A Model for Optimizing Ideal Hospital Location for Students in Durham University

IMDS Mini Group Project - Group 27

Anran Ma, Yingcheng Lin, Yi-Ting Tsou, Yung-Wei Ko

Page count: 10

Introduction: 1
Methodology: 4
Findings and Conclusions: 5
Appendix: 3
References: 1

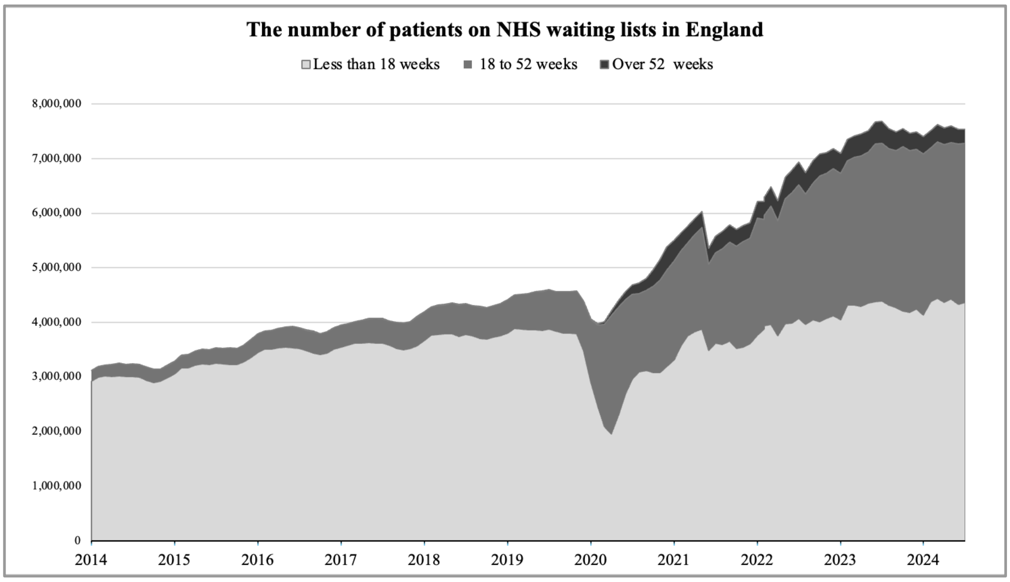
Table of Contents

1.0 Introduction

1.1 Research Background

In recent years, the United Kingdom's National Health Service (NHS) has been encountering a great range of escalating challenges. According to NHS statistics, the issue of surging demand and unequal resource distribution is one of the biggest problems among these intensifying healthcare difficulties. Since the beginning of the pandemic in 2020, the number of patients waiting for hospital treatment in England has risen sharply. By October 2023, the waiting list for planned NHS treatment in England had soared to another record high at 7.7 million (NHS, 2024; Statista, 2023; The Guardian, 2022). These situations highlight a tremendous mismatch between the demand for healthcare services and the NHS's capacity.

Graph 1. The number of patients on NHS waiting lists in England from 2014 to 2024



In Durham County, where Durham university is located, the issue is similarly pressing. The shortest estimated hospital waiting time here is around 7 weeks, while the longest one can reach over 19 weeks (Appendix 1; NHS, 2024). This further highlights the fact that Durham County is also facing the capacity issues within the NHS.

1.2 Research Aims

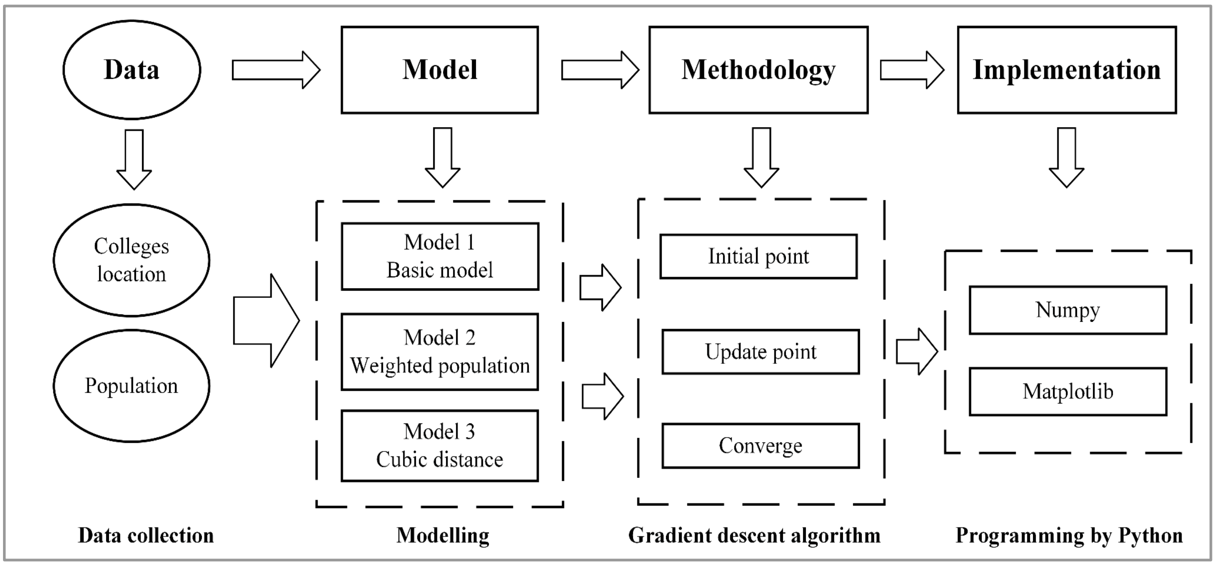
The aim of this report is to determine the optimal location for a potential student-only hospital in Durham, using partial differentiation, linear algebra and gradient descent. By using the college locations, straight-line distance and demographic factors for the main data set, this study seeks to find the best location to minimize the overall distance from colleges, thereby reducing access times and improving healthcare accessibility.

2.0 Methodology

2.1 Research Approach

The first stage is the data collection stage with college locations and population counts as the required input. Next, three models with different approaches: the basic model (Model 1), a population-weighted model (Model 2), and a cubic distance model (Model 3) are generated to best optimize the potential hospital location. Later, the gradient descent algorithm is then applied, starting with an initial point derived from the data, iteratively updating the hospital location based on the gradient, and converging when the change between iterations is minimal. Finally, NumPy and Matplotlib are employed for efficient computations and output visualization, ensuring a systematic and accurate optimization process.

Graph 2. Flowchart of research approach



2.2 Data Collection

The raw data (Appendix 2) gathered from 17 colleges could be categorized into two groups: the location and the population. The former was collected via Google Maps using latitude and longitude, while the latter was obtained through the official webpage of each college. It is noteworthy that the definition of the "population" here refers specifically to the numbers of students living in the college accommodation rather than the total student's population of the college. Therefore, students living outside of the college accommodation, such as those in private accommodation or college subsidiary dormitories, are not included.

2.3 Mathematical Model

2.3.1 Development

In the modelling stage, the position of every college is illustrated with coordinate pairs on a 2D plane where x and y coordinates represent latitude and longitude respectively. The aim is to identify the best possible site for the hospital through performing gradient descent under three different models.

2.3.2 Model 1 - Linear Distance

In the first model, the goal is to minimize the total linear distance between the hospital and all colleges.

① Gradient of Model1:

$\nabla f(x) = \sum_{i=1}^{n} \frac{x - c_i}{\lVert x - c_i \rVert}$	• x: The location vector of the hospital • c_i: The coordinate pair (latitude, longitude) of the i-th college

② The gradient descent update rule is applied to update the location of the hospital:

$x^{(k+1)} = x^{(k)} - \eta \nabla f(x^{(k)})$	• $x^{(k)}$: The location of the hospital at the k-th step • $x^{(k+1)}$: The updated location • η: The learning rate, controlling the step size taken in the direction of the negative gradient

2.3.3 Model 2 - Weighted Population

In the second model, the goal is to make the model closely representative of the real world. Thus, colleges with larger populations are weighted to ensure colleges with larger populations have a greater influence on the optimization.

