def drawKorea(targetData, blockedMap, cmapname):

gamma = 0.75

whitelabelmin = (max(blockedMap[targetData]) - min(blockedMap[targetData]))\*0.25 + \

min(blockedMap[targetData])

datalabel = targetData

vmin = min(blockedMap[targetData])

vmax = max(blockedMap[targetData])

mapdata = blockedMap.pivot\_table(index='y',columns='x', values=targetData)

masked\_mapdata = np.ma.masked\_where(np.isnan(mapdata),mapdata)

plt.figure(figsize=(9,11))

plt.pcolor(masked\_mapdata, vmin=vmin, vmax=vmax, cmap=cmapname, edgecolor='#aaaaaa',linewidth=0.5)

#지역이름 표시

for idx, row in blockedMap.iterrows():

if len(row['ID'].split())==2:

dispname = '{}\n{}'.format(row['ID'].split()[0],row['ID'].split()[1])

elif row['ID'][:2]=='고성':

dispname = '고성'

else:

dispname = row['ID']

if len(dispname.splitlines()[-1]) >= 3:

fontsize, linespacing = 10.0,1.1

else:

fontsize, linespacing = 11,1.

annocolor = 'white' if row[targetData] > whitelabelmin else 'black'

plt.annotate(dispname, (row['x']+0.5,row['y']+0.5),weight='bold',

fontsize=fontsize, ha='center', va='center',color=annocolor,

linespacing=linespacing)

for path in BORDER\_LINES:

ys, xs = zip(\*path)

plt.plot(xs,ys,c='black',lw=2)

plt.gca().invert\_yaxis()

#plt.gca().set\_aspect(1)

plt.axis('off')

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

cb.set\_label(datalabel)

plt.tight\_layout()

plt.show()