This is a processing script to aggregate MIT's Election Data for United States presidental election at the state and county levels. I use this data for teaching an Analysis in GIS course at Virginia Tech.

In [90]:
```
import pandas as pd
import numpy as np

import geopandas
```

County Election Data

In [91]:
```
mit_data = pd.read_csv('original_data/countypres_2000-2020.csv',dtype={'county_fips':str})
mit_data = mit_data.rename(columns={'county_fips':'FIPS'})
mit_data = mit_data[~mit_data['FIPS'].isnull()]
mit_data['FIPS'] = mit_data['FIPS'].str.zfill(5)
```

Data Repair: Not all counties have vote totals, so calculate new vote totals based on candidatevotes

2000: North Carolina, Oklahoma; 2004: Oklahoma

In [92]:
```
grp = mit_data.groupby(by=['year','FIPS']).sum().reset_index()
grp = grp.drop(labels=['totalvotes','version'],axis=1)
grp = grp.rename(columns={'candidatevotes':'totalvotes2'})
mit_data = mit_data.merge(grp,on=['year','FIPS'])

mit_data['totalvotes'] = mit_data['totalvotes2']
mit_data = mit_data.drop(labels=['totalvotes2'],axis=1)
print(mit_data.head())
```

```
   year    state state_po county_name   FIPS        office       candidate  \
0  2000  ALABAMA       AL     AUTAUGA  01001  US PRESIDENT         AL GORE
1  2000  ALABAMA       AL     AUTAUGA  01001  US PRESIDENT  GEORGE W. BUSH
2  2000  ALABAMA       AL     AUTAUGA  01001  US PRESIDENT     RALPH NADER
3  2000  ALABAMA       AL     AUTAUGA  01001  US PRESIDENT           OTHER
4  2000  ALABAMA       AL     BALDWIN  01003  US PRESIDENT         AL GORE

        party  candidatevotes  totalvotes   version   mode
0    DEMOCRAT            4942       17208  20220315  TOTAL
1  REPUBLICAN           11993       17208  20220315  TOTAL
2       GREEN             160       17208  20220315  TOTAL
3       OTHER             113       17208  20220315  TOTAL
4    DEMOCRAT           13997       56480  20220315  TOTAL
C:\Users\Thomas Pingel\AppData\Local\Temp\ipykernel_33132\2106145405.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  grp = mit_data.groupby(by=['year','FIPS']).sum().reset_index()
```

Data Repair: Reclassify Shannon County FIPS as Oglala Lakota County FIPS

In [93]:
```
mit_data.loc[mit_data['FIPS']=='46113','FIPS'] = '46102'
```

Data Repair: Some counties in 2020 list separate tallies for different kinds of ballots

In [94]:
```
mit_data.loc[mit_data.county_name=='MANASSAS PARK CITY']
```

Out[94]:

	year	state	state_po	county_name	FIPS	office	candidate	party	candidatevotes	totalvotes	versi
71706	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	JOSEPH R BIDEN JR	DEMOCRAT	3137	6088	202203
71707	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	JOSEPH R BIDEN JR	DEMOCRAT	834	6088	202203
71708	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	JOSEPH R BIDEN JR	DEMOCRAT	21	6088	202203
71709	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	JO JORGENSEN	LIBERTARIAN	61	6088	202203
71710	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	JO JORGENSEN	LIBERTARIAN	38	6088	202203
71711	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	JO JORGENSEN	LIBERTARIAN	1	6088	202203
71712	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	OTHER	OTHER	10	6088	202203
71713	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	OTHER	OTHER	7	6088	202203
71714	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	OTHER	OTHER	0	6088	202203
71715	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	DONALD J TRUMP	REPUBLICAN	1239	6088	202203
71716	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	DONALD J TRUMP	REPUBLICAN	733	6088	202203
71717	2020	VIRGINIA	VA	MANASSAS PARK CITY	51685	US PRESIDENT	DONALD J TRUMP	REPUBLICAN	7	6088	202203

In [95]:
```
# Pull just 2020
df = mit_data[mit_data['year'] == 2020].copy()

# Add a field to count, when we sum this, it will tell us how many rows there were
df['count'] = 1

# Sum the votes by fips code and candidate
group = df.groupby(by=['FIPS','candidate'])
out = group.sum().reset_index()
out['totalvotes'] = out['totalvotes'] / out['count']
out = out.drop(columns=['count','version'])
out['year'] = 2020
```

```
C:\Users\Thomas Pingel\AppData\Local\Temp\ipykernel_33132\3458510559.py:9: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  out = group.sum().reset_index()
```

In [96]:
```
out.loc[out.FIPS=='51685']
```

Out[96]:

	FIPS	candidate	year	candidatevotes	totalvotes
11070	51685	DONALD J TRUMP	2020	1979	6088.0
11071	51685	JO JORGENSEN	2020	100	6088.0
11072	51685	JOSEPH R BIDEN JR	2020	3992	6088.0
11073	51685	OTHER	2020	17	6088.0

In [97]:
```
mit_data = mit_data[mit_data['year'] != 2020]
mit_data = mit_data.append(out)

mit_data.tail()
```

```
C:\Users\Thomas Pingel\AppData\Local\Temp\ipykernel_33132\1241646329.py:2: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  mit_data = mit_data.append(out)
```

Out[97]:

	year	state	state_po	county_name	FIPS	office	candidate	party	candidatevotes	totalvotes	version	mode
11893	2020	NaN	NaN	NaN	56043	NaN	OTHER	NaN	71	4032.0	NaN	NaN
11894	2020	NaN	NaN	NaN	56045	NaN	DONALD J TRUMP	NaN	3107	3560.0	NaN	NaN
11895	2020	NaN	NaN	NaN	56045	NaN	JO JORGENSEN	NaN	46	3560.0	NaN	NaN
11896	2020	NaN	NaN	NaN	56045	NaN	JOSEPH R BIDEN JR	NaN	360	3560.0	NaN	NaN
11897	2020	NaN	NaN	NaN	56045	NaN	OTHER	NaN	47	3560.0	NaN	NaN

Continue with data processing

In [98]:
```
presidential_candidates = {2000:{'gop':'GEORGE W. BUSH','dem':'AL GORE'},
                           2004:{'gop':'GEORGE W. BUSH','dem':'JOHN KERRY'},
                           2008:{'gop':'JOHN MCCAIN','dem':'BARACK OBAMA'},
                           2012:{'gop':'MITT ROMNEY','dem':'BARACK OBAMA'},
                           2016:{'gop':'DONALD TRUMP','dem':'HILLARY CLINTON'},
                           2020:{'gop':'DONALD J TRUMP','dem':'JOSEPH R BIDEN JR'}
                           }
```

In [99]:
```
output_df = pd.DataFrame()
output_df['FIPS'] = mit_data['FIPS'].unique()

years = np.sort(list(presidential_candidates.keys()))

for year in years:
    # Pull this year as a dataframe, pull this year's candidates, and
    # convert year to a string, since it will now be used to name fields
    df=mit_data[mit_data['year']==year]
    candidates = presidential_candidates[year]
    year = str(year)

    # Get candidate info for this year, rename
    gop = df.candidate == candidates['gop']
    gop = df.loc[gop,['FIPS','candidatevotes']]
    gop = gop.rename(columns={'candidatevotes':'gop' + '_' + year + '_votes'})
    dem = df.candidate == candidates['dem']
    dem = df.loc[dem,['FIPS','candidatevotes','totalvotes']]
    dem = dem.rename(columns={'candidatevotes':'dem' + '_' + year + '_votes'})
    dem = dem.rename(columns={'totalvotes':'totalvotes' + '_' + year})

    # Write this information to the output dataframe and calculate some fields
    output_df = output_df.merge(gop,on='FIPS',how='left')
    output_df = output_df.merge(dem,on='FIPS',how='left')
    output_df['gop_' + year + '_prc'] = np.round(100 * output_df['gop_' + year + '_votes'] / output_df['tota
    output_df['dem_' + year + '_prc'] = np.round(100 * output_df['dem_' + year + '_votes'] / output_df['tota
    output_df['gop_minus_dem_prc_' + year] = output_df['gop_' + year + '_prc'] - output_df['dem_' + year + '

output_df.head()
```

