1 Pen-and-Paper Formulation

1.1 Decision Variables

Let there be n cells. Each cell i (for $i = 1, \ldots, n$) corresponds to a combination of Region, Size, and Industry in the data. The decision variable is:

$$x_i \quad \text{(the sample size chosen for cell } i\text{)}.$$

1.2 Objective Function

The code seeks to minimize the deviation between x_i and a "proportional" target p_i.

$$p_i = \text{Population}_i \times \frac{\text{total\_sample}}{\sum_{j=1}^{n} \text{Population}_j}.$$

The objective function is:

$$\min \sum_{i=1}^{n} \left(x_i - p_i\right)^2.$$

This is a least-squares type objective that penalizes deviation of x_i from the proportional target p_i.

1.3 Constraints

1.3.1 Sum of Samples Must Equal Total

$$\sum_{i=1}^{n} x_i \;=\; \text{total\_sample}.$$

1.3.2 Cell-Level Lower and Upper Bounds

Each cell i has:

A lower bound:

$$x_i \;\geq\; \max\left(\frac{\text{Population}_i}{\text{max\_base\_weight}},\; \text{min\_cell\_size},\; 0\right).$$

This ensures each cell has enough sample to limit base weights and respect the cell's minimum size requirement.

An upper bound:

$$x_i \;\leq\; \min\left(\text{Population}_i,\; \text{max\_cell\_size},\; \lceil \text{Population}_i \times \text{conversion\_rate} \rceil\right).$$

This ensures the sample drawn for a cell does not exceed realistic or user-imposed limits.

1.3.3 Dimension-Wise Minimums

For a given dimension (e.g., Region, Size, or Industry), let us say for Region r, we require:

$$\sum_{i \in \text{cells for region } r} x_i \;\geq\; \text{dimension\_mins[Region]}[r].$$

The code generalizes this to any specified dimension (Region, Size, or Industry).

1.3.4 Integrality

$$x_i \;\in\; \mathbb{Z}_{\geq 0} \quad \text{(each } x_i \text{ is a non-negative integer).}$$

1.4 Putting It All Together

In mathematical form:

$$\text{Minimize} \sum_{i=1}^{n} \Big(x_i - p_i\Big)^2$$

subject to

$$\sum_{i=1}^{n} x_i = \text{total\_sample},$$

$$x_i \;\geq\; \max\Big(\frac{\text{Pop}_i}{\text{max\_base\_weight}},\; \text{min\_cell\_size},\; 0\Big), \qquad \forall i,$$

$$x_i \;\leq\; \min\Big(\text{Pop}_i,\; \text{max\_cell\_size},\; \Big\lceil \text{Pop}_i \times \text{conversion\_rate} \Big\rceil\Big), \qquad \forall i,$$

$$\sum_{i \in D(r)} x_i \;\geq\; \text{dimension\_mins}[D][r], \qquad \forall \text{dimension } D,\; \forall \text{value } r,$$

$$x_i \;\in\; \mathbb{Z}_{\geq 0}, \qquad \forall i.$$

2 Closed-Form Solution (Under Simplified Assumptions)

In general, because x_i must be integer and must satisfy multiple lower/upper bounds and dimension constraints, there is no simple closed-form formula for the exact solution.

However, if we ignore:

- Integrality (allowing x_i to be any real number),
- The lower bound constraints,
- The upper bound constraints,
- The dimension minimum constraints,

then the problem reduces to:

$$\min \sum_{i=1}^{n} (x_i - p_i)^2 \quad \text{subject to} \quad \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} p_i.$$

Since

$$\sum_{i=1}^{n} p_i = \text{total\_sample},$$

the constraint

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} p_i$$

is automatically satisfied by choosing $x_i = p_i$. That is the global minimum, giving an objective of 0.

Hence, in that simplified scenario, the closed-form optimal solution is:

$$x_i^* = p_i.$$

Once we impose the integer requirement (plus additional bounds), we must use a mixed-integer optimization method (like the one in the code) instead of a simple formula.

3 Step-by-Step Code Explanation

Data Reshaping

The code transforms your wide-format data (e.g., columns for each Industry, plus Region/Size identifiers) into a long format where each row is (Region, Size, Industry, Population).

Proportional Targets

For each cell i, it computes the proportional target p_i based on population shares and `total_sample`.

Feasibility Checks

The function `detailed_feasibility_check` verifies if it is possible to meet:

- Minimum cell sizes,
- Maximum cell sizes,
- Maximum base weight constraints,
- Dimension-wise minimums,
- Conversion rate limits.

Formulating the MIP

1. Declares integer decision variables x_i.
2. Minimizes $\sum(x_i - p_i)^2$.
3. Constrains $\sum x_i = \text{total\_sample}$.
4. Imposes lower and upper bounds for each cell.
5. Applies dimension-wise minimums if given.

Solving

Tries solvers like **SCIP** or **ECOS_BB** via CVXPY. If no feasible solution is found, a slack-based diagnostic helps identify which constraints are violated.

Outputs

On success, returns the integer x_i and computes the corresponding base weight $\frac{\text{Population}_i}{x_i}$ (if $x_i > 0$).

4 Brief Description of the Solver

The code uses **CVXPY** to model the mixed-integer quadratic problem. CVXPY's role:

- **Model Construction**: Define the decision variables, the objective function (least squares), and the constraints (equalities, inequalities, integrality).
- **Solver Backend**:
  - **SCIP**: A well-known solver for mixed-integer optimization problems.
  - **ECOS_BB**: A branch-and-bound variant of the ECOS solver that supports integer constraints.
- **Result**: The solver attempts to find a feasible solution that minimizes $\sum(x_i - p_i)^2$. If infeasible, the code can diagnose which constraints cause the conflict.

Happy Optimizing!