In [1]:
```
#Importing required libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.figure_factory as ff
import chart_studio.plotly as py
import scipy.stats as st
import re
import numpy as np
```

In [2]:
```
#Loading the datasets
demographics = pd.read_csv('demographics_train.csv')
election = pd.read_csv('election_train.csv')
```

In [3]:
```
#Inspecting election:
election.head()
```

Out[3]:

	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
3	2018	AZ	Cochise County	US Senator	Republican	26929.0
4	2018	AZ	Coconino County	US Senator	Democratic	34240.0

In [4]:
```
#Inspecting demographics:
demographics.head()
```

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	Percent Female	Percent Age 29 and Under	F and
0	Wisconsin	La Crosse	55063	117538	0	90.537528	1.214075	1.724549	2.976059	51.171536	43.241335	14.
1	Virginia	Alleghany	51005	15919	12705	91.940449	5.207614	1.432251	1.300333	51.077329	31.660280	23.
2	Indiana	Fountain	18045	16741	12750	95.705155	0.400215	2.359477	1.547100	49.770026	35.899887	18.
3	Ohio	Geauga	39055	94020	0	95.837056	1.256116	1.294405	2.578175	50.678579	36.281642	18.
4	Wisconsin	Jackson	55053	20566	15835	86.662453	1.983857	3.082758	1.376058	46.649810	36.292911	17.

In [5]:
```
#Inspecting the dimensions of election data
print('election dimensions:{}'.format(election.shape))
```

```
election dimensions:(2410, 6)
```

In [6]:
```
#Inspecting the dimensions of demographics data
print('demographics dimensions:{}'.format(demographics.shape))
```

```
demographics dimensions:(1216, 17)
```

In [7]:
```
#Task 1:
#Reshaping dataset election_train from long format to wide format
election_tidy = pd.pivot_table(election,index=['Year','State','County','Office'], columns='Part
election_tidy.head()
```

Out[7]:

	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

In [8]:
```
#Inspecting the dimensions of election data
print('election_tidy dimensions:{}'.format(election_tidy.shape))
```

```
election_tidy dimensions:(1205, 6)
```

In [9]:
```
#Accounting for inconsistencies in both datasets before merging
##As in the election dataset, each county name is followed by the string 'County' which is not
election_tidy['County'] = election_tidy['County'].str.replace('County','').str.strip()
```

In [10]:

```
##As in the election dataset, each State name is abbreviation which is not the case with demogr
change_values = {
    'Alabama': 'AL',
    'Alaska': 'AK',
    'Arizona': 'AZ',
    'Arkansas': 'AR',
    'California': 'CA',
    'Colorado': 'CO',
    'Connecticut': 'CT',
    'Delaware': 'DE',
    'District of Columbia': 'DC',
    'Florida': 'FL',
    'Georgia': 'GA',
    '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',
    'Palau': 'PW',
    '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',
}
demographics['State']=demographics['State'].map(change_values)
```

In [11]:

```
##As there are some case-sensitive issues in names of Counties in both datasets, converting bot
demographics['County'] = demographics['County'].str.lower()
election_tidy['County'] = election_tidy['County'].str.lower()
```

In [12]:
```
#Task 2:
#Merging reshaped dataset election_train with dataset demographics_train
merged_data=pd.merge(election_tidy,demographics, how = 'inner', on = ['State','County'])
merged_data.head()
```

Out[12]:

	Year	State	County	Office	Democratic	Republican	FIPS	Total Population	Citizen Voting-Age Population	Percent White, not Hispanic or Latino	...	Percent Hispanic or Latino	Perc Fore B
0	2018	AZ	apache	US Senator	16298.0	7810.0	4001	72346	0	18.571863	...	5.947806	1.719
1	2018	AZ	cochise	US Senator	17383.0	26929.0	4003	128177	92915	56.299492	...	34.403208	11.458
2	2018	AZ	coconino	US Senator	34240.0	19249.0	4005	138064	104265	54.619597	...	13.711033	4.825
3	2018	AZ	gila	US Senator	7643.0	12180.0	4007	53179	0	63.222325	...	18.548675	4.249
4	2018	AZ	graham	US Senator	3368.0	6870.0	4009	37529	0	51.461536	...	32.097844	4.385

5 rows × 21 columns

In [13]:
```
#Inspecting the dimensions of merged data
print('merged_data dimensions:{}'.format(merged_data.shape))
```

```
merged_data dimensions:(1200, 21)
```

In [14]:
```
#Task 3:
#Exploring the merged dataset
merged_data.shape
```

Out[14]: (1200, 21)

In [15]:
```
#Number of variables = 21
#Type of the variables
merged_data.dtypes
```

Out[15]:
```
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 [16]:
```
1 merged_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1200 entries, 0 to 1199
Data columns (total 21 columns):
Year                                  1200 non-null int64
State                                 1200 non-null object
County                                1200 non-null object
Office                                1200 non-null object
Democratic                            1200 non-null float64
Republican                            1200 non-null float64
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
dtypes: float64(13), int64(5), object(3)
memory usage: 206.2+ KB
```

In [17]:
```
1 merged_data.Year.unique()
```

Out[17]:
```
array([2018], dtype=int64)
```

In [18]:
```
1 merged_data.Office.unique()
```

Out[18]:
```
array(['US Senator'], dtype=object)
```

In [19]:
```
1 #As seen both variables Year and Office have same values for all 1200 observations and are irre
2 #Dealing with Irrelevant variables: It is better to drop them
3 merged_data = merged_data.drop(['Year','Office'], axis =1)
```

In [20]:
```
1 merged_data.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
Democratic                           1200 non-null float64
Republican                           1200 non-null float64
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
dtypes: float64(13), int64(4), object(2)
memory usage: 187.5+ KB
```

In [21]:
```
1 #Task 4:
2 #Searching the merged data for missing values
3 merged_data.replace(0, np.nan, inplace=True)
4 merged_data.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
Democratic                           1195 non-null float64
Republican                           1195 non-null float64
FIPS                                 1200 non-null int64
Total Population                     1200 non-null int64
Citizen Voting-Age Population        520 non-null float64
Percent White, not Hispanic or Latino 1200 non-null float64
Percent Black, not Hispanic or Latino 1155 non-null float64
Percent Hispanic or Latino           1195 non-null float64
Percent Foreign Born                 1197 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                   1197 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                        1181 non-null float64
dtypes: float64(14), int64(3), object(2)
memory usage: 187.5+ KB
```

In [22]:
```
1 #As we search and analyse, Democratic and Republican variables have 5 observations that are mis
2 ##Citizen Voting-Age Population variable has 0 values for more than half the number of observat
3 #for counties where total population is highly significant - can be attributed as missing value
4 ##Rest of the variables have 0 values, which seems logically right in terms of percentages divi
```

In [23]:
```
#Dealing with missing values
merged_data.replace(np.nan, 0, inplace=True)
merged_data.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
Democratic                           1200 non-null float64
Republican                           1200 non-null float64
FIPS                                 1200 non-null int64
Total Population                     1200 non-null int64
Citizen Voting-Age Population        1200 non-null float64
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
dtypes: float64(14), int64(3), object(2)
memory usage: 187.5+ KB
```

In [24]:
```
merged_data['Democratic'].replace(0, np.nan, inplace=True)
merged_data['Republican'].replace(0, np.nan, inplace=True)
merged_data.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
Democratic                           1195 non-null float64
Republican                           1195 non-null float64
FIPS                                 1200 non-null int64
Total Population                     1200 non-null int64
Citizen Voting-Age Population        1200 non-null float64
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
dtypes: float64(14), int64(3), object(2)
memory usage: 187.5+ KB
```

In [25]:
```
#Dropping 5 observations having null values for Decromatic and Republican variables
merged_data=merged_data.dropna()
```

In [26]:
```
merged_data.shape
```

Out[26]: (1195, 19)

In [27]:
```
#Dropping the variable Citizen Voting-age Population as it has many observations that have miss
merged_data = merged_data.drop(['Citizen Voting-Age Population'], axis =1)
```

In [28]:
```
merged_data.shape
```

Out[28]: (1195, 18)

In [29]:
```
#Task 5
#Creating a new variable named "Party" that labels each county as Democratic or Republican
merged_data['Party'] = np.where(merged_data['Democratic'] > merged_data['Republican'], 1, 0)
merged_data.head()
```

Out[29]:

	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	Per Ag U
0	AZ	apache	16298.0	7810.0	4001	72346	18.571863	0.486551	5.947806	1.719515	50.598513	45.85
1	AZ	cochise	17383.0	26929.0	4003	128177	56.299492	3.714395	34.403208	11.458374	49.069646	37.90
2	AZ	coconino	34240.0	19249.0	4005	138064	54.619597	1.342855	13.711033	4.825298	50.581614	48.94
3	AZ	gila	7643.0	12180.0	4007	53179	63.222325	0.552850	18.548675	4.249798	50.296170	32.23
4	AZ	graham	3368.0	6870.0	4009	37529	51.461536	1.811932	32.097844	4.385942	46.313518	46.39

In [30]:
```
merged_data.shape
```

Out[30]: (1195, 19)

In [31]:
```
#Task 6
#Computing the mean population for Democratic counties and Republican counties
merged_data.groupby('Party').agg({"Total Population" : np.mean})
```

Out[31]:

Party	Total Population
0	53864.672414
1	300998.316923

In [32]:
```
#Which one is higher? Mean population of Democractic is higher than mean population of Republic
```

In [33]:
```
#2-sample t hypothesis test
[statistic, pvalue]=st.ttest_ind(merged_data.loc[merged_data['Party'] == 1, 'Total Population']
print(statistic)
print(pvalue)
```

```
8.004638577960957
2.0478717602973023e-14
```

In [34]:
```
#Result of the test
##p-value= 1.0239358801486512e-14
#Conclusion
## Since p-value < significance level - reject the null hypothesis
```

In [35]:
```
#Task 7:
#Compute the mean median household income for Democratic counties and Republican counties
merged_data.groupby('Party').agg({"Median Household Income" : np.mean})
```

Out[35]:

Party	Median Household Income
0	48746.819540
1	53798.732308

In [36]:
```
#Which one is higher? Mean Median Household Income of Democractic is higher than mean Median Ho
```

In [37]:
```
#2-sample t hypothesis test
[statistic, pvalue]=st.ttest_ind(merged_data.loc[merged_data['Party'] == 1, 'Median Household I
print(statistic)
print(pvalue)
```

```
5.479141589767387
7.149437363182598e-08
```

In [38]:
```
#Result of the test
##p-value= 3.574718681591299e-08
#Conclusion
## Since p-value < significance level - reject the null hypothesis
```

In [39]:
```
#Task 8:
#Getting the Descriptive statistics:
merged_data.describe()
```

Out[39]:

	Democratic	Republican	FIPS	Total Population	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino	Percent Foreign Born	
count	1195.000000	1195.000000	1195.000000	1.195000e+03	1195.000000	1195.000000	1195.000000	1195.000000	119
mean	25132.953138	20443.984100	38283.497071	1.210768e+05	79.128457	5.563599	10.509368	5.076938	4
std	72648.876156	45260.494035	13009.708572	3.189417e+05	19.756652	9.290601	15.787826	6.065171	
min	6.000000	46.000000	4001.000000	7.600000e+01	2.776702	0.000000	0.000000	0.000000	2
25%	1424.500000	2669.000000	27144.000000	1.212200e+04	70.187494	0.536934	1.819735	1.466434	4
50%	4222.000000	6694.000000	39137.000000	3.265300e+04	86.809245	1.607865	3.880463	2.865353	5
75%	14265.000000	16739.000000	48417.000000	8.587900e+04	93.868714	6.440764	10.994025	6.325443	5
max	881802.000000	672505.000000	56043.000000	4.434257e+06	99.627329	63.953279	95.479801	52.229868	5

In [40]:
```
#For Age Comparison
#Creating separate column for Percent Age between 30 and 64
merged_data['Percent Age 30 to 64'] = 100 - (merged_data["Percent Age 29 and Under"] + merged_d
```

In [41]:
```
#Grouping by Different age categories:
Under30 = merged_data.groupby('Party', as_index=False)['Percent Age 29 and Under'].mean()
Under65 = merged_data.groupby('Party', as_index=False)['Percent Age 30 to 64'].mean()
Older65 = merged_data.groupby('Party', as_index=False)['Percent Age 65 and Older'].mean()
```

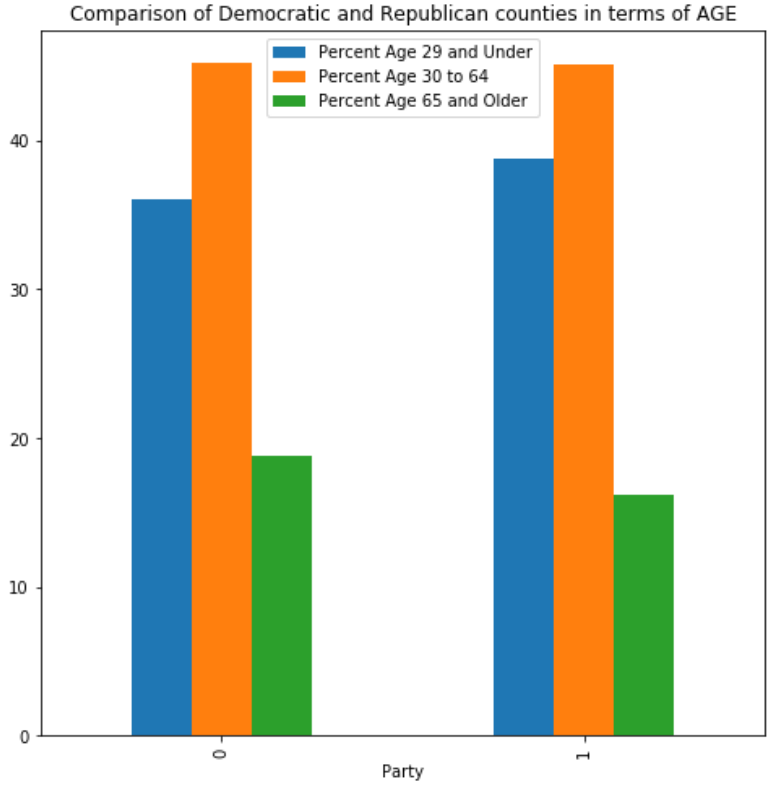
In [42]:
```
#Merging datasets:
age1 = pd.merge(Under30,Under65, on=['Party'])
age = pd.merge(age1, Older65, on= ['Party'])
age
```

Out[42]:

	Party	Percent Age 29 and Under	Percent Age 30 to 64	Percent Age 65 and Older
0	0	36.005719	45.166015	18.828267
1	1	38.726959	45.078214	16.194826

In [43]:
```
#Plotting bar graph for Democratic and Republican parties based on Age:

ax1 = age.plot.bar(x ='Party',  title='Comparison of Democratic and Republican counties in term
```



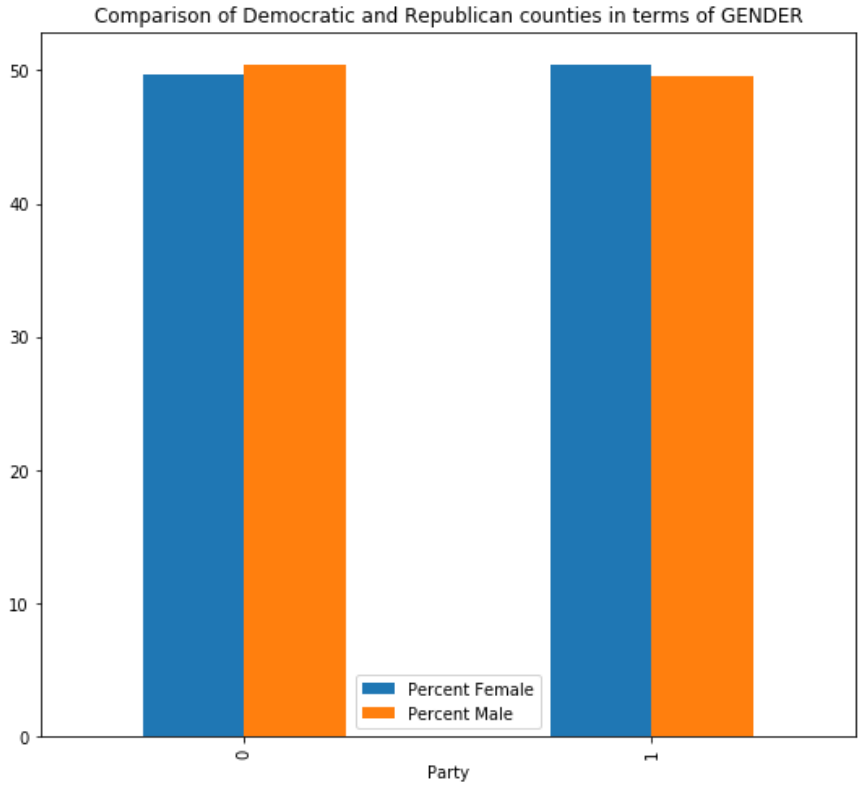
In [44]:
```
#For Gender Comparison
#Creating a separate column for Percent Male in each county
merged_data['Percent Male'] = 100 - merged_data['Percent Female']
```

In [45]:
```
#Grouping by gender and merging to form a dataset gender
female = merged_data.groupby('Party', as_index=False)['Percent Female'].mean()
male = merged_data.groupby('Party', as_index=False)['Percent Male'].mean()
gender = pd.merge(female,male, on=['Party'])
gender
```

Out[45]:

	Party	Percent Female	Percent Male
0	0	49.630898	50.369102
1	1	50.385433	49.614567

In [46]:
```
#Plotting Comparison between Democratic and Republican counties based on Gender:
ax2 = gender.plot.bar(x ='Party', title = 'Comparison of Democratic and Republican counties in
```



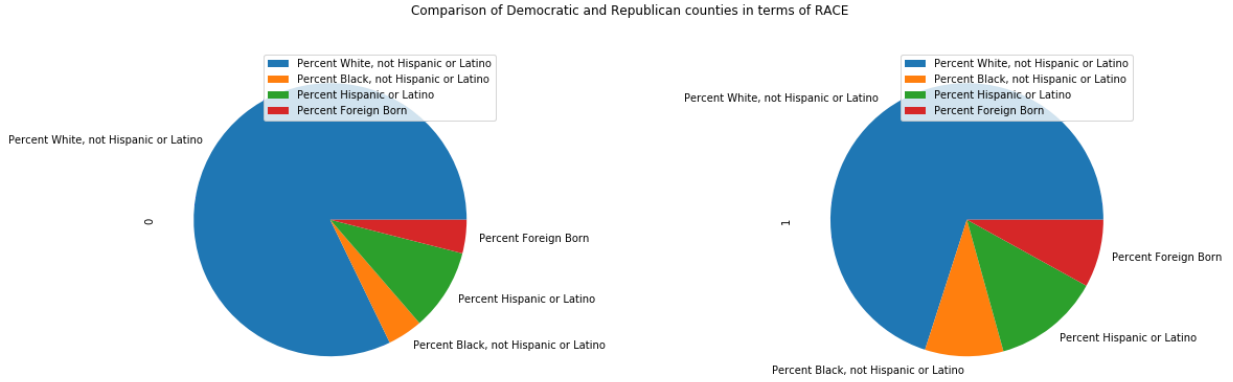
In [47]:
```
#For Race and Ethnicity comparison
#Grouping in terms of Race and Ethnicity:
White = merged_data.groupby('Party', as_index = False)['Percent White, not Hispanic or Latino']
Black = merged_data.groupby('Party', as_index = False)['Percent Black, not Hispanic or Latino']
Hispanic = merged_data.groupby('Party', as_index = False)['Percent Hispanic or Latino'].mean()
Foreign=merged_data.groupby('Party', as_index = False)['Percent Foreign Born'].mean()
```

In [48]:
```
#Merging to form a dataset race
race1 = pd.merge(White,Black, on = ['Party'])
race2 = pd.merge(race1, Hispanic, on = ['Party'])
race = pd.merge(race2, Foreign, on = ['Party'])
race
```

Out[48]:

	Party	Percent White, not Hispanic or Latino	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino	Percent Foreign Born
0	0	82.656646	4.189241	9.733094	3.990096
1	1	69.683766	9.242649	12.587391	7.986330

In [49]:
```
#Plotting Comparison of Democratic and Republic counties in terms of Race:
race_transpose = race.T
race_drop = race_transpose.drop(['Party'])
race_drop
ax3 = race_drop.plot.pie(subplots=True, figsize=(20,6), title='Comparison of Democratic and Rep
```



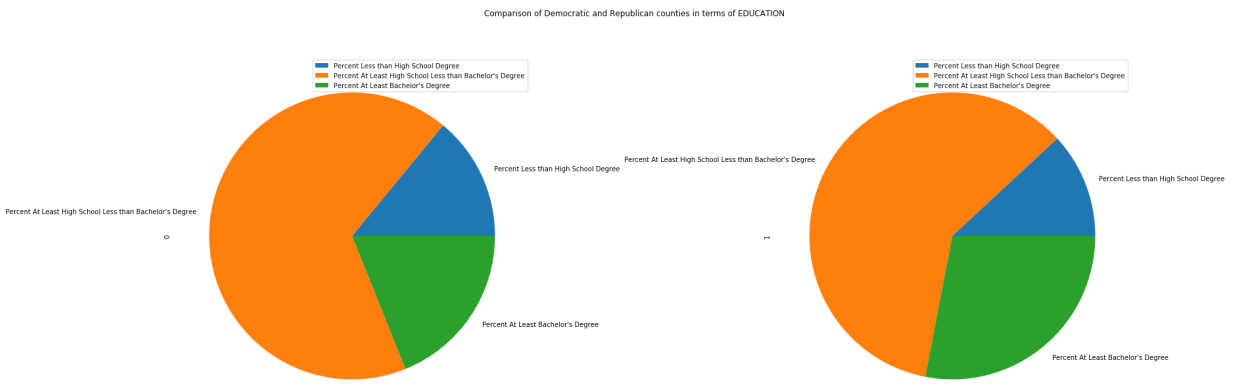
In [50]:
```
#Creating separate column for right percentages for Education
merged_data["Percent At Least High School Less than Bachelor's Degree"] = merged_data["Percent
merged_data["Percent At Least Bachelor's Degree"] = 100 - merged_data["Percent Less than Bachel
```

In [51]:
```
# Grouping and merging in terms of Education:
LessHighSchool = merged_data.groupby('Party', as_index = False)['Percent Less than High School
LessBachelor = merged_data.groupby('Party', as_index = False)["Percent At Least High School Les
BachelorUp = merged_data.groupby('Party', as_index = False)["Percent At Least Bachelor's Degree
ed1 = pd.merge(LessHighSchool, LessBachelor, on = ['Party'])
ed = pd.merge(ed1, BachelorUp, on = ['Party'])
ed
```

Out[51]:

	Party	Percent Less than High School Degree	Percent At Least High School Less than Bachelor's Degree	Percent At Least Bachelor's Degree
0	0	14.009112	67.086315	18.904573
1	1	11.883760	60.084465	28.031775

In [52]:
```
#Plotting Comparison of Democratic and Republican counties in terms of Education
ed_transpose = ed.T
ed_drop = ed_transpose.drop(['Party'])
graph3 = ed_drop.plot.pie(subplots = True, figsize = (30,10), title = 'Comparison of Democratic
```



In [53]:

```
#Task 9
#As we are looking at percentages, other than the variable GENDER, all other variables are impo
#labelled as Democractic and Republican
```

In [55]:

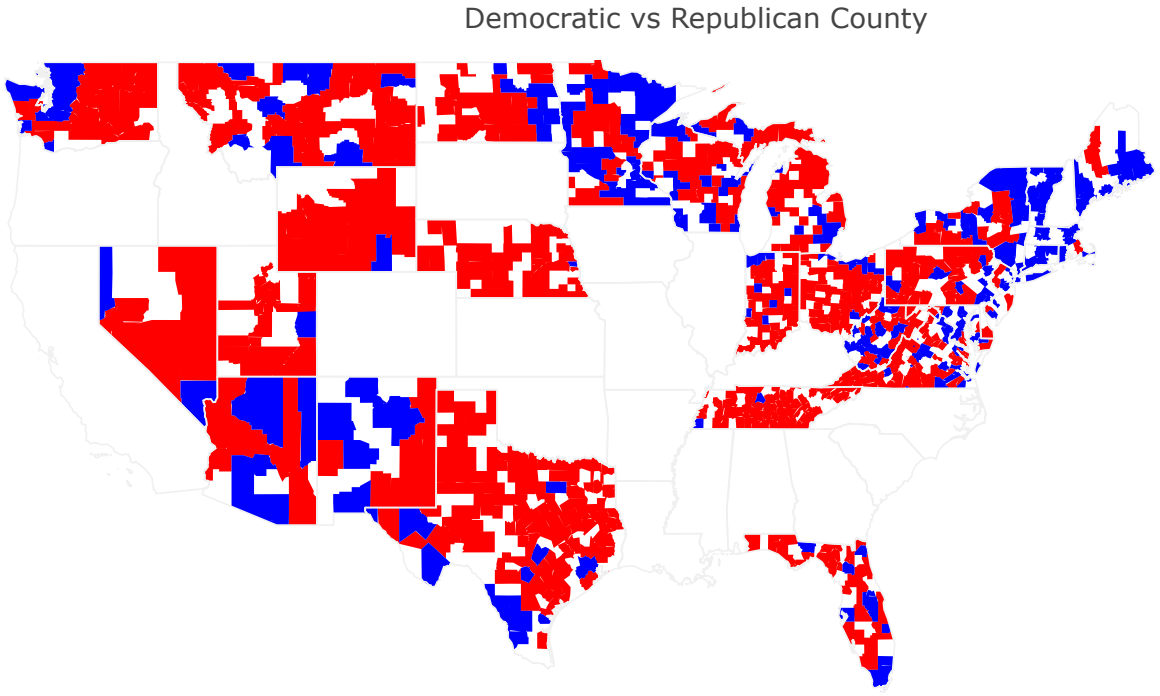
```
#Task 10
#Creating a map of Democratic counties and Republican counties using the counties' FIPS codes
merged_data['FIPS'] = merged_data['FIPS'].apply(lambda x: str(x).zfill(4))

colorscale = ["red","blue"]

fips = merged_data['FIPS'].tolist()
values1 = merged_data['Party'].tolist()


fig = ff.create_choropleth(
    fips = fips, values = values1,
    colorscale =  colorscale,
    show_state_data = True,
    show_hover = True, centroid_marker = {'opacity': 0},
    asp = 2.9, title = 'Democratic vs Republican County',
    legend_title = 'D(1) or R(0)'
)

fig.layout.template = None
fig.show()
```



In [ ]:

```

```