Out[99]:

	FIPS	gop_2000_votes	dem_2000_votes	totalvotes_2000	gop_2000_prc	dem_2000_prc	gop_minus_dem_prc_2000	gop_2004_
0	01001	11993.0	4942.0	17208.0	69.69	28.72	40.97	
1	01003	40872.0	13997.0	56480.0	72.37	24.78	47.59	
2	01005	5096.0	5188.0	10395.0	49.02	49.91	-0.89	
3	01007	4273.0	2710.0	7101.0	60.17	38.16	22.01	
4	01009	12667.0	4977.0	17973.0	70.48	27.69	42.79	

5 rows × 37 columns

In [100…]:
```
# Fix for [Kalawao County, Hawaii](https://en.wikipedia.org/wiki/Kalawao_County,_Hawaii), which doesn't have
# GEOID = 15005
# https://en.wikipedia.org/wiki/Kalawao_County,_Hawaii#Politics
# gop_minus_dem_prc_2020 = 91.66 with 24 total votes (of which 1 was republican)

d = {'FIPS':['15005'],'gop_2020_votes':[1],'dem_2020_votes':[23],'totalvotes_2020':[24],
                      'gop_2016_votes':[1],'dem_2016_votes':[14],'totalvotes_2016':[20],
                      'gop_2012_votes':[2],'dem_2012_votes':[25],'totalvotes_2012':[27],
                      'gop_2008_votes':[6],'dem_2008_votes':[24],'totalvotes_2008':[31],
                      'gop_2004_votes':[14],'dem_2004_votes':[26],'totalvotes_2004':[40],
                      'gop_2000_votes':[11],'dem_2000_votes':[30],'totalvotes_2000':[45],}

new_row = pd.DataFrame(d)

for year in [2000,2004,2008,2012,2016,2020]:
    year = str(year)
    new_row['gop_' + year + '_prc'] = np.round(100 * new_row['gop_' + year + '_votes'] / new_row['totalvotes
    new_row['dem_' + year + '_prc'] = np.round(100 * new_row['dem_' + year + '_votes'] / new_row['totalvotes
    new_row['gop_minus_dem_prc_' + year] = new_row['gop_' + year + '_prc'] - new_row['dem_' + year + '_prc']

output_df = pd.concat([output_df, new_row], ignore_index=True)
```

In [101…]:
```
output_df.to_csv('county_election_data_2000-2020.csv',index=False,float_format='%.2f')
```

State Election Data

In [12]:
```
mit_data = pd.read_csv('original_data/1976-2020-president.csv',dtype={'state_fips':str})
mit_data = mit_data.rename(columns={'state_fips':'FIPS'})
mit_data = mit_data[~mit_data['FIPS'].isnull()]
mit_data['FIPS'] = mit_data.FIPS.str.zfill(2)
```

In [13]:
```
presidential_candidates = {1976:{'gop':'FORD, GERALD','dem':'CARTER, JIMMY'},
                           1980:{'gop':'REAGAN, RONALD','dem':'CARTER, JIMMY'},
                           1984:{'gop':'REAGAN, RONALD','dem':'MONDALE, WALTER'},
                           1988:{'gop':'BUSH, GEORGE H.W.','dem':'DUKAKIS, MICHAEL'},
                           1992:{'gop':'BUSH, GEORGE H.W.','dem':'CLINTON, BILL'},
                           1996:{'gop':'DOLE, ROBERT','dem':'CLINTON, BILL'},
                           2000:{'gop':'BUSH, GEORGE W.','dem':'GORE, AL'},
                           2004:{'gop':'BUSH, GEORGE W.','dem':'KERRY, JOHN'},
                           2008:{'gop':'MCCAIN, JOHN','dem':'OBAMA, BARACK H.'},
                           2012:{'gop':'ROMNEY, MITT','dem':'OBAMA, BARACK H.'},
                           2016:{'gop':'TRUMP, DONALD J.','dem':'CLINTON, HILLARY'},
                           2020:{'gop':'TRUMP, DONALD J.','dem':'BIDEN, JOSEPH R. JR'}
                           }
```

In [14]:
```
# Mitt Romney's name is reversed for Washington for 2012.  This has been fixed in previous versions, but it'
# left here for instructional purposes:

idx = (mit_data['state_po']=='WA') & (mit_data['year']==2012) & (mit_data['party_detailed']=='REPUBLICAN')
mit_data[idx]
```

Out[14]:

	year	state	state_po	FIPS	state_cen	state_ic	office	candidate	party_detailed	writein	candidatevotes	t
3371	2012	WASHINGTON	WA	53	91	73	US PRESIDENT	MITT, ROMNEY	REPUBLICAN	False	1290670	

In [15]:
```
# Perform the fix
mit_data.loc[idx,'candidate'] = 'ROMNEY, MITT'
mit_data[idx]
```

Out[15]:

	year	state	state_po	FIPS	state_cen	state_ic	office	candidate	party_detailed	writein	candidatevotes	t
3371	2012	WASHINGTON	WA	53	91	73	US PRESIDENT	ROMNEY, MITT	REPUBLICAN	False	1290670	

In [16]:
```
output_df = mit_data.loc[:,['state','state_po','FIPS']]
output_df = output_df.drop_duplicates()

years = np.sort(list(presidential_candidates.keys()))

for year in years:
    # Pull this year as a dataframe, pull this year's candidates, and
    # convert year to a string, since it will now be used to name fields
    df=mit_data[mit_data['year']==year]
    candidates = presidential_candidates[year]
    year = str(year)

    # Get candidate info for this year, rename
    gop = df.candidate == candidates['gop']
    gop = df.loc[gop,['state_po','candidatevotes']]
    gop = gop.groupby('state_po').sum()
    gop = gop.rename(columns={'candidatevotes':'gop' + '_' + year + '_votes'})
    dem = df.candidate == candidates['dem']
    dem = df.loc[dem,['state_po','candidatevotes']]
    dem = dem.groupby('state_po').sum()
    dem2 = df.candidate == candidates['dem']
    # New York has the same candidates twice, so you can't just sum totalvotes.
    tot = df.loc[dem2,['state_po','totalvotes']]
    tot = tot.drop_duplicates()
    dem = pd.merge(dem,tot,on='state_po')
    dem = dem.rename(columns={'candidatevotes':'dem' + '_' + year + '_votes'})
    dem = dem.rename(columns={'totalvotes':'totalvotes' + '_' + year})

    # Write this information to the output dataframe and calculate some fields
    output_df = output_df.merge(gop,on='state_po',how='left')
    output_df = output_df.merge(dem,on='state_po',how='left')
    output_df['gop_' + year + '_prc'] = np.round(100 * output_df['gop_' + year + '_votes'] / output_df['tota
    output_df['dem_' + year + '_prc'] = np.round(100 * output_df['dem_' + year + '_votes'] / output_df['tota
    output_df['gop_minus_dem_prc_' + year] = output_df['gop_' + year + '_prc'] - output_df['dem_' + year + '



output_df.head()
```

Out[16]:

	state	state_po	FIPS	gop_1976_votes	dem_1976_votes	totalvotes_1976	gop_1976_prc	dem_1976_prc	gop_minus_dem
0	ALABAMA	AL	01	504070	659170	1182850	42.61	55.73	
1	ALASKA	AK	02	71555	44058	123574	57.90	35.65	
2	ARIZONA	AZ	04	418642	295602	742719	56.37	39.80	
3	ARKANSAS	AR	05	267903	498604	767535	34.90	64.96	
4	CALIFORNIA	CA	06	3882244	3742284	7803770	49.75	47.95	

5 rows × 75 columns

In [17]:
```
output_df.to_csv('state_election_data_1976-2020.csv',index=False,float_format='%.2f')
```

Creating a County GeoPackage

In [102…]:
```
fn = "zip://original_data/cb_2020_us_county_20m.zip"
county_df = geopandas.read_file(fn,dtype={'GEOID':str})
county_df = county_df.drop(columns=['STATEFP','COUNTYFP','COUNTYNS','AFFGEOID','ALAND','AWATER','LSAD','NAME
county_df.head()
```

