In [1]:

```
import pandas as pd
import matplotlib.pyplot as ply
import seaborn as sns
import numpy as np
import scipy.stats as st
```

Task 01: Reshape the data from long to wide. Split Party into Democrative

In [2]:

```
demo_data = pd.read_csv("demographics_train.csv")
election_data = pd.read_csv("election_train.csv")
election_data = pd.pivot_table(election_data, index=['Year','Cou
nty','State','Office'],columns = 'Party', aggfunc = np.sum, valu
es='Votes').reset_index()
election_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1205 entries, 0 to 1204
Data columns (total 6 columns):
Year              1205 non-null int64
County            1205 non-null object
State             1205 non-null object
Office            1205 non-null object
Democratic        1205 non-null float64
Republican        1205 non-null float64
dtypes: float64(2), int64(1), object(3)
memory usage: 56.6+ KB
```

Task 02 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 [3]:

```
#fix inconsisties with column State in both demo_data and election_data
change_values = {
'Alabama': 'AL', 'Alaska': 'AK', 'Arizona': 'AZ', 'Arkansas': 'AR', 'California': 'CA', 'Colorado': 'CO',
'Connecticut': 'CT', 'Delaware': 'DE', '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', 'Ohio': 'OH', 'Oklahoma': 'OK',
'Oregon': 'OR', 'Pennsylvania': 'PA', 'Rhode Island': 'RI', 'South Carolina': 'SC', 'South Dakota': 'SD',
'Tennessee': 'TN', 'Texas': 'TX', 'Utah': 'UT', 'Vermont': 'VT', 'Virginia': 'VA', 'Washington': 'WA',
'West Virginia': 'WV', 'Wisconsin': 'WI', 'Wyoming': 'WY'}
demo_data['State'] = demo_data['State'].map(change_values)
```

In [4]:

```
#fix inconsisties with the column County in both demo_data and election_data
election_data['County'] = election_data['County'].str.replace(' County', '')
election_data['County'] = election_data['County'].str.lower()
demo_data['County'] = demo_data['County'].str.lower()
```

In [5]:

```
df=pd.merge(election_data, demo_data,on=['County','State'],how='inner') #inner means intersection
pd.set_option('display.max_rows',10)
pd.set_option('display.max_columns',21)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1200 entries, 0 to 1199
```

```
Data columns (total 21 columns):
Year                                     1200 non-nu
ll int64
County                                   1200 non-nu
ll object
State                                    1200 non-nu
ll object
Office                                   1200 non-nu
ll object
Democratic                               1200 non-nu
ll float64
Republican                               1200 non-nu
ll float64
FIPS                                     1200 non-nu
ll int64
Total Population                         1200 non-nu
ll int64
Citizen Voting-Age Population            1200 non-nu
ll int64
Percent White, not Hispanic or Latino    1200 non-nu
ll float64
Percent Black, not Hispanic or Latino    1200 non-nu
ll float64
Percent Hispanic or Latino               1200 non-nu
ll float64
Percent Foreign Born                     1200 non-nu
ll float64
Percent Female                           1200 non-nu
ll float64
Percent Age 29 and Under                 1200 non-nu
ll float64
Percent Age 65 and Older                 1200 non-nu
ll float64
Median Household Income                  1200 non-nu
ll int64
Percent Unemployed                       1200 non-nu
ll float64
Percent Less than High School Degree     1200 non-nu
ll float64
Percent Less than Bachelor's Degree      1200 non-nu
ll float64
Percent Rural                            1200 non-nu
ll float64
dtypes: float64(13), int64(5), object(3)
memory usage: 206.2+ KB
```

Task 03: 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 [6]:

```
# Task 04: Search the dataset for missing values
print('Number of missing values per column:')
countMissing = df.isin([0]).sum()
print(countMissing)
```

```
Number of missing values per column:
Year                                     0
County                                   0
State                                    0
Office                                   0
Democratic                               5
                                        ..
Median Household Income                  0
Percent Unemployed                       3
Percent Less than High School Degree     0
Percent Less than Bachelor's Degree      0
Percent Rural                           19
Length: 21, dtype: int64
```

In [7]:

```python
df = df.drop('Citizen Voting-Age Population',  axis=1)

df = df[df.Democratic != 0]
df=df.rename(columns = {'Percent Hispanic or Latino':'His'})
df = df[df.His != 0]
df=df.rename(columns = {'His':'Percent Hispanic or Latino'})

df=df.rename(columns = {'Percent Hispanic or Latino':'His'})
df = df[df.His != 0]
df=df.rename(columns = {'His':'Percent Hispanic or Latino'})

df=df.rename(columns = {'Percent Unemployed':'UN'})
df = df[df.UN != 0]
df=df.rename(columns = {'UN':'Percent Unemployed'})


df=df.rename(columns = {'Percent Black, not Hispanic or Latino':'Black'})
df[df.Black == 0]['Black'] = df.Black.mean()
df=df.rename(columns = {'Black':'Percent Black, not Hispanic or Latino'})
countMissing = df.isin([0]).sum()
print(countMissing)
```

```
Year                                   0
County                                 0
State                                  0
Office                                 0
Democratic                             0
                                      ..
Median Household Income                0
Percent Unemployed                     0
Percent Less than High School Degree   0
Percent Less than Bachelor's Degree    0
Percent Rural                         19
Length: 20, dtype: int64

/Users/kirunhaque/anaconda3/lib/python3.7/site-packa
ges/ipykernel_launcher.py:18: SettingWithCopyWarning
:
A value is trying to be set on a copy of a slice fro
m a DataFrame.
Try using .loc[row_indexer,col_indexer] = value inst
ead

See the caveats in the documentation: http://pandas.
pydata.org/pandas-docs/stable/indexing.html#indexing
-view-versus-copy
```

Task 05: 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 [8]:

```
df['Party'] = np.where(df['Democratic'] > df['Republican'], 1, 0
)
my_column = df.pop('Party')
df.insert(6,my_column.name,my_column)
```

Task 06: 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 [9]:

```
dem = df[df.Party == 1]['Total Population'].mean()
rep =df[df.Party == 0]['Total Population'].mean()
print("Democratic population mean: ",dem)
print("Republican population mean: ",rep)
```

```
Democratic population mean:  300998.3169230769
Republican population mean:  54354.71693735499
```

In [10]:

```
[statistic, pvalue] = st.ttest_ind(df[df.Party == 0]['Total Population'], df[df.Party == 1]['Total Population'], equal_var = False)
print("\nStatistic value: ", statistic)
print("P-value: ", pvalue/2)
```

```
Statistic value:  -7.988095948482815
P-value:  1.1449459405635236e-14
```

Task 07: 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 [11]:

```
demHouseMed = df[df.Party == 1]['Median Household Income'].mean()
repHouseMed = df[df.Party == 0]['Median Household Income'].mean()
print("Mean for Democratic Median Household Income : ", demHouseMed)
print("Mean for Republican Median Household Income: ", repHouseMed)

[statistic, pvalue] = st.ttest_ind(df[df.Party == 1]['Median Household Income'],df[df.Party == 0]['Median Household Income'], equal_var = False)
print("\nStatistic value for Median Household Income: ",statistic)
print("pvalue for Median Household Income: ",pvalue/2)
```

```
Mean for Democratic Median Household Income :  53798.732307692306
Mean for Republican Median Household Income:  48770.51276102088

Statistic value for Median Household Income:  5.449357147792327
pvalue for Median Household Income:  4.1785610883104794e-08
```

Task 08: 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 [12]:

```
Democratic_Summary = df[df.Party == 1][['Percent White, not Hisp
anic or Latino','Percent Black, not Hispanic or Latino','Percent
Hispanic or Latino','Percent Foreign Born','Percent Female','Per
cent Age 29 and Under','Percent Age 65 and Older','Percent Less
than High School Degree',"Percent Less than Bachelor's Degree"]]
.describe()
Republican_Summary = df[df.Party == 0][['Percent White, not Hisp
anic or Latino','Percent Black, not Hispanic or Latino','Percent
Hispanic or Latino','Percent Foreign Born','Percent Female','Per
cent Age 29 and Under','Percent Age 65 and Older','Percent Less
than High School Degree',"Percent Less than Bachelor's Degree"]]
.describe()
```

In [13]:

```
print("Democratic Summary:\n ", Democratic_Summary)
```

```
Democratic Summary:
          Percent White, not Hispanic or Latino  \
count                                325.000000
mean                                  69.683766
std                                   24.981502
min                                    2.776702
25%                                   53.271579
50%                                   77.786090
75%                                   90.300749
max                                   98.063495

       Percent Black, not Hispanic or Latino  Percen
t Hispanic or Latino  \
count                                325.000000
325.000000
mean                                   9.242649
12.587391
std                                   13.351340
19.575030
min                                    0.000000
0.193349
25%                                    0.839103
2.531017
50%                                    3.485992
5.039747
```

```
75%                                11.058843
11.857116
max                                63.953279
95.479801

       Percent Foreign Born  Percent Female  Percent
Age 29 and Under  \
count            325.000000      325.000000
325.000000
mean               7.986330       50.385433
38.726959
std                8.330740        2.149359
6.252786
min                0.179769       34.245291
23.156452
25%                2.470508       49.854280
34.488444
50%                5.105490       50.653830
38.074151
75%               10.144555       51.492075
42.161162
max               52.229868       56.418468
67.367823

       Percent Age 65 and Older  Percent Less than H
igh School Degree  \
count                325.000000
325.000000
mean                  16.194826
11.883760
std                    4.282422
6.505613
min                    6.653188
3.215803
25%                   13.106233
7.893714
50%                   15.698087
10.370080
75%                   18.806426
13.637059
max                   31.642106
49.673777

       Percent Less than Bachelor's Degree
count                           325.000000
```

```
mean                             71.968225
std                              11.192404
min                              26.335440
25%                              65.711800
50%                              72.736143
75%                              79.903653
max                              94.849957
```

In [14]:

```
print("Republican Summary:\n ", Republican_Summary)
```

```
Republican Summary:
         Percent White, not Hispanic or Latino  \
count                                 862.000000
mean                                   82.623951
std                                    15.969406
min                                    18.758977
25%                                    75.016397
50%                                    89.351430
75%                                    94.435931
max                                    98.743894

       Percent Black, not Hispanic or Latino  Percen
t Hispanic or Latino  \
count                                 862.000000
862.000000
mean                                    4.228121
9.721479
std                                     6.740653
13.934183
min                                     0.000000
0.013791
25%                                     0.471147
1.715916
50%                                     1.335736
3.447823
75%                                     4.926921
10.709696
max                                    41.563041
78.397012

       Percent Foreign Born  Percent Female  Percent
Age 29 and Under  \
count            862.000000      862.000000
```

```
862.000000
mean               3.982130       49.625933
36.060964
std                4.452447        2.425508
5.079622
min                0.019249       21.513413
19.565830
25%                1.334304       49.235072
33.051646
50%                2.344084       50.179023
35.864703
75%                5.149429       50.827195
38.539787
max               37.058317       55.885023
58.749116

       Percent Age 65 and Older  Percent Less than H
igh School Degree  \
count                862.000000
862.000000
mean                  18.780542
14.002060
std                    4.696865
6.213336
min                    6.954387
2.134454
25%                   15.781262
9.692476
50%                   18.355587
12.572435
75%                   21.081440
17.447168
max                   37.622759
47.812773

       Percent Less than Bachelor's Degree
count                           862.000000
mean                             81.084613
std                               6.809488
min                              43.419470
25%                              78.134387
50%                              82.406700
75%                              85.546272
max                              93.602862
```

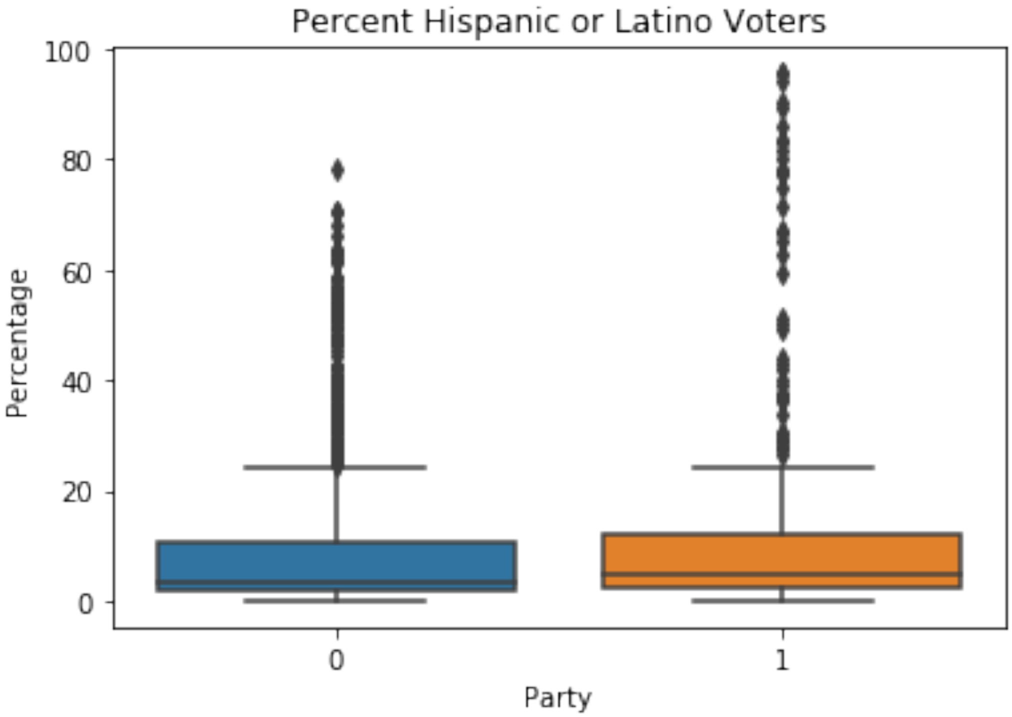
Task 09: Based on your previous analysis, which variables in the dataset do you think are more important to determine whether a county is labeled as Democratic or Republican? Justify your answer

In [15]:

```
ax = sns.boxplot(x = 'Party', y = 'Percent Hispanic or Latino', data = df)
ax.set(title = 'Percent Hispanic or Latino Voters', xlabel = 'Party', ylabel = 'Percentage')
```

Out[15]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Hispanic or Latino Voters')
]
```

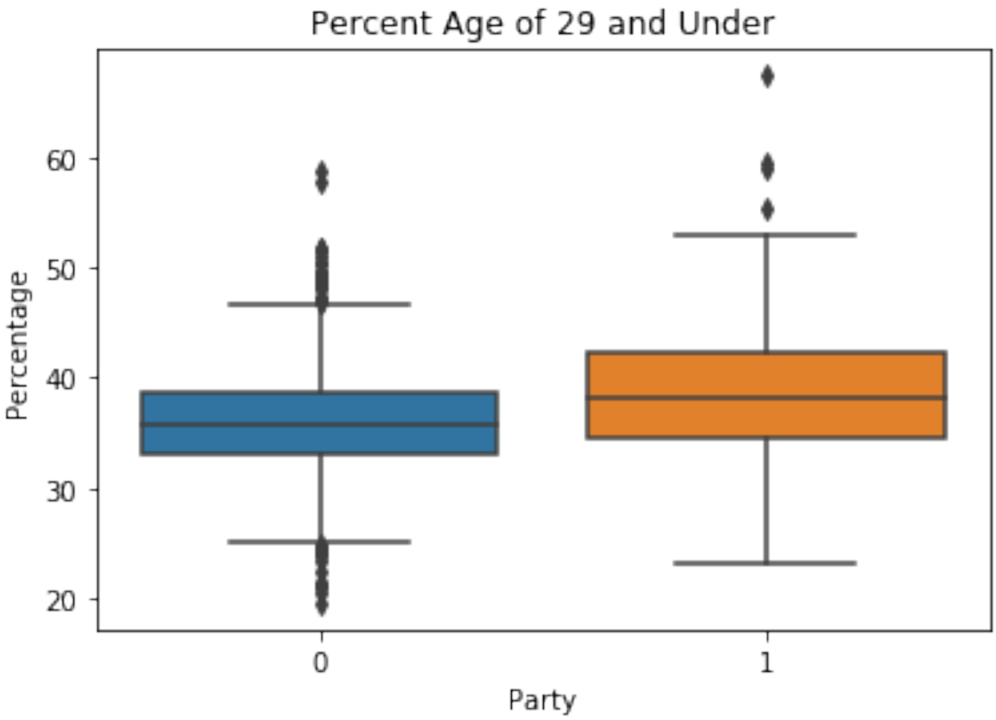


In [16]:

```
ax = sns.boxplot(x = 'Party', y = 'Percent Age 29 and Under', da
ta = df)
ax.set(title = 'Percent Age of 29 and Under', xlabel = 'Party',
ylabel = 'Percentage')
```

Out[16]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Age of 29 and Under')]
```

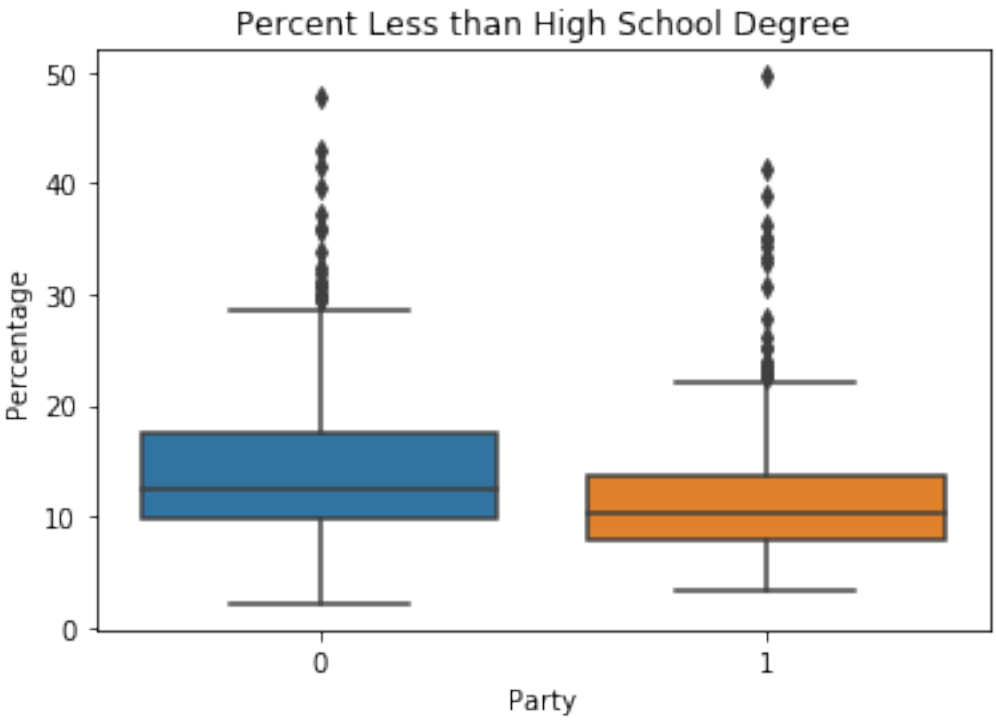


In [17]:

```
ax = sns.boxplot(x = 'Party', y = 'Percent Less than High School
Degree', data = df)
ax.set(title = 'Percent Less than High School Degree', xlabel =
'Party', ylabel = 'Percentage')
```

Out[17]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Less than High School Degre
e')]
```

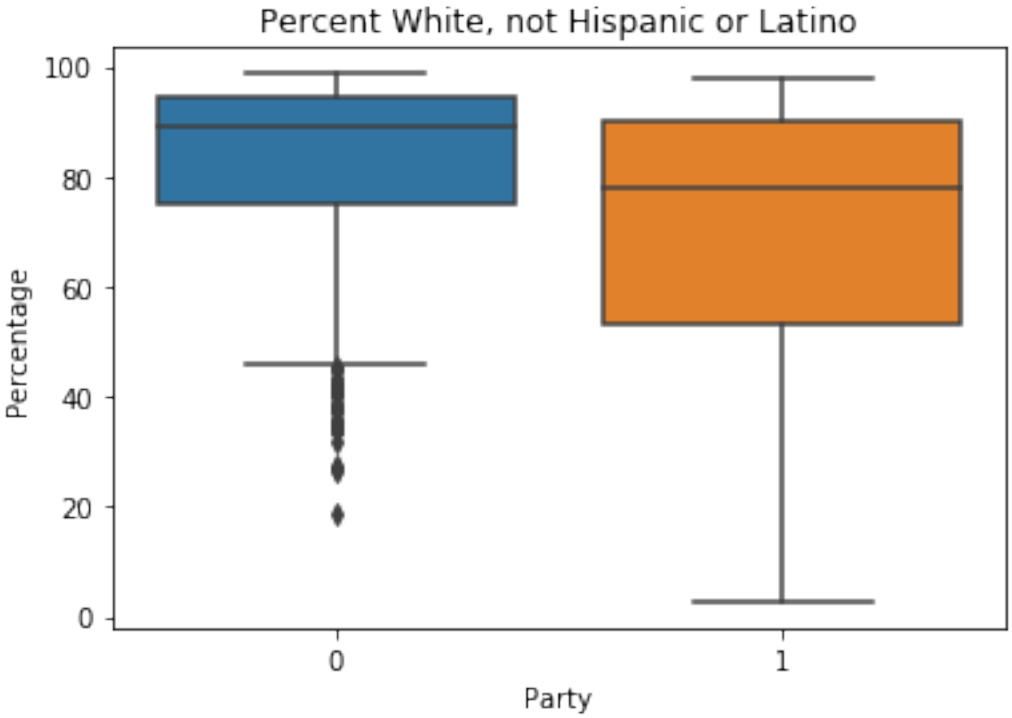


In [18]:

```
ax = sns.boxplot(x = 'Party', y = 'Percent White, not Hispanic o
r Latino', data = df)
ax.set(title = 'Percent White, not Hispanic or Latino', xlabel =
'Party', ylabel = 'Percentage')
```

Out[18]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent White, not Hispanic or Lati
no')]
```

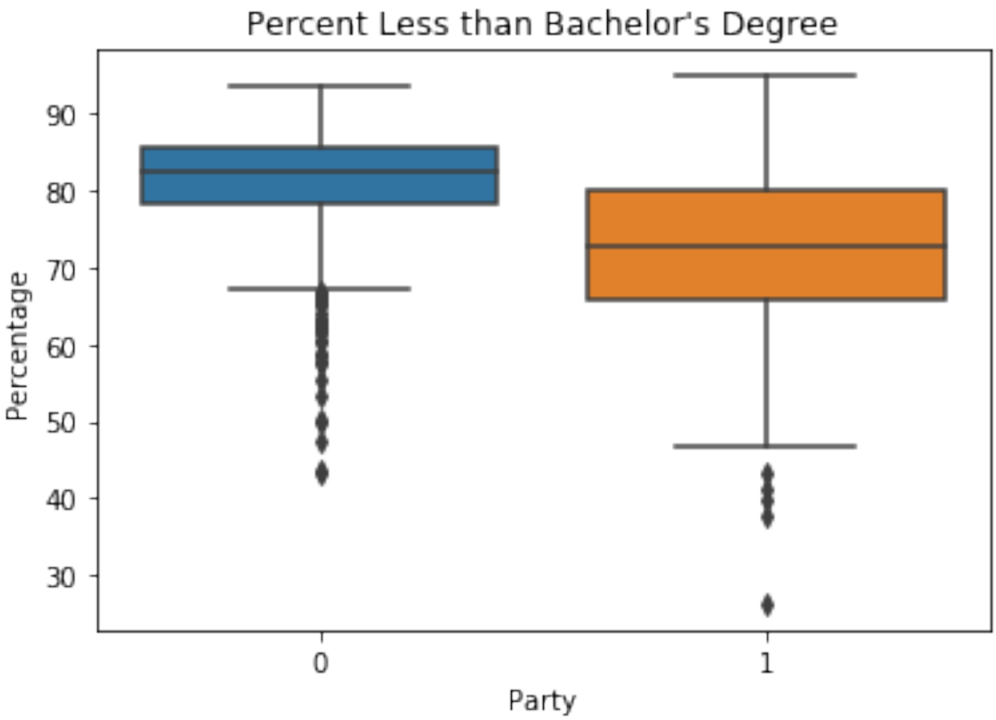


```
In [19]:
```

```
ax = sns.boxplot(x = 'Party', y = "Percent Less than Bachelor's
Degree", data = df)
ax.set(title = "Percent Less than Bachelor's Degree", xlabel = '
Party', ylabel = 'Percentage')
```

```
Out[19]:
```

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, "Percent Less than Bachelor's Degree
")]
```

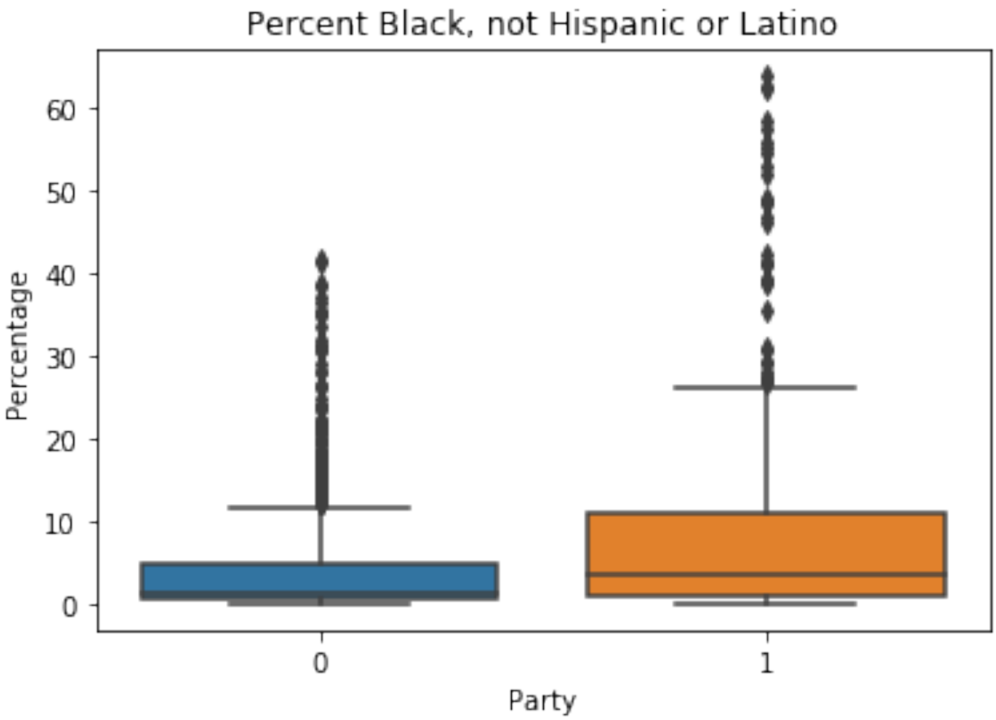


In [20]:

```
ax = sns.boxplot(x = 'Party', y = 'Percent Black, not Hispanic or Latino', data = df)
ax.set(title = 'Percent Black, not Hispanic or Latino', xlabel = 'Party', ylabel = 'Percentage')
```

Out[20]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Black, not Hispanic or Lati
no')]
```

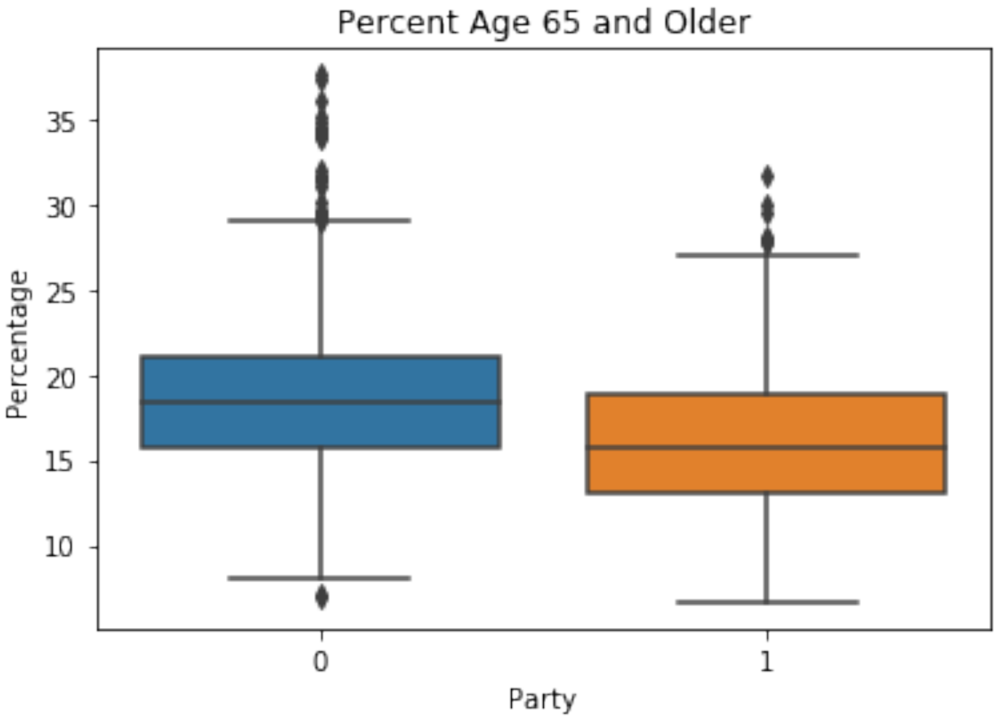


In [21]:

```
ax = sns.boxplot(x = 'Party', y = 'Percent Age 65 and Older', da
ta = df)
ax.set(title = 'Percent Age 65 and Older', xlabel = 'Party', yla
bel = 'Percentage')
```

Out[21]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Age 65 and Older')]
```

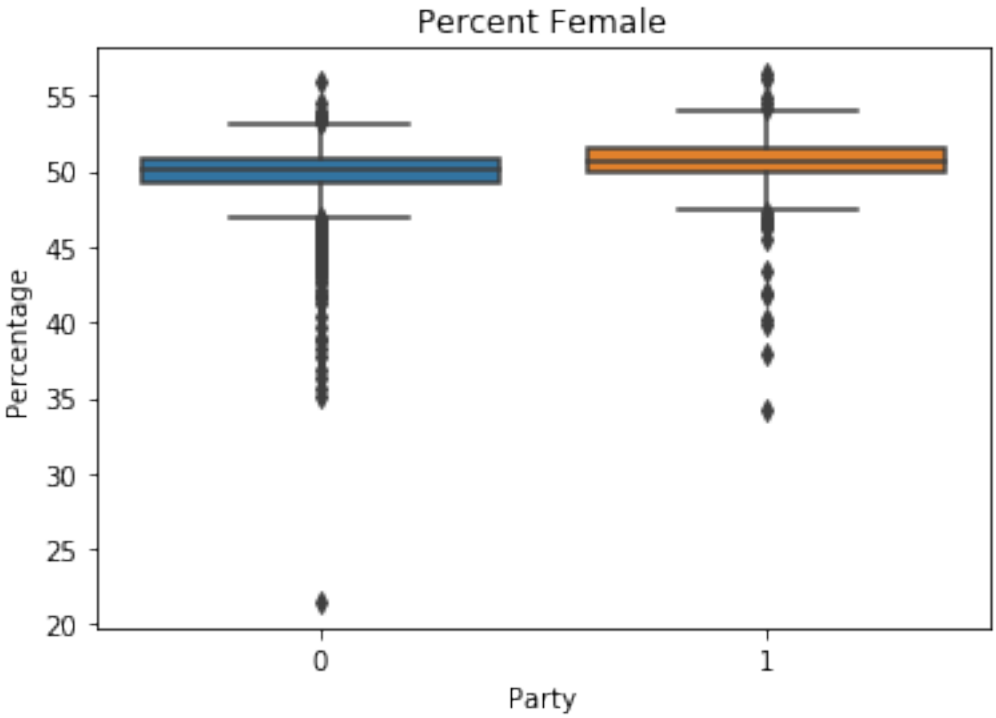


In [22]:

```
ax = sns.boxplot(x = 'Party', y = 'Percent Female', data = df)
ax.set(title = 'Percent Female', xlabel = 'Party', ylabel = 'Per
centage')
```

Out[22]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Female')]
```

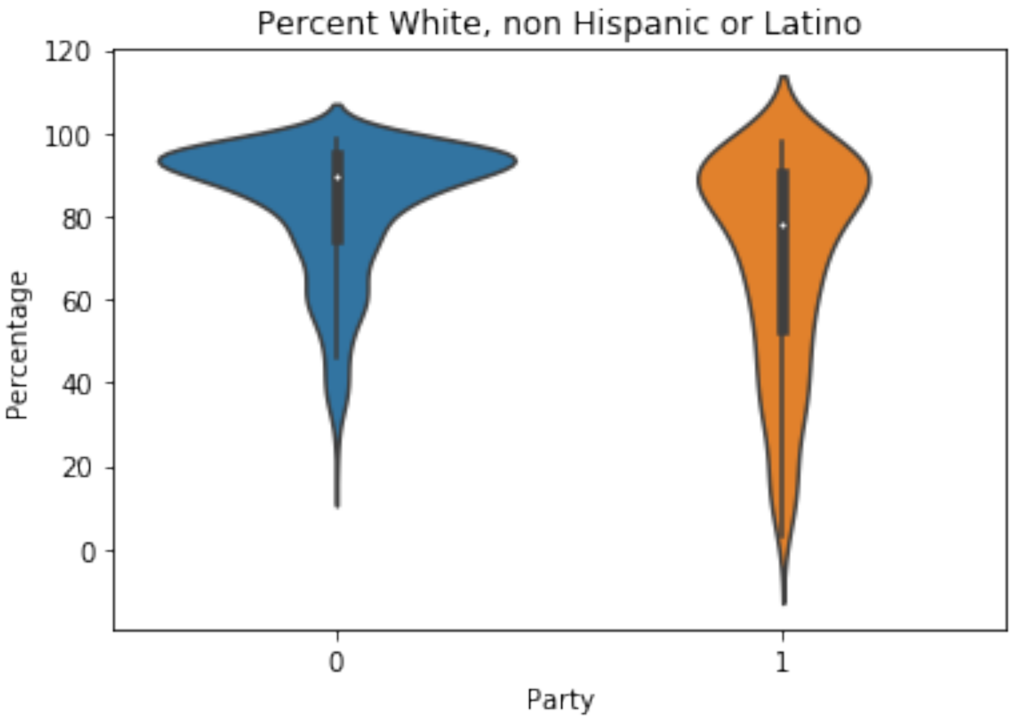


In [23]:

```python
# Violin plot for Percent White
ax = sns.violinplot(x = 'Party', y = 'Percent White, not Hispanic or Latino', data = df)
ax.set(title = 'Percent White, non Hispanic or Latino', xlabel = 'Party', ylabel = 'Percentage')
```

Out[23]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent White, non Hispanic or Latino')]
```

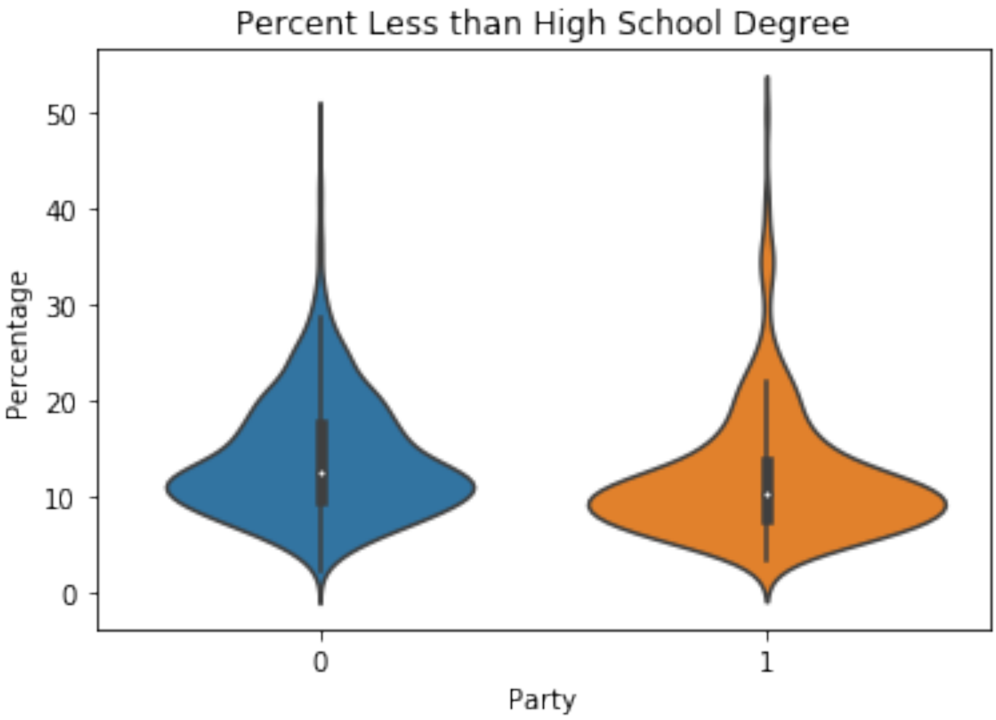


In [24]:

```
# Violin plot for Percent Less than High School
ax = sns.violinplot(x = 'Party', y = 'Percent Less than High Sch
ool Degree', data = df)
ax.set(title = 'Percent Less than High School Degree', xlabel =
'Party', ylabel = 'Percentage')
```

Out[24]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Less than High School Degre
e')]
```

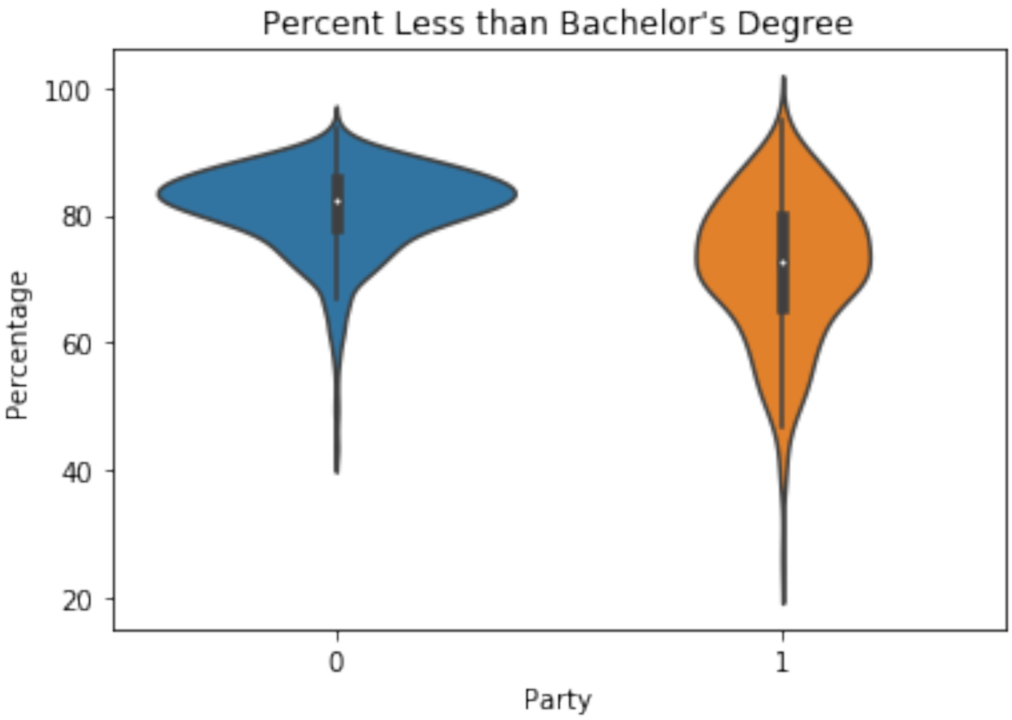


In [25]:

```
# Violin plot for Percent Less than Bachelor's
ax = sns.violinplot(x = 'Party', y = "Percent Less than Bachelor's Degree", data = df)
ax.set(title = "Percent Less than Bachelor's Degree", xlabel = 'Party', ylabel = "Percentage")
```

Out[25]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, "Percent Less than Bachelor's Degree
")]
```

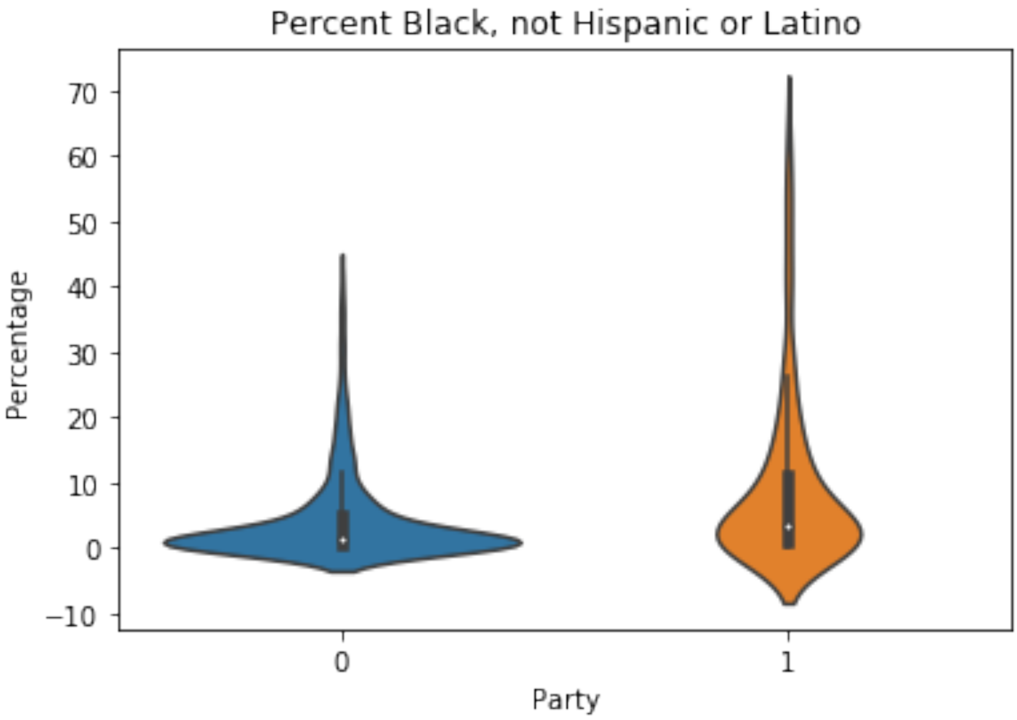


In [26]:

```
# Violin plot for Percent Black
ax = sns.violinplot(x = 'Party', y = 'Percent Black, not Hispani
c or Latino', data = df)
ax.set(title = 'Percent Black, not Hispanic or Latino', xlabel =
'Party', ylabel = 'Percentage')
```

Out[26]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Black, not Hispanic or Lati
no')]
```

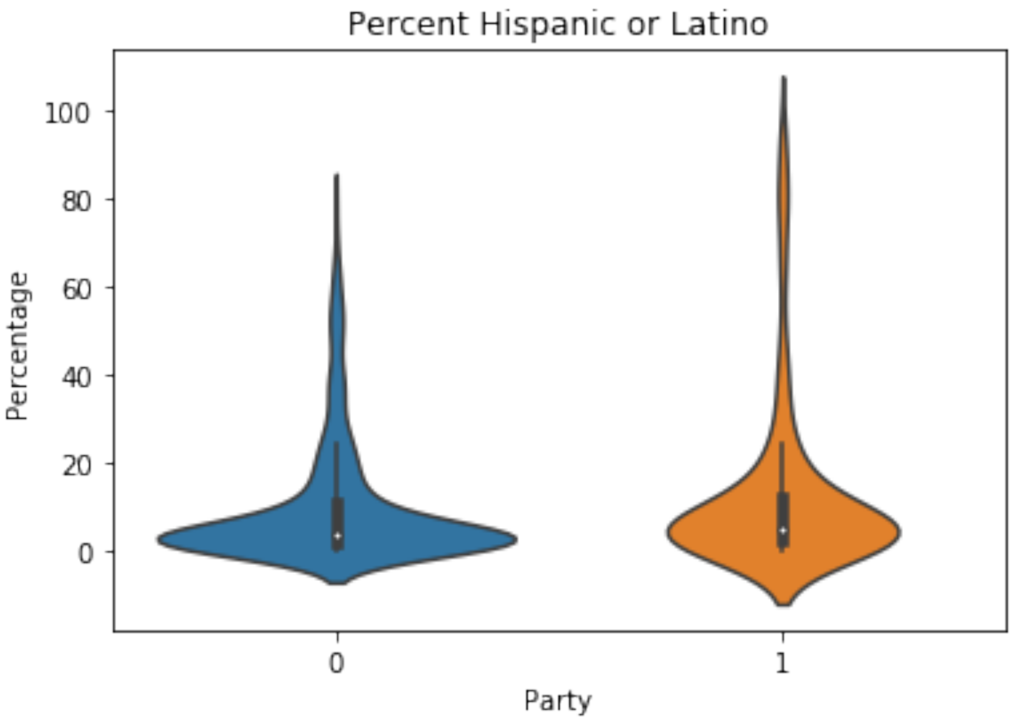


In [27]:

```
# Violin plot for Percent Hispanic or Latino
ax = sns.violinplot(x = 'Party', y = 'Percent Hispanic or Latino', data = df)
ax.set(title = 'Percent Hispanic or Latino', xlabel = 'Party', ylabel = 'Percentage')
```

Out[27]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Hispanic or Latino')]
```

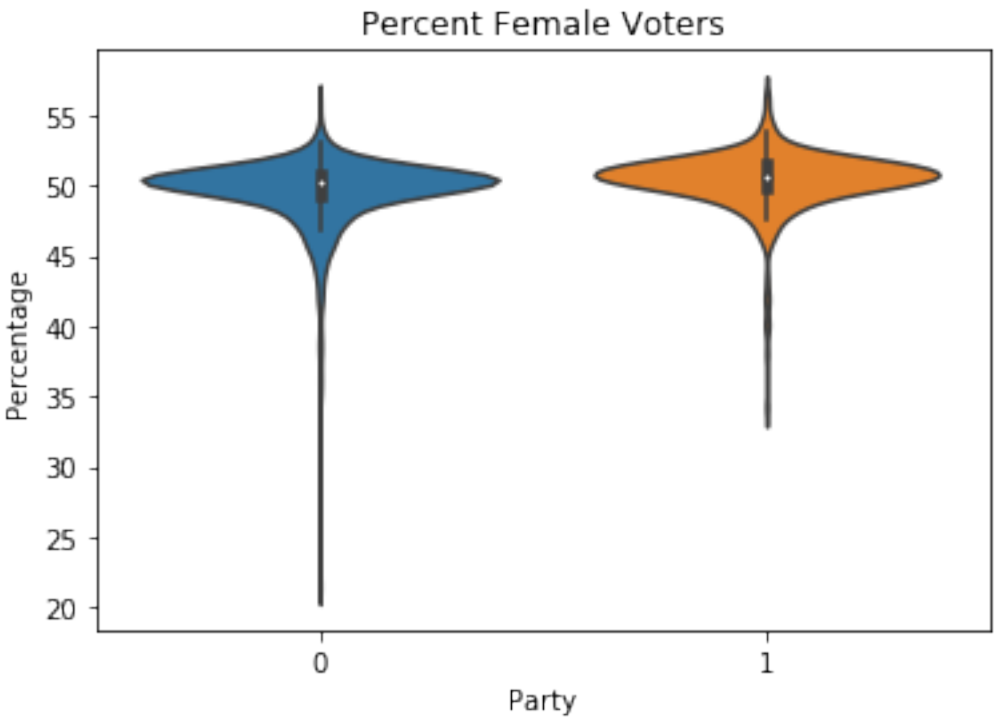


In [28]:

```
# Violin plot for Percent Female
ax = sns.violinplot(x = 'Party', y = 'Percent Female', data = df
)
ax.set(title = 'Percent Female Voters', xlabel = 'Party', ylabel
= 'Percentage')
```

Out[28]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Female Voters')]
```

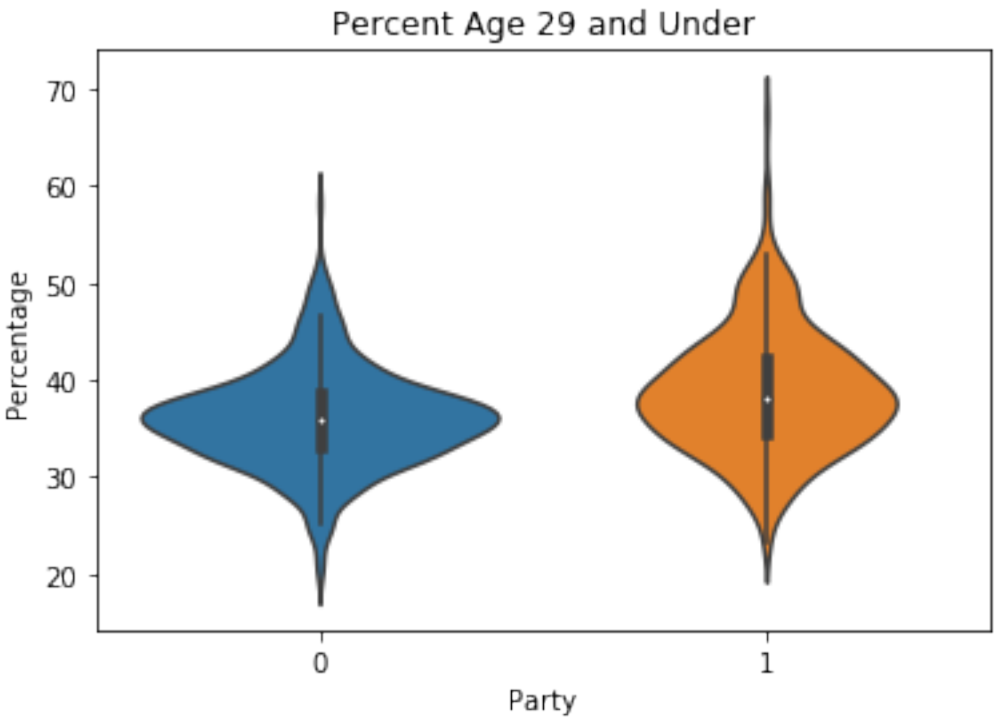


In [29]:

```
# Violin plot for Age 29 and Under
ax = sns.violinplot(x = 'Party', y = 'Percent Age 29 and Under',
data = df)
ax.set(title = 'Percent Age 29 and Under', xlabel = 'Party', yla
bel = 'Percentage')
```

Out[29]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Age 29 and Under')]
```

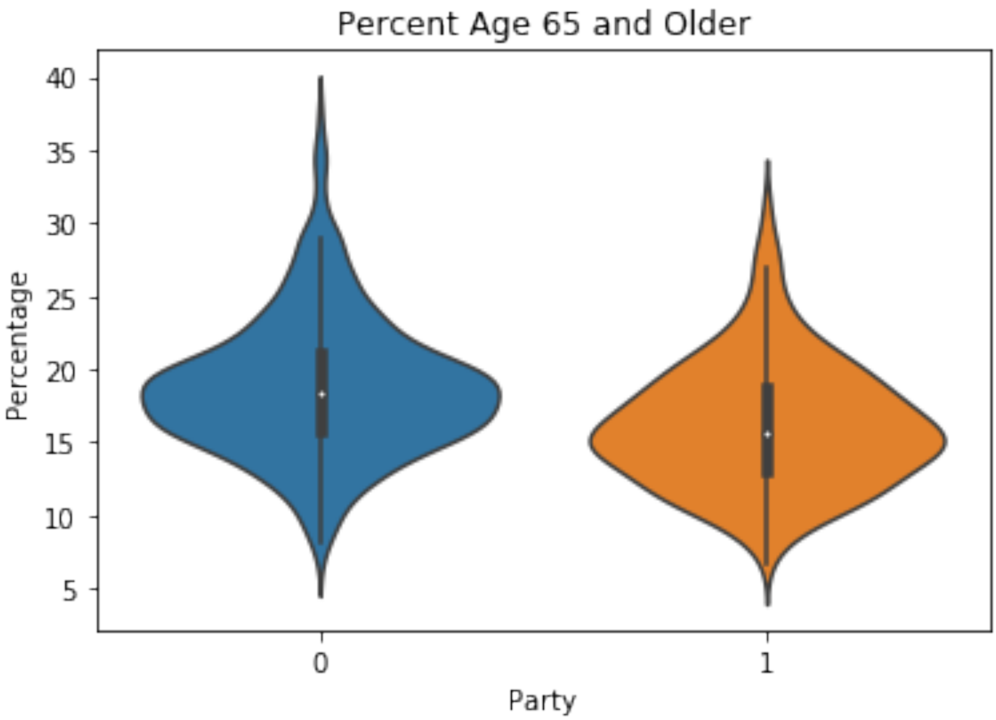


In [30]:

```
# Violin plot for Age 65 and Older
ax = sns.violinplot(x = 'Party', y = 'Percent Age 65 and Older', data = df)
ax.set(title = 'Percent Age 65 and Older', xlabel = 'Party', ylabel = 'Percentage')
```

Out[30]:

```
[Text(0, 0.5, 'Percentage'),
 Text(0.5, 0, 'Party'),
 Text(0.5, 1.0, 'Percent Age 65 and Older')]
```



Task 10: 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 [31]:

```
import plotly.figure_factory as ff
values = range(len(df['FIPS']))
newDF = df
change_values = {1:'Democratic', 0:'Republican'}
newDF['Party'] = newDF['Party'].map(change_values)
```

In [32]:

```
fig = ff.create_choropleth(fips=newDF['FIPS'], values= newDF['Party'],colorscale = ['rgb(0, 0, 255)','rgb(255,0,0)'],
                           county_outline={'width': 0.5},legend_title='Party by County',
                           title='Counties by FIPS codes')
fig.layout.template = None
fig.show()
```

```
/Users/kirunhaque/anaconda3/lib/python3.7/site-packages/pandas/core/frame.py:6692: FutureWarning:

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

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

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

Counties by