2011-05-05

Development of a Decision Support Tool to Test Energy Management Alarming Thresholds

Aaron Tarjan
University of Miami, aaron@tarjan.net

UNIVERSITY OF MIAMI

DEVELOPMENT OF A DECISION SUPPORT TOOL TO TEST ENERGY MANAGEMENT ALARMING THRESHOLDS

By

Aaron V. Tarjan

A THESIS

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Master of Science

Coral Gables, Florida

May 2011

UNIVERSITY OF MIAMI

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Science

DEVELOPMENT OF A DECISION SUPPORT TOOL TO TEST
ENERGY MANAGEMENT ALARMING THRESHOLDS

Aaron V. Tarjan

Approved:

Shihab Asfour, Ph.D. Professor and Chairman of Industrial Engineering	Terri A. Scandura, Ph.D. Dean of the Graduate School
Murat Erkoc, Ph.D. Assistant Professor of Industrial Engineering	Arzu Onar-Thomas, Ph.D. Associate Member Department of Biostatistics St. Jude Children's Research Hospital

TARJAN, AARON V. (M.S., Industrial Engineering)

Development of a Decision Support Tool to Test Energy Management Alarming Thresholds (May 2011)

Abstract of a thesis at the University of Miami.

Thesis supervised by Professor Shihab Asfour, Ph.D.

No. of pages in text. (114)

A novel model was developed to test the use of short data sets for testing various alarm thresholds as part of an energy management program. Several years of 15-minute interval data were utilized from five buildings in Jacksonville, Florida. The model aggregated the data by day type and occupancy so that there were four period types used. For all of the buildings' meters, their daily usage by period type was tested against the threshold to determine if an alarm would be triggered, which would then be assigned a reward and cost based upon the type and duration of response. The risk management value was converted to dollars, in order to normalize the energy and time. It was determined that the 5-month short data set was the most appropriate choice for short data sets. In addition, it was concluded that the thresholds should be set between 0.8 and 1.0 standard deviation above the average of the short window. Several recommendations for further study are also enclosed.

Acknowledgement

I would like to thank my Committee – Dr. Shihab Asfour, Dr. Murat Erkoc and Dr. Arzu Onar-Thomas for assisting me in this endeavor. I'm especially grateful for the opportunity Dr. Asfour provided me by allowing to return to school and work in the Industrial Assessment Center. The training I received there was invaluable for my career. Also, the guidance from Dr. Onar-Thomas was extremely valuable as she was so able in dissecting my work and helping me patch the holes.

Along the way, a couple of professor friends provided some key guidance in helping me move the project along. Big thanks to Jan Schwartz and Sam Brody. Their simple tweaks came just at the right time. Some colleagues also proved crucial to the completion of the project. I want to thank Gary Reams, for allowing me to utilize all the data I needed. Cary Fukada proved invaluable as several lengthy discussions helped me to develop the "risk management" concept.

I can't thank my father, Dr. Peter P. Tarjan, enough for all of his help during the final stages of this project. His commitment to excellence has been one of life's greatest lessons.

Lastly, I want to thank Liz, my best friend, chief advisor, wife and love of my life. When I couldn't see clearly, her patience, tolerance, strength and determination kept refocusing me.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AHU	air handling unit
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
BAM	baseline adjustment multiplier
BR	billable rate
CV	constant volume
DAS	data acquisition system
ECM	energy conservation measure[1]
ESCO	energy service company[2]
FDD	fault detection and diagnostics
FEMP	Federal Energy Management Program
IPMVP	International Performance Measurement and Verification Protocol
LF	load factor
LFM	load factor multiplier
M&V	measurement and verification[3]
NG	natural gas
OAT	outside air temperature
PA	period average
PM	projected multiplier
PU	period usage
PV	projected value
RT	energy rate
RTU	Rooftop unit
SF	size factor
SOC	standard of occupancy and control
TM	time spent
TTM	twelve trailing months
US DOE	United States Department of Energy
VAV	variable air volume
VBA	Visual Basic for Applications
VEE	validation, estimating and editing
VFD	variable frequency drive

[1] ECM: A set of activities designed to increase the energy efficiency of a facility (IPMVP, 2002).

[2] ESCO: A firm which provides a range of energy efficiency and financing services and guarantees that the specified results will be achieved under an energy performance contract (IPMVP, 2002).

[3] M&V: The process of determining savings using one of the four IPMVP Options (IPMVP, 2002).

Chapter 1: Introduction

1.1 The Business Case

The Energy Performance Contracting industry is a complex multi-billion dollar industry based upon considerable uncertainty in the data on which decisions are made. The basic model for an energy performance project is that an entity, typically a public one, originates a loan to pay an energy services company (ESCO) for capital equipment upgrades on its property. An ESCO is a firm which provides a range of energy efficiency and financing services and guarantees that the specified results will be achieved under an energy performance contract (IPMVP, 2002). The energy efficiency of the post-retrofit environment is guaranteed to generate an amount of financial savings over a given period so that the borrower, using those same savings, will be able to meet its debt service obligations. In the case that the performance of the energy savings is below the guaranteed amount, the ESCO would then make up the difference with a payment to the customer equal to the shortfall.

Much has been written about the modeling of energy savings, but most of the literature typically focuses on the pre-retrofit period and provides methods for calculating energy savings based upon those pre-existing conditions. Although an ESCO may have internal processes in place for reviewing and assessing the risk associated with a project prior to implementation, the engineering models project only best estimates of potential performance. In other words, "when the rubber meets the road," the assumptions in the models may fall short in the real world.

The retrofits, which are commonly referred to as energy conservation measures (ECMs), will have varying degrees of interdependence. ECMs are sets of activities designed to increase the energy efficiency of a facility (IPMVP, 2002). They could be operational changes, equipment upgrades, and improvements to maintenance practices. Accounting for the energy performance in a post-retrofit environment can be very problematic. This may be due to several factors, such as operating characteristics of the building, which were unknown to the engineering team during the development phase, changes to the physical building that may be made outside the scope of the project, modifications to the operating conditions of the building which affects the internal load, and many more. For the energy management team responsible for managing the risk associated with ten to twenty years of contingent liability, this can be extremely challenging.[4]

One powerful tool which can help a team with the ongoing monitoring of a building's energy usage is a system of fault detection. This is typically accomplished using software, commonly known as a "rules engine," which can evaluate data sets and generate alarms based upon pre-defined business rules. For example, if a building has several ECMs completed in it, and their interdependence is so great that it is impossible

[4] Contingent liability is the liability resulting from the future risk associated with the possible payment due to a shortfall with the guarantee. The guarantee is reconciled annually, so the liability is therefore contingent upon the positive performance of the program in future years.

to calculate their savings individually, the best monitoring approach may be one involving the whole building using interval data that has been calibrated to the building's utility meter to report energy usage.[5] If this interval data were uploaded to a central server for analysis on a daily basis, ideally more frequently, a set of rules could be run against the data to determine the existence of any abnormal conditions. Some of these conditions might be strings of zeros, unusually high values, negative numbers and null values. If a condition considered abnormal were met, the system would then generate an alarm to notify the energy management team which might lead to possible investigation.

When dealing with an entire building which is very dynamic, the question becomes, what is the appropriate level for determining if a value is abnormal? The problem is that if a threshold is set too high, then an interval of high usage, which should be investigated, may not be detected, resulting in a lost opportunity. Conversely, if the threshold is set too low, then the alarm recipient might become numb to their arrival and ignore them, even though some percentage will be worthy of further investigation. The ideal situation is then to have the thresholds set at an appropriate point.

---

[5] Interval data is time stamped data collected at regular intervals used to measure a commodity over time. The data could represent many things, such as kWh, power factor, voltage, gallons of water, pounds of steam, etc.

1.2 The Problem

Although the performance guarantee is based upon projected savings, those savings don't always materialize, so it is very important to begin to monitor and track a building's post-retrofit usage as early as possible in the first year of the performance period. This is especially critical since the building has just undergone many changes.

The pre-retrofit models used for the engineering of the projects are typically based upon at least one full year of usage data, often more. This usage period is known as the baseline period. Once the energy efficiency retrofits are complete, this baseline data is nothing more than a reference point, since the energy profile of the building will have been modified.

As mentioned previously, part of the ongoing monitoring program should include thresholds being set for the building for the alarming tool. Ideally, predicted energy usage will be based upon a model with at least one full year of data. Since the new energy usage profile for the building is going to have very little history, it is necessary to develop an improved method to determine where the thresholds should be set, so that the alarming tool would work appropriately during that first year. One of the keys to success is to have reasonable thresholds established in the beginning of the monitoring period so that potential performance issues can begin to be addressed as early as possible. Later on, once a full year's worth of data is available, the process of determining and setting alarm thresholds can become an iterative process and should be revisited, as presumably it would be possible to make more accurate estimates.

1.3 The Study

The purpose of this study is to test a novel decision tool that was developed by the author, to evaluate how well energy usage thresholds can be estimated using short data sets.

Chapter 2:
Background Information

2.1 Performance Contracting

The energy performance contracting business has its roots in the early 1970's after the oil embargo, when the spike in energy prices drove up interest in conservation. At that time, among other concerns, such as national security, there was a realization that energy efficiency was not just about being "green," but also about reducing the burden on constrained operating budgets that were being squeezed by higher energy costs.

Performance Contracting is a diverse industry with a multitude of contractors, who typically provide services to publicly funded entities. The range of services provided varies widely and includes energy efficiency, water conservation, and energy education. Over the years it has largely been driven by the need for public entities to upgrade capital equipment and better manage their resources.

Although the business has evolved over the years, and the specifics can vary by state and by project, the typical arrangement today is one of a performance contract. The energy performance contract is an agreement entered into between an owner/operator, which is typically a public entity, such as a city, a school district, a correctional organization, or housing authority, and an ESCO. The scope of the work will vary, but the Owner will receive equipment upgrades which increase the operating efficiency of the building. The list of ECMs often includes items such as lighting, boilers, chillers, and building management systems. The financial savings that will be achieved by performing the retrofits are guaranteed to be great enough over the life of the project to pay for the costs of the retrofits, including the debt service on the construction loans. The guarantee component of the project differentiates these projects from capital expense efficiency projects. The third party in this arrangement is the financing institution that provides the funding to the Owner in order for the project to proceed.

2.2 The Energy Guarantee

The guarantee is the vehicle which separates a performance contract from a simple energy efficiency project. It is a separate component from the financing, and is used to reduce the risk of the owner/financier by ensuring the ability of the owner to perform on the debt service. It is based upon the financial performance of a project over its life and guarantees that the energy efficiency project will achieve a certain level of performance, enabling the borrower to perform on the debt service. If the guaranteed savings are not achieved, the shortfall would then be made up through financial remuneration from the ESCO to the owner.

Energy savings and energy cost savings are separate terms, which are usually defined within a contract, and refer to use avoidance and cost avoidance, respectively. These are important distinctions which have different impacts. For example, over the life of a project there will usually be a normal rate of degradation in the performance of the installed equipment. However, at the same time, there will also most likely be an increase in the cost of energy over that same period. Ideally, over the long term, the two will balance each other out.

The pre-retrofit performance is considered to be the baseline period and can be thought of as a reference point for what would be consumed without the retrofit project. At its most basic level, the energy savings is calculated as the difference between the measured pre-retrofit performance of the equipment minus the actual post-retrofit usage of the equipment. This can be stated as:

$$Energy\ Savings = (Baseperiod\ Energy\ Use) - (Post - Retrofit\ Energy\ Use) \pm\ Adjustments$$

The energy savings, or use avoidance, is then multiplied by the established energy unit rate to calculate the cost avoidance. Assumptions about how the equipment would have operated are part of the original energy savings modeling, and can include variables such as operating schedule, occupancy patterns, weather, and electrical loads. A change in any of these factors may create a need to adjust the baseline. Adjustments to the baseline are therefore made in order to standardize the two time periods to a similar set of conditions. Stated another way, a baseline adjustment is used to account for what the usage of the building would have been if the retrofit work had not taken place but the same change in operation had been made.

2.3 International Performance Measurement and Verification Protocol

The Energy Policy Act of 1992 mandated that the Federal Government achieve a certain level of efficiency in its energy utilization (EPAct 1992). In 1994, the U.S. Department of Energy, tasked with determining how to manage the large scope of this work began to lead the effort to reduce the uncertainty in the performance contracting business by bringing together many business partners and industry experts throughout North America. Two years later, the first edition of the collaborative work was released. This is known as the International Performance Measurement and Verification Protocol (IPMVP). Today, the U.S. government is the largest consumer of energy performance contracting work.

A major factor in the drive to create a framework was the need for one standard for the three parties - the owner, the ESCO and the financial institution - to utilize when negotiating details of a project. Some of the questions raised were: How would the owner and financer be certain that the projected savings would be achieved? What certainty would the financer have that the borrower would be able to generate the savings stream to repay the debt? How could the ESCO reduce the owner's and lender's uncertainty with their measurement and verification (M&V) plan, when there was no standard language or framework for defining and outlining an M&V Plan?

The development of a framework within which lenders, buyers and sellers could communicate helped to increase understanding among the parties and effectively reduce the risk and level of uncertainty being assumed by the parties. In the years that followed the release of the first edition, the IPMVP became the de facto reference for outlining program savings. It has been translated into several languages and is used around the world. Although some other, more prescriptive M&V documents have been developed, such as the US Department of Energy's Federal Energy Management Program (FEMP) and ASHRAE's Guideline 14P, the basis of their contents comes from the IPMVP, maintaining the Protocol as a key reference for M&V.

When verifying the performance of an ECM, there are different approaches to measuring the energy savings. Each method will have different levels of accuracy and cost, and those two variables must be weighed against the amount of risk associated with the ECM in order to help determine the best approach. The methods would be laid out in the M&V Plan and agreed upon by all parties. The IPMVP therefore creates a common language to clarify expectations. The protocol defines four M&V Options.

2.3.1 Option A: Partially Measured Retrofit Isolation

For Option A, there is only a partial measurement of the ECM, with one or more of the parameters being stipulated. The agreed upon assumptions, or stipulations, would typically be well known and will be such that their impact on the ECMs savings will not be significant. In addition, the stipulations should be parameters that are outside the control of the ESCO and are assumed to be relatively constant. The measured variables would be those that are important to determine the performance of the ESCO's work on the ECM. For example, if the reduced load comes from a lighting retrofit, the operating hours might be stipulated since they are controlled by the owner. However, the fixture count, and the loads coming from the ballasts and lamps would be measured.

One of the benefits in utilizing an Option A plan is that it reduces the overall cost of verifying the savings. Typically, the stipulated parameter will be one which is much less costly to obtain than the measured variable.

2.3.2 Option B: Retrofit Isolation

This Option is typically much more complicated than Option A. With Option B, there is ongoing measurement of the retrofit with all of the ECM’s parameters being measured. Since there are no stipulated parameters, Option B will typically be more costly than Option A. However, the additional cost of full measurement may be important when there is a higher degree of uncertainty in the potential savings as a result of the greater variance among the parameters.

For example, Option B might be preferable in the case of a photovoltaic installation, where there is net metering in place, and the impact of weather on the power generation is significant. The isolation of a chiller plant presents another example. This equipment will usually have its load vary throughout the day and year, and is dependent on many variables such as space occupancy and outside air temperatures.

There are several reasons why the partners may choose to use Option B. One is that the interaction of the ECM with the rest of the building may not be important. In addition, sub-metering may be necessary to guarantee state and utility rebates. Lastly, the potential savings from the single ECM are not great enough in proportion to the whole building’s usage to be a part of an Option C program.

2.3.3 Option C: Whole Building

This approach uses the metering of a whole building to determine the collective performance of the ECMs. Although there may be sub-meters within the building to isolate the performance of specific systems and help with the ongoing monitoring, the performance of individual ECMs is not reported; only the performance of the whole building is reported. Any changes to the building's operating characteristics would need to be evaluated to make necessary adjustments to the energy baseline; that is whether they impact the baseline positively or negatively. Some of the building parameters to track would include issues such as operating schedule, occupancy, electrical plug load, and HVAC set-points and scheduling.

Option C would be used when the potential energy savings are significant enough compared to the energy consumption of the entire building. A general rule of thumb is that the energy savings should be, at the very least, 10% of the baseline energy usage. Option C might also be employed if there is considerable interaction between the boundaries of the individual systems. An example of this might be a building where multiple ECMs were performed, such as the installation of lighting, a new high efficiency chiller, variable air volume boxes, high efficiency water pump motors with variable frequency drives (VFD), and lastly, an energy management system to control operating set-points and schedules. In this case, although the energy savings may have been calculated and assessed separately for each ECM as part of the overall project, their interactions would be far too great to report on individually. Lastly, Option C may also be employed if there are too many ECMs within a building such that reporting on them independently would not be cost effective.

The reporting of energy data can be performed by using interval data collected from a pulse relay connected to the utility's energy meter, or with data from the utility bill. Each of these has benefits and disadvantages. When receiving a pulse from the utility's meter, there is potential for a higher level of confidence in the reported performance of the building as there will be a lot of interval data which the ESCO and owner/operator can use for analysis. However, no system is perfect, and in the event of missing data, there is difficulty in reporting usage for time periods which do not overlap the meter's service period. Missing data would then need to be estimated or interpolated through a predetermined procedure. This procedure would need to be explained in detail in the M&V Plan. In addition, if variables, such as peak demand or power factor are being reported as part of the savings program, it is imperative that the data be collected in the same manner that the utility captures them.

When using the utility bill data, there is an assumption that the bill data are accurate. Sometimes bills have estimates, which may not be clearly stated on the bill. For an individual month, the impact may be great, however, over time this would be corrected once the meter is actually read, and then actual usage could be reported by backfilling the missing data based upon meter reads. One of the biggest disadvantages of not having interval data and relying on the few data points provided by the utility's bill is that there may be great uncertainty in how the building is actually performing. For example, without interval data, there is no way to graphically view the building's load to help determine if the characteristics of the load may have changed over time.

Independent variables, such as weather, non-ECM equipment load, occupancy and operating hours need to be specified in the M&V Plan. These factors will be critical to determining future baseline adjustments. Often these variables can be modeled using statistical methods such as regression analysis.

2.3.4 Option D: Calibrated Simulation

With Option D, the building or the ECM is modeled using computer simulation software to calculate the impact on energy usage for either the baseline period or the post-retrofit periods. Typically, the models employ hourly calculation techniques. In order to ensure that the estimates provided by the modeling are reasonable, there must be some method for calibrating the model for testing. The simulation can be used to test the individual ECMs or for all of them combined.

One of the keys to successful modeling is the user's understanding of the software being employed. In addition, the adage, "garbage in, garbage out" applies here. The quality of the input data is critical and must be thoroughly reviewed. Any modifications or corrective procedures performed on the data should be clearly stated as part of the modeling process.

2.4 Data Diagnostics

Collecting energy interval data from utility meters or trend data from an EMS can lead to enormous quantities of data. When this amount is multiplied across hundreds of meters along with a multitude of EMS systems, each with hundreds of points, the volume of data can quickly become too cumbersome to manage, visualize and analyze without some form of automation. A lack of automation causes the data's value to diminish as they are rendered useless due to sheer magnitude.

2.4.1 Rules Engines

In recent years, a new class of software, called a rules engine, has been developed for business applications (Chisholm, 2004). This type of software has simplified the software development process by having a programming team tasked with only creating a platform, which can then have sets of business rules created and modified by an end user such as a business analyst (Chisholm, 2004). By removing the rules from within the software code and out of the control of IT professionals, it can improve the speed of deployment, enable rapid updates to the rules, and increase the flexibility of the software for the end user. A by-product of the rules is that based upon conditions being met, workflows can then be created. For example, if the engine discovers that a condition is met, such as high electrical usage in a building, an alarm might be sent via email to notify the appropriate user.

Rules engines may be confused with data mining and artificial intelligence tools, but are actually quite different. The former would be formed around known parameters defined by business practices, while the latter two are looking to extract patterns from large data sets or generate inferences without the engine necessarily posing a specific question (Chisholm, 2004).

2.4.2 Fault Detection and Diagnostics

Within the energy management industry, the application of these tools is commonly known as fault detection and diagnostics (FDD). A system fault is typically defined when a system, or one of its components, operates outside its design range. This may be caused by hardware failure, maintenance troubles, system programming errors or a number of other things. Some cases where a fault detection rule might be applied are:

- verifying building zone temperatures during an unoccupied state to confirm a building's inside temperature is drifting to its unoccupied set point;
- checking the energy usage of an air handling unit (AHU) during an occupied period while the outside air temperature is below a certain range to verify that the system is cycling appropriately;
- ensuring that the flow rate in a steam condensate return meter is high enough to satisfy a minimum rate of steam loss within a distribution system.

An extensive study performed by the U.S. Department of Energy (US DOE, 2005 estimated that approximately 11% of the annual energy consumed by lighting, HVAC and refrigeration systems in large commercial buildings was a direct result of systems faults. This amount is quite significant and is projected to be equivalent to about one quad of energy.[6] Although they analyzed over a dozen types of faults, they found that three faults alone – duct leakage, HVAC left on when space unoccupied, and lights being

---

[6] A quad is shorthand for an amount of energy equal to a quadrillion or 10^{15} BTU.

left on when space unoccupied – accounted for approximately 68% of that figure (DOE, 2005). These and other issues can go undetected for long periods of time, which makes their detection such an important part of the management of the energy performance guarantee. Although this study does not analyze these types of faults directly, with the first level of alarming that is being reviewed here, responding to those alarms more adequately is critical in facilitating fault detection.

2.5 The Study

One ESCO with a national market position utilizes a rules engine to test thresholds. However, the ESCO has a difficult time setting the thresholds at appropriate levels. The process currently involves not much more than best guesses, which are modified over time. They have developed their own rules engine software tool for performing a wide array of diagnostics called Facility Diagnostics. Although, they also collect trend data from various EMS systems and weather stations, the data, which are of primary interest, are the energy interval data received from all over the country by way of either a direct pulse from utility meters or meters installed by the ESCO. The greatest number of these points comes from electric meters, but depending upon the specific needs of the M&V plan, they may also include natural gas, propane and water meters.

For the ongoing monitoring of Option C sites, one of the most common diagnostic rules which the company's energy management team has implemented for existing projects is a rule which checks if energy usage or demand has exceeded a prespecified threshold. This initial alarm provides opportunity for additional investigation. A key factor to the successful implementation of this rule is that the thresholds must be set to appropriate levels. If the threshold for alarming is too low, then the people involved in support of the energy monitoring might begin to ignore the notifications. Conversely, if the threshold is too high, then abnormal usage patterns which should prompt further investigation would be missed. Therefore, finding the appropriate threshold level is critically important.

The equipment used to collect the interval data is typically installed during the energy project's implementation period alongside all of the other energy conservation measures. A consequence of this is that the energy calculations performed during the development phase most often are not based upon any interval data, and almost all of the baseline data comes solely from utility bill data. Once one to two years of interval data have been collected and the energy manager has a strong working knowledge of the building's operation, it is then obviously much easier to establish limits for the alarm thresholds by analyzing available historical data. Absent this, a key question then is how best to set the threshold levels on a new project, which has no more than a few months of regular interval data. To begin to try and answer this, the IPMVP provides the following statement:

> "For buildings, one or more full years of energy use and weather data should be used to construct regression models. Shorter periods introduce more uncertainty through not having data on all operating modes. The best predictors of both cooling and heating annual energy use are models from data sets with mean temperatures close to the annual mean temperature. The range of variation of daily temperature values in the data set seems to be of secondary importance. One month data sets in spring and fall, when the above condition applies, can be

> better predictors of annual energy use than five month data sets from winter and summer.” (IPMVP, 2002)

This statement came from a body of work by researchers at the Texas A&M Energy Systems Laboratory (Kissock, Reddy, Fletcher, and Claridge, 1993). Utilizing parts of Kissock’s procedure, the question becomes: Can this basic assumption be extended to other buildings, whose operating conditions differ from those used in the original study?

The most important component of this study that comes from the Kissock study is the use of short data sets. Short data sets are defined as 1-month, 3-month and 5-month sets of continuous data. Each of the data sets was compared to the annual usage.

There are several parameters for this study which differ from Kissock’s. The buildings used in the original study had constant volume (CV) air distribution, while the data in this study was collected from buildings in Jacksonville which all had variable air volume (VAV) boxes. In the original study, the analysis was performed by only looking specifically at heating and cooling energy use and removing the internal load of the building from their model. In this study, the whole building’s usage is utilized for the prediction method. While all of the heating in the original study was from natural gas, most of the heating and cooling needs for the buildings in this study are provided by electric power from the local utility. Lastly, Kissock did not segment the energy data. For this study, the data are segmented into four groups, based upon day type and occupancy status. These time periods are then: weekday occupied, weekday unoccupied, weekend occupied and weekend unoccupied.

2.6 Data Management

For all of the electric meters, time stamped 15-minute interval data are collected for calculation of energy savings. The data come from pulse initiators that are connected to the meter and are installed by the local utility. This relay sends a pulse based upon a fixed usage that is collected by a data acquisition system (DAS). The size of the pulse is predetermined, and is then sent to a central server for storage, analysis and reporting.

The quality of the data is an issue and involves an ongoing process. The first step is to confirm that the data are accurate through calibration. Using meter read dates and estimated meter read times, it is possible to sum up the interval data to get a value that can be tested against the utility's bill. The ESCO that uses the data has an established procedure that, for reporting purposes, uses an acceptable error rate of +/-2%. The data are also re-commissioned regularly, typically at least once a year, to ensure that the pulse multiplier is accurate.[7]

On an ongoing basis the data are checked through a process that includes validation, estimating and editing (VEE). This is a set of requirements that verify the accuracy of interval data to make corrections on a frequency, ranging from real-time to daily. VEE is used to ensure data have been generated according to specifications, satisfy acceptance criteria, and are appropriate and consistent with their intended use.

---

[7] The pulse multiplier is the predetermined value, depending on the utility meter that is assigned to a pulse.

2.6.1 Calibration

A primary concern with metered data is the accuracy of the readings. As mentioned above, the ESCO which collected the data utilized in this study has a method for calibrating the data, where the standard is to achieve readings within +/- 2% of the utility's bill. The key factor impacting the calibration process is that it is being calibrated to the utility's bill. This has a few main issues. The first is the accuracy of the utility's own meters. The local utility, where the buildings are located for this study, states in its tariff sheets that they guarantee accuracy to within +/- 2%.[8] Due to the inability to test the meters, the assumption is that the readings provided by the utility are indeed accurate. The second issue is that if a meter is being read in person, there is no way to know the exact time of day that the meter read is taking place. To provide a best guess, the estimate used for testing purposes is that the read occurs at noon. With a few to several months of data to calibrate, the slight variation will wash out over time. A third issue comes from estimated reads. Although these may occur, they are typically corrected in a month or two, once an actual read finally occurs. Again, these errors will rectify themselves with time.

2.6.2 Missing Data

With any network, there are multiple points of failure. When part of the network fails, the impact to the data quality can vary and will cause varying degrees of data loss. One common occurrence is that a meter will get replaced. Sometimes when

[8] http://www.jea.com/about/pub/downloads/ElectricTariff-LEGAL.pdf

this happens, the wires providing the pulse from the meter's contacts may not be terminated correctly, or not at all. Another is that there is a transmission failure, which causes data to be lost. This could be with a cable on site carrying a signal that may be cut, a wireless radio failing, or a network issue, where the local IT team has modified its network schema, rendering communication from the remote to the central server impossible. Ideally, this is discovered within no more than a few days, and addressed rapidly. However, the result is that the data on the central server during this down-time are either non-existent, show up as a large spike or end up simply being a string of zeros. In either case, the data for the days in question would need to be corrected.

The two most common occurrences are blank periods and usage spikes. With blank periods, the data have to be filled in using an estimate. If the gap was less than a month, then usage from the week or two immediately before and after was copied and used to fill in the blank space. If the length was greater than a month, then usage from the previous year in the same time frame was copied over. For the case of data spikes, a process of linear interpolation is used. This simply counts the number of missing intervals and divides the data spike by that number, effectively spreading the usage uniformly over the time period.

2.7 Summary

The purpose of this thesis is to develop a procedure, using similar methodology to the Kissock study which could be used to test alarming thresholds on interval data. The model created uses Excel spreadsheets with Visual Basic for Applications (VBA) programming. Although the basis of the Excel model created here differs from the original work by Kissock, et al., 1993, this may find many applications in the realm of decision making with threshold alarms.

Chapter 3: Methodology

3.1 Building Descriptions

The set of buildings being used for the study are five government buildings located in the downtown area of Jacksonville, Florida. Four of them have operating schedules and occupancy patterns which are fairly regular and consistent with energy load factors that are relatively stable throughout the year. The fifth building is a convention center and has a much more varied and unpredictable utilization which results in a much lower and more varied load factor.

Three of the buildings have their cooling needs met by chilled water, while the exterior zones of the building have VAV boxes that are fan powered with electric heating elements. The fourth building also has chillers providing chilled water for cooling, while two natural gas (NG) fired boilers provide hot water. Most years, the boilers operate less than 30 days per year. The last building has eight package rooftop units (RTU), which are also equipped with natural gas fired furnaces. The five buildings have a total of eight electric meters which are all being used in the study.

Table 1: Summary of Buildings

Building	Cooling	Heating	# of Meters
Building 1	Chilled water	Electric heating element in VAV	2
Building 2	Chilled water	Electric heating element in VAV	2
Building 3	Chilled water	Electric heating element in VAV	1
Building 4	Chilled water	Hot water from NG boilers	1
Building 5	Rooftop package units	Natural gas furnaces in RTU	2

In a normal year, the heating degree days are 1,354 per year, while the cooling degree days are 2,627.[9] For 2006 through 2009, the absolute deviation from the annual average temperature for the shoulder months of April and October were -2 and 2, respectively. These data can be seen in Table 2 and Figure 1.

Table 2: Monthly and Annual Mean Daily Temperatures

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Monthly	54	54	61	67	74	80	82	83	79	71	61	58
Annual	69	69	69	69	69	69	69	69	69	69	69	69
Difference	(15)	(14)	(8)	(2)	5	12	13	14	10	2	(8)	(11)

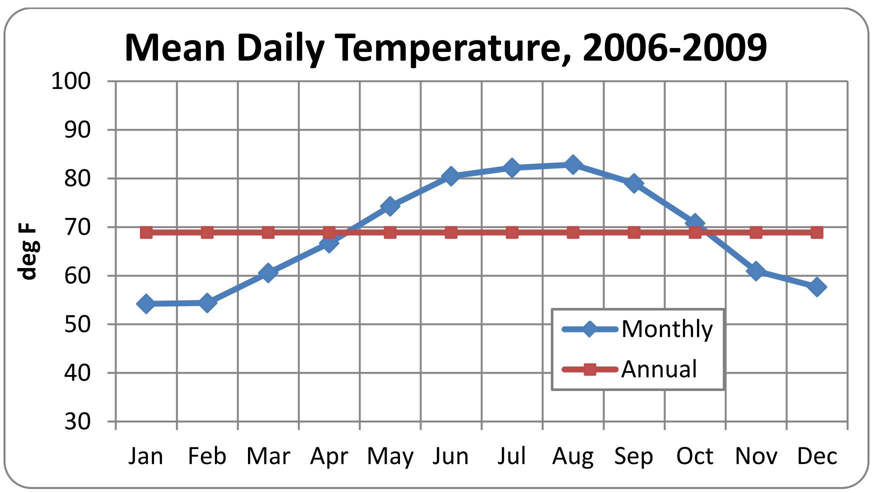


Figure 1: Mean Daily Temperature for the Months and Years of 2006-2009

3.2 Data Summation

Once the data were validated and any necessary corrections were made as discussed in Section 2.6, the interval data needed to be summed up into daily values so

[9] Weather data from Jacksonville International Airport, NOAA Weather Station 13889

they could be used in the decision model. This was done by developing a tool in Excel which would aggregate the data using VBA. (See Appendix B.)

The interval data were summed up by day according to period type. The period types were based upon building occupancy and day type. The resulting four period types were:

- Weekday Occupied
- Weekday Unoccupied
- Weekend Occupied
- Weekend Unoccupied

The primary reason for wanting to study these four periods individually is that in a performance guarantee with ongoing Option C whole building measurement, there is a contractual document called the “Standards of Occupancy and Control” (SOC). This contract schedule is a table that states how the building is supposed to be operated in regard to HVAC operating schedules and temperature set-points. The values of these parameters are the same as those used for the post-retrofit period in the original energy savings calculations. Therefore, it is very important to break out the building’s energy profiles for the four period types. All of the occupancy schedules used in the Excel tool came from the SOC.

In addition to calculating the sum of the interval data for the daily period types, other values were calculated, such as the number of data points, the average for the interval data, the minimum and maximum values, plus the standard deviation. These were calculated on a daily basis as seen in Figure 2.

Date	Average Daily Temp	Weekday Occupied Count (# intervals)	Sum (Usage per period)	Average (usage / interval)	Min (interval / period)	Max (interval / period)	StDev (interval / period)
Fri 9/1/2006	84	40	3,580.13	89.50	78.38	98.25	3.65
Sat 9/2/2006	84						
Sun 9/3/2006	84						
Mon 9/4/2006	82	40	2,831.50	70.79	68.75	73.13	1.06
Tue 9/5/2006	82	40	3,860.63	96.52	87.13	106.13	4.33
Wed 9/6/2006	83	40	3,928.38	98.21	87.88	108.00	3.84
Thu 9/7/2006	78	40	3,346.50	83.66	57.88	96.50	12.72
Fri 9/8/2006	78	40	3,091.63	77.29	50.00	89.38	14.64
Sat 9/9/2006	79						
Sun 9/10/2006	79						
Mon 9/11/2006	79	40	3,044.13	76.10	48.50	91.75	15.70
Tue 9/12/2006	80	40	3,222.00	80.55	50.00	100.75	19.25

Figure 2: Sample of calculated data for the Weekday Occupied period type.

The data were then calculated for the 1, 3 and 5-month windows as shown in Figure 3.

	Average	Weekday Occupied						
Date	**Daily Temp**	**Include**	**Count**	**Sum**	**Average**	**Min**	**Max**	**StDev**
Nov-06	60.2	Yes	880	49,598.41	56.36	26.20	70.60	10.13
Dec-06	60.5	Yes	840	47,008.73	55.96	28.80	69.90	8.82
Jan-07	56.8	Yes	920	54,204.59	58.92	30.30	74.80	9.09
Feb-07	54.6	Yes	800	55,956.02	69.95	30.90	108.00	18.33
Mar-07	63.0	Yes	880	73,973.11	84.06	20.80	110.10	17.75
Apr-07	65.8	Yes	840	61,074.02	72.71	16.35	110.20	18.85
May-07	72.1	Yes	920	61,630.11	66.99	30.90	113.80	18.81
Jun-07	78.7	Yes	840	62,807.14	74.77	46.00	103.70	16.97
Jul-07	82.0	Yes	880	74,324.24	84.46	48.90	113.70	17.70
Aug-07	83.2	Yes	920	79,669.96	86.60	48.60	122.40	17.04
Sep-07	79.1	Yes	800	61,454.30	76.82	27.40	103.70	20.49
Oct-07	74.0	Yes	920	83,398.27	90.65	48.60	114.10	14.70

		Count	**Avg Daily**	**Avg Intrvl**	**StDev**
Nov-06	3-Month Windows	2,600	2,374.50	59.36	11.60
Dec-06		2,640	2,285.03	57.13	9.37
Jan-07		2,560	2,455.77	61.39	12.66
Feb-07		2,600	2,832.83	70.82	15.47
Mar-07		2,520	3,031.80	75.79	18.31
Apr-07		2,640	2,979.96	74.50	18.48
May-07		2,600	2,854.02	71.35	18.25
Jun-07		2,640	3,011.54	75.29	17.87
Jul-07		2,640	3,284.87	82.12	17.24
Aug-07		2,600	3,314.59	82.86	18.38
Sep-07		2,640	3,401.86	85.05	17.43
Oct-07		2,600	2,985.23	74.63	15.57

		Count	**Avg Daily**	**Avg Intrvl**	**StDev**
Nov-06	5-Month Windows	4,360	2,529.08	63.23	12.44
Dec-06		4,320	2,449.10	61.23	12.68
Jan-07		4,320	2,599.45	64.99	13.42
Feb-07		4,280	2,731.00	68.27	15.18
Mar-07		4,360	2,815.03	70.38	16.90
Apr-07		4,280	2,948.04	73.70	18.16
May-07		4,360	3,062.46	76.56	18.04
Jun-07		4,400	3,086.41	77.16	17.89
Jul-07		4,360	3,118.22	77.96	18.21
Aug-07		4,360	3,317.93	82.95	17.39
Sep-07		4,400	3,163.95	79.10	16.33
Oct-07		4,360	3,050.01	76.25	15.19

Figure 3: Sample of calculate data for the 1-month, 3-month and 5-month windows by period type.

3.3 The Decision Model

A model was developed using Excel and VBA that calculates a "risk management" value for the energy program. (See Appendix C.) This value is based upon the alarms generated for all of the meters over a period of time along with the response to those alarms. This was done by having the predetermined values for the different sized short data set used as the threshold for triggering alarms on an entire year's worth of data. A flowchart of the model follows:

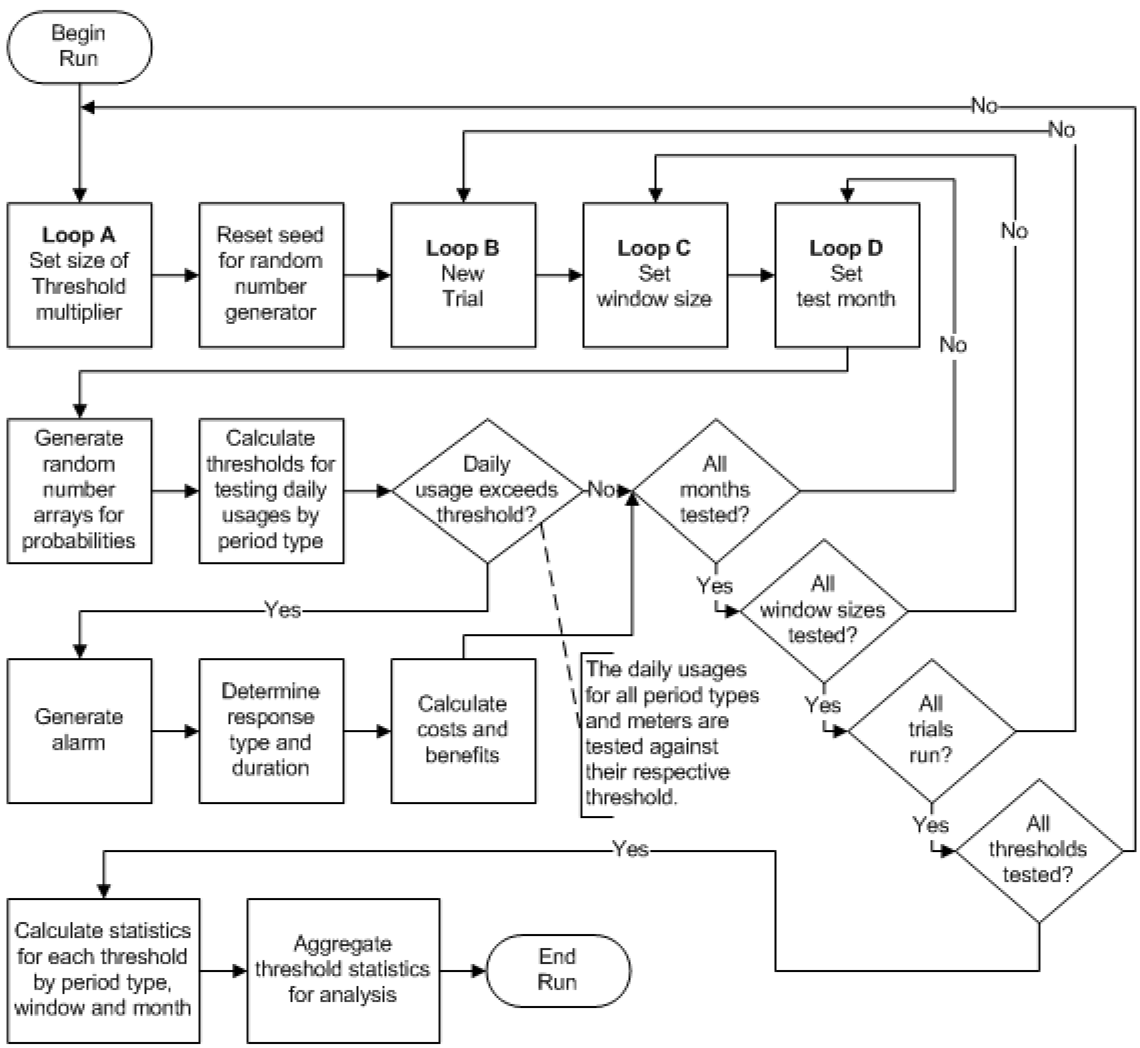


Figure 4: Decision model flowchart

3.3.1 Loops

The first loop was for testing the size of the thresholds. This loop has seven intervals which begin at 1 standard deviation and are uniformly distributed by an absolute change in probability with a maximum value of 2 standard deviations. Table 3 shows the thresholds used for the 7 increments.

Table 3: Schedule of threshold increments

Increment	Probability Threshold	Standard Deviation
1	0.8413	**1.000**
2	0.8640	1.098
3	0.8866	1.209
4	0.9093	1.336
5	0.9319	1.490
6	0.9546	1.691
7	0.9772	**2.000**

At the beginning of each increase in threshold the random number generator was reset with a seed of the same value. This was done in order to ensure that during the following three loops, for all of the trials, as the data are tested against the different window sizes and months, the random numbers used to determine the generation of alarms and the responses to those alarms remain the same.

The second loop was for testing the specified number of trials. For this study, all of the runs had 250 trials. The third loop was for the four window sizes. As mentioned previously, those sizes are 1-month, 3-month, 5-month, and 12-months. The last loop was for the individual month of the year being tested.

3.3.2 Alarm Generation

In order to generate alarms, every day's four period type usages were tested against the daily threshold for each meter in the month being tested. Using the probability thresholds designated in the previous section, the daily threshold for each meter and period type was calculated using the Excel NORMINV function with the interval average and standard deviation for the respective period type. This number was then multiplied by the number of intervals in the period type to arrive at the threshold. If any of the daily period values being tested was greater than its respective threshold, then an alarm was generated.

The data shown in Table 4, provides a sampling of the data points used as inputs in the decision model for the 3-month window. The daily thresholds shown at the top of Table 4 were for the month of December. All of the thresholds were calculated using the Excel NORMINV function.

Table 4: Sample of Weekly Occupied, 3-Month window data, along with daily usages by period type

Period Average	2,285.03	5,625.77	4,528.47	477.33	12,302.24	4,779.03
Normal Probability	84.134%	84.134%	84.134%	84.134%	84.134%	84.134%
Daily Threshold	2,659.72	6,233.77	5,393.40	552.73	14,368.57	6,371.22

3-Month Window Data Avgs	Weekday Occupied					
	Meter1-1	Meter1-2	Meter2-1	Meter2-2	Meter3-1	Meter4-1
Nov-06	2,374.50	5,568.74	4,137.38	456.52	11,078.70	5,351.27
Dec-06	2,285.03	5,625.77	4,528.47	477.33	12,302.24	4,779.03
Jan-07	2,455.77	5,590.28	4,553.41	494.32	12,153.88	4,880.44

3-Month Window Interval Avgs	Weekday Occupied					
	Meter1-1	Meter1-2	Meter2-1	Meter2-2	Meter3-1	Meter4-1
Nov-06	59.36	139.22	79.56	8.78	213.05	111.48
Dec-06	57.13	140.64	87.09	9.18	236.58	99.56
Jan-07	61.39	139.76	87.57	9.51	233.73	101.68

3-Month Window Std Dev	Weekday Occupied					
	Meter1-1	Meter1-2	Meter2-1	Meter2-2	Meter3-1	Meter4-1
Nov-06	11.60	13.16	15.18	1.15	33.68	51.53
Dec-06	9.37	15.20	16.63	1.45	39.74	33.17
Jan-07	12.66	17.72	15.83	1.47	33.82	32.18

Month Count	Weekday Occupied					
	Meter1-1	Meter1-2	Meter2-1	Meter2-2	Meter3-1	Meter4-1
Nov-06	40.00	40.00	52.00	52.00	52.00	48.00
Dec-06	40.00	40.00	52.00	52.00	52.00	48.00
Jan-07	40.00	40.00	52.00	52.00	52.00	48.00

Daily Period Data	Weekday Occupied					
	Meter1-1	Meter1-2	Meter2-1	Meter2-2	Meter3-1	Meter4-1
Wed 11/1/2006	2,184.50	5,790.60	3,492.68	439.59	7,265.42	6,826.54
Thu 11/2/2006	2,435.00	5,632.20	3,497.36	444.27	8,994.08	7,164.50
Fri 11/3/2006	2,147.00	5,398.20	4,276.58	414.69	7,906.54	6,584.48
Sat 11/4/2006	-	-	-	-	-	-
Sun 11/5/2006	-	-	-	-	-	-
Mon 11/6/2006	2,520.00	5,592.60	4,236.34	424.06	7,505.58	3,835.16
Tue 11/7/2006	2,472.50	5,737.30	3,634.49	462.24	12,612.20	4,621.40

The next step was to determine the response to the alarm. This was done by assigning a random probability to the alarm which was used to determine the response type. The response type had four possibilities, which were to

- ignore the alarm,
- make a cursory response,

- perform an extended response, and
- have a response which ended up with a baseline adjustment.[10]

The distribution for determining the response type can be seen in Table 5. Using a second random probability, the duration of the response was calculated using a pre-defined distribution that was assigned to the response type. The distribution of response durations can also be found in Table 5.

Table 5: Distribution for response types, and response lengths

Alarm Response	Value	Upper Limit	Length Min	Length Max	Distribution	Mean	St Dev	Mass Probability
No Alarm	0							
Ignore	1	50.0%						
Cursory	2	80.0%	1	2	Uniform			
Extended	3	99.0%	2	15	Normal	8.5	2	99.899%
Baseline Adj	4	100.0%	15	75	Normal	45	8	99.996%
Baseline Adj Multiplier			3	15	Normal	9	3	98.200%
Threshold Limits			1.00	2.00				

Since the current system for generating the alarms is riddled with many operational challenges, it was impossible to review historical rates of response. Therefore, in order to arrive at the distributions listed in Table 5, several subject matter experts were consulted. These people were professional peers of the author who work with different types of alarms and have broad experience in the energy management field. The numbers used are best estimates of what the rates of response and consequent time spent should be.

---

[10] A baseline adjustment is an accounting adjustment made to the energy baseline which accounts for a change as if the addition of or decrease to the building's usage had occurred during the baseline period. One example might be additional equipment. The additional usage would be added to the baseline so that the energy could be accounted for without affecting the net savings that are part of the energy program

3.3.3 Multipliers

There were three multipliers developed which were used to assist in calculating the risk management value. The first multiplier was the Baseline Adjustment Multiplier (BAM). The BAM was established in order to create additional value for the times when an extended alarm turns into a baseline adjustment, which results in reduced risk in managing the energy guarantee. The BAM was based upon a random probability and was calculated using the distribution shown in Table 5.

The second was the projected multiplier (PM). The intent of this multiplier was to provide weight to meters where there is a larger portion of the energy guarantee at risk. The PM was based upon the projected savings for each building and is independent of the period type. The percent of projected value is,

$$\%PV = \frac{\sum Projected\ Meter\ Savings}{\sum Projected\ Program\ Savings} * 100$$

and

$$PM = \left(\frac{1}{1 - \frac{\%PV}{100}}\right)^{SF}$$

where,

SF Size factor

The size factor was set so that the meter with the greatest amount of projected savings would have a value of 3.00. All the other meters would then be relative to that one. In this study, the size factor was equal to 1.94. The data used in the calculation of the multiplier can be seen in Figure 5.

Projected Multipliers	Meter1-1	Meter1-2	Meter2-1	Meter2-2	Meter3-1	Meter4-1	Meter5-1	Meter5-2
12-month % Projected	1.59%	0.92%	12.59%	1.85%	43.14%	17.57%	19.36%	2.99%
Projected Multiplier	1.032	1.018	1.299	1.037	3.000	1.457	1.520	1.061

Projected Savings	All Bldgs	Meter1-1	Meter1-2	Meter2-1	Meter2-2	Meter3-1	Meter4-1	Meter5-1	Meter5-2
Nov-06	557,901	8,365	3,705	43,873	7,147	299,762	72,061	106,190	16,798
Dec-06	585,606	7,526	4,027	32,328	7,341	351,574	71,745	95,353	15,712
Jan-07	606,483	7,228	4,332	32,968	6,809	364,477	80,005	92,982	17,682
Feb-07	538,888	7,191	4,442	32,044	5,639	328,011	56,628	87,895	17,038
Mar-07	570,842	8,036	4,229	46,825	7,743	306,031	74,125	104,773	19,080
Apr-07	523,428	8,253	5,250	68,925	10,481	200,910	92,082	118,005	19,522
May-07	536,299	9,072	5,875	90,186	11,528	172,510	113,261	116,560	17,307
Jun-07	536,666	10,165	5,754	102,005	11,837	158,010	125,470	108,484	14,941
Jul-07	521,960	10,209	6,349	108,617	12,532	144,626	127,005	99,595	13,027
Aug-07	533,530	10,096	6,282	105,458	14,700	150,357	127,998	105,149	13,490
Sep-07	535,279	9,460	6,076	95,743	13,122	163,866	119,857	112,289	14,866
Oct-07	556,488	9,131	4,486	72,091	13,087	208,280	100,132	131,159	18,122

Figure 5: Data used for the calculation of the Projected Multipliers

The last multiplier was the Load Factor Multiplier (LFM). This multiplier was created with the goal of escalating the value of an alarm from a building with a lower load factor since a lower value is indicative of a greater degree of variability in the building's usage.

The load factor (LF) of a building for a month is calculated as:

$$LF = \frac{\sum kWh}{Peak\ kW * 24 * (days\ in\ month)}$$

A building's load factor is typically calculated using an entire month of data. However, for this study, the energy usages and peak demands for every meter were broken out for each month by period type. This then allowed for the calculation of a load factor for each meter's period type by month.

$$LF_{mp} = \frac{\sum kWh_{mp}}{Peak\ kW_{mp} * (number\ of\ hours\ in\ month_p)}$$

where,

m	Month
p	Period Type

Load factor is a value between 0 and 1. In order to convert it to a number which would provide the desired effect, the load factor was inverted so as to arrive at the LFM.

$$LFM = \frac{1}{LF}$$

Sample data of the load factor multipliers for one period type can be seen in Figure 6.

Load Factor Multiplier	Weekday Occupied							
	Meter1-1	Meter1-2	Meter2-1	Meter2-2	Meter3-1	Meter4-1	Meter5-1	Meter5-2
Nov-06	1.093	1.061	1.207	1.101	1.092	1.262	1.199	1.363
Dec-06	1.093	1.067	1.163	1.103	1.073	1.274	1.183	1.369
Jan-07	1.084	1.072	1.195	1.116	1.125	1.276	1.180	1.372
Feb-07	1.085	1.063	1.211	1.108	1.092	1.264	1.183	1.380
Mar-07	1.094	1.097	1.206	1.101	1.104	1.251	1.185	1.381
Apr-07	1.122	1.084	1.172	1.079	1.088	1.254	1.166	1.411
May-07	1.101	1.059	1.157	1.093	1.081	1.212	1.127	1.413
Jun-07	1.203	1.083	1.169	1.091	1.084	1.228	1.146	1.417
Jul-07	1.128	1.097	1.116	1.093	1.068	1.179	1.125	1.368
Aug-07	1.219	1.080	1.130	1.084	1.073	1.197	1.116	1.361
Sep-07	1.206	1.088	1.157	1.091	1.074	1.266	1.140	1.364
Oct-07	1.130	1.092	1.152	1.098	1.074	1.233	1.154	1.342

Figure 6: Load Factor Multiplier data for Weekday Occupied meters

3.3.4 Risk Management

In order to calculate the "risk management" value, alarms were assigned benefits and costs based upon the response type. The formulas for calculating the various production modes for an alarm are listed in the table below. Depending upon the assigned response, an associated cost and benefit was calculated. Benefits are associated with an alarm being reviewed, while the cost associated with any investigation was equal to the billable time spent on the alarm. However, in the case of ignoring an alarm, while there was no possible reward, the cost was not measured in time, but in lost opportunity. Therefore, the cost of ignoring an alarm was set as equal to the reward of a cursory investigation.

$$RM = \sum Reward - \sum Cost$$

The formulas for calculating the financial impact were:

Production mode	Reward	Cost
Ignore Alarm	0	=(PU-PA)*RT*LFM*PM
Cursory Investigation	=(PU-PA)*RT*LFM*PM	=TM*BR
Extended Investigation	=PU*RT*LFM*PM	=TM*BR
Baseline Adjustment	=PU*RT*LFM*PM*BAM	=TM*BR

where,

PU	Period usage
PA	Period average
RT	Energy rate ($/kWh)[11]
TM	Time spent on investigation
BR	Billable rate ($/minute = $1.25)
LFM	Load Factor Multiplier
PM	Projected Multiplier
BAM	Baseline adjustment multiplier

[11] The energy rates used are the contractual base energy rates and were not escalated.

An example of the output data appears in Figure 7. The first five trials are shown here, however, data from all of the trials were used to calculate the statistics which appear at the top of the table. The statistics for the different thresholds were later compiled into a table with all of the thresholds for analysis.

	Weekday Occupied											
Average	193	28	153	-41	267	40	93	157	37	45	40	115
St Dev	1,683	1,022	1,121	508	1,234	1,083	804	610	645	599	1,073	1,845
Max	7,429	4,006	4,724	3,022	4,772	6,266	6,015	2,969	3,356	3,450	8,324	10,135
Min	-3,113	-2,113	-1,899	-1,109	-2,678	-2,327	-1,085	-890	-1,052	-1,052	-1,295	-3,366
Deviation												
Run Time	3-Month Windows											
3:01:40	1	2	3	4	5	6	7	8	9	10	11	12
Trial #	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
1	(127)	2,114	1,752	(218)	(1,739)	934	385	(370)	457	82	505	2,642
2	1,052	617	(1,288)	506	293	666	(435)	386	(173)	277	(365)	(1,225)
3	341	(189)	(696)	8	(1,705)	638	758	(206)	(365)	276	(67)	510
4	(1,138)	149	(239)	(185)	(699)	(929)	730	(153)	275	151	985	(332)
5	(126)	(899)	(738)	(85)	727	(580)	756	454	(32)	(663)	(67)	851

Figure 7: Sample of risk management values for Threshold 1, Weekday Occupied, 3-month window

3.3.5 Data Analysis

The last step in the process was to aggregate all of the data from the different thresholds into a single data set for analysis. The data was assembled around the four period types, plus a total risk value, along with four window sizes. Figure 8 is a sample of the data for one window size and period type.

			Weekday Occupied											
			3-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.8413	193	28	153	-41	267	40	93	157	37	45	40	115
	2	0.8640	173	-1	105	-36	269	19	96	161	35	53	40	110
	3	0.8866	178	-18	97	8	209	13	86	127	33	58	27	115
	4	0.9093	154	7	47	5	209	6	43	118	33	35	25	106
	5	0.9319	154	7	55	9	197	4	44	60	33	10	25	115
	6	0.9546	75	7	-10	-8	196	27	50	63	45	10	20	121
	7	0.9772	62	-10	-10	11	157	24	29	0	41	0	20	114
Std Dev	1	0.8413	1,683	1,022	1,121	508	1,234	1,083	804	610	645	599	1,073	1,845
	2	0.8640	1,619	768	824	508	1,239	1,074	803	607	648	601	1,008	1,846
	3	0.8866	1,617	462	814	408	1,219	1,052	769	548	642	475	931	1,833
	4	0.9093	1,503	429	514	400	1,212	939	643	536	642	361	909	1,840
	5	0.9319	1,503	429	514	392	1,180	926	633	429	642	152	909	1,828
	6	0.9546	1,142	429	324	330	1,171	658	633	429	567	152	910	1,827
	7	0.9772	962	381	324	271	1,118	646	595	0	557	0	910	1,822

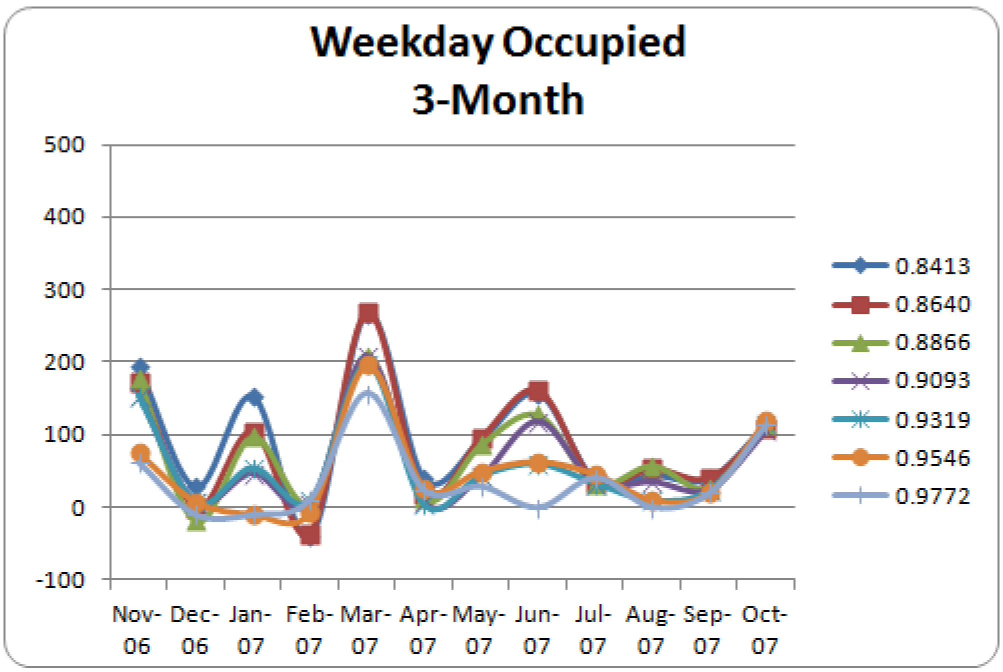


Figure 8: Sample of Weekday Occupied data for all 7 thresholds

3.3.6 Test Runs

For the purpose of generating a base value, the model was run against itself. This means that for each of the twelve months tested, the respective 1-month, 3-month and 5-month windows were used to calculate the risk management metric within the same 12-month period.

In order to test the base model, the test months for the alarms were changed so that any test month would never run against itself using data from its base period. The thresholds for the test run were calculated based upon the base model data. However, for the purpose of generating alarms, the data used to test never overlapped with the base data. The dates used for the base windows and test model can be seen in Figure 9 and Figure 10.

		1-Month				3-Month			
Test Length	24	Base Model Window		Test Model Dates		Base Model Window		Test Model Dates	
		Start	End	Start	End	Start	End	Start	End
Base Model Months	Nov-06	11/1/06	- 11/30/06	12/1/06	- 11/30/08	10/1/06	- 12/31/06	1/1/07	- 12/31/08
	Dec-06	12/1/06	- 12/31/06	1/1/07	- 12/31/08	11/1/06	- 1/31/07	2/1/07	- 1/31/09
	Jan-07	1/1/07	- 1/31/07	2/1/07	- 1/31/09	12/1/06	- 2/28/07	3/1/07	- 2/28/09
	Feb-07	2/1/07	- 2/28/07	3/1/07	- 2/28/09	1/1/07	- 3/31/07	4/1/07	- 3/31/09
	Mar-07	3/1/07	- 3/31/07	4/1/07	- 3/31/09	2/1/07	- 4/30/07	5/1/07	- 4/30/09
	Apr-07	4/1/07	- 4/30/07	5/1/07	- 4/30/09	3/1/07	- 5/31/07	6/1/07	- 5/31/09
	May-07	5/1/07	- 5/31/07	6/1/07	- 5/31/09	4/1/07	- 6/30/07	7/1/07	- 6/30/09
	Jun-07	6/1/07	- 6/30/07	7/1/07	- 6/30/09	5/1/07	- 7/31/07	8/1/07	- 7/31/09
	Jul-07	7/1/07	- 7/31/07	8/1/07	- 7/31/09	6/1/07	- 8/31/07	9/1/07	- 8/31/09
	Aug-07	8/1/07	- 8/31/07	9/1/07	- 8/31/09	7/1/07	- 9/30/07	10/1/07	- 9/30/09
	Sep-07	9/1/07	- 9/30/07	10/1/07	- 9/30/09	8/1/07	- 10/31/07	11/1/07	- 10/31/09
	Oct-07	10/1/07	- 10/31/07	11/1/07	- 10/31/09	9/1/07	- 11/30/07	12/1/07	- 11/30/09

Figure 9: Base model window and test model dates for the 1-month and 3-month windows

Test Length	24	5-Month				12-Month			
		Base Model Window		Test Model Dates		Base Model Window		Test Model Dates	
		Start	End	Start	End	Start	End	Start	End
Base Model Months	Nov-06	9/1/06	1/31/07	2/1/07	1/31/09	11/1/06	10/31/07	11/1/07	10/31/09
	Dec-06	10/1/06	2/28/07	3/1/07	2/28/09	11/1/06	10/31/07	12/1/07	11/30/09
	Jan-07	11/1/06	3/31/07	4/1/07	3/31/09	11/1/06	10/31/07	1/1/08	12/31/09
	Feb-07	12/1/06	4/30/07	5/1/07	4/30/09	11/1/06	10/31/07	2/1/08	1/31/10
	Mar-07	1/1/07	5/31/07	6/1/07	5/31/09	11/1/06	10/31/07	3/1/08	2/28/10
	Apr-07	2/1/07	6/30/07	7/1/07	6/30/09	11/1/06	10/31/07	4/1/08	3/31/10
	May-07	3/1/07	7/31/07	8/1/07	7/31/09	11/1/06	10/31/07	5/1/08	4/30/10
	Jun-07	4/1/07	8/31/07	9/1/07	8/31/09	11/1/06	10/31/07	6/1/08	5/31/10
	Jul-07	5/1/07	9/30/07	10/1/07	9/30/09	11/1/06	10/31/07	7/1/08	6/30/10
	Aug-07	6/1/07	10/31/07	11/1/07	10/31/09	11/1/06	10/31/07	8/1/08	7/31/10
	Sep-07	7/1/07	11/30/07	12/1/07	11/30/09	11/1/06	10/31/07	9/1/08	8/31/10
	Oct-07	8/1/07	12/31/07	1/1/08	12/31/09	11/1/06	10/31/07	10/1/08	9/30/10

Figure 10: Base model window and test model dates for the 5-month and 12-month windows

One last point about the test model is that the base month was run against 2 years of data. In order to normalize the calculated “risk management” values to the 12-month base model, all of the calculated values in the test run were divided by two.

Chapter 4:
Results and Analysis

4.1 Weather Data

During the development of the interval data summation tool, described in Section 3.2, weather data were included for graphing purposes. The weather data came from the NOAA website (www.noaa.gov), and each day's energy usage was graphed against the mean daily temperature by period type. Although the decision model does not account for weather, the graphs demonstrate the usage patterns of the buildings. These graphs could be used by an energy manager to better understand a building's usage by period type based upon outside air temperature (OAT). All thirty-two graphs can be seen in Appendix A. Two interesting results are shown below. The first has a clear inflection point around 73 degrees. This well defined "hockey stick" is indicative of a large piece of equipment, such as a chiller, not being utilized below a certain OAT.

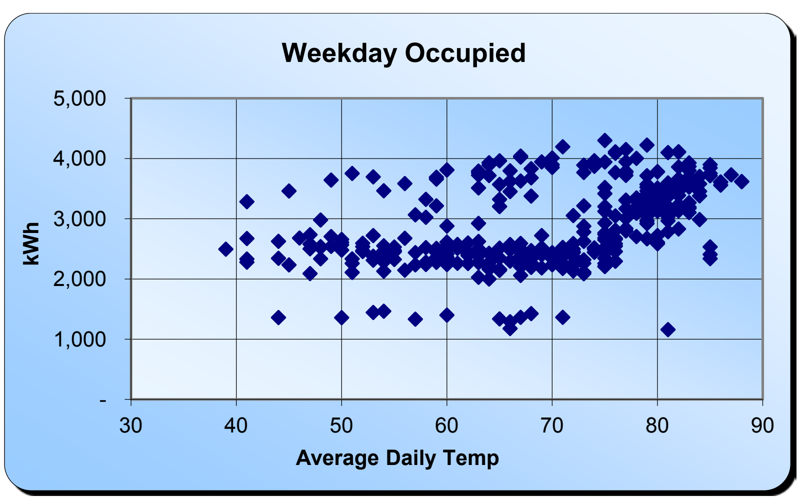


Figure 11: Weekday Occupied plotted against mean daily temperature, Building 1, Meter 1

The second one is worthy of note due to the way that the usage converges above 70 degrees. This building is one which has electric heat in the building's exterior zones. Once the OAT is great enough, the graph suggests that the building is in cooling mode as the usage converges.

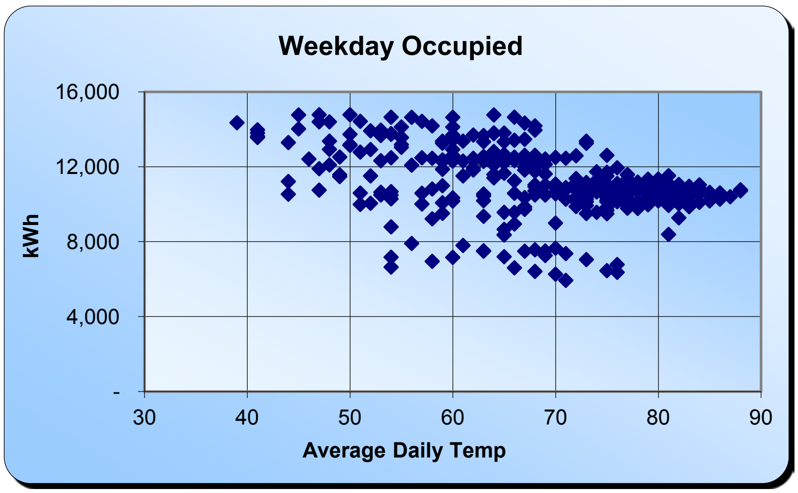


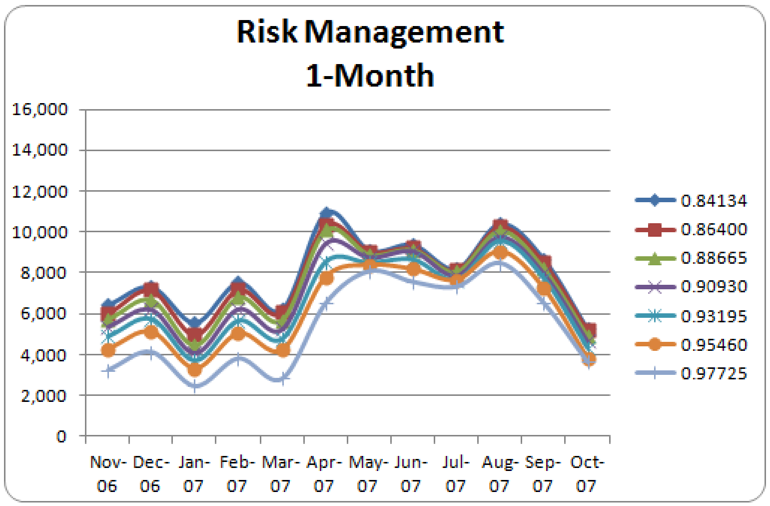
Figure 12: Weekday Occupied plotted against mean daily temperature, Building 3, Meter 1

4.2 Decision Model Results

The following graphs are the results of the decision model Base and Test runs.

		Run Time 2:55:08	Risk Management											
			1-Month Windows											
Max		10,945	1	2	3	4	5	6	7	8	9	10	11	12
Min		858	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	6,457	7,333	5,593	7,537	6,228	10,945	9,123	9,385	8,210	10,411	8,686	5,283
	2	0.86400	6,027	7,159	4,984	7,192	6,067	10,363	9,052	9,215	8,120	10,272	8,504	5,181
	3	0.88665	5,728	6,675	4,454	6,803	5,688	10,101	8,910	9,136	8,060	10,076	8,201	4,886
	4	0.90930	5,388	6,215	4,100	6,246	5,253	9,453	8,767	9,064	7,872	9,781	7,970	4,664
	5	0.93195	4,894	5,742	3,709	5,659	4,769	8,538	8,484	8,635	7,651	9,548	7,728	4,340
	6	0.95460	4,278	5,117	3,325	5,050	4,220	7,815	8,395	8,186	7,663	9,064	7,250	3,809
	7	0.97725	3,256	4,135	2,467	3,813	2,834	6,516	8,063	7,543	7,309	8,461	6,508	3,643
Std Dev	1	0.84134	8,402	9,018	7,074	9,056	8,426	14,117	10,491	11,076	10,324	10,970	10,061	7,454
	2	0.86400	8,228	8,902	6,767	8,912	8,366	13,974	10,379	10,972	10,289	10,876	9,997	7,519
	3	0.88665	8,087	8,621	6,427	8,847	8,260	13,821	10,327	10,919	10,229	10,962	9,844	7,307
	4	0.90930	7,746	8,181	5,935	8,570	8,187	13,411	10,358	10,895	10,050	10,841	9,607	7,078
	5	0.93195	7,404	7,969	5,620	8,181	7,982	12,973	10,264	10,732	10,001	10,717	9,321	6,927
	6	0.95460	7,081	7,746	5,541	7,827	7,594	12,374	10,232	10,571	10,101	10,505	9,116	6,503
	7	0.97725	6,425	7,179	5,069	7,265	6,455	11,404	9,949	10,153	9,855	9,852	8,707	6,187

			Risk Management											
			3-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	7,685	6,063	6,735	5,434	7,255	8,110	9,135	8,730	8,951	9,000	7,754	5,694
	2	0.86400	6,873	5,631	6,037	4,979	6,797	7,974	8,948	8,708	8,920	8,853	7,526	5,351
	3	0.88665	6,465	5,143	5,282	4,671	6,283	7,554	8,698	8,592	8,772	8,740	7,290	4,868
	4	0.90930	5,969	4,788	4,838	3,593	5,676	7,149	8,408	8,558	8,664	8,602	6,930	4,553
	5	0.93195	5,220	4,508	4,216	2,549	5,240	6,477	8,116	8,132	8,553	8,313	6,536	4,125
	6	0.95460	3,675	4,149	3,738	1,504	4,732	6,118	7,863	7,773	8,185	7,698	6,135	3,837
	7	0.97725	1,846	3,097	2,695	858	4,297	5,421	7,183	7,266	7,562	6,919	5,747	3,363
Std Dev	1	0.84134	9,913	7,727	8,253	7,956	9,594	10,779	11,384	10,587	10,672	10,433	9,422	7,610
	2	0.86400	9,322	7,609	7,694	7,512	9,470	10,526	11,249	10,630	10,577	10,325	9,307	7,283
	3	0.88665	9,221	7,388	7,398	7,303	9,160	10,489	11,170	10,514	10,539	10,274	9,054	7,080
	4	0.90930	8,931	7,300	7,270	6,507	8,863	10,130	11,096	10,478	10,491	10,242	8,706	6,643
	5	0.93195	8,485	6,956	6,739	5,398	8,621	9,882	11,042	10,141	10,427	10,043	8,640	6,500
	6	0.95460	6,728	6,659	6,337	4,366	8,480	9,683	10,688	10,087	10,104	9,725	8,449	6,258
	7	0.97725	4,748	6,027	5,568	3,509	7,829	9,299	10,370	9,939	9,881	9,118	8,131	5,913



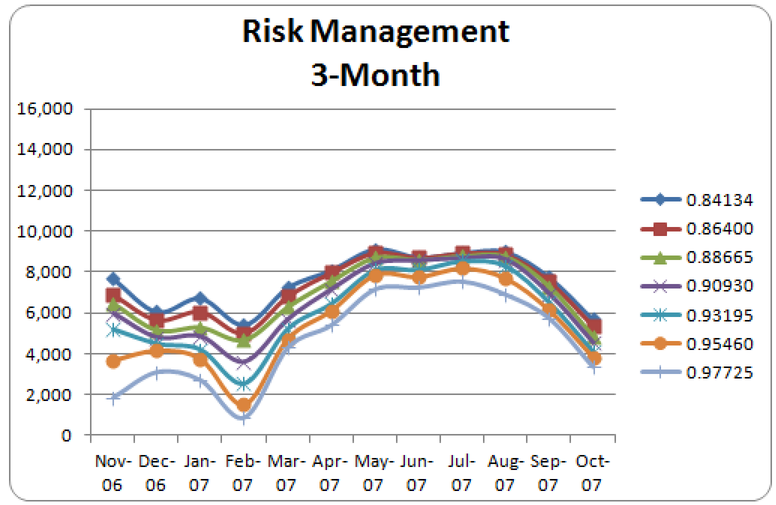
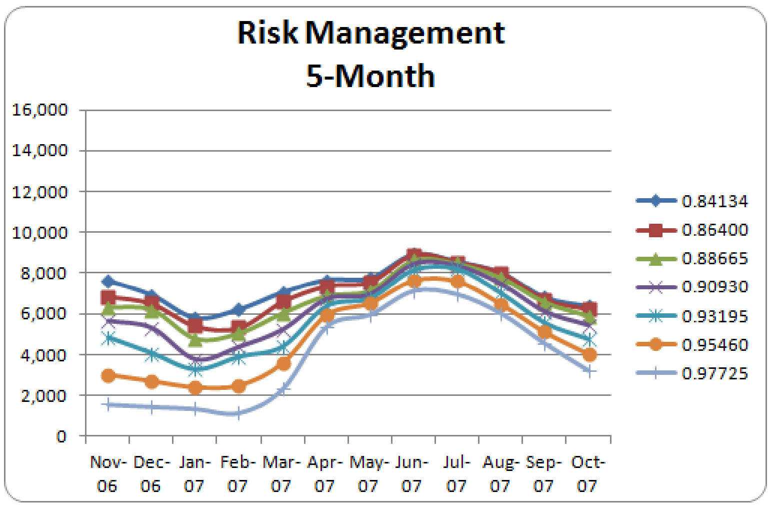


Figure 13: BASE run, Risk Management total, 1-month and 3-month

		Run Time 2:55:08	Risk Management – 5-Month Windows											
Max		10,945	1	2	3	4	5	6	7	8	9	10	11	12
Min		858	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	7,624	6,909	5,791	6,216	7,052	7,660	7,746	8,919	8,549	8,009	6,798	6,357
	2	0.86400	6,860	6,504	5,414	5,335	6,613	7,366	7,558	8,892	8,524	8,006	6,710	6,251
	3	0.88665	6,332	6,189	4,752	5,028	6,031	6,881	7,148	8,656	8,483	7,782	6,565	5,892
	4	0.90930	5,659	5,311	3,778	4,395	5,244	6,736	6,975	8,468	8,371	7,475	6,123	5,451
	5	0.93195	4,843	4,070	3,272	3,898	4,397	6,379	6,764	8,162	8,203	7,012	5,550	4,751
	6	0.95460	3,030	2,717	2,416	2,507	3,606	5,958	6,550	7,663	7,614	6,486	5,115	4,046
	7	0.97725	1,591	1,480	1,379	1,185	2,355	5,379	5,979	7,139	6,966	6,028	4,569	3,259
Std Dev	1	0.84134	9,994	9,237	7,970	8,675	9,345	9,929	10,250	11,029	10,350	9,810	8,771	8,309
	2	0.86400	9,287	8,944	7,457	8,143	9,163	9,915	10,204	10,987	10,327	9,753	8,713	8,197
	3	0.88665	9,204	8,649	6,917	7,851	8,973	9,783	10,059	10,838	10,209	9,619	8,584	8,020
	4	0.90930	8,938	8,022	6,323	7,613	8,306	9,684	9,898	10,757	10,182	9,397	8,288	7,764
	5	0.93195	8,114	6,663	5,793	6,957	8,093	9,449	9,803	10,558	9,976	9,332	8,061	7,362
	6	0.95460	6,010	5,861	5,350	5,813	7,490	9,126	9,714	10,453	9,853	8,873	7,662	6,757
	7	0.97725	4,718	4,335	4,630	3,944	5,289	8,956	9,226	10,162	9,568	8,628	7,214	6,183

			Risk Management – 12-Month											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	6,875	6,838	7,029	6,905	6,894	6,892	6,440	6,434	6,369	6,597	6,715	6,756
	2	0.86400	6,742	6,707	6,887	6,774	6,756	6,750	6,315	6,323	6,257	6,485	6,604	6,647
	3	0.88665	6,486	6,453	6,620	6,519	6,502	6,490	6,080	6,105	6,047	6,268	6,380	6,423
	4	0.90930	5,842	5,808	5,962	5,876	5,853	5,834	5,463	5,500	5,444	5,660	5,760	5,800
	5	0.93195	5,273	5,240	5,377	5,304	5,285	5,262	4,933	4,968	4,919	5,118	5,204	5,244
	6	0.95460	4,434	4,403	4,529	4,459	4,462	4,433	4,182	4,199	4,149	4,307	4,359	4,383
	7	0.97725	2,861	2,838	2,923	2,884	2,884	2,868	2,703	2,700	2,651	2,769	2,797	2,799
Std Dev	1	0.84134	9,264	9,258	9,460	9,297	9,248	9,240	8,784	8,818	8,740	8,942	9,114	9,185
	2	0.86400	9,219	9,212	9,412	9,252	9,198	9,191	8,732	8,761	8,687	8,888	9,063	9,140
	3	0.88665	8,908	8,904	9,094	8,948	8,894	8,884	8,433	8,471	8,402	8,607	8,775	8,856
	4	0.90930	8,445	8,435	8,614	8,490	8,415	8,408	7,975	8,021	7,946	8,152	8,324	8,406
	5	0.93195	8,175	8,162	8,343	8,214	8,158	8,143	7,734	7,774	7,712	7,898	8,058	8,142
	6	0.95460	7,499	7,488	7,658	7,535	7,491	7,471	7,102	7,135	7,074	7,245	7,390	7,463
	7	0.97725	5,621	5,597	5,741	5,644	5,615	5,600	5,332	5,331	5,271	5,411	5,513	5,553



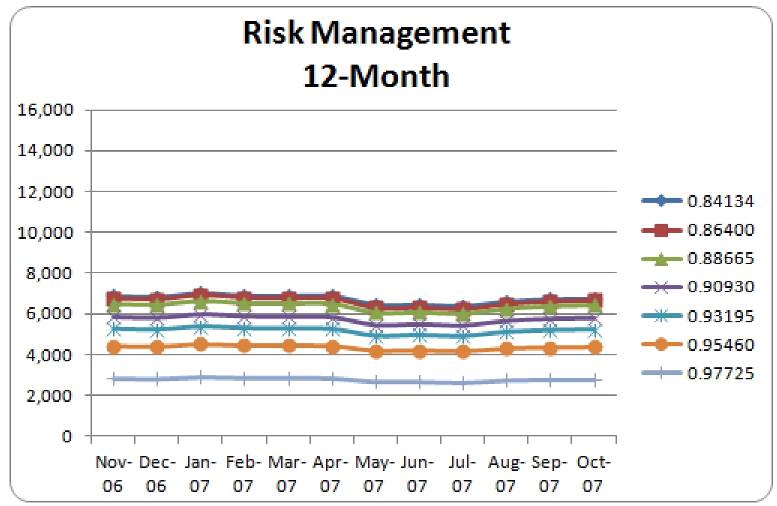
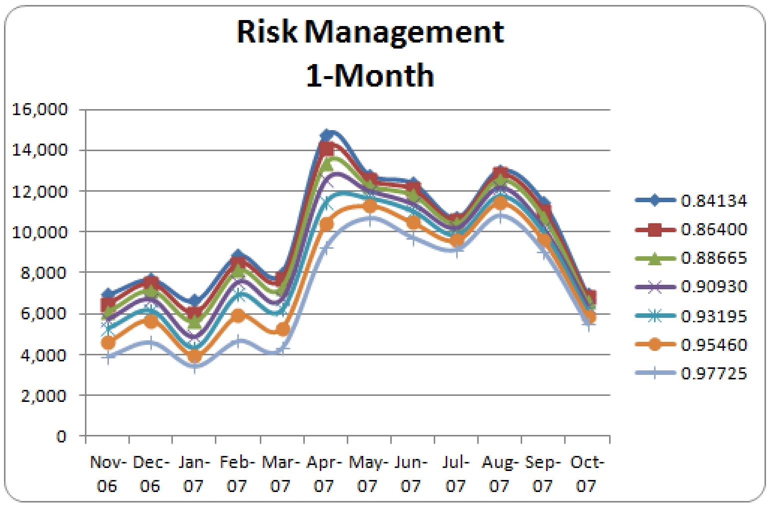


Figure 14: BASE run, Risk Management total, 5-month and 12-month

Run Time 13:37:12			Risk Management 1-Month Windows											
Min		1,731	1	2	3	4	5	6	7	8	9	10	11	12
Max		14,716	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	6,960	7,679	6,647	8,846	8,011	14,716	12,784	12,376	10,697	13,019	11,416	6,936
	2	0.86400	6,488	7,463	6,050	8,468	7,716	14,058	12,535	12,086	10,566	12,811	11,018	6,818
	3	0.88665	6,115	7,136	5,642	8,111	7,301	13,370	12,265	11,796	10,412	12,554	10,713	6,583
	4	0.90930	5,742	6,705	4,884	7,568	6,739	12,519	11,979	11,349	10,184	12,156	10,170	6,420
	5	0.93195	5,295	6,165	4,341	6,936	6,212	11,483	11,672	11,044	9,871	11,792	9,991	6,110
	6	0.95460	4,616	5,679	3,962	5,916	5,304	10,417	11,302	10,467	9,598	11,441	9,594	5,869
	7	0.97725	3,920	4,600	3,436	4,682	4,327	9,239	10,660	9,709	9,118	10,789	9,034	5,501
Std Dev	1	0.84134	4,937	5,254	4,597	5,820	6,384	9,760	7,668	7,754	7,305	7,341	6,630	5,237
	2	0.86400	4,758	5,209	4,469	5,745	6,443	9,755	7,648	7,717	7,260	7,323	6,602	5,184
	3	0.88665	4,588	5,211	4,340	5,719	6,222	9,590	7,609	7,707	7,224	7,259	6,690	5,143
	4	0.90930	4,437	4,896	3,855	5,553	6,126	9,339	7,541	7,687	7,248	7,226	6,653	5,127
	5	0.93195	4,349	4,653	3,596	5,400	5,940	9,149	7,393	7,667	7,257	7,265	6,724	5,154
	6	0.95460	4,134	4,513	3,402	5,127	5,806	8,850	7,321	7,671	7,287	7,328	6,610	5,153
	7	0.97725	3,839	4,206	3,166	4,535	5,309	8,782	7,220	7,412	7,221	7,284	6,501	5,197

			Risk Management 3-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	8,643	7,044	7,514	6,726	8,973	10,934	12,802	11,657	11,934	11,608	9,906	7,021
	2	0.86400	7,878	6,715	7,023	6,158	8,638	10,511	12,338	11,452	11,766	11,370	9,718	6,774
	3	0.88665	7,252	6,165	6,421	5,723	8,229	10,116	11,955	11,230	11,543	11,116	9,392	6,508
	4	0.90930	6,523	5,659	5,771	5,006	7,776	9,536	11,590	10,994	11,124	10,672	9,075	6,235
	5	0.93195	5,822	5,004	5,178	4,078	7,174	8,844	11,118	10,530	10,690	10,259	8,795	5,831
	6	0.95460	4,615	4,311	4,339	2,983	6,418	8,499	10,332	10,037	10,359	9,793	8,360	5,152
	7	0.97725	3,292	3,377	3,312	1,731	5,292	7,653	9,274	9,313	9,517	9,324	7,952	4,305
Std Dev	1	0.84134	6,648	4,646	5,077	5,415	6,778	7,639	8,221	7,431	7,369	6,993	6,316	4,636
	2	0.86400	6,450	4,536	4,751	5,266	6,696	7,550	8,209	7,389	7,327	6,981	6,311	4,579
	3	0.88665	6,126	4,316	4,613	4,889	6,608	7,546	8,161	7,355	7,285	6,941	6,285	4,588
	4	0.90930	5,859	4,203	4,444	4,531	6,513	7,437	8,044	7,301	7,208	6,943	6,285	4,450
	5	0.93195	5,391	4,009	4,105	3,835	6,324	7,305	7,919	7,305	7,291	6,863	6,294	4,370
	6	0.95460	4,344	3,823	3,773	3,291	6,168	7,153	7,830	7,297	7,360	6,937	6,209	4,344
	7	0.97725	3,643	3,459	3,448	2,410	6,018	7,009	7,692	7,142	7,237	6,942	6,101	4,189



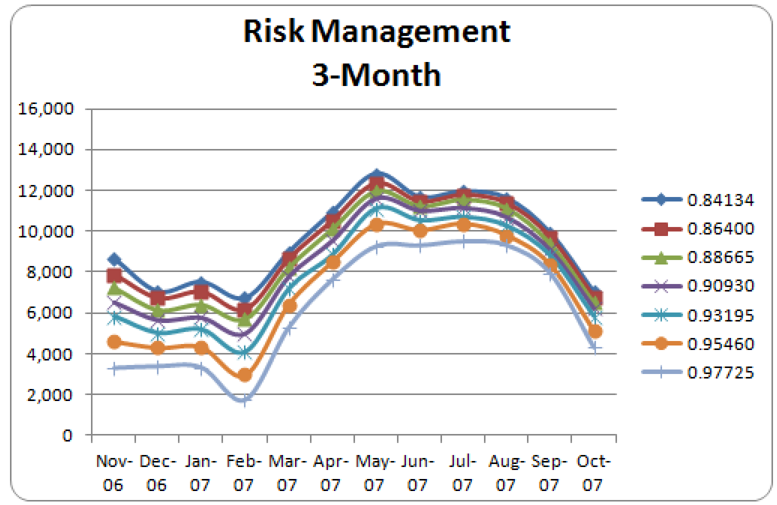
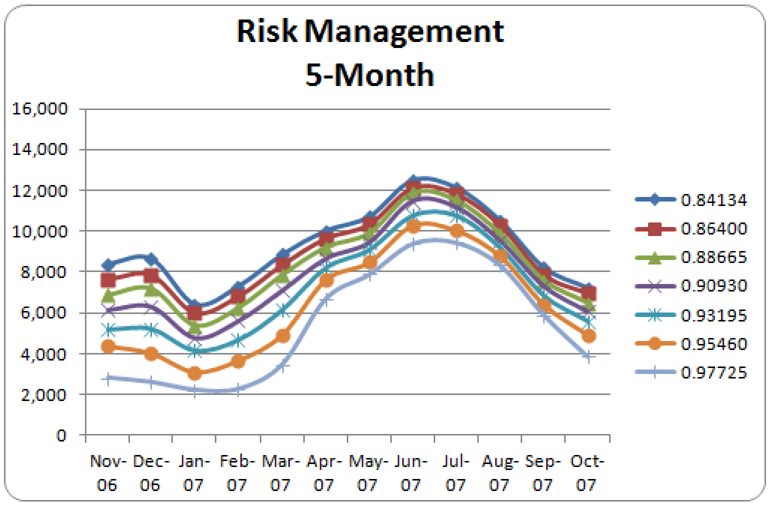


Figure 15: TEST run, Risk Management total, 1-month and 3-month

Run Time 13:37:12		Risk Management											
		5-Month Windows											
Min	1,731	1	2	3	4	5	6	7	8	9	10	11	12
Max	14,716	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average 1	0.84134	8,401	8,641	6,410	7,301	8,851	10,003	10,707	12,484	12,109	10,504	8,209	7,226
2	0.86400	7,654	7,842	5,992	6,805	8,342	9,639	10,329	12,129	11,828	10,289	7,865	6,951
3	0.88665	6,879	7,178	5,388	6,254	7,854	9,201	9,899	11,859	11,488	9,870	7,596	6,494
4	0.90930	6,146	6,305	4,784	5,615	7,096	8,649	9,443	11,467	11,147	9,516	7,310	6,070
5	0.93195	5,191	5,213	4,154	4,686	6,147	8,211	9,105	10,778	10,770	9,188	6,855	5,609
6	0.95460	4,373	4,012	3,097	3,666	4,918	7,628	8,484	10,294	10,039	8,783	6,410	4,927
7	0.97725	2,819	2,606	2,223	2,268	3,470	6,703	7,895	9,412	9,452	8,336	5,886	3,855
Std Dev 1	0.84134	6,466	6,234	4,866	5,657	6,533	7,022	7,548	7,720	7,184	6,681	5,449	4,463
2	0.86400	6,271	5,909	4,629	5,438	6,370	7,045	7,463	7,712	7,167	6,692	5,429	4,447
3	0.88665	6,079	5,579	4,143	5,359	6,260	6,929	7,303	7,659	7,133	6,614	5,402	4,386
4	0.90930	5,821	5,236	3,905	5,150	6,035	6,800	7,236	7,633	7,078	6,646	5,331	4,327
5	0.93195	5,217	4,532	3,755	4,603	5,606	6,684	7,191	7,485	7,083	6,655	5,322	4,243
6	0.95460	4,359	4,175	3,309	3,718	5,334	6,638	7,131	7,483	7,029	6,578	5,234	4,165
7	0.97725	3,504	3,015	2,619	3,040	4,037	6,299	6,992	7,446	6,925	6,409	5,075	3,902

		Risk Management											
		12-Month											
		1	2	3	4	5	6	7	8	9	10	11	12
		Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average 1	0.84134	7,713	7,716	7,960	8,273	8,241	8,079	8,092	8,124	8,183	8,082	8,315	8,229
2	0.86400	7,325	7,292	7,568	7,787	7,661	7,525	7,569	7,605	7,702	7,643	7,785	7,724
3	0.88665	6,864	6,783	6,852	7,216	7,141	7,036	7,050	7,096	7,167	6,995	7,170	7,063
4	0.90930	6,155	6,085	6,311	6,537	6,423	6,304	6,349	6,447	6,295	6,251	6,317	6,320
5	0.93195	5,409	5,337	5,366	5,678	5,537	5,389	5,471	5,471	5,522	5,429	5,276	5,235
6	0.95460	4,532	4,411	4,372	4,509	4,597	4,443	4,441	4,470	4,597	4,342	4,157	3,954
7	0.97725	3,335	3,310	3,137	2,885	2,898	2,788	2,710	2,671	2,733	2,531	2,503	2,328
Std Dev 1	0.84134	6,080	6,227	5,998	6,243	6,020	5,852	5,874	5,731	5,627	5,504	5,590	5,577
2	0.86400	5,947	6,041	5,875	6,109	5,907	5,683	5,667	5,588	5,484	5,418	5,456	5,432
3	0.88665	5,795	5,850	5,734	5,974	5,735	5,521	5,528	5,373	5,188	5,164	5,276	5,323
4	0.90930	5,553	5,672	5,627	5,737	5,457	5,287	5,241	5,157	5,016	5,053	5,090	5,098
5	0.93195	5,152	5,403	5,335	5,568	5,289	5,126	5,099	5,104	4,936	4,963	4,741	4,839
6	0.95460	4,872	5,112	4,891	4,988	4,926	4,782	4,736	4,715	4,593	4,447	4,217	4,227
7	0.97725	4,052	4,043	3,838	3,971	3,832	3,523	3,673	3,621	3,515	3,511	3,442	3,384



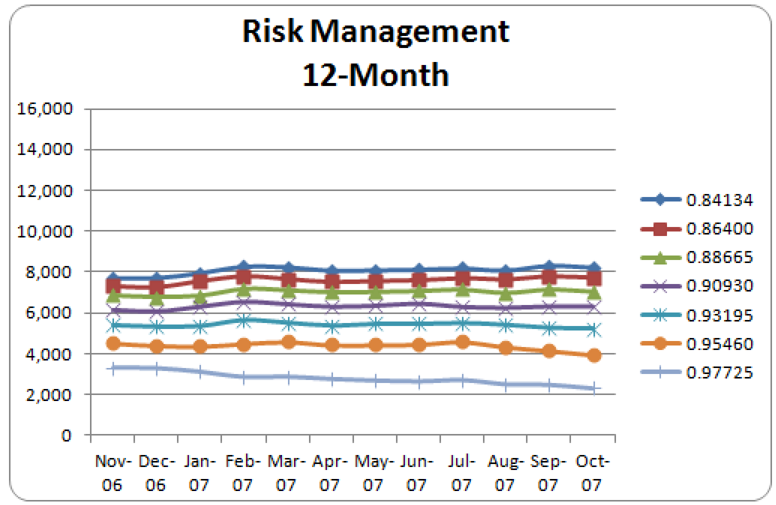
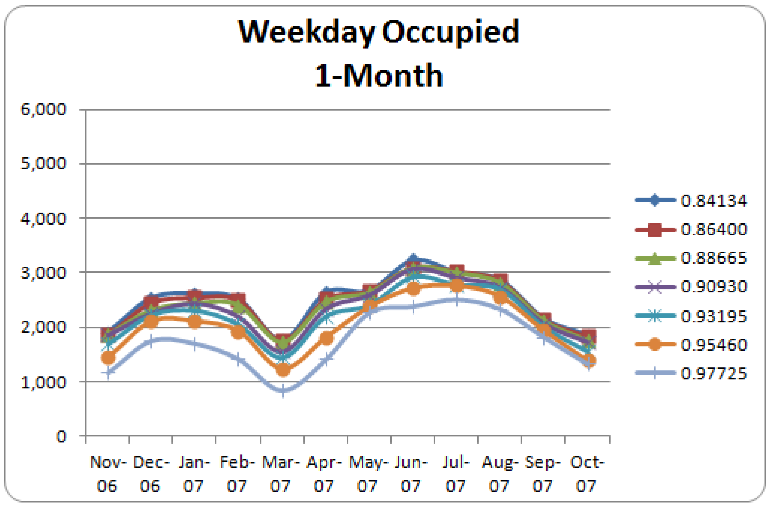


Figure 16: TEST run, Risk Management total, 5-month and 12-month

			Weekday Occupied											
			1-Month Windows											
Max		3,238	1	2	3	4	5	6	7	8	9	10	11	12
Min		690	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,910	2,540	2,625	2,520	1,757	2,643	2,673	3,238	3,012	2,871	2,156	1,856
	2	0.86400	1,866	2,454	2,548	2,502	1,763	2,520	2,676	3,095	3,028	2,854	2,150	1,839
	3	0.88665	1,866	2,309	2,436	2,404	1,723	2,465	2,637	3,085	2,998	2,810	2,115	1,757
	4	0.90930	1,862	2,256	2,439	2,196	1,556	2,340	2,595	3,071	2,910	2,733	2,076	1,735
	5	0.93195	1,704	2,237	2,318	2,061	1,452	2,212	2,420	2,936	2,792	2,710	2,021	1,567
	6	0.95460	1,456	2,116	2,119	1,936	1,246	1,819	2,381	2,723	2,767	2,570	1,946	1,394
	7	0.97725	1,177	1,761	1,711	1,425	849	1,418	2,279	2,393	2,522	2,345	1,828	1,339
Std Dev	1	0.84134	4,303	4,758	4,853	4,551	4,072	5,531	4,926	5,656	5,510	5,333	4,583	4,030
	2	0.86400	4,167	4,653	4,730	4,534	4,010	5,483	4,919	5,634	5,470	5,304	4,560	4,017
	3	0.88665	4,077	4,567	4,489	4,493	3,994	5,455	4,903	5,592	5,451	5,295	4,486	3,925
	4	0.90930	4,026	4,508	4,365	4,466	3,920	5,352	4,892	5,588	5,382	5,302	4,424	3,892
	5	0.93195	4,071	4,522	4,389	4,367	3,895	5,310	4,798	5,513	5,390	5,232	4,428	3,775
	6	0.95460	3,936	4,434	4,361	4,252	3,721	5,007	4,780	5,355	5,348	5,145	4,438	3,549
	7	0.97725	3,703	4,341	4,013	4,066	3,372	4,733	4,632	5,228	5,167	4,980	4,308	3,329

			Weekday Occupied											
			3-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,263	2,348	2,467	2,191	2,089	2,111	2,756	2,927	3,017	2,694	2,084	1,468
	2	0.86400	2,093	2,304	2,347	2,051	1,920	2,103	2,626	2,901	3,001	2,668	2,076	1,422
	3	0.88665	2,018	2,239	2,252	2,061	1,749	2,043	2,495	2,853	2,929	2,566	2,016	1,353
	4	0.90930	1,847	2,214	2,163	1,732	1,653	2,003	2,333	2,865	2,858	2,549	1,992	1,363
	5	0.93195	1,601	2,133	1,978	1,470	1,539	1,806	2,309	2,721	2,791	2,461	1,955	1,373
	6	0.95460	1,148	1,909	1,835	1,058	1,485	1,650	2,323	2,595	2,607	2,244	1,830	1,264
	7	0.97725	833	1,588	1,603	824	1,269	1,471	1,997	2,258	2,424	2,121	1,804	1,121
Std Dev	1	0.84134	4,856	4,495	4,595	4,423	4,619	4,792	5,277	5,396	5,430	5,148	4,557	3,822
	2	0.86400	4,674	4,388	4,540	4,345	4,555	4,765	5,258	5,403	5,361	5,117	4,524	3,754
	3	0.88665	4,640	4,359	4,533	4,338	4,391	4,675	5,190	5,355	5,330	5,091	4,482	3,734
	4	0.90930	4,593	4,345	4,346	4,104	4,255	4,652	5,079	5,297	5,337	5,033	4,470	3,641
	5	0.93195	4,359	4,294	4,236	3,881	4,123	4,536	5,007	5,186	5,322	4,958	4,464	3,650
	6	0.95460	3,806	4,183	4,163	3,391	4,097	4,358	4,953	5,080	5,205	4,813	4,297	3,458
	7	0.97725	3,285	4,039	3,898	3,040	3,821	4,178	4,772	4,971	5,068	4,718	4,253	3,389



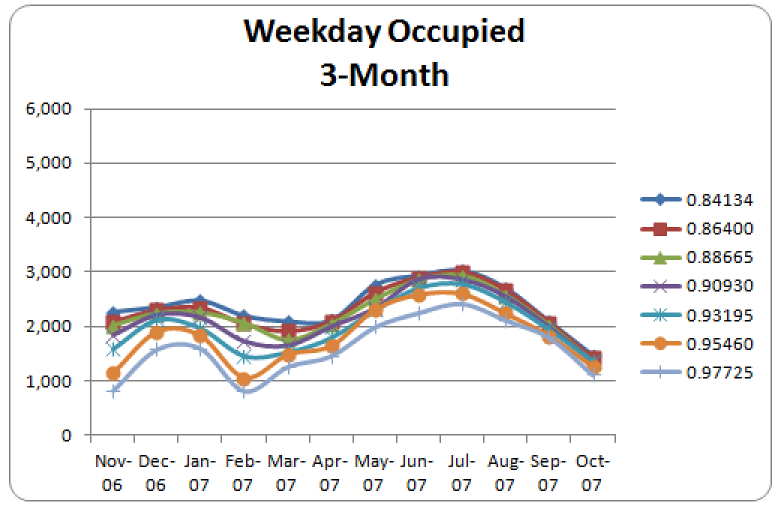
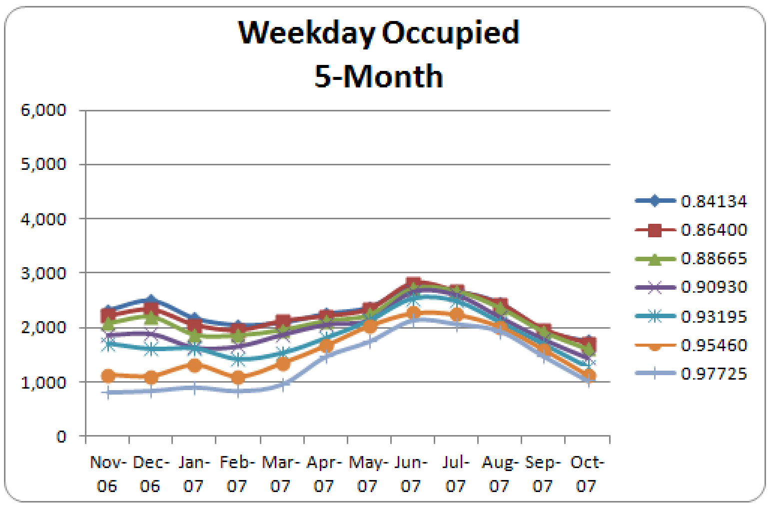


Figure 17: BASE run, Weekday Occupied, 1-month and 3-month

			Weekday Occupied											
			5-Month Windows											
Max		3,238	1	2	3	4	5	6	7	8	9	10	11	12
Min		690	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,322	2,495	2,171	2,050	2,095	2,255	2,373	2,764	2,667	2,445	1,949	1,750
	2	0.86400	2,218	2,327	2,047	1,952	2,119	2,211	2,341	2,811	2,661	2,432	1,966	1,714
	3	0.88665	2,091	2,204	1,880	1,871	1,967	2,132	2,244	2,721	2,673	2,365	1,932	1,620
	4	0.90930	1,873	1,892	1,642	1,665	1,877	2,063	2,144	2,665	2,604	2,195	1,788	1,452
	5	0.93195	1,718	1,619	1,628	1,430	1,544	1,817	2,139	2,544	2,491	2,104	1,704	1,285
	6	0.95460	1,136	1,114	1,319	1,102	1,350	1,674	2,038	2,271	2,244	2,017	1,596	1,128
	7	0.97725	820	847	912	844	966	1,474	1,752	2,140	2,072	1,939	1,477	1,032
Std Dev	1	0.84134	4,899	4,800	4,434	4,424	4,520	4,793	5,043	5,428	5,133	4,919	4,529	4,282
	2	0.86400	4,736	4,667	4,406	4,315	4,496	4,767	5,043	5,411	5,123	4,885	4,513	4,171
	3	0.88665	4,664	4,551	4,273	4,182	4,411	4,704	4,974	5,286	5,031	4,795	4,421	4,118
	4	0.90930	4,644	4,464	4,094	4,160	4,337	4,578	4,840	5,213	4,984	4,674	4,251	3,983
	5	0.93195	4,423	4,180	3,847	3,853	4,234	4,487	4,791	5,165	4,904	4,632	4,188	3,830
	6	0.95460	3,940	3,726	3,628	3,632	4,059	4,335	4,689	5,019	4,791	4,603	4,088	3,703
	7	0.97725	3,321	3,145	3,355	3,002	3,435	4,283	4,457	4,908	4,729	4,517	4,030	3,583

			Weekday Occupied											
			12-Month											
Max		3,238	1	2	3	4	5	6	7	8	9	10	11	12
Min		690	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,685	1,658	1,715	1,682	1,688	1,664	1,613	1,653	1,575	1,607	1,662	1,631
	2	0.86400	1,718	1,692	1,749	1,717	1,721	1,697	1,648	1,684	1,608	1,638	1,693	1,664
	3	0.88665	1,690	1,664	1,719	1,688	1,694	1,670	1,626	1,656	1,589	1,614	1,662	1,639
	4	0.90930	1,569	1,545	1,595	1,567	1,571	1,550	1,512	1,538	1,479	1,500	1,543	1,523
	5	0.93195	1,451	1,430	1,474	1,450	1,453	1,434	1,401	1,423	1,370	1,389	1,428	1,410
	6	0.95460	1,283	1,261	1,303	1,282	1,286	1,267	1,241	1,257	1,214	1,228	1,258	1,246
	7	0.97725	739	727	747	740	738	727	710	718	690	698	722	714
Std Dev	1	0.84134	4,628	4,609	4,716	4,631	4,633	4,604	4,509	4,552	4,424	4,471	4,601	4,537
	2	0.86400	4,577	4,558	4,665	4,581	4,582	4,553	4,459	4,501	4,373	4,420	4,550	4,486
	3	0.88665	4,495	4,473	4,582	4,498	4,501	4,470	4,381	4,421	4,297	4,342	4,464	4,405
	4	0.90930	4,416	4,395	4,500	4,419	4,420	4,390	4,301	4,340	4,218	4,262	4,385	4,326
	5	0.93195	4,239	4,218	4,320	4,242	4,244	4,215	4,130	4,167	4,049	4,092	4,209	4,152
	6	0.95460	4,045	4,024	4,123	4,048	4,049	4,020	3,940	3,972	3,862	3,899	4,011	3,960
	7	0.97725	3,471	3,464	3,530	3,474	3,466	3,448	3,369	3,400	3,296	3,332	3,450	3,395



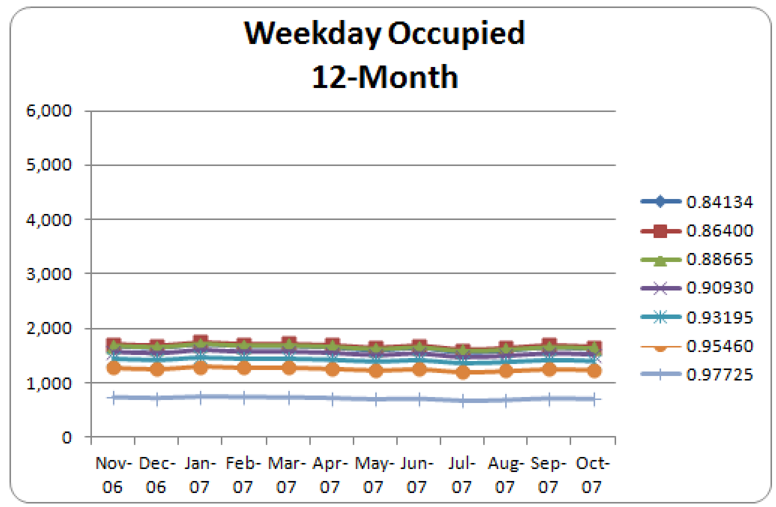
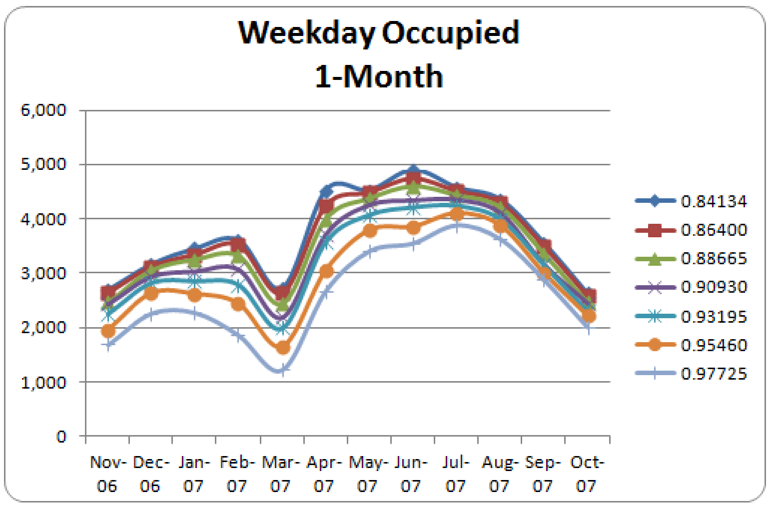


Figure 18: BASE run, Weekday Occupied, 5-month and 12-month

			Weekday Occupied 1-Month Windows											
Min		966	1	2	3	4	5	6	7	8	9	10	11	12
Max		4,893	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,703	3,168	3,458	3,611	2,732	4,525	4,523	4,893	4,572	4,353	3,563	2,636
	2	0.86400	2,640	3,116	3,341	3,517	2,634	4,243	4,484	4,739	4,508	4,292	3,492	2,576
	3	0.88665	2,472	3,060	3,259	3,326	2,449	4,001	4,377	4,598	4,436	4,214	3,368	2,476
	4	0.90930	2,439	2,945	3,039	3,074	2,195	3,718	4,255	4,340	4,352	4,112	3,172	2,459
	5	0.93195	2,264	2,824	2,857	2,792	2,000	3,571	4,063	4,200	4,231	3,984	3,153	2,310
	6	0.95460	1,967	2,635	2,622	2,442	1,652	3,057	3,793	3,857	4,104	3,878	3,006	2,223
	7	0.97725	1,693	2,267	2,282	1,868	1,236	2,680	3,411	3,553	3,884	3,633	2,892	2,022
Std Dev	1	0.84134	3,034	3,225	3,639	3,876	3,539	5,349	4,520	4,832	4,767	4,705	4,172	3,340
	2	0.86400	2,925	3,180	3,529	3,855	3,520	5,252	4,528	4,814	4,754	4,673	4,136	3,355
	3	0.88665	2,819	3,183	3,412	3,798	3,523	5,178	4,532	4,768	4,748	4,665	4,140	3,360
	4	0.90930	2,804	3,148	3,176	3,618	3,471	5,084	4,485	4,752	4,705	4,650	4,151	3,367
	5	0.93195	2,684	3,097	2,998	3,530	3,277	5,085	4,343	4,767	4,672	4,642	4,141	3,307
	6	0.95460	2,501	2,984	2,816	3,338	3,032	4,879	4,287	4,717	4,664	4,664	4,119	3,254
	7	0.97725	2,285	2,742	2,617	3,070	2,714	4,615	4,325	4,573	4,667	4,626	4,062	3,225

			Weekday Occupied 3-Month Windows											
Min		966	1	2	3	4	5	6	7	8	9	10	11	12
Max		4,893	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	3,201	3,191	3,339	3,067	3,227	3,796	4,649	4,669	4,518	4,106	3,288	2,553
	2	0.86400	2,953	3,127	3,215	2,886	3,063	3,645	4,507	4,558	4,437	4,002	3,216	2,482
	3	0.88665	2,780	2,958	3,063	2,789	2,880	3,459	4,267	4,384	4,325	3,896	3,126	2,360
	4	0.90930	2,494	2,866	2,987	2,416	2,782	3,208	4,100	4,251	4,197	3,804	3,036	2,213
	5	0.93195	2,322	2,778	2,839	2,023	2,436	2,916	3,899	4,011	4,039	3,726	2,985	1,992
	6	0.95460	1,872	2,483	2,590	1,632	2,130	2,783	3,532	3,804	3,943	3,576	2,887	1,679
	7	0.97725	1,523	2,068	2,176	1,057	1,673	2,317	3,159	3,545	3,674	3,389	2,691	1,415
Std Dev	1	0.84134	4,027	3,257	3,423	3,546	4,070	4,370	4,808	4,631	4,747	4,463	3,926	2,867
	2	0.86400	3,972	3,216	3,412	3,401	3,987	4,307	4,866	4,610	4,739	4,467	3,905	2,833
	3	0.88665	3,791	3,088	3,285	3,352	3,886	4,334	4,755	4,631	4,707	4,454	3,890	2,779
	4	0.90930	3,611	3,022	3,113	3,180	3,786	4,275	4,738	4,587	4,667	4,456	3,909	2,739
	5	0.93195	3,286	2,943	2,998	2,957	3,781	4,175	4,646	4,609	4,670	4,442	3,912	2,790
	6	0.95460	2,961	2,776	2,860	2,468	3,547	4,077	4,633	4,558	4,687	4,457	3,885	2,735
	7	0.97725	2,494	2,565	2,698	1,977	3,217	3,980	4,493	4,404	4,649	4,391	3,855	2,613



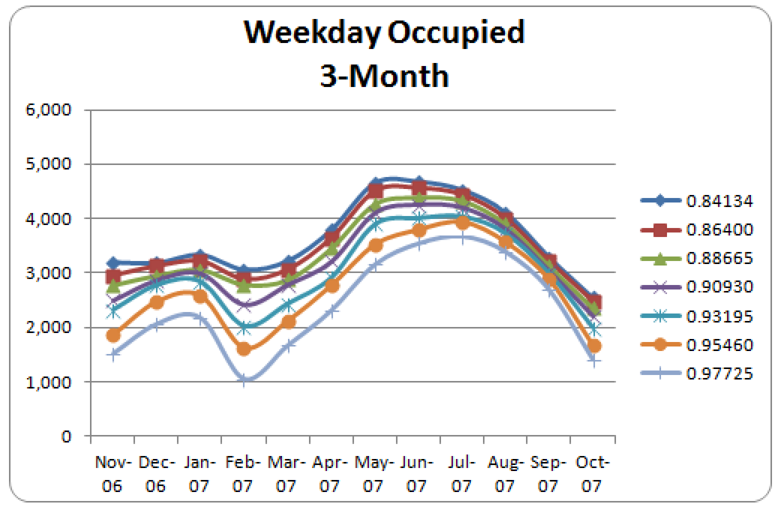
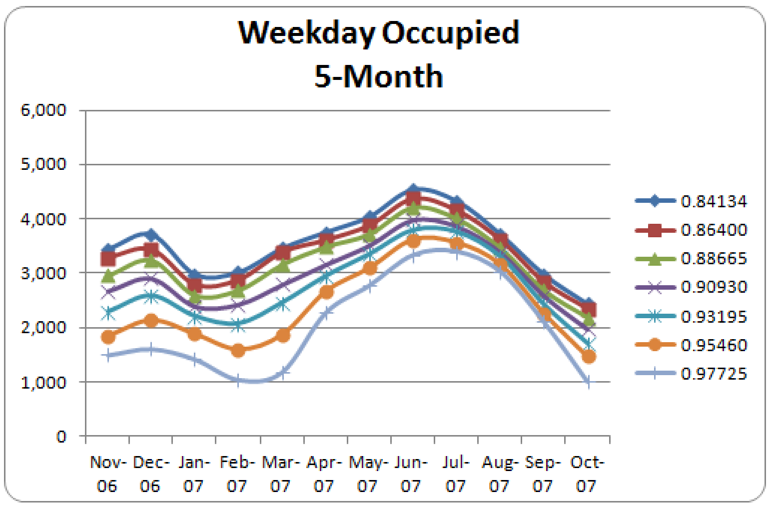


Figure 19: TEST run, Weekday Occupied, 1-month and 3-month

			Weekday Occupied												Weekday Occupied											
			5-Month Windows												12-Month											
Min		966	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Max		4,893	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	3,450	3,715	2,984	3,026	3,460	3,755	4,040	4,533	4,327	3,719	2,984	2,443	2,600	2,563	2,776	3,020	3,135	3,057	3,013	3,006	2,975	2,811	2,884	2,830
	2	0.86400	3,285	3,433	2,786	2,873	3,380	3,605	3,875	4,366	4,153	3,613	2,834	2,339	2,472	2,414	2,632	2,776	2,829	2,787	2,776	2,787	2,801	2,610	2,679	2,592
	3	0.88665	2,959	3,239	2,591	2,686	3,158	3,488	3,724	4,216	4,011	3,469	2,688	2,183	2,280	2,184	2,381	2,577	2,618	2,538	2,560	2,556	2,548	2,442	2,439	2,335
	4	0.90930	2,681	2,910	2,398	2,433	2,794	3,164	3,517	3,984	3,869	3,390	2,594	1,978	2,125	2,001	2,184	2,324	2,385	2,248	2,284	2,265	2,167	2,118	2,124	2,066
	5	0.93195	2,291	2,584	2,208	2,074	2,458	2,947	3,347	3,788	3,757	3,303	2,423	1,707	1,866	1,769	1,756	1,980	2,047	2,031	2,046	2,016	2,019	1,801	1,756	1,729
	6	0.95460	1,857	2,144	1,890	1,609	1,879	2,671	3,097	3,616	3,560	3,162	2,266	1,476	1,596	1,454	1,445	1,593	1,676	1,585	1,603	1,623	1,611	1,453	1,402	1,298
	7	0.97725	1,500	1,610	1,422	1,049	1,191	2,277	2,773	3,331	3,396	3,030	2,110	1,020	1,208	1,116	1,098	1,141	1,142	1,038	1,050	1,042	1,088	1,010	1,061	966
Std Dev	1	0.84134	4,065	4,007	3,239	3,465	3,945	4,234	4,511	4,741	4,486	4,161	3,408	2,743	3,622	3,674	3,648	3,822	3,796	3,727	3,683	3,578	3,492	3,389	3,316	3,167
	2	0.86400	4,006	3,877	3,133	3,391	3,943	4,216	4,475	4,747	4,501	4,155	3,425	2,733	3,565	3,598	3,569	3,746	3,717	3,649	3,565	3,479	3,378	3,249	3,222	3,127
	3	0.88665	3,847	3,692	2,997	3,271	3,913	4,217	4,437	4,722	4,470	4,149	3,417	2,698	3,523	3,516	3,472	3,524	3,539	3,493	3,455	3,384	3,318	3,205	3,165	2,985
	4	0.90930	3,596	3,521	2,859	3,142	3,730	4,167	4,375	4,715	4,451	4,119	3,368	2,645	3,418	3,404	3,311	3,434	3,428	3,332	3,293	3,228	3,044	3,027	2,963	2,892
	5	0.93195	3,196	3,293	2,710	2,924	3,478	3,995	4,370	4,643	4,492	4,105	3,322	2,515	3,299	3,313	3,106	3,291	3,266	3,196	3,144	3,088	3,007	2,849	2,734	2,682
	6	0.95460	2,901	2,865	2,452	2,533	3,133	3,900	4,301	4,548	4,426	4,117	3,285	2,456	3,032	3,049	2,903	3,034	3,026	2,962	2,907	2,863	2,789	2,725	2,621	2,502
	7	0.97725	2,569	2,246	2,058	1,984	2,738	3,817	4,163	4,556	4,358	4,053	3,247	2,167	2,564	2,661	2,552	2,630	2,593	2,476	2,531	2,581	2,514	2,379	2,255	2,116



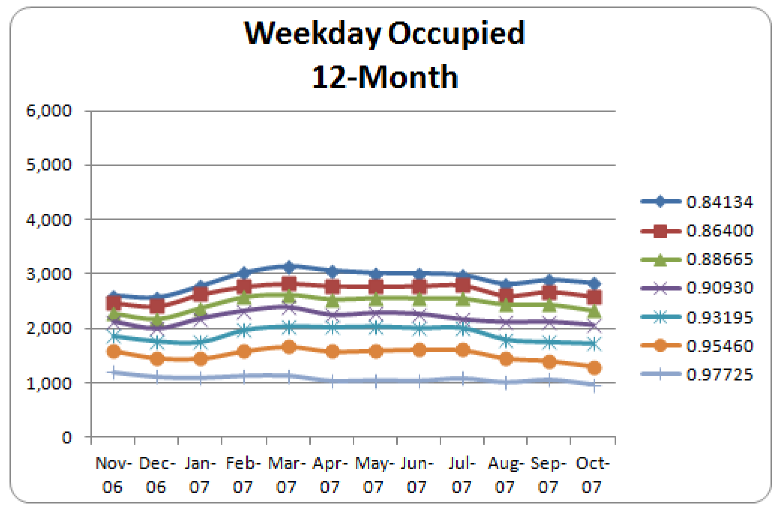
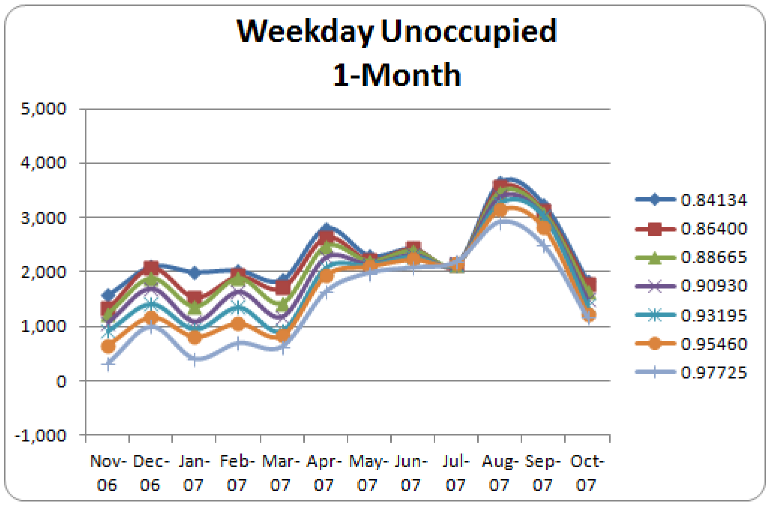


Figure 20: TEST run, Weekday Occupied, 5-month and 12-month

			Weekday Unoccupied – 1-Month Windows											
Max		3,643	1	2	3	4	5	6	7	8	9	10	11	12
Min		-283	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,576	2,101	2,000	2,032	1,860	2,803	2,301	2,430	2,142	3,643	3,240	1,840
	2	0.86400	1,339	2,085	1,545	1,931	1,716	2,646	2,227	2,437	2,138	3,564	3,130	1,783
	3	0.88665	1,232	1,877	1,357	1,877	1,422	2,468	2,191	2,390	2,142	3,469	3,083	1,642
	4	0.90930	1,072	1,698	1,085	1,639	1,176	2,271	2,150	2,320	2,158	3,393	3,034	1,502
	5	0.93195	927	1,417	957	1,356	930	2,076	2,123	2,284	2,150	3,261	3,011	1,386
	6	0.95460	664	1,172	813	1,061	834	1,942	2,107	2,237	2,177	3,155	2,829	1,233
	7	0.97725	330	996	406	696	631	1,639	1,985	2,081	2,191	2,919	2,483	1,165
Std Dev	1	0.84134	3,467	3,951	4,064	4,154	4,179	5,901	4,643	4,851	4,715	5,310	5,009	4,084
	2	0.86400	3,245	3,968	3,730	4,034	4,110	5,865	4,603	4,834	4,686	5,253	5,010	4,009
	3	0.88665	3,080	3,917	3,488	4,033	3,987	5,812	4,622	4,833	4,692	5,286	4,922	3,950
	4	0.90930	2,805	3,729	3,168	3,834	3,890	5,702	4,607	4,792	4,689	5,176	4,892	3,750
	5	0.93195	2,633	3,506	2,957	3,725	3,643	5,592	4,595	4,786	4,676	5,190	4,940	3,661
	6	0.95460	2,396	3,321	2,770	3,530	3,578	5,501	4,587	4,731	4,653	5,140	4,796	3,357
	7	0.97725	1,971	3,007	2,397	3,114	3,192	5,148	4,578	4,667	4,666	5,120	4,680	3,287

			Weekday Unoccupied – 3-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,269	1,699	2,068	1,529	2,114	2,240	2,327	2,268	2,521	3,014	2,764	1,976
	2	0.86400	2,116	1,546	1,896	1,307	1,889	2,181	2,292	2,257	2,558	2,921	2,712	1,921
	3	0.88665	2,070	1,254	1,579	1,112	1,683	1,915	2,278	2,228	2,527	2,917	2,672	1,707
	4	0.90930	1,846	1,109	1,266	970	1,379	1,749	2,270	2,193	2,543	2,860	2,455	1,601
	5	0.93195	1,531	1,060	1,003	598	1,141	1,612	2,159	2,079	2,476	2,747	2,319	1,345
	6	0.95460	1,277	932	832	5	836	1,563	2,086	1,987	2,380	2,707	2,159	1,217
	7	0.97725	477	513	318	-283	848	1,285	1,836	1,995	2,346	2,476	1,933	995
Std Dev	1	0.84134	4,658	3,605	4,081	3,817	4,575	4,937	4,998	4,702	4,824	4,973	4,680	3,778
	2	0.86400	4,514	3,518	3,893	3,559	4,436	4,876	4,990	4,702	4,808	4,971	4,673	3,693
	3	0.88665	4,563	3,221	3,634	3,385	4,287	4,765	4,968	4,684	4,808	4,967	4,627	3,606
	4	0.90930	4,341	3,062	3,369	3,210	4,195	4,626	4,997	4,683	4,838	4,899	4,585	3,541
	5	0.93195	4,021	2,952	2,978	2,943	4,052	4,586	4,936	4,641	4,850	4,907	4,607	3,421
	6	0.95460	3,717	2,760	2,707	1,931	3,907	4,397	4,849	4,651	4,819	4,954	4,510	3,260
	7	0.97725	2,364	2,399	2,385	954	3,714	4,313	4,718	4,639	4,762	4,821	4,299	3,134



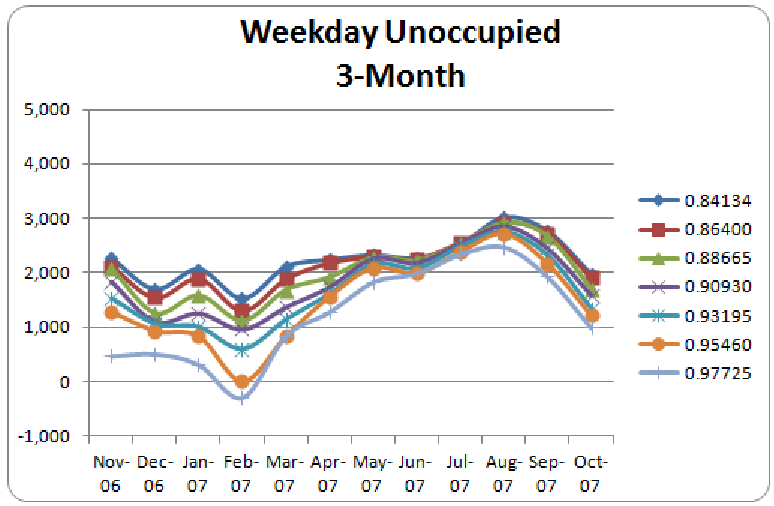


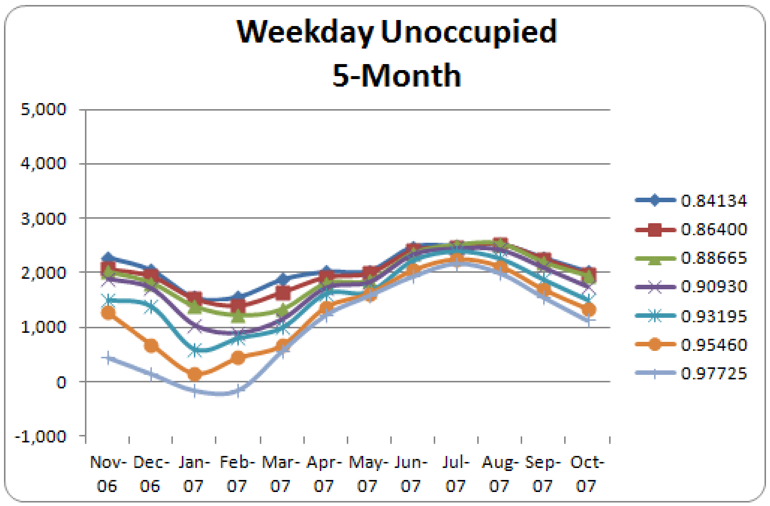
Figure 21: BASE run, Weekday Unoccupied, 1-month and 3-month

Weekday Unoccupied — 5-Month Windows

			1	2	3	4	5	6	7	8	9	10	11	12
Max		3,643												
Min		-283	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,275	2,045	1,549	1,567	1,878	2,020	2,035	2,467	2,487	2,516	2,270	2,022
	2	0.86400	2,069	1,933	1,519	1,396	1,641	1,919	1,993	2,413	2,471	2,530	2,258	1,971
	3	0.88665	2,015	1,811	1,381	1,218	1,328	1,801	1,853	2,355	2,507	2,530	2,189	1,944
	4	0.90930	1,897	1,721	1,036	894	1,146	1,730	1,837	2,347	2,461	2,435	2,103	1,752
	5	0.93195	1,513	1,394	609	816	1,006	1,625	1,661	2,231	2,398	2,265	1,882	1,514
	6	0.95460	1,272	692	167	457	682	1,373	1,624	2,047	2,247	2,114	1,693	1,346
	7	0.97725	450	153	-153	-142	571	1,232	1,590	1,941	2,177	1,998	1,553	1,134
Std Dev	1	0.84134	4,848	4,400	3,567	3,781	4,263	4,603	4,710	4,916	4,723	4,658	4,198	3,834
	2	0.86400	4,558	4,206	3,473	3,648	4,155	4,590	4,674	4,920	4,721	4,701	4,174	3,785
	3	0.88665	4,555	4,065	3,303	3,481	4,050	4,494	4,628	4,924	4,765	4,663	4,153	3,813
	4	0.90930	4,485	3,886	2,855	3,234	3,915	4,452	4,583	4,916	4,757	4,640	4,112	3,671
	5	0.93195	4,104	3,438	2,500	3,073	3,815	4,425	4,551	4,895	4,706	4,653	4,065	3,461
	6	0.95460	3,724	2,733	2,014	2,649	3,527	4,156	4,470	4,867	4,757	4,563	3,916	3,429
	7	0.97725	2,508	1,799	1,476	1,392	2,881	4,114	4,329	4,734	4,681	4,422	3,746	3,187