Out[102]:

	GEOID	NAME	STUSPS	STATE_NAME	geometry
0	01061	Geneva	AL	Alabama	POLYGON ((-86.19348 31.19221, -86.12541 31.182...
1	08125	Yuma	CO	Colorado	POLYGON ((-102.80377 40.00255, -102.79358 40.3...
2	17177	Stephenson	IL	Illinois	POLYGON ((-89.92647 42.50579, -89.83759 42.504...
3	28153	Wayne	MS	Mississippi	POLYGON ((-88.94335 31.82456, -88.91046 31.826...
4	34041	Warren	NJ	New Jersey	POLYGON ((-75.19261 40.71587, -75.17748 40.764...

In [103…]:
```
fn = "zip://original_data/cb_2020_02_sldl_500k.zip"
ak_df = geopandas.read_file(fn,dtype={'GEOID':str})
ak_df = ak_df.drop(columns=['STATEFP','SLDLST','AFFGEOID','NAME','ALAND','AWATER','LSAD','LSY'])
ak_df = ak_df.rename(columns={'NAMELSAD': 'NAME'})
ak_df.head()
```

Out[103]:

	GEOID	NAME	STUSPS	STATE_NAME	geometry
0	02004	State House District 4	AK	Alaska	POLYGON ((-148.66120 65.20987, -148.14022 65.2...
1	02033	State House District 33	AK	Alaska	MULTIPOLYGON (((-134.70152 58.59839, -134.6972...
2	02029	State House District 29	AK	Alaska	MULTIPOLYGON (((-151.01291 60.50222, -151.0117...
3	02031	State House District 31	AK	Alaska	POLYGON ((-151.87191 59.77248, -151.86825 59.7...
4	02040	State House District 40	AK	Alaska	MULTIPOLYGON (((-147.26509 70.21282, -147.2616...

In [104…]:
```
county_df = county_df[county_df['STUSPS']!='AK']
```

In [105…]:
```
county_df = pd.concat([county_df,ak_df])
county_df.head()
```

Out[105]:

	GEOID	NAME	STUSPS	STATE_NAME	geometry
0	01061	Geneva	AL	Alabama	POLYGON ((-86.19348 31.19221, -86.12541 31.182...
1	08125	Yuma	CO	Colorado	POLYGON ((-102.80377 40.00255, -102.79358 40.3...
2	17177	Stephenson	IL	Illinois	POLYGON ((-89.92647 42.50579, -89.83759 42.504...
3	28153	Wayne	MS	Mississippi	POLYGON ((-88.94335 31.82456, -88.91046 31.826...
4	34041	Warren	NJ	New Jersey	POLYGON ((-75.19261 40.71587, -75.17748 40.764...

In [106…]:
```
election_df = pd.read_csv('county_election_data_2000-2020.csv',dtype={'FIPS':str})
election_df.head()
```

Out[106]:

	FIPS	gop_2000_votes	dem_2000_votes	totalvotes_2000	gop_2000_prc	dem_2000_prc	gop_minus_dem_prc_2000	gop_2004_
0	01001	11993.0	4942.0	17208.0	69.69	28.72	40.97	
1	01003	40872.0	13997.0	56480.0	72.37	24.78	47.59	
2	01005	5096.0	5188.0	10395.0	49.02	49.91	-0.89	
3	01007	4273.0	2710.0	7101.0	60.17	38.16	22.01	
4	01009	12667.0	4977.0	17973.0	70.48	27.69	42.79	

5 rows × 37 columns

In [107…]:
```
county_df = county_df.merge(election_df,how='left',left_on='GEOID',right_on='FIPS')
```

In [108…]:
```
county_df.to_file("election.gpkg", layer='county', driver="GPKG")
```

Creating a CONUS GeoPackage

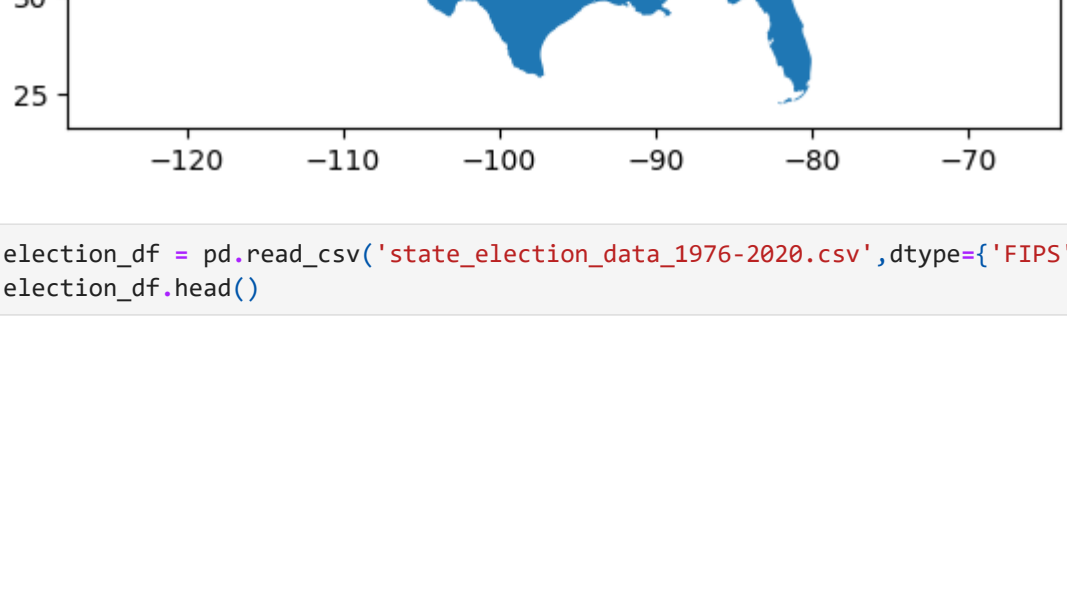
In [22]:
```
fn = "zip://original_data/cb_2020_us_state_20m.zip"
conus_df = geopandas.read_file(fn,dtype={'GEOID':str})
conus_df = conus_df.drop(columns=['STATEFP','ALAND','AWATER','LSAD','STATENS','AFFGEOID','GEOID'])
conus_df = conus_df[~conus_df['STUSPS'].isin(['PR','AK','HI'])]
conus_df.head()
```

Out[22]:

	STUSPS	NAME	geometry
0	CA	California	MULTIPOLYGON (((-118.59397 33.46720, -118.4847...
1	WI	Wisconsin	MULTIPOLYGON (((-86.93428 45.42115, -86.83575 ...
2	ID	Idaho	POLYGON ((-117.24303 44.39097, -117.21507 44.4...
3	MN	Minnesota	POLYGON ((-97.22904 49.00069, -96.93096 48.999...
4	IA	Iowa	POLYGON ((-96.62187 42.77925, -96.57794 42.827...

In [23]:
```
conus_df.plot()
```

Out[23]: <AxesSubplot: >



In [24]:
```
election_df = pd.read_csv('state_election_data_1976-2020.csv',dtype={'FIPS':str})
election_df.head()
```

Out[24]:

	state	state_po	FIPS	gop_1976_votes	dem_1976_votes	totalvotes_1976	gop_1976_pct	dem_1976_pct	gop_minus_dem
0	ALABAMA	AL	01	504070	659170	1163830	42.61	55.73	
1	ALASKA	AK	02	71555	44058	123574	57.90	35.65	
2	ARIZONA	AZ	04	418642	295602	742719	56.37	39.80	
3	ARKANSAS	AR	05	267903	498604	767535	34.90	64.96	
4	CALIFORNIA	CA	06	3882244	3742284	7803770	49.75	47.95	

5 rows × 75 columns

In [25]:
```
conus_df = conus_df.merge(election_df,how='left',left_on='STUSPS',right_on='state_po')
```

In [26]:
```
conus_df.to_file("election_conus.gpkg", layer='state', driver="GPKG")
```

In [ ]: