

팀장: 민전기
팀원: 최윤서, 송인영, 이차영

파사모팀

프로젝트 개요

최윤서

폐기물발생량-인구수

인구수와 폐기물 발생량이 비례할 것이다.

시/도시별 인구수

https://kosis.kr/statHtml/statHtml.do?orgId=202&tblId=DT_202N_B4&conn_path=I2

생활 폐기물 발생량

<https://www.bigdata-map.kr/img/story/env/household-waste1.png>

송인영

폐기물발생량-코로나방역률

코로나 방역률에 따라 각 나라의 생활폐기물 발생 조사

- 코로나 방역을 잘한 나라일수록 생활폐기물 발생이 많을 것이다.

우리나라에서 각 지역의 코로나 확진자 증가 추이와 각 지역 생활폐기물 증가 추이는 비례할 것이다.

코로나 실시간 상황판

http://ncov.mohw.go.kr/bdBoardList_Real.do?brdId=1&brdGubun=13&ncvContSeq=&contSeq=&board_id=&gubun=

<https://m.yna.co.kr/view/AKR20220228166700530>

민전기

폐기물발생량-전력사용량

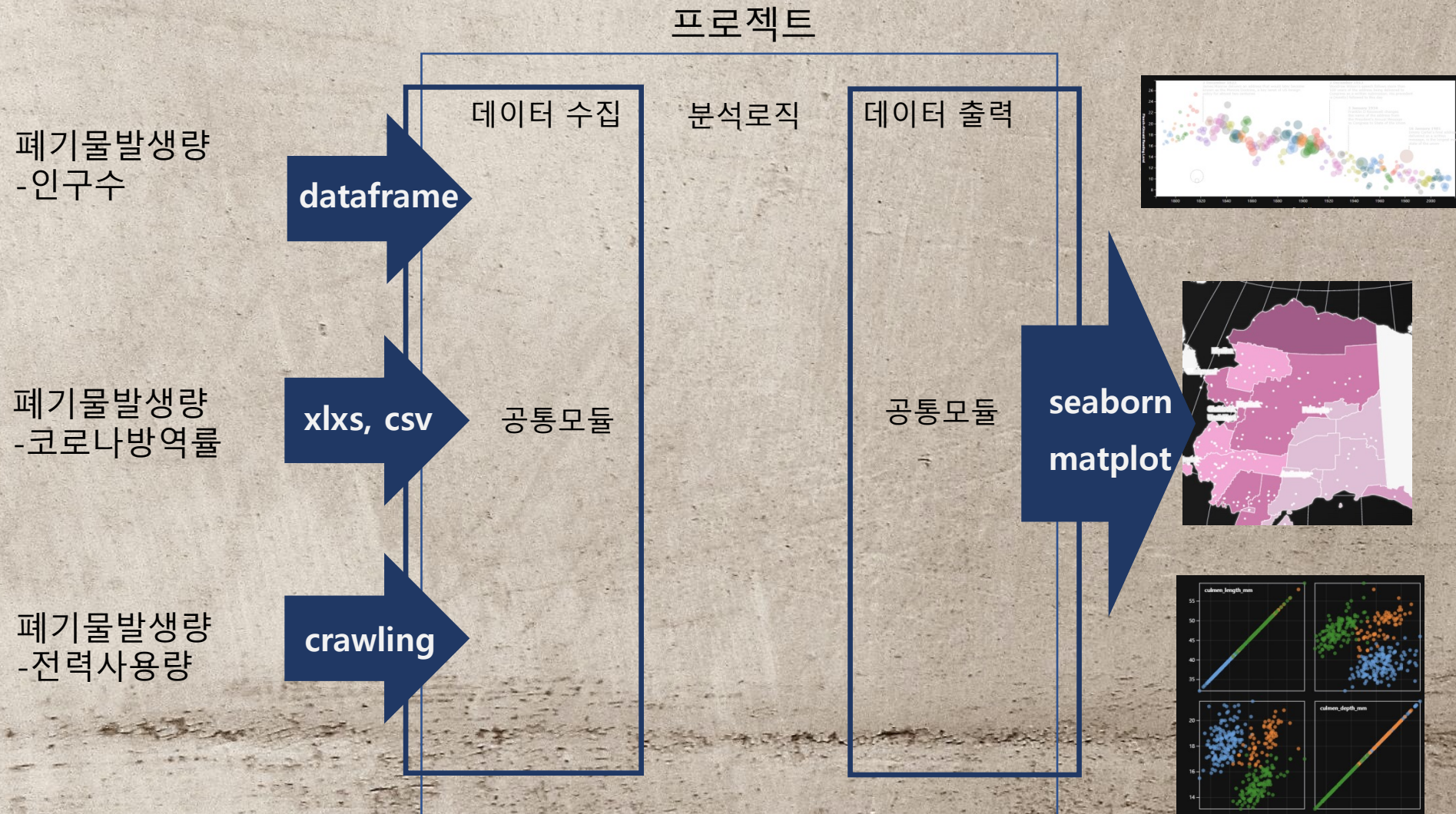
지역별 주택용 전력사용량이 많은곳일수록 생활폐기물양도 많을것이다.

전국폐기물 발생 및 처리현황(2021년)

https://www.recycling-info.or.kr/rrs/stat/envStatDetail.do?menuNo=M13020201&pageIndex=1&bbsId=BBSMSTR_000000000002&s_nttSj=KEC005

[&nttId=1200&searchBgnDe=&searchEndDe=](#)

프로세스



1. 모두 의견모아서 데이터 수집 공통모듈 생성
2. 모두 의견모아서 데이터 출력 공통모듈 생성
3. 각자 분석 로직 생성

가설: 폐기물발생량-인구수
인구수와 폐기물 발생량이 비례할 것이다.

- **사용된 자료**

시/도시별 인구수

https://kosis.kr/statHtml/statHtml.do?orgId=202&tblId=DT_202N_B4&conn_path=l2

생활폐기물 발생량

<https://www.bigdata-map.kr/img/story/env/household-waste1.png>

데이터 수집

시/도에 대한 인구

https://kosis.kr/statHtml/statHtml.do?orgId=202&tblId=DT_202N_B4&conn_path=I2

```
In [2]: import numpy
        from pandas import ExcelFile # Excel 파일을 읽어들이기 위한 클래스
```

```
In [3]: # 데이터 준비

xls = ExcelFile('stat_100701.xls')

df = xls.parse(xls.sheet_names[0])

df
```

Out [3]:

	시/도	인구
0	경기	13,238
1	서울	9,662
2	부산	3,373
3	경남	3,350
4	인천	2,944
5	경북	2,665
6	대구	2,432
7	충남	2,188
8	전북	1,803
9	전남	1,773
10	충북	1,626
11	강원	1,517
12	대전	1,509
13	광주	1,494
14	울산	1,147
15	제주	660
16	세종	331

데이터 수집

시/도에 대한 생활폐기물 발생량

<https://www.bigdata-map.kr/img/story/env/household-waste1.png>

```
In [4]: data2={
        '시/도' : ['경기', '서울', '부산', '경남', '인천', '경북', '대구', '충남'],
        '생활폐기물 총 발생량(톤/일)' : [9544, 8658, 2951, 3553, 1914, 2367, 2568, 2288]
    }
    data2
```

```
In [5]: import pandas as pd
        df2=pd.DataFrame(data2)
```

```
In [6]: df2
```

Out [6]:

	시/도	생활폐기물 총 발생량(톤/일)
0	경기	9544
1	서울	8658
2	부산	2951
3	경남	3553
4	인천	1914
5	경북	2367
6	대구	2568
7	충남	2288
8	전북	1578
9	전남	1720
10	충북	1906
11	강원	1866
12	대전	1287
13	광주	1160
14	울산	1267
15	제주	959
16	세종	325

데이터 전처리

```
In [7]: # 프레임의 정렬, 내림차순  
df2.sort_values('생활폐기물 총 발생량(톤/일)', ascending=False)
```

```
Out [7]:
```

	시/도	생활폐기물 총 발생량(톤/일)
--	-----	------------------

0	경기	9544
1	서울	8658
3	경남	3553
2	부산	2951
6	대구	2568
5	경북	2367
7	충남	2288
4	인천	1914
10	충북	1906
11	강원	1866
9	전남	1720
8	전북	1578
12	대전	1287
14	울산	1267
13	광주	1160
15	제주	959
16	세종	325

데이터 전처리

```
In [35]: data3={
    '시/도' : ['경기', '서울', '부산', '경남', '인천', '경북', '대구', '충남', '충북', '전남', '전북', '대전', '광주', '울산', '제주', '세종'],
    '인구' : [13238, 9662, 3373, 3350, 2944, 2665, 2432, 2188, 1803, 1773, 1626, 1517, 1509, 1494, 1147, 660, 331],
    '생활폐기물 총 발생량(톤/일)' : [9544, 8658, 2951, 3553, 1914, 2367, 2568, 2288, 1578, 1720, 1906, 1866, 1287, 1160, 1267, 959, 325]
}

import pandas as pd
df3=pd.DataFrame(data3)
df3
```

Out [35]:

	시/도	인구	생활폐기물 총 발생량(톤/일)
0	경기	13238	9544
1	서울	9662	8658
2	부산	3373	2951
3	경남	3350	3553
4	인천	2944	1914
5	경북	2665	2367
6	대구	2432	2568
7	충남	2188	2288
8	전북	1803	1578
9	전남	1773	1720
10	충북	1626	1906
11	강원	1517	1866
12	대전	1509	1287
13	광주	1494	1160
14	울산	1147	1267
15	제주	660	959
16	세종	331	325

그래프 작성_한글 적용

```
In [52]: import matplotlib
from matplotlib import font_manager, rc
import matplotlib.pyplot as plt
import platform

# 한글 및 마이너스 표시 설정
path = "E:/나눔 글꼴/나눔고딕/NanumFontSetup_TTF_GOTHIC/NanumGothic.ttf"
# path = "C:/Windows/Fonts/malgun.ttf"
if platform.system() == "Windows":
    font_name = font_manager.FontProperties(fname=path).get_name()
    matplotlib.rc('font', family=font_name)
elif platform.system()=="Darwin":
    rc('font', family='AppleGothic')
else:
    print("Unknown System")

matplotlib.rcParams['axes.unicode_minus'] = False
%matplotlib inline
```


그래프 작성_시각화

```
In [55]: import matplotlib.pyplot as plt
import numpy as np

# plt.style.use('default')
plt.rcParams['figure.figsize'] = (12, 6)
plt.rcParams['font.size'] = 14

x = df3.loc[:, '시/도'] #x축을 시/도로 지정
y1 = df3.loc[:, '인구'] #y축에 인구, 생활폐기물 총 발생량(톤/일)이라는 변수를 각각 지정
y2 = df3.loc[:, '생활폐기물 총 발생량(톤/일)']
```

```
fig, ax1 = plt.subplots()
ax1.set_xlabel('시/도')
ax1.set_ylabel('인구')
line1 = ax1.plot(x, y1, color='green')

ax2 = ax1.twinx()
ax2.set_ylabel('생활폐기물 총 발생량(톤/일)')
line2 = ax2.plot(x, y2, color='magenta')

lines = line1 + line2
labels = [l.get_label() for l in lines]
ax1.legend(lines, labels, loc='upper right')
plt.show()
```



결론: 인구수와 폐기물 발생량이

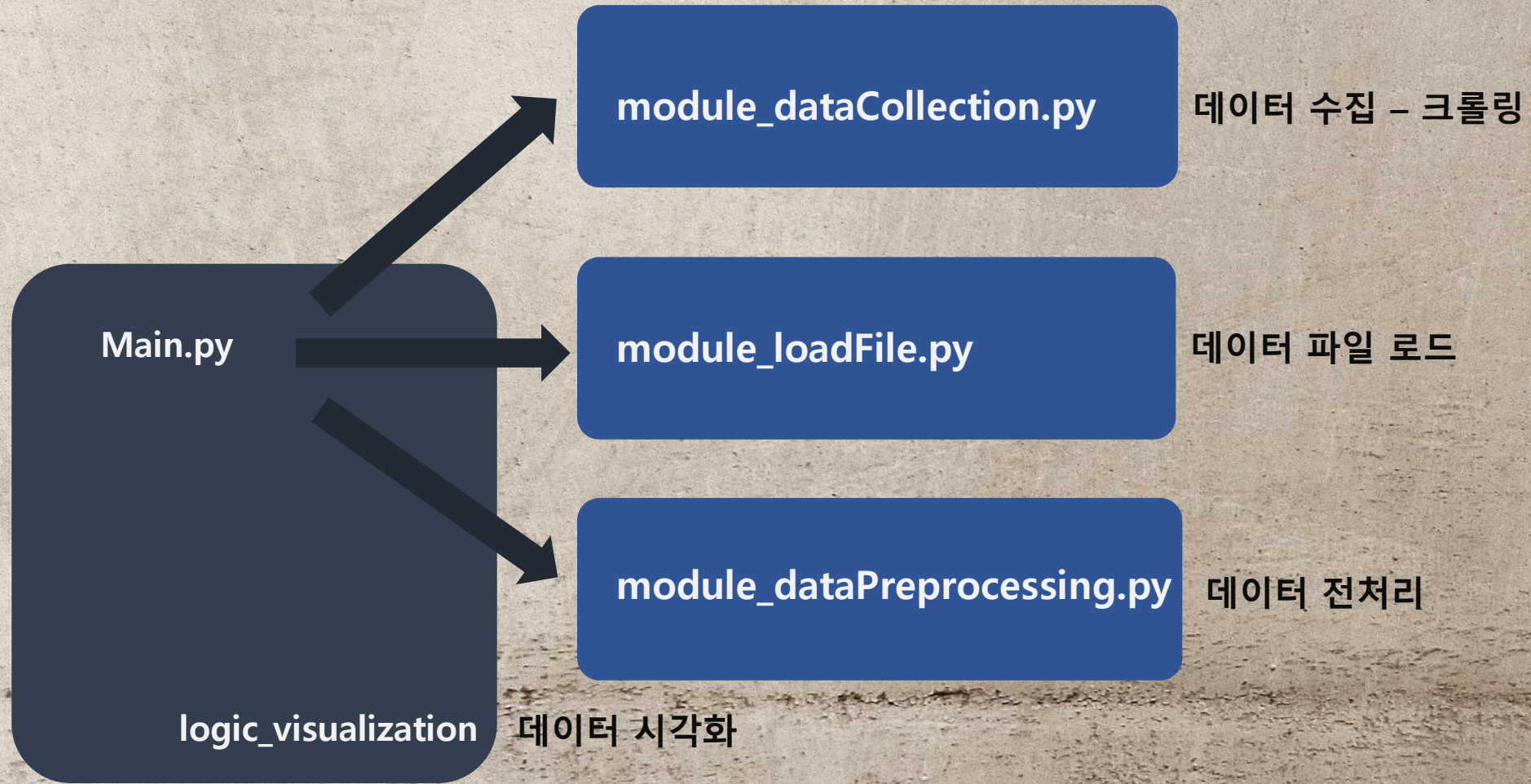
가설: 폐기물발생량-전력사용량
지역별 주택용 전력사용량이 많은곳일수록 생활폐기물양도 많을것이다.

• 사용된 자료

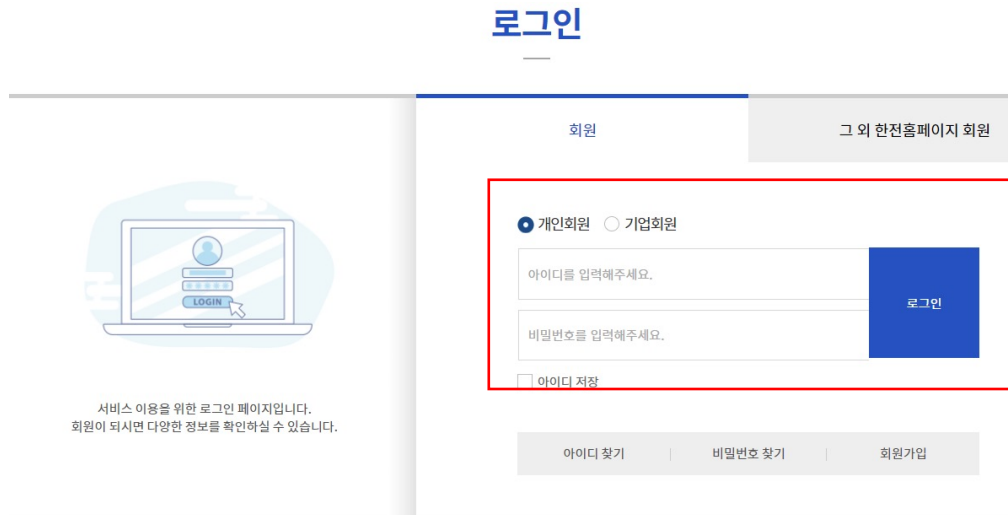
전국폐기물 발생 및 처리현황(2020년)

[https://www.recycling-info.or.kr/rrs/stat/envStatDetail.do?menuNo=M13020201&pageIndex=1&bbsId=BBSMSTR_0000000000002&s_nttSj=KEC005
&nttId=1200&searchBgnDe=&searchEndDe=](https://www.recycling-info.or.kr/rrs/stat/envStatDetail.do?menuNo=M13020201&pageIndex=1&bbsId=BBSMSTR_0000000000002&s_nttSj=KEC005&nttId=1200&searchBgnDe=&searchEndDe=)

파일구조



데이터 수집_로그인



로그인 화면

```
# 웹화면 이동 순서 : 로그인 -> 데이터수집화면(검색, 데이터다운)
def getElectData():
    # 사용자정보
    id = "jackwam"
    pwd = "1"

    # 화면-로그인
    dvr.get("https://en-ter.co.kr/ft/login/login.do") # 로그인
    time.sleep(2)
    dvr.find_element_by_id("id").send_keys(id)
    dvr.find_element_by_id("pwd").send_keys(pwd)
    dvr.find_element_by_id("btn_login").click() # action-login 버튼 클릭
```

m_dataCollection.py

데이터 수집_데이터

전력데이터 개방포털시스템

환전가산 방안을 환영합니다

사이트맵

데이터 공개

맞춤 데이터 제공

공유 센터

데이터 공유

소통과 참여

센터소개


데이터공개

주요전력통계

계약종별 전력사용량

계약종별 전력사용량

전력데이터 개방 계약종별 전력사용량입니다.



선택

☒ 전국 ☐ 행정구역별

전체(시도)

전체(시/군/구)

기간

2021

01

 ~

2021

06

추출정보

☒ 전체선택

☒ 주택용 ☐ 일반용

☐ 교육용 ☐ 산업용

☐ 농사용 ☐ 가로등

☐ 심야

검색

계약종별 읍/면/동 자료 확인

전체

테이블

그래프

추이그래프

월별 데이터 다운로드

다운로드

```
# 화면-데이터위치
dvr.get("https://bigdata.kepco.co.kr/cmsmain.do?scode=S01&pcode=000166")
## 검색 셀렉트-날짜 제어
from selenium.webdriver.support.select import Select # select box 이용시 필요
Select(dvr.find_element_by_id("sdateYear")).select_by_value("2021")
Select(dvr.find_element_by_id("sdateMonth")).select_by_value("01")
Select(dvr.find_element_by_id("edateYear")).select_by_value("2021")
Select(dvr.find_element_by_id("edateMonth")).select_by_value("06")

## 검색 체크박스-카테고리
chk_eles = dvr.find_elements_by_css_selector(".multi_chk input[type=checkbox]") # 7개 체크박스 선택자
for idx, ele in enumerate(chk_eles): # 7개 중 주택용만 제외하고 전체 체크해지
    check_txt = dvr.find_elements_by_css_selector(".multi_chk label")[idx].text
    if "주택용" == check_txt:
        continue
    dvr.execute_script('arguments[0].removeAttribute("checked")', ele) # 체크 해지

dvr.find_element_by_id("btn_search").click() # action-검색 버튼 클릭
time.sleep(2)
dvr.find_element_by_id("btn_mondownex").click() # action-월별파일 엑셀다운로드 버튼 클릭
```

데이터 조회 및 다운로드 화면

m_dataCollection.py

데이터 다운로드

```
# 파일 불러오기
## 최신 파일 찾기
import os
import pandas as pd
f_path = "C:\\Users\\jack\\Downloads\\"

def getFileData():
    file = []
    for f_name in os.listdir(f"{f_path}"):
        cDate = os.path.getctime(f"{f_path}{f_name}")

        # 최신 날짜 파일 찾기 로직
        if len(file) != 0:
            if file[0]['cDate'] > cDate:
                continue
            file[0] = {'f_name':f_name, 'cDate':cDate}
        else:
            file.append({'f_name':f_name, 'cDate':cDate})

    ## 파일 읽기
    return pd.read_excel(f_path + file[0]['f_name'], header=14, usecols=[0, 1, 3, 5])
```

m_loadFile.py

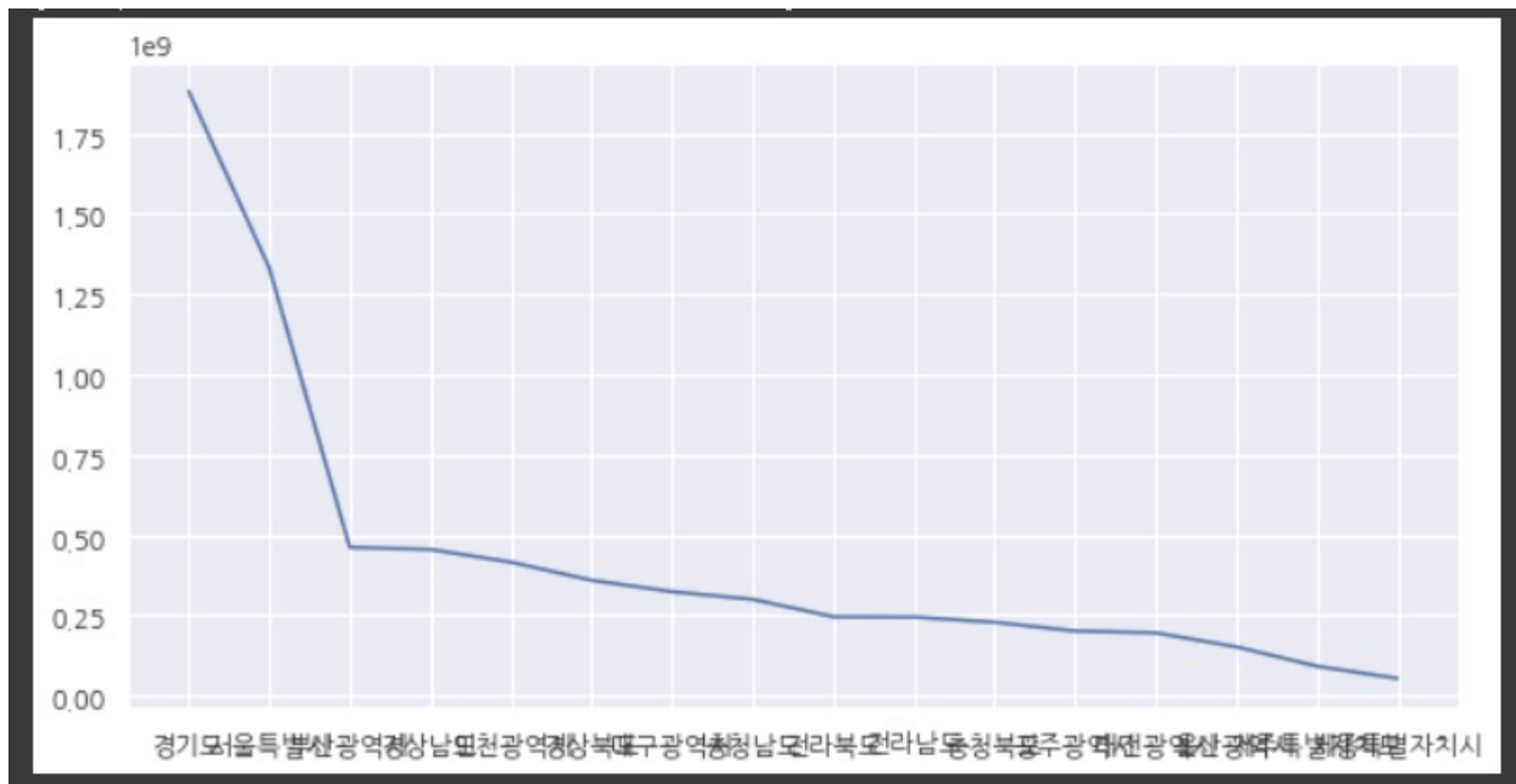
데이터 전처리

```
## 파일 읽기
elect.columns=['년월','지역','구분','전력량']

# 3. 데이터 전처리
## 조건에 맞는 데이터 추출
elect = elect[(elect['구분']=='주택용') & (elect['지역']!='황해북도')] # 주택용만
## 타입 변환
elect['전력량']=elect['전력량'].str.replace(pat=r'^\w+', repl='', regex=True) #특수문자 제거
elect = elect.astype({'전력량':'int'})
## 정렬
elect.sort_values(by=['년월','전력량'], ascending=[True,False], inplace=True) # 정렬
elect = elect[elect['년월']==202101]
```

main.py

데이터 시각화



main.py

**Tha
nk
you**

