

ICCS240: Database Project

Population Estimation in Thailand

21 March 2020

Group Members:

Sirin Chankao (5980753)
Chakeera Wansoh (6080566)
Natthakan Euaumpon (6081213)

Note to Professor:

Due to the current situation, even if we have limited time, we really want to complete this project so even after we submit this project today (21-03-2020) we will continue to work on it. Therefore, there will be constant update in our github site until the last final date of submission (i.e, 29-03-2020)

Overview:

This project uses the different concepts of database management system and linear algebra. From linear algebra, we learned that eigenvector can be use to estimate the population. With this idea in mind, we decided to combine this with our DBMS knowledge. Therefore, the aim of this project is to estimate the future population of regions in Thailand by calculating from the previous years. Many of the project's information (or evidences that the database exists) can be seen in the project's github. <https://github.com/chakeera/PopulationEstimation>

Theory:

For the sake of explanation, let's assume that A is a matrix. This will work only if A is a symmetric matrix. We need to find all positive eigenvalues which are required for our numerical approach. We are going to implement this by combining Power Method with Rayleigh quotient iteration. This iteration has the following formula:

$$b_{i+1} = \frac{(A - \mu_i I)^{-1} b_i}{\|(A - \mu_i I)^{-1} b_i\|}$$

where b_{i+1} is the next approximation of the eigenvector and I is the identity matrix. After that we can set the next approximation of an eigenvalue to the Rayleigh quotient:

$$\mu_{i+1} = \frac{b_{i+1}^* A b_{i+1}}{b_{i+1}^* b_{i+1}}$$

However, we need to compute all eigenvalues, so we need to combine this technique with a deflation technique. Now we will get the largest possible eigenvalue. But we need to consider all of them. There are many deflation methods, but for this project, we are going to use Hotelling's deflation. This is one of the most popular techniques,

$$(A - \lambda_1 u_1 u_1^T)$$

where u is elements in a set of N vectors, as eigenvectors are the set of N vectors of A . From the equation above, we will get the matrix that has the same eigenvectors and eigenvalues as matrix A . But the largest eigenvalue is replaced with 0. Thus we can combine this with power method and Rayleigh quotient to find the next biggest and so on until we find all the eigenvalues. Then we substitute the eigenvalues into a matrix to get our eigenvectors where it will be multiplied with $n \times n$ matrix that is the matrix of the offsprings and number of survivors and its result will give us the estimated population value.

Database schema:

We have a total of three tables in our database with the following schema:
(These tables import data from "dataset" folder in github)

AgeStructure(Year INT, Age TEXT, Percentage NUMERIC)

RegionPopulation(Region TEXT, Year INT, Population NUMERIC)

TotalPopulation(Year INT, Population NUMERIC, Yearly_percent_change NUMERIC,
Yearly_change NUMERIC, Migrants INT, Median_Age NUMERIC,
Fertility_Rate NUMERIC, Density INT, Urban_percent NUMERIC,
Urban_Population NUMERIC, Country_Share NUMERIC, Global_Rank INT)

Backend:

For back end, we select values from database and insert them into a 6×6 matrix. This is done by creating a function called "getValueX" and "getValueY". Let x be a fraction of the change in population in each year and y be the total number of offspring produced each year. Let x_1 and y_1 be the value for the first year and so on. Then we would get a matrix that looks like this:

$$\begin{bmatrix} 0 & y_2 & y_3 & y_4 & y_5 & y_6 \\ x_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_5 & 0 \end{bmatrix}$$

We will put this in the form of a nested 2D array. Where each sub-array will contain values in each row. After that we compute the eigenvalues. For this project, we only want positive eigenvalues. This is done by using power method, Rayleigh quotient, LU decomposition and deflation techniques. We, then, convert a matrix from above function to matrix form and compute. We will continue doing until we get satisfied values. Then we compute eigenvector, which will then be multiplied with the original given matrix that we retrieved from a database to get our final results. Our final results would be an array that each values are the approximation of the total population in each age group.

Frontend:

For front end, we use Java GUI to present our resulting data. There will be mainly three windows. The first window as shown in the first figure below. This window will act as the menu window where users can select which option they wanted to use. The option includes 'Population in Thailand' and 'Estimation of Population.' If the user selects 'Population in Thailand,' the 'Population in Thailand' window will appear as shown in the second figure below. If the user selects 'Estimation of Population,' the 'Estimation of Population' window will appear as shown in the third figure below.

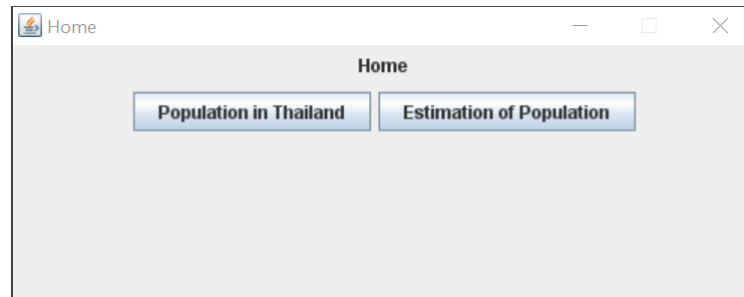


Figure 1: Home Window

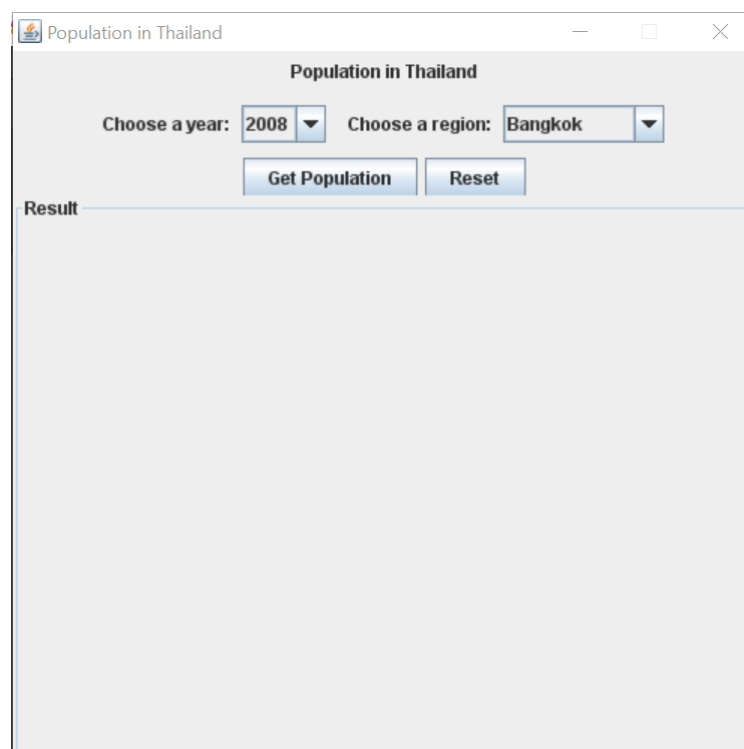


Figure 2: Current Population Window

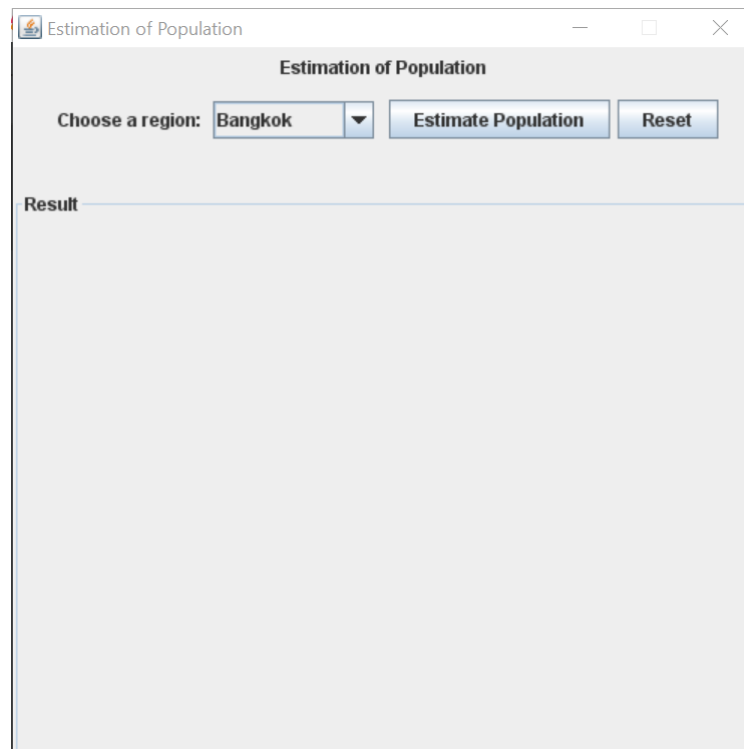


Figure 3: Estimation Window

Video:

Video will be post in github under the folder "Report and Video"

Unfinished functions:

- Linking the postgres database with IntelliJ to compute the matrix function.
- Finish `getValueX()` and `getValueY()` functions.
- Present the resulting data in the front end after computing matrix function.
- Making video