
거리 내 학교 수와 주택 가격의 상관관계 분석

목차

01 주제 선정 이유

02 데이터 수집 방법

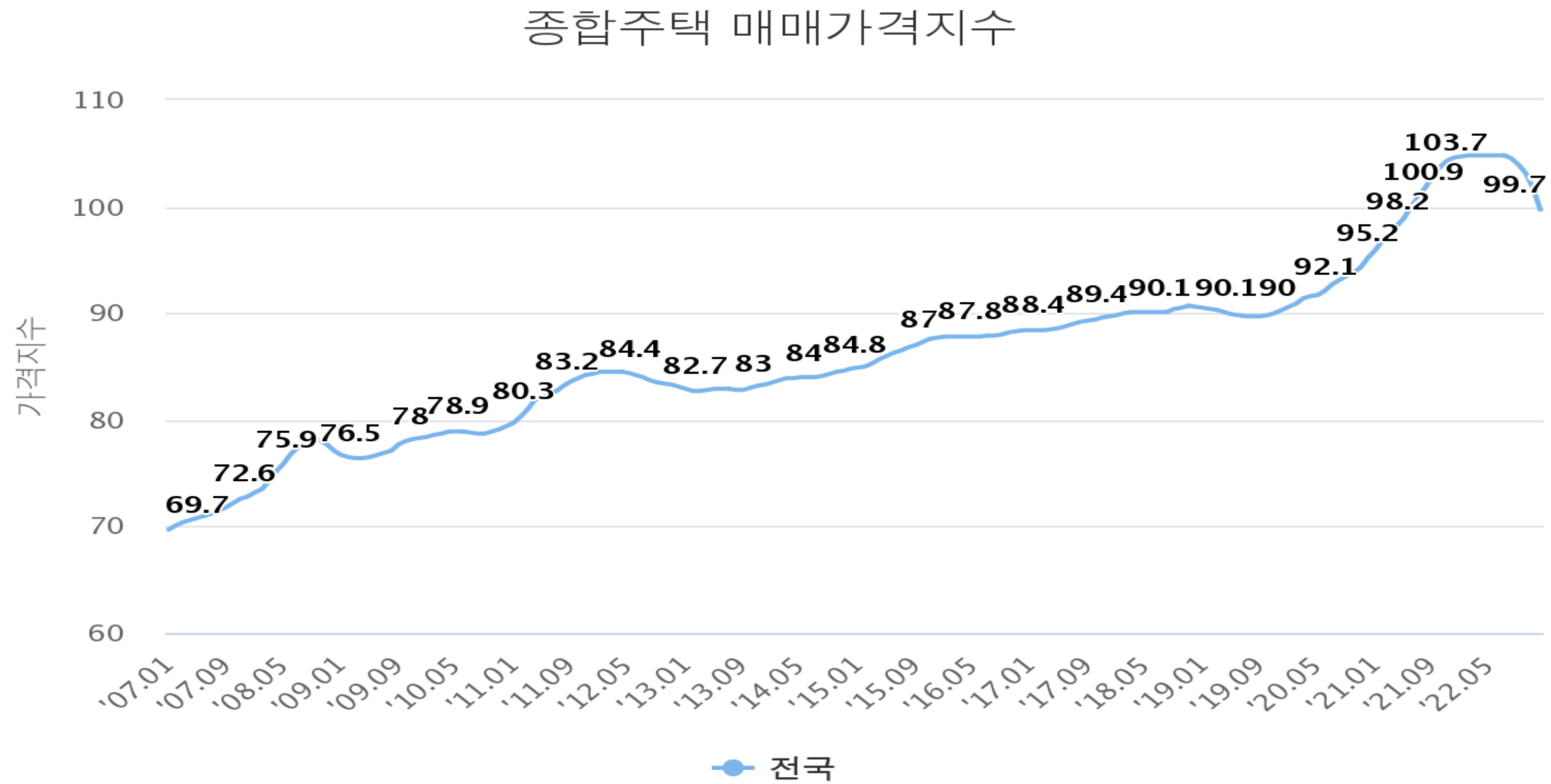
03 데이터 전처리

04 데이터 분석

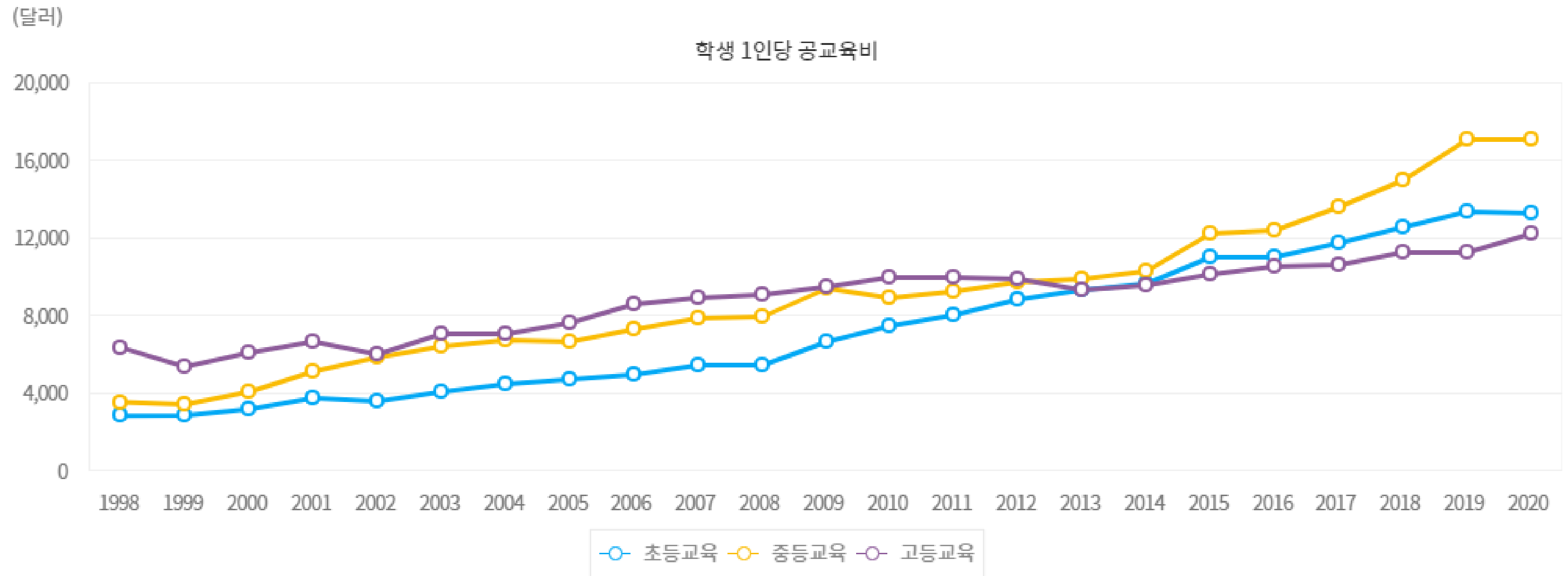
05 결과 및 해석

06 결론 및 활용방안

01 주제 선정 이유

