```
# -*- coding: utf-8 -*-
"""
Template for Datist Extensions
***请根据具体开发情况修改大写字母部分信息***
Created on Mon Feb 18 09:07:42 2019
@Summary: 代码的功能描述
@author: 代码编写人
@Copyright：版权声明，如不公开源码，请编译成pyc文件发布
@CI/CD: pass in Python3.6 with Datist2019Q1 请简要说明您的节点测试环境
@DevOps：https://github.com/gravity-igpcea/geoist_ext
"""

def run_PKGNAME(data):
    # part 1 ----------import and set loginfo---------- #
    import pathlib
    # output logging
    import geoist as gi
    gi.log.setname('Datist_PKGNAME')
    gi.log.info('pkgname START')  #如果使用Geoist包，可以根据从log信息记录节点运行时间等
    ####--------------End PART ONE--------------####
    # part 2 ----------读取前节点界面参数---------- #
    PARAMS1 = data['pars']['params']
    print('debug: Threshold//'+str(PARAMS1))    #Debug开头为输出到Datist调试信息
    data_path = data['OutputPath']
    orig_file = pathlib.Path(data_path,'YOURDATA.TXT')
    ####--------------End PART TWO--------------####
    # part 3 ----------设置程序运行参数---------- #
    param_xxx = PARAMS1
    #Res_flag = data['GetResult'] #Print,JSON #Datist用户界面设置结果收集方式
    Mod_flag = data['DataMode']  #FileList,Url,DataTable #用户界面设置显示与输出
    # parameters for saving data
    tmppath = pathlib.Path(__file__).parent
    res_file = pathlib.Path(tmppath,'tmpData','gradata.csv')
    png_file = pathlib.Path(tmppath,'tmpData','gradata.png')
    html_file = pathlib.Path(tmppath,'tmpData','gradata.html')
    ####--------------End PART THREE--------------####
    # part 4 ----------专业逻辑实现部分---------- #
    # data processing
    import pandas as pd
    # 请在此编写您的专业逻辑部分代码，建议采用Pandas的Dataframe数据格式进行处理
    d=pd.read_csv(pathlib.Path(orig_file),delimiter=';')
    d.to_csv(pathlib.Path(res_file),sep=';')
    gi.log.info('thresh_hold = ' + str(param_xxx))
    # output html
    if Mod_flag == 'Url':
        import bokeh.plotting as bp
        # 请在此编写您要输出的结果验证可视化结果
        print(str(html_file)) #输出网络地址
        gi.log.info('output html finished')
     #output static pic
    if Mod_flag == 'FileList':
        import matplotlib.pyplot as plt
        # 请在此编写您要输出的静态成果图片
        plt.savefig(str(png_file),format='png')
```

```
        gi.log.info('output png file finished')
        print(png_file) #输出一个图片
    ####--------------End PART FOUR--------------####
    # part 5 ------------输出数据----------------- #
    print(res_file) #输出数据表格文件,向后流转
    ####--------------End PART FIVE--------------####

if __name__ == '__main__':
    """
    通过__main__和def run_的逻辑分离，可以使自定义模块支持import形式的加载与管理
    Datist自定义节点的JSON接口格式示例,argfile为传入的JSON文件名
    {
      "pars": {
        "Threshold": "100"
      },
      "allfields": true,
      "names": {
        "Threshold": "100"
      },
      "gradata": "C:\\Users\\chens\\AppData\\Local\\Temp\\gradata.txt",
      "OutputPath": "C:\\Users\\chens\\AppData\\Local\\Temp\\",
      "ResultFile": "C:\\Users\\chens\\AppData\\Local\\Temp\\result.json"
    }
    """
    print('debug: starting//PKGNAME.py by author')
    import sys, json
    argfile=sys.argv[1] #json参数
    print('debug: open//'+argfile)
    with open(argfile,'rb') as f:
        data = json.load(f)
    run_PKGNAME(data)  #业务实现
    print('debug: finished//PKGNAME is finished and output now...')
```