A Stochastic Integer Programming Approach to Long-term Nursing Home Staffing and Scheduling

Report Submitted to 2025 – 2026 IISE OR Division Case Hackathon

William Vass, Ranem Aldhirgham, Parker Coates, Matthew De Armas

Faculty Advisor: Sergiy Butenko
Graduate Student Mentor: Miltiades Pardalos

Wm Michael Barnes '64 Department of Industrial & Systems Engineering
Texas A&M University, 3131 TAMU, College Station, TX 77843

March 2, 2026

Abstract

This report describes the model we proposed and the results we obtained in our submission to 2025 – 2026 IISE OR Division Case Hackathon.

We develop a two-stage stochastic integer programming model for long-term nursing home staffing under uncertain patient demand. The model addresses the challenge of determining cost-effective staffing plans when daily resident census and case mix are not known with certainty in advance. In the first stage, baseline weekly staffing schedules are selected using repeatable shift templates for registered nurses (RNs), licensed practical nurses (LPNs), and certified nursing assistants (CNAs). In the second stage, recourse hours are introduced to respond to realized demand across scenarios, allowing the model to adjust staffing levels when actual care needs differ from expectations. The proposed framework is designed not only to maintain adequate staffing coverage, but also to satisfy important operational objectives in long-term care settings. In particular, the model enforces weekly work patterns to support labor regulation compliance, promotes schedule continuity and balance through recurring assignment structures, and supports staff-resident consistency by creating stable and predictable coverage patterns over time. By combining proactive schedule design with scenario-based demand response, the model provides a practical decision-support tool for nursing home administrators seeking to balance labor cost, regulatory feasibility, and continuity of care under uncertainty.

1 Introduction

2025 – 2026 IISE OR Division Case Hackathon asks "*to apply operations research principles, techniques, and tools to prototype a long-term staffing and scheduling algorithmic framework that provides intelligent long-term nursing home staffing and scheduling decision support.*" The method of approach we use to address this challenge is based on two-stage stochastic integer programming [2]. We focus on a setting in which resident census and case mix vary day-to-day across six clinical categories (ESS, SCH, SCL, SCC, BSC, and RPF), each inducing different RN, LPN, and CNA workload requirements. In addition, facilities face large and persistent cost differentials between baseline staffing and recourse actions such as overtime and agency labor, making the staffing problem fundamentally a risk–cost trade-off under uncertainty.

From an operational standpoint, nursing-home staffing is naturally a two-stage decision problem. Management must commit to an implementable baseline schedule before the exact day-by-day patient mix is known, and then react to realized fluctuations through overtime, call-ins, or agency hours. The proposed stochastic integer programming model mirrors this workflow: first-stage variables assign nurses of each type to a finite menu of weekly schedule templates built from 8-hour and 12-hour shift patterns, while second-stage recourse variables represent additional nurse-hours required to cover demand shortfalls under realized scenarios. This structure makes the approach particularly suitable for practice, because it (i) enforces feasibility across modeled demand realizations via complete recourse, (ii) explicitly trades off baseline staffing against premium labor through an expected-cost objective, and (iii) remains computationally tractable due to its linear constraints and compact deterministic-equivalent formulation.

A key feature of the case is the long-term nature of planning. Rather than prescribing a single static staffing pattern for the entire year, our framework supports season-specific baseline plans by leveraging historical variation in resident volume and composition. Using the provided Indiana case data and PDPM-based nurse-hour conversion coefficients, we generate representative demand scenarios and solve for baseline template allocations that can be repeated over extended horizons. This enables a transparent comparison between staffing strategies across seasons and quantifies the implied changes in labor mix and cost as demand shifts throughout the year.

The remainder of this report is organized as follows. Section 2 describes the data sources and assumptions, including Indiana labor cost projections, benefit loading, regulatory and operational requirements, and the construction of demand scenarios from resident case-mix data. Section 3 presents the two-stage stochastic integer programming model and its deterministic equivalent implementation. Section 4 reports numerical results, validates the model on an illustrative week, and summarizes recommended baseline schedules for two six-month planning horizons. Finally, Section 5 concludes the report.

2 Assumptions and Data Used

The parameters for the stochastic integer programming model are derived from 2026 labor market projections for Indiana [3–7, 11] and updated federal regulatory standards [1]. The following assumptions govern the cost structures, staffing requirements, and operational constraints:

2.1 Labor Costs and Benefit Loading

The regular hourly cost rate (r_t) for each nurse type includes a 38% benefits multiplier to account for healthcare, 401(k) matching, and other employer-paid benefits [10] . Recourse costs (e_t) reflect the higher premiums of contract or agency staffing.

Table 1: Indiana 2026 nursing wage assumptions with 38% benefit load

Nurse Type (t)	Base Hourly Wage (FT)	Total Regular Rate (r_t)	Agency/Contract Rate (e_t)
RN	\$43.00 – \$47.50	$\approx$ \$62.45	\$52.00 – \$68.00
LPN	\$28.50 – \$31.50	$\approx$ \$41.40	\$34.00 – \$43.00
CNA	\$18.50 – \$21.50	$\approx$ \$27.60	\$24.00 – \$31.00

Note: Total regular rates are calculated using the midpoint of Indiana salary ranges plus the 38% load.

The model is highly flexible, allowing it to be easily changed to accommodate the high variance in hourly wages present at different locations or in different times. Although the regular rate was calculated with the midpoint of full-time nurses wages, the highest contract rate was used in the model to give a slight preference to full-time nurses. This preference reduces staff turnover, allowing nurses to stay with the same patient for a longer period of time.

2.2 Regulatory and Operational Requirements

The model operates under the 2026 CMS regulatory environment. While previous specific minimum nursing hours per resident day (e.g., the 3.48-hour total care requirement) have been repealed, the following constraints remain enforced:

- **RN Presence:** An RN must be onsite for at least 8 consecutive hours per day, seven days a week [1].
- **Burnout Mitigation:** To address the primary causes of nurse burnout—specifically long hours and rotating shifts—the model includes both 8-hour and 12-hour shift templates (Sch). While 12-hour shifts are common, 8-hour shifts are prioritized for improved work-life balance [8].
- **Recourse Availability:** Agency and contract nurses are assumed to be available for recourse (y_{tsd}) to cover fluctuations in demand, typically under contracts lasting up to 13 weeks [9].

2.3 Demand Scenarios and Patient Categorization

Patient demand (dem_{pds}) is modeled using six distinct case types (P) based on care intensity: ESS, SCH, SCL, SCC, BSC, and RPF. Uncertainty is captured via 10 equiprobable scenarios ($|S| = 10$) for each day of the week, derived from the provided Indiana case data. This allows the model to quantify the cost of robustness by preparing for high-demand realizations through a mix of fixed templates and planned recourse.

3 A Stochastic Integer Programming Model for Weekly Nurse Staffing with Daywise Recourse

We consider weekly staffing in a nursing-home setting with three nurse types and a finite menu of fixed weekly schedule templates. Patient demand is uncertain and is represented via scenarios that may vary by day. Staffing decisions are separated into (i) *regular* assignments of nurses to schedule templates, made prior to demand realization, and (ii) *recourse* decisions (extra hours) taken after demand is revealed. The model implemented in the accompanying AMPL file (see Appendix A) is a deterministic equivalent of a two-stage stochastic (integer) program with complete recourse.

3.1 Sets and indices

- N: nurse types, indexed by t: $N = \{\text{RN}, \text{LPN}, \text{CNA}\}$.
- P: patient categories (case types), indexed by p: $P = \{\text{ESS}, \text{SCH}, \text{SCL}, \text{SCC}, \text{BSC}, \text{RPF}\}$.
- D: days of the planning horizon, indexed by d, with $D = \{\text{Su}, \text{Mo}, \text{Tu}, \text{We}, \text{Th}, \text{Fr}, \text{Sa}\}$.
- S: demand scenarios, indexed by s. In the prototype data instance, we use $S = \{1, 2, \ldots, 10\}$.
- Sch: fixed schedule templates, indexed by i. We consider two types of regular schedule templates, based on 12-hour shifts (with three work days followed by four days off) and 8-hour shifts (with five work days followed by two days off). Each schedule template is defined by the day of the week the first shift starts and the number of hours in a shift:

  $$Sch = \{\text{Su12}, \text{M12}, \text{T12}, \text{W12}, \text{R12}, \text{F12}, \text{Sa12}, \text{Su8}, \text{M8}, \text{T8}, \text{W8}, \text{R8}, \text{F8}, \text{Sa8}\}.$$

  For example, a nurse on a T8 schedule has 8-hour day or night shifts on Tuesday through Saturday and has Sunday and Monday off.

3.2 Parameters

- $r_t \geq 0$: regular hourly cost rate for nurse type t.
- $e_t \geq 0$: additional (extra-hour / overtime / agency) hourly cost rate for nurse type t.
- $\text{prob}_{sd} \geq 0$: probability weight of scenario s on day d.
- $\text{wh}_{di} \geq 0$: number of *regular work hours on day d* contributed by one nurse assigned to schedule template i. This parameter encodes the schedule-template structure. For example, in the data instance, a template such as `M12` contributes 12 hours on Monday through Wednesday, and 0 hours on the remaining days, as given by the matrix $\{\text{wh}_{di}\}_{d \in D, i \in Sch}$.
- $h_{pt} \geq 0$: nurse-hours of type t required (per day) to serve one patient of category p. The data instance specifies h as a $|P| \times |N|$ matrix.
- $\text{dem}_{pds} \geq 0$: number of patients of category p on day d under scenario s. In the data file, dem is provided as a set of matrices indexed by $p \in P$, each with rows $d \in D$ and columns $s \in S$.

3.3 Decision variables

- $x_{ti} \in \mathbb{Z}_+$: number of nurses of type t assigned to schedule template i (regular staffing). These decisions are made prior to demand realization and thus do not depend on (s, d).
- $y_{tsd} \in \mathbb{Z}_+$: extra nurse-hours of type t used on day d if scenario s occurs (recourse).

3.4 Demand aggregation and coverage

Given a realized demand scenario s on day d, the total required nurse-hours of type t are computed by aggregating category-level needs:

$$R_{tsd} := \sum_{p\in P} h_{pt}\,\mathrm{dem}_{pds}, \qquad t\in N,\ d\in D,\ s\in S. \tag{1}$$

Regular coverage induced by schedule assignments on day d is

$$C_{td}(x) := \sum_{i\in Sch} \mathrm{wh}_{di}\,x_{ti}, \qquad t\in N,\ d\in D. \tag{2}$$

In the proposed model, feasibility is enforced by allowing extra hours y_{tsd} to supplement regular coverage:

$$C_{td}(x)+y_{tsd} \ \geq\ R_{tsd}, \qquad t\in N,\ d\in D,\ s\in S. \tag{3}$$

3.5 Deterministic equivalent mixed-integer program

The objective is to minimize the sum of (i) regular staffing cost, computed from weekly scheduled hours implied by the templates, and (ii) expected additional cost of extra hours. Define the total weekly hours associated with template i by

$$H_i := \sum_{d\in D} \mathrm{wh}_{di}, \qquad i\in Sch.$$

Then the proposed model can be written as:

$$\begin{aligned}
\min\quad & \sum_{t\in N}\sum_{i\in Sch} r_t\,H_i\,x_{ti} + \sum_{t\in N}\sum_{d\in D}\sum_{s\in S}\mathrm{prob}_{sd}\,e_t\,y_{tsd} && && (4)\\
\text{s.t.}\quad & \sum_{i\in Sch}\mathrm{wh}_{di}\,x_{ti}+y_{tsd}\ \geq\ \sum_{p\in P} h_{pt}\,\mathrm{dem}_{pds}, && t\in N,\ d\in D,\ s\in S, && (5)\\
& x_{ti}\in\mathbb{Z}_+, && t\in N,\ i\in Sch, && (6)\\
& y_{tsd}\in\mathbb{Z}_+, && t\in N,\ d\in D,\ s\in S. && (7)
\end{aligned}$$

Interpretation. The first term in (4) represents the cost of regular staffing assignments: each nurse of type t assigned to template i works H_i hours over the week at hourly rate r_t. The second term is the expected cost of extra hours, where y_{tsd} captures the amount of additional labor used on day d under scenario s at rate e_t, weighted by prob_{sd}.

Constraint (5) imposes daywise, scenario-wise coverage requirements: the sum of regular coverage (from assigned schedule templates) and extra hours must meet the total required nurse-hours implied by the patient mix. The integrality of x is natural since it counts staff. The variable y is modeled as integer in the AMPL implementation; if divisible labor-hours are acceptable for the additional, one may relax (7) to $y_{tsd}\geq 0$ without integrality. Conversely, if y represents whole-person overtime shifts, additional structure (e.g., shift-length constraints) may be introduced.

