



Estatística Espacial

Trabalho 1

Luiz Henrique Barretta Francisco - 202100155302

outubro/2025

Introdução

A Lei nº 11.340/2006, popularmente conhecida como Lei Maria da Penha, representa um marco na legislação brasileira no combate à violência doméstica e familiar contra a mulher. Para garantir a celeridade e a especificidade no tratamento desses casos, o Conselho Nacional de Justiça (CNJ) padronizou diversas classes processuais, dentre as quais se destaca a de Medidas Protetivas de Urgência (Lei Maria da Penha), sob o código 1268. Este trabalho se debruça sobre a análise espacial dos processos pertencentes a essa classe, utilizando uma base de dados extraída dos sistemas de Business Intelligence (BI) do Tribunal de Justiça do Paraná (TJPR). Em estrito respeito à Lei Geral de Proteção de Dados (LGPD), o conjunto de dados foi rigorosamente anonimizado, removendo-se todas as informações de identificação pessoal das partes e mantendo-se apenas os registros de endereço das vítimas (partes promoventes) para a análise geográfica.

A escolha desta classe processual é metodologicamente estratégica. Em casos de medidas protetivas, o endereço residencial da vítima é frequentemente o local onde o risco ou a violência ocorre, tornando-se um forte indicativo geográfico do fenômeno. Isso difere da maioria dos outros tipos de crime, nos quais o endereço da ocorrência não é um campo parametrizado nos sistemas judiciais, exigindo a análise manual e inviável de milhares de Boletins de Ocorrência policiais. Dessa forma, este conjunto de dados oferece uma oportunidade única para estudar a distribuição espacial da violência doméstica com base em dados estruturados e confiáveis.

1. Preparação dos Dados e Georreferenciamento

1.1. Carga e Filtragem Inicial

Primeiramente, os dados são carregados de uma planilha Excel. Em seguida, são filtrados para conter apenas os registros do ano de 2024 e cujos CEPs pertencem à faixa do estado do Paraná.

```
# Carregamento e Instalação Automática de Pacotes
pacotes_necessarios <- c("readxl", "dplyr", "geocodebr", "geobr", "sf", "geoR", "spatstat",
                         "sidrar", "parallel", "ggplot2", "lubridate", "viridis", "tictoc")
for (pacote in pacotes_necessarios) {
  if (!require(pacote, character.only = TRUE)) {
    install.packages(pacote, dependencies = TRUE)}
  library(pacote, character.only = TRUE)}

# Carga e Filtro dos Dados de Ocorrências
dados_originais <- read_excel("processos_violencia_domestica.xlsx")
df <- dados_originais %>%
  rename(Data_Distribuicao = `Data_Distribuicao`) %>% filter(!is.na(CEP),
    as.numeric(CEP) >= 80000000 & as.numeric(CEP) <= 87999999,
    format(as.Date(Data_Distribuicao), "%Y") == "2024")

cat(sprintf("Total de registros após a filtragem: %d\n", nrow(df)))

## Total de registros após a filtragem: 20255
```

1.2. Georreferenciamento dos Endereços

Utiliza-se a biblioteca `geocodebr` para converter os endereços em coordenadas. Esta abordagem é eficiente por usar uma base de dados local do IBGE (CNEFE) e permitir processamento paralelo.

```

# Geocodificação
df_para_geocodificar <- df %>% mutate(Estado = "PR")
mapeamento_campos <- definir_campos(logradouro = "Logradouro", numero = "Número",
                                       localidade = "Bairro", cep = "CEP",
                                       municipio = "Cidade", estado = "Estado")

tic("Tempo de execução (geocodebr):")
df_geo <- geocode(enderecos = df_para_geocodificar, campos_endereco = mapeamento_campos,
  n_cores = detectCores() - 1, resolver_empates = TRUE)
toc()

## Tempo de execução (geocodebr):: 20.66 sec elapsed

```

2. Análise Exploratória de Dados Espaciais

2.1. Análise de Precisão

Um primeiro passo é analisar a qualidade do georreferenciamento, verificando a distribuição dos níveis de precisão retornados pela função `geocode`.

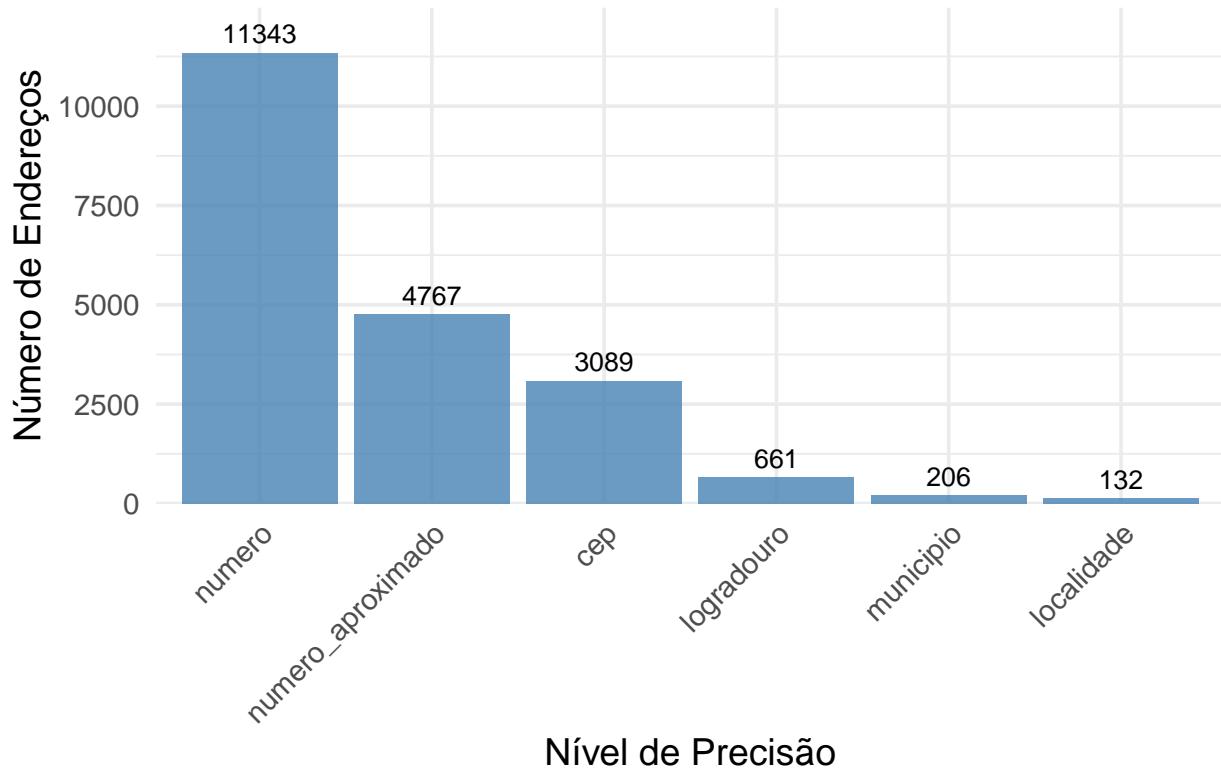
```

# Precisão da geocodificação
df_precisao_contagem <- df_geo %>%
  filter(!is.na(precisao)) %>%
  count(precisao, sort = TRUE)

ggplot(df_precisao_contagem, aes(x = reorder(precisao, -n), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue", alpha = 0.8) +
  geom_text(aes(label = n), vjust = -0.5, size = 3.5) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  labs(title = "Distribuição da Precisão da Geocodificação",
       x = "Nível de Precisão", y = "Número de Endereços") +
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

Distribuição da Precisão da Geocodificação



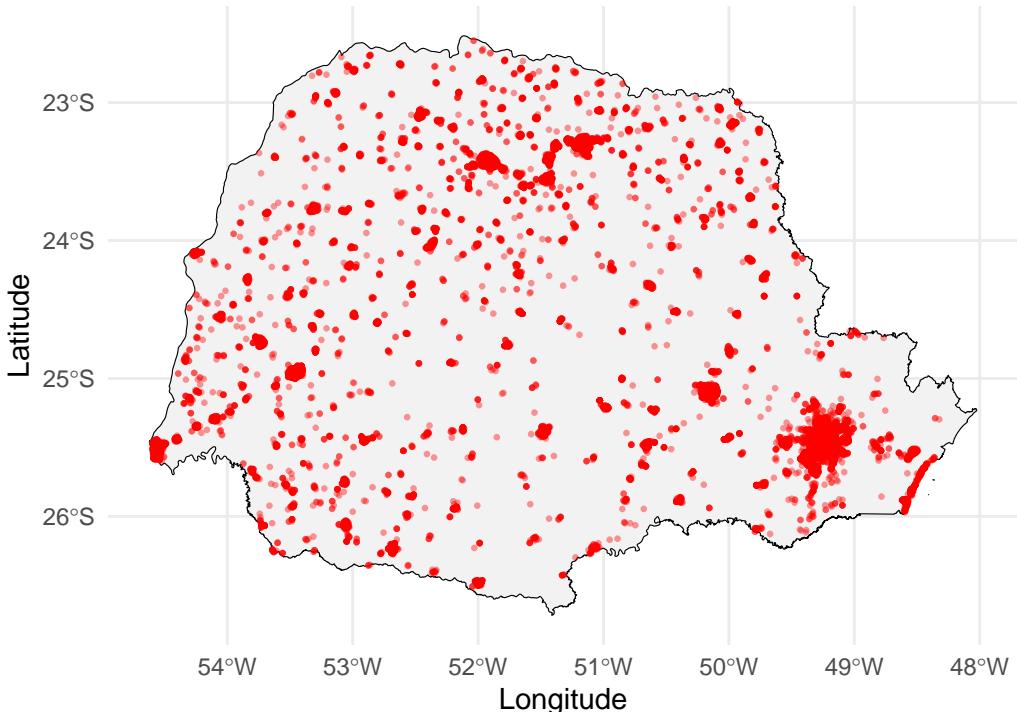
2.2. Mapa de Distribuição de Pontos

Com os dados georreferenciados, o próximo passo é a análise exploratória visual, plotando os pontos sobre o mapa do Paraná para obter uma primeira impressão da sua distribuição.

```
# Distribuição espacial
df_geo_validos <- df_geo %>% filter(!is.na(lat) & !is.na(lon))
df_geo_sf <- st_as_sf(df_geo_validos, coords = c("lon", "lat"), crs = 4674)
mapa_pr <- read_state(code_state = "PR", year = 2020)

ggplot() +
  geom_sf(data = mapa_pr, fill = "gray95", color = "black") +
  geom_sf(data = df_geo_sf, color = "red", size = 0.5, alpha = 0.4) +
  labs(title = "Distribuição Espacial das Ocorrências no Paraná",
       subtitle = "Medidas Protetivas de Urgência (Lei Maria da Penha) - 2024",
       x = "Longitude", y = "Latitude", caption = "Fonte: Dados anonimizados do TJPR") +
  theme_minimal()
```

Distribuição Espacial das Ocorrências no Paraná Medidas Protetivas de Urgência (Lei Maria da Penha) – 2024



Fonte: Dados anonimizados do TJPR

3. Análise de Padrão de Pontos

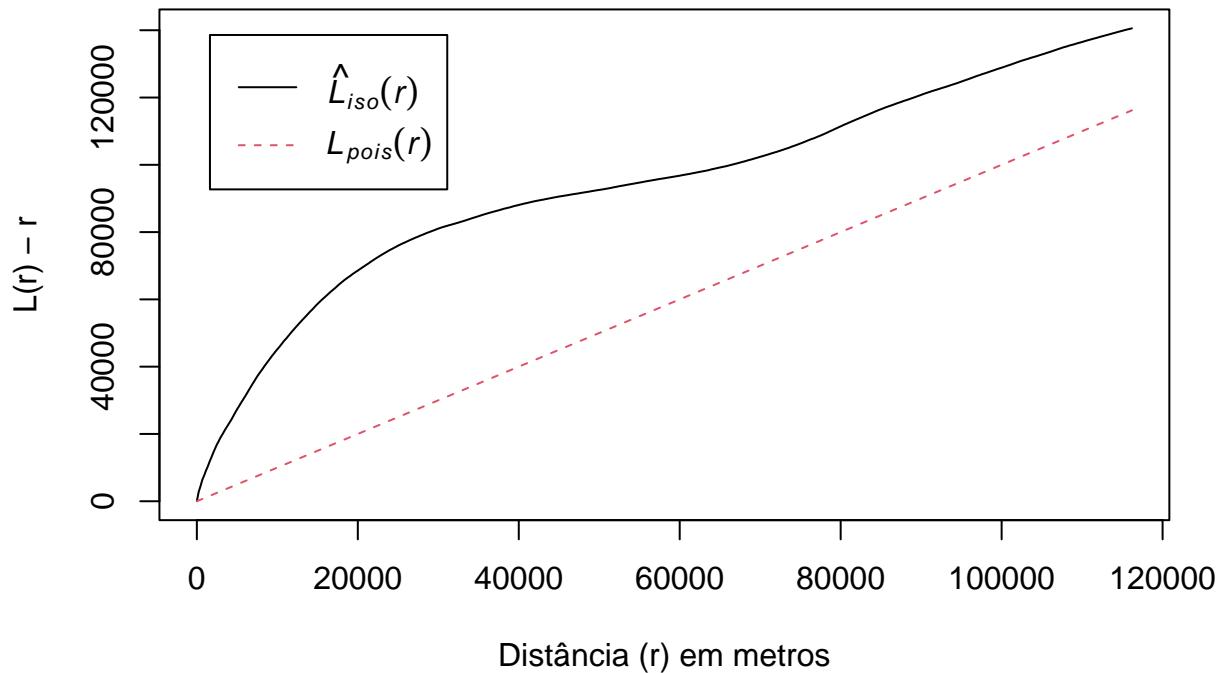
A visualização inicial revela um padrão com nítidos aglomerados. Para verificar formalmente se este padrão de agrupamento é estatisticamente significativo, utiliza-se a Função K de Ripley. Dado o alto custo computacional, a análise é realizada sobre uma amostra.

```
# Padrão de pontos
mapa_pr_proj <- st_transform(mapa_pr, crs = 31982)
df_geo_sf_proj <- st_transform(df_geo_sf, crs = 31982)
df_amostra <- df_geo_sf_proj %>% sample_n(1000)

coords_amostra <- st_coordinates(df_amostra)
janela_pr_proj <- as.owin(mapa_pr_proj)
padraoPontosAmostra <- ppp(x = coords_amostra[,1], y = coords_amostra[,2], window = janela_pr_proj)

L <- Lest(padraoPontosAmostra, correction = "isotropic")
plot(L, main = "Análise de Padrão de Pontos (Função L de Ripley)",
     xlab = "Distância (r) em metros", ylab = "L(r) - r")
```

Análise de Padrão de Pontos (Função L de Ripley)



Interpretação do Gráfico da Função L O gráfico da Função L de Ripley confirma a observação visual do mapa. Ele compara a função observada nos dados com o valor teórico esperado para um padrão completamente aleatório. Como a linha observada está consistentemente acima da área de simulação (não mostrada no plot simples, mas implícita na teoria), há uma forte evidência estatística de **agrupamento (clustering)**. Isso significa que as ocorrências tendem a estar mais próximas umas das outras do que o esperado pelo acaso.

4. Análise Geoestatística

Para aprofundar a análise, investiga-se a estrutura da dependência espacial através do semivariograma.

4.1. Incorporando a População como Covariável

Para remover a tendência causada pela distribuição demográfica, a população é incorporada como uma covariável. Os dados são obtidos do IBGE via pacote `sidrar` e agregados em uma grade de análise.

```
# Obter população
populacao_municipios_sidra <- get_sidra(x = 6579, period = "last", geo = "City",
                                             variable = 9324, header = FALSE)
populacao_recente <- populacao_municipios_sidra %>%
```

```

select(code_muni = D1C, pop_total = V) %>%
  mutate(code_muni = as.character(code_muni))

municipios_com_pop <- read_municipality(code_muni = "PR", year = 2020) %>%
  mutate(code_muni = as.character(code_muni)) %>%
  left_join(populacao_recente, by = "code_muni") %>%
  filter(!is.na(pop_total))
municipios_proj <- st_transform(municipios_com_pop, crs = 31982)

# Grade no mapa
tamanho_grade <- 12000
grade_proj <- st_make_grid(df_geo_sf_proj, cellsize = tamanho_grade) %>%
  st_as_sf() %>% mutate(cell_id = 1:n())

contagens_por_celula <- st_join(df_geo_sf_proj, grade_proj) %>%
  st_drop_geometry() %>% count(cell_id, name = "contagem")

# Calculacao população em cada célula da grade
municipios_densidade <- municipios_proj %>%
  mutate(area_m2 = as.numeric(st_area(geom)),
         densidade_pop = pop_total / area_m2)
pop_por_celula <- st_join(st_centroid(grade_proj), select(municipios_densidade, densidade_pop)) %>%
  st_drop_geometry() %>% mutate(pop_estimada = densidade_pop * (tamanho_grade * tamanho_grade))

dados_modelo <- grade_proj %>%
  left_join(contagens_por_celula, by = "cell_id") %>%
  left_join(select(pop_por_celula, cell_id, pop_estimada), by = "cell_id") %>%
  mutate(contagem = ifelse(is.na(contagem), 0, contagem)) %>%
  filter(!is.na(pop_estimada) & pop_estimada > 0)

```

4.2. Semivariograma dos Resíduos e Mapas Comparativos

Ajustamos um modelo de regressão Poisson para remover o efeito da população e da tendência espacial de larga escala. O semivariograma é então calculado sobre os resíduos deste modelo.

```

# Transformação para modelar
coords_modelo_df <- as.data.frame(st_coordinates(st_centroid(dados_modelo)))
dados_modelo$X <- coords_modelo_df$X
dados_modelo$Y <- coords_modelo_df$Y

# Ajuste de modeo
modelo_final <- glm(contagem ~ poly(X, Y, degree = 2), offset = log(pop_estimada),
                      family = poisson, data = dados_modelo)

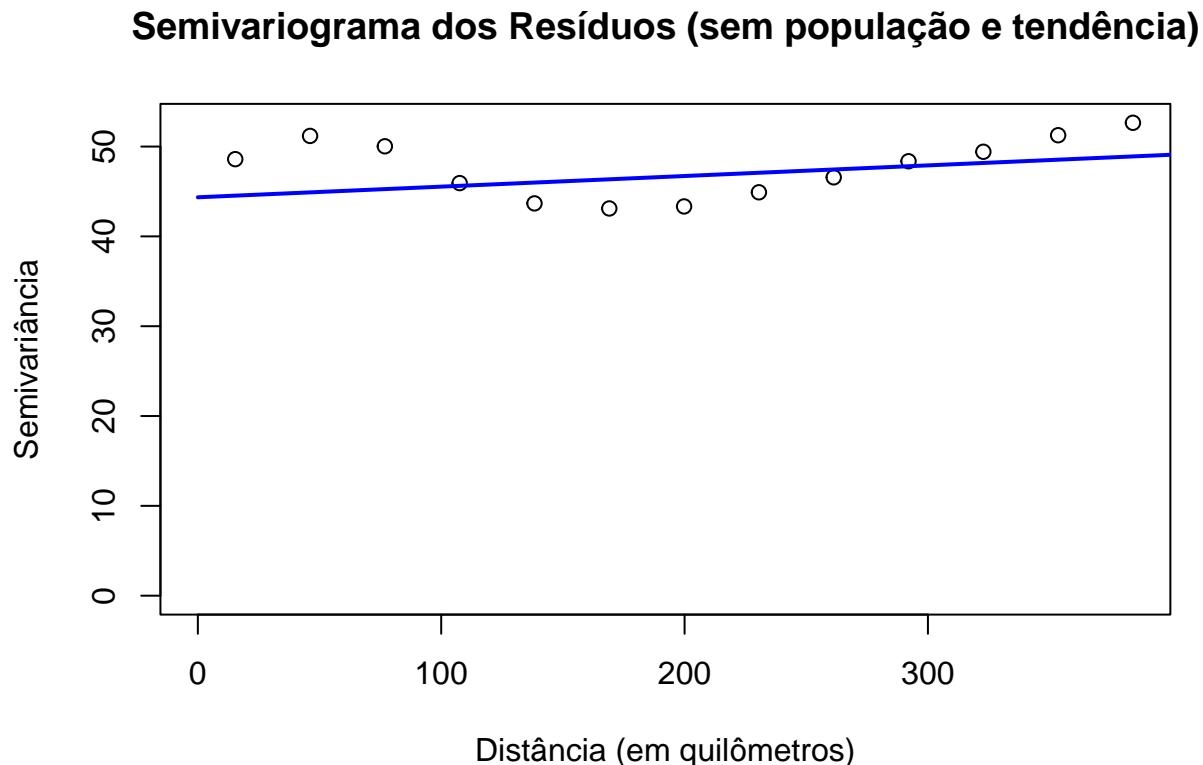
# Variograma dos resíduos
dados_modelo$residuos_finais <- residuals(modelo_final, type = "pearson")
dados_para_geor_df <- st_drop_geometry(dados_modelo)
dados_residuos_finais_geor <- as.geodata(obj = dados_para_geor_df, coords.col = c("X", "Y"),
                                             data.col = which(names(dados_para_geor_df) == "residuos_finais"))
dados_residuos_geor_sem_duplicatas <- jitterDupCoords(dados_residuos_finais_geor, max = 1)
variograma_final <- variog(dados_residuos_geor_sem_duplicatas, max.dist = 400000, messages = FALSE)
variograma_final$u <- variograma_final$u / 1000

```

```

modelo_ajustado <- variofit(variograma_final, ini.cov.pars = c(5, 350), nugget = 48,
                                cov.model = "spherical", messages = FALSE)
plot(variograma_final, main = "Semivariograma dos Resíduos (sem população e tendência)",
      xlab = "Distância (em quilômetros)", ylab = "Semivariância")
lines(modelo_ajustado, col = "blue", lwd = 2)

```



Interpretação do Semivariograma dos Resíduos O semivariograma final, após a remoção do efeito da população e da tendência espacial de larga escala, apresenta uma estrutura “clássica”. Ele começa em um valor não nulo (efeito pepita), cresce com a distância (indicando que locais próximos possuem um risco residual semelhante) e estabiliza em um patamar. Isso confirma que o modelo de regressão foi bem-sucedido em filtrar os padrões de grande escala, revelando a **autocorrelação espacial local** do risco, que pode agora ser modelada.

4.3. Mapas de Risco e Resíduos

Os mapas a seguir compararam a distribuição bruta, a taxa por habitante e os resíduos do modelo, que indicam o risco acima ou abaixo do esperado.

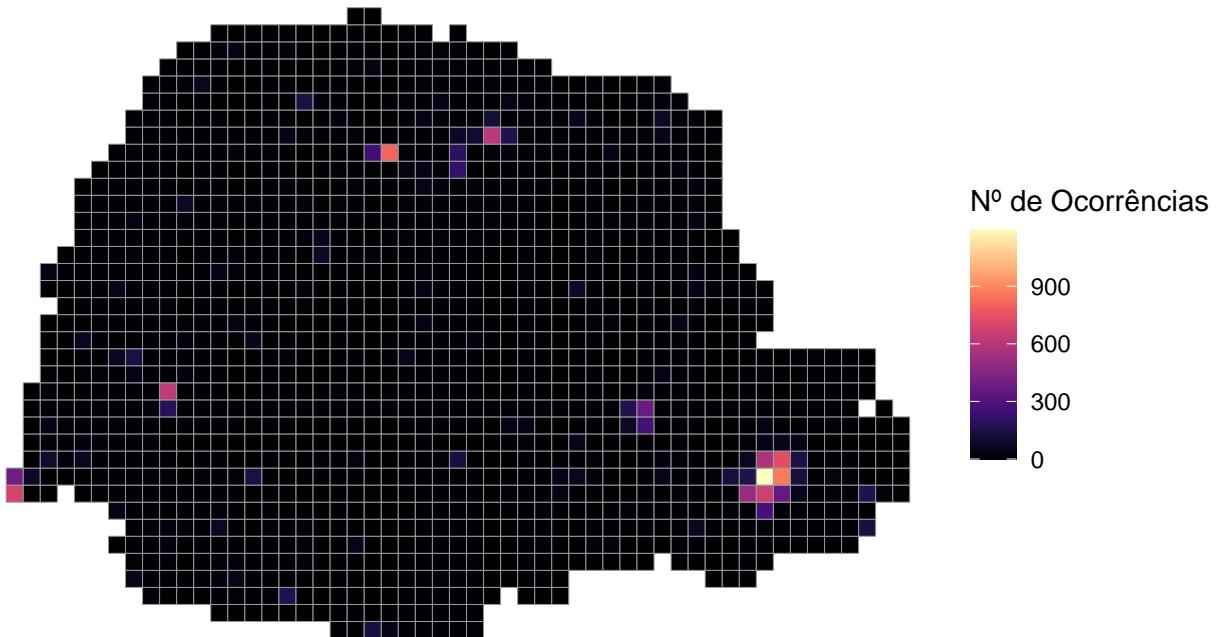
```

