

```

203     pop['ID'] = sl_name
204     del pop['20-39세남자']
205     del pop['65세이상남자']
206     del pop['65세이상여자']
207
208     ic(pop.head(5))
209     return pop
210
211 def cartogram_map(self):
212     pop = self.visualize_map()
213     file = self.file
214     file.fname = 'draw_korea_raw'
215     draw_korea_raw = self.xlsx(file=file, header=0)
216     # ic(draw_korea_raw.head(5))
217
218     # 이제 각 행정 구역의 화면상 표표를 그리기 위해 pivot_table의 반대 개념으로 .stack() 명령을 사용한다.
219     draw_korea_raw_stacked = pd.DataFrame(draw_korea_raw.stack())
220     draw_korea_raw_stacked.reset_index(inplace=True)
221     draw_korea_raw_stacked.rename(columns={'level_0': 'y', 'level_1': 'x', 0: 'ID'},
222                                   inplace=True)
223     ic(draw_korea_raw_stacked)
224
225     # 인덱스를 재설정하고 컬럼의 이름을 다시 설정해 준다.
226     draw_korea = draw_korea_raw_stacked
227
228     plt.figure(figsize=(8, 11))
229
230     # 지역 이름 표시
231     for idx, row in draw_korea.iterrows():
232
233         # 광역시는 구 이름이 겹치는 경우가 많아서 시당위 이름도 같이 표시한다.
234         # (중구, 서구)
235         if len(row['ID'].split()) == 2:
236             dispname = '{}\n{}'.format(row['ID'].split()[0], row['ID'].split()[1])
237         elif row['ID'][:2] == '고령':
238             dispname = '고령'
239         else:
240             dispname = row['ID']
241
242         # 서대문구, 서귀포시 같이 이름이 3자 이상의 경우에 작은 글자로 표시한다.
243         if len(dispname.splitlines()[-1]) >= 3:
244             fontsize, linespacing = 9.5, 1.5
245         else:
246             fontsize, linespacing = 11, 1.2
247
248         plt.annotate(dispname, (row['x'] + 0.5, row['y'] + 0.5), weight='bold',
249                     fontsize=fontsize, ha='center', va='center',
250                     linespacing=linespacing)

```