Weekday Unoccupied — 12-Month

			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,223	2,184	2,284	2,237	2,209	2,222	2,041	2,044	2,021	2,163	2,164	2,147
	2	0.86400	2,066	2,029	2,122	2,082	2,049	2,057	1,890	1,905	1,879	2,022	2,023	2,007
	3	0.88665	1,946	1,912	1,997	1,963	1,931	1,940	1,783	1,799	1,771	1,912	1,912	1,893
	4	0.90930	1,960	1,924	2,009	1,973	1,936	1,940	1,786	1,806	1,775	1,918	1,919	1,904
	5	0.93195	1,847	1,813	1,891	1,858	1,823	1,822	1,684	1,706	1,675	1,807	1,809	1,796
	6	0.95460	1,570	1,544	1,608	1,579	1,554	1,550	1,438	1,455	1,422	1,533	1,534	1,518
	7	0.97725	1,327	1,306	1,360	1,333	1,321	1,313	1,227	1,243	1,211	1,301	1,301	1,280
Std Dev	1	0.84134	4,382	4,286	4,473	4,353	4,290	4,274	4,045	4,077	4,073	4,218	4,237	4,288
	2	0.86400	4,301	4,207	4,392	4,271	4,212	4,196	3,974	4,003	4,001	4,136	4,155	4,206
	3	0.88665	4,161	4,070	4,248	4,135	4,077	4,059	3,845	3,878	3,870	4,010	4,028	4,072
	4	0.90930	4,171	4,077	4,258	4,143	4,074	4,054	3,832	3,870	3,861	4,009	4,028	4,081
	5	0.93195	4,031	3,938	4,116	4,000	3,931	3,910	3,695	3,730	3,725	3,862	3,883	3,941
	6	0.95460	3,713	3,627	3,791	3,685	3,619	3,598	3,398	3,432	3,424	3,558	3,576	3,628
	7	0.97725	3,315	3,238	3,383	3,294	3,227	3,210	3,025	3,062	3,048	3,186	3,201	3,243



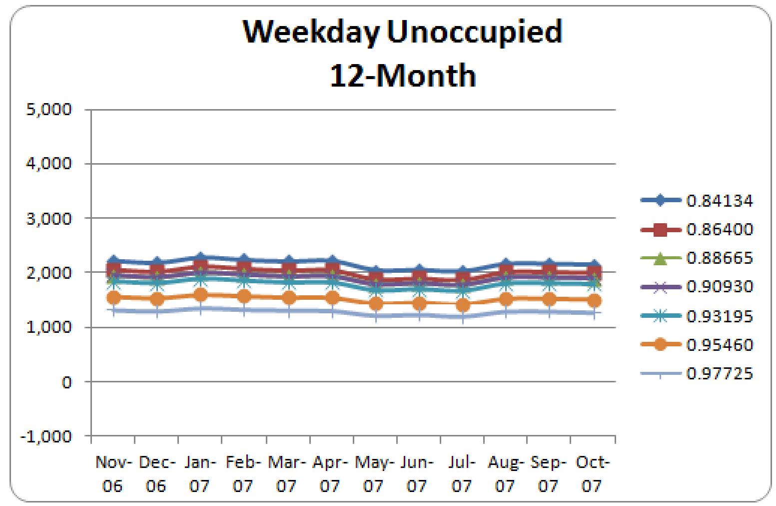
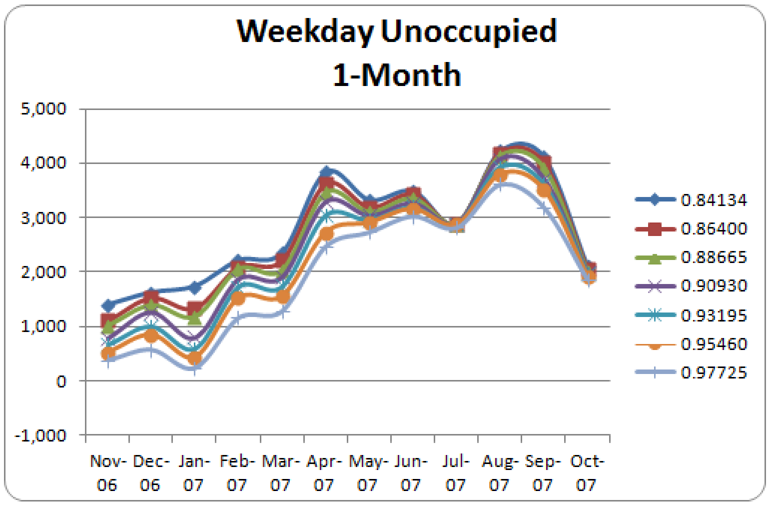


Figure 22: BASE run, Weekday Unoccupied, 5-month and 12-month

			Weekday Unoccupied — 1-Month Windows												Weekday Unoccupied — 3-Month Windows											
Min		-109	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Max		4,221	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,397	1,622	1,738	2,212	2,351	3,839	3,324	3,488	2,870	4,221	4,119	2,101	2,667	1,503	1,852	1,843	2,575	2,953	3,537	3,142	3,612	3,818	3,351	2,356
	2	0.86400	1,113	1,530	1,341	2,095	2,215	3,621	3,193	3,432	2,888	4,184	4,014	2,055	2,540	1,336	1,639	1,619	2,454	2,820	3,326	3,118	3,610	3,762	3,294	2,273
	3	0.88665	1,011	1,383	1,174	2,047	2,031	3,451	3,094	3,344	2,888	4,128	3,911	1,997	2,457	1,121	1,332	1,491	2,326	2,713	3,268	3,104	3,567	3,711	3,193	2,223
	4	0.90930	796	1,268	803	1,873	1,913	3,287	3,033	3,263	2,894	4,071	3,718	1,988	2,209	936	1,001	1,232	2,116	2,517	3,181	3,090	3,484	3,618	3,084	2,138
	5	0.93195	669	1,007	597	1,732	1,739	3,048	2,960	3,198	2,878	3,935	3,620	1,970	1,919	762	739	788	1,943	2,361	3,042	2,972	3,386	3,483	2,937	2,050
	6	0.95460	525	845	434	1,522	1,553	2,727	2,905	3,155	2,868	3,795	3,511	1,917	1,618	495	477	257	1,800	2,261	2,893	2,903	3,278	3,265	2,744	1,969
	7	0.97725	383	575	237	1,163	1,287	2,476	2,737	3,031	2,818	3,612	3,197	1,869	871	336	157	-109	1,540	2,106	2,616	2,870	3,166	3,147	2,658	1,640
Std Dev	1	0.84134	1,833	1,966	2,151	2,582	2,845	4,283	3,402	3,477	3,261	3,433	3,229	2,465	2,787	1,913	2,134	2,482	3,056	3,396	3,738	3,342	3,367	3,300	2,982	2,173
	2	0.86400	1,674	1,945	1,973	2,555	2,819	4,239	3,397	3,480	3,263	3,456	3,241	2,460	2,809	1,896	2,036	2,372	3,031	3,398	3,693	3,341	3,371	3,310	2,980	2,159
	3	0.88665	1,640	1,905	1,942	2,494	2,696	4,211	3,415	3,498	3,271	3,464	3,236	2,415	2,749	1,760	2,004	2,293	2,956	3,354	3,674	3,325	3,357	3,290	2,925	2,157
	4	0.90930	1,491	1,860	1,648	2,436	2,645	4,058	3,401	3,463	3,289	3,413	3,221	2,378	2,740	1,620	1,896	2,226	2,927	3,327	3,658	3,339	3,329	3,280	2,925	2,149
	5	0.93195	1,456	1,791	1,597	2,415	2,591	4,005	3,390	3,437	3,284	3,411	3,205	2,319	2,631	1,608	1,716	1,860	2,818	3,216	3,572	3,318	3,336	3,270	2,940	2,097
	6	0.95460	1,334	1,664	1,492	2,361	2,568	3,859	3,349	3,460	3,258	3,392	3,123	2,319	2,374	1,501	1,525	1,443	2,763	3,141	3,471	3,313	3,341	3,236	2,875	2,059
	7	0.97725	1,207	1,582	1,322	2,064	2,317	3,720	3,308	3,432	3,232	3,418	3,118	2,334	1,812	1,380	1,298	916	2,686	3,049	3,448	3,282	3,308	3,232	2,898	1,978



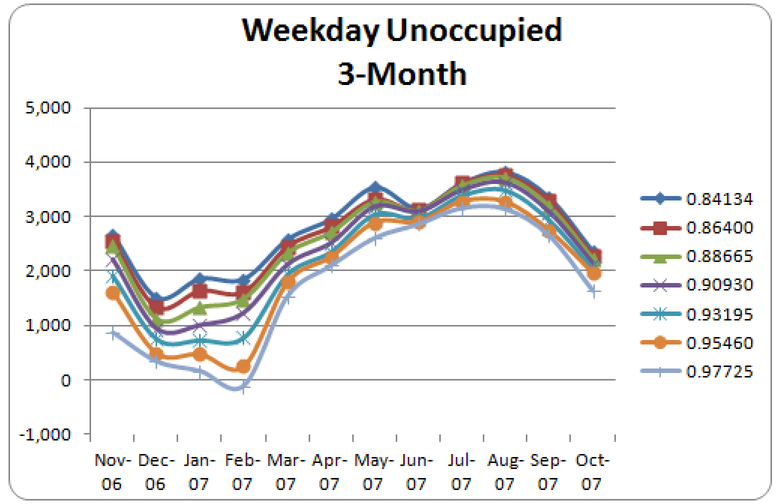
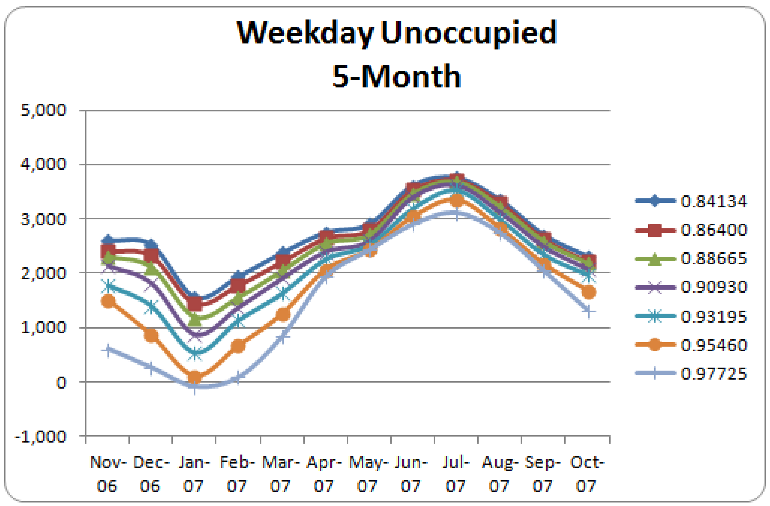


Figure 23: TEST run, Weekday Unoccupied, 1-month and 3-month

			Weekday Unoccupied											
			5-Month Windows											
Min		-109	1	2	3	4	5	6	7	8	9	10	11	12
Max		4,221	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,599	2,520	1,557	1,938	2,372	2,744	2,899	3,610	3,751	3,340	2,701	2,307
	2	0.86400	2,414	2,325	1,455	1,789	2,216	2,648	2,789	3,529	3,710	3,293	2,634	2,219
	3	0.88665	2,304	2,107	1,186	1,552	2,042	2,545	2,705	3,464	3,683	3,221	2,582	2,180
	4	0.90930	2,136	1,816	868	1,368	1,904	2,397	2,596	3,407	3,624	3,122	2,489	2,077
	5	0.93195	1,783	1,402	544	1,138	1,633	2,262	2,524	3,195	3,528	2,993	2,349	1,978
	6	0.95460	1,504	862	110	671	1,252	2,067	2,428	3,040	3,348	2,825	2,169	1,683
	7	0.97725	610	266	-85	87	837	1,937	2,428	2,899	3,112	2,738	2,045	1,323
Std Dev	1	0.84134	2,991	2,787	2,110	2,421	2,935	3,224	3,398	3,513	3,339	3,113	2,586	2,027
	2	0.86400	2,901	2,771	2,040	2,443	2,849	3,203	3,370	3,501	3,329	3,104	2,583	2,017
	3	0.88665	2,891	2,647	1,849	2,299	2,807	3,230	3,272	3,513	3,324	3,056	2,572	2,008
	4	0.90930	2,830	2,556	1,779	2,235	2,707	3,105	3,250	3,478	3,320	3,056	2,514	1,971
	5	0.93195	2,643	2,177	1,613	2,082	2,671	3,005	3,267	3,447	3,279	3,097	2,513	1,936
	6	0.95460	2,457	1,967	1,128	1,701	2,410	2,942	3,204	3,343	3,297	3,059	2,464	1,936
	7	0.97725	1,815	1,311	777	1,324	1,910	2,871	3,137	3,340	3,231	3,010	2,434	1,829

			Weekday Unoccupied											
			12-Month											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,587	2,609	2,552	2,605	2,591	2,591	2,585	2,628	2,679	2,732	2,812	2,826
	2	0.86400	2,510	2,546	2,466	2,517	2,520	2,513	2,497	2,532	2,610	2,646	2,715	2,693
	3	0.88665	2,383	2,451	2,366	2,388	2,372	2,415	2,353	2,380	2,475	2,491	2,584	2,514
	4	0.90930	2,225	2,261	2,196	2,217	2,170	2,210	2,194	2,251	2,295	2,278	2,319	2,339
	5	0.93195	2,098	2,037	2,001	1,990	1,909	1,942	1,926	1,983	2,046	2,103	2,054	2,045
	6	0.95460	1,786	1,742	1,716	1,732	1,696	1,682	1,666	1,674	1,725	1,714	1,586	1,434
	7	0.97725	1,482	1,371	1,261	1,060	1,032	1,125	1,050	1,036	1,049	929	825	719
Std Dev	1	0.84134	2,533	2,540	2,622	2,649	2,576	2,498	2,525	2,441	2,401	2,309	2,383	2,426
	2	0.86400	2,499	2,518	2,615	2,605	2,546	2,457	2,477	2,414	2,345	2,311	2,384	2,433
	3	0.88665	2,458	2,476	2,535	2,569	2,556	2,454	2,488	2,416	2,340	2,285	2,300	2,310
	4	0.90930	2,457	2,475	2,519	2,489	2,406	2,348	2,377	2,305	2,235	2,200	2,243	2,269
	5	0.93195	2,382	2,346	2,374	2,387	2,301	2,260	2,280	2,220	2,166	2,168	2,181	2,192
	6	0.95460	2,241	2,231	2,254	2,251	2,222	2,159	2,174	2,163	2,080	2,053	1,846	1,887
	7	0.97725	2,026	2,040	1,929	1,836	1,786	1,776	1,753	1,673	1,637	1,490	1,491	1,523



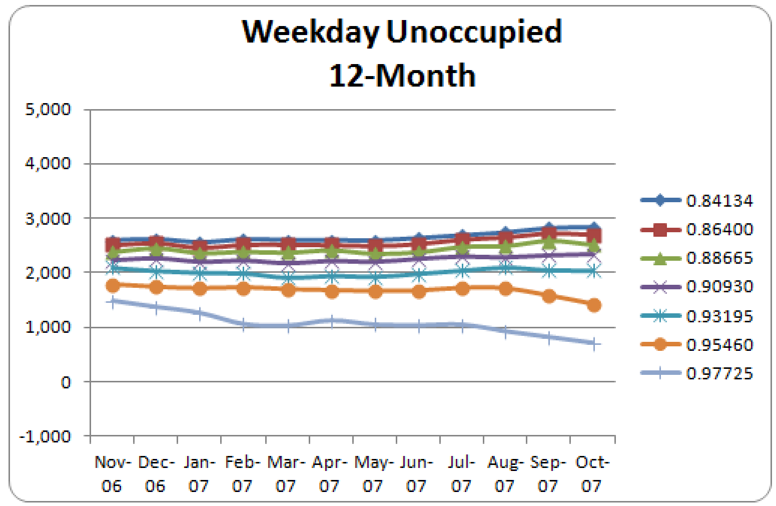
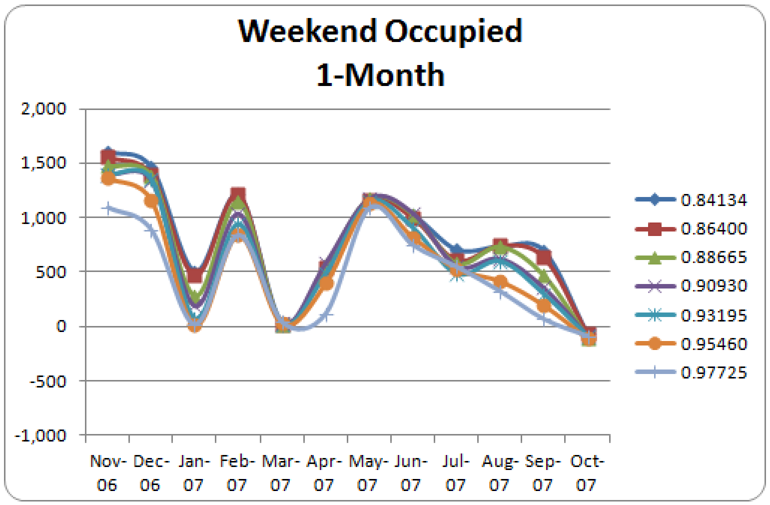


Figure 24: TEST run, Weekday Unoccupied, 5-month and 12-month

			Weekend Occupied – 1-Month Windows											
	Max	1,605	1	2	3	4	5	6	7	8	9	10	11	12
	Min	-107	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,605	1,464	506	1,210	38	542	1,147	1,021	704	732	696	-71
	2	0.86400	1,557	1,398	471	1,214	15	542	1,161	1,001	611	746	640	-65
	3	0.88665	1,473	1,379	278	1,145	25	569	1,173	1,028	577	731	467	-102
	4	0.90930	1,399	1,344	194	1,039	34	588	1,179	1,043	557	627	360	-78
	5	0.93195	1,395	1,349	70	940	22	485	1,144	902	480	596	295	-91
	6	0.95460	1,362	1,161	19	842	28	400	1,127	821	531	419	199	-107
	7	0.97725	1,095	889	19	841	40	120	1,092	748	551	323	67	-88
Std Dev	1	0.84134	5,047	5,038	2,671	4,383	742	3,337	3,838	3,828	3,294	3,110	3,315	1,023
	2	0.86400	5,040	5,010	2,596	4,388	641	3,257	3,842	3,752	3,275	3,125	3,099	1,023
	3	0.88665	5,026	4,942	1,964	4,404	641	3,247	3,840	3,745	3,239	3,101	2,521	763
	4	0.90930	4,969	4,821	1,504	4,351	643	3,259	3,830	3,747	3,180	2,911	2,133	739
	5	0.93195	4,923	4,826	818	4,190	503	3,111	3,825	3,690	3,117	2,849	1,736	706
	6	0.95460	4,810	4,769	279	4,026	503	2,907	3,827	3,615	3,060	2,221	1,286	675
	7	0.97725	4,664	4,459	279	3,939	498	1,758	3,806	3,441	3,016	1,762	788	668

			Weekend Occupied – 3-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	847	1,193	947	441	509	461	869	946	751	644	488	500
	2	0.86400	784	1,038	867	430	511	511	892	956	711	661	389	355
	3	0.88665	797	947	786	404	497	502	848	960	693	658	286	254
	4	0.90930	808	850	797	244	422	486	800	968	622	569	229	206
	5	0.93195	775	858	806	173	391	402	738	839	674	523	138	86
	6	0.95460	587	870	657	148	250	380	677	778	695	312	106	25
	7	0.97725	251	628	420	19	163	262	700	690	539	112	112	39
Std Dev	1	0.84134	3,769	4,336	3,985	2,367	2,376	2,509	3,453	3,722	3,345	3,117	2,727	2,854
	2	0.86400	3,745	4,296	3,896	2,288	2,376	2,506	3,464	3,716	3,311	3,134	2,258	2,292
	3	0.88665	3,613	4,195	3,823	2,232	2,353	2,513	3,472	3,646	3,255	3,104	1,961	1,968
	4	0.90930	3,628	4,105	3,679	1,724	2,205	2,483	3,428	3,642	3,189	3,009	1,629	1,570
	5	0.93195	3,576	3,956	3,686	1,343	2,146	2,408	3,325	3,600	3,126	2,826	1,257	1,092
	6	0.95460	3,198	3,961	3,473	1,198	1,666	2,276	3,241	3,512	3,126	1,966	949	686
	7	0.97725	1,814	3,481	2,571	286	1,174	1,795	3,167	3,407	2,861	1,107	938	671



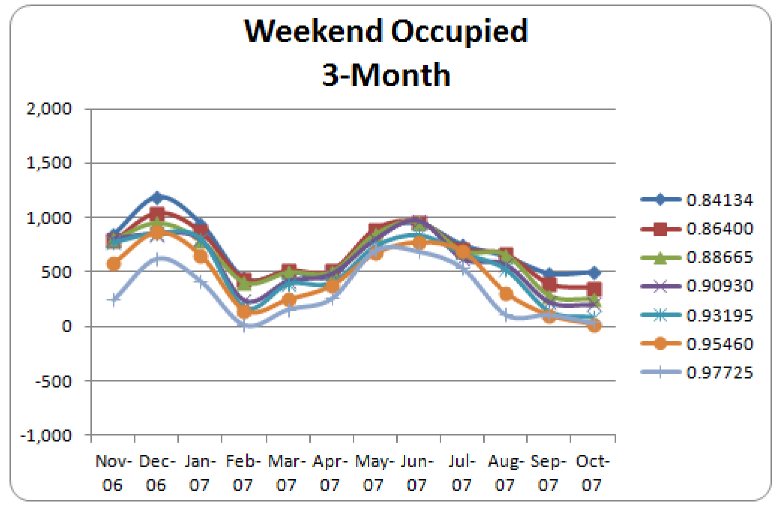
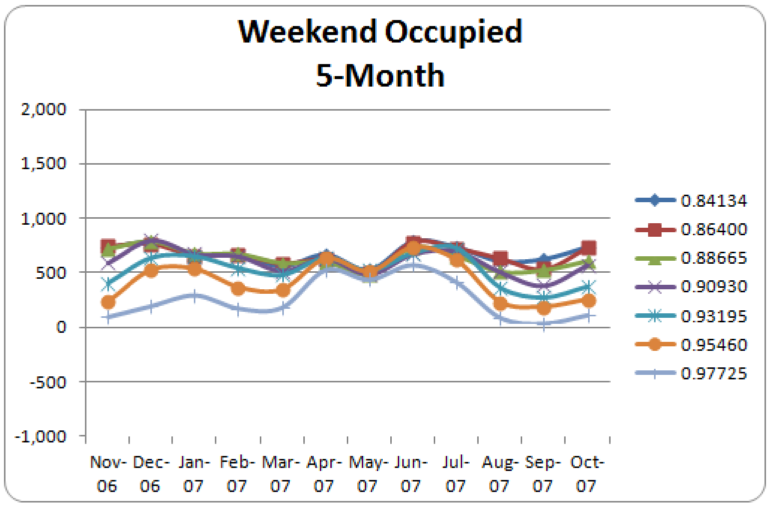


Figure 25: BASE run, Weekend Occupied, 1-month and 3-month

			Weekend Occupied											
			5-Month Windows											
Max		1,605	1	2	3	4	5	6	7	8	9	10	11	12
Min		-107	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	730	776	664	665	564	668	531	784	732	608	627	735
	2	0.86400	742	766	658	665	588	624	497	782	725	632	535	728
	3	0.88665	720	789	678	673	588	598	486	734	654	508	522	608
	4	0.90930	593	800	674	653	521	633	488	675	699	509	383	574
	5	0.93195	410	637	650	543	481	638	521	681	726	366	277	376
	6	0.95460	235	528	537	368	346	633	513	738	617	230	189	256
	7	0.97725	98	198	297	175	184	533	443	578	418	90	35	113
Std Dev	1	0.84134	3,354	3,738	3,224	3,069	2,671	3,080	2,681	3,446	3,297	3,011	3,140	3,434
	2	0.86400	3,370	3,629	3,140	3,069	2,686	3,036	2,660	3,448	3,272	2,998	3,044	3,399
	3	0.88665	3,327	3,607	3,120	3,084	2,686	3,000	2,592	3,389	3,207	2,893	2,935	3,169
	4	0.90930	3,078	3,615	3,122	3,048	2,655	2,955	2,582	3,328	3,156	2,787	2,343	3,090
	5	0.93195	2,401	3,402	3,088	2,824	2,475	2,952	2,596	3,243	3,172	2,221	1,993	2,350
	6	0.95460	1,726	2,927	2,862	2,214	1,964	2,919	2,574	3,233	3,029	1,663	1,579	1,689
	7	0.97725	947	1,580	2,094	1,410	1,295	2,714	2,401	3,073	2,290	1,063	687	1,158

			Weekend Occupied											
			12-Month											
Max		1,605	1	2	3	4	5	6	7	8	9	10	11	12
Min		-107	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	674	687	663	696	631	652	577	609	596	628	691	712
	2	0.86400	708	721	697	731	664	687	609	642	628	662	727	748
	3	0.88665	716	729	705	739	672	695	617	650	636	670	735	756
	4	0.90930	633	645	624	653	594	614	546	575	563	592	649	668
	5	0.93195	535	544	527	552	503	518	462	487	477	500	547	563
	6	0.95460	329	334	325	339	311	318	287	302	295	307	335	344
	7	0.97725	209	212	208	216	200	202	185	194	190	196	211	216
Std Dev	1	0.84134	3,230	3,289	3,174	3,320	3,034	3,153	2,810	2,944	2,894	3,048	3,321	3,414
	2	0.86400	3,196	3,255	3,140	3,285	3,001	3,120	2,778	2,911	2,862	3,016	3,288	3,380
	3	0.88665	3,214	3,273	3,157	3,303	3,018	3,137	2,793	2,927	2,878	3,033	3,306	3,398
	4	0.90930	3,172	3,229	3,116	3,260	2,979	3,096	2,758	2,890	2,841	2,993	3,262	3,352
	5	0.93195	2,862	2,913	2,812	2,941	2,689	2,793	2,490	2,609	2,565	2,701	2,942	3,024
	6	0.95460	2,163	2,201	2,127	2,223	2,035	2,110	1,886	1,975	1,941	2,041	2,222	2,283
	7	0.97725	1,478	1,502	1,458	1,518	1,399	1,440	1,302	1,360	1,337	1,396	1,512	1,551



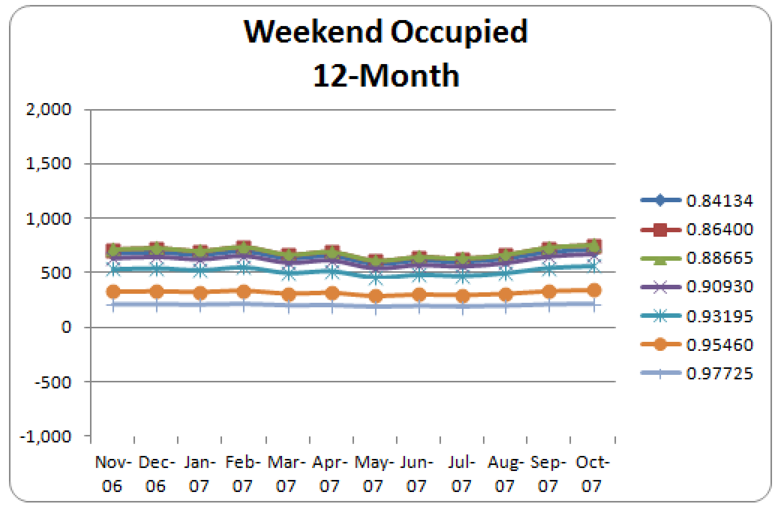
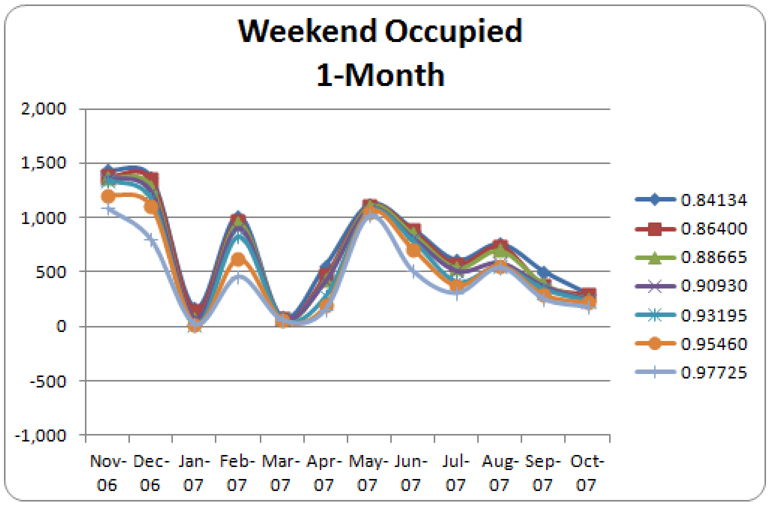


Figure 26: BASE run, Weekend Occupied, 5-month and 12-month

			Weekend Occupied											
			1-Month Windows											
Min		10	1	2	3	4	5	6	7	8	9	10	11	12
Max		1,433	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,433	1,365	167	1,015	80	561	1,118	892	607	756	502	298
	2	0.86400	1,378	1,348	140	966	70	472	1,110	883	573	741	383	293
	3	0.88665	1,371	1,286	67	950	75	435	1,106	851	543	700	394	253
	4	0.90930	1,369	1,234	74	904	76	419	1,075	810	509	583	367	244
	5	0.93195	1,340	1,168	17	824	71	282	1,053	785	411	553	345	240
	6	0.95460	1,208	1,100	25	621	64	193	1,048	709	375	559	290	220
	7	0.97725	1,091	800	24	463	62	161	1,021	512	310	546	256	183
Std Dev	1	0.84134	2,777	2,673	992	2,142	337	1,323	1,948	1,827	1,494	1,235	1,009	850
	2	0.86400	2,729	2,690	918	2,108	312	1,275	1,941	1,803	1,444	1,214	869	848
	3	0.88665	2,717	2,649	646	2,065	312	1,221	1,930	1,787	1,402	1,195	807	743
	4	0.90930	2,706	2,601	575	1,992	312	1,206	1,914	1,730	1,350	1,058	728	662
	5	0.93195	2,661	2,526	186	1,872	309	1,039	1,897	1,682	1,270	1,038	717	663
	6	0.95460	2,590	2,374	180	1,756	298	957	1,902	1,605	1,224	952	591	663
	7	0.97725	2,332	2,133	169	1,514	300	730	1,905	1,434	1,148	830	587	482

			Weekend Occupied											
			3-Month Windows											
Min		10	1	2	3	4	5	6	7	8	9	10	11	12
Max		1,433	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	845	975	871	133	265	349	841	822	705	551	478	368
	2	0.86400	757	930	809	81	257	339	815	815	667	529	455	338
	3	0.88665	607	876	736	81	231	333	802	784	660	513	442	326
	4	0.90930	550	763	569	58	131	300	757	761	556	396	433	321
	5	0.93195	475	585	489	53	104	241	716	731	508	310	425	309
	6	0.95460	220	507	272	46	90	176	593	649	483	309	367	293
	7	0.97725	137	167	58	10	64	155	480	511	243	239	322	193
Std Dev	1	0.84134	1,889	2,086	1,903	706	869	945	1,706	1,744	1,413	1,139	1,096	1,052
	2	0.86400	1,826	1,986	1,812	660	853	934	1,673	1,715	1,382	1,114	1,078	1,026
	3	0.88665	1,698	1,897	1,779	660	834	932	1,630	1,707	1,379	1,091	1,044	979
	4	0.90930	1,616	1,839	1,671	537	710	909	1,570	1,676	1,289	963	1,050	976
	5	0.93195	1,520	1,677	1,506	482	707	832	1,545	1,611	1,245	844	1,042	976
	6	0.95460	1,244	1,561	1,324	366	545	776	1,441	1,524	1,223	688	897	979
	7	0.97725	851	1,180	819	115	429	621	1,323	1,396	969	544	784	693



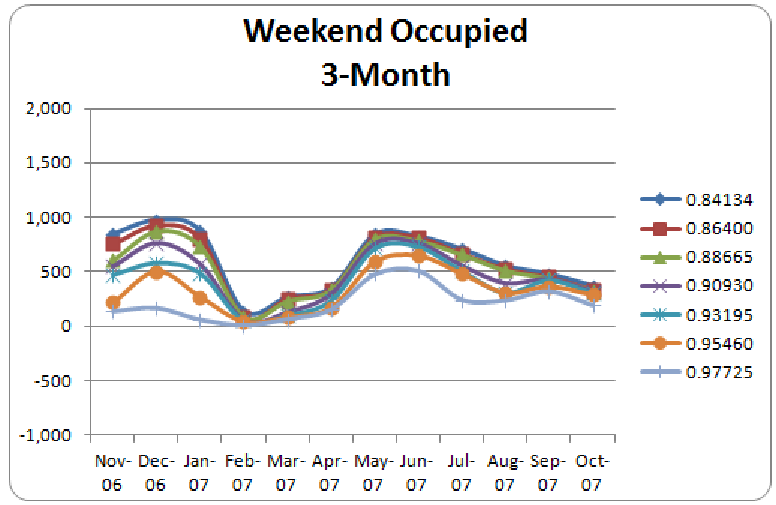
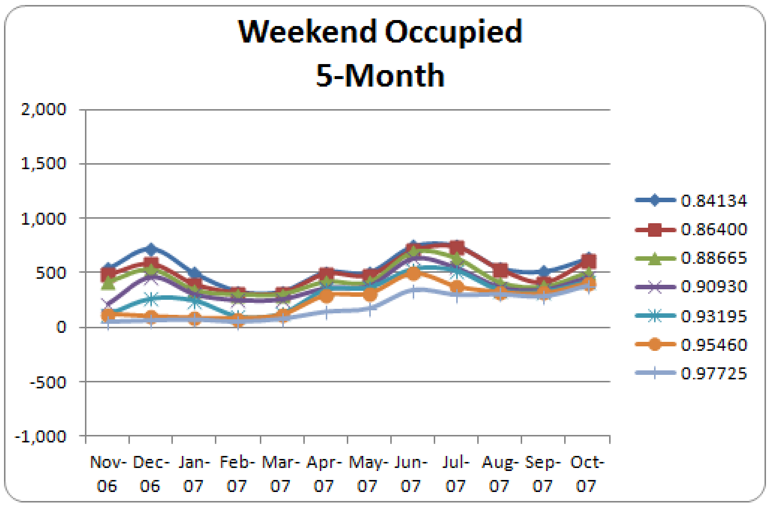


Figure 27: TEST run, Weekend Occupied, 1-month and 3-month

			Weekend Occupied											
			5-Month Windows											
Min		10	1	2	3	4	5	6	7	8	9	10	11	12
Max		1,433	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	540	726	495	327	327	506	506	748	751	544	514	631
	2	0.86400	480	581	390	311	310	490	475	707	739	530	409	608
	3	0.88665	416	531	337	297	300	416	413	693	635	408	370	501
	4	0.90930	214	464	302	247	256	360	382	636	548	368	345	460
	5	0.93195	120	263	244	99	127	345	365	533	517	339	328	419
	6	0.95460	121	98	82	79	116	297	313	501	379	325	315	409
	7	0.97725	48	61	69	50	76	141	172	340	297	301	282	379
Std Dev	1	0.84134	1,447	1,680	1,267	1,071	1,007	1,221	1,184	1,452	1,381	1,241	1,201	1,310
	2	0.86400	1,370	1,581	1,195	1,034	996	1,193	1,142	1,398	1,351	1,235	1,127	1,284
	3	0.88665	1,304	1,514	1,140	1,024	976	1,145	1,117	1,400	1,264	1,131	1,084	1,223
	4	0.90930	1,139	1,412	1,089	983	933	1,079	1,089	1,337	1,206	1,097	1,062	1,179
	5	0.93195	850	1,256	1,038	811	798	1,066	1,068	1,296	1,176	1,040	1,004	1,097
	6	0.95460	660	1,001	843	703	694	1,020	1,027	1,239	1,006	1,020	1,004	1,097
	7	0.97725	281	489	622	441	462	848	854	1,074	806	1,026	998	1,100

			Weekend Occupied											
			12-Month											
Min		10	1	2	3	4	5	6	7	8	9	10	11	12
Max		1,433	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	772	767	792	772	729	704	724	744	727	682	752	734
	2	0.86400	648	645	721	696	680	664	692	692	663	683	707	702
	3	0.88665	571	582	617	601	658	635	650	659	627	621	650	655
	4	0.90930	518	527	574	559	510	528	540	590	552	587	616	587
	5	0.93195	445	458	496	485	437	425	433	437	440	423	484	467
	6	0.95460	255	259	274	247	338	383	391	394	397	383	375	380
	7	0.97725	131	128	116	112	146	147	149	150	150	117	175	158
Std Dev	1	0.84134	1,732	1,709	1,646	1,616	1,554	1,583	1,624	1,666	1,666	1,595	1,632	1,642
	2	0.86400	1,654	1,655	1,626	1,577	1,497	1,534	1,573	1,596	1,581	1,591	1,572	1,629
	3	0.88665	1,570	1,578	1,554	1,529	1,472	1,505	1,542	1,574	1,553	1,519	1,510	1,591
	4	0.90930	1,504	1,506	1,483	1,462	1,387	1,408	1,441	1,489	1,454	1,478	1,506	1,517
	5	0.93195	1,419	1,424	1,380	1,371	1,311	1,299	1,326	1,341	1,350	1,358	1,365	1,431
	6	0.95460	1,258	1,245	1,180	1,154	1,186	1,244	1,270	1,284	1,293	1,284	1,254	1,357
	7	0.97725	916	874	820	795	966	981	998	1,007	1,013	980	994	1,153



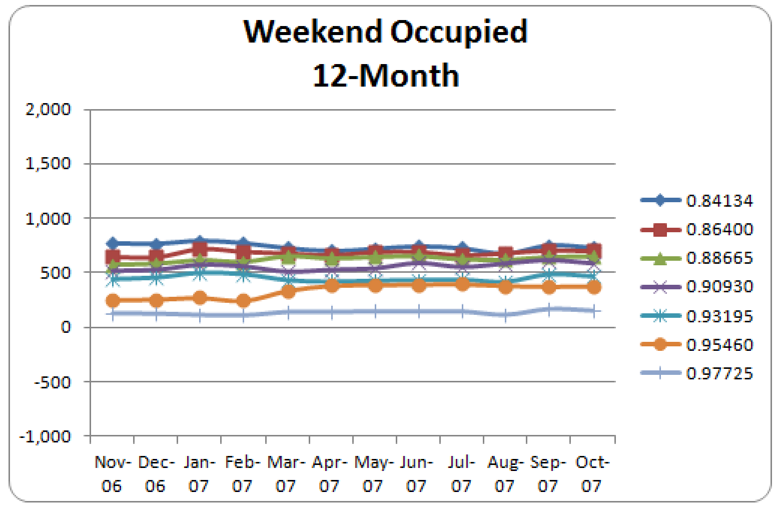
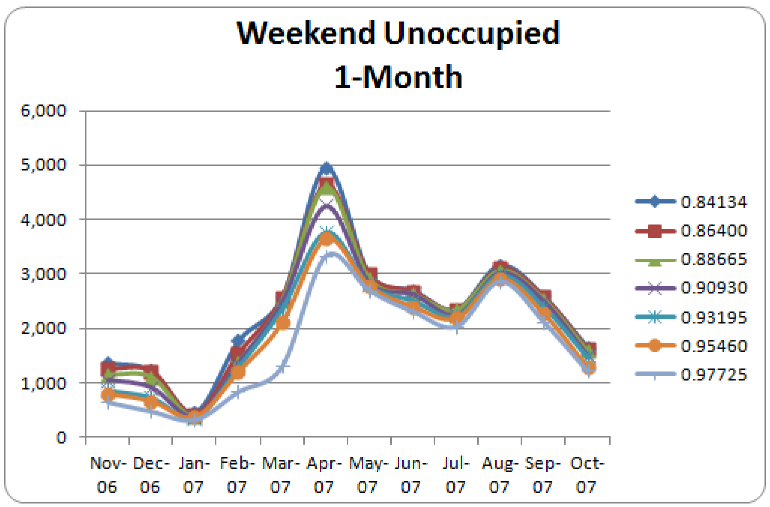


Figure 28: TEST run, Weekend Occupied, 5-month and 12-month

			Weekend Unoccupied											
			1-Month Windows											
Max		4,958	1	2	3	4	5	6	7	8	9	10	11	12
Min		224	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,367	1,228	461	1,775	2,573	4,958	3,001	2,697	2,352	3,166	2,594	1,657
	2	0.86400	1,265	1,222	420	1,544	2,573	4,656	2,989	2,682	2,343	3,108	2,585	1,624
	3	0.88665	1,156	1,110	384	1,378	2,517	4,598	2,909	2,634	2,342	3,066	2,536	1,589
	4	0.90930	1,055	917	382	1,372	2,487	4,254	2,843	2,630	2,247	3,028	2,500	1,505
	5	0.93195	868	740	364	1,302	2,365	3,765	2,797	2,512	2,229	2,981	2,401	1,479
	6	0.95460	796	668	375	1,211	2,112	3,653	2,781	2,405	2,188	2,920	2,277	1,289
	7	0.97725	653	489	332	850	1,315	3,339	2,707	2,321	2,046	2,873	2,130	1,228
Std Dev	1	0.84134	3,131	3,148	2,016	3,860	6,207	10,109	6,438	6,579	6,195	6,565	5,997	4,662
	2	0.86400	3,119	3,104	1,966	3,649	6,197	10,136	6,388	6,557	6,193	6,483	5,971	4,624
	3	0.88665	2,974	2,937	1,889	3,577	6,135	9,982	6,354	6,592	6,164	6,502	5,945	4,579
	4	0.90930	2,929	2,607	1,826	3,478	6,042	9,668	6,345	6,560	6,087	6,530	5,917	4,576
	5	0.93195	2,736	2,447	1,696	3,345	5,976	9,373	6,355	6,452	6,072	6,534	5,891	4,529
	6	0.95460	2,642	2,260	1,682	3,200	5,762	9,333	6,349	6,460	6,084	6,511	5,950	4,439
	7	0.97725	2,326	1,950	1,672	2,950	4,873	8,888	6,248	6,434	6,089	6,449	5,846	4,392

			Weekend Unoccupied											
			3-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,306	823	1,253	1,273	2,544	3,298	3,182	2,588	2,662	2,648	2,418	1,750
	2	0.86400	1,879	743	927	1,191	2,476	3,179	3,138	2,593	2,650	2,602	2,349	1,653
	3	0.88665	1,581	702	665	1,094	2,355	3,094	3,077	2,552	2,623	2,599	2,316	1,554
	4	0.90930	1,469	615	611	647	2,223	2,910	3,005	2,532	2,641	2,624	2,254	1,383
	5	0.93195	1,313	457	429	309	2,170	2,656	2,909	2,493	2,612	2,582	2,124	1,321
	6	0.95460	663	438	414	292	2,161	2,524	2,777	2,413	2,503	2,435	2,040	1,331
	7	0.97725	286	368	355	297	2,017	2,404	2,651	2,323	2,254	2,211	1,899	1,207
Std Dev	1	0.84134	5,245	2,534	3,230	4,066	6,115	7,429	7,363	6,341	6,395	6,175	5,646	4,056
	2	0.86400	4,715	2,445	2,627	3,880	6,123	7,320	7,311	6,345	6,370	6,159	5,595	3,955
	3	0.88665	4,446	2,402	2,291	3,740	6,089	7,289	7,304	6,365	6,403	6,163	5,595	3,954
	4	0.90930	4,151	2,229	2,159	2,865	6,004	7,040	7,363	6,382	6,418	6,165	5,591	3,899
	5	0.93195	4,030	1,952	1,932	1,709	5,926	6,966	7,323	6,294	6,372	6,133	5,576	3,810
	6	0.95460	2,506	1,850	1,840	1,582	5,905	6,870	7,206	6,237	6,255	6,065	5,571	3,799
	7	0.97725	1,638	1,761	1,760	1,497	5,774	6,800	7,151	6,284	6,229	6,078	5,528	3,724



