

Chapter 6 - Spatial autocorrelation

November 10, 2020

This notebook generates measures spatial autocorrelation of morphometric characters. Also used to generate figure 12.

```
[3]: import matplotlib.pyplot as plt
import geopandas as gpd
import esda
import libpysal
from spplot.esda import plot_local_autocorrelation
import multiprocessing
import pandas as pd
```

```
[4]: esda.__version__, gpd.__version__, libpysal.__version__, pd.__version__
```

```
[4]: ('2.2.1', '0.7.0', '4.2.2', '1.0.3')
```

```
[ ]: blg = gpd.read_file('data/cadastre/blg_cadvals.shp')

characters = ['area', 'lal', 'circom', 'shapeix', 'rectan', 'fractal',
              'orient', 'freq', 'car', 'gini_area', 'gini_car', 'Reach']

singleuids = pd.read_csv('data/single_uids.csv')
singles = singleuids['2'].to_list()

def worker(k):
    print(k)
    file = gpd.read_file('data/tessellation/{k}_tessellation.shp'.format(k=k))
    file_s = file.loc[file['uID'].isin(singles)].copy()
    file_m = file.loc[~file['uID'].isin(singles)].copy()
    for ch in characters:
        print(ch)
        try:
            local_moran = esda.Moran_Local(file[[ch]], weights)
            file['m_{}'.format(ch)] = local_moran.q
            file['p_{}'.format(ch)] = local_moran.p_sim
            single_moran = esda.Moran_Local(file_s[[ch]], weights_s)
            file.loc[file['uID'].isin(singles), 'ms_{}'.format(ch)] = single_moran.q
        except:
            pass
```

```

        file.loc[file['uID'].isin(singles), 'ps_{}'.format(ch)] =
↪single_moran.p_sim
        multi_moran = esda.Moran_Local(file_m[[ch]], weights_m)
        file.loc[~file['uID'].isin(singles), 'mm_{}'.format(ch)] =
↪multi_moran.q
        file.loc[~file['uID'].isin(singles), 'pm_{}'.format(ch)] =
↪multi_moran.p_sim
        # plot_local_autocorrelation(local_moran, file, ch)
        # plt.savefig('files/moran/{k}_{ch}.png'.format(k=k, ch=ch))
        # plt.gcf().clear()
    except Exception:
        print('missing, skipped')
    file.to_file('data/tessellation/{k}_tessellation.shp'.format(k=k))
    print('saved')

weights = libpysal.weights.DistanceBand.from_dataframe(blg, 200)
single = blg.loc[blg['uID_left'].isin(singles)].copy()
weights_s = libpysal.weights.DistanceBand.from_dataframe(single, 200)
multi = blg.loc[~blg['uID_left'].isin(singles)].copy()
weights_m = libpysal.weights.DistanceBand.from_dataframe(multi, 200)

for ch in characters:
    print(ch)
    local_moran = esda.Moran_Local(blg[[ch]], weights)
    blg['m_{}'.format(ch)] = local_moran.q
    blg['p_{}'.format(ch)] = local_moran.p_sim
    single_moran = esda.Moran_Local(single[[ch]], weights_s)
    blg.loc[blg['uID_left'].isin(singles), 'ms_{}'.format(ch)] = single_moran.q
    blg.loc[blg['uID_left'].isin(singles), 'ps_{}'.format(ch)] = single_moran.
↪p_sim
    multi_moran = esda.Moran_Local(multi[[ch]], weights_m)
    blg.loc[~blg['uID_left'].isin(singles), 'mm_{}'.format(ch)] = multi_moran.q
    blg.loc[~blg['uID_left'].isin(singles), 'pm_{}'.format(ch)] = multi_moran.
↪p_sim
    blg.to_file('data/cadastre/blg_cadvals.shp')

```