Part 1

In [87]:

```python
import gurobipy as gp
from gurobipy import GRB
import numpy as np
import random
import pandas as pd

# random seed for reproducibility
random.seed(42)
np.random.seed(42)

# load in population per LGA for 2001, 2006, and 2021
data = pd.read_excel('2001 population.xlsx')
# data = pd.read_excel('2006 population.xlsx')
# data = pd.read_excel('2021 population.xlsx')
```

explanation of what has been done: I have downloaded data from the ABS (please check references for link)and then extracted the population for each LGA for the years 2001, 2006, and 2021 and made three new CSV files with only four relevant attributes year, S/T name, LGA name, and total person no. I uncomment the 2006 or 2021 data for part 4 and part 5.

In [88]:

```python
# define the LGAs
lgas_nsw = data['LGA name'].tolist()

# define the demographic groups for all LGAs
demographic_groups = [
    "0-4yrs old", "5-9yrs old", "10-14yrs old", "15-19yrs old", "20-24yrs old",
    "25-29yrs old", "30-34yrs old", "35-39yrs old", "40-44yrs old", "45-49yrs old",
    "50-54yrs old", "55-59yrs old", "60-64yrs old", "65-69yrs old", "70-74yrs old",
    "75-79yrs old", "80-84yrs old", "85 and over"
]

# generate demographic groups for each LGA
population_data = {lga: {} for lga in lgas_nsw}
# generate random age breakdowns based on population data for each LGA
for lga in lgas_nsw:
    total_population = data[data['LGA name'] == lga]['Total persons no.'].values[0]
    remaining_population = total_population

    population_data[lga] = {}
    for group in demographic_groups:
        if group == demographic_groups[-1]:
            population_data[lga][group] = remaining_population
        else:
            min_population = 0
            max_population = min(remaining_population, total_population // 5)  # ensure the group doesn't exceed 20% of the total population
            population_data[lga][group] = random.randint(min_population, max_population)
            remaining_population -= population_data[lga][group]

# (RDI) data for different age groups
# the RDI data is approximate from the RDI data given in the report converted into the same demographic groups with the census data
rdi_data = {
    "0-4yrs old": 500,
    "5-9yrs old": 700,
    "10-14yrs old": 1000,
    "15-19yrs old": 1300,
    "20-24yrs old": 1000,
    "25-29yrs old": 1000,
    "30-34yrs old": 1000,
    "35-39yrs old": 1000,
    "40-44yrs old": 1000,
    "45-49yrs old": 1000,
    "50-54yrs old": 1300,
    "55-59yrs old": 1300,
    "60-64yrs old": 1300,
    "65-69yrs old": 1300,
    "70-74yrs old": 1300,
    "75-79yrs old": 1300,
    "80-84yrs old": 1300,
    "85 and over": 1300
}

# total daily intake of calcium(mg) demand for each LGA
total_milk_demand_by_lga = {lga: 0 for lga in lgas_nsw}

for lga in lgas_nsw:
    total_demand = sum(population_data[lga][age_group] * rdi_data[age_group] for age_group in demographic_groups)
    total_milk_demand_by_lga[lga] = total_demand

# convert calcium(mg) demand into to liters (milk demand)
for lga, total_demand_mg in total_milk_demand_by_lga.items():
    total_demand_liters = total_demand_mg * 1.0E-6
    total_milk_demand_by_lga[lga] = total_demand_liters

# generate random distance data
distance_data = {(i, j): np.random.randint(1, 200) for i in lgas_nsw for j in lgas_nsw if i != j}
```

In [89]:

```python
# define a model
model = gp.Model("MilkDistribution")

# decision variables
x = model.addVars(lgas_nsw, lgas_nsw, vtype=GRB.BINARY, name="x")
y = model.addVars(lgas_nsw, vtype=GRB.BINARY, name="y")
n = model.addVars(lgas_nsw, vtype=GRB.INTEGER, name="n")

# objective function (minimizing total distribution costs)
```

You have assumed that each LGA can only receive milk from one distribution centre

```python
obj = (
     gp.quicksum(distance_data[i, j] * x[i, j] * total_milk_demand_by_lga[j] * 0.10 for i in lgas_nsw for j in lgas_nsw if i != j) +
    gp.quicksum(n[i] * 1000 for i in lgas_nsw)
)
model.setObjective(obj, GRB.MINIMIZE)

# constraints
# each LGA should receive its total milk demand
for j in lgas_nsw:
    model.addConstr(gp.quicksum(x[i, j] for i in lgas_nsw) == 1, name=f"Meet_Demand_{j}")

# each distribution center can supply milk to LGAs if it's built
for i in lgas_nsw:
    for j in lgas_nsw:
        model.addConstr(x[i, j] <= y[i], name=f"Center_Supply_{i}_{j}")
```

How is n related to the rest of the model?

```python
# total distribution capacity of a center in LGA i must not exceed 50% of the state's demand
for i in lgas_nsw:
    model.addConstr(n[i] <= 0.5 * total_milk_demand_by_lga[i], name=f"Capacity_{i}")

# max of 3 distribution centers can be built
model.addConstr(gp.quicksum(y[i] for i in lgas_nsw) <= 3, name="Max_Centers")

model.optimize()
model.update()
```

```
Gurobi Optimizer version 10.0.2 build v10.0.2rc0 (mac64[rosetta2])

CPU model: Apple M1
Thread count: 8 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 16900 rows, 16899 columns and 50181 nonzeros
Model fingerprint: 0x7b740abb
Variable types: 0 continuous, 16899 integer (16770 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [2e-01, 6e+03]
  Bounds range     [1e+00, 1e+00]
  RHS range        [6e-01, 2e+02]
Presolve removed 129 rows and 129 columns
Presolve time: 0.12s
Presolved: 16771 rows, 16770 columns, 50052 nonzeros
Variable types: 0 continuous, 16770 integer (16770 binary)
Found heuristic solution: objective 31101.761420

Root relaxation: objective 1.508833e+04, 8689 iterations, 0.47 seconds (0.99 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 15088.3263    0 1155 31101.7614 15088.3263  51.5%     -    0s
H    0     0                    20974.308940 15088.3263  28.1%     -    0s
     0     0 15122.6173    0 1350 20974.3089 15122.6173  27.9%     -    1s
     0     0 15158.0194    0 1387 20974.3089 15158.0194  27.7%     -    2s
     0     0 15158.4833    0 1432 20974.3089 15158.4833  27.7%     -    2s
     0     0 15158.5188    0 1430 20974.3089 15158.5188  27.7%     -    2s
     0     0 15158.6389    0 1430 20974.3089 15158.6389  27.7%     -    2s
H    0     0                    18978.798970 15789.6044  16.8%     -    2s
     0     0 15789.6044    0 1518 18978.7990 15789.6044  16.8%     -    2s
     0     0 15789.6044    0 1602 18978.7990 15789.6044  16.8%     -    3s
     0     0 15792.8332    0 1688 18978.7990 15792.8332  16.8%     -    3s
     0     0 15792.8332    0 1689 18978.7990 15792.8332  16.8%     -    3s
     0     0 15792.8332    0 1780 18978.7990 15792.8332  16.8%     -    3s
     0     0 16072.7411    0 1730 18978.7990 16072.7411  15.3%     -    4s
     0     0 16072.7411    0 1728 18978.7990 16072.7411  15.3%     -    4s
     0     2 16072.7411    0 1728 18978.7990 16072.7411  15.3%     -    4s
     1     4 16072.7411    1 1374 18978.7990 16072.7411  15.3%  2528    5s
*    6     6               2    18098.714870 16072.7411  11.2%  1967    5s
   128     2     cutoff   49      18098.7149 17192.2086  5.01%   645   10s

Cutting planes:
  MIR: 3
  Zero half: 36

Explored 181 nodes (102083 simplex iterations) in 10.67 seconds (25.56 work units)
Thread count was 8 (of 8 available processors)

Solution count 4: 18098.7 18978.8 20974.3 31101.8

Optimal solution found (tolerance 1.00e-04)
Best objective 1.809871487000e+04, best bound 1.809871487000e+04, gap 0.0000%
```

In [90]:

```python
# print results
if model.status == GRB.OPTIMAL:
    print("Selected Distribution Centers:")
    for i in lgas_nsw:
        if y[i].x > 0.5:
            print(f"Build a center in {i}")

    total_distribution_cost = 0.0
    for i in lgas_nsw:
        for j in lgas_nsw:
            if i != j and x[i, j].x > 0.5:
                distribution_cost = distance_data[i, j] * x[i, j].x * total_milk_demand_by_lga[j] * 0.10
                total_distribution_cost += distribution_cost

    print(f"Total Distribution Costs: ${total_distribution_cost:.2f}")

# calculations of the new transportation cost after adjusting for inflation for part 5
# inflation_rate = 0.556
# new_distribution_cost_2021 = model.objVal * (1 + inflation_rate)
# print(f"Total Distribution Costs in 2021 (adjusted for inflation): ${new_distribution_cost_2021:.2f}")
```

```
Selected Distribution Centers:
Build a center in Port Macquarie-Hastings
Build a center in Sydney
Build a center in Temora
Total Distribution Costs: $18098.71
```

2001 formulation: Total Distribution Costs: 18098.71 AUD

2006 formulation: Total Distribution Costs: 18143.84 AUD

2021 formulation: Total Distribution Costs: 21611.41 AUD, Distribution Cost with inflation: 33627.36 AUD

Short description

The goal of this model was to develop a sophisticated optimization model to address the complex task of strategically positioning three distribution centers for the efficient supply of milk in New South Wales. To capture the diversity of age groups within each location, the model was created utilizing accurate population demographic data from 2001, as well as randomly generated age distributions for all LGAs. This strategy produced a more nuanced picture of the population and its varying milk demand in each LGA. The addition of recommended dietary intake (RDI) data enables for an accurate evaluation of daily milk demand in each LGA in NSW per day. The calcium intake demand for each LGA is calculated by utilizing the RDI data and multiplying by differenet demographic breakdowns for all LGAs. After that I convert the demand from milligrams (mg) into liters. For this model I assume a liter of milk consists a liter of calcium.

I then uses randomly generated distance data between LGAs to provide realism to the model, imitating the complexities of the actual transportation network. It also guaranteed that each LGA received its whole milk demand, thereby addressing the nutritional needs of the people it served. Furthermore, it limited the capacity of the distribution centers to 50% of the state's total demand, preventing any single facility from becoming overburdened. The model compensated for the financial implications and resource management by restricting the number of distribution centers to a maximum of three, giving a comprehensive and realistic answer to NSW's milk distribution demands. After running the model the three selected centers will be built at Port Macquarie-Hastings, Sydney, and Temora. And the total distribution costs is 18098.71 AUD.

Part 2 sensitive analysis (Modelling 10% population growth in the next three years)

In [91]:

```python
# initialize selected centers as an empty list
selected_centers_by_year = {}

years = 3

# loop through each year and run the model with the updated data
for year in range(1, years + 1):
    # create a model for each year and apply the adjustments
    model = gp.Model(f"MilkDistribution_Year{year}")

    # initialize demographic_data dictionary
    demographic_data = {year: {lga: {group: None for group in demographic_groups} for lga in lgas_nsw} for year in range(1, years + 1)}

    # adjust demographic data for each year (10% growth per annum)
    for lga in lgas_nsw:
        for group in demographic_groups:
            initial_population = population_data[lga][group]
            # new population with 10% growth for each year
            new_population = initial_population * (1.0 + 0.10) ** year
            demographic_data[year][lga][group] = new_population

    # recalculate total milk demand based on the updated demographic data for this year
    total_milk_demand_by_lga[year] = {lga: sum(demographic_data[year][lga][age_group] * rdi_data[age_group] for age_group in demographic_groups) * 1.0E-6 for lga in lga

    # decision variables
    x = model.addVars(lgas_nsw, lgas_nsw, vtype=GRB.BINARY, name="x")
    y = model.addVars(lgas_nsw, vtype=GRB.BINARY, name="y")
    n = model.addVars(lgas_nsw, vtype=GRB.INTEGER, name="n")

    # obj func (minimizing total distribution costs)
    obj = (
        gp.quicksum(distance_data[i, j] * x[i, j] * total_milk_demand_by_lga[year][j] * 0.10 for i in lgas_nsw for j in lgas_nsw if i != j) +
        gp.quicksum(n[i] * 1000 for i in lgas_nsw)
    )
    model.setObjective(obj, GRB.MINIMIZE)

    # constraints
    # each LGA should receive its total milk demand
    for j in lgas_nsw:
        model.addConstr(gp.quicksum(x[i, j] for i in lgas_nsw) == 1, name=f"Meet_Demand_{j}")

    # each distribution center can supply milk to LGAs if it's built
    for i in lgas_nsw:
        for j in lgas_nsw:
            model.addConstr(x[i, j] <= y[i], name=f"Center_Supply_{i}_{j}")

    # total distribution capacity of a center in LGA i must not exceed 50% of the state's demand
    for i in lgas_nsw:
        model.addConstr(n[i] <= 0.5 * total_milk_demand_by_lga[year][i], name=f"Capacity_{i}")

    # max of 3 distribution centers can be built
    model.addConstr(gp.quicksum(y[i] for i in lgas_nsw) <= 3, name="Max_Centers")

    model.optimize()
    model.update()

    # store the selected centers and total distribution cost for this year
    selected_centers = [lga for lga in lgas_nsw if y[lga].x > 0.5]
    selected_centers_by_year[year] = selected_centers

    # initialize total_distribution_cost_by_year dictionary
    total_distribution_cost_by_year = {}

    total_distribution_cost = sum(
        distance_data[i, j] * x[i, j].x * total_milk_demand_by_lga[year][j] * 0.10
        for i in lgas_nsw
        for j in lgas_nsw
        if i != j and x[i, j].x > 0.5
    )
    total_distribution_cost_by_year[year] = total_distribution_cost

    # print results
    print('--------------------------------------------')
    print(f"Year {year} - Selected Distribution Centers: {selected_centers}")
    print(f"Year {year} - Total Distribution Costs: ${total_distribution_cost_by_year[year]:.2f}")
    print('--------------------------------------------')
```

```
Gurobi Optimizer version 10.0.2 build v10.0.2rc0 (mac64[rosetta2])

CPU model: Apple M1
Thread count: 8 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 16900 rows, 16899 columns and 50181 nonzeros
Model fingerprint: 0x00bf5883
Variable types: 0 continuous, 16899 integer (16770 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [3e-01, 7e+03]
  Bounds range     [1e+00, 1e+00]
  RHS range        [7e-01, 2e+02]
Presolve removed 129 rows and 129 columns
Presolve time: 0.08s
Presolved: 16771 rows, 16770 columns, 50052 nonzeros
Variable types: 0 continuous, 16770 integer (16770 binary)
Found heuristic solution: objective 34211.937562

Root relaxation: objective 1.659716e+04, 8912 iterations, 0.50 seconds (0.99 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 16597.1590    0 1155 34211.9376 16597.1590  51.5%     -    0s
H    0     0                    23859.894509 16597.1590  30.4%     -    1s
     0     0 16634.8790    0 1349 23859.8945 16634.8790  30.3%     -    1s
     0     0 16684.2009    0 1346 23859.8945 16684.2009  30.1%     -    2s
     0     0 16704.4660    0 1597 23859.8945 16704.4660  30.0%     -    3s
     0     0 16704.4889    0 1601 23859.8945 16704.4889  30.0%     -    3s
     0     0 17372.1166    0 1689 23859.8945 17372.1166  27.2%     -    3s
     0     0 17372.1166    0 1736 23859.8945 17372.1166  27.2%     -    3s
     0     0 17372.1166    0 1647 23859.8945 17372.1166  27.2%     -    3s
     0     0 17372.1166    0 1700 23859.8945 17372.1166  27.2%     -    3s
     0     0 17680.0152    0 1789 23859.8945 17680.0152  25.9%     -    4s
     0     0 17680.0152    0 1796 23859.8945 17680.0152  25.9%     -    4s
     0     0 17680.0152    0 1794 23859.8945 17680.0152  25.9%     -    4s
     0     0 17680.2564    0 1794 23859.8945 17680.2564  25.9%     -    4s
     0     0 17680.2564    0 1815 23859.8945 17680.2564  25.9%     -    4s
     0     0 17680.2564    0 1813 23859.8945 17680.2564  25.9%     -    4s
     0     2 17680.2564    0 1812 23859.8945 17680.2564  25.9%     -    4s
     1     4 17680.2564    1 1377 23859.8945 17680.2564  25.9%  3084    5s
*    6     6               2    20265.720101 17680.2564  12.8%  1858    5s
*   11     6               3    19908.586357 17680.2564  11.2%  1619    6s
   145     3     cutoff   32      19908.5864 18314.4776  8.01%   594   10s

Cutting planes:
  MIR: 2
  Zero half: 46

Explored 273 nodes (129901 simplex iterations) in 11.94 seconds (32.35 work units)
Thread count was 8 (of 8 available processors)

Solution count 4: 19908.6 20265.7 23859.9 34211.9

Optimal solution found (tolerance 1.00e-04)
Best objective 1.990858635700e+04, best bound 1.990858635700e+04, gap 0.0000%
--------------------------------------------
Year 1 - Selected Distribution Centers: ['Port Macquarie-Hastings', 'Sydney', 'Temora']
Year 1 - Total Distribution Costs: $19908.59
--------------------------------------------
Gurobi Optimizer version 10.0.2 build v10.0.2rc0 (mac64[rosetta2])

CPU model: Apple M1
Thread count: 8 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 16900 rows, 16899 columns and 50181 nonzeros
Model fingerprint: 0x5e063758
Variable types: 0 continuous, 16899 integer (16770 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [3e-01, 8e+03]
  Bounds range     [1e+00, 1e+00]
  RHS range        [7e-01, 2e+02]
Presolve removed 129 rows and 129 columns
Presolve time: 0.08s
Presolved: 16771 rows, 16770 columns, 50052 nonzeros
Variable types: 0 continuous, 16770 integer (16770 binary)
Found heuristic solution: objective 37633.131318

Root relaxation: objective 1.825687e+04, 8842 iterations, 0.44 seconds (0.98 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 18256.8749    0 1155 37633.1313 18256.8749  51.5%     -    0s
H    0     0                    26394.613627 18256.8749  30.8%     -    0s
     0     0 18298.3669    0 1351 26394.6136 18298.3669  30.7%     -    1s
     0     0 19106.2181    0 1427 26394.6136 19106.2181  27.6%     -    2s
     0     0 19106.2181    0 1431 26394.6136 19106.2181  27.6%     -    2s
     0     0 19448.0168    0 1548 26394.6136 19448.0168  26.3%     -    3s
H    0     0                    25378.913817 19448.0168  23.4%     -    3s
     0     0 19448.0168    0 1681 25378.9138 19448.0168  23.4%     -    3s
     0     0 19448.0168    0 1638 25378.9138 19448.0168  23.4%     -    3s
     0     0 19448.2821    0 1587 25378.9138 19448.2821  23.4%     -    4s
     0     0 19448.2821    0 1584 25378.9138 19448.2821  23.4%     -    4s
     0     2 19448.2821    0 1584 25378.9138 19448.2821  23.4%     -    4s
     1     4 19448.2821    1 1375 25378.9138 19448.2821  23.4%  2643    5s
*    6     6               2    21899.444993 19448.2821  11.2%  1852    5s

Cutting planes:
  MIR: 1
  Zero half: 34

Explored 165 nodes (77130 simplex iterations) in 8.47 seconds (20.73 work units)
Thread count was 8 (of 8 available processors)

Solution count 4: 21899.4 25378.9 26394.6 37633.1

Optimal solution found (tolerance 1.00e-04)
Best objective 2.189944499270e+04, best bound 2.189944499270e+04, gap 0.0000%
--------------------------------------------
Year 2 - Selected Distribution Centers: ['Port Macquarie-Hastings', 'Sydney', 'Temora']
Year 2 - Total Distribution Costs: $21899.44
--------------------------------------------
Gurobi Optimizer version 10.0.2 build v10.0.2rc0 (mac64[rosetta2])

CPU model: Apple M1
Thread count: 8 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 16900 rows, 16899 columns and 50181 nonzeros
Model fingerprint: 0x2a193fe1
Variable types: 0 continuous, 16899 integer (16770 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [3e-01, 8e+03]
  Bounds range     [1e+00, 1e+00]
  RHS range        [8e-01, 2e+02]
Presolve removed 129 rows and 129 columns
Presolve time: 0.08s
Presolved: 16771 rows, 16770 columns, 50052 nonzeros
Variable types: 0 continuous, 16770 integer (16770 binary)
Found heuristic solution: objective 41396.444450

Root relaxation: objective 2.008256e+04, 8594 iterations, 0.44 seconds (0.94 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 20082.5624    0 1155 41396.4445 20082.5624  51.5%     -    0s
H    0     0                    28870.472356 20082.5624  30.4%     -    1s
     0     0 20129.0334    0 1358 28870.4724 20129.0334  30.3%     -    1s
     0     0 20159.8468    0 1445 28870.4724 20159.8468  30.2%     -    2s
     0     0 20160.2637    0 1448 28870.4724 20160.2637  30.2%     -    2s
     0     0 20160.3649    0 1447 28870.4724 20160.3649  30.2%     -    2s
     0     0 21020.2610    0 1488 28870.4724 21020.2610  27.2%     -    3s
     0     0 21020.2610    0 1488 28870.4724 21020.2610  27.2%     -    3s
     0     0 21392.8184    0 1692 28870.4724 21392.8184  25.9%     -    4s
     0     0 21393.1103    0 1742 28870.4724 21393.1103  25.9%     -    4s
     0     0 21393.1103    0 1766 28870.4724 21393.1103  25.9%     -    4s
     0     0 21393.1103    0 1762 28870.4724 21393.1103  25.9%     -    4s
     0     2 21393.1103    0 1762 28870.4724 21393.1103  25.9%     -    4s
     1     4 21393.1103    1 1380 28870.4724 21393.1103  25.9%  2689    5s
*    6     6               2    24089.389492 21393.1103  11.2%  1911    6s
   138     3 23135.8493   49 1030 24089.3895 23135.8493  3.96%   533   10s

Cutting planes:
  MIR: 2
  Zero half: 39

Explored 191 nodes (90328 simplex iterations) in 10.42 seconds (23.18 work units)
Thread count was 8 (of 8 available processors)

Solution count 3: 24089.4 28870.5 41396.4

Optimal solution found (tolerance 1.00e-04)
Best objective 2.408938949197e+04, best bound 2.408938949197e+04, gap 0.0000%
--------------------------------------------
Year 3 - Selected Distribution Centers: ['Port Macquarie-Hastings', 'Sydney', 'Temora']
Year 3 - Total Distribution Costs: $24089.39
--------------------------------------------
```

Discussion

The sensitivity analysis, which was carried out over a three-year period in response to a 10% yearly population growth, gives useful insights into the dynamics of milk distribution within the defined LGAs. The output consistently indicates 'Port Macquarie-Hastings,' 'Sydney,' and 'Temora' as the most cost-effective distribution centers for all three years. This consistent selection of centers suggests a strong strategic advantage that will remain even in the face of growing population demand. It reflects a model that is optimized to manage escalating milk demand effectively without requiring a change in distribution center options.

Given the model is basically the same as the above, I only adjusts demographic data for each year (10% growth per annum). Parallel to the selection of centers, total distribution expenses show a significant rising trend throughout the three-year period. The overall distribution cost in Year 1 is 19908.59AUD, Year 2 is 21899.44AUD, whereas the cost in Year 3 is 24,089.39AUD. This cost increase is closely related to the predicted 10% yearly population growth, showing the model's capacity to adjust to spikes in milk demand. These findings highlight the model's ability to optimize distribution decisions in the face of increasing demographics, displaying a high level of efficiency in addressing expanding needs.

The model efficiently changes distribution amounts to suit rising population demands, as evidenced by the consistency in center selection and cost increases. It reflects the model's capacity to adapt to changing conditions while sticking to its primary approach of center selection. These findings lay a solid framework for long-term planning, ensuring that milk distribution stays responsive and cost-effective even during population shifts.

To accommodate rising demand, the government could consider expanding the capacity of current facilities or establishing new ones. It is critical to guarantee that the planned facilities have a lifespan of at least 20 ears in order to correspond with long-term objectives. This adjustment may also include analyzing the effectiveness of the transportation network and ensuring that it can manage the additional demand. The government can discover possible modifications needed to meet population expansion while maintaining an efficient and cost-effective distribution system by providing the sensitivity analysis, it explains how the best plan changes with different parameters and assists in making educated decisions to guarantee the model remains viable and responsive to dynamic demographic changes throughout time. This long-term outlook is essential for efficient urban and regional planning, assuring the government's capacity to fulfill rising milk demand while optimizing resource allocation.

Part 3 short discussion

Governments are challenged with a complicated terrain of elements that transcend beyond population dynamics in their quest for successful long-term planning. One of the most important factors is economic resiliency. The stability and expansion of a country's economy over a twenty-year horizon are not assured. Governments must remain adaptive, with budgetary policies geared to handle variations in inflation, interest rates, and economic progress. Budget decisions must be flexible in order to ensure financial sustainability and the capacity to support important public services. Economic stability serves as the foundation for a country's success.

Another critical aspect of long-term government planning is technological improvement. Technology is expected to grow at an unprecedented rate over the next two decades, ushering in automation, artificial intelligence, and revolutionary shifts in the workplace. Governments must be at the forefront of training their labor forces for new tech-driven employment, supporting R&D, and solving complex concerns such as data privacy and cybersecurity. Promoting innovation and technology preparedness is critical since these variables will drive economic development and competitive advantage in the future global scene.

Environmental sustainability, with a particular emphasis on climate change mitigation, is an unavoidable problem for government agencies. To tackle environmental challenges and cut carbon emissions in the next decades, rigorous eco-friendly legislation is required. Investments in clean energy, conservation, and ecologically sound policies are critical to ensuring environmental quality and, eventually, economic resilience. Finally, the convergence of economic, technical, and environmental factors will influence the trajectory of nations during the next twenty years. Addressing these diverse difficulties and charting the course towards a successful and sustainable future will need flexibility, adaptation, and a forward-thinking attitude.

Part 4

The comparison of the model's projections with actual 2006 data reveals important insights. Initially, the model anticipated an additive trend in distribution and transportation costs over three years, which corresponded to an expected 10% annual population growth. Overall distribution expenses increased from 19908.59 AUD in Year 1 to 21899.44 AUD in Year 2 and 24089.39 AUD in Year 3. However, when the actual 2006 data is examined, a significant disparity appears. The actual overall distribution cost was 18143.84 AUD, which was significantly less than the model's forecasts even though it has already been five years.

The difference between the model's projections and the actual data for 2006 might be ascribed to a number of variables, including the assumption of a 10% annual population increase. In truth, population growth may have been less than predicted between 2001 and 2006. Furthermore, differences in age breakdowns each year can have a significant influence on milk demand, emphasizing the importance of exact demographic data and flexible modeling. It is critical to recognize that the model's strength resides in its adaptability and flexibility. While the model's projections may differ from reality, it provide useful insights into how to make distribution decisions depending on various scenarios. By updating the model on a regular basis with the most recent data, it may function as a proactive tool for decision-makers, allowing them to meet changing market situations successfully while being cost-effective.

Part 5

Three significant conclusions resulted from running models from 2001 to 2021. First, without accounting for inflation, the total distribution cost in 2021 was 21611.41 AUD. However, after accounting for a 55.6 percent inflation rate throughout this time period, the new distribution cost that was calculated with inflation has increased dramatically to 33627.36 AUD. This significant increase highlights the undeniable impact of inflation on long-term operational expenses, emphasizing the need for businesses and organizations involved in distribution and supply chain management to meticulously consider the effects of inflation when developing financial strategies and long-term planning. It emphasizes the vital function of your optimization model in assisting with educated judgments in a dynamic economic context, where it enables cost, capacity, and demand evaluation to adapt and thrive amid changing conditions.

Moreover, even when the distribution cost in 2021 without inflation is compared, it is still significantly less than the 10% annual population growth, which amounts to 24,089.39 AUD after three years. The actual population growth rate from 2001 to 2021 was about 24%, implying that our randomly generated population breakdowns for each LGA may not exactly reflect real-life realities. This also highlights the need of using real-world data for population breakdowns to provide more accurate and realistic distribution planning solutions.

Furthermore, I realized a notable trend when analyzing the 2021 data. From 2001 to 2006, costs exhibited modest growth, but from 2006 to 2021, they increased dramatically for about three thoudsands dollars(without inflation). This illustrates there are a significant population increase (about 1,354,372 people) according to the ABS from 2006 to 2021 generates a rise in milk demand. Nonetheless, the underlying cause of the significant cost shift is the high inflation rate. In conclusion, analyzing the consequences of inflation underlines the importance of adaptation and constant optimization in navigating the changing economic environment, demanding periodic strategy revisions.

References

National Health and Medical Research Council. (2021, April 19). Calcium | Eat For Health. Eatforhealth.gov.au. https://www.eatforhealth.gov.au/nutrient-reference-values/nutrients/calcium

Regional population by age and sex, 2021 | Australian Bureau of Statistics. (2022, July 9). Www.abs.gov.au. https://www.abs.gov.au/statistics/people/population/regional-population-age-and-sex/latest-release#data-downloads