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.

Modifications:

- The original file was edited to include data for Keya Paha, Nebraska: 460 votes Trump, 40 votes Clinton, 19 votes other, 519 total

In [1]:
```
import pandas as pd
import numpy as np
```

County Election Data

In [2]:
```
mit_data = pd.read_csv('original_data/countypres_2000-2016.csv',dtype={'FIPS':str})
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 [3]:
```
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   FIPS     office       candidate  \
0  2000  Alabama       AL  Autauga  01001  President        Al Gore
1  2000  Alabama       AL  Autauga  01001  President  George W. Bush
2  2000  Alabama       AL  Autauga  01001  President     Ralph Nader
3  2000  Alabama       AL  Autauga  01001  President           Other
4  2000  Alabama       AL  Baldwin  01003  President        Al Gore

        party  candidatevotes  totalvotes   version
0    democrat          4942.0     17208.0  20181011
1  republican         11993.0     17208.0  20181011
2       green           160.0     17208.0  20181011
3         NaN           113.0     17208.0  20181011
4    democrat         13997.0     56480.0  20181011
```

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

In [4]:
```
mit_data.loc[mit_data['FIPS']=='46113','FIPS'] = '46102'
```

Continue with data processing

In [5]:
```
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'}
                           }
```

In [6]:
```
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[
'totalvotes_' + year],decimals=2)
    output_df['dem_' + year + '_prc'] = np.round(100 * output_df['dem_' + year + '_votes'] / output_df[
'totalvotes_' + year],decimals=2)
    output_df['gop_minus_dem_prc_' + year] = output_df['gop_' + year + '_prc'] - output_df['dem_' + yea
r + '_prc']

output_df.to_csv('county_election_data_2000-2016.csv',index=False,float_format='%.2f')
```

State Election Data

In [7]:
```
mit_data = pd.read_csv('original_data/1976-2016-president.csv',dtype={'state_fips':str})
mit_data = mit_data[~mit_data['state_fips'].isnull()]
mit_data['state_fips'] = mit_data.state_fips.str.zfill(2)
```

In [8]:
```
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'}
                           }
```

In [9]:
```
output_df = mit_data.loc[:,['state','state_po','state_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','totalvotes']]
    dem = dem.groupby('state_po').sum()
    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[
'totalvotes_' + year],decimals=2)
    output_df['dem_' + year + '_prc'] = np.round(100 * output_df['dem_' + year + '_votes'] / output_df[
'totalvotes_' + year],decimals=2)
    output_df['gop_minus_dem_prc_' + year] = output_df['gop_' + year + '_prc'] - output_df['dem_' + yea
r + '_prc']
```

In [10]:
```
output_df.to_csv('state_election_data_1976-2016.csv',index=False,float_format='%.2f')
```