데이터 수집
계획 및 보고서

[데이터 수집] 프로젝트 계획

프로젝트명	지역별 인구 이동 분석
능력단위분류	데이터 전처리 및 분석
수행 기간	2023.05.18. - 2023.05.26

■ 상세 정보

	설명 및 의견
데이터 수집 목적	및 연령별 인구 이동량을 파악하여 지역 간의 국토개발, 교통, 교육 및 주택 등의 각종 정책수립을 위한 자료로 활용할 수 있음
데이터 수집 계획	공공데이터포털에서 행정안전부_지역별 인구이동 현황 OpenAPI 수집 https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15108093
데이터 수집 시스템 환경	Windows 11, Python 3.10.9, conda 23.1.0, pandas 2.0.1, matplotlib 3.7.1, seaborn 0.12.2
데이터 수집 후 저장 방법	Open API로 데이터를 수집한 후 data.json로 데이터 저장 데이터 정제 후 population_data.csv로 파일 저장
데이터 수집 후 정제 계획	불필요한 컬럼 삭제, 컬럼 타입 변경(NmprCnt가 포함된 컬럼 타입을 int로 변경), 컬럼명 변경(영어를 한국말로), 컬럼 순서 변경, 남성과 여성을 연령대로 컬럼 수정(10~19세는 10대와 같은 방식)

[데이터 수집] 프로젝트 보고서

■ 데이터 수집 작업

	설명
코딩	```import requests import pandas as pd import matplotlib.pyplot as plt import seaborn as sns import numpy as np import math import json from pandas import DataFrame```
	- open API 통해 2023년 1월부터 2023년 4월까지의 모든 데이터 추출
	``headers = { 'accept': 'application/json', } # 모든 페이지의 데이터 추출 all_response = [] for y in range(202301, 202305): for i in range(math.ceil(289/100)): params = { 'serviceKey': 'l4LZrZUxATppYDmrEIW59Ezozth1YopBM2a7lsi2lpd/v+/0t6WqiLJXQ2Am/Iadh9L0y/6Ach7Xy4km6WF8LA==', 'mvinAdmmCd': '1000000000', 'mvtAdmmCd': '1000000000', 'srchFrYm': y, 'srchToYm': y, 'lv': '1', 'type': 'JSON', 'numOfRows': '100', 'pageNo': i+1 } response = requests.get('https://apis.data.go.kr/1741000/ppltnDataStus/selectPpltnDataStus', params=params, headers=headers) result = response.json() all_response.append(result)```
	- data.json 파일로 저장
	``with open('data.json', 'w') as f: json.dump(all_response, f)```
	- data.json 파일 불러오기
	``with open('data.json', 'r') as file: raw_data = json.load(file) raw_data```
	``[{'Response': {'head': {'pageNo': '1', 'resultCode': '0', 'totalCount': '289', 'numOfRows': '100', 'resultMsg': 'NORMAL_SERVICE'}, 'items': {'item': [{'male4AgeNmprCnt': '755', 'male76AgeNmprCnt': '328', 'male31AgeNmprCnt': '4320', 'feml55AgeNmprCnt': '1244', 'mvtCtpvNm': '서울특별시', 'feml47AgeNmprCnt': '1636', 'male84AgeNmprCnt': '134', 'male68AgeNmprCnt': '647', 'male101AgeNmprCnt': '1', 'feml63AgeNmprCnt': '1084', 'mvinSggNm': '', 'feml10AgeNmprCnt': '836', 'male23AgeNmprCnt': '2139', 'feml3AgeNmprCnt': '732',```
	- 데이터프레임 생성

구분	설명
	``item_list = [] for r in raw_data: item = r['Response']['items']['item'] item_list.extend(item) df = DataFrame(item_list) df```

	male4AgeNmprCnt	male76AgeNmprCnt	male31AgeNmprCnt	feml55AgeNmprCnt	mvtCtpvNm	feml47AgeNmprCnt	male84AgeNmprCnt	male68AgeNmprCnt
0	143	71	985	242	서울특별시	359	19	130
1	1	2	29	2	부산광역시	4	0	1
2	2	0	12	1	대구광역시	3	1	0
3	3	1	46	9	인천광역시	12	0	7
4	0	0	12	0	광주광역시	6	0	1
...	...	...	...	...	...	...	...	...
1151	1	0	2	1	전라북도	2	0	0
1152	1	1	2	0	전라남도	1	0	0
1153	0	0	3	1	경상북도	2	0	1
1154	0	0	1	2	경상남도	0	0	0
1155	12	0	48	28	제주특별자치도	19	1	14

1156 rows × 234 columns

기타 설명 및 의견	해당 데이터는 지역별로 전출 및 전입 인구 이동을 알 수 있으며 나이별로도 인구 이동이 나타나 있어 연령별로도 이동량을 알 수 있습니다. 데이터를 수집하는 과정에서 계속 데이터가 추출되지 않고 INVALID_REQUEST_PARAMETER_ERROR 에러가 발생하여 원인을 알아내는데 힘들었으나 자료에서 파라미터에 대한 자세한 설명이 부족하여 파라미터를 잘못 설정해 오류가 발생했다는 것을 알게 되었습니다. 또한 모든 페이지를 추출하는 과정에서 계속 데이터 100개만 추출되어서 for문을 추가하여 모든 데이터를 추출하는 과정과 4개월 데이터를 해당 월별로 추출하는 작업도 진행하였습니다.

■ 데이터 수집 결과

구분	설명
데이터 수집 후 정제 결과	- 불필요한 컬럼 삭제 ```# 컬럼 삭제 df.drop(columns=['mvinAdmmCd', 'mvtAdmmCd', 'mvinDongNm', 'mvtDongNm', 'mvinSggNm', 'mvtSggNm'], inplace=True)```
	- 컬럼 타입 변경 ```# 컬럼 타입 변경 type_change = df.columns[df.columns.str.contains('NmprCnt')] df[type_change] = df[type_change].astype(int) df['statsYm'] = df['statsYm'].astype(int) df.dtypes```
	``male4AgeNmprCnt int32 male76AgeNmprCnt int32 male31AgeNmprCnt int32 feml55AgeNmprCnt int32 mvtCtpvNm object ... male60AgeNmprCnt int32 male103AgeNmprCnt int32 feml83AgeNmprCnt int32 feml70AgeNmprCnt int32 feml96AgeNmprCnt int32 Length: 234, dtype: object```
	-컬럼명 변경

	설명
	# 컬럼명 변경 df.rename(columns={'statsYm':'년월', 'mvinCtpvNm': '전입시도명', 'mvtCtpvNm': '전출시도명', 'totNmprCnt':'총인구수', 'maleNmprCnt':'남성인구수', 'femlNmprCnt':'여성인구수'}, inplace=True) - 순서 변경 # 컬럼 순서 변경 male_age_columns = ['male' + str(age) + 'AgeNmprCnt' for age in range(0, 111)] female_age_columns = ['feml' + str(age) + 'AgeNmprCnt' for age in range(0, 111)] new_column_order = male_age_columns + female_age_columns df = df[['년월','전입시도명','전출시도명','총인구수','남성인구수','여성인구수'] + new_column_order] df[:5]

	년월	전입시도명	전출시도명	총인구수	남성인구수	여성인구수	male0AgeNmprCnt	male1AgeNmprCnt	male2AgeNmprCnt	male3AgeNmprCnt	...	feml
0	202301	서울특별시	서울특별시	54618	26347	28271	157	150	139	142	...	
1	202301	서울특별시	부산광역시	1675	823	852	2	0	0	2	...	

- 남성, 여성 연령대로 컬럼 수정 및 NmprCnt 포함 컬럼 삭제

```
# 남성, 여성 연령대로 컬럼 수정
male_columns = []
female_columns = []
for i in range(0, 91, 10):
    start = i
    end = i + 9
    male_column_name = '{}대 남성'.format(start)
    female_column_name = '{}대 여성'.format(start)
    male_columns.append(df.loc[:, 'male{}AgeNmprCnt'.format(start):'male{}AgeNmprCnt'.format(end)].sum(axis=1))
    female_columns.append(df.loc[:, 'feml{}AgeNmprCnt'.format(start):'feml{}AgeNmprCnt'.format(end)].sum(axis=1))

# 100대에 100세부터 110세까지 추가
male_columns.append(df.loc[:, 'male100AgeNmprCnt':'male110AgeNmprCnt'].sum(axis=1))
female_columns.append(df.loc[:, 'feml100AgeNmprCnt':'feml110AgeNmprCnt'].sum(axis=1))

# 데이터프레임으로 연결하여 합치기
df_male = pd.concat(male_columns, axis=1)
df_female = pd.concat(female_columns, axis=1)

# 컬럼명 설정
male_column_names = ['{}대 남성'.format(i) for i in range(0, 91, 10)] + ['100대 남성']
female_column_names = ['{}대 여성'.format(i) for i in range(0, 91, 10)] + ['100대 여성']
df_male.columns = male_column_names
df_female.columns = female_column_names

# 기존 데이터프레임과 합치기
df = pd.concat([df, df_male, df_female], axis=1)

# NmprCnt 를 포함한 컬럼을 삭제
df = df.drop(columns=df.columns[df.columns.str.contains('NmprCnt')])

df[:2]
```

	년월	전입시도명	전출시도명	총인구수	남성인구수	여성인구수	0대 남성	10대 남성	20대 남성	30대 남성	...	10대 여성	20대 여성	30대 여성	40대 여성	50대 여성	60대 여성	70대 여성	80대 여성	90대 여성	100대 여성
0	202301	서울특별시	서울특별시	54618	26347	28271	1717	1968	6343	6666	...	1898	8373	6465	3818	2695	1911	896	442	99	1
1	202301	서울특별시	부산광역시	1675	823	852	18	45	430	152	...	66	458	126	71	53	30	16	11	0	0

2 rows × 28 columns

구분	설명
	- 정제한 데이터 population_data.csv로 저장 df.to_csv('population_data.csv', encoding='utf-8', index=False) df = pd.read_csv('population_data.csv') df[:2]

	년월	전입시도명	전출시도명	총인구수	남성인구수	여성인구수	0대 남성	10대 남성	20대 남성	30대 남성	...	10대 여성	20대 여성	30대 여성	40대 여성	50대 여성	60대 여성	70대 여성	80대 여성	90대 여성	100대 여성
0	202301	서울특별시	서울특별시	54618	26347	28271	1717	1968	6343	6666	...	1898	8373	6465	3818	2695	1911	896	442	99	1
1	202301	서울특별시	부산광역시	1675	823	852	18	45	430	152	...	66	458	126	71	53	30	16	11	0	0

2 rows × 28 columns

구분	설명
분석 결과 (의견)	불필요한 컬럼 삭제, 컬럼 타입 변경, 컬럼명 변경, 컬럼 순서 변경, 남성과 여성을 연령대로 컬럼 수정을 통해 정제된 데이터는 1156행 28열로 이루어져 있습니다.