CS418 Project1 - Exploratory Data Analysis

Find the Project Description here.

This project is done as part of **CS418 - Introduction to DataScience** at UIC.

In [1]:
```
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as st
import plotly.figure_factory as ff
```

Load Dataset

In [2]:
```
election_train_raw = pd.read_csv('data/election_train.csv')
demographics_train = pd.read_csv('data/demographics_train.csv')
```

In [3]:
```
print(election_train_raw.shape)
election_train_raw.head()
```

(2405, 6)

Out[3]:

	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 [4]:
```
print(demographics_train.shape)
demographics_train.head()
```

(1216, 17)

Out[4]:

	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	Perc Fem
0	Wisconsin	La Crosse	55063	117538	0	90.537528	1.214075	1.724549	2.976059	51.171
1	Virginia	Alleghany	51005	15919	12705	91.940449	5.207614	1.432251	1.300333	51.077
2	Indiana	Fountain	18045	16741	12750	95.705155	0.400215	2.359477	1.547100	49.770

	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	Perc Fem
3	Ohio	Geauga	39055	94020	0	95.837056	1.256116	1.294405	2.578175	50.678
4	Wisconsin	Jackson	55053	20566	15835	86.662453	1.983857	3.082758	1.376058	46.649

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 [5]:

```
election_train = election_train_raw.pivot(index=['Year','State','County','Office'], col
print(election_train.shape)
election_train
```

```
(1205, 6)
```

Out[5]:

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. 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 [6]:

```
state_abbr = {'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',
 'MP': 'Northern Mariana Islands',
 'OH': 'Ohio',
 'OK': 'Oklahoma',
 'OR': 'Oregon',
 'PW': 'Palau',
 'PA': 'Pennsylvania',
 'RI': 'Rhode Island',
 'PR': 'Puerto Rico',
 'SC': 'South Carolina',
 'SD': 'South Dakota',
 'TN': 'Tennessee',
 'TX': 'Texas',
 'UT': 'Utah',
 'VT': 'Vermont',
 'VA': 'Virginia',
 'WA': 'Washington',
 'DC': 'Washington, DC',
 'WV': 'West Virginia',
 'WI': 'Wisconsin',
 'WY': 'Wyoming',
 'VI': 'Virgin Islands'}
```

In [7]:

```
election_train['State'] = election_train['State'].map(state_abbr)
```

In [8]:

```
def standardize_county_name(county):
    county = county.replace('County', '').strip()
    return county.lower()

election_train['County'] = election_train['County'].apply(standardize_county_name)
demographics_train['County'] = demographics_train['County'].apply(standardize_county_na
```

In [9]:
```
election_dataset = election_train.merge(demographics_train, how='inner', on=['State', '
election_dataset.shape
```

Out[9]: (1200, 21)

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 [10]:
```
print('Shape: ', election_dataset.shape)
election_dataset.info()
```

```
Shape:  (1200, 21)
<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
```

In [11]:
```
election_dataset.describe()
```

Out[11]:

	Year	Democratic	Republican	FIPS	Total Population	Citizen Voting-Age Population	Percent White, not Hispanic or Latino	I
count	1200.0	1197.000000	1198.000000	1200.000000	1.200000e+03	1.200000e+03	1200.000000	1.
mean	2018.0	25096.309106	20436.841402	38315.355000	1.208766e+05	3.226592e+04	79.099685	
std	0.0	72593.640184	45218.050721	13001.996705	3.183773e+05	1.247969e+05	19.782542	
min	2018.0	6.000000	46.000000	4001.000000	7.600000e+01	0.000000e+00	2.776702	
25%	2018.0	1427.000000	2667.500000	27146.500000	1.208350e+04	0.000000e+00	70.168347	

	Year	Democratic	Republican	FIPS	Total Population	Citizen Voting-Age Population	Percent White, not Hispanic or Latino
50%	2018.0	4213.000000	6691.000000	39140.000000	3.264300e+04	0.000000e+00	86.801005
75%	2018.0	14206.000000	16740.500000	48416.000000	8.582300e+04	1.893250e+04	93.876656
max	2018.0	881802.000000	672505.000000	56043.000000	4.434257e+06	2.723565e+06	99.627329

Merged Dataset Summary:

- **Number of variables**: 21
- **Types of data:** float object: 13, int object: 5, string object: 3
- **Irrelevant or redundant variables?** Office and Year are the irrelevant or redundant variables

- **Dealing with Irrelevant/Redundant Variables**: These variables can be removed from the dataframe since they are of no signifant importance for further analysis

In [12]:
```
election_dataset=election_dataset.drop(columns=['Office','Year'])
```

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]:
```
election_dataset[pd.isna(election_dataset['Democratic'])]
```

Out[13]:

	State	County	Democratic	Republican	FIPS	Total Population	Citizen Voting-Age Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino
425	Nebraska	lancaster	NaN	49449.0	31109	301707	0	82.659667	3.783472
714	Tennessee	meigs	NaN	2694.0	47121	11804	0	94.713656	1.330058
865	Texas	menard	NaN	632.0	48327	2163	0	56.310680	1.248266

In [14]:
```
election_dataset[pd.isna(election_dataset['Republican'])]
```

Out[14]:

	State	County	Democratic	Republican	FIPS	Total Population	Citizen Voting-Age Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino
750	Texas	bee	2811.0	NaN	48025	32706	0	32.660674	7.989360

	State	County	Democratic	Republican	FIPS	Total Population	Citizen Voting-Age Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino
1114	Wisconsin	lafayette	3592.0	NaN	55065	16793	0	94.771631	0.339427

In [15]:

```
election_dataset[election_dataset['Citizen Voting-Age Population']==0]
```

Out[15]:

	State	County	Democratic	Republican	FIPS	Total Population	Citizen Voting-Age Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino
0	Arizona	apache	16298.0	7810.0	4001	72346	0	18.571863	0.486551
3	Arizona	gila	7643.0	12180.0	4007	53179	0	63.222325	0.552850
4	Arizona	graham	3368.0	6870.0	4009	37529	0	51.461536	1.811932
7	Arizona	mohave	19214.0	50209.0	4015	203629	0	78.252606	0.951731
9	Arizona	pima	221242.0	160550.0	4019	1003338	0	53.271579	3.199719
...	...	...	...	...	...	...	...	...	...
1188	Wyoming	converse	834.0	3959.0	56009	14223	0	88.849047	0.007031
1190	Wyoming	goshen	1020.0	3658.0	56015	13546	0	86.409272	0.147645
1192	Wyoming	lincoln	1152.0	5846.0	56023	18543	0	92.600982	0.210322
1196	Wyoming	sublette	668.0	2653.0	56035	10032	0	91.646730	0.000000
1199	Wyoming	washakie	588.0	2423.0	56043	8351	0	82.397318	0.790325

680 rows × 19 columns

Missing values information : There 3 rows with missing values for 'Democratic' votes and 2 rows with missing values for 'Republican' votes. Additionally, 680 rows are present with 'Citizen Voting-Age Population' as 0. These can be considered as a missing value.

Dealing with Missing values: The total 5 rows with missing votes for one party should be deleted as it can influence the result but both the party votes are unknown. Since there are 680 rows missing in the 'Citizen Voting-Age Population', it can be best resolved by deleting the feature from the data-frame

In [16]:

```
election_dataset = election_dataset.dropna().drop(columns=['Citizen Voting-Age Populati
election_dataset['Democratic'] = election_dataset['Democratic'].astype(int)
election_dataset['Republican'] = election_dataset['Republican'].astype(int)
election_dataset.sample(3)
```

Out[16]:

	State	County	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino
172	Maine	androscoggin	22150	18931	23001	107376	91.319289	1.551557	1.733162
1149	West Virginia	doddridge	746	1352	54017	8363	95.731197	1.171828	0.310893
552	Ohio	carroll	3788	6503	39019	28108	96.549025	0.761349	1.170485

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 [17]:

```
election_dataset['Party'] = (election_dataset['Democratic'] > election_dataset['Republi
election_dataset.sample(3)
```

Out[17]:

	State	County	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino	F F
1033	Virginia	stafford	28536	26368	51179	139548	64.701035	16.566343	11.188982	8.
362	North Dakota	eddy	555	675	38027	2370	90.168776	0.000000	1.814346	1.
850	Texas	leon	855	5711	48289	16923	76.629439	7.209124	13.975064	6.

In [18]:

```
# Save the cleaned data as a CSV file for potential future use
election_dataset.to_csv('data/clean_data.csv', index=False)
```

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 [19]:

```
mean_income_democratic = election_dataset[election_dataset['Party'] == 1]['Median House
mean_income_republican = election_dataset[election_dataset['Party'] == 0]['Median House
print("Mean 'Median Household income' of Democratic County's:", mean_income_democratic)
print("Mean 'Median Household income' of Republican County's:", mean_income_republican)
```

```
Mean 'Median Household income' of Democratic County's: 53798.732307692306
Mean 'Median Household income' of Republican County's: 48746.81954022989
```

Mean 'Median Household Income' of Democratic Counties are higher than Republican Counties.

Null Hypotheses: Median Household Income of Democratic counties is equal to Republican counties.($\mu d == \mu r$)
Alternative Hypotheses: Median Household Income of Democratic counties are higher than Republican counties. ($\mu d > \mu r$)

We do a t-test on the data since population standard deviation is unknown.
We do a right tailed t-test since the alternative hypothesis is μd>μr

In [20]:
```
(t_test_statistic,p_value) = st.ttest_ind(election_dataset[election_dataset['Party'] ==
                                          election_dataset[election_dataset['Party'] ==
                                          equal_var=False)
```

In [21]:
```
#Since the function return two sided test result, convert it into right tailed test.
p_value=p_value/2
print('T-Test Statistic: ', t_test_statistic)
print('p value: ', p_value)
```

```
T-Test Statistic:  5.479141589767387
p value:  3.574718681591299e-08
```

*The p value for the Null hypothesis is 3.57*10^-8 which is way lesser than the significance level 0.05. Hence, we reject the null hypothesis and there is sufficient evidence to conclude that Median Household Income of Democratic counties may be higher than that of the republican ones*

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 [22]:
```
mean_population_democratic = election_dataset[election_dataset['Party'] == 1]['Total Po
mean_population_republican = election_dataset[election_dataset['Party'] == 0]['Total Po
print("Mean 'Total Population' of Democratic County's:", mean_population_democratic)
print("Mean 'Total Population' of Republican County's:", mean_population_republican)
```

```
Mean 'Total Population' of Democratic County's: 300998.3169230769
Mean 'Total Population' of Republican County's: 53864.6724137931
```

Mean 'Total Population' of Democratic Counties are higher than Republican Counties.

Null Hypotheses: Mean population of Democratic counties is equal to Republican counties. ($\mu d == \mu r$)
Alternative Hypotheses: Mean population of Democratic counties are higher than Republican counties. ($\mu d > \mu r$)

We do a t-test on the data since population standard deviation is unknown.
We do a right tailed t-test since the alternative hypothesis is μd>μr

In [23]:
```
(t_test_statistic,p_value) = st.ttest_ind(election_dataset[election_dataset['Party'] ==
```

```
                                          election_dataset[election_dataset['Party'] ==
                                          equal_var=False)
```

In [24]:

```
#Since the function return two sided test result, convert it into right tailed test.
p_value=p_value/2
print('T-Test Statistic: ', t_test_statistic)
print('p value: ', p_value)
```

```
T-Test Statistic:  8.004638577960957
p value:  1.0239358801486512e-14
```

*The p value for the Null hypothesis is 1.024*10^-14 which is way lesser than the significance level 0.05.

Hence, we reject the null hypothesis and there is sufficient evidence to conclude that mean population of Democratic counties **maybe** higher than that of the republican ones*

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 [25]:

```
democratic_counties = election_dataset[election_dataset['Party'] == 1]
republican_counties = election_dataset[election_dataset['Party'] == 0]
```

Gender:

In [26]:

```
election_dataset[['Percent Female','Party']].groupby(by=['Party']).describe().T
```

Out[26]:

	Party	0	1
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

Age:

- Columns 'Percent Age 29 and Under','Percent Age 65 and Older' are available in the dataset.
- Column 'Percent Age between 30 and 64' can be computed, which can be very helpful in visualizing the distribution.

In [27]:

```
age_columns = ['Percent Age 29 and Under', 'Percent Age from 30 to 64', 'Percent Age 65
election_dataset[age_columns[1]] = 100 - (election_dataset[age_columns[0]] + election_d
```

In [28]:

```
election_dataset.groupby(by=['Party'])[age_columns].describe().T
```

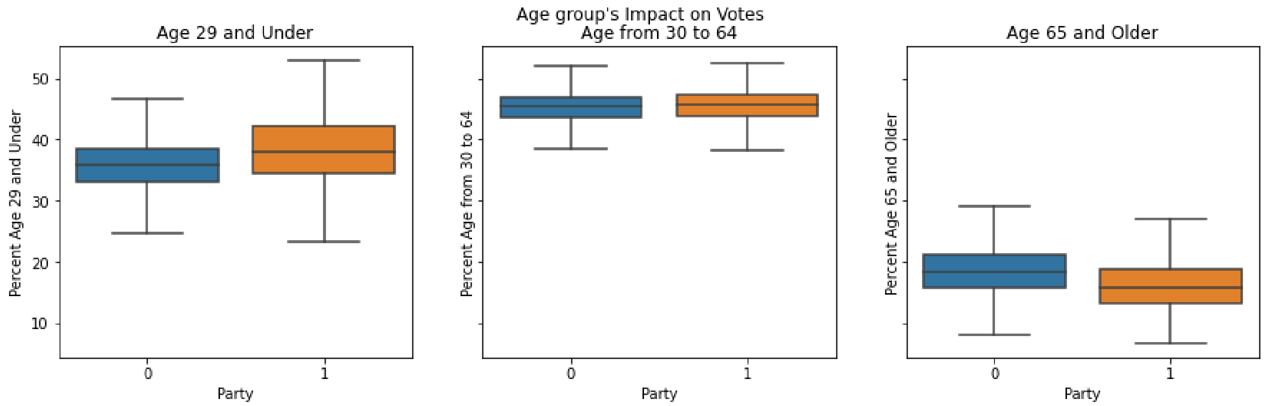
Out[28]:

	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 from 30 to 64	**count**	870.000000	325.000000
	mean	45.166015	45.078214
	std	2.910264	3.907598
	min	27.421759	18.433769
	25%	43.522522	43.741937
	50%	45.553295	45.817819
	75%	46.975771	47.448269
	max	63.157895	57.478906
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

In [29]:

```
fig, axes = plt.subplots(1, 3, sharex=True, sharey=True, figsize=(15, 4))
fig.suptitle('Age group\'s Impact on Votes')

for index, y_col in enumerate(age_columns):
    sns.boxplot(ax=axes[index], x="Party", y=y_col, data=election_dataset,showfliers=Fa
    axes[index].set_title(y_col.replace('Percent ',''))
```



Race and Ethinicity:

In [30]:
```
ethnicity_columns = ['Percent White, not Hispanic or Latino', 'Percent Black, not Hispa
election_dataset.groupby(by=['Party'])[ethnicity_columns].describe().T
```

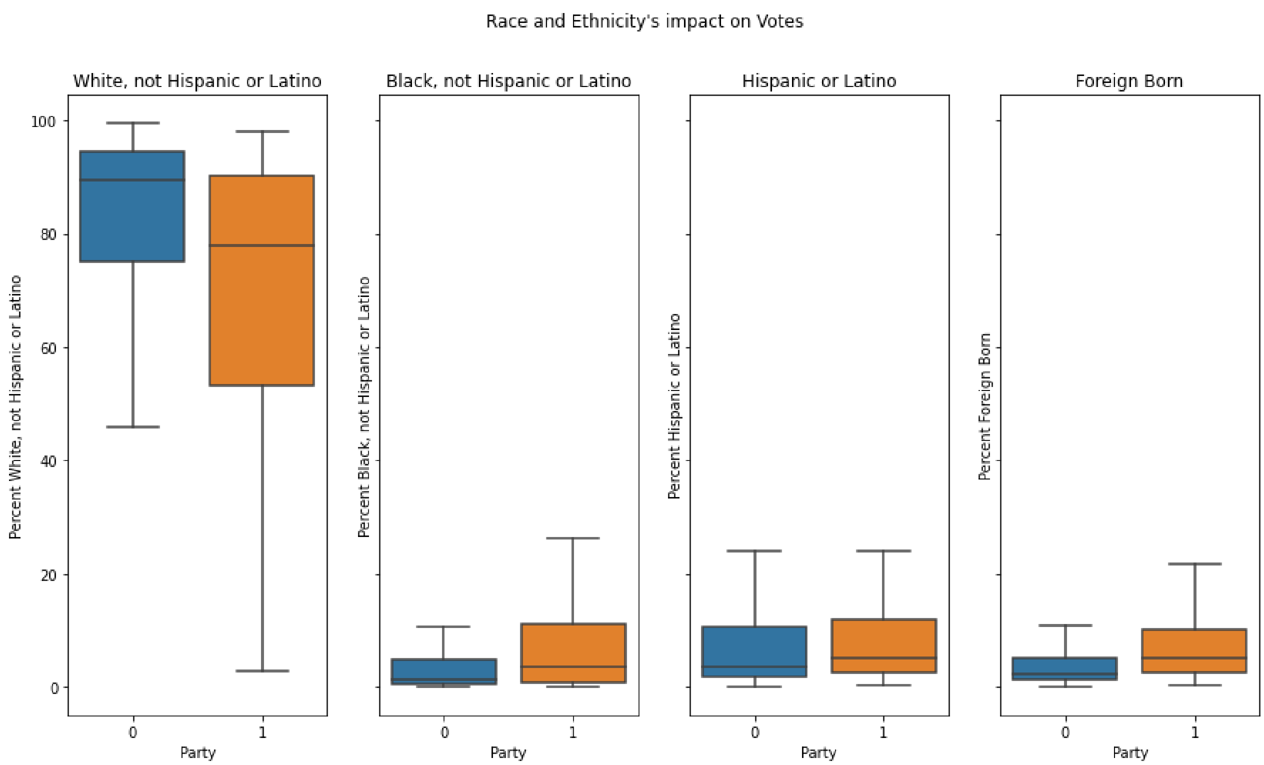
Out[30]:

	Party	0	1
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
	75%	94.466596	90.300749
	max	99.627329	98.063495
Percent Black, not Hispanic or Latino	**count**	870.000000	325.000000
	mean	4.189241	9.242649
	std	6.721695	13.351340
	min	0.000000	0.000000
	25%	0.460419	0.839103
	50%	1.318311	3.485992
	75%	4.753831	11.058843
	max	41.563041	63.953279
Percent Hispanic or Latino	**count**	870.000000	325.000000
	mean	9.733094	12.587391
	std	14.049576	19.575030
	min	0.000000	0.193349
	25%	1.704539	2.531017
	50%	3.427435	5.039747

	Party	0	1
	75%	10.709696	11.857116
	max	78.397012	95.479801
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

In [31]:
```
fig, axes = plt.subplots(1, 4, sharex=True, sharey=True, figsize=(15, 8))
fig.suptitle('Race and Ethnicity\'s impact on Votes')

for index, y_col in enumerate(ethnicity_columns):
    sns.boxplot(ax=axes[index], x="Party", y=y_col, data=election_dataset,showfliers=Fa
    axes[index].set_title(y_col.replace('Percent ',''))
```



Education:

- Columns 'Percent Less than High School Degree', 'Percent Less than Bachelor's Degree' are related to Education in the dataset.
- 'Percent Higher than Bachelor's Degree' can be computed using these two values.

- New column is added to visualize and understand better how education level affects the voting pattern

In [32]:
```
education_columns = ['Percent Less than High School Degree', 'Percent Less than Bachelo
election_dataset[education_columns[-1]] = 100 - election_dataset[education_columns[1]]
```

In [33]:
```
election_dataset.groupby(by=['Party'])[education_columns].describe().T
```

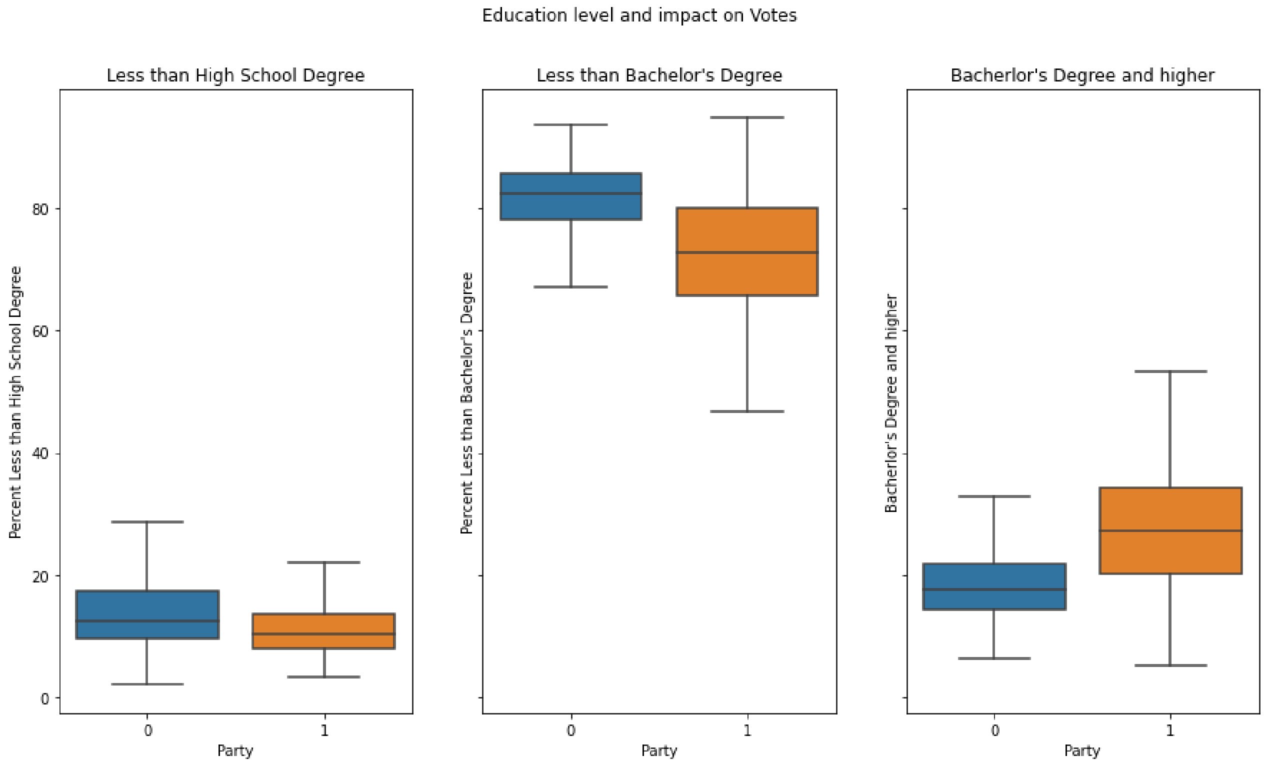
Out[33]:

	Party	0	1
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
Bacherlor's Degree and higher	**count**	870.000000	325.000000
	mean	18.904573	28.031775
	std	6.815537	11.192404
	min	2.985075	5.150043
	25%	14.453728	20.096347
	50%	17.593300	27.263857
	75%	21.891576	34.288200
	max	56.580530	73.664560

In [34]:
```
fig, axes = plt.subplots(1, 3, sharex=True, sharey=True, figsize=(15, 8))
fig.suptitle('Education level and impact on Votes')

for index, y_col in enumerate(education_columns):
```

```
    sns.boxplot(ax=axes[index], x="Party", y=y_col, data=election_dataset,showfliers=Fa
    axes[index].set_title(y_col.replace('Percent ',''))
```



Gender and Voting Patters:

- We have 'Percent Female' column in the dataset.
- 'Non-Females' (Including Male and Transgender voters) can be computed from the female voters data for visualization purposes.

In [35]:

```
election_dataset['Percent Non-Females'] = 100 - election_dataset['Percent Female']
gender_columns = ['Percent Female', 'Percent Non-Females']
```

In [36]:

```
election_dataset.groupby(by=['Party'])[gender_columns].describe().T
```

Out[36]:

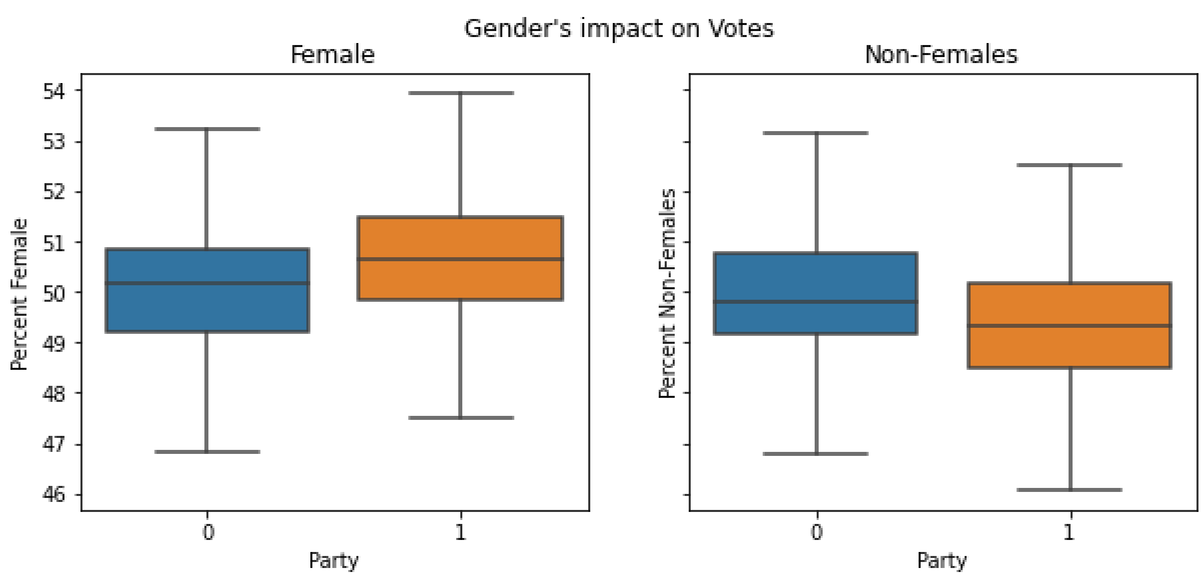
	Party	0	1
Percent Female	**count**	870.000000	325.000000
	mean	49.630898	50.385433
	std	2.429013	2.149359
	min	21.513413	34.245291
	25%	49.222905	49.854280
	50%	50.176792	50.653830
	75%	50.829770	51.492075
	max	55.885023	56.418468
Percent Non-Females	**count**	870.000000	325.000000

Party	0	1
mean	50.369102	49.614567
std	2.429013	2.149359
min	44.114977	43.581532
25%	49.170230	48.507925
50%	49.823208	49.346170
75%	50.777095	50.145720
max	78.486587	65.754709

In [37]:

```
fig, axes = plt.subplots(1, 2, sharex=True, sharey=True, figsize=(10, 4))
fig.suptitle('Gender\'s impact on Votes')

for index, y_col in enumerate(gender_columns):
    sns.boxplot(ax=axes[index], x="Party", y=y_col, data=election_dataset,showfliers=Fa
    axes[index].set_title(y_col.replace('Percent ',''))
```



Removing Redundant Computed Values:

In [38]:

```
election_dataset.drop(columns=[age_columns[1]])
election_dataset.drop(columns=[education_columns[-1]])
election_dataset.drop(columns=[gender_columns[-1]])
```

Out[38]:

	State	County	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino
0	Arizona	apache	16298	7810	4001	72346	18.571863	0.486551	5.947806
1	Arizona	cochise	17383	26929	4003	128177	56.299492	3.714395	34.403208

	State	County	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino
2	Arizona	coconino	34240	19249	4005	138064	54.619597	1.342855	13.711033
3	Arizona	gila	7643	12180	4007	53179	63.222325	0.552850	18.548675
4	Arizona	graham	3368	6870	4009	37529	51.461536	1.811932	32.097844
...	...	...	...	...	...	...	...	...	..
1195	Wyoming	platte	801	2850	56031	8740	89.359268	0.057208	7.814645
1196	Wyoming	sublette	668	2653	56035	10032	91.646730	0.000000	7.814992
1197	Wyoming	sweetwater	3943	8577	56037	44812	79.815674	0.865840	15.859591
1198	Wyoming	uinta	1371	4713	56041	20893	87.718375	0.186665	8.959939
1199	Wyoming	washakie	588	2423	56043	8351	82.397318	0.790325	13.962400

1195 rows × 21 columns

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.

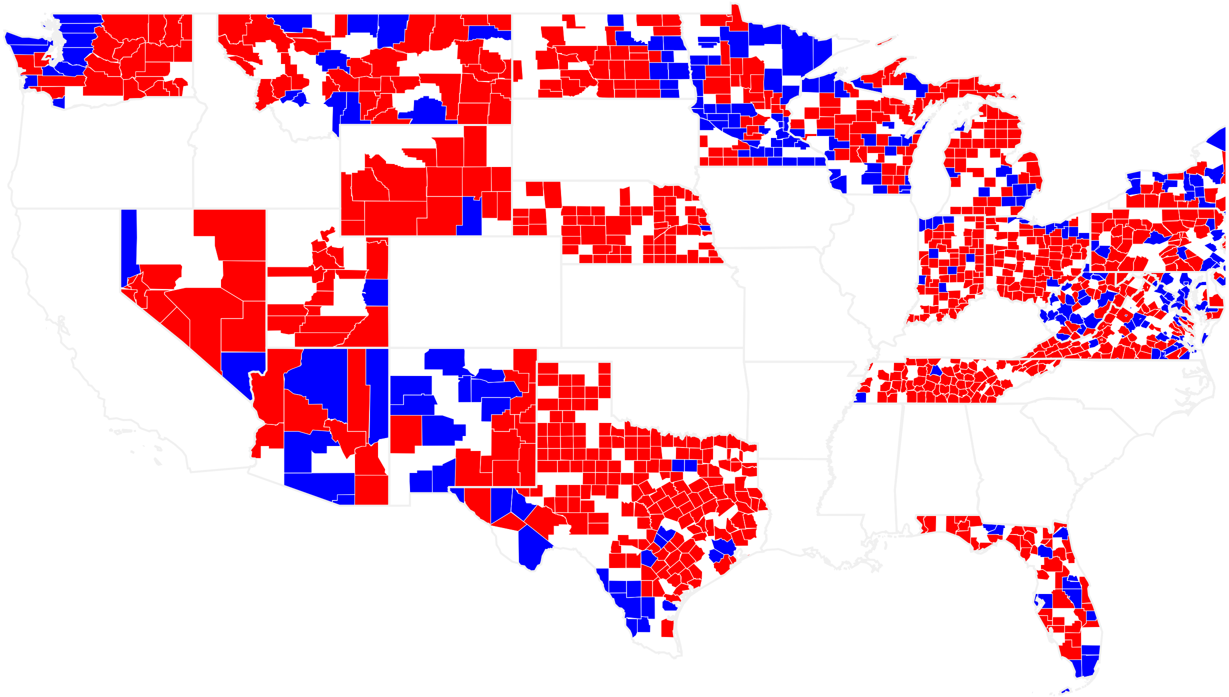
According to the results from tasks 6-8, the 'Total Population' of a county and Education Level('Bachelor degree or higher' 'less than bachelor degree' and 'less than high school degree') are more important to determine whether a county is labeled as Democratic or Republican.

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

In [39]:
```
# These libraries need to be installed
# pip install geopandas==0.3.0
# pip install pyshp==1.2.10
# pip install shapely==1.6.3
```

In [40]:
```
fips = election_dataset['FIPS'].to_list()
values = election_dataset['Party']

fig = ff.create_choropleth(fips=fips, values=values,colorscale=['#ff0000','#0000ff'],le
    county_outline={'color': 'rgb(255,255,255)', 'width': 0.2},)
fig.layout.template = None
fig.show()
```



In [ ]: