import numpy as np  
# np.set\_printoptions(threshold=np.inf)  
import pandas as pd  
import netCDF4 as nc  
import xarray as xr  
import rioxarray  
import openpyxl  
import math  
  
import matplotlib.cm as cm  
import matplotlib as mpl  
import matplotlib.pyplot as plt  
import matplotlib.ticker as mticker  
import matplotlib.colors as mcolors  
import matplotlib.transforms as mtransforms  
  
import cartopy  
import cartopy.crs as ccrs  
import cartopy.feature as cfeature  
import cartopy.mpl.ticker as cticker  
from cartopy.mpl.ticker import LongitudeFormatter**,** LatitudeFormatter  
from cartopy.mpl.gridliner import LONGITUDE\_FORMATTER**,** LATITUDE\_FORMATTER  
from cartopy.feature import NaturalEarthFeature  
import cartopy.io.shapereader as shpreader  
import shapefile  
import geopandas as gpd  
from shapely import geometry  
import os  
import rioxarray  
  
def adjust\_sub\_axes(ax\_main**,** ax\_sub**,** shrink):  
 *'''  
 将ax\_sub调整到ax\_main的右下角. shrink指定缩小倍数.  
 当ax\_sub是GeoAxes时, 需要在其设定好范围后再使用此函数.  
 '''* bbox\_main = ax\_main.get\_position()  
 bbox\_sub = ax\_sub.get\_position()  
 # 使shrink=1时ax\_main与ax\_sub等宽或等高.  
 if bbox\_sub.width > bbox\_sub.height:  
 ratio = bbox\_main.width / bbox\_sub.width \* shrink  
 else:  
 ratio = bbox\_main.height / bbox\_sub.height \* shrink  
 wnew = bbox\_sub.width \* ratio  
 hnew = bbox\_sub.height \* ratio  
 bbox\_new = mtransforms.Bbox.from\_extents(bbox\_main.x1 - wnew**,** bbox\_main.y0**,** bbox\_main.x1**,** bbox\_main.y0 + hnew)  
 ax\_sub.set\_position(bbox\_new)  
  
path = r'C:\Users\28166\Desktop\SecondOutput\_China\_Grid\_Population\_Density\_Y2000toY2021.nc'  
data = nc.Dataset(path)  
lat = np.array(data.variables['latitude'][:]) # from 15.05 to 59.95 by 0.1 (450)  
lon = np.array(data.variables['longitude'][:]) # from 70.05 to 139.95 by 0.1 (700)  
pop = np.array(data.variables['pop\_density4each\_age'][**13**:**22,** :**,** :**,** :]) # (year, age, lat, lon),(9, 18, 450, 700)  
# pop中为0  
# pop\_nan = np.where(pop == 0, np.nan, pop) # (9, 18, 450, 700)  
# # 这步换成了nan  
# print(pop\_nan)  
# print(pop\_nan.shape)  
# pop\_new = np.nansum(pop\_nan, axis=1) # (9, 450, 700)  
# 使用nansum之后又变成了0，所以一开始最后画图的时候还是用的有0的数据  
  
pop\_new = np.nansum(pop**,** axis=**1**)  
pop\_new = np.where(pop\_new == **0,** np.nan**,** pop\_new)  
  
# 设置标题和字体  
plt.rcParams['font.family'] = 'Times New Roman'  
plt.rcParams['font.size'] = **14**# 画图  
proj = ccrs.PlateCarree() # 创建地图投影  
fig = plt.figure(figsize=(**33.4, 20**))  
fig.suptitle('Space Distribution of Grid Population '**,** fontsize=**20,** weight='bold'**,** y=**0.94**) # 设置大图标题,y用来控制大标题的相对位置  
axes\_main = fig.subplots(**3, 3,** subplot\_kw=dict(projection=proj))  
axes\_sub = fig.subplots(**3, 3,** subplot\_kw=dict(projection=proj))  
extent\_main = [**70, 140, 15, 55**]  
extents\_sub = [**105, 125, 0, 25**]  
fig.subplots\_adjust(right=**0.85**) # 设置色条  
china = gpd.read\_file("D:/各类文件/china\_map/分省各级别shp（超全）/1. Country/country.shp")  
nanhai = gpd.read\_file("D:/各类文件/china\_map/9duanxian/9duanxian.shp")  
shengji = gpd.read\_file("D:/各类文件/china\_map/分省各级别shp（超全）/2. Province/province.shp")  
nineduanxian = gpd.read\_file("D:\各类文件\china\_map\9duanxian\9duanxian.shp")  
  
# colorlevel = np.linspace(0, 100000, 11)  
# colordict = ['#FFFFFF', '#C2E8FA', '#86C5EB', '#5196CF', '#49A383', '#6ABF4A', '#D9DE58', '#F8B246', '#F26429', '#DD3528', '#BC1B23', '#921519']  
# color\_map = mcolors.ListedColormap(colordict)  
# norm = mcolors.BoundaryNorm(colorlevel, 12)  
# norm = mcolors.Normalize(vmin=0, vmax=100000)  
# 'inferno' ,  
  
# color\_list = ['#FFFFFF', '#FCBA1D', '#E35634', '#7A1C6C', '#110931']  
# new\_cmap = mcolors.LinearSegmentedColormap.from\_list('new\_cmap', color\_list, 20)  
norm = mpl.colors.Normalize(vmin=**0,** vmax=**100000**)  
  
n = **2013**k = **0**for i in range(**3**):  
 for j in range(**3**):  
 axes\_main[i**,** j].set\_extent(extent\_main**,** crs=proj)  
 china.plot(ax=axes\_main[i**,** j]**,** color='None'**,** edgecolor='gray'**,** linewidths=**1,** zorder=**3**)  
 shengji.plot(ax=axes\_main[i**,** j]**,** color='None'**,** edgecolor='gray'**,** linewidths=**0.4,** zorder=**3**)  
 nineduanxian.plot(ax=axes\_main[i**,** j]**,** color='None'**,** edgecolor='gray'**,** linewidths=**2,** zorder=**4**)  
 # 绘制填色图  
 sc = axes\_main[i**,** j].contourf(lon**,** lat**,** pop\_new[k]**,** cmap='cividis'**,** levels=np.linspace(**0, 100000, 11**)**,** transform=proj**,** zorder=**2**)  
 # 设置经纬度  
 gl = axes\_main[i**,** j].gridlines(crs=proj**,** draw\_labels=True**,** linestyle=":"**,** linewidth=**0.1,** x\_inline=False**,** y\_inline=False**,** color='k'**,** alpha=**0.5,** xlines=False**,** ylines=False)  
 gl.top\_labels = False  
 gl.right\_labels = False  
 gl.xformatter = LONGITUDE\_FORMATTER # x轴设为经度的格式  
 gl.yformatter = LATITUDE\_FORMATTER # y轴设为纬度的格式  
 gl.xlocator = mticker.FixedLocator([**80, 90, 100, 110, 120, 130, 140**]) # extent[0], extent[1]+0.5, 10  
 gl.ylocator = mticker.FixedLocator([**20, 30, 40, 50, 60**]) # extent[2], extent[3]+0.5, 10  
 gl.xlines = False  
 gl.ylines = False  
 font2 = {'size': **10,** 'family': 'Times New Roman'**,** 'weight': 'normal'}  
 gl.xlabel\_style = font2  
 gl.ylabel\_style = font2  
 # 画南海及九段线  
 axes\_sub[i**,** j].set\_extent(extents\_sub**,** crs=proj)  
 sc2 = axes\_sub[i**,** j].contourf(lon**,** lat**,** pop\_new[k]**,** cmap='cividis'**,** levels=np.linspace(**0, 100000, 11**)**,** transform=proj**,** zorder=**2**)  
 china.plot(ax=axes\_sub[i**,** j]**,** color='white'**,** edgecolor='gray'**,** zorder=**0,** linewidths=**0.35**)  
 nanhai.plot(ax=axes\_sub[i**,** j]**,** color='gray'**,** edgecolor='gray'**,** zorder=**1**)  
 nineduanxian.plot(ax=axes\_main[i**,** j]**,** color='None'**,** edgecolor='gray'**,** linewidths=**2,** zorder=**3**)  
 adjust\_sub\_axes(axes\_main[i**,** j]**,** axes\_sub[i**,** j]**,** shrink=**0.3**)  
 # 添加年份文字  
 axes\_main[i**,** j].text(**0.03, 0.91,** f'{n}'**,** bbox={'facecolor': 'white'**,** 'alpha': **1**}**,** fontsize=**8,** transform=axes\_main[i**,** j].transAxes**,** color='k'**,** weight='bold')  
  
 cbar\_ax = fig.add\_axes([**0.89, 0.15, 0.01, 0.7**]) # 位置  
 cbar = fig.colorbar(mpl.cm.ScalarMappable(norm=norm**,** cmap='cividis')**,** cax=cbar\_ax**,** format='%.2f'**,** shrink=**0.88,** ticks=np.linspace(**0, 100000, 11**)**,** drawedges=False**,** extend='neither')  
 # drawedges=True,  
 ax0 = cbar.ax # 将colorbar变成一个新的ax对象，可通过ax对象的各种命令来调整colorbar  
 ax0.set\_title(' Population (person)'**,** fontproperties='Times New Roman'**,** weight='normal'**,** size=**12,** pad=**20**)  
 ax0.tick\_params(which='major'**,** direction='in'**,** labelsize=**12,** length=**4**)  
 n = n + **1** k = k + **1**plt.show()

边缘等于0的数据，替换成nan就不影响绘图，但是有时候经过一个操作之后就会从nan变成0，所以要注意。

比如说：

含nan的np数组转换成datarray时会将nan转换成0

进行nanmean,nansum之后也会把nan变成0