3.6 Instance-specific structure encoded in the data

The data file instantiates the model for a nursing home with three staff categories

$$N = \{\text{RN}, \text{LPN}, \text{CNA}\}$$

and six patient types

$$P = \{\text{ESS}, \text{SCH}, \text{SCL}, \text{SCC}, \text{BSC}, \text{RPF}\}.$$

Weekly schedules are drawn from the template set *Sch* consisting of 14 patterns labeled `Su12`, `M12`, ..., `Sa12`, `Su8`, ..., `Sa8`. Each template i specifies a day-by-day work-hour vector $\{\text{wh}_{di}\}_{d\in D}$; for instance, `Su12` assigns 12 hours on Sunday and selected subsequent days, while `Su8` assigns 8 hours on Sunday and selected subsequent days, as given in the parameter table wh.

Demand uncertainty is represented by $|S| = 10$ equiprobable scenarios for each day (see Appendix F for details on scenario generation). For each patient type p, the data provides a matrix $\{\text{dem}_{pds}\}_{d\in D, s\in S}$ specifying the number of patients of category p on day d under scenario s. The nurse-hours required per patient category are encoded by the matrix $h = \{h_{pt}\}_{p\in P, t\in N}$, and the resulting total nurse-hours requirements are computed via (1). Regular and additional wage rates are given by $\{r_t\}_{t\in N}$ and $\{e_t\}_{t\in N}$, respectively. Finally, the scenario weights $\{\text{prob}_{sd}\}$ define the expected additional cost contribution in (4). Since we generated scenarios to be equiprobable, $\text{prob}_{sd} = 0.1\ \forall s, d$.

3.7 Discussion and extensions

Model (4)–(7) provides a parsimonious baseline that cleanly separates long-horizon staffing (template assignment) from short-horizon recourse (extra hours). It is readily extendable to incorporate facility-specific requirements such as (i) explicit shift-level coverage and handoff constraints, (ii) upper bounds on overtime/agency usage, (iii) skill-mix and substitution rules (e.g., controlled RN-to-LPN delegation), (iv) fairness and workload-balance constraints across schedule templates, and (v) multi-week planning with continuity constraints and time-off requests.

4 Results and Discussion

4.1 An illustrative example

To verify the model and to provide an illustrative example, we generated 10 equiprobable scenarios for each patient category using the data from the first week of the data set. The corresponding data file is available in Appendix B. The results obtained for the first-stage variables are summarized in Table 2.

This solution can be easily converted to an implementable schedule, as demonstrated in Figure 1 for the case of RNs. Note that the chart allows for an arbitrary selection of starting times of the shifts. It can be conveniently adjusted to include overlaps for shift transition intervals.

To verify the results and to validate the model, we compare the nurse time availability for each day to the actual demand figures from the first-week data used to generate the scenarios, as shown in Table 3.

Table 2: First-stage optimal solution for the model based on the first-week data. The rows correspond to the three types of nurses and the columns represent the 14 schedule templates considered. The entries in the table describe the number of nurses of each type assigned to each schedule. For example, one RN, one LPN, and one CNA is assigned to 12-hour shifts on Sunday, Monday, and Tuesday (Su12 template).

	Su12	M12	T12	W12	R12	F12	Sa12	Su8	M8	T8	W8	R8	F8	Sa8
RN	1	0	3	2	1	2	3	0	5	0	2	0	2	2
LPN	1	0	1	4	1	0	2	1	0	1	0	0	3	2
CNA	1	2	9	4	1	10	11	2	16	1	4	0	3	0

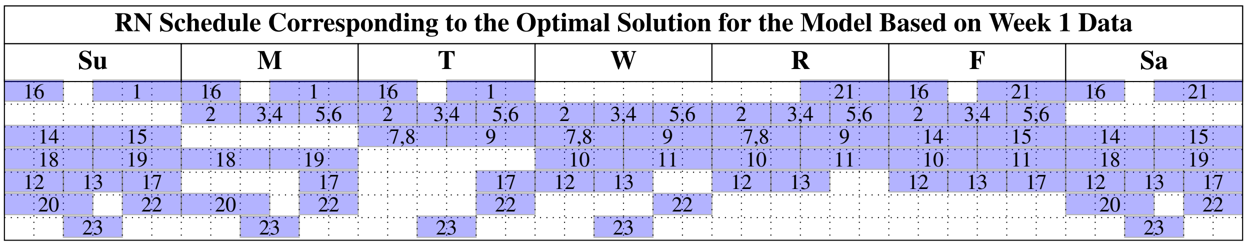

Figure 1: A Gantt chart of the regular (first-stage) schedule for RNs obtained using the optimization model based on the first-week data. The numbers in the bars $(1, \dots, 23)$ are the identification (ID) numbers of 23 RNs. Each nurse's schedule is allocated to a single row of the chart. Each cell corresponds to a 4-hour period during the respective day. For example, RN 1 works 12-hour shifts on Sunday, Monday, and Tuesday, and RN 2 works 8-hour shifts on Monday through Friday.

Table 3: Comparison of actual and model-implied nursing hours by day of week using first-week data. For each nurse type, we report the realized demand in hours ("Actual"), the regular hours provided by the optimal schedule produced by our model ("Model"), and their difference, $\Delta = \text{Model} - \text{Actual}$.

Day	RN			LPN			CNA		
	Actual	**Model**	**Δ**	**Actual**	**Model**	**Δ**	**Actual**	**Model**	**Δ**
Su	120.86	120	0.86	82.82	84	-1.18	337.16	336	1.16
Mo	120.20	120	0.20	82.49	84	-1.51	336.92	336	0.92
Tu	118.03	120	-1.97	81.07	80	1.07	321.04	320	1.04
We	130.88	132	-1.12	90.26	92	-1.74	364.44	364	0.44
Th	126.89	128	-1.11	86.94	88	-1.06	354.00	352	2.00
Fr	133.06	132	1.06	91.31	92	-0.69	374.00	372	2.00
Sa	121.84	120	1.84	84.11	84	0.11	328.20	328	0.20
Total	871.76	872	-0.24	599.00	604	-5.00	2415.76	2408	7.76

4.2 Long-term Schedules

Long-term schedules should take into account the seasonal variability in demand while avoiding abrupt changes in personnel.

4.2.1 Schedules Based on 6-Month Data

The first-stage solutions shown in Tables 4 and 5 specify the weekly staffing patterns that the nursing home should implement and repeat over the two six-month planning horizons. Table 4 corresponds to the October–April period, while Table 5 corresponds to the April–October period. These periods were chosen in correspondence to the beginning and middle of the fiscal year.

