

# Estatística Espacial (MI418) / Geoestatística (ME907)

Guilherme Ludwig

2019-02-01

Reticulados irregulares

Teste de Moran

Estimação dos modelos CAR, SAR

## Reticulados irregulares



Figure 1: Mapa da Região Metropolitana de Campinas (da Wikipédia)

## Reticulados irregulares

Algumas dicas sobre como usar dados do site do IBGE aqui:

<http://rpubs.com/andreasancheztapia/176768>

Em particular, para dados de lattice, temos interesse em recuperar malhas, disponíveis no site

<https://www.ibge.gov.br/geociencias-novoportal/downloads-geociencias.html>

Seguindo os links: Geociências -> Listar Tudo -> Malhas -> Malha Municipal / Por Estado (SP)

## Reticulados irregulares

```
library(rgdal) # kind of annoying to insatall in Linux
library(maptools)
m <- readOGR("municipal_sp/35MUE250GC_SIR.shp",
             stringsAsFactors = FALSE,
             use_iconv = TRUE, encoding = "utf8")
```

```
## OGR data source with driver: ESRI Shapefile
```

```
## Source: "/home/gvludwig/Dropbox/Classes/2019s0-me907-mi418/municipal_sp.
```

```
## with 645 features
```

```
## It has 2 fields
```

Note: `use_iconv = TRUE`, `encoding = "utf8"` é porque o site do IBGE possui acentos nos nomes.

## Reticulados irregulares

Força bruta para pegar as cidades que eu quero.

```
rmc <- c("CAMPINAS", "VALINHOS", "VINHEDO", "ITATIBA",  
        "MORUNGABA", "PEDREIRA", "JAGUARIÚNA",  
        "SANTO ANTÔNIO DE POSSE", "HOLAMBRA",  
        "ARTUR NOGUEIRA", "ENGENHEIRO COELHO",  
        "COSMÓPOLIS", "AMERICANA", "PAULÍNIA",  
        "NOVA ODESSA", "SUMARÉ", "HORTOLÂNDIA",  
        "MONTE MOR", "INDAIATUBA", "SANTA BÁRBARA D'OESTE")  
RMC <- m[m@data$NM_MUNICIP %in% rmc, ]
```

Podemos inserir covariáveis aqui; note que o slot @data é simplesmente um `data.frame`.

## Reticulados irregulares

```
head(RMC@data)
```

```
##           NM_MUNICIP CD_GEOCMU
## 18           AMERICANA  3501608
## 43    ARTUR NOGUEIRA  3503802
## 108          CAMPINAS  3509502
## 143    COSMÓPOLIS    3512803
## 171 ENGENHEIRO COELHO  3515152
## 218          HOLAMBRA  3519055
```

## Reticulados irregulares

```
str(x <- read.csv("rmcWikipedia.csv", stringsAsFactors = FALSE,  
                 row.names = 1))
```

```
## 'data.frame':    20 obs. of  3 variables:  
## $ Pop : int  231621 51126 1173370 67960 19059 13698 219039 235367 1149  
## $ HDI : num  0.84 0.786 0.852 0.799 0.792 0.827 0.79 0.829 0.828 0.829  
## $ Gini: num  0.4 0.4 0.42 0.399 0.42 0.43 0.37 0.4 0.4 0.4 ...
```

```
# Lines match, otherwise need to match by name
```

```
RMC@data <- cbind(RMC@data, x)
```



# Reticulados irregulares



## Reticulados irregulares



## Relação de adjacência

A função `poly2nb` por default acha vizinhanças baseadas em adjacência.

```
library(spdep)
RMC_nb <- poly2nb(RMC, row.names = row.names(RMC))
str(RMC_nb)
```

```
## List of 20
## $ : int [1:4] 4 13 14 16
## $ : int [1:3] 4 5 6
## $ : int [1:10] 7 8 9 10 11 12 14 15 18 19
## $ : int [1:4] 1 2 6 14
## $ : int 2
## $ : int [1:5] 2 4 10 14 17
## $ : int [1:3] 3 11 18
## $ : int [1:2] 3 11
## $ : int [1:4] 3 12 19 20
## $ : int [1:5] 3 6 14 15 17
```

## Relação de adjacência

```
summary(RMC_nb)
```

```
## Neighbour list object:  
## Number of regions: 20  
## Number of nonzero links: 82  
## Percentage nonzero weights: 20.5  
## Average number of links: 4.1  
## Link number distribution:  
##  
## 1 2 3 4 5 6 7 10  
## 1 2 5 6 3 1 1 1  
## 1 least connected region:  
## 171 with 1 link  
## 1 most connected region:  
## 108 with 10 links
```

## Relação de adjacência

A função `nb2listw` encontra, com base em uma relação de vizinhos, a matriz de pesos. Há alguns critérios importantes para se decidir quando escolhermos a matriz de pesos, na entrada `style = "B"`. Eu estou usando "B" que é a mais simples (binary) mas há diversas outras opções, por exemplo ponderadas pela conectividade.

```
RMC_weights <- nb2listw(RMC_nb, style = "B")  
coords <- coordinates(RMC)  
str(coords)
```

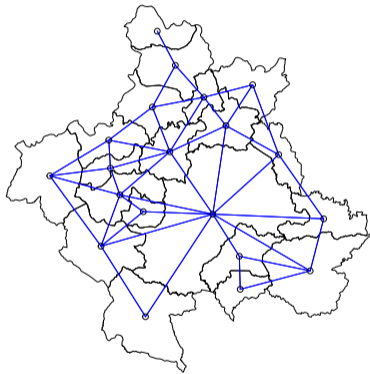
```
## num [1:20, 1:2] -47.3 -47.1 -47 -47.2 -47.2 ...  
## - attr(*, "dimnames")=List of 2  
## ..$ : chr [1:20] "18" "43" "108" "143" ...  
## ..$ : NULL
```

## Relação de adjacência

```
plot(RMC, main = "Região Metropolitana de Campinas")  
plot(RMC_weights, coords, col = "blue", add = TRUE)
```

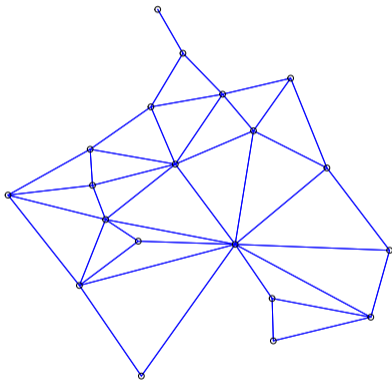
# Relação de adjacência

**Região Metropolitana de Campinas**



# Triangulação de Delaunay

Região Metropolitana de Campinas





## Modificando relação de adjacência

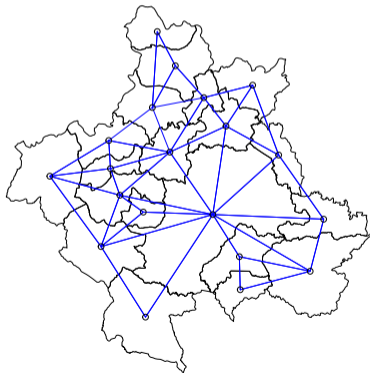
Com cuidado, é possível conectar e desconectar centros usando as funções disponíveis. Você deve saber a posição relativa dos centros que deseja conectar. Por exemplo, conectando Cosmópolis e Engenheiro Coelho:

```
RMC_nb2 <- RMC_nb
RMC_nb2[[4]] <- c(RMC_nb2[[5]], 5L) # row: Engenheiro Coelho
RMC_nb2[[5]] <- c(RMC_nb2[[5]], 4L) # row: Cosmópolis
RMC_weights2 <- nb2listw(RMC_nb2, style = "B")
```

## Modificando relação de adjacência

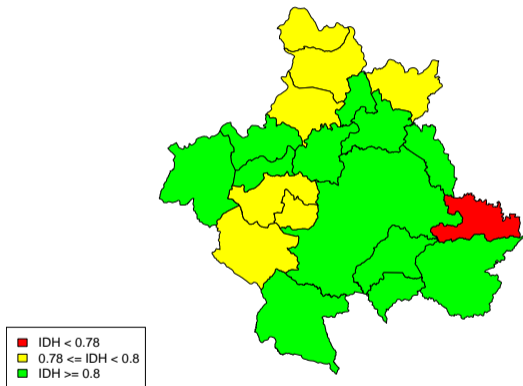
Com cuidado, é possível conectar e desconectar centros usando as funções disponíveis. Você deve saber a posição relativa dos centros que deseja conectar. Por exemplo, conectando Cosmópolis e Engenheiro Coelho:

**Região Metropolitana de Campinas**



# Teste de Moran

Região Metropolitana de Campinas: IDH



## Teste de Moran

Se  $Z_i$  é a variável de interesse no local  $s_i$ , e  $\mathbf{W}$  é a matriz de adjacências binária, então a estatística de Moran é

$$I = \frac{\mathbf{Z}^t \mathbf{W} \mathbf{Z}}{\mathbf{Z}^t \mathbf{Z}}$$

Sob  $H_0$  : independência espacial,  $\mathbb{E}(I) = -1/(n + 1)$  e  $\text{Var}(I) = \sigma_I^2$ , onde  $\sigma_I^2$  é alguma coisa bem complicada, mas com forma fechada

([https://en.wikipedia.org/wiki/Moran%27s\\_I](https://en.wikipedia.org/wiki/Moran%27s_I)). Assintoticamente a estatística  $I$  é Normal.

Na prática, podemos fazer um teste de permutações dos valores  $Z_i$ , o que é mais seguro para  $n$  pequeno.

## Teste de Moran

```
set.seed(1)
moran.test(RMC$HDI, RMC_weights, alternative = "two.sided")

##
## Moran I test under randomisation
##
## data: RMC$HDI
## weights: RMC_weights
##
## Moran I statistic standard deviate = 0.63286, p-value = 0.5268
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.02448111      -0.05263158      0.01484698
```

## Dados da Secretaria de Segurança Pública

Pra tentar olhar algo mais interessante, vou baixar dados da SSP de São Paulo, do site:  
<http://www.ssp.sp.gov.br/Estatistica/Mapas.aspx>

```
str(mu <- read.csv("rmcSSP.csv", stringsAsFactors = FALSE,  
                  row.names = 1))
```

```
## 'data.frame':    20 obs. of  1 variable:  
## $ Murder: int  9 2 136 6 4 0 22 7 6 3 ...
```

```
RMC@data <- cbind(RMC@data, mu)
```

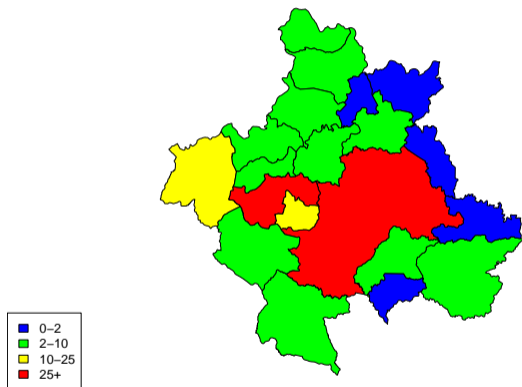
## Dados da Secretaria de Segurança Pública

```
head(RMC@data)
```

##	NM_MUNICIP	CD_GEOCMU	Pop	HDI	Gini	Murder
## 18	AMERICANA	3501608	231621	0.840	0.400	9
## 43	ARTUR NOGUEIRA	3503802	51126	0.786	0.400	2
## 108	CAMPINAS	3509502	1173370	0.852	0.420	136
## 143	COSMÓPOLIS	3512803	67960	0.799	0.399	6
## 171	ENGENHEIRO COELHO	3515152	19059	0.792	0.420	4
## 218	HOLAMBRA	3519055	13698	0.827	0.430	0

# Dados da Secretaria de Segurança Pública

Região Metropolitana de Campinas: Homicídio Doloso em 2017





## Teste de Moran

```
set.seed(1)
moran.test(RMC$Murder, RMC_weights, alternative = "two.sided")

##
## Moran I test under randomisation
##
## data: RMC$Murder
## weights: RMC_weights
##
## Moran I statistic standard deviate = -1.0527, p-value = 0.2925
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic      Expectation      Variance
##      -0.120798534      -0.052631579      0.004193463
```

## Teste de Moran

Falha por problemas de convergência

```
model <- glm(Murder ~ HDI + offset(Pop),  
            family = poisson(link = 'log'), data = RMC@data)
```

## Teste de Moran

Saída de emergência

```
summary(model <- lm(sqrt(Murder / Pop) ~ HDI, data = RMC@data))
```

```
##  
## Call:  
## lm(formula = sqrt(Murder/Pop) ~ HDI, data = RMC@data)  
##  
## Residuals:  
##           Min           1Q           Median           3Q           Max  
## -0.0072269 -0.0015424  0.0006187  0.0019701  0.0065158  
##  
## Coefficients:  
##           Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  0.04072   0.02044   1.993  0.0617 .  
## HDI          -0.04135   0.02512  -1.647  0.1170  
## ---
```

## Teste de Moran

Não tem correlação mesmo...

```
set.seed(1)
moran.test(model$residuals, RMC_weights, alternative = "two.sided")
```

```
##
## Moran I test under randomisation
##
## data:  model$residuals
## weights: RMC_weights
##
## Moran I statistic standard deviate = 0.57307, p-value = 0.5666
## alternative hypothesis: two.sided
## sample estimates:
```

## Moran I statistic	Expectation	Variance
## 0.02143038	-0.05263158	0.01670234

## Modelo CAR, SAR Gaussiano

Modelo incorreto pois não há dependência: só para ilustrar...

```
modelC <- spautolm(sqrt(Murder / Pop) ~ HDI, data = RMC@data,  
                  family = "CAR", listw = RMC_weights)  
modelS <- spautolm(sqrt(Murder / Pop) ~ HDI, data = RMC@data,  
                  family = "SAR", listw = RMC_weights)
```

## Modelo CAR Gaussiano

```
summary(modelC)
```

```
##
```

```
## Call:
```

```
## spautolm(formula = sqrt(Murder/Pop) ~ HDI, data = RMC@data, listw = RMC
```

```
##     family = "CAR")
```

```
##
```

```
## Residuals:
```

```
##           Min           1Q           Median           3Q           Max
## -0.00725278 -0.00133063  0.00066989  0.00181735  0.00659038
```

```
##
```

```
## Coefficients:
```

```
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.040882  0.019381  2.1094  0.03491
## HDI          -0.041593  0.023831 -1.7453  0.08093
```

```
##
```

```
## Lambda: 0.000777 LR test value: 0.000047 p-value: 0.00005
```

## Modelo SAR Gaussiano

```
summary(modelS)
```

```
##  
## Call:  
## spautolm(formula = sqrt(Murder/Pop) ~ HDI, data = RMC@data, listw = RMC.  
##      family = "SAR")  
##  
## Residuals:  
##           Min           1Q           Median           3Q           Max  
## -0.00722317 -0.00140329  0.00066646  0.00188478  0.00657067  
##  
## Coefficients:  
##           Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  0.040891  0.019383  2.1097  0.03489  
## HDI          -0.041605  0.023835 -1.7456  0.08088  
##  
## Lambda: 0.018005 LR test value: 0.001201 p-value: 0.99206
```

## Modelos não-Gaussianos

Vocês podem olhar o pacote `spBayes`, em particular `spBayes::spGLM`.