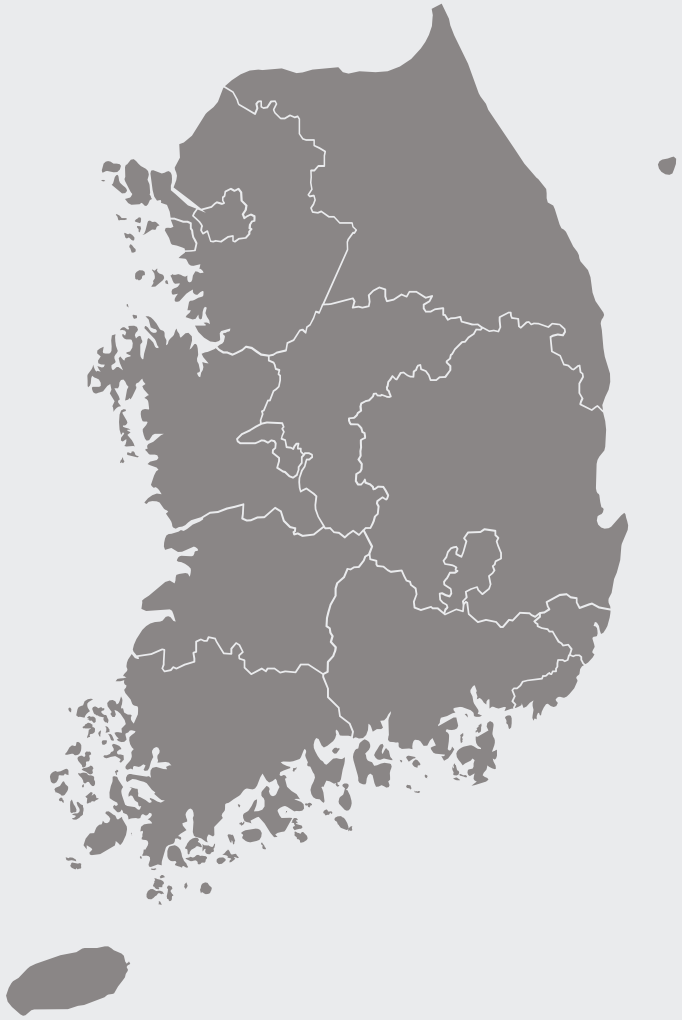
데이터 분석 포트폴리오

지역별 인구 이동 분석

목차



Part 1

개요

◆ 프로젝트명 : 지역별 인구 이동 분석

◆ 수행 기간 : 2023.05.18. - 2023.05.26

◆ 시스템 환경: Windows 11, Python 3.10.9, conda 23.1.0, pandas 2.0.1, matplotlib 3.7.1, seaborn 0.12.2

Part 2

분석배경 및 필요성

◆ 인구 이동은 대규모 신규아파트 입주나 재개발 재건축 , 부동산경기나 고용상황 등의 영향으로 변화하고 있다.

◆ 인구 이동은 경제 상황에 따라 변화하고 이는 지역 경제에 큰 영향을 미치고 있다.

◆ 지역별 및 연령별 인구 이동량을 파악하여 추후에는 지역 간의 국토개발, 교통, 교육 및 주택 등의 각종 정책수립을 위한 자료로 활용할 수 있다.

Part 3

데이터 수집

```
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)
```

공공데이터포털에서 Open API를 통해 데이터 수집

2023년 1월부터 2023년 4월까지(4개월) 인구 이동 데이터 추출

Part 3

데이터수집

```
with open('data.json', 'w') as f:
    json.dump(all_response, f)
```

```
with open('data.json', 'r') as file:
    raw_data = json.load(file)
raw_data
```

추출한 데이터 data.json에 저장 후
파일 읽어와서 데이터프레임 생성

```
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

Part 3

데이터 수집

컬럼명	의미
statsYm	년월
mvinAdmmCd	전입행정기관코드
mvtAdmmCd	전출행정기관코드
mvinCtpvNm	전입시도명
mvinSggNm	전입시군구명
mvinDongNm	전입행정동명
mvtCtpvNm	전출시도명
mvtSggNm	전출시군구명
mvtDongNm	전출행정동명
totNmprCnt	총인구수
maleNmprCnt	남자인구수
femlNmprCnt	여자인구수
male0-110AgeNmprCnt	만 0-110세 남자
feml0-110AgeNmprCnt	만 0-110세 여자

데이터프레임 구성

◆ 1156행 234열로 구성됨

◆ 모든 컬럼은 object 타입

◆ 결측치 없음

Part 4

전처리

