Exploratory Data Analysis on Senate Election 2018 ¶

In [1]:

```
#Importing necessary libraries
import pandas as pd
import numpy as np
import scipy.stats as st
import plotly.figure_factory as ff
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
# Reading in unclean data from demograpics train file
data_a = pd.read_csv("demographics_train.csv")
```

In [3]:

```
# Reading in unclean data from election train file
data_b = pd.read_csv("election_train.csv")
```

In [4]:

```
# Temporarily fill in the missing values with 0.
fill_with_zero= data_b.fillna(0)
```

In [5]:

```
data_b_tidy = pd.pivot_table(fill_with_zero, index=['Year', 'State','County', 'Office'
],columns='Party', values="Votes").reset_index()
print(data_b_tidy.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1205 entries, 0 to 1204
Data columns (total 6 columns):
Year          1205 non-null int64
State         1205 non-null object
County        1205 non-null object
Office        1205 non-null object
Democratic    1205 non-null float64
Republican    1205 non-null float64
dtypes: float64(2), int64(1), object(3)
memory usage: 56.6+ KB
None
```

In [6]:

```
# For dataset B, for the state column, change
change_values_state = {'AL' : 'Alabama', 'AK' : 'Alaska', 'AZ' : 'Arizona', 'AR' : 'Ark
ansas', 'CA' : 'California', 'CO' : 'Colorado', 'CT' : 'Connecticut', 'DE' : 'Delaware'
, 'FL' : 'Florida', 'GA' : 'Georgia', 'HI' : 'Hawaii', 'ID' : 'Idaho', 'IL' : 'Illinoi
s', 'IN' : 'Indiana', 'IA' : 'Iowa', 'KS' : 'Kansas', 'KY' : 'Kentucky', 'LA' : 'Louisi
ana', 'ME' : 'Maine', 'MD' : 'Maryland', 'MA' : 'Massachusetts', 'MI' : 'Michigan', 'M
N' : 'Minnesota', 'MS' : 'Mississippi', 'MO' : 'Missouri', 'MT' : 'Montana', 'NE' : 'Ne
braska', 'NV' : 'Nevada', 'NH' : 'New Hampshire', 'NJ' : 'New Jersey', 'NM' : 'New Mexi
co', '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 Virgini
a', 'WI' : 'Wisconsin', 'WY' : 'Wyoming'}
data_b_tidy['State'] = data_b_tidy["State"].map(change_values_state)
# print(data_b_tidy)
```

In [7]:

```
data_b_tidy['County'] = data_b_tidy['County'].map(lambda x: x.replace(' County',''))
# print(data_b_tidy)
```

In [8]:

```
# Remove inconsistencies in case from both files
data_a['County'] = data_a['County'].str.upper()
data_b_tidy['County'] = data_b_tidy['County'].str.upper()
```

In [9]:

```
# Merge dataset A and dataset B
merged_data_set = pd.merge(data_a,data_b_tidy, how='inner', on=['State', 'County'])
print(merged_data_set.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1200 entries, 0 to 1199
Data columns (total 21 columns):
State                                   1200 non-null object
County                                  1200 non-null object
FIPS                                    1200 non-null int64
Total Population                        1200 non-null int64
Citizen Voting-Age Population           1200 non-null int64
Percent White, not Hispanic or Latino   1200 non-null float64
Percent Black, not Hispanic or Latino   1200 non-null float64
Percent Hispanic or Latino              1200 non-null float64
Percent Foreign Born                    1200 non-null float64
Percent Female                          1200 non-null float64
Percent Age 29 and Under                1200 non-null float64
Percent Age 65 and Older                1200 non-null float64
Median Household Income                 1200 non-null int64
Percent Unemployed                      1200 non-null float64
Percent Less than High School Degree    1200 non-null float64
Percent Less than Bachelor's Degree     1200 non-null float64
Percent Rural                           1200 non-null float64
Year                                    1200 non-null int64
Office                                  1200 non-null object
Democratic                              1200 non-null float64
Republican                              1200 non-null float64
dtypes: float64(13), int64(5), object(3)
memory usage: 206.2+ KB
None
```

In [10]:

```
# The Dataset has 21 variables
# The variables in this dataset has types object or int64 or float64
# The irrelevant irrelevant are Year because all the observations are from the same yea
r 2018, Office which contains same value US Senator
# We decided to drop these variables since they do not provide a any useful information
for the current analysis.
merged_data_set = merged_data_set.drop(columns=['Year', 'Office'])
merged_data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1200 entries, 0 to 1199
Data columns (total 19 columns):
State                                  1200 non-null object
County                                 1200 non-null object
FIPS                                   1200 non-null int64
Total Population                       1200 non-null int64
Citizen Voting-Age Population          1200 non-null int64
Percent White, not Hispanic or Latino  1200 non-null float64
Percent Black, not Hispanic or Latino  1200 non-null float64
Percent Hispanic or Latino             1200 non-null float64
Percent Foreign Born                   1200 non-null float64
Percent Female                         1200 non-null float64
Percent Age 29 and Under               1200 non-null float64
Percent Age 65 and Older               1200 non-null float64
Median Household Income                1200 non-null int64
Percent Unemployed                     1200 non-null float64
Percent Less than High School Degree   1200 non-null float64
Percent Less than Bachelor's Degree    1200 non-null float64
Percent Rural                          1200 non-null float64
Democratic                             1200 non-null float64
Republican                             1200 non-null float64
dtypes: float64(13), int64(4), object(2)
memory usage: 187.5+ KB
```

In [11]:

```
#There are missing values in the following variables in the datase.(Citizen Voting-Age
 Population,Democratic,Republican)
#Citizen Voting-Age Population has many 0 values in the variable according to this colu
mn the zero signifies missing value.
#So we decided drop the column since 680 of 1200 observations have 0 in it.(more than 5
0 % of observations)
#There are five observations in the dataset which have both republican and Democratic v
ariables both zero(filled with zero for missing observation)
# These observations will be deleted from the dataset since we cannot decide the party
 of the county based on missing values.
merged_data_set = merged_data_set.drop(columns=['Citizen Voting-Age Population'])
merged_data_set = merged_data_set.drop(merged_data_set[(merged_data_set.Democratic == 0
) & (merged_data_set.Republican == 0)].index)
merged_data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1195 entries, 0 to 1199
Data columns (total 18 columns):
State                                1195 non-null object
County                               1195 non-null object
FIPS                                 1195 non-null int64
Total Population                     1195 non-null int64
Percent White, not Hispanic or Latino    1195 non-null float64
Percent Black, not Hispanic or Latino    1195 non-null float64
Percent Hispanic or Latino           1195 non-null float64
Percent Foreign Born                 1195 non-null float64
Percent Female                       1195 non-null float64
Percent Age 29 and Under             1195 non-null float64
Percent Age 65 and Older             1195 non-null float64
Median Household Income              1195 non-null int64
Percent Unemployed                   1195 non-null float64
Percent Less than High School Degree     1195 non-null float64
Percent Less than Bachelor's Degree      1195 non-null float64
Percent Rural                        1195 non-null float64
Democratic                           1195 non-null float64
Republican                           1195 non-null float64
dtypes: float64(13), int64(3), object(2)
memory usage: 177.4+ KB
```

In [12]:

```
# Created a new column called 'Party' dependending on which party has more votes
merged_data_set['Party'] = np.where(merged_data_set['Democratic'] > merged_data_set['Re
publican'],1,0)
```

In [13]:

```
merged_data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1195 entries, 0 to 1199
Data columns (total 19 columns):
State                                   1195 non-null object
County                                  1195 non-null object
FIPS                                    1195 non-null int64
Total Population                        1195 non-null int64
Percent White, not Hispanic or Latino   1195 non-null float64
Percent Black, not Hispanic or Latino   1195 non-null float64
Percent Hispanic or Latino              1195 non-null float64
Percent Foreign Born                    1195 non-null float64
Percent Female                          1195 non-null float64
Percent Age 29 and Under                1195 non-null float64
Percent Age 65 and Older                1195 non-null float64
Median Household Income                 1195 non-null int64
Percent Unemployed                      1195 non-null float64
Percent Less than High School Degree    1195 non-null float64
Percent Less than Bachelor's Degree     1195 non-null float64
Percent Rural                           1195 non-null float64
Democratic                              1195 non-null float64
Republican                              1195 non-null float64
Party                                   1195 non-null int32
dtypes: float64(13), int32(1), int64(3), object(2)
memory usage: 182.1+ KB
```

In [14]:

```
#data_population_mean=merged_data_set.groupby('Party')['Total Population'].mean()
republican_population_mean=merged_data_set[merged_data_set.Party == 0]['Total Populatio
n'].mean()
democratic_population_mean=merged_data_set[merged_data_set.Party == 1]['Total Populatio
n'].mean()
print("Mean population for Democratic counties "+str(democratic_population_mean))
print("Mean population for Republican counties "+str(republican_population_mean))
print("The mean population of Democratic counties is higher")
```

```
Mean population for Democratic counties 300998.3169230769
Mean population for Republican counties 53864.6724137931
The mean population of Democratic counties is higher
```

In [15]:

```
[statistic, pvalue] = st.ttest_ind(merged_data_set[merged_data_set.Party == 1]['Total P
opulation'], merged_data_set[merged_data_set.Party == 0]['Total Population'], equal_var
= False)
print("t-test statistic "+str(statistic))
print("pvalue "+str(pvalue))
print("Since pvalue "+str(pvalue)+" less than "+"𝛼=0.05")
print("We reject the null hypothesis")
```

```
t-test statistic 8.004638577960957
pvalue 2.0478717602973023e-14
Since pvalue 2.0478717602973023e-14 less than 𝛼=0.05
We reject the null hypothesis
```

In [16]:

```
#Median Household Income(mean)
republican_mhi_mean=merged_data_set[merged_data_set.Party == 0]['Median Household Income'].mean()
democratic_mhi_mean=merged_data_set[merged_data_set.Party == 1]['Median Household Income'].mean()
print("Mean Median Household Income for Democratic counties "+str(democratic_mhi_mean))
print("Mean Median Household Income for Republican counties "+str(republican_mhi_mean))
print("The Mean Median Household Income of Democratic counties is higher")
```

```
Mean Median Household Income for Democratic counties 53798.732307692306
Mean Median Household Income for Republican counties 48746.81954022989
The Mean Median Household Income of Democratic counties is higher
```

In [17]:

```
[statistic_mhi, pvalue_mhi] = st.ttest_ind(merged_data_set[merged_data_set.Party == 1]['Median Household Income'], merged_data_set[merged_data_set.Party == 0]['Median Household Income'], equal_var = False)
print("t-test statistic "+str(statistic_mhi))
print("pvalue "+str(pvalue_mhi))
print("Since pvalue "+str(pvalue_mhi)+" less than "+"𝛼=0.05")
print("We reject the null hypothesis")
```

```
t-test statistic 5.479141589767388
pvalue 7.149437363182572e-08
Since pvalue 7.149437363182572e-08 less than 𝛼=0.05
We reject the null hypothesis
```

In [18]:

```
#descriptive statistics
#data.groupby('Signed_In').describe().transpose()
subset = ['Percent Age 29 and Under','Percent Age 65 and Older','Percent Female','Perce
nt White, not Hispanic or Latino',"Percent Black, not Hispanic or Latino",'Percent Hisp
anic or Latino','Percent Foreign Born','Percent Less than High School Degree',"Percent
 Less than Bachelor's Degree",'Party']
merged_data_set[subset].groupby('Party').describe().transpose()
#merged_data_set.info()
```

Out[18]:

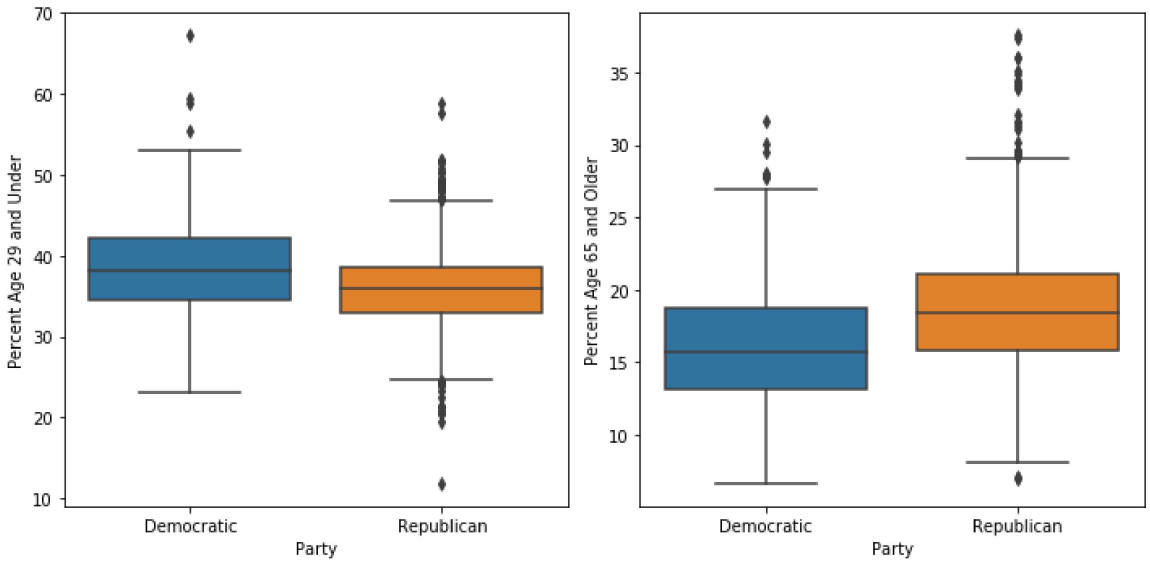
	Party	0	1
Percent Age 29 and Under	**count**	870.000000	325.000000
	mean	36.005719	38.726959
	std	5.181522	6.252786
	min	11.842105	23.156452
	25%	32.983652	34.488444
	50%	35.846532	38.074151
	75%	38.539787	42.161162
	max	58.749116	67.367823
Percent Age 65 and Older	**count**	870.000000	325.000000
	mean	18.828267	16.194826
	std	4.733155	4.282422
	min	6.954387	6.653188
	25%	15.784982	13.106233
	50%	18.377896	15.698087
	75%	21.112847	18.806426
	max	37.622759	31.642106
Percent Female	**count**	870.000000	325.000000
	mean	49.630898	50.385433
	std	2.429013	2.149359
	min	21.513413	34.245291
	25%	49.222905	49.854280
	50%	50.176792	50.653830
	75%	50.829770	51.492075
	max	55.885023	56.418468
Percent White, not Hispanic or Latino	**count**	870.000000	325.000000
	mean	82.656646	69.683766
	std	16.056122	24.981502
	min	18.758977	2.776702
	25%	75.016397	53.271579
	50%	89.434849	77.786090
...	**...**	...	...
Percent Hispanic or Latino	**std**	14.049576	19.575030
	min	0.000000	0.193349
	25%	1.704539	2.531017
	50%	3.427435	5.039747
	75%	10.709696	11.857116
	max	78.397012	95.479801

	Party	0	1
Percent Foreign Born	**count**	870.000000	325.000000
	mean	3.990096	7.986330
	std	4.507786	8.330740
	min	0.000000	0.179769
	25%	1.320101	2.470508
	50%	2.326317	5.105490
	75%	5.149429	10.144555
	max	37.058317	52.229868
Percent Less than High School Degree	**count**	870.000000	325.000000
	mean	14.009112	11.883760
	std	6.303126	6.505613
	min	2.134454	3.215803
	25%	9.662491	7.893714
	50%	12.572435	10.370080
	75%	17.447168	13.637059
	max	47.812773	49.673777
Percent Less than Bachelor's Degree	**count**	870.000000	325.000000
	mean	81.095427	71.968225
	std	6.815537	11.192404
	min	43.419470	26.335440
	25%	78.108424	65.711800
	50%	82.406700	72.736143
	75%	85.546272	79.903653
	max	97.014925	94.849957

72 rows × 2 columns

In [19]:

```
#plots for age variables
change_values = {1: 'Democratic', 0:'Republican'}
visualize_data = merged_data_set[['Percent Age 29 and Under','Percent Age 65 and Older'
,'Party']]
num_columns = len(visualize_data.columns) - 1
fig, axes = plt.subplots(1, num_columns, figsize = (10, 5))
for i in range(num_columns):
    sns.boxplot(x = visualize_data['Party'].map(change_values), y = visualize_data.colu
mns[i], data = visualize_data, orient = 'v', ax = axes[i])
plt.tight_layout()
```

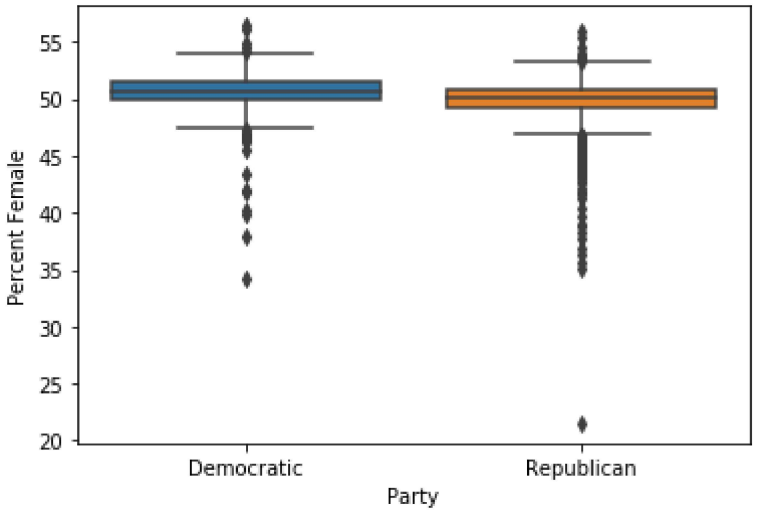


In [20]:

```
#plots for gender variable
visualize_data = merged_data_set[['Percent Female','Party']]
sns.boxplot(x = visualize_data['Party'].map(change_values), y = 'Percent Female', data
= visualize_data, orient = 'v')
```

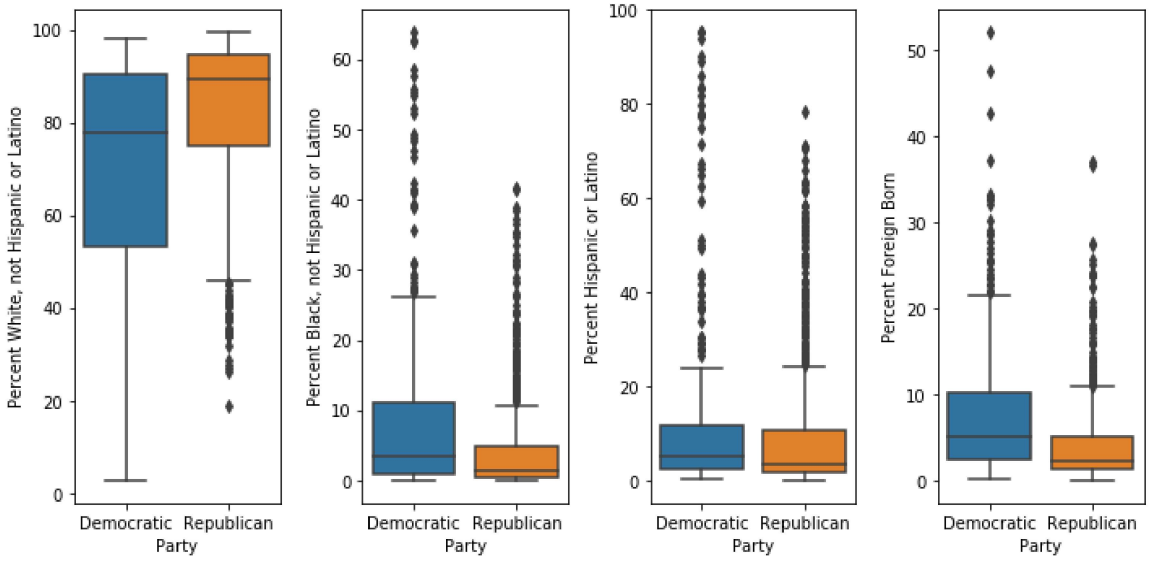
Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x19c8aaad8d0>
```



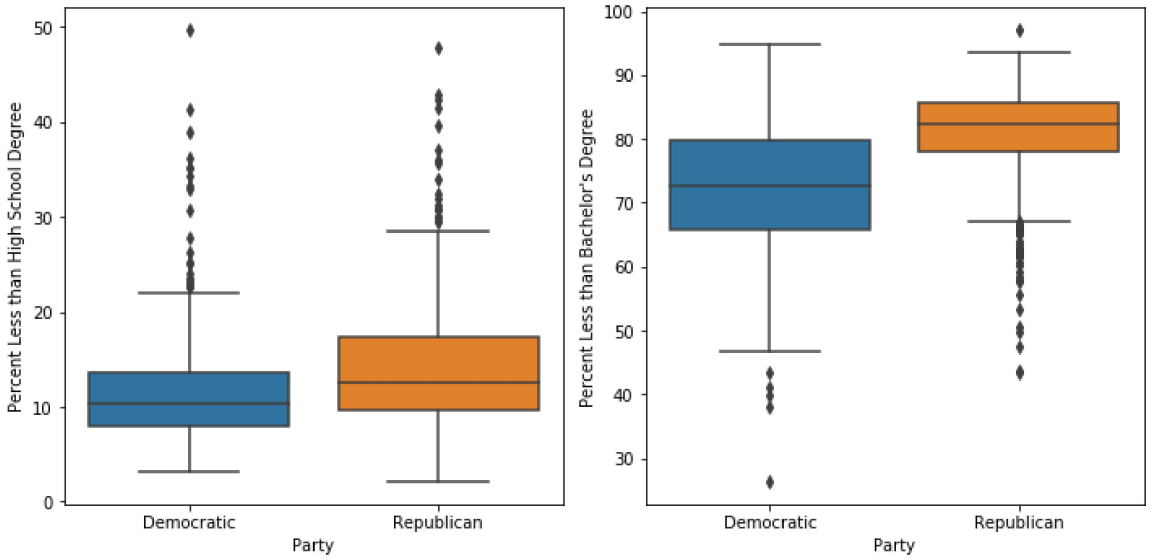
In [21]:

```
#plots for race and ethnicity variables
visualize_data = merged_data_set[['Percent White, not Hispanic or Latino',"Percent Blac
k, not Hispanic or Latino",'Percent Hispanic or Latino','Percent Foreign Born','Party'
]]
num_columns = len(visualize_data.columns) - 1
fig, axes = plt.subplots(1, num_columns, figsize = (10, 5))
for i in range(num_columns):
    sns.boxplot(x = visualize_data['Party'].map(change_values), y = visualize_data.colu
mns[i], data = visualize_data, orient = 'v', ax = axes[i])
plt.tight_layout()
```



In [22]:

```
#plots for Education
visualize_data = merged_data_set[['Percent Less than High School Degree',"Percent Less
 than Bachelor's Degree",'Party']]
num_columns = len(visualize_data.columns) - 1
fig, axes = plt.subplots(1, num_columns, figsize = (10, 5))
for i in range(num_columns):
    sns.boxplot(x = visualize_data['Party'].map(change_values), y = visualize_data.colu
mns[i], data = visualize_data, orient = 'v', ax = axes[i])
plt.tight_layout()
```



In [23]:

```
#Percent White, not Hispanic or Latino , Percent Less than Bachelor's Degree are the im
portant variables and Percent Less than High School Degree
```

In [24]:

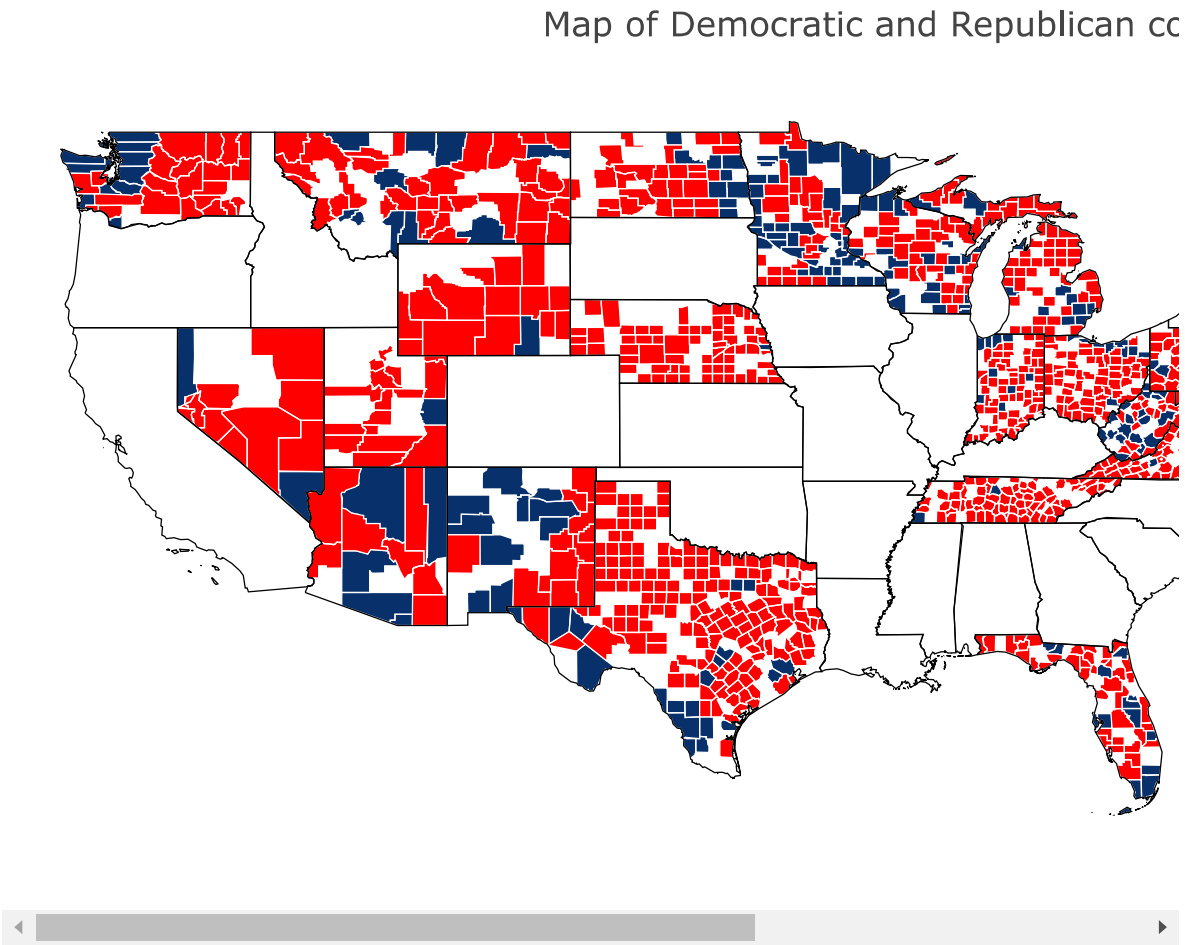
```
change_values = {1: 'Democratic', 0:'Republican'}
values = merged_data_set['Party'].map(change_values)
fips = merged_data_set['FIPS']
colorscale = ["#08306b","#ff0000"]
fig = ff.create_choropleth(fips=fips, values=values,colorscale=colorscale,county_outlin
e={'color': 'rgb(255,255,255)', 'width': 0.5},
    state_outline={'color': 'rgb(0,0,0)','width': 0.5},legend_title='Party Name',title=
'Map of Democratic and Republican counties')
fig.layout.template = None
fig.show()
```

```
C:\Users\USER\Anaconda3\lib\site-packages\pandas\core\frame.py:6692: Futur
eWarning:

Sorting because non-concatenation axis is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.
```



In [ ]: