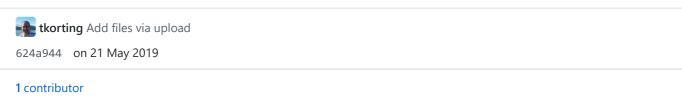
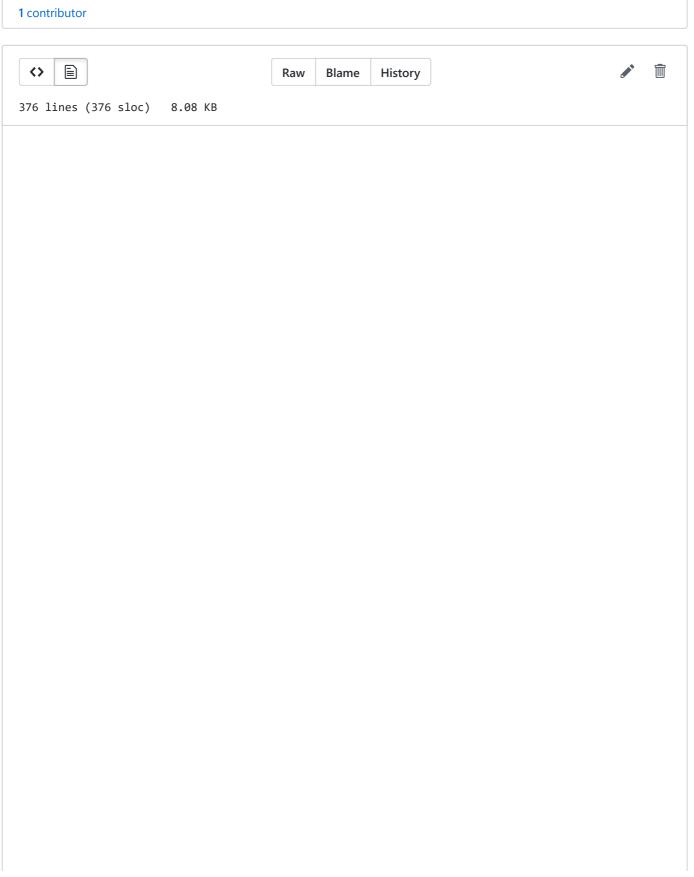
Branch: master ▼

Find file Copy path

pgser-ser347 / 2019 / aula-21-raster2vector.ipynb





Exemplo Contador de focos no pixel

Nesta aula será apresentado o processo de extração de valores de um dado matricial para criação de um arquivo CSV e posteriormente um dado vetorial.

Também será demonstrada a utilização de uso da biblioteca rasterio, Pyproj e Affine.

```
In [ ]: # Comando utilizado para instalação de uma biblioteca
        # Depois da instalação deve ser reiniciado o kernel do Jupyter Noteb
        #! conda install -y rasterio
In [ ]: # Importação das biblitecas necessárias
        import rasterio
        from rasterio.plot import show
        import numpy as np
        import os
        from affine import Affine
        from pyproj import Proj, transform
        %matplotlib inline
In [ ]: # Verificar diretorio de trabalho
        os.getcwd()
In [ ]: # Modificar o diretorio de trabalho para o local onde estão os dados
        de interesse
        os.chdir('/Users/fabianomorelli/ownCloud/notebooks/ipython_notebook
        s/dados_exemplo/')
In [ ]: # O dado a ser utilizado deve conter o arquivo de cabeçalho
        # neste caso o arquivo esperado é foc sev 20161010.hdr
        fname = './foc_sev_20161010.bin'
In [ ]: # Confirma se existe o arquivo
        os.path.exists(fname)
In [ ]: # Read raster
        with rasterio.open(fname) as r:
            geo_transfom = r.profile.get("transform") # upper-left pixel co
        rner affine transform
            projecao = r.profile.get("crs")
            arr_focos = r.read() # pixel values
```

Verificação do conteúdo das variáveis

In []: | r.profile

```
In [ ]: | geo_transfom
  In [ ]: projecao
Verificação do dado matricial armazenado dentro do array
  In [ ]: arr_focos
  In [ ]: type(arr_focos)
  In [ ]: arr_focos.shape
  In [ ]: arr_focos.max()
Operação para gerar uma lista de valores de linhas e colunas que possuem dados de focos
  In [ ]: listagem = np.where(arr_focos > 0)
  In [ ]: |listagem
  In [ ]: # apenas um exemplo de outro método para listar os píxeis diferente
            de zero
           arr_focos.nonzero()
  In [ ]: #tamanho das linhas
           len(listagem[1])
  In [ ]: | # Cria uma função para fazer a conversao linha,coluna para lat,lon
           def lc2ll(colula,linha):
                lonlat = (colula,linha) * geo_transfom
                return lonlat
           # CRIAR UMA FUNCAO PARA RETORNAR O VALOR DO PIXEL
           deg getValue(banda, linha, coluna, matriz):
                n = matriz[banda, linha, coluna]
                return n
  In [ ]: # Exemplo de uso da função
           1c211(0,0)
  In [ ]: # Exemplo que cria um objeto da inversão da matriz de afine
           # para converter de latitude/longitude para linha/coluna
           rev = ~geo_transfom
  In [ ]: # Exemplo de conversão
           rev * (-83.005, 13.005)
Processamento para criação de um arquivo CSV
  In [ ]: # Loop de exemplo para verificar o valor de cada píxel
           for i in np.arange(len(listagem[1])):
```

--:-+ /: 1-211/1:-+----[21[:1

```
In []: # Processamento completo

# Abertura de um arquivo para escrita e gravacao da primeira linha
f = open('/tmp/myfile', 'w')
f.write(str("id,longitude,latitude\n"))

# Loop para escrever cada linha do arquivo destino
for i in np.arange(len(listagem[1])):
    latlon = lc2ll(listagem[2][i], listagem[1][i])
```