Spatial Stats of Raster Upon Polygons

By Dr. Chonghua Yin

08/Sep/2023

One of the most prevalent tasks for developers of spatial analysis applications is extracting raster values that overlap with vector data, followed by subsequent spatial statistical analysis. For example, under climate change, banks or insurance companies are eager to assess the risk of their holdings or insured properties being flooded in small-scale areas (defined by polygons) (Figure 1).

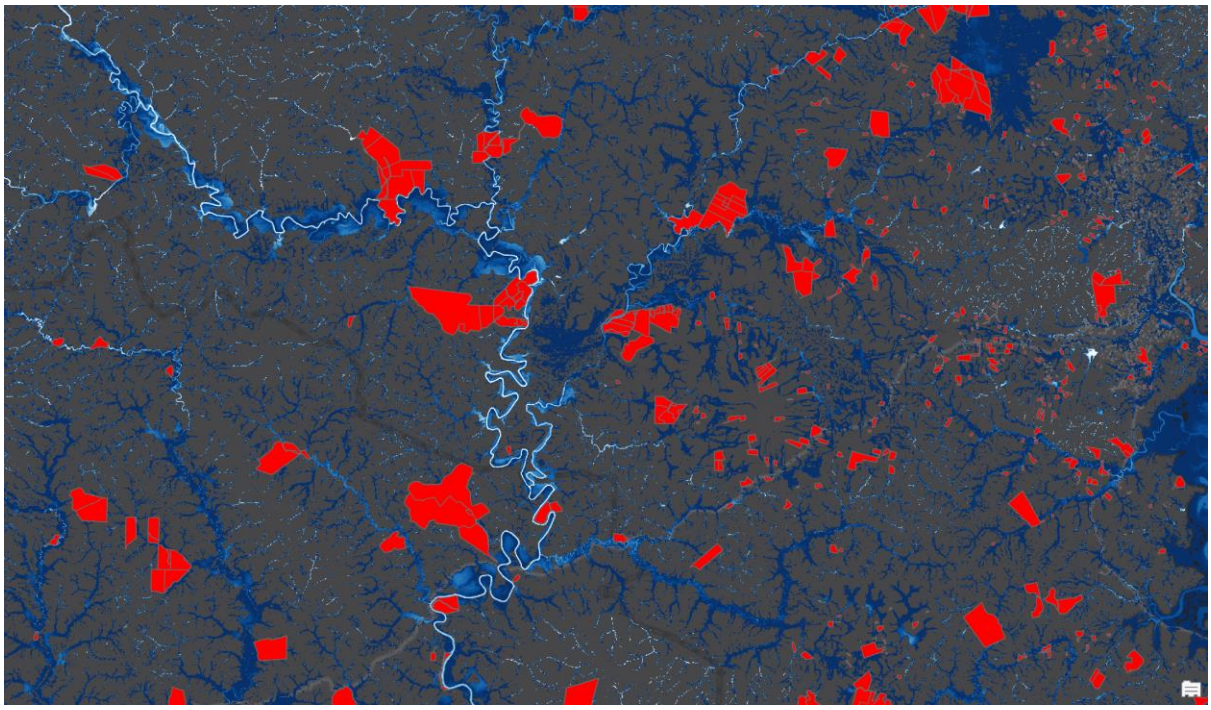

Figure 1. Assets in red under the risk of flood (map created by the author)

The conventional method involves the clipping of raster data with each asset polygon and then performing statistical analysis individually (e.g., using rasterstats, rioxarray, salem, xarray, geopandas, etc.). This method is intuitive and easy to understand. However, this approach will become highly inefficient when there are many polygons (e.g., over hundreds of thousands). As a result, some people have considered using a parallel approach, where asset polygons are divided into several groups, and extraction and statistics are performed on each group separately (e.g., Dask-GeoPandas). This approach indeed increases processing speed, but its essence remains iterative, albeit with grouping, so the speed improvement is still relatively limited when dealing with many polygons.

We can contemplate a reverse approach as well. Why not digitize the asset polygons directly into raster data with a style closely mirroring the target raster data (e.g., the flood raster in Figure 1), including the same spatial domain and resolution? By pursuing this strategy, we can leverage a distinct field within the property polygons as a key to pinpoint the location of each asset polygon. This process effectively generates a mask matrix to extract data from the target raster and conduct subsequent

statistical analyses (e.g., Figure 2). The tools to rasterize polygons include rasterio, regionmask, geocube, superstar gdal, etc.

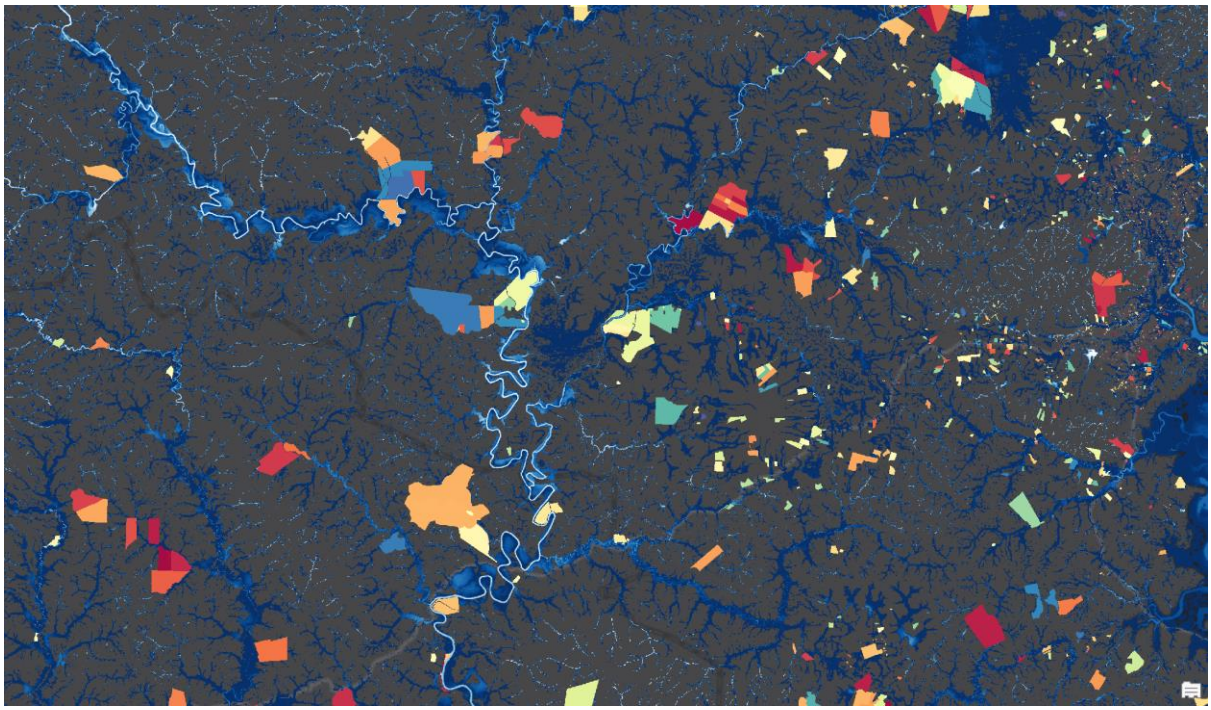

Figure 2. Assets masks overlapped on a flood map (map created by the author)

Indeed, there are other choices available. We can employ specialized spatial analysis software to finish such kinds of tasks. For instance, ArcGIS Pro provides the ZonalStatisticsAsTable function, or you can utilize QgsZonalStatistics within QGIS. It's important to note that this approach necessitates the installation of these software packages to access their functionalities. This might not align with your preferences, especially if you seek to utilize lightweight scripts exclusively. Read the technical document about ArcGIS Pro's ZonalStatisticsAsTable. You'll notice that it initiates the rasterization of polygons to generate masking arrays, essentially sharing the same concept as the second method.

Of course, each approach has its strengths and weaknesses. If you have a relatively small amount of polygonal data, you may opt for the first method. However, the second method may be more suitable for dealing with many polygons. The first two methods provide greater flexibility and versatility. On the other hand, if your analysis workflow is closely tied to a specific software, such as ArcGIS or QGIS, then it's advisable to consider the third approach. This ensures compatibility with your software and can streamline your analysis process within that particular environment.

References

https://pythonhosted.org/rasterstats/

https://corteva.github.io/rioxarray/stable/

https://docs.xarray.dev/en/stable/

https://salem.readthedocs.io/en/stable/

https://geopandas.org/en/stable/index.html

https://dask-geopandas.readthedocs.io/en/stable/

https://regionmask.readthedocs.io/en/stable/

https://rasterio.readthedocs.io/en/stable/#

https://gdal.org/index.html

https://corteva.github.io/geocube/stable/index.html

https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/zonal-statistics-as-table.htm

https://qgis.org/pyqgis/master/analysis/QgsZonalStatistics.html