

```

def draw_border_lines(self, datalabel):
    # 시도 경계 그린다.
    for path in self.BORDER_LINES:
        ys, xs = zip(*path)
        plt.plot(xs, ys, c='black', lw=2)

    plt.gca().invert_yaxis()

    plt.axis('off')

    cb = plt.colorbar(shrink=.1, aspect=10)
    cb.set_label(datalabel)

    plt.tight_layout()
    plt.rc('axes', unicode_minus=False)
    plt.show()

def demographic_crisis(self):
    pop = self.cartogram_map()
    pop['여성비'] = (pop['인구수여성'] / pop['인구수합계'] - 0.5) * 100
    pop['2030여성비'] = (pop['20-39세여성'] / pop['20-39세합계'] - 0.5) * 100
    pop.to_csv('./save/pop.csv', index=False)
    pop_folium = pop.set_index('ID')
    file = self.file
    file.fname = 'skorea_municipalities_geo_simple'
    geo_str = self.map_json(file)

    pop_sum_map = folium.Map(location=[36.2002, 127.054], zoom_start=7)
    pop_sum_map.choropleth(geo_data=geo_str,
                           data=pop_folium['인구수합계'],
                           columns=[pop_folium.index, pop_folium['인구수합계']],
                           fill_color='YlGnBu', # PuRd, YlGnBu
                           key_on='feature.id')

    pop_sum_map.save('./save/pop_sum_map.html')

    extinction_danger_map = folium.Map(location=[36.2002, 127.054], zoom_start=7)
    extinction_danger_map.choropleth(geo_data=geo_str,
                                      data=pop_folium['소멸위기지역'],
                                      columns=[pop_folium.index, pop_folium['소멸위기지역']],
                                      fill_color='PuRd', # PuRd, YlGnBu
                                      key_on='feature.id')
    extinction_danger_map.save('./save/extinction_danger_map.html')

if __name__ == '__main__':
    Solution().hook()

```