Project 3

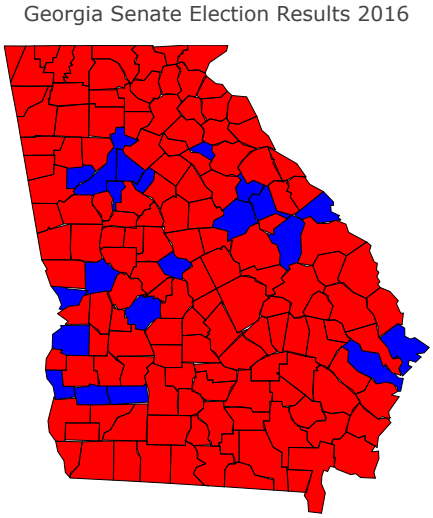
Mohammed Aljubori

This must be run before anything else

In [1]:
```
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly.plotly as py
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly import tools

import numpy as np
import pandas as pd
init_notebook_mode(connected=True)
```

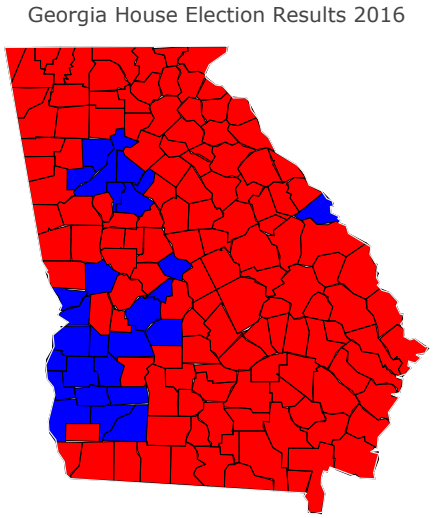
2a. Figure 1) A Choropleth map showing the 2016 county by county senate election results in the state of Georgia. The map should show the map of Georgia and only Georgia. i. For each county, compare the columns “demsen16”, “repsen16”, and “othersen16”. If “demsen16” has the highest number, color the county blue in the map. If “repsen16” has the highest number, color the county red in the map. If “othersen16” has the highest number, color the county white in the map. ii. The border of each county should be black.

In [16]:
```
df_sample = pd.read_csv(r'election-context-2018.csv')

#Georgia Rows
GA = df_sample.iloc[358:517]

#Checks what color is needed
#1=blue, 2=red, 3=white
def label_race (row):
    if (row['demsen16'] >= row['repsen16']) and (row['demsen16'] >= row['othersen16']) :
        return '1'
    elif  (row['repsen16'] >= row['demsen16']) and (row['repsen16'] >= row['othersen16']) :
        return '2'
    else:
        return '3'
    return 'Other'

#Applies it
GA['Color'] = GA.apply(label_race, axis=1)
#cleaner DataFrame
Votes = GA.groupby(['fips', 'demsen16', 'repsen16', 'othersen16', 'Color'], as_index=False)
nl = Votes.first()

#Figure
fips = nl['fips']
values = nl['Color']

colorscale = ['blue', 'red', 'white']
fig = ff.create_choropleth(
    fips=fips, values=values, scope=['Georgia'], show_state_data=False,
    colorscale=colorscale, round_legend_values=True, showlegend = False,
    county_outline={'color': 'black', 'width': 0.5},
    exponent_format=True,
    title='Georgia Senate Election Results 2016')
sort=True
iplot(fig)
```



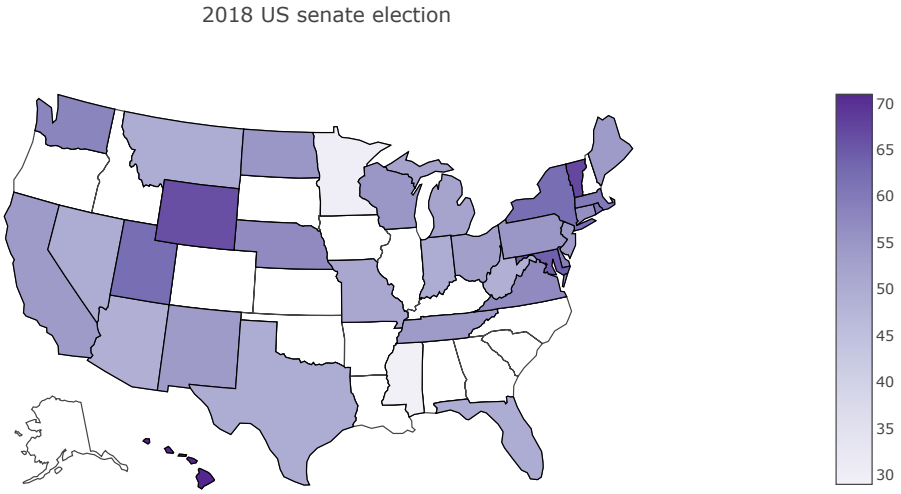
2b. Create a similar Choropleth map showing the 2016 county by county house election results in the state of Georgia, using the same color scheme as specified above.

In [17]:
```
df = pd.read_csv(r'election-context-2018.csv')

#Georgia Rows
house = df.iloc[358:517]

#Checks what color is needed
#1=blue, 2=red, 3=white
def label_race (row):
    if (row['demhouse16'] >= row['rephouse16']) and (row['demhouse16'] >= row['otherhouse16']) :
        return '1'
    elif  (row['rephouse16'] >= row['demhouse16']) and (row['rephouse16'] >= row['otherhouse16']) :
        return '2'
    else:
        return '3'
    return 'Other'

#Applies it
house['Color'] = house.apply(label_race, axis=1)
#cleaner DataFrame
h_votes = house.groupby(['fips', 'demhouse16', 'rephouse16', 'otherhouse16', 'Color'], as_index=False)
ll = h_votes.first()

#Figure
fips = ll['fips']
values = ll['Color']

colorscale = ['blue', 'red', 'white']
fig = ff.create_choropleth(
    fips=fips, values=values, scope=['Georgia'], show_state_data=True,
    colorscale=colorscale, round_legend_values=True,showlegend = False,
    county_outline={'color': 'black', 'width': 0.5},
    exponent_format=True,
    title='Georgia House Election Results 2016'
)
iplot(fig)
```



1. Create a Choropleth map for the 2018 US senate election. The map should show the entire United States. Each state should be color coded. The value for each state is the winning candidate's vote percentage. Divide the vote percentage into 6 bins and create a color scale for it. When the mouse cursor hovers over each state, the winning candidate's name and party affiliation should be displayed in the tooltip window. For example, Wyoming's 2018 winning candidate was John Barrasso, Repblication. His vote percentage was 136210/203420.

In [9]:
```
df = pd.read_csv(r'1976-2018-senate.csv', encoding = "ISO-8859-1")
#2018 only
_2018 = df.iloc[3269:3422]
#states_codes
state_codes=_2018.groupby('state').first()
state_codes1 = state_codes['state_po']
#Winner votes per state
winners_votes= _2018.groupby(['state'], sort=False, as_index=True)['candidatevotes'].max()
#total votes per state
total_votes= _2018.groupby(['state'], sort=False)['totalvotes'].max()
#actual avg per state, clean
final_avg = winners_votes/total_votes*100


#New DataSet with everyhing needed only
zzz = pd.merge(winners_votes,total_votes, left_index=True, right_index=True)

#6 bins for color
scl = [
    [0.0, 'rgb(242,240,247)'],#1
    [0.2, 'rgb(218,218,235)'],#2
    [0.4, 'rgb(188,189,220)'],#3
    [0.6, 'rgb(158,154,200)'],#4
    [0.8, 'rgb(117,107,177)'],#5
    [1.0, 'rgb(84,39,143)']#6
]



data = [go.Choropleth(
    colorscale = scl,
    autocolorscale = False,
    locations = state_codes1,
    text = 'Winner Percentage: '+ winners_votes.astype('str') + ' / '+ total_votes.astype('str')
    ,z = final_avg.astype('int'),
    locationmode = 'USA-states',
)]

layout = go.Layout(
    title = go.layout.Title(
        text = '2018 US senate election'),
    geo = go.layout.Geo(
        scope = 'usa',
        projection = go.layout.geo.Projection(type = 'albers usa'),
        showlakes = False)
)
fig = go.Figure(data = data, layout = layout)
iplot(fig, filename = 'd3-cloropleth-map')
```



1. Load 1962_2006_walmart_store_openings.csv and use Plotly to create the following map. The map must have a title and legend. a. (Figure 4) Create a Scattergeo map that shows the location of every Walmart store opened since 2000 (including 2000) in the United States. b. The map should show the entire United States. c. If it's a "Supercenter", use a dark blue color to fill the marker. If it's a "Wal-Mart", use a light blue color.

In [11]:
```
dz = pd.read_csv(r'1962_2006_walmart_store_openings.csv')
#Takes only years 2000+
wm = dz.iloc[2352:2993]
#checks if supercenter or not
def center_type (row):
    if (row['type_store'] == 'Supercenter'):
        return '1'
    else:
        return '2'
    return 'Other'
#applies it
wm['color'] = wm.apply(center_type, axis=1)

z1 = wm.groupby(['type_store','color','LAT', 'LON'], as_index=False).first()

scl = [ [0,'lightblue'], [1.0,'blue']]

data = [ go.Scattergeo(
        locationmode = 'USA-states',
        lon = z1['LON'],
        lat = z1['LAT'],
        text = z1['type_store'],
        mode = 'markers',
        marker = dict(
            size = 5,
            opacity = 0.8,
            reversescale = True,
            autocolorscale = False,
            symbol = 'circle',
            line = dict(
                width=.5,
                color='rgba(102, 102, 102)'
            ),

        ))]

layout = dict(
        title = 'Walmarts Stores<br>(Hover for Store Type)',
        geo = dict(
            scope='usa',
            projection=dict( type='albers usa' ),
            showland = True,
            landcolor = "rgb(250, 250, 250)",
            countrywidth = 0.5,
            subunitwidth = 0.5
        ),
    )

fig = go.Figure(data=data, layout=layout)
iplot(fig)
```

1. Load wimbledons_champions.csv and use Plotly to create the following map. The map must have a title and legend. a. (Figure 5) Create a Scattergeo map showing the number of champions for different countries. b. The map should show the entire world. c. Calculate how many Wimbledon champions each country has produced. d. Place a marker for each country that has produced a champion. Use the latitude and longitude of the capital of the country as the location. i. You may need to find the latitude and longitude for the capitals yourself. e. The size of the marker should be proportional to the number of champions this country has produced.

In [10]:

```
do = pd.read_csv(r'wimbledons_champions.csv')
cities = pd.DataFrame()
cities['wins'] = do.pivot_table(index=['Champion Nationality'], aggfunc='size')
#lat of cities
lat = ['-35.2809', '-15.8267', '45.8150', '50.0755', '40.4168',
       '48.8566','51.5074','52.5200','52.3680','-41.2865',
       '55.7558','44.7866','46.204391','59.3293','38.9072']
#Lon of cities
lon = ['149.1300', '-47.9218', '15.9819', '14.4378', '-3.7038',
       '2.3522','-0.1278','13.4050','4.9036','174.7762',
       '37.6173','20.4489','6.143158','18.0686','-77.0369',


      ]
cities['lat'] = lat
cities['lon'] = lon

insert = [
go.Scattergeo(
            lon = cities['lon'],
            lat = cities['lat'],
            text = cities['wins'],
            mode = 'markers',
            showlegend = False,
            marker = go.scattergeo.Marker(
                size = cities['wins'],
                color = 'red',
                sizemode='area',
                sizemin=1,
            )
    )
]
layout = go.Layout(
    title = go.layout.Title(text = 'Winner World Map'),
    geo = go.layout.Geo(
        resolution = 50,
        scope = 'world',
        showframe = False,
        showcoastlines = True,
        showland = True,
        landcolor = "rgb(229, 229, 229)",
        countrycolor = "rgb(255, 255, 255)" ,
        coastlinecolor = "rgb(255, 255, 255)",
        projection = go.layout.geo.Projection(
            type = 'eckert4'
        ),
    )
)

fig = go.Figure(layout=layout, data=insert)
iplot(fig)
```