Nathan He, Yushen Li, Samujjwaal Dey 11/3/2019
CS418 - Fall 2019

Project 1 Report

1 & 2.

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

```
#import dataset
original_election_data = pd.read_csv("election_train.csv")
demographic_data = pd.read_csv("demographics_train.csv")
original_election_data.count()
#demographic_data
```

	Year	State	County	Office	Party	Votes
0	2018	AZ	Apache County	US Senator	Democratic	16298.0
1	2018	AZ	Apache County	US Senator	Republican	7810.0
2	2018	AZ	Cochise County	US Senator	Democratic	17383.0
2409	2018	WY	Washakie County	US Senator	Republican	2423.0

2410 rows × 6 columns

For this project, we used two standard library, pandas and numpy for some additional computation. First, we import the database files as usual and check if we have all the data. The output is expected. We read 2410 lines of data from the election_train.csv file.

```
#Task 1 : Reshape from long format to wide format
election_data = pd.pivot_table(original_election_data, index=['Year','State','County', 'Office'], columns=['Party'], values=['Votes'], aggfunc=np.sum).reset_index()
election_data.columns = election_data.columns.droplevel(1)
election_data.columns = ["Year", "State", "County", "Office", "Democratic Votes", "Republican Votes"]
election_data.count()
```

```
Year                1205
State               1205
County              1205
Office              1205
Democratic Votes    1205
Republican Votes    1205
dtype: int64
```

We notice that each county has basically the same data except for the party. Therefore, reshape action is performed to convert the dataset from long format to wide format. The reshaped result matches the expectation since 2410/2 = 1205

Then we need to merge the databases into one dataframe. However, we notice that we need to have the same key for merging. And for the 'State', one has AZ the other one has Arizona; for the 'County', one has Cook, the other one has Cook County. As a result, we need to perform some pre-processing in the dataframe.

```
#election_data['County'].replace({'County', ''}, inplace=True, regex=False )
election_data['County'] = election_data['County'].str.replace('County', '')
election_data['County'] = election_data['County'].str.strip()
election_data['State'] = election_data['State'].str.strip()
#election_data
states = {
        'AK': 'Alaska',
        'AL': 'Alabama',
        'AR': 'Arkansas',
        'AS': 'American Samoa',
        'AZ': 'Arizona',
        'CA': 'California',
        'CO': 'Colorado',
        'CT': 'Connecticut',
        'DC': 'District of Columbia',
        'DE': 'Delaware',
        'FL': 'Florida',
```

```
        'WI': 'Wisconsin',
        'WV': 'West Virginia',
        'WY': 'Wyoming'
}
election_data = election_data.replace({"State":states})
demographic_data['State'] = demographic_data['State'].str.strip()
demographic_data['County'] = demographic_data['County'].str.strip()

demographic_data['State'] = demographic_data['State'].str.upper()
election_data['State'] = election_data['State'].str.upper()
election_data['County'] = election_data['County'].str.upper()
demographic_data['County'] = demographic_data['County'].str.upper()
election_data
#demographic_data
#election_data.count()
```

So first we remove county from one of the dataframe and strip all additional spaces. Then we created a dictionary to convert all the abbreviation to full name and make all the changes into uppercase. See the screenshot below.

	Year	State	County	Office	Democratic Votes	Republican Votes
0	2018	ARIZONA	APACHE	US Senator	16298.0	7810.0
1	2018	ARIZONA	COCHISE	US Senator	17383.0	26929.0
2	2018	ARIZONA	COCONINO	US Senator	34240.0	19249.0
3	2018	ARIZONA	GILA	US Senator	7643.0	12180.0
4	2018	ARIZONA	GRAHAM	US Senator	3368.0	6870.0
5	2018	ARIZONA	LA PAZ	US Senator	1609.0	3265.0
6	2018	ARIZONA	MARICOPA	US Senator	732671.0	672505.0

At this point, both datasets should share the same keys and ready to merge. So we performed the merge action. We choose inner join because we want to merge the datasets together based on their similarities.

```
#Task 2
merged_data = pd.merge(election_data, demographic_data, how='inner', on=['State','County'])
#merged_data[merged_data['County']==""]
merged_data
```

	Year	State	County	Office	Democratic Votes	Republican Votes	FIPS	Total Population	Citizen Voting-Age Population	Percent White, 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	Percent Unemployed	Percent Less than High School Degree	Percent Less than Bachelor's Degree	Percent Rural
0	2018	ARIZONA	APACHE	US Senator	16298.0	7810.0	4001	72346	0	18.571863	...	5.947806	1.719515	50.598513	45.854643	13.322091	32460	15.807433	21.758252	88.941063	74.061076
1	2018	ARIZONA	COCHISE	US Senator	17383.0	26929.0	4003	128177	92915	56.299492	...	34.403208	11.458374	49.069646	37.902276	19.756275	45383	8.567108	13.409171	76.837055	36.301067
2	2018	ARIZONA	COCONINO	US Senator	34240.0	19249.0	4005	138064	104265	54.619597	...	13.711033	4.825298	50.581614	48.946141	10.873943	51106	8.238305	11.085381	65.791439	31.466066
1198	2018	WYOMING	UINTA	US Senator	1371.0	4713.0	56041	20893	14355	87.718375	...	8.959939	3.986981	49.327526	43.205858	10.678218	53323	6.390755	10.361224	81.793082	43.095937
1199	2018	WYOMING	WASHAKIE	US Senator	588.0	2423.0	56043	8351	0	82.397318	...	13.962400	3.783978	51.359119	34.774279	19.650341	46212	7.441860	12.577108	78.923920	35.954529

1200 rows × 21 columns

<u>At the end after the merging, we got 1200 rows of observation and 21 attributes</u>

3. The new merged dataset has <u>21 variables</u>. They are the following:

```
Year                                        int64
State                                      object
County                                     object
Office                                     object
Democratic Votes                          float64
Republican Votes                          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
```

The variables <u>'Year'</u> and <u>'Office'</u> have a single value for all observations.
Therefore they can be assumed to be redundant/irrelevant variables and dropped from the dataset. If we drop it, then the total number of variables would become 19.

```
print(merged_data['Year'].unique())
```

```
[2018]
```

```
print(merged_data['Office'].unique())
```

```
['US Senator']
```

4. There aren't any null values in the dataset.

```
merged_data.isnull().sum()
#merged_data
```

```
Year                                      0
State                                     0
County                                    0
Office                                    0
Democratic Votes                          0
Republican Votes                          0
FIPS                                      0
Total Population                          0
Citizen Voting-Age 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
```

But the attribute 'Citizen Voting-Age Population' has more than 50% observations with value 0. So we assume them to be missing value as the value 0 does not make sense for this attribute. We dropped the attribute 'Citizen Voting-Age Population' from the dataset.

```
a = merged_data['County'].count()
b = merged_data['County'][merged_data['Citizen Voting-Age Population']==0].count()
print(a)
print(b)
print("Percentage of missing values = ",b / a * 100)
```

```
1200
680
Percentage of missing values =  56.666666666666664
```

5. By calling df['Party'], this command creates a new column and set 1 or 0 based on if democratic has more votes than republican as showing below.

```
[47]: #Task 5
merged_data['Party'] = np.where(merged_data['Democratic Votes'] > merged_data['Republican Votes'], 1, 0)
merged_data[['Democratic Votes', 'Republican Votes', 'Party']]
```

[47]:

	Democratic Votes	Republican Votes	Party
0	16298.0	7810.0	1
1	17383.0	26929.0	0
2	34240.0	19249.0	1
3	7643.0	12180.0	0
4	3368.0	6870.0	0
5	1609.0	3265.0	0
6	732671.0	672505.0	1

6. The mean population is higher for Democratic counties.

H0: mean for Democratic party's population **is** the same as Republican party's population.
H1: mean for Democratic party's population **is not** the same as Republican party's population.

```
print("Mean county population:")
print("For Democratic counties :",demo['Total Population'].mean())
print("For Republican counties :",repub['Total Population'].mean())
print(demo['Total Population'].mean() > repub['Total Population'].mean())
```

```
Mean county population:
For Democratic counties : 300998.3169230769
For Republican counties : 53974.214857142855
True
```

```
[statistic,pvalue] = st.ttest_ind(demo['Total Population'],repub['Total Population'],equal_var=False)
print("t =",statistic)
print("p value =",pvalue," = ",format(pvalue,'.16f'))
print(pvalue < alpha)
```

```
t = 8.001207114045041
p value = 2.0965719353509958e-14  =  0.0000000000000210
True
```

As the conclusion shows that the p value (2.0965719353509958e-14 = 0.0000000000000210) < the significant level (0.05). The difference is statistically significant.
Therefore, we can safely reject the null hypothesis and accept the population is different hypothesis.

7. The mean median household income is higher for Democratic counties.

H0: mean for Democratic party's median household income **is** the same as Republican party's median household income.
H1: mean for Democratic party's median household income **is not** the same as Republican party's median household income.

```
print("Mean median household income:")
print("For Democratic counties :",demo['Median Household Income'].mean())
print("For Republican counties :",repub['Median Household Income'].mean())
print(demo['Median Household Income'].mean() > repub['Median Household Income'].mean())
```

```
Mean median household income:
For Democratic counties : 53798.732307692306
For Republican counties : 48724.15085714286
True
```

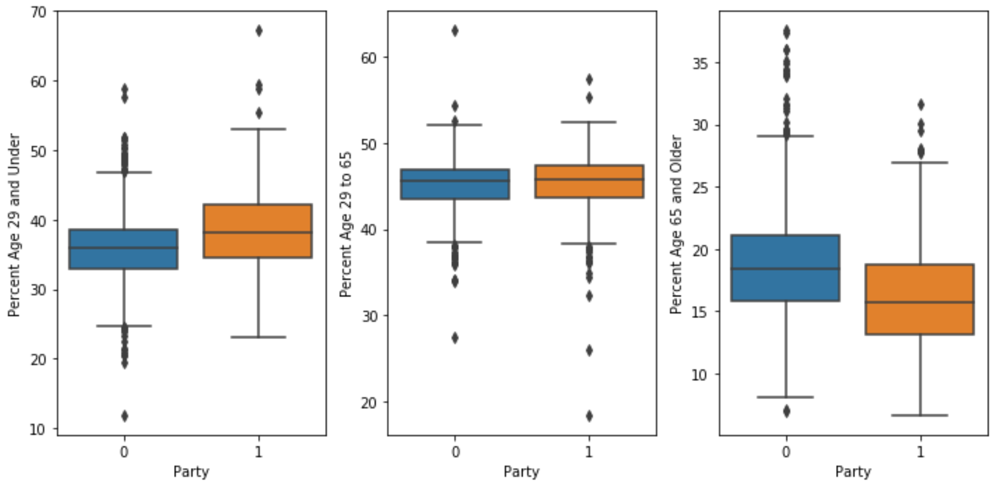
```
[statistic,pvalue] = st.ttest_ind(demo['Median Household Income'],repub['Median Household Income'],equal_var=False)
print("t =",statistic)
print("p value =",pvalue," = ",format(pvalue,'.10f'))
print(pvalue < alpha)
```

```
t = 5.507012409466501
p value = 6.173239891230373e-08  =  0.0000000617
True
```

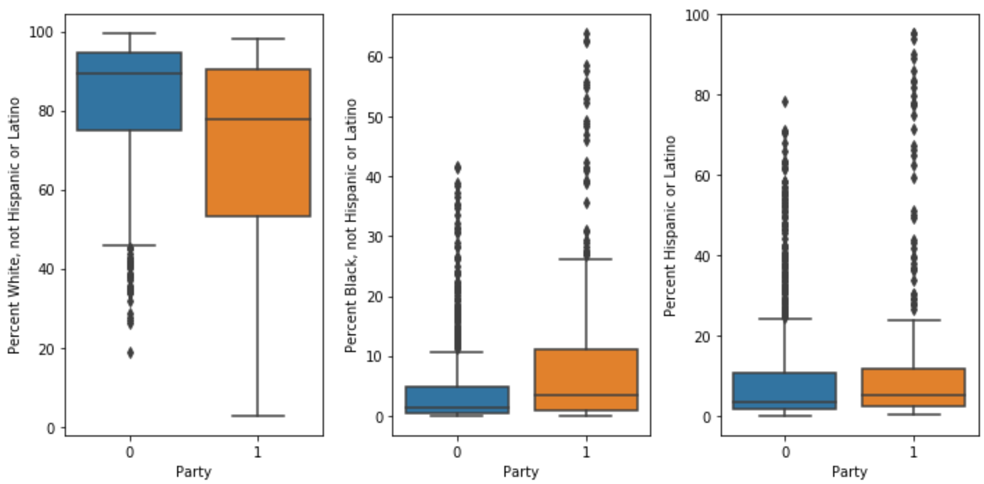
As the conclusion shows that the p value(6.173239891230373e-08 = 0.0000000617) < the significant level (0.05). The difference is statistically significant.
Therefore, we can safely reject the null hypothesis and accept the median household income for both parties is different.

8. Blue = Republican Counties
Orange = Democratic Counties

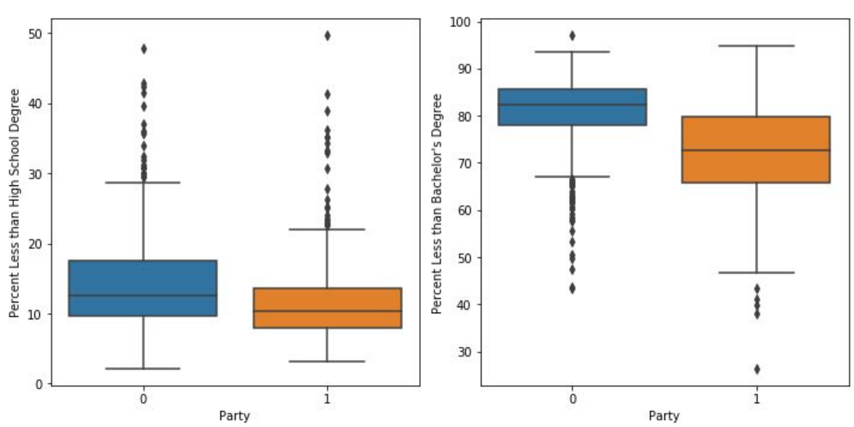
<u>For Age</u>



<u>For Race</u>



<u>For Education</u>



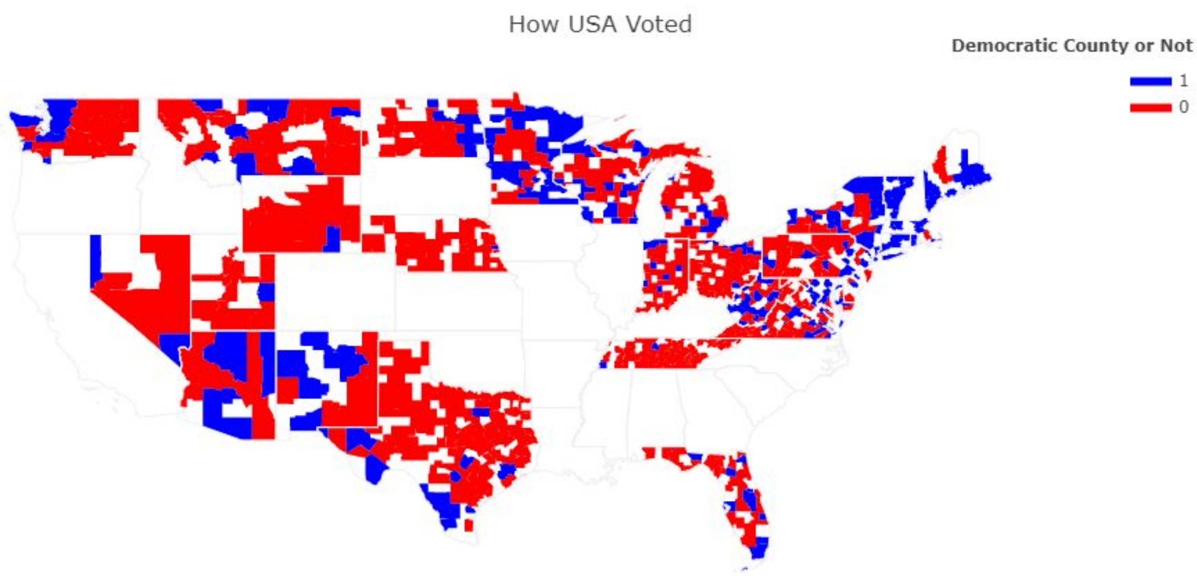
9. Based on our analysis, the features in the dataset that we think are more important to determine whether a county should be labeled as Democratic or Republican are Age, Race, and Education.

For Age, we noticed that people in the age bracket (29 - 65) are more involved in politics as they have more percentage of votes, which makes that age group more weighted than the others. The vote share of people above 65 years of age is low. The variables are 'Percent Age 29 and Under' and 'Percent Age 65 and Older'.

As for Race, we notice that white voters tend to be more on the Republican side and for minorities, it seems like they favor the Democratic Party. The variables are 'Percent White, not Hispanic or Latino','Percent Black, not Hispanic or Latino' and 'Percent Hispanic or Latino'.

As for Education, people without High School Degree have less vote share and have a higher chance of voting for the Republican Party. The same can be also said about people without a Bachelor's Degree. The variables are 'Percent Less than High School Degree' and "Percent Less than Bachelor's Degree".

10.



Blue denotes the counties where Democrats secured more votes than Republicans.
Red denotes the counties where Republicans secured more votes than Democrats.