

1. Load election\_context-2018.csv and use Plotly to create the following maps. Each map must have a title and legend.

a. (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”, “repesen16”, and “othersen16”. If “demsen16” has the highest number, color the county blue in the map. If “repesen16” 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.

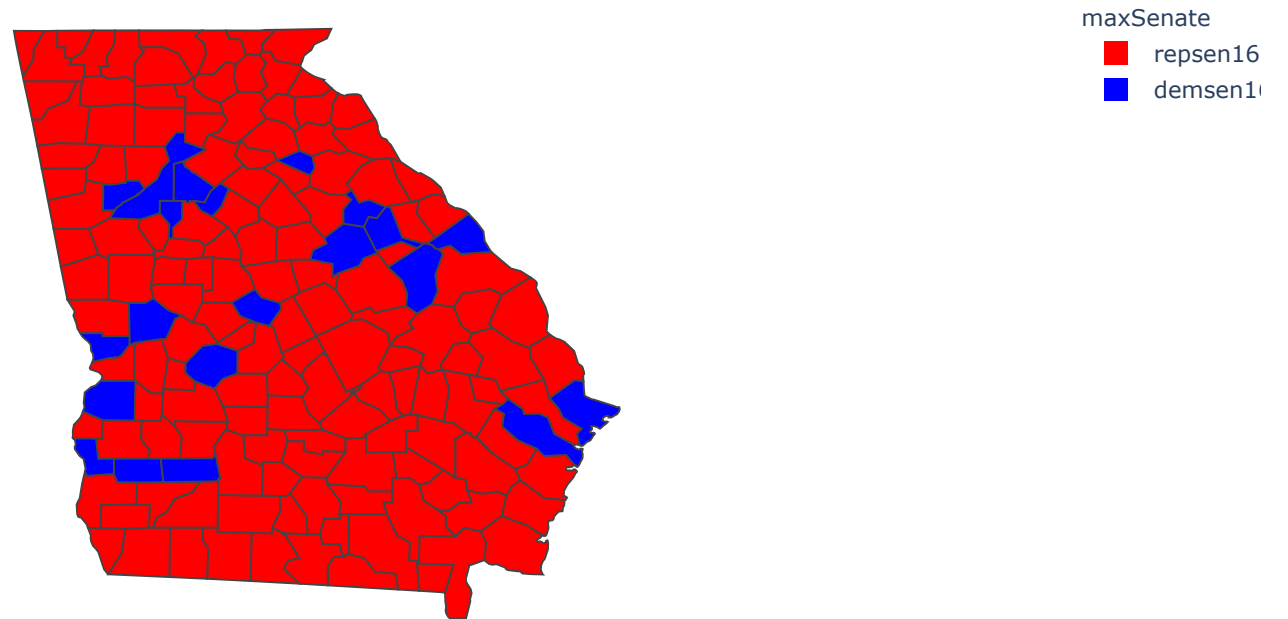
```
In [1]: import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: import pandas as pd
df3 = pd.read_csv('election-context-2018.csv')
#print(df3.head())
df3_ga = df3[df3.state.isin(['Georgia'])]
df3_ga['maxSenate'] = df3_ga[['demsen16', 'repesen16', 'othersen16']].idxmax(axis=1)
#print(df3_ga.head())
```

```
In [3]: import plotly as py
import plotly.express as px
from urllib.request import urlopen
import json
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)

fig_1 = px.choropleth(df3_ga, geojson=counties,
                      locations='fips',
                      color='maxSenate',
                      color_discrete_sequence=['red', 'blue', 'white'],
                      title='2016 Georgia Senate Election result')
fig_1.update_geos(fitbounds="locations", visible=False)
fig_1.show()
```

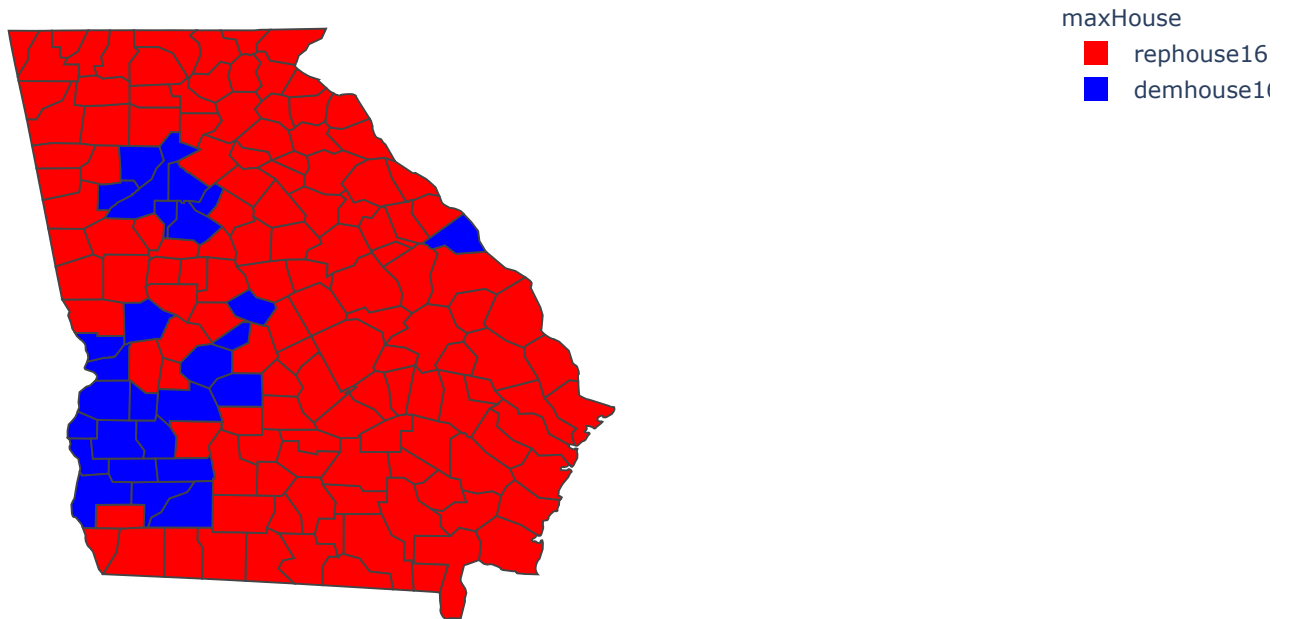
## 2016 Georgia Senate Election result



b. (Figure 2) 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 [4]: df3_ga['maxHouse'] = df3_ga[['demhouse16', 'rephouse16', 'otherhouse16']].idxmax(axis=1)
fig_2 = px.choropleth(df3_ga, geojson=counties,
                      locations='fips',
                      color='maxHouse',
                      color_discrete_sequence=['red', 'blue', 'white'],
                      title='2016 Georgia House Election result')
fig_2.update_geos(fitbounds="locations", visible=False)
fig_2.show()
```

## 2016 Georgia House Election result



1. Load 1976-2018-senate.csv and use Plotly to create the following map. Every figure must have a title and a legend.

a. (Figure 3) 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, Republican. His vote percentage was 136210/203420.

```

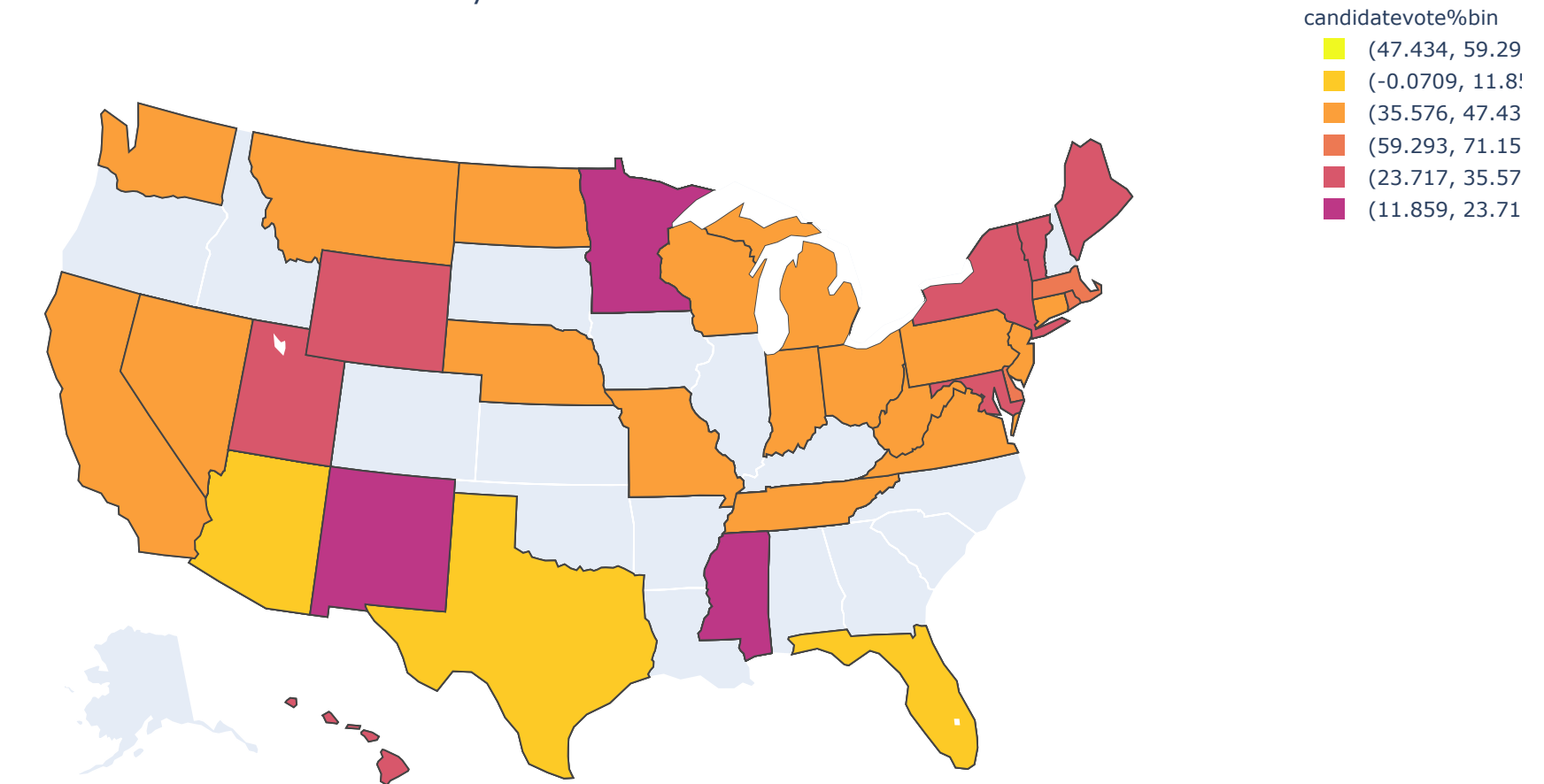
In [6]: import plotly.express as px
df4 = pd.read_csv('1976-2018-senate.csv', encoding = "unicode_escape")
#print(df4.head())
df4_2018 = df4[df4.year.isin([2018])]
#print(df4_2018.head())
df4_2018['candidatevote%']=df4_2018['candidatevotes']/df4_2018['totalvotes']*100
#print(df4_2018['candidatevote%'].describe())
#Divide the vote percentage into 6 equal width bins
df4_2018['candidatevote%bin']=pd.cut(df4_2018['candidatevote%'], 6)

for col in df4_2018.columns:
    df4_2018[col] = df4_2018[col].astype(str)

df4_2018['text'] = df4_2018['state'] + "'s 2018 winning candidate was " + '<br>' + \
    df4_2018['candidate'] + ", " + df4_2018["party"] + "." + '<br>' + \
    'His vote percentage was ' + df4_2018["candidatevotes"] + "/" + df4_2018["totalvotes"] + ". "

fig3 = px.choropleth(df4_2018, locations='state_po', color='candidatevote%bin',
                    color_discrete_sequence=px.colors.sequential.Plasma_r,
                    hover_name='text',
                    scope="usa",
                    locationmode="USA-states",
                    title = '2018 US Candidate votes result by State')
fig3.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig3.show()

```



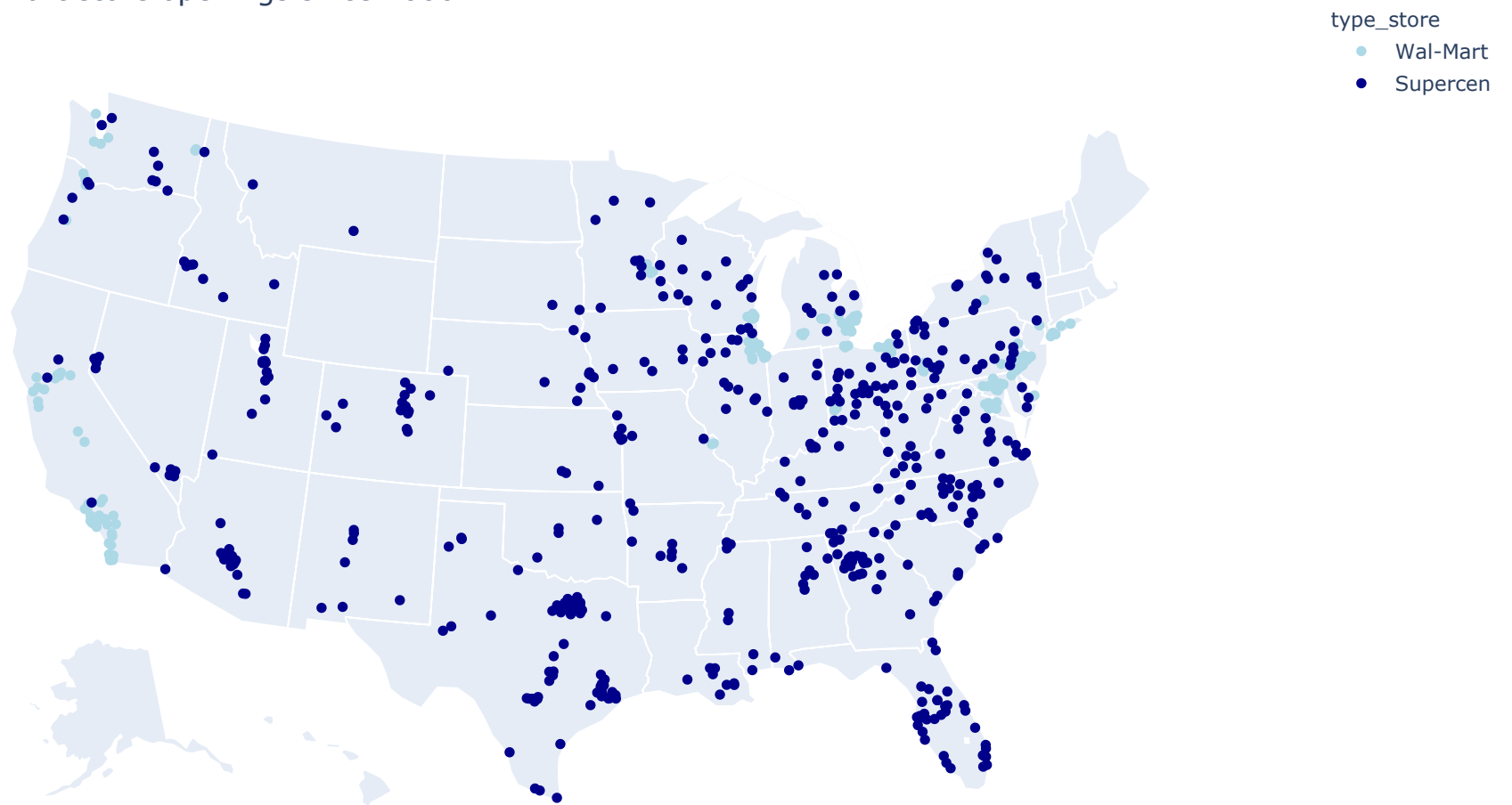
1. Load 1962\_2006\_walmart\_store\_openings.csv and use Plotly to create the following map. The map must have a title and legend.
  - a. (Figure 4) Create a Scattergeo map that shows the location of every Walmart store opened since 2000 (including 2000) in the United States.
  - b. The map should show 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 [7]: df5=pd.read_csv("1962_2006_walmart_store_openings.csv")
        #print(df5.head())
        df5_2000=df5[df5['YEAR']>1999]

        fig4 = px.scatter_geo(df5_2000, lat='LAT', lon='LON', color="type_store",
                               locationmode="USA-states",
                               color_discrete_sequence=['lightblue','darkblue'],
                               scope="usa",
                               title = 'Walmart store openings since 2000')
        fig4.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
        fig4.show()

```

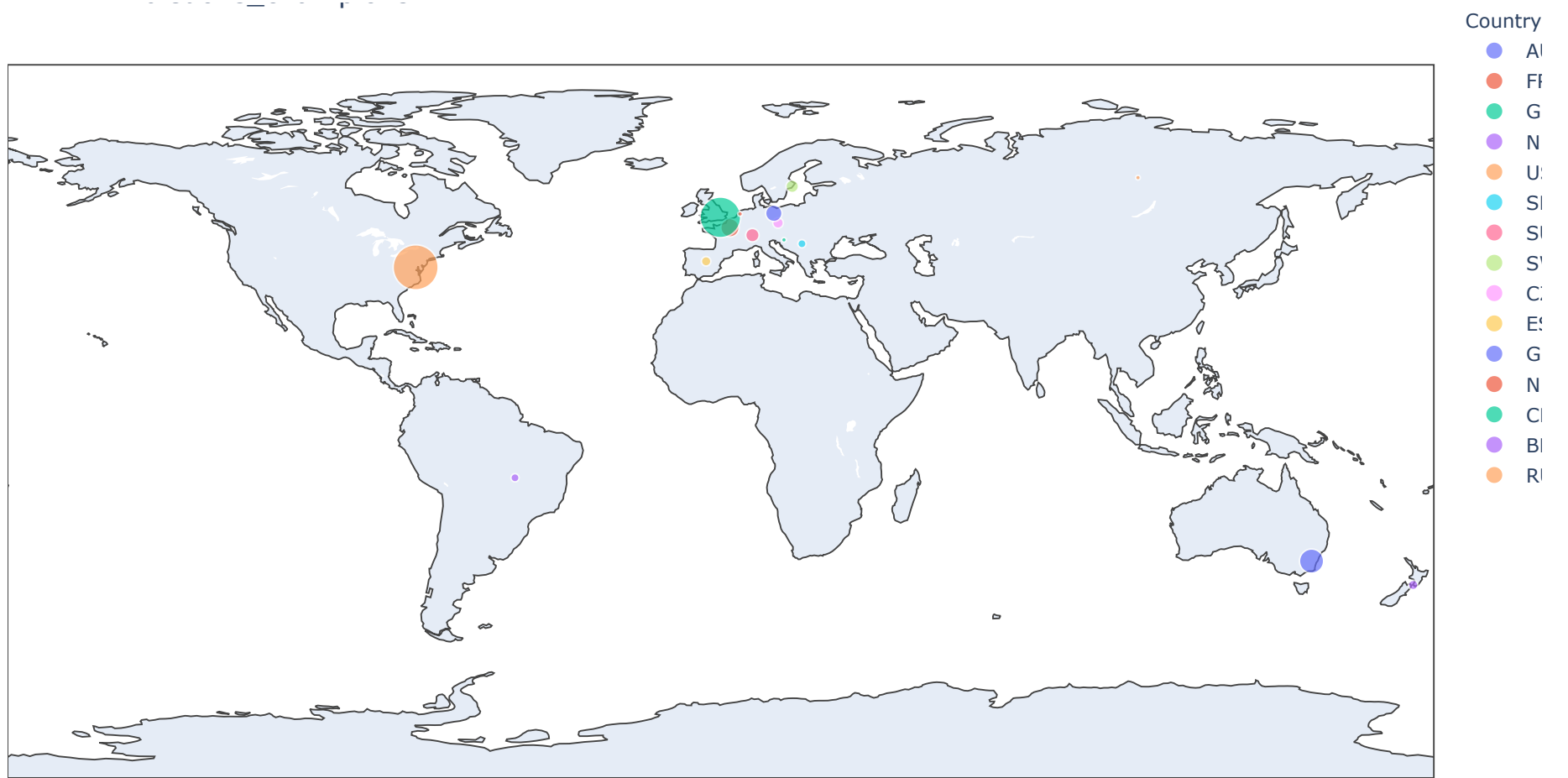




1. Load `wimbledons_champions.csv` and use Google Maps Services Python client to create the following map.
  - a. (Figure 5) Create a map showing the number of champions for different countries.
  - b. The map should show the entire world.
  - c. Calculate how many Wimbledon champions each country has 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 will 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 [8]: df6=pd.read_csv('wimbledons_champions.csv')
        from collections import Counter
        import numpy as np
        counter=Counter(df6["Champion Nationality"])
        kk=[list(counter.keys()),list(counter.values())]
        df6_count = pd.DataFrame(np.array(kk).T, columns=['Country','Count'])
        #print(df6_count)
        df6_count['lat']=[-35.26666667,48.86666667,51.5,-41.3,38.883333,44.83333333,47,59.33333333,50.08333333,
                           40.4,52.5200,52.3667,45.815399,-14.2350,61.5240]
        df6_count['lon']=[149.133333,2.333333,-0.083333,174.783333,-77,20.5,8,18.05,14.466667,-3.683333,13.4050,
                           4.8945,15.966568,-51.9253,105.3188]

        fig5 = px.scatter_geo(df6_count, lat='lat',lon='lon',
                               color="Country",
                               size=df6_count['Count'].astype(float),
                               title = 'wimbledons_champions')
        fig5.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
        fig5.show()
```



In [ ]: