To: Emergency Medicine Residency Program, Kings County/SUNY Downstate
Subject: Advocating for Research Integration into Residency Program Scheduling

Introduction

Established in 1992, the Department of Emergency Medicine at Kings County/SUNY Downstate has consistently upheld its mission of delivering high-quality, compassionate care to the diverse communities of Brooklyn, New York. The department is strengthened by a multidisciplinary team of dedicated physicians, nurses, and allied healthcare professionals who work collaboratively in a dynamic and challenging clinical environment.

The class of 2024-2025, which began residency in July 2021, has now reached its final year of training. As part of their leadership structure, residents select chief residents to represent and advocate on behalf of the entire residency body. One of their main responsibilities is scheduling shifts for resident physicians rotating in the program**.** This comes with details and caveats that we'll slightly highlight in this memo.

What's the problem?

This past year, my friends were chiefs, and I got to witness them struggling through hours on Google Sheets, countless revisions, general scheduling complaints, as well as shift experience complaints. Shift scheduling as been manual and long lasting time sucking industry problem and there are a lot of buzz word promises of 'easy' AI and money driven solutions. In this case, the burden falls on time that could be used on more valuable and impactful applications to residents' physicians' time (not Google Sheets). Most importantly, the complaint that I heard the most was how hectic shifts would be because there were not enough doctors scheduled when there was a big demand of patients.

After a year, friends are graduating to other jobs and fellowships, and a new senior class will take over as chiefs. However, this problem continues to be interesting and actionable because of the engagement around this issue:

1) Former Chief Resident physicians will take on an Attending physician role in the same hospital and will be in a key position to mentor incoming 2026 chief residents in the program, as well as continue to influence programming in the hospital system
2) Former Resident, incoming NYU Clinical Informatics Fellowship, will have access to IT resources and hospital system institutional and medical knowledge to provide feedback or help implementation of research
3) Presents the opportunity to better serve and learn from a unique patient population. Hypothesis that it will reveal patterns unique to the hospital that relate to ways to collaborate better and help the community
4) Presents the opportunity to enhance scheduling policy to increase the effectiveness, efficiency, equity, and admin feasibility of the process. The downstream impacts will not only benefit the scheduler's time but will also improve patient overcrowding, patient care, doctors' well-being, and better control and allocation of hospital resources.

The aims of this research is to be able to recommend and implement incremental actionable improvements to the scheduling process in equitable and ethical ways for the higher goal of improving the experience of both physicians and patients.

Context:

People:

Patient Population	• Large West Indian and Caribbean population • Smaller population from Latin America, Asia, the Middle East, and Eastern Europe • Safety net hospital for many of the medically underserved • Our department often is the first line of health care for many of our patients

Policies:

Kings County Hospital Center

County Hospital

One of the largest Level 1 Trauma Centers

Stroke Center

Sexual Assault Forensic Examiner (SAFE) Center of Excellence

140,000 visits per year - One of the top 15 busiest hospitals in the nation!

SUNY Downstate Medical Center

University setting

Stroke Center

Tertiary care center for transplant, dialysis, and cardiac patients

70,000 visits per year

Designated COVID-19 Facility

Residents rotate between two hospital systems that are across the street from each other.
Kings County Hospital Center

- NYC Health + Hospitals (NYC's public hospital system) - City funded
- Very large, Level 1 Trauma Center, separate Pediatric and Psychiatric EDs. 140,000 visits per year

SUNY Downstate Medical Center & University Hospital of Brooklyn

- State University of New York (SUNY) - State funded
- Moderate-sized ED at UHB; no Trauma Center; mainly adult and pediatric emergency care for training. 70,000 visits per year

Rotations:
Before the monthly scheduling can be created, the chiefs have to first organize the yearly block schedule. It's created once a year and dedicated to assigning doctors to different rotations for the year (13 blocks) which also includes non-emergency department rotations. For the purposes of this memo, this task is ‘out of scope.’ We are assuming this has already been created as it is an upstream dependency to create monthly schedules.

Processes
For this research project, we have simplified the scope to focus on the smaller of the two emergency departments: University Hospital of Brooklyn (UHB)

Requirements are simpler because there is only one floor or one general department to schedule (for example, the Adults and Pediatrics are together). Meanwhile, in the larger city hospital, there are not only 2X more patients and beds but there more more dedicated specialized patient services to schedule for. For example, within the emergency department, there is a separate pediatrics section, trauma section, fast track section, etc.

UHB ED Current Monthly Scheduling Process:
Because it is a smaller hospital, there is a general assumption that 3 total resident doctors are needed to support the larger ED (with nurses and other doctor staff) at the base level. Meaning, 3 residents should be able to handle the average patient demand. It is part of the scheduler task to know when to assign more doctors when the patient demand is expected to be larger.

1. Assign a minimum of one senior and one junior resident to cover the Adult ED at a given time
2. Assign a minimum of one senior to cover the Pediatrics ED at a given time
    - Seniors (8-hour shifts): 7 am-3 pm, 3 pm-11 pm, 11 pm-7 am
    - Juniors (10-hour shifts): 11 am-9 pm, 9 pm-7 am
3. Assign the remaining doctors
    - to the same shifts OR
    - to ‘flex’ shifts (Seniors: 1 pm-9 pm, Juniors: 1 pm-11 pm)
4. While considering
    - Doctors available / on rotation
    - Doctors' shift/work requirements policies
    - Time off requests
    - Wednesdays mornings blocked off for mandatory conference
    - Limit of consecutive shifts in a row policies
    - Manually distributing equitably among days/nights/weekends

Platforms:
Chiefs use Google Sheets to manually create and revise schedules. Time off requests are also submitted through Google Forms. Once the schedule is finalized, it gets uploaded to the hospital's IT system. Over the years, templates passed down between classes and the use of count() functions have been providing incremental opportunities to innovate the process.

Senior	BLOCK 1		Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	...
Adult			1-Jul	2-Jul	3-Jul	4-Jul	5-Jul	6-Jul	7-Jul	8-Jul	...
Min Required	Senior 1	7AM-3PM	Doctor_1	Doctor_1		Doctor_8	Doctor_8	Doctor_2	Doctor_2	Doctor_2	...
	Senior 2	7AM-3PM									...
	Senior Flex 1	1PM-9PM		Doctor_5	Doctor_5	Doctor_2	Doctor_1	Doctor_1	Doctor_1	Doctor_5	...
	Senior Flex 2	1PM - 9PM		Doctor_6	Doctor_9						...
Min Required	Senior 1	3PM-11PM	Doctor_2	Doctor_7	Doctor_7	Doctor_5	Doctor_6	Doctor_8	Doctor_8	Doctor_3	...
	Senior 2	3PM-11PM	Doctor_3	Doctor_3	Doctor_3	Doctor_7	Doctor_3	Doctor_5	Doctor_5	Doctor_7	...
Min Required	Senior 1	11PM-7AM	Doctor_4	Doctor_4	Doctor_4	Doctor_9	Doctor_9	Doctor_9	Doctor_9	Doctor_6	...
	Senior 2	11PM-7AM				Doctor_4					...
Notes / Requests				Doctor_2 OFF			Doctor_5 OFF				
Counts		Doctor_1	Doctor_2	Doctor_3	Doctor_4						
	Nights	3	4	4	5						
	Weekend	5	4	4	5						
	MORNING	3	2	3	3						
	Overall	17	16	15	17						
	PEDS	1	1	2	1						

Problems
- Spoken word patient trends decide the doctor count at a time
- Manually fairly distributing the doctor count per shift across days in the month
- Manually fairly distributing doctors' shift assignments per month

The processes above make the program subject to unrealistic supply and demand of patients to doctor ratios, leading to hetic shifts and bad experiences for both. The following are two data analyses that can improve the program's process and outcomes.

Actionable Insight:

Shift Demand Analysis:
For the memo, since we do not have access to real patient arrival data, we're able to simulate the number of patients arriving per hour. In many healthcare scenarios (especially emergency departments), events such as patient arrivals often fit the Poisson process. This is particularly true when patients arrive from a large population where the rate of arrivals per unit time can be modeled as random, but with a known average (rate). In real-world emergency departments, patient arrivals are typically random events that occur independently of each other. The Poisson distribution is ideal for modeling the number of events (in this case, patients) happening in a fixed time interval when those events are independent of each other.

The intention of this analysis is to be done with real hospital patient data. Studying arrival rates, even at a high-level aggregate, will reveal patterns unique to the hospital and patient population. This will formalize and provide evidence for or against 'spoken word trends,' increasing the probability that we can schedule more doctors when there are more patients, and fewer doctors when there are fewer patients.

While "patients per hour" is a useful metric for gauging overall patient demand, it is not an appropriate measure of individual physician productivity. This metric fails to account for the complexity and variability of individual cases, as physicians may manage multiple patients simultaneously and devote varying amounts of time to each case. It may be more suitable for roles like triage or nursing, where patient interaction is more consistent in duration and nature. Relying on this measure oversimplifies the nuances of medical care, ignores the time-intensive nature of complex cases, and risks fostering a toxic learning environment. Although it's expected that residents will gradually improve their skills, using patients per hour as a benchmark can misrepresent work ethic and motivation and lead to misguided incentives.

Example Shift Demand Analysis:

Simulated data - Estimate Patients/hour & Patients/shift

- Uses the Poisson distribution to simulate the number of patients arriving each hour.

Calculate Doctor Count/shift

If the **total number of patients** in a given shift (calculated through simulated patient demand) exceeds a certain **threshold**, the system decides whether extra doctors are needed.

extra_required <- floor(total_patients / avg_patients_per_hour) - shift$min_required

If the **extra required doctors** calculated from the demand exceed the **base requirement** then,

- The system first **adds extra doctors to the base shift**.
- However, if **flexible coverage** is allowed, the system will **consider adding flex shifts** in addition to the base.

```r
# ---------------------------
# Simulate Patient Arrivals
# ---------------------------
simulate_patient_arrivals <- function(dates) {
  expand.grid(date = dates, hour = 0:23) %>%
    mutate(
      is_weekend = wday(date) %in% c(1, 7),
      is_holiday = date %in% holidays,
      base_lambda = case_when(
        hour >= 7 & hour < 11 ~ 5,
        hour >= 11 & hour < 15 ~ 8,
        hour >= 15 & hour < 20 ~ 10,
        hour >= 20 & hour < 23 ~ 6,
        hour >= 23 | hour < 7 ~ 2,
        TRUE ~ 3
      ),
      adjusted_lambda = base_lambda * ifelse(is_weekend | is_holiday, 1.2, 1),
      patients = rpois(n(), lambda = adjusted_lambda)
    )
}

arrival_data <- simulate_patient_arrivals(dates)
```

```r
# --------------------------
# Generate Shift Demand Table
# --------------------------
generate_shift_demand <- function(arrival_data, shift_templates, avg_patients_per_hour) {
  shift_needs <- list()
  for (i in 1:nrow(shift_templates)) {
    shift <- shift_templates[i, ]
    for (d in unique(arrival_data$date)) {
      shift_hours <- shift$start_hour:(shift$end_hour - 1)
      shift_hours <- shift_hours %% 24
      subset <- arrival_data %>%
        filter(date == d, hour %in% shift_hours)

      total_patients <- sum(subset$patients)
      # Adjust extra required doctors logic
      # based on the average number of patients per hour
      extra_required <- floor(total_patients / avg_patients_per_hour) - shift$min_required
      extra_required <- ifelse(extra_required < 0, 0, extra_required)

      shift_needs[[length(shift_needs) + 1]] <- data.frame(
        date = d,
        shift = shift$shift,
        role = shift$role,
        min_required = shift$min_required,
        extra_required = extra_required,
        is_flex = shift$is_flex,
        is_weekend = wday(d) %in% c(1, 7),
        is_holiday = d %in% holidays,
        type = ifelse(shift$start_hour >= 7 & shift$start_hour < 19, "day", "night"),
        ED_TYPE = shift$ED_TYPE
      )
    }
  }
  bind_rows(shift_needs)
}

shift_demand <- generate_shift_demand(arrival_data, shift_templates, avg_patients_per_hour)
```

Shift Demand Analysis Impact

What It Does	Why It's Important
Dynamically adjusts demand by day	Some days are just busier (weekends, holidays, random fluctuations)
Creates a more realistic schedule	Not every day has the same number of shifts — reflects real hospital needs
Prevents understaffing	Busy days automatically get extra coverage assigned
Prevents overstaffing	Quiet days don't force extra unnecessary shifts
Feeds into the assignment engine	Bigger or smaller shift tables change how doctors are distributed across the month

Programmatic Point system:

The current scheduling process already incorporates counting each doctor's shift assignment to ensure they are meeting requirements as well as that everyone gets the same amount of type of shift (morning, nights, weekends, holidays, etc.) By implementing a point-based system programmatically, the process not only maintains and scales this task automatically but also ranks and provides the next available doctors. By incorporating shift demand analysis, we can then loop through the needs and programmatically assign doctors and create a monthly schedule in seconds.

Benefits of the point system include:

- Reduced risk of manual errors in shift tracking and assignment
- Faster schedule generation, allowing more time to review and refine the schedule before release.
- Flexible and scalable to write in more and new requirements and rules
- Earlier scheduled releases, which support better planning and well-being for doctors
- More time for analysis and feedback, enabling focus on improving scheduling outcomes and identifying broader patterns or issues, rather than spending time on tedious manual tasks

Example Point System:

When the code has to **assign a doctor** to a shift, it **ranks eligible doctors** by a calculated "**penalty" or point score**. **Doctors who haven't met their requirements** are **favored first**. Among those still needing shifts, preference goes to those:

- With fewer weekend or holiday shifts.
- With more balanced day vs night shifts.
- With fewer total shifts assigned.

Loop through shifts needed based on shift demand. The lower the penalty, the more desirable the doctor is for assignment.

```r
# Initialize doctor tracking
doctor_stats <- doctor_stats %>%
  mutate(
    total_shifts = 0,
    weekday_shifts = 0,
    weekend_shifts = 0,
    day_shifts = 0,
    night_shifts = 0,
    holiday_shifts = 0,
    last_day = as.Date(NA),
    streak = 0
  )

assignments$assigned <- NA

for (i in 1:nrow(assignments)) {
  row <- assignments[i, ]
  candidates <- doctor_stats %>%
    filter(role == row$role) %>%
    filter(!(doctor_id %in% doctor_time_off$doctor_id[doctor_time_off$day_off == row$date])) %>%
    filter(!(weekdays(row$date) == "Wednesday" & row$type == "day")) %>%
    filter(is.na(last_day) | row$date - last_day <= 6) %>%
    mutate(
      penalty = abs(day_shifts - night_shifts) +
                weekend_shifts +
                holiday_shifts
    ) %>%
    arrange(total_shifts, penalty)

  if (nrow(candidates) > 0) {
    selected <- candidates$doctor_id[1]
    assignments$assigned[i] <- selected

    # Update tracking
    doctor_stats <- doctor_stats %>%
      mutate(
        total_shifts = ifelse(doctor_id == selected, total_shifts + 1, total_shifts),
```

Point System Impact

Points	**What it does**	**Why it helps fairness**
5 × (over target?)	Big penalty if the doctor has already met their required number of weekday or weekend shifts.	**Pushes doctors who still need shifts** toward the top of the list.
abs(day_shifts - night_shifts)	Penalize the imbalance between day and night shifts.	**Keeps day/night assignments balanced** for each doctor individually.
weekend_shift	Slight penalty for each weekend shift already worked.	**Spreads out weekends more evenly** among doctors.
holiday_shifts	Slight penalty for each holiday already worked.	**Shares the holiday burden** across doctors.
total_shifts	Minor penalty for the total number of shifts worked.	**Avoids overloading doctors** with too many shifts early.

Next Steps

Implementation

- Access patient aggregated hourly arrival trends. Create a process to validate and compare forecasted patient trends with actuals
- Collaborate with the audience to manually run scripts and output results in a familiar Google sheet template to start integrating analyses in the process
- Formally create an app that gives front-end flexibility to run, output, and update results
- Collect doctor feedback on shift experience to validate if the new process helps reduce shift busyness and overload

Future scaling

- Research and expand to larger hospital use cases
- Expand the use case to programmatically create yearly block schedules
- Continue expanding assignment eligibility requirements
- Expand features like preference weighting: e.g., some doctors prefer nights, others days.

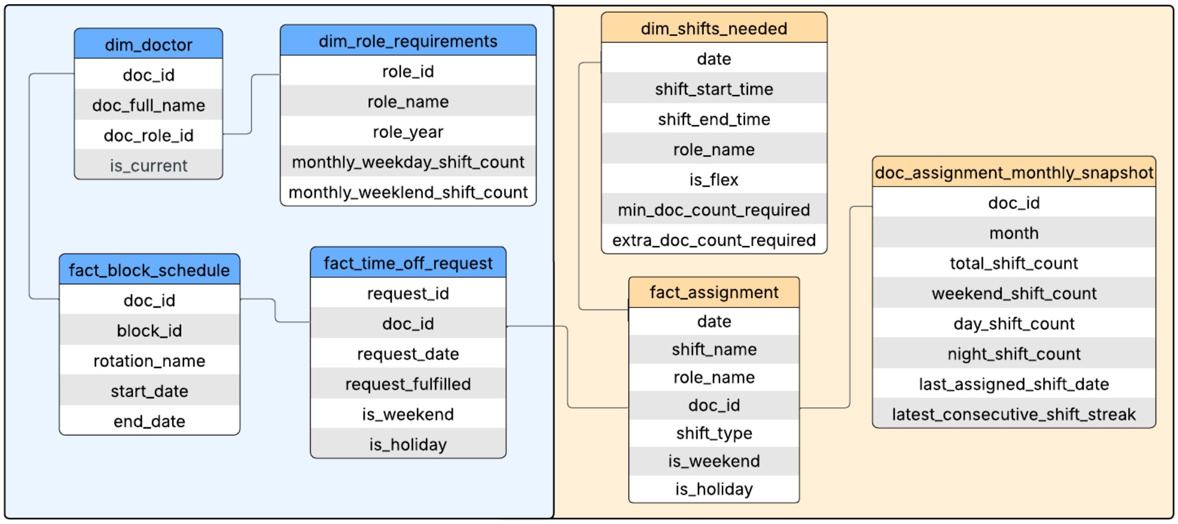
Conclusion:

Integrating data research into residency scheduling is a critical opportunity to improve both physician well-being and patient care through data-driven innovation. By analyzing shift demand through patient arrival patterns, we can determine the appropriate number of doctors needed for each shift. This insight enables the development of a fair and transparent point-based system for assigning shifts, ensuring equitable workload distribution among residents. This approach will streamline the scheduling process while promoting fairness and balance among residents. Most importantly, it ensures that staffing levels are better aligned with patient care demands, creating a more responsive and sustainable system for both physicians and patients.

Appendix:

R Script , README

ETL Diagram proof of concept for app backend and data processing



Prior Knowledge & Research

- Productivity-driven physician scheduling in emergency departments
    - **Business Problem:** ED overcrowding and long patient wait times due to inefficient physician scheduling.
    - **Metrics:**
        - **Patients Seen per Hour (Pt/hr):** This ratio denotes the average number of patients seen by a physician per hour. It is a straightforward measure of productivity but does not account for the complexity or length of consultations.
        - **Relative Value Unit (RVU):** This indicator measures estimates of physicians' effort and practice expense. It reflects the complexity of tasks, technical skills, mental effort, and psychological stress involved in patient care.
        - Other performance metrics:
            - Reduction in demand under-covering.
            - Fairness in the distribution of shifts.
            - Physician dissatisfaction (measured by the difference between wanted and allocated shifts).
            - Patient Wait Times
    - **Methodology:**
        - **Data Analysis:** Historical data from 2008 to 2017 was analyzed to predict patient demand and estimate physician productivity.
        - **Mathematical Model:** A mixed-integer programming model was developed to generate schedules that minimize the difference between patient demand and physician productivity while considering physician preferences and fairness.
        - **Productivity Index:** A productivity index was created based on the number of patients seen per hour and the length of consultations

- **Hypothesis:** Incorporating physician productivity and patient demand into the scheduling process will reduce ED overcrowding and improve the overall quality of healthcare services.

- Stochastic model for physician staffing and scheduling in emergency departments with multiple treatment stages
    - ER schedules are difficult because of the uncertainty of patient arrivals
    - **Metrics**
        - Average waiting time for patients.
        - Average Length of Stay (LOS) for patients.
        - Efficiency improvements in resource utilization.
    - **Methodology:**
        - **Stochastic Modeling:** A two-stage stochastic mixed-integer linear programming model is developed to address uncertainty using the Sample Average Approximation (SAA) method and Latin Hypercube Sampling (LHS).
        - **Simulation:** A discrete event simulation is used to evaluate the results and validate the model.
        - **Application:** The model is applied to a large hospital, with data from 72,988 medical encounters and 85 physicians over a ten-month period.
    - **Hypothesis:** A stochastic model that considers uncertainty in patient arrivals and multiple treatment stages can significantly reduce waiting times and LOS in EDs.

- A GRASP-based algorithm for solving the emergency room physician scheduling problem
    - **Methodology:**
        - **Mathematical Modeling:** The problem is formulated as an Integer Linear Programming (ILP) problem.
        - **Hybrid 22:** A Greedy Randomized Adaptive Search Procedure (GRASP) is combined with Variable Neighborhood Descent (VNDS) and Network Flow Optimization (NFO) to generate and improve schedules.

- Improving Emergency Department Efficiency: A Study of Physician Scheduling Strategies to Reduce Patient Wait Times
    - Emergency Departments (EDs) in Canada, specifically the Foothills Medical Center (FMC) in Calgary.
    - **Metrics**: Average patient wait time, Number of patients leaving without being seen (LWBS), Number of shift extensions and surge calls, Patient Per Hour PPH
    - **Math Used**:

        - **Regression Models**: Multiple regression to validate the variable PPH rate assumption.
            - Dependent Variable: Average PPH.
            - Independent Variables: Hour of the shift, physician code, Waiting Room census, etc.
            - Results: Hour of the shift is highly significant, showing a notable decrease in PPH as the shift progresses.
        - **Stochastic Programming**: Benders Decomposition for solving the two-stage stochastic problem.
        - **MDP**: Policy iteration algorithm to derive optimal policies for shift extensions and surge calls.
    - Business Decision Informed
        - **Scheduling**: Adjust start times of existing shifts to better align with patient arrival patterns.
        - **Shift Extensions**: Implement a control-limit policy for extending shifts based on the number of patients waiting and the hour of the day.
        - **Surge Calls**: Use an optimal policy for issuing surge calls, considering the response rate and relative costs, to manage overcrowding effectively.

- Staffing and Scheduling Emergency Rooms in Two Public Hospitals: A Case Study
    - Baghdad, Iraq
    - **Hypothesis:** It is possible to downsize the number of physicians and nurses while maintaining 24-hour emergency services.
    - Metris:
        - Number of physicians and nurses required per shift.
        - Comparison of available and required treatment times.
        - Reduction in the number of physicians and nurses while maintaining service quality.

- Internet-based self-scheduling is associated with a high degree of physician satisfaction in an academic emergency medicine group