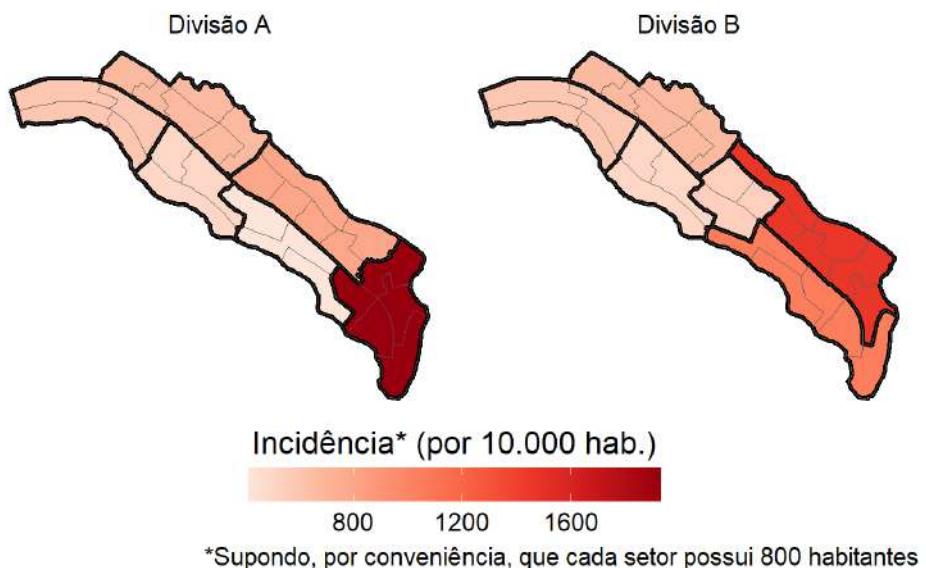
Vamos tentar ilustrar esse efeito a partir de dados hipotéticos de casos de uma doença sobre um conjunto de setores censitários (Figura 28).

**Figura 28: Número de casos de uma doença hipotética  
sobre um conjunto de setores censitários.**



Agora, vamos supor que esses dados são agregados para uma unidade espacial maior (como bairro) para serem disponibilizados. A forma com que essas delimitações de bairro são definidas a partir dos setores censitários pode mudar a visualização da incidência da doença que obtemos. Vamos comparar duas delimitações distintas na Figura 29:

**Figura 29: Comparação da incidência da doença hipotética em duas divisões de território diferentes (A e B) a partir dos setores censitários.**



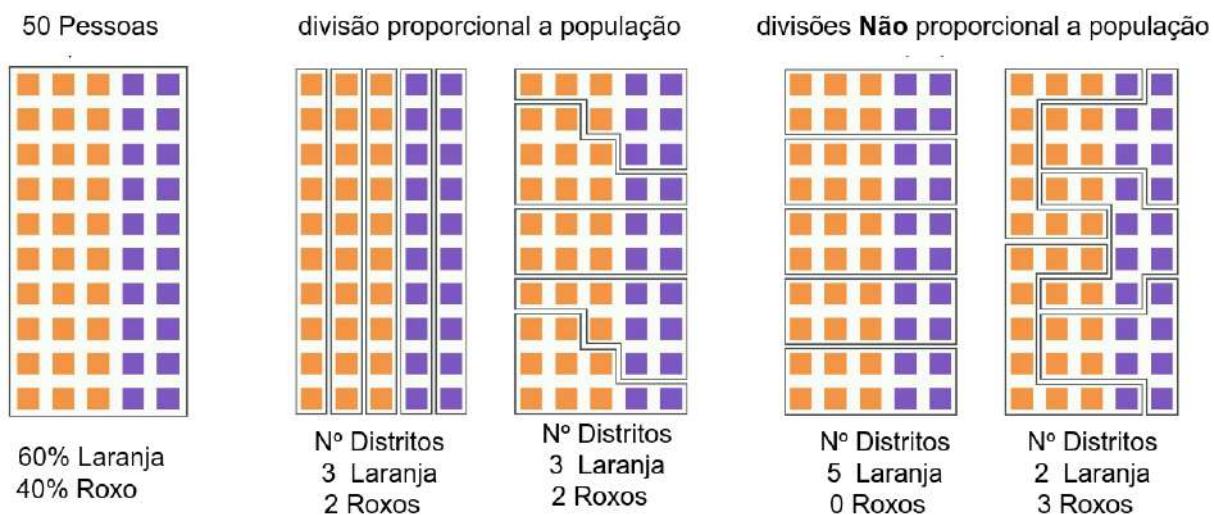
Vemos que as duas divisões, embora passem a mesma informação de distribuição da doença, mostram impressões ligeiramente diferentes de onde ela se concentra. Além disso, taxas de incidência consideravelmente mais altas são obtidas a depender do zoneamento. Como o processo de agregação pode se dar em vários níveis a partir dos dados referenciados, deve-se ter atenção para esses possíveis efeitos de atenuação ou de amplificação das taxas de incidência a depender dos níveis e divisões no processo de agregação.

## Gerrymandering

A variação de proporções e taxas de acordo com a mudança dos limites territoriais pode ser usado intencionalmente e com a finalidade de alterar resultados importantes - como é feito no **contexto eleitoral estadunidense**. Em tal país, parte do processo eleitoral envolve contar qual partido é ganhador em cada distrito (voto distrital) e a partir disso, contar quantos distritos cada partido “tem”. A forma com que os distritos são delimitados é extremamente importante, pois pode alterar os resultados obtidos no final das contas.

Observe o exemplo a seguir na Figura 30, onde temos 50 pessoas e 60% delas votaram no partido laranja, e os 40% restantes no partido roxo.

**Figura 30: Exemplo de Gerrymandering - A depender da divisão dos distritos, o resultado final da contagem é alterado.**



Perceba que, dependendo da forma com que são traçados os distritos, temos uma mudança significativa na contagem de distritos que cada partido possui: no terceiro cenário, temos todos os distritos com vitória do partido laranja. Por outro lado, mesmo com menor proporção total de votos no total da população, a quarta configuração faz com que o partido roxo tenha mais distritos - 3 contra 2 do laranja. Esse procedimento de calcular as divisões distritais com a finalidade de alterar os resultados obtidos no processo eleitoral ocorre nos Estados Unidos, e é chamado de Gerrymandering.

## *Prática em R: Criação de mapas temáticos*

Vamos realizar nossas primeiras visualizações, utilizando os dados de casos de dengue no município de São Paulo em 2015. Para manipular dados espaciais, vamos utilizar a biblioteca `sf`. É importante se atentar aos pré-requisitos de instalação da biblioteca e suas dependências. Mais detalhes sobre sua instalação podem ser encontrados [aqui](#).

```
library(tidyverse)
library(sf)
```

- `library(tidyverse)`: carrega o conjunto de pacotes do `R` chamado `tidyverse`, que inclui ferramentas para manipular dados, fazer gráficos e análises estatísticas de forma prática (ex.: `dplyr`, `ggplot2`, `readr` etc.).
- `library(sf)`: carrega o pacote `sf`, que é usado para trabalhar com dados espaciais (mapas, coordenadas, polígonos) de forma integrada com o R.

Vamos agora ler os dados de casos de dengue em São Paulo, obtidos através do TAB-NET do DATASUS ([link](#)):

```
dengue_sp <- read.csv("https://raw.githubusercontent.com/joaohmoraes/dados-mod-temp/refs/heads/main/csv/dengue_SP.csv")
head(dengue_sp)
```

#>	cod_ibge	cod_mun6	nome_mun	ano	pop	casos
#> 1	3500105	350010	Adamantina	2014	34185	39
#> 2	3500204	350020	Adolfo	2014	3829	3
#> 3	3500303	350030	Aguai	2014	32206	529
#> 4	3500402	350040	Águas da Prata	2014	7565	5
#> 5	3500501	350050	Águas de Lindóia	2014	17623	11
#> 6	3500550	350055	Águas de Santa Bárbara	2014	6171	7

- O código lê um arquivo `.csv` do repositório *github* (um conjunto de dados sobre dengue em São Paulo) e guarda esses dados em um objeto chamado `dengue_sp`.
- A função `head(dengue_sp)`, mostra as primeiras linhas da tabela para você visualizar um resumo dos dados carregados.

Temos os dados referente aos municípios, população estimada, o ano de ocorrência, e o número de casos prováveis. A partir dessas informações, podemos calcular a incidência por 10.000 habitantes:

```
dengue_sp ← dengue_sp %>%
  mutate(inc = (casos/pop)*10000)
```

Esse código cria uma nova coluna chamada `inc` na tabela `dengue_sp`, que representa a incidência de dengue como o número de casos (`casos`) dividido pela população (`pop`), multiplicado por 10.000.

Em seguida, vamos filtrar somente o ano de 2015 e verificar um resumo da variável incidência:

```
dengue_2015 ← dengue_sp %>%
  filter(ano == 2015)
summary(dengue_2015$inc)
```

```
#>      Min. 1st Qu. Median     Mean 3rd Qu.      Max.
#>      0.00   58.45 135.09  218.71 297.54 1773.28
```

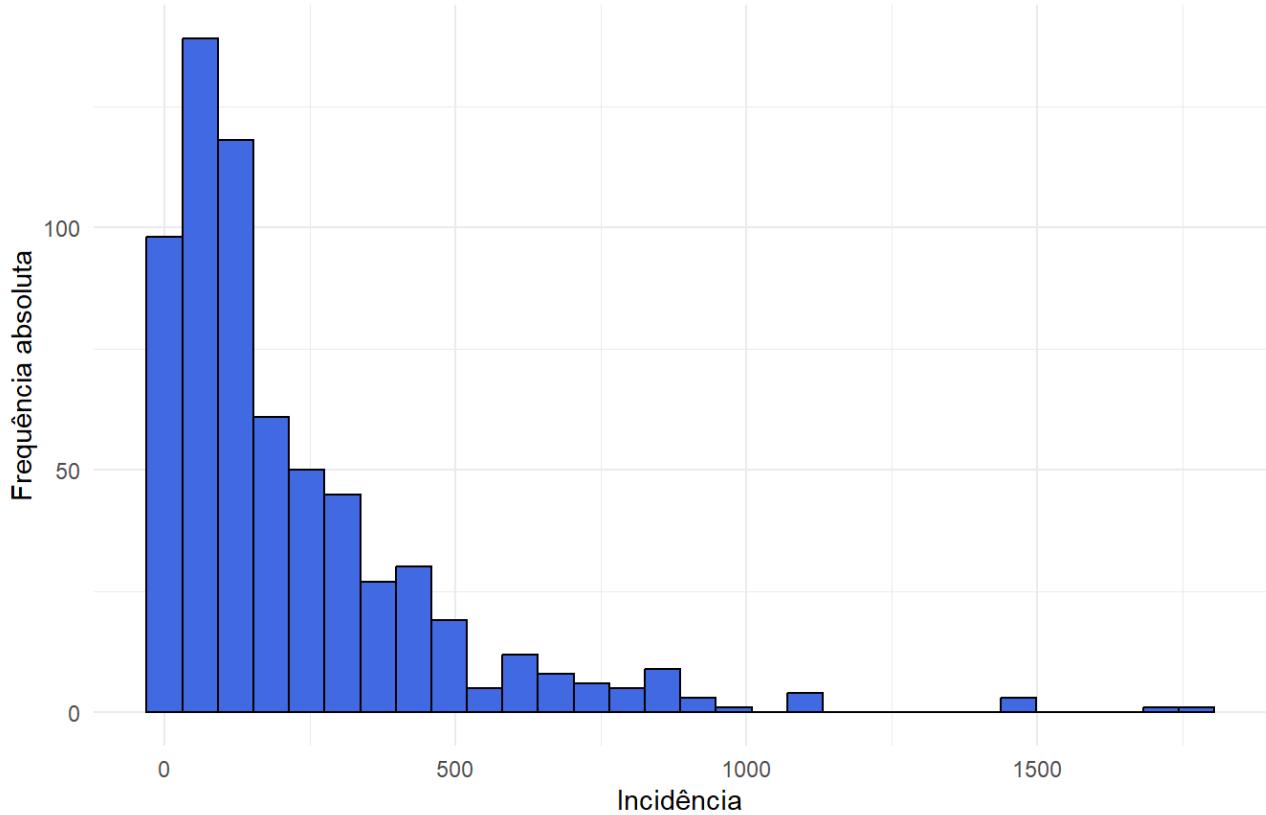
Esse código faz duas coisas:

- Ele filtra o conjunto de dados `dengue_sp` para manter apenas os registros notificados no ano de 2015, criando um novo objeto chamado `dengue_2015`.
- E por fim mostra o resumo estatístico, ou seja, mostra estatísticas básicas (mínimo, máximo, média, mediana, etc.) da variável `inc` (incidência de dengue) do objeto `dengue_2015`.

Podemos visualizar sua distribuição em um histograma simples:

```
ggplot(dengue_2015, aes(x=inc)) +  
  geom_histogram(fill = "royalblue", color = "black") +  
  labs(  
    x = "Incidência",  
    y = "Frequência absoluta",  
    title = "Distribuição da Incidência de Dengue por 10.000 \nhabitantes nos  
municípios de São Paulo, 2015"  
  ) +  
  theme_minimal()
```

Distribuição da Incidência de Dengue por 10.000 habitantes nos municípios de São Paulo, 2015



Esse código usa a função `ggplot2` para criar um histograma que mostra como a incidência de dengue (variável `inc`) está distribuída nos municípios de São Paulo em 2015.

- As barras do histograma são definidas como azuis (`fill = "royalblue"`) com contorno preto (`color = "black"`).
- Os eixos são rotulados como `Incidência` no eixo dos x e `Frequência absoluta` no eixo y.
- O tema visual do gráfico é o minimalista (`theme_minimal()`), deixando o fundo mais limpo.

Vemos que a distribuição é assimétrica - o que é esperado para uma variável originada por uma contagem - com uma maior concentração de valores mais baixos de incidência mas com alguns poucos municípios registrando incidências maiores.

## *Malhas geográficas com geobr*

Agora vamos visualizar nossos dados no espaço!

Tradicionalmente, o carregamento de arquivos espaciais para dentro do ambiente R é feito através da leitura de *shapefiles* (arquivos que incorporam a dimensão espacial aos dados), e usualmente através da função `st_read()`. Essa função pode ser utilizada da seguinte forma:

```
malha_exemplo ← st_read("caminho/para/o/shape/malha_br.shp").
```

Contudo, com a ampla comunidade colaborativa do R, certas operações têm se tornado mais práticas. Hoje, o pacote `geobr`, desenvolvido pelo IPEA, reúne malhas territoriais oficiais de diferentes fontes, como IBGE, CEMADEN e INEP em um repositório único que pode ser acessado diretamente via R, sem necessidade de baixar nenhum shapefile manualmente. Vamos ver como isso funciona:

```
# se não estiver instalado, rodar:  
install.packages("geobr")  
library(geobr)
```

O código está instalando e carregando o pacote `geobr` no R.

O pacote possui funções específicas para cada nível de agregação espacial, todas começando com `read_*`. Por exemplo, se quiséssemos uma malha dos estados brasileiros, poderíamos usar `read_state()`. Para bairros, `read_neighborhood()`. No caso, iremos recuperar a malha de municípios de São Paulo, e portanto utilizaremos `read_municipality()`:

```
# carregando a malha do estado de São Paulo  
malha_sp ← read_municipality("SP", showProgress = F)  
head(malha_sp)
```

```
#> Simple feature collection with 6 features and 4 fields  
#> Geometry type: MULTIPOLYGON  
#> Dimension: XY  
#> Bounding box: xmin: -51.17838 ymin: -23.03648 xmax: -46.54881 ymax:  
-21.19701  
#> Geodetic CRS: SIRGAS 2000  
#>   code_muni          name_muni code_state abbrev_state  
#> 1  3500105          Adamantina    35        SP  
#> 2  3500204          Adolfo       35        SP  
#> 3  3500303          Aguai       35        SP  
#> 4  3500402          Águas Da Prata 35        SP  
#> 5  3500501          Águas De Lindóia 35        SP  
#> 6  3500550          Águas De Santa Bárbara 35        SP  
#>           geom  
#> 1 MULTIPOLYGON (((-51.09093 - ...  
#> 2 MULTIPOLYGON (((-49.69668 - ...  
#> 3 MULTIPOLYGON (((-47.01254 - ...  
#> 4 MULTIPOLYGON (((-46.73069 - ...  
#> 5 MULTIPOLYGON (((-46.635 -22...  
#> 6 MULTIPOLYGON (((-49.28903 - ...
```

Esse código carrega o mapa dos municípios do estado de São Paulo ("SP") e depois exibe as primeiras linhas (informações) desse mapa.

Vemos que os dados estão dispostos de forma similar a uma base de dados tradicional, possuindo agora uma coluna espacial: a coluna `geom`. Ela contém as informações dos polígonos referentes a cada um dos municípios. É importante notar também o Sistema de Coordenadas (CRS) listado - nesse caso, o SIRGAS 2000.

Podemos verificar rapidamente o contorno dos municípios:

```
plot(malha_sp$geom)
```



O código `plot(malha_sp$geom)` gera um gráfico que desenha a forma geográfica (polígonos ou linhas) que está armazenada no objeto `malha_sp`, usando a coluna `geom`, que contém a geometria espacial.

Vemos que em ambos os objetos (dados de dengue e malha espacial) possuímos a coluna referente ao **código do município**, e portanto a usaremos para juntar as duas informações - as informações epidemiológicas e espaciais.

Podemos realizar essa operação através do comando `left_join()`. Perceba que o código do município é chamado de `code_muni` na malha espacial e `cod_ibge` no conjunto de dados de dengue:

```
malha_sp_dengue ← malha_sp %>%
  left_join(
    dengue_2015 %>% select(cod_ibge, casos, pop, inc),
    by = c("code_muni" = "cod_ibge")
  )

head(malha_sp_dengue)
```

```
#> Simple feature collection with 6 features and 7 fields
#> Geometry type: MULTIPOLYGON
#> Dimension:      XY
#> Bounding box:  xmin: -51.17838 ymin: -23.03648 xmax: -46.54881 ymax:
#> -21.19701
#> Geodetic CRS:  SIRGAS 2000
#>   code_muni           name_muni code_state abbrev_state casos   pop
#> 1  3500105          Adamantina       35             SP  1140 34285
#> 2  3500204            Adolfo        35             SP   121  3903
#> 3  3500303            Aguaiá        35             SP  2487 32192
#> 4  3500402        Águas Da Prata       35             SP   336  7558
#> 5  3500501        Águas De Lindóia       35             SP   165 17690
#> 6  3500550 Águas De Santa Bárbara       35             SP    16   6313
#>   inc                      geom
#> 1 332.50693 MULTIPOLYGON (((-51.09093 -...
#> 2 310.01793 MULTIPOLYGON (((-49.69668 -...
#> 3 772.55219 MULTIPOLYGON (((-47.01254 -...
#> 4 444.56205 MULTIPOLYGON (((-46.73069 -...
#> 5 93.27304 MULTIPOLYGON (((-46.635 -22...
#> 6 25.34453 MULTIPOLYGON (((-49.28903 -...
```

O código pega a malha dos municípios de SP (`malha_sp`) e junta (`left_join`) com uma tabela de casos de dengue de 2015 (`dengue_2015`), usando como chave primária de ligação o código do município (`code_muni` em `malha_sp` e `cod_ibge` em `dengue_2015`).

Ele seleciona apenas as colunas `cod_ibge`, `casos`, `pop` (população) e `inc` (incidência) da tabela `dengue_2015` para se vincularem ao objeto `malha_sp`. No final, o resultado é guardado no objeto `malha_sp_dengue`.

`head(malha_sp_dengue)` exibe as primeiras linhas deste novo objeto.

Agora sim! Temos as informações espaciais (coluna `geom`) e epidemiológicas juntas em um objeto. A partir disso, podemos construir nosso primeiro mapa de incidência de dengue em São Paulo no ano de 2015. Vamos utilizar a função `ggplot()` que possui uma função específica para visualização de mapas: a `geom_sf()`. Precisamos especificar duas estéticas (`aes()`): a `geometry`, que recebe o nome da coluna que possui as informações espaciais; e `fill`, que recebe o nome da variável pela qual o mapa será colorido:

```
ggplot(malha_sp_dengue, aes(geometry=geom)) +  
  geom_sf(aes(fill = inc), linewidth = .1) +  
  labs(title = "Incidência de dengue nos municípios de São Paulo, 2015", fill  
  = "Incidência (por 10.000 hab.)") +  
  theme_void()
```



Incidência de dengue nos municípios de São Paulo, 2015



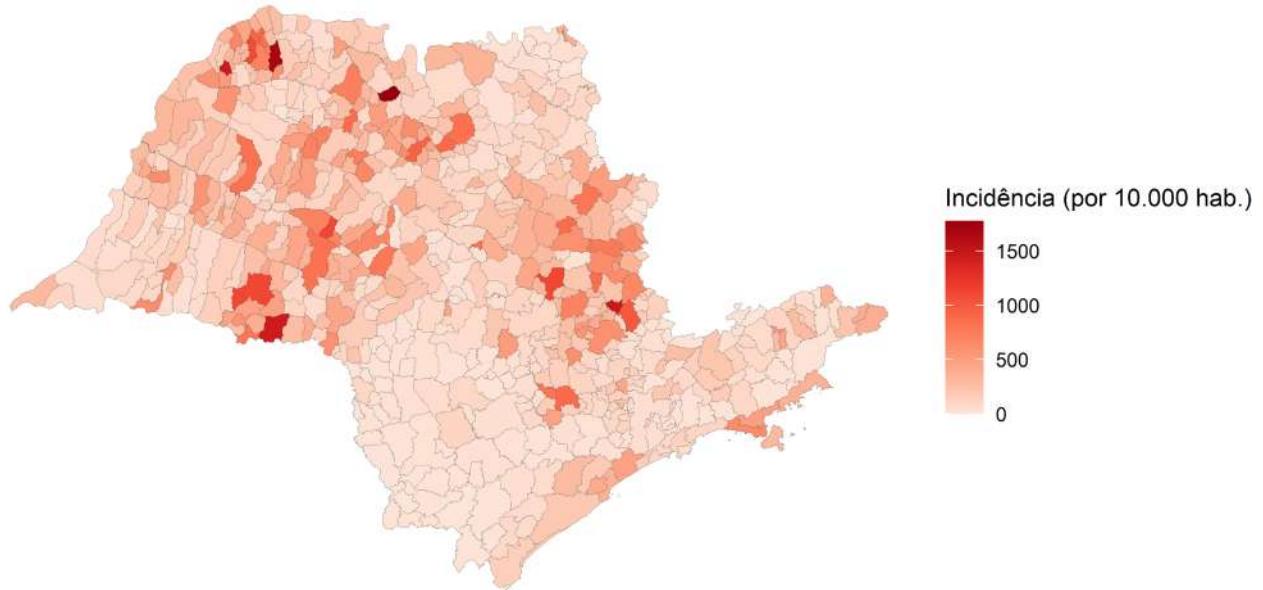
O código gera um mapa temático mostrando a incidência de dengue nos municípios de São Paulo em 2015.

- `ggplot()` inicia o gráfico usando o conjunto de dados `malha_sp_dengue`, mapeando a geometria dos municípios (`geom`).
- `geom_sf(aes(fill = inc), linewidth = .1)` desenha os municípios coloridos conforme a variável `inc` (incidência) e determina a largura das linhas do mapa (`linewidth = .1`).
- `labs()` adiciona um título ao mapa e um rótulo para a legenda.
- `theme_void()` remove elementos visuais como eixos e grades para destacar apenas o mapa.

Podemos alterar a paleta de cores através dos comandos `scale_fill_*`, como `scale_fill_distiller()` e `scale_fill_viridis_c()` para variáveis contínuas e `scale_fill_brewer()` e `scale_fill_viridis_d()` para variáveis discretas ou categorizadas.

```
ggplot(malha_sp_dengue, aes(geometry=geom)) +
  geom_sf(aes(fill = inc), linewidth = .1) +
  labs(title = "Incidência de dengue nos municípios de São Paulo, 2015", fill
  = "Incidência (por 10.000 hab.)") +
  scale_fill_distiller(palette = "Reds", direction=1) +
  theme_void()
```

Incidência de dengue nos municípios de São Paulo, 2015



Nesse código utilizado para criar o mapa temático da incidência de dengue nos municípios de São Paulo em 2015 foi utilizado o argumento `scale_fill_distiller(palette = "Reds", direction = 1)`. Ele define o uso da paleta de cores degradê ("Reds") variando de tons claros a tons escuros de vermelho, seguindo o sentido normal (`direction=1`), para pintar o mapa segundo os valores de incidência.

Pronto! Temos nosso primeiro mapa de incidência de dengue. Conforme discutido anteriormente, o mapa representa uma taxa contínua, que nem sempre é a melhor opção para visualizar os padrões espaciais - ainda mais quando a distribuição da variável não é simétrica.

Vamos definir, portanto, as **quebras** - pontos de corte para divisão da incidência no mapa, conforme estratégias que vimos anteriormente.

```
# quebras pré-definidas (arbitrarias)
quebras_arbitrarias <- c(0, 10, 50, 100, 250, 500, 2000)

malha_sp_dengue <- malha_sp_dengue %>%
  mutate(
    inc_cat = cut(inc, breaks = quebras_arbitrarias, include.lowest=T)
  )
```

O código está criando categorias para a variável `inc` (incidência de dengue) usando faixas de valores pré-definidas (`quebras_arbitrarias`). Em seguida, adiciona uma nova coluna chamada `inc_cat` na tabela `malha_sp_dengue`, indicando a qual intervalo cada valor de `inc` pertence.

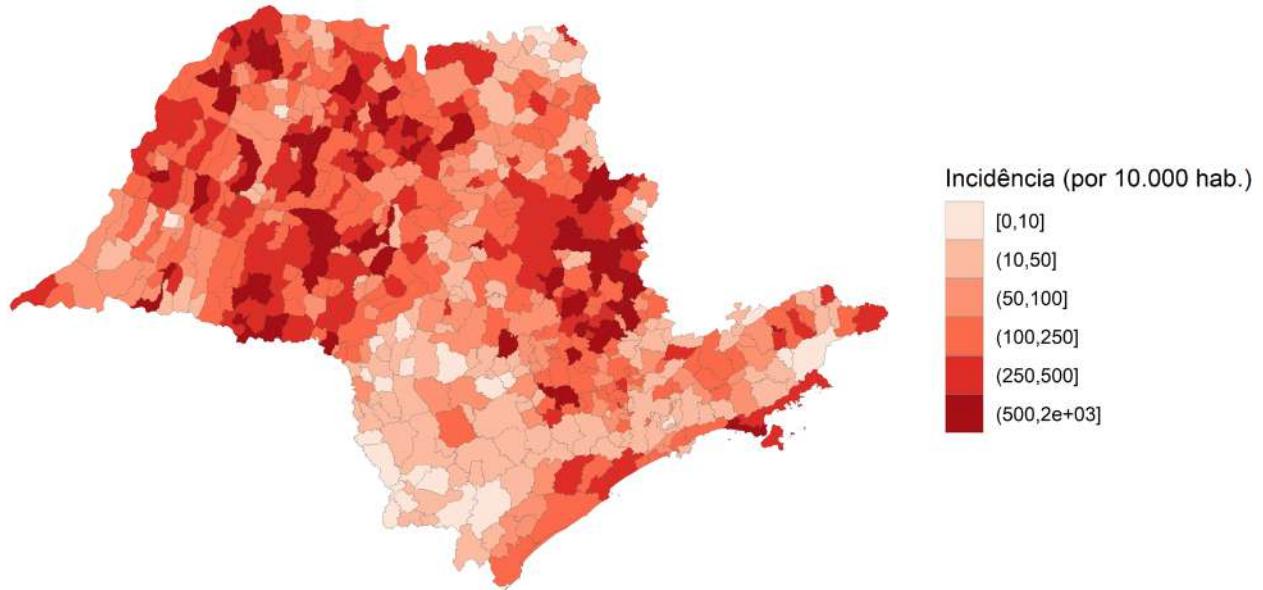
Descrevendo o passo a passo do código:

- Define as quebras dos intervalos: 0, 10, 50, 100, 250, 500, 2000.
- Usa a função `cut()` para categorizar os valores de `inc` conforme essas faixas.
- Garante que o menor valor (0) seja incluído no primeiro intervalo (`include.lowest = TRUE`).
- Salva essas categorias em uma nova variável chamada `inc_cat`.

E, ao realizar o gráfico, usamos [ ] como variável e [ ] para especificar as cores:

```
ggplot(malha_sp_dengue, aes(geometry=geom)) +
  geom_sf(aes(fill = inc_cat), linewidth = .1) +
  labs(title = "Incidência de dengue nos municípios de São Paulo, 2015", fill =
  "Incidência (por 10.000 hab.)") +
  scale_fill_brewer(palette = "Reds", direction=1) +
  theme_void()
```

Incidência de dengue nos municípios de São Paulo, 2015



Vamos explorar as outras formas de definir as quebras (fique à vontade para trocá-las no mapa e comparar as diferenças).

```
# quebras regulares
quebras_regulares <- seq(0,
                           max(dengue_2015$inc),
                           length.out=7) # length.out define o número de categorias + 1

# quebras quantílicas
quebras_quantilicas <- quantile(
                           dengue_2015$inc,
                           probs = seq(0, 1, length.out=7))
```

O código cria dois conjuntos de “quebras” (intervalos) para a variável `inc` do conjunto de dados `dengue_2015`:

- 1. Quebras regulares (quebras\_regulares):** Divide o intervalo de `inc` em 6 categorias de tamanho igual, do valor 0 até o valor máximo de `inc`.
- 2. Quebras quantílicas (quebras\_quantilicas):** Divide os valores de `inc` em 6 categorias que contêm aproximadamente o mesmo número de observações, usando os quantis (percentis).

Para as quebras de Jenks (ou quebras naturais), é necessário ter instalada uma implementação do algoritmo de Jenks. Uma opção é através da biblioteca `BAMMtools`:

```
install.packages("BAMMtools")
library(BAMMtools)
```

```
# quebras naturais ou quebras de Jenks

quebras_jenks ← BAMMtools::getJenksBreaks(
  dengue_sp$inc, k=7
)

quebras_regulares
```

```
#> [1] 0.0000 295.5472 591.0944 886.6416 1182.1888 1477.7360 1773.2832
```

```
quebras_quantilicas
```

```
#> 0% 16.66667% 33.33333% 50% 66.66667% 83.33333% 100%
#> 0.00000 33.81285 79.98215 135.09475 232.76623 393.97020 1773.28316
```

```
quebras_jenks
```

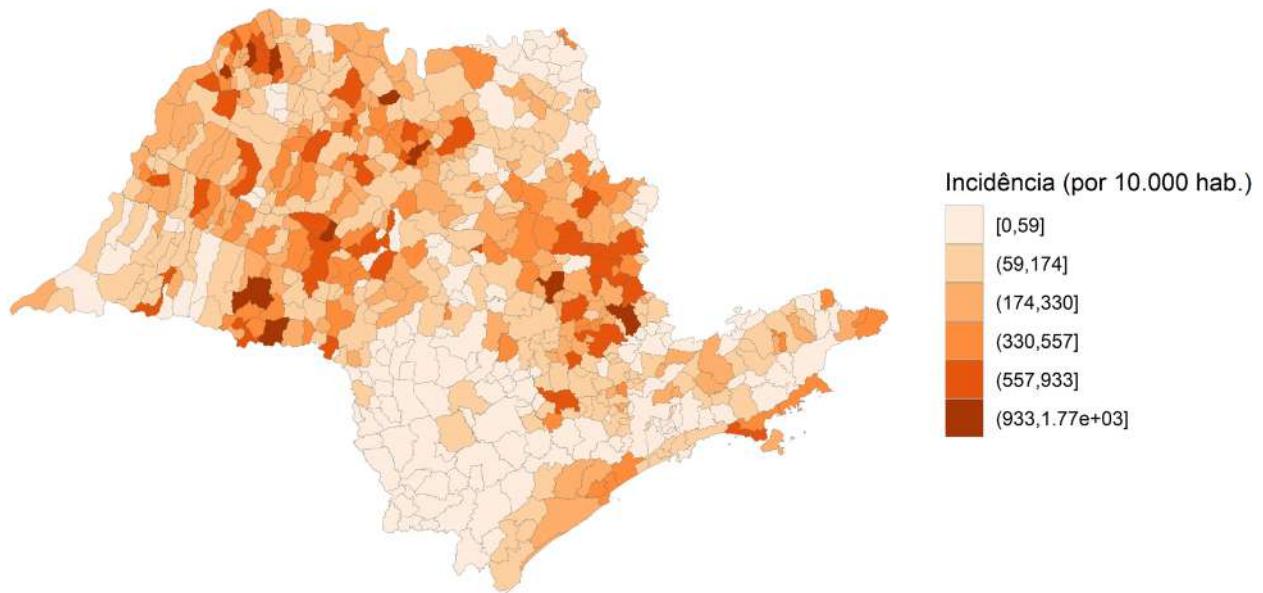
```
#> [1] 0.00000 59.01676 174.00383 329.79113 556.57492 933.26643 1773.28316
```

```
malha_sp_dengue ← malha_sp_dengue %>%
  mutate(
    inc_cat2 = cut(inc, breaks = quebras_jenks, include.lowest=T)
  )
```

O código calcula as quebras naturais (de *Jenks*) para a variável `inc` do conjunto `dengue_sp`, criando 7 classes. Em seguida exibe os três objetos de quebras. E por final o código cria uma nova variável categórica (`inc_cat2`) na tabela `malha_sp_dengue`, classificando `inc` conforme as faixas definidas pelas quebras de *Jenks*.

```
ggplot(malha_sp_dengue, aes(geometry=geom)) +  
  geom_sf(aes(fill = inc_cat2) , linewidth = .1) +  
  labs(title = "Incidência de dengue nos municípios de São Paulo, 2015",  
    subtitle = "Quebras de Jenks", fill = "Incidência (por 10.000 hab.)") +  
  scale_fill_brewer(palette = "Oranges", direction=1) +  
  theme_void()
```

Incidência de dengue nos municípios de São Paulo, 2015  
Quebras de Jenks



Pronto! Através da junção de dados epidemiológicos com malhas espaciais construímos nossas primeiras visualizações em mapas com dados de área, alternando paletas e maneiras de definir os pontos de corte.

Vamos voltar agora para o conteúdo teórico do curso!

## *Autocorrelação espacial*

Quando analisamos dados distribuídos espacialmente, há a possibilidade de os dados possuírem **autocorrelação espacial**. Assim como em séries temporais (veja o curso “Análises de séries temporais em R aplicadas à vigilância em saúde”), a autocorrelação é a presença de correlação, ou dependência, da variável com ela mesma (no caso temporal, com ela mesma em diferentes pontos no tempo). No contexto espacial, nos referimos à correlação da variável de interesse (como por exemplo, incidência) com ela mesma entre áreas mais próximas.

Ou seja, se há aglomerados de uma certa doença (como vimos nos mapas de dengue), possivelmente os dados possuem uma autocorrelação espacial positiva, pois se um determinado município possui uma alta incidência de dengue, há uma **maior probabilidade** de seus municípios vizinhos estarem em uma situação parecida.

Há também o caso de autocorrelação inversa, que indica uma correlação inversa entre um município e seus vizinhos. Ou seja, quando seus vizinhos estão com uma incidência baixa, há uma maior probabilidade de o município ter uma incidência alta - esse tipo de autocorrelação pode nos ajudar a identificar áreas destoantes.

Veremos a seguir formas de identificar e mensurar essa autocorrelação presente nos dados, de forma global e local.

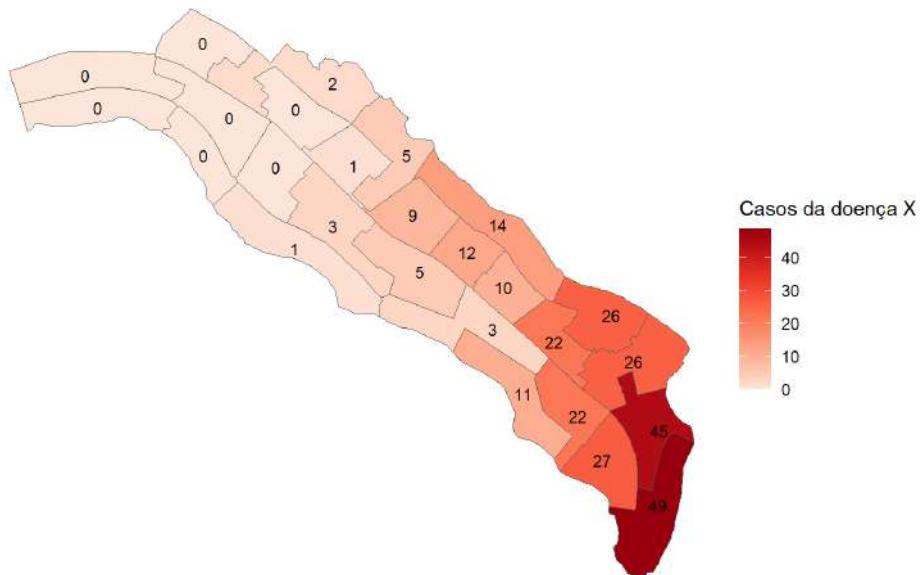
Antes de mensurar a presença de dependência espacial nos dados, devemos distinguir esse fenômeno de dependência entre efeitos de primeira e segunda ordem.

## *Efeitos de primeira ordem - tendência*

Quando a dependência espacial sobre os dados se dá devida a efeitos de uma variação de larga escala (ou estrutural), ela é caracterizada como um efeito de primeira ordem. Esses efeitos são observados quando, por exemplo, há tendência de concentração da doença em regiões litorâneas: há uma determinação geográfica subjacente que induz essa tendência, levando a maiores valores de incidência em uma determinada região. Outro caso pode acontecer quando há uma tendência clara de aumento de incidência de uma doença no sentido Norte - Sul.

Vejamos um exemplo de efeito de primeira ordem. Retornando ao nosso conjunto de dados hipotético, vamos supor que uma doença X tenha a seguinte distribuição em um município (suponha população aproximadamente igual entre as áreas):

**Figura 31: Exemplo de efeito espacial de primeira ordem: conforme vamos em direção ao leste, os casos da doença X parecem aumentar.**



Vemos na Figura 31 que a distribuição da doença se dá de forma mais intensa conforme andamos em sentido Leste (ou Sudeste). Pode-se considerar que há, portanto, uma tendência “macro” no processo, em que o mesmo aumenta de intensidade ao avançar em determinado sentido, caracterizando um **efeito de primeira ordem**. Geralmente, estes efeitos se dão devido a alguma característica ambiental, como maior presença de áreas verdes, maior proximidade com corpos fluviais, maior/menor temperatura, umidade, pressão, entre outras.

## *Efeitos de segunda ordem - dependência local*

Por outro lado, efeitos de segunda ordem (ou efeitos meso, ou efeitos locais) caracterizam uma dependência de uma região sobre seus vizinhos mais próximos. Um exemplo clássico é quando observamos “clusters” de uma doença infecciosa em algumas regiões: se há uma alta de casos em um determinado município, é muito provável que haja também uma alta de casos nos municípios vizinhos, seja pela própria dinâmica infecciosa da doença ou por características ambientais locais propícias para o desenvolvimento de vetores.

Vamos agora nos voltar ao exemplo dos casos de dengue em São Paulo, e verificar a incidência da distribuição de casos em 2020:

```
dengue_2020 <- dengue_sp %>%
  filter(ano == 2020)

malha_sp_dengue <- malha_sp %>%
  left_join(
    dengue_2020 %>% select(cod_ibge, casos, pop, inc),
    by = c("code_muni" = "cod_ibge")
  )
```

O código filtra os dados de dengue para o ano de 2020 e junta esses dados com a malha dos municípios de São Paulo, criando um novo objeto `malha_sp_dengue` que contém informações sobre casos, população e incidência de dengue.

Vamos optar pela classificação das faixas pelas quebras de Jenks:

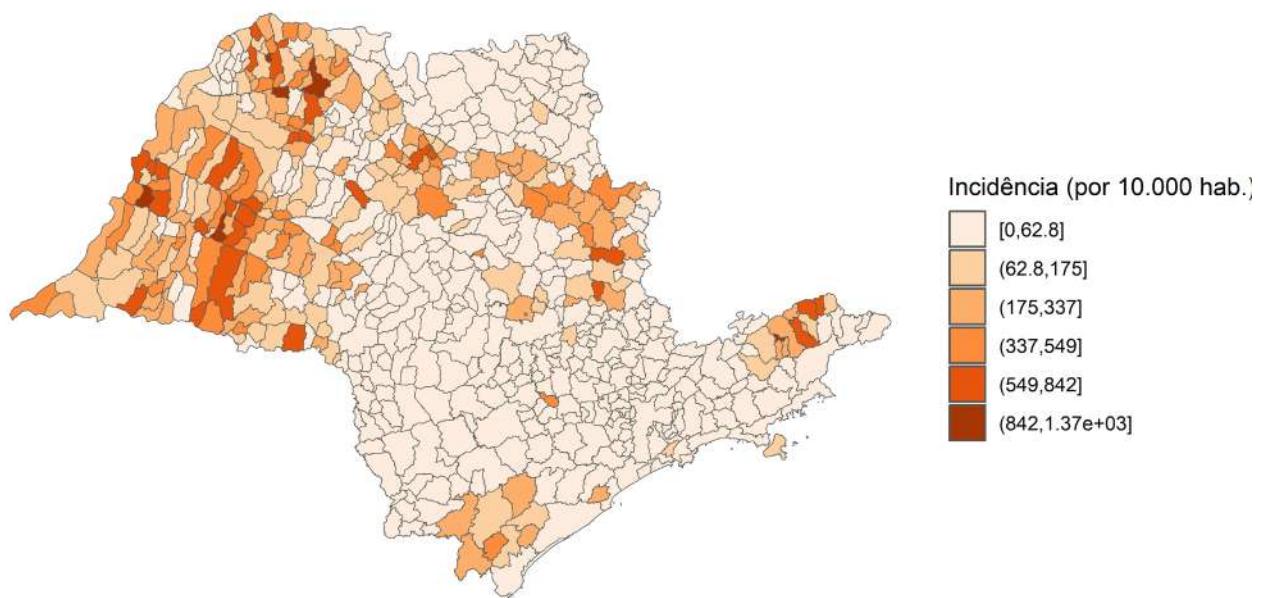
```
quebras_jenks ← BAMMtools::getJenksBreaks(
  dengue_2020$inc, k=7
)

malha_sp_dengue ← malha_sp_dengue %>%
  mutate(
    inc_cat = cut(inc, breaks = quebras_jenks, include.lowest=T)
  )

ggplot(malha_sp_dengue, aes(geometry=geom)) +
  geom_sf(aes(fill = inc_cat), linewidth = .1) +
  labs(title = "Incidência de dengue nos municípios de São Paulo, em 2020",
       subtitle = "Quebras de Jenks",
       fill = "Incidência (por 10.000 hab.)") +
  scale_fill_brewer(palette = "Oranges", direction=1) +
  theme_void()
```

**Figura 32: Incidência de dengue nos municípios de São Paulo, 2020.**

Incidência de dengue nos municípios de São Paulo, em 2020  
Quebras de Jenks



O código gera um mapa temático da incidência de dengue nos municípios de São Paulo em 2020, usando o método de quebras naturais de *Jenks* para categorizar os valores.

Passo a passo:

- **Definição das quebras:** Calcula 7 intervalos de *Jenks* ( $k = 7$ ) para a variável `inc` (incidência) do conjunto `dengue_2020`.
- **Criação da variável categórica:** No objeto `malha_sp_dengue`, cria-se uma nova coluna `inc_cat`, categorizando `inc` de acordo com as quebras de *Jenks*.
- **Construção do mapa** usando o `ggplot2`.

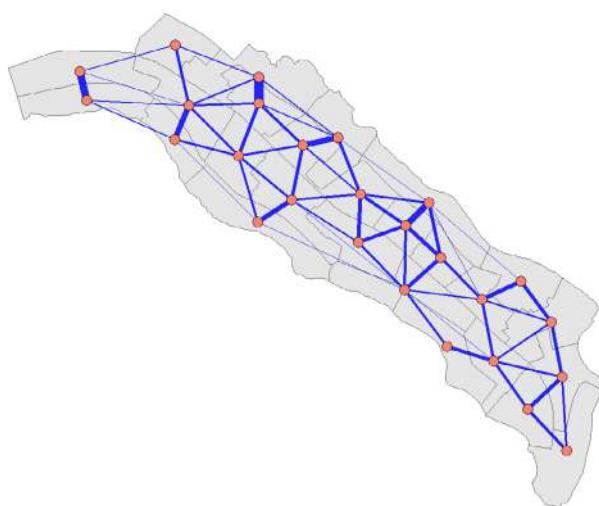
Vemos na Figura 32 que há certos “grupos de concentração” da doença no estado. Pode-se listar ao menos quatro regiões em que há um foco de incidência maior entre municípios próximos. Não vemos, no geral, uma tendência clara ao longo de todo o estado como no exemplo anterior. Nesse caso, temos aglomerados da doença nas regiões Oeste e Noroeste do estado, assim como no Sul e Leste também. Esse processo de dependência espacial se aproxima, portanto, de um processo de **segunda ordem**.

## *Medidas de proximidade em dados de área*

Vimos que as definições de autocorrelação espacial relacionam a variável de interesse entre as áreas próximas ou vizinhas. Portanto, para identificar, testar e quantificar a presença de autocorrelação nos dados de interesse, temos que definir quais áreas são próximas das outras a partir de **medidas de proximidade**. Essas podem se basear exclusivamente nas fronteiras territoriais, ou em outros fatores como distância (exemplo: distância entre os centroides, ou distância entre suas sedes) ou fluxo (exemplo: fluxo pendular) entre os municípios. Essa escolha se dará a partir dos mecanismos envolvidos no estudo do agravo em questão; há bastante sentido, por exemplo, em considerar o fluxo de deslocamento entre municípios para relacionar espacialmente a distribuição de uma doença como a covid-19. Dessa forma, municípios que tem maior fluxo de pessoas entre si serão mais “próximos” do que municípios geograficamente próximos, mas que não possuem tanto fluxo. Contudo, a disponibilidade de dados sobre esses fatores pode ser um fator limitador, o que nos leva a optar por abordagens mais simples.

Vejamos a Figura 33. Neste caso, calculamos a proximidade com base na distância entre os centroides das áreas. Quanto mais perto duas áreas são, mais forte (pois mais próxima) é sua ligação (representada pelas linhas azuis).

**Figura 33:** Exemplo de cálculo de proximidade entre as áreas do exemplo hipotético.



Essas distâncias serão incorporadas a um elemento essencial ao tratar de autocorrelação espacial: a matriz de vizinhança, que veremos a seguir.

## *Matriz de vizinhança*

Uma matriz de vizinhança é uma forma que temos de relacionar as áreas entre si, indicando quais possuem conexões (vizinhança) com as outras e quais os **pesos** dessas conexões. Geralmente temos uma matriz  $n \times n$ , onde temos todos os municípios nas linhas e os mesmos municípios nas colunas. Cada célula então, de uma linha  $l$  com uma coluna  $C$  indica se o município  $l$  e  $C$  tem relação de vizinhança - e qual o peso dessa relação. Os pesos podem ser fixos (como 0 para uma relação de não vizinhança e 1 para uma relação de vizinhança) ou dependerem de algum outro fator, como a distância entre os municípios.

Vamos considerar o seguinte exemplo, de um recorte de alguns municípios do estado de São Paulo (Figura 34).

**Figura 34: Recorte de 7 municípios do estado de São Paulo.**



Uma possível representação das relações entre os municípios exibidos e sua vizinhança seria:

Município	Arujá	Biritiba-Mirim	Guararema	Itaquaquecetuba	Mogi Das Cruzes	Salesópolis	Suzano
Arujá	0	0	0	1	1	0	0
Biritiba-Mirim	0	0	1	0	1	1	0
Guararema	0	1	0	0	1	1	0
Itaquaquecetuba	1	0	0	0	1	0	1
Mogi Das Cruzes	1	1	1	1	0	0	1
Salesópolis	0	1	1	0	0	0	0
Suzano	0	0	0	1	1	0	0

Vemos que a primeira linha, que representa o município de Arujá, possui valor 1 nas colunas Itaquaquecetuba e Mogi das Cruzes, pois são os municípios com quem faz fronteira. Mogi das Cruzes, por sua vez, possui valor 1 para quase todos os municípios listados, com exceção de Salesópolis. A matriz exibida é chamada de matriz de vizinhança por contiguidade binária; pois define se um município é vizinho de outro se compartilham fronteira, e caso exista, o peso é atribuído é 1. Caso contrário, o peso é 0. Este tipo de matriz de vizinhança é chamado de vizinhança **por contiguidade**.

As estratégias mais comuns para definição da matriz de vizinhança são:

**Por contiguidade:** Como no exemplo acima, a célula  $l, C$  assumirá valor 1 se o município  $l$ , e o município  $C$  compartilharem fronteira e 0 caso contrário;

**Por  $k$  vizinhos mais próximos:** a célula  $l, C$  será 1 se  $C$  estiver entre os  $k$  municípios mais próximos de  $l$ . Pode-se considerar, por exemplo, a distância entre as sedes ou centroides dos municípios;

**Por distância:** a célula  $l, C$  será 1 se  $C$  estiver até a uma distância  $d$  pré-definida de  $l$ . Caso contrário, será 0.

Vamos implementar cada uma dessas estratégias em nosso mapa do estado de São Paulo.

## *Por contiguidade*

O pacote `spdep` traz diversas operações envolvendo correlação e dependência espacial implementadas, e portanto o utilizaremos para definir nossas matrizes de vizinhança. Caso não esteja instalado, precisaremos rodar `install.packages("spdep")`.

```
# se não estiver instalado, rodar:  
install.packages("spdep")  
library(spdep)
```

A `library(spdep)` no R é usada para análise de estatística espacial. Ela fornece funções para criar estruturas de vizinhança, calcular pesos espaciais e aplicar testes e modelos que mensuram a autocorrelação espacial.

A partir da malha de municípios de São Paulo, vamos definir nossa matriz de vizinhança. O pacote `spdep` possui diferentes funções de transformação nesse sentido, onde podemos partir de objetos espaciais como o que temos (`malha_sp`) e transformá-lo em objetos como de relação entre vizinhos. Uma dessas funções é a `poly2nb` (polígono (`poly`) para (`to - 2`) vizinhança (`neighbours - nb`)). Ela automaticamente define uma matriz de **contiguidade** a partir da presença de fronteiras na malha espacial em questão. Vamos testá-la e criar o objeto `viz_sp`, que utilizaremos mais adiante.

```
viz_sp <- poly2nb(malha_sp)  
viz_sp
```

```
#> Neighbour list object:  
#> Number of regions: 645  
#> Number of nonzero links: 3530  
#> Percentage nonzero weights: 0.8485067  
#> Average number of links: 5.472868  
#> 1 region with no links:  
#> 233  
#> 2 disjoint connected subgraphs
```

A função `poly2nb(malha_sp)` cria uma lista de vizinhança com base no polígono `malha_sp`.

- O objeto `malha_sp` deve ser um objeto `sf` ou `SpatialPolygons`.
- A função `poly2nb()` é do pacote `spdep`.
- Ela define vizinhos com base no compartilhamento de bordas (isto é, dois polígonos são vizinhos se eles dividem um lado).
- O objeto `viz_sp` representa o objeto resultante, que é da classe `nb` (`neighbour list`). Cada elemento da lista contém os índices dos polígonos vizinhos de cada polígono de `malha_sp`.
- `viz_sp` impresso no console vai mostrar para cada polígono, quais são os seus vizinhos.

Ótimo! Temos, então, um objeto do tipo “lista de vizinhos” (*Neighbour list*). Ao chamar o objeto, temos um resumo dessa matriz gerada: são 645 municípios, 3.526 links (ligações), tendo cada região em média 5.47 links (ou seja, 5 vizinhos).

Somos informados também, que uma região (de *id* 233) não possui links, ou seja, não possui vizinhos. Isso geralmente é um problema quando falamos de análise de dependência espacial. Para testar nossas hipóteses de autocorrelação espacial e utilizar outras técnicas relacionadas, é ideal que todas as regiões tenham ao menos um vizinho. Pode acontecer que, em caso de linhas ou regiões desconexas, ao gerar a lista de vizinhos automaticamente a partir da malha, alguma dessas regiões fique desconectada das demais. Mas lidaremos com isso mais à frente.

Vamos tentar visualizar nossa vizinhança obtida:

```
# obtendo as coordenadas dos centroides da malha

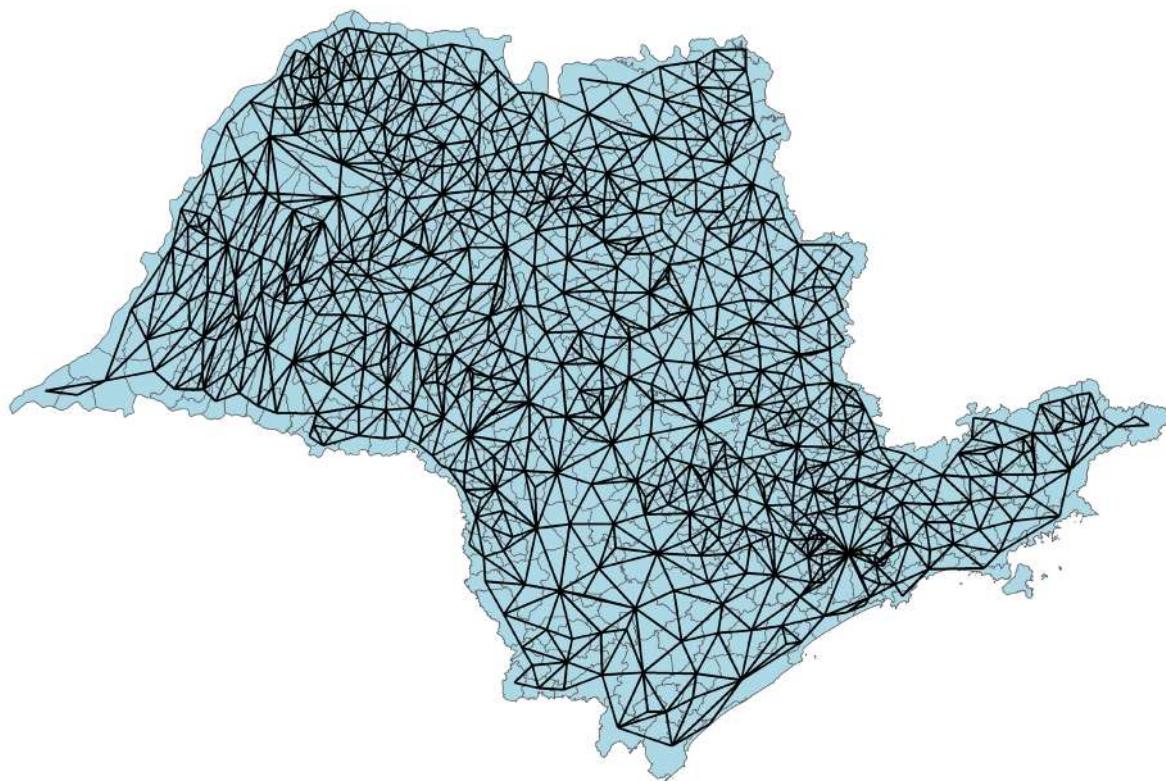
centroides_sp <- st_centroid(malha_sp)
centroides_coordenadas <- st_coordinates(centroides_sp)

# a partir delas, transformamos nossas relações de vizinhança em linhas:

viz_linhos <- nb2lines(nb = viz_sp,
                      coords = centroides_coordenadas) %>%
  st_as_sf()
st_crs(viz_linhos) <- st_crs(malha_sp)

ggplot(malha_sp, aes(geometry=geom)) +
  geom_sf(fill = "lightblue") +
  geom_sf(data = viz_linhos, aes(geometry=geometry)) +
  ggtitle("Matriz de vizinhança por contiguidade, estado de São Paulo.") +
  theme_void()
```

**Figura 35: Matriz de vizinhança por contiguidade dos municípios do estado de São Paulo.**



Esses códigos criam e desenham as relações de vizinhança entre os municípios do estado de São Paulo, representando essas relações com linhas conectando os centroides (pontos centrais) dos polígonos da malha.

- `centroides_sp ← st_centroid(malha_sp)`: calcula o ponto central (centroide) de cada área (polígono) da malha `malha_sp`.
- `centroides_coordenadas ← st_coordinates(centroides_sp)`: Extrai as coordenadas desses centroides para trabalhar com elas depois.
- `viz_linhos ← nb2Lines(nb = viz_sp, coords = centroides_coordenadas) %>% st_as_sf()`: Converte a lista de vizinhos `viz_sp` em linhas, usando as coordenadas dos centroides. O resultado é transformado em um objeto `sf` (simple features).
- `st_crs(viz_linhos) ← st_crs(malha_sp)`: Define o sistema de referência de coordenadas (CRS) do objeto `viz_linhos` para ser o mesmo que o da malha original `malha_sp`.
- Logo em seguida com a função `ggplot`, o mapa é plotado primeiro com a malha dos polígonos (`malha_sp`) com cor azul clara. E depois com as linhas de vizinhança sobre a malha, ligando áreas vizinhas.

Vemos na Figura 35 que todos os municípios que tocam suas fronteiras estão conectados. Os municípios da “borda” do estado tendem a ter menos vizinhos (alguns têm apenas um) e agora conseguimos visualizar o município sem nenhum vizinho. Trata-se de Ilhabela, no Litoral Norte de São Paulo, que justamente por ser uma ilha, não possui fronteira física com nenhum outro município na malha considerada.

Vamos, então, voltar ao nosso objeto de vizinhança `viz_sp`.

`viz_sp`

```
#> Neighbour list object:
#> Number of regions: 645
#> Number of nonzero links: 3530
#> Percentage nonzero weights: 0.8485067
#> Average number of links: 5.472868
#> 1 region with no links:
#> 233
#> 2 disjoint connected subgraphs
```

Somos informados que o município que não possui vizinhos é o 233. Vamos verificar se é realmente Ilhabela:

```
malha_sp$name_muni[233]
```

```
#> [1] "Ilhabela"
```

Pegando o nome do município que está na posição 233 do objeto `malha_sp`.

Sim, é Ilhabela. Vamos, portanto, conectá-la manualmente a um outro município. Como a chegada a Ilhabela é feita através de balsa partindo de São Sebastião, faz sentido conectar-a a esse município.

Vamos verificar qual o índice que corresponde a São Sebastião:

```
which(malha_sp$name_muni == "São Sebastião")
```

```
#> [1] 567
```

Procura a posição (o número do índice) onde o nome do município é “São Sebastião” dentro da coluna `name_muni` do objeto `malha_sp`.

Agora que sabemos o índice dos dois municípios, podemos modificar nosso objeto `viz_sp` para conectar-los:

Incluindo São Sebastião como vizinho de Ilhabela:

```
viz_sp[[233]] <- 567L
```

Está forçando a região número 233 (que antes não tinha vizinhos) a ter como vizinho o município de número 567.

Note que utilizamos a notação `567L` ao invés de somente `567`. Isso se deve porque a estrutura de vizinhança aceita somente índices inteiros. No ambiente R, quando trabalhamos com números sem informar explicitamente seu tipo, o ambiente assume que é um `double` (número real de precisão dupla):

```
class(567)
```

```
#> [1] "numeric"
```

```
class(567L)
```

```
#> [1] "integer"
```

```
is.double(567)
```

```
#> [1] TRUE
```

```
is.double(567L)
```

```
#> [1] FALSE
```

O comando `viz_sp[[233]] ← 567L` altera manualmente a lista de vizinhança espacial `viz_sp`, atribuindo ao município de índice 233 o número 567 como seu vizinho. A letra `L` após o número indica que ele é tratado como um inteiro (`integer`) no R, o que é necessário porque listas de vizinhança esperam índices inteiros. Essa modificação é útil para corrigir problemas de regiões isoladas, garantindo que todas as áreas tenham pelo menos um vizinho para análises espaciais posteriores.

Logo, precisamos ter essa rigorosidade ao alterar os vizinhos senão o pacote lança-á um erro. Agora, vamos fazer o inverso (que também é necessário!): incluir Ilhabela como vizinho de São Sebastião:

```
viz_sp[[567]] ← c(viz_sp[[567]], 233L)
```

O comando `viz_sp[[567]] <- c(viz_sp[[567]], 233L)` está atualizando a lista de vizinhança do município 567, adicionando o município 233 como seu novo vizinho. Ou seja, ele pega os vizinhos que 567 já tinha (`viz_sp[[567]]`) e junta (`c(...)`) o número 233 (especificado como inteiro com `L`), garantindo que a relação de vizinhança seja bidirecional: se 233 tem 567 como vizinho, então 567 também passa a ter 233 como vizinho.

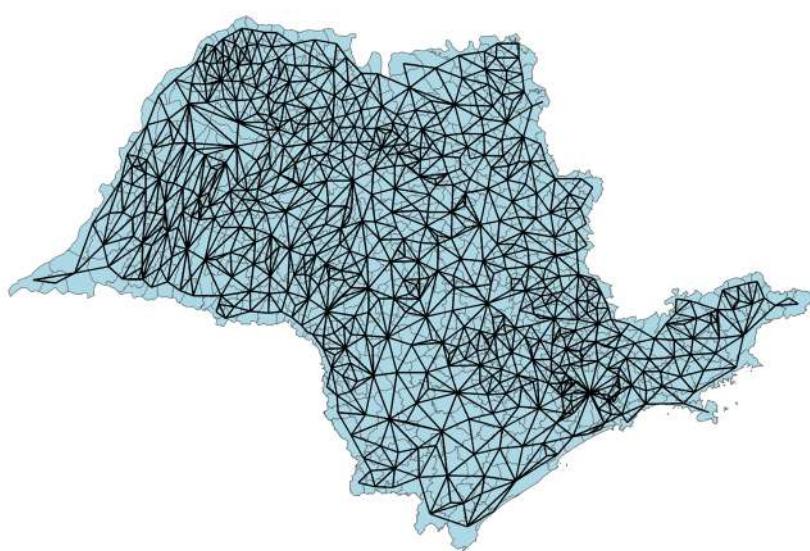
E agora, se repetirmos o comando do gráfico realizado anteriormente, vemos a inclusão da ligação entre Ilhabela e São Sebastião (Figura 36):

```
viz_linhas <- nb2lines(nb = viz_sp,
                      coords = centroides_coordenadas) %>%
  st_as_sf()
st_crs(viz_linhas) <- st_crs(malha_sp)

ggplot(malha_sp, aes(geometry=geom)) +
  geom_sf(fill = "lightblue") +
  geom_sf(data = viz_linhas, aes(geometry=geometry)) +
  labs(title="Matriz de vizinhança por contiguidade, estado de São Paulo.",
       subtitle = "Após inclusão de Ilhabela") +
  theme_void()
```

**Figura 36: Matriz de vizinhança por contiguidade modificada dos municípios do estado de São Paulo.**

Matriz de vizinhança por contiguidade, estado de São Paulo.  
Após inclusão de Ilhabela



O código acima transforma a lista de vizinhança `viz_sp` em linhas que conectam os centroides das regiões (`centroides_coordenadas`) usando a função `nb2Lines`, depois converte essas linhas em um objeto espacial (`sf`) com `st_as_sf()`. Em seguida, ajusta o sistema de coordenadas das linhas para ser igual ao da malha `malha_sp` com `st_crs()`. Por fim, cria um mapa com `ggplot2`, desenhando as áreas (`malha_sp`) preenchidas de azul claro e sobrepondo as linhas de vizinhança (`viz_linhas`), adicionando título, subtítulo e removendo elementos gráficos extras (`theme_void()`).

Verificando o objeto de vizinhança novamente:

```
viz_sp
#> Neighbour list object:
#> Number of regions: 645
#> Number of nonzero links: 3532
#> Percentage nonzero weights: 0.8489874
#> Average number of links: 5.475969
#> 2 disjoint connected subgraphs
```

Agora sim! Todos os municípios estão conectados, e não temos nenhum município sem link.

## *Por vizinhos mais próximos*

A segunda estratégia possível que vimos é a definição dos vizinhos a partir dos  $k$  municípios mais próximos a um município específico. Repare essa estratégia no exemplo, considerando os dois municípios mais próximos ( $k = 2$ ) para os municípios mostrados anteriormente na Figura 33:

Município	Arujá	Biritiba-Mirim	Guararema	Itaquaquecetuba	Mogi Das Cruzes	Salesópolis	Suzano
Arujá	0	0	0	1	0	0	1
Biritiba-Mirim	0	0	0	0	1	1	0
Guararema	0	1	0	0	1	0	0
Itaquaquecetuba	1	0	0	0	0	0	1
Mogi Das Cruzes	0	1	0	0	0	0	1
Salesópolis	0	1	1	0	0	0	0
Suzano	0	0	0	1	1	0	0

Vemos que agora nesse caso podemos ter **assimetrias**: não necessariamente se  $C$  é um dos  $k$  municípios mais próximos de  $l$ ,  $l$  será um dos vizinhos mais próximos de  $C$ . Isso pode ser visualizado no exemplo apresentado visto que Mogi das Cruzes é vizinha de Guararema (pois está entre suas duas cidades mais próximas) mas Guararema não é vizinha de Mogi, que tem outros dois municípios que são mais próximos (Figura 37).

```

viz_sp2 ← centroides_coordenadas %>%
  knearneigh(k=3) %>%
  knn2nb()

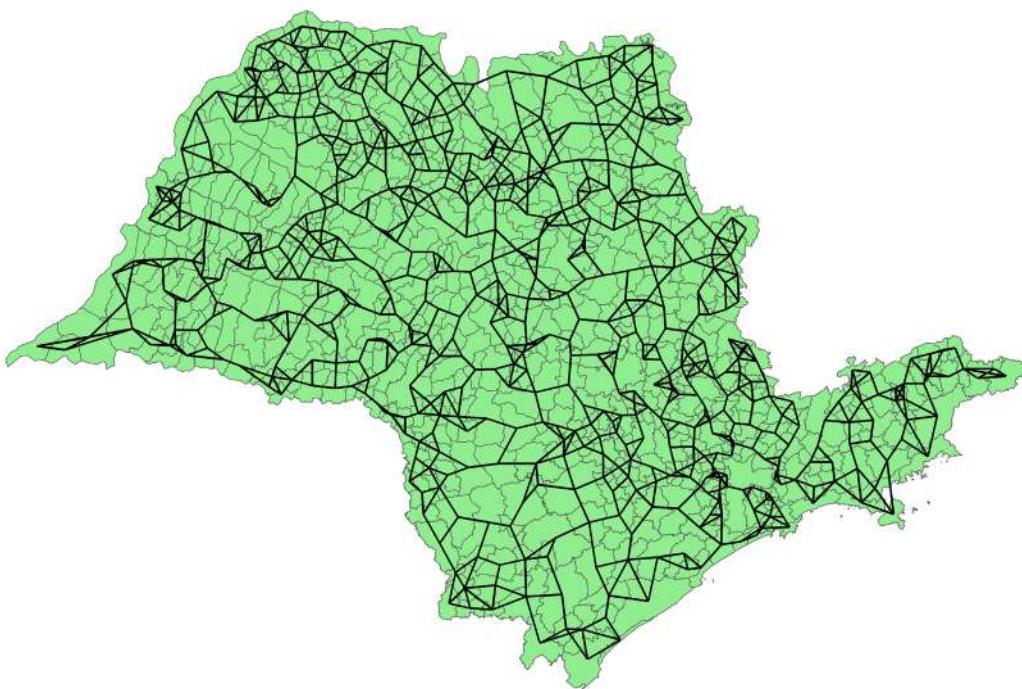
viz_linhas2 ← nb2lines(nb = viz_sp2,
                      coords = centroides_coordenadas) %>%
  st_as_sf()
st_crs(viz_linhas2) ← st_crs(malha_sp)

ggplot(malha_sp, aes(geometry=geom)) +
  geom_sf(fill = "lightgreen") +
  geom_sf(data = viz_linhas2, aes(geometry=geometry)) +
  ggtitle("Matriz de vizinhança por k = 3 vizinhos mais próximos, estado de São Paulo.") +
  theme_void()

```

**Figura 37: Matriz de vizinhança por k vizinhos mais próximos dos municípios do estado de São Paulo.**

Matriz de vizinhança por k = 3 vizinhos mais próximos, estado de São Paulo.



O script cria uma matriz de vizinhança baseada nos 3 vizinhos mais próximos (k=3) a partir dos centroides dos municípios, desenha as linhas de conexão entre eles e depois plota isso sobre o mapa do estado de São Paulo.

## *Por distância*

Uma terceira estratégia que podemos adotar é a definição de vizinhos a partir de uma distância pré-definida. Nesse caso, a célula  $l, C$  será 1 se  $C$  estiver até a uma distância  $d$  pré-definida de  $l$ . Para definir essa distância, é importante entender como as distâncias entre as áreas em questão estão distribuídas.

Vamos exibir a matriz de distâncias e sua distribuição para o nosso exemplo de 7 municípios do estado de São Paulo:

	Arujá	Biritiba-Mirim	Guararema	Itaquaquecetuba	Mogi Das Cruzes	Salesópolis	Suzano
Arujá	0,0	40,4	27,1	8,5	24,5	53,4	24,7
Biritiba-Mirim	40,4	0,0	22,2	36,7	17,9	18,9	29,6
Guararema	27,1	22,2	0,0	28,5	20,5	27,9	32,7
Itaquaquecetuba	8,5	36,7	28,5	0,0	19,3	52,0	16,5
Mogi Das Cruzes	24,5	17,9	20,5	19,3	0,0	35,2	13,4
Salesópolis	53,4	18,9	27,9	52,0	35,2	0,0	48,0
Suzano	24,7	29,6	32,7	16,5	13,4	48,0	0,0

Geralmente, trabalhamos com conjunto de áreas muito maiores (estado de São Paulo: 645 municípios), o que torna inviável olhar uma matriz de distância entre cada área (uma matriz  $645 \times 645 = 416.025$  distâncias). Portanto, é comum recorrermos a métricas resumo, como médias, medianas e quantis da distribuição de distâncias - ou até mesmo a técnicas gráficas como um histograma - para a definição de um ponto de corte (veremos a seguir). Por enquanto, parece razoável estabelecer o limite de  $d = 20\text{km}$  para definição de nossa matriz de vizinhança com base nas distâncias:

Município	Arujá	Biritiba-Mirim	Guararema	Itaquaquecetuba	Mogi Das Cruzes	Salesópolis	Suzano
Arujá	0	0	0	1	0	0	0
Biritiba-Mirim	0	0	0	0	1	1	0
Guararema	0	0	0	0	0	0	0
Itaquaquecetuba	1	0	0	0	1	0	1
Mogi Das Cruzes	0	1	0	1	0	0	1
Salesópolis	0	1	0	0	0	0	0
Suzano	0	0	0	1	1	0	0

Agora, temos uma matriz simétrica novamente - pois se  $l$  está a uma distância de até 20km de  $C$ , então o contrário também é verdadeiro. Vemos também que o número de vizinhos muda: alguns municípios como Mogi das Cruzes possuem 3 vizinhos pois têm três municípios em até um raio de 20km de distância; outros possuem 1, como Arujá; e Guararema neste caso ficou sem vizinhos, pois está mais afastada dos demais.

Vamos ver como essa situação se aplica ao contexto dos 645 municípios do estado de São Paulo. Primeiro, vamos calcular as distâncias entre os centroides dos municípios. Para isso, convertemos a malha para o sistema de coordenadas UTM (31982), que permite termos coordenadas em metros, ao invés de graus:

```
# convertendo para UTM, calculando as coordenadas dos centroides novamente e por fim, as distâncias:
```

```
centroides_coordenadas_metros ← malha_sp %>%
  st_transform(crs = 31982) %>%
  st_centroid() %>%
  st_coordinates()

distancias_sp ← centroides_coordenadas_metros %>%
  dist()

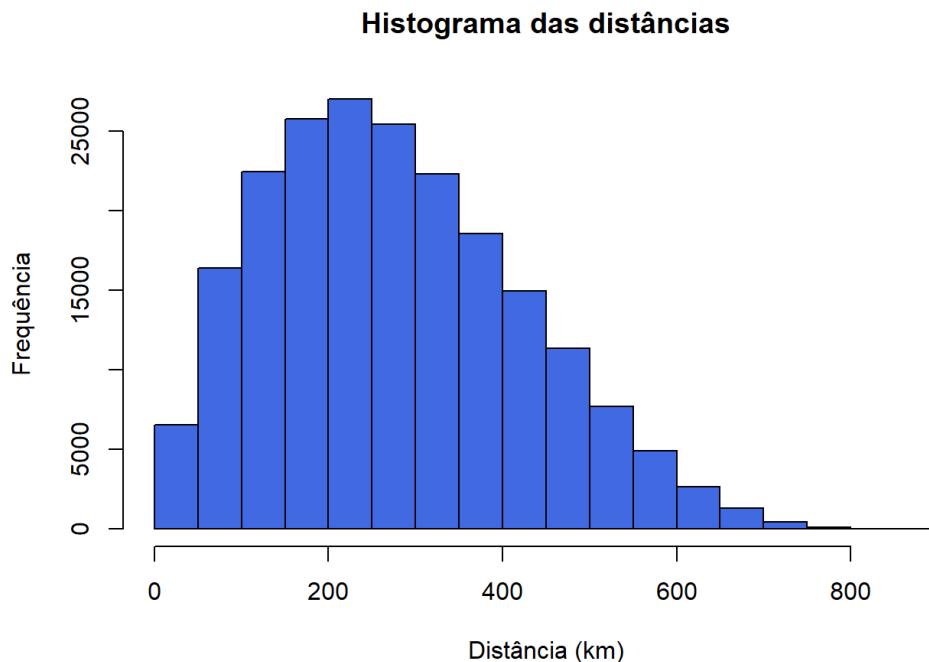
# um objeto da classe dist
class(distancias_sp)
```

```
#> [1] "dist"
```

```
# convertendo para km
distancias_sp <- distancias_sp/1000

distancias_sp %>%
  as.vector() %>%
  hist(main = "Histograma das distâncias",
       xlab = "Distância (km)",
       ylab = "Frequência",
       col = "royalblue")
```

**Figura 38: Histograma das distâncias entre os municípios de São Paulo.**



O script começa transformando a malha espacial `malha_sp` para o sistema de coordenadas UTM (EPSG 31982), que utiliza metros como unidade de medida. Em seguida, calcula o centroide de cada polígono (isto é, o ponto central de cada município) e extrai as suas coordenadas X e Y.

`distancias_sp ← centroides_coordenadas_metros %>% dist()`: A partir dessas coordenadas, é calculada a matriz de distâncias entre todos os centroides, resultando em um objeto do tipo `dist`, onde as distâncias estão inicialmente em metros.

Depois, todas as distâncias são convertidas de metros para quilômetros, dividindo os valores por 1.000.

Por fim, o código transforma essas distâncias em um vetor e constrói um histograma, visualizando a distribuição da frequência das distâncias (em quilômetros) entre os municípios da malha.

Vemos na Figura 38 que a distribuição das distâncias entre os municípios de São Paulo se encontra na faixa de 200-250km, podendo chegar até 800km. Vamos observar os quantis dessa distribuição:

```
quantile(distancias_sp,
          probs = c(0.01, 0.025, 0.05, 0.10, 0.25, 0.5))
```

```
#>      1%     2.5%      5%     10%     25%     50%
#> 28.12763 44.34926 64.25050 94.52491 163.10356 260.96230
```

O comando calcula o valor das distâncias que se encontram nos percentis 1%, 2,5%, 5%, 10%, 25% e 50% da distribuição de distâncias entre os municípios.

É ideal que não escolhamos um valor muito alto - escolher 60km, por exemplo, implicará que 5% de todas as relações entre municípios serão consideradas relações de vizinhança - o que pode ser muita coisa. Vamos tentar com o ponto de corte  $d = 40\text{km}$ :

```
viz_sp3 <- centroides_coordenadas_metros %>%
  dnearneigh(d1=0, d2=40000)

viz_sp3
```

```
#> Neighbour list object:
#> Number of regions: 645
#> Number of nonzero links: 8464
#> Percentage nonzero weights: 2.034493
#> Average number of links: 13.12248
```

O script cria uma nova lista de vizinhança espacial chamada `viz_sp3` baseada na distância entre centroides.

- o comando `centroides_coordenadas_metros`: contém as coordenadas X e Y dos centroides de cada município, em metros (no sistema UTM).
- `dnearneigh(d1=0, d2=40000)`: Cria a vizinhança espacial considerando que `d1 = 0` é a distância mínima entre vizinhos é zero metros (ou seja, qualquer distância positiva serve). E `d2 = 40000` é a distância máxima para serem considerados vizinhos é 40.000 metros (40 km).

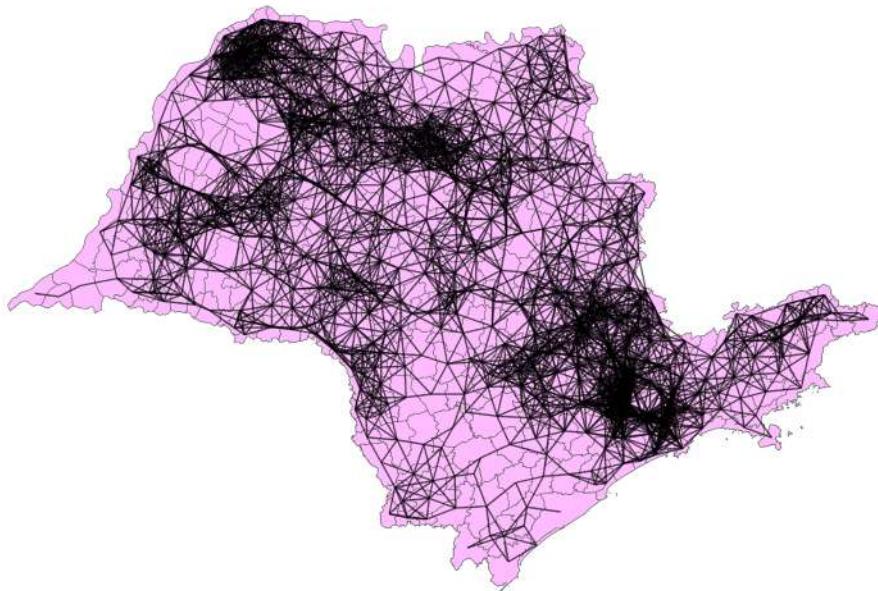
```

viz_linhas3 ← nb2lines(nb = viz_sp3,
                      coords = centrodoides_coordenadas) %>%
  st_as_sf()
st_crs(viz_linhas3) ← st_crs(malha_sp)

ggplot(malha_sp, aes(geometry=geom)) +
  geom_sf(fill = "plum1") +
  geom_sf(data = viz_linhas3, aes(geometry=geometry),
         alpha=0.4) +
  ggtitle("Matriz de vizinhança por distância (40km), estado de São Paulo.") +
  theme_void()

```

**Figura 39: Matriz de vizinhança por distância dos municípios do estado de São Paulo.**



O script primeiro cria linhas de vizinhança conectando os centroides dos municípios que são considerados vizinhos segundo o objeto `viz_sp3`. Essas linhas são convertidas para um formato espacial (`sf`) para que possam ser usadas em visualizações no `ggplot2`. Em seguida, o sistema de referência espacial dessas linhas (`viz_linhas3`) é ajustado para ser o mesmo da malha de municípios (`malha_sp`), garantindo que os dados estejam sobre a mesma base de coordenadas.

Depois, o script monta o mapa: ele desenha os polígonos dos municípios preenchidos com uma cor lilás clara, sobrepõe as linhas de vizinhança com transparência (para dar destaque ao mapa de fundo), adiciona um título explicativo e remove todos os elementos visuais desnecessários (como eixos, grades e bordas) para deixar o mapa mais limpo e focado na informação espacial.

Vemos na Figura 39 todos os municípios possuem vizinhos, mas algumas áreas são bem mais densas do que outras - municípios rurais tendem a ser maiores e mais espaçados entre uns e outros, enquanto os grandes centros parecem possuir municípios menores e mais próximos um dos outros. Isso se confirma quando vemos o número de vizinhos por município (Figura 40):

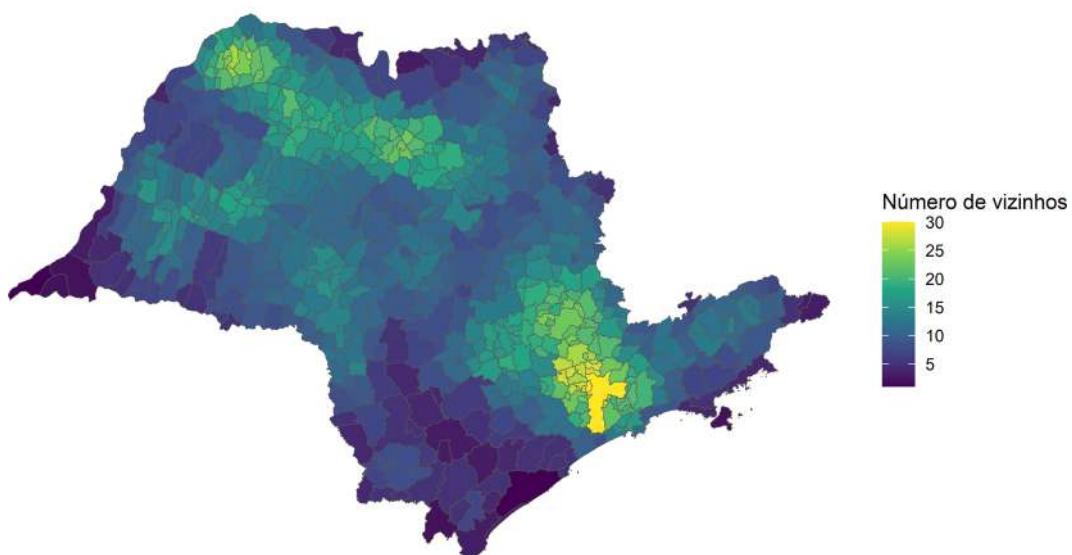
```
# extrair número de vizinhos de cada municipio

num_vizinhos3 <- viz_sp3 %>%
  lapply(length) %>%
  unlist()

malha_sp %>%
  mutate(num_vizinhos = num_vizinhos3) %>%
  ggplot(aes(geometry=geom)) +
  geom_sf(aes(fill=num_vizinhos)) +
  scale_fill_viridis_c(name="Número de vizinhos",
                        breaks = seq(0,30,by=5)) +
  ggtitle("Número de vizinhos de cada município, segundo a estratégia\nde vizinhança por distância (40km)") +
  theme_void()
```

**Figura 40: Número de vizinhos segundo matriz de vizinhança por distância de 40km, municípios do estado de São Paulo.**

Número de vizinhos de cada município, segundo a estratégia de vizinhança por distância (40km)



O script calcula quantos vizinhos cada município possui com base na lista de vizinhança `viz_sp3`. Primeiro, ele aplica a função `length` em cada elemento da lista para contar o número de vizinhos de cada município. Esses valores são então organizados em um vetor simples (`num_vizinhos3`).

Depois, o script usa a malha espacial dos municípios (`malha_sp`), adiciona uma nova coluna chamada `num_vizinhos` contendo o número de vizinhos de cada município, e cria um mapa. No mapa, os municípios são coloridos de acordo com a quantidade de vizinhos, utilizando a escala de cores contínuas `viridis`, que é perceptível para pessoas com daltonismo e facilita a interpretação. O título explica que o critério de vizinhança foi baseado em uma distância de até 40 km entre municípios, e o tema `theme_void()` é usado para deixar o mapa limpo, sem eixos nem grades.

Dada as diferenças de resultados obtidos, vemos que a escolha da definição da matriz de vizinhança é importante e pode impactar os resultados obtidos nas análises subsequentes. Por isso, é importante ter conhecimento do agravo em questão e entender quais fatores são determinantes para caracterização de uma relação entre um município e outro (divisão de fronteiras, fluxo migratório e/ou pendular, distância, entre outros). Os tópicos a seguir utilizam as matrizes de vizinhança definidas para aplicação de técnicas de análise espacial aos dados.

## *Testando a autocorrelação espacial global*

Durante o capítulo, comentamos sobre o efeito da correlação espacial - o pressuposto de que o espaço - ou seja, quais municípios estão próximos uns dos outros - influencia na determinação do agravo. Esse pressuposto é importante, visto que determinadas abordagens estatísticas - como modelos de regressão - possuem a independência como pressuposto, e caso assumamos independência entre os dados quando há correlação espacial entre eles podemos obter estimativas enviesadas. Mesmo quando o agravo não é diretamente ou intuitivamente relacionado ao espaço, outros fatores determinantes podem se distribuir espacialmente na população e gerar uma dependência espacial do processo de interesse. Por isso, é sempre importante verificar a distribuição dos dados de maneira exploratória e testar a existência dessa correlação.

Geralmente testamos para a **autocorrelação** espacial - ou seja, a correlação da variável com ela mesma ao longo do espaço - utilizando índices globais conhecidos, como o Índice I de Moran ou o Índice C de Geary.

## Índice I de Moran

Para aplicar estes testes, precisamos voltar para a matriz de vizinhança que definimos anteriormente - elas definirão “quem está próximo de quem” e impactarão diretamente em nossas estimativas.

A ideia de cálculo do índice é comparar a variância entre as áreas vizinhas com a variância que seria esperada em caso de aleatoriedade espacial (supondo que o espaço não importa no número de casos do agravo). Os resultados podem ser interpretados da seguinte forma:

- Índices positivos ( $I > 0$ ): quando  $I$  é significativamente maior do que 0, há evidências de que há uma **autocorrelação espacial positiva**, ou seja: áreas próximas umas das outras **tendem a ter taxas parecidas**. Geralmente é o caso quando há aglomerados de doença - se um município possui uma taxa alta, é muito provável que seus vizinhos também tenham; quando há baixa atividade da doença, espera-se que os vizinhos também apresentem situação semelhante.
- Índices negativos ( $I < 0$ ): há uma **autocorrelação espacial negativa** ou inversa, o que indica que áreas próximas tendem a estar em situações distintas. A definição soa de forma contraintuitiva, mas pode indicar situações em que há outliers presentes: municípios com altas taxas rodeados de municípios com atividade baixa da doença.
- Índice próximo ou igual a zero ( $I \approx 0$ ): indica ausência de autocorrelação espacial. Os dados estão distribuídos aleatoriamente no espaço, ou seja: um município estar próximo de outro não gera influência no valor do número de casos observado em cada um dos municípios.

No exemplo de casos de dengue em 2020 em São Paulo, há indícios de que existe uma autocorrelação espacial na Figura 32 - visto que há bolsões com maior incidência e concentração da doença. Vamos, portanto, testar a hipótese de autocorrelação espacial.

Para isso, precisamos obter uma **matriz de pesos** a partir da matriz de vizinhança definida. A partir da relação entre os vizinhos, escolhe-se uma estratégia para ponderar cada uma das ligações (relações de vizinhança) identificadas. É comum a estratégia de ponderar de forma padronizada, de forma que a soma de todos os pesos dos vizinhos de um município  $m$  seja igual a 1. Ou seja, se o município  $m$  possui dois vizinhos, cada aresta de ligação terá peso 0,5, de forma que ao somar os pesos temos  $0,5 + 0,5 = 1$ . Outras estratégias existem e podem ser consultadas com o comando `?nb2listw`.

```
# style = "W" corresponde a abordagem informada de somar as arestas e obter 1
pesos_viz ← viz_sp %>% nb2listw(style = "W")
```

O script cria uma estrutura de pesos espaciais a partir da lista de vizinhança `viz_sp`, usando o estilo “W” (row-standardized).

A função `nb2Listw()` transforma o objeto de vizinhança em um objeto de pesos (`listw`), onde:

- Para cada região, os pesos dos vizinhos são normalizados de forma que a soma dos pesos de cada linha (ou seja, de cada região) seja igual a 1.
- O argumento `style = "W"` especifica essa normalização, chamada de “row-standardized weights”.

E agora com os pesos definidos, podemos rodar o teste:

```
moran.test(
  dengue_2020$inc,
  pesos_viz
)
```

```
#>
#> Moran I test under randomisation
#>
#> data: dengue_2020$inc
#> weights: pesos_viz
#>
#> Moran I statistic standard deviate = 18.034, p-value < 2.2e-16
#> alternative hypothesis: greater
#> sample estimates:
#> Moran I statistic      Expectation      Variance
#>       0.4408150828    -0.0015527950     0.0006017108
```

O script realiza o teste de Moran para verificar a presença de autocorrelação espacial na incidência bruta de dengue em 2015.

A função `moran.test()` recebe:

- `dengue_2020$inc`: os valores da variável de interesse (neste caso, a incidência de dengue).
- `pesos_viz`: a matriz de pesos espaciais, construída anteriormente com a vizinhança padronizada ("W").

O teste de Moran avalia se municípios com incidência semelhante estão espacialmente agrupados. A função `moran.test()` calcula:

- O valor do índice de Moran ( $I$ ), que mede a autocorrelação espacial da variável `dengue_2020$inc` (incidência bruta de dengue).
- A expectativa do índice sob a hipótese de aleatoriedade espacial (sem padrão).
- O p-valor associado, que testa se a autocorrelação observada é estatisticamente significativa.

Outros detalhes, como variância e método de cálculo.

Observa-se que o Índice  $I$  estimado é de 18,03, e o p-valor é menor do que 0,05, indicando a existência de uma **autocorrelação espacial positiva**. Ou seja, o espaço é um fator importante para a explicação da distribuição da dengue em São Paulo no ano de 2020, de forma que municípios próximos tendem a ter situações semelhantes.



## Índice C de Geary

Podemos calcular também o Índice C de Geary, que possui um intuito semelhante, mas se baseia nas diferenças absolutas de valores entre o município e seus vizinhos. Seu valor de neutralidade é o valor 1, com valores de  $C < 1$  indicando autocorrelação positiva e  $C > 1$  autocorrelação negativa:

```
geary.test(  
  dengue_2020$inc,  
  pesos_viz  
)
```

```
#>  
#> Geary C test under randomisation  
#>  
#> data: dengue_2020$inc  
#> weights: pesos_viz  
#>  
#> Geary C statistic standard deviate = 13.006, p-value < 2.2e-16  
#> alternative hypothesis: Expectation greater than statistic  
#> sample estimates:  
#> Geary C statistic      Expectation      Variance  
#>          0.544092361    1.0000000000    0.001228851
```

O script realiza o teste de Geary para avaliar a autocorrelação espacial da incidência de dengue em 2015.

A função `geary.test()` recebe:

- `dengue_2020$inc`: a variável de interesse (incidência bruta de dengue).
- `pesos_viz`: a matriz de pesos espaciais construída anteriormente.

Ao observar a direção do índice estimado (Geary C statistic  $< 1$ ) e um p-valor muito baixo, chegamos à mesma conclusão que obtivemos com o teste de Moran.

Ou seja, ambos os índices apontam para uma existência de autocorrelação espacial global nos dados, no sentido positivo. Podemos, contudo, estar interessados em mais: saber onde essa correlação é mais ou menos forte, e se ela muda de sentido ao longo do estado. Para isso, utilizaremos índices **locais** de associação espacial.

## *Testando a autocorrelação espacial local*

Além de um indicador que estima a existência ou ausência de autocorrelação espacial do processo no conjunto de dados como um todo, pode ser interessante averiguar a existência de fenômenos que ocorrem localmente. Mesmo com ausência de autocorrelação espacial global, podemos observar algumas manifestações locais de autocorrelação, que foram mascaradas ao olhar para a região como um todo. Esses indicadores locais permitem uma exploração mais detalhada das áreas envolvidas em possíveis aglomerados da doença (autocorrelação local positiva), ou a identificação de possíveis outliers (autocorrelação local negativa).

## *Indicadores Locais de Associação Espacial*

O índice de Moran Local (que é um *LISA - Local Indicators of Spacial Association*) é uma adaptação do índice I de Moran para um contexto local, calculado para cada área . Possui interpretação semelhante, mas voltada para cada área, medindo seu grau de associação com suas áreas vizinhas.

```
lisa_dengue2020 ← localmoran_perm(dengue_2020$inc, pesos_viz)

head(lisa_dengue2020)
```

```
#>           Ii      E.Ii    Var.Ii      Z.Ii Pr(z ≠ E(Ii))
#> 1  1.40116921  0.010362547 0.20960579  3.03784158  0.002382792
#> 2 -0.24463453 -0.010723014 0.04967033 -1.04954983  0.293925134
#> 3  0.47277760  0.006618033 1.05671851  0.45347667  0.650205542
#> 4  0.02506444  0.003231677 0.10332043  0.06792281  0.945847082
#> 5  0.36274715 -0.005694462 0.12166351  1.05630298  0.290829815
#> 6  0.37898181 -0.007048969 0.07344926  1.42438773  0.154334264
#>   Pr(z ≠ E(Ii)) Sim Pr(folded) Sim Skewness Kurtosis
#> 1                 0.024          0.012  1.0589117 0.5245390
#> 2                 0.300          0.150 -1.0392633 1.2729922
#> 3                 0.604          0.302  0.8987260 0.7709809
#> 4                 0.744          0.372 -1.7630536 3.2617769
#> 5                 0.112          0.056 -1.1696210 0.9156516
#> 6                 0.008          0.004 -0.8935898 0.6438619
```

O script calcula o índice de autocorrelação espacial local (LISA) para a incidência de dengue em 2020 usando permutação aleatória.

A função `localmoran_perm()`:

- Recebe `dengue_2020$inc` (incidência bruta) e `pesos_viz` (estrutura de pesos espaciais).
- Calcula o índice LISA para cada município, que mede a autocorrelação local — ou seja, identifica clusters de valores altos (hotspots), valores baixos (coldspots) e áreas de transição.
- Utiliza permutações para gerar uma distribuição nula e calcular p-valores para cada município.

O objeto `lisa_dengue2020` armazena:

- O valor do índice local de Moran para cada área.
- Estatísticas associadas como p-valor, valores esperados sob aleatoriedade, e variância.

O comando `head(lisa_dengue2020)` exibe as primeiras linhas, mostrando os resultados dos primeiros municípios analisados.

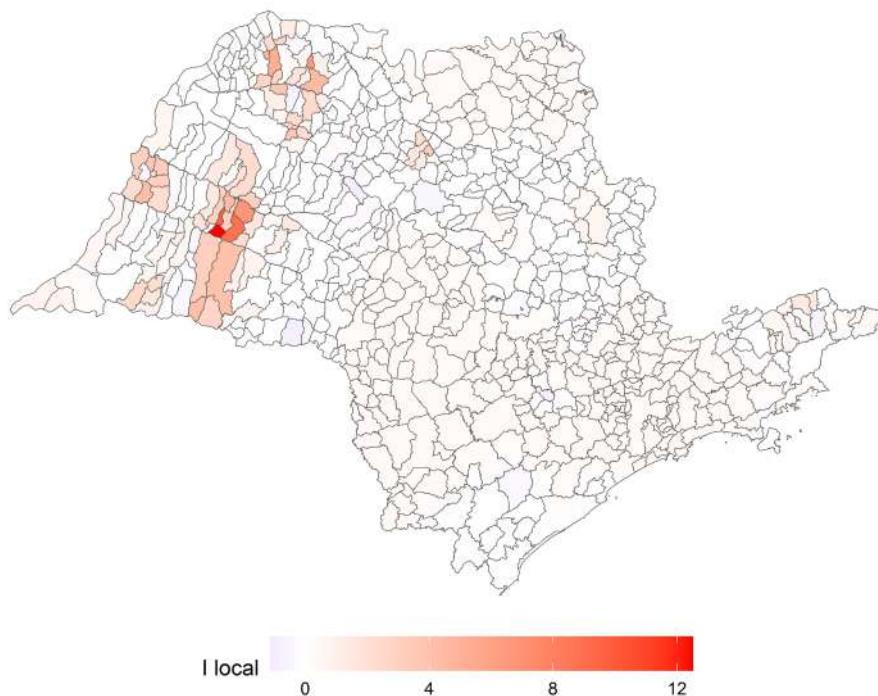
Agora vemos, para cada município, uma estimativa de  $I$  ( $I_{ii}$ ) e seu p-valor  $Pr(z \neq E(I_{ii}))$ . Podemos representar esse resultado visualmente na Figura 41:

```
malha_sp_dengue$local_I ← lisa_dengue2020[, 1]
malha_sp_dengue$local_I_p_valor ← lisa_dengue2020[, 5]

ggplot(malha_sp_dengue, aes(geometry=geom)) +
  geom_sf(aes(fill = local_I)) +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,
                       name = "I local") +
  theme_void() +
  theme(legend.position = "bottom",
        legend.key.width = unit(1.5, "cm"))
```



**Figura 41: Índice de Moran Local da incidência de dengue nos municípios de São Paulo, 2020.**



O script associa os resultados do LISA (Índice de Moran Local) à malha espacial dos municípios de São Paulo.

Primeiramente:

- A primeira coluna de `lisa_dengue2020` (valores do índice local) é atribuída à nova variável `local_I` em `malha_sp_dengue`.
- A quinta coluna de `lisa_dengue2020` (p-valor associado) é atribuída à nova variável `local_I_p_valor`.

Em seguida, o script cria um mapa utilizando `ggplot2`:

- Cada município é preenchido com uma cor baseada no valor do índice local (`local_I`).
- Utiliza `scale_fill_gradient2()`, que aplica uma escala de cores: azul para valores negativos, vermelho para valores positivos e branco para valores próximos de zero.

O mapa resultante destaca visualmente áreas com alta ou baixa autocorrelação local na incidência de dengue em 2015.

Vemos que boa parte do estado concentrou valores de I local próximos de zero. Alguns municípios, no entanto, se destacaram por possuírem valores de I bem maiores que zero, o que significa uma forte autocorrelação espacial local.

Uma forma comum de visualizar os resultados de um LISA é através do chamado LISA map. Nele, utilizamos o p-valor obtido para classificar os municípios com autocorrelação espacial local significativa ( $p < 0,5$ ) ou não significativa ( $p \geq 0,05$ ). Para os significativos, vemos o sinal da associação: se é direta ( $I_{local} > 0$ ) ou inversa ( $I_{local} < 0$ ). Por fim, olhamos para a magnitude da variável: se há uma associação espacial local positiva e a taxa no município é alta (acima da média), então há um cluster **Alto-alto**, pois é uma região onde altos valores costumam se concentrar. Se a taxa é baixa, contudo, trata-se de um cluster **Baixo-baixo**, pois há uma concentração de municípios com taxas abaixo do esperado. Quando a associação é significativa e negativa, então há os cenários **Alto-baixo** e **Baixo-alto**, que ocorre quando o município é um *outlier* entre seus vizinhos - se ele está com a taxa alta, seus vizinhos tendem a estar baixo e vice-versa.

No resultado do comando, esse agrupamento já é realizado e armazenado no atributo `quadr.`

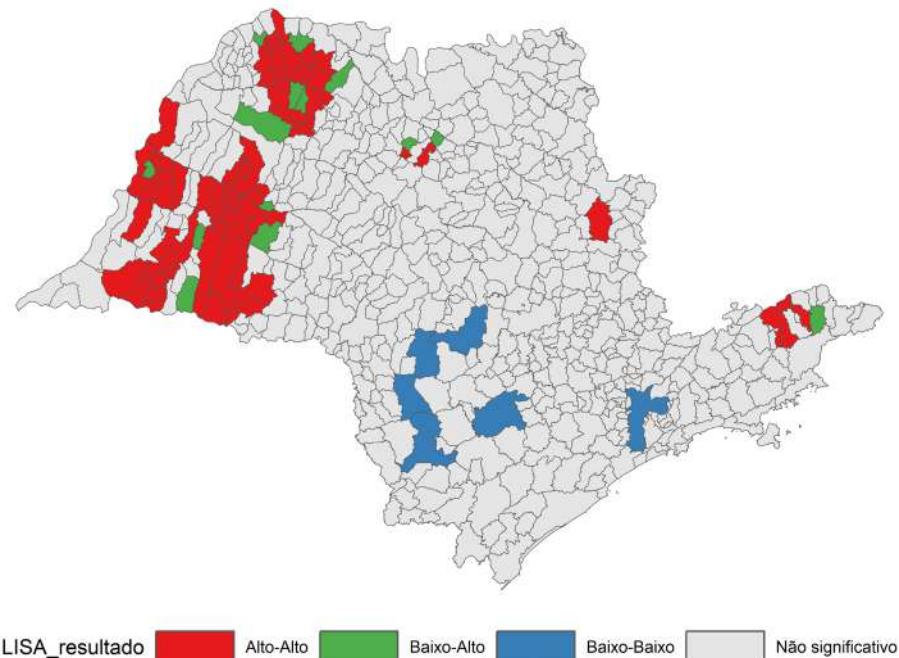
```
quadrantes ← attr(lisa_dengue2020, "quadr")$mean

malha_sp_dengue$quadrante ← case_when(
  quadrantes == "High-High" ~ "Alto-Alto",
  quadrantes == "Low-Low" ~ "Baixo-Baixo",
  quadrantes == "High-Low" ~ "Alto-Baixo",
  quadrantes == "Low-High" ~ "Baixo-Alto"
)

malha_sp_dengue ← malha_sp_dengue %>%
  mutate(LISA_resultado = case_when(
    local_I_p_valor < 0.05 ~ quadrante,
    local_I_p_valor ≥ 0.05 ~ "Não significativo"
  ))

ggplot(malha_sp_dengue, aes(geometry=geom)) +
  geom_sf(aes(fill = LISA_resultado)) +
  scale_fill_manual(
    values = c("#e41a1c", "#4daf4a", "#377eb8", "gray90")
  ) +
  theme_void() +
  theme(legend.position = "bottom",
        legend.key.width = unit(1.5, "cm"))
```

**Figura 42: LISA Map da incidência de dengue nos municípios de São Paulo, 2020.**



LISA\_resultado    Alto-Alto    Baixo-Alto    Baixo-Baixo    Não significativo

O script organiza os resultados do teste LISA em quadrantes e gera um mapa temático para interpretar os padrões espaciais da incidência de dengue em 2020.

Primeiramente:

- A informação sobre o tipo de associação espacial (quadrante) é extraída dos atributos do objeto `lisa_dengue2020`, identificando se cada município pertence a “High-High”, “Low-Low”, “High-Low” ou “Low-High”.
- Cria-se uma nova variável `quadrante` em `malha_sp_dengue`, traduzindo esses nomes para português.

Em seguida:

- Uma nova variável `LISA_resultado` é criada:
  - Municípios com p-valor menor que 0,05 são classificados no quadrante correspondente (“Alto-Alto”, “Baixo-Baixo”, etc.).
  - Municípios com p-valor maior ou igual a 0,05 são classificados como “Não significativo”.

O mapa final é construído com `ggplot2`:

- Cada município é preenchido com uma cor de acordo com o seu resultado no LISA:
  - “Alto-Alto” (vermelho) indica áreas de alta incidência cercadas por outras áreas de alta incidência (hotspots).
  - “Baixo-Baixo” (azul) indica áreas de baixa incidência cercadas por outras de baixa incidência (coldspots).
  - “Alto-Baixo” (verde) e “Baixo-Alto” (verde) indicam áreas de potencial outlier espacial.
  - “Não significativo” (cinza) indica ausência de autocorrelação estatisticamente significativa.
- O layout do mapa é limpo com `theme_void()`, e a legenda é posicionada abaixo do mapa para melhor organização visual.

Este mapa LISA permite identificar padrões locais de agregação e outliers espaciais na distribuição da dengue.

Com o resultado, temos os agrupamentos dos municípios de acordo com o resultado do LISA map (Figura 42). Vê-se que a maioria dos municípios não apresentou significância estatística; contudo, alguns aglomerados foram identificados - ressalta-se um *cluster* de baixa incidência no Sul do estado (grupo “Baixo-Baixo”, em azul), e alguns pontos de **clusters** de alta incidência (grupo “Alto-Alto”, em vermelho), em regiões já identificadas como destoantes nas visualizações anteriores. O LISA map aponta, inclusive, alguns municípios *outliers* - que estão com valores de baixa incidência, mas rodeados de municípios com taxas mais altas (grupo “Baixo-Alto”, em verde).

## *Suavização espacial*

Ao representar dados espacialmente, frequentemente convém recorrer a técnicas que facilitam a visualização de padrões espaciais. Isso porque a visualização “crua” dos dados pode conter artefatos que dificultam a percepção e entendimentos dos padrões presentes. Como mencionamos anteriormente, municípios com populações pequenas podem ter “taxas infladas” - caso em que as taxas de incidência de uma doença nesses municípios são muito maiores que os demais devido ao baixo denominador (população) - que faz com que poucos casos (muitas vezes, 2 ou 5 casos) gerem uma taxa muito alta de casos por habitante.

As técnicas de suavização utilizam das relações de vizinhança e proximidade para evidenciar padrões e tendências espaciais, facilitando a visualização e entendimento dos dados. Veremos a seguir duas técnicas diferentes: a suavização por Kernel de Área e pelo Estimador Bayesiano Empírico.



## Kernel de área

Como vimos no Módulo 2, podemos interpolar uma superfície contínua de densidade a partir de pontos que representam casos de uma doença. Nesse caso, não temos a localização pontual dos casos - apenas a contagem de casos por municípios, ou outra unidade de área. A aplicação da técnica consiste, portanto, na representação dessas áreas como pontos (geralmente se considera o centroide da área ou alguma outra localização de referência, como a sede do município) - e na **ponderação** desses pontos com a contagem agregada de casos para cada região.

Para essa transformação de dados de área para pontos, vamos utilizar novamente os centroides dos municípios (Figura 43). Além disso, vamos utilizar o pacote `patchwork` para organizar as imagens de uma forma mais simples. Caso não tenha instalado, siga os comandos abaixo:

```
# se não estiver instalado, rodar:  
install.packages("patchwork")  
library(patchwork)
```

```
malha_sp_pontos ← malha_sp %>%  
st_centroid()  
  
g_malha_sp ← malha_sp %>%  
ggplot(aes(geometry=geom)) +  
geom_sf(fill = "darkolivegreen1") +  
ggtitle("Malha dos municípios de São Paulo") +  
theme_void()  
  
g_pontos_sp ← malha_sp_pontos %>%  
ggplot(aes(geometry=geom)) +  
geom_sf(color = "darkgreen") +  
ggtitle("Municípios de São Paulo, transformados\nem pontos (centroides)") +  
theme_void()  
  
g_malha_sp | g_pontos_sp
```

**Figura 43: Divisão territorial e centroides dos municípios do estado de São Paulo.**

Malha dos municípios de São Paulo



Municípios de São Paulo, transformados  
em pontos (centroides)



O script começa calculando o centroide de cada município da malha `malha_sp`, ou seja, o ponto central de cada polígono, utilizando a função `st_centroid()` e criando o objeto `malha_sp_pontos`.

Em seguida, cria dois gráficos separados com o `ggplot2`:

- O primeiro (`g_malha_sp`) desenha a malha completa dos municípios colorida de verde claro e sem elementos de fundo, apenas os limites dos municípios.
- O segundo (`g_pontos_sp`) desenha somente os centroides (pontos centrais dos municípios), com os pontos plotados em preto, também sobre um fundo limpo.

Por fim, usando o pacote `patchwork`, os dois gráficos são colocados lado a lado para comparação: de um lado aparece o mapa tradicional com os limites dos municípios e, do outro, o mapa reduzido a pontos centrais.

Agora, temos nossos pontos para aplicação do método Kernel. Utilizaremos o **Kernel por atributo** - onde cada ponto (cada município) terá um valor que corresponde ao que desejamos visualizar, neste caso, a taxa de incidência de uma doença.

Para isso, precisamos criar um objeto da classe `ppp` (point pattern).

```
library(spatstat)
library(nngeo)

## Contorno do estado

contorno_sp ← malha_sp %>%
  st_transform(crs = 31982) %>%
  st_union() %>%
  st_remove_holes()

# Transformando o contorno para janela (owin)
sp_owin ← contorno_sp %>% as.owin()

sp_ppp ← ppp(
  centroides_coordenadas_metros[,1], # Coordenada em X
  centroides_coordenadas_metros[,2], # Coordenada em Y
  sp_owin # Janela
)

plot(sp_ppp, pch = 19, cex = 0.5, main = "Objeto PPP dos municípios")
```

**Figura 44: Visualização do objeto de classe `ppp` contendo os centroides dos municípios e o contorno do estado de São Paulo.**

### Objeto PPP dos municípios



O script começa carregando as bibliotecas necessárias para manipulação espacial.

Em seguida, ele cria o contorno do estado de São Paulo: para isso, transforma a malha de municípios para o sistema de coordenadas UTM (metros) através da função `st_transform()`, une todos os municípios em um único polígono (`st_union()`) e remove eventuais buracos internos (`st_remove_holes()`). O contorno gerado servirá para definir a área de referência para análises espaciais futuras, como a estimativa de densidades por kernel.

O script primeiro converte o contorno do estado de São Paulo, que estava em formato `sf`, para o formato `owin`, usado pelo pacote `spatstat` para representar áreas de estudo em análises de padrões pontuais.

Em seguida, cria um objeto do tipo `ppp`, que é a estrutura de dados do `spatstat` para conjuntos de pontos no espaço. Para isso, utiliza as coordenadas X e Y dos centroides dos municípios e define o contorno do estado como a janela espacial (o limite de estudo).

Por fim, o script plota esse objeto `ppp`, exibindo os municípios como pequenos pontos pretos dentro do contorno do estado de São Paulo.

Agora temos um objeto do tipo `ppp` - *point pattern*, tal qual utilizamos no Módulo 2. Agora, vamos visualizar de forma suavizada no espaço os casos de dengue em 2015 no estado de São Paulo. Podemos voltar a utilizar a função `density()`, para estimação do kernel, mas teremos que nos atentar a dois parâmetros:

- **`sigma`**: Largura de banda - indicará a ordem de suavização aplicada aos dados (maior ou menor).
- **`weights`**: o parâmetro utilizado para o kernel de atributo. Aqui é importante informarmos o atributo que estamos interessados em suavizar, como por exemplo, a incidência de dengue.
- **`eps`**: indicará a resolução (qualidade gráfica) do resultado final. Quanto menor, maior resolução.

Vamos testar as seguintes larguras de banda: `5000`, `10000`, `20000` e `50000`.

```
kernel_den_2015_b5000 ← density(  
  sp_ppp,  
  5000,  
  weights = dengue_2015$inc,  
  scalekernel = T,  
  diggle = T,  
  eps=2000  
)  
  
kernel_den_2015_b10000 ← density(  
  sp_ppp,  
  10000,  
  weights = dengue_2015$inc,  
  scalekernel = T,  
  diggle = T,  
  eps=2000  
)  
  
kernel_den_2015_b20000 ← density(  
  sp_ppp,  
  20000,  
  weights = dengue_2015$inc,  
  scalekernel = T,  
  diggle = T,  
  eps=2000  
)  
  
kernel_den_2015_b50000 ← density(  
  sp_ppp,  
  50000,  
  weights = dengue_2015$inc,  
  scalekernel = T,  
  diggle = T,  
  eps=2000  
)
```

O script realiza a estimativa de densidades por Kernel sobre o objeto de pontos `sp_ppp`, utilizando diferentes larguras de banda (5000, 10000, 20000 e 50000 metros).

Em cada estimativa:

- A função `density()` do pacote `spatstat` é utilizada para calcular a densidade espacial.
- A largura de banda (`sigma`) define o grau de suavização: valores menores capturam detalhes locais, enquanto valores maiores suavizam mais amplamente.
- O argumento `weights = dengue_2015$inc` pondera os pontos pelo número de casos de dengue em 2015 (`inc`), ou seja, cada ponto influencia a densidade conforme a intensidade de dengue.
- `scalekernel = TRUE` ajusta o kernel para garantir que o total de massa (peso) se mantenha consistente.
- `diggle = TRUE` aplica uma correção para bordas, seguindo o método de Diggle, melhorando a estimativa nas áreas próximas às extremidades da janela de estudo.
- `eps = 2000` define a resolução da grade para o cálculo, com espaçamento de 2000 metros entre os pontos de avaliação.

Como resultado, são gerados quatro mapas de densidade (`kernel_den_2015_b5000`, `kernel_den_2015_b10000`, `kernel_den_2015_b20000`, `kernel_den_2015_b50000`), cada um com um nível diferente de suavização espacial.

Agora, vamos gerar gráficos para compará-los lado a lado. Podemos transformar o resultado em uma `tibble()`:

```
kernel_den_2015_b5000 ← kernel_den_2015_b5000 %>%  
  as_tibble()  
head(kernel_den_2015_b5000)
```

```
#> # A tibble: 6 × 3
#>       x     y   value
#>   <dbl> <dbl>   <dbl>
#> 1 284117. 7498067. 6.26e-14
#> 2 284117. 7500065. 1.77e-13
#> 3 286114. 7494070. 3.57e-14
#> 4 286114. 7496068. 1.39e-13
#> 5 286114. 7498067. 4.60e-13
#> 6 286114. 7500065. 1.30e-12
```

O script transforma o objeto de densidade `kernel_den_2015_b5000`, originalmente em formato de grade espacial (`im` do pacote `spatstat`), em um `tibble`, que é uma estrutura de tabela mais moderna e amigável para manipulação no R.

Ao aplicar `as_tibble()`, a matriz de valores da densidade é convertida em uma tabela onde cada linha representa uma célula da grade espacial, contendo as informações de coordenadas e valor estimado da densidade.

O comando `head(kernel_den_2015_b5000)` exibe as primeiras linhas dessa tabela, permitindo visualizar como os dados de densidade foram organizados após a conversão.

Nessa `tibble`, vemos a coordenada estimada em X, Y e o valor de densidade estimado pelo kernel. Lembramos que o kernel sempre estima a **densidade** do evento em cada localização.

```

g_kernel_5000 ← kernel_den_2015_b5000 %>%
  ggplot() +
  geom_tile(aes(x=x, y=y, fill=value)) +
  scale_fill_viridis_c(option = "H") +
  labs(title = "sigma: 5000", fill = "densidade") +
  coord_fixed() + theme_void()

kernel_den_2015_b10000 ← kernel_den_2015_b10000 %>%
  as_tibble()

g_kernel_10000 ← kernel_den_2015_b10000 %>%
  ggplot() +
  geom_tile(aes(x=x, y=y, fill=value)) +
  scale_fill_viridis_c(option = "H") +
  labs(title = "sigma: 10000", fill = "densidade") +
  coord_fixed() + theme_void()

kernel_den_2015_b20000 ← kernel_den_2015_b20000 %>%
  as_tibble()

g_kernel_20000 ← kernel_den_2015_b20000 %>%
  ggplot() +
  geom_tile(aes(x=x, y=y, fill=value)) +
  scale_fill_viridis_c(option = "H") +
  labs(title = "sigma: 20000", fill = "densidade") +
  coord_fixed() + theme_void()

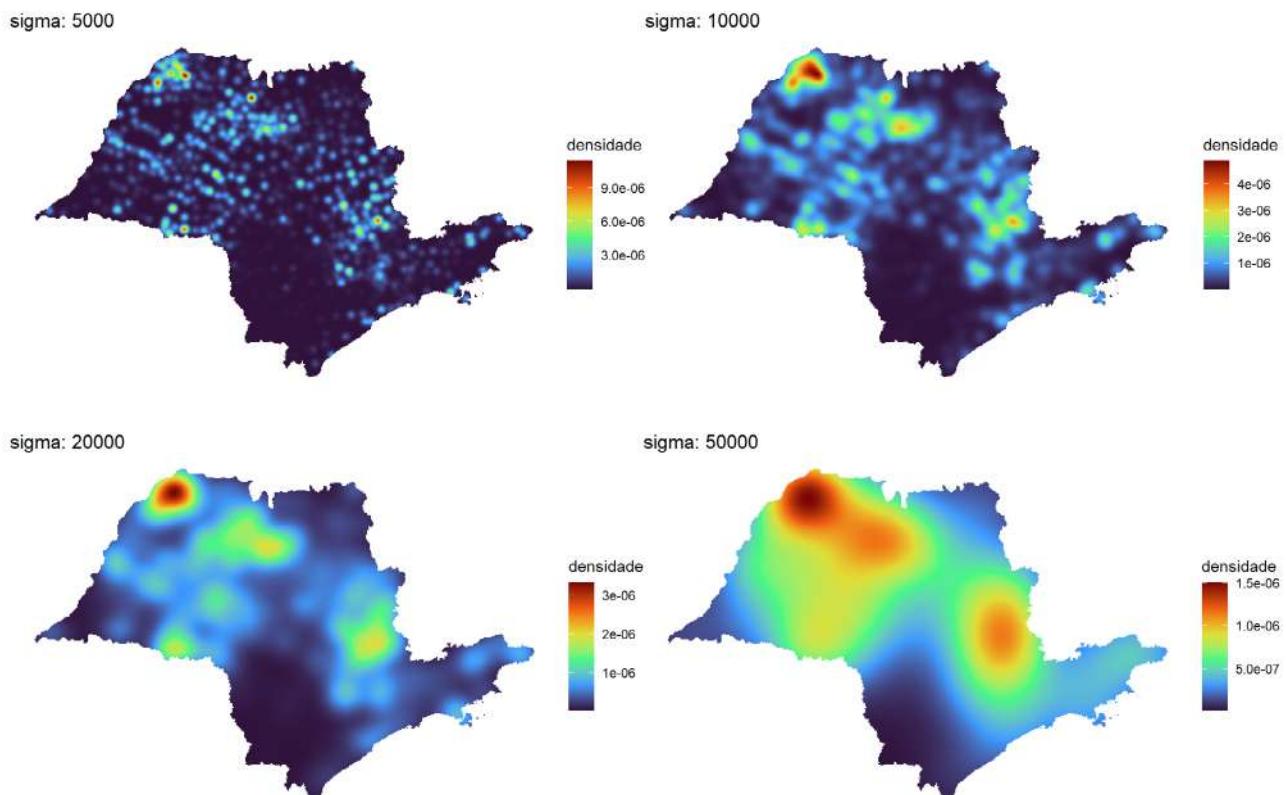
kernel_den_2015_b50000 ← kernel_den_2015_b50000 %>%
  as_tibble()

g_kernel_50000 ← kernel_den_2015_b50000 %>%
  ggplot() +
  geom_tile(aes(x=x, y=y, fill=value)) +
  scale_fill_viridis_c(option = "H") +
  labs(title = "sigma: 50000", fill = "densidade") +
  coord_fixed() + theme_void()

(g_kernel_5000 | g_kernel_10000) /
  (g_kernel_20000 | g_kernel_50000)

```

**Figura 45: Estimativa de densidade de kernel da incidência de dengue por município no estado de São Paulo, 2015.**



O script gera mapas de densidade kernel para quatro larguras de banda diferentes (5000, 10000, 20000 e 50000 metros), convertendo as densidades previamente calculadas para tibbles e visualizando-as com o `ggplot2`.

Para cada largura de banda:

- O objeto de densidade (`kernel_den_2015_b...`) é transformado em tibble usando `as_tibble()`, organizando as informações de posição (`x`, `y`) e valor da densidade (`value`).
- Um gráfico (`g_kernel_...`) é criado usando `ggplot()`, com `geom_tile()` para desenhar cada célula da grade como um pequeno quadrado colorido proporcional ao valor da densidade.
- A escala de cores é definida com `scale_fill_viridis_c(option = "H")`, que usa um gradiente de cores perceptível e amigável para daltônicos.
- Títulos e rótulos são adicionados com `labs()`, informando o valor de `sigma` utilizado na suavização.
- `coord_fixed()` mantém a proporção entre os eixos, e `theme_void()` remove elementos como eixos e grades para deixar o mapa mais limpo.

Por fim, utilizando o operador de composição de gráficos do patchwork, os quatro mapas (`g_kernel_5000`, `g_kernel_10000`, `g_kernel_20000` e `g_kernel_50000`) são organizados em uma grade 2x2 para facilitar a comparação visual dos diferentes níveis de suavização.

Vemos na Figura 45 que a utilização de um `sigma` baixo, como `sigma = 5000`, gerou uma visualização muito concentrada ainda no centroide dos municípios, suavizando muito pouco e dificultando a percepção de padrões espaciais. Com valores de largura de banda maiores, como `sigma = 10000` ou `sigma = 20000` ou vemos uma suavização interessante da incidência, permitindo ver regiões de maior concentração da doença naquele ano. Vemos o destaque principalmente na região Noroeste de São Paulo, marcada em tons vermelho na escala de cores. O `sigma = 50000` já realizou uma suavização demasiadamente grosseira, permitindo ver a distribuição em termos macro apenas.

Perceba também, que a escala de densidade muda em cada gráfico: com larguras de banda menores, os tons vermelhos representam uma densidade muito alta ( $9e10^{-6} = 0,000009$ ), e na maior largura de banda os tons vermelhos representam uma densidade menor ( $1.5e10^{-6} = 0,000015$ , seis vezes menor).

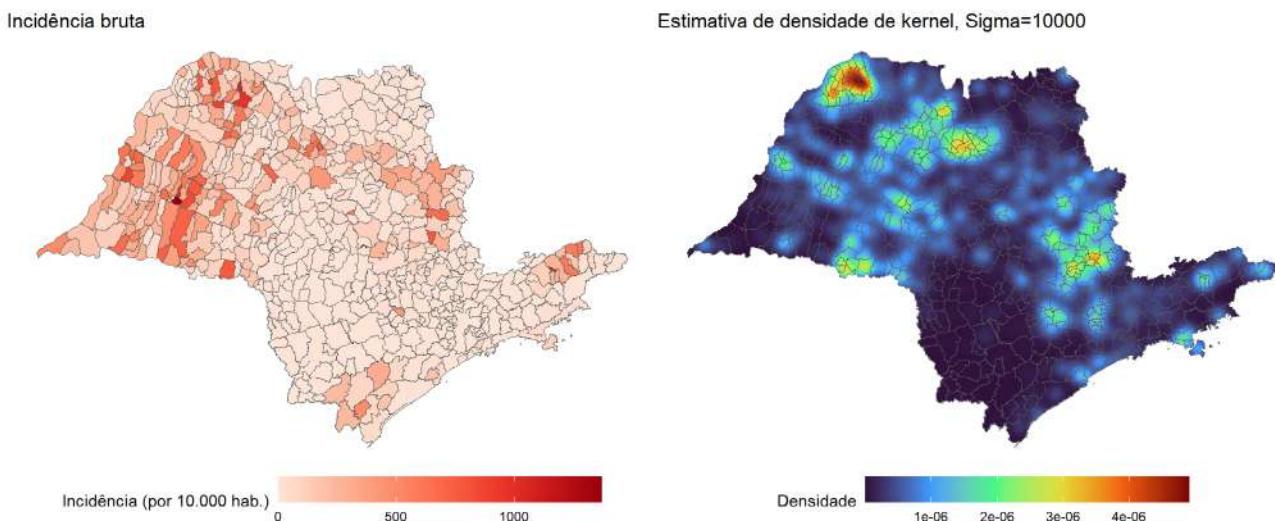
Vamos comparar a visualização que tínhamos anteriormente da incidência bruta com a visualização suavizada por kernel na Figura 46:

```
# gráfico feito anteriormente
g_tradicional <- ggplot(malha_sp_dengue, aes(geometry=geom)) +
  geom_sf(aes(fill = inc)) +
  labs(title = "Incidência bruta", fill = "Incidência (por 10.000 hab.)") +
  scale_fill_distiller(palette = "Reds", direction=1) +
  theme_void() +
  theme(legend.position = "bottom",
        legend.key.width = unit(1.5, "cm"))

g_kernel <- malha_sp %>%
  st_transform(crs = 31982) %>%
  ggplot() +
  geom_tile(
    data = kernel_den_2015_b10000,
    aes(x=x, y=y, fill = value)
  ) +
  geom_sf(aes(geometry=geom), fill="transparent") +
  scale_fill_viridis_c(option = "H") +
  labs(title = "Estimativa de densidade de kernel, Sigma=10000", fill =
  "Densidade") +
  theme_void() +
  theme(legend.position = "bottom",
        legend.key.width = unit(1.5, "cm"))

g_tradicional | g_kernel
```

**Figura 46: Comparação da visualização dos valores brutos e da estimativa de densidade de kernel da incidência de dengue nos municípios do estado de São Paulo, 2015.**



O script cria dois gráficos diferentes para comparar a representação da dengue nos municípios de São Paulo em 2015 em relação a incidência bruta e a estimativa de densidade de kernel.

O primeiro gráfico (`g_tradicional`) utiliza a malha de municípios (`malha_sp_dengue`) e representa a incidência bruta de cada município usando preenchimento de cor (`fill`) proporcional ao valor de incidência. O gradiente de cores escolhido é a paleta “Reds”, indicando maior incidência com tons de vermelho mais intensos. A legenda é posicionada na parte inferior e o layout é limpo utilizando `theme_void()`.

O segundo gráfico (`g_kernel`) utiliza a densidade suavizada gerada pelo método de kernel (`kernel_den_2015_b10000`). Primeiro, a malha de municípios é transformada para o sistema de coordenadas UTM (metros) usando a função `st_transform()`. Em seguida, o mapa de densidade é desenhado como uma grade de cores (`geom_tile()`), e sobre ele é sobreposta a borda dos municípios (`geom_sf()`) com preenchimento transparente, apenas delineando os limites. A escala de cores é baseada na paleta “viridis” e o sigma utilizado na suavização foi 10000 metros.

Finalmente, utilizando o operador do pacote `patchwork`, os dois gráficos (`g_tradicional` e `g_kernel`) são exibidos lado a lado para permitir uma comparação visual entre a representação tradicional por município e a representação contínua por densidade espacial.

Vemos que, por meio do Kernel por atributo, conseguimos obter uma superfície contínua de densidade de incidência, sobre a qual podemos capturar possíveis padrões espaciais existentes nos dados. Na Figura 46, ao olhar a distribuição da densidade de incidência de dengue em 2015 no estado de São Paulo, notamos a concentração de valores no tom vermelho mais escuro (e mais alto) da escala no Noroeste do estado, além de outros dois ou três aglomerados que atingem os tons amarelados e chamam atenção. Contudo, ao olharmos um dado suavizado por Kernel, olhamos para **densidade** - o que dificulta a interpretação dos resultados obtidos em termos de casos ou incidência. Veremos a seguir um método de suavização que mantém a mesma unidade da variável de interesse, ou seja, conseguimos suavizar os dados e interpretá-los em termos de incidência.

## Método Bayesiano Empírico

Ainda com o intuito de suavização de taxas no mapeamento de doenças, o Método Bayesiano Empírico introduz uma média das taxas que é ponderada de acordo com o tamanho da população do município. Essa **média** pode ser em relação ao valor **global** (taxa no estado como um todo, por exemplo) ou local (taxa média nos municípios vizinhos). O método **local** costuma ser mais interessante no mapeamento de doenças ao fornecer uma suavização mais realista - com base no contexto específico do município e seus vizinhos aos invés de uma média generalizada. A fórmula do método é a seguinte:

$$\hat{\theta}_i = w_i r_i + (1 - w_i) \mu_i$$

Perceba que se trata de uma operação de média ponderada entre dois valores ( $r_i$  e  $\mu_i$ ), onde o peso é dado por  $w_i$ .

- $\hat{\theta}_i$  é a taxa suavizada no município  $i$ .
- $r_i$  é a taxa bruta de incidência no município  $i$ .
- $\mu_i$  é a taxa média nos **vizinhos** de  $i$  (ou taxa média geral, se for utilizado o método Bayesiano Empírico Global).
- e  $w_i$  é o peso a ser considerado nessa operação de média.

O peso  $w_i$  é calculado com base nas variâncias da taxa global e da taxa no município  $i$ . Quanto **maior a população**, menor é a variância e assim  $w_i$  é próximo de 1, dando mais peso à taxa bruta. Quanto **menor a população**, temos uma variância maior e taxas mais instáveis, e assim mais peso é dado para a média dos vizinhos.

No R temos uma função que realiza esses cálculos diretamente - `EBLocal()`, do pacote `spdep`. Vamos aplicá-la ao nosso mesmo caso de incidência de dengue em São Paulo em 2015:

```
# Relembrando a estrutura dos dados
head(dengue_2015)
```

	<code>#&gt; cod_ibge</code>	<code>cod_mun6</code>	<code>nome_mun</code>	<code>ano</code>	<code>pop</code>	<code>casos</code>	<code>inc</code>
<code>#&gt; 1</code>	3500105	350010	Adamantina	2015	34285	1140	332.50693
<code>#&gt; 2</code>	3500204	350020	Adolfo	2015	3903	121	310.01793
<code>#&gt; 3</code>	3500303	350030	Aguai	2015	32192	2487	772.55219
<code>#&gt; 4</code>	3500402	350040	Águas da Prata	2015	7558	336	444.56205
<code>#&gt; 5</code>	3500501	350050	Águas de Lindóia	2015	17690	165	93.27304
<code>#&gt; 6</code>	3500550	350055	Águas de Santa Bárbara	2015	6313	16	25.34453

A função `EBLocal()` recebe três parâmetros: o número de casos em cada área, a população de cada área, e a estrutura de vizinhança, que definimos anteriormente. Ou seja, vamos optar por uma das estratégias experimentadas (vizinhança por conectividade, vizinhos mais próximos, ou por distância).

```
inc_bayes_empirico ← EBlocal(
  ri = dengue_2015$casos,
  ni = dengue_2015$pop,
  nb = viz_sp # matriz de vizinhança definida por contiguidade
)

head(inc_bayes_empirico)
```

	<code>#&gt; raw</code>	<code>est</code>
<code>#&gt; 1</code>	0.033250693	0.033242118
<code>#&gt; 2</code>	0.031001793	0.030968000
<code>#&gt; 3</code>	0.077255219	0.077160864
<code>#&gt; 4</code>	0.044456205	0.044903289
<code>#&gt; 5</code>	0.009327304	0.009712328
<code>#&gt; 6</code>	0.002534453	0.002692929

O script calcula a estimativa de incidência suavizada usando o método Bayesiano Empírico Local.

A função `EBlocal()` recebe:

- `ri`: o número de casos de dengue observados em cada município (`dengue_2015$casos`),
- `ni`: a população exposta em cada município (`dengue_2015$pop`),
- `nb`: a matriz de vizinhança (`viz_sp`), que define quais municípios são vizinhos.

O Bayesiano Empírico Local ajusta as taxas observadas considerando a informação dos municípios vizinhos, suavizando variações extremas que podem ser causadas por populações pequenas ou números baixos de casos.

O resultado (`inc_bayes_empirico`) é um vetor com a taxa de incidência suavizada para cada município. O comando `head(inc_bayes_empirico)` exibe as primeiras linhas do vetor para inspecionar os valores calculados.

Vemos que o objeto retorna as taxas bruta (`raw`) e suavizadas (`est`). Vamos extrair a segunda coluna (suavizada) e multiplicá-la por 10.000, como fizemos com a incidência:

```
dengue_2015 ← dengue_2015 %>%
  mutate(inc_EB = 1e4*inc_bayes_empirico[,2])
```

O script adiciona uma nova variável `inc_EB` ao conjunto de dados `dengue_2015`.

A operação faz o seguinte:

- Utiliza `mutate()` para criar uma nova coluna.
- `inc_bayes_empirico[,2]` seleciona a segunda coluna do objeto `inc_bayes_empirico`, que corresponde à incidência suavizada pelo método de Bayes Empírico.
- Multiplica esses valores por `10.000` (`1e4`) para expressar a incidência no formato padrão de casos por 10.000 habitantes.

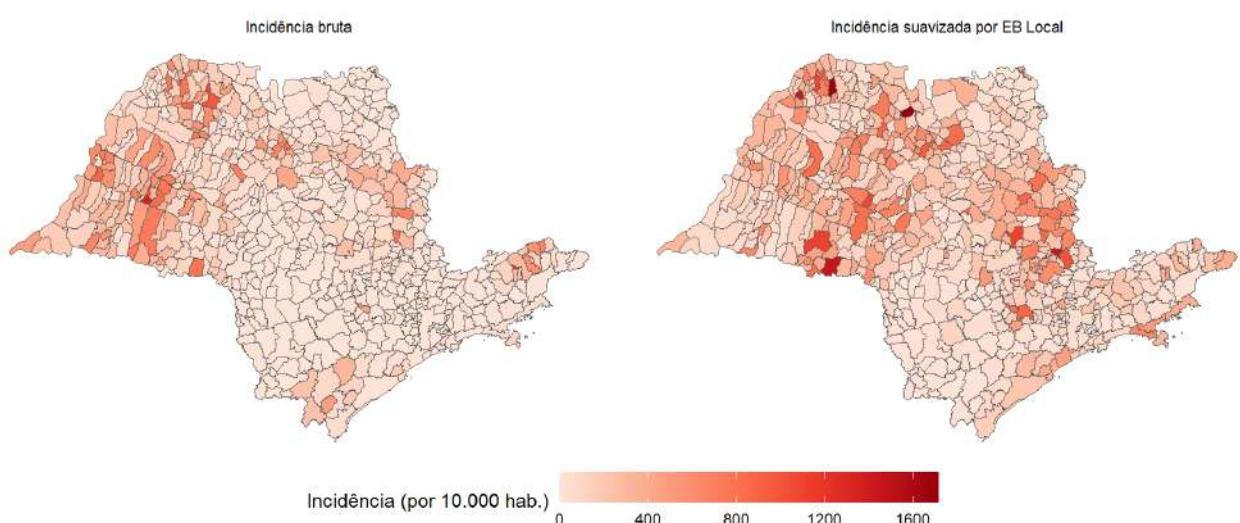
Assim, a variável `inc_EB` passa a representar a incidência de dengue suavizada, ajustada para facilitar a comparação com a incidência bruta originalmente calculada.



```
malha_sp_dengue$inc_EB ← dengue_2015$inc_EB

malha_sp_dengue %>%
  pivot_longer(cols = c(inc, inc_EB), names_to = "metodo", values_to = "inc")
%>%
  mutate(metodo = factor(metodo,
                         levels = c("inc", "inc_EB"),
                         labels = c("Incidência bruta", "Incidência suavizada
por EB Local"))) %>%
  ggplot(aes(geometry=geom)) +
  geom_sf(aes(fill = inc)) +
  labs(fill = "Incidência (por 10.000 hab.)") +
  scale_fill_distiller(palette = "Reds", direction=1) +
  facet_wrap(~metodo) +
  theme_void() +
  theme(legend.position = "bottom",
        legend.key.width = unit(1.5, "cm"),
        plot.title = element_text(hjust=0.5, size=16))
```

**Figura 47: Incidência bruta e suavizada de dengue nos municípios de São Paulo, 2015.**



O script compara a incidência bruta de dengue com a incidência suavizada por Bayes Empírico Local (EB Local) nos municípios de São Paulo em 2015.

Primeiramente, a variável `inc_EB` (incidência suavizada) é adicionada ao objeto espacial `malha_sp_dengue`.

Depois, utilizando `pivot_longer()`, os dados são reorganizados de formato “largo” para “longo”, empilhando as duas colunas de incidência (`inc` e `inc_EB`) em uma só, com uma nova coluna `metodo` identificando a origem dos valores.

Em seguida, `mutate()` ajusta os nomes dos métodos para labels mais amigáveis: “Incidência” para os valores brutos e “Incidência corrigida por EB Local” para os valores suavizados.

O gráfico é construído usando `ggplot2`, onde:

- Cada município é desenhado com `geom_sf()`, preenchido por uma escala de cor proporcional à incidência.
- `scale_fill_distiller()` aplica uma paleta de vermelhos para indicar diferentes níveis de incidência.
- `facet_wrap(~metodo)` cria dois mapas lado a lado: um para a incidência bruta e outro para a incidência suavizada.
- O layout é limpo com `theme_void()`, e elementos como título e legenda são personalizados para melhor apresentação.

O resultado final permite comparar visualmente o efeito da suavização por Bayes Empírico Local na distribuição da incidência de dengue.

Vemos que outros padrões chamam a atenção ao comparar a taxa bruta e a taxa suavizada na Figura 47: vê-se que houve redução de algumas flutuações principalmente no sul do estado, e alguns municípios tiveram uma elevação na taxa após a suavização. As maiores diferenças entre incidência bruta e suavizada se dão principalmente nos municípios menores - onde a baixa população implica num menor peso para a incidência bruta calculada, levando a uma maior influência das taxas dos vizinhos.

## *Principais modelos de regressão espacial para dados de área*

Além das análises exploratórias para os dados espaciais de área, é possível avançar na modelagem estatística dos dados por meio de modelos que considerem a dependência espacial dos dados. Entre os mais utilizados estão os modelos de autocorrelação espacial global, como o **SAR (Simultaneous Autoregressive Model)** e o **CAR (Conditional Autoregressive Model)** (CRESSIE, 1993). O modelo SAR incorpora a dependência espacial diretamente no valor da variável dependente (por exemplo, a incidência de uma doença), assumindo que o valor observado em uma região é influenciado pelos valores das regiões vizinhas. Já o modelo CAR considera essa dependência na estrutura do erro ou resíduo do modelo, sendo muito utilizado em abordagens Bayesianas para modelagem de risco em áreas geográficas.

Outra abordagem é a dos modelos com efeitos espaciais locais, como a **Regressão Geograficamente Ponderada (GWR)** (FOTHERINGHAM et al., 2002). Diferente dos modelos SAR e CAR, que assumem relações espaciais globais, o GWR permite que os coeficientes da regressão variem ao longo do espaço, ajustando uma equação específica para cada localidade. Isso possibilita identificar como os efeitos de uma variável explicativa sobre a resposta mudam de acordo com a localização, sendo especialmente útil em contextos em que há forte heterogeneidade espacial.



### Algumas referências para aprofundamento no tema:

CRESSIE, N. (1993). Statistics for Spatial Data (Revised edition). New York: Wiley.

FOTHERINGHAM, A. S., BRUNSDON, C., & CHARLTON, M. (2002). Geographically Weighted Regression: The Analysis of Spatially Varying Relationships. Wiley.

## *Considerações finais*

As técnicas exploratórias e os modelos de área são ferramentas cruciais na análise de dados espaciais em epidemiologia, permitindo trabalhar com dados de saúde agregados por unidades geográficas, como municípios, bairros, estados ou distritos sanitários. Sua utilidade é imensa e pragmática, pois alinham a técnicas de visualização e análise estatística à realidade da gestão e vigilância em saúde pública.

A maioria dos dados epidemiológicos da vigilância tais como casos de doenças, óbitos, cobertura vacinal, etc... é disponibilizada publicamente em formato agregado. Isso ocorre tanto por razões operacionais dos sistemas de saúde quanto para proteger a confidencialidade dos indivíduos (conforme a Lei nº 13.709/2018; Lei Geral de Proteção de Dados – LGPD).

A criação dos mapas temáticos que evidenciam o risco de uma doença na região de estudo ajuda a visualizar e identificar padrões espaciais claros — como aglomerados de alto risco (clusters) ou gradientes de risco entre regiões, padrões esse que não seriam visíveis em uma simples tabela de dados.

O uso de modelos espaciais é fundamental e possui diversas aplicação. Algumas delas são:

1. Permitir a correção de instabilidade em taxas observadas em pequenas áreas com populações pequenas, as quais podem gerar taxas de doença muito instáveis
2. Identificação de fatores de risco: modelos de regressão, como a regressão de poisson ou binomial negativa, são aplicados para entender quais características da área (variáveis de contexto) estão associadas ao aumento do número de casos. É possível, por exemplo, analisar se municípios com menor cobertura de saneamento básico ou com menor renda média apresentam, de fato, maiores taxas de certas doenças infecciosas.
3. Ao gerar mapas de risco suavizados e identificar áreas prioritárias, eles permitem que os gestores de saúde pública direcionem recursos de forma mais eficiente, como campanhas de prevenção, fiscalização ou alocação de equipes de saúde, para os locais que mais necessitam.

Essas são apenas alguns exemplos de como os mapas exploratórios e os modelos espaciais podem contribuir com a epidemiologia, vigilância em saúde e tomada de decisões.

## Referências

- BAILEY, T. C. Spatial statistical methods in health. *Cadernos de Saúde Pública*, Rio de Janeiro, vol. 17, no. 5, p. 1083–1098, Oct. 2001.
- PEBESMA, E.; BIVAND, R. Spatial Data Science: With Applications in R. 1st ed. New York: Chapman and Hall/CRC, 2023. Available at: <https://www.taylorfrancis.com/books/9780429459016>. Accessed on: 19 May 2025.
- PFEIFFER, D. (Ed.). Spatial analysis in epidemiology. Oxford; New York: Oxford University Press, 2008.
- WALLER, L. A.; GOTWAY, C. A. Applied spatial statistics for public health data. Hoboken, N.J: John Wiley & Sons, 2004(Wiley series in probability and statistic).

## Módulo 4: Geoestatística

A geoestatística é uma abordagem estatística voltada para a análise de **fenômenos espaciais**, permitindo a modelagem e predição de variáveis contínuas distribuídas geograficamente. Essa metodologia baseia-se na premissa de que os valores observados em pontos próximos tendem a ser mais semelhantes entre si do que aqueles mais distantes, um conceito conhecido como **dependência espacial**.

Embora amplamente utilizada em áreas como **geologia, ciências ambientais, agricultura de precisão e estudos de recursos hídricos**, a geoestatística também desempenha um papel estratégico em **epidemiologia e vigilância em saúde**.

Na vigilância em saúde, os profissionais frequentemente lidam com dados que possuem um componente espacial claro. Por exemplo, a propagação de doenças transmitidas por vetores, como a dengue, não ocorre de forma aleatória, mas tende a se concentrar em áreas onde as condições ambientais favorecem a proliferação do mosquito. No entanto, **nem sempre é possível coletar dados em todos os locais de interesse**, o que torna a geoestatística uma ferramenta essencial. Por meio de suas técnicas, é possível estimar valores em locais não amostrados com base nas observações disponíveis.

No contexto da vigilância em saúde, a geoestatística pode ser aplicada em diferentes áreas:

**Vigilância Epidemiológica:** Mapear a incidência de doenças transmissíveis, como dengue, zika ou chikungunya, identificando áreas de maior risco para direcionar intervenções preventivas e controle vetorial.

**Vigilância Ambiental:** Avaliar a qualidade da água ou do ar em diferentes regiões, prever áreas com maior probabilidade de contaminação e compreender a dispersão espacial de vetores de doenças, como a distribuição de ovos do Aedes aegypti em ovitrampas.

**Saúde do Trabalhador:** Identificar regiões com maior exposição a fatores de risco ambientais que possam afetar a saúde ocupacional, como áreas próximas a indústrias ou locais com alta poluição atmosférica.

Ao longo deste módulo, você aprenderá a utilizar ferramentas geoestatísticas no R para explorar dados espaciais, gerar mapas interpolados e interpretar resultados que possam subsidiar ações estratégicas em saúde pública.

## *Conceitos e objetivos*

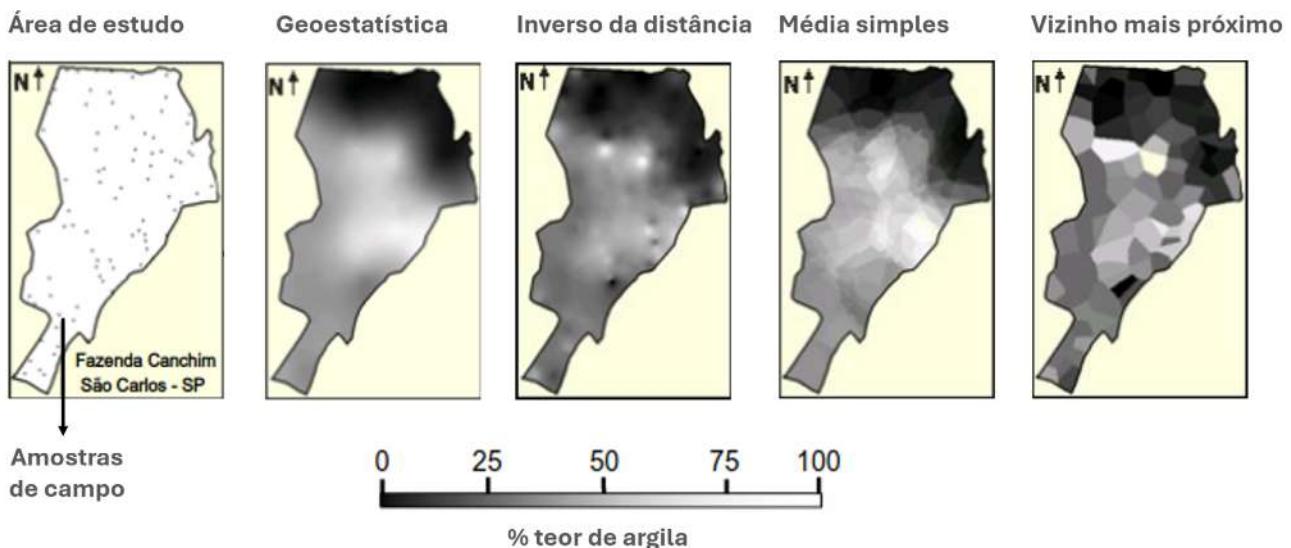
Os dados analisados pela geoestatística geralmente são representados por coordenadas geográficas (latitude e longitude) associadas a uma variável de interesse, como o número de casos de uma doença, níveis de poluição do ar, concentrações de contaminantes em corpos d'água ou indicadores socioambientais. A capacidade de explorar a distribuição espacial desses dados possibilita uma compreensão mais profunda dos fenômenos que afetam a saúde da população.

Os principais objetivos da geoestatística aplicados à vigilância em saúde incluem:

- I. Analisar a distribuição espacial dos dados amostrados e identificar padrões ou tendências dentro da área de estudo. Isso permite, por exemplo, detectar áreas de maior risco de ocorrência de doenças, zonas de exposição ambiental crítica ou áreas vulneráveis a desastres naturais.
- II. Estimar valores em locais não amostrados com base nas observações disponíveis. Por meio de técnicas de interpolação é possível criar mapas contínuos que representam a distribuição do fenômeno analisado em toda a região, mesmo onde não há dados coletados diretamente.

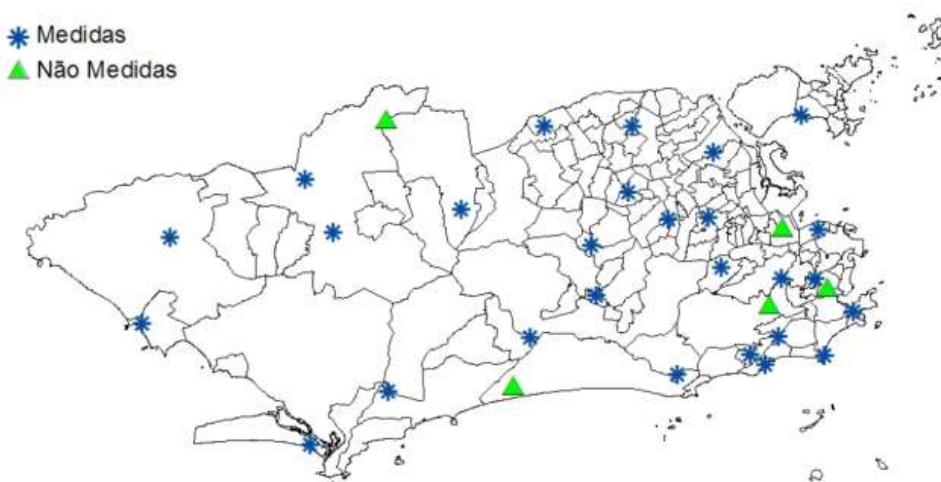
De modo geral, outros exemplos podem caracterizar a aplicação da geoestatística em saúde, como veremos a seguir.

**Figura 48: Interpolação geoestatística do teor de argila no solo na Fazenda Canchin (São Carlos - SP).**



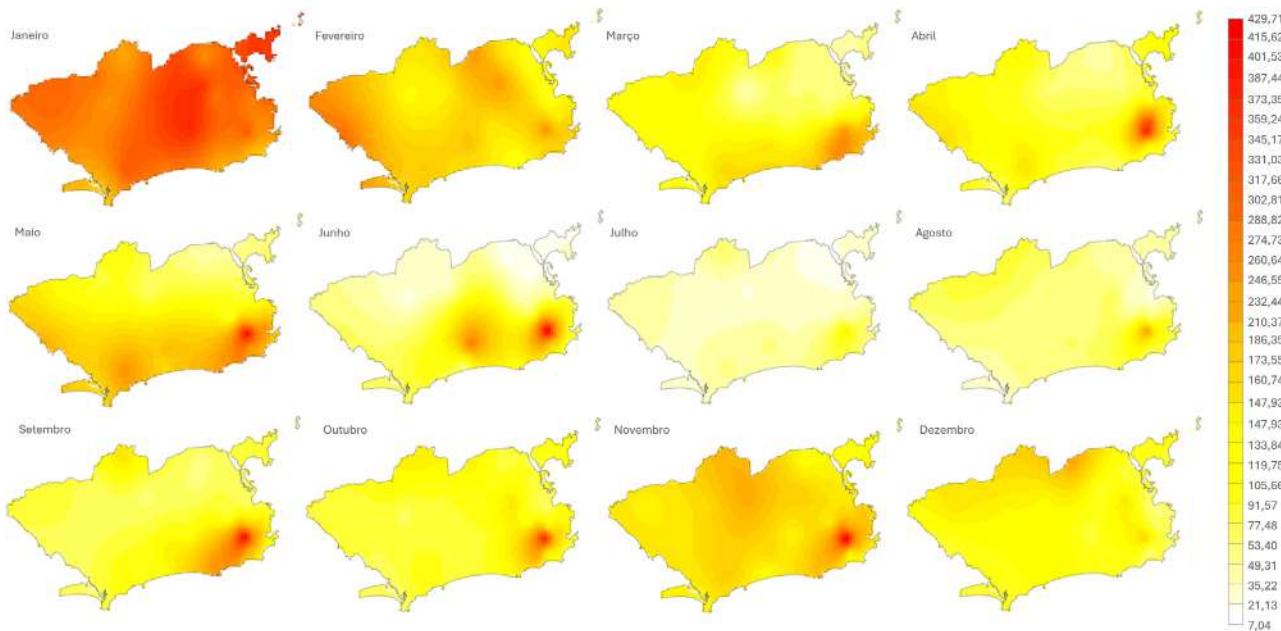
A Figura 48 ilustra um exemplo da aplicação da geoestatística para estimar o teor de argila no solo na Fazenda Canchin, em São Carlos - SP. Diferentes métodos de interpolação são utilizados para prever os valores do teor de argila no solo em locais não amostrados, destacando as variações espaciais da variável estudada.

**Figura 49: Distribuição das estações pluviométricas no município do Rio de Janeiro.**



A Figura 49 exibe o mapa com as localizações das estações de monitoramento pluviométrico na cidade do Rio de Janeiro. Os pontos azuis representam as estações onde há medições da precipitação, enquanto os triângulos verdes indicam áreas sem medições diretas. A partir desses dados, é possível utilizar técnicas de interpolação geoestatística para estimar a precipitação em toda a cidade, permitindo uma análise mais completa da variabilidade espacial das chuvas. Esse tipo de estudo é fundamental para previsão de eventos extremos, planejamento de drenagem urbana e mitigação de impactos de enchentes.

**Figura 50: Interpolação geoestatística da precipitação mensal no município do Rio de Janeiro (2008), baseada na metodologia de modelagem espacial aplicada por [Teixeira & Cruz \(2010\)](#).**



Já a Figura 50 apresenta a interpolação geoestatística da precipitação mensal na cidade do Rio de Janeiro ao longo do ano de 2008. Utilizando dados de estações pluviométricas distribuídas pelo município, foi possível estimar a precipitação em áreas não medidas. Cada mapa representa um mês do ano, permitindo a visualização da variação espacial das chuvas. As regiões em tons avermelhados indicam os locais com maior índice pluviométrico, enquanto as áreas em amarelo apresentam menor precipitação. Esse tipo de análise é essencial para compreender padrões sazonais e subsidiar o planejamento urbano e a gestão de riscos climáticos.



## *Padrões espaciais: efeitos de primeira e segunda ordem*

Ao analisar fenômenos espaciais em saúde pública, não estamos apenas interessados **onde** os eventos ocorrem, mas também **porque** eles se distribuem de determinada forma. Para isso, é fundamental compreender os padrões espaciais que podem influenciar a sua ocorrência.

Esses padrões podem ser explicados pela combinação de dois tipos de efeitos: efeitos de **primeira ordem**, que captam variações mais amplas, e efeitos de **segunda ordem**, que revelam relações locais, conforme visto nos módulos 2 e 3.

Vamos entender esses conceitos no uso da geoestatística.

### *Efeito de primeira ordem*

O efeito de primeira ordem refere-se à **variação do valor médio** do fenômeno analisado ao longo do espaço, indicando uma **tendência global**, ou seja, variações médias de grande escala no território analisado.

Um exemplo comum de efeito de primeira ordem é a temperatura média em uma região, onde a temperatura tende a aumentar ou diminuir de forma contínua em função da latitude, altitude ou até proximidade do oceano. Esse efeito pode ser representado por um gradiente térmico no espaço, mesmo sem considerar interações locais.

Em termos matemáticos, o efeito de primeira ordem está relacionado à função média do processo espacial, representada por:

$$Y(s) = \mu(s) + e(s)$$

Onde:

- $\mu(s)$  representa a tendência espacial;
- $e(s)$  representa a variação residual ou o erro.

## *Efeito de segunda ordem*

O efeito de segunda ordem, por sua vez, está associado à **dependência espacial entre pontos próximos**. Ou seja, descreve como as observações feitas em locais geograficamente vizinhos tende a apresentar valores semelhantes (estrutura de correlação espacial).

Enquanto o efeito de primeira ordem captura tendências globais (padrões amplos), o efeito de segunda ordem revela padrões locais ou variações em pequena escala, que geralmente são detectadas ao observar a correlação espacial entre os pontos. Um exemplo comum é a precipitação, onde áreas vizinhas apresentam volumes de chuvas similares. Isso se dá devido à continuidade espacial do fenômeno.

Esse tipo de efeito pode ser analisado por meio da função de **covariância espacial** ou do **semivariograma**, técnicas que quantificam a relação entre pontos geograficamente próximos.

É fundamental saber distinguir esses dois efeitos ao elaborar modelos espaciais. Enquanto o efeito de primeira ordem pode ser tratado por funções de tendência, o efeito de segunda ordem requer modelagem da estrutura de dependência espacial, como krigagem e interpolação geoestatística. Veremos mais sobre isso adiante.

A análise conjunta desses efeitos permite a produção de estimativas mais precisas e uma melhor interpretação dos padrões espaciais subjacentes a diversos fenômenos naturais e ambientais, muitas vezes presentes na temática da saúde pública.

## Por que essa distinção importa na vigilância em saúde?

Porque os efeitos de primeira ordem ajudam a identificar tendências gerais, como a maior ocorrência de doenças em regiões mais urbanizadas ou com piores indicadores socioeconômicos. Já os efeitos de segunda ordem revelam padrões locais que podem indicar áreas de risco ou a ocorrência de surtos.

Reconhecer essa diferença é essencial para evitar interpretações equivocadas e permite orientar melhor as ações de vigilância, alocação de recursos e planejamento de intervenções.

No próximo tópico, veremos como esses efeitos podem ser explorados na prática.

### *Análise exploratória do efeito de primeira ordem*

A análise exploratória é o primeiro passo na investigação de dados espaciais. Ela permite identificar padrões e tendências que ajudam a diferenciar os efeitos de primeira e segunda ordem. Para isso, a visualização de dados ajuda a compreender a distribuição espacial e a intensidade do fenômeno em estudo.

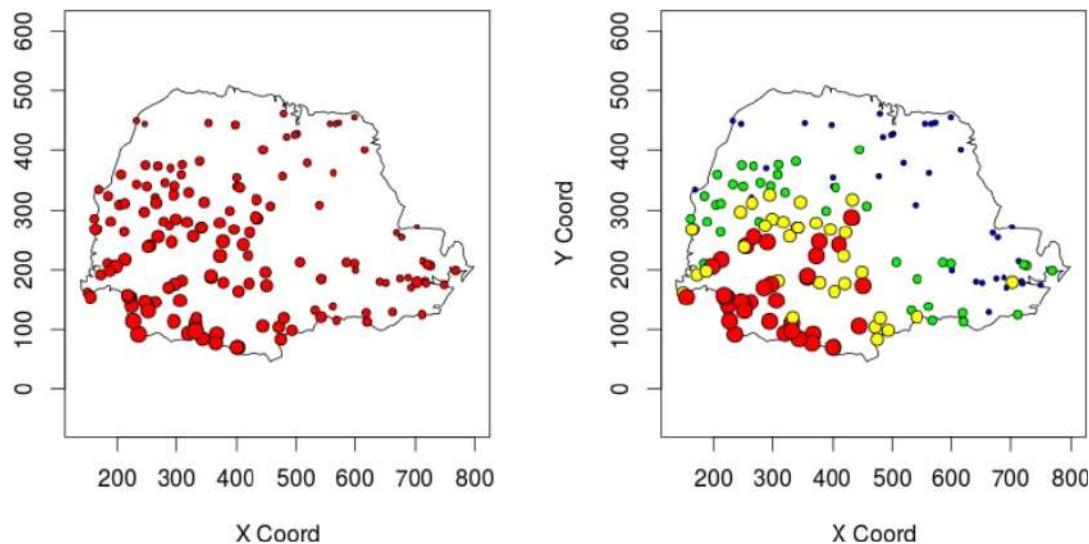
No caso do efeito de primeira ordem, buscamos identificar variações na intensidade do fenômeno ao longo do espaço. As principais ferramentas para essa análise incluem:

**Mapas de distribuição espacial:** Representam a localização dos pontos amostrados e a intensidade do fenômeno estudado. São úteis para visualizar padrões globais e regiões com maior ou menor concentração do evento analisado.

**Gráficos de dispersão entre a variável de interesse ( $Y(s)$ ) versus coordenadas espaciais (por exemplo, latitude e longitude):** Auxiliam na detecção de tendências espaciais, permitindo relacionar a variável de interesse com coordenadas e verificar se há aumento ou diminuição sistemática dos valores a depender da direção.

Vamos acompanhar alguns exemplos em seguida.

**Figura 51: Medidas pluviométricas em 143 estações monitoradoras no estado do Paraná.**



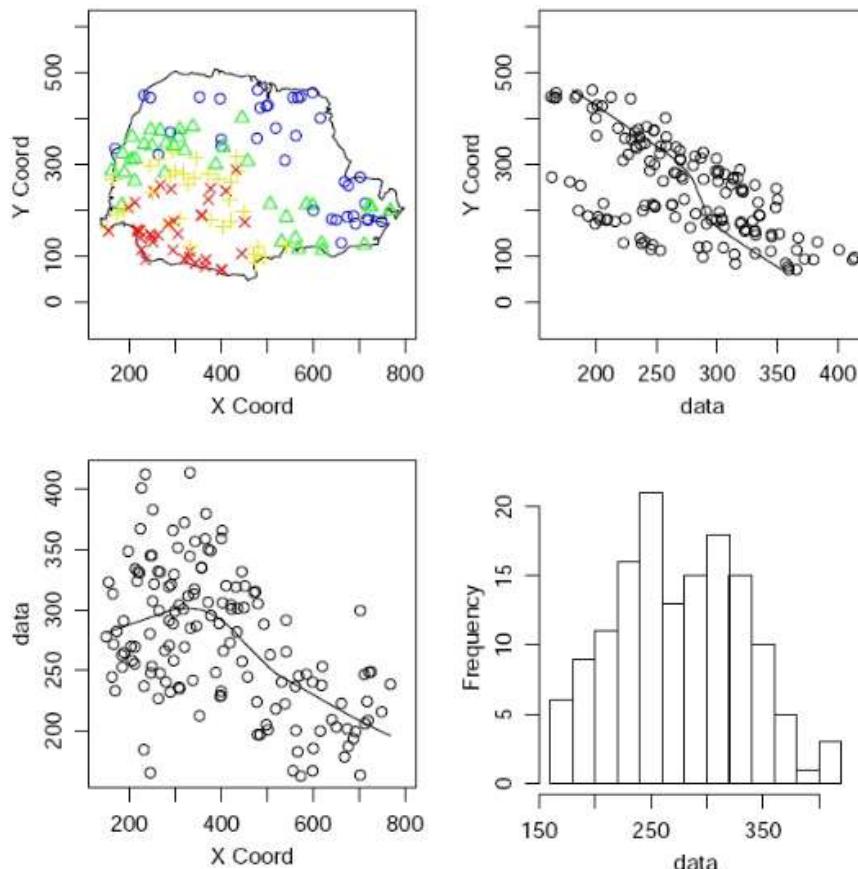
A Figura 51, mostra as medições de chuvas realizadas em 143 estações de monitoramento no estado do Paraná. Os gráficos de dispersão exibem a distribuição espacial dessas estações com base em coordenadas X e Y, representando a localização dos pontos de medição.

O gráfico à esquerda, exibe a localização das estações pluviométricas no estado do Paraná, destacando os pontos de medição (em vermelho). Já com essa visualização é possível identificar a densidade e a cobertura espacial dos pontos de medição para, assim, ter uma compreensão da rede de monitoramento. Já o gráfico à direita apresenta uma diferenciação por cores, que pode indicar variações na precipitação medida. As cores distintas (vermelho, amarelo, verde e azul) sugerem possíveis agrupamentos ou padrões espaciais na distribuição dos dados.

Essas representações visuais ajudam a compreender se há zonas com maior ou menor volume de chuva, padrões espaciais consistentes ou necessidade de melhorar a cobertura das estações na rede de monitoramento.

Agora, vamos para outro exemplo.

**Figura 52: Analisando a variação da intensidade da chuva segundo Latitude (Y) e Longitude (X).**



A Figura 52 apresenta diferentes visualizações de um conjunto de dados espaciais, também no estado do Paraná. São elas:

- O mapa (superior esquerdo) mostra a distribuição espacial dos pontos de medição, com símbolos e cores que podem representar diferentes faixas de precipitação (ou outro evento ambiental).
- O gráfico de dispersão (superior direito), a coordenada Y versus data, revela uma tendência de redução dos valores ao longo do eixo norte-sul (coordenada Y), principalmente a medida que a coordenada Y aumenta (gradiente espacial).
- O outro gráfico de dispersão (inferior esquerdo), que relaciona a coordenada X com a variável de interesse, mostra um padrão de tendência que pode indicar variações sistemáticas ao longo do eixo leste-oeste (coordenada X).
- O histograma (inferior direito) mostra a distribuição de frequência dos valores analisados, sugerindo que os dados seguem uma distribuição aproximadamente normal, porém com uma leve assimetria.

## O que podemos concluir?

A análise exploratória mostrou que os dados possuem boa cobertura territorial e tendências espaciais claras (com tendências nos eixos X e Y) indicando a presença de efeitos de primeira ordem. Esses padrões ajudam a identificar regiões com maior risco para determinados agravos, como áreas com alta pluviosidade associadas a surtos de dengue, ou zonas de menor temperatura com maior incidência de síndromes respiratórias. Já o histograma dos dados a entender a distribuição dos valores da variável medida, fornecendo *insights* sobre sua dispersão e comportamento estatístico. Para a vigilância em saúde, essa etapa inicial é essencial: ela orienta onde priorizar investigações, planejar ações de campo e alocar recursos de forma mais eficiente.

### Análise exploratória do efeito de segunda ordem

Como vimos antes, os efeitos de segunda ordem refletem a dependência espacial entre observações próximas. Em outras palavras, eles ajudam a identificar se há padrões locais nos dados, uma característica essencial para a detecção de áreas de risco, surtos e clusters de agravos na vigilância em saúde.

Para investigá-los, precisaremos aprofundar alguns conceitos e ferramentas como o **covariograma** e o **variograma**:

- **Covariograma:** Mede a covariância espacial entre pares de pontos em função da distância. Essa medida permite avaliar a correlação entre locais distintos.
- **Variograma:** Analisa a variabilidade espacial do fenômeno, quantificando como as diferenças entre valores observados aumentam com a distância. É uma ferramenta essencial para modelagem geoestatística.

No contexto da vigilância em saúde, essa abordagem exploratória é fundamental para orientar a escolha dos modelos espaciais mais adequados e garantir interpretações mais precisas dos dados analisados. Veremos mais desses conceitos logo mais adiante.

A análise de processos espaciais pode ser simplificada ao considerar certas propriedades estatísticas, como a **estacionariedade** e a **isotropia**:

- **Estacionariedade:** supõe que as propriedades estatísticas do processo não dependem da localização em si, mas apenas da separação entre os pontos. Nesse caso:

- A média do processo é constante em todo o espaço:

$$E(Y(s)) = \mu, \quad \forall s$$

- A variância da diferença entre valores em dois pontos depende apenas da distância  $h$ , e não da localização:

$$\text{Var}(Y(s+h) - Y(s)) = 2\gamma(h)$$

para a qual  $2\gamma(h)$  é chamado de variograma, e  $\gamma(h)$  representa o semi-variograma.

- A covariância entre dois pontos depende apenas do deslocamento entre eles:

$$C(s_i, s_j) = C(s_i - s_j) = C(h)$$

para a qual  $C(h)$  representa o covariograma do processo.

- **Isotropia:** Supõe que suas propriedades estatísticas não variam em relação à direção. Isso significa que a covariância entre dois pontos depende apenas da distância euclidiana entre eles:

$$C(s_i, s_j) = C(\|s_i - s_j\|) = C(h)$$

para a qual  $\|\cdot\|$  denota a distância euclidiana. Em um processo isotrópico, a correlação espacial não depende da orientação, apenas da magnitude da separação entre os pontos.

Essas suposições, quando verificadas, permitem aplicar modelos mais simples e eficientes — o que é especialmente útil em contextos de rotina da vigilância, onde é preciso equilibrar robustez estatística com agilidade na produção de informações. Contudo, vamos detalhar uma pouco mais os conceitos técnicos.

## Variograma, covariograma e correlograma

Retomando ao conceito, o variograma mede como a semelhança entre os valores observados diminui com o aumento da distância entre os pontos. Em outras palavras, o variograma nos ajuda a responder: “até que ponto os dados estão espacialmente relacionados?”

O variograma empírico é a estimativa prática do variograma baseada nos dados observados. Ele calcula, para diferentes distâncias  $h$ , a média das diferenças quadráticas entre valores de pontos separados por essa distância.

Um estimador para o variograma empírico é dado pela fórmula:

$$2\gamma^2(h) = \frac{1}{n(h)} \sum_{s_i-s_j=h} (y(s_i) - y(s_j))^2$$

para a qual:

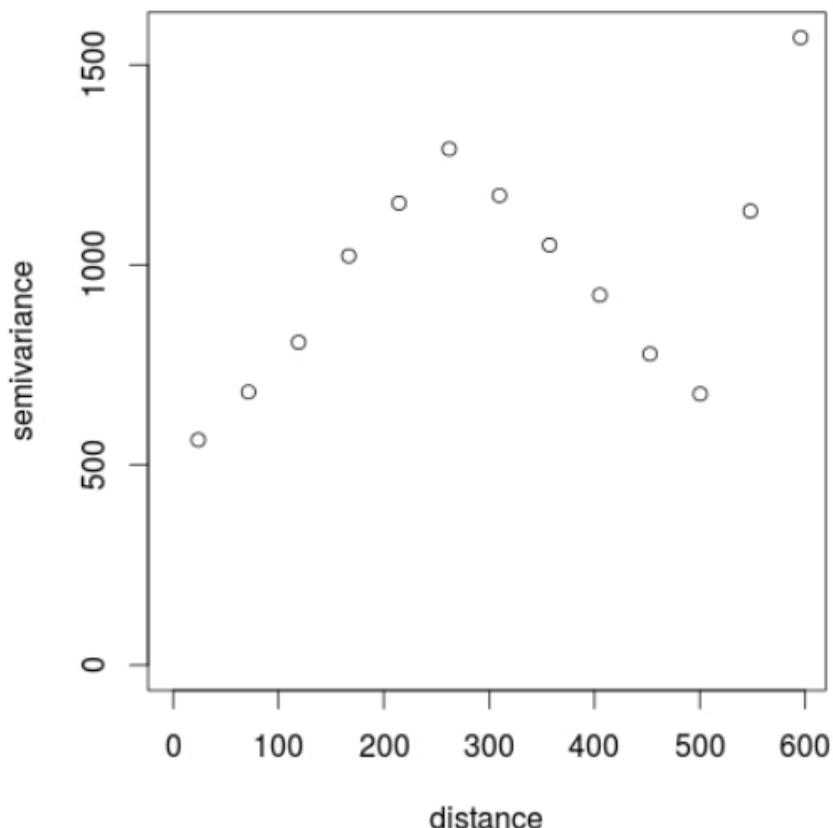
- $\gamma^2(h)$  é o semi-variograma para a distância  $h$ ;
- $n(h)$  é o número de pares de pontos separados pela distância  $h$ ;
- $y(s_i)$  e  $y(s_j)$  são os valores observados nos pontos  $s_i$  e  $s_j$ , respectivamente;
- A soma é realizada sobre todos os pares de observações que possuem uma separação espacial  $h$ .

Essa função descreve como a variabilidade dos dados muda com a distância e, em geral:

- Para pequenas distâncias  $h$ , espera-se que os valores de  $y(s)$  sejam mais similares, resultando em valores menores de  $\gamma(h)$ .
- À medida que a distância  $h$  aumenta, a variabilidade também aumenta, refletindo uma menor correlação espacial entre os pontos.
- De modo geral, o variograma se estabiliza a partir de uma certa distância, atingindo um, indicando que os valores deixam de apresentar dependência espacial significativa.

A Figura 53 mostra um exemplo de variograma empírico construído a partir de dados de precipitação no estado do Paraná. Essa curva representa a relação entre a semivariância (no eixo vertical) e a distância entre os pontos (eixo horizontal).

**Figura 53:** Variograma empírico para os dados de chuva do Paraná.



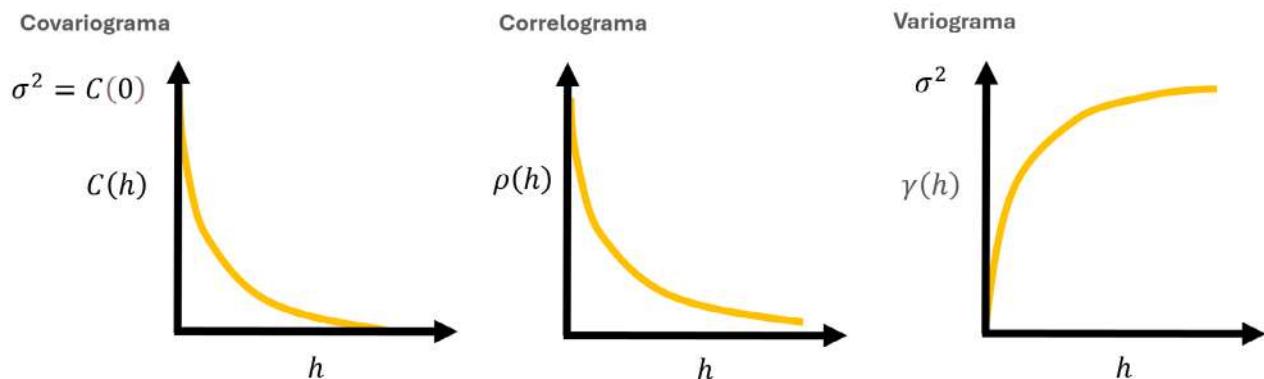
A interpretação desse gráfico pode ser feita considerando alguns aspectos importantes:

- **Tendência Geral:** A semivariância aumenta à medida que a distância cresce, o que indica que pontos próximos têm valores mais semelhantes, enquanto pontos mais distantes apresentam maior variabilidade. Após atingir um pico (cerca de 250-300 unidades de distância), a semivariância começa a diminuir, o que pode indicar ciclo espacial, anisotropia ou um problema na estimativa do variograma.
- **Alcance (Range):** Em um variograma típico, a semivariância atinge um platô (sill), indicando a distância a partir da qual os valores deixam de estar correlacionados. Na figura, não há um platô claramente definido, mas o pico da semivariância pode indicar um alcance em torno de 250-300.
- **Comportamento Anômalo:** A queda na semivariância após o pico pode indicar que o processo espacial subjacente não é puramente aleatório e pode haver alguma estrutura cíclica na variabilidade. O último ponto da curva parece um outlier, o que pode ser devido a um número reduzido de pares de pontos naquela distância, afetando a estimativa.

O variograma empírico é usado para ajustar modelos teóricos, como o esférico, exponencial ou gaussiano. Esses modelos são fundamentais para métodos de interpolação geoestatística como a krigagem, frequentemente aplicada em saúde pública para estimar indicadores em áreas não amostradas ou com falhas de notificação.

Agora, vamos entender a **relação entre variograma, covariograma e correlograma**. Acompanhe abaixo na Figura 54.

**Figura 54: Comparação entre Covariograma Correlograma e Variograma.**



Na Figura 54 há três gráficos fundamentais na análise geoestatística e na modelagem de dependência espacial: **covariograma, correlograma** e **variograma**. Cada um deles descreve de maneira diferente a relação entre os valores de uma variável em função da distância  $h$  entre pontos de amostragem.

### **Covariograma ( $C(h)$ ):**

- Mede a covariância espacial entre pares de pontos separados por uma distância  $h$ .
- Possui valores altos para pequenas distâncias, indicando forte correlação entre pontos próximos.
- Decresce à medida que  $h$  aumenta, pois a influência espacial diminui.

### **Correlograma ( $\rho(h)$ ):**

- Representa a correlação espacial normalizada (coeficiente de correlação) entre pontos separados por  $h$ .
- Tem comportamento semelhante ao covariograma, mas varia entre -1 e 1, sendo útil para comparar diferentes variáveis e escalas.

### **Variograma ( $\gamma(h)$ ):**

- Mede a dispersão dos valores em função da distância, sendo definido como a metade da variância das diferenças entre pontos separados por  $h$ .
- Cresce conforme a distância aumenta, refletindo maior variabilidade entre pontos mais distantes.
- Tende a se estabilizar em um patamar (*sill*), representando a variância total do processo.

As principais diferenças entre eles são:

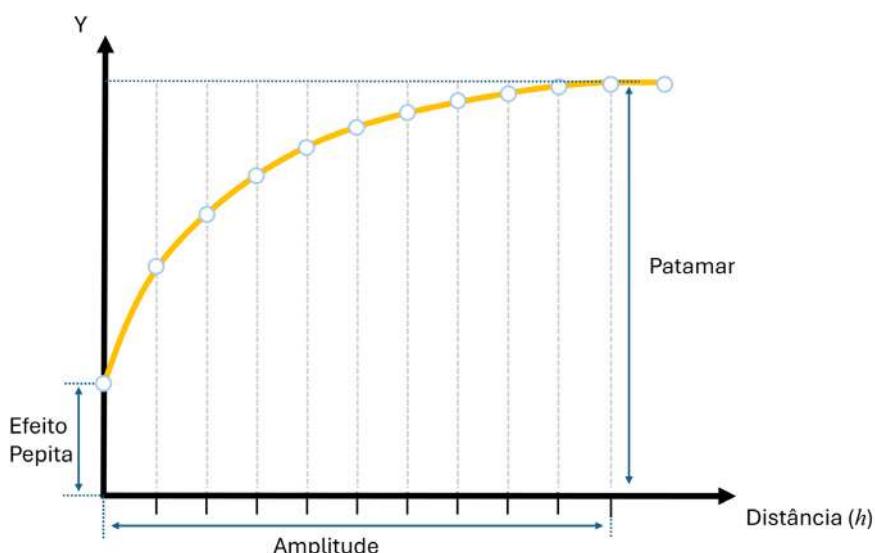
- Correlograma e covariograma são usados para descrever a dependência direta entre pontos, enquanto o variograma descreve a variabilidade.
- Para um processo espacial estacionário, o covariograma, correlograma e variograma fornecem informações semelhantes sobre a dependência espacial dos dados.
- O covariograma e o correlograma possuem a mesma forma, mas com uma diferença importante: o correlograma é normalizado, tendo seu máximo igual a 1.
- Variograma e covariograma estão diretamente relacionados, porém de maneira invertida, pois mede a variabilidade em vez da similaridade.
- Enquanto o covariograma começa com um valor máximo de  $\sigma^2$  quando  $h = 0$  e diminui até se aproximar de zero, o variograma inicia em zero e cresce até atingir o máximo de  $\sigma^2$ .



## Estrutura do variograma

Um variograma possui uma estrutura composta por três elementos fundamentais que ajudam a entender a estrutura de dependência espacial de uma variável regionalizada. Esses componentes são mostrados na Figura 55.

**Figura 55:** Principais componentes do variograma e sua interpretação.



Abaixo, destacamos suas principais características:

- **Efeito Pepita(Nugget):** Representa as variações aleatórias, erros de medição ou pequenas flutuações nos dados devido ao processo de coleta. É o valor do variograma quando  $h = 0$ , expresso por:

$$\gamma(0) = \tau^2$$

- **Patamar (Sill):** Corresponde ao valor máximo da semivariância. Indica o ponto a partir do qual as observações deixam de estar espacialmente correlacionadas. Ou seja, a variabilidade se estabiliza.
- **Amplitude (Range):** Corresponde a distância a partir da qual a correlação entre os pontos se torna insignificante. No gráfico, é o ponto no eixo das distâncias ( $x$ ) onde a curva do variograma atinge o **sill**, indicando que além dessa distância, as observações são essencialmente independentes.



## Variogramas para modelos isotrópicos

Como vimos, o variograma é uma ferramenta central na geoestatística. Para modelar a dependência espacial de um fenômeno, utiliza-se um modelo teórico ajustado ao variograma empírico. Em casos em que se assume que a dependência espacial é a mesma em todas as direções, isto é, o processo é isotrópico, alguns modelos são amplamente utilizados:

**Modelo Gaussiano:** Caracteriza-se por um crescimento suave e gradual da semivariância, apresentando um comportamento parabólico próximo à origem. Esse modelo é adequado para processos espaciais contínuos e suavizados.

$$\gamma(h) = \sigma^2 \left( 1 - \exp \left\{ - \left( \frac{h}{\phi} \right)^2 \right\} \right), \quad h > 0$$

**Modelo Exponencial:** Possui um crescimento mais acentuado no início, indicando forte dependência espacial em curtas distâncias. No entanto, a aproximação ao patamar ocorre de forma mais lenta, sendo útil para fenômenos com correlação espacial de curto alcance.

$$\gamma(h) = \sigma^2 \left( 1 - \exp \left\{ - \frac{h}{\phi} \right\} \right), \quad h > 0$$

**Modelo Esférico:** Apresenta um crescimento inicial rápido e, posteriormente, atinge o patamar de forma mais abrupta, tornando-se uma boa escolha para fenômenos que possuem um limite bem definido de correlação espacial.

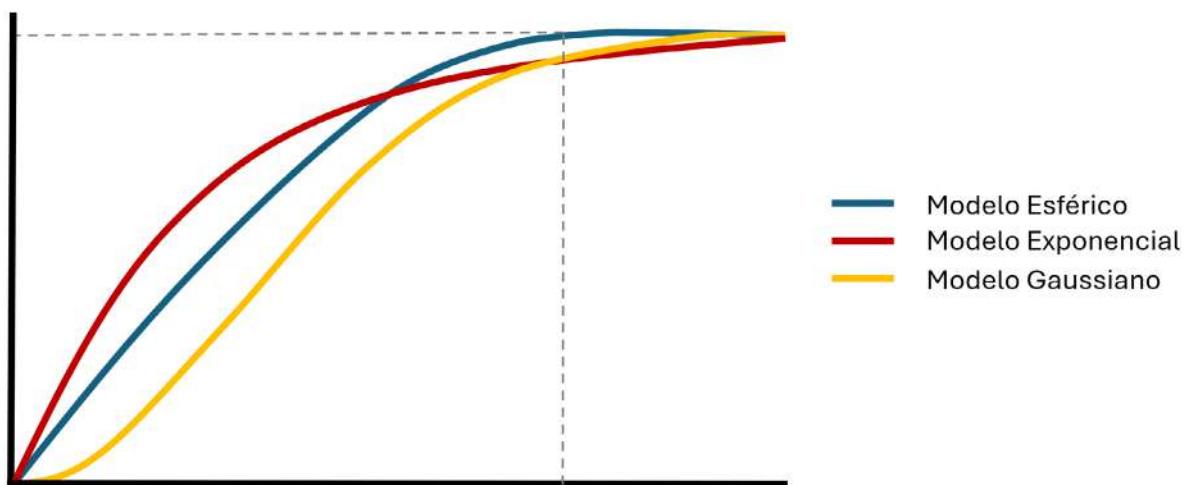
$$\gamma(h) = \begin{cases} \sigma^2, & h > \phi \\ \sigma^2 \left\{ \frac{2}{3} \left( \frac{h}{\phi} \right) - \frac{1}{2} \left( \frac{h}{\phi} \right)^3 \right\}, & 0 < h < \phi \end{cases}$$

A Figura 56 apresenta a comparação gráfica entre os três modelos. Cada curva mostra como diferentes estruturas espaciais se comportam em relação à distância entre os pontos:

- O modelo gaussiano, por exemplo, apresenta um crescimento lento e um comportamento parabólico próximo a origem e fornece um modelo para fenômenos extremamente contínuos.
- O modelo exponencial cresce mais rapidamente perto da origem, mas a aproximação da função ao patamar é mais lenta.
- Frequentemente os modelos são ajustados aos dados observados no variograma empírico, apenas por uma comparação visual.

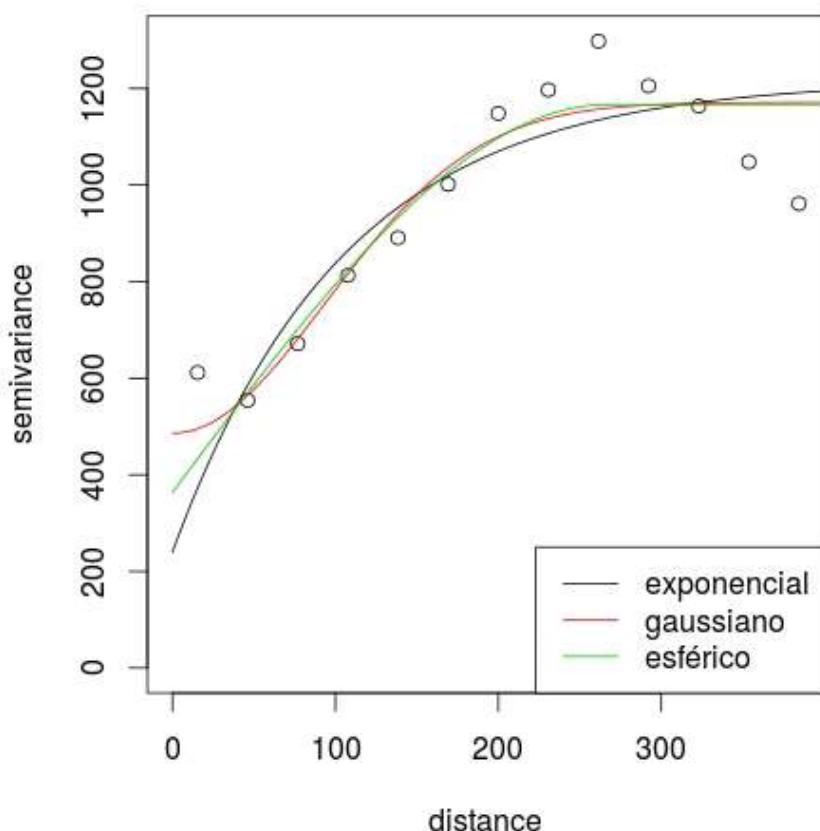
Acompanhe na Figura 56:

**Figura 56: Comparação entre Modelos de Variograma (Esférico, Exponencial e Gaussiano).**



Já a Figura 57 representa um exemplo que vimos do Estado do Paraná. Na figura é representado o ajuste dos três modelos.

**Figura 57: Ajuste de modelos de variograma aos dados de chuva no Paraná.**



Com base na figura, podemos dizer que:

- O modelo exponencial apresenta um ajuste mais rápido nas curtas distâncias, capturando melhor a estrutura local dos dados.
- O modelo gaussiano cresce de maneira mais suave e contínua, sendo adequado para fenômenos com transição gradual da dependência espacial.
- O modelo esférico se ajusta bem até uma determinada distância, após a qual a semivariância estabiliza rapidamente. Característica útil para representar fenômenos com dispersão limitada.

Agora, vamos ver alguns exemplos que ilustram como esses conceitos são utilizados na análise de dados espaciais em saúde pública e outras áreas.

## *Algumas aplicações da geoestatística*

A geoestatística é uma ótima estratégia para analisar fenômenos ambientais que afetam direta ou indiretamente a saúde da população. A seguir, apresentamos dois exemplos didáticos de aplicação geoestatística: o primeiro com base em dados de temperatura, que ilustra o potencial dessas técnicas para avaliar padrões espaciais e temporais de variáveis ambientais, e o segundo utilizando dados de oviposição do *Aedes*, evidenciando o potencial dessas técnicas para avaliar padrões espaciais e temporais relevantes para a vigilância em saúde.

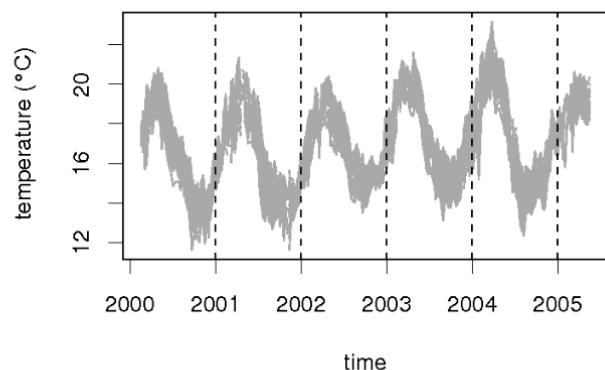
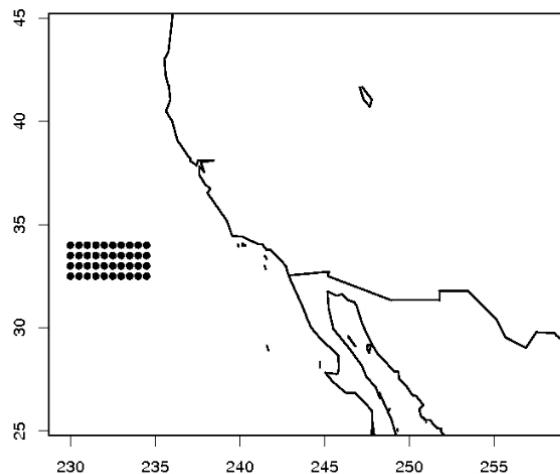
### *Aplicação 1*

A Figura 58 mostra uma aplicação da geoestatística utilizando dados de temperatura coletados em uma região do Oceano Pacífico, localizada ao longo da costa da Califórnia. O objetivo da análise é compreender a variabilidade espacial e temporal da temperatura ao longo do período estudado.

As medições foram realizadas em uma grade espacial regular de  $10 \times 4$  pontos (longitude x latitude), com uma resolução espacial de 0,5 graus, totalizando 40 pontos de amostragem ( $n = 40$ ). Os dados foram registrados a cada 8 dias, no intervalo de julho de 2000 a maio de 2005, resultando em 240 observações no tempo ( $T = 240$ ).



**Figura 58: Localização da grade de amostragem e série temporal da temperatura no Oceano Pacífico.**



O gráfico à esquerda exibe a localização da grade de amostragem sobre um mapa da região costeira da Califórnia. Já o gráfico à direita apresenta a série temporal da temperatura ( $^{\circ}\text{C}$ ) ao longo do período analisado, destacando padrões sazonais recorrentes.

Esse tipo de informação é essencial para compreender a dinâmica térmica regional, permitindo o uso de modelos geoestatísticos para estimar temperaturas em áreas não monitoradas, identificar tendências sazonais e avaliar possíveis impactos ambientais e em saúde associados às mudanças climáticas.

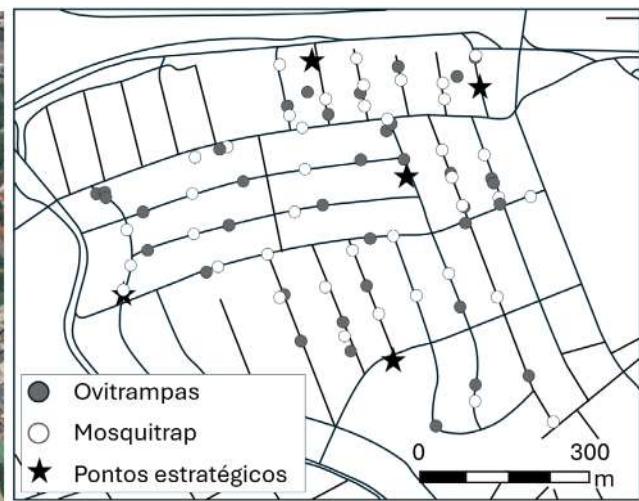
## Aplicação 2

A Figura 59 representa uma aplicação da geoestatística que tem como objetivo monitorar a presença do *Aedes aegypti*, vetor de doenças como dengue, zika e chikungunya, por meio da contagem de ovos depositados em armadilhas de oviposição instaladas em uma área urbana, mais especificamente no bairro de Higienópolis na cidade do Rio de Janeiro.

As armadilhas foram distribuídas estratégicamente ao longo do bairro para capturar padrões espaciais da oviposição do mosquito. Os dados foram coletados em base semanal, seguindo o calendário epidemiológico, durante o período de setembro de 2006 a março de 2008.

A imagem à esquerda apresenta uma visão aérea da área onde as armadilhas foram instaladas. O mapa à direita exibe a distribuição espacial das armadilhas, com diferentes formas representando os tipos de armadilhas e localização de espaço estratégico para coleta de ovos.

**Figura 59: Distribuição espacial de armadilhas para *Aedes aegypti* e mapeamento da área de estudo da pesquisa de Reis et al (2010).**



Essa abordagem possibilitam a aplicação de técnicas geoestatísticas, como a elaboração de variogramas experimentais e interpolação espacial por krigagem. Dessa forma, é possível gerar mapas preditivos que representem padrões espaciais e permitam à vigilância em saúde em identificar de áreas de maior risco para proliferação do vetor, permitindo o desenvolvimento de estratégias mais eficazes para o controle da transmissão de arboviroses.

Agora, entraremos na introdução teórica da geoestatística, abordando conceitos fundamentais e técnicas de modelagem espacial e, em seguida, aplicaremos esses conceitos a dados reais.

## *Modelagem em geoestatística*

Na modelagem geoestatística, costuma-se assumir que o fenômeno que estamos analisando (como, por exemplo, temperatura, número de casos ou concentração de poluentes) segue um **processo Gaussiano** espacial, representado por:

$$Y(\cdot) \sim PG(\mu(\cdot), c(\cdot, \cdot))$$

para a qual:

- $\mu(\cdot)$  representa a tendência do processo (efeito de primeira ordem);
- $Y(\cdot)$  e  $c(\cdot, \cdot)$  é a função de covariância associada, responsável por descrever a estrutura de dependência espacial (efeito de segunda ordem).

A tendência espacial pode ser modelada por diferentes abordagens, como:

- **Funções polinomiais** das coordenadas geográficas;
- **Funções suaves**, como thin plate splines;

Ou ainda o uso de **covariáveis ambientais ou demográficas** medidas nas mesmas localizações da variável de interesse.

## Propriedades importantes para a modelagem

- Os variogramas e covariogramas empíricos fornecem estimativas da estrutura de covariância, assumindo alguma forma de estacionaridade.
- Caso o processo não seja estacionário, os variogramas e covariogramas podem ser influenciados por efeitos de primeira ordem.
- A matriz de covariância deve ser simétrica e positiva definida, condição necessária para garantir validade estatística das estimativas.
- A matriz de covariância é geralmente estimada por modelos paramétricos, como as funções de covariância exponencial, gaussiana, Matérn, entre outras.

## Krigagem

O principal objetivo da geoestatística é prever valores de uma variável com continuidade espacial em locais onde não há observações diretas, o que é altamente relevante para a vigilância em saúde em territórios com lacunas de dados.

A técnica utilizada para realizar essa previsão é conhecida como **krigagem**. O termo krigagem deriva do geólogo sul-africano *D. G. Krige*, que desenvolveu a primeira versão do método. O método consiste em um processo de interpolação que estima o valor em uma localização não observada  $s'$ , denotado por  $\hat{Y}(s')$ .

Em sua forma mais simples, é assumido que:

$$\hat{y}(s') = \hat{\mu}(s')$$

Nesse caso, estamos considerando apenas efeitos globais, ignorando efeitos locais. Incorporar a função de covariância  $C$  permite melhorar significativamente a acurácia das previsões, levando em conta efeitos locais.

Os principais métodos de krigagem são:

- **Krigagem simples:** assume média constante conhecida;
- **Krigagem ordinária:** assume média constante, mas desconhecida;
- **Krigagem universal:** incorpora uma tendência explícita no modelo;
- **Krigagem bayesiana:** utiliza inferência probabilística, sendo apropriada quando há incerteza sobre parâmetros ou estrutura espacial.

## *Krigagem Universal*

A **krigagem universal** é um método que assume a existência de um componente de tendência no processo, ou seja:

$$\mu(s) = x(s)\beta$$

para a qual:

- $x(s)$  é um vetor de covariáveis (ou funções de base);
- $\beta$  é um vetor de parâmetros a serem estimados.

Portanto, a estimativa  $\hat{y}(s')$  é obtida por meio de uma **combinação linear ponderada** dos valores observados em locais previamente amostrados, de acordo com a equação:

$$\hat{y}(s') = \sum_{i=1}^n \lambda_i(s') y(s_i)$$

para a qual:

- $\lambda_i(s')$  representa o peso atribuído a cada observação;
- $y(s_i)$  sendo esse peso uma função da covariância espacial.

Este modelo permite incorporar tanto a **tendência global** quanto a **estrutura de dependência espacial** dos dados, tornando-se uma ferramenta essencial para a interpolação e predição geoestatística.

## *Considerações gerais sobre a Krigagem:*

- É importante destacar que a qualidade das previsões obtidas por krigagem depende da escolha adequada dos modelos utilizados para representar a tendência e o vario-grama.
- Para avaliar a eficácia do procedimento, um método amplamente utilizado é a validação cruzada, que permite verificar a precisão das previsões geradas pelo modelo.
- No processo de validação cruzada, cada valor observado  $y(s_i)$  é temporariamente removido do conjunto de dados. Em seguida, uma previsão para essa localização é realizada utilizando apenas as demais observações disponíveis.
- Como resultado, obtemos um conjunto de  $n$  erros de previsão, calculados a partir da diferença entre os valores observados e preditos.
- A análise desses erros possibilita uma avaliação detalhada do desempenho da krigagem. Se necessário, ajustes podem ser realizados nos parâmetros do vario-grama, na superfície de tendência ou em outros aspectos do modelo para melhorar a precisão das previsões.

Essa etapa é fundamental para garantir que os produtos gerados (por exemplo, mapas de risco ou superfícies interpoladas) representem adequadamente a realidade do território, permitindo que ações em saúde sejam orientadas com maior segurança e precisão.

Para ilustrar a aplicação da krigagem, vamos seguir um exemplo utilizando os dados de precipitação e, em seguida, vamos para a prática no R. Vamos lá?

## *Exemplo de aplicação*

### Modelo espacial

Neste exemplo, vamos retomar com os dados de chuva no estado do Paraná. Vamos acompanhar a estruturação do modelo da precipitação como uma função da localização geográfica (latitude e longitude), além de uma componente espacial que captura variações locais:

Podemos ajustar o seguinte modelo espacial:

$$chuva(s) = \beta_0 + \beta_1 lat(s) + \beta_2 long(s) + Z(s) + \varepsilon(s)$$

para a qual:

- $Z(\cdot)$  é um **processo Gaussiano espacial** com média zero e estrutura de correlação  $\rho(\cdot; \phi)$ , definida pela função exponencial, com variância  $\sigma^2$ .
- O termo  $\varepsilon(\cdot)$  representa o **erro de medida (efeito pepita)**, assumindo uma distribuição normal:  $\varepsilon(\cdot) \sim N(0, \tau^2)$

Esse modelo permite capturar a dependência espacial da precipitação no Paraná, considerando a influência da latitude e longitude, além de variações locais representadas pelo processo espacial  $Z(s)$ .

### Modelo Ajustado

Após o ajuste do modelo, obtemos as seguintes estimativas para os parâmetros:

$$chuva(s) = 421.8 - 0.15lat(s) - 0.39long(s) + Z(s) + \varepsilon(s)$$

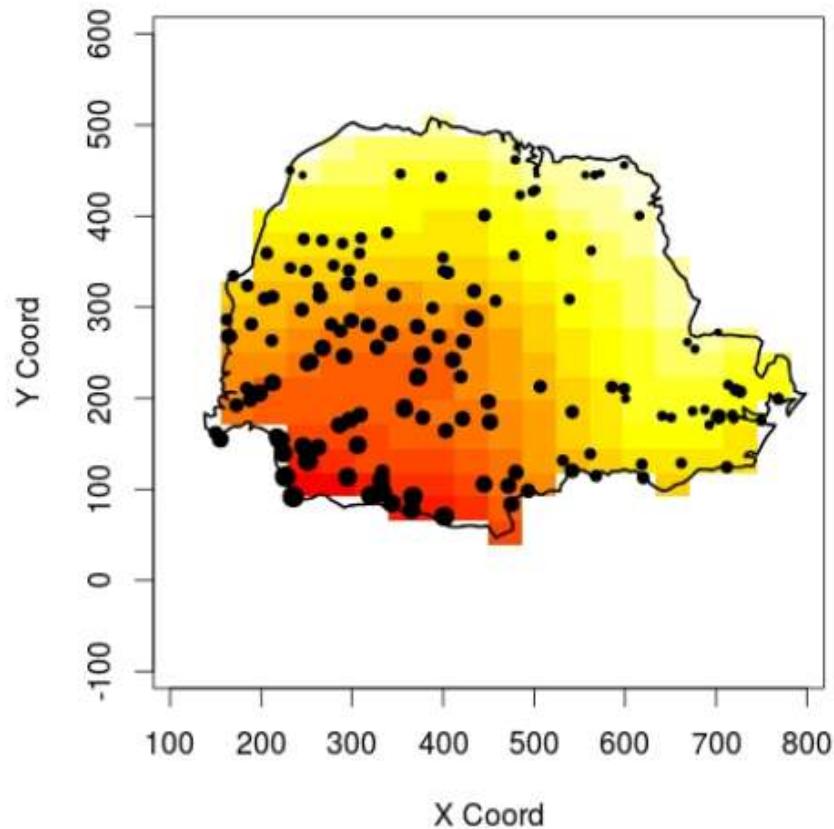
para a qual:

- $Z(s)$  é um **processo Gaussiano** com média zero e estrutura de correlação exponencial, com parâmetro de alcance  $\phi = 130$  e variância  $\sigma^2 = 685$ .
- O termo  $\varepsilon(s)$  representa o **efeito pepita**, cuja variância é  $\tau^2 = 480$ .

### Interpretação

Na Figura 60, é apresentado o mapa de interpolação espacial da precipitação. Cada ponto preto indica uma localização de observação utilizada no ajuste do modelo, ou seja, indicam as localizações onde foram feitas medições reais da chuva.

**Figura 60: Mapa interpolado de precipitação usando Krigagem Universal.**



A interpolação espacial revela um gradiente de precipitação, com valores mais elevados na região sul (representados pelas áreas em vermelho) e mais baixos na região norte (tons mais claros). Essa variação pode ser explicada por fatores geográficos, como altitude e latitude, ou por padrões climáticos regionais.

O efeito pepita ( $\tau^2$ ) indica a presença de variabilidade não explicada pelo modelo espacial, possivelmente relacionada a erros de medição ou fatores de pequena escala. A estrutura de correlação definida pelo variograma exponencial permite capturar a dependência espacial dos dados, garantindo que as estimativas interpoladas refletem padrões reais da precipitação.

Esse tipo de análise é fundamental em vigilância ambiental, planejamento de ações de saúde pública e na gestão territorial de recursos hídricos. Além disso, são essenciais para gestão de recursos hídricos e tomada de decisão em políticas públicas relacionadas ao clima e à agricultura.

## Prática em R

Nesta prática, vamos reproduzir uma análise exploratória dos dados de precipitação no estado do Paraná, conforme apresentado nos exemplos anteriores. Esses dados, disponíveis no pacote geoR, referem-se a medições de chuva coletadas em diversas estações meteorológicas distribuídas pelo estado.

A área de estudo compreende todo o território paranaense, onde as medições foram registradas ao longo de um período específico. Esses dados permitem avaliar a distribuição espacial da chuva, identificando padrões de variabilidade e tendências regionais.

Com essa abordagem, buscamos compreender melhor a distribuição espacial da precipitação no Paraná, aplicando métodos geoestatísticos como a análise de semivariogramas e a interpolação por krigagem. Esses procedimentos possibilitam estimar os valores de precipitação em locais sem observações diretas e entender a estrutura espacial da variabilidade climática no estado.

Nesta análise, aplicaremos métodos geoestatísticos para:

- i. Explorar a distribuição espacial das medições de chuva no Paraná;
- ii. Ajustar um modelo de semivariograma para descrever a estrutura de dependência espacial;
- iii. Interpolar os dados usando krigagem ordinária, gerando um mapa preditivo da distribuição da precipitação.

Ao final, obteremos um mapa interpolado da precipitação, que facilitará a visualização da variação espacial da chuva no estado. Esse tipo de análise é útil para ações de monitoramento climático, planejamento de políticas ambientais e vigilância de agravos sensíveis a variações meteorológicas, como as arboviroses.

```
# se não estiver instalado, rodar:  
install.packages("geoR")  
library(geoR)
```

O comando `library(geoR)` está carregando o pacote geoR no R. Esse pacote é utilizado para análise geoestatística.

```
# Carregando os dados
data(parana)
```

```
# Exibindo a estrutura dos dados
str(parana)
```

#### List of 4

```
$ east      , north      : num [1:143, 1:2] 403 502 556 573 702 ...
..- attr(*, "dimnames")=List of 2
... $ : NULL
... $ : chr [1:2] "east" "north"
$ data      : num [1:143] 306 201 167 163 164 ...
$ borders    : num [1:369, 1:2] 670 664 656 650 643 ...
..- attr(*, "dimnames")=List of 2
... $ : NULL
... $ : chr [1:2] "east" "north"
$ loci.paper : num [1:4, 1:2] 300 648 362 410 484 ...
- attr(*, "class")= chr "geodata"
```

- `data(parana)`: Carrega o conjunto de dados parana do pacote geoR, que contém medições de precipitação no estado do Paraná.
- `str(parana)`: Exibe a estrutura dos dados, mostrando o tipo de objeto, as variáveis armazenadas (como coordenadas e valores de precipitação) e sua organização.

```
# Resumo dos dados  
summary(parana)
```

Number of data points: 143

#### Coordinates summary

east north

min 150.1220 70.3600

max 768.5087 461.9681

#### Distance summary

min max

1.0000 619.4925

#### Borders summary

east north

min 137.9873 46.7695

max 798.6256 507.9295

#### Data summary

Min. 1st Qu. Median Mean 3rd Qu. Max.

162.7700 234.1900 269.9200 274.4106 318.2300 413.7000

#### Other elements in the geodata object

[1] "loci.paper"

- `summary(parana)`: fornece um resumo estatístico e espacial do conjunto de dados parana.

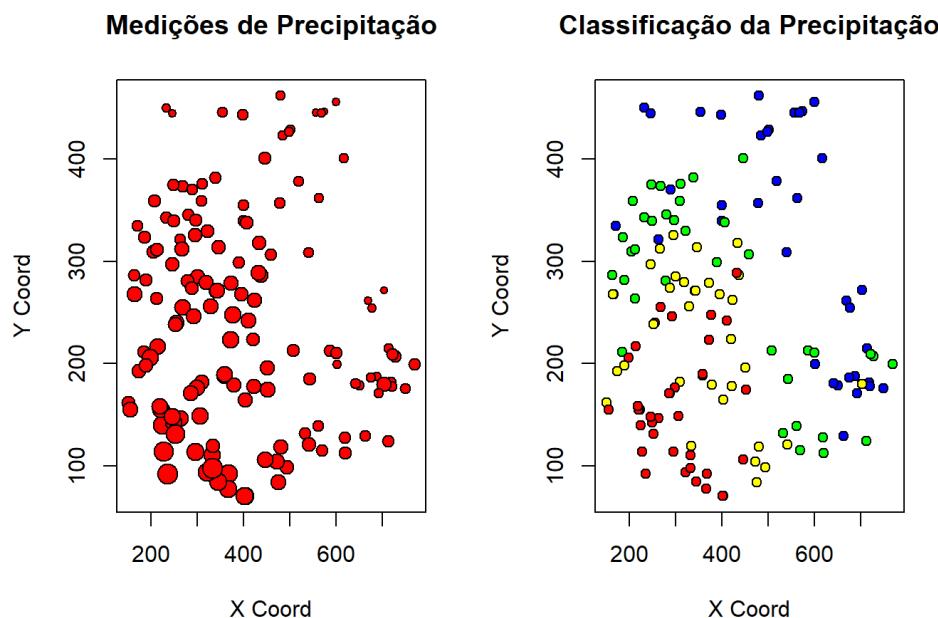
Com essa saída, observamos que temos disponível (amostradas) 143 estações pluviométricas no Paraná, incluindo a extensão espacial dos dados com coordenadas mínimas e máximas (east, north), a variação das distâncias entre os pontos e os limites geográficos da área de estudo. Além disso, exibe estatísticas descritivas da precipitação, como mínimo (162.77), mediana (269.92), média (274.41) e máximo (413.70), indicando a distribuição dos valores observados.

```
par(mfrow = c(1, 2)) # Criar layout com 2 gráficos lado a lado

# Mapa 1: Tamanho dos pontos proporcional à precipitação
plot(parana$coords, pch = 21, bg = "red", cex = parana$data / max(parana$data) * 2,
     main = "Medições de Precipitação", xlab = "X Coord", ylab = "Y Coord")

# Mapa 2: Colorir os pontos de acordo com a quantidade de precipitação
cores ← cut(parana$data, breaks = quantile(parana$data, probs = seq(0, 1,
length.out = 5), na.rm = TRUE),
            labels = c("blue", "green", "yellow", "red"), include.lowest = TRUE)

plot(parana$coords, pch = 21, bg = as.character(cores),
     main = "Classificação da Precipitação", xlab = "X Coord", ylab = "Y Coord")
```

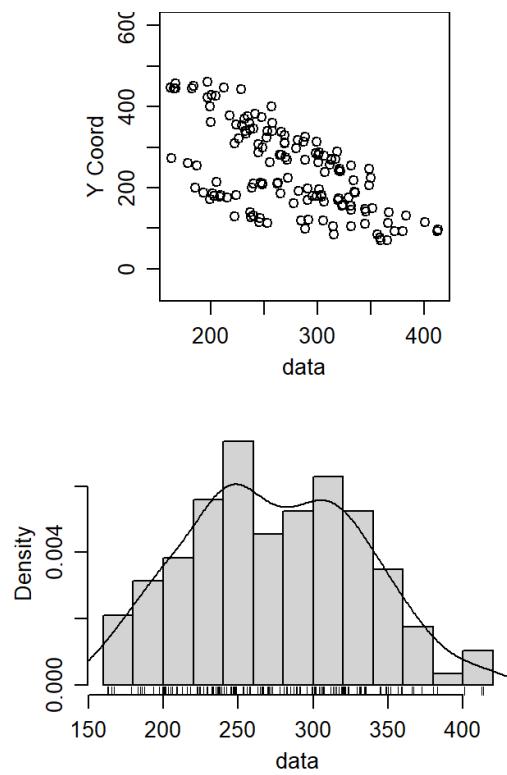
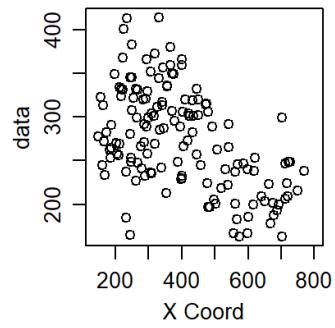
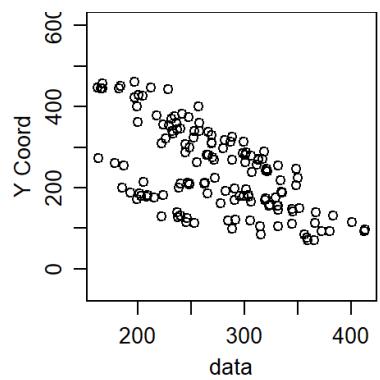
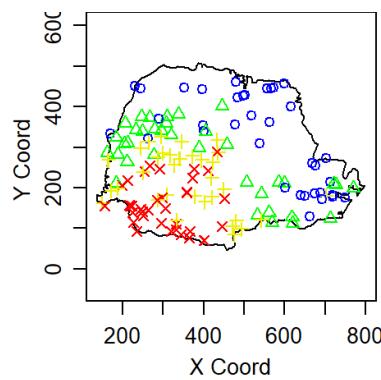


- O primeiro gráfico representa a precipitação pelo tamanho dos pontos.
- O segundo gráfico representa a precipitação pela cor dos pontos.

Os gráficos apresentam a distribuição espacial das medições de precipitação no estado do Paraná. No primeiro gráfico, o tamanho dos pontos é proporcional à quantidade de chuva registrada, permitindo identificar regiões com maior ou menor precipitação de forma intuitiva. Nota-se uma concentração de chuvas mais intensas na parte central e sul do estado, enquanto áreas no extremo norte apresentam valores menores.

No segundo gráfico, a precipitação foi categorizada em quatro classes representadas por cores: azul (valores mais baixos), verde (médios), amarelo (altos) e vermelho (valores mais elevados). Essa classificação destaca padrões regionais na distribuição das chuvas, evidenciando que áreas no centro-sul do Paraná possuem maior concentração de precipitação elevada, enquanto regiões ao norte apresentam menor volume acumulado.

```
plot(parana)
```



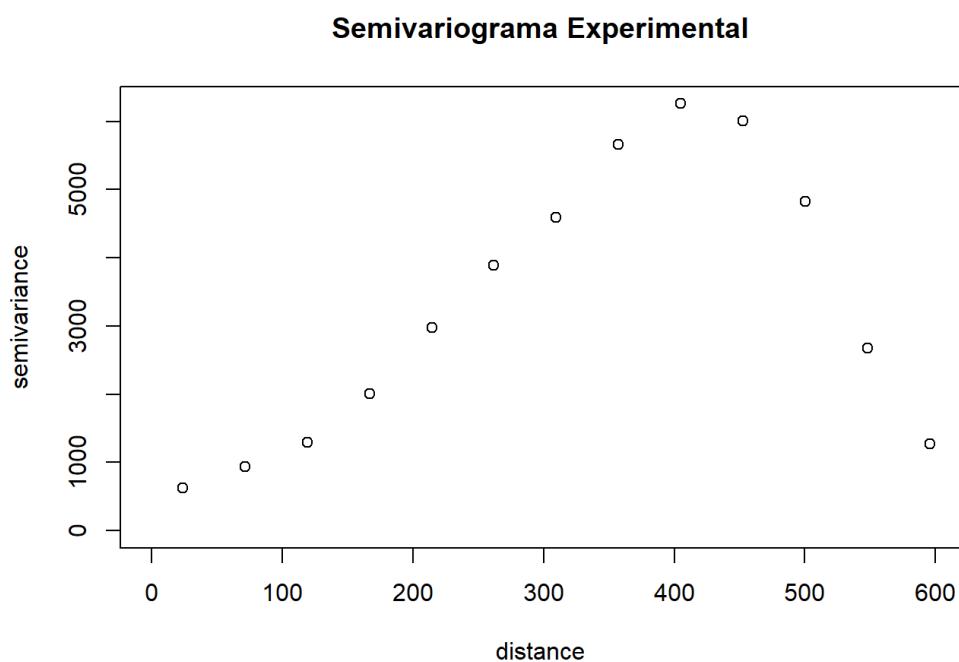
`plot(parana)`: gera uma visualização exploratória dos dados de precipitação no Paraná, mostrando diferentes aspectos espaciais e estatísticos através de gráficos e mapas.

Os gráficos exploratórios mostram a distribuição espacial das estações pluviométricas no Paraná, evidenciando variações na precipitação por região. A relação entre os valores de precipitação e as coordenadas sugere um padrão decrescente ao longo da coordenada Y, indicando possível influência da latitude na distribuição da chuva. Já o histograma revela uma distribuição aproximadamente normal, com a maioria dos valores concentrados entre 250 e 350 mm, embora haja leve assimetria. Esses padrões iniciais indicam a presença de estrutura espacial nos dados, justificando o uso de técnicas geoestatísticas para interpolação e análise.

```
# Ajustar um semivariograma experimental
variograma ← variog(parana)
```

**variog:** computing omnidirectional variogram

```
plot(variograma, main = "Semivariograma Experimental")
```



- `variograma ← variog(parana)`: Calcula o semivariograma experimental a partir dos dados de precipitação no Paraná. Um dos maiores objetivos do variograma é medir a dependência espacial entre os pontos, analisando como a variabilidade dos dados muda com a distância.
- `plot(variograma)`: Gera um gráfico do semivariograma, mostrando a relação entre a distância e a variabilidade dos valores de precipitação. Esse gráfico ajuda a identificar padrões espaciais, como o alcance da dependência espacial e a presença de estruturas de correlação.

O semivariograma experimental apresentado mostra a variação da semivariância em função da distância entre os pontos de medição da precipitação no Paraná. Inicialmente, a semivariância aumenta com a distância, indicando que medições mais distantes possuem maior diferença nos valores de chuva, até atingir um platô em torno de 400 km, o que sugere o alcance da dependência espacial — ou seja, a partir dessa distância, os valores de precipitação deixam de estar correlacionados espacialmente.

```
# Criar um semivariograma experimental e ajustar um modelo
modelo ← variofit(variograma, cov.model = "spherical")
```

`variofit`: covariance model used is spherical

`variofit`: weights used: npairs

`variofit`: minimisation function used: optim

`variofit`: searching for best initial value ... selected values:

`sigmasq phi tausq kappa`

`initial.value` "4693.86" "476.53" "0" "0.5"

`status` "est" "est" "est" "fix"

`loss value`: 6635717862.43715

Esse comando está realizando duas etapas essenciais da análise geoestatística:

- Ajuste do Modelo de Semivariograma: A função `variofit()` recebe um semivariograma experimental (variograma) e ajusta um modelo teórico para representar a dependência espacial dos dados.
- Escolha do Modelo Específico: O argumento `cov.model = "spherical"` define que o modelo esférico será usado para descrever a variação espacial da precipitação. Esse modelo assume que a correlação entre os pontos diminui até atingir um patamar (sill) em uma determinada distância (range).

```
# Criar uma grade de pontos para interpolação
grid_pred ← expand.grid(
  x = seq(min(parana$coords[, 1]), max(parana$coords[, 1]), length = 50),
  y = seq(min(parana$coords[, 2]), max(parana$coords[, 2]), length = 50)
)
```

Esses comandos criam uma grade regular de pontos dentro da área de estudo para a interpolação da krigagem.

- `seq(min, max, length = 50)`: Gera 50 valores igualmente espaçados entre os limites mínimo e máximo das coordenadas X e Y do conjunto de dados `parana`.
- `expand.grid(x, y)`: Cria uma grade de coordenadas combinando todas as possíveis posições de X e Y, formando uma malha sobre a área de estudo.

Essa grade de pontos servirá como base para calcular os valores interpolados de precipitação nesses locais ainda não amostrados.

```
# Realizar a krigagem ordinária
krigagem ← krige.conv(parana, locations = grid_pred, krige = krige,
control(obj.model = modelo))
```



krige.conv: results will be returned only for prediction locations inside the borders

krige.conv: model with constant mean

krige.conv: Kriging performed using global neighbourhood

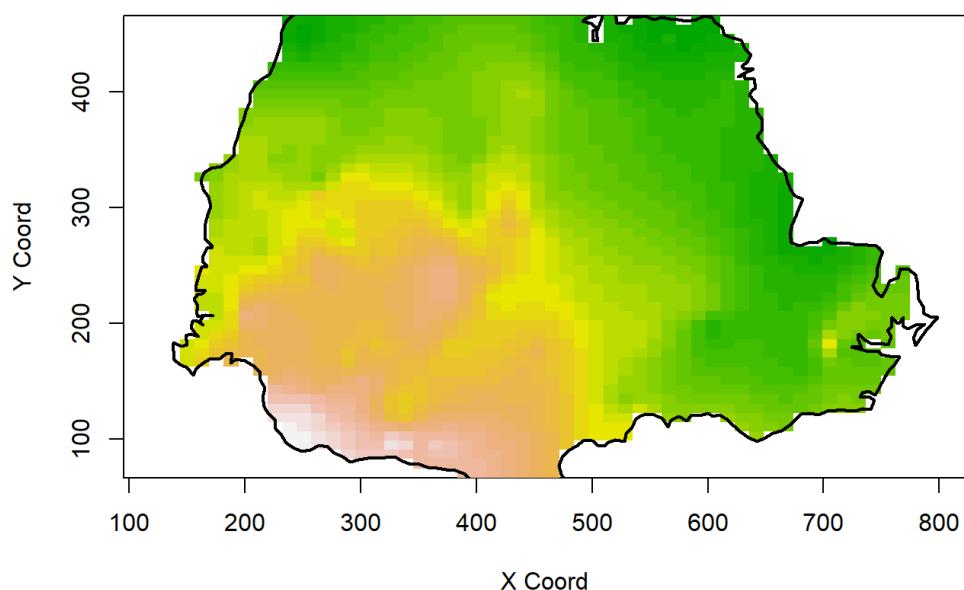
O comando `krige.conv` está realizando a krigagem ordinária com base nos dados de precipitação do Paraná.

- `grid_pred`: Grade de pontos onde a krigagem irá prever valores de precipitação.
- `krige.control(obj.model = modelo)` : Define os parâmetros da krigagem usando o modelo ajustado ao semivariograma.
- `krige.conv()`: Executa a krigagem ordinária, gerando previsões para os pontos da grade com base nos dados observados.

Esse comando estima valores de precipitação em locais não amostrados, utilizando a krigagem ordinária baseada no modelo de semivariograma ajustado.

```
# Criar o mapa de interpolação
image(krigagem, col = terrain.colors(30), main = "Interpolação por Krigagem Ordinária")
```

Interpolação por Krigagem Ordinária



A função `image` plota o mapa da interpolação gerada pela krigagem, exibindo a distribuição espacial dos valores previstos. Usando uma escala de cores (`terrain.colors(30)`) para representar as variações da variável interpolada (ex.: intensidade da chuva).

Basicamente, esse comando visualiza os resultados da krigagem como um mapa interpolado, onde as cores representam os valores estimados em cada ponto da área de estudo.

Este mapa mostra a distribuição espacial da precipitação no Paraná, gerada por krigagem ordinária. As cores verdes indicam áreas com menor precipitação, enquanto tons amarelos e marrons representam regiões com maior volume de chuva. A região sul/sudoeste parece apresentar os menores valores, enquanto o norte/leste tem níveis mais altos de precipitação, possivelmente devido a fatores climáticos e topográficos.

## *Considerações finais*

Em suma, apesar de complexos os métodos geoestatísticos são ferramentas indispensáveis para a epidemiologia moderna. Ao desvendar os padrões espaciais das doenças, especialmente na área ambiental, eles capacitam os profissionais de saúde a entender melhor os padrões espaciais dos fenômenos contínuos, como por exemplo, temperatura, poluição e chuvas associados a eventos de saúde, transformando dados em inteligência para a ação e contribuindo para a promoção da saúde e a prevenção de doenças de forma mais precisa e estratégica.

## Referências

- CHILÈS, Jean-Paul & DELFINER, Pierre; Geostatistics: Modeling Spatial Uncertainty 2012 John Wiley & Sons, Inc.
- CRESSIE, N. A. Statistics For Spatial Data. Revised edition. Iowa State University, New York: A Wiley Interscience Publication, 1993.
- DIGGLE, Peter J.; GIORGI, Emanuele. Model-based geostatistics for global public health: methods and applications. Chapman and Hall/CRC, 2019.
- DIGGLE, Peter J. & RIBEIRO JR, Paulo Justiniano; Model-based Geostatistics Series: Springer Series in Statistics, 2007. (Primeira edição 1999).
- DORMAN, Michael. Learning R for geospatial analysis. Packt Publishing Ltd, 2014.
- ISAAKS and SRIVASTAVA; An Introduction to Applied Geostatistics 1st Edition, 1989.
- REIS, Izabel Cristina et al. Relevance of differentiating between residential and non-residential premises for surveillance and control of Aedes aegypti in Rio de Janeiro, Brazil. *Acta Tropica*, v. 114, n. 1, p. 37-43, 2010.
- PEBESMA, E.J., 2004. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30: 683-691.
- TEIXEIRA, Tatiana Rodrigues de Araujo; CRUZ, Oswaldo Gonçalves. Spatial modeling of dengue and socio-environmental indicators in the city of Rio de Janeiro, Brazil. *Cadernos de Saúde Pública*, v. 27, p. 591-602, 2011.

## Módulo 5: Dados espaço-temporais

Os elementos do espaço e tempo são essenciais para diferentes estratégias de vigilância em saúde. O uso do espaço permite compreender, descrever e analisar as variações geográficas dos fenômenos de interesse, enquanto o tempo é fundamental para detectar tendências e mudanças na incidência de doenças. Essas estratégias evidenciam como nossas análises podem ser enriquecidas com técnicas voltadas à detecção oportuna de padrões epidemiológicos que acometem a população.

A combinação entre espaço e tempo potencializa nossa capacidade de identificar e antecipar cenários e orientar respostas localizadas de forma eficaz para a saúde pública. A análise espaço-temporal nasce justamente dessa articulação, ao integrar informações geográficas e temporais para revelar a dinâmica de fenômenos como a propagação de doenças, a distribuição de recursos de saúde e a evolução de condições epidemiológicas ao longo do tempo.

Esse olhar integrado não é recente e sempre esteve presente desde os primórdios da epidemiologia, assim como em outras áreas. Em setembro de 1908, o matemático Hermann Minkowski, ao comentar os avanços da física, declarou:

“Senhores, as ideias sobre espaço e tempo que eu gostaria de falar a vocês nasce do solo da física experimental. É daí que provêm sua força. A proposta é radical. De agora em diante, espaço por si só e tempo por si só devem desaparecer nas sombras, enquanto somente a união dos dois preserva sua independência.”

Tal provocação, ainda que formulada no campo da física, possui referência forte na área da saúde: compreender espaço e tempo de forma isolada pode limitar nossa capacidade analítica. Na vigilância em saúde, muitas vezes é justamente na intersecção entre os elementos “onde” e “quando” que emergem os sinais mais relevantes para ação. Apesar de técnicas que permitam a incorporação das dimensões tempo e espaço serem conhecidas e utilizadas na saúde, apenas recentemente a interação espaço-tempo tem sido considerada.

Neste módulo, você vai aprender os conceitos básicos da abordagem espaço-temporal para interpretar mapas, gráficos e relatórios que combinam informações espaciais e temporais. Compreender como a malária se espalha em regiões amazônicas, considerando rios e estradas como corredores de difusão, ou como a cobertura vacinal contra o sarampo varia entre bairros de um município, são exemplos concretos da aplicação desse olhar no cotidiano da vigilância. A abordagem espaço-temporal também permite, por exemplo, identificar áreas em períodos determinados com alta concentração de casos de dengue. Assim, é possível intensificar as ações de controle do mosquito *Aedes aegypti* de forma direcionada e em momento oportuno. Ou seja, essas análises subsidiam decisões estratégicas para a proteção da população, direcionando medidas de intervenção para onde e quando elas são mais necessárias.

Neste módulo, vamos explorar os fundamentos da análise espaço-temporal, com foco nos conceitos e estratégias de análise. Vamos lá?

## Análise exploratória espaço-temporal

A análise exploratória de dados é uma etapa fundamental em qualquer investigação em saúde pública, especialmente na vigilância epidemiológica. Quando trabalhamos com dados espaço-temporais (como, por exemplo, o número de casos de uma doença por município e semana), a análise exploratória nos permite identificar como estes dados se comportam no espaço e no tempo, identificando padrões, tendências e possíveis heterogeneidades na distribuição.

Uma boa análise exploratória antecede o uso de modelos preditivos ou inferenciais. É essencial compreender como os dados se distribuem geograficamente e evoluem ao longo do tempo. Para isso, utilizamos recursos visuais como mapas temáticos, séries temporais, gráficos comparativos e animações. Esses elementos ajudam a revelar dinâmicas complexas que, muitas vezes, não são percebidas apenas por tabelas numéricas.

Entre as principais possibilidades da análise exploratória espaço-temporal estão:

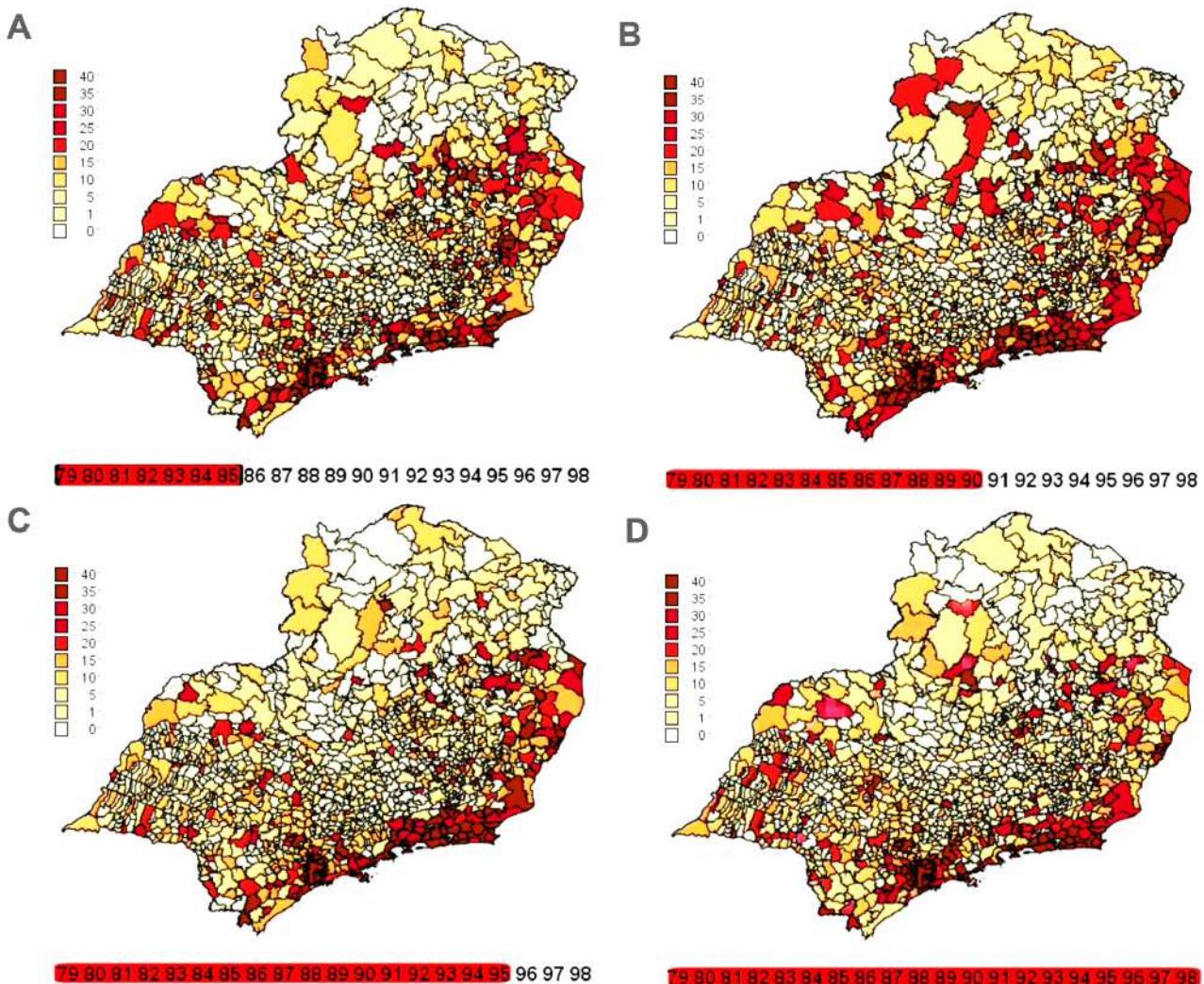
- a produção de mapas coropléticos com taxas padronizadas por área;
- a construção de séries temporais estratificadas por território;
- a visualização de mapas em facetas (para comparação entre períodos);
- o uso de animações temporais para observar o avanço de eventos no espaço; e
- a identificação visual de possíveis agrupamentos (clusters) e padrões sazonais.

Esses produtos auxiliam na formulação de hipóteses, definição de prioridades e orientação de ações em campo.

Por exemplo, considere uma situação em que é necessária uma análise da evolução dos homicídios ao longo dos anos em uma região específica. Uma análise nesse escopo permite observar a persistência e o agravamento do problema. Essa tendência pode ser evidenciada por meio de mapas coropléticos que mostram a distribuição espacial da taxa bruta de homicídios no período pesquisado, destacando áreas e momentos com altas taxas, além de sugerir a presença de possíveis clusters de risco, por meio da variação de cores e intensidade.

A Figura 61 ilustra essa distribuição em homens de 15 a 49 anos na Região Sudeste, entre 1980 e 1998, revelando como os homicídios se concentraram em regiões específicas ao longo de quatro períodos acumulados no tempo. Uma versão animada dessa figura, que mostra a evolução da taxa de homicídios ao longo do tempo, pode ser encontrada no [link](#).

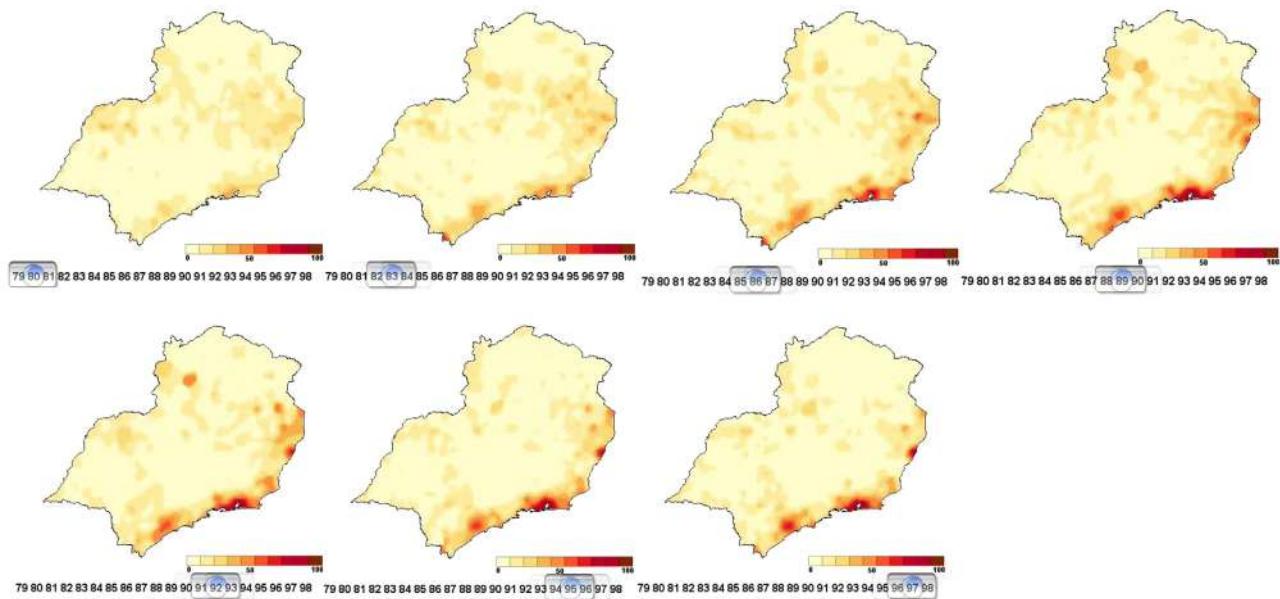
**Figura 61: Evolução da distribuição espacial da taxa bruta dos homicídios em homens de 15-49 anos no Sudeste de 1980 a 1998.**



Um outro exemplo envolve uma análise utilizando a mesma taxa de homicídios no mesmo período e região, mas de forma suavizada por uma média móvel e um kernel 2D. A suavização permite reduzir o ruído nos dados, permitindo uma melhor visualização de tendências e padrões. A média móvel é uma abordagem comum para suavizar séries temporais, enquanto o kernel 2D é usado para suavizar dados espaciais. Essa combinação, aplicada à taxa de homicídios, resulta em um mapa que não apenas mostra a distribuição espacial dos homicídios, mas também destaca áreas com alta concentração de casos ao longo do tempo.

Essa análise, portanto, não apenas revela a intensidade dos homicídios em diferentes regiões, mas também fornece insights sobre como esses padrões mudaram ao longo do tempo e como eles podem estar relacionados a fatores sociais e econômicos locais. A Figura 62 ilustra essa suavização, mostrando a evolução da taxa de homicídios em homens de 15 a 49 anos na Região Sudeste entre 1980 e 1998, por triênios, e também possui uma versão animada no [link](#).

**Figura 62: Evolução da distribuição da taxa suavizada no tempo (média móvel) e no espaço (Kernel 2d) de homicídios em homens de 15-49 anos no Sudeste de 1980 a 1998.**



As inspeções visuais na análise exploratória permitem que padrões sejam revelados e orientam a formulação de hipóteses sobre a distribuição do agravo avaliado. Mas, em seguida, vamos avançar um pouco utilizando algumas técnicas que estão entre a análise exploratória e a modelagem estatística: a detecção de clusters. Além disso, veremos alguns exemplos de implementação dessa técnica.

## *Detecção de clusters espaço-temporais de doenças*

A análise de clusters constitui uma metodologia essencial para compreender padrões espaço-temporais de ocorrência de doenças, surtos e outros fenômenos. Essa metodologia tem como objetivo identificar se os eventos de saúde estão distribuídos aleatoriamente no espaço e no tempo, ou se há concentrações incomuns que sugerem um padrão não aleatório. Essas concentrações, quando estatisticamente significativas, são denominadas clusters ou aglomerados.

Formalmente, um cluster é um agrupamento de eventos, como casos de uma doença, que ocorre em uma área geográfica e período de tempo com frequência maior (ou menor) do que o esperado pelo acaso. Isso significa que há indícios de que o risco naquela região e período é diferente do restante da área de estudo, justificando atenção especial por parte das equipes de vigilância.

O principal objetivo da detecção de clusters é identificar áreas de risco elevado para um fenômeno de saúde. Isso permite a priorização de recursos e a atuação direcionada. Por exemplo, considere uma análise de cobertura vacinal. Se os dados de imunização revelam um cluster de crianças não vacinadas contra o sarampo em determinada vizinhança, equipes podem ser mobilizadas para ações de busca ativa e campanhas locais. Da mesma forma, identificar regiões com risco significativamente menor do que o esperado pode revelar fatores de proteção ou o êxito de políticas locais.

Entre os métodos disponíveis, destaca-se a estatística de varredura (scan), amplamente utilizada por meio do software SaTScan™. Essa técnica é capaz de localizar e dimensionar os clusters espaço-temporais de forma mais precisa do que métodos de autocorrelação global, como o I de Moran ou o C de Geary, que medem associação, mas não apontam onde estão os agrupamentos.

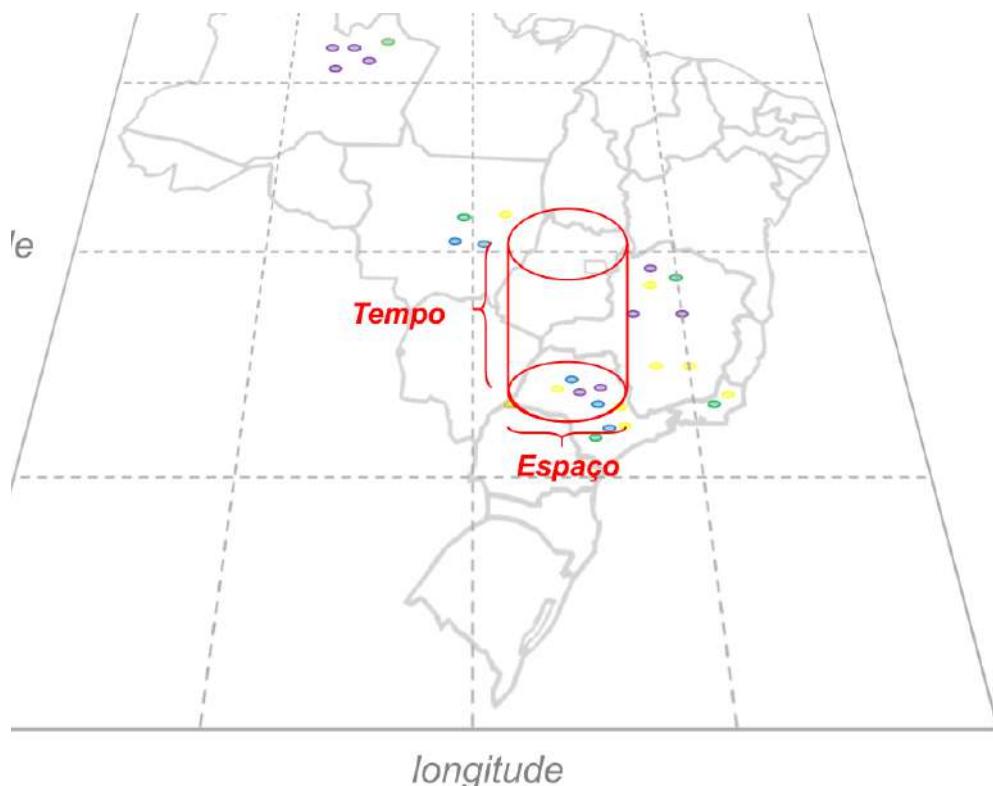
A seguir, vamos revisar os conceitos fundamentais por trás da estatística de varredura e como ela pode ser aplicada na detecção de clusters em contextos reais da saúde pública.

## A estatística scan (SaTScan)

A estatística de varredura, ou scan statistic, desenvolvida por Martin Kulldorff e colaboradores, é uma técnica amplamente utilizada para detectar e avaliar aglomerados espaciais, temporais ou espaço-temporais de doenças. Para sua aplicação em dados espaciais, frequentemente a informação de uma área (como um município ou bairro) é resumida a um único ponto geográfico, como o seu centroide (o centro geométrico da área). Isso simplifica a análise, permitindo que o algoritmo faça uma varredura na região de estudo em busca de clusters.

Para detectar aglomerados espaço-temporais, a estatística de varredura funciona como se um cilindro imaginário se movesse pela área de estudo. A base desse cilindro se move no espaço (cobrindo diferentes conjuntos de localidades) e a altura do cilindro se move no tempo (cobrindo diferentes períodos) (Figura 63). O algoritmo avalia inúmeros cilindros de variados tamanhos (raios da base e alturas) para encontrar áreas e períodos onde a ocorrência da doença é significativamente mais provável do que o esperado sob a hipótese nula de que não existem clusters. Essa hipótese é testada usando um teste da razão de verossimilhança, e a significância estatística dos clusters encontrados é geralmente avaliada por meio de simulações de Monte Carlo. O software SaTScan™, gratuito e amplamente utilizado, implementa essa metodologia e pode ser integrado com outras ferramentas, como o software R através do pacote rsatscan, facilitando a automatização das análises e a integração com outros dados de vigilância.

**Figura 63: Representação do algoritmo de varredura espacial (scan) para detecção de clusters espaço-temporais.**



Quando lidamos com dados de contagem, como o número de casos de dengue por setor censitário ou o número de óbitos por COVID-19 por município, os modelos de probabilidade mais comuns usados no SaTScan são o de Poisson e o de Permutação Espaço-Tempo. O modelo de Poisson assume que os casos seguem essa distribuição sob a hipótese nula, enquanto o modelo de Permutação Espaço-Tempo, embora compartilhe a ideia básica, foca exclusivamente na interação espaço-tempo, não necessita de dados populacionais (trabalha apenas com os casos) e assume uma distribuição Hipergeométrica para os casos sob a hipótese nula. A escolha do modelo adequado dependerá da natureza dos dados e dos objetivos da análise, sendo fundamental para a correta identificação de áreas prioritárias para a saúde pública.

Vamos agora acompanhar dois exemplos de implementação de detecção de clusters espaço-temporais. Um deles é um estudo de caso sobre a análise de leptospirose no Rio de Janeiro, e o outro é um estudo sobre a dinâmica de surtos de dengue, chikungunya e zika na mesma cidade. Esses exemplos ilustram como a detecção de clusters pode ser aplicada na prática para informar decisões em saúde pública.

## *Exemplos de aplicação*

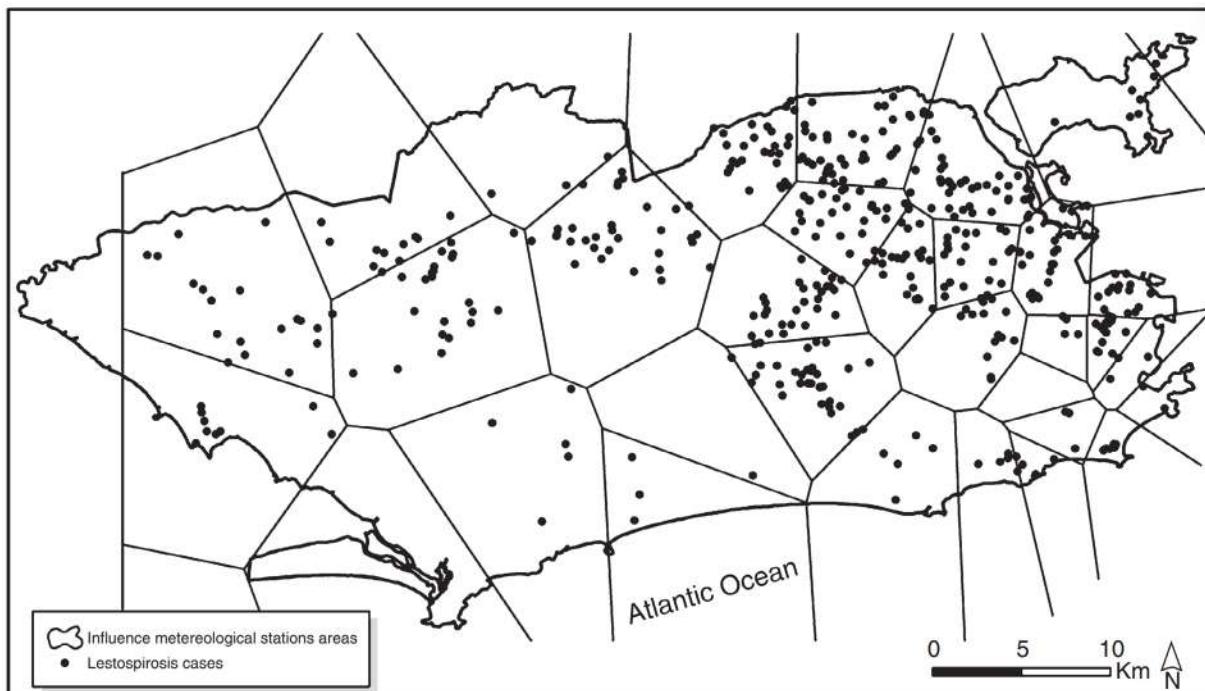
### **Análise espaço-temporal da leptospirose no município do Rio de Janeiro**

Este estudo de caso aborda a análise espaço-temporal da leptospirose urbana no município do Rio de Janeiro, no período de 1997 a 2002. TASSINARI et al. (2008) utilizaram detecção de clusters para identificar agrupamentos de casos da doença associados a fatores ambientais e sociodemográficos. Por meio da aplicação da estatística de varredura espacial (scan statistics) e de modelos lineares generalizados mistos, investigou-se a ocorrência de aglomerações e seus determinantes, destacando-se a chuva intensa como principal fator de risco para a formação dos clusters.

Foram utilizados dados de casos notificados à Secretaria Municipal de Saúde, informações socioeconômicas dos setores censitários e dados meteorológicos sobre precipitação. Trata-se, portanto, de um exemplo prático e relevante de análise espaço-temporal, que ilustra o uso integrado de dados epidemiológicos, ambientais e geográficos para compreender e controlar doenças infecciosas em áreas urbanas.

A Figura 64 apresenta a distribuição dos casos de leptospirose no município e os polígonos de Voronoi associados às 32 estações meteorológicas. Essa representação espacial divide o território em regiões associadas a cada estação, permitindo observar como os casos se distribuem em relação aos padrões de precipitação, e identificar áreas mais vulneráveis à ocorrência da doença.

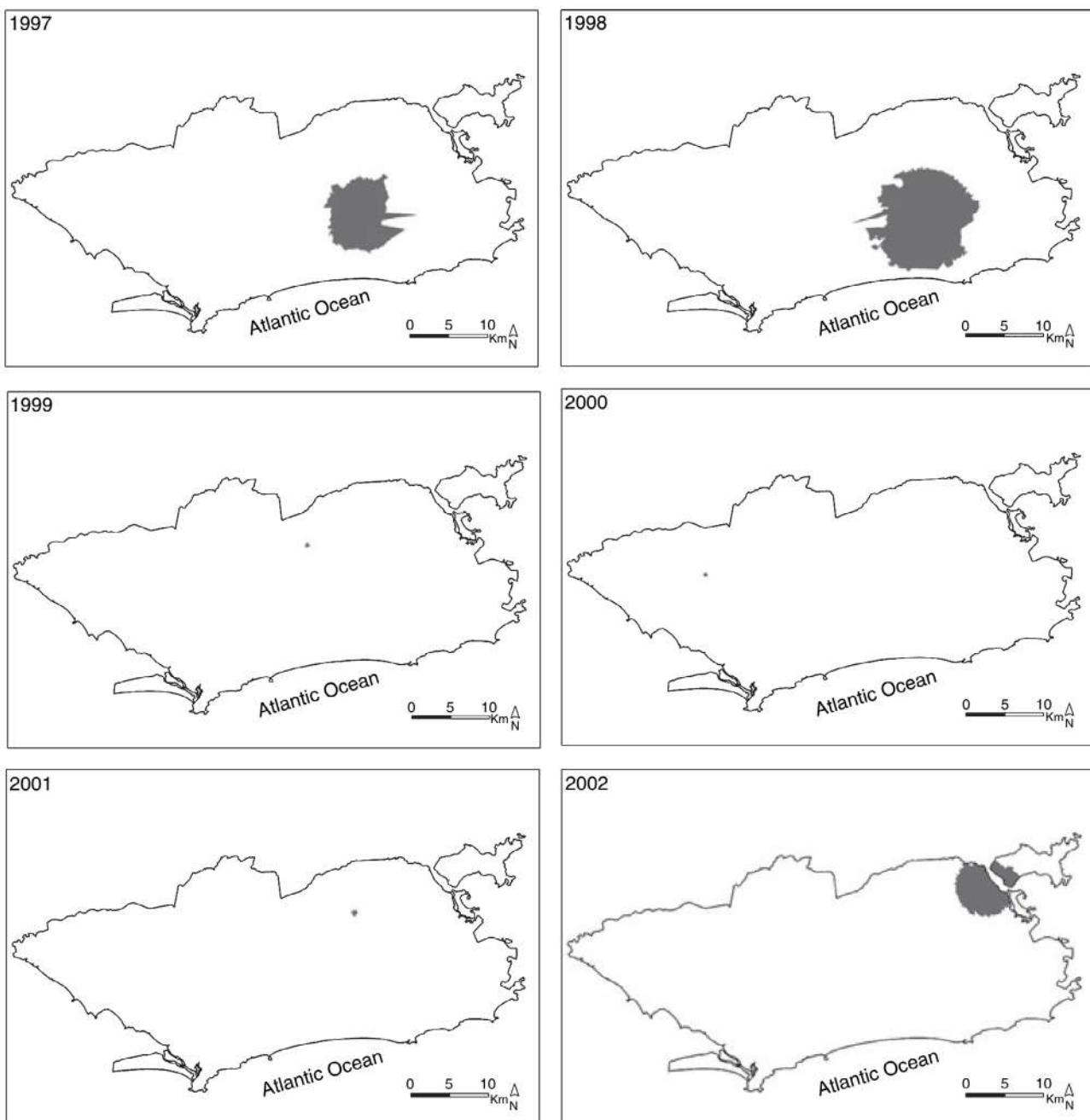
**Figura 64: Distribuição dos casos de leptospirose e polígonos de Voronoi associados a cada uma das 32 estações meteorológicas no Rio de Janeiro, Brasil.**



Fonte: TASSINARI, W. S. et al. Tropical Medicine & International Health, 13(4):503–512, 2008.

A Figura 65 mostra os clusters espaço-temporais identificados ao longo do período analisado. Observa-se que os eventos de 1999, 2000 e 2001 abrangeram um número menor de localidades, enquanto os de 1997, 1998 e 2002 envolveram áreas mais extensas da cidade. Esses clusters concentraram-se, predominantemente, em regiões de favelas, caracterizadas por maior vulnerabilidade social e suscetibilidade a alagamentos.

**Figura 65: Distribuição dos clusters de casos de leptospirose identificados no município do Rio de Janeiro de 1997 a 2002.**



Fonte: TASSINARI, W. S. et al. Tropical Medicine & International Health, 13(4):503–512, 2008.

O estudo evidenciou o potencial da análise espaço-temporal na detecção precoce de surtos e no desenvolvimento de sistemas de alerta que apoiem intervenções direcionadas em áreas urbanas vulneráveis. A identificação de clusters permite o direcionamento eficiente de ações de controle, especialmente em regiões densamente povoadas e suscetíveis a enchentes. Além disso, a associação dos agrupamentos com eventos climáticos extremos, como chuvas intensas, reforça a importância de incorporar variáveis ambientais na vigilância epidemiológica.

Na sequência, veremos um segundo estudo de caso que aplica a análise espaço-temporal para investigar a dinâmica conjunta de surtos de dengue, chikungunya e zika no município do Rio de Janeiro, entre 2015 e 2016. Este exemplo é particularmente interessante por abordar doenças transmitidas por um mesmo vetor, explorando como elas se espalham e interagem em um ambiente urbano compartilhado.

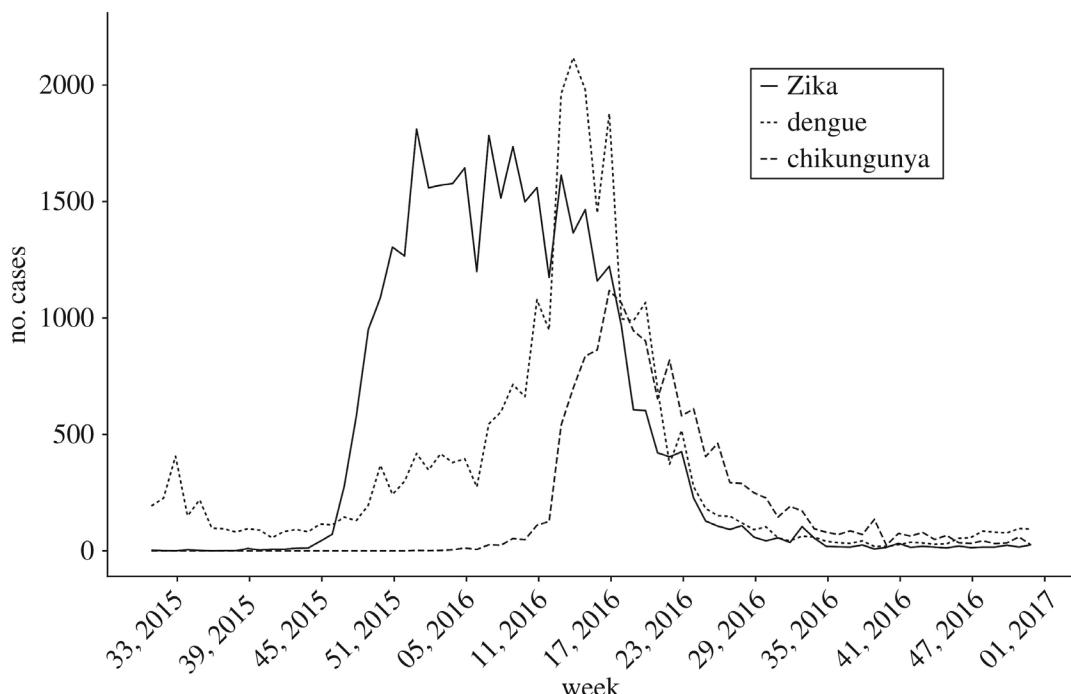
### Análise de surtos de dengue, chikungunya e zika no Rio de Janeiro

Neste estudo, FREITAS et al. (2019) utilizaram dados de casos notificados de dengue, chikungunya e zika no município do Rio de Janeiro, entre 2015 e 2016, para investigar a dinâmica espaço-temporal dos surtos dessas arboviroses. O objetivo foi compreender o comportamento das doenças dengue, chikungunya e zika durante uma situação de tripla epidemia na cidade do Rio de Janeiro.

Foram geocodificados os endereços dos casos notificados pela Secretaria Municipal de Saúde, possibilitando a análise em escala intraurbana. A investigação ocorreu em duas etapas: na primeira, aplicou-se a estatística de varredura de Kulldorff para detectar clusters de alta incidência em espaço e tempo, separadamente para cada doença. Em seguida, os autores analisaram a sobreposição e interação entre os clusters de dengue, chikungunya e zika, buscando compreender os motivos das diferenças observadas.

A Figura 66 apresenta a distribuição temporal dos casos no município. As curvas epidêmicas das três doenças mostram que, embora todas tenham apresentado alta incidência entre abril e junho de 2016, seus picos não coincidiram exatamente. Os casos de zika começaram a diminuir em março de 2016, enquanto os de dengue e chikungunya ainda estavam em ascensão. Dengue e zika já estavam em circulação no final de 2015, ao passo que os primeiros registros significativos de chikungunya só ocorreram a partir de março de 2016. Após maio, houve queda nas notificações das três doenças.

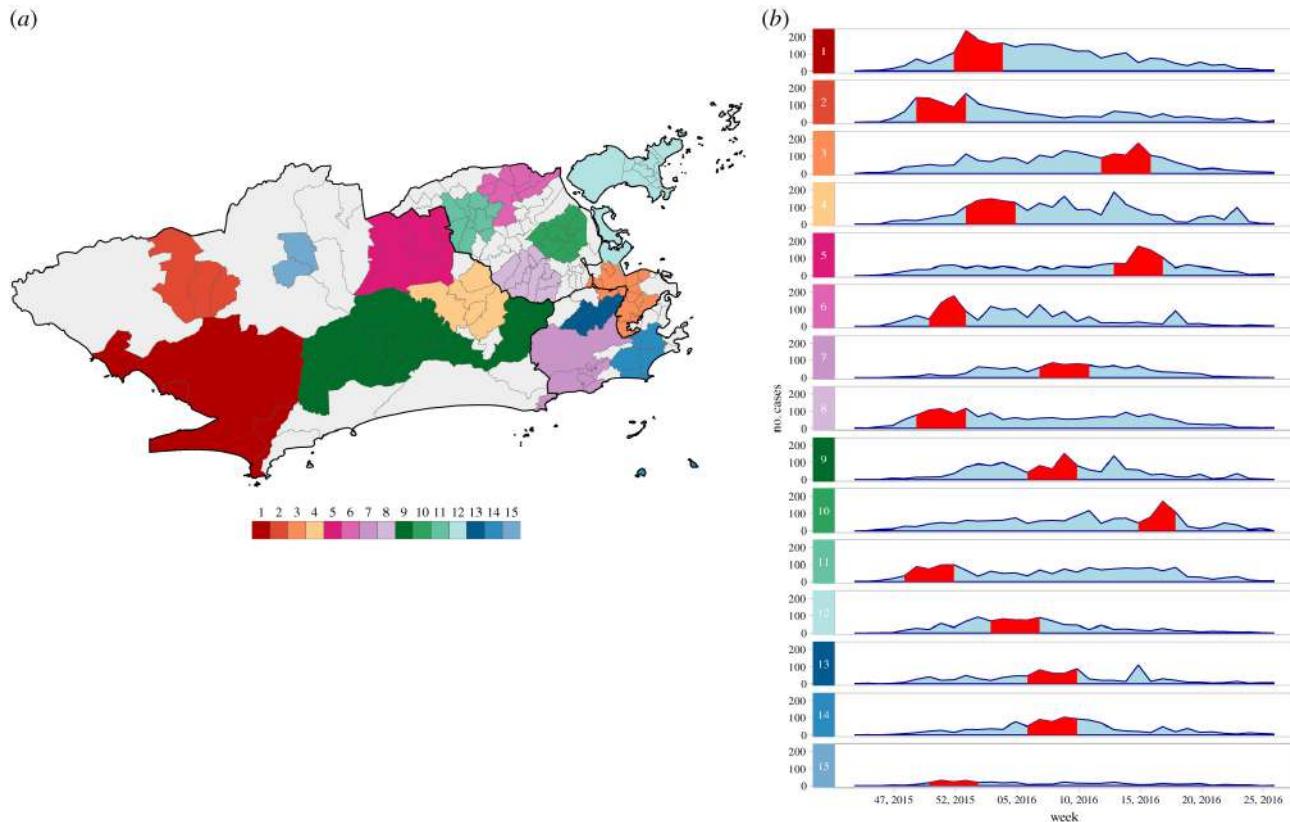
**Figura 66: Número de casos reportados de dengue (linha pontilhada), chikungunya (linha tracejada) e zika (linha sólida) entre 2 de agosto de 2015 e 31 de dezembro de 2016, município do Rio de Janeiro.**



Fonte: FREITAS, L. P. et al. Proceedings of the Royal Society B. 286:20191867, 2019.

A Figura 67 mostra os clusters espaço-temporais de zika identificados no período analisado. O mapa (a) indica onde os grupos com maior incidência se concentraram, enquanto os gráficos (b) apresentam os períodos em que esses agrupamentos foram detectados, representados pelos intervalos em vermelho. Nota-se que os clusters de zika estiveram amplamente distribuídos pelo município, com destaque para o período entre novembro de 2015 e maio de 2016.

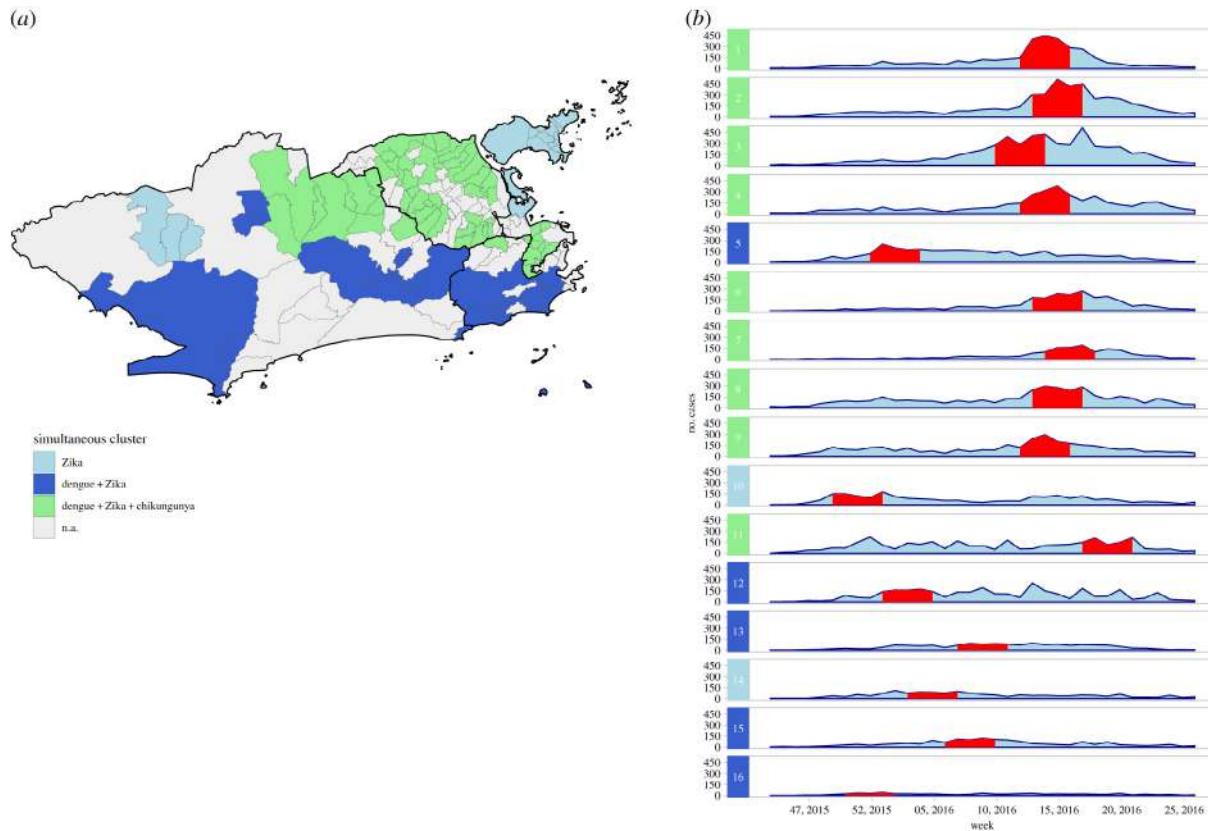
**Figura 67: Clusters de casos de zika (a) e distribuição temporal dos casos de zika por cluster (b), entre as semanas epidemiológicas 31 de 2015 e 52 de 2016, cidade do Rio de Janeiro, Brasil.**



Fonte: FREITAS, L. P. et al. Proceedings of the Royal Society B. 286:20191867, 2019.

A Figura 68 apresenta uma abordagem multivariada, utilizando a estatística de varredura para múltiplos conjuntos de dados simultaneamente. Foram detectados 16 clusters multivariados, dos quais nove apresentaram simultaneidade de casos de dengue, chikungunya e zika; cinco indicaram sobreposição entre dengue e zika; e dois revelaram agrupamentos apenas de zika. Essa análise permite compreender como diferentes doenças interagem em um mesmo território e período.

**Figura 68: Clusters de dengue, chikungunya e zika detectados usando a estatística de varredura multivariada (a) e distribuição temporal de casos por cluster (b), entre as semanas epidemiológicas 31 de 2015 e 52 de 2016, município do Rio de Janeiro.**



Fonte: FREITAS, L. P. et al. Proceedings of the Royal Society B. 286:20191867, 2019.

Os resultados indicam que, embora as três arboviroses tenham circulado amplamente no município, a zika apresentou maior incidência e maior número de bairros com transmissão significativa. Dengue e chikungunya também se espalharam de forma expressiva, mas com intensidade e temporalidade distintas. Fatores como competição entre vírus, momento de introdução das doenças, esgotamento da população suscetível e mudanças no comportamento da população (como intensificação do controle vetorial após o aumento de casos) podem explicar as diferenças nos padrões de agrupamento observados.

Esse estudo exemplifica como a análise espaço-temporal, associada a métodos estatísticos robustos, pode oferecer insights valiosos sobre a dinâmica de doenças em contextos urbanos complexos. Casos como esse reforçam a importância de capacitar as equipes de vigilância para interpretar e utilizar esses resultados no planejamento de ações mais eficientes.

Na próxima seção, vamos abordar um tema mais denso, porém introdutório, com exemplos que tornarão o conteúdo mais acessível.

## *Modelagem estatística espaço-temporal*

Após a análise exploratória, que nos permite visualizar e compreender padrões básicos nos dados de saúde, é comum avançarmos para etapas de modelagem estatística. A modelagem espaço-temporal tem como objetivo não apenas descrever esses padrões, mas também quantificar relações com fatores de risco, identificar efeitos estruturais e realizar previsões em áreas e períodos ainda não observados.

Existem diversas estratégias e famílias de modelos estatísticos que podem ser aplicadas à análise espaço-temporal. Esta é uma área em constante evolução, com métodos cada vez mais sofisticados. Um exemplo bastante utilizado é o dos Modelos Hierárquicos Bayesianos, que permitem incorporar múltiplas fontes de incerteza e variabilidade, acomodando estrutura espacial e dependência temporal de forma conjunta.

## *Considerações finais*

Considerações finais Na epidemiologia moderna, a compreensão de que as doenças ocorrem em padrões complexos influenciados por onde e quando acontecem, é fundamental. Nesse contexto, os métodos exploratórios e de modelagem de processos espaço-temporais emergem como indispensável, permitindo que epidemiologistas, técnicos em saúde pública e gestores não apenas reajam a surtos, mas também os antecipem e planejem intervenções de forma estratégica e eficaz.

A análise exploratória de dados espaço-temporais e permite a identificação de aglomerados (Clusters), a visualização da disseminação do agravo através de mapas dinâmicos e séries temporais, que possibilitam visualizar a trajetória de uma epidemia, compreendendo sua velocidade, direção e alcance geográfico, e cruzar mapas de incidência de doenças com dados ambientais (clima, vegetação), socioeconômicos (renda, saneamento) ou de infraestrutura (proximidade a hospitais). Os métodos exploratórios ajudam a formular hipóteses sobre os fatores de risco e determinantes sociais da saúde que podem estar impulsionando a transmissão.

Modelos espaço-temporais podem incorporar diversas variáveis como padrão sazonal, crescimento populacional, cobertura vacinal, intervenções de saúde, etc. permitindo entender fator contribui para a disseminação de doenças. Utilizando dados históricos, os modelos podem prever cenários futuros, estimando o número provável de casos, hospitalizações e óbitos em diferentes localidades auxiliando na previsão pro demandas do serviço de saúde, medicamentos, vacinas, etc. Com os modelos podemos gerar mapas de risco preditivo, que estimam a probabilidade de ocorrência de uma doença em áreas onde talvez ainda não haja notificações, com base em suas características e na situação de seus vizinhos.

Assim a utilização de técnicas exploratórias e a modelagem espaço-temporal representam um avanço significativo para a saúde pública.

## Referências

- BERNARDINELLI, L.; CLAYTON, D.; PASCUTTO, C.; MONTOMOLI, C.; GHISLANDI, M.; SONGINI, M. Bayesian analysis of space-time variation in disease risk. *Statistics in Medicine*, v. 14, n. 21-22, p. 2433-2443, 1995. DOI: 10.1002/sim.4780142112
- CROMLEY, E. K.; MCLAFFERTY, S. L. *GIS and public health*. 2. ed. New York: Guilford Press, 2012.
- DIGGLE, P. J.; CHETWYND, A. G.; HÄGGKVIST, R.; MORRIS, S. E. Second-order analysis of space-time clustering. *Statistical Methods in Medical Research*, v. 4, n. 2, p. 124-136, 1995. DOI: 10.1177/096228029500400203.
- FREITAS, L. P.; CRUZ, O. G.; LOWE, R.; CARVALHO, M. S. Space-time dynamics of a triple epidemic: dengue, chikungunya and Zika clusters in the city of Rio de Janeiro. *Proceedings of the Royal Society B: Biological Sciences*, v. 286, n. 1906, 2019. DOI: 10.1098/rspb.2019.1867.
- KNORR-HELD, L. Bayesian modelling of inseparable space-time variation in disease risk. *Statistics in Medicine*, v. 19, n. 17-18, p. 2555-2567, 2000. DOI: 10.1002/1097-0258(20000915/30)19:17/18<2555::AID-SIM587>3.0.CO;2-#.
- KNORR-HELD, L.; BESAG, J. Modelling risk from a disease in time and space. *Statistics in Medicine*, v. 17, n. 18, p. 2045-2060, 1998. DOI: 10.1002/(SICI)-1097-0258(19980930)17:18<2045::AID-SIM943>3.0.CO;2-P.
- KULLDORFF, M. SaTScan™ User Guide for version 10, jul. 2022. 2022. Disponível em: <http://www.satscan.org/techdoc.html>.
- MORAGA, P. Geospatial health data: modeling and visualization with R-INLA and Shiny. Boca Raton: Chapman & Hall/CRC, 2019. Disponível em: <https://www.paulamoraga.com/book-geospatial/>.
- PORTA, M. (ed.). *A dictionary of epidemiology*. 6. ed. New York: Oxford University Press, 2014.
- TASSINARI, W. S. et al. Detection and modelling of case clusters for urban leptospirosis. *Tropical Medicine & International Health*, v. 13, n. 4, p. 503-512, abr. 2008. Disponível em: <https://www.arca.fiocruz.br/bitstream/icict/12207/2/Tassiniari%20WS%20Detection%20and%20modeling....pdf>.
- WALLER, L. A.; CARLIN, B. P.; XIA, H.; GELFAND, A. E. Hierarchical spatio-temporal mapping of disease rates. *Journal of the American Statistical Association*, v. 92, n. 438, p. 607-617, 1997. DOI: 10.1080/01621459.1997.10474012.



UFGM

