Mapping

May 20, 2020

#
Mapping Parameter Coefficients
Notebook Outline:
An example of hedonic house price modeling using MGWR - Section 1 - Section 2 - Section **??** - Section 4
Back to the main page

0.0.1 If you want to follow along with the code, follow this link

1 Mask the Significant Coefficients

```
In [ ]: b_cols = ['beta_Intercept', 'beta_avg_age','beta_avg_water_dist', 'beta_avg_sqft', 'bet
        bt_cols = ['bt_constant','bt_age','bt_water_dist','bt_sqft','bt_round_basement','bt_inc
        t_cols = ['t_Intercept','t_avg_age', 't_avg_water_dist', 't_avg_sqft', 't_avg_basement
        t_crit = [2.92,2.94,2.67,2.99,3.01,2.14,2.20,2.41]
```

```
In [ ]: for i in range(8):
            census.loc[census[t_cols[i]] >=t_crit[i], bt_cols[i]] = census[b_cols[i]]
            census.loc[census[t_cols[i]] <=-t_crit[i], bt_cols[i]] = census[b_cols[i]]
```

2 Spatial Join - Results to Shapefile

```
In [ ]: c='census_tracts/census_tracts.shp'
        crs = {'EPSG':'4326'}
        geo = gpd.read_file(c,crs=crs)[['geometry','objectid']]
        fig,ax = plt.subplots(figsize=(20,15))
        geo.plot(ax=ax)

        coords = np.array(list(zip(census['x_coor'],census['y_coor'])))
        geom_points = [Point(xy) for xy in coords]
        geo_df = gpd.GeoDataFrame(census,crs={'init':'epsg:4326'},geometry=geom_points)
        geo_df = geo_df.rename(columns={'OBJECTID':'index'})
        geo_df = geo_df.to_crs(epsg=3857)
        final_geo = gpd.sjoin(geo, geo_df, how='inner',op='contains',lsuffix='left',rsuffix='r

        fig,ax = plt.subplots(figsize=(20,15))
        ax.set_facecolor('white')
```

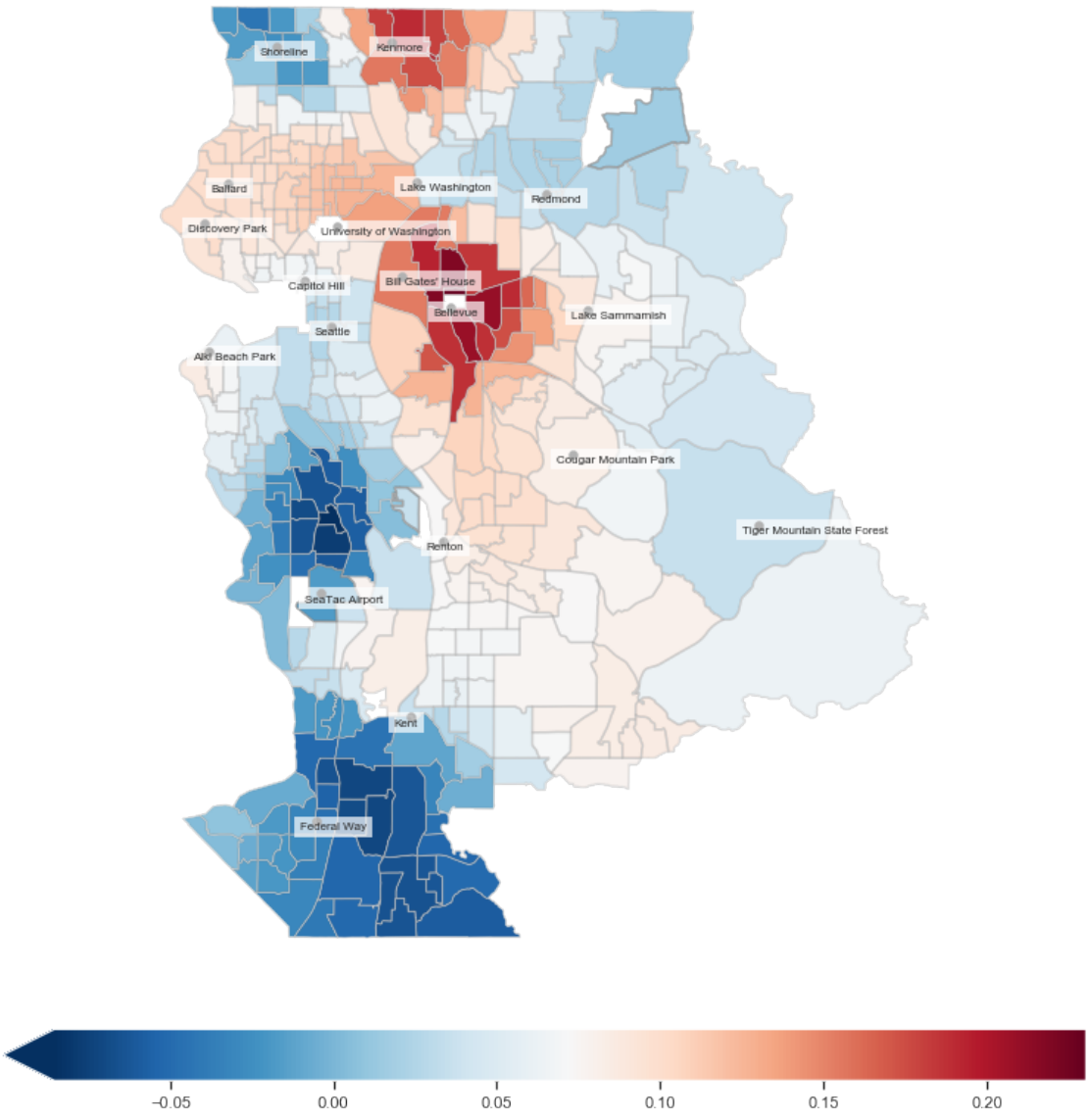
```
final_geo.plot(ax=ax, color='gold')
geo_df.plot(ax=ax, markersize=8,alpha=1,color='tomato',marker="o")
```