# Mapas Paraná
dados_para_mapas <- dados_modelo %>%
  mutate(taxa_por_10k = (contagem / (pop_estimada + 1)) * 10000)

```

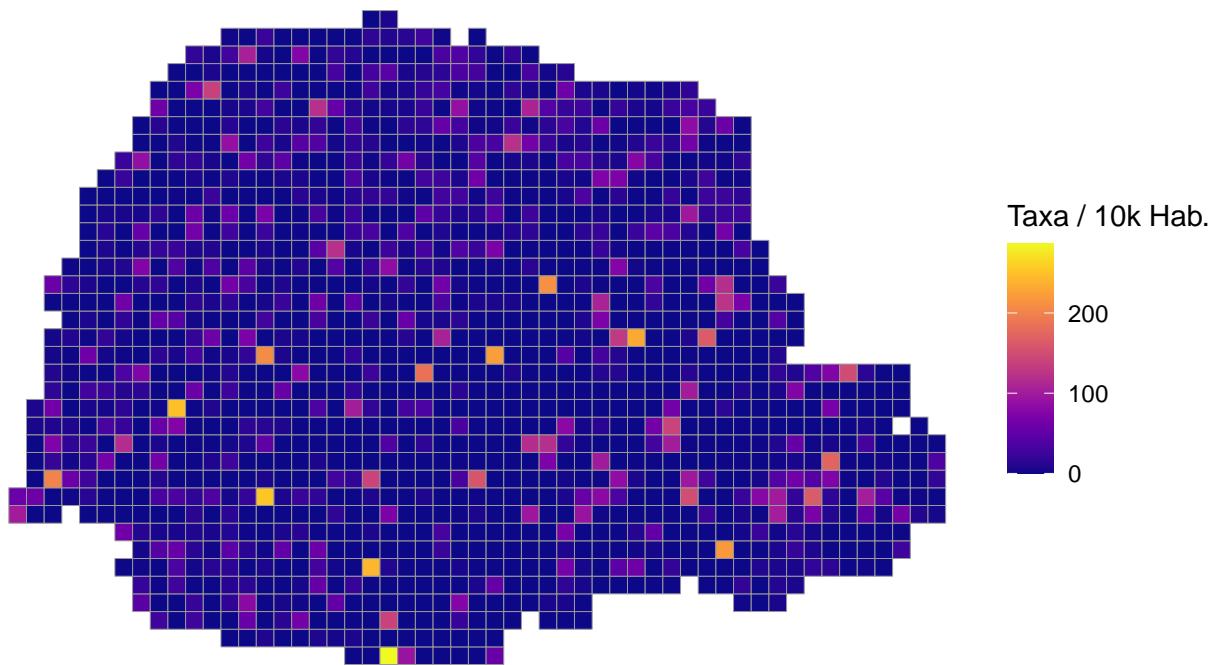
```
ggplot() +
  geom_sf(data = dados_para_mapas, aes(fill = contagem), color = "grey60", linewidth=0.1) +
  scale_fill_viridis(name = "Nº de Ocorrências", option = "A") +
  labs(title = "Mapa 1: Contagem Absoluta de Ocorrências") + theme_void()
```

Mapa 1: Contagem Absoluta de Ocorrências



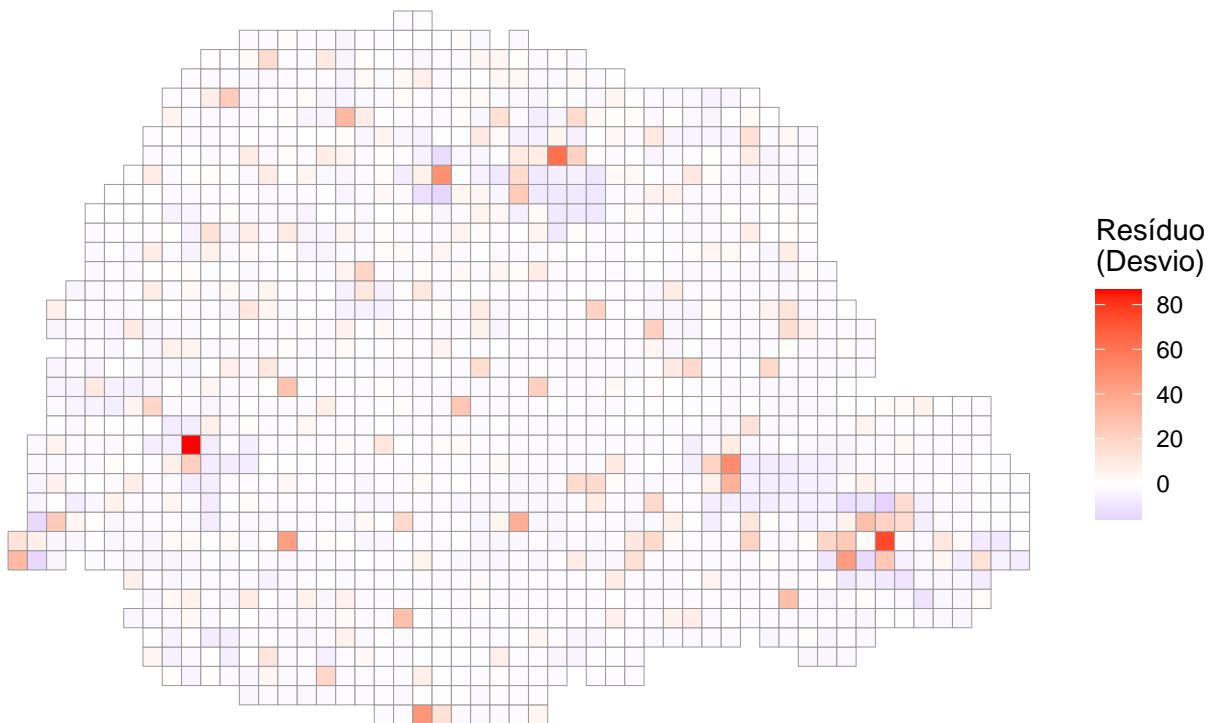
```
ggplot() +
  geom_sf(data = dados_para_mapas, aes(fill = taxa_por_10k), color = "grey60", linewidth=0.1) +
  scale_fill_viridis(name = "Taxa / 10k Hab.", option = "C") +
  labs(title = "Mapa 2: Risco por Habitante") + theme_void()
```

Mapa 2: Risco por Habitante



```
ggplot() +
  geom_sf(data = dados_para_mapas, aes(fill = residuos_finais), color = "grey60",
          linewidth=0.1) +
  scale_fill_gradient2(name = "Resíduo\n(Desvio)", low = "blue", mid = "white",
                       high = "red", midpoint = 0) +
  labs(title = "Mapa 3: Resíduos do Modelo (Risco não explicado)") + theme_void()
```

Mapa 3: Resíduos do Modelo (Risco não explicado)



Interpretação dos Mapas Comparativos A diferença fundamental entre os mapas é a remoção do efeito populacional. O **Mapa 1 (Contagem Absoluta)** mostra onde há o maior número de casos, refletindo a distribuição da população. O **Mapa 2 (Risco por Habitante)**, ao normalizar pela população, revela o risco relativo, destacando áreas que podem ter um risco por habitante elevado mesmo sem uma grande população. Finalmente, o **Mapa 3 (Resíduos do Modelo)** representa a análise mais aprofundada. As células em **vermelho** são os “hotspots” estatísticos, áreas onde a incidência de casos é significativamente **maior** do que o previsto pelo modelo (que já considera população e tendência espacial), indicando a presença de outros fatores de risco locais.

5. Modelagem Preditiva (Krigagem)

Com um modelo de variograma bem definido, o passo final é a krigagem, que permite criar uma superfície de risco contínua para todo o estado.

```
# Krigagem
grade_predicao <- st_make_grid(mapa_pr_proj, cellsize = 10000) %>% st_as_sf()
coords_predicao <- st_coordinates(st_centroid(grade_predicao))
coords_predicao_df <- as.data.frame(coords_predicao)

kc_universal <- krige.conv(geodata = dados_residuos_finais_geor,
```

```

    locations = coords_predicao, krigue = krigue.control(
trend.d = ~poly(X, Y, degree = 2),
trend.l = ~poly(coords_predicao_df$X, coords_predicao_df$Y, degree = 2),
obj.model = modelo_ajustado)

## krigue.conv: model with constant mean
## krigue.conv: Kriging performed using global neighbourhood

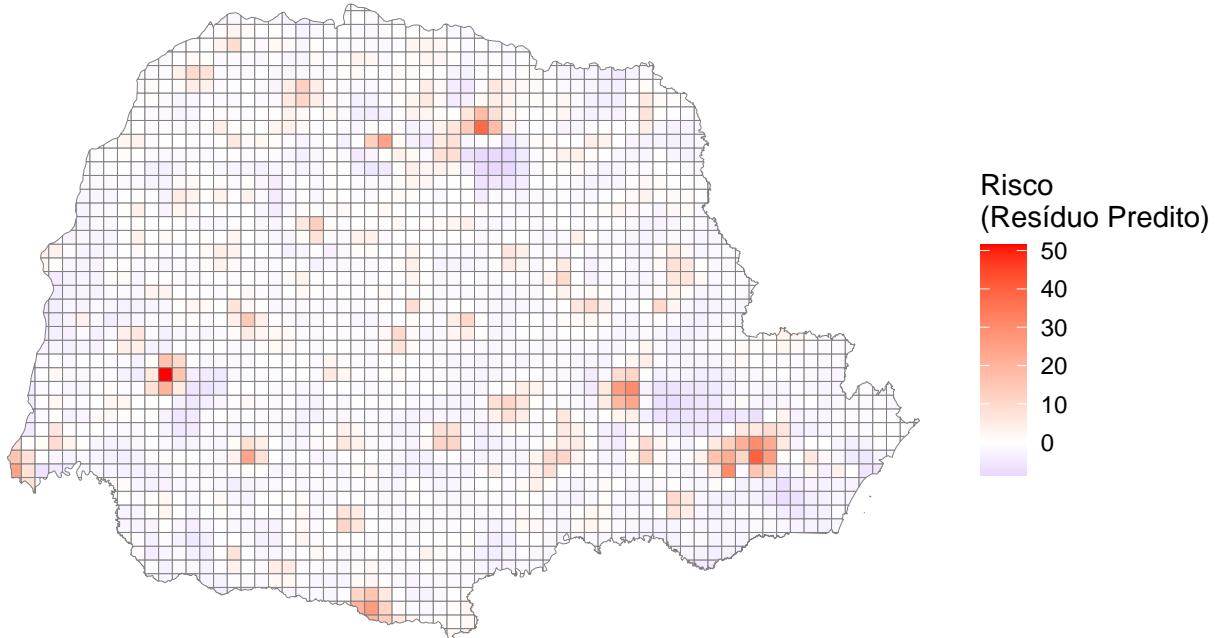
grade_predicao$predicao <- kc_universal$predict
grade_predicao$variancia <- kc_universal$krigue.var
grade_recortada <- st_intersection(grade_predicao, mapa_pr_proj)

# Mapa de Risco Final
ggplot() +
  geom_sf(data = grade_recortada, aes(fill = predicao), color = "gray50", linewidth = 0.1) +
  scale_fill_gradient2(name = "Risco\n(Resíduo Predito)", low = "blue",
                        mid = "white", high = "red", midpoint = 0) +
  labs(title = "Mapa de Risco Final (Krigagem Universal dos Resíduos)",
       subtitle = "Vermelho: Risco maior que o esperado | Azul: Risco menor que o esperado") +
  theme_void()

```

Mapa de Risco Final (Krigagem Universal dos Resíduos)

Vermelho: Risco maior que o esperado | Azul: Risco menor que o esperado



Mapa de Incerteza da Predição

```

ggplot() +
  geom_sf(data = grade_recortada, aes(fill = variancia), color = "gray50", linewidth = 0.1) +

```

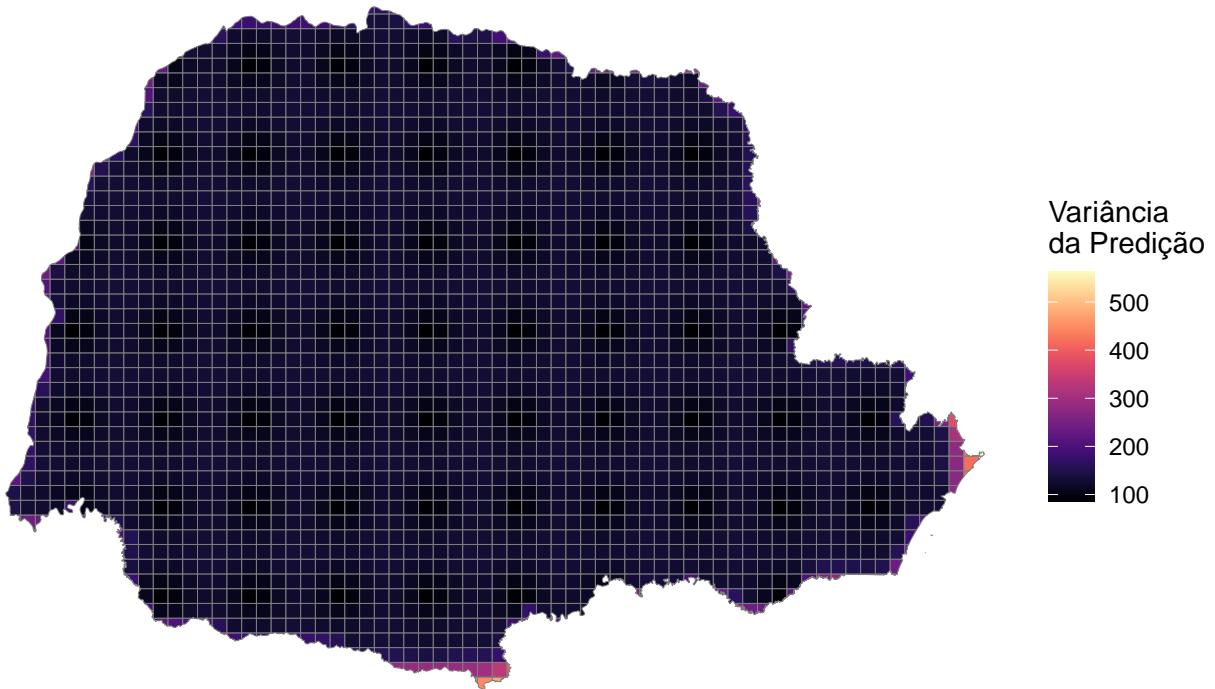
```

scale_fill_viridis(name = "Variância\nda Predição", option = "A") +
  labs(title = "Mapa de Incerteza da Predição (Variância da Krigagem)",
       subtitle = "Áreas mais claras indicam maior incerteza na estimativa") +
  theme_void()

```

Mapa de Incerteza da Predição (Variância da Krigagem)

Áreas mais claras indicam maior incerteza na estimativa



Interpretação dos Mapas de Krigagem Com a krigagem universal, a análise culmina em dois mapas complementares: o de risco predito e o de sua respectiva incerteza. O mapa de risco final destaca em vermelho os “hotspots” — áreas onde a incidência de violência doméstica é significativamente maior do que o esperado, mesmo após o ajuste para os efeitos da população e das tendências espaciais de larga escala, apontando para regiões que podem exigir maior atenção. A interpretação desses hotspots, contudo, é qualificada pelo mapa de incerteza, que indica a confiabilidade das previsões. Neste, as áreas mais escuras representam baixa variância e, portanto, alta certeza nos resultados, enquanto as áreas mais claras nas periferias do estado apontam para uma maior incerteza, advertindo contra interpretações conclusivas sobre o risco nessas localidades onde o modelo realiza maior extrapolação.

Finalmente, para traduzir os resultados do mapa de risco em informações práticas, identificamos as células da grade com o maior risco residual predito (os “hotspots”) e listamos quais municípios estão contidos ou são intersectados por essas áreas de alta criticidade. A tabela a seguir apresenta os 1% de locais com o maior risco, ou seja, onde a incidência de casos é mais alta do que o esperado, mesmo após considerar a população e as tendências espaciais.

```

# Hotspots
limite_risco <- quantile(grade_recortada$predicao, 0.99, na.rm = TRUE)
hotspots_sf <- grade_recortada %>%
  filter(predicao >= limite_risco) %>%
  mutate(hotspot_id = 1:n())
hotspots_com_cidades <- st_join(hotspots_sf, select(municipios_proj, name_muni))

tbla_hotspots <- hotspots_com_cidades %>%
  mutate(Lon_UTM = st_coordinates(st_centroid(.))[,1],
         Lat_UTM = st_coordinates(st_centroid(.))[,2]) %>%
  st_drop_geometry() %>%
  group_by(hotspot_id, Risco = predicao, Lat_UTM, Lon_UTM) %>%
  summarise(`Município(s)` = paste(unique(name_muni), collapse = ", "),
            .groups = 'drop') %>%
  select(`Risco Pred` = Risco, `Lat` = Lat_UTM,
         `Lon` = Lon_UTM, `Município(s)` ) %>%
  arrange(desc(`Risco Pred`))
print(tbla_hotspots)

```

```

## # A tibble: 22 x 4
##   `Risco Pred`     Lat     Lon `Município(s)`
##       <dbl>    <dbl>   <dbl> <chr>
## 1      51.4 7239829. 250891. Cascavel
## 2      40.1 7179829. 680891. Curitiba, Pinhais, Piraquara, São José Dos Pin-
## 3      38.3 7419829. 480891. Cambé, Londrina
## 4      30.4 7169829. 660891. Araucária, Curitiba
## 5      30.3 7229829. 590891. Carambeí, Ponta Grossa
## 6      30.1 7189829. 680891. Colombo, Curitiba, Pinhais
## 7      26.3 7069829. 400891. Palmas
## 8      26.3 7179829. 690891. Pinhais, Piraquara, São José Dos Pinhais
## 9      26.0 7229829. 580891. Carambeí, Ponta Grossa
## 10     24.6 7409829. 410891. Marialva, Maringá, Sarandi
## # i 12 more rows

```

6. Região Metropolitana de Curitiba

Após a análise em escala estadual ter revelado a presença de “hotspots” de risco, especialmente na porção leste do estado, prosseguimos com uma análise de foco para investigar com maior detalhe a estrutura espacial do fenômeno na Região Metropolitana de Curitiba (RMC). Para isso, a mesma metodologia geoestatística será reaplicada: os dados de ocorrências e população serão agregados em uma grade de análise mais fina, um novo modelo de regressão será ajustado para a região, e o semivariograma dos resíduos será calculado para, finalmente, gerar um mapa de risco de alta resolução específico para a RMC.

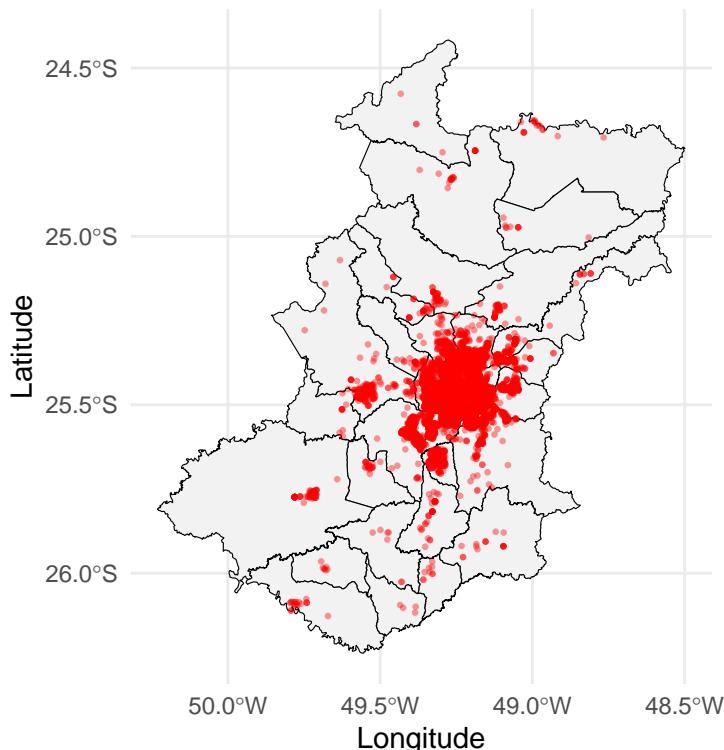
```

# Importação mapa
mapa_rmc <- read_metro_area(year = 2018) %>% filter(name.metro == "RM Curitiba")
municipios_rmc <- mapa_rmc$name_muni
df_geo_rmc <- df_geo %>% filter(Cidade %in% toupper(municipios_rmc))
df_geo_sf_rmc <- df_geo_rmc %>%
  filter(!is.na(lat) & !is.na(lon)) %>%
  st_as_sf(coords = c("lon", "lat"), crs = 4674)

```

```
# Mapa de Distribuição de Pontos na RMC
ggplot() +
  geom_sf(data = mapa_rmc, fill = "gray95", color = "black") +
  geom_sf(data = df_geo_sf_rmc, color = "red", size = 0.5, alpha = 0.4) +
  labs(title = "Distribuição Espacial das Ocorrências na RMC",
       subtitle = "Dados de 2024", x = "Longitude", y = "Latitude") +
  theme_minimal()
```

Distribuição Espacial das Ocorrências na RMC Dados de 2024



O mapa da distribuição espacial das ocorrências na Região Metropolitana de Curitiba (RMC) revela um padrão de concentração ainda mais nítido do que a análise em escala estadual. A grande maioria dos casos, representada pela mancha vermelha densa no centro da imagem, está localizada dentro dos limites do município de Curitiba. Os demais municípios que compõem a RMC apresentam uma incidência consideravelmente menor e mais dispersa, com pequenos agrupamentos que tendem a se formar em suas respectivas sedes urbanas. Este padrão visual reforça a forte correlação entre o número de ocorrências e a densidade populacional, servindo como ponto de partida para a modelagem geoestatística mais detalhada, que buscará identificar o risco real por habitante dentro desta região.

```
# Mapa e População
crs_proj <- 31982
mapa_rmc_proj <- st_transform(mapa_rmc, crs = crs_proj)
pontos_rmc_proj <- st_transform(df_geo_sf_rmc, crs = crs_proj)

populacao_rmc <- municipios_com_pop %>% filter(name_muni %in% municipios_rmc)

municipios_rmc_densidade <- st_transform(populacao_rmc, crs=crs_proj) %>%
  mutate(area_m2 = as.numeric(st_area(geom)),
```

```

densidade_pop = pop_total / area_m2)

tamanho_grade_rmc <- 5000
grade_rmc_proj <- st_make_grid(pontos_rmc_proj, cellsize = tamanho_grade_rmc) %>%
  st_as_sf() %>% mutate(cell_id = 1:n())

contagens_por_celula_rmc <- st_join(pontos_rmc_proj, grade_rmc_proj) %>%
  st_drop_geometry() %>% count(cell_id, name = "contagem")
pop_por_celula_rmc <- st_join(st_centroid(grade_rmc_proj),
                                select(municipios_rmc_densidade, densidade_pop)) %>%
  st_drop_geometry() %>% mutate(pop_estimada = densidade_pop * (tamanho_grade_rmc^2))

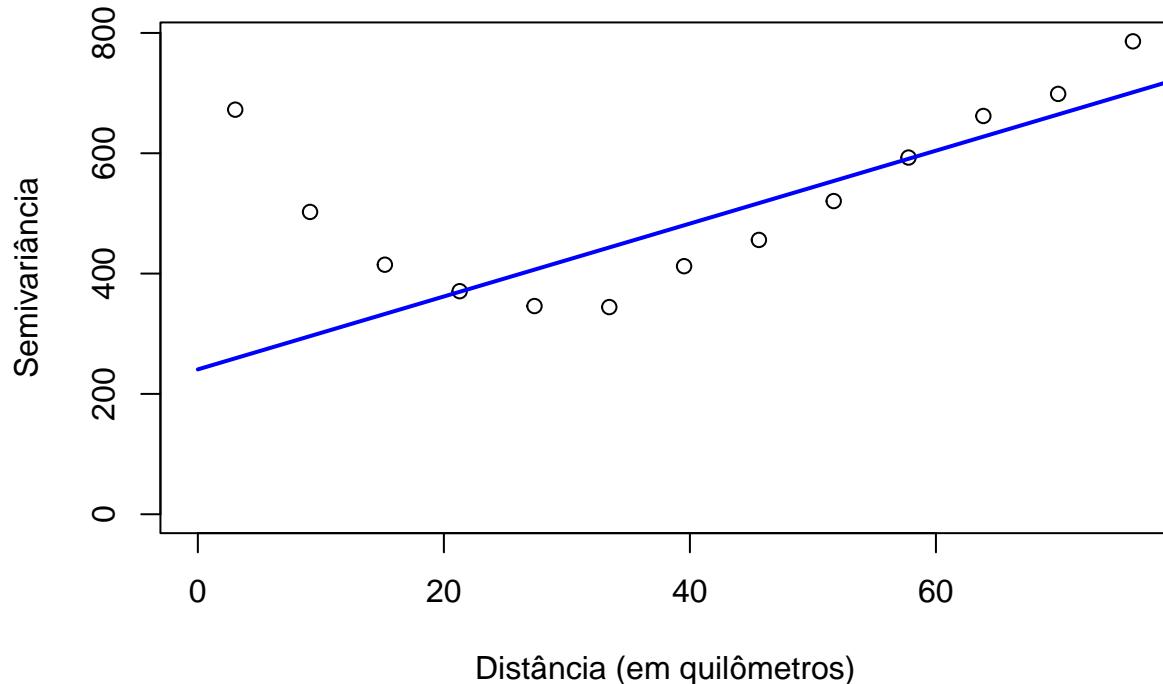
# Preparação dados
dados_modelo_rmc <- grade_rmc_proj %>%
  left_join(contagens_por_celula_rmc, by = "cell_id") %>%
  left_join(select(pop_por_celula_rmc, cell_id, pop_estimada), by = "cell_id") %>%
  mutate(contagem = ifelse(is.na(contagem), 0, contagem)) %>%
  filter(!is.na(pop_estimada) & pop_estimada > 0)
dados_modelo_final_rmc <- dados_modelo_rmc %>%
  mutate(X = st_coordinates(st_centroid(.))[,1],
         Y = st_coordinates(st_centroid(.))[,2])

#Ajuste modelo
modelo_final_rmc <- glm(contagem ~ poly(X, Y, degree = 2),
                           offset = log(pop_estimada),
                           family = poisson,
                           data = dados_modelo_final_rmc)
dados_modelo_final_rmc$residuos_finais <- residuals(modelo_final_rmc, type = "pearson")
dados_para_geor_df <- st_drop_geometry(dados_modelo_final_rmc)
dados_residuos_geor_rmc <- as.geodata(obj = dados_para_geor_df,
                                         coords.col = c("X", "Y"),
                                         data.col = which(names(dados_para_geor_df) == "residuos_finais"))

# Variograma
variograma_rmc <- variog(dados_residuos_geor_rmc, max.dist = 80000, messages = FALSE)
variograma_rmc$u <- variograma_rmc$u / 1000
modelo_ajustado_rmc <- variofit(variograma_rmc, cov.model = "spherical", messages = FALSE)
plot(variograma_rmc, main = "Semivariograma dos Resíduos - Região Metropolitana",
      xlab = "Distância (em quilômetros)", ylab = "Semivariância")
lines(modelo_ajustado_rmc, col = "blue", lwd = 2)

```

Semivariograma dos Resíduos – Região Metropolitana



A análise de foco na Região Metropolitana de Curitiba (RMC) revelou uma estrutura espacial mais complexa. Após a repetição do processo de modelagem, o semivariograma dos resíduos, ilustrado no gráfico acima, apresentou uma estrutura que o modelo teórico esférico não conseguiu se ajustar adequadamente, como demonstra a linha de ajuste azul que não captura a forma dos pontos empíricos. Essa falta de ajuste persistiu mesmo ao se tentar remover a tendência espacial com polinômios de primeiro e segundo grau no modelo de regressão, indicando que a variabilidade do risco na RMC é influenciada por fatores mais complexos do que a simples localização geográfica. Dado que um modelo de variograma mal ajustado não representa a real estrutura de autocorrelação espacial, a krigagem não foi realizada para esta região, pois produziria resultados de predição não confiáveis. Para aprimorar a modelagem nesta escala mais detalhada, seria necessário incorporar covariáveis socioeconômicas em uma resolução mais fina (como por setor censitário), explorar modelos de regressão espacial mais avançados, como a Regressão Geograficamente Ponderada, que podem capturar variações locais nos fatores de risco, e testar diferentes tamanhos de grade de agregação para otimizar a detecção da estrutura espacial intraurbana.

7. Bairros de Curitiba

Este capítulo aprofunda a investigação em uma escala intraurbana, focalizando a análise exclusivamente nos bairros do município. Foram incorporados os dados populacionais de cada bairro, obtidos na plataforma Infocuritiba, mantida pelo Instituto de Pesquisa e Planejamento Urbano de Curitiba (IPPUC), com referência ao ano de 2022. O objetivo, portanto, é reaplicar a metodologia geoestatística para identificar, com alta granularidade, quais bairros apresentam um risco de violência doméstica que excede o esperado com base em sua população, revelando os padrões de hotspots dentro da cidade.

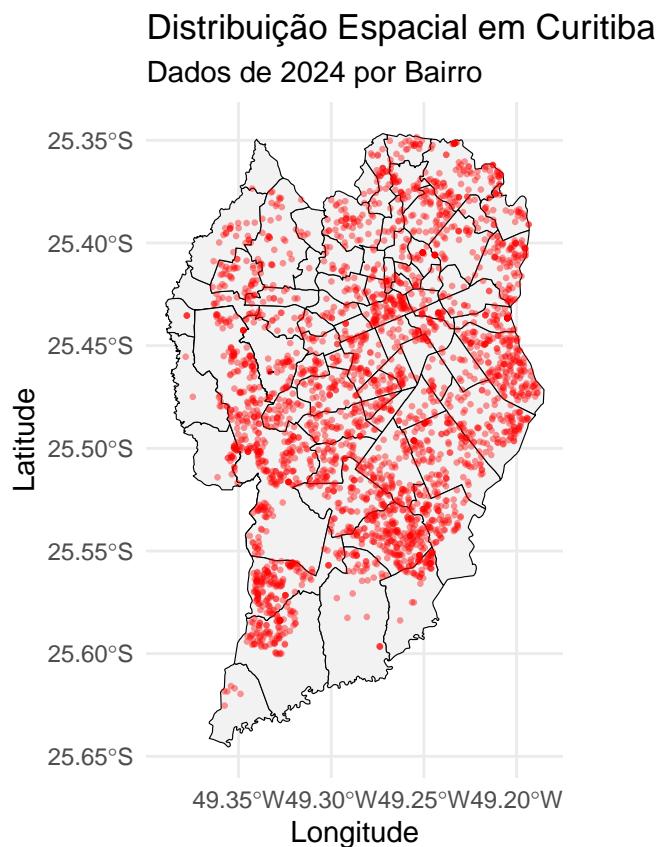
```

# Mapa
mapa_cwb <- geobr::read_neighborhood(year = 2010) %>% filter(name_muni == "Curitiba")

# Ocorrências
df_geo_cwb <- df_geo %>% filter(Cidade == "CURITIBA")
df_geo_sf_cwb <- df_geo_cwb %>%
  filter(!is.na(lat) & !is.na(lon)) %>%
  st_as_sf(coords = c("lon", "lat"), crs = 4674)

# Mapa de Distribuição de Ocorrências em Curitiba
ggplot() +
  geom_sf(data = mapa_cwb, fill = "gray95", color = "black", linewidth = 0.2) +
  geom_sf(data = df_geo_sf_cwb, color = "red", size = 0.5, alpha = 0.4) +
  labs(title = "Distribuição Espacial em Curitiba",
       subtitle = "Dados de 2024 por Bairro", x = "Longitude", y = "Latitude") +
  theme_minimal()

```



```

# Mapa
crs_proj <- 31982
mapa_cwb_proj <- st_transform(mapa_cwb, crs = crs_proj)
pontos_cwb_proj <- st_transform(df_geo_sf_cwb, crs = crs_proj)

# Dados de população
populacao_bairros_preparado <- read.csv("pop_bairros_cwb.csv", sep = ";") %>%
  rename(nome_bairro = Localidade, pop_str = X2022) %>%

```

```

    mutate(pop_total = as.numeric(gsub("\\.", "", pop_str)),
           name_neighborhood_clean = toupper(iconv(nome_bairro, from = "UTF-8",
                                         to = "ASCII//TRANSLIT")))

# Transformações
bairros_com_pop <- mapa_cwb_proj %>%
  mutate(name_neighborhood_clean = toupper(iconv(name_neighborhood, from = "UTF-8",
                                                 to = "ASCII//TRANSLIT"))) %>%
  left_join(populacao_bairros_preparado, by = "name_neighborhood_clean") %>%
  filter(!is.na(pop_total)) %>% mutate(area_m2 = as.numeric(st_area(geom)),
                                             densidade_pop = pop_total / area_m2)

# Grade e contagem por célula
tamanho_grade_cwb <- 1000
grade_cwb_proj <- st_make_grid(pontos_cwb_proj, cellsize = tamanho_grade_cwb) %>%
  st_as_sf() %>% mutate(cell_id = 1:n())

contagens_por_celula_cwb <- st_join(pontos_cwb_proj, grade_cwb_proj) %>%
  st_drop_geometry() %>% count(cell_id, name = "contagem")

pop_por_celula_cwb <- st_join(st_centroid(grade_cwb_proj), bairros_com_pop) %>%
  st_drop_geometry() %>% mutate(pop_estimada = densidade_pop * (tamanho_grade_cwb^2))

dados_modelo_cwb <- grade_cwb_proj %>%
  left_join(contagens_por_celula_cwb, by = "cell_id") %>%
  left_join(select(pop_por_celula_cwb, cell_id, pop_estimada), by = "cell_id") %>%
  mutate(contagem = ifelse(is.na(contagem), 0, contagem)) %>%
  filter(!is.na(pop_estimada) & pop_estimada > 0)

# Modelagem
dados_modelo_cwb <- dados_modelo_cwb %>%
  mutate(X = st_coordinates(st_centroid(.))[, 1],
         Y = st_coordinates(st_centroid(.))[, 2])
modelo_final_cwb <- glm(contagem ~ poly(X, Y, degree = 2),
                           offset = log(pop_estimada), family = poisson,
                           data = dados_modelo_cwb)