① Gradient of Model 2 (with the weighted population):

$\nabla f_{weighted}(x) = \sum_{i=1}^{n} \frac{w_i(x - c_i)}{\lVert x - c_i \rVert}$	• $w_i = \frac{p_i}{\sum_{j=1}^{n} p_j}$: The normalized population weight for the i-th college, based on its population p_i • j: The index used to sum over all colleges, from 1 to n. • p_j: The population of the j-th college

② The gradient descent update rule remains the same:

$$x^{(k+1)} = x^{(k)} - \eta \nabla f_{weighted}(x^{(k)})$$

Compared with the first model, the second model more accurately illustrates actual circumstances, as those colleges with larger populations would be favored.

2.3.4 Model 3 - Cubic Distance

There is merely a 1-minute walking distance difference between the location of Model 1 and Model 2. Thus, in the third model, to address the potential issue of insufficient optimization for colleges located farther away, the population-weighted cubic distance is used as another optimization criterion. This approach makes the influence of distance becomes stronger for larger distances, because it places more emphasis on reducing the impact of colleges with larger distances and less emphasis on reducing the impact of colleges with smaller ones.

① Gradient of Model 3 (with the cubic distance):

$$\nabla f_{cubic}(x) = \sum_{i=1}^{n} 3w_i \|x - c_i\| \frac{x - c_i}{\|x - c_i\|}$$

② The gradient descent update rule:

$$x^{(k+1)} = x^{(k)} - \eta \nabla f_{cubic}(x^{(k)})$$

The third model effectively reduces the impact of colleges with smaller distances while prioritizing those farther away, providing a more equitable optimization result.

2.4 Gradient Descent Approach

Gradient descent is chosen as the main approach of this report, because it is a numerical optimization technique used to iteratively minimize the objective function (Table 1).

Table 1. The process of gradient descent

Step 1	Step 2	Step 3	Step 4
Initialization	**Gradient Calculation**	**Position Update**	**Convergence Check**
Start with an initial location for the hospital, typically the weighted or simple mean of all college locations.	Compute the gradient of the objective function at the current location to determine the direction of steepest descent.	Update the hospital's location using the gradient and a predefined learning rate η.	Stop the iterations when the change in location is less than a predefined tolerance ε.

The stopping condition is expressed as:

$$\|x^{(k+1)} - x^{(k)}\| < \epsilon$$

ϵ is the tolerance threshold. In all three models, the learning rate $\eta = 0.01$ ensures stable convergence. This iterative process helps find the best hospital location for each model.

2.5 Implementation of Gradient Descent Algorithm

The gradient descent algorithm was implemented using the NumPy and Matplotlib libraries in Python. The former was used for efficient array handling and the computation of gradients, while the later for visualizing the convergence process and the optimized location.

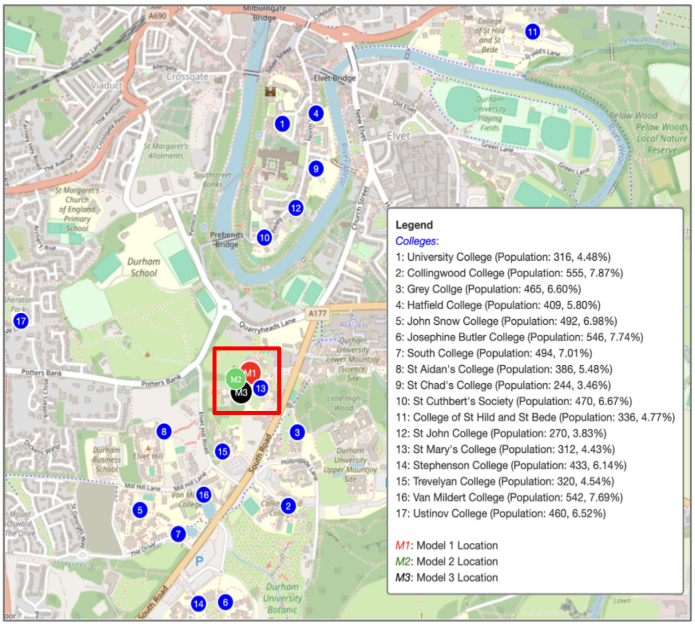
The iteration starts at computing the mean of each college with population weighted as the initial point. As the iteration continues, the computer keeps updating the location according to the gradient until the convergence criterion is met, which is less than the tolerance. Throughout the iterations, the distance is recalculated, and the hospital location is adjusted accordingly.

3.0 Findings and Conclusions

3.1 Findings

Firstly, the three optimized locations are all concentrated in the southern area of Durham University campus, near core teaching and learning buildings, such like the Teaching and Learning Centre and the Durham University Library (Graph 3). The average distance between each optimized location to the Teaching and Learning Centre is about 200 meters, which is approximately a 4-minute walk considering actual walking paths; while to the Bill Bryson Library is roughly 400 meters, approximately 8-minute walk.

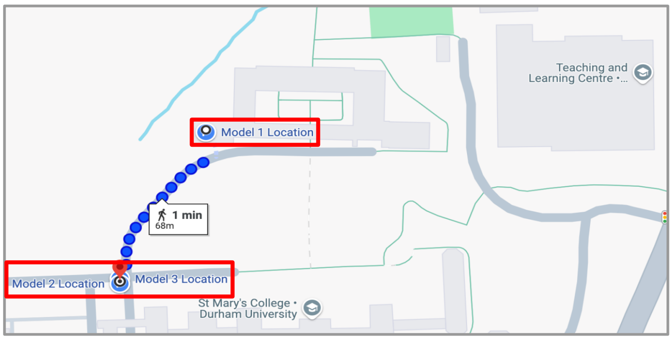
Graph 3. Colleges map with optimized points



● represents the location of each college; ●● ●represent optimized locations.

Secondly, the three optimized locations are relatively close to each other. The first optimized location is across from St. Mary's College, while the second and the third optimized locations almost overlap, with a difference of only 68 meters, around 1-minute walk (Graph 4).

Graph 4. Generated results from three models



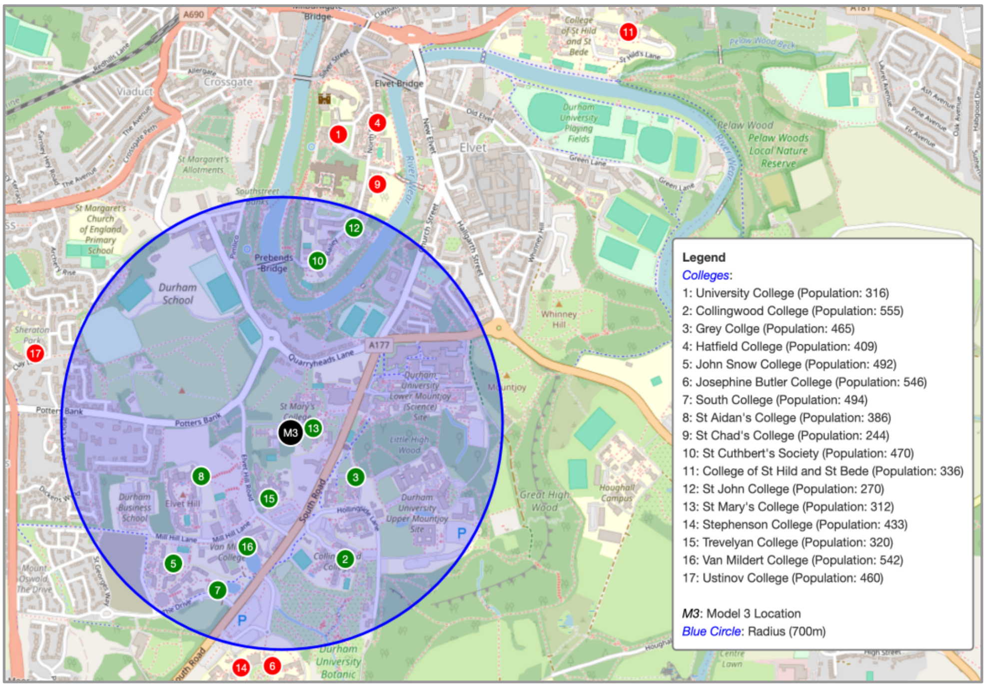
The second finding raises curiosity about why the optimized locations from Model 2 and Model 3 are geographically similar, even when population-weighted cubic distance is used as the optimization criterion. This leads to our third finding: the similarity is primarily due to the symmetrical distribution of the distant colleges and their limited population impact.

Here are the steps leading to the third finding:

Step 1. Exclude the two distant colleges which are geographically distinct from both the Bailey and Elvet Hill areas (Graph 5).

Just like every data analysis, the two distant colleges are excluded from the data set because of two main reasons. First, according to the official guideline to colleges (Durham University, 2022), colleges can be classified as the Hill and the Bailey colleges, based on their geographic distribution in Durham. On one hand, the Hill Colleges are located in the south of Durham City, within a 5-10 minute walk from the main academic area. They are known for being near facilities like the Bill Bryson Library and Teaching and Learning Centre. On the other hand, the Bailey Colleges are in the historic city centre around the River Wear peninsula, near landmarks like Durham Cathedral and Elvet Riverside lecture theatres. They are close to the city centre and the Students' Union. However, ⓫ the College of St Hild and St Bede and ⓱ the Ustinov College do not fit into the traditional grouping of Bailey and Hill colleges, because their unique geographical positions make them distinct from the traditional Bailey and Hill classifications. Thus, these are omitted from the data set.

Graph 5. Map with a circle with a radius of 700 metres



Second, even though the cubic distance approach from Model 3 is supposed to place more emphasis on reducing the impact of colleges with larger distances, the total population from the Ustinov College (835 meters away) and the College of St Hild and St Bede College (1638 meters away) is merely 796, which is around 11.2% of the college's overall population (7050). As the impact of these two colleges is not strong enough to affect the final result (Table 2), they are then regarded as outliers and are excluded in the later analysis.

Table 2. Colleges population and distance between final optimizing point

College Name	Population	Distance (m)	Within 700m
College of St Hild and St Bede	336	1638	FALSE
Hatfield College	409	990	FALSE
Castle College	316	925	FALSE
Ustinov College	460	835	FALSE
St Chad's College	244	810	FALSE
Stephenson College	433	754	FALSE
Josephine Butler College	546	735	FALSE
St John College	270	656	TRUE
John Snow College	492	553	TRUE
South College	494	545	TRUE
St Cuthbert's Society	470	528	TRUE
Collingwood College	555	444	TRUE
Van Mildert College	542	387	TRUE
St Aidan's College	386	310	TRUE
Grey Collge	465	265	TRUE
Trevelyan College	320	222	TRUE
St Mary's College	312	84	TRUE

Step 2. Analyse the distribution of colleges outside the circle with a radius of 700 meters from the centre point (Graph 5)

If we take the optimisation point of the Model 3 as the centre of the circle with a radius of 700 meters, we will get a blue circle covering more than half of the colleges. The remaining five colleges ❶ ❹ ❻ ❾ ⓮ away from the circle coverage are almost symmetrically distributed on both sides of the circle, meaning they have a limited impact on the optimisation results. It is because the influence of these colleges on the optimized points will be neutralised by each other.

Step 3. Compare two groups of colleges

If we group the five colleges located beyond a 700-meter radius circle into two groups: Group 1 (the northern group) ❶ ❹ ❾ and Group 2 (the southern group) ❻ ⓮, it is noticeable that the sum of their population is nearly equal. Given the symmetrically distributed positions between two groups, it is not surprising that the difference in optimized points obtained based on different weights is not significant (Table 3).

Table 3. Comparing the population and distance between two groups of colleges

College Name	Population	Distance (m)
Hatfield College	409	990
University College	316	925
St Chad's College	244	810
Group 1 total population	969	
Stephenson College	433	754
Josephine Butler College	546	735
Group 2 total population	979	

3.2 Conclusions

Here are the conclusions from the analysis of Durham University's college locations and optimized locations. Firstly, there is a pattern for the three optimized locations, as they are concentrated in the southern campus area, near key academic buildings. Secondly, the results from the three models are quite identical. In fact, two of the three optimized points nearly overlap. Thirdly, Model 2 and Model 3 produced similar geographical outcomes, despite using different optimization criteria. This similarity can be explained by two factors: symmetrically location distribution and similar sum of population.

3.3 Limitations

The biggest strength of the three models lies in their fully consideration of Durham University students' needs. Nevertheless, there are two main limitations for this research: the deficiency of student accommodation data and the discrepancy between ideal distance and actual distance.

Firstly, the inadequate account for students living off-campus or residing in subsidiary campus causes a significant data gap in the models. Students living off-campus have various types of accommodations, including student apartments and private residence, making it almost impossible to pinpoint their exact coordinates, let alone their ratios in the college population. Due to these difficulties, the models are only based on the number of students living in each college. However, according to the information on the university's official website, in the academic year 2023/24, the total number of students in Durham University is 21,588. The number of students living in the 17 colleges is merely 7050 in the data set, accounting for only about 30% of the total number (Durham University, 2024), making the models fail to represent the reality. In addition, even knowing that most of the colleges have multiple accommodation areas, it is hard to accurately determine the distribution ratios of students across various accommodation areas within colleges. As a result, only one coordinate address is used to stand for each college, causing an inevitable limitation to the actual student distribution.

Secondly, the models calculate geographical distances based solely on straight-line measurements, failing to take actual traffic routes into consideration, which significantly impact the accessibility of hospitals and the response times of emergency services. The deficiency results from the fact that it is challenging to obtain the actual traffic route from each college to the predicted coordinate point before the results are obtained. Thus, the prediction of the hospital location coordinates is based on the college address, leading to the inability to balance different colleges according to specific transportation routes.

3.4 Further Application

This research approach lays a solid foundation for future applications in a wider range of fields. The method is not only applicable to Durham University, but can also be extended to other collegiate universities, such as Oxford or Cambridge, and any universities that are in need for student-only facilities. Besides, apart from the key factors used in this project (the location, the population and the distance), the models will be allowed to provide a more accurate site prediction in real life if more matrixes are collected into the data set.

Appendix

Appendix 1 – Average waiting time for treatment

Average waiting time for treatment at County Durham and Darlington NHS Foundation Trust			
Specialty	weeks	Specialty	weeks
Cardiology	12	Ophthalmology	15
Dermatology	14	Oral Surgery	19
Ear, Nose and Throat	18	Orthopaedics	17
Gastroenterology	14	Pain Management	7
General Surgery	17	Plastic Surgery	18
Gynaecology	16	Respiratory	17
Haematology	13	Rheumatology	13

Appendix 2 - College's location and population in 24/25

College Name	Latitude	Longitude	Population
Castle College	54.77464458420797	-1.576497757671006	316
Collingwood College	54.76283479544438	-1.576163443216447	555
Grey Collge	54.76511428707511	-1.5756370888690325	465
Hatfield College	54.77497422041353	-1.5745572177040537	409
John Snow College	54.76270950546286	-1.5846439716786258	492
Josephine Butler College	54.759876594032555	-1.5797600293497571	546
South College	54.761985335533495	-1.5824495759295376	494
St Aidan's College	54.765151297111885	-1.5832706888690402	386
St Chad's College	54.773264758166	-1.5745549986650644	244
St Cuthbert's Society	54.77112787768793	-1.57751293117834	470
College of St Hild and St Bede	54.77749247912141	-1.562159671677916	336
St John College	54.77204491365252	-1.5757302177041808	270
St Mary's College	54.76647378125351	-1.5777228140074246	312
Stephenson College	54.75981700359356	-1.581282100514264	433
Trevelyan College	54.76453882995069	-1.5799236465400976	320
Van Mildert College	54.76319194383774	-1.5810200311981304	542
Ustinov College	54.76856791572089	-1.591486938591941	460

Appendix 3 – Model 1

```
# First model (distance)
import pandas as pd
import numpy as np

file_path = '/Users/nanatsou/Desktop/IMDS/Mini Project/College Location & Population_for Python.xlsx'

# Load the Excel file into a Pandas DataFrame
# Assuming the Excel file has columns: "X", "Y" (e.g., latitude and longitude)
data = pd.read_excel(file_path)
print("Column Names:", data.columns)
# Extract the X and Y columns
collegeslocation = data[['Latitude', 'Longitude']].to_numpy()

# Starting point: mean of all college locations
initial_location = np.mean(collegeslocation, axis=0)
current_location = initial_location

# Parameters for gradient descent
learning_rate = 0.01
tolerance = 1e-5
max_iterations = 1000

# Function to compute the gradient
def compute_gradient(current_location, collegeslocation):
    gradient = np.zeros(2)
    for college in collegeslocation:
        diff = current_location - college
        distance = np.linalg.norm(diff)
        if distance != 0:  # Avoid division by zero
            gradient += diff / distance
    return gradient

# Gradient descent optimization
for iteration in range(max_iterations):
    gradient = compute_gradient(current_location, collegeslocation)
    new_location = current_location - learning_rate * gradient
    if np.linalg.norm(new_location - current_location) < tolerance:
        break
    current_location = new_location

# Print the optimized location
print("Optimized Location (X, Y):", current_location/100000)
```

```
Column Names: Index(['College Name', 'Latitude', 'Longitude', 'Population'], dtype='object')
Optimized Location (X, Y): [54.76704747 -1.57849402]
```

Appendix 4 – Model 2

```
# Second Model (population weighted gradient)

import pandas as pd
import numpy as np

file_path = '/Users/nanatsou/Desktop/IMDS/Mini Project/College Location & Population_for Python.xlsx'
data = pd.read_excel(file_path)

# Extract the latitude, longitude, and population columns
collegeslocation = data[['Latitude', 'Longitude']].to_numpy()
population = data['Population'].to_numpy()
# Normalize the population weights so that the sum equals 1
population_weights = population / np.sum(population)

# Starting point: weighted mean of all college locations
initial_location = np.average(collegeslocation, axis=0, weights=population_weights)
current_location = initial_location

# Parameters for gradient descent
learning_rate = 0.01
tolerance = 1e-5
max_iterations = 1000

# Function to compute the population-weighted gradient
def weighted_gradient(current_location, collegeslocation, population_weights):
    gradient = np.zeros(2)
    for i, college in enumerate(collegeslocation):
        diff = current_location - college
        distance = np.linalg.norm(diff)
        if distance != 0:  # Avoid division by zero
            gradient += (population_weights[i] * diff) / distance
    return gradient

# Population-weighted gradient descent optimization
for iteration in range(max_iterations):
    gradient = weighted_gradient(current_location, collegeslocation, population_weights)
    new_location = current_location - learning_rate * gradient
    if np.linalg.norm(new_location - current_location) < tolerance:
        break
    current_location = new_location

# Print the population-weighted optimized location
print("Population-Weighted Optimized Location (Latitude, Longitude):", current_location/100000)
```

```
Population-Weighted Optimized Location (Latitude, Longitude): [54.76644979 -1.57902919]
```

Appendix 5 – Model 3

```python
# Third Model (cubed distance gradient)

import pandas as pd
import numpy as np

# Load data
file_path = '/Users/nanatsou/Desktop/IMDS/Mini Project/College Location & Population_for Python.xlsx'
data = pd.read_excel(file_path)

# Extract latitude, longitude, and population
collegeslocation = data[['Latitude', 'Longitude']].to_numpy()
population = data['Population'].to_numpy()
population_weights = population / np.sum(population)  # Normalize weights

# Initial location: weighted mean
initial_location = np.average(collegeslocation, axis=0, weights=population_weights)
current_location = initial_location

# Parameters for gradient descent
learning_rate = 0.01
tolerance = 1e-5
max_iterations = 1000

# Function to compute the population-weighted gradient
def weighted_gradient(current_location, collegeslocation, population_weights):
    gradient = np.zeros(2)
    for i, college in enumerate(collegeslocation):
        diff = current_location - college
        distance = np.linalg.norm(diff)
        if distance != 0:  # Avoid division by zero
            gradient += (population_weights[i] * diff) / distance
    return gradient

# Function to compute the population-weighted gradient for cubic distance
def cubed_distance_gradient(current_location, collegeslocation, population_weights):
    gradient = np.zeros(2)
    for i, college in enumerate(collegeslocation):
        diff = current_location - college
        distance = np.linalg.norm(diff)
        if distance != 0:  # Avoid division by zero
            gradient += 3 * population_weights[i] * distance * diff / distance  # Simplifies to 3*w*d*diff/d
    return gradient

# Population-weighted gradient descent optimization for linear distance
for iteration in range(max_iterations):
    gradient = weighted_gradient(current_location, collegeslocation, population_weights)
    new_location = current_location - learning_rate * gradient
    if np.linalg.norm(new_location - current_location) < tolerance:
        print(f"Linear distance optimization converged in {iteration} iterations.")
        break
    current_location = new_location

# Save the result of linear optimization
optimized_linear = current_location

# Reinitialize the starting point for cubic distance optimization
current_location = initial_location

# Population-weighted gradient descent optimization for cubic distance
for iteration in range(max_iterations):
    gradient = cubed_distance_gradient(current_location, collegeslocation, population_weights)
    new_location = current_location - learning_rate * gradient
    if np.linalg.norm(new_location - current_location) < tolerance:
        print(f"Cubic distance optimization converged in {iteration} iterations.")
        break
    current_location = new_location

# Save the result of cubic optimization
optimized_cubic = current_location

# Print the results
print("Optimized Location (Cubic Distance) (Latitude, Longitude):", optimized_cubic / 100000)
```

```
Cubic distance optimization converged in 0 iterations.
Optimized Location (Cubic Distance) (Latitude, Longitude): [54.76646795 -1.57902807]
```

References

Durham University (2022) *Your Guide to Colleges*. Available at: https://www.durham.ac.uk/media/durham-university/visit-us/documents/open-days/Open-Day-Colleges-Guide-2022.pdf (Accessed: 12 Dec 2024).

Durham University (2024) *About Us.* Available at: https://www.durham.ac.uk/about-us/ (Accessed: 12 Dec 2024).

National Health Service (2024) *County Durham and Darlington NHS Foundation Trust.* Available at: https://www.myplannedcare.nhs.uk/ney/durham-darlington/ (Accessed: 25 Nov 2024).

National Health Service (2024) *Waiting List Minimum Data Set (WLMDS) Information.* Available at: https://www.england.nhs.uk/statistics/statistical-work-areas/rtt-waiting-times/wlmds/ (Accessed: 25 Nov 2024).

Statista (2023) *Number and percentage of patient's waiting for elective (non-urgent) treatment within 18 weeks in England from August 2007 to April 2023*. Available at: https://www.statista.com/statistics/986875/waiting-times-for-elective-treatment-in-england/ (Accessed: 25 Nov 2024).

The Guardian (2022) *NHS England waiting list reaches another record high in March*. Available at: https://www.theguardian.com/society/2022/may/12/nhs-england-waiting-list-another-record-high-march (Accessed: 25 Nov 2024).