Mission 3

dict\_keys(['latitude', 'longitude', 'Daily\_PM25'])

dict\_items([

('latitude', <class 'netCDF4.\_netCDF4.Variable'>

float32 latitude(lat)

long\_name: Latitude from 15.05 to 59.95 by 0.1

unlimited dimensions:

current shape = (450,)

filling on, default \_FillValue of 9.969209968386869e+36 used),

('longitude', <class 'netCDF4.\_netCDF4.Variable'>

float32 longitude(lon)

long\_name: Longitude from 70.05 to 139.95 by 0.1

unlimited dimensions:

current shape = (700,)

filling on, default \_FillValue of 9.969209968386869e+36 used),

('Daily\_PM25', <class 'netCDF4.\_netCDF4.Variable'>

float32 Daily\_PM25(year, day, lat, lon)

units: ug/m3

long\_name: TAP Daily PM25 (ug/m3) from Y2013 to Y2020

\_FillValue: -999.0

unlimited dimensions:

current shape = (20, 366, 450, 700)

filling on)])

lats:

[15.05 15.150001 15.25 15.35 15.45 15.55 15.650001

15.75 15.85 15.95 16.050001 16.15 16.25 16.35

16.45 16.550001 16.65 16.75 16.85 16.95 17.050001

17.15 17.25 17.35 17.45 17.550001 17.65 17.75

17.85 17.95 18.050001 18.15 18.25 18.35 18.45

18.550001 18.65 18.75 18.85 18.95 19.050001 19.15

19.25 19.35 19.45 19.550001 19.65 19.75 19.85

19.95 20.050001 20.15 20.25 20.35 20.45 20.550001

20.65 20.75 20.85 20.95 21.050001 21.15 21.25

21.35 21.45 21.550001 21.65 21.75 21.85 21.95

22.050001 22.15 22.25 22.35 22.45 22.550001 22.65

22.75 22.85 22.95 23.050001 23.15 23.25 23.35

23.45 23.550001 23.65 23.75 23.85 23.95 24.050001

24.15 24.25 24.35 24.45 24.550001 24.65 24.75

24.85 24.95 25.050001 25.15 25.25 25.35 25.45

25.550001 25.65 25.75 25.85 25.95 26.050001 26.15

26.25 26.35 26.45 26.550001 26.65 26.75 26.85

26.95 27.050001 27.15 27.25 27.35 27.45 27.550001

27.65 27.75 27.85 27.95 28.050001 28.15 28.25

28.35 28.45 28.550001 28.65 28.75 28.85 28.95

29.050001 29.15 29.25 29.35 29.45 29.550001 29.65

29.75 29.85 29.95 30.050001 30.15 30.25 30.35

30.45 30.550001 30.65 30.75 30.85 30.95 31.050001

31.15 31.25 31.35 31.45 31.550001 31.65 31.75

31.85 31.95 32.05 32.15 32.25 32.350002 32.45

32.55 32.65 32.75 32.850002 32.95 33.05 33.15

33.25 33.350002 33.45 33.55 33.65 33.75 33.850002

33.95 34.05 34.15 34.25 34.350002 34.45 34.55

34.65 34.75 34.850002 34.95 35.05 35.15 35.25

35.350002 35.45 35.55 35.65 35.75 35.850002 35.95

36.05 36.15 36.25 36.350002 36.45 36.55 36.65

36.75 36.850002 36.95 37.05 37.15 37.25 37.350002

37.45 37.55 37.65 37.75 37.850002 37.95 38.05

38.15 38.25 38.350002 38.45 38.55 38.65 38.75

38.850002 38.95 39.05 39.15 39.25 39.350002 39.45

39.55 39.65 39.75 39.850002 39.95 40.05 40.15

40.25 40.350002 40.45 40.55 40.65 40.75 40.850002

40.95 41.05 41.15 41.25 41.350002 41.45 41.55

41.65 41.75 41.850002 41.95 42.05 42.15 42.25

42.350002 42.45 42.55 42.65 42.75 42.850002 42.95

43.05 43.15 43.25 43.350002 43.45 43.55 43.65

43.75 43.850002 43.95 44.05 44.15 44.25 44.350002

44.45 44.55 44.65 44.75 44.850002 44.95 45.05

45.15 45.25 45.350002 45.45 45.55 45.65 45.75

45.850002 45.95 46.05 46.15 46.25 46.350002 46.45

46.55 46.65 46.75 46.850002 46.95 47.05 47.15

47.25 47.350002 47.45 47.55 47.65 47.75 47.850002

47.95 48.05 48.15 48.25 48.350002 48.45 48.55

48.65 48.75 48.850002 48.95 49.05 49.15 49.25

49.350002 49.45 49.55 49.65 49.75 49.850002 49.95

50.05 50.15 50.25 50.350002 50.45 50.55 50.65

50.75 50.850002 50.95 51.05 51.15 51.25 51.350002

51.45 51.55 51.65 51.75 51.850002 51.95 52.05

52.15 52.25 52.350002 52.45 52.55 52.65 52.75

52.850002 52.95 53.05 53.15 53.25 53.350002 53.45

53.55 53.65 53.75 53.850002 53.95 54.05 54.15

54.25 54.350002 54.45 54.55 54.65 54.75 54.850002

54.95 55.05 55.15 55.25 55.350002 55.45 55.55

55.65 55.75 55.850002 55.95 56.05 56.15 56.25

56.350002 56.45 56.55 56.65 56.75 56.850002 56.95

57.05 57.15 57.25 57.350002 57.45 57.55 57.65

57.75 57.850002 57.95 58.05 58.15 58.25 58.350002

58.45 58.55 58.65 58.75 58.850002 58.95 59.05

59.15 59.25 59.350002 59.45 59.55 59.65 59.75

59.850002 59.95 ]

Lons:

[ 70.05 70.15 70.25 70.350006 70.450005 70.55

70.65 70.75 70.850006 70.950005 71.05 71.15

71.25 71.350006 71.450005 71.55 71.65 71.75

71.850006 71.950005 72.05 72.15 72.25 72.350006

72.450005 72.55 72.65 72.75 72.850006 72.950005

73.05 73.15 73.25 73.350006 73.450005 73.55

73.65 73.75 73.850006 73.950005 74.05 74.15

74.25 74.350006 74.450005 74.55 74.65 74.75

74.850006 74.950005 75.05 75.15 75.25 75.350006

75.450005 75.55 75.65 75.75 75.850006 75.950005

76.05 76.15 76.25 76.350006 76.450005 76.55

76.65 76.75 76.850006 76.950005 77.05 77.15

77.25 77.350006 77.450005 77.55 77.65 77.75

77.850006 77.950005 78.05 78.15 78.25 78.350006

78.450005 78.55 78.65 78.75 78.85 78.950005

79.05 79.15 79.25 79.35 79.450005 79.55

79.65 79.75 79.85 79.950005 80.05 80.15

80.25 80.35 80.450005 80.55 80.65 80.75

80.85 80.950005 81.05 81.15 81.25 81.35

81.450005 81.55 81.65 81.75 81.85 81.950005

82.05 82.15 82.25 82.35 82.450005 82.55

82.65 82.75 82.85 82.950005 83.05 83.15

83.25 83.35 83.450005 83.55 83.65 83.75

83.85 83.950005 84.05 84.15 84.25 84.35

84.450005 84.55 84.65 84.75 84.85 84.950005

85.05 85.15 85.25 85.35 85.450005 85.55

85.65 85.75 85.85 85.950005 86.05 86.15

86.25 86.35 86.450005 86.55 86.65 86.75

86.85 86.950005 87.05 87.15 87.25 87.35

87.450005 87.55 87.65 87.75 87.85 87.950005

88.05 88.15 88.25 88.35 88.450005 88.55

88.65 88.75 88.85 88.950005 89.05 89.15

89.25 89.35 89.450005 89.55 89.65 89.75

89.85 89.950005 90.05 90.15 90.25 90.35

90.450005 90.55 90.65 90.75 90.85 90.950005

91.05 91.15 91.25 91.35 91.450005 91.55

91.65 91.75 91.85 91.950005 92.05 92.15

92.25 92.35 92.450005 92.55 92.65 92.75

92.85 92.950005 93.05 93.15 93.25 93.35

93.450005 93.55 93.65 93.75 93.85 93.950005

94.05 94.15 94.25 94.35 94.450005 94.55

94.65 94.75 94.85 94.950005 95.05 95.15

95.25 95.35 95.450005 95.55 95.65 95.75

95.85 95.950005 96.05 96.15 96.25 96.35

96.45 96.55 96.65 96.75 96.85 96.95

97.05 97.15 97.25 97.35 97.45 97.55

97.65 97.75 97.85 97.95 98.05 98.15

98.25 98.35 98.45 98.55 98.65 98.75

98.85 98.95 99.05 99.15 99.25 99.35

99.45 99.55 99.65 99.75 99.85 99.95

100.05 100.15 100.25 100.35 100.45 100.55

100.65 100.75 100.85 100.95 101.05 101.15

101.25 101.35 101.45 101.55 101.65 101.75

101.85 101.95 102.05 102.15 102.25 102.35

102.45 102.55 102.65 102.75 102.85 102.95

103.05 103.15 103.25 103.35 103.45 103.55

103.65 103.75 103.85 103.95 104.05 104.15

104.25 104.35 104.45 104.55 104.65 104.75

104.85 104.95 105.05 105.15 105.25 105.35

105.45 105.55 105.65 105.75 105.85 105.95

106.05 106.15 106.25 106.35 106.45 106.55

106.65 106.75 106.85 106.95 107.05 107.15

107.25 107.35 107.45 107.55 107.65 107.75

107.85 107.95 108.05 108.15 108.25 108.35

108.45 108.55 108.65 108.75 108.85 108.95

109.05 109.15 109.25 109.35 109.45 109.55

109.65 109.75 109.85 109.95 110.05 110.15

110.25 110.35 110.45 110.55 110.65 110.75

110.85 110.95 111.05 111.15 111.25 111.35

111.45 111.55 111.65 111.75 111.85 111.95

112.05 112.15 112.25 112.35 112.45 112.55

112.65 112.75 112.85 112.95 113.05 113.15

113.25 113.35 113.45 113.55 113.65 113.75

113.85 113.95 114.049995 114.15 114.25 114.35

114.45 114.549995 114.65 114.75 114.85 114.95

115.049995 115.15 115.25 115.35 115.45 115.549995

115.65 115.75 115.85 115.95 116.049995 116.15

116.25 116.35 116.45 116.549995 116.65 116.75

116.85 116.95 117.049995 117.15 117.25 117.35

117.45 117.549995 117.65 117.75 117.85 117.95

118.049995 118.15 118.25 118.35 118.45 118.549995

118.65 118.75 118.85 118.95 119.049995 119.15

119.25 119.35 119.45 119.549995 119.65 119.75

119.85 119.95 120.049995 120.15 120.25 120.35

120.45 120.549995 120.65 120.75 120.85 120.95

121.049995 121.15 121.25 121.35 121.45 121.549995

121.65 121.75 121.85 121.95 122.049995 122.15

122.25 122.35 122.45 122.549995 122.65 122.75

122.85 122.95 123.049995 123.15 123.25 123.35

123.45 123.549995 123.65 123.75 123.85 123.95

124.049995 124.15 124.25 124.35 124.45 124.549995

124.65 124.75 124.85 124.95 125.049995 125.15

125.25 125.35 125.45 125.549995 125.65 125.75

125.85 125.95 126.049995 126.15 126.25 126.35

126.45 126.549995 126.65 126.75 126.85 126.95

127.049995 127.15 127.25 127.35 127.45 127.549995

127.65 127.75 127.85 127.95 128.05 128.15

128.25 128.34999 128.45 128.55 128.65 128.75

128.84999 128.95 129.05 129.15 129.25 129.34999

129.45 129.55 129.65 129.75 129.84999 129.95

130.05 130.15 130.25 130.34999 130.45 130.55

130.65 130.75 130.84999 130.95 131.05 131.15

131.25 131.34999 131.45 131.55 131.65 131.75

131.84999 131.95 132.05 132.15 132.25 132.34999

132.45 132.55 132.65 132.75 132.84999 132.95

133.05 133.15 133.25 133.34999 133.45 133.55

133.65 133.75 133.84999 133.95 134.05 134.15

134.25 134.34999 134.45 134.55 134.65 134.75

134.84999 134.95 135.05 135.15 135.25 135.34999

135.45 135.55 135.65 135.75 135.84999 135.95

136.05 136.15 136.25 136.34999 136.45 136.55

136.65 136.75 136.84999 136.95 137.05 137.15

137.25 137.34999 137.45 137.55 137.65 137.75

137.84999 137.95 138.05 138.15 138.25 138.34999

138.45 138.55 138.65 138.75 138.84999 138.95

139.05 139.15 139.25 139.34999 139.45 139.55

139.65 139.75 139.84999 139.95 ]

# 画一张图

import numpy as np  
import pandas as pd  
import netCDF4 as nc  
import xarray as xr  
  
import math  
  
import matplotlib  
import matplotlib.cm as cm  
import matplotlib as mpl  
import matplotlib.pyplot as plt  
import matplotlib.ticker as mticker  
import matplotlib.colors as mcolors  
from matplotlib.colors import LinearSegmentedColormap, ListedColormap  
  
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  
  
# C:/Users/28166/Desktop/huanke peixun/China\_shapefile/CN-sheng-A.shp  
  
file = r'D:\huankepeixun\mission 3\TAP\_Daily\_PM25\_Y2001toY2020\TAP\_Daily\_PM25\_Y2001toY2020.nc'  
# 以nc文件对象格式打开  
dataset = nc.Dataset(file)  
all\_vars = dataset.variables.keys()  
all\_vars\_info = dataset.variables.items()  
# dict\_keys(['latitude', 'longitude', 'Daily\_PM25'])  
# latitude(450),Latitude from 15.05 to 59.95 by 0.1  
# longitude(700),Longitude from 70.05 to 139.95 by 0.1  
# float32 Daily\_PM25(year, day, lat, lon)(20, 366, 450, 700)  
  
lats = dataset.variables['latitude'][:]  
lons = dataset.variables['longitude'][:]  
lats\_data = np.array(lats)  
lons\_data = np.array(lons)  
  
# (20, 366, 450, 700) 的数组，这个数组的大小超过了你的计算机的可用内存，因此程序无法执行  
# 取出其中一年  
  
  
'''concs1 = dataset.variables['Daily\_PM25'][1, :, :, :]  
concs2 = dataset.variables['Daily\_PM25'][2, :, :, :]  
concs3 = dataset.variables['Daily\_PM25'][3, :, :, :]  
concs4 = dataset.variables['Daily\_PM25'][4, :, :, :]  
concs5 = dataset.variables['Daily\_PM25'][5, :, :, :]  
concs6 = dataset.variables['Daily\_PM25'][6, :, :, :]  
concs7 = dataset.variables['Daily\_PM25'][7, :, :, :]  
concs8 = dataset.variables['Daily\_PM25'][8, :, :, :]  
concs9 = dataset.variables['Daily\_PM25'][9, :, :, :]  
concs10 = dataset.variables['Daily\_PM25'][10, :, :, :]  
concs11 = dataset.variables['Daily\_PM25'][11, :, :, :]  
concs12 = dataset.variables['Daily\_PM25'][12, :, :, :]  
concs13 = dataset.variables['Daily\_PM25'][13, :, :, :]  
concs14 = dataset.variables['Daily\_PM25'][14, :, :, :]  
concs15 = dataset.variables['Daily\_PM25'][15, :, :, :]  
concs16 = dataset.variables['Daily\_PM25'][16, :, :, :]  
concs17 = dataset.variables['Daily\_PM25'][17, :, :, :]  
concs18 = dataset.variables['Daily\_PM25'][18, :, :, :]  
concs19 = dataset.variables['Daily\_PM25'][19, :, :, :]  
'''  
concs0 = dataset.variables['Daily\_PM25'][2, :, :, :]  
concs0 = np.array(concs0)  
concs0 = np.where(concs0 == -999, np.nan, concs0)  
print(concs0.shape) # (366, 450, 700)  
  
data = xr.DataArray(concs0, dims=("day", "lasts","lons"))  
print(data)  
  
# 设置标题和字体  
plt.rcParams['font.family'] = 'Times New Roman'  
plt.rcParams['font.size'] = 14  
  
# 创建画布和子图  
fig = plt.figure(figsize=(13,10))  
proj = ccrs.PlateCarree()  
ax = fig.add\_subplot(1, 1, 1, projection=proj)  
plt.title(' title ', fontdict={'family': 'STSong', 'size': 15,'weight': 'bold'},pad=18)  
# 设置色条，颜色  
colorlevel = [0,5,10,20,30,40,60,80,100,120,150,200,240]  
colordict = ['#FFFFFF', '#C2E8FA', '#86C5EB', '#5196CF', '#49A383', '#6ABF4A', '#D9DE58', '#F8B246','#F26429', '#DD3528', '#BC1B23', '#921519']  
color\_map = mcolors.ListedColormap(colordict)#产生颜色映射  
norm = mcolors.BoundaryNorm(colorlevel, 12)  
'''  
extent = [70, 140, 15, 60]  
ax.set\_extent(extent)   
'''  
# 绘制填色图  
sc = ax.contourf(lons, lats, data.loc[:, :, :].mean("day"), cmap=color\_map,norm=norm, levels=[0,5,10,20,30,40,60,80,100,120,150,200,240],transform=proj)  
# 添加中国地图边界  
# axs[i, j].coastlines(linestyle=':', edgecolor='black')  
# 添加中国地图边界,海岸线把大陆填上色,添加省份线  
ax.add\_feature(cfeature.BORDERS, linestyle='-', edgecolor='black')  
ax.add\_feature(cfeature.COASTLINE, linestyle='-', edgecolor='black')  
ax.add\_feature(cfeature.LAND,facecolor='white')  
ax.add\_feature(cfeature.STATES, linestyle=':', edgecolor='gray', visible=False)  
  
# 设置经纬度  
gl = ax.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': 8, 'family': 'Times New Roman', 'weight': 'normal'}  
gl.xlabel\_style = font2  
gl.ylabel\_style = font2  
  
# 添加左上角年份的小方块  
n = 2013  
ax.text(0.03, 0.90, f'{n}', bbox={'facecolor': 'white', 'alpha': 1}, fontsize=8,transform=ax.transAxes)  
  
# 添加颜色条  
cbar\_ax = fig.add\_axes([0.93, 0.15, 0.02, 0.7]) # 位置  
cbar = fig.colorbar(mappable=sc,cax=cbar\_ax,format='%.2f',shrink=0.88,ticks=[0,5,10,20,30,40,60,80,100,120,150,200,240])  
# 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 = cbar.ax  
ax0.set\_title('Concs', fontproperties='Times New Roman', weight='normal', size=15, pad=20)  
ax0.tick\_params(which='major',direction='in',labelsize=4,length=7.5)  
  
plt.show()

# 一次画九张图，2012-2020

import numpy as np  
import cartopy.crs as ccrs  
import cartopy.feature as cfeat  
from cartopy.mpl.gridliner import LONGITUDE\_FORMATTER**,** LATITUDE\_FORMATTER  
from cartopy.io.shapereader import Reader  
import matplotlib.pyplot as plt  
import matplotlib.ticker as mticker  
import cartopy.io.shapereader as shpreader  
import geopandas as gpd  
  
import pandas as pd  
import netCDF4 as nc  
import xarray as xr  
  
import math  
  
import matplotlib  
import matplotlib.cm as cm  
import matplotlib as mpl  
import matplotlib.pyplot as plt  
import matplotlib.ticker as mticker  
import matplotlib.colors as mcolors  
from matplotlib.colors import LinearSegmentedColormap**,** ListedColormap  
  
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  
shp\_path=r'D:\huankepeixun\China\_shapefile\CN-sheng-A.shp'  
  
# 设置标题和字体  
plt.rcParams['font.family'] = 'Times New Roman'  
plt.rcParams['font.size'] = **14**# contourf数据处理  
file = r'D:\huankepeixun\mission 3\TAP\_Daily\_PM25\_Y2001toY2020\TAP\_Daily\_PM25\_Y2001toY2020.nc'  
dataset = nc.Dataset(file)  
# dict\_keys(['latitude', 'longitude', 'Daily\_PM25'])  
# latitude(450),Latitude from 15.05 to 59.95 by 0.1  
# longitude(700),Longitude from 70.05 to 139.95 by 0.1  
# float32 Daily\_PM25(year, day, lat, lon)(20, 366, 450, 700)  
lats = dataset.variables['latitude'][:]  
lons = dataset.variables['longitude'][:]  
lats\_data = np.array(lats)  
lons\_data = np.array(lons)  
concs = dataset.variables['Daily\_PM25'][**11**:**20,** :**,** :**,** :]  
concs = np.array(concs)  
concs = np.where(concs == -**999,** np.nan**,** concs)  
data = xr.DataArray(concs**,** dims=("year"**,**"day"**,** "lasts"**,**"lons"))  
data = data.loc[:**,** :**,** :**,** :].mean("day")  
print(data.shape)  
  
  
# 画图  
proj = ccrs.PlateCarree() # 创建地图投影  
fig**,** axs = plt.subplots(nrows=**3,** ncols=**3,** figsize=(**16.7, 10**)**,** subplot\_kw=dict(projection=proj)) # 创建画布和子图  
fig.suptitle('Space Distribution of PM2.5'**,** 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  
  
fig.subplots\_adjust(right=**0.85**) # 设置色条  
  
colorlevel = [**0,5,10,20,30,40,60,80,100,120,150,200,240**]  
colordict = ['#FFFFFF'**,** '#C2E8FA'**,** '#86C5EB'**,** '#5196CF'**,** '#49A383'**,** '#6ABF4A'**,** '#D9DE58'**,** '#F8B246'**,**'#F26429'**,** '#DD3528'**,** '#BC1B23'**,** '#921519']  
color\_map = mcolors.ListedColormap(colordict)  
norm = mcolors.BoundaryNorm(colorlevel**, 12**)  
  
  
n = **2012**k = **0**for i in range(**3**):  
 for j in range(**3**):  
 '''extent = [70, 140, 15, 55]  
 axs[i,j].set\_extent(extent, crs=proj)'''  
  
 # 绘制填色图  
 sc = axs[i**,**j].contourf(lons**,** lats**,** data.loc[k**,** :**,** :]**,** cmap=color\_map**,** norm=norm**,** levels=[**0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240**]**,** transform=proj)  
 # 添加中国地图边界,axs[i, j].coastlines(linestyle=':', edgecolor='black')  
 axs[i**,**j].add\_feature(cfeature.BORDERS**,** linestyle='-'**,** edgecolor='black')  
 axs[i**,**j].add\_feature(cfeature.COASTLINE**,** linestyle='-'**,** edgecolor='black')  
 axs[i**,**j].add\_feature(cfeature.LAND**,** facecolor='white')  
 axs[i**,**j].add\_feature(cfeature.STATES**,** linestyle=':'**,** edgecolor='gray'**,** visible=False)  
 # 设置经纬度  
 gl = axs[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  
 # 添加年份文字  
 axs[i**,**j].text(**0.03, 0.91,** f'{n}'**,** bbox={'facecolor': 'white'**,** 'alpha': **1**}**,** fontsize=**8,** transform=axs[i**,**j].transAxes)  
 # 添加颜色条  
 cbar\_ax = fig.add\_axes([**0.88, 0.15, 0.01, 0.7**]) # 位置  
 cbar = fig.colorbar(mappable=sc**,** cax=cbar\_ax**,** format='%.2f'**,** shrink=**0.88,** ticks=[**0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240**])  
 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('Concs'**,** fontproperties='Times New Roman'**,** weight='normal'**,** size=**15,** pad=**20**)  
 ax0.tick\_params(which='major'**,** direction='in'**,** labelsize=**12,** length=**7.5**)  
  
 n = n + **1** k = k + **1**fig.savefig(f'D:/huankepeixun/mission 3/大图.png')  
plt.show()

# 绘制南海和九段线

import numpy as np  
import pandas as pd  
import netCDF4 as nc  
import xarray as xr  
  
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  
from matplotlib.colors import LinearSegmentedColormap**,** ListedColormap  
  
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  
  
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)  
  
shp\_path = r'C:\Users\28166\Desktop\china\china.shp'  
  
# 设置标题和字体  
plt.rcParams['font.family'] = 'Times New Roman'  
plt.rcParams['font.size'] = **14**# contourf数据处理  
file = r'D:\huankepeixun\mission 3\TAP\_Daily\_PM25\_Y2001toY2020\TAP\_Daily\_PM25\_Y2001toY2020.nc'  
dataset = nc.Dataset(file)  
# dict\_keys(['latitude', 'longitude', 'Daily\_PM25'])  
# latitude(450),Latitude from 15.05 to 59.95 by 0.1  
# longitude(700),Longitude from 70.05 to 139.95 by 0.1  
# float32 Daily\_PM25(year, day, lat, lon)(20, 366, 450, 700)  
lats = dataset.variables['latitude'][:]  
lons = dataset.variables['longitude'][:]  
lats\_data = np.array(lats)  
lons\_data = np.array(lons)  
concs = dataset.variables['Daily\_PM25'][**11**:**20,** :**,** :**,** :]  
concs = np.array(concs)  
concs = np.where(concs == -**999,** np.nan**,** concs)  
data = xr.DataArray(concs**,** dims=("year"**,**"day"**,** "lasts"**,**"lons"))  
data = data.loc[:**,** :**,** :**,** :].mean("day")  
print(data.shape)  
  
# 画图  
proj = ccrs.PlateCarree() # 创建地图投影  
fig = plt.figure(figsize=(**16.7, 10**))  
fig.suptitle('Space Distribution of PM2.5'**,** 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  
  
colorlevel = [**0,5,10,20,30,40,60,80,100,120,150,200,240**]  
colordict = ['#FFFFFF'**,** '#C2E8FA'**,** '#86C5EB'**,** '#5196CF'**,** '#49A383'**,** '#6ABF4A'**,** '#D9DE58'**,** '#F8B246'**,**'#F26429'**,** '#DD3528'**,** '#BC1B23'**,** '#921519']  
color\_map = mcolors.ListedColormap(colordict)  
norm = mcolors.BoundaryNorm(colorlevel**, 12**)  
  
china = gpd.read\_file('C:/Users/28166/Desktop/china\_map/china.shp')  
nanhai = gpd.read\_file('C:/Users/28166/Desktop/china\_map/9duanxian/9duanxian.shp')  
  
n = **2012**k = **0**for i in range(**3**):  
 for j in range(**3**):  
 axes\_main[i**,**j].set\_extent(extent\_main**,** crs=proj)  
 axes\_main[i**,**j].stock\_img()  
 # 添加中国地图边界,axs[i, j].coastlines(linestyle=':', edgecolor='black')  
 china.plot(ax=axes\_main[i**,**j]**,** color='white'**,** edgecolor='gray'**,**zorder=**0**)  
 # 绘制填色图  
 sc = axes\_main[i**,** j].contourf(lons**,** lats**,** data.loc[k**,** :**,** :]**,** cmap=color\_map**,** norm=norm**,** levels=[**0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240**]**,** 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  
 # 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)  
 # 添加颜色条  
 cbar\_ax = fig.add\_axes([**0.88, 0.15, 0.01, 0.7**]) # 位置  
 cbar = fig.colorbar(mappable=sc**,** cax=cbar\_ax**,** format='%.2f'**,** shrink=**0.88,** ticks=[**0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240**])  
 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('Concs'**,** fontproperties='Times New Roman'**,** weight='normal'**,** size=**15,** pad=**20**)  
 ax0.tick\_params(which='major'**,** direction='in'**,** labelsize=**12,** length=**7.5**)  
  
 # 画南海及九段线  
 axes\_sub[i**,**j].set\_extent(extents\_sub**,** crs=proj)  
 axes\_sub[i**,**j].stock\_img()  
 nanhai.plot(ax=axes\_sub[i**,**j]**,** color='white'**,** edgecolor='gray'**,**zorder=**1**)  
 adjust\_sub\_axes(axes\_main[i**,**j]**,** axes\_sub[i**,**j]**,** shrink=**0.3**)  
 n = n + **1** k = k + **1**plt.show()

# 散点图和填色图画一起3\*3，但是看不清。还有一丢丢问题，散点图是13-21，填色图是13-20（97%那张图）

import numpy as np  
import pandas as pd  
import netCDF4 as nc  
import xarray as xr  
  
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  
from matplotlib.colors import LinearSegmentedColormap**,** ListedColormap  
  
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  
  
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)  
  
shp\_path = r'C:\Users\28166\Desktop\china\china.shp'  
  
# 设置标题和字体  
plt.rcParams['font.family'] = 'Times New Roman'  
plt.rcParams['font.size'] = **14**# scatter数据处理  
file2 = r'D:\huankepeixun\mission 2\mission 2 data-DailyConc\_China\_CNEMC\_EachStation\_2013to2021\_nc\Daily\_OBS\_Concs4EachStation\_Y2013toY2021.nc'  
dataset2 = nc.Dataset(file2)  
# dict\_keys(['char\_year', 'char\_species', 'char\_std\_index', 'std\_lat', 'std\_lon', 'daily\_conc'])9，366，8，1674  
lats2 = dataset2.variables['std\_lat'][:]  
lons2 = dataset2.variables['std\_lon'][:]  
concs2 = dataset2.variables['daily\_conc'][:**,**:**,0,**:]  
lats\_data2 = np.array(lats2)  
lons\_data2 = np.array(lons2)  
concs\_data2 = np.array(concs2)  
concs\_data2 = np.where(concs\_data2 == -**999,** np.nan**,** concs\_data2)  
# 分割三维数组成9个二维数组,先切成9个(1,366,1674)  
sub\_arrays = np.split(concs\_data2**, 9,** axis=**0**)  
all\_data = []  
reshaped\_subarrays = [sub\_array.reshape(**366, 1674**) for sub\_array in sub\_arrays] # 将9个(1,366,1674)转换成9个(366,1674)  
for i**,** sub\_array in enumerate(reshaped\_subarrays):  
 last\_list = [np.nanmean(x) for x in zip(\*sub\_array)] # 在每一个（366,1674）跑的过程中，计算每个站点的年平均，即二维按列求和  
 all\_data.append(last\_list)  
all\_data = np.array(all\_data)  
print(all\_data.shape)  
  
  
# contourf数据处理  
file3 = r'D:\huankepeixun\mission 3\TAP\_Daily\_PM25\_Y2001toY2020\TAP\_Daily\_PM25\_Y2001toY2020.nc'  
dataset3 = nc.Dataset(file3)  
# dict\_keys(['latitude', 'longitude', 'Daily\_PM25'])  
# latitude(450),Latitude from 15.05 to 59.95 by 0.1  
# longitude(700),Longitude from 70.05 to 139.95 by 0.1  
# float32 Daily\_PM25(year, day, lat, lon)(20, 366, 450, 700)  
lats3 = dataset3.variables['latitude'][:]  
lons3 = dataset3.variables['longitude'][:]  
lats\_data3 = np.array(lats3)  
lons\_data3 = np.array(lons3)  
concs3 = dataset3.variables['Daily\_PM25'][**11**:**20,** :**,** :**,** :]  
concs3 = np.array(concs3)  
concs3 = np.where(concs3 == -**999,** np.nan**,** concs3)  
data3 = xr.DataArray(concs3**,** dims=("year"**,**"day"**,** "lasts"**,**"lons"))  
data3= data3.loc[:**,** :**,** :**,** :].mean("day")  
print(data3.shape)  
  
  
# 画图  
proj = ccrs.PlateCarree() # 创建地图投影  
fig = plt.figure(figsize=(**16.7, 10**))  
fig.suptitle('Space Distribution of PM2.5'**,** 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  
  
colorlevel = [**0,5,10,20,30,40,60,80,100,120,150,200,240**]  
colordict = ['#FFFFFF'**,** '#C2E8FA'**,** '#86C5EB'**,** '#5196CF'**,** '#49A383'**,** '#6ABF4A'**,** '#D9DE58'**,** '#F8B246'**,**'#F26429'**,** '#DD3528'**,** '#BC1B23'**,** '#921519']  
color\_map = mcolors.ListedColormap(colordict)  
norm = mcolors.BoundaryNorm(colorlevel**, 12**)  
  
china = gpd.read\_file('C:/Users/28166/Desktop/china\_map/china.shp')  
nanhai = gpd.read\_file('C:/Users/28166/Desktop/china\_map/9duanxian/9duanxian.shp')  
  
n = **2012**k = **0**for i in range(**3**):  
 for j in range(**3**):  
 axes\_main[i**,**j].set\_extent(extent\_main**,** crs=proj)  
 '''axes\_main[i,j].stock\_img()'''  
 # 添加中国地图边界,axs[i, j].coastlines(linestyle=':', edgecolor='black')  
 china.plot(ax=axes\_main[i**,**j]**,** color='white'**,** edgecolor='gray'**,**zorder=**0**)  
 axes\_main[i**,** j].add\_feature(cfeature.STATES**,** linestyle=':'**,** edgecolor='gray'**,** visible=False)  
 # 绘制填色图  
 sc = axes\_main[i**,** j].contourf(lons3**,** lats3**,** data3.loc[k**,** :**,** :]**,** cmap=color\_map**,** norm=norm**,** levels=[**0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240**]**,** transform=proj**,**zorder=**2**)  
 # 绘制散点图  
 sc2 = axes\_main[i**,** j].scatter(lons2**,** lats2**,** c=all\_data[k]**,** s=**0.12,** cmap=color\_map**,** norm=norm**,** transform=proj**,** zorder=**3**)  
 # 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  
 # 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)  
 # 添加颜色条  
 cbar\_ax = fig.add\_axes([**0.88, 0.15, 0.01, 0.7**]) # 位置  
 cbar = fig.colorbar(mappable=sc**,** cax=cbar\_ax**,** format='%.2f'**,** shrink=**0.88,** ticks=[**0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240**])  
 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('Concs'**,** fontproperties='Times New Roman'**,** weight='normal'**,** size=**15,** pad=**20**)  
 ax0.tick\_params(which='major'**,** direction='in'**,** labelsize=**12,** length=**7.5**)  
  
 # 画南海及九段线  
 axes\_sub[i**,**j].set\_extent(extents\_sub**,** crs=proj)  
 '''axes\_sub[i,j].stock\_img()'''  
 nanhai.plot(ax=axes\_sub[i**,**j]**,** color='gray'**,** edgecolor='gray'**,**zorder=**1**)  
 adjust\_sub\_axes(axes\_main[i**,**j]**,** axes\_sub[i**,**j]**,** shrink=**0.3**)  
 n = n + **1** k = k + **1**plt.show()

# 最终图

import numpy as np  
import pandas as pd  
import netCDF4 as nc  
import xarray as xr  
  
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  
from matplotlib.colors import LinearSegmentedColormap, ListedColormap  
  
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  
  
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)  
  
shp\_path = r'C:\Users\28166\Desktop\china\china.shp'  
  
# 设置标题和字体  
plt.rcParams['font.family'] = 'Times New Roman'  
plt.rcParams['font.size'] = 14  
  
# scatter数据处理  
file2 = r'D:\huankepeixun\mission 2\mission 2 data-DailyConc\_China\_CNEMC\_EachStation\_2013to2021\_nc\Daily\_OBS\_Concs4EachStation\_Y2013toY2021.nc'  
dataset2 = nc.Dataset(file2)  
# dict\_keys(['char\_year', 'char\_species', 'char\_std\_index', 'std\_lat', 'std\_lon', 'daily\_conc'])9，366，8，1674  
lats2 = dataset2.variables['std\_lat'][:]  
lons2 = dataset2.variables['std\_lon'][:]  
concs2 = dataset2.variables['daily\_conc'][0:8, :, 0, :]  
lats\_data2 = np.array(lats2)  
lons\_data2 = np.array(lons2)  
concs\_data2 = np.array(concs2)  
concs\_data2 = np.where(concs\_data2 == -999, np.nan, concs\_data2)  
sub\_arrays = np.split(concs\_data2, 8, axis=0)  
all\_data = []  
reshaped\_subarrays = [sub\_array.reshape(366, 1674) for sub\_array in sub\_arrays] # 将8个(1,366,1674)转换成9个(366,1674)  
for i, sub\_array in enumerate(reshaped\_subarrays):  
 last\_list = [np.nanmean(x) for x in zip(\*sub\_array)] # 在每一个（366,1674）跑的过程中，计算每个站点的年平均，即二维按列求和  
 all\_data.append(last\_list)  
all\_data = np.array(all\_data)  
print(all\_data.shape)  
  
# contourf数据处理  
file3 = r'D:\huankepeixun\mission 3\TAP\_Daily\_PM25\_Y2001toY2020\TAP\_Daily\_PM25\_Y2001toY2020.nc'  
dataset3 = nc.Dataset(file3)  
# dict\_keys(['latitude', 'longitude', 'Daily\_PM25'])  
# latitude(450),Latitude from 15.05 to 59.95 by 0.1  
# longitude(700),Longitude from 70.05 to 139.95 by 0.1  
# float32 Daily\_PM25(year, day, lat, lon)(20, 366, 450, 700)  
lats3 = dataset3.variables['latitude'][:]  
lons3 = dataset3.variables['longitude'][:]  
lats\_data3 = np.array(lats3)  
lons\_data3 = np.array(lons3)  
concs3 = dataset3.variables['Daily\_PM25'][12:20, :, :, :]  
concs3 = np.array(concs3)  
concs3 = np.where(concs3 == -999, np.nan, concs3)  
data3 = xr.DataArray(concs3, dims=("year","day", "lasts","lons"))  
data3= data3.loc[:, :, :, :].mean("day")  
print(data3.shape)  
  
# 画图  
proj = ccrs.PlateCarree() # 创建地图投影  
fig = plt.figure(figsize=(33, 20))  
fig.suptitle('Space Distribution of PM2.5', 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  
  
colorlevel = [0,5,10,20,30,40,60,80,100,120,150,200,240]  
colordict = ['#FFFFFF', '#C2E8FA', '#86C5EB', '#5196CF', '#49A383', '#6ABF4A', '#D9DE58', '#F8B246','#F26429', '#DD3528', '#BC1B23', '#921519']  
color\_map = mcolors.ListedColormap(colordict)  
norm = mcolors.BoundaryNorm(colorlevel, 12)  
  
china = gpd.read\_file('C:/Users/28166/Desktop/china\_map/china.shp')  
nanhai = gpd.read\_file('C:/Users/28166/Desktop/china\_map/9duanxian/9duanxian.shp')  
  
n = 2013  
k = 0  
for i in range(3):  
 for j in range(3):  
 if i==2 and j==2:  
 # 添加颜色条  
 axes\_main[i,j].axis('off')  
 axes\_sub[i,j].axis('off')  
 else:  
 axes\_main[i, j].set\_extent(extent\_main, crs=proj)  
 '''axes\_main[i,j].stock\_img() # 根据图片内容自行添加一张底图'''  
 # 添加中国地图边界,axs[i, j].coastlines(linestyle=':', edgecolor='black')  
 china.plot(ax=axes\_main[i, j], color='white', edgecolor='gray', zorder=0)  
 axes\_main[i, j].add\_feature(cfeature.STATES, linestyle=':', edgecolor='gray', visible=False)  
 # 绘制填色图  
 sc = axes\_main[i, j].contourf(lons3, lats3, data3.loc[k, :, :], cmap=color\_map, norm=norm,  
 levels=[0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240],  
 transform=proj, zorder=2)  
 # 绘制散点图  
 sc2 = axes\_main[i, j].scatter(lons2, lats2, c=all\_data[k], s=12, edgecolor='k', linewidths=0.1,  
 cmap=color\_map, norm=norm, transform=proj, zorder=3)  
 # 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  
 # 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)  
 # 画南海及九段线  
 axes\_sub[i, j].set\_extent(extents\_sub, crs=proj)  
 '''axes\_sub[i,j].stock\_img()'''  
 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)  
 '''axes\_sub[i, j].add\_feature(cfeature.STATES, linestyle=':', edgecolor='gray',linewidths=0.1)  
 axes\_sub[i, j].add\_feature(cfeature.BORDERS, linestyle='-', edgecolor='gray',linewidths=0.1)  
 axes\_sub[i, j].add\_feature(cfeature.COASTLINE, linestyle='-', edgecolor='black',linewidths=0.08)'''  
 adjust\_sub\_axes(axes\_main[i, j], axes\_sub[i, j], shrink=0.3)  
 cbar\_ax = fig.add\_axes([0.88, 0.15, 0.01, 0.7]) # 位置  
 cbar = fig.colorbar(mappable=sc, cax=cbar\_ax, format='%.2f', shrink=0.88,  
 ticks=[0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240])  
 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('Concs', fontproperties='Times New Roman', weight='normal', size=15, pad=20)  
 ax0.tick\_params(which='major', direction='in', labelsize=12, length=7.5)  
 n = n + 1  
 k = k + 1  
plt.show()

array(145.66923523),

array(124.60797119),

array(125.52253723),

array(102.65300751),

array(99.14395142),

array(86.37554932),

array(77.7843399),

array(78.1639328)]

[145.66924, 124.60797, 125.52254, 102.65301, 99.14395, 86.37555, 77.78434, 78.16393,]

[8.858791, 3.6717033, 5.3101654, 4.423497, 4.395879, 7.137088, 5.5324173, 7.751366,]

# 图片最终美化，加省界

import numpy as np  
import pandas as pd  
import netCDF4 as nc  
import xarray as xr  
  
import math  
  
import matplotlib  
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  
from matplotlib.colors import LinearSegmentedColormap, ListedColormap  
  
import cnmaps  
from cnmaps import get\_adm\_maps, draw\_map  
from cnmaps.sample import load\_dem  
  
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  
  
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)  
  
shp\_path = r'C:\Users\28166\Desktop\china\china.shp'  
  
# 设置标题和字体  
plt.rcParams['font.family'] = 'Times New Roman'  
plt.rcParams['font.size'] = 14  
  
# scatter数据处理  
file2 = r'D:\huankepeixun\mission 2\mission 2 data-DailyConc\_China\_CNEMC\_EachStation\_2013to2021\_nc\Daily\_OBS\_Concs4EachStation\_Y2013toY2021.nc'  
dataset2 = nc.Dataset(file2)  
# dict\_keys(['char\_year', 'char\_species', 'char\_std\_index', 'std\_lat', 'std\_lon', 'daily\_conc'])9，366，8，1674  
lats2 = dataset2.variables['std\_lat'][:]  
lons2 = dataset2.variables['std\_lon'][:]  
concs2 = dataset2.variables['daily\_conc'][0:8, :, 0, :]  
lats\_data2 = np.array(lats2)  
lons\_data2 = np.array(lons2)  
concs\_data2 = np.array(concs2)  
concs\_data2 = np.where(concs\_data2 == -999, np.nan, concs\_data2)  
sub\_arrays = np.split(concs\_data2, 8, axis=0)  
all\_data = []  
reshaped\_subarrays = [sub\_array.reshape(366, 1674) for sub\_array in sub\_arrays] # 将8个(1,366,1674)转换成9个(366,1674)  
for i, sub\_array in enumerate(reshaped\_subarrays):  
 last\_list = [np.nanmean(x) for x in zip(\*sub\_array)] # 在每一个（366,1674）跑的过程中，计算每个站点的年平均，即二维按列求和  
 all\_data.append(last\_list)  
all\_data = np.array(all\_data)  
print(all\_data.shape)  
  
# contourf数据处理  
file3 = r'D:\huankepeixun\mission 3\TAP\_Daily\_PM25\_Y2001toY2020\TAP\_Daily\_PM25\_Y2001toY2020.nc'  
dataset3 = nc.Dataset(file3)  
# dict\_keys(['latitude', 'longitude', 'Daily\_PM25'])  
# latitude(450),Latitude from 15.05 to 59.95 by 0.1  
# longitude(700),Longitude from 70.05 to 139.95 by 0.1  
# float32 Daily\_PM25(year, day, lat, lon)(20, 366, 450, 700)  
lats3 = dataset3.variables['latitude'][:]  
lons3 = dataset3.variables['longitude'][:]  
lats\_data3 = np.array(lats3)  
lons\_data3 = np.array(lons3)  
concs3 = dataset3.variables['Daily\_PM25'][12:20, :, :, :]  
concs3 = np.array(concs3)  
concs3 = np.where(concs3 == -999, np.nan, concs3)  
data3 = xr.DataArray(concs3, dims=("year","day", "lasts","lons"))  
data3= data3.loc[:, :, :, :].mean("day")  
print(data3.shape)  
  
# 画图  
proj = ccrs.PlateCarree() # 创建地图投影  
fig = plt.figure(figsize=(33, 20))  
fig.suptitle('Space Distribution of PM2.5', 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  
  
colorlevel = [0,5,10,20,30,40,60,80,100,120,150,200,240]  
colordict = ['#FFFFFF', '#C2E8FA', '#86C5EB', '#5196CF', '#49A383', '#6ABF4A', '#D9DE58', '#F8B246','#F26429', '#DD3528', '#BC1B23', '#921519']  
color\_map = mcolors.ListedColormap(colordict)  
norm = mcolors.BoundaryNorm(colorlevel, 12)  
  
china = gpd.read\_file('C:/Users/28166/Desktop/china\_map/china.shp')  
nanhai = gpd.read\_file('C:/Users/28166/Desktop/china\_map/9duanxian/9duanxian.shp')  
shengji = gpd.read\_file("C:/Users/28166/Desktop/china\_map/shengji/shengji.shp")  
n = 2013  
k = 0  
for i in range(3):  
 for j in range(3):  
 if i==2 and j==2:  
 # 添加颜色条  
 axes\_main[i,j].axis('off')  
 axes\_sub[i,j].axis('off')  
 else:  
 axes\_main[i, j].set\_extent(extent\_main, crs=proj)  
 '''axes\_main[i,j].stock\_img() # 根据图片内容自行添加一张底图'''  
 china.plot(ax=axes\_main[i, j], color='white', edgecolor='k', zorder=2)  
 '''axes\_main[i, j].add\_geometries(china.geometry, proj, edgecolor='k', facecolor='None', lw=0.4, zorder=3)'''  
 '''axes\_main[i, j].add\_feature(cfeature.STATES, linestyle=':', edgecolor='gray', visible=False, zorder=3)'''  
 # 绘制填色图  
 sc = axes\_main[i, j].contourf(lons3, lats3, data3.loc[k, :, :], cmap=color\_map, norm=norm,  
 levels=[0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240],  
 transform=proj, zorder=3)  
 # 绘制散点图  
 sc2 = axes\_main[i, j].scatter(lons2, lats2, c=all\_data[k], s=12, edgecolor='k', linewidths=0.1,  
 cmap=color\_map, norm=norm, transform=proj, zorder=5)  
 shengji.plot(ax=axes\_main[i, j], color='None', edgecolor='gray', linewidths=0.3, zorder=4)  
 # 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  
 # 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)  
 # 画南海及九段线  
 axes\_sub[i, j].set\_extent(extents\_sub, crs=proj)  
 '''axes\_sub[i,j].stock\_img()'''  
 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)  
 '''axes\_sub[i, j].add\_feature(cfeature.STATES, linestyle=':', edgecolor='gray',linewidths=0.1)  
 axes\_sub[i, j].add\_feature(cfeature.BORDERS, linestyle='-', edgecolor='gray',linewidths=0.1)  
 axes\_sub[i, j].add\_feature(cfeature.COASTLINE, linestyle='-', edgecolor='black',linewidths=0.08)'''  
 adjust\_sub\_axes(axes\_main[i, j], axes\_sub[i, j], shrink=0.3)  
 cbar\_ax = fig.add\_axes([0.88, 0.15, 0.01, 0.7]) # 位置  
 cbar = fig.colorbar(mappable=sc, cax=cbar\_ax, format='%.2f', shrink=0.88,  
 ticks=[0, 5, 10, 20, 30, 40, 60, 80, 100, 120, 150, 200, 240])  
 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('Concs', fontproperties='Times New Roman', weight='normal', size=15, pad=20)  
 ax0.tick\_params(which='major', direction='in', labelsize=12, length=7.5)  
 n = n + 1  
 k = k + 1  
plt.show()

