HUC_Func

January 25, 2024

[1]:
```
##Import libraries
import pandas
import numpy
import geopandas
import rasterio
import xarray
import rioxarray
```

0.1 Read files

[2]:
```
##Shapefile with HUC regions
regionsfilepath = "/home/jesse-ubuntu/Documents/DrainageDitches/WBDHU10_MN.shp"
regions = geopandas.read_file(regionsfilepath)

##Raster file with terrain data
demfilepath = '/home/jesse-ubuntu/Documents/DrainageDitches/N44W094.hgt'
dem = rioxarray.open_rasterio(demfilepath).squeeze()
```

0.2 Reproject to common CRS

[3]:
```
#Raster file already in this CRS
regions=regions.to_crs(4326)
```

0.3 Clip shapefile to raster size

[4]:
```
#Get a subset of the region geodataframe with the raster bounds
regions_ss = geopandas.clip(regions, (-94.0, 44.0, -93.0, 45.0), False)
```

0.4 Create function that takes a shapefile and raster with the same size and CRS then clips and saves a raster for each shapefile feature

[5]:
```
def superclip(gdf, raster):
    ##Loop through the rows in the geodataframe(shapefile)
    for i in range(len(gdf)):
        ##Make a geodataframe for each row
```

```
        ##I tried without this step by just using gdf.iloc[[i]] in place of␣
↪rowGdf and rowGdf.iloc[0], but I had some trouble with extra text in the␣
↪strings
        rowGdf = gdf.iloc[[i]]

        ##Clip out each of the shapes
        raster_clip = raster.rio.clip(rowGdf.geometry.values, rowGdf.crs,␣
↪drop=True, invert=False)

        ##I had problems where a rectangle would be created from the clip and␣
↪everything outside the shape would be super low number for nodata, this␣
↪should fix that
        nodata = raster_clip.rio.nodata
        raster_clip = raster_clip.where(raster_clip != nodata)

        ##Get name of watershed and HUC10 number for file name
        filepath = r"/home/jesse-ubuntu/Documents/DrainageDitches/"+rowGdf.
↪iloc[0]['name']+'_'+rowGdf.iloc[0]['huc10']+'.tif'
        ##Save to filepath
        raster_clip.rio.to_raster(filepath)
```

0.5 Run the function with your geodataframe and dem!

[6]:
```
superclip(regions_ss, dem)
```