



# Análise de Dependência Espacial em R

**Ângela Terumi Fushita**

**Vitor Vieira Vasconcelos**

Introdução ao uso de dados espaciais para estudos ambientais  
Programa de Pós-Graduação em Ciência e Tecnologia Ambiental

Universidade Federal do ABC

Maio, 2020

Santo André - SP

# **Próximas 2 aulas**

- Análise de Dependência Espacial
- Geoestatística

# Objetivo

Adquirir os conhecimentos e habilidades básicas relacionados a análise de dependência espacial no ambiente R

# Conteúdo

- Análise de vizinhança
- Autocorrelação espacial
- Suavização espacial
- Regressão espacial
- Agrupamento

# **Materiais de aula disponíveis em:**

<https://app.box.com/s/kqwpbxgvagtl9ygsodaat380nqjn1mp2>

Baixar os dados em: D:/R\_CTA/aula8/

# Leitura Prévia



## Capítulos

1 – Análise Espacial e Geoprocessamento

5 – Análise Espacial de Áreas

DRUCK, S.; CARVALHO, M. S.; CÂMARA, G.;  
MONTEIRO, A.V.M (eds). **Análise Espacial de Dados  
Geográficos**. Brasília: EMBRAPA, 2004. Disponível  
em: <http://www.dpi.inpe.br/gilberto/livro/analise/>

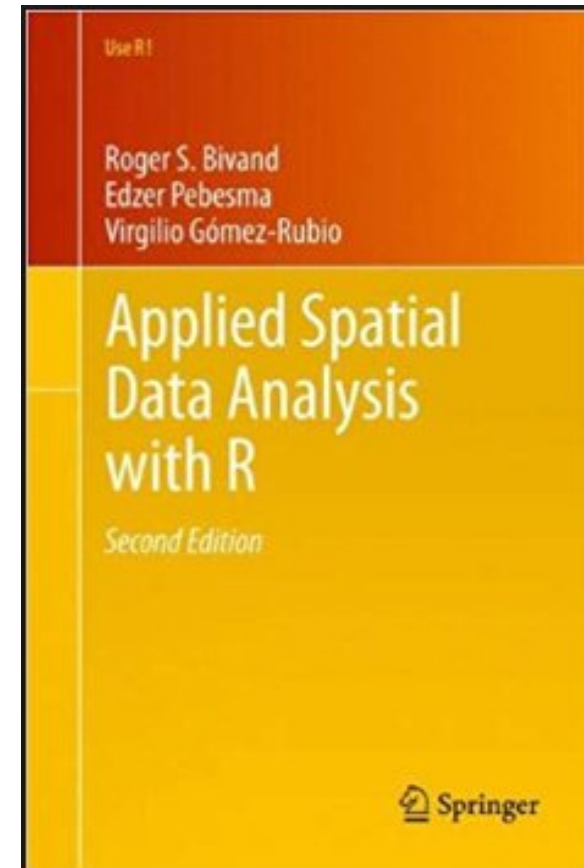
# Livros de Referência

Bivand, Roger. S., Pebesma, E. J., Gomez-Rubio, V., & Pebesma, E. J. (2013). **Applied spatial data analysis with R**. New York: Springer..

<https://app.box.com/s/uti6bqyiscqpoqu2dsmd06yk5xw5m9qw>

Site de apoio: <https://asdar-book.org/>

- Conteúdo de referência
  - Dados vetoriais e raster (formato sp)
  - Interpolação e Geoestatística
  - Autocorrelação espacial

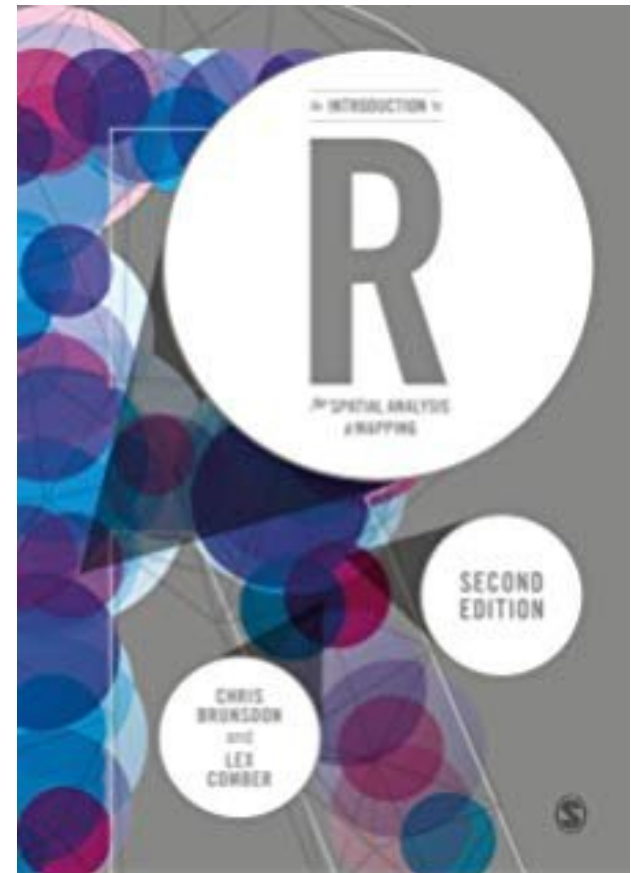


# Livros de Referência

Brunsdon, C. and Comber, L., 2015. **An introduction to R for spatial analysis and mapping.** Sage.

<https://app.box.com/s/wv3j8cuvvznyl47t8t9nkbod87vcvujx>

- Conteúdo de referência
  - Análise de pontos
  - Autocorrelação espacial
  - Regressão ponderada geográfica





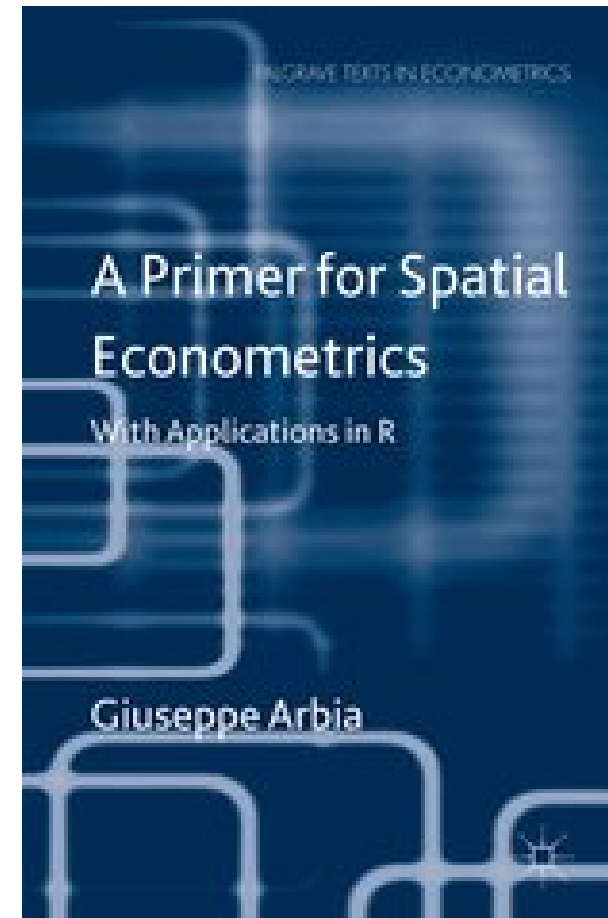
# Livros de Referência

Arbia, Giuseppe. **A primer for spatial econometrics**: with applications in R. Springer, 2014.

PDF: <https://app.box.com/s/xznmpvb21inuf50867c7lxnjn2cbtsct>

Conteúdos:

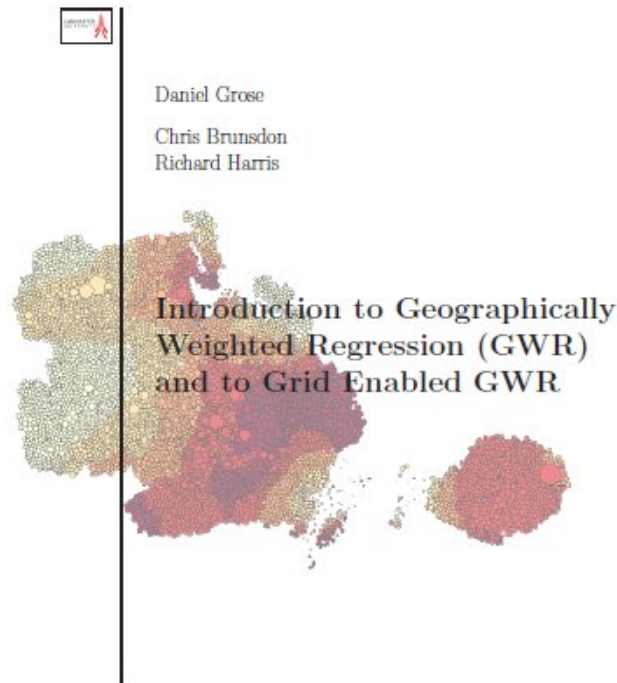
- Matrizes de vizinhança
- Modelos espaciais autorregressivos



# Livros de Referência

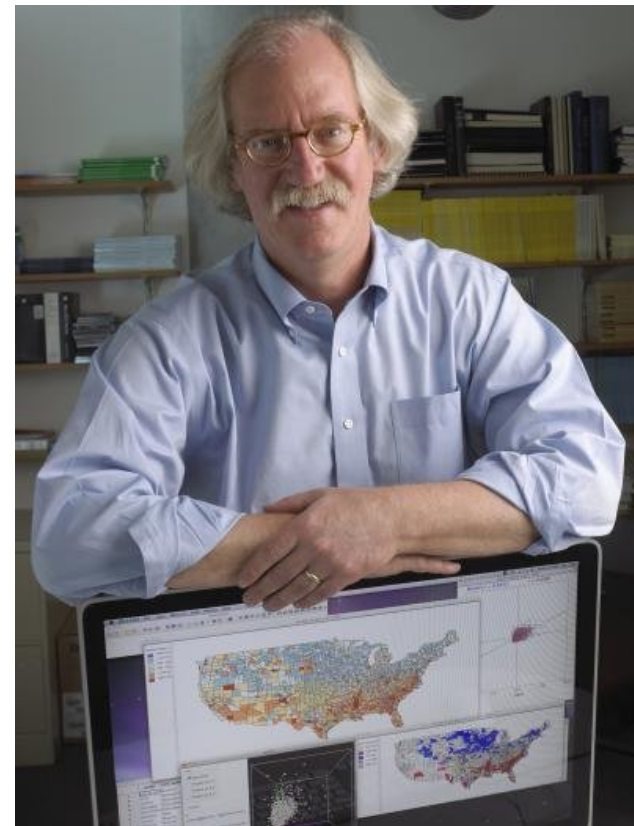
Grose, D., Brunsdon, C., Harris, R. 2011.  
**Introduction to Geographically Weighted  
Regression (GWR) and to Grid Enabled GWR.**  
Lancaster University.

<https://s3-eu-west-1.amazonaws.com/esrc-files/outputs/29-K1tVQBEGILqcChUnuNw/ZIUv3WBmuE-fs1sTU4vzSw.pdf>



# Econometria Espacial

- Dados socioeconômicos agrupados em regiões administrativas
- Incorporação para análise epidemiológica (saúde pública)
- Luc Anselin
  - Geoda
  - Pysal
- Roger Bivand
  - spdep (R)
  - Supervisionado pelo Luc Anselin

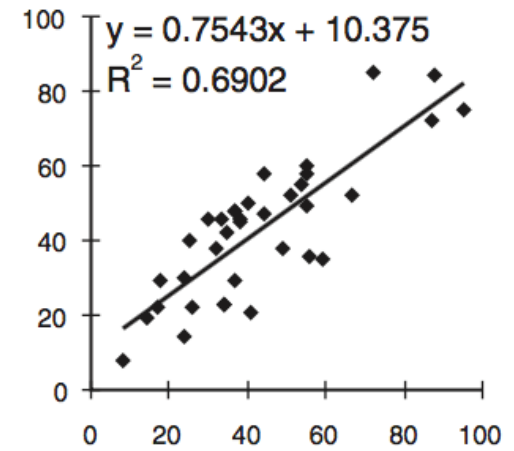


# O Problema da Unidade de Área Modificável

Independent variable    Dependent variable

87	95	72	37	44	24
40	55	55	38	88	34
41	30	26	35	38	24
14	56	37	34	8	18
49	44	51	67	17	37
55	25	33	32	59	54

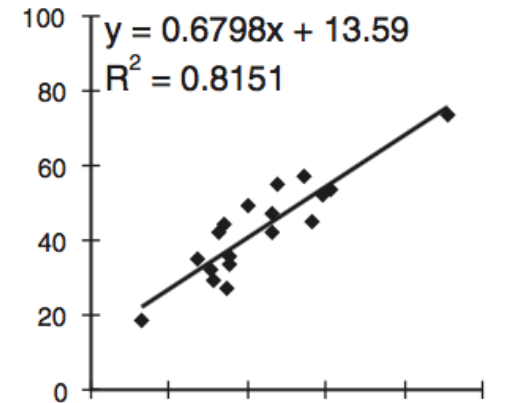
72	75	85	29	58	30
50	60	49	46	84	23
21	46	22	42	45	14
19	36	48	23	8	29
38	47	52	52	22	48
58	40	46	38	35	55



Aggregation scheme 1

91	54.5	34
47.5	46.5	61
35.5	30.5	31
35	35.5	13
46.5	59	27
40	32.5	56.5

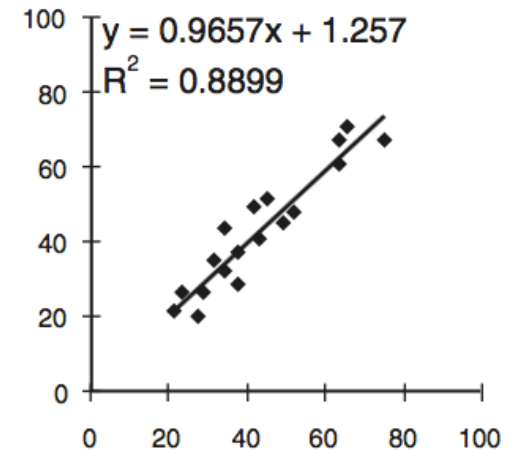
73.5	57	44
55	47.5	53.5
33.5	32	29.5
27.5	35.5	18.5
42.5	52	35
49	42	45



Aggregation scheme 2

63.5	75	63.5	37.5	66	29
27.5	43	31.5	34.5	23	21
52	34.5	42	49.5	38	45.5

61	67.5	67	37.5	71	26.5
20	41	35	32.5	26.5	21.5
48	43.5	49	45	28.5	51.5

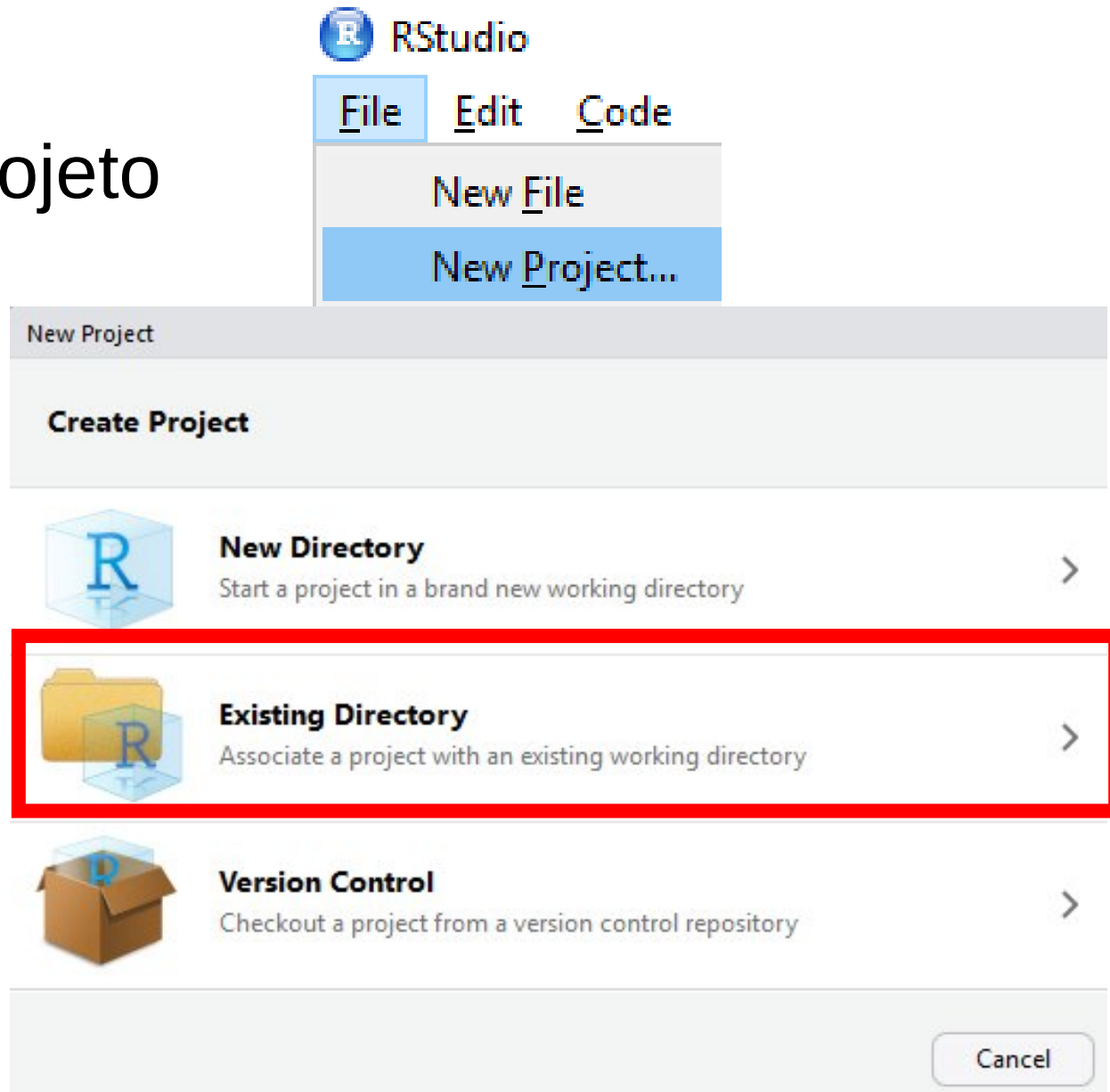


# Conteúdo

- **Análise de vizinhança**
- Autocorrelação espacial
- Suavização espacial
- Regressão espacial
- Agrupamento

# Atividade

- Novo projeto



Back

## Create Project from Existing Directory



Project working directory:

C:/R\_CTA/aula8

Browse...

☐ Open in new session

Create Project

Cancel

R C:/r/cta/aula3 - RStudio

File Edit Code View



R Script Ctrl+Shift+N

- Criar um novo script de programação

R C:/R\_CTA/aula6 - RStudio

File Edit Code View Plots Session

New File

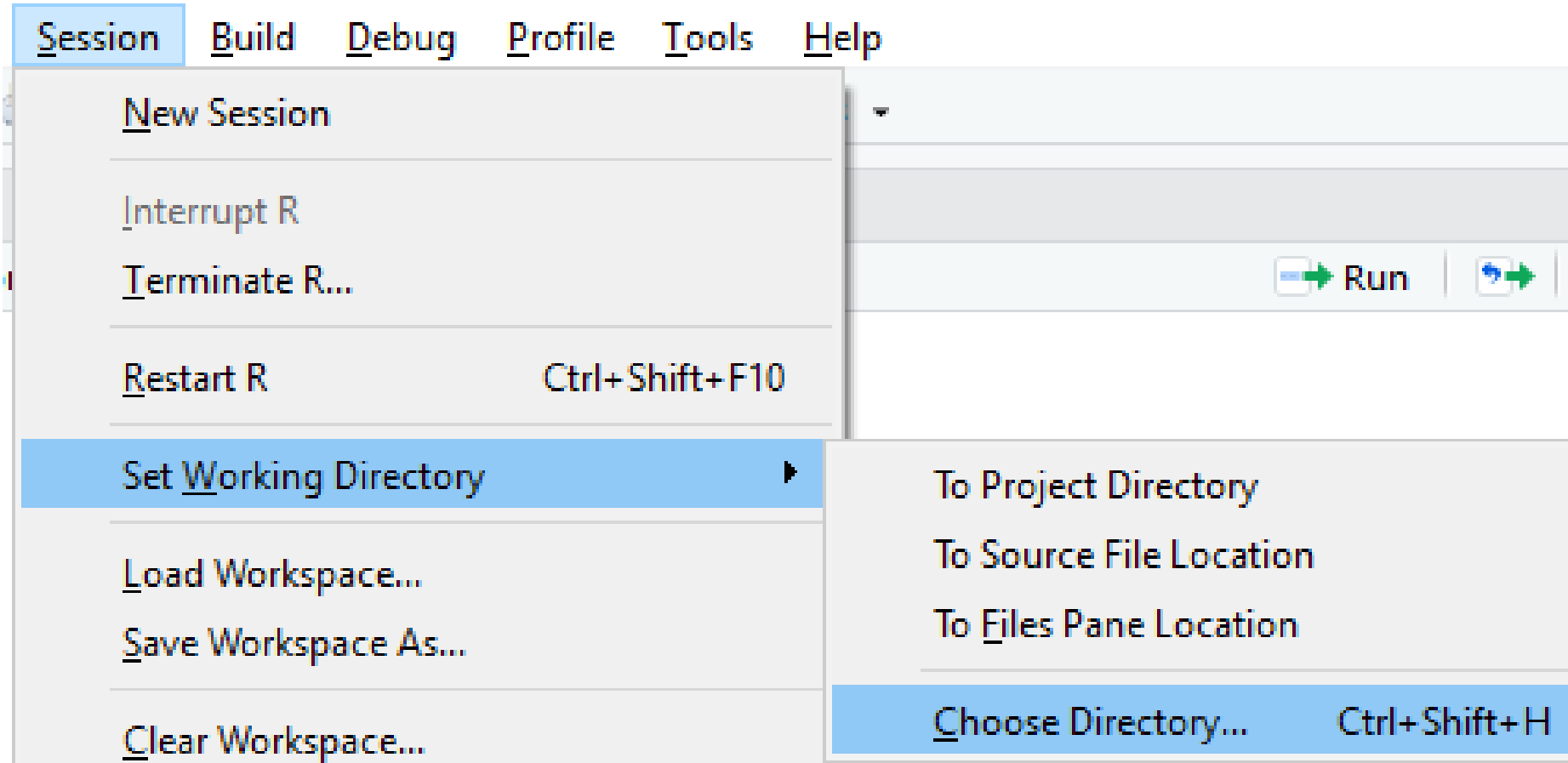
New Project...

Open File... Ctrl+O

- Abrir o script aula8.R



- Configurar o diretório de trabalho



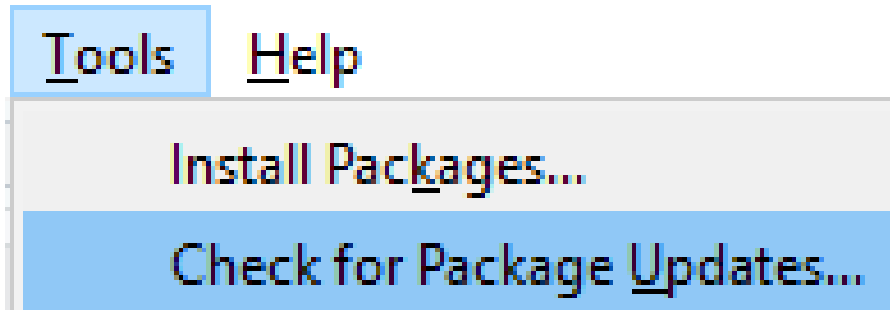
- Exemplo de código:

```
setwd("D:/R_CTA/aula8")
```

- Confirmando o diretório de trabalho

```
getwd()
```

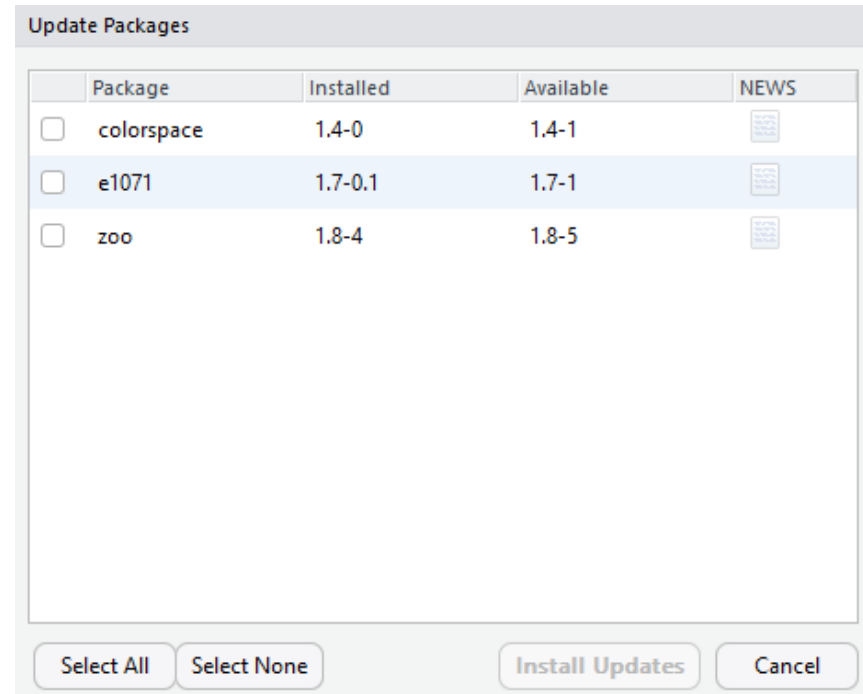
- É sempre recomendável verificar atualizações nos pacotes instalados antes de começar a trabalhar



Comando:  
`update.packages(ask=FALSE)`

Pacotes básicos:

```
install.packages("sf")  
install.packages("sp")  
install.packages("rgdal")  
install.packages("tmap")  
library(sf)  
library(rgdal)  
library(sp)  
library(raster)  
library(tmap)
```



## ##outros pacotes que vamos utilizar na aula

```
install.packages("cleangeo")
```

```
install.packages("spdep")
```

```
install.packages("pgirmess")
```

```
install.packages("spatialreg")
```

```
install.packages("spgwr")
```

```
library(cleangeo)
```

**Correção topológica**

```
library(spdep)
```

**Dependência espacial**

```
library(pgirmess)
```

**Correlograma de distância**

```
library(spatialreg)
```

**Regressão espacial global**

```
library(spgwr)
```

**Regressão ponderada geográfica**

# Importar dados

```
st_layers("aula8.gpkg")
```

**1- Municípios da  
Região  
Metropolitana de  
São Paulo**

**2- Setores  
censitários do  
ABC Paulista**

Driver: GPKG

Available layers:

	layer_name	geometry_type	features	fields
1	mun_rmsp	Polygon	39	9
2	setores_abc	Polygon	4157	20

# Importar datos

```
setores_sf<-st_read("aula8.gpkg", layer="setores_abc")
```

View(setores\_sf) **Ordenar por ordem crescente de renda**

# Censo 2010

# Defesa Civil 2014-2016

Domicilios	Pessoas	Renda	rede_esg	col_lixo	deslizam	inundaco	geometry
91	345	682.13	97.802198	100.00000	0	0	list(c(3345
267	901	578.21	100.000000	100.00000	0	0	list(c(3347
191	649	933.95	99.476440	100.00000	0	0	list(c(3347

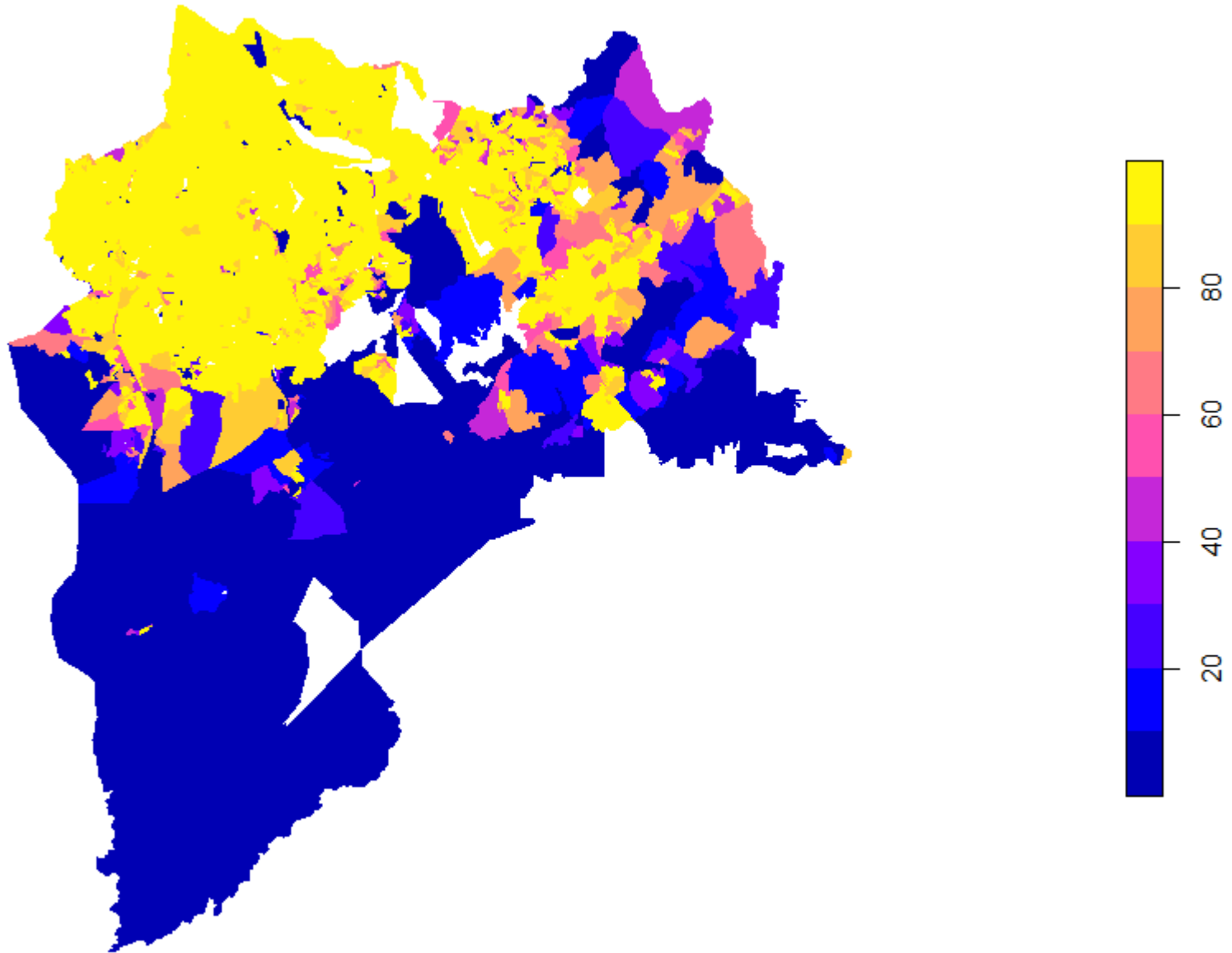
## Setores zerados

Domicilios	Pessoas	Renda	rede_esg	col_lixo
0	0	0.00	0.0000000	0.000000
0	0	0.00	0.0000000	0.000000
0	0	0.00	0.0000000	0.000000
0	0	0.00	0.0000000	0.000000
0	0	0.00	0.0000000	0.000000
0	0	0.00	0.0000000	0.000000
1	4	0.00	0.0000000	0.000000
1	9	0.00	0.0000000	0.000000
0	0	0.00	0.0000000	0.000000

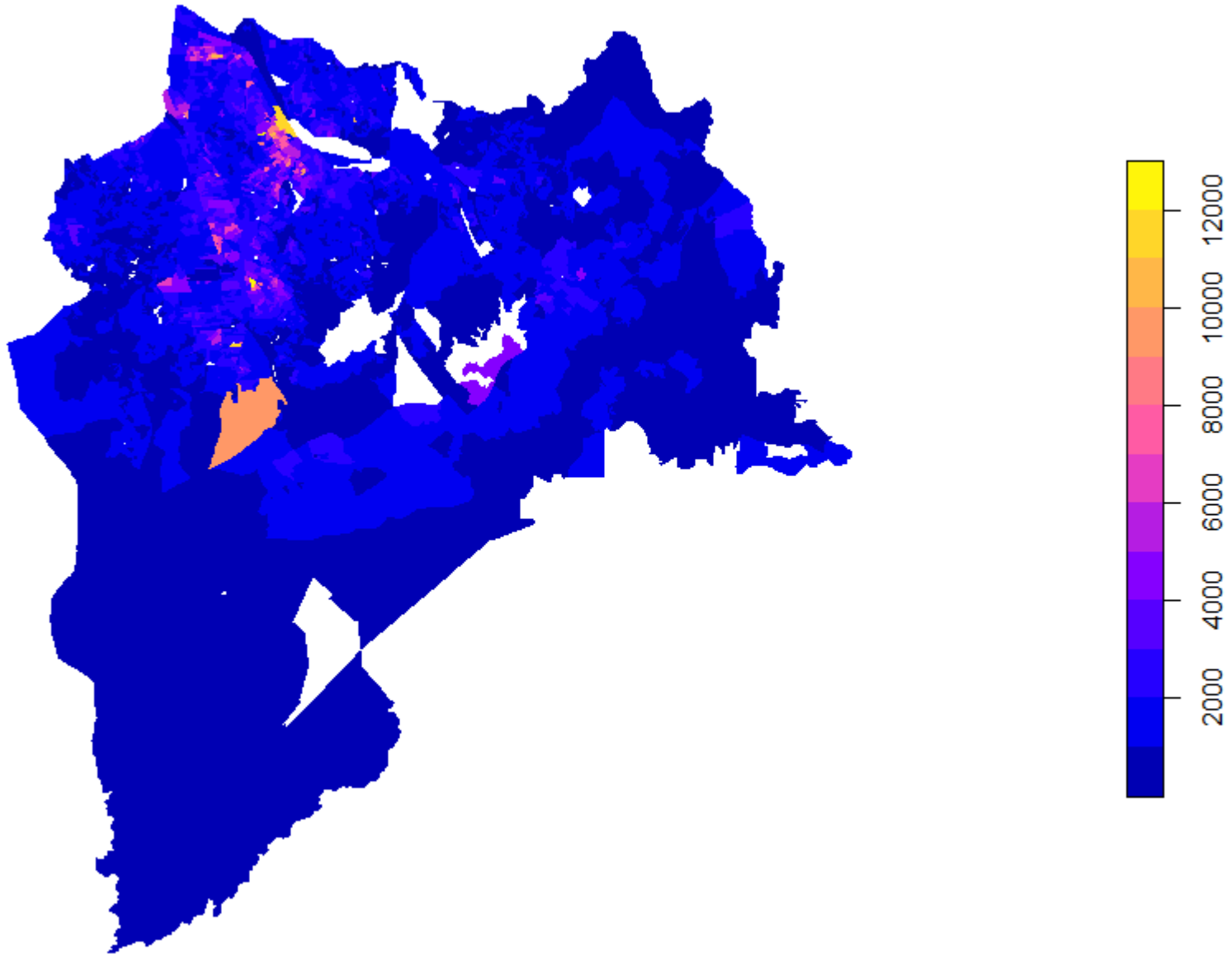
## Setores nulos

[illegible]

```
plot(setores_sf["rede_esg"], border=NA)
```



```
plot(setores_sf["Renda"], border=NA)
```



# O que fazer com os valores nulos e zerados?

- Por que eles estão nulos ou zerados?
  - Áreas desertas
  - Falhas de preenchimento (interpolar?)
- Faz sentido analisar esses elementos?
  - Focar o estudo em áreas não desertas
  - Trabalhar com amostras e não com universo dos dados
- Os modelos aceitam dados faltantes ou zerados?
  - Zerar ao agregar em unidades maiores
  - Descartar para análise de vizinhança
  - Dados zerados podem atrapalhar regressões



```
setores_nao_nulo <- subset(setores_sf, is.na(setores_sf$Renda)==FALSE)
setores_valido <- subset(setores_nao_nulo, setores_nao_nulo$Renda != 0)
View(setores_valido)
```

aula8.R\* x

setores\_valido x

setores\_sf x

Filter

Tipo_setor	Domicilios	Pessoas	Renda	rede_esg
1	112	545	27.06	100.00000000
1	125	456	72.20	100.00000000
1	144	748	91.94	100.00000000
0	170	659	207.86	100.00000000
1	76	317	216.88	96.0526316

# Análise Topológica

```
install.packages("cleangeo")
```

```
library(cleangeo)
```

```
setores_sp <- as(setores_valido,"Spatial")
```

```
clgeo_IsValid(setores_sp)
```

```
[1] FALSE
```

```
analise_topologica <- clgeo_CollectionReport(setores_sp)
```

```
clgeo_SummaryReport(analise_topologica)
```

type	valid	issue_type
rgeos_validity: 40	Mode :logical	GEOM_VALIDITY: 40
NA's :3990	FALSE:40	NA's :3990
	TRUE :3990	

```
clgeo_SuspiciousFeatures(analise_topologica)
```

```
[1] 2 40 52 84 936 2140 2141 2559 3797 3798 3802 3803 3805 3806 3817 3818 3819 3855 3856  
[20] 3857 3858 3865 3866 3876 3892 3893 3913 3914 3940 3941 3944 3945 3956 3958 3959 3992 3993 3998  
[39] 3999 4020
```

# Correção topológica

```
setores_clean <- clgeo_Clean(setores_sp)  
clgeo_IsValid(setores_clean)
```

```
[1] TRUE
```

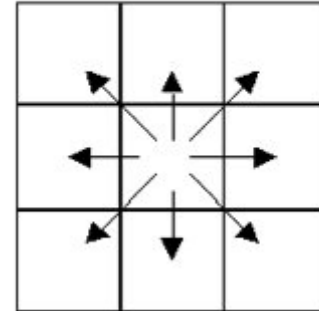
# CRITÉRIOS DE VIZINHANÇA

## ■ CONTIGUIDADE:

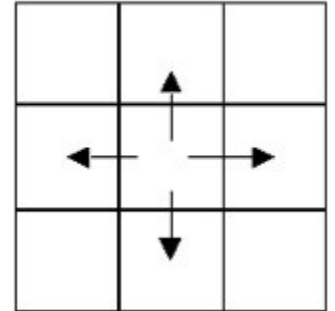
- Rook (torre)
- Queen (rainha)

(para polígonos ou raster)

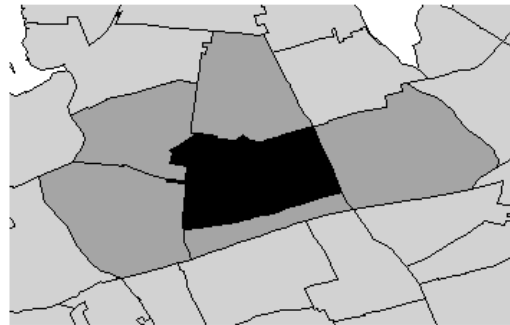
Queen's (Kings) Case



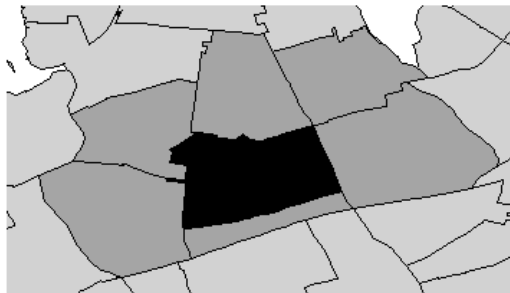
Rooks Case



Rook

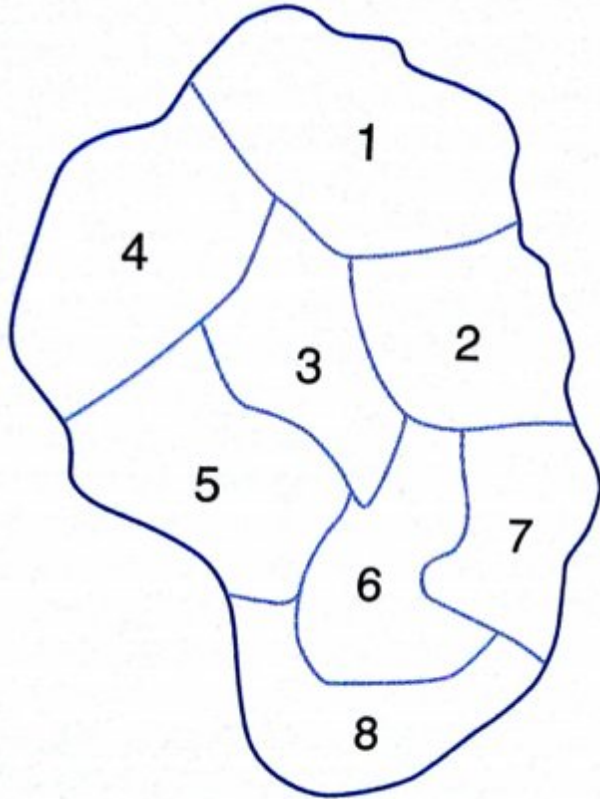


Queen



# MATRIZ DE VIZINHANÇA

Critério de contiguidade



	1	2	3	4	5	6	7	8
1	0	1	1	1	0	0	0	0
2	1	0	1	0	0	1	1	0
3	1	1	0	1	1	1	0	0
4	1	0	1	0	1	0	0	0
5	0	0	1	1	0	1	0	1
6	0	1	1	0	1	0	1	1
7	0	1	0	0	0	1	0	1
8	0	0	0	0	1	1	1	0

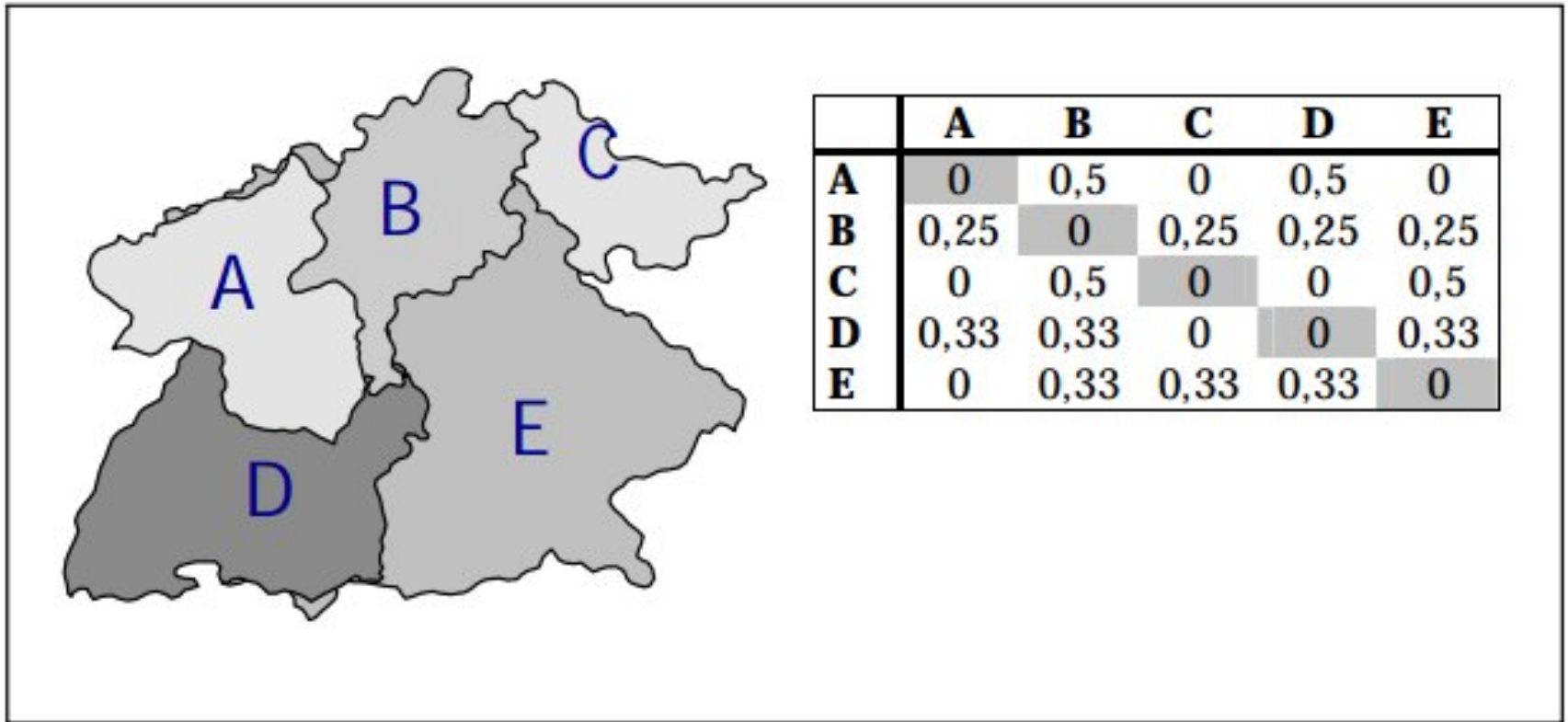
**MATRIZ DE PESOS**

A matriz pode ser binária (1 ou 0, como no exemplo) ou geral (pesos contínuos, dependendo da distância – mais próximo com peso maior, é “mais vizinho”).

A matriz também pode ser normalizada.

# MATRIZ DE VIZINHANÇA

## MATRIZ NORMALIZADA



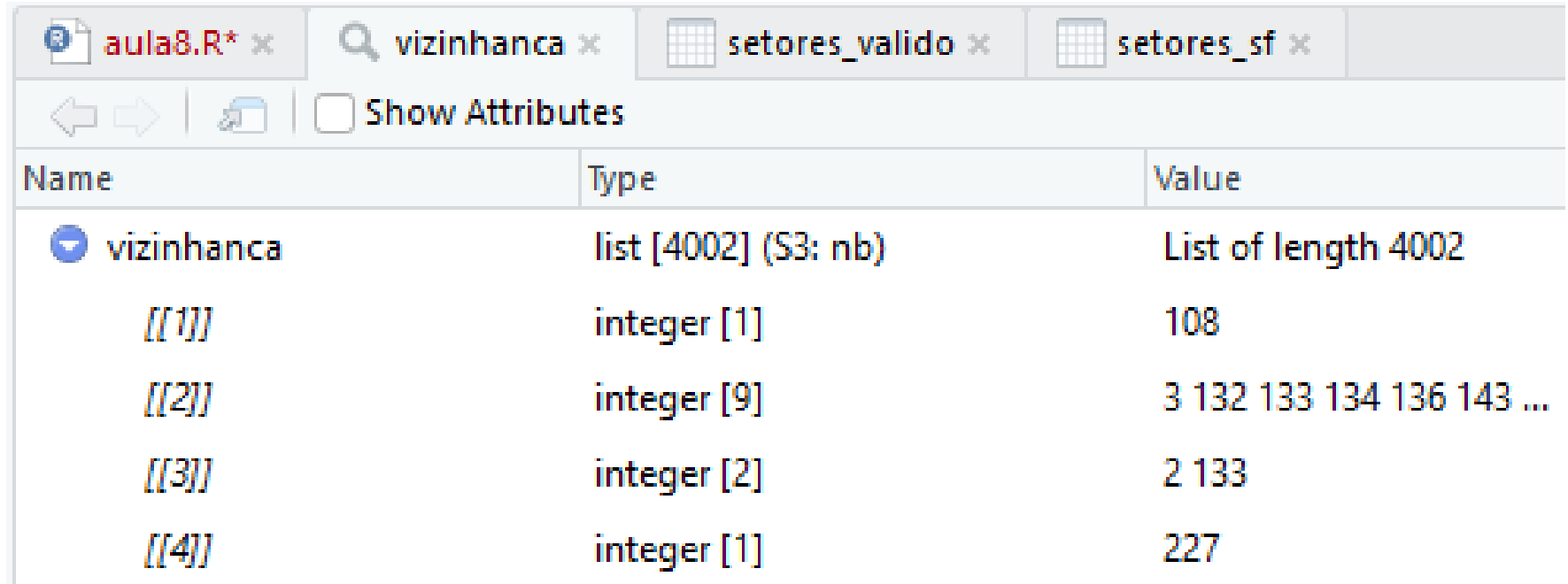
# Relações de vizinhança

```
install.packages("spdep")
```

```
library(spdep)
```

```
vizinhanca <- poly2nb(setores_clean)
```

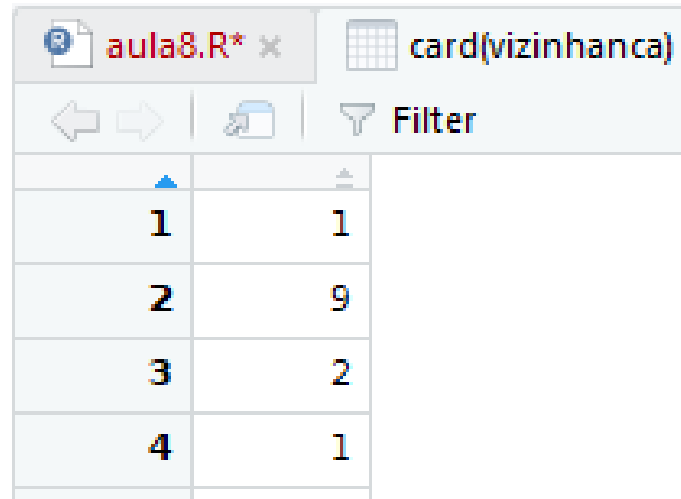
```
View(vizinhanca)
```



The screenshot shows the RStudio interface with the 'vizinhanca' object selected in the Environment pane. The object is a list of length 4002, containing neighborhood relationships. The first four elements are shown:

Name	Type	Value
vizinhanca	list [4002] (S3: nb)	List of length 4002
[[1]]	integer [1]	108
[[2]]	integer [9]	3 132 133 134 136 143 ...
[[3]]	integer [2]	2 133
[[4]]	integer [1]	227

View(card(vizinhanca))



1	1
2	9
3	2
4	1

```
setores_clean$vizinhos <- card(vizinhanca)
```



View(setores\_clean@data)

Ordenar por ordem crescente de vizinhos

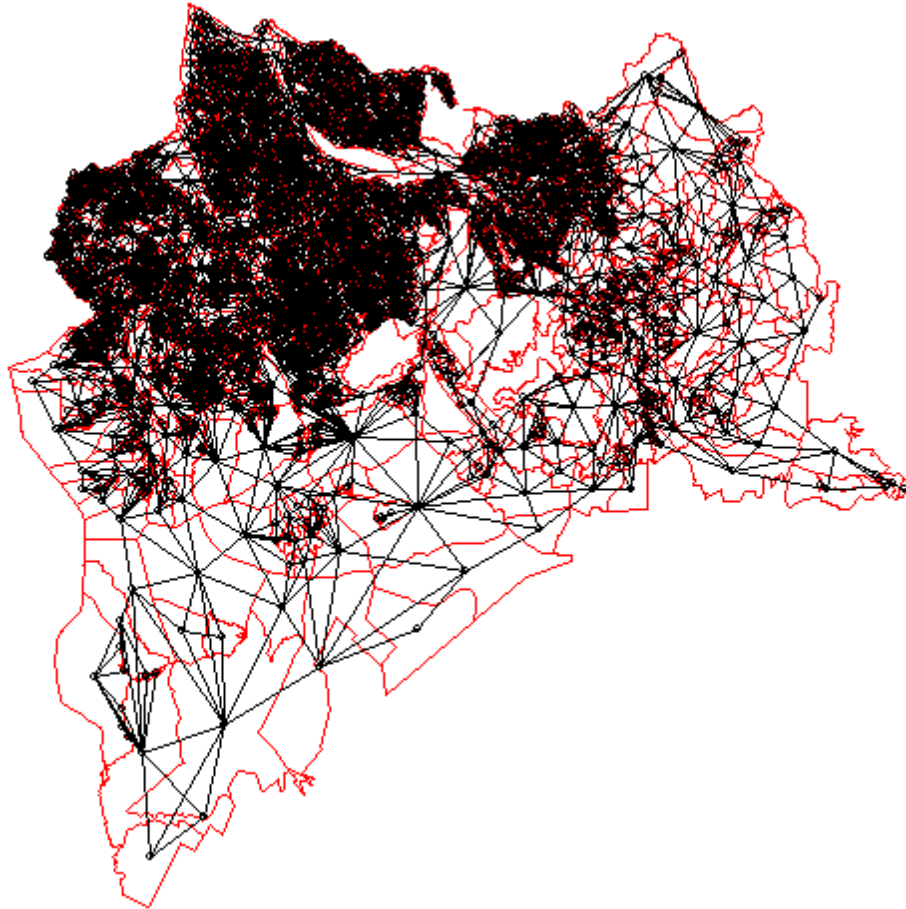


CRO	NM_MESO	Tipo_setor	Domicilios	Pessoas	Renda	rede_esg	col_lixo	vizinhos
JLO	METROPOLITANA DE S?O PAULO	0	135	470	617.10	5.925926	84.44444	0
JLO	METROPOLITANA DE S?O PAULO	0	420	1197	3919.24	95.952381	100.00000	0
JLO	METROPOLITANA DE S?O PAULO	0	117	387	816.20	100.000000	100.00000	0
JLO	METROPOLITANA DE S?O PAULO	0	152	539	913.04	99.342105	100.00000	0
JLO	METROPOLITANA DE S?O PAULO	0	60	221	294.62	98.333333	100.00000	0
JLO	METROPOLITANA DE S?O PAULO	0	216	840	492.69	100.000000	100.00000	0
JLO	METROPOLITANA DE S?O PAULO	0	175	450	2918.63	100.000000	99.42857	0
JLO	METROPOLITANA DE S?O PAULO	1	91	345	682.13	97.802198	100.00000	1
JLO	METROPOLITANA DE S?O PAULO	1	416	1402	792.14	99.759615	99.75962	1
JLO	METROPOLITANA DE S?O PAULO	1	91	333	350.78	97.802198	100.00000	1
JLO	METROPOLITANA DE S?O PAULO	0	177	508	2311.30	100.000000	100.00000	1

setores\_juntos <- subset(setores\_clean, setores\_clean\$vizinhos != 0)  
View(setores\_juntos@data)

vizinhos
1
1
1
1

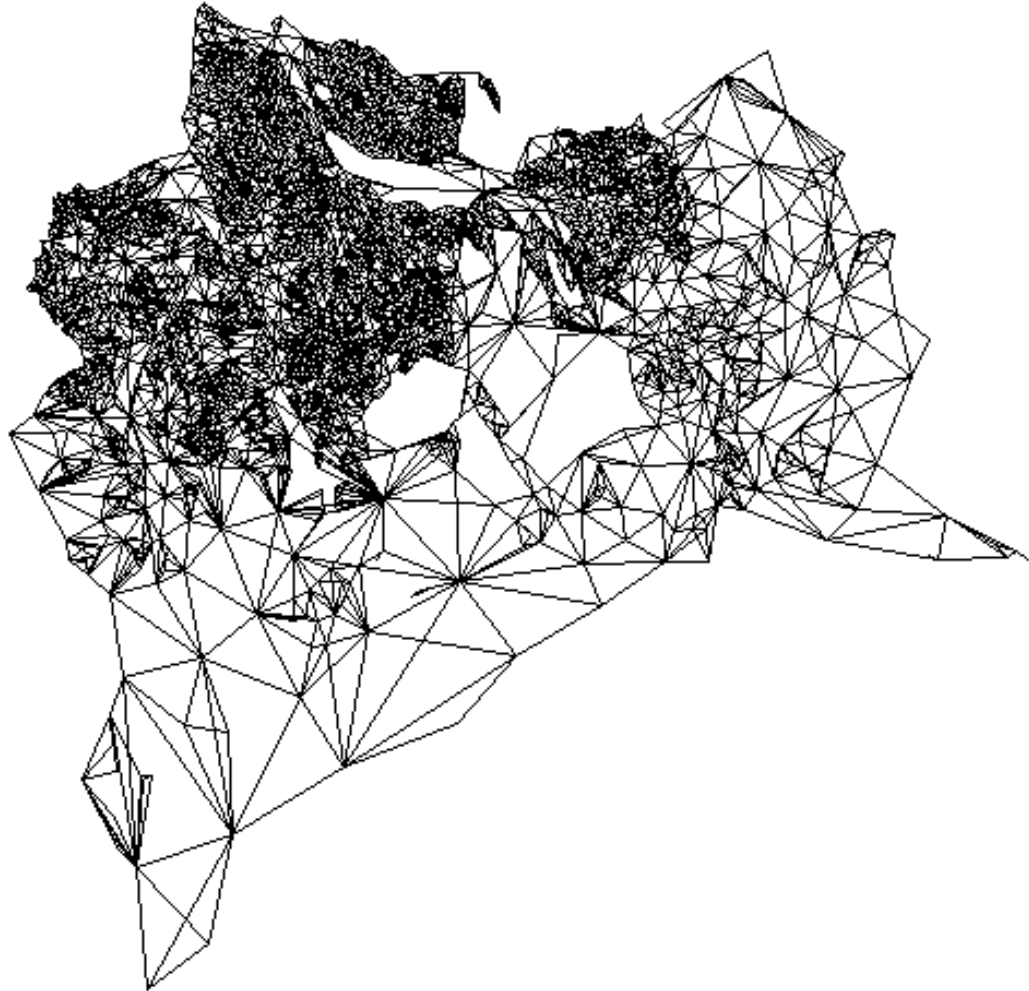
```
vizinhanca2 <- poly2nb(setores_juntos)
setores_xy <- coordinates(setores_juntos)
plot(setores_sp, border="red")
plot(x= vizinhanca2, coord = setores_xy, cex=0.6, add=TRUE)
```



# Exportando grafo de vizinhança para sp

```
vizinhanca_sp <- nb2lines(vizinhanca2, coords = setores_xy,  
proj4string = crs(setores_valido))
```

```
plot(vizinhanca_sp)
```



# Vizinhança normalizada (pesos)

```
vizinhanca_pesos <- nb2listw(vizinhanca2)
```

```
View(vizinhanca_pesos)
```

aula8.R* x vizinhanca_pesos x		
◀ ▶ 📄 <input type="checkbox"/> Show Attributes		
Name	Type	Value
▼ vizinhanca_pesos	list [3] (S3: listw, nb)	List of length 3
style	character [1]	'W'
▶ neighbours	list [3995] (S3: nb)	List of length 3995
▼ weights	list [3995]	List of length 3995
[[1]]	double [1]	1
[[2]]	double [9]	0.111 0.111 0.111 0.111 0.111 0.111 ...
[[3]]	double [2]	0.5 0.5

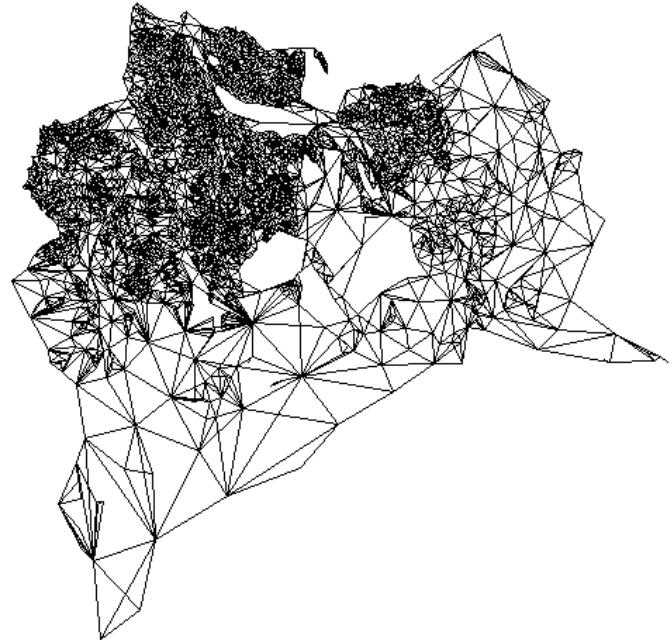
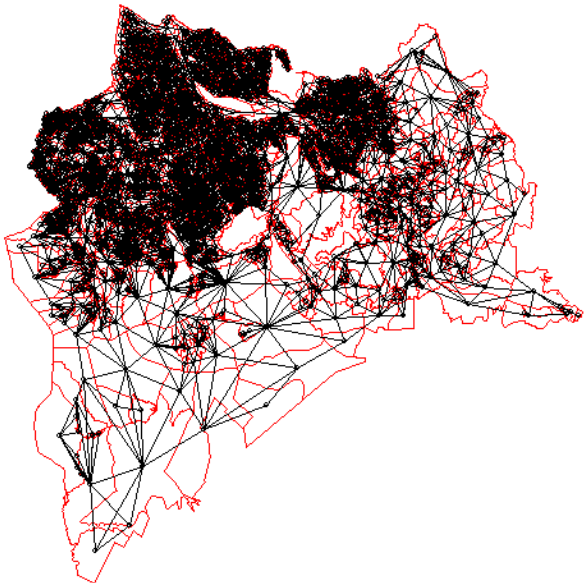
# Vizinhança normalizada (pesos)

```
plot(setores_sp, border="red")
```

```
plot(vizinhanca_pesos, setores_xy, cex=0.6, add=TRUE)
```

```
vizinhanca_pesos_sp <- listw2lines(vizinhanca_pesos,  
coords = setores_xy, proj4string = crs(setores_sp))
```

```
plot(vizinhanca_pesos_sp)
```



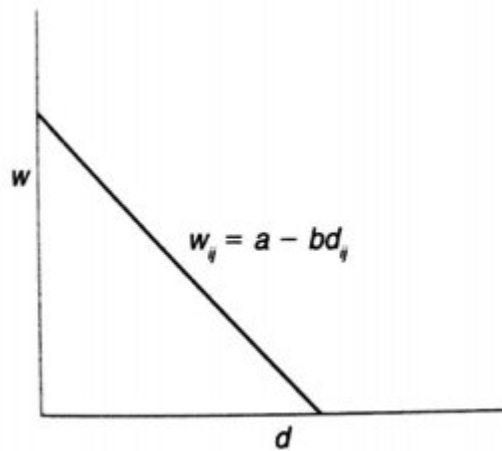
# Exercício 1

Fazer verificação de correção topológica, vizinhança e vizinhança normalizada para os municípios da RMSP.

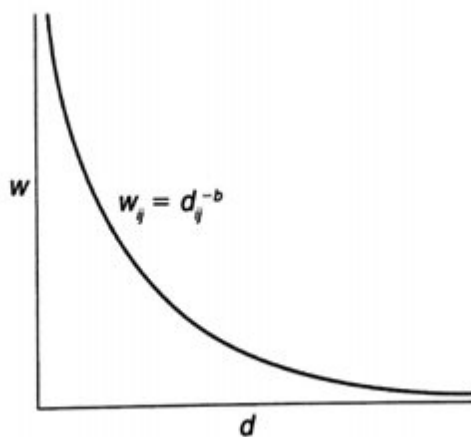
Visualizar grafo de vizinhança como mapa.

# Vizinhança por distância

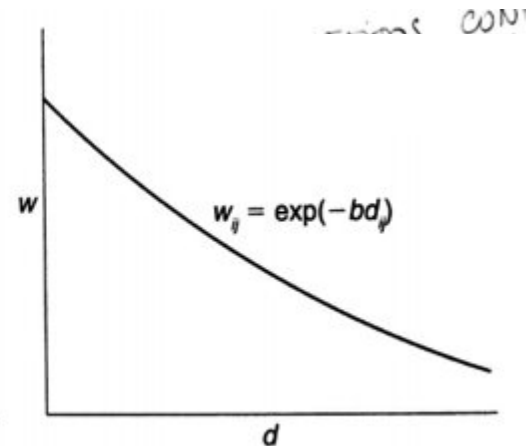
- Pode utilizar bases pontuais ou centróides de polígonos
- Métodos:
  - “n” vizinhos mais próximos
  - distância máxima
  - funções de distância ( $w_{ij}$  com valores contínuos)



DECAIMENTO LINEAR



DECAIMENTO  
EXPONENCIAL NEGATIVO  
DA DISTÂNCIA

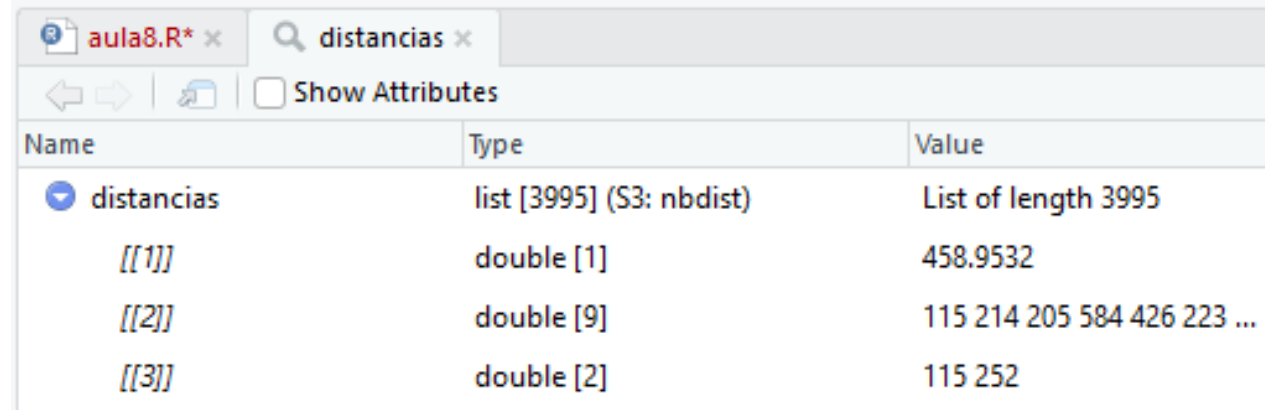


DECAIMENTO NEGATIVO  
EXPONENCIAL DA  
DISTÂNCIA

# Análise de Distâncias de Vizinhança

```
distancias <- nbdists(vizinhanca2, setores_xy)
```

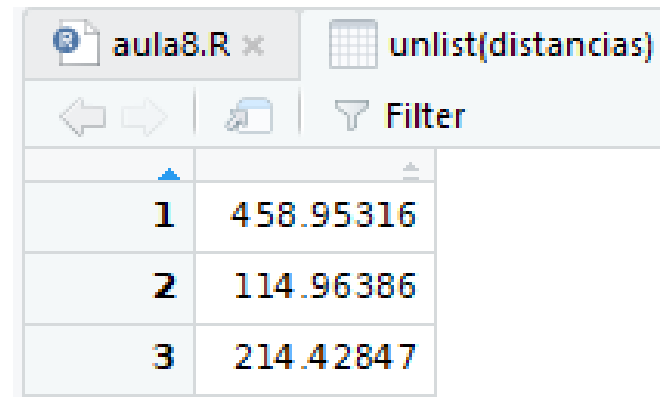
```
View(distancias)
```



The screenshot shows the RStudio Environment pane with the 'distancias' object. The object is a list of length 3995, created by the 'nbdist' function. The first three elements are expanded, showing their types and values.

Name	Type	Value
distancias	list [3995] (S3: nbdist)	List of length 3995
[[1]]	double [1]	458.9532
[[2]]	double [9]	115 214 205 584 426 223 ...
[[3]]	double [2]	115 252

```
View(unlist(distancias))
```



The screenshot shows the RStudio Environment pane with the 'unlist(distancias)' object. The object is a numeric vector. The first three elements are expanded, showing their values.

	Value
1	458.95316
2	114.96386
3	214.42847

```
summary(unlist(distancias))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0	194.2	283.9	398.7	423.9	7526.6



# Vizinhos por raio de distância

```
vizinhanca_400m <- dnearneigh(setores_xy, d1=0, d2=400)
```

```
View(vizinhanca_400m)
```

**Distância**   **Distância**  
**mínima**   **máxima**

aula8.R x vizinhanca_400m x		
◀ ▶   📄 <input type="checkbox"/> Show Attributes		
Name	Type	Value
▼ vizinhanca_400m	list [3995] (S3: nb)	List of length 3995
[[1]]	integer [3]	105 107 133
[[2]]	integer [12]	3 128 129 131 137 138 ...

# Vizinhos por raio de distância

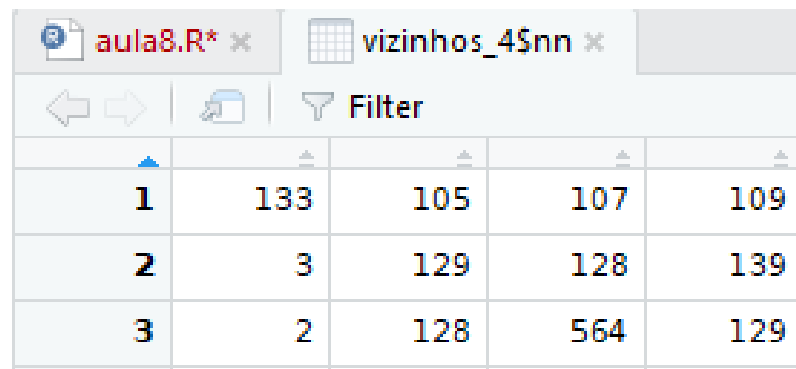
```
plot(vizinhanca_400m, setores_xy, cex=0.3)
```



# K vizinhos mais próximos

```
vizinhos_4 <- knearneigh(setores_xy, k = 4)
```

```
View(vizinhos_4$nn)
```

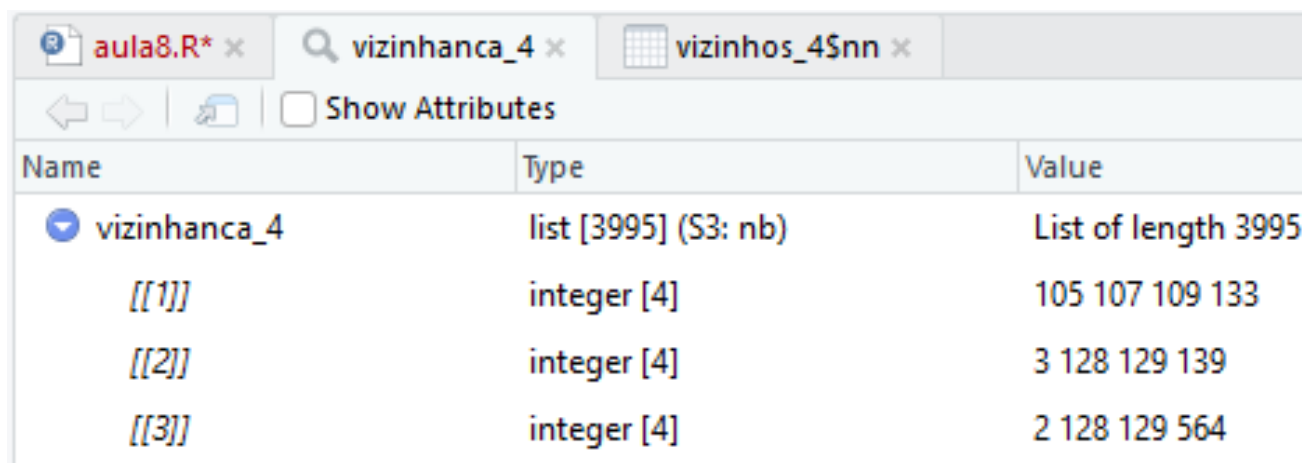


The screenshot shows the RStudio interface with the 'vizinhos\_4\$nn' data frame selected. The data is displayed in a table with 5 columns and 3 rows. The first column contains indices 1, 2, and 3. The other columns contain the nearest neighbors for each point.

1	133	105	107	109
2	3	129	128	139
3	2	128	564	129

```
vizinhanca_4 <- knn2nb(vizinhos_4)
```

```
View(vizinhanca_4)
```



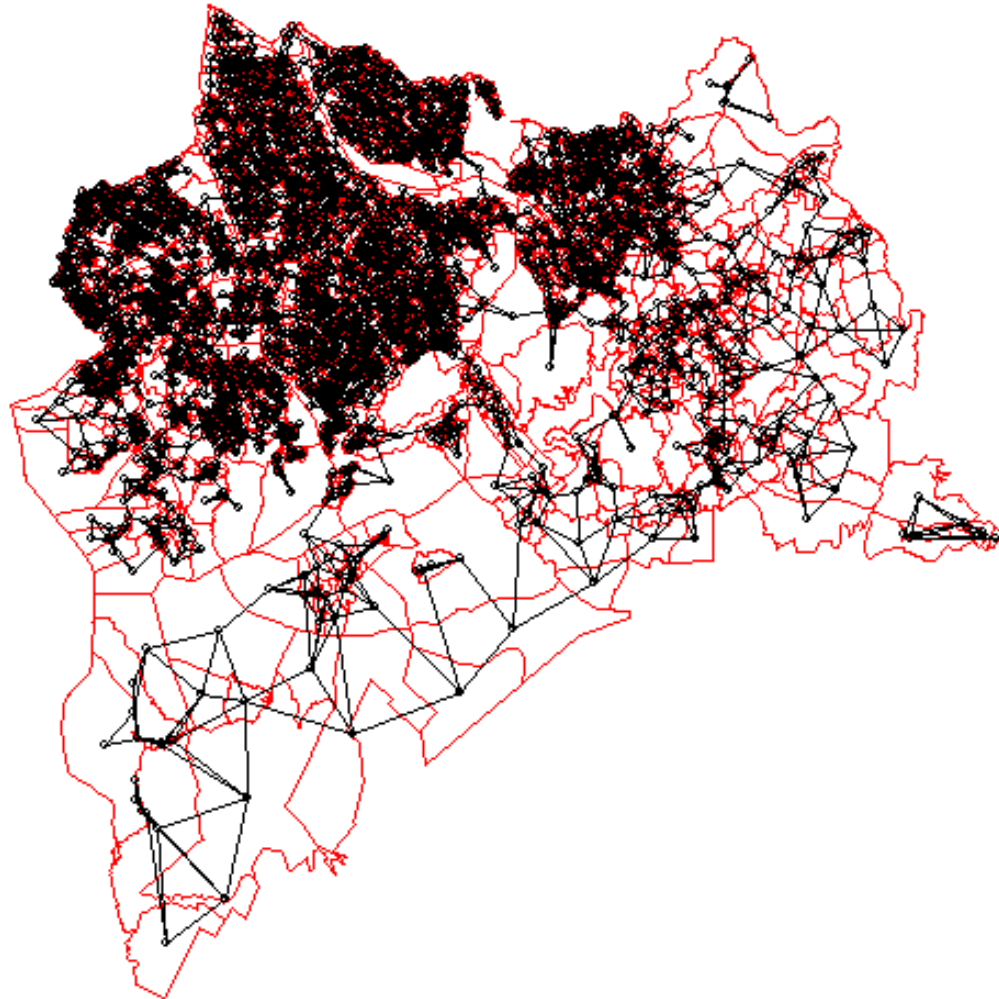
The screenshot shows the RStudio interface with the 'vizinhanca\_4' data frame selected. The data is displayed in a table with 3 columns: Name, Type, and Value. The first column contains the variable name 'vizinhanca\_4' and its elements '[[1]]', '[[2]]', and '[[3]]'. The second column shows the type 'list [3995] (S3: nb)' and 'integer [4]'. The third column shows the values 'List of length 3995' and the neighbor indices for each point.

Name	Type	Value
vizinhanca_4	list [3995] (S3: nb)	List of length 3995
[[1]]	integer [4]	105 107 109 133
[[2]]	integer [4]	3 128 129 139
[[3]]	integer [4]	2 128 129 564

# K vizinhos mais próximos

```
plot(setores_sp, border="red")
```

```
plot(vizinhanca_4, setores_xy, cex=0.6, add=TRUE)
```



# Exercício 2

Fazer análise de distâncias, vizinhança com o raio de distância media e com 3 vizinhos mais próximos dos municípios da RMSP, visualizando os mapas.

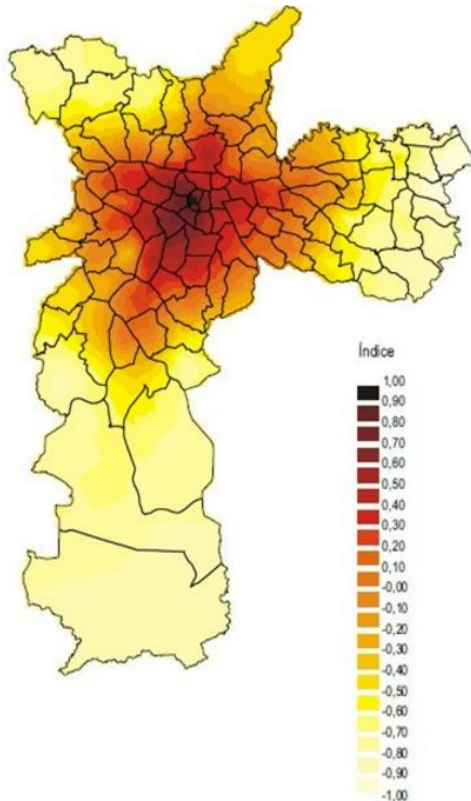
# Conteúdo

- Análise de vizinhança
- **Autocorrelação espacial**
- Suavização espacial
- Regressão espacial
- Agrupamento

# Dependência espacial

“As coisas mais próximas se parecem mais entre si do que as mais distantes” – Waldo Tobler (1970)

Mapa de inclusão/exclusão social em São Paulo

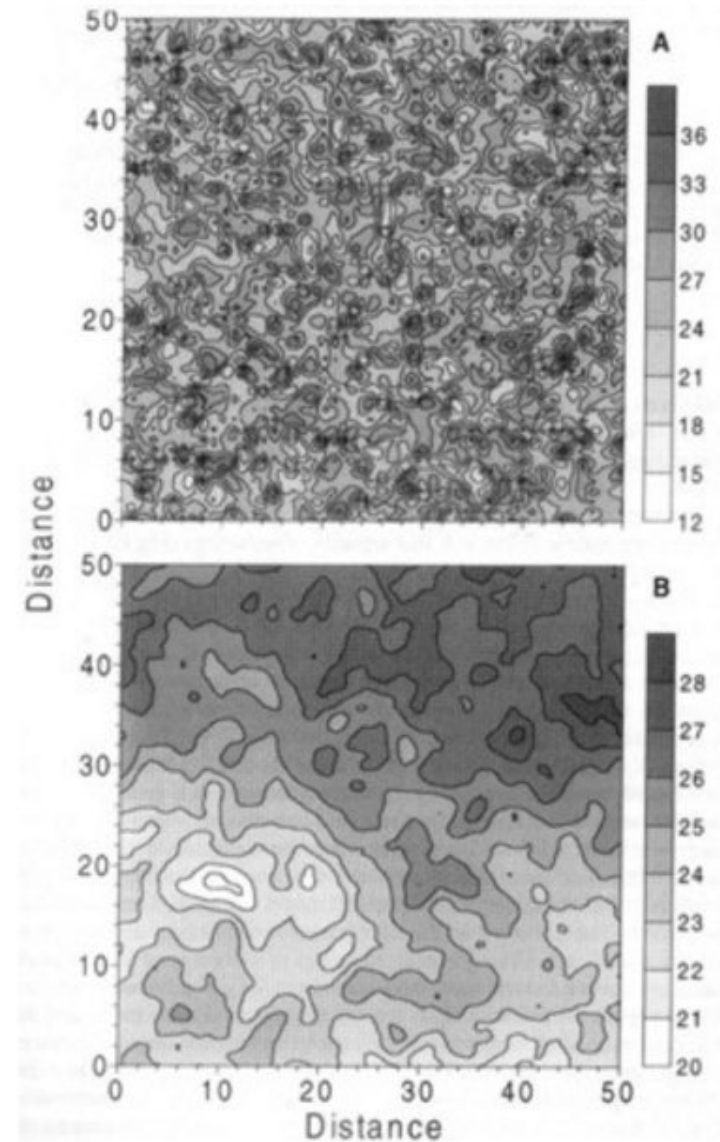


- Auto-correlação espacial (grau de dependência espacial)

Tobler, W. R. 1970. A computer movie simulating urban growth in the Detroit region. *Economic Geography* 46: 234-40.

# Dependência espacial

- Qual dos mapas ao lado tem maior autocorrelação espacial?





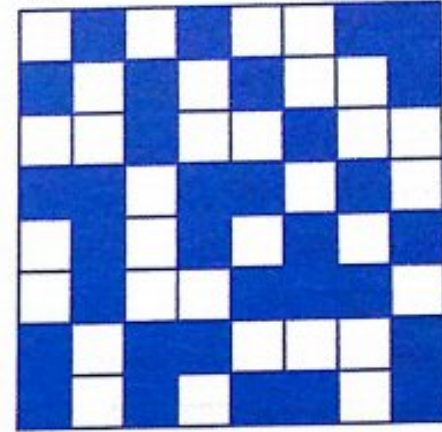
# Autocorrelação Espacial

- **Autocorrelação Positiva (Lei de Tobler):**  
Feições similares em localização também são similares em atributos
- **Autocorrelação Negativa (oposição à Lei de Tobler):**  
Feições similares em localização tendem a ter atributos menos similares do que feições mais distantes
- **Ausência de Autocorrelação:**  
Quando atributos são independentes da localização

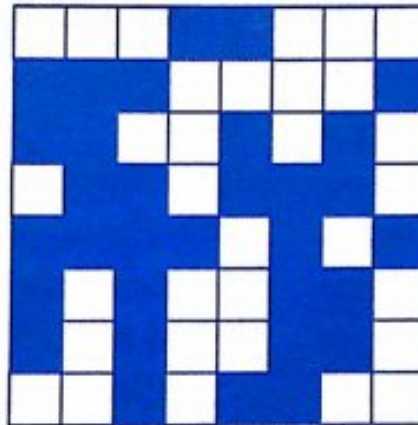
**EXTREMA  
AUTOCORRELAÇÃO  
NEGATIVA**



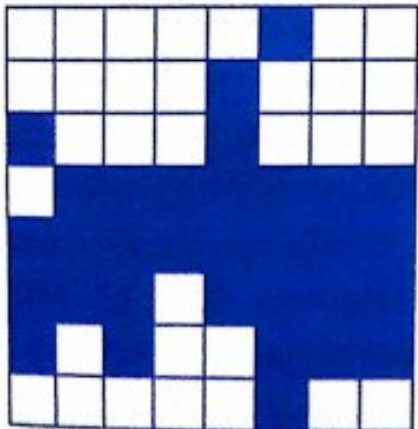
**ARRANJO  
DISPERSO**



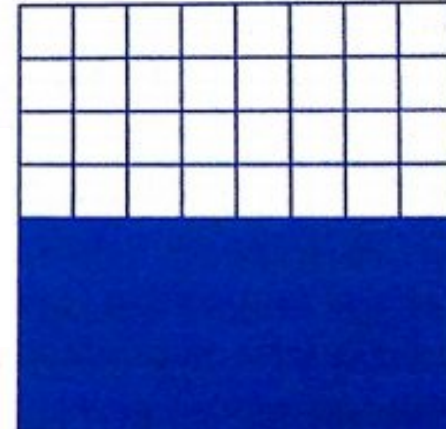
**INDEPENDÊNCIA  
ESPACIAL**



**AGRUPAMENTO  
ESPACIAL**



**EXTREMA  
AUTOCORRELAÇÃO  
ESPACIAL POSITIVA**



# Índices de autocorrelação espacial

## 1. Índices Globais de Associação Espacial

- Apresenta uma medida única para toda a área analisada.
- Índice global de Moran ( $I$ )

## 2. Índices Locais de Associação Espacial (LISA)

- Decomposições dos índices globais, podem ser visualizados na forma de mapas.
- Permite a identificação de diferentes regimes de associação espacial
- Índice local de Moran ( $I_i$ )

# Índice Global de Moran

- Índice global de autocorrelação espacial, que varia entre -1 e 1
- Correlação de um atributo de um elemento no espaço, em relação ao mesmo atributo nos vizinhos

- **$I = 1 \rightarrow$  Extrema Autocorrelação Positiva (Lei de Tobler):** Feições similares em localização também são similares em atributos

- **$I = -1 \rightarrow$  Extrema Autocorrelação Negativa (oposição à Lei de Tobler):**

Feições similares em localização tendem a ter atributos menos similares do que feições mais distantes

- **$I = 0 \rightarrow$  Ausência de Autocorrelação:**

Quando atributos são independentes da localização

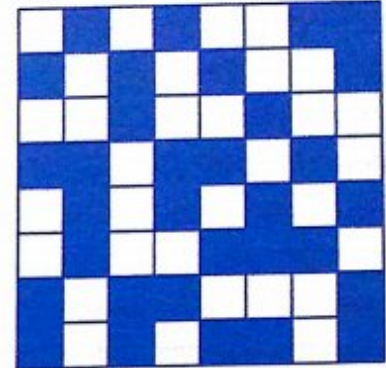
# Índice Global de Moran

(A) **EXTREMA AUTOCORRELAÇÃO NEGATIVA**



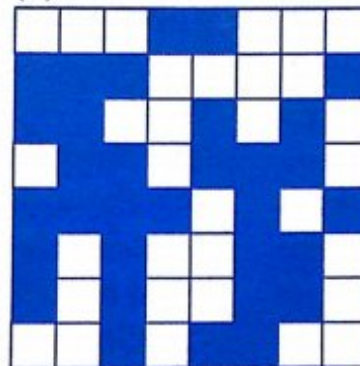
$$I = -1,000$$

(B) **ARRANJO DISPERSO**



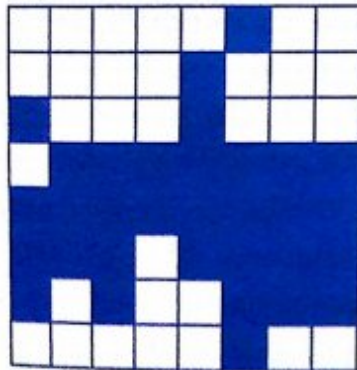
$$I = -0,393$$

(C) **INDEPENDÊNCIA ESPACIAL**



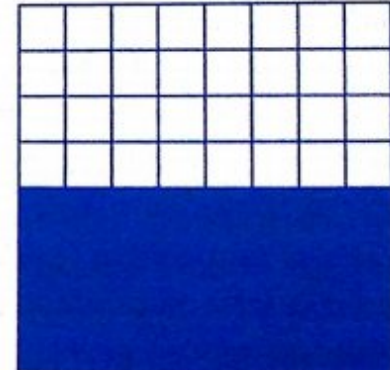
$$I = 0,000$$

(D) **AGRUPAMENTO ESPACIAL**



$$I = +0,393$$

(E) **EXTREMA AUTOCORRELAÇÃO ESPACIAL POSITIVA**



$$I = +0,857$$

# Índice global de Moran

```
moran.test(x = setores_juntos$Renda, listw = vizinhanca_pesos)
```

Moran I statistic standard deviate = 63.972, p-value < 2.2e-16  
alternative hypothesis: greater  
sample estimates:

Moran I statistic	Expectation	Variance
0.6209651089	-0.0002503756	0.0000942998

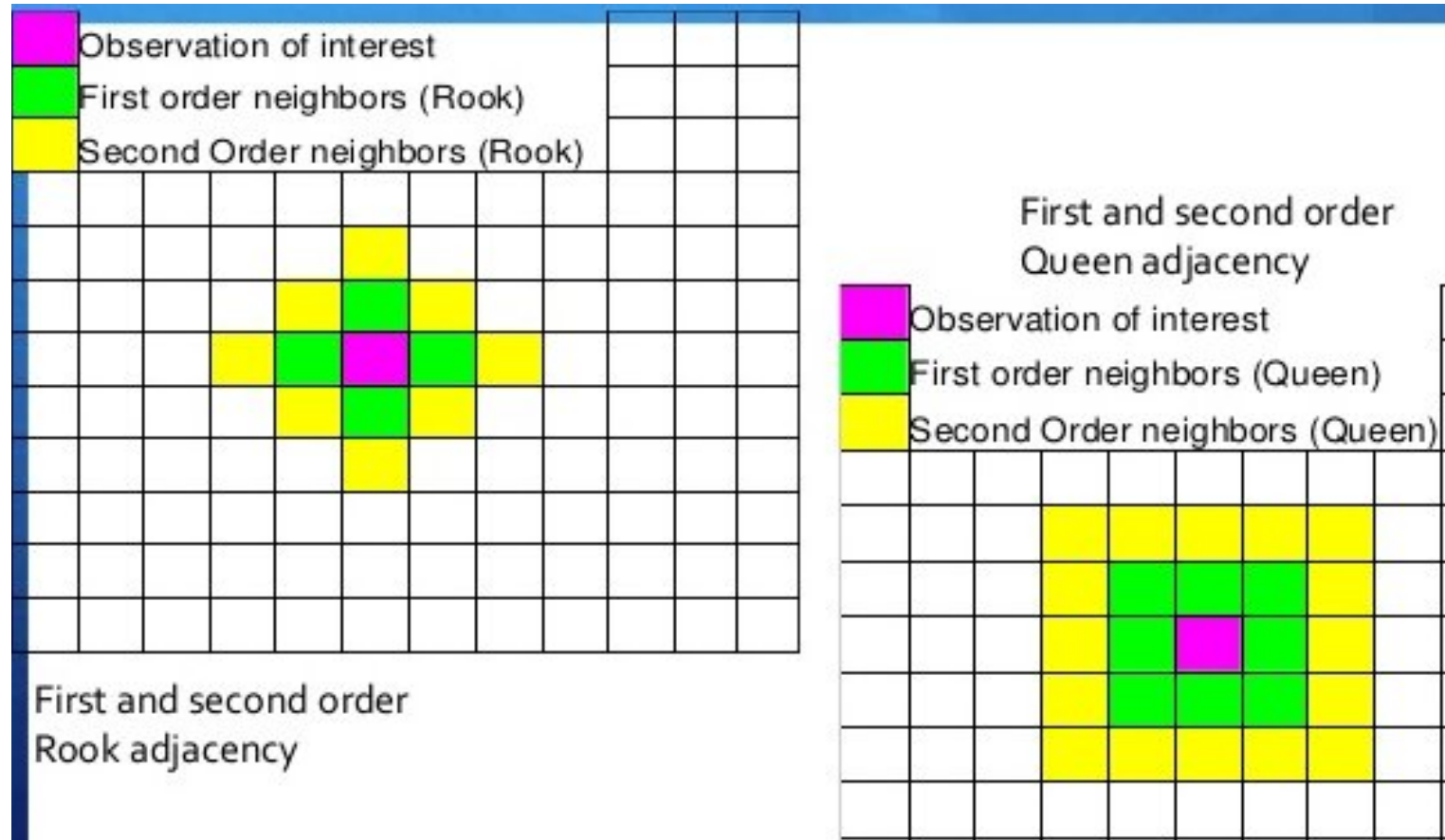
**p<0.05, então  
é significativo**

**Moran I > 0,  
autocorrelação  
espacial positiva**

# Correlograma de Índice de Moran

`correlograma_contiguidade <- sp.correlogram(neighbours = vizinhanca2, var = setores_juntos$Renda, order = 5, method = "I")` **I = Índice de Moran**

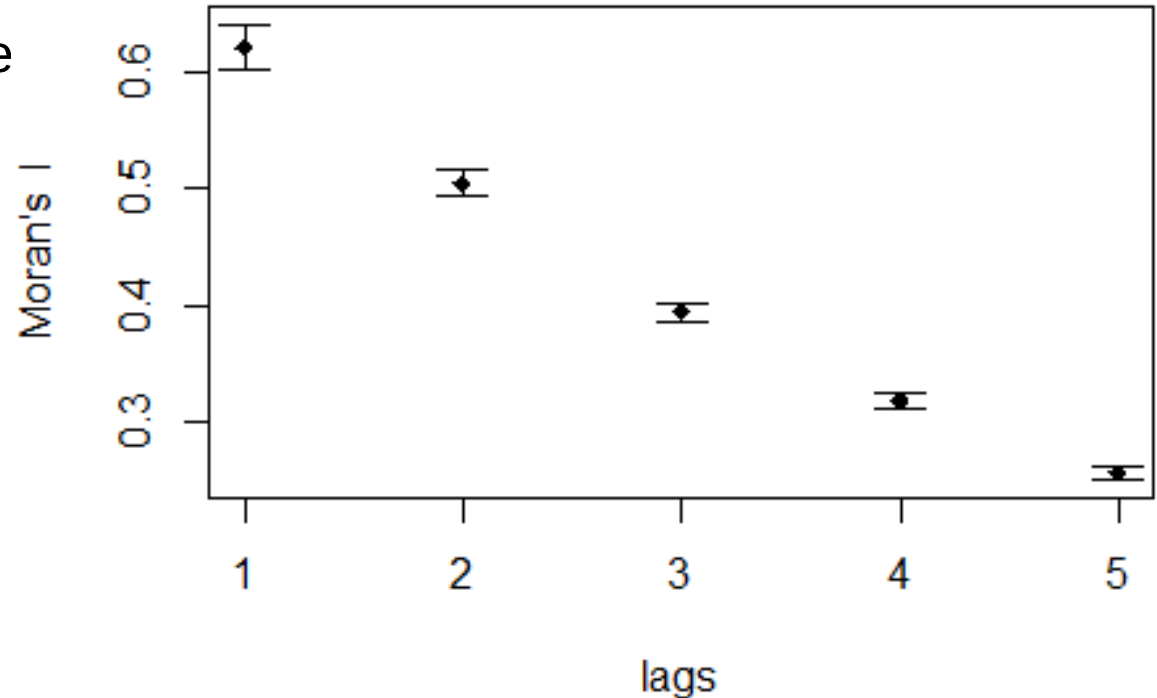
**Até qual ordem  
de contiguidade**



# Correlograma de Índice de Moran

```
plot(correlograma_contiguidade)
```

correlograma\_contiguidade



	estimate	expectation	variance	standard deviate	Pr(I) two sided
1 (3995)	6.2097e-01	-2.5038e-04	9.4300e-05	63.972	< 2.2e-16 ***
2 (3995)	5.0449e-01	-2.5038e-04	3.2272e-05	88.850	< 2.2e-16 ***
3 (3995)	3.9445e-01	-2.5038e-04	1.6690e-05	96.615	< 2.2e-16 ***
4 (3995)	3.1846e-01	-2.5038e-04	1.0376e-05	98.943	< 2.2e-16 ***
5 (3995)	2.5703e-01	-2.5038e-04	7.0927e-06	96.607	< 2.2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1



# Correlograma de Distância

## NÃO RODAR, DEMORA MUITO!

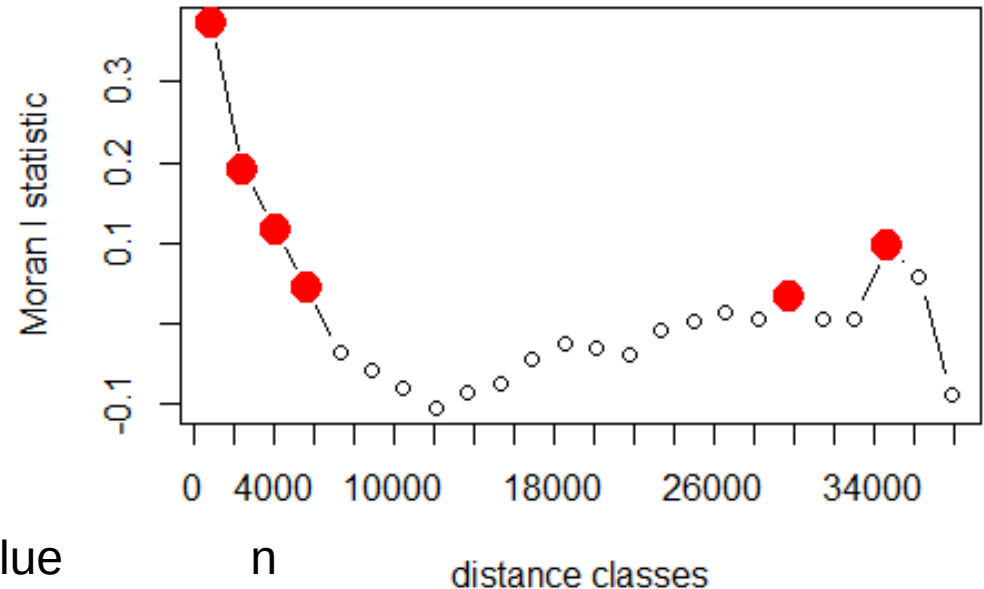
```
install.packages("pgirmess")
```

```
library(pgirmess)
```

```
correlograma_distancia <- correlog(setores_xy, setores_juntos$Renda)
```

```
plot(correlograma_distancia)
```

```
correlograma_distancia
```



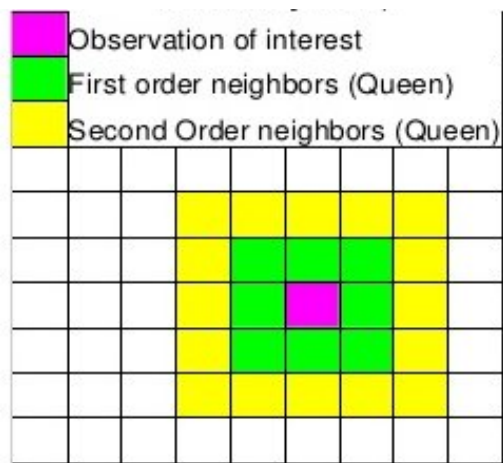
Moran I statistic

	dist.class	coef	p.value	n
[1,]	806.6981	0.372223766	0.000000e+00	464334
[2,]	2420.0942	0.190925123	0.000000e+00	1118650
[3,]	4033.4904	0.117576367	0.000000e+00	1580616
[4,]	5646.8865	0.044909747	8.516147e-242	1845730
[5,]	7260.2827	-0.035684296	1.000000e+00	1996306

# Vizinhança acumulada de ordens superiores

```
vizinhanca_1e2 <- nblag(vizinhanca2, maxlag = 2)
```

```
View(vizinhanca_1e2)
```



aula8.R* x vizinhanca_1e2 x		
Show Attributes		
Name	Type	Value
vizinhanca_1e2	list [2]	List of length 2
[[1]]	list [3995] (S3: nb)	List of length 3995
[[2]]	list [3995] (S3: nb)	List of length 3995
[[1]]	integer [8]	89 92 105 130 133 396 ...
[[2]]	integer [23]	92 104 122 131 133 134 ...
[[3]]	integer [8]	128 130 131 132 139 141 ...

```
vizinhanca_ordem_2 <- nblag_cumul(vizinhanca_1e2)
```

```
View(vizinhanca_ordem_2)
```

aula8.R* x vizinhanca_ordem_2 x vizinhanca_1e2 x		
Show Attributes		
Name	Type	Value
vizinhanca_ordem_2	list [3995] (S3: nb)	List of length 3995
[[1]]	integer [9]	89 92 104 105 130 133 ...
[[2]]	integer [32]	3 92 104 122 128 129 ...
[[3]]	integer [10]	2 128 129 130 131 132 ...

# Vizinhança acumulada de ordens superiores

```
vizinhanca_peso_ordem_2 <- nb2listw(vizinhanca_ordem_2)
```

```
moran.test(x = setores_juntos$Renda, listw = vizinhanca_peso_ordem_2)
```

Moran I statistic standard deviate = 110.62, p-value < 2.2e-16  
alternative hypothesis: greater  
sample estimates:

Moran I statistic	Expectation	Variance
5.354296e-01	-2.503756e-04	2.344854e-05

**p<0.05, então  
é significativo**

**Autocorrelação  
positiva, mas  
menor do que o  
de 1ª ordem**

# Exercício 3

Calcule o Índice Global de Moran para a expectativa de vida nos municípios da RMSP e faça dois correlogramas, um de contiguidade (até 3ª ordem) e um de distância

# Indicadores Locais de Associação Espacial (LISA)

- Valor específico para cada objeto.
- Identificação de:
  - “Clusters”: objetos com valores de atributos semelhantes
  - “Outliers”: objetos anômalos em relação aos vizinhos
  - Regimes espaciais distintos

# Índice Local de Moran

- Como o atributo de um objeto está correlacionado ou não com os seus vizinhos

```
localmoran <- localmoran(x = setores_juntos$Renda, listw = vizinhanca_pesos)
```

```
View(localmoran)
```

	li	E.li	Var.li	Z.li	Pr(z > 0)
1	0.293399690	-0.0002503756	0.99619987	0.294209616	3.842989e-01
2	0.445880809	-0.0002503756	0.11046795	1.342284038	8.975196e-02
3	0.395813099	-0.0002503756	0.49797567	0.561255658	2.873116e-01

# Índice Local de Moran

```
class(localmoran)
```

```
[1] "localmoran" "matrix"
```

```
localmoran_df <- as.data.frame(localmoran)
```

```
setores_juntos$moran <- localmoran_df$li
```

```
setores_juntos$moran_p <- localmoran_df$`Pr(z > 0)`
```

```
View(setores_juntos@data)
```

moran	moran_p
0.293399690	3.842989e-01
0.445880809	8.975196e-02
0.395813099	2.873116e-01
0.309787440	3.780415e-01

# Índice Local de Moran

```
summary(setores_juntos$moran)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-2.98404	0.04022	0.22081	0.62096	0.44659	28.77329

```
tm_shape(setores_juntos) +
```

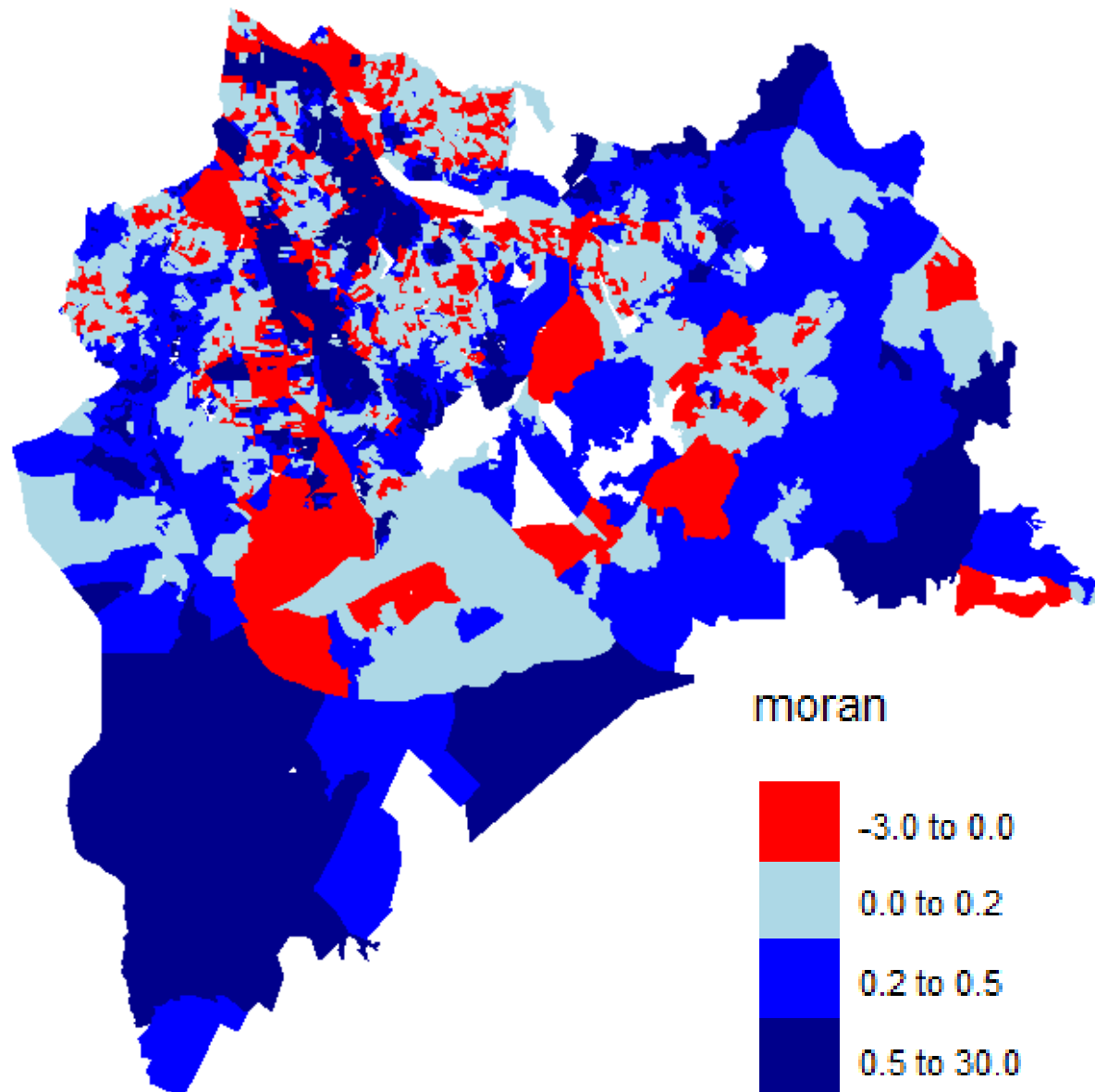
**Intervalos informados manualmente**

```
tm_fill("moran", style="fixed", breaks=c(-3, 0, 0.2, 0.5, 30),  
palette=c("red", "lightblue", "blue", "blue4"))
```

**Cores de cada intervalo**



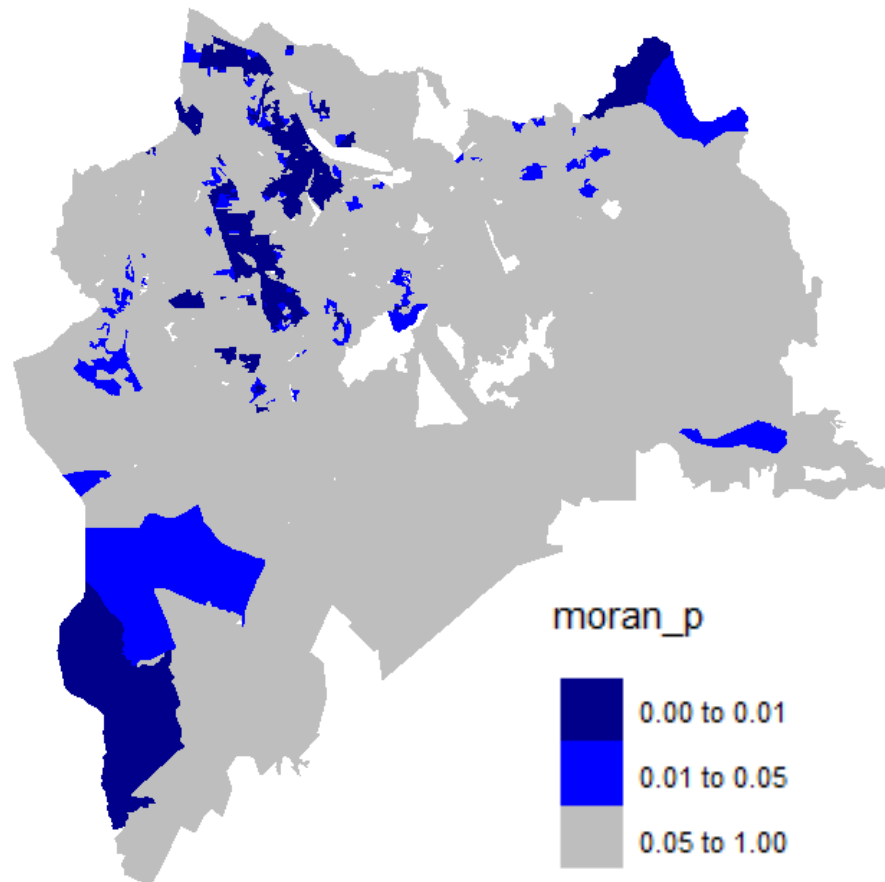
# Índice Local de Moran



# Índice Local de Moran

```
tm_shape(setores_juntos) +
```

```
  tm_fill(col="moran_p", style="fixed", breaks=c(0, 0.01, 0.05, 1),  
    palette=c("darkblue", "blue", "gray"))
```



# Diagrama de Espalhamento de Moran

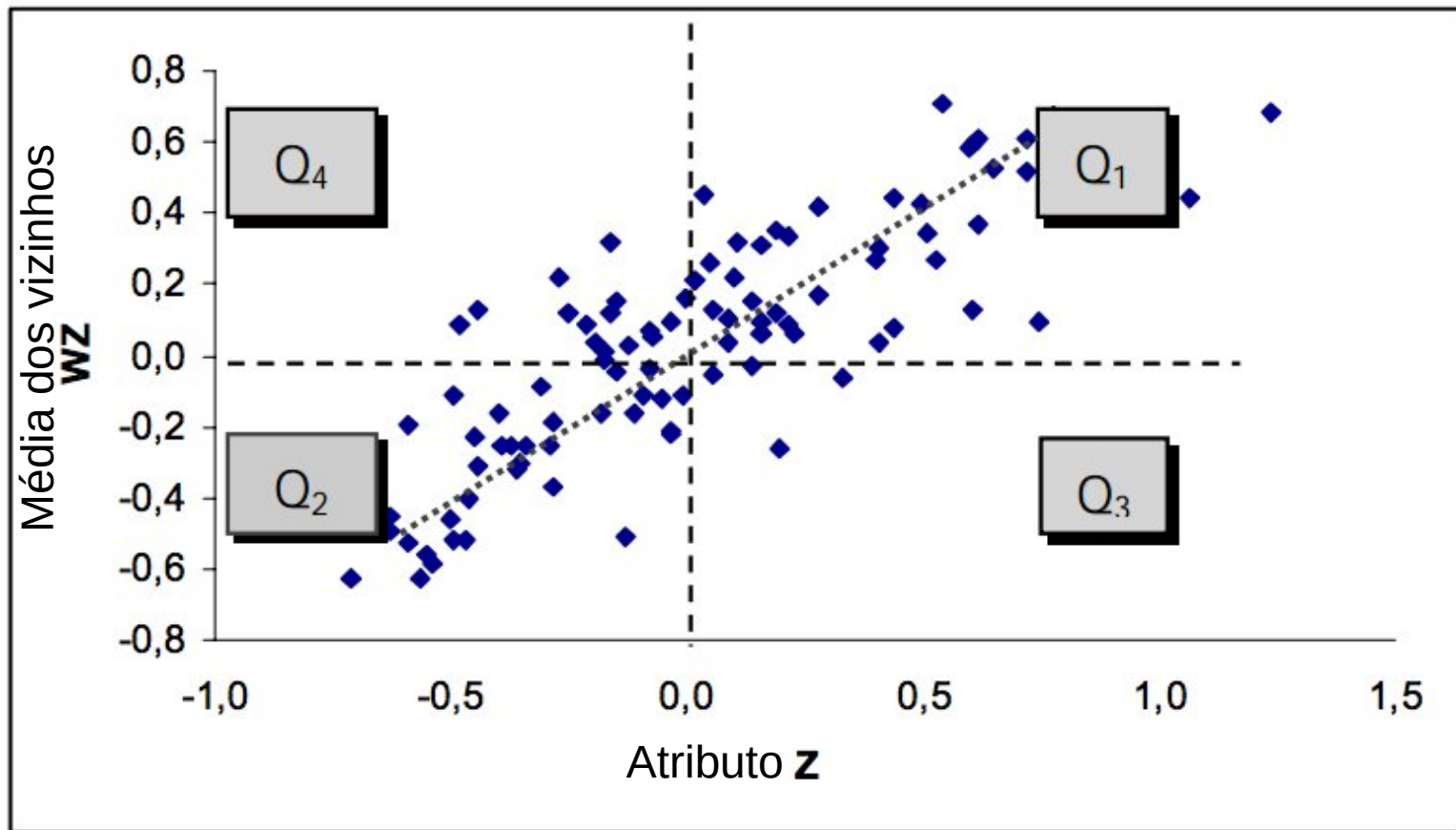
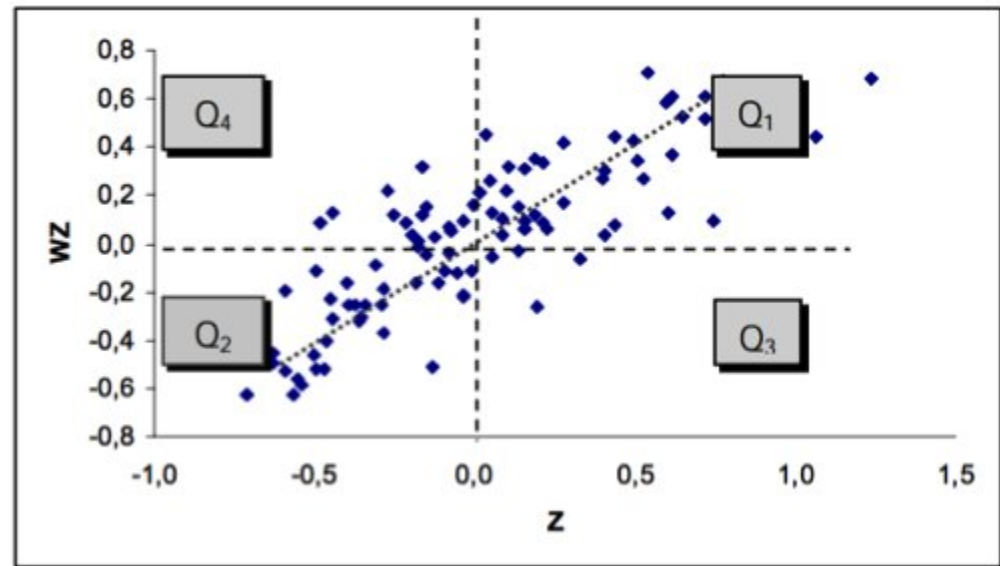


Diagrama de Espalhamento de Moran para o índice de exclusão/inclusão social de São Paulo, censo de 1991.

Anselin, Luc. 1996. "The Moran Scatterplot as an ESDA Tool to Assess Local Instability in Spatial Association." In *Spatial Analytical Perspectives on Gis in Environmental and Socio-Economic Sciences*, edited by Manfred Fischer, Henk Scholten, and David Unwin, 111–25. London: Taylor; Francis.

# Diagrama de Espalhamento de Moran



É dividido em **4 quadrantes**:

**Q<sub>1</sub>** (valores positivos, médias positivas) e

**Q<sub>2</sub>** (valores negativos, médias negativas):

indicam pontos de associação espacial positiva, no sentido que uma localização possui vizinhos com valores semelhantes.

**Q<sub>3</sub>** (valores positivos, médias negativas) e

**Q<sub>4</sub>** (valores negativos, médias positivas):

indicam pontos de associação espacial negativa, no sentido que uma localização possui vizinhos com valores distintos.

# Diagrama de Espalhamento de Moran

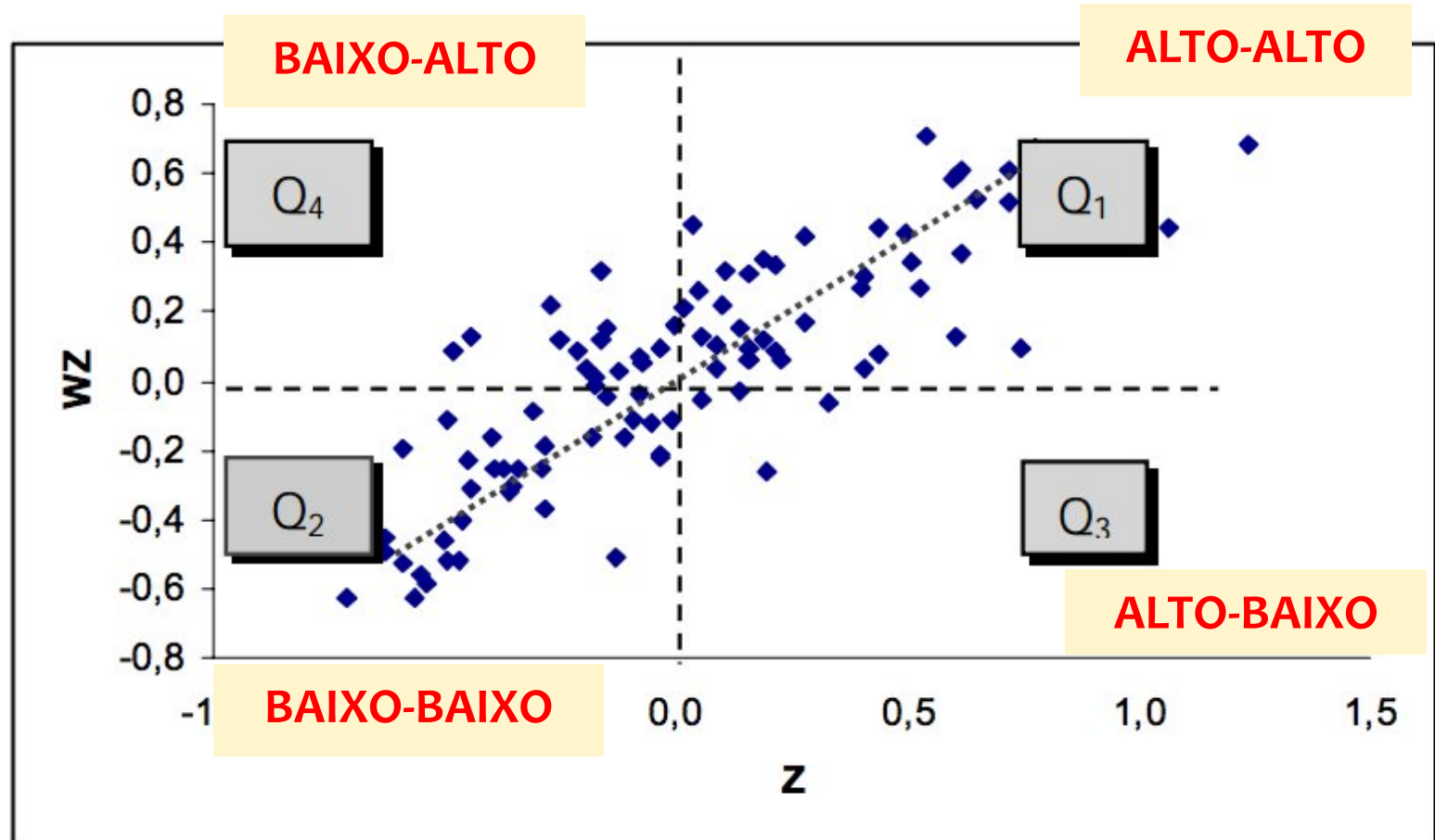
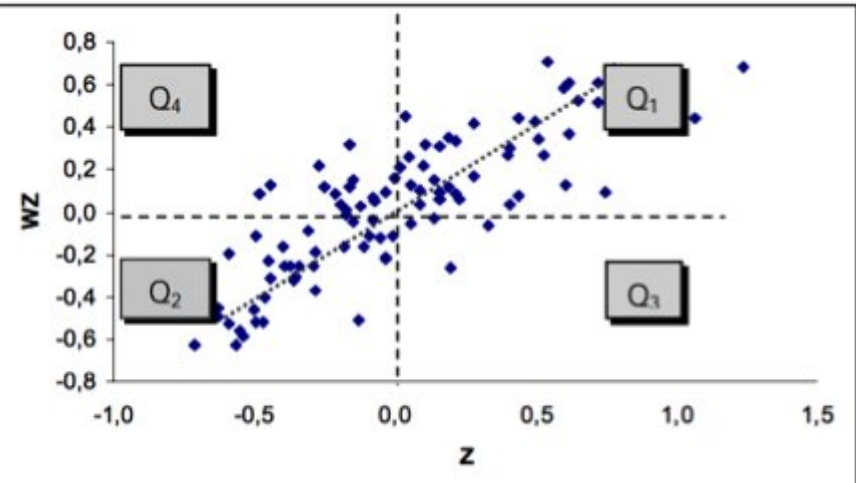
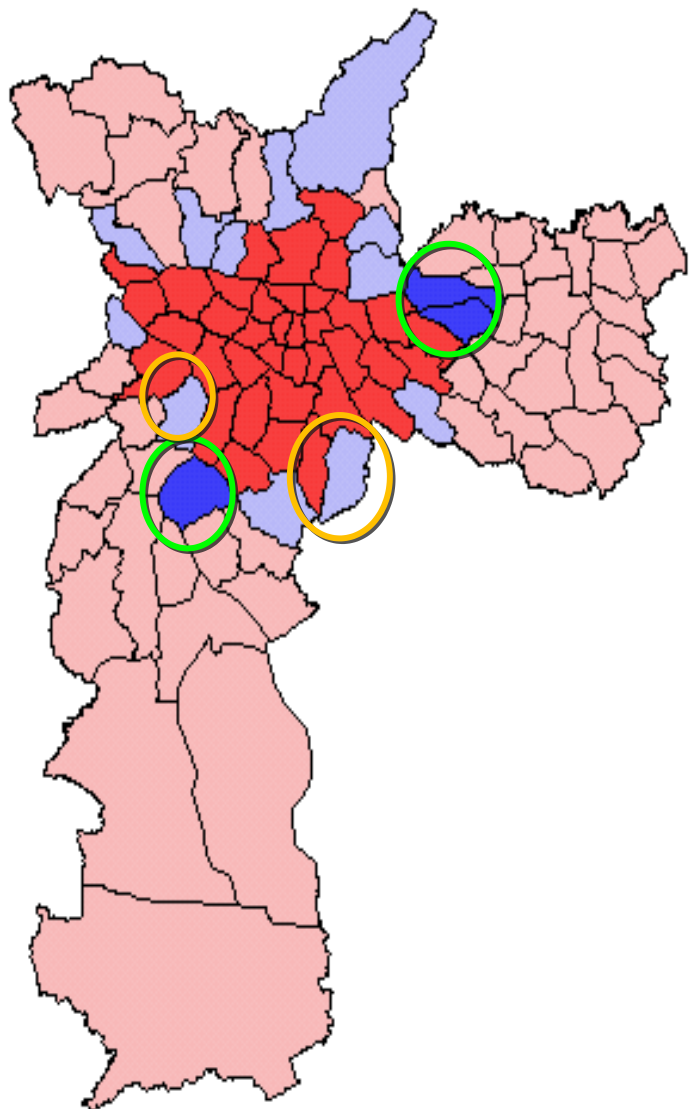
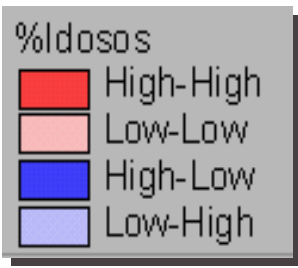
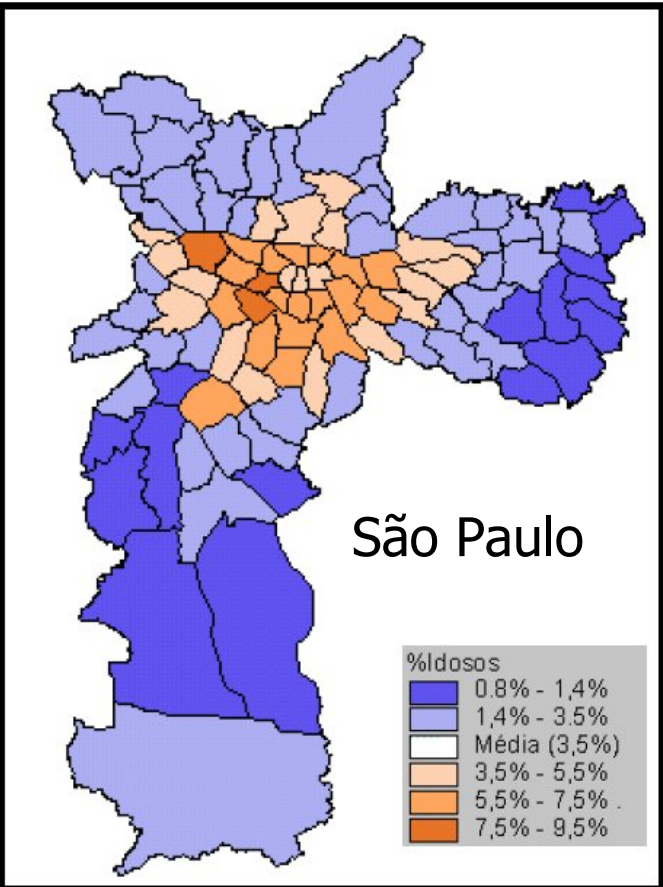


Diagrama de Espalhamento de Moran para o índice de exclusão/inclusão social de São Paulo, censo de 1991.

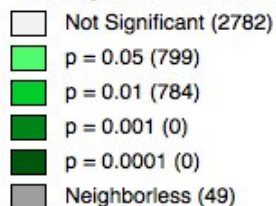
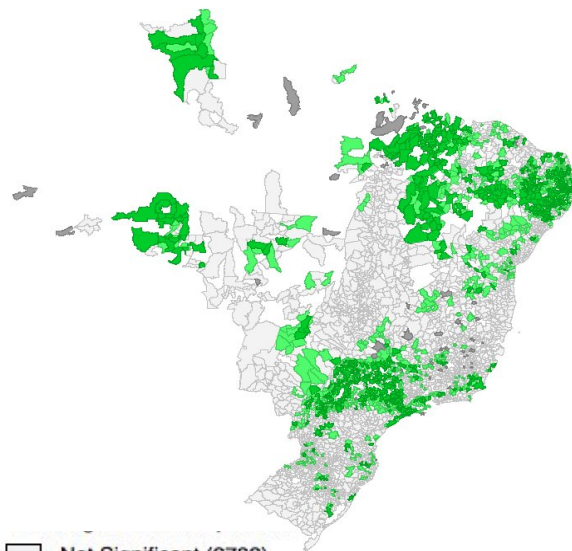


Atributo considerado  
percentagem de idosos



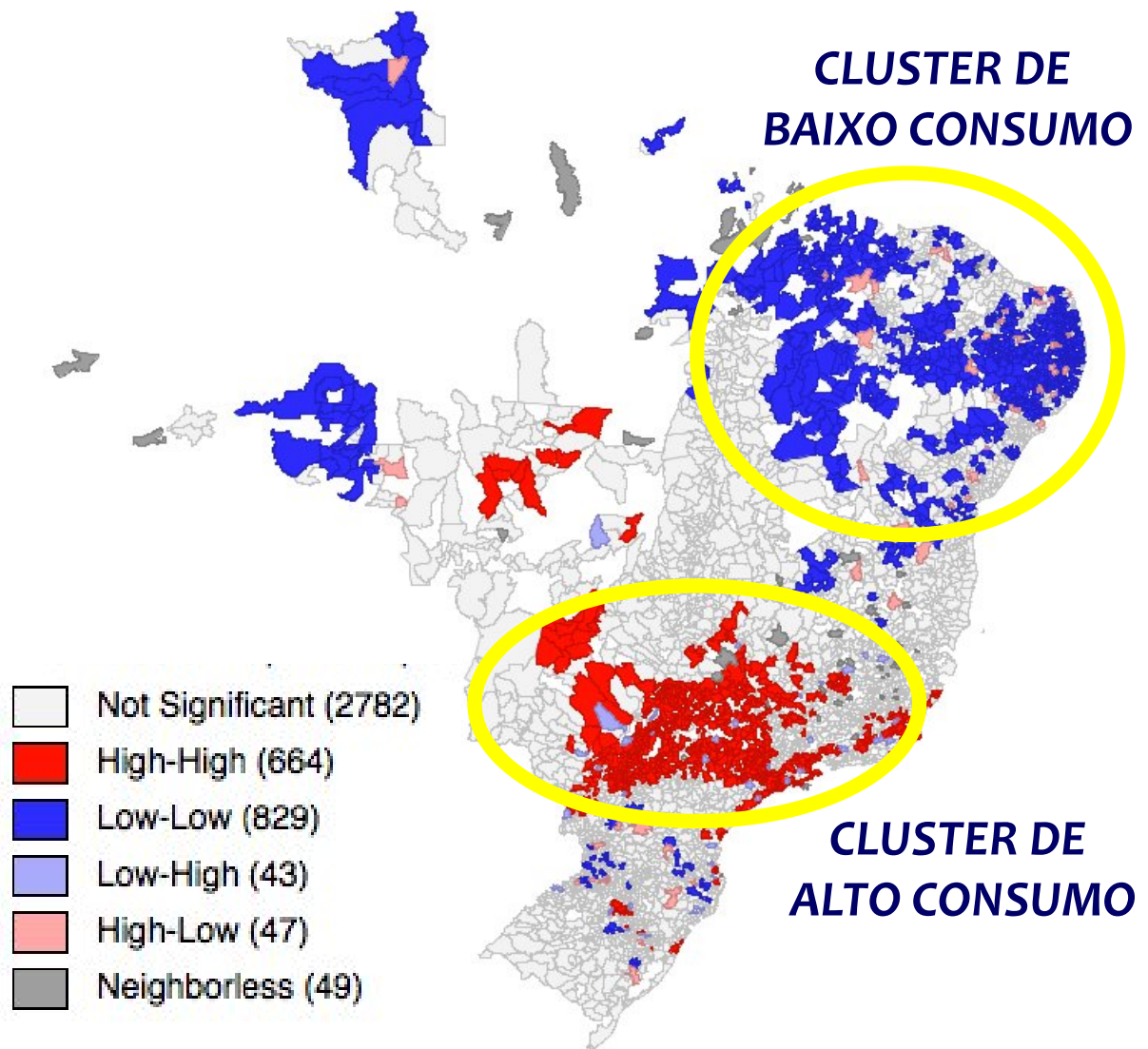
Mapear apenas  
objetos  
com  $p < 0.05$  no  
Índice Local  
de Moran

Significância



# LISA Map

Consumo per capita de água



# Spatial Lag – Média dos valores dos vizinhos

```
setores_juntos$lag_renda <- lag.listw(vizinhanca_pesos, var=setores_juntos$Renda)
```

```
View(setores_juntos@data)
```

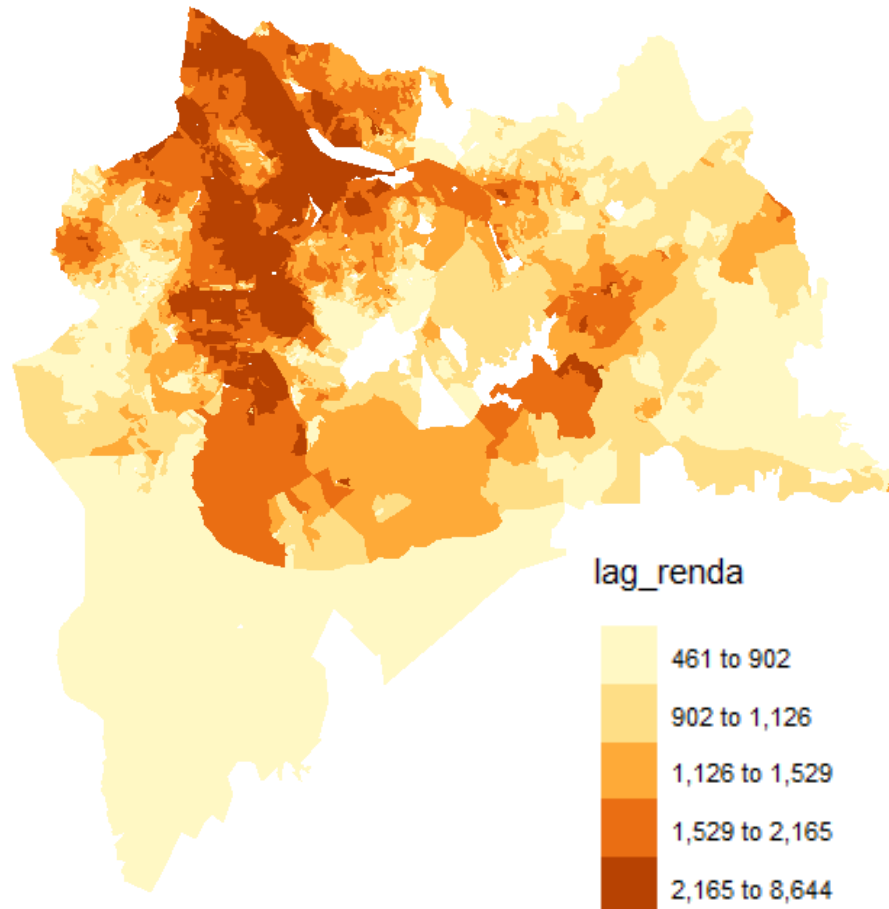
```
tm_shape(setores_juntos) + tm_fill("lag_renda", style="quantile")
```

lag\_renda

1147.1200

965.9167

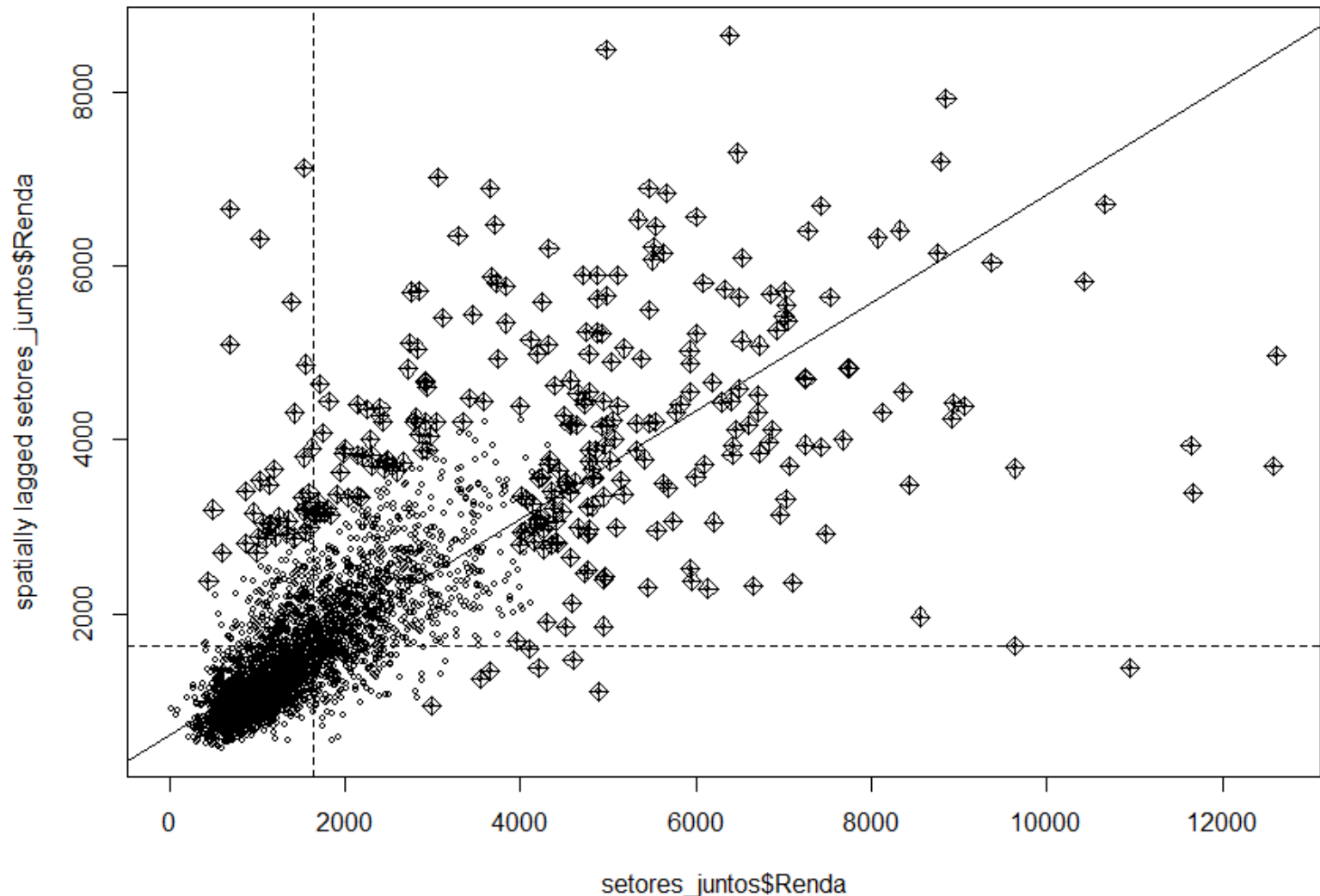
739.7700





# Diagrama de espalhamento de Moran

```
moran.plot(x = setores_juntos$Renda, listw = vizinhanca_pesos,  
cex=0.6, labels=FALSE)
```



# LISA Maps

```
L1 <- factor(setores_juntos$Renda < mean(setores_juntos$Renda),  
labels=c("H", "L"))
```

```
L2 <- factor(setores_juntos$lag_renda < mean(setores_juntos$lag_renda),  
labels=c("H", "L"))
```

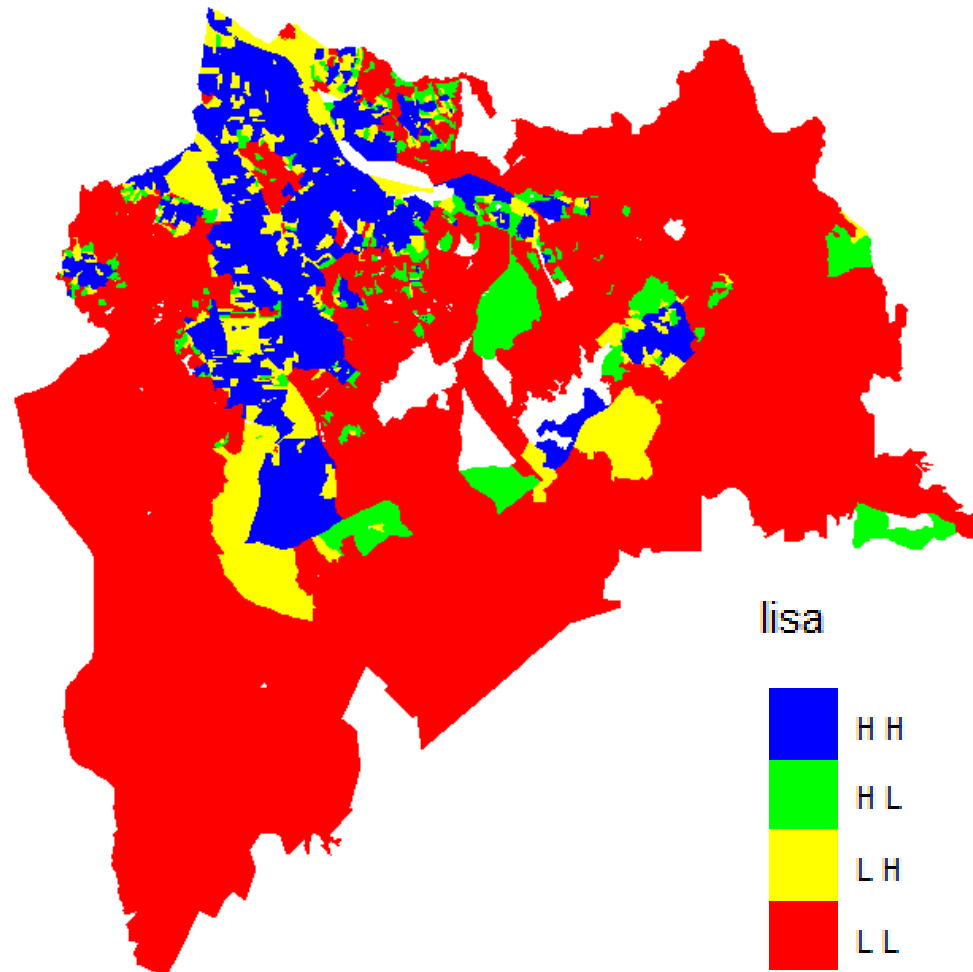
```
setores_juntos$lisa <- paste(L1, L2)
```

```
View(setores_juntos@data)
```

<b><i>lisa</i></b>
H L
L L
H H
L L

# LISA Maps

```
tm_shape(setores_juntos) +  
  tm_fill("lisa", palette=c("blue","green","yellow","red"))
```



# LISA Maps

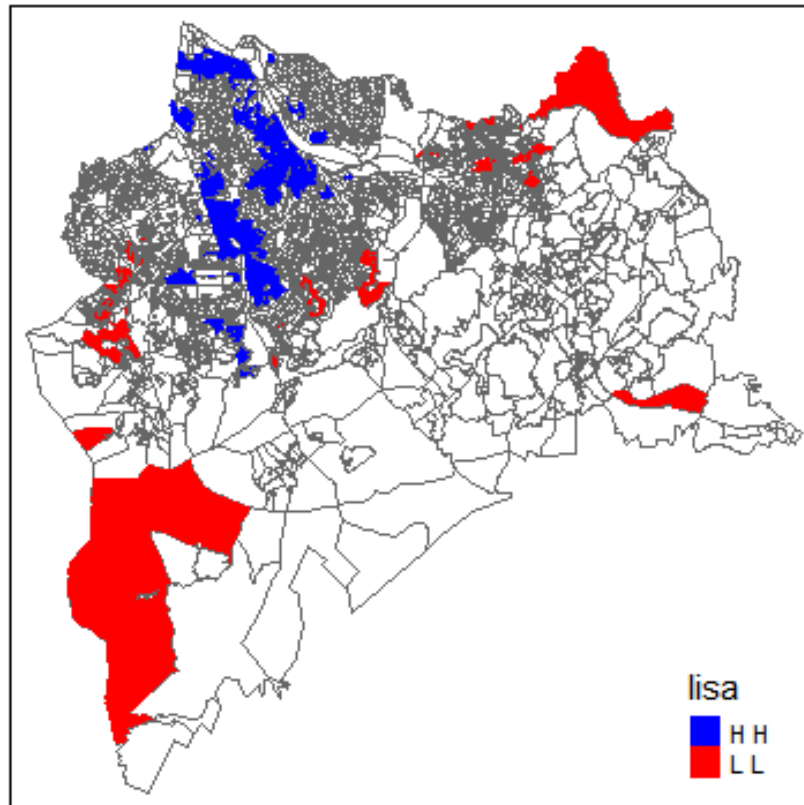
```
lisa_map <- setores_juntos[setores_juntos$moran_p <= 0.05,]
```

```
table(lisa_map$lisa)
```

H H	L L
414	70

```
tm_shape(setores_sp) + tm_borders() +
```

```
tm_shape(lisa_map) + tm_fill("lisa", palette = c("blue", "red"))
```



# Exercício 4

A- Faça um mapa do Índice Local de Moran para expectativa de vida nos municípios da RMSP, comparando com o mapa de significância estatística.

B - Elabore um mapa de média dos vizinhos, um diagrama de espalhamento de Moran, e um LISA map dos municípios da RMSP

# Conteúdo

- Análise de vizinhança
- Autocorrelação espacial
- **Suavização espacial**
- Regressão espacial
- Agrupamento

# Suavização por estimadores bayesianos empíricos

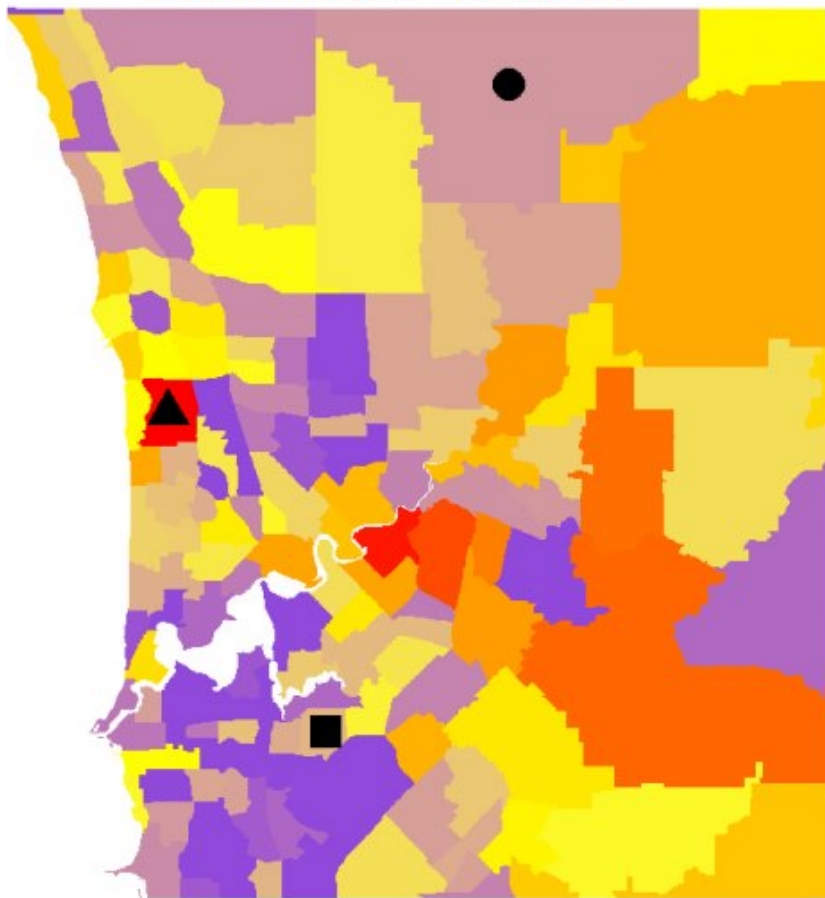
- Casos de risco em locais com baixa população
  - Geram altas taxas de risco
  - Podem ser gerados por acaso
- Pode-se “redistribuir” o risco dos locais com baixa população para as demais áreas
  - Altera de risco observado para “previsão do risco”
- Método global
  - Redistribui risco para todas as demais regiões
- Método local
  - Redistribui riscos para os vizinhos
  - Usa estrutura de autocorrelação espacial

Anselin, Luc, Nancy Lozano-Gracia, and Julia Koschinky. 2006. “Rate Transformations and Smoothing.” Technical Report. Urbana, IL: Spatial Analysis Laboratory, Department of Geography, University of Illinois.

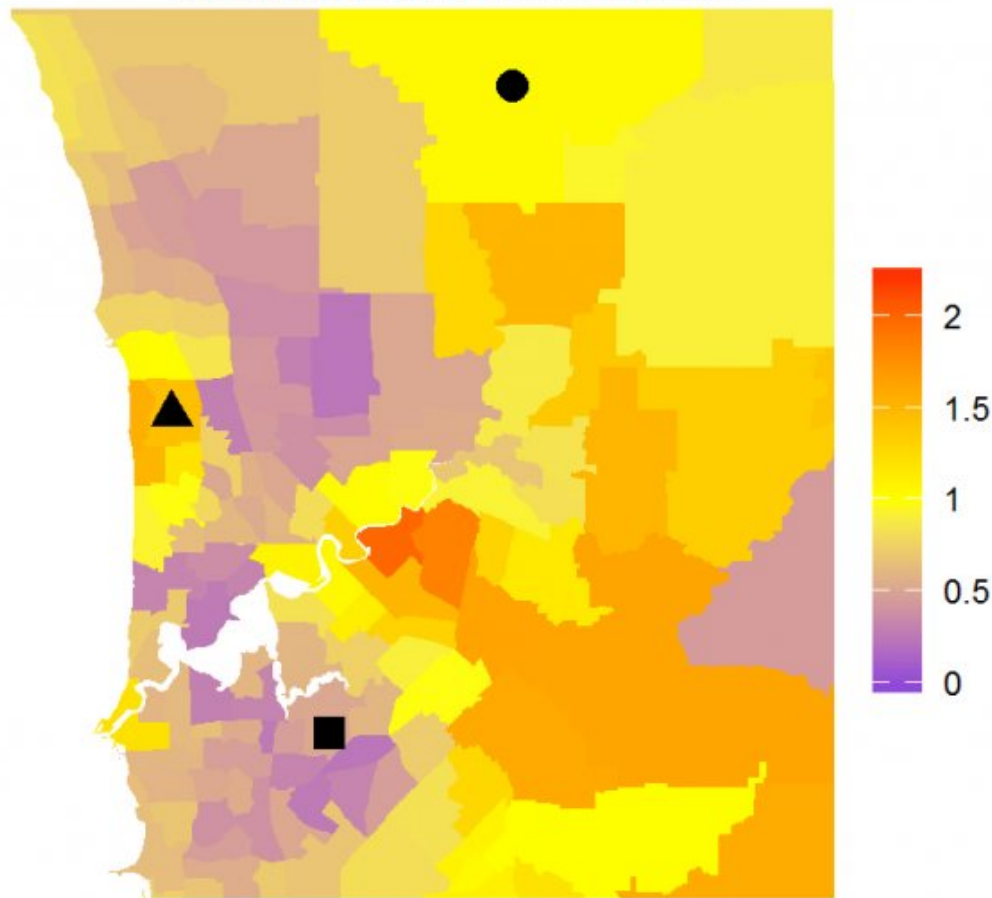
Clayton, David, and John Kaldor. 1987. “Empirical Bayes Estimates of Age-Standardized Relative Risks for Use in Disease Mapping.” *Biometrics* 43:671–81.

# Suavização por estimadores bayesianos empíricos locais

Valores observados



Valores suavizados





# Suavização por estimadores bayesianos empíricos

**Casos de risco**

**População em risco**

```
bayes_global <- EBest(n=setores_juntos$deslizam, x=setores_juntos$Pessoas)
```

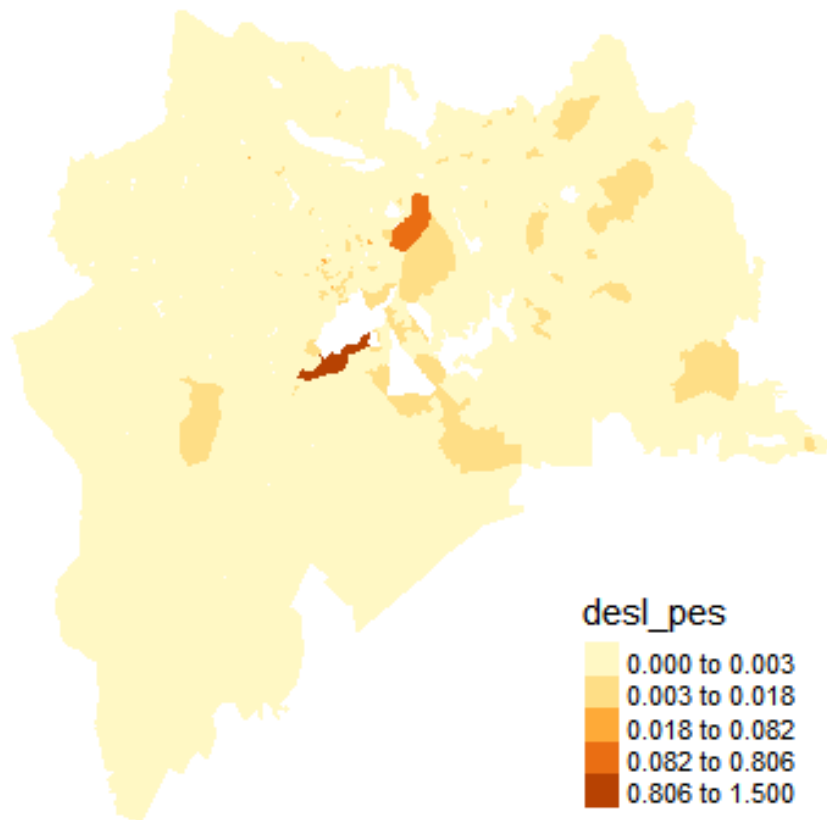
```
View(bayes_global)
```

```
setores_juntos$desl_pes <- bayes_global$raw
```

```
tm_shape(setores_juntos) + tm_fill("desl_pes", style="fisher")
```

	raw	estmm
1340	1.500000000	0.032893193
895	0.111111111	0.010334017
1930	0.052631579	0.009384743

**Classificação por  
quebras naturais**

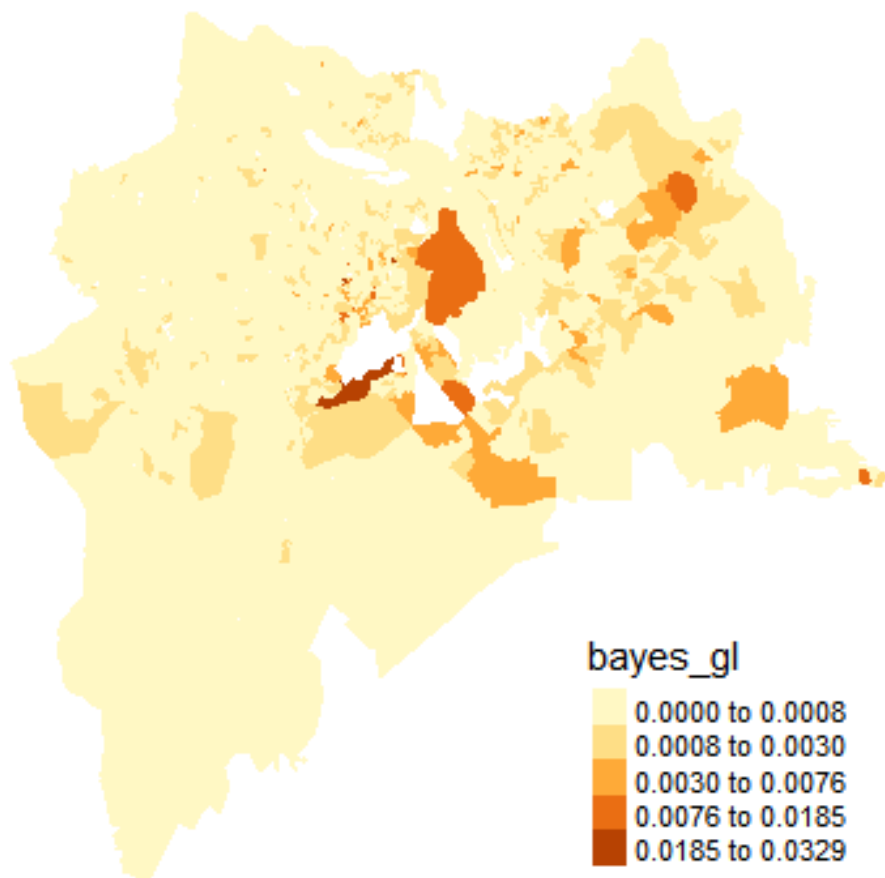


# Suavização por estimadores bayesianos empíricos

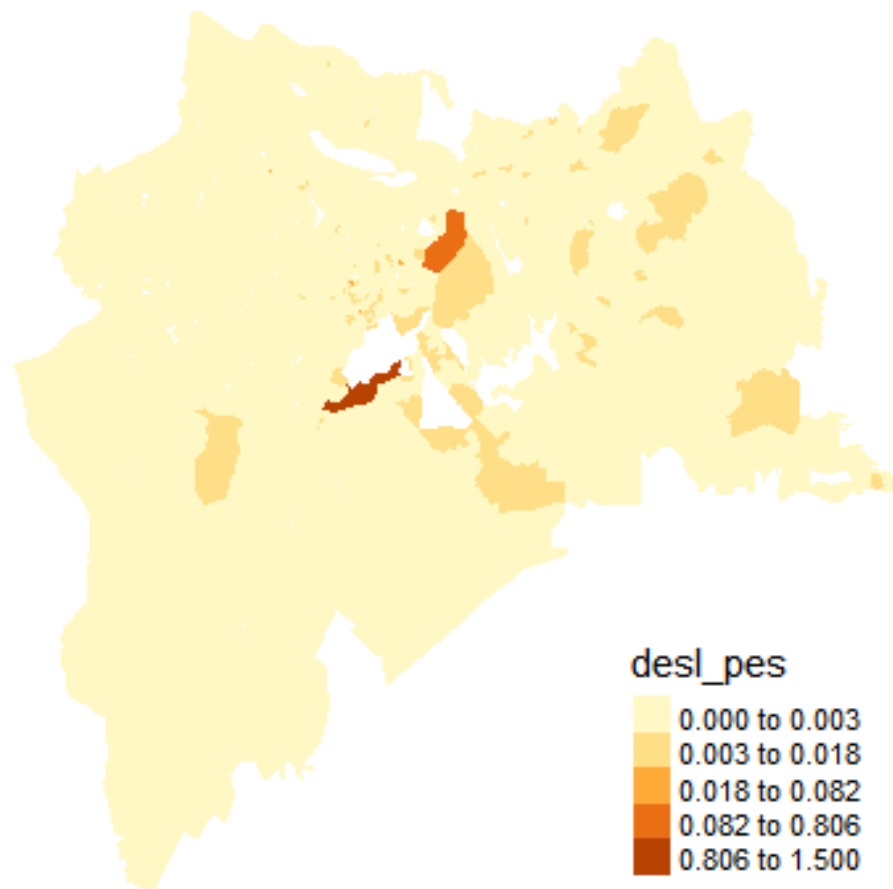
```
setores_juntos$bayes_gl <- bayes_global$estmm
```

```
tm_shape(setores_juntos) + tm_fill("bayes_gl", style="fisher")
```

**Estimador global bayesiano**



**Risco original**



# Suavização por estimadores bayesianos empíricos

```
bayes_local <- EBlocal(ri=setores_juntos$deslizam,  
ni=setores_juntos$Pessoas, nb=vizinhanca2)
```

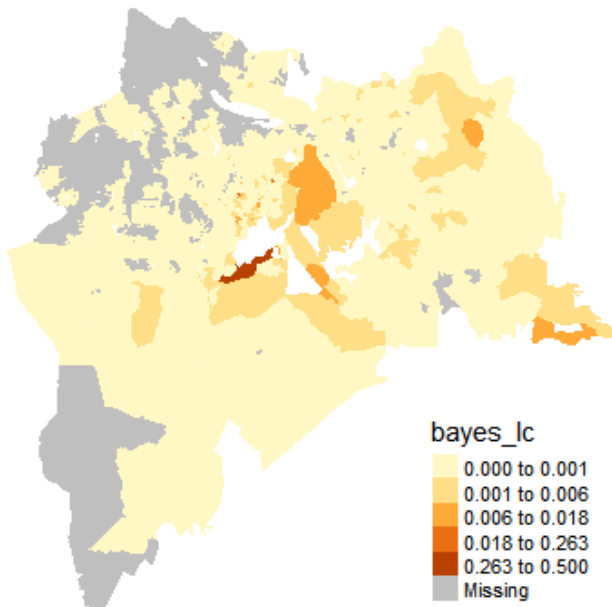
```
View(bayes_local)
```

```
setores_juntos$bayes_lc <- bayes_local$est
```

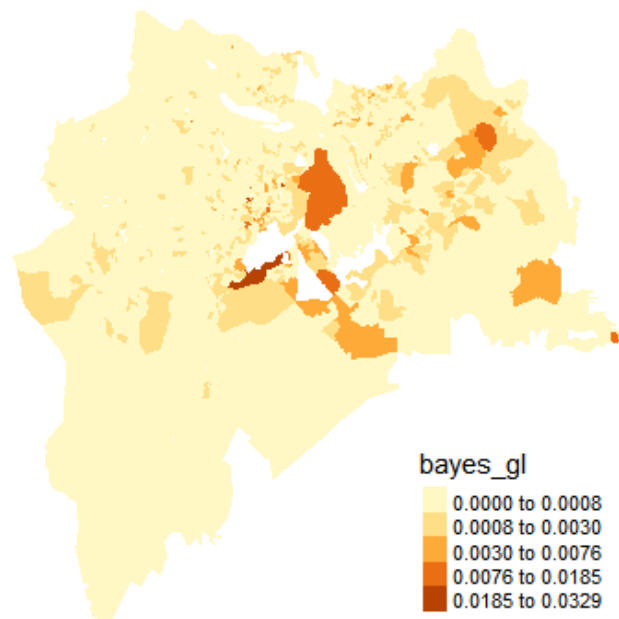
```
tm_shape(setores_juntos) + tm_fill("bayes_lc", style="fisher")
```

	raw	est
1340	1.50000000000	0.500492458
1678	0.0281923715	0.026422418
1930	0.0526315789	0.026073260

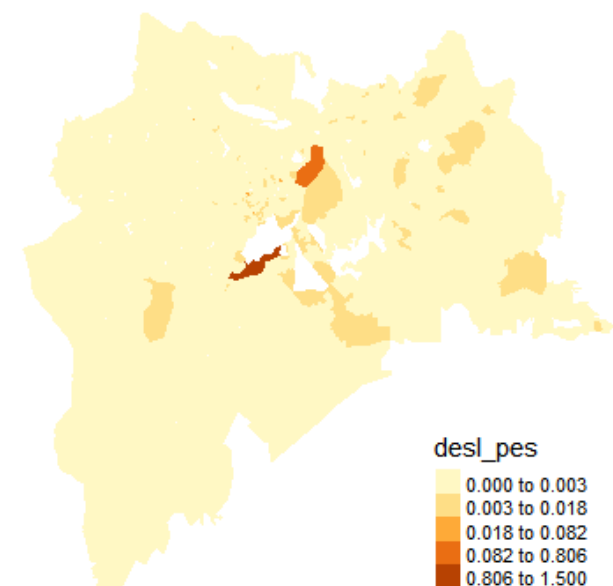
**Estimador local  
bayesiano**



**Estimador global  
bayesiano**



**Risco original**



# Exercício 5

Fazer os mapas de risco observado, suavização bayesiana geral e local para as ocorrências de inundação no ABC

# Conteúdo

- Análise de vizinhança
- Autocorrelação espacial
- Suavização espacial
- **Regressão espacial**
- Agrupamento

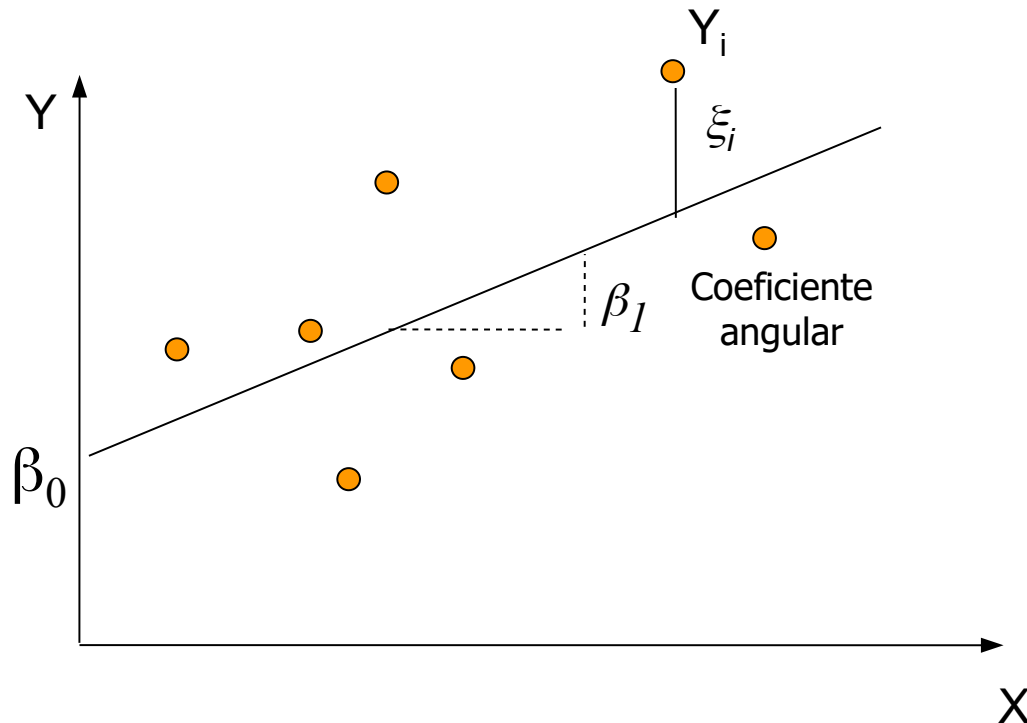
# Regressão Linear Simples

Diagram illustrating the components of the Simple Linear Regression equation:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Labels and arrows pointing to the equation components:

- Variável Resposta** (Response Variable) points to  $Y_i$ .
- Intercepto Populacional** (Population Intercept) points to  $\beta_0$ .
- Inclinação Populacional** (Population Slope) points to  $\beta_1$ .
- Variável Preditora** (Predictor Variable) points to  $X_i$ .
- Erro Aleatório** (Random Error) points to  $\varepsilon_i$ .