3 Map of all Parameter Coefficients

```
In [1]: import mapping_results as maps
```

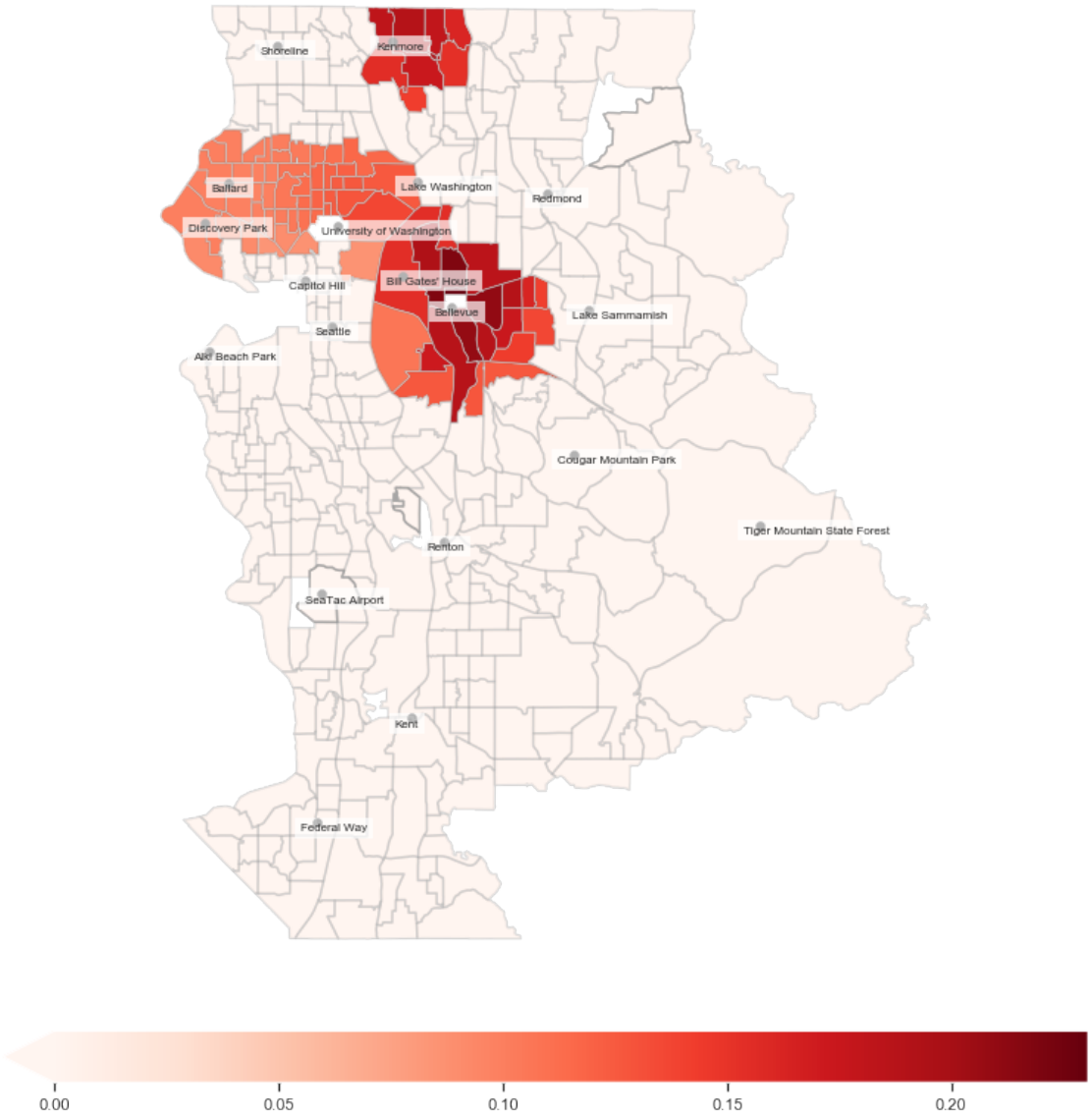
3.0.1 All parameter estimates for age covariate

```
In [7]: maps.mapp(name='beta_avg_age',color='RdBu_r',filename="b_age",normal=False)
```

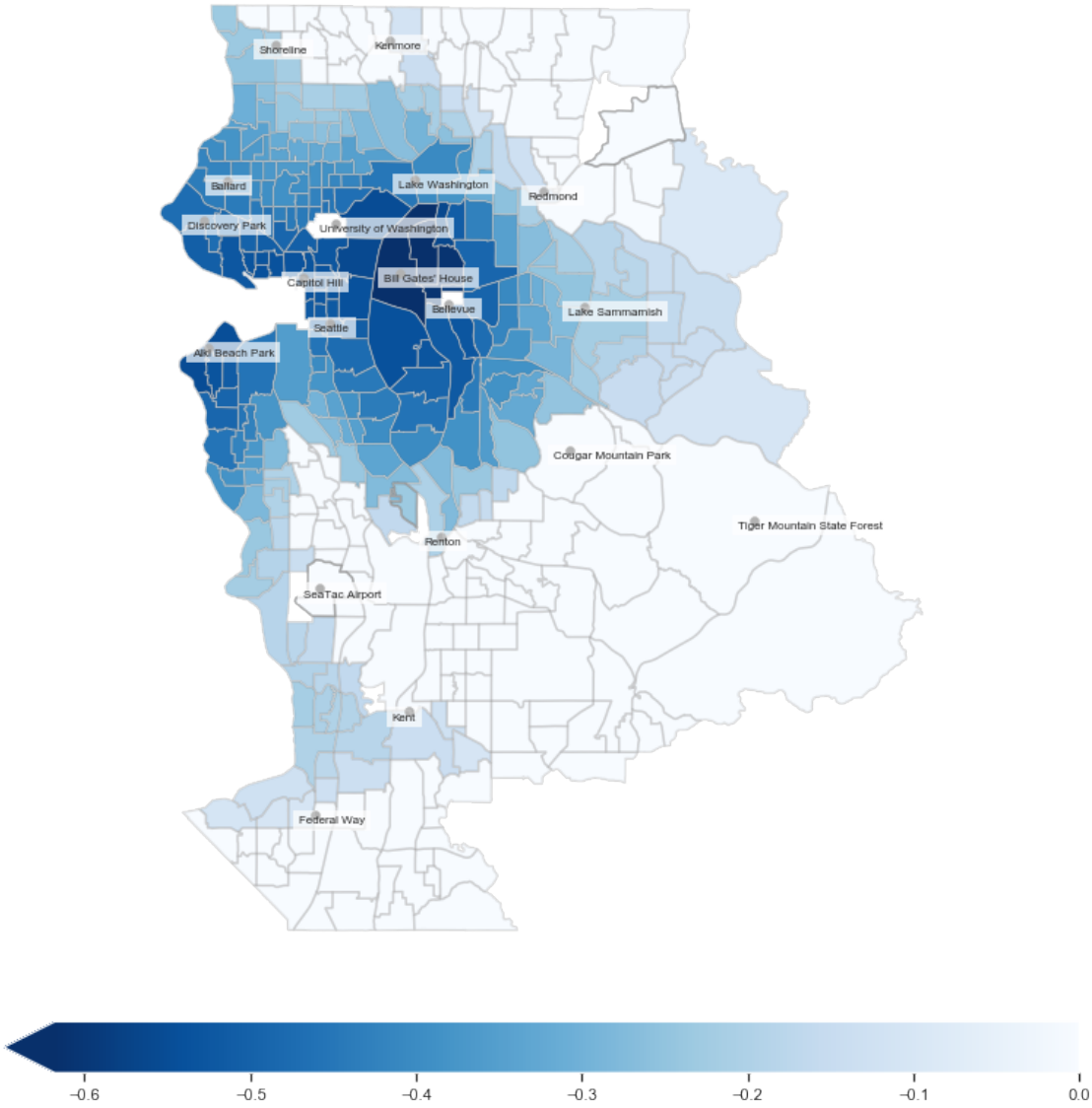


3.0.2 Only significant parameter estimates for age covariate

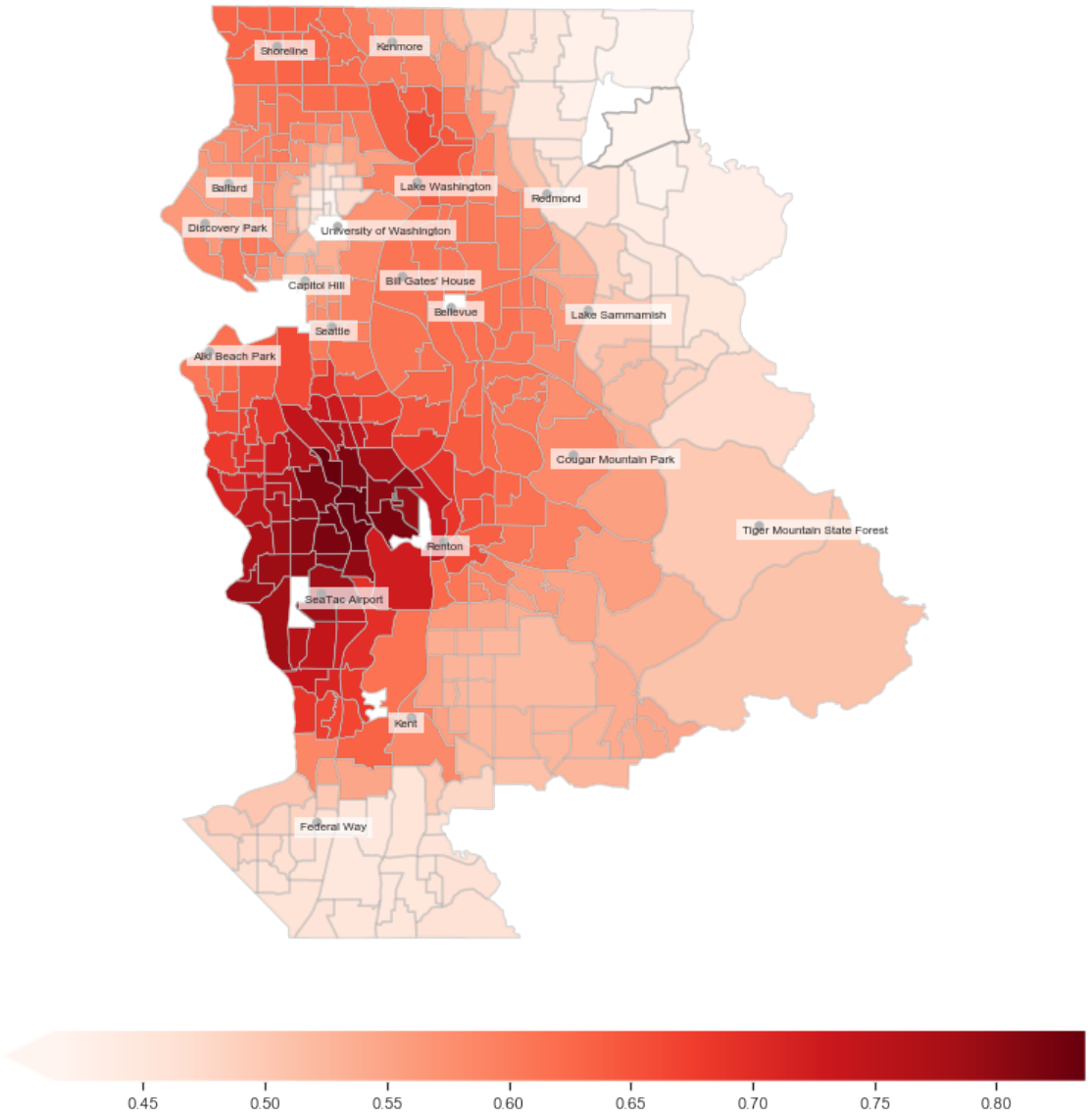
```
In [6]: maps.mapp(name='bt_age',color='Reds',filename="bt_age",normal=False)
```



