Pseudocode explaining the Approach

# Read the sample data and population characteristics

Read the sample data file "Data.csv" and store it in a suitable data structure (e.g., a pandas DataFrame)

Read the population characteristics from Table 2 and store them in a dictionary or other appropriate data structure

# Calculate probabilities for each category of the three variables

Initialize a dictionary to store probabilities for each category of the "Sex" variable

For each category in the "Sex" variable:

Calculate the probability by dividing the frequency of the category by the total population

Store the category and its probability in the "Sex" probabilities dictionary

Initialize a dictionary to store probabilities for each category of the "Age\_group" variable

For each category in the "Age\_group" variable:

Calculate the probability by dividing the frequency of the category by the total population

Store the category and its probability in the "Age\_group" probabilities dictionary

Initialize a dictionary to store probabilities for each category of the "Highest\_education\_level" variable

For each category in the "Highest\_education\_level" variable:

Calculate the probability by dividing the frequency of the category by the total population

Store the category and its probability in the "Highest\_education\_level" probabilities dictionary

# Generate synthetic population

Initialize

an empty list to store the synthetic population

For 50,000 iterations:

Initialize a dictionary to store the values for the current agent

# Generate Sex

Generate a random number between 0 and 1

Iterate through the "Sex" probabilities dictionary:

Subtract the current category's probability from the generated random number

If the result is less than or equal to 0:

Assign the current category as the "Sex" value for the current agent

Break the loop

# Generate Age\_group

Generate a random number between 0 and 1

Iterate through the "Age\_group" probabilities dictionary:

Subtract the current category's probability from the generated random number

If the result is less than or equal to 0:

Assign the current category as the "Age\_group" value for the current agent

Break the loop

# Generate Highest\_education\_level

Generate a random number between 0 and 1

Iterate through the "Highest\_education\_level" probabilities dictionary:

Subtract the current category's probability from the generated random number

If the result is less than or equal to 0:

Assign the current category as the "Highest\_education\_level" value for the current agent

Break the loop

Add the agent dictionary to the synthetic population list

# Save synthetic population as a CSV file

Create a pandas DataFrame from the synthetic population list

Save the DataFrame as a CSV file named "synthetic\_population.csv"

# Compute frequencies for the generated synthetic population

Initialize counters for each category of the three variables

For each agent in the synthetic population:

Increment the counter for the corresponding category based on the agent's values

# Save output frequencies in a text file

Open a text file named "output\_frequencies.txt" for writing

Write the "Sex" frequencies to the file in the required format

Write the "Age\_group" frequencies to the file in the required format

Write the "Highest\_education\_level" frequencies to the file in the required format

Close the text file.

Name: Gurajapu Likhitha Chuvya

Email: <glchu833@gmail.com>