# Regressão Linear Simples

$$Y_i = \beta_0 + \beta_1 X_i + \xi_i$$

**onde:**

$Y_i$  é o valor da variável resposta na  $i$ -ésima observação

$\beta_0$  e  $\beta_1$  são parâmetros;

$X_i$  é o valor da variável preditora na  $i$ -ésima observação

$\xi_i$  é o termo de erro aleatório

**Lembrando:**

$$\text{Saída}_i = (\text{Modelo}_i) + \text{erro}_i$$

<u>x</u>	<u>y</u>
3	10
7	12
8	19
12	18
14	24
17	25
18	30
23	40
25	40
30	45

# Regressão Linear Múltipla

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + \varepsilon_i$$

$Y_i$  é o valor da variável resposta na  $i$ -ésima observação

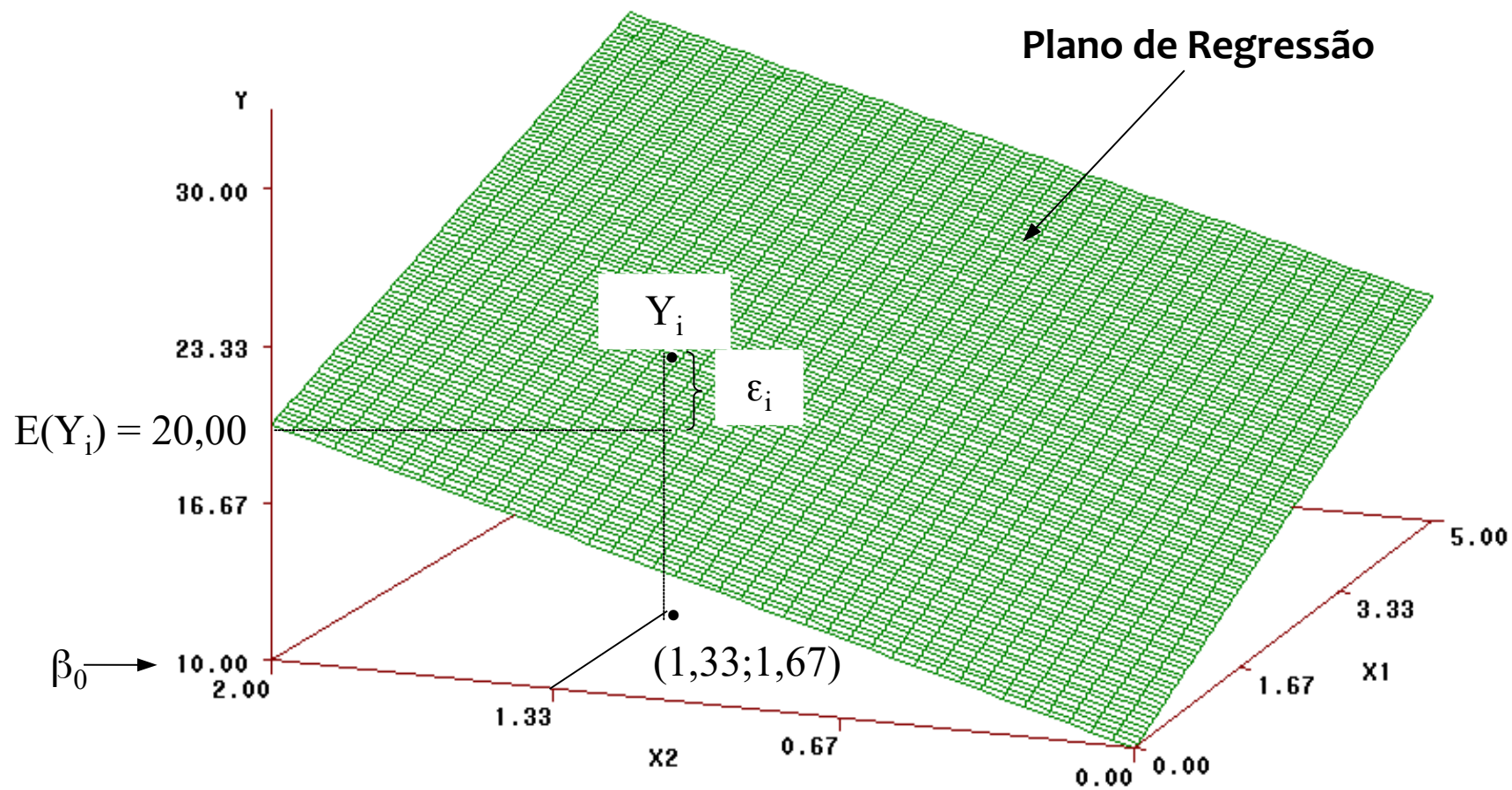
$\beta_0, \dots, \beta_p$  são parâmetros

$X_{i1}, \dots, X_{ip}$  são os valores das variáveis preditoras na  $i$ -ésima observação

$\varepsilon_i$  é o termo de erro aleatório

$X_1$	$X_2$	$Y$
9	3	54
7	1	35
5	4	42
11	8	74
8	9	65
2	1	15





# Coeficiente de Determinação ( $R^2$ )

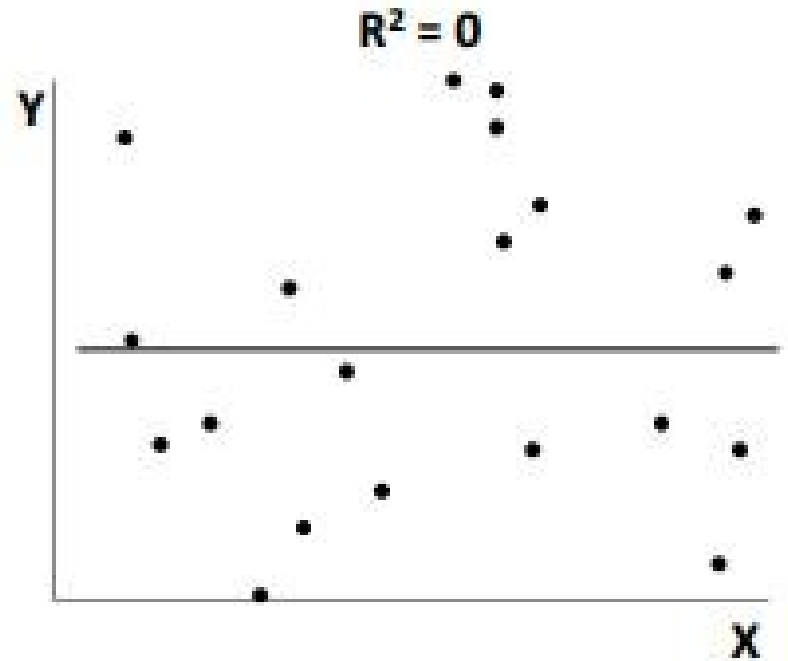
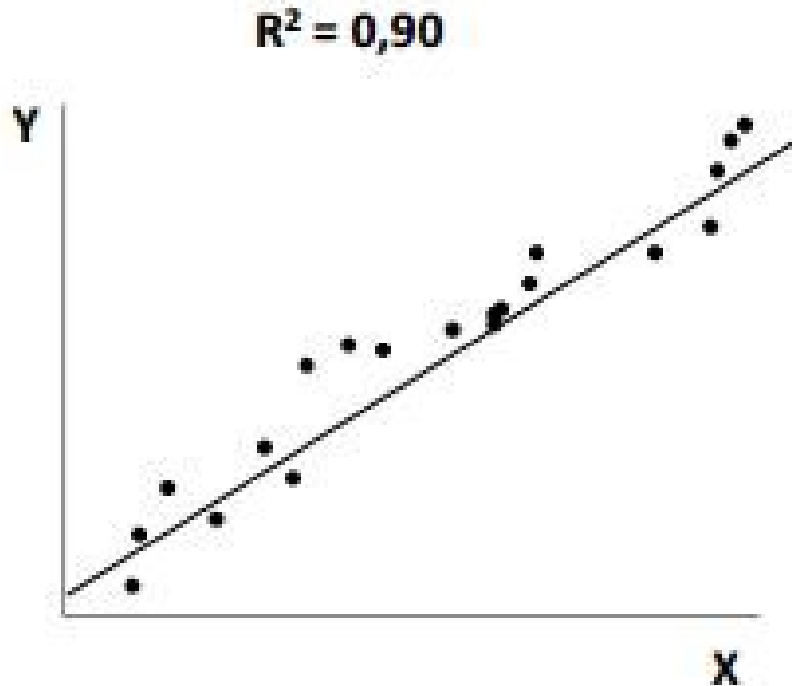
Quantidade de variação em Y explicada pelo modelo

Temos dois casos extremos:

**$R^2 = 1$**  todas as observações caem na linha de regressão ajustada. A variável preditora X explica toda a variação nas observações.

**$R^2 = 0$**  Não existe relação linear em Y e X. A variável X não ajuda a explicar a variação dos  $Y_i$ .

# Coeficiente de Determinação ( $R^2$ )

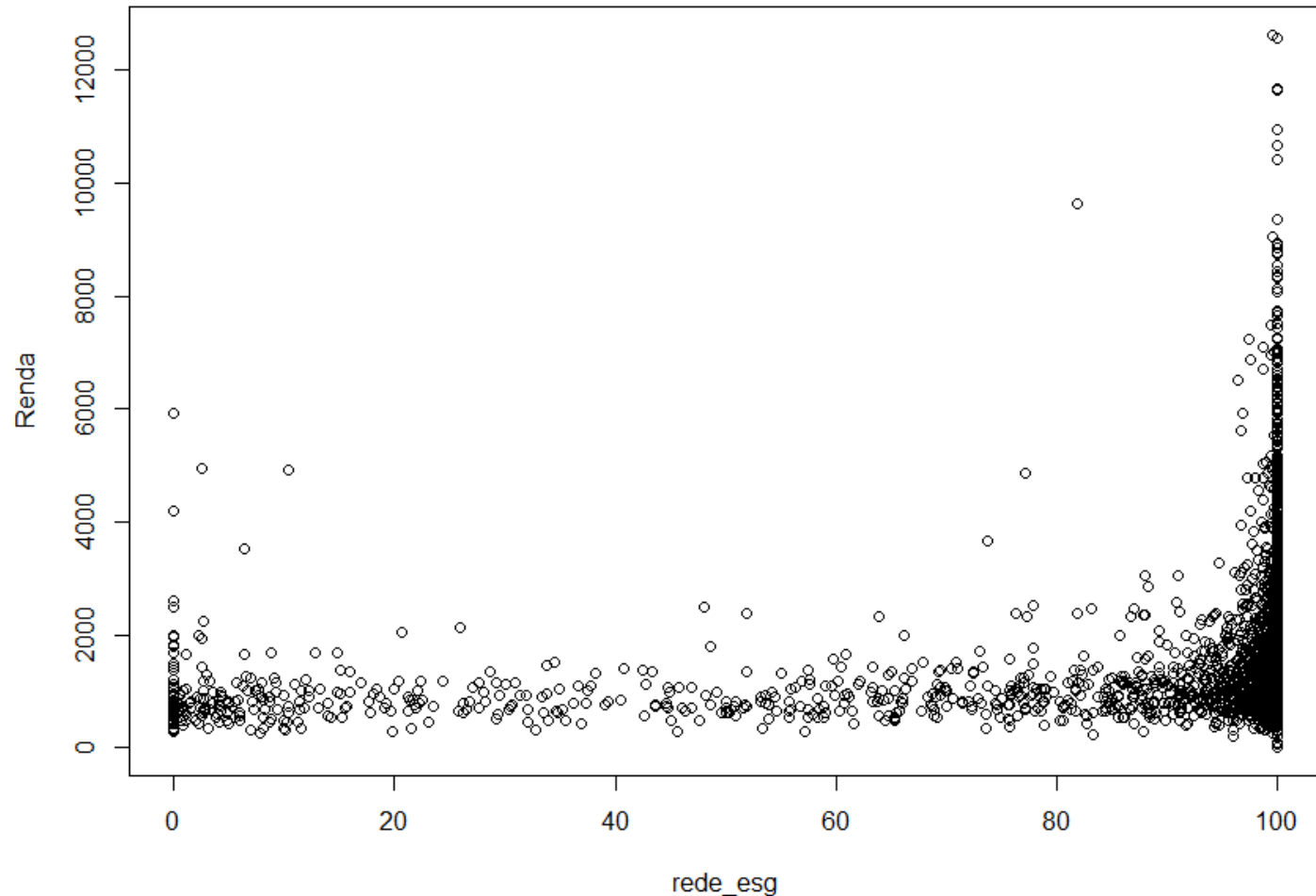


# Critério de Informação de Akaike

- Potencial de explicação do modelo
  - Desconto por “complexidade” (incerteza)
- Quanto menor melhor
- Pode comparar modelos que não possuem  $R^2$

# Regressão Linear

```
plot(data=setores_juntos, Renda ~ rede_esg)
```



# Regressão Linear

```
regressao_convencional <- lm(data=setores_juntos, Renda ~ rede_esg)
summary(regressao_convencional)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	673.2948	73.5371	9.156	<2e-16 ***
rede_esg	10.7171	0.7891	13.582	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1233 on 3993 degrees of freedom

Multiple R-squared: 0.04416, Adjusted R-squared: 0.04392

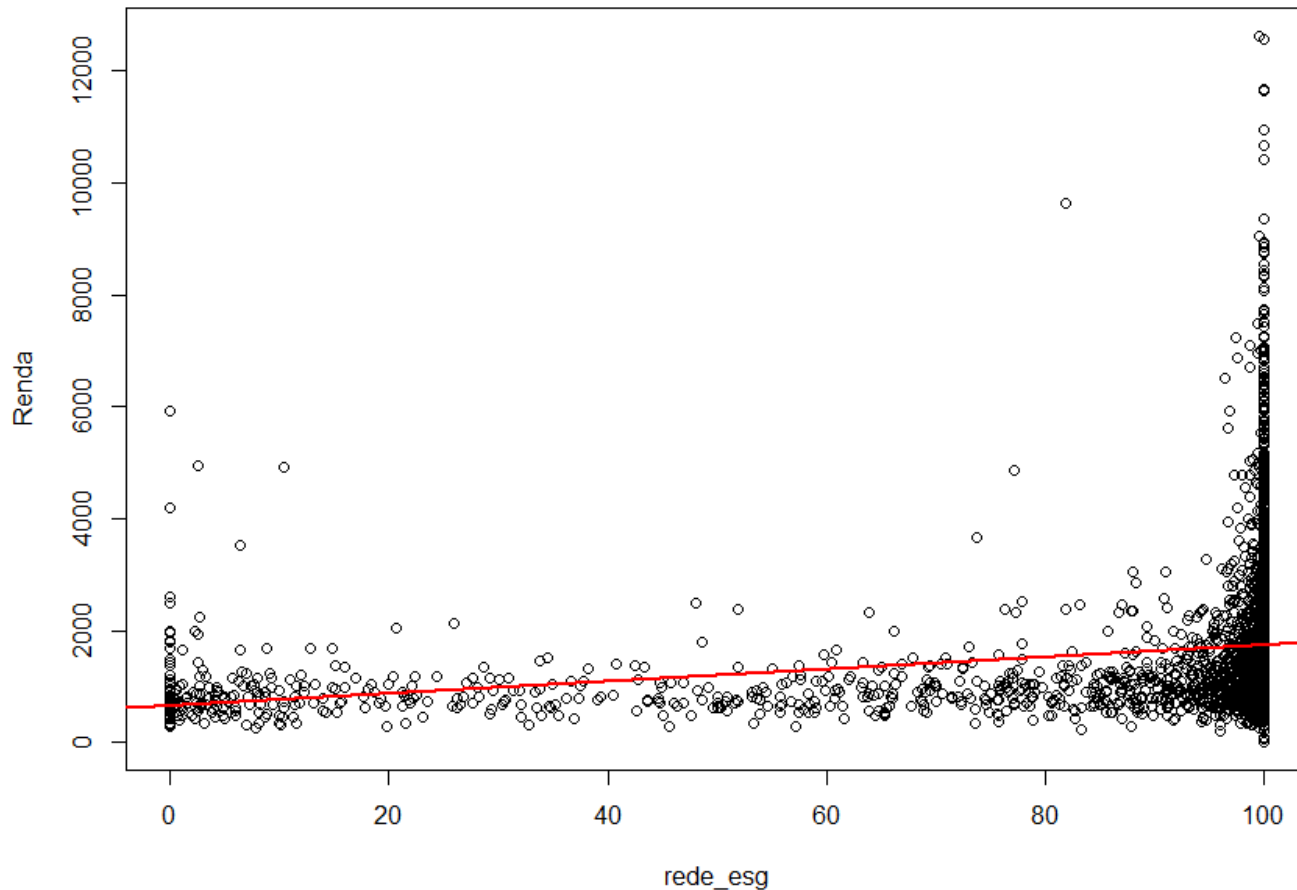
F-statistic: 184.5 on 1 and 3993 DF, p-value: < 2.2e-16

```
AIC(regressao_convencional)
```

```
[1] 68210.32
```

# Regressão Linear

```
plot(data=setores_juntos, Renda ~ rede_esg)  
abline(regressao_convencional, col="red", lwd=2)
```



# Análise dos Resíduos

- Medir a *autocorrelação espacial* dos *resíduos da regressão* (Índice de Moran dos *resíduos*)
- Se houver autocorrelação espacial:
  - Pode haver alguma variável ou padrão espacial que não foi investigado pelo modelo.
  - Possibilidade de aplicar modelos de regressão espacial

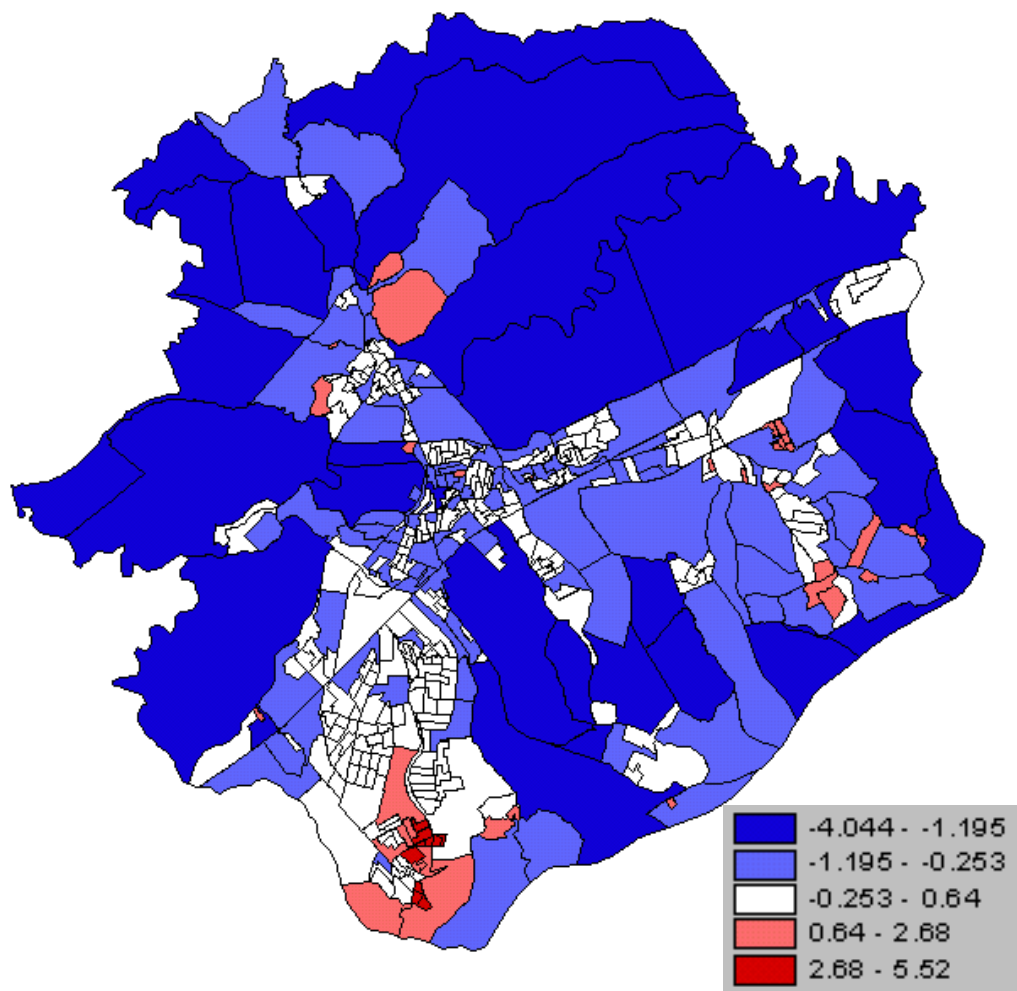


# Exemplo

## São José dos Campos

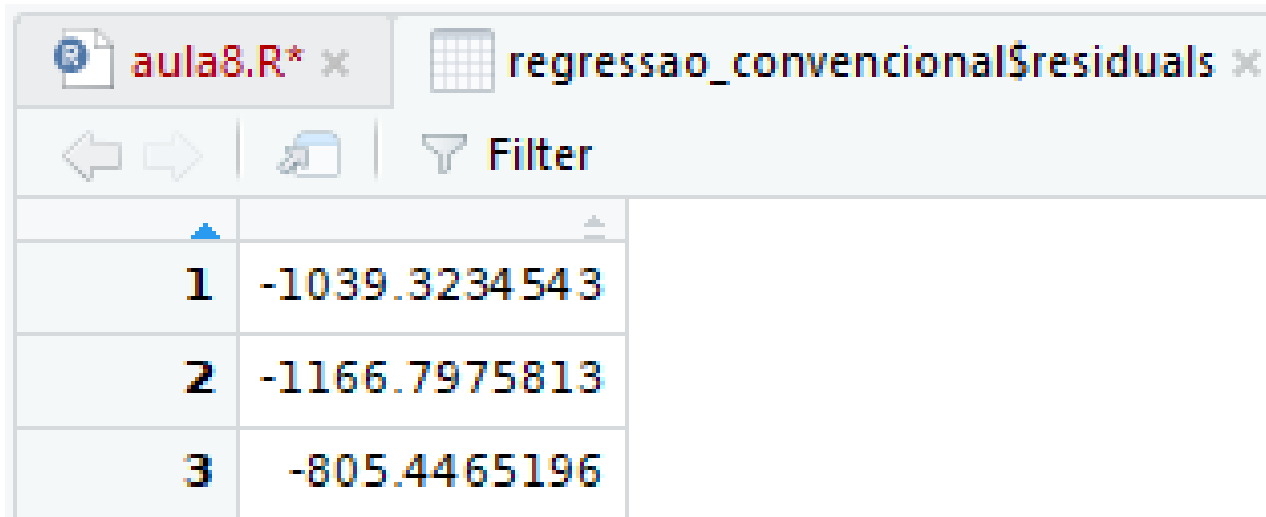
Crescimento Populacional 1991-2000 X Densidade Populacional 1991

- Mapear os resíduos da regressão – índices de correlação
- Índice de Moran sobre **mapa de resíduos  $I=0,45$ , com  $p < 0.05$**



# Análise dos Resíduos

`View(regressao_convencional$residuals)`



The screenshot shows the RStudio interface with two tabs: 'aula8.R\*' and 'regressao\_convencional\$residuals'. The 'regressao\_convencional\$residuals' tab is active, displaying a data frame with three rows of residuals. The interface includes navigation arrows, a filter icon, and a 'Filter' label.

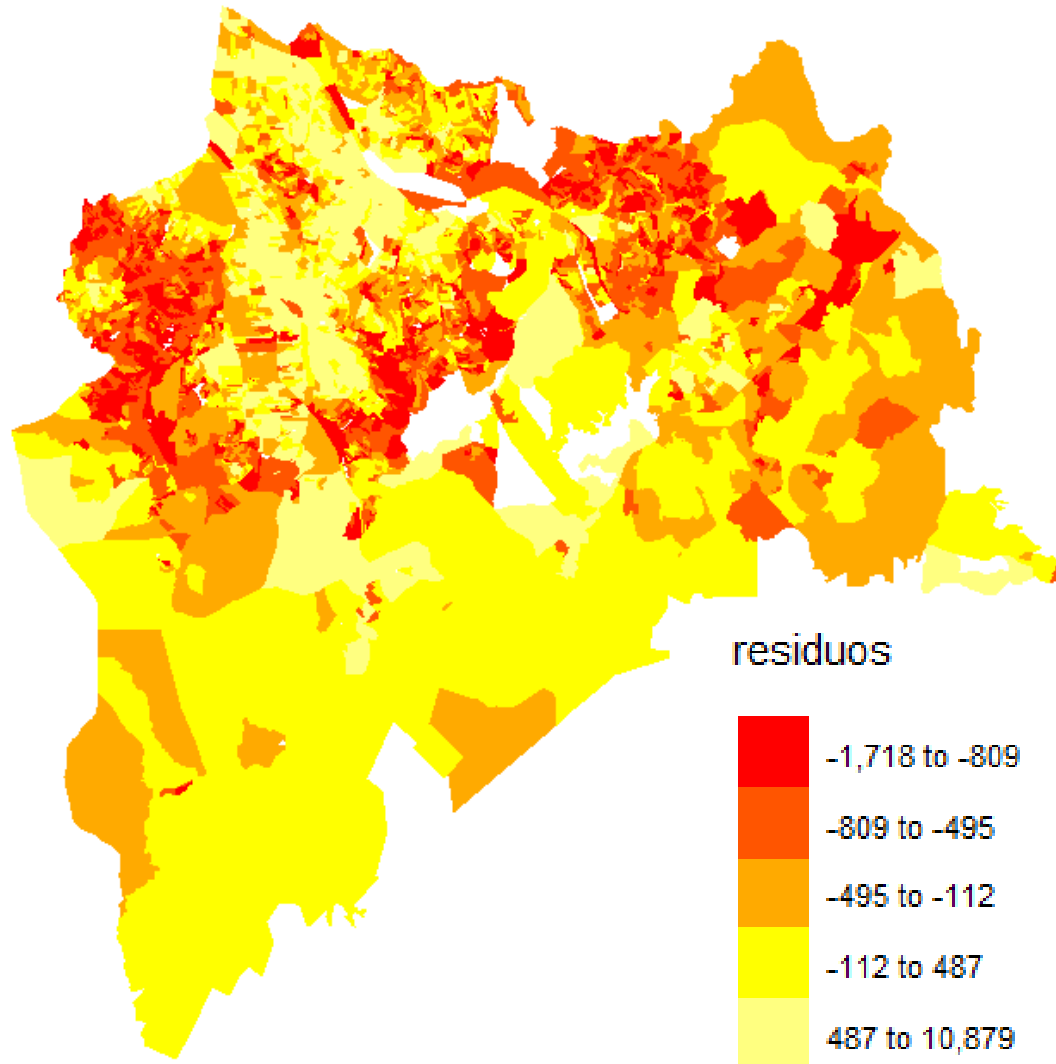
1	-1039.3234543
2	-1166.7975813
3	-805.4465196

```
setores_juntos$residuos <- regressao_convencional$residuals
```

# Análise dos Resíduos

```
tm_shape(setores_juntos) +
```

```
tm_fill("residuos", style = "quantile", palette = heat.colors(5))
```



# Análise dos Resíduos

```
lm.morantest(regressao_convencional, listw = vizinhanca_pesos)
```

Moran I statistic standard deviate = 61.541, p-value < 2.2e-16  
alternative hypothesis: greater  
sample estimates:

Observed Moran I	Expectation	Variance
5.979397e-01	-3.991722e-04	9.452949e-05

**p<0.05, então  
é significativo**

**Moran I > 0,  
autocorrelação  
espacial positiva**

# Regressão Espacial

- **Globais:**

- inclui no modelo de regressão um parâmetro para capturar a estrutura de autocorrelação espacial na área de estudo como um todo

- **Locais:**

- parâmetros variam continuamente no espaço

# Global vs. Local

Global	Local
Estatísticas dizem respeito à região como um todo (1 valor)	Disagregações locais das estatísticas globais (Muitos valores)
Estatísticas globais e não mapeáveis	Estatísticas locais e mapeáveis
Ênfase nas similaridades da região	Ênfase nas diferenças ao longo do espaço
Procura regularidades ou “leis”	Procura por exceções ou “hot-spots” locais
Ex.: Regressão Clássica, Spatial Lag, Spatial Error	Ex.: GWR, Regimes Espaciais

# Modelos com Efeitos Espaciais Globais

## PREMISSA

É possível capturar a estrutura de correlação espacial num único parâmetro  
(adicionado ao modelo de regressão)

## Alternativas

***Spatial Lag Models:*** atribuem a autocorrelação espacial à variável resposta Y

***Spatial Error Models:*** atribuem a autocorrelação ao erro

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

# Modelo *Spatial Lag*

**PREMISSA:** A variável  $Y_i$  é afetada pelos valores da variável resposta nas áreas vizinhas a  $i$ .

$$Y = \rho WY + X\beta + \varepsilon$$

$\rho$  = *coeficiente espacial autoregressivo* - medida de correlação espacial ( $\rho = 0$ , se autocorrelação é nula - hipótese nula)

$W$  = matriz de proximidade espacial

$WY$  expressa a dependência espacial em  $Y$



# Modelo *Spatial Lag* NÃO RODAR, DEMORA MUITO

```
regressao_espacial_lag <- lagsarlm(data=setores_juntos, Renda ~  
rede_esg, listw=vizinhanca_pesos)
```

```
summary(regressao_espacial_lag, Nagelkerke = TRUE)
```

**Pseudo-R<sup>2</sup> de Nagelkerke**

Type: lag

Coefficients: (numerical Hessian approximate standard errors)

	Estimate	Std. Error	z value	Pr(> z )
--	----------	------------	---------	----------

(Intercept)	14.22490	40.60841	0.3503	0.7261
-------------	----------	----------	--------	--------

rede_esg	3.75346	0.45946	8.1693	2.22e-16
----------	---------	---------	--------	----------

**p<0.05, então  
é significativo**

Rho 0.79148 LR test value: 2797.3, p-value: < 2.22e-16

Approximate (numerical Hessian) standard error: 0.01049

z-value: 75.453, p-value: < 2.22e-16

Wald statistic: 5693.1, p-value: < 2.22e-16

Log likelihood: -32703.52 for lag model

ML residual variance (sigma squared): 651670, (sigma: 807.26)

Nagelkerke pseudo-R-squared: 0.52544

**0,04 na regressão  
convencional**

Number of observations: 3995

Number of parameters estimated: 4

AIC: 65415, (AIC for lm: 68210)

**Compara com a  
regressão convencional**

**10,7 na regressão  
convencional  
Coeficiente de  
vizinhança**

**Critério de  
Informação de  
Akaike**

# Modelo *Spatial Error*

## **PREMISSA:**

As observações são interdependentes graças a variáveis não mensuradas, e que são espacialmente correlacionadas

Ou seja: **efeitos espaciais são um ruído**

Assume que, se pudéssemos adicionar as variáveis certas para remover o erro do modelo, o espaço não importaria mais.

# Modelo *Spatial Error*

## MODELO:

$$Y = X\beta + \varepsilon$$

$$\varepsilon = \rho W\varepsilon + \xi$$

$W\varepsilon$  = erro com efeitos espaciais

$\rho$  = medida de correlação espacial

$\xi$  = componente do erro com variância constante e não correlacionada.

# Modelo *Spatial Error* NÃO RODAR, DEMORA MUITO

```
regressao_espacial_CAR <- spautolm(data=setores_juntos,  
Renda ~ rede_esg, listw=vizinhanca_pesos, family="CAR")
```

Tipo de modelo  
**Spatial Error**

```
summary(regressao_espacial_CAR, Nagelkerke = TRUE)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	3801.01120	185.29551	20.5132	< 2.2e-16
rede_esg	2.52377	0.77833	3.2426	0.001185

Lambda: 0.99459 LR test value: 3016.4 p-value: < 2.22e-16  
Numerical Hessian standard error of lambda: NaN

Log likelihood: -32593.97

ML residual variance (sigma squared): 605840, (sigma: 778.36)

Number of observations: 3995

Number of parameters estimated: 4

AIC: 65196 **65415 no modelo Spatial Lag**

Nagelkerke pseudo-R-squared 0.55076 **0,52 no Spatial Lag**

**3,75 no modelo  
Spatial Lag**

**Coeficiente de  
vizinhança dos  
resíduos**

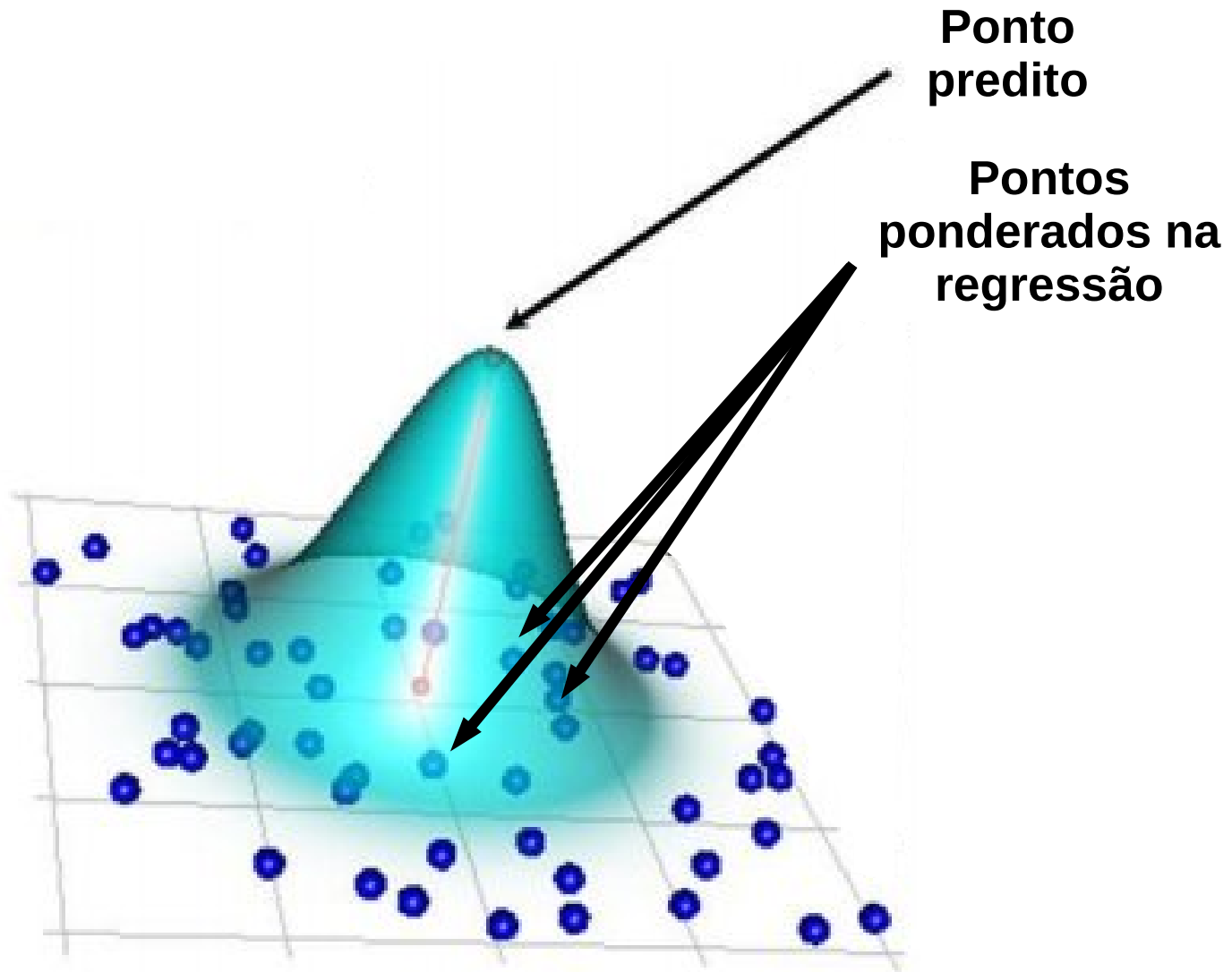
**Critério de  
Informação de  
Akaike**

# Exercício 6

Modele a expectativa de vida em função de porcentagem de analfabetos na RMSP

Compare os resultados de regressão convencional (e autocorrelação espacial dos resíduos), regressão spatial lag e error.

