

# Leandro Tavares Vieira



Leandro Tavares Vieira is a graduate student in the Ecology Graduate Program of the University of Campinas, Brazil, and has research interest in plant community ecology, currently investigating geographic patterns of woody species of northeastern Brazilian Cerrado.

## Meus Exercícios

Exercício 01 [LeandroVieira\\_Ex01\\_F](#)

Exercício 02 [LeandroVieira\\_Ex02\\_F](#)

Exercício 03 [LeandroVieira\\_Ex03\\_F](#)

Exercício 04 [LeandroVieira\\_Ex04\\_F](#)

Exercício 05 [LeandroVieira\\_Ex05\\_F](#)

Exercício 06 [LeandroVieira\\_Ex06\\_F](#)

Exercício 07 parcial [LeandroVieira\\_Ex07\\_P](#)

Exercício 08 [LeandroVieira\\_Ex08\\_F](#)

Exercício 09 [LeandroVieira\\_Ex09\\_F](#)

## Proposta de Trabalho Final

### Principal

#### Análise Espacial de Superfícies

Amostras contêm valores representativos do fenômeno estudado <sup>1)</sup>, usualmente obtidos de levantamento de campo que apresentam consistência de metodologia e unidade. Esses dados levantados estão disponíveis na forma de amostras pontuais e para utilizá-los de maneira efetiva necessitamos de um procedimento de interpolação para gerar sua representação no espaço onde não há informação. Para gerar superfícies que aproximem o fenômeno estudo de forma realista, é necessário modelar sua variabilidade espacial (Camargo et al. 2004).

Os modelos que objetivam criar superfícies com base em procedimentos de interpolação podem ter três grandes abordagens: Modelos determinísticos de efeitos locais, de efeitos globais e de efeitos globais e locais (Krigagem) (Camargo et al. 2004).

A idéia inicial da proposta é criar uma função ou um conjunto de funções que contemplem estas três

abordagens. Porém isso pode ser uma tarefa extremamente difícil dando margem para a execução de apenas uma das abordagens e sem a construção de gráficos de superfície.

Camargo, E. C. G, Druck, S. & Câmara, G. (2004). Análise Espacial de Superfícies. in: Análise Espacial de Dados Geográficos. Embrapa Cerrados, Planaltina, DF.

## Plano B

Criar uma função que calcule todos os parâmetros fitossociológicos (frequência, abundância, área basal, etc.) realizados com o método de parcelas. Os parâmetros poderiam sair por espécie, por família e por parcela.

## Comentário

A primeira proposta parece mais sedutora, pois pode prover instrumento interessante para análise espacial. Entretanto, fica difícil avaliá-la porque faltam elementos: 1. qual a informação de entrada? ; 2. qual o objeto de saída?; 3. Quais cálculos serão efetuados no processo? Fica difícil avaliar a viabilidade da proposta sem esses elementos. De qq forma é necessário que tenha algum treinamento prévio nos métodos de interpolação. Aqui o problema é que antes de construir uma função precisamos definir exatamente qual o resultado que será produzido e estrutura dos dado de entrada, ou seja qual o procedimento de manipulação dos dados. Nesse sentido, o plano B, apesar de mais simples em sua apresentação, parece mais concreto pois deixa claro quais parâmetros que serão calculados. Defina bem aqui como será a estrutura de entrada de dados.

## Contexto

A avaliação da primeira proposta realmente ficou difícil por falta de elementos, e era exatamente esse elementos que eu estava buscando. Quando escrevi a proposta não sabia de fato como era o procedimento das análises espaciais, só sabia que a entrada era uma data.frame X,Y, e Z, cujo resultado era um gráfico de superfície com valores interpolados, o que eu conseguia com softwares como o Geostatistics Sciences Plus. Foi uma proposta audaciosa pois não tenho o treinamento prévio dos métodos de interpolação, bem observado pelos professores, mas insisti nesta proposta pois o plano B era desafiador apenas na linguagem R pois os cálculos dos parâmetros fitossociológicos não são enigmas nenhum. Como a beleza do R é ter uma melhor compreensão do que estamos fazendo, e não ficar preso aos pacotes estatísticos, me mantive no plano A.

O resultado final foi realmente a compreensão de todas as etapas para a construção de um gráfico de superfícies por krigagem por exemplo, porém, empaquei justamente na interpolação, e dado o espaço de tempo, me ative na primeira etapa. O que eu fiz então foi um programa que faz um Semivariograma para qualquer tipo de dado espacial cuja entrada seja por X, Y, Z. A próxima etapa seria ajustar o semivariograma à modelos teóricos como gaussiano, exponencial, esférico etc, e de acordo com o modelo ajustado, interpolar para áreas não amostradas. Meu maior problema foi entender a matemática de estimar os valores de acordo com os modelos que achei um pouco complicado, pelo menos pelos livros que eu tenho, e nem tanto para transformar para a linguagem R. Bom, é isso, não consegui fazer tudo o que eu queria, mas atingi meus objetivos de compreender a análise espacial e “falar” a linguagem R. Muito obrigado por tudo.

## Página de Ajuda

semivariance{}  
R Documentation

package:nenhum

### Semivariância para dados espaciais

#### Description:

Retorna um gráfico da Semivariância em função das Classes de Distâncias. A função também

retorna os limites superiores das classes de distância, o número de pares de observações para cada classe e os valores de semivariância.

#### Usage:

```
semivariance (x, nclasse=0, equal.obs=F, n.obs=0, width.class=0)
```

#### Arguments:

**x:** data.frame ou Matriz. Valores das coordenadas espaciais e a amostra, em ordem X, Y, e Z

**nclasse** numérico. Determina o número de classes de distância. No padrão "0" fica determinado pelo produto da raiz quadrada do número de observações de Z.

**equal.obs** lógico. Se as classes de distância conterão iguais números de pares de observações.

**n.obs** numérico. Determina o número de pares de observações por classe de distância. No padrão "0", o tamanho é definido de acordo com os argumentos anteriores.

**width.class** numérico. Determina o tamanho das classes de distância. No padrão "0", o tamanho é definido de acordo com os argumentos anteriores.

#### Details:

O cálculo da semivariância para análise de estrutura espacial é utilizado para dados ambientais que não apresentam distribuição normal, pois o cálculo é baseado nos desvios entre

as sucessivas classes de distância. Outra característica importante é a não necessidade de média e variância constante. Os valores de semivariância são plotados contra classes de distância ou “lag”, o que é chamado de Semivariograma. As classes de distância pode ser construída de acordo com o pesquisador tanto em número de classes, número de pares de observações por classe e o tamanho (distância) das classes. Quando selecionado o tamanho das classes, não é possível determinar o número de classes nem o número de pares de observações para cada classe.

O semivariograma tem duas grandes aplicações, reconhecimento de estrutura espacial e interpolação ótima.

No reconhecimento de estrutura espacial, três características do semivariograma são importantes. A semivariância cresce gradualmente até atingir um ponto, chamado de “amplitude” ou “alcance”. O alcance é a distância onde as observações são espacialmente dependentes. O “patamar” é o valor da semivariância correspondente ao alcance, onde as observações, teoricamente, possuem variância constante portanto espacialmente independentes. E o “efeito pepita” que é o valor da semivariância para a distância zero e representa a variabilidade ao acaso.

Para interpolações, o semivariograma precisa ser ajustado de acordo com modelos teóricos como o modelo esférico, modelo linear, modelo exponencial, e modelo gaussiano. De acordo com o modelo, valores para locais não amostrados podem ser estimados.

Value:

semivariance retorna uma lista contendo:

os limites superiores das classes de distância,

o número de pares de observações por classe de distância,  
a semivariância por classe de distância,

um gráfico, chamado de semivariograma.

Note:

O objeto de entrada tem que estar na ordem X, Y, Z.

Quando definido o argumento "n.obs", os outros argumentos são invalidados.

Quando definido o argumento "width.class", os outros argumentos são invalidados.

Warning:

O semivariograma é definido de acordo com as classes de distância. Cabe ao pesquisador definir como será esta classe de distância.

#### Author(s):

Leandro Tavares Vieira  
(vieiralt@yahoo.com.br)

#### References:

Jongman, R.H.G, ter Braak, C.J.F & van Tongeren, O.F.R. (1995). Data Analysis in Community and Landscape Ecology. Cambridge University Press.  
Camargo, E. C. G, Druck, S. & Câmara, G. (2004). Análise Espacial de Superfícies. in: Análise Espacial de Dados Geográficos. Embrapa Cerrados, Planaltina, DF.

#### See Also:

Os pacotes spatstat e spatial.

#### Examples:

Ex.1:

X <-

```
c(200,348,200,232,233,209,178,331,323,168,302,212,256,122,214,317,305,72,307,149,196,286,331,181,192,219,95,306,251,95,173,142,326,44,47,332,264,67,101,65,85,132,87,232,158,180,271,186,273,84)
```

Y <-

```
c(123,100,284,208,160,238,36,86,241,244,259,138,128,82,273,161,200,158,230,213,88,190,100,152,103,199,198,119,169,134,65,82,178,213,228,73,289,235,114,94,223,204,168,275,38,135,258,168,272,124)
```

Z <-

```
c(102.01,4.04,1108.22,662.03,82.45,898.80,0.00,0.00,669.77,942.49,560.27,53.29,5.29,18.49,1036.20,164.87,340.03,277.22,481.36,617.52,9.92,226.20,4.62,344.10,46.92,572.64,376.36,23.33,107.12,100.00,3.06,51.84,283.59,451.56,475.24,0.03,594.87,571.21,33.64,20.25,552.25,333.06,257.60,881.50,0.30,226.50,658.95,464.40,494.62,108.16)
```

```
data.frame(X,Y,Z) -> slope  
semivariance(slope, width.class=50)  
semivariance(slope, 15, T)  
semivariance(slope, 18)  
semivariance(slope, n.obs=800)
```

## Ex.2

```
rep(seq(10,100, by=10), each=10) -> X
rep(seq(10,100, by=10), time=10) -> Y
c(rep(1:8, each=12), 9,9,9,9) -> Z
data.frame(X,Y,Z) -> ha
plot(Y~X, data=ha, pch=0, cex=5, main="Um Hectare")

unique(X) -> x; unique(Y) -> y
matrix(NA, ncol=x, nrow=y) -> mat
colnames(mat) <- x ; rownames(mat) <- y
Z -> mat[]
library(lattice)
image(mat)
wireframe(mat, shade=TRUE, aspect=c(40/65,0.4), light.source = c(10,0,10))
semivariance(ha)
semivariance(ha, 13, T)
semivariance(ha, n.obs=350)
semivariance(ha, width.class=25)
```

## Código da Função

```
semivariance <- function(x, nclasse=0, equal.obs=F, n.obs=0, width.class=0)
{
  colnames(x) <- c("X", "Y", "Z")

  ##### Criando objetos base

  x[,c(1,2)] ->xy
  dist(xy, diag=F, upper=F) ->dxy
  dist(x[,3]) -> zdist # isso é a diferença dos valores de Z para todos os
pares de pontos
  diff(range(dxy)) -> ampli
  zvalue <- list()
  as.numeric(zdist) -> vetzdist
  as.numeric(dxy) -> vetdxy
  data.frame(vetzdist,vetdxy) -> datadists
  order(datadists$vetdxy) -> orddxy
  orddxy -> datadists[,3]
  datadists[orddxy,] -> matord
  ##### Definindo as classes
  if(nclasse==0)
  {
    round(sqrt(length(x[,1]))) -> numclass # Definido o número de classes
baseado na raiz do número de
    observações. Isso pode ser alterado de acordo com o pesquisador
  }
  else
```

```

{
  numclass <- nclasse
}
if(equal.obs==F)
{
  1 + numclass -> numclasse
  (ampli/numclass) -> sizeclass
  classes <- rep(NA, numclasse)
  names(classes) <- paste("classe", 1:numclasse)
  for(i in 2:numclasse)
  {
    dados <- (min(dxy)+ sizeclass*(i-1))
    classes[i] <- dados
  }
  classes[1] <- 0
  round(classes, digits=2) -> classes
  tamanho <- rep(0, numclass)
  for(i in 1:(length(classes)-1))
  {
    which(dxy<=classes[i+1] & dxy> classes[i]) -> conta
    sum(conta !=0) -> soma
    soma -> tamanho[i]
  }
  tamanho[1:length(classes)-1] -> tamanho
### Semivariância
for(i in 1:(length(classes)-1))
{
  zdist[which(dxy<=classes[(i+1)] & dxy>classes[i])] -> zvalue[[i]]
}
Semivariancia <- rep(NA, time=(length(classes)-1))
for(i in 1:length(Semivariancia))
{
  sum(zvalue[[i]]^2)/(2*(length(zvalue[[i]]))) -> Semivariancia[i]
}
}
else
{
  round(length(dxy)/numclass) -> pares
  matrix(sort(dxy), ncol=numclass, nrow=pares)-> dataclass
  ifelse((length(dataclass) - length(dxy))==0, dataclass ,
dataclass[(length(dxy)+1):
length(dataclass)] <- rep(NA, (length(dataclass) - length(dxy)))
  classes <- rep(NA, (numclass+1))
  names(classes) <- paste("classe", 1:(numclass+1))
  round(apply(dataclass, 2, max, na.rm=T),digits=2) ->
classes[2:(numclass+1)]
  classes[1] <- 0
  tamanho <- rep(pares, numclass)
  tamanho[numclass] <- pares - (length(dataclass) - length(dxy))

### Semivariância

```

```
matrix(matord$vetzdist, ncol=numclass, nrow=pares)-> dataequal
  ifelse((length(dataequal) - length(dxy))==0, dataequal ,
dataequal[(length(dxy)+1):
  length(dataclass)] <- rep(NA, (length(dataclass) - length(dxy)))
  if((length(dataequal) - length(dxy))!=0)
    {
      cat("\n\t0 número de pares de observações por classe de distância
não é múltiplo do número
      de pares de observações total.\n\tProblema corrigido. Repare na
última classe\n")
    }
  Semivariância <- rep(NA, time=(length(classes)-1))
  dataequal^2 -> dataequal2
  apply(dataequal2, 2, sum, na.rm=T) -> sumequal
  for(i in 1:numclass)
    {
      sumequal[i]/(2*(tamanho[i])) -> Semivariância[i]
    }
}
if(n.obs==0)
{
  #nada
}
else
{
  n.obs ->nobs
  matrix(sort(dxy), nrow=nobs)-> matobs
  ifelse((length(matobs) - length(dxy))==0, matobs ,
matobs[(length(dxy)+1):length(matobs)] <-
  rep(NA, (length(matobs) - length(dxy)))
  dim(matobs) -> dmatobs
  dmatobs[2] -> numclass
  classes <- rep(NA, (numclass+1))
  names(classes) <- paste("classe", 1:(numclass+1))
  round(apply(matobs, 2, max, na.rm=T),digits=2) ->
classes[2:(numclass+1)]
  classes[1] <- 0
  tamanho <- rep(nobs, numclass)
  tamanho[numclass] <- nobs - (length(matobs) - length(dxy))
  cat("\n\tSelecione o número de observações por classe de distância.
Argumentos nclass, equal.obs
  e width.class não são válidos.\n\n")
  if((length(matobs) - length(dxy))!=0)
    {
      cat("\n\t0 número de pares de observações por classe de distância
escolhido não é múltiplo do
      número de pares de observações total.\n\tProblema corrigido.
Repare na última classe\n")
    }
}
####Semivariância
```



```

matrix(matord$vetzdist, ncol=numclass, nrow=nobs)-> dataobs
ifelse((length(dataobs) - length(dxy))==0, dataobs ,
dataobs[(length(dxy)+1):length(dataobs)] <-
rep(NA, (length(dataobs) - length(dxy))))
Semivariância <- rep(NA, time=(length(classes)-1))
dataobs^2 -> dataobs2
apply(dataobs2, 2, sum, na.rm=T) -> sumobs
for(i in 1:numclass)
{
sumobs[i]/(2*(tamanho[i])) -> Semivariância[i]
}
}
if(width.class==0)
{
#nada
}
else
{
width.class -> sizeclass
ceiling(max(dxy)/sizeclass) -> numclass # É incompatível com o
argumento nclass e com equal.obs
1 + numclass -> numclasse
classes <- rep(NA, numclasse)
names(classes) <- paste("classe", 1:numclasse)
for(i in 2:numclasse)
{
dados <- (sizeclass*(i-1))
classes[i] <- dados
}
classes[1] <- 0
round(classes, digits=2) -> classes
tamanho <- rep(NA, numclass)
for(i in 1:length(classes))
{
which(dxy<=classes[i+1] & dxy> classes[i]) -> conta
sum(conta !=0) -> soma
soma -> tamanho[i]
}
tamanho[1:length(classes)-1] -> tamanho
cat("\n\tSelecione o tamanho das classes. Argumentos nclass,
equal.obs e n.obs não são válidos\n\n")
##### Semivariância
for(i in 1:(length(classes)-1))
{
zdist[which(dxy<classes[i+1] & dxy>=classes[i])] -> zvalue[[i]]
}
Semivariância <- rep(NA, time=(length(classes)-1))
for(i in 1:length(Semivariância))
{
sum(zvalue[[i]]^2)/(2*(length(zvalue[[i]]))) -> Semivariância[i]
}
}

```

```
    }  
#####  
    round(Semivariância, digits=2) -> Semivariância  
    round(classes[2:length(classes)], digits=2) -> DistanceClass  
    names(DistanceClass) <- paste("Classe", 1:(length(classes)-1))  
    plot(Semivariância~DistanceClass, pch=16, cex=1.3, col="red",  
main="Semivariograma", xlab="Classes  
de Distância", ylab="Semivariância",  
    cex.axis=1.1, font.lab=2, font.main=3, cex.lab=1.2, cex.main=1.5) ->  
Gráfico  
    for(i in 1:length(DistanceClass))  
    {  
        segments(DistanceClass[i],Semivariância[i],  
DistanceClass[i+1],Semivariância[i+1], col="red")  
    }  
    for(i in 1:length(DistanceClass))  
    {  
        segments(DistanceClass[i],-10, DistanceClass[i],Semivariância[i],  
col="lightgray", lty=5)  
    }  
    list(DistanceClass, tamanho, Semivariância, Gráfico) -> lista  
    names(lista) <- c("\n\tClasses de Distância", "\tNúmero de Pares de  
Observações por Classe de Distância",  
    "\tSemivariância", "\tGráfico")  
    return(lista)  
}
```

## Arquivo da Função

### Função da Semivariância

1)

Pouco informativo! O que significa “valores representativos”, “fenomeno estudos”: precisas ser mais preciso!

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:r2010:alunos:trabalho\\_final:vieiralt:start](http://labtrop.ib.usp.br/doku.php?id=cursos:ecor:05_curso_antigo:r2010:alunos:trabalho_final:vieiralt:start)

Last update: **2020/07/27 18:46**