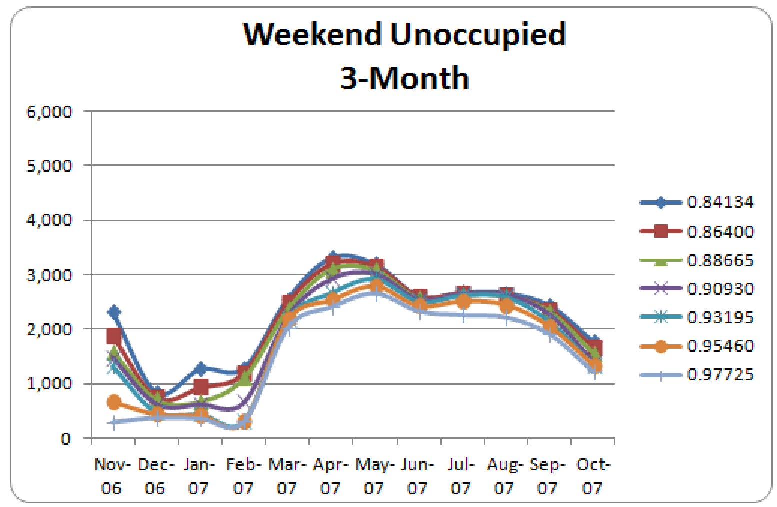
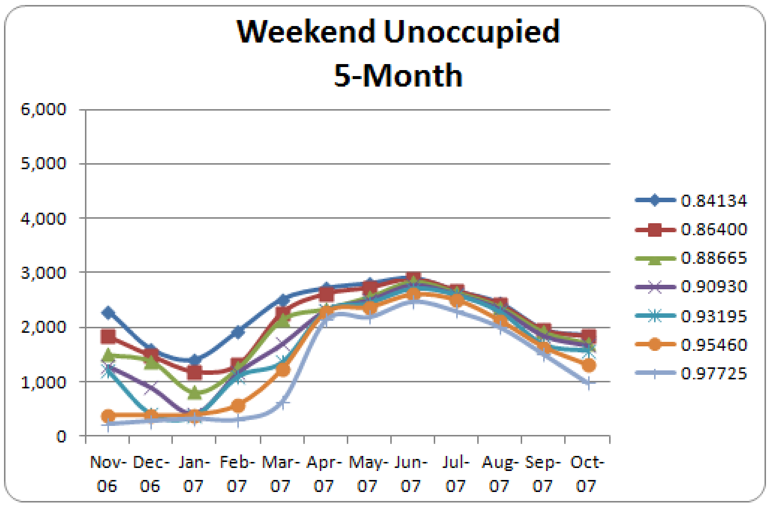


Figure 29: BASE run, Weekend Unoccupied, 1-month and 3-month

			Weekend Unoccupied												Weekend Unoccupied											
			5-Month Windows												12-Month											
Max		4,958	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Min		224	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	2,296	1,593	1,406	1,934	2,516	2,718	2,807	2,903	2,663	2,439	1,952	1,850	2,293	2,309	2,366	2,289	2,366	2,354	2,210	2,129	2,178	2,200	2,198	2,265
	2	0.86400	1,831	1,478	1,190	1,322	2,266	2,613	2,727	2,886	2,668	2,412	1,952	1,838	2,249	2,265	2,320	2,245	2,323	2,309	2,167	2,092	2,141	2,163	2,162	2,228
	3	0.88665	1,505	1,385	814	1,266	2,148	2,349	2,566	2,846	2,649	2,380	1,922	1,720	2,133	2,147	2,199	2,129	2,206	2,186	2,054	2,000	2,049	2,072	2,071	2,135
	4	0.90930	1,296	898	426	1,182	1,700	2,310	2,505	2,781	2,607	2,336	1,848	1,673	1,681	1,695	1,735	1,683	1,751	1,730	1,619	1,581	1,627	1,649	1,648	1,706
	5	0.93195	1,202	419	386	1,109	1,366	2,299	2,443	2,706	2,588	2,278	1,687	1,576	1,441	1,453	1,485	1,444	1,506	1,487	1,386	1,353	1,398	1,421	1,419	1,475
	6	0.95460	388	383	393	581	1,228	2,277	2,375	2,607	2,505	2,125	1,637	1,316	1,253	1,264	1,293	1,258	1,311	1,298	1,216	1,185	1,219	1,237	1,232	1,276
	7	0.97725	224	282	323	308	634	2,139	2,195	2,481	2,299	2,002	1,503	980	586	594	608	596	625	626	581	545	560	574	563	589
Std Dev	1	0.84134	5,513	4,264	3,582	4,751	5,865	6,193	6,707	6,896	6,266	5,889	4,764	4,194	5,293	5,338	5,473	5,292	5,448	5,396	5,188	5,144	5,167	5,174	5,184	5,241
	2	0.86400	4,897	4,060	3,161	4,243	5,818	6,208	6,696	6,891	6,276	5,848	4,776	4,194	5,234	5,278	5,411	5,233	5,391	5,334	5,126	5,090	5,118	5,126	5,137	5,198
	3	0.88665	4,522	3,931	2,455	4,120	5,754	6,107	6,646	6,933	6,299	5,854	4,767	4,186	5,171	5,217	5,347	5,172	5,330	5,274	5,066	5,029	5,057	5,067	5,077	5,138
	4	0.90930	4,268	3,010	1,901	3,881	5,044	6,170	6,686	6,926	6,258	5,840	4,689	4,118	4,609	4,650	4,766	4,613	4,758	4,708	4,519	4,482	4,511	4,524	4,531	4,593
	5	0.93195	4,199	1,850	1,816	3,790	4,764	6,142	6,538	6,798	6,227	5,848	4,662	4,018	4,360	4,399	4,506	4,365	4,506	4,456	4,268	4,230	4,267	4,284	4,292	4,361
	6	0.95460	2,076	1,776	1,777	2,430	4,419	6,131	6,476	6,810	6,215	5,855	4,693	3,884	4,005	4,044	4,146	4,015	4,144	4,101	3,935	3,898	3,923	3,934	3,938	3,991
	7	0.97725	1,524	1,680	1,717	1,607	2,655	6,004	6,419	6,730	6,197	5,735	4,573	3,445	2,419	2,449	2,508	2,451	2,545	2,542	2,410	2,317	2,347	2,374	2,352	2,416



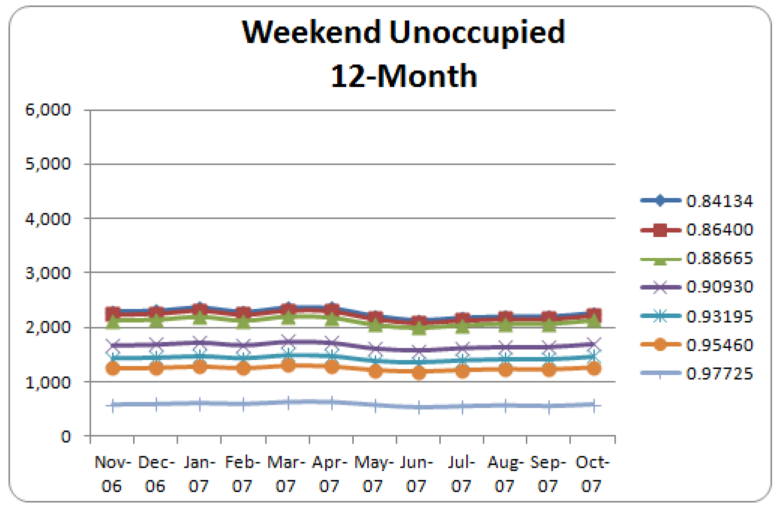


Figure 30: BASE run, Weekend Unoccupied, 5-month and 12-month

			Weekend Unoccupied											
			1-Month Windows											
Min		441	1	2	3	4	5	6	7	8	9	10	11	12
Max		5,790	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,427	1,524	1,283	2,008	2,847	5,790	3,819	3,103	2,649	3,689	3,232	1,900
	2	0.86400	1,356	1,469	1,229	1,890	2,796	5,721	3,748	3,032	2,596	3,594	3,129	1,894
	3	0.88665	1,261	1,407	1,143	1,788	2,746	5,484	3,688	3,003	2,545	3,512	3,040	1,857
	4	0.90930	1,139	1,258	968	1,717	2,556	5,095	3,617	2,936	2,429	3,390	2,912	1,729
	5	0.93195	1,022	1,166	870	1,589	2,403	4,582	3,595	2,861	2,350	3,320	2,873	1,590
	6	0.95460	916	1,099	881	1,331	2,035	4,439	3,555	2,746	2,250	3,210	2,787	1,509
	7	0.97725	753	957	893	1,188	1,743	3,922	3,491	2,613	2,106	2,998	2,688	1,427
Std Dev	1	0.84134	1,858	2,475	2,213	3,206	4,630	7,244	5,052	5,121	4,785	5,014	4,634	3,680
	2	0.86400	1,843	2,469	2,126	3,164	4,644	7,255	5,054	5,131	4,806	5,028	4,677	3,670
	3	0.88665	1,780	2,422	2,003	3,127	4,629	7,302	5,048	5,121	4,819	5,024	4,694	3,663
	4	0.90930	1,716	2,126	1,393	3,058	4,558	7,105	5,050	5,145	4,861	4,971	4,680	3,657
	5	0.93195	1,641	1,957	1,346	3,016	4,541	6,961	5,039	5,135	4,856	4,977	4,662	3,642
	6	0.95460	1,616	1,889	1,333	2,854	4,496	6,910	5,049	5,092	4,807	4,956	4,629	3,607
	7	0.97725	1,535	1,734	1,345	2,483	4,234	6,696	5,038	5,051	4,787	4,961	4,600	3,549

			Weekend Unoccupied											
			3-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,930	1,375	1,453	1,682	2,905	3,835	3,775	3,024	3,099	3,133	2,789	1,744
	2	0.86400	1,628	1,323	1,359	1,573	2,864	3,707	3,690	2,960	3,052	3,077	2,753	1,681
	3	0.88665	1,407	1,210	1,289	1,363	2,792	3,611	3,617	2,958	2,991	2,995	2,631	1,598
	4	0.90930	1,270	1,094	1,214	1,300	2,746	3,512	3,551	2,892	2,887	2,854	2,522	1,563
	5	0.93195	1,105	879	1,110	1,215	2,691	3,326	3,460	2,816	2,758	2,740	2,449	1,480
	6	0.95460	905	826	999	1,048	2,399	3,278	3,314	2,680	2,655	2,644	2,362	1,210
	7	0.97725	762	806	920	773	2,014	3,075	3,019	2,388	2,435	2,548	2,282	1,057
Std Dev	1	0.84134	3,882	2,089	2,627	3,613	4,770	5,538	5,683	4,962	4,999	4,799	4,503	3,046
	2	0.86400	3,776	2,064	2,262	3,462	4,755	5,507	5,694	4,978	5,000	4,804	4,485	3,030
	3	0.88665	3,609	1,918	2,108	3,026	4,771	5,506	5,698	4,966	4,993	4,819	4,486	3,034
	4	0.90930	3,462	1,828	2,038	2,693	4,740	5,475	5,675	4,945	4,975	4,831	4,460	3,041
	5	0.93195	3,335	1,428	1,857	2,142	4,684	5,454	5,647	4,967	4,984	4,828	4,457	3,028
	6	0.95460	2,547	1,383	1,588	1,953	4,612	5,343	5,612	4,967	4,966	4,794	4,427	3,006
	7	0.97725	2,142	1,395	1,357	1,207	4,556	5,276	5,517	4,897	4,850	4,740	4,378	2,942

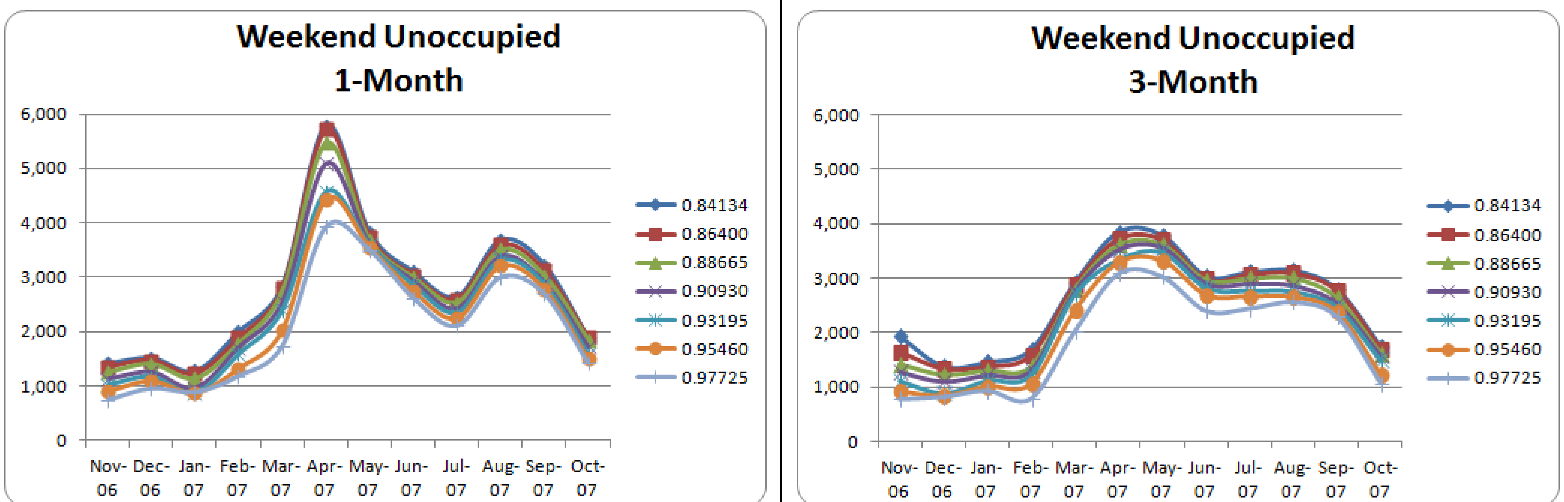
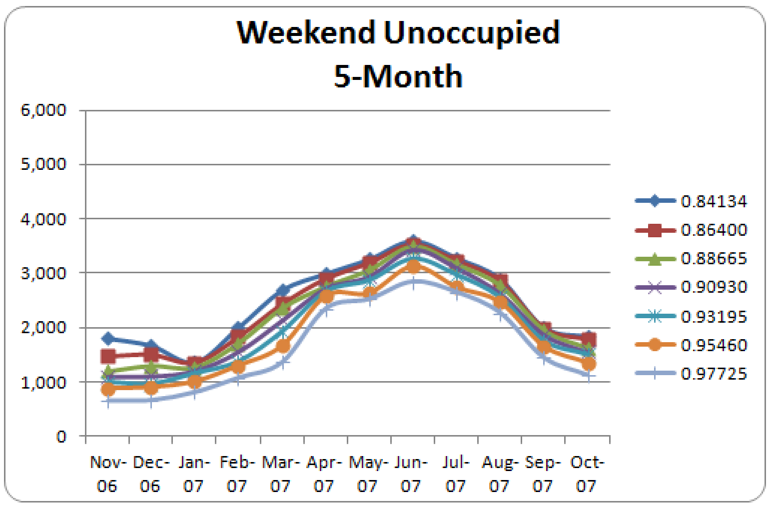


Figure 31: TEST run, Weekend Unoccupied, 1-month and 3-month

			Weekend Unoccupied												Weekend Unoccupied											
			5-Month Windows												12-Month											
Min		441	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Max		5,790	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.84134	1,811	1,680	1,375	2,010	2,692	2,998	3,262	3,593	3,281	2,900	2,011	1,844	1,755	1,776	1,840	1,876	1,785	1,726	1,770	1,745	1,802	1,857	1,866	1,838
	2	0.86400	1,475	1,503	1,361	1,832	2,436	2,896	3,190	3,526	3,226	2,854	1,989	1,785	1,694	1,687	1,749	1,798	1,632	1,561	1,604	1,594	1,629	1,705	1,683	1,737
	3	0.88665	1,199	1,301	1,275	1,718	2,355	2,752	3,056	3,486	3,158	2,772	1,955	1,629	1,630	1,565	1,489	1,650	1,493	1,448	1,487	1,501	1,518	1,440	1,496	1,558
	4	0.90930	1,114	1,116	1,215	1,567	2,141	2,728	2,949	3,440	3,105	2,636	1,881	1,555	1,287	1,297	1,357	1,437	1,359	1,318	1,332	1,340	1,281	1,268	1,258	1,329
	5	0.93195	997	964	1,159	1,375	1,929	2,658	2,869	3,263	2,968	2,553	1,755	1,505	1,000	1,074	1,113	1,224	1,144	992	1,066	1,034	1,017	1,102	982	994
	6	0.95460	891	907	1,015	1,307	1,671	2,593	2,645	3,137	2,751	2,471	1,660	1,359	894	957	937	937	887	793	782	779	864	792	795	843
	7	0.97725	661	669	817	1,082	1,365	2,348	2,522	2,842	2,647	2,268	1,448	1,133	514	694	661	573	578	479	460	443	445	475	441	485
Std Dev	1	0.84134	4,168	3,768	2,740	3,942	4,591	4,868	5,319	5,396	4,907	4,740	3,606	2,951	3,495	3,677	3,767	3,923	3,835	3,706	3,691	3,624	3,626	3,546	3,542	3,688
	2	0.86400	3,991	3,545	2,614	3,796	4,561	4,847	5,295	5,402	4,880	4,742	3,637	2,949	3,463	3,653	3,723	3,866	3,714	3,578	3,557	3,494	3,524	3,488	3,446	3,597
	3	0.88665	3,777	3,088	2,217	3,689	4,552	4,798	5,275	5,362	4,909	4,760	3,639	2,947	3,435	3,590	3,615	3,771	3,597	3,432	3,474	3,390	3,397	3,331	3,378	3,547
	4	0.90930	3,665	2,721	2,039	3,522	4,332	4,774	5,250	5,374	4,918	4,723	3,620	2,967	3,217	3,490	3,537	3,633	3,515	3,405	3,378	3,308	3,282	3,212	3,246	3,435
	5	0.93195	3,198	2,241	2,016	3,069	4,115	4,775	5,225	5,390	4,960	4,678	3,604	2,972	2,901	3,263	3,394	3,496	3,364	3,257	3,251	3,227	3,236	3,178	2,869	3,022
	6	0.95460	2,333	2,146	1,828	2,413	3,848	4,783	5,171	5,357	4,925	4,643	3,582	2,870	2,696	3,180	2,929	2,980	2,933	2,846	2,794	2,775	2,765	2,579	2,551	2,361
	7	0.97725	1,925	1,594	1,322	2,048	2,602	4,701	5,059	5,361	4,856	4,537	3,438	2,776	2,045	2,382	2,153	2,025	1,934	1,837	1,959	1,824	1,787	1,789	1,823	1,873



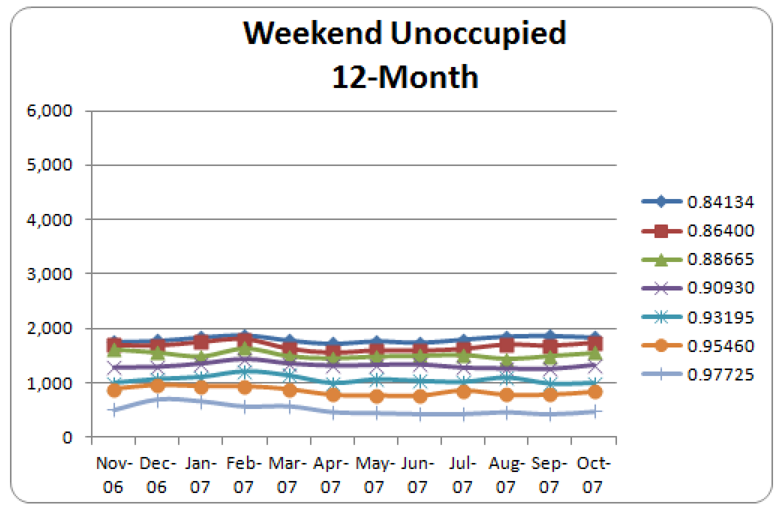


Figure 32: TEST run, Weekend Unoccupied, 5-month and 12-month

4.3 Data Quality

In order to perform a review of the data from the base and test periods, the daily total usage data for the eight buildings were plotted over a four-year period. Next, a 12 trailing month (TTM) average was calculated for the daily usages and a linear trend was added to determine the slope of the line. A positive slope indicates an increase in energy over time, while a negative one would imply a decreasing trend. The slopes of the lines for the TTM data are in Table 6. All eight of the graphs can be seen in Appendix D. Four of the buildings had negative slopes, while the remaining four had positive slopes. Using Meter 1-1, the slope would suggest a 5.4% increase in usage per year.

Table 6: Slopes of TTM building data for all daily energy usages over Base and Test period.

Meter	Slope
1-1	0.774
1-2	(1.606)
2-1	0.948
2-2	0.122

Meter	Slope
3-1	(0.172)
4-1	(2.445)
5-1	0.146
5-2	(0.105)

In addition, normal probability plots of the base period energy usage were created for all of the meters by period type. They can be seen in Appendix E. The plots utilized the same 16-months of data that from the base case. These plots were generated since the assumption in the model was that the daily usage values were normally distributed. As discussed previously, the thresholds for triggering the alarms were calculated under that assumption.

Approximately one-third of the meters have profiles with good fits to a normal distribution. However, in the majority of the cases, it is clear that above and below 2 standard deviations, as the data approaches the tails, the plots broke down and changed their slope. For example, this can be demonstrated by using two graphs from Building 5, Meter1 as seen in Figure 33 and Figure 34. The weekday occupied graph has an excellent fit as it is very straight. However, looking at the lower end of the weekend unoccupied, the data clearly drops off which would suggest that the data is skewed.

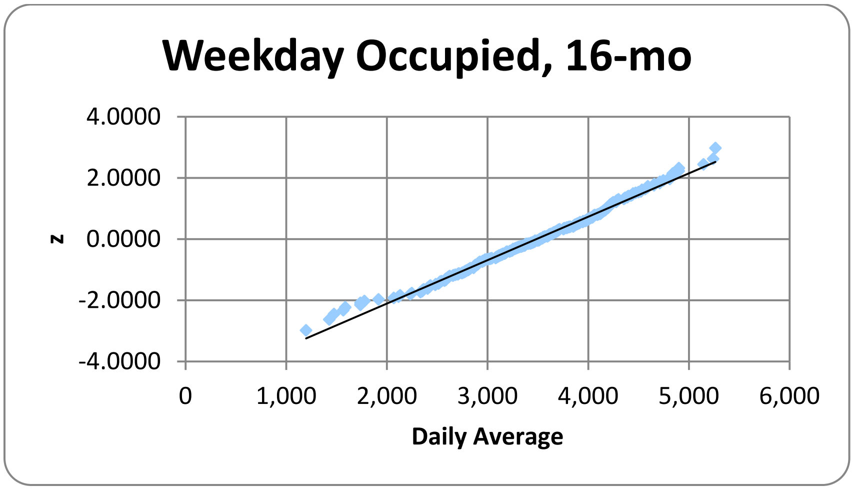


Figure 33: Building 5, Meter 1, normal probability plot for the weekday occupied period

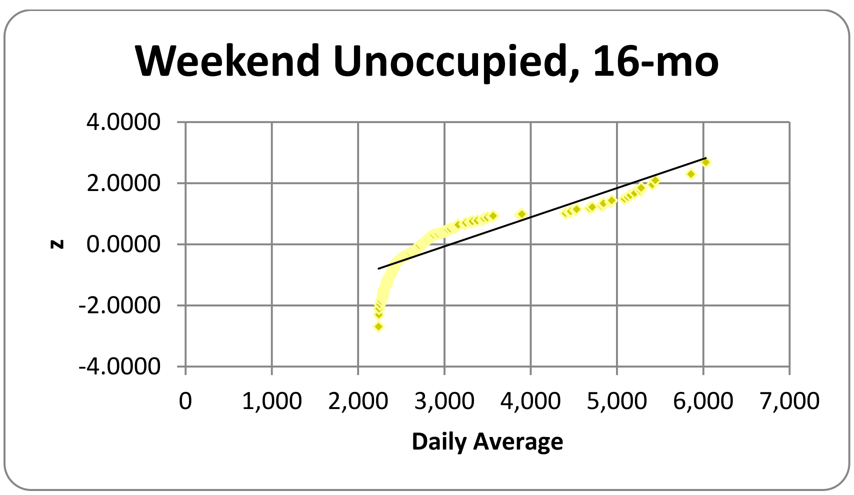


Figure 34: Building 5, Meter 1, normal probability plot for the weekend unoccupied period

4.4 Analysis of Results

By observation, the graphs of the data from the base and test runs for all of the windows have very similar profiles. A simple test of the resulting data shows a high degree of correlation between the Base and Test results. All of the result arrays for the averages and standard deviations were tested for correlation. The results can be seen in Table 7 and Table 8.

Table 7: Correlations of Base and Test run result arrays for data averages

Correlation	1-month	3-month	5-month	12-month
Risk Management	0.976	0.980	0.972	0.977
Weekday Occupied	0.942	0.951	0.972	0.903
Weekday Unoccupied	0.927	0.939	0.952	0.936
Weekend Occupied	0.952	0.828	0.632	0.932
Weekend Unoccupied	0.981	0.963	0.940	0.975

Table 8: Correlations of Base and Test run result arrays for standard deviations

Correlation	1-month	3-month	5-month	12-month
Risk Management	0.953	0.970	0.957	0.957
Weekday Occupied	0.863	0.921	0.938	0.900
Weekday Unoccupied	0.954	0.949	0.950	0.906
Weekend Occupied	0.948	0.876	0.745	0.923
Weekend Unoccupied	0.988	0.993	0.982	0.962

For the data average array, with the exception of the 3-month and 5-month windows for the Weekend Occupied period type, all of the correlations were above 0.9. For the standard deviation array, all of the correlations were above 0.9, with the exception of the 3-month and 5-month windows for the Weekend Occupied period type, plus the Weekday Occupied 1-month window.

For all of the period types, there is a smoothing of the profiles as the window size increases. In all cases, the 1-month period had the most dramatic changes with many peaks and valleys. The size of the threshold had little to no effect on this. For the 12-month window, the result was an essentially flat profile. The 5-month window had the smoothest profile of the three short data sets.

The 5-month window most clearly demonstrates the seasonality of the various buildings' energy profiles. The risk management values in the winter are mostly lower. This can be explained by the increased electrical usage of several of the buildings in the winter. With an increased usage during those months, the alarms throughout the year would not be triggered much less frequently.

It is clear that when utilizing this model, the shorter the data set, the less predictive value there is for setting alarming thresholds for an entire year. One example of this can be seen by reviewing the Weekday Occupied 5-month Test results, as seen in Table 9 and Figure 35. For the months of June through August, the results for the seventh threshold were all greater than the first threshold for the months of January and February.

Table 9: Weekday Occupied Test run results for the 5-month window

			Weekday Occupied											
			5-Month Windows											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.8413	3,450	3,715	2,984	3,026	3,460	3,755	4,040	4,533	4,327	3,719	2,984	2,443
	2	0.8640	3,285	3,433	2,786	2,873	3,380	3,605	3,875	4,366	4,153	3,613	2,834	2,339
	3	0.8866	2,959	3,239	2,591	2,686	3,158	3,488	3,724	4,216	4,011	3,469	2,688	2,183
	4	0.9093	2,681	2,910	2,398	2,433	2,794	3,164	3,517	3,984	3,869	3,390	2,594	1,978
	5	0.9319	2,291	2,584	2,208	2,074	2,458	2,947	3,347	3,788	3,757	3,303	2,423	1,707
	6	0.9546	1,857	2,144	1,890	1,609	1,879	2,671	3,097	3,616	3,560	3,162	2,266	1,476
	7	0.9772	1,500	1,610	1,422	1,049	1,191	2,277	2,773	3,331	3,396	3,030	2,110	1,020

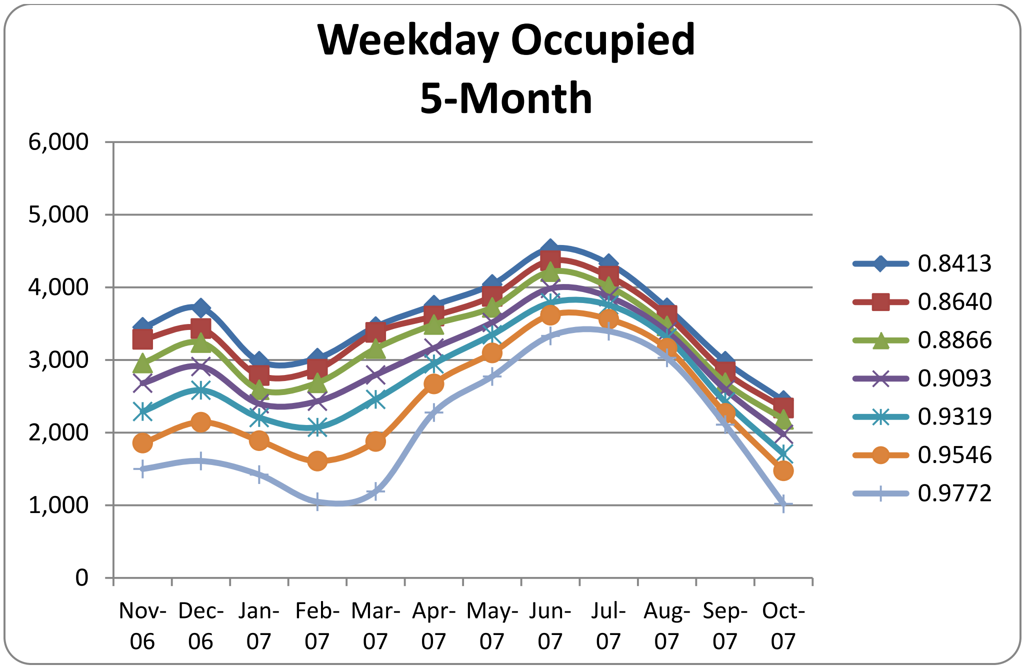


Figure 35: Weekday Occupied Test run results for the 5-month window

While the 5-month window clearly has the least variability of the three short data sets, it also appears to be more valuable in setting alarm thresholds than the 12-month window. Due to the loss of seasonality and flatness in the profiles, if the 12-month window were used, there would be lost opportunities for investigating alarms in the summer and winter. For example, by utilizing the weekday occupied results again, the rate of alarming in the summer would be lower than needed, leading to missed opportunities to investigate usage.

Table 10: Weekday Occupied Test run results for the 12-month window

			Weekday Occupied											
			12-Month											
			1	2	3	4	5	6	7	8	9	10	11	12
			Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07	May-07	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07
Average	1	0.8413	2,600	2,563	2,776	3,020	3,135	3,057	3,013	3,006	2,975	2,811	2,884	2,830
	2	0.8640	2,472	2,414	2,632	2,776	2,829	2,787	2,776	2,787	2,801	2,610	2,679	2,592
	3	0.8866	2,280	2,184	2,381	2,577	2,618	2,538	2,560	2,556	2,548	2,442	2,439	2,335
	4	0.9093	2,125	2,001	2,184	2,324	2,385	2,248	2,284	2,265	2,167	2,118	2,124	2,066
	5	0.9319	1,866	1,769	1,756	1,980	2,047	2,031	2,046	2,016	2,019	1,801	1,756	1,729
	6	0.9546	1,596	1,454	1,445	1,593	1,676	1,585	1,603	1,623	1,611	1,453	1,402	1,298
	7	0.9772	1,208	1,116	1,098	1,141	1,142	1,038	1,050	1,042	1,088	1,010	1,061	966

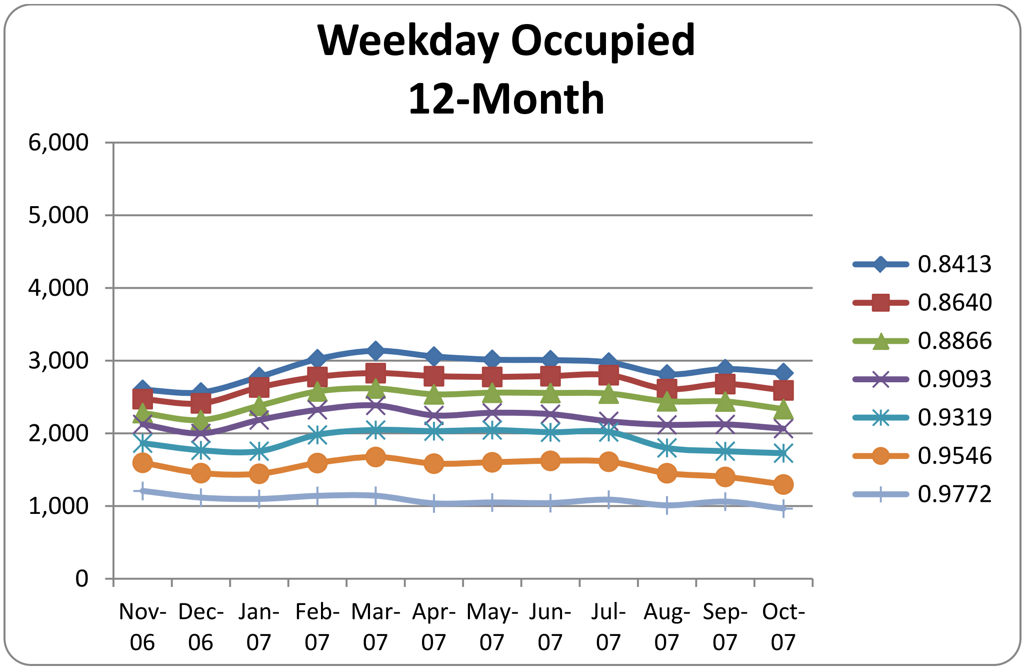


Figure 36: Weekday Occupied Test run results for the 12-month window

The ultimate question for future testing and examination of other building portfolios is, what should the appropriate level for setting thresholds be? In order draw a final conclusion, the arrays of the values found in the results of Figure 13 through Figure 32 were simplified by averaging the values over the twelve test months for each threshold. These averages were then plotted against their respective threshold values. Next, a second order polynomial was calculated for its best curve fit. Using the equation for the three window sizes, the parabolic curves were extended by calculating their theoretical values using the two probability values of 0.500 and 1.000. Then, for each equation, the derivative was taken. For any 2nd order equation of the form

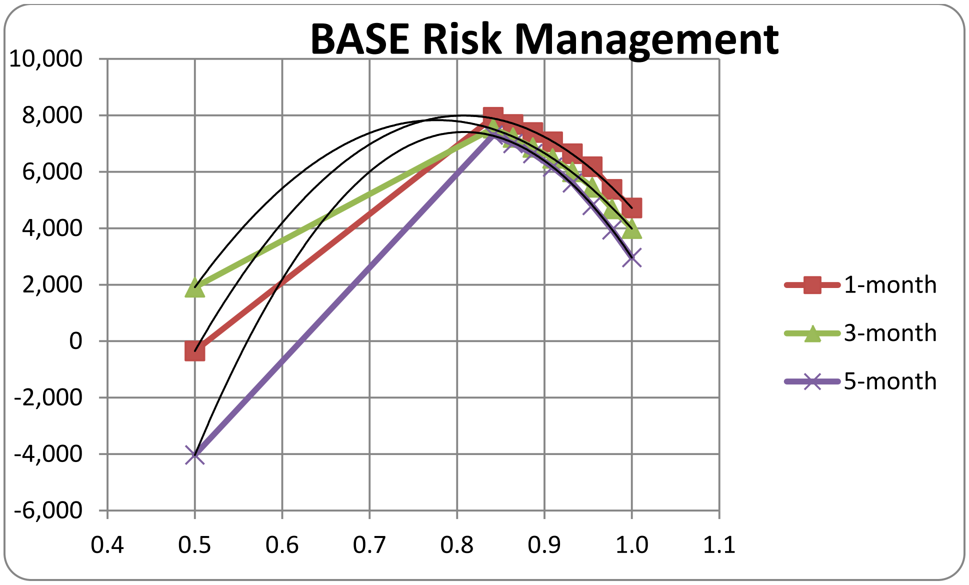
$$y = ax^2 + bx + c,$$

by taking the derivative and then by setting the resulting equation to be equal to 0, the maxima value is found to be:

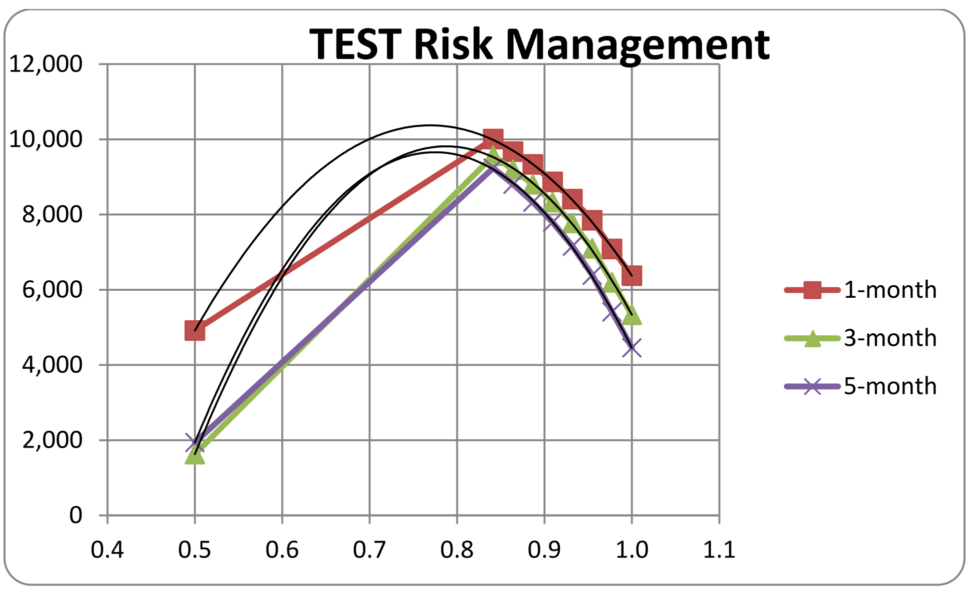
$$x^* = \frac{-b}{2a}$$

This optimal value was calculated for the Base and Test cases in each of the three short window sizes and four period types plus the total Risk Management type. Finally, using the optimal probabilities, the standard deviation was calculated. All of the graphs and results can be found in the tables on the following five pages.

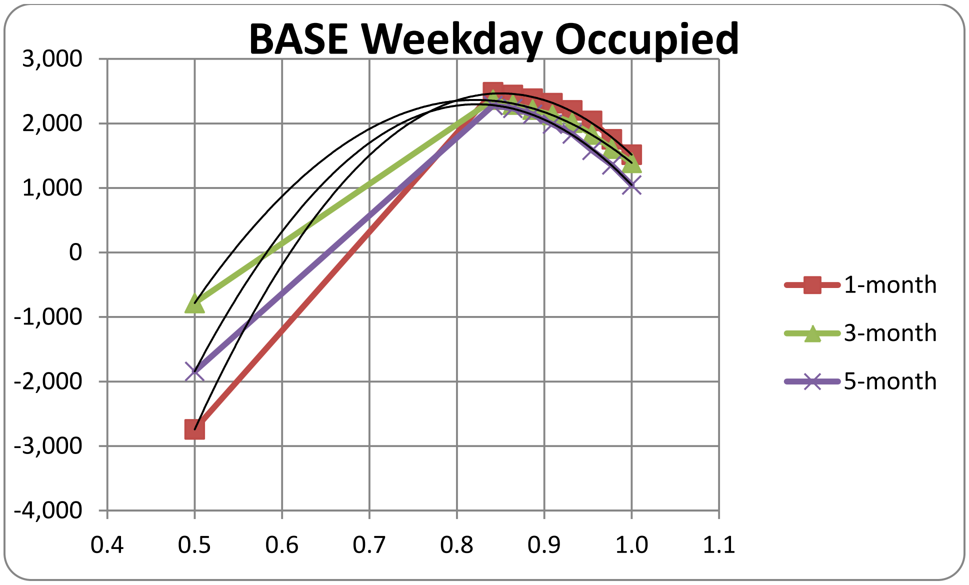
An examination of the results shows that the optimal point is found to be in the range between 0.8 and 1.0.



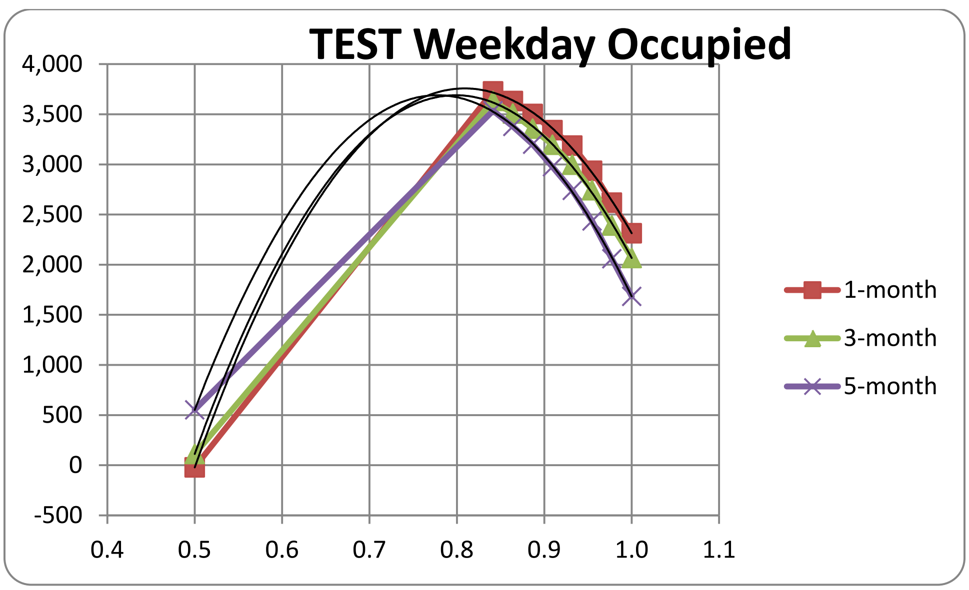
Window	Best Fit	x*	z (x*)
1-month	$y = -88{,}250x^2 + 142{,}507x + -49{,}537$	0.8074	0.8684
3-month	$y = -77{,}225x^2 + 119{,}992x + -38{,}779$	0.7769	0.7618
5-month	$y = -120{,}632x^2 + 194{,}936x + -71{,}341$	0.8080	0.8705



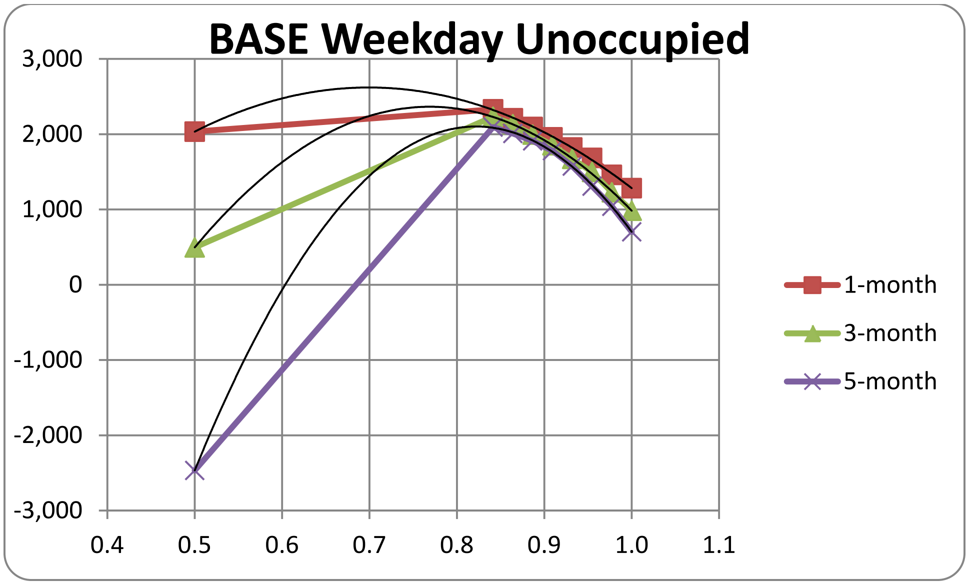
Window	Best Fit	x'	z (x*)
1-month	$y = -75{,}194x^2 + 115{,}707x + -34{,}140$	0.7694	0.7368
3-month	$y = -99{,}097\ x^2 + 156{,}058x + -51{,}626$	0.7874	0.7974
5-month	$y = -102{,}364\ x^2 + 158{,}581x + -51{,}761$	0.7746	0.7541



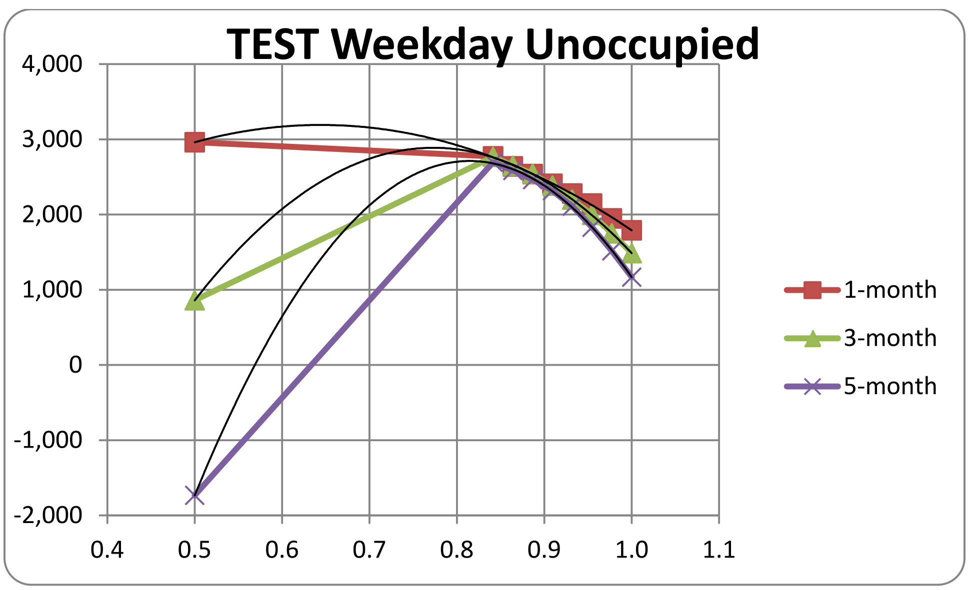
Window	Best Fit	x*	z (x*)
1-month	$y = -42{,}429x^2 + 72{,}159x + -28{,}214$	0.8503	1.0379
3-month	$y = -30{,}496x^2 + 50{,}090x + -18{,}204$	0.8213	0.9202
5-month	$y = -39{,}777x^e + 65{,}442x + -24{,}620$	0.8226	0.9254



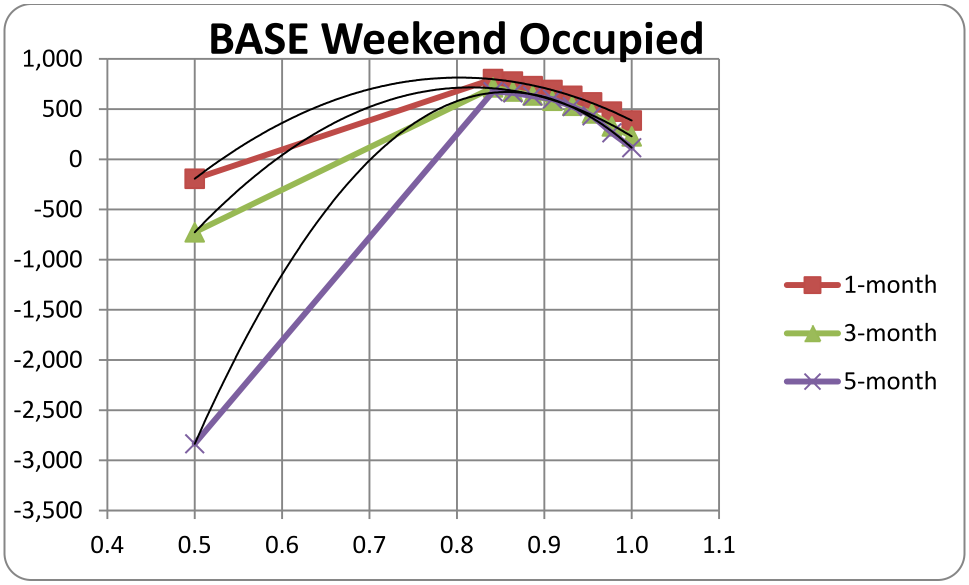
Window	Best Fit	x*	z (x*)
1-month	$y = -39{,}579x^2 + 64{,}039x + -22{,}146$	0.8090	0.8742
3-month	$y = -40{,}081x^2 + 64{,}030x + -21{,}883$	0.7988	0.8372
5-month	$y = -40{,}643x^2 + 63{,}227x + -20{,}901$	0.7778	0.7649



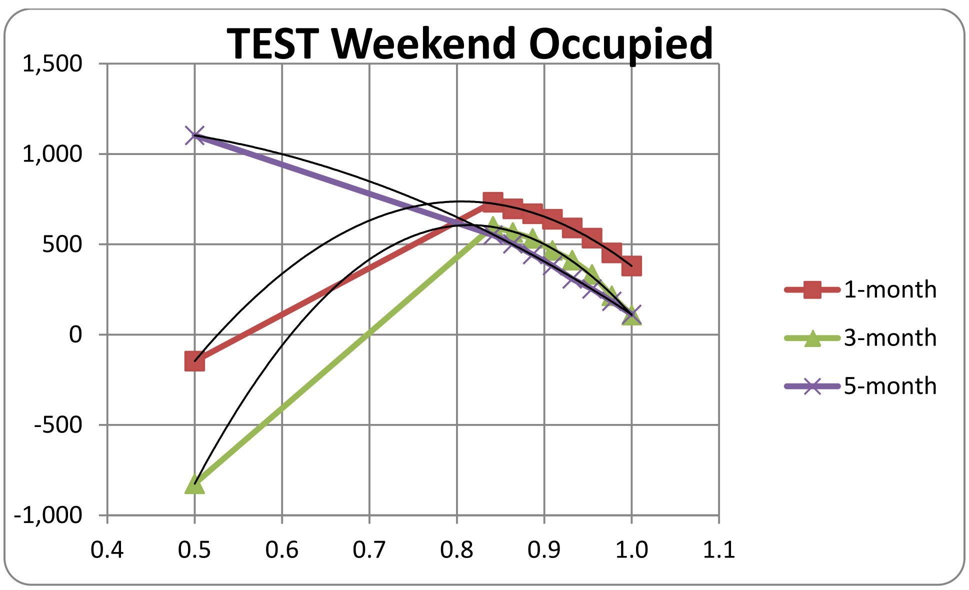
Window	Best Fit	x*	z (x*)
1-month	$y = -14{,}776x^2 + 20{,}663x + -4{,}603$	0.6992	0.5221
3-month	$y = -25{,}851x^2 + 39{,}753x + -12{,}918$	0.7689	0.7351
5-month	$y = -44{,}106x^2 + 72{,}499x + -27{,}690$	0.8219	0.9225



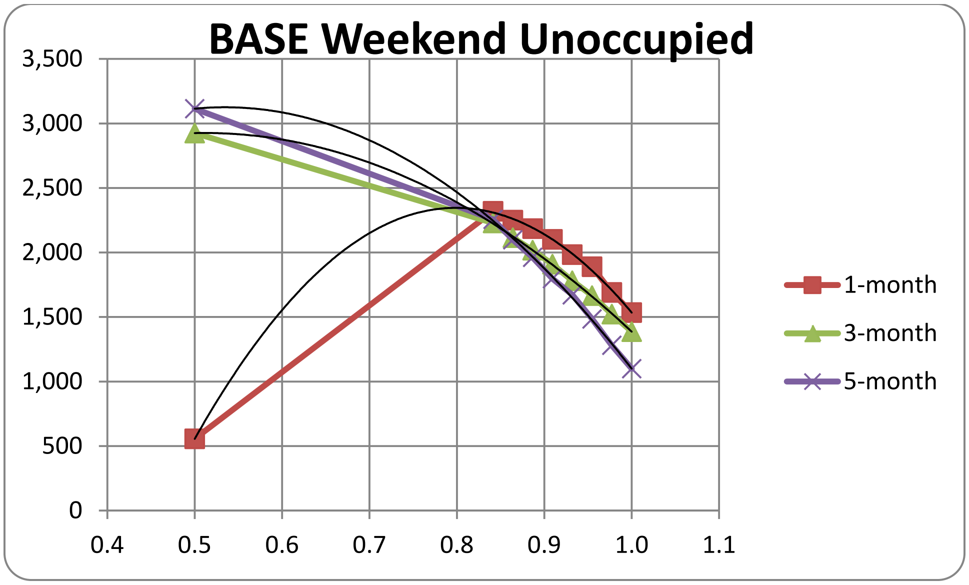
Window	Best Fit	x*	z (x*)
1-month	$y = -11{,}063x^2 + 14{,}253x + -1{,}399$	0.6442	0.3696
3-month	$y = -27{,}203x^2 + 42{,}058x + -13{,}368$	0.7730	0.7489
5-month	$y = -44{,}928x^2 + 73{,}194x + -27{,}098$	0.8146	0.8948



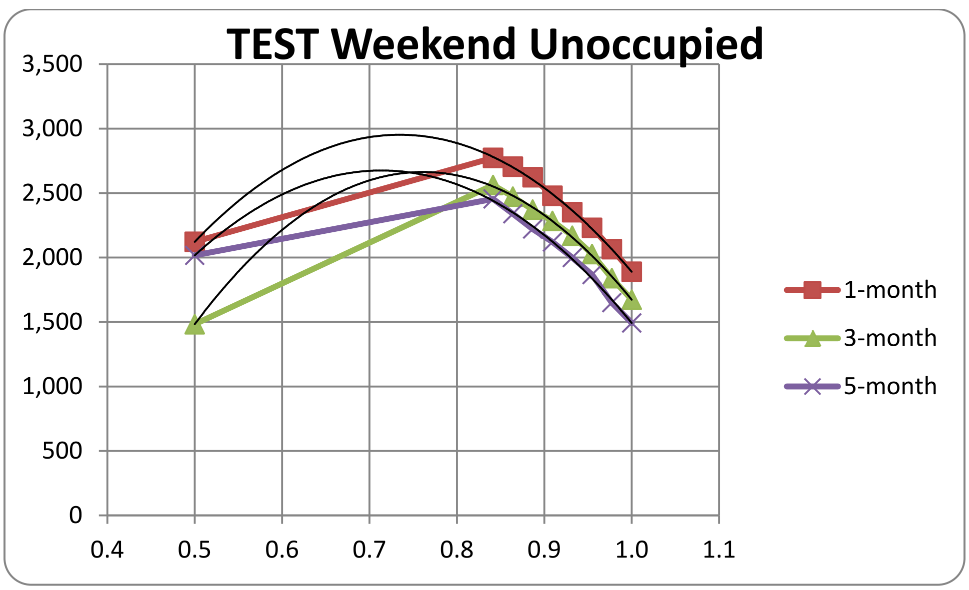
Window	Best Fit	x*	z (x*)
1-month	y = -10,994x^2 + 17,652x + -6,272	0.8028	0.8516
3-month	y = -14,438x^2 + 23,568x + -8,902	0.8162	0.9010
5-month	y = -27,394x^2 + 46,992x + -19,482	0.8577	1.0700



Window	Best Fit	x*	z (x*)
1-month	y = -9,462x^2 + 15,246x + -5,404	0.8057	0.8620
3-month	y = -14,478x^2 + 23,585x + -8,998	0.8145	0.8946
5-month	y = -2,367x^2 + 1,569x + 910	0.3315	(0.4359)



Window	Best Fit	x*	z (x*)
1-month	y = -20,049x^2 + 32,030x + -10,447	0.7988	0.8373
3-month	y = -6,441x^2 + 6,584x + 1,244	0.5111	0.0277
5-month	y = -9,355x^2 + 10,002x + 452	0.5346	0.0869



Window	Best Fit	x*	z (x*)
1-month	y = -15,089x^2 + 22,170x + -5,191	0.7346	0.6268
3-month	y = -17,334x^2 + 26,382x + -7,375	0.7610	0.7095
5-month	y = -14,424x^2 + 20,587x + -4,671	0.7136	0.5641

Chapter 5: Conclusions

5.1 Summary

The goal of the study was to determine what an optimal threshold value would be for setting alarm threshold to test energy usage in a building using short data sets. Several years of 15-minute interval data was utilized from five buildings in Jacksonville, Florida. For this paper, the concept of “risk management” was developed in order to calculate a value, in dollars which would normalize the cost of the time associated with investigating alarms, and the benefit of energy “saved” which would result from the same investigation.

The results of the decision model developed for this study indicate that when utilizing short data sets for predicting annual alarming thresholds, a 5-month window provides better results than a 1-month or 3-month window. An assumption in the original formulation of the model was that the optimal value for establishing alarm thresholds would most likely fall somewhere between 1.0 and 2.0 standard deviations. In the end, it was observed that the optimal value for setting the thresholds may be in the 0.8 to 1.0 standard deviation range above the average for this data set.

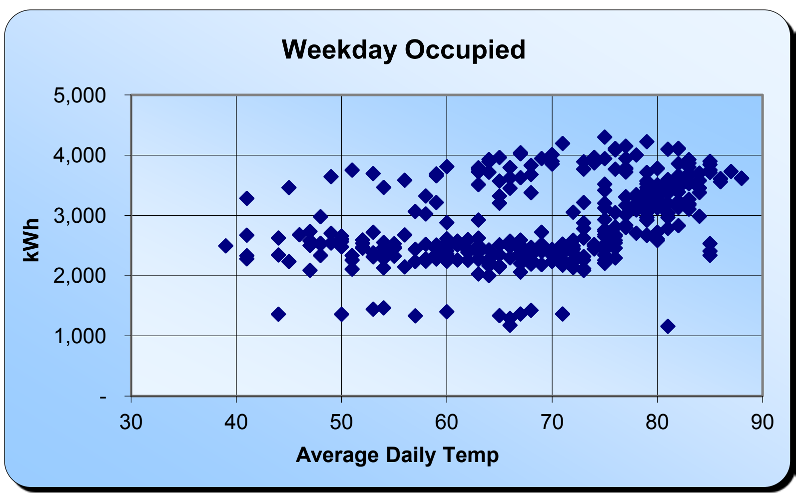
5.2 Recommendations for Future Research

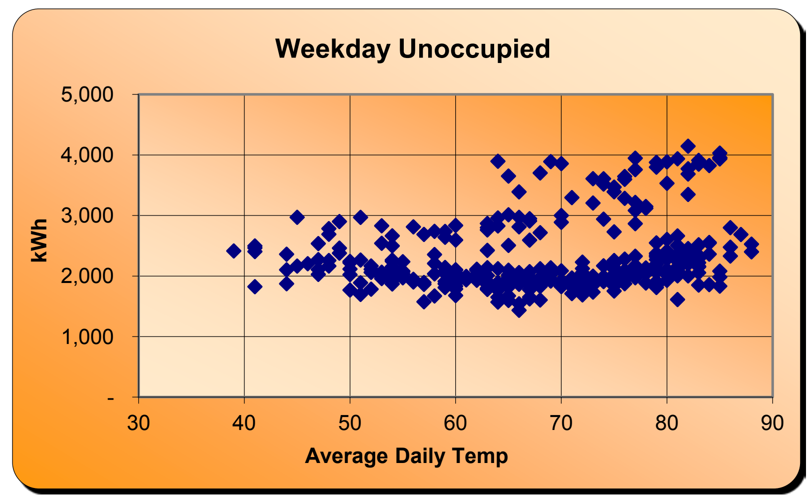
Due to the nature of this project, this study was limited in scope and in several of its assumptions. Based upon the work done, there are various items that could be addressed in future studies.

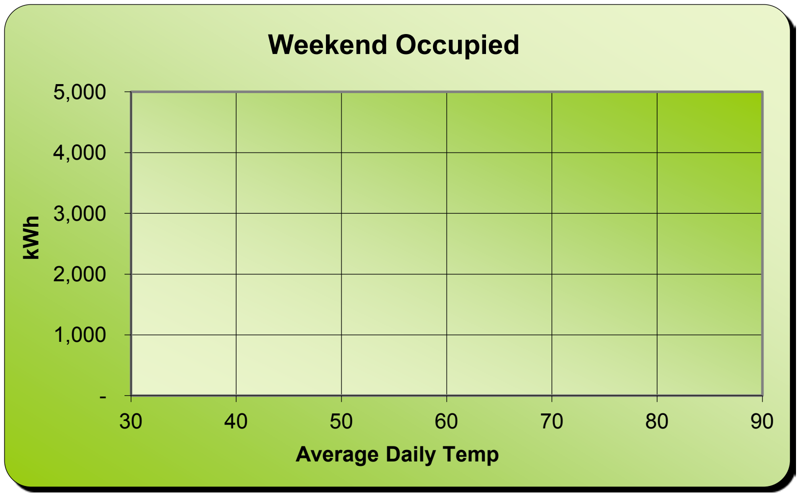
1) Modifications could be made to the calculation of the risk management value. In this paper, the projected savings and load factor multipliers were factored in after the alarm response and alarm length were determined. This variable should be part of the formula for determining response type and length, so that critical buildings would be considered more carefully.
2) The interval data could be corrected so that it is more normal. There are various statistical procedures that could be used, but one might be to perform a log normal transformation on the data.
3) The thresholds were calculated using the same probability for all the meters at any given time during the runs. Efforts to optimize the thresholds would provide further benefit. The result would be lower thresholds for larger buildings, while smaller, less "risky" buildings would have higher thresholds.
4) Variations in the time horizons might provide better insight on how useful short data sets are. Instead of running the model by testing the windows against twelve or twenty-four months of data, further study could result from testing the short data sets against shorter time horizons.
5) A sensitivity analysis could be performed on the various response type and duration distributions to see how their change might impact the results.

Appendix A:
Building Profiles by Occupancy Type

Figure 37: Building 1, Meter 1, energy usage by period type






Weekend Occupied
5,000
4,000
3,000
2,000
1,000
-
kWh
30
40
50
60
70
80
90
Average Daily Temp

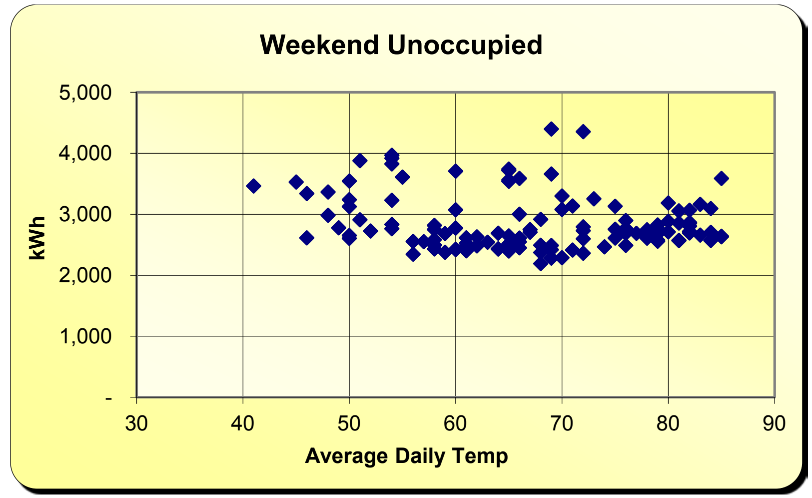

Weekend Unoccupied
5,000
4,000
3,000
2,000
1,000
-
kWh
30
40
50
60
70
80
90
Average Daily Temp

Figure 38: Building 1, Meter 2, energy usage by period type

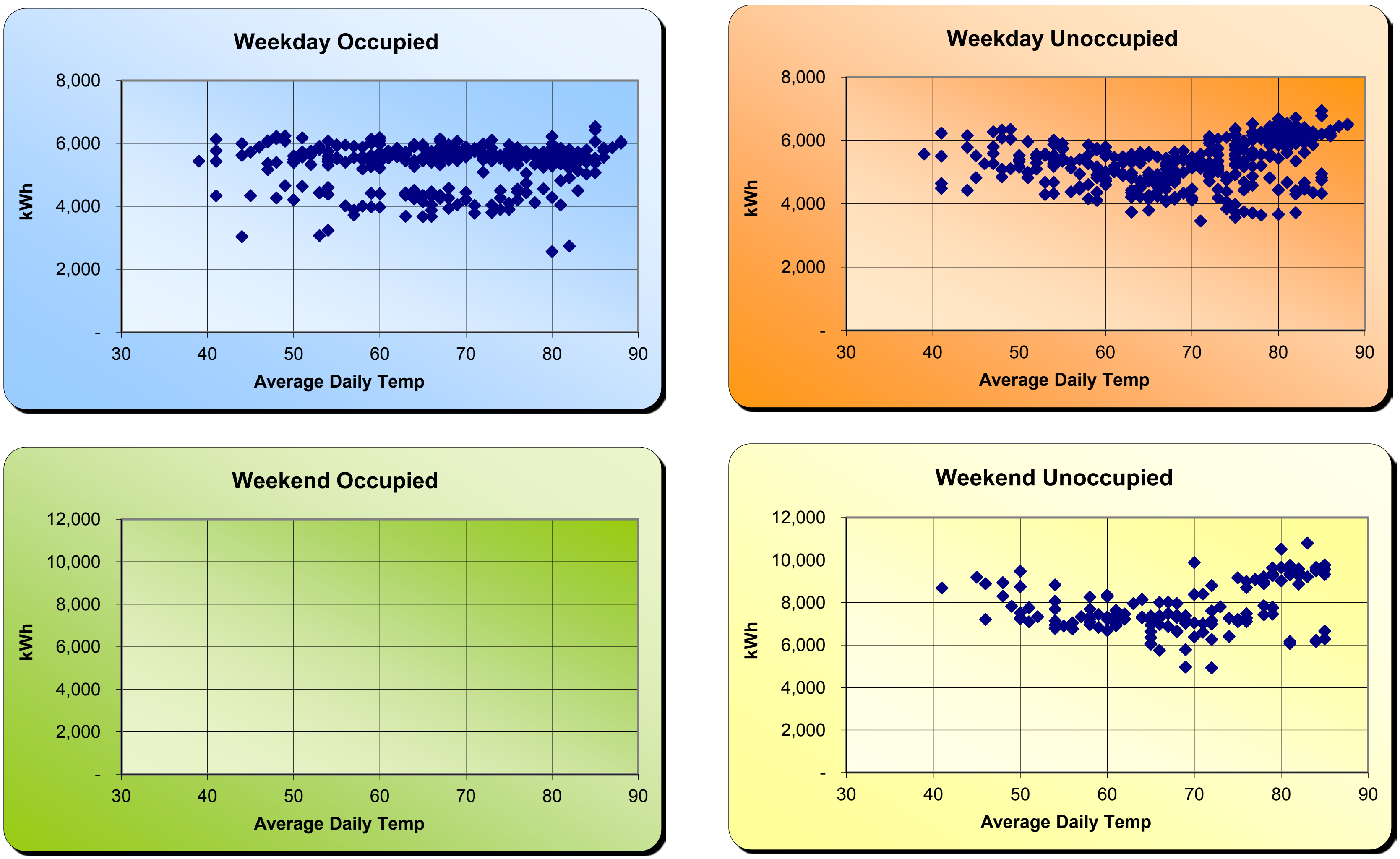


Figure 39: Building 2, Meter 1, energy usage by period type

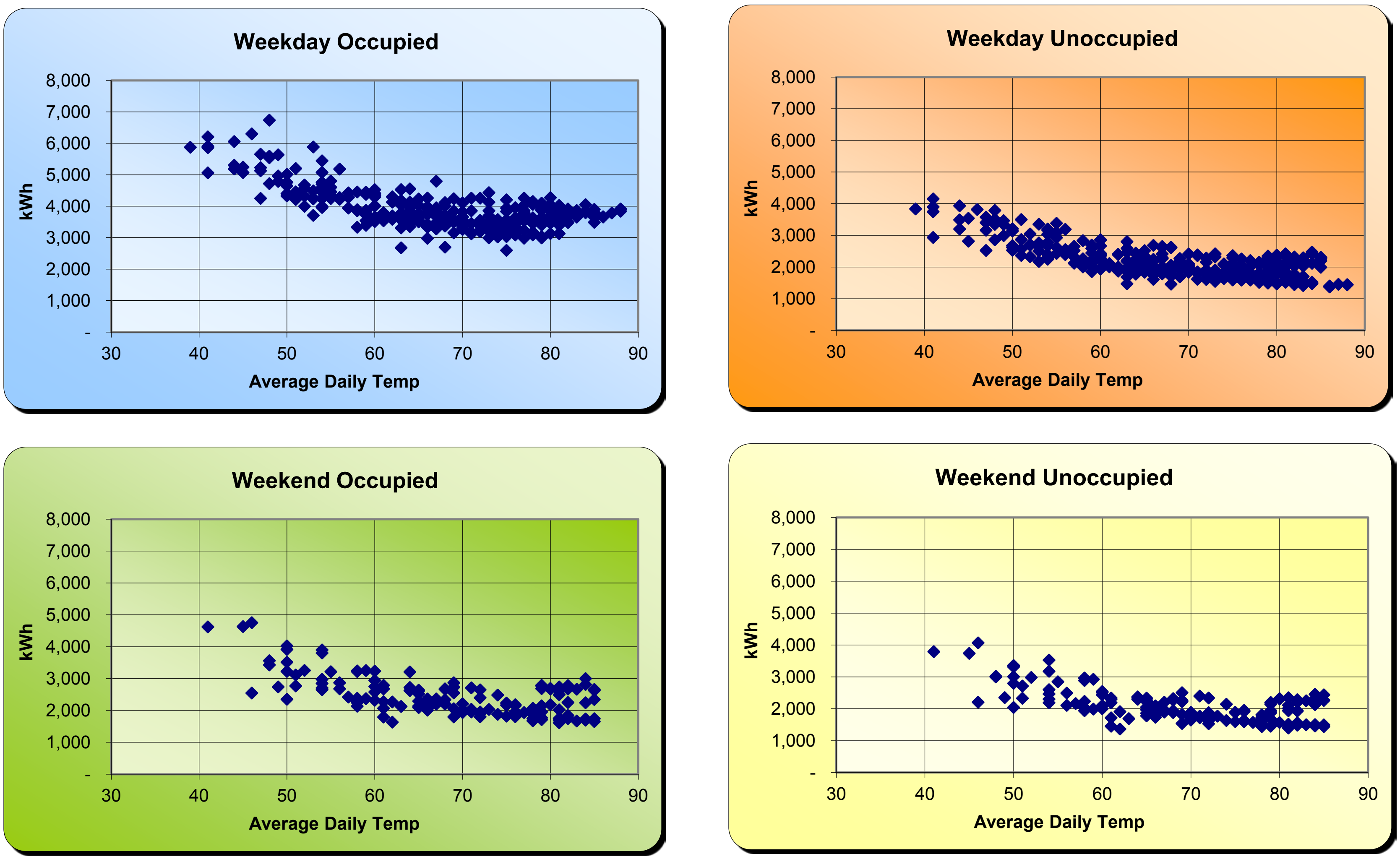


Figure 40: Building 2, Meter 2, energy usage by period type

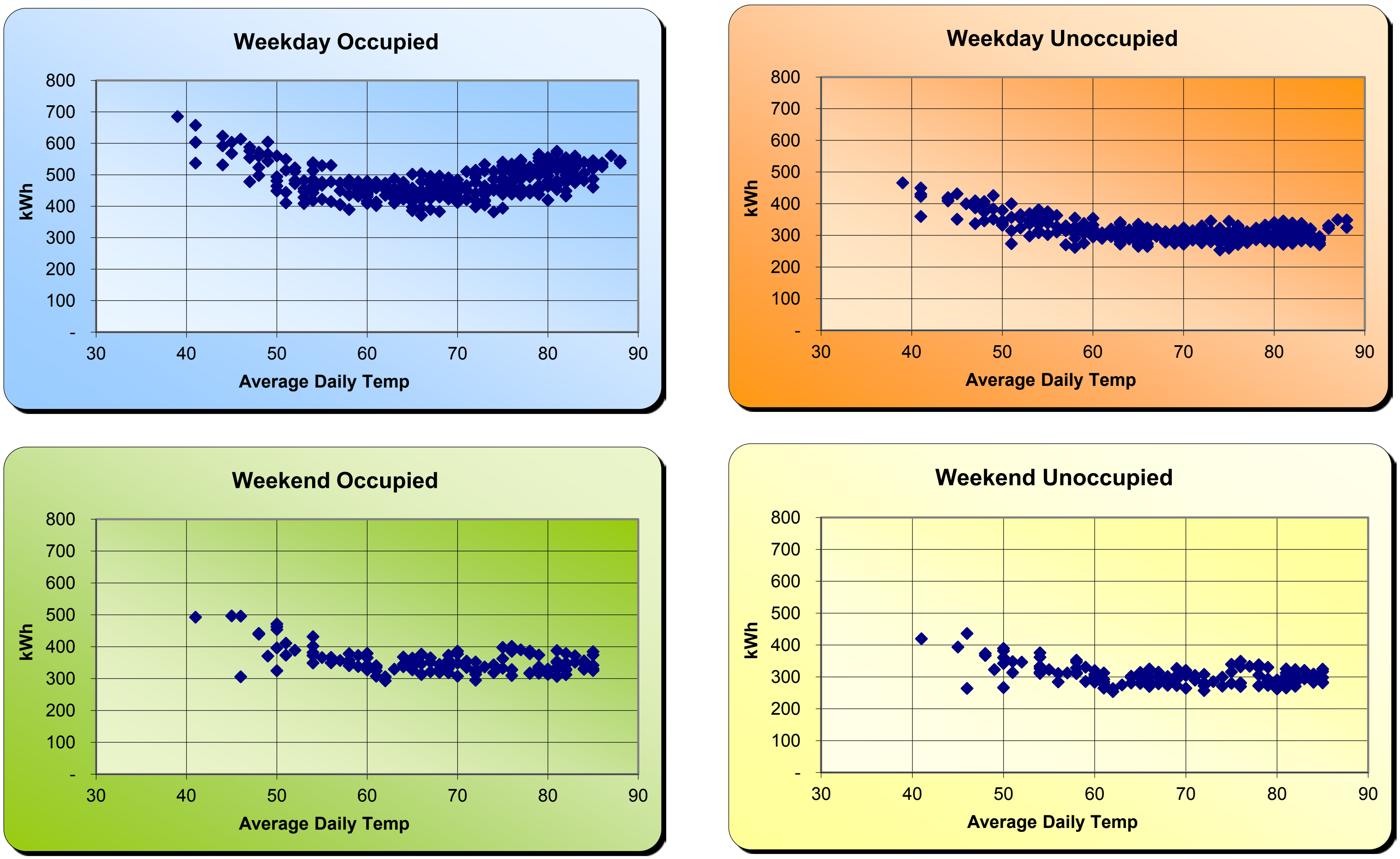


Figure 41: Building 3, Meter 1, energy usage by period type

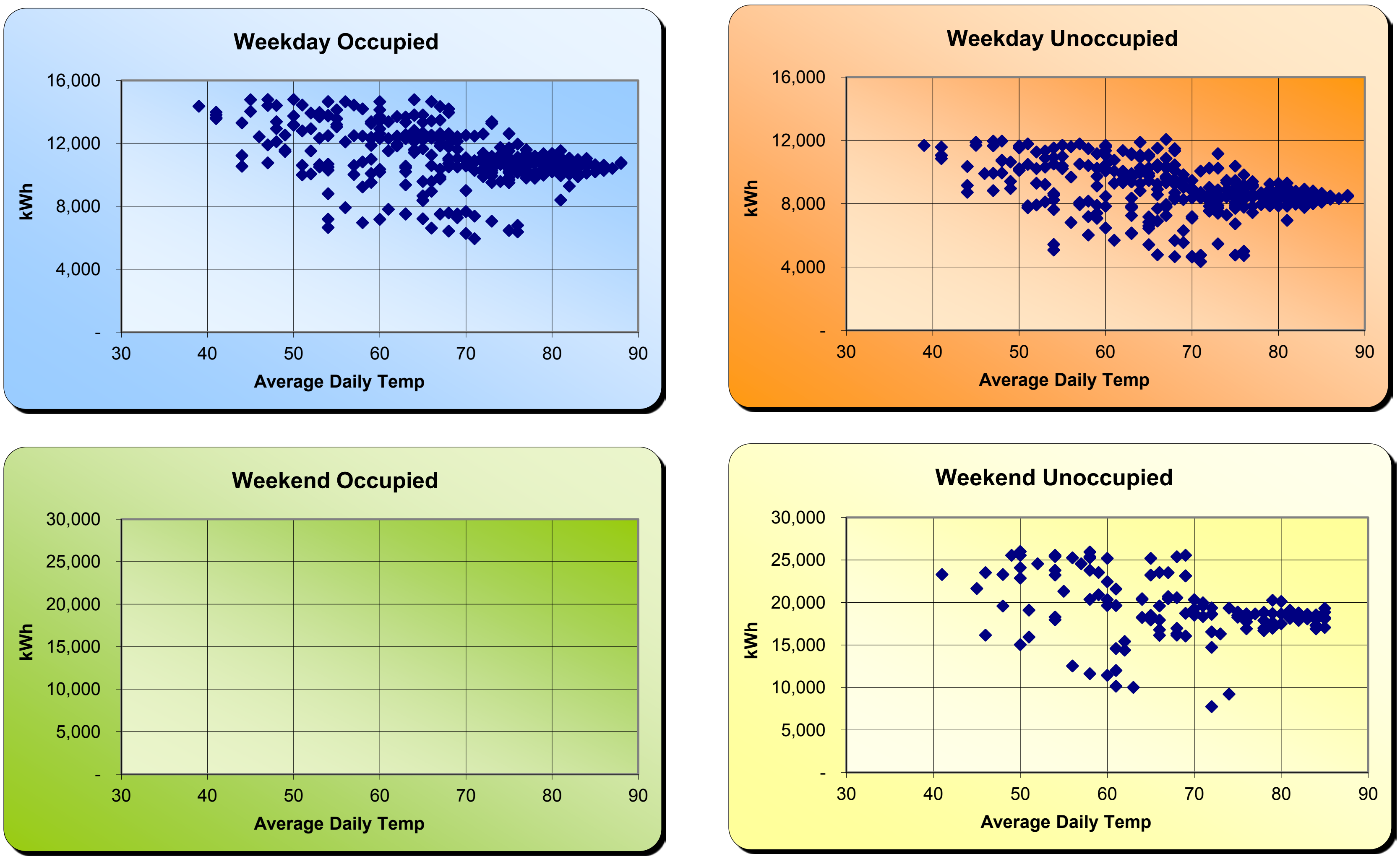


Figure 42: Building 4, Meter 1, energy usage by period type

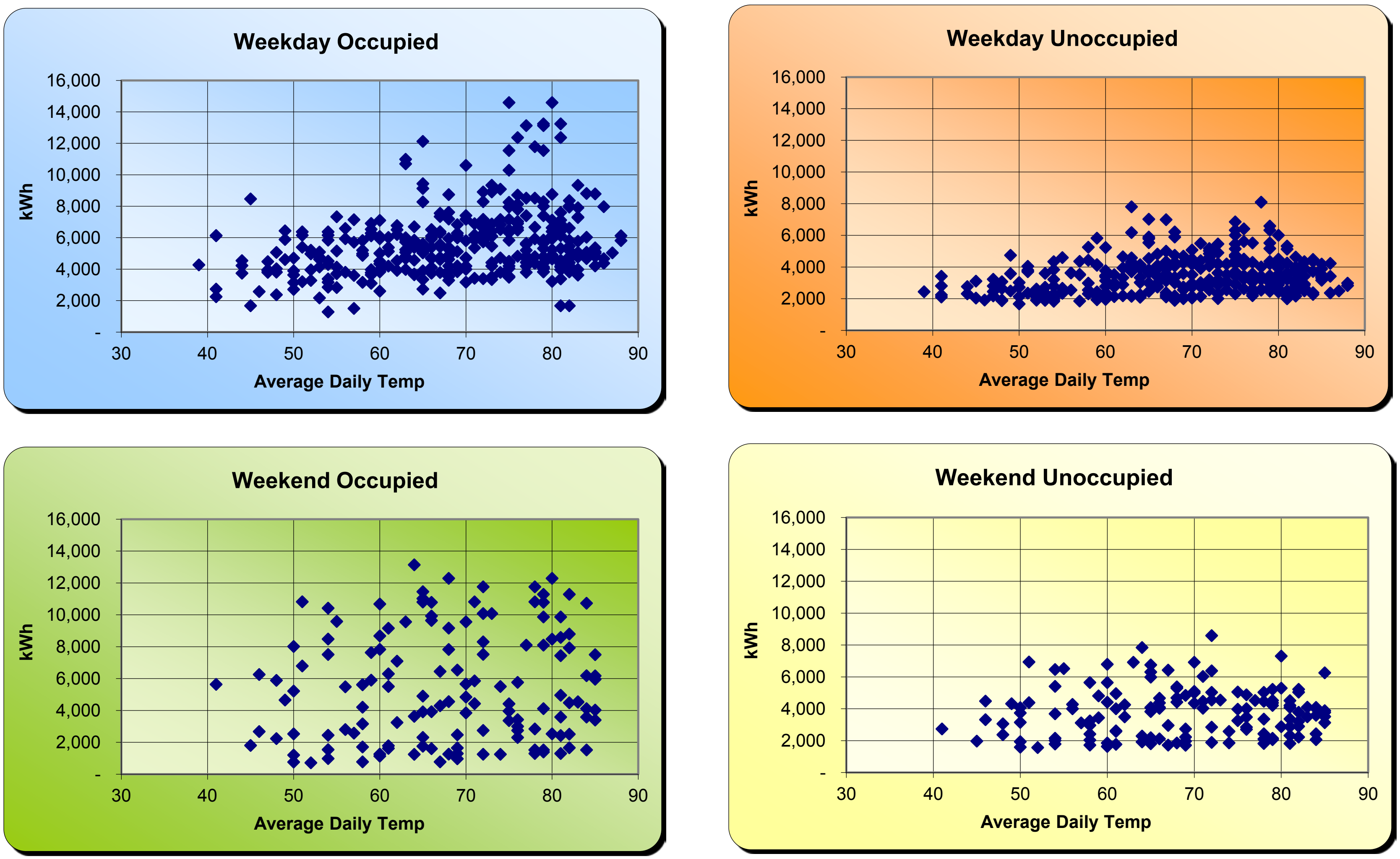


Figure 43: Building 5, Meter 1, energy usage by period type

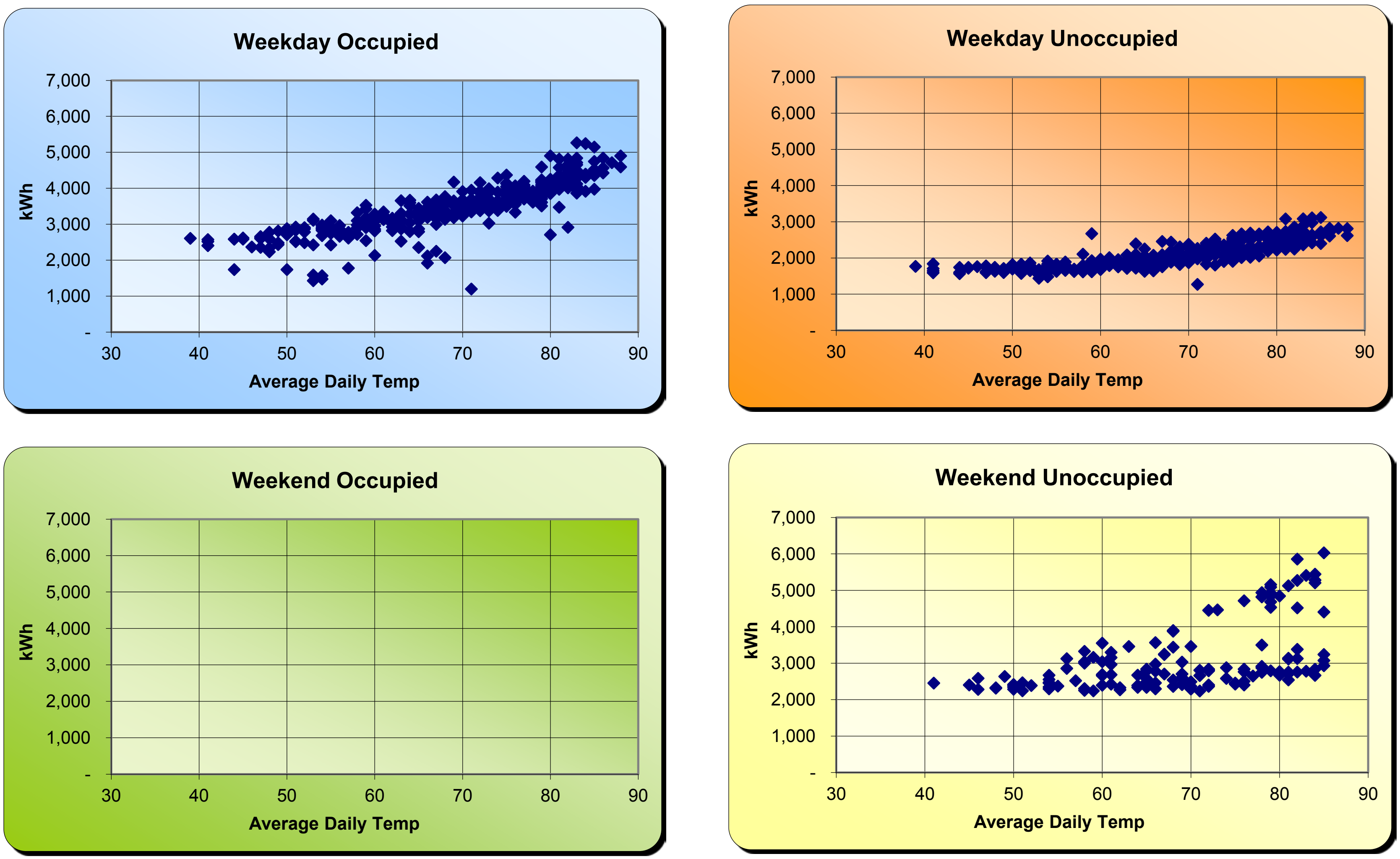
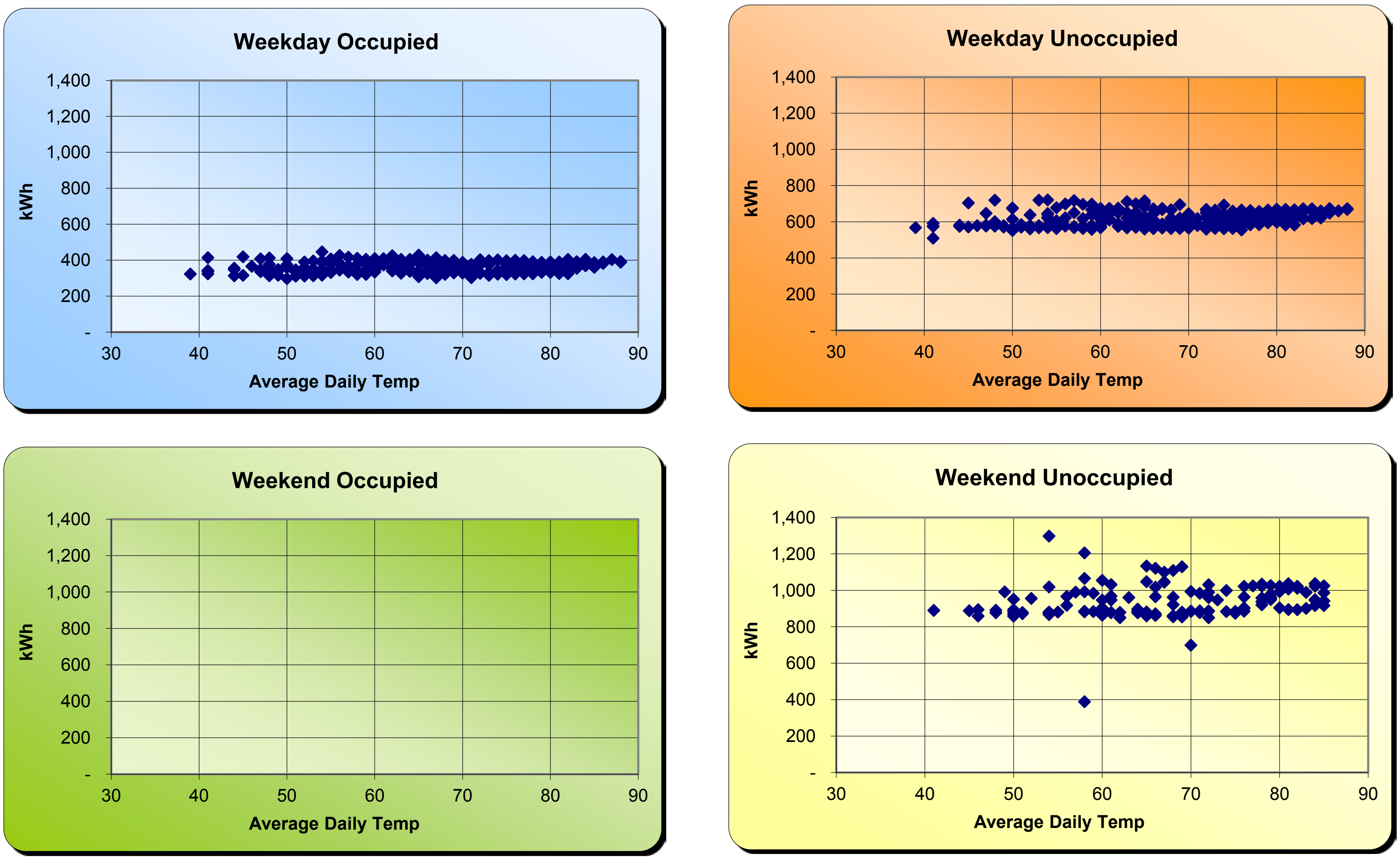


Figure 44: Building 5, Meter 2, energy usage by period type



Appendix B:
VBA for Interval Data Aggregation

```
Sub Step1_SummarizeIntervalData()

Application.ScreenUpdating = False
Application.Calculation = xlCalculationManual

'***Clear Workbook

    Sheets("DailyPeriodStats").Select
    Range("A4:AD503").Select
    Selection.ClearContents

    Sheets("MonthlyPeriodStats").Select
    Range("B4:AD20").Select
    Selection.ClearContents

'***This section loads the interval data into the RawData array

    Dim rowRaw As Long, rowDaily As Long
    Dim row As Long, col As Byte
    Dim StartDate As Single, EndDate As Single
    Dim NumberOfIntervals As Long
    Dim StartTime As Single, EndTime As Single, RunTime As Single

    StartTime = Time
    StartDate = Sheets("inputs").Range("D4")
    EndDate = Sheets("inputs").Range("D5")
    NumberOfIntervals = (EndDate - StartDate) * 96 + 1

    Dim RawData() As Variant
    ReDim RawData(1 To NumberOfIntervals, 1 To 3) As Variant

    For rowRaw = LBound(RawData) To UBound(RawData)
        For col = 1 To 2
            If Sheets("IntervalData").Cells(rowRaw + 5, col) = " " _
                Then RawData(rowRaw, col) = 0 _
                Else RawData(rowRaw, col) = _
                    Sheets("IntervalData").Cells(rowRaw + 5, col)
        Next col
    Next rowRaw

'***This section assigns a period type To the interval. _
    This is for use later on with both the Daily & Monthly Summaries.

    Dim WkDayOcc As Single, WkDayUnocc As Single
    Dim WkEndOcc As Single, WkEndUnocc As Single
    Dim DayType As Byte, OccState As Byte

    Sheets("inputs").Select
```

```
    WkDayOcc = Range("D8").Value        'Case 3 (11)
    WkDayUnocc = Range("E8").Value      'Case 2 (10)
    WkEndOcc = Range("D9").Value        'Case 1 (01)
    WkEndUnocc = Range("E9").Value      'Case 0 (00)

    Sheets("IntervalData").Select
    For rowRaw = LBound(RawData) To UBound(RawData)
        OccState = 0
        'Determines Weekday or Weekend
        If (Int(RawData(rowRaw, 1)) Mod 7) <= 1 _
            Then DayType = 0 _
            Else DayType = 1   'WkEnd=0to1, WkDay=2to6 (Sa=0,...,F=6))
        'Sets Mon 12AM to Weekend
        If RawData(rowRaw, 1) Mod 7 = 2 _
            And RawData(rowRaw, 1) = Int(RawData(rowRaw, 1)) _
            Then DayType = 0
        'Sets Sat 12AM to Weekday
        If RawData(rowRaw, 1) Mod 7 = 0 _
            And RawData(rowRaw, 1) = Int(RawData(rowRaw, 1)) _
            Then DayType = 1

        'Determines Occupied or Unoccupied  (Occ=1, Unocc=0)
        If DayType = 0 _
            And ((RawData(rowRaw, 1) - Int(RawData(rowRaw, 1))) > _
                WkEndOcc) _
            And ((RawData(rowRaw, 1) - Int(RawData(rowRaw, 1))) <= _
                WkEndUnocc) _
            Then OccState = 1
        If DayType = 1 _
            And ((RawData(rowRaw, 1) - Int(RawData(rowRaw, 1))) > _
                WkDayOcc) _
            And ((RawData(rowRaw, 1) - Int(RawData(rowRaw, 1))) <= _
                WkDayUnocc) _
            Then OccState = 1
        'Calculates Period Type
        RawData(rowRaw, 3) = DayType * 2 + OccState * 1
    Next rowRaw

'***This calculates the basic DAILY stats For the 4 period types
    'They are Count, Sum, Average, Min, Max, StDev

    Dim NumberOfDays As Integer
    'Weekday Occupied
    Dim WdOccCount As Single, WdOccSum As Single
    Dim WdOccAvg As Single, WdOccStDev As Single
    Dim WdOccMin As Single, WdOccMax As Single
    Dim WdOccSumSqr As Single
    'Weekday Unoccupied
    Dim WdUnoccCount As Single, WdUnoccSum As Single
    Dim WdUnoccAvg As Single, WdUnoccStDev As Single
    Dim WdUnoccMin As Single, WdUnoccMax As Single
    Dim WdUnoccSumSqr As Single
    'Weekend Occupied
    Dim WeOccCount As Single, WeOccSum As Single
    Dim WeOccAvg As Single, WeOccStDev As Single
    Dim WeOccMin As Single, WeOccMax As Single
    Dim WeOccSumSqr As Single
```

```
'Weekend Unoccupied
Dim WeUnoccCount As Single, WeUnoccSum As Single
Dim WeUnoccAvg As Single, WeUnoccStDev As Single
Dim WeUnoccMin As Single, WeUnoccMax As Single
Dim WeUnoccSumSqr As Single

Dim junk As Single
Dim Counter As Integer

NumberOfDays = EndDate - StartDate

Dim DailyData() As Variant
ReDim DailyData(1 To NumberOfDays, 1 To 30) As Variant

'Loads the dates into column 1 of the DailyData array
For rowDaily = LBound(DailyData) To UBound(DailyData)
    DailyData(rowDaily, 1) = StartDate + (rowDaily - 1)
    DailyData(rowDaily, 2) = _
        Sheets("inputs").Cells(rowDaily + 3, 13)
Next rowDaily

For rowDaily = LBound(DailyData) To UBound(DailyData)

    'Reset variables for next day
    WdOccCount = 0
    WdOccSum = 0
    WdOccMin = 1000000
    WdOccMax = 0
    WdOccSumSqr = 0
    WdOccAvg = 0
    WdOccStDev = 0

    WdUnoccCount = 0
    WdUnoccSum = 0
    WdUnoccMin = 1000000
    WdUnoccMax = 0
    WdUnoccSumSqr = 0
    WdUnoccAvg = 0
    WdUnoccStDev = 0

    WeOccCount = 0
    WeOccSum = 0
    WeOccMin = 1000000
    WeOccMax = 0
    WeOccSumSqr = 0
    WeOccAvg = 0
    WeOccStDev = 0

    WeUnoccCount = 0
    WeUnoccSum = 0
    WeUnoccMin = 1000000
    WeUnoccMax = 0
    WeUnoccSumSqr = 0
    WeUnoccAvg = 0
    WeUnoccStDev = 0

    For rowRaw = (((rowDaily - 1) * 96) + 2) To _
```

```
    (((rowDaily - 1) * 96) + 97)

Select Case RawData(rowRaw, 2)
    Case Is > 0
        Select Case RawData(rowRaw, 3)
            Case 3          'Weekday Occupied
                WdOccCount = WdOccCount + 1
                WdOccSum = WdOccSum + RawData(rowRaw, 2)
                If RawData(rowRaw, 2) < WdOccMin Then _
                    WdOccMin = RawData(rowRaw, 2)
                If RawData(rowRaw, 2) > WdOccMax Then _
                    WdOccMax = RawData(rowRaw, 2)
                WdOccSumSqr = _
                    WdOccSumSqr + RawData(rowRaw, 2) ^ 2
            Case 2          'Weekday Unoccupied
                WdUnoccCount = WdUnoccCount + 1
                WdUnoccSum = WdUnoccSum + RawData(rowRaw, 2)
                If RawData(rowRaw, 2) < WdUnoccMin Then _
                    WdUnoccMin = RawData(rowRaw, 2)
                If RawData(rowRaw, 2) > WdUnoccMax Then _
                    WdUnoccMax = RawData(rowRaw, 2)
                WdUnoccSumSqr = _
                    WdUnoccSumSqr + RawData(rowRaw, 2) ^ 2
            Case 1          'Weekend Occupied
                WeOccCount = WeOccCount + 1
                WeOccSum = WeOccSum + RawData(rowRaw, 2)
                If RawData(rowRaw, 2) < WeOccMin Then _
                    WeOccMin = RawData(rowRaw, 2)
                If RawData(rowRaw, 2) > WeOccMax Then _
                    WeOccMax = RawData(rowRaw, 2)
                WeOccSumSqr = _
                    WeOccSumSqr + RawData(rowRaw, 2) ^ 2
            Case 0          'Weekend Unoccupied
                WeUnoccCount = WeUnoccCount + 1
                WeUnoccSum = WeUnoccSum + RawData(rowRaw, 2)
                If RawData(rowRaw, 2) < WeUnoccMin Then _
                    WeUnoccMin = RawData(rowRaw, 2)
                If RawData(rowRaw, 2) > WeUnoccMax Then _
                    WeUnoccMax = RawData(rowRaw, 2)
                WeUnoccSumSqr = _
                    WeUnoccSumSqr + RawData(rowRaw, 2) ^ 2
        End Select
End Select

Next rowRaw

'Weekday Occupied
If WdOccCount > 0 Then _
    WdOccAvg = WdOccSum / WdOccCount
If WdOccSumSqr <= WdOccCount * WdOccAvg ^ 2 Then _
    WdOccSumSqr = WdOccSumSqr + _
    (WdOccSumSqr - WdOccCount * WdOccAvg ^ 2) * -1 + 1
If WdOccCount > 1 Then _
    WdOccStDev = Sqr((WdOccSumSqr - WdOccCount _
    * WdOccAvg ^ 2) / (WdOccCount - 1))
If WdOccCount > 0 Then DailyData(rowDaily, 3) = "Yes"
If WdOccCount > 0 Then DailyData(rowDaily, 4) = WdOccCount
```

```
If WdOccSum > 0 Then DailyData(rowDaily, 5) = WdOccSum
If WdOccAvg > 0 Then DailyData(rowDaily, 6) = WdOccAvg
If WdOccMin < 1000000 Then _
    DailyData(rowDaily, 7) = WdOccMin
If WdOccMax > 0 Then DailyData(rowDaily, 8) = WdOccMax
If WdOccStDev > 0 Then DailyData(rowDaily, 9) = WdOccStDev

'Weekday Unoccupied
If WdUnoccCount > 0 Then _
    WdUnoccAvg = WdUnoccSum / WdUnoccCount
If WdUnoccSumSqr <= WdUnoccCount * WdUnoccAvg ^ 2 Then _
    WdUnoccSumSqr = WdUnoccSumSqr + _
    (WdUnoccSumSqr - WdUnoccCount * WdUnoccAvg ^ 2) * -1 + 1
If WdUnoccCount > 1 Then _
    WdUnoccStDev = Sqr((WdUnoccSumSqr - WdUnoccCount _
    * WdUnoccAvg ^ 2) / (WdUnoccCount - 1))
If WdUnoccCount > 0 Then DailyData(rowDaily, 10) = "Yes"
If WdUnoccCount > 0 Then DailyData(rowDaily, 11) = WdUnoccCount
If WdUnoccSum > 0 Then DailyData(rowDaily, 12) = WdUnoccSum
If WdUnoccAvg > 0 Then DailyData(rowDaily, 13) = WdUnoccAvg
If WdUnoccMin < 1000000 Then _
    DailyData(rowDaily, 14) = WdUnoccMin
If WdUnoccMax > 0 Then DailyData(rowDaily, 15) = WdUnoccMax
If WdUnoccStDev > 0 Then DailyData(rowDaily, 16) = WdUnoccStDev

'Weekend Occupied
If WeOccCount > 0 Then _
    WeOccAvg = WeOccSum / WeOccCount
If WeOccSumSqr <= WeOccCount * WeOccAvg ^ 2 Then _
    WeOccSumSqr = WeOccSumSqr + _
    (WeOccSumSqr - WeOccCount * WeOccAvg ^ 2) * -1 + 1
If WeOccCount > 1 Then _
    WeOccStDev = Sqr((WeOccSumSqr - WeOccCount _
    * WeOccAvg ^ 2) / (WeOccCount - 1))
If WeOccCount > 0 Then DailyData(rowDaily, 17) = "Yes"
If WeOccCount > 0 Then DailyData(rowDaily, 18) = WeOccCount
If WeOccSum > 0 Then DailyData(rowDaily, 19) = WeOccSum
If WeOccAvg > 0 Then DailyData(rowDaily, 20) = WeOccAvg
If WeOccMin < 1000000 Then _
    DailyData(rowDaily, 21) = WeOccMin
If WeOccMax > 0 Then DailyData(rowDaily, 22) = WeOccMax
If WeOccStDev > 0 Then DailyData(rowDaily, 23) = WeOccStDev

'Weekend Unoccupied
If WeUnoccCount > 0 Then _
    WeUnoccAvg = WeUnoccSum / WeUnoccCount
If WeUnoccSumSqr <= WeUnoccCount * WeUnoccAvg ^ 2 Then _
    WeUnoccSumSqr = WeUnoccSumSqr + _
    (WeUnoccSumSqr - WeUnoccCount * WeUnoccAvg ^ 2) * -1 + 1
If WeUnoccCount > 1 Then _
    WeUnoccStDev = Sqr((WeUnoccSumSqr - WeUnoccCount _
    * WeUnoccAvg ^ 2) / (WeUnoccCount - 1))
If WeUnoccCount > 0 Then DailyData(rowDaily, 24) = "Yes"
If WeUnoccCount > 0 Then DailyData(rowDaily, 25) = WeUnoccCount
If WeUnoccSum > 0 Then DailyData(rowDaily, 26) = WeUnoccSum
If WeUnoccAvg > 0 Then DailyData(rowDaily, 27) = WeUnoccAvg
If WeUnoccMin < 1000000 Then _
```

```
            DailyData(rowDaily, 28) = WeUnoccMin
        If WeUnoccMax > 0 Then DailyData(rowDaily, 29) = WeUnoccMax
        If WeUnoccStDev > 0 Then DailyData(rowDaily, 30) = WeUnoccStDev

    Next rowDaily

'***Copies DailyData array back to workbook

    Dim SumString As String
    Sheets("DailyPeriodStats").Select

    For row = 1 To NumberOfDays

        For col = 1 To 4
            Cells(row + 3, col).Value = DailyData(row, col)
        Next col

        'col=5
        If DailyData(row, col - 1) > 0 Then _
            Cells(row + 3, col) = _
            "=IF(C" & (row + 3) & "=""Yes""," & _
            DailyData(row, col) & ","""")"

        For col = 6 To 11
            Cells(row + 3, col).Value = DailyData(row, col)
        Next col

        'col=12
        If DailyData(row, col - 1) > 0 Then _
            Cells(row + 3, col) = _
            "=IF(J" & (row + 3) & "=""Yes""," & _
            DailyData(row, col) & ","""")"

        For col = 13 To 18
            Cells(row + 3, col).Value = DailyData(row, col)
        Next col

        'col=19
        If DailyData(row, col - 1) > 0 Then _
            Cells(row + 3, col) = _
            "=IF(Q" & (row + 3) & "=""Yes""," & _
            DailyData(row, col) & ","""")"

        For col = 20 To 25
            Cells(row + 3, col).Value = DailyData(row, col)
        Next col

        'col=26
        If DailyData(row, col - 1) > 0 Then _
            Cells(row + 3, col) = _
            "=IF(X" & (row + 3) & "=""Yes""," & _
            DailyData(row, col) & ","""")"

        For col = 27 To 30
            Cells(row + 3, col).Value = DailyData(row, col)
        Next col
```

```
    Next row

'***This calculates the basic MONTHLY stats For the 4 period types
    Dim MonthlyData() As Variant
    ReDim MonthlyData(1 To 16, 1 To 31) As Variant
    Dim rowMonthly As Byte
    Dim NumDaysInMonth(1 To 16) As Integer
    Dim StartRow As Long, EndRow As Long
    Dim MonthlyTempSum As Integer

    StartRow = 0
    EndRow = 1

    'Loads Months into MonthlyData array
    Sheets("MonthlyPeriodStats").Select
    For rowMonthly = 1 To 16
        MonthlyData(rowMonthly, 1) = Cells(rowMonthly + 3, 1)
        NumDaysInMonth(rowMonthly) = _
            Cells(rowMonthly + 4, 1) - Cells(rowMonthly + 3, 1)

        'Calculate the monthly average temp
        MonthlyTempSum = 0
        For i = 1 To NumberOfDays
            If (Month(DailyData(i, 1)) = _
                Month(MonthlyData(rowMonthly, 1)) _
                And Year(DailyData(i, 1)) = _
                Year(MonthlyData(rowMonthly, 1))) _
                Then MonthlyTempSum = MonthlyTempSum + DailyData(i, 2)
        Next i
        MonthlyData(rowMonthly, 2) = _
            Round(MonthlyTempSum / NumDaysInMonth(rowMonthly), 1)
    Next rowMonthly

    For rowMonthly = 1 To 16

        'Resets variables For the Next month
        WdOccCount = 0
        WdOccSum = 0
        WdOccMin = 1000000
        WdOccMax = 0
        WdOccSumSqr = 0
        WdOccAvg = 0
        WdOccStDev = 0

        WdUnoccCount = 0
        WdUnoccSum = 0
        WdUnoccMin = 1000000
        WdUnoccMax = 0
        WdUnoccSumSqr = 0
        WdUnoccAvg = 0
        WdUnoccStDev = 0

        WeOccCount = 0
        WeOccSum = 0
        WeOccMin = 1000000
        WeOccMax = 0
        WeOccSumSqr = 0
```

```
WeOccAvg = 0
WeOccStDev = 0

WeUnoccCount = 0
WeUnoccSum = 0
WeUnoccMin = 1000000
WeUnoccMax = 0
WeUnoccSumSqr = 0
WeUnoccAvg = 0
WeUnoccStDev = 0

StartRow = EndRow + 1
EndRow = StartRow + (NumDaysInMonth(rowMonthly) * 96 - 1)

For rowRaw = StartRow To EndRow

Select Case RawData(rowRaw, 2)
    Case Is > 0
        Select Case RawData(rowRaw, 3)
            Case 3          'Weekday Occupied
                WdOccCount = WdOccCount + 1
                WdOccSum = WdOccSum + RawData(rowRaw, 2)
                If RawData(rowRaw, 2) < WdOccMin Then _
                    WdOccMin = RawData(rowRaw, 2)
                If RawData(rowRaw, 2) > WdOccMax Then _
                    WdOccMax = RawData(rowRaw, 2)
                WdOccSumSqr = _
                    WdOccSumSqr + RawData(rowRaw, 2) ^ 2
            Case 2          'Weekday Unoccupied
                WdUnoccCount = WdUnoccCount + 1
                WdUnoccSum = WdUnoccSum + RawData(rowRaw, 2)
                If RawData(rowRaw, 2) < WdUnoccMin Then _
                    WdUnoccMin = RawData(rowRaw, 2)
                If RawData(rowRaw, 2) > WdUnoccMax Then _
                    WdUnoccMax = RawData(rowRaw, 2)
                WdUnoccSumSqr = _
                    WdUnoccSumSqr + RawData(rowRaw, 2) ^ 2
            Case 1          'Weekend Occupied
                WeOccCount = WeOccCount + 1
                WeOccSum = WeOccSum + RawData(rowRaw, 2)
                If RawData(rowRaw, 2) < WeOccMin Then _
                    WeOccMin = RawData(rowRaw, 2)
                If RawData(rowRaw, 2) > WeOccMax Then _
                    WeOccMax = RawData(rowRaw, 2)
                WeOccSumSqr = _
                    WeOccSumSqr + RawData(rowRaw, 2) ^ 2
            Case 0          'Weekend Unoccupied
                WeUnoccCount = WeUnoccCount + 1
                WeUnoccSum = WeUnoccSum + RawData(rowRaw, 2)
                If RawData(rowRaw, 2) < WeUnoccMin Then _
                    WeUnoccMin = RawData(rowRaw, 2)
                If RawData(rowRaw, 2) > WeUnoccMax Then _
                    WeUnoccMax = RawData(rowRaw, 2)
                WeUnoccSumSqr = _
                    WeUnoccSumSqr + RawData(rowRaw, 2) ^ 2
        End Select
End Select
```

```
Next rowRaw

'Weekday Occupied
If WdOccCount > 0 Then _
    WdOccAvg = WdOccSum / WdOccCount
If WdOccSumSqr <= WdOccCount * WdOccAvg ^ 2 Then _
    WdOccSumSqr = WdOccSumSqr + _
    (WdOccSumSqr - WdOccCount * WdOccAvg ^ 2) * -1 + 1
If WdOccCount > 1 Then _
    WdOccStDev = Sqr((WdOccSumSqr - WdOccCount _
    * WdOccAvg ^ 2) / (WdOccCount - 1))
If WdOccCount > 0 Then MonthlyData(rowMonthly, 3) = "Yes"
If WdOccCount > 0 Then _
    MonthlyData(rowMonthly, 4) = WdOccCount
If WdOccSum > 0 Then MonthlyData(rowMonthly, 5) = WdOccSum
If WdOccAvg > 0 Then MonthlyData(rowMonthly, 6) = WdOccAvg
If WdOccMin < 1000000 Then _
    MonthlyData(rowMonthly, 7) = WdOccMin
If WdOccMax > 0 Then MonthlyData(rowMonthly, 8) = WdOccMax
If WdOccStDev > 0 Then _
    MonthlyData(rowMonthly, 9) = WdOccStDev

'Weekday Unoccupied
If WdUnoccCount > 0 Then _
    WdUnoccAvg = WdUnoccSum / WdUnoccCount
If WdUnoccSumSqr <= WdUnoccCount * WdUnoccAvg ^ 2 Then _
    WdUnoccSumSqr = WdUnoccSumSqr + _
    (WdUnoccSumSqr - WdUnoccCount * WdUnoccAvg ^ 2) * -1 + 1
If WdUnoccCount > 1 Then _
    WdUnoccStDev = Sqr((WdUnoccSumSqr - WdUnoccCount _
    * WdUnoccAvg ^ 2) / (WdUnoccCount - 1))
If WdUnoccCount > 0 Then MonthlyData(rowMonthly, 10) = "Yes"
If WdUnoccCount > 0 Then _
    MonthlyData(rowMonthly, 11) = WdUnoccCount
If WdUnoccSum > 0 Then MonthlyData(rowMonthly, 12) = WdUnoccSum
If WdUnoccAvg > 0 Then MonthlyData(rowMonthly, 13) = WdUnoccAvg
If WdUnoccMin < 1000000 Then _
    MonthlyData(rowMonthly, 14) = WdUnoccMin
If WdUnoccMax > 0 Then MonthlyData(rowMonthly, 15) = WdUnoccMax
If WdUnoccStDev > 0 Then _
    MonthlyData(rowMonthly, 16) = WdUnoccStDev

'Weekend Occupied
If WeOccCount > 0 Then _
    WeOccAvg = WeOccSum / WeOccCount
If WeOccSumSqr <= WeOccCount * WeOccAvg ^ 2 Then _
    WeOccSumSqr = WeOccSumSqr + _
    (WeOccSumSqr - WeOccCount * WeOccAvg ^ 2) * -1 + 1
If WeOccCount > 1 Then _
    WeOccStDev = Sqr((WeOccSumSqr - WeOccCount _
    * WeOccAvg ^ 2) / (WeOccCount - 1))
If WeOccCount > 0 Then MonthlyData(rowMonthly, 17) = "Yes"
If WeOccCount > 0 Then _
    MonthlyData(rowMonthly, 18) = WeOccCount
If WeOccSum > 0 Then MonthlyData(rowMonthly, 19) = WeOccSum
If WeOccAvg > 0 Then MonthlyData(rowMonthly, 20) = WeOccAvg
```

```
        If WeOccMin < 1000000 Then _
            MonthlyData(rowMonthly, 21) = WeOccMin
        If WeOccMax > 0 Then MonthlyData(rowMonthly, 22) = WeOccMax
        If WeOccStDev > 0 Then _
            MonthlyData(rowMonthly, 23) = WeOccStDev

        'Weekend Unoccupied
        If WeUnoccCount > 0 Then _
            WeUnoccAvg = WeUnoccSum / WeUnoccCount
        If WeUnoccSumSqr <= WeUnoccCount * WeUnoccAvg ^ 2 Then _
            WeUnoccSumSqr = WeUnoccSumSqr + _
            (WeUnoccSumSqr - WeUnoccCount * WeUnoccAvg ^ 2) * -1 + 1
        If WeUnoccCount > 1 Then _
            WeUnoccStDev = Sqr((WeUnoccSumSqr - WeUnoccCount _
            * WeUnoccAvg ^ 2) / (WeUnoccCount - 1))
        If WeUnoccCount > 0 Then MonthlyData(rowMonthly, 24) = "Yes"
        If WeUnoccCount > 0 Then _
            MonthlyData(rowMonthly, 25) = WeUnoccCount
        If WeUnoccSum > 0 Then MonthlyData(rowMonthly, 26) = WeUnoccSum
        If WeUnoccAvg > 0 Then MonthlyData(rowMonthly, 27) = WeUnoccAvg
        If WeUnoccMin < 1000000 Then _
            MonthlyData(rowMonthly, 28) = WeUnoccMin
        If WeUnoccMax > 0 Then MonthlyData(rowMonthly, 29) = WeUnoccMax
        If WeUnoccStDev > 0 Then _
            MonthlyData(rowMonthly, 30) = WeUnoccStDev

    Next rowMonthly

'***Copies MonthlyData array back to workbook

    Sheets("MonthlyPeriodStats").Select

    For row = 1 To 16

        For col = 2 To 4
            Cells(row + 3, col).Value = MonthlyData(row, col)
        Next col

        'col 5
        If MonthlyData(row, col - 1) > 0 Then _
            Cells(row + 3, col) = _
            "=IF(C" & (row + 3) & "=""Yes""," & _
            MonthlyData(row, col) & ","""")"

        For col = 6 To 11
            Cells(row + 3, col).Value = MonthlyData(row, col)
        Next col

        'col 12
        If MonthlyData(row, col - 1) > 0 Then _
            Cells(row + 3, col) = _
            "=IF(J" & (row + 3) & "=""Yes""," & _
            MonthlyData(row, col) & ","""")"

        For col = 13 To 18
            Cells(row + 3, col).Value = MonthlyData(row, col)
        Next col
```

```
        'col 19
        If MonthlyData(row, col - 1) > 0 Then _
            Cells(row + 3, col) = _
            "=IF(Q" & (row + 3) & "=""Yes""," & _
            MonthlyData(row, col) & ","""")"

        For col = 20 To 25
            Cells(row + 3, col).Value = MonthlyData(row, col)
        Next col

        'col26
        If MonthlyData(row, col - 1) > 0 Then _
            Cells(row + 3, col) = _
            "=IF(X" & (row + 3) & "=""Yes""," & _
            MonthlyData(row, col) & ","""")"

        For col = 27 To 30
            Cells(row + 3, col).Value = MonthlyData(row, col)
        Next col

    Next row

'***Wrap up
EndTime = Time
RunTime = EndTime - StartTime
Sheets("inputs").Select
Sheets("inputs").Range("e13").Value = RunTime

Application.ScreenUpdating = True
Application.Calculation = xlCalculationAutomatic

End Sub
```

Appendix C: VBA for Decision Tool

```
Sub Trials()

Application.ScreenUpdating = False
Application.Calculation = xlCalculationManual

Application.StatusBar = "Ok, let's rock this thing..."

Dim StartTime As Single, EndTime As Single, RunTime As Single
StartTime = Time

Dim i As Integer, Trial As Integer, RM As Byte
Dim NumTrials As Integer
Dim WindowLoop As Byte, WindowSize As Byte
Dim TrialNum As Byte, ThresholdLoop As Single

Dim Months()
ReDim Months(1 To 12)
For i = 1 To 12
    Months(i) = Sheets("DailyData").Cells(49 + i, 1)
Next i

Dim Window() As Byte
ReDim Window(1 To 4)
For i = 1 To 4
    Window(i) = Sheets("Inputs").Cells(1 + i, 8)
Next i

Dim Prob() As Single
ReDim Prob(1 To 7)
For i = 1 To 7
    Prob(i) = Sheets("Inputs").Cells(2 + i, 11)
Next i

For ThresholdLoop = 1 To 7

    Sheets("DailyData").Range("B16") = Prob(ThresholdLoop)

    Select Case ThresholdLoop
        Case 1
            Sheets("T1").Select
            Range("B11:IK1010").Select
            Selection.ClearContents
        Case 2
            Sheets("T2").Select
            Range("B11:IK1010").Select
            Selection.ClearContents
        Case 3
            Sheets("T3").Select
            Range("B11:IK1010").Select
```

```
        Selection.ClearContents
    Case 4
        Sheets("T4").Select
        Range("B11:IK1010").Select
        Selection.ClearContents
    Case 5
        Sheets("T5").Select
        Range("B11:IK1010").Select
        Selection.ClearContents
    Case 6
        Sheets("T6").Select
        Range("B11:IK1010").Select
        Selection.ClearContents
    Case 7
        Sheets("T7").Select
        Range("B11:IK1010").Select
        Selection.ClearContents
End Select

NumTrials = Sheets("Inputs").Range("NumberOfTrials")
Dim TrialsArray() As Long
ReDim TrialsArray(1 To NumTrials, 1 To 244) As Long

Rnd (-100871)   'Random number generator resets at beginning
                'of each threshold loop

For Trial = 1 To NumTrials
    StaticRandomNumberGenerator
    For WindowLoop = 1 To 4
        WindowSize = Window(WindowLoop)
        Sheets("DailyData").Range("B4") = WindowSize
            Dim Mo As Byte
            For Mo = 1 To 12
                Sheets("DailyData").Range("B3") = Months(Mo)
                Application.Calculation = xlCalculationAutomatic
                Application.Calculation = xlCalculationManual
                For RM = 1 To 5
                    TrialsArray(Trial, Mo + (WindowLoop - 1) * 12 _
                    + (RM - 1) * 49) _
                        = Sheets("DailyData").Cells(14, RM)
                Next RM
            Next Mo
    Next WindowLoop

EndTime = Time
RunTime = EndTime - StartTime
Application.StatusBar = "Complete: " & _
    Format(((ThresholdLoop - 1) * NumTrials + Trial) / _
    (NumTrials * 7), "0%") & _
    " (Threshold " & ThresholdLoop & ", Trial " & Trial & _
    ") Elapsed Run Time: " & Format(RunTime, "h:mm:ss")

Next Trial

Select Case ThresholdLoop
    Case 1
        Sheets("T1").Range(Cells(11, 2), _
```

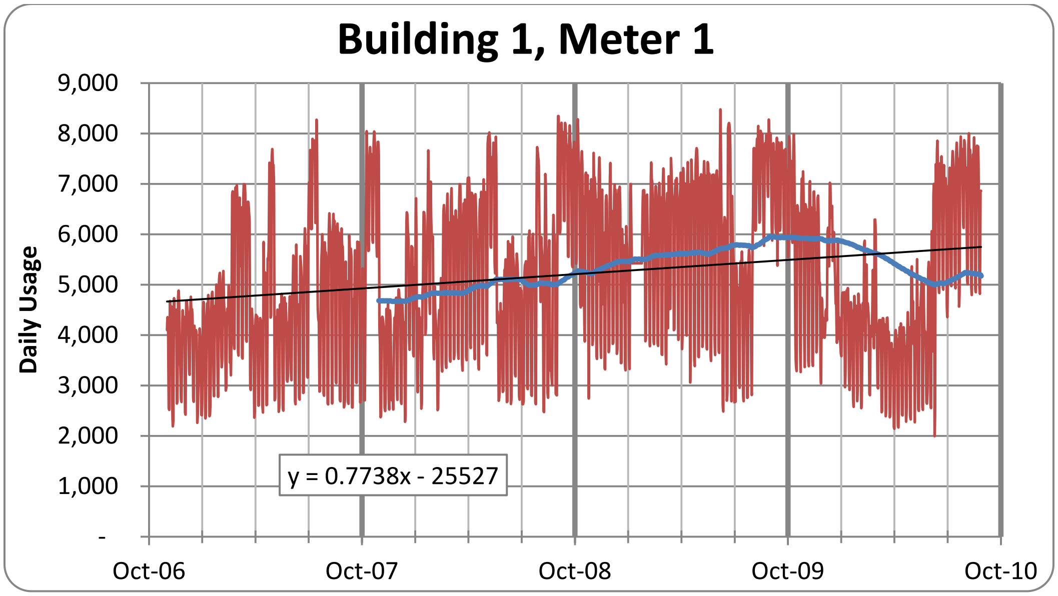
```
                Cells(10 + NumTrials, 245)) = TrialsArray()
            Sheets("T1").Range("B16").Select
        Case 2
            Sheets("T2").Range(Cells(11, 2), _
                Cells(10 + NumTrials, 245)) = TrialsArray()
            Sheets("T2").Range("B16").Select
        Case 3
            Sheets("T3").Range(Cells(11, 2), _
                Cells(10 + NumTrials, 245)) = TrialsArray()
            Sheets("T3").Range("B16").Select
        Case 4
            Sheets("T4").Range(Cells(11, 2), _
                Cells(10 + NumTrials, 245)) = TrialsArray()
            Sheets("T4").Range("B16").Select
        Case 5
            Sheets("T5").Range(Cells(11, 2), _
                Cells(10 + NumTrials, 245)) = TrialsArray()
            Sheets("T5").Range("B16").Select
        Case 6
            Sheets("T6").Range(Cells(11, 2), _
                Cells(10 + NumTrials, 245)) = TrialsArray()
            Sheets("T6").Range("B16").Select
        Case 7
            Sheets("T7").Range(Cells(11, 2), _
                Cells(10 + NumTrials, 245)) = TrialsArray()
            Sheets("T7").Range("B16").Select
    End Select

Next ThresholdLoop

Application.StatusBar = False
Application.ScreenUpdating = True
Application.Calculation = xlCalculationAutomatic
EndTime = Time
RunTime = EndTime - StartTime
Sheets("T1").Range("A9") = RunTime

End Sub
```

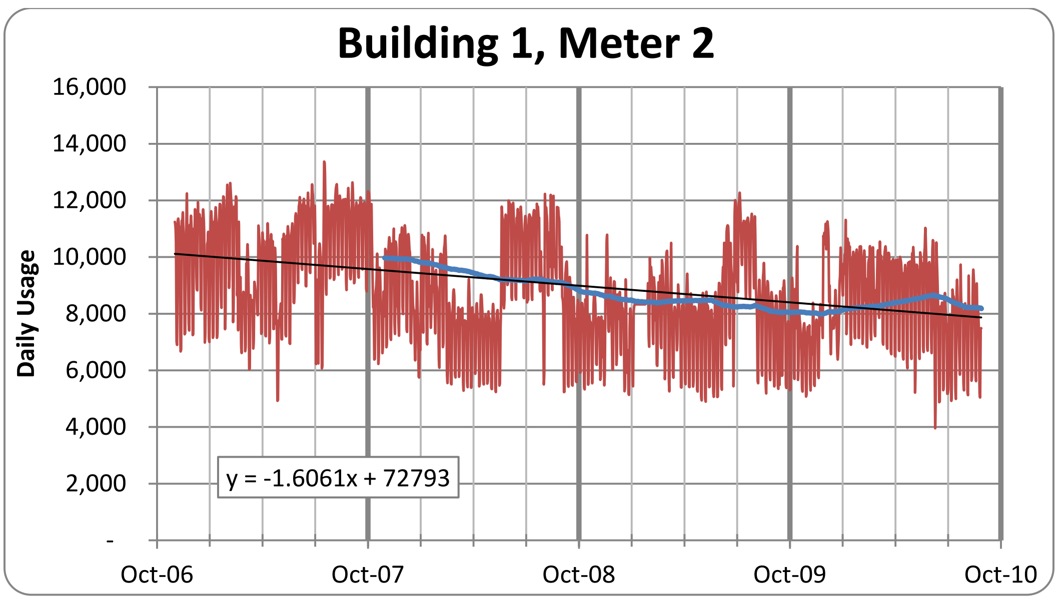
```
Sub StaticRandomNumberGenerator()

Dim Col As Integer, Row As Integer
Dim NumRows As Integer

Dim StartDate As Long, EndDate As Long
StartDate = Sheets("Inputs").Range("FirstTestMo").Value2
EndDate = Sheets("Inputs").Range("LastAvailDate").Value2
NumRows = EndDate - StartDate + 1

Dim RandomArray() As Single
ReDim RandomArray(1 To NumRows, 1 To 32)

'*** Response Length
    For Col = 1 To 32
        For Row = 1 To NumRows
        RandomArray(Row, Col) = Rnd()
```

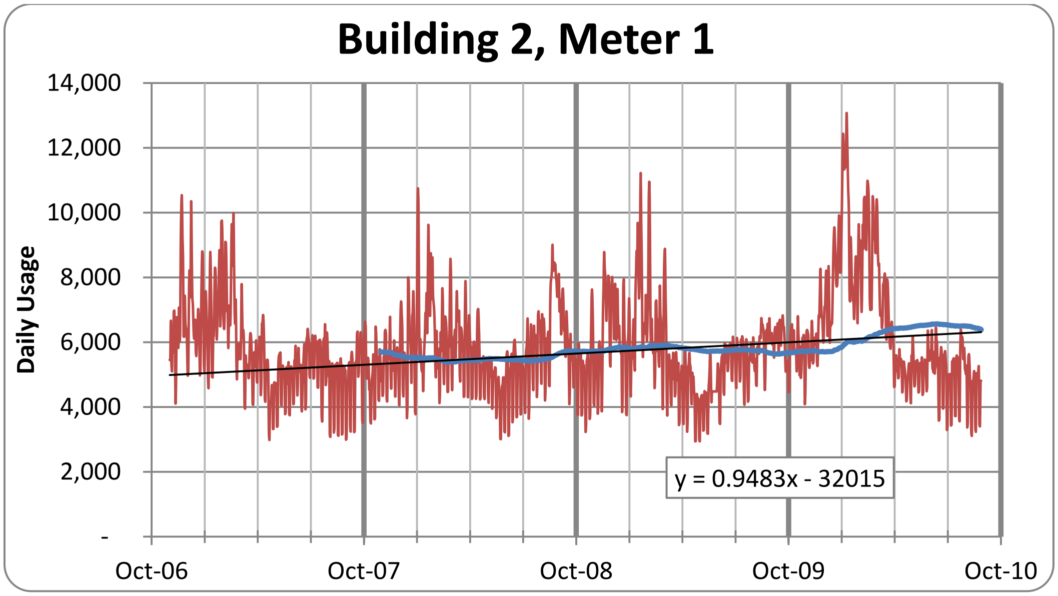
```
        Next Row
    Next Col
    Sheets("DailyData").Range("ProbResponseLength") = RandomArray()

'*** Response Type
    For Col = 1 To 32
        For Row = 1 To NumRows
        RandomArray(Row, Col) = Rnd()
        Next Row
    Next Col
    Sheets("DailyData").Range("ProbResponseType") = RandomArray()

'*** Baseline Adjust Multiplier
    For Col = 1 To 32
        For Row = 1 To NumRows
        RandomArray(Row, Col) = Rnd()
        Next Row
    Next Col
    Sheets("DailyData").Range("ProbBaseAdjMult") = RandomArray()

End Sub
```

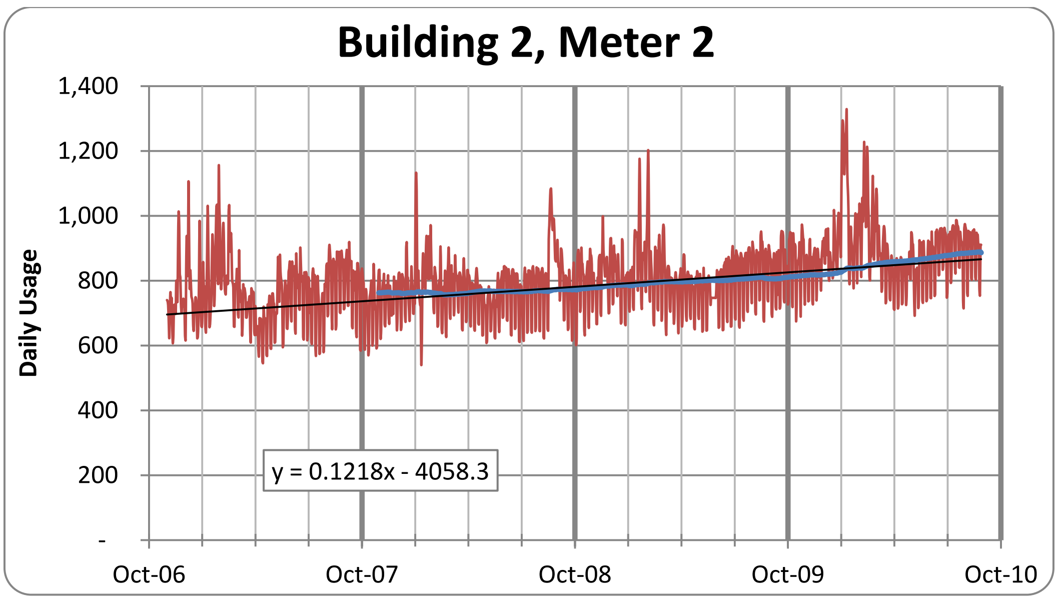
```
Sub GenerateDynamicRandomNumbers()

Application.ScreenUpdating = False
Application.Calculation = xlCalculationManual

Range("ProbResponseLength").Formula = "=RAND()"
Range("ProbResponseType").Formula = "=RAND()"
Range("ProbBaseAdjMult").Formula = "=RAND()"

Application.ScreenUpdating = True
Application.Calculation = xlCalculationAutomatic

End Sub
```

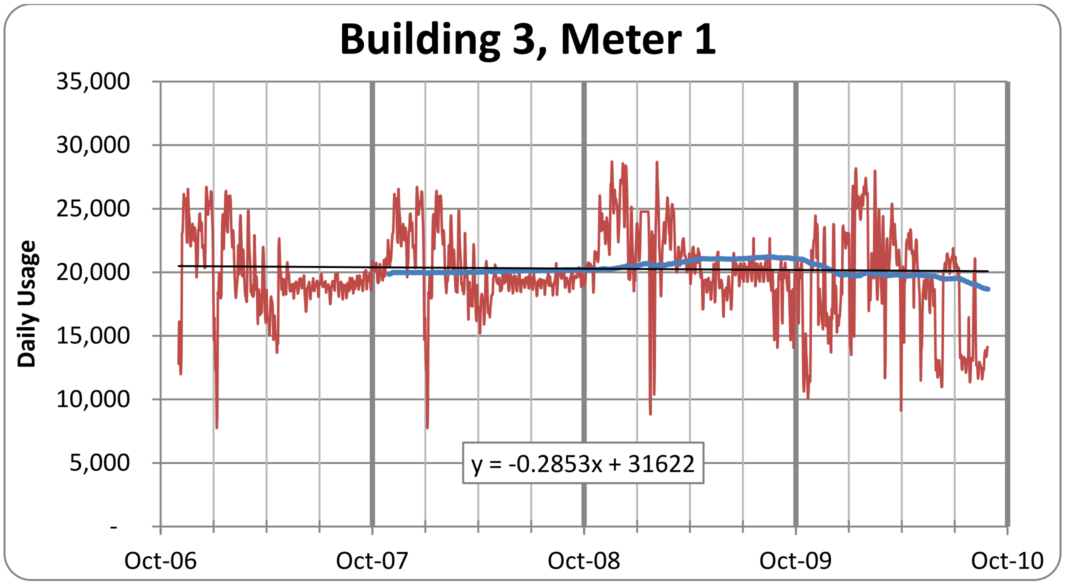
Appendix D:
Building Daily Usage and TTM

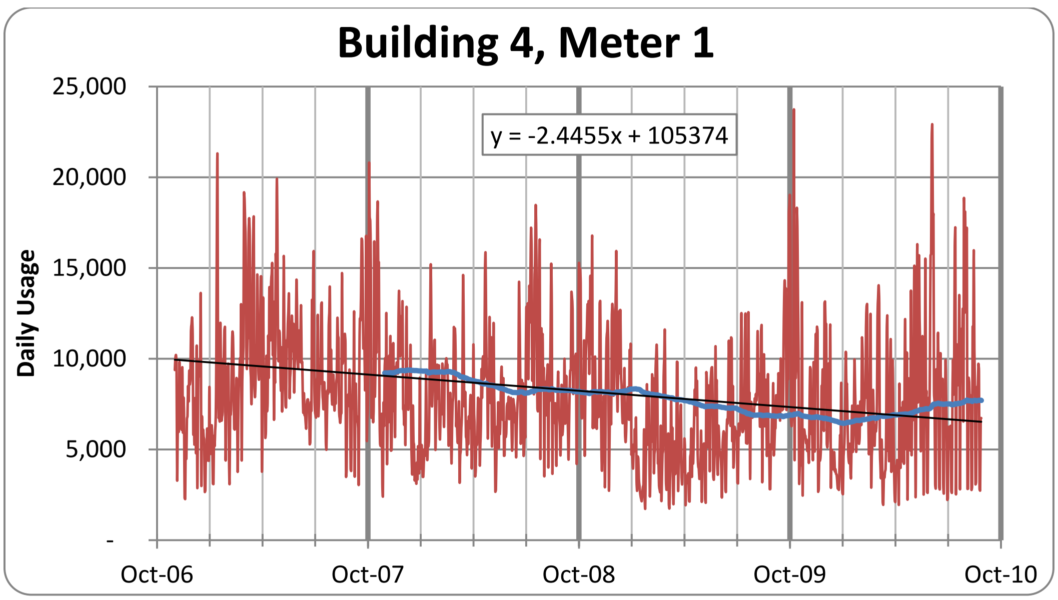


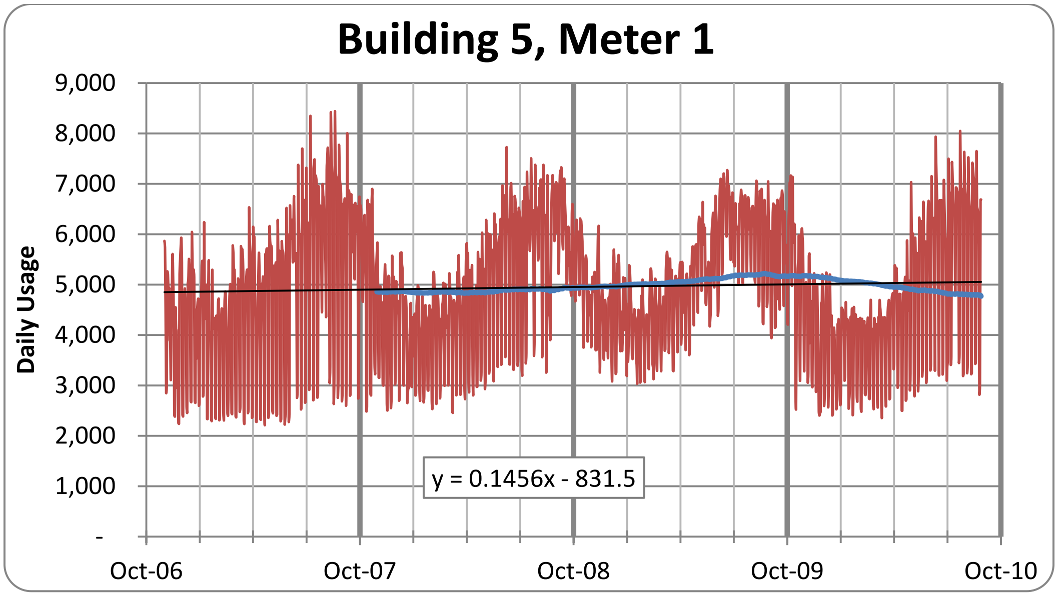


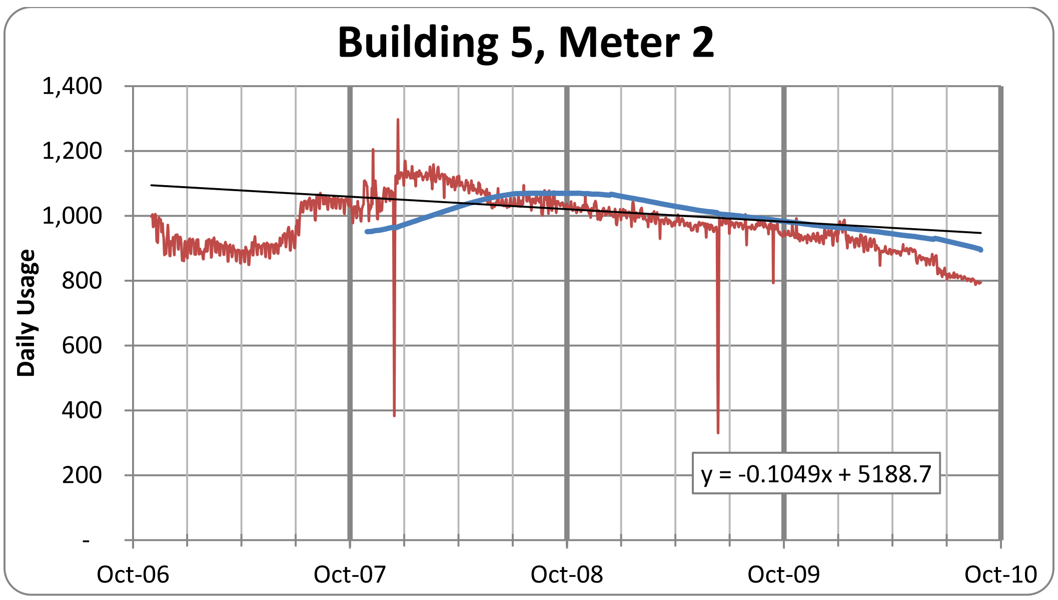

Building 2, Meter 1
Daily Usage
14,000
12,000
10,000
8,000
6,000
4,000
2,000
-
Oct-06
Oct-07
Oct-08
Oct-09
Oct-10
y = 0.9483x - 32015


Building 2, Meter 2
Daily Usage
1,400
1,200
1,000
800
600
400
200
-
Oct-06
Oct-07
Oct-08
Oct-09
Oct-10
y = 0.1218x - 4058.3


Building 3, Meter 1
35,000
30,000
25,000
20,000
15,000
10,000
5,000
-
Daily Usage
y = -0.2853x + 31622
Oct-06
Oct-07
Oct-08
Oct-09
Oct-10


Building 4, Meter 1
25,000
20,000
15,000
10,000
5,000
-
Daily Usage
y = -2.4455x + 105374
Oct-06
Oct-07
Oct-08
Oct-09
Oct-10


Building 5, Meter 1
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
-
Daily Usage
y = 0.1456x - 831.5
Oct-06
Oct-07
Oct-08
Oct-09
Oct-10


Building 5, Meter 2
1,400
1,200
1,000
800
600
400
200
-
Daily Usage
y = -0.1049x + 5188.7
Oct-06
Oct-07
Oct-08
Oct-09
Oct-10

Appendix E:
Normal Probability Plots

Figure 45: Building 1, Meter 1, normal probability plots by period type

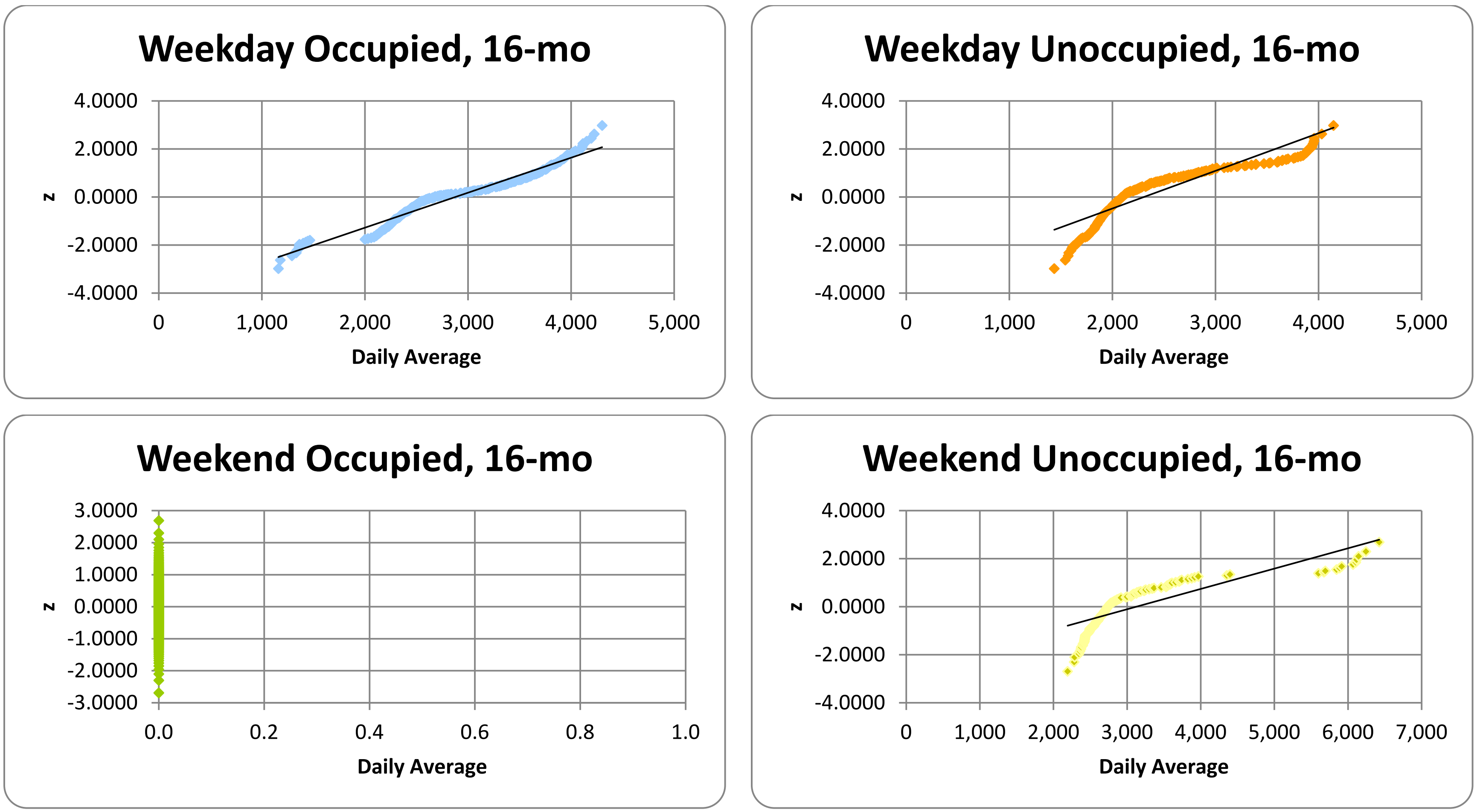


Figure 46: Building 1, Meter 2, normal probability plots by period type

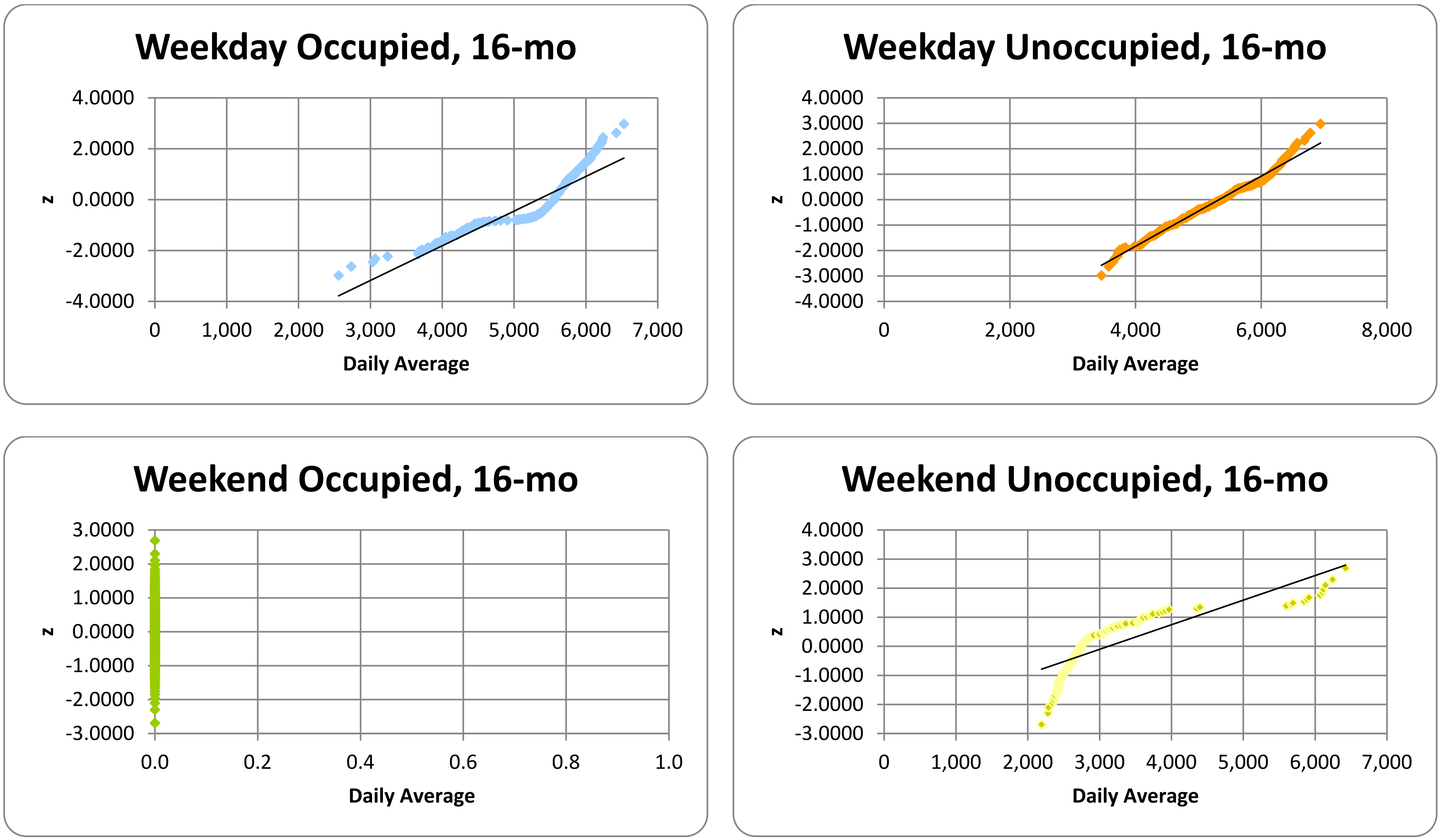


Figure 47: Building 2, Meter 1, normal probability plots by period type

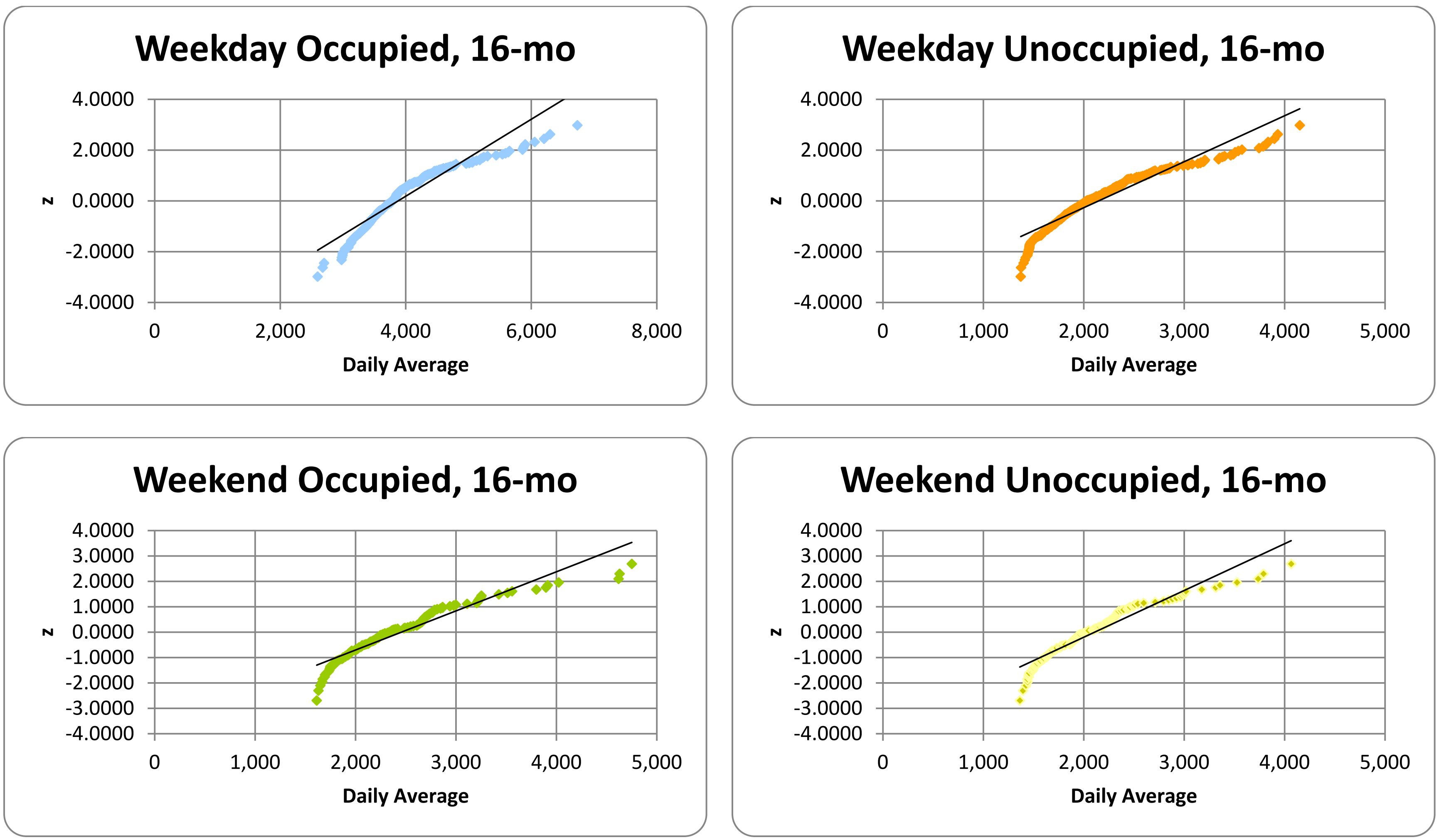


Figure 48: Building 2, Meter 2, normal probability plots by period type

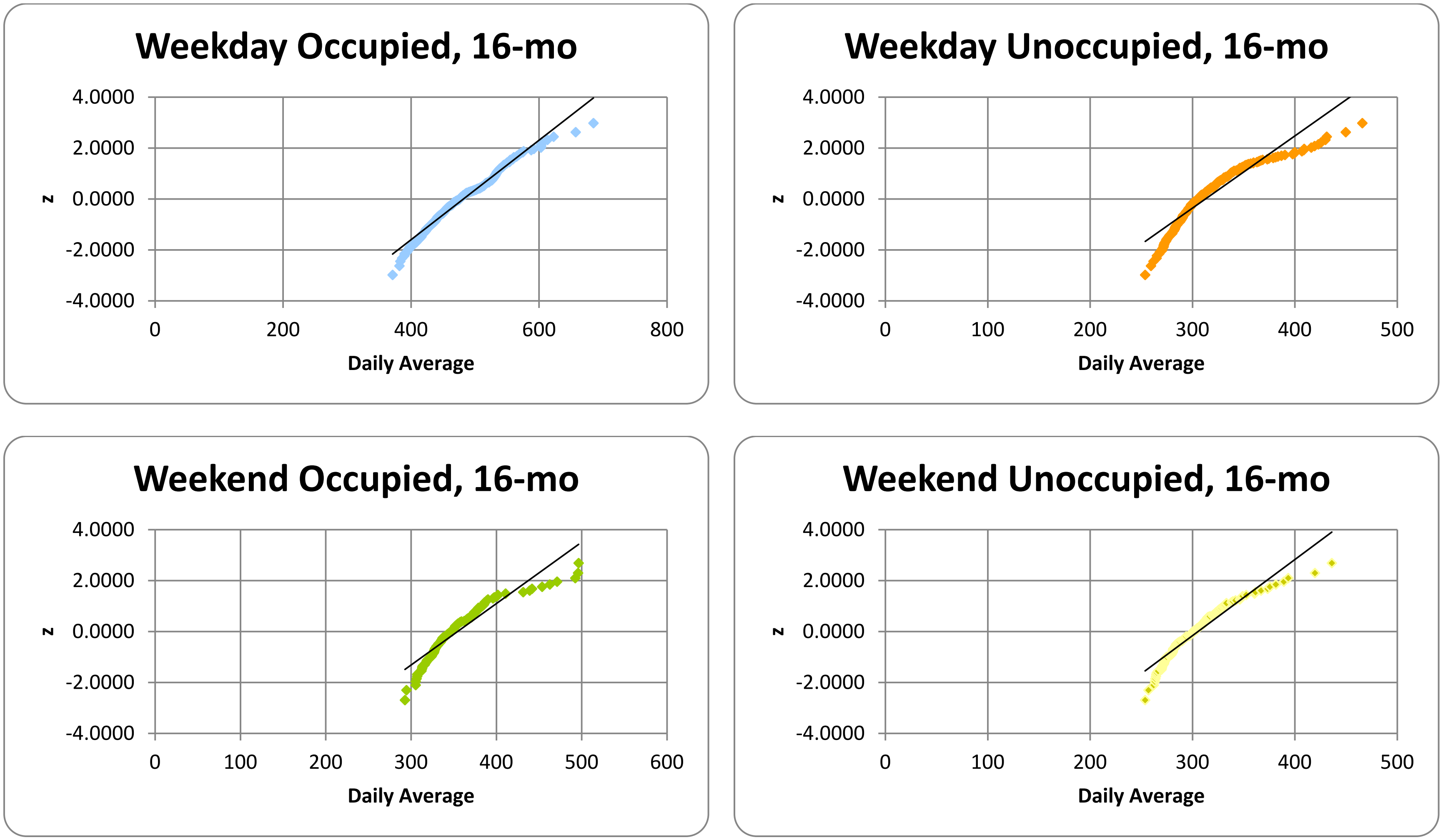


Figure 49: Building 3, Meter 1, normal probability plots by period type

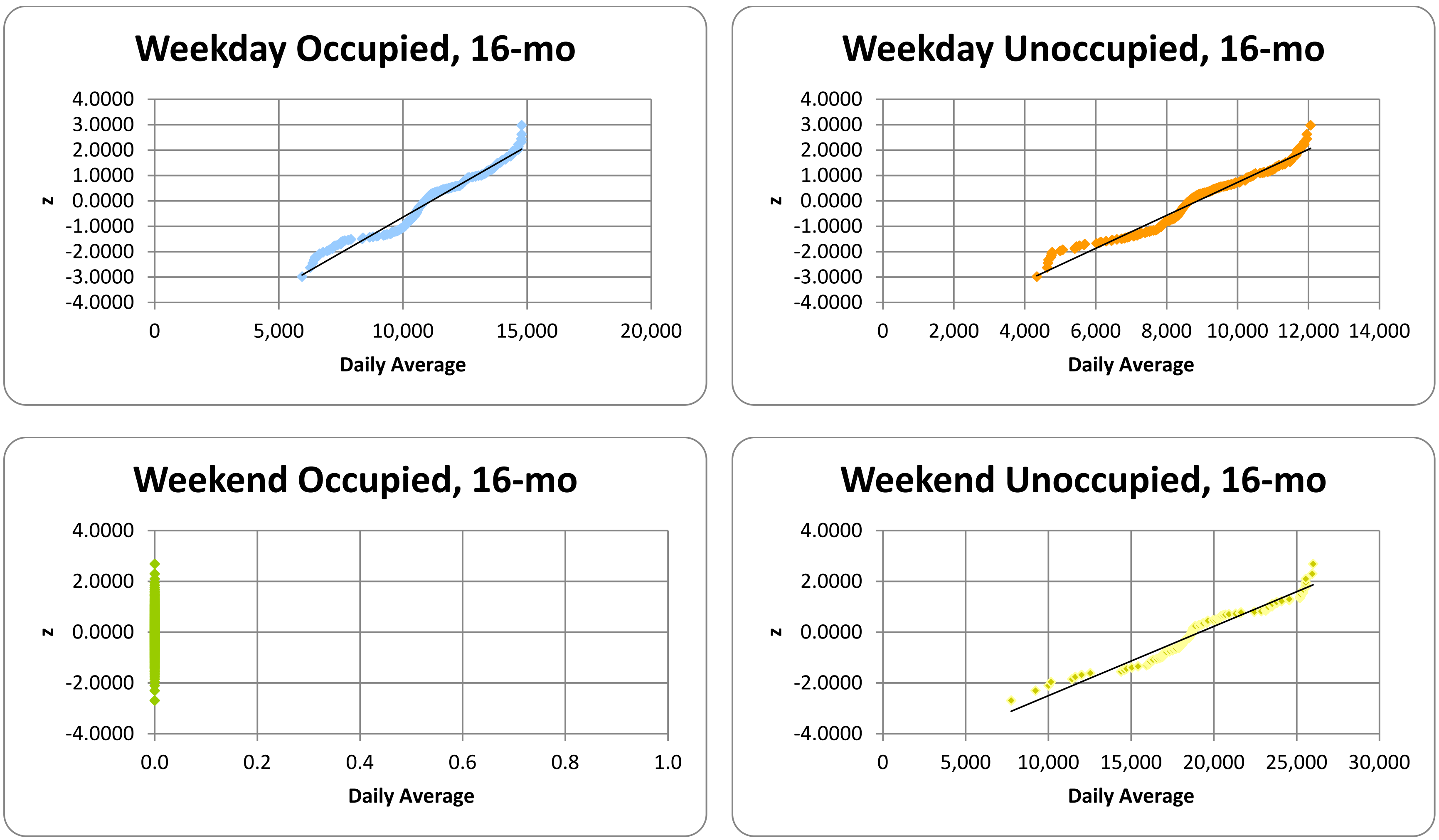


Figure 50: Building 4, Meter 1, normal probability plots by period type

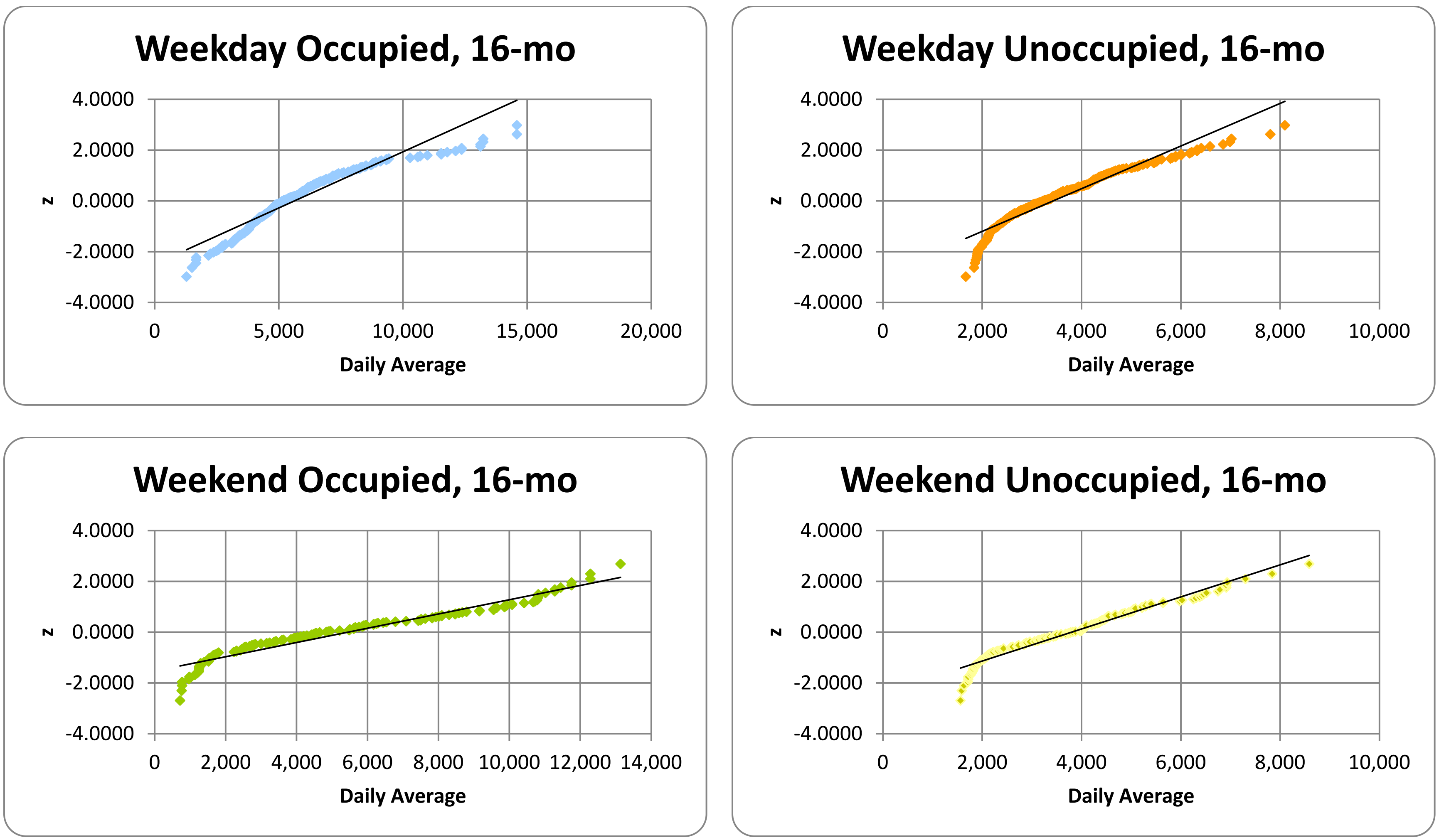


Figure 51: Building 5, Meter 1, normal probability plots by period type

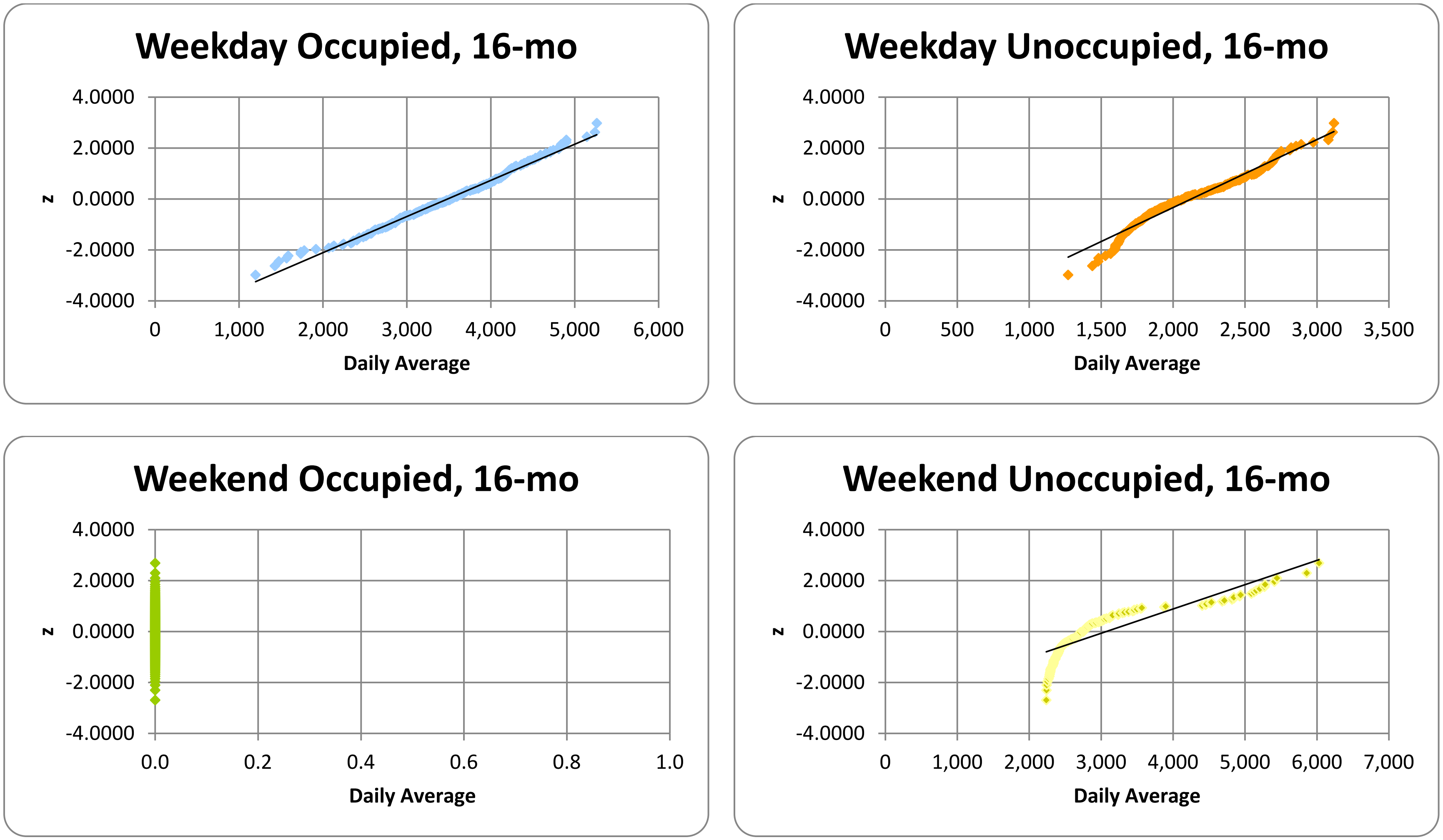
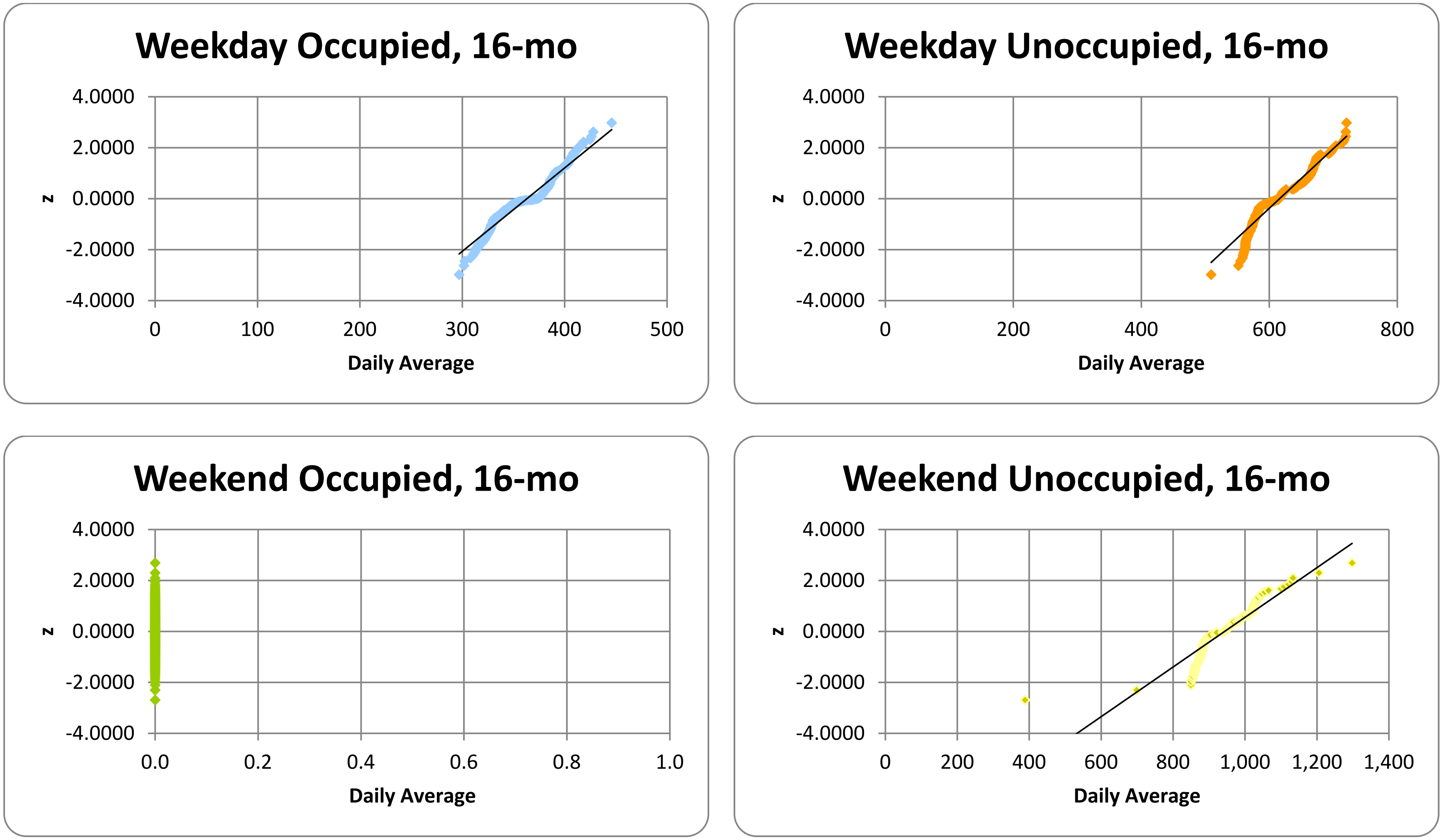


Figure 52: Building 5, Meter 2, normal probability plots by period type



s

References

1. Chisholm, Malcolm, How to Build a Business Rules Engine, Morgan Kauffman, San Francisco, 2004.

2. Energy Policy Act of 1992, 102nd Congress, H.R. 776, Full text at: http://thomas.loc.gov/cgi-bin/query/z?c102:H.R.776.ENR:

3. International Performance Measurement & Verification Protocol Committee, *International Performance Measurement & Verification Protocol: Concepts and Options for Determining Energy and Water Savings Volume I*, 2002.

4. Kissock, J.K., Reddy, T.A., Fletcher, D. and Claridge, D.E., 1993, Joint Solar Engineering Conference, ASME

5. US DOE, “Energy Impact of Commercial Building Controls and Performance Diagnostics: Market Characterization, Energy Impact of Building Faults and Energy Savings Potential”, November 2005.