

D17 – SIG no ambiente de Ciências de Dados - Aula 3

Agenda

- Ambiente Ciência de Dados com SIG
- Jupyter Notebook (Revisão)
- Biblioteca GDAL
- Outras bibliotecas de SIG
- GeoPandas
- Bibliotecas de Mapas

Ambiente Ciência de Dados com SIG

- Existe vários ambientes, como:
 - Ferramenta de desenvolvimento (IDE) - Ex.: VSCode
 - BI e Visualização - Ex.: Power BI
 - Notebook – Ex.: Jupyter Lab, **Google Colab** (nuvem)
 - Big Data – Ex.: Apache Spark.
- Google Colab:
 - As bibliotecas precisam ser instaladas a cada “Execução da máquina”
 - A sessão inicia as variáveis do Notebook (não altera o estado máquina)
- Bibliotecas: GDAL, Shapely, RasterIO, XArray e Geopandas.
- Notebook (“ambiente_sig_ciencia_dados.ipynb”):
<https://colab.research.google.com/drive/1PrXCTcpzszp9Z8cbMWGT6P86ZtR0xNT7>
* Precisa ter uma conta na Google

Jupyter Notebook (Revisão)

- No notebook (ambiente_sig_ciencia_dados.ipynb) executar a célula com as instruções de instalação.
- No Google Colab o Jupyter Notebook é customizado.
- Ambiente:
 - Menu/Arquivo:
 - Modelo playground (não grava)
 - Locais: Google Drive (cria uma pasta), GitHub
 - Revisão (versionamento)
 - “Célula”:
 - Unidade do Notebook, pode ser do tipo “Texto” ou “Código”
 - O Texto utiliza a linguagem Markdown
 - No código podemos fazer “chamadas” para o Sistema Operacional (!)

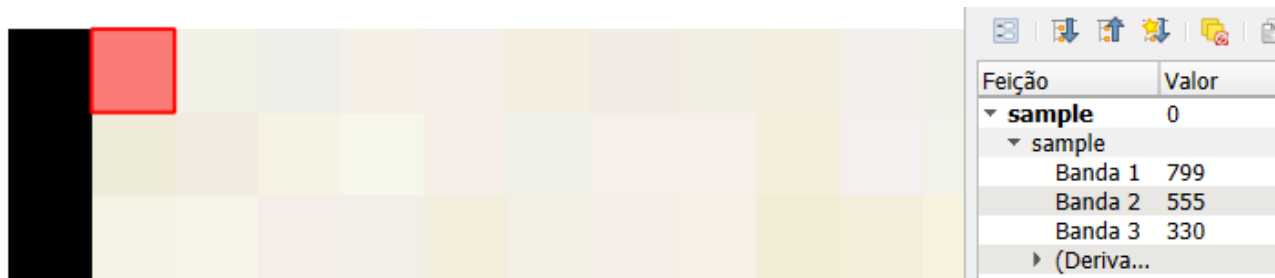
Jupyter Notebook (Revisão) – Continuação

- Ambiente:
 - Menu/Arquivo:
 - Modelo playground (não grava)
 - Locais: Google Drive (cria uma pasta), GitHub
 - Revisão (versionamento)
 - “Célula”:
 - Unidade do Notebook, pode ser do tipo “Texto” ou “Código”
 - O Texto utiliza a linguagem Markdown
 - No código podemos fazer “chamadas” para o Sistema Operacional (!)
 - Formatação do Notebook (linguagem Markdown):
 - O ambiente facilita a edição
 - Índice: Agrupa conteúdos (facilita a navegação)
 - Snippets: trechos prontos de código
 - Formulário: GUI para entrada de dados (vai colocando o valor no código)

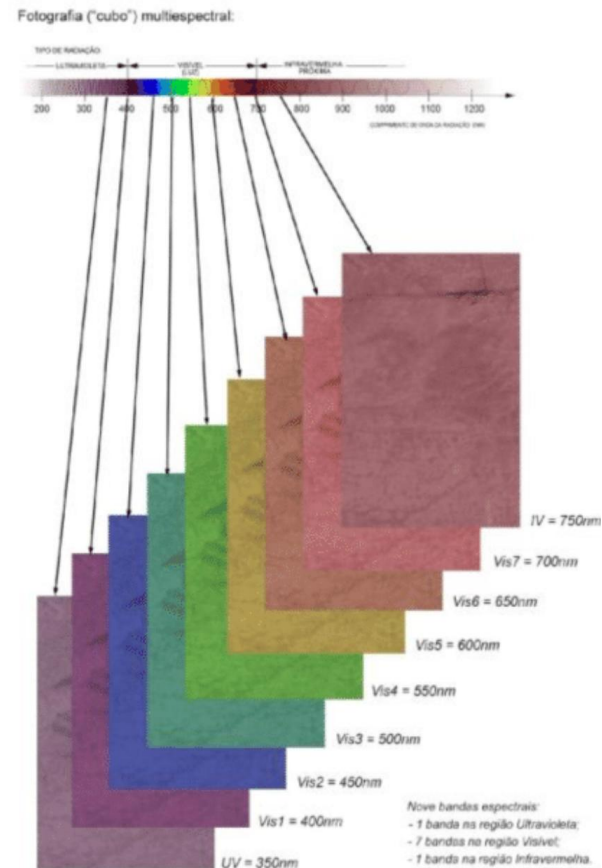
Biblioteca GDAL

- Biblioteca base de vários projetos (SIG e outras bibliotecas).
www.gdal.org
- Origem em 1998 (Frank Warmedam) evoluindo p/ pertencer a OSGeo.
- Começou como conversão de formatos de imagem, depois para vetor, e chegando a algoritmos de processamento.
 - Engloba outras bibliotecas.
- Biblioteca em C++, oferecendo “Binding” para Python.
- Alto desempenho.
- A biblioteca de Python é puramente um “Binding” não sendo “Pythônica Origem em 1998 (Frank Warmedam) evoluindo p/ pertencer a OSGeo”.

Biblioteca GDAL (Continuação)

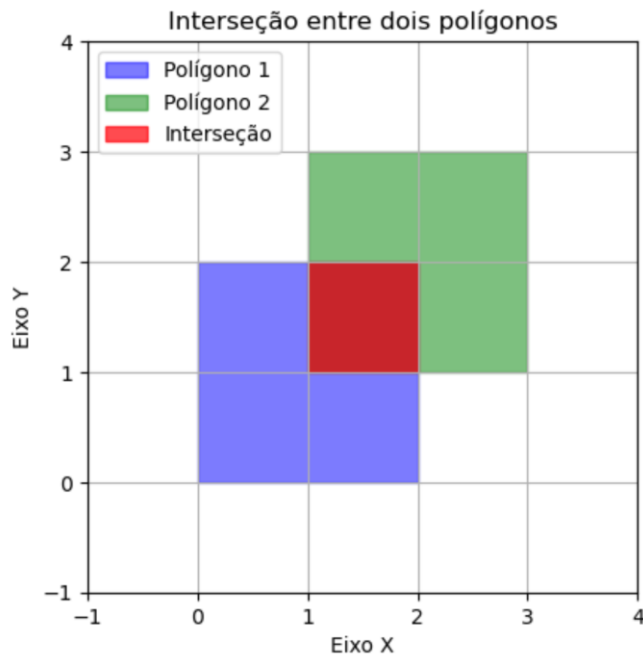


- Imagem com 3 bandas - Array 3D
 - 1ª index: Array 2D (Bandas)
 - 2ª index: Array 1D (Linhas)
 - 3ª index: Célula (Coluna)
 - Valor = Array(idx_band, idx_line, idx_column)



Outras bibliotecas - Shapely

- “Pythônica”.
- Utilizada pelo GeoPandas
- Criar (Polygon, Point, ...) e analisar geometria (Interseção, Diferença, ...).
- <https://shapely.readthedocs.io/en/stable/>



Outras bibliotecas - RasterIO

- Ler, escrever e manipular dados raster (imagem)
- <https://rasterio.readthedocs.io/en/stable/>

Imagem de Satélite (RGB)

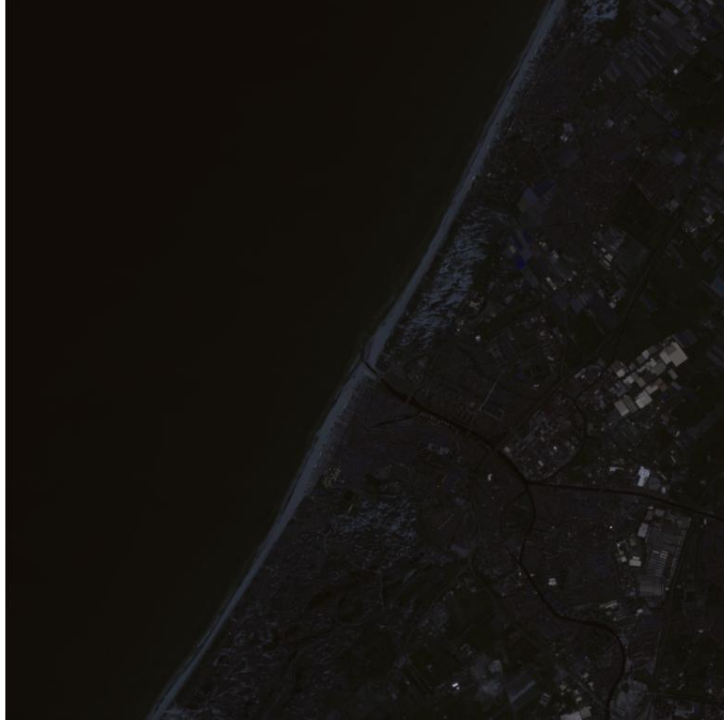
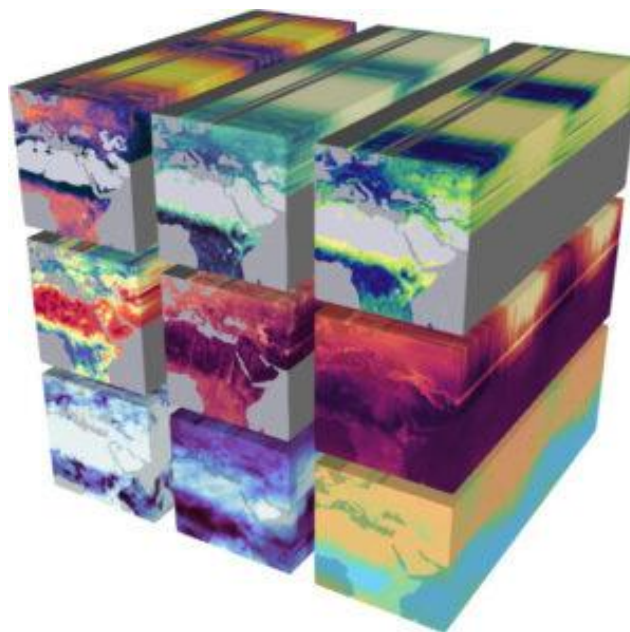


Imagem de Satélite (RGB)



Outras bibliotecas - XArray

- Trabalhar com dados multidimensionais (2D, 3D ou 4D)
- Uso em série histórica (alteração dos valores dos pixels ao longo do tempo)
- Dados climáticos e meteorológicos (NetCDF, GRIB)
- Séries temporais espaciais
 - Open Data Cube (ODC), conhecido como Data Cube, usa o Xarray
- <https://docs.xarray.dev/en/stable/>
- <https://www.opendatacube.org/>



GeoPandas

- Combina a funcionalidade do Pandas com bibliotecas de SIG
 - Lê arquivos geográficos como .shp, .geojson, .kml, **.gpkg**
 - Manipula geometrias (funções do Shapely)
 - Joins espaciais (dataframes)
- <https://geopandas.org/en/stable/index.html>

Bibliotecas de Mapas



Ipyleaflet:

- Base: Leaflet.js (via widgets do Jupyter)
- Interatividade: Alta (suporte a eventos, camadas, popups, widgets)
- Ideal para: Aplicações interativas em notebooks, integração com ipywidgets
- Destaques: Suporte a camadas vetoriais e raster, controle de camadas, integração com GeoDataFrame

<https://ipyleaflet.readthedocs.io/en/latest/>



folium

- Base: Leaflet.js (renderiza HTML/JS)
- Interatividade: Média (interativo no navegador, mas não reativo no notebook)
- Ideal para: Mapas rápidos e bonitos com pouco código
- Destaques: Fácil de usar, exporta mapas como HTML, bom para dashboards



keplergl

- Base: Kepler.gl (Uber)
- Interatividade: Muito alta (interface gráfica rica)
- Ideal para: Exploração visual de grandes volumes de dados geoespaciais
- Destaques: Visualizações 3D, animações temporais, integração com geopandas

Bibliotecas de Mapas (Cont.)



geemap

- Base: ipyleaflet + Google Earth Engine
- Interatividade: Alta
- Ideal para: Análise de dados ambientais e sensoriamento remoto
- Destaques: Acesso direto ao Google Earth Engine, ferramentas de análise espacial, visualização de imagens de satélite



geopandas + matplotlib

- Base: matplotlib
- Interatividade: Baixa (mapas estáticos)
- Ideal para: Análise exploratória e visualização simples
- Destaques: Integração com pandas, fácil de usar para análises vetoriais



plotly (com Mapbox)

- Base: Plotly.js + Mapbox
- Interatividade: Alta
- Ideal para: Dashboards interativos e visualizações web
- Destaques: Gráficos 3D, animações, integração com Dash