Problem- Facility location
Team- Go Girls

Data required-
1. Ratio of number of existing hospitals and total population in a neighborhood. If a neighborhood has a low ratio, establishing new facilities will be more impactful.
2. Ratio of number of public transportation stops, such as bus and tram stops, and total population
    Motivation - People living in poorly connected regions find it difficult to reach facilities. Accordingly, more hospitals should be set up here.
3. Average waiting time in hours
4. Tax percentage- income, assets such as property
5. Highest level of education of individual who do not study anymore
    Categories- dropouts, high school, Bachelor's, Master's, PhD
6. Immigration- to consider cultural and social disadvantages
    - depends on each country's definition and society
    - for the Netherlands, non-western immigrants tend to be at higher disadvantage socially
7. Women safety- percentage of women who feel safe in a given neighborhood when traveling alone
    - Areas which are known to be safe might be better candidates to establish new facilities
    - Alternatively, women living in unsafe areas might not want to travel far for help.

Accordingly, another approach could be to set up more facilities in unsafe regions to help women. However, for this challenge at the current time, we are focusing on establishing facilities in safer areas. Unsafe areas can be considered in future improvements.

Special needs- Depends on countries and culture. For example, USA has abortion related laws, where women in states such as Texas struggle more in finding facilities which provide related treatment. However, as we are focusing on Rotterdam for this challenge, no such specific requirements were found. However, other needs such as osteoporosis, maternal treatment, breast cancer, and other issues such as PCOD/PCOS have been focused upon.

Approach-
Divide the region, Rotterdam, to such regions or neighborhoods. Assign score to each data category mentioned above. Use a sum of these scores to generate a weight for each neighborhood in the city.

Scoring-
1. Ratio of number of hospitals and total population- If this ratio is high, a lower score should be assigned. Accordingly, the score is the inverse of this ratio, i.e. population per hospital.
Note- pseudo counts to be added to number of hospitals to avoid division by zero when inversing
2. Ratio of public transportation stops and total population- If this ratio is high, a lower score is assigned. Accordingly, the score is the inverse of this ratio, i.e. population per transportation facility.
3. Average waiting time- Use directly
4. Tax percentage - 1 - % of income spent on tax

5. Education-
    Dropout - 1
    High school - 2
    Bachelor's - 3
    Master's - 4
    PhD - 5

Assign inverse of the value. For example, PhD gets ⅕ = 0.2 score. Whereas high school dropout gets 0.5 score

6. Immigration - if non-western, in case of the Netherlands, assign a 1. Otherwise 0.
7. Safety- Use directly

Example for population = 10,000

	Hospital/population	Transportation/population	Waiting time in hours	Tax %	Education level until completion of studies	Immigration	Women safety %
Data	0	5	0.3	21%	Bachelor's	Non-western	85%
Assigned Score	1/(0 + 1). Adding a one to avoid division by zero. Score = 1	⅕ = 0.2	0.3	1 - 0.21 = 0.79	⅓ = 0.33	1	0.85

Final weight assigned to this neighborhood = 1 + 0.2 + 0.3 + 0.79 + 0.33 + 1.0 + 0.85

This weight can be used for linear programming to minimize the distance between individuals living in a neighborhood and the hospitals in that area, subject to some constraints, such as-

    Number of new hospitals < 50
    0 <= distance between individual location and nearest hospital <= 20 km (chosen arbitrarily)

Special medical needs-
Conditions being considered based on Dutch global health strategy- TB, HIV, Mental health, sexual and reproductive issues including menstrual, maternal, breast cancer, others like iron deficiency and PCOD/PCOS
Option 1- Assign a probability to each condition based on information, such as from GP. Use this probability to decide if a given location provides this care.

Option 2- It can be seen that TB, HIV, and mental health affect both genders, whereas the sexual and reproductive issues mentioned are women-specific. Accordingly, there are 3 general conditions and 4 women-specific conditions. Probability of a facility offering women-specific needs = 4/7

Pseudo code-

Inputs-
1. Population in each neighborhood **i**
2. Number of hospitals in each neighborhood- $\mathbf{hospitals_i}$
3. Number of public transportation stops in each neighborhood $\mathbf{transport_i}$
4. Median waiting time $\mathbf{time_i}$
5. Tax % $\mathbf{tax_i}$
6. Median education level $\mathbf{edu_i}$
7. Immigration information $\mathbf{imm_i}$
8. % of women who feel safe in that neighborhood when traveling alone $\mathbf{safe_i}$
9. xi = set of coordinates that are candidates for new hospitals; obtained from Open Street API. For Rotterdam, we consider locations which have walking, biking, and public transportation facilities simultaneously, as the city is well connected. For other locations, this criteria can be changed accordingly.

```
global variables
number of neighbourhoods = n
xi
population_dict = {}                    # Dictionary for population in each neighbourhood
hospital_dict = {}                      # Dictionary for number of hospitals in each neighbourhood
stop_dict = {}          # Dictionary for number of public transportation stops in each neighbourhood
waiting_time_dict = {}          # Dictionary for median waiting time in each neighbourhood
tax_dict = {}                   # Dictionary for median tax in each neighbourhood
edu_dict = {}   # Most common education level post completion of studies in each neighbourhood
immigr_dict = {}                # Most common immigration status in each neighbourhood
safety_dict = {}                # Percentage of women who feel safe when alone in each neighbourhood


# Time complexity = O(n), as the number of values depend on number of neighbourhoods
def normalize_data():
        # Normalize data in each dictionary to avoid one criteria getting more weightage than other


# Time complexity = O(1) as operations and dictionary access takes constant time.
def find_weight(i):
        weight = 0
        weight += population_dict[i] / (hospital_dict[i] + 1)
        weight += population_dict[i] / (stop_dict [i] + 1)
        weight += waiting_time_dict[i]
        weight += 1 - tax_dict[i]
        weight += 1/edu_dict[i]
        weight += immigr_dict[i]
        weight += safety_dict[i]
```

```
        return weight


def main():
        normalize_data()
        weights = {}
        for i in range(n):                              # O(n)
                weights[i] = find_weight(i)

# Use these weights to determine the need for a new facility. This could be based on further linear programming, or through simpler approaches such as proposing the center of the neighborhood as the ideal location.

# Time complexity of accessing Open Street API data - O(1)
# Time complexity of proposing weight as find recommendation for new facilities - O(1)
# Linear programming time complexity = Dependent on number of constraints


Final-
# Time complexity without linear programming = O(n)
# Space complexity without linear programming = O(n)
```