```python
In [1]:  import json
         geo = json.load(open('SIG.geojson', encoding = 'UTF-8'))
```

```python
In [2]:  geo['features'][0]['properties']
```

```
Out[2]:  {'SIG_CD': '42110', 'SIG_ENG_NM': 'Chuncheon-si', 'SIG_KOR_NM': '춘천시'}
```

```python
In [27]:  # geo['features'][0]['geometry']
```

```python
In [4]:  import pandas as pd
         df_pop = pd.read_csv('Population_SIG.csv')
         df_pop.head()
```

Out[4]:

	code	region	pop
0	11	서울특별시	9509458
1	11110	종로구	144683
2	11140	중구	122499
3	11170	용산구	222953
4	11200	성동구	285990

```python
In [5]:  df_pop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 278 entries, 0 to 277
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   code    278 non-null    int64
 1   region  278 non-null    object
 2   pop     278 non-null    int64
dtypes: int64(2), object(1)
memory usage: 6.6+ KB
```

```python
In [6]:  df_pop['code'] = df_pop['code'].astype(str)
```

```python
In [7]:  import folium
         folium.Map(location = [35.95, 127.7],
                    zoom_start = 6,
                    width = '50%',
                    height = '80%')
```

Out[7]: Make this Notebook Trusted to load map: File -> Trust Notebook

```python
In [8]:  map_sig = folium.Map(location = [35.95, 127.7],
                              zoom_start = 6,
                              width = '50%',
                              height = '80%',
                              tiles = 'cartodbpositron')
         map_sig
```

Out[8]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [9]:
```python
folium.Choropleth(geo_data = geo,
                  data = df_pop,
                  columns = ('code', 'pop'),
                  key_on = 'feature.properties.SIG_CD') \
        .add_to(map_sig)
```

Out[9]: <folium.features.Choropleth at 0x7fc7cbaafa30>

In [10]:
```python
map_sig
```

Out[10]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [11]:
```python
bins = list(df_pop['pop'].quantile([0, 0.2, 0.4, 0.6, 0.8, 1]))
bins
```

Out[11]: [8867.0, 50539.6, 142382.20000000004, 266978.6, 423107.20000000024, 13565450.0]

In [12]:
```python
map_sig = folium.Map(location = [35.95, 127.7],
                     zoom_start = 6,
                     width = '50%',
                     height = '80%',
                     tiles = 'cartodbpositron')

folium.Choropleth(geo_data = geo,
```

```
                    data = df_pop,
                    columns = ('code', 'pop'),
                    key_on = 'feature.properties.SIG_CD',
                    fill_color = 'YlGnBu',
                    fill_opacity = 1,
                    line_opacity = 0.5,
                    bins = bins) \
         .add_to(map_sig)
map_sig
```

Out[12]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [13]:
```
import json
geo_seoul = json.load(open('EMD_Seoul.geojson', encoding = 'UTF-8'))
```

In [14]:
```
geo_seoul['features'][0]['properties']
```

Out[14]:
```
{'BASE_DATE': '20200630',
 'ADM_DR_CD': '1101053',
 'ADM_DR_NM': '사직동',
 'OBJECTID': '1'}
```

In [25]:
```
# geo_seoul['features'][0]['geometry']
```

In [16]:
```
foreigner = pd.read_csv('Foreigner_EMD_Seoul.csv')
foreigner.head()
```

Out[16]:

	code	region	pop
0	1101053	사직동	418.0
1	1101054	삼청동	112.0
2	1101055	부암동	458.0
3	1101056	평창동	429.0
4	1101057	무악동	102.0

In [17]:
```
foreigner.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3490 entries, 0 to 3489
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   code    3490 non-null   int64
 1   region  3490 non-null   object
 2   pop     3486 non-null   float64
dtypes: float64(1), int64(1), object(1)
memory usage: 81.9+ KB
```

In [18]:
```
foreigner['code'] = foreigner['code'].astype(str)
```

In [19]:
```
bins = list(foreigner['pop'].quantile([0, 0.2, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]))
bins
```

Out[19]: [7.0, 98.0, 200.0, 280.0, 386.0, 529.5, 766.0, 1355.5, 26896.0]

```
In [20]: map_seoul = folium.Map(location = [37.56, 127],
                                 zoom_start = 10,
                                  width = '60%',
                                  height = '60%',
                                 tiles = 'cartodbpositron')

         folium.Choropleth(geo_data = geo_seoul,
                           data = foreigner,
                           columns = ('code', 'pop'),
                           key_on = 'feature.properties.ADM_DR_CD',
                           fill_color = 'Blues',
                           nan_fill_color = 'White',
                           fill_opacity = 1,
                           line_opacity = 0.5,
                           bins = bins) \
                 .add_to(map_seoul)
         map_seoul
```

Out[20]: Make this Notebook Trusted to load map: File -> Trust Notebook

```
In [21]: geo_seoul_sig = json.load(open('SIG_Seoul.geojson', encoding = 'UTF-8'))
```

```
In [22]: folium.Choropleth(geo_data = geo_seoul_sig,
                           fill_opacity = 0,
                           line_weight = 4) \
                 .add_to(map_seoul)
         map_seoul
```

Out[22]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [23]:
```python
map_seoul.save('map_seoul.html')
```

In [24]:
```python
import webbrowser
webbrowser.open_new('map_seoul.html')
```

Out[24]: True