```
# 컬럼 삭제
df.drop(columns=['mvinAdmmCd', 'mvtAdmmCd', 'mvinDongNm', 'mvtDongNm', 'mvinSggNm', 'mvtSggNm'],
        inplace=True)
```

➡

불필요한 mvinAdmmCd, mvtAdmmCd, mvinDongNm, mvtDongNm, mvinSggNm, mvtSggNm 컬럼 삭제

```
# 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
```

➡

NmprCnt가 포함된 컬럼과 statsYm 컬럼의 타입을 object 에서 int로 변환

Part 4

전처리

```
# 컬럼명 변경
df.rename(columns={'statsYm':'년월',
                   'mvinCtpvNm': '전입시도명',
                   'mvtCtpvNm': '전출시도명',
                   'totNmprCnt':'총인구수',
                   'maleNmprCnt':'남성인구수',
                   'femlNmprCnt':'여성인구수'},
          inplace=True)
```

➡ 기존 컬럼명을 알아보기 쉽게 한국어 명으로 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	...	

➡ 연령별로 묶기 위해 0-110세까지의 남성과 여성 수 컬럼을 오름차순으로 순서 변경

Part 4

전처리

```
# 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 변환하여 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 컬럼들 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

0-110세 남성과 여성 수 컬럼을 연령대별 컬럼으로 수정

0-9세는 0대, 10-19세는 10대, 20-29세는 20대와 같이 컬럼을 생성하여 해당 연령의 인구 수를 합함

이때 110세는 100대 컬럼으로 처리

연령대별로 남성과 여성의 컬럼을 생성한 후 기존에 있던 NmprCnt 포함하는 컬럼을 삭제

Part 4

전처리

```
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

➡ 전처리를 끝낸 데이터프레임을 population.csv로 index를 제외하고 저장함
전처리 후 데이터는 1156행 28열로 구성됨

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1156 entries, 0 to 1155
Data columns (total 28 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   년월         1156 non-null   int64
 1   전입시도명      1156 non-null   object
 2   전출시도명      1156 non-null   object
 3   총인구수       1156 non-null   int64
 4   남성인구수      1156 non-null   int64
 5   여성인구수      1156 non-null   int64
 6   0대 남성      1156 non-null   int64
 7   10대 남성     1156 non-null   int64
 8   20대 남성     1156 non-null   int64
 9   30대 남성     1156 non-null   int64
 10  40대 남성     1156 non-null   int64
 11  50대 남성     1156 non-null   int64
 12  60대 남성     1156 non-null   int64
 13  70대 남성     1156 non-null   int64
 14  80대 남성     1156 non-null   int64
 15  90대 남성     1156 non-null   int64
 16  100대 남성    1156 non-null   int64
 17  0대 여성      1156 non-null   int64
 18  10대 여성     1156 non-null   int64
 19  20대 여성     1156 non-null   int64
 20  30대 여성     1156 non-null   int64
 21  40대 여성     1156 non-null   int64
 22  50대 여성     1156 non-null   int64
 23  60대 여성     1156 non-null   int64
 24  70대 여성     1156 non-null   int64
 25  80대 여성     1156 non-null   int64
 26  90대 여성     1156 non-null   int64
 27  100대 여성    1156 non-null   int64
dtypes: int64(26), object(2)
memory usage: 253.0+ KB
```

Part 5

가설

가설 1	2023년 1월부터 2023년 4월까지 서울특별시로 전입한 인구 중 비율이 여성보다 남성이 더 높을 것이다.
가설 2	2023년 1월부터 2023년 4월까지 전입률이 가장 높은 시도는 서울특별시일 것이다.
가설 3	2023년 1월부터 2023년 4월까지 서울특별시에서 전출 인구 수는 연령대가 높아질수록 줄어들 것이다.
가설 4	2023년 1월부터 2023년 4월까지 같은 시도 내에서 전입, 전출한 인구 수는 서울특별시가 가장 많을 것이다.
가설 5	2023년 1월부터 2023년 4월까지 여성과 남성의 전입 및 전출률이 가장 높은 월은 3월일 것이다.

Part 5

가설 1)

가설 1 2023년 1월부터 2023년 4월까지 서울특별시로 전입한 인구 중 비율이 여성보다 남성이 더 높을 것이다.

```
df[['남성인구수', '여성인구수']][df.전입시도명 =='서울특별시']
```

	남성인구수	여성인구수
0	26347	28271
1	823	852
2	592	668
3	1299	1298
4	446	458
...	...	...
879	447	453
880	344	323
881	442	483
882	552	566
883	276	275

68 rows × 2 columns

```
df[['남성인구수', '여성인구수']][df.전입시도명 =='서울특별시'].sum(axis=0)
```

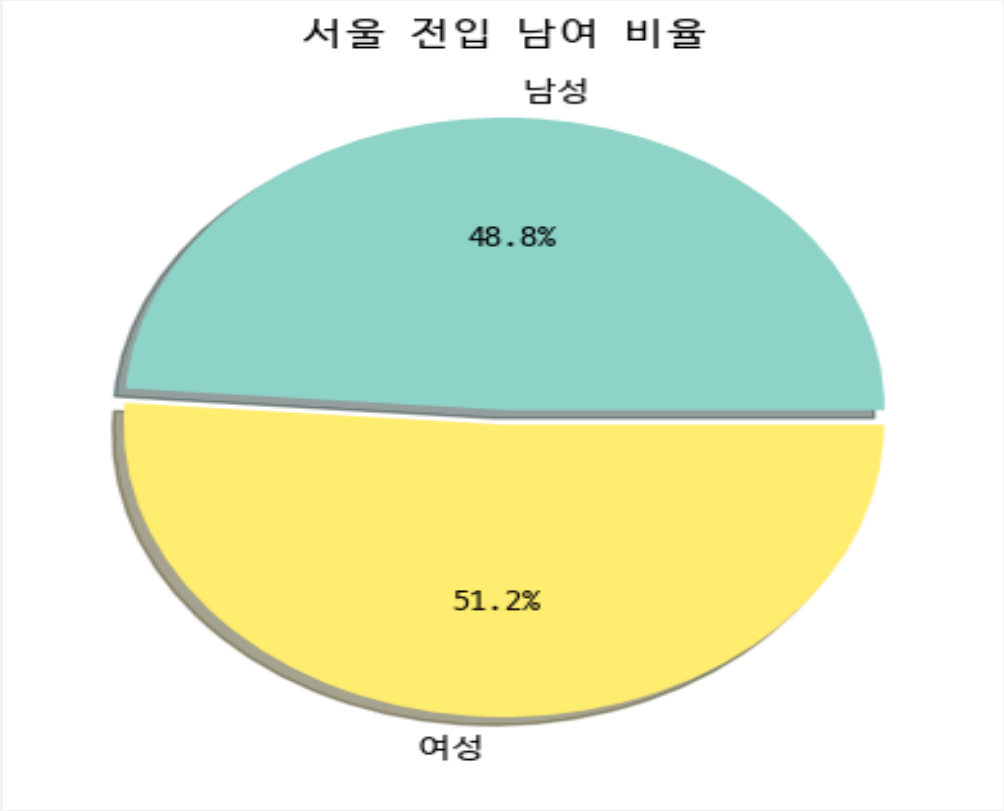
```
남성인구수    207691
여성인구수    217784
dtype: int64
```

➡ 전입시도명이 서울특별시인 남성인구수와 여성인구수 컬럼을 뽑아내 행의 합을 구함

Part 5

가설 1)

가설 1 2023년 1월부터 2023년 4월까지 서울특별시로 전입한 인구 중 비율이 여성보다 남성이 더 높을 것이다.



분석 결과 서울특별시로 전입한 인구 중 남성은 총 207,691명, 여성은 217,784명으로 나타났다.

그래프를 통해 확인한 결과 여성은 약 51.2%, 남성은 약 48.8%로 서울특별시로 전입한 인구는 여성의 비율이 남성보다 약 2.4% 높게 측정되었다.

따라서 서울특별시로 전입한 인구 중 남성의 비율이 높을 것이라는 가설은 기각되었다.

Part 5

가설2)

가설 2 2023년 1월부터 2023년 4월까지 전입률이 가장 높은 시도는 서울특별시일 것이다.

```
region_sum = df.groupby('전입시도명').총인구수.sum().sort_values(ascending=False).to_frame()
region_sum.reset_index(inplace=True)
region_sum
```

