

```

---
252     # 시도 경계 그래프.
253     for path in self.BORDER_LINES:
254         ys, xs = zip(*path)
255         plt.plot(xs, ys, c='black', lw=1.5)
256
257     plt.gca().invert_yaxis()
258     # plt.gca().set_aspect(1)
259
260     plt.axis('off')
261
262     plt.tight_layout()
263     plt.rc('axes', unicode_minus=False)
264     plt.show()
265
266     # ic(pop)
267
268     # 인구에 대한 통계 결과인 pop과 지도를 그리기 위한 draw_korea의 데이터를 합칠 때 사분할 key와 ID 합칠의 내용이 문제가 없는지 확인하자.
269     set(draw_korea['ID'].unique()) - set(pop['ID'].unique()) # set()
270     set(pop['ID'].unique()) - set(draw_korea['ID'].unique()) # {'충주', '진주', '의양', '충원', '포항', '천안', '부천', '안산', '성남', '고양', '수원', '동원'}
271
272     # 위 결과에 따르면, pop에 행정구별 가진 시도의 데이터가 더 있다는 것을 알 수 있다.
273     # 어차피 지도에서는 표시되지 못하니 삭제한다.
274     tmp_list = list(set(pop['ID'].unique()) - set(draw_korea['ID'].unique()))
275
276     for tmp in tmp_list:
277         pop = pop.drop(pop[pop['ID'] == tmp].index)
278
279     set(pop['ID'].unique()) - set(draw_korea['ID'].unique()) # set()
280
281     # ic(pop.head(5))
282
283     # 이제 pop과 draw_korea의 ID 합일이 일치했다고 보고, ID를 key로 merge를 시키도록 한다.
284     pop = pd.merge(pop, draw_korea, how='left', on=['ID'])
285     # ic(pop.head(5))
286
287     # 이제 위 pop 데이터에서 지도에 표현하고자 하는 데이터가 인구+행정계라는 이 라벨이 아파 모든 각 해당 행정구역에 위치하면 된다.
288     mapdata = pop.pivot_table(index='y', columns='x', values='인구+행정')
289     masked_mapdata = np.ma.masked_where(np.isnan(mapdata), mapdata)
290     ic(mapdata)
291     ic(masked_mapdata)
292
293     draw_korea.to_csv("./save/draw_korea.csv", encoding='utf-8', sep=',')
294     pop.to_csv("./save/pop.csv", index=False)
295     return pop
---

```