



## Alexandre Toshio Igari

Sou doutorando no Laboratório de Ecologia de Paisagens e Conservação (LEPAC), no Departamento de Ecologia da USP, sob orientação da Prof. Dra. Vânia Regina Pivello. Meu projeto de doutorado aborda mecanismos econômicos e legais para conservação de vegetação nativa em áreas agrícolas privadas, e meu interesse na disciplina é conhecer uma nova abordagem para a análise dos dados da minha pesquisa.

### Proposta de Função em R

#### Contextualização

O meu trabalho aborda a relação entre fatores econômicos (como o crédito rural tomado por fazendeiros no município) e a quantidade de áreas nativas nas fazendas (APP + Reserva Legal total no município). Os meus dados representam os 645 municípios do estado de São Paulo e, na análise exploratória dos dados, eu gostaria de identificar se os atributos CRÉDITO RURAL e ÁREAS NATIVAS estão relacionados com a distância entre cada município e os demais. O cenário nulo é que municípios próximos não diferem de municípios distantes quanto a CRÉDITO RURAL ou ÁREAS NATIVAS. A hipótese alternativa é que municípios próximos são mais semelhantes em relação a CRÉDITO RURAL ou ÁREAS NATIVAS do que municípios distantes.

#### Plano A

Eu gostaria de fazer uma função genérica que lesse um dataframe com o nome da localidade, coordenadas x/y (em UTM ou em qualquer unidade de medida) e a variável (CRÉDITO, ÁREAS NATIVAS, ou outras) que será testada espacialmente:



Em seguida, serão geradas duas matrizes de distância, uma com a distância euclidiana entre as localidades e outra com a diferença com relação às variáveis de interesse (dados fictícios):



A partir dos dados das matrizes, será gerado um gráfico com a distribuição das DIFERENÇAS entre os valores recebidos de crédito (que pode ser também qualquer outra variável de interesse) e as distâncias euclidianas entre as cidades. Este resultado será comparado com uma aleatorização dos dados, testando a hipótese do cenário nulo (isso eu pretendo aprender a fazer na próxima semana). Seria desejável também elaborar um modelo explicativo da relação entre as duas matrizes, o que poderia gerar uma função complementar. A estrutura da função permitiria identificar outros casos de autocorrelação espacial em outros conjuntos de dados, Por exemplo quando temos x e y como coordenadas de parcelas e a variável de interesse = diversidade, riqueza, DAP, altura.

## Comentários Ale

Parece bem bacana, factível e útil. Acho que começar apenas com a produção das matrizes de distâncias é mais prudente (plano B). Caso consiga avançar para o modelo nulo, pense em uma função independente cuja a entrada são as matrizes geradas pela primeira função. Geralmente é o que fazemos: decompomos o problema em funções subordinadas e que depois podem ser

concatenadas em uma função final. Quanto ao modelo nulo, não sei se entendi direito, mas parece que vc. quer fazer um teste de Mantel, que olha a correlação entre os valores de duas matrizes e cria um cenário nulo permutando uma delas e calculando a correlação muitas vezes (há uma solução analítica, mas gosto da permutação). Isso é legal tb, mas dá mais trabalho... posso te ajudar com isso depois que terminar a parte de gerar as matrizes!

## Plano B

O plano B, mais simples, pode ser somente ler os dados, calcular as distâncias euclidianas e montar as matrizes e calcular uma correlação (Pearson ou Spearman, dependendo das características dos dados)

# Comentários Paulo

Cuidado para que sua função fique muito específica apenas para seus dados. Pense em como generalizar para um problema geral na área.

---

## Ajuda da Função

### cor.esp()

### Correlacao espacial entre distancias geograficas e diferencas de uma variavel de interesse

#### Descricao

Calcula a correlacao entre as distancias euclidianas de um conjunto de localidades e as diferencas de uma dada variavel de interesse de localidade para localidade

#### Uso

`cor.esp(tabela, plot.entr=TRUE, plot.hist=TRUE, metodo="pearson", permut=10000)`

#### Argumentos

tabela: dataframe com 4 colunas (com titulos de colunas): coluna 1 - identificacao das localidades; colunas 2 e 3 - coordenadas geograficas x e y das localidades; coluna 4 - variavel de interesse associada as localidades (quantidade de areas nativas, riqueza de especies).

plot.entr: TRUE - retorna 2 graficos representando os dados de entrada

plot.hist: TRUE - retorna o histograma com os valores dos coeficientes de correlacao (pearson ou spearman) e o valor de p resultante das permutacoes

metodo: metodo de correlacao "pearson" ou "spearman"

permut: numero de permutacoes para o calculo do p da correlacao espacial

## Detalhes

Os nomes das colunas e nomes das localidades da “tabela” serao usados para identificar titulos, eixos de graficos e dimensoes das matrizes de saida de resultados. Assim, nao e conveniente utilizar nomes grandes, para facilitar a visualizacao dos resultados.

Para a visualizacao de conjuntos de dados grandes, recomenda-se, para uma primeira visualizacao dos dados, selecionar `plot.hist=FALSE` e `permut=100` (ou menor). Com isso a visualizacao dos dados de entrada por meio de `plot.entr=TRUE` sera mais rapida e apresentada maior no dispositivo grafico. Caso se deseje expandir a visualizacao dos dados de entrada, basta aumentar o tamanho do dispositivo grafico e rodar novamente a funcao.

Com `plot.entr= TRUE` e `plot.hist=TRUE`, o histograma sera apresentado juntamente com os graficos de entrada. Se quisermos visualizar somente o histograma ocupando todo o dispositivo grafico, basta selecionar `plot.entr=FALSE` e `plot.hist=TRUE`.

O histograma de saida (`plot.hist`) refere-se sempre ao metodo de correlacao selecionado (`pearson` ou `spearman`).

## Valores

A tabela (dataframe de entrada) nao deve conter celulas sem informacao (NA) e e preciso verificar se as celulas com zeros nao representam na verdade NA. Celulas com NA ou falsos zeros podem viesar o resultado da correlacao.

O grafico Localizacao x Variavel de interesse permite visualizar a localizacao geografica das localidades (xy) e distinguir as localidades quanto aos quartis da variavel de interesse.

O grafico de correlacao espacial permite a visualizacao da correlacao entre as matrizes de distancia apresentadas no console (matriz de distancia euclidiana entre as localidades x diferencas da variavel de interesse de localidade para localidade).

O histograma retorna as frequencias do coeficiente de correlacao (R quando Pearson e rho quando Spearman) decorrentes das permutacoes da variavel de interesse. A linha vermelha representa o valor obtido (R ou rho) e o p representa a probabilidade acumulada até o extremo da distribuicao. Quando o coeficiente (R ou rho) e negativo, o p refere-se a probabilidade acumulada entre a linha vermelha e o extremo negativo. Quando o coeficiente e positivo, p refere-se a probabilidade acumulada entre a linha vermelha e o extremo positivo. O coeficiente (R ou rho) e o p estao expressos no titulo do grafico.

## Nota

Quando o metodo escolhido e “spearman”, e comum surgirem mensagens de aviso no console. As mensagens de aviso referem-se a funcao `cor.test()` utilizada para calcular o rho de spearman. No calculo do rho, a funcao avisa que nao e possivel retornar o p exato. Os avisos nao influenciam no calculo do rho, e o p da funcao `cor.esp` e obtido posteriormente por meio da permutacao dos vetores de distancia e do calculo de rho para cada uma das permutacoes.

Um numero grande de permutacoes (maior que 1000) pode fazer com que funcao demore alguns segundos para retornar os resultados. O numero padrao e de 10000 permutacoes, mas pode ser reduzido quando se deseja apenas visualizar rapidamente os graficos de entrada e matrizes de

distancia.

## Autor:

Alexandre Toshio Igari

alexandre.igari@usp.br

## Referencias

Eu ainda tenho que estudar um pouco mais quais os metodos que possibilitam calcular a correlacao espacial. O Ale deu a dica de estudar o metodo de Mantel, que me pareceu muito semelhante ao que eu fiz na funcao, mas preciso entender o metodo mais detalhadamente.

## Veja tambem

cor.test()

## Exemplos

# Criando um dataframe com dados para os exemplos:

```
ex.1 ← data.frame(Localidade=c("A", "B", "C", "D", "E"), x=c(1:5), y=c(2,9,3,12,5),  
Riqueza=c(101,110,149,121,130))
```

# Funcao com matrizes de distancia (no console), todos os graficos e correlacao de pearson (no dispositivo grafico) # Para ver os graficos maiores, aumente a janela do dispositivo grafico e rode a funcao novamente:

```
cor.esp (ex.1, permut=200)
```

# So com os graficos de entrada (maiores):

```
cor.esp (ex.1, plot.entr=TRUE, plot.hist=FALSE, metodo="pearson", permut=200)
```

# So com o histograma de saida (maior):

```
cor.esp (ex.1, plot.entr=FALSE, plot.hist=TRUE, metodo="pearson", permut=200)
```

# Funcao com correlacao de spearman e 10000 permutacoes (leva alguns segundos para rodar:

```
cor.esp (ex.1, plot.entr=FALSE, plot.hist=TRUE, metodo="spearman")
```

