

Python 기반 P-Median 알고리즘을 활용한 은평구 공공자전거 대여소 입지 선정:

도시 공학적 데이터 분석 및 교통 최적화 접근

문창민 · 이준규 · 정은준 · 홍찬교

하나고등학교

Optimal Location Selection for Public Bike Rental Stations in Eunpyeong-gu Using  
Python-Based P-Median algorithm: Urban Engineering Data Analysis and Transport  
Optimization Approach

Chang-Min Moon · Joon-Kyu Lee · Eun-Jun Jung · Chan-Gyo Hong

Hana Academy Seoul

**Abstract** This study proposes a methodology to enhance the efficiency of public bicycle stations in Eunpyeong-gu by applying the P-Median algorithm to optimize their placement. Data were collected through a survey of Eunpyeong-gu residents and analyzed to assess the correlation between public bicycle stations and surrounding facilities (e.g., schools, apartments, supermarkets). Using Python-based data analysis, distances and weights between facilities and stations were calculated to design an optimal layout that minimizes costs and maximizes accessibility. The findings address the imbalance in station distribution within Eunpyeong-gu, identifying new locations that could improve bicycle usage. This research contributes to advancing sustainable transportation and urban infrastructure.

**Keywords:** Public Bicycle System, P-Median Algorithm, Urban Mobility, Bike Station Optimization, Eunpyeong-gu

**초록:** 본 연구는 은평구 공공자전거 대여소의 효율성을 높이고자 P-Median 알고리즘을 활용하여 최적의 대여소 위치를 재배치하는 방법을 제안한다. 은평구 주민을 대상으로 한 설문조사를 통해 데이터가 수집되었으며, 이를 바탕으로 공공 자전거 대여소와 주변 주요 시설(학교, 아파트, 마트 등) 간의 상관관계를 분석하였다. Python 기반 데이터 분석을 통해 시설물과 대여소 간의 거리 및 가중치를 계산하였으며, 이를 통해 비용 절감과 접근성을 극대화하는 최적 입지 구조를 설계하였다. 연구 결과, 은평구 내 공공 자전거 대여소의 분포 불균형을 해소하고 자전거 이용률을 향상시킬 수 있는 새로운 위치가 도출되었다. 이는 도시 교통 인프라 개선 및 지속 가능한 이동 수단 활성화에 기여할 것이다.

**키워드:** 공공 자전거 시스템, P-Median 알고리즘, 도시 이동성, 자전거 대여소 최적화, 은평구

## 1. 서론

도시의 지속 가능한 발전과 환경 친화적인 교통수단의 보급은 오늘날의 중요한 과제 중 하나이다. 특히 대기오염 배출 감소와 에너지 절약을 목표로 한 공공 자전거 이용 서비스는 전 세계적으로 주목받는 해결책으로 자리 잡았다. 그러나 특히 은평구의 공공 자전거 대여소는 비효율적인 분포와 이용량의 지역적 불균형 등으로 인해 사용량 감소 등의 문제를 발생한다. 실제로 서울시 공공 자전거 ‘따릉이’는 최초 대여소 배치 당시 대여소 별 수요 예측 실패로 하루에 한 명도 이용하지 않는 대여소마저 존재하였다. 이와 같은 문제를 해결하기 위해 은평구의 자전거 대여소의 위치를 최적화하고 수요를 기반으로 예측하는 과정을 구축하는 연구가 요구된다.

은평구는 지리적으로 삼면이 산으로 둘러싸여 있고, 산책길이 조성되어 있는 근린공원이 상당히 많이 있으며, 불광천과 진관천을 따라 많은 구민들이 이용하는 산책로가 조성되어있다. 이러한 산책로는 자전거 이용 활성화를 위해 굉장히 유리한 조건이다. 은평구 주민들 역시 자전거를 이용하고자 했으며, 이는 정책적 부분에서도 드러난다. 은평구청에서 2021년에 발표된 자전거이용 활성화를 위한 5개년 종합계획을 위해 실시한 설문조사에서 은평구민들은 “자전거 인프라 구축”을 가장 크게 요구하였다.

본 연구는 도시 내 공공 자전거 대여소의 효율성을 높이기 위한 통계적 접근을 시도한다. 특히 은평구 도심 내 주요 시설(주택, 공원, 학원, 학교, 마트 등)의 경도와 위도를 Python을 통해 구현한다. 구현된 자료는 CSV 파일의 형식으로 저장한다. 이러한 자료를 통해 특정 유형의 시설과 대여소 이용률 사이

의 상관관계를 규명하고 이를 바탕으로 P-Median 알고리즘을 적용해 대여소의 최적 분포를 설계하는데 목표를 둔다.

결과적으로 본 연구는 공공 자전거 대여소의 효율적인 배치 구조를 고안하여 공공 자전거 대여소와 관련한 문제점을 해소하는 것을 목적으로 한다. 이후 더 나아가 도시의 교통 인프라를 효율적으로 개선할 수 있도록 한다. 또한 본 연구는 공공 자전거 서비스가 지닌 사회적인 가치를 증대시키고 지역 사회 문제 해결에 기여할 수 있는 학술적 연구 토대를 마련하고자 한다.

## 2. 이론적 배경

### 2.1. P - Median 알고리즘

P-Median 알고리즘이란 휴리스틱 알고리즘의 일종으로 입지 선정 문제 중 공공시설 등을 설치할 수 있는 후보입지가 주어져 있을 때, 최소의 비용으로 모든 소비자의 수요를 충족시킬 수 있는  $p$ 개 이하의 시설 설치 입지를 결정하는 문제인 P-Median 문제에 적용되는 알고리즘이다. 해당 알고리즘은 시설물(공공 자전거 대여소)을 이용하는 수요자(공공 자전거 이용자)를 위해 시설물의 위치에서 평균거리 또는 평균 통행시간, 평균 통행비용이 최소화되도록 위치를 선정하는 알고리즘인 까닭에 P-Median 알고리즘을 적용하는 것이 적합하다고 판단하였다. 알고리즘의 기본 모형은 이렇다.

*Inputs:*

$h_i$  = 수요지  $i$ 의 수요량

$d_{ij}$  = 수요지  $i$ 와 시설물의 입지점  $j$ 의 거리

$p$  = 시설물의 수

*Decision variables:*

$x_j$  = 만약 노드  $j$ 에 시설물이 설치되면 1, 그렇지 않으면 0으로 표현된다.

$y_{ij}$  = 만약 노드  $j$ 에 시설물이 노드  $i$ 의 총 수요를 충족시키면 1, 그렇지 않으면 0으로 표현된다.

$$\text{Subject to Min } \sum_i \sum_j h_i d_{ij} y_{ij} \quad (1-1)$$

$$\sum_j y_{ij} = 1 \text{ (for all } i) \quad (1-2)$$

$$\sum_j x_j = p \quad (1-3)$$

$$y_{ij} \leq x_j \text{ (for all } i, j) \quad (1-4)$$

$$y_{ij} \in 0,1 \text{ (for all } i, j) \quad (1-5)$$

$$y_j \in 0,1 \text{ (for all } j) \quad (1-6)$$

(1-1)의 식에 나타나는 목적 함수 수식은 시설물과 수요자 사이의 전체 통행 거리를 나타낸다. 여기서 수요지와 i와 시설물의 입지점 j 사이의 거리  $d_{ij}$ 는 경로상에서 여러 지점 사이의 최단경로를 표현한다. (1-2) 식의 제약조건은 각각의 수요발생지점은 항상 하나의 시설물에게 서비스를 받고 있음을 의미하고 중복 서비스나 서비스 부재지역이 존재하지 않음을 나타낸다. (1-3) 식의 제약조건은 자신의 지역의 시설물에 의해 서비스를 받는 수요 지의 수는 시설물의 수와 같다는 것을 나타낸다. (1-4) 식의 제약조건은 만약  $x_j = 0$  이면 j 지역에 시설물이 존재하지 않으며 i 지역 이용자는 j 지역의 시설에 할당되지 않아  $y_{ij} = 0$  이 되고, 반대의 경우  $y_{ij}$ 는 0 혹은 1의 값을 가지게 되는 경우를 표현한다. 식 (1-5)과 식 (1-6)은 결정 변수  $y_{ij}$ 와  $x_j$ 가 0 혹은 1의 값을 가지는 것을 가능하게 하는 제약 조건들이다.

### 3. 연구 방법 및 절차

#### 3.1. 연구 방법

본 연구의 방법은 유사한 선행 연구인 ‘휴리스틱 P-Median 알고리즘을 이용한 자전거주차장 최적 입지선정’을 참고하여 진행했다.

전체적인 순서는 다음과 같다.

1. 한 도시 내의 특정 종류(주택, 공원, 학원, 학교, 마트)의 모든 시설의 경도 위도를 포함한 csv 파일을 찾는다
2. 각 시설 종류 클래스의 각 객체의 경도 위도를 모두 좌표로 변환하고 csv파일 정보를 가공한다

3. 한 도시 내의 모든 공용자전거 대여소의 경도와 위도, 대여량을 포함한 csv파일을 찾고 각 대여소의 위도와 경도를 좌표로 변환한다
4. 모든 클래스의 좌표를 활용해 각 대여소에 대해 특정 거리 내에 어떤 종류의 시설이 몇 개 있는지 파악한 후 해당 데이터를 통해 따릉이의 수요와 주변 시설의 연관성을 회귀 분석을 통해 분석한다.
5. 수요 분석으로 도출한 값을 기반으로 각 시설 별 가중치를 부여하고 p-median 알고리즘을 적용해 공용 자전거 대여소를 재배치한다.
6. 재배치한 대여소의 위치를 지도로 시각화하고 이를 분석한다.

위 선행 연구에서는 대중교통 정거장을 이용할 때 보행 접근거리의 임계치(약 400m-500m)를 넘어서게 되면 대중교통 이용률이 급격히 감소한다는 연구결과를 바탕으로 수요 분석 범위를 설정하였지만, 본 연구에서는 자전거는 기타 대중교통 정거장과 달리 어느 정거장을 이용하더라도 목적지에 도달할 수 있다는 점과 자전거 대여소 사이의 거리가 보통 해당 거리보다 짧다는 점을 고려해 특정 대여소와 사용자의 거리가 대여소와 대여소의 평균 거리보다 멀 경우 사용자가 다른 대여소를 이용할 것이라 판단하고 이를 기반으로 수요 분석을 진행했다. 이를 대여소 위치 선정에 중점적으로 고려했고, 시험 네트워크는 25m단위로 격자망 네트워크를 구성했다. 이를 바탕으로 기본 도로망을 구축하고 기존 데이터를 활용하여 수요를 분석한 후, 수요지를 선정해 기존 대여소 수와 동일한 98개의 대여소를 배치하는 연구를 진행했다. 여기서 고려한 시설은 설문에서 은평구민들이 자전거를 이용하여 가장 많이 오간다고 답한 공원, 학교, 학원, 아파트, 지하철역이 있다.

### 3.2. 연구 절차

본 연구에서는 가장 효율적인 은평구의 대여소 위치를 선정하기 위해 P-median 알고리즘을 사용하고자 한다. P-median 알고리즘을 실행하기 전에 알아야 할 정보들이 있다. 우선 은평구 주민들의 따릉이 사용에 영향을 주는 요인들을 파악해야 하며 p-median에서 클러스터링, 즉 대여소들이 한 지점으로 몰리는 현상을 방지하기 위해 따릉이 별 최소 거리를 결정해야한다. 우선 주민들의 따릉이 사용

에 영향을 주는 요인들을 파악하기 위해 은평구 주민 대상의 설문조사와 ‘서울시 따릉이 자전거 이용 현황’ 데이터를 참고하였다. 그 결과 마트, 학원, 공원, 지하철역, 학교, 아파트가 따릉이 이용에 영향을 미치는 요소로 선정하였다.

클러스터링을 방지하기 위한 최소 거리를 설정할 때에는 서울시의 따릉이 배치도를 참고 했다. 이 데이터는 서울시 모든 따릉이의 위도/경도가 나와있는 자료이다. 이를 이용해 각 대여소와 다른 대여소 간의 거리를 한 번에 계산한 뒤, 각 대여소별로 가장 가까운 대여소와의 최소 거리를 추출한 후 모든 대여소에서 추출된 최소 거리의 평균값을 계산하였다. 이때 자신과의 거리를 제외하기 위해 대각선 값을 무한대로 설정하였으며 대여소 간 거리는, 지구 곡률을 고려하기 위해 Haversine 공식을 사용하여 계산하였다. 이 공식은 위도와 경도를 라디안 단위로 변환한 뒤 삼각 함수 연산을 통해 두 지점 간의 실제 거리를 도출하는 역할을 한다. 일반적인 Haversine 공식은 다음과 같이 표현된다.

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

다음으로 P-Median 알고리즘을 이용하기 위해서 각 시설물들과 대여소와의 가중치를 시설물의 타입을 기준으로 각각 도출하는 과정이 필요 한다. 이를 위해 각 대여소와 각 시설물들의 거리를 계산하고, 이를 바탕으로 유효한 범위 안에 있는 데이터들만 유효한 값으로 간주하였다. 유효한 값을 카운트한 횟수가 해당 대여소에 나타나는 것을 해당 대여소의 마트에 대한 계수라고 정의하였다. 예시로, 마트가 해당 대여소의 범위 안에 3 지점이 있으면, 마트에 대한 해당 대여소의 계수는 3이다. 이렇게 계수를 부여하는 과정을 마트뿐만 아니라 모든 종류의 시설물에 진행하였다. 이렇게 찾은 계수들은 연립방정식을 통해 가중치를 찾는 것에 사용될 것이다. 이를 연립방정식으로 표현하게 되면, 마트의 가중치가 m이고 공원의 가중치가 n이라 할 때, (특정 대여소의 범위 안에 있는 마트의 수) x m + (특정 대여소의 범위 안에 있는 공원의 수) x n = (특정 대여소의 대여건수) + (특정 대여소의 반납건수) 라는 연립방정식을 구성한다. 이는 마트와 공원에 대해서만 적용한 것으로 이를 마트, 공원, 학교, 학원, 지하철역, 아파트에 대해 모두 적용하여 6개의 항을 가지는 선형연립방정식을 구현하였다.

또한 유효 범위를 구현하기 위해선 대여소와 시설물 간의 거리를 모두 안다는 것을 전제조건으로 해야한다. 대여소의 위치를 경도와 위도로 포함한 csv의 모든 대여소와 마트 csv의 모든 마트들을 각각 모두 한 번씩 대응되게 하여 거리를 전부 도출하였다. 은평구 주민들을 대상으로 설문조사를 하여 얻은 결과와 역 근처에 자전거 보관소가 많이 위치해 있다는 자료를 바탕으로, 은평구 주민들을 대상으로 설문조사를 하여 얻은 결과와 역 근처에 자전거 보관소가 많이 위치해 있다는 자료를 바탕으로, 마트뿐만 아니라 공원, 학원, 아파트, 지하철역, 학교에도 이를 전부 적용하여 거리를 구하였다. 대여소와 시설물들의 좌표는 위도와 경도, 즉 정확한 m의 단위를 가지는 거리가 아니라 각도로 표현되기 때문에, 이를 하버사인 공식을 통해서 구면거리를 근사적으로 접근하였다. 이는 colab에서 geodesic이라는 내장된 함수가 있기 때문에 geodesic을 이용하여 거리를 계산하였다.

이렇게 규정한 거리를 바탕으로 가중치를 구할 때는, 기존의 사람들의 수요를 중심으로 분석해야 하기에 기존에 있는 대여소의 위치를 기준으로 구성하였다. 또한, 유한한 수치로 규정하는 범위를 구하기 위해서 서울시 따릉이가 평균적으로 190.32m의 간격을 가지기 때문에 영역을 절반으로 나누는 약 95m를 초기 기준으로 잡고, 이 초기 기준으로 부터 지하철 역의 크기 등 특정 시설물의 자료의 특성에 맞게 추가적인 수치를 더해서 유한한 범위를 규정하였다. 이를 마트, 공원, 학원, 아파트, 지하철역, 학교에 적용하였다. 이러한 학교에 대해서 6개의 가중치를 미지수로 가지는 선형 연립방정식을 얻으면, 대여소의 개수인 대략 100개의 선형 연립방정식을 얻게 된다. 이를 바탕으로 가중치의 값을 도출하기 위해서 공분산을 이용하였다. 100개의 연립방정식을 만족시키는 해는 실질적으로 존재하지 않기에, 이를 최대한으로 상관관계를 도출하기 위해서 공분산을 사용하였고, csv 파일을 통해서 구현할 때는 corr이라는 함수를 사용하여 가중치를 도출하였다.

```

import pandas as pd
from geopy.distance import geodesic

bike_station_df7 = pd.read_csv('/content/merged_data.csv')
mart_df7 = pd.read_csv('/content/drive/SharedDrives/지역연계/은평csv/은평 마트new2.csv')
park_df7 = pd.read_csv('/content/drive/SharedDrives/지역연계/은평csv/은평 공원.csv')
academy_df7 = pd.read_csv('/content/drive/SharedDrives/지역연계/은평csv/은평 학원.csv')

def count_facilities_within_radius(station_coords, facility_coords, radius):
    """
    특정 반경 내의 시설 개수를 계산하는 함수
    """
    count = 0
    for facility in facility_coords:
        if geodesic(station_coords, facility).meters <= radius:
            count += 1
    return count

bike_station_coords7 = bike_station_df7[['위도', '경도']].values

mart_coords7 = mart_df7[['위도', '경도']].values
park_coords7 = park_df7[['위도', '경도']].values
academy_coords7 = academy_df7[['위도', '경도']].values

density_data = []

mart_radius = 155
park_radius = 95
academy_radius = 115

```

```

for index, station in bike_station_df7.iterrows():
    station_coords = (station['위도'], station['경도'])

    mart_density = count_facilities_within_radius(station_coords, mart_coords7, radius=mart_radius)
    park_density = count_facilities_within_radius(station_coords, park_coords7, radius=park_radius)
    academy_density = count_facilities_within_radius(station_coords, academy_coords7, radius=academy_radius)

    total_usage = station['대여건수'] + station['반납건수']

    density_data.append([
        '대여소명': station['대여소명'],
        '위도': station['위도'],
        '경도': station['경도'],
        '총거래량': total_usage,
        '마트_밀도': mart_density,
        '공원_밀도': park_density,
        '학원_밀도': academy_density,
    ])

result_df7 = pd.DataFrame(density_data)

correlation_matrix = result_df7[['총거래량', '마트_밀도', '공원_밀도', '학원_밀도']].corr()

print("상관계수 분석 결과:")
print(correlation_matrix)

result_df7.to_csv('bike_station_density_with_total_usage.csv', index=False, encoding='utf-8')

```

상관계수 분석 결과:

	총거래량	마트_밀도	공원_밀도	학원_밀도
총거래량	1.000000	0.221395	0.035704	0.094878
마트_밀도	0.221395	1.000000	-0.015706	0.109879
공원_밀도	0.035704	-0.015706	1.000000	-0.050550
학원_밀도	0.094878	0.109879	-0.050550	1.000000

상관계수 분석 결과:

	총거래량	학원_밀도	아파트_밀도	지하철역_밀도
총거래량	1.000000	0.094878	0.070436	0.275411
학원_밀도	0.094878	1.000000	-0.167916	-0.040155
아파트_밀도	0.070436	-0.167916	1.000000	-0.036309
지하철역_밀도	0.275411	-0.040155	-0.036309	1.000000

상관계수 분석 결과:

	총거래량	학원_밀도	학교_밀도
총거래량	1.000000	0.094878	0.059134
학원_밀도	0.094878	1.000000	0.248757
학교_밀도	0.059134	0.248757	1.000000



위와 같이 p-median을 위한 준비과정을 마쳤다. 이제 P-Median 알고리즘을 적용하여 수요 지점과 가장 가까운 후보 위치 사이의 거리 비용을 최소화할 것이다. 이 과정에서 수요 지점별 중요도를 고려하기 위해 가중치를 반영하여 최적의 위치를 도출하는 것을 목표로 한다.

우선 후보 위치 중 무작위로 ‘설치가 가능한 대여소 개수(n)’를 선택하여 초기 해로써 설정한다. 이러한 초기 해는 가중치를 고려한 총 거리 비용, 다시 말해 떨어진 거리의 합을 계산하기 위한 시발점 역할을 위함이다. 이러한 거리 비용은 수요 지점과 가장 가까운 후보 위치 사이의 거리와 해당 지점의 가중치를 곱한 값으로 산출되며, 결과적으로 모든 수요 지점에 대해 이를 합산하여 총 거리 비용을 도출한다. 이후 임의로 선택된 후보 위치 중 거리 비용이 가장 큰 하나의 점을 다른 임의의 후보 위치로 교체하며 총 거리 비용을 다시 계산하는 과정을 반복한다. 요컨대 비용이 감소하는 경우 교체된 위치를 유지하고, 다른 후보 위치를 대상으로 다시 계산한다. 이러한 과정은 총 거리 비용이 더 이상 감소하지 않을 때까지 진행해야하며 그 결과 최적의 후보 위치 집합이 결정된다. 마지막으로 도출된 n개의 최적 후보 위치를 지도에 시각화 하기 위해 위도 및 경도 좌표 형태로 출력했다. 시각화에 대해선 연구 결과에 자세히 기술하겠다.

## 4. 연구 결과

상기의 절차를 통해 진행된 연구 결과는 다음과 같다.

### 4.1 P-Median 적용 이전 절차

먼저 은평구의 구조를 불러오기 위해 Open Street Map을 통해 python코드로 은평구를 구현하였다. 이후 0.01km 간격의 그리드로 처리했다. 해당 코드는 다음과 같다.

```
def generate_grid_within_region(region_name, step_km=0.1):

    region = ox.geocode_to_gdf(region_name)
    boundary = region.geometry.values[0]

    minx, miny, maxx, maxy = boundary.bounds
    km_to_degree = 1 / 111
    step = step_km * km_to_degree

    latitudes = np.arange(miny, maxy, step)
    longitudes = np.arange(minx, maxx, step)
    grid = np.array([[lat, lon] for lat in latitudes for lon in longitudes])

    inside_points = []
    for point in grid:
        shapely_point = Point(point[1], point[0])
        if boundary.contains(shapely_point):
            inside_points.append(point)

    return np.array(inside_points)
```

본 코드에서는 ox.geocode\_to\_gdf 함수를 이용하여 은평구의 경계 좌표를 불러왔다. 이후 위도와 경도를 앞서 말한 바와 같이 0.01km(약 100m) 간격으로 규정하여 그리드를 생성했다. 이후 shapely\_point를 통해 은평구 경계 내에만 속하는 점을 선택했다. 또한 calculate\_weighted\_distance\_matrix 를 통해 각 수요 지점과 후보지 사이의 유클리드 거리 행렬을 계산하고 수요 가중치를 반영했다. 여기서 demand\_points, candidate\_locations, demand\_weights 에서 각각 수요지점의 위도/경도 좌표, 후보 위치의 위도/경도 좌표, 수요지점별 가중치를 입력하여 가중치를 적용한 수요지점과 후보지의 거리를 출력받았다.

## 4.2. P-Median 알고리즘 적용

이후 P-Median 알고리즘을 활용해 해당 거리를 분석했다. 앞서 말한 바와 같이 무작위로 초기 후보지를 설정하고, 수요 지점을 연결해 거리 비용을 계산했다. 후보지를 교체하며 비용을 최소화하는 과정을 반복해 최적의 후보 위치를 도출하고 최종 결과는 대여소 설치에 적합한 위치를 위도와 경도로 제공받았다.

출력된 결과의 일부는 다음과 같다.

총 실행 시간: 2213.02초

최적 대여소 위치 (위도, 경도):

위도: 37.634559558557704, 경도: 126.95773407567458

위도: 37.6129379369364, 경도: 126.94129263423338

위도: 37.6201451441435, 경도: 126.90638272432398

위도: 37.639289288287365, 경도: 126.94219353513427

위도: 37.63185685585504, 경도: 126.9579593008998

(이하 생략)

### 4.3 데이터 가공 및 시각화

먼저 Folium과 OSMnx를 활용하여 지정된 지역의 후보 위치를 지도에 삽입하는 과정을 진행했다. 해당 코드는 다음과 같다.

```
import folium
import osmnx as ox

def visualize_candidate_locations(candidate_locations, region_name):

    region = ox.geocode_to_gdf(region_name)
    center_lat, center_lon = region.geometry.centroid.y.values[0], region.geometry.centroid.x.values[0]

    map = folium.Map(location=[center_lat, center_lon], zoom_start=13)

    for loc in candidate_locations:
        folium.CircleMarker(location=loc.tolist(), radius=1, color='yellow').add_to(map)

    return map
```

OSMnx의 geocode\_to\_gdf를 통해 입력된 지역명(본 연구의 경우 ‘은평구’)의 중심 좌표를 계산하고 이를 바탕으로 Folium 지도를 초기화 했다. 이후 후보 위치 배열의 각 좌표를 순회하며 지도에 원형 점으로 위치를 표시했다. 해당 표식은 후보 위치의 위도와 경도를 나타내며 지도 상에서 후보지의 분포와 밀집도를 직관적으로 확인할 수 있었다.

이후 해당 결과를 통계 분석과 이상값 처리를 수행해 시각적으로 나타냈다. 먼저 앞서 도출된 데이터를 입력해 산점도를 그려 데이터의 분포를 확인했다. 다음으로 산술 평균, 기하 평균, 표준편차, 분산의 주요 통계량으로 계산을 진행했다.

여기서 이상값은 IQR의 방식을 이용해 처리했다. 데이터의 1사분위수(Q1)와 Q3값을 계산한 뒤, 이를 바탕으로 IQR를 구했다. 이후 일반적인 데이터 범위로 간주되는  $[Q1 - 1.5/IQR, Q3 + 1.5/IQR]$  사이에

속하지 않는 값을 이상값으로 간주했다. 이후 이상값을 제거한 데이터를 별도로 저장하고 저장된 데이터의 산술 평균을 계산했다. 마지막으로 이상값이 제거된 데이터의 산점도를 그려 원래의 데이터와 비교했다. 이를 통해 이상값이 데이터 분석과 평균 값에 미치는 영향을 시각적으로 확인했다. 해당 코드와 실행 결과는 다음과 같다.

```
plt.scatter(range(len(data)), data)
plt.title('Scatter Plot of Data')
plt.xlabel('Index')
plt.ylabel('Value')
plt.show()

arithmetic_mean = np.mean(data)

geometric_mean = stats.gmean(data)

std_dev = np.std(data)
variance = np.var(data)

Q1 = np.percentile(data, 25)
Q3 = np.percentile(data, 75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

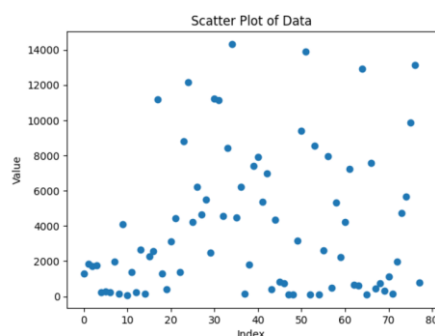
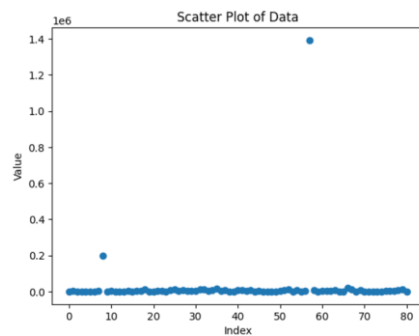
```
filtered_data = [x for x in data if lower_bound <= x <= upper_bound]

filtered_mean = np.mean(filtered_data)

print(f"산술평균 (Arithmetic Mean): {arithmetic_mean}")
print(f"기하평균 (Geometric Mean): {geometric_mean}")
print(f"표준편차 (Standard Deviation): {std_dev}")
print(f"분산 (Variance): {variance}")
print(f"이상값을 제거한 평균 (Filtered Mean): {filtered_mean}")

plt.scatter(range(len(filtered_data)), filtered_data)
plt.title('Scatter Plot of Data')
plt.xlabel('Index')
plt.ylabel('Value')
plt.show()
```

## <실행 결과>



마지막으로 얻게 된 결과를 지도에 시각화하여 실제 지전거 대여소의 위치와 비교를 진행했다. 먼저 Folium 라이브러리를 사용하여 위도와 경도 좌표들을 지도에 시각화했다. 해당 코드는 다음과 같다.

```
import folium

def visualize_points(latitudes, longitudes, zoom_start=13):
    center_lat, center_lon = latitudes[0], longitudes[0]
    map = folium.Map(location=[center_lat, center_lon], zoom_start=zoom_start)

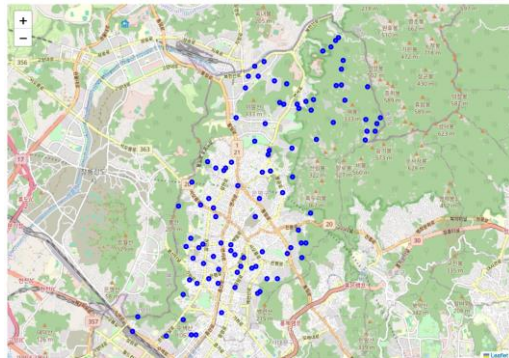
    for lat, lon in zip(latitudes, longitudes):
        folium.CircleMarker(location=[lat, lon], radius=3, color='blue', fill=True, fill_opacity=0.6).add_to(map)

    return map

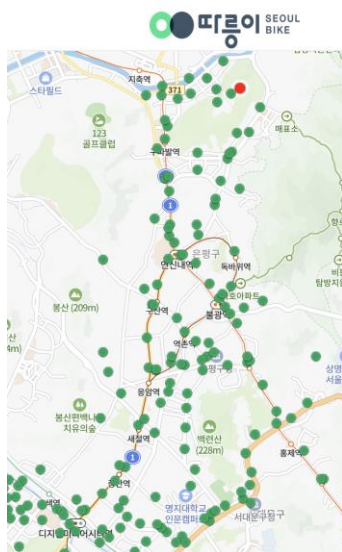
latitudes = [
    37.634559558557704, 37.6129379369364, 37.6201451441435, 37.639289288287365,
    37.63185685585504, 37.6273523513506, 37.63883883783692, 37.58455955855843,
```

본 코드에서 visualize\_points 함수는 위도와 경도의 리스트를 입력받아 지도를 생성했다. 먼저 지도

중심을 첫 번째 좌표 (latitudes[0], longitudes[0])으로 설정해 folium.Map 객체를 초기화했다. 이후 입력된 위도와 경도를 묶어 반복문을 통해 각각의 좌표에 folium.CircleMarker를 생성했다. 각 표시의 반지름 3의 파란색 원형으로 표시될 수 있도록 수정했다. 이때 fill=True와 fill\_opacity=0.6 속성으로 채워진 반투명 형식으로 삽입했다. 최종적으로 해당 표시가 삽입된 Folium 지도 객체를 반환하여 시각화했다. 해당 결과는 다음과 같다.



<출력된 시각화 결과>



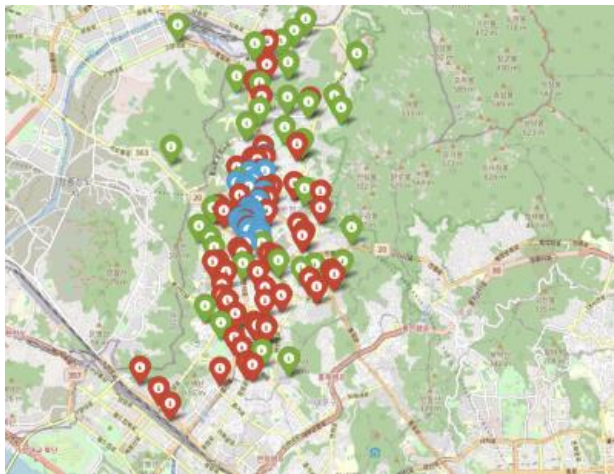
<실제 공공 자전거 따릉이 대여소 위치 분포>

결과로 도출된 최적의 대여소 위치는 은평구 다양한 지점에 위치해 있다는 것을 알 수 있다. 그러나 실제 대여소 위치와는 달리 일부 최적 대여소의 위치가 산림 지역인 북한산과 그 주변에 포함되어 있다는 것을 알 수 있다.

이와 같은 결과는 수요 지점과 후보지의 거리 계산 과정에서 은평구 경계 내부의 후보지가 고려되었기 때문에 발생했을 것으로 추정된다. 이로 인해 산림 지역처럼 상대적으로 인구 밀도가 낮고 자전거 대여소 수요가 낮은 경향의 장소도 분석에 포함되었다. 따라서 산림과 같은 지역은 대여소를 설치하기

에 부적합함에도 불구하고 P-Median 알고리즘은 단순히 거리와 가중치 데이터를 기반으로 계산하므로 해당 지점이 선택될 수 있다.

또한 수요 지점의 분포와 가중치 설정 또한 영향을 미쳤을 것으로 추정된다. 특정 수요 지점이 산림 근처에 위치하거나 해당 지점이 높은 가중치를 가질 때 P-Median 알고리즘은 수요 지점의 거리 비용을 최소화하기 위해 인근 지역을 후보지로 선택할 수 있다. 이는 산림 주변의 등산로 입구나 휴식 공간 근처에서 자전거 대여 수요가 존재할 가능성과 연결될 수 있다.



<학교, 아파트, 마트 등 주요 지점 위치>

마지막으로 본 연구에서 특정 종류(주택, 공원, 학원, 학교, 마트)의 모든 시설의 분포와 결과로 출려된 최적 대여소의 위치가 유사한 형태로 나와 있다는 것을 알 수 있다. 결과적으로 해당 지점들이 공공 자전거의 대여소 위치에 큰 영향을 미친다는 것을 시각적으로 알 수 있다.

## 5. 결론 및 논의

최종적으로 본 연구를 통해 p-median 알고리즘을 적용하여 은평구 공용자전거 대여소 배치를 최적화하였다. 국가기관에서 제공하는 csv파일들을 통해 따릉이 수요 분석을 위한 데이터를 수집하고 가공했으며, Haversin 공식과 Python의 geodesic 모듈을 통해 위치 데이터를 m단위로 변환하였다. 공간 분석과 유클리드 거리 행렬을 기반으로 주변 시설에 따라 달라지는 대여소의 수요를 분석하였다. 분석한 수요를 기반으로 가중치를 부여해 p-median 알고리즘을 기반으로 기존 대여소와 동일한 수의 대여소를 효율적으로 재배치한 후, Folium과 Osmnx를 활용해 최종 결과를 지도에 시각화하였다.

본 연구를 통해 P-Median 알고리즘의 유용성과 적용 가능성을 입증하였으며, 도시 교통 인프라 배치에서 p-median 알고리즘이 최적 입지 선정 문제의 실질적 도구로 활용될 수 있음을 보여준다. 하지만, 본 연구 결과를 바로 현실에 적용하기에 무리가 있다. 먼저 수요 분석 과정에서 채택한 시설 외의 다른 시설이나 아예 다른 요인이 수요에 영향을 미칠 가능성이 적지 않고, 또한 수요 분석 과정에서 각 시설의 크기를 고려하기는 하였으나 원본 데이터 값의 편차가 매우 커 연구에서 수요 분석 결과의 신빙성을 입증하기 어렵다. csv 파일 자체에 일반적으로 이해할 수 없는 몇 극단적인 데이터 값이 있는 등의 오류가 존재하기도 했다. 또한 p-median 알고리즘을 적용하는 것에 있어 실제로 대여소가 위치할 수 있는 도로만 입지점으로 선정하는 것이 이상적이나 데이터 제작에 한계가 있어 대여소가 산 중앙에 배치되거나 하는 등의 결함이 존재했다.

위와 같은 한계가 존재하지만, 본 연구를 통해 수요지에 따른 시설 입지 최적화 문제를 해결하는 능력을 기를 수 있었고, 이와 같이 공용 자전거 대여소를 특정 지역에 최적화하는 것은 적은 비용으로 자전거의 접근성을 최대화하고 실질적인 사용량을 크게 증가시킬 수 있다. 이는 지역 주민들의 요구를 충족함과 동시에 환경파괴를 예방하는 의미를 가지고, 지속가능한 발전과 지역 주민의 건강 및 활력 증진, 교통 개선이라는 다양한 긍정적 효과를 유도하여 궁극적으로 지역 발전에 중요한 역할을 수행할 수 있다.

## 6. 참고문헌

이태현 (2014, February 23), '전기자전거, 차세대 친환경 이동수단으로 자리매김',

<https://www.snilbo.co.kr/46521>

정대연 (2017, June 36), '서울 자전거 '따릉이' 대여소, 하루 1명도 이용않는 곳도',

<https://www.khan.co.kr/article/201706261710001>

정민구 (2021, August 10), '안전하고 쾌적한' 자전거 환경 만들기 집중한다',

<https://www.epnews.net/news/articleView.html?idxno=30364>

한국민족문화 대백과 사전, 은평구, 2021.11.25, <https://encykorea.aks.ac.kr/Article/E0067016>

구도완 & 김성주(2006). 은평뉴타운사업의 사회영향평가. 환경사회학연구. 10(2). 9-17

박성재, 양훈 & 한건희(2021). P-Median 모델을 활용한 서울시 임시 선별 검사소 공간적 효율성

증대 방안 연구. 서울 시립대학교 도시과학대학 온라인 아카이브. 5-10

박보라, 이규진 & 최기진(2013). 휴리스틱 P-Median 알고리즘을 이용한 자전거주차장 최적입지선정.

대한 토목학회 논문집. 33(5). 1-5

Wikipedia contributors. (n.d.). *Haversine formula*. In Wikipedia. Retrieved November 24, 2024,

from [https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula)

이문섭(2017). 서울시 공공자전거 무인 대여소의 이용통성 분석: 동대문구 38개 대여소의 이용비율

분석을 중심으로. 서울시립대학교 일반대학원. 86-88

서울열린데이터광장. (2024). 서울시 공공자전거 대여소별 이용정보(월별)

[<https://data.seoul.go.kr/>]. 서울특별시.

[서울시 공공자전거 대여소별 이용정보\(월별\)> 데이터셋> 공공데이터 | 서울열린데이터광장](#)

서울열린데이터광장. (2024). 서울시 공공자전거 대여소 정보 [<https://data.seoul.go.kr/>].

서울특별시. [서울시 공공자전거 대여소 정보> 데이터셋> 공공데이터 | 서울열린데이터광장](#)