第六次作业

数据分析类练习

经济管理中通常有大量的数据以excel格式存在，如本次作业提供的中国各省长周期CO2排放数据（资源/data/co2_demo.zip)。数据格式如下：所有数据按年份存储在不同的文件中,Province_sectoral_CO2_emissions_20xx.xlsx，其中20xx为年份。单个excel文件中，"sum"数据页给出了各省的总CO_2排放量以及来源明细，其余以省命名的数据页则给出了各省不同行业不同来源的CO_2排放量。

前期准备

```
from typing import Text
import pandas as pd
import os
from matplotlib import pyplot as plt
import numpy as np

#windows 中文字显示问题
#plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
#plt.rcParams['axes.unicode_minus']=False #用来正常显示负号

#mac 中文字显示问题
plt.rcParams['font.sans-serif'] = ['Arial Unicode MS']

#file_path = 'E:\何熙1908\大三上课程\现代程序设计\第六次作业\co2_demo\'
```

1. 至少实现一个数据分析类，以提供数据的读取及基本的时间（如某区域某类型排放随时间的变化）和空间分析（某一年全国排放的空间分布态势）方法。

```
class Data_analyze:
    #  至少实现一个数据分析类，以提供数据的读取及基本的时间
    #  （如某区域某类型排放随时间的变化）和空间分析（某一年全国排放的空间分布态势）方法。
    def __init__(self,path_list):
        self.df_list = []
        self.time_list = []
        for i in path_list:
            df = pd.read_excel('co2_demo/'+i,sheet_name=None)
```

```
            self.time_list.append(i[-9:-5])
            self.df_list.append(df)
        self.province_list = list(self.df_list[0]['Sum'][self.df_list[0]
['Sum'].columns[0]])[0:-2]


    def time_analyze(self,province,co2_type):
        self.time_data_list = []
        for i in self.df_list:
            df_time = i['Sum']
            #print(df_time)
            province_list = list(df_time[df_time.columns[0]])
            #print(nation)
            data = df_time.loc[province_list.index(province),co2_type]
            #print(data)
            self.time_data_list.append(data)

        print(f'省份：{province}  碳排放量类型：{co2_type}')
        for i in range(len(self.time_list)):
            print(f'年份：{self.time_list[i]}  排放量：{self.time_data_list[i]}')

        return self.time_data_list


    def space_analyze(self,year,co2_type):
        df_space = self.df_list[self.time_list.index(year)]['Sum']
        #print(df_space)
        self.space_data_list = []
        province_list = list(df_space[df_space.columns[0]])
        for i in range(len(province_list)):
            data = df_space.loc[i, co2_type]
            self.space_data_list.append(data)

        print(f'年份：{year}  碳排放量类型：{co2_type}')
        for i in range(0,len(province_list)-2):
            print(f'省份：{province_list[i]}  排放量：{self.space_data_list[i]}')
        print(f'总计：{province_list[len(province_list)-1]}  排放量：
{self.space_data_list[len(province_list)-1]}')

        return self.space_data_list[0:-2]
```

2. 至少实现一个数据可视化类，以提供上述时空分析结果的可视化，如以曲线、饼等形式对结果进行呈现。

```
class Data_view(Data_analyze):
    def __init__(self,path_list):
        Data_analyze.__init__(self,path_list)
```

```
    def time_view(self):
        time_list = self.time_list
        for i in self.province_list:
            time_data_list = self.time_analyze(i,'Total')
            #print(time_data_list)
            plt.figure(figsize=(20,6.5))
            plt.bar(range(len(time_list)),time_data_list,tick_label =
time_list,color = 'violet')
            plt.xlabel('年份')
            plt.ylabel(i+' Co2 排放量')
            plt.savefig('figure/'+i+'_total.png')

    def space_view(self):
        province_list = self.province_list
        space_data_list = np.array(self.space_analyze('1997','Total'))
        plt.pie(space_data_list,
                labels=province_list,
                autopct='%.2f%%'
                )
        plt.title("Beijing Co2 排放量 总体占比情况")
        plt.show()
```

3. 由于数据中包含空值等异常值，在进行数据分析以及可视化前需要检查数据。因此需要实现NotNumError类，继承ValueError，并加入新属性year，province，industry，type，对数据进行检测，若取到的一列数据中包含nan，则抛出该异常，并提供异常相关的年份，省份，工业和排放类型等信息。在此基础上，利用try except捕获该异常，打印异常信息，并对应位置的数据进行适当的填充。

```
# 由于数据中包含空值等异常值，在进行数据分析以及可视化前需要检查数据。因此需要实现NotNumError类，
# 继承ValueError，并加入新属性year，province，industry，type，对数据进行检测，若取到的一列数
据中包含nan，则抛出该异常，
# 并提供异常相关的年份，省份，工业和排放类型等信息。
# 在此基础上，利用try except捕获该异常，打印异常信息，并对应位置的数据进行适当的填充。

class NotnumError(ValueError):
    def __init__(self,year,province,industry,type):
        self.year = year
        self.province = province
        self.industry = industry
        self.type = type
        self.message = f"the data of {province} in {year} has nan about {industry}
and the type is {type}"

class NotnumberTest(Data_analyze):
    def __init__(self,path_list):
```

```
        Data_analyze.__init__(self,path_list)

    def read_data(self):
        doc_list = self.df_list
        year_list = self.time_list
        province_list = self.province_list
        for y in range(len(doc_list)):
            self._year = year_list[y]
            for sheet in doc_list[y]:
                if sheet != 'Sum':
                    self._province = sheet
                    df_temp = doc_list[y][sheet]
                    row_index = list(df_temp[df_temp.columns[0]])
                    col_index = list(df_temp.columns)
                    #print(col_index)
                    values = df_temp.values
                    for industry_index in range(2,len(row_index)):
                        if not pd.isnull(row_index[industry_index]):
                            self._industry = row_index[industry_index]
                            for type_index in range(1,len(col_index)-3):
                                if not pd.isnull(col_index[type_index]):
                                    self._type = col_index[type_index]
                                    if pd.isnull(values[industry_index]
[type_index]):
                                        raise
NotnumError(self._year,self._province,self._industry,self._type)
                                    else:
                                        continue
                    #print(sheet)
        #print(len(doc_list))
```

4. 由于部分省份排放总量数据为0，要求在计算比例时进行检验，若检验发现总量为0，则抛出ZeroDivisionError，并打印对应的行名等信息。

```
# 4. 由于部分省份排放总量数据为0，要求在计算比例时进行检验，若检验发现总量为0，
# 则抛出ZeroDivisionError，并打印对应的行名等信息。

class Zero_test(Data_analyze):
    def __init__(self, path_list):
        super().__init__(path_list)

    def zero_test(self):
        doc_list = self.df_list
        year_list = self.time_list
        province_list = self.province_list
        for y in range(len(doc_list)):
            self._year = year_list[y]
```

```
            df_year = doc_list[year_list.index(self._year)]['Sum']
            col_index = list(df_year.columns)
            values = df_year.values
            #print(col_index)
            for i in range(len(province_list)):
                self._province = province_list[i]
                for j in range(1,len(col_index)):
                    self._type = col_index[j]
                    if values[i][j] == 0:
                        #print(f"the total data of {self._province} has a zero in
{self._year} and the type is {self._type}.")
                        raise ZeroDivisionError
                    else:
                        pass
```

5. # 5. （附加）按时间分析时，注意观察不同区域随时间排放量的变化，

   # 是否存在一些明显的趋势，以及趋势的空间差异，并思考这些趋势及差异的管理意义与政策启发。

- 从各省的时间分布数据中可以看出整体碳排放呈上升趋势，碳排放在某种意义上代表着经济发展，在最近几年碳排放的上升趋势逐渐放缓，意味着我国在解决碳排放问题保护环境方面做出了很多努力。

代码测试

```
def main():
    path_list = os.listdir('co2_demo')
    path_list = sorted(path_list)
    print(path_list)

    Data = Data_analyze(path_list)

    Data.time_analyze('Beijing','Total')


    print(Data.time_list)
    print(Data.df_list)


    Data.space_analyze('1997','Total')


    Data_draw = Data_view(path_list)

    Data_draw.time_view()
    Data_draw.space_view()
```

```
    Text = NotnumberTest(path_list)


    try:
        Text.read_data()
    except NotnumError as Nn:
        print(Nn.message)
    else:
        print('There is no nan in data.')


    Zero = Zero_test(path_list)
    try:
        Zero.zero_test()
    except ZeroDivisionError as ZD:
        print(f"the total data of {Zero._province} has a zero in {Zero._year} and the
type is {Zero._type}.")
    else:
        print("There is no zero in data")

main()
```

结果展示

```
['Province sectoral CO2 emissions 1997.xlsx', 'Province sectoral CO2 emissions
1998.xlsx', 'Province sectoral CO2 emissions 1999.xlsx', 'Province sectoral CO2
emissions 2000.xlsx', 'Province sectoral CO2 emissions 2001.xlsx', 'Province sectoral
CO2 emissions 2002.xlsx', 'Province sectoral CO2 emissions 2003.xlsx', 'Province
sectoral CO2 emissions 2004.xlsx', 'Province sectoral CO2 emissions 2005.xlsx',
'Province sectoral CO2 emissions 2006.xlsx', 'Province sectoral CO2 emissions
2007.xlsx', 'Province sectoral CO2 emissions 2008.xlsx', 'Province sectoral CO2
emissions 2009.xlsx', 'Province sectoral CO2 emissions 2010.xlsx', 'Province sectoral
CO2 emissions 2011.xlsx', 'Province sectoral CO2 emissions 2012.xlsx', 'Province
sectoral CO2 emissions 2013.xlsx', 'Province sectoral CO2 emissions 2014.xlsx',
'Province sectoral CO2 emissions 2015.xlsx']
```

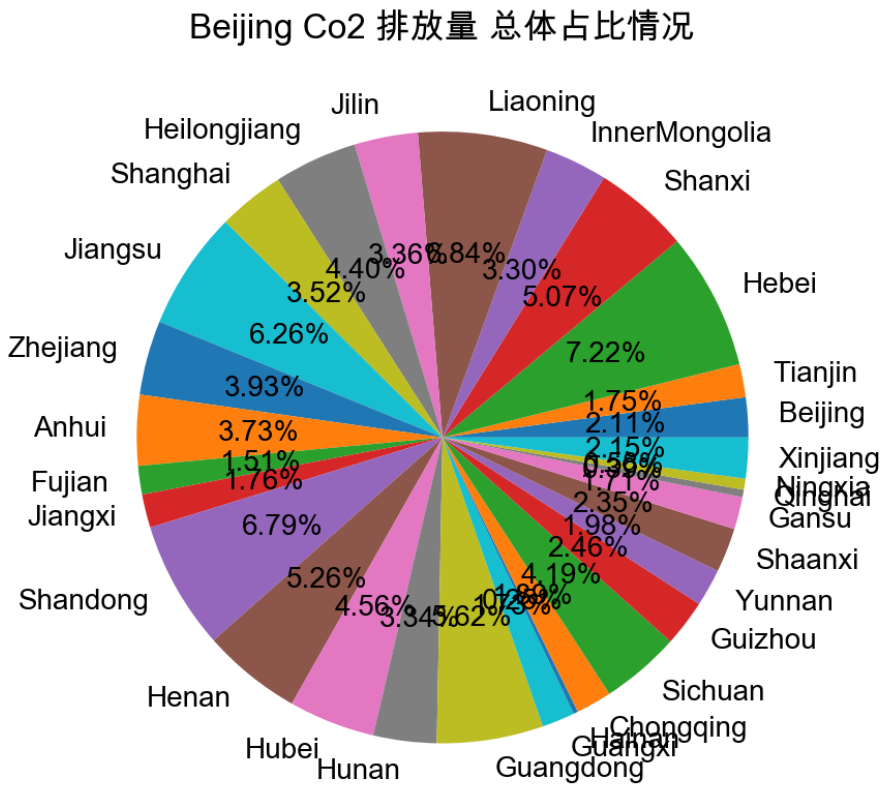
```
省份: Beijing  碳排放量类型: Total
年份: 1997  排放量: 61.9
年份: 1998  排放量: 63.3
年份: 1999  排放量: 66.6
年份: 2000  排放量: 68.2
年份: 2001  排放量: 77.4
年份: 2002  排放量: 77.9
```

```
年份：2003  排放量：82.0
年份：2004  排放量：88.1
年份：2005  排放量：92.1
年份：2006  排放量：96.7
年份：2007  排放量：102.9
年份：2008  排放量：99.2
年份：2009  排放量：100.4
年份：2010  排放量：103.0
年份：2011  排放量：94.4
年份：2012  排放量：97.2
年份：2013  排放量：93.4
年份：2014  排放量：92.5
年份：2015  排放量：95.2
['1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004', '2005', '2006',
'2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015']

年份：1997  碳排放量类型：Total
省份：Beijing  排放量：61.9
省份：Tianjin  排放量：51.4
省份：Hebei  排放量：212.1
省份：Shanxi  排放量：148.7
省份：InnerMongolia  排放量：97.0
省份：Liaoning  排放量：200.7
省份：Jilin  排放量：98.6
省份：Heilongjiang  排放量：129.1
省份：Shanghai  排放量：103.2
省份：Jiangsu  排放量：183.9
省份：Zhejiang  排放量：115.4
省份：Anhui  排放量：109.5
省份：Fujian  排放量：44.2
省份：Jiangxi  排放量：51.8
省份：Shandong  排放量：199.3
省份：Henan  排放量：154.3
省份：Hubei  排放量：133.9
省份：Hunan  排放量：98.0
省份：Guangdong  排放量：165.1
省份：Guangxi  排放量：50.8
省份：Hainan  排放量：7.2
省份：Chongqing  排放量：55.4
省份：Sichuan  排放量：123.1
省份：Guizhou  排放量：72.2
省份：Yunnan  排放量：58.0
省份：Shaanxi  排放量：68.9
省份：Gansu  排放量：50.3
省份：Qinghai  排放量：11.5
省份：Ningxia  排放量：17.1
省份：Xinjiang  排放量：63.2
总计：Sum-CO2  排放量：2935.8
```

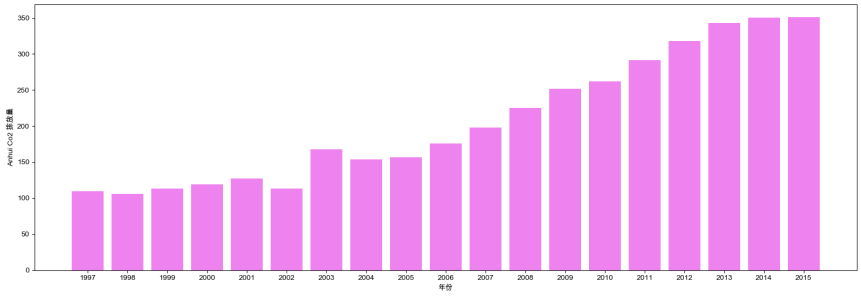
```
年份：1997  碳排放量类型：Total
省份：Beijing  排放量：61.9
省份：Tianjin  排放量：51.4
省份：Hebei  排放量：212.1
省份：Shanxi  排放量：148.7
省份：InnerMongolia  排放量：97.0
省份：Liaoning  排放量：200.7
省份：Jilin  排放量：98.6
省份：Heilongjiang  排放量：129.1
省份：Shanghai  排放量：103.2
省份：Jiangsu  排放量：183.9
省份：Zhejiang  排放量：115.4
省份：Anhui  排放量：109.5
省份：Fujian  排放量：44.2
省份：Jiangxi  排放量：51.8
省份：Shandong  排放量：199.3
省份：Henan  排放量：154.3
省份：Hubei  排放量：133.9
省份：Hunan  排放量：98.0
省份：Guangdong  排放量：165.1
省份：Guangxi  排放量：50.8
省份：Hainan  排放量：7.2
省份：Chongqing  排放量：55.4
省份：Sichuan  排放量：123.1
省份：Guizhou  排放量：72.2
省份：Yunnan  排放量：58.0
省份：Shaanxi  排放量：68.9
省份：Gansu  排放量：50.3
省份：Qinghai  排放量：11.5
省份：Ningxia  排放量：17.1
省份：Xinjiang  排放量：63.2
总计：Sum-CO2  排放量：2935.8


the data of Ningxia in 2000 has nan about Total Consumption and the type is Raw Coal.


the total data of Beijing has a zero in 1997 and the type is Other Petroleum Products.
```

可视化结果

- 空间分布


Beijing Co2 排放量 总体占比情况
Jilin
Liaoning
Heilongjiang
InnerMongolia
Shanghai
Shanxi
Jiangsu
Hebei
Zhejiang
Tianjin
Anhui
Beijing
Xinjiang
Fujian
Ningxia
Jiangxi
Qinghai
Gansu
Shaanxi
Shandong
Yunnan
Guizhou
Sichuan
Henan
Chongqing
Hubei
Hainan
Hunan
Guangdong
Guangxi
3.36%
3.84%
4.40%
3.30%
3.52%
5.07%
6.26%
7.22%
3.93%
1.75%
2.11%
3.73%
1.51%
1.76%
2.35%
6.79%
1.98%
2.46%
5.26%
4.19%
4.56%
3.34%
5.62%

- 时间分布


Anhui Co2 排放量
年份
0
50
100
150
200
250
300
350
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015