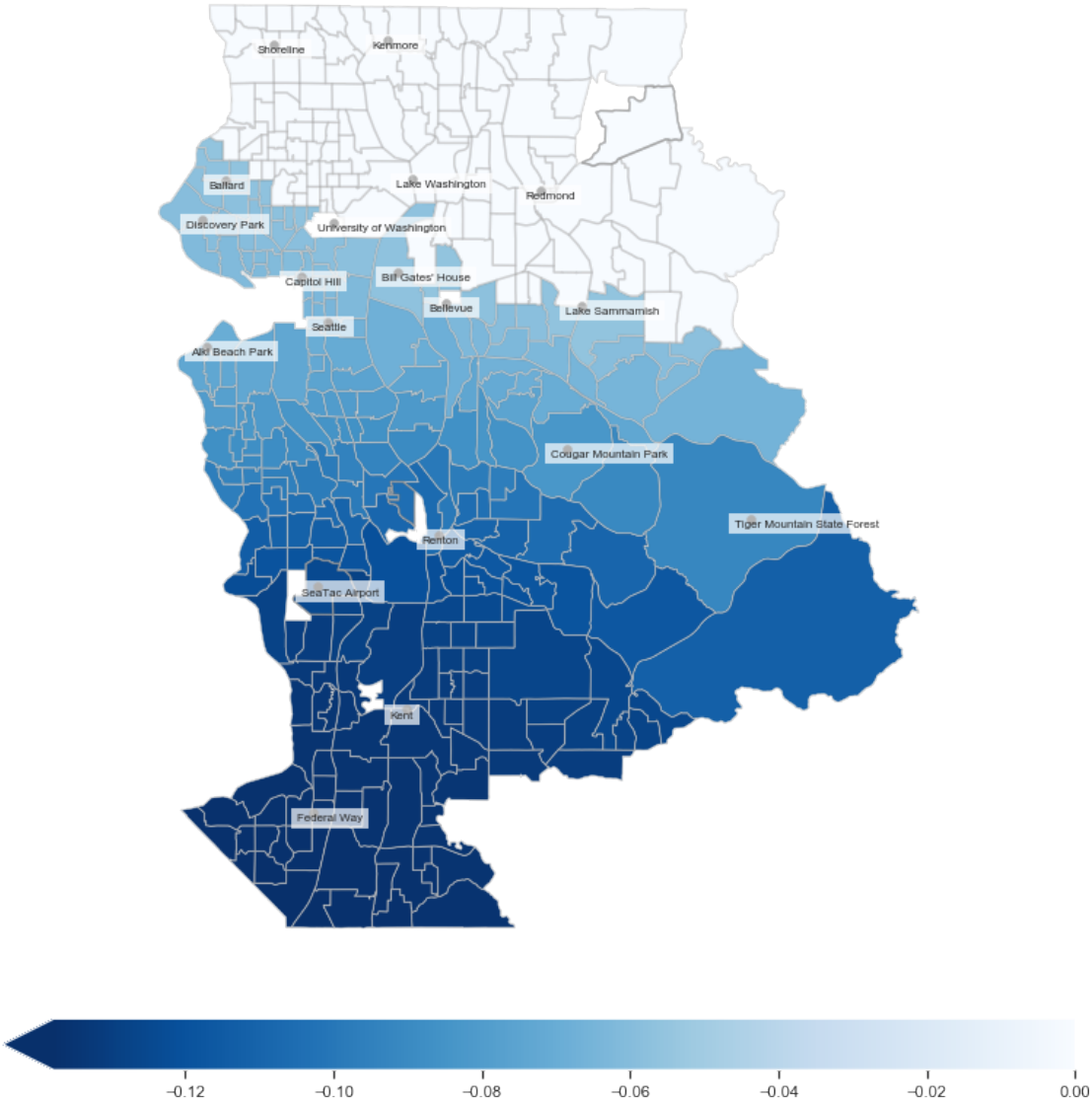

```
In [9]: maps.mapp(name='bt_water_dist',color='Blues_r',filename="bt_water_dist",normal=False)
```