# Resíduos
dados_modelo_cwb$residuos_finais <- residuals(modelo_final_cwb, type = "pearson")

# Variograma
dados_para_geor_df_cwb <- st_drop_geometry(dados_modelo_cwb)
dados_residuos_geor_cwb_raw <- as.geodata(obj = dados_para_geor_df_cwb, coords.col = c("X", "Y"),
                                             data.col = which(names(dados_para_geor_df_cwb) == "residuos_finais"))

## as.geodata: 99 replicated data locations found.
## Consider using jitterDupCoords() for jittering replicated locations.
## WARNING: there are data at coincident or very closed locations, some of the geoR's functions may not
## Use function dup.coords() to locate duplicated coordinates.
## Consider using jitterDupCoords() for jittering replicated locations

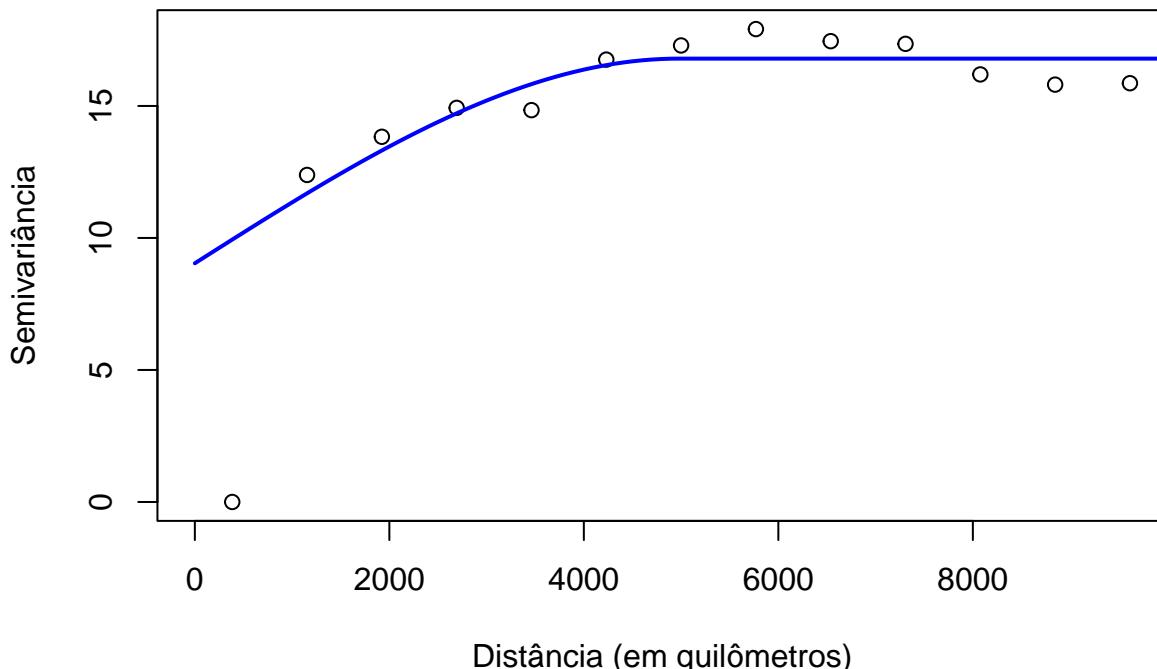
```

```

dados_residuos_geor_cwb <- jitterDupCoords(dados_residuos_geor_cwb_raw, max = 1)
variograma_cwb <- variog(dados_residuos_geor_cwb, max.dist = 10000, messages = FALSE)
modelo_ajustado_cwb <- variofit(variograma_cwb, cov.model = "spherical", messages = FALSE)
#variograma_cwb$u <- variograma_cwb$u / 1000
plot(variograma_cwb, main = "Semivariograma dos Resíduos - Curitiba",
      xlab = "Distância (em quilômetros)", ylab = "Semivariância")
lines(modelo_ajustado_cwb, col = "blue", lwd = 2)

```

Semivariograma dos Resíduos – Curitiba



O gráfico do semivariograma dos resíduos para a análise de Curitiba demonstra o sucesso da modelagem. A linha azul (modelo teórico) se ajusta bem aos pontos empíricos (círculos), que agora apresentam uma estrutura “clássica”: a semivariância cresce com a distância e estabiliza em um patamar a um alcance (range) de aproximadamente 4-5 quilômetros. Isso confirma a existência de uma autocorrelação espacial local no risco residual, validando a aplicação da krigagem para a interpolação e criação do mapa de risco final.

```

# Krigagem
grade_predicao_cwb <- st_make_grid(mapa_cwb_proj, cellsize = 250) %>% st_as_sf()
coords_predicao_cwb <- st_coordinates(st_centroid(grade_predicao_cwb))
coords_predicao_cwb_df <- as.data.frame(coords_predicao_cwb)
kc_universal_cwb <- krige.conv(geodata = dados_residuos_geor_cwb, locations = coords_predicao_cwb,
                                 krige = krige.control(
                                   trend.d = ~ poly(X, Y, degree = 2),
                                   trend.l = ~ poly(coords_predicao_cwb_df$X, coords_predicao_cwb_df$Y, degree = 2),
                                   obj.model = modelo_ajustado_cwb))

## krige.conv: model with constant mean

```

```

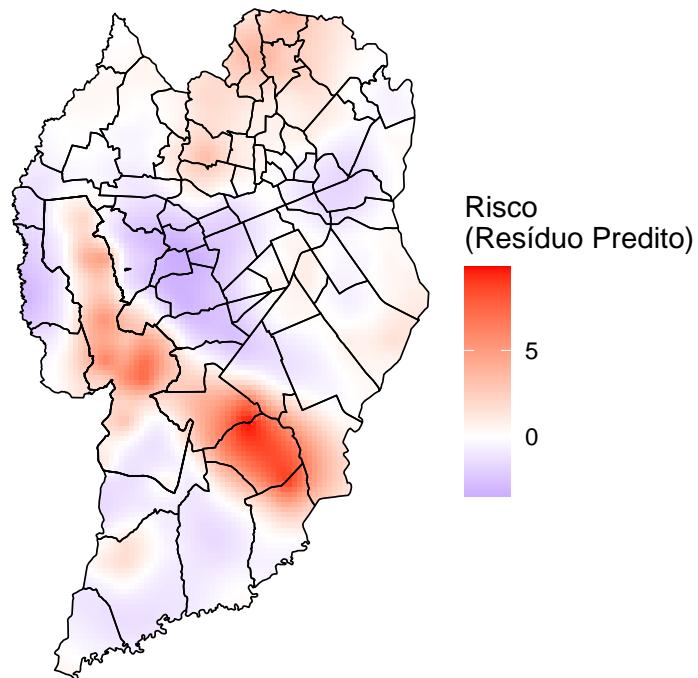
## krig.conv: Kriging performed using global neighbourhood

grade_predicao_cwb$predicao <- kc_universal_cwb$predict
grade_recortada_cwb <- st_intersection(grade_predicao_cwb, mapa_cwb_proj)
ggplot() +
  geom_sf(data = grade_recortada_cwb, aes(fill = predicao), color = NA) +
  geom_sf(data = mapa_cwb_proj, fill = "transparent", color = "black", linewidth = 0.3) +
  scale_fill_gradient2(name = "Risco\n(Resíduo Preditivo)",
                        low = "blue", mid = "white", high = "red", midpoint = 0) +
  labs(title = "Mapa de Risco de Violência Doméstica em Curitiba",
       subtitle = "Vermelho: Risco maior que o esperado | Azul: Risco menor que o esperado",
       caption = "Fonte: Dados anonimizados do TJPR (2024)") +
  theme_void()

```

Mapa de Risco de Violência Doméstica em Curitiba

Vermelho: Risco maior que o esperado | Azul: Risco menor que o esperado



Fonte: Dados anonimizados do TJPR (2024)

O mapa de risco final, produto da krigagem universal, sintetiza os resultados da análise, revelando a superfície contínua do risco de violência doméstica não explicado pela população ou por tendências de larga escala. O mapa destaca “hotspots” (áreas em vermelho) onde o risco é significativamente maior que o esperado, com focos nítidos no bairro Sítio Cercado, na Cidade Industrial de Curitiba (CIC) e em partes da região norte. Em contrapartida, uma grande área de “coldspot” (em azul), representando um risco menor que o previsto, concentra-se na região central da cidade, que pode ter a ver com bairros de maior renda como Batel, Água Verde e Bigorrilho. É importante ressaltar que a modelagem deste trabalho foi simples, utilizando um dos modelos mais fundamentais da estatística espacial e sem a inclusão de covariáveis socioeconômicas além da população, o que abre caminho para trabalhos futuros que busquem explicar as causas por trás dos padrões aqui identificados.