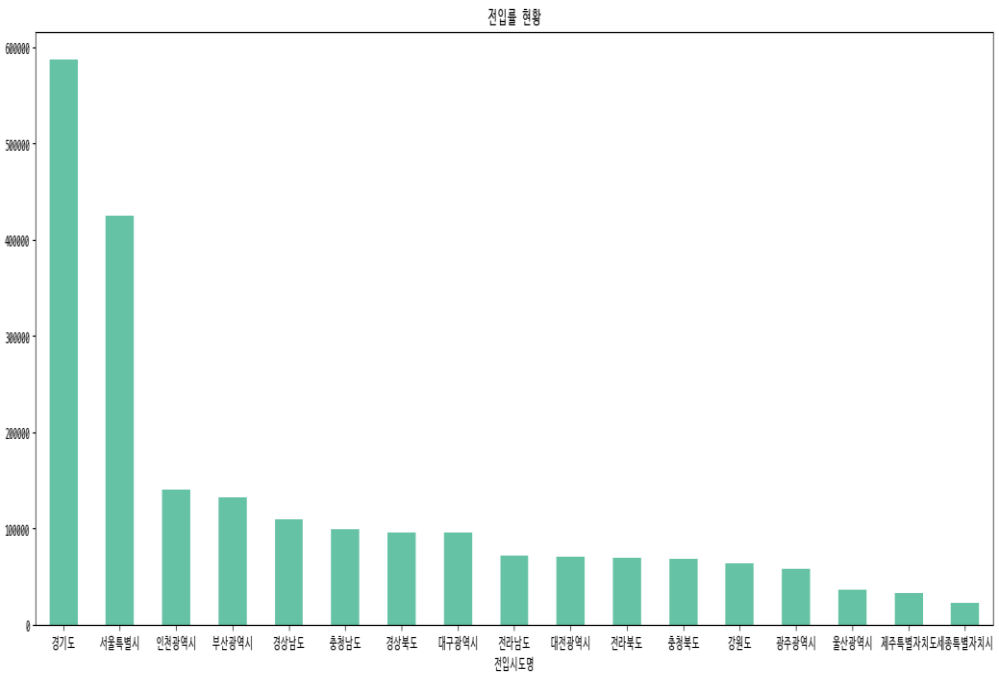
전입시도명	총인구수
경기도	586809
서울특별시	425475
인천광역시	139855
부산광역시	132716
경상남도	109157
충청남도	99290
경상북도	95781
대구광역시	95546
전라남도	71177
대전광역시	70205
전라북도	69681
충청북도	68616
강원도	63536
광주광역시	57749
울산광역시	36650
제주특별자치도	32483
세종특별자치시	22320

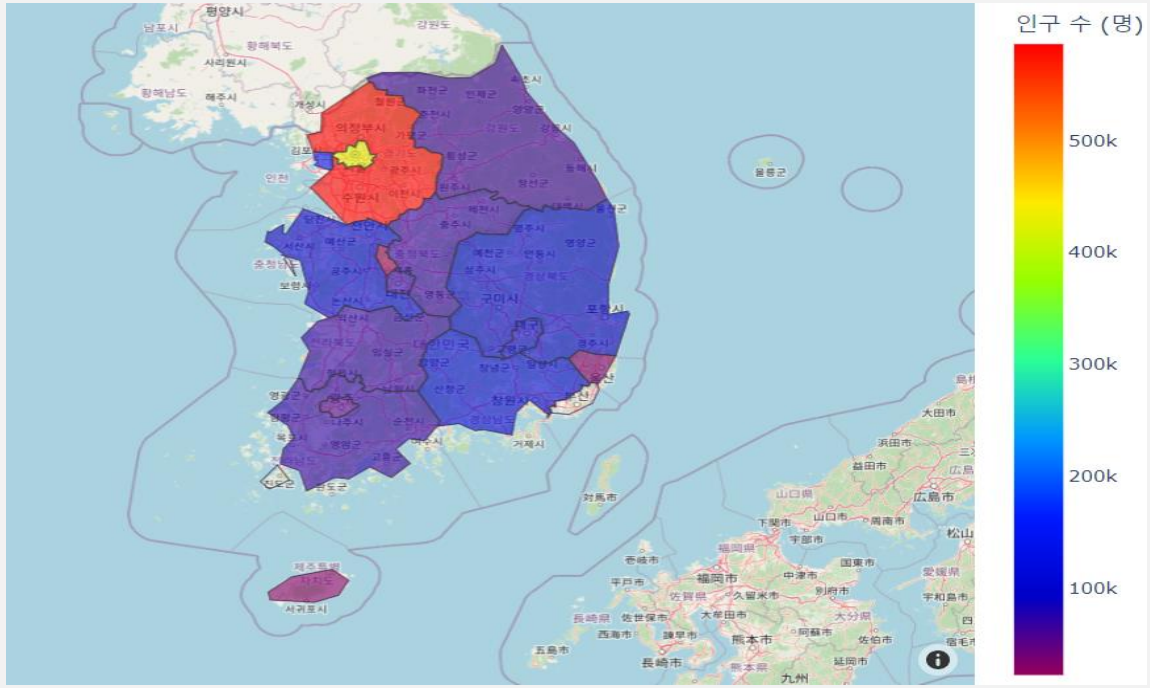
➡ 전입시도명으로 그룹화를 한 다음 총인구수 컬럼을 합하여 내림차순 정렬

Part 5

가설2)

가설 2	2023년 1월부터 2023년 4월까지 전입률이 가장 높은 시도는 서울특별시일 것이다.

