In [2]:
```
### Project1

import pandas as pd
import numpy as np
from scipy.stats import ttest_ind
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
## Task1

data_election = pd.read_csv("election_train.csv")
print(data_election)
```

```
      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
...    ...   ...                ...         ...         ...      ...
2405  2018    WY  Sweetwater County  US Senator  Republican   8577.0
2406  2018    WY       Uinta County  US Senator  Democratic   1371.0
2407  2018    WY       Uinta County  US Senator  Republican   4713.0
2408  2018    WY    Washakie County  US Senator  Democratic    588.0
2409  2018    WY    Washakie County  US Senator  Republican   2423.0

[2410 rows x 6 columns]
```

In [4]:
```
## Task1

data_election = pd.pivot_table(data_election, index=['State','County','Year','Office'], columns=['Party'], values=['Votes'], aggfunc=np.sum)
data_election = data_election.reset_index()

print(data_election)
```

```
      State             County  Year      Office      Votes
Party                                          Democratic Republican
0        AZ      Apache County  2018  US Senator    16298.0     7810.0
1        AZ     Cochise County  2018  US Senator    17383.0    26929.0
2        AZ    Coconino County  2018  US Senator    34240.0    19249.0
3        AZ        Gila County  2018  US Senator     7643.0    12180.0
4        AZ      Graham County  2018  US Senator     3368.0     6870.0
...     ...                ...   ...         ...        ...        ...
1200     WY      Platte County  2018  US Senator      801.0     2850.0
1201     WY    Sublette County  2018  US Senator      668.0     2653.0
1202     WY  Sweetwater County  2018  US Senator     3943.0     8577.0
1203     WY       Uinta County  2018  US Senator     1371.0     4713.0
1204     WY    Washakie County  2018  US Senator      588.0     2423.0

[1205 rows x 6 columns]
```

In [5]:
```
## Task2

data_election['County'] = data_election['County'].replace({' County':''
}, regex=True)
print(data_election)
```

```
      State      County  Year      Office      Votes
Party                                     Democratic Republican
0        AZ      Apache  2018  US Senator     16298.0     7810.0
1        AZ     Cochise  2018  US Senator     17383.0    26929.0
2        AZ    Coconino  2018  US Senator     34240.0    19249.0
3        AZ        Gila  2018  US Senator      7643.0    12180.0
4        AZ      Graham  2018  US Senator      3368.0     6870.0
...     ...         ...   ...         ...         ...        ...
1200     WY      Platte  2018  US Senator       801.0     2850.0
1201     WY    Sublette  2018  US Senator       668.0     2653.0
1202     WY  Sweetwater  2018  US Senator      3943.0     8577.0
1203     WY       Uinta  2018  US Senator      1371.0     4713.0
1204     WY    Washakie  2018  US Senator       588.0     2423.0

[1205 rows x 6 columns]
```

In [6]:
```
## Task2

data_demographics = pd.read_csv("demographics_train.csv")
print(data_demographics)
```

```
          State      County   FIPS  Total Population  \
0     Wisconsin   La Crosse  55063            117538
1      Virginia   Alleghany  51005             15919
2       Indiana    Fountain  18045             16741
3          Ohio      Geauga  39055             94020
4     Wisconsin     Jackson  55053             20566
...         ...         ...    ...               ...
1211    Montana     Lincoln  30053             19268
1212       Ohio  Tuscarawas  39157             92579
1213   Michigan     Newaygo  26123             47957
1214  Tennessee  Lauderdale  47097             27261
1215      Texas      Sabine  48403             10367

      Citizen Voting-Age Population  Percent White, not Hispanic or Lat
ino  \
0                                 0                                90.537
528
1                             12705                                91.940
449
2                             12750                                95.705
155
3                                 0                                95.837
056
4                             15835                                86.662
453
...                             ...
...
1211                          15640                                93.351
671
1212                          70485                                95.155
489
1213                              0                                90.716
684
1214                              0                                60.456
330
1215                              0                                86.341
275

      Percent Black, not Hispanic or Latino  Percent Hispanic or Latino
\
0                                  1.214075                    1.724549
1                                  5.207614                    1.432251
2                                  0.400215                    2.359477
3                                  1.256116                    1.294405
4                                  1.983857                    3.082758
...                                     ...                         ...
1211                               0.057089                    2.678015
1212                               0.804718                    2.349345
1213                               1.317847                    5.728048
1214                              34.789626                    2.380690
1215                               7.080158                    3.839105

      Percent Foreign Born  Percent Female  Percent Age 29 and Under  \
0                 2.976059       51.171536                 43.241335
1                 1.300333       51.077329                 31.660280
2                 1.547100       49.770026                 35.899887
3                 2.578175       50.678579                 36.281642
```

```
4                 1.376058       46.649810                 36.292911
...                    ...             ...                       ...
1211              2.345858       49.974050                 27.979033
1212              1.650482       50.823621                 36.475875
1213              1.978856       49.656984                 36.620306
1214              1.757089       47.734859                 39.528997
1215              0.752387       50.506415                 31.243368

      Percent Age 65 and Older  Median Household Income  Percent Unempl
oyed  \
0                    14.702479                    51477            4.79
6952
1                    23.902255                    45538            4.56
0986
2                    18.941521                    45924            7.97
8789
3                    18.028079                    74165            4.03
6902
4                    17.587280                    49608            5.56
9698
...                        ...                      ...
...
1211                 24.828732                    35461           11.57
0571
1212                 17.836658                    46992            5.73
4798
1213                 17.728382                    44049            8.88
5564
1214                 13.891640                    32353           11.42
7738
1215                 28.995852                    32500            8.25
9134

      Percent Less than High School Degree  \
0                                 5.474767
1                                15.537543
2                                12.032155
3                                 8.928599
4                                11.792912
...                                    ...
1211                             11.891078
1212                             14.068037
1213                             14.194114
1214                             24.485474
1215                             14.643237

      Percent Less than Bachelor's Degree  Percent Rural
0                                67.529757      16.827753
1                                83.711604      52.393846
2                                85.538940      65.951276
3                                62.730824      63.968990
4                                86.129256      72.238251
...                                    ...            ...
1211                             81.041325      79.793773
1212                             85.020218      41.580437
1213                             85.660708      83.840281
1214                             91.425468      58.662592
```

```
1215                            87.060703     100.000000

[1216 rows x 17 columns]
```

In [7]:
```
## Task2

us_state_dictionary = {
    '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',
}

data_demographics = data_demographics.replace({'State':us_state_dictiona
ry})
print(data_demographics)
```

	State	County	FIPS	Total Population
0	WI	La Crosse	55063	117538
1	VA	Alleghany	51005	15919
2	IN	Fountain	18045	16741
3	OH	Geauga	39055	94020
4	WI	Jackson	55053	20566
...	...	...	...	...
1211	MT	Lincoln	30053	19268
1212	OH	Tuscarawas	39157	92579
1213	MI	Newaygo	26123	47957
1214	TN	Lauderdale	47097	27261
1215	TX	Sabine	48403	10367

	Citizen Voting-Age Population	Percent White, not Hispanic or Latino
0	0	90.537528
1	12705	91.940449
2	12750	95.705155
3	0	95.837056
4	15835	86.662453
...	...	...
1211	15640	93.351671
1212	70485	95.155489
1213	0	90.716684
1214	0	60.456330
1215	0	86.341275

	Percent Black, not Hispanic or Latino	Percent Hispanic or Latino
0	1.214075	1.724549
1	5.207614	1.432251
2	0.400215	2.359477
3	1.256116	1.294405
4	1.983857	3.082758
...	...	...
1211	0.057089	2.678015
1212	0.804718	2.349345
1213	1.317847	5.728048
1214	34.789626	2.380690
1215	7.080158	3.839105

	Percent Foreign Born	Percent Female	Percent Age 29 and Under
0	2.976059	51.171536	43.241335
1	1.300333	51.077329	31.660280
2	1.547100	49.770026	35.899887
3	2.578175	50.678579	36.281642

```
4                 1.376058       46.649810                 36.292911
...                    ...             ...                       ...
1211              2.345858       49.974050                 27.979033
1212              1.650482       50.823621                 36.475875
1213              1.978856       49.656984                 36.620306
1214              1.757089       47.734859                 39.528997
1215              0.752387       50.506415                 31.243368

      Percent Age 65 and Older  Median Household Income  Percent Unempl
oyed  \
0                    14.702479                    51477            4.79
6952
1                    23.902255                    45538            4.56
0986
2                    18.941521                    45924            7.97
8789
3                    18.028079                    74165            4.03
6902
4                    17.587280                    49608            5.56
9698
...                        ...                      ...
...
1211                 24.828732                    35461           11.57
0571
1212                 17.836658                    46992            5.73
4798
1213                 17.728382                    44049            8.88
5564
1214                 13.891640                    32353           11.42
7738
1215                 28.995852                    32500            8.25
9134

      Percent Less than High School Degree  \
0                                 5.474767
1                                15.537543
2                                12.032155
3                                 8.928599
4                                11.792912
...                                    ...
1211                             11.891078
1212                             14.068037
1213                             14.194114
1214                             24.485474
1215                             14.643237

      Percent Less than Bachelor's Degree  Percent Rural
0                                67.529757      16.827753
1                                83.711604      52.393846
2                                85.538940      65.951276
3                                62.730824      63.968990
4                                86.129256      72.238251
...                                    ...            ...
1211                             81.041325      79.793773
1212                             85.020218      41.580437
1213                             85.660708      83.840281
1214                             91.425468      58.662592
```

```
1215                            87.060703     100.000000

[1216 rows x 17 columns]
```

In [8]:
```
## Task2

# Convert to lower and capitalize for County
data_election['County'] = data_election['County'].str.lower()
data_demographics['County'] = data_demographics['County'].str.lower()

# merge action
data_merged=pd.merge(data_election, data_demographics, on=['State','Coun
ty'], how='inner')
print(data_merged)
```

```
     State      County (State, )  (County, )  (Year, )  (Office, )  \
0       AZ      apache        AZ      apache      2018  US Senator
1       AZ     cochise        AZ     cochise      2018  US Senator
2       AZ    coconino        AZ    coconino      2018  US Senator
3       AZ        gila        AZ        gila      2018  US Senator
4       AZ      graham        AZ      graham      2018  US Senator
...    ...         ...       ...         ...       ...         ...
1195    WY      platte        WY      platte      2018  US Senator
1196    WY    sublette        WY    sublette      2018  US Senator
1197    WY  sweetwater        WY  sweetwater      2018  US Senator
1198    WY       uinta        WY       uinta      2018  US Senator
1199    WY    washakie        WY    washakie      2018  US Senator

      (Votes, Democratic)  (Votes, Republican)   FIPS  Total Population
...  \
0                 16298.0               7810.0   4001             72346
...
1                 17383.0              26929.0   4003            128177
...
2                 34240.0              19249.0   4005            138064
...
3                  7643.0              12180.0   4007             53179
...
4                  3368.0               6870.0   4009             37529
...
...                   ...                  ...    ...               ...
...
1195                801.0               2850.0  56031              8740
...
1196                668.0               2653.0  56035             10032
...
1197               3943.0               8577.0  56037             44812
...
1198               1371.0               4713.0  56041             20893
...
1199                588.0               2423.0  56043              8351
...

      Percent Hispanic or Latino  Percent Foreign Born  Percent Female
\
0                       5.947806             1.719515       50.598513
1                      34.403208            11.458374       49.069646
2                      13.711033             4.825298       50.581614
3                      18.548675             4.249798       50.296170
4                      32.097844             4.385942       46.313518
...                          ...                  ...             ...
1195                    7.814645             2.780320       47.711670
1196                    7.814992             2.053429       46.949761
1197                   15.859591             5.509685       47.824244
1198                    8.959939             3.986981       49.327526
1199                   13.962400             3.783978       51.359119

      Percent Age 29 and Under  Percent Age 65 and Older  \
0                    45.854643                 13.322091
1                    37.902276                 19.756275
2                    48.946141                 10.873943
3                    32.238290                 26.397638
```

```
4                    46.393456                 12.315809
...                        ...                       ...
1195                 32.700229                 22.013730
1196                 36.393541                 13.337321
1197                 44.153352                  9.417120
1198                 43.205858                 10.678218
1199                 34.774279                 19.650341

      Median Household Income  Percent Unemployed  \
0                       32460           15.807433
1                       45383            8.567108
2                       51106            8.238305
3                       40593           12.129932
4                       47422           14.424104
...                       ...                 ...
1195                    41051            3.901047
1196                    76004            2.786971
1197                    68233            5.072255
1198                    53323            6.390755
1199                    46212            7.441860

      Percent Less than High School Degree  \
0                                 21.758252
1                                 13.409171
2                                 11.085381
3                                 15.729958
4                                 14.580797
...                                     ...
1195                               9.675889
1196                               4.658830
1197                               9.314606
1198                              10.361224
1199                              12.577108

      Percent Less than Bachelor's Degree  Percent Rural
0                               88.941063      74.061076
1                               76.837055      36.301067
2                               65.791439      31.466066
3                               82.262624      41.062000
4                               86.675944      46.437399
...                                   ...            ...
1195                            80.300395      58.647744
1196                            75.645069     100.000000
1197                            78.628507      10.916313
1198                            81.793082      43.095937
1199                            78.923920      35.954529

[1200 rows x 23 columns]

//anaconda3/envs/cs418/lib/python3.7/site-packages/pandas/core/reshape/
merge.py:617: UserWarning:

merging between different levels can give an unintended result (2 level
s on the left, 1 on the right)
```

In [8]:
```
# Task2: verify with csv file (optional)

data_merged.to_csv('p1_t2_data_merged.csv')
```

In [9]:
```
## Task3

# Exploring the merged data set

# In the  table we can see that we have 23 variables in data frame now.
 Types of the variables are also given in the table
data_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1200 entries, 0 to 1199
Data columns (total 23 columns):
State                                   1200 non-null object
County                                  1200 non-null object
(State, )                               1200 non-null object
(County, )                              1200 non-null object
(Year, )                                1200 non-null int64
(Office, )                              1200 non-null object
(Votes, Democratic)                     1200 non-null float64
(Votes, 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(5)
memory usage: 225.0+ KB
```

In [10]:
```
## Task3

# exploring columns
data_merged.columns
```

Out[10]:
```
Index([                                  'State',
                                        'County',
                                  ('State', ''),
                                 ('County', ''),
                                   ('Year', ''),
                                 ('Office', ''),
                          ('Votes', 'Democratic'),
                          ('Votes', 'Republican'),
                                          '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',
                        'Percent Unemployed',
        'Percent Less than High School Degree',
         'Percent Less than Bachelor's Degree',
                                'Percent Rural'],
      dtype='object')
```

In [11]:
```
## Task3

# Removing irrelevant or redundant variables
data_merged = data_merged.drop([data_merged.columns[2],
                                data_merged.columns[3],
                                data_merged.columns[4],
                                data_merged.columns[5]], axis = 1 )
```

In [12]:
```
## Task3

# Rename and organize the column names

data_merged.columns = ['State',
                       'County',
                       'Votes_Democratic',
                       'Votes_Republican',
                       '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',
                       'Percent_Unemployed',
                       'Percent_Less_than_High_School_Degree',
                       'Percent_Less_than_Bachelors_Degree',
                       'Percent_Rural']
```

In [16]:
```
# Task3 verification (optional)

data_merged.to_csv('p1_t3_data_dropped_variables.csv')
```

In [13]:
```
data_merged.head()
```

Out[13]:

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Voting-Age_Population
0	AZ	apache	16298.0	7810.0	4001	72346	0
1	AZ	cochise	17383.0	26929.0	4003	128177	92915
2	AZ	coconino	34240.0	19249.0	4005	138064	104265
3	AZ	gila	7643.0	12180.0	4007	53179	0
4	AZ	graham	3368.0	6870.0	4009	37529	0

In [14]:
```
## Task 4

# Find any explicit missing values
data_merged.isnull().sum()

# Like the below result, we can see that there is no explicit missing va
lue such as null.
```

Out[14]:
```
State                                   0
County                                  0
Votes_Democratic                        0
Votes_Republican                        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_Bachelors_Degree      0
Percent_Rural                           0
dtype: int64
```

In [15]:
```
## Task4

pd.isna(data_merged).sum()
```

Out[15]:
```
State                                   0
County                                  0
Votes_Democratic                        0
Votes_Republican                        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_Bachelors_Degree      0
Percent_Rural                           0
dtype: int64
```

In [16]:
```
## Task4

# To find the sum of three demographics variables
data_merged['Sum_Percent_Races'] = data_merged.iloc[:,7:10].sum(axis=1)

data_merged.head()
```

Out[16]:

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Voting-Age_Population
0	AZ	apache	16298.0	7810.0	4001	72346	0
1	AZ	cochise	17383.0	26929.0	4003	128177	92915
2	AZ	coconino	34240.0	19249.0	4005	138064	104265
3	AZ	gila	7643.0	12180.0	4007	53179	0
4	AZ	graham	3368.0	6870.0	4009	37529	0

In [17]:
```
## Task4

# The Sum_Percent_Races shows that the sum percentage of the three races
reaches to 100% even if one of the races is 0%.

data_merged[
    (data_merged.Percent_White_not_Hispanic_or_Latino == 0) |
    (data_merged.Percent_Black_not_Hispanic_or_Latino == 0) |
    (data_merged.Percent_Hispanic_or_Latino == 0)]
```

Out[17]:

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Vo Age_Popul
315	MT	carter	128.0	602.0	30011	1295	
318	MT	daniels	281.0	631.0	30019	1787	
326	MT	golden valley	130.0	303.0	30037	730	
329	MT	judith basin	388.0	752.0	30045	1981	
330	MT	liberty	365.0	586.0	30051	2292	
332	MT	mccone	227.0	773.0	30055	1678	
333	MT	meagher	319.0	629.0	30059	1960	
337	MT	phillips	577.0	1426.0	30071	4150	
339	MT	powder river	203.0	748.0	30075	1648	
341	MT	prairie	177.0	450.0	30079	1414	
352	MT	wheatland	315.0	586.0	30107	2109	
353	MT	wibaux	140.0	390.0	30109	1143	
356	ND	benson	1427.0	828.0	38005	6802	
362	ND	eddy	555.0	675.0	38027	2370	
363	ND	emmons	391.0	1433.0	38029	3426	
365	ND	golden valley	171.0	704.0	38033	1895	
370	ND	kidder	364.0	911.0	38043	2419	
385	ND	sheridan	179.0	607.0	38083	1395	
386	ND	steele	557.0	399.0	38091	1969	
389	ND	wells	622.0	1601.0	38103	4179	
392	NE	arthur	24.0	202.0	31005	437	
414	NE	gosper	147.0	667.0	31073	1977	
415	NE	grant	14.0	282.0	31075	647	
421	NE	jefferson	924.0	1913.0	31095	7354	
428	NE	loup	51.0	247.0	31115	542	
430	NE	mcpherson	22.0	228.0	31117	425	
437	NE	pawnee	332.0	774.0	31133	2695	
451	NE	thomas	36.0	291.0	31171	675	
455	NE	wheeler	60.0	343.0	31183	805	
472	NM	de baca	267.0	342.0	35011	1977	
476	NM	harding	198.0	199.0	35021	565	
494	NV	esmeralda	52.0	280.0	32009	1069	
754	TX	borden	22.0	320.0	48033	698	

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Vo Age_Popul
807	TX	glasscock	37.0	513.0	48173	1253	
830	TX	irion	96.0	636.0	48235	1631	
834	TX	jim hogg	1060.0	409.0	48247	5218	
840	TX	kenedy	77.0	100.0	48261	558	
843	TX	king	6.0	124.0	48269	274	
856	TX	loving	6.0	46.0	48301	76	
863	TX	mcmullen	41.0	387.0	48311	671	
925	TX	yoakum	335.0	1558.0	48501	8316	
930	UT	daggett	80.0	335.0	49009	751	
990	VA	highland	400.0	772.0	51091	2230	
1065	WA	garfield	445.0	880.0	53023	2231	
1196	WY	sublette	668.0	2653.0	56035	10032	

In [18]:
```
## Task4

# We need to figure out the missing values in the dataset.


seriesObj = data_merged.apply(lambda x: True if x.iloc[2] == 0 else Fals
e , axis=1)
numOfRows = len(seriesObj[seriesObj == True].index)
print('Number of Rows in dataframe in which Votes_Democratic is 0 : ', n
umOfRows)

seriesObj = data_merged.apply(lambda x: True if x.iloc[3] == 0 else Fals
e , axis=1)
numOfRows = len(seriesObj[seriesObj == True].index)
print('Number of Rows in dataframe in which Votes_Republican is 0 : ', n
umOfRows)

seriesObj = data_merged.apply(lambda x: True if x.iloc[2] == 0 or x.iloc
[3] == 0 else False , axis=1)
numOfRows = len(seriesObj[seriesObj == True].index)
print('Number of Rows in dataframe in which Votes_Democratic or Votes_Re
public is 0 : ', numOfRows)
print(seriesObj[seriesObj == True].index)

# Citizen Voting-Age Population variable can be dropped because there ar
e too many observations with 0's.

seriesObj = data_merged.apply(lambda x: True if x['Citizen_Voting-Age_Po
pulation'] == 0 else False , axis=1)
numOfRows = len(seriesObj[seriesObj == True].index)
print('Number of Rows in dataframe in which Citizen Voting-Age Populatio
n is 0 : ', numOfRows)


seriesObj = data_merged.apply(lambda x: True if x['Percent_Unemployed']
== 0 else False , axis=1)
numOfRows = len(seriesObj[seriesObj == True].index)
print('Number of Rows in dataframe in which Unemployment Rate is 0 : ',
numOfRows)


seriesObj = data_merged.apply(lambda x: True if x['Percent_Rural'] == 0
else False , axis=1)
numOfRows = len(seriesObj[seriesObj == True].index)
print('Number of Rows in dataframe in which Percent Rural is 0 : ', numO
fRows)

seriesObj = data_merged.apply(lambda x: True if x['Percent_Rural'] == 10
0 else False , axis=1)
numOfRows = len(seriesObj[seriesObj == True].index)
print('Number of Rows in dataframe in which Percent Rural is 100 : ', nu
mOfRows)
```

```
Number of Rows in dataframe in which Votes_Democratic is 0 :  5
Number of Rows in dataframe in which Votes_Republican is 0 :  5
Number of Rows in dataframe in which Votes_Democratic or Votes_Republic
is 0 :  5
Int64Index([425, 714, 750, 865, 1114], dtype='int64')
Number of Rows in dataframe in which Citizen Voting-Age Population is 0
:  680
Number of Rows in dataframe in which Unemployment Rate is 0 :  3
Number of Rows in dataframe in which Percent Rural is 0 :  19
Number of Rows in dataframe in which Percent Rural is 100 :  238
```

In [19]:
```
## Task4

# To drop the 'Citizen Voting-Age Population' column
data_merged.drop(['Citizen_Voting-Age_Population'], axis=1)
```

Out[19]:

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Percent_W
0	AZ	apache	16298.0	7810.0	4001	72346	
1	AZ	cochise	17383.0	26929.0	4003	128177	
2	AZ	coconino	34240.0	19249.0	4005	138064	
3	AZ	gila	7643.0	12180.0	4007	53179	
4	AZ	graham	3368.0	6870.0	4009	37529	
...	...	...	...	...	...	...	
1195	WY	platte	801.0	2850.0	56031	8740	
1196	WY	sublette	668.0	2653.0	56035	10032	
1197	WY	sweetwater	3943.0	8577.0	56037	44812	
1198	WY	uinta	1371.0	4713.0	56041	20893	
1199	WY	washakie	588.0	2423.0	56043	8351	

1200 rows × 19 columns

In [20]:
```
## Task4

# ignore the value

# To estimate and replace the 'Votes_Democratic' and 'Votes_Democratic'
 variables if the the values are 0's.
# How to estimate:
#   If the value is 0, then get the mean of the same State to set the me
an value to the variable.

#print (data_merged['Votes_Democratic'] == '0')

#(data_merged['Votes_Democratic'].value_counts()/data_merged['Total_Popu
lation'].count())*100
```

In [21]:
```
## Task5

# Create a new variable named "Party" with values 0 and 1 (Democratic :
 1 and Republican : 0)
def get_party(data_merged):
    if data_merged['Votes_Democratic'] > data_merged['Votes_Republican'
]:
        val = 1
    else:
        val = 0

    # For the rows where the number of votes are 0, set val as -1 as inv
alid value.
    if data_merged['Votes_Democratic'] == 0 or data_merged['Votes_Republ
ican'] == 0:
        val = -1

    return val

# To expect correct result on this, we need to figure out the missing va
lues on votes_democratic and votes_republican at Task4.

data_merged.head()
```

Out[21]:

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Voting-Age_Population
0	AZ	apache	16298.0	7810.0	4001	72346	0
1	AZ	cochise	17383.0	26929.0	4003	128177	92915
2	AZ	coconino	34240.0	19249.0	4005	138064	104265
3	AZ	gila	7643.0	12180.0	4007	53179	0
4	AZ	graham	3368.0	6870.0	4009	37529	0

In [22]:
```
data_merged['Party'] = data_merged.apply(get_party, axis=1)
```

In [23]:
```
# delete the five rows which do not have the number of votes.

data_merged = data_merged[data_merged.Party != -1]
```

In [175]:
```
data_merged.to_csv('p1_t3_data_deleted_5rows.csv')
```

In [24]:
```
data_merged.head()
```

Out[24]:

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Voting-Age_Population
0	AZ	apache	16298.0	7810.0	4001	72346	0
1	AZ	cochise	17383.0	26929.0	4003	128177	92915
2	AZ	coconino	34240.0	19249.0	4005	138064	104265
3	AZ	gila	7643.0	12180.0	4007	53179	0
4	AZ	graham	3368.0	6870.0	4009	37529	0

5 rows × 21 columns

In [177]:
```
## Task5: verify with csv file

data_merged.to_csv('p1_t5_data_with_party.csv')
```

In [25]:
```
## Task 6

df_democratic = data_merged[(data_merged['Party'] == 1)]
df_democratic.reset_index(inplace= True)

df_republican = data_merged[(data_merged['Party'] == 0)]
df_republican.reset_index(inplace= True)
```

In [26]:
```
## Task6

# Democratic mean total population
# The mean population of Democratic Counties is greater than Republican
 counties

df_democratic.groupby('Party')['Total_Population'].describe()
```

Out[26]:

Party	count	mean	std	min	25%	50%	75%	max
1	325.0	300998.316923	553600.025712	1969.0	23645.0	82049.0	284788.0	4434257.0

In [27]:
```
## Task6

# Republican mean total population

df_republican.groupby('Party')['Total_Population'].describe()
```

Out[27]:

Party	count	mean	std	min	25%	50%	75%	max
0	870.0	53864.672414	94192.572794	76.0	9559.5	25465.0	53721.0	1092518.0

In [28]:
```
## Task6

# get the t-test statistics and p-value.
ttest_ind(df_democratic['Total_Population'], df_republican['Total_Popula
tion'], equal_var=True)
```

Out[28]:
```
Ttest_indResult(statistic=12.692959605629676, pvalue=1.0170787111521377
e-34)
```

In [29]:
```
## Task6

# assign the result to variables
t6_ttest_stat, t6_pvalue = ttest_ind(df_democratic['Total_Population'],
df_republican['Total_Population'], equal_var=True)
```

In [30]:
```
## Task6

# Null Hypythosis: The mean of total population between Democratic and R
epublic counties are the same.
# Used student's t-test

# t-test statistics result
print('t-test statistics:', t6_ttest_stat)
print('pvalue:', t6_pvalue)

# interpretation
alpha = 0.05
if t6_pvalue > alpha:
    print('Same distributions (fail to reject Null Hypythosis). The mean
population of democratic counties is the same as republican counties.')
else:
    print('Different distributions (reject Null Hypythosis). The mean po
pulation of democratic counties is greater than republican counties.')
```

```
t-test statistics: 12.692959605629676
pvalue: 1.0170787111521377e-34
Different distributions (reject Null Hypythosis). The mean population o
f democratic counties is greater than republican counties.
```

In [265]:
```
## Task6

# Conclusion:
# We get a p-value (5.085393555760689e-35) which is approaching to 0 tha
t indicates strong evidence against the null hypothesis,
# so we reject the null hypothesis.
# And we conclude that this difference in the mean of population of Rebu
blic and democratic countries is statistically significant at the α = 0.0
5 significance level.
# The interpretation of the statistic finds that the means are differen
t, with a significance of at least 5%,
# and this differencce in the means is not due to some chance there a st
rong eveidence behind it.
```

In [31]:
```
## Task7

# Democratic mean median household income
# The mean median household income of Democratic Counties is greater tha
n Republican counties

df_democratic.groupby('Party')['Median_Household_Income'].describe()
```

Out[31]:

Party	count	mean	std	min	25%	50%	75%	max
1	325.0	53798.732308	15289.130077	21190.0	44140.0	51477.0	59132.0	125672.0

In [32]:
```
## Task7

# Republican mean total population

df_republican.groupby('Party')['Median_Household_Income'].describe()
```

Out[32]:

Party	count	mean	std	min	25%	50%	75%	max
0	870.0	48746.81954	10670.729412	24000.0	41506.5	47168.5	53423.25	108177.0

In [33]:
```
## Task7

# get the t-test statistics and p-value.
ttest_ind(df_democratic['Median_Household_Income'], df_republican['Media
n_Household_Income'], equal_var=True)
```

Out[33]:
```
Ttest_indResult(statistic=6.421918696825302, pvalue=1.9389021460182502e
-10)
```

In [34]:
```
## Task7

# assign the result to variables
t7_ttest_stat, t7_pvalue = ttest_ind(df_democratic['Median_Household_Inc
ome'], df_republican['Median_Household_Income'], equal_var=True)
```

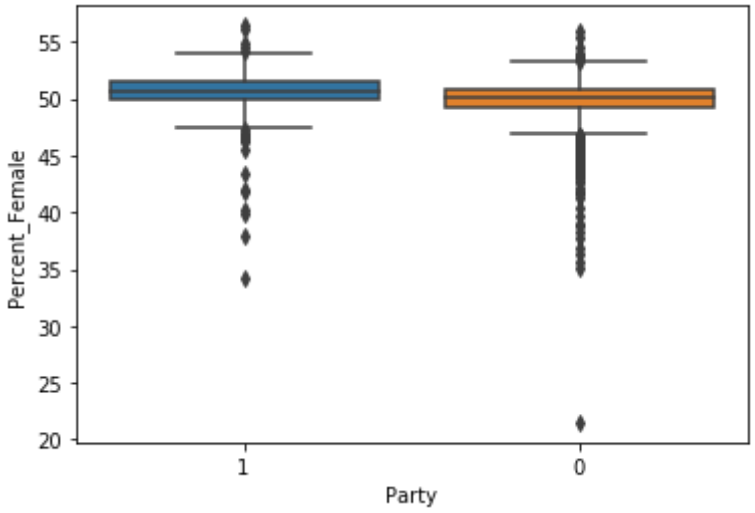
In [35]:
```
## Task7

# Null Hypythosis: The mean of median household income between Democrati
c and Republic counties are the same.
# Used student's t-test

# t-test statistics result
print('t-test statistics:', t7_ttest_stat)
print('pvalue:', t7_pvalue)

# interpretation
alpha = 0.05
if t7_pvalue > alpha:
    print('Same distributions (fail to reject Null Hypythosis). The mean
population of democratic counties is the same as republican counties.')
else:
    print('Different distributions (reject Null Hypythosis). The mean po
pulation of democratic counties is greater than republican counties.')
```

```
t-test statistics: 6.421918696825302
pvalue: 1.9389021460182502e-10
Different distributions (reject Null Hypythosis). The mean population o
f democratic counties is greater than republican counties.
```

In [117]:
```
## Task7

# Conclusion:
# We get a p-value (9.694510730091251e-11) which is approaching to 0 tha
t indicates strong evidence against the null hypothesis,
# so we reject the null hypothesis.
# And we conclude that this difference in the mean of median household i
ncome of Rebublic and democratic countries is statistically significant
 at the α = 0.05 significance level.
# The interpretation of the statistic finds that the means are differen
t, with a significance of at least 5%,
# and this differencce in the means is not due to some chance there a st
rong eveidence behind it.
```

In [36]:
```
## Task8

# gender
sns.boxplot(x='Party', y='Percent_Female', data=data_merged, order=[1,0
])
```

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x10805ba10>



In [37]:
```
df_democratic['Percent_Female'].describe()
```

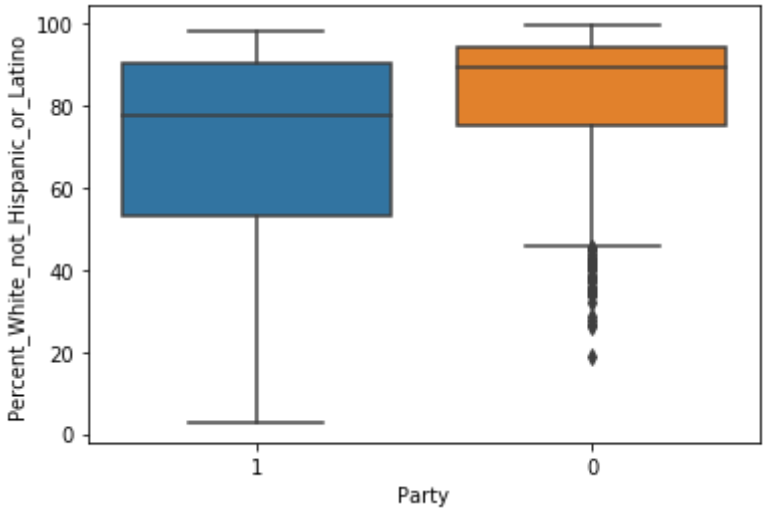
Out[37]:
```
count    325.000000
mean      50.385433
std        2.149359
min       34.245291
25%       49.854280
50%       50.653830
75%       51.492075
max       56.418468
Name: Percent_Female, dtype: float64
```

In [38]:
```
df_republican['Percent_Female'].describe()
```

Out[38]:
```
count    870.000000
mean      49.630898
std        2.429013
min       21.513413
25%       49.222905
50%       50.176792
75%       50.829770
max       55.885023
Name: Percent_Female, dtype: float64
```

In [39]:
```
## Task8

# race (white)
sns.boxplot(x='Party', y='Percent_White_not_Hispanic_or_Latino', data=da
ta_merged, order=[1,0])
```

Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x11a49d510>



In [40]:
```
df_democratic['Percent_White_not_Hispanic_or_Latino'].describe()
```

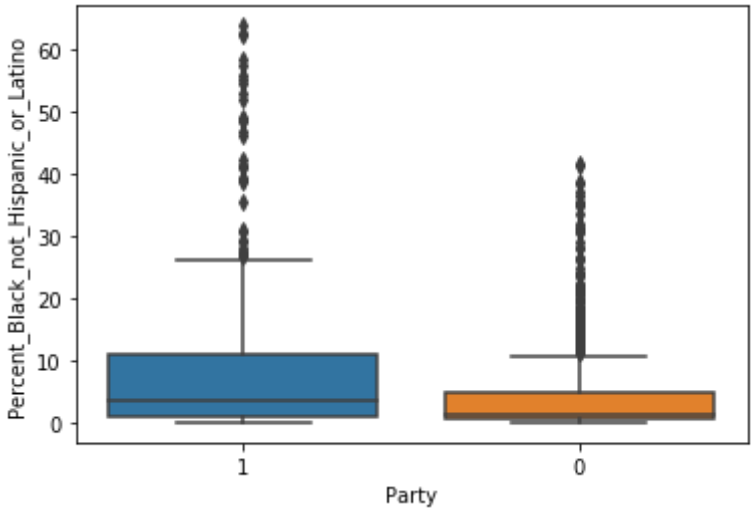
Out[40]:
```
count    325.000000
mean      69.683766
std       24.981502
min        2.776702
25%       53.271579
50%       77.786090
75%       90.300749
max       98.063495
Name: Percent_White_not_Hispanic_or_Latino, dtype: float64
```

In [41]:
```
df_republican['Percent_White_not_Hispanic_or_Latino'].describe()
```

Out[41]:
```
count    870.000000
mean      82.656646
std       16.056122
min       18.758977
25%       75.016397
50%       89.434849
75%       94.466596
max       99.627329
Name: Percent_White_not_Hispanic_or_Latino, dtype: float64
```

In [42]:
```
## Task8

# race (black)
sns.boxplot(x='Party', y='Percent_Black_not_Hispanic_or_Latino', data=da
ta_merged, order=[1,0])
```

Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x11a605210>



In [43]:
```
df_democratic['Percent_Black_not_Hispanic_or_Latino'].describe()
```

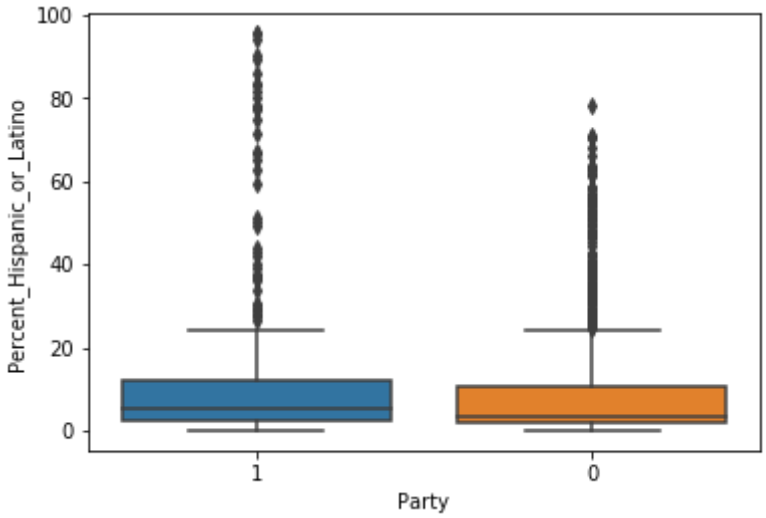
Out[43]:
```
count    325.000000
mean       9.242649
std       13.351340
min        0.000000
25%        0.839103
50%        3.485992
75%       11.058843
max       63.953279
Name: Percent_Black_not_Hispanic_or_Latino, dtype: float64
```

In [44]:
```
df_republican['Percent_Black_not_Hispanic_or_Latino'].describe()
```

Out[44]:
```
count    870.000000
mean       4.189241
std        6.721695
min        0.000000
25%        0.460419
50%        1.318311
75%        4.753831
max       41.563041
Name: Percent_Black_not_Hispanic_or_Latino, dtype: float64
```

In [45]:
```
## Task8

# race (hispanic)
sns.boxplot(x='Party', y='Percent_Hispanic_or_Latino', data=data_merged,
order=[1,0])
```

Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x11a6b0d10>



In [46]:
```
df_democratic['Percent_Hispanic_or_Latino'].describe()
```

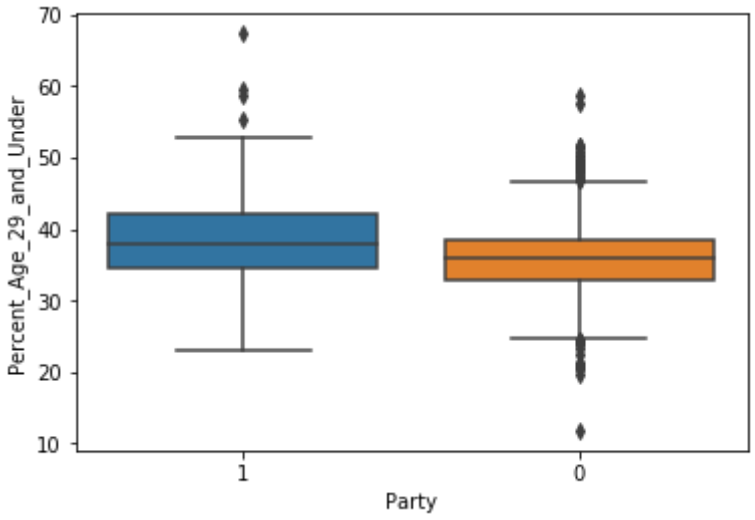
Out[46]:
```
count    325.000000
mean      12.587391
std       19.575030
min        0.193349
25%        2.531017
50%        5.039747
75%       11.857116
max       95.479801
Name: Percent_Hispanic_or_Latino, dtype: float64
```

In [47]:
```
df_republican['Percent_Hispanic_or_Latino'].describe()
```

Out[47]:
```
count    870.000000
mean       9.733094
std       14.049576
min        0.000000
25%        1.704539
50%        3.427435
75%       10.709696
max       78.397012
Name: Percent_Hispanic_or_Latino, dtype: float64
```

In [48]:
```
## Task8

# age (29 and under)
sns.boxplot(x='Party', y='Percent_Age_29_and_Under', data=data_merged, o
rder=[1,0])
```

Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x11a7d8bd0>



In [49]:
```
df_democratic['Percent_Age_29_and_Under'].describe()
```

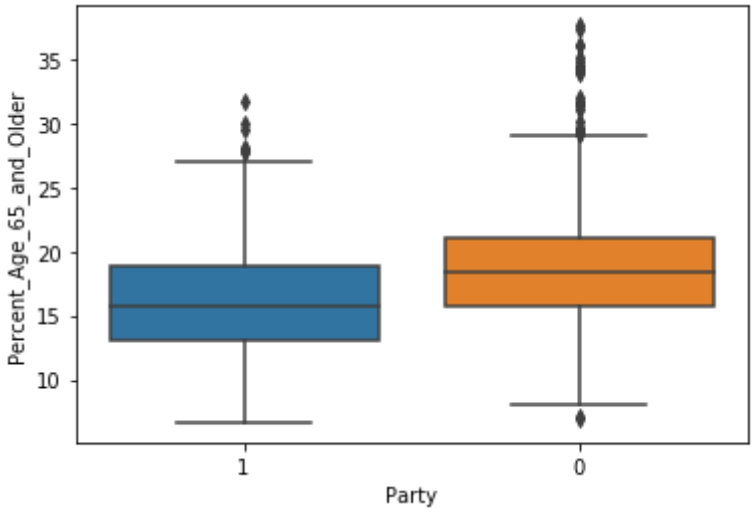
Out[49]:
```
count    325.000000
mean      38.726959
std        6.252786
min       23.156452
25%       34.488444
50%       38.074151
75%       42.161162
max       67.367823
Name: Percent_Age_29_and_Under, dtype: float64
```

In [50]:
```
df_republican['Percent_Age_29_and_Under'].describe()
```

Out[50]:
```
count    870.000000
mean      36.005719
std        5.181522
min       11.842105
25%       32.983652
50%       35.846532
75%       38.539787
max       58.749116
Name: Percent_Age_29_and_Under, dtype: float64
```

In [51]:
```
## Task8

# age (65 and older)
sns.boxplot(x='Party', y='Percent_Age_65_and_Older', data=data_merged, o
rder=[1,0])
```

Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x1c23f77f50>



In [52]:
```
df_democratic['Percent_Age_65_and_Older'].describe()
```

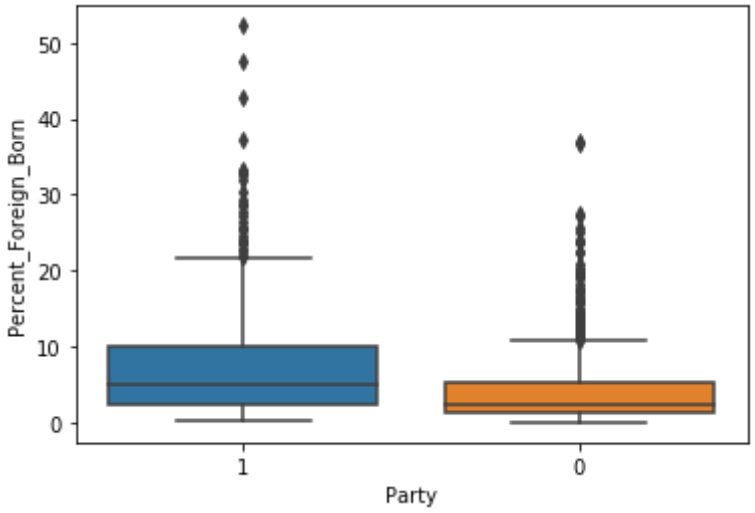
Out[52]:
```
count    325.000000
mean      16.194826
std        4.282422
min        6.653188
25%       13.106233
50%       15.698087
75%       18.806426
max       31.642106
Name: Percent_Age_65_and_Older, dtype: float64
```

In [53]:
```
df_republican['Percent_Age_65_and_Older'].describe()
```

Out[53]:
```
count    870.000000
mean      18.828267
std        4.733155
min        6.954387
25%       15.784982
50%       18.377896
75%       21.112847
max       37.622759
Name: Percent_Age_65_and_Older, dtype: float64
```

In [54]:
```
## Task8

# ethnicity
sns.boxplot(x='Party', y='Percent_Foreign_Born', data=data_merged, order
=[1,0])
```

Out[54]: <matplotlib.axes._subplots.AxesSubplot at 0x1c24237ad0>



In [55]:
```
df_democratic['Percent_Foreign_Born'].describe()
```

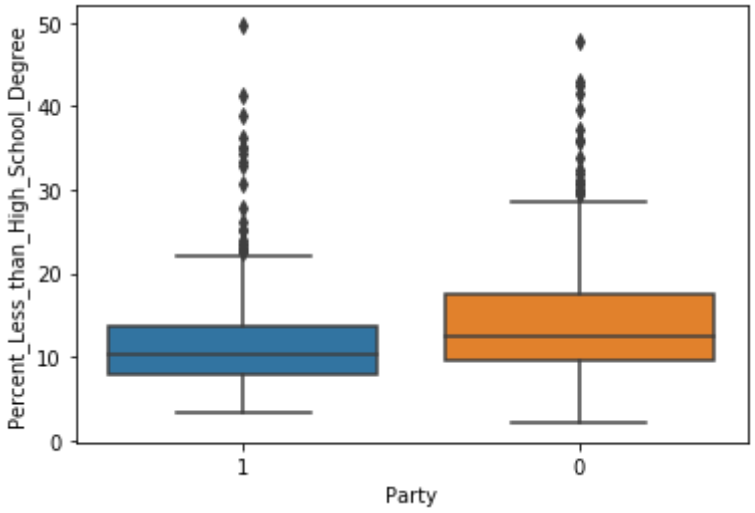
Out[55]:
```
count    325.000000
mean       7.986330
std        8.330740
min        0.179769
25%        2.470508
50%        5.105490
75%       10.144555
max       52.229868
Name: Percent_Foreign_Born, dtype: float64
```

In [56]:
```
df_republican['Percent_Foreign_Born'].describe()
```

Out[56]:
```
count    870.000000
mean       3.990096
std        4.507786
min        0.000000
25%        1.320101
50%        2.326317
75%        5.149429
max       37.058317
Name: Percent_Foreign_Born, dtype: float64
```

In [57]:
```
## Task8

# education (less than high school)
sns.boxplot(x='Party', y='Percent_Less_than_High_School_Degree', data=da
ta_merged, order=[1,0])
```

Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x1c247eebd0>



In [58]:
```
df_democratic['Percent_Less_than_High_School_Degree'].describe()
```

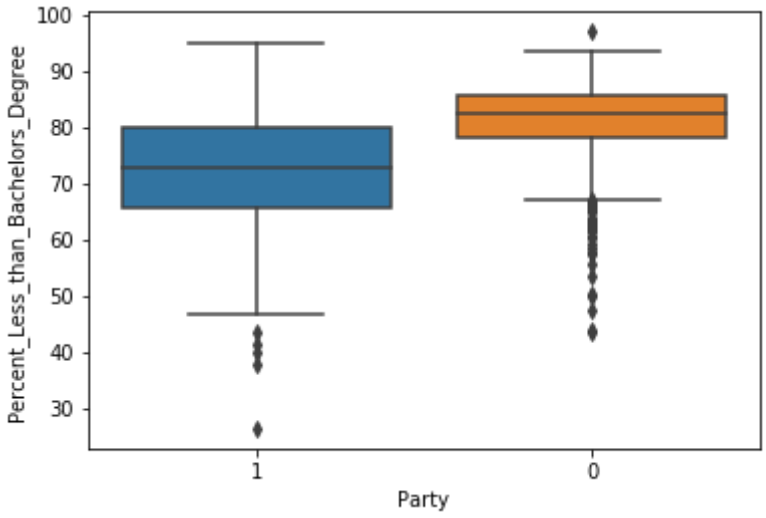
Out[58]:
```
count    325.000000
mean      11.883760
std        6.505613
min        3.215803
25%        7.893714
50%       10.370080
75%       13.637059
max       49.673777
Name: Percent_Less_than_High_School_Degree, dtype: float64
```

In [59]:
```
df_republican['Percent_Less_than_High_School_Degree'].describe()
```

Out[59]:
```
count    870.000000
mean      14.009112
std        6.303126
min        2.134454
25%        9.662491
50%       12.572435
75%       17.447168
max       47.812773
Name: Percent_Less_than_High_School_Degree, dtype: float64
```

In [60]:
```
## Task8

# education (less than bachelor degree)
sns.boxplot(x='Party', y='Percent_Less_than_Bachelors_Degree', data=data
_merged, order=[1,0])
```

Out[60]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1c248bcc90>
```



In [61]:
```
df_democratic['Percent_Less_than_Bachelors_Degree'].describe()
```

Out[61]:
```
count    325.000000
mean      71.968225
std       11.192404
min       26.335440
25%       65.711800
50%       72.736143
75%       79.903653
max       94.849957
Name: Percent_Less_than_Bachelors_Degree, dtype: float64
```

In [62]:
```
df_republican['Percent_Less_than_Bachelors_Degree'].describe()
```

Out[62]:
```
count    870.000000
mean      81.095427
std        6.815537
min       43.419470
25%       78.108424
50%       82.406700
75%       85.546272
max       97.014925
Name: Percent_Less_than_Bachelors_Degree, dtype: float64
```

In [63]:
```
## Task8
# Seperate the Democratic Countries and displaying first 5 rows:
df_democratic_t8 = data_merged.loc[data_merged.Party == 1, :]
```

In [64]:
```
df_democratic_t8.head()
```

Out[64]:

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Voting-Age_Population
0	AZ	apache	16298.0	7810.0	4001	72346	0
2	AZ	coconino	34240.0	19249.0	4005	138064	104265
6	AZ	maricopa	732671.0	672505.0	4013	4088549	2723565
9	AZ	pima	221242.0	160550.0	4019	1003338	0
10	AZ	santa cruz	9241.0	3828.0	4023	46547	27155

5 rows × 21 columns

In [65]:
```
# General statistics about the data set for Democratic countries
democratic_desc = df_democratic_t8.describe()
democratic_desc
#In the below table we can observe all the descriptive statistics relate
d to all the variables
```

Out[65]:

	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Voting-Age_Population	Perc
count	325.000000	325.000000	325.000000	3.250000e+02	3.250000e+02	
mean	71193.172308	41322.861538	37130.873846	3.009983e+05	7.249500e+04	
std	125306.803889	74689.108440	13860.571592	5.536000e+05	2.222767e+05	
min	521.000000	220.000000	4001.000000	1.969000e+03	0.000000e+00	
25%	5242.000000	3611.000000	27027.000000	2.364500e+04	0.000000e+00	
50%	18159.000000	12348.000000	36103.000000	8.204900e+04	0.000000e+00	
75%	72677.000000	46403.000000	51095.000000	2.847880e+05	3.441500e+04	
max	881802.000000	672505.000000	56001.000000	4.434257e+06	2.723565e+06	

In [66]:
```
# Seperate the Republic Counties and displaying first 5 rows:
df_republican_t8 = data_merged.loc[data_merged.Party == 0, :]
```

In [67]:
```
df_republican_t8.head()
```

Out[67]:

	State	County	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Voting-Age_Population	I
1	AZ	cochise	17383.0	26929.0	4003	128177	92915	
3	AZ	gila	7643.0	12180.0	4007	53179	0	
4	AZ	graham	3368.0	6870.0	4009	37529	0	
5	AZ	la paz	1609.0	3265.0	4012	20304	15245	
7	AZ	mohave	19214.0	50209.0	4015	203629	0	

5 rows × 21 columns

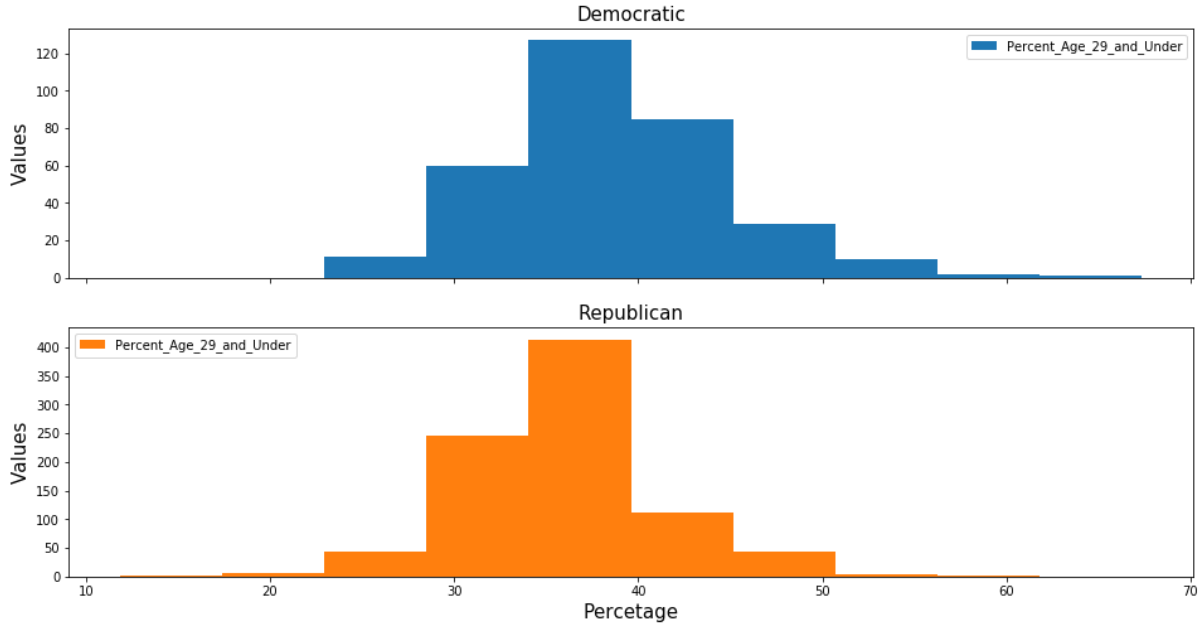
In [68]:
```
# General statistics about the data set for Democratic countries
republic_desc = df_republican_t8.describe()
republic_desc
#In the below table we can observe all the descriptive statistics relate
d to all the variables
```

Out[68]:

	Votes_Democratic	Votes_Republican	FIPS	Total_Population	Citizen_Voting-Age_Population	Perc
count	870.000000	870.000000	870.000000	8.700000e+02	870.000000	
mean	7926.549425	12644.403448	38714.074713	5.386467e+04	17423.247126	
std	17538.649168	22601.266060	12658.615292	9.419257e+04	47283.487587	
min	6.000000	46.000000	4003.000000	7.600000e+01	0.000000	
25%	951.500000	2544.000000	30073.500000	9.559500e+03	0.000000	
50%	2807.500000	5932.500000	42040.000000	2.546500e+04	0.000000	
75%	7010.750000	12632.750000	48342.500000	5.372100e+04	15635.000000	
max	215190.000000	219990.000000	56043.000000	1.092518e+06	460215.000000	

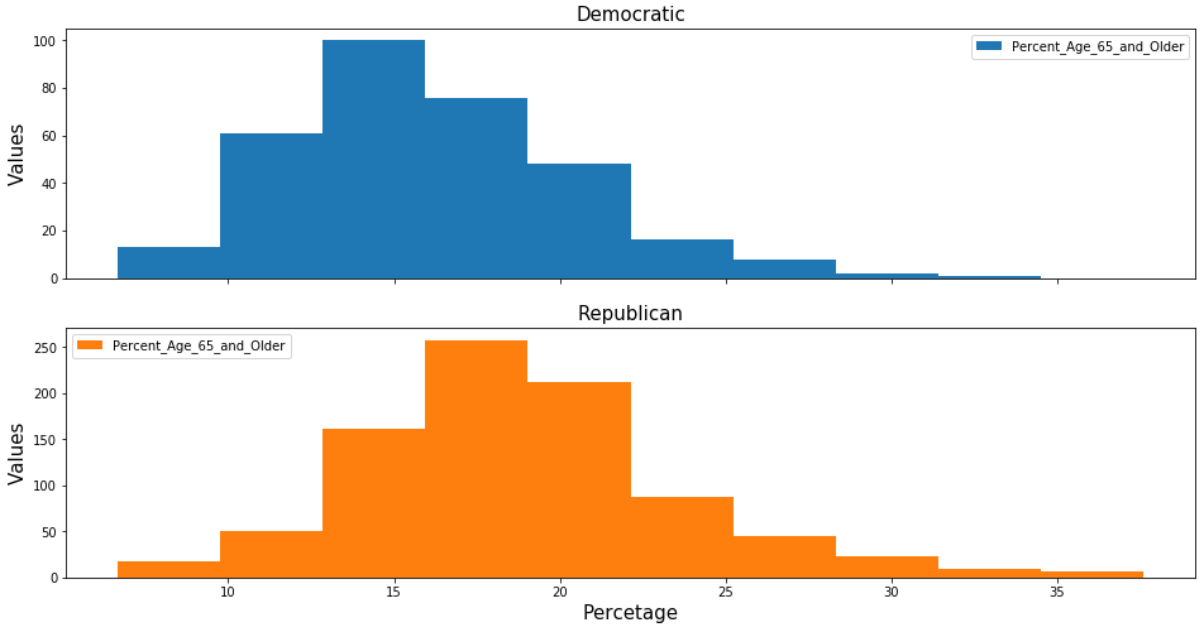
In [78]:
```
# Age Comparison : Democratic Vs Republic
#Hitogram for Percent Age 29 and Under

Percent_Age_29 = pd.DataFrame([df_democratic_t8['Percent_Age_29_and_Unde
r'],df_republican_t8['Percent_Age_29_and_Under']])
# Convert row to column
Percent_Age_29 = Percent_Age_29.transpose()
# Creatinng a bar plot
axes = Percent_Age_29.plot.hist(rot=0, subplots=True,figsize = (16,8))
axes[1].legend(loc=2)
#subplot for Democratic

axes[0].set_xlabel("Percetage",fontsize=15)
axes[0].set_ylabel("Values",fontsize=15)


axes[1].set_xlabel("Percetage",fontsize=15)
axes[1].set_ylabel("Values",fontsize=15)


axes[0].set_title('Democratic',fontsize = 15)
# Subplot for Republic
axes[1].set_title('Republican',fontsize = 15)
```

Out[78]:
```
Text(0.5, 1.0, 'Republican')
```



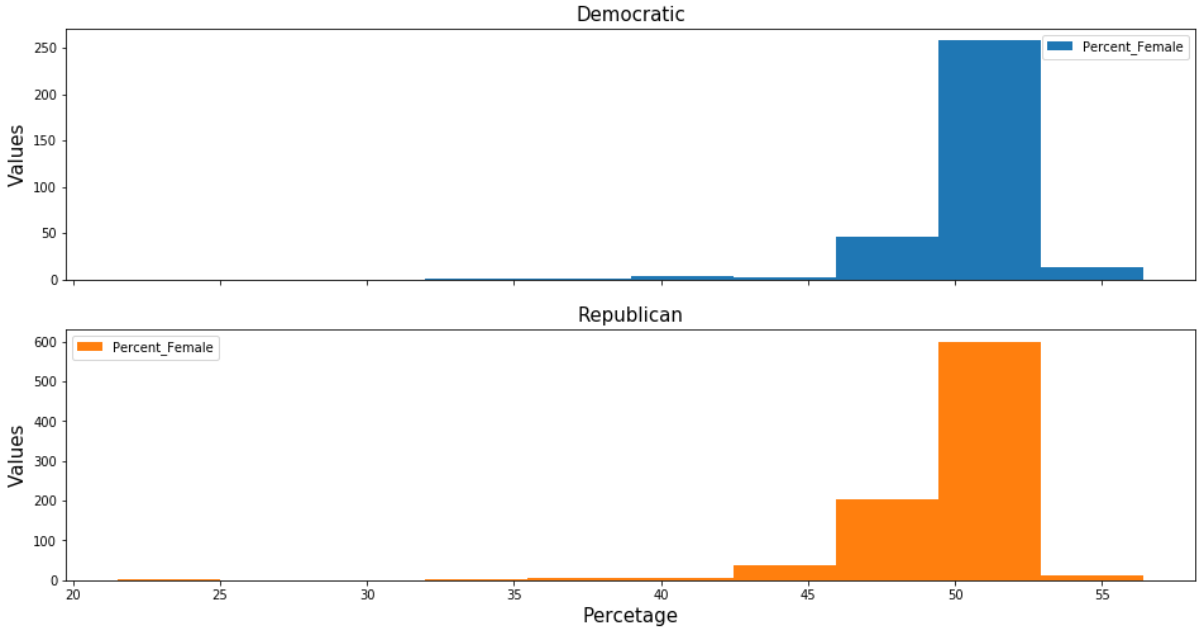
In [79]:
```
# Histogram for Percent Age 65 and Older
Percent_Age_65 = pd.DataFrame([df_democratic_t8['Percent_Age_65_and_Olde
r'],df_republican_t8['Percent_Age_65_and_Older']])
# Convert row to column
Percent_Age_65 = Percent_Age_65.transpose()
# Creatinng a bar plot
axes = Percent_Age_65.plot.hist(rot=0, subplots=True,figsize = (16,8))
axes[1].legend(loc=2)
#subplot for Democratic

axes[0].set_xlabel("Percetage",fontsize=15)
axes[0].set_ylabel("Values",fontsize=15)


axes[1].set_xlabel("Percetage",fontsize=15)
axes[1].set_ylabel("Values",fontsize=15)


axes[0].set_title('Democratic',fontsize = 15)
# Subplot for Republic
axes[1].set_title('Republican',fontsize = 15)
```

Out[79]: Text(0.5, 1.0, 'Republican')



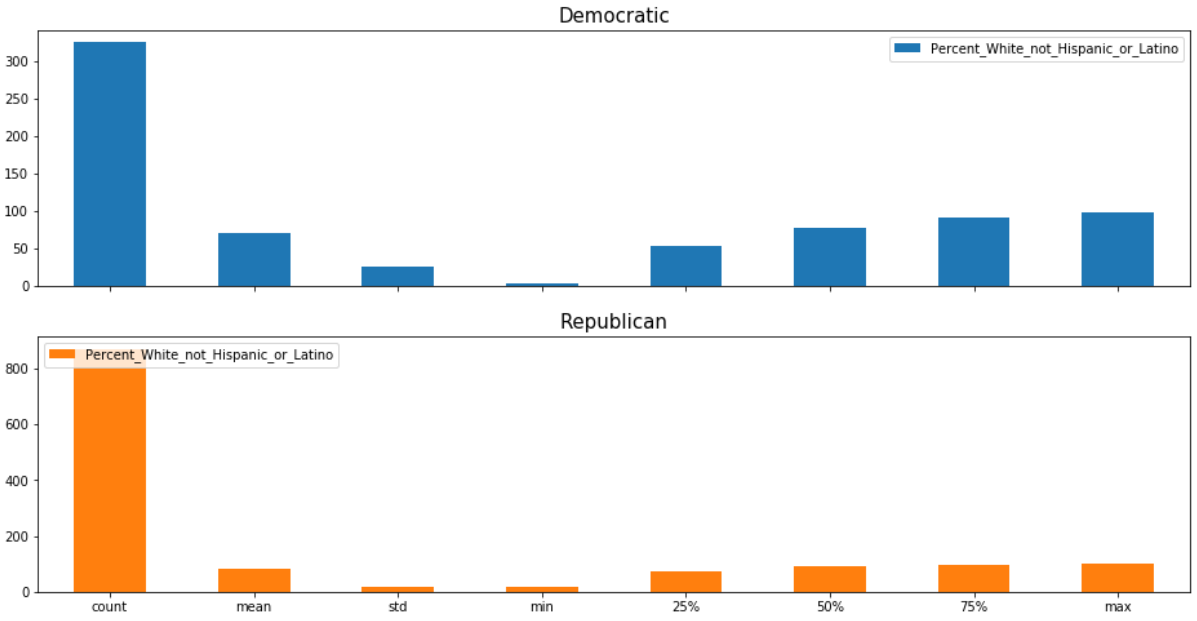
In [80]:
```
#Gender Comparison: Democratic Vs Republic
Percent_Female = pd.DataFrame([df_democratic_t8['Percent_Female'],df_rep
ublican_t8['Percent_Female']])
# Convert row to column
Percent_Female = Percent_Female.transpose()
# Creatinng a bar plot
axes = Percent_Female.plot.hist(rot=0, subplots=True,figsize = (16,8))
axes[1].legend(loc=2)
#subplot for Democratic

axes[0].set_xlabel("Percetage",fontsize=15)
axes[0].set_ylabel("Values",fontsize=15)


axes[1].set_xlabel("Percetage",fontsize=15)
axes[1].set_ylabel("Values",fontsize=15)



axes[0].set_title('Democratic',fontsize = 15)
# Subplot for Republic
axes[1].set_title('Republican',fontsize = 15)
```

Out[80]: Text(0.5, 1.0, 'Republican')



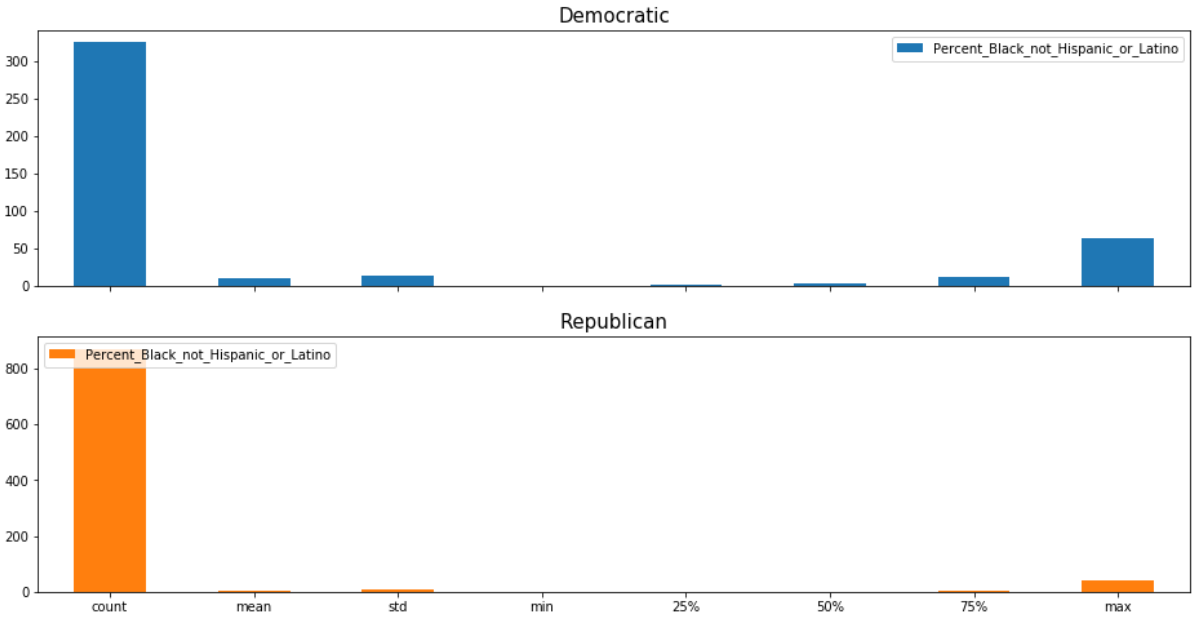
In [81]:
```
# Race and Ethnicity Comparison: Democratic Vs Republic
#Descriptive Statistic for Percent White, not Hispanic or Latino
Percent_White = pd.DataFrame([democratic_desc['Percent_White_not_Hispani
c_or_Latino'],republic_desc['Percent_White_not_Hispanic_or_Latino']])
# Convert row to column
Percent_White = Percent_White.transpose()
# Creatinng a bar plot
axes = Percent_White.plot.bar(rot=0, subplots=True,figsize = (16,8))
axes[1].legend(loc=2)
#subplot for Democratic
axes[0].set_title('Democratic',fontsize = 15)
# Subplot for Republic
axes[1].set_title('Republican',fontsize = 15)
```

Out[81]: Text(0.5, 1.0, 'Republican')



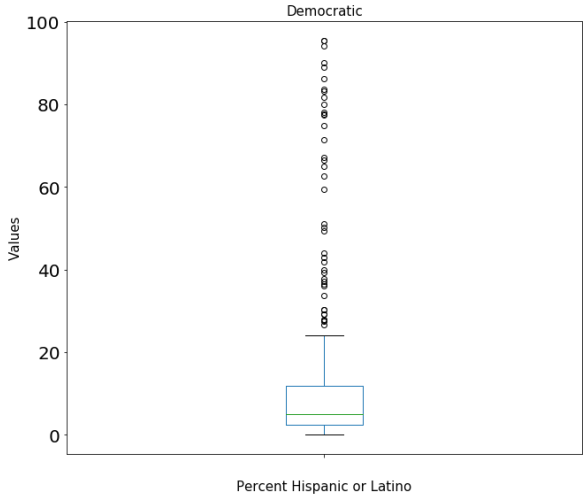
In [82]:
```
#Descriptive Statistic for Percent Black, not Hispanic or Latino
Percent_Black = pd.DataFrame([democratic_desc['Percent_Black_not_Hispanic_or_Latino'],republic_desc['Percent_Black_not_Hispanic_or_Latino']])
# taking transpose
Percent_Black = Percent_Black.transpose()
# Creatinng a bar plot
axes = Percent_Black.plot.bar(rot=0, subplots=True,figsize = (16,8))
axes[1].legend(loc=2)
#subplot for Democratic
axes[0].set_title('Democratic',fontsize = 15)
# Subplot for Republic
axes[1].set_title('Republican',fontsize = 15)
```

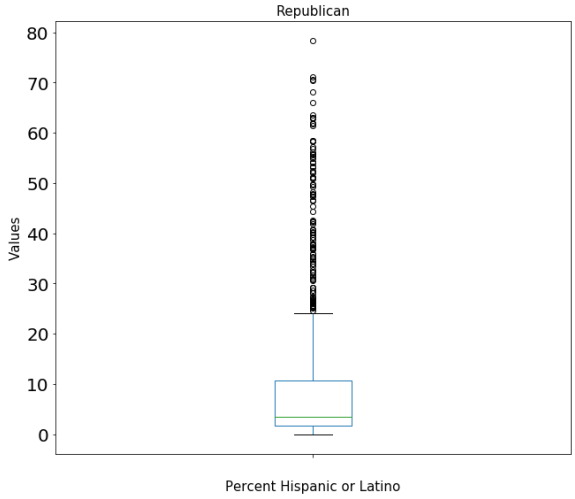
Out[82]: Text(0.5, 1.0, 'Republican')



In [83]:
```
#Descriptive Statistic for Percent Hispanic or Latino
#selecting two col. and 1 rows for subplots
fig, axes = plt.subplots(nrows=1, ncols=2)
#subplot for Democratic
ax = df_democratic_t8['Percent_Hispanic_or_Latino'].plot(ax=axes[0],kind
= 'box',figsize=(23,9),label=' ',fontsize=20,legend=True,title='')
#subplot for Republic
ay = df_republican_t8['Percent_Hispanic_or_Latino'].plot(ax=axes[1],kind
= 'box',figsize=(23,9),label=' ',fontsize=20,legend=True,title='')

# setting x labels
ax.set_xlabel("Percent Hispanic or Latino",fontsize=15)
ay.set_xlabel("Percent Hispanic or Latino",fontsize=15)
#setting y label
ax.set_ylabel("Values",fontsize=15)
ay.set_ylabel("Values",fontsize=15)
#title
ax.set_title('Democratic',fontsize=15 )
ay.set_title('Republican',fontsize=15 )
```

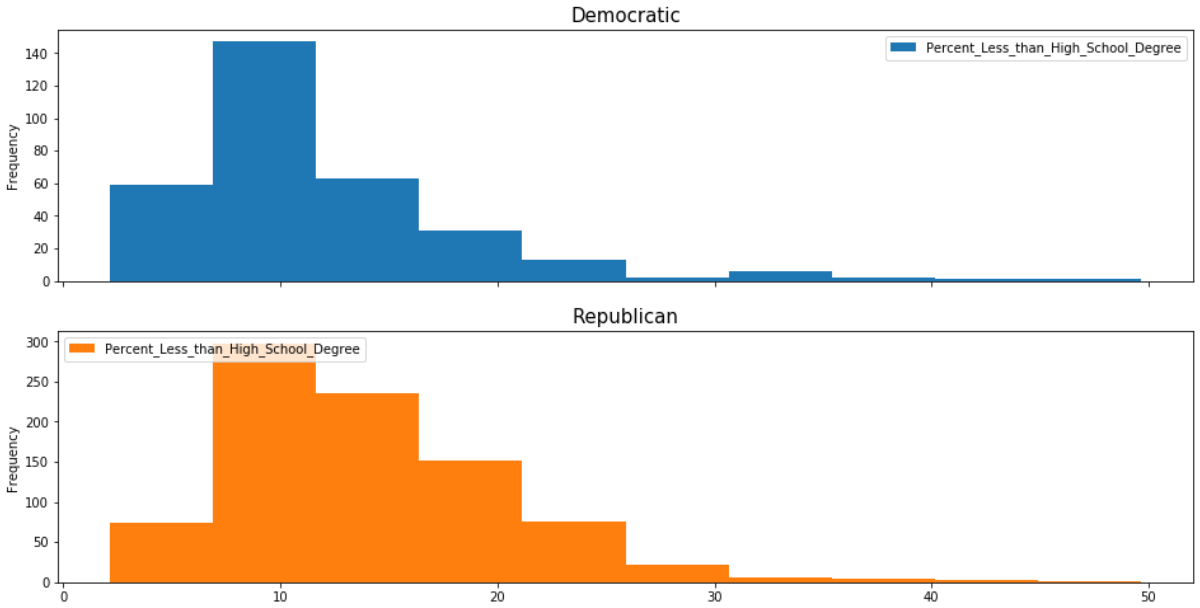
Out[83]: Text(0.5, 1.0, 'Republican')





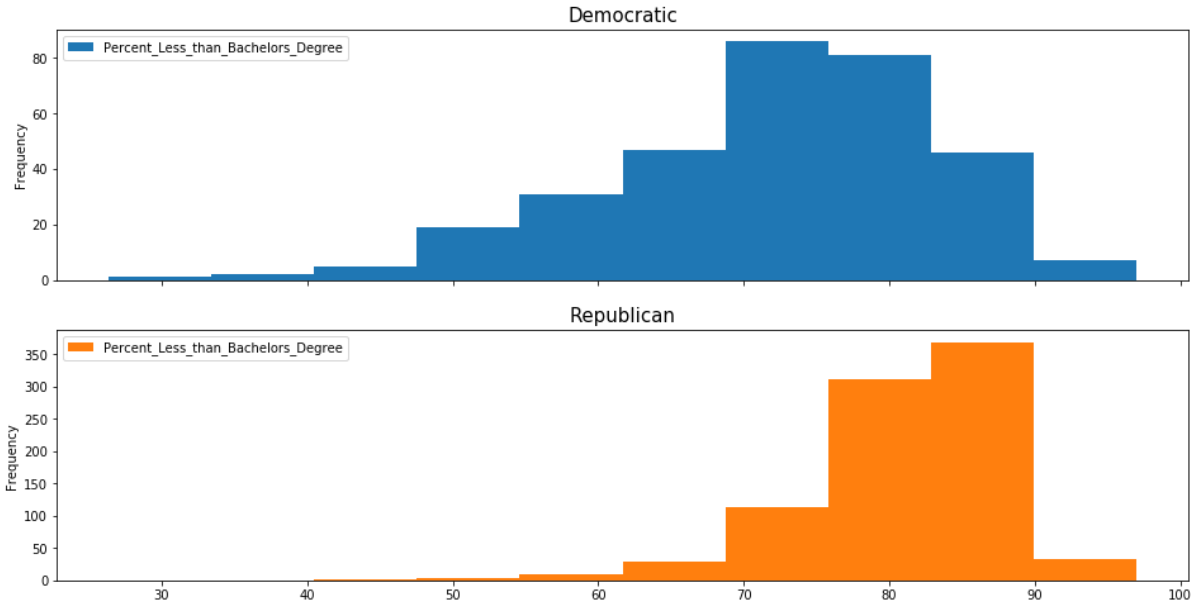
In [84]:
```
#Comparison of Education: Democratic Vs Republic
#Histogram Distribution for Percent Less than High School Degree
School_Degree = pd.DataFrame([df_democratic_t8['Percent_Less_than_High_School_Degree'],df_republican_t8['Percent_Less_than_High_School_Degree']])
# taking transpose
School_Degree = School_Degree.transpose()
# Creatinng a histograme
axes = School_Degree.plot.hist(rot=0, subplots=True,figsize = (16,8))
axes[1].legend(loc=2)
#subplot for Democratic
axes[0].set_title('Democratic',fontsize = 15)
# Subplot for Republic
axes[1].set_title('Republican',fontsize = 15)
```

Out[84]: Text(0.5, 1.0, 'Republican')



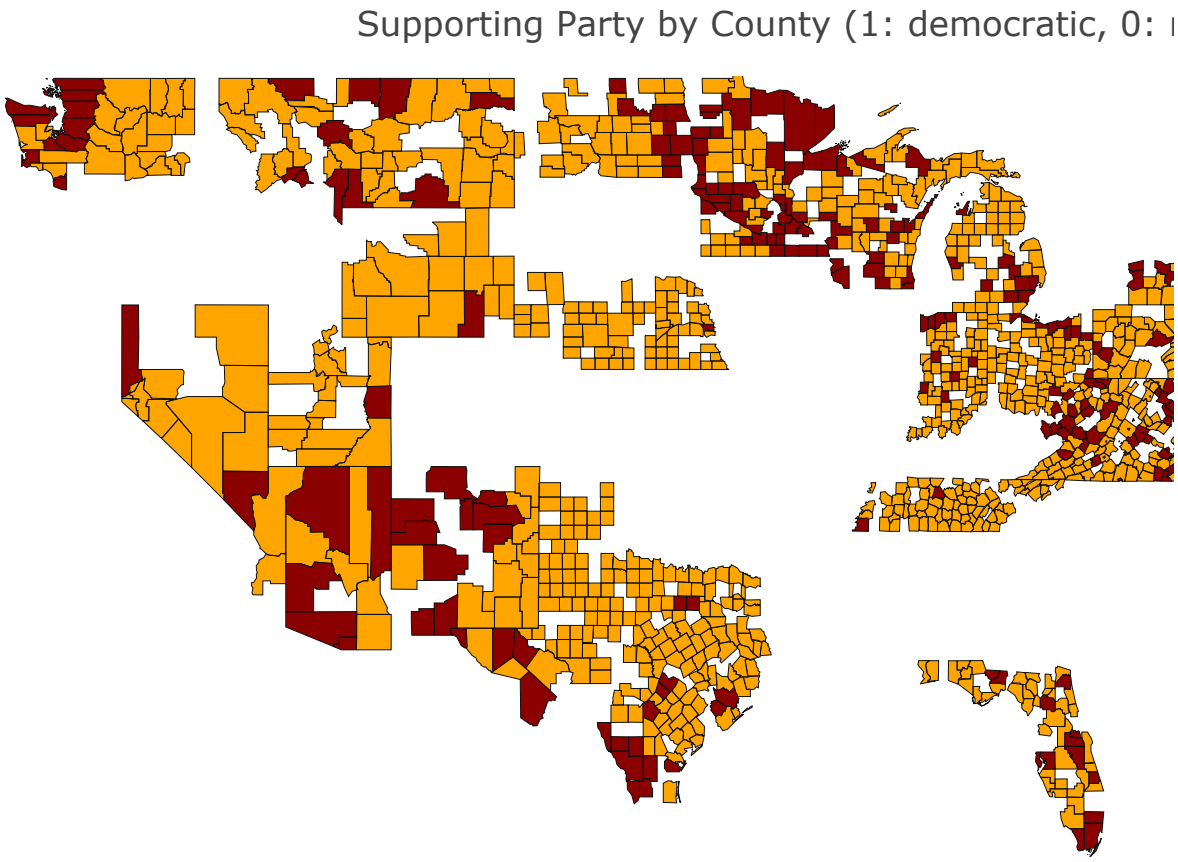
In [85]:
```
#Histogram Distribution for Percent Less than Bachelor's Degree
School_Degree = pd.DataFrame([df_democratic_t8["Percent_Less_than_Bachel
ors_Degree"],df_republican_t8["Percent_Less_than_Bachelors_Degree"]])
School_Degree = School_Degree.transpose()
# Creatinng a histograme
axes = School_Degree.plot.hist(rot=0, subplots=True,figsize = (16,8))
axes[1].legend(loc=2)
#subplot for Democratic
axes[0].set_title('Democratic',fontsize = 15)
# Subplot for Republic
axes[1].set_title('Republican',fontsize = 15)
```

Out[85]: Text(0.5, 1.0, 'Republican')



In [77]:
```
## Task10

import plotly.figure_factory as ff

import numpy as np
import pandas as pd

colorscale = ['#FFA500','#8B0000']

fips = data_merged['FIPS'].tolist()
values = data_merged['Party'].tolist()

fig = ff.create_choropleth(
    fips=fips, values=values,
    county_outline={'color': 'rgb(15,15,15)', 'width': 0.5},
    colorscale=colorscale,
    show_state_data=False,
    show_hover=True, centroid_marker={'opacity': 0},
    asp=2.9, title='Supporting Party by County (1: democratic, 0: republ
ican)',
    legend_title='County'
)

fig.layout.template = None

fig.show()
```

```
//anaconda3/envs/cs418/lib/python3.7/site-packages/pandas/core/frame.p
y:7123: 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=Tru
e'.
```



In [ ]:

In [ ]: