

Escola Superior de Agricultura “Luiz de Queiroz”  
Departamento de Ciências Exatas – LCE

Disciplina: LCE5700 – Geoestatística  
Prof: Paulo Justiniano Ribeiro Junior

Relatório para conclusão de disciplina

**Simulação e Análise geoestatística**

Aluno: Gabriel Bruno de Lemos – n° USP: 3705082  
Mestrando em Estatística e Experimentação Agronômica

Novembro  
2011

## Sumário

Resumo.....	3
Objetivo.....	3
Introdução.....	3
Material e Métodos .....	3
Resultados e Discussão.....	8
Conclusões .....	18
Referências bibliográficas .....	19
Anexo – Comandos do programa R utilizados para realização das análises.....	20

## **Resumo**

Procurou-se realizar uma síntese dos conhecimentos adquiridos na disciplina LCE5700 – Geoestatística, ministrada pelo Prof. Paulo Justiniano durante o 2º semestre de 2011. Primeiramente foi feita uma revisão das etapas necessárias para a realização de uma análise geoestatística e, em seguida, duas análises utilizando dados simulados e dados reais. O atributo analisado na análise de dados reais foi a distribuição do teor de Cálcio em uma área cultivada com seringueira. A simulação de dados e/ou cenários esteve presente em praticamente todo o estudo e foi muito importante para a construção de mapas de probabilidade de ocorrência / ausência do objeto de interesse no estudo.

## **Objetivo**

Formalizar os conhecimentos adquiridos pelo autor na disciplina de geoestatística em um documento acessível a usuários desta técnica de análise.

## **Introdução**

A geoestatística é o ramo da estatística espacial onde os dados consistem de uma amostra finita de valores medidos relativos a um fenômeno espacial e de caráter contínuo (Diggle e Justiniano, 2007).

A análise geoestatística busca compreender como um determinado atributo está variando no espaço e, com isso, realizar inferências sobre sua presença em um determinado local. Uma ferramenta muito interessante e que está começando a ganhar espaço entre os pesquisadores da área é a utilização da simulação para gerar diversos cenários e, com isso, diminuir a incerteza na tomada de decisão.

A simulação estocástica tem como objetivo principal acessar a incerteza relacionada às estimativas dos atributos.

A simulação estocástica definida por Deutsch & Journel (1998) é o processo por meio do qual é possível gerar inúmeros cenários equiprováveis, imagens estocásticas, para uma variável aleatória que obedece a modelos de funções aleatórias. Assim, cada um dos cenários gerados deve reproduzir as características impostas pelo modelo de funções aleatórias.

Este estudo mostrará uma forma de se utilizar o processo de simulação de forma a aumentar a confiança na tomada de decisão de uma análise geoestatística.

## **Material e Métodos**

Para que o objetivo principal desta disciplina seja cumprido, será utilizado como pano de fundo duas situações distintas – simulação de dados e análise de dados reais (amostras de solo) da Fazenda Arvoredo, cedido pelo Grupo de Mecanização e Agricultura de Precisão - gMAP, orientados pelo Prof. José Paulo Molin, da Esalq / Usp. A área de onde as amostras de solo foram retiradas é cultivada com seringueira.

As análises, tanto reais, quanto simuladas, foram realizadas no software livre R, por meio do pacote geoR.

O processo de análise geoestatístico destas duas situações pode ser resumido nas seguintes etapas:

- Criar uma simulação de dados / Carregar dados no software R;
- Análise exploratória dos dados;
- Definição de possíveis modelos de covariância possam se ajustar a situação de análise;
- Estimação dos parâmetros dos modelos de covariância escolhidos;
- Comparação e escolha do melhor modelo;
- Plotar mapas de predição utilizando o modelo previamente selecionado;
- Gerar simulações condicionais de forma a “validar” os mapas de predição por meio da construção de intervalos de confiança para cada ponto estimado.

## 1. Criar uma simulação de dados / Carregar dados no software R;

### a. Simulando dados

A simulação de dados é uma ótima forma para testar o nível de precisão de uma estimativa, uma vez que conhecemos os parâmetros verdadeiros do modelo, algo que dificilmente ocorre em situações reais. Para realizarmos esta simulação utilizaremos a função “grf”, que acompanha o pacote geoR, utilizando os seguintes comandos:

```
grf(n, grid = "irreg", nx, ny, xlims = c(0, 1), ylims = c(0, 1), borders, nsim = 1,  
cov.model = "matern", cov.pars = (sigma2, phi), kappa = 0.5, nugget = 0,  
lambda = 1, aniso.pars, mean = 0, method, RF=TRUE, messages)
```

Nos campos onde o valor já está colocado – como grid="irreg" – significam o padrão da função e será executada caso não seja alterado ou mesmo citado na função. Segue um exemplo:

```
require(geoR)  
set.seed(171) # Fixar semente da simulação de forma a poder reproduzir os mesmos resultados  
em uma nova análise.  
dat <- grf(100, cov.pars = c(20, .25), nug=4, mean=40)
```

O comando acima está criando um banco de dados chamado “dat” com 100 valores sorteados. Os parâmetros adotados pelo modelo de covariância entre os dados foram  $\sigma^2 = 20$  e  $\phi = .25$ . O efeito pepita (nug) foi adotado como sendo 4 e a média geral do processo é 40.

Perguntas:

- Qual o modelo de covariância adotado?

Res: O modelo utilizado foi “matern”, que é o padrão (default) da função “grf”.

- Como os valores estarão dispostos na “área” simulada?

Res: Os valores estarão dispostos em um grid (x,y) com dimensões 1x1 e de forma irregular (aleatória). Verificar para isso  $xlims = c(0, 1)$ ,  $ylims = c(0, 1)$  e  $grid = "irreg"$

- Os dados são iso ou anisotrópicos, ou seja, existe diferença na covariância entre os dados de acordo com a direção que analiso?

Res: Os dados são isotrópicos, pois o comando "aniso.pars" não aparece na função.

- Qual método foi utilizado para se realizar a inversão de matrizes de dados, etapa necessária para a criação dos valores simulados?

Res: O método utilizado foi a decomposição de Cholesky, pois o número de pontos amostrados foi menor do que 500. Acima disso a função "grf" chama o pacote RandomFields automaticamente para fazer esta decomposição com um algoritmo mais eficiente.

Os parâmetros acima adotados podem ser alterados de acordo com a necessidade do usuário. Estas e outras informações podem ser obtidas por meio da ajuda no R → help(grf)

Para finalizar o processo deve-se transformar os dados simulados em geodata, de forma que o programa crie as coordenadas dos pontos amostrados com o respectivo valor do atributo desejado. Para tal utiliza-se o seguinte comando:

```
class(dat) <- "geodata"
```

Este comando transformou os dados armazenados em "dat" para uma estrutura de geodata.

## **b. Carregando dados a partir de uma planilha csv**

Imaginando um arquivo com 3 colunas (1,2,3), representando respectivamente: coordenada x, coordenada y e atributo a ser analisado. Nomeia-se o arquivo como "dados".

```
dados <- read.csv("dados.csv", sep=";", dec=",")  
M <- cbind(dados$1, dados$2) # Para criar a matriz de coordenadas  
Y <- dados$3 # Para criar o vetor com as medidas da variável a ser estudada  
dados <- list(coords = M, data = Y) # Cria uma lista com as 3 colunas  
class(dados) <- "geodata" # Para transformar em geodata
```

Caso não estejam disponível as bordas (matriz de coordenadas) que definem os limites da área de estudo, pode se fazê-lo manualmente por meio do comando "locator".

O Anexo conterà toda a rotina do R utilizada nas análises geoestatísticas deste relatório.

Desta forma, após simular ou carregar os dados no programa R, podem ser feitas as primeiras análises → Análise exploratória dos dados, também chamada de Estatística Descritiva.

## **2. Análise exploratória dos dados;**

Nesta etapa o pesquisador poderá verificar:

- Presença de pontos discrepantes;

Detectar erros de digitação ou leitura de dados de forma a diminuir as incertezas sobre a predição que será feita.

- Existência ou não de padrão espacial na distribuição do atributo na área estudada;

Verificar se existe a formação de clusters, que são regiões de agrupamento de pontos semelhantes. Caso não se observe nenhum padrão espacial logo nesta etapa, não faz muito sentido continuar as análises

- Normalidade dos dados e necessidade de transformação dos mesmos;
- Tendências na distribuição dos dados;

Verificar se existem covariáveis afetando os dados, ou seja, existe uma alteração da média dos valores do atributo ao longo da área – e em qual sentido? Pode ser que outras covariáveis estejam afetando, como altitude. Isso poderá ser verificado caso esta outra covariável tenha sido coletada.

- Distância mínima e máxima entre os pontos

Importante para a elaboração do variograma, caso opte por esta ferramenta para ajudar no processo de estimação de parâmetros do modelo a ser escolhido.

## **3. Definição de possíveis modelos de covariância que possam se ajustar a situação de análise;**

Diversos são os modelos que podem ser adotados, tais como cauchy, circular, esférico, exponencial, gaussiano, linear, matern, etc . Neste relatório as análises se concentrarão entre os modelos exponencial e matern.

## **4. Estimação dos parâmetros dos modelos de correlação escolhidos;**

A partir deste ponto começa a diferenciação entre os pacotes de análises prontos, ou de “prateleiras”, com a análise por meio do pacote geoR. Neste último será exigido do usuário conhecimento estatístico sobre estimação e inferência dos parâmetros que compõem o modelo, podendo utilizar os métodos dos mínimos quadrados ponderados, da máxima verossimilhança e máxima verossimilhança restrita. Nos programas mais simples o usuário não precisa se preocupar com a escolha destes métodos e normalmente se faz um ajuste “ad-hoc” – ou sem muito rigor científico. No pacote geoR também existe esta possibilidade por meio da função interativa “eyefit”.

## **5. Comparação e escolha do melhor modelo;**

Ao utilizarmos métodos estatísticos para a estimação de parâmetros, podemos escolher o modelo que melhor se ajustou aos dados com a ajuda de outro método numérico – que são os critérios AIC, BIC ou o pela função de maximização do log da verossimilhança. Uma possível utilização do método eyefit

## **6. Plotar mapas de predição utilizando o modelo previamente selecionado;**

Escolhido o modelo de covariâncias, estimado os parâmetros deste modelo, realiza-se a predição em toda a área desejada- ou seja – a partir de alguns pontos amostrados pretende-se inferir sobre o restante da área. A krigagem ordinária é um tipo de predição e será a escolhida para esta análise. Sua krigagem será tão boa quanto foi a escolha dos pontos amostrais, escolha do modelo e dos métodos de estimação dos parâmetros. Muitos pacotes geoestatísticos simplesmente pulam da estimação do variograma diretamente para a krigagem, negligenciando as etapas de definição dos modelos de covariância, estimação de parâmetros e escolha final do modelo a ser usado para a predição espacial. Estas, na verdade, são feitas de uma forma pré-programada e sem possibilidade de escolhas por parte do usuário final, que acaba tratando todo este processo como uma grande caixa preta.

Além do mapa de predição de valores pode-se também plotar o mapa de variância das estimativas, possibilitando enxergar os pontos com maior variabilidade e, conseqüentemente, maiores dúvidas a respeito do valor do atributo no local. Assim sendo, pode-se verificar a necessidade de mais pontos a serem amostrados em uma futura coleta de dados, ou até mesmo uma possível redução, caso a área seja bem homogênea e a variância das estimativas baixa.

## **7. Gerar simulações condicionais de forma a “validar” os mapas de predição por meio da construção de intervalos de confiança para cada ponto estimado.**

A predição espacial feita pela krigagem, embora válida, é de pouca utilidade prática, uma vez que não se sabe o intervalo de confiança das previsões realizadas por esta simulação, pois só foi gerado um processo. Esta última etapa – geração de simulações condicionais – visa resolver este problema, simulando vários mapas (cenários) da distribuição do atributo na área, respeitando os parâmetros e modelos previamente definidos – por isso o nome **CONDICIONAL**. Com isso obtém-se uma idéia da possível variabilidade do atributo em determinado ponto, construindo intervalos de confiança, mapas de mediana, mínimo e máximo, além de mapas de probabilidades.

## Resultados e Discussão

Primeiramente vamos aplicar os sete passos citados na etapa de Material e Métodos para dados simulados. Em seguida será feita a análise com os dados da Fazenda Arvoredo, cultivado com a cultura da seringueira.

A programação completa, feita em R, está disponível no Anexo.

### A. Simulação de dados

#### 1. Criar uma simulação de dados;

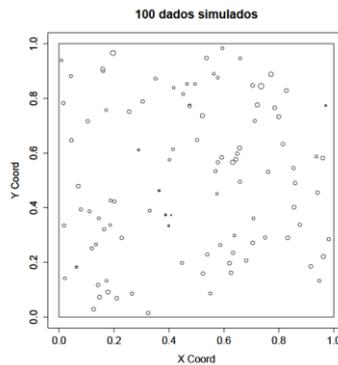


Figura 1: Simulação de 100 valores entre as coordenadas  $x(0,1)$  e  $y(0,1)$ .

Na Figura 1 estão plotados os 100 valores simulados. O atributo em questão foi definido como possuindo uma distribuição normal, com média = 10, efeito pepita = 4, variância = 3 e parâmetro de decaimento do função de correlação ( $\phi$ ) = 0.25.

#### 2. Análise exploratória dos dados;

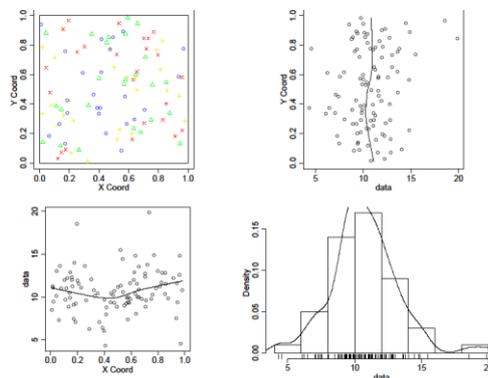


Figura 2: Análise exploratória dos dados simulados

Na Figura 2 observa-se uma pequena concentração de atributos “azul” no lado esquerdo do gráfico 1, indicando um possível padrão espacial presente na distribuição dos dados. Os gráficos 2 e 3 mostram uma leve alteração nos valores médios dos atributos a medida que se desloca entre os eixos x e y, indicando duas possíveis covariáveis que interferem na distribuição do atributo. O último gráfico mostra que

nossa variável de interesse possui (aparentemente) distribuição normal, com média aproximadamente de 11.

Vale lembrar que estes dados foram gerados a partir de uma distribuição NORMAL, com média = 10, variância = 3, etc., conforme citado no item 1. Portanto deve-se tomar muito cuidado com as primeiras impressões retiradas da amostra.

O comando “summary()” solta as seguintes informações:

```
Number of data points: 100
Número de pontos “amostrados” na area.
Coordinates summary
      x          y
min 0.009060411 0.01353962
max 0.982308927 0.98321011
```

Valores mínimos e máximos de coordenadas x e y.

```
Distance summary
      min      max
0.002806482 1.237385020
```

Distância mínima e máxima entre os pontos. Utilizaremos a distância máxima para montar o variograma e estimar os parâmetros do modelo.

```
Borders summary
      Var1 Var2
min  0    0
max  1    1
```

```
Data summary
      Min. 1st Qu. Median Mean 3rd Qu. Max.
4.332  9.387 10.530 10.720 12.010 19.880
```

Informações importantes para a construção de mapas de probabilidade do atributo na área, objetivo final de nossa análise.

### **3. Definição de possíveis modelos de covariância possam se ajustar a situação de análise;**

Diversos são os modelos de covariância disponíveis para se ajustar aos dados. Neste momento a função `eyefit()` mostra-se muito útil, fornecendo estimativas iniciais do modelo e de seus respectivos parâmetros a serem ajustados posteriormente.

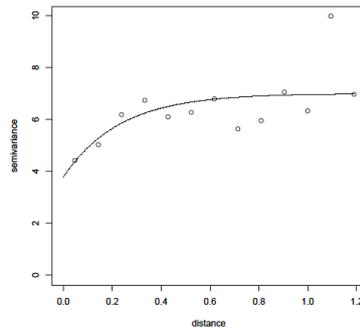


Figura 3: Variograma estimado pela função eyefit()

Na Figura 3 apresenta-se o variograma estimado pela função eyefit. O modelo de covariância escolhido foi o exponencial, com os seguintes parâmetros:

```
cov.model      sigmasq phi tausq kappa kappa2 practicalRange
exponential 3.23  0.23 3.77 <NA> <NA>      0.69
```

Em muitos programas de análise geoestatística estes parâmetros seriam usados diretamente para a extrapolação dos valores em toda a área por meio do método da krigagem. Em nossa análise iremos utilizar este variograma empírico apenas como “chutes” iniciais para a estimação dos parâmetros pelo método da mássima verossimilhança, realizada na próxima etapa. Observe que os valores estimados dos parâmetros já estão próximos aos valores reais – sigmasq =3, phi = 0.25, tausq = 4

#### 4. Estimação dos parâmetros dos modelos de covariância escolhidos;

Propõe-se então 2 modelos:

- ml1 → Exponencial, sem tendência;

- ml2 → Exponencial, com tendência de primeira ordem – ou seja – as coordenadas x e y podem estar influenciando a variação do atributo na área de uma forma linear.

#### 5. Comparação e escolha do melhor modelo;

Comparação entre os dois modelos previamente selecionados:

> ml1

likfit: estimated model parameters:

```
beta tausq sigmasq phi
"10.6884" "2.5484" "3.7046" "0.0868"
```

Practical Range with cor=0.05 for asymptotic range: 0.2600427

likfit: maximised log-likelihood = -227

> ml2

likfit: estimated model parameters:

```
beta0 beta1 beta2 tausq sigmasq phi
"9.8856" "0.0693" "1.4806" "2.2083" "3.8070" "0.0660"
```

Practical Range with cor=0.05 for asymptotic range: 0.1978294

likfit: maximised log-likelihood = -226.5

O modelo 2 (ml2) deve ser o escolhido pois retornou o maior valor para a maximização do log da verossimilhança (-226.5 > 227).

## 6. Plotar mapas de predição utilizando o modelo previamente selecionado;

Com o modelo ml2 escolhido, realiza-se a predição do atributo em toda a área de interesse. Em nossa análise a maneira de predição escolhida será krigagem ordinária.

Para delimitar a área de interesse deve-se plotar uma malha de pontos (regular ou irregular) na área de estudo e pedir ao programa que calcule o valor do atributo em cada ponto da malha, criando assim o mapa de predição.

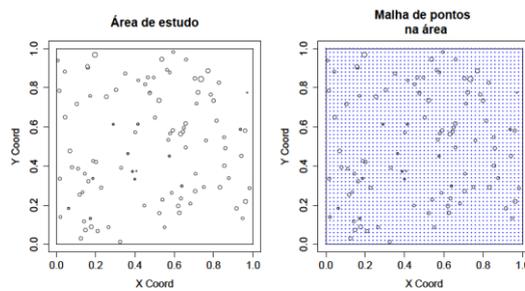


Figura 4: Criação do grid de predição

Na Figura 4 observa-se o grid de predição plotado sobre a área de estudo. Em seguida realiza-se a krigagem.

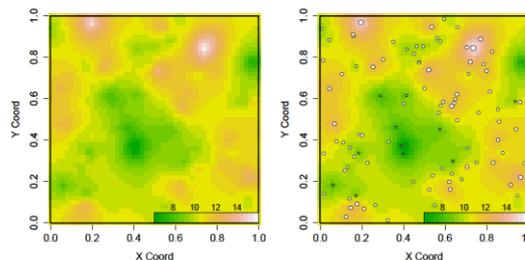


Figura 5: Predição do atributo de interesse na área

Na Figura 5 observa-se o mapa de predição feito por krigagem. O segundo mapa é o mesmo do primeiro, apenas com a presença dos pontos “amostrados” inicialmente.

Outro gráfico de interesse é o de variância de krigagem. Este gráfico permite “enxergar” possíveis pontos com alta variância de predição, o que significa regiões de baixa confiabilidade sobre a predição. Nestes pontos, em amostragens futuras, seria interessante reforçar a amostragem, de forma a diminuir as incertezas nas predições futuras.

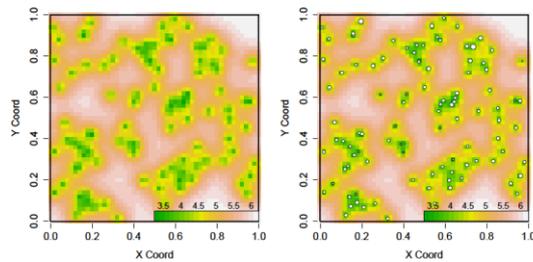


Figura 6: Mapas de variância de krigagem

### 7. Gerar simulações condicionais de forma a “validar” os mapas de predição por meio da construção de intervalos de confiança para cada ponto estimado.

Para finalizar a análise geoestatística, pode-se criar os mapas de probabilidade. O objetivo destes mapas é dividir a área em regiões mais ou menos prováveis de se encontrar valores abaixo de um determinado valor de corte. Em nosso exemplo poderíamos fixar como interesse toda a região com presença do atributo em estudo com valores abaixo de 9 como sendo o nosso alvo para possível intervenção ou correção.

Primeiramente realizam-se simulações condicionais ao modelo e parâmetros previamente definidos e, desta forma, constrói-se diversos cenários prováveis de ocorrerem, de acordo com o modelo usado nas análises anteriores. Para este exemplo realizaremos 1.000 simulações.

O grid de predição definido na etapa anterior era composto por 51 colunas com 51 pontos em cada coluna. Isso resulta em um total de 2.601 (51 x 51) pontos a serem estimados. Como realizamos 1.000 simulações, teremos uma matriz resultante com 2.601 linhas x 1.000 colunas. Cada linha desta matriz representa um ponto. Cada coluna, por sua vez, uma simulação (ou um mapa).

Para gerar os mapas de probabilidades usaremos o conceito de frequência de ocorrência de um evento para estimarmos a sua chance de ocorrência. Desta forma, ao simularmos 1.000 cenários diferentes, aplicamos uma função e pedimos para calcular o número de vezes que o valor estimado em determinado ponto ficou abaixo de nosso valor de interesse – 9 neste exemplo. Assim sendo, as regiões que mais vezes tiveram predições com valores abaixo de 9 estarão pintadas com as cores vermelho e amarelo.

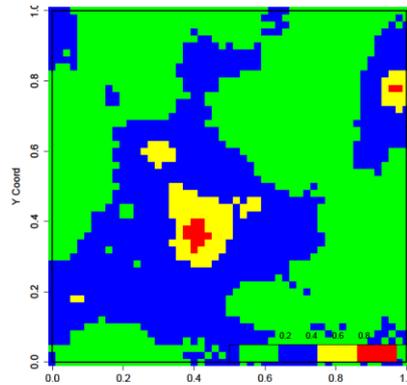


Figura 7: Mapa de probabilidade de ocorrência de valores abaixo de 9

A região em vermelho, na Figura 7, indica que a chance de ocorrência de um valor abaixo de 9 nestes pontos é igual ou maior do que 80% - ou seja – nas 1.000 simulações realizadas, pelo menos em 800 (ou mais!) o atributo nestes pontos ficou abaixo de 9. Embora na krigagem também possa ser elaborado um mapa semelhante, a grande diferença da simulação condicional é que estes resultados são estimativas baseadas em 1.000 cenários diferentes – e não apenas 1, como é o caso da krigagem. Desta forma, tem-se maior segurança sobre as áreas prováveis com deficiência (ou excesso) do atributo em estudo, do que se analisado apenas uma situação. Esta é a forma para se construir intervalos de confiança para a predição de valores em um mapa.

## B. Análise geoestatística das amostras de solo da Fazenda Arvoredo

Agora seguiremos para a análise geoestatística com dados reais. Estaremos interessados na variável Cálcio (Ca) em uma área cultivada com Seringueira. Suponhamos que valores abaixo de 13 unidades sejam prejudiciais para a cultura e precisará de uma intervenção com adubação. Queremos, portanto, mapear as possíveis áreas que precisarão receber essa adubação.

### 1. Carregar dados no software R;

Ao carregar um banco de dados no R devem-se realizar as alterações necessárias de forma que o programa entenda o que está sendo lido. A primeira etapa em nosso exemplo é transformar os dados brutos em dados da classe “geodata”. Outro ponto importante é a definição das bordas da área de estudo, caso não tenha sido fornecido pelos pesquisadores de campo. Existe uma função – locator() – apenas para ajudar nesta tarefa. A Figura 8 mostra um mapa com as bordas definidas por este comando. Após estes ajustes prossegue-se para a análise exploratória dos dados.

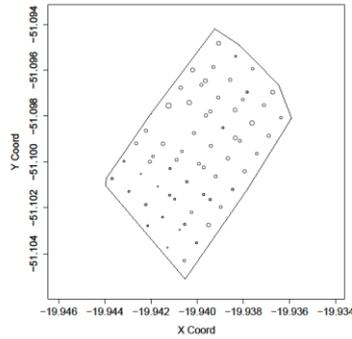


Figura 8: Disposição dos pontos amostrados na Fazenda Arvoredo

## 2. Análise exploratória dos dados;

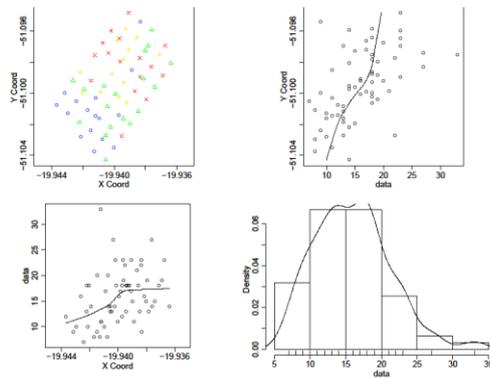


Figura 9: Análise exploratória da quantidade de Ca

Na Figura 9 observa-se uma tendência de aglomeração da cor azul no canto inferior esquerdo no primeiro mapa. No segundo e terceiro pode-se também inferir sobre uma possível tendência de afetando a quantidade de Ca à medida que nos deslocamos entre os eixos y e x. O último gráfico desta figura mostra a distribuição da quantidade de Ca que, embora não de forma escancarada, lembra uma distribuição normal.

*Number of data points: 63*

*Coordinates summary*

*LATITUDE LONGITUDE*

*min -19.94369 -51.10430*

*max -19.93637 -51.09482*

*Distance summary*

*min max*

*0.0002355814 0.0095980583*

*Data summary*

*Min. 1st Qu. Median Mean 3rd Qu. Max.*

*7.00 12.50 16.00 15.97 18.50 33.00*

Observar que o nosso valor de interesse na quantidade de Ca – 13 unidades – fica acima do primeiro quartil – ou seja – teremos pelo menos 25% dos valores amostrados abaixo de nosso ponto de corte. Espera-se que essa proporção de valores mantenha-se nas simulações condicionais.

### 3. Definição de possíveis modelos de covariância possam se ajustar a situação de análise;

Para começar a escolha de modelos de covariância, partimos para o variograma, onde teremos uma boa idéia inicial sobre o modelo e parâmetros a ser utilizado.

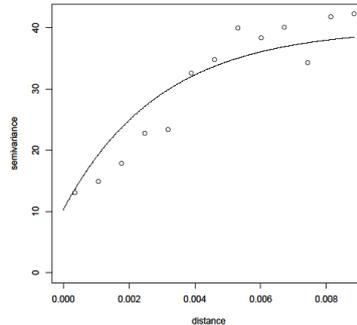


Figura 10: Variograma ajustado pelo eyefit()

```
cov.model    sigmasq phi tausq kappa kappa2 practicalRange
exponential  29.66  0.0029 10.27 <NA> <NA> 0.008803
```

Dois modelos foram previamente selecionados:

-ml1 → Exponencial, sem tendência;

-ml2 → Exponencial, com tendência de primeira ordem – ou seja – as coordenadas x e y podem estar influenciando a variação do atributo na área de uma forma linear.

### 4. Estimação dos parâmetros dos modelos de covariância escolhidos;

```
> ml1
```

```
likfit: estimated model parameters:
```

```
beta tausq sigmasq phi
"15.9926" " 8.5816" "24.7838" " 0.0029"
```

```
Practical Range with cor=0.05 for asymptotic range: 0.008700121
```

```
> ml2
```

```
likfit: estimated model parameters:
```

```
beta0 beta1 beta2 tausq sigmasq phi
"5.495e+04" "2.284e+02" "9.858e+02" "1.000e+01" "1.206e+01" "1.600e-03"
```

```
Practical Range with cor=0.05 for asymptotic range: 0.004871592
```

### 5. Comparação e escolha do melhor modelo;

ml1 → likfit: maximised log-likelihood = -181.9

ml2 → likfit: maximised log-likelihood = -180.2

O modelo ml2, por obter o maior valor do log da verossimilhança (-180.2 > -181.9), deve ser o escolhido.

## 6. Plotar mapas de predição utilizando o modelo previamente selecionado;

Antes de plotar os mapas de predição é necessário construir o grid onde serão feitas as predições.

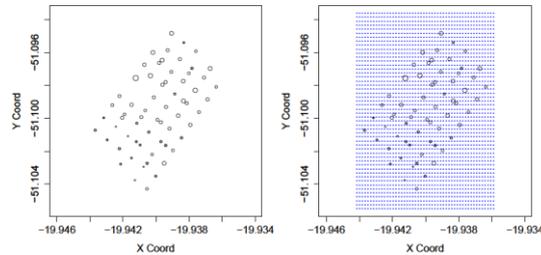


Figura 11: Grid de predição da Fazenda Arvoredo

Neste ponto, diferentemente da simulação realizada anteriormente, não desejamos prever valores fora dos limites da propriedade. Precisaremos, portanto, indicar ao programa para que faça a predição apenas na área de nosso interesse.

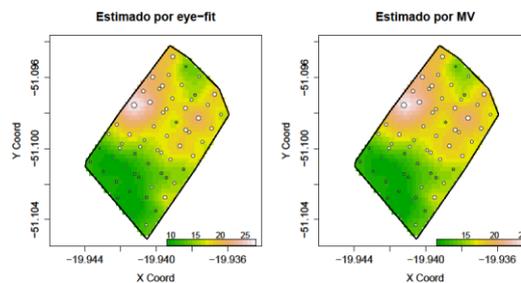


Figura 12: Mapa de predição por krigagem estimado pelo eyefit e por máxima verossimilhança

O primeiro mapa da Figura 12 mostra uma predição realizada com os parâmetros do variograma empírico, ajustado pelo eyefit. Já o segundo mapa mostra um ajuste feito pelo método da máxima verossimilhança e que usou os valores do variograma empírico como chute inicial para seu processo. O resultado foram dois mapas bem semelhantes.

O mapa de variância de krigagem também ficou bem semelhante para as duas situações, conforme pode-se observar na Figura 13.

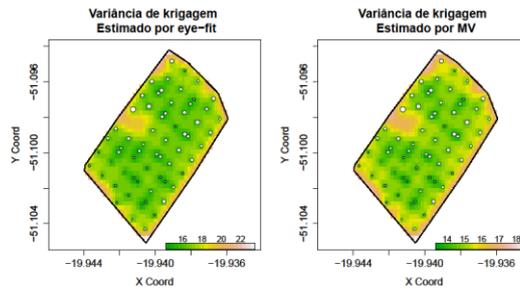


Figura 13: Mapas de variância de Krigagem estimado por eyefit e MV

Neste momento já é possível plotar o mapa de predição relativo a pontos que possam ser nosso interesse, como min, 1°, 2°, 3° e max.

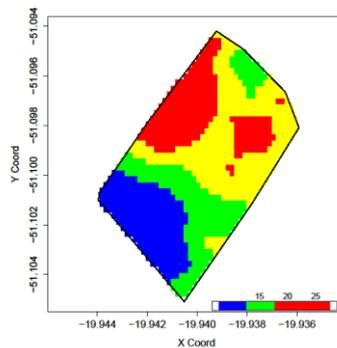


Figura 14: Mapa de predição dividido em poucas classes de interesse.

Na Figura 14 escolheu-se fazer uma krigagem com apenas 4 classes delimitadas pelo menor valor, 1° quartil, 2° quartil (mediana), 3° quartil e o máximo valor amostrado. Poderíamos dividi-lo também em pontos abaixo e acima de 13.

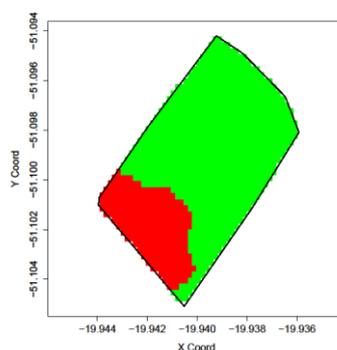


Figura 15: Mapa de predição dividido em duas classes de interesse – abaixo e acima de 13 unidades de Ca.

Na Figura 15 podemos ter uma idéia dos locais onde o nível de Ca está abaixo de nosso limite pré-estabelecido de 13 unidades, e onde está acima. No entanto, devemos lembrar que esse mapa foi construído baseado em apenas uma observação (uma predição). Na etapa seguinte construiremos o mesmo mapa, mas baseado em 1.000 simulações.

## 7. Gerar simulações condicionais de forma a “validar” os mapas de predição por meio da construção de intervalos de confiança para cada ponto estimado.

Da mesma forma que foi feito com os dados simulados, nesta etapa iremos simular 1.000 cenários com modelo de covariância e estimativa de parâmetros iguais as que foram usadas para a krigagem – ou seja – simulação de cenários condicional aos valores que já temos.

Utilizaremos o mesmo grid de predição definido anteriormente – 51 colunas e 51 pontos por coluna. No entanto, como informamos ao programa para fazer predição apenas dentro dos limites da área de estudo, teremos não mais 2.601 pontos estimados e sim 1.094, que foi o total de pontos que “caiu” dentro da borda da área. Assim sendo, teremos como resultado desta simulação uma matriz com 1.094 linhas e 1.000 colunas, onde cada linha representa um ponto estimado e cada coluna uma simulação de toda a área.

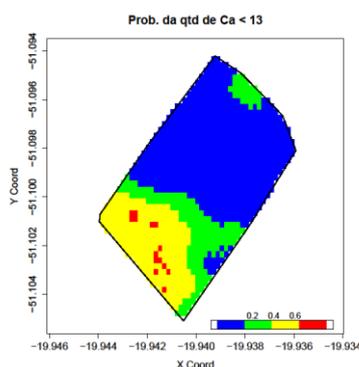


Figura 16: Probabilidade da quantidade de Ca estar abaixo de 13

Na Figura 16 podemos observar a probabilidade da quantidade de Ca estar abaixo do nível crítico pré-determinado – 13 unidades. Esta figura vem confirmar a informação dos locais que mais precisam da adubação com Ca, com um nível de confiança maior do que apenas uma krigagem.

## Conclusões

A análise geoestatística está evoluindo à medida que métodos computacionais ficam cada vez mais disponíveis para os pesquisadores da área. Neste artigo realizou-se uma série de análises, principalmente as que envolveram simulações, inviáveis até pouco tempo atrás.

A utilização da simulação, além de ajudar bastante na calibragem de modelos e estimativas, aumenta o nível de confiança nas análises, uma vez que se pode inferir sobre determinado parâmetro baseado em sua ocorrência média de muitos cenários diferentes.

## Referências bibliográficas

DIGGLE, P.J.; RIBEIRO JUNIOR, P.J.; *Model-based geostatistics. Springer Series in Statistics*. 2007. 229p.

DEUTSCH, C.V. & JOURNEL, A.G. 1998. *GSLIB: Geostatistical Software Library and User's Guide*. Oxford University Press, New York, 2ª Edição, 369p

# Anexo – Comandos do programa R utilizados para realização das análises

## 1. Dados Simulados

```
par.ori <- par(no.readonly=T)
##
require(geoR)
## Dado simulado
## Criando dados com distribuição normal e parâmetros  $\sigma^2=20$ ,  $\phi=.25$ ,
## média = 10 e efeito pepita = 4
set.seed(171)
dat <- grf(100, cov.pars = c(3, .25), nug=4, mean=10)
class(dat) <- "geodata" # transformou em geodata
View(dat)
points(dat,main="100 dados simulados")
## Etapa de análise exploratória dos dados
## Plotar o variograma para ver como os dados se comportam
plot(dat, low=T)
summary(dat)
maior_d = summary(dat)$distances.summary[2]
## Variando a distância máxima de alcance dos dados
variog1 <- variog(dat, max.dist=maior_d)
x11()
plot(variog1)
vario.ef <- eyefit(variog1)
vario.ef
plot(variog(dat))
# Os valores acima servirão para ajudar no chute inicial
# para estimar o modelo (o mesmo que fora simulado mais acima)
## estimação de parâmetros
## e escolha de modelo
var(dat$data)
sigma2<- 3.23
phi<- 0.23
ml1 <- likfit(dat, ini=c(sigma2,phi), cov.model="exponential")
ml2 <- likfit(dat, ini=c(sigma2,phi), trend="1st",cov.model="exponential")
ml1; ml2;
ml <- ml2
## definindo o grid de predição
# Para meus dados, utilizar a área amostrada
par(mfrow=c(1,2))
points(dat,main="Área de estudo")
points(dat,main="Malha de pontos \n na área")
gr <- expand.grid((0:50)/50,(0:50)/50)
points(gr, pch=19, cex=0.25, col=4)
## krigagem (ordinária)
## A krigagem faz a predição, por meio da extrapolação, do atributo
## estudado em toda a área, usando os dados "amostrados" para predizer
## a região que não foi "amostrada"
```

```

kc <- krige.conv(dat, loc=gr,krige=krige.control(obj.model=ml))
## mapa de predição por krigagem (E[S|y])
par(mfrow=c(1,2))
image(kc, col=terrain.colors(21), x.leg=c(0.5,1), y.leg=c(0, 0.05))
image(kc, col=terrain.colors(21), x.leg=c(0.5,1), y.leg=c(0, 0.05))
points(dat, add=T)
## mapa de variância de krigagem (E[S|y])
par(mfrow=c(1,2))
image(kc, col=terrain.colors(21), val=kc$krige.var,
      x.leg=c(0.5,1), y.leg=c(0, 0.05))
image(kc, col=terrain.colors(21), val=kc$krige.var,
      x.leg=c(0.5,1), y.leg=c(0, 0.05))
points(dat, add=T)
View(kc$krige.var)
min_dados<-summary(dat)$data.summary[1]
Q1<-summary(dat)$data.summary[2]
Q2<-summary(dat)$data.summary[4]
Q3<-summary(dat)$data.summary[5]
max_dados<-summary(dat)$data.summary[6]
## dividindo o mapa de krigagem em alguns poucas (3) classes
par(mfrow=c(1,1))
image(kc, breaks=c(min_dados,Q1,Q2,Q3,max_dados),
      col=c("black","blue","yellow","red"),
      x.leg=c(0.5,1), y.leg=c(0, 0.05))
image(kc, breaks=c(5, 7.5, 10, 13), col=c("green","yellow","red"))
## gerando simulações condicionais
OC <- output.control(n.predictive=1000)
## O PROXIMO COMANDO DEMORA UM POUCO PARA RODAR
kc <- krige.conv(dat, loc=gr,krige=krige.control(obj.model=ml),output=OC)
names(kc)
## cada coluna do objeto a seguir guarda uma simulação condicional
## cada linha corresponde a um ponto/pixel do grid de predição
dim(kc$simulations)

## mapa de krigagem e de algumas simulacoes condicionais
par(mfrow=c(1,2))
image(kc,col=terrain.colors(21),main="Krigagem ordinária")
image(kc, val=kc$simulations[,1], col=terrain.colors(21),main="1° Simulação")
image(kc, val=kc$simulations[,10], col=terrain.colors(21),main="10° Simulação")
image(kc, val=kc$simulations[,214], col=terrain.colors(21),main="214° Simulação")
image(kc, val=kc$simulations[,520], col=terrain.colors(21),main="520° Simulação")
image(kc, val=kc$simulations[,923], col=terrain.colors(21),main="923° Simulação")
par(par.ori)
OC <- output.control(quantile=c(0.1, 0.5, 0.9))
kc <- krige.conv(dat, loc=gr,krige=krige.control(obj.model=ml),output=OC)
names(kc)
## Distribuição preditiva do funcional:
## Mapa de probabilidades P[S|y] < 35]
par(mfrow=c(1,1))
P1 <- apply(kc$simulations, 1,function(x) mean(x < 9))
## separando em 4 faixas/intervalos de probabilidade
image(kc, val=P1, breaks=c(0, .25, .50, .75, 1),

```

```
col=c("green","blue","yellow","red"),
x.leg=c(0.5,1), y.leg=c(0, 0.05))
```

## 2. Dados reais – Fazenda Arvoredo

```
getwd()
setwd("C:/Users/Gabriel/Desktop/relatorio_geo/dados_faz_arvoredo")
par.ori <- par(no.readonly=T)
require(geoR)
fazenda.arvoredo <- read.csv("fazenda arvoredo.csv",sep=";", dec=",")
## Montando a geodata a partir dos dados de campo - Fazenda Arvoredo
#View(fazenda.arvoredo) # Visualizar a tabela de dados completa
#head(fazenda.arvoredo) # Visualizar a apenas o cabeçalho e as 6 primeiras linhas
#names(fazenda.arvoredo)# Exibe o nome de todas as colunas no documento
attach(fazenda.arvoredo)
M <- cbind(LATITUDE, LONGITUDE)
y <- Ca
dat <- list(coords = M, data = y)
class(dat) <- "geodata"
#View(dat) # Visualizar a tabela de dados selecionados e
# transformados em geodata. Estes serão os dados analisados

##Definindo as bordas manualmente
# plot(M)

## Fixando os limites do gráfico
## Para definir os limites do gráficos, usarei as próprias coordenadas dos pontos
## amostrados. Os limites do eixo X serão as coord. mínimas e máximas coordenadas de X
## Os limites do eixo Y serão as coord. mínimas e máximas coordenadas de Y
## Acrescentei uma "gordura" de cada lado para o gráfico ficar mais bonito.
## A esta gordura eu dei o nome de "c", que nada mais é do que uma %
## da menor distância entre os pontos

c = 0.1 * summary(dat)$distances.summary[1] #
lim_x = c(summary(dat)$coords.summary[1]*(1+c),summary(dat)$coords.summary[2]*(1-c))
lim_y = c(summary(dat)$coords.summary[3]*(1+c),summary(dat)$coords.summary[4]*(1-c))
plot(M, xlim=lim_x,ylim=lim_y)

# bor <- locator(type="l") # clicar os pontos na tela gráfica (esquerda)
# e direita para encerrar
##View(bor)
##convertendo lista em matrix
# bor <- matrix(unlist(bor), ncol=2)
# bor
## fechar o poligono
# bor <- rbind(bor, bor[1,])
# plot(M, xlim=lim_x,ylim=lim_y)
# polygon(bor)
```

```

#library("MASS")

## Carregar o pacote MASS para poder usar a função write.matrix abaixo

# write.matrix(bor,file="bor.txt")

## O comando acima exporta as coordenadas da borda criada manualmente para um arquivo
chamado "bor"
## na pasta escolhida como diretório pelo R. Nas próximas análises não precisaremos
## definir a borda manualmente novamente.

bor <- read.table("C:/Users/Gabriel/Desktop/relatorio_geo/dados_faz_arvoredo/bor.txt",
                 quote="\")

points(dat,xlim=lim_x,ylim=lim_y)
polygon(bor)

# Análise exploratória dos dados
plot(dat, low=T)
summary(dat)
maior_d = summary(dat)$distances.summary[2]

# Elaboração do Variograma
variog1 <- variog(dat, max.dist=maior_d)

x11()
plot(variog1)
vario.ef <- eyefit(variog1)

## Com a ajuda do vario.ef, encontrar os valores estimados de sigma, phi e outros
vario.ef

## estimação de parâmetros e escolha do melhor modelo

var(dat$data)

sigmasq<-29.66
phi<-0.0029
tausq<-10.27

ml1 <- likfit(dat, ini=c(sigmasq, phi), nug=tausq)
ml2 <- likfit(dat, ini=c(sigmasq, phi), nug=tausq, trend="1st")

ml1
ml2

## definindo o grid de predição
# Para meus dados, utilizar a área amostrada
par(mfrow=c(1,2))
points(dat,xlim=lim_x, ylim=lim_y)

```

```

points(dat,xlim=lim_x, ylim=lim_y)
gr <- expand.grid(seq(lim_x[1],lim_x[2],len=50),seq(lim_y[1],lim_y[2],len=50))
points(gr, pch=19, cex=0.25, col=4)

## krigagem (ordinária)
# A krigagem faz a predição, por meio da extrapolação, do atributo
# estudado em toda a área, usando os dados "amostrados" para predizer
# a região que não foi "amostrada"
kc1 <- krige.conv(dat, loc=gr,krige=krige.control(obj.model=vario.ef))

kc2 <- krige.conv(dat, loc=gr,krige=krige.control(obj.model=ml2))

## mapa de predição por krigagem (E[S|y])

par(mfrow=c(1,2))

y_leg =c(lim_y[1]+ 0.00006,lim_y[1]+ 0.000425)
x_leg =c(lim_x[2]-0.0035,lim_x[2]+0.0015)

image(kc1, col=terrain.colors(21),borders=bor,
      x.leg=x_leg,y.leg=y_leg,
      xlim=lim_x,ylim=lim_y,
      main="Estimado por eye-fit")

points(dat, add=T)

image(kc2, col=terrain.colors(21),borders=bor,
      x.leg=x_leg,y.leg=y_leg,
      xlim=lim_x,ylim=lim_y,
      main="Estimado por MV")

points(dat, add=T)

## mapa de variância de krigagem (E[S|y])

par(mfrow=c(1,2))

image(kc1, col=terrain.colors(21), val=kc1$krige.var,borders=bor,
      x.leg=x_leg,y.leg=y_leg,
      xlim=lim_x,ylim=lim_y,
      main="Variância de krigagem \n Estimado por eye-fit")

points(dat, add=T)

image(kc2, col=terrain.colors(21), val=kc2$krige.var,borders=bor,
      x.leg=x_leg,y.leg=y_leg,
      xlim=lim_x,ylim=lim_y,
      main="Variância de krigagem \n Estimado por MV")

points(dat, add=T)

## mapa de erro padrão de predição/krigagem (E[S|y])

```

```

par(mfrow=c(1,2))

image(kc1, col=terrain.colors(21), val=sqrt(kc1$krige.var),borders=bor,
      x.leg=x_leg,y.leg=y_leg,
      xlim=lim_x,ylim=lim_y,
      main="Variância de krigagem \n Estimado por eye-fit")

points(dat, add=T)

image(kc2, col=terrain.colors(21), val=sqrt(kc2$krige.var),borders=bor,
      x.leg=x_leg,y.leg=y_leg,
      xlim=lim_x,ylim=lim_y,
      main="Variância de krigagem \n Estimado por MV")

points(dat, add=T)

par(mfrow=c(1,1))

# A partir deste momento será adotado apenas a krigagem estimada por MV

kc <- kc2

# Dividindo o mapa de krigagem em poucas classes
min_dados<-summary(dat)$data.summary[1]
Q1<-summary(dat)$data.summary[2]
Q2<-summary(dat)$data.summary[4]
Q3<-summary(dat)$data.summary[5]
max_dados<-summary(dat)$data.summary[6]

summary(dat)
par(mfrow=c(1,2))
image(kc, breaks=c(min_dados,Q1,Q2,Q3,max_dados),
      col=c("blue","green","yellow","red"),borders=bor,
      x.leg=x_leg,y.leg=y_leg,
      xlim=lim_x,ylim=lim_y)
max_dados
# Dividindo o mapa de krigagem em 2 classes - Abaixo e Acima de 13
image(kc, breaks=c(7,13,33),
      col=c("red","green"),borders=bor,
      x.leg=x_leg,y.leg=y_leg,
      xlim=lim_x,ylim=lim_y)

## gerando simulações condicionais
OC <- output.control(n.predictive=1000)
## O PROXIMO COMANDO DEMORA UM POUCO PARA RODAR
kc <- krige.conv(dat, loc=gr,krige=krige.control(obj.model=ml2),output=OC,
                borders=bor)

names(kc)
## cada coluna do objeto a seguir guarda uma simulação condicional
## cada linha corresponde a um ponto/pixel do grid de predição

```

```
dim(kc$simulations)

## Mapa de probabilidades  $P[S|y] < 13$ 
P1 <- apply(kc$simulations, 1,function(x) mean(x < 13))

## separando em 4 faixas/intervalos de probabilidade

image(kc, val=P1, breaks=c(0, .25, .50, .75, 1),
      col=c("blue", "green", "yellow", "red"),
      x.leg=x_leg,y.leg=y_leg,
      main="Prob. da qtd de Ca < 13")
```