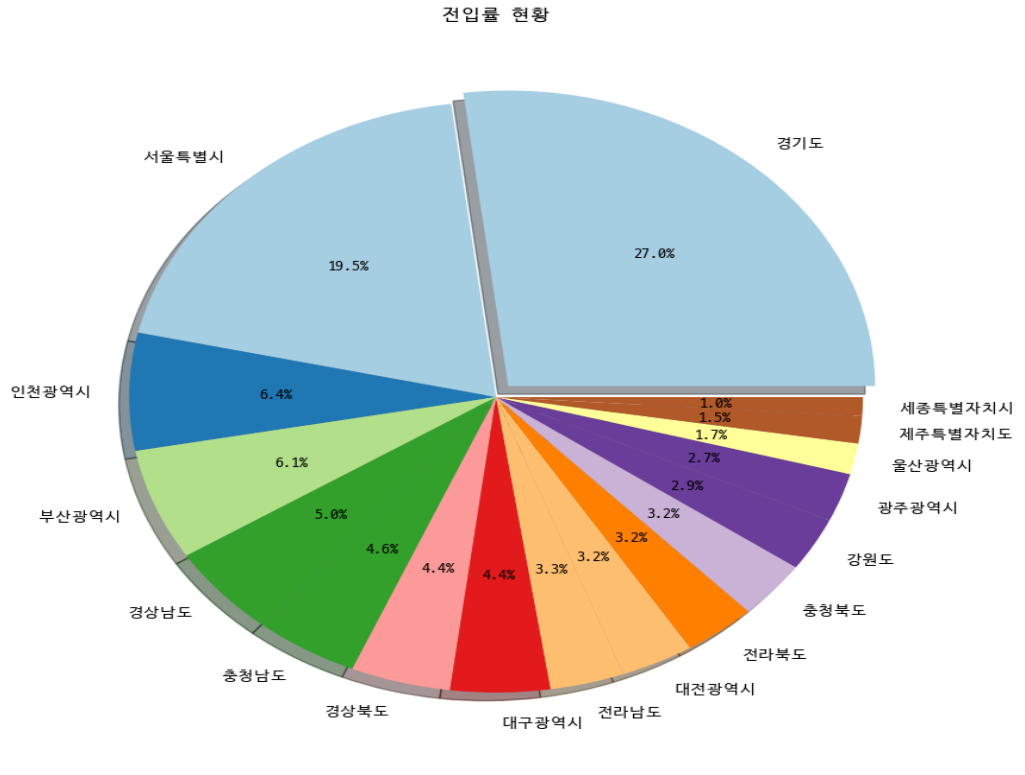




Part 5

가설2)

가설 2 2023년 1월부터 2023년 4월까지 전입률이 가장 높은 시도는 서울특별시일 것이다.



전입한 인구 수는 높은 순으로 경기도는 586,809명, 서울특별시는 425,475명, 인천광역시는 139,855명이며 세종특별자치시가 22,320명으로 가장 낮은 것으로 나타났다.

분석 결과 전입률이 높은 시도는 경기도가 약 27%로 가장 높으며 서울특별시가 약 19.5%로 두 번째로 높고 세종특별자치시가 약 1%로 가장 낮은 것으로 측정되었다.

따라서 서울특별시가 전입률가 가장 높을 것이라는 가설은 기각되었다.

Part 5

가설 3)

가설 3 2023년 1월부터 2023년 4월까지 서울특별시에서 전출 인구 수는 연령대가 높아질수록 줄어들 것이다.

```
seoul_sum = df[df.전출시도명 =='서울특별시'].sum()
seoul_sum[6:].rename('합계').to_frame().sort_values(by='합계', ascending=False)
```

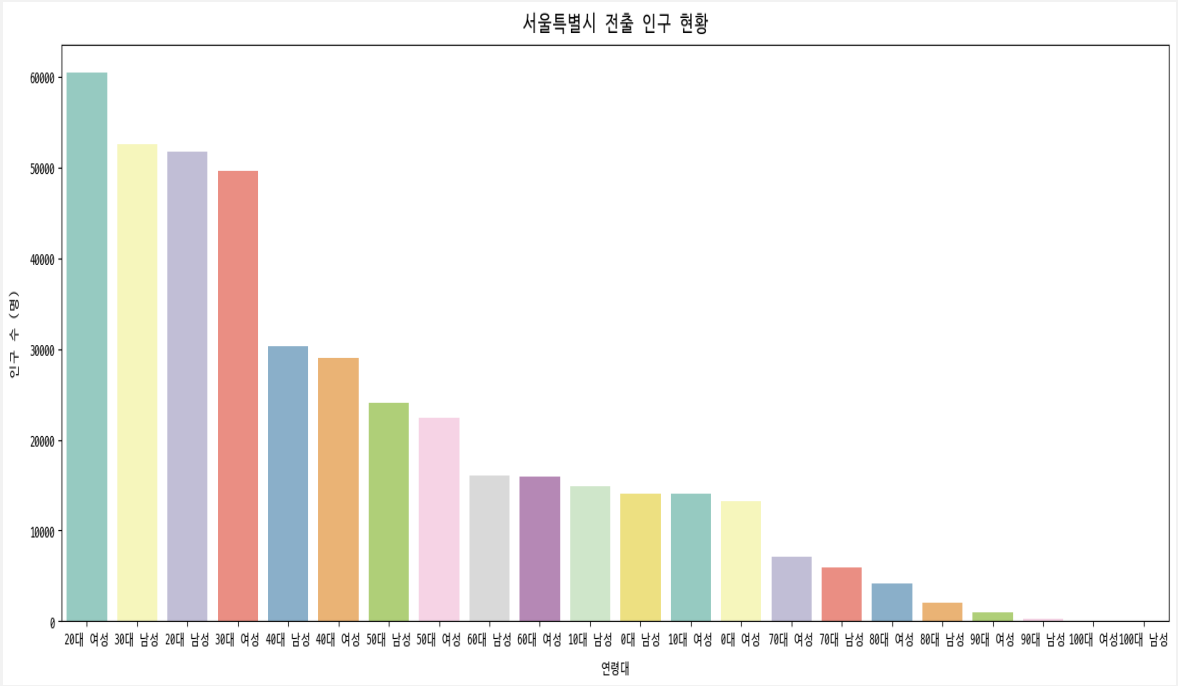
	합계
20대 여성	60471
30대 남성	52639
20대 남성	51811
30대 여성	49646
40대 남성	30308
40대 여성	29033
50대 남성	24088
50대 여성	22457
60대 남성	16057
60대 여성	15953
10대 남성	14945
0대 남성	14040
10대 여성	14038
0대 여성	13211
70대 여성	7077
70대 남성	5974
80대 여성	4222
80대 남성	2098
90대 여성	1014
90대 남성	226
100대 여성	34
100대 남성	6

➡ 전출시도명이 서울특별시인 데이터에서 연령대별 합계를 구한 후 합계를 기준으로 내림차순 정렬

Part 5

가설 3)

가설 3 2023년 1월부터 2023년 4월까지 서울특별시에서 전출 인구 수는 연령대가 높아질수록 줄어들 것이다.



서울특별시에서 전출한 인구 중 20대 여성이 60,471명으로 가장 높게 나타났으며 30대 남성은 52,639명, 20대 남성은 51,811명, 30대 여성은 49,646명으로 나타나 20-30대가 서울특별시에서 전출 인구 수가 많은 것으로 나타났다.

90대 여성 1014명, 90대 남성 226명, 100대 여성 34명, 100대 남성 6명으로 전출 인구 수가 점차 줄어드는 것으로 나타났다.

따라서 미성년자인 0대와 10대를 제외한 연령대에서 연령대가 높아질수록 전출 인구 수가 줄어들어 연령대가 높아질수록 전출 인구 수가 감소할 것이라는 가설은 기각되지 않았다.

Part 5

가설 4)

가설 4 2023년 1월부터 2023년 4월까지 같은 시도 내에서 전입, 전출한 인구 수는 서울특별시가 가장 많을 것이다.

```
same_region = df[['전입시도명', '총인구수']][df.전입시도명 == df.전출시도명].groupby('전입시도명').sum().sort_values(by='총인구수', ascending=False)
same_region
```

전입시도명	총인구수
경기도	385007
서울특별시	258795
부산광역시	90757
인천광역시	82224
경상남도	68821
대구광역시	63337
충청남도	53614
경상북도	52956
전라북도	47695
전라남도	42769
충청북도	39635
대전광역시	39577
강원도	36848
광주광역시	36676
울산광역시	22319
제주특별자치도	19609
세종특별자치시	7885

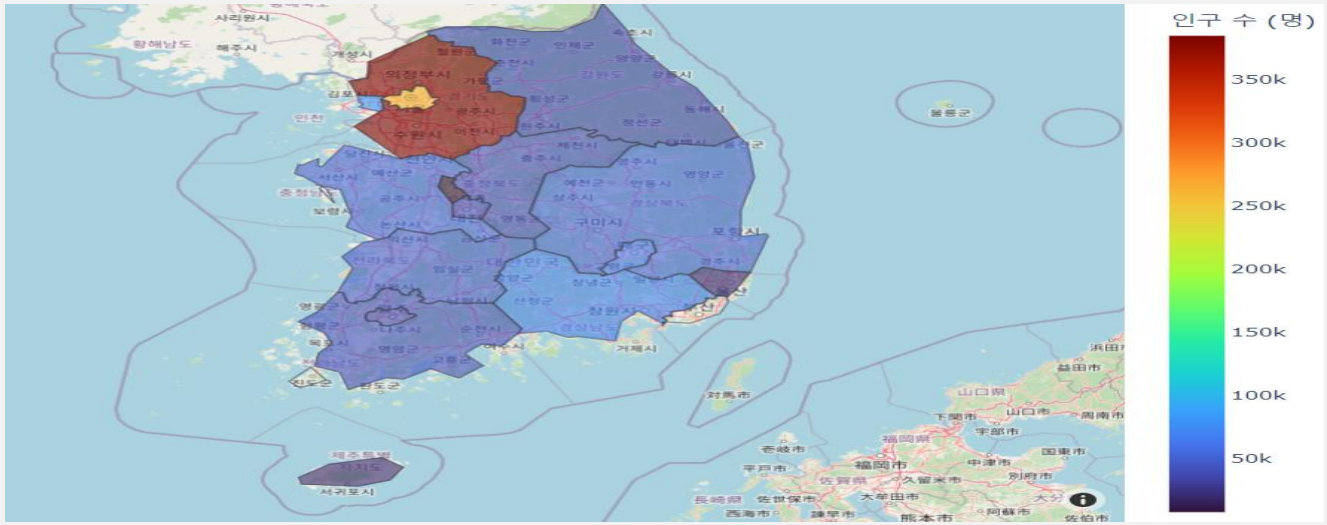
➡

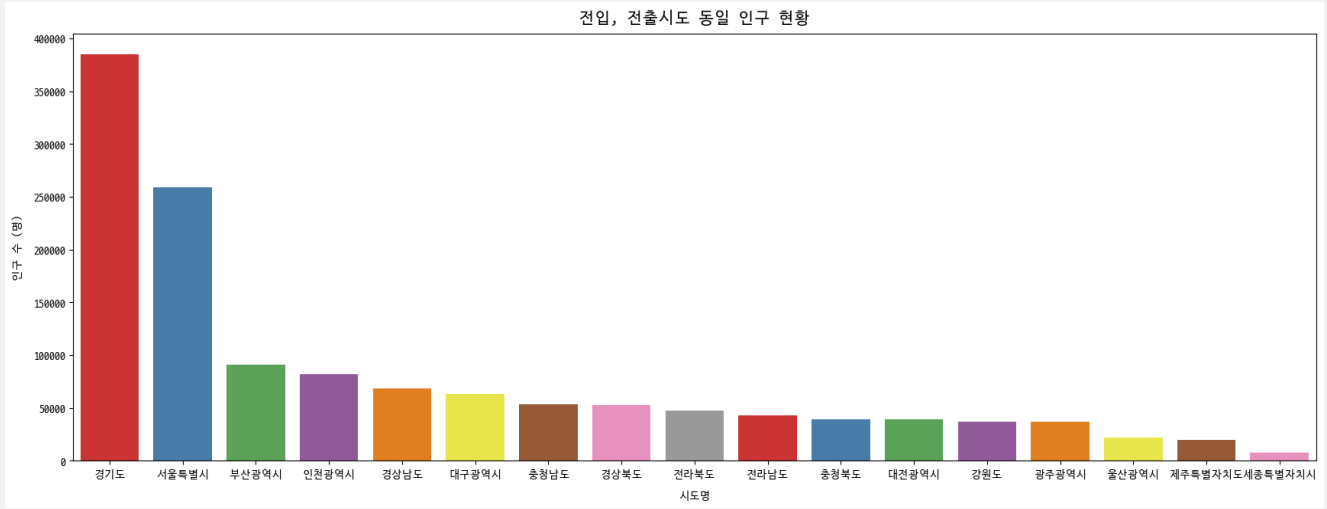
전입시도명과 전출시도명이 같은 데이터 중 전입시도명과 총인구수 컬럼만 추출하여 총 인구수를 기준으로 내림차순 정렬

Part 5

가설 4)

가설 4 2023년 1월부터 2023년 4월까지 같은 시도 내에서 전입, 전출한 인구 수는 서울특별시가 가장 많을 것이다.





같은 시도 내에서 전입, 전출한 인구 수는 경기도가 385,007명으로 가장 높으며 서울특별시가 258,795명으로 두 번째로 높고 세종특별자치시가 7,885명으로 가장 낮은 것으로 나타났다.

따라서 서울특별시 전입시도와 전출시도가 동일한 인구 수가 가장 많을 것이라는 가설은 기각되었다.

Part 5

가설 5)

가설 5 2023년 1월부터 2023년 4월까지 여성과 남성의 전입 및 전출률이 가장 높은 월은 3월일 것이다.

```
df.groupby('년월')[['여성인구수', '남성인구수']].sum()
```

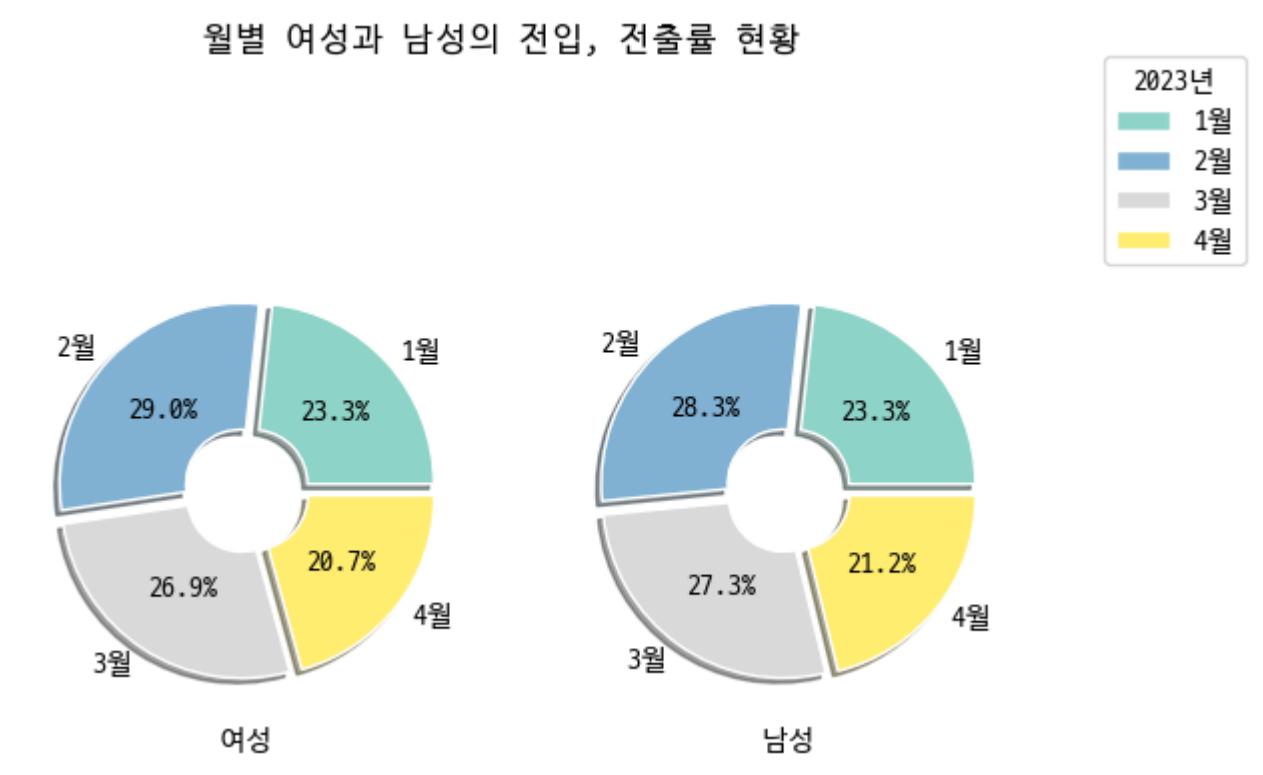
년월	여성인구수	남성인구수
202301	246635	260346
202302	306976	316578
202303	284950	305188
202304	219125	237248

➡ 년월에 따라 여성인구수와 남성 인구수를 더함

Part 5

가설 5)

가설 5 2023년 1월부터 2023년 4월까지 여성과 남성의 전입 및 전출률이 가장 높은 월은 3월일 것이다.



여성과 남성의 전입 및 전출 인구 수는 1월에는 여성은 246,635명, 남성은 260,346명이며 2월은 여성은 306,976명, 남성은 316,578명, 3월은 여성은 284,950명, 남성은 305,188명, 4월은 여성은 219,125명, 남성은 237,248명으로 여성과 남성 모두 2월에 전입 및 전출한 인구 수가 가장 많은 것으로 나타났다.

2월 전입 및 전출률이 여성은 29%, 남성은 28.3%로 다른 월에 비해 조금 더 높게 나타났다.

따라서 여성과 남성의 전입 및 전출률이 가장 높은 월은 3월이라는 가설은 기각되었다.

Part 6

한계점

◆ 인구 이동 현황은 알 수 있으나 인구 이동 원인을 해당 데이터 분석만으로는 알 수 없음

→ 인구 이동 원인이 나타난 데이터와 같이 분석을 하면 더 좋은 분석 결과가 나타날 것임

◆ 4개월간의 인구 이동 데이터로 한정되었기에 데이터가 정확하다고 할 수 없음

→ 4개월로 특정 지어 분석하는 것이 아닌 1년 단위로 월별로 인구 이동을 분석하면 더 구체적인 분석 결과가 나올 것임

감사합니다