# ***GWR – Geographically Weighted Regression***



# ***GWR – Geographically Weighted Regression***

Ajusta um modelo de regressão a cada ponto observado, ponderando todas as demais observações como função da distância a este ponto

$$y(i) = b_0(i) + b_1(i) x_1 + e(i)$$

$b_0(i)$ ,  $b_1(i)$  → para cada ponto  $i$  do espaço há um  $b_0$  e  $b_1$  diferentes

Função de *kernel* sobre cada ponto do espaço para ponderar os pontos vizinhos em razão da distância. Pontos mais próximos do ponto central tem maior peso.

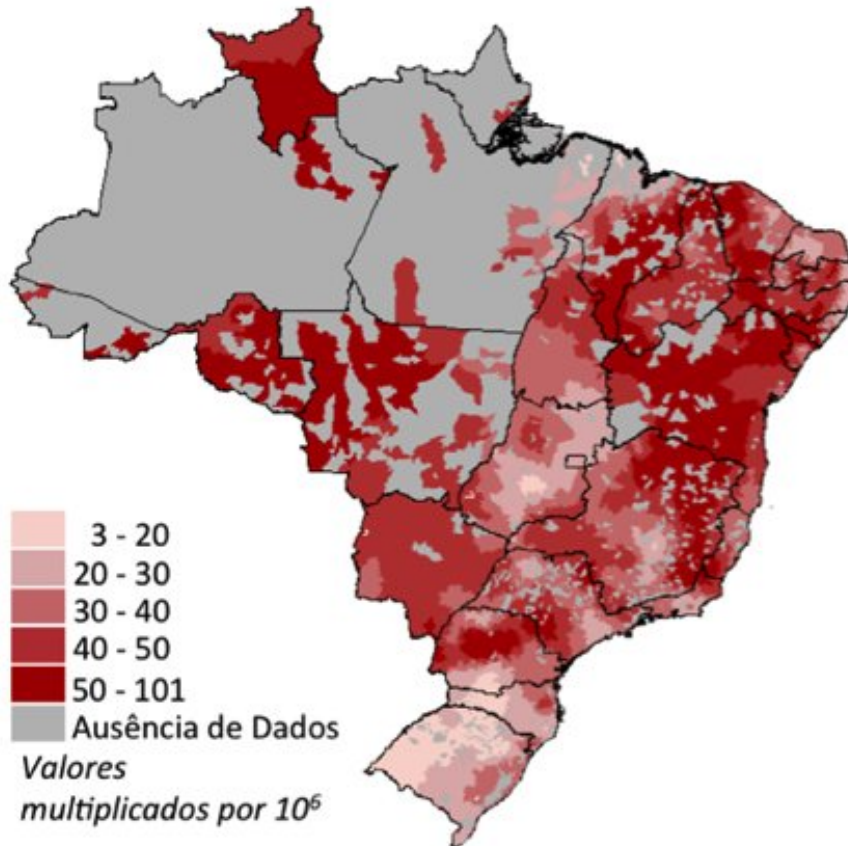
Assim como no *kernel* – a escolha da largura da banda é importante  
(pode ser fixa ou adaptável à densidade dos dados)

# ***GWR – Geographically Weighted Regression***

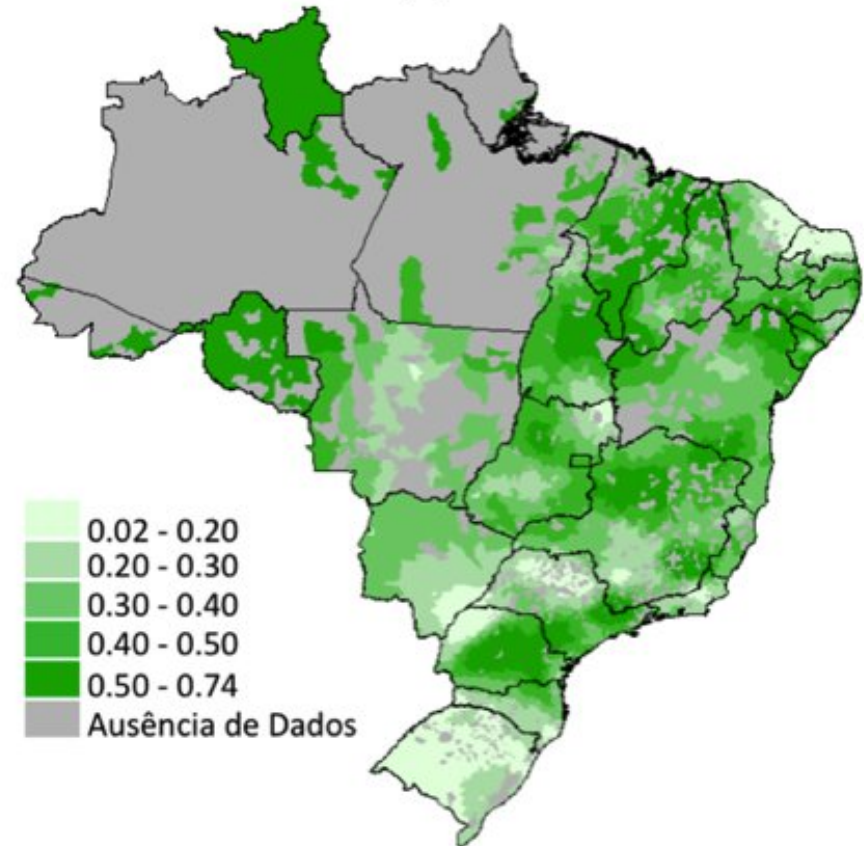
**Consumo de Água per Capita (resposta) X Renda per capita(preditora)**

**GWR:**  $CONSUMO_i = \beta_0(u_i, v_i) + \beta_1(u_i, v_i) \cdot RENDA_i + \varepsilon_i$

(a) PARÂMETROS LOCAIS ESTIMADOS PARA A VARIÁVEL "RENDAS PER CAPITA"



(b) COEFICIENTE LOCAL DE DETERMINAÇÃO (r<sup>2</sup>)





# ***GWR – Geographically Weighted Regression***

- Com raio fixo ***NÃO RODAR! DEMORA MUITO!***

```
raio <- gwr.sel(data=setores_juntos, Renda ~ rede_esg)  
raio
```

```
raio <- 782.3348 Atribuir o resultado ótimo para ganharmos tempo
```

```
setores_gwr <- gwr(data=setores_juntos, Renda ~ rede_esg, bandwidth=raio)
```

- Com raio adaptativo

***NÃO RODAR! DEMORA MUITO!***

**Opção de raio  
adaptativo**

```
raio_adaptativo <- gwr.sel(data=setores_juntos, Renda ~ rede_esg, adapt = TRUE)
```

```
raio_adaptativo
```

```
raio_adaptativo <- 0.007275952 Atribuir o resultado ótimo para ganharmos tempo
```

```
setores_gwr_adaptativo <- gwr(data=setores_juntos, Renda ~ rede_esg,  
adapt=raio_adaptativo)
```

**Opção de raio  
adaptativo**

# ***GWR – Geographically Weighted Regression***

- Visualizando os resultados

View(setores\_gwr\$SDF@data)

			Coeficiente de predição	Erro de predição	Valor predito	R <sup>2</sup> local
	sum.w	X.Intercept.	rede_esg	gwr.e	pred	localR2
1	53.30495	2198.51142	-10.3182768	-507.231274	1189.3613	0.37003434
2	69.44592	863.16486	1.4142608	-426.380941	1004.5909	0.23669567
3	68.47954	945.92153	0.7030401	-81.907450	1015.8574	0.24878558

- Comparando o raio fixo e o adaptivo

- Raiz do erro médio quadrático (Root Mean Square Error – RMSE) =  $\sqrt{\frac{\sum \varepsilon^2}{n}}$

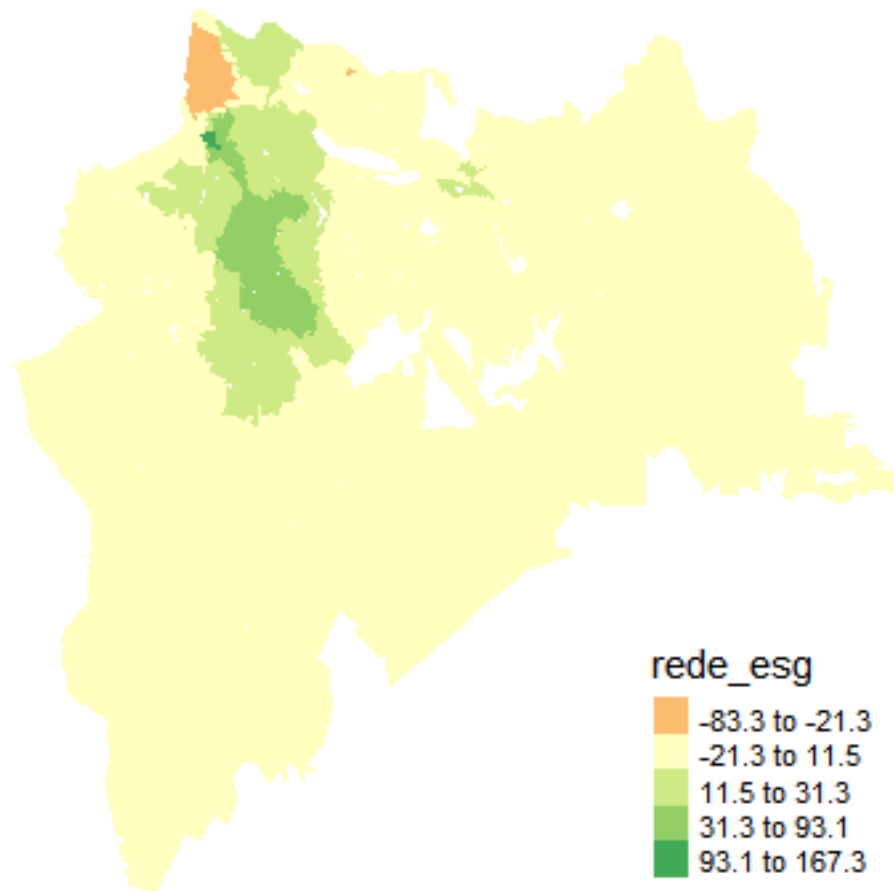
`sqrt(mean(setores_gwr$SDF$gwr.e^2))` [1] 875.9002

`sqrt(mean(setores_gwr_adaptativo$SDF$gwr.e^2))` [1] 867.847

# ***GWR – Geographically Weighted Regression***

```
tm_shape(setores_gwr_adaptativo$SDF) +  
tm_fill("rede_esg", style="fisher")
```

**Classificação por  
quebras naturais**



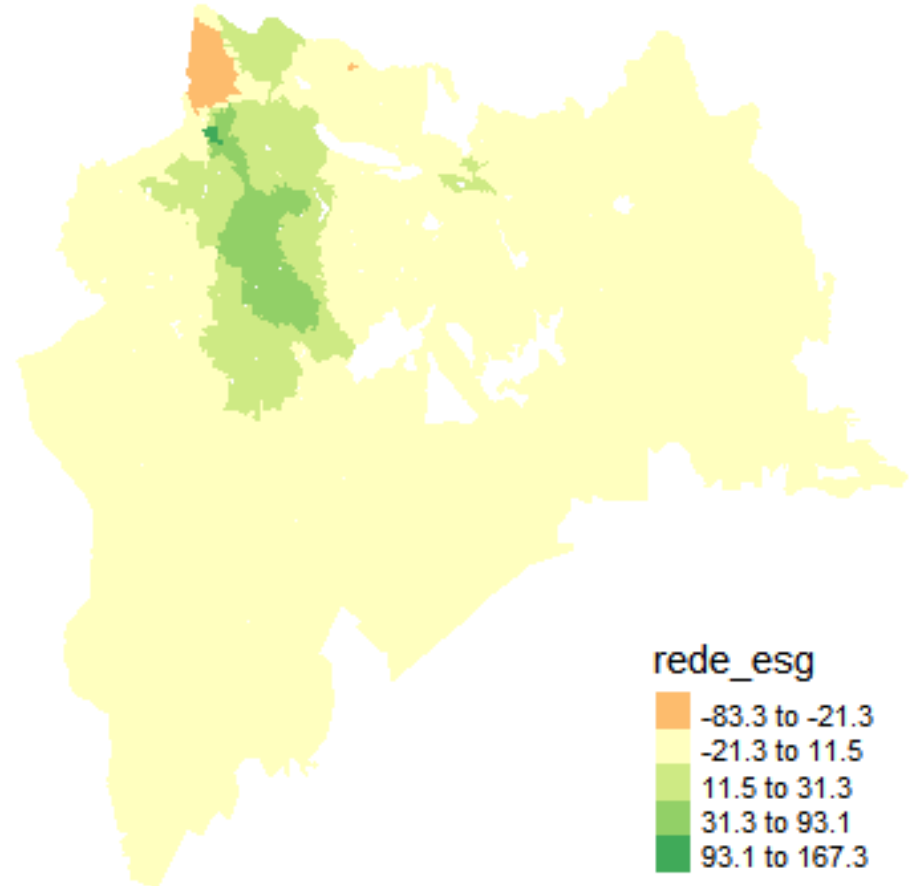
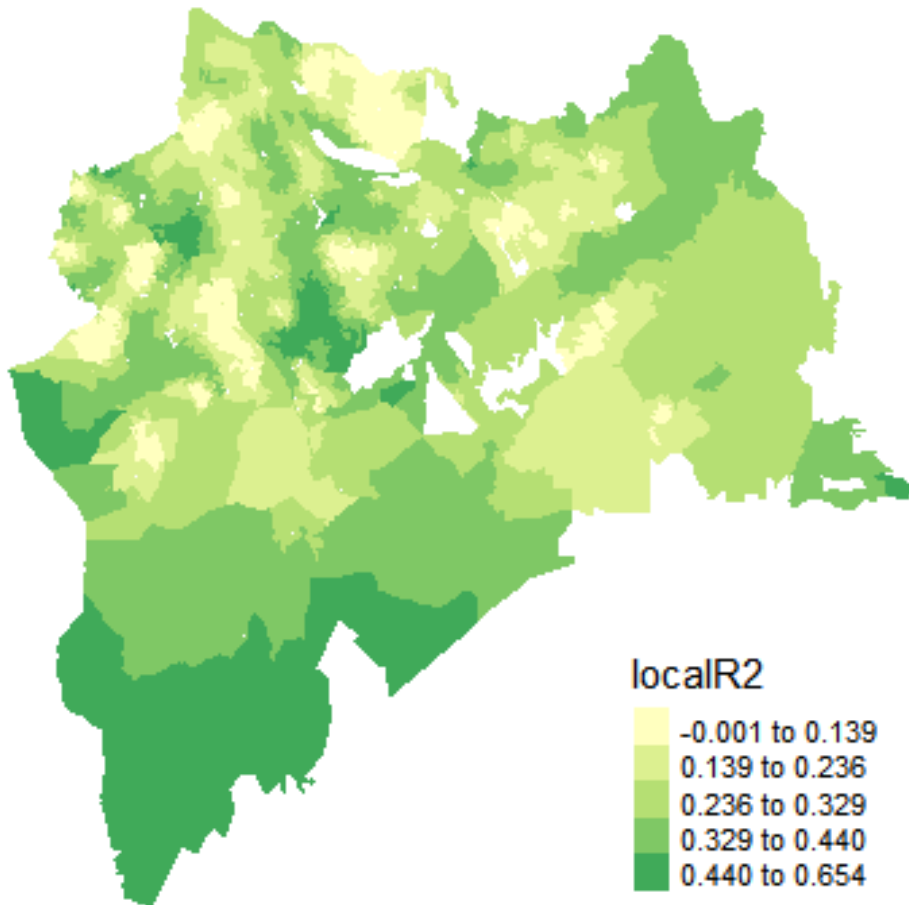
# ***GWR – Geographically Weighted Regression***

```
tm_shape(setores_gwr_adaptativo$SDF) +  
tm_fill("localR2", style="fisher")
```

**R<sup>2</sup> local**

**Classificação por  
quebras naturais**

**Coeficiente de  
porcentagem de  
rede de esgoto**

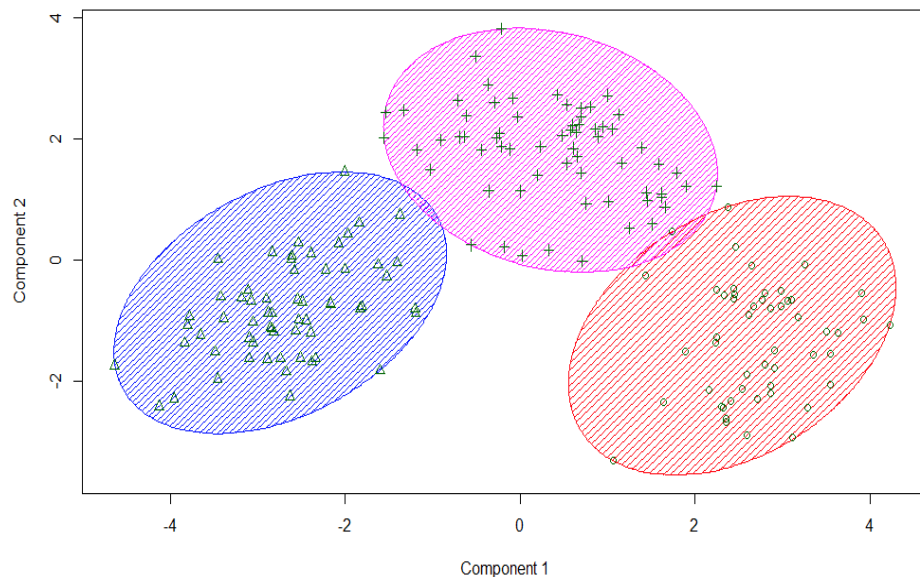


# Pacote GWmodel

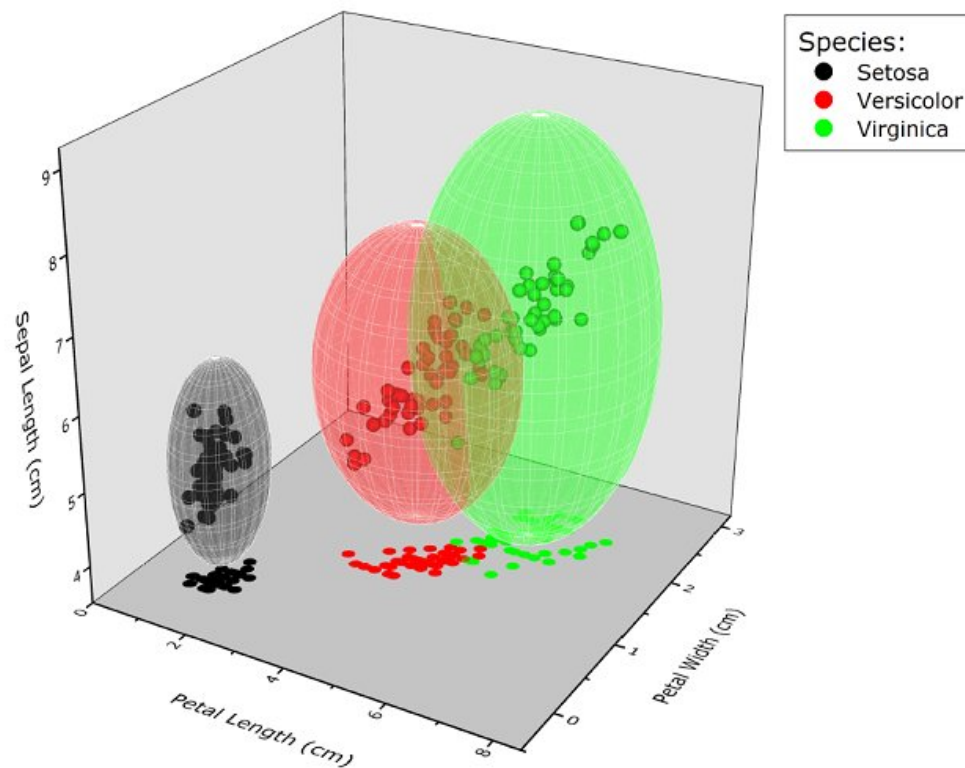
- Abordagem de análise geográfica ponderada (kernel) para:
  - Estatísticas descritivas
    - Média, desvio-padrão, normalidade
  - Modelos avançados de regressão geográfica
    - Generalizada, heteroscedástica, robusta, e semi-paramétrica (distribuições não-normais)
    - Ridge regression (variáveis com multicolinearidade)
    - Multiescalar (sintetiza vários raios de kernel)
    - Kernel 3D (espaço-temporal)
  - Análise de componentes principais
  - Análise discriminante (classificação)

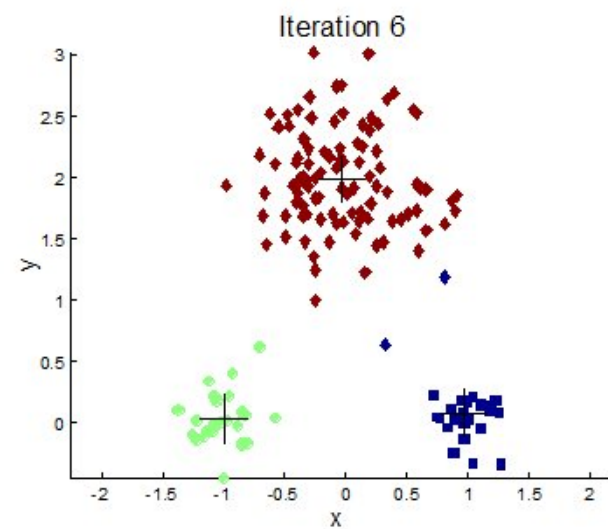
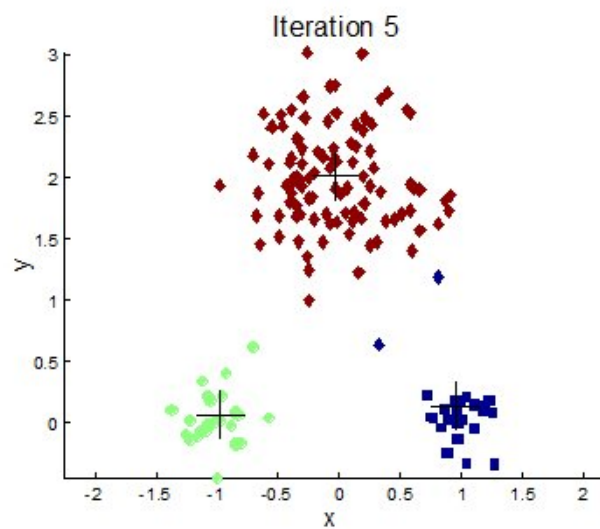
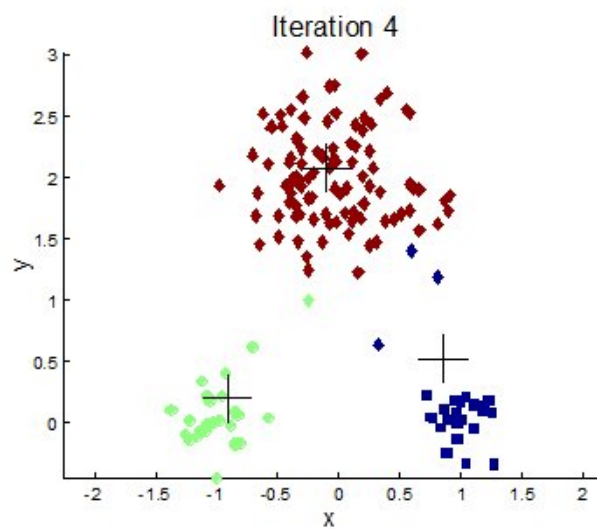
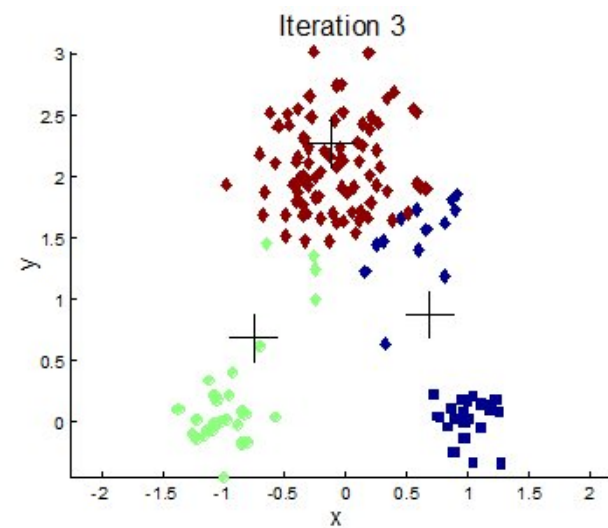
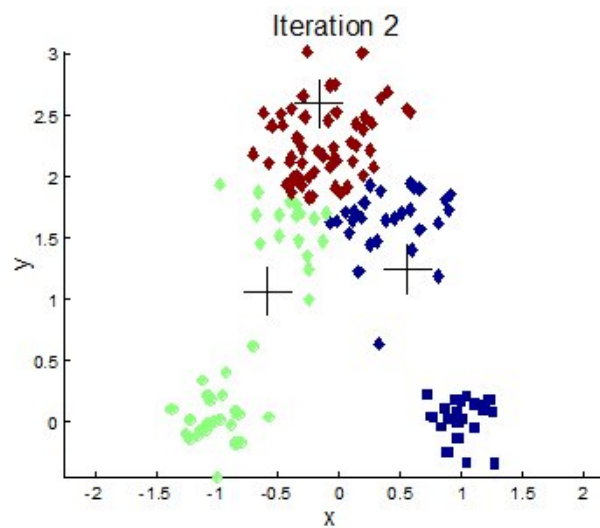
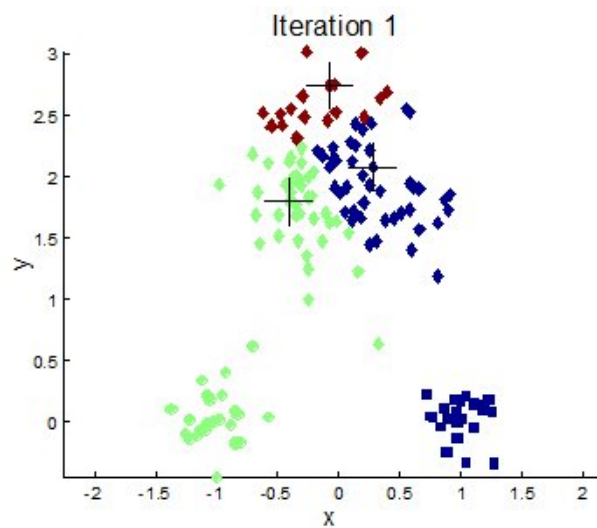
# Conteúdo

- Análise de vizinhança
- Autocorrelação espacial
- Suavização espacial
- Regressão espacial
- **Agrupamento**



These two components explain 55.61 % of the point variability.







# Agrupamento

- Padronizar as variáveis de entrada para que todas tenham a mesma influência

```
setores_normalizados <-  
as.data.frame(scale(setores_juntos@data[,c("Renda","rede_esg","col_lixo")]))
```

**Formato Padronização**  
**Data Frame**

**Variáveis a serem padronizadas**

```
View(setores_normalizados)
```

	Renda	rede_esg	col_lixo
1	-0.75641459	0.32135040	0.12045893
2	-0.83879912	0.41021004	0.12045893
3	-0.55677955	0.38904191	0.12045893

# Agrupamento

- Kmeans

```
agrupamento <- kmeans(setores_padronizados, centers = 4,  
iter.max = 10000, nstart=10000)
```

**Número de  
iterações**

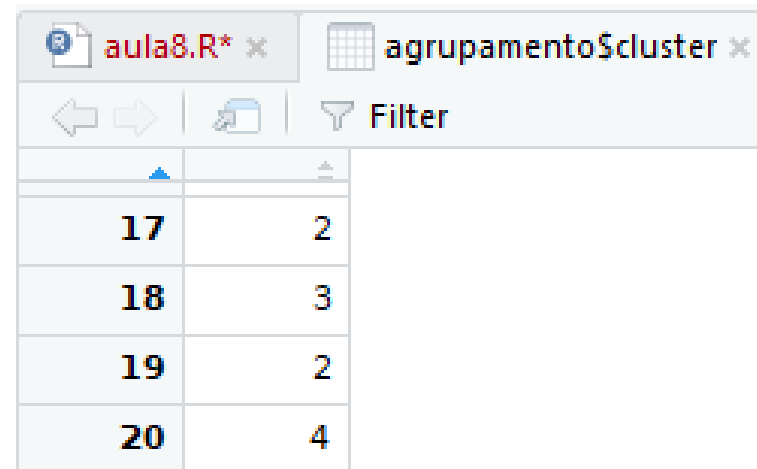
**Número de  
repetições**

**Número de  
grupos**

`agrupamento$centers`

	Renda	rede_esg	col_lixo
1	-0.2481984	-3.2965019	-10.07706542
2	-0.5970576	-2.7606881	-0.08160829
3	-0.1923904	0.2991240	0.11099831
4	2.5557501	0.3903877	0.11784712

`View(agrupamento$cluster)`



The screenshot shows the R Studio interface with two open windows: 'aula8.R\*' and 'agrupamento\$cluster'. The 'agrupamento\$cluster' window displays a table with 2 columns and 4 rows of data. The first column contains the values 17, 18, 19, and 20. The second column contains the values 2, 3, 2, and 4. The table is titled 'agrupamento\$cluster' and has a 'Filter' button in the top right corner.

17	2
18	3
19	2
20	4

```
setores_juntos$kmeans <- agrupamento$cluster
```

```
tm_shape(setores_juntos) +
```

```
tm_fill("kmeans", style = "cat", palette = c("blue","green", "yellow", "red"))
```

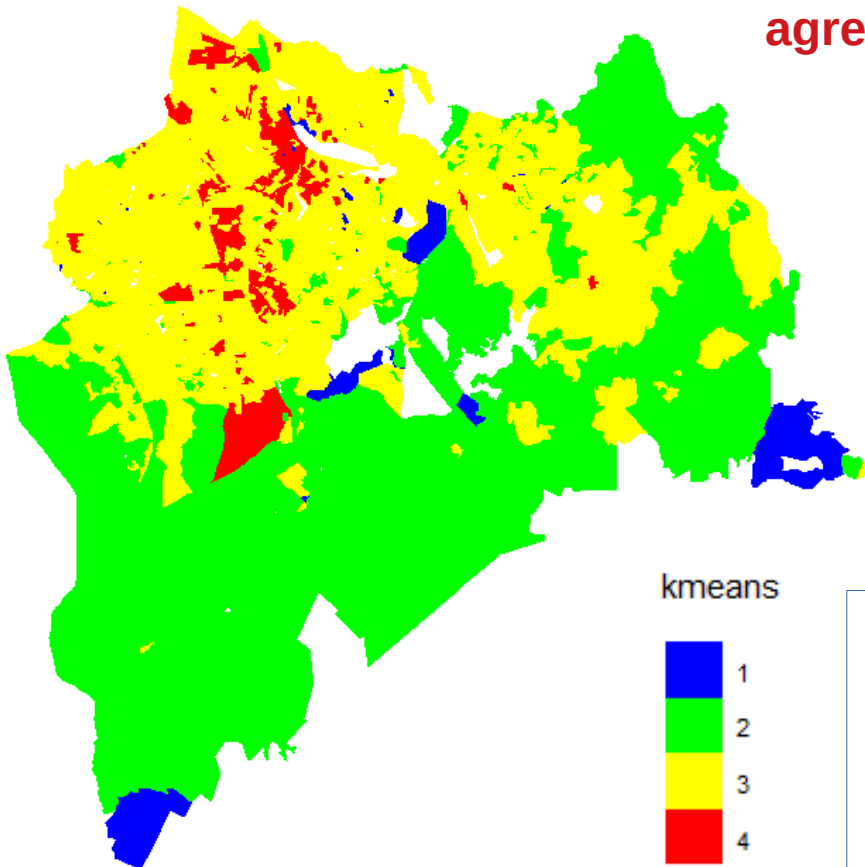
**Mapa categórico**

**Variáveis a serem agregadas**

```
aggregate(setores_juntos@data[,c("Renda","rede_esg","col_lixo")],  
by=list(setores_juntos$kmeans), FUN = mean)
```

**Atributo com os grupos**

**Cálculo da  
agregação (média)**



Group.1	Renda	rede_esg	col_lixo	
1	1	1323.1949	8.320363	3.086127
2	2	883.1432	21.572868	98.079621
3	3	1393.5911	97.252462	99.910089
4	4	4860.1006	99.509727	99.975178

# **Regionalização**

## **Método Skater**

### **Técnica:**

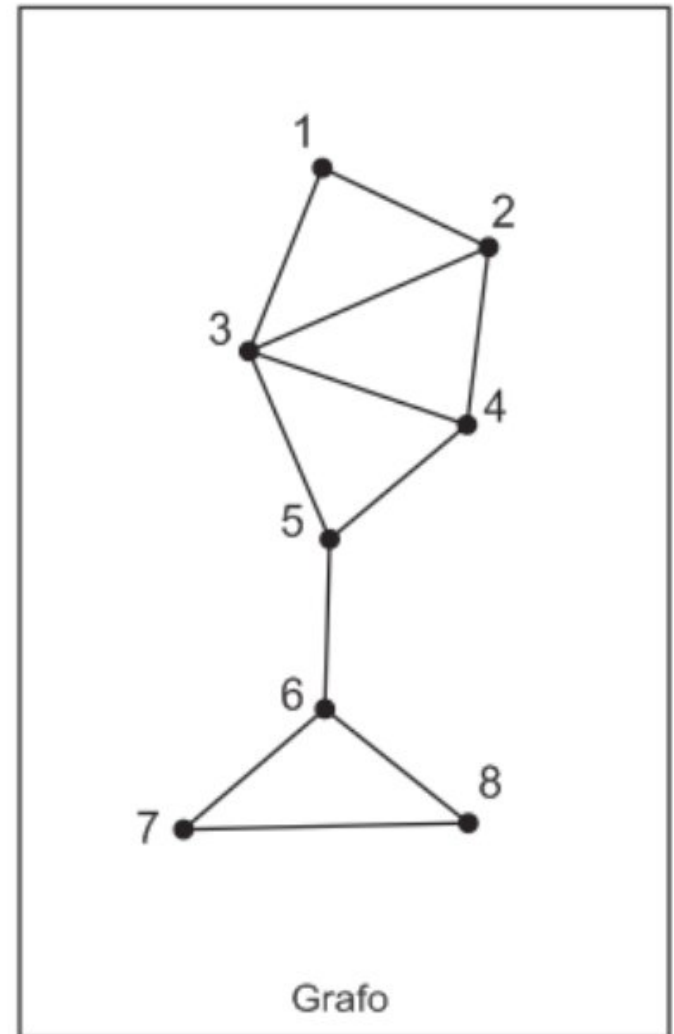
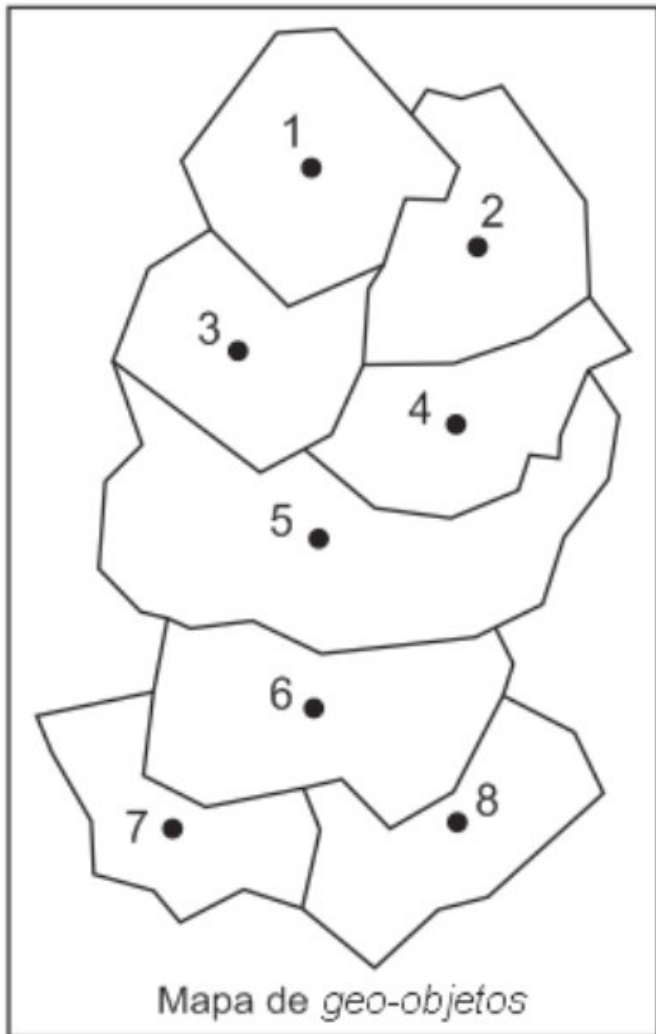
**Minimizar a variabilidade entre os agrupamentos, mas mantendo a contiguidade de todos os elementos em cada agrupamento**

### **Resultado:**

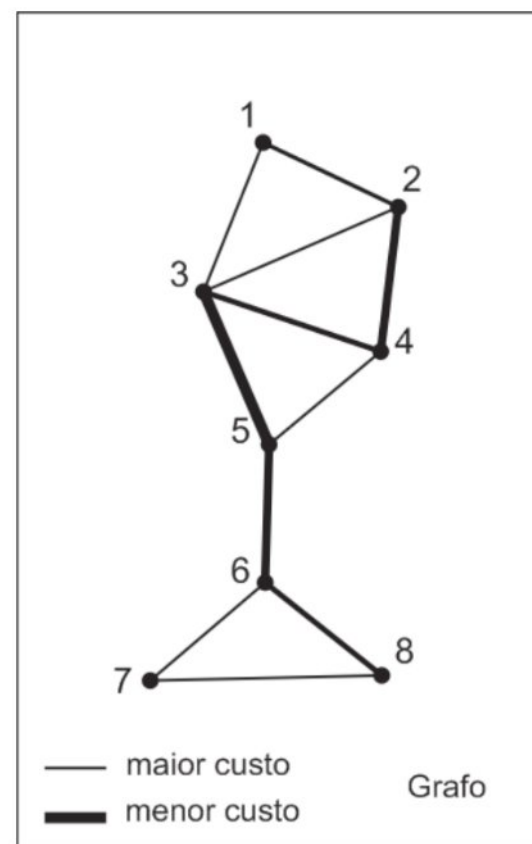
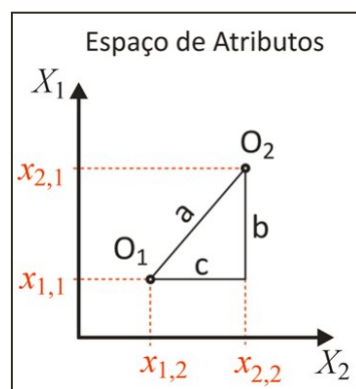
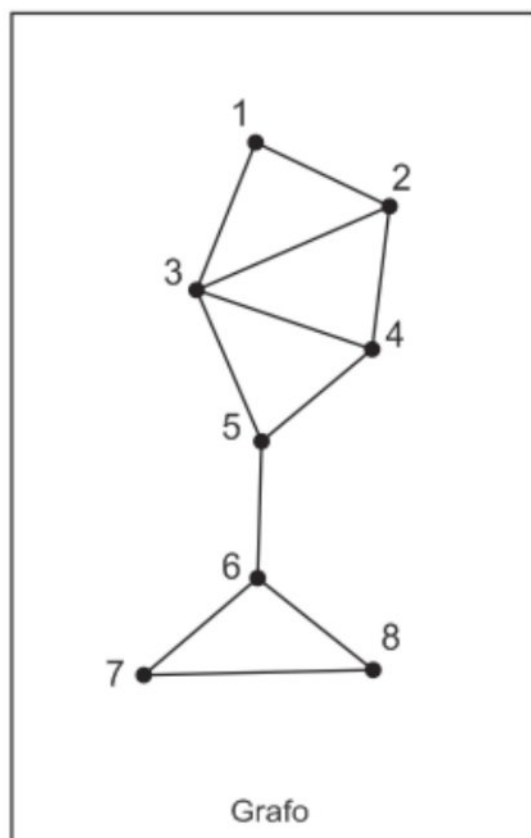
**Regiões relativamente homogêneas espacialmente contínuas**

# Regionalização

## Grafo de vizinhança

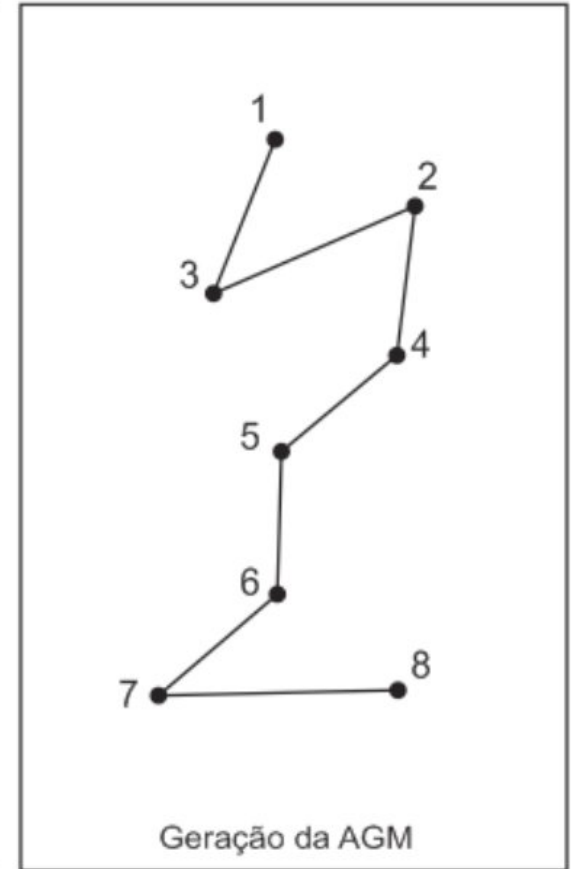
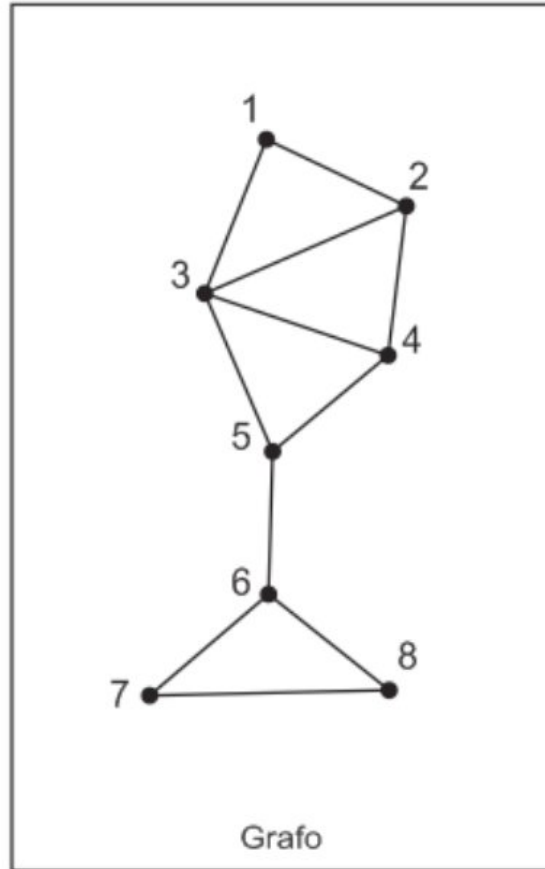
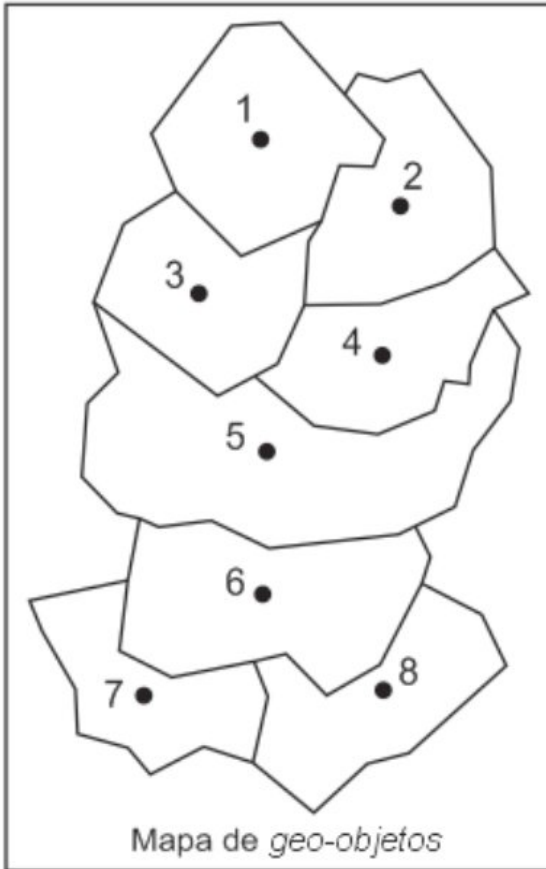


- Inicialmente associa-se "custos" às arestas
- Os "custos" são calculados em função da similaridade entre os geo-objetos
- Depois são eliminadas as arestas de menor "custo"

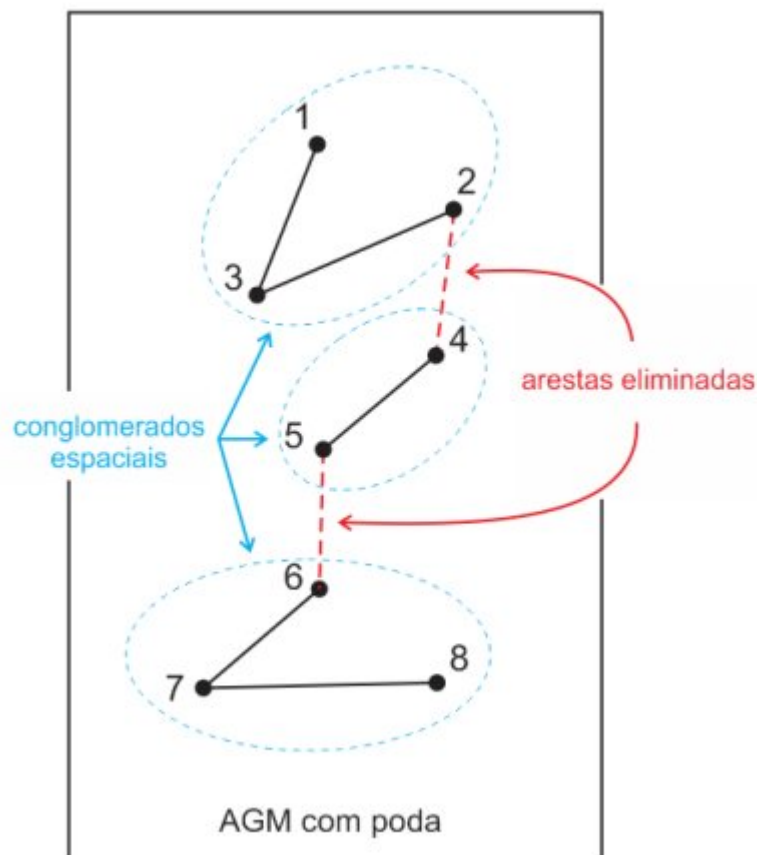


# Regionalização

## Árvore Geradora Mínima



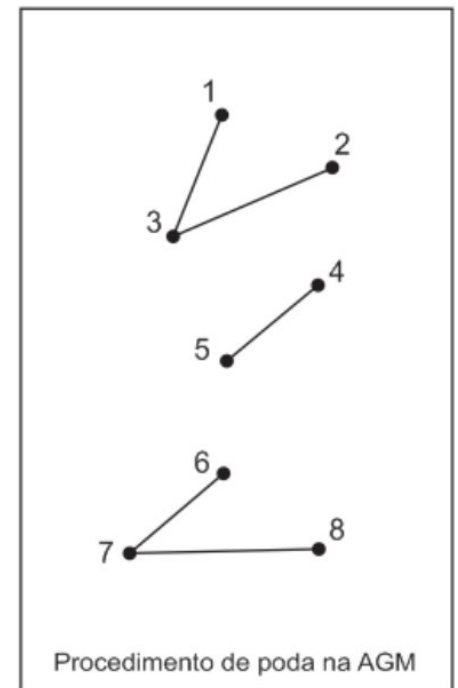
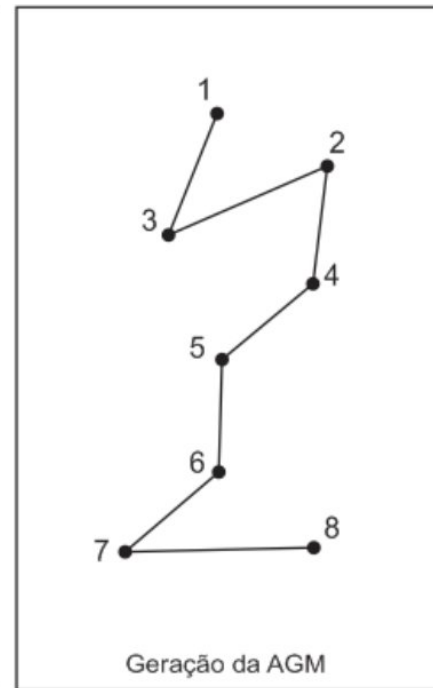
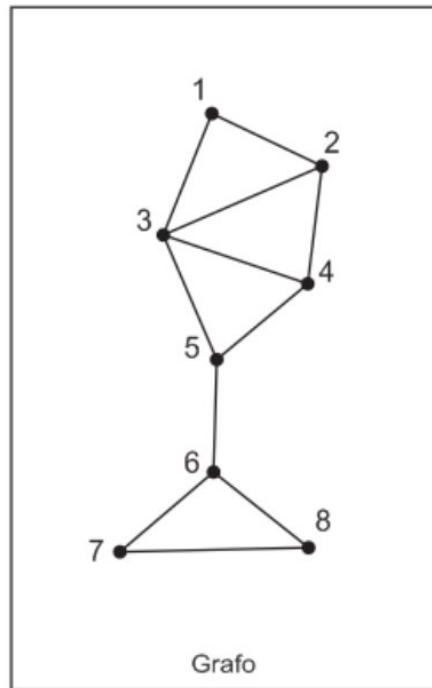
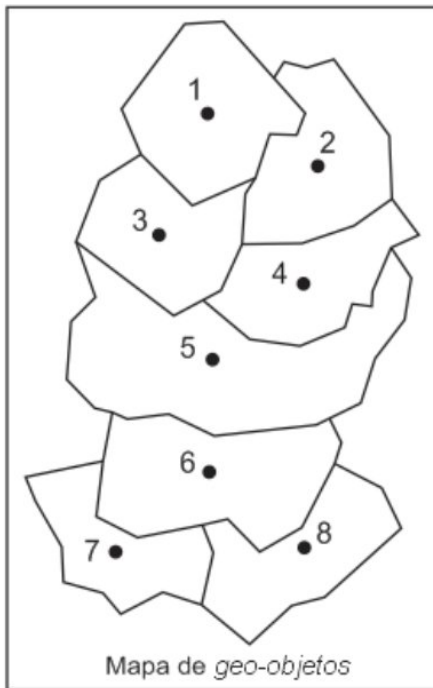
- Procedimento de poda procura obter:
  - regiões mais homogêneas
  - mais equilibradas em termos de números de geo-objetos por região.





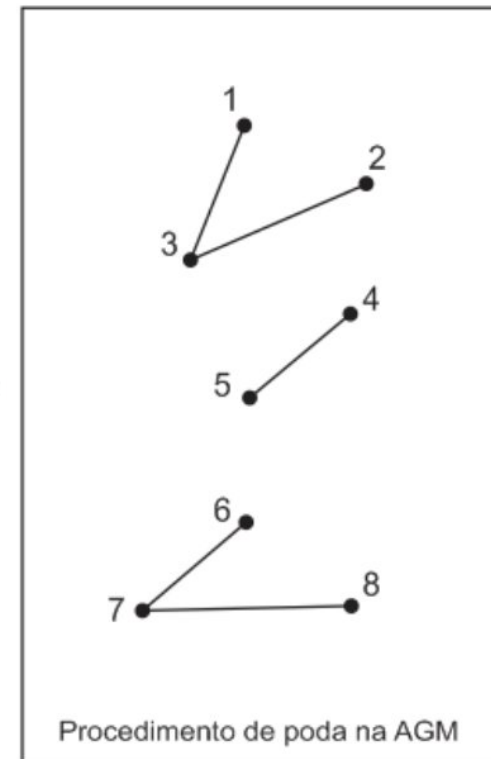
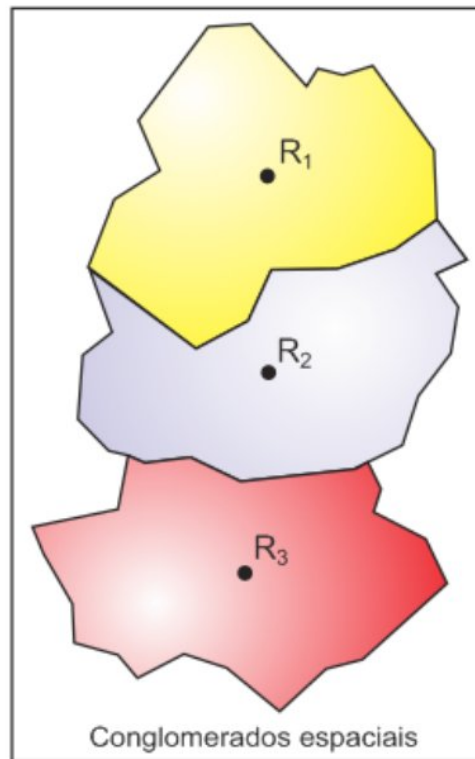
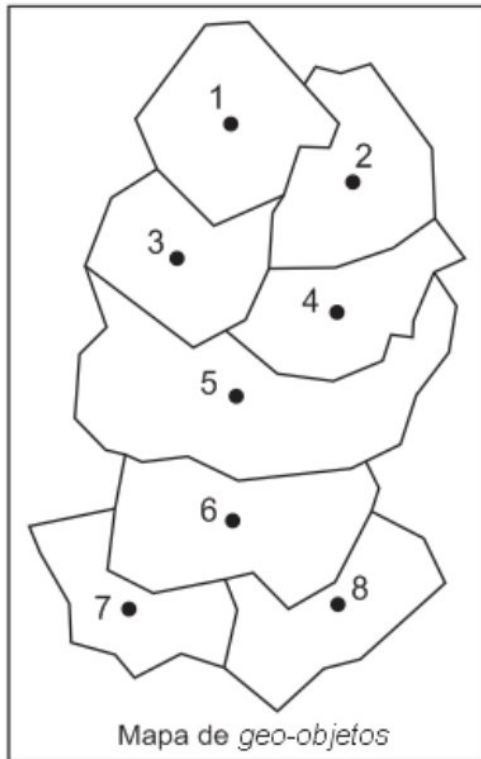
# Regionalização

## Árvore Geradora Mínima



# Regionalização

## Árvore Geradora Mínima



# **Passos para Regionalização**


- 1. Correção Topológica**
- 2. Matriz de Vizinhança**
- 3. Escolha dos Atributos**
- 4. Execução do Algoritmo de Regionalização**

# Skater

- Atribuir custos (grau de dissimilaridade) às relações de vizinhança

```
distancia_vizinhanca <- nbcosts(nb = vizinhanca2, data = setores_padronizados)
```

```
View(distancia_vizinhanca)
```

aula8.R* x distancia_vizinhanca x		
← →    Show Attributes		
Name	Type	Value
▼ distancia_vizinhanca	list [3995] (S3: nbdist)	List of length 3995
[[1]]	double [1]	0.4693201
[[2]]	double [9]	0.283 0.181 0.256 0.431 0.377 0.470 ...
[[3]]	double [2]	0.2828 0.0334

# Skater

- Atribuir custos (grau de dissimilaridade) às relações de vizinhança

**Peso de cada relação  
de vizinhança**

```
pesos_distancia <- nb2listw(vizinhanca2, glist=distancia_vizinhanca, style="B")
```

```
View(pesos_distancia)
```

**Matriz binária,  
não normalizada**




aula8.R x pesos_distancia x distancia_vizinhanca x		
Show Attributes		
Name	Type	Value
pesos_distancia	list [3] (S3: listw, nb)	List of length 3
style	character [1]	'B'
neighbours	list [3995] (S3: nb)	List of length 3995
weights	list [3995]	List of length 3995
[[1]]	double [1]	0.4693201
[[2]]	double [9]	0.283 0.181 0.256 0.431 0.377 0.470 ...
[[3]]	double [2]	0.2828 0.0334

# Skater

- Criar a árvore mínima geradora

```
mst_setores <- mstree(pesos_distancia)
```

```
View(mst_setores)
```

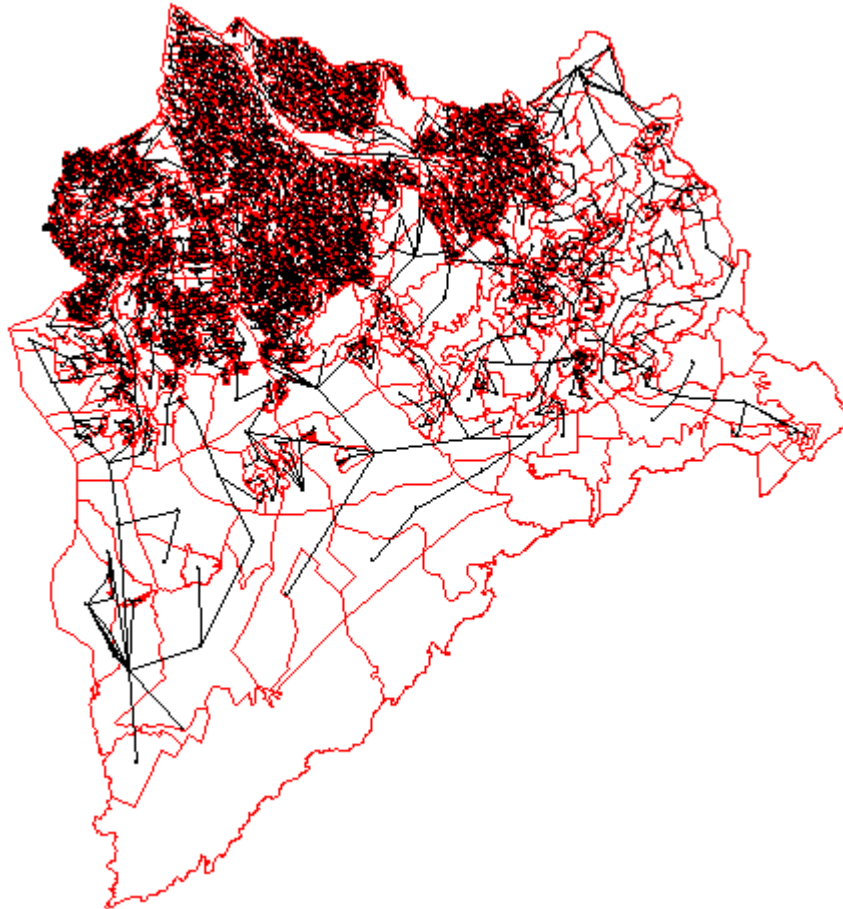
	Setor de origem	Setor de destino	Custo (dissimilaridade)
			
<b>1</b>	2888	2881	0.45630533
<b>2</b>	2881	2887	0.08568426
<b>3</b>	2881	2885	0.36348909

# Skater

```
plot(st_geometry(setores_sf), border="red")
```

```
plot(mst_setores, setores_xy, lab=NA, add=TRUE)
```

**Não mostrar  
nomes dos setores**



# NÃO RODAR! DEMORA MUITO!

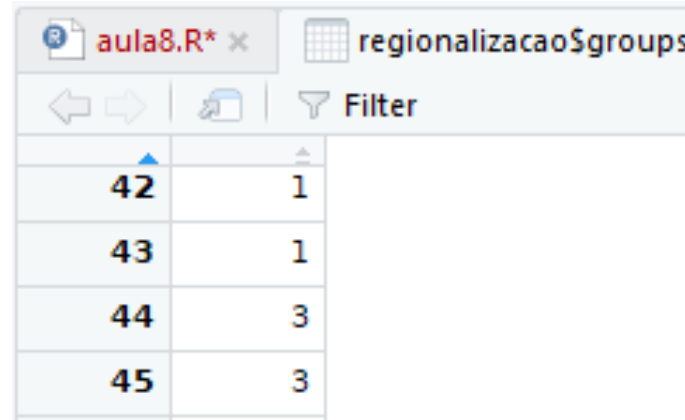
```
regionalizacao <- skater(mst_setores[,1:2], setores_padronizados,  
ncuts=3)
```

**Cortes na  
árvore**

```
View(regionalizacao$groups)
```

**Setores de  
origem e destino**

**Dados  
normalizados**



42	1
43	1
44	3
45	3

```
table(regionalizacao$groups)
```

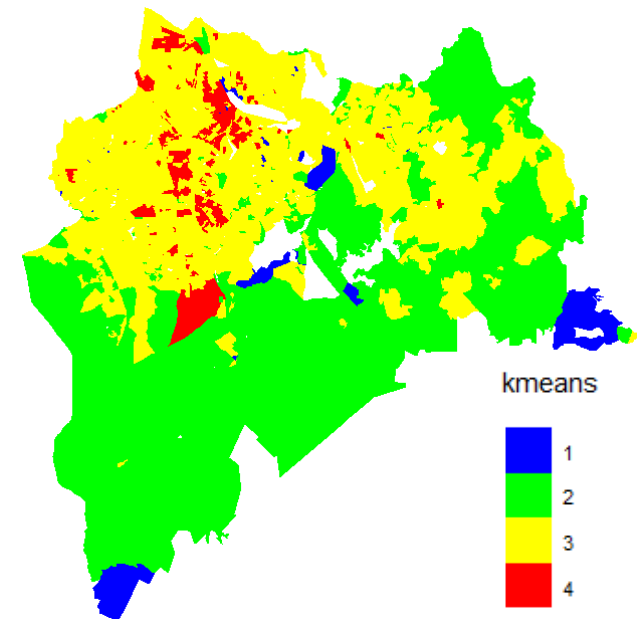
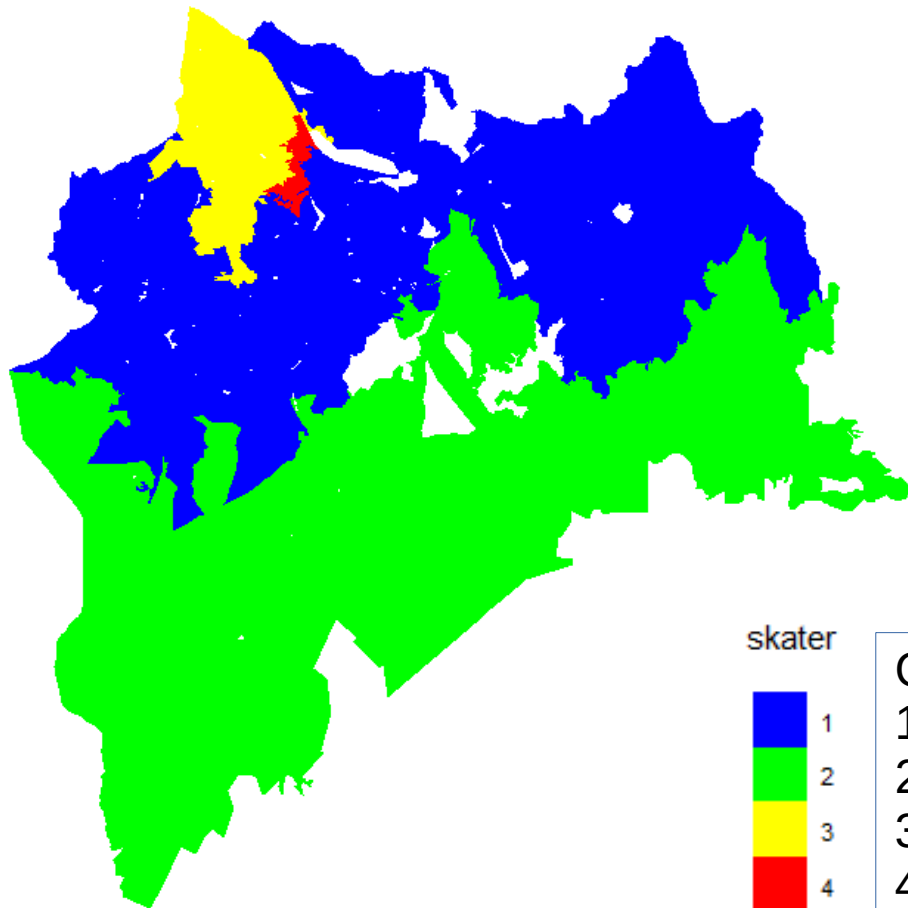
1	2	3	4
3141	276	520	58

```
setores_juntos$skater <- regionalizacao$groups
```



# Regionalização

```
tm_shape(setores_juntos) +  
  tm_fill("skater", style = "cat", palette = c("blue","green", "yellow", "red"))  
  
aggregate(setores_juntos@data[,c("Renda","rede_esg","col_lixo")],  
by=list(setores_juntos$skater), FUN = mean)
```



Group.1		Renda	rede_esg	col_lixo
1	1	1466.066	93.36553	99.21439
2	2	919.426	31.55389	94.58041
3	3	2596.243	98.69135	99.01988
4	4	5658.427	97.89045	98.26891

# Exercício 7

- A - Realize um agrupamento (kmeans) e uma regionalização (skater) de 3 grupos para os municípios da RMSP, considerando os dados socioeconômicos disponíveis (menos população).
- B - Agregue os atributos com a média de cada grupo ou região

# Obrigado!

**Ângela Terumi Fushita**  
**Vitor Vieira Vasconcelos**