Optimization of Operating Room Utilization at Children's Hospital Colorado

Phillip M. Callahan

University of Wisconsin – Green Bay

DS785: Capstone

Dr. Tracy Bibelnieks

May 1, 2022

Abstract

One of the most valuable services a hospital provides is surgery. Surgical procedures bring in more revenue than any other department for nearly every hospital examined (Merritt Hawkins, 2019; MiraMed, 2019; Surgical Directions, 2021). Due to this criticality, all system functions in the perioperative department have higher visibility. Even the smallest improvements can have extreme economic implications for its healthcare facility. Controlling this bottom line can help fund other hospital areas and contribute to its overall fiscal wellbeing.

Children's Hospital Colorado (CHCO) is no exception to these rules. This client-based capstone project applied analytic techniques and data science to explore their surgical procedure/case scheduling and operating room (OR) utilization. Using historical OR data, this paper documents the building of a programmatic tool which uncovered novel insights into OR usage. The tool compared trends between parameters like surgical specialties/departments, surgeons, weekdays, or ORs by aggregating temporal surgical case buildup leading up to the procedure date.

Even by comparing records from a limited dataset, the project uncovered key scheduling differences between surgeons, weekdays, and ORs, proving the need for additional investigation. Further insights into OR scheduling would help optimize profitability in this vital area for CHCO, thus improving overall patient quality of care.

Keywords: time series, hospital, operating room, utilization, optimization,

Table of Contents

List of Figures

Chapter 1 - Introduction

Children's Hospital Colorado (CHCO) is a pediatric teaching hospital located in Aurora, Colorado (southern Denver). From its humble beginnings as a former residence, Children's Colorado Association was officially incorporated in May of 1908 and treated nearly 300 patients in its first year (Children's Hospital Colorado, n.d.). Children's Hospital Colorado, as it is now known, services a region containing seven states and is the only resource with a comprehensive range of specialized surgical capabilities within six of those states. However, patients hail from all parts of the globe seeking care. These specialized pediatric surgical capabilities include cardiothoracic surgery, neurosurgery, ophthalmology, orthopedic surgery, otolaryngology, plastic surgery, pediatric surgery, urology, transplant surgery, burn surgery, and trauma (The Regents of the University of Colorado, 2022). CHCO averages nearly 15,000 surgical procedures per year, roughly one-third being inpatient services (leaving the remaining two-thirds as outpatient). It has been top-ranked in pediatric specialties for more than 20 years (Olmsted, 2021).

Support for the Study

Operating costs incurred from daily business expenses at hospitals can vary extensively. Despite this variation, the cost of maintaining an operating room (OR) for any given healthcare facility ranks among the highest. National and surgical healthcare expenditures estimate that nearly one-third of all healthcare spending is surgical care (Muñoz E., 2010), with OR time accounting for the second-highest cost category of that third (Stey, 2015). To give a more specific example of costs incurred, a study of 445 California healthcare facilities found the mean cost of OR time surpasses $37 per minute (Childers, 2018). In other words, the mean cost of delivering the service of surgery for those facilities was found to be $37.45 per minute for each hospital. As can be expected, OR costs differ based on many factors including, but not limited to location, discipline, and procedure (Macario, 2010). Though more research is needed for average cost factors like hospital type (e.g., teaching vs. nonteaching) and OR layout (hybrid vs. conventional), leading drivers ranked total personnel costs as most influential, followed by inventory and overhead (Patel, 2020). For these reasons, managing OR time efficiently is a critical task for any healthcare facility, including CHCO (Moazzez, 2016).

Because of the importance of time management surrounding OR scheduling, clinical software companies have addressed the need for such tools. They created stand-alone analytical packages or included them as enhanced features added to electronic medical record (EMR) patient-charting platforms. The latter is the case with CHCO; they employ a popular scheduling tool to manage OR time utilization. This application stores the schedule data in an extractable online analytical processing (OLAP) database. Hospitals can explore, understand, and optimize OR utilization management better by querying this database and building descriptive/predictive models.

Operating room time is divided into blocks and time slots, building out each room's scheduled procedures (or cases) for a given date. Surgeons can move from case to case and OR to OR throughout a date. This process involves many nuanced calculations and edge cases (Giridharadas, 2022). For example, rooms can be overbooked, cases can last longer than scheduled, ORs can be scheduled far in advance (> 1 year), less urgent procedures can be canceled or rescheduled (e.g., as an effect of COVID-19), the hospital has limited patient capacity, etc. Because of issues like these, there is often a gap between scheduled OR utilization and predicted OR utilization.

Problem

Children's Hospital Colorado employed an OR utilization scheduling tool to fill blocks of time with surgical procedures. Though they knew how blocks were scheduled, CHCO wanted to understand how efficient this process was. Predicting block utilization 30 days or earlier than a procedure date was crucial for understanding efficiency. One model for this predictive function already existed in the scheduling software itself. However, a goal of this project was to create a software tool at a more granular level (e.g., at the surgeon level). Using this tool, Perioperative management could recommend the next steps for hospital actions. If leadership decided the existing predictive model and utilization tool performed well enough together, no further action may be required. If they decided to improve the model, they could explore other options (e.g., vetting different scheduling packages).

Purpose

Because all scheduled surgical procedures at CHCO existed as auditable records, this study intended to leverage those records to understand the scheduling process better. By accessing this stored historical data, it was possible to view snapshots of individual OR utilization throughout time. The project then aggregated snapshots to understand how far prior to the procedure date they were scheduled.

For example, this OR utilization tool could explore questions about scheduling trends prior to procedure date: How likely was a procedure to be rescheduled if originally scheduled a year prior to the procedure date vs. a month prior? How efficient were procedure time estimates as compared to actual time estimates? What percent time utilization was an OR averaging? Which departments were most efficient? How often were time blocks refilled if a procedure was rescheduled one month before its original date – did it often sit empty?

The over-arching goal of this project was to learn from and answer questions like these. From this new-found knowledge, the project team’s goal was to construct predictive models so CHCO could better predict block utilization and optimize OR scheduling while accounting for constraints they faced. This project's primary hypothesis was that similar OR time blocks also had similar scheduling patterns. For example, if we knew the number of minutes scheduled for a particular OR on March 6, we could predict what it would be like on March 15 so leadership could make adjustments early to maximize occupancy.

CHCO could use the added revenue from this increased efficiency to update equipment or be reallocated into department personnel, thereby increasing overall hospital quality of care. CHCO employed a block utilization committee to explore OR time scheduling optimization and this project will provide key insights into this group's ongoing mission. By creating visuals of scheduling actions, like when particular OR time blocks fill up with procedures, this project will help provide greater understanding for everyone involved. Through this elevated understanding, procedures can be updated with best practices to optimize future OR utilization.

Exploratory Questions

This project sought to better understand the OR scheduling process through descriptive and predictive analytical techniques. Some questions this project set out to answer were:

1) Exploratory data analysis – What scheduling information did the hospital collect for surgical cases, and how did it relate to the workflow of the Perioperative group?
2) What typically happened with scheduled cases in an OR leading up to a particular surgical procedure date? What were typical case scheduling trends?
3) Develop an OR utilization tool for surgical management to observe trends on a more granular level than the EMR solution provided.
4) What next steps would the project team recommend for improving the existing OR scheduling process? Can the process be optimized? What were recommended next steps for the hospital to take?

Significance

This study hoped to explore and uncover key trends surrounding one of the most profitable services provided not only at CHCO but healthcare facilities in general. While third-party EMR companies may have conducted similar research to develop their platforms' scheduling functionality, it is not readily available to others. This client-based project will do a great deal to begin the process of optimizing this critical task at Children's Hospital Colorado.

Procedures

The study began by extracting historical surgical scheduling records from the OLAP data warehouse. The data was then cleaned and prepared for analysis. Once the team's data scientist/developer (Phil Callahan) prepared the dataset, a model was established over a window of past, completed surgical procedures. Then, the project built and aggregated time series data from this window to establish a predictive model for scheduling. Once this programmatic tool was developed, it could take several desired parameters and apply them to updated datasets to perform the analysis again on different subject combinations. Finally, the project compiled findings for a selected dataset and recommended subsequent actions to CHCO.

Definition of Terms

Action Date – A date when an action is timestamped in the surgical scheduling system (e.g., Scheduled, Canceled, Moved, Removed, Rescheduled, etc.)

Block – Reserved chunk of time when surgical procedures are scheduled. A typical time block might be 510 minutes (8.5 hours) on a weekday.

Children's Hospital Colorado (CHCO) - A pediatric teaching hospital located in Aurora, Colorado, in southern Denver and "The Client" of this capstone project.

Electronic Health Record (EHR) platform – See Electronic Medical Record; different term for the same platform.

Electronic Medical Record (EMR) platform – An online transaction processing (OLTP), patient-charting, software platform used in the healthcare industry to securely record and store patient data/results.

Health Insurance Portability and Accountability Act of 1996 (HIPAA) – A US federal law requiring the creation of national standards to protect sensitive patient health information from being disclosed without consent or knowledge (Centers for Disease Control and Prevention, 2018).

OLAP – Online analytical processing database.

OLTP – Online transactional processing database.

Operating Room (OR) – A space where surgical services/procedures are provided/performed.

Owner – The owner of a surgical time block is, for the purposes of this capstone, the head surgeon assigned to it.

Perioperative Group (Periop) – This term translates as "around the time" (peri-) "of surgery" (operative). It refers to the time the patient goes into the hospital for surgery to when the patient leaves for home (National Cancer Institute, n.d.).

Protected Healthcare Information (PHI) – Confidential, potentially identifiable, patient information protected by HIPAA.

Schedule/Procedure Date – The date a selected procedure of interest is scheduled to be performed.

Chapter 2 – Literature Review

Balancing cost management with incoming revenue is considered critical to every business. If left unchecked, no business would exist for long (Rust, 2002). Because of this, cost minimization is vital for an organization’s profitability. Healthcare providers are not immune to these economic principles either (Burrill, 2017). Indeed, supply items and time management have played key roles in any business’s bottom line, but recent findings suggest this is particularly true in the healthcare industry (Stey, 2015; Park, 2009). Many studies have looked at income-optimization techniques that hospitals can employ. Hospital cost optimization has experienced renewed interest with the advent of the COVID-19 pandemic beginning in early 2020 (Bai, 2020; Mazzaferro, 2020).

Even prior to the pandemic, many studies of this type found that providers' surgical service departments were among the most impactful because, while they brought in the most revenue, they also were among the highest cost to hospitals. Again, the COVID-19 pandemic amplified this cost-revenue relationship (Payerchin, 2022). Beginning in April of 2020, over 30 states had issued temporary executive orders requiring the cancelation of elective procedures to prioritize lower patient capacities and the safety of hospital staff (American College of Surgeons, 2020). For example, this cessation lasted for just under three months in Colorado, from March through May 2020 (Polis, 2020). Investigations from Bai et al. found that cancelations of elective surgeries and declining hospital visits due to COVID-19 were primary drivers for shrinking revenue from these services. This situation forced many hospitals to react with cost-saving measures (e.g., layoffs, salary cuts) as they began experiencing revenue drop-offs. With the OR occupying such a significant portion of a hospital's cost/revenue, it is a clear candidate for impactful initiatives.

Operating Room Cost to Hospitals

Industry studies have found operating rooms (ORs) to be one of the most costly departments within the healthcare industry (Babu, 2018). With such critical price points surrounding this area, one can see the importance of running an OR as efficiently as possible. At a cost to its hospital ranging anywhere from $15 to $100 per minute of operation (Childers, 2018), optimizing OR utilization is vital to this service. Industry documentation points out some reasons for the extreme variability in cost factors like procedure type, supplies, and staffing (Macario, 2010; Matityaho, 2019). For example, a heart transplant costs a hospital more than an appendectomy (Bentley, 2020; Malhotra, 2020) because of direct expenses like procedure/personnel time, complexity, and transplant cost (i.e., getting the heart from donor to patient). As one can imagine, these two procedures require distinctly different support from their facilities.

