原始代码

import numpy as np  
import numpy as nan  
import pandas as pd  
import datetime  
  
i = 1  
count = 0  
monthdata = [] # 一个月的数据  
yeardata = [] # 存放每个月的平均值  
with open('C:/Users/28166/Desktop/环科院培训/编程任务一/Beijing/Y2013/1001A.txt', 'r') as f:  
 f.readline() # 跳过第一行  
 while i <= 12:  
 for line in f:  
 # 获取时间字符串  
 time\_str = line.split()[0]  
 # 将时间字符串转换为datetime对象  
 time\_obj = datetime.datetime.strptime(time\_str, '%Y-%m-%d')  
 # 使用strip()方法去除行末的换行符,然后使用split()方法将时间和数据分开,并取第一列数据  
 data = line.strip().split()[1]  
 data = float(data)  
 if data == -999.000:  
 data = np.nan  
 # 判断时间是否符合要求  
 if time\_obj.month == i:  
 # 计算有效天数  
 if data != np.nan:  
 count = count + 1  
 monthdata.append(data)  
 if time\_obj.month == 12 and time\_obj.day == 31:  
 # 若有效天数小于20天，则这个月的平均值设置为缺测  
 if count < 20:  
 emonth = np.nan  
 else:  
 emonth = np.nanmean(monthdata) # 算每个月的平均值  
 yeardata.append(emonth)  
 else:  
 if count < 20:  
 emonth = np.nan  
 else:  
 emonth = np.nanmean(monthdata)  
 yeardata.append(emonth)  
 monthdata.clear()  
 count = 0  
 if time\_obj.day == 1 :  
 if data != np.nan:  
 count = count + 1  
 monthdata.append(data)  
 i = i+1  
print(yeardata)

# 最新版功能最全的代码  
i = 1  
count = 0  
emonth = 0  
monthdata = [] # 一个月的数据  
yeardata = [] # 存放每个月的平均值  
with open('C:/Users/28166/Desktop/环科院培训/编程任务一/Beijing/Y2013/1001A.txt', 'r') as f:  
 f.readline() # 跳过第一行  
 while i <= 12:  
 for line in f:  
 # 获取时间字符串  
 time\_str = line.split()[0]  
 # 将时间字符串转换为datetime对象  
 time\_obj = datetime.datetime.strptime(time\_str, '%Y-%m-%d')  
 # 使用strip()方法去除行末的换行符,然后使用split()方法将时间和数据分开,并取第一列数据  
 data = line.strip().split()[1]  
 data = float(data)  
 # 判断时间是否符合要求  
 if time\_obj.month == i:  
 if data != -999.000:  
 count = count + 1  
 if data == -999.000:  
 data = np.nan  
 # 如果先做赋值nan，那么假设读的数据确实是-999，会先被赋值成nan然后再做后面的“判断是否等于-999”，这个条件永远不成立，所以所有  
 monthdata.append(data)  
 if time\_obj.month == 12 and time\_obj.day == 31:  
 if count < 20:  
 emonth = np.nan  
 else:  
 emonth = np.nanmean(monthdata) # 算每个月的平均值  
 yeardata.append(emonth)  
 print(yeardata)  
  
  
 else:  
 if count < 20:  
 emonth = np.nan  
 else:  
 emonth = np.nanmean(monthdata)  
 yeardata.append(emonth)  
 print(yeardata)  
 monthdata.clear()  
 count = 0  
 if data != -999.000:  
 count = count + 1  
 if data == -999.000:  
 data = np.nan  
 monthdata.append(data)  
 i = i + 1

import os

i = 1

count = 0

emonth = 0

monthdata = [] # 一个月的数据

yeardata = [] # 存放每个月的平均值

folder\_path = 'D:/Beijing/Y2013'

files = os.listdir(folder\_path)

files.sort() # 对文件名进行排序

for file\_name in files:

file\_path = os.path.join(folder\_path, file\_name)

with open(file\_path, 'r') as f:

# 逐个读取文件

'''content = f.read()

print(content)'''

f.readline() # 跳过第一行

import os

folder\_path = "/your/folder/path" # 文件夹路径

for filename in os.listdir(folder\_path):

if filename.endswith(".txt"): # 仅处理txt文件

file\_path = os.path.join(folder\_path, filename)

with open(file\_path, "r") as file:

lines = file.readlines()

with open(file\_path, "w") as file:

for line in lines:

line = line.replace("Y", "") # 删除Y字符（大小写敏感）

file.write(line)

# 在自己电脑上的可完整运行的代码（在最后部分的计算略有问题，算的不是每个站点取出同一个月，算的是一个站点12个月的平均值（没啥意义））

import numpy as np  
import numpy as nan  
import pandas as pd  
import datetime  
import os  
import matplotlib.pyplot as plt  
import matplotlib.dates as mdates  
  
# 遍历根目录下的所有文件夹和文件,,批量去除二级文件夹下txt文件的Y  
for root, dirs, files in os.walk('C:/Users/28166/Desktop/环科院培训/编程任务一/Beijing'): # 在beijing这个文件夹疯狂遍历  
 for dir in dirs: # dir是Y2013文件夹  
 dir\_path = os.path.join(root, dir)  
 for filename in os.listdir(dir\_path):  
 if filename.endswith(".txt"): # 仅处理txt文件  
 file\_path = os.path.join(dir\_path, filename)  
 with open(file\_path, "r") as file:  
 lines = file.readlines()  
 with open(file\_path, "w") as file:  
 for line in lines:  
 line = line.replace("Y", "") # 删除Y字符（大小写敏感）  
 file.write(line)  
  
totaldata = []  
  
# 遍历根目录下的所有文件夹和文件  
for root, dirs, files in os.walk('C:/Users/28166/Desktop/环科院培训/编程任务一/Beijing'): # 在beijing这个文件夹疯狂遍历  
 # 遍历二级文件夹  
 for dir in dirs: # dir是Y2013文件夹  
 # 拼接二级文件夹路径  
 dir\_path = os.path.join(root, dir) # 在这个级别，接下来要跑12个txt  
 yeardata = [] # 放12个站点1年的数据，共144个数  
 # 遍历二级文件夹下的所有文件  
 for file in os.listdir(dir\_path):  
 # 拼接文件路径  
 file\_path = os.path.join(dir\_path, file) # beijing-Y2013-接着12个文本文件  
 # 处理文件  
 i = 1  
 count = 0  
 emonth = 0  
 monthdata = [] # 一个月的数据  
 with open(file\_path, 'r') as f:  
 # 逐个读取文件  
 '''content = f.read()  
 print(content)'''  
 f.readline() # 跳过第一行  
 for line in f:  
 # 获取时间字符串  
 time\_str = line.split()[0]  
 # 将时间字符串转换为datetime对象  
 time\_obj = datetime.datetime.strptime(time\_str, '%Y-%m-%d')  
 # 使用strip()方法去除行末的换行符,然后使用split()方法将时间和数据分开,并取第一列数据  
 data = line.strip().split()[1]  
 data = float(data)  
 # 判断时间是否符合要求  
 if time\_obj.month == i:  
 if data != -999.000:  
 count = count + 1  
 if data == -999.000:  
 data = np.nan  
 monthdata.append(data)  
 if time\_obj.month == 12 and time\_obj.day == 31:  
 if count < 20:  
 emonth = np.nan  
 else:  
 emonth = np.nanmean(monthdata) # 算每个月的平均值  
 yeardata.append(emonth)  
 else:  
 if count < 20:  
 emonth = np.nan  
 else:  
 emonth = np.nanmean(monthdata)  
 yeardata.append(emonth)  
 monthdata.clear()  
 count = 0  
 if data != -999.000:  
 count = count + 1  
 if data == -999.000:  
 data = np.nan  
 monthdata.append(data)  
 i = i + 1  
 # 定义新列表  
 new\_list = [] #  
  
 # 循环遍历长列表  
 for i in range(0, len(yeardata), 12):  
 # 取出每间隔12个数据的子列表  
 sub\_list = yeardata[i:i+12]  
 # 对子列表进行求和  
 sum\_value = np.nanmean(sub\_list)  
 # 将求和结果存入新列表中  
 new\_list.append(sum\_value)  
 # 输出新列表  
 print(new\_list)  
 totaldata.append(new\_list)  
print(totaldata)  
  
# 二维列表变成一维  
totaldata = [num for row in totaldata for num in row]  
print(totaldata)  
print(len(totaldata))  
  
#画图  
time\_index = pd.date\_range('01/2013', '12/2021', freq='MS')  
y = totaldata  
# 创建画布和子图  
fig, ax = plt.subplots()  
# 设置横轴坐标格式为年月  
ax.xaxis.set\_major\_formatter(mdates.DateFormatter('%Y-%m'))  
# 绘制折线图  
ax.plot(time\_index, y, 'o-')  
# 设置横轴坐标刻度  
ax.xaxis.set\_major\_locator(mdates.MonthLocator(interval=1))  
  
# 设置横纵坐标的名称以及对应字体格式  
font2 = {'family': 'Times New Roman', 'weight': 'normal', 'size': 14, }  
plt.xlabel('Year-Month', font2)  
plt.ylabel('PM25(ug/m3)', font2)  
  
# 横纵轴刻度一起设置  
labels = ax.get\_xticklabels() + ax.get\_yticklabels()  
# 设置横轴刻度旋转程度为30度  
[label.set\_rotation(75) for label in ax.get\_xticklabels()]  
# 刻度对应字体格式  
[label.set\_fontname('Times New Roman') for label in labels]  
  
plt.title('Monthly average PM2.5 concentration variation curve at 12 stations in Beijing', fontdict={'family': 'Times New Roman', 'size': 20, })  
# 显示图形  
plt.show()

# 最最最终版任务一代码（这回彻底没问题了）

9年各年月平均值变化

import numpy as np  
import numpy as nan  
import pandas as pd  
import datetime  
import os  
import matplotlib.pyplot as plt  
import matplotlib.dates as mdates  
  
totaldata = []  
  
# 遍历根目录下的所有文件夹和文件  
for root, dirs, files in os.walk('C:/Users/28166/Desktop/环科院培训/编程任务一/Beijing'):  
 # 遍历二级文件夹  
 for dir in dirs:  
 # 拼接二级文件夹路径  
 dir\_path = os.path.join(root, dir)  
 yeardata = [] # 存放每个月的平均值  
 # 遍历二级文件夹下的所有文件  
 for file in os.listdir(dir\_path):  
 # 拼接文件路径  
 file\_path = os.path.join(dir\_path, file)  
 # 处理文件  
 i = 1  
 count = 0  
 emonth = 0  
 monthdata = [] # 一个月的数据  
  
 with open(file\_path, 'r') as f:  
 f.readline() # 跳过第一行  
 for line in f:  
 # 获取时间字符串  
 time\_str = line.split()[0]  
 # 将时间字符串转换为datetime对象  
 time\_obj = datetime.datetime.strptime(time\_str, '%Y-%m-%d')  
 # 使用strip()方法去除行末的换行符,然后使用split()方法将时间和数据分开,并取第一列数据  
 data = line.strip().split()[1]  
 data = float(data)  
 # 判断时间是否符合要求  
 if time\_obj.month == i:  
 if data != -999.000:  
 count = count + 1  
 if data == -999.000:  
 data = np.nan  
 monthdata.append(data)  
 if time\_obj.month == 12 and time\_obj.day == 31:  
 if count < 20:  
 emonth = np.nan  
  
 else:  
 emonth = np.nanmean(monthdata) # 算每个月的平均值  
 yeardata.append(emonth)  
  
 else:  
 if count < 20:  
 emonth = np.nan  
  
 else:  
 emonth = np.nanmean(monthdata)  
 yeardata.append(emonth)  
 monthdata.clear()  
 count = 0  
 if data != -999.000:  
 count = count + 1  
 if data == -999.000:  
 data = np.nan  
 monthdata.append(data)  
 i = i + 1  
 print(yeardata)  
 print(len(yeardata))  
 # 定义新列表  
 new\_list = []  
  
 # 循环遍历长列表  
 for i in range(0, len(yeardata), 12):  
 # 取出每间隔12个数据的子列表  
 sub\_list = yeardata[i:i+12] # sub\_list是每相邻12个数捆在一起的子列表  
 # 将求平均结果存入新列表中  
 new\_list.append(sub\_list) # 把子列表append进去之后，new\_list的元素仍是以列表的形式存在的，意思就是new\_list是个二维列表  
 # 求出12个站点一年每个月的平均  
  
 last\_list = [np.nanmean(x) for x in zip(\*new\_list)] # 二维列表按列求和  
 print(last\_list)  
 print(len(last\_list))  
 totaldata.append(last\_list)  
  
print(totaldata)  
print(len(totaldata))  
  
# 二维列表变成一维  
totaldata = [num for row in totaldata for num in row]  
print(totaldata)  
print(len(totaldata))  
  
# 统计数据中有多少个nan，（不能直接用for i in totaldata:if totaldata[i] == np.nan），要用专门的函数  
a = np.array(totaldata)  
print(np.isnan(a).sum())  
  
#画图  
time\_index = pd.date\_range('01/2013', '12/2021', freq='MS')  
y = totaldata  
# 创建画布和子图  
fig, ax = plt.subplots()  
# 设置横轴坐标格式为年月  
ax.xaxis.set\_major\_formatter(mdates.DateFormatter('%Y-%m'))  
# 绘制折线图  
ax.plot(time\_index, y, 'o-')  
# 设置横轴坐标刻度  
ax.xaxis.set\_major\_locator(mdates.MonthLocator(interval=1))  
  
# 设置横纵坐标的名称以及对应字体格式  
font2 = {'family': 'Times New Roman', 'weight': 'normal', 'size': 14, }  
  
# 横纵轴刻度一起设置  
labels = ax.get\_xticklabels() + ax.get\_yticklabels()  
# 设置横轴刻度旋转程度为30度  
[label.set\_rotation(75) for label in ax.get\_xticklabels()]  
# 刻度对应字体格式  
[label.set\_fontname('Times New Roman') for label in labels]  
  
plt.xlabel('Year-Month', font2)  
plt.ylabel('PM25(ug/m3)', font2)  
  
plt.title('Monthly average PM2.5 concentration variation curve at 12 stations in Beijing', fontdict={'family': 'Times New Roman', 'size': 20, })  
  
# 显示图形  
plt.show()

# 求二维列表各个子列表的平均值（含nan）

data = [[1,2,np.nan,4,5, 6],  
 [np.nan,4,6,8,10,12],  
 [1, 3,5,7,9,11],  
 [1.1,2.2,3.3,4.4,5.5,6.6],  
 [3,6,9,12,15,18],  
 [4,8,12,16,20,24],  
 [5,10,15,20,25,30],  
 [1.5, 2.5,3.5,4.5, 5.5, 6.5],  
 [1, 4, 7, 10, 13, 16],  
 [2, 5, 8, 11, 14, 17]]  
y = [np.mean(i) for i in data]

（2）# 计算超标天数并分析变化特征

# 每年一级超标天数  
total1 = []  
# 每年二级超标天数  
total2 = []  
year = 2013  
# 遍历根目录下的所有文件夹和文件  
for root, dirs, files in os.walk('C:/Users/28166/Desktop/环科院培训/编程任务一/Beijing'):  
 # 遍历二级文件夹  
 for dir in dirs:  
 dir\_path = os.path.join(root, dir)  
 # 一年12\*365/366的所有数据  
 totaldata = []  
 day1 = 0  
 day2 = 0  
 if year % 400 == 0 or (year % 4 == 0 and year % 100 != 0):  
 sum = 366  
 else:  
 sum = 365  
 for file in os.listdir(dir\_path):  
 file\_path = os.path.join(dir\_path, file)  
 # 处理文件  
 with open(file\_path, 'r') as f:  
 f.readline() # 跳过第一行  
 for line in f:  
 # 获取时间字符串  
 time\_str = line.split()[0]  
 # 将时间字符串转换为datetime对象  
 time\_obj = datetime.datetime.strptime(time\_str, '%Y-%m-%d')  
 # 使用strip()方法去除行末的换行符,然后使用split()方法将时间和数据分开,并取第一列数据  
 data = line.strip().split()[1]  
 data = float(data)  
 # 判断时间是否符合要求  
 if data == -999.000:  
 data = np.nan  
 totaldata.append(data)  
 # 定义新列表  
 new\_list = []  
 # 循环遍历长列表  
 for i in range(0, len(totaldata), sum):  
 # 取出每间隔12个数据的子列表  
 sub\_list = totaldata[i:i + sum] # sub\_list是每相邻12个数捆在一起的子列表  
 new\_list.append(sub\_list) # 把子列表append进去之后，new\_list的元素仍是以列表的形式存在的，意思就是new\_list是个二维列表  
 year = year +1  
 last\_list = [np.nanmean(x) for x in zip(\*new\_list)] # 二维列表按列求和  
 print(last\_list)  
 print(len(last\_list))  
 # 在跑这一年数据的数据的时候把本年度的一级超标天和二级超标天统计出来  
 for i in last\_list:  
 if i > 35 :  
 day1 = day1 + 1  
 if i > 75:  
 day2 = day2 + 1  
 print(day1)  
 print(day2)  
 total1.append(day1)  
 total2.append(day2)  
print(total1)  
print(total2)

# 绘制折线图  
x = [2013,2014,2015,2016,2017,2018,2019,2020,2021]  
fig, ax = plt.subplots()  
ax.plot(x, total1, 'o-', label = 'first(>35)')  
ax.plot(x, total2, 'o-', label = 'second(>75)')  
  
# 设置横轴坐标刻度  
  
plt.title('Trend of days exceeding the standard', fontdict={'family': 'Times New Roman', 'size': 20, })  
# 设置横纵坐标的名称以及对应字体格式  
font2 = {'family': 'Times New Roman', 'weight': 'normal', 'size': 14, }  
plt.xlabel('Year', font2)  
plt.ylabel('Number of days', font2)  
# 横纵轴刻度一起设置  
labels = ax.get\_xticklabels() + ax.get\_yticklabels()  
# 设置横轴刻度旋转程度为30度  
[label.set\_rotation(75) for label in ax.get\_xticklabels()]  
# 刻度对应字体格式  
[label.set\_fontname('Times New Roman') for label in labels]  
  
ax.legend()  
# 显示图形  
plt.show()

画子图

fig, ax = plt.subplots(1,2)  
# 绘制折线图  
ax[0].set\_title('Trend of days exceeding the standard', fontdict={'family': 'Times New Roman', 'size': 20, })  
font2 = {'family': 'Times New Roman', 'weight': 'normal', 'size': 14, }  
ax[0].set\_xlabel('Year', font2)  
ax[0].set\_ylabel('Number of days', font2)  
ax[0].plot(x, total1, 'o-', label = 'first(>35)')  
ax[0].plot(x, total2, 'o-', label = 'second(>75)')  
ax[0].plot(x, total3, 'o-', label = 'middle(35<c<75)')

（3）

画年平均值变化趋势

# 每年一级超标天数  
total1 = []  
# 每年二级超标天数  
total2 = []  
yearmean = []  
year = 2013  
# 遍历根目录下的所有文件夹和文件  
for root, dirs, files in os.walk('C:/Users/28166/Desktop/环科院培训/编程任务一/Beijing'):  
 # 遍历二级文件夹  
 for dir in dirs:  
 dir\_path = os.path.join(root, dir)  
 # 一年12\*365/366的所有数据  
 totaldata = []  
 day1 = 0  
 day2 = 0  
 if year % 400 == 0 or (year % 4 == 0 and year % 100 != 0):  
 sum = 366  
 else:  
 sum = 365  
 for file in os.listdir(dir\_path):  
 file\_path = os.path.join(dir\_path, file)  
 # 处理文件  
 with open(file\_path, 'r') as f:  
 f.readline() # 跳过第一行  
 for line in f:  
 # 获取时间字符串  
 time\_str = line.split()[0]  
 # 将时间字符串转换为datetime对象  
 time\_obj = datetime.datetime.strptime(time\_str, '%Y-%m-%d')  
 # 使用strip()方法去除行末的换行符,然后使用split()方法将时间和数据分开,并取第一列数据  
 data = line.strip().split()[1]  
 data = float(data)  
 # 判断时间是否符合要求  
 if data == -999.000:  
 data = np.nan  
 totaldata.append(data)  
 # 定义新列表  
 new\_list = []  
 # 循环遍历长列表  
 for i in range(0, len(totaldata), sum):  
 # 取出每间隔12个数据的子列表  
 sub\_list = totaldata[i:i + sum] # sub\_list是每相邻12个数捆在一起的子列表  
 new\_list.append(sub\_list) # 把子列表append进去之后，new\_list的元素仍是以列表的形式存在的，意思就是new\_list是个二维列表  
 year = year +1  
 last\_list = [np.nanmean(x) for x in zip(\*new\_list)] # 二维列表按列求和  
 print(last\_list)  
 print(len(last\_list))  
  
 yearmeanvalue = np.nanmean(last\_list)  
 yearmean.append(yearmeanvalue)  
  
 # 在跑这一年数据的数据的时候把本年度的一级超标天和二级超标天统计出来  
 for i in last\_list:  
 if i > 35 :  
 day1 = day1 + 1  
 if i > 75:  
 day2 = day2 + 1  
 print(day1)  
 print(day2)  
 print(day3)  
 total1.append(day1)  
 total2.append(day2)  
print(total1)  
print(total2)  
print(yearmean)  
# 每个数保留两位小数  
round\_yearmean = np.round(yearmean,2)  
print(round\_yearmean)  
x = [2013,2014,2015,2016,2017,2018,2019,2020,2021]  
fig, ax = plt.subplots()  
ax.plot(x, yearmean, 'o-')  
plt.title('Trend of Annual Average of PM2.5', fontdict={'family': 'Times New Roman', 'size': 20, })  
# 设置横纵坐标的名称以及对应字体格式  
font2 = {'family': 'Times New Roman', 'weight': 'normal', 'size': 14, }  
plt.xlabel('Year', font2)  
plt.ylabel('Annual Average of PM2.5', font2)  
# 横纵轴刻度一起设置  
labels = ax.get\_xticklabels() + ax.get\_yticklabels()  
# 设置横轴刻度旋转程度为30度  
# 刻度对应字体格式  
[label.set\_fontname('Times New Roman') for label in labels]  
plt.show()