In [1]:

```
import geopandas

# Pull in example dataset of NYC Boroughs
path_to_data = geopandas.datasets.get_path("nybb")
gdf = geopandas.read_file(path_to_data)

gdf
```

Out[1]:

	BoroCode	BoroName	Shape_Leng	Shape_Area	geometry
0	5	Staten Island	330470.010332	1.623820e+09	MULTIPOLYGON (((970217.022 145643.332, 970227....
1	4	Queens	896344.047763	3.045213e+09	MULTIPOLYGON (((1029606.077 156073.814, 102957...
2	3	Brooklyn	741080.523166	1.937479e+09	MULTIPOLYGON (((1021176.479 151374.797, 102100...
3	1	Manhattan	359299.096471	6.364715e+08	MULTIPOLYGON (((981219.056 188655.316, 980940....
4	2	Bronx	464392.991824	1.186925e+09	MULTIPOLYGON (((1012821.806 229228.265, 101278...

In [2]:

```
geopandas.datasets.available
```

Out[2]:

```
['naturalearth_cities', 'naturalearth_lowres', 'nybb']
```

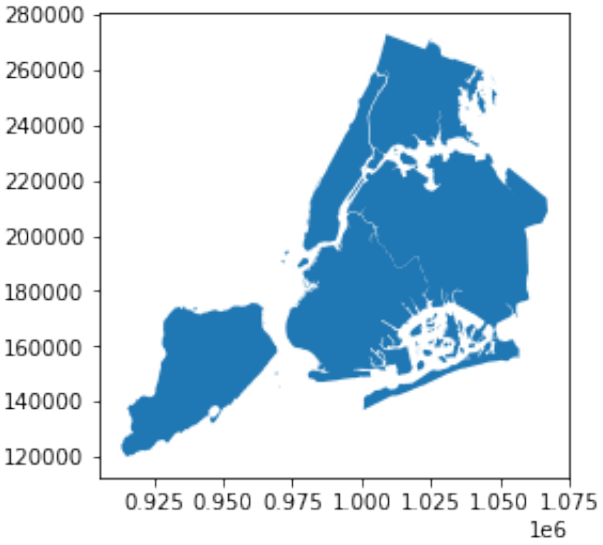
In [3]:

```
# Writing files (defaults to shapefile but can be changed, e.g., driver="GeoJ
gdf.to_file("Data/myfile.shp")
gdf.to_file("Data/my_file.geojson", driver="GeoJSON")
```

Making Maps

In [4]:

```
gdf.plot();
```



Adding data columns

In [5]:

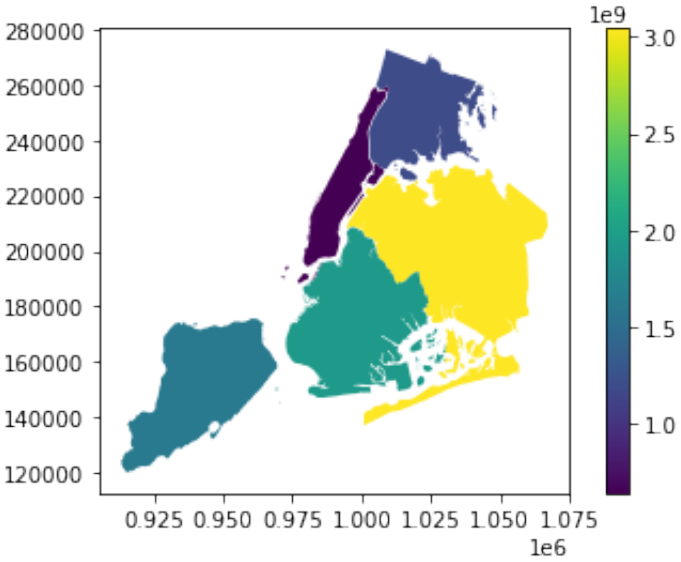
```
gdf = gdf.set_index("BoroName")

gdf["area"] = gdf.area
gdf["area"]
```

Out[5]:

```
BoroName
Staten Island    1.623822e+09
Queens           3.045214e+09
Brooklyn         1.937478e+09
Manhattan        6.364712e+08
Bronx            1.186926e+09
Name: area, dtype: float64
```

In [6]:

```
gdf.plot("area", legend=True);
```



In [7]:
```
# Interactive maps
gdf.explore("area", legend=False)
```

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

Read in CSV file with latitude/longitude coordinates

In [8]:
```
import pandas as pd

ft = pd.read_csv("Data/freedom_trail.csv")
ft.head()
```

Out[8]:

	id	name	address	city	state	zip
0	1	Boston Common	139 Tremont St	Boston	MA	2111
1	2	Massachusetts State House	24 Beacon St	Boston	MA	2133
2	3	Park Street Church	121 Tremont St	Boston	MA	2108
3	4	Granary Burying Ground	83 Tremont St	Boston	MA	2108
4	5	King’s Chapel/King’s Chapel Burying Ground	58 Tremont St	Boston	MA	2108

In [9]:
```
# Load in CSV dataset with geocoded point coordinates
go = pd.read_csv("Data/geocode_output.csv")

go = go.drop(columns=["Unnamed: 0"])
go.head()
```

Out[9]:

	id	name	address	city	state	zip	latitude	longitude
0	1	Boston Common	139 Tremont St	Boston	MA	2111	42.354274	-71.063835
1	2	Massachusetts State House	24 Beacon St	Boston	MA	2133	42.359244	-71.063870
2	3	Park Street Church	121 Tremont St	Boston	MA	2108	42.356641	-71.061984
3	4	Granary Burying Ground	83 Tremont St	Boston	MA	2108	42.357599	-71.060977
4	5	King's Chapel/King's Chapel Burying Ground	58 Tremont St	Boston	MA	2108	42.358013	-71.060027

In [10]:

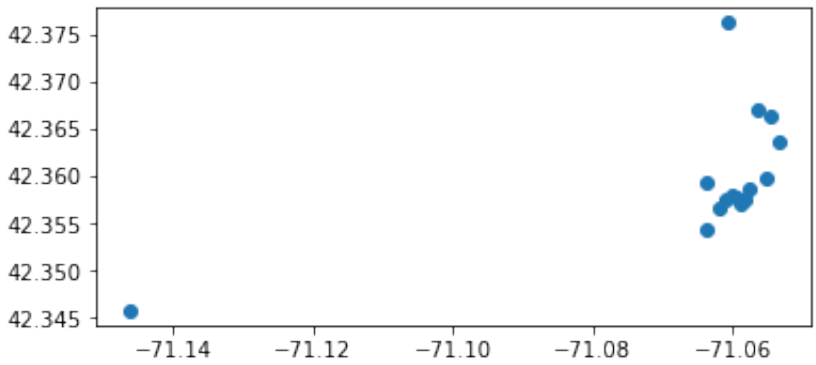
```
# Read in latitude/longitude into a GeoPandas dataframe
# crs - optional input for coordinate system, EPSG:4326 = WGS 1984
gdf2 = geopandas.GeoDataFrame(geometry=geopandas.points_from_xy(go.longitude,
```

In [11]:

```
gdf2.plot()
```

Out[11]:

```
<AxesSubplot:>
```



In [12]:

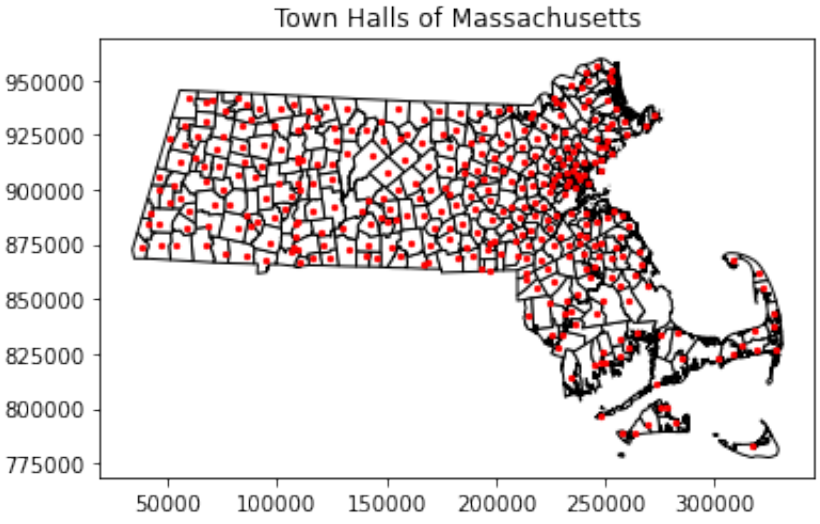
```
# Plot coordinates and exclude empty geometries (there's a couple of no-hits
# ["OpenStreetMap", "Stamen Terrain", "Stamen Toner", "Stamen Watercolor", "C
gdf2.loc[~gdf2.geometry.is_empty, :].explore(
    tiles="OpenStreetMap", cmap=["Blues"], popup=True, marker_kwds={"radius":
    highlight=True, legend=True, legend_kwds={"caption":"A test caption", "co
```

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

Static, Multi-Layered Maps (To Export as PNGs)

Create a map of the city/towns of Massachusetts

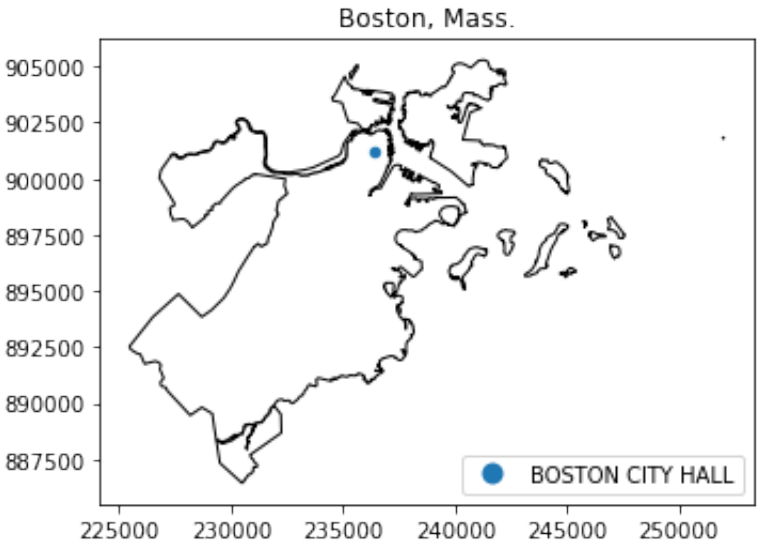
In [13]:
```
mass_towns = geopandas.read_file("Data/townssurvey_shp/TOWNSSURVEY_POLY.shp")
town_halls = geopandas.read_file("Data/townhalls_pt/TOWNHALLS_PT_MEMA.shp")

# Set background layer and axis information
ax = mass_towns.plot(color="white", edgecolor="black")
ax.set_title("Town Halls of Massachusetts")

town_halls.plot(ax=ax, color="red", markersize=5);
# plt.show()
```



Use a query to localize Boston only

In [14]:

```
boston = mass_towns.query('TOWN == "BOSTON"')
boston_th = town_halls.query('CITY == "BOSTON"')

ax = boston.plot(color="white", edgecolor="black")
ax.set_title("Boston, Mass.")

boston_th.plot(ax=ax, markersize=20, column="NAME", legend=True, legend_kwds=
```



In [15]:

```
mass_towns.head()
```

Out[15]:

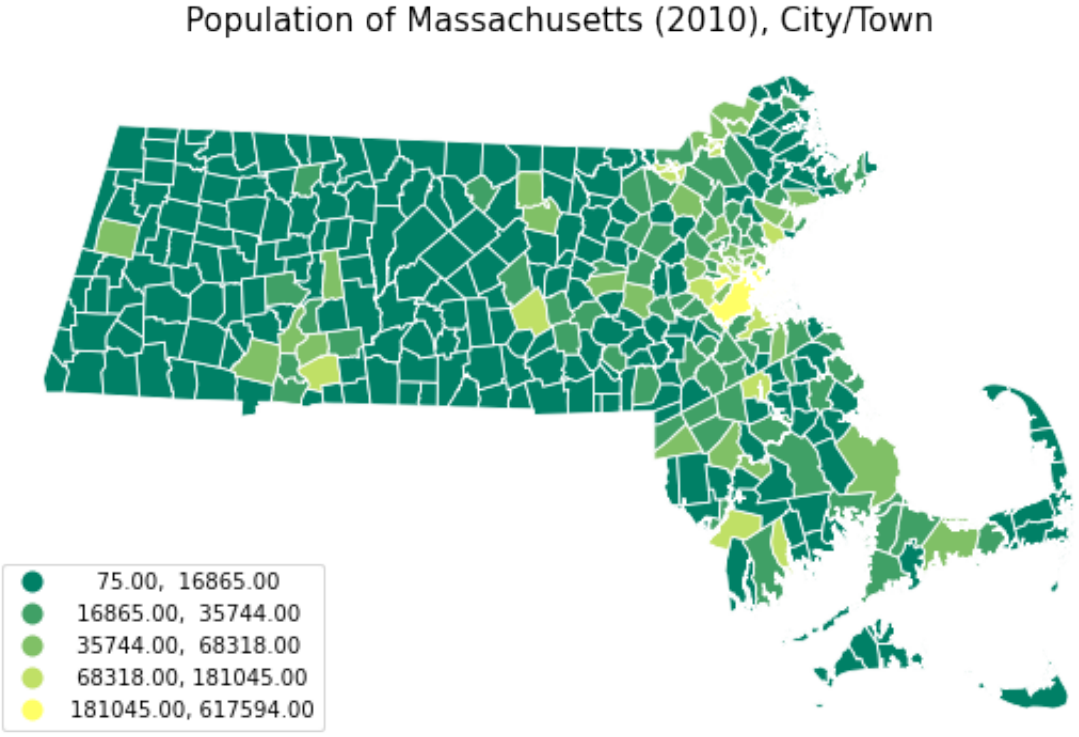
	TOWN	TOWN_ID	POP1980	POP1990	POP2000	POPCH90_00	TYPE	ISLAND	COA
0	WESTPORT	334	13861	13852	14183	331	T	1	
1	FALMOUTH	96	25007	27960	32660	4700	T	1	
2	WESTPORT	334	13861	13852	14183	331	T	1	
3	FALMOUTH	96	25007	27960	32660	4700	T	1	
4	DARTMOUTH	72	24840	27244	30666	3422	T	1	

5 rows × 23 columns

Choropleth Map of Population in Massachusetts

In [16]:

```
import geopandas as gpd
import matplotlib.pyplot as plt
import matplotlib.colors as colors

fig, ax = plt.subplots(1, figsize=(10, 6))
ax.axis('off')
ax.set_title("Population of Massachusetts (2010), City/Town", fontdict={'font

mass_towns.plot(column="POP2010", cmap="summer", scheme='natural_breaks', lin
                missing_kwds={"color": "lightgrey", "label": "Missing values"
                legend_kwds={"loc": "lower left"});
```

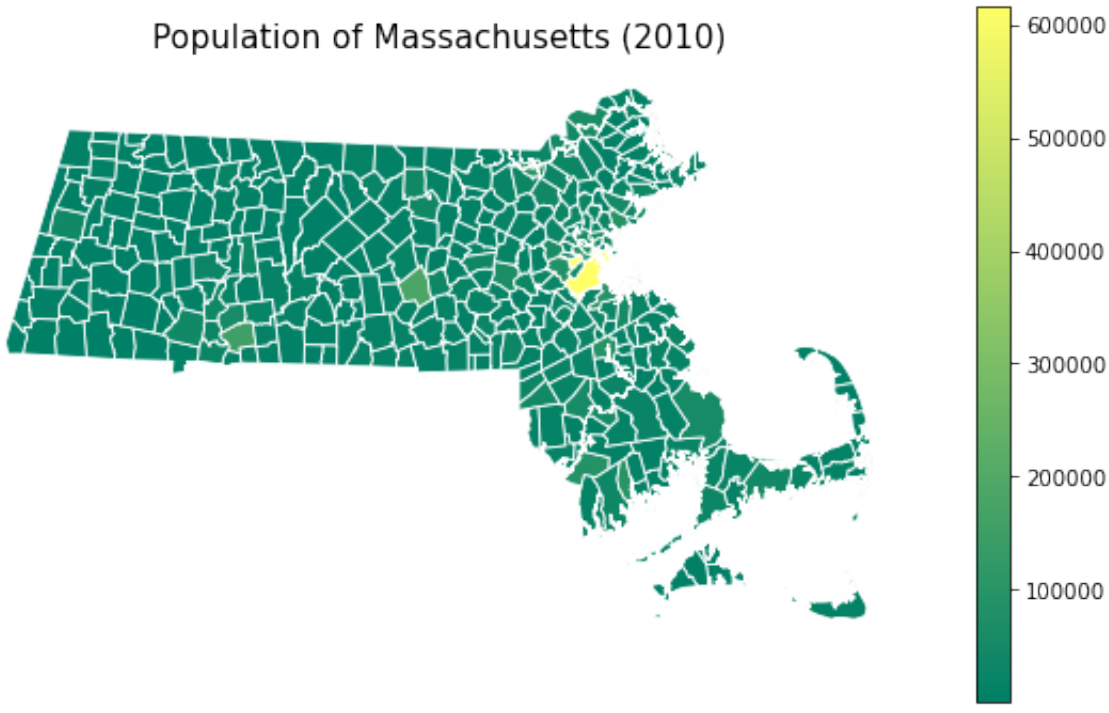


In [17]:

```
fig.savefig("mass_population_2010.png", dpi=300)
```

In [18]:

```
fig, ax = plt.subplots(1, figsize=(10, 6))
ax.axis('off')
ax.set_title("Population of Massachusetts (2010)", fontdict={'fontsize': '15'

mass_towns.plot(column="POP2010", cmap="summer", linewidth=0.9, ax=ax, edgeco
                missing_kwds={"color": "lightgrey", "label": "Missing values"
```



In [19]:

```
fig.savefig("mass_population_2020_cmap.png", dpi=300)
```