Childers et al. estimated the mean OR cost at $37 per minute, which reflected a generalized base rate for all direct-cost surgical factors. This price point did not distinguish between elements like personnel wages, location, and procedure type. However, it noted that indirect costs such as security and parking would be unlikely to change with time-efficiency adjustments (Childers, 2018). They estimated that, of the $37, about $16 was an indirect cost to the hospital, leaving $21 categorized as a direct cost. This split inferred that any attempt to control cost would be most impactful if aimed at the latter slice.

Operating Room Revenue Generation

Healthcare providers have realized that they were able to balance this high cost of operation (both meanings of the word are applicable here – operation the “act of surgery” or operation referring to the “functioning of business”) with the revenue stream ORs bring into the hospital. They found that a hospital’s surgical services provided approximately 42% of its revenue (Gillespie, 2011), more than any other department within the organization. Healthcare facilities often leveraged revenue brought in by the OR to subsidize other, less-profitable areas of the hospital like research, infectious disease, and mental health (Hegji, 2007).

Operating Room Utilization

These reasons mentioned above have motivated several investigations into OR utilization to optimize hospitals' survival and prosperity. Aside from healthcare providers themselves, EMR software developers have shown interest in OR time optimization since they service these providers as clients.

To minimize controllable cost and maximize revenue, OR utilization time efficiency has been an industry focus as of late (Muñoz, 2010; Patel, 2020). Most ORs have employed scheduling software for filling out surgical dates in advance. However, because of the dynamic nature of surgery (e.g., inpatient/outpatient, emergency), the accuracy of such schedules was known to vary widely between service types. The literature pointed to factors such as accuracy of case duration projections, unplanned procedures, and procedural complications, all playing significant roles in scheduling accuracy (Kayis, 2012). Because of the uncertain nature surrounding surgical case scheduling, healthcare providers have faced the difficult challenge of accurately predicting OR caseloads. On the one hand, they wanted to keep their OR time blocks full enough to maximize capacity (thereby optimizing revenue and cost) while accommodating unplanned add-ons like emergency procedures. On the other hand, they did not want ORs to sit empty by estimating too conservatively.

Optimizing OR Scheduling Using More Focused Data

Most current research on this topic has trained OR scheduling models on broadly generalized factors. That meant studies combined things like OR types and locations into the same predictive models. This approach was justifiable since they created those studies to inform either the public (i.e., published journal article) or a client (i.e., EMR internal documentation). For them, it made sense to report findings using a broad range of surgical factors. For their purpose, their models' functions applied to as many healthcare facilities as possible. The trade-off of this approach was that those models were less accurate than they would have been if trained on a narrower dataset specific to a single factor, like a particular surgeon (Clemen, 1989).

For example, literature about a predictive model trained on many hospitals in California may not be as accurate at predicting OR utilization in Colorado. Similarly, literature using a predictive model trained on gastroenterology OR utilization may not be as accurate when applied to neurosurgery ORs. Because of that variability between factors, this capstone's goal was to create a modeling tool using data specific to CHCO (higher variance but lower bias) and compare performance to models trained on broader datasets (higher bias but lower variance). This capstone applied a similar methodology to industry literature even though their scope was more generalized (Kayis, 2012; Tiwari, 2014).

Literature with Similar Methodology

In a 2014 study, Tiwari et al. developed a model to predict caseloads for a given surgical date. They trained their modeling tool on 146 days of historical data, each with a 30-day lookback snapshot. For example, a surgical date of 17Mar would observe a window of 30 days prior (t-1 to t-30 days) to see how many surgical cases a hospital scheduled leading up to the selected procedure date (17Mar). They repeated this process for each of the 146 days in the selected historical dataset. From this data, they were able to test several models and establish a predictive tool to improve scheduling surrounding ORs at both Vanderbilt University Hospital and Monroe Carell Jr. Children's Hospital. The study found that, while there was an uptick in the number of cases added two days before a scheduled surgical date, the overall trend was linear. Their tool moved OR utilization predictability out from a couple of days to weeks by leveraging this insight. This predictability improvement allowed planners the freedom to flex off staff further in advance of low-volume days while pre-emptively spotting under-staffed, high-volume days and adjusting personnel accordingly.

The methodology in the Tiwari study was not all that different from CHCO's EMR predictive scheduling snapshot. By isolating data with factors like specific surgeons in specific ORs, they were able to observe predictive trends. When looking back over a chosen window of historical data (> 6 months), they found these trends changed based on other included factors (e.g., day of the week). This capstone project set out to use similar predictive techniques to Tiwari et al. but create a predictive model tool projecting OR utilization in a time block percentage (rather than caseload).

Chapter 3 – Methodology

The methodology for this project followed similar basic data science road maps. The team began by extracting raw data from the EMR OLAP database and cleaned it for analysis. Once prepped, the team's data scientist/developer (Phil Callahan) developed code to first analyze a simplified unit of one temporal case buildup (for a selected surgeon, schedule date, and OR) as can be seen in Figure 7. Once that was functional, he aggregated those results to repeat for every date falling on that weekday (e.g., every Monday in the data) with that surgeon/OR combination (Figure 8). From the resulting aggregated time series, he was able to develop code to build the mean trend line including confidence banding (Figure 10). This served as the predictive model for that combination of parameters (surgeon, weekday, OR).

Considerations Prior to Analysis

Before beginning any data analysis originating from patients, or protected health information (PHI), everyone involved with this project adhered to strict securities already in place at CHCO to ensure they kept data on-premises at all times. The team took additional actions to ensure they did not expose identifiable patient and procedural factors during results and analysis reporting. Considerations like these were essential to remaining HIPPA-compliant throughout the project.

The Health Insurance Portability and Accountability Act (HIPAA) contains what is referred to as *The Standards for Privacy of Individually Identifiable Health Information* ("Privacy Rule"). This rule established for the first time, a collection of national protective standards for particular health information (HHS, 2003). One of the top goals of this Privacy Rule was to ensure individuals' health information remained private and protected while promoting high quality healthcare and protected the public's health and wellbeing. This project aligned with those goals since it was helping to increase OR utilization of a major healthcare provider in CHCO, while keeping PHI safe and secure.

With these considerations in mind, approximately six months before the start of this project, the team's data scientist began paperwork necessary to work with CHCO in a temporary capacity. Once the University of Wisconsin – Green Bay and CHCO established legal parameters, the IT department assembled a virtual desktop where all their data analysis took place. From there, the project lead, Dr. Greer, worked with IT to install necessary approved software packages through the hospital's Software Center installation portal to setup a suitable work environment. The project team chose Python to extract, modify, and establish time series data from raw SQL datasets because of familiarity with the language amongst key team members. Again, because of established pathways at CHCO, the team chose to run Python in Jupyter Notebooks. When approaching a client-based project with tight timelines and security considerations, many of the tool choices (i.e., software titles and work environments) are dependent on what the client has vetted and incorporated into their system. For sharing between project team members, the team used the open-source version control system Git. The team configured Git to wipe the output of notebooks before committing using the Python utility *nbstripout*. Using this utility, the receiving team member was able to run the code and reconstruct the results only if they had access to the same raw data. This stop-gap allowed sharing of the coded Python tool but kept the results securely available only to those with access to the raw dataset.

The project utilized other standard titles ranging from Microsoft Excel for initial data exploration to Notepad++ for script modification in the initial environment to Windows Terminal for Git commits and pushes. Excel was chosen to generate hypothetical data visuals (e.g., Figure 1 and 7) because of its ease of use and ubiquitous availability. The data could also be inputted much easier into the Python utilization tool when actual output (using hypothetical data) was needed (e.g., Figures 2-6, 8-10). That's why Excel was chosen over other spreadsheet titles (e.g., Google Sheets).

Data Extraction

As part of the initial data extraction, project members needed to meet with several key stakeholders and subject matter experts (SMEs) to learn basic terminology and become familiar with complex nuances involved with scheduling OR usage at CHCO. These meetings included

Perioperative (Periop) departmental leadership which oversaw OR scheduling to answer initial questions ranging from number of rooms to typical surgeon hours of operation.

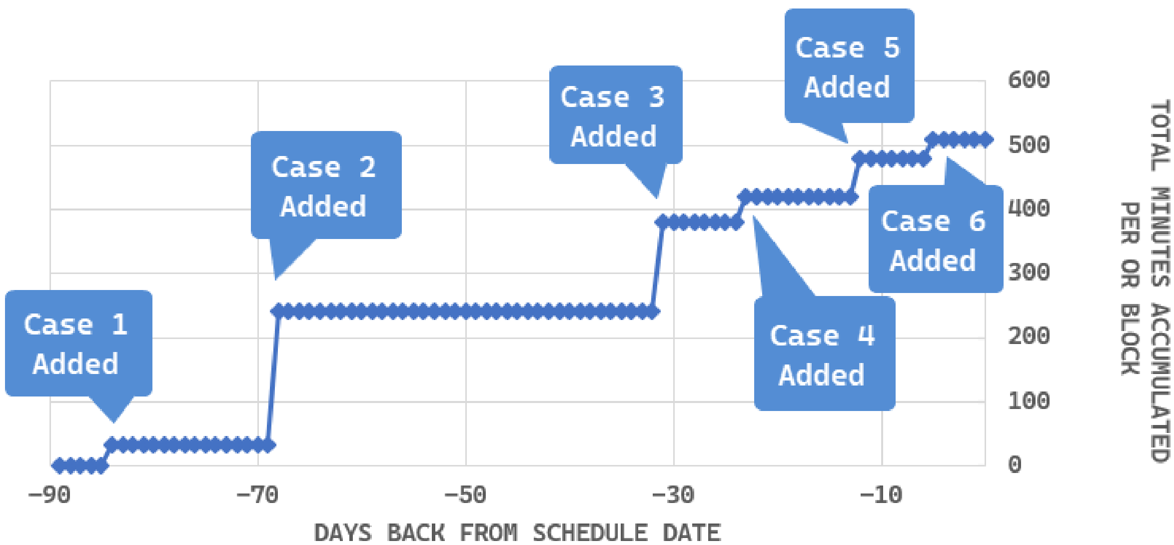
The project team also met with CHCO's EMR data reporting analyst for the actual extraction of historical data to be analyzed. EMR (sometimes referred to as electronic health record, or EHR) platforms utilize online transaction processing (OLTP) technology not unlike companies in the retail sector (El-Sayed, 2021; IBM, 2022). Instead of retail transactions, healthcare facilities use OLTP to process patient vitals and results which are stored in a short-term database and backed up into a secure online analytical database (OLAP) daily, typically running on Microsoft SQL Server or Oracle database software. This second type of system is where our project extracted historical OR utilization data for analysis. After several iterations to procure useful OR information, the team began exploratory data analysis on two datasets.

Iterative Exploratory Data Analysis

To gain insight, the team first isolated what was estimated to be one of the least complex surgeon's schedules (complexity from the perspective of scheduling, certainly not procedurally). From this scheduling data, the project team was able to perform an initial exploratory analysis and formulate further questions to iterate back and forth with subject matter experts. With the initial dataset, using one surgeon, the data scientist was able to create a "lookback" window from a single selected surgical date. As the Tiwari et al. study did, this capstone chose a window of time to "look back" from a chosen "schedule date" ("schedule date" and "procedure date" are used interchangeably throughout this paper). This lookback window served as a snapshot in time to see when each procedure (assigned procedure/case numbers for differentiation) was added, removed, or canceled from the chosen schedule date. The result was typically a linear, step-like graph progressing towards 80 - 100% utilization (Figure 1). As with the Tiwari study, this investigation expected to find steeper curves closer to the schedule date. Recall that because of the dynamic nature of scheduling surgical procedures, it is difficult to optimize this lead-up to a surgical date. Even (or especially) if a procedure is scheduled farther out (earlier), many variables can cause that procedure to be rescheduled or canceled.

Figure 1

90-Day Lookback Time Series Chart



Note. Example using fabricated data. It is representative of a 90-day lookback time series chart leading up to a selected procedure schedule date (day 0) for a single selected surgeon and OR. Also note that this example uses minutes, but final results were displayed in % utilization.

To iterate this example, imagine a surgical department has a schedule date of 02May2022 with surgeon A. This date has a "time block" reserved in OR1111 of 510 minutes (8.5 hours). This department has projected data surrounding average surgical times for each procedure it performs. Using EMR scheduling software and these average procedure times, schedulers can estimate how many, and what types of procedures to schedule in that OR on that schedule date.

The goal of this capstone was to look back at historical data, observe when these different procedures were added (or removed), tally the minutes, and expose any emerging trends.

Data Prep – Cleaning and Wrangling

With this roadmap in mind, the project team settled on two separate datasets to perform the analysis – surgeon data and snapshot data. A snapshot query pulled historical data associated with each case and its respective action events (e.g., when it was first scheduled, if/when it was moved, removed, or canceled, etc). Because of the way the data was arranged, these two datasets allowed for strategic modification to acquire necessary results to create historical time series trends. CHCO's EMR data reporting analyst provided the raw OR datasets as Microsoft Excel files. These files worked well for initial exploration, but as the team's data scientist began extracting data from them in the analysis environments, he noticed slow performance. To optimize run times, he converted each of the two datasets to comma separated values (.csv) text files. This approach came with tradeoffs - it eliminated much of the formatting but decreased run time for the import of the larger dataset (snapshot) dramatically. The snapshot raw dataset had nearly 800 K records (172,945) and the surgeon dataset had over 17 K (17,486). To reestablish the lost formatting, the data scientist implemented code to restore necessary attributes like converting date strings back to datetime objects, useful for filtering chronologically later in the analysis. The much shorter data import times made the tradeoffs of this code implementation a small price to pay especially if this code were to be used on larger datasets in the future.

Further optimizing performance, the data scientist cleaned and filtered the raw csv datasets to eliminate unnecessary information and minimize the data each subsequent computation needed to hold in memory. After this process, the surgeon raw data had the following fields remaining:

- **redacted_surgn_col_1**: Unique, identifying number assigned to each procedure/case scheduled to be performed. Note that the procedure may ultimately have been canceled or removed and never rescheduled, in which case, the unique identifier still remained.
- **redacted_surgn_col_2**: Date where the procedure ultimately terminated. If it was performed, the procedure took place on this date. If it was canceled, this was the date last scheduled before cancelation.
- **redacted_surgn_col_3:** This is a date where an action took place on a scheduled procedure. Each of these would be represented as one of the steps/plateaus in Figure 1.
- **redacted_surgn_col_4:** An action could include cancelation of a procedure, moving a scheduled procedure to a new time, or removing from one scheduled date to reschedule on another.
- **redacted_surgn_col_5:** This column was populated if a procedure was rescheduled, canceled, or removed. As long as the procedure was previously scheduled, it would have come *from* some date. Situations where this would not have been populated were when a procedure was being scheduled in the system for the first time (i.e., it didn't have a previous date) or the action that took place was to complete a procedure record.
- **redacted_surgn_col_6:** This field was the opposite of the redacted_surgn_col_5 – it was the *scheduled date* the procedure was being scheduled "to," timestamped by the corresponding date an action took place.
- **redacted_surgn_col_7:** Similar to the redacted_surgn_col_5 & 6, this field was concerned with the *location* the procedure was coming "from," on the corresponding action date. Like

the redacted_surgn_col_5, this field would not be populated if it were being scheduled for the first time or being rescheduled after being removed.

- **redacted_surgn_col_8:** Similar to the redacted_surgn_col_5 & 6, this field was concerned with the *location* the procedure was being moved "to," on the corresponding action date. Like the redacted_surgn_col_6, it would not be populated if a procedure were being removed or canceled.
- **redacted_surgn_col_9:** This was a free form field used to provide more information if needed (e.g., why a procedure was removed or canceled).
- **redacted_surgn_col_10:** This field identified the head surgeon on the team performing the procedure.
- **redacted_surgn_col_12:** This field contained the estimate of the typical elapsed time (in minutes) this procedure has taken to complete in the past.

The data scientist abbreviated the snapshot raw data down to the following fields:

- **redacted_snap_col_1**: This field provided unique strings that categorized a row's purpose. These were used for further filtering records later in the process.
- **redacted_snap_col_2:** This date was used to correspond with the filtered surgeon data frame schedule date.
- **redacted_snap_col_3:** This unique identifier for ORs was used to key off redacted_surgn_col_7 & 8 in the filtered surgeon data frame.
- **redacted_snap_col_4:** This was used to identify certain record behaviors similar to the redacted_snap_col_1 field.

- **redacted_snap_col_5:** This field was used if filtering by department.
- **redacted_snap_col_6:** Corresponded with the surgeon data frame's procedure number.
- **redacted_snap_col_7:** This was the time in minutes a procedure *actually* took to perform. These times included procedure time and turnover time (minutes between procedures to prep for the next one).

After filtering these raw datasets down to smaller data frames using Python v3.8.3 inside a Jupyter Notebook v6.0.3 and Anaconda Navigator v1.9.12, the data scientist wrote code to prepare the data for analysis. This was also a critical point to understand what fields were unpopulated and why. Was it necessary to impute results into unpopulated fields? What were the reasons for unpopulated fields? Most of the specifics were covered above in the definition for each field.

Initial Observations

As the Tiwari study had done, one of this investigation's datasets was a collection of snapshots for a given date range. Ranging back over a year's time, this capstone's dataset contained 332 unique surgical dates where surgeons performed procedures. The data scientist found this number by filtering the surgeon data frame to only completed case actions. A surgeon was able to mark a procedure as completed up to 30 days after the schedule date. Taking this step ensured the procedure was actually performed and not canceled or rescheduled. The data scientist then removed duplicates from this list of filtered procedure dates. This result was the number of unique dates where procedures took place (332).

Moving to the snapshot data frame, he isolated the total procedures performed within the date range (10,947), and divided by the dates that surgeries were performed (332), which gave an average of approximately 33 cases per day for the particular datasets.

Modeling Approach

The Tiwari study explored alternative volume prediction modeling at this point. Because of time constraints that exploration was beyond the scope of this project. The team decided to go straight into Tiwari's second method of accumulating surgical case schedules and look for signals to predict OR volume weeks before their actual schedule date. As the purpose of this project was exploring the need for a data science action, where Tiwari used a machine learning statistical software platform (i.e., Tiwari leveraged IBM SPSS v. 21), this capstone utilized Python integrated with various open-source libraries for a more manual approach.

Temporal Buildup of Case Schedules

After initial exploratory data analysis, the data scientist developed the code that created the temporal buildup of OR schedules to predict final case volume expressed as percent utilization of the total. First, the team chose to focus on building out the lookback functionality of the analysis tool with one simplified set of criteria – a surgeon, date, and OR number. Once that was working, it was a matter of adding filterable attributes and running the same lookback function repeatedly to create a robust average of multiple attributes (e.g., multiple dates for instance).

1) Surgeon, Schedule Date, OR Lookback Function

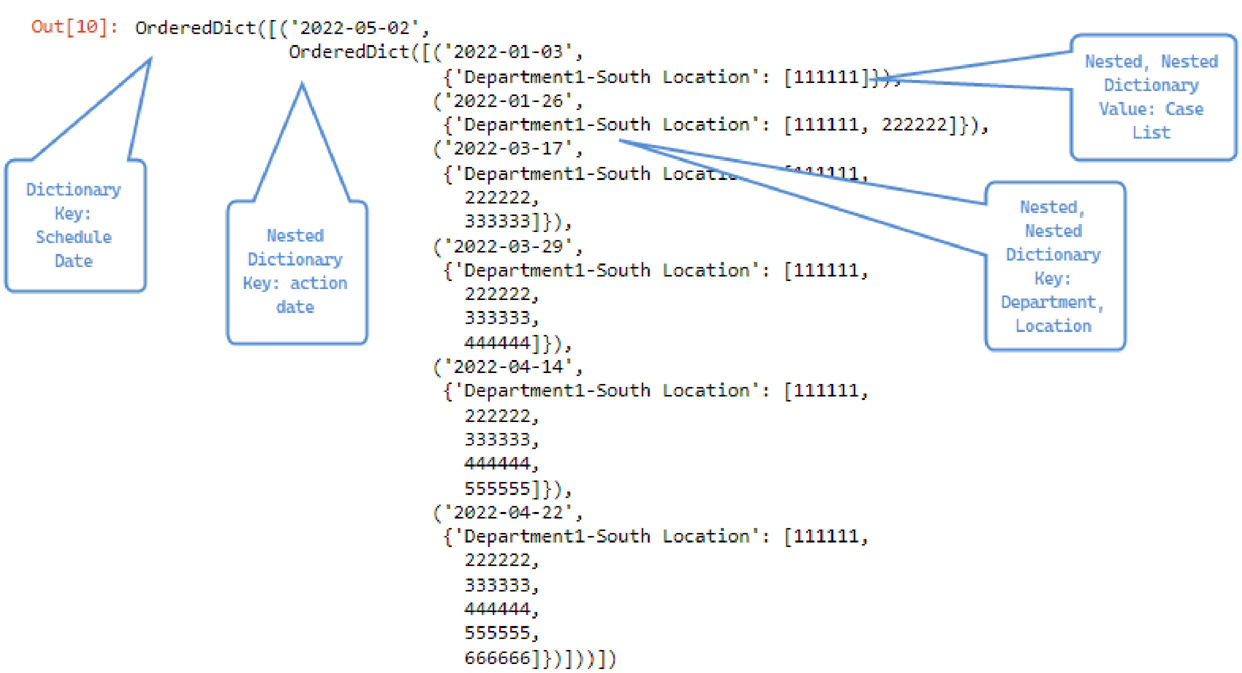
If given a data frame with only one surgeon, one date, and one OR, the data scientist coded a function (*makeCaseDict* in Appendix A) which output a dictionary that, when plotted, produced a graph similar to Figure 1. The capstone team had to make some choices about how to approach this critical data structure. Early on, they decided to use an Ordered Dictionary file structure because common Python dictionaries are unordered. This meant a common dictionary rearranged its structure as each new key (action date) was added. Maintaining an action date's order was critical to the purpose of this function. Because action dates were used as keys to sub-dictionaries which held accumulated lists of procedure numbers, it was imperative they stay arranged correctly. Figure 2 shows the data structure using hypothetical data and what a set of nested dictionaries would look like to create Figure 1.

This is a good time to mention that this data structure included all dates associated with this composite key combination (surgeon, schedule date, and OR). For example, if a procedure were scheduled 365 days before its schedule date, it would appear in this structure. Later in the process, the data scientist created a helper function to tally all dates prior to an input of n days, creating the lookback window seen in Figure 1. In Figure 1 and 2's hypothetical example, no case actions existed further out than 85 days prior to the schedule date (in other words, at 85 days prior, 03Jan2022 was the earliest case action for the schedule date of 02May2022).

To further iterate this concept, one could imagine a scenario in which all 510 minutes were created over 90 days prior to the procedure date. If this were the case with Figure 1, the temporal case minutes would have gotten built up (and tallied) before the lookback window and would appear as a horizontal line (at y = 510 mins) throughout the graph. To find the buildup "steps," a user would have to expand the lookback window for this scenario.

Figure 2

Sequence of Nested Dictionaries



Note. *Sequence of nested dictionaries representing time series of one combination of surgeon, schedule date, and OR. The deepest nested key/value pairs create the temporal case buildup – action dates and case numbers are associated with each.*

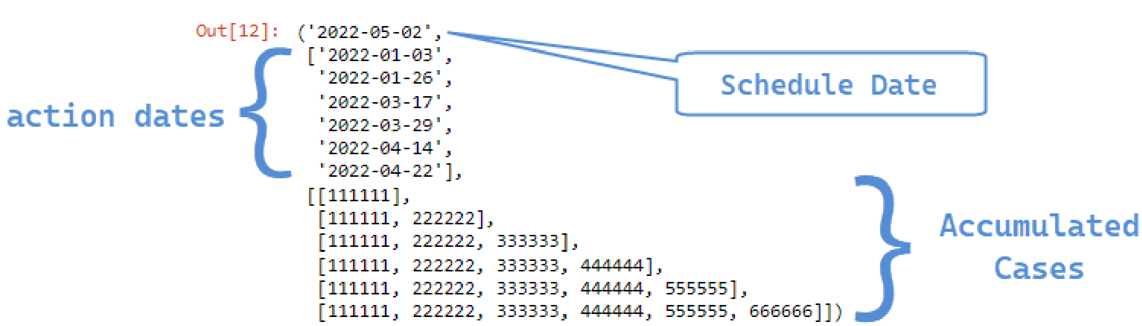
It is also important to mention that the simple example scenario in Figure 1 and 2 does not include any removals or cancelations. In the real world, surgical scheduling was quite dynamic and included many removals, reschedules, and cancelations. A case can even be removed and put back in the same schedule date several times. This was reflected in time series appearing as a "step down," in which new cases continued upwards from there. Additionally, a case can be moved back and forth from OR locations but was reflected as a removal and reschedule in this capstone's data structure.

2) Preparing Dictionary for Plotting

Once the lookback data structure was established, under these narrow conditions, the data scientist created a function to prepare the dictionary for plotting by separating it into lists (*makeDictPlottable* in Appendix A). When the resultant dictionary from the *makeCaseDict* function was passed into the *makeDictPlottable* function, it returned three variables: the schedule date, the action dates where something actionable happened to procedures on the schedule date, and a running case tally list for each action date. These three variables can be seen for our hypothetical example in Figure 3. It should be noted here that cases which were scheduled for the procedure date at one point, but were ultimately not run (e.g., cancelations) were not included in this these resultant variables at this point because the snapshot data only included cases that were

Figure 3

Three Variables Used to Construct Temporal Case Buildup



Note. *Three variables returned for the hypothetical example from Figure 1: 1) the schedule date, 2) action dates (where something actionable happened to a case set for the schedule date), and 3) a tallied list of cases resulting from each action date.*

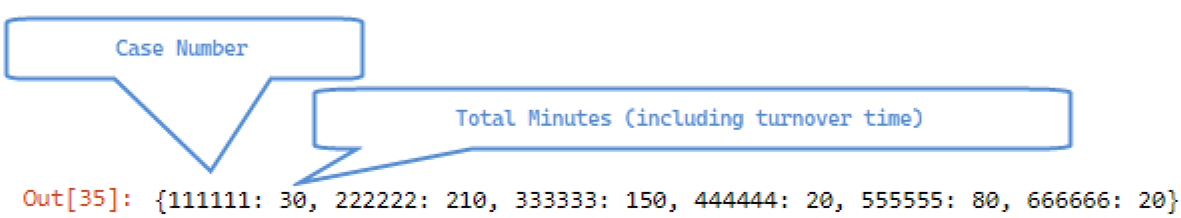
performed. Because this investigation was interested in the effect these cases had on a final model, they needed another function to include them in this accumulated list (see Step 4).

3) Getting Case Minutes

Switching tracks slightly, after the utilization tool had an ordered dictionary created with a schedule date, audit dates, and a list of lists accumulating procedure numbers (Figure 3), the data scientist developed a function to build a dictionary of procedure numbers and total minutes each procedure lasted on a selected schedule date (*blockDict_per_OR_and_date* in Appendix A). Now that the OR utilization tool had the list of accumulating cases for a schedule date, the data scientist could find the minutes from each procedure and store them in another dictionary. For this purpose, storage order was unimportant. The output for this report's hypothetical example would look like Figure 4. Aside from the dictionary, this function also computed the numerator and denominator to be used for the percentage-of-total calculation downstream in the analysis.

Figure 4

Dictionary of Case Minutes



Note. The total minutes for each case/procedure are reflected in each value. To get the total for the date (block) they would need to be tallied.

4) Including Cancelations

To add to the case dictionary above (Figure 4), the capstone team's data scientist developed a function to get cases that were included in the surgeon data for the schedule date, but not included in the snapshot data (*includeCancelations* in Appendix A). Remember, the snapshot data only included cases that were performed, if a procedure was canceled it didn't show up in this list. A quick filter of the surgeon data fields found 79 such cases in the dataset. While representing under one percent of the total cases performed in the date range, the team decided to include them for bookkeeping purposes downstream in the analysis.

For the minutes the canceled procedure ran, the project had no actual record (because the procedure was never performed), so the best option was to take the projected time in the surgeon dataset. While this was obviously less accurate than an empirical record of an actual procedure, leaving it in the model served the purpose for prediction when averaged together with other dates as if it *were* performed. The purpose of this investigation was not to predict likelihood of cancelations, though that could prove interesting future exploration.

5) Tallying Case Minutes

Since the project had the function to create a dictionary of total procedure lengths associated with each case as well as the temporal buildup of cases associated with a schedule date after the previous steps, the data scientist needed to add (or subtract in the occurrence of a removal, reschedule, or cancelation) a running tally of minutes associated with each procedure/case. This function (*tallyCaseMins* in Appendix A) passed in the accumulated list of procedure numbers (bottom list in Figure 3) and the case/minutes dictionary (Figure 4, except with cancelations included) and outputted a list of total minutes tallied for each action date along with the numerator to be used in the percent utilization calculation next. This can be seen in Figure 5.

Figure 5

List of Accumulating Case Minutes for Each Action Date

```
Out[36]: [30, 240, 390, 410, 490, 510]
```

Note: Each list item corresponds with the minutes y, of the chart in Figure 1 of our hypothetical example.

Since this investigation decided to bracket the lookback window starting at 1159 the night before the scheduled surgical date and moving reverse chronologically n days, the final number in this list of accumulated minutes served as the numerator in the final percent utilization calculation. The team decided that day-of-surgery activity was too dynamic and therefore out of scope. Little would be gained training a predictive model up until the minute of the procedure.

6) Converting Case Minutes to Percent Utilization

In the interest of obfuscating as many potential identifying factors as possible, the project decided to report final results as percent of OR utilization rather than minutes of procedure duration. Surgical procedure times had been well documented (Crespin, 2022) and presenting results in total percentage rather than minutes, added to the level of anonymization. Aside from aggregation which was performed later in the analysis, people viewing the results had no way of knowing the length of the procedure since the whole block of scheduled time was unknown. Aside from that, converting results to percentage also had an empirical advantage – the results were easier to compare with different sized blocks. For example, if a surgeon came in to perform a single procedure versus a full day of them, percentage provided an elegant way to compare utilization even when total block times were hours apart.

Figure 6

List of Accumulating Percentages for Each Action Date

```
Out[38]: [0.058823529411764705,
          0.47058823529411764,
          0.7647058823529411,
          0.803921568627451,
          0.9607843137254902,
          1.0]
```

Percentage gained at each action date

Note. *List of accumulating percentages for each action date of a selected schedule date (observe that each list item corresponded with the minutes y, of the chart in Figure 1 of our hypothetical example).*

Therefore, a function was created, *percentUtil* in Appendix A, to take the numerator (the final number of the output of the previous function *tallyCaseMins*) and divide it by the denominator (the total reserved block time). This structure can be seen in Figure 6 for the hypothetical example. Here we should note that the denominator could vary based on block activity. Sometimes surgeons ran two rooms, in which case two ORs needed to be summed. If a surgeon came in specially for one procedure, all the slot time might count as “outside time” in which case the only total value available for computation was the final minutes tallied. This scenario always resulted in 100% utilization because dividing a total by itself always equals 1 (e.g., (40+12)/52 = 1). Another situation this function guarded against was when the procedure was ultimately canceled and no projected time was provided in the surgeon dataset so the bookkeeping could get thrown off and result in a null value.

7) Creating a Lookback Window

Once the format of the tallied cases was correct, the data scientist needed to build the lookback window. Recall that the list of accumulated percentages was still every case tallied since the beginning of the dataset. Certain schedule dates running more predictable elective surgeries would likely be scheduled further before their schedule date than schedule dates/departments with more emergent surgery types (see Figure 15). This was where it was beneficial to establish an adjustable lookback window. Perhaps for some departments, the team would want to look back 180 days and some 90. After reviewing initial data patterns, the team ultimately settled on a standard 180-day lookback window for the results. Nevertheless, the function was built to be adjustable (*makeDaysbackDict* and *limitDaysback* in Appendix A). Again, choosing a standard lookback window allowed comparisons on the same field of view (x-axis in Figure 1) between all manner of factors (e.g., surgeons, day of week, etc.)

This section’s hypothetical example doesn’t have any accumulated procedures prior to its lookback window of 90 days because the accumulated minutes were at zero until -85 days before the schedule date; however, this does happen in real-world data (esp. if the lookback window is too narrow). To deal with data prior to the chosen lookback window, the minutes (or percent utilization) were tallied and the time series simply started at whatever the total was prior to the earliest lookback date (*limitDaysback* in Appendix A). This is why some time series started higher than zero (y > 0) at x = 0.

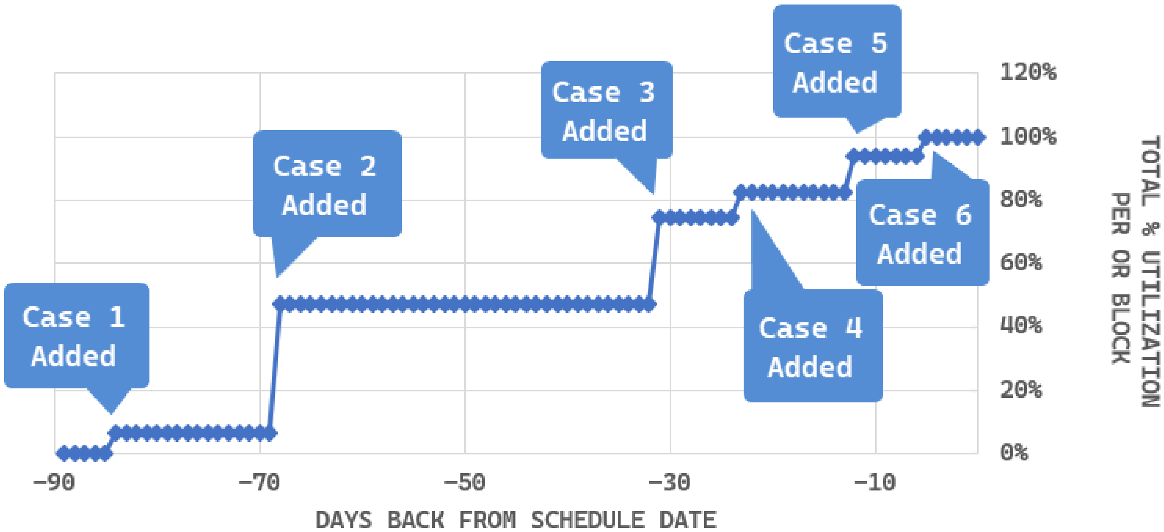
8) Plotting the Time Series Data

Finally, the utilization tool had all the functionality needed to plot one time series: the user had an ordered dictionary of the temporal buildup of scheduled cases/procedures and respective procedure lengths (formatted in percent utilization) for a selected date, surgeon, and OR combination. The last step before creating the code to repeat this process on multiple composite key combinations was to actually plot the time series (Figure 7). This was accomplished with a combination of functions. The first function created an ordered array of pairs of the selected number of lookback days and zero (*makeEmptyDict* in Appendix A). Since surgical blocks had varying numbers of procedures for any given schedule date – this report's hypothetical example had six but there could easily be 20 short procedures – the analysis needed this function to standardize where each case action happened. When the tool aggregated multiple procedure dates, action dates were critical for determining the predictive mean and confidence interval at a particular point in the time series model. Finding that structural relationship was the main point of this investigation – observing how closely to the schedule date procedures were usually scheduled or rescheduled (given differing criteria).

Next, the function *assemblePlotDict* (see Appendix A), populated the values in the empty dictionary using the collection of days back and the tallied % utilization values from previous functions.

Figure 7

Updated 90-Day Lookback Time Series Chart



Note. Same example of hypothetical 90-day lookback time series chart as Figure 1 except reported % utilization in y-axis rather than minutes.

Finally, the last step before aggregation, using the *plotUtil* function (see Appendix A) and the key-value pairs output from *assemblePlotDict*, the tool was able to plot one time series temporal buildup of cases prior to a schedule date. If using this report's hypothetical example data, it would look like Figure 7 (without the callout boxes indicating where each case was added).

Aggregation of Time Series Data

After the data scientist successfully implemented functionality to plot the time series of a single composite combination of criteria (surgeon, schedule date, and OR), he implemented code to repeat that analysis many times with aggregations of schedule dates. The team chose to investigate aggregation of weekdays hypothesizing they would find an observable difference amongst them. But first they needed to create the aggregate structures.

1) Aggregating All Surgeons in The Data

Using the same filtered data frames created from the two raw datasets, the team's data scientist built a function for finding all surgeons in the dataset (*getAllSurgns* in Appendix A). This simply output a list of all surgeons who had procedures in the data. Since this was not an exhaustive dataset of the entire hospital, it was intended to serve as a list of corner cases (some of the most complicated), this function produced only a list of six unique surgeons. CHCO employs over 20 specialized surgeons (Children's Hospital Colorado, 2022).

2) Getting a List of All Dates a Surgeon Worked

The next task was assembling a list of all dates a surgeon worked (*makeSurgnDict* in Appendix A). This function, called *getAllSurgns* to create a list, further filtered the data frame passed in, created a list of dates, then attached that list as a value in a dictionary. The resultant output from this function was a dictionary of aggregated lists of all schedule dates associated with each of the six surgeons.

3) Adding Day of Week (DOW) Into Aggregate

Because the team decided early in the investigation that they wanted to observe trending associated with weekdays, they chose to add a DOW column to both the *filterSurgnDF* and *filterSnapDF* functions. This function utilized the Pandas package's dt.day_name( ) native function and provided a day of week key for future filtering (weekends were not part of this project's scope).

To get a sense of blocks in the whole population prior to creating an aggregated time series graph, the data scientist implemented functionality for the utilization tool to create a data frame with each surgeon, DOW, OR, and block tally combination. Before eventually making a dictionary with the surgeon, DOW, schedule date, and OR composite key (with the number of blocks as values) the tool needed to create a few more helper functions.

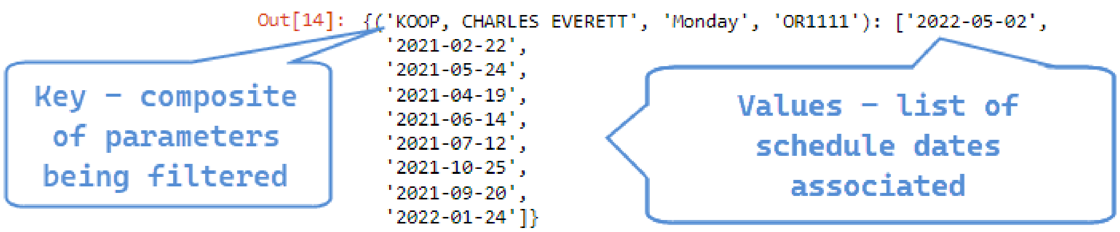
First, given a surgeon and schedule date, the data scientist built a function that returned a dictionary of each OR used on that date (*getSurgnORs* in Appendix A). Next, using *getSurgnORs*, he created a dictionary returning the list of ORs for each surgeon, DOW, schedule date combination (*makeSurgnDOWORdict* in Appendix A). Then, using that resulting dictionary, he made a list of unique values (i.e., unique ORs used for a given surgeon, DOW combination) using the function *getUniqueORs*. Combining these functions, he was able to refine a full population dictionary of schedule dates for each surgeon, DOW, and OR combination (*make_unspec_pop_dict, spec_pop_dict,* and *removeEmptyDFs* in Appendix A). With *tally_popDict_dates* he tallied the total number of dates in each list and created a data frame with *makeTalliedDictDF*.

4) Find Blocks Associated with a Given Surgeon, DOW, OR

Given the previous dictionary, the data scientist implemented functionality into the utilization tool that was able to find blocks associated with surgeon, DOW, OR combinations using *find_blocks* in Appendix A. Revisiting our hypothetical example for clarity, the block dictionary might look like Figure 8.

Figure 8

Aggregated Block Dictionary



Note. Hypothetical aggregated block dictionary. Note that the first date in the value list matches the hypothetical data in Figure 7. The remaining eight list items are hypothetical dates when imaginary Dr. Koop performed procedures.

5) Plotting Multiple Dates

From this point, the tool simply repeated the steps in the previous section (Temporal Buildup of Case Schedules) for each schedule date in the list from this dictionary. Using the function *agg_dates_to_plot* in Appendix A, the tool utilized several methodologies already discussed, to output an aggregated version of the cases, minutes, and denominator for each key composite. Adding the schedule date provided the extra layer of detail to the previous key combination of surgeon, DOW, and OR.

Using the function *agg_plot_dicts_per_date*, the tool created the temporal case buildup for each date, then combined with *ts_plots* built an ordered dictionary of xy values for each schedule date.

To prepare this resultant aggregate dictionary for plotting, the data scientist built the function *get_xy_from_ts_dict* to combine the y-values (% utilizations) at each lookback day using the x-value (days back) as the key. He chose the days back value as the key because it didn't vary from one schedule date to the next. He chose to store the lookback list as keys to their corresponding % utilization lists to keep the code more performant rather than repeating this list over and over since the lookback days didn't change.

With the multidimensional y-value array as the values for each point (lookback day), the tool was able to easily modify them to obtain the overall average time series trendline as well as the confidence interval for each point (*calc_ts_model* in Appendix A). An example of what the key-value pair would look like in this dictionary for the hypothetical data can be seen in Figure 9. A and B both represent the two ends of the lookback window. A is 90 days prior to the schedule date and B is one day prior (as can be seen in the keys). Note that the first and last list items in the values corresponds to the % utilization in Figure 7 (on day -90 it is 0%, on day -1 it is 100%). If one were to observe lookback days between these two, they would find that the days correspond to the data as well (the first list item would jump from 0 to 6% on day -85 and so on).

Note also that there are nine list items (values in the dictionary); these each correspond with the schedule dates in Figure 8. Only the first one is the imaginary data graphed in Figure 7; therefore, this dictionary represents nine such time series trendlines.

Figure 9

Aggregated Days Back Dictionary

A

```
Out[16]: OrderedDict([(-90,
                       [0,
                        0,
                        0.15833333333333333,
                        0.49583333333333335,
                        0,
                        0.25,
                        0.225,
                        0,
                        0.46458333333333335])])
```

<Repeats here 88 times in a 90-day lookback example>

B

```
Out[18]: OrderedDict([(-1,
                       [1.0,
                        0.93125,
                        0.8854166666666666,
                        0.8729166666666667,
                        0.925,
                        0.66875,
                        0.8145833333333333,
                        1.0,
                        0.8041666666666667])])
```

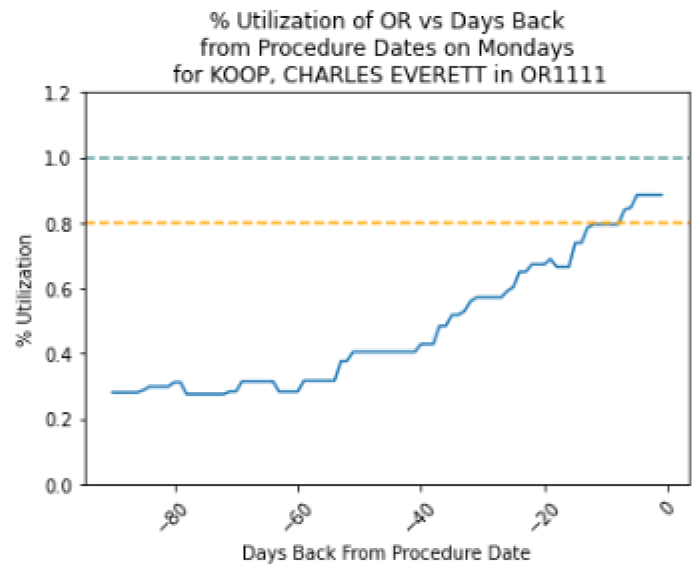
Note. *(A) represented 90 days back from the schedule date. Notice there were nine y list items in the value portion of this ordered dictionary, these each correspond with the aggregated dates in Figure 8. Remember, only the first list item corresponds with the hypothetical date graphed in Figure 1 and 7, but tracked with those values (it started out at zero and ended at 100% utilization). If one were to look at day -85 in this dictionary, one would see the value of the first list item jump from 0 to 6% because 30 minutes were added from case 111111, and so on for all the lookback days when an actionable event took place.*

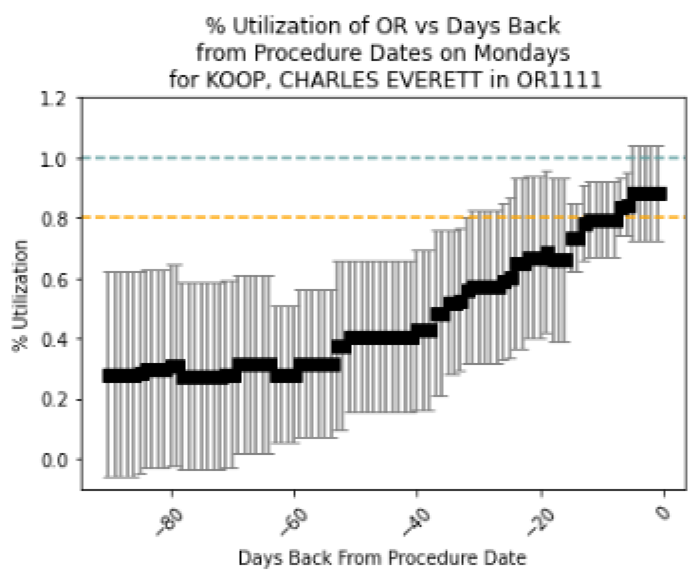
With *calc_ts_model* the aggregated values of each lookback day were averaged to get the mean time series predictive trendline using the Statistics package for Python. The same module was used to calculate one standard deviation for the confidence interval. The tool's function outputs these values as two lists: one of mean values for each point and the other as a list of ± one standard deviation from each point (Figure 10).

The Statistics package was chosen over manual calculations of mean and standard deviation because when compared in earlier code variations, the results were negligible (if not identical) and the package outperformed manual calculations for speed.

Figure 10

Final Hypothetical Time Series Model





Note. Graphs that represented the final temporal buildup of surgical cases as time series models for the hypothetical example. The top graph was the mean result of each point and the bottom was the same with added confidence intervals.

Chapter 4 – Initial Findings/Results

Children's Hospital Colorado began this investigation as an initiative to optimize one of its most visible departments – Perioperative Medicine (Periop). The original proposal was to validate CHCO's existing Periop scheduling solution. Working closely with the EMR software company during exploratory data analysis for this investigation, the collaboration uncovered several novel methods for extracting information to bring CHCO closer to its goal. Due to time constraints restricting this portion of the capstone project, the team decided to concentrate on the analysis of existing OR utilization trends at the owner (surgeon), weekday, and OR levels using the exploratory tool developed by the team's data scientist.

Results at the Owner, Day of Week, and OR Level

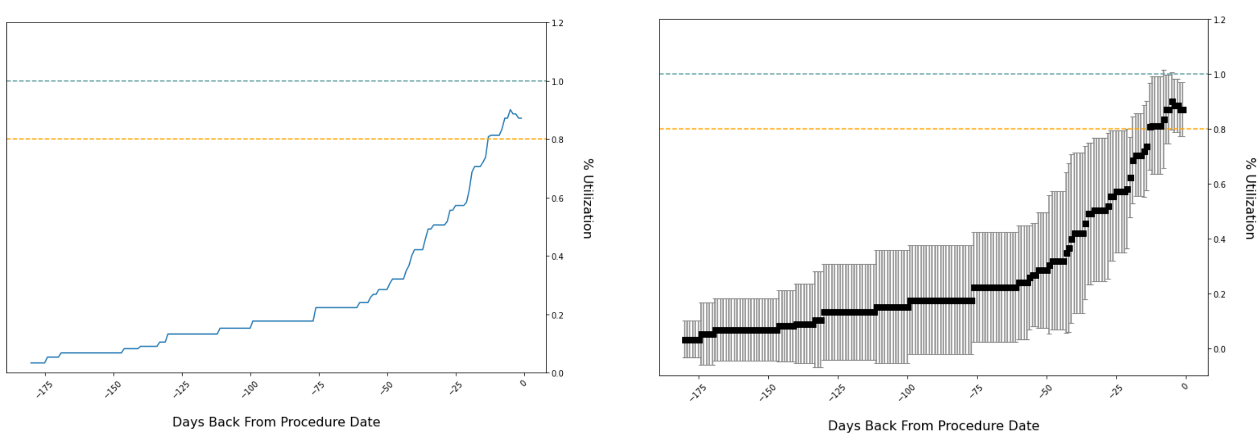
A typical practice for healthcare facilities is to shoot for 75 – 80% utilization of ORs. Capping surgical time at this usage level builds flexibility for about 20% of a scheduled time block designated as "open" time to fill emergent needs (Peters, 2017). The visuals in this section have target lines placed throughout to denote these optimal utilization levels (i.e., at 80% and 100%).

Results using the OR percent utilization tool developed for this project compared the time blocks on several hierarchical levels. This project's data scientist developed the tool following the methodology outlined in Chapter 3 of this report. He designed it to output a certain granularity of utilization information when provided with specific parameters – owner, weekday, and OR number. The typical hospital OR scheduling structure and the lessons learned from the initial exploratory data analysis also suggested that the tool should have an additional feature to toggle the granularity of its output information. The tool's developer incorporated this feature by allowing the two most granular input parameters to be optional – weekday and OR number. If a user left the OR number out of the input parameters, the tool would return all surgical blocks (schedule dates) owned by the inputted surgeon on the inputted weekday (see Figure 13 for blocks specified by owner and weekday). The user could roll up still another level by leaving out the weekday from the input parameters. This omission would return all the surgical blocks owned by the specified surgeon in the dataset (see Figure 15 for representative examples of the highest block granularity the tool provided).

As an example of this relationship, if Periop management wanted to see how efficient Surgeon A was on Mondays in OR1111, they would provide these parameters in the designated code block. This capstone performed that analysis for a selected surgeon/OR combination (kept anonymous) on Mondays. The results of this analysis are in Figure 11. This owner, OR, weekday combination had nine procedure dates (blocks) which the analysis overlayed in the figures. On the left was the average time series; on the right was the average time series with a confidence banding of one standard deviation out from each point.

Figure 11

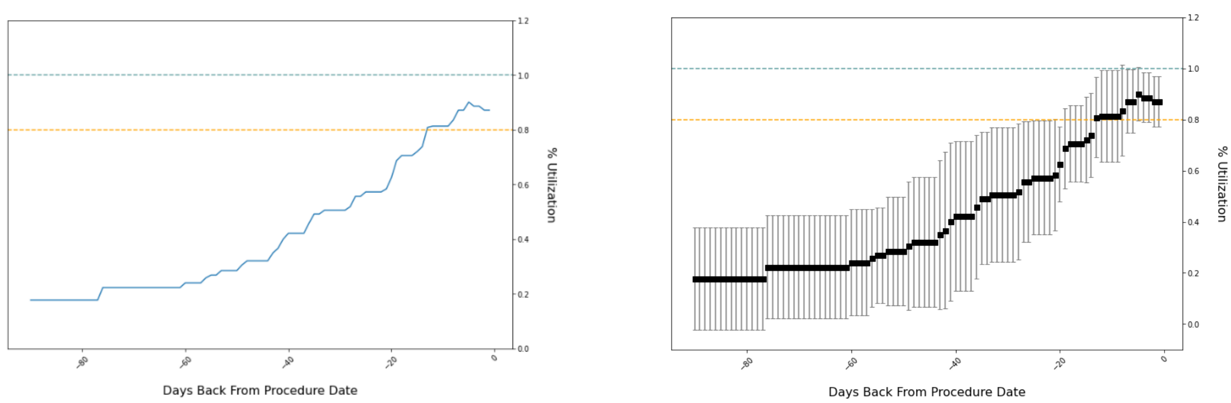
Example of Highest Level of Utilization Tool Granularity (180-day Lookback)



Several observations were seen from the two time series side by side. Firstly, note the linear nature of the trendline as it approached the procedure date. Also, note that these graphs had lookback dates of 180 days instead of the standard 90 days found in other similar scheduling studies. If the lookback window were only 90 days, the graph would be missing essential information – there was an uptick in caseload starting around 60 days prior to the procedure date. If this visualization were only looking in a 90-day window, the time series would appear more gradual and steady. Because of these variations between departments, surgeons, and ORs, the tool's parameter to control the lookback window allowed the ability to zoom in and out with a simple input adjustment. Figure 12 showed the same surgeon, weekday, OR combination with only a 90-day lookback. Note the incomplete picture of the data here. The uptick was much less pronounced.

Figure 12

Example of Highest Level of Utilization Tool Granularity (90-day Lookback)



Another characteristic visible in the time series was the narrowing of the confidence banding as it approached the schedule/procedure date while reaching maximum thickness in the middle. Again, this was much easier to observe in the 180-day lookback.

Comparisons at the Owner and Day of Week Level

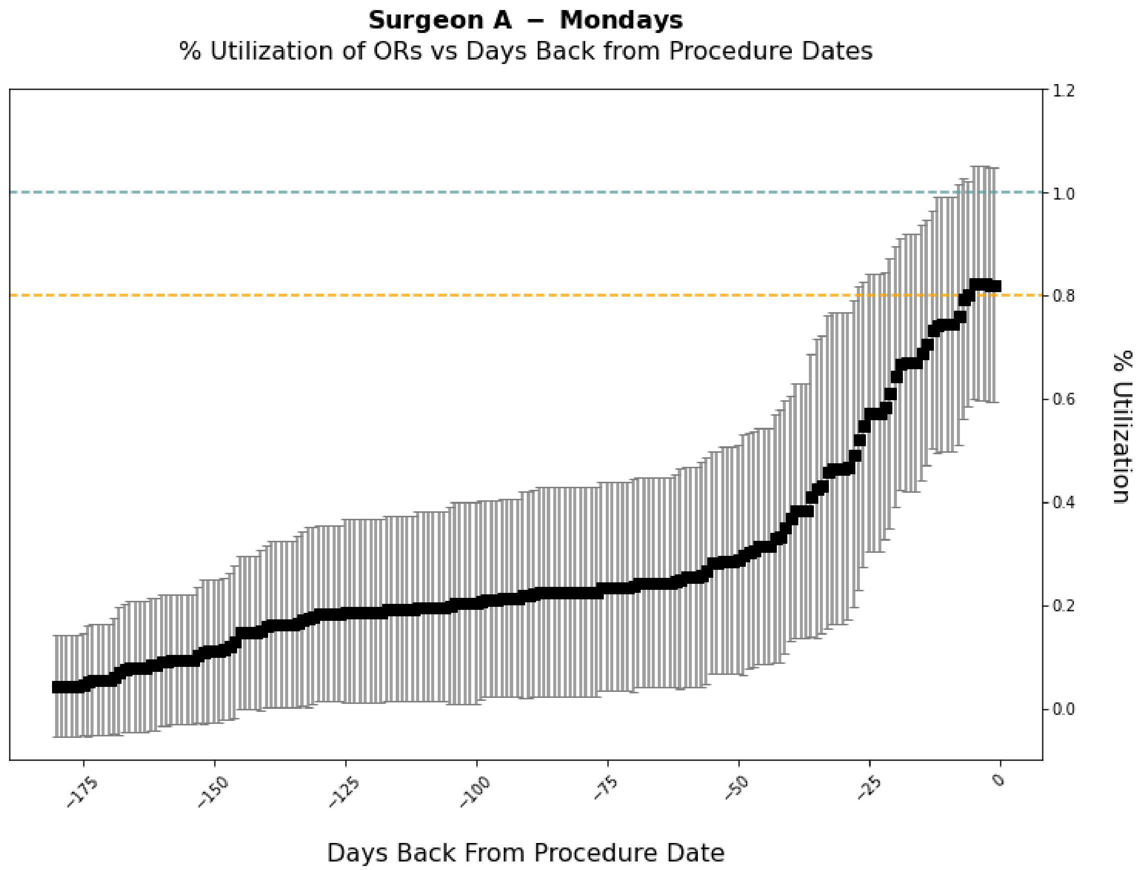
Recall that the project's data scientist built the OR utilization tool with roll-up and drill-down functionality. If Periop management wanted to roll up their view and investigate how efficient all of Surgeon A's blocks were for all Mondays in the dataset, they would omit the OR number (OR1111 in our example). This project performed such an analysis by running the previous surgeon through the tool for all Mondays (Figure 13). While the previous example displayed nine dates that the surgeon performed surgery in a particular OR, rolling up a level increased the dates to 46. This broader sample count meant the selected surgeon performed procedures on 46 Mondays in the dataset throughout various ORs. Figure 13 reflects how this increase in the hierarchy affected the surgeon's overall utilization curve.

Note the differences between Figure 11 and Figure 13 even though they were the same owner and weekday. Combining more procedure dates has altered some characteristics of its shape:

- The uptick in caseloads seems to have moved closer to the procedure date.
- The slight downtick still existed a few days before the procedure date.
- The trendline's confidence banding was now tapered only at the furthest points prior to the procedure date (rather than just before).
- And the overall percent utilization seemed slightly lower than the single room comparison (though right on the ideal target of 80%).

Figure 13

Example of One Surgeon % Utilization on All Mondays



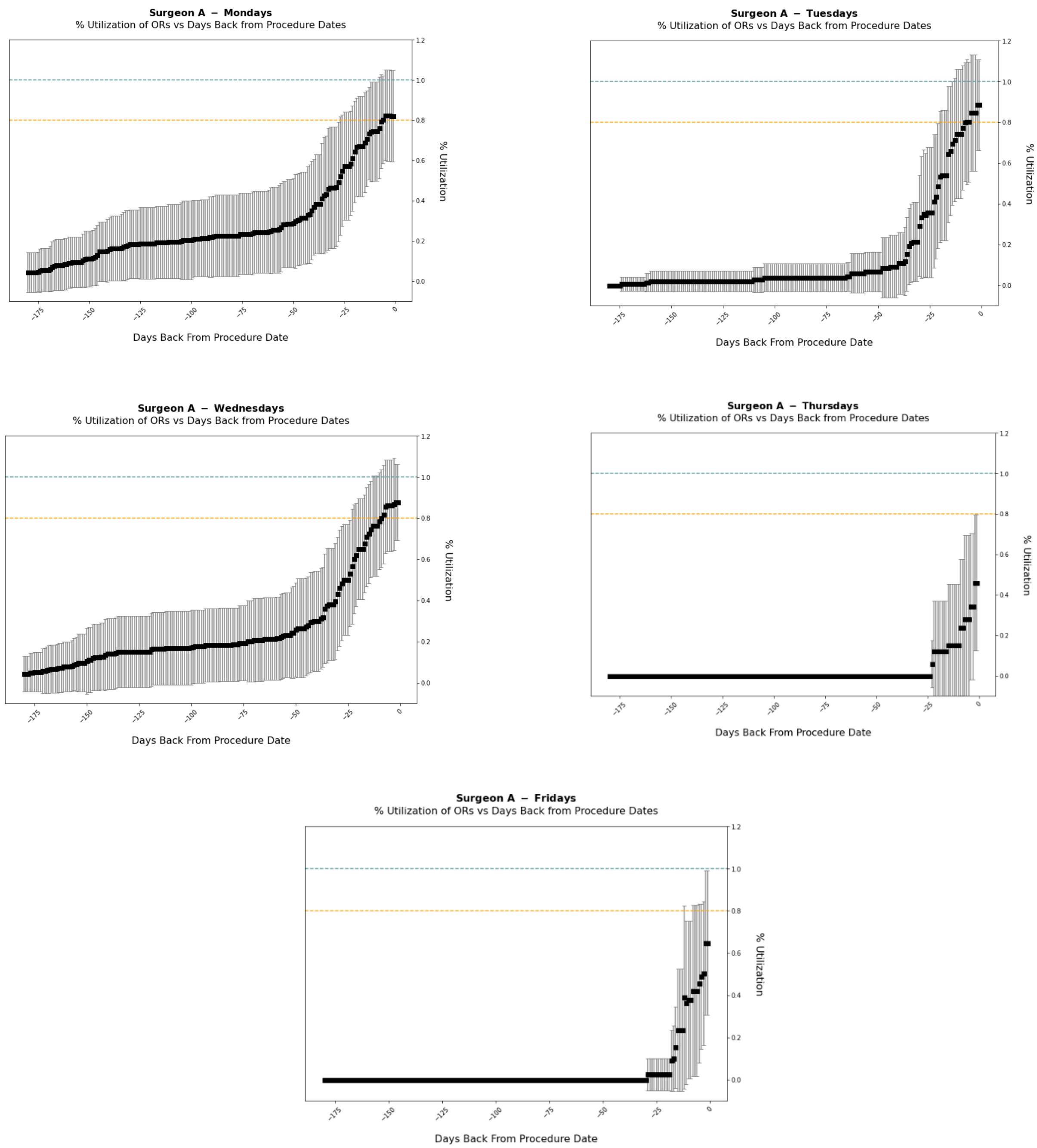
Comparisons/Contrasts Between Weekdays

When aggregate weekday time series were placed side by side for Surgeon A, trends emerged that were unique to each graph. Figure 14 shows that this surgeon had 46 Mondays, 22 Tuesdays, 57 Wednesdays, 5 Thursdays, and 11 Fridays with scheduled procedures. The data also showed a smooth, nearly identical curve to their Mondays and Wednesdays. Note where the uptick happened on Mondays and Wednesdays versus Thursdays and Fridays – it was nearly half as far out. This comparison showed Tuesdays falling almost directly between.

Those comparisons are examples of what the developer designed the utilization tool to uncover. When trends like this were exposed, it presented Periop management with the data needed to make informed decisions to interpret and optimize utilization.

Figure 14

Comparison of One Surgeon's % Utilization on Each Weekday



Comparisons/Contrasts Between Surgeons from Different Departments

Recall that the dataset used for this project contained six representative surgeons throughout CHCO. As part of the analysis run by the OR utilization tool built for this project, the analyst compared the time series of surgeons in different departments. In the interest of obfuscation, the team assigned the departments arbitrary designations: Departments I, II, and III.

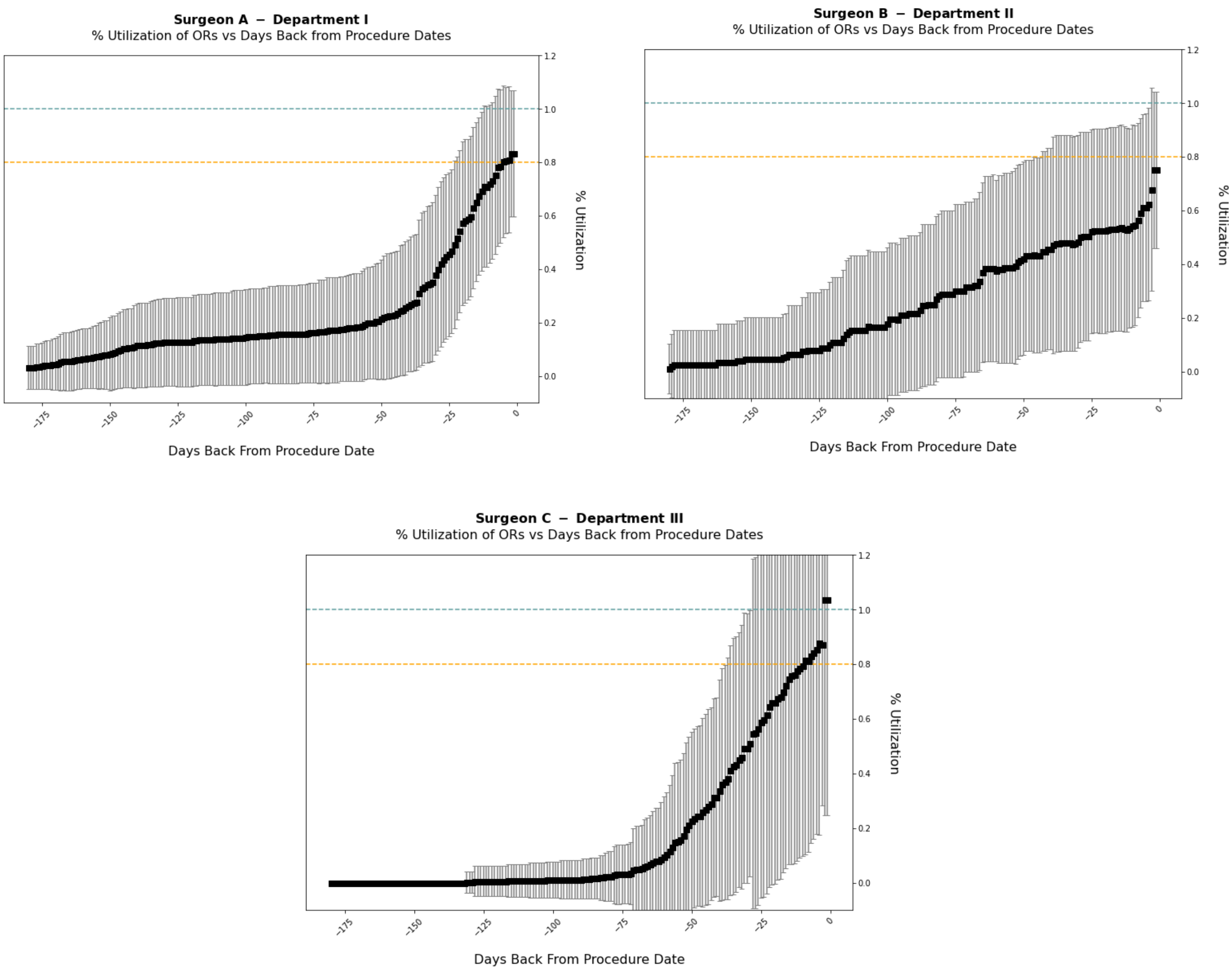
As discussed in the previous section, the project data scientist developed the OR utilization tool to roll up in granularity of the caseload. The analysis performed for this section compared/contrasted a representative surgeon from each department (highest level of granularity the tool provided). Again, in the future, it would be helpful to run this analysis on many surgeons from many departments at CHCO to get a fuller picture of trending.

This analysis used the same data from Surgeon A but combined all weekdays (i.e., from Figure 14). The data scientist repeated this analysis for two other surgeons selected from separate departments (Surgeon B from Department II and Surgeon C from Department III). Figure 15 shows the resulting time series comparison.

As expected, the combined graph from Surgeon A contained 141 procedure dates (surgical blocks) which is what the team saw when they summed all weekdays from the previous comparison (i.e., 46 Mondays, 22 Tuesdays, 57 Wednesdays, 5 Thursdays, and 11 Fridays = 141 total procedure dates). The other two surgeons contained the following date counts: Surgeon B totaled 65 procedures in the dataset, and Surgeon C totaled 213 procedure dates. Even with a single representative subject from a department, the analysis showed noticeable variation.

Figure 15

Comparison of Representative Surgeons from Different Departments



The representatives from Departments II and III had much different scheduling rates. Department II had a very gradual ascent beginning 180 days prior to the scheduled procedure date, while department III was much more turbulent from a planning perspective. Department III had a mean % utilization of zero as close as about 75 days prior to the schedule date. This scheduling behavior contrasted with the elbow shape of the time series from the representative in

Department I, which appeared to be a hybrid of Departments II and III. The first half had a gradual caseload buildup until about 50 days prior to the schedule date, when it suddenly accelerated at a similar rate to Department III.

Another stark contrast between the three departments was the variance in percent utilization denoted by the confidence banding surrounding the darker mean trendline. Department I had the least variance throughout its entirety. While its trendline ascended more gradually, Department II had more significant variance as it got nearer to its schedule date. Department III's trendline showed an even more extreme variance as it approached its schedule date.

Chapter 5 – Discussion/Conclusion

Keeping in mind that this project used limited data to build the surgical utilization tool, the results still offered interesting, observable trends. Looking closer at the different time series comparisons, we see the results showed differences at nearly all comparison levels.

Interpretation of Results at Highest Granularity

Figure 11 showed Surgeon A's temporal case buildup for Mondays in OR1111. This time series displayed some interesting behaviors. First, the overall shape resembled an elbow. This formation suggested that scheduling was slower until about 50 days prior to the procedure date. At this point, the scheduling took a more dramatic upturn which lasted until a couple of days prior to the procedure date. Then the percent utilization took a slight downturn. Dexter et al. explained that this dip resulted from last-minute cancelations (Dexter, 2012). A second characteristic that Figure 11 showed, was the difference in variance (denoted as confidence banding around the darker mean line) of percent utilization throughout the temporal buildup. Note how it tapered at both ends of the trendline in this figure. In other words, each procedure date incorporated in this time series had fewer add-ons/cancelations further and closer from/to the procedure date. One could interpret this characteristic as evidence of having a relatively controlled schedule. Control would indicate a fairly high concentration of elective procedures for this surgeon, day, OR combination.

Interpretation of Results at Specified Weekdays (Medial Granularity)

For this same surgeon/weekday combination (Surgeon A, Mondays), the analysis rolled up all ORs throughout the dataset into one time series (Figure 13). Aggregating all the ORs on this day had several effects on the graph in Figure 11. Firstly, the shape remained nearly identical, perhaps the exception being the darker mean trendline on the day before surgery, terminating closer to the target of 80%. This behavior made sense since more procedure dates were combined. It also meant Surgeon A was doing a fantastic job hitting their % utilization target for Mondays (recall they want to leave about 20% unblocked to accommodate emergent procedures). A second noted difference was that the variance lost its tapering just before the procedure date. This characteristic could indicate a higher likelihood of more cancelations/add-ons in that time.

Moving to Figure 14, which compared/contrasted all of Surgeon A's weekdays, the results presented a more dynamic picture which corroborated the team’s pre-project hypothesis about weekday trend variation. Note how alike Mondays and Wednesdays were. Note also how alike Thursdays and Fridays were. This left Tuesdays, which appeared almost like a hybrid shape between the two sets. Like Monday, Wednesday had a gradual buildup with a sharp uptick around 20% utilization within 50 days of its procedure dates. Thursdays and Fridays were in stark contrast remaining at nearly zero percent caseload until it got within 25 days of the procedure date. Tuesdays displayed the same % utilization but took its uptick at 50 days prior, same as Mondays and Wednesdays. This graph behavior suggested that Surgeon A preferred to schedule electives on Mondays and Wednesdays, reserving first Tuesdays, then Fridays and Thursdays for overflow cases. The lower final percent utilization totals at Thursday's and Friday's time series terminus, coupled with the more dynamic confidence banding, reflected this assumption.

Interpretation of Results Between Different Departments (Lowest Granularity)

Figure 15 painted a whole other unique set of comparisons. Recall that the analysis generated each of these graphs from separate representative surgeons within three different specialty departments. Further investigations are needed to aggregate more robust representation within these departments. This collection of graphs began by showing what the time series looked like if we combined all the weekdays in Figure 14 (upper left graph). It still resembled Mondays and Wednesdays very closely. This panel brings to mind yet another reason to provide roll-up and drill-down functionality in the utilization tool. If the developer created the tool to only work at the lowest granularity, the user would have again been missing a vital part of the picture. As it happened, the tool *was* able to drill down and see that Fridays for Surgeon A were much different from the aggregate's overall shape.

The next time series graphed Surgeon B from Department II. This panel presented much differently than anything seen so far. A few characteristics immediately stood out. Firstly, the temporal caseload was built much more gradually than any other time series seen. This characteristic suggested that the department serviced more elective cases than the other two. To see this, examine each graph’s 20% utilization on the y-axis. For the other two departments, the time series reached this point at fewer than 50 days prior to the procedure date; for Department II, this was about double (100 days prior). The second noticeable difference was that the variance was more significant throughout Department II than I (observed as wider confidence banding). Though, similar to Department I, there was a slight taper a few days before the procedure. Thirdly, Department II had an uptick about a week prior to the procedure date but ended just below the target utilization of 80%. Perhaps there tend to be some last-minute add-ons for this department that Periop management could explore to help increase optimization.

The last time series (bottom) was the most dynamic. This department's % utilization stayed at zero until just 75 days back. Then the schedule was quickly built to arrive at a mean percent utilization of 90 just before the procedure date. Additionally, the variance with this department was more significant than anything seen up to this point. That trait suggested that this department serviced less predictable surgeries because the caseload was nearly zero until 75 days before the schedule date. The other two were already at about 20 and 35% by this day back.

Limitations of this Capstone

Due to the capstone's hard deadline (course completion), this investigation could not adjust its timeline to incorporate unforeseen complications so endemic to typical business projects. Because of this, the team adjusted its original goals when collaborations between the capstone and EMR teams uncovered an unknown query to extract data from the EMR in aggregate. This new dataset may allow a more sustainable solution to validate the scheduling software's predictive model, which was an initial goal of this project.

A second limitation of this investigation was the abbreviated dataset used to build the utilization tool. Again, the team only iterated over data extraction several times because of the semester-dictated timeline. If the project had a longer runway, the team would extract larger datasets to get as representative a sample size as the hospital's hardware could handle.

Thirdly, the utilization tool should be validated before CHCO makes any business decisions based on the software's results. Without validation, there is no way of confirming the accuracy of the predictive models it generated.

Conclusion

This project has shown that developing a surgical utilization tool can be beneficial in explaining OR time usage at several levels – owner, weekday, and OR. The project uncovered several actionable trends using only a fraction of the data available at CHCO. Understanding OR usage at a granular level will help CHCO maximize revenue, leading to increased overall quality of treatment.

Next Steps

The logical next steps are to validate the predictability of the utilization tool and measure performance. Once that is completed, a trial phase would be beneficial to observe the tool's performance in real-world situations. After that, a comparison against the scheduling software's predictive tool would be helpful. Perhaps these two models could be used in tandem to help improve every possible aspect of OR utilization.

References

American College of Surgeons. (2020). COVID-19: Executive Orders by State on Dental, Medical, and Surgical Procedures. https://www.facs.org/covid-19/archives/legislative-regulatory/executive-orders

Babu, M. A., Dalenberg, A. K., Goodsell, G., Holloway, A. B., Belau, M. M., & Link, M. J. (2019). Greening the Operating Room: Results of a Scalable Initiative to Reduce Waste and Recover Supply Costs. Neurosurgery, 85(3), 432–437. https://doi.org/10.1093/neuros/nyy275

Bai, G., & Zare, H. (2020). Hospital Cost Structure and the Implications on Cost Management During COVID-19. Journal of general internal medicine, 35(9), 2807–2809. https://doi.org/10.1007/s11606-020-05996-8

Bentley, T., & Ortner, N. (2020). 2020 U.S. Organ and Tissue Transplants: Cost Estimates, Discussion, and Emerging Issues. Milliman Research Report. Milliman Inc. https://www.milliman.com/-/media/milliman/pdfs/articles/2020-us-organ-tissue-transplants.ashx

Burrill, S., Kane, A., & Thomas, S. (2017). Deloitte 2017 Survey of US Health System CEOs: Moving forward in an uncertain environment. Chapter 3: Margins. Deloitte Center for Health Solutions. https://www2.deloitte.com/us/en/pages/life-sciences-and-health-care/articles/health-system-ceos.html

Centers for Disease Control and Prevention (CDC). (2018). CDC Public Health Professionals Gateway. United States Centers for Disease Control and Prevention. https://www.cdc.gov/phlp/publications/topic/hipaa.html

Childers, C.P., & Maggard-Gibbons, M. (2018). Understanding Costs of Care in the Operating Room. JAMA Surg. 2018;153(4):e176233. doi:10.1001/jamasurg.2017.6233

Children's Hospital Colorado. (2022). Department of Pediatric Surgery. https://www.childrenscolorado.org/doctors-and-departments/departments/pediatric-surgery/

Children's Hospital Colorado. (n.d.). Our History. Retrieved March 1, 2022, from https://www.childrenscolorado.org/about/history/

Clemen, R. (1989). Combining forecasts: A review and annotated bibliography. International Journal of Forecasting. Volume 5, Issue 4, p. 559-583. ISSN 0169-2070. https://doi.org/10.1016/0169-2070(89)90012-5

Crespin, D., Ruder, T., Mulcahy, A., & Mehrotra, A. (2022). Variation in Estimated Surgical Procedure Times Across Patient Characteristics and Surgeon Specialty. JAMA Surg. Published online March 02, 2022. https://jamanetwork.com/journals/jamasurgery/article-abstract/2789489?resultClick=1

Dexter, F., Shi, P., & Epstein, R. H. (2012). Descriptive study of case scheduling and cancellations within 1 week of the day of surgery. Anesthesia and analgesia, 115(5), 1188–1195. https://doi.org/10.1213/ANE.0b013e31826a5f9e

Gillespie, B. M., Chaboyer, W., & Fairweather, N. (2012). Factors that influence the expected length of operation: results of a prospective study. BMJ quality & safety, 21(1), 3–12. https://doi.org/10.1136/bmjqs-2011-000169

Giridharadas, M. (2022). The future of capacity management lies in predictive analytics, digitization. Physicians Practice. https://www.physicianspractice.com/view/the-future-of-capacity-management-lies-in-predictive-analytics-digitization

Hegji, Charles, (2007) "A brief look at hospital profits by outpatient services offered." Economics Bulletin, Vol. 9, No. 12 pp. 1-10. http://accessecon.com/pubs/EB/2007/Volume9/EB-07I10006A.pdf

IBM Corporation. (2022). Electronic Health Records with Epic and IBM FlashSystem 9500 Blueprint Version 2 Release 4. http://www.redbooks.ibm.com/redpapers/pdfs/redp5539.pdf

Kayis, E., Wang, H., Patel, M., Gonzalez, T., Jain, S., Ramamurthi, R. J., Santos, C., Singhal, S., Suermondt, J., & Sylvester, K. (2012). Improving prediction of surgery duration using operational and temporal factors. AMIA ... Annual Symposium proceedings. AMIA Symposium, 2012, 456–462. PMCID: PMC3540440

Macario, A., (2010). What does one minute of operating room time cost? Journal of Clinical Anesthesia. 2010;22, 233-236. doi:10.1016/j.jclinane.2010.02.003

Malhotra, L., Pontarelli, E. M., Grinberg, G. G., Isaacs, R. S., Morris, J. P., & Yenumula, P. R. (2022). Cost analysis of laparoscopic appendectomy in a large integrated healthcare system. Surgical endoscopy, 36(1), 800–807. https://doi.org/10.1007/s00464-020-08266-0

Matityaho, S. (2019). Why Collecting Information in the OR Is Vital. Para. 3,4. Chief Healthcare Executive. https://www.chiefhealthcareexecutive.com/view/collecting-information-operating-room-data-vital

Mazzaferro, et al. The Financial Impact of Covid-19 on a Surgical Department: the Effects of Surgical Shutdowns and the Impact on a Health System. *Scientific Forum Presentation*. American College of Surgeons Clinical Congress 2021. https://www.facs.org/media/press-releases/2021/financial-losses-102321

Merritt Hawkins. (2019). 2019 Physician Inpatient/Outpatient Revenue Survey. Merritt Hawkins an AMN Healthcare Company. https://www.merritthawkins.com/uploadedFiles/MerrittHawkins_RevenueSurvey_2019.pdf

MiraMed. (2019). The List is in: Hospitals' Most Profitable Specialties. MiraMed Global Services. https://www.miramedgs.com/ealerts/869-the-list-is-in-hospitals-most-profitable-specialties.html

Moazzez, A., & De Virgilio, C. (2016). Role of Surgical Services in Profitability of Hospitals in California: An Analysis of office of Statewide Health Planning and Development Annual Financial Data. The American Surgeon, 82(10), 894–897. https://doi.org/10.1177/000313481608201007

Muñoz, E., Muñoz, W., & Wise, L. (2010). National and surgical health care expenditures, 2005-2025. Ann Surg. 2010 Feb;251(2):195-200. doi: 10.1097/SLA.0b013e3181cbcc9a. PMID: 20054269.

El-Sayed, N., Sun, Z., Sun, K., & Mayerhofer, R. "OLTP In Real Life: A Large-scale Study of Database Behavior in Modern Online Retail," 2021 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2021, pp. 1-8, doi:10.1109/MASCOTS53633.2021.9614295.

National Cancer Institute. (n.d.). NCI Dictionaries. United States National Cancer Institute. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/perioperative

Olmsted, M., Powell, R., Murphy, J., Bell, D., Stanley, M., Torcasso Sanchez, R., & Allen, R. (2021). U.S. News & World Report's Best Children's Hospitals 2021-2022. https://health.usnews.com/media/best-hospitals/BCH_Methodology_2021-22.pdf

Park, K. W., & Dickerson, C. (2009). Can efficient supply management in the operating room save millions?. Current opinion in anaesthesiology, 22(2), 242–248. https://doi.org/10.1097/ACO.0b013e32832798ef

Patel, S., Lindenberg, M., Rovers, M., van Harten, W., Ruers, T., Poot, L., Retel, V., & Grutters, J. (2020). et al. Understanding the Costs of Surgery: A Bottom-Up Cost Analysis of Both a Hybrid Operating Room and Conventional Operating Room. International Journal of Health Policy and Management. doi:10.34172/IJHPM.2020.119

Payerchin, R. (2022). Pandemic trends could hurt hospital revenues. Medical Economics. https://www.medicaleconomics.com/view/pandemic-trends-could-hurt-hospital-revenues

Peters, J., & Young, D. (2017). Field Report: Pull Right Levers to Boost Operating Room Value. American Association for Physician Leadership.

https://www.physicianleaders.org/news/field-report-pull-the-right-levers-to-boost-value-of-the-operating-room

Polis, J. (Colorado, 2020). Exec. Order D 2020 045, Colorado Disaster Emergency Act, C.R.S. § 24-33.5-701 et seq. https://www.colorado.gov/governor/sites/default/files/inline-files/D%202020%20045%20Elective%20Surgeries_1.pdf

Rust, R., Moorman, C., & Dickson, P. (2002). Getting Return on Quality: Revenue Expansion, Cost Reduction, or Both. Journal of Marketing, Vol. 66, No. 4 (Oct., 2002), pp. 7-24. http://www.jstor.org/stable/3203355?origin=JSTOR-pdf

Stey, A.M., Brook, R.H., Needleman, J., Hall, B.L., Zingmond, D.S., Lawson, E.H., & Ko, C.Y. (2015). Hospital costs by cost center of inpatient hospitalization for medicare patients undergoing major abdominal surgery. J Am Coll Surg. 2015 Feb;220(2):207-17.e11. doi: 10.1016/j.jamcollsurg.2014.10.021. Epub 2014 Nov 8. PMID: 25529900.

Surgical Directions. (2021). How Hospitals Can Increase OR Profitability. https://www.surgicaldirections.com/wp-content/uploads/2021/07/11538_SD_IncreaseProfitability_v3.pdf

The Regents of the University of Colorado. (2022). University of Colorado Anschutz Medical Campus. https://medschool.cuanschutz.edu/surgery/education/general-surgery-residency/teaching-hospitals

Tiwari, V., Furman, W. R., & Sandberg, W. S. (2014). Predicting case volume from the accumulating elective operating room schedule facilitates staffing improvements. Anesthesiology, 121(1), 171–183. https://doi.org/10.1097/ALN.0000000000000287

United States Department of Health & Human Services (HHS). (2003). Summary of the HIPAA Privacy Rule. Office for Civil Rights Privacy Brief. https://www.hhs.gov/sites/default/files/privacysummary.pdf

Appendix A: Utilization Tool Redacted Source Code

This project was completed using Python v3.8.3 code in a Jupyter Notebook. The data is confidential. The code (minus the data sources) is available in a Github repository: https://github.com/philcallahan/capstone

- **OR Utilization Tool:** orutil_tool_redacted.ipynb
- **Function File:** orutil_redacted.py