

Bishop claims that a “Landslide County” is any county that the presidential election vote was determined by greater than or equal to 20 percent of the vote. He also claimed that a “Competitive County” is where the election was determined by less than or equal to 10 percent of the vote.

| Dave Liep  | FEC   | Tony McGovern   | Timothy Renner  |
|--|---|---|---|
| The website is very cool, a tad outdated in the technology. I really like how the maps tell you the information if you hover your mouse over a state. It provides only the state information for free, which is crazy to me that he sells the county data. I also like the charts he has on the sides of the map telling you which way the state leaned. However the data is not all that detailed and I don't think it includes all the states. | This dataset is huge, and has data that is not necessary for our analysis. I got lost in the pages because of how much information there is. It is also by state, so we cannot analyze the counties. The maps created are great though, I like how he uses granulated colors and for each popular vote and all states, not just the states that went left or right. | I like how tony has the county data, it is a lot like the data we were provided for this assignment. It has the total, dem, and gop for each year. He does a really good job at showing the data for each county, and I like how he includes parties other than gop and dem, even though it is not necessary for this assignment. He provides the fips class code, which I dont think is needed but it could maybe help down the road or something. Kind of cool how he combines the State name with the county name, but it really is not needed. I do wish he created maps as well as the charts. | I like the data table, however it is very simplistic. I don't see why the election, chamber or district is really necessary. Same with the information regarding if it was a write in or not. This data also seems to be only in states as well so it is not very easy to use for this assignment. The layout of the states are odd as well, I am not really a fan of this data set as a whole. |

This is a processing script to aggregate [MIT's Election Data](#) for United States presidential 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 #It is importing pandas module into a variable called pd
import numpy as np #It is importing numpy module into a variable called 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)
#It is having pandas read a csv file and stores it in the variable mit_data maybe making the FIPS line a string?
#I am not really sure what the second two lines are doing... making the FIPS column null as in empty? And maybe making the now null FIPS have a constant of 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'])
#It is grouping the above variable mit_data by the year and the FIPS, then getting the sum i believe and idk what reset index does but then it stores that into the variable grp
#Next I think it drops the columns titled totalvotes and version and stores it back into the grp variable
#I think it renames the columns to candidatevotes and totalvotes2 and stores it back into the grp variable
#I believe the last line merges the new columns that grp created into the mit_data csv file

mit_data['totalvotes'] = mit_data['totalvotes2']
mit_data = mit_data.drop(labels=['totalvotes2'],axis=1)
print(mit_data.head())
#It is making the grp column totalvotes2 be renamed / set as the column name totalvotes in the mit_data csv file
#Now mit_data is getting rid of the totalvotes2 column label since that and total votes are the same
#Prints the beginning of the csv file mit_data as seen below

  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' #mit_data is locating the FIPS of a county in the csv file and then makes it a different county ?
```

**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'}}
} #It is creating a list or dictionary or library of the presidential candidates for democrats and republicans from 2000-2016 and naming the list or dictionary presidential_candidates

In [6]: output_df = pd.DataFrame()
output_df['FIPS'] = mit_data['FIPS'].unique()
#It is having pandas create a data frame called output_df
#The output_df is making the FIPS column equal mit_data's FIPS with unique values??

years = np.sort(list(presidential_candidates.keys())) #It is sorting the list of presidential candidates by their keys which is the year and storing it into the variable 'years'

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)
  #It is creating a for loop that says for each year in the list of years it will create the variable df which is the mit_data 'year' and make it equal year? And then makes the variable candidates which is the year in presidential_candidates and then creates the year into a string?

  # 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})
  #It gets the name of each candidate for first the republican party and names it 'gop', then it grabs the 'gop'and locates the FIPS and candidate votes. and then renames the columns to have 'gop' the name of gop the year and amount of votes? And then does the same thing for democrat candidates in the variable dem?

  # 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_' + year + '_prc']

output_df.to_csv('county_election_data_2000-2016.csv',index=False,float_format='%.2f')
#In all honesty i have no idea what is going on... I think it is merging the output_df and the gop data into output_df and same with the dem data. Then it calculates fields with the gop year and prc which idk what that is and then it divides the output of gop year votes by totalvotes and year and multiples it by 100 rounds everything by 2 decimal places and does the same for the dem data. And then i believes it creates the gop_minus_dem_prc year column which is the gop year and prc minus the dem year prc!
#then it exports the data of output_df as a csv called county_election_data_2000-2016.csv, gets rid of the index so they dont count themselves down the list, and i do not know what a float format is... what is it?
```

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)
#Alright i think its the same deal from the county data, it reads a csv file and makes the state_fips column a string i believe and stores it in mit_data - then it makes the column null i believe, and then makes a new column? And makes the number 2 maybe idk

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'}}
#It is creating a list and/or dictionary for the presidential_candidates from 1976 to 2016, and each party with candidates name

In [9]: output_df = mit_data.loc[:, ['state', 'state_po', 'state_fips']]
output_df = output_df.drop_duplicates()
#It is creating the variable output_df that is the mit_data csv file and locates the state state_po and state_fips columns and then drops the duplicates of each state maybe

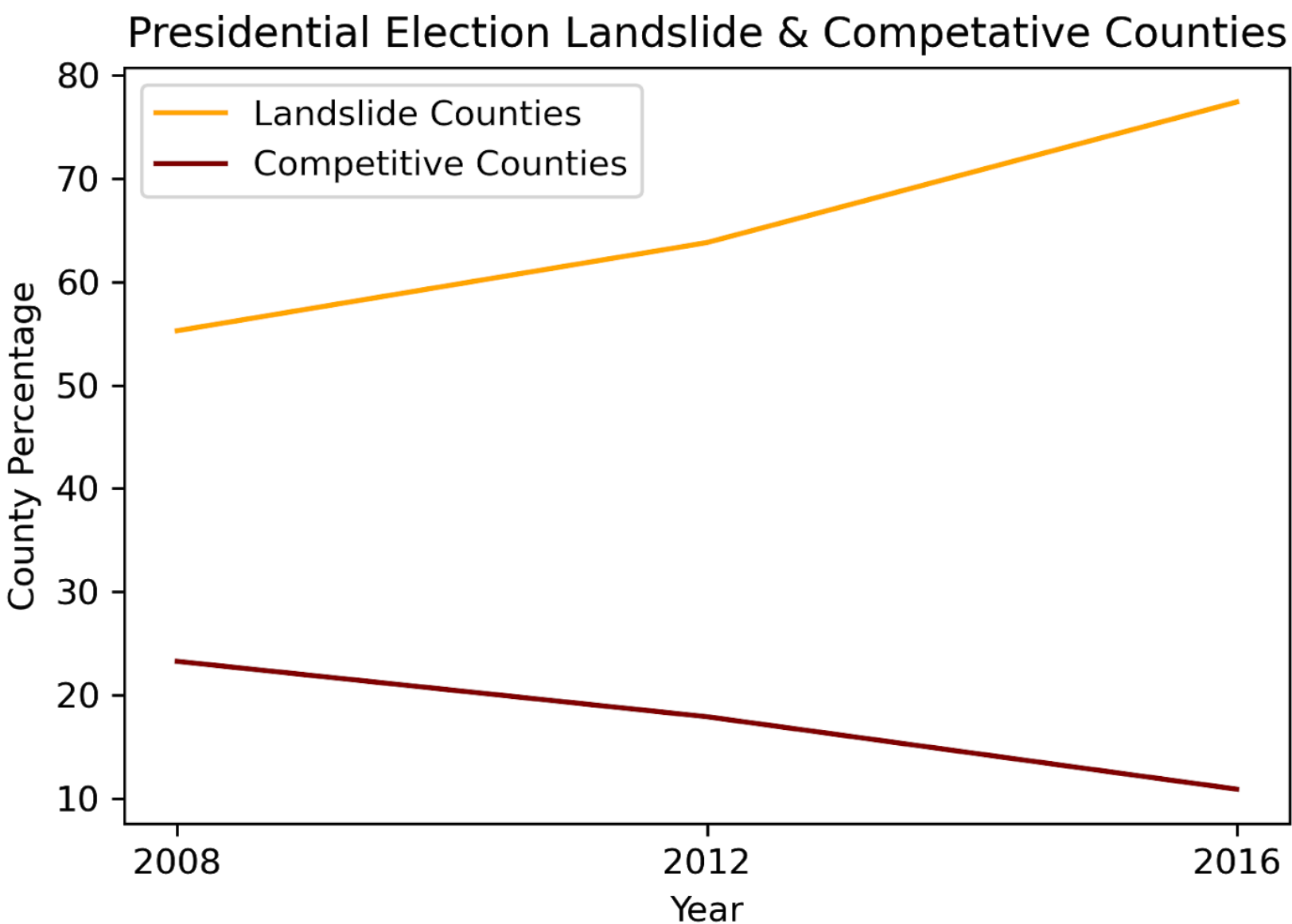
years = np.sort(list(presidential_candidates.keys()))
#It is creating a variable 'years' which is the sorted list of the presidential candidates by their keys which are the years

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)
#same thing as before I believe it creates a for loop for the year in the list of years and creates df variable which is the mit_data column of year and makes it equal the variable 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})
#I believe it is creating the variable gop which is the candidates in the column of gop in the list. Then it locates the gop, state_po (population i am assuming?) and then the amount of votes and stores it back into gop. It then groups it by the state_po and gets the sum and stores it back into gop. Then the columns are renamed to candidatevotes with gop a space the year and amount of votes. It then does the same for the democrat party and also locates the totalvotes column and stores it in the variable dem, and then renames two columns with dem a space the year and amount of votes, and totalvotes with totalvotes a space and the year.

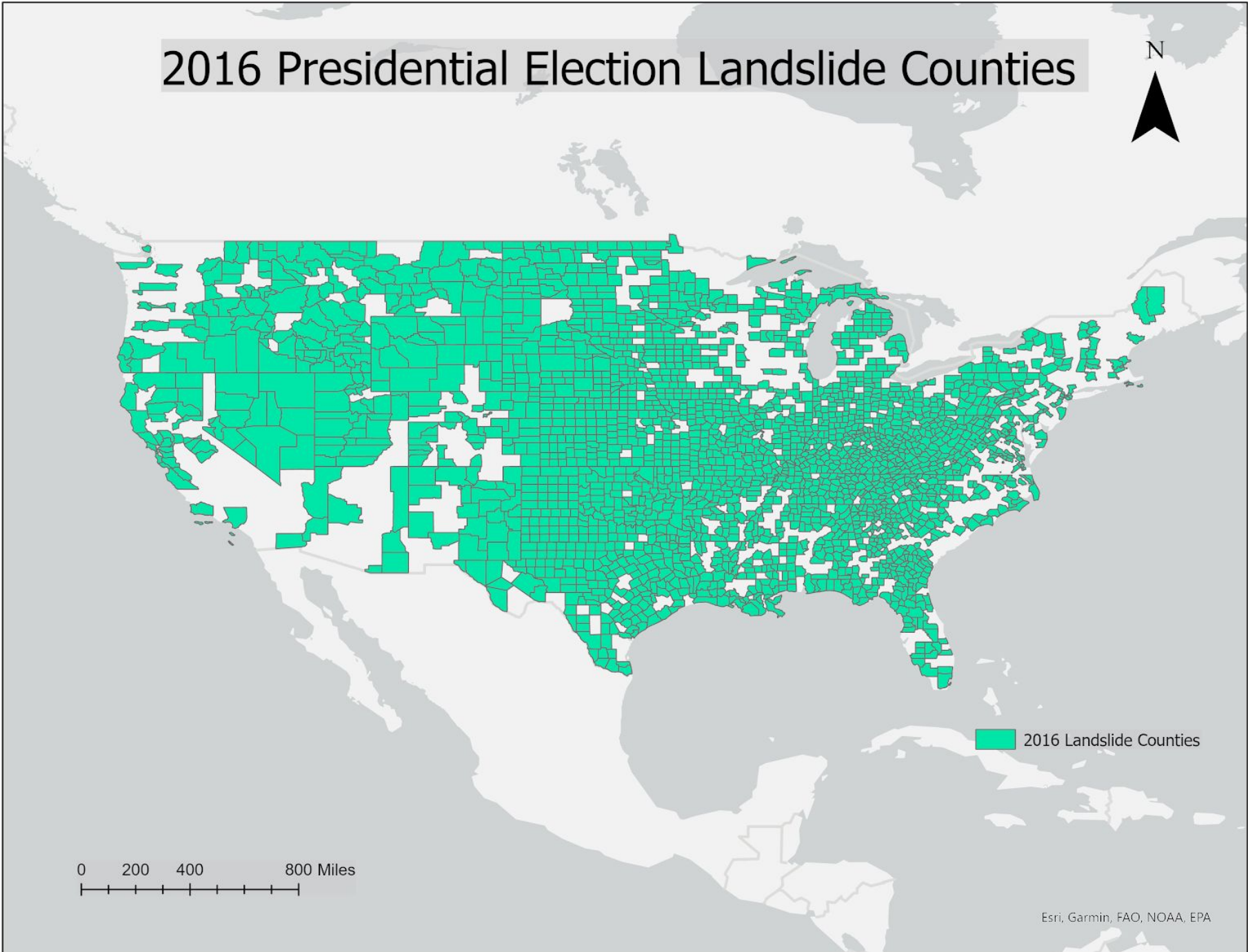
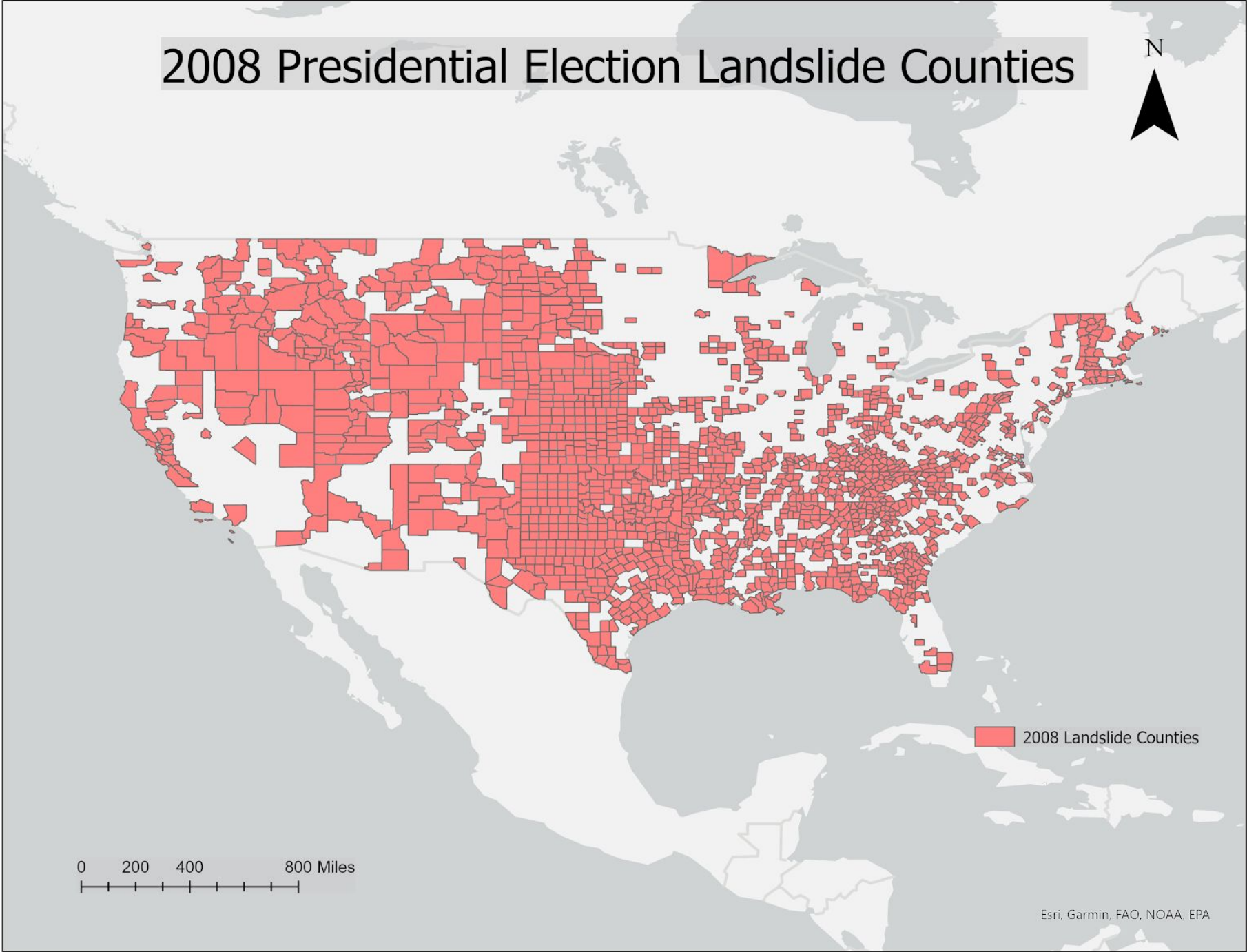
    # Write this information to the output dataframe and calculate some fields
    output_df = output_df.merge(gop,on='state_po',how='left')
    output_df = output_df.merge(dem,on='state_po',how='left')
    output_df['gop_' + year + '_prc'] = np.round(100 * output_df['gop_' + year + '_votes'] / output_df['totalvotes_' + 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_' + year + '_prc']
#this creates a new data frame called output_df and merges the gop and dem data i ~believe~ and then calculates a new field called gop year percent and then it divides the output of gop year votes by totalvotes and year and multiples it by 100 and rounds everything by 2 decimal places, and does the same for the dem votes called dem year prc.then it creates a new column called gop_minus_dem_prc_year column which is the gop year and prc minus the dem year prc.
```

```
In [10]: output_df.to_csv('state_election_data_1976-2016.csv',index=False,float_format='%.2f') #It exports output_df to a csv file called state_election_data_1976-2016.csv, it makes the index false so the states do not count themselves going down the list, and then i have no idea what float format means.
```



The amount of landslide counties increased positively from the 2008 Presidential Election to the 2016 Presidential Election. The amount of competitive counties decreased from the 2008 Presidential Election to the 2016 Presidential Election.





In his book, *The Big Sort*, Bishop claims that the amount of locally polarized Americans is increasing. Americans are living in more counties where the presidential election vote is determined by greater than or equal to 20% of the vote percentage, or ‘landslide counties’. The amount of ‘competitive counties’ are decreasing drastically since landslide counties are the majority of the population.

Looking at the graph above, it is very noticeable that the landslide counties, depicted by the orange line, are increasing significantly between the election years. In 2008, the county percentage was ~55% and increased a bit to ~62% the following election four years later. However, in 2016, the percentage skyrocketed to ~78%, having a ~20% increase from 2008 to 2016. The competitive counties decreased dramatically. In 2008, the county percentage was ~23% and went down to ~19% in 2012. In 2016, the competitive counties were at an all time low at ~12%, making the total difference ~10% over the eight years.

In regards to the 2008 Landslide map, there is a noticeable amount of counties missing. There is a large cluster of counties in the Midwest area, with the majority of Texas counties on the map. The Michigan, Wisconsin, and Minnesota area is very close to empty along with a good portion of Ohio. The New England area has a large disbursement of counties present, with New York and Pennsylvania having little amount of their counties showing, while almost the entirety of Maine is missing. The Florida peninsula has only a few visible counties. The West coast has a good amount of clustering particularly in Idaho and a portion of Nevada. Washington State, Oregon, and California have little visible counties, with Southern California almost empty.

In regards to the 2016 Landslide map, there has been a noticeable increase in visible counties in comparison to 2008. On the East Coast, Maine gained a few counties, The Carolinas have filled in a bit, and Florida has almost a full peninsula. The Michigan, Wisconsin, and Minnesota area has gained a lot of counties, going from entirely empty to almost completely covered. Montana and North Dakota filled in a little bit, same with Texas. There is still a noticeable amount of counties missing to the right of the Mississippi River. Northern California and Oregon filled a lot, leaving Southern California still empty. Arizona actually lost a few counties, and there was little to no increase of counties in the Colorado area.

Seeing how the amount of landslide counties has increased dramatically from 2008 to 2016, leaving us to make the conclusion that Bishop was right in his claim that the states are becoming more polarized. With a huge increase shown by a trendline in the graph above, and visually by the two maps, it is safe to say that Americans are polarized. I am extremely interested to see how the map of the 2020 Presidential Election would look in comparison to the maps above.