Advanced GIS

Lab 8

Xin Ding

05/23/2018

In this lab Assignment, I used data from the City of Chicago data portal and Instructor in class 4 to study the Jail locations in the city of Chicago.

GIS Functions included:

Reprojection, Spatial Join, Centroid, Buffer, Erase(did not work) and Envelope

Cursor Function:

Cursor functions are only available in arcpy, which is not available for me since I use Linux for my works. So I wrote a function to locate information within the GeoDataFrame by with pandas.

Major products:

Number of Crimes in the city of Chicago in different community area in a choropleth map,

An interactive version of such map,

Percent Aged 16 or plus unemployed in different community area in a choropleth map,

Number of Jails within each community area,

Centroid of community areas,

500-meter buffer for the jails,

Envelope for the community area.

And finally, setting proper projection for each GeoDataFrame and export to shapefiles for ArcGIS Online

The procedure for analyzing jails locations:

Find out the locations of jails with respect to different community areas in the city of Chicago =>

Find out the distribution of jails within different community areas in Chicago =>

Find out areas with no jail in the vicinity (by creating a 500 meter buffer and erase it from the City of Chicago)

Last but not least, find out jails located in the Hyde Park area.

Major Code:

import pandas as pd

import json

import requests

from urllib.parse import quote

%matplotlib inline

soql = "https://data.cityofchicago.org/resource/6zsd-86xi.json?$query="

soql += quote("SELECT community\_area AS Community, count(\*)/5 AS No\_Crimes, primary\_type AS Primary\_Type ")

soql += quote("WHERE 2007 < year AND year < 2013 ")

soql += quote("GROUP BY Community, Primary\_Type LIMIT 10000")

resp = requests.get(soql).json()

crime\_type = pd.DataFrame(resp)

import fiona; fiona.supported\_drivers

socioeconomics = pd.read\_csv("../../data/Census\_Data\_-\_Selected\_socioeconomic\_indicators\_in\_Chicago\_\_2008\_\_\_2012.csv")

crime\_type.rename(columns={'Community':'Community Area Number'}, inplace = True)

crime\_type['Community Area Number'] = crime\_type['Community Area Number'].astype(float)

crime\_df = pd.DataFrame.merge(crime\_type, socioeconomics, on = 'Community Area Number')

import matplotlib.pyplot as plt

import geopandas as gpd

chicomm\_df = gpd.read\_file("../../data/chicomm/chicomm.shp")

chicomm\_df.rename(columns = {'CHICOMNO':'Community Area Number'}, inplace = True)

chicomm\_df.set\_index('Community Area Number', inplace = True)

chicomm\_df.index = chicomm\_df.index.astype(int)

chicomm\_df.sort\_index(inplace = True)

chicomm\_df.plot()

crime\_sum = crime\_df.groupby('Community Area Number').sum()

crime\_sum.index = crime\_sum.index.astype(int)

crime\_sum\_geo = chicomm\_df.join(crime\_sum)

df\_4 = gpd.read\_file('../advanced\_gis/inclassdata\_unit4.gdb')

df\_4\_proj = df\_4.to\_crs({'init': 'epsg:4269'})

crime\_sum\_geo\_proj = crime\_sum\_geo.to\_crs({'init': 'epsg:26971'})

crime\_sum\_geo\_proj.plot()

crime\_sum\_geo\_proj.to\_file("crime\_sum\_geo\_proj.shp")

ax\_sum = crime\_sum\_geo.plot(column = 'PERCENT AGED 16+ UNEMPLOYED', cmap = 'plasma', scheme = 'equal\_interval',

k = 10, linewidth = 2,

legend = True, figsize = (10, 10),

)

ax\_sum.set\_title("PERCENT AGED 16+ UNEMPLOYED", fontsize = 20, y = 1.05)

ax\_sum.get\_legend().set\_bbox\_to\_anchor((0.3, 0.3))

ax\_sum.get\_legend().get\_frame().set\_linewidth(0)

ax\_sum.set\_axis\_off()

Jails\_within\_community\_area = gpd.sjoin(df\_4, crime\_sum\_geo\_proj, how="inner", op='intersects')

Jails\_within\_community\_area.DISTITLE.value\_counts()

crime\_sum\_geo\_proj.centroid.plot()

crime\_sum\_geo\_proj\_centroid = crime\_sum\_geo\_proj.centroid

crime\_sum\_geo\_proj\_centroid.to\_file("crime\_sum\_geo\_proj\_centroid.shp")

df\_4\_buffer = df\_4.buffer(500)

df\_4\_buffer.plot(figsize = (15, 15))

df\_4\_buffer = gpd.GeoDataFrame({'geometry': df\_4\_buffer})

df\_4\_buffer.crs = {'init': 'epsg:26971'}

df\_4\_buffer.to\_file("df\_4\_buffer.shp")

gpd.read\_file("df\_4\_buffer.shp").plot()

crime\_sum\_geo\_proj.envelope.plot(figsize = (15, 15))

crime\_sum\_geo\_proj\_envelope = crime\_sum\_geo\_proj.envelope

crime\_sum\_geo\_proj\_envelope = gpd.GeoDataFrame({'geometry': crime\_sum\_geo\_proj\_envelope})

crime\_sum\_geo\_proj\_envelope.crs = {'init': 'epsg:26971'}

crime\_sum\_geo\_proj\_envelope.to\_file("crime\_sum\_geo\_proj\_envelope.shp")

hyde\_park = []

for i in range(len(Jails\_within\_community\_area)):

if Jails\_within\_community\_area.iloc[i, 6] == 'Hyde Park':

hyde\_park.append(Jails\_within\_community\_area.iloc[i, 3])

hyde\_park = gpd.GeoDataFrame({'geometry': hyde\_park})

hyde\_park.crs = {'init': 'epsg:26971'}

chicomm\_df\_proj = chicomm\_df.to\_crs({'init': 'epsg:26971'})

hyde\_park\_context = gpd.sjoin(hyde\_park, chicomm\_df\_proj, how="inner", op='intersects')

ax = chicomm\_df\_proj.plot(figsize = (15, 15))

hyde\_park\_context.plot(ax = ax, color = "yellow")

def search\_cursor(jailid, col):

return Jails\_within\_community\_area.loc[jailid, col]