Import tools.

In [1]:

```python
import pandas as pd
from pickle import load
from pickle import dump
import numpy as np
pd.set_option("max_rows", None)
pd.set_option("max_columns", None)
```

First Dataset: RESULTS

This dataset includes 2016 Presidential election results by county, 2012 presidential election results by county, county density, county type, and land area. It does not include Alaska. Alaska information can be found in the next section of this notebook. https://data.world/garyhoov/2016-pres-election-by-county (https://data.world/garyhoov/2016-pres-election-by-county)

Open dataset.

In [2]:

```python
results = pd.read_csv('data/2016 Presidential Election Analysis.csv')
```

In [3]:

```python
results.head()
```

Out[3]:

	State Code	County Name	County Population	Clinton or Trump State	Clinton	Trump	Total	% Clinton	% Trump	Vote Difference C-T	V Differe
0	AL	Autauga County	55,347	Trump	5,908	18,110	24,661	23.96%	73.44%	(12,202)	12,
1	AL	Baldwin County	203,709	Trump	18,409	72,780	94,090	19.57%	77.35%	(54,371)	54,
2	AL	Barbour County	26,489	Trump	4,848	5,431	10,390	46.66%	52.27%	(583)	
3	AL	Bibb County	22,583	Trump	1,874	6,733	8,748	21.42%	76.97%	(4,859)	4,
4	AL	Blount County	57,673	Trump	2,150	22,808	25,384	8.47%	89.85%	(20,658)	20,

Only keep helpful columns. Keep Clinton and Trump total votes until target is established.

In [4]:

```python
results = results.iloc[:, [0, 1, 4, 5, 6, 13, 14, 15, 24, 25, 38]]
```

In [5]:
```
results.head()
```

Out[5]:

	State Code	County Name	Clinton	Trump	Total	Obama	Romney	2012 Total Votes	2010 Land Area	Density	Central/Outlyin Count
0	AL	Autauga County	5,908	18,110	24,661	6,354	17,366	23,909	594	93	Centr
1	AL	Baldwin County	18,409	72,780	94,090	18,329	65,772	84,988	1590	128	Centr
2	AL	Barbour County	4,848	5,431	10,390	5,873	5,539	11,459	885	30	Centr
3	AL	Bibb County	1,874	6,733	8,748	2,200	6,131	8,391	623	36	Outlyir
4	AL	Blount County	2,150	22,808	25,384	2,961	20,741	23,980	645	89	Outlyir

Remove punctuation from column names and values.

In [6]:
```
results.columns = results.columns.str.strip().str.replace('[^\w\s]', '')
```

```
<ipython-input-6-12a2c89aa82b>:1: FutureWarning: The default value of regex wil
l change from True to False in a future version.
  results.columns = results.columns.str.strip().str.replace('[^\w\s]', '')
```

In [7]:
```
def remove_punctuation(x):
    try:
        x = x.str.replace('[^\w\s]','')
    except:
        pass
    return x
```

In [8]:
```
results = results.apply(remove_punctuation)
```

```
<ipython-input-7-fabae1a16f96>:3: FutureWarning: The default value of regex wil
l change from True to False in a future version.
  x = x.str.replace('[^\w\s]','')
```

In [9]:
```
results.head()
```

Out[9]:

	State Code	County Name	Clinton	Trump	Total	Obama	Romney	2012 Total Votes	2010 Land Area	Density	CentralOutlying County
0	AL	Autauga County	5908	18110	24661	6354	17366	23909	594	93	Central
1	AL	Baldwin County	18409	72780	94090	18329	65772	84988	1590	128	Central
2	AL	Barbour County	4848	5431	10390	5873	5539	11459	885	30	Central
3	AL	Bibb County	1874	6733	8748	2200	6131	8391	623	36	Outlying
4	AL	Blount County	2150	22808	25384	2961	20741	23980	645	89	Outlying

Lengthen state names. Pretty sure we don't actually need to do this. Drop?

In [10]:

```
us_state_abbrev = {
    'AL': 'Alabama',
    'AK': 'Alaska',
    'AZ': 'Arizona',
    'AR': 'Arkansas',
    'CA': 'California',
    'CO': 'Colorado',
    'CT': 'Connecticut',
    'DE': 'Delaware',
    'FL': 'Florida',
    'GA': 'Georgia',
    'HI': 'Hawaii',
    'ID': 'Idaho',
    'IL': 'Illinois',
    'IN': 'Indiana',
    'IA': 'Iowa',
    'KS': 'Kansas',
    'KY': 'Kentucky',
    'LA': 'Louisiana',
    'ME': 'Maine',
    'MD': 'Maryland',
    'MA': 'Massachusetts',
    'MI': 'Michigan',
    'MN': 'Minnesota',
    'MS': 'Mississippi',
    'MO': 'Missouri',
    'MT': 'Montana',
    'NE': 'Nebraska',
    'NV': 'Nevada',
    'NH': 'New Hampshire',
    'NJ': 'New Jersey',
    'NM': 'New Mexico',
    'NY': 'New York',
    'NC': 'North Carolina',
    'ND': 'North Dakota',
    'OH': 'Ohio',
    'OK': 'Oklahoma',
    'OR': 'Oregon',
    'PA': 'Pennsylvania',
    'RI': 'Rhode Island',
    'SC': 'South Carolina',
    'SD': 'South Dakota',
    'TN': 'Tennessee',
    'TX': 'Texas',
    'UT': 'Utah',
    'VT': 'Vermont',
    'VA': 'Virginia',
    'WA': 'Washington',
    'WV': 'West Virginia',
    'WI': 'Wisconsin',
    'WY': 'Wyoming',
}
```

In [11]:

```
results['State Code'] = results['State Code'].map(us_state_abbrev)
```

Rename columns.

In [12]:
```
results.rename(columns = {'Total': '2016_total_votes', '2012 Total Votes':'2012_t
```

If state is missing, fill with county name. Pretty sure this only applies to DC.

In [13]:
```
results['State'] = results['State'].fillna(results['County'])
```

Alaska is not broken down by county in this dataset. Dropping all Alaska info and pulling in results and info from another source.

In [14]:
```
results.drop(results.loc[results['State'].str.contains('Alaska', case=False)].inc
```

Simplify long string to easier to read string.

In [15]:
```
results.replace('Not Metro or Micro Presumed Rural', 'Rural', inplace = True)
```

Change datatypes to numeric where necessary.

In [16]:
```
results.iloc[:, 2:9] = results.iloc[:, 2:9].apply(pd.to_numeric)
```

In [17]:
```
results.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3112 entries, 0 to 3112
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   State             3112 non-null   object
 1   County            3112 non-null   object
 2   Clinton           3112 non-null   int64 
 3   Trump             3112 non-null   int64 
 4   2016_total_votes  3112 non-null   int64 
 5   Obama             3112 non-null   int64 
 6   Romney            3112 non-null   int64 
 7   2012_total_votes  3112 non-null   int64 
 8   2010_land_area    3112 non-null   int64 
 9   Density           3112 non-null   int64 
 10  central_outlying  3112 non-null   object
dtypes: int64(8), object(3)
memory usage: 291.8+ KB
```

Create new column indicating who won the county. To be used as target for modeling.

In [18]:
```
conditions = [(results['Clinton'] < results['Trump']), (results['Clinton'] > resu
choices = ['Trump', 'Clinton']
results['Target'] = np.select(conditions, choices, default = np.nan)
```

Drop Clinton and Trump individual columns.

In [19]:
```
results = results.drop(results.iloc[:, 2:4], axis = 1)
```

In [20]:
```
results.head()
```

Out[20]:

	State	County	2016_total_votes	Obama	Romney	2012_total_votes	2010_land_area	Density
0	Alabama	Autauga County	24661	6354	17366	23909	594	93
1	Alabama	Baldwin County	94090	18329	65772	84988	1590	128
2	Alabama	Barbour County	10390	5873	5539	11459	885	30
3	Alabama	Bibb County	8748	2200	6131	8391	623	36
4	Alabama	Blount County	25384	2961	20741	23980	645	89

ALASKA RESULTS

**PUT DATA INFO HERE make sure to mention how write ins are distributed.

Open Alaska results file.

In [23]:
```
alaska_results = pd.read_csv('data/ak_2012_2016.csv')
```

In [24]:
```
alaska_results.head()
```

Out[24]:

	Weighted/Muni	Registered Voters	Clinton, Hillary	Trump, Donald J.	Write-in 60	WtTotal	Unnamed: 6	Weig
0	Ketchikan Gateway	10512	1966.695802	3451.907138	153.011809	6267.150170	NaN	
1	Prince of Wales-Hyder	4630	1076.455803	1295.125060	91.039696	2830.673904	NaN	Prir
2	Sitka	7218	2110.994000	1811.544401	115.703622	4427.915868	NaN	
3	Petersburg	2741	569.639406	915.365934	50.796562	1706.007455	NaN	
4	Wrangell	1731	270.992898	751.941311	19.081913	1124.152497	NaN	

Drop unnecessary columns.

In [25]:
```
alaska_results = alaska_results.drop(alaska_results.iloc[:, [1, 4, 6, 7, 8, 9, 11
```

Get rid of commas in column names and rename columns.

In [26]:
```
alaska_results.columns = [col.replace(',', '') for col in alaska_results.columns]
```

In [27]:
```
alaska_results = alaska_results.rename(columns = {'Registered Voters': '2016_regi
```

Create target column.

In [28]:
```
conditions = [alaska_results.iloc[:, 2] < alaska_results.iloc[:, 3], alaska_resul
choices = ['Trump', 'Clinton']
alaska_results['Target'] = np.select(conditions, choices, default = np.nan)
```

In [29]:
```
alaska_results.head()
```

Out[29]:

	County	Clinton Hillary	Trump Donald J.	2016_total_votes	2012_total_votes	Obama	Romn
0	Ketchikan Gateway	1966.695802	3451.907138	6267.150170	5905.592707	2262.784210	3266.6354
1	Prince of Wales-Hyder	1076.455803	1295.125060	2830.673904	2482.062762	1298.248629	1045.1016
2	Sitka	2110.994000	1811.544401	4427.915868	4415.265365	2340.002643	1830.6941
3	Petersburg	569.639406	915.365934	1706.007455	1729.880554	776.066018	867.4595
4	Wrangell	270.992898	751.941311	1124.152497	1143.064441	362.631179	738.6862

In [30]:
```
alaska_results = alaska_results.drop(alaska_results.iloc[:, [1, 2]], axis = 1)
```

Row 29 is a summed column of all counties. Dropping.

In [31]:
```
alaska_results.drop(alaska_results.index[29], inplace = True)
```

Wade Hampton county was renamed Kusilvak.

In [32]:
```
alaska_results = alaska_results.replace({'Wade Hampton': 'Kusilvak'}, regex = Tru
```

Add state column and sort by county.

In [33]:
```
alaska_results['State'] = 'Alaska'
```

In [34]:
```
alaska_results = alaska_results.sort_values(by = 'County')
```

In [35]:
```
alaska_results.head()
```

Out[35]:

	County	2016_total_votes	2012_total_votes	Obama	Romney	Target	State
22	Aleutians East	529.293851	549.375577	234.120530	292.395684	Trump	Alaska
24	Aleutians West	1213.502975	1238.761919	777.428504	426.573343	Trump	Alaska
19	Anchorage	130040.329900	125169.133300	54042.760210	66387.084670	Trump	Alaska
12	Bethel	4892.232820	4810.611592	3425.621480	1151.530057	Trump	Alaska
25	Bristol Bay	453.270615	425.845526	147.147402	251.541638	Trump	Alaska

Additional Alaska Information

**put data info here. I guess just say you pulled it from Wikipedia and made a spreadsheet...

Open file.

In [38]:
```
missing_columns = pd.read_csv('data/alaska_missing_columns.csv')
```

Rename some columns.

In [39]:
```
missing_columns.rename(columns = {'Land Area': '2010_land_area', 'Metro/Nonmetro
```

Drop strange extra columns.

In [40]:
```
missing_columns = missing_columns.drop(missing_columns.iloc[:, 4:6], axis = 1)
```

Rename Wade Hampton to Kusilvak.

In [41]:
```
missing_columns.replace('Wade Hampton(kusilvak)', 'Kusilvak', inplace = True)
```

Change rural to Rural to match other datasets.

In [42]:
```
missing_columns.replace('rural', 'Rural', inplace = True)
```

Sort for merge.

In [43]:
```
missing_columns = missing_columns.sort_values(by = 'Weighted/Muni')
```

In [44]:
```
missing_columns.head()
```

Out[44]:

	Weighted/Muni	Density	2010_land_area	central_outlying
0	Aleutians East	0.49	6982	Rural
1	Aleutians West	1.19	4390	Rural
2	Anchorage	170.62	1705	Central
3	Bethel	0.46	40570	Rural
4	Bristol Bay	1.75	504	Rural

Join Alaska election results with Alaska additional columns.

Merge.

In [45]:
```
alaska_total_results = alaska_results.reset_index(drop=True).merge(missing_colum
```

In [46]:
```
alaska_total_results.head()
```

Out[46]:

	County	2016_total_votes	2012_total_votes	Obama	Romney	Target	State	Weig
0	Aleutians East	529.293851	549.375577	234.120530	292.395684	Trump	Alaska	Aleu
1	Aleutians West	1213.502975	1238.761919	777.428504	426.573343	Trump	Alaska	Aleu
2	Anchorage	130040.329900	125169.133300	54042.760210	66387.084670	Trump	Alaska	/
3	Bethel	4892.232820	4810.611592	3425.621480	1151.530057	Trump	Alaska	
4	Bristol Bay	453.270615	425.845526	147.147402	251.541638	Trump	Alaska	I

Look at county columns side by side to double check.

In [47]:

```
alaska_total_results.iloc[:, [0, 7]]
```

Out[47]:

	County	Weighted/Muni
0	Aleutians East	Aleutians East
1	Aleutians West	Aleutians West
2	Anchorage	Anchorage
3	Bethel	Bethel
4	Bristol Bay	Bristol Bay
5	Denali	Denali
6	Dillingham	Dillingham
7	Fairbanks North Star	Fairbanks North Star
8	Haines	Haines
9	Hoonah-Angoon	Hoonah-Angoon
10	Juneau	Juneau
11	Kenai Peninsula	Kenai Peninsula
12	Ketchikan Gateway	Ketchikan Gateway
13	Kodiak Island	Kodiak Island
14	Kusilvak	Kusilvak
15	Lake and Peninsula	Lake and Peninsula
16	Matanuska-Susitna	Matanuska-Susitna
17	Nome	Nome
18	North Slope	North Slope
19	Northwest Arctic	Northwest Arctic
20	Petersburg	Petersburg
21	Prince of Wales-Hyder	Prince of Wales-Hyder
22	Sitka	Sitka
23	Skagway	Skagway
24	Southeast Fairbanks	Southeast Fairbanks
25	Valdez-Cordova	Valdez-Cordova
26	Wrangell	Wrangell
27	Yakutat	Yakutat
28	Yukon-Koyukuk	Yukon-Koyukuk

Drop second county column.

In [48]:
```
alaska_total_results.drop('Weighted/Muni', axis = 1, inplace = True)
```

Join Alaska results with the rest of the country.

alaska_total_results, results

In [49]:
```
total_results = results.append(alaska_total_results)
```

In [128]:
```
total_results.head()
```

Out[128]:

	State	County	2016_total_votes	Obama	Romney	2012_total_votes	2010_land_area	Density
0	Alabama	Autauga County	24661.0	6354.0	17366.0	23909.0	594	93.0
1	Alabama	Baldwin County	94090.0	18329.0	65772.0	84988.0	1590	128.0
2	Alabama	Barbour County	10390.0	5873.0	5539.0	11459.0	885	30.0
3	Alabama	Bibb County	8748.0	2200.0	6131.0	8391.0	623	36.0
4	Alabama	Blount County	25384.0	2961.0	20741.0	23980.0	645	89.0

In [51]:
```
total_results.isna().sum()
```

Out[51]:
```
State               0
County              0
2016_total_votes    0
Obama               0
Romney              0
2012_total_votes    0
2010_land_area      0
Density             0
central_outlying    0
Target              0
dtype: int64
```

In [52]:
```
total_results.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3141 entries, 0 to 28
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   State             3141 non-null   object 
 1   County            3141 non-null   object 
 2   2016_total_votes  3141 non-null   float64
 3   Obama             3141 non-null   float64
 4   Romney            3141 non-null   float64
 5   2012_total_votes  3141 non-null   float64
 6   2010_land_area    3141 non-null   int64  
 7   Density           3141 non-null   float64
 8   central_outlying  3141 non-null   object 
 9   Target            3141 non-null   object 
dtypes: float64(5), int64(1), object(4)
memory usage: 269.9+ KB
```

Race info from 2010 Census

**add data info here

In [54]:
```
df_race = pd.read_csv('data/DECENNIALPL2020.P1_data_with_overlays_2021-10-04T2028
```

Sort out columns and index.

In [55]:
```
df_race.columns = df_race.iloc[1]
```

In [56]:
```
df_race.drop(df_race.index[1], inplace = True)
```

In [57]:
```
 df_race.drop(df_race.index[0], inplace = True)
```

Get rid of columns detailing multi-race breakdown but keep total of multi-race population.

In [58]:
```
df_race.drop(df_race.iloc[:, 11:73], axis = 1, inplace = True)
```

Split county and state into two columns and drop old column.

In [59]:
```
df_race[['County', 'State']] = df_race['Geographic Area Name'].str.split(',', exp
```

In [60]:
```
df_race.drop('Geographic Area Name', axis = 1, inplace = True)
```

Get rid of punctuation.

In [61]:
```
df_race.columns = df_race.columns.str.strip().str.replace('[^\w\s]', '')
```

```
<ipython-input-61-3176a88a0be0>:1: FutureWarning: The default value of regex wi
ll change from True to False in a future version.
  df_race.columns = df_race.columns.str.strip().str.replace('[^\w\s]', '')
```

In [62]:
```
def remove_punctuation(x):
    try:
        x = x.str.replace('[^\w\s]','')
    except:
        pass
    return x
df_race = df_race.apply(remove_punctuation)
```

```
<ipython-input-62-5daa401cd4a4>:3: FutureWarning: The default value of regex wi
ll change from True to False in a future version.
  x = x.str.replace('[^\w\s]','')
```

Convert race/population info into floats.

In [63]:
```
df_race[df_race.columns[1:-2]] = df_race[df_race.columns[1:-2]].astype(float)
```

Chugach and Copper River counties in Alaska were combined between the time of the census and the time of the election. Here's my super duper annoying way of handling that...

In [64]:
```
chugach = (df_race.loc[df_race['County'] == 'Chugach Census Area'])
```

In [65]:
```
copper_river = (df_race.loc[df_race['County'] == 'Copper River Census Area'])
```

In [66]:
```
old_counties = chugach.append(copper_river)
valdez = pd.DataFrame(old_counties.sum(numeric_only = False, axis = 0)).T
df_race.drop(old_counties.index, axis = 0, inplace= True)
df_race = df_race.append(valdez)
```

In [67]:
```
df_race = df_race.replace({'0500000US020630500000US02066': 'akcombined', 'Chugach
df_race['State'] = df_race['State'].str.replace('Alaska Alaska', 'Alaska')
```

Drop Puerto Rico because they can not vote in presidential elections.

In [68]:
```
pr = df_race.loc[df_race['State'].str.contains('Puerto Rico', case=False)]
```

In [69]:
```
df_race = df_race.drop(pr.index, axis = 0)
```

Dropping Kalawoa. Only has 82 residents and does not come up in census pull. (read through... what? This is the census pull...)

```
In [70]: Kalawao = df_race.loc[df_race['County'].str.contains('Kalawao', case=False)]
```

```
In [71]: df_race = df_race.drop(Kalawao.index, axis = 0)
```

Rename annoying columns.

```
In [72]: df_race.rename(columns = {'Total': 'total_pop', 'TotalPopulation of one race': 't
```

Sort and reset index for joining.

```
In [73]: df_race = df_race.sort_values(by = ['State', 'County']).reset_index(drop = True)
```

Poverty census info

```
In [76]: df = pd.read_csv('data/poverty.csv')
```

```
In [77]: df.columns = df.iloc[0]
```

```
In [78]: df.drop(df.index[0], inplace = True)
```

```
In [79]: df = df.reset_index(drop = True)
```

Remove columns that contain the word 'Bound'. These are not needed.

```
In [80]: df = df.loc[:, ~df.columns.str.contains('Bound')]
```

Remove columns that are percentages.

```
In [81]: df = df.loc[:, ~df.columns.str.contains('Percent')]
```

Locate DC and change County FIPS so it isn't dropped during the next step.

```
In [82]: df.loc[(df['County FIPS'] == '000') & (df['Name'] == 'District of Columbia')]
```

Out[82]:

	State FIPS	County FIPS	Postal	Name	Poverty Estimate All Ages	Poverty Estimate Under Age 18	Poverty Estimate Ages 5-17	Median Household Income	Poverty Estimate Ages 0-4
328	11	000	DC	District of Columbia	107,279	31,147	20,872	60,729	9,786

In [83]:
```
df.iloc[327:330, :]
```

Out[83]:

	State FIPS	County FIPS	Postal	Name	Poverty Estimate All Ages	Poverty Estimate Under Age 18	Poverty Estimate Ages 5-17	Median Household Income	Poverty Estimate Ages 0-4
327	10	005	DE	Sussex County	26,924	9,501	6,123	48,582	NaN
328	11	000	DC	District of Columbia	107,279	31,147	20,872	60,729	9,786
329	11	001	DC	District of Columbia	107,279	31,147	20,872	60,729	NaN

Nevermind. DC is in there alone and summed (as if it was a state). Just drop all County FIPS 000.

Drop all rows with county FIPS 000. These are just states summed. We don't need them. Might be useful to look at for any missing information later though.

In [84]:
```
drop = df.loc[(df['County FIPS'] == '000')]
```

In [85]:
```
df = df.drop(drop.index, axis = 0)
```

Check on the Hawaii counties. May need to drop Kalawao (no election results. Very low pop)
Change states to full words. Look for Puerto Rico. Strip county, borough, etc.

In [86]:
```
Kalawao = (pd.DataFrame(df.loc[561]))
df = df.drop(Kalawao)
```

Change state abbreviations to full names.

In [87]:

```
us_state_abbrev = {
    'AL': 'Alabama',
    'AK': 'Alaska',
    'AZ': 'Arizona',
    'AR': 'Arkansas',
    'CA': 'California',
    'CO': 'Colorado',
    'CT': 'Connecticut',
    'DE': 'Delaware',
    'FL': 'Florida',
    'GA': 'Georgia',
    'HI': 'Hawaii',
    'ID': 'Idaho',
    'IL': 'Illinois',
    'IN': 'Indiana',
    'IA': 'Iowa',
    'KS': 'Kansas',
    'KY': 'Kentucky',
    'LA': 'Louisiana',
    'ME': 'Maine',
    'MD': 'Maryland',
    'MA': 'Massachusetts',
    'MI': 'Michigan',
    'MN': 'Minnesota',
    'MS': 'Mississippi',
    'MO': 'Missouri',
    'MT': 'Montana',
    'NE': 'Nebraska',
    'NV': 'Nevada',
    'NH': 'New Hampshire',
    'NJ': 'New Jersey',
    'NM': 'New Mexico',
    'NY': 'New York',
    'NC': 'North Carolina',
    'ND': 'North Dakota',
    'OH': 'Ohio',
    'OK': 'Oklahoma',
    'OR': 'Oregon',
    'PA': 'Pennsylvania',
    'RI': 'Rhode Island',
    'SC': 'South Carolina',
    'SD': 'South Dakota',
    'TN': 'Tennessee',
    'TX': 'Texas',
    'UT': 'Utah',
    'VT': 'Vermont',
    'VA': 'Virginia',
    'WA': 'Washington',
    'WV': 'West Virginia',
    'WI': 'Wisconsin',
    'WY': 'Wyoming',
}
```

In [88]:

```
df.Postal = df.Postal.map(us_state_abbrev)
```

Drop unwanted columns.

In [89]:
```
df = df.drop(df.iloc[:, [0, 1, 6, 8]], axis = 1)
```

Chugach and Copper River are already combined into Valdez-Cordova so no need to change that. Renaming Wade Hampton to Kusilvak to match results.

In [90]:
```
df.loc[96]['Name'] = 'Kusilvak'
```

Remove punctuation.

In [91]:
```
def remove_punctuation(x):
    try:
        x = x.str.replace('[^\w\s]','')
    except:
        pass
    return x
df = df.apply(remove_punctuation)
```

```
<ipython-input-91-8e12971db89d>:3: FutureWarning: The default value of regex wi
ll change from True to False in a future version.
  x = x.str.replace('[^\w\s]','')
```

Rename columns.

In [92]:
```
df.columns = ['state', 'county', 'poverty_total', 'poverty_under_18', 'median_hou
```

Sort and reset index.

In [93]:
```
df = df.sort_values(by = ['state', 'county']).reset_index(drop = True)
```

Drop empty rows at the bottom.

In [94]:
```
drop = df.loc[3142:]
```

In [95]:
```
df = df.drop(drop.index, axis = 0)
```

Change column types.

In [96]:
```
df[df.columns[2:]] = df[df.columns[2:]].astype(float)
```

Bedford, Virginia was a city and a county during this census. They were not separated for the elections. *** come back and fix this median household income. This is not correct.

```
In [97]: bedfords = df.loc[2828:2829]
```

```
In [98]: new_bedford = pd.DataFrame(bedfords.sum(numeric_only = False, axis = 0)).T
```

```
In [99]: new_bedford['state'] = 'Virginia'
         new_bedford['county'] = 'Bedford'
         new_bedford['median_household_income'] = 43660
```

```
In [100]: df.drop(bedfords.index, axis = 0, inplace = True)
```

```
In [101]: df = df.append(new_bedford)
```

Fill state name for DC.

```
In [102]: df.loc[3141, 'state'] = 'District of Columbia'
```

```
In [103]: df.loc[3141]
```

```
Out[103]: state                          District of Columbia
          county                         District of Columbia
          poverty_total                              107279.0
          poverty_under_18                            31147.0
          median_household_income                     60729.0
          Name: 3141, dtype: object
```

```
In [104]: df.loc[df.county == 'District of Columbia']
```

Out[104]:

	state	county	poverty_total	poverty_under_18	median_household_income
3141	District of Columbia	District of Columbia	107279.0	31147.0	60729.0

Rename Shannon County, SD to Oglala Lakota County to match other datasets.

```
In [105]: df.loc[2416, 'county'] = 'Oglala Lakota'
```

Change the datatypes again. Maybe something during the Virginia Bedfords append messed it up?

```
In [106]: df[df.columns[2:]] = df[df.columns[2:]].astype(float)
```

```
In [107]: poverty_df = df
```

In [108]:

```
poverty_df.head()
```

Out[108]:

	state	county	poverty_total	poverty_under_18	median_household_income
0	Alabama	Autauga County	6459.0	2530.0	53049.0
1	Alabama	Baldwin County	24056.0	8357.0	47618.0
2	Alabama	Barbour County	6098.0	2145.0	33074.0
3	Alabama	Bibb County	4316.0	1448.0	35472.0
4	Alabama	Blount County	9358.0	3356.0	42906.0

Merge race and poverty census data with results

poverty_df, df_race, total_results

In [109]:

```
poverty_df = poverty_df.sort_values(by = ['state', 'county']).reset_index(drop =
```

In [110]:

```
poverty_df.head()
```

Out[110]:

	state	county	poverty_total	poverty_under_18	median_household_income
0	Alabama	Autauga County	6459.0	2530.0	53049.0
1	Alabama	Baldwin County	24056.0	8357.0	47618.0
2	Alabama	Barbour County	6098.0	2145.0	33074.0
3	Alabama	Bibb County	4316.0	1448.0	35472.0
4	Alabama	Blount County	9358.0	3356.0	42906.0

In [111]:

```
poverty_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3141 entries, 0 to 3140
Data columns (total 5 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   state                    3141 non-null   object 
 1   county                   3141 non-null   object 
 2   poverty_total            3141 non-null   float64
 3   poverty_under_18         3141 non-null   float64
 4   median_household_income  3141 non-null   float64
dtypes: float64(3), object(2)
memory usage: 122.8+ KB
```

In [112]:

```
df_race = df_race.sort_values(by = ['State', 'County']).reset_index(drop = True)
```

In [113]:
```
df_race.head()
```

Out[113]:

1	id	total_pop	total_pop_one_race	pop_white	pop_african_american	pop_native	p
0	0500000US01001	58805.0	55648.0	42160.0	11445.0	217.0	
1	0500000US01003	231767.0	216743.0	189399.0	18217.0	1582.0	
2	0500000US01005	25223.0	24523.0	11317.0	11933.0	116.0	
3	0500000US01007	22293.0	21534.0	16555.0	4413.0	60.0	
4	0500000US01009	59134.0	55478.0	50663.0	845.0	337.0	

In [114]:
```
df_race.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3141 entries, 0 to 3140
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    3141 non-null   object 
 1   total_pop             3141 non-null   float64
 2   total_pop_one_race    3141 non-null   float64
 3   pop_white             3141 non-null   float64
 4   pop_african_american  3141 non-null   float64
 5   pop_native            3141 non-null   float64
 6   pop_asian             3141 non-null   float64
 7   pop_islander          3141 non-null   float64
 8   pop_other             3141 non-null   float64
 9   total_pop_two_races   3141 non-null   float64
 10  County                3141 non-null   object 
 11  State                 3141 non-null   object 
dtypes: float64(9), object(3)
memory usage: 294.6+ KB
```

In [115]:
```
total_results = total_results.sort_values(by = ['State', 'County']).reset_index(
```

In [116]:
```
total_results.head()
```

Out[116]:

	State	County	2016_total_votes	Obama	Romney	2012_total_votes	2010_land_area	Density
0	Alabama	Autauga County	24661.0	6354.0	17366.0	23909.0	594	93.0
1	Alabama	Baldwin County	94090.0	18329.0	65772.0	84988.0	1590	128.0
2	Alabama	Barbour County	10390.0	5873.0	5539.0	11459.0	885	30.0
3	Alabama	Bibb County	8748.0	2200.0	6131.0	8391.0	623	36.0
4	Alabama	Blount County	25384.0	2961.0	20741.0	23980.0	645	89.0

In [117]:
```
total_results.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3141 entries, 0 to 3140
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   State             3141 non-null   object 
 1   County            3141 non-null   object 
 2   2016_total_votes  3141 non-null   float64
 3   Obama             3141 non-null   float64
 4   Romney            3141 non-null   float64
 5   2012_total_votes  3141 non-null   float64
 6   2010_land_area    3141 non-null   int64  
 7   Density           3141 non-null   float64
 8   central_outlying  3141 non-null   object 
 9   Target            3141 non-null   object 
dtypes: float64(5), int64(1), object(4)
memory usage: 245.5+ KB
```

In [118]:
```
df_race_results = df_race.reset_index(drop=True).merge(total_results.reset_index(
```

In [127]:
```
df_race_results.loc[:, ['County_x', 'County_y', ]].tail()
```

Out[127]:

	County_x	County_y
3136	Sweetwater County	Sweetwater County
3137	Teton County	Teton County
3138	Uinta County	Uinta County
3139	Washakie County	Washakie County
3140	Weston County	Weston County

In [120]:
```
df_poverty_race_results = df_race_results.reset_index(drop=True).merge(poverty_d
```

In [121]:
```
df_poverty_race_results.head()
```

Out[121]:

	id	total_pop	total_pop_one_race	pop_white	pop_african_american	pop_native	p
0	0500000US01001	58805.0	55648.0	42160.0	11445.0	217.0	
1	0500000US01003	231767.0	216743.0	189399.0	18217.0	1582.0	
2	0500000US01005	25223.0	24523.0	11317.0	11933.0	116.0	
3	0500000US01007	22293.0	21534.0	16555.0	4413.0	60.0	
4	0500000US01009	59134.0	55478.0	50663.0	845.0	337.0	

In [126]:
```
df_poverty_race_results.loc[:, ['County_x', 'County_y', 'county']].tail()
```

Out[126]:

	County_x	County_y	county
3136	Sweetwater County	Sweetwater County	Sweetwater County
3137	Teton County	Teton County	Teton County
3138	Uinta County	Uinta County	Uinta County
3139	Washakie County	Washakie County	Washakie County
3140	Weston County	Weston County	Weston County

In [123]:
```
df_all = df_poverty_race_results
```

In [124]:
```
dump(df_all, open('df_all.pkl', 'wb'))
```

In [ ]: