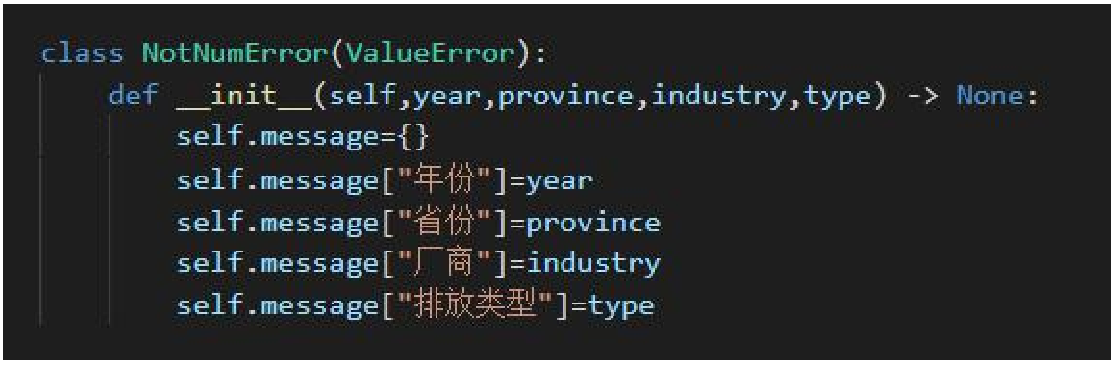
1. 错误类型定义

初始化储存相关错误信息，便于输出

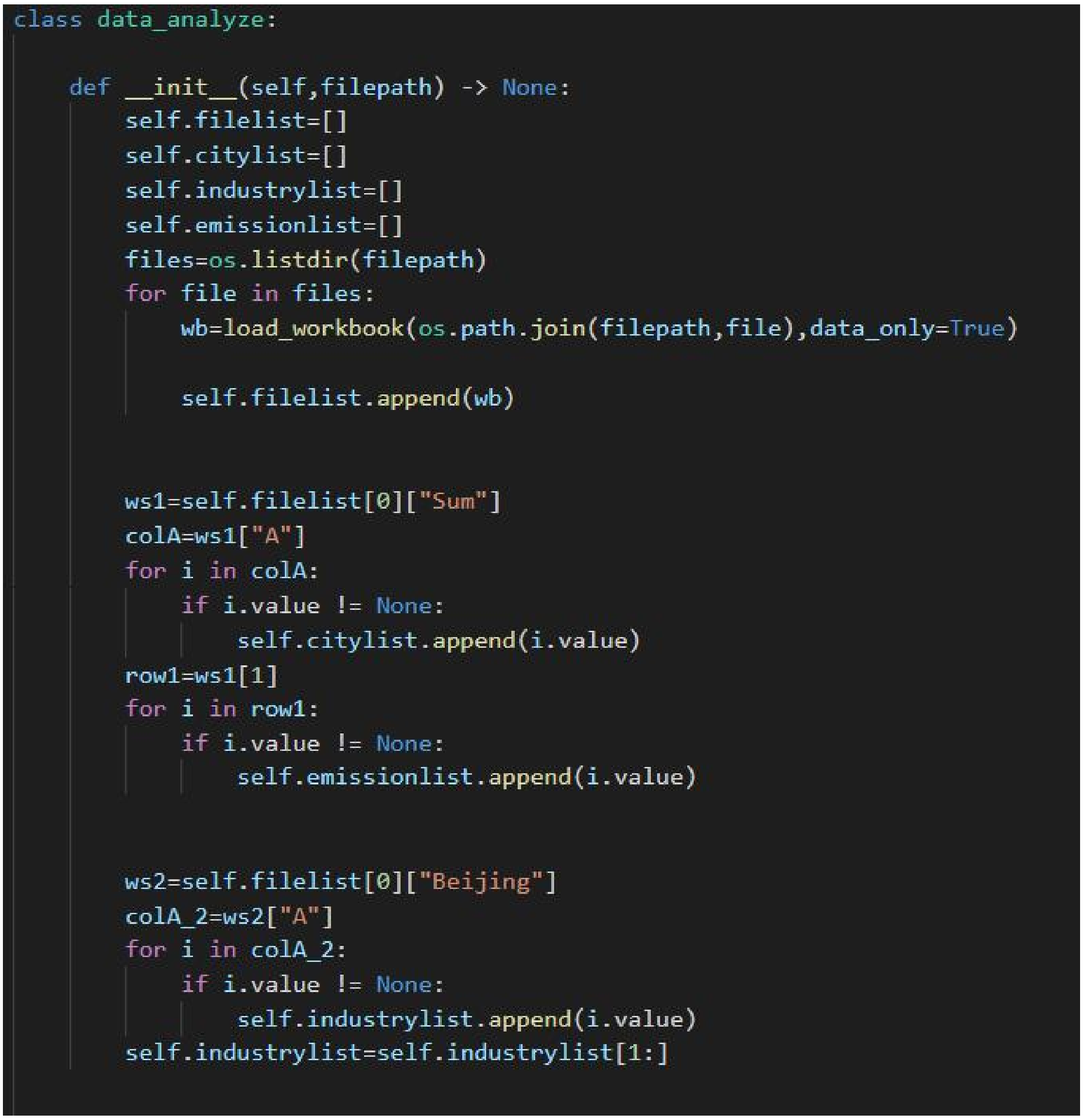


```python
class NotNumError(ValueError):
    def __init__(self,year,province,industry,type) -> None:
        self.message={}
        self.message["年份"]=year
        self.message["省份"]=province
        self.message["厂商"]=industry
        self.message["排放类型"]=type
```

2. 数据分析类

（1） 数据的加载和初始化

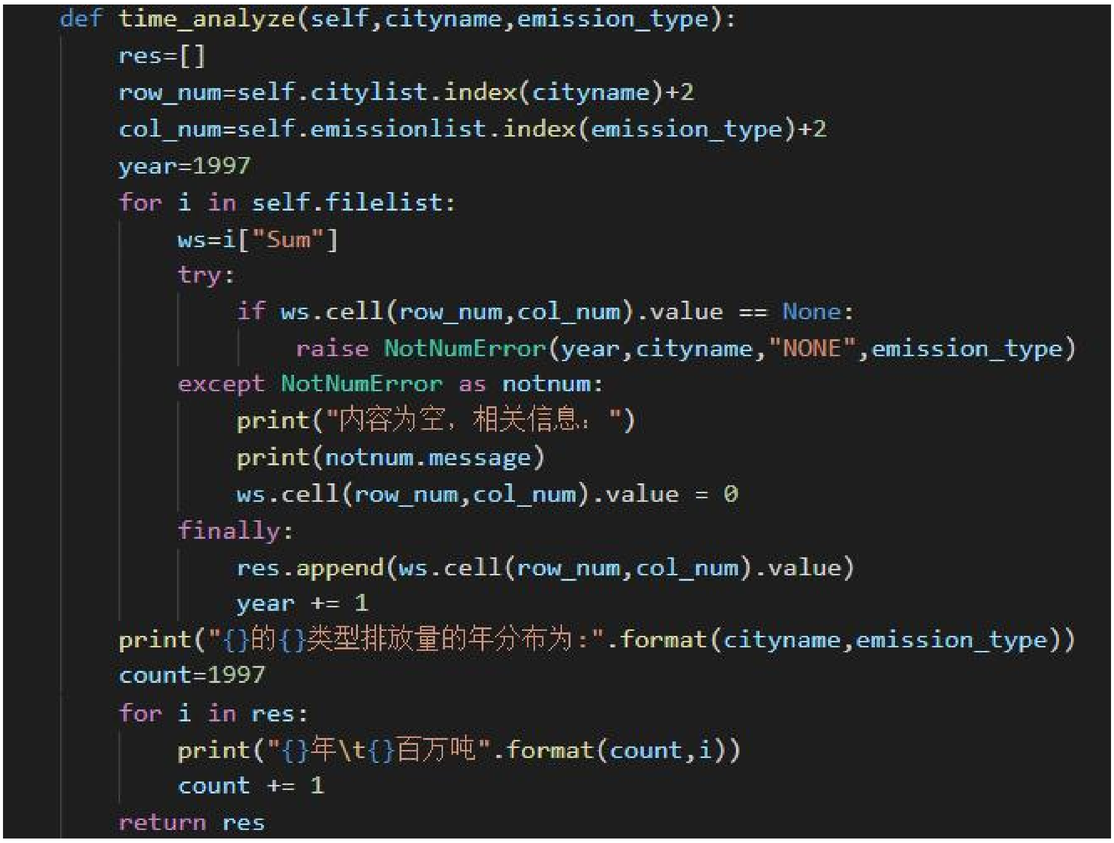
对所有的 excel 文档进行加载读取，并储存在列表中。同时分别储存 excel 的表头数据，如城市，排放类型，工昌等数据，都用列表储存，方便后续的操作。



```python
class data_analyze:

    def __init__(self,filepath) -> None:
        self.filelist=[]
        self.citylist=[]
        self.industrylist=[]
        self.emissionlist=[]
        files=os.listdir(filepath)
        for file in files:
            wb=load_workbook(os.path.join(filepath,file),data_only=True)

            self.filelist.append(wb)


        ws1=self.filelist[0]["Sum"]
        colA=ws1["A"]
        for i in colA:
            if i.value != None:
                self.citylist.append(i.value)
        row1=ws1[1]
        for i in row1:
            if i.value != None:
                self.emissionlist.append(i.value)


        ws2=self.filelist[0]["Beijing"]
        colA_2=ws2["A"]
        for i in colA_2:
            if i.value != None:
                self.industrylist.append(i.value)
        self.industrylist=self.industrylist[1:]
```

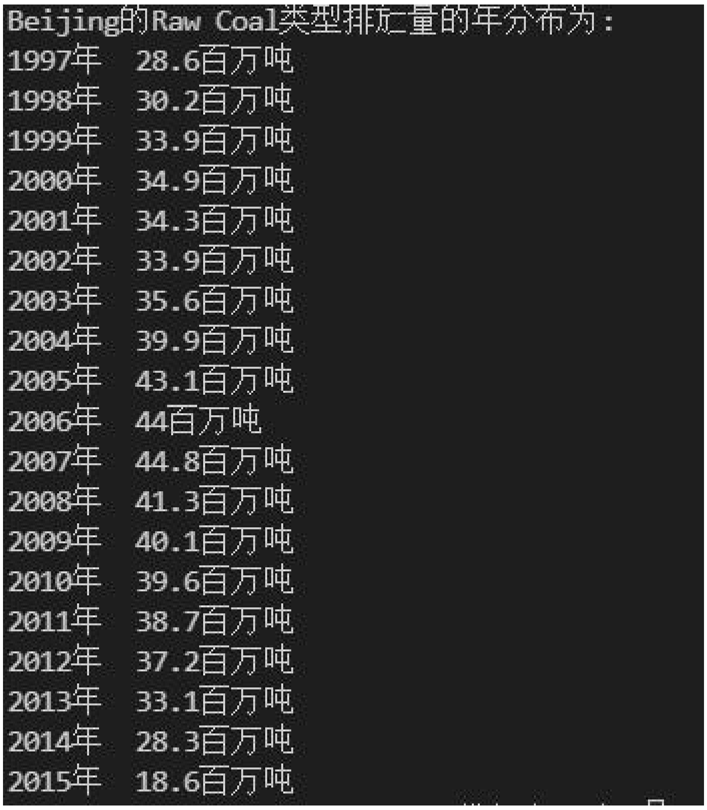
（2） 时间分布分析

根据输入的城市和排放的类型输出 1997-2015 年的 CO2 排放情况；在读取数据前判断表格内容是否为空，如果是抛出异常并且填入“0”，异常处理后再读取结果。



```
    def time_analyze(self,cityname,emission_type):
        res=[]
        row_num=self.citylist.index(cityname)+2
        col_num=self.emissionlist.index(emission_type)+2
        year=1997
        for i in self.filelist:
            ws=i["Sum"]
            try:
                if ws.cell(row_num,col_num).value == None:
                    raise NotNumError(year,cityname,"NONE",emission_type)
            except NotNumError as notnum:
                print("内容为空，相关信息：")
                print(notnum.message)
                ws.cell(row_num,col_num).value = 0
            finally:
                res.append(ws.cell(row_num,col_num).value)
                year += 1
        print("{}的{}类型排放量的年分布为:".format(cityname,emission_type))
        count=1997
        for i in res:
            print("{}年\t{}百万吨".format(count,i))
            count += 1
        return res
```

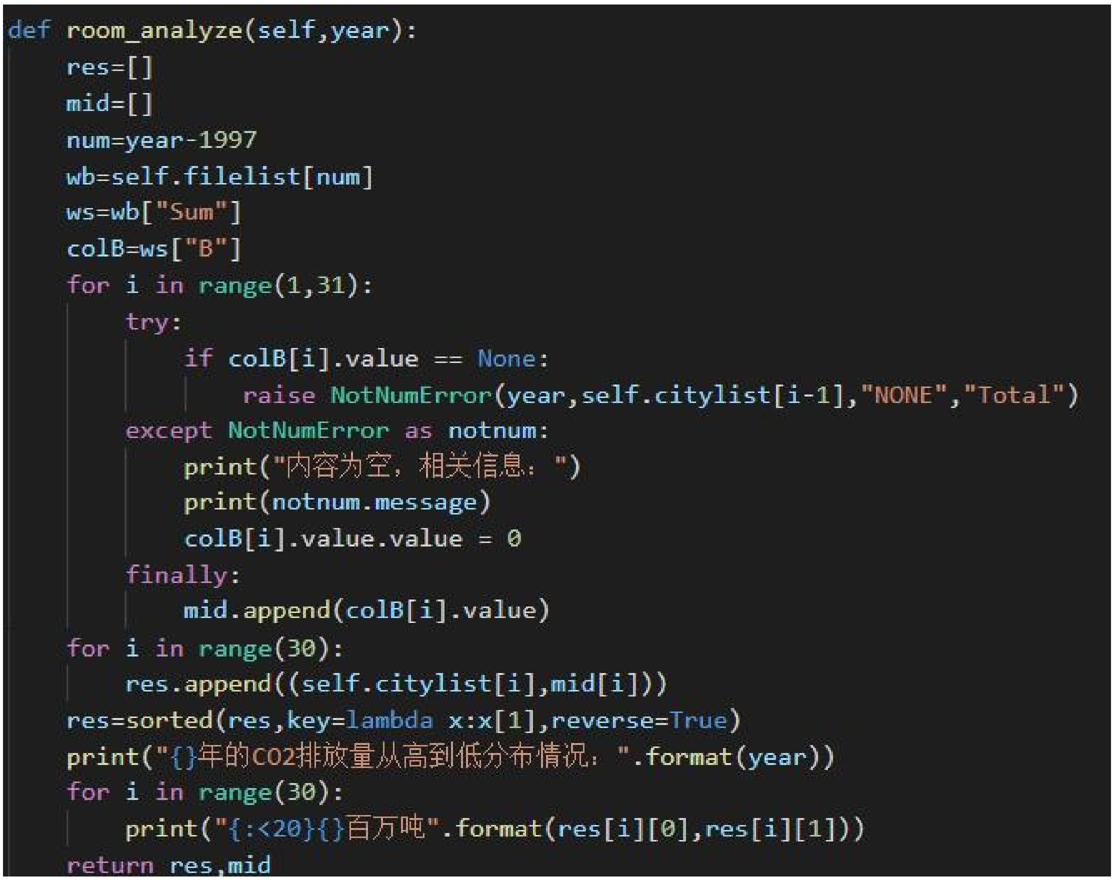
运行结果:



```
Beijing的Raw Coal类型排放量的年分布为:
1997年  28.6百万吨
1998年  30.2百万吨
1999年  33.9百万吨
2000年  34.9百万吨
2001年  34.3百万吨
2002年  33.9百万吨
2003年  35.6百万吨
2004年  39.9百万吨
2005年  43.1百万吨
2006年  44百万吨
2007年  44.8百万吨
2008年  41.3百万吨
2009年  40.1百万吨
2010年  39.6百万吨
2011年  38.7百万吨
2012年  37.2百万吨
2013年  33.1百万吨
2014年  28.3百万吨
2015年  18.6百万吨
```

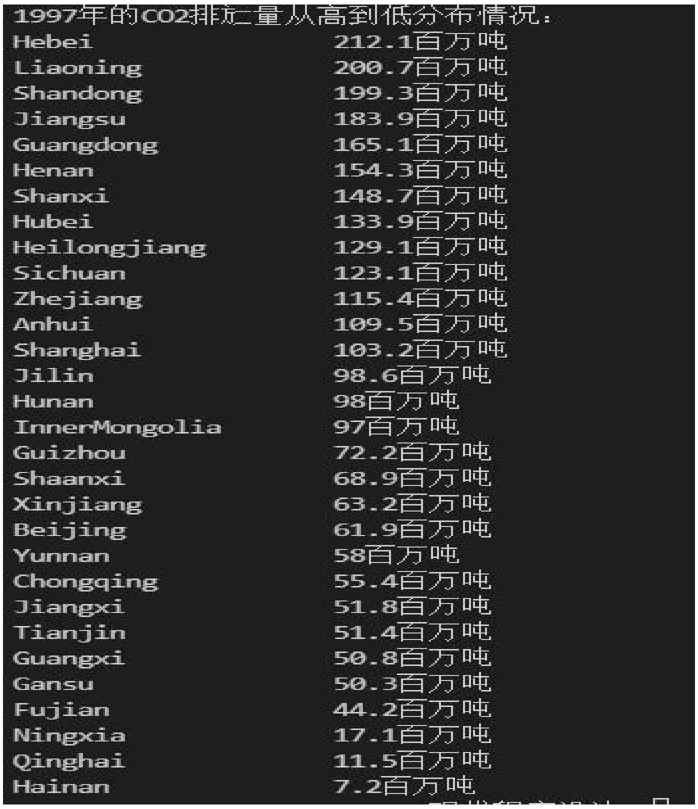
（3）　空间分布分析

根据年份读取对应所有城市的碳排放量，并按照从高到低的顺序输出结果；在读取数据前判断表格内容是否为空，如果是抛出异常并且填入“0”，异常处理后再读取结果。



```python
def room_analyze(self,year):
    res=[]
    mid=[]
    num=year-1997
    wb=self.filelist[num]
    ws=wb["Sum"]
    colB=ws["B"]
    for i in range(1,31):
        try:
            if colB[i].value == None:
                raise NotNumError(year,self.citylist[i-1],"NONE","Total")
        except NotNumError as notnum:
            print("内容为空，相关信息：")
            print(notnum.message)
            colB[i].value.value = 0
        finally:
            mid.append(colB[i].value)
    for i in range(30):
        res.append((self.citylist[i],mid[i]))
    res=sorted(res,key=lambda x:x[1],reverse=True)
    print("{}年的CO2排放量从高到低分布情况：".format(year))
    for i in range(30):
        print("{:<20}{}百万吨".format(res[i][0],res[i][1]))
    return res,mid
```

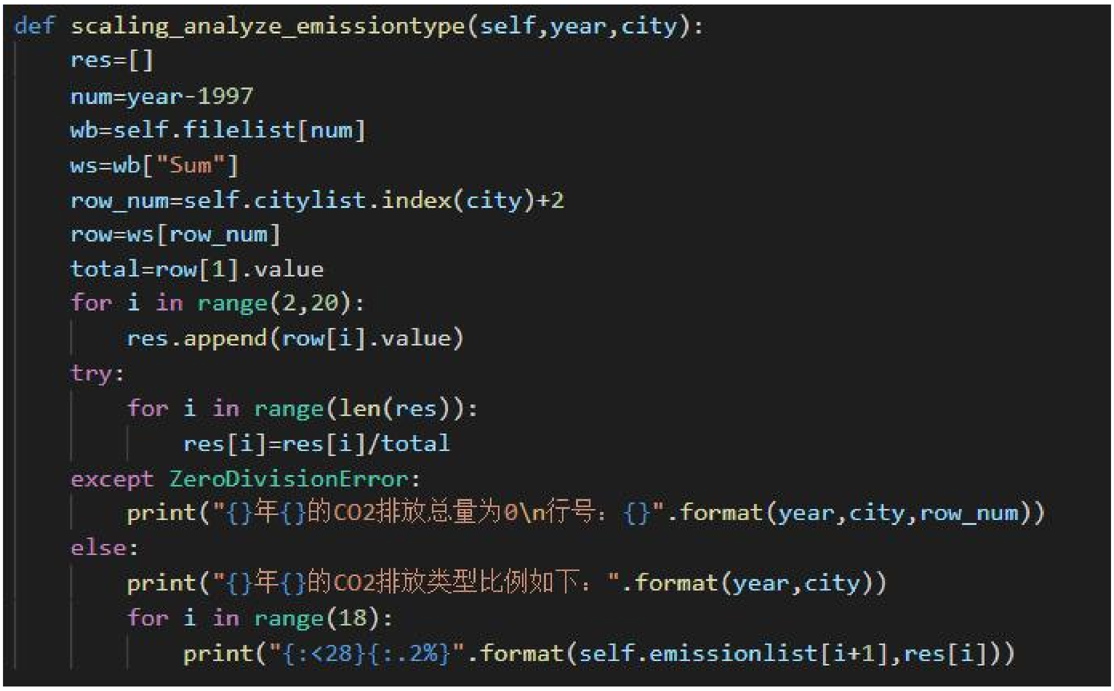
运行结果：



```
1997年的CO2排放量从高到低分布情况：
Hebei                   212.1百万吨
Liaoning                200.7百万吨
Shandong                199.3百万吨
Jiangsu                 183.9百万吨
Guangdong               165.1百万吨
Henan                   154.3百万吨
Shanxi                  148.7百万吨
Hubei                   133.9百万吨
Heilongjiang            129.1百万吨
Sichuan                 123.1百万吨
Zhejiang                115.4百万吨
Anhui                   109.5百万吨
Shanghai                103.2百万吨
Jilin                   98.6百万吨
Hunan                   98百万吨
InnerMongolia           97百万吨
Guizhou                 72.2百万吨
Shaanxi                 68.9百万吨
Xinjiang                63.2百万吨
Beijing                 61.9百万吨
Yunnan                  58百万吨
Chongqing               55.4百万吨
Jiangxi                 51.8百万吨
Tianjin                 51.4百万吨
Guangxi                 50.8百万吨
Gansu                   50.3百万吨
Fujian                  44.2百万吨
Ningxia                 17.1百万吨
Qinghai                 11.5百万吨
Hainan                  7.2百万吨
```

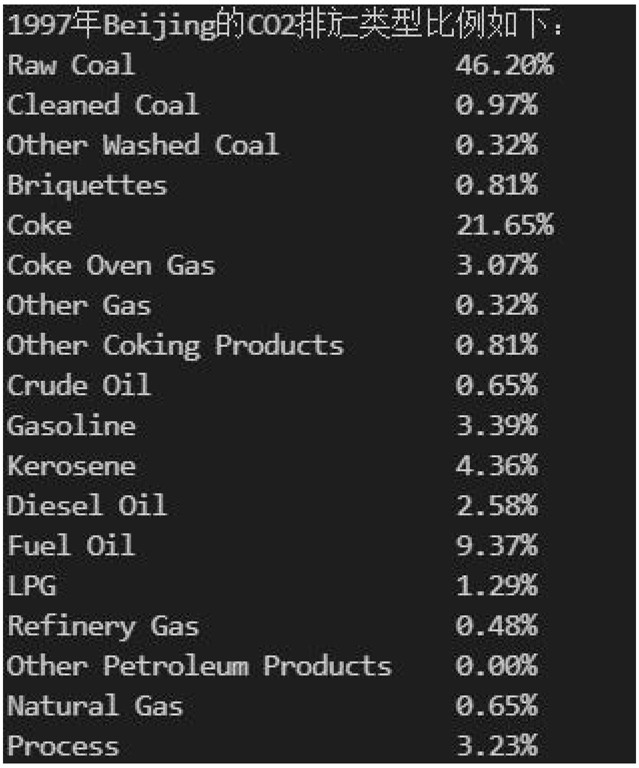
（4）　城市 C02 排放类型占比分析

根据输入的年份和城市输出当年该城市的 CO2 排放类型占比；如果城市的排放总量为 0 会抛出异常，包含年份，城市，行号异常信息



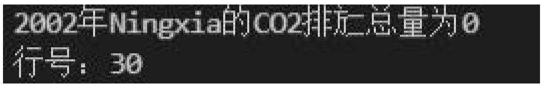
```
def scaling_analyze_emissiontype(self,year,city):
    res=[]
    num=year-1997
    wb=self.filelist[num]
    ws=wb["Sum"]
    row_num=self.citylist.index(city)+2
    row=ws[row_num]
    total=row[1].value
    for i in range(2,20):
        res.append(row[i].value)
    try:
        for i in range(len(res)):
            res[i]=res[i]/total
    except ZeroDivisionError:
        print("{}年{}的CO2排放总量为0\n行号：{}".format(year,city,row_num))
    else:
        print("{}年{}的CO2排放类型比例如下：".format(year,city))
        for i in range(18):
            print("{:<28}{:.2%}".format(self.emissionlist[i+1],res[i]))
```

运行结果:



```
1997年Beijing的CO2排放类型比例如下：
Raw Coal                    46.20%
Cleaned Coal                0.97%
Other Washed Coal           0.32%
Briquettes                  0.81%
Coke                        21.65%
Coke Oven Gas               3.07%
Other Gas                   0.32%
Other Coking Products       0.81%
Crude Oil                   0.65%
Gasoline                    3.39%
Kerosene                    4.36%
Diesel Oil                  2.58%
Fuel Oil                    9.37%
LPG                         1.29%
Refinery Gas                0.48%
Other Petroleum Products    0.00%
Natural Gas                 0.65%
Process                     3.23%
```

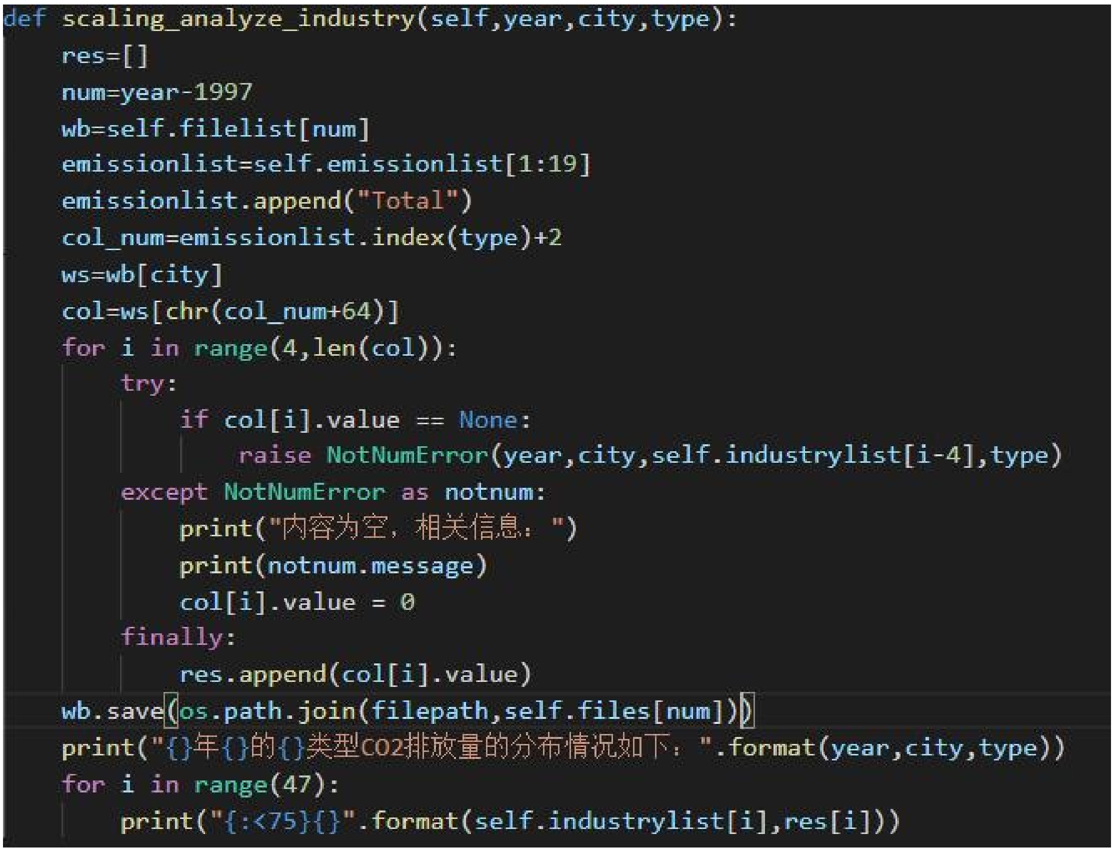
异常结果



```
2002年Ningxia的CO2排放总量为0
行号：30
```

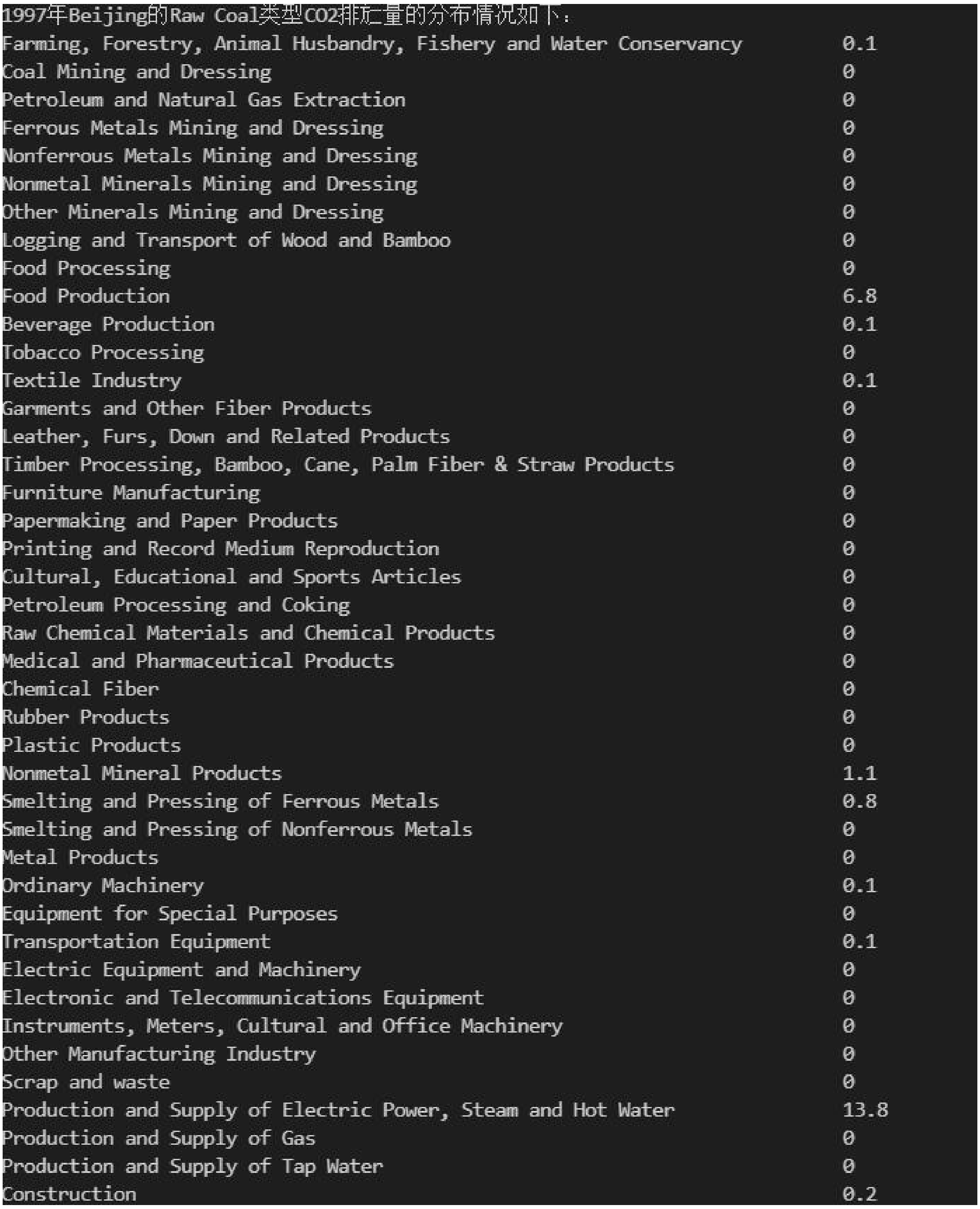
（5） 工业排放占比分析

根据输入的年份和城市，输出对应类型的 CO2 排放量中不同工业的占比；在读取数据时如果遇到空值会填充“0”，并且输出相关错误信息



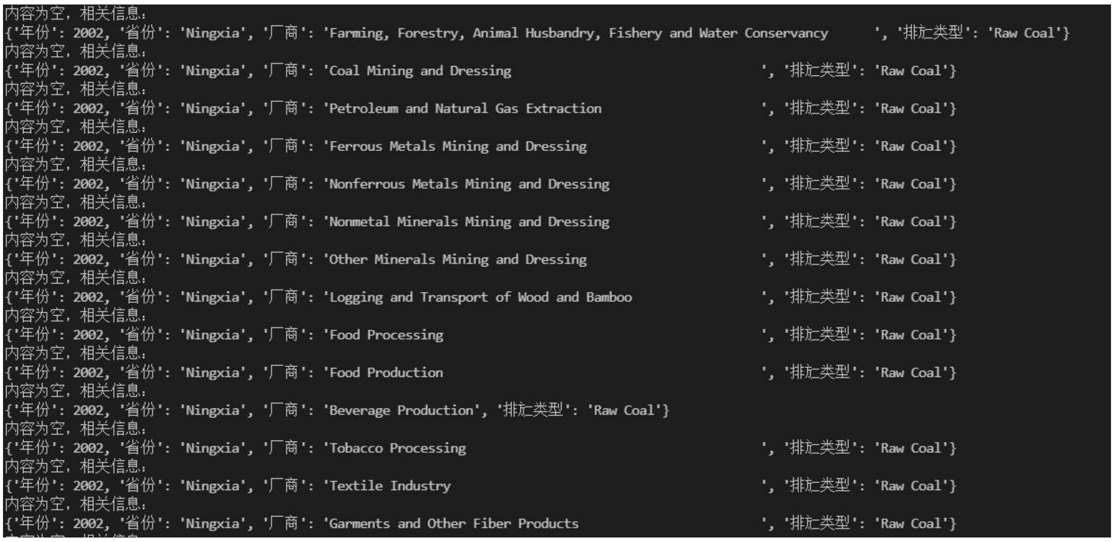
```python
def scaling_analyze_industry(self,year,city,type):
    res=[]
    num=year-1997
    wb=self.filelist[num]
    emissionlist=self.emissionlist[1:19]
    emissionlist.append("Total")
    col_num=emissionlist.index(type)+2
    ws=wb[city]
    col=ws[chr(col_num+64)]
    for i in range(4,len(col)):
        try:
            if col[i].value == None:
                raise NotNumError(year,city,self.industrylist[i-4],type)
        except NotNumError as notnum:
            print("内容为空，相关信息：")
            print(notnum.message)
            col[i].value = 0
        finally:
            res.append(col[i].value)
    wb.save(os.path.join(filepath,self.files[num]))
    print("{}年{}的{}类型CO2排放量的分布情况如下：".format(year,city,type))
    for i in range(47):
        print("{:<75}{}".format(self.industrylist[i],res[i]))
```

正常运行结果:

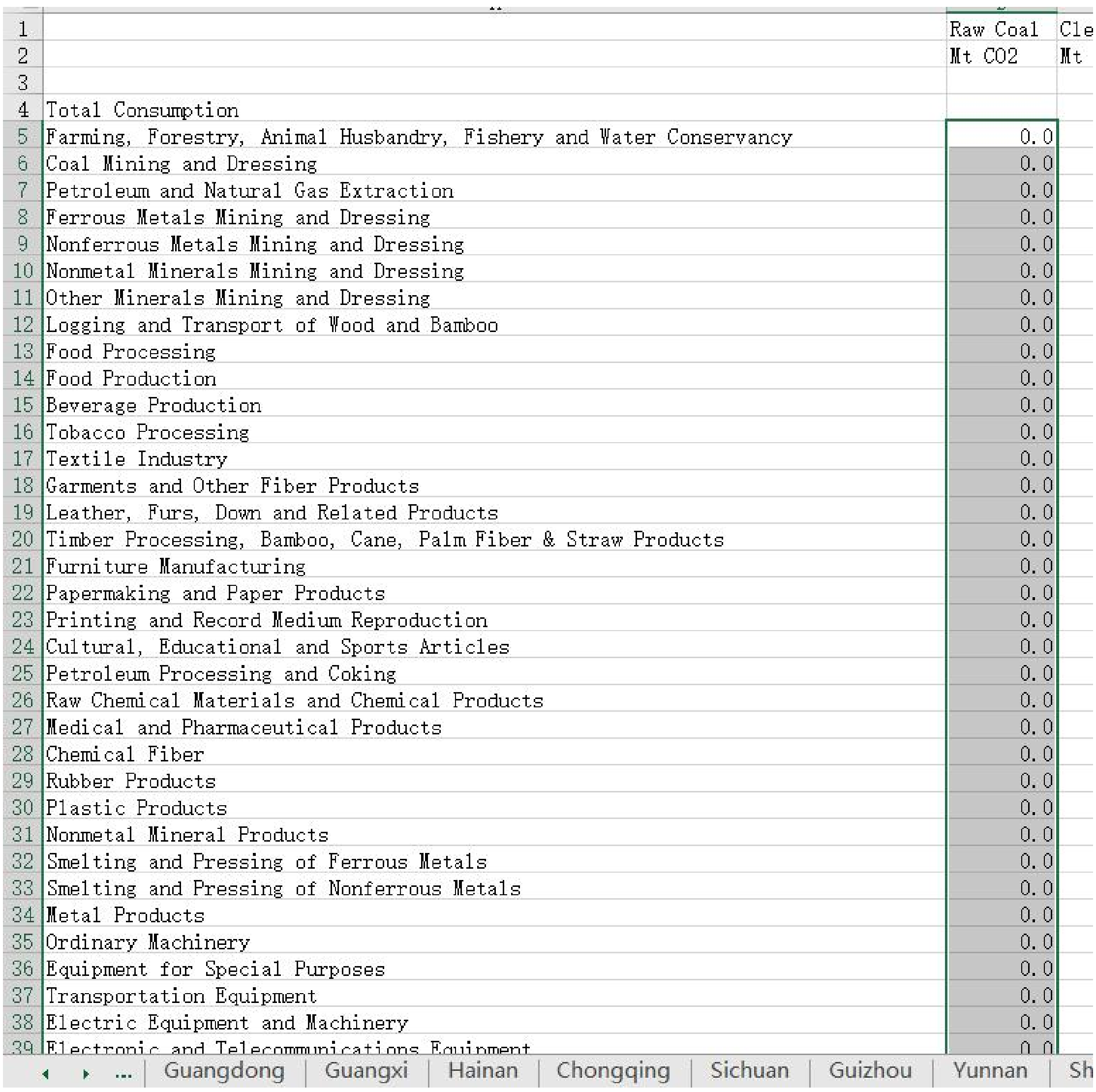


```
1997年Beijing的Raw Coal类型CO2排放量的分布情况如下：
Farming, Forestry, Animal Husbandry, Fishery and Water Conservancy         0.1
Coal Mining and Dressing                                                   0
Petroleum and Natural Gas Extraction                                       0
Ferrous Metals Mining and Dressing                                         0
Nonferrous Metals Mining and Dressing                                      0
Nonmetal Minerals Mining and Dressing                                      0
Other Minerals Mining and Dressing                                         0
Logging and Transport of Wood and Bamboo                                   0
Food Processing                                                            0
Food Production                                                            6.8
Beverage Production                                                        0.1
Tobacco Processing                                                         0
Textile Industry                                                           0.1
Garments and Other Fiber Products                                          0
Leather, Furs, Down and Related Products                                   0
Timber Processing, Bamboo, Cane, Palm Fiber & Straw Products               0
Furniture Manufacturing                                                    0
Papermaking and Paper Products                                             0
Printing and Record Medium Reproduction                                    0
Cultural, Educational and Sports Articles                                  0
Petroleum Processing and Coking                                            0
Raw Chemical Materials and Chemical Products                               0
Medical and Pharmaceutical Products                                        0
Chemical Fiber                                                             0
Rubber Products                                                            0
Plastic Products                                                           0
Nonmetal Mineral Products                                                  1.1
Smelting and Pressing of Ferrous Metals                                    0.8
Smelting and Pressing of Nonferrous Metals                                 0
Metal Products                                                             0
Ordinary Machinery                                                         0.1
Equipment for Special Purposes                                             0
Transportation Equipment                                                   0.1
Electric Equipment and Machinery                                           0
Electronic and Telecommunications Equipment                                0
Instruments, Meters, Cultural and Office Machinery                         0
Other Manufacturing Industry                                               0
Scrap and waste                                                            0
Production and Supply of Electric Power, Steam and Hot Water               13.8
Production and Supply of Gas                                               0
Production and Supply of Tap Water                                         0
Construction                                                               0.2
```

异常运行结果:



```
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Farming, Forestry, Animal Husbandry, Fishery and Water Conservancy     ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Coal Mining and Dressing                                  ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Petroleum and Natural Gas Extraction                      ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Ferrous Metals Mining and Dressing                        ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Nonferrous Metals Mining and Dressing                     ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Nonmetal Minerals Mining and Dressing                     ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Other Minerals Mining and Dressing                        ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Logging and Transport of Wood and Bamboo                  ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Food Processing                                           ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Food Production                                           ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Beverage Production', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Tobacco Processing                                        ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Textile Industry                                          ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
{'年份': 2002, '省份': 'Ningxia', '厂商': 'Garments and Other Fiber Products                         ', '排放类型': 'Raw Coal'}
内容为空，相关信息：
```

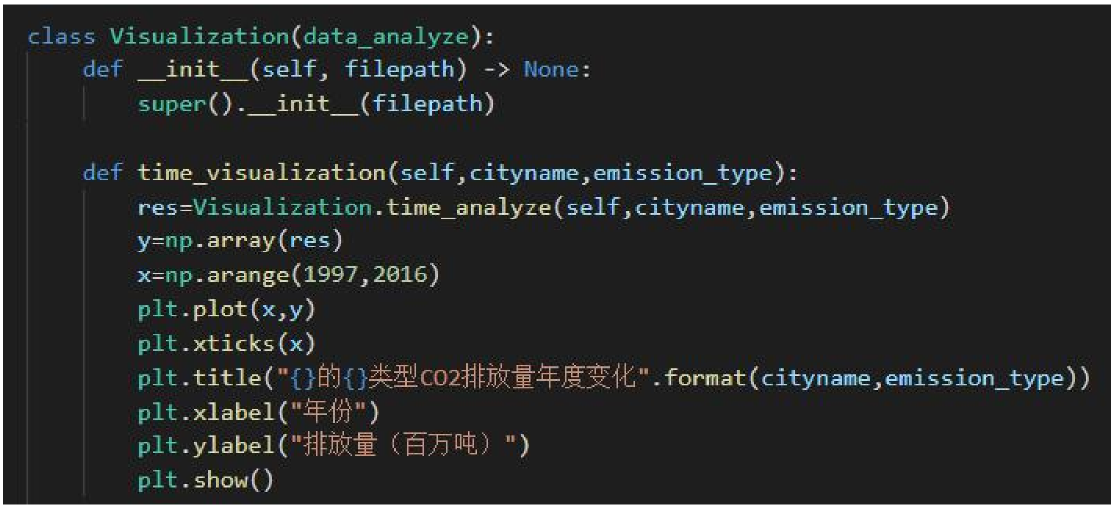


1		Raw Coal	Cle
2		Mt CO2	Mt
3			
4	Total Consumption		
5	Farming, Forestry, Animal Husbandry, Fishery and Water Conservancy	0.0	
6	Coal Mining and Dressing	0.0	
7	Petroleum and Natural Gas Extraction	0.0	
8	Ferrous Metals Mining and Dressing	0.0	
9	Nonferrous Metals Mining and Dressing	0.0	
10	Nonmetal Minerals Mining and Dressing	0.0	
11	Other Minerals Mining and Dressing	0.0	
12	Logging and Transport of Wood and Bamboo	0.0	
13	Food Processing	0.0	
14	Food Production	0.0	
15	Beverage Production	0.0	
16	Tobacco Processing	0.0	
17	Textile Industry	0.0	
18	Garments and Other Fiber Products	0.0	
19	Leather, Furs, Down and Related Products	0.0	
20	Timber Processing, Bamboo, Cane, Palm Fiber & Straw Products	0.0	
21	Furniture Manufacturing	0.0	
22	Papermaking and Paper Products	0.0	
23	Printing and Record Medium Reproduction	0.0	
24	Cultural, Educational and Sports Articles	0.0	
25	Petroleum Processing and Coking	0.0	
26	Raw Chemical Materials and Chemical Products	0.0	
27	Medical and Pharmaceutical Products	0.0	
28	Chemical Fiber	0.0	
29	Rubber Products	0.0	
30	Plastic Products	0.0	
31	Nonmetal Mineral Products	0.0	
32	Smelting and Pressing of Ferrous Metals	0.0	
33	Smelting and Pressing of Nonferrous Metals	0.0	
34	Metal Products	0.0	
35	Ordinary Machinery	0.0	
36	Equipment for Special Purposes	0.0	
37	Transportation Equipment	0.0	
38	Electric Equipment and Machinery	0.0	
39	Electronic and Telecommunications Equipment	0.0	

3. 数据可视化类

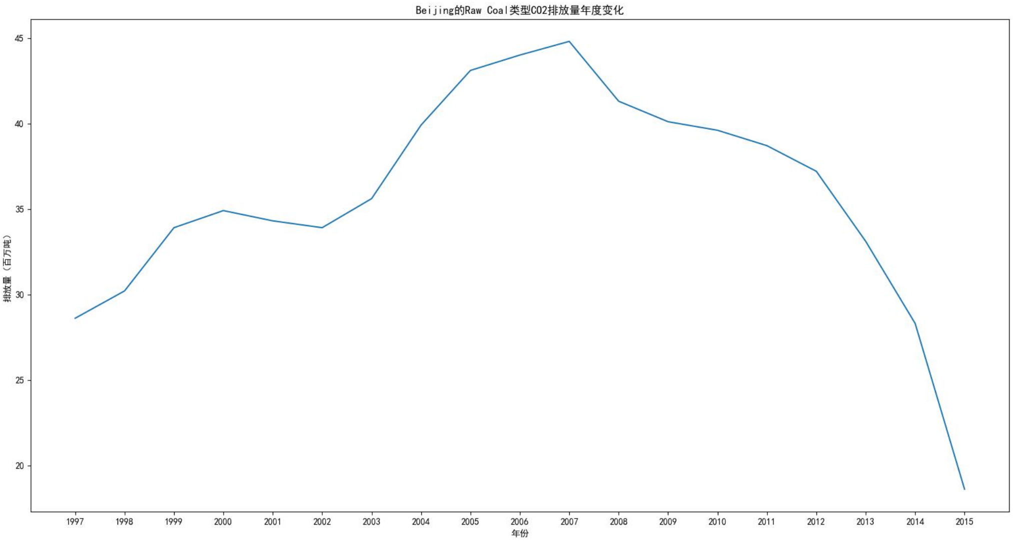
（1） 时间分布可视化

可视化类继承数据分析类，先进行数据分析，用列表储存结果，然后利用 plot 绘制出随时间该城市的排放总量的变化，设置坐标轴刻度按照年份显示



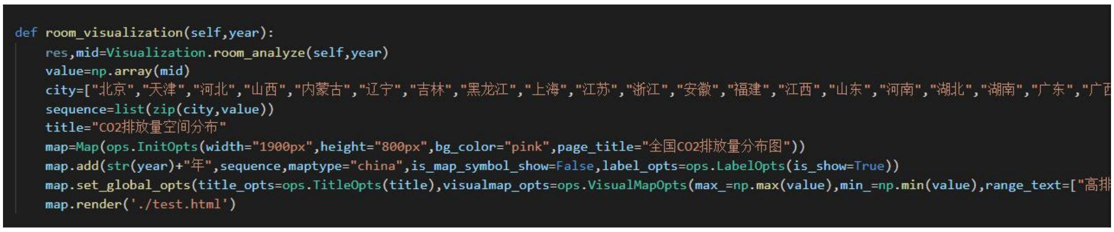
```
class Visualization(data_analyze):
    def __init__(self, filepath) -> None:
        super().__init__(filepath)

    def time_visualization(self,cityname,emission_type):
        res=Visualization.time_analyze(self,cityname,emission_type)
        y=np.array(res)
        x=np.arange(1997,2016)
        plt.plot(x,y)
        plt.xticks(x)
        plt.title("{}的{}类型CO2排放量年度变化".format(cityname,emission_type))
        plt.xlabel("年份")
        plt.ylabel("排放量（百万吨）")
        plt.show()
```

运行结果:



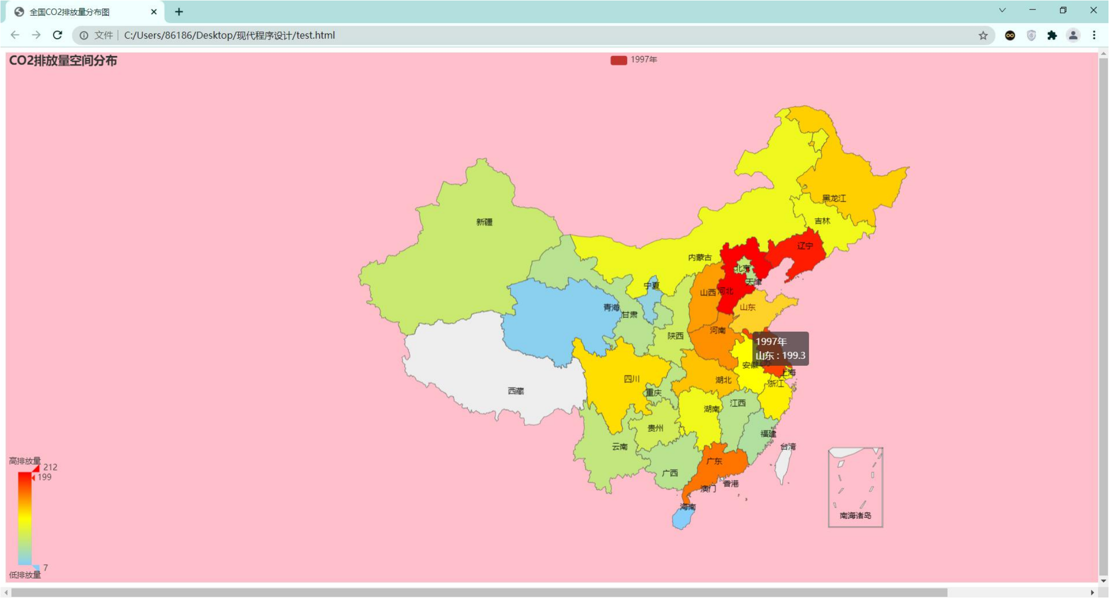
（2） 空间分布可视化

在地图上显示某一年所有省份的 CO2 排放量分布，用颜色区分排放量的相对大小



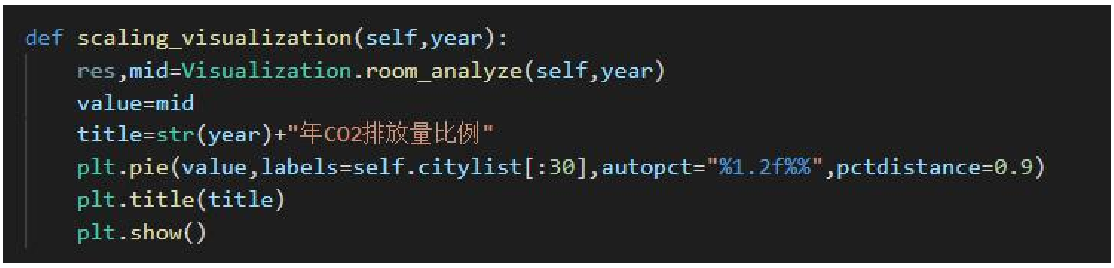
```
def room_visualization(self,year):
    res,mid=Visualization.room_analyze(self,year)
    value=np.array(mid)
    city=["北京","天津","河北","山西","内蒙古","辽宁","吉林","黑龙江","上海","江苏","浙江","安徽","福建","江西","山东","河南","湖北","湖南","广东","广西
    sequence=list(zip(city,value))
    title="CO2排放量空间分布"
    map=Map(ops.InitOpts(width="1900px",height="800px",bg_color="pink",page_title="全国CO2排放量分布图"))
    map.add(str(year)+"年",sequence,maptype="china",is_map_symbol_show=False,label_opts=ops.LabelOpts(is_show=True))
    map.set_global_opts(title_opts=ops.TitleOpts(title),visualmap_opts=ops.VisualMapOpts(max_=np.max(value),min_=np.min(value),range_text=["高排
    map.render('./test.html')
```

运行结果：



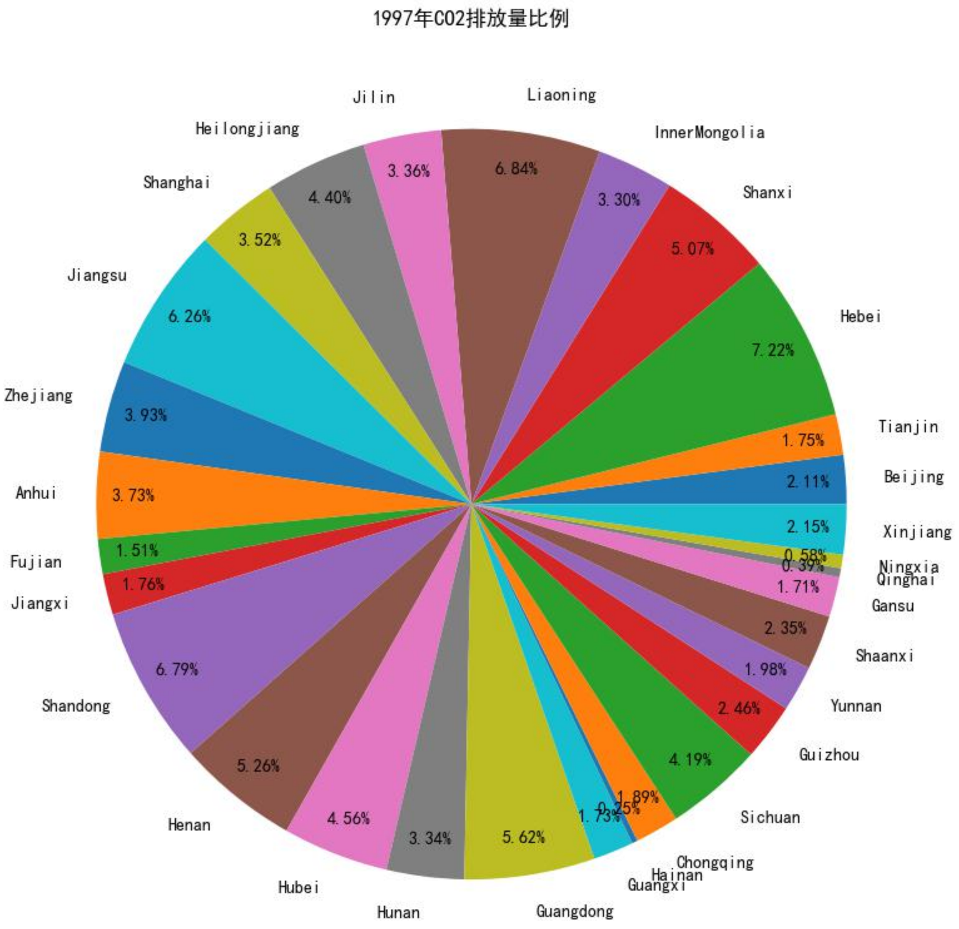
（3） CO2 排放城市占比可视化

展示某一年份不同城市 CO2 排放量占比的饼图

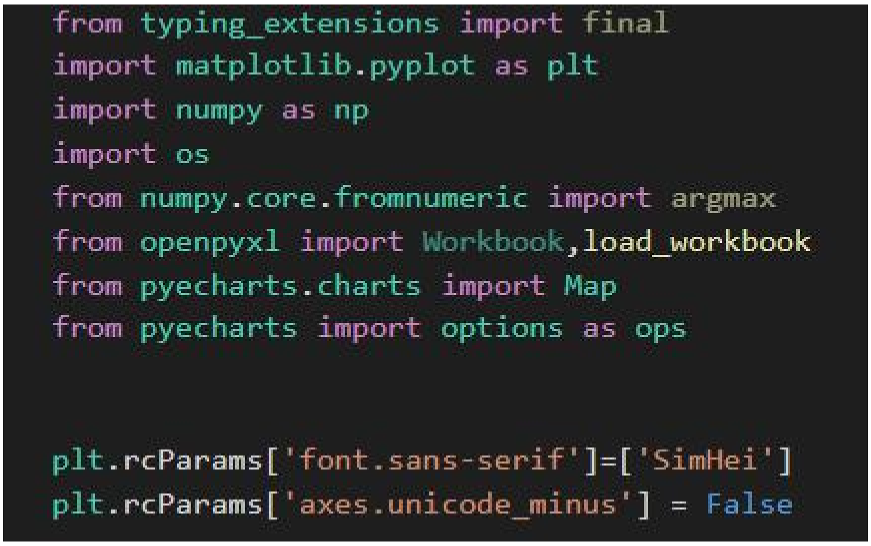


```
def scaling_visualization(self,year):
    res,mid=Visualization.room_analyze(self,year)
    value=mid
    title=str(year)+"年CO2排放量比例"
    plt.pie(value,labels=self.citylist[:30],autopct="%1.2f%%",pctdistance=0.9)
    plt.title(title)
    plt.show()
```

运行结果：



作业采用的库：



```
from typing_extensions import final
import matplotlib.pyplot as plt
import numpy as np
import os
from numpy.core.fromnumeric import argmax
from openpyxl import Workbook,load_workbook
from pyecharts.charts import Map
from pyecharts import options as ops


plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False
```