---

# Código da Função

---

#ESTRUTURA DA FUNCAO

```
cor.esp ← function(tabela, plot.entr=TRUE, plot.hist=TRUE, metodo="pearson", permut=10000)
```

```
{  
  
nomes←colnames(tabela)  
  
# MONTANDO AS MATRIZES DE DISTANCIA  
#####  
  
# Matriz de distancias entre coordenadas X das localidades  
  
dist.x ← dist(tabela[,2], method="manhattan",diag=TRUE, upper=FALSE)  
  
dist.x  
  
# Matriz de distancias entre coordenadas Y das localidades  
  
dist.y ← dist(tabela[,3], method="manhattan",diag=TRUE, upper=FALSE)  
  
dist.y  
  
# Calculando a matriz de distancia euclidiana entre as localidades ( $D^2 = X^2 + Y^2$ )  
  
euclid ← (dist.x^2 + dist.y^2)^0.5  
  
dist.xy ← as.matrix(euclid)  
  
dimnames(dist.xy) ← list(tabela[,1],tabela[,1])  
  
dist.xy  
  
# Calculando a matriz de distancia absoluta da variavel de interesse  
  
dist.var ← dist(tabela[,4], method="manhattan",diag=TRUE, upper=FALSE)  
  
dist.var ←as.matrix(dist.var)  
  
dimnames(dist.var) ← list(tabela[,1],tabela[,1])  
  
dist.var  
  
# CALCULANDO A AUTOCORRELACAO ESPACIAL #####  
  
# VETOR DE ORDENAMENTO DAS DISTANCIAS EUCLIDIANAS XY  
  
# Matriz de distancias entre coordenadas X das localidades (sem zeros)  
  
dist.x1 ← dist(tabela[,2], method="manhattan",diag=FALSE, upper=FALSE)  
  
dist.x1  
  
# # Matriz de distancias entre coordenadas Y das localidades (sem zeros)  
  
dist.y1 ← dist(tabela[,3], method="manhattan",diag=FALSE, upper=FALSE)  
  
dist.y1
```

```
# Calculando a matriz de distancia euclidiana entre as localidades ( $D^2 = X^2 + Y^2$ )
```

```
dist.xy1 ← (dist.x1^2 + dist.y1^2)^0.5
```

```
dist.xy1
```

```
# Extraindo o vetor de ordenacao das distancias euclidianas
```

```
vet.xy ← as.vector(dist.xy1)
```

```
vet.xy
```

```
length(vet.xy)
```

```
# VETOR DE ORDENAMENTO DAS DISTANCIAS COM RELACAO A VARIABEL DE INTERESSE
```

```
# Calculando a matriz de distancia absoluta da variavel de interesse
```

```
dist.var1 ← dist(tabela[,4], method="manhattan",diag=FALSE, upper=FALSE)
```

```
vet.var ← as.vector (dist.var1)
```

```
vet.var
```

```
length(vet.var)
```

```
# #
```

```
if (metodo=="spearman")
```

```
{
```

```
# CALCULO DO COEFICIENTE DE CORRELACAO DE SPEARMAN (rho)
```

```
spear←cor.test(dist.var1, dist.xy1, method="spearman")
```

```
spear$estimate
```

```
rho.spear←spear$estimate
```

```
rho.spear
```

```
# ALEATORIZACAO PARA O CALCULO DO p DE SPEARMAN
```

```
rho←spear$estimate
```

```
rho
```

```
for(i in 2:permut)
```

```
{
```

```
rand.var1 ← sample(dist.var1)
```

```
spear←cor.test(rand.var1, dist.xy1, method="spearman")

rho[i]←spear$estimate

}

if(rho.spear>=0)

{

n←rho[rho>=rho.spear]

n

length(n)

p.spear ← length(n)/permut

p.spear

}

else

{

n←rho[rho< =rho.spear]

n

length(n)

p.spear ← length(n)/permut

p.spear

}

}

# #

if (metodo=="pearson")

{

# CALCULO DO COEFICIENTE DE CORRELACAO DE PEARSON (R)

pear←cor.test(dist.var1, dist.xy1, method="pearson")

pear$estimate

r.pear← pear$estimate

r.pear
```

## # ALEATORIZACAO PARA O CALCULO DO p DE PEARSON

```
r←pear$estimate
```

```
r
```

```
for(i in 2:permut)
```

```
{
```

```
rand.var1 ← sample(dist.var1)
```

```
pear←cor.test(rand.var1, dist.xy1, method="pearson")
```

```
r[i]←pear$estimate
```

```
}
```

```
if(r.pear>=0)
```

```
{
```

```
n←r[r>=r.pear]
```

```
n
```

```
length(n)
```

```
p.pear← length(n)/permut
```

```
p.pear
```

```
}
```

```
else
```

```
{
```

```
n←r[r< =r.pear]
```

```
n
```

```
length(n)
```

```
p.pear ← length(n)/permut
```

```
p.pear
```

```
}
```

```
}
```

```
#####  
#####
```



## ## CENARIO NULO

# As diferencas entre as variaveis de interesse sao aleatoriamente distribuidas pelas

# localidades

## # # CENARIO ALTERNATIVO

# As diferencas entre as variaveis de interesse estao correlacionadas com as distancias euclidianas

# entre as localidades

# # SE  $p < 0.05$ , REJEITA-SE O CENARIO NULO

# Quanto maior o rho, maior a autocorrelacao espacial da variavel de

# interesse

#####  
#####

#

## # GRAFICOS

#####

#

if (plot.entr==TRUE)

{

# SEPARANDO A VARIABEL DE INTERESSE EM QUARTIS #####

quartis ← quantile (tabela[,4])

quartis

str (quartis)

q25 ← quartis [2]

q50 ← quartis [3]

q75 ← quartis [4]

q25

q50

q75

pt.q1 ← tabela[tabela[,4] ≤ q25, ]

pt.q2 ← tabela[tabela[,4]>q25 & tabela[,4]≤q50, ]

```
pt.q3 ← tabela[tabela[,4]>q50 & tabela[,4]≤q75, ]

pt.q4 ← tabela[tabela[,4]>q75, ]

pt.q1

pt.q2

pt.q3

pt.q4

#

# APRESENTANDO O GRAFICO DE COORDENADAS XY COM A VARIABEL DE INTERESSE EM QUARTIS

x11

if (plot.hist==TRUE)

{

par(mfrow=c(2,2))

}

else

{

par(mfrow=c(1,2))

}

plot (tabela[,2], tabela[,3], bty="l", ylim=c(min(tabela[,3]),max(tabela[,3])+50), xlab=nomes[2],
ylab=nomes[3], pch=21, main=paste("Localizacao x ", nomes[4]), cex.main=1 )

points (pt.q1[,2], pt.q1[,3], col="red1", bg="red1", pch=21, cex=1.2)

points (pt.q2[,2], pt.q2[,3], col="orange1", bg="orange1", pch=22)

points (pt.q3[,2], pt.q3[,3], col="green1" , bg="green3" , pch=23)

points (pt.q4[,2], pt.q4[,3], col="blue1", bg="blue1", pch=24)

legend (cex=0.8,(min(tabela[,2])),(max(tabela[,3])+50), bty="n" ,title=paste(nomes[4], "
quartil"),legend=c("0-25%","25-50%", "50-75%", "75-100%"),ncol=2, pch=c(21,22,23,24),
pt.bg=c("red1","orange1","green3","blue1"))

#

# APRESENTANDO GRAFICO DE DISTANCIAS EUCLIDIANAS x DIFERENCA DA VARIABEL DE INTERESSE
ENTRE AS LOCALIDADES
```

```
plot (vet.xy, vet.var, bty="l", xlab="Distancia Euclidiana", ylab=paste("Diferenca: ",nomes[4],
names(vet.var)), main="Correlacao Espacial", cex.main=1)

abline(lm(vet.var~vet.xy), col="red1")

teste←lm(vet.var~vet.xy)

}

# APRESENTANDO O GRAFICO DO HISTOGRAMA DOS COEFICIENTES DE CORRELACAO RESULTANTES
DAS PERMUTACOES

if (plot.hist==TRUE & metodo=="spearman")

{

# Histograma Spearman

if (plot.entri!=TRUE)

{

par(mfrow=c(1,1))

}

hist (rho, ylab="Frequencia", xlab="rho (Spearman)", main=paste("Histograma de rho\n", "rho= ",
round(rho.spear, digits=3), "\t p= ", round(p.spear, digits=3)), cex.main=1)

abline (v=rho.spear, col="red1")

}

#

if (plot.hist==TRUE & metodo=="pearson")

{

# Histograma Pearson

if (plot.entri!=TRUE)

{

par(mfrow=c(1,1))

}

hist (r, ylab="Frequencia", xlab="R (Pearson)", main=paste("Histograma de R\n", "R= ",
round(r.pear, digits=3), "\t p= ", round(p.pear, digits=3)), cex.main=1)

abline (v=r.pear, col="red1")

}
```

```
return(list("Matriz de distancias euclidianas:", dist.xy, paste("Matriz de distancia -", nomes[4], ":"),  
dist.var))
```

```
}
```

```
#
```

```
#
```

```
##### f i m #####
```

---

## Arquivos da Função (código e help)

---

[help\\_funcao\\_cor\\_esp.txt](#) [cor\\_esp.r](#)

From:

<http://labtrop.ib.usp.br/> - Laboratório de Ecologia de Florestas Tropicais

Permanent link:

[http://labtrop.ib.usp.br/doku.php?id=cursos:ecor:05\\_curso\\_antigo:r2011:alunos:trabalho\\_final:alexandre:start](http://labtrop.ib.usp.br/doku.php?id=cursos:ecor:05_curso_antigo:r2011:alunos:trabalho_final:alexandre:start) 

Last update: **2020/07/27 18:48**