In [1]:

```
import pandas as pd
import numpy as np
import scipy.stats as st
import seaborn as sns
from collections import Counter
```

In [2]:

```
data = pd.read_csv("election_train.csv")
data2 = pd.read_csv("demographics_train.csv")
```

1 (5 pts.) Reshape dataset election_train from long format to wide format. Hint: the reshaped dataset should contain 1205 rows and 6 columns.

In [3]:

```
data_tidy = pd.pivot_table(data, index = ['Year','State','County','Office'], values = 'Votes',co
lumns = 'Party').reset_index()
data_tidy
```

Out[3]:

Party	Year	State	County	Office	Democratic	Republican
0	2018	AZ	Apache County	US Senator	16298.0	7810.0
1	2018	AZ	Cochise County	US Senator	17383.0	26929.0
2	2018	AZ	Coconino County	US Senator	34240.0	19249.0
3	2018	AZ	Gila County	US Senator	7643.0	12180.0
4	2018	AZ	Graham County	US Senator	3368.0	6870.0
...	...	...	...	...	...	...
1200	2018	WY	Platte County	US Senator	801.0	2850.0
1201	2018	WY	Sublette County	US Senator	668.0	2653.0
1202	2018	WY	Sweetwater County	US Senator	3943.0	8577.0
1203	2018	WY	Uinta County	US Senator	1371.0	4713.0
1204	2018	WY	Washakie County	US Senator	588.0	2423.0

1205 rows × 6 columns

2 (20 pts.) Merge reshaped dataset election_train with dataset demographics_train. Make sure that you address all inconsistencies in the names of the states and the counties before merging. Hint: the merged dataset should contain 1200 rows.

In [4]:

```
data_tidy['County'] = data_tidy['County'].str.replace('County','').str.lower().str.strip()
```

In [5]:

```
change_values = {
    'Alabama': 'AL',
    'Alaska': 'AK',
    'American Samoa': 'AS',
    'Arizona': 'AZ',
    'Arkansas': 'AR',
    'California': 'CA',
    'Colorado': 'CO',
    'Connecticut': 'CT',
    'Delaware': 'DE',
    'District of Columbia': 'DC',
    'Florida': 'FL',
    'Georgia': 'GA',
    'Guam': 'GU',
    'Hawaii': 'HI',
    'Idaho': 'ID',
    'Illinois': 'IL',
    'Indiana': 'IN',
    'Iowa': 'IA',
    'Kansas': 'KS',
    'Kentucky': 'KY',
    'Louisiana': 'LA',
    'Maine': 'ME',
    'Maryland': 'MD',
    'Massachusetts': 'MA',
    'Michigan': 'MI',
    'Minnesota': 'MN',
    'Mississippi': 'MS',
    'Missouri': 'MO',
    'Montana': 'MT',
    'Nebraska': 'NE',
    'Nevada': 'NV',
    'New Hampshire': 'NH',
    'New Jersey': 'NJ',
    'New Mexico': 'NM',
    'New York': 'NY',
    'North Carolina': 'NC',
    'North Dakota': 'ND',
    'Northern Mariana Islands':'MP',
    'Ohio': 'OH',
    'Oklahoma': 'OK',
    'Oregon': 'OR',
    'Pennsylvania': 'PA',
    'Puerto Rico': 'PR',
    'Rhode Island': 'RI',
    'South Carolina': 'SC',
    'South Dakota': 'SD',
    'Tennessee': 'TN',
    'Texas': 'TX',
    'Utah': 'UT',
    'Vermont': 'VT',
    'Virgin Islands': 'VI',
    'Virginia': 'VA',
    'Washington': 'WA',
    'West Virginia': 'WV',
    'Wisconsin': 'WI',
    'Wyoming': 'WY'
}
data2['State'] = data2['State'].map(change_values)
```

In [6]:

```
data2['County']=data2['County'].str.lower().str.strip()
data_tidy['State']=data_tidy['State']
```

In [7]:

```
data_merged = pd.merge(data_tidy,data2, how='inner',on = ['State','County'])
```

In [8]:

```
data_merged
```

Out[8]:

	Year	State	County	Office	Democratic	Republican	FIPS	Total Population	Citize Votin Ag Populatio
0	2018	AZ	apache	US Senator	16298.0	7810.0	4001	72346	
1	2018	AZ	cochise	US Senator	17383.0	26929.0	4003	128177	9291
2	2018	AZ	coconino	US Senator	34240.0	19249.0	4005	138064	10426
3	2018	AZ	gila	US Senator	7643.0	12180.0	4007	53179	
4	2018	AZ	graham	US Senator	3368.0	6870.0	4009	37529	
...	...	...	...	...	...	...	...	...	
1195	2018	WY	platte	US Senator	801.0	2850.0	56031	8740	683
1196	2018	WY	sublette	US Senator	668.0	2653.0	56035	10032	
1197	2018	WY	sweetwater	US Senator	3943.0	8577.0	56037	44812	3056
1198	2018	WY	uinta	US Senator	1371.0	4713.0	56041	20893	1435
1199	2018	WY	washakie	US Senator	588.0	2423.0	56043	8351	

1200 rows × 21 columns

3 (5 pts.) Explore the merged dataset. How many variables does the dataset have? What is the type of these variables? Are there any irrelevant or redundant variables? If so, how will you deal with these variables?

In [9]:

```
data_merged.dtypes
```

Out[9]:

```
Year                                    int64
State                                  object
County                                 object
Office                                 object
Democratic                            float64
Republican                            float64
FIPS                                    int64
Total Population                        int64
Citizen Voting-Age Population           int64
Percent White, not Hispanic or Latino  float64
Percent Black, not Hispanic or Latino  float64
Percent Hispanic or Latino             float64
Percent Foreign Born                   float64
Percent Female                         float64
Percent Age 29 and Under               float64
Percent Age 65 and Older               float64
Median Household Income                 int64
Percent Unemployed                     float64
Percent Less than High School Degree   float64
Percent Less than Bachelor's Degree    float64
Percent Rural                          float64
dtype: object
```

In [10]:

```
for i in data_merged:
    print(i,len(data_merged[i].unique()))
a=Counter(data_merged['Citizen Voting-Age Population'])[0]
print("0 in Citizen Voting-Age Population: " + str(a))
```

```
Year 1
State 30
County 881
Office 1
Democratic 1143
Republican 1161
FIPS 1200
Total Population 1190
Citizen Voting-Age Population 513
Percent White, not Hispanic or Latino 1200
Percent Black, not Hispanic or Latino 1155
Percent Hispanic or Latino 1196
Percent Foreign Born 1197
Percent Female 1199
Percent Age 29 and Under 1200
Percent Age 65 and Older 1200
Median Household Income 1181
Percent Unemployed 1195
Percent Less than High School Degree 1200
Percent Less than Bachelor's Degree 1200
Percent Rural 945
0 in Citizen Voting-Age Population: 680
```

In [11]:

```
data_merged=data_merged.drop(['Year', 'Office', 'Citizen Voting-Age Population'], axis=1)
```

In [12]:

```
data_merged
```

Out[12]:

	State	County	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	I H o
0	AZ	apache	16298.0	7810.0	4001	72346	18.571863	0.486551	5.
1	AZ	cochise	17383.0	26929.0	4003	128177	56.299492	3.714395	34.
2	AZ	coconino	34240.0	19249.0	4005	138064	54.619597	1.342855	13
3	AZ	gila	7643.0	12180.0	4007	53179	63.222325	0.552850	18.
4	AZ	graham	3368.0	6870.0	4009	37529	51.461536	1.811932	32.
...	...	...	...	...	...	...	...	...	
1195	WY	platte	801.0	2850.0	56031	8740	89.359268	0.057208	7.
1196	WY	sublette	668.0	2653.0	56035	10032	91.646730	0.000000	7.
1197	WY	sweetwater	3943.0	8577.0	56037	44812	79.815674	0.865840	15.
1198	WY	uinta	1371.0	4713.0	56041	20893	87.718375	0.186665	8.
1199	WY	washakie	588.0	2423.0	56043	8351	82.397318	0.790325	13.

1200 rows × 18 columns

Answer:

21 variables, float64(13), int64(5), object(3), 'Year' and 'Office' are irrelevant they only have one obeservation,'Citizen Voting-Age Population' has too many 0 we can drop them.

4 (10 pts.) Search the merged dataset for missing values. Are there any missing values? If so, how will you deal with these values?

In [13]:

```
data_merged.isnull().sum()
```

Out[13]:

```
State                                  0
County                                 0
Democratic                             3
Republican                             2
FIPS                                   0
Total Population                       0
Percent White, not Hispanic or Latino  0
Percent Black, not Hispanic or Latino  0
Percent Hispanic or Latino             0
Percent Foreign Born                   0
Percent Female                         0
Percent Age 29 and Under               0
Percent Age 65 and Older               0
Median Household Income                0
Percent Unemployed                     0
Percent Less than High School Degree   0
Percent Less than Bachelor's Degree    0
Percent Rural                          0
dtype: int64
```

In [14]:

```
data_merged=data_merged.fillna(0)
```

Answer:

There are 3 missing value in Democratic and 2 missing value in Republican, since there are not that much, we choice to set 0 them.

5 (5 pts.) Create a new variable named “Party” that labels each county as Democratic or Republican. This new variable should be equal to 1 if there were more votes cast for the Democratic party than the Republican party in that county and it should be equal to 0 otherwise.

In [15]:

```
data_merged['Party'] =  data_merged.apply(lambda row:1 if row.Democratic > row.Republican else
0,axis=1)
```

In [16]:

```
data_merged
```

Out[16]:

	State	County	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	I H or
0	AZ	apache	16298.0	7810.0	4001	72346	18.571863	0.486551	5.
1	AZ	cochise	17383.0	26929.0	4003	128177	56.299492	3.714395	34.
2	AZ	coconino	34240.0	19249.0	4005	138064	54.619597	1.342855	13
3	AZ	gila	7643.0	12180.0	4007	53179	63.222325	0.552850	18.
4	AZ	graham	3368.0	6870.0	4009	37529	51.461536	1.811932	32.
...	...	...	...	...	...	...	...	...	
1195	WY	platte	801.0	2850.0	56031	8740	89.359268	0.057208	7.
1196	WY	sublette	668.0	2653.0	56035	10032	91.646730	0.000000	7.
1197	WY	sweetwater	3943.0	8577.0	56037	44812	79.815674	0.865840	15.
1198	WY	uinta	1371.0	4713.0	56041	20893	87.718375	0.186665	8.
1199	WY	washakie	588.0	2423.0	56043	8351	82.397318	0.790325	13.

1200 rows × 19 columns

6 (10 pts.) Compute the mean median household income for Democratic counties and Republican counties. Which one is higher? Perform a hypothesis test to determine whether this difference is statistically significant at the $\alpha = 0.05$ significance level. What is the result of the test? What conclusion do you make from this result?

In [17]:

```
da=data_merged.groupby('Party')
da['Median Household Income'].describe()
```

Out[17]:

Party	count	mean	std	min	25%	50%	75%	max
0	873.0	48724.615120	10669.835532	24000.0	41478.0	47163.0	53432.0	108177.0
1	327.0	53766.455657	15251.831306	21190.0	44138.0	51477.0	59075.0	125672.0

In [18]:

```
Democratic=data_merged.loc[data_merged['Party'] == 1]
Republican=data_merged.loc[data_merged['Party'] == 0]
[statistic, pvalue] = st.ttest_ind(Republican['Median Household Income'],Democratic['Median Hous
ehold Income'],equal_var = False)
print(pvalue)
```

6.536254891102229e-08

Answer:

Democratic is higher. p-value is less than 0.05 which means there is sufficient data to reject the null hypothesis that mean median household income of Republican contries and Republican counties are equal.

7 (10 pts.) Compute the mean population for Democratic counties and Republican counties. Which one is higher? Perform a hypothesis test to determine whether this difference is statistically significant at the $\alpha = 0.05$ significance level. What is the result of the test? What conclusion do you make from this result?

In [19]:

```
da['Total Population'].describe()
```

Out[19]:

Party	count	mean	std	min	25%	50%	75%	max
0	873.0	54041.167239	94431.046253	76.0	9554.0	25403.0	53808.0	1092518.0
1	327.0	299308.721713	552321.003945	1969.0	22988.5	81505.0	278375.0	4434257.0

In [20]:

```
[statistic, pvalue] = st.ttest_ind(Republican['Total Population'],Democratic['Total Population'
],equal_var = False)
print(pvalue)
```

2.2795809094677384e-14

Answer:

Democratic counties is higher. p-value is less than 0.05 which means there is sufficient data to reject the null hypothesis that population of Republican contries and Republican counties are equal.

8 (20 pts.) Compare Democratic counties and Republican counties in terms of age, gender, race and ethnicity, and education by computing descriptive statistics and creating plots to visualize the results. What conclusions do you make for each variable from the descriptive statistics and the plots?

Answer:

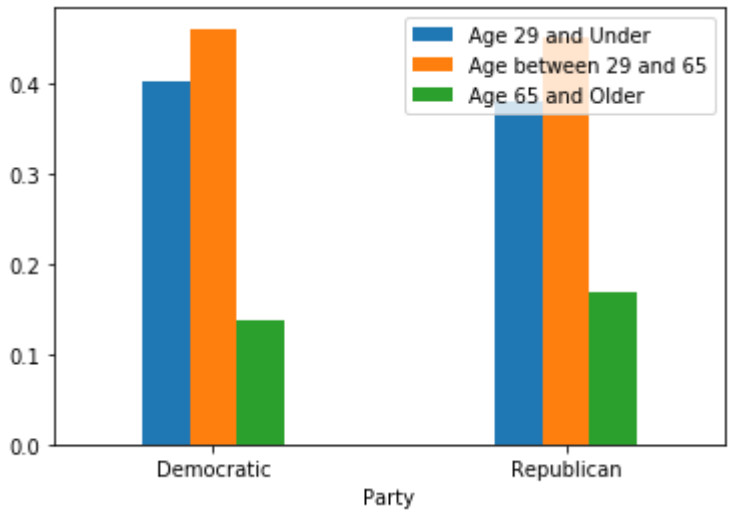
from the age, Democratic counties have more percent of people whose age is 29 and under, and have less people whose age is 65 and above than that in Republican counties. from the gender, the percent of female in Democratic counties is a litter higher. from the race and ethnicity, the percent of White people in Democratic counties is less than that in Republican counties from the education, the percent of people in Democratic counties who has a degree more than Bachelor is more than that in Republican counties

In [21]:

```
data_Democratic = data_merged.loc[data_merged.Party == 1]
data_Republican = data_merged.loc[data_merged.Party == 0]
a = [{'Total Population' : sum(data_Democratic['Total Population'])}]
data_age = pd.DataFrame(a)
data_age['Age 65 and Older'] = sum(data_Democratic['Percent Age 65 and Older']/100*data_Democrat
ic['Total Population'])
data_age['Age 29 and Under'] = sum(data_Democratic['Percent Age 29 and Under']/100*data_Democrat
ic['Total Population'])
data_age['Age between 29 and 65'] = (data_age['Total Population'] - data_age['Age 65 and Older']
- data_age['Age 29 and Under'])
data_age['Party'] = 'Democratic'
data_age = data_age.append({'Total Population':sum(data_Republican['Total Population']),
                            'Age 65 and Older':sum(data_Republican['Percent Age 65 and Older']/1
00*data_Republican['Total Population']),
                            'Age 29 and Under':sum(data_Republican['Percent Age 29 and Under']/1
00*data_Republican['Total Population']),
                            'Party':'Republican'},ignore_index=True)
data_age['Age between 29 and 65']  = data_age['Total Population'] - data_age['Age 65 and Older']
- data_age['Age 29 and Under']
data_age['Age 65 and Older'] = data_age['Age 65 and Older']/data_age['Total Population']
data_age['Age between 29 and 65'] = data_age['Age between 29 and 65']/data_age['Total Populatio
n']
data_age['Age 29 and Under'] = data_age['Age 29 and Under']/data_age['Total Population']
data_age.plot.bar(x='Party', y=['Age 29 and Under','Age between 29 and 65','Age 65 and Older'],
width=0.4, rot=0)
```

Out[21]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xb1aa948>
```

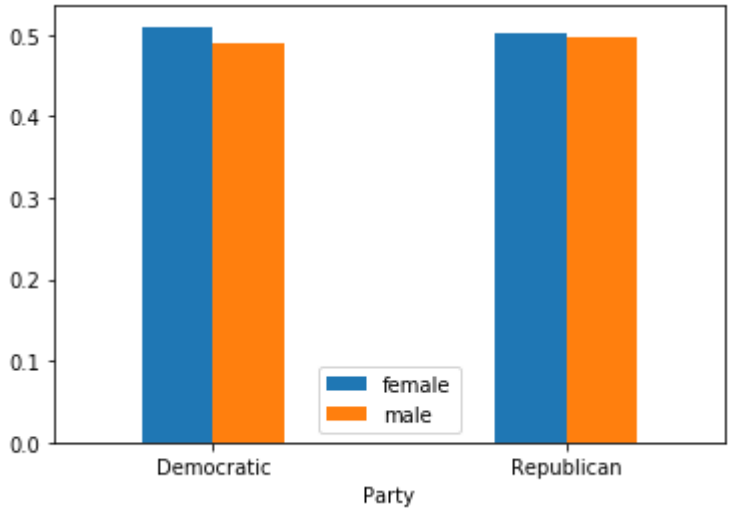


In [22]:

```
a = [{'Total Population' : sum(data_Democratic['Total Population'])}]
data_gender = pd.DataFrame(a)
data_gender['female'] = sum(data_Democratic['Percent Female']/100*data_Democratic['Total Populat
ion'])
data_gender['male'] = data_gender['Total Population'] - data_gender['female']
data_gender['Party'] = 'Democratic'
data_gender = data_gender.append({'Total Population':sum(data_Republican['Total Population']),
                                  'female':sum(data_Republican['Percent Female']/100*data_Republ
ican['Total Population']),
                                  'Party':'Republican'}, ignore_index=True)
data_gender['male']  = data_gender['Total Population'] - data_gender['female']
data_gender['male'] = data_gender['male']/data_gender['Total Population']
data_gender['female'] = data_gender['female']/data_gender['Total Population']
data_gender.plot.bar(x='Party', y=['female','male'], width=0.4, rot=0)
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xb71f6c8>
```

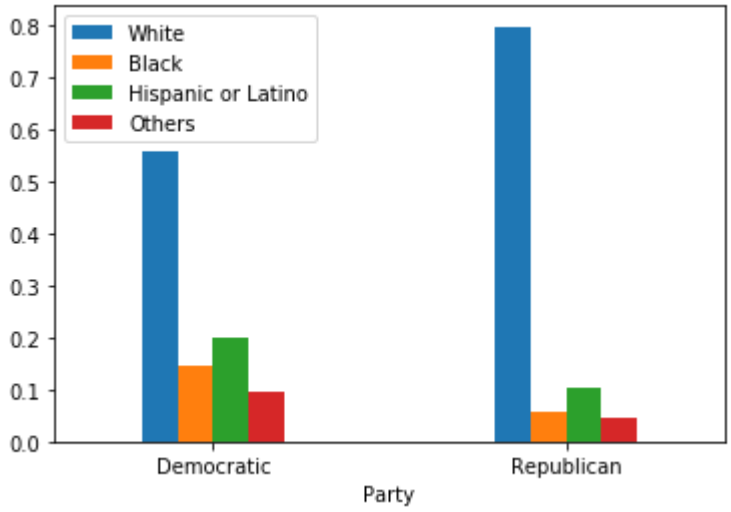


In [23]:

```
data_race = pd.DataFrame(a)
data_race['White'] = sum(data_Democratic['Percent White, not Hispanic or Latino']/100*data_Democ
ratic['Total Population'])
data_race['Black'] = sum(data_Democratic['Percent Black, not Hispanic or Latino']/100*data_Democ
ratic['Total Population'])
data_race['Hispanic or Latino'] = sum(data_Democratic['Percent Hispanic or Latino']/100*data_Dem
ocratic['Total Population'])
data_race['Others'] = (data_race['Total Population'] - data_race['White'] - data_race['Black'] -
data_race['Hispanic or Latino'])
data_race['Party'] = 'Democratic'
data_race = data_race.append({'Total Population':sum(data_Republican['Total Population']),
                              'White':sum(data_Republican['Percent White, not Hispanic or Latin
o']/100*data_Republican['Total Population']),
                              'Black':sum(data_Republican['Percent Black, not Hispanic or Latin
o']/100*data_Republican['Total Population']),
                              'Hispanic or Latino':sum(data_Republican['Percent Hispanic or Lati
no']/100*data_Republican['Total Population']),
                              'Party':'Republican'},ignore_index=True)
data_race['Others']  = (data_race['Total Population'] - data_race['White'] - data_race['Black']
- data_race['Hispanic or Latino'])
data_race['White'] = data_race['White']/data_race['Total Population']
data_race['Black'] = data_race['Black']/data_race['Total Population']
data_race['Hispanic or Latino'] = data_race['Hispanic or Latino']/data_race['Total Population']
data_race['Others'] = data_race['Others']/data_race['Total Population']
data_race.plot.bar(x='Party', y=['White','Black','Hispanic or Latino','Others'], width=0.4, rot=
0)
```

Out[23]:

<matplotlib.axes._subplots.AxesSubplot at 0xb99ecc8>

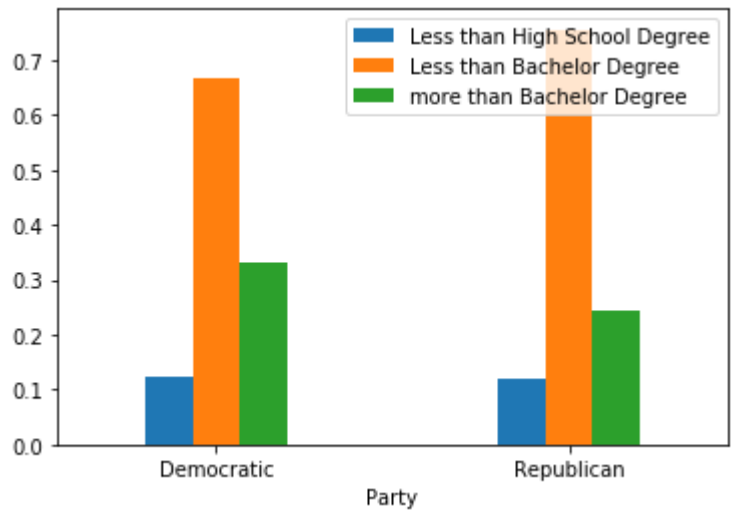


In [24]:

```
data_education = pd.DataFrame(a)
data_education['Less than High School Degree'] = sum(data_Democratic['Percent Less than High School Degree']/100*data_Democratic['Total Population'])
data_education['Less than Bachelor Degree'] = sum(data_Democratic["Percent Less than Bachelor's Degree"]/100*data_Democratic['Total Population'])
data_education['more than Bachelor Degree'] = data_education['Total Population'] - data_education['Less than Bachelor Degree']
data_education['Party'] = 'Democratic'
data_education = data_education.append({'Total Population':sum(data_Republican['Total Population']),
                                        'Less than High School Degree':sum(data_Republican['Percent Less than High School Degree']/100*data_Republican['Total Population']),
                                        'Less than Bachelor Degree':sum(data_Republican["Percent Less than Bachelor's Degree"]/100*data_Republican['Total Population']),
                                        'Party':'Republican'},ignore_index=True)
data_education['more than Bachelor Degree']  = data_education['Total Population'] - data_education['Less than Bachelor Degree']
data_education['Less than High School Degree'] = data_education['Less than High School Degree']/data_education['Total Population']
data_education['Less than Bachelor Degree'] = data_education['Less than Bachelor Degree']/data_education['Total Population']
data_education['more than Bachelor Degree'] = data_education['more than Bachelor Degree']/data_education['Total Population']
data_education.plot.bar(x='Party', y=['Less than High School Degree','Less than Bachelor Degree','more than Bachelor Degree'], width=0.4, rot=0)
```

Out[24]:

<matplotlib.axes._subplots.AxesSubplot at 0xba202c8>



9 (5 pts.) Based on your results for tasks 6-8, which variables in the dataset do you think are more important to determine whether a county is labeled as Democratic or Republican? Justify your answer.

Answer:

'Median Household Income', 'Total population' and 'race and ethnicity' are important to determine a county is labeled as Democratic or Republican. they have significant difference from different parties. Education, gender and age are almost same between different parties which means they are not affected by the paries but maybe be affected by different counties.

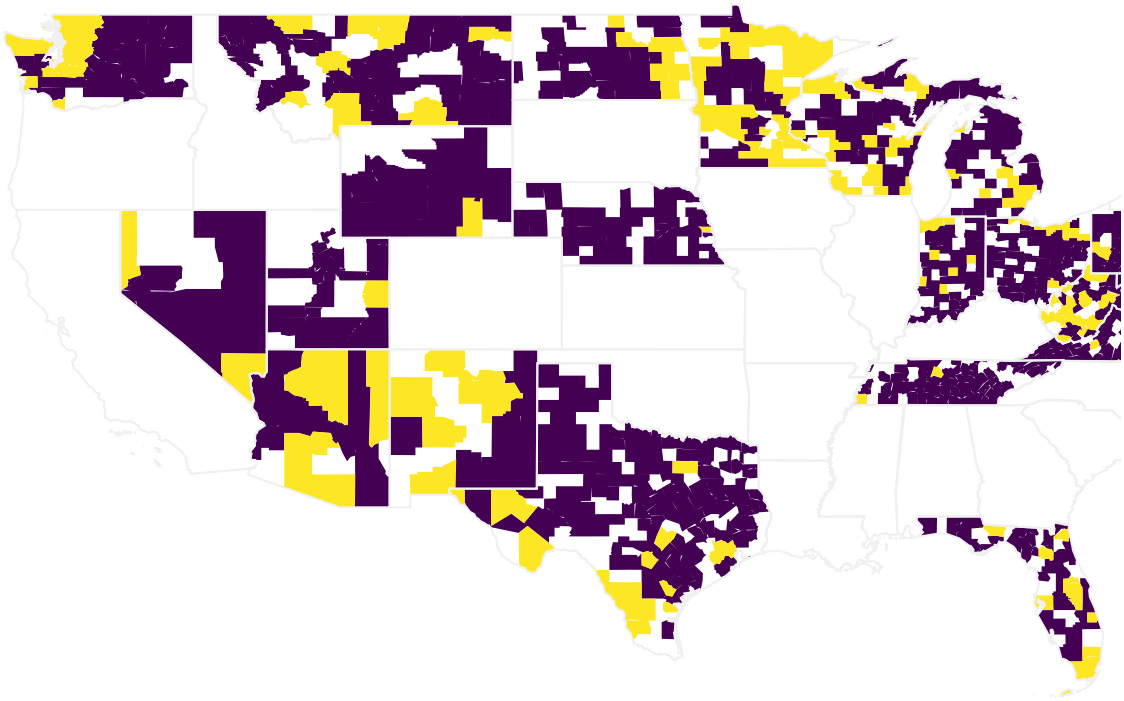
10 (10 pts.) Create a map of Democratic counties and Republican counties using the counties' FIPS codes and Python's Plotly library (plot.ly/python/county-choropleth/). Note that this dataset does not include all United States counties.

In [25]:

```
import plotly.figure_factory as ff
import plotly

fips = data_merged['FIPS'].tolist()
values =data_merged['Party'].tolist()

fig = ff.create_choropleth(fips=fips, values=values)
fig.layout.template = None
fig.show()
```



In [ ]:

```

```