Project 01: Exploratory Data Analysis

Libraries

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import linear_model
import matplotlib.pyplot as plot
import seaborn as sns
```

Data Import

In [2]:
```
election_train = pd.read_csv('election_train.csv')
election_train.head()
```

Out[2]:

	Year	State	County	Office	Party	Votes
0	2018	AZ	Apache County	US Senator	Democratic	16298
1	2018	AZ	Apache County	US Senator	Republican	7810
2	2018	AZ	Cochise County	US Senator	Democratic	17383
3	2018	AZ	Cochise County	US Senator	Republican	26929
4	2018	AZ	Coconino County	US Senator	Democratic	34240

In [3]:
```
demographics_train = pd.read_csv('demographics_train.csv')
demographics_train.head()
```

Out[3]:

	State	County	FIPS	Total Population	Citizen Voting-Age Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino	Percent Foreign Born	Percent Female	Percent Age 29 and Under	Percent Age 65 and Older	Median Household Income	U
0	Wisconsin	La Crosse	55063	117538	0	90.537528	1.214075	1.724549	2.976059	51.171536	43.241335	14.702479	51477	
1	Virginia	Alleghany	51005	15919	12705	91.940449	5.207614	1.432251	1.300333	51.077329	31.660280	23.902255	45538	
2	Indiana	Fountain	18045	16741	12750	95.705155	0.400215	2.359477	1.547100	49.770026	35.899887	18.941521	45924	
3	Ohio	Geauga	39055	94020	0	95.837056	1.256116	1.294405	2.578175	50.678579	36.281642	18.028079	74165	
4	Wisconsin	Jackson	55053	20566	15835	86.662453	1.983857	3.082758	1.376058	46.649810	36.292911	17.587280	49608	

Task 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 [4]:
```
election_train_tidy = pd.pivot_table(election_train,
                                     values='Votes',
                                     columns='Party',
                                     index=['Year',
                                            'State',
                                            'County',
                                            'Office']).reset_index()
election_train_tidy.head(5)
```

Out[4]:

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

Task 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.

County in **election_train** has word 'County' whereas **demographics_train** has only the name of county.

We will therefore remove word county in **election_train** to match the *County* names in **demographics_train**.

In [5]:
```
def remove_word_county(county_name):
    county_name = county_name.split(' ')
    if 'County' in county_name:
        county_name.remove('County')
    return ' '.join(county_name)
election_train_tidy['County'] = election_train_tidy['County'].apply(lambda x:
                                                                    remove_word_county(x))
election_train_tidy.head(5)
```

Out[5]:

Party	Year	State	County	Office	Democratic	Republican
0	2018	AZ	Apache	US Senator	16298.0	7810.0
1	2018	AZ	Cochise	US Senator	17383.0	26929.0
2	2018	AZ	Coconino	US Senator	34240.0	19249.0
3	2018	AZ	Gila	US Senator	7643.0	12180.0
4	2018	AZ	Graham	US Senator	3368.0	6870.0

State in **election_train** is abbreviated whereas **demographics_train** has full state name.

We will therefore replace the state abbreviation in **election_train** to its full state name to match the *State* in **demographics_train**.

In [6]:
```
us_state_abbrev = {
    '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'
}
change_values = {value : key for (key, value) in us_state_abbrev.items()}
election_train_tidy['State'] = election_train_tidy['State'].map(change_values)
election_train_tidy.sample(5)
```

Out[6]:

Party	Year	State	County	Office	Democratic	Republican
456	2018	Nebraska	York	US Senator	1281.0	3659.0
351	2018	Montana	Valley	US Senator	1545.0	2137.0
1149	2018	West Virginia	Berkeley	US Senator	14508.0	18111.0
706	2018	Tennessee	Lewis	US Senator	1177.0	2836.0
30	2018	Florida	Duval	US Senator	192381.0	185904.0

We will now merge **election_train_tidy** & **demographics_train** based on *State* as

In [7]:
```
election_train_tidy['County'] = election_train_tidy['County'].apply(lambda x: x.lower())
demographics_train['County'] = demographics_train['County'].apply(lambda x: x.lower())

data = pd.merge(election_train_tidy, demographics_train, on=['State', 'County'], how='inner')
data.head(5).transpose()
```

Out[7]:

	0	1	2	3	4
Year	2018	2018	2018	2018	2018
State	Arizona	Arizona	Arizona	Arizona	Arizona
County	apache	cochise	coconino	gila	graham
Office	US Senator	US Senator	US Senator	US Senator	US Senator
Democratic	16298	17383	34240	7643	3368
Republican	7810	26929	19249	12180	6870
FIPS	4001	4003	4005	4007	4009
Total Population	72346	128177	138064	53179	37529
Citizen Voting-Age Population	0	92915	104265	0	0
Percent White, not Hispanic or Latino	18.5719	56.2995	54.6196	63.2223	51.4615
Percent Black, not Hispanic or Latino	0.486551	3.71439	1.34286	0.55285	1.81193
Percent Hispanic or Latino	5.94781	34.4032	13.711	18.5487	32.0978
Percent Foreign Born	1.71951	11.4584	4.8253	4.2498	4.38594
Percent Female	50.5985	49.0696	50.5816	50.2962	46.3135
Percent Age 29 and Under	45.8546	37.9023	48.9461	32.2383	46.3935
Percent Age 65 and Older	13.3221	19.7563	10.8739	26.3976	12.3158
Median Household Income	32460	45383	51106	40593	47422
Percent Unemployed	15.8074	8.56711	8.2383	12.1299	14.4241
Percent Less than High School Degree	21.7583	13.4092	11.0854	15.73	14.5808
Percent Less than Bachelor's Degree	88.9411	76.8371	65.7914	82.2626	86.6759
Percent Rural	74.0611	36.3011	31.4661	41.062	46.4374

Task 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 [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1200 entries, 0 to 1199
Data columns (total 21 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   Year                                   1200 non-null   int64  
 1   State                                  1200 non-null   object 
 2   County                                 1200 non-null   object 
 3   Office                                 1200 non-null   object 
 4   Democratic                             1197 non-null   float64
 5   Republican                             1198 non-null   float64
 6   FIPS                                   1200 non-null   int64  
 7   Total Population                       1200 non-null   int64  
 8   Citizen Voting-Age Population          1200 non-null   int64  
 9   Percent White, not Hispanic or Latino  1200 non-null   float64
 10  Percent Black, not Hispanic or Latino  1200 non-null   float64
 11  Percent Hispanic or Latino             1200 non-null   float64
 12  Percent Foreign Born                   1200 non-null   float64
 13  Percent Female                         1200 non-null   float64
 14  Percent Age 29 and Under               1200 non-null   float64
 15  Percent Age 65 and Older               1200 non-null   float64
 16  Median Household Income                1200 non-null   int64  
 17  Percent Unemployed                     1200 non-null   float64
 18  Percent Less than High School Degree   1200 non-null   float64
 19  Percent Less than Bachelor's Degree    1200 non-null   float64
 20  Percent Rural                          1200 non-null   float64
dtypes: float64(13), int64(5), object(3)
memory usage: 206.2+ KB
```

Answer:

Number of variables: 21

Types of variables: object, int64, float64

Redundant Variables: *Year* and *Office* do not give any information about a county as it remains the same for all observations. So we can drop these variables.

In [9]:
```
data = data.drop(columns=['Year', 'Office'], axis=1)
```

Task 4

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

Answer:

In the merged dataset there are 5 observations with missing values for *Democratic* or *Republican* votes: we will be ignoring these observations

In [10]:
```
data = data.dropna(0)
```

Citizen Voting-Age Population has implicit missing values for 675 out of 1200 observations. We will be dropping this variable

In [11]:
```python
data = data.drop(columns=['Citizen Voting-Age Population'])
data.head(5)
```

Out[11]:

	State	County	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino	Percent Foreign Born	Percent Female	Percent Age 29 and Under	Percent Age 65 and Older	Hou
0	Arizona	apache	16298.0	7810.0	4001	72346	18.571863	0.486551	5.947806	1.719515	50.598513	45.854643	13.322091	
1	Arizona	cochise	17383.0	26929.0	4003	128177	56.299492	3.714395	34.403208	11.458374	49.069646	37.902276	19.756275	
2	Arizona	coconino	34240.0	19249.0	4005	138064	54.619597	1.342855	13.711033	4.825298	50.581614	48.946141	10.873943	
3	Arizona	gila	7643.0	12180.0	4007	53179	63.222325	0.552850	18.548675	4.249798	50.296170	32.238290	26.397638	
4	Arizona	graham	3368.0	6870.0	4009	37529	51.461536	1.811932	32.097844	4.385942	46.313518	46.393456	12.315809	

We have zero values in *Percent Black, not Hispanic or Latino* and *Percent Rural* but these contain useful information and are not treated as missing values.

Task 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 [12]:
```python
# Party variable (Democratic - 1; Republican - 0)
# Calculate Party
data['Party'] = np.where(data['Democratic']>data['Republican'], '1', '0')
data.head(5)
```

Out[12]:

	State	County	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino	Percent Foreign Born	Percent Female	Percent Age 29 and Under	Percent Age 65 and Older	Ho
0	Arizona	apache	16298.0	7810.0	4001	72346	18.571863	0.486551	5.947806	1.719515	50.598513	45.854643	13.322091	
1	Arizona	cochise	17383.0	26929.0	4003	128177	56.299492	3.714395	34.403208	11.458374	49.069646	37.902276	19.756275	
2	Arizona	coconino	34240.0	19249.0	4005	138064	54.619597	1.342855	13.711033	4.825298	50.581614	48.946141	10.873943	
3	Arizona	gila	7643.0	12180.0	4007	53179	63.222325	0.552850	18.548675	4.249798	50.296170	32.238290	26.397638	
4	Arizona	graham	3368.0	6870.0	4009	37529	51.461536	1.811932	32.097844	4.385942	46.313518	46.393456	12.315809	

Task 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 conclusion? What conclusion do you make from this result?

In [13]:
```
# Mean of Democratic counties
democratic_ = data.loc[data['Democratic']>data['Republican']]
democratic_mean = democratic_[['Median Household Income', 'Party']]
democratic_sample_mean = democratic_mean['Median Household Income'].mean()
print("Mean Median Household Income for Democratic Counties:", democratic_sample_mean)

#Mean of Republican counties
republican_ = data.loc[data['Democratic']<data['Republican']]
republican_mean = republican_[['Median Household Income', 'Party']]
republican_sample_mean = republican_mean['Median Household Income'].mean()
print("Mean Median Household Income for Republican Counties:", republican_sample_mean)
```

```
Mean Median Household Income for Democratic Counties: 53798.732307692306
Mean Median Household Income for Republican Counties: 48746.81954022989
```

Answer:

Mean Median Household Income for **Democratic** Counties: 53798.732307692306

Mean Median Household Income for **Republican** Counties: 48746.81954022989

Therefore clearly, mean median household income for **democratic** counties is *greater* than mean population for **republican** counties.

Hypthesis test:

$\bar{x}_1$ = 53798.732307692306 (Democratic Mean Median Household Income)

$\bar{x}_2$ = 48746.81954022989 (Republic Mean Median Household Income)

H_0: $\mu_1 = \mu_2$

H_α: $\mu_1 \neq \mu_2$

In [14]:
```
# hypothesis test
import scipy.stats as st
[statistic, pvalue] = st.ttest_ind(democratic_mean['Median Household Income'], republican_mean['Median Ho
usehold Income'], equal_var = False)
print("t-test statistic:", statistic)
print("p-value:", pvalue)
```

```
t-test statistic: 5.479141589767387
p-value: 7.149437363182598e-08
```

Answer:

t-test statistic: 5.479141589767387

p-value: 7.149437363182598e-08

Since pvalue < α, we reject H_0: Null hypothesis.

There is sufficient evidence to conclude that the mean median household income of democratic counties is different from republican counties.

Task 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 $\alpha = \mathbf{0.05}$ significance level. What is the result of the test? What conclusion do you make from this result?

```
In [15]: # Mean of Democratic counties
democratic_ = data.loc[data['Democratic']>data['Republican']]
democratic_tp = democratic_[['Total Population', 'Party']]
democratic_sample_mean = democratic_tp['Total Population'].mean()
print("Mean Population for Democratic Counties:", democratic_sample_mean)

#Mean of Republican counties
republican_ = data.loc[data['Democratic']<data['Republican']]
republican_tp = republican_[['Total Population', 'Party']]
republican_sample_mean = republican_tp['Total Population'].mean()
print("Mean Population for Republican Counties:", republican_sample_mean)
```

```
Mean Population for Democratic Counties: 300998.3169230769
Mean Population for Republican Counties: 53864.6724137931
```

Answer:

Mean population for **Democratic** Counties: 300998.3169230769

Mean population for **Republican** Counties: 53864.6724137931

Therefore clearly, mean population for **democratic** counties is *greater* than mean population for **republican** counties.

Hypthesis test:

$\bar{x}_1$ = 300998.3169230769 (Democratic Mean Population)

$\bar{x}_2$ = 53864.6724137931 (Republic Mean Population)

H_0: $\mu_1 = \mu_2$

H_α: $\mu_1 \neq \mu_2$

In [16]:
```
# hypothesis test
import scipy.stats as st
[statistic, pvalue] = st.ttest_ind(democratic_tp['Total Population'], republican_tp['Total Population'],
equal_var = False)
print("t-test statistic:", statistic)
print("pvalue:", pvalue)
```

```
t-test statistic: 8.004638577960957
pvalue: 2.0478717602973023e-14
```

Answer:

t-test statistic: 8.004638577960957

p-value: 2.0478717602973023e-14

Since pvalue < α, we reject H_0: Null hypothesis.

There is sufficient evidence to conclude that the mean population of democratic counties is different from republican counties.

Task 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?

In [17]:
```
#dataframe with age, gender, race and ethinitcy, and education
data_statistics = data.drop(columns= ['Democratic',
                                      'Republican',
                                      'FIPS',
                                      'Total Population',
                                      'Median Household Income',
                                      'Percent Unemployed',
                                      'Percent Rural' ])

#percentage of males
male_percent = []
for i in range(len(data_statistics['Percent Female'])):
    male_percent.append(100 - data_statistics.iloc[i,7])
data_statistics['Percent Male'] = male_percent

#Reording the columns
data_statistics = data_statistics[['Percent White, not Hispanic or Latino',
                                   'Percent Black, not Hispanic or Latino',
                                   'Percent Hispanic or Latino',
                                   'Percent Foreign Born',
                                   'Percent Female',
                                   'Percent Male',
                                   'Percent Age 29 and Under',
                                   'Percent Age 65 and Older',
                                   'Percent Less than High School Degree',
                                   'Percent Less than Bachelor\'s Degree',
                                   'Party']]
#information for each county
counties_democratic = data_statistics.loc[data['Democratic']>data['Republican']]
counties_republican = data_statistics.loc[data['Democratic']<data['Republican']]
```

In [18]:
```
#statistic description for democratic county
counties_democratic.describe().transpose().round(2)
```

Out[18]:

	count	mean	std	min	25%	50%	75%	max
Percent White, not Hispanic or Latino	325.0	69.68	24.98	2.78	53.27	77.79	90.30	98.06
Percent Black, not Hispanic or Latino	325.0	9.24	13.35	0.00	0.84	3.49	11.06	63.95
Percent Hispanic or Latino	325.0	12.59	19.58	0.19	2.53	5.04	11.86	95.48
Percent Foreign Born	325.0	7.99	8.33	0.18	2.47	5.11	10.14	52.23
Percent Female	325.0	50.39	2.15	34.25	49.85	50.65	51.49	56.42
Percent Male	325.0	61.27	6.25	32.63	57.84	61.93	65.51	76.84
Percent Age 29 and Under	325.0	38.73	6.25	23.16	34.49	38.07	42.16	67.37
Percent Age 65 and Older	325.0	16.19	4.28	6.65	13.11	15.70	18.81	31.64
Percent Less than High School Degree	325.0	11.88	6.51	3.22	7.89	10.37	13.64	49.67
Percent Less than Bachelor's Degree	325.0	71.97	11.19	26.34	65.71	72.74	79.90	94.85

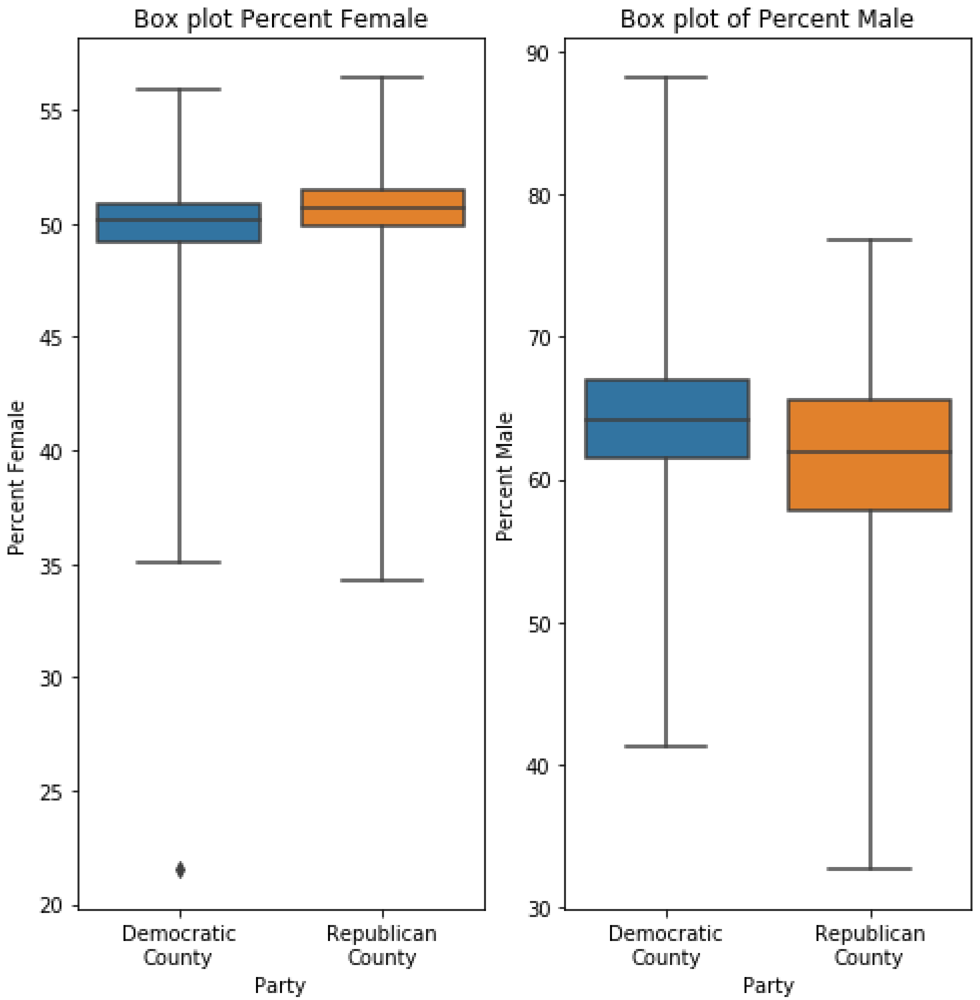
In [19]:
```
#statistic description for republican county
counties_republican.describe().transpose()
```

Out[19]:

	count	mean	std	min	25%	50%	75%	max
Percent White, not Hispanic or Latino	870.0	82.656646	16.056122	18.758977	75.016397	89.434849	94.466596	99.627329
Percent Black, not Hispanic or Latino	870.0	4.189241	6.721695	0.000000	0.460419	1.318311	4.753831	41.563041
Percent Hispanic or Latino	870.0	9.733094	14.049576	0.000000	1.704539	3.427435	10.709696	78.397012
Percent Foreign Born	870.0	3.990096	4.507786	0.000000	1.320101	2.326317	5.149429	37.058317
Percent Female	870.0	49.630898	2.429013	21.513413	49.222905	50.176792	50.829770	55.885023
Percent Male	870.0	63.994281	5.181522	41.250884	61.460213	64.153468	67.016348	88.157895
Percent Age 29 and Under	870.0	36.005719	5.181522	11.842105	32.983652	35.846532	38.539787	58.749116
Percent Age 65 and Older	870.0	18.828267	4.733155	6.954387	15.784982	18.377896	21.112847	37.622759
Percent Less than High School Degree	870.0	14.009112	6.303126	2.134454	9.662491	12.572435	17.447168	47.812773
Percent Less than Bachelor's Degree	870.0	81.095427	6.815537	43.419470	78.108424	82.406700	85.546272	97.014925

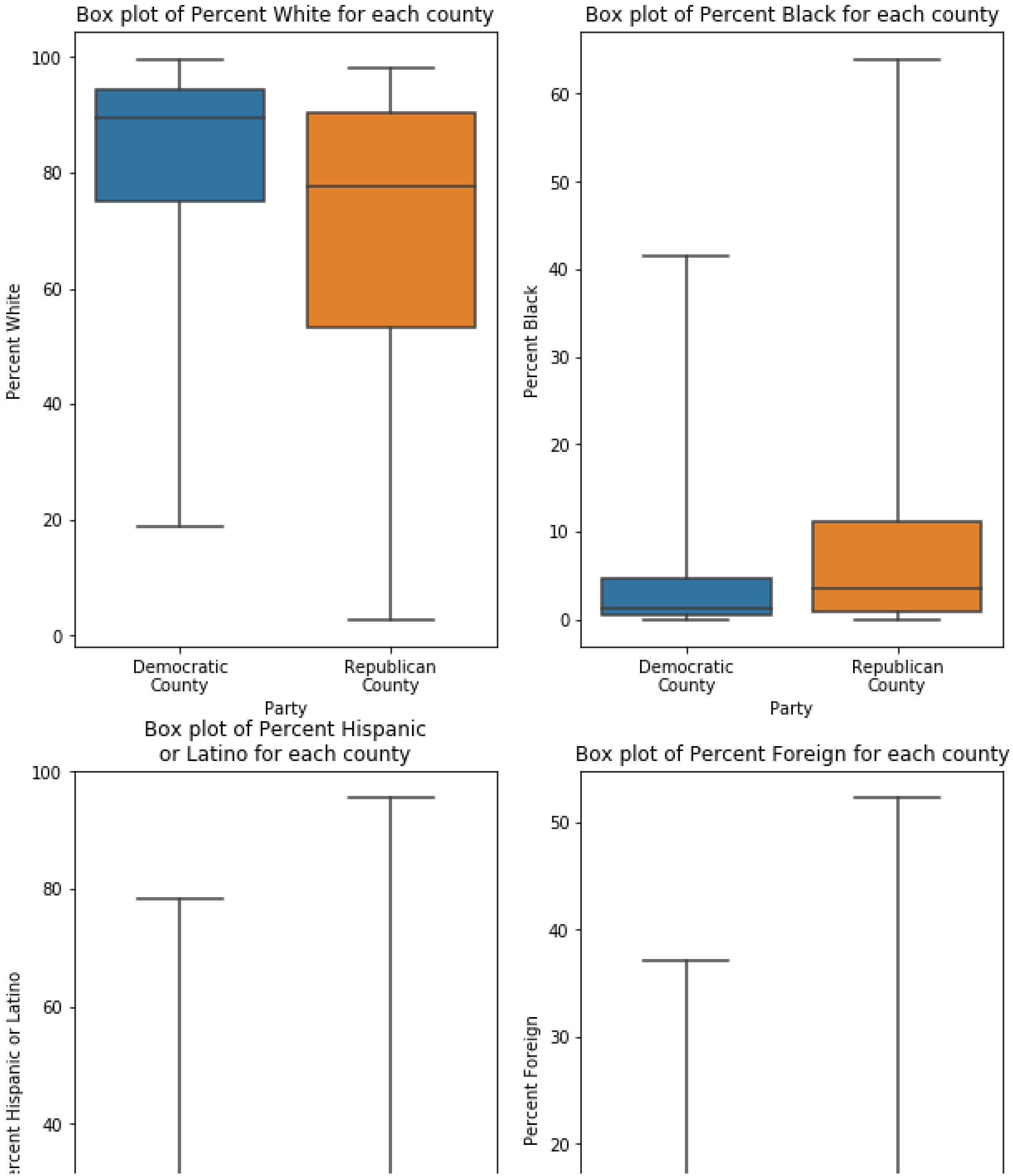
In [20]:
```
fig, axs = plot.subplots(1, 2)
fig.set_figheight(8)
fig.set_figwidth(8)
axis_ = sns.boxplot(x = 'Party', y = 'Percent Female', data = data_statistics, whis=10, ax=axs[0])
axis_.set(title = 'Box plot Percent Female', xlabel = 'Party', ylabel = 'Percent Female')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])

axis_ = sns.boxplot(x = 'Party', y = 'Percent Male', data = data_statistics, whis=10, ax=axs[1])
axis_.set(title = 'Box plot of Percent Male', xlabel = 'Party', ylabel = 'Percent Male')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])
```

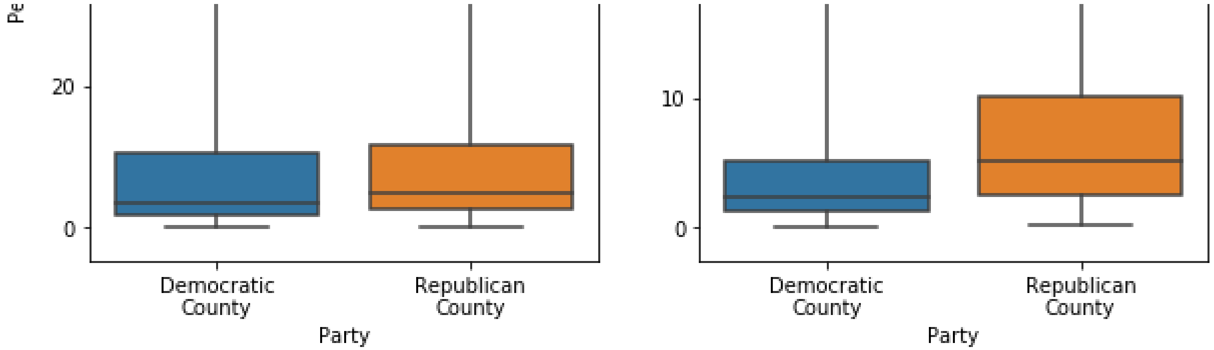
```
Out[20]: [Text(0, 0, 'Democratic\nCounty'), Text(0, 0, 'Republican\nCounty')]
```



In [21]:
```
fig, axs = plot.subplots(2, 2)
fig.set_figheight(15)
fig.set_figwidth(10)

axis_ = sns.boxplot(x = 'Party', y = 'Percent White, not Hispanic or Latino', data = data_statistics, whis=10, ax=axs[0][0])
axis_.set(title = 'Box plot of Percent White for each county', xlabel = 'Party', ylabel = 'Percent White')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])

axis_ = sns.boxplot(x = 'Party', y = 'Percent Black, not Hispanic or Latino', data = data_statistics, whis=10, ax=axs[0][1])
axis_.set(title = 'Box plot of Percent Black for each county', xlabel = 'Party', ylabel = 'Percent Black')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])

axis_ = sns.boxplot(x = 'Party', y = 'Percent Hispanic or Latino', data = data_statistics, whis=10, ax=axs[1][0])
axis_.set(title = 'Box plot of Percent Hispanic\nor Latino for each county', xlabel = 'Party', ylabel = 'Percent Hispanic or Latino')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])

axis_ = sns.boxplot(x = 'Party', y = 'Percent Foreign Born', data = data_statistics, whis=10, ax=axs[1][1])
axis_.set(title = 'Box plot of Percent Foreign for each county', xlabel = 'Party', ylabel = 'Percent Foreign')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])
```

Out[21]:
```
[Text(0, 0, 'Democratic\nCounty'), Text(0, 0, 'Republican\nCounty')]
```


Box plot of Percent White for each county
Percent White
100
80
60
40
20
0
Democratic County
Republican County
Party
Box plot of Percent Black for each county
Percent Black
60
50
40
30
20
10
0
Democratic County
Republican County
Party
Box plot of Percent Hispanic or Latino for each county
100
80
60
40
Box plot of Percent Foreign for each county
Percent Foreign
50
40
30
20

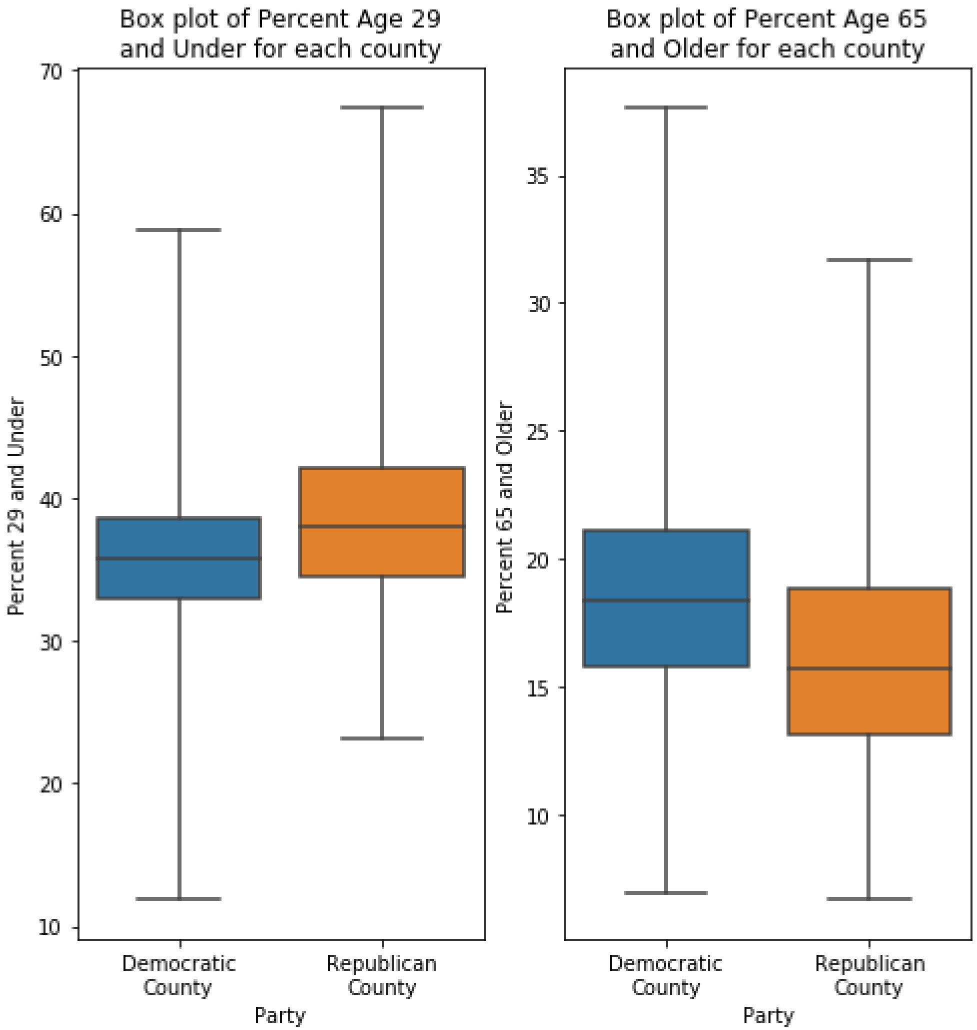


Pe
20
0
Democratic County
Republican County
Party
10
0
Democratic County
Republican County
Party

In [22]:

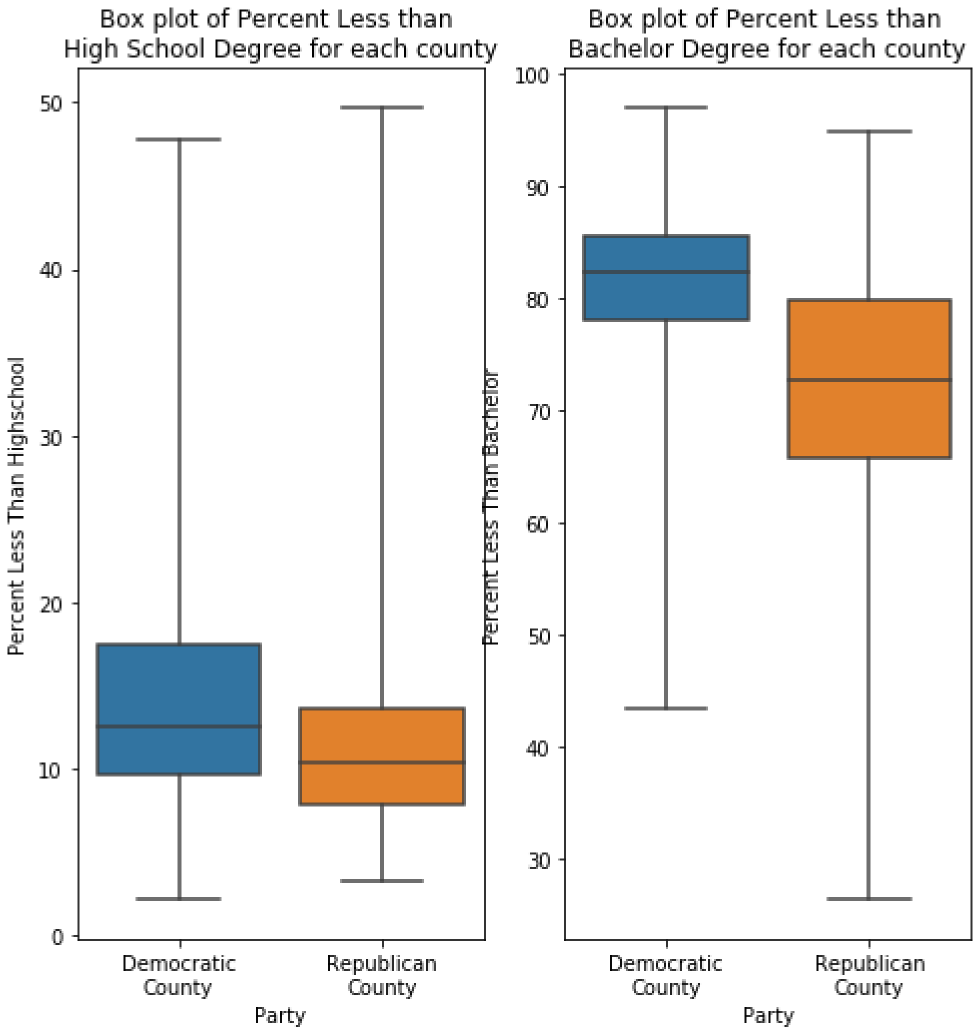
```
fig, axs = plot.subplots(1, 2)
fig.set_figheight(8)
fig.set_figwidth(8)

axis_ = sns.boxplot(x = 'Party', y = 'Percent Age 29 and Under', data = data_statistics, whis=10, ax=axs[0])
axis_.set(title = 'Box plot of Percent Age 29\nand Under for each county', xlabel = 'Party', ylabel = 'Percent 29 and Under')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])

axis_ = sns.boxplot(x = 'Party', y = 'Percent Age 65 and Older', data = data_statistics, whis=10, ax=axs[1])
axis_.set(title = 'Box plot of Percent Age 65\nand Older for each county', xlabel = 'Party', ylabel = 'Percent 65 and Older')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])
```

```
Out[22]: [Text(0, 0, 'Democratic\nCounty'), Text(0, 0, 'Republican\nCounty')]
```



In [23]:
```
fig, axs = plot.subplots(1, 2)
fig.set_figheight(8)
fig.set_figwidth(8)

axis_ = sns.boxplot(x = 'Party', y = 'Percent Less than High School Degree', data = data_statistics, whis
=10, ax=axs[0])
axis_.set(title = 'Box plot of Percent Less than \nHigh School Degree for each county', xlabel = 'Party',
ylabel = 'Percent Less Than Highschool')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])

axis_ = sns.boxplot(x = 'Party', y = 'Percent Less than Bachelor\'s Degree', data = data_statistics, whis
=10, ax=axs[1])
axis_.set(title = 'Box plot of Percent Less than \nBachelor Degree for each county', xlabel = 'Party', yl
abel = 'Percent Less Than Bachelor')
axis_.set_xticklabels(['Democratic\nCounty', 'Republican\nCounty'])
```

```
Out[23]: [Text(0, 0, 'Democratic\nCounty'), Text(0, 0, 'Republican\nCounty')]
```



Task 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:

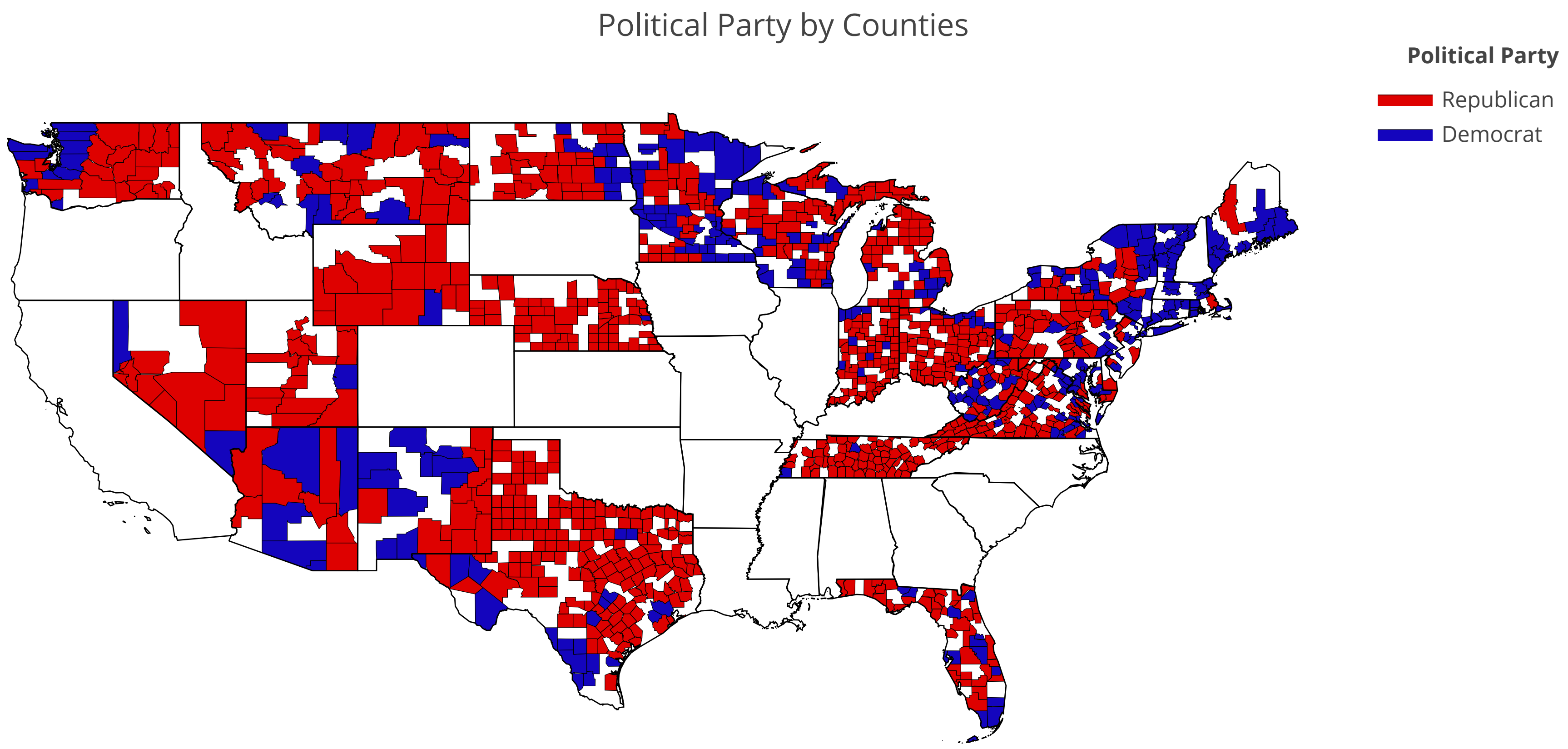
- The variables *Race and Ethnicity*, *Education and Age* are important to determine whether a county is marked Democratic or Republican.
- The Percent White make up about
  - 83% of Democrats
  - 70% of Republicans
- In Education
  - 72 % of Democrats have a degree less than a bachelor’s degree
  - 81 % of Republicans have a degree less than a bachelor’s degree.
- In age,
  - 39 % of Republicans are 29 and younger
  - 36 % of Democrats are age 29 and younger.

Task 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 [24]:

```python
#map of Democratic & Republic counties with FIPS codes based on the dataset
import plotly.figure_factory as ff
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)

fips = data['FIPS'].tolist()
party_values = data['Party'].map({'1': 'Democrat',
                  '0': 'Republican'}).tolist()
colorscale = ["#1405BD", "#DE0100"]
figure = ff.create_choropleth(fips=fips,
                              values=party_values,
                              colorscale=colorscale,
                              county_outline={'color': '#000000', 'width': 0.3},
                              state_outline={'color': '#000000', 'width': 0.7},
                              show_hover=False,
                              title='Political Party by Counties',
                              legend_title='Political Party')
figure.layout.template = None
iplot(figure, validate=False)
```


Political Party by Counties
Political Party
Republican
Democrat

In [ ]: