```
1 Choropleth Map Exercise
```

In [1]:
```
1 # Plotly import
2 import plotly.graph_objs as go
3 from plotly.offline import init_notebook_mode,iplot
4 init_notebook_mode(connected=True)
```

Import pandas and read the csv file: 2014_World_Power_Consumption

In [2]:
```
1 import pandas as pd
2 df = pd.read_csv('2014_World_Power_Consumption')
3 df.head()
```

Out[2]:

	Country	Power Consumption KWH	Text
0	China	5.523000e+12	China 5,523,000,000,000
1	United States	3.832000e+12	United 3,832,000,000,000
2	European	2.771000e+12	European 2,771,000,000,000
3	Russia	1.065000e+12	Russia 1,065,000,000,000
4	Japan	9.210000e+11	Japan 921,000,000,000

In [3]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 219 entries, 0 to 218
Data columns (total 3 columns):
Country                   219 non-null object
Power Consumption KWH     219 non-null float64
Text                      219 non-null object
dtypes: float64(1), object(2)
memory usage: 5.2+ KB
```

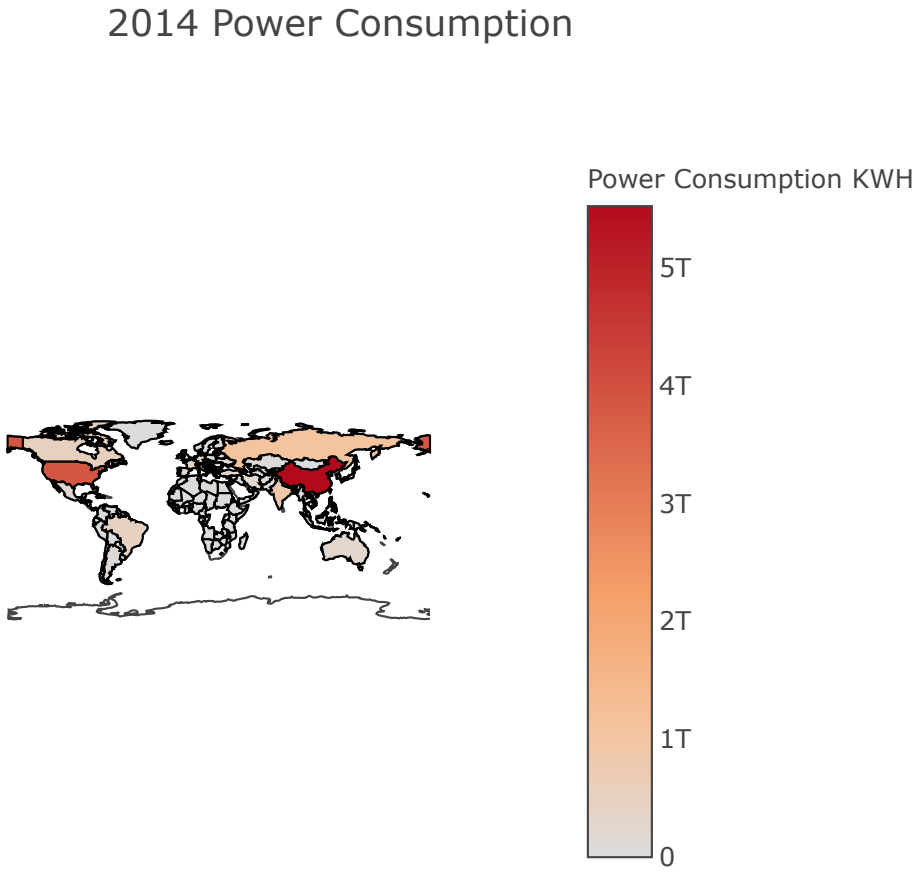
Creating a Choropleth Plot of the Power Consumption for Countries using the data and layout dictionary.

In [4]:
```
#data
data = dict(type = 'choropleth',
            locations = df['Country'],
            locationmode='country names',
            z = df['Power Consumption KWH'],
            text = df['Country'],
            colorbar = {'title': 'Power Consumption KWH'}
            )
```

In [5]:
```
#layout
layout = dict(title = '2014 Power Consumption',
              geo = dict(showframe = False,
                         projection = {'type': 'Mercator'}))


```

In [6]:
```
1 choromap = go.Figure(data = [data],layout = layout)
2 iplot(choromap,validate=False)
```



Export to plot.ly »

```
1 USA Choropleth
```

In [8]:
```
#Import 2012_Election_data csv using pandas
df = pd.read_csv('2012_Election_Data')
df.head()
```

Out[8]:

	Year	ICPSR State Code	Alphanumeric State Code	State	VEP Total Ballots Counted	VEP Highest Office	VAP Highest Office	Total Ballots Counted	Highest Office	Po
0	2012	41	1	Alabama	NaN	58.6%	56.0%	NaN	2,074,338	3,
1	2012	81	2	Alaska	58.9%	58.7%	55.3%	301,694	300,495	
2	2012	61	3	Arizona	53.0%	52.6%	46.5%	2,323,579	2,306,559	4,
3	2012	42	4	Arkansas	51.1%	50.7%	47.7%	1,078,548	1,069,468	2,
4	2012	71	5	California	55.7%	55.1%	45.1%	13,202,158	13,038,547	23,

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 17 columns):
Year                                51 non-null int64
ICPSR State Code                    51 non-null int64
Alphanumeric State Code             51 non-null int64
State                               51 non-null object
VEP Total Ballots Counted           41 non-null object
VEP Highest Office                  51 non-null object
VAP Highest Office                  51 non-null object
Total Ballots Counted               41 non-null object
Highest Office                      51 non-null object
Voting-Eligible Population (VEP)    51 non-null object
Voting-Age Population (VAP)         51 non-null float64
% Non-citizen                       51 non-null object
Prison                              51 non-null object
Probation                           51 non-null object
Parole                              51 non-null object
Total Ineligible Felon              51 non-null object
State Abv                           51 non-null object
dtypes: float64(1), int64(3), object(13)
memory usage: 6.9+ KB
```

```
** Creating a plot that displays the Voting-Age Population (VAP)
per state.
Note: VAP has already been transformed to a float **
```

In [11]:

```
1
2  data = dict(type = 'choropleth',
3              locations = df['State Abv'],
4              colorscale = 'Viridis',
5              reversescale = True,
6              z = df['Voting-Age Population (VAP)'],
7              locationmode = 'USA-states',
8              text = df['State'],
9              marker = dict(line = dict(color = 'rgb(255,255,255)',
10                                       width = 2)),
11            colorbar = {'title': 'Voting-Age Popultaion'})
12
13 layout = dict(title = '2012 Election Data',
14              geo = dict(scope = 'usa',
15                         showlakes = True,
16                         lakecolor = 'rgb(85, 173, 240)')
17              )
```

In [12]:
```
1 choromap = go.Figure(data = [data],layout = layout)
2 iplot(choromap,validate=False)
```