A Monte Carlo Analysis of Staffing at a Mathematics Tutoring Center

By

BETHANY GORALCZYK

July 2013

A Thesis Submitted
in Partial Fulfillment of
the Requirements for Degree of

MASTER OF SCIENCE

The Graduate Mathematics Program
Curriculum Content Option
Department of Mathematics and Statistics
Texas A&M University-Corpus Christi

APPROVED: ___________________________ Date:__________

Dr. Joe Champion, Chair

Dr. Sarah Ives, Member

Dr. George Tintera, Member

Dr. George Tintera, Chair
Department of Mathematics & Statistics

Dr. Frank Pezold, Dean
College of Science & Engineering

Style: APA, 5th Edition
Abstract

Mathematics tutoring centers are academic support services established by most universities and community colleges to assist students enrolled in undergraduate and remedial mathematics courses (Perin, 2004). Typically offering non-credit, free-of-charge drop-in tutoring to students with limited funding and mostly part-time staff, these centers may face challenges of scheduling tutors weeks in advance with incomplete information about tutors' adherence to the schedule, students' arrival times, duration of tutoring visits, the timing and quantity of questions, or individual response times between tutors and students. This study applies a simulation-based analysis of a mathematics tutoring center by analyzing existing and potential new staffing strategies under realistic operating assumptions and parameter constraints. The Monte Carlo analysis applied weekly simulations of a multi-server tutoring queue, updated every minute of operations, using data on student demand and existing tutor scheduling at a mid-sized community college mathematics tutoring center in the South. Results include evaluation of sensitivity and stability of model predictions, probabilistic analysis of tutoring quality measures, and recommendations for modified staffing to improve student outcomes.
# Table of Contents

Abstract .......................................................................................................................... ii

Introduction .................................................................................................................. 1

Literature Review .......................................................................................................... 5

Methodology .................................................................................................................. 13

Results ............................................................................................................................ 28

Discussion ...................................................................................................................... 39

Summary ......................................................................................................................... 44

Conclusion ...................................................................................................................... 45

References ....................................................................................................................... 48

Appendix A: Source Code for Execution of Simulation Algorithm in R ................. 51

Appendix B: Source Code for Implementation of Simulation Algorithm in R...... 56
Introduction

Public universities and two-year colleges are often affected by a combination of large student enrollment in general education mathematics courses, increasing needs of students for mathematics remediation (Attewal, Lavin, Domina, & Levey, 2006), and changing economic and political conditions which have led to reduced state funding for higher education (Katsinas, D'Amico, & Friedel, 2011). These conditions make it important to maintain efficient and effective programs for supporting mathematics students, and free drop-in mathematics tutoring is usually one of the academic support services offered at universities and community-colleges. However, there has been limited research on mathematics tutoring centers (Perin, 2004), and a review of the literature found no studies on the problem of staffing a mathematics tutoring center to serve student and institutional needs. The author addressed this problem by developing a theoretical mathematical model of tutor-student interaction, implemented a simulation-based analysis of the model, and compared simulation results to data from a single mathematics tutoring center at a mid-sized community college in South Texas.

Mathematics courses at the research site account for 15% of the annual course registrations among over 12,000 enrolled students (Keas, Andrus, Haas, & Pallemoni, 2011). Many of these students use a mathematics learning center for help in courses ranging from beginning arithmetic and algebra to upper-level undergraduate mathematics courses. The mathematics learning center is a centralized two-room space seating up to 40 students who typically arrive
unannounced to work on homework, study for exams, or ask questions about class material. When students have a question, they raise their hand and are assisted by one of the (on average) seven mathematics tutors circulating the room. Support is provided by the next available tutor, who may work one-on-one or with a small group of students for a variable amount of time before moving to another student with a question. The staffing, hiring, training, and individual tutoring sessions are facilitated and coordinated by a small number of supervisors, one of which is the researcher.

The researcher is a former high school mathematics teacher who has been working at the mathematics tutoring center full-time as a supervisor for one year. Her main job duties consist of being available to tutor all levels of mathematics students throughout the week and managing the weekly schedule. As one of four current supervisors, each with at least a bachelor’s degree in mathematics, the researcher is particularly concerned with organizing a weekly schedule that is both cost efficient and leads to a high likelihood of successfully serving students throughout the scheduling period.

Existing methods to organize the weekly schedule start by charting tutors’ personal availability for the upcoming week, followed by informal consideration of computerized records of recent student attendance during operating hours, including peak days of the week, and general understanding of variation in demand for tutoring assistance throughout the semester (e.g., high demand prior to final exams). Schedulers often add some additional tutors to a few operating
hours in order to plan for tutor training activities during times where there is a surplus of tutors.

The tutoring center is open Monday through Saturday for a total of 58 hours per week (Monday – Thursday, 8 a.m. – 8 p.m.; Friday 8 a.m. – 1 p.m., Saturday 10 a.m. – 3 p.m.). Students may come at their own convenience, and no formal appointments are scheduled. The students sign-in on a central terminal using a unique student ID, sit down at any available space and begin working or studying independently. Tutors stand by; ready to assist if a student raises their hand. Generally, any tutor can help the student depending on the type of problem. Some students tend to ask questions seeking quick responses, while others ask questions requiring more extended responses. The relative numbers of tutors and students can greatly impact whether or not a student will receive help in a timely manner.

Despite the variability in demand and student-tutor interactions, a successful tutoring center must consistently provide enough available tutors to meet the needs of students while limiting the financial costs of being overstaffed. In particular, any potential benefits to students due to overstaffing may lead to harm later in the year due to a limited budget. At times when student demand for assistance is low, especially when these times can be anticipated in advance, the tutor-supervisors have the opportunity to train the tutoring staff by clarifying mathematical ideas or discussing techniques for effectively supporting students.

The purpose of this study was to use a mathematical model to optimize staffing at postsecondary mathematics tutoring centers while considering
educational outcomes. The research questions focus on four aspects of a
mathematics tutoring center's operations:

1. Given existing staffing strategies and variation in student demand,
   what is the expected number of students who leave the mathematics
   tutoring center due to excessive wait times or limited seating capacity?

2. Given existing staffing strategies and variation in student demand,
   during which days and times is the mathematics tutoring center likely
to be over- or understaffed?

3. Given existing staffing strategies and variation in student demand,
   what tutor to student ratio is most likely to minimize wait time and tutor
downtime?

4. Under realistic potential changes in student demand, what staffing
   strategies are most likely to minimize wait time and tutor downtime? To
   what extent do these outcomes warrant changes to budget, operating
   hours, and/or seating capacity?
Literature Review

More students fail or withdraw from college mathematics courses than any other subject (Halcrow & liams, 2011). Though there has been little research on the particular effects of attending mathematics learning centers on performance in postsecondary mathematics, a meta-analysis of over 60 studies on the effects of instruction which emphasizes worked examples suggests that students who engage regularly with mathematics tutors are likely to experience positive performance gains (Hattie, 2009). Informal analyses at the research site have suggested that students who spend extended time in the mathematics tutoring center tend to have higher than average course grades. In addition, a qualitative case study on Learning Assistance Centers in fifteen community colleges reported positive outcomes of participation, including improved retention and increased overall GPA (Perin, 2004). Outside of post-secondary education, Mathnasium is one of many mathematics tutoring companies offering paid tutoring for students from kindergarten to high school. Operating more than 300 centers in the U.S., their commissioned pre-test /post-test evaluation studies suggest a moderate to high increase in student performance on standards-based tests (Staggs, 2013; Watson, 2011).

Dvorack's (2004) research has identified four key components to managing a Learning Assistance Center (LAC) in a college or university. The political frame focuses on the importance of gaining support and funding for such programs from administrators. The symbolic frame emphasizes investment in getting students and tutors more connected to the mission of the LAC. The
structural frame of the tutoring center deals with effective organization, staffing, training of tutors, and evaluations, and is the component on which most LAC directors spend the majority of their time. The fourth component, or the human resource frame, centers on motivating staff to do their best by, for example, encouraging the strengths of individual tutors in serving a diverse student population. Tutor training also has a significant impact on students' academic success. That is, “Having a sound structure, knowing how theory informs practice, and applying different management approaches can help in the complex task of managing a tutoring program. The responsibilities of organizing, staffing, training, and evaluating tutors can be daunting, but also very rewarding.” (Dvorack, p 49).

Client-based Staffing

Bechet (2002) conceptualizes the notion of strategic staffing, also known as workforce planning, as the effective combination of staffing strategy and staffing planning. A staffing strategy is a long-term, directional plan that describes what an organization is going to do over the course of its planning to ensure that its supply of staff matches its demand for services provided by the staff. The other side of the coin is staffing plans, which are processes the organization will implement in the short term to address immediate staffing gaps and surpluses.

When operational objectives for an organization are tied to staffing decisions, a combination of analytical techniques and data-based strategies can be used to address the objectives while accounting for specific constraints and goals of the organization. For example, a linear programming procedure was
used to reduce staffing costs of nursing personnel by 16% while simultaneously improving conditions for nursing staff to be optimally productive by assigning accountability and by improving employee and patient satisfaction (Matthews, 2005). Much like mathematics tutoring centers, health care staffing can be made difficult by increasing demand for services and administrative pressure to deliver effective one-on-one services in highly variable circumstances within a fixed staffing budget. The strategy of balancing staffing ratios, for example, was found to be a useful tool in workforce planning for nine allied health professions (Cartmill, Comans, & Clark 2012). Zhu, You, and Zheng (2012) found that increased nursing staff per patient had statistically significant positive effects on patient outcomes, and inadequate nurse staffing was linked to negative patient outcomes, which can in turn reflect negatively on the overall clinic.

**Queues & Staffing at Tutoring Centers**

There is often a strong negative correlation between wait time and a customer’s evaluation of service quality (Houston, Bettencourt, & Wenger, 1998). This can have negative impacts on both the company and the consumer. For example, we may hypothesize students who experience extended time periods waiting for help at a tutoring center may not come back in the future, and may possibly tell other students that there was no available help. Returning to the analogous operation of nurse staffing, De Vericourt and Jennings (2011) use a queue perspective to model a nurse staffing situation. They modeled the workload experienced by $s$ nurses with $n$ homogenous patients using the $M/M/s/n$ framework from Queue Theory where the first two elements of the
notation represent the arrival process and service time distribution respectively. $M$ refers to a Poisson or random process. Each patient arrives at random times of the day and alternates between two states, needing assistance and not needing assistance, or "needy" and "stable". This is similar to the mathematics tutoring center scenario where students may alternate between working independently and needing assistance. Service is delivered in a first come, first serve manner, and the perceived (and actual) quality of services is negatively related to wait time. The researcher’s goal was to minimize the probability of excessive delay in which a needy patient is waiting for services. They found their staffing method significantly differed from typical staffing strategies, including legislation that mandates fixed nurse-to-patient staffing ratios.

Queue Theory is also used in analyzing staffing levels at call centers (Green, Kolesar, & Whitt, 2007). For example, adapted stationary queuing models for nonstationary environments in which time dependent variables are considered can be used to analyze staffing strategies. As a point of reference, Liu, Yunan, Whitt, and Ward (2012) used contextual factors which resemble conditions in a typical mathematics tutoring center to develop a systematic algorithmic procedure to stabilize customer abandonment in many-server queues with time-varying arrivals. They compared their formula-based estimates to outcomes of simulations and found the algorithmic procedures to be effective (Liu et al.).
Monte Carlo Methods & Statistical Modeling

Developing a practical solution to a mathematical modeling problem is sometimes obtained through computer simulations, especially when dependencies in the model are too complex to be well understood using equations or formulas (Rubinstein, 2008). *Stochastic simulations*, also called Monte Carlo simulations, contain some random variables as opposed to *deterministic simulations* where all of the variables are predetermined. In even very basic queue-based models, Monte Carlo simulation may be used by selecting randomized numbers of people entering the queue and allowing for randomized service times. Then, simulating the queue through a fixed length of time can reveal which variables have the greatest impact on the modeled system and which have the least.

Another reason for using Monte Carlo techniques is the potential benefits of running simulations before implementing changes in real life operations. It might cost a tutoring center undue financial hardship or waste productive employee time if seemingly promising changes lead to little benefits, and simulations may be able to suggest both the likelihood and potential magnitude of changed outcomes due to implementing operational changes. It is important to note these queuing models are considered dynamic because the number of people in the queue may depend on the time of day and the day of the week. For instance, in the mathematics tutoring center, more students may visit before lunch and then again after 5:00 when people get off work. Upcoming tests may
also contribute to higher numbers of students. This leads to the importance of accounting for both uncertainty and variability among the factors in the model.

**Guiding Principles for Planning Monte Carlo Analysis**

The U.S. Environmental Protection Agency [EPA] (1997) has established eight guiding principles for a Monte Carlo analysis that consider the uncertainty and variability in many modeling situations that depend on random events.

1. There should be time and resources to complete a complex analysis of the situation.

2. The potential benefits of the analysis should warrant the level of required effort.

3. The analyst(s) should have the skills and experience needed to perform the evaluation.

4. The analysts should describe how quantification of uncertainty and variability will improve the evaluation of outcomes.

5. The analysis should identify major sources of variability and uncertainty, and should describe how variability and uncertainty will be kept separate in the analysis.

6. There should be evidence that a quantitative estimate of uncertainty will improve decision-making, even after accounting for variability.

7. Weaknesses and strengths of the methods should be evaluated.

8. The project should include a plan for communicating the variability and uncertainty analysis to the public and decision makers.
Guiding Principles for Conducting Monte Carlo Analysis

In addition to the important considerations of planning Monte Carlo analysis, the EPA (1997) suggests the following sequential procedure for conducting analysis of risk-based contexts (in the proposed context, "risk" refers to the possibility of students leaving the tutoring center due to staffing conditions):

1. **select input data and distributions** – use sensitivity analysis or numerical experiments to identify model structures, assess significant pathways and parameters, and use observational or surrogate data to inform choices.

2. **evaluate variability and uncertainty** – improve accountability and transparency by separating variability and uncertainty, apply methodological differences as appropriate, investigate stability of descriptive measures of distributions, and identify all sources of uncertainty (quantitatively or qualitatively).

3. **communicate results of analyses** – describe the model and associated equations (including limitations), describe input distributions, provide detailed information and graphs for output distributions, discuss dependencies and correlations, and present point estimates of outcomes.

   Since the mathematics tutoring center does not make appointments, the number of students that come for tutoring varies greatly from hour to hour and day to day. A tracking software system is used to keep track of how many students sign-in and sign-out. There may be some uncertainty in the data due to
students forgetting to sign-in. The actual number of students could be slightly different from what the data states. The available number of tutors can also vary due to illness or no shows. Intuition says that these uncertainties will not greatly affect the overall outcome of a general staffing strategy. As the model is developed, weaknesses and strengths will be addressed.
Methodology

The study addressed the research questions by implementing a Monte Carlo procedure which included the development of a theoretical mathematical model for week-long mathematics tutoring schedules, simulation of educational outcomes, and cycles of analysis in which model parameters were adjusted based on data from the research site. The analysis used the following procedure:

1. Use a review of the literature to identify a justified mathematical model for tutoring queues in the context of variable student demand.
2. Use existing tutoring data to estimate distributions of model factors and parameters under current staffing strategies at the research site.
3. Implement 5 to 10 runs of a simulation program to estimate tutoring outcomes under model assumptions and estimated parameters.
4. Compare simulation results to existing tutoring data. Modify the simulation algorithm, modeling assumptions, and/or input parameters.
5. Repeat Steps 3 & 4 until simulation results provide adequate fit to the distribution of outcomes in the existing data set.
6. Simulate a sufficient number of runs of the modified algorithm to account for variability in model factors while ensuring stability of predicted outcomes.
7. Analyze simulation-based estimates of model outcomes using guidelines identified in the literature review.
8. Develop alternate staffing strategy for the research site to improve simulated outcomes.
9. Repeat Steps 6 & 7 for the new staffing strategy until staffing
  recommendations optimize predicted measures of tutoring quality.

10. Repeated Steps 2-9 to develop and analyze staffing strategies under
  plausible changes in student demand or tutoring center resources.

Due to the complex, stochastic nature of the tutoring context and limited prior
research on the related modeling problem, preliminary numerical experiments
were conducted to identify important input variables and parameters. Following
literature-based guidelines (EPA, 1997), the Monte Carlo processes were
simplified by using the results of the preliminary sensitivity analysis by “fixing”
some parameters instead of using probability distributions. Sampling strategies
were used to develop distributions for input data, and the data were analyzed for
possible sources of uncertainty. For instance, the tracking software used at the
tutoring center automatically assigns one hour as a student’s time spent if the
student forgets to sign-out.

*Initial Modeling Assumptions*

As an initial step in the modeling procedure, and as part of the
researcher’s coursework in mathematical modeling, the researcher collaborated
with classmates and the research advisor to develop a working initial simulation
program that was based on an initial diagram that showed the minute by minute
student and tutor process (Figure 1).
This design helped guide the following fixed and variable input factors and assumptions about tutoring operations. Initial assumptions originally included:

a) The tutoring center selects a staffing schedule one week in advance.

b) Funding constraints limit the total number of tutor service hours to a fixed weekly number.

c) The number of tutors can randomly differ from the schedule (supervisors meetings and other duties, illness).

d) Interarrival time of students varies randomly, but depends on both time of the day and the day of the week.

---

*Figure 1. Student and tutor actions during tutoring (updated each minute).*
e) The tutoring center has a fixed capacity for the number of students at any given time.

f) Students have variable neediness levels (frequency with which they ask questions of tutors).

g) Tutors are homogenous in their response time to questions, serve students in a first come, first serve (FCFS) manner, and knowledgeable enough to respond to any question.

h) Question response times are variable and independently distributed.

i) Students will leave the tutoring center if the current or total wait time exceeds personal thresholds.

j) The tutoring center can be overstaffed if a fixed number of tutors remain in standby beyond a fixed threshold.

k) The tutoring center is functioning well when the center is rarely overstaffed, students typically receive assistance on a high percentage of their questions, and question wait times are low, and few students leave the center due to wait times or room capacity.

Model Specification

Since the goal of this model was to identify student- and question-level outcomes in a tutoring center over a one week period, the model used arriving students as the unit of analysis by identifying each arriving student with time-dependent vectors (updated once per minute) to track the student's experience in the tutoring center. The student vector included arguments taken to be parameter-based independent random variables, together with some "dummy
variables” to track temporary quantities, such as the number of available tutors in
the room and the amount of time the student has been waiting for help on the
current question.

The simulator program comprises two $R$ script files. The primary script for
obtaining results uses single commands to initiate multi-week simulations by
specifying values for parameters and input variables and file path information to
store simulated outcomes (see Appendix A). The second file is a customized
algorithm with functions and loops to simulate the minute by minute process of
the tutoring queues (Appendix B). Documentation of the final simulator notation
are given in Table 1.

Table 1.
Description of Input Factors and Outcome Measures in the Model Simulator.

<table>
<thead>
<tr>
<th>Factor(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutor_supply</td>
<td>Table of positive whole numbers giving the average number of tutors scheduled during each half hour of each day of the week.</td>
</tr>
<tr>
<td>maxsubtract/maxadd</td>
<td>Integer constants serving as boundaries for uniformly random deviation in tutor_supply values.</td>
</tr>
<tr>
<td>student_demand</td>
<td>Table of positive numbers giving mean numbers of students arriving each hour of each day of the week (arrival times are uniformly distributed within the hour)</td>
</tr>
<tr>
<td>student_variation</td>
<td>Standard deviation of random normal variation in student arrivals per hour (means given by student_demand).</td>
</tr>
<tr>
<td>n.weeks.min, n.weeks.max, n.weeks.stable</td>
<td>The number of runs will be no less than the specified minimum, and may be less than the maximum if predicted mean wait times are within a 99% confidence for a specified number of weeks.</td>
</tr>
<tr>
<td>stayave</td>
<td>Mean duration (in minutes) that arriving students will...</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>plan to stay in the learning center.</td>
<td></td>
</tr>
<tr>
<td>staystd</td>
<td>Standard deviation of randomly varying durations of students’ planned stays in the learning center.</td>
</tr>
<tr>
<td>neediness</td>
<td>Random student-level estimate of need for tutoring help (continuous uniform distribution between 0=&quot;rarely ask for help, mostly short-duration questions&quot; and 1=&quot;frequently ask for help, mostly long-duration questions&quot;)</td>
</tr>
<tr>
<td>qtimes</td>
<td>Three number array of constant response times required for &quot;short&quot;, &quot;medium&quot;, and &quot;long&quot; questions, respectively. Actual response times are distributed among the qtimes with uniformly random weighted probabilities.</td>
</tr>
<tr>
<td>intermax</td>
<td>Maximum mean interarrival times between questions (normally distributed for each student with mean linearly dependent on intermax and neediness, and standard deviation equal to one-sixth of mean interarrival time)</td>
</tr>
<tr>
<td>maxwaitnow</td>
<td>maximum time a student will wait to receive help on question before leaving the tutoring center unsatisfied</td>
</tr>
<tr>
<td>scapacity</td>
<td>Maximum number of students who may be present in the room during any minute of tutoring</td>
</tr>
<tr>
<td>maxwaitperstay</td>
<td>Constant percentage of planned stay time in which a student will cumulatively wait to receive help on questions before leaving the tutoring center unsatisfied</td>
</tr>
<tr>
<td>extratutormin</td>
<td>minimum number of available tutors needed in order to identify time for training, (also depends on trainmin)</td>
</tr>
<tr>
<td>trainmin</td>
<td>minimum consecutive number of minutes needed in order to identify a time period as ideal for training available tutors</td>
</tr>
</tbody>
</table>

For each week of a simulation run, the scripts store a detailed student-by-student summary of tutoring outcomes. These values were used to debug program errors, train input parameters. Some of the output values were temporary "dummy" parameters, but others provide qualitative evidence of
tutoring outcomes while suggesting whether the simulator was working properly.

See Table 2 for a summary of these output quantities.

Table 2.

Description of Estimated Quantities in Simulation Output.

<table>
<thead>
<tr>
<th>Output Quantities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArriveDay, Arrive Hour, ArriveTime</td>
<td>The minute in which each student arrived at the room for a drop-in tutoring visit</td>
</tr>
<tr>
<td>PlanStay</td>
<td>Planned duration (in minutes) of stay for each student</td>
</tr>
<tr>
<td>TimeinRoom</td>
<td>The total number of minutes the student spent in the room.</td>
</tr>
<tr>
<td>neediness</td>
<td>Assigned neediness level for each student</td>
</tr>
<tr>
<td>P_Easy, P_Med, P_Hard</td>
<td>Probability weights for short, medium, and questions for each student</td>
</tr>
<tr>
<td>QuesNext</td>
<td>Number of minutes until each student's next question, based on neediness</td>
</tr>
<tr>
<td>QuesLength</td>
<td>Duration of the student's current (if being helped) or next question, as appropriate</td>
</tr>
<tr>
<td>N_Ans</td>
<td>Number of students’ questions that were completely answered by a tutor</td>
</tr>
<tr>
<td>HelpTime</td>
<td>The total number of minutes the student has been helped by tutors during their stay</td>
</tr>
<tr>
<td>WaitTime</td>
<td>Cumulative number of minutes for which the student had waited for help during the student's stay</td>
</tr>
<tr>
<td>TrainTime</td>
<td>Cumulative minutes of identified training time during each student's time in the room</td>
</tr>
<tr>
<td>Leave</td>
<td>Reason the student left the room (e.g., &quot;as planned&quot;, &quot;due to question wait time&quot;, &quot;due to total wait time&quot;, &quot;due to room being filled at capacity&quot;)</td>
</tr>
</tbody>
</table>
### Output Quantities

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ArriveDay, Arrive Hour, ArriveTime</strong></td>
</tr>
<tr>
<td>The minute in which each student arrived at the room for a drop-in tutoring visit</td>
</tr>
<tr>
<td><strong>StudentsPerTutor</strong></td>
</tr>
<tr>
<td>Time-weighted average ratio of students to tutors during the student's time in the room.</td>
</tr>
</tbody>
</table>

In the case of multi-week runs of the simulator, the algorithm also produces a log of measures of tutoring quality during each simulated week. Table 3 summarizes these point estimates.

Table 3.

Description of Composite Measures of Tutoring Quality per Week of Simulation.

<table>
<thead>
<tr>
<th>Composite Measures</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StudDemand</strong></td>
<td>Counts of student visits to the center for each day of each week</td>
</tr>
<tr>
<td><strong>QuesAnswered</strong></td>
<td>Number of students' questions answered during their stays (mean, median, standard deviation, minimum, maximum)</td>
</tr>
<tr>
<td><strong>HelpTime</strong></td>
<td>Cumulative minutes spent with tutors being helped during students' stays (mean, median, standard deviation, minimum, maximum)</td>
</tr>
<tr>
<td><strong>WaitTime</strong></td>
<td>Cumulative minutes spent waiting for help from tutors during students' stays (mean, median, standard deviation, minimum, maximum)</td>
</tr>
<tr>
<td><strong>StudentsPerTutor</strong></td>
<td>Composite time-weighted ratio of students per tutor during students' stays (mean, median, standard deviation, minimum, maximum)</td>
</tr>
<tr>
<td><strong>TrainTime</strong></td>
<td>Cumulative minutes identified for tutor training during students' stays (mean, median, standard deviation, minimum, maximum)</td>
</tr>
<tr>
<td><strong>PercentLeaveRoom</strong></td>
<td>Frequency table of reasons for students' leaving the room (as planned, due to wait time, or capacity)</td>
</tr>
</tbody>
</table>
Training and Development of the Simulator Algorithm

Applying the EPA (1997) guiding principles, the simulation program was systematically trained and debugged by conducting numerical experiments to achieve reasonable sensitivity to changes in model parameters, appropriate distributions of stochastic processes, and informative summaries of predicted outcomes. The initial data for training was Fall 2012 summaries of student demand taken from the research site’s AccuSQL system, which is a widely-used commercial management software for academic centers such as the research site. The software provides a user-friendly interface for accessing accurate records of tutoring attendance and scheduling by generating charts and graphs to analyze traffic. A typical tutor supply was estimated by taking the average number of tutors scheduled for each half hour increment of a week for the entire semester.
Table 4.

Average Number of Scheduled Tutors at the Research Site, Fall 2012.

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-8:29</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>8:30-8:59</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>9:00-9:29</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>9:30-9:59</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>10-10:29</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>10:30-10:59</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>11-11:29</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>11:30-11:59</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>12-12:29</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>12:30-12:59</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>1-1:29</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30-1:59</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2-2:29</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2:30-2:59</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3-3:29</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30-3:59</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-4:29</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:30-4:59</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-5:29</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:30-5:59</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-6:29</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:30-6:59</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-7:29</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30-7:59</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5.

Average Hourly Number of Student Arrivals at the Research Site, Fall 2012.

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-8:59</td>
<td>9</td>
<td>15</td>
<td>11</td>
<td>13</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>9-9:59</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>12</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10-10:59</td>
<td>9</td>
<td>11</td>
<td>9</td>
<td>11</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>11-11:59</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>12-12:59</td>
<td>11</td>
<td>12</td>
<td>10</td>
<td>11</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>1-1:59</td>
<td>8</td>
<td>11</td>
<td>8</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2-2:59</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>11</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3-3:59</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-4:59</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-5:59</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-6:59</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-7:59</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The researcher went through a number of testing and revision cycles with the researcher's advisor in developing the model simulator. Some of the major changes in the revision process are listed below to document the process and to familiarize the reader with the model training process:

Problems with Output

- **ArrivalHour** initially always appeared as 11. The algorithm was fixed to loop throughout a 12 hour day.
- A frequency analysis of the number of student visits each day showed the number of visits did not follow a random distribution. The algorithm was changed to ensure random variation in student visits per day.
Validation of Tutoring Processes

- The researcher fixed all parameters in the simulation except the capacity of the room, which was decreased to check for changes in students’ reasons for leaving the room.
- The researcher increased students’ personal threshold of wait times to see if fewer students would leave due to wait time.
- The researcher improved the handling of interarrival times of students’ questions by changing the manner in which a student had a question. Initially, a student arrived with a predetermined number of questions that would be addressed consecutively. This caused a backup in the queue where a tutor would be with that particular student until all questions were answered. In the updated code, the intermax variable was added so that the next question arrived based on a neediness level. When students arrive, they may alternate between needing assistance and working independently without tying up a tutor for long periods of time. Students’ leaving due to wait time was decreased to 4-16% from 50%.
- The researcher ran simulations with excessively abundant tutor supply to see if the percent of students leaving due to wait time would decrease. Predictions were validated by checking that predicted periods of overstaffing increased while the understaffed times decreased.
  - During this process, the following input parameters were added so the simulation process could be easily adjusted and verified:
    - `tutor_supply.df`, `student_demand.df`,
    - `student_variation.df`, `nweeks.max`, `qtimes`,

```
maxsubtract, maxadd, scapacity, stayave, staystd,  
intermax, maxwaitnow, extratutormin, trainmin,  
maxwaitperstay.

- The researcher worked with her advisor to add input parameters to  
ensure the simulator would repeat for enough weeks to indicate  
stabilization of the tutoring processes while avoiding uncontrolled run  
times: nweeks.max, nweeks.min, nweeks.stable

The estimated number of students that leave due to wait time is highly  
sensitive to the number of tutors available and the student demand. For instance,  
by arbitrarily choosing 10 tutors to always be scheduled, the average number of  
students that leave due to wait time dropped to less than 1% from 8%. The  
outcomes are much less sensitive to changes in question response times,  
personal thresholds of wait times, and interarrival time between students’  
questions. The simulator outcomes are least sensitive to the number of weeks  
simulated. For example, throughout the simulation trials in this study, the percent  
of students leaving due to wait time in the first run (week) is typically within 1-2%  
of the average for that measure across multiple weeks of simulation. This is likely  
due to the fact that one weekly simulation actually involves simulation of tutoring  
processes over thousands of individual minutes. Overall, the simulator  
parameters were tuned so that outcomes were plausible and representative of  
the experience of the researcher in the math tutoring center.
Justification of Methods

This research represents a significant component of a master’s degree in mathematics because of the application of rigorous mathematical modeling methods and the potential benefits the methodology and results will provide to the mathematics education community, including the community college serving as the research site and the many other institutions which operate similar support services. Although optimal staffing strategies and uses have been widely researched, the methods of analysis have not been directly applied to mathematics tutoring centers. In addition, these types of services at community colleges and universities are regularly exposed to risks of reduced funding due to budget cuts from administrators, states, and the federal government. The analysis of optimal staffing strategies can also be used for grant writing or reports to administration to support claims regarding the efficiency of tutoring programs and needs for increased resources.

Students will also reap benefits from this study. The overall goal was for the researcher to use mathematics education to maintain an efficient program that continues to reach out to more students. The more students that receive quality tutoring service, the more likely they are to tell classmates, who in turn bring more students into the center. Even though this particular tutoring center is located in a community college, the results can be used in any tutoring program, whether it is in secondary schools or universities. Tutoring programs could possibly alleviate the pressures on teachers who have too many students to give the much needed individual attention. Private tutoring services could also benefit
from similar staffing strategies because improved cost effectiveness is likely to enhance profits.

**Research Timeline**

The proposed research plan followed the procedure outlined in the methodology section. Emphasis was on developing the simulation algorithm, addressing model fit, optimizing tutoring outcomes, and developing recommendations for staffing at the research site and similar tutoring facilities.

Table 6.

Timeline of Thesis Work.

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 8 – April 17, 2013</td>
<td>Work on proposal</td>
</tr>
<tr>
<td>April 18, 2013</td>
<td>Send proposal to committee</td>
</tr>
<tr>
<td>April 23, 2013</td>
<td>Apply for IRB at Research Site &amp; TAMUCC</td>
</tr>
<tr>
<td>May 7, 2013</td>
<td>Defend proposal</td>
</tr>
<tr>
<td>May 20, 2013</td>
<td>Begin analysis of initial model</td>
</tr>
<tr>
<td>May 27, 2013</td>
<td>Stabilize model, run initial simulations</td>
</tr>
<tr>
<td>June 3, 2013</td>
<td>Finish model simulations, begin analysis of results</td>
</tr>
<tr>
<td>June 17, 3013</td>
<td>Complete interpretation of results</td>
</tr>
<tr>
<td>July 10, 2013</td>
<td>Send final thesis draft to committee</td>
</tr>
<tr>
<td>July 23, 2013</td>
<td>Defend thesis</td>
</tr>
</tbody>
</table>
Results

The culminating results of the modeling procedures are presented in this chapter alongside the four research questions. Subsequent chapters discuss the findings in the context of potential implications for scheduling tutoring at a mathematics learning center, sources of variability and uncertainty, and possible extensions to the study methodology and results.

Question 1: Student Wait Times and Tutoring Capacity

Given existing staffing strategies and variation in student demand, what is the expected number of students who leave the mathematics tutoring center due to excessive wait times or limited seating capacity?

After the extensive training and simulation development, it was possible to answer the first research question using a single use of the `simulate.weeks()` command in the simulator program R script. Using the nomenclature given in the Methodology chapter, the command was the following:

```r
simulate.weeks(tutor_supply.df, student_demand.df, student_variation.df, nweeks.min=15, nweeks.stable=5, nweeks.max=40, qtimes=c(2,7,30), maxsubtract=2, maxadd=0, scapacity=35, stayave=90, staystd=30, intermax=20, maxwaitnow=10, extratutormin=2, trainmin=15, maxwaitperstay=.3)
```

In natural language, the simulation command translates to the following collection of modeling assumptions and estimates:
1. Base the estimates on a minimum of 15 weeks of simulations and maximum of 40 simulations, with at least 5 consecutive weeks of statistically stable average wait times.

2. Anywhere from 0 to 2 tutors may fail to show up in a scheduled tutoring time; no tutors will work unless scheduled.

3. The capacity of the center at any given minute is 35 students.

4. Students' planned length of stay is normally distributed with a mean of 90 minutes and standard deviation of 30 minutes.

5. All question response times are one of 2, 7, or 30 minutes in duration, with likelihoods of each question type depending on individual students' (uniformly random) neediness levels.

6. The maximum cumulative time a student will wait to get their question answered is 30% of their planned length of stay; the maximum individual question wait time is 10 minutes.

7. Students' next request for help on a question is normally distributed with mean between 0 and 20 minutes (depending on each student's neediness level). The standard deviations of these interarrival times is one-sixth of the individual means.

8. If two or more tutors are not helping students for 15 or more consecutive minutes, those minutes will be identified as potentially appropriate for tutor training.

Each week of the 19 weeks run by the simulation program used approximately 20 to 30 seconds of processor time on the researcher's computer. Summarized
in Table 7, the output of the simulation provides statistical estimates of the central tendency and spread of tutoring operations. While the results included several indicators of tutoring quality, the point estimates specifically described in the research question include approximately 12% of students left due to wait times and 0% of students left due to tutoring center capacity.

Table 7.

Results for a Typical Week at the Mathematics Tutoring Center.

<table>
<thead>
<tr>
<th>Composite Measures</th>
<th>Mean</th>
<th>SD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students Arriving on Monday</td>
<td>99.7</td>
<td>12.3</td>
<td>76.0</td>
<td>125.0</td>
</tr>
<tr>
<td>Students Arriving on Tuesday</td>
<td>113.2</td>
<td>12.8</td>
<td>90</td>
<td>142.0</td>
</tr>
<tr>
<td>Students Arriving on Wednesday</td>
<td>100.1</td>
<td>10.1</td>
<td>80</td>
<td>126.0</td>
</tr>
<tr>
<td>Students Arriving on Thursday</td>
<td>105.3</td>
<td>12.6</td>
<td>75</td>
<td>129.0</td>
</tr>
<tr>
<td>Students Arriving on Friday</td>
<td>35.2</td>
<td>6.9</td>
<td>23.0</td>
<td>50.0</td>
</tr>
<tr>
<td>Students Arriving on Saturday</td>
<td>23.8</td>
<td>4.2</td>
<td>16.0</td>
<td>31.0</td>
</tr>
<tr>
<td>Questions Answered (per Student)</td>
<td>2.7</td>
<td>2.7</td>
<td>0</td>
<td>16.1</td>
</tr>
<tr>
<td>Minutes Being Helped (per Student)</td>
<td>34.7</td>
<td>39.9</td>
<td>0.0</td>
<td>237.4</td>
</tr>
<tr>
<td>Minutes Spent Waiting (per Student)</td>
<td>12.0</td>
<td>8.4</td>
<td>0.0</td>
<td>52.4</td>
</tr>
<tr>
<td>Time Weighted Ratio of Students to Tutors (per Student)</td>
<td>6.3</td>
<td>13.0</td>
<td>0.0</td>
<td>144.7</td>
</tr>
<tr>
<td>Minutes Available for Training (per Student)</td>
<td>2.3</td>
<td>8.3</td>
<td>0.0</td>
<td>62.8</td>
</tr>
<tr>
<td>Students leaving as planned (%)</td>
<td>87.8</td>
<td>2.6</td>
<td>81.2</td>
<td>91.7</td>
</tr>
<tr>
<td>Students leaving due to individual question wait time (%)</td>
<td>10.5</td>
<td>2.5</td>
<td>6.4</td>
<td>17.3</td>
</tr>
<tr>
<td>Students leaving due to cumulative question wait time (%)</td>
<td>1.7</td>
<td>0.6</td>
<td>0.6</td>
<td>3.0</td>
</tr>
<tr>
<td>Students leaving due to room being filled to capacity (%)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Question 2: Periods of Understaffing and Overstaffing

Given existing staffing strategies and variation in student demand, during which days and times is the mathematics tutoring center likely to be over- or understaffed?

The same carefully calibrated simulation command used in the results for Question 1 allowed for estimates of times during which the tutoring center is likely to be under- or overstaffed. During a given run of the simulation, a 30-minute scheduling period was identified as "understaffed" if one or more students left due to question wait times or the room being filled to capacity (or both) during the scheduled period. Moreover, (applying the same threshold as used for identifying tutor training times) a scheduling period was identified as "overstaffed" if there were at least 2 tutors not helping any students for 15 or more consecutive minutes. Note that these operational definitions of understaffing and overstaffing are not mutually exclusive. For example, if a large number of students arrive during the second half of a 30-minute scheduling block, the period would be identified as understaffed and also potentially identified as overstaffed (depending on the wait times and planned stay lengths of the arriving students).

Tables 8 and 9 give the percentage of the 19 simulated weeks in which each scheduled tutoring period was identified as over- or under-staffed, respectively. Estimated 0 probability values are omitted to make the table easier to read.
Table 8.

Estimated Percentage Probabilities of Overstaffing during Scheduled Tutoring.

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-8:29</td>
<td>100</td>
<td>89</td>
<td>100</td>
<td>84</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>8:30-8:59</td>
<td>84</td>
<td>5</td>
<td>58</td>
<td>5</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>9-9:29</td>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>58</td>
</tr>
<tr>
<td>9:30-9:59</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-10:29</td>
<td>16</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>95</td>
</tr>
<tr>
<td>10:30-10:59</td>
<td>5</td>
<td>11</td>
<td></td>
<td>5</td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>11-11:29</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30-11:59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12-12:29</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:30-12:59</td>
<td>16</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-1:29</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30-1:59</td>
<td>11</td>
<td></td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-2:29</td>
<td>5</td>
<td>21</td>
<td>42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:30-2:59</td>
<td>11</td>
<td>21</td>
<td>68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-3:29</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30-3:59</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-4:29</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:30-4:59</td>
<td>11</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-5:29</td>
<td>5</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:30-5:59</td>
<td>11</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-6:29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:30-6:59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-7:29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30-7:59</td>
<td>5</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 9.

Estimated Percentage Probabilities of Understaffing during Scheduled Tutoring.

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-8:29</td>
<td>100</td>
<td>26</td>
<td>95</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:30-8:59</td>
<td>5</td>
<td>100</td>
<td>42</td>
<td>100</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>9-9:29</td>
<td>89</td>
<td>21</td>
<td>89</td>
<td>47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:30-9:59</td>
<td>63</td>
<td>11</td>
<td>95</td>
<td>42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-10:29</td>
<td>5</td>
<td>68</td>
<td>5</td>
<td>100</td>
<td>16</td>
<td>42</td>
</tr>
<tr>
<td>10:30-10:59</td>
<td>11</td>
<td>63</td>
<td>5</td>
<td>74</td>
<td>11</td>
<td>47</td>
</tr>
<tr>
<td>11-11:29</td>
<td>11</td>
<td>53</td>
<td>26</td>
<td>42</td>
<td>11</td>
<td>26</td>
</tr>
<tr>
<td>11:30-11:59</td>
<td>21</td>
<td>32</td>
<td>42</td>
<td>37</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>12-12:29</td>
<td>16</td>
<td>26</td>
<td>42</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:30-12:59</td>
<td>11</td>
<td>26</td>
<td>53</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-1:29</td>
<td>26</td>
<td>32</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30-1:59</td>
<td>16</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-2:29</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:30-2:59</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-3:29</td>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30-3:59</td>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-4:29</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:30-4:59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-5:29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:30-5:59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-6:29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:30-6:59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-7:29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30-7:59</td>
<td>5</td>
<td>21</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Interpretations of the relative likelihoods of under- and overstaffing in the simulations depend partly on subjective thresholds for acceptable risks of
understaffing due to variation in student arrivals, question frequency and duration, and students' planned stay times. However, the results do suggest the first and last scheduled periods are more likely to be overstaffed, especially the first scheduled period. Moreover, the tutoring center is more likely to be understaffed during Tuesday and Thursday mornings.

**Question 3: Optimized Student to Tutor Ratio**

*Given existing staffing strategies and variation in student demand, what tutor to student ratio is most likely to minimize wait time and tutor downtime?*

The research site was presumed to be operating "better" when, after fixing all parameters except the tutoring schedule, simulated student wait times and periods of understaffing and overstaffing could be reduced while students' help times increased. Toward this objective, the researcher used the results from Question 2 to adjust the tutoring schedule in the simulations by decreasing or increasing the number of tutors available during individual 30-minute tutoring periods. Then the simulation was run again, for six runs each time, and the results were again analyzed. This process was repeated about five times until the likelihood of being over- and understaffed was minimized and the number of students that left due to wait time was less than 5%, if possible. The final optimized student outcomes are presented in Table 10. The table includes an estimate of the optimal average student-to-tutor ratio of 5.9 to 1.
Table 10.

Student Outcomes with Optimal Staffing Results.

<table>
<thead>
<tr>
<th>Composite Measures</th>
<th>Mean</th>
<th>SD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questions Answered (per Student)</td>
<td>2.9</td>
<td>2.9</td>
<td>0.0</td>
<td>17.2</td>
</tr>
<tr>
<td>Minutes Being Helped (per Student)</td>
<td>37.2</td>
<td>41.4</td>
<td>0.0</td>
<td>245.8</td>
</tr>
<tr>
<td>Minutes Spent Waiting (per Student)</td>
<td>11.3</td>
<td>7.6</td>
<td>0.0</td>
<td>47.0</td>
</tr>
<tr>
<td>Time Weighted Ratio of Students to Tutors (per Student)</td>
<td>5.9</td>
<td>11.9</td>
<td>0.0</td>
<td>121.9</td>
</tr>
<tr>
<td>Minutes Available for Training (per Student)</td>
<td>2.5</td>
<td>8.3</td>
<td>0.0</td>
<td>56.4</td>
</tr>
<tr>
<td>Students leaving as planned (%)</td>
<td>94.3</td>
<td>2.3</td>
<td>89.7</td>
<td>98.3</td>
</tr>
<tr>
<td>Students leaving due to individual question wait time (%)</td>
<td>5.0</td>
<td>2.3</td>
<td>1.3</td>
<td>10.3</td>
</tr>
<tr>
<td>Students leaving due to cumulative question wait time (%)</td>
<td>0.6</td>
<td>0.5</td>
<td>0.0</td>
<td>1.9</td>
</tr>
<tr>
<td>Students leaving due to room being filled to capacity (%)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 11 lists the adjustments to the original tutor supply associated with the optimized student tutoring outcomes. The net change in scheduled tutoring hours (+21 hours) is less than a 1% increase from the original total scheduled hours.
Table 11.

Adjustments to Tutor Supply Associated with Optimized Tutoring Outcomes.

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-8:29</td>
<td>8:30-8:59</td>
<td>+2</td>
<td>+2</td>
<td>-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:29</td>
<td>9:30-9:59</td>
<td>+4</td>
<td>+4</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:29</td>
<td>10:30-10:59</td>
<td>+2</td>
<td>+3</td>
<td>+3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:29</td>
<td>11:30-11:59</td>
<td>+2</td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:29</td>
<td>12:30-12:59</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:29</td>
<td>1:30-1:59</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:29</td>
<td>2:30-2:59</td>
<td>-2</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Question 4: Scheduling Responses to Change in Student Demand*

*Under realistic potential changes in student demand, what staffing strategies are most likely to minimize wait time and tutor downtime? To what extent do these outcomes warrant changes to budget, operating hours, and/or seating capacity?*

To explore the sensitivity of tutoring outcomes to changes in student demand for tutoring, the original student demand was decreased by 25% and
increased up to three times the original student demand (all other parameters were fixed at the values listed in Question 1). Due to the extended run times for each simulation and high stability in predicted outcomes, a minimum of five simulations were initiated while requiring at least two stable runs. Table 12 provides a comparison of students' reasons for leaving under these simulations results (all distributions except the baseline estimates are averages of 6 runs).

Table 12.

Effects of Student Demand on Students' Reasons for Leaving Tutoring.

<table>
<thead>
<tr>
<th>Proportion of Original Student Demand</th>
<th>Left as Planned (%)</th>
<th>Left Due to Individual Question Wait Time (%)</th>
<th>Students who left due to cumulative question wait time (%)</th>
<th>Students who left due to room being filled to capacity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.75</td>
<td>97.5</td>
<td>1.8</td>
<td>0.7</td>
<td>0</td>
</tr>
<tr>
<td>1 (baseline)</td>
<td>87.8</td>
<td>10.5</td>
<td>1.7</td>
<td>0</td>
</tr>
<tr>
<td>1.5</td>
<td>62</td>
<td>35.6</td>
<td>2.2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>43.7</td>
<td>54.5</td>
<td>1.6</td>
<td>0</td>
</tr>
<tr>
<td>2.5</td>
<td>33</td>
<td>66.7</td>
<td>1.3</td>
<td>0.02</td>
</tr>
<tr>
<td>3</td>
<td>27.4</td>
<td>71.4</td>
<td>1.2</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 12 also illustrates a trend in the simulation results in which very few students leave due to room capacity. For example, when both the overall student demand and tutor supply were doubled, the predicted percentage of students leaving due to capacity was still less than 1%. Tripled student demand and tutor supply resulted in just 16% of students leaving the room due to
capacity. The research site is unlikely to experience tripled student demand for tutoring in the near future.
Discussion

Following the structure of the results chapter, the discussion of findings is organized by research question.

**Question 1: Student Wait Times and Tutoring Capacity**

*Given existing staffing strategies and variation in student demand, what is the expected number of students who leave the mathematics tutoring center due to excessive wait times or limited seating capacity?*

According to the simulation results, the percentage of students expected to leave due to wait time (combined wait time with total wait time percentage) in one week was approximately 12%. Since about 477 students arrived for tutoring per week, this means about 58 students would leave due to a lack of tutoring help. This estimate is reasonable in the context of the modeling assumptions, but also suggests room for improvement in a staffing strategy. Though there are no published thresholds for this measure of tutoring quality, the importance of "repeat business" and word-of-mouth for academic support services, combined with the researcher’s experience at the research site, suggest the percentage of students that leave due to wait time should probably be kept below 5%, or possibly even less.

The results also indicated that students rarely leave the tutoring center due to limited seating capacity. This is plausible because the initial estimates of student demand are based on logs of students logging into tracking software at the research site, and anyone who may have left the tutoring center due to perceiving a full room is likely to not have been counted in the actual data. Such
a student would probably leave the room before entering. In addition, experience at the research site suggests that some students, depending on their neediness level, may leave before the room reaches capacity, and many students who may be affected by overcrowding at the tutoring center would also be likely to be affected by extended wait times when they ask for help.

**Question 2: Periods of Understaffing and Overstaffing**

*Given existing staffing strategies and variation in student demand, during which days and times is the mathematics tutoring center likely to be over- or understaffed?*

At first glance, the overstaffing and understaffing results identified several time periods during which there was likely to be both understaffing and overstaffing. In particular, this tended to happen within 30 minutes of opening. This is partly an artifact of how the simulator determined overstaffing. If there was at least one instance of a 15 minute period of 2 or more available tutors, then the time period was identified as overstaffed. Due to the relatively low numbers of student arrivals and uniform spread of arrival times in the first half hour, there is a high likelihood of 2 tutors not being needed in the first 15 minutes. But once students began arriving, there was also a potential risk of understaffing, where students may have left due to wait time. In situations like these, the priority should likely be to err on the side of overstaffing, while noting the opportunity to conduct tutor training.

In interpreting other times identified with nonzero probabilities of overstaffing and/or understaffing, initial emphasis should be placed on higher
The research site was considerably overstaffed on Monday and Friday morning, from 8:00 – 9:30 a.m. This finding aligns with what might be predicted by looking at the tutoring schedules and student arrival logs: there is lower student demand for tutoring on Mondays and Fridays, whereas the tutor supply is more consistent throughout the week. Saturday was also identified as being overstaffed in the first hour, from 10 – 11 a.m., as well as in the last hour, from 2 – 3 p.m. In general, the simulations suggest there are not as many problems with overstaffing as there are with understaffing. With the exception of Monday, there was a high chance of being understaffed in the first half of the day. Friday and Saturday are less likely to be understaffed. The simulation findings suggest evenings are typically well staffed, with sporadic low percentages of over- or understaffing issues. In the researcher’s experience, these results are reasonable, except possibly for a few days just before exams which cause student demand to increase unexpectedly.

Question 3: Optimized Student to Tutor Ratio

Given existing staffing strategies and variation in student demand, what tutor to student ratio is most likely to minimize wait time and tutor downtime?

The optimized student to tutor ratio was found by increasing and decreasing the initial tutor supply counts during scheduled tutoring periods using a cyclical process of interpreting over- and understaffing results, analyzing the updated results, and making changes until the number of students leaving due to wait time was eventually very near to 5%. As noted in the discussion of Question...

likelyhoods. The research site was considerably overstaffed on Monday and... and student arrival logs: there is lower student demand for tutoring on Mondays and Fridays, whereas the tutor supply is more consistent throughout the week. Saturday was also identified as being overstaffed in the first hour, from 10 – 11 a.m., as well as in the last hour, from 2 – 3 p.m. In general, the simulations suggest there are not as many problems with overstaffing as there are with understaffing. With the exception of Monday, there was a high chance of being understaffed in the first half of the day. Friday and Saturday are less likely to be understaffed. The simulation findings suggest evenings are typically well staffed, with sporadic low percentages of over- or understaffing issues. In the researcher’s experience, these results are reasonable, except possibly for a few days just before exams which cause student demand to increase unexpectedly.

Question 3: Optimized Student to Tutor Ratio

Given existing staffing strategies and variation in student demand, what tutor to student ratio is most likely to minimize wait time and tutor downtime?

The optimized student to tutor ratio was found by increasing and decreasing the initial tutor supply counts during scheduled tutoring periods using a cyclical process of interpreting over- and understaffing results, analyzing the updated results, and making changes until the number of students leaving due to wait time was eventually very near to 5%. As noted in the discussion of Question...
2, the overstaffing concerns in the morning may continue to persist while the other areas of staffing concerns were further addressed.

At first, the tutor supply was increased by two tutors if the percentage of understaffing was higher than fifty percent and increased by one if the percentage was between 30% and 50%. The same plan was implemented for the overstaffed time periods, except a reduction in the tutor supply was implemented. After about five trials of making adjustments, the simulated optimal ratio was approximately six students per tutor, which is similar to ratio observed in the baseline data. This suggests that the average student to tutor ratio should not be the only factor determining whether or not tutor supply meets the needs of student demands for tutoring. The percentage of students that leave due to wait time (combined individual and cumulative) in the baseline simulations was approximately 12.2% while the adjusted tutor supply halved this risk to just 5.6%. Another component revealed in finding the best ratio was how much the tutor supply needed to be changed in order to optimize student outcomes. The net change in tutor supply was 21 additional hours, which is less than a 1% increase from the original tutor supply. This suggests a simulation-based methodology could be used to substantially improve student outcomes at the tutoring center without having much effect on funding needs for the services.

**Question 4: Scheduling Responses to Change in Student Demand**

*Under realistic potential changes in student demand, what staffing strategies are most likely to minimize wait time and tutor downtime? To what extent do these outcomes warrant changes to budget, operating hours, and/or seating capacity?*
Several different scenarios of student demand were compared to investigate the changes in student outcomes without changes in tutor supply. It was found that the student outcomes generally behaved as expected, with increased student demand positively correlated to student to tutor ratios and the percentages of students that left the room due to wait time. Interestingly, if all other parameters were held constant as student demand increased, the average student’s total wait time for tutoring plateaued around 18 minutes, and room capacity was rarely a concern. This suggests more tutors would eventually be needed so students will not leave due to wait time before capacity was reached.

In the test day scenario, reasonable changes to student demand and tutor supply were applied and results continued to show no signs of capacity issues. The method of matching the tutor supply to the student demand is also slightly flawed due to the fact that it is harder to predict the actual amount of increase in student demand unless further studies are done to compare the typical days to the testing days. At any rate, the attempt to match the tutor supply to student demand maintained acceptable percentages of students who left as planned until the student demand was tripled. Because of this fact, the immediate staffing strategies would not include requesting more space. Even though student demand does fluctuate from semester to semester, and it is plausible that more instructors may require their students to spend time in the tutoring center, it is not likely that the student demand would increase by a factor of 3 or more without additional resources and space.
Summary

Given the assumptions made in the modeling procedures, the mathematics tutoring center at the research site appeared to be operating slightly below optimized staffing strategies during the Fall 2012 semester. The model predicted approximately 88% of students intending to stay approximately 90 minutes at the tutoring center would leave the tutoring center satisfied, after experiencing approximately 35 minutes of tutoring help, 12 minutes of wait time, and an average of 2.7 questions answered by tutoring staff. In absolute numbers, this would mean about 50 students leaving due to question wait time each week. The simulations identified a few periods of time in the mornings where training could be implemented and the tutoring center was not reaching full capacity. With small changes to the original scheduling of tutors during operations, the optimal student to tutor ratio was found to be approximately 6 to 1 and the numbers of students leaving due to question times reduced by half. An exploratory analysis of changes to student demand suggested student demand and tutor supply could potentially double at the existing tutoring location under model assumptions, and scheduling could be optimized with only minor increases (approximately 1%) in expenditures for tutor supply during understaffed periods.
Conclusion

If the mission of the mathematics tutoring center is to assist students in their mathematics related courses, and if simulation results suggest there is room to reach more students with very minimally increased costs, then why not? This objective is directly tied to the concept of a long term staffing strategy (Bechet, 2008). By trying different scenarios of increased tutor and student demand with the current space provided, the analysis suggests the mathematics tutoring center would not need to expand its facility until the student demand nearly tripled. This would mean more than 1,400 student visits per week. If this were to happen, then the budget would be heavily impacted due to the need for more tutors and more space. For now, the research site may be better served by preparing for a doubling of student demand. Since there are four full time supervisors that take up most of the current expenditures, doubling the tutor supply with part time tutors would not have as much of an effect on the overall budget. The research site plans to advertise and increase the quality of tutors to get increased student demand all while addressing existing understaffing issues.

The immediate staffing plan at the research site is to input the previous weeks’ tutor supply and the corresponding student demand from the log-in software and to analyze simulation-based results. Tutors will be added to the time periods whenever the simulation results suggest there is a higher than 50% chance of being understaffed during the scheduled period, while more tutor training will be scheduled during the time periods where it is highly likely to be overstaffed. Potential training sessions could include a discussion of common
misconceptions on particular problems or strengthen tutor skills by reviewing advanced topics such as Calculus or probability and statistics. Upcoming exam days will also be noted on the schedule in an attempt to anticipate short-term increases in student demand and to supply more tutors as needed. In this way, the simulation methods in this thesis will be used as a data-based tool to assist the scheduling process.

**Extensions**

One way to make the results of the simulation more accurate would be to conduct observational studies regarding some of the input parameters. For example, an observational study could estimate the average frequency of questions per student and the average time tutors spend answering each question. Also, there is uncertainty in the baseline simulation data due to uncertainty in the validity of student arrivals and stay times at the tutoring center. This is due to the possibility of students, for whatever reason, not signing in or out of the tracking software at the research site. The system is also set up to automatically assign a one hour stay time to those who forget to sign out, which could bias estimates of stay times. Further analysis should be also conducted to see how much fluctuation in student demand can be identified during high and low demand times across weeks of the academic year, such as before exams and holidays. These results would help the researcher schedule ideal numbers of tutors during operating hours.

Further, a staffing schedule may be easier to access by using the simulation results to generate a suggested tutoring schedule based solely on
student demand and a desired student to tutor ratio. A researcher could use hourly student demand and estimates of tutoring process parameters to generate a baseline for the number of tutors needed per hour. For example, if the ratio is selected at 6:1 and student arrivals are expected at approximately 40 students in a given hour, the baseline number of tutors may be about 7. Then, the algorithm could simulate many weeks of tutoring outcomes, use estimates of understaffing and overstaffing to modify the tutoring schedule, simulate more weeks of tutoring, and so on, eventually producing a suggested tutoring schedule.

Another source of uncertainty in this study was the question of maximum capacity for students. The number of seats available for students may not accurately represent the effective capacity for students, who may choose to not go to the learning center if they perceive it to be full, regardless of tutoring outcomes. A study on perceived capacity of mathematics tutoring centers could improve the validity of the study methods and will help staff have a more informed answer when students or instructors ask about "good times" to go for tutoring.
References


Perin, D. (2004). Remediation beyond developmental education: The use of learning assistance centers to increase academic preparedness in


Appendix A: Source Code for Execution of Simulation Algorithm in R

# Author: Bethany Goralczyk
# Project: Monte Carlo Simulation of Staffing at a Mathematics Tutoring Center
# Created: 4/12/2013
# Updated: 7/11/2013
# Notes: * depends on helper functions from tutoring-scripts.R.
# Purpose of scripts: run the minute by minute tutoring process
# Purpose of FINAL simulator file: Run final results for each research question

# Preliminaries
# clear the working memory in R to avoid conflicts
rm(list = ls(all = TRUE))

# set the location of the working directory (depends on the computer, use / as the folder separator)
setwd("C:/Users/bgora_001/Desktop/SIMULATOR/training");

# Load required packages
require(reshape)
# Note: We need a special package ("reshape") to use the rbind.fill command in the monte-carlo simulation
require(reshape)

# Load custom tutoring scripts
# if you change the scripts, save the file, and then return to here and rerun this code
# ".../filename.R" means the file named filename in the folder "above" the working
directory (adjust as needed)
source("../tutoring-scripts.R")

# LOAD INPUTS (Fall 2012 Tutor Schedule and Student Demand)

# load hourly schedule for tutoring supply
tutor_supply.df<-as.data.frame(read.csv("schedules/tutor-schedule-F12.csv"));

# load hourly schedule for mean student demand
student_demand.df<-as.data.frame(read.csv("schedules/student-schedule-F12.csv"));

# load hourly schedule for stddev of student demand
student_variation.df<-as.data.frame(read.csv("schedules/student-schedule-stddev-F12.csv"));

# Double check the sampled data
tutor_supply.df
student_demand.df
student_variation.df

##### RESEARCH QUESTIONS #1,2

system.time(
  results.alternate<-simulate.weeks(tutor_supply.df, student_demand.df, student_variation.df, 
    nweeks.min=15, nweeks.stable=5, nweeks.max=40, folder="results/Research Questions 1 & 2",simname="sim_RQ_1&2_checked",
RESEARCH QUESTION #3: Adjust tutor supply by analyzing days and time where over or undertagged to student ratio in order to minimize wait time and tutor downtime.

```r
new_tutor_supply.df <- as.data.frame(read.csv("schedules/tutor-schedule-F12-FindRatio.csv"));
new_tutor_supply.df

system.time {
  results.minwaits <- simulate.weeks(new_tutor_supply.df, student_demand.df, student_variation.df,
                                     nweeks.min = 10, nweeks.stable = 5, nweeks.max = 40, folder = "results/Research Question 3", simname = "simFindRatio3",
                                     qtimes = c(2, 7, 30), maxsubtract = 2, maxadd = 0, scapacity = 35, stayave = 90,
                                     staystd = 30, intermax = 20,
                                     maxwaitnow = 10, extratutormin = 2, trainmin = 15, maxwaitperstay = .3 )
}

RESEARCH QUESTION #4: Realistic Potential Changes in Student Demand

# Scenario 1) Decrease/Increase in student demand overall (NO CHANGE in tutor supply)

tutor_supply.df
new_student_demand.df <- as.data.frame(read.csv("schedules/student-schedule-F12-.75.csv"));
new_student_demand.df <- as.data.frame(read.csv("schedules/student-schedule-F12-1.5.csv"));
new_student_demand.df <- as.data.frame(read.csv("schedules/student-schedule-F12-2.csv"));
new_student_demand.df <- as.data.frame(read.csv("schedules/student-schedule-F12-2.5.csv"));
new_student_demand.df <- as.data.frame(read.csv("schedules/student-schedule-F12-3.csv"));
new_student_demand.df

system.time (
    results.minwaits <- simulate.weeks(tutor_supply.df, new_student_demand.df, student_variation.df, nweeks.min=5, 
    nweeks.stable=2, nweeks.max=10, folder=\"results/IncreaseStudents-3\", simname=\"sim-Students3\", 
        qtimes=c(2,7,30), maxsubtract=2, maxadd=0, scapacity=35, stayave=90, 
        staystd=30, intermax=20, 
        maxwaitnow=10, extratutormin=2, trainmin=15, maxwaitperstay=.3 )
)

new_student_demand.df <- as.data.frame(read.csv("schedules/student-schedule-F12-1.5.csv"));
new_student_demand.df <- as.data.frame(read.csv("schedules/student-schedule-F12-2.csv"));

# Scenario 2) Increased Student Demand AND Tutor supply for Test Days
#  i) double students & tutors on Monday & Tuesday
#  ii) half students& tutors on Wednesday (after test)
#  iii) check results to see if optimized

new_tutor_supply.df <- as.data.frame(read.csv("schedules/tutor-schedule-F12-TestDay.csv"));
new_tutor_supply.df
new_student_demand.df <- as.data.frame(read.csv("schedules/student-schedule-F12-TestDay.csv"));
new_student_demand.df
system.time (
  results.minwaits<-simulate.weeks(new_tutor_supply.df, new_student_demand.df,
  student_variation.df, nweeks.min=5,
  nweeks.stable=2, nweeks.max=10, folder="results/TestDay", simname="sim2-TestDay",
  qtimes=c(2,7,30), maxsubtract=2, maxadd=0, scapacity=35, stayave=90,
  staystd=30, intermax=20,
  maxwaitnow=10, extratutormin=2, trainmin=15, maxwaitperstay=.3 )
)

# Results for Scenario 2 indicated no need for expanded capacity or extended hours

# Scenario 3) Double & Triple Student Demand and Tutor Supply to see when capacity becomes an issue

new_tutor_supply.df<-as.data.frame(read.csv("schedules/tutor-schedule-F12-3.csv"));
new_tutor_supply.df
new_student_demand.df<-as.data.frame(read.csv("schedules/student-schedule-F12-3.csv"));
new_student_demand.df

system.time (
  results.minwaits<-simulate.weeks(new_tutor_supply.df, new_student_demand.df,
  student_variation.df, nweeks.min=5,
  nweeks.stable=2, nweeks.max=10, folder="results/Extensions", simname="sim3_Students_Tutor",
  qtimes=c(2,7,30), maxsubtract=2, maxadd=0, scapacity=35, stayave=90,
  staystd=30, intermax=20,
  maxwaitnow=10, extratutormin=2, trainmin=15, maxwaitperstay=.3 )
)
Appendix B: Source Code for Implementation of Simulation Algorithm in R

# R SCRIPTS TO SUPPORT SOURCE MONTE CARLO SIMULATION OF a MATH LEARNING CENTER
# Author: Joe Champion
# Project: Monte Carlo Simulation of Staffing at a Mathematics Tutoring Center
# Created: 4/12/2013
# Updated: 7/11/2013
# Notes: * This file stores the "behind the scenes" functions and procedures for the simulation process. Actual simulations will be run in a separate file.
#
#******************************************************************************

#helper function to add random noise to input data
addnoise<-function(data, stdevdata=NULL, maxsubtract=NULL, maxadd=NULL, globalmin=NULL, globalmax=NULL, ...){
data.new<-data
if (all(is.null(stdevdata),is.null(maxsubtract),is.null(maxadd))){
  #do nothing
  return(data.new)
}
# resample data according to a table of standard deviations
if(!is.null(stdevdata)){
options(warn=-1)
sample.row<-function(row){
  xx<-as.numeric(data[row,])
  dev<-as.numeric(stdevdata[row,])
  sampled<-round(rnorm(xx, mean=xx, sd=dev))
}
# sample the data by applying the sample.row function to each row
data.new<-t(sapply(1:dim(data)[1],sample.row))
}
data.new[is.nan(data.new)] <- NA # clean-up any broken numbers

# jitter the data uniformly
if(!is.null(maxsubtract)||!is.null(maxadd)){
  jitter.row<-function(row){
    jittered<-round(as.numeric(data.new[row,])+runif(dim(data.new)[2],-
                maxsubtract,maxadd)) # subtract up to maxsubtract, add up to maxadd to this row
  }

  # jitter the data by applying the jitter.row function to each row
  data.new<-t(sapply(1:dim(data.new)[1],jitter.row))
}

# replace newly out-of-bounds data with globalmin and globalmax
if(!is.null(globalmin)){
  data.new[data.new<globalmin]<-globalmin
}
if(!is.null(globalmax)){
  data.new[data.new>globalmax]<-globalmax
}

data.new

# Sample the length of the next question(s) from student(s) given a distribution of question times with a vector of probabilities
qtime.next<-function(qtimes=c(1,3,5),peasy,pmed,phard){
  # this is a helper function
# given a set of question times and vectors of probabilities for a set of students, 
sample the next question time

```r
nextqtime <- function(row){
  if(any(is.na(c(peasy[row],pmed[row],phard[row])))){
    return(NA)
  }
  sample(qtimes, 1, prob=c(peasy[row],pmed[row],phard[row]))
}
sapply(1:length(peasy), nextqtime)
```

# example uses of qtime.next

```r
# qtime.next(c(1,3,5),x$P_Easy,x$P_Med,x$P_Hard)

# Sample the time until the next question from a given student
qarrive.next <- function(neediness, intermax = 41){
  # this is a helper function
  # given a neediness quantity between 0 and 1, return the time until the next question

  nextqarrival <- function(rate){
    max(round(rnorm(1, intermax - (intermax - 1)*rate, (intermax - (intermax - 1)*rate)/6)), 0)
  # ensure at least zero minutes until the next question
  }
  out <- sapply(neediness, nextqarrival)
  out[is.nan(out)] <- 60*24  # replace errors with a full day (essentially infinity for our purposes)
  out
}
```

# example uses of qarrive.next
# qarrive.next(0)
# sapply(1:10,function(x){qarrive.next(.2)})

# Generate a sample of student demand
initstudents.time<-function(n, tstart=0, tend=12, tdur=1, day="Monday", stayave=30, staystd=10, qtimes=c(1,3,5), intermax=41, ...){
  # n = the number of students who will arrive this time period
  studs<-data.frame()
  if(!is.na(n)& n>0){
    arrival <-round((tstart-floor(tstart))*60) + sample(1:round(tdur*60-1), n, replace=TRUE)  # assume students' arrival times to the room are random uniform during the time period
    stay <-round(rnorm(n,stayave,staystd))  # assume intended stay time is normally distributed
    stay[stay<1]<-1  # assume everyone intends to stay at least 1 minute
    overcap<- (60*floor(tstart)+arrival+stay > tend*60)
    stay[overcap]<-tend*60 - ( 60*floor(tstart) + arrival[overcap] )
    peasy <-sample(1:100,n, replace=TRUE)  # assume uniform random proportions of easy/med/hard
    pmed <-sample(1:100,n, replace=TRUE)
    phard <-sample(1:100,n, replace=TRUE)
    firstq <-qtime.next(qtimes,peasy,pmed,phard)
neediness <- runif(n, 0, 1)  # assume neediness of students is uniform
firstq.arrive <- qarrive.next(neediness, intermax = intermax)

studs <- rbind(studs, data.frame(
  ArriveDay = rep(day, n), ArriveHour = rep(floor(tstart), n),
  ArriveMin = arrival, PlanStay = stay, Neediness = neediness,
  P_Easy = peasy / (peasy + pmed + phard),
  P_Med = pmed / (peasy + pmed + phard),
  P_Hard = phard / (peasy + pmed + phard),
  QuesNextArrive = firstq.arrive,
  QuesLength = firstq, N_Ans = rep(0, n),
  HelpTime = rep(0, n), HelpNow = rep(0, n), WaitTime = rep(0, n),
  WaitNow = rep(0, n),
  Room = factor(rep("not yet in room", n), levels=c("denied entry", "not yet in room", "in the room", "left the room"), ordered=TRUE),
  TimeinRoom = rep(0, n), TrainNow = rep(0, n), TrainTime = rep(0, n),
  FreeTutors = rep(0, n),
  GlobalTrainNow = rep(0, n),
  Leave = factor(rep(NA, n)),
  StudentsPerTutor = rep(0, n) ))

# PlanStay = students' intended length of stay in minutes (the students will leave happy if they are well-served this long)
# N_Ans = how many of this student's questions have been answered?
# QuesLength = total mins required to answer the current question
# QuesNextArrive = mins until the next time this student will have a question
# WaitTime = total mins spent waiting
# HelpTime = current mins of being helped on a problem
# WaitNow = current mins spent waiting time
# Room = record the student's location in or out of the room
StudentsPerTutor = average number of students in the room divided by average number of tutors in the room during the student's stay

FreeTutors = a global count of tutors who are available to help but not helping anyone.

levels(studs$Leave)<-c("left as planned", "question wait time", "total wait time", "room at capacity")
studs<-studs[with(studs, order(ArriveHour,ArriveMin)), ] #sort studs by arrival time
return(studs)

#example use of the students.hour function
x<-initstudents.time(30,tstart=6.5, tend=7.5, tdur=0.5, stayave=90)
# summary(x$PlanStay)

# simulate one minute of tutoring
update.min<-function(df,t.hour, t.min, ntutors, scapacity=40,qtimes=c(1,3,5), maxwaitnow=10,
          maxwaitperstay=.75, intermax=41, trainmin=10, extratutormin=3, ...){
  #note: the input df is a dataframe in the form of the output from students.hour
  if(dim(df)[1]==0){ #nothing to update
    return (df)
  }

df<-df[order(df$ArriveHour,df$ArriveMin), ] #sort students by arrival time

#Try to let students into the room
rooms <- max(scapacity-sum(df$Room=="in the room", na.rm=TRUE),0) #there is space for at least 0 people
arriving <- (df$Room == "not yet in room") & (df$ArriveHour <= t.hour) & (df$ArriveMin <= t.min)

if (sum(arriving) <= roomspace) {
  df$Room[arriving] <- "in the room"  # let all these people into the room
}
else {  # arriving exceeds roomspace
  if (roomspace > 0) {
    # let some people into the room
    df$Room[arriving][1:roomspace] <- "in the room"
    df$Leave[arriving][-(1:roomspace)] <- "room at capacity"  # everyone else turned away due to capacity
    df$Room[arriving][-(1:roomspace)] <- "denied entry"
  }
  else {  # room is already filled to capacity
    df$Room[arriving] <- "denied entry"
    df$Leave[arriving] <- c("room at capacity")
  }
}

# Who's in the room:
isinroom <- df$Room == "in the room"

# update TimeinRoom for the students that are now in the room
df$TimeinRoom[isinroom] <- df$TimeinRoom[isinroom] + 1

# Select who will be helped by the tutors
# first sort students by waiting time
df <- df[order(-df$WaitNow),]

# Classify the student's interest in help
help.new <- isinroom & df$HelpNow == 0 & df$QuesNextArrive == 0
help.more <- df$HelpNow > 0 & (df$HelpNow < df$QuesLength) & df$QuesNextArrive == 0
help.nomore <- (df$HelpNow >= df$QuesLength) & df$QuesNextArrive == 0

# Restrict who can be helped according to the number of available tutors
busytutors <- sum(df$HelpNow > 0) - sum(help.nomore)
freetutors <- ntutors - busytutors

if (freetutors > 0 && sum(help.new) > 0) {
  # assign available tutors to those who need help
  if (freetutors <= sum(help.new)) {
    nhelp.new <- freetutors
  } else {
    nhelp.new <- -sum(help.new)
  }
  help.new[which(help.new)[(1:nhelp.new)]] <- FALSE
  busytutors <- busytutors + nhelp.new
  freetutors <- freetutors - nhelp.new
} else {
  help.new <- FALSE*help.new  # no avail tutors, so don't help anyone
}
#Classify those who won't be helped
waiting<-isinroom & !(help.new|help.more|help.nomore) & df$QuesNextArrive==0
workbyself<-isinroom & df$QuesNextArrive>0

#update TrainTime
if(freetutors >= extratutormin){
    df$TrainNow[isinroom]<-df$TrainNow[isinroom]+1  # if extratutormin or more tutors are free, we can do training during this 1 minute block
    df$GlobalTrainNow<-df$GlobalTrainNow+1
} else{
    df$TrainNow[isinroom]<-0
    df$GlobalTrainNow<-0
}

#Update FreeTutors (dummy variable used to identify overstaffing)
df$FreeTutors<-freetutors

#if there have been at least ten consecutive minutes of enough tutors to do training, we could plan for training
df$TrainTime[isinroom & df$TrainNow==trainmin]<-df$TrainTime[isinroom & df$TrainNow==trainmin] + trainmin
df$TrainTime[isinroom & df$TrainNow>trainmin]<-df$TrainTime[isinroom & df$TrainNow>trainmin] + 1

#Update N_Ans
df$N_Ans[help.nomore]<- df$N_Ans[help.nomore]+1
#Update HelpTime & HelpNow


df$HelpNow[help.nomore]<-0

#Update WaitNow & WaitTime

df$WaitNow[waiting]<-df$WaitNow[waiting]+1 #current problem

df$WaitTime[waiting]<-df$WaitTime[waiting]+1 #running total over all problems

df$WaitNow[help.new]<-0

df$HelpNow[waiting]<-0

df$QuesNextArrive[workbyself]<-df$QuesNextArrive[workbyself]-1

df$QuesNextArrive[df$QuesNextArrive<0]<-0

#Update Leave & Room based on Wait Time

df$Leave[df$WaitTime>maxwaitperstay*df$PlanStay]<"total wait time"

df$Leave[df$WaitNow>=maxwaitnow]<"question wait time"

df$Room[df$WaitTime > maxwaitperstay*df$PlanStay]< "left the room"

df$Room[df$WaitNow >= maxwaitnow]< "left the room"

#Update QuesLength & QuesNextArrive

#pick new question times for those who are done being helped

if(any(help.nomore)){

df$QuesLength[help.nomore]<-qtime.next(qtimes, df$P_Easy[help.nomore],

df$P_Med[help.nomore], df$P_Hard[help.nomore])

df$QuesNextArrive[help.nomore]<-qarrive.next(df[help.nomore,"Neediness"],intermax=intermax)
# Allow students to leave the room if they've stayed as long as they intended
```
df$Room[df$TimeinRoom >= df$PlanStay] <- "left the room"
df$Leave[df$TimeinRoom >= df$PlanStay] <- "left as planned"
```

# Update AveTutors
```
t.elapsed <- (t.hour - df$ArriveHour) * 60 + df$ArriveMin + t.min
avetutors <- (t.elapsed * df$StudentsPerTutor + n.tutors / sum(isinroom)) / (t.elapsed + 1)
df$StudentsPerTutor[isinroom] <- 1 / avetutors[isinroom]
```

return(df <- df[order(df$ArriveHour, df$ArriveMin), ]) # sort students by arrival time)

# example use of update.min
```
# x.1min <- update.min(x, t.hour=6.5, t.min=30, n.tutors=5, scapacity=10)
```

# simulate one week of tutoring
```
simulate.week <- function(sdemand, tsupply, timecolumn="ElapsedTime", maxwaitnow=15,
                           stayave=120, staystd=45, qtimes=c(2, 5, 30), scapacity=30,
                           maxwaitperstay=.3, intermax=30, trainmin=10, n.weeks=1,
                           simname="", folder="", extratutormin=3, ...){

    # assume student demand is hourly
    # assume tutoring schedule is not necessarily hourly

    session_starts <- tsupply[, , timecolumn] # beginnings of sessions (in elapsed hours)
    session_durations <- c(diff(session_starts), (sdemand[dim(sdemand)[1]], timecolumn) + 1 -
    session_starts[length(session_starts)]))
```
# create a table to track times of overstaffing
overstaffed<-tsupply
overstaffed[,names(overstaffed)!="ElapsedTime"]<-0

# simulate the week
df.week<-data.frame()
for ( day in names(sdemand)[names(sdemand)!=timecolumn] ){

    # initialize a dataframe for the simulation results
df.today<-data.frame()

    # day length
dayend<-max(sdemand[,timecolumn])+1 # assume sdemand times are hourly

    # simulate the day
    for ( i in 1:length(session_starts) ){

        ntuts<-tsupply[i,day]
        if(!is.na(ntuts)){

            # generate student demand for this session
            # student demand during the session is proportional to the hourly
            student demand
            hourdemand<-sdemand[sdemand[,timecolumn]==floor(session_starts[i]),day]
            nstuds<-round(session_durations[i]*hourdemand)

            df.new<-initstudents.time(nstuds,tstart=session_starts[i],
tend=dayend, tdur=session_durations[i],
            }
### Code Snippet

```r
# simulate tutoring for this session (minute by minute)

t.lag <- round(60 * (session_starts[i] - floor(session_starts[i])))
for (t.clock in 0:(round(session_durations[i] * 60) - 1)) {
  df.today <- update.min(df.today, t.hour = session_starts[i],
                         t.min = t.lag + t.clock,
                         ntutors = ntuts, maxwaitnow = maxwaitnow,
                         scapacity = scapacity, qtimes = qtimes,
                         maxwaitperstay = maxwaitperstay, intermax = intermax,
                         trainmin = trainmin, extratutormin = extratutormin)
  if (df.today$GlobalTrainNow[1] > trainmin &&
      df.today$FreeTutors >= extratutormin &&
      overstaffed[i, day] == 0) {
    overstaffed[i, day] <- 1
  }
}
}
```

```r
# save the overstaffing data to a spreadsheet
runnumber <- sprintf("%03d", n.weeks)
write.csv(overstaffed, file.path(folder, paste(simname, "," , 
                                   "-overstaffing-run-", runnumber, ".csv", sep="")), row.names=FALSE, append=FALSE)
```
df.week$Leave[is.na(df.week$Leave)] <- "left as planned"
print(summary(df.week$Leave))
df.week
}

# Example use of simulate.week
# system.time(test.fs <- simulate.week(student_demand.df, tutor_supply.df, scapacity=20))

# table(test.fs$ArriveDay, test.fs$Leave)

# example to save the simulated data in a spreadsheet
# write.csv(test.fs, "results/test-simulations-complete-6-2-2013.csv", row.names=FALSE)

# summarize results of one week of simulated tutoring
simweek.summary <- function(simweek, sdemand, tsupply, days=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"), timecol="ElapsedTime", ...){

  # summary number of tutors and students
  sdemand.days <- c(colSums(sdemand[days], na.rm=TRUE), Total=sum(sdemand[days], na.rm=TRUE))
  names(sdemand.days) <- paste("StudDemand", names(sdemand.days), sep="_")

  results <- data.frame(t(sdemand.days))

  results <- cbind(results,
                    QuesAnswered_Mean = mean(simweek$N_Ans),
                    QuesAnswered_Std = sd(simweek$N_Ans),
                    QuesAnswered_Min = min(simweek$N_Ans),
                    QuesAnswered_Max = max(simweek$N_Ans),...
HelpTime_Mean = mean(simweek$HelpTime),
HelpTime_Std = sd(simweek$HelpTime),
HelpTime_Min = min(simweek$HelpTime),
HelpTime_Max = max(simweek$HelpTime),

WaitTime_Mean = mean(simweek$WaitTime),
WaitTime_Std = sd(simweek$WaitTime),
WaitTime_Min = min(simweek$WaitTime),
WaitTime_Max = max(simweek$WaitTime),

StudentsPerTutor_Mean = mean(simweek$StudentsPerTutor),
StudentsPerTutor_Std = sd(simweek$StudentsPerTutor),
StudentsPerTutor_Min = min(simweek$StudentsPerTutor),
StudentsPerTutor_Max = max(simweek$StudentsPerTutor),

TrainTime_Mean = mean(simweek$TrainTime),
TrainTime_Std = sd(simweek$TrainTime),
TrainTime_Min = min(simweek$TrainTime),
TrainTime_Max = max(simweek$TrainTime)

table.temp <- round(100*table(simweek$Leave)/sum(table(simweek$Leave),na.rm=TRUE),1)
names(table.temp) <- paste("PercentLeaveRoom","_",names(table.temp),sep="\"")
table.temp <- as.data.frame.matrix(t(table.temp))

return( cbind(results,table.temp) )

Example Use of simweek.summary
#simweek.summary(test.fs,student_demand.df,tutor_supply.df)
# simulate multiple weeks of tutoring, saving spreadsheets of the results to a folder
simulate.weeks<-function(tsupply, sdemand, sdev, nweeks.stable=1, nweeks.min=1, nweeks.max=30, scapacity=40, addnoise=TRUE, simname="sim-X", folder="results", timecolumn="ElapsedTime", conf.level=.99, maxwaitnow=15, maxsubtract=3, maxadd=0, stayave=30, staystd=10, maxwaitperstay=.75, intermax=41, qtimes=c(1,3,5), trainmin=10, extratutormin=3, ...){

    # create a folder to store the simulation results
    dir.create(file.path(getwd(),folder), showWarnings = FALSE)

daynames<-names(tsupply)[names(sdemand)!=timecolumn] #assume same day names in tsupply, sdemand, sdev

    #helper function to test whether the simulation results have stabilized
    test.stable<-function(data.new, mean.old, n.runs=10, conf.level=.99,...){
        if(n.runs < nweeks.min){ #assume the data is stable if mean.old is based on few
            return(FALSE)
        }
        #apply a t-test to see if the new data is outside the earlier data
        if( t.test(data.new, mu=mean.old, conf.level=conf.level)$p.value < 1-conf.level ){#p-value is below the critical value, this data does not jive with prior means
            return(FALSE)
        }
        else{
            return(TRUE)
        }
    }

    #Helper function to determine if a week's simulation identified times of understaffing
staffing.week<-function(df.thisrun, title="U", ...)
{
  # Helper function to identify if there is any understaffing during a given day
  staffing.today<-function(df.thisrun, day, times, title="Understaffed", ...)
  {
    left.wait<-sapply(unique(times),
      function(thishalf)
    {
      ifelse(any(df.thisrun$Leave[times==thishalf &
        df.thisrun$ArriveDay==day] %in% c("question wait time", "total wait time")), 1, 0)
    }
    left.wait
  }
  halfhours<- floor(2* (df.thisrun$ArriveHour+df.thisrun$ArriveMin/60)/2
  
  df.out<-data.frame(cbind(sapply(levels(df.thisrun$ArriveDay),
    function(day){staffing.today(df.thisrun, day=day,times=halfhours, title=title) }  )))
  rownames(df.out)<-unique(halfhours)
  names(df.out)<-paste(names(df.out),title, sep="_")
  df.out
}

##### Run Simulations for each Week
results.summary<-data.frame()
current.stable<-0
n.weeks<-0
is.stable<-TRUE
while ((current.stable < nweeks.stable | n.weeks < nweeks.min )& n.weeks < nweeks.max)
{
  n.weeks<-n.weeks+1
  #### Initialize tutoring supply and student demand
### Jigger the Inputted Spreadsheets for Tutoring Supply & Student Demand

```r
if(addnoise){
    # randomly vary the tutoring supply
    # - subtract up to 3 tutors from all but the first column
    # - restrict to at least 1 tutor (at least one tutor will arrive during every scheduled time)
    tsupply.this[,daynames]<-
    addnoise(tsupply.this[,daynames],maxsubtract=maxsubtract,maxadd=maxadd,globalmin=1)

    # randomly sample student demand according to observed mean and stddev
    # - sample student demand from normal distributions using mean and stddev for each hour of each week
    # - restrict sampled values to between 0 and 30
    # - the -1 in the code is because we're ignoring the first column of the student demand
    sdemand.this[,daynames]<-
    addnoise(sdemand.this[,daynames],stdevdata=sdev[,daynames], globalmin=0,
             globalmax=scapacity)
}
```

```r
sim.thisweek<-simulate.week(sdemand, tsupply, 
              maxwaitnow=maxwaitnow, stayave=stayave, staystd=staystd, 
              qtimes=qtimes, scapacity=scapacity, maxwaitperstay=maxwaitperstay,
```
intermax=intermax, trainmin=trainmin,
extratutormin=extratutormin,
n.weeks=n.weeks, simname=simname, folder=folder)
runnumber<-sprintf("%03d", n.weeks)
write.csv(sim.thisweek, file.path(folder,paste(simname,"-run-",runnumber,".csv",sep="")), row.names=TRUE)

#Update understaffing
if(n.weeks==1){
  staffing.weeks<-staffing.week(sim.thisweek)
}
else {
  staffing.weeks<-staffing.weeks+staffing.week(sim.thisweek)
}

#combine a summary of this new simulation with previous summaries
results.summary<-rbind.fill(results.summary,
simweek.summary(sim.thisweek,sdemand.this,tsupply.this,days=daynames))
writeLines(strwrap(paste("Completed Run ", runnumber)))

is.stable<-test.stable(sim.thisweek$WaitTime,
mean(results.summary$WaitTime_Mean),n.runs=n.weeks, conf.level=conf.level)
current.stable<-ifelse(is.stable, current.stable+1, 0)

#Combine Overstaffing Counts to a single value
overstaffing.weeks<-list.files(folder, pattern = paste(simname,'overstaffing',sep="-")
overstaffing.combined<-as.data.frame(read.csv(file.path(folder,
overstaffing.weeks[1])),rownames=FALSE) #initialize
for (filename in overstaffing.weeks[-1] ){
    overstaffing.combined[-1]<-as.data.frame(read.csv(file.path(folder, filename)),rownames=FALSE)[-1]+overstaffing.combined[-1]
}

write.csv(overstaffing.combined, file.path(folder,paste(simname,"over-staffing.csv",sep="-")), row.names=FALSE)
write.csv(staffing.weeks, file.path(folder,paste(simname,"under-staffing.csv",sep="-")), row.names=FALSE)
write.csv(results.summary, file.path(folder,paste(simname,"summary-data.csv",sep="-")), row.names=TRUE)
writeLines("Successfully saved the summary results to spreadsheets.")
return(results.summary)

#Example Use of simulate.weeks
#test.mc<-simulate.weeks(tutor_supply.df, student_demand.df, student_variation.df, nweeks=2, folder="results/test", simname="sim-6-2-13")