Table 4: Optimal weekly baseline schedule by nurse type and shift template for the October 1-March 31 planning horizon. The expected weekly staffing cost is $125,602.

	Su12	M12	W12	R12	F12	Sa12	Su8	M8	T8	W8	R8	F8	Sa8	Total
RN	1	0	0	1	1	1	2	4	2	2	1	1	2	18
LPN	2	1	1	1	0	1	1	0	2	2	0	1	0	12
CNA	12	0	11	12	0	1	0	2	0	0	0	0	16	54

Table 5: Optimal weekly baseline schedule by nurse type and shift template for the April 1-September 30 planning horizon. The expected weekly full-time staffing cost is $125,530.

	Su12	M12	T12	W12	R12	F12	Sa12	Su8	M8	T8	W8	R8	Sa8	Total
RN	0	1	0	1	0	1	0	1	1	1	0	2	2	10
LPN	1	0	1	0	1	0	0	0	1	1	0	1	1	7
CNA	1	0	7	0	7	0	0	0	8	0	1	0	11	35

A comparison of the two tables shows that the recommended staffing schemes differ substantially across the two horizons. These differences reflect seasonal changes in patient demand throughout the year. Historical data suggest seasonal variation in the level and composition of care needs, leading the stochastic optimization model to recommend different mix of nurses and shift-template allocations in the two periods.

In particular, the October–April schedule places a greater emphasis on several CNA-intensive templates, such as `Su12`, `R12`, and `Sa8`, indicating a greater need for baseline CNA coverage during that part of the year. By contrast, the April–October schedule distributes CNA staffing more broadly throughout the week and uses a different mix of RN and LPN templates, reflecting a shift in both total workload and its day-by-day distribution. Thus, the differences between the two staffing plans provide evidence that a single static schedule would be less effective than season-specific baseline plans.

4.2.2 Schedule Based on 12-Month Data

From a managerial perspective, the 12-month plan highlights the trade-off between adaptability and stability. A season-specific approach better matches staffing levels and labor mix to predictable changes in resident volume and case mix throughout the year, while a single annual schedule may be easier to communicate, administer, and sustain over time. Because the proposed model uses repeatable weekly templates, both approaches remain operationally feasible; however, the comparison suggests that facilities expecting meaningful seasonal variation may benefit from periodic schedule updates rather than relying exclusively on one static annual plan.

Table 6: Optimal weekly baseline schedule by nurse type and shift template for the January 1-December 31 planning horizon. The expected weekly full-time staffing cost is $133,723.

	Su12	M12	T12	W12	R12	F12	Sa12	Su8	M8	T8	W8	R8	F8	Sa8
RN	1	0	1	1	2	0	0	1	1	2	0	0	0	2
LPN	0	1	0	1	0	1	0	1	0	1	0	1	0	1
CNA	8	0	8	1	1	10	1	0	0	0	8	0	0	0

5 Conclusion

This report developed a two-stage stochastic integer programming model for long-term nursing home staffing under uncertain patient demand. By determining baseline weekly staffing schedules in advance and supplementing them with recourse hours when realized demand differs from expectations, the model provides a practical framework for balancing cost, coverage, and operational feasibility in a nursing home setting. The first-stage schedule-template decisions and second-stage recourse structure mirror the real planning process faced by administrators, who must commit to regular staffing before exact resident census and case mix are known, then adjust when day-to-day conditions change. In this way, the model achieves its primary goal of intelligent long-term staffing support while remaining aligned with the realities of nursing home operations.

The model also achieves the required secondary objective of labor regulation compliance through its feasible weekly scheduling structure. Because baseline staffing is selected from a finite set of implementable 8-hour and 12-hour weekly templates, the resulting schedules are not arbitrary hour allocations but recurring work patterns designed to satisfy the report's regulatory and operational assumptions, including required RN presence and burnout-conscious shift design. This makes the resulting staffing plans more realistic and more directly usable in practice than a purely aggregate hour-allocation model.

The model further achieves schedule continuity and balance by assigning nurses to repeatable weekly templates that can be carried over a weekly horizon and extended across longer planning periods. Rather than rebuilding schedules from scratch each day, the framework produces stable recurring shift patterns, which promote consistency in work assignments over time. This continuity helps limit unnecessary variation in when employees work, supports a more balanced and predictable staffing structure, and reduces reliance on disruptive last-minute adjustments. The season-specific results in the report also show that these repeatable schedules can be adapted to different demand environments without abandoning their underlying stability, allowing continuity to be preserved even as staffing levels change across planning horizons.

Finally, the model supports staff-resident assignment consistency through recurring nurse schedules that increase the likelihood that residents will encounter familiar caregivers on a regular basis. Although the formulation does not explicitly assign individual nurses to specific residents, it creates a structured and repeatable staffing environment in which the same categories of staff are present on the same shifts over time. This recurring pattern strengthens continuity of care, supports stronger staff-resident familiarity, and contributes to a more dependable daily care experience. In that sense, the model meaningfully advances the objective of assignment consistency at an operational level, even if resident-level matching could be incorporated more explicitly in a future extension.

Overall, the report shows that the proposed stochastic optimization framework does more than minimize expected staffing cost. It achieves labor-feasible weekly schedules through template-based constraints, promotes long-term continuity and balance through repeatable work patterns, and supports caregiver consistency through stable recurring coverage. These features make the model valuable not only as a mathematical scheduling tool, but also as a practical decision-support system for nursing home administrators seeking cost-effective staffing plans that also protect regulatory compliance, workforce stability, and continuity of care. Future work could strengthen these accomplishments even further by incorporating explicit fairness metrics, tighter limits on recourse use, or resident-level assignment variables that directly model caregiver-resident continuity.

References

[1] American Hospital Association. Cms repeals minimum staffing requirements for skilled nursing & long-term care facilities. https://www.aha.org/news/headline/2025-12-02-cms-repeals-minimum-staffing-requirements-skilled-nursing-long-term-care-facilities, 2025. Accessed: 2026-03-02.

[2] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, 2006.

[3] CCI Training. Cna salary guide. https://ccitraining.edu/blog/cna-salary-guide/, 2026. Accessed: 2026-03-02.

[4] CompTool. Licensed practical nurse salary in indiana. https://comptool.com/salaries/indiana/healthcare/licensed-practical-nurse/, 2026. Accessed: 2026-03-02.

[5] Indeed. Cna agency jobs in indianapolis, in. https://www.indeed.com/q-cna-agency-l-indianapolis,-in-jobs.html, 2026. Accessed: 2026-03-02.

[6] Indeed. Lpn nursing agencies jobs in indiana. https://www.indeed.com/q-lpn-nursing-agencies-l-indiana-jobs.html, 2026. Accessed: 2026-03-02.

[7] Indeed. Nurse agency recruiter jobs in indiana. https://www.indeed.com/q-nurse-agency-recruiter-l-indiana-jobs.html, 2026. Accessed: 2026-03-02.

[8] Texas Nurses Association. Practice tip: Reconsidering the 12-hour work shift. https://www.texasnurses.org/news/670791/Practice-Tip-of-the-Week--Reconsidering-the-12-Hour-Work-Shift.htm, 2026. Accessed: 2026-03-02.

[9] TriageStaff. Travel nurse contract length. https://triagestaff.com/blog/travel-nurse-contract-length/, 2026. Accessed: 2026-03-02.

[10] U.S. Bureau of Labor Statistics. Employer costs for employee compensation (ecec). https://www.bls.gov/ecec/, 2026. Accessed: 2026-03-02.

[11] ZipRecruiter. Nursing home rn salary in indiana. https://www.ziprecruiter.com/Salaries/Nursing-Home-Rn-Salary--in-Indiana, 2026. Accessed: 2026-03-02.

Appendices

A AMPL Model

nurse_scheduling1.mod

```
set N;  # types of nurses
set S;  # scenarios
set P;  # patient categories
set D;  # days of the week
set Sch; # schedule options

var x {t in N, i in Sch} >=0, integer;
# the number of regular nurses of type t on schedule i

var y {t in N, s in S, d in D} >=0, integer;
# the number of extra nurse hours of type t on day d

param dem {p in P, d in D, s in S};
# the number of patients of category p in day d under scenario s

param r {t in N};
# regular per hour rate of type-t nurse

param e {t in N};
# irregular (extra) per hour rate of type-t nurse

param prob {s in S, d in D};
# probability of scenario s for day d

param h {p in P, t in N};
# type-t nurse hours required (per day) for a patient of category p

param wh {d in D, i in Sch};
# work hours, per day of the week, for a nurse on schedule i


minimize f: sum {t in N} r[t]*sum{i in Sch}((sum{d in D} wh[d,i])*x[t,i])
+ sum {t in N, s in S, d in D} prob[s,d]*e[t]*y[t,s,d];

subject to demand {t in N, s in S, d in D}:
sum {i in Sch} wh[d,i]*x[t,i] + y[t,s,d] >= sum {p in P} h[p,t]*dem[p,d,s];
```

B Data File Based on the First Week Demands

```
set N := RN LPN CNA;
set S := 1 2 3 4 5 6 7 8 9 10;
set P := ESS SCH SCL SCC BSC RPF;
set D := Su Mo Tu We Th Fr Sa;
set Sch := Su12 M12 T12 W12 R12 F12 Sa12 Su8 M8 T8 W8 R8 F8 Sa8;


param r :=
RN 62.45
LPN 41.4
CNA 27.6;

param e :=
RN 68
LPN 43
CNA 31;

param wh : Su12 M12 T12 W12 R12 F12 Sa12 Su8 M8 T8 W8 R8 F8 Sa8 :=
Su 12  0  0  0  0 12 12 8 0 0 8 8 8 8
Mo 12 12  0  0  0  0 12 8 8 0 0 8 8 8
Tu 12 12 12  0  0  0  0 8 8 8 0 0 8 8
We  0 12 12 12  0  0  0 8 8 8 8 0 0 8
Th  0  0 12 12 12  0  0 8 8 8 8 8 0 0
Fr  0  0  0 12 12 12  0 0 8 8 8 8 8 0
Sa  0  0  0  0 12 12 12 0 0 8 8 8 8 8;


param h : RN LPN CNA :=
ESS 1.85 1.36 3.6
SCH 1.365 0.845 3.4
SCL 1.365 0.845 3.4
SCC 1.034 0.682 3.28
BSC 0.75 0.55 3
RPF 0.75 0.55 3.28;

param prob : Su Mo Tu We Th Fr Sa :=
1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
2 0.1 0.1 0.1 0.1 0.1 0.1 0.1
3 0.1 0.1 0.1 0.1 0.1 0.1 0.1
4 0.1 0.1 0.1 0.1 0.1 0.1 0.1
5 0.1 0.1 0.1 0.1 0.1 0.1 0.1
6 0.1 0.1 0.1 0.1 0.1 0.1 0.1
7 0.1 0.1 0.1 0.1 0.1 0.1 0.1
```

```
8 0.1 0.1 0.1 0.1 0.1 0.1 0.1
9 0.1 0.1 0.1 0.1 0.1 0.1 0.1
10 0.1 0.1 0.1 0.1 0.1 0.1 0.1
;

param dem :=

[ESS,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  21  21  21  21  21  21  21  21  21  21
  Mo  21  21  21  21  21  21  21  21  21  21
  Tu  23  23  23  23  23  23  23  23  23  23
  We  25  25  25  25  25  25  25  25  25  25
  Th  22  22  22  22  22  22  22  22  22  22
  Fr  23  23  23  23  23  23  23  23  23  23
  Sa  26  26  26  26  26  26  26  26  26  26

[SCH,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  17  17  17  17  17  17  17  17  17  17
  Mo  15  15  15  15  15  15  15  15  15  15
  Tu  17  17  17  17  17  17  17  17  17  17
  We  14  14  14  14  14  14  14  14  14  14
  Th  18  18  18  18  18  18  18  18  18  18
  Fr  17  17  17  17  17  17  17  17  17  17
  Sa  16  16  16  16  16  16  16  16  16  16

[SCL,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   8   8   8   8   8   8   8   8   8   8
  Mo   8   8   8   8   8   8   8   8   8   8
  Tu   5   5   5   5   5   5   5   5   5   5
  We   9   9   9   9   9   9   9   9   9   9
  Th   8   8   8   8   8   8   8   8   8   8
  Fr   9   9   9   9   9   9   9   9   9   9
  Sa   3   3   3   3   3   3   3   3   3   3

[SCC,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  26  26  26  26  26  26  26  26  26  26
  Mo  28  28  28  28  28  28  28  28  28  28
  Tu  28  28  28  28  28  28  28  28  28  28
  We  29  29  29  29  29  29  29  29  29  29
  Th  28  28  28  28  28  28  28  28  28  28
  Fr  30  30  30  30  30  30  30  30  30  30
  Sa  31  31  31  31  31  31  31  31  31  31

[BSC,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   2   2   2   2   2   2   2   2   2   2
```

```
  Mo   2   2   2   2   2   2   2   2   2   2
  Tu   2   2   2   2   2   2   2   2   2   2
  We   2   2   2   2   2   2   2   2   2   2
  Th   2   2   2   2   2   2   2   2   2   2
  Fr   2   2   2   2   2   2   2   2   2   2
  Sa   2   2   2   2   2   2   2   2   2   2

[RPF,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  26  26  26  26  26  26  26  26  26  26
  Mo  26  26  26  26  26  26  26  26  26  26
  Tu  20  20  20  20  20  20  20  20  20  20
  We  29  29  29  29  29  29  29  29  29  29
  Th  27  27  27  27  27  27  27  27  27  27
  Fr  30  30  30  30  30  30  30  30  30  30
  Sa  19  19  19  19  19  19  19  19  19  19

;
```

C Data File Based on Year-Long Demands

```
set N := RN LPN CNA;
set S := 1 2 3 4 5 6 7 8 9 10;
set P := ESS SCH SCL SCC BSC RPF;
set D := Su Mo Tu We Th Fr Sa;
set Sch := Su12 M12 T12 W12 R12 F12 Sa12 Su8 M8 T8 W8 R8 F8 Sa8;


param r :=
RN 62.45
LPN 41.4
CNA 27.6;

param e :=
RN 68
LPN 43
CNA 31;

param wh : Su12 M12 T12 W12 R12 F12 Sa12 Su8 M8 T8 W8 R8 F8 Sa8 :=
Su 12  0  0  0  0 12 12 8 0 0 8 8 8 8
Mo 12 12  0  0  0  0 12 8 8 0 0 8 8 8
Tu 12 12 12  0  0  0  0 8 8 8 0 0 8 8
We  0 12 12 12  0  0  0 8 8 8 8 0 0 8
Th  0  0 12 12 12  0  0 8 8 8 8 8 0 0
Fr  0  0  0 12 12 12  0 0 8 8 8 8 8 0
Sa  0  0  0  0 12 12 12 0 0 8 8 8 8 8;


param h : RN LPN CNA :=
ESS 1.85 1.36 3.6
SCH 1.365 0.845 3.4
SCL 1.365 0.845 3.4
SCC 1.034 0.682 3.28
BSC 0.75 0.55 3
RPF 0.75 0.55 3.28;

param prob : Su Mo Tu We Th Fr Sa :=
1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
2 0.1 0.1 0.1 0.1 0.1 0.1 0.1
3 0.1 0.1 0.1 0.1 0.1 0.1 0.1
4 0.1 0.1 0.1 0.1 0.1 0.1 0.1
5 0.1 0.1 0.1 0.1 0.1 0.1 0.1
6 0.1 0.1 0.1 0.1 0.1 0.1 0.1
7 0.1 0.1 0.1 0.1 0.1 0.1 0.1
```

```
8 0.1 0.1 0.1 0.1 0.1 0.1 0.1
9 0.1 0.1 0.1 0.1 0.1 0.1 0.1
10 0.1 0.1 0.1 0.1 0.1 0.1 0.1
;

param dem :=

[ESS,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  11  14  16  17  19  21  25  25  27  28
  Mo  11  14  15  17  19  21  24  26  28  30
  Tu  11  13  16  17  20  22  24  25  27  29
  We  12  14  15  17  19  22  25  26  26  29
  Th  12  14  15  16  18  22  23  25  26  28
  Fr  11  13  15  18  19  22  23  24  27  29
  Sa  12  15  15  18  20  22  24  25  27  29

[SCH,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   7   8  10  11  12  12  13  15  16  17
  Mo   7   9   9  10  11  12  13  15  15  17
  Tu   7   9  10  10  11  12  13  14  16  17
  We   8   9  10  11  11  12  13  13  16  18
  Th   8   8  10  11  12  13  13  14  16  18
  Fr   8   9  10  11  11  12  13  14  15  17
  Sa   7   9  10  11  11  12  13  14  16  17

[SCL,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   4   4   5   8   9  11  13  14  15  17
  Mo   3   4   6   7   8  11  13  14  15  17
  Tu   3   4   5   7   9  10  12  14  16  17
  We   3   4   6   6   8  11  14  15  16  17
  Th   2   4   5   7  10  12  13  14  15  17
  Fr   3   4   5   6   9  10  12  14  15  17
  Sa   3   4   6   7   9  12  13  15  16  17

[SCC,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   6   9  10  13  19  21  22  23  26  30
  Mo   6   8  10  13  17  21  23  24  26  29
  Tu   6  10  12  14  19  20  21  23  25  29
  We   6   9  11  13  18  21  22  24  25  29
  Th   8   8  11  12  16  21  22  24  26  28
  Fr   5   8  10  13  14  21  22  23  26  28
  Sa   6   9  12  14  18  21  22  24  26  30

[BSC,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   2   2   2   2   2   2   2   2   2   2
```

```
  Mo   2   2   2   2   2   2   2   2   2   2
  Tu   2   2   2   2   2   2   2   2   2   2
  We   2   2   2   2   2   2   2   2   2   2
  Th   2   2   2   2   2   2   2   2   2   2
  Fr   2   2   2   2   2   2   2   2   2   2
  Sa   2   2   2   2   2   2   2   2   2   2

[RPF,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  21  22  24  26  28  29  31  31  34  35
  Mo  19  22  25  26  27  29  30  32  33  35
  Tu  19  23  25  26  27  29  31  32  33  36
  We  20  22  24  27  28  29  31  33  35  37
  Th  20  21  24  27  28  29  31  32  33  35
  Fr  19  23  24  25  27  28  29  31  33  35
  Sa  19  21  24  26  27  29  31  33  34  36

;
```

D Data File Based on October-March Demands

```
set N := RN LPN CNA;
set S := 1 2 3 4 5 6 7 8 9 10;
set P := ESS SCH SCL SCC BSC RPF;
set D := Su Mo Tu We Th Fr Sa;
set Sch := Su12 M12 T12 W12 R12 F12 Sa12 Su8 M8 T8 W8 R8 F8 Sa8;


param r :=
RN 62.45
LPN 41.4
CNA 27.6;

param e :=
RN 68
LPN 43
CNA 31;

param wh : Su12 M12 T12 W12 R12 F12 Sa12 Su8 M8 T8 W8 R8 F8 Sa8 :=
Su 12  0  0  0  0 12 12 8 0 0 8 8 8 8
Mo 12 12  0  0  0  0 12 8 8 0 0 8 8 8
Tu 12 12 12  0  0  0  0 8 8 8 0 0 8 8
We  0 12 12 12  0  0  0 8 8 8 8 0 0 8
Th  0  0 12 12 12  0  0 8 8 8 8 8 0 0
Fr  0  0  0 12 12 12  0 0 8 8 8 8 8 0
Sa  0  0  0  0 12 12 12 0 0 8 8 8 8 8;


param h : RN LPN CNA :=
ESS 1.85 1.36 3.6
SCH 1.365 0.845 3.4
SCL 1.365 0.845 3.4
SCC 1.034 0.682 3.28
BSC 0.75 0.55 3
RPF 0.75 0.55 3.28;

param prob : Su Mo Tu We Th Fr Sa :=
1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
2 0.1 0.1 0.1 0.1 0.1 0.1 0.1
3 0.1 0.1 0.1 0.1 0.1 0.1 0.1
4 0.1 0.1 0.1 0.1 0.1 0.1 0.1
5 0.1 0.1 0.1 0.1 0.1 0.1 0.1
6 0.1 0.1 0.1 0.1 0.1 0.1 0.1
7 0.1 0.1 0.1 0.1 0.1 0.1 0.1
```

```
8 0.1 0.1 0.1 0.1 0.1 0.1 0.1
9 0.1 0.1 0.1 0.1 0.1 0.1 0.1
10 0.1 0.1 0.1 0.1 0.1 0.1 0.1
;

param dem :=

[ESS,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  12  14  16  17  18  21  24  26  27  29
  Mo  12  14  15  18  18  20  22  25  26  30
  Tu  11  15  16  17  19  21  23  26  28  32
  We  13  15  15  15  18  22  23  25  26  29
  Th  12  15  15  16  18  23  25  26  28  30
  Fr  12  15  15  18  18  22  23  25  26  31
  Sa  13  15  16  18  20  23  25  27  28  30

[SCH,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  10  12  12  13  14  16  16  17  17  18
  Mo  10  11  13  13  14  15  16  17  17  18
  Tu  10  11  12  13  15  15  15  16  17  18
  We  10  11  12  13  13  16  16  17  17  19
  Th  10  11  11  12  13  14  14  16  17  18
  Fr  10  11  12  12  13  14  16  16  17  18
  Sa  11  12  12  13  14  15  15  16  16  17

[SCL,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   3   5   6   7   8  10  10  13  14  17
  Mo   3   5   6   8   8  11  11  14  14  17
  Tu   4   6   6   7   7  10  10  12  14  17
  We   2   5   5   6   7   9  11  14  16  16
  Th   4   6   6   8   8  10  10  12  14  16
  Fr   2   4   5   7   8   9   9  14  16  17
  Sa   2   4   5   6   8  10  12  13  14  17

[SCC,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   8  10  20  21  22  25  26  27  28  31
  Mo   8  12  18  22  23  25  26  26  28  31
  Tu  10  15  20  21  22  24  26  26  27  31
  We   9  12  21  21  22  24  24  27  28  32
  Th   8  12  20  21  22  24  25  27  29  31
  Fr   8  10  21  23  23  24  24  26  28  31
  Sa   8  12  21  21  22  24  26  26  27  30

[BSC,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   2   2   2   2   2   2   2   2   2   2
```

```
  Mo   2   2   2   2   2   2   2   2   2   2
  Tu   2   2   2   2   2   2   2   2   2   2
  We   2   2   2   2   2   2   2   2   2   2
  Th   2   2   2   2   2   2   2   2   2   2
  Fr   2   2   2   2   2   2   2   2   2   2
  Sa   2   2   2   2   2   2   2   2   2   2

[RPF,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  20  24  25  27  28  29  29  30  32  36
  Mo  20  24  25  27  29  31  31  33  33  35
  Tu  22  24  25  26  26  28  28  30  32  35
  We  18  20  23  26  27  28  29  31  34  36
  Th  22  25  27  28  28  30  30  31  33  34
  Fr  20  20  24  25  27  28  29  31  32  35
  Sa  18  24  24  26  28  29  29  31  34  36

;
```

E Data File Based on March-September Demands

```
set N := RN LPN CNA;
set S := 1 2 3 4 5 6 7 8 9 10;
set P := ESS SCH SCL SCC BSC RPF;
set D := Su Mo Tu We Th Fr Sa;
set Sch := Su12 M12 T12 W12 R12 F12 Sa12 Su8 M8 T8 W8 R8 F8 Sa8;


param r :=
RN 62.45
LPN 41.4
CNA 27.6;

param e :=
RN 68
LPN 43
CNA 31;

param wh : Su12 M12 T12 W12 R12 F12 Sa12 Su8 M8 T8 W8 R8 F8 Sa8 :=
Su 12  0  0  0  0 12 12 8 0 0 8 8 8 8
Mo 12 12  0  0  0  0 12 8 8 0 0 8 8 8
Tu 12 12 12  0  0  0  0 8 8 8 0 0 8 8
We  0 12 12 12  0  0  0 8 8 8 8 0 0 8
Th  0  0 12 12 12  0  0 8 8 8 8 8 0 0
Fr  0  0  0 12 12 12  0 0 8 8 8 8 8 0
Sa  0  0  0  0 12 12 12 0 0 8 8 8 8 8;


param h : RN LPN CNA :=
ESS 1.85 1.36 3.6
SCH 1.365 0.845 3.4
SCL 1.365 0.845 3.4
SCC 1.034 0.682 3.28
BSC 0.75 0.55 3
RPF 0.75 0.55 3.28;

param prob : Su Mo Tu We Th Fr Sa :=
1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
2 0.1 0.1 0.1 0.1 0.1 0.1 0.1
3 0.1 0.1 0.1 0.1 0.1 0.1 0.1
4 0.1 0.1 0.1 0.1 0.1 0.1 0.1
5 0.1 0.1 0.1 0.1 0.1 0.1 0.1
6 0.1 0.1 0.1 0.1 0.1 0.1 0.1
7 0.1 0.1 0.1 0.1 0.1 0.1 0.1
```

```
8 0.1 0.1 0.1 0.1 0.1 0.1 0.1
9 0.1 0.1 0.1 0.1 0.1 0.1 0.1
10 0.1 0.1 0.1 0.1 0.1 0.1 0.1
;

param dem :=

[ESS,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su  11  14  16  18  19  21  25  26  27  28
  Mo  10  14  15  16  19  21  24  27  28  29
  Tu  10  12  16  18  20  22  24  26  27  29
  We  12  14  16  18  20  22  24  26  26  27
  Th   9  13  14  16  18  23  23  25  25  27
  Fr  10  13  14  18  19  22  23  23  24  26
  Sa  11  14  15  18  20  22  24  25  27  29

[SCH,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   7   7   8   9  10  11  11  12  12  12
  Mo   6   8   8   9   9  10  10  10  11  13
  Tu   7   8   8   9   9  10  10  11  11  12
  We   7   8   9  10  10  10  11  11  11  13
  Th   8   8   8   9  10  11  11  12  13  13
  Fr   7   8   8   9  10  10  11  11  12  14
  Sa   7   8   8   9  10  10  11  11  11  13

[SCL,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   4   4   4   5  10  14  14  15  15  17
  Mo   2   3   5   8  10  12  14  15  16  17
  Tu   3   4   5   8  10  13  14  15  16  17
  We   2   3   4   6  10  14  15  16  17  18
  Th   3   4   4   9  12  14  14  15  15  18
  Fr   3   4   5   7  11  12  13  15  15  17
  Sa   3   4   4   7  10  13  15  15  16  17

[SCC,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   5   8   9  11  12  14  16  21  21  23
  Mo   5   6   8  10  12  13  16  20  23  24
  Tu   5   8  10  12  12  14  16  20  21  23
  We   3   8  10  11  12  14  18  20  21  24
  Th   6   8   9  11  12  13  16  18  20  24
  Fr   5   6   9  10  12  14  14  19  21  21
  Sa   6   8   9  12  13  14  16  20  22  24

[BSC,*,*] :  1   2   3   4   5   6   7   8   9  10 :=
  Su   2   2   2   2   2   2   2   2   2   2
```

```
  Mo   2   2   2   2   2   2   2   2   2   2
  Tu   2   2   2   2   2   2   2   2   2   2
  We   2   2   2   2   2   2   2   2   2   2
  Th   2   2   2   2   2   2   2   2   2   3
  Fr   2   2   2   2   2   2   2   2   2   2
  Sa   2   2   2   2   2   2   2   2   2   2

[RPF,*,*] :  1    2    3    4    5    6    7    8    9   10 :=
  Su  22  22  22  24  26  30  31  33  34  35
  Mo  18  20  24  27  29  29  30  32  33  36
  Tu  20  23  24  26  28  30  31  32  34  36
  We  19  20  24  24  27  29  32  34  35  38
  Th  20  22  23  27  29  31  32  33  35  36
  Fr  20  23  24  25  26  29  30  31  33  35
  Sa  19  20  22  25  27  30  33  34  34  36

;
```

F Scenario Generation Python Code

```
import pandas as pd
import numpy as np

# ----------------------------
# USER SETTINGS
# ----------------------------
file_path = "Dailied.csv"
start_date = "2017-04-01"
end_date = "2017-09-30"
num_scenarios = 10
round_to_int = True

# ----------------------------
# READ DATA
# ----------------------------
df = pd.read_csv(file_path)
df.columns = df.columns.str.strip()

df["Date"] = pd.to_datetime(df["Date"], errors="coerce")
df = df.dropna(subset=["Date"]).copy()

# Filter by date range
mask = (df["Date"] >= pd.to_datetime(start_date)) & (df["Date"] <= pd.to_datetime(end_da
filtered_df = df.loc[mask].copy()

if filtered_df.empty:
    raise ValueError("No data found in the specified date range.")

# ----------------------------
# SETTINGS
# ----------------------------
# Match the order in your .dat file exactly
patient_cols = ["ESS", "SCH", "SCL", "SCC", "BSC", "RPF"]

day_order = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday
day_abbr = {
    "Sunday": "Su",
    "Monday": "Mo",
    "Tuesday": "Tu",
    "Wednesday": "We",
    "Thursday": "Th",
    "Friday": "Fr",
    "Saturday": "Sa"
```

```
}
day_rows = [day_abbr[d] for d in day_order]

filtered_df["Weekday"] = filtered_df["Date"].dt.day_name()

# -----------------------------
# BUILD EQUIPROBABLE SCENARIO MATRICES
# -----------------------------
scenario_matrices = {}

for col in patient_cols:
    scenario_matrix = pd.DataFrame(
        np.nan,
        index=day_rows,
        columns=list(range(1, num_scenarios + 1))
    )

    for day in day_order:
        short_day = day_abbr[day]

        values = filtered_df.loc[
            filtered_df["Weekday"] == day, col
        ].dropna().astype(float).values

        if len(values) == 0:
            continue

        # Equiprobable scenarios using midpoint quantiles:
        # 0.05, 0.15, ..., 0.95
        quantile_levels = [(i - 0.5) / num_scenarios for i in range(1, num_scenarios + 1
        scenario_values = np.quantile(values, quantile_levels)

        if round_to_int:
            scenario_values = np.rint(scenario_values).astype(int)

        scenario_matrix.loc[short_day, :] = scenario_values

    scenario_matrices[col] = scenario_matrix

# -----------------------------
# BUILD PROBABILITY MATRIX
# -----------------------------
prob_matrix = pd.DataFrame(
    1.0 / num_scenarios,
    index=list(range(1, num_scenarios + 1)),
```

```
    columns=day_rows
)

# -----------------------------
# PRINT AMPL .dat FORMAT
# -----------------------------
# prob block
print("param prob : Su Mo Tu We Th Fr Sa :=")
for s in prob_matrix.index:
    row_values = " ".join(f"{prob_matrix.loc[s, d]:.1f}" for d in prob_matrix.columns)
    print(f"{s} {row_values}")
print(";")
print()

# dem block
print("param dem :=")
print()

scenario_header = " ".join(f"{s:>3}" for s in range(1, num_scenarios + 1))

for p in patient_cols:
    mat = scenario_matrices[p]

    print(f"[{p},*,*] :{scenario_header} :=")

    for day in mat.index:
        row_values = []
        for s in mat.columns:
            val = mat.loc[day, s]
            if pd.isna(val):
                row_values.append("  .")
            else:
                if round_to_int:
                    row_values.append(f"{int(val):>3}")
                else:
                    row_values.append(f"{val:>3.1f}")

        print(f"  {day:<2} {' '.join(row_values)}")

    print()

print(";")
```