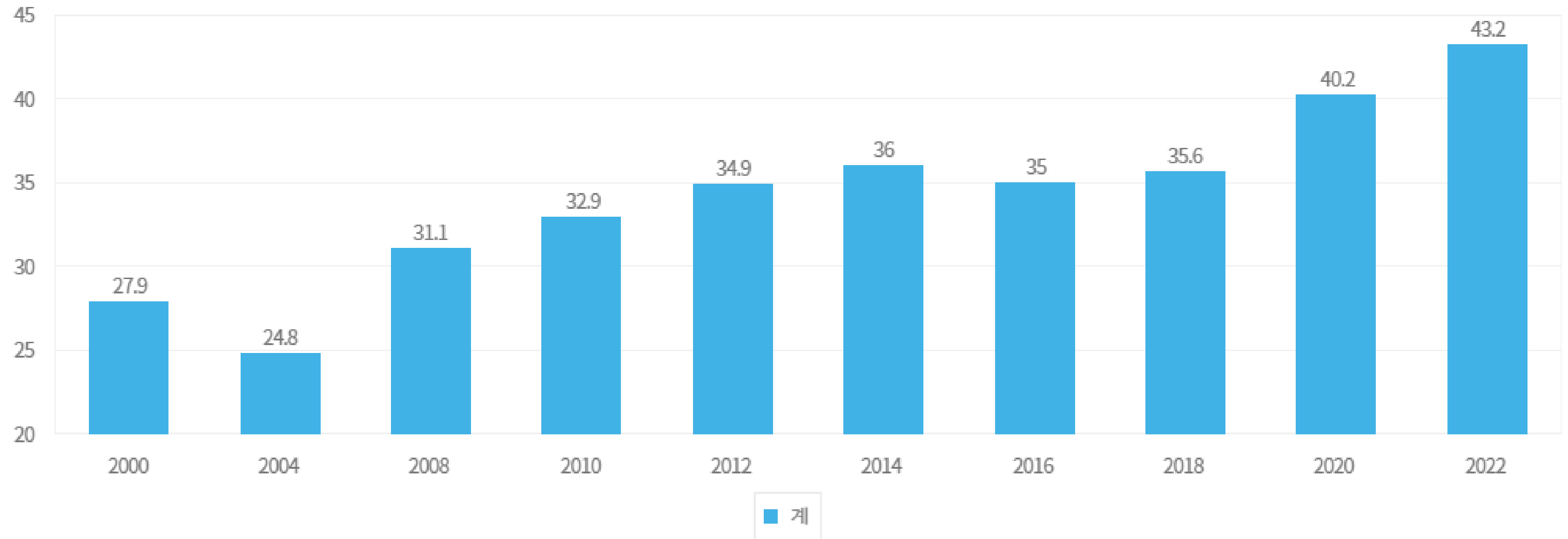


01 주제 선정 이유



01 주제 선정 이유

학교 교육의 효과에 대해 긍정적으로 인식하는 인구의 비율



01 주제 선정 이유

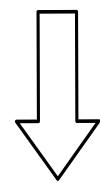
IF?

01 주제 선정 이유

주변에 학교가 많아지면 주택 가격은 비싸질까?

01 주제 선정 이유

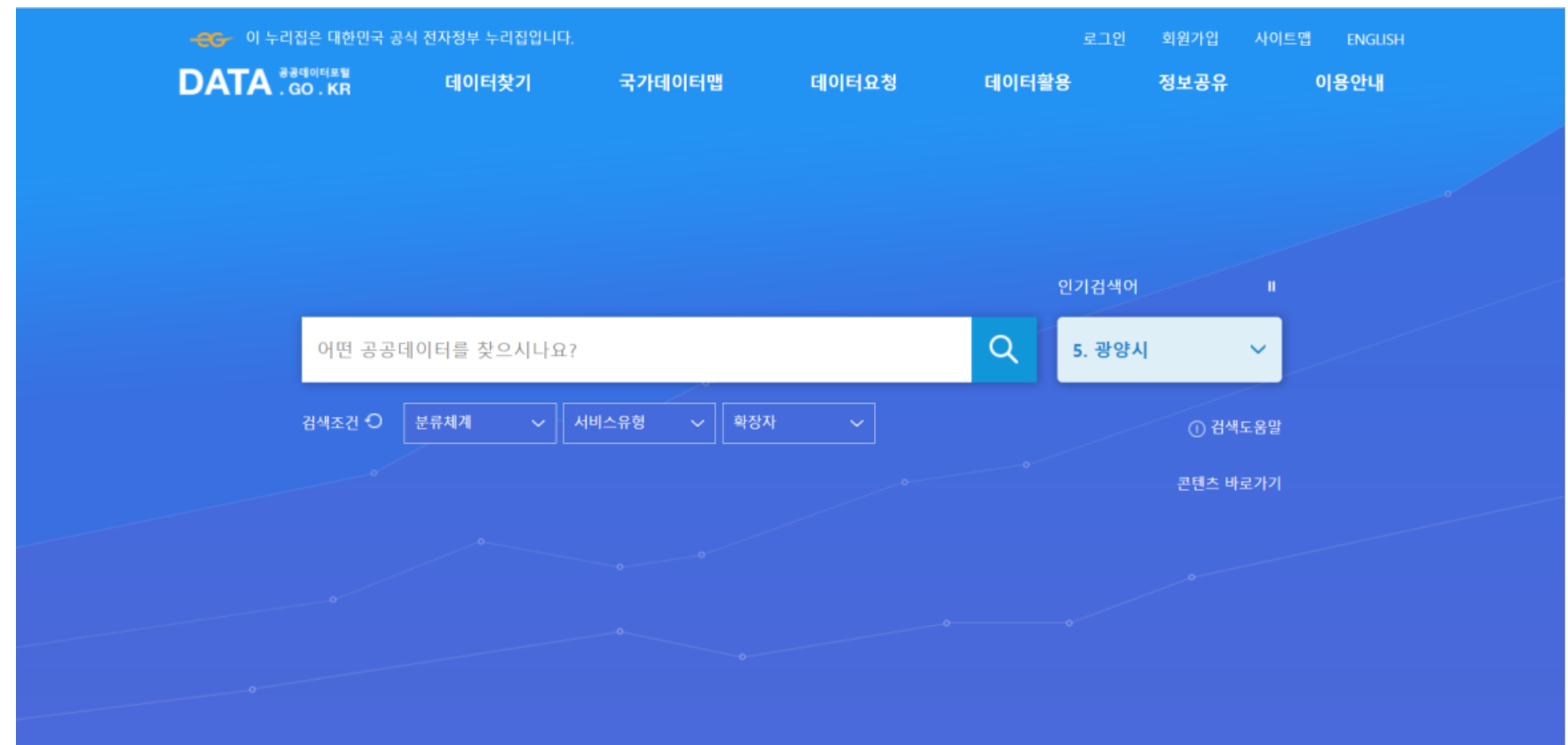
**공교육의 긍정적인 인식이 증가함에 따라
학교 주변을 선호하는 사람들이 증가한다면?**



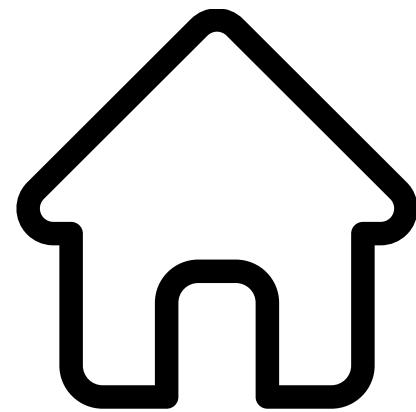
학교랑 가까울수록 주택 가격이 상승할 가능성이 있음.

02 데이터 수집 방법

공공데이터포털

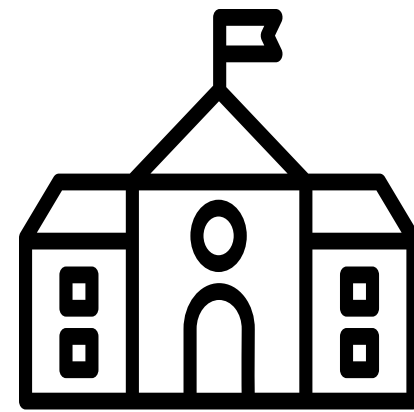


02 데이터 수집 방법



국토교통부에서 제공하는 2023년도 공동주택 공시가격 정보
<https://www.data.go.kr/data/3073746/fileData.do>

02 데이터 수집 방법



지방교육재정연구원에서 제공하는 전국 초등학교, 중학교, 고등학교의 위치정보(좌표)

<https://www.data.go.kr/data/15099519/fileData.do#/tab-layer-file>

03 데이터 전처리

```
▶ # CSV 파일 경로  
house_file_path = "/content/data/house.csv"  
school_file_path = "/content/data/school.csv"
```

```
[9] house_data = pd.read_csv(house_file_path, encoding = 'cp949')
```

```
↳ <ipython-input-9-d5c73c4a2990>:1: DtypeWarning: Columns (5,6,11,14) have mixed types. Specify dtype option on import or set low_memory=False.  
house_data = pd.read_csv(house_file_path, encoding = 'cp949')
```

```
[14] school_data = pd.read_csv(school_file_path, encoding = 'cp949')
```

**다운로드 받은 csv 파일의 경로를 지정하고
파이썬 pandas의 read_csv()를 이용해서 파일 불러오기**

03 데이터 전처리

house_data.head()
house_data.tail()

	기준연 도	기준 월	법정동코드	도로명주소	시도	시군 구	읍면	동리	특수지 코드	본번	부 번	특수지 명	단지명	동명	호 명	전용면 적	공시가격	단지코드	동코 드	호코 드
14863976	2023	1	5013032025	제주특별자치도 서귀포시 세화로26번길 11	제주특별자치도	서귀포시	표선면	세화리	0	1512	5	NaN	(1512-5)	103동	202	63.6582	151000000	20373811	3	2
14863977	2023	1	5013032025	제주특별자치도 서귀포시 세화로26번길 11	제주특별자치도	서귀포시	표선면	세화리	0	1512	5	NaN	(1512-5)	103동	301	63.6582	153000000	20373811	3	3
14863978	2023	1	5013032025	제주특별자치도 서귀포시 세화로26번길 11	제주특별자치도	서귀포시	표선면	세화리	0	1512	5	NaN	(1512-5)	103동	302	63.6582	153000000	20373811	3	4
14863979	2023	1	5013032025	제주특별자치도 서귀포시 세화로26번길 11	제주특별자치도	서귀포시	표선면	세화리	0	1512	5	NaN	(1512-5)	103동	401	63.6582	153000000	20373811	3	5
14863980	2023	1	5013032025	제주특별자치도 서귀포시 세화로26번길 11	제주특별자치도	서귀포시	표선면	세화리	0	1512	5	NaN	(1512-5)	103동	402	63.6582	153000000	20373811	3	6

head()와 tail()을 이용해서 불러온 주택 데이터의 일부를 확인

03 데이터 전처리

```
[15] school_data.head()  
school_data.tail()
```

	학교 ID	학교명	학교구분	설립일자	설립형태	본교분교구분	운영상태	소재지지번주소	소재지도로명주소	시도교육청코드	시도교육청명	교육지원청코드	교육지원청명	생성일자	변경일자	위도	경도	데이터기준일자
11984	B000027484	광주예술고등학교	고등학교	1983-03-15	공립	본교	운영	광주광역시 북구 매곡동 385	광주광역시 북구 서하로 72	7380000	광주광역시교육청	7391000	광주광역시동부교육지원청	2013-11-29	2023-07-03	35.186556	126.888059	2023-09-22
11985	B000023850	양덕중학교	중학교	2018-11-12	공립	본교	운영	경상북도 포항시 북구 양덕동 2027	경상북도 포항시 북구 장량로241번길 22	8750000	경상북도교육청	8761000	경상북도포항교육지원청	2013-11-29	2023-07-03	36.083986	129.407206	2023-09-22
11986	B000025977	망포조등학교	초등학교	2019-05-01	공립	본교	운영	경기도 수원시 영통구 망포동 750	경기도 수원시 영통구 동탄지성로 550-10	7530000	경기도교육청	7541000	경기도수원교육지원청	2013-11-29	2023-07-03	37.240598	127.047323	2023-09-22
11987	B000011074	서포중학교	중학교	1955-09-26	공립	본교	운영	경상남도 사천시 서포면 구평리 916	경상남도 사천시 서포면 서포로 293-6	9010000	경상남도교육청	9081000	경상남도사천교육지원청	2013-11-29	2023-07-03	35.008273	127.974528	2023-09-22
11988	B000013326	대아고등학교	고등학교	1966-01-22	사립	본교	운영	경상남도 진주시 이현동 316	경상남도 진주시 서장대로185번길 14	9010000	경상남도교육청	9051000	경상남도진주교육지원청	2013-11-29	2023-07-03	35.189314	128.057571	2023-09-22

head()와 tail()을 이용해서 불러온 학교 데이터의 일부를 확인

03 데이터 전처리

```
house_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14863981 entries, 0 to 14863980
Data columns (total 20 columns):
#   Column      Dtype
---  ---
0   기준연도    int64
1   기준월      int64
2   법정동코드  int64
3   도로명주소  object
4   시도        object
5   시군구      object
6   읍면        object
7   동리        object
8   특수지코드  int64
9   본번        int64
10  부번        int64
11  특수지명    object
12  단지명      object
13  동명        object
14  호명        object
15  전용면적    float64
16  공시가격    int64
17  단지코드    int64
18  동코드      int64
19  호코드      int64
dtypes: float64(1), int64(10), object(9)
memory usage: 2.2+ GB
```

```
[16] school_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11989 entries, 0 to 11988
Data columns (total 18 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   학교ID      11989 non-null  object
1   학교명      11989 non-null  object
2   학교급구분  11989 non-null  object
3   설립일자    11989 non-null  object
4   설립형태    11989 non-null  object
5   본교분교구분 11989 non-null  object
6   운영상태    11989 non-null  object
7   소재지지번주소 11989 non-null  object
8   소재지도로명주소 11989 non-null  object
9   시도교육청코드 11989 non-null  int64
10  시도교육청명  11989 non-null  object
11  교육지원청코드 11989 non-null  int64
12  교육지원청명  11989 non-null  object
13  생성일자    11989 non-null  object
14  변경일자    11989 non-null  object
15  위도        11989 non-null  float64
16  경도        11989 non-null  float64
17  데이터기준일자 11989 non-null  object
dtypes: float64(2), int64(2), object(14)
memory usage: 1.6+ MB
```

info()를 이용해서 결측치가 있는지 확인

03 데이터 전처리

```
▶ # 사용할 컬럼만 따로 지정
house_columns = ['도로명주소', '전용면적', '공시가격']
school_columns = ['소재지도로명주소', '학교명']
```


```
[19] # 데이터를 건너뛰면서 불러오기 위함
def skip_logic(index, skip_num):
    if index % skip_num == 0:
        return False
    return True
```

```
▶ # 읽어들 행의 수
num_rows_to_read = 14000
```

```
[ ] # 필요한 컬럼만 선택하여 로드
house_data = pd.read_csv(house_file_path, encoding='cp949', usecols=house_columns, skiprows = lambda x: skip_logic(x, 1000), nrows = num_rows_to_read)
school_data = pd.read_csv(school_file_path, encoding='cp949', usecols=school_columns)
```

체계적 표본추출 방법을 이용해서 주택 데이터 중 14000개 데이터만 사용

03 데이터 전처리



```
# 주소를 이용하여 위도와 경도를 반환하는 함수 정의
def get_lat_lon(address):
    try:
        location = geolocator.geocode(address, timeout=10)
        return (location.latitude, location.longitude) if location else (None, None)
    except Exception as e:
        print(f"Error during geocoding: {e}")
        return (None, None)
```

geopy를 이용해서 주소를 위도, 경도 값으로 변환하는 함수 정의

03 데이터 전처리

```
▶ # 주택 데이터에 대한 위도, 경도 계산
house_data['주택 좌표'] = house_data['도로명주소'].apply(lambda x: get_lat_lon(x))
house_data[['주택 위도', '주택 경도']] = pd.DataFrame(house_data['주택 좌표'].tolist(), index=house_data.index)
house_data['단위 면적당 가격'] = house_data['공시가격'] / house_data['전용면적']
house_data = house_data[['주택 위도', '주택 경도', '도로명주소', '단위 면적당 가격']]
```

```
[ ] # 학교 데이터에 대한 위도, 경도 계산
school_data['학교 좌표'] = school_data['소재지도로명주소'].apply(lambda x: get_lat_lon(x))
school_data[['학교 위도', '학교 경도']] = pd.DataFrame(school_data['학교 좌표'].tolist(), index=school_data.index)
school_data = school_data[['학교 위도', '학교 경도', '학교명']]
```

**주택과 학교의 도로명주소를 위도, 경도 값으로 변환
이때, 주택의 공시가격을 전용면적으로 나누어 단위 면적당 가격을 계산**

03 데이터 전처리

```
[ ] #결측치 유무 확인
    house_data.info()
    school_data.info()

    #결측치가 있는 행 제거
    house_data = house_data.dropna()
    school_data = school_data.dropna()
```

새로 생성된 데이터에 결측치가 있는지 확인 후 결측치 제거

03 데이터 전처리

```
[ ] # 동일한 주소의 주택에 대한 단위 면적당 평균가격 계산
    house_data['단위 면적당 평균가격'] = house_data.groupby('도로명주소')['단위 면적당 가격'].transform('mean')
```

```
[ ] # 중복된 도로명 주소 제거
    unique_house_data = house_data.drop_duplicates(subset='도로명주소')
```

주소가 동일한 주택에 대한 단위 면적당 평균가격을 계산 후 중복 제거

03 데이터 전처리

```
[ ] # 반경 내에 있는 학교의 수를 세는 함수
def count_schools_within_radius(house_coords, school_data, radius_km):
    count = 0
    for _, school_row in school_data.iterrows():
        school_coords = (school_row['학교 위도'], school_row['학교 경도'])
        distance = geodesic(house_coords, school_coords).km
        if distance <= radius_km:
            count += 1
    return count
```

주택에서 일정 거리 안에 있는 학교 수를 계산하는 함수 선언

03 데이터 전처리

```
▶ # 반경 내에 있는 학교의 수 계산
for radius in [1, 3, 5, 10]:
    column_name = f'{radius}KM 내 학교 수'
    house_coordinate_data[column_name] = house_coordinate_data[['주택 위도', '주택 경도']].apply(
        lambda coords: count_schools_within_radius(coords, school_data, radius), axis=1
    )
```

for문으로 1, 3, 5, 10km 내에 있는 학교 수를 각각 계산

03 데이터 전처리

```
# 완성된 데이터 파일 경로 설정
data_path = '/content/data/house_coordinates.xlsx'

# Excel 파일을 DataFrame으로 읽어오기
df = pd.read_excel(data_path)

# Excel 파일을 csv 파일로 변환하고 다시 읽어오기
df.to_csv('test.csv')
df = pd.read_csv('test.csv')

# csv 파일 일부 확인
df.head()
df.tail()
```

	Unnamed: 0	주택 위도	주택 경도	단위 면적당 평균가격	1KM 내 학교 수	3KM 내 학교 수	5KM 내 학교 수	10KM 내 학교 수
9984	9984	37.101827	129.364137	6.057880e+05	1	1	1	5
9985	9985	37.108233	129.369956	1.338455e+06	1	1	1	5
9986	9986	37.062045	129.417569	4.942241e+05	3	3	3	9
9987	9987	37.046455	129.411013	4.654238e+05	0	3	3	9
9988	9988	36.685998	129.438072	6.651070e+05	2	3	6	10

완성된 데이터 파일 저장하고 읽어온 후 일부 정보 확인

03 데이터 전처리

```
[ ] #데이터 전처리(결측치 없으므로 패스)
    df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9989 entries, 0 to 9988
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Unnamed: 0            9989 non-null   int64   
1   주택 위도              9989 non-null   float64  
2   주택 경도              9989 non-null   float64  
3   단위 면적당 평균가격    9989 non-null   float64  
4   1KM 내 학교 수         9989 non-null   int64   
5   3KM 내 학교 수         9989 non-null   int64   
6   5KM 내 학교 수         9989 non-null   int64   
7   10KM 내 학교 수        9989 non-null   int64   
dtypes: float64(3), int64(5)
memory usage: 624.4 KB
```

결측치를 확인했지만 없으므로 넘어감

04 데이터 분석

```
[ ] #산점도 분석
data_for_analysis = df[['단위 면적당 평균가격', '1KM 내 학교 수', '3KM 내 학교 수', '5KM 내 학교 수', '10KM 내 학교 수']]

plt.figure(figsize=(20, 5))

for i, distance_range in enumerate(['1KM', '3KM', '5KM', '10KM']):
    plt.subplot(1, 4, i+1)
    plt.scatter(data_for_analysis[f'{distance_range} 내 학교 수'], data_for_analysis['단위 면적당 평균가격'])
    plt.title(f'{distance_range} 내 학교 수와 주택 가격의 관계')
    plt.xlabel(f'{distance_range} 내 학교 수')
    plt.ylabel('단위 면적당 평균가격')

plt.tight_layout()
print("산점도 분석이 완료되었습니다.")
plt.show()
```

거리별 산점도 분석

04 데이터 분석

```
[ ] # 상관 관계 분석
correlation_matrix = data_for_analysis.corr()

# 시각화
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=.5)
plt.title('단위 면적당 평균가격과 범위 내 학교 수의 상관 관계')

print("상관 관계 분석이 완료되었습니다.")
plt.show()
```

상관관계 분석

04 데이터 분석

```
[31] # 독립변수(X)와 종속변수(y) 설정
      school_1km = df[["1KM 내 학교 수"]]
      school_3km = df[["3KM 내 학교 수"]]
      school_5km = df[["5KM 내 학교 수"]]
      school_10km = df[["10KM 내 학교 수"]]

      X = np.column_stack((school_1km, school_3km, school_5km, school_10km))
      y = df['단위 면적당 평균가격']

      # 데이터를 학습용과 테스트용으로 분할
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

선형회귀분석을 위한 독립변수, 종속변수 설정

04 데이터 분석

```
▶ # 4개의 독립 변수에 대해 각각 모델을 학습하고 예측
for i, km in enumerate([1, 3, 5, 10]):
    # i번째 독립 변수만 선택
    X_train_single = X_train[:, i].reshape(-1, 1)
    X_test_single = X_test[:, i].reshape(-1, 1)

    # 선형 회귀 모델 초기화
    model = LinearRegression()

    # 모델 학습
    model.fit(X_train_single, y_train)

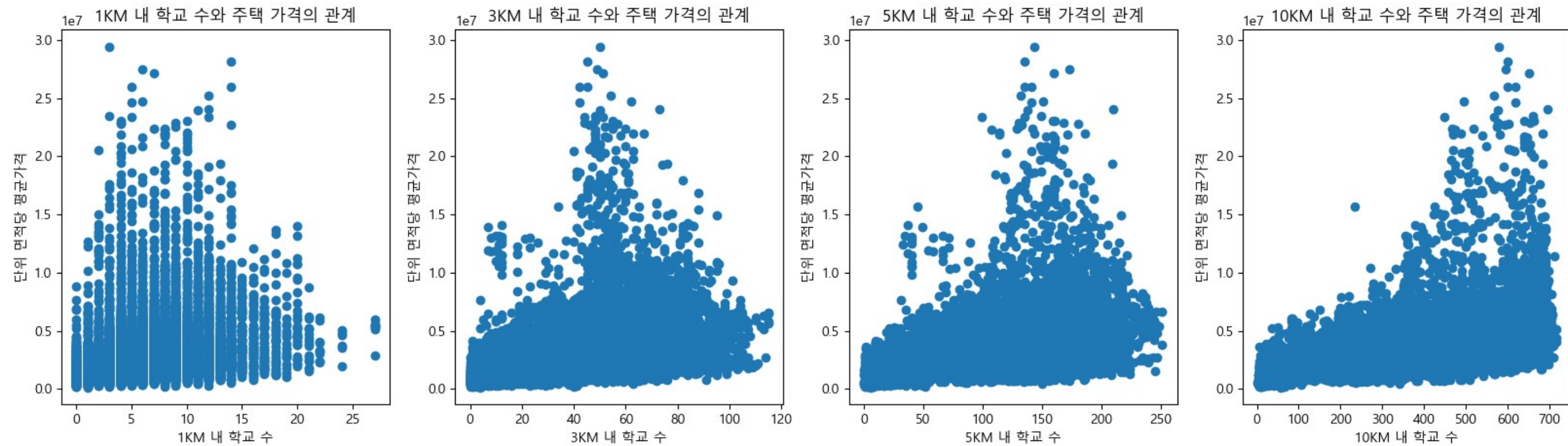
    # 학습된 모델을 사용하여 테스트 데이터에 대한 예측 수행
    y_pred_single = model.predict(X_test_single)

    # 모델 평가 (예측값과 실제값 비교)
    mse_single = mean_squared_error(y_test, y_pred_single)
    print(f"MSE for {km}KM school distance: {mse_single}")

    # 산점도와 회귀 직선 시각화
    plt.scatter(X_test_single, y_test, label='Actual Data')
    plt.plot(X_test_single, y_pred_single, color='red', linewidth=2, label='Regression Line')
    plt.xlabel(f'Schools within {km}KM')
    plt.ylabel('House Price')
    plt.title(f'Regression Analysis for Schools within {km}KM')
    plt.legend()
    plt.show()
```

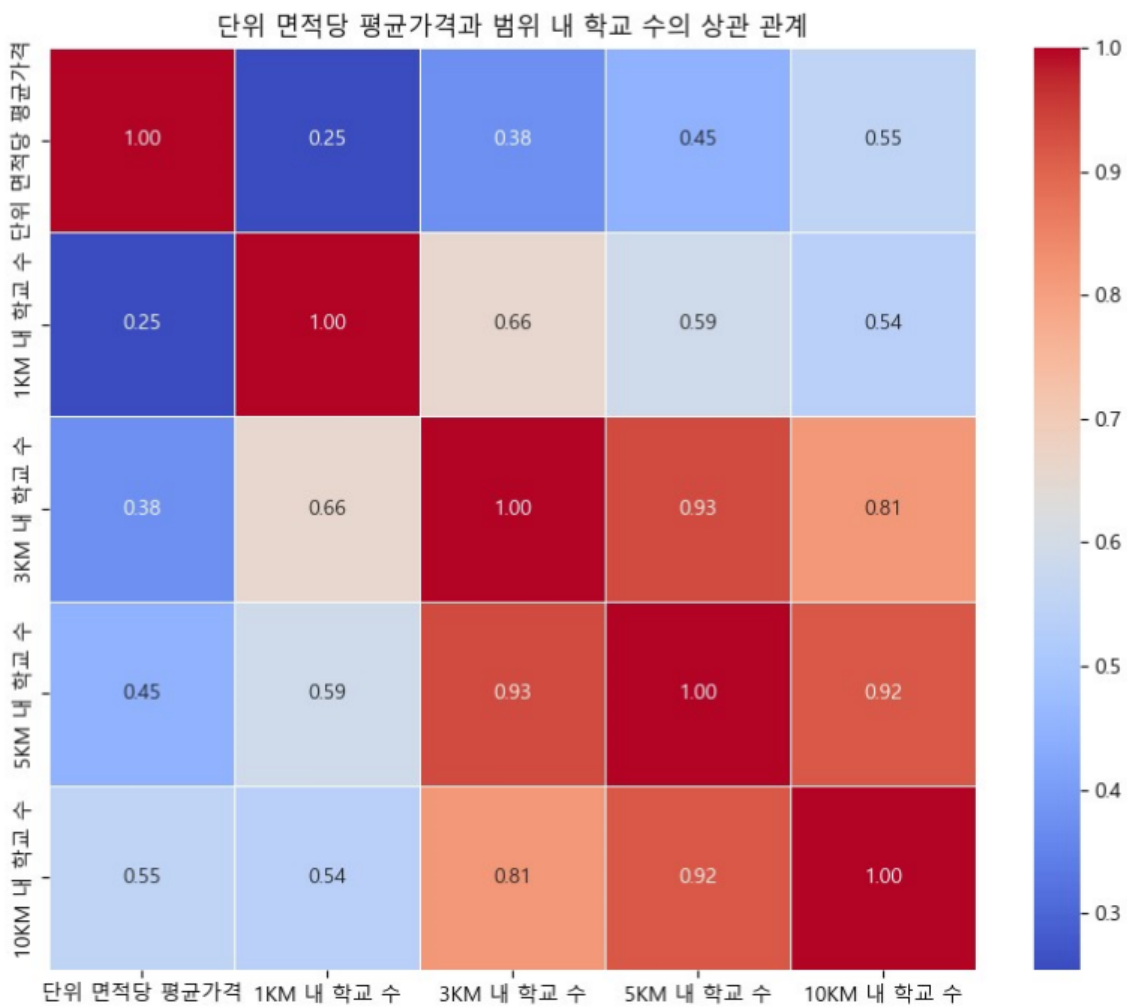
선형회귀분석 모델을 학습하고 모델 평가

05 결과 및 해석



산점도 분석 결과 그래프

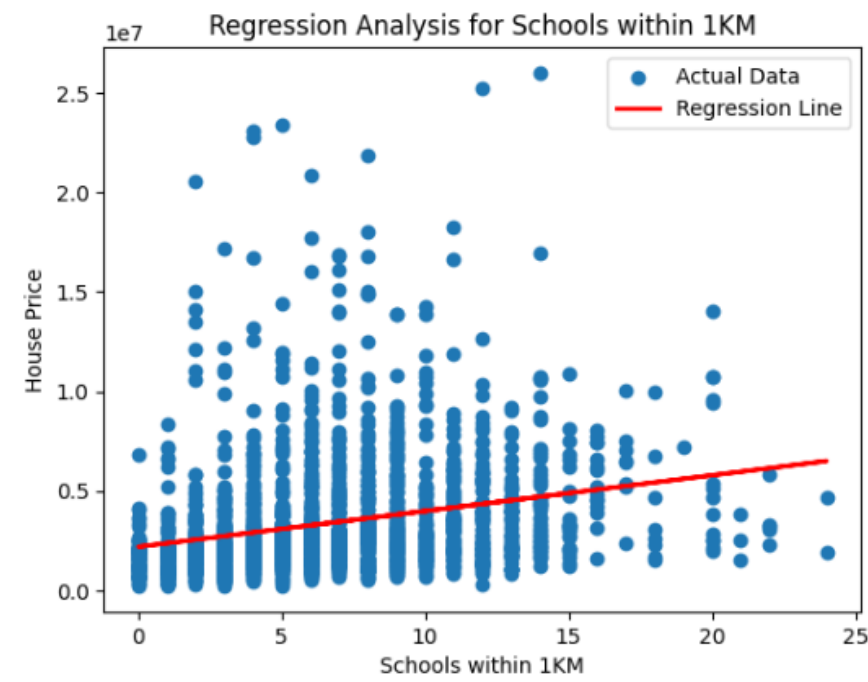
05 결과 및 해석



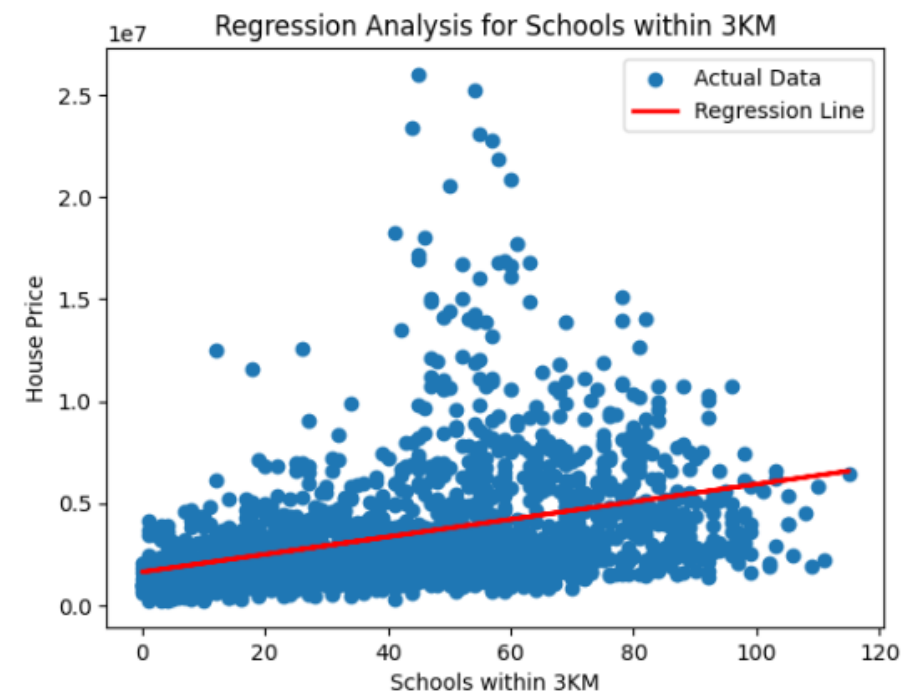
상관관계 분석 히트맵 - 양의 상관관계

05 결과 및 해석

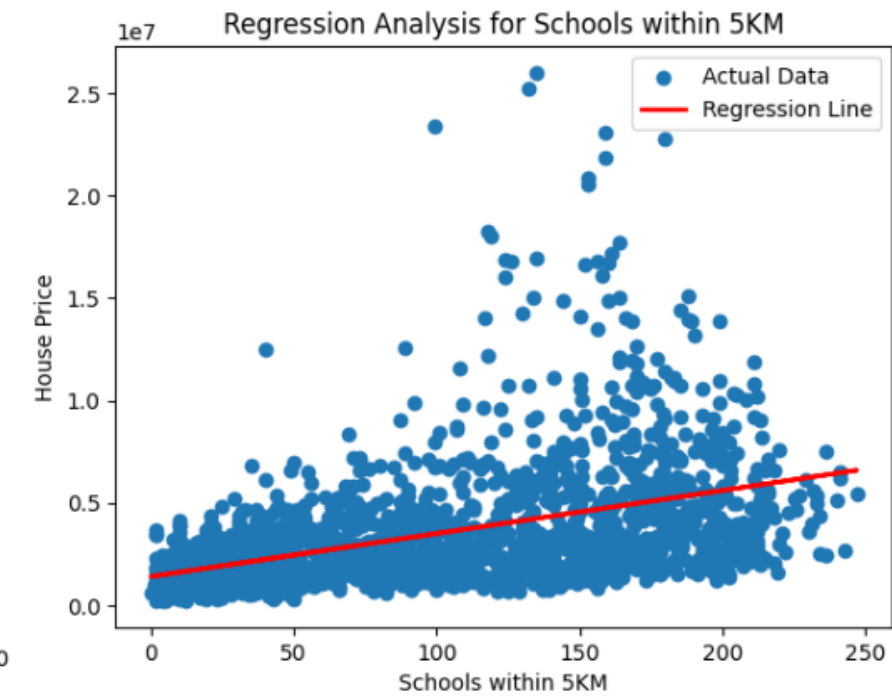
MSE for 1KM school distance: 7970545671986.056



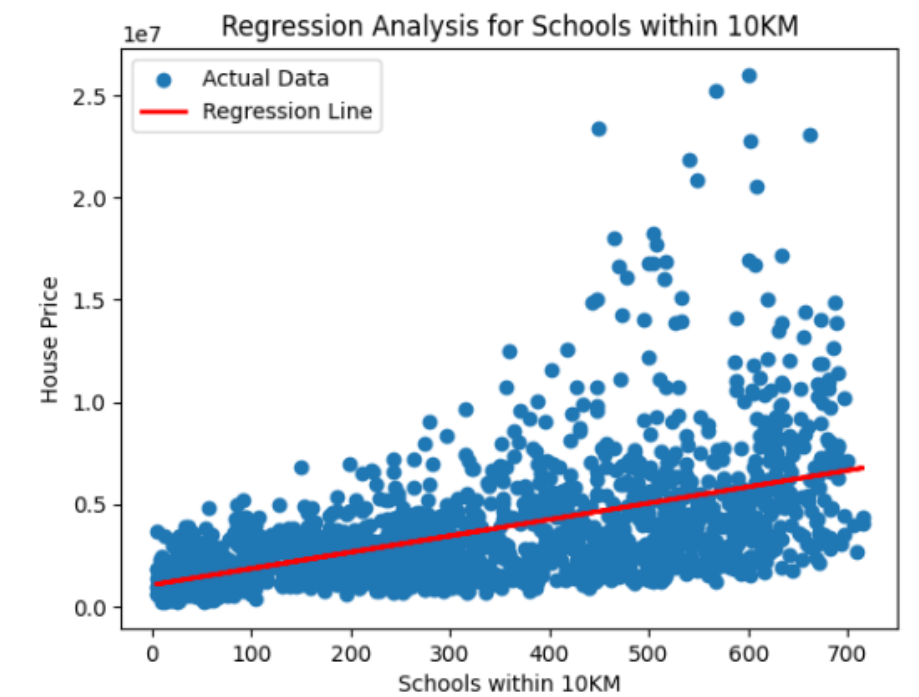
MSE for 3KM school distance: 7286927352315.001



MSE for 5KM school distance: 6814650448126.656



MSE for 10KM school distance: 6034129840763.127

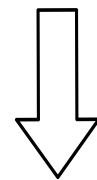


선형회귀분석 결과

05 결과 및 해석

**주택 가격이 거리 내 학교 수의 영향을 받지만
그 영향이 크지 않은 것으로 보임.**

**학교별 학업 성취도, 교육 품질, 지역 인프라 등
다른 데이터를 추가**



더 정확한 분석 결과

부동산 시장 예측 및 투자 전략 개발

도시 계획 수립 시 적절한 학교 시설의 배치

마케팅 및 광고 전략 수립

부동산 관련 정책 수립

감사합니다.