projeto

June 23, 2024

1 Imports

```
[1]: #from src import␣
      ↪Build_mc,Upload_geof,Upload_geof_geografica,Upload_litologia,plot_filtered_values,plot_raw_m
      ↪plotBoxplots,remove_negative_values,sintetic_grid,traditional_interpolation,plot_interpolate
      from src import *
      from pylab import *

      from tqdm import tqdm
      from shapely.ops import transform
      from shapely.geometry import Point, Polygon

      import matplotlib.pyplot as plt
      import pandas as pd
      import geopandas as gpd
      import verde as vd
      import numpy as np
      import os
      import pyproj
      import verde_source as vds

      import seaborn as sns
      import xarray as xr
      import matplotlib

      from sklearn_som.som import SOM
      from sklearn.preprocessing import StandardScaler
```

```
[2]: import warnings
      warnings.filterwarnings("ignore")
      %matplotlib widget
      %load_ext autoreload
      %autoreload 2
      %reload_ext autoreload
```

```
[3]: '''def sintetic_grider(quadricula=None,p_size=None):
          ids = list(quadricula.keys())
```

```
    wgs84=pyproj.CRS('EPSG:4326')
    for id in tqdm(ids):
        folha = quadricula[id]['folha']
        utm = pyproj.CRS('EPSG:'+folha['EPSG'])
        carta_geografica = folha['geometry']
        project = pyproj.Transformer.from_crs(wgs84,utm, always_xy=True).
 ↪transform
        carta_utm = transform(project,carta_geografica)
        break
'''
```

[3]:
```
"def sintetic_grider(quadricula=None,p_size=None):\n    ids =
list(quadricula.keys())\n    wgs84=pyproj.CRS('EPSG:4326')\n    for id in
tqdm(ids):\n        folha = quadricula[id]['folha']\n        utm =
pyproj.CRS('EPSG:'+folha['EPSG'])\n        carta_geografica =
folha['geometry']\n        project = pyproj.Transformer.from_crs(wgs84,utm,
always_xy=True).transform\n        carta_utm =
transform(project,carta_geografica)\n        break\n"
```

1.1 Construindo Quadrícula

cula|

cula|

cula| |

[breakable,   size=fbox,   boxrule=1pt,   pad   at   break*=1mm,colback=cellbackground,
colframe=cellborder]incolor[3]: -   [commandchars=\80\173\80\175]   quadricula   =
Build_mc(escala=\80\047100k\80\047,ID=[\80\047SF23_VD\80\047,\80\047SF23_VC\80\047,\80\047SF23_`

{}

b |
| 0/4 [00:00<?, ?it/s]

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/tmp/ipykernel_4181/2212415174.py in ?()
----> 1 quadricula =␣
 ↪Build_mc(escala='100k',ID=['SF23_VD','SF23_VC','SF23_YA','SF23_YB'],verbose=Tr e)

~/projetos/geologist/src.py in ?(escala, ID, verbose)
    779 def Build_mc(escala='50k',ID=['SF23_YA'],verbose=None):
```

```
--> 780        mc = import_mc(escala,ID)
    781        mc.set_index('id_folha',inplace=True)
    782        quadricula = {}
    783        wgs84 = pyproj.CRS('EPSG:4326')

~/projetos/geologist/src.py in ?(escala, ID)
     83        mc = gpd.read_file(set_gdb('geodatabase.
 ↪gpkg'),driver='GPKG',layer='mc_'+escala)
     84        mc_slct = gpd.GeoDataFrame()
     85        if type(ID) == list:
     86            for id in tqdm(ID):
---> 87                mc_slct = mc_slct.append(mc[mc['id_folha'].str.contains(id)])
     88            return mc_slct
     89        elif type(ID) == str:
     90            mc_slct = mc[mc['id_folha'] == ID]

~/.config/ambiente_geologico/lib/python3.11/site-packages/pandas/core/generic.py
 ↪in ?(self, name)
   5985                and name not in self._accessors
   5986                and self._info_axis._can_hold_identifiers_and_holds_name(name)
   5987            ):
   5988                return self[name]
-> 5989            return object.__getattribute__(self, name)

AttributeError: 'GeoDataFrame' object has no attribute 'append'
```

```python
list_ids = list(quadricula.keys())
#print(list_ids)

'''
for id in list_ids:
    print(f' - Folha: {id}')
    carta=quadricula[id]
    data_list = list(carta.keys())
    print(data_list)
    for data in data_list[4:]:
        print(f'    - {data}')
        print(f'        - {list(quadricula[id][data].columns)}')
'''
def print_quadriculas(quadricula=quadricula):
    list_ids = list(quadricula.keys())
    for id in list_ids:
        print(f' - Folha {id}')
        carta = quadricula[id]
        list_carta_keys = list(carta.keys())
        print(f'    - {list_carta_keys}:')
```