3-1-2018

Influencing Effective Electrical Distribution Modernization through Advanced Metering

Jared R. Erickson

INFLUENCING EFFECTIVE ELECTRICAL DISTRIBUTION MODERNIZATION THROUGH ADVANCED METERING

THESIS

Jared R. Erickson, Captain, USAF

AFIT-ENV-MS-18-M-201

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-MS-18-M-201

INFLUENCING EFFECTIVE ELECTRICAL DISTRIBUTION MODERNIZATION THROUGH ADVANCED METERING

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Engineering Management

Jared R. Erickson, BS

Captain, USAF

March 2018

DISTRIBUTION STATEMENT A.

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENV-MS-18-M-201

INFLUENCING EFFECTIVE ELECTRICAL DISTRIBUTION MODERNIZATION THROUGH ADVANCED METERING

Jared R. Erickson, BS

Captain, USAF

Committee Membership

Lt Col Jeffrey Parr
Chair

Dr. Diedrich Prigge
Member

Maj Josh Aldred
Member

Capt Craig Poulin
Member

AFIT-ENV-MS-18-M-201

Abstract

In 2013, the Department of Defense (DoD) required its services to implement advanced metering for the purpose of reducing energy usage (Department of Defense, 2013). Additionally, the DoD has aimed to improve its ability to assure a continuous energy supply to all of its installations. This study investigated processes for applying advanced meters on Air Force bases to increase energy assurance. This study also identified strategies for using advanced meters to influence infrastructure funding. This was accomplished through the use of extensive advanced meter data. The data was analyzed for outages and a procedure was created to locate outages in energy usage datasets by using means and standard deviations. Advanced meters with more frequent data collection were able to locate outages easier than meters with less frequent data collection. Advanced meters do not only reduce energy usage, but they also have the ability to report outages. By collecting outage data, funding can be applied to the least reliable electrical infrastructure.

Acknowledgments

Thank you to my beautiful, sweet wife for always supporting me. Thank you for listening to me day after day talk about advanced meters and always asking me how you can help. I don't think I would have made it here without you. You are the best. To my committee, thank you for giving me direction and help every time I asked for it. Thank you for guiding and supporting me even when I was chasing wild ideas. Thank you to all those people and organizations that provided data and programs that made my research possible.

Jared R. Erickson

Table of Contents

Page

List of Figures

Page

List of Tables

Page

List of Equations

Page

INFLUENCING EFFECTIVE ELECTRICAL DISTRIBUTION MODERNIZATION THROUGH ADVANCED METERING

I. Introduction

1.1 Background

The United States Department of Defense (DoD) defines energy resilience as "the ability to prepare for and recover from energy disruptions that impact mission assurance on military installations"(Department of Defense, 2016a). This energy resilience is increased and protected by the expenditure of tax payer dollars to modernize and improve utility systems (Department of Defense, 2004). The then Assistant Secretary of the Air Force for Installations, Environment, and Energy, Ms. Miranda Ballentine, stated, "Over the last 10 to 15 years our missions have become more and more dependent on [the] steady flow of electrons." She also stated that the steady flow of electrons are as important, for some missions, as jet fuel to the Air Force's aircraft (Pew Charitable Trusts, 2017). The leaders of the United States Air Force understand that accomplishing core missions to solidify the United States' National Security requires "significant amounts of energy" (Department of the Air Force, 2017b). This significant amount equated to \$3.5 billion or 198 TBtu of installation energy in Fiscal Year 2016 (FY16) for the DoD (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2016b). This amount of energy is equivalent to three times the annual energy produced by the Grand Coulee Dam, the largest power plant in North America

(United States Department of the Interior: Bureau of Reclamation, 2015). Military leaders also understand that clear access to energy is also vital to National Security. The DoD has set up an energy strategy to ensure "resilient, available, reliable, and continuous power" is provided to its installations (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2016b).

The Air Force's strategy includes "improving resiliency" by seeking to "mitigate impacts from disruptions in energy supplies to critical assets, installations, and priority missions" (Department of the Air Force, 2017b). These disruptions and the recovery of those disruptions are outlined in Air Force Policy Directive 90-17, where it states that "The Air Force will…[b]e able to power any infrastructure identified as critical to the performance of mission essential functions independent of the utility grid for the period of time needed to relocate the mission or for at least seven days, whichever is longer" (Department of the Air Force, 2016c). The direction of a minimum of seven days grid independence allows mission owners to have an objective and a goal for planning and infrastructure upgrades.

Issues arise when infrastructure upgrades must be tailored to fit in the bounds of a finite construction effort where they compete against other projects at bases around the world for funding. This competition is necessary because the DoD cannot fund all of its facility requirements. Choosing which energy project to fund in the Air Force is guided by the Secretary of the Air Force's priority of "Cost-Effective Modernize…to increase the lethality of the force" (Secretary of the Air Force, 2017). Due of prioritization models, if the impact of an energy project is not tangible, then the project does not get funded (see section 2.4 Asset Management). Identifying exactly where the issues are and where funding needs to be sent can be answered by applying data from advanced electrical meters.

Electrical metering of "all Federal buildings" is required by the National Energy Conservation Policy Act (The United States Congress, 2005). A 2013 policy memorandum from the Office of the Secretary of Defense further outlined additional metering requirements for the DoD. Contained therein was the goal of capturing 60 percent of the DoD's electrical energy usage by year 2020 through the use of "advanced meters" (Air Force Civil Engineer Center, 2017a; Department of Defense, 2013). Additionally, a five percent energy cost savings was used to justify the installation costs of these meters (Department of Defense, 2013). These meters have the ability to meet congressional requirements while also identifying issues in real property and electrical infrastructure owned by the Air Force and the DoD.

1.2 Problem Statement

The DoD seeks to minimize the number of utility outages experienced on its installations. The DoD requires all installations to report utility outage details in order to inform Congress on the status of its energy usage (The United States Congress, 2017b). The causes of utility outages for the DoD during the last five fiscal years are shown in Figure 1; the largest cause being equipment failure as shown in Figure 2. In an effort to reduce these outages and increase the quality of power, the DoD has created energy objectives. These objectives are: reduce demand, expand supply, and adapt future forces and technologies (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2016b). If focus is placed on modernizing and upgrading the equipment, equipment failure could become a smaller percentage of the problem. The issue remains to identify where exactly to modernize and upgrade.

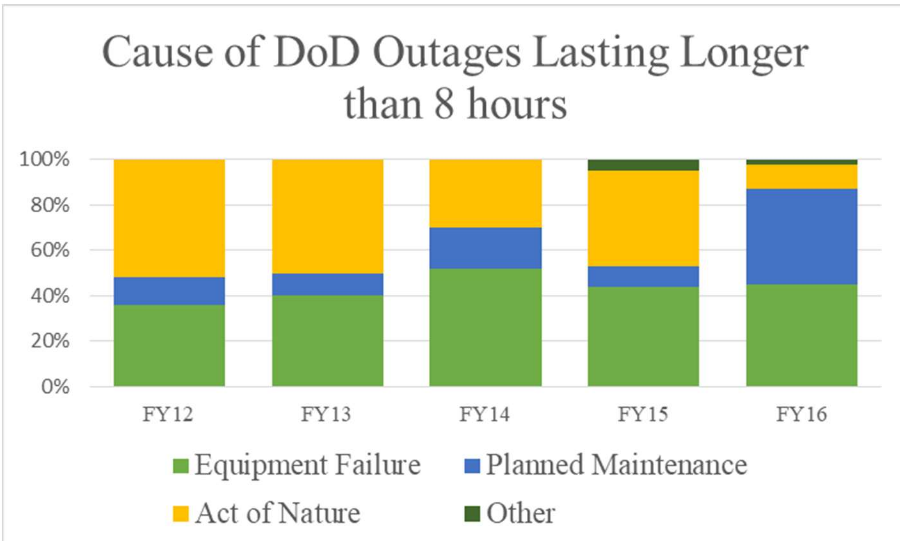


Figure 1. Cause of DoD Outages Lasting Longer than 8 hour

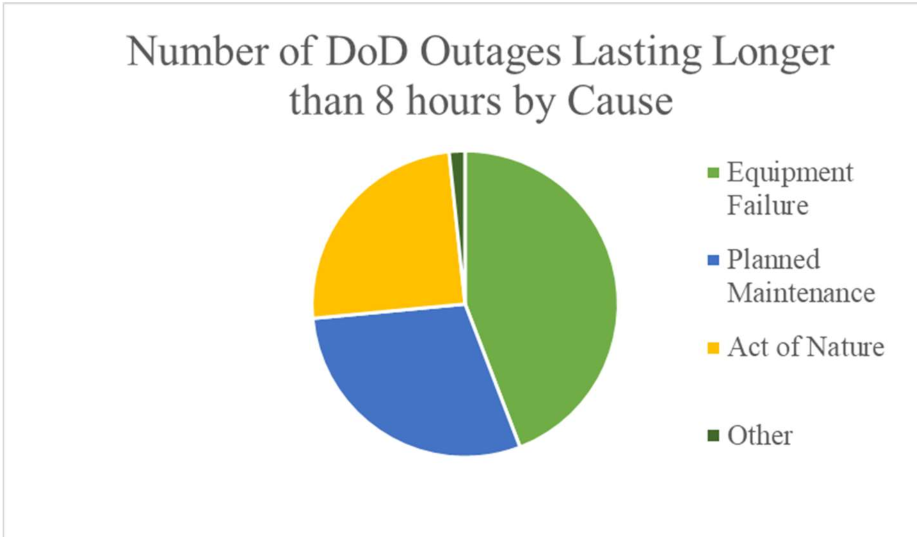


Figure 2. (FY12-16) Number of DoD Outages Lasting Longer than 8 hours by Cause

Without clear understanding of where the exact issues lay, infrastructure funds can be spent in less than optimum ways. The DoD and the Air Force fund energy projects through the same appropriations and financial channels that other infrastructure components fall under (Department of Defense, 2004). This makes it difficult for energy projects to stand out amongst the rest of the projects that also compete for this money. With a better understanding of where the weakest links in the energy network exist, more directed and equitable funding can be disbursed. This understanding can be gained with the use of electrical energy meter data.

1.3 Research Objectives

The objective of this research is to develop best practices for the Advanced Metering Infrastructure (AMI) in order to provide the greatest energy resilience impact for the minimum economic input. Additionally, this research aims to build a framework for pin-pointing the most unreliable components of electrical distribution systems at individual installations. This research effort is also geared towards evaluating the current prioritization of energy related projects and how applying advanced meter data to those priorities could assist in funding more effective projects. With a more informed strategy of how to manage the electrical distribution systems, more fiscally minded investments can be made to improve the DoD's energy resilience.

1.4 Research Focus

The focus of this research is to better inform decision makers on the best way to spend the next available energy dollar. Before spending money on modernizing electrical distribution networks the Air Force needs to know where to apply that funding. The Air Force recognizes the importance of meter implementation and how it can "drive cost-effective, energy management and investment practices," but more can be done to make these far-reaching principles a reality throughout the Air Force (Department of the Air Force, 2016b). It is imperative to ensure taxpayer dollars are being spent effectively and efficiently on these investments especially with an increasingly diminishing DoD budget.

1.5 Investigative Questions

Investigative questions help give more focus to the research. Three main questions for this thesis on energy resilience and advanced metering are as follows:

i. Can outages be found using historical advanced meter data with Means and Standard Deviations?

    A. If so, what Standard Deviation should be used?

ii. What is the optimum data interval and combination to identify outages using historical advanced meter data?

iii. How does advanced meter data affect electrical grid modernization planning including advanced energy production technology?

1.6 Methodology

The methodology used in this research was built around Sensitivity Analysis. These methods will be further explained and applied in Chapter 3. Sensitivity Analysis will be conducted in order to identify electrical outages using different meter data intervals and configurations. Simulations and data analysis were primarily built upon Pecan Street Data from the University of Texas at Austin; a residential advance meter database. Findings were tested against electrical distribution networks and electrical energy usage from Ellsworth Air Force Base (AFB). Applications that were used in the thesis are the Microgrid Design Tool Kit from the Sandia National Laboratory and a tool developed by the Massachusetts Institute of Technology Lincoln Laboratory Energy Systems Group.

1.7 Assumptions/Limitations

The assumptions and limitations of this study are primarily based on the nature of the current data and information that is available within the DoD. Energy metering, outage and consumption data through the DoD Annual Energy Management Report and data obtained by facility operators are assumed to be accurate to the point that clear conclusions can be based. A limitation of this thesis is that costs of advanced meters are not investigated or applied in the analysis.

1.8 Implications

Possible implications include a refined way of looking at meter data to inform the investments the Air Force makes on its electrical distribution systems. Under the current prioritization models for utility projects, including electrical distribution, incorrect ranking is occurring inherently because models are not perfect. If electrical distribution projects for the Air Force are documented with outage information, then more equitable funding can be dispersed for important but misrepresented projects. Doing so would help the Air Force make more objective decisions thus improving their decision making ability.

II. Literature Review

2.1 Introduction

The purpose of this chapter is to provide the baseline for the research of this thesis. Much work as already been done in the fields of advanced metering, military facility asset management and infrastructure modernization. By exploring these and other research topics, synergistic benefits can be seen by applying these topics cross functionally.

2.2 Definitions

When investigating the quality of power at a distribution level three related definitions are vital to understand: resilience, availability, and reliability. First, energy resilience is defined by DoD Instruction 4170.11, *Installation Energy Management,* as "the ability to prepare for and recover from energy disruptions that impact mission assurance on military installations" (Department of Defense, 2016a). Thus, military energy resilience is the ability to minimize the impact on mission accomplishment from a distribution outage or disruption. The Air Force's approach to energy management is defined by the mantra, "Mission assurance through energy assurance" while concentrating on energy resilience (Department of the Air Force, 2016c).

Next, Power availability is defined by Army Technical Manual (TM) 5-698-1, as "the percentage of time that a system is available to perform its required function(s)" (Department of the Army, 2007). The Institute of Electrical and Electronic Engineers

(IEEE) similarly, but with more expansion, define availability as "the ability of an item under combined aspects of its reliability, maintainability, and maintenance support to perform its required function at a stated instant of time or over a stated period of time" (Institute of Electrical and Electronic Engineers, 2007). This makes availability a measure of access to electricity to perform an intended function. Availability is sometimes measured in unserved load in watt-hours, the amount of energy that would have been used if electricity was available (N Judson, Pina, Dydek, Castillo, & Van Broekhoven, 2016; Nick Judson & Pina, 2017).

Lastly, reliability is also defined by Army TM 5-698-1 as "the probability and frequency of failures (or more correctly, the lack of failures)" (Department of the Army, 2007). Again, the IEEE also defines reliability as, "the ability of a component or system to perform required functions under stated conditions for a stated period of time" (Institute of Electrical and Electronic Engineers, 2007). Reliability, in a basic sense, can be seen as the percentage of times a light will turn on when the switch is flipped.

Even with proper definitions, these topics can become confusing and the goals of each specific energy attribute can be blurred. Resilience is different from availability which is different from reliability. All definitions could also fit under the umbrella of energy assurance. To aid to the confusion, most State Public Utility Commissions do not distinguish between resilience and reliability when evaluating the economics of an energy project (K. LaCommare, Larsen, & Eto, 2016). With additional attention to detail, these definitions assist in the planning and evaluating energy related projects. These three definitions, with no one definition being more important than the other, lay the ground work for communicating about power quality and provide metrics to measure then improve and achieve.

2.3 Funding

All of the services in the DoD manage their installation energy programs differently. The Air Force uses Head Quarters U.S. Air Force (HAF) to provide “policy, guidance, oversight, and resources,” and it uses the Air Force Civil Engineer Center (AFCEC) to “manage and facilitate the execution of energy programs” (Office of the Assistant Secretary of Defense, 2016). These two offices work in tandem to meet the three Air Force Energy Goals: Improve Resiliency, Optimize Demand, and Assure Supply (Department of the Air Force, 2017). Some of the ways they meet these goals are through the use of appropriated funds and utility privatization approved through Congress (Department of Defense, 2004; The United States Congress, 2017).

Utility privatization is accomplished by setting up a contract with a private company to operate and maintain a given utility. Utilities include the generation, treatment, collection, transmission, or supply of electric power, water, wastewater, steam, hot water, chilled water, natural gas or telecommunications (The United States Congress, 2017a). Defense Logistics Agency (DLA) is a governmental agency that provides logistical support to the DoD and other government departments and agencies. DLA also has an energy division that “provides petroleum products/lubes, alternative fuel/renewable energy, aerospace energy, fuel quality/technical support, fuel card programs, and installation energy services" (Defense Logistics Agency, 2017). This division assists the DoD with completing the privatization process on all of its service's installations. In DoD Instruction 4140.25, it states that DLA "assists the DoD Components with global energy commodity infrastructure privatization and demand management, including technical and contract support for the management of energy commodities" (Department of Defense, 2015a). DLA through privatization has saved the Air Force 511 million dollars in cost avoidance of infrastructure maintenance and commodity purchasing through more efficient systems (Air Force Civil Engineer Center, 2017f). In a capitalistic economy the contractor wants to maximize profit and is incentivized to reduce energy usage. Privatization is a useful tool because it places the opportunity to succeed and risks on the contractor.

Energy Savings Performance Contract (ESPC) is a program where third-party investing is incorporated to projects where energy can be saved or reduced. The Air Force will continue paying at the normal, pre-ESPC, billing rate and any savings that is generated will be used to pay the third party back for providing the capital for the upgrades. These efforts are over seen by an Energy Service Company (ESCO) who is charged providing the service to the Air Force while paying back the third party through energy savings. If the project does not reduce energy the contractor will not be paid (Air Force Civil Engineer Center, 2017b). ESPCs require basic metering and often use advanced metering to validate the upgrades and success of the project (Marqusee, Schultz, & Robyn, 2017). Again, the contractor is given the opportunity to provide a service and maximize profit like utility privatization.

The Energy Resilience and Conservation Investment Program (ERCIP) is a portion of the Military Construction (MILCON) appropriation that is used to perform major construction efforts in the DoD (Air Force Civil Engineer Center, 2017c; Defense Acquisition University, 2017). In the Air Force, the 3300 MILCON appropriations are received from the US Congress for execution through the annual National Defense Authorization Act (Defense Acquisition University, 2017). According to the Energy Portfolio Integration Manager of the Air Force, the Air Force receives 40-50 million dollars annually for ERCIP related construction (Ramos, 2017). These projects are funded to produce or reduce energy usage in all its forms (Air Force Civil Engineer Center, 2017c). The Air Force can also receive money directly from the MILCON program for energy related projects, but that is unlikely because of the creation of the ERCIP subset (Ramos, 2017).

Air Force and the DoD also use power purchase agreements (PPA) and enhanced use leases (EULs) to provide power to their installations. These programs leverage real property and private contractors to create an advantageous situation for both parties involved. PPA are set up with contractors to provide renewable energy specifically for use on a DoD installation (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2017). EULs are agreements that provide a lease on land owned by the DoD to contractors for the purpose of energy production or reduction

(Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2017). PPA and EUL are very reliant on good contracting practices and require just as much, if not more, involvement than any other project performed by the Air Force.

The final way the Air Force updates and modernizes their infrastructure through the appropriations for Facilities Sustainment, Restoration and Modernization (FSRM) under the 3400 Operations and Maintenance (O&M) appropriation made available to the DoD through the U.S. Congress (Defense Acquisition University, 2017). This money is designated "for means the alteration or replacement of facilities solely to implement new or higher standards, to accommodate new functions, or to replace building components" including "utility plants [and] distribution systems" (Department of Defense, 2004). These funds are used to sustain the operations of the military and their funding level is usually given to the Air Force as a function or fraction of their current assets (Department of Defense, 2015b). The assets must then be prioritized so that money can be dispersed in a systematic way. This system is called asset management.

2.4 Asset Management

In Executive Order 13327, the term asset management is coined as the "efficient and economical use of America's real property assets" (Bush, 2004). Many times for the DoD that means attempting to manage assets—facilities and equipment—that were constructed or installed around World War II (Department of Defense, 2015b). The condition and importance of the many different and complex infrastructure items have driven the need for a focused approach to maintenance and modernization. This approach entails focusing investments and manpower in a systematic way to prevent mission failure—an approach known as asset management. One of these systematic methods is applying a Mission Dependency Index (MDI) to aid in prioritizing facility funding in the Air Force. Grussing et al. explains MDI as "the relative importance of an infrastructure asset (facility) in terms of its mission criticality" (Grussing et al., 2010). Standard MDIs are created for all facilities enterprise wide and deviations from the standard are reviewed at AFCEC (Mills et al., 2017). The standard list of MDIs are For Official Use Only and will not appear in this thesis, however a breakdown of MDI can be seen in Table 1.

Table 1. MDI Breakdowns (Mills et al., 2017)

Tier	Criteria	Recommended MDI
1	Mission critical	85 to 99
2	Direct mission support	70 to 84
3	Base support	60 to 69
4	Community support	below 60

An issue with the standard MDI is that it does not account for connected systems like utilities, also known as "network facilities," that service multiple facilities (Department of the Air Force, 2016a). Rather, standard MDIs are set for each piece of infrastructure individually. A study conducted by the Research and Development (RAND) Corporation by Mills et al. contends that while these MDIs are useful they should not be blindly trusted. Mills et al. claim that additional work needs to be done to "reveal and clarify critical linkages" that exist on Air Force installations (Mills et al., 2017). The issue arises because MDIs are used directly in the prioritization of FSRM projects. The MDI of a facility accounts for 60 percent of the Consequence of Failure (CoF) metric (Mills et al., 2017). Without the "critical linkages," projects and infrastructure could be misrepresented and their true impact misunderstood.

The other aspect of asset management is doing predictive maintenance or condition based maintenance where equipment and facilities are not repaired or maintained unless the health of the equipment or facility indicates it is warranted (Software Engineering Institute, 2011). This leaves room for interpretation and flexibility in maintaining infrastructure. Mills et al. report that infrastructure that is degraded will require more repair funds as time goes by and they are more susceptible to outages and catastrophic failures (Mills et al., 2017). Specific to electrical distribution systems, Bahmanyar et al. recommend that investments in the distribution lines and preventative maintenance, such as increased tree trimming, are required in order to improve performance (A Bahmanyar, Jamali, Estebsari, & Bompard, 2017). To quantify and standardize the condition of assets across all military services, a probability of failure (PoF) metric was developed through the use of a Sustainment Management System (SMS) (Mills et al., 2017; Under Secretary of Defense for Acquisition Technology and Logistics, 2013). The SMS is a computer based program for evaluating the condition of a facility or a piece of equipment. This PoF value is a vital component to the prioritization of projects in the Air Force (Mills et al., 2017).

CoF and PoF account for 200 out of 210 points of the scoring metric which decides the fate of which projects will receive funds from Congress. The final 10 points, approximately five percent of the total points, are awarded to a project based off of the savings to investment ratio (SIR). The SIR is a calculation of the total discounted operational savings divided by the total investment for the project (Air Force Civil Engineer Center, 2017c; Mills et al., 2017). This value is found by using the Building Life Cycle Cost (BLCC) tool developed by the Department of Energy following the Federal Life Cycle Cost Methodology and Procedures found in Title 10 of the Code of Federal Regulations (Air Force Civil Engineer Center, 2017c). While seemingly small, the SIR points could mean the difference in getting funded or not when prioritized against all the projects throughout the Air Force.

The CoF and PoF are even more specific for "Utilities" projects. Utilities are defined as "a facility or system composed of one or more pieces of equipment connected to or part of a structure and designed to produce, transmit, or distribute a service such as heat, electricity, water, or sewage disposal" (Department of the Air Force, 2016a). The PoF of electrical utilities is calculated using the remaining useful life and direct condition rating (found using visual inspection) (Air Force Civil Engineer Center, 2017e). For other utilities, including water and steam infrastructure, PoF is also calculated using the total number of outages experienced. AFCEC's Electrical Sub-Activity Management Plan Program Manager stated that electrical outages are not considered in the PoF metric because if an outage occurs it is usually because equipment fails and that equipment is directly replaced (Benson, 2017). Equipment could be a large part for systemic failures, but the issues may not be solved with a simple swap of equipment. In fact, Krajnak states that the collection of electrical fault information can help identify "problem areas" for future modernization (Krajnak, 2000). Advanced meter data could be used to find these systemic issues and historical patterns.

2.4.1 Critical Review of Facilities Sustainment, Restoration and Modernization

A review was performed on the FY17 and 18 Integrated Priority Lists (IPLs). The IPLs were analyzed for utility projects and how electric projects compare to other utilities that use outages as a measure of prioritization. The FY18 IPL contained 4,950 projects or other funding line items. The Air Force announced 3,376 projects or line items, worth approximately 2.1 billion dollars, for funding on its Construction Task Order. The IPL does not designate what type of project it is because some projects are complex. However, the IPL does break down by the recommended Activity Management Plan (AMP). AMP categories in the FY18 IPL included: Utilities, Facilities, Transportation, Natural Infrastructure and Real Estate. Electrical projects are a part of the Utilities AMP, which contains 358 projects. Other types of projects in the Utilities AMP included: Water, Wastewater, HVAC, and Natural Gas. To locate electrical projects in the IPL a word search of the project title and project description was conducted within the Utilities AMP projects. The first word used was "electric," from there additional words were used to locate electrical projects. These additional words included: switch gear, substation, low volt, high volt, volt, cables, transformer, generator, distribution, power, light, energy, circuit, and feeder. The Utilities AMP projects were again reviewed line by line to locate missing electrical projects. If project included multiple utilities, then the project would still be designated as an electrical project. This process was repeated for the FY17 IPL to increase the sample size.

The Facilities Sustainment, Restoration and Modernization (FSRM) budget is executed by using an IPL to prioritize projects for future funding. As described in the previous section, electrical projects in the FSRM program do not use outage information to influence funding through the probability of failure (PoF) metric. This review was conducted by comparing characteristic between Utilities Activity Management Plan projects. Figure 3 gives the breakdown of the relevant data comparisons.

		# of proj	In CTO	Below CTO	Cost per Project	Mean MAJCOM Priority	Medain MAJCOM Priority	Mean Prob of Failure	Median Prob of Failure
2017 IPL	Non-Electrical	408	39%	61%	$946K	137	107	85	100
	Electrical	155 (28% total)	30%	70%	$1,070K	193	204	75	80
2018 IPL	Non-Electrical	341	67%	33%	$1,426K	83	72	97	100
	Electrical	111 (25% total)	68%	32%	$1,519K	94	70	92	100

Figure 3. Review Outputs of Integrated Priority List

The 2017 and 2018 ILPs showed similar trends points to issues that could be occurring in FSRM program. In 2017, a higher percentage of non-electrical projects were in the Construction Task Order (CTO) for funding. For 2018, all utilities projects were in the CTO close to or at the same rate. For both 2017 and 2018, electrical projects were more expensive, received a worse priority, and rated lower for the PoF metric. It is unclear if outage information could have changed the PoF. Both electrical and non-electrical utilities projects have the same potential to score 100 on the PoF. It is very possible that non-electrical utilities actually do have a worse condition overall in the Air

Force. This comparison does not prove statistical difference, but simply shows the apparent gap between electrical and non-electrical utility projects. This review is not rigorous analysis, but analysis should be completed on this topic to see if this issue is systemic or what is the true cause for this discrepancy.

2.5 Electrical Meters

As required by 10 U.S.C. §2924, 10 U.S.C. §2911 and 42 U.S.C. §8258, the DoD is required to report energy usage, energy management activities, and energy reduction to Congress (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2017). The military has begun the implementation of Advanced Metering Infrastructure (AMI) to better collect this data for Congress. According to the Software Engineering Institute, AMI are, "Systems that measure, collect, and analyze energy usage from advanced devices such as electricity meters, gas meters, and/or water meters, through various communication media on request or on a predefined schedule" (Software Engineering Institute, 2011). The term "advanced meter" is commonly interchanged with the term "smart meter." In fact, Congress uses the terms interchangeably in the Energy Policy Act of 2005 (The United States Congress, 2005). Liu et al. state that advanced meters are only used by producers and consumers. They state that producers are interested in usage habits to best serve their customers while making the most money and consumers just want to know how to save money (Liu, Golab, Golab, Ilyas, & Jin, 2016). The DoD appears to fit into both the producer and consumer roles, making advanced metering a conceivable program. The Air Force's AMI program is called the Advanced Meter Reading System (AMRS).

The AMRS collects energy usage data every 15 minutes on specific single facilities or single transformers if a facility has more than one transformer (Air Force Civil Engineer Center, 2017a; Carnley, 2017; Department of the Air Force, 2017a). The specific data stored includes but is not limited to, "power factor, peak power, total consumption over time, and energy use" (Department of the Air Force, 2017a). The meters specified in the Air Force Meter Data Management Plan (MDMP) have the ability to store and communicate other energy data as well—including highlighting outages as seen later in this section. The AMRS is forecasted to save over 20 million dollars in energy costs a year once in place (Department of the Air Force, 2017a). By applying the MDMP and other guidance from AFCEC, the Air Force is on track for meeting the congressional mandates for advanced metering.

Electrical metering of "all Federal buildings" is required by the National Energy Conservation Policy Act. This law required basic metering to be accomplished by the end of FY12 (The United States Congress, 2005). The DoD met Congresses' meter requirement and then DoD took facility metering even further by requiring advanced meters deployment by the services. On April 16, 2013, the then Acting Deputy Under Secretary of Defense for Installations and Environment, John Conger, signed an energy meter policy memorandum for all services. The policy expands the Advanced Meter deployment to 60 percent of all electricity usage by the end of Fiscal Year 2020

(Department of Defense, 2013). The progress of each services' advanced meter deployment is tracked in the Annual Energy Management and Resilience Report (AEMRR), formerly known as the Annual Energy Management Report (AEMR) prior to 2017, to Congress (Air Force Civil Engineer Center Public Affairs, 2017; Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2016b). The Air Force captured eight percent of the total energy usage at the end of FY16 and hope to have 27 percent captured by the end of FY17 through the AMRS (Air Force Civil Engineer Center, 2017a). This retrofitting of existing infrastructure will culminate at the end of FY20 by deploying advanced meters on 42 bases (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2016b).

Advanced meters are also required on all newly constructed facilities and repair projects (MILCON, ERCIP, ESPC and FSRM projects) with few exceptions. Those meters must communicate with that services' meter program, making them advanced meters (Department of Defense, 2014). The DoD directs services to use five percent of annual energy usage as a cost saving factor to identify cost effective advanced meter placement; if the meter is not cost effective, it will not be required (Department of Defense, 2013). If meters are not cost effective, then the program is missing a major motivation for installing AMI which is reducing facility expenditures.

The DoD's estimate of five percent cost savings is very conservative and could be restricting the full potential of the AMI. AFCEC reports that up to five percent savings can be realized by simply increasing the occupant's awareness with mock energy bills.

AFCEC even reports savings of 15 to 45 percent by continuously commissioning and evaluating facility energy (Air Force Civil Engineer Center, 2017a). The U.S. Army even reported a cost savings of 60 percent in a single facility because of the identification of simultaneous heating and cooling in a facility (Parker et al., 2015). The DoD's Environmental Security Technology Certification Program (ESTCP) commissioned a report that found energy efficient projects resulting in 14 percent energy reduction could be identified by using hourly data from advanced meters alone. This was proven using 5.5 million square feet of DoD buildings (Shah, 2014). The Department of Energy also reported 3 other projects that reported anywhere from 10 to 20 percent energy reduction through the use of advanced metering (Parker et al., 2015).

Advanced meters are not always seen in the best light. One issue is the cost to install, maintain and operate the advanced meters and the AMRS. The AFCEC AMRS Chief Program Officer reported that individual meter installations could cost anywhere from 12.5 thousand dollars to 50 thousand dollars (Carnley, 2017). Additional funds must be used to operated and maintain these systems to keep them current and functional. Operations and maintenance costs of these advanced meters are estimated at 300 dollars per year, but the true value will become more apparent and accurate as the AMRS continues to be deployed in the military (Air Force Civil Engineer Center, 2017a). Data storage is also a vital piece of operating the AMRS. Zhou et al. describe how data from advanced meters can grow to overwhelming levels. They point out that one million meters reporting at 15-minute intervals (the Air Force standard) in a single year will produce 2920 Terabytes of data (Zhou, Fu, & Yang, 2016). In the Air Force MDMP it makes no mention to the plan for data storage, but does claim that in 2015 the Air Force operated 13 thousand advanced meters, prior to the AMRS rollout (Department of the Air Force, 2017a). At these levels, the Air Force would collect approximately 40 Terabytes every year. This amount of data requires special systems to handle and store the data. Not all of the 13 thousand meters can be converted for use on the AMRS and it is unclear how many will be converted. However, the amount of data will continue to grow from these advanced meters as they are installed at the 42 bases chosen for the AMRS initial roll out and as facilities are constructed and repaired across the Air Force.

Another possible issue is the failure rate of the advanced meters. A study of PV solar implementation showed that the electrical meter was the cause of one percent of all PV failures. The study had 350 individual PV systems with over 3500 failures in a 27-month period (Formica & Pecht, 2017; Golnas, 2012). Unlike PV failures, broken advanced meters will not cause a facility to lose access to power. Broken meters that do not communicate with the AMRS only cause holes in the energy usage data. However, holes in the data are nontrivial. An ESTCP project recommended the DoD to implement better procedures to eliminate these holes because they greatly affect the usefulness of meter data while completing an energy audit (Shah, 2014). A broken meter could also mask an outage event or other power quality issues. The Pecan Street Data Director claims that approximately only one percent of the NULLs or zeros in their 1,300-home dataset of minute by minute data can be attributed to a utility wide outage. The other 99 percent of the NULLs or zeros could be a mixture of small distribution sized outages, transformer failures, meters being unplugged, and broken meters (Fisher, 2017). Similar to the operations and maintenance, actual failure rates will be understood as more and more meters are installed.

Other issues not examined in this thesis include worries of invasion of privacy surrounding advanced meters. Fortunately, these privacy issues do not exist in the military as transparency and proper stewardship is demanded of the government. Rather cyber security is arguably the largest issue surrounding AMI deployment (Air Force Civil Engineer Center, 2017a). The DoD requires the military services to implement AMI without compromising the DoD network or leaking power usage information for sensitive missions—a task that is difficult to define and continues to change day by day (Department of Defense, 2013).

Even with of the above named issues, advanced meters have the ability to report more than just energy usage, making them a force multiplier. Air Force Pamphlet 32-10144, *Implementing Utilities at U.S. Air Force Installations,* highlights the fact the "energy data systems" should be used to generated cost savings by identifying equipment to replace (Department of the Air Force, 2016b). This idea may be applied to facilities and infrastructure as well, if correctly used. Advanced meters also have the possibility of be used to understand the absence of energy. Lawton et al. emphasized the importance of knowing exactly when and for how long outages occurred as they would affect the economic and monetary impact of the loss of power (Lawton, Sullivan, Liere, & Katz, 2003). “Last Gasp” messages are used by most AMIs to communicate real time electrical outage occurrences. It is called Last Gasp because meters—and their accompanying communication systems—are powered by the same electricity that it is measuring and when the power goes down, the meter has to send the Last Gasp message prior to losing power (National Electrical Manufacturers Association, 2017). Tram explicitly states that commercial power companies rely on these Last Gasp messages to improve their power quality and reduce operations cost associated with outages (Tram, 2008). The U.S. Department of Energy created a design guide for these Last Gasp messages, in which they explain that communication becomes backlogged and unusable when a large number of meters are all sending their Last Gasp messages at the same time during a massive outage event (The Department of Energy Office of Electricity Delivery and Energy Reliability, 2010). The exact source of the failure will still be found while manually troubleshooting the fault and bring the power back on. This does not mean the limited data collected during the communication backlog is useless, but rather could still have a story to tell. By collecting Last Gasp messages, information on when and how long these outages are occurring can be archived and analyzed to identify problem areas of the base grid.

In some cases, Last Gasp messages are not used for electrical meters. If that is the case, it makes locating outages harder because there is no clear indication that electricity is not flowing when looking at the dataset. One strategy could be looking at a dataset for areas of zero reported energy usage. This is not completely accurate because the zeros in the dataset could also be from other malfunctions in the energy meter, as described above in this section (Fisher, 2017). If the assumption was made that something went wrong with the meter or facility, including the possibility of an outage, then one could begin to analyze the data. Any electrical consumer, residential or commercial, will see a constant leak of electrical energy due to modern-day electronics. In a report created by the Lawrence Berkeley National Laboratory for the California Energy Commission, Meier et al. show that in some cases 13 percent of all energy usage is due to this constant leak caused by electronics in “low power” or “sleep” modes that are constantly draining as long as they are plugged into a wall (Meier et al., 2008). Anything plugged into a convenience receptacle, or wall plug, that is either turned on or off has the potential to draw some small amperage. That small amperage can be measured and stored any energy meters, thus showing periods of time with low energy usage. Because of this, it is assumed that an outage or malfunction occurs when there is a zero in an energy usage dataset for occupied buildings.

Past outages or malfunctions can be identified if Last Gasp message are not used by looking at historical data if the following conditions apply:

(1) the meters have an auxiliary power source to enable communications and measurement in the absence of power or the dataset stores NULL or zero values when disconnected from the energy meter,

(2) the meters are correctly communicating energy usages over their prescribed intervals, and

(3) broken or inoperable meters are clearly identified in the dataset.

The Unified Facilities Guide Specifications (UFGS) are followed and used in all military projects to define what exactly is required when facilities are constructed or repaired. UFGS 26 27 13.10 30, *Electric Meters*, defines 15 minutes to be the standard interval with which the energy usage is to be reported on advanced electrical meters (Air Force Civil Engineer Center, 2017d). The Department of Energy also recommends a 15-minute interval for “energy system diagnostics” but also states “shorter intervals for end use diagnostics” and “as frequent as required” to enable some flexibility in implementation (Parker et al., 2015). It is unclear if a 15-minute data interval is too long or too short to be effective at pinpointing outages if the above three conditions exist.

2.6 Electrical Outages

As required by 10 U.S.C. §2925, the DoD is required to report utility outages (electricity, water or gas) lasting longer that eight hours to Congress (The United States Congress, 2017b). As reported in the previous five AEMRs, the DoD experienced multiple utility outages as seen in Figure 4 (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2014, 2015, 2016b, 2017). The interesting spike in FY16 was attributed to including, for the first time, outages that were mitigated by backup systems or generators. (The United States Congress, 2017b). These outages cost the DoD upwards of half a million dollars per 24-hour period due to fuel costs or downed operations (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2017). The data obviously does not capture everything; specifically not reporting any outages less than eight hours. In a report commissioned by the Pew Charitable Trusts, Marqusee et al. emphasis the importance of understanding the real outages that are occurring on military bases. They highlight the limited outage information in the AEMR, but also looked at reliability metrics of the utility providers for the 30 largest military installations and maintenance logs from the Navy. They make the claim that there are many outages that are not being captured by the current business rules and that reliability and availability on military installations could be and is worse than reported (Marqusee et al., 2017).

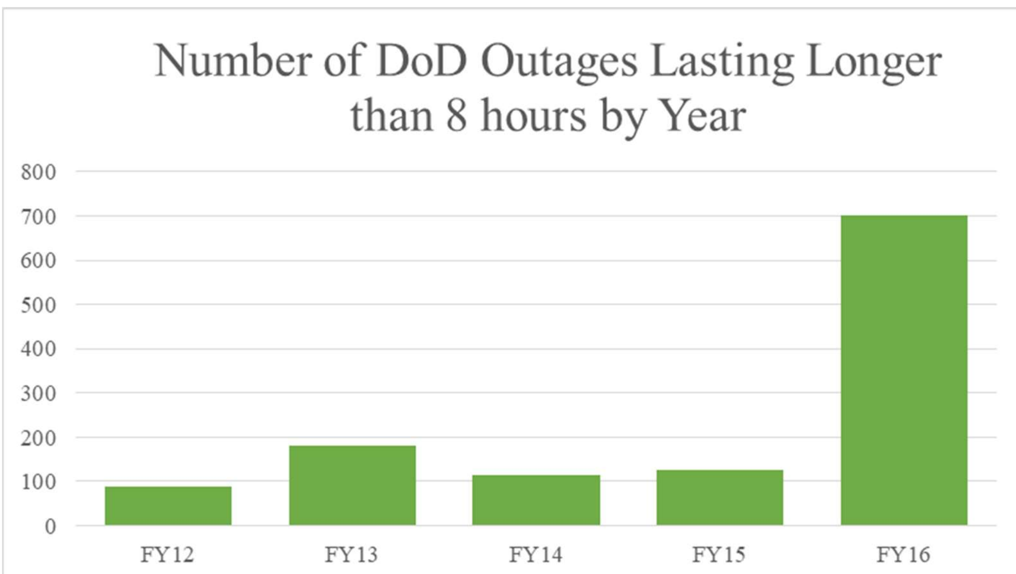


Figure 4. Number of DoD Outages Lasting Longer than 8 hours by Year

Current outage detection in the academic and private sectors is accomplished by using many different advanced technology and techniques. In a review of fault detection techniques, the definition of a fault is given as "an unpermitted deviation from its standard operating conditions" (Gururajapathy, Mokhlis, & Illias, 2017). These "faults" create outages and disrupt the flow of electricity to the end user. Jamali and Bahmanyar investigated previous fault/outage identification processes. They claim the majority of processes rely on measuring current at multiple locations in a grid. Their processes use voltage measurements at sectionalizers, re-closers and at other electric distribution infrastructure (S. Jamali & Bahmanyar, 2016). Using specially configured advanced meters is vital to their processes. Menchafou et al. theoretically showed that outage physical location could be found using voltage sag by measuring voltage and amperage at substations (Menchafou, Youssef; Markhi, Hassan; Zahri, Mustapha; Habibi, 2015). This measuring would be done by advanced meters. Bahmanyar et al. show a way to autonomously locate and restore power to "healthy" sections of a distribution network. Their processes require "voltage and current measurements at the head of main feeder and the magnitude of voltage sags recorded at some nodes equipped with voltage measurements, such as power quality meters or digital fault recorders" (Alireza Bahmanyar et al., 2016). Power quality meters meet the requirement described by AFCEC's Class I Meters (Air Force Civil Engineer Center, 2017a; Department of the Air Force, 2017a). In another paper, Jamali, Bahmanyar and Bompard, developed an algorithm for finding fault locations using only a limited number of advanced meters located at substations and a few additional points in the distribution. Their algorithm requires near real-time measurement, something AMRS has the ability to do but is not mandated (Sadegh Jamali, Bahmanyar, & Bompard, 2017).

Outage recording at the Air Force level is improving and exceeding the eight hour outage reportable limit, but is not on par with the academic and private sectors. AFCEC has begun to use a reporting system to make outage reporting easier and more accurate. The system is called the Utility System Outage Reporting and Tracking (USORT) Tool (Watley, 2017). The tool will allow operators and engineers to report utility outages and categorize them for improved analysis efforts. As described by Chen et al. outage detection and management is accomplished by many different sources of data (Chen, Dokic, & Kezunovic, 2014). Solely relying on advanced meter data will not tell the whole picture, but the addition of USORT tool is a step in the right direction.

Ernest Orlando Lawrence Berkeley National Laboratory, under a U.S. Department of Energy contract, investigated the monetary effects of outages on commercial and residential facilities. This economic investigation into energy reliability concluded that an outage could cost upwards of 80 thousand dollars per hour, or two million dollars for 24 hours, for large commercial businesses (Lawton et al., 2003). Compared to the daily rate of the DoD of 500 thousand dollars, this drastic difference may imply that the government is undervaluing their operations. Lawton et al. developed their numbers from 24 datasets from the 1990s and early 2000s, thus under reporting the economic impact in the current value of the dollar. Sullivan and Kean point out the dichotomy between customers that want cheap electricity without regard to the quality of power verses customers who put a on power quality and are willing to pay for the increased reliability (Sullivan & Keane, 1995). The DoD wants and is willing to pay for the increased reliability. Outages can be avoided and reliability increased with better asset management of electrical infrastructure; over 40 percent of outages are caused by equipment failure in the DoD (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2017). Paying this "premium" will avoid the high cost of operating without power.

The Institute of Electrical and Electrical Engineers (IEEE) defines standards for electrical distribution systems in the U.S. through Standard 1366, *IEEE Guide for Electric Power Distribution Reliability Indices*. These standards of electricity reliability for private utility companies were defined and explained further in a report also completed by LaCommare and Eto of the Lawrence Berkeley National Laboratory. They define three important metrics; Momentary Average Interruption Frequency Index (MAIFI), System Average Interruption Duration Index (SAIDI) and System Average Interruption Frequency Index (SAIFI). MAIFI is a measurement of interruptions lasting less than five minutes in duration (K. H. LaCommare & Eto, 2004). These momentary "blips" in the system can have far reaching and serious effects; Ms. Ballentine emphasizing this issue through a story about the energy dependence Unmanned Aerial Vehicle operations in her address to the Pew Charitable Trusts (Pew Charitable Trusts, 2017). SAIDI is a measure of the average amount of downtime that each customer will experience in a given year as seen in Equation 1.

$$SAIDI = \frac{\begin{array}{c} Sum\ of\ customer\ (sustained)\ interruption \\ durations\ for\ all\ customers \end{array}}{Total\ number\ of\ customers\ served}$$

Equation 1. System Average Interruption Duration Index (K. H. LaCommare & Eto, 2004)

SAIFI is similar to SAIDI but instead counts the frequency of times that there is an interruption in power supply as seen in Equation 2. MAIFI is an event less than five minutes and SAIFI is everything longer than five minutes.

$$SAIFI = \frac{\begin{array}{c}Total\ number\ of\ customer\ (sustained)\\ interruptions\ for\ all\ customers\end{array}}{Total\ number\ of\ customers\ served}$$

Equation 2. System Average Interruption Frequency Index (K. H. LaCommare & Eto, 2004)

These interruptions are defined as, "The total loss of electric power on one or more normally energized conductors service" and does not include "power quality issues such as: sags, swells, impulses, or harmonics" (Institute of Electrical and Electronic Engineers, 2012). Collection of these metrics are regulated by state Public Utility Commissions (PUC). In 1998 Warren et al. performed a survey of private utilities to investigate the usage of IEEE's standards for distribution reliability and how they are being implemented. This study found that standards were not being followed. Many inconsistencies made the metrics useless for comparisons across the nation and additional guidance was required from the IEEE (Warren, Pearson, & Sheehan, 2003). The IEEE did and continues to update Standard 1366 for those reasons, but PUCs are responsible for enforcing the IEEE standards. Eto and LaCommare tracked PUCs to see how these state commissions were using the reliability data. They reported in 2008 that 35 of the 51

(50 states and Washington D.C.) vPUCs required routine/annual reliability reports using SAIDI and SAIFI which was improvement from 10 PUCs in 2004. Additionally only two PUCs and 12 of the 123 used the MAIFI metric for momentary interruptions (Eto & Lacommare, 2008). In 2016 LaCommare et al. continue to report that there are inconsistencies with reporting but that the addition of the Department of Energy's Energy Information Administration and their Form 861 has greatly improved these inconsistencies (K. LaCommare et al., 2016).

The Pew Charitable Trusts evaluated the private utility's energy reliability of the 30 largest military bases between 2013 and 2014 using SAIFI and SAIDI metrics. This investigation showed that on average, each of the 30 bases experienced one major power outage lasting seven hours. While SAIFI and SAIDI do not describe the exact time of day or part of the year these outages occur, they do help leaders understand the vulnerabilities that are realities on military installations.

One solution offered back in 2000 by Krajnak showed the usefulness of faulted circuit indicators (FCIs). He states the FCIs reduce the time that repair crews spend locating the cause of an outage (Krajnak, 2000). The function that FCIs perform can be performed by advanced meters. This research of FCIs and their placement continues 17 years later with Li et al. showing that the placement of these meters can be optimized to pinpoint locations of outage causing faults (Li, Chen, & Guo, 2017). This research does not comment on the physical location of the outages, but rather the process of identifying their duration and time in historical datasets. Krajnak also states that the FCIs can help identify "problem areas" for future modernization (Krajnak, 2000). Identification of problem areas is something the DoD could use to influence the funding of infrastructure modernization projects.

Whether reporting outages using the current DoD business rules or the IEEE Standard, the main issue continues to be accurately determining, reporting, and collecting the correct data. The DoD could adopt the refined procedures used by the Department of Energy to better report energy quality. The true number of outages, at any duration and not just those over eight hours, is obviously some number larger than that which is indicted to Congress. The impact of these shorter outages will not be understood without a mandate or directive to collect this information. Using advanced meters could automate this reporting and decrease the uncertainty surrounding the true status of the DoD's energy usage.

2.7 Upgrade Infrastructure

In a study commissioned by the Assistant Secretary of Defense for Installation Energy, the Massachusetts Institute of Technology Lincoln Laboratory (MITLL) created a Matlab based tool to analyze the electrical reliability of a military installation in order to "develop the business case framework to support budget and alternative financing decisions" (N Judson et al., 2016). Their technical report investigated the way the military currently manages installation energy at four different bases on each of the four military service components. A finding in the report, which was reiterated in an interview with the author, pointed to the fact that the military does not keep the correct reliability metrics data to fully implement their developed tool. The recommendations to rectify these issues included:

1. Collect Data on actual electrical, heating and cooling loads for critical missions and how those loads vary throughout the day, week and year
2. Track performance data (hours run, failure rates, and maintenance logs) for energy generation in a central electronic database
3. Track existing electrical distribution system outages in a systematic way (N Judson et al., 2016; Nick Judson & Pina, 2017).

This data is not only important to the MITLL reliability tool but also to designing and operating Net-Zero Buildings and Microgrids (Booth, Barnett, Burman, Hambrick, & Westby, 2010; CH2MHILL & Clark-Nexsen, 2016).

According to Ersoz and Colak, using deterministic methods for evaluating the optimum Combine Cooling, Heating and Power (CCHP) implementation is difficult and ineffective due to the number of unknowns and high number of uncertainties (Ersoz & Colak, 2016). CCHP is a high efficiency system where electrical power and cooling is produced and the waste heat is re-used. Ersoz and Colak show that deterministic methods that rely on definite information fail because parameters change or are unknown. Applying probabilities with sensitivity analysis creates a better understanding of the risks involved for the decision maker in this complex investment (Ersoz & Colak, 2016). Applying this approach of stochastic decision making to any type of infrastructure upgrade can prove to be invaluable in current Air Force funding models. Meter data can provide the basis for the application of these stochastic models.

2.7.1 Net Zero Buildings

Brost researched the implementation of Net-Zero Buildings (buildings that produce as much power and/or water that they use) on Air Force installations. In commenting on the current application of advanced meters in Net-Zero Buildings, Brost stated, "As Smart Grid technologies increase, Advanced meters will be incorporated which allow for more efficient management of the grid" (Brost, 2013). Fort Carson is one of the Army's premier locations for their Net-Zero Installation initiative. In Fort Carson's Final Environmental Assessment, the Army stresses the need for having energy meters to properly assess their energy usage, including electrical energy. A follow-on objective is to implement a microgrid on the installation for the purpose of "energy surety" (Department of the Army, 2012).

A technical report written by the National Renewable Energy Laboratory (NREL) highlights the usefulness of advanced meters in modernizing electrical infrastructure through Net Zero energy application on military installations. It states the recommended approach to sizing and completing a successful electrical design is to collect hourly load data at the facility or substation level. Additionally, the report recommends that five years of data be kept for all important facilities (Booth et al., 2010). This data can be collected and stored through the Air Force's AMI and AMRS.

2.7.2 Microgrids

The Department of Energy defines a microgrid as "a local energy grid with control capability, which means it can disconnect from the traditional grid and operate autonomously" (Department of Energy, 2014). Similarly, the Energy Independence and Security Act in 2007 defines a microgrid as "an integrated energy system consisting of interconnected loads and distributed energy resources (including generators and energy storage devices), which as an integrated system can operate in parallel with the utility grid or in an intentional islanding mode" (The United States Congress, 2007).

Microgrids appeared in three of 11 energy and sustainability presentations at the 2017 Society of American Military Engineers' (SAME) Joint Engineer Training and Conference (Society of American Military Engineers, 2017). The private industry has proven microgrids work and their use is broad and very applicable to the military (Department of Defense, 2015c; Morgan, Valentine, Blomberg, Limpaecher, & Dydek, 2016; Van Broekhoven, Judson, Nguyen, & Ross, 2012). In a study of Urban Microgrids, MITLL highlights 11 facilities in New York City that were able to continue operation during Superstorm Sandy due to their microgrids. They state, "Microgrids can be leveraged to maintain normalcy during major catastrophes" (Morgan et al., 2016). This normalcy is critical to military operations, especially during times of major catastrophes and during sudden electrical grid failure.

The level of reliability available from microgrids fits in well with the DoD's Operational Energy Strategy Objective of "Enhancing current mission effectiveness" where the priority is given to initiatives that improve "robustness and flexibility of the energy supply chain" (Office of the Assistant Secretary of Defense for Energy Installations and Environment, 2016a). This same operation strategy is mirrored in the DoD's use of the Strategic Sustainability Performance Plan (SSPP) for the department's management of infrastructure. The SSPP guides fiscally minded DoD-wide decisions to ensure the "continued availability of critical resources" including facility energy (Department of Defense, 2016b). Inadvertently the DoD has pointed towards microgrids as a possible solution in their strategic documents.

To help reduce the confusion and direct the excitement about microgrids, the Naval Facilities Engineering Command (NAVFAC) developed the *Microgrid Design and Reference Guide* (CH2MHILL & Clark-Nexsen, 2016). This document provides a guide for Navy engineers to follow to implement and construct microgrids on their installations. A key requirement in the design of a microgrid is identifying requirements and gathering background information. Specifically highlighted in the guide is the use of advanced meters data to understand the site electrical load including time-based full, peak, average and minimized load (CH2MHILL & Clark-Nexsen, 2016). The IEEE Standard 1547.4, *Guide for Design, Operation, and Integration of Distributed Resource Island Systems with Electric Power Systems*, agrees with the NAVFAC guide in the load considerations section and the call for better energy usage data. It states that when sizing a distributed resource island system, like a microgrid, having historical demand profiles are important for times throughout the day, week or season (Institute of Electrical and Electronic

Engineers, 2011). This data can be collected and stored through the use of advanced meters.

The U.S. Department of Energy worked in tandem with many other nations in Europe and Asia to help progress microgrid technology. Eight of the nine microgrid projects in Japan between 2003 and 2013 have advanced meters (Ton & Smith, 2012) MITLL completed a technical report for the DoD on the application of microgrids on military installations. The report highlights the work of Naval Support Facility Dahlgren by implementing a three phase approach to installing a microgrid; the first phase being data collection with advanced meters (Van Broekhoven et al., 2012). With data from advanced meters, a more informed decision can be made about what specific equipment and facility upgrades can be cost effective and impactful.

Van Broekhoven et al. also point out that understating the existing infrastructure is vital for successfully implementing a microgrid on a military installation. They state, "In developing microgrid architectures, the DoD needs to be cognizant of the legacy infrastructure on each installation" (Van Broekhoven et al., 2012). This understanding and information could come from objective advanced meter data. As seen with Net-Zero Buildings and Smart Grids, meters play a large role in defining the bounds of a successful project or program (Booth et al., 2010; Software Engineering Institute, 2011).

2.7.3 Smart Grid

Smart grid application is another major electrical distribution modernization that can be completed on an Air Force installation. A smart grid is a grid that uses "digital technology for communications, monitoring (e.g., sensing), computation, and control" (Software Engineering Institute, 2011). European policy experts point to technologies to reduce energy usage and smart grids were identified as one of the top five technologies for accomplishing this reduction in energy usage (Obrecht & Denac, 2016). The Department of Energy funded the Software Engineering Institute (SEI) to create guidance on the creation of smart grids (Software Engineering Institute, 2011). The SEI state that monitoring (e.g., sensing with advanced meters) is one of the four pillars of a smart grid. Without monitoring there is no control or feedback to how the smart grid is performing.

2.8 Conclusion

The Air Force continues to maintain operations on its installations through the funding of infrastructure projects to ensure uninterrupted ability to project military power. Electrical outages are a huge concern of the U.S. Congress; they are looking for anything that can be done to reduce them. Advanced meters are currently being deployed around the Air Force and their effectiveness is only beginning to be realized. Their ability to positively affect asset management decisions is a resource the Air Force has yet to fully explore. The current cutting-edge electrical distribution upgrades require advanced meters to be successful. After investigating these topics, advanced meters can be seen as a vital tool to increase the resilience, availability and reliability of electrical power to Air Force missions.

III. Methodology

3.1 Introduction

Existing datasets were collected and analyzed to best answer the research questions proposed in Chapter 1. These datasets were obtained through private and public organizations. The data obtained was energy meter data for homes in Austin, Texas (referred to as the Pecan Street data) and facility energy usage on Ellsworth Air Force Base (AFB) in South Dakota. Additionally, the Integrated Priority Lists (IPL), a project priority spreadsheet, from Fiscal Years (FY) 17 and 18 were obtained from the Air Force Civil Engineer Center (AFCEC). Data was cleaned and analyzed using R, R Studio, and Matlab. Programing codes were created by the researcher and obtained from the Massachusetts Institute of Technology Lincoln Laboratory (MITLL). Computing was done on the researcher's personal computer and government machines. This chapter will outline the methodology and research approach.

3.2 Data Resource

The two sources of energy usage data used in this analysis are from Pecan Street and the U.S. Air Force. Pecan Street is a consortium from the University of Texas at Austin which focuses on the reduction of energy and water usage. They claim that in order to reduce energy and water usage, the information about how that energy is used is critical. Pecan Street has over 1,300 volunteers that provide their residential energy usage (Pecan Street, 2017). Data collection began in 2012 and has continued from that point. Each volunteer has the ability to store up to 50 different recorder sources (for example refrigerator, microwave, receptacles, etc.) of energy usage down to minute by minute time intervals. Total energy usage was evaluated in this thesis. This data was analyzed for patterns of electrical outages and used for sensitivity analysis of different data characteristics.

Other data was received from many resources around the Air Force. First, the IPLs for FY 17 and FY18 were obtained through discussions with AFCEC. The IPL is a project priority listing of all of the FSRM projects (facility upgrades) required by the Air Force (Mills et al., 2017; Ramos, 2017). The projects are ranked and funded according to condition and significance, as described in Chapter 2. A greater understanding of how electrical distribution projects have historically scored in relation to other facility upgrades was obtained by reviewing the ILP. This review was placed in Section 2.4.1.

Ellsworth AFB in South Dakota has historical energy usage data of specific buildings on their installation. This data was collected by electronic means with a one-day interval. Accompanying the meter data was Geographic Information System (GIS) data and real property data that describes the location and equipment characteristics. This Air Force data was also evaluated against the Pecan Street data to validate its findings.

3.3 Data Collection and Validation

Data collection was accomplished through personal contact and through internet file sharing. Meter data from Ellsworth AFB was collected through smart meters and stored in an Excel spreadsheet format. GIS data was created by the Ellsworth Civil Engineers and is readable with ArcGIS, the Air Force GIS standard software. This data was received through the use of the Aviation and Missile Research, Development, and Engineering Center Safe Access File Exchange (AMRDEC SAFE) system. Additional information and contacts were obtained via email. Detailed IPL spreadsheets were received through the Air Force Civil Engineer Portal SharePoint after gaining access from AFCEC personnel. Energy usage data from Pecan Street was received through the "Dataport" on the Pecan Street website and a Secure File Transfer Protocol (SFTP) setup directly with the University of Texas at Austin Database Manager through the computer program FileZilla. Access was granted after proof was given that the researcher was a student. Whether through email or online systems, existing raw datasets were collected and then validated prior to analysis.

Techniques for evaluating the data followed past research. Olive used meter data from Ellsworth and other Air Force bases to attempt to find correlations in building characteristics and energy usage (Olive, 2017). Olive's methodology of reading meter data was applied to avoid errors while performing analysis of the data. In particular, Olive's strategy for avoiding errors when meters "roll over," data collection interval width manipulation, and removal of erroneous data was used (Olive, 2017). Roll over errors occur if a meter has reached its limit of reportable integers and starts over counting at zero. This only occurs if the meter is reporting cumulative energy usage. Minimal roll over issues are expected with advanced meters, but could arise depending on the manufacture specification. No roll over errors were experienced with Pecan Street data because the dataset reports exact energy usage and not cumulative energy usage. Next, Olive manipulated data collection interval widths of Ellsworth AFB data from daily intervals to monthly intervals by simply adding intervals together to reach the desired length. Doing so she was able to compare the data with Tinker AFB which only had monthly intervals (Olive, 2017). Finally, the removal of erroneous data was required because advanced meters do eventually fail, as proved by Golnas (Golnas, 2012). Clean data means that analysis of outages and sensitivity analysis can be trusted with more certainty.

Pecan Street data was used and is applicable to the purposes of this thesis. Pecan Street data is very valuable because of its extensive size, interconnected relationship and extremely small data collection interval. The data's applicability to the Air Force as a whole may be questioned because the datasets are of residential energy usage. While it is true that the usage, scale, and distribution are vastly different from industrial or military energy usage, Pecan Street is a real-world, naturally-occurring dataset that can be analyzed against itself. Additionally, power companies supply electricity to Air Force bases and residential customers from the same power plants and similar distribution networks. This means if an outage occurs on-base, depending on the system, off-base customers can feel the same effects. For these reasons, residential data was extensively used in this thesis.

3.4 Data Analysis Methods

The analysis of the data relies on the understanding of two key terms: combination and interval (see Table 2 for their definition and examples). Sensitivity analysis was conducted by altering these two factors through the use of computer simulations and not through data collection. The combination was varied by changing the size of the groupings that were used to find the mean energy usage. The combination of data was not organic or specified by the dataset, but rather it was user defined. The data collection interval was varied by amassing the dataset together which was given in one-minute intervals.

Table 2. Data Analysis Key Terms

Term	Definition	Example 1	Example 2
Combination	The grouping of energy usage measured by a meter from the same time period for a specified number of weeks for the purpose of finding a mean energy usage for the given period	Two weeks' worth of energy usage from 1300hr to 1310hrs, with a mean energy usage of 2.3 kWh	13 weeks' (or ~four months') worth of energy usage from 0830hrs to 900hrs, with a mean energy usage of 7 kWh
Interval	The width of time that a meter collects energy usage data prior to reporting to a data storage system	1 minute intervals of energy usage for 01 Aug 2011 to 31 Dec 2013	15 minute intervals of energy usage for 15 Oct 2013 to 01 Jul 2015

Sensitivity analysis is, according to Saltelli et al., “The study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input” (Saltelli et al., 2008). For this analysis the “model inputs” were controlled by varying combination and interval. The output measured was detection of outages with the associated errors. The effects of having smaller and larger data collection intervals were evaluated against the 15-minute DoD standard for advanced meters (Air Force Civil Engineer Center, 2017d). The data collection interval was varied from one minute (the smallest interval available through Pecan Street) to 1440 minutes (or 1 day/24 hours) and evaluated all 36 mathematical factors of 1440. Alternatively, this concept can be thought of as varying the frequency of the data collection from once a minute to once a day. This data was amassed, following Olive's procedures, using the one-minute interval dataset and adding subsequent intervals to reach the desired interval width. This data was then analyzed for outages or malfunctions in the meter—periods with no usage reported as a "0" in Pecan Street—and checked against known outages or malfunctions. Each individual energy usage was measured against its specific mean energy usage to define a standardized distance, or standard deviation, from the mean.

The analysis of outages used specified combinations of data to find a mean energy usage. These groupings were 1, 2, 4, 13, 26 and 52 weeks (or the mathematical factors of 52). For combination of 13 and 26 weeks, seasonal effects to energy usage can be seen. To help explain this variance, these combinations begin on the 15th of April or a 3-month multiple of the 15th of April (e.i. 15th of July, October, and January). Using the National Oceanic and Atmospheric Association's published 30-year mean outdoor air temperature, the 15th of April was chosen because it is the midpoint between January and July, the coldest and hottest months, respectively (National Oceanic and Atmospheric Association, 2018). See Figure 5 for a visualization of the 30-year mean outdoor air temperature in the United States.

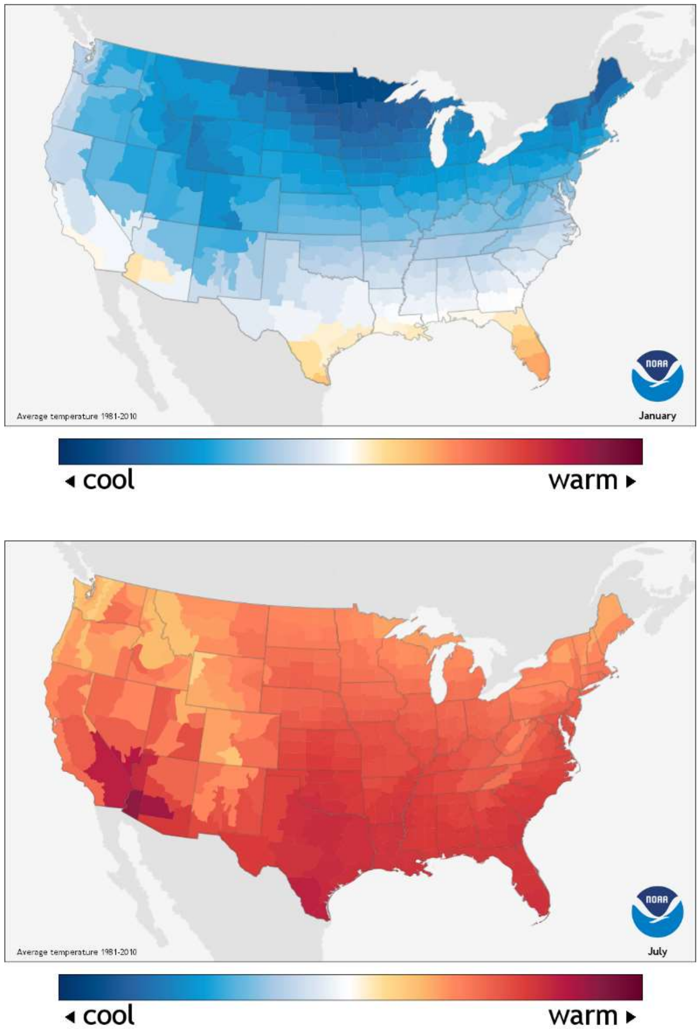


Figure 5. 30-Year Mean Outdoor Air Temperature (National Oceanic and Atmospheric Association, 2018)

Outage detection was accomplished by using combinations of energy usage for a specific time of year and specific time period in the day. Each individual member of the combination was then measured against the mean energy usage for that combination to create a standardized distance from the mean. The standardized distance for known outages/malfunctions were saved for each home in the dataset. This can be done only when outages are known. Without knowing when the outages occur, one could simply look for zeros in the dataset. However, relying solely on counting reports of zero energy usage to detect outages is insufficient. This is because outages can happen within the data collection interval, thus creating intervals with reduced energy usage. This concept is illustrated in Figure 6. The "Smallest" interval is the most accurate and robust. The "Small" interval is twice the size of the "Smallest" interval, and so on. It can be seen the "Largest" interval shows no visible signs of an outage. By applying these statistics, the outages may be calculated. The larger the distance from that specific time of year's mean energy usage, the more likely that an outage occurred.

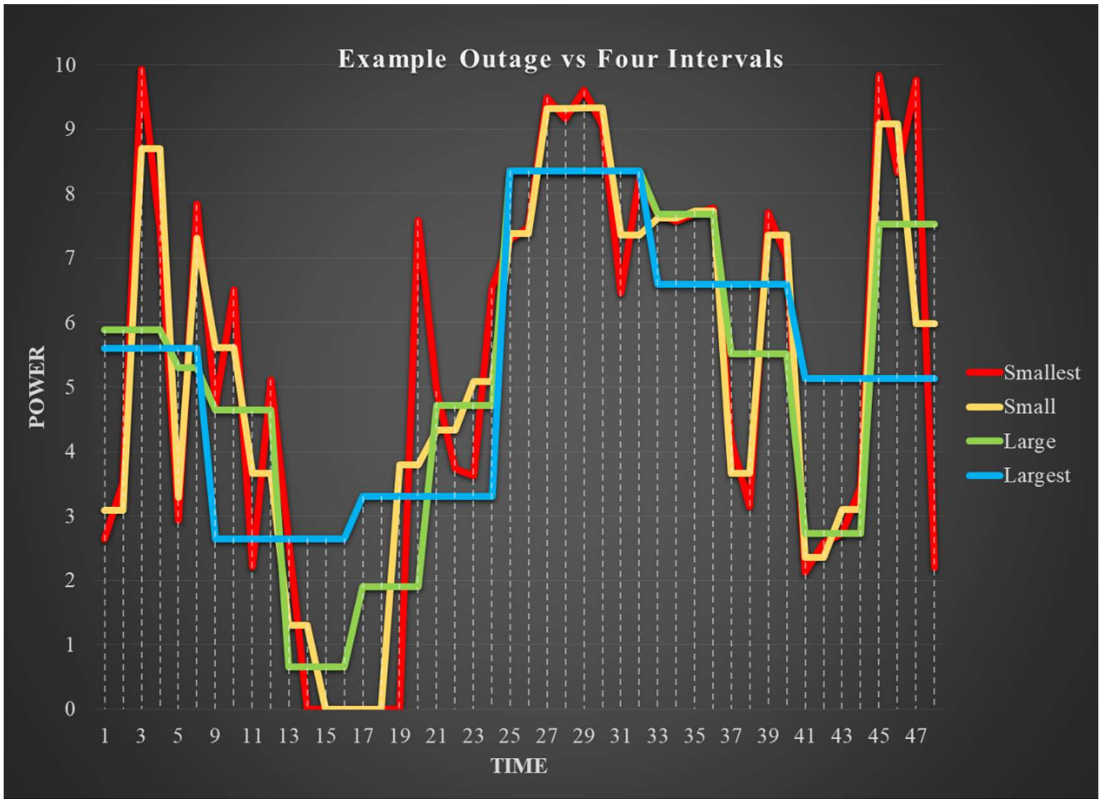


Figure 6. Example Outage vs Four Intervals

Outages and malfunctions were assumed to be any reported energy usage of less than or equal to zero. It is key to note that no verified outage information was obtained and the existence of the outage was assumed. Outages and malfunctions lasting longer than one minute were found by counting the zero and negative values in the one minute by one minute raw dataset. By using a dataset with these "known" outages and pinpointing their location, an ideal distance from the mean can be estimated by increasing the interval the data to mask those zeros/outages. This ideal distance, defined in the rest of this thesis as the ideal critical value, is found by sorting data as seen in Figure 7. First, critical values are determined for each individual Data ID. Then, the ideal critical value is chosen from the 88 critical values for each combination and interval. Once the ideal critical values are determined, they can then be tested on a virgin dataset to evaluate its effectiveness at identifying outages. Additionally, the number of weeks used in the combination to find the mean energy usage was also varied to ascertain the effects of grouping the energy usage differently.

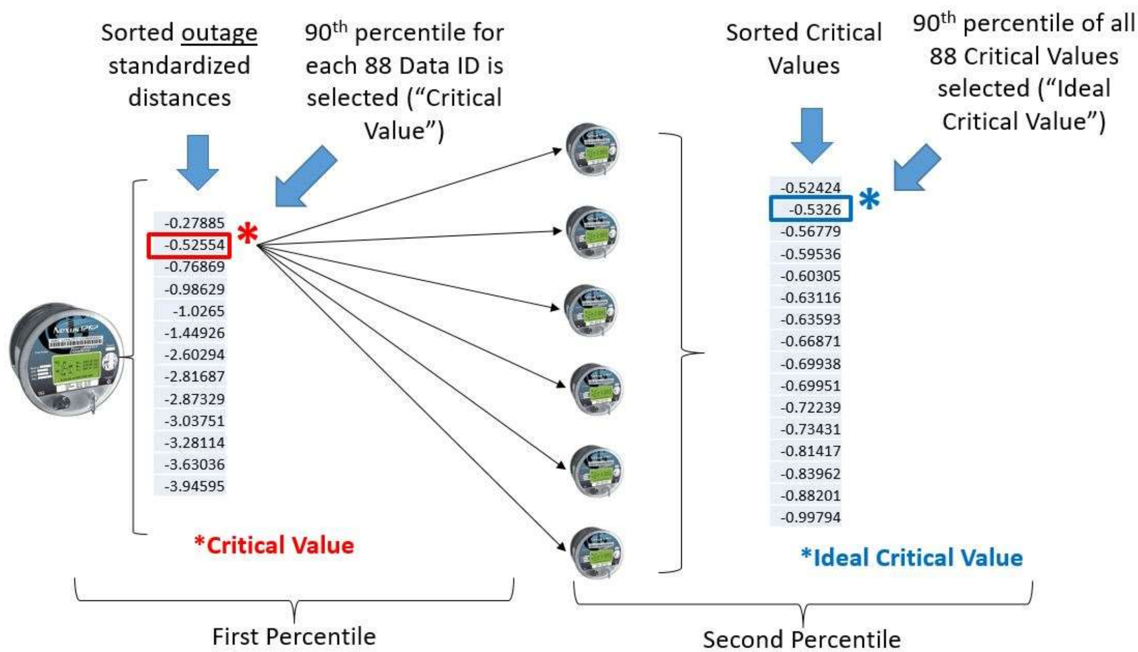


Figure 7. Infographic of How the Ideal Critical Value is Determined

This Data Analysis is concentrated on confirming the hypothesis provided below for each data collection interval:

H_o = No outage or malfunction occurred during the measured interval

H_a = An outage or malfunction occurred during the measured interval

If the standardized distance from the mean is below the ideal critical value, as defined above, then the null hypothesis is rejected in favor of the alternative. Therefore, this hypothesis is used to improve the effectiveness outage detection.

Inherently, by using hypothesis testing, there arises the opportunity for errors. A Type I Error occur when the null hypothesis is rejected in favor of the alternative hypothesis when the null hypothesis was correct all along. Conversely, Type II Error occurs when the null hypothesis is accepted as truth, when really it should have been rejected in favor of the alternative hypothesis. For this analysis a Type I Error, known as a False Positive, is where the model states there was an outage when really there wasn't one. A Type II Error, known as a False Negative, is where the model states that no outage has occurred when really there was one. Figure 8 gives a visual breakdown of each situation possible for the hypothesis testing.

		Identified as	
		No Outage	Outage
Actual	No Outage	Good Match (True Negative)	Type I Error (False Positive)
	Outage	Type II Error (False Negative)	Good Match (True Positive)

Figure 8. Error Definitions

When measuring outage detection error, there is a distinction between an "identified" and "known" outage. Identified outages are those intervals whose standardized distances, or standard deviations, from the mean energy usage are less than that of the ideal critical value. Identified outages contain all Type I Errors and all true identified outages (the right two quadrants of Figure 8). Known outages are only "known" because they are located in the model by highlighting zeros in the dataset prior to increasing data intervals. Known outages include Type II Errors and true identified outages (the bottom two quadrants of Figure 8).

In the real world Type I Errors would make an operator believe that their system is performing worse than it really is. If they act upon information with Type I Errors they could be wasting money on a problem that doesn't exist. Type I Errors in this analysis are measured against identified outage to give a confidence of how many outages that were identified were actually true outages.

Conversely, Type II Errors effect how confident an operator is that they have captured all of the true outages. High Type II Errors mean an operator is not confident that all the outages have been captured or identified. Type II Errors in this analysis are measured against known outages to illustrate how many outages are not being identified. In the real world, the existence of a Type II Error must be discovered or reported by some external means. This would mean users reporting outages and operators verifying that outages have occurred; a process that has many opportunities to fail. Also, electrical infrastructure has the ability to reset itself with reclosers, taking up to three minutes to operate (Pereira, Pérez-yauli, & Quilumba, 2017). These momentary outages could be lost if users do not detect and report them. Additionally, for single facility outages, users could reset breakers if they have access to them and the information could be lost or not communicated that an outage had occurred.

A Design of Experiments (DOE) gives the analysis validity. The DOE began by using 20 percent of all of the possible homes, or 88 houses, in Austin, TX from the Pecan Street dataset. These 88 homes were selected using a random number generator in excel. These homes amounted to 480 million data points. R, with R Studio, was used to clean and analyze the data. An additional 5 percent, or 22 homes, of the Pecan Street homes in Austin, TX were also collected and used to test the ideal critical values that were found by using the other 88 homes.

This ideal critical value was also tested against the Ellsworth AFB energy usage data. Data collected from Ellsworth AFB was given in cumulative energy usage. This means energy usage was calculated by finding the difference from the previous energy usage. Additionally, roll over error effected two of the 87 electrical meters reported. Because Ellsworth AFB is located in South Dakota and the ideal critical values were created using data from Texas, it is unclear if the ideal critical values will be effective. Another issue is that the Ellsworth AFB data was only given in daily energy usage intervals, making locating outages difficult and limiting validation of outages to only outages lasting longer than 24 hours. Error information for locating outages was also collected for facilities that contained outages (zeros in the data).

Figure 9 shows the trade space for the sensitivity analysis of the data interval. The data becomes more robust and useful as the interval decreases, but not without a cost. Data storage and handling, including transfer, becomes more difficult as the interval decreases. Additionally, meters that can handle higher amounts of energy are more expensive (Carnley, 2017). Less data handling and storage can be accomplished with larger intervals, thus reducing the required server sizes and saving money on acquisition costs, operations and maintenance of those servers. With larger intervals, the granularity of the data is lost, thus making it less useful for understanding minute variations. Aggregation of multiple facilities under a single meter was not looked at in this analysis, but it is also a key factor that effects the cost of the meter. Currently, as described in Chapter 2, the AMRS uses 15-minute intervals and is used only for single facilities or single transformers (when a facility has more than one transformer).

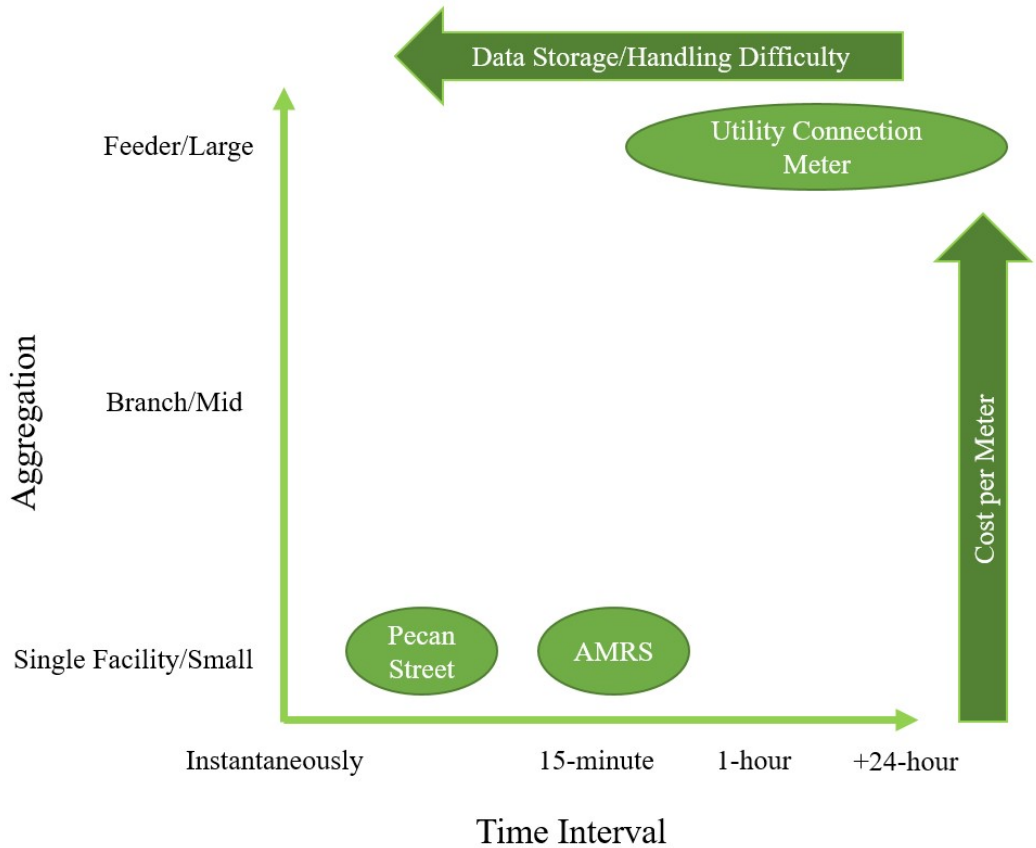


Figure 9. Meter Configuration

Once the above methods were used, an in-depth analysis of how to select the ideal critical values was conducted. In the initial analysis, the 90th percentile of all the outages' standardized distances were used to select the critical value. This concept is illustrated on the *left* half of Figure 7, referred to as the "First Percentile." In this sensitivity analysis, the selection of the critical values were evaluated 10 different times for the 95th, 85th … 15th, and 5th percentiles for each combination's outage standardized distances. In the initial analysis, the 90th percentile was also used to calculate *ideal* critical value from the critical values, as seen in the *right* half of Figure 7, referred to as the "Second

Percentile." This percentile was also altered in this sensitivity analysis in order to calculate the *ideal* critical value. Again, the selection of the *ideal* critical values were evaluated 10 different times for the 95th, 85th … 15th, and 5th percentiles for each combination's critical value. All possible combinations were explored; a total of 100 different ways to choose the *ideal* critical value. Each of these 100 percentile combinations contain error information for locating outages. As the percentiles go down, Type II Errors will increase and Type I Errors will decrease. This will create a risk decision for which error to minimize. Theoretically, the 95th percentile will find almost all of the outages, but it will also increase the number of incorrectly identified intervals as outages (Type I Errors). Conversely, the 5th percentile will find less real outages (increase Type II Errors) but it will also decrease the number of incorrectly identified intervals as outages. There was a percentile with the least number of total errors (Type I and Type II Errors combine), but the specific type of error must not be overlooked.

An additional sensitivity analysis was conducted on the Matlab tool developed by the Massachusetts Institute of Technology Lincoln Laboratory (MITLL) Energy Systems Group. The change in recommended electrical system architectures were recorded and analyzed while varying the number of reported outages for a single year by altering the System Average Interruption Frequency Index (average number of outages in a year). Additionally, the unserved load for each architecture was also analyzed for the different reported outages. MITLL stated their largest issue was valid outage data (Nick Judson & Pina, 2017). The sensitivities of the tool can be seen by simulating improved outage data or in other words the outage collection effectiveness. This was simulated by comparing the current Air Force Standard of 15-minute data collection intervals and the currently available 1-day intervals to a one-minute data collection interval. As described in Chapter 2, the Air Force only collects outages lasting longer than 8 hours. The assumption is made that the methods outlined in this thesis were used to collect outages; all outages including those less than 8 hours. By comparing the system architectures, with their accompanying amount of unserved load, the risks of using a 15-minute or 1-day interval can be seen against using a one-minute interval.

3.5 Data Processing

All figures for this section will be found in Appendix 1 because of their size. R, with R Studio, was used to process the data. There were two major phases in the programing: Model Building and Model Validation. Figure 41 shows the conceptual flow of the Model Building code. This model sets the framework for the validation of the data processing, as seen in Figure 42. Aggregation of multiple buildings on a single meter was not looked at in this thesis, but Figure 43 gives a concept of how it could be done.

All data from Pecan Street are first read into the program. The raw data has 3 columns: Time (with day, month, and year), energy usage (in units of average kW over the one minute data collection interval), and Data ID (or the specific home in the dataset) (Pecan Street Dataport, 2017). While it is true that a kW is a measure of power, it can be classified as energy usage because the time period over which the average power occurred is given.

The Model Building code takes each individual Data ID and runs the process outlined below. The data must be cleaned for errors by placing the data in chronological order, removing duplicate entries, removing "NA" values, and making all negative energy usage equal to zero, thus treating them as an outage/malfunction. The code finds the first full day of data and then replaces any chronological gaps in the data with zeros, again treating them as an outage/malfunction. The outages/malfunctions are then located by searching for zeros and their locations are saved for use later in the code. The code then runs another loop for the 36 different possible data collection intervals as described in the previous section. The data collection interval is simulated by adding the next minute's worth of data to the first minute's worth of data until the desired data interval width is reached. The code will then enter the final loop for each of the six different possible data combinations, as described in the previous section. The code groups the data by the desired combination and then finds that specific time of year's mean energy usage. Next, the code finds the standardized distance from the mean for all of the outages/ malfunctions according to its specific corresponding mean. These values are defined as "z-scores" in the code. These z-scores are then placed in descending order to locate the 90th percentile of all z-scores for that specific Data ID, interval, and combination. This value is defined as the "critical value" in the code because 90 percent of the other z-scores are at or below this value. All other values, regardless of if it was an outage/ malfunction or not, are also checked against their specific corresponding means. Outages/malfunctions can then be predicted by defining all data below or at this critical value as an outage or malfunction.

This process will repeat until all Data IDs have been evaluated. With all 88 Data IDs complete the ideal distance from the mean, defined as the "ideal critical value," can be estimated by finding the 90th percentile of all critical values by interval and combination. This ideal critical value is the ninth largest critical value of the 88 different Data IDs test (90 percent of 88 rounded up is 80). This value can theoretically locate 90 percent of the outages in any other data set. To test this hypothesis another virgin dataset was tested using these ideal critical values.

The Model Validation follows a similar process to the Model Building phase. The ideal critical values from the Model Building are the only data used in the Model Validation phase. Again, the data must be cleaned and outage location is stored in order to gauge the effectiveness of the ideal critical values. Additionally, the data collection interval and the combination are varied. All values are measured against their own specific time of year's mean energy usage to determine the standardized distance from the mean. With the ideal critical values already predetermined, the standardized distances can be placed in one of the four quadrants defined in Figure 8: good match outage (true negative), good match no outage (true positive), Type I Error (false positive), and Type II Error (false negative). This will be the true measure of the effectiveness of the ideal critical values.

The aggregation data processing could done by following both the Model Building and Model Validation exactly the same except for adding an additional step. This additional step is the aggregation of energy usage data from different homes. It is vital that the aggregation is performed using the exact same timetable to account for variation in energy usage due to weather, holidays or any other global event that could affect energy usage. Once the ideal critical values are calculated and they are validated against the virgin dataset, the effectiveness of the ideal critical values are again measured using the four quadrants defined in Figure 8.

IV. Analysis and Results

4.1 Introduction

Many in-depth topics emerged and were included after applying the research questions in Chapter 1. The analysis was complete using R, R Studio, and Matlab. Sensitivity analysis was performed to optimize solutions and expound on relevant questions. First, the basis of the analysis was the use of Pecan Street data to see if locating outages using advanced meters can be accomplished by using means and standard deviations of combined data. All other analysis builds off of these initial findings. Next, locating the outages was conducted using aggregated data (see Table 2). After that, the findings from the Pecan Street analysis were applied to Ellsworth Air Force Base (AFB) to measure applicability of the methods to Air Force data. Then, the Massachusetts Institute of Technology Lincoln Laboratory (MITLL) Tool was used to compare different data collection intervals and their effect on selecting electrical system architectures. Finally, a critical review of the electrical projects was conducted on the Integrated Priority List (IPL).

4.2 Pecan Street – Proof of Concept

Meticulous data processing was vital to ensuring the proper data was being saved and operated upon. The Pecan Street one-minute interval data was analyzed as described in Chapter 3. The model building was complete on 88 homes from Austin, TX. With six different possible combinations and 36 different possible intervals, there were 216 different "Ideal Critical Values." These ideal critical values were found by, first, taking the 90th percentile for the critical values for each Data ID, interval, and combination (known as a critical value). Next, the 90th percentile of all 88 Data IDs' critical values for each interval and combination to find the ideal critical value. These ideal critical values can be seen in Figure 10.

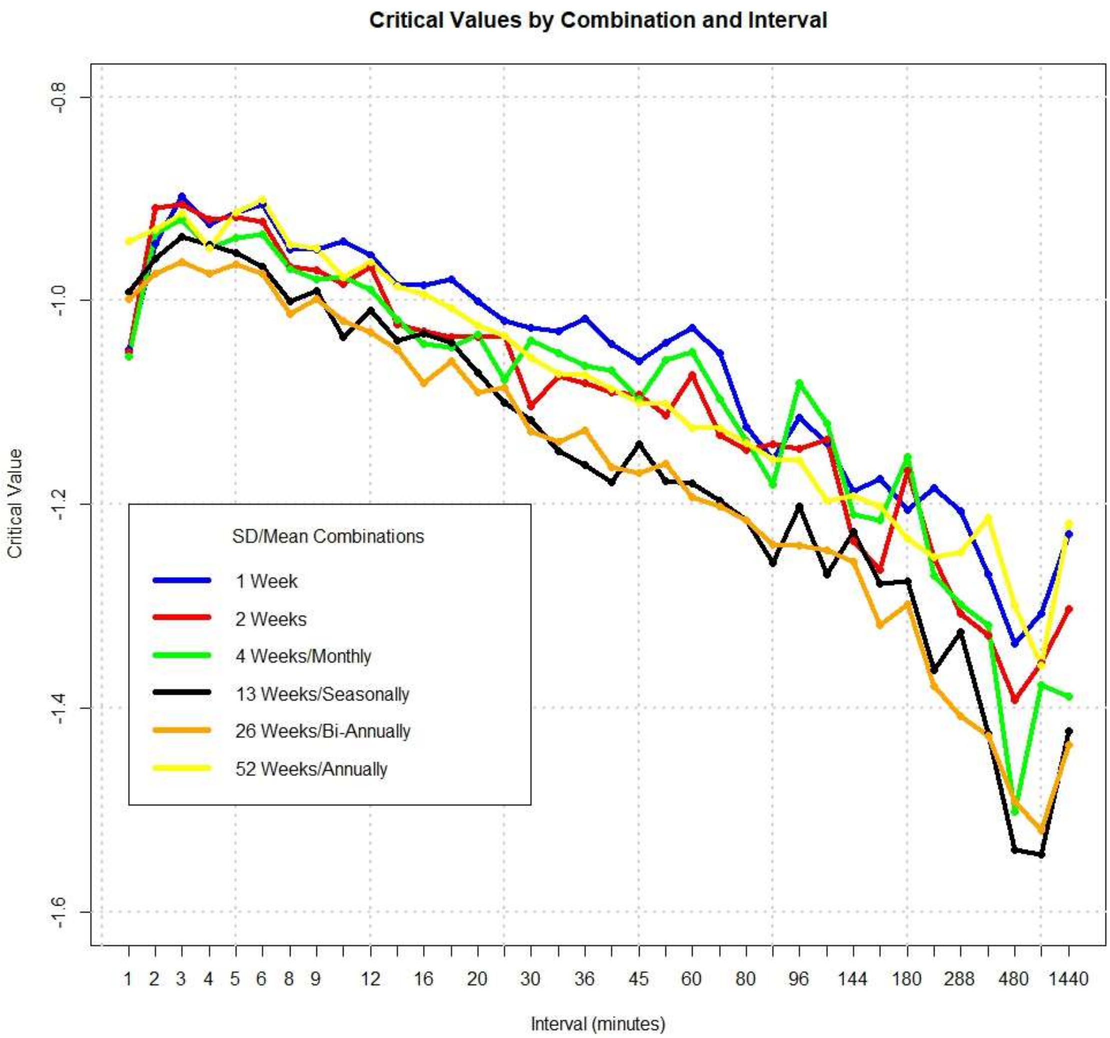


Figure 10. Ideal Critical Values for each Combination and Interval

With the model building complete, the ideal critical values can be tested against a virgin validation test dataset of 22 homes. Again for each combination and interval, the minute by minute data is transformed to the correct interval width and tested against the ideal critical value. Errors, as described in Figure 8, are calculated to examine the effectiveness of the ideal critical values at pointing out the outages. Figure 11 shows the overall effectiveness of the model. It is important to note that the x-axis is on a custom scale reporting the actual intervals used (e.i. it is not a log or normal scale). This figure says, unconditionally, we can trust the results of the model X percent of the time. The 52-week combination, or annual combination, and the 26-week combination, or bi-annual combination, clearly out perform all other combinations as seen in Figure 11. Using a combination of 52 weeks and a 12-hour interval in Austin, TX proves to be the most reliable of the six combinations and 36 intervals investigated in this part of the analysis. This particular interval and combination was correct 98.4 percent of the time. The 1.6 percent that was incorrect can be accounted to either Type I or Type II Errors.

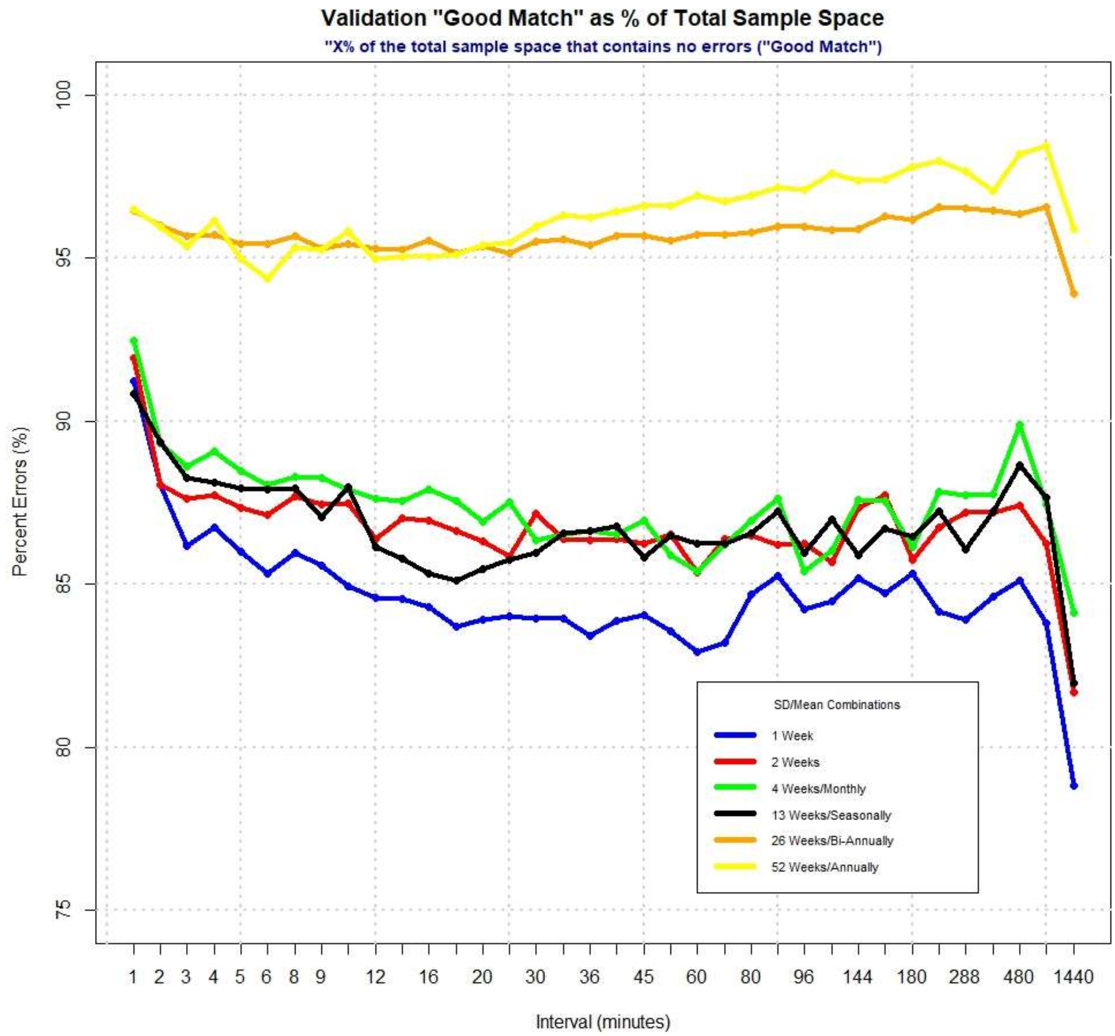


Figure 11. "Good Match" vs. Total Sample Space for each Combination and Interval

Further analysis of the errors is required because Type I and Type II Errors have different impacts on real world outage detection and advanced meter operations. Type II Errors are known outages that had a standard deviation higher than the ideal critical value. Due to this, they are labeled as “good” intervals when really an outage has occurred (known as a false negative). Type II Errors were very small compared to Type I

Errors (compare Figure 12 to Figure 13). There is also no clear distinction between any of the combinations, but the errors increase as the interval increases.

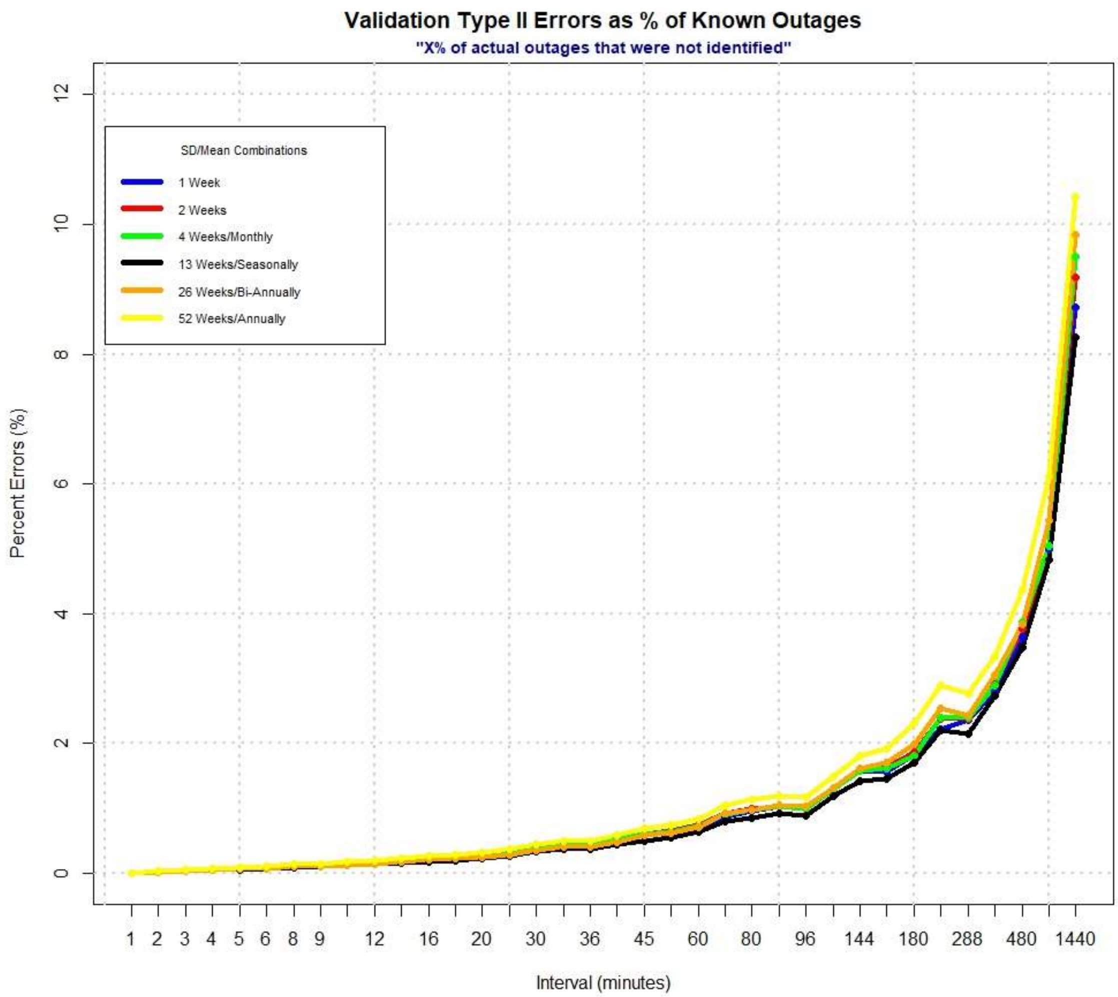


Figure 12. Type II Errors vs. Known Outages for each Combination and Interval

Type I Errors are measured against identified outages to give a confidence of how many outages that were identified are actual true outages. Unlike Type II Errors, Type I Errors have a stratification and clear best combination for most intervals as seen in Figure

13. The Type I Errors vary from interval to interval and do not have the clear pattern that the Type II Errors have.

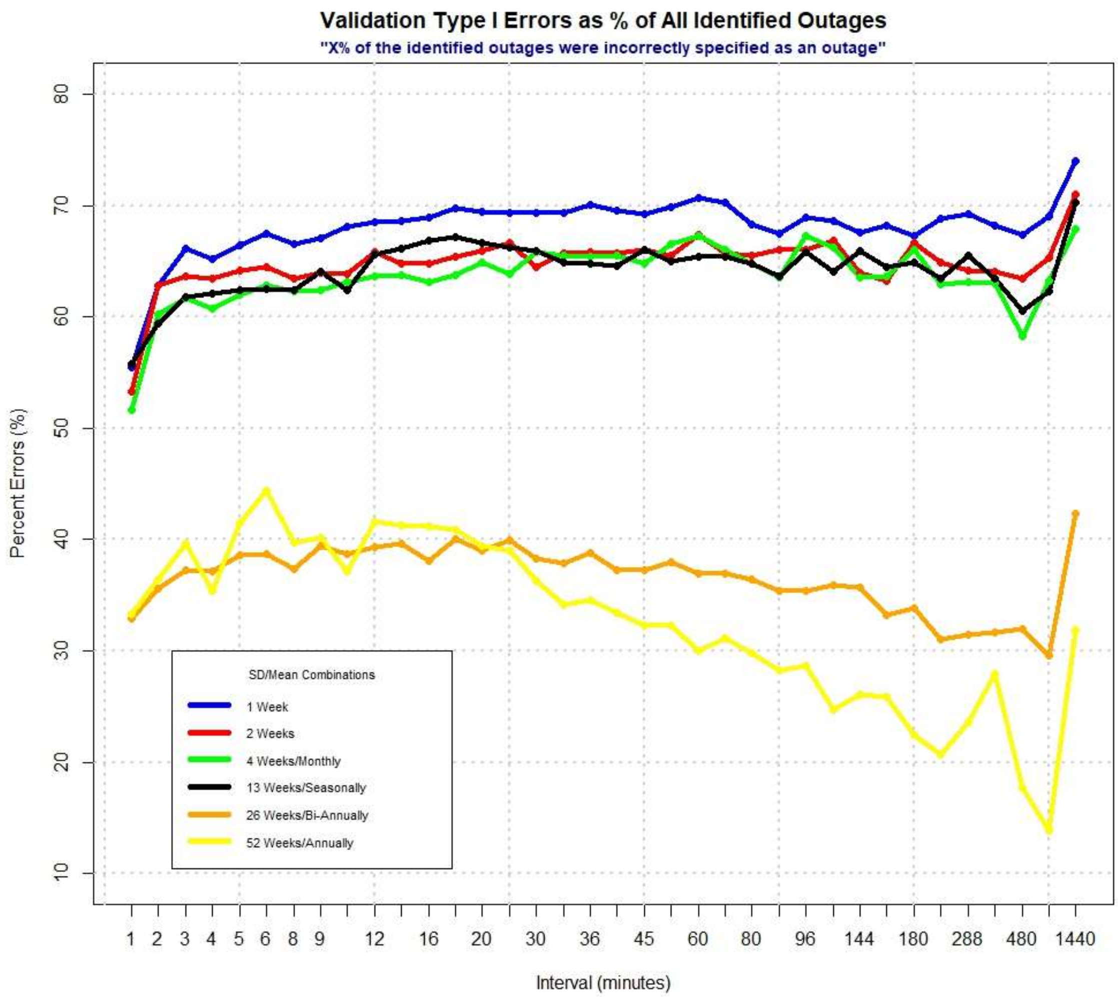


Figure 13. Type I Errors vs Identified Outages for each Combination and Interval

Each Data ID was effected differently by the ideal critical values. Figure 13 shows each individual Data ID from the virgin dataset and how many errors are found for all possible intervals. It shows that some Data IDs are not effected at all by errors while others account for a large portion of total errors. It is also important to note that both Type I and Type II Errors are graphed together and in no case are Type II Errors visible clearly showing that Type I Errors dominate in this analysis. This finding sparked another analysis for looking at the sensitivities of the ideal critical value (see section 4.3 Pecan Street Ideal Critical Value Sensitivity).

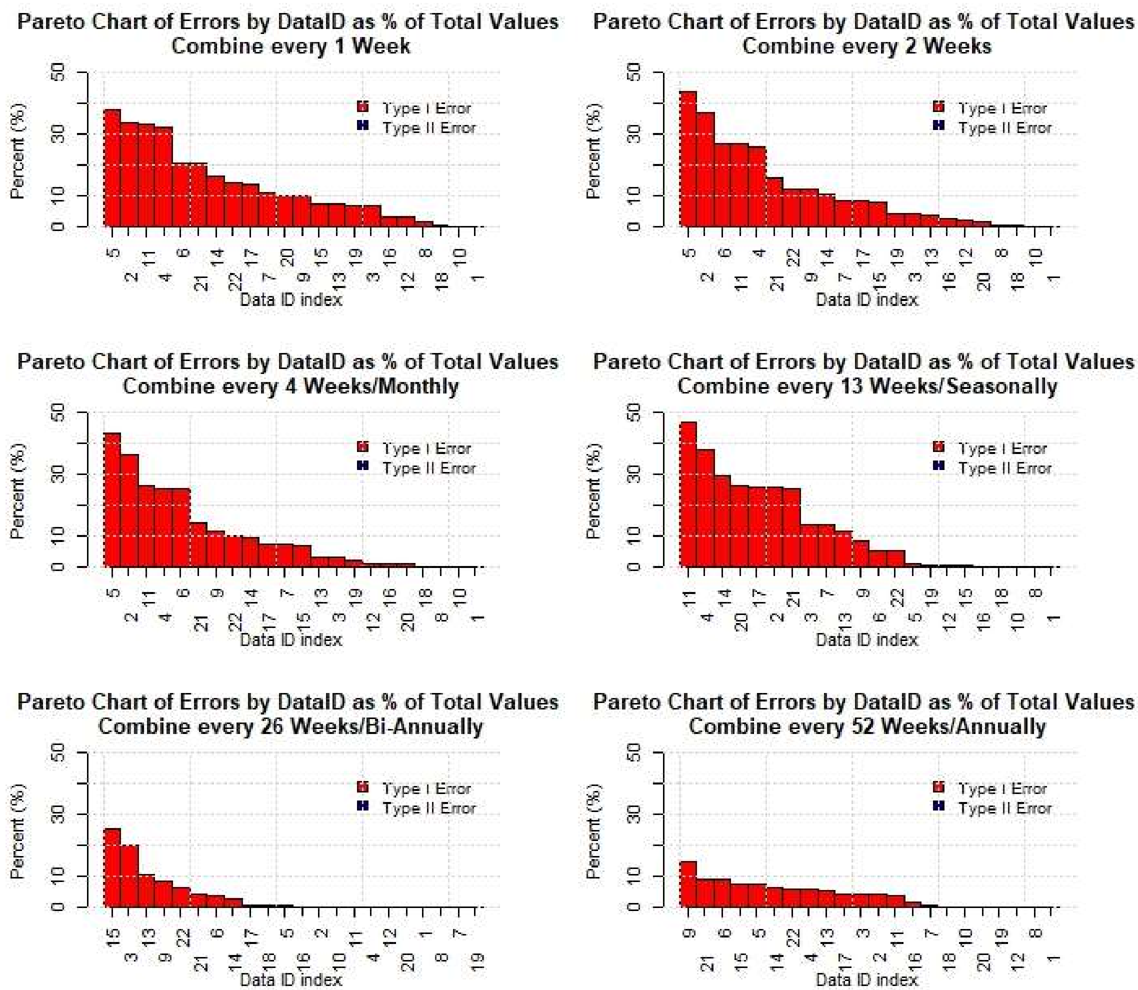


Figure 14. Pareto Chart of Errors by Data ID for all Interval

Figure 15 was used to observe if a trend exists between how many years' worth of data was used and errors in the different Data IDs. No clear trend can be seen. The Data IDs are arranged in the same order as Figure 14.

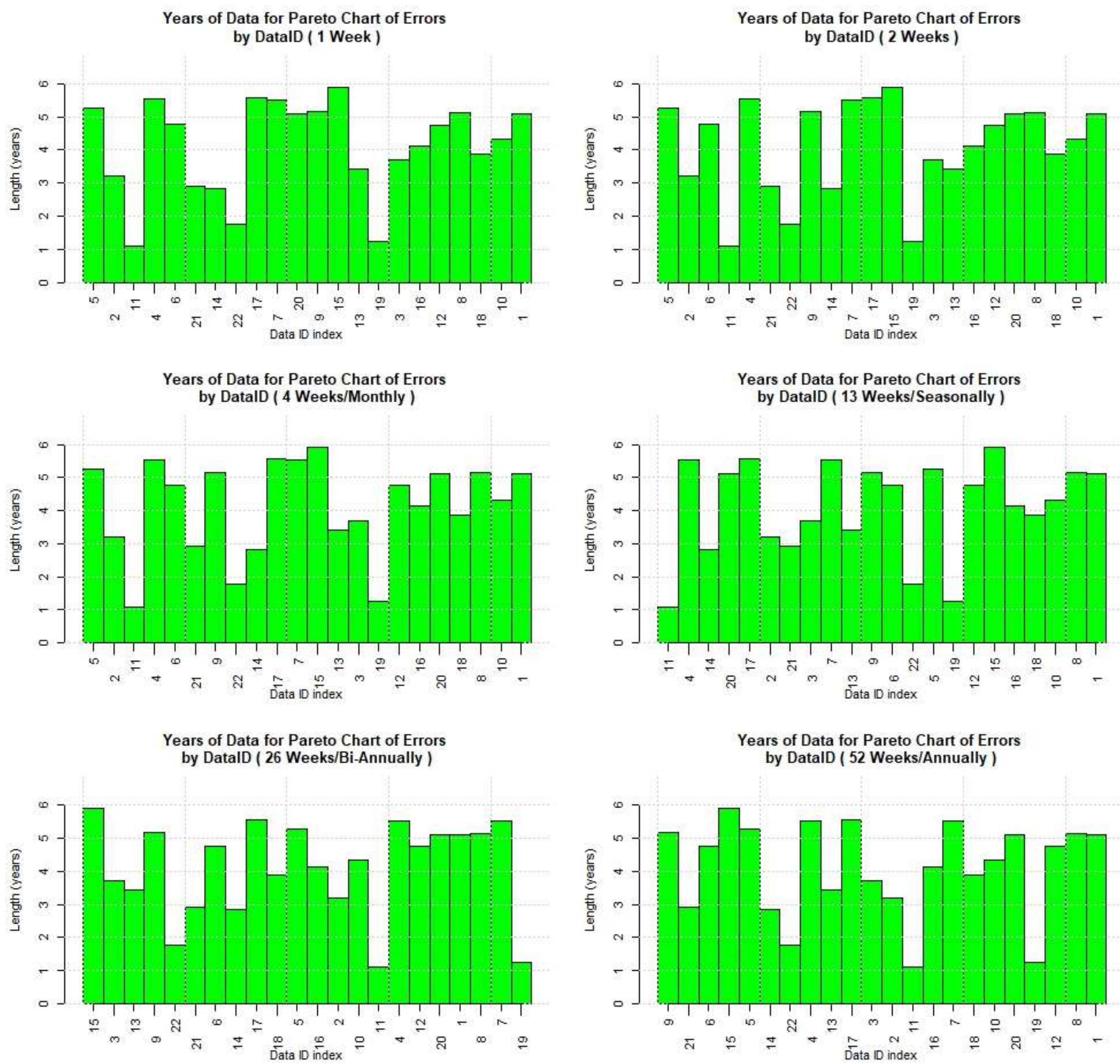


Figure 15. Amount of Data in Descending Total Error Order for all Intervals

As in Figure 15, Figure 16 was created to attempt to identify a trend between mean outage length and total errors observed errors. Again, Data IDs are arranged in the same order as Figure 14 and there appears to be no clear trend.

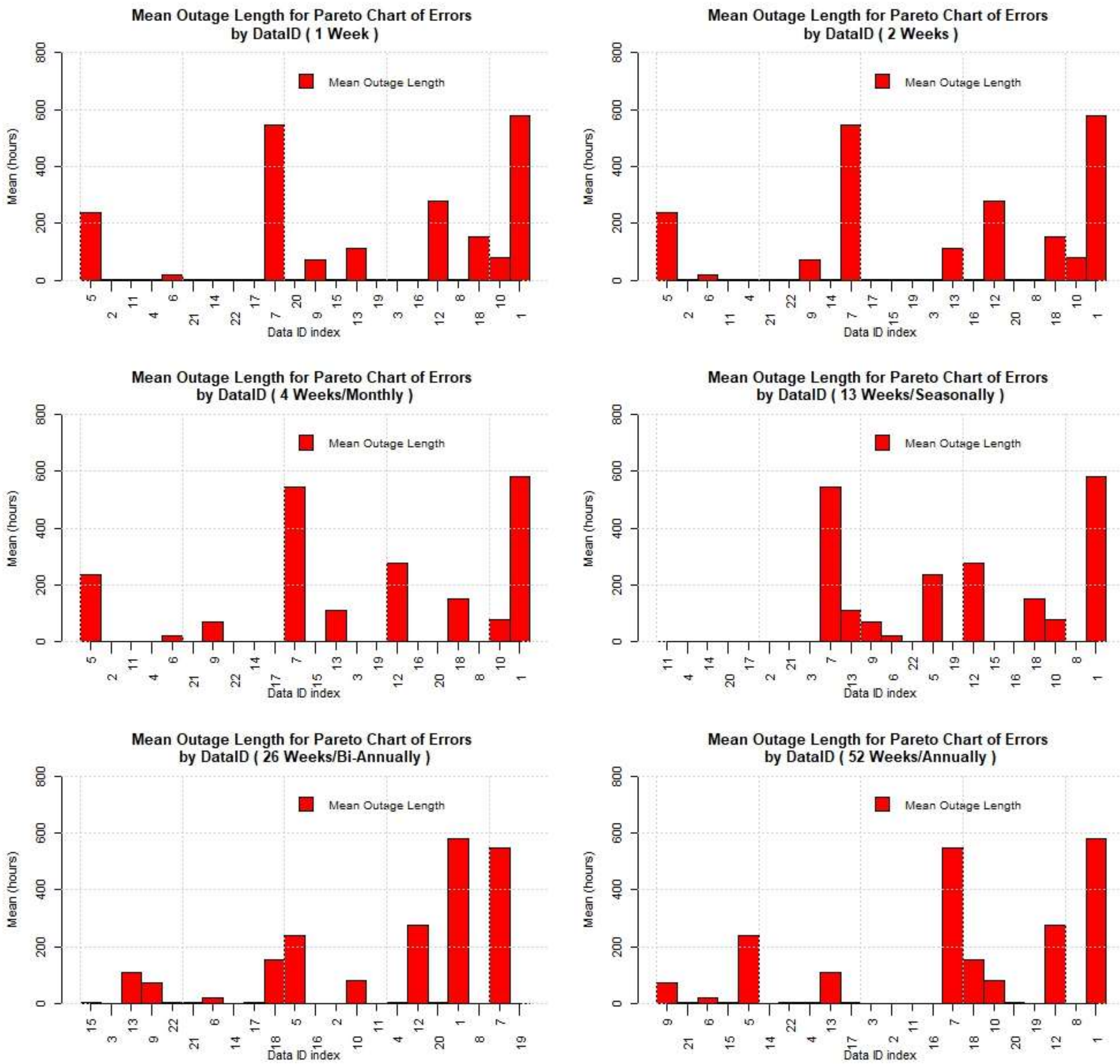


Figure 16. Mean Outage Length in Descending Total Error Order for all Intervals

The findings from this part of the analysis were that, first, the 52-week and 26-week combination, with the former being slightly better, performed the best for Type I

Errors as well as overall model effectiveness. Second, all combinations perform similarly for Type II Errors. Third, the number of Type I Errors is far greater than the number of Type II Errors. And finally there is no clear trend between mean outage length or the amount of data to the amount of errors. The R Studio raw code for this analysis can be found in Appendix 1.

4.3 Pecan Street Ideal Critical Value Sensitivity

This analysis was conducted because the ideal critical value is found using user defined parameters that were not optimized for performance, but rather just proof of concept. In all other analysis to this point, the critical value was found using the 90th first percentile and 90th second percentile. For this analysis, to calculate the individual Data ID's critical value, the standardized distances from the specific mean energy usage for each outage were ordered and the given percentile was used to locate the critical value for each Data ID (see the left half of Figure 7). Once complete, the 88 critical values for each combination and interval were ordered and the second percentile, used for finding the *ideal* critical value, was also varied (see the right half of Figure 7). This amounts to 100 different possible ways of selecting the ideal critical value. Again these ideal critical values are tested against the same virgin data set as in previous sections.

Figure 17 shows the 100 different Ideal Critical Values that were used in this section. The values represented are the average value with one standard error for each percentile. They are ordered by second percentile (right side of Figure 7) then first percentile (left side of Figure 7). This figure is a summation of all intervals and combinations (similar to Figure 10). Appendix 3 contains all the individual ideal critical value graphs for the 95th percentile of the second percentile. These graphs are represented by the first ten values of Figure 17. Appendix 3 shows how the ideal critical values change as the first percentiles changes. Appendix 4 contains all the individual ideal critical value graph for the 95th percentile of the first percentile. These graphs are represented by the top value of each group of ten as shown in Figure 17. As predicted in Chapter 3, the ideal critical values decrease as the percentile size decreases.

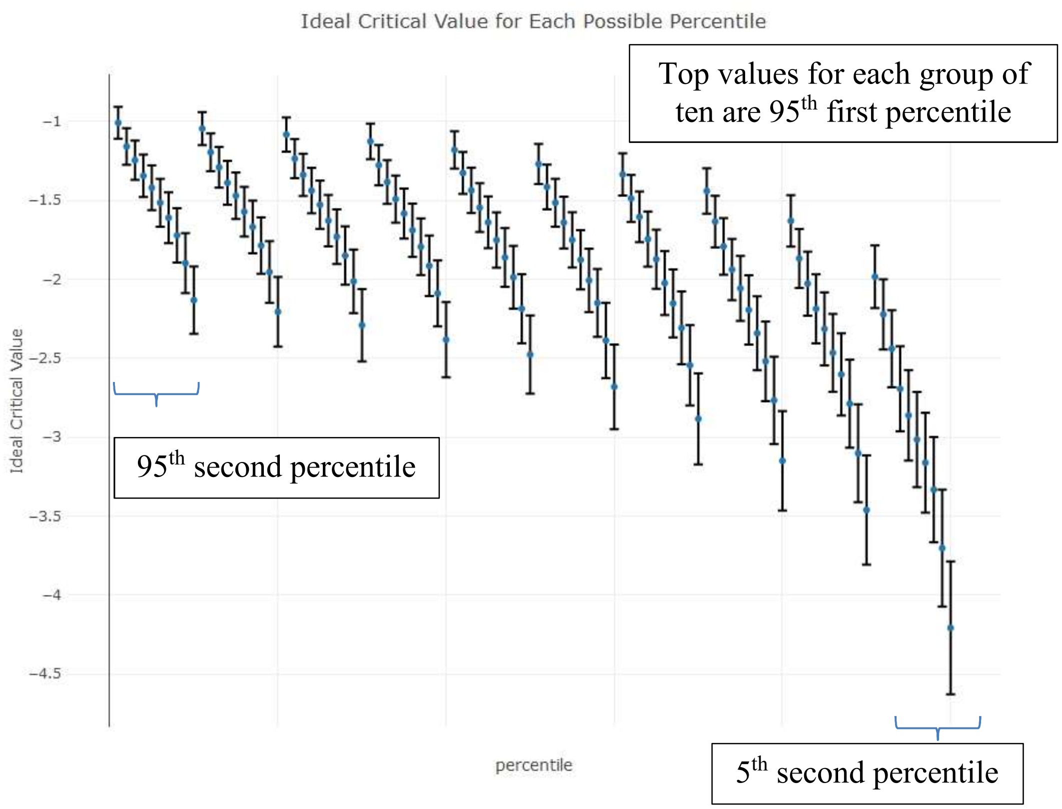


Figure 17. Ideal Critical Values for Each Percentile

Error information was gathered by using these ideal critical values to located outages in the virgin dataset. The prediction that Type I Errors increase and Type II Errors decrease as percentiles increases is proven in Figure 18 and Figure 19, respectively. These examples are the extremes of the spectrum, but they effectively show how important it is to select the correct first and second percentiles to manage the trade-off between Type I and Type II Errors.

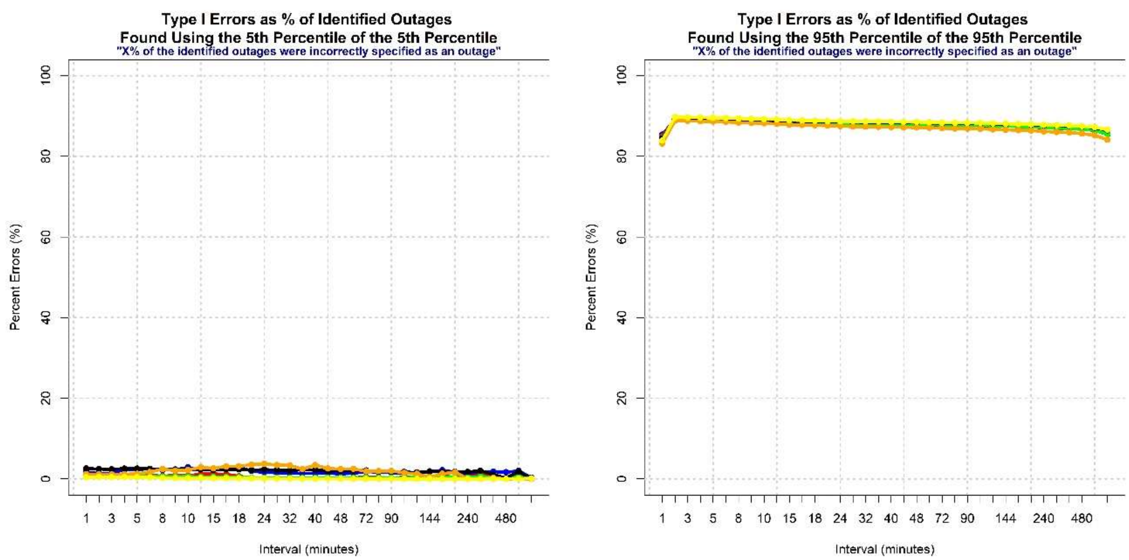


Figure 18. Type I Errors for the 5th and 95th (first and second) percentiles

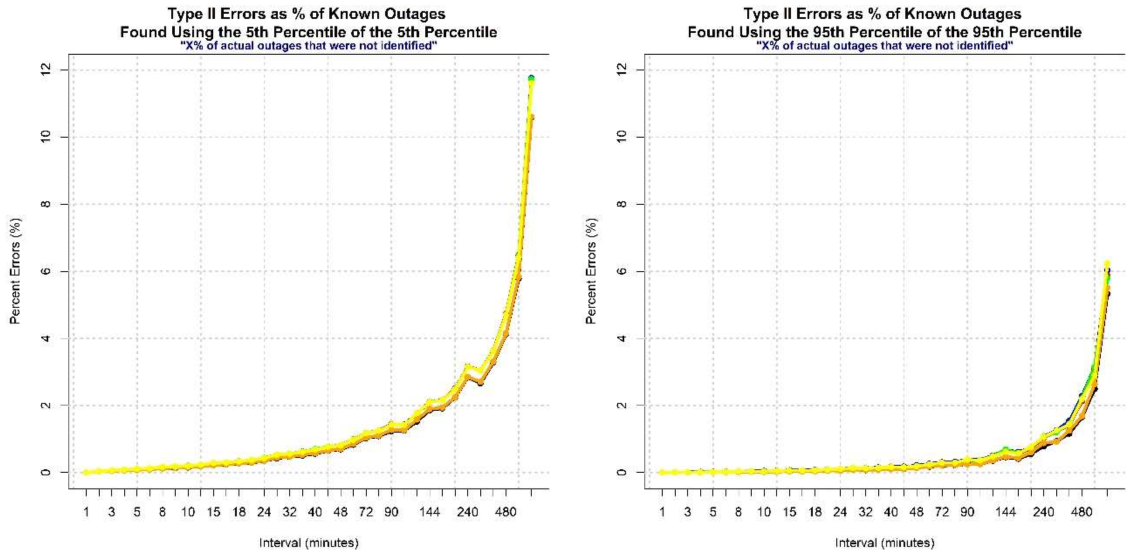


Figure 19. Type II Errors for the 5th and 95th (first and second) percentiles

The key decision now becomes finding the balance between Type I and Type II Errors. Figure 20 shows the Type I Errors for all 100 different percentile combinations used (the percentiles are in the same order as Figure 17). As the second percentile goes down (the second percentiles are separated by distinct black and gray bars in Figure 20) there is a greater stratification and variance in the amount of errors. These errors are a summation of all errors for all six combinations and 36 intervals used in this analysis. A 5^{th} first percentile and 5^{th} second percentile minimizes Type I Errors to 1.26 percent. However, this percentile combination creates the largest amount Type II Errors of 0.195 percent, as seen in Figure 21. The specific breakdown for interval and combination for this percentile combination can be seen on the left of Figure 18 and Figure 19.

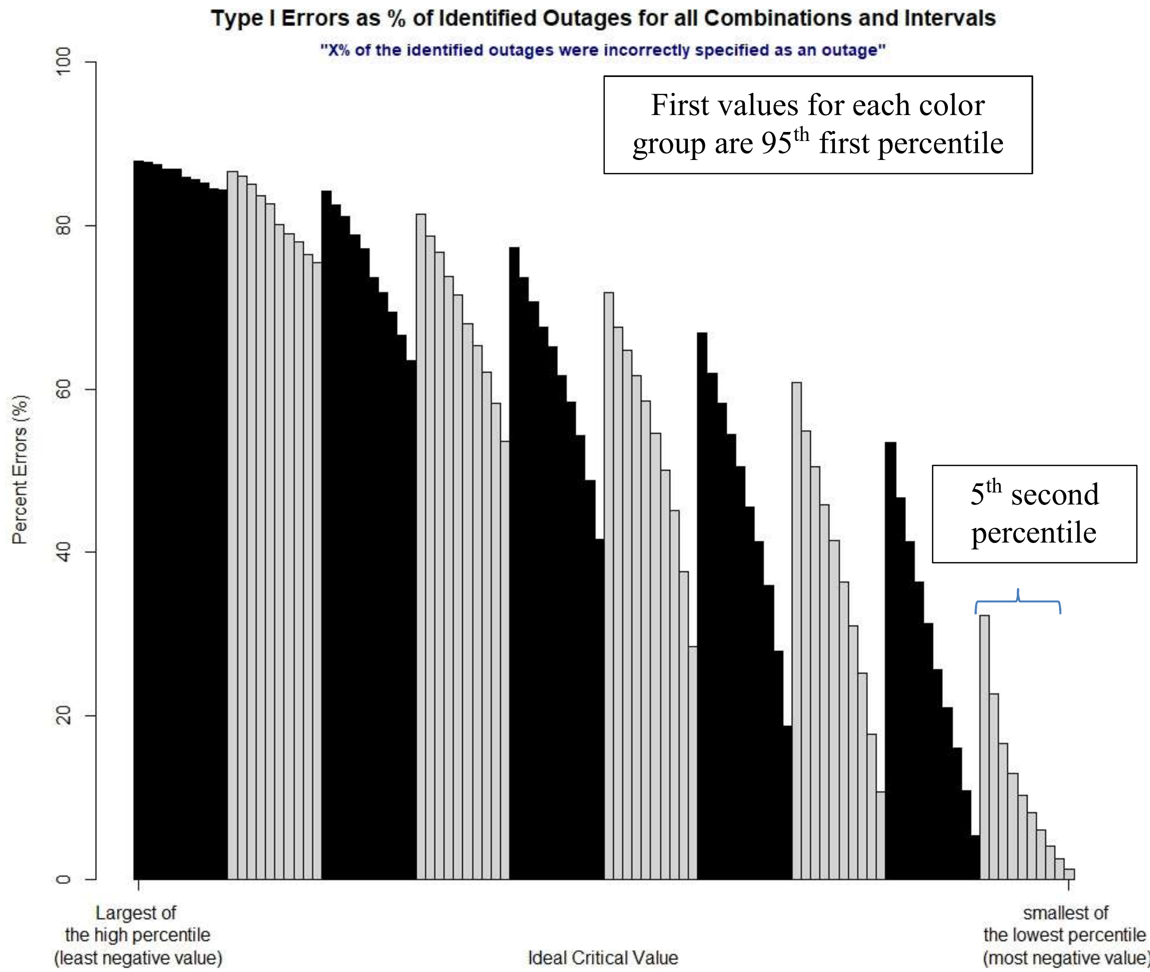


Figure 20. Type I Errors Ordered by Second then First Percentile

Similar to the Type I Errors, the Type II Errors for all 100 percentile combinations are given in Figure **21**. Again, the stratification and variance of errors can be seen as the second percentile increases. Also, these errors are a summation of all errors for all six combinations and 36 intervals used in this analysis. The total number of Type II Errors is roughly 1/2000th of the total numbers of Type I Errors. Therefore, a 5th first and 5th second percentiles reduces the total of both types of errors (regardless of which type of error) based on sheer numbers. A 95th first and 95th second percentile reduces Type II errors to 0.0367 percent but maximizes Type I Errors to 87.9 percent as seen in Figure 20. The specific breakdown for interval and combination for this percentile combination can be seen on the right of Figure 18 and Figure 19.

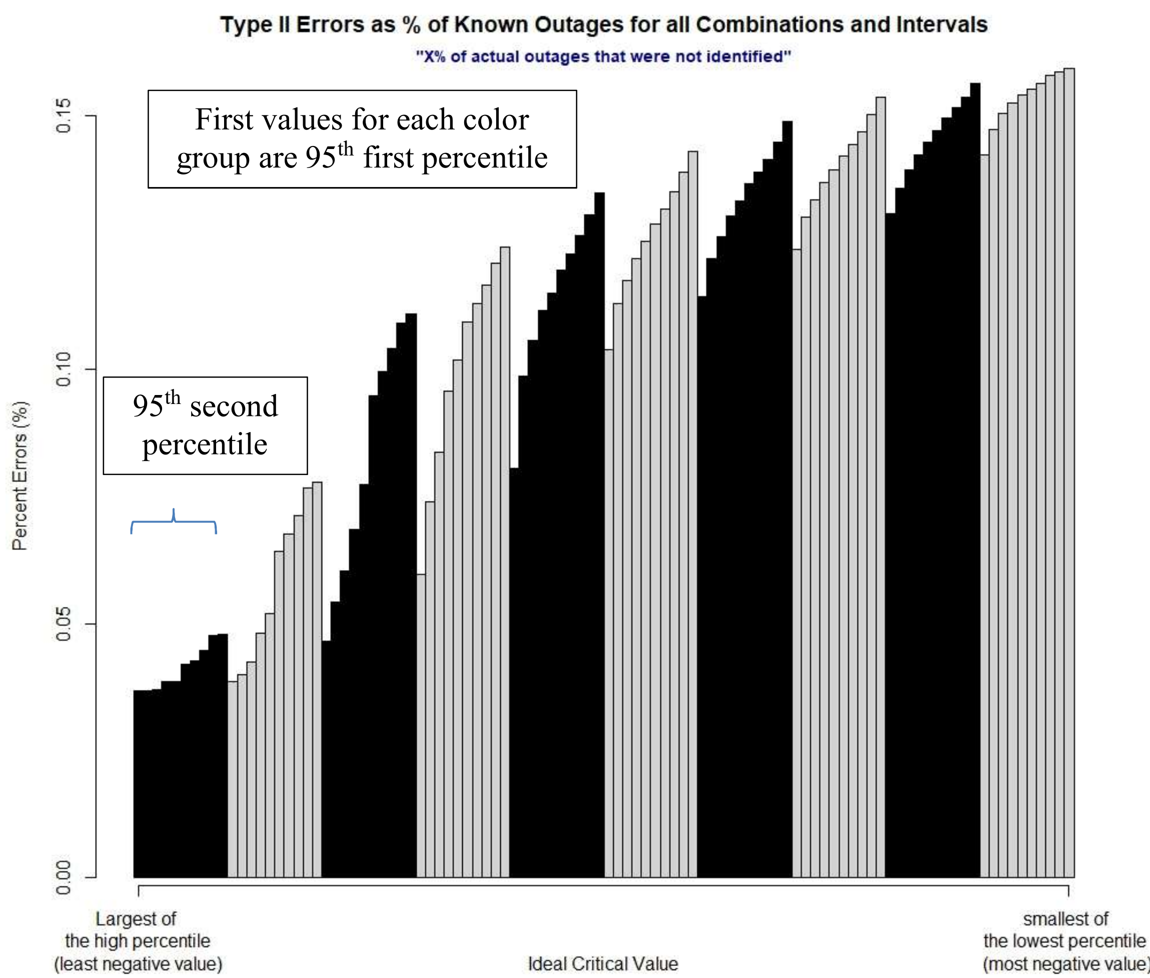


Figure 21. Type II Errors Ordered by Second then First Percentile

Total Type I and Type II Errors were evaluated against their respective maximums to normalize their size difference. Figure 22 and Figure 23 represent the normalized Type I and Type II Errors, respectively. These values are normalized by dividing each value by its respective maximum.

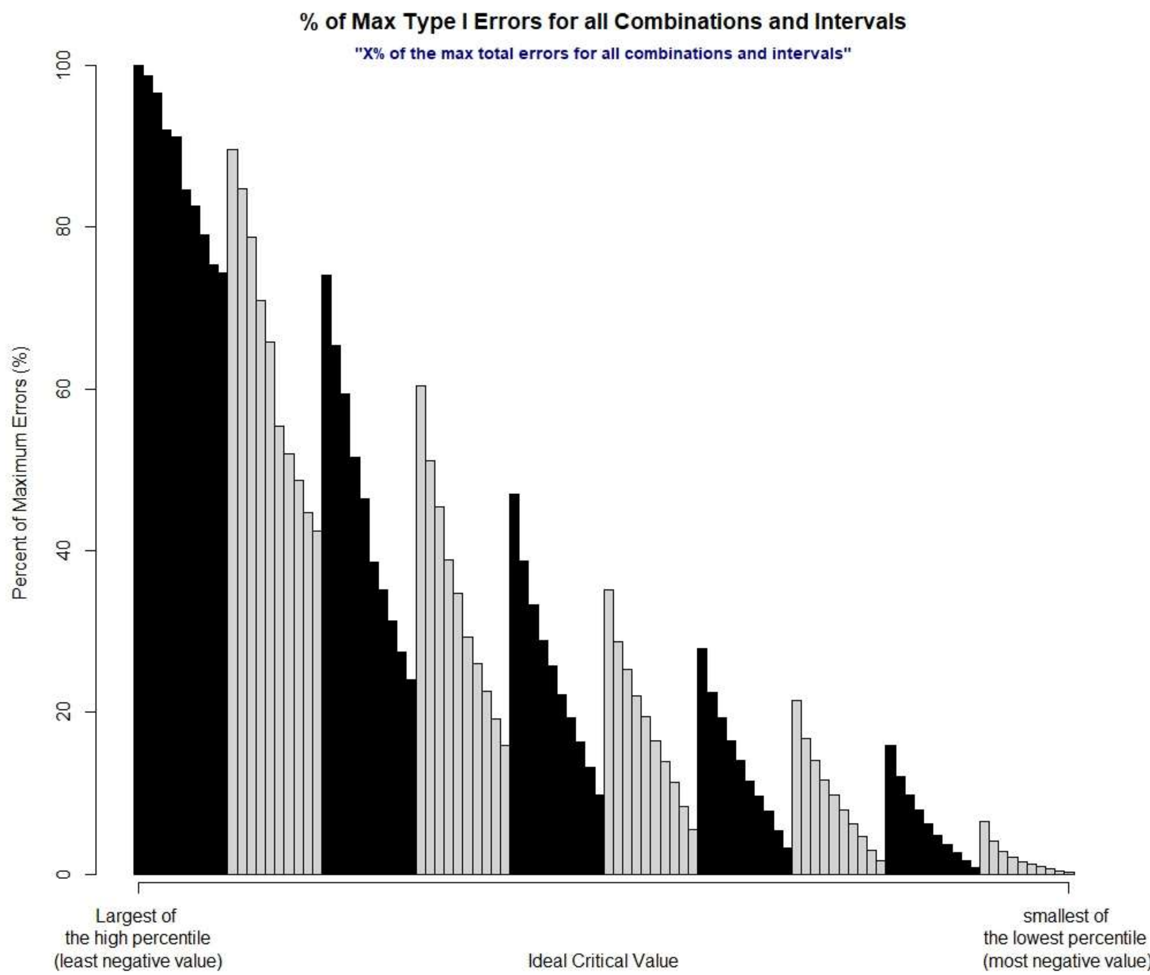


Figure 22. Maximum Normalized Type I Error

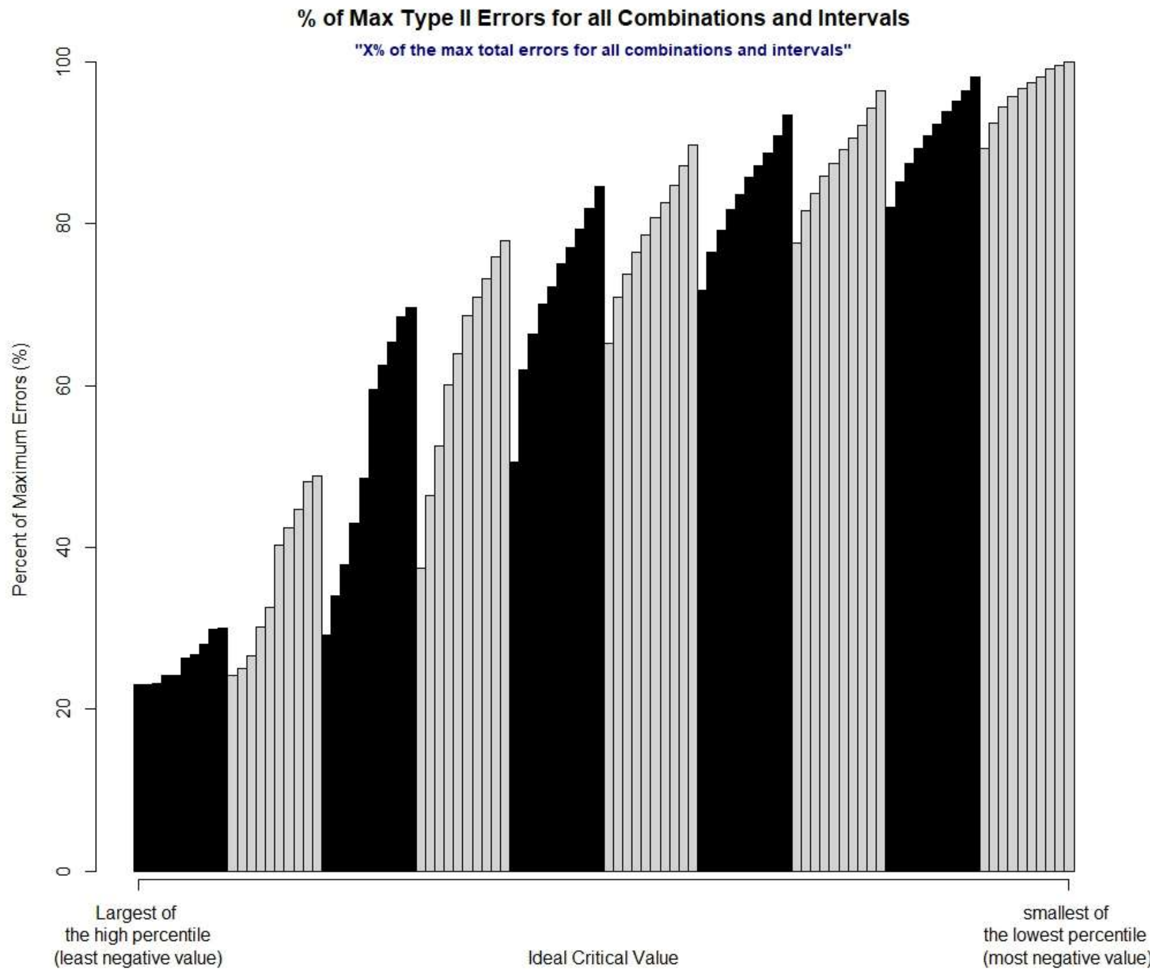


Figure 23. Maximum Normalized Type II Error

To optimize the model and reduce the number of outages, the two normalized errors are added together to see which percentile combination minimizes the combine percentage of maximum errors for each type. Figure 24 graphs both Type I and Type II Error percentage of maximum errors together on the same plot. This graph exposes the smallest combine percentage of maximum errors at the 5th first percentile and 85th second percentile. These results are limited because they aggregate all the intervals and combinations together, thus eliminating their potentially significant effects.

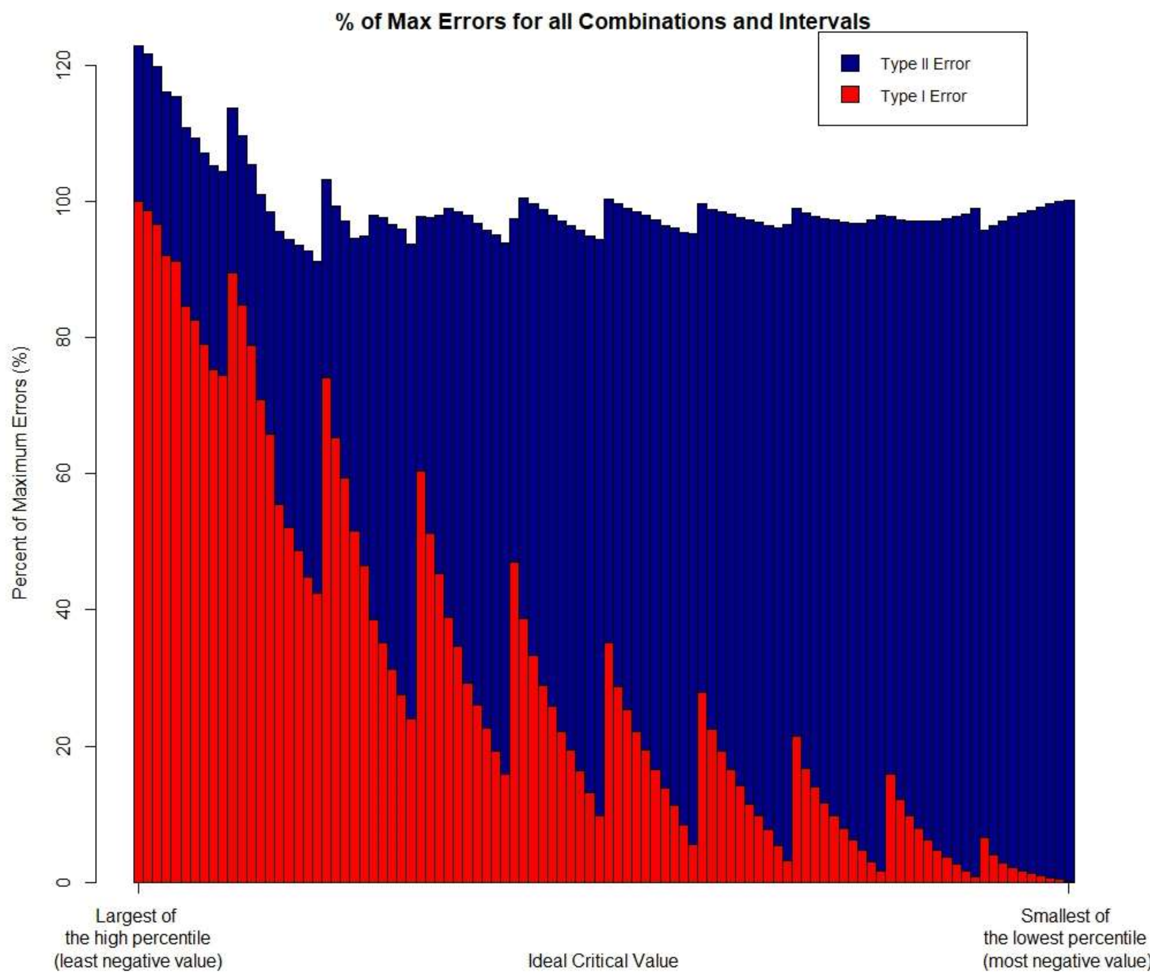


Figure 24. Combine Maximum Normalized Error

At the 5th first percentile and 85th second percentile, Type I Errors are 42 percent of the maximum and Type II Errors are 49 percent of the maximum. The specific breakdown for interval and combination for this percentile combination can be seen in

Figure **25**. A 52-week interval and one-day interval are the best option for this percentile combination with 0 percent Type II Error and 59 percent Type I Error.

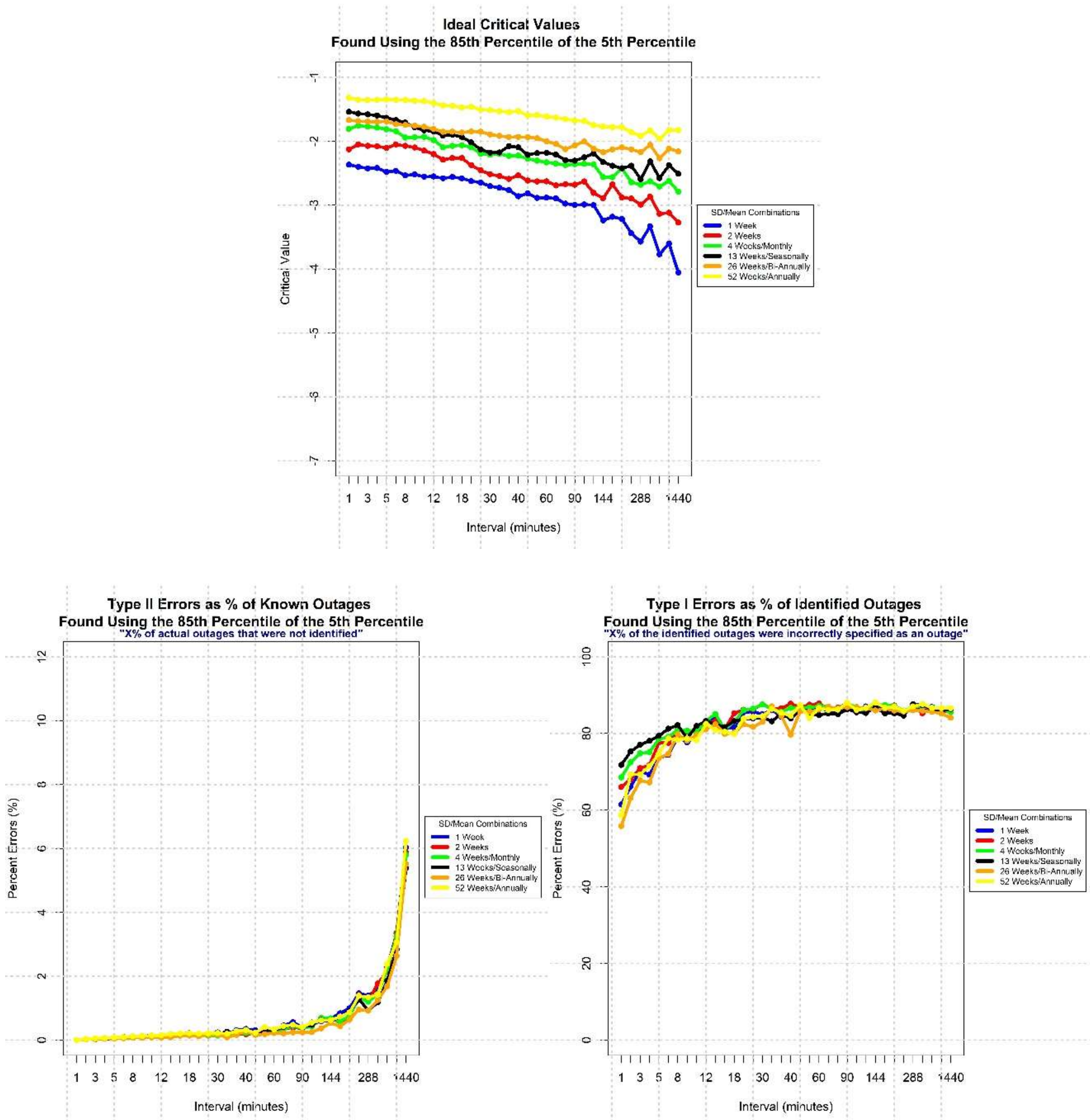


Figure 25. Ideal Critical Value, Type I and Type II Error for the 5th and 85th (first and second) percentiles

All Type II Errors for each of the 100 percentile combinations in this analysis have a similar shape and stay very small the closer they are to the one-minute interval. This area that has the ability to have the greatest impact on the number of total errors. By using a smaller interval the Type II Errors are minimized and the focus can be placed on the Type I Errors (the larger proportion of errors). Figure 26 illustrates this idea.

Figure 26 through Figure 29 shows the one percent of the lowest total errors for all iterations of interval, combination, first percentile, and second percentile. Figure 26 show that to reduce error intervals of 96 minutes or less are ideal. There is no clear interval that stands out amongst the rest, but the graph does show that no intervals were highlighted above the 96-minute interval. **Figure 27. Histogram of Lowest One Percent by Combination**Figure 27 shows that a combinations of 52 weeks is ideal for reducing errors; accounting for 62.5 percent of the sample space. Figure 28 show that the 5^{th} and 15^{th} first percentiles account for 67 percent of the sample space with 5^{th} first percentile accounting for 44 percent of the sample space. Figure 29 shows that the 5^{th} second percentile accounts for 91 percent of the sample space. The single best iteration was a 9-minute interval, 52-week combination, and a 5^{th} percentile for both for the first and second percentiles which is within the ranges specified above (see Table 4).

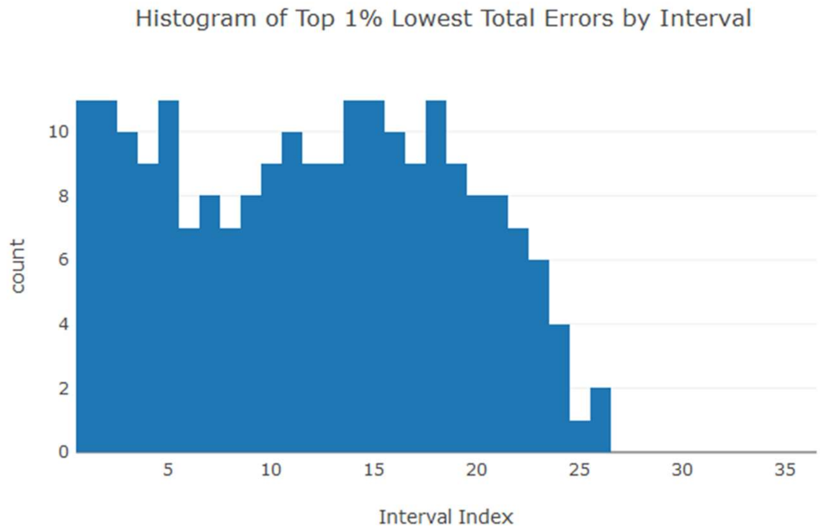


Figure 26. Histogram of Lowest One Percent by Interval

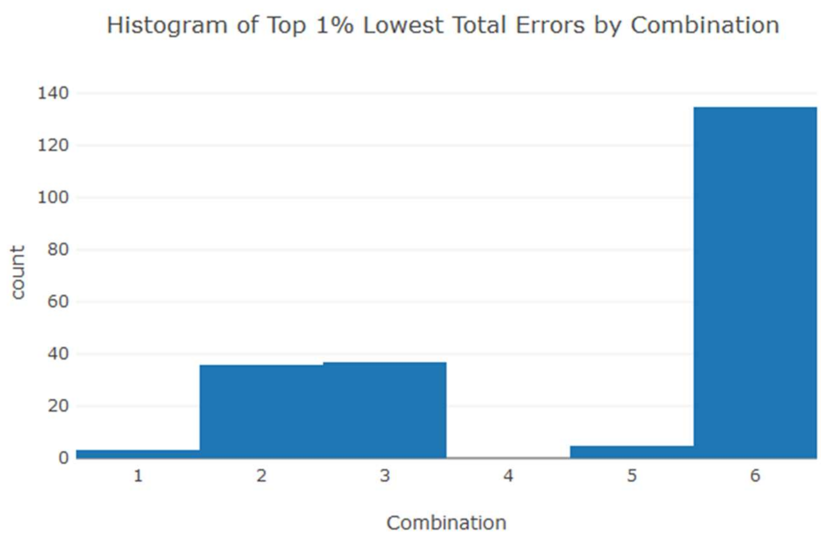


Figure 27. Histogram of Lowest One Percent by Combination

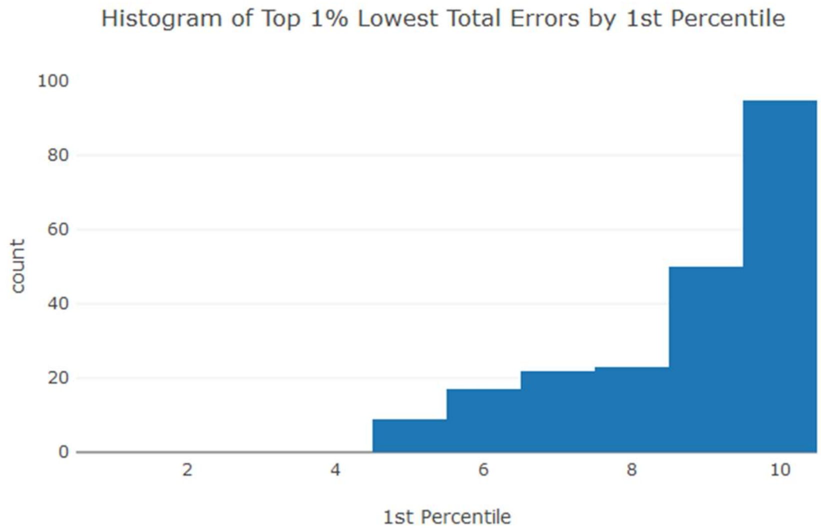


Figure 28. Histogram of Lowest One Percent by First Percentile

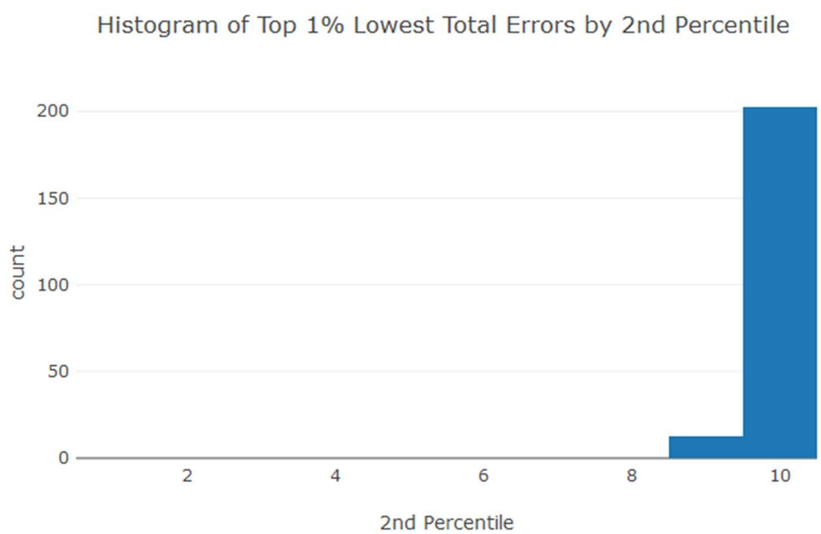


Figure 29. Histogram of Lowest One Percent by Second Percentile

The above process was completed again for the reduction of Type I and then Type II Errors and their graphs are located in Appendix 5 and Appendix 6, respectively. This analysis summarized in Table **3**. There is no consensus for interval range between the three reduction strategies. 52-week combination, and a 5th percentile for both first and second percentiles were seen in each of the three reduction strategies.

Table 3. Best Iterations Ranges to Reduce Errors

Best at Reducing	Interval Range	Combination Range	1st Percentile Range	2nd Percentile Range
Total Error	1-minute to 96-minute	**52-Week only**	15th to **5th**	**5th only**
Type I Error	15-minute to 1-day	26-Week to **52-Week**	25th to **5th**	**5th only**
Type II Error*	1-minute only	**52-Week only**	**5th only**	**5th only**
	*52-Week Combination, 5th first and second percentiles also reduced for Type I Errors			

The individual best iteration for reducing total errors allowed on average 76 minutes of Type I Error and 66 minutes of Type II Error every year. In contrast, if the same methodology was applied for reducing only Type I Errors, there would be two minutes of Type I Error and 533 minutes of Type II Error every year. Again, if Type II Errors were reduced, there would be 152 minutes of Type I Error and less than one second of Type II Error every year. Minutes per year is used to evaluate the amount of error because the different interval lengths change how many individual errors are reported. For example, an iteration with a one-minute interval could report 50 Type I Errors while an iteration with a 50-minute interval would report only one error for the same situation. Table 4 displays the different individual "best" iterations and their effectiveness at reducing the errors. It is important to note that combination, and both the first and second percentiles are the same for all iterations; the same values identified in Table **3**. Appendix 5 and Appendix 6 shows similar graphs as Figure 26 through Figure 29 for the reduction of Type I and Type II Errors, respectively.

Table 4. Individual Best Iterations for Reducing Errors

Best at Reducing	Interval	Combination	1st Percentile	2nd Percentile	Minutes of Type I Error per Year	Minutes of Type II Error per Year
Total Error	9-minute	**52-Week**	**5th**	**5th**	76	66
Type I Error	96-minute	**52-Week**	**5th**	**5th**	2	533
Type II Error*	1-minute	**52-Week**	**5th**	**5th**	152	0.01
	*400 of the 600 1-minute intervals contained only 1 Type II Error. Represented here is the iteration that also reduced the number of Type I Errors					

Success in this analysis is when the number of errors are reduced. Errors can be reduced by using different percentiles to select the ideal critical value. The effects of interval and combination should also be considered when selecting an outage detection strategy as outlined in this research. Leaders or decision makers must determine which risks they are willing to take. Type I Errors generally happen 2000 times more often than Type II Errors when using this methodology, but this number can be greatly decreased if specific iterations are chosen. A leader or decision maker could assign weights to the two types of error to make the decision more quantitative. The analysis conducted above did not assign weights and simply attempted to reduce the number of errors.

4.4 Ellsworth Air Force Base Validation

The dataset received from Ellsworth Air Force Base (AFB) contained 87 electric meters with daily energy usage. After removal of nine meters that reported zero energy usage for all days reported, analysis was completed on 78 meters. Some meters had very sporadic energy usage with the majority of the data being zeros (Data IDs of 21 and 70 in this analysis). All meters only reported daily energy usage intervals which did not allow for interval optimization because daily usage is the maximum available interval in this analysis. The ideal critical values used were the values that reduced the overall percentage of Type I and Type II Errors; 5th first percentile and 85th second percentile (see Figure 25). This resulted in a "Good Match" for over 98 percent of the data for each combination (see Figure 30). Ellsworth AFB data had the best good matches with a 1-week combination instead of a 52-week combination as seen in section 4.2. A 1-week combination equated to 99.8 percent good match. Additionally, there is no Type II Errors because there is no varying of the interval width and the processing code finds "known" and "identified" outages by searching for zeros. Further analysis needs to be completed with Air Force data with a smaller data collection interval.

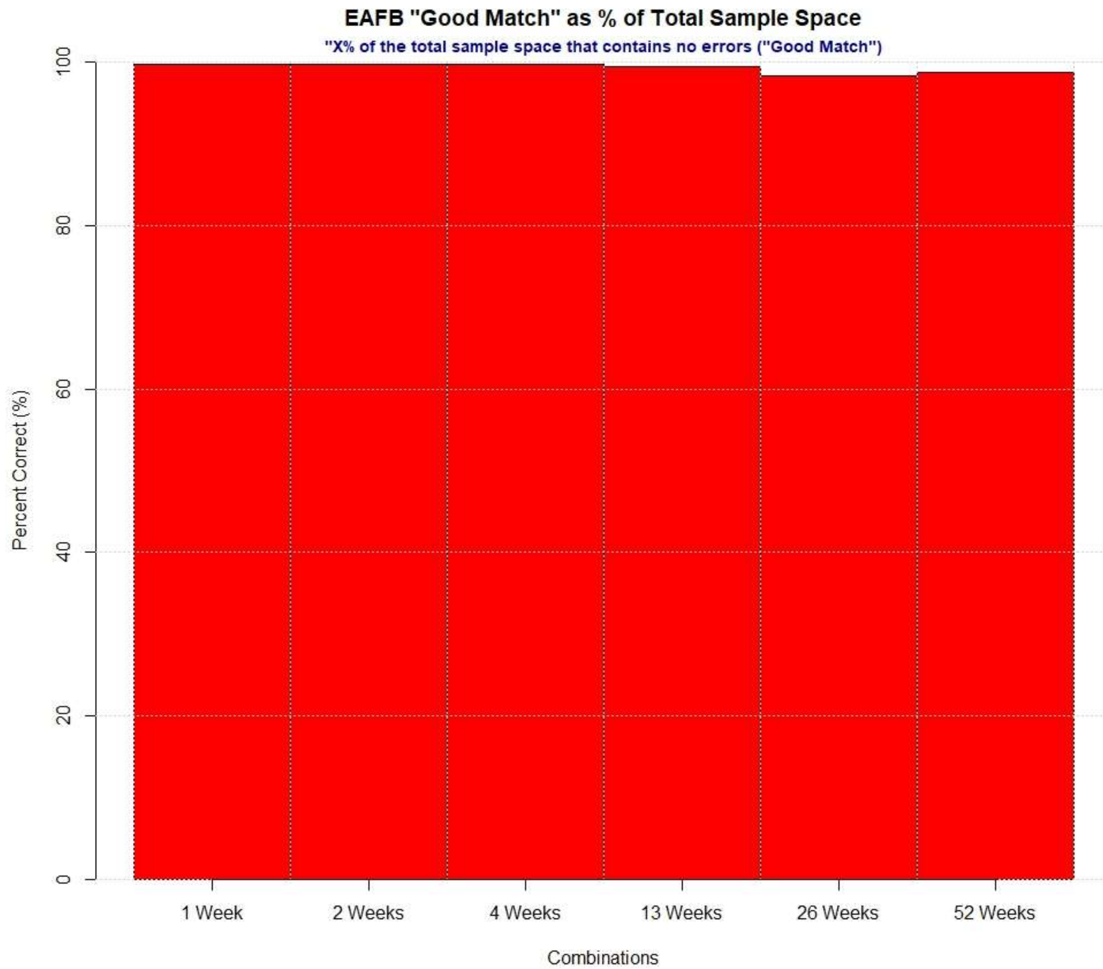


Figure 30. Ellsworth AFB "Good Match" vs. Total Sample Space

The ideal critical values and code did locate all the known outages but it also identified many intervals that weren't outages (Type I Error). There was more errors than true outage detection in all combinations but two as seen in Figure 31. A 1-week combination is the best of the six combinations. A 1-week combination has a 10 percent error, meaning 90 percent of indicated outages are indicated correctly.

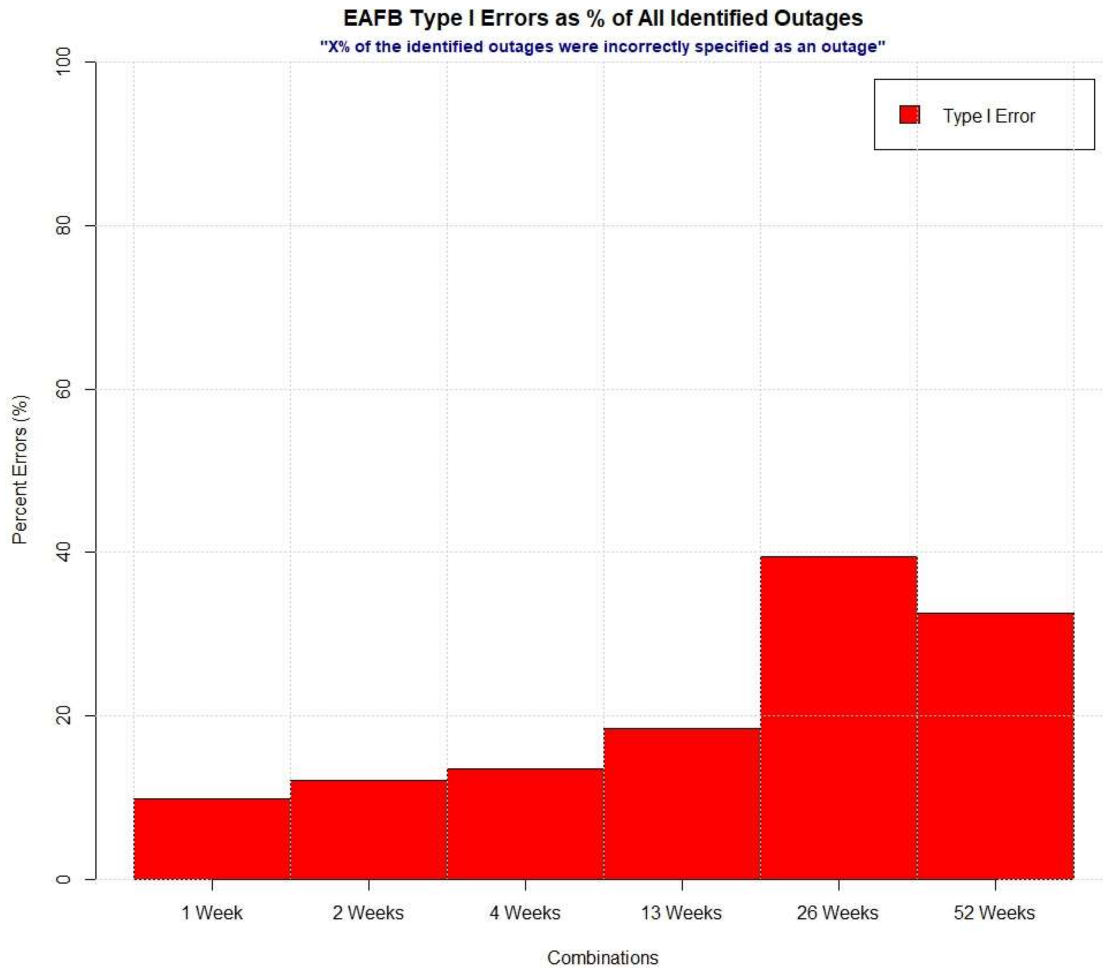


Figure 31. Ellsworth AFB Type I Errors vs Identified Outages

Figure 32 shows the percentage of outages by the Data ID. There was Data IDs that contained a few errors, one that contained five times the errors and some that contain none. The figure shows unconditionally that Data ID number 6 could not be trusted more than 10 percent of the time, regardless of if there was an outage or not.

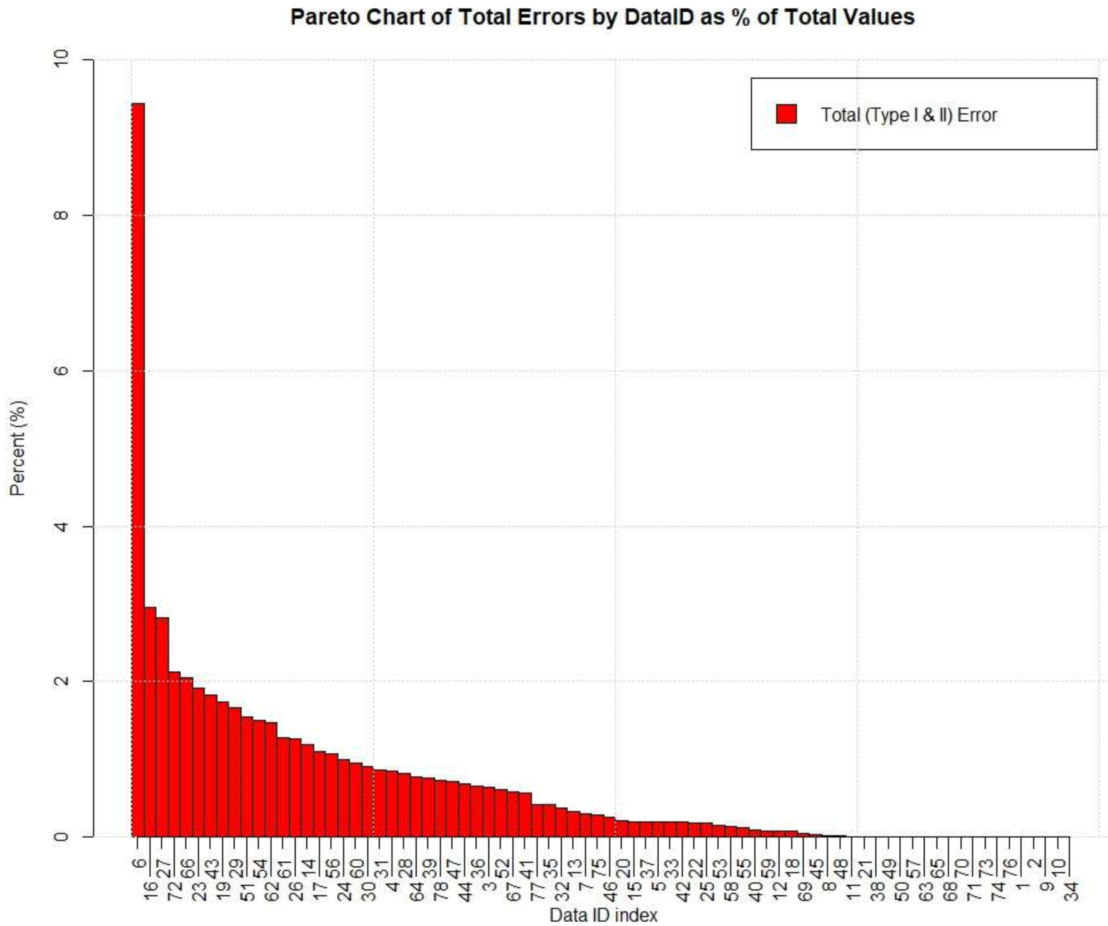


Figure 32. Pareto Chart of Ellsworth AFB Errors by Data ID

These Data IDs were then graphed in the same order to look for trends in outage length. Figure 33 shows that most Data IDs with sporadic energy usage, Data IDs 21 and 70, are actually better for locating outages or times with zero energy usage; located to the far right. While it is apparent that these facilities are only used in short bursts, they can still be analyzed for energy usage and outage information if the occupation characteristics and patterns are better understood.

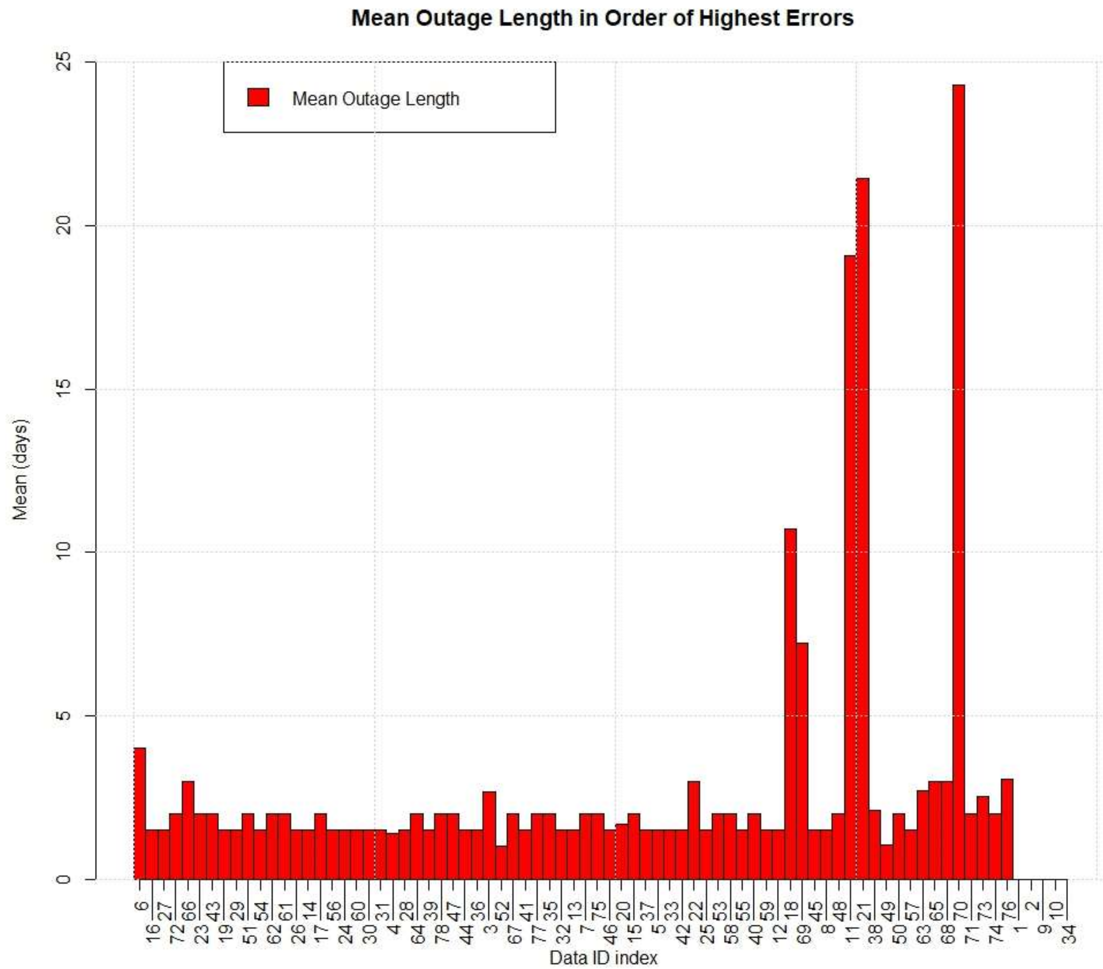


Figure 33. Ellsworth AFB Mean Outage Length in Descending Total Error Order

By mapping the exact date of the outages, a few trends appeared. In Figure 34, a few days pop out as being days with increased outages. Because two meters contained very sporadic energy usage, there is a constant outage level throughout time period. These two meters break the earlier assumption that facilities will always have a small amount of power draw. By looking at the metadata for these two meters, they report energy usage for an empty warehouse and an alternate location for the command post to run emergency operations (buildings 7262 and 1011 or meters 23991390 and 56212956).

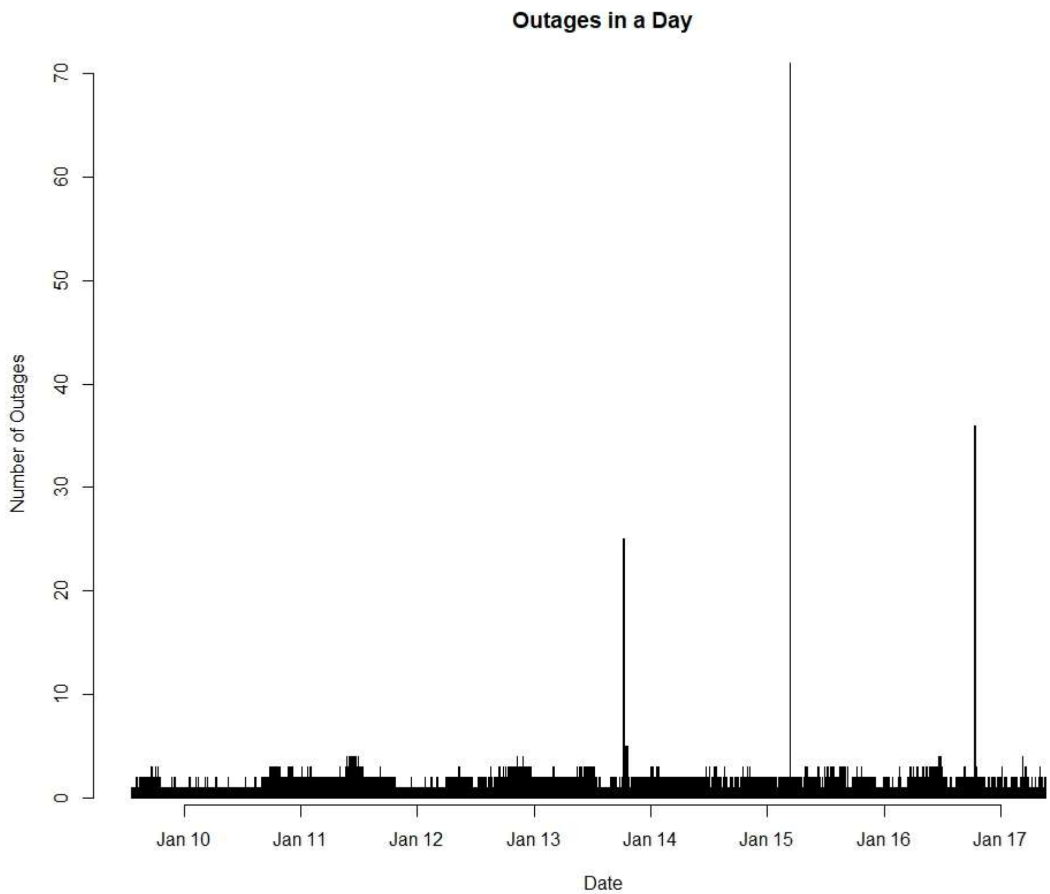


Figure 34. Timeline of the Number of Outages

Figure 35 shows the exact date of the top ten days of outages in the seven years' worth of data. South Dakota's hard winters may have been a factor in these top 10 outages. With only 78 meters in the analysis, having 71 go down on 10-11 March 2015 possibly shows a base wide outage that was only mitigated by 7 generators. These outages are at the mercy of the commercial power supplier. If many outages are being seen base wide, then a more political strategy must be used to work with the commercial supplier to provide a better power quality. Other, smaller groupings of outages could point to certain areas of the base that have degraded electrical infrastructure. This type of analysis can be used to identify specific electrical branches that have more outages than others. This stratification can advocate for funding in specific areas over other areas on the same base. Doing so could also trace the facility outages back to a single point between the facilities and the commercial power supplier connection.

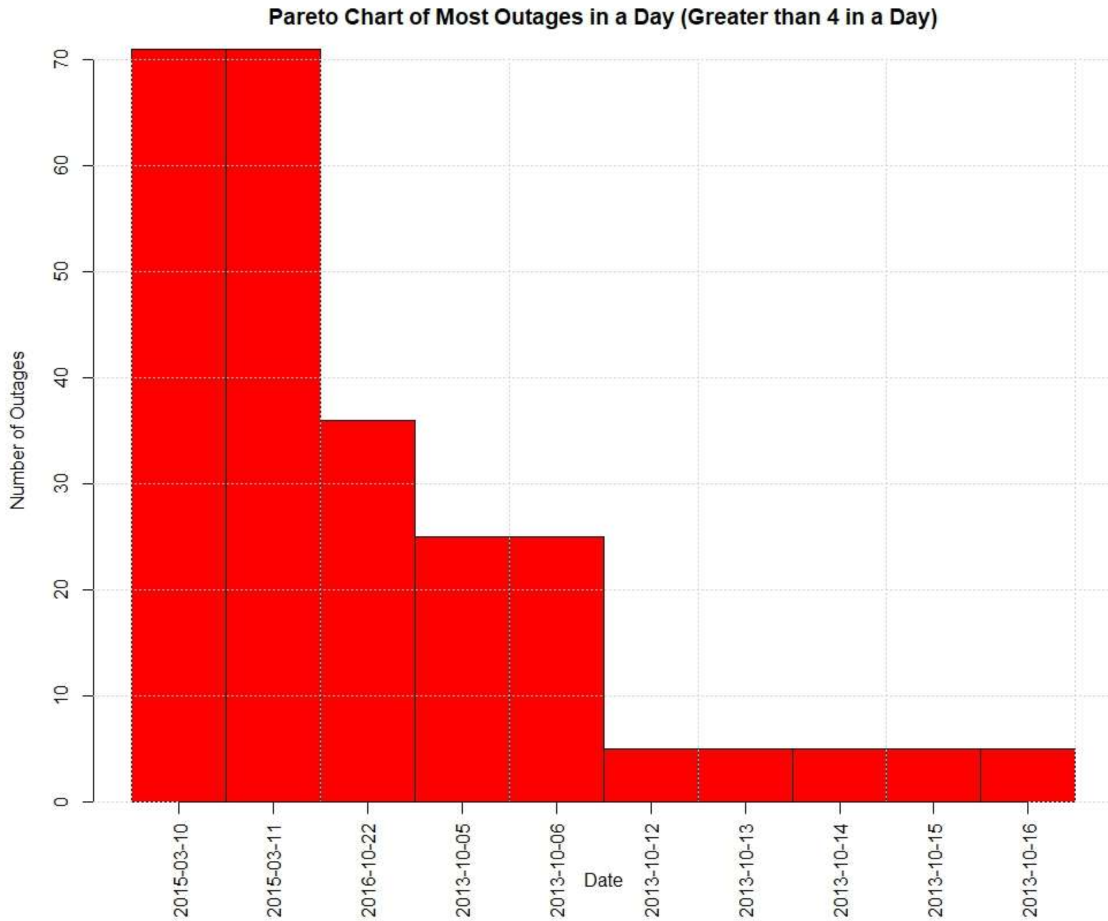


Figure 35. Pareto Chart of Most Outages in a Day

In conclusion, better data is required for a thorough analysis of the ideal critical values for Ellsworth AFB. Additionally, with the limited access to real Air Force data, it is unclear if ideal critical values from differing geographical locations (like Texas and South Dakota) are universally effective. Finally, it was seen that interconnected systems, like Air Force Installations, will have system wide effects that can be reported from multiple individual measuring points (see Figure 34 and Figure 35).

4.5 Massachusetts Institute of Technology Lincoln Laboratory Tool

The Massachusetts Institute of Technology Lincoln Laboratory (MITLL) Tool was developed for the DoD to quantify energy resilience and create a way to produce life cycle assessments of different electrical grid projects. MITLL stated their largest issue was valid outage data (Nick Judson & Pina, 2017). This analysis was tailored to show the impact of having better outage information provided by advanced meters. This tool can be tailored to any base and any configuration. For this analysis the base line for all inputs were left except for one; the System Average Interruption Frequency Index (SAIFI). The SAIFI tells the Matlab program how many outages occur on average per year. A smaller SAIFI indicates a more reliable electrical distribution infrastructure. This is then converted into a mean time to failure metric that is used in a probabilistic Monte Carlo simulation with 1,000 runs. The preset location used was Joint Base Pearl Harbor-Hickam. This base was used because it is an Air Force Joint Base and it has high pressure to perform because of Hawaii's high cost of electricity.

Type II Errors for 12 distinct iteration of interval, combination, and both first and second percentiles were used to setup the sensitivity analysis of the MITLL Tool. Table 5 displays the SAIFIs associated with these 12 iterations. The 52-week combination was used in all iterations because it has proven to be the most useful in this analysis. The first percentile had very minimal effects on the outcome of the SAIFI. A one-minute interval from Table **4** proved to reduce the Type II Errors close to zero; this was chosen to be the baseline. A baseline of 10-SAIFI simulates a system that has extremely few to no Type

II Errors. Type I Errors were not analyzed in this section. The SAIFI reduces as the intervals grow in width from one minute to one day. And as the Type II Errors increase (as the interval width increases), there is more outages being missed, thus the calculated number of outages is underestimated and the SAIFI becomes incorrect. For all 12 situations in Table 5 the SAIFI should be 10, this causes issues with designing electrical systems. The differences in SAIFI and Type II Errors can be seen as the iterations change in Table 5.

Table 5. MITLL Sensitivity Analysis Setup

Interval	Combination	1st Percentile	2nd Percentile	Calculated SAIFI	Type II Errors
1-min	52-week	5th	95th	10	~0
1-min	52-week	95th	95th	10	~0
1-min	52-week	5th	5th	10	~0
1-min	52-week	95th	5th	10	~0
15-min	52-week	5th	95th	9.994	0.000616
15-min	52-week	95th	95th	9.994	0.000642
15-min	52-week	5th	5th	9.973	0.002655
15-min	52-week	95th	5th	9.973	0.002802
1-day	52-week	5th	95th	9.377	0.06228
1-day	52-week	95th	95th	9.377	0.06228
1-day	52-week	5th	5th	8.839	0.114955
1-day	52-week	95th	5th	8.839	0.116109

The baseline SAIFI was set to 10 outages per year. This number is within the range used in other simulations preloaded on the MITLL Tool. A critical output from the tool was the recommended electrical system architecture and its corresponding unserved electrical load. The top graph is ordered in descending cost per KWh and the bottom shows the unserved load in the same order as the top graph. The black bars in the bottom graph shows the current electrical infrastructure architecture. This baseline is facility sized generators and commercial power with a total unserved load of 105 MWh (200 times more than the maximum on the graphs below). Appendix 7 shows the makeup of the different architectures found on top of each set of graphs. Blue bars indicate architectures that are worse off than the black baseline bar. Green bars are architectures whose cost and unserved load are lower than the baseline. Figure 36 shows the 10-SAIFI base line for a one-minute interval of all first and second percentile iterations.

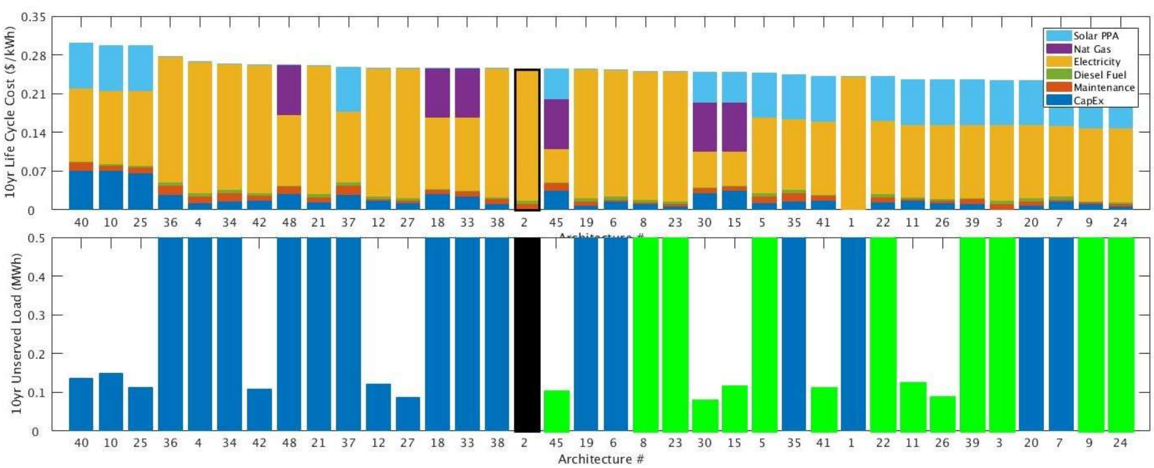


Figure 36. MITLL Tool Output for One-Minute Interval/ 10.0 SAIFI

For a 15-minute interval the Type II Error was 0.3 percent and 0.06 percent for the 5th and 95th second percentiles, respectively. Applied to the SAIFI, a 15-minute interval would suggest that 9.97 and 9.994 outages per year occurred. This was calculated because a 0.3 percent Type II Error translates to 3 missed outage for every 1000 actual outages. On the surface level it may seem desirable to only have 3 missed outages, but with a one-minute interval there is, or theoretically is, zero missed outages. Figure 37 shows the 15-minute interval/5th second percentile output with a corresponding SAIFI of 9.97. Figure 38 shows the 15-minute interval/95th second percentile output with a corresponding SAIFI of 9.994. Very little difference can be seen visually between these two figures. As with most iterations from Table 5, the following architectures were at least 100 times better at providing continuous power and consistently cheaper: 45, 30, 15, 41, 11, and 26.

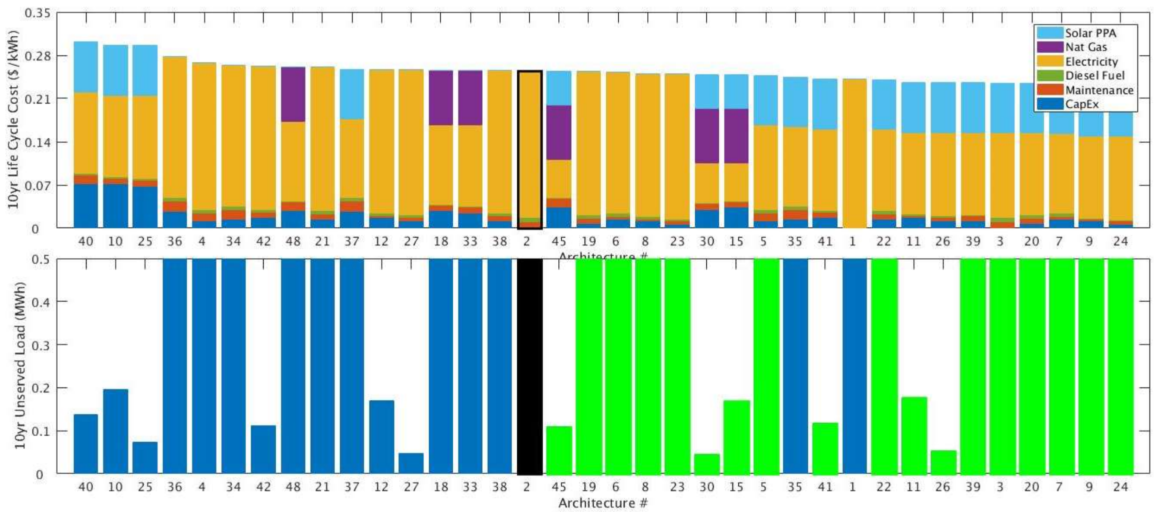


Figure 37. MITLL Tool Output for 15-Minute Interval/5th Second Percentile/9.97 SAIFI

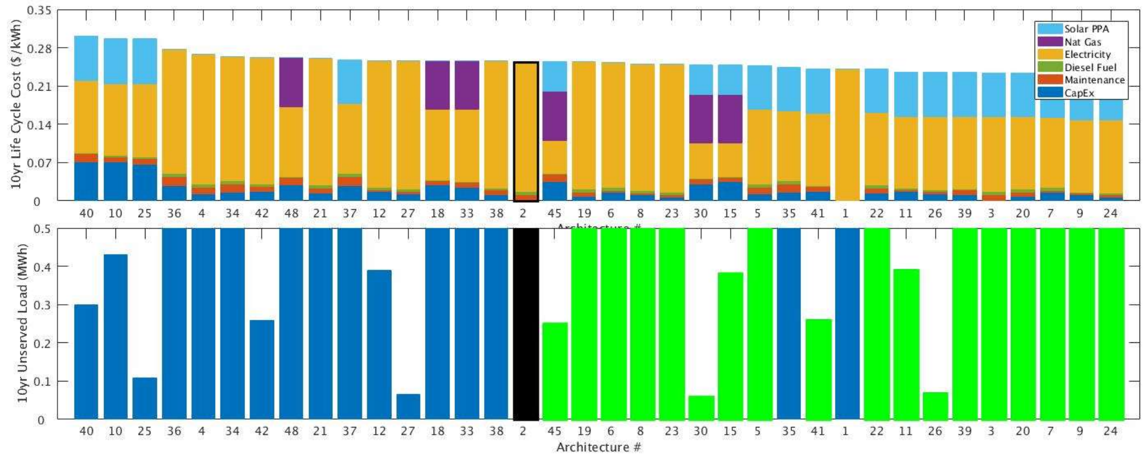


Figure 38. MITLL Tool Output for 15-Minute Interval/95th Second Percentile/9.994 SAIFI

Figure 39 shows the one-day interval/5th second percentile output with a corresponding SAIFI of 9.377. Figure 40 shows the one-day interval/95th second percentile output with a corresponding SAIFI of 8.84. Theoretically, as SAIFI goes down so will the unserved load. This lower SAIFI hides outages and the true number of outages are not used to select the ideal system using the MITLL Tool. Figure 39 and Figure 40 show near perfect outputs for the top six architectures. Again this is a false representation of the system due to the Type II Errors that masked the true number of outages. Figure 40 also shows that architecture 45 is no longer a cheaper option. Depending on the situation, 45 could have been used, but may have been eliminated as an option due to these Type II Errors.

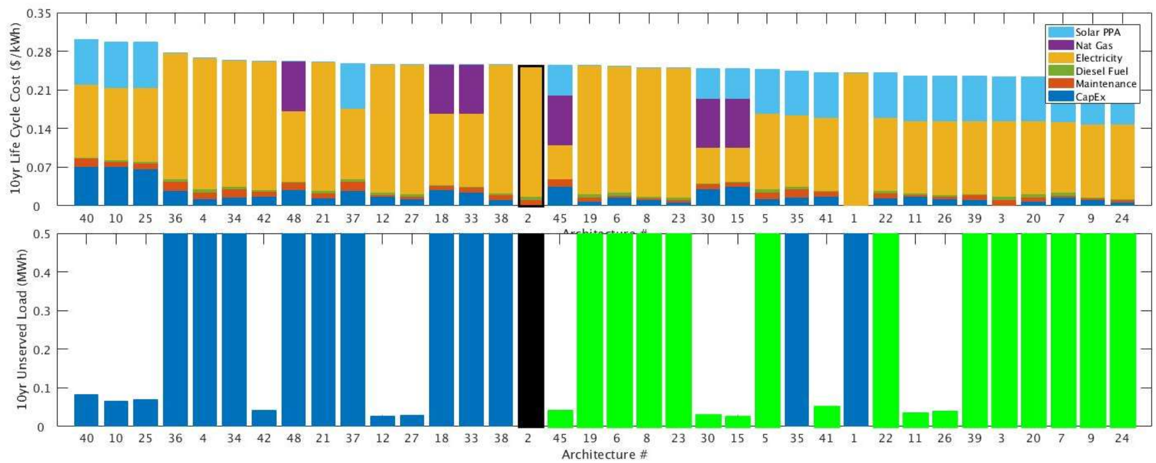


Figure 39. MITLL Tool Output for 15-Minute Interval/5th Second Percentile/9.377 SAIFI

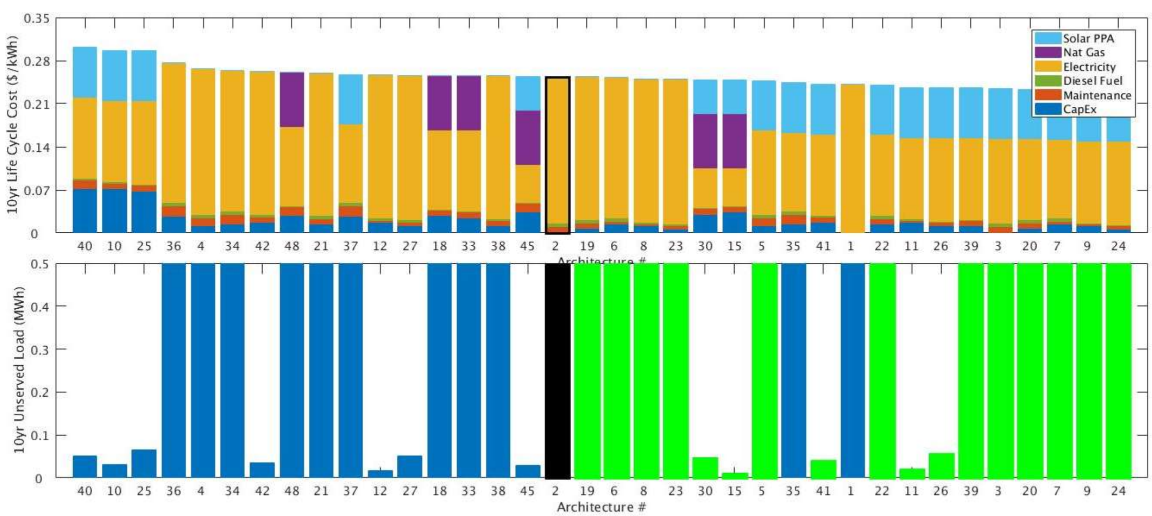


Figure 40. MITLL Tool Output for 15-Minute Interval/95th Second Percentile 8.84 SAIFI

The problem with the additional Type II Error is that the operator is over confident in the status of their electrical system. They think they have less outages a year than they really do. Even a small difference effects the outputs greatly as seen in Figure 36 compared to Figure 40. Table **6** shows the results from the analysis of the effects of Type II Errors. The baseline ideal architecture was 30 which included: facility level and centralized standby generators, islandable photovoltaics, uninterrupted power supplies, fuel cells and microgrids. This architecture was the same for all cases but the last which had an ideal architecture of 15. Architecture 15 included the same items at 30 except twice as many centralized generators with no facility level standby generators and cogeneration of heat and power. If anything other than the baseline is used then the realized benefits would be less than what is expected. If a design was created thinking 15 is the best architecture and the truth was that 30 was the best, then not only would there be less benefits but also the incorrect architecture would be constructed.

Table 6. MITLL Ideal Architecture Results

	Interval	Combination	1st Percentile	2nd Percentile	Calculated SAIFI	Ideal Architecture	Unserved load (KWh)
Baseline (best)	1-min	52-week	5th	95th	10	30	79
	1-min	52-week	95th	95th	10	30	79
	1-min	52-week	5th	5th	10	30	79
	1-min	52-week	95th	5th	10	30	79
Air Force Standard	15-min	52-week	5th	95th	9.994	30	60
	15-min	52-week	95th	95th	9.994	30	60
	15-min	52-week	5th	5th	9.973	30	45
	15-min	52-week	95th	5th	9.973	30	45
Ellsworth AFB	1-day	52-week	5th	95th	9.377	30	30
	1-day	52-week	95th	95th	9.377	30	30
	1-day	52-week	5th	5th	8.839	15	10
	1-day	52-week	95th	5th	8.839	15	10

The system will perform as seen in Figure 36 but the operator will think they are paying for the performance as seen in Figure 40. Proper identification is helpful because the more options to choose from makes modernizing and upgrading easier and possibly cheaper.

4.6 Summary

The three investigative questions and their simplified answers are as follows:

i. Can outages be found using historical advanced meter data with Means and Standard Deviations? If so, what Standard Deviation should be used?

   A. Yes, Means and SDs can be used to identify outages. Ideal critical values can be used to estimate an outage timestamp, but there are associated risks and errors for which ideal critical values are chosen as seen in the percentile sensitivity analysis.

ii. What is the optimum data interval and combination to identify outages using historical advanced meter data?

   A. A 52-week combination (annual average) and 5th first and second percentiles produced the best results for correctly identifying data. Selecting which interval to use depends on which type of error is attempting to be reduced.

iii. How does advanced meter data affect electrical grid modernization planning including advanced energy production technology?

A. Advanced meter data can deliver outage information for the use of influencing future electrical projects. Good meter data gives confidence that designs will be accurate.

These questions were answered through the use of the analysis in this chapter. Analysis was conducted in a way to produce results and recommend improvements to the Air Force. Advanced meter data and systems already in place have the power to help influence operations, maintenance and construction efforts in the DoD.

V. Conclusions and Recommendations

5.1 Conclusions of Research

The United States Air Force standard of 15-minute data collection intervals for advanced meters is good for benchmarking, facility energy intensity comparison, and unit energy use comparison (Department of the Air Force, 2017a). The data collected could also be used for locating outages and improving the way the Air Force spends its money to improve its electrical systems. This research attempted to show that historical electrical meter data could be used to find outages. The concept combination was created to analyze different data collection intervals. These concepts are far from being perfect and warrant a closer look by future researchers.

A 52-week combination and 5th first and second percentiles produced the best results for correctly categorizing data. It was also found that smaller intervals also assist in reducing errors. This was accomplished through the use of ideal critical values that were developed by using data from residential homes in Austin, TX through Pecan Street. These specific conclusions are limited to Austin, TX but the methods used could be validated for a larger geographic region with further testing and more data. The ideal critical values were applied to electrical energy usage for Ellsworth Air Force Base (AFB). Due to the large width of the data collection interval, results from this analysis were very limited. The Ellsworth AFB analysis did, however, prove an assumption wrong that if an advanced meter was reporting zero energy usage that there was an outage or malfunction. This is because low use or unoccupied facilities could actually have zero registered energy usage or amperage. This finding limits the application of this thesis to occupied facilities with electronics. In the end, the ideal critical values located outages with a 10-40 percent Type I Error (see Figure 31). Additionally, power can be disconnected for safety purposes during routine maintenance and by using the methodology in this thesis this would be classified as an outage. These issues can be overcome with detailed record keeping of the maintenance, repair and real property records. This points to the fact that outage detection is as much analytics as it is documentation.

The way the critical values are determined was also investigated. Like much of the other outcomes, there is not exactly a perfect solution because there is a lot of uncertainty and risk involved with using these methods. Leaders must understand the risks to better inform their decisions on how to apply these methods. The Massachusetts Institute of Technology Lincoln Laboratory (MITLL) Tool was also used to identify how the planning of different electrical projects are affected by differing outage data. Specifically, Type II Errors from a 15-minute and 1-day data collection intervals influenced over confident decisions. With better meter data, more outages can be found and better designs can be created by looking at the energy usage trends. These positive effects could compound because designing systems for proper conditions reduce the chances of overloading electrical systems (leading to more outages).

5.2 Significance of Research

Significant improvements to electrical distribution repair planning and programming could be seen by incorporating meter data in addition to linear segmentation and geographic information system (GIS) data. The impacts to funding a particular project over another could be better realized with the addition of outage and energy usage information. Asset management companies in the private sector do similar surveys and studies for public utilities. By using the systems the Air Force is developing and already has in place, like the Advanced Meter Reading System (AMRS), better funding can be provided to installations around the world. Doing so would be a positive move towards the Air Force priority to cost effectively modernize (Secretary of the Air Force, 2017).

The Air Force's advanced meter data collection interval width could be decreased to accommodate finding smaller outages. The reduction of the interval width will create even more data to store and transfer, as described by Zhou et al. (Zhou et al., 2016). These changes could be implemented depending on the ability of the current AMRS system. Locating more outages will give the Air Force a better understanding of the hazards that could be unnoticed. Finding and reducing outages will create a more resilient and reliable grid which is the main focus of the Air Force Energy Flight Plan (Department of the Air Force, 2017b).

5.3 Recommendations for Action

It is recommended that The Air Force Civil Engineer Center (AFCEC) investigates the effectiveness of the AMRS to identify outages and make locating outages one of their key pillars for justifying advanced meter investments. This research uses the idea that Last Gasp messages are not being used. With the high error rates of this research more work need to be done, but the use of Last Gasp messages removes the requirement of guessing if an outage occurred. Last Gasp messages can give a start time and meter reconnection can give an end time for the outage. This process could be automated and integrated within the Air Force's current AMRS. AFCEC states that a similar, but less precise, function already exists in the current AMRS. It is accomplished by highlighting the interval that communication was lost with the meter (possibly from an outage), but this protocol does not provide an exact timestamp for when the loss of communication occurred. Regardless, its ability to inform asset management decisions is not happening systemically throughout the Air Force (Gerdes, 2017).

Finally, if the ARMS is able to effectively locate outages, then its capabilities should be combine with the Utility System Outage Reporting and Tracking (USORT) Tool. With AMRS and USORTS working together, a complete picture of the vulnerabilities in the Air Force's electrical system can be seen. This data could then be used to inform IPL decisions.

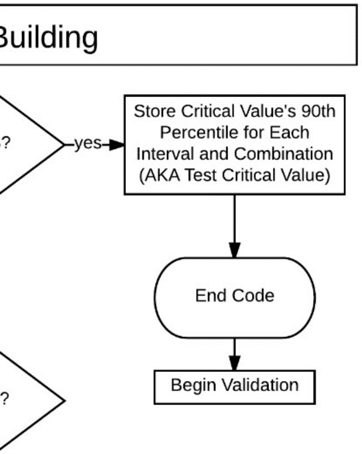
5.4 Recommendations for Future Research

Future follow-on research is a must to refine and validate the findings from this thesis to other geographic areas. Additionally, other outage detection strategies could also be investigated, like analyzing the maximum difference between sequential intervals to locate outages (a sudden drop in energy usage). Future research could also be conducted on other utilities that utilize advanced meters. Another topic could be to investigate if there is a seasonal dependency of how long the intervals should be. For example, in summer the intervals could be smaller because there is a higher Heating Ventilation and Air Conditioning load or in the Fall and Spring the intervals can be wider to alleviate additional unnecessary data storage and transfer.

Another aspect of this research that could be expanded is looking at multiple facilities on a single meter. This could be simulated or organic. By looking at trends in this aggregated energy usage there could be dependencies that exist. These aggregated meters could work in tandem with single facility meters to give more fidelity that an outage has occurred.

Another very useful analysis could be an artificial intelligence application to outage detection with advanced meters. Computerization of when and where outages were occurring can help the system learn what outages at what locations are being missed and when they are being correctly identified. This application is advanced and rigorous, but also become obsolete by using Last Gasp messages.

Finally, the Microgrid Design Toolkit (MDT) created by Sandia Laboratories is another useful tool that could also be used in future iterations of this research involved in energy resilience. The MDT is a software using Monte Carlo simulations to evaluate the performance of different electrical infrastructure architectures, making it a perfect candidate for any type of sensitivity analysis. This program is free and available online through the Department of Energy.



lding Flow Chart

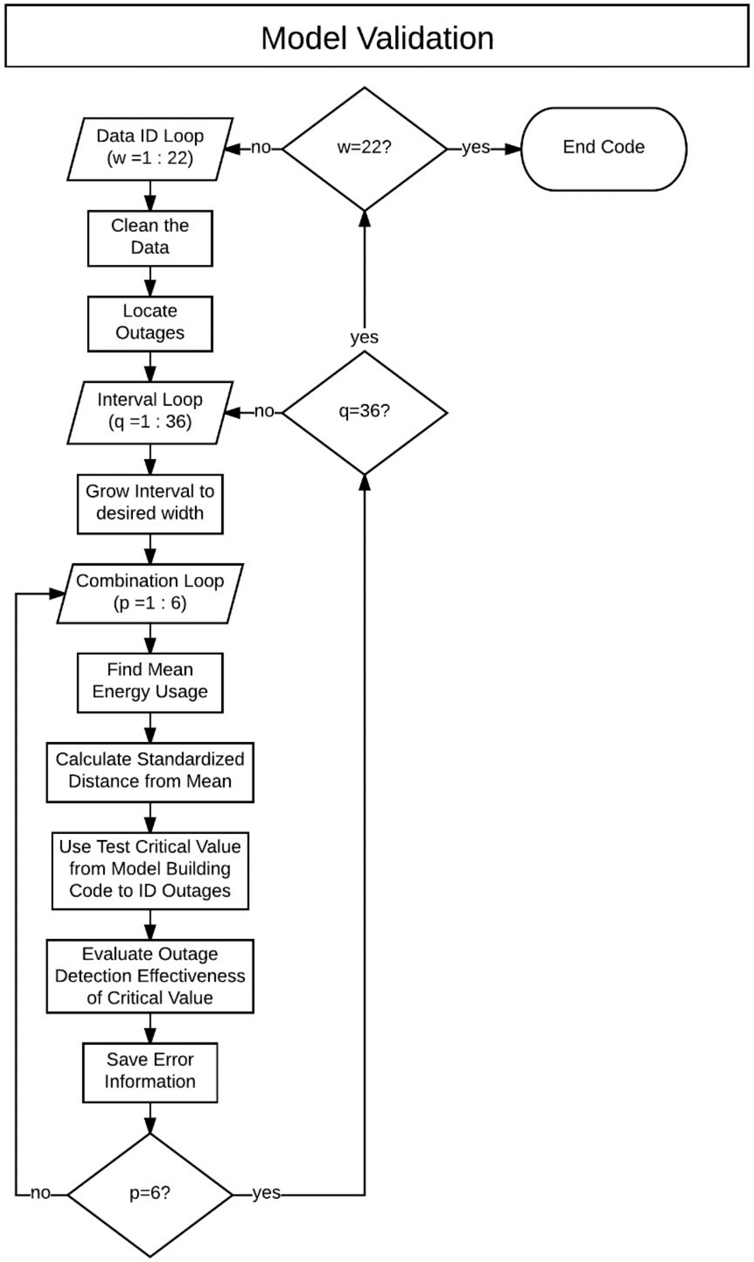


Figure 42. Model Validation Flow Chart

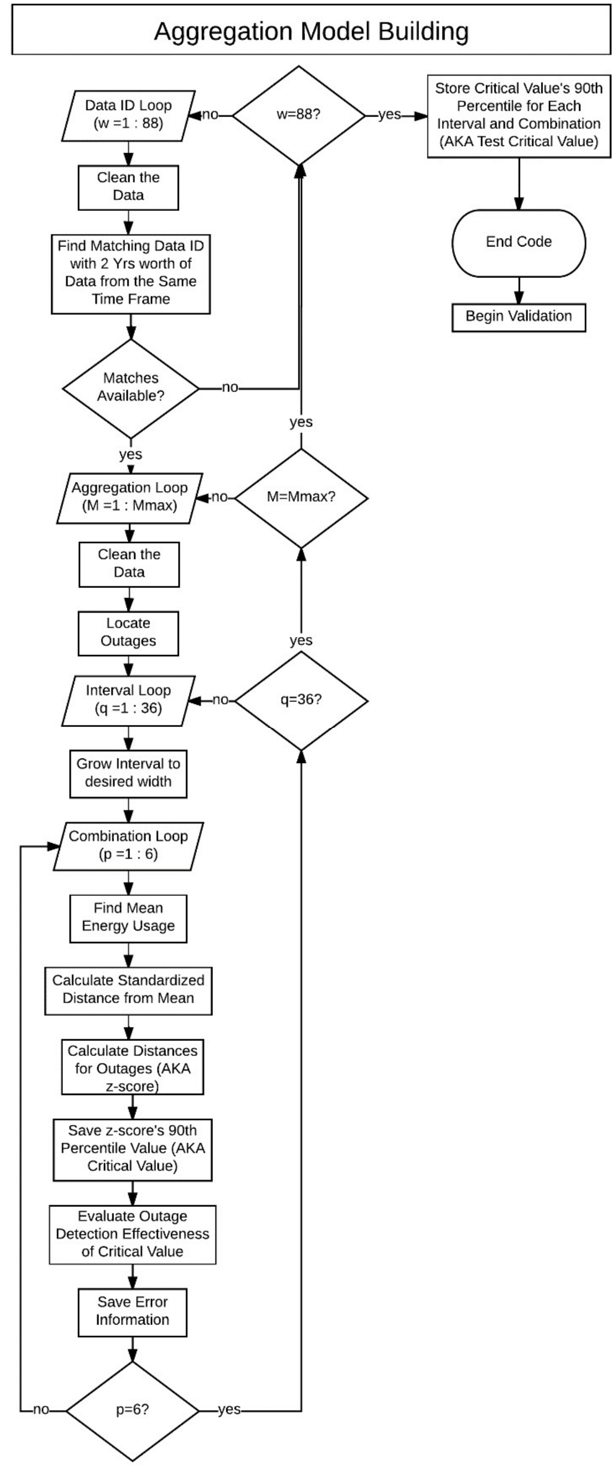


Figure 43. Aggregation Model Building

Appendix 2. Raw R Studio Code

Pecan Street

```
###############MODEL BUILD###############
#library(readr)
#rawdata <- read_csv("pecan/rawdata.csv", col_names = FALSE)
colnames(rawdata)=(c("X1","X2","X3"))
dataidlist=unlist(unique(rawdata[,3]))
numdataid=length(dataidlist) #number of distinct data IDs=88
possible_int_leng_min=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,72
0,1440)
possible_combo_int=c(1,2,4,13,26,52)
crit_val=array(dim=c(6,36,numdataid))
store_x=array(dim=c(6,36,numdataid))
pos_pos=array(dim=c(6,36,numdataid))
neg_neg=array(dim=c(6,36,numdataid))
type1error=array(dim=c(6,36,numdataid))
type2error=array(dim=c(6,36,numdataid))
outlengstr=array(dim=c(numdataid,100000))#I dont know how many contiguous outages there will be, but i know it is highly unlikely
that there will be over 100,000
day=array(dim=c(numdataid,2))
duplicates=numeric(numdataid)
minperyear=60*24*365
w=1#####TESTS
{start_time=Sys.time()
  for(w in 1:numdataid){#for each DataID
    run=subset(rawdata,rawdata$X3==as.numeric(dataidlist[w])) #run is created for the individual DataID
    run$X2[which(run$X2<0)]=0#clean negatives out and use 0. assumption is that negative usage is due to backfeeding of power in an
outage or a meter malfunction
    if(sum(duplicated(run$X1))>0){duplicates[w]=1#notifies if there is any duplicates in the specific DataID. hypothetically this is a
malfunction and would need to be fixed
    run=run[!duplicated(run$X1),]}#if there was a duplicate, the 1st value is assumped to be correct and everything else is an error.
    run=run[!is.na(run$X1),]#removes the rows that do not have a valid date
    run=run[order(run$X1),]#errors in the dataset save out of cronological order, this puts them in assending order
    xleng=length(run$X1)

    {i=1
     j=1
     while (i<2){chk=as.numeric(run[j,1])
     if (chk%%(60*60*24)==60){
       if (is.na(run[j,2])|run[j,2]==0) {j=j+60*24} #if the first full day starts with a 0 or NA, move to the next day because the begining
of data set is not clean
       else{i=i+1}}
     else {j=j+1}}#this function finds the first position in the data to being clean_date
     k=j
     clean_date=as.numeric(unlist(run[j:xleng,1]))
    }#this function creates dates as intergers and begins with the first full day of data that is not an NA or 0

    xleng=length(clean_date)#new length with trimmed times
    totalmin=1+(clean_date[xleng]-clean_date[1])/60
    day[w,1]=as.Date(clean_date[1], origin="1970-01-01")
    day[w,2]=as.Date(clean_date[xleng], origin="1970-01-01")

    {i=1
     j=1
     clean=numeric(totalmin)
     while (i<totalmin+1){
       if(clean_date[1]+60*(i-1)-clean_date[j]==0){
         if(is.na(run[k,2])){clean[i]=0}
```

```
      else{clean[i]=(run[k,2])}
      k=k+1
      i=i+1
      j=j+1}
     else{clean[i]=0
     i=i+1}}
  }#fills gaps in data with 0 energy usage
  xleng=length(clean)#new length with filled in missing data
  clean=unlist(clean)

  {i=1#where are the 0s?
   j=1
   k=1
   zeros=sort(which(clean==0)) #tells me which rows are 0!
   if(length(zeros)>1){
     for (i in 1:(length(zeros)-1)){
      if (zeros[i]-zeros[i+1]==-1){j=j+1}
      else{outlengstr[w,k]=j
      k=k+1
      j=1}
      outlengstr[w,k]=j}}#last one!
   else{if(length(zeros==1)){outlengstr[w,1]=1}
     else{outlengstr[w,1]=0}}
  }#finds outage lengths and and frequency

  q=36####TEST
  for (q in 1:36){  #for each interval (36 different intervals)
   int_leng_min=possible_int_leng_min[q]#tested interval in "q-th" minutes
   int_day=(60*24)/int_leng_min #how many intervals in a single day
   x=floor(xleng/int_leng_min) #combination rule....new length rounded down
   zeros_new=unique(ceiling(zeros/int_leng_min))
   zeros_new1=zeros_new

   {run_new=(sapply(1:x,function(i){
     o=(sapply(1:int_leng_min,function(j){
      (clean[int_leng_min*i-int_leng_min+j])}))
     sum(o[1:int_leng_min])
   }))
   }#combines usage data by the given interval (total energy usage) !!!!NO LONGER MIN/MIN!!!!!

   run_new1=run_new
   p=1####TEST
   if(length(zeros_new)!=0){
     run_new=run_new1
     {ddays=(as.numeric(clean_date[1])-1302840000)/(60*60*24)
      j=((91-floor(ddays%%(365/4))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule of
15 April 2011 every 3 months...each year add an additional day...each leap year add an additional day.
      chk1=run_new[(j):x]
      if(length(zeros)!=1 &
zeros[length(zeros)]/int_leng_min>j){zchk1=unique(ceiling(zeros[(1+length(which(zeros<j*int_leng_min))):length(zeros)]/int_leng_
min))-j+1}
      else{if ((zeros[1]/int_leng_min)<j){zchk1=numeric()}
        else{zchk1=unique(ceiling(zeros[1]/int_leng_min))-j+1}}#in case there is only one "0" and that zero is below the new start
     }#starts data within a day of the the middle of Jan, Apr, Jul, Oct

     {ddays=(as.numeric(clean_date[1])-1302840000)/(60*60*24)
      j=((182-floor(ddays%%(365/2))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule
of 15 April 2011 every 6 months...each year add an additional day...each leap year add an additional day.
      chk2=run_new[(j):x]
      if(length(zeros)!=1 &
zeros[length(zeros)]/int_leng_min>j){zchk2=unique(ceiling(zeros[(1+length(which(zeros<j*int_leng_min))):length(zeros)]/int_leng_
min))-j+1}
      else{if ((zeros[1]/int_leng_min)<j){zchk2=numeric()}
        else{zchk2=unique(ceiling(zeros[1]/int_leng_min))-j+1}}#in case there is only one "0" and that zero is below the new start
```

```
        #cuts out any zeros that occur before j and re-indexs the values
      }#important for SINE wave of outdoor temp to minimize variance in energy useage. place summer and winter in different
categories
      zscore=array(dim=c(length(zeros_new),6))
      chk=array(dim=c(length(zeros_new),6))
      for(p in 1:6){
        wk=possible_combo_int[p] #combine the data for "p" weeks

        {if(wk==13){run_new=chk1
        x=length(run_new)
        zeros_new=zchk1}
          if(wk==26){run_new=chk2
          x=length(run_new)
          zeros_new=zchk2}
          if(wk==52){run_new=run_new1
          x=length(run_new1)
          zeros_new=zeros_new1}}#alters the data if looking at seasonal or bi-yearly combinations

        combo_yr=ifelse(x>365*int_day,floor(364*int_day/(wk*7)),ceiling(x/(7*wk)))#how many combinations in a year (using 364
days/yr for calculations)
        intmean=numeric(combo_yr)
        intsd=numeric(combo_yr)
        #must use 364 to keep whole weeks, starting and ending on the same day of the week!  The specific month or paticular "number
date" are overlooked
        #each year the combination will continue leaving an additional day to the next year... the following year will leave 2 days to the
next! and so on...
        groups=numeric()
        x_combo=floor(x/int_day)
        x=x_combo*int_day
        if(length(zeros_new)>0){if (tail(zeros_new,n=1)>x){zeros_new=zeros_new[1:(min(which(zeros_new>x))-1)]}}#trim
"zeros_new" as well, if there are zeros in the trim, zscore fails
        n=ceiling(x/(364*int_day))
        run_new=run_new[1:(x)]

        for(i in 1:combo_yr) {groups[i]=list(sapply(1:n,function(y){run_new[int_day*sequence(7*wk)-int_day+i+((y-
1)*int_day*364)]}))}#creates a pattern for the combination to create groups of data
        for (i in 1:combo_yr) {intsd[i]=sd(unlist(groups[i])[(which(unlist(groups[i])>=0))])}
        for (i in 1:combo_yr) {intmean[i]=mean(unlist(groups[i])[(which(unlist(groups[i])>=0))])}#for a given data interval (q), a mean
and SD are created by combining using "p"th weeks

        if(length(zeros_new)!=0){
          for (i in 1:length(zeros_new)) {k=zeros_new[i] #p=4&5 for run_new could be shorter....
          j=ifelse(k%%combo_yr==0,combo_yr,k%%combo_yr)#if the remainer is 0, i want the "combo_yr"th index not "0"th index
          zscore[i,p]=(run_new[k]-intmean[j])/intsd[j]  #use the "new_run" that contains a 0, find its z score away from the mean.
          chk[i,p]=ifelse(run_new[k]==0,1,0)#if run_new=0 then it is already a known outage by simply counting the 0s at any interval
          }#finds the standardized distance way from the mean (defined as zscores) of all the outages
          crit_val[p,q,w]=zscore[order(-zscore[which(is.na(zscore[,p])!=1),p]),p][ceiling(.1*length(which(is.na(zscore[,p])!=1)))]##90%
(or if <10 zeros... ((z-1)/z)% (less than 90%)) of the zeros are at or below this zscore
          crit_val[p,q,w]=ifelse(crit_val[p,q,w]>0,0,crit_val[p,q,w])
          #consider makeing crit value neg or 0... reduces type 2 0s but increases type 1
          #crit_val says everything at or below this value is an outage....additional care must be taken if NAs exist in the zscore array
when calculating errors and crit val because it messes with the sorting vectors.
          store_x[p,q,w]=x
          type1error[p,q,w]=length(which(zscore[,p]>crit_val[p,q,w]))-sum(chk[which(zscore[,p]>crit_val[p,q,w]),p]) #outputs the
number of zeros that are larger than the "critical value", because "zscore" only contains outages,
          #those indicated are not ID'd as outages. also if there are any zscores that do not make the cut BUT have a "run_new" of 0 then
the outage can be ID'd by simply searching for 0s.
          intsdopr=rep(intsd,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
          intmeanopr=rep(intmean,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
          type2error[p,q,w]=ifelse(length(setdiff(which(((run_new-
intmeanopr)/intsdopr)<=crit_val[p,q,w]),zeros_new))==0,0,length(setdiff(which(((run_new-
intmeanopr)/intsdopr)<=crit_val[p,q,w]),zeros_new)))
          neg_neg[p,q,w]=length(which(is.na(zscore[,p])!=1))-type1error[p,q,w]#any outage with a zscore equal to or less than the
critical value is correctly ID'd as an outage (true reject the null)
```

```
            pos_pos[p,q,w]=x-neg_neg[p,q,w]-type1error[p,q,w]-type2error[p,q,w]#all other values are correctly ID'd no outage
            #type2error finds the percentage of non-zero containing usages that are incorrectly highlighted using crit_val (when locating zeros)
          }#IF everything below crit_val was assumed to be an outage... what percentage of highlighted data from the entire base would be an error?
        }#end of P loop
      }#end of STD DEV VS. OUTAGE
    }}#end of W & Q loop
  {out_num=numeric(numdataid)
    out_median=numeric(numdataid)
    out_mean=numeric(numdataid)
    for (i in 1:numdataid)out_num[i]=length(outlengstr[i,which(outlengstr[i,]>0)])
    for (i in 1:numdataid)out_median[i]=median(outlengstr[i,which(outlengstr[i,]>0)])
    for (i in 1:numdataid)out_mean[i]=mean(outlengstr[i,which(outlengstr[i,]>0)])
    out_mean_total=mean(outlengstr[which(outlengstr>0)])
    out_median_total=median(outlengstr[which(outlengstr>0)])
  }#mean and median information about outages
  {test_crit_val=array(dim=c(6,36))
    for (i in 1:6){for (j in 1:36){
      test_crit_val[i,j]=sort(crit_val[i,j,])[ceiling(numdataid*.9)]
    }}
  }#crit_val key metrics and error information

  {write.table(day, file = "C:/Users/Jared/Google Drive/day.txt", sep = "\t", row.names = FALSE, col.names = FALSE)
  write.table(test_crit_val, file = "C:/Users/Jared/Google Drive/test_crit_val.txt", sep = "\t", row.names = FALSE, col.names = FALSE)
  #write.table(day, file = "pecan/day.txt", sep = "\t", row.names = FALSE, col.names = FALSE)#umbunto code
  #write.table(test_crit_val, file = "pecan/test_crit_val.txt", sep = "\t", row.names = FALSE, col.names = FALSE)#umbunto code
  }#print the tables that will be needed in validation and aggregation processes

  time1=Sys.time()-start_time
  time1}#end of model building code

write.table(testdata, file = "C:/Users/Jared/Google Drive/testdata.txt", sep = "\t", row.names = FALSE, col.names = FALSE)
####Graph Code####  PRINT ALL AT 1000 pixels wide!!!!!!!
#plot(x, y, main="title", sub="subtitle", xlab="X-axis label", ylab="y-axix label", xlim=c(xmin, xmax), ylim=c(ymin, ymax))# Specify axis options within plot()
{
{jpeg(filename = paste("crit_val.jpg"),width = 7.5, height = 7.5, units ="in",res = 750)
  plot(c(1,36),c(-1.6,-.8),type="n",main="Critical Values by Combination and Interval", xlab="Interval (minutes)", ylab="Critical Value",xaxt="n")
  axis(side=1, at=1:36, labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
  grid(NULL,NULL, lwd = 2)
  lines((1:(36)),test_crit_val[(1:36)*6-5], col="blue", lty=1, lwd=4, pch=19)
  lines((1:(36)),test_crit_val[(1:36)*6-4], col="red", lty=1, lwd=4, pch=19)
  lines((1:(36)),test_crit_val[(1:36)*6-3], col="green", lty=1, lwd=4, pch=19)
  lines((1:(36)),test_crit_val[(1:36)*6-2], col="black", lty=1, lwd=4, pch=19)
  lines((1:(36)),test_crit_val[(1:36)*6-1], col="orange", lty=1, lwd=4, pch=19)
  lines((1:(36)),test_crit_val[(1:36)*6], col="yellow", lty=1, lwd=4, pch=19)
  points((1:(36)),test_crit_val[(1:36)*6-5], col="blue", pch=19, cex=1)
  points((1:(36)),test_crit_val[(1:36)*6-4], col="red", pch=19, cex=1)
  points((1:(36)),test_crit_val[(1:36)*6-3], col="green", pch=19, cex=1)
  points((1:(36)),test_crit_val[(1:36)*6-2], col="black", pch=19, cex=1)
  points((1:(36)),test_crit_val[(1:36)*6-1], col="orange", pch=19, cex=1)
  points((1:(36)),test_crit_val[(1:36)*6], col="yellow",  pch=19, cex=1)
  legend(1,-1.2,c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 Weeks/Annually"),cex=1,y.intersp=1,title=("SD/Mean Combinations"),
         lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=c("blue","red","green","black","orange","yellow"))#imports a legend for the plot
dev.off()}#shows the specified 10 percentile "Critical Values"
}#end of graphics code
################VALIDATE################
#requires "test_crit_val" from Model_Building.R
library(readr)
```

```
testdata <- read_csv("pecan/testdata.txt", col_names = FALSE)#import the test data... format= '%Y-%m-%d %H:%M:%S-%z'
colnames(testdata)=(c("X1","X2","X3"))
{dataidlist_test=unlist(unique(testdata[,3]))
  numdataid=length(dataidlist_test)

possible_int_leng_min=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,72
0,1440)
  possible_combo_int=c(1,2,4,13,26,52)
  store_x_test=array(dim=c(6,36,numdataid))
  pos_pos_test=array(dim=c(6,36,numdataid))
  neg_neg_test=array(dim=c(6,36,numdataid))
  typeIIerror_test=array(dim=c(6,36,numdataid))
  typeIerror_test=array(dim=c(6,36,numdataid))
  outlengstr_test=array(dim=c(numdataid,1000000))
  day_test=array(dim=c(numdataid,2))
  duplicates_test=numeric(numdataid)
  minperyear=60*24*365}#creates the variables
w=1#####TESTS
{start_time=Sys.time()
  for(w in 1:numdataid){#for each DataID
    run=subset(testdata,testdata$X3==as.numeric(dataidlist_test[w])) #run is created for the individual DataID
    run$X2[which(run$X2<0)]=0#clean negatives out and use 0. assumption is that negative usage is due to backfeeding of power in an
outage or a meter malfunction
    if(sum(duplicated(as.numeric(run$X1)))>0){duplicates_test[w]=1#notifies if there is any duplicates in the specific DataID.
hypothetically this is a malfunction and would need to be fixed
    run=run[!duplicated(run$X1),]}#if there was a duplicate the 1st value is assumped to be correct and everything else is an error.
    run=run[!is.na(run$X1),]#removes the rows that do not have a valid date
    run=run[order(run$X1),]#errors in the dataset save out of cronological order, this puts them in assending order
    xleng=length(run$X1)

    {i=1
     j=1
     while (i<2){chk=as.numeric(run[j,1])
     if (chk%%(60*60*24)==60){
       if (is.na(run[j,2])|run[j,2]==0) {j=j+60*24} #if the first full day starts with a 0 or NA, move to the next day because the begining
of data set is not clean
       else{i=i+1}}
     else {j=j+1}}#this function finds the first position in the data to being clean_date
     k=j
     clean_date=as.numeric(unlist(run[j:xleng,1]))
    }#this function creates dates as intergers and begins with the first full day of data that is not an NA or 0

    xleng=length(clean_date)#new length with trimmed times
    totalmin=1+(clean_date[xleng]-clean_date[1])/60
    day_test[w,1]=as.Date(clean_date[1], origin="1970-01-01")
    day_test[w,2]=as.Date(clean_date[xleng], origin="1970-01-01")

    {i=1
     j=1
     clean=numeric(totalmin)
     while (i<totalmin+1){
       if(clean_date[1]+60*(i-1)-clean_date[j]==0){
         if(is.na(run[k,2])){clean[i]=0}
         else{clean[i]=(run[k,2])}
         k=k+1
         i=i+1
         j=j+1}
       else{clean[i]=0
       i=i+1}}
    }#fills gaps in data with 0 energy usage
    xleng=length(clean)#new length with filled in missing data
    clean=unlist(clean)

    {i=1#where are the 0s?
```

```
    j=1
    k=1
    zeros=sort(which(clean==0)) #tells me which rows are 0!
    if(length(zeros)>1){
      for (i in 1:(length(zeros)-1)){
        if (zeros[i]-zeros[i+1]==-1){j=j+1}
        else{outlengstr_test[w,k]=j
        k=k+1
        j=1}
        outlengstr_test[w,k]=j}}#last one!
    else{if(length(zeros==1)){outlengstr_test[w,1]=1}
      else{outlengstr_test[w,1]=0}}
  }#finds outage lengths and and frequency

  q=1####TEST
  for (q in 1:36){  #for each interval (36 different intervals)
    int_leng_min=possible_int_leng_min[q]#tested interval in "q-th" minutes
    int_day=(60*24)/int_leng_min #how many intervals in a single day
    x=floor(xleng/int_leng_min) #combination rule....new length rounded down
    zeros_new=unique(ceiling(zeros/int_leng_min))
    zeros_new1=zeros_new

    {run_new=(sapply(1:x,function(i){
      o=(sapply(1:int_leng_min,function(j){
        (clean[int_leng_min*i-int_leng_min+j])}))
      sum(o[1:int_leng_min])
    }))
    }#combines usage data by the given interval (total energy usage) !!!!NO LONGER MIN/MIN!!!!!

    run_new1=run_new
    p=1####TEST
    if(length(zeros_new)!=0){
      run_new=run_new1
      {ddays=(as.numeric(clean_date[1])-1302840000)/(60*60*24)
        j=((91-floor(ddays%%(365/4))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule of
15 April 2011 every 3 months...each year add an additional day...each leap year add an additional day.
        chk1=run_new[(j):x]
        if(length(zeros)!=1 &
zeros[length(zeros)]/int_leng_min>j){zchk1=unique(ceiling(zeros[(1+length(which(zeros<j*int_leng_min))):length(zeros)]/int_leng_
min))-j+1}
        else{if ((zeros[1]/int_leng_min)<j){zchk1=numeric()}
          else{zchk1=unique(ceiling(zeros[1]/int_leng_min))-j+1}}#in case there is only one "0" and that zero is below the new start
      }#starts data within a day of the the middle of Jan, Apr, Jul, Oct

      {ddays=(as.numeric(clean_date[1])-1302840000)/(60*60*24)
        j=((182-floor(ddays%%(365/2))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule
of 15 April 2011 every 6 months...each year add an additional day...each leap year add an additional day.
        chk2=run_new[(j):x]
        if(length(zeros)!=1 &
zeros[length(zeros)]/int_leng_min>j){zchk2=unique(ceiling(zeros[(1+length(which(zeros<j*int_leng_min))):length(zeros)]/int_leng_
min))-j+1}
        else{if ((zeros[1]/int_leng_min)<j){zchk2=numeric()}
          else{zchk2=unique(ceiling(zeros[1]/int_leng_min))-j+1}}#in case there is only one "0" and that zero is below the new start
        #cuts out any zeros that occur before j and re-indexs the values
      }#important for SINE wave of outdoor temp to minimize variance in energy useage. place summer and winter in different
categories
      zscore=array(dim=c(length(zeros_new),6))
      chk=array(dim=c(length(zeros_new),6))
      for(p in 1:6){
        wk=possible_combo_int[p] #combine the data for "p" weeks

        {if(wk==13){run_new=chk1
        x=length(run_new)
        zeros_new=zchk1}
```

```
    if(wk==26){run_new=chk2
    x=length(run_new)
    zeros_new=zchk2}
    if(wk==52){run_new=run_new1
    x=length(run_new1)
    zeros_new=zeros_new1}}#alters the data if looking at seasonal or bi-yearly combinations

   combo_yr=ifelse(x>365*int_day,floor(364*int_day/(wk*7)),ceiling(x/(7*wk)))#how many combinations in a year (using 364
days/yr for calculations)
   intmean=numeric(combo_yr)
   intsd=numeric(combo_yr)
   #must use 364 to keep whole weeks, starting and ending on the same day of the week!  The specific month or paticular "number
date" are overlooked
   #each year the combination will continue leaving an additional day to the next year... the following year will leave 2 days to the
next! and so on...
   groups=numeric()
   x_combo=floor(x/int_day)
   x=x_combo*int_day
   if(length(zeros_new)>0){if (tail(zeros_new,n=1)>x){zeros_new=zeros_new[1:(min(which(zeros_new>x))-1)]}}#trim
"zeros_new" as well, if there are zeros in the trim, zscore fails
   n=ceiling(x/(364*int_day))
   run_new=run_new[1:(x)]

   for(i in 1:combo_yr) {groups[i]=list(sapply(1:n,function(y){run_new[int_day*sequence(7*wk)-int_day+i+((y-
1)*int_day*364)]}))}#creates a pattern for the combination to create groups of data
   for (i in 1:combo_yr) {intsd[i]=sd(unlist(groups[i])[(which(unlist(groups[i])>=0))])}
   for (i in 1:combo_yr) {intmean[i]=mean(unlist(groups[i])[(which(unlist(groups[i])>=0))])}#for a given data interval (q), a mean
and SD are created by combining using "p"th weeks

   if(length(zeros_new)!=0){
    for (i in 1:length(zeros_new)) {k=zeros_new[i] #p=4&5 for run_new could be shorter....
    j=ifelse(k%%combo_yr==0,combo_yr,k%%combo_yr)#if the remainer is 0, i want the "combo_yr"th index not "0"th index
    zscore[i,p]=(run_new[k]-intmean[j])/intsd[j]  #use the "new_run" that contains a 0, find its z score away from the mean.
    chk[i,p]=ifelse(run_new[k]==0,1,0)#if run_new=0 then it is already a known outage by simply counting the 0s at any interval
    }#finds the standardized distance way from the mean (defined as zscores) of all the outages
    store_x_test[p,q,w]=x
    typeIIerror_test[p,q,w]=length(which(zscore[,p]>test_crit_val[p,q]))-sum(chk[which(zscore[,p]>test_crit_val[p,q]),p])
#outputs the number of zeros that are larger than the "critical value", because "zscore" only contains outages,
    #those indicated are not ID'd as outages. also if there are any zscores that do not make the cut BUT have a "run_new" of 0 then
the outage can be ID'd by simply searching for 0s.
    intsdopr=rep(intsd,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
    intmeanopr=rep(intmean,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
    typeIerror_test[p,q,w]=ifelse(length(setdiff(which(((run_new-
intmeanopr)/intsdopr)<=test_crit_val[p,q]),zeros_new))==0,0,length(setdiff(which(((run_new-
intmeanopr)/intsdopr)<=test_crit_val[p,q]),zeros_new)))
    neg_neg_test[p,q,w]=length(which(is.na(zscore[,p])!=1))-typeIIerror_test[p,q,w]#any outage with a zscore equal to or less than
the critical value is correctly ID'd as an outage (true reject the null)
    pos_pos_test[p,q,w]=x-neg_neg_test[p,q,w]-typeIIerror_test[p,q,w]-typeIerror_test[p,q,w]#all other values are correctly ID'd
no outage
    #typeIerror finds the percentage of non-zero containing usages that are incorrectly highlighted using crit_val (when locating
zeros)
   }#IF everything below crit_val was assumed to be an outage... what percentage of highlighted data from the entire base would
be an error?
  }#end of P loop
 }#end of STD DEV VS. OUTAGE
}}#end of W & Q loop
{out_num_test=numeric(numdataid)
 out_median_test=numeric(numdataid)
 out_mean_test=numeric(numdataid)
 for (i in 1:numdataid)out_num_test[i]=length(outlengstr_test[i,which(outlengstr_test[i,]>0)])
 for (i in 1:numdataid)out_median_test[i]=median(outlengstr_test[i,which(outlengstr_test[i,]>0)])
 for (i in 1:numdataid)out_mean_test[i]=mean(outlengstr_test[i,which(outlengstr_test[i,]>0)])
 out_mean_total_test=mean(outlengstr_test[which(outlengstr_test>0)])
 out_median_total_test=median(outlengstr_test[which(outlengstr_test>0)])
```

```
  }
  time1=Sys.time()-start_time
  time1}#end of model validiation

#PREP FOR MATLAB/MITLL
mean((typeIerror_test[6,11,])/store_x_test[6,11,])#5.04%
mean((typeIerror_test[6,1,])/store_x_test[6,1,])#3.3%
mean((typeIIerror_test[6,11,])/store_x_test[6,11,])#.01%
mean((typeIIerror_test[6,1,])/store_x_test[6,1,])#0%

(sum(typeIerror_test[6,11,])/sum(neg_neg_test[6,11,]+typeIerror_test[6,11,]))#41.2%
(sum(typeIerror_test[6,1,])/sum(neg_neg_test[6,1,]+typeIerror_test[6,1,]))#33.3%
(sum(typeIIerror_test[6,11,])/sum(neg_neg_test[6,11,]+typeIerror_test[6,11,]))#.1%
(sum(typeIIerror_test[6,1,])/sum(neg_neg_test[6,1,]+typeIerror_test[6,1,]))#0.0%
(sum(typeIIerror_test[6,11,])/sum(neg_neg_test[6,11,]+typeIIerror_test[6,11,]))#.2%
(sum(typeIIerror_test[6,1,])/sum(neg_neg_test[6,1,]+typeIIerror_test[6,1,]))#0.0%

#if typeIIerrors are fix before anyone knows... we will never know they happend
#what is correctly ID'd?  ID outages higher isnt always the best thing....missed outages between 15 and and 1 min are not substantial

####Graph Code####  PRINT ALL AT 1000 pixels wide!!!!!!!
par(mfrow=c(1,1)) #reset to 1 by 1 graph
#plot(x, y, main="title", sub="subtitle", xlab="X-axis label", ylab="y-axix label", xlim=c(xmin, xmax), ylim=c(ymin, ymax))# Specify axis options within plot()

{plot(c(1,36),c(0,12),type="n",main="Validation Type II Errors as % of Known Outages", xlab="Interval (minutes)", ylab="Percent Errors (%)",xaxt="n")
  axis(side=1, at=1:36, labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
  grid(NULL,NULL, lwd = 2)
  title(" \"X% of actual outages that were not identified\" ",line = .5,col.main="dark blue",cex.main=.9)

lines((1:(36)),100*rowSums(typeIIerror_test[1,,],na.rm=TRUE)/(rowSums(typeIIerror_test[1,,],na.rm=TRUE)+rowSums(neg_neg_test[1,,],na.rm=TRUE)), col="blue", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIIerror_test[2,,],na.rm=TRUE)/(rowSums(typeIIerror_test[2,,],na.rm=TRUE)+rowSums(neg_neg_test[2,,],na.rm=TRUE)), col="red", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIIerror_test[3,,],na.rm=TRUE)/(rowSums(typeIIerror_test[3,,],na.rm=TRUE)+rowSums(neg_neg_test[3,,],na.rm=TRUE)), col="green", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIIerror_test[4,,],na.rm=TRUE)/(rowSums(typeIIerror_test[4,,],na.rm=TRUE)+rowSums(neg_neg_test[4,,],na.rm=TRUE)), col="black", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIIerror_test[5,,],na.rm=TRUE)/(rowSums(typeIIerror_test[5,,],na.rm=TRUE)+rowSums(neg_neg_test[5,,],na.rm=TRUE)), col="orange", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIIerror_test[6,,],na.rm=TRUE)/(rowSums(typeIIerror_test[6,,],na.rm=TRUE)+rowSums(neg_neg_test[6,,],na.rm=TRUE)), col="yellow", lty=1, lwd=4, pch=19)
points((1:(36)),100*rowSums(typeIIerror_test[1,,],na.rm=TRUE)/(rowSums(typeIIerror_test[1,,],na.rm=TRUE)+rowSums(neg_neg_test[1,,],na.rm=TRUE)), col="blue", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIIerror_test[2,,],na.rm=TRUE)/(rowSums(typeIIerror_test[2,,],na.rm=TRUE)+rowSums(neg_neg_test[2,,],na.rm=TRUE)), col="red", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIIerror_test[3,,],na.rm=TRUE)/(rowSums(typeIIerror_test[3,,],na.rm=TRUE)+rowSums(neg_neg_test[3,,],na.rm=TRUE)), col="green", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIIerror_test[4,,],na.rm=TRUE)/(rowSums(typeIIerror_test[4,,],na.rm=TRUE)+rowSums(neg_neg_test[4,,],na.rm=TRUE)), col="black", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIIerror_test[5,,],na.rm=TRUE)/(rowSums(typeIIerror_test[5,,],na.rm=TRUE)+rowSums(neg_neg_test[5,,],na.rm=TRUE)), col="orange", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIIerror_test[6,,],na.rm=TRUE)/(rowSums(typeIIerror_test[6,,],na.rm=TRUE)+rowSums(neg_neg_test[6,,],na.rm=TRUE)), col="yellow",  pch=19, cex=1)
  legend(0,11.5,c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 Weeks/Annually"),cex=.7,y.intersp=1,title=("SD/Mean Combinations"),
        lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=c("blue","red","green","black","orange","yellow"))#imports a legend for the plot
}#shows the Type II errors as a percentage of the known outages

####NOT USED
{plot(c(1,36),c(0,15),type="n",main="Validation Type I Errors as % of Total Sample Space", xlab="Interval (minutes)", ylab="Percent Errors (%)",xaxt="n")
```

```
 axis(side=1, at=1:36,
labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
 grid(NULL,NULL, lwd = 2)
 title(" \"X% of the total sample space was incorrectly identified as an outage\" ",line = .5,col.main="dark blue",cex.main=.9)
 lines((1:(36)),100*rowSums(typeIerror_test[1,,],na.rm=TRUE)/rowSums(store_x_test[1,,],na.rm=TRUE), col="blue", lty=1, lwd=4,
pch=19)
 lines((1:(36)),100*rowSums(typeIerror_test[2,,],na.rm=TRUE)/rowSums(store_x_test[2,,],na.rm=TRUE), col="red", lty=1, lwd=4,
pch=19)
 lines((1:(36)),100*rowSums(typeIerror_test[3,,],na.rm=TRUE)/rowSums(store_x_test[3,,],na.rm=TRUE), col="green", lty=1,
lwd=4, pch=19)
 lines((1:(36)),100*rowSums(typeIerror_test[4,,],na.rm=TRUE)/rowSums(store_x_test[4,,],na.rm=TRUE), col="black", lty=1, lwd=4,
pch=19)
 lines((1:(36)),100*rowSums(typeIerror_test[5,,],na.rm=TRUE)/rowSums(store_x_test[5,,],na.rm=TRUE), col="orange", lty=1,
lwd=4, pch=19)
 lines((1:(36)),100*rowSums(typeIerror_test[6,,],na.rm=TRUE)/rowSums(store_x_test[6,,],na.rm=TRUE), col="yellow", lty=1,
lwd=4, pch=19)
 points((1:(36)),100*rowSums(typeIerror_test[1,,],na.rm=TRUE)/rowSums(store_x_test[1,,],na.rm=TRUE), col="blue", pch=19,
cex=1)
 points((1:(36)),100*rowSums(typeIerror_test[2,,],na.rm=TRUE)/rowSums(store_x_test[2,,],na.rm=TRUE), col="red", pch=19,
cex=1)
 points((1:(36)),100*rowSums(typeIerror_test[3,,],na.rm=TRUE)/rowSums(store_x_test[3,,],na.rm=TRUE), col="green", pch=19,
cex=1)
 points((1:(36)),100*rowSums(typeIerror_test[4,,],na.rm=TRUE)/rowSums(store_x_test[4,,],na.rm=TRUE), col="black", pch=19,
cex=1)
 points((1:(36)),100*rowSums(typeIerror_test[5,,],na.rm=TRUE)/rowSums(store_x_test[5,,],na.rm=TRUE), col="orange", pch=19,
cex=1)
 points((1:(36)),100*rowSums(typeIerror_test[6,,],na.rm=TRUE)/rowSums(store_x_test[6,,],na.rm=TRUE), col="yellow",  pch=19,
cex=1)
 legend(1,15,c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52
Weeks/Annually"),cex=.7,y.intersp=1,title=("SD/Mean Combinations"),
     lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=c("blue","red","green","black","orange","yellow"))#imports a legend for the plot
}#shows the Type I errors as a percentage of the total sample space

{plot(c(1,36),c(0,80),type="n",main="Validation Type I Errors as % of All Identified Outages", xlab="Interval (minutes)",
ylab="Percent Errors (%)",xaxt="n")
 axis(side=1, at=1:36,
labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
 grid(NULL,NULL, lwd = 2)
 title(" \"X% of the identified outages were incorrectly specified as an outage\" ",line = .5,col.main="dark blue",cex.main=.9)
lines((1:(36)),100*rowSums(typeIerror_test[1,,],na.rm=TRUE)/(rowSums(typeIerror_test[1,,],na.rm=TRUE)+rowSums(neg_neg_test[
1,,],na.rm=TRUE)), col="blue", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIerror_test[2,,],na.rm=TRUE)/(rowSums(typeIerror_test[2,,],na.rm=TRUE)+rowSums(neg_neg_test[
2,,],na.rm=TRUE)), col="red", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIerror_test[3,,],na.rm=TRUE)/(rowSums(typeIerror_test[3,,],na.rm=TRUE)+rowSums(neg_neg_test[
3,,],na.rm=TRUE)), col="green", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIerror_test[4,,],na.rm=TRUE)/(rowSums(typeIerror_test[4,,],na.rm=TRUE)+rowSums(neg_neg_test[
4,,],na.rm=TRUE)), col="black", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIerror_test[5,,],na.rm=TRUE)/(rowSums(typeIerror_test[5,,],na.rm=TRUE)+rowSums(neg_neg_test[
5,,],na.rm=TRUE)), col="orange", lty=1, lwd=4, pch=19)
lines((1:(36)),100*rowSums(typeIerror_test[6,,],na.rm=TRUE)/(rowSums(typeIerror_test[6,,],na.rm=TRUE)+rowSums(neg_neg_test[
6,,],na.rm=TRUE)), col="yellow", lty=1, lwd=4, pch=19)
points((1:(36)),100*rowSums(typeIerror_test[1,,],na.rm=TRUE)/(rowSums(typeIerror_test[1,,],na.rm=TRUE)+rowSums(neg_neg_tes
t[1,,],na.rm=TRUE)), col="blue", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIerror_test[2,,],na.rm=TRUE)/(rowSums(typeIerror_test[2,,],na.rm=TRUE)+rowSums(neg_neg_tes
t[2,,],na.rm=TRUE)), col="red", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIerror_test[3,,],na.rm=TRUE)/(rowSums(typeIerror_test[3,,],na.rm=TRUE)+rowSums(neg_neg_tes
t[3,,],na.rm=TRUE)), col="green", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIerror_test[4,,],na.rm=TRUE)/(rowSums(typeIerror_test[4,,],na.rm=TRUE)+rowSums(neg_neg_tes
t[4,,],na.rm=TRUE)), col="black", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIerror_test[5,,],na.rm=TRUE)/(rowSums(typeIerror_test[5,,],na.rm=TRUE)+rowSums(neg_neg_tes
t[5,,],na.rm=TRUE)), col="orange", pch=19, cex=1)
points((1:(36)),100*rowSums(typeIerror_test[6,,],na.rm=TRUE)/(rowSums(typeIerror_test[6,,],na.rm=TRUE)+rowSums(neg_neg_tes
t[6,,],na.rm=TRUE)), col="yellow",  pch=19, cex=1)
```

```
 legend(2.5,30,c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 
Weeks/Annually"),cex=.7,y.intersp=1,title=("SD/Mean Combinations"),
       lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=c("blue","red","green","black","orange","yellow"))#imports a legend for the plot
}#shows the Type I errors as a percentage of all identified outages (good negatives and bad negatives)

{plot(c(1,36),c(75,100),type="n",main="Validation \"Good Match\" as % of Total Sample Space", xlab="Interval (minutes)", 
ylab="Percent Errors (%)",xaxt="n")
 axis(side=1, at=1:36, 
labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
 grid(NULL,NULL, lwd = 2)
 title(" \"X% of the total sample space that contains no errors (\"Good Match\") ",line = .5,col.main="dark blue",cex.main=.9)
 chk=c("blue","red","green","black","orange","yellow")
 for (i in 
1:6){lines((1:(36)),100*(rowSums(neg_neg_test[i,,],na.rm=TRUE)+rowSums(pos_pos_test[i,,],na.rm=TRUE))/(rowSums(store_x_tes
t[i,,],na.rm=TRUE)), col=chk[i], lty=1, lwd=4, pch=19)

points((1:(36)),100*(rowSums(neg_neg_test[i,,],na.rm=TRUE)+rowSums(pos_pos_test[i,,],na.rm=TRUE))/(rowSums(store_x_test[i,,
],na.rm=TRUE)), col=chk[i], pch=19, cex=1)}
 legend(22,82,c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 
Weeks/Annually"),cex=.7,y.intersp=1,title=("SD/Mean Combinations"),
       lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=c("blue","red","green","black","orange","yellow"))#imports a legend for the plot
}#shows how good at ID'ing the outages as a whole by compairing Positive ID with total ID'd


######NOT USED
{plot(c(1,36),c(0,15),type="n",main="Validation Total Errors as % of Total Sample Space", xlab="Interval (minutes)", ylab="Percent 
Errors (%)",xaxt="n")
 axis(side=1, at=1:36, 
labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
 grid(NULL,NULL, lwd = 2)
 title(" \"X% of the total sample space that contains errors ",line = .5,col.main="dark blue",cex.main=.9)
 chk=c("blue","red","green","black","orange","yellow")
 for (i in 1:6){lines((1:(36)),100-
(100*(rowSums(neg_neg_test[i,,],na.rm=TRUE)+rowSums(pos_pos_test[i,,],na.rm=TRUE))/(rowSums(store_x_test[i,,],na.rm=TRUE
))), col=chk[i], lty=1, lwd=4, pch=19)
  points((1:(36)),100-
(100*(rowSums(neg_neg_test[i,,],na.rm=TRUE)+rowSums(pos_pos_test[i,,],na.rm=TRUE))/(rowSums(store_x_test[i,,],na.rm=TRUE
))), col=chk[i], pch=19, cex=1)}
 legend(25,15,c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 
Weeks/Annually"),cex=.7,y.intersp=1,title=("SD/Mean Combinations"),
       lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=c("blue","red","green","black","orange","yellow"))#imports a legend for the plot
}#shows how good at ID'ing the outages as a whole by compairing Errors

##########bar charts
{par(mfrow=c(3,2),xpd=F,mar=c(5,4,4,4))
for(i in 1:6){chk=order(-
(colSums(typeIIerror_test[i,,],na.rm=TRUE)+colSums(typeIerror_test[i,,],na.rm=TRUE))/colSums(store_x_test[i,,],na.rm=TRUE))
chk1=c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 Weeks/Annually")
barplot(100*matrix(c(colSums(typeIIerror_test[i,,],na.rm=TRUE)[chk],colSums(typeIerror_test[i,,],na.rm=TRUE)[chk])/(colSums(sto
re_x_test[i,,],na.rm=TRUE)[chk]),nrow=2, byrow=TRUE), col=c("darkblue","red"),
      ylim=c(0,50),space=0, axes=TRUE, main=paste("Pareto Chart of Errors by DataID as % of Total Values \nCombine 
every",chk1[i]), xlab="Data ID index", ylab="Percent (%)", legend=c("Type II Error","Type I Error"),args.legend=list(bty = 
"n",cex=.9))
grid()
axis(1,at=((1:11)*2)-1.5,labels=chk[((1:11)*2)-1],las=2)
axis(1,at=(1:11)*2-.5,labels=paste(chk[(1:11)*2],"    "),las=2,tck=-.05)}}#shows where and how big the errors are by percentage for 
all combinations

{par(mfrow=c(3,2))
for(i in 1:6){chk=order(-
(colSums(typeIIerror_test[i,,],na.rm=TRUE)+colSums(typeIerror_test[i,,],na.rm=TRUE))/colSums(store_x_test[i,,],na.rm=TRUE))
chk1=c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 Weeks/Annually")
barplot(out_mean_test[chk]/60, col=c("red"), ylim=c(0,800),space=0, axes=TRUE, main=paste("Mean Outage Length for Pareto 
Chart of Errors \nby DataID (", chk1[i],")"),
```

```
    xlab="Data ID index", ylab="Mean (hours)", legend="Mean Outage Length",args.legend=list(x=22, y=800,bty = "n",cex=.9))
grid()
axis(1,at=((1:11)*2)-1.5,labels=chk[((1:11)*2)-1],las=2)
axis(1,at=(1:11)*2-.5,labels=paste(chk[(1:11)*2],"    "),las=2,tck=-.05)}}#shows the mean outage length in the order found using total errors by percentage
#COMBINE TO A SINGLE PLOT 2 wide by 3 deep

{par(mfrow=c(3,2))
 for(i in 1:6){chk=order(-(colSums(typeIIerror_test[i,,],na.rm=TRUE)+colSums(typeIerror_test[i,,],na.rm=TRUE))/colSums(store_x_test[i,,],na.rm=TRUE))
 chk1=c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 Weeks/Annually")
 barplot((day_test[chk,2]-day_test[chk,1])/(60*24*60*365), col=c("green"), ylim=c(0,(2500/365)),space=0, axes=TRUE, main=paste("Years of Data for Pareto Chart of Errors \nby DataID (", chk1[i],")"),
     xlab="Data ID index", ylab="Length (years)")
 grid()
 axis(1,at=((1:11)*2)-1.5,labels=chk[((1:11)*2)-1],las=2)
 axis(1,at=(1:11)*2-.5,labels=paste(chk[(1:11)*2],"    "),las=2,tck=-.05)}}#shows the mean outage length in the order found using total errors by percentage
#COMBINE TO A SINGLE PLOT 2 wide by 3 deep

{numdataid=22
par(mfrow=c(1,1))
chk1=numeric(numdataid)
 for(i in 1:numdataid)
chk1[i]=(sum(typeIIerror_test[,,i],na.rm=TRUE)+sum(typeIerror_test[,,i],na.rm=TRUE))/sum(store_x_test[,,i],na.rm=TRUE)
 chk=order(-chk1)
 chk1=chk1[chk]
 barplot(100*chk1, col=c("red"),ylim=c(0,25),space=0, axes=TRUE, main=paste("Pareto Chart of Total Errors by DataID as % of Total Values for All Combinations"),
     xlab="Data ID index", ylab="Percent (%)", legend=c("Total (Type I & II) Error"))
 grid()
 axis(1,at=((1:11)*2)-1.5,labels=chk[((1:11)*2)-1],las=2)
 axis(1,at=(1:11)*2-.5,labels=paste(chk[(1:11)*2],"    "),las=2,tck=-.05)}
```

Ellsworth Air Force Base

```
################VALIDATE################
#requires "test_crit_val" from critval_sensitivity.R
#library(readr)
#EAFBrawdata <- read_delim("pecan/EAFBrawdata.txt","\t", escape_double = FALSE, col_types = cols(DATE = col_datetime(format = "%m/%d/%Y")), trim_ws = TRUE)#import the test data... format= %Y-%m-%d
colnames(EAFBrawdata)=(c("X1","X2","X3"))#data is in daily intervals already!!!
{dataidlist_EAFB=unlist(unique(EAFBrawdata[,3]))
 numdataid=length(dataidlist_EAFB)
 possible_combo_int=c(1,2,4,13,26,52)
 store_x_EAFB=array(dim=c(6,numdataid))
 pos_pos_EAFB=array(dim=c(6,numdataid))
 neg_neg_EAFB=array(dim=c(6,numdataid))
 typeIIerror_EAFB=array(dim=c(6,numdataid))
 typeIerror_EAFB=array(dim=c(6,numdataid))
 outages_EAFB=array(dim=c(6,numdataid))
 outlengstr_EAFB=array(dim=c(numdataid,200))
 outloc_EAFB=array(dim=c(numdataid,20000))
 day_EAFB=array(dim=c(numdataid,2))
 duplicates_EAFB=numeric(numdataid)
 minperyear=60*24*365
 q=36 #q is the index for which interval width. EAFB only has dayily readings so only interval 36 (1440 minutes) is used
}#creates the variables
w=12#####TESTS
{start_time=Sys.time()
 for(w in 1:numdataid){
  run=subset(EAFBrawdata,EAFBrawdata$X3==as.numeric(dataidlist_EAFB[w])) #run is created for the individual DataID
```

```
run$X2[which(run$X2<0)]=0#clean negatives out and use 0. assumption is that negative usage is due to backfeeding of power in an outage or a meter malfunction
  if(sum(duplicated(as.numeric(run$X1)))>0){duplicates_EAFB[w]=1#notifies if there is any duplicates in the specific DataID. hypothetically this is a malfunction and would need to be fixed
  run=run[!duplicated(run$X1),]}#if there was a duplicate the 1st value is assumped to be correct and everything else is an error.
  run=run[!is.na(run$X1),]#removes the rows that do not have a valid date
  run=run[order(run$X1),]#errors in the dataset save out of cronological order, this puts them in assending order
  xleng=length(run$X1)

  {i=1
   j=1
   while (i<2){chk=as.numeric(run[j,1])
   if (chk%%(60*60*24)==0){
     if (is.na(run[j,2])|run[j,2]==0) {j=j+1} #if the first day starts with a 0 or NA, move to the next day because the begining of data set is not clean
     else{i=i+1}}
   else {j=j+1}}#this function finds the first position in the data to being clean_date
   k=j
   clean_date=as.numeric(unlist(run[j:xleng,1]))
  }#this function creates dates as intergers and begins with the first full day of data that is not an NA or 0

  xleng=length(clean_date)#new length with trimmed times
  totalday=1+(clean_date[xleng]-clean_date[1])/(60*60*24)
  day_EAFB[w,1]=as.Date(clean_date[1], origin="1970-01-01")
  day_EAFB[w,2]=as.Date(clean_date[xleng], origin="1970-01-01")

  {i=1
   j=1
   clean=numeric(totalday)
   while (i<totalday+1){
     if(clean_date[1]+(60*24*60)*(i-1)-clean_date[j]==0){
       if(is.na(run[k,2])){clean[i]=0}
       else{clean[i]=(run[k,2])}
       k=k+1
       i=i+1
       j=j+1}
     else{clean[i]=0
     i=i+1}}
  }#fills gaps in data with 0 energy usage
  xleng=length(clean)#new length with filled in missing data
  clean=unlist(clean)

  #dates 2015-03-09 and 03-10 are days with missing data
  {i=1#where are the 0s?
   j=1
   k=1
   zeros=sort(which(clean==0)) #tells me which rows are 0!
   if(length(zeros)>1){
     for (i in 1:(length(zeros)-1)){
       if (zeros[i]-zeros[i+1]==-1){j=j+1}
       else{outlengstr_EAFB[w,k]=j
       k=k+1
       j=1}
       outlengstr_EAFB[w,k]=j}}#last one!
   else{if(length(zeros==1)){outlengstr_EAFB[w,1]=1}
     else{outlengstr_EAFB[w,1]=0}}
  }#finds outage lengths and and frequency
  ##start collecting dates when outages happen
  for (i in 1:length(zeros)) outloc_EAFB[w,i]=clean_date[1]+(zeros[i]-1)*60*60*24

  #Usually we would change the interval length but 1 day is the maximum interval tested in this research

  {int_leng_min=1440#only full day interval
   int_day=(60*24)/int_leng_min #how many intervals in a single day
```

```
    x=xleng
    zeros_new=zeros
    zeros_new1=zeros_new
    run_new=clean}#full day interval

   run_new1=run_new
   p=1####TEST
   if(length(zeros_new)!=0){
    run_new=run_new1
    {ddays=(as.numeric(clean_date[1])-1208232000)/(60*60*24)  #use of int_day in next line makes this function also usage for daily
energy intervals
     j=((91-floor(ddays%%(365/4))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule of
15 April 2008 every 3 months...each year add an additional day...each leap year add an additional day.
     chk1=run_new[(j):x]
     zchk1=which(chk1==0)#in case there is only one "0" and that zero is below the new start
    }#starts data within a day of the the middle of Jan, Apr, Jul, Oct

    {ddays=(as.numeric(clean_date[1])-1208232000)/(60*60*24)
     j=((182-floor(ddays%%(365/2))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule of
15 April 2008 every 6 months...each year add an additional day...each leap year add an additional day.
     chk2=run_new[(j):x]
     zchk2=which(chk2==0)#in case there is only one "0" and that zero is below the new start
     #cuts out any zeros that occur before j and re-indexs the values
    }#important for SINE wave of outdoor temp to minimize variance in energy useage. place summer and winter in different
categories
    zscore=array(dim=c(length(zeros_new),6))
    chk=array(dim=c(length(zeros_new),6))
    for(p in 1:6){
     wk=possible_combo_int[p] #combine the data for "p" weeks

     {if(wk==13){run_new=chk1
     x=length(run_new)
     zeros_new=zchk1}
      if(wk==26){run_new=chk2
      x=length(run_new)
      zeros_new=zchk2}
      if(wk==52){run_new=run_new1
      x=length(run_new1)
      zeros_new=zeros_new1}}#alters the data if looking at seasonal or bi-yearly combinations

     combo_yr=ifelse(x>365*int_day,floor(364*int_day/(wk*7)),ceiling(x/(7*wk)))#how many combinations in a year (using 364
days/yr for calculations)
     intmean=numeric(combo_yr)
     intsd=numeric(combo_yr)
     #must use 364 to keep whole weeks, starting and ending on the same day of the week!  The specific month or paticular "number
date" are overlooked
     #each year the combination will continue leaving an additional day to the next year... the following year will leave 2 days to the
next! and so on...
     groups=numeric()
     x_combo=floor(x/int_day)
     x=x_combo*int_day#trims the energy usage data
     if(length(zeros_new)>0){if (tail(zeros_new,n=1)>x){zeros_new=zeros_new[1:(min(which(zeros_new>x))-1)]}}#trim
"zeros_new" as well, if there are zeros in the trim, zscore fails
     n=ceiling(x/(364*int_day))
     run_new=run_new[1:(x)]

     for (i in 1:combo_yr) {groups[i]=list(sapply(1:n,function(y){run_new[int_day*sequence(7*wk)-int_day+i+((y-
1)*int_day*364)]}))}#creates a pattern for the combination to create groups of data
     for (i in 1:combo_yr) {intsd[i]=sd(unlist(groups[i])[(which(unlist(groups[i])>=0))])}
     for (i in 1:combo_yr) {intmean[i]=mean(unlist(groups[i])[(which(unlist(groups[i])>=0))])}#for a given data interval (q), a mean
and SD are created by combining using "p"th weeks

     if(length(zeros_new)!=0){ #alot of no 0s indicated
      for (i in 1:length(zeros_new)) {k=zeros_new[i] #p=4&5 for run_new could be shorter....
```

```
        j=ifelse(k%%combo_yr==0,combo_yr,k%%combo_yr)#if the remainer is 0, i want the "combo_yr"th index not "0"th index
        zscore[i,p]=(run_new[k]-intmean[j])/intsd[j]  #use the "new_run" that contains a 0, find its z score away from the mean.
        chk[i,p]=ifelse(run_new[k]==0,1,0)#if run_new=0 then it is already a known outage by simply counting the 0s at any interval
        }#finds the standardized distance way from the mean (defined as zscores) of all the outages
        store_x_EAFB[p,w]=x
        typeIIerror_EAFB[p,w]=length(which(zscore[,p]>test_crit_val[p,q,10,2]))-
sum(chk[which(zscore[,p]>test_crit_val[p,q,10,2]),p]) #outputs the number of zeros that are larger than the "critical value", because
"zscore" only contains outages,
        #those indicated are not ID'd as outages. also if there are any zscores that do not make the cut BUT have a "run_new" of 0 then
the outage can be ID'd by simply searching for 0s.
        intsdopr=rep(intsd,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
        intmeanopr=rep(intmean,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
        typeIerror_EAFB[p,w]=ifelse(length(setdiff(which(((run_new-
intmeanopr)/intsdopr)<=test_crit_val[p,q,10,2]),zeros_new))==0,0,length(setdiff(which(((run_new-
intmeanopr)/intsdopr)<=test_crit_val[p,q,10,2]),zeros_new)))
        neg_neg_EAFB[p,w]=length(which(is.na(zscore[,p])!=1))-typeIIerror_EAFB[p,w]#any outage with a zscore equal to or less
than the critical value is correctly ID'd as an outage (true reject the null)
        pos_pos_EAFB[p,w]=x-neg_neg_EAFB[p,w]-typeIIerror_EAFB[p,w]-typeIerror_EAFB[p,w]#all other values are correctly
ID'd no outage
        #typeIerror finds the percentage of non-zero containing usages that are incorrectly highlighted using crit_val (when locating
zeros)
      }#IF everything below crit_val was assumed to be an outage... what percentage of highlighted data from the entire base would be
an error?
      #THAT is good if there is a 0... but there probably isnt with daily energy usage... I just want to know how many outages the
model thinks happend...
      {intsdopr=rep(intsd,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
        intmeanopr=rep(intmean,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
        outages_EAFB[p,w]=length(which(((run_new-intmeanopr)/intsdopr)<=test_crit_val[p,q,10,2]))}#outputs number of outages
according to the numbers found in the Pecan street model building

     }#end of P loop
   }#end of STD DEV VS. OUTAGE
  }#end of W loop
  {out_num_EAFB=numeric(numdataid)
   out_median_EAFB=numeric(numdataid)
   out_mean_EAFB=numeric(numdataid)
   for (i in 1:numdataid)out_num_EAFB[i]=length(outlengstr_EAFB[i,which(outlengstr_EAFB[i,]>0)])
   for (i in 1:numdataid)out_median_EAFB[i]=median(outlengstr_EAFB[i,which(outlengstr_EAFB[i,]>0)])
   for (i in 1:numdataid)out_mean_EAFB[i]=mean(outlengstr_EAFB[i,which(outlengstr_EAFB[i,]>0)])
   out_mean_total_EAFB=mean(outlengstr_EAFB[which(outlengstr_EAFB>0)])
   out_median_total_EAFB=median(outlengstr_EAFB[which(outlengstr_EAFB>0)])
  }
  time1=Sys.time()-start_time
  time1}#end of model validiation
####Graph Code####  PRINT ALL AT 1000 pixels wide!!!!!!!
#plot(x, y, main="title", sub="subtitle", xlab="X-axis label", ylab="y-axix label", xlim=c(xmin, xmax), ylim=c(ymin, ymax))#
Specify axis options within plot()

  {chk=c("1 Week","2 Weeks","4 Weeks","13 Weeks","26 Weeks","52 Weeks")
  chk1=numeric(6)
  for(i in 1:6)
{chk1[i]=100*sum(typeIerror_EAFB[i,],na.rm=TRUE)/(sum(typeIerror_EAFB[i,],na.rm=TRUE)+sum(neg_neg_EAFB[i,],na.rm=TR
UE))}
  barplot(chk1,col=c("red"), ylim=c(0,100),space=0, axes=TRUE, main="EAFB Type I Errors as % of All Identified Outages",
        xlab="Combinations", ylab="Percent Errors (%)", legend=c("Type I Error"))
  title(" \"X% of the identified outages were incorrectly specified as an outage\" ",line = .5,col.main="dark blue",cex.main=.9)
  grid()
  axis(1,at=(1:6)-.5,labels=chk[1:6])
  }#shows where and how big the errors are by percentage for all combinations

####NOT USED (ITS ALL ZERO!!!)
  {chk=c("1 Week","2 Weeks","4 Weeks","13 Weeks","26 Weeks","52 Weeks")
   chk1=numeric(6)
```

```
    for(i in 1:6)
{chk1[i]=100*sum(typeIIerror_EAFB[i,],na.rm=TRUE)/(sum(typeIIerror_EAFB[i,],na.rm=TRUE)+sum(neg_neg_EAFB[i,],na.rm=T
RUE))}
    barplot(chk1,col=c("red"), ylim=c(0,100),space=0, axes=TRUE, main="EAFB Type II Errors as % of All known Outages",
          xlab="Combinations", ylab="Percent Errors (%)", legend=c("Type II Error"))
    title(" \"X% of actual outages that were not identified\" ",line = .5,col.main="dark blue",cex.main=.9)
    grid()
    axis(1,at=(1:6)-.5,labels=chk[1:6])
  }#THERE is NO Type II ERRORS!!!

{chk=c("1 Week","2 Weeks","4 Weeks","13 Weeks","26 Weeks","52 Weeks")
  chk1=numeric(6)
  for(i in 1:6)
{chk1[i]=100*(sum(neg_neg_EAFB[i,],na.rm=TRUE)+sum(pos_pos_EAFB[i,],na.rm=TRUE))/(sum(store_x_EAFB[i,],na.rm=TRUE
))}
  barplot(chk1,col=c("red"), ylim=c(0,100),space=0, axes=TRUE, main="EAFB \"Good Match\" as % of Total Sample Space",
        xlab="Combinations", ylab="Percent Correct (%)")
  title(" \"X% of the total sample space that contains no errors (\"Good Match\") ",line = .5,col.main="dark blue",cex.main=.9)
  grid()
  axis(1,at=(1:6)-.5,labels=chk[1:6])
}#shows how good at ID'ing the outages as a whole by compairing Positive ID with total ID'd

##########bar charts
{
  {chk=sort(outloc_EAFB[which(is.na(outloc_EAFB)!=1)])
  class(chk)=c('POSIXt','POSIXct')
  setdiff(1:2951,((chk1/(60*60*24))-14367))#which days didnt have an outage...93 days between May 2009 and May 2017
  chk1=sort(outloc_EAFB[which(is.na(outloc_EAFB)!=1)])
  barplot(table((chk1/(60*60*24))-14367),xlim = c(1,2951),space = 0,main=paste("Outages in a Day"),ylab="Number of
Outages",xlab="Date",axes =TRUE,xaxt='n', ann=FALSE)
  axis(1,at=(1:8)*365-200,labels=c("Jan 10","Jan 11","Jan 12","Jan 13","Jan 14","Jan 15","Jan 16","Jan 17"))
  }#code for day of outage #because its a day value...daylights savings time affects the output

   {chk=order(-
(colSums(typeIIerror_EAFB[,],na.rm=TRUE)+colSums(typeIerror_EAFB[,],na.rm=TRUE))/colSums(store_x_EAFB[,],na.rm=TRUE
))
    barplot(out_mean_EAFB[chk], col=c("red"), ylim=c(0,25),space=0, axes=TRUE, main="Mean Outage Length in Order of Highest
Errors",
          xlab="Data ID index", ylab="Mean (days)", legend=c("Mean Outage Length"),args.legend=list(x=35, y=25,bty = "y"))
    grid()
    axis(1,at=((1:39)*2)-1.5,labels=chk[((1:39)*2)-1],las=2)
    axis(1,at=(1:39)*2-.5,labels=paste(chk[(1:39)*2],"     "),las=2,tck=-.05)}#shows the mean outage length in the order found using
total errors by percentage

  {chk1=numeric(numdataid)
    for(i in 1:numdataid)
chk1[i]=(sum(typeIIerror_EAFB[,i],na.rm=TRUE)+sum(typeIerror_EAFB[,i],na.rm=TRUE))/sum(store_x_EAFB[,i],na.rm=TRUE)
    chk=order(-chk1)
    chk1=chk1[chk]
    barplot(100*chk1, col=c("red"),ylim=c(0,10),space=0, axes=TRUE, main=paste("Pareto Chart of Total Errors by DataID as % of
Total Values"),
          xlab="Data ID index", ylab="Percent (%)", legend=c("Total (Type I & II) Error"))
    grid()
    axis(1,at=((1:39)*2)-1.5,labels=chk[((1:39)*2)-1],las=2)
    axis(1,at=(1:39)*2-.5,labels=paste(chk[(1:39)*2],"     "),las=2,tck=-.05)}
}
```

Pecan Street Ideal Critical Value Sensitivity

```
###############MODEL BUILD###############
#library(readr)
#rawdata <- read_csv("pecan/rawdata.csv", col_names = FALSE)
colnames(rawdata)=(c("X1","X2","X3"))
```

```
dataidlist=unlist(unique(rawdata[,3]))
numdataid=length(dataidlist) #number of distinct data IDs=88
possible_int_leng_min=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,72
0,1440)
possible_combo_int=c(1,2,4,13,26,52)
day=array(dim=c(numdataid,2))
duplicates=numeric(numdataid)
minperyear=60*24*365
percent=c(.05,.15,.25,.35,.45,.55,.65,.75,.85,.95)
percent1=c(.05,.15,.25,.35,.45,.55,.65,.75,.85,.95)
crit_val=array(dim=c(6,36,numdataid,length(percent)))
test_crit_val=array(dim=c(6,36,length(percent),length(percent1)))
v=1#####TESTS
w=1#####TESTS
{start_time=Sys.time()
 for(w in 1:numdataid){#for each DataID
   run=subset(rawdata,rawdata$X3==as.numeric(dataidlist[w])) #run is created for the individual DataID
   run$X2[which(run$X2<0)]=0#clean negatives out and use 0. assumption is that negative usage is due to backfeeding of power in an
outage or a meter malfunction
   if(sum(duplicated(run$X1))>0){duplicates[w]=1#notifies if there is any duplicates in the specific DataID. hypothetically this is a
malfunction and would need to be fixed
   run=run[!duplicated(run$X1),]}#if there was a duplicate, the 1st value is assumped to be correct and everything else is an error.
   run=run[!is.na(run$X1),]#removes the rows that do not have a valid date
   run=run[order(run$X1),]#errors in the dataset save out of cronological order, this puts them in assending order
   xleng=length(run$X1)

   {i=1
    j=1
    while (i<2){chk=as.numeric(run[j,1])
    if (chk%%(60*60*24)==60){
      if (is.na(run[j,2])|run[j,2]==0) {j=j+60*24} #if the first full day starts with a 0 or NA, move to the next day because the begining
of data set is not clean
      else{i=i+1}}
    else {j=j+1}}#this function finds the first position in the data to being clean_date
    k=j
    clean_date=as.numeric(unlist(run[j:xleng,1]))
   }#this function creates dates as intergers and begins with the first full day of data that is not an NA or 0

   xleng=length(clean_date)#new length with trimmed times
   totalmin=1+(clean_date[xleng]-clean_date[1])/60
   day[w,1]=as.Date(clean_date[1], origin="1970-01-01")
   day[w,2]=as.Date(clean_date[xleng], origin="1970-01-01")

   {i=1
    j=1
    clean=numeric(totalmin)
    while (i<totalmin+1){
      if(clean_date[1]+60*(i-1)-clean_date[j]==0){
        if(is.na(run[k,2])){clean[i]=0}
        else{clean[i]=(run[k,2])}
        k=k+1
        i=i+1
        j=j+1}
      else{clean[i]=0
      i=i+1}}
   }#fills gaps in data with 0 energy usage
   xleng=length(clean)#new length with filled in missing data
   clean=unlist(clean)

   {i=1#where are the 0s?
    j=1
    k=1
    zeros=sort(which(clean==0)) #tells me which rows are 0!
    if(length(zeros)>1){
```

```
    for (i in 1:(length(zeros)-1)){
      if (zeros[i]-zeros[i+1]==-1){j=j+1}
      else{outlengstr[w,k]=j
      k=k+1
      j=1}
      outlengstr[w,k]=j}}#last one!
  else{if(length(zeros==1)){outlengstr[w,1]=1}
    else{outlengstr[w,1]=0}}
}#finds outage lengths and and frequency

q=36####TEST
for (q in 1:36){  #for each interval (36 different intervals)
  int_leng_min=possible_int_leng_min[q]#tested interval in "q-th" minutes
  int_day=(60*24)/int_leng_min #how many intervals in a single day
  x=floor(xleng/int_leng_min) #combination rule....new length rounded down
  zeros_new=unique(ceiling(zeros/int_leng_min))
  zeros_new1=zeros_new

  {run_new=(sapply(1:x,function(i){
    o=(sapply(1:int_leng_min,function(j){
      (clean[int_leng_min*i-int_leng_min+j])}))
    sum(o[1:int_leng_min])
  }))
  }#combines usage data by the given interval (total energy usage) !!!!NO LONGER MIN/MIN!!!!!

  run_new1=run_new
  p=1####TEST
  if(length(zeros_new)!=0){
    run_new=run_new1
    {ddays=(as.numeric(clean_date[1])-1302840000)/(60*60*24)
      j=((91-floor(ddays%%(365/4))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule of
15 April 2011 every 3 months...each year add an additional day...each leap year add an additional day.
      chk1=run_new[(j):x]
      if(length(zeros)!=1 &
zeros[length(zeros)]/int_leng_min>j){zchk1=unique(ceiling(zeros[(1+length(which(zeros<j*int_leng_min))):length(zeros)]/int_leng_
min))-j+1}
      else{if ((zeros[1]/int_leng_min)<j){zchk1=numeric()}
        else{zchk1=unique(ceiling(zeros[1]/int_leng_min))-j+1}}#in case there is only one "0" and that zero is below the new start
    }#starts data within a day of the the middle of Jan, Apr, Jul, Oct

    {ddays=(as.numeric(clean_date[1])-1302840000)/(60*60*24)
      j=((182-floor(ddays%%(365/2))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule
of 15 April 2011 every 6 months...each year add an additional day...each leap year add an additional day.
      chk2=run_new[(j):x]
      if(length(zeros)!=1 &
zeros[length(zeros)]/int_leng_min>j){zchk2=unique(ceiling(zeros[(1+length(which(zeros<j*int_leng_min))):length(zeros)]/int_leng_
min))-j+1}
      else{if ((zeros[1]/int_leng_min)<j){zchk2=numeric()}
        else{zchk2=unique(ceiling(zeros[1]/int_leng_min))-j+1}}#in case there is only one "0" and that zero is below the new start
      #cuts out any zeros that occur before j and re-indexs the values
    }#important for SINE wave of outdoor temp to minimize variance in energy useage. place summer and winter in different
categories
    zscore=array(dim=c(length(zeros_new),6))
    chk=array(dim=c(length(zeros_new),6))
    for(p in 1:6){
      wk=possible_combo_int[p] #combine the data for "p" weeks

      {if(wk==13){run_new=chk1
      x=length(run_new)
      zeros_new=zchk1}
        if(wk==26){run_new=chk2
        x=length(run_new)
        zeros_new=zchk2}
        if(wk==52){run_new=run_new1
```

```
          x=length(run_new1)
          zeros_new=zeros_new1}}#alters the data if looking at seasonal or bi-yearly combinations

        combo_yr=ifelse(x>365*int_day,floor(364*int_day/(wk*7)),ceiling(x/(7*wk)))#how many combinations in a year (using 364
days/yr for calculations)
        intmean=numeric(combo_yr)
        intsd=numeric(combo_yr)
        #must use 364 to keep whole weeks, starting and ending on the same day of the week!  The specific month or paticular "number
date" are overlooked
        #each year the combination will continue leaving an additional day to the next year... the following year will leave 2 days to the
next! and so on...
        groups=numeric()
        x_combo=floor(x/int_day)
        x=x_combo*int_day
        if(length(zeros_new)>0){if (tail(zeros_new,n=1)>x){zeros_new=zeros_new[1:(min(which(zeros_new>x))-1)]}}#trim
"zeros_new" as well, if there are zeros in the trim, zscore fails
        n=ceiling(x/(364*int_day))
        run_new=run_new[1:(x)]

        for(i in 1:combo_yr) {groups[i]=list(sapply(1:n,function(y){run_new[int_day*sequence(7*wk)-int_day+i+((y-
1)*int_day*364)]}))}#creates a pattern for the combination to create groups of data
        for (i in 1:combo_yr) {intsd[i]=sd(unlist(groups[i])[(which(unlist(groups[i])>=0))])}
        for (i in 1:combo_yr) {intmean[i]=mean(unlist(groups[i])[(which(unlist(groups[i])>=0))])}#for a given data interval (q), a mean
and SD are created by combining using "p"th weeks

        if(length(zeros_new)!=0){
          for (i in 1:length(zeros_new)) {k=zeros_new[i] #p=4&5 for run_new could be shorter....
          j=ifelse(k%%combo_yr==0,combo_yr,k%%combo_yr)#if the remainer is 0, i want the "combo_yr"th index not "0"th index
          zscore[i,p]=(run_new[k]-intmean[j])/intsd[j]  #use the "new_run" that contains a 0, find its z score away from the mean.
          chk[i,p]=ifelse(run_new[k]==0,1,0)#if run_new=0 then it is already a known outage by simply counting the 0s at any interval
          }#finds the standardized distance way from the mean (defined as zscores) of all the outages
          for(v in 1:length(percent)){
            crit_val[p,q,w,v]=zscore[order(-
zscore[which(is.na(zscore[,p])!=1),p]),p][ceiling(percent[v]*length(which(is.na(zscore[,p])!=1)))]##90% (or if <10 zeros... ((z-1)/z)%
(less than 90%)) of the zeros are at or below this zscore
            crit_val[p,q,w,v]=ifelse(crit_val[p,q,w,v]>0,0,crit_val[p,q,w,v])
            #scores are ordered high to low (because of the "-zscore") and then the value is chosen (95% is chosen 1st)
          }#end of v loop
        }#IF everything below crit_val was assumed to be an outage... what percentage of highlighted data from the entire base would
be an error?
      }#end of P loop
    }#end of STD DEV VS. OUTAGE
  }}#end of W & Q loop
 {for (i in 1:6){for (j in 1:36){for (v in 1:length(percent)){for (f in length(percent1):1){
   test_crit_val[i,j,v,11-f]=sort(crit_val[i,j,,v])[ceiling(numdataid*percent1[f])]
#test_crit_val[combonaiton,interval,percentile_crit,percentile_idealcrit]
   #low to high then chooses a number...in reverse (95% is chosen 1st)
 }}}}
 }#crit_val key metrics and error information

 #{write.table(day, file = "C:/Users/Jared/Google Drive/day.txt", sep = "\t", row.names = FALSE, col.names = FALSE)
 #write.table(test_crit_val, file = "C:/Users/Jared/Google Drive/test_crit_val.txt", sep = "\t", row.names = FALSE, col.names =
FALSE)
 #write.table(day, file = "pecan/day.txt", sep = "\t", row.names = FALSE, col.names = FALSE)#umbunto code
 #write.table(test_crit_val, file = "pecan/test_crit_val.txt", sep = "\t", row.names = FALSE, col.names = FALSE)#umbunto code
 #}#print the tables that will be needed in validation and aggregation processes

 time1=Sys.time()-start_time
 time1}#end of model building code


################VALIDATE###############
#requires "test_crit_val" from Model_Building.R
```

```
#library(readr)
#testdata <- read_csv("pecan/testdata.txt", col_names = FALSE)#import the test data... format= '%Y-%m-%d %H:%M:%S-%z'
#colnames(testdata)=(c("X1","X2","X3"))
{dataidlist_test=unlist(unique(testdata[,3]))
numdataid=length(dataidlist_test)
possible_int_leng_min=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,72
0,1440)
possible_combo_int=c(1,2,4,13,26,52)
percent=c(.05,.15,.25,.35,.45,.55,.65,.75,.85,.95)
store_x_test=array(dim=c(6,36,numdataid))
pos_pos_test=array(dim=c(6,36,numdataid,length(percent),length(percent1)))
neg_neg_test=array(dim=c(6,36,numdataid,length(percent),length(percent1)))
typeIIerror_test=array(dim=c(6,36,numdataid,length(percent),length(percent1)))
typeIerror_test=array(dim=c(6,36,numdataid,length(percent),length(percent1)))
outlengstr_test=array(dim=c(numdataid,1000000))
day_test=array(dim=c(numdataid,2))
duplicates_test=numeric(numdataid)
minperyear=60*24*365}#creates the variables
w=1#####TESTS
{start_time=Sys.time()
  for(w in 1:numdataid){#for each DataID
    run=subset(testdata,testdata$X3==as.numeric(dataidlist_test[w])) #run is created for the individual DataID
    run$X2[which(run$X2<0)]=0#clean negatives out and use 0. assumption is that negative usage is due to backfeeding of power in an
outage or a meter malfunction
    if(sum(duplicated(as.numeric(run$X1)))>0){duplicates_test[w]=1#notifies if there is any duplicates in the specific DataID.
hypothetically this is a malfunction and would need to be fixed
    run=run[!duplicated(run$X1),]}#if there was a duplicate the 1st value is assumed to be correct and everything else is an error.
    run=run[!is.na(run$X1),]#removes the rows that do not have a valid date
    run=run[order(run$X1),]#errors in the dataset save out of cronological order, this puts them in assending order
    xleng=length(run$X1)

    {i=1
     j=1
     while (i<2){chk=as.numeric(run[j,1])
     if (chk%%(60*60*24)==60){
       if (is.na(run[j,2])|run[j,2]==0) {j=j+60*24} #if the first full day starts with a 0 or NA, move to the next day because the begining
of data set is not clean
       else{i=i+1}}
     else {j=j+1}}#this function finds the first position in the data to being clean_date
     k=j
     clean_date=as.numeric(unlist(run[j:xleng,1]))
    }#this function creates dates as intergers and begins with the first full day of data that is not an NA or 0

    xleng=length(clean_date)#new length with trimmed times
    totalmin=1+(clean_date[xleng]-clean_date[1])/60
    day_test[w,1]=as.Date(clean_date[1], origin="1970-01-01")
    day_test[w,2]=as.Date(clean_date[xleng], origin="1970-01-01")

    {i=1
     j=1
     clean=numeric(totalmin)
     while (i<totalmin+1){
       if(clean_date[1]+60*(i-1)-clean_date[j]==0){
         if(is.na(run[k,2])){clean[i]=0}
         else{clean[i]=(run[k,2])}
         k=k+1
         i=i+1
         j=j+1}
       else{clean[i]=0
       i=i+1}}
    }#fills gaps in data with 0 energy usage
    xleng=length(clean)#new length with filled in missing data
    clean=unlist(clean)
```

```
{i=1#where are the 0s?
 j=1
 k=1
 zeros=sort(which(clean==0)) #tells me which rows are 0!
 if(length(zeros)>1){
   for (i in 1:(length(zeros)-1)){
     if (zeros[i]-zeros[i+1]==-1){j=j+1}
     else{outlengstr_test[w,k]=j
     k=k+1
     j=1}
     outlengstr_test[w,k]=j}}#last one!
 else{if(length(zeros==1)){outlengstr_test[w,1]=1}
   else{outlengstr_test[w,1]=0}}
}#finds outage lengths and and frequency

q=1####TEST
for (q in 1:36){  #for each interval (36 different intervals)
 int_leng_min=possible_int_leng_min[q]#tested interval in "q-th" minutes
 int_day=(60*24)/int_leng_min #how many intervals in a single day
 x=floor(xleng/int_leng_min) #combination rule....new length rounded down
 zeros_new=unique(ceiling(zeros/int_leng_min))
 zeros_new1=zeros_new

 {run_new=(sapply(1:x,function(i){
   o=(sapply(1:int_leng_min,function(j){
     (clean[int_leng_min*i-int_leng_min+j])}))
   sum(o[1:int_leng_min])
 }))
 }#combines usage data by the given interval (total energy usage) !!!!NO LONGER MIN/MIN!!!!!

 run_new1=run_new
 p=1####TEST
 if(length(zeros_new)!=0){
   run_new=run_new1
   {ddays=(as.numeric(clean_date[1])-1302840000)/(60*60*24)
     j=((91-floor(ddays%%(365/4))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule of
15 April 2011 every 3 months...each year add an additional day...each leap year add an additional day.
     chk1=run_new[(j):x]
     if(length(zeros)!=1 &
zeros[length(zeros)]/int_leng_min>j){zchk1=unique(ceiling(zeros[(1+length(which(zeros<j*int_leng_min))):length(zeros)]/int_leng_
min))-j+1}
     else{if ((zeros[1]/int_leng_min)<j){zchk1=numeric()}
       else{zchk1=unique(ceiling(zeros[1]/int_leng_min))-j+1}}#in case there is only one "0" and that zero is below the new start
   }#starts data within a day of the the middle of Jan, Apr, Jul, Oct

   {ddays=(as.numeric(clean_date[1])-1302840000)/(60*60*24)
     j=((182-floor(ddays%%(365/2))+floor(ddays/(365))+ifelse(ddays<365,0,floor((365+ddays)/(365*4))))*int_day)+1 #multipule
of 15 April 2011 every 6 months...each year add an additional day...each leap year add an additional day.
     chk2=run_new[(j):x]
     if(length(zeros)!=1 &
zeros[length(zeros)]/int_leng_min>j){zchk2=unique(ceiling(zeros[(1+length(which(zeros<j*int_leng_min))):length(zeros)]/int_leng_
min))-j+1}
     else{if ((zeros[1]/int_leng_min)<j){zchk2=numeric()}
       else{zchk2=unique(ceiling(zeros[1]/int_leng_min))-j+1}}#in case there is only one "0" and that zero is below the new start
     #cuts out any zeros that occur before j and re-indexs the values
   }#important for SINE wave of outdoor temp to minimize variance in energy useage. place summer and winter in different
categories
   zscore=array(dim=c(length(zeros_new),6))
   chk=array(dim=c(length(zeros_new),6))
   for(p in 1:6){
     wk=possible_combo_int[p] #combine the data for "p" weeks

     {if(wk==13){run_new=chk1
     x=length(run_new)
```

```
        zeros_new=zchk1}
         if(wk==26){run_new=chk2
         x=length(run_new)
         zeros_new=zchk2}
         if(wk==52){run_new=run_new1
         x=length(run_new1)
         zeros_new=zeros_new1}}#alters the data if looking at seasonal or bi-yearly combinations

        combo_yr=ifelse(x>365*int_day,floor(364*int_day/(wk*7)),ceiling(x/(7*wk)))#how many combinations in a year (using 364
days/yr for calculations)
        intmean=numeric(combo_yr)
        intsd=numeric(combo_yr)
        #must use 364 to keep whole weeks, starting and ending on the same day of the week!  The specific month or paticular "number
date" are overlooked
        #each year the combination will continue leaving an additional day to the next year... the following year will leave 2 days to the
next! and so on...
        groups=numeric()
        x_combo=floor(x/int_day)
        x=x_combo*int_day
        if(length(zeros_new)>0){if (tail(zeros_new,n=1)>x){zeros_new=zeros_new[1:(min(which(zeros_new>x))-1)]}}#trim
"zeros_new" as well, if there are zeros in the trim, zscore fails
        n=ceiling(x/(364*int_day))
        run_new=run_new[1:(x)]

        for(i in 1:combo_yr) {groups[i]=list(sapply(1:n,function(y){run_new[int_day*sequence(7*wk)-int_day+i+((y-
1)*int_day*364)]}))}#creates a pattern for the combination to create groups of data
        for (i in 1:combo_yr) {intsd[i]=sd(unlist(groups[i])[(which(unlist(groups[i])>=0))])}
        for (i in 1:combo_yr) {intmean[i]=mean(unlist(groups[i])[(which(unlist(groups[i])>=0))])}#for a given data interval (q), a mean
and SD are created by combining using "p"th weeks

        if(length(zeros_new)!=0){
         for (i in 1:length(zeros_new)) {k=zeros_new[i] #p=4&5 for run_new could be shorter....
         j=ifelse(k%%combo_yr==0,combo_yr,k%%combo_yr)#if the remainer is 0, i want the "combo_yr"th index not "0"th index
         zscore[i,p]=(run_new[k]-intmean[j])/intsd[j]  #use the "new_run" that contains a 0, find its z score away from the mean.
         chk[i,p]=ifelse(run_new[k]==0,1,0)#if run_new=0 then it is already a known outage by simply counting the 0s at any interval
         }#finds the standardized distance way from the mean (defined as zscores) of all the outages
         store_x_test[p,q,w]=x
         intsdopr=rep(intsd,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
         intmeanopr=rep(intmean,ceiling(x/combo_yr))[1:x]#operator to make calculations quicker
         for(v in 1:length(percent)){for(f in 1:length(percent1)){
          typeIIerror_test[p,q,w,v,f]=length(which(zscore[,p]>test_crit_val[p,q,v,f]))-
sum(chk[which(zscore[,p]>test_crit_val[p,q,v,f]),p]) #outputs the number of zeros that are larger than the "critical value", because
"zscore" only contains outages,
          #those indicated are not ID'd as outages. also if there are any zscores that do not make the cut BUT have a "run_new" of 0
then the outage can be ID'd by simply searching for 0s.
          typeIerror_test[p,q,w,v,f]=ifelse(length(setdiff(which(((run_new-
intmeanopr)/intsdopr)<=test_crit_val[p,q,v,f]),zeros_new))==0,0,length(setdiff(which(((run_new-
intmeanopr)/intsdopr)<=test_crit_val[p,q,v,f]),zeros_new)))
          neg_neg_test[p,q,w,v,f]=length(which(is.na(zscore[,p])!=1))-typeIIerror_test[p,q,w,v,f]#any outage with a zscore equal to or
less than the critical value is correctly ID'd as an outage (true reject the null)
          pos_pos_test[p,q,w,v,f]=x-neg_neg_test[p,q,w,v,f]-typeIIerror_test[p,q,w,v,f]-typeIerror_test[p,q,w,v,f]#all other values are
correctly ID'd no outage
         }}#end of v & F loop

         #typeIerror finds the percentage of non-zero containing usages that are incorrectly highlighted using crit_val (when locating
zeros)
        }#IF everything below crit_val was assumed to be an outage... what percentage of highlighted data from the entire base would
be an error?
      }#end of P loop
     }#end of STD DEV VS. OUTAGE
    }}#end of W & Q loop
  {out_num_test=numeric(numdataid)
   out_median_test=numeric(numdataid)
   out_mean_test=numeric(numdataid)
```

```
    for (i in 1:numdataid)out_num_test[i]=length(outlengstr_test[i,which(outlengstr_test[i,]>0)])
    for (i in 1:numdataid)out_median_test[i]=median(outlengstr_test[i,which(outlengstr_test[i,]>0)])
    for (i in 1:numdataid)out_mean_test[i]=mean(outlengstr_test[i,which(outlengstr_test[i,]>0)])
    out_mean_total_test=mean(outlengstr_test[which(outlengstr_test>0)])
    out_median_total_test=median(outlengstr_test[which(outlengstr_test>0)])
  }
  time1=Sys.time()-start_time
  time1}#end of model validiation
####Graph Code####  PRINT ALL AT 1000 pixels wide!!!!!!!
#plot(x, y, main="title", sub="subtitle", xlab="X-axis label", ylab="y-axix label", xlim=c(xmin, xmax), ylim=c(ymin, ymax))#
Specify axis options within plot()
{
  setwd("C:/Users/Jared/Google Drive/01-Erickson - Thesis/1-Working Docs/Data/Pecan/Outputs")
  #####THESE ARE ONLY FOR ONE PART OF THE ANALYSIS.....
  #{j=10
  par(mfrow=c(1,1),xpd=T,mar=c(5,4,4,9))
    for(j in 1:1){
    chk1=c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 Weeks/Annually")
    chk2=c("95th Percentile","85th Percentile","75th Percentile","65th Percentile","55th Percentile","45th Percentile","35th
Percentile","25th Percentile","15th Percentile","5th Percentile")
    chk3=c("blue","red","green","black","orange","yellow")
    for(k in 1:1){
    #{k=2
     #jpeg(filename = paste("crit_",k,"of",j,".jpg"),width = 7.5, height = 7.5, units ="in",res = 750)

      plot(c(1,36),c(-7,-1),type="n",main=paste("Ideal Critical Values \nFound Using the",chk2[k],"of the",chk2[j]), xlab="Interval
(minutes)", ylab="Critical Value",xaxt="n")
      axis(side=1, at=1:36,
labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
      grid(NULL,NULL, lwd = 2)
      for(i in 1:6){
        lines((1:(36)),test_crit_val[i,,j,k], col=chk3[i], lty=1, lwd=4, pch=19)
        points((1:(36)),test_crit_val[i,,j,k], col=chk3[i], pch=19, cex=1)}
      legend(38,-3,chk1,cex=.7,y.intersp=1,title=("SD/Mean Combinations"),
             lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=chk3)#imports a legend for the plot
      #dev.off()
    }
  }#shows the specified "Critical Values" for each percentile


  {
    {
    k=2
    chk3=c("95th Percentile","85th Percentile","75th Percentile","65th Percentile","55th Percentile","45th Percentile","35th
Percentile","25th Percentile","15th Percentile","5th Percentile")
    #for(j in 1:10){
    {j=10
     jpeg(filename = paste("TIE_",k,"of",j,".jpg"),width = 7.5, height = 7.5, units ="in",res = 750)
      par(mfrow=c(1,1),xpd=T,mar=c(5,4,4,9))
      plot(c(1,36),c(0,100),type="n",main=paste("Type I Errors as % of Identified Outages \nFound Using the",chk3[k],"of
the",chk3[j]), xlab="Interval (minutes)", ylab="Percent Errors (%)",xaxt="n")
      axis(side=1, at=1:36,
labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
      grid(NULL,NULL, lwd = 2)
      title(" \"X% of the identified outages were incorrectly specified as an outage\" ",line = .3,col.main="dark blue",cex.main=.9)
      chk=c("blue","red","green","black","orange","yellow")
      for (i in
1:6){lines((1:(36)),(100*(rowSums(typeIerror_test[i,,,j,k],na.rm=TRUE)/(rowSums(typeIerror_test[i,,,j,k],na.rm=TRUE)+rowSums(n
eg_neg_test[i,,,j,k],na.rm=TRUE)))), col=chk[i], lty=1, lwd=4, pch=19)

points((1:(36)),(100*(rowSums(typeIerror_test[i,,,j,k],na.rm=TRUE)/(rowSums(typeIerror_test[i,,,j,k],na.rm=TRUE)+rowSums(neg_
neg_test[i,,,j,k],na.rm=TRUE)))), col=chk[i], pch=19, cex=1)}
      legend(38,60,c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52
Weeks/Annually"),cex=.7,y.intersp=1,title=("SD/Mean Combinations"),
```

```
        lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=c("blue","red","green","black","orange","yellow"))#imports a legend for the plot
     dev.off()
    }
   }#shows how good at ID'ing the outages as a whole by compairing Positive ID with total ID'd

   {k=2
    chk3=c("95th Percentile","85th Percentile","75th Percentile","65th Percentile","55th Percentile","45th Percentile","35th
Percentile","25th Percentile","15th Percentile","5th Percentile")
    #for(j in 1:10){
    {j=10
      jpeg(filename = paste("TIIE_",k,"of",j,".jpg"),width = 7.5, height = 7.5, units ="in",res = 750)
      par(mfrow=c(1,1),xpd=T,mar=c(5,4,4,9))
      plot(c(1,36),c(0,12),type="n",main=paste("Type II Errors as % of Known Outages \nFound Using the",chk3[k],"of the",chk3[j]),
xlab="Interval (minutes)", ylab="Percent Errors (%)",xaxt="n")
      axis(side=1, at=1:36,
labels=c(1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,60,72,80,90,96,120,144,160,180,240,288,360,480,720,1440))
      grid(NULL,NULL, lwd = 2)
      title(" \"X% of actual outages that were not identified\" ",line = .3,col.main="dark blue",cex.main=.9)
      chk=c("blue","red","green","black","orange","yellow")
      for (i in
1:6){lines((1:(36)),(100*(rowSums(typeIIerror_test[i,,,j,k],na.rm=TRUE))/(rowSums(typeIIerror_test[i,,,j,k],na.rm=TRUE)+rowSums
(neg_neg_test[i,,,j,k],na.rm=TRUE))), col=chk[i], lty=1, lwd=4, pch=19)

points((1:(36)),(100*(rowSums(typeIIerror_test[i,,,j,k],na.rm=TRUE))/(rowSums(typeIIerror_test[i,,,j,k],na.rm=TRUE)+rowSums(ne
g_neg_test[i,,,j,k],na.rm=TRUE))), col=chk[i], pch=19, cex=1)}
      legend(38,7,c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52
Weeks/Annually"),cex=.7,y.intersp=1,title=("SD/Mean Combinations"),
           lty=c(1,1,1,1,1,1),lwd=c(5,5,5,5,5,5),col=c("blue","red","green","black","orange","yellow"))#imports a legend for the plot
      dev.off()
     }
   }
  }#shows how good at ID'ing the outages as a whole by compairing Positive ID with total ID'd
  ####sensitivity!!###
  {
   par(xpd=T,mar=c(5, 4, 4, 2) + 0.1)
   #line graph instead of the bar graph
   {par(mfrow=c(1,1))
    j=1
    chk=character()
    chk=c("gray1","gray6","gray11","gray16","gray21","gray26","gray31","gray36","gray41","gray46")
    chk1=numeric()
    for(f in 1:10){for(v in 1:10){chk1[j]=100*(sum(typeIIerror_test[,,,v,f],na.rm = T))/(sum(neg_neg_test[,,,v,f],na.rm =
T)+sum(typeIIerror_test[,,,v,f],na.rm = T))
    j=j+1}}
    plot(c(1,10),c(0,.165),type="n",main=paste("Type II Errors as % of Known Outages for all Combinations and Intervals"),xlab =
"Ideal Critical Value", ylab="Percent Errors (%)")
    for(i in 1:10){lines(1:10,chk1[(i-1)*10+(1:10)], col=chk[i], lty=1, lwd=4, pch=19)
      points(1:10,chk1[(i-1)*10+(1:10)], col=chk[i], pch=19, cex=1)}
    title(" \"X% of actual outages that were not identified\" ",line = .3,col.main="dark blue",cex.main=.9)
    grid()
   }

   {par(mfrow=c(1,1))
    j=1
    chk1=numeric()
    for(f in 1:10){for(v in 1:10){chk1[j]=100*(sum(typeIIerror_test[,,,v,f],na.rm = T))/(sum(neg_neg_test[,,,v,f],na.rm =
T)+sum(typeIIerror_test[,,,v,f],na.rm = T))
    j=j+1}}
    barplot(chk1,col=c(rep("black",10),rep("light gray",10)),main=paste("Type II Errors as % of Known Outages for all Combinations
and Intervals"),space = 0,xlab = "Ideal Critical Value", ylab="Percent Errors (%)")
    title(" \"X% of actual outages that were not identified\" ",line = .3,col.main="dark blue",cex.main=.9)
    axis(side=1, at=c(1,100)-.5,mgp=c(0,3,0), labels=c("Largest of \nthe high percentile\n(least negative value)","smallest of \nthe
lowest percentile \n(most negative value)"))
   } #THIS SHOWS HOW SMALL A NUMBER OF TYPE II Errors there really are compared to type I
```

```
{par(mfrow=c(1,1))
 j=1
 chk1=numeric()
 for(f in 1:10){for(v in 1:10){chk1[j]=100*(sum(typeIerror_test[,,,v,f],na.rm = T))/(sum(neg_neg_test[,,,v,f],na.rm =
T)+sum(typeIerror_test[,,,v,f],na.rm = T))
 j=j+1}}
 barplot(chk1,col=c(rep("black",10),rep("light gray",10)),ylim = c(0,100),main=paste("Type I Errors as % of Identified Outages for
all Combinations and Intervals"),space = 0,xlab = "Ideal Critical Value", ylab="Percent Errors (%)")
 title(" \"X% of the identified outages were incorrectly specified as an outage\" ",line = .3,col.main="dark blue",cex.main=.9)
 axis(side=1, at=c(1,100)-.5,mgp=c(0,3,0), labels=c("Largest of \nthe high percentile\n(least negative value)","smallest of \nthe
lowest percentile \n(most negative value)"))
}

{par(mfrow=c(1,1))
 j=1
 chk1=numeric()
 chk2=numeric()
 for(f in 1:10){for(v in 1:10){chk1[j]=(sum(typeIIerror_test[,,,v,f],na.rm = T))
 chk2=chk1/max(chk1)
 j=j+1}}
 barplot(100*chk2,col=c(rep("black",10),rep("light gray",10)),main=paste("% of Max Type II Errors for all Combinations and
Intervals"),space = 0,xlab = "Ideal Critical Value", ylab="Percent of Maximum Errors (%)")
 title(" \"X% of the max total errors for all combinations and intervals\" ",line = .3,col.main="dark blue",cex.main=.9)
 axis(side=1, at=c(1,100)-.5,mgp=c(0,3,0), labels=c("Largest of \nthe high percentile\n(least negative value)","smallest of \nthe
lowest percentile \n(most negative value)"))
}
{par(mfrow=c(1,1))
 j=1
 chk1=numeric()
 chk3=numeric()
 for(f in 1:10){for(v in 1:10){chk1[j]=(sum(typeIerror_test[,,,v,f],na.rm = T))
 chk3=chk1/max(chk1)
 j=j+1}}
 barplot(100*chk3,col=c(rep("black",10),rep("light gray",10)),main=paste("% of Max Type I Errors for all Combinations and
Intervals"),space = 0,xlab = "Ideal Critical Value", ylab="Percent of Maximum Errors (%)")
 title(" \"X% of the max total errors for all combinations and intervals\" ",line = .3,col.main="dark blue",cex.main=.9)
 axis(side=1, at=c(1,100)-.5,mgp=c(0,3,0), labels=c("Largest of \nthe high percentile\n(least negative value)","smallest of \nthe
lowest percentile \n(most negative value)"))
}
{ par(xpd=T,mar=c(5, 4, 4, 2) + 0.1)

 barplot(100*matrix(c(chk3,chk2),nrow=2, byrow=TRUE),col=c("red","dark blue"),ylim=c(0,120) ,main=paste("% of Max Errors
for all Combinations and Intervals"),space = 0,legend=c("Type I Error","Type II Error"),
    args.legend=list(x=95,y=125,bty = "y",cex=.9),xlab = "Ideal Critical Value", ylab="Percent of Maximum Errors (%)")
 axis(side=1, at=c(1,100)-.5,mgp=c(0,3,0), labels=c("Largest of \nthe high percentile\n(least negative value)","Smallest of \nthe
lowest percentile \n(most negative value)"))
}
sort(chk2)[1]#1.2%   BEST FOR TYPE II ERRORS
order(chk2)[1]#1 (95% then 95% of the 95%!!) (v=1,f=1)
sort(chk3)[1]#0.1%  BEST FOR TYPE I ERRORS
order(chk3)[1]#100 (5% then 5% of the 5%!!) (v=10,f=10)

sort(chk2+chk3)[1]#73.1% BEST FOR lowest amount of TYPE I and TYPE II ERRORS
order(chk2+chk3)[1]# (65% then 85% of the 65%!!) (v=5,f=2)
chk2[20]#24% type II
chk3[20]#49% type I
} # this was before putting in a 0 block for critical value calculation
#23.0% best reduction of TIIE .18% best reduction of TIE (same locations)
#91.2% lowest combine (out of 200%; worst situation for both) 48.8% of max for TIIE...42.4% of max for TIE... position(v=10,f=2)
85% second percentile of the 5% first percentile

sum(typeIerror_test[,,,,],na.rm = T)/(sum(typeIerror_test[,,,,],na.rm = T)+sum(typeIIerror_test[,,,,],na.rm = T))#99.96% of errors are
type I errors
```

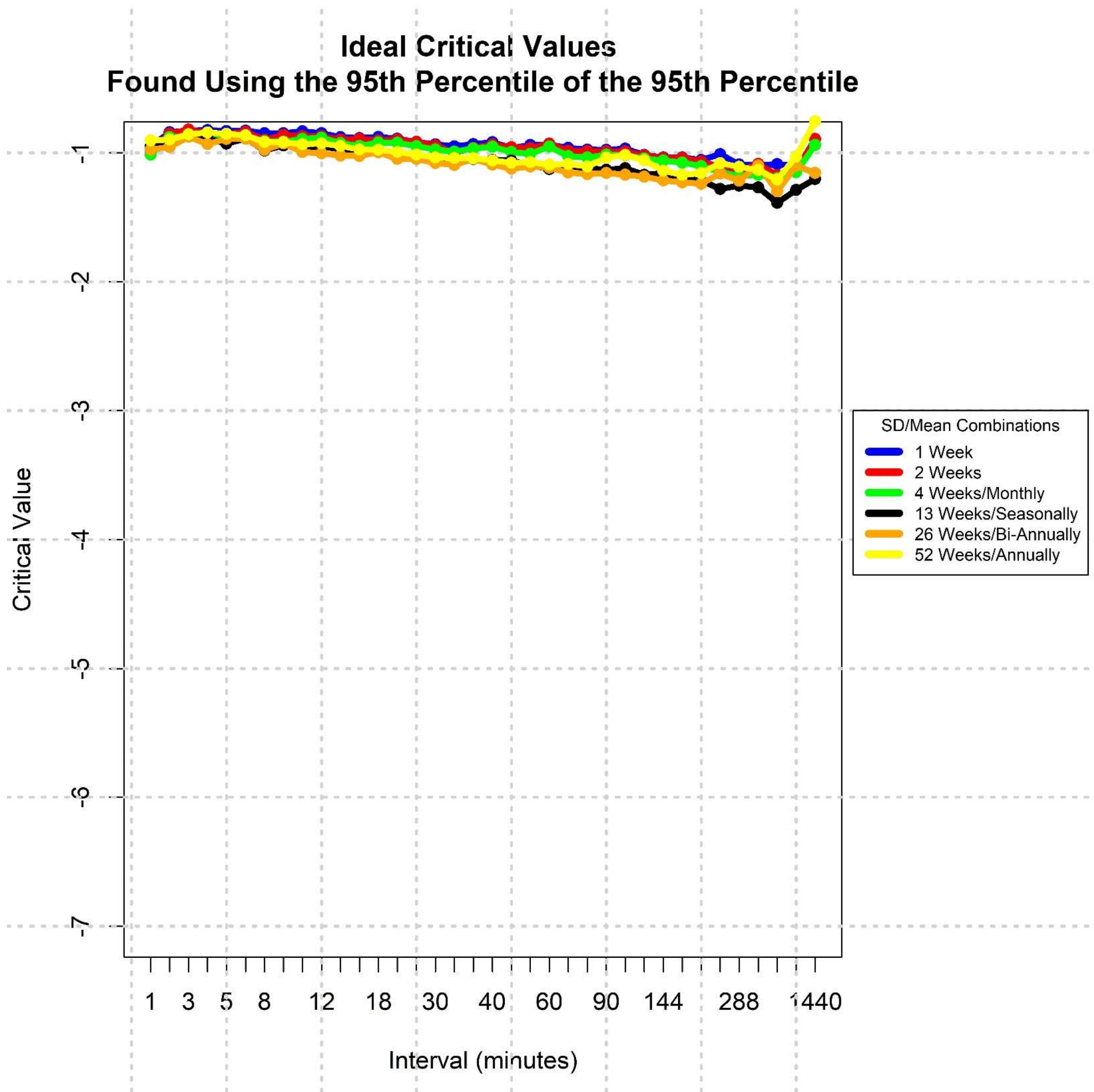
```
 v=1
 f=1
 sum(typeIerror_test[,,,v,f],na.rm = T)/(sum(typeIerror_test[,,,v,f],na.rm = T)+sum(typeIIerror_test[,,,v,f],na.rm = T))#99.9998% of
errors are type I errors
 v=10
 f=10
 sum(typeIerror_test[,,,v,f],na.rm = T)/(sum(typeIerror_test[,,,v,f],na.rm = T)+sum(typeIIerror_test[,,,v,f],na.rm = T))#88.86% of
errors are type I errors
 v=10
 f=2
 sum(typeIerror_test[,,,v,f],na.rm = T)/(sum(typeIerror_test[,,,v,f],na.rm = T)+sum(typeIIerror_test[,,,v,f],na.rm = T))#99.992% of
errors are type I errors
 {
 m=1
 Ideal_Critical_Value=numeric( )
 for(j in 1:10){ for(i in 1:10){Ideal_Critical_Value[m]=mean(test_crit_val[,,i,j])
 m=m+1}}
 m=1
 errors=numeric( )
 for(j in 1:10){ for(i in 1:10){errors[m]=sd(test_crit_val[,,i,j])
 m=m+1}}
 m=data.frame(errorm,errors)
 percentile=1:100
 library(plotly)
 plot_ly(type = 'scatter',x=percentile,y=~Ideal_Critical_Value,mode='markers',error_y=~list(value = errors,color = '#000000'))%>%
   layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(title='percentile',showticklabels = F)),yaxis=(list(title='Ideal
Critical Value')),title='Ideal Critical Value for Each Possible Percentile')

 }#all percentiles are represented

 {m=1
   errorm=numeric( )
   { for(i in 1:10){errorm[m]=mean(test_crit_val[,,i,])
   m=m+1}}
   m=1
   errors=numeric( )
   { for(i in 1:10){errors[m]=sd(test_crit_val[,,i,])
   m=m+1}}

   m=1
   errorm=numeric( )
   { for(i in 1:10){errorm[m]=mean(test_crit_val[,,,i])
   m=m+1}}
   m=1
   errors=numeric( )
   { for(i in 1:10){errors[m]=sd(test_crit_val[,,,i])
   m=m+1}}
   plot_ly(type = 'scatter',x=1:10,y=~errorm,mode='markers',error_y=~list(value = errors,color = '#000000'))
   m=data.frame(errorm,errors)}#averages over one set of percentiles

 {
   typeI=matrix(nrow = 36,ncol = 6)
   for(j in 1:36){ for(i in
1:6){typeI[j,i]=sum(typeIerror_test[i,j,,,],rm.na=T)/(sum(typeIerror_test[i,j,,,],rm.na=T)+sum(neg_neg_test[i,j,,,]))}}
   typeII=matrix(nrow = 36,ncol = 6)
   for(j in 1:36){ for(i in
1:6){typeII[j,i]=sum(typeIIerror_test[i,j,,,],rm.na=T)/(sum(typeIIerror_test[i,j,,,],rm.na=T)+sum(neg_neg_test[i,j,,,]))}}

   y=1:36
   x=c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 Weeks/Annually")
   plot_ly(x=x,y=y,z=100*typeI,type = 'heatmap',colorbar=(list(title="Percentage \nof Errors (%)",nticks=10)))%>%
     layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(title='Combination')),yaxis=(list(title='Interval
Index')),title='Heatmap of Type I Errors for All Percentiles \nby Combination and Interval')
   y=1:36
```

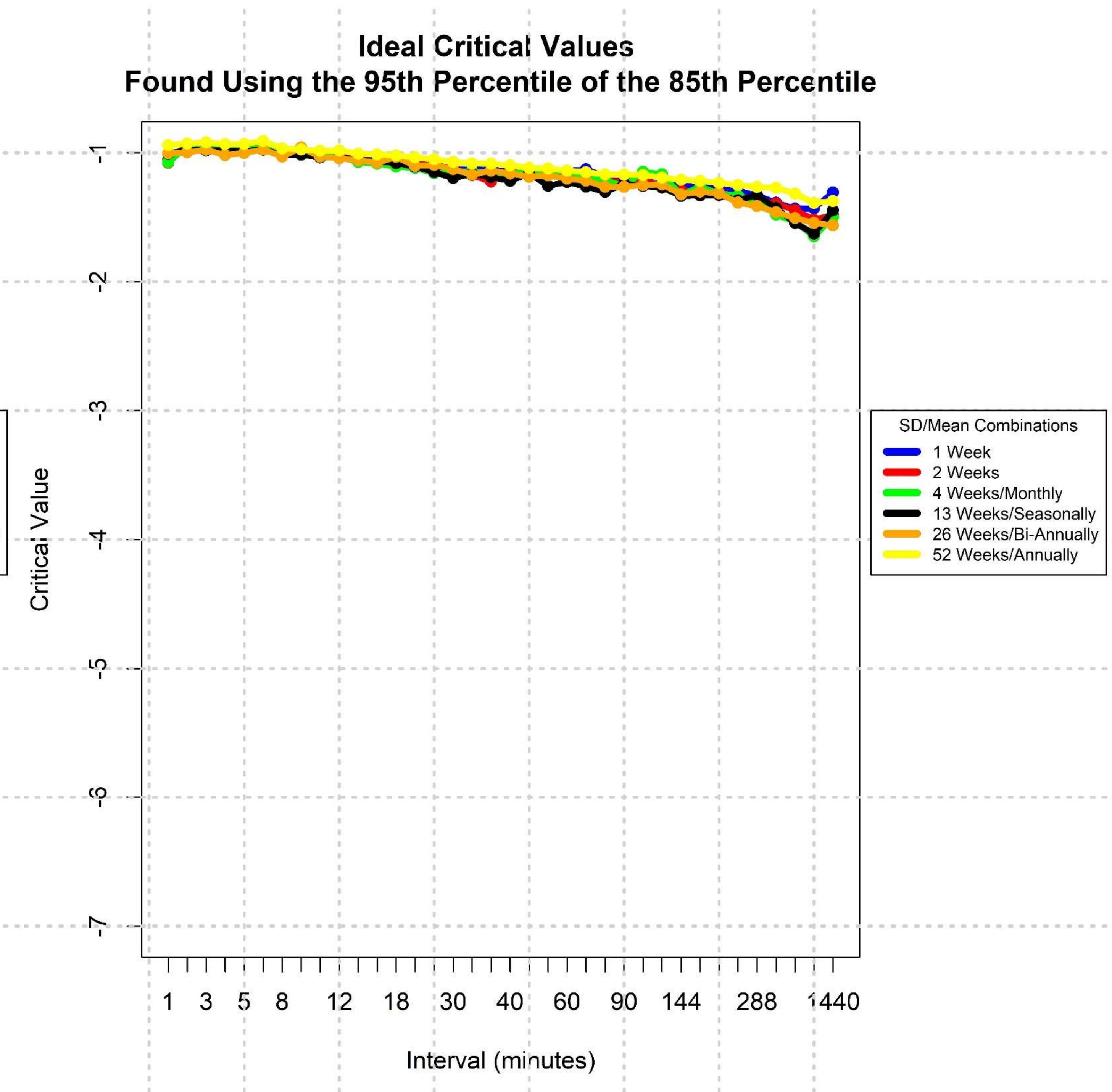
```
x=c("1 Week","2 Weeks","4 Weeks/Monthly","13 Weeks/Seasonally","26 Weeks/Bi-Annually","52 Weeks/Annually")
plot_ly(x=x,y=y,z=100*typeII,type = 'heatmap',colorbar=(list(title="Percentage \nof Errors (%)",nticks=10)))%>%
  layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(title='Combination')),yaxis=(list(title='Interval
Index')),title='Heatmap of Type II Errors for All Percentiles \nby Combination and Interval')
}#Heatmaps of best intervals and combos

{
  typeI=matrix(nrow = 10,ncol = 10)
  for(j in 1:10){ for(i in
1:10){typeI[j,i]=sum(typeIerror_test[,,,i,j],rm.na=T)/(sum(typeIerror_test[,,,i,j],rm.na=T)+sum(neg_neg_test[,,,i,j]))}}
  typeII=matrix(nrow = 10,ncol = 10)
  for(j in 1:10){ for(i in
1:10){typeII[j,i]=sum(typeIIerror_test[,,,i,j],rm.na=T)/(sum(typeIIerror_test[,,,i,j],rm.na=T)+sum(neg_neg_test[,,,i,j]))}}

  y=1:10
  x=1:10
  plot_ly(x=x,y=y,z=100*typeI,type = 'heatmap',colorbar=(list(title="Percentage \nof Errors (%)",nticks=10)))%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(title='Combination')),yaxis=(list(title='Interval
Index')),title='Heatmap of Type I Errors for All Percentiles \nby Combination and Interval')
  y=1:10
  x=1:10
  plot_ly(x=x,y=y,z=100*typeII,type = 'heatmap',colorbar=(list(title="Percentage \nof Errors (%)",nticks=10)))%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(title='Combination')),yaxis=(list(title='Interval
Index')),title='Heatmap of Type II Errors for All Percentiles \nby Combination and Interval')
}#Heatmaps of best intervals and combos
{typeI_sumhouse=array(dim = c(6,36,10,10))
  for(j in 1:36){ for(i in 1:6){for(k in 1:10){for(l in 1:10){typeI_sumhouse[i,j,k,l]=sum(typeIerror_test[i,j,,k,l],rm.na=T)}}}}
  typeII_sumhouse=array(dim = c(6,36,10,10))
  for(j in 1:36){ for(i in 1:6){for(k in 1:10){for(l in 1:10){typeII_sumhouse[i,j,k,l]=sum(typeIIerror_test[i,j,,k,l],rm.na=T)}}}}
  storex_sumhouse=array(dim = c(6,36,10,10))
  for(j in 1:36){ for(i in 1:6){for(k in 1:10){for(l in 1:10){storex_sumhouse[i,j,k,l]=sum(store_x_test[i,j,],rm.na=T)}}}}
  length(which(chk$dim1==6  & chk$dim3>=9 & chk$dim4==10))#51 of the 216 meet these specifications
  min((typeI_sumhouse+typeII_sumhouse)/storex_sumhouse)#21,600 options......
  chk1=sort((typeI_sumhouse+typeII_sumhouse)/storex_sumhouse)[216]#.01012645... TOP 1% .... errors per interval
  chk=data.frame(which((typeI_sumhouse+typeII_sumhouse)/storex_sumhouse<=chk1,arr.ind=TRUE))
  plot_ly(data=chk,x=chk$dim1, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad =
4),xaxis=(list(range=list(.5,6.5),title='Combination')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Total Errors by
Combination')
  plot_ly(data=chk,x=chk$dim2, type='histogram',nbinsx =36)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,36.5),title='Interval
Index')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Total Errors by Interval')
  plot_ly(data=chk,x=chk$dim3, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='1st
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Total Errors by 1st Percentile')
  plot_ly(data=chk,x=chk$dim4, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='2nd
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Total Errors by 2nd Percentile')
  (typeI_sumhouse[6,8,10,10])/(storex_sumhouse[6,8,10,10])*60*24*365 #minutes of error per year
  (typeII_sumhouse[6,8,10,10])/(storex_sumhouse[6,8,10,10])*60*24*365 #minutes of error per year
  chk1=sort((typeI_sumhouse)/storex_sumhouse)[216]#.009055... TOP 10 %
  chk=data.frame(which((typeI_sumhouse)/storex_sumhouse<=chk1,arr.ind=TRUE))
  plot_ly(data=chk,x=chk$dim1, type='histogram',nbinsx =6)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad =
4),xaxis=(list(range=list(.5,6.5),title='Combination')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Type I Errors by
Combination')
  plot_ly(data=chk,x=chk$dim2, type='histogram',nbinsx =36)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,36.5),title='Interval
Index')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Type I Errors by Interval')
  plot_ly(data=chk,x=chk$dim3, type='histogram',nbinsx =10)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='1st
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Type I Errors by Percentile')

  plot_ly(data=chk,x=chk$dim4, type='histogram',nbinsx =10,marker=list(line=list(width=0d0)))%>%
```

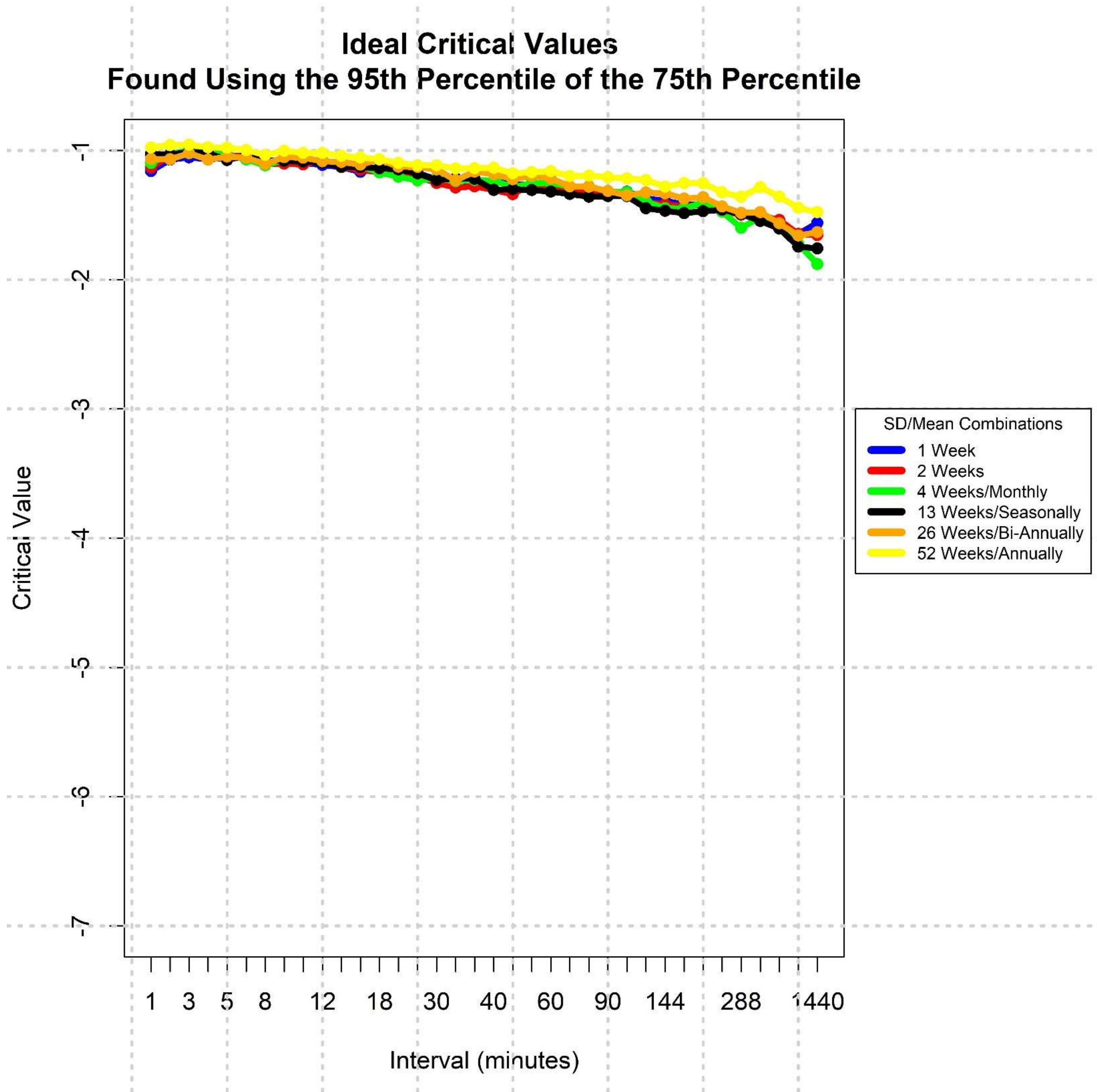
```
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='2nd
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Type I Errors by 2nd Percentile')

  (typeI_sumhouse[6,26,10,10])/(storex_sumhouse[6,26,10,10])*60*24*365 #minutes of error per year
  (typeII_sumhouse[6,26,10,10])/(storex_sumhouse[6,26,10,10])*60*24*365 #minutes of error per year

  chk1=sort((typeII_sumhouse)/storex_sumhouse)[216]#.0000324... TOP 10 %
  chk=data.frame(which((typeII_sumhouse)/storex_sumhouse<=chk1,arr.ind=TRUE))
  plot_ly(data=chk,x=chk$dim1, type='histogram',nbinsx =6)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad =
4),xaxis=(list(range=list(.5,6.5),title='Combination')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Type II Errors by
Combination')
  plot_ly(data=chk,x=chk$dim2, type='histogram',nbinsx =36)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,36.5),title='Interval
Index')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Type II Errors by Interval')
  plot_ly(data=chk,x=chk$dim3, type='histogram',nbinsx =10)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='1st
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Type II Errors by Percentile')
  plot_ly(data=chk,x=chk$dim4, type='histogram',nbinsx =10)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='2nd
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 1% Lowest Type II Errors by 2nd Percentile')

  chk3=numeric(400)
  for(i in 1:400)chk3[i]=typeI_sumhouse[chk[i,1],chk[i,2],chk[i,3],chk[i,4]]
  order(chk3)[c(1)]
  (typeI_sumhouse[6,1,10,10])/(storex_sumhouse[6,1,10,10])*60*24*365 #minutes of error per year
  (typeII_sumhouse[6,1,10,10])/(storex_sumhouse[6,1,10,10])*60*24*365 #minutes of error per year
  }#histograms of top 10% options

 {typeI_sumhouse=array(dim = c(6,36,10,10))
  for(j in 1:36){ for(i in 1:6){for(k in 1:10){for(l in 1:10){typeI_sumhouse[i,j,k,l]=sum(typeIerror_test[i,j,,k,l],rm.na=T)}}}}
  typeII_sumhouse=array(dim = c(6,36,10,10))
  for(j in 1:36){ for(i in 1:6){for(k in 1:10){for(l in 1:10){typeII_sumhouse[i,j,k,l]=sum(typeIIerror_test[i,j,,k,l],rm.na=T)}}}}
  storex_sumhouse=array(dim = c(6,36,10,10))
  for(j in 1:36){ for(i in 1:6){for(k in 1:10){for(l in 1:10){storex_sumhouse[i,j,k,l]=sum(store_x_test[i,j,],rm.na=T)}}}}
  min((typeI_sumhouse+typeII_sumhouse)/storex_sumhouse)#21,600 options......
  chk1=sort((typeI_sumhouse+typeII_sumhouse)/storex_sumhouse)[21600-2160]#.51401... BOTTOM 10 %
  chk=data.frame(which((typeI_sumhouse+typeII_sumhouse)/storex_sumhouse>=chk1,arr.ind=TRUE))
  plot_ly(data=chk,x=chk$dim1, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad =
4),xaxis=(list(range=list(.5,6.5),title='Combination')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Total Errors by
Combination')
  plot_ly(data=chk,x=chk$dim2, type='histogram',nbinsx =36)%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,36.5),title='Interval
Index')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Total Errors by Interval')
  plot_ly(data=chk,x=chk$dim3, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='1st
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Total Errors by Percentile')
  plot_ly(data=chk,x=chk$dim4, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='2nd
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Total Errors by 2nd Percentile')
  chk1=sort((typeI_sumhouse)/storex_sumhouse)[21600-2160]#.513791... BOTTOM 10 %
  chk=data.frame(which((typeI_sumhouse)/storex_sumhouse>=chk1,arr.ind=TRUE))
  plot_ly(data=chk,x=chk$dim1, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad =
4),xaxis=(list(range=list(.5,6.5),title='Combination')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Type I Errors by
Combination')
  plot_ly(data=chk,x=chk$dim2, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4,nbinsx =36),xaxis=(list(range=list(.5,36.5),title='Interval
Index')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Type I Errors by Interval')
  plot_ly(data=chk,x=chk$dim3, type='histogram')%>%
    layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='1st
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Type I Errors by Percentile')
  plot_ly(data=chk,x=chk$dim4, type='histogram')%>%
```

```
      layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='2nd
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Type I Errors by 2nd Percentile')
    chk1=sort((typeII_sumhouse)/storex_sumhouse)[21600-2160]#.001938... BOTTOM 10 %
    chk=data.frame(which((typeII_sumhouse)/storex_sumhouse>=chk1,arr.ind=TRUE))
    plot_ly(data=chk,x=chk$dim1, type='histogram')%>%
      layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad =
4),xaxis=(list(range=list(.5,6.5),title='Combination')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Type II Errors by
Combination')
    plot_ly(data=chk,x=chk$dim2, type='histogram')%>%
      layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4,nbinsx =36),xaxis=(list(range=list(.5,36.5),title='Interval
Index')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Type II Errors by Interval')
    plot_ly(data=chk,x=chk$dim3, type='histogram')%>%
      layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=list(.5,10.5),title='1st
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Type II Errors by Percentile')

    plot_ly(data=chk,x=chk$dim4, type='histogram')%>%
      layout(margin=list(l = 80,r = 80,b = 100,t = 80, pad = 4),xaxis=(list(range=c(.5,10.5),title='2nd
Percentile')),yaxis=(list(title='count')),title='Histogram of Top 10% Largest Type II Errors by 2nd Percentile')
  }#histograms of BOTTOM 10% options
```

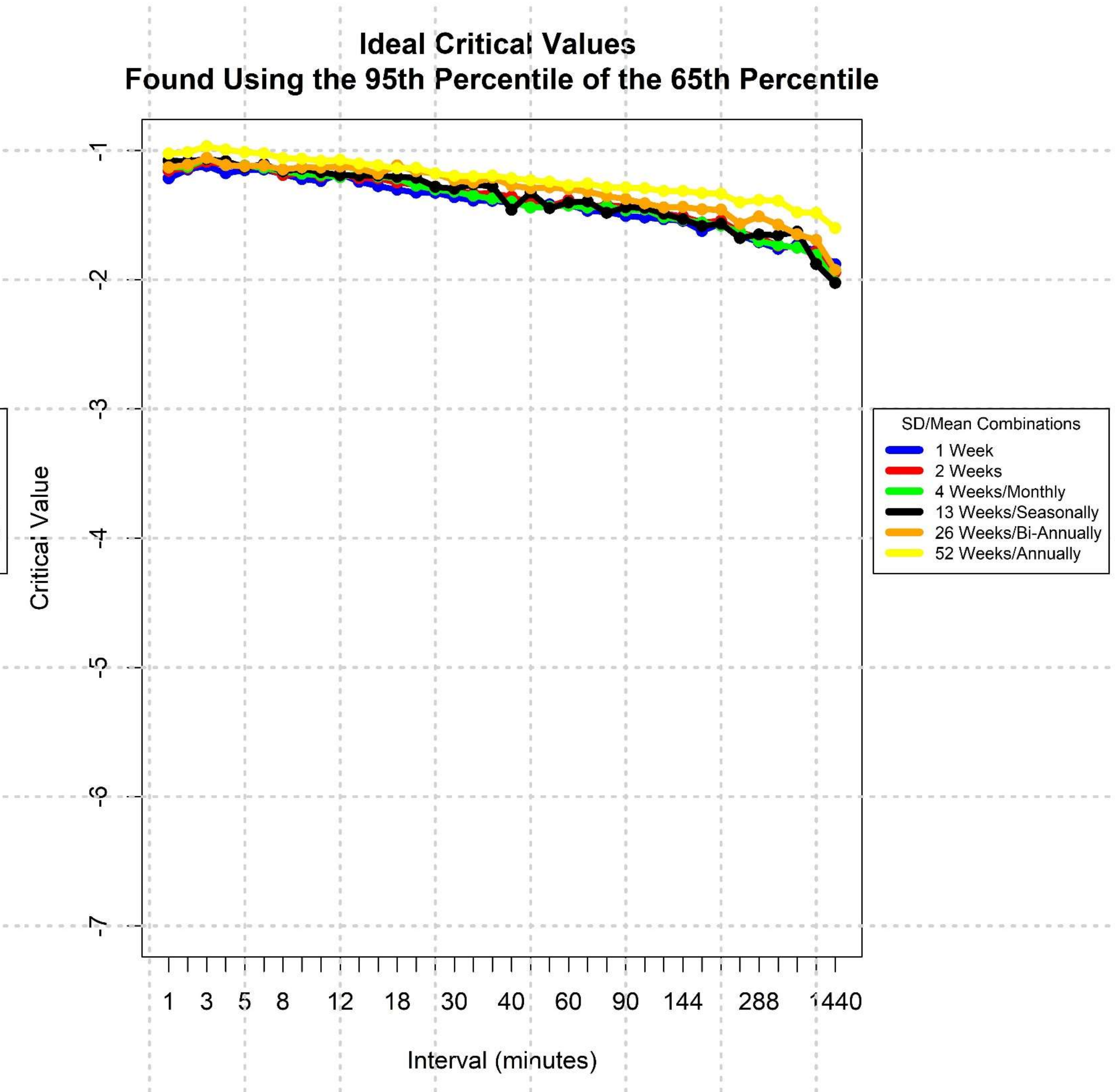
Appendix 3. Pecan Street Ideal Critical Value Sensitivity- Critical Value Graphs for 95th Second Percentile



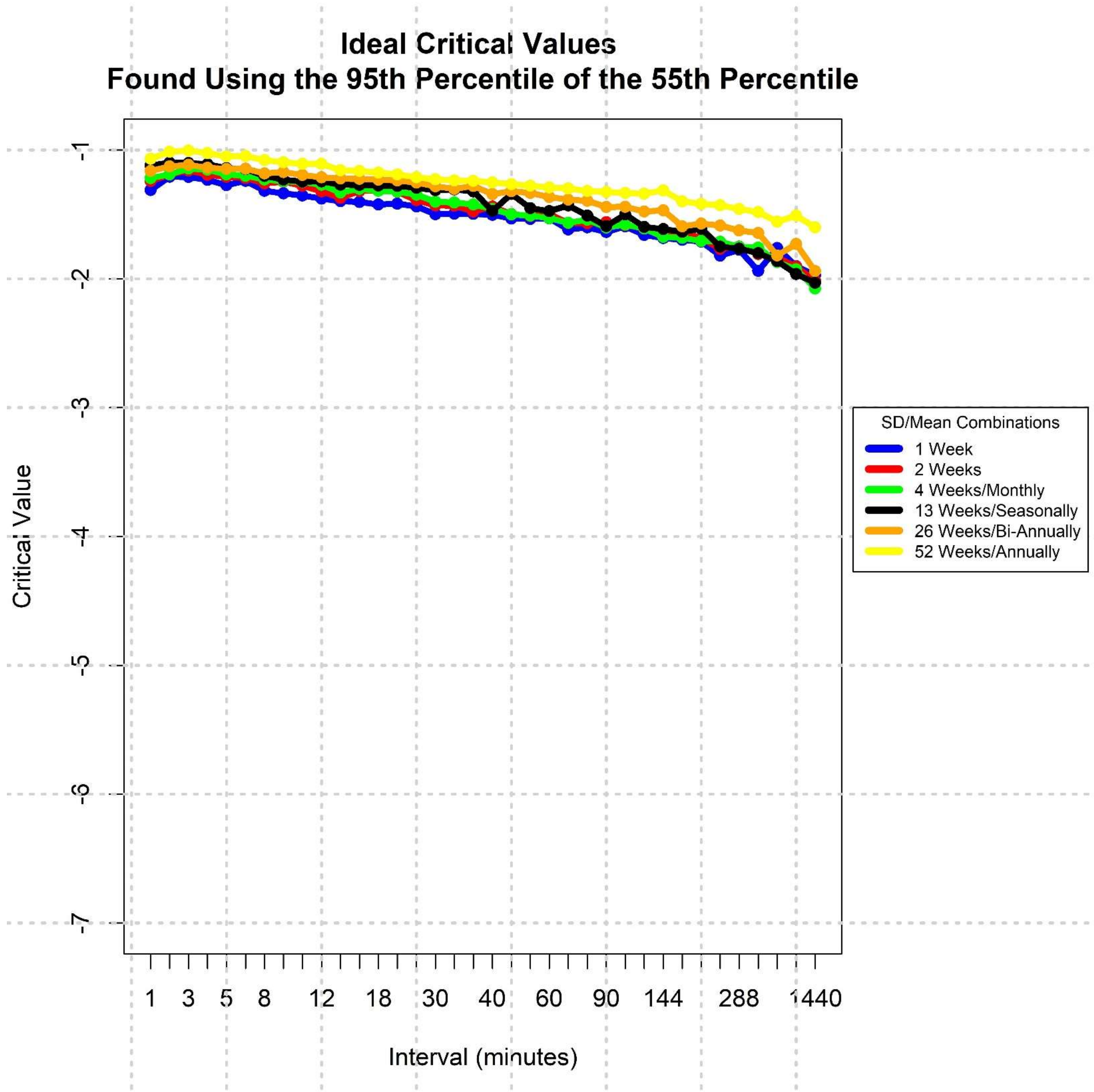


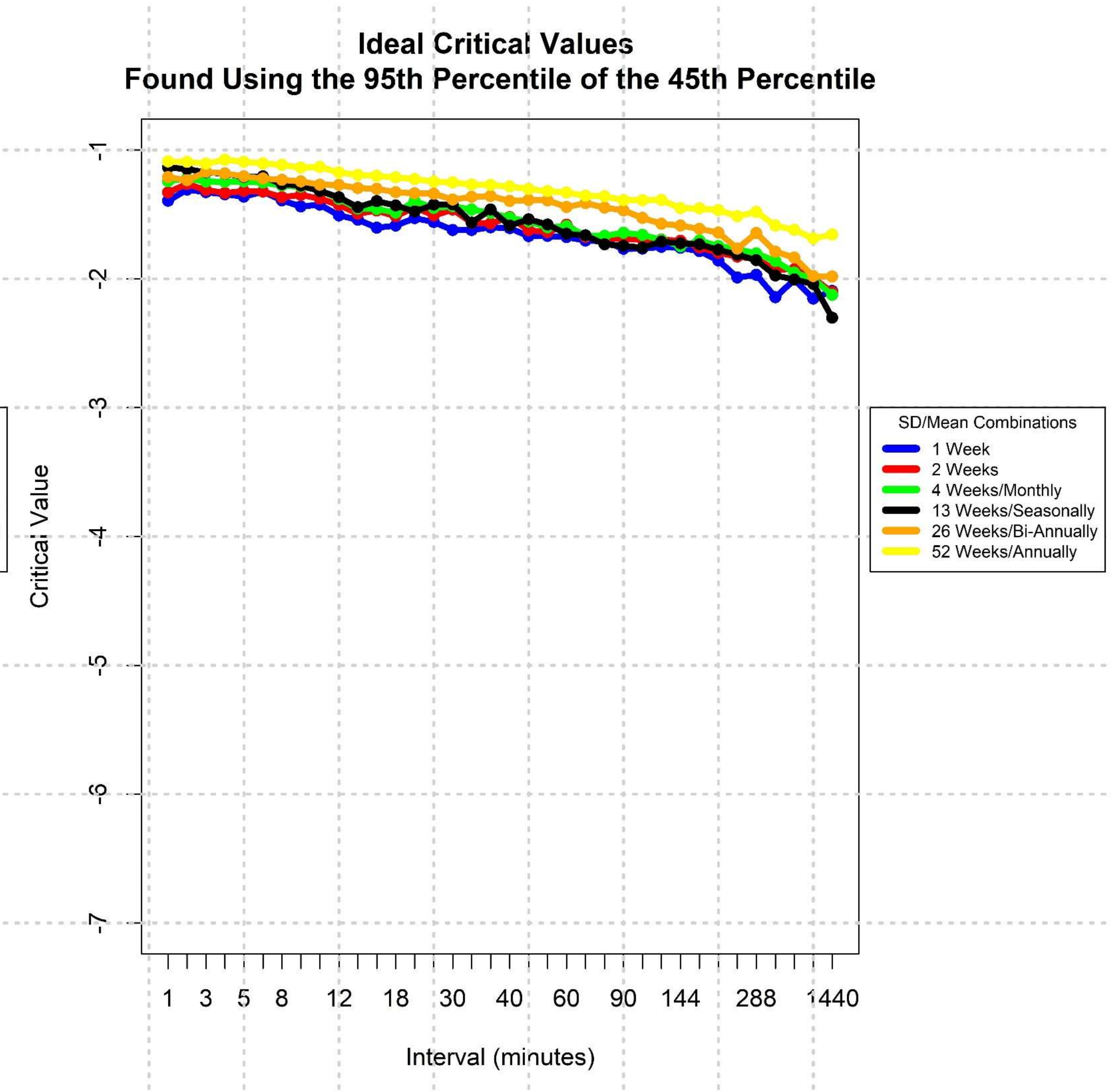


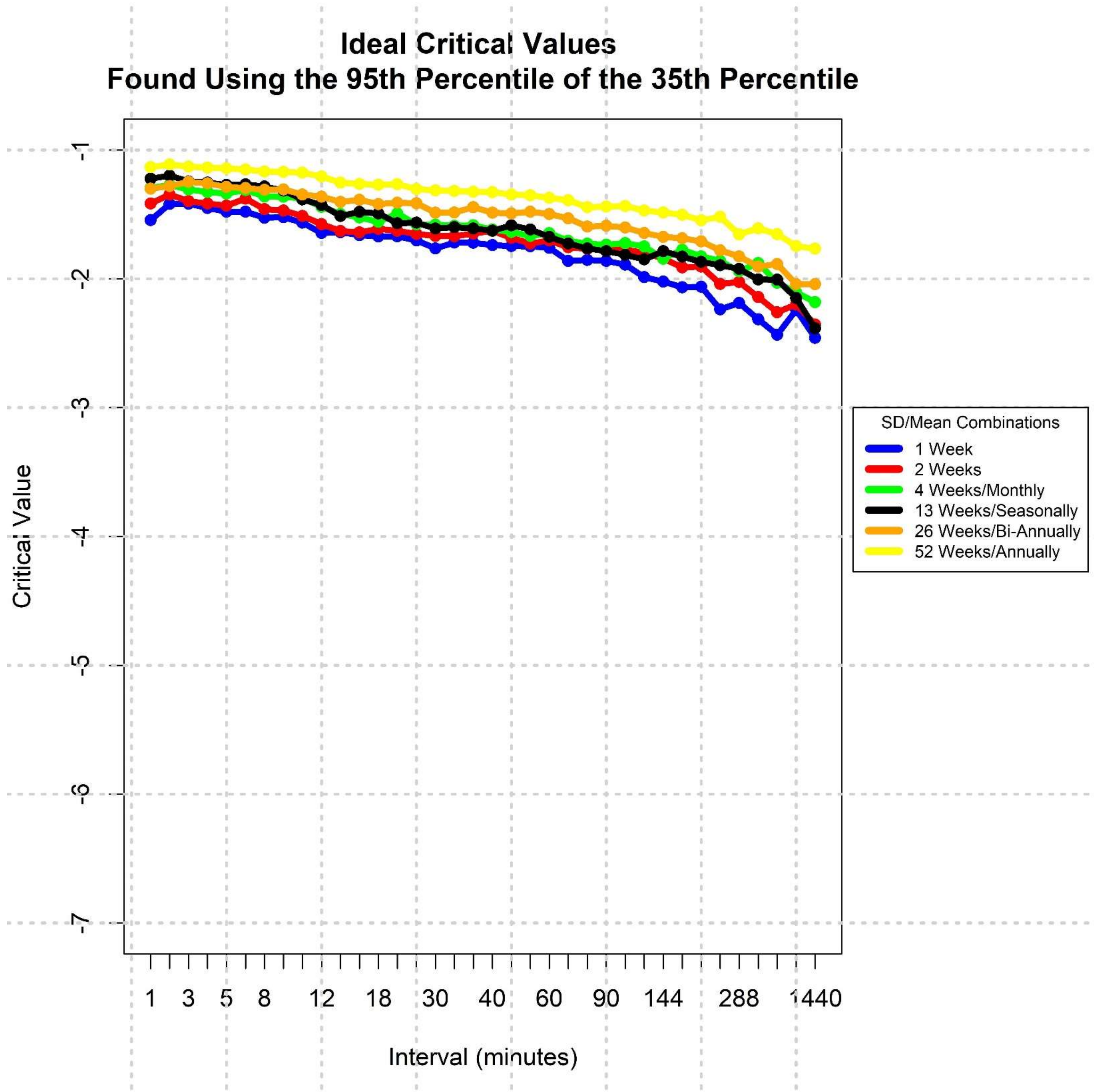
Ideal Critical Values
Found Using the 95th Percentile of the 75th Percentile
Critical Value
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

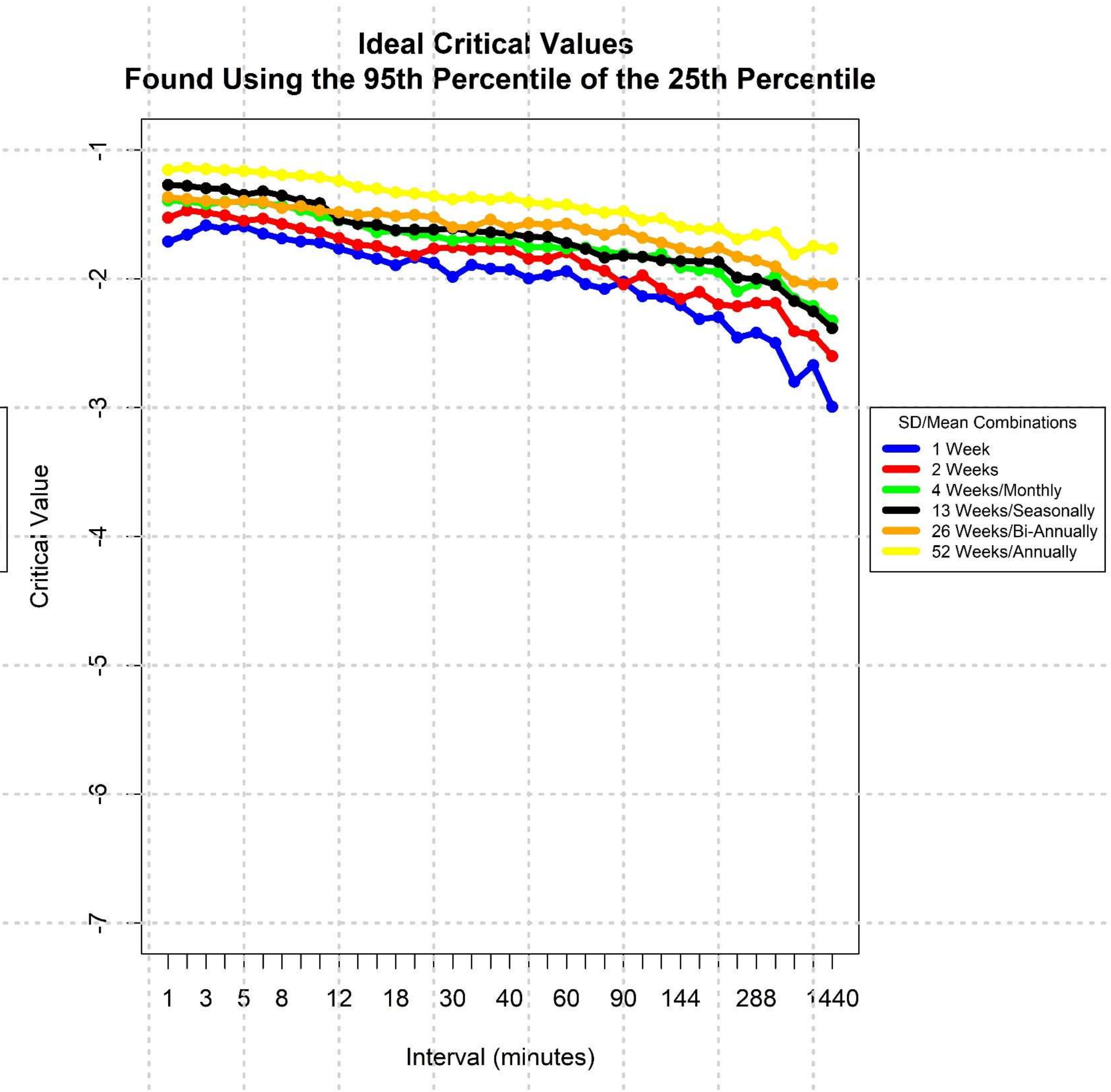


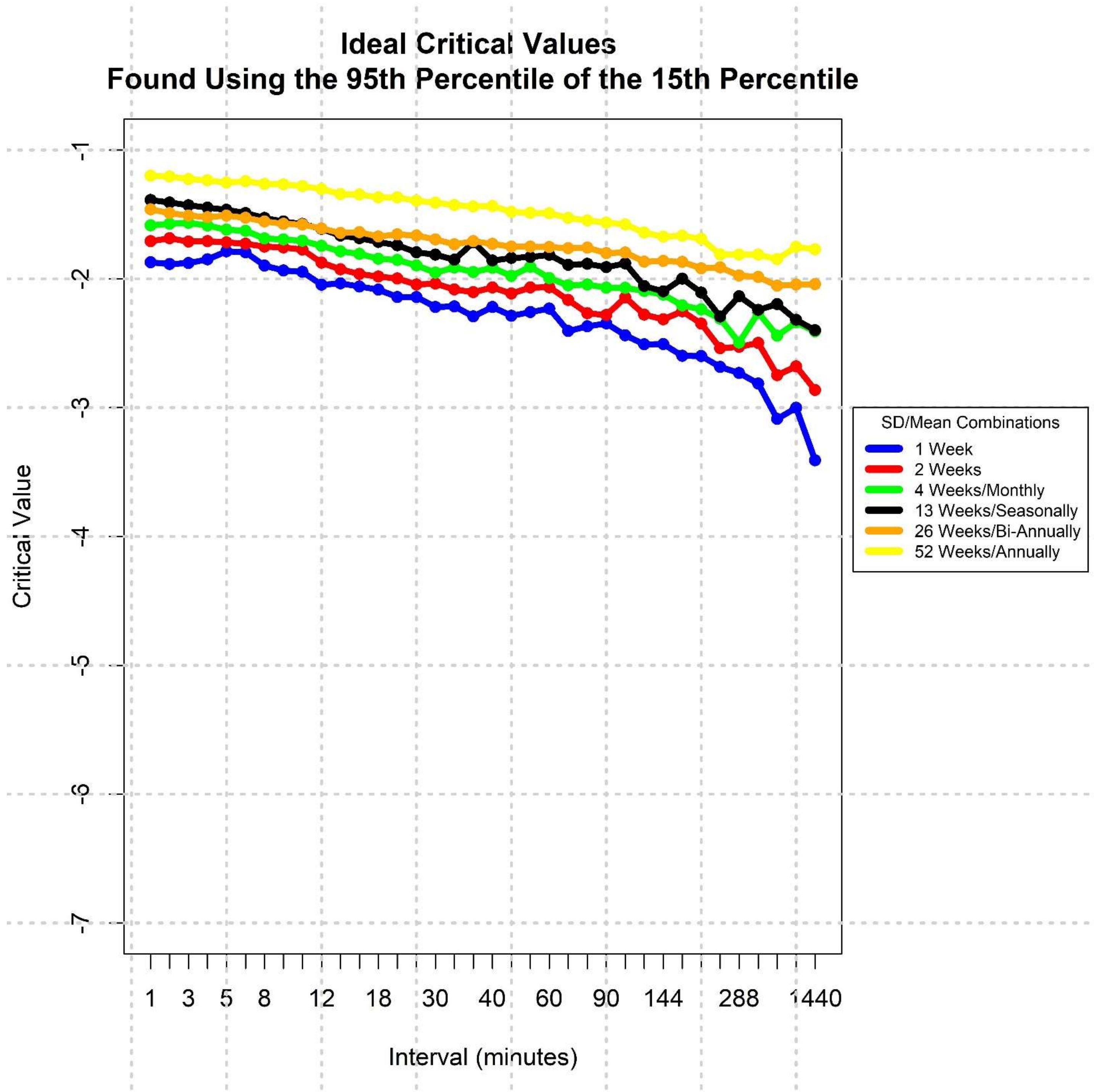
Ideal Critical Values
Found Using the 95th Percentile of the 65th Percentile
Critical Value
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

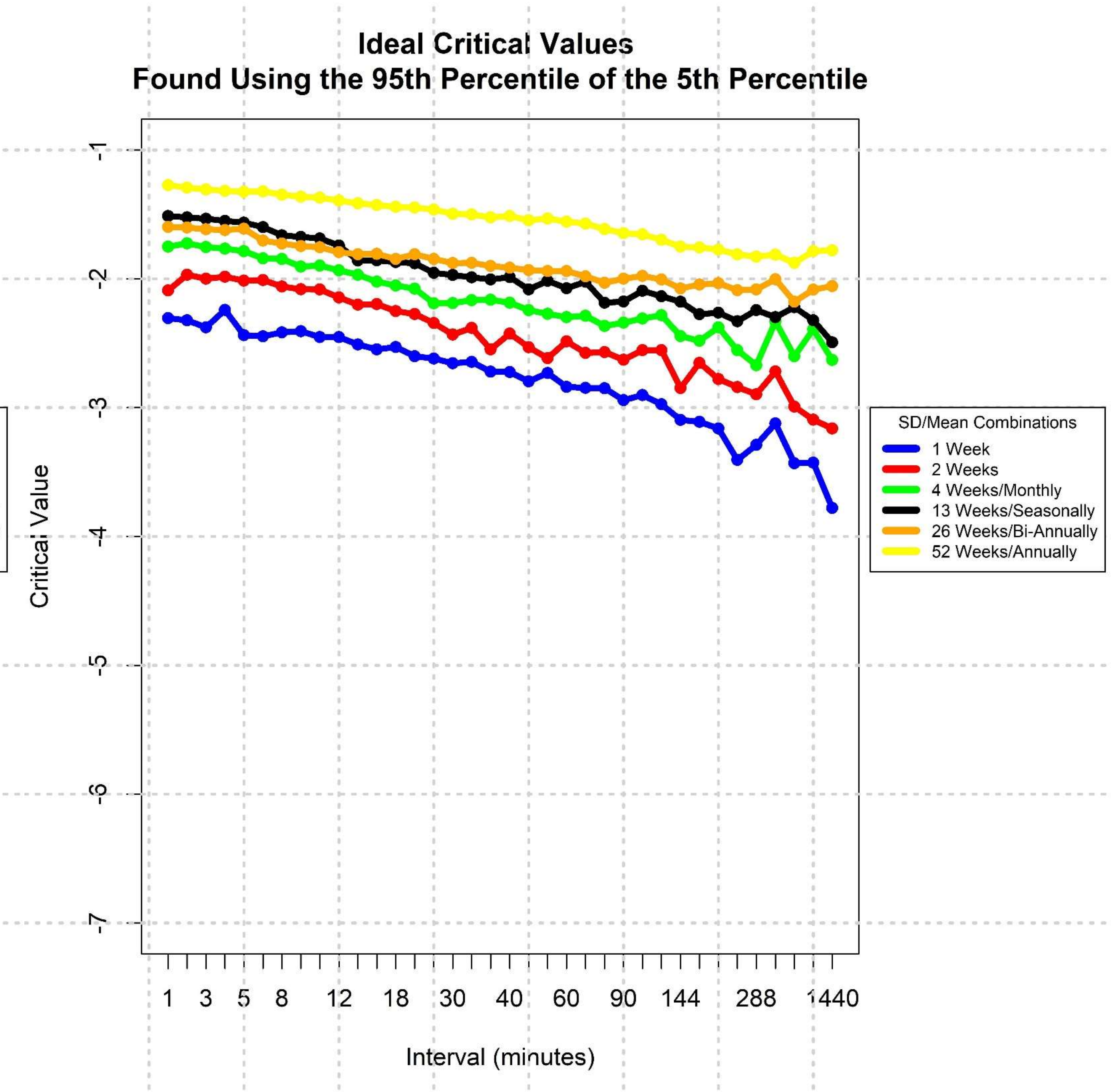

Ideal Critical Values
Found Using the 95th Percentile of the 55th Percentile
Critical Value
-1
-2
-3
-4
-5
-6
-7
1 3 5 8 12 18 30 40 60 90 144 288 1440
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually


Ideal Critical Values
Found Using the 95th Percentile of the 45th Percentile
Critical Value
-1
-2
-3
-4
-5
-6
-7
1 3 5 8 12 18 30 40 60 90 144 288 1440
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

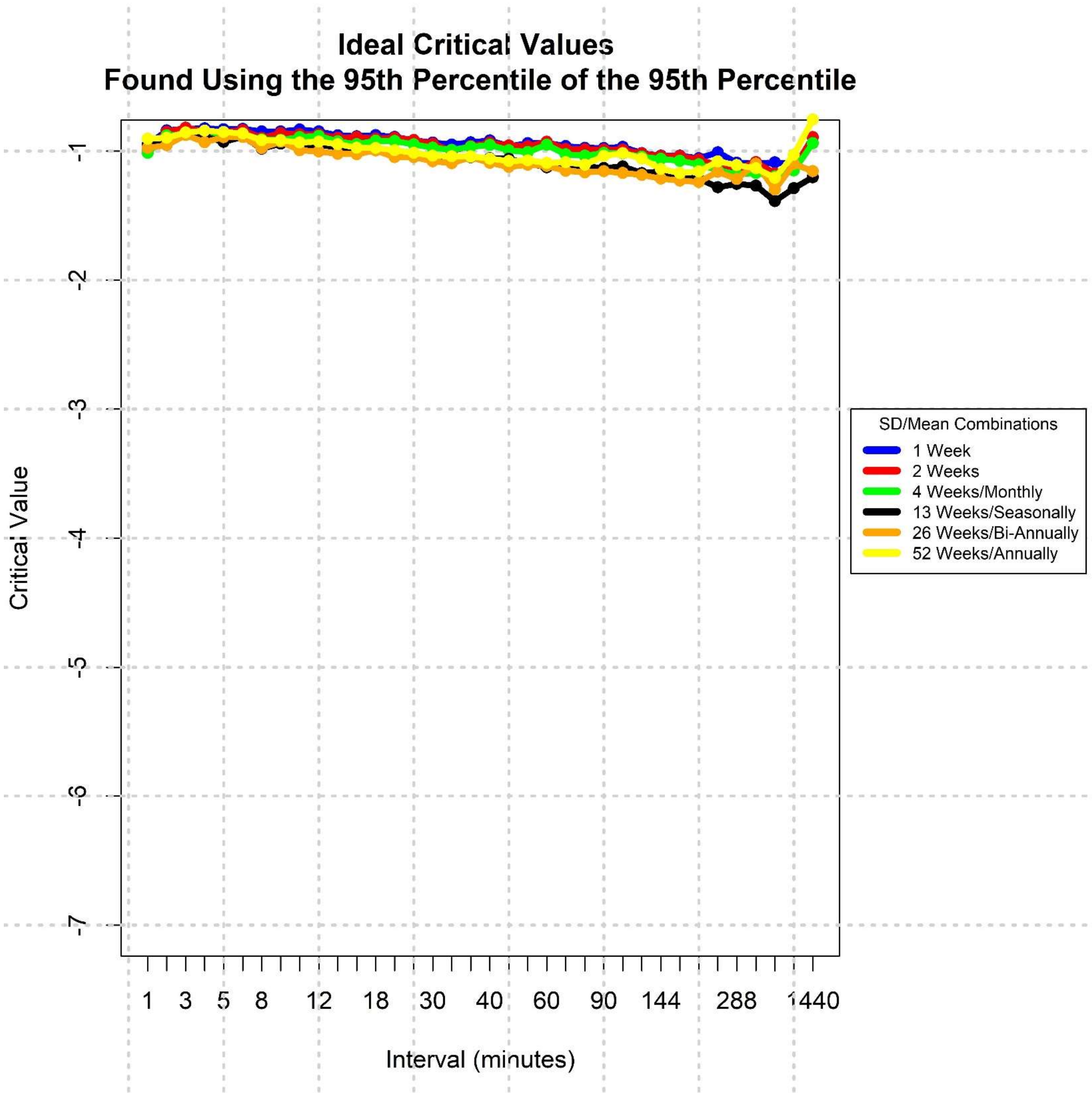

Ideal Critical Values
Found Using the 95th Percentile of the 35th Percentile
Critical Value
-1
-2
-3
-4
-5
-6
-7
1 3 5 8 12 18 30 40 60 90 144 288 1440
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

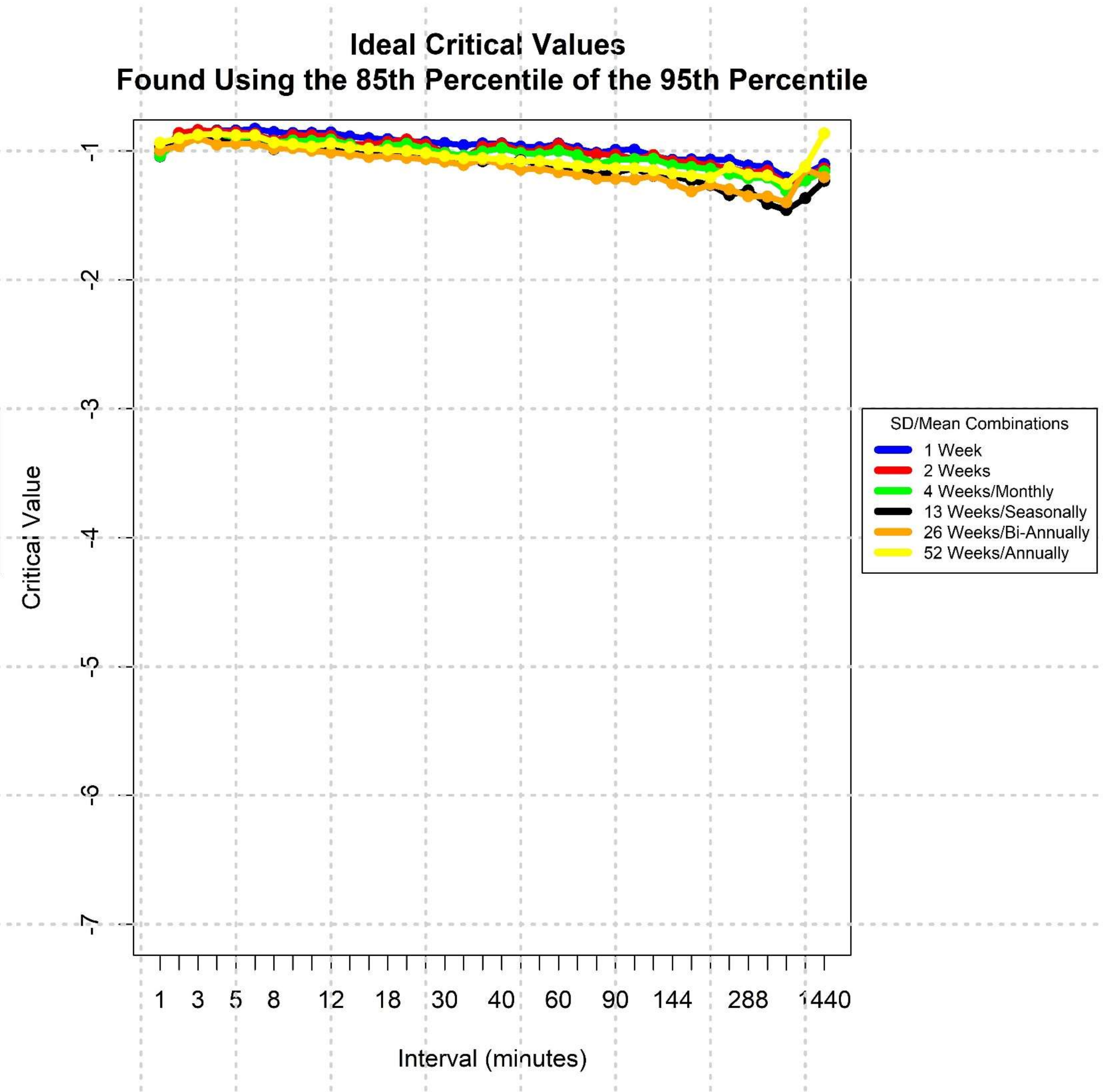

Ideal Critical Values
Found Using the 95th Percentile of the 25th Percentile
Critical Value
-1
-2
-3
-4
-5
-6
-7
1 3 5 8 12 18 30 40 60 90 144 288 1440
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

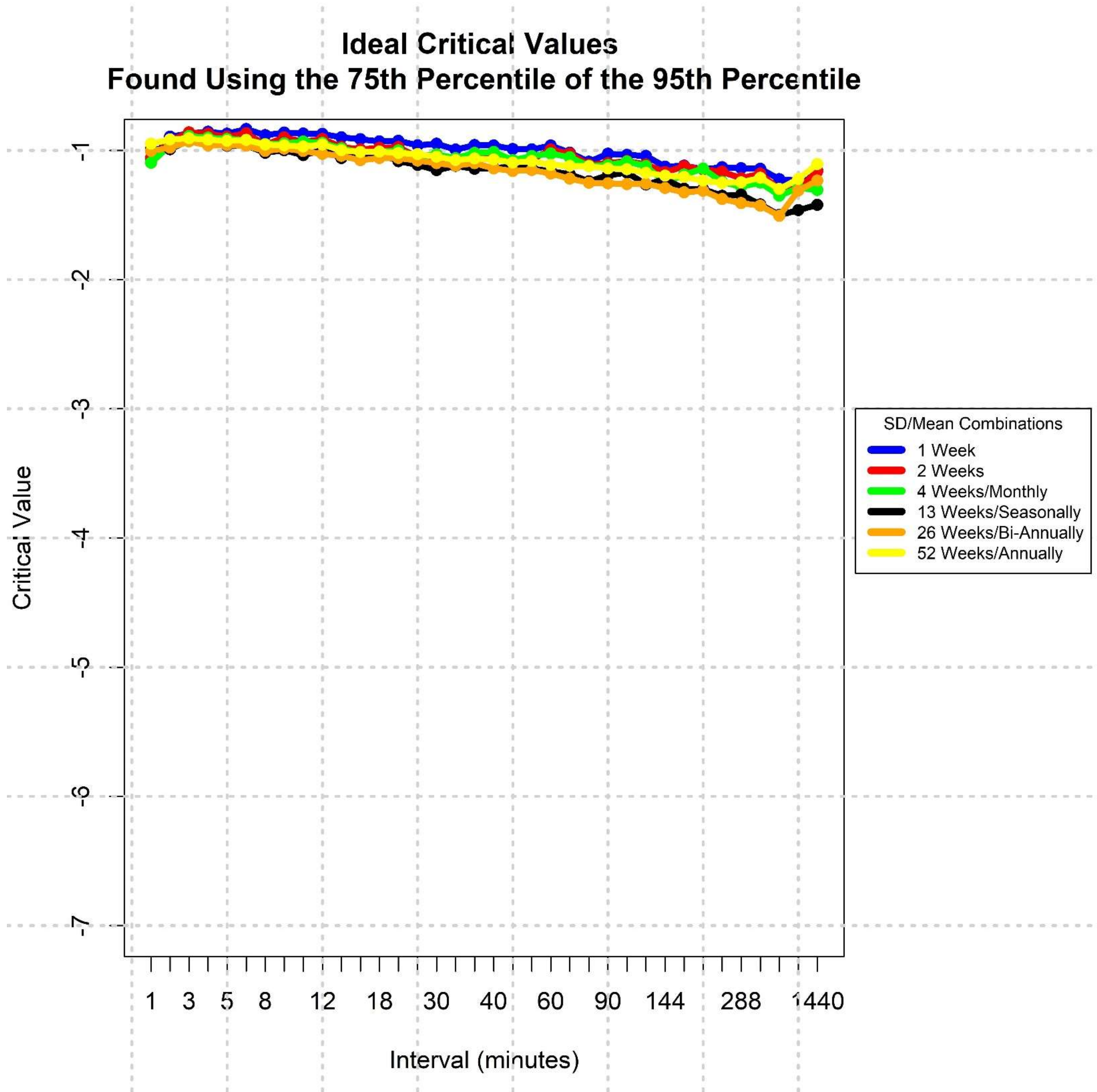

Ideal Critical Values
Found Using the 95th Percentile of the 15th Percentile
Critical Value
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually
-1
-2
-3
-4
-5
-6
-7
1
3
5
8
12
18
30
40
60
90
144
288
1440

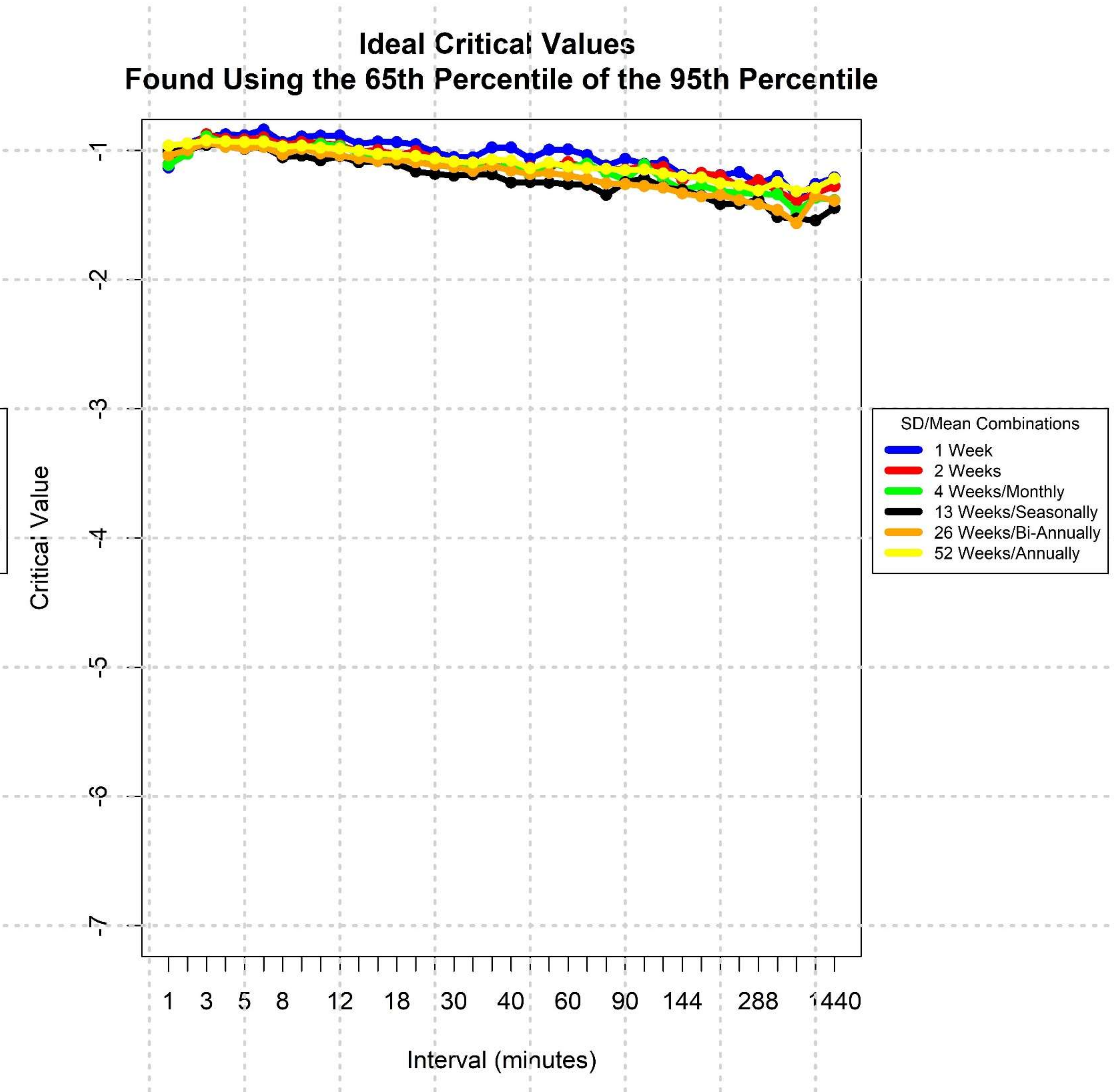

Ideal Critical Values
Found Using the 95th Percentile of the 5th Percentile
Critical Value
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually
-1
-2
-3
-4
-5
-6
-7
1
3
5
8
12
18
30
40
60
90
144
288
1440

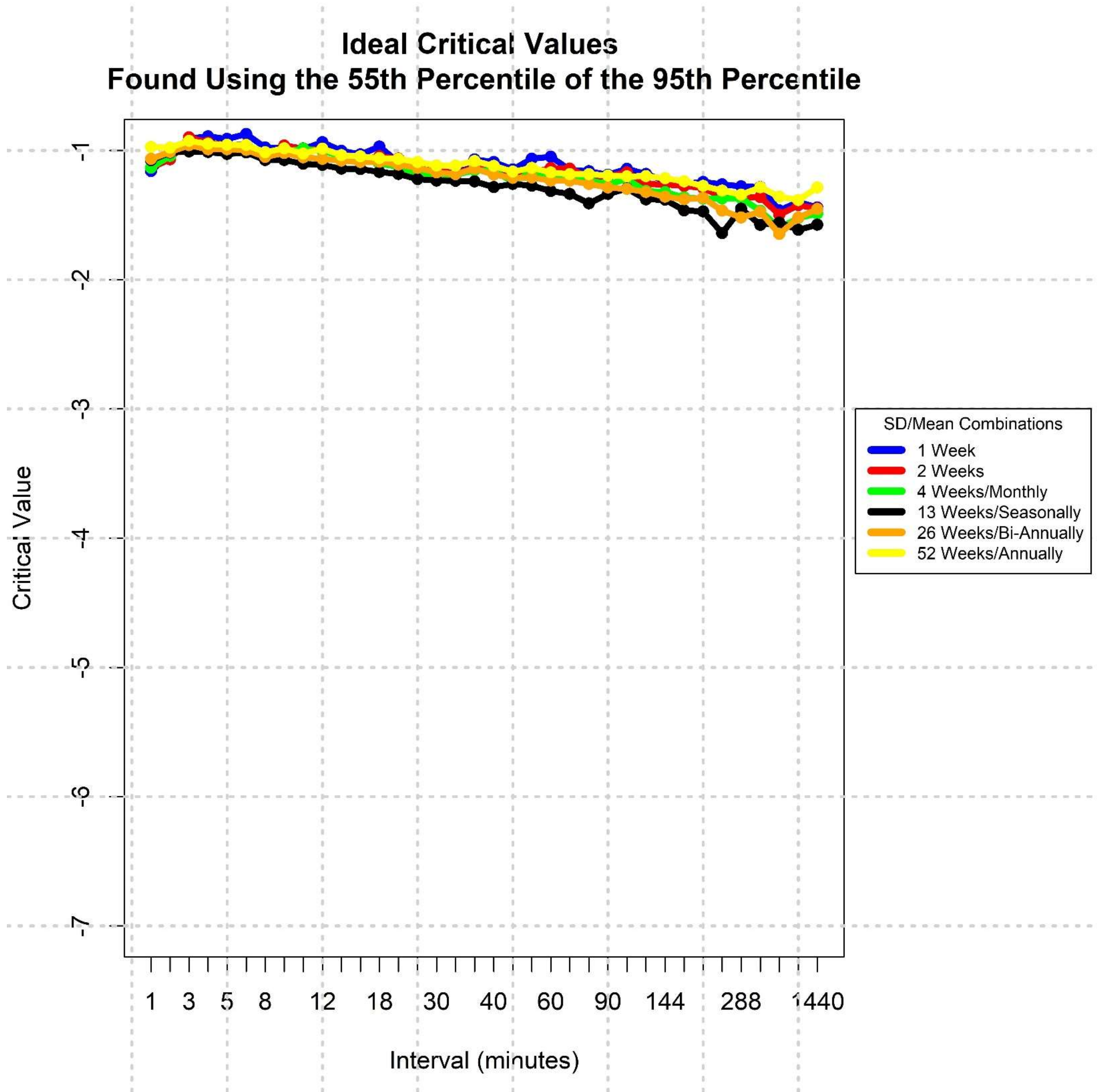
Appendix 4. Pecan Street Ideal Critical Value Sensitivity- Critical Value Graphs for 95th First Percentile

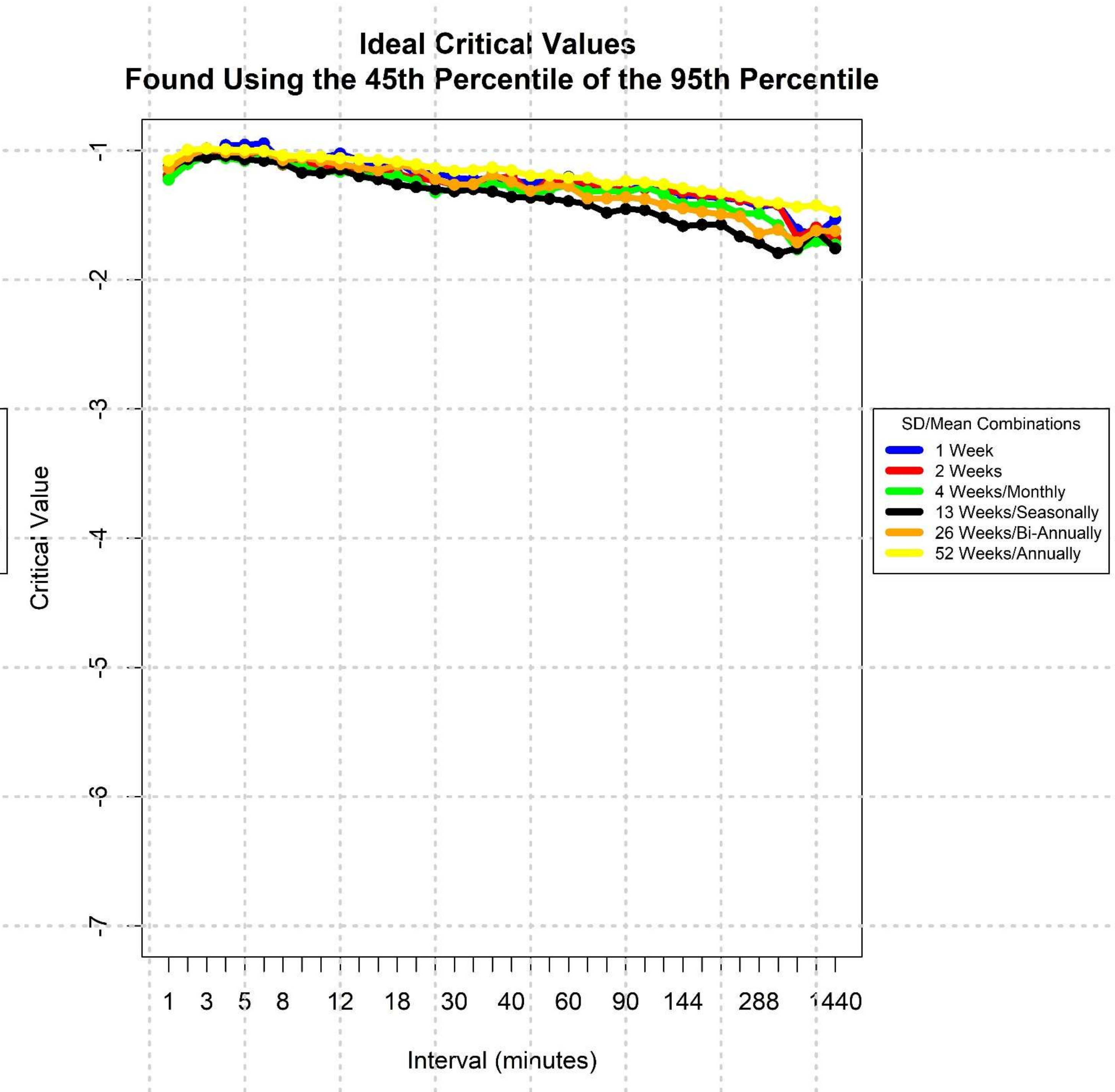


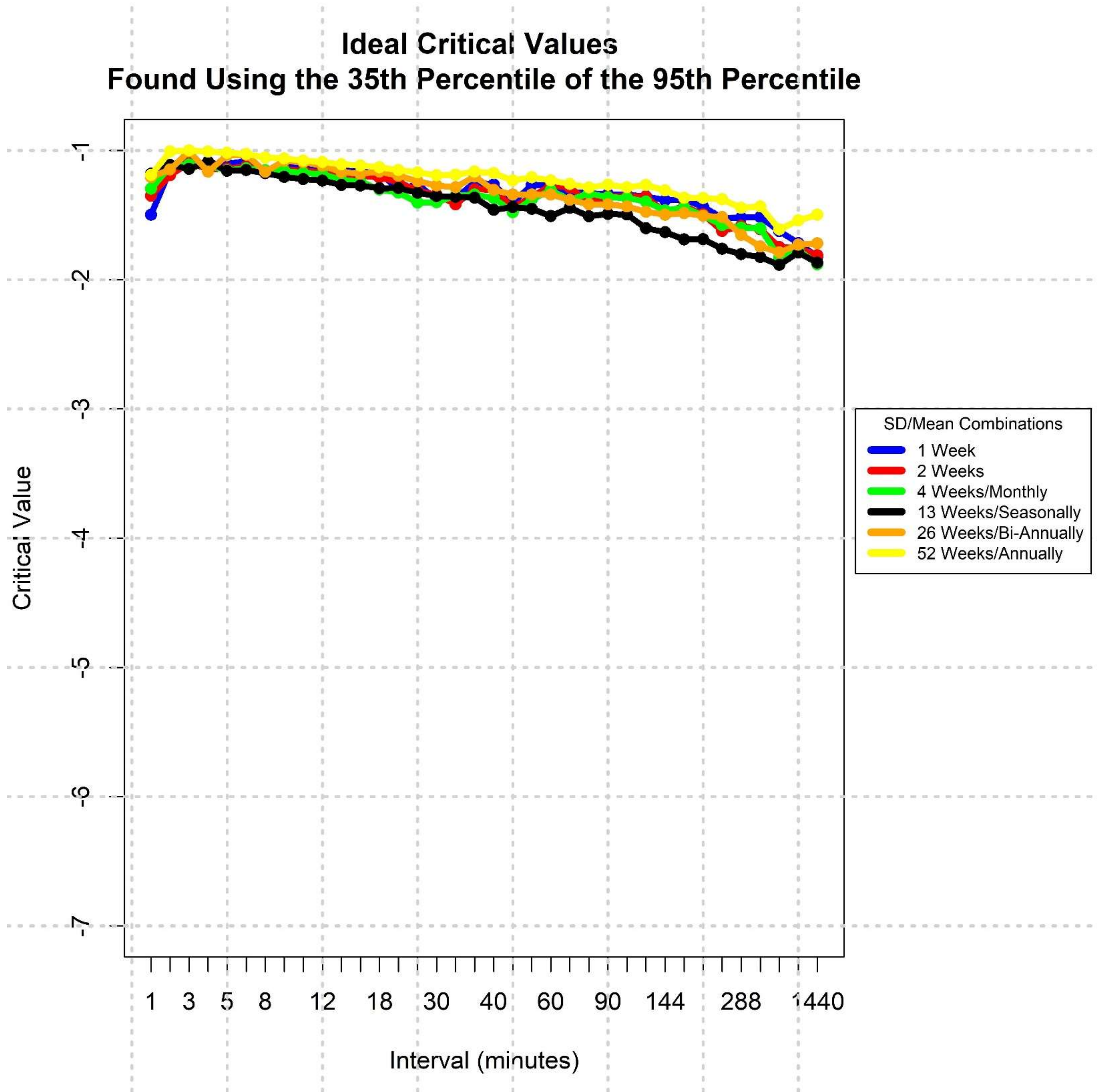


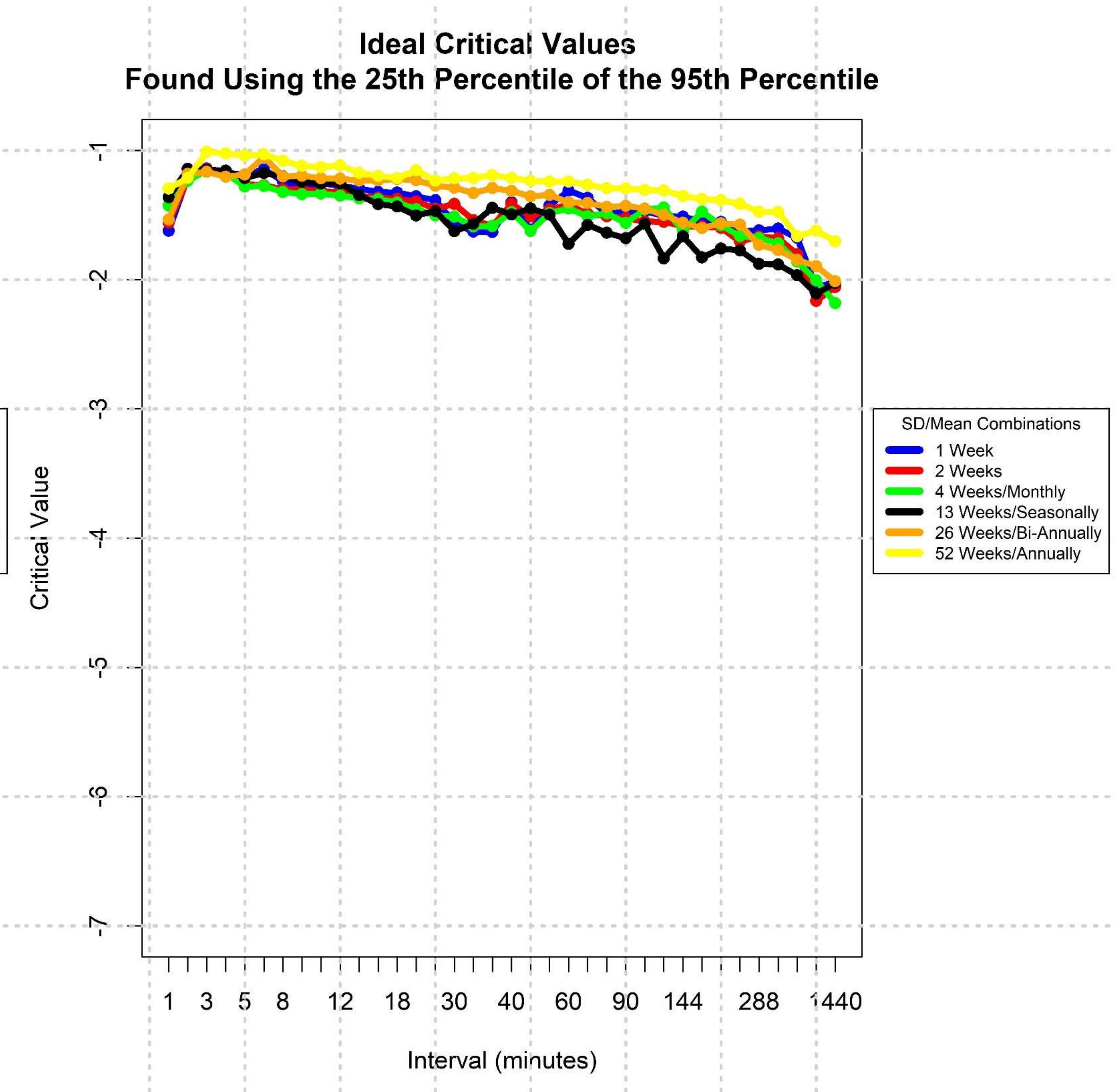

Ideal Critical Values
Found Using the 75th Percentile of the 95th Percentile
Critical Value
-1
-2
-3
-4
-5
-6
-7
1 3 5 8 12 18 30 40 60 90 144 288 1440
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

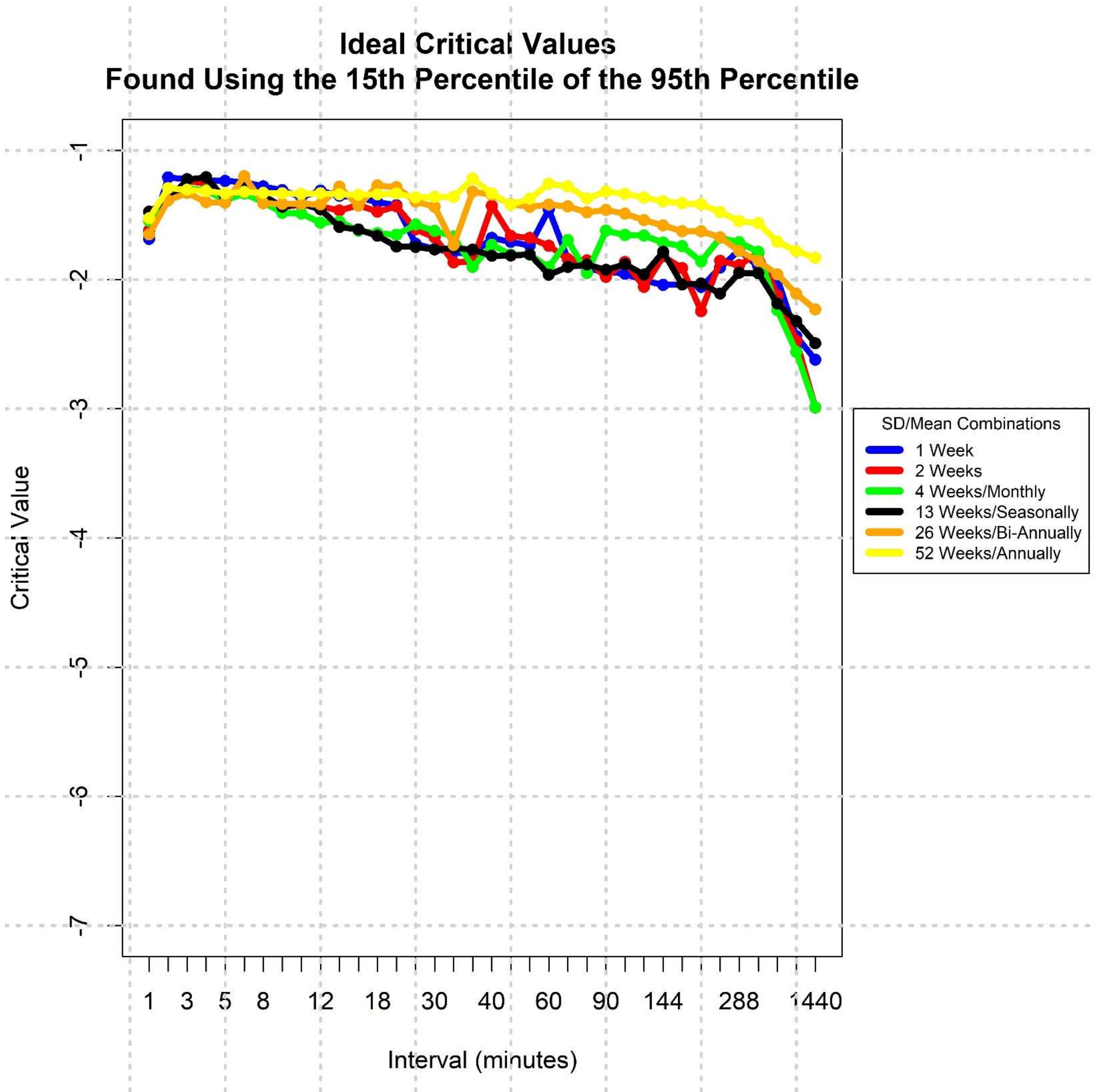

Ideal Critical Values
Found Using the 65th Percentile of the 95th Percentile
Critical Value
-1
-2
-3
-4
-5
-6
-7
1 3 5 8 12 18 30 40 60 90 144 288 1440
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

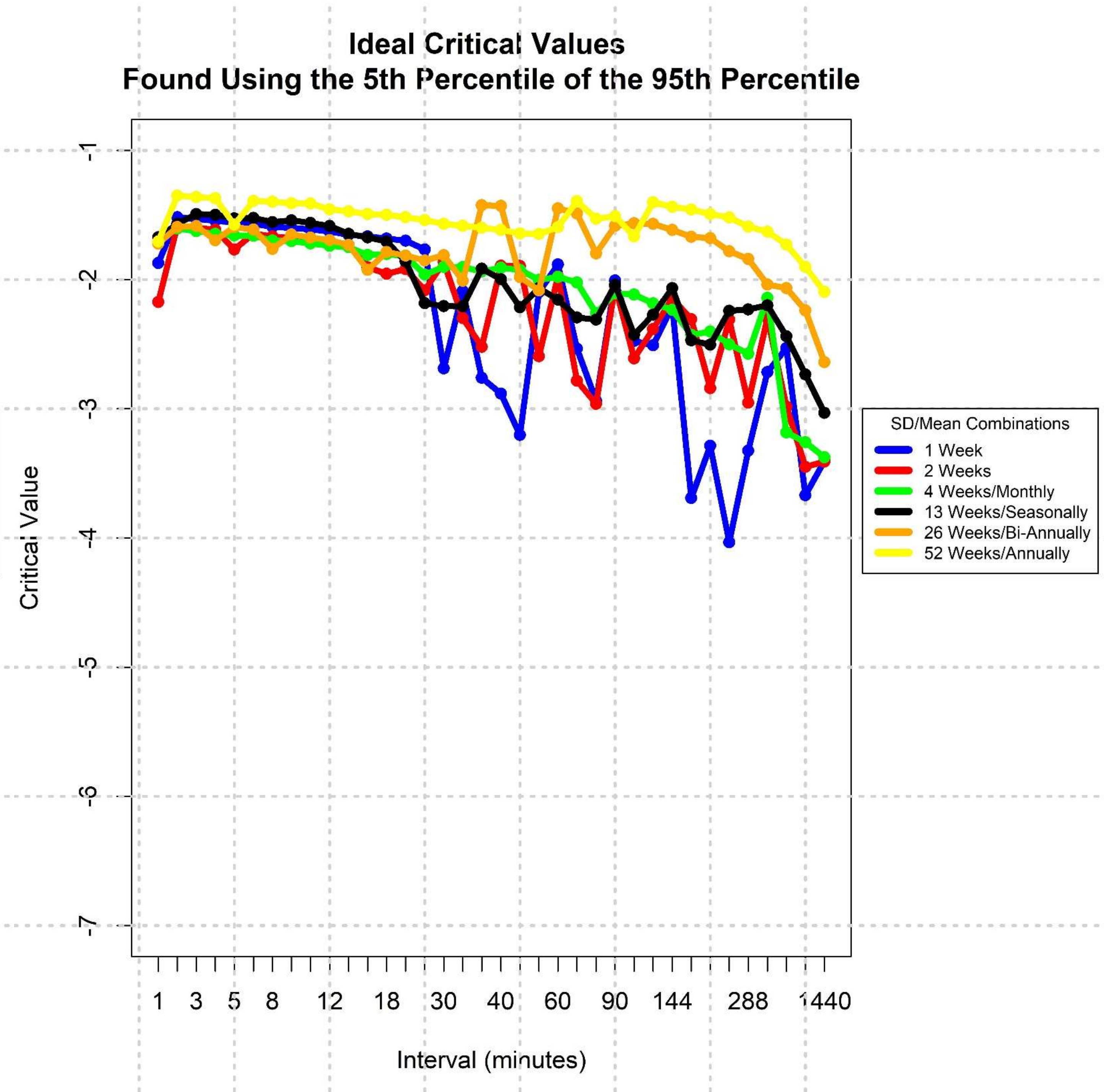

Ideal Critical Values
Found Using the 55th Percentile of the 95th Percentile
Critical Value
-1
-2
-3
-4
-5
-6
-7
1
3
5
8
12
18
30
40
60
90
144
288
1440
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually


Ideal Critical Values
Found Using the 45th Percentile of the 95th Percentile
Critical Value
-1
-2
-3
-4
-5
-6
-7
1
3
5
8
12
18
30
40
60
90
144
288
1440
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

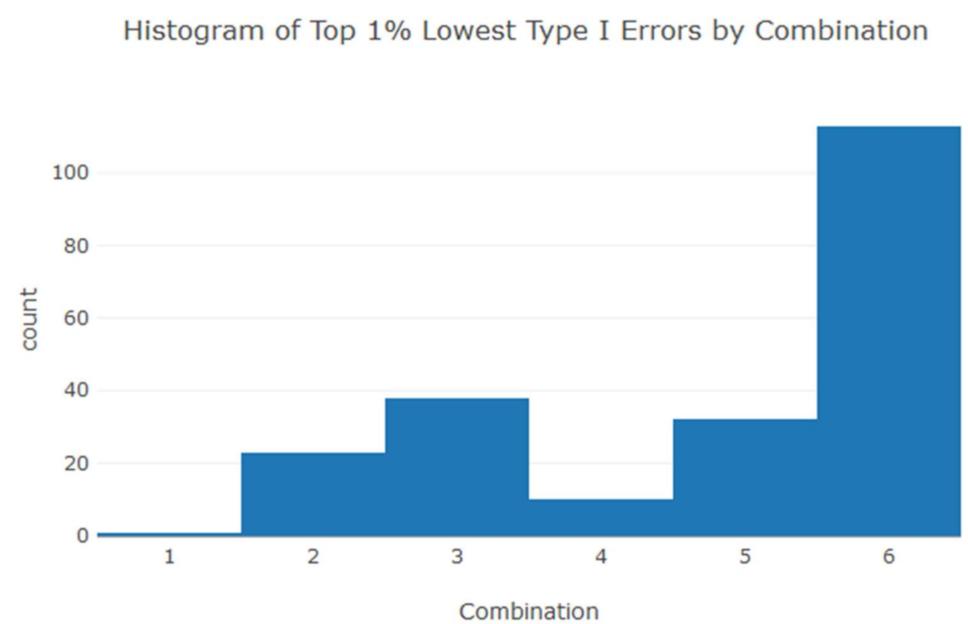

Ideal Critical Values
Found Using the 35th Percentile of the 95th Percentile
Critical Value
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

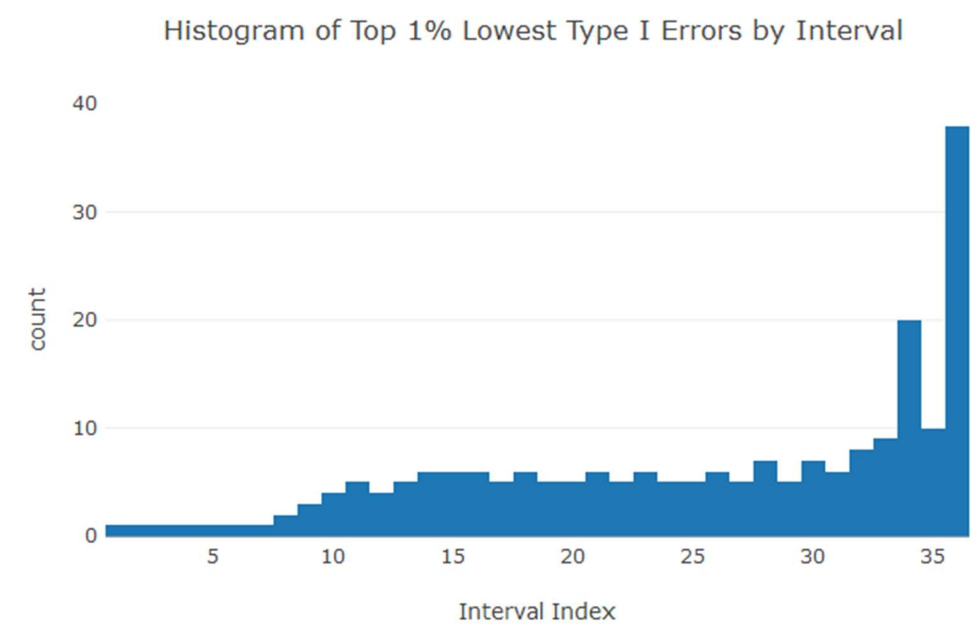

Ideal Critical Values
Found Using the 25th Percentile of the 95th Percentile
Critical Value
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

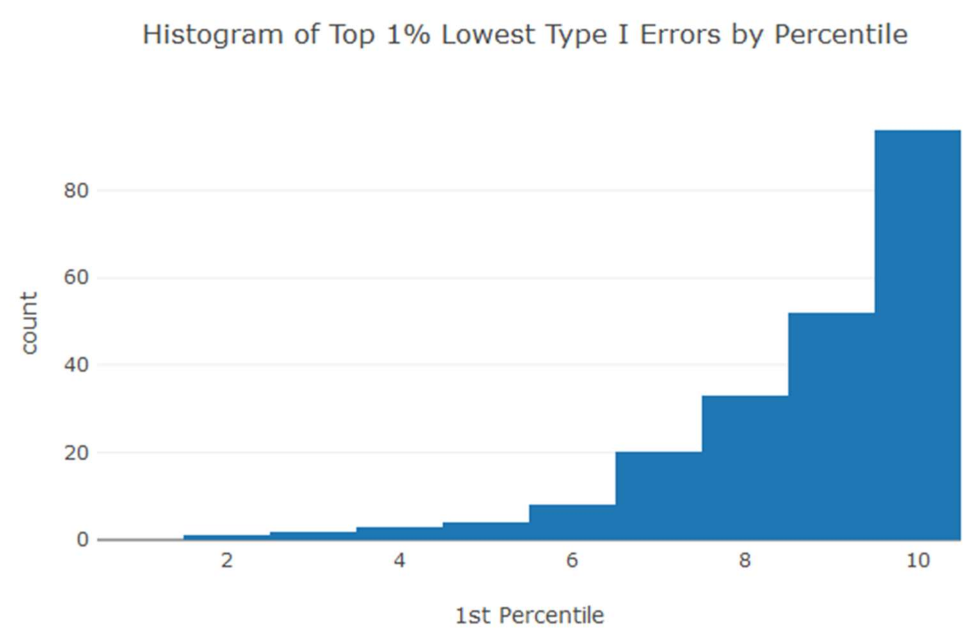

Ideal Critical Values
Found Using the 15th Percentile of the 95th Percentile
Critical Value
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

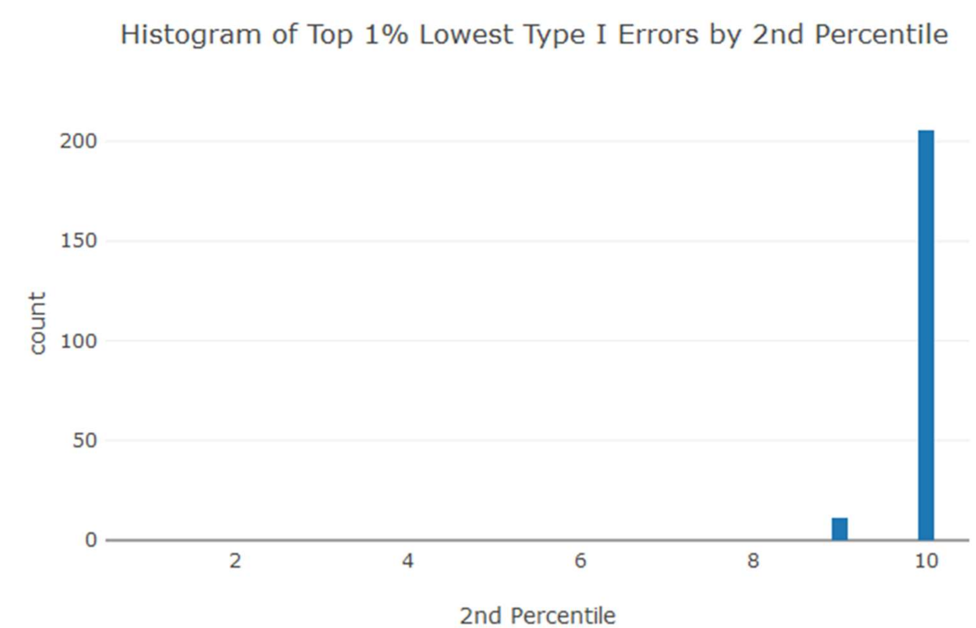

Ideal Critical Values
Found Using the 5th Percentile of the 95th Percentile
Critical Value
Interval (minutes)
SD/Mean Combinations
1 Week
2 Weeks
4 Weeks/Monthly
13 Weeks/Seasonally
26 Weeks/Bi-Annually
52 Weeks/Annually

Appendix 5. Pecan Street Ideal Critical Value Sensitivity -Reduction of Type I Errors

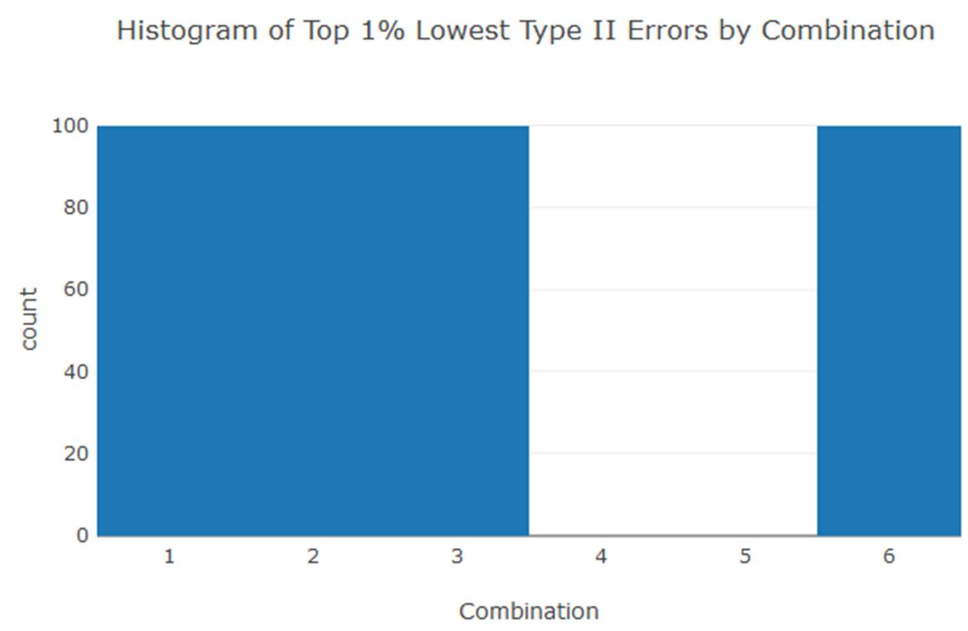


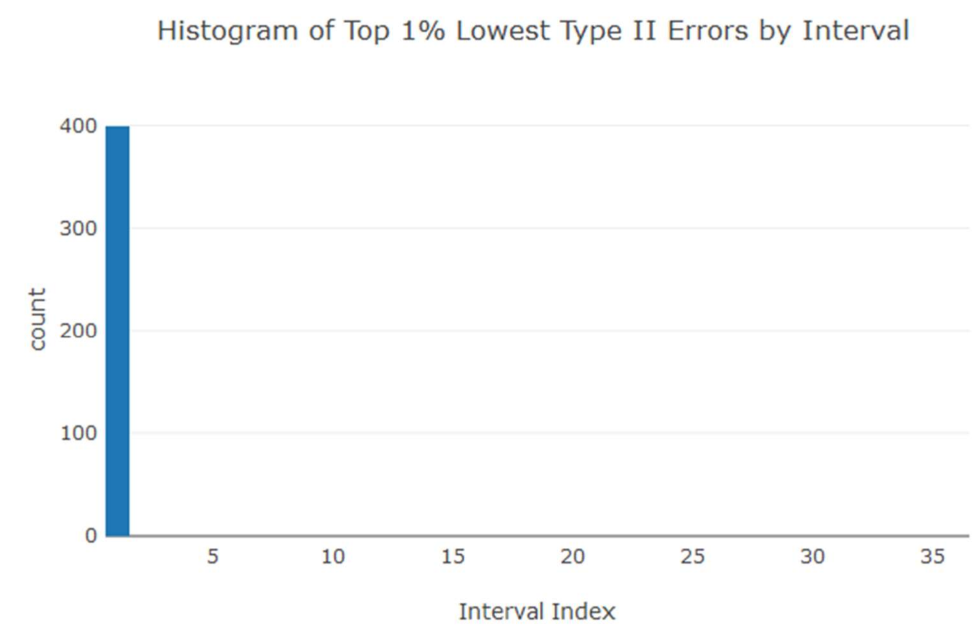


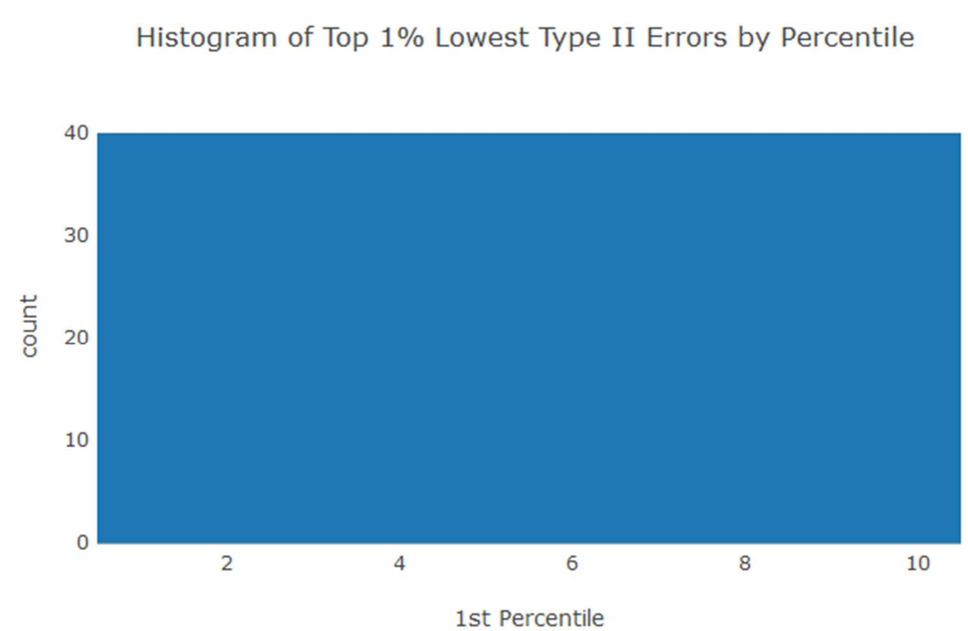


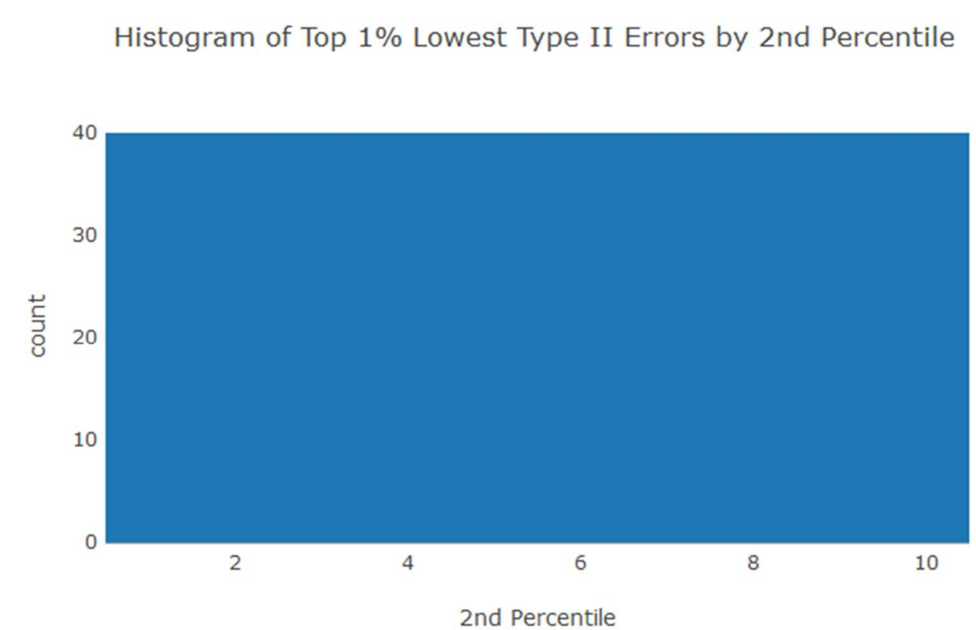


Appendix 6. Pecan Street Ideal Critical Value Sensitivity -Reduction of Type II Errors









Appendix 7. MITLL Electrical System Architectures

Architectures Used for Installation Modeling

Architecture Option	Building Generators Standby	Building Generators Running	Central Generators Standby	Central Generators Running	Grid-Tied PV	Islandable PV	UPS	Battery	Microgrid	Cogeneration	Fuel Cells	Electrical Grid
1	0	0	0	0	0	0	0	0	0	0	0	1
2	72	0	0	0	0	0	0	0	0	0	0	1
3	72	0	0	0	1	0	0	0	0	0	0	1
4	72	0	0	0	0	0	72	0	0	0	0	1
5	72	0	0	0	1	0	72	0	0	0	0	1
6	0	0	16	0	0	0	0	0	0	0	0	1
7	0	0	16	0	1	0	0	0	0	0	0	1
8	0	0	16	0	0	0	0	0	1	0	0	1
9	0	0	16	0	0	1	0	0	1	0	0	1
10	0	0	16	0	0	1	0	1	1	0	0	1
11	0	0	16	0	0	1	1	0	1	0	0	1
12	0	0	16	0	0	0	1	0	1	0	0	1
13	0	0	16	0	0	1	0	0	1	1	0	1
14	0	0	16	0	0	1	1	0	1	1	0	1
15	0	0	16	0	0	1	1	0	1	0	1	1
16	0	0	16	0	0	1	1	0	1	1	1	1
17	0	0	16	0	0	0	0	0	1	1	0	1
18	0	0	16	0	0	0	0	0	1	0	1	1
19	36	0	8	0	0	0	0	0	0	0	0	1
20	36	0	8	0	1	0	0	0	0	0	0	1
21	36	0	8	0	0	0	36	0	0	0	0	1
22	36	0	8	0	1	0	36	0	0	0	0	1
23	36	0	8	0	0	0	0	0	1	0	0	1
24	36	0	8	0	0	1	0	0	1	0	0	1
25	36	0	8	0	0	1	0	1	1	0	0	1
26	36	0	8	0	0	1	1	0	1	0	0	1
27	36	0	8	0	0	0	1	0	1	0	0	1
28	36	0	8	0	0	1	0	0	1	1	0	1
29	36	0	8	0	0	1	1	0	1	1	0	1
30	36	0	8	0	0	1	1	0	1	0	1	1
31	36	0	8	0	0	1	1	0	1	1	1	1
32	36	0	8	0	0	0	0	0	1	1	0	1
33	36	0	8	0	0	0	0	0	1	0	1	1
34	72	0	16	0	0	0	0	0	0	0	0	1
35	72	0	16	0	1	0	0	0	0	0	0	1
36	72	0	16	0	0	0	72	0	0	0	0	1
37	72	0	16	0	1	0	72	0	0	0	0	1
38	72	0	16	0	0	0	0	0	1	0	0	1
39	72	0	16	0	0	1	0	0	1	0	0	1
40	72	0	16	0	0	1	0	1	1	0	0	1
41	72	0	16	0	0	1	1	0	1	0	0	1
42	72	0	16	0	0	0	1	0	1	0	0	1
43	72	0	16	0	0	1	0	0	1	1	0	1
44	72	0	16	0	0	1	1	0	1	1	0	1
45	72	0	16	0	0	1	1	0	1	0	1	1
46	72	0	16	0	0	1	1	0	1	1	1	1
47	72	0	16	0	0	0	0	0	1	1	0	1
48	72	0	16	0	0	0	0	0	1	0	1	1

Bibliography

Air Force Civil Engineer Center. (2017a). AFIT 466 – Meters and AMRS.

Air Force Civil Engineer Center. (2017b). *Air Force Civil Engineer Energy Savings Performance Contracts (ESPC) Playbook*.

Air Force Civil Engineer Center. (2017c). *Energy Resilience & Conservation Investment Program Explained*.

Air Force Civil Engineer Center. Unified Facilities Guide Specifications - 26 27 13.10 30 - Electric Meters (2017).

Air Force Civil Engineer Center. (2017e). Utilities Scoring Sheet guide.

Air Force Civil Engineer Center. (2017f). Utility Privitization. Retrieved August 17, 2017, from http://www.afcec.af.mil/Home/Energy/Utilities-Privatization/

Air Force Civil Engineer Center Public Affairs. (2017, August 1). AEMR is now AEMRR - 2017 Annual Energy Management and Resilience Report. *AFCEC News*. Retrieved from http://www.afcec.af.mil/News/Article-Display/Article/1263629/aemr-is-now-aemrr-2017-annual-energy-management-and-resilience-report/

Bahmanyar, A., Estebsari, A., Pons, E., Patti, E., Jamali, S., Bompard, E., & Acquaviva, A. (2016). Fast fault location for fast restoration of smart electrical distribution grids. *IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016 - Proceedings*, 2–7. https://doi.org/10.1109/ISC2.2016.07580741

Bahmanyar, A., Jamali, S., Estebsari, A., & Bompard, E. (2017). A comparison framework for distribution system outage and fault location methods. *Electric Power Systems Research*, *145*, 19–34.

Benson, B. (2017). Electrical Sub-AMP Prioritization. *Air Force Civil Engineer Center*.

Booth, S., Barnett, J., Burman, K., Hambrick, J., & Westby, R. (2010). Net Zero Energy Military Installations : A Guide to Assessment and Planning. *NREL Technical Report*, (August).

Brost, D. G. (2013). *Successfully Implementing Net-Zero Energy Policy through the Air Force Military Construction Program*. Air Force Institue of Technology.

Bush, G. W. Executive Order 13327 - Federal Real Property Asset Management (2004).

Carnley, F. (2017). AMRS Meters. *Air Force Civil Engineer Center*. Air Force Civil Engineer Center.

CH2MHILL & Clark-Nexsen. (2016). *Naval Facilities Engineering Command Microgrid Design and Reference Guide*.

Chen, P.-C., Dokic, T., & Kezunovic, M. (2014). The use of big data for outage management in distribution systems. *International Conference on Electricity Distribution (CIRED) Workshop*, (406), 11–12.

Defense Acquisition University. (2017). Types of Funds. Retrieved August 17, 2017, from https://dap.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=9f96cbe4-ed8f-4d20-94c9-b89130c0eb70

Defense Logistics Agency. (2017). *Defense Logistics Agency Fact Sheet*.
Department of Defense. (2004). Chapter 8: Facilities Sustainment, Restoration and Modernization. In *DoD Financial Mangement* (Vol. 2B).
Department of Defense. (2013). DoD Utilities Meter Policy.
Department of Defense. (2014). UFC 1-200-02: HIGH PERFORMANCE AND SUSTAINABLE BUILDING REQUIREMENTS. Retrieved from http://www.wbdg.org/ccb/DOD/UFC/ufc_1_200_02.pdf
Department of Defense. (2015a). DoD Instruction 4140.25: DoD Management Policy for Energy Commodities and Related Services. *Department of Defense Instruction*, (4140). Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/850001_2014.pdf
Department of Defense. (2015b). *Operations and Maintenance Overview: Fiscal Year 2016 Budget Estimates*.
Department of Defense. (2015c). *Technology Transition Final Public Report Smart Power Infrastructure Demonstration for Energy Reliability and Security (SPIDERS) Joint Capability Technology Demonstration (JCTD)*.
Department of Defense. (2016a). DoD Instruction 4170.11: Installation Energy Management, (8510), 1–49.
Department of Defense. (2016b). *Strategic Sustainability Performance Plan*. *DoD*.
Department of Energy. (2014). How Microgrids Work. Retrieved June 17, 2017, from https://energy.gov/articles/how-microgrids-work
Department of the Air Force. (2016a). AFI 32-1032: Planning and Programming Appropriated Fund Maintenance, Repair, and Construction Projects, (1), 1–89. Retrieved from https://www.wbdg.org/FFC/AF/AFI/afi_32_1032.pdf
Department of the Air Force. (2016b). *AFPAM 32-10144, Implementing Utilities at U.S. Air Force Installations*.
Department of the Air Force. (2016c). *AFPD 90-17: Energy And Water Management Compiance*.
Department of the Air Force. (2017a). *Air Force Meter Data Management Plan (MDMP)* (Vol. 2017).
Department of the Air Force. (2017b). *United States Air Force Energy Flight Plan*.
Department of the Army. (2007). *TM 5-698-1 - RELIABILITY / AVAILABILITY OF ELECTRICAL & MECHANICAL SYSTEMS FOR COMMAND , CONTROL , COMMUNICATIONS , COMPUTER , INTELLIGENCE , SURVEILLANCE AND RECONNAISSANCE (C4ISR) FACILITIES*.
Ersoz, I., & Colak, U. (2016). Combined cooling, heat and power planning under uncertainty. *Energy*, *109*, 1016–1025. https://doi.org/10.1016/j.energy.2016.04.071
Eto, J. H., & Lacommare, K. H. (2008). Tracking the Reliability of the U.S. Electric Power System: An Assessment of Publicly Available Information Reported to State Public Utility Commissions. *LBNL Reports*, *LBNL-1092E*(October). Retrieved from http://certs.lbl.gov/pdf/lbnl1092e-puc-reliability-data.pdf
Fisher, G. (2017). Pecan Street Database Data Director. *Pecan Street*. Pecan Street Inc.

Formica, T., & Pecht, M. (2017). Return on investment analysis and simulation of a 9.12kW (kW) solar photovoltaic system. *Solar Energy*, *144*, 629–634. https://doi.org/10.1016/j.solener.2017.01.069

Gerdes, D. (2017). AMRS Meters. *Air Force Civil Engineer Center*.

Golnas, A. (2012). PV System Reliability : An Operator's Perspective. *Photovoltaic Specialists Conference (PVSC). IEEE 38th*, 1–6. https://doi.org/10.1109/PVSC-Vol2.2012.6656744

Grussing, M. N., Gunderson, S., Canfield, M., Falconer, E., Antelman, A., & Hunter, S. L. (2010). Development of the Army Facility Mission Dependency Index for Infrastructure Asset Management, (September).

Gururajapathy, S. S., Mokhlis, H., & Illias, H. A. (2017). Fault location and detection techniques in power distribution systems with distributed generation: A review. *Renewable and Sustainable Energy Reviews*, *74*(March), 949–958. https://doi.org/10.1016/j.rser.2017.03.021

Institute of Electrical and Electronic Engineers. (2007). *IEEE Gold Book: Recommended Practices for the Design of Reliable Industrial and Commercial Power Systems* (Vol. Std 493).

Institute of Electrical and Electronic Engineers. (2011). *IEEE Guide for Design, Operation, and Integration of Distributed Resource Island Systems with Electric Power Systems IEEE Standards Coordinating Committee 21 Sponsored by the*.

Institute of Electrical and Electronic Engineers. (2012). *IEEE Guide for Electric Power Distribution Reliability Indices* (Vol. Std 1366).

Jamali, S., & Bahmanyar, A. (2016). A new fault location method for distribution networks using sparse measurements. *International Journal of Electrical Power and Energy Systems*, *81*, 459–468. https://doi.org/10.1016/j.ijepes.2016.02.046

Jamali, S., Bahmanyar, A., & Bompard, E. (2017). Fault location method for distribution networks using smart meters. *Measurement*, *102*, 150–157. https://doi.org/10.1016/j.measurement.2017.02.008

Judson, N., & Pina, A. (2017). Personal Interview with Massachusetts Institute of Technology Lincoln Laboratory Energy Division. Boston.

Judson, N., Pina, A. L., Dydek, E. V., Castillo, A. S., & Van Broekhoven, S. B. (2016). *UNCLASSIFIED // FOR OFFICIAL USE ONLY Application of a Resilience Framework to Military Installations: A Methodology for Energy Resilience Business Case Decisions*. *Massachusetts Institute of Technology Lincoln Laboratory*.

Krajnak, D. J. (2000). Faulted circuit indicators and system reliability. *2000 Rural Electric Power Conference. Papers Presented at the 44th Annual Conference (Cat. No.00CH37071)*, A4/1-A4/4. https://doi.org/10.1109/REPCON.2000.848037

LaCommare, K. H., & Eto, J. H. (2004). *Understanding the Cost of Power Interruption to US Electric Consumers*. *Ernest Orlando Lawrence Berkeley National Laboratory*.

LaCommare, K., Larsen, P., & Eto, J. (2016). Evaluating Proposed Investments in Power System Reliability and Resilience: Preliminary Results from Interviews with Public Utility Commission Staff, (January). Retrieved from

https://pubarchive.lbl.gov/islandora/object/ir%3A1006971/
Lawton, L., Sullivan, M., Liere, K. Van, & Katz, A. (2003). *A Framework and Review of Customer Outage Costs: Integration and Analysis of Electric Utility Outage Cost Surveys*. *Ernest Orlando Lawrence Berkeley National Laboratory*.
Li, W., Chen, W., & Guo, C. (2017). Optimal Placement of Fault Indicators in Distribution System Using PSO Algorithm, 375–380.
Liu, X., Golab, L., Golab, W., Ilyas, I. F., & Jin, S. (2016). Smart Meter Data Analytics: Systems, Algorithms, and Benchmarking. *ACM Trans. Database Syst*, *42*(2), 1–39. https://doi.org/10.1145/3004295
Marqusee, J., Schultz, C., & Robyn, D. (2017). Power Begins at Home: Assured Energy for U.S. Military Bases. *Noblis*.
Meier, A., Nordman, B., Busch, J., Payne, C., Brown, R., Homan, G., … Webber, C. (2008). Low-Power Mode Energy Consumption in California Homes. *Lawrence Berkeley National Laboratory*, (September).
Menchafou, Youssef; Markhi, Hassan; Zahri, Mustapha; Habibi, M. (2015). Impact of distributed generation integration in electric power distribution systems on fault location methods. *3rd International Renewable and Sustainable Energy Conference (IRSEC)*, (1998). https://doi.org/10.1109/IRSEC.2015.7455137
Mills, P., Mane, M., Kuhn, K., Narayanan, A., Powers, J. D., Buryk, P., … Lynch, K. F. (2017). *Articulating the Effects of Infrastructure Resourcing on Air Force Missions*.
Morgan, E. R., Valentine, S., Blomberg, C. A., Limpaecher, E. R., & Dydek, E. V. (2016). *Boston Community Energy Study – Zonal Analysis for Urban Microgrids*. *Massachusetts Institute of Technology Lincoln Laboratory*. Lexington, Massachusetts.
National Electrical Manufacturers Association. (2017). Smart Meters Can Reduce Power Outages and Restoration Time. Retrieved September 17, 2017, from https://www.nema.org/Storm-Disaster-Recovery/Smart-Grid-Solutions/Pages/Smart-Meters-Can-Reduce-Power-Outages-and-Restoration-Time.aspx
National Oceanic and Atmospheric Association. (2018). Average Monthly Temperature.
Obrecht, M., & Denac, M. (2016). Technology forecast of sustainable energy development prospects. *Futures*, *84*, 12–22. https://doi.org/10.1016/j.futures.2016.09.002
Office of the Assistant Secretary of Defense for Energy Installations and Environment. (2014). Department of Defense Annual Energy Management Report (Fiscal Year 2013), (May), 3–4.
Office of the Assistant Secretary of Defense for Energy Installations and Environment. (2015). Department of Defense Annual Energy Management Report (Fiscal Year 2014), (May), 3–4.
Office of the Assistant Secretary of Defense for Energy Installations and Environment. (2016a). 2016 Operational Energy Strategy.
Office of the Assistant Secretary of Defense for Energy Installations and Environment. (2016b). Department of Defense Annual Energy Management Report (Fiscal Year

2015). Retrieved from http://www.acq.osd.mil/eie/Library.html
Office of the Assistant Secretary of Defense for Energy Installations and Environment. (2017). Department of Defense Annual Energy Management Report (Fiscal Year 2016), (July), 1–94.
Olive, M. C. (2017). *An Energy Benchmarking Categorization Scheme and Consumption Data Validation for Air Force Facilities*. Air Force Institute of Technology.
Parker, S. A., Boyd, B. K., Hunt, W. D., Stoughton, K. L. M., Fowler, K. M., Koehler, T. M., … Sullivan, G. P. (2015). Metering Best Practices: A Guide to Achieving Utility Resource Efficiency, Release 3.0, (March).
Pecan Street. (2017). Pecan Street: About Us. Retrieved from http://www.pecanstreet.org/about/
Pecan Street Dataport. (2017). Database Table and View Catalog, 1–4. Retrieved from https://dataport.pecanstreet.org/Data/Dataresources
Pereira, A., Pérez-yauli, F., & Quilumba, F. L. (2017). Three-phase recloser time delays determination in 138 kV and 46 kV lines of the Empresa Eléctrica Quito.
Pew Charitable Trusts. (2017). Power Begins at Home: Assured energy for U.S. military bases. Retrieved September 17, 2017, from http://www.pewtrusts.org/en/about/events/2017/power-begins-at-home
Ramos, R. D. (2017). ERCIP and Energy Project Guidance. *Air Force Civil Engineer Center*. Air Force Civil Engineer Center.
Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., … Tarantola, S. (2008). *Global Sensitivity Analysis: The Primer*. *Handbook of Uncertainty Quantification*. John Wiley & Sons Ltd.
Secretary of the Air Force. (2017). 2017 Air Force Priorities.
Shah, S. (2014). Final Report: Rapid Building Assessment Project. *Environmental Security Technology Certification Program*.
Society of American Military Engineers. (2017). Joint Engineer Training and Conference: Session Descriptions and Speakers. Retrieved June 17, 2017, from http://s3.goeshow.com/same/jetc/2017/conference_sessions.cfm
Software Engineering Institute. (2011). Smart Grid Maturity Model: Model Definition, version 1.2 (CMU/SEI-2011-TR-025), 84. Retrieved from https://www.sei.cmu.edu/reports/11tr025.pdf
Sullivan, M. J., & Keane, D. M. (1995). Outage Cost Estimation Guidebook. *Electric Power Research Institute*, *3*(3).
The Department of Energy Office of Electricity Delivery and Energy Reliability. (2010). Last Gasp Message - Outage Notification. Department of Energy.
The United States Congress. Energy Policy Act of 2005 (2005). Washington D.C. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4042170
The United States Congress. (2007). Energy Independence and Security Act of 2007, H.R. 6--28-30. Retrieved from https://www.gpo.gov/fdsys/pkg/BILLS-110hr6enr/pdf/BILLS-110hr6enr.pdf
The United States Congress. (2017a). 10 USC 2688: Utility systems: conveyance

authority.
The United States Congress. 10 USC 2925: Annual Department of Defense Energy Management Reports (2017).
Ton, D. T., & Smith, M. A. (2012). The U.S. Department of Energy's Microgrid Initiative. *The Electricity Journal*, *25*(8), 84–94.
Tram, H. (2008). Technical and operation considerations in using smart metering for outage management. *Transmission and Distribution Exposition Conference: 2008 IEEE PES Powering Toward the Future, PIMS 2008*, 1–3. https://doi.org/10.1109/TDC.2008.4517273
Under Secretary of Defense for Acquisition Technology and Logistics. (2013). Standardizing Facility Condition Assessments. Retrieved from http://www.acq.osd.mil/ie/fim/library/DoD Facility Inspection Policy.pdf
United States Department of the Interior: Bureau of Reclamation. (2015). Grand Coulee Dam Statistics and Facts. Retrieved from https://www.usbr.gov/pn/grandcoulee/about/index.html
Van Broekhoven, S. B., Judson, N., Nguyen, S. V. T., & Ross, W. D. (2012). *Microgrid Study : Energy Security for DoD Installations*. *Massachusetts Institute of Technology Lincoln Laboratory*. Lexington, Massachusetts.
Warren, C. A., Pearson, D. J., & Sheehan, M. T. (2003). A nationwide survey of recorded information used for calculating distribution reliability indices. *IEEE Transactions on Power Delivery*, *18*(2), 449–453. https://doi.org/10.1109/TPWRD.2002.803693
Watley, T. (2017). AFIT Research on Microgrids. *Air Force Civil Engineer Center*, 1–4.
Zhou, K., Fu, C., & Yang, S. (2016). Big data driven smart energy management: From big data to big insights. *Renewable and Sustainable Energy Reviews*, *56*, 215–225. https://doi.org/10.1016/j.rser.2015.11.050

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
22-03-2018	Master's Thesis	Sept 16 - Mar 18

4. TITLE AND SUBTITLE
Influencing Effective Electrical Distribution Modernization Through Advanced Metering

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Erickson, Jared R, Capt

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT-ENV-MS-18M-201

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Civil Engineer Center
Maj Josh Aldred
139 Barnes Drive
Tyndall AFB, FL 32403
josh.aldred@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
AFCEC/CN

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
In 2013, the Department of Defense (DoD) required its services to implement advanced metering for the purpose of reducing energy usage (Department of Defense, 2013). Additionally, the DoD has aimed to improve its ability to assure a continuous energy supply to all of its installations. This study investigated processes for applying advanced meters on Air Force bases to increase energy assurance. This study also identified strategies for using advanced meters to influence infrastructure funding. This was accomplished through the use of extensive advanced meter data. The data was analyzed for outages and a procedure was created to locate outages in energy usage datasets by using means and standard deviations. Advanced meters with more frequent data collection were able to locate outages easier than meters with less frequent data collection. Advanced meters do not only reduce energy usage, but they also have the ability to report outages. By collecting outage data, funding can be applied to the least reliable electrical infrastructure.

15. SUBJECT TERMS
Advanced Meters, Outage Detection, Energy, Electrical Distribution, Energy Resilience

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			JEFFREY C. PARR, Lt Col, USAF, AFIT/ENV
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	171	937-255-3636 x4709 Jeffrey.Parr@afit.edu

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18
Adobe Professional 7.0