from scipy.interpolate import interp1d  
import pandas as pd  
import numpy as np  
  
  
from scipy.interpolate import interp1d  
import pandas as pd  
import numpy as np  
  
def read\_csv\_to\_dict(csv\_path, column\_name):  
 sea\_data = pd.read\_csv(csv\_path)  
 data\_dict = {'sea\_pressure': sea\_data['Sea pressure - decibar'].values,  
 'sea\_salinity': sea\_data[column\_name].values}  
 return data\_dict  
  
def get\_salinity\_value\_from\_dict(data\_dict, pressure\_value):  
 salinity\_interp = interp1d(data\_dict['sea\_pressure'], data\_dict['sea\_salinity'], kind='linear', fill\_value='extrapolate')  
 salinity\_value = salinity\_interp(pressure\_value)  
 return salinity\_value  
  
def calculate\_density(depth, salinity\_dict, temperature\_dict):  
 pressure\_value = depth \* 0.10045  
 salinity\_value = get\_salinity\_value\_from\_dict(salinity\_dict, pressure\_value)  
 temperature\_value = get\_salinity\_value\_from\_dict(temperature\_dict, pressure\_value)  
 density\_value = temperature\_value \* (-173.4852) + salinity\_value \* 114.9477 + 1025  
 if depth > 2000:  
 return 2928.716731784  
 return density\_value  
  
def get\_density\_at\_depth(desired\_depth, salinity\_dict, temperature\_dict):  
 # 生成深度值数组  
 depth\_values = np.arange(0, 6001, 10)  
  
 # 计算每个深度对应的密度值  
 density\_values = [calculate\_density(depth, salinity\_dict, temperature\_dict) for depth in depth\_values]  
  
 # 使用 calculate\_density 函数计算给定深度对应的密度值  
 desired\_density = calculate\_density(desired\_depth, salinity\_dict, temperature\_dict)  
  
 # 压缩密度值到指定范围  
 min\_density, max\_density = 1023, 1071  
 density\_range = max\_density - min\_density  
 compressed\_density = min\_density + density\_range \* (desired\_density - min(density\_values)) / (max(density\_values) - min(density\_values))  
  
 return compressed\_density  
  
# 提供一个深度值，例如 3000 米  
desired\_depth = 1077.88  
  
# 提供盐度和温度 CSV 文件的路径  
salinity\_csv\_path = "D:\\数模\\argo\_db\\linear\_90382204\_70-28.csv"  
temperature\_csv\_path = "D:\\数模\\argo\_db\\linear\_90382204\_68-28.csv"  
  
# 读取 CSV 数据为字典  
salinity\_dict = read\_csv\_to\_dict(salinity\_csv\_path, 'Practical salinity adjusted - psu')  
temperature\_dict = read\_csv\_to\_dict(temperature\_csv\_path, 'Sea temperature adjusted - degree\_Celsius')  
  
# 调用函数获取给定深度的压缩后密度值  
result\_density = get\_density\_at\_depth(desired\_depth, salinity\_dict, temperature\_dict)  
  
print(f"The density at depth {desired\_depth} is: {result\_density}")

def get\_salinity\_value\_from\_csv(salinity\_csv\_path, pressure\_value):  
 sea\_data = pd.read\_csv(salinity\_csv\_path)  
 sea\_pressure = sea\_data['Sea pressure - decibar'].values  
 sea\_salinity = sea\_data['Practical salinity adjusted - psu'].values  
 salinity\_interp = interp1d(sea\_pressure, sea\_salinity, kind='linear', fill\_value='extrapolate')  
 salinity\_value = salinity\_interp(pressure\_value)  
 return salinity\_value  
  
def get\_temperature\_value\_from\_csv(temperature\_csv\_path, pressure\_value):  
 sea\_data = pd.read\_csv(temperature\_csv\_path)  
 sea\_pressure = sea\_data['Sea pressure - decibar'].values  
 sea\_temperature = sea\_data['Sea temperature adjusted - degree\_Celsius'].values  
 temperature\_interp = interp1d(sea\_pressure, sea\_temperature, kind='linear', fill\_value='extrapolate')  
 temperature\_value = temperature\_interp(pressure\_value)  
 return temperature\_value  
  
  
  
def calculate\_density(depth, salinity\_csv\_path, temperature\_csv\_path):  
 pressure\_value = depth \* 0.10045  
  
 salinity\_value = get\_salinity\_value\_from\_csv(salinity\_csv\_path, pressure\_value)  
 temperature\_value = get\_temperature\_value\_from\_csv(temperature\_csv\_path, pressure\_value)  
  
 density\_value = temperature\_value \* (-173.4852) + salinity\_value \* 114.9477 + 1025  
 if depth > 2000:  
 return 2928.716731784  
 return density\_value  
  
def get\_density\_at\_depth(desired\_depth, salinity\_csv\_path, temperature\_csv\_path):  
 # 生成深度值数组  
 depth\_values = np.arange(0, 6001, 10)  
  
 # 计算每个深度对应的密度值  
 density\_values = [calculate\_density(depth, salinity\_csv\_path, temperature\_csv\_path) for depth in depth\_values]  
  
 # 使用calculate\_density函数计算给定深度对应的密度值  
 desired\_density = calculate\_density(desired\_depth, salinity\_csv\_path, temperature\_csv\_path)  
  
 # 压缩密度值到指定范围  
 min\_density, max\_density = 1023, 1071  
 density\_range = max\_density - min\_density  
 compressed\_density = min\_density + density\_range \* (desired\_density - min(density\_values)) / (max(density\_values) - min(density\_values))  
  
 return compressed\_density  
  
# 提供一个深度值，例如3000米  
# desired\_depth = 1077.88  
  
# 提供盐度和温度CSV文件的路径  
salinity\_csv\_path = "D:\\数模\\argo\_db\\linear\_90382204\_70-28.csv"  
temperature\_csv\_path = "D:\\数模\\argo\_db\\linear\_90382204\_68-28.csv"  
  
# 调用函数获取给定深度的压缩后密度值  
# result\_density = get\_density\_at\_depth(desired\_depth, salinity\_csv\_path, temperature\_csv\_path)  
  
# print(f"The density at depth {desired\_depth} is: {result\_density}")  
  
  
  
  
# 生成深度值数组  
# depth\_values = np.arange(0, 3001, 10)  
#  
# # 计算每个深度对应的密度值  
# density\_values = [calculate\_density(depth) for depth in depth\_values]  
# print(density\_values)  
# # 压缩密度值到指定范围  
# min\_density, max\_density = 1023, 1071  
# density\_range = max\_density - min\_density  
# density\_values = min\_density + density\_range \* (density\_values - min(density\_values)) / (max(density\_values) - min(density\_values))  
#  
# # 绘制深度与密度的变化曲线  
# plt.plot(depth\_values, density\_values, label='Density vs Depth')  
# plt.xlabel('Depth (m)')  
# plt.ylabel('Density')  
# plt.title('Density vs Depth')  
# plt.ylim(1023, 1071)  
# plt.yticks(np.arange(1020, 1090, 20))  
# plt.savefig("density.svg")  
# plt.show()