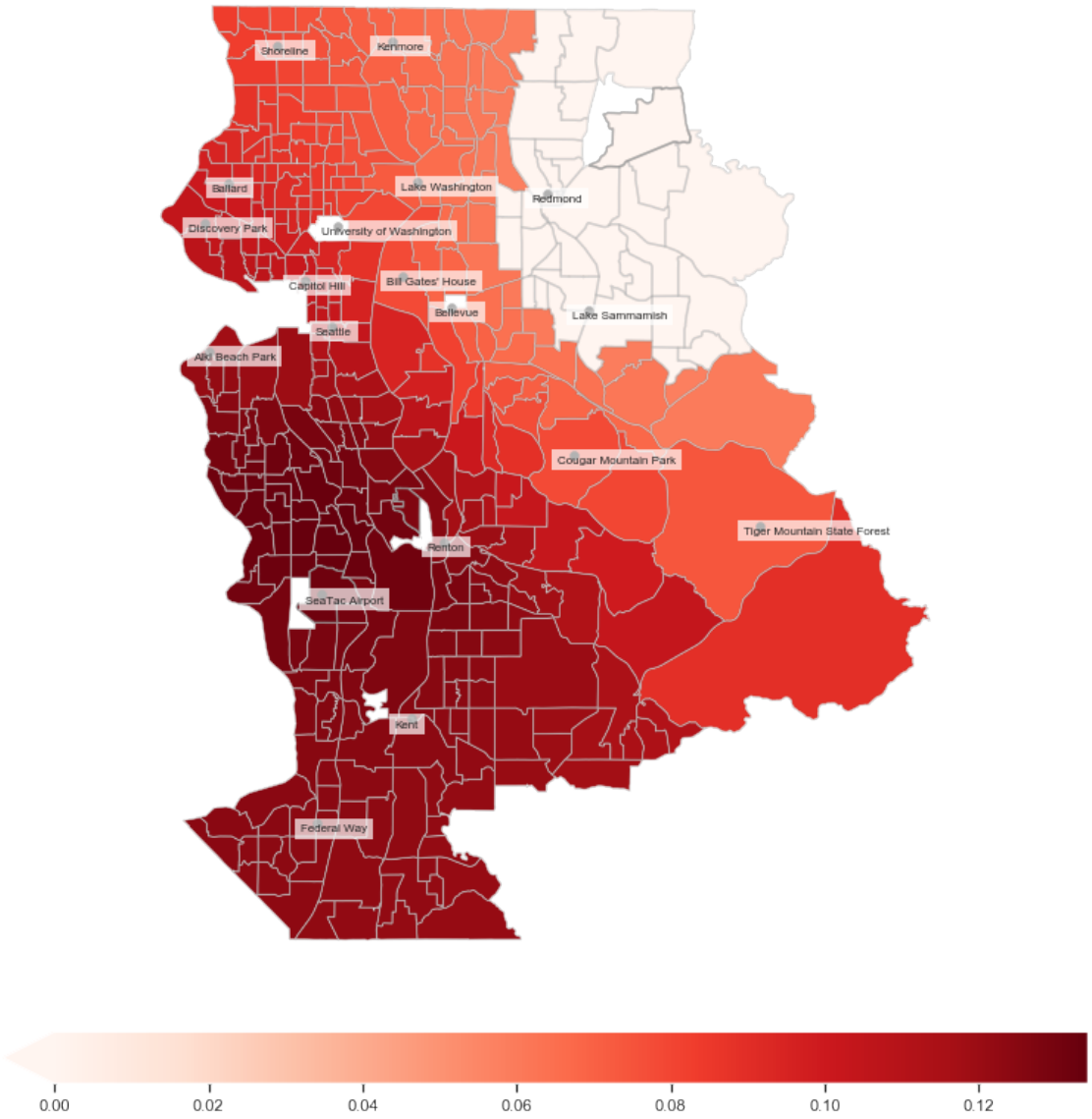

```
In [10]: maps.mapp(name='bt_sqft',color='Reds',filename="bt_sqft",normal=False)
```



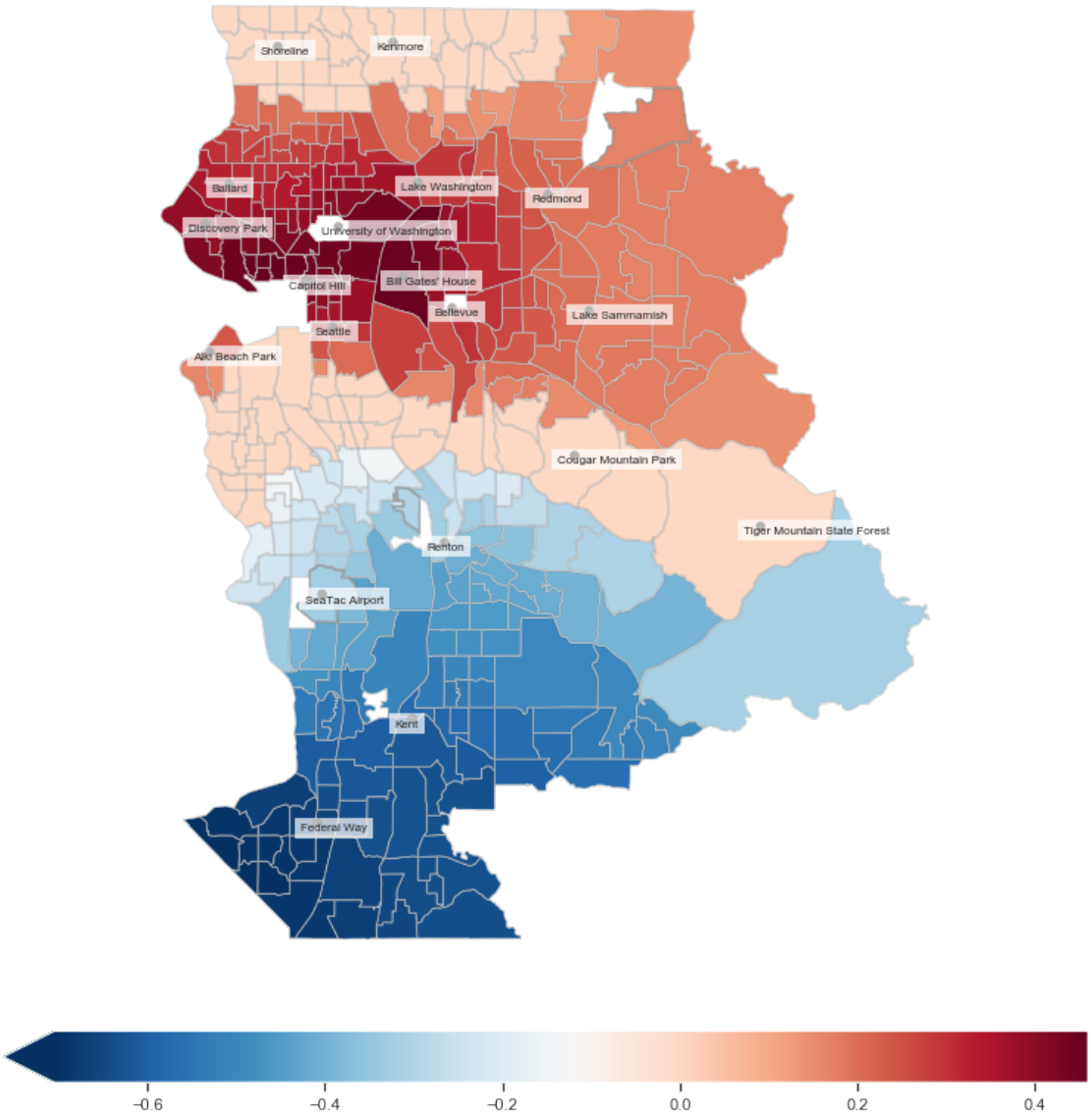

```
In [13]: maps.mapp(name='bt_unemp',color='Blues_r',filename="bt_unemp",normal=False)
```




```
In [17]: maps.mapp(name='bt_tech',color='Reds',filename="bt_tech",normal=False)
```



In [20]: maps.mapp(name='bt_constant',color='RdBu_r',filename="bt_constant",normal=False)



4 Interpretation of Maps