PYLPM – UMA BIBLIOTECA GEOESTATÍSTICA DE CÓDIGO ABERTO EM PYTHON

David AlvarengaDrumond, UFRGS (Universidade Federal do Rio Grande do Sul)
david.engminas.ufmg@gmail.com

Roberto Mentizingen Rolo, UFRGS(Universidade Federal do Rio Grande do Sul)

João Felipe Coimbra Leite Costa, UFRGS(Universidade Federal do Rio Grande do Sul)

RESUMO

Na indústria da mineração, a geoestatística é aplicada em diversas etapas durante a vida de um projeto. Algumas aplicações dos algoritmos utilizados na geoestatística podem demandar grande esforço computacional. Por esse motivo, é importante que os algoritmos sejam otimizados e desenvolvidos em linguagens de programação eficientes. Essas características fazem com que a produção de softwares específicos demande mão de obra especializada, seja custosa e demorada. Apesar dessas características, existem algumas alternativas gratuitas e de código aberto. Esse trabalho apresenta uma biblioteca geoestatística *de código aberto*, cujo objetivo é auxiliar professores e alunos no ensino da geoestatística. Um estudo de caso no banco de dados Walker Lake é apresentado para ilustrar seu funcionamento.

Palavras Chave:Geoestatística; Bibliotecas de código aberto; Python; GSLib

ABSTRACT

In the mining industry, geostatistics is applied in several stages during the life of a project. Some applications of the algorithms used in geostatistics can demand great computational effort. For this reason, it is important that the algorithms are optimized and developed in efficient programming languages. These characteristics make the production of specific software demanding specialized labor, costly and time-consuming. Despite these characteristics, there are some free and open source alternatives. This work presents an

open source geostatistics library, whose objective is to assist teachers and students in the teaching of geostatistics. A case study in the Walker Lake dataset is presented to illustrate how it works.

Keywords:Geostatistics; Open source package; Python; GSLib

INTRODUÇÃO

Geoestatística é o ramo da estatística aplicado aos bancos de dados georreferenciados. Foi originalmente desenvolvido por Georges Matheron [1] e Daniel Krige, em meados dos anos 1960, a partir da construção de modelos probabilísticos para modelagem matemática de depósitos minerais.

Atualmente, a geoestatística encontra aplicação em diversas disciplinas, incluindo geologia do petróleo, hidrogeologia, hidrologia, meteorologia, oceanografia, geoquímica, geometalurgia, geografia, silvicultura, controle ambiental, ecologia da paisagem, ciência e agricultura.

Na indústria da mineração a geoestatística é aplicada em diversas etapas durante a vida de um projeto: inicialmente, utiliza-se o modelo para previsão de recursos minerais do depósito e avaliação da viabilidade econômica do projeto (projetos greenfield). Por conseguinte, modelos atualizados fazem parte de uma rotina diária no controle de teores do material explotado, auxiliando na decisão do encaminhamento do material extraído (projetos brownfield) para a pilha de estéril ou usina de beneficiamento. Dessa forma, a geoestatística se torna uma importante ferramenta de decisão na mineração.

Algumas aplicações dos algoritmos utilizados na geoestatística podem demandar grande esforço computacional. Alguns exemplos envolvem a busca no domínio por pares de amostras georreferenciadas, ou a construção e inversão de matrizes de covariância em alguns casos até mesmo milhões de vezes. Por esse motivo, é importante que os algoritmos sejam otimizados e desenvolvidos em linguagens de programação eficientes. Essas características fazem com que a produção de softwares específicos demande mão de obra especializada, seja custosa e demorada.

Existem uma gama de softwares profissionais de engenharia que permitem a integração de módulos geoestatísticos. De forma geral, os softwares de planejamento de mina, como *Datamine Studio* , *Surpac, Micromine, MinePlan, Deswik* e *Vulcan* contam com um módulos específicos de geoestatística. Existem também softwares exclusivamente destinados aos cálculos geoestatísticos como o *Isatis*, e o *Resource Modeling Solutions Platform*. Há também soluções que usam a geoestatística para criação de modelos que não sejam de teores de minério, como o Leapfrog por exemplo, que utiliza algoritmos de estimativa na modelagem geológica.

Apesar das características que fazem com que a produção de softwares específicos seja onerosa, existem algumas alternativas gratuitas e de código aberto. O Geostatistical Stanford Library (GSLib) [2] é uma biblioteca de softwares escritos na linguagem de programação FORTRAN, uma linguagem de programação baixo nível conhecida por sua velocidade de execução. As soluções oferecidas são modularizadas em arquivos executáveis individuais. A **Figura 1** apresenta o fluxograma para a geração de uma estatística gráfica no GSLib: (1) Prepara-se o banco de dados no formato adequado para a entrada no GSLib, (2) Utiliza-se um dos programas da biblioteca GSLib, com parâmetros da estatística definidos em um arquivo independente. Os resultados são fornecidos em um arquivo de output. (3) Utilizar como entrada os dados gerados pela estatística em outro programa de apresentação gráfica (4) Obtenção do resultado da estatística gráfica.

Figura 1 – Fluxograma para geração das estatísticas gráficas no GSLib. (1) Edição do arquivo de dados para formato original utilizado no programa (2) Alteração de arquivos de parâmetros da geoestatística e execução do programa (3) Edição do arquivo de parâmetros da estatística gráfica e execução do programa (4) Obtenção do resultado final da estatística gráfica.

Como observado na **Figura 1**, caso o usuário deseje modificar apenas um parâmetro da estatística, é necessário realizar, pelo menos, quatro etapas. Ao mesmo tempo que sua concepção modular torna o GSLib bastante flexível, são necessárias muitas etapas para certos cálculos em um ambiente sem interface gráfica, o que pode dificultar a curva de aprendizagem da ferramenta, principalmente para iniciantes.

Para sanar as deficiências na operação do GSLib, algumas soluções para administrar as etapas de execução dos programas da biblioteca foram desenvolvidas em linguagem Python. Em conjunto, também são utilizadas diferentes ferramentas para permitir o registro de cálculo das análises e utilização destes algoritmos. Uma das alternativas para usuários de Python é o *jupyter notebooks*. Esta tecnologia permite a utilização concomitante de código, juntamente com textos em Markdown e LaTex, permitindo a visualização de gráficos e resultados em células distintas. Os resultados obtidos pelas estatísticas, assim como os parâmetros utilizados nas funções estatísticas e gráficos, permanecem armazenados em um único documento.

A biblioteca Pygeostats [3] foi pioneira. Apresentada pelo Centro de Geoestatística Computacional da Universidade de Alberta (CCG), desenvolvedor da biblioteca GSLib. Outras soluções apresentadas são o pyGSLIB [4] e o Geostatspy [5]. Todas essas utilizam a linguagem Python como ferramenta administrativa para executáveis do GSLib.

Há também softwares de código aberto para geoestatística sem a utilização da biblioteca GSLib como cerne, como o Stanford Geostatistical Modelling Software (SgeMS) [6], escrito em C++. Esse fornece aos usuários uma interface amigável, com visualização interativa de dados em 3D, e uma ampla coleção de algoritmos. Uma das soluções mais recentes é o GeoStats.jl [7], uma aplicação extensível para geoestatística de alto desempenho escrita em linguagem Julia.

Esse trabalho apresenta uma biblioteca geoestatística *de código aberto*, cujo objetivo é auxiliar professores e alunos no ensino da geoestatística. Os fluxos de trabalho são executados em *jupyter notebooks*. Diferentemente das soluções anteriores, a biblioteca implementada possui maiores recursos didáticos, como gráficos interativos e possibilidade de organização do código em um único ambiente de trabalho.

METODOLOGIA

A biblioteca PyLPM [8] foi uma criação desenvolvida por alunos do Laboratório de Pesquisa Mineral e Planejamento Mineiro da Universidade Federal do Rio Grande do Sul (LPM), com finalidade didática. A documentação pode ser encontrada na plataforma Read the Docs [9], juntamente com o manual de instalação, a descrição de todos os módulos, classes, métodos e funções. Na documentação é possível encontrar o estudo de caso completo no banco de dados *Walker Lake* [10], mostrado em partes na seção estudo de caso deste trabalho.

A biblioteca é composta por cinco módulos com diferentes funções:

- **GSLIB:** este módulo permite acessar os executáveis do GSLib a partir de funções em Python. A definição de parâmetros é a mesma da biblioteca original, porém, é possível obter os resultados, gráficos e análises diretamente em um *jupyter notebook*.
- **Geostat algorithms:** esse módulo contém os algoritmos de geoestatística desenvolvidos em Python pela equipe do LPM.
- **Variography:** módulo de análise variográfica. Esse módulo foi desenvolvido pela equipe do LPM e apresenta ferramentas para o cálculo e a modelagem de funções de continuidade espaciais.
- **Utilities:** esse módulo abriga funções auxiliares na análise geoestatística.
- **Plots**: módulo com as funções gráficas. Há uma extensa gama de funções para apresentação de histogramas, *scatterplots, QQ-plots*, mapa de amostras, *grids*, dentre outras.

O módulo GSLib da biblioteca permite a execução dos programas da biblioteca homônima, a partir de funções em Python, em um jupyter notebook. A **Figura 2** apresenta o fluxograma mostrando como os programas são manipulados pela interface Python desenvolvida. A edição dos arquivos de parâmetros do GSLib, o preenchimento dos arquivos de entrada de dados dos programas, a execução e retroalimentação dos resultados no Jupyter Notebook, são realizados automaticamente pelo módulo em Python, sem que ocorra interferência externa do usuário.

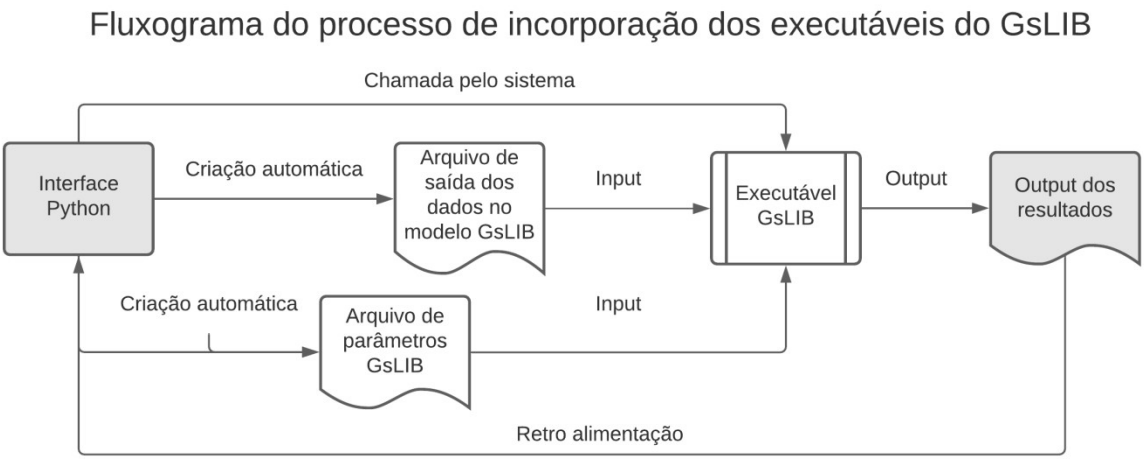

Figura 2 – Fluxograma mostrando a integração “sem emendas” dos programas GSLib com a biblioteca PyLPM.

Atualmente, a biblioteca possui ferramentas limitadas à análise univariada de dados. Esse assunto compreende os programas da maioria das disciplinas introdutórias dos cursos de geoestatística. Entre as ferramentas citam-se:

- As análises exploratórias (uni e bi-variadas) dos dados;
- A Variografia e análise espacial;
- A estimativa geoestatística: krigagem simples e ordinária;
- A Validação dos modelos de estimativa: validação cruzada e swath plots (gráficos de deriva);

O fluxograma apresentado pela **Figura 2** permite com que novos programas do GSLib sejam incorporadas à biblioteca de forma simples e direta, possibilitando o acréscimo de soluções referentes às disciplinas mais avançadas, como simulação geoestatística e análise multivariada.

ESTUDO DE CASO

Um estudo de caso no banco de dados *Walker Lake* é apresentado para exemplificar o funcionamento da biblioteca desenvolvida. Os resultados dessa análise podem ser encontrados na documentação apresentada na plataforma Read the Docs. O fluxo de trabalho apresenta uma análise geoestatística convencional, incluindo da importação dos dados até a validação dos modelos de krigagem.

O objeto utilizado pela biblioteca desenvolvida para armazenar e manipular os dados é o *DataFrame*, da biblioteca Pandas [11], comumente utilizada para manipulação de dados em Python. A importação dos dados pode ser feita de diferentes formas, a partir de diferentes extensões de arquivos (eg.: “.csv” ou “.dat”). No entanto, a biblioteca PyLPM apresenta um módulo contendo bancos de dados tradicionais em geoestatística, os bancos de dados

podem ser carregados a partir da função *datasets,* como mostrado pela **Figura 2**. A **Figura 3** apresenta as instruções utilizadas para a importação do banco de dados *Walker Lake*, que apresenta duas variáveis contínuas V, e U, e uma variável categórica T.

```
01. import pyLPM
02. wl = pyLPM.datasets.dataset('Walker_Lake')
```

Figura 3 – Importação do banco de dados do *Walker Lake* usando a função *datasets*.

A **Figura 4** apresenta o bloco de código necessário para apresentar um mapa de localização da variável categórica T. O resultado é mostrado na **Figura 5**.

```
01. pyLPM.plots.locmap(
02.     x=wl.X,
03.     y=wl.Y,
04.     variable=wl['T'],
05.     categorical=True,
06.     title='Map for T')
```

Figura 4 – Função *locmap* sendo utilizada para apresentar a variável categórica *T* do *Walker Lake*.

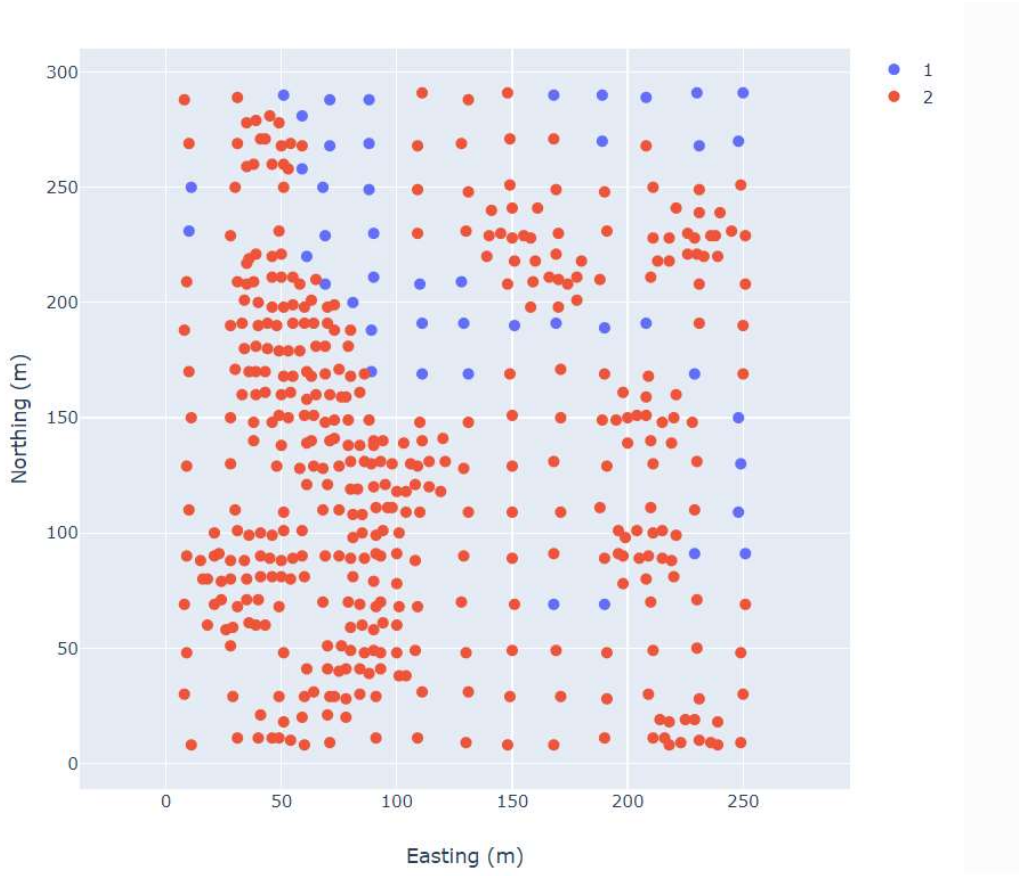

Figura 5 – Mapa de localização das amostras para a variável categórica T.

Da mesma forma que é possível criar mapas de variáveis categóricas, a biblioteca também apresenta uma solução para variáveis contínuas. A **Figura 6** mostra o mapa de localização das 470 amostras, da variável contínua V.

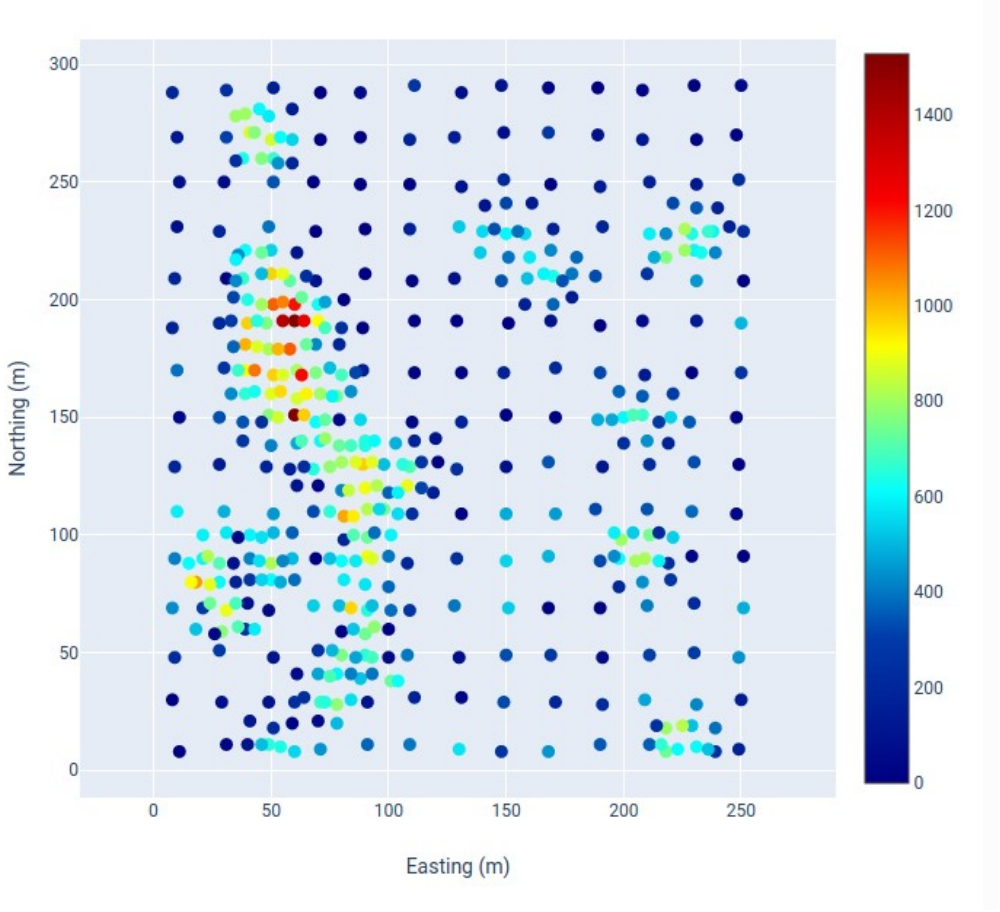

Figura 6 – Mapa de localização das amostras para a variável contínua V.

É possível apresentar histogramas interativos, que apresentam resultados das classes ao posicionar o cursor do mouse sobre as barras. Os histogramas acompanham um sumário com valores de estatísticas pontuais como mostrado na **Figura 7**.

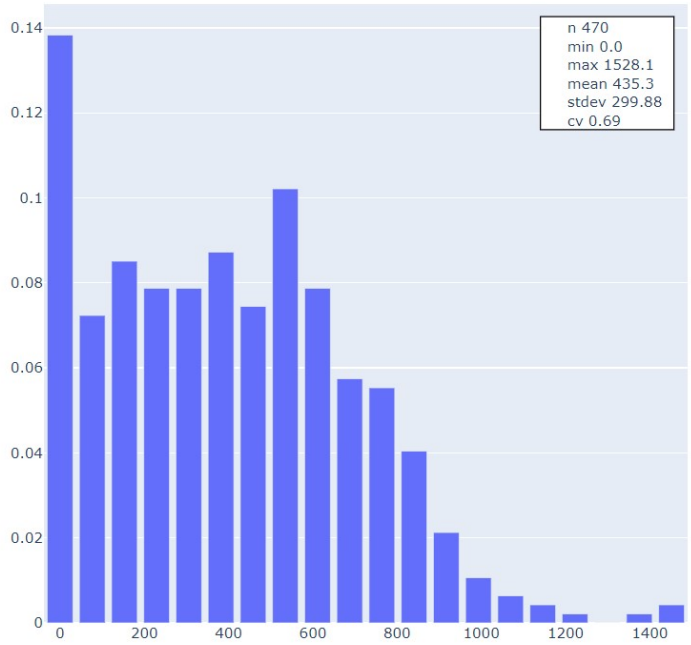

Figura 7 – Histograma e sumário estatístico para a variável V.

A **Figura 8** apresenta o bloco de código para a utilização do algoritmos de desagrupamento por células móveis. A função em Python é homônima ao programa da biblioteca GSLib, sendo os parâmetros os mesmos do programa da biblioteca original.

```
pyLPM.gslib.declus(
    df=wl,
    x='X',
    y='Y',
    z=None,
    var='V',
    x_anis=1.0,
    z_anis=1.0,
    n_cell=50,
    min_size=5.0,
    max_size=45.0,
    keep_min = True,
    number_offsets=100,
    usewine=False)
```

Figura 8 – Execução do software *declus* a partir da biblioteca desenvolvida.

A vantagem da utilização da função é evidente: o *output* do programa *declus, ao ser utilizado na biblioteca GSLib,* é apresentado no prompt de comando, e automaticamente eliminado da memória após fechamento da janela, pois é armazenado na memória RAM do computador. Ao contrário dessa situação, a biblioteca PyLPM permite armazenar esse resultado em um arquivo *jupyter notebook,* trazendo clareza e rastreabilidade dos resultados, como mostrado na **Figura 9**.

```
DECLUS Version: 2.906

 data file = C:\Users\Roberto Rolo\Documents\pyLPM\py
 columns =           1           2           0           3
 tmin,tmax =   -1.000000E+21    1.000000E+21
 summary file = C:\Users\Roberto Rolo\Documents\pyLPM\py
 output file = C:\Users\Roberto Rolo\Documents\pyLPM\py
 anisotropy =        1.000000        1.000000
 minmax flag =          -1
 ncell min max =          50        5.000000       45.000000
 offsets =         100


There are      470 data with:
  mean value             =    435.29860
  minimum and maximum    =       .00000  1528.10000
  size of data vol in X =    243.00000
  size of data vol in Y =    283.00000
  size of data vol in Z =       .00000

  declustered mean       =    290.10570
  min and max weight     =       .28911     2.78945
  equal weighting        =      1.00000


DECLUS Version: 2.906 Finished

Stop - Program terminated.
```

Figura 9 – *Output* do programa *declus* mostrado em uma célula de saída do *jupyter notebook*.

É possível gerar gráficos automáticos a partir das funções da biblioteca em Python, que não são apresentadas pelo GSLib. A **Figura 10** mostra o ponto de decisão do tamanho da célula o qual reduz a média desagrupada, a partir de um gráfico gerado automaticamente. O mesmo processo de decisão demandaria mais tempo, se analisado pelo GSLib, pois esse não apresenta um conjunto de ferramentas para a geração desse gráfico.

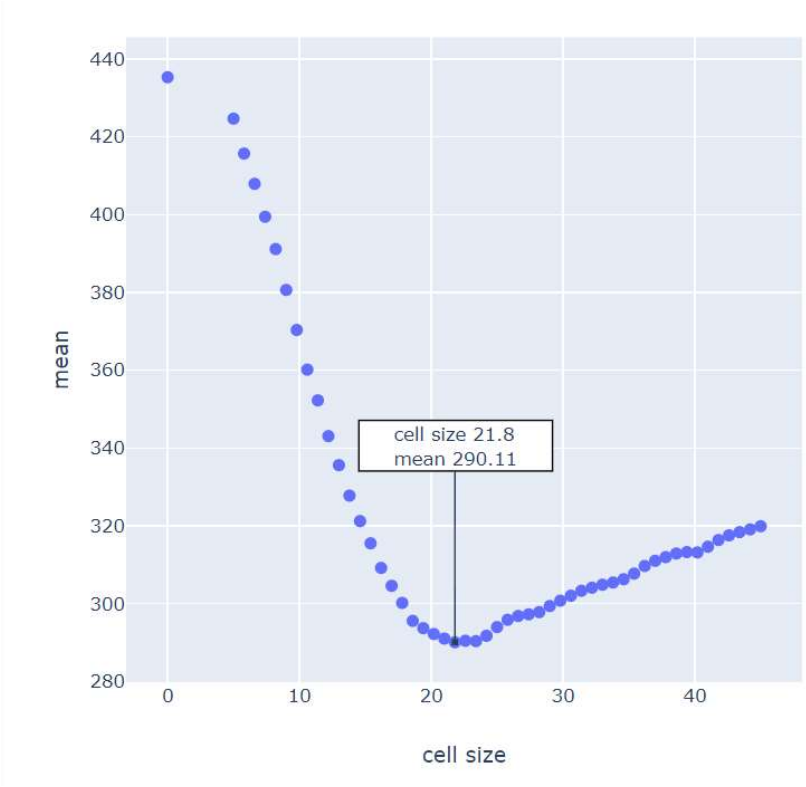

Figura10 – Análise dos resultados do desagrupamento por células.

A biblioteca permite a implementação de funções estatísticas personalizadas a partir de um módulo específico (*geostatalgo*). A **Figura 11** apresenta os parâmetros de entrada do algoritmo de vizinhos mais próximos, desenvolvida em Python pelos alunos do LPM. Este algoritmo não possui integração com a biblioteca GSLib e pode ser utilizado independentemente. Dessa forma, a biblioteca PyLPM permite utilizar tanto as soluções provenientes do GSLib como também a utilização de soluções personalizadas, em um mesmo ambiente de trabalho.

```
01. resultsnn = pyLPM.geostatalgo.nn(
02.     x=wl.X,
03.     y=wl.Y,
04.     z=None,
05.     var=wl.V,
06.     grid=grid)
```

Figura 11 – Apresentação dos parâmetros para criação do mapa dos vizinhos mais próximos.

O resultado do mapa de estimativas por vizinho mais próximo para a variável V é apresentado na **Figura 12.**

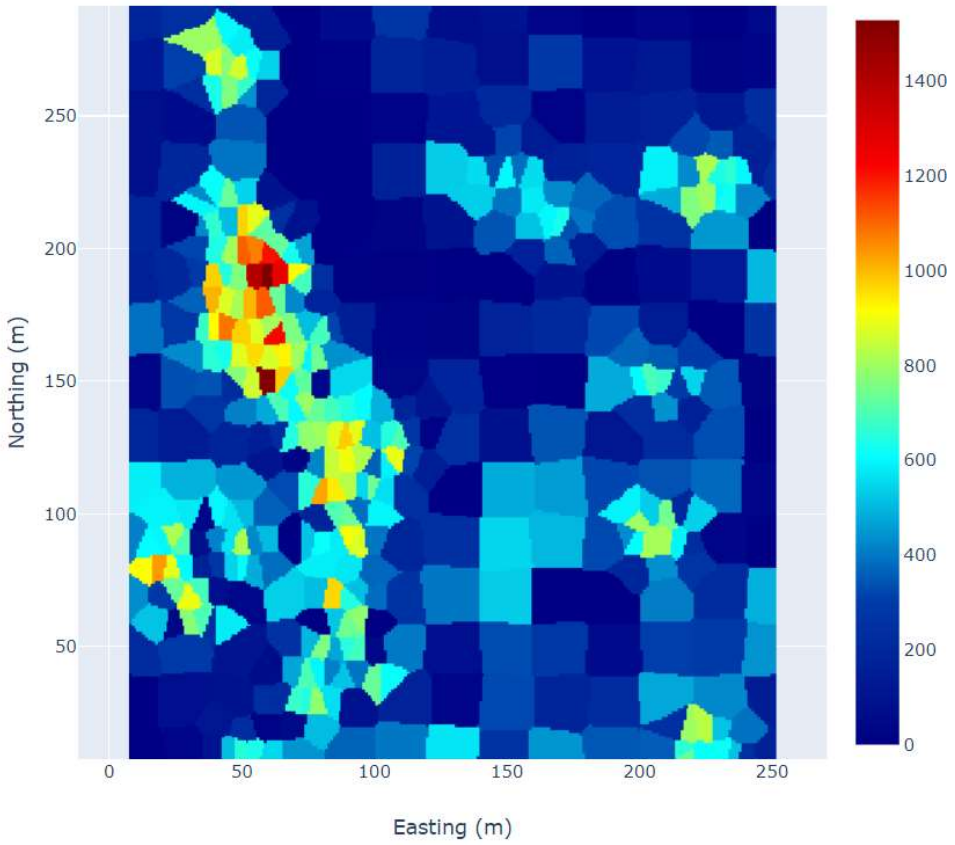

Figura 12 – Mapa de estimativas por vizinho mais próximo para a variável V.

A biblioteca PyLPM apresenta uma série de funções para análise variográfica, desde criação de mapas até o cálculo das funções experimentais de covariância e sua modelagem interativa. A **Figura 13** apresenta o bloco de código para geração de um mapa de

variograma da variável V. Estes mapas não são ferramentas encontradas na biblioteca do GSLib, mas possuem importância significativa na determinação das direções principais da continuidade espacial.

```
pyLPM.gammapy.varmap(
    dataset=wl,
    X='X',
    Y='Y',
    head='V',
    tail='V',
    azimuth=0.,
    dip=0.,
    nlags=10,
    lagdistance=10,
    ndirections=12,
    type_var='Variogram',
    Z=None,
    choice=1.0,
)
```

Figura 13 – Função *varmap* do módulo *gammapy* (*Variography*) sendo usada para criar um mapa de variograma para a variável V.

O mapa de variograma gerado é mostrado na **Figura 14**. Pode-se observar que o resultado do elipsoide de anisotropia apresenta direção principal em N157, como apresentado pelo banco de dados *Walker Lake*.

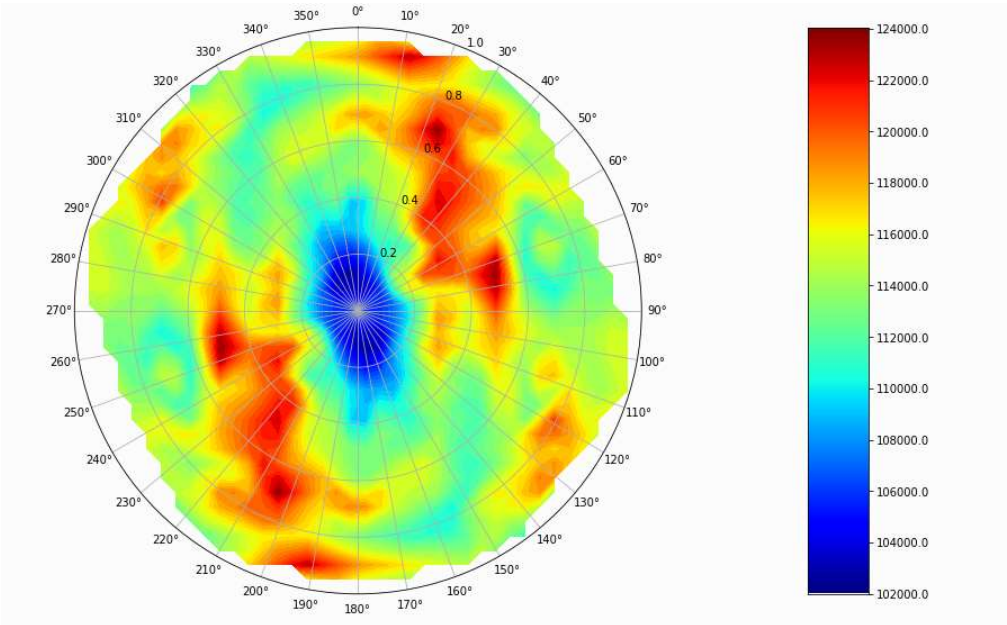

Figura 14 – Mapa de variograma para a variável V.

O cálculo dos variogramas experimentais e sua modelagem são feitos de forma interativa no *jupyter notebook,* a partir de estruturas de controle, como os mostrados na **Figura 15**. Caso seja necessário o uso de soluções mais rápidas em linguagem compilada, estão disponíveis as funções correlatas do GSLib. A **Figura 16** mostra os variogramas calculados e modelados para a variável V.

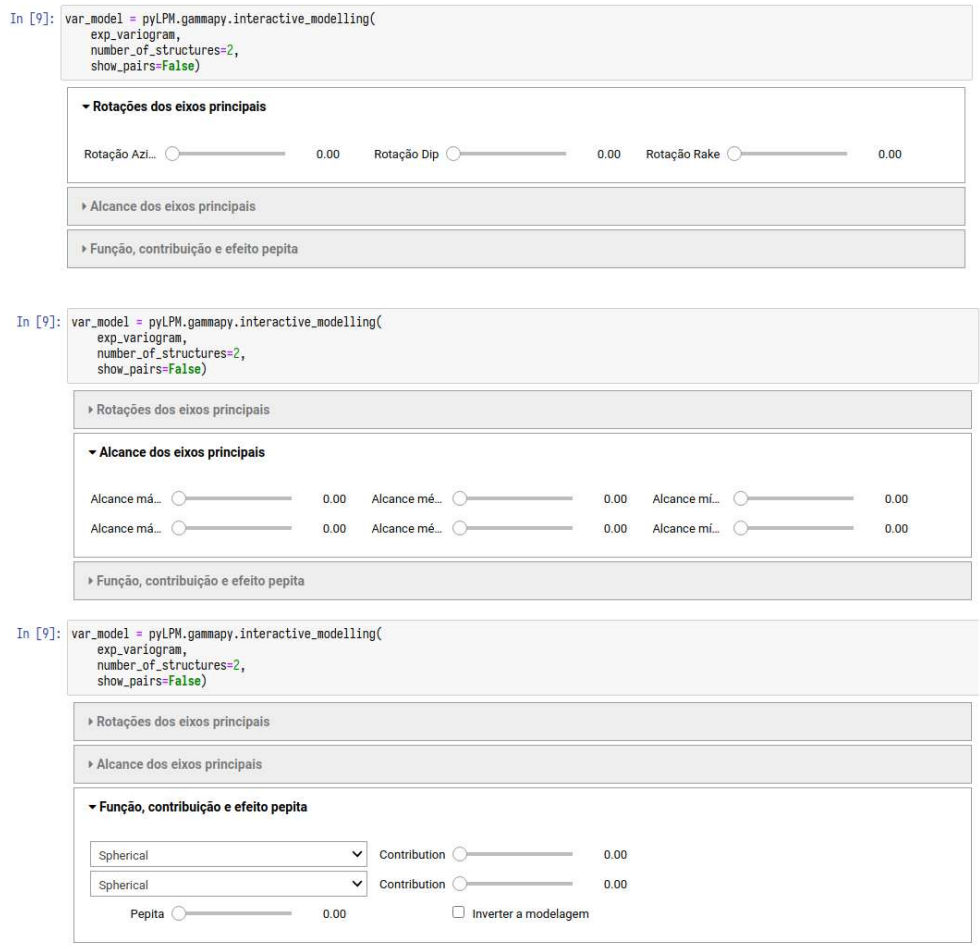

Figura 15 – Controles para modelagem interativa no *jupyter notebook*.

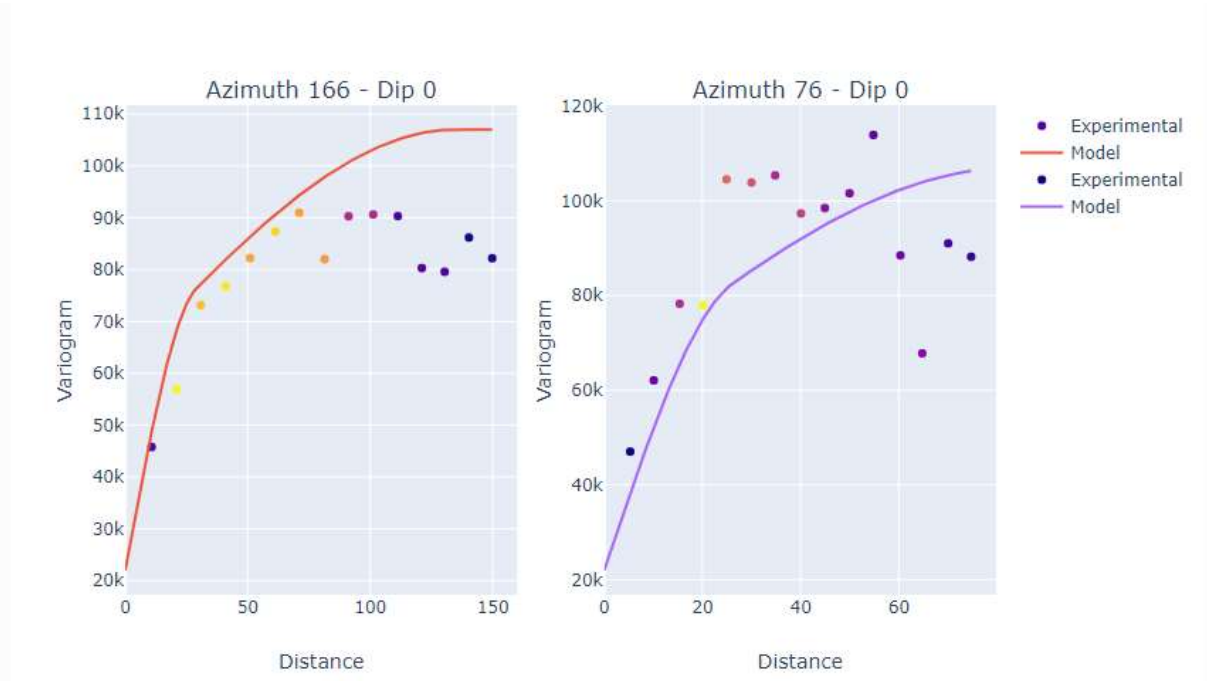

Figura 16 – Variogramas calculados e modelados para a variável V.

A **Figura 17** apresenta os resultados da estimativa do Walker Lake por krigagem ordinária no estudo de caso desenvolvido. As amostras são apresentadas em conjunto com os resultados dos blocos estimados. Além disso, é possível verificar os valores interpolados interativamente com o gráfico, apenas pelo posicionamento do cursor do mouse sobre as células estimadas.

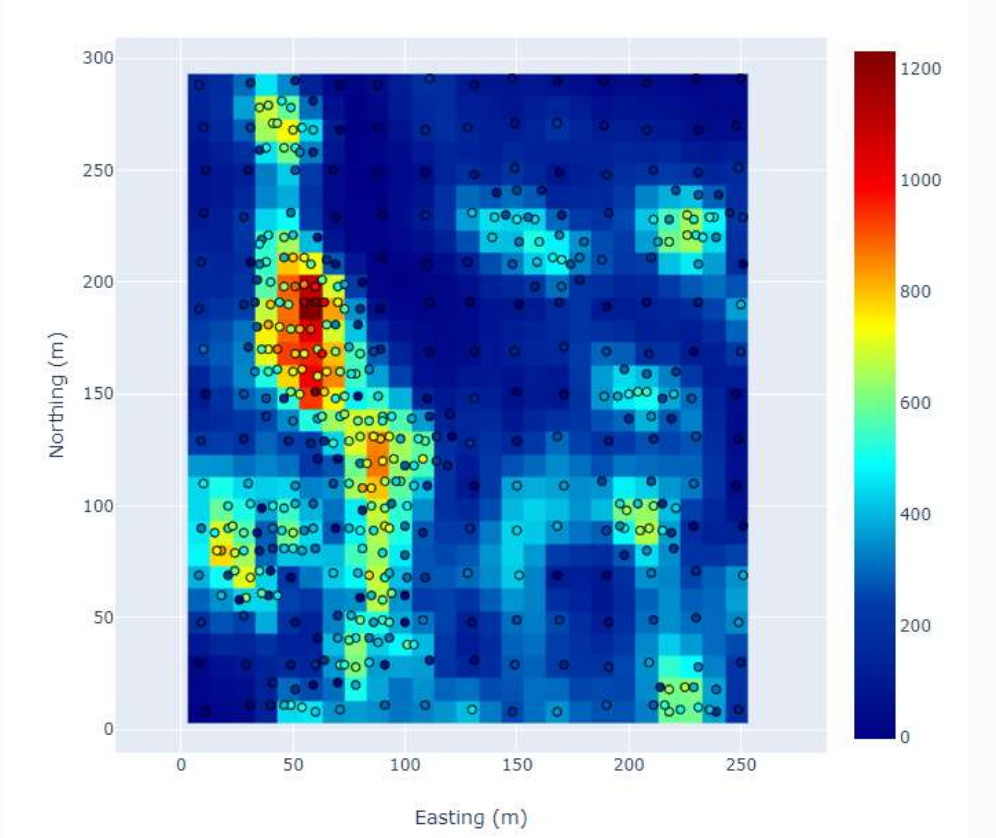

Figura 17 – Resultados da krigagem ordinária para a variável V.

A biblioteca conta com ferramentas de validação dos modelos de estimativa. A **Figura 18** mostra os resultados da validação cruzada, enquanto a **Figura 19** apresenta o gráfico de deriva ou swath plots, que plota as médias das amostras e dos blocos estimados em fatias verticais e horizontais.

Figura 18– Resultados da validação cruzada da krigagem ordinária para a variável V.

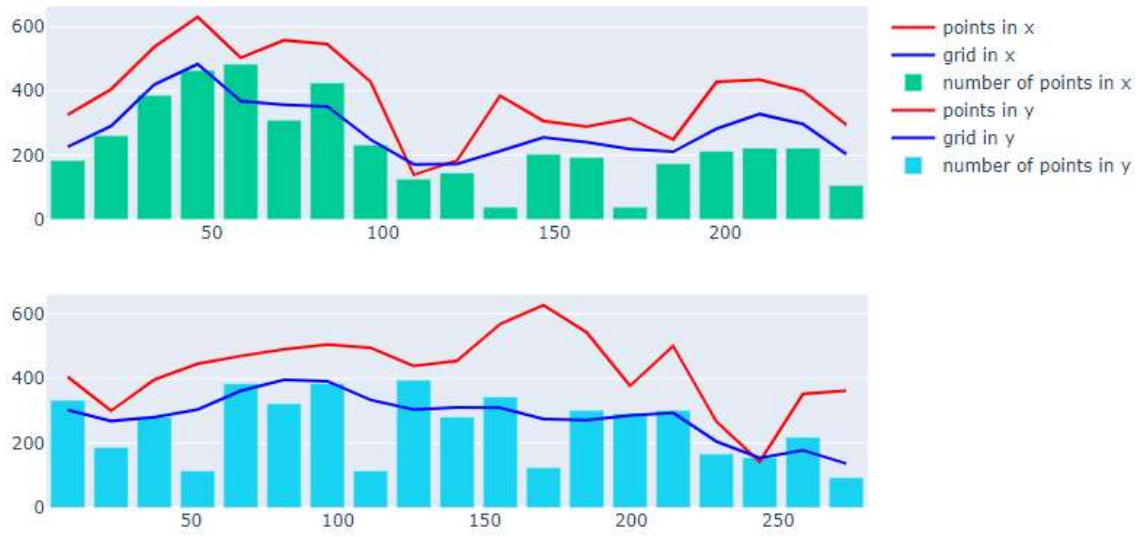

Figura 19– Gráfico de deriva para a variável V.

DISCUSSÃO

Embora existam outras bibliotecas similares, a PyLPM apresenta alguns diferenciais para ensino da geoestatística. O primeiro deles é a possibilidade de gerar gráficos interativos, permitindo obter informações diretas de resultados das estatísticas gráficas. Este tipo de apresentação permite o entendimento e explicação dos processos teóricos da disciplina de forma integrada com a aplicação prática. Apesar das soluções gráficas não permitirem flexibilidade dos estilos gráficos, a ferramenta é facilmente operável, permitindo a criação de exemplos da análise espacial com facilidade.

Um outro diferencial da biblioteca desenvolvida é o módulo de variografia. Para a geração de estatísticas da continuidade espacial, é muitas vezes necessário interagir com os parâmetros destas funções várias vezes. Em um processo tedioso que consiste e alterar valores em um arquivo de texto, executar o programa GSLib em modo linha de comando, e verificar o resultado até que seja satisfatório, a biblioteca Python permite a redução do tempo da análise estatística e garante a visualização de estatísticas gráficas em conjunto com os resultados. O uso de estruturas de controle no *jupyter notebook*, permite variar os parâmetros dos modelos de continuidade espacial em tempo real, facilitando a variografia.

No momento, apenas algoritmos de estimativa univariada estão implementados, entretanto, qualquer usuário pode modificar e distribuir a biblioteca corrigindo eventuais *bugs* e implementando novas funcionalidades como suporte à geoestatística multivariada e simulação geoestatística por exemplo. Adicionar novos programas da biblioteca GSLib é um processo simples e direto, graças ao módulo GSLib da PyLPM.

REFERÊNCIAS

1. Matheron, G. (1971). The theory of regionalised variables and its applications. Les Cahiers du Centre de Morphologie Mathématique.

2. Deutsch, C.V. and Journel, A.G., (1997). GSLIB Geostatistical Software Library and. User's Guide, Oxford University Press, New York, second edition.

3. Documentação da biblioteca Pygeostats. Universidade de Alberta. url: (https://www.ccgalberta.com/pygeostat/). Acessado em 16/03/2021

4. Documentação da biblioteca pyGSLIB. url: (https://opengeostat.github.io/pygslib). Acessado em 16/03/2021

5. Documentação GeostatsPy. url: (https://github.com/GeostatsGuy/GeostatsPy). Acessado em 16/03/2021

6. SgeMS. Universidade de Stanford. url: (https://sgems.sourceforge.net/) . Acessado em 16/03/2021

7. Repositório do GeoStats.jl. url:(https://github.com/JuliaEarth/GeoStats.jl) . Acessado em 16/03/2021

8. Repositório da biblioteca PyLPM. url:(https://github.com/robertorolo/pyLPM) . Acessado em 16/03/2021

9. Issaks, E. H., & Srivastava, R. M. (1989). An introduction to applied geostatistics. New York.

10.Documentação da biblioteca PyLPM. url:(https://pylpm.readthedocs.io/en/latest/) . Acessado em 16/03/2021

11 .Biblioteca Pandas. url:(https://pandas.pydata.org